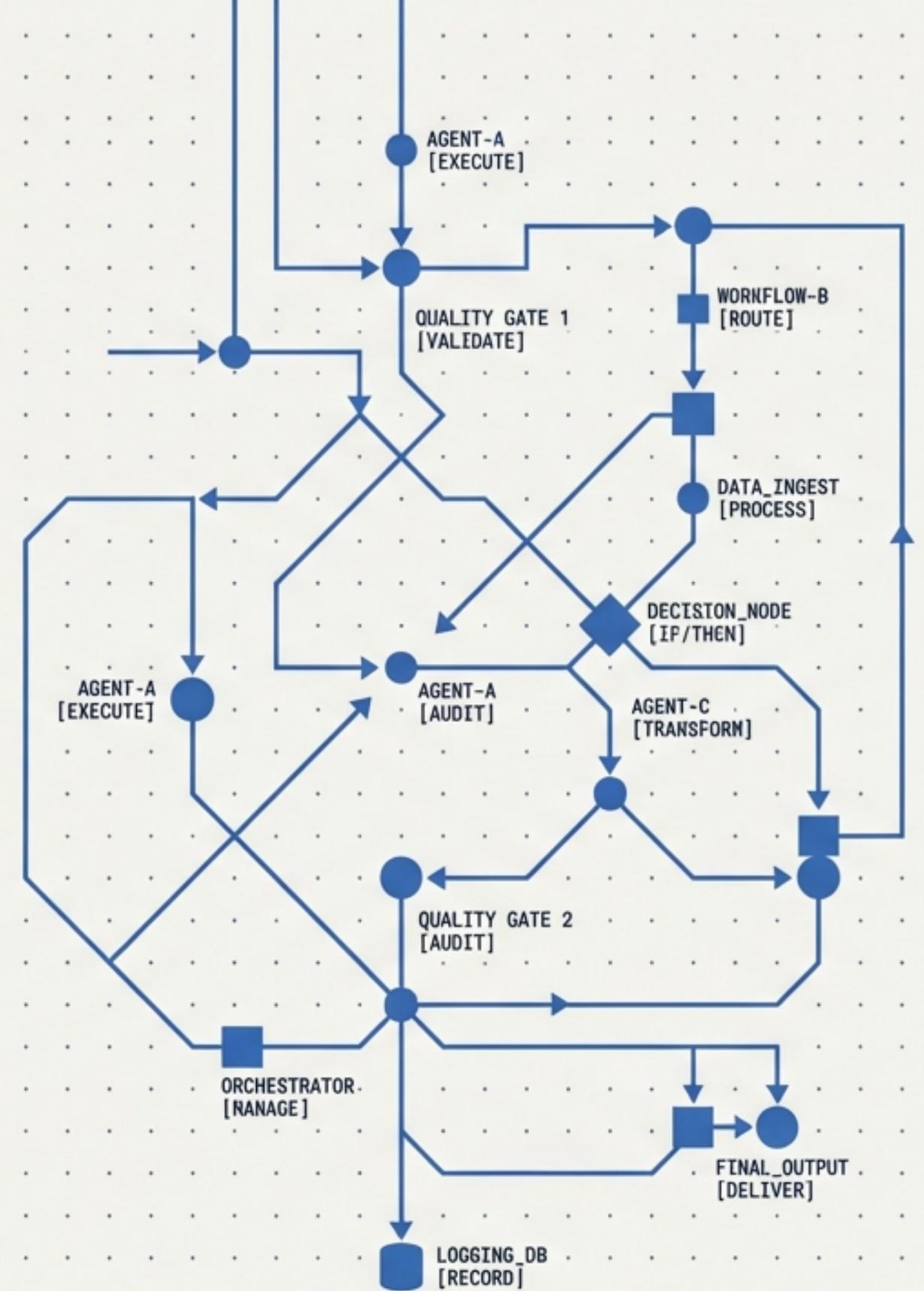
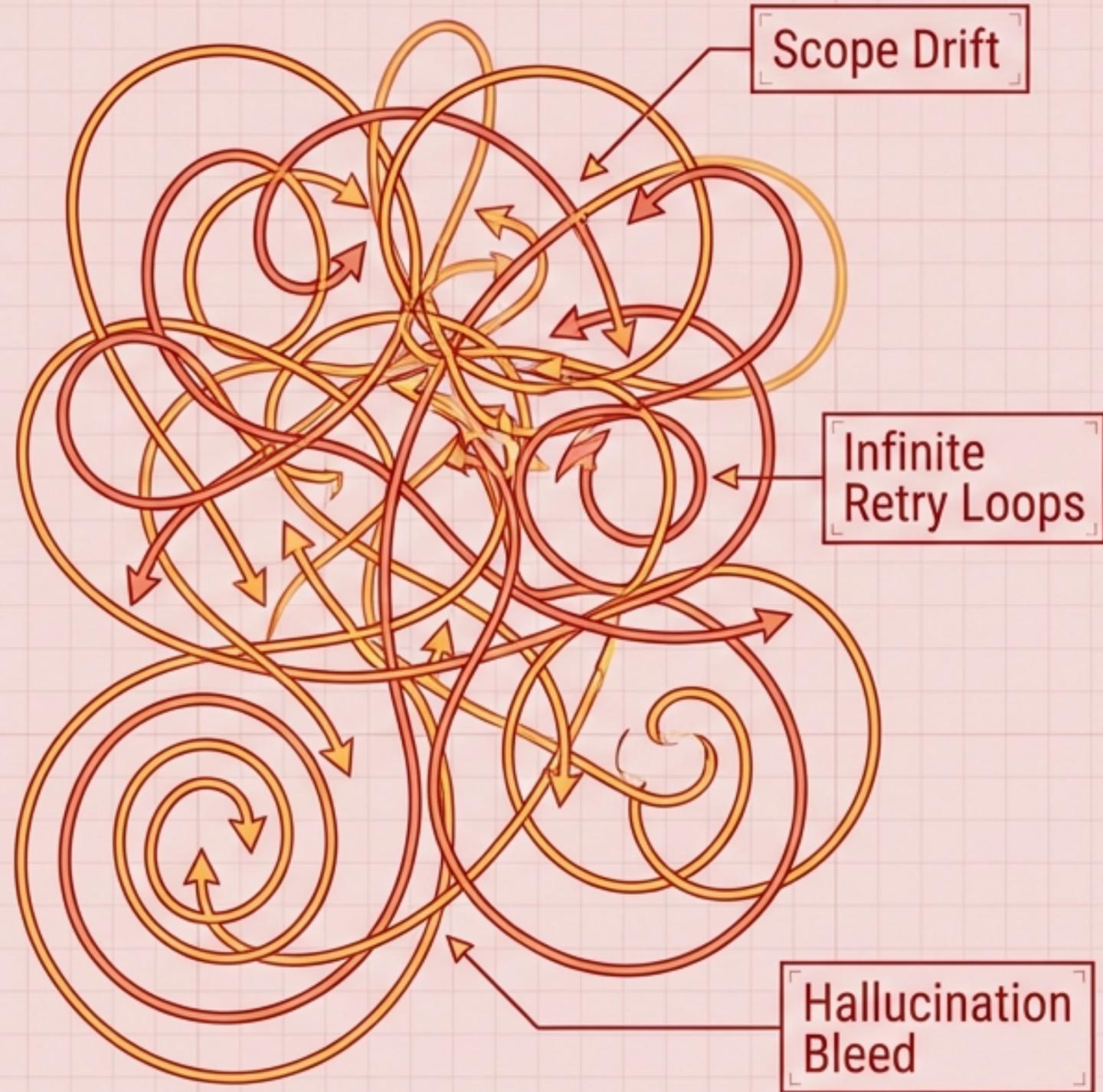


Mastering CC-302: Orchestration & Quality Gates

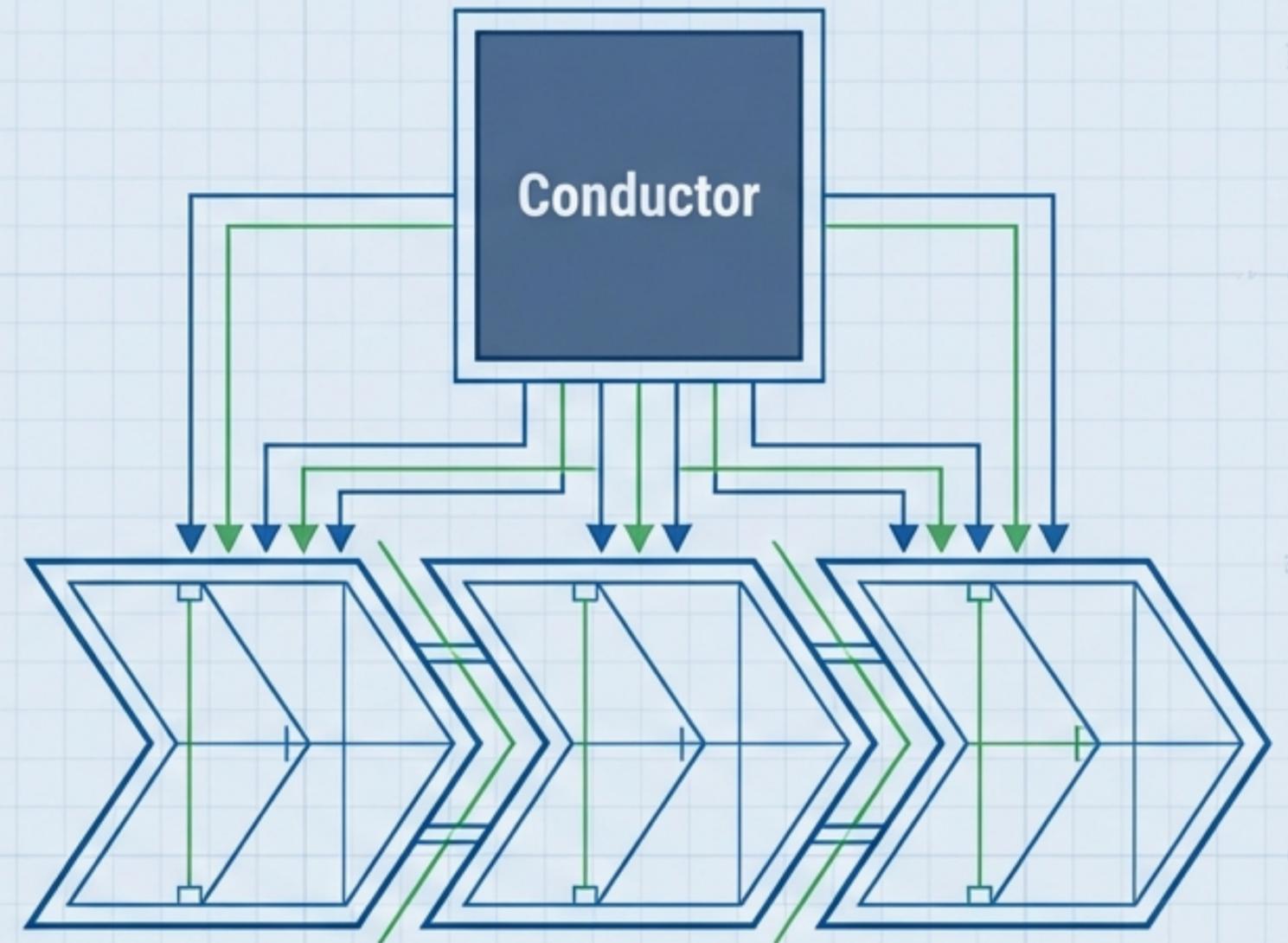
The definitive guide to building multi-agent workflows.



Chaos: Unmanaged Agents



Control: Orchestration



Moving beyond single-threaded assistants requires disciplined routing, strict artifact handoffs, and objective, automated quality control.

The 4 Pillars of CC-302 Architecture



Pillar 1: Tier Classification

Scoring task rigor via the 5-signal matrix.



Pillar 2: Phase Sequencing

The rigid Phase 0-6 automated pipeline.



Pillar 3: Quality Gates

The Critic agent's objective checkpoints.



Pillar 4: State Management

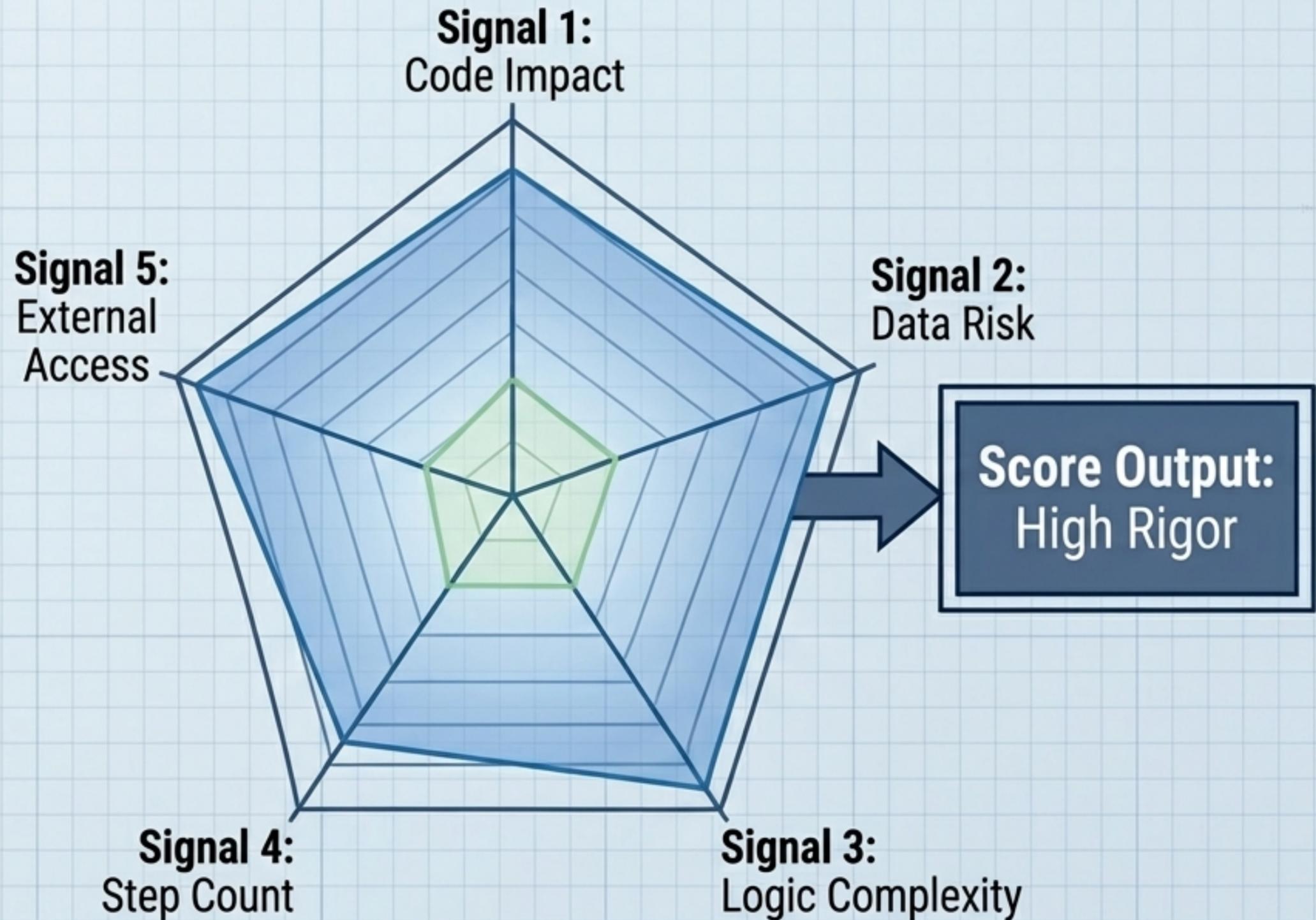
The conductor's singular source of truth.

Pillar 1: The 5-Signal Tier Matrix

Tasks are not treated equally. The orchestrator classifies incoming requests using a 5-signal weighted matrix to determine the **required** riigor, balancing token cost against operational risk.

Warning Amber

Rule: System overhead must scale proportionally to task complexity.



Mapping Tiers to Workflows

Level 5: CRITICAL

Enforces strict Opus-model checkpoints and human-in-the-loop escalation.

Level 4: MAJOR

Tightens phase checks, requires full event logging.

Level 3: STANDARD

Enforces full SDLC execution.

Level 2: MINOR

Executes a standard 3-phase sequence.

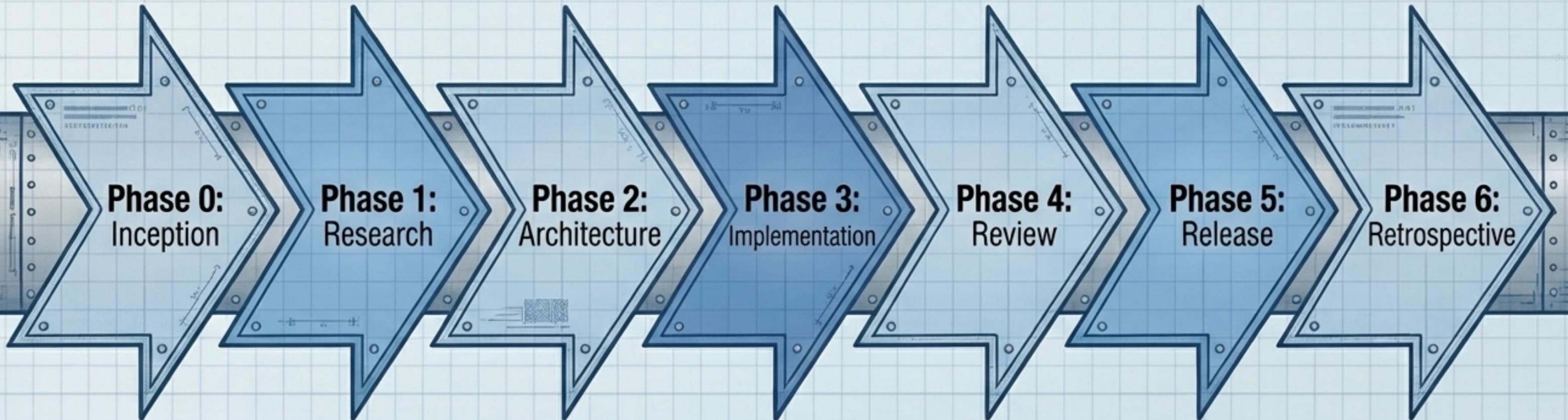
Level 1: TRIVIAL

Relaxes checks, skips most quality gates.

30%

Pillar 2: Phase Sequencing

The **Orchestrator's Pipeline** establishes a rigid, chronological sequence for all multi-agent work.



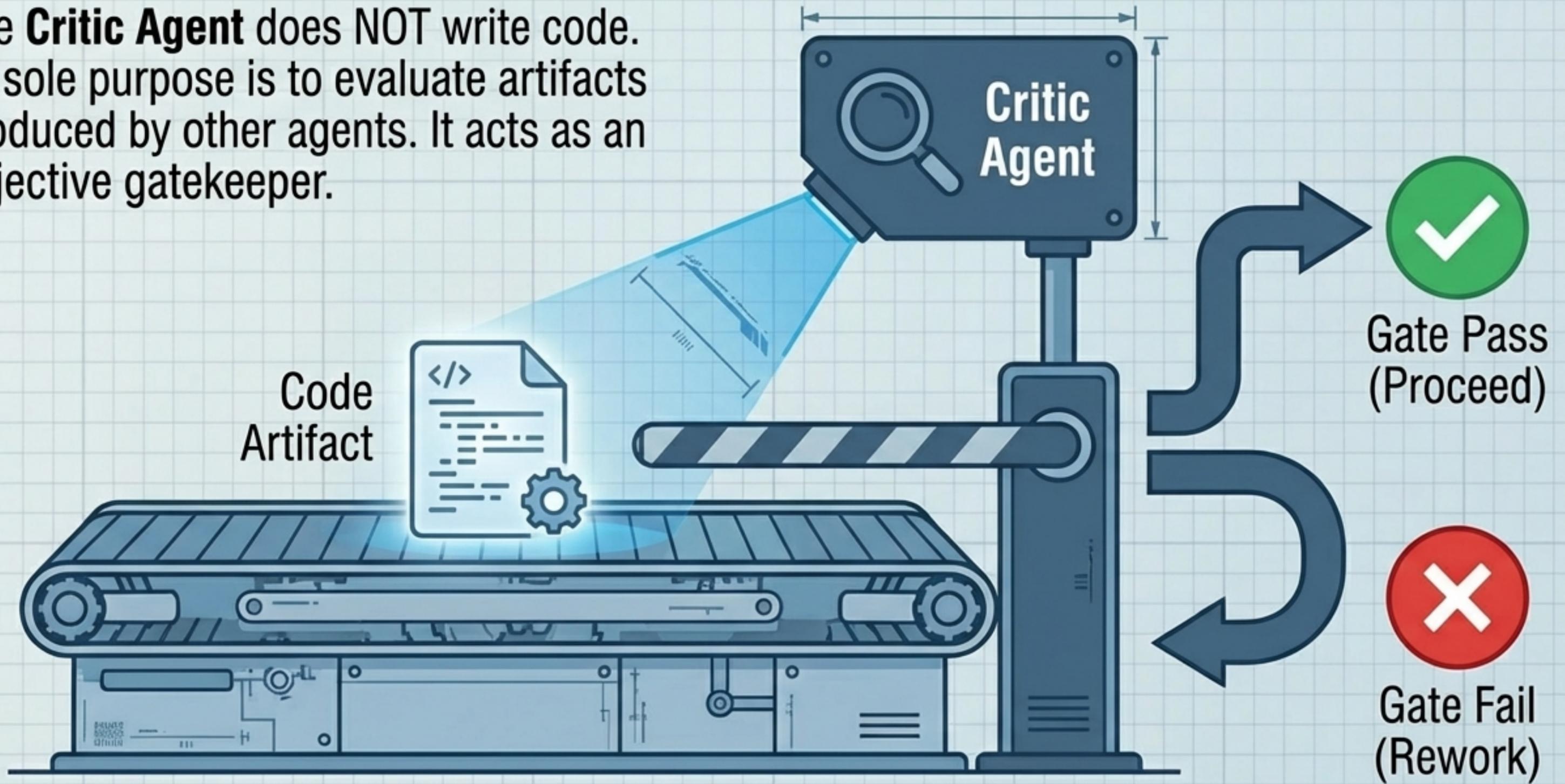
STRICT RULE: Phases **MUST** execute in order. Bypassing a phase requires explicit human escalation or predefined tier-relaxation rules.

Workflow Discipline: Keeping Agents on Track

Check Type	Failure Mode Prevented	Orchestrator Action
Sequence Check	Prevents skipping vital pipeline steps	Halts execution if phase prerequisites are missing.
Drift Check	Prevents output from drifting off-prompt	Re-anchors agent to original BRD constraints.
Scope Check	Prevents the agent from doing too much	Terminates early if agent exceeds authorization.
Loop Check	Prevents getting stuck in a retry loop	Triggers bounded escalation after max attempts.

Pillar 3: Quality Gates & The Critic Agent

The **Critic Agent** does NOT write code. Its sole purpose is to evaluate artifacts produced by other agents. It acts as an objective gatekeeper.

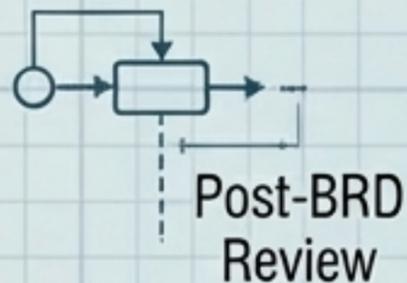


Gate Modes: Advisory vs. Blocking



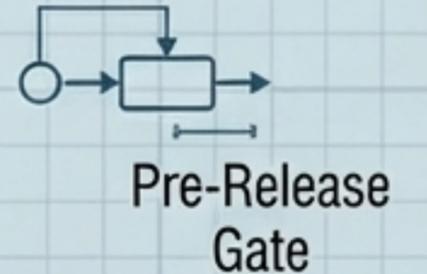
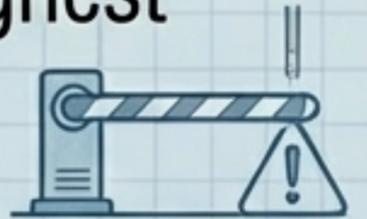
Advisory Mode

- **Model Selection:** Sonnet (Medium Cost)
- **Consequence:** Logs warnings, finds gaps, but allows progression.
- **Placement Example:** Post-BRD Review.



Blocking Mode

- **Model Selection:** Opus (Highest Cost / Max Reasoning)
- **Consequence:** Hard stop. Demands immediate rework or escalation.
- **Placement Example:** Pre-Release Gate.



False positives cost time; false negatives cost production stability.

The 7 Checkpoint Protocols

The **Critic Agent** validates completeness and flags false positives or negatives using a strict 7-point evaluation protocol at the gate.

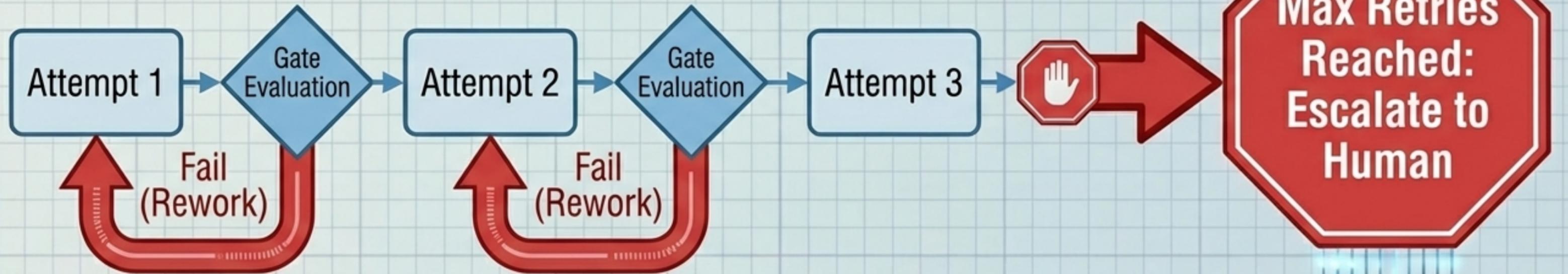
Prompt engineering for the Critic requires rigid **MUST NOT** constraints to prevent rubber-stamping flawed code. If the Critic is too lenient, the gate fails its purpose.

HUD CHECKLIST

- 1. Requirement Traceability
- 2. Schema Validation
- 3. Constraint Adherence
- 4. Scope Containment 
- 5. Contextual Accuracy
- 6. Edge Case Handling
- 7. Format Compliance

Retry Policies & Escalation Paths

Autonomous agents must not be allowed to loop indefinitely. Orchestrators enforce bounded retries on all Gate Fails.



After the maximum retry threshold is reached, the system pauses the state and alerts a human operator to resolve the impasse.

Pillar 4: State Management

The conductor's single source of truth. The state schema tracks the active phase, current tier, and artifact inventory.



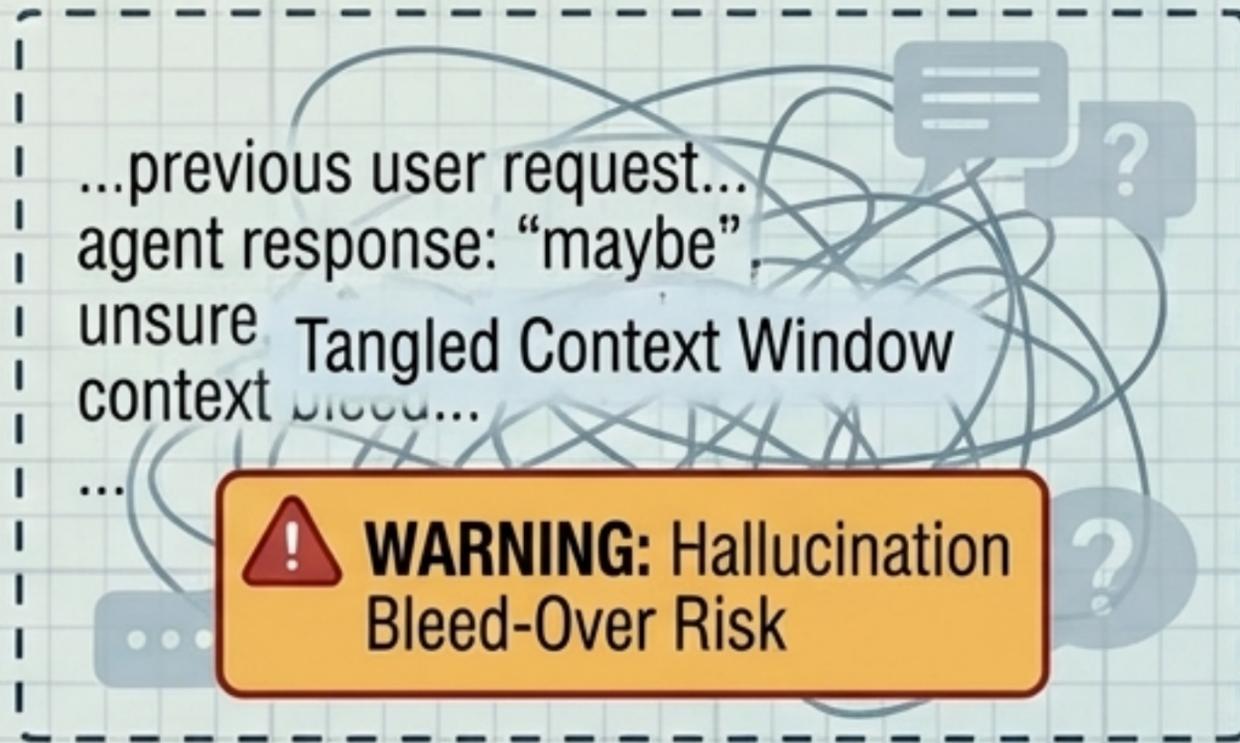
WARNING: Multiple agents CANNOT write to state concurrently without corruption. The state schema enforces strict Read/Write/Validate locks.

```
{  
  "current_phase": "Phase 3: Implementation",  
  "tier": "MINOR",  
  "artifacts_produced": [  
    "brd.md", "architecture.md"],  
  "status": "Awaiting Critic Review"  
}
```

Fresh Context Isolation

Rule: Pass artifacts, not context.

Bad Pattern



CC-302 Standard



The Conductor wipes the context window clean between phases. Downstream agents only receive finalized artifacts from upstream, never the messy conversation history that produced them.

CC-302 Assessment: Common Pitfalls

ANTI-PATTERN



Overcomplicating tier classification.

STANDARD FIX



Fix: Start simple. Base initial tiers on just 3 distinct signals.

ANTI-PATTERN



Critic agent rubber-stamping.

STANDARD FIX



Fix: Critic prompts demand strict MUST NOT rules to force objective critique.

ANTI-PATTERN



State file corruption.

STANDARD FIX



Fix: Implement strict sequential locks to manage concurrent writes.

Lab Progression Checklist

1. Classify 5 abstract signals manually.

2. Build the programmatic Tier Classifier.

3. Create the MINOR workflow (3-phase sequence).

4. Implement a Critic Gate with a blocking rule.

5. Wire the conductor-state.json file to manage transitions.

You are now ready to build the Control Tower.