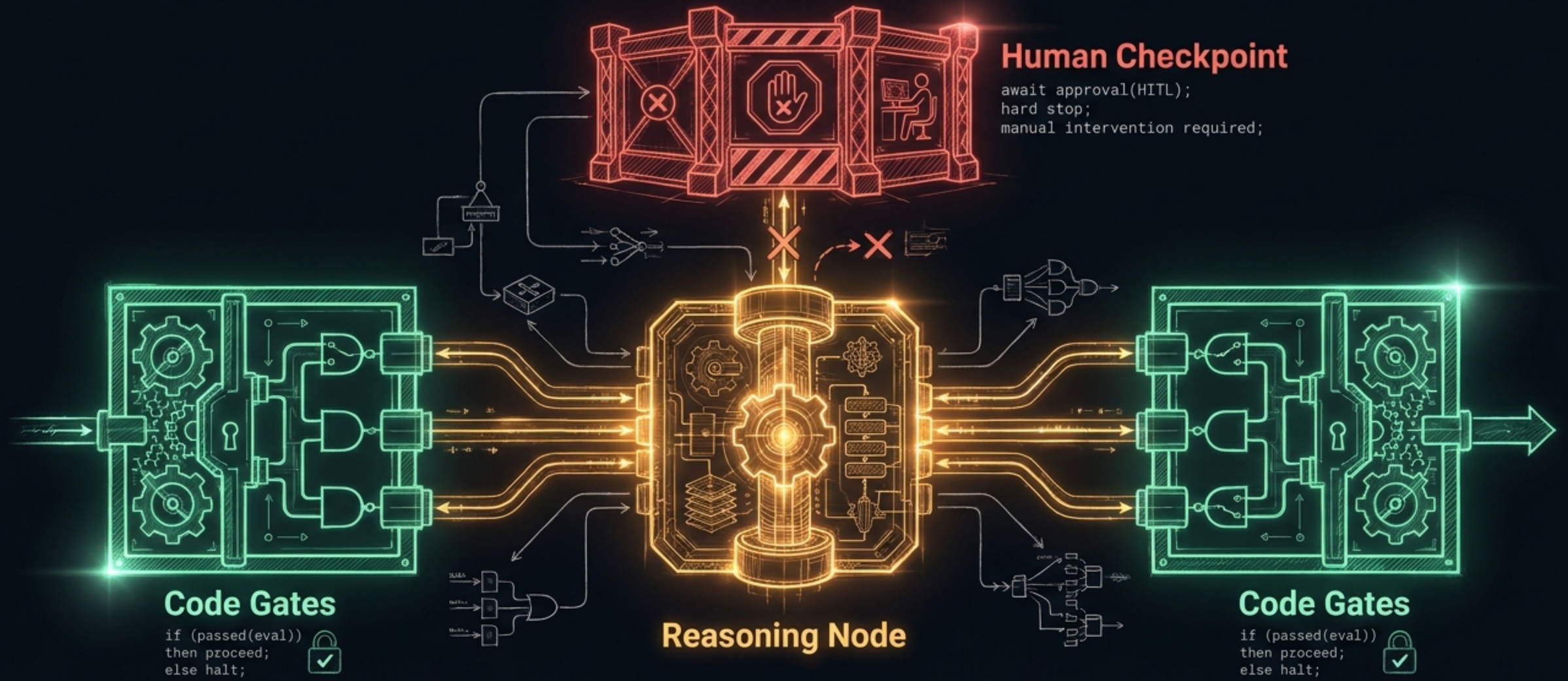
Agentic Design Patterns for Production-Grade LLM Systems

Architectural Blueprints for Orchestration, Routing, and Autonomous Memory.



Presented as an Engineering Playbook.

LLMs are Probabilistic Reasoning Engines, Not Execution Architectures

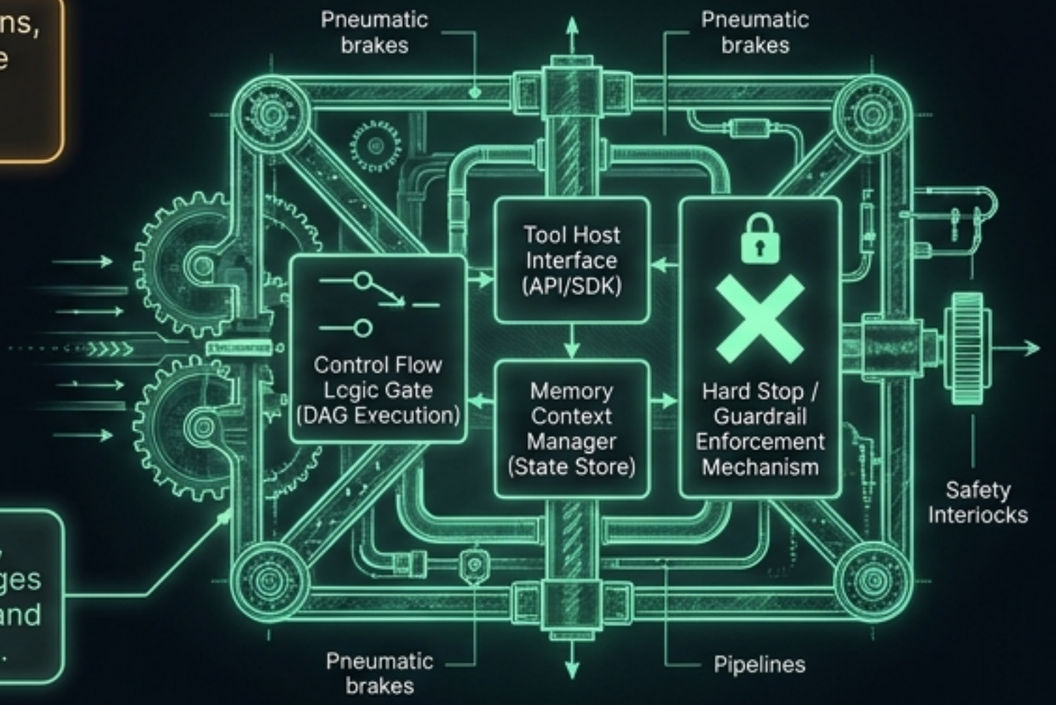
THE ENGINE



Can propose actions, but cannot enforce invariants or hold state.

Owns control flow, hosts tools, manages memory context, and applies hard stops.

THE HARNESS



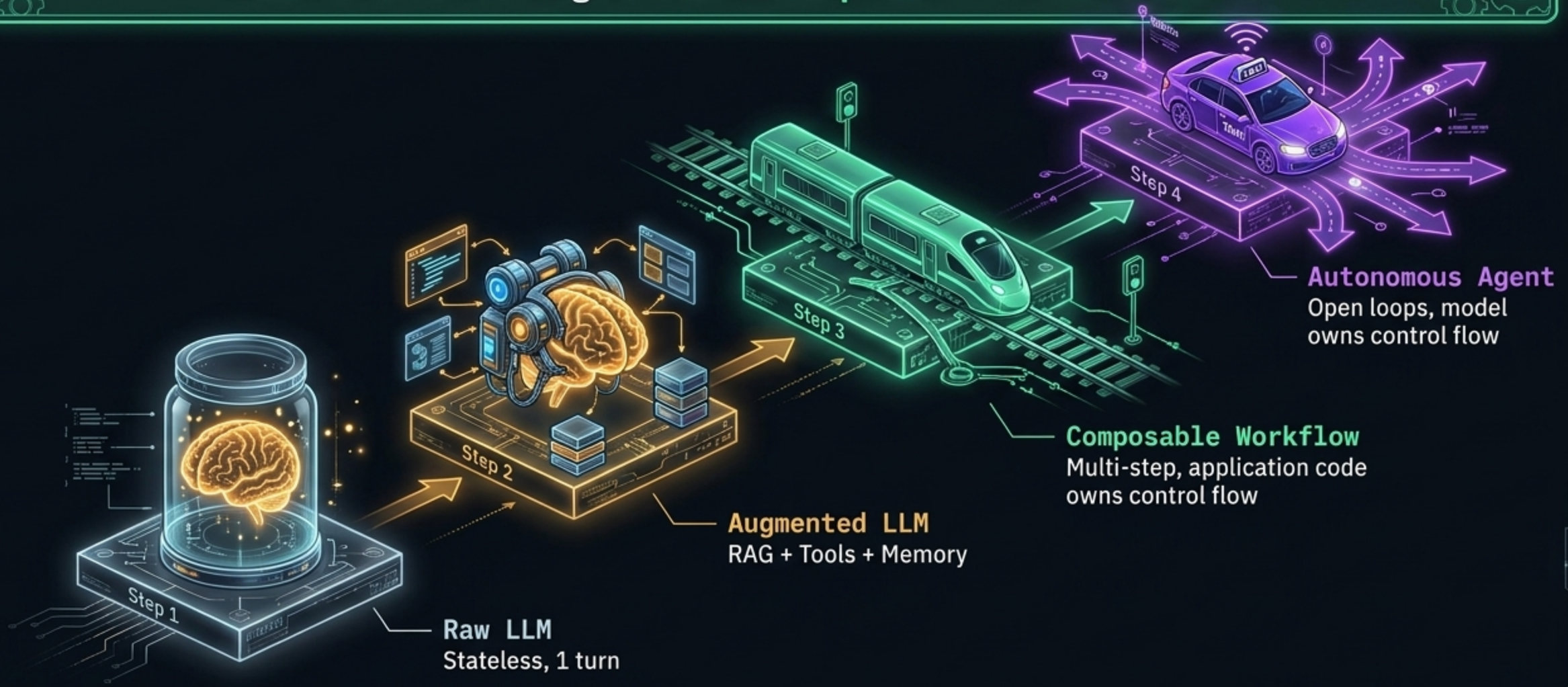
Stateless Probabilistic Token Predictor

Deterministic Architecture

Architectural patterns define who owns the control flow (Mint Green), where authority sits, and how context stays clean as tasks scale.

Scale Complexity Only When Empirical Evaluations Prove Simpler Methods Fail

Rule 0: Start with the **simplest architecture** that passes your evaluation suite. Never build an autonomous agent when a **composable workflow** suffices.



Every Enterprise Workflow is a Hybrid of Three Execution Regimes

	Regime 1: Deterministic Code	Regime 2: Probabilistic Inference	Regime 3: Human-in-the-Loop (HITL)
Tasks	Exact arithmetic, RBAC, schema validation, PII redaction.	Unstructured synthesis, semantic classification, fuzzy matching.	High-blast-radius irreversible actions, wire refunds, policy exceptions.
Why	LLMs introduce silent tail-risk degradation.	Rigid code shatters on phrasing variations.	Models cannot bear financial or legal accountability.



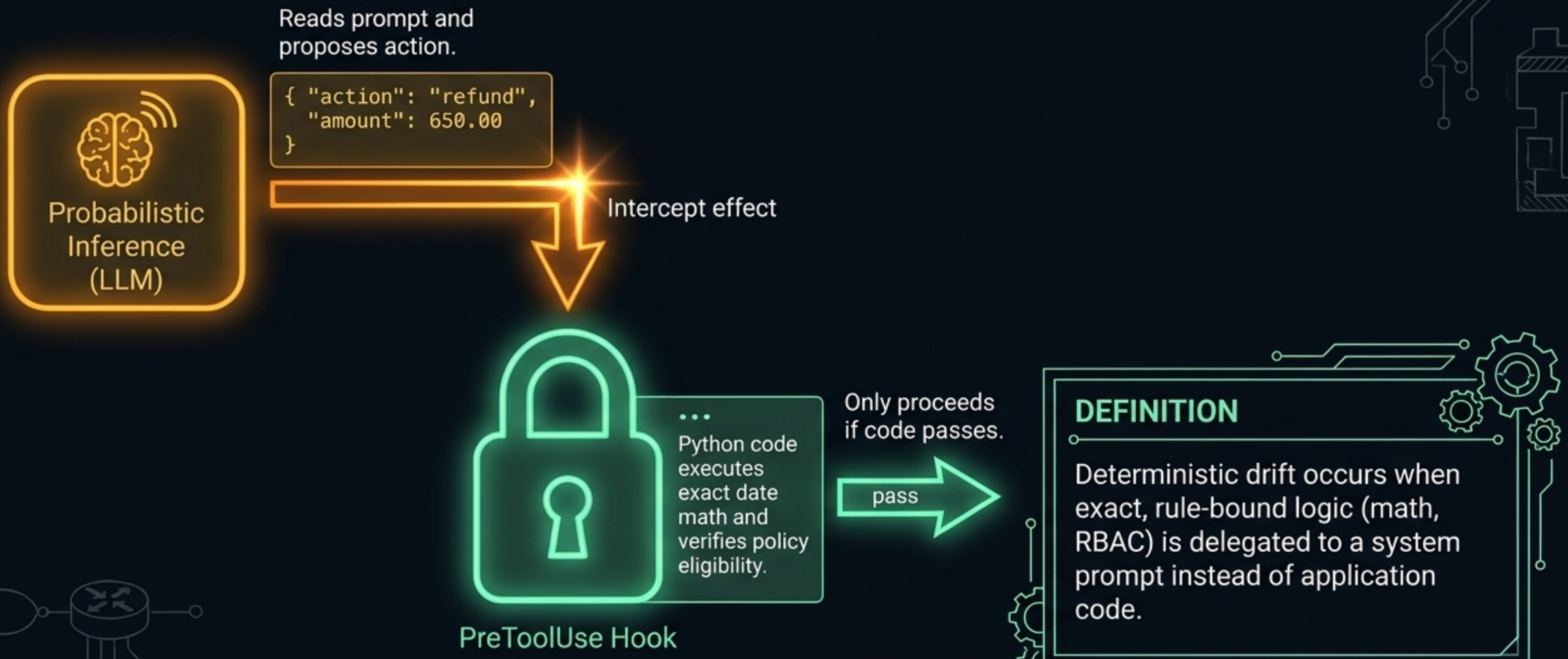




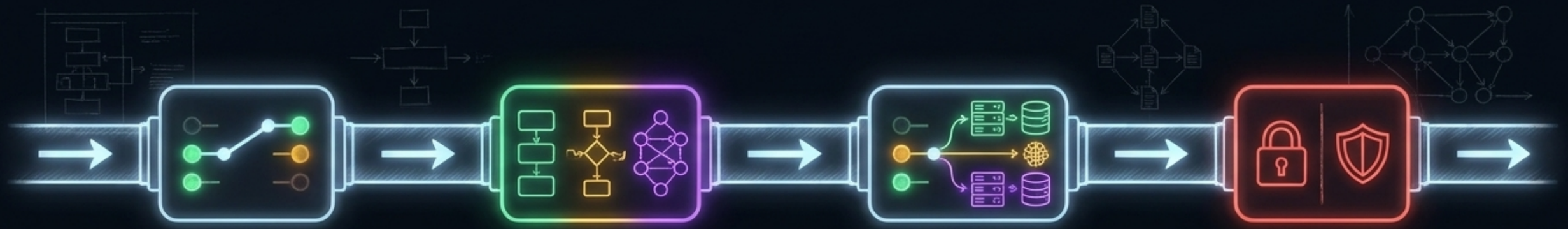


Never assign Regime 1 or Regime 3 responsibilities to Regime 2.

Prevent Deterministic Drift by Forcing Hard Invariants into Application Code



The Sequential Decision Framework Maps Task Constraints to Architectural Shapes

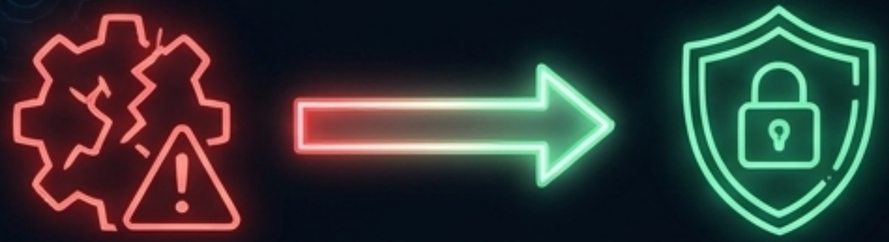


1) Triage Execution Regimes
2) Select Work-Shape
3) Route to Model Tier
4) Wrap in Guardrails

	Single Call	Workflow	Autonomous Agent
Task Predictability	1 step	Known Decomposition (K steps)	Unknown Emergent (N steps)
Latency Budget	Sub-second	~3s to ~15s	Seconds to minutes+
Token Cost	Base rate	Bounded multiplier	High exponential ceiling (10-15x)

Agentic Architectures Must Defend Against Inherent Probabilistic Flaws

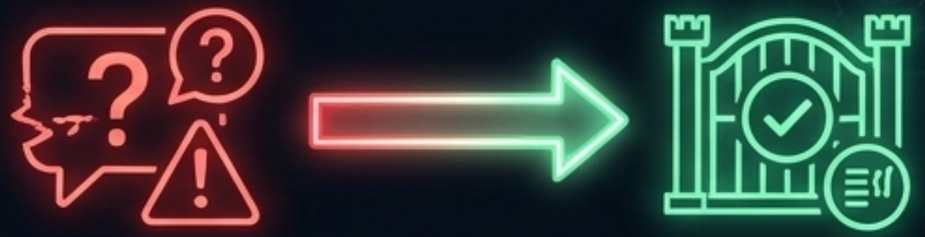
Non-determinism



Flaw: Identical prompts yield variance.

Mitigation: Token-level strict JSON schemas and structural rubrics.

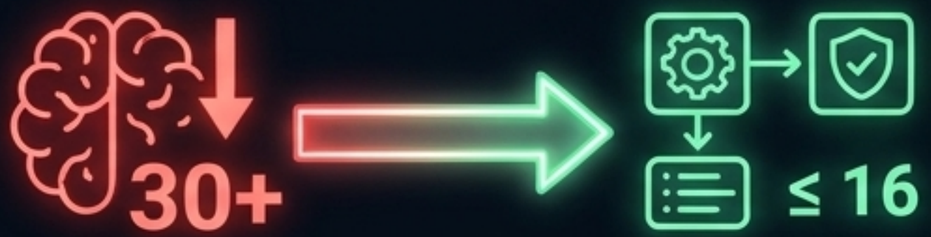
Plausible-Sounding Hallucinations



Flaw: Empty tool outputs generate confident lies.

Mitigation: Pre-LLM epistemic quality gates and deterministic citation verification.

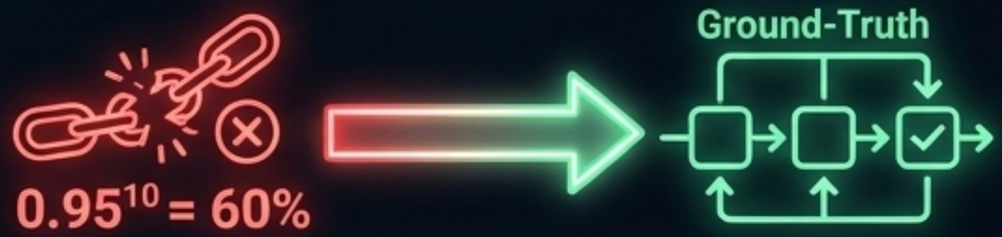
Context Sensitivity



Flaw: Mid-window recall drops with 30+ rules.

Mitigation: Separate generation from verification; cap tool catalogs at <= 16

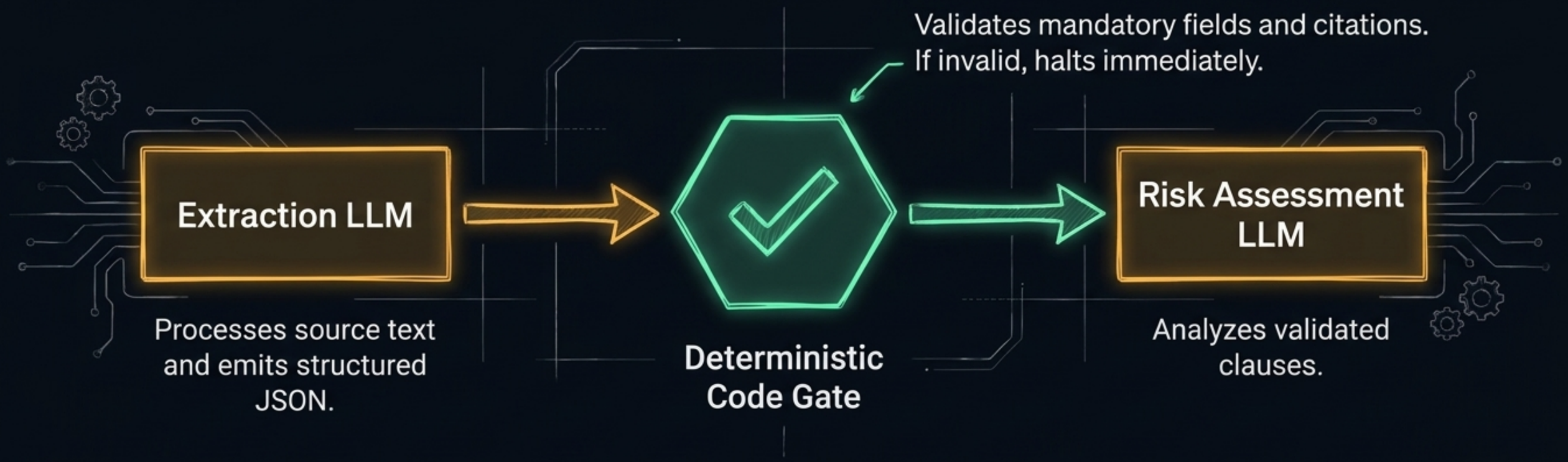
Compounding Multi-Step Errors



Flaw: Unverified chains fail exponentially (0.95^10 = 60%).

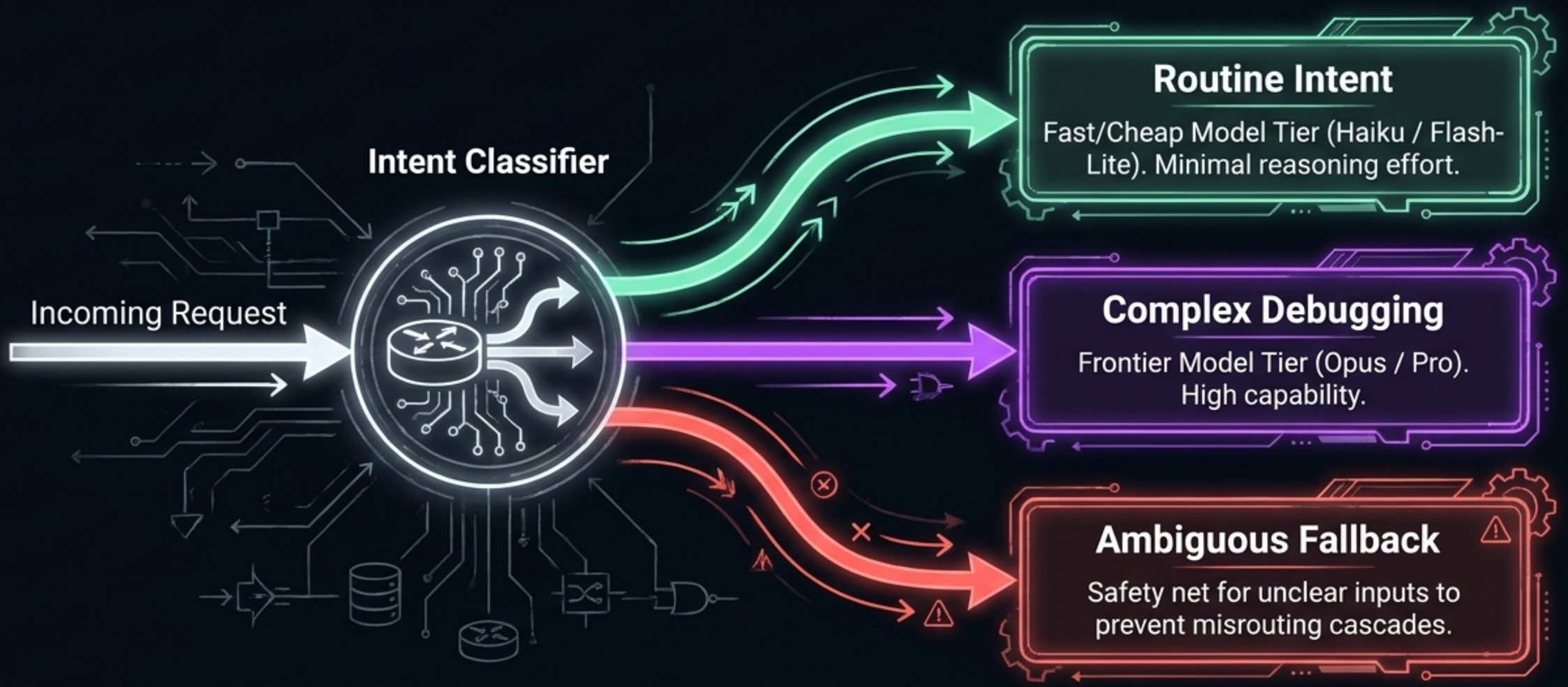
Mitigation: Insert ground-truth environmental feedback loops after every step.

Canonical Pattern 1: Prompt Chaining Secures Sequential Steps with Code Gates



Core Concept: This pattern ensures $0 downstream spend if extraction fails early. It decomposes complex tasks into narrower, focused LLM calls where application code owns the sequence.

Canonical Pattern 2: Routing Dispatches Intents to Specialized Handlers and Tiers

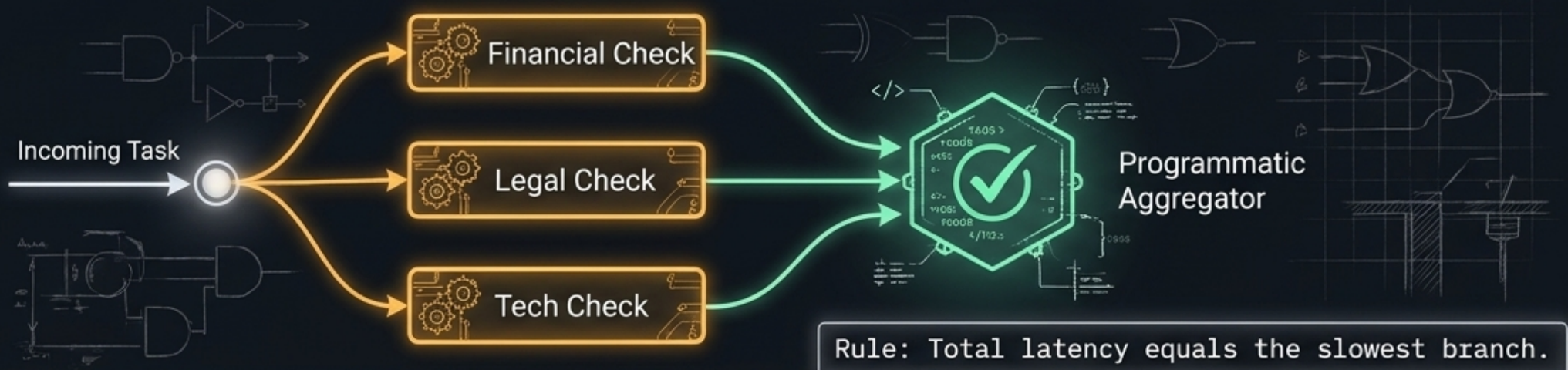


Takeaway: Routing beats a monolith prompt by eliminating instruction clash and reserving expensive frontier models only for tasks that demand them.

Canonical Pattern 3: Parallelization Optimizes for Wall-Clock Speed or High-Confidence Consensus

Sectioning

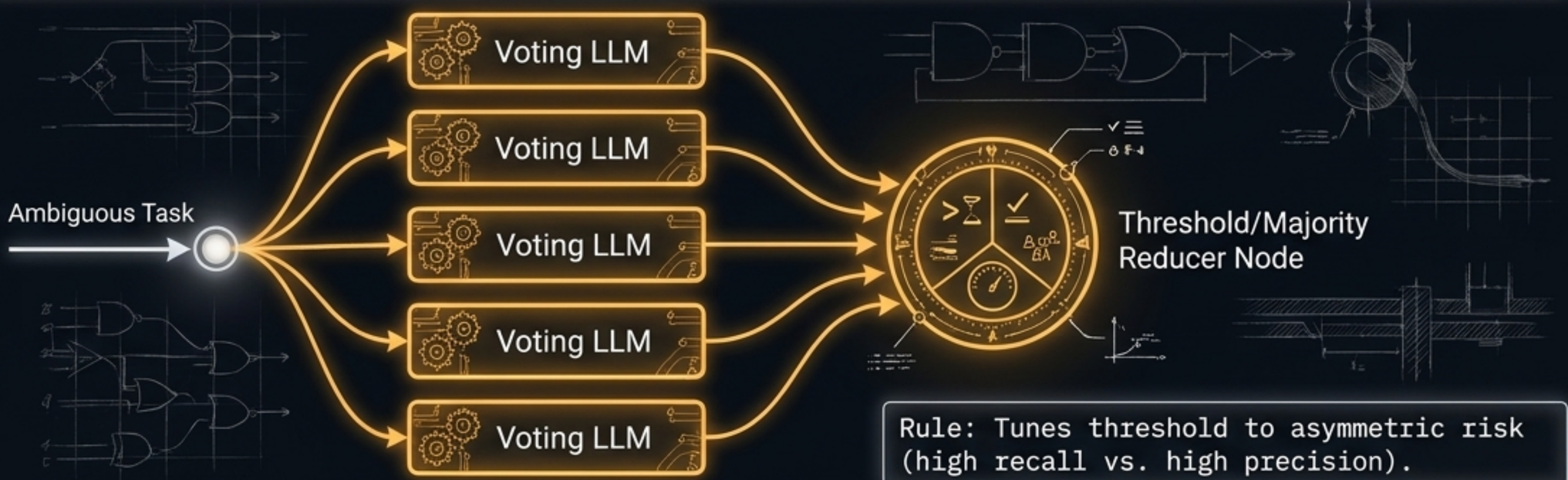




Rule: Total latency equals the slowest branch.

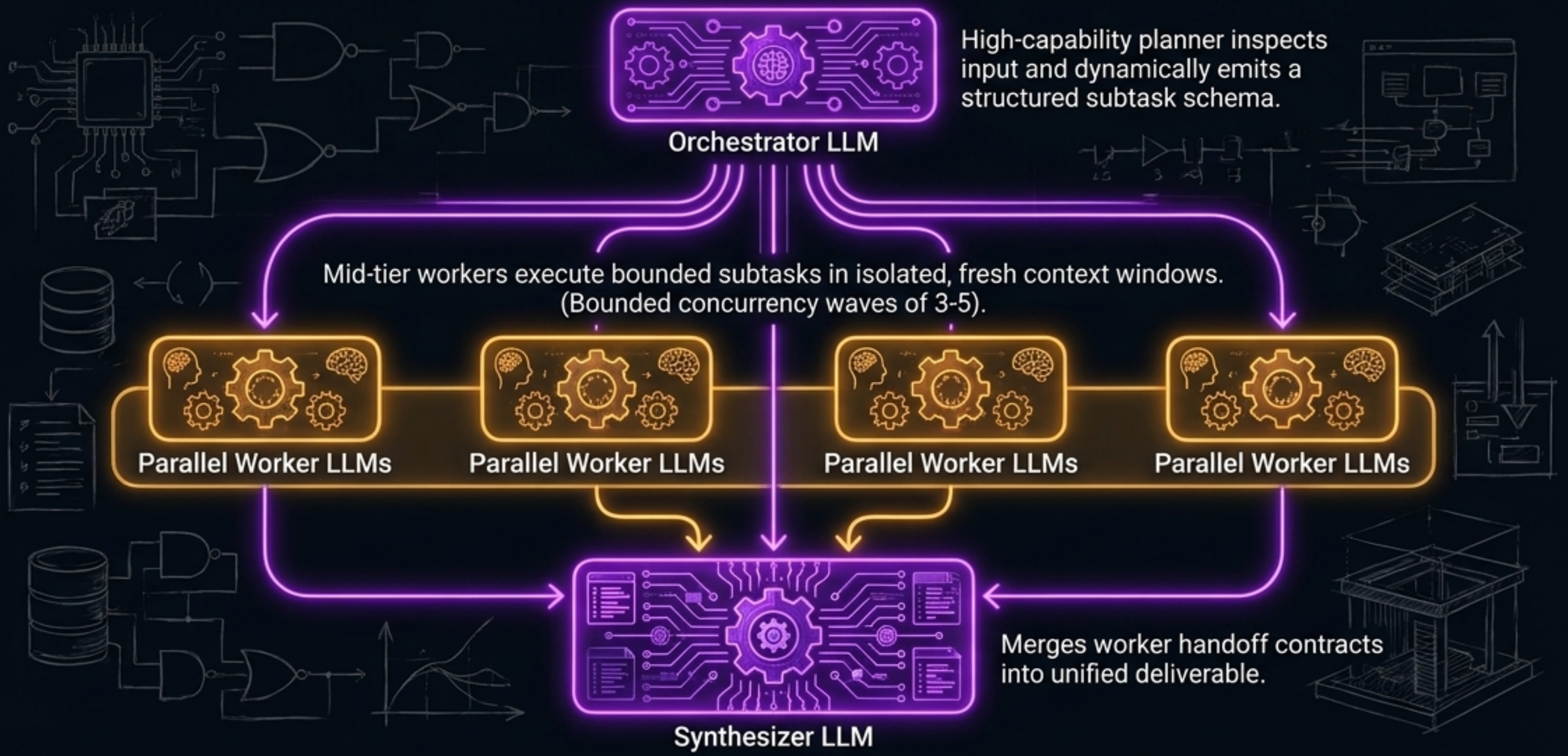
Voting





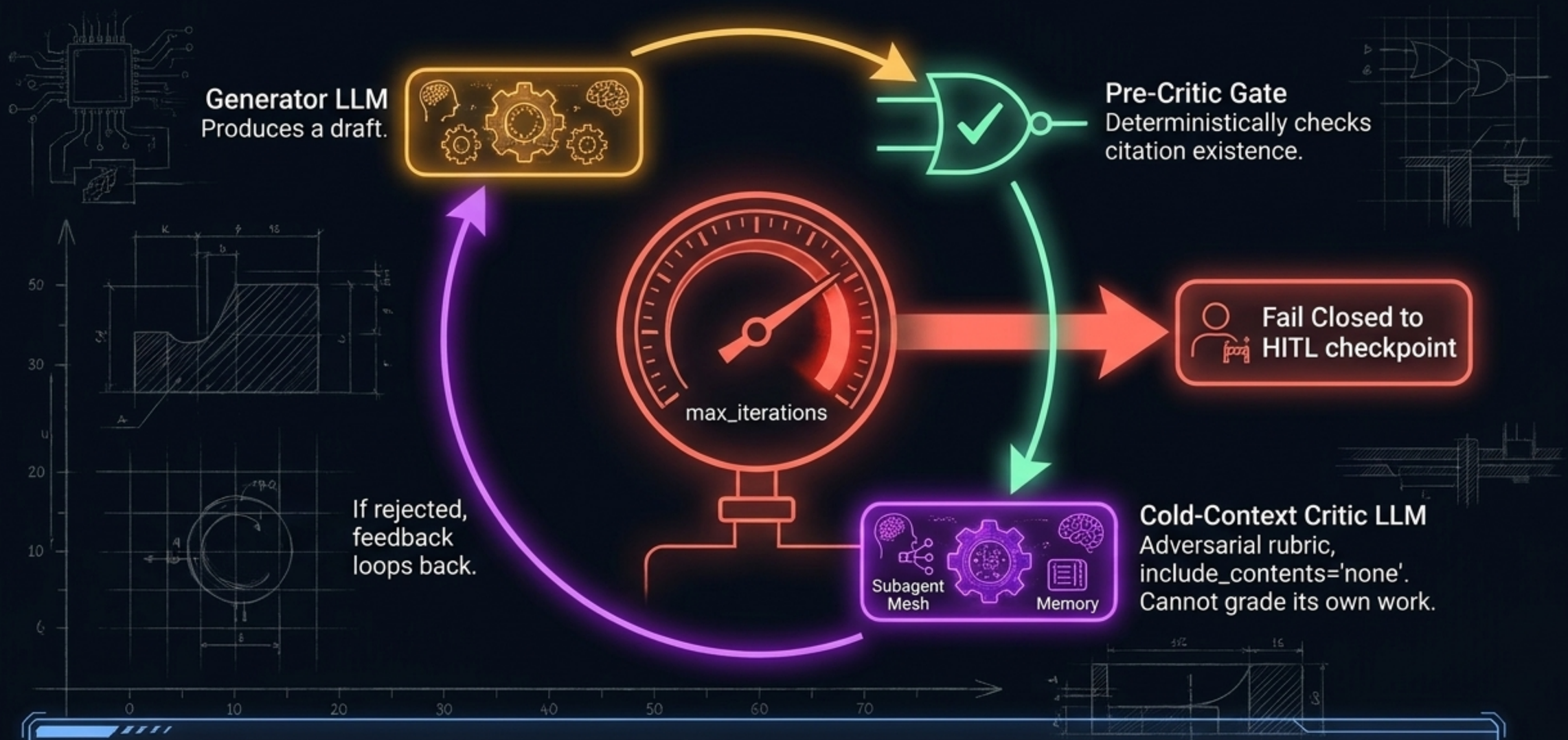
Rule: Tunes threshold to asymmetric risk (high recall vs. high precision).

Canonical Pattern 4: Orchestrator-Workers Dynamically Decompose Unpredictable Task Graphs



Key Distinction: Uniike standard Parallelization, the exact number and nature of branches are determined at runtime, not hardcoded in advance.

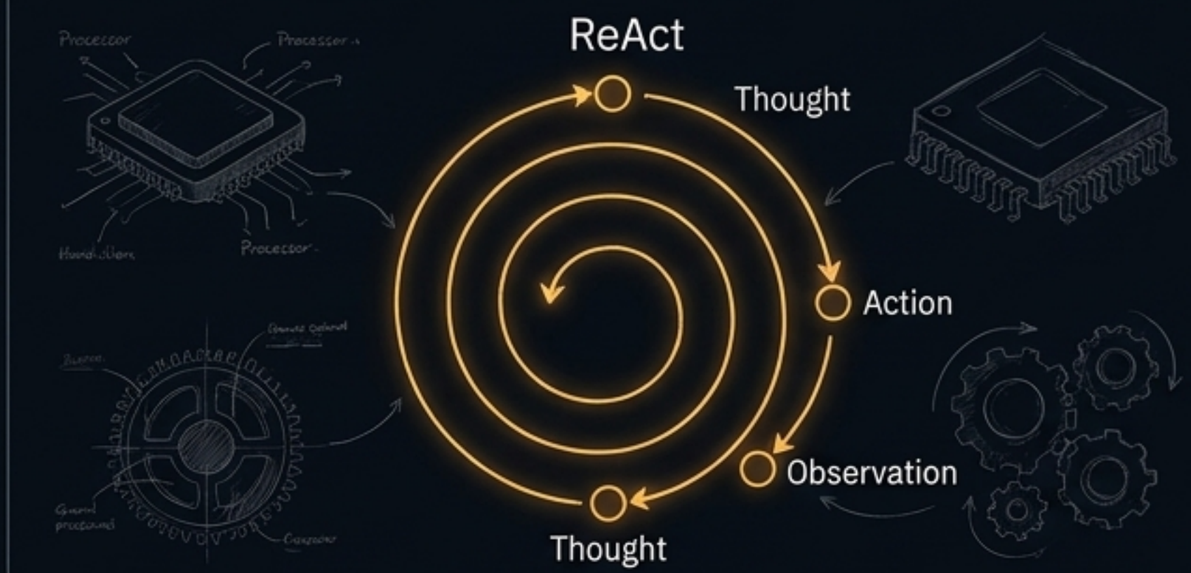
Canonical Pattern 5: Evaluator-Optimizer Loops Enforce the Separation of Authority

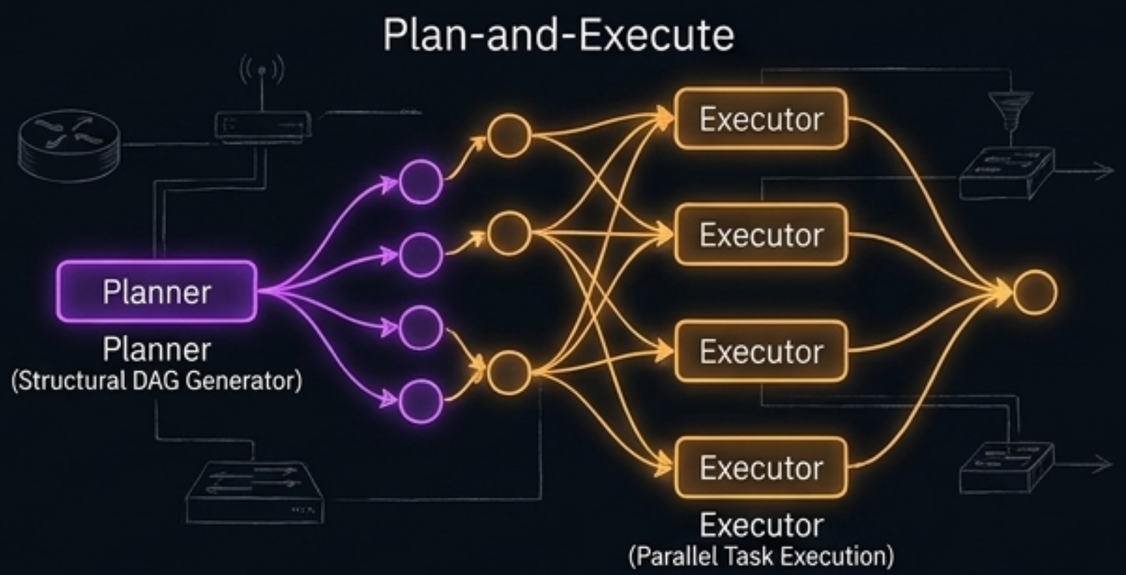


Core Rule: Never ask the same prompt in the same turn to draft an answer and grade its own compliance—it will rationalize its mistakes.

Autonomous Loop Paradigms Demand Strict Execution and Termination Hygiene

Loop Architectures





The 5 Guardrails



Explicit completion criteria
(Ground-truth exit)

Hard turn ceilings
(max_turns limiter)

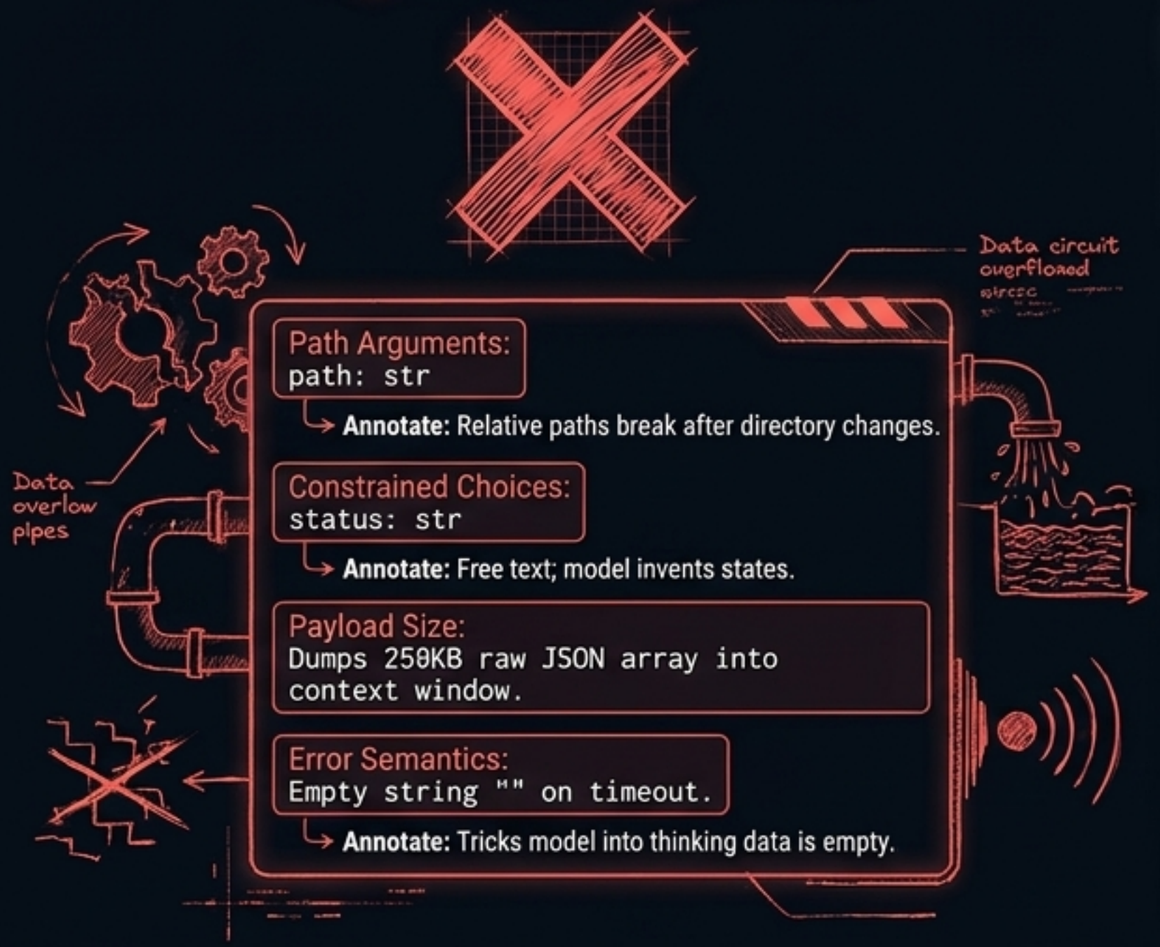
Token/Cost ceilings
(Financial kill switch)

Duplicate call traps
(Intercepts 3x repeating loops)

Structured Error Poka-Yoke
(is_error: true)

Mistake-Proof the Agent-Computer Interface (ACI) with Poka-Yoke Tooling

Fragile Prototype Tool

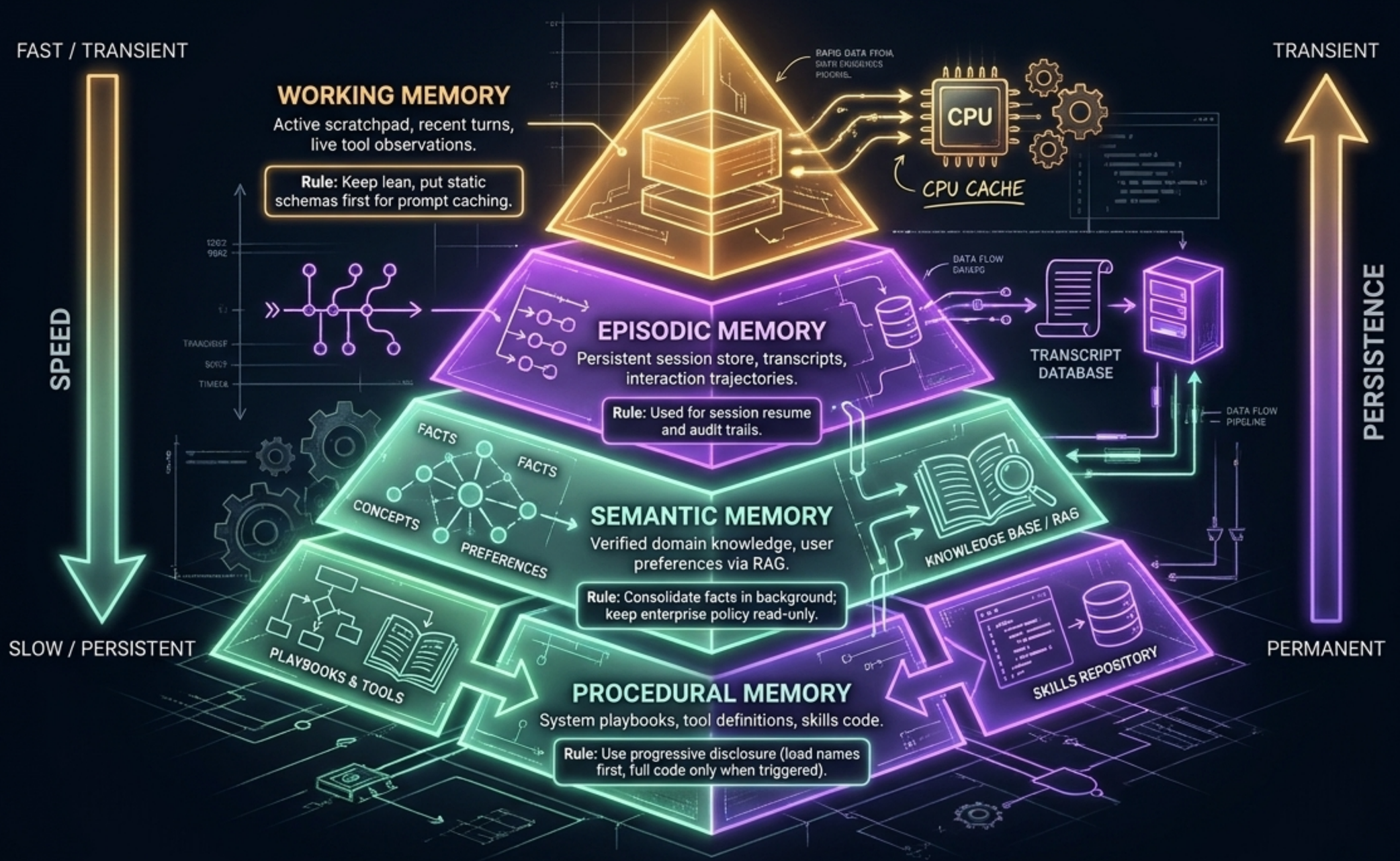


Production Poka-Yoke Tool

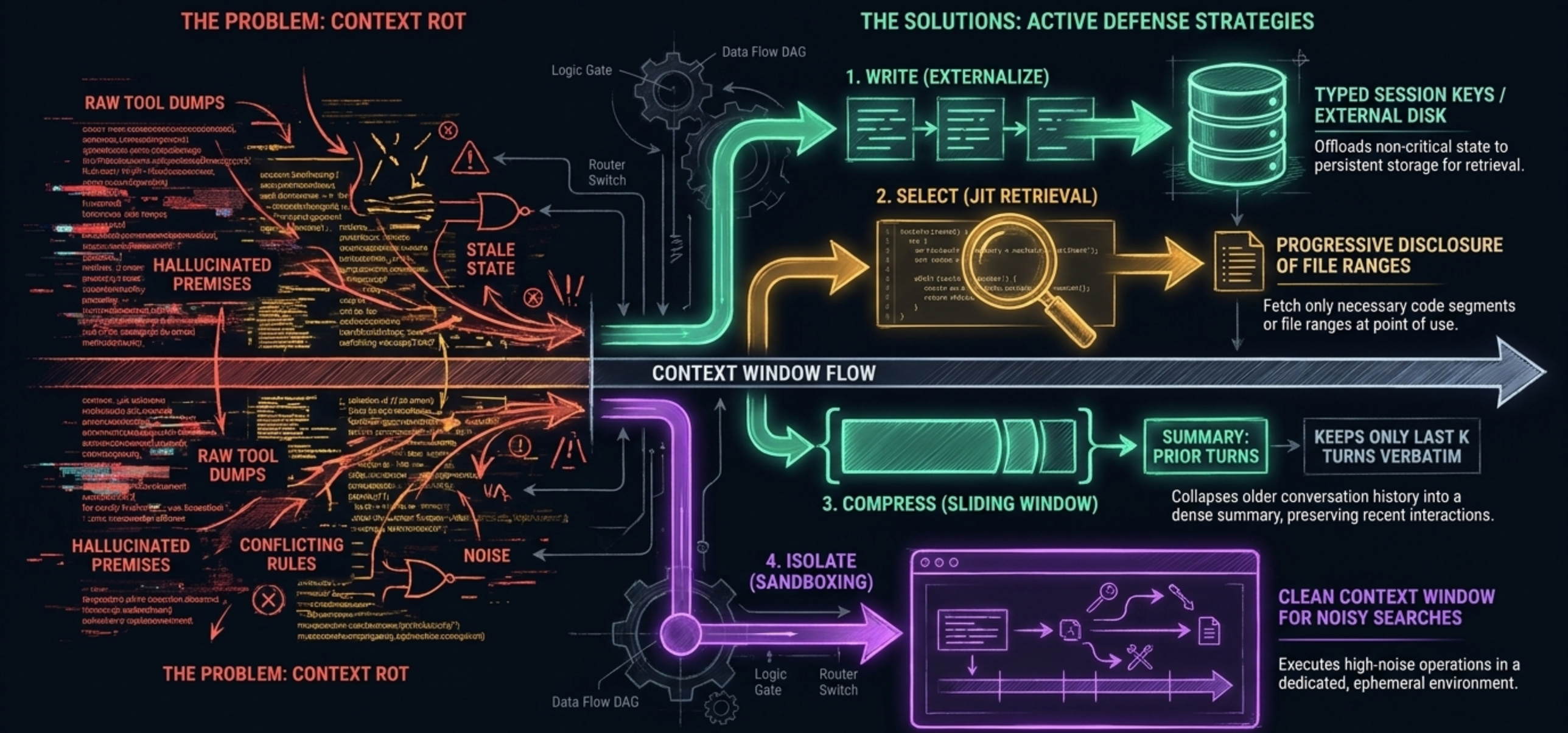


Core Principle: Shift left on error detection. Make invalid states unrepresentable in the ACI to prevent agent hallucinations and execution failures.

The 4-Tier Agent Memory Architecture Balances Cognitive Load and Persistence



Defend Against Context Rot with Active Compaction and Sandboxing Strategies



COLOR LEGEND: **MINT GREEN:** DETERMINISTIC CODE / SAFE PATHWAYS. **AMBER:** PROBABILISTIC REASONING / JIT. **CORAL RED:** CONTEXT ROT / ERRORS. **BRIGHT PURPLE:** AUTONOMOUS AGENTS / SANDBOXES.

Valid Multi-Agent Topologies Require Specialized Tools and Isolated Contexts



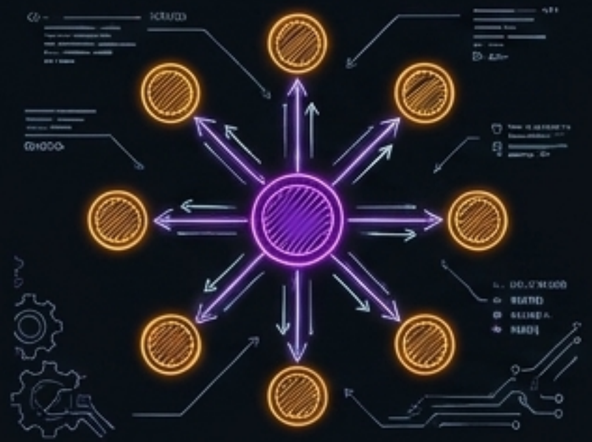
Agent Theater

The Litmus Test

Do not use multi-agent without distinct tools, parallel I/O, or adversarial verification.

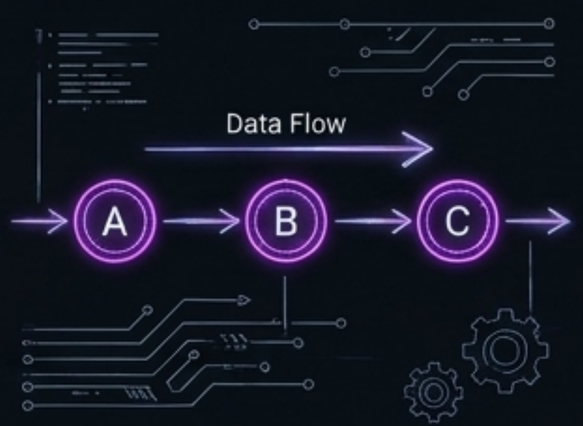
4 Canonical Topologies

1 Hub-and-Spoke



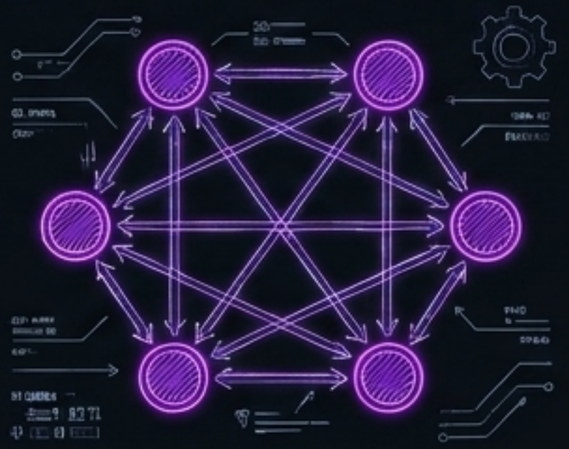
Hub-and-Spoke

2 Sequential Pipeline



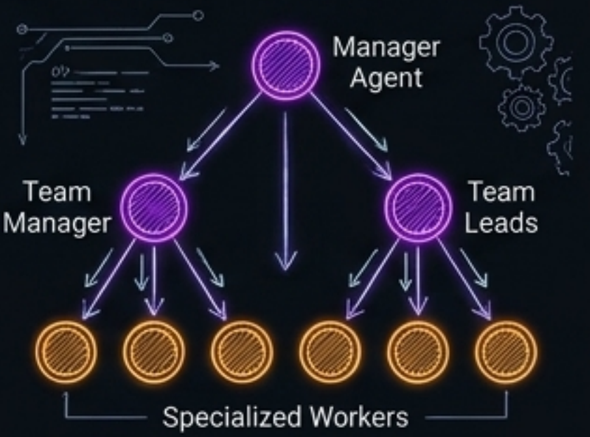
Sequential Pipeline

3 Mesh / Debate



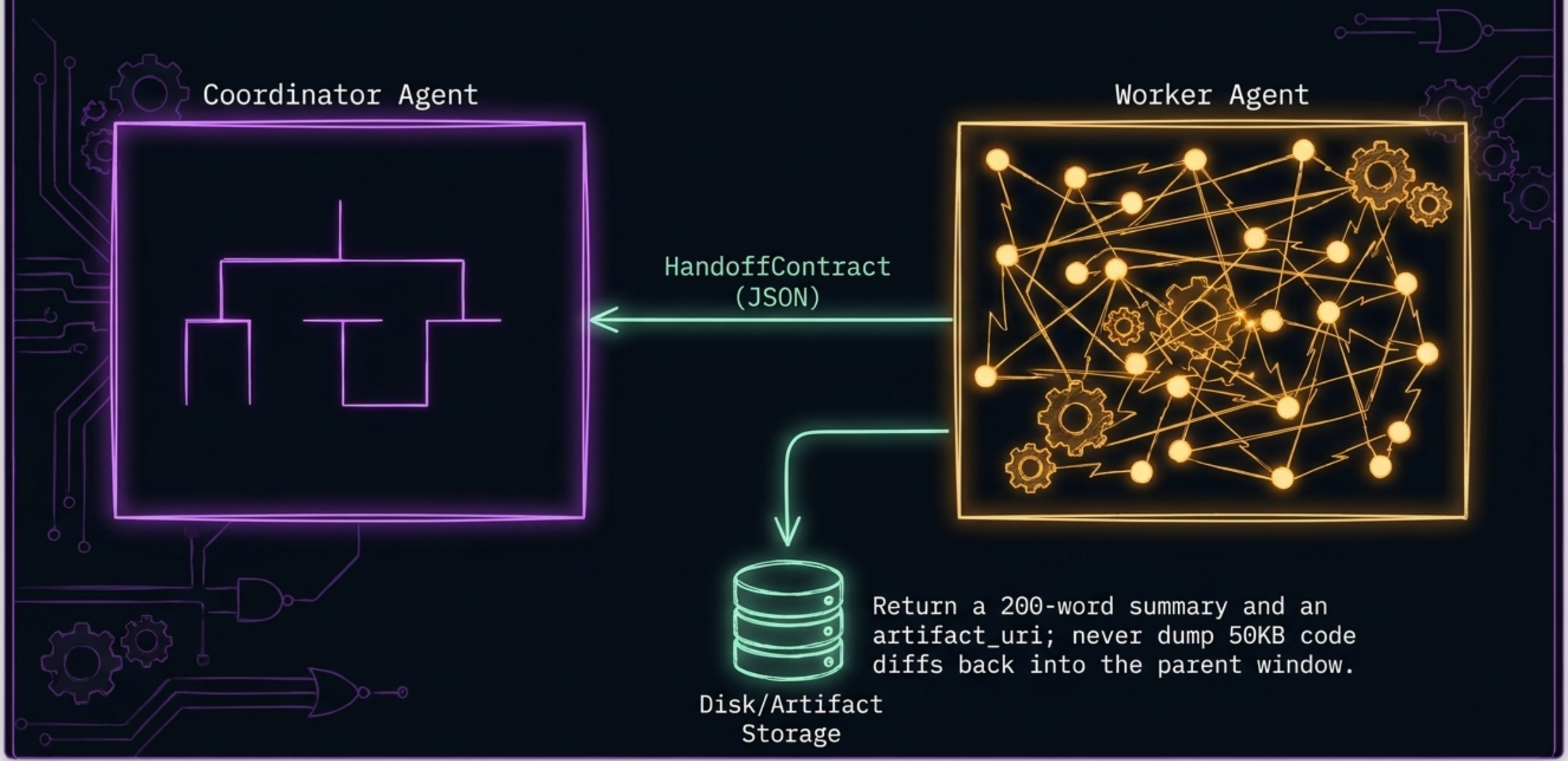
Mesh / Debate

4 Hierarchical Teams

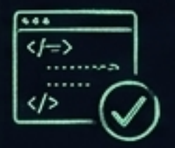


Hierarchical Teams

Handoff Contracts Prevent Context Contamination Across Agent Boundaries



Diagnosing Production Failures Through the Matrix of Doom

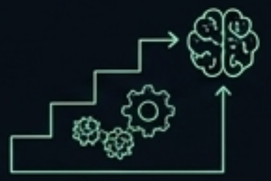
Production Symptom	Root Architectural Cause	Deterministic Fix
 Agent miscalculates fees on edge cases.	 Deterministic drift.	 Move threshold checks to Python PreToolUse.
 Parallel workers intermittently overwrite findings.	 Shared mutable state race.	 Assign unique output_keys per branch.
 Generator-Critic loop burns 15 turns and ships hallucination.	 Ungrounded reflection & unbounded loop.	 Cap iterations at 3 + regex verify citations against source.

The 8-Point Pre-Production Architectural Checklist Validates Enterprise Readiness

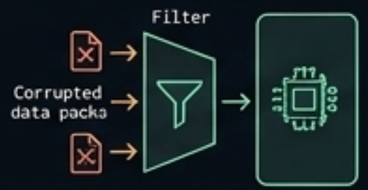
1. **3-Bucket Triage:** Math, RBAC, and policy executed in code; HITL for irreversible actions.



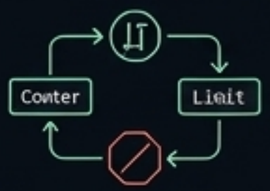
2. **Complexity Ladder:** Empirical baseline proves simple workflows fail before adding autonomy.



3. **Epistemic Gating:** $0 deterministic pre-LLM quality checks block corrupted inputs.



4. **Loop Hygiene:** Hard ceilings on turns, tokens, and duplicate calls active.



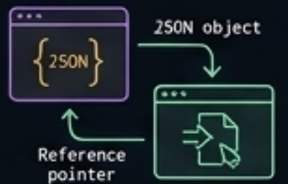
5. **ACI Poka-Yoke:** Tool catalogs <= 16, strict Literal schemas, absolute paths, and is_error feedback.



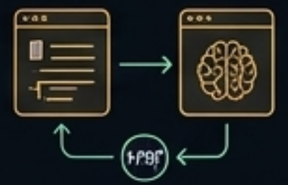
6. **Context Budgeting:** Static instructions prioritized for cache; sliding window compaction enabled.



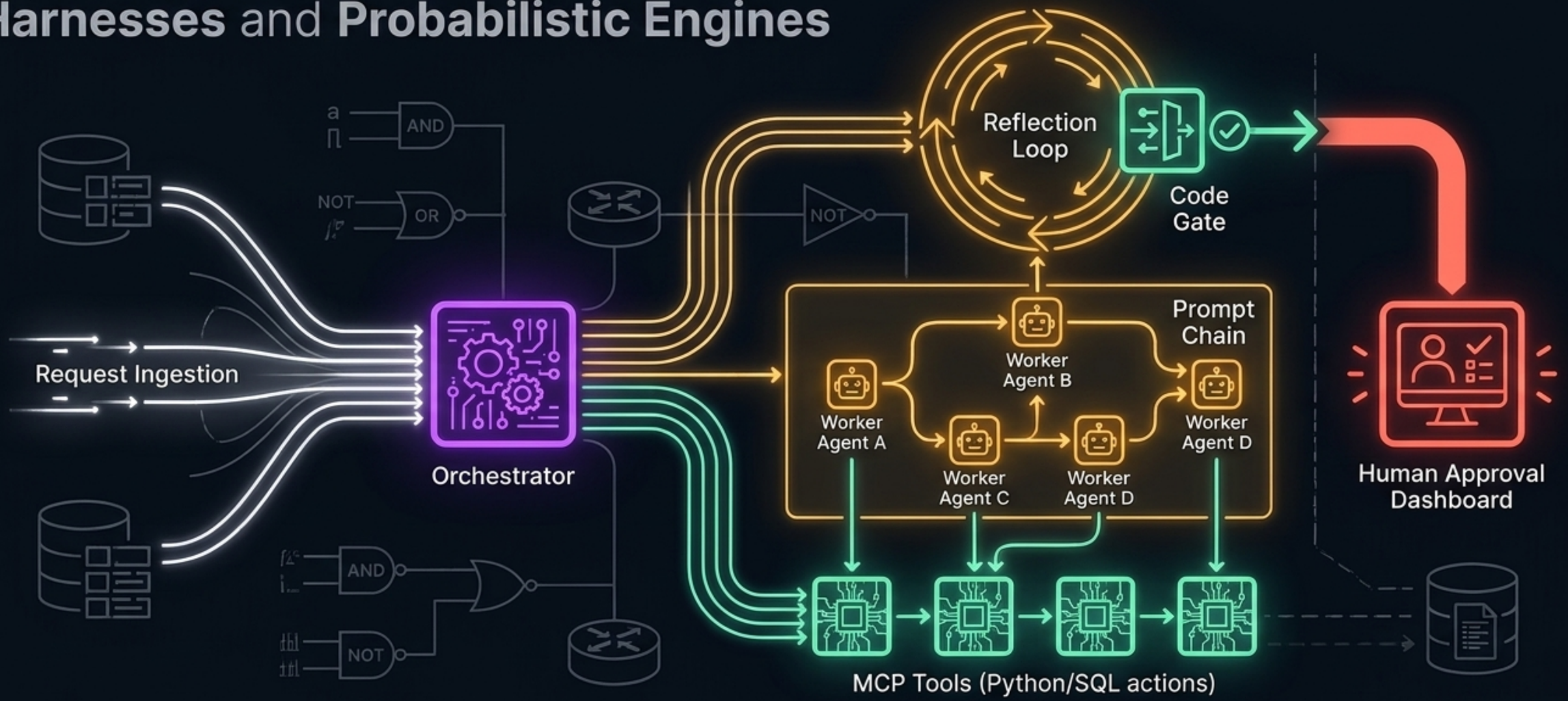
7. **Handoff Contracts:** Subagents run in isolated windows, returning typed JSON and artifact URIs.



8. **Separation of Authority:** High-stakes generation separated from independent cold-context evaluation.



The **Enterprise Blueprint** Synthesizes **Deterministic Harnesses** and **Probabilistic Engines**



True agentic engineering is not about maximizing model intelligence; it is about building the deterministic harness that binds it safely to reality.