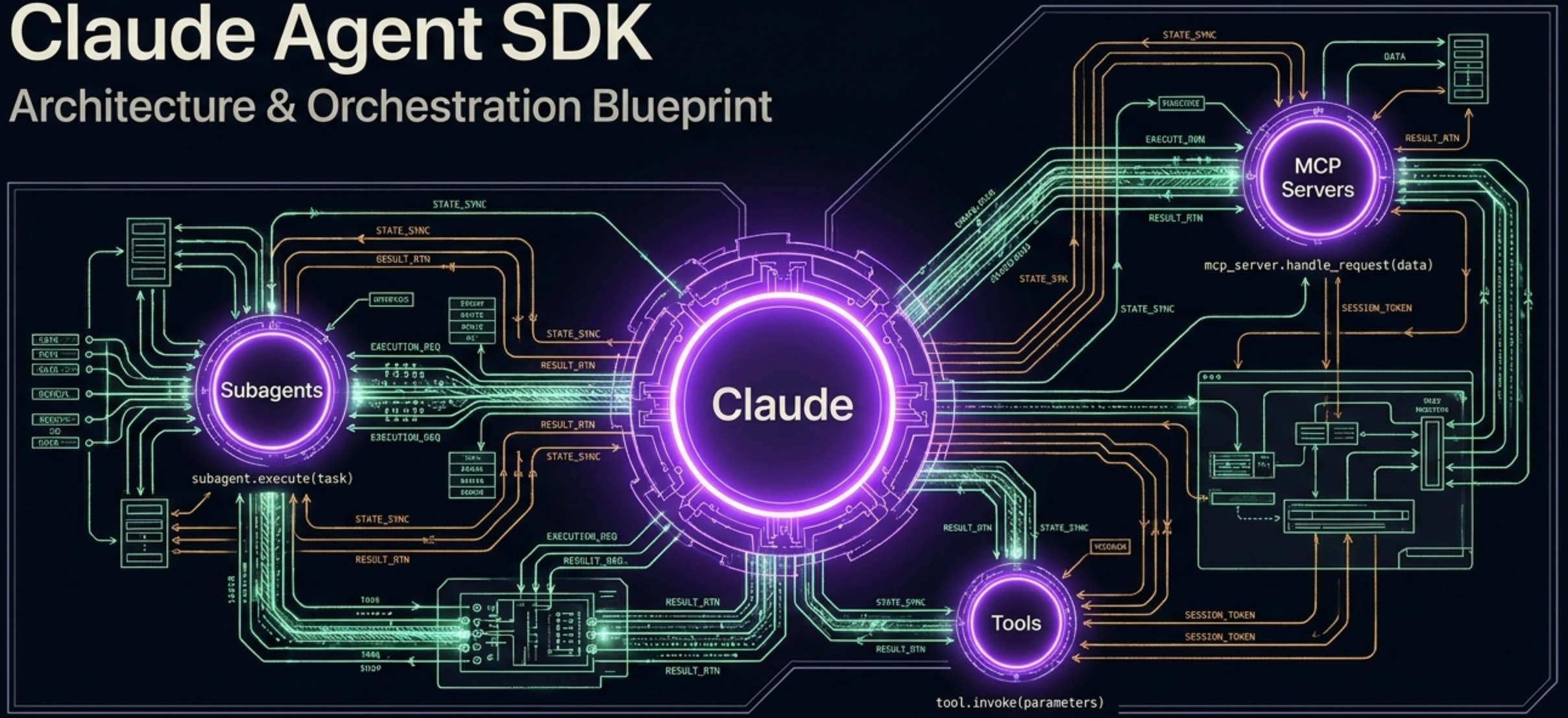
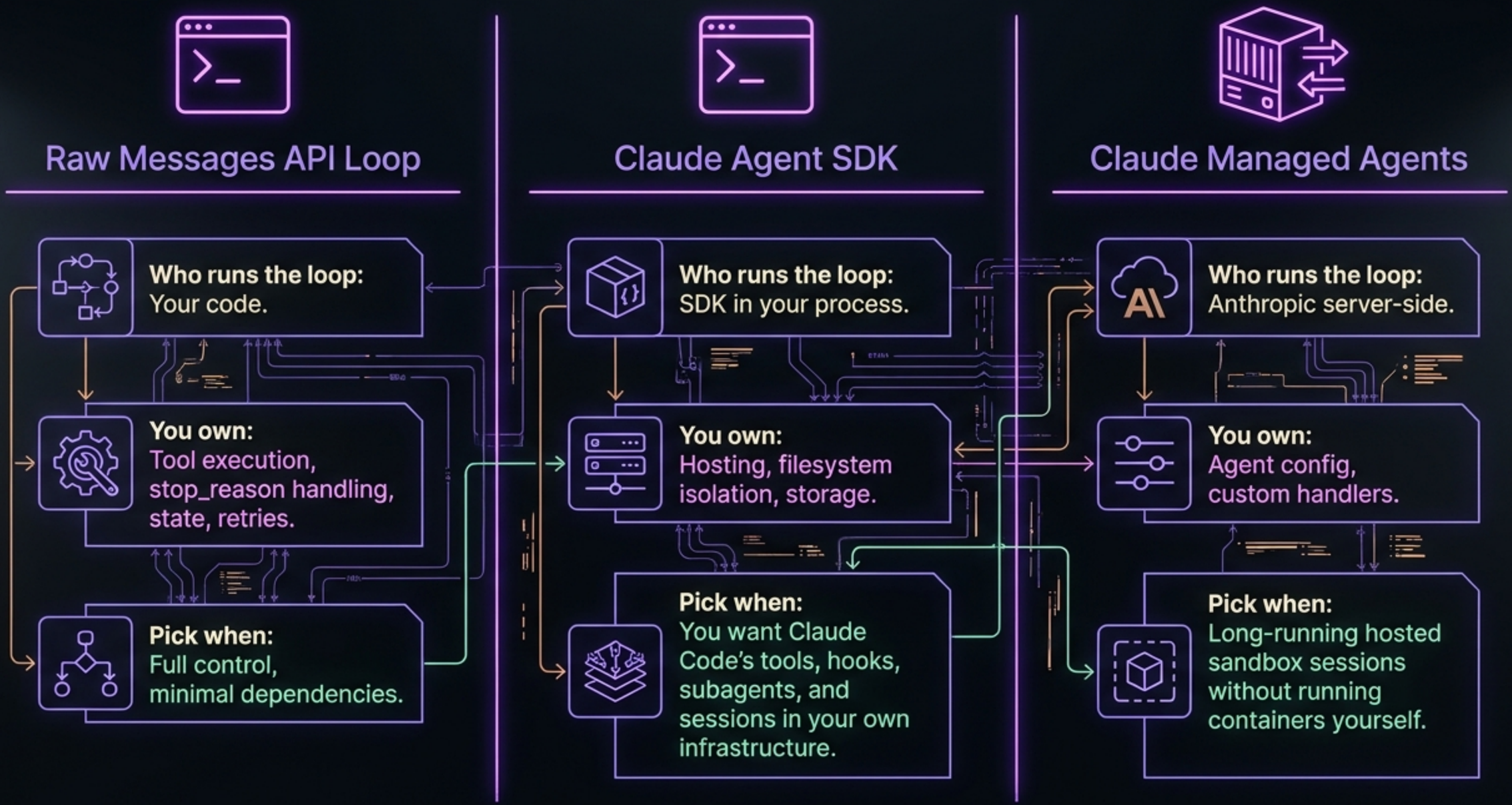
Claude Agent SDK

Architecture & Orchestration Blueprint



Mastering the execution loop, subagent isolation, deterministic hooks, and managed infrastructure.

Three Ways to Wire an Agent



The Model Decides



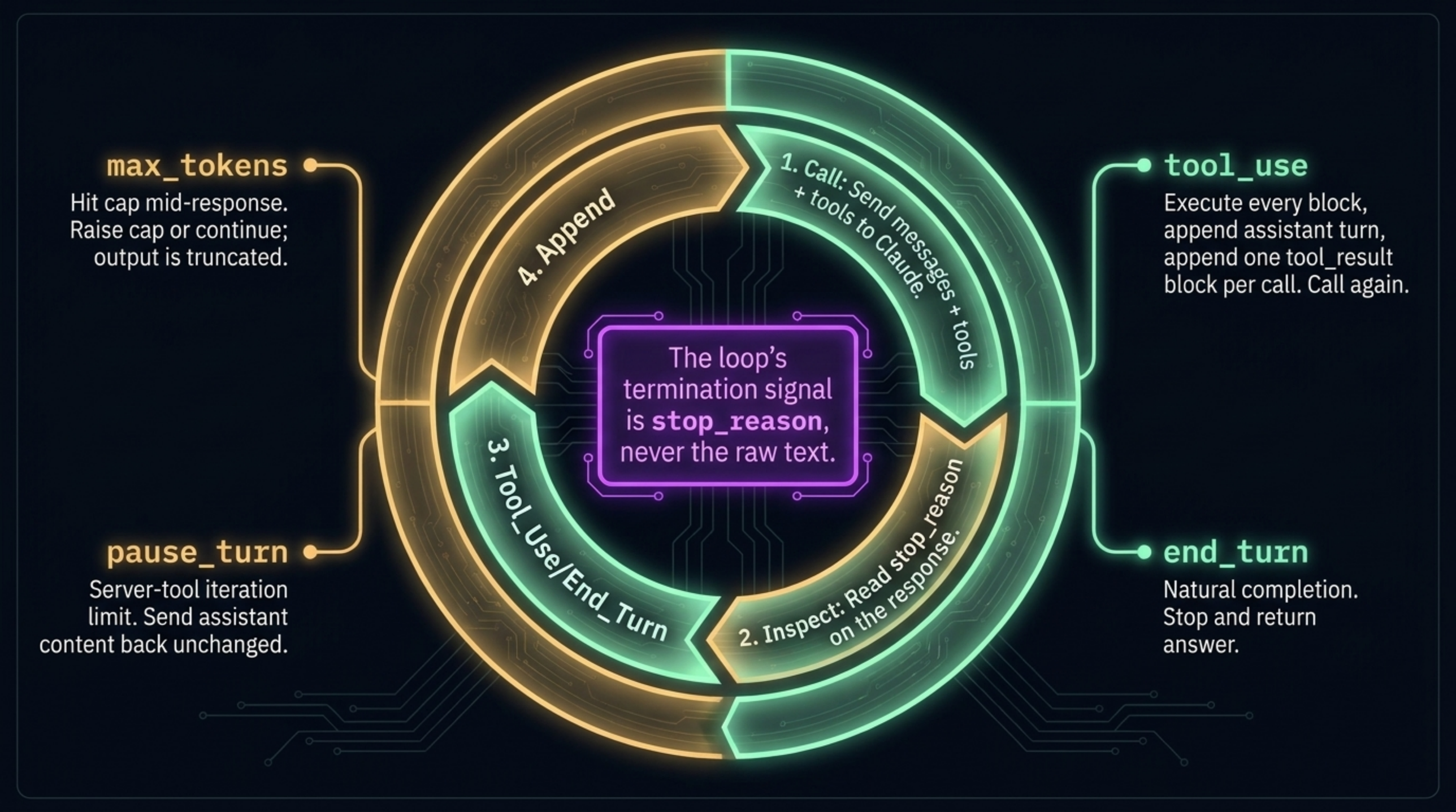
The model is responsible for probabilistic reasoning—choosing which tool to call, which subagent to invoke, and when the task is complete.

The Harness Enforces

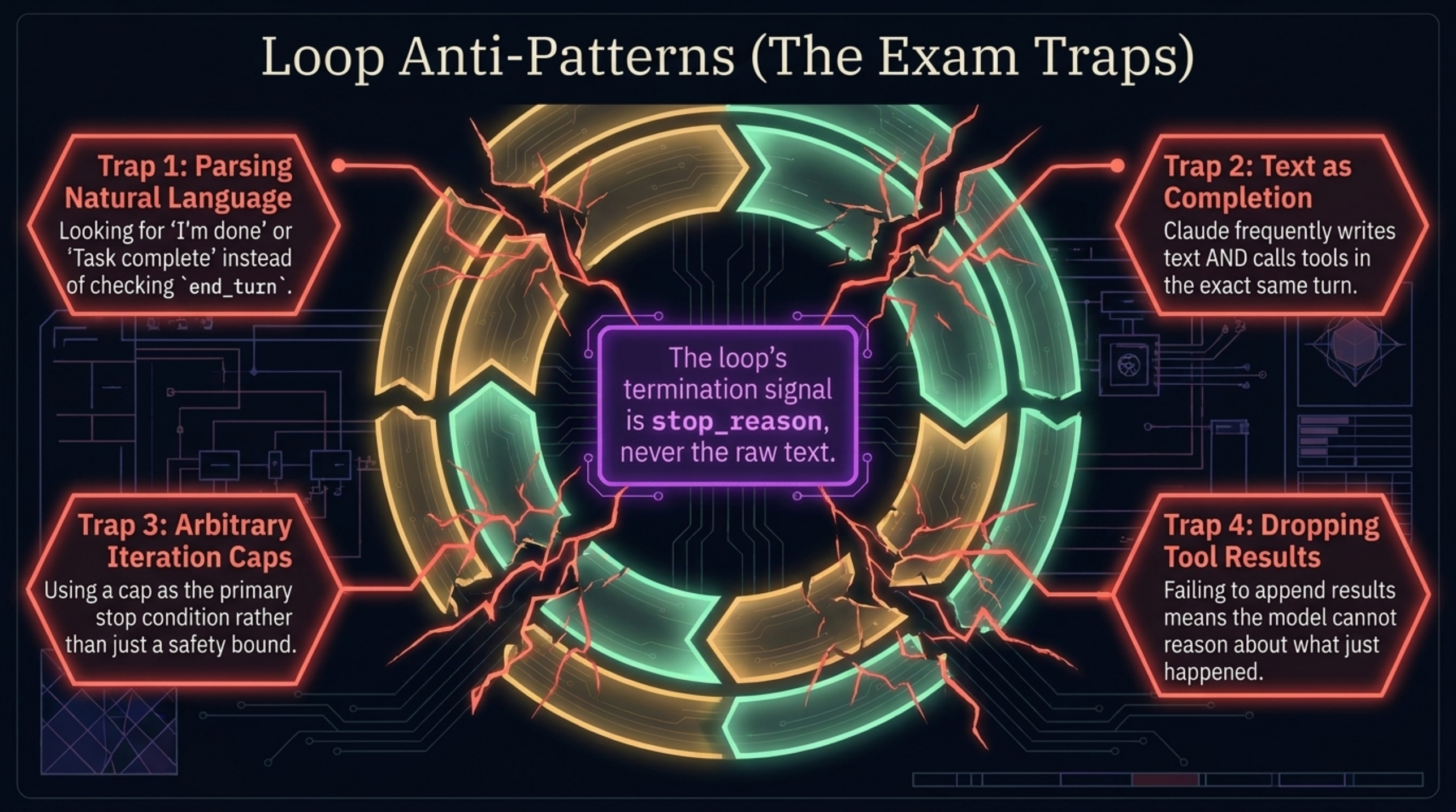


The SDK harness controls rigid boundaries—permissions, deterministic hooks, and financial budgets.

Anything that must hold 100% of the time belongs in code, never only in the prompt.


max_tokens
Hit cap mid-response. Raise cap or continue; output is truncated.
pause_turn
Server-tool iteration limit. Send assistant content back unchanged.
tool_use
Execute every block, append assistant turn, append one tool_result block per call. Call again.
end_turn
Natural completion. Stop and return answer.
1. Call: Send messages + tools to Claude.
2. Inspect: Read stop_reason on the response.
3. Tool_Use/End_Turn
4. Append
The loop's termination signal is stop_reason, never the raw text.

Loop Anti-Patterns (The Exam Traps)



SDK Core API Mechanisms

query(prompt, options)

Mechanics: Starts a fresh session per call. Single-shot.

Behavior: Raises an exception immediately after yielding an error result (e.g., error_max_turns).

Fix: Wrap in try/catch to resume with continue_conversation=True.

ClaudeSDKClient

Mechanics: Reuses one session across multiple await client.query() turns. Asynchronous server session.

Behavior: Supports await client.interrupt() to cleanly drain buffered messages up to the interrupted turn's ResultMessage before sending the next query.

Context Settings & Multi-Tenant Traps

`setting_sources=[]`

Trap: Disables filesystem loading, BUT still loads managed policies, global ~/.claude.json, auto-memory, and claude.ai MCP connectors.

Fix:
```
CLAUDE_CODE_DISABLE_AUTO_MEMORY='1'
strict_mcp_config=True
```

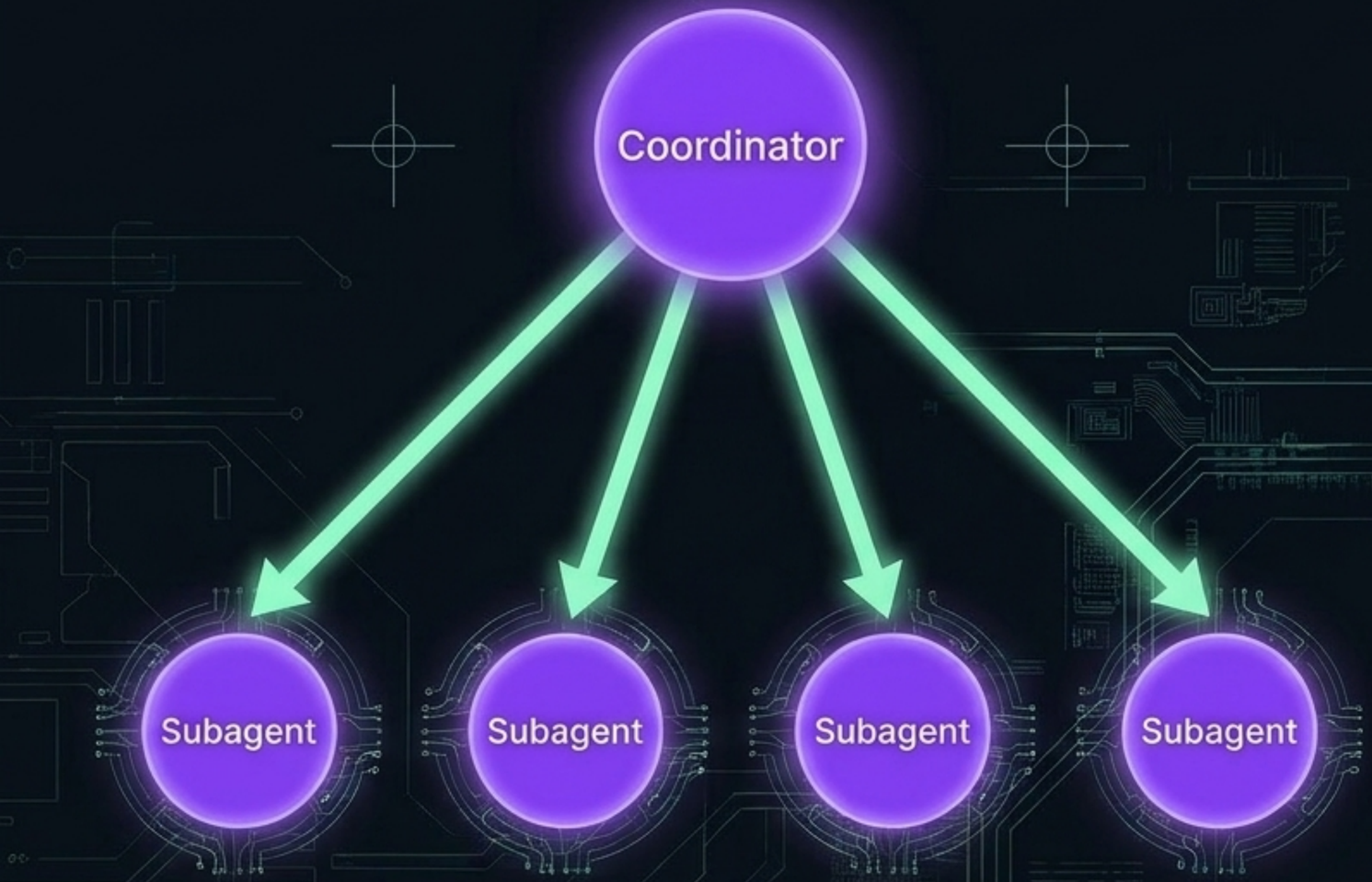
`allowed_tools` vs `disallowed_tools`

Trap: `allowed_tools` only pre-approves. `disallowed_tools` removes from context entirely or denies execution.

`env` handling

Trap: In Python, env merges onto the process environment. In TypeScript, it REPLACES `process.env`. If you don't spread `...process.env`, PATH is lost and CLI tools fail with ENOENT.

Hub-and-Spoke Orchestration



Core Principle: Subagents never communicate directly. All routing goes through the coordinator for observability and control.

Coordinator Responsibilities:

1. **Select:** Pick specific subagents based on query needs.
2. **Partition:** Give each subagent a non-overlapping scope.
3. **Decompose:** Ensure the whole question is covered.
4. **Instruct:** State clear goals and quality criteria.
5. **Refine:** Evaluate synthesis and re-delegate gaps.

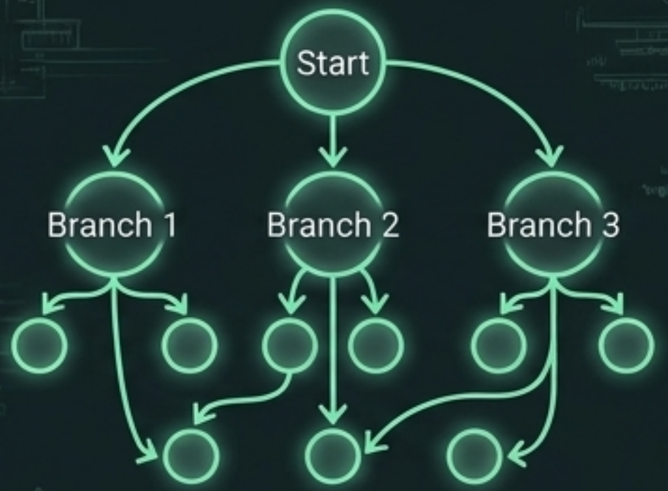
Task Decomposition Patterns

Prompt Chaining (Sequential)



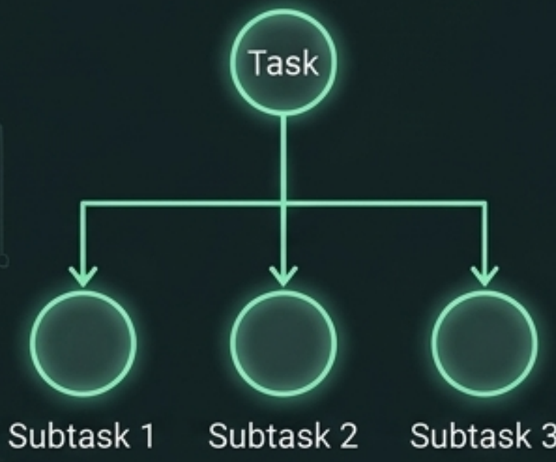
Use when: Steps and order are known in advance (e.g., File review -> Integration pass).

Dynamic Branching (Adaptive)



Use when: Open-ended investigation where next steps depend on previous tool results (e.g., Legacy codebase mapping).

Parallel Fan-out (Simultaneous)

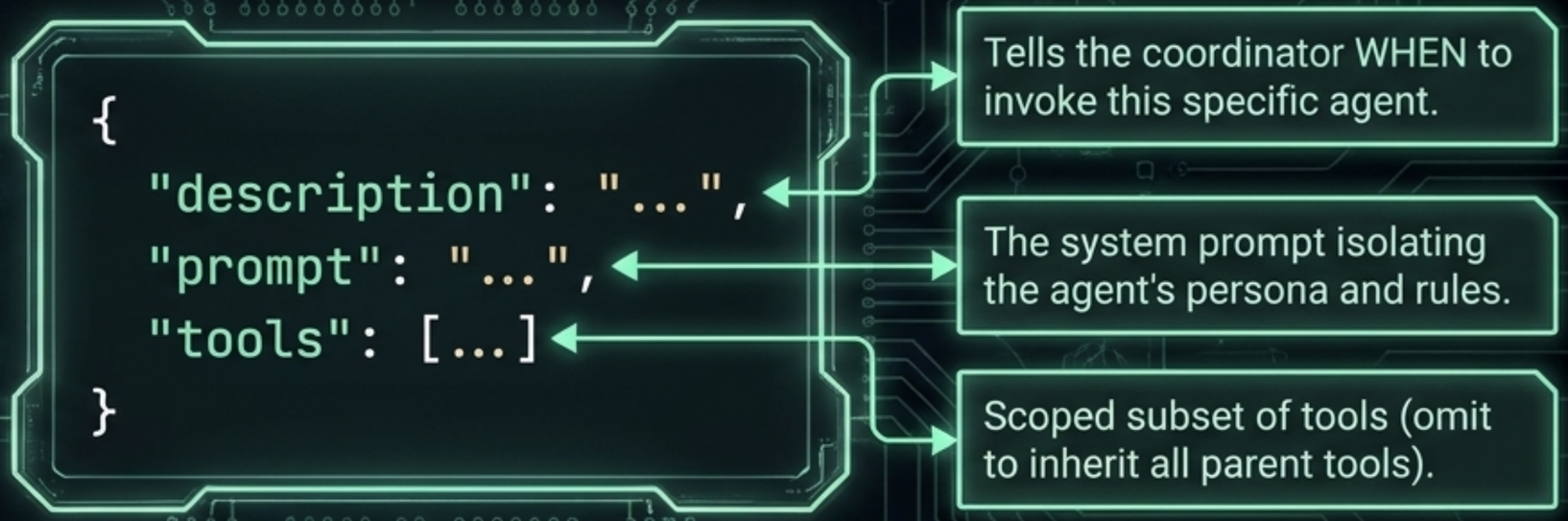


Use when: Subtasks are entirely independent (e.g., Search, document analysis, data lookup).

Attention Dilution Trap: Do not use one giant prompt to review 30 files at once. Split into per-file local passes + one integration pass.

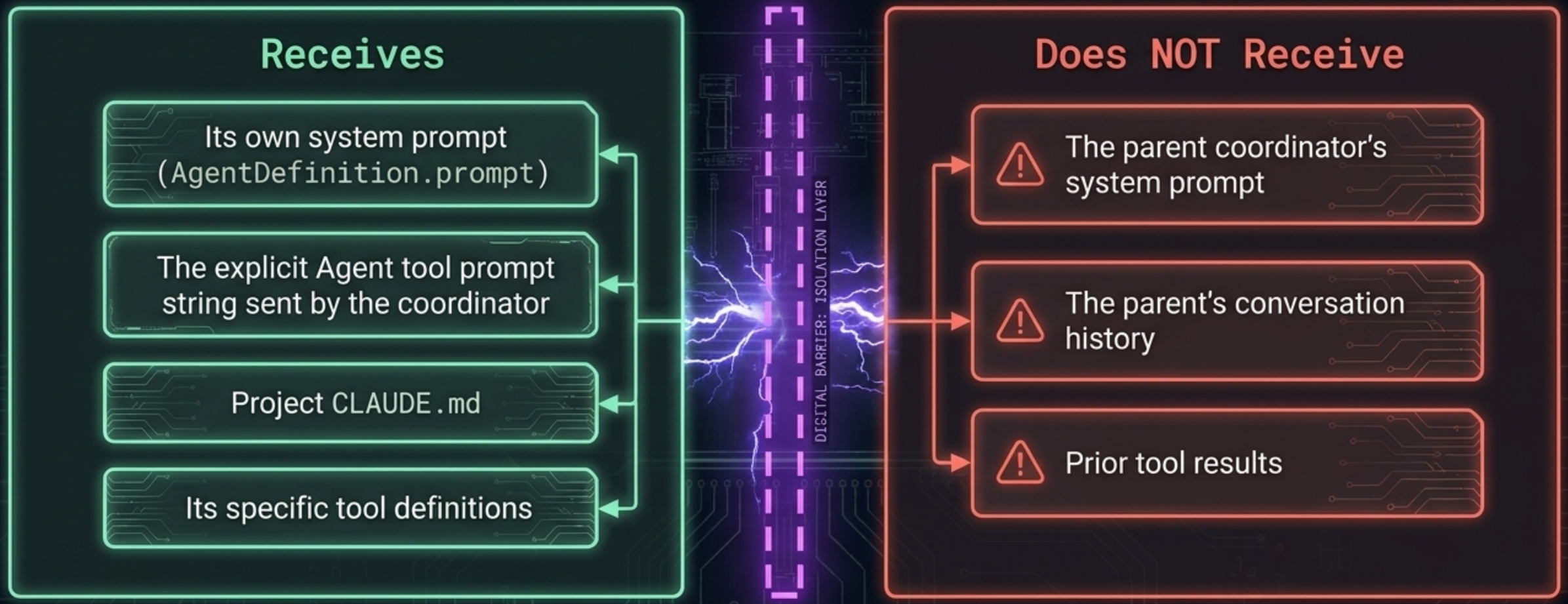
Subagent Invocation & The Agent tool

Spawning Mechanism: The coordinator uses the Agent tool (legacy: Task) to spawn subagents. The tool must be in context and approved.



⚠ **Concurrency Note:** Fan-out depth and concurrency are strictly capped by `CLAUDE_CODE_MAX_SUBAGENT_SPAWN_DEPTH` and `CLAUDE_CODE_MAX_CONCURRENT_SUBAGENTS`. Emit multiple Agent tool calls in one response to run concurrently.

Context Passing & Isolation



Golden Rule: The only channel from coordinator to subagent is the prompt string. Complete prior findings, file paths, and decisions must be injected there.

Deterministic Enforcement via Hooks

The Problem: Prompts are probabilistic. Strict rules (e.g., 'never write to .env', 'verify ID before refund') require code-level guarantees.

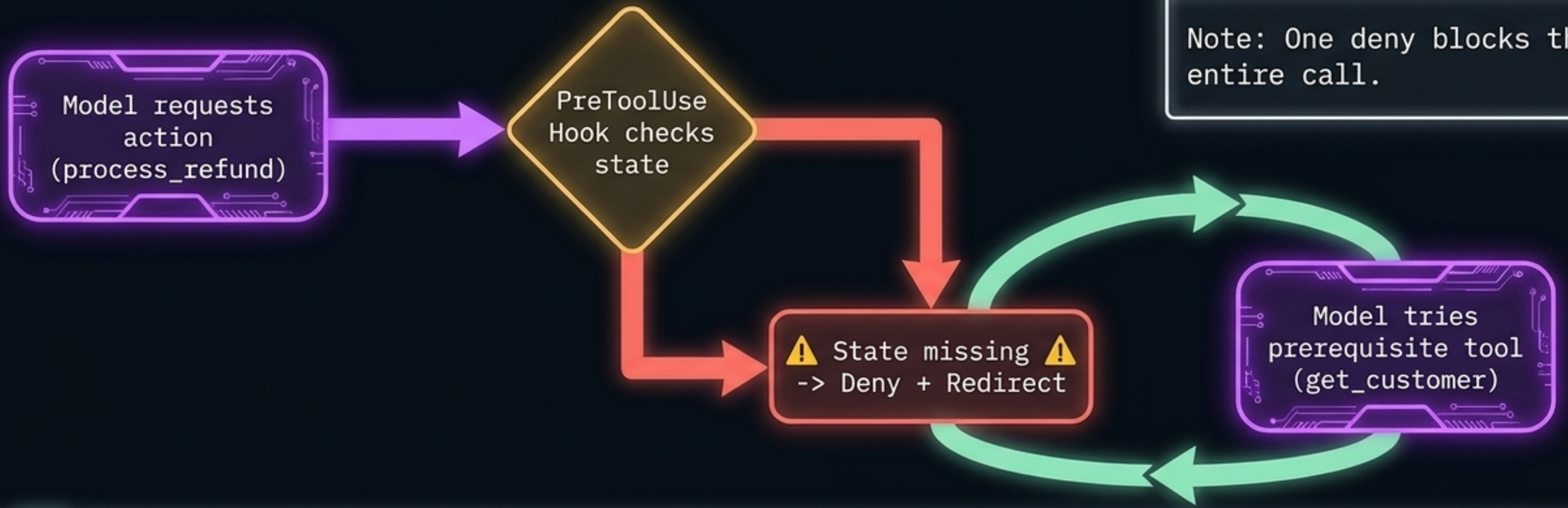


Deterministic enforcement before execution. Capable of evaluating state, denying permissions, or overriding inputs.

Normalization and redaction after execution. Used to format heterogeneous data (Unix epoch vs ISO 8601) before the model sees it.

The Prerequisite Gate Pattern

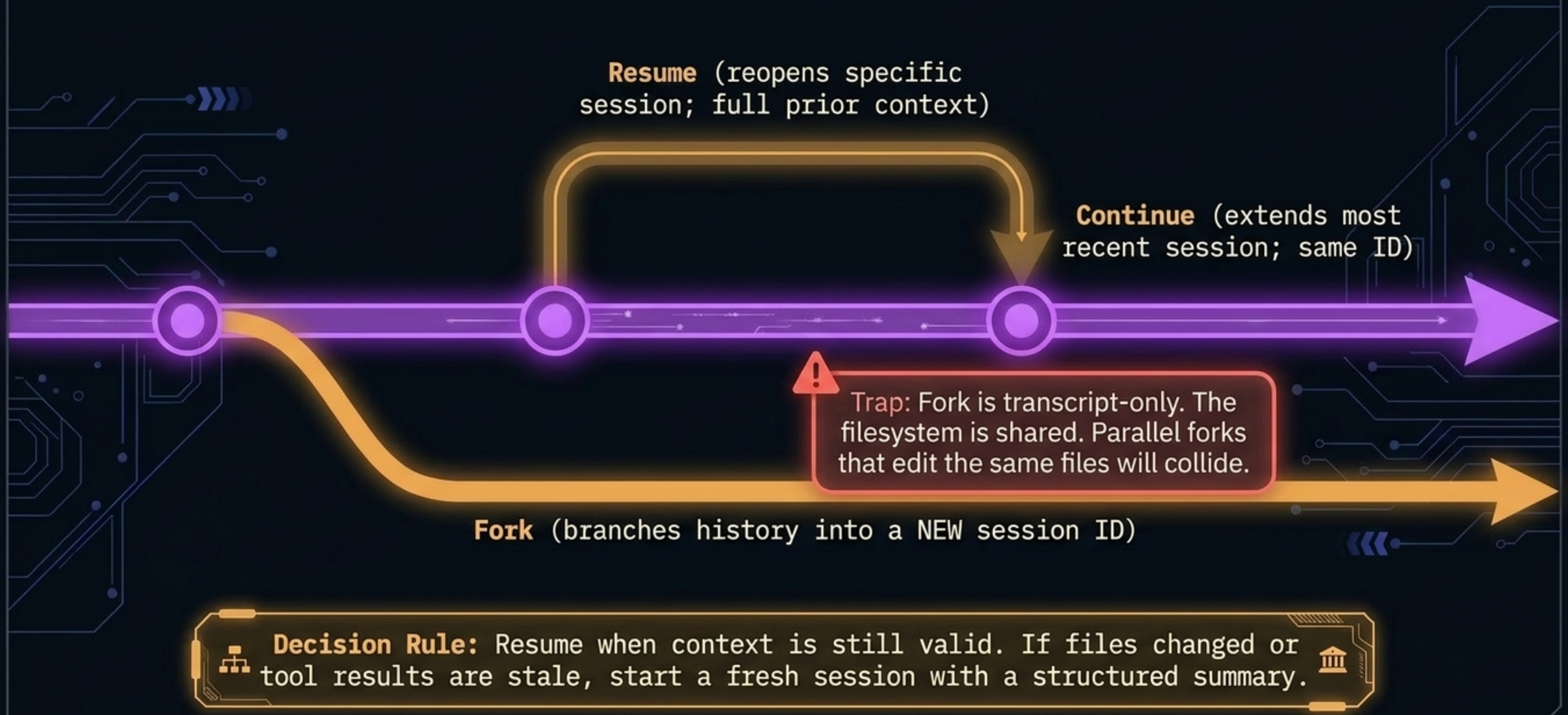
Hook Precedence Hierarchy
Deny > Defer > Ask > Allow
Note: One deny blocks the entire call.



The Redirect Fix:

The permissionDecisionReason must not just block the model; it must explicitly name the alternative path. This redirects Claude to the right path instead of causing infinite failure loops.

Session State: Resume vs. Fork



Decision Rule: Resume when context is still valid. If files changed or tool results are stale, start a fresh session with a structured summary.

Structured Tool Errors (MCP)

Mechanics: Set **isError: true** in the tool result payload. Include structured error content so the agent can decide its next move.

Transient	Timeout, 503	isRetryable: true	**Action:** Bounded local retry.
Validation	Missing field	isRetryable: false	**Action:** Fix input schema and recall.
Permission	Lacks access	isRetryable: false	**Action:** Stop path, escalate.
Business	Policy violation	isRetryable: false	**Action:** Explain policy and offer alternative.

The Trap: Returning [] (empty list) on a timeout. This silent suppression makes the agent confidently hallucinate that data doesn't exist.

Tool Distribution & Least Privilege

Scope Visualization



Give each subagent only the tools for its role.

Expose narrow tools to synthesis agents instead of full database access.

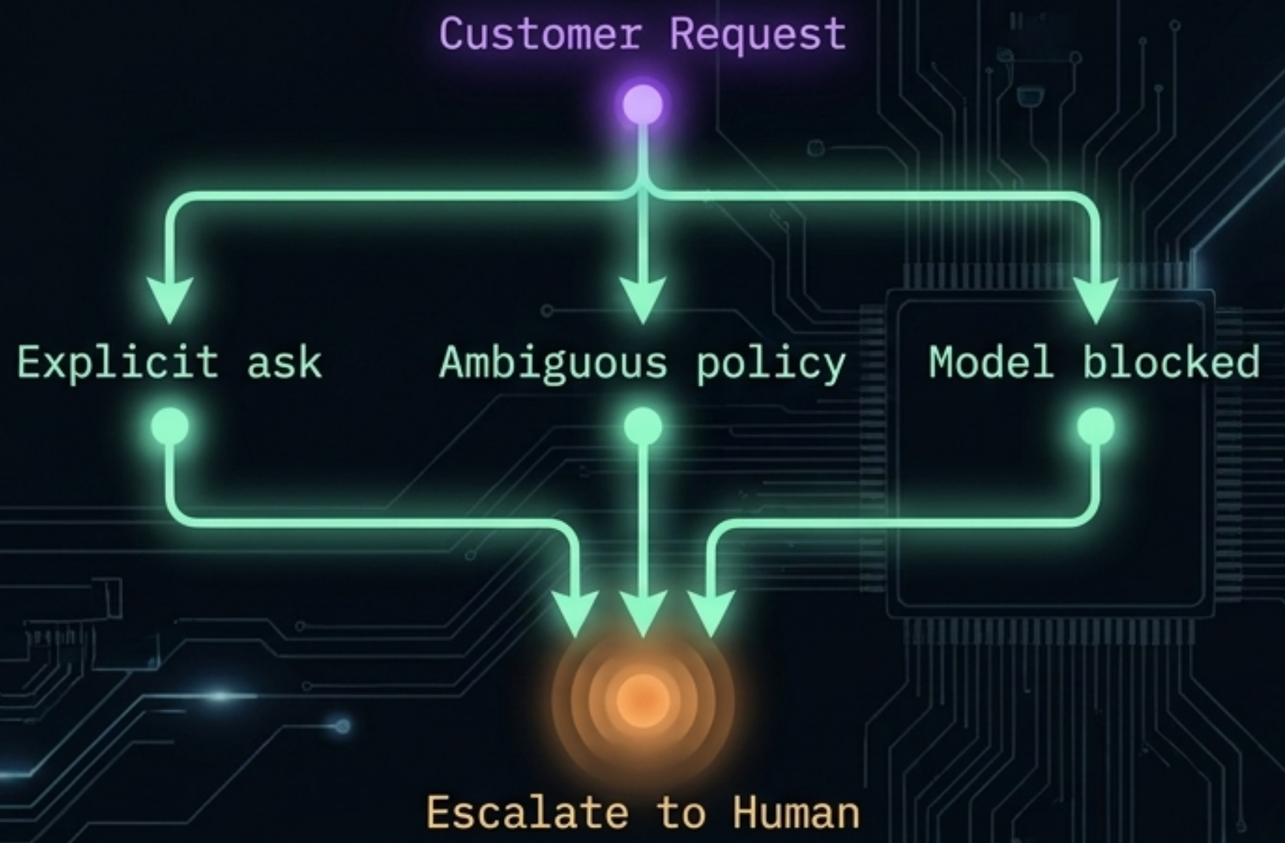
More tools = more schema tokens and mis-selection.

Forcing Tool Execution (tool_choice)



Trap: Forcing a tool every single turn breaks **end_turn**, converting an autonomous agent into a rigid, non-terminating pipeline.

Escalation & Ambiguity Resolution



Do NOT Use for Escalation

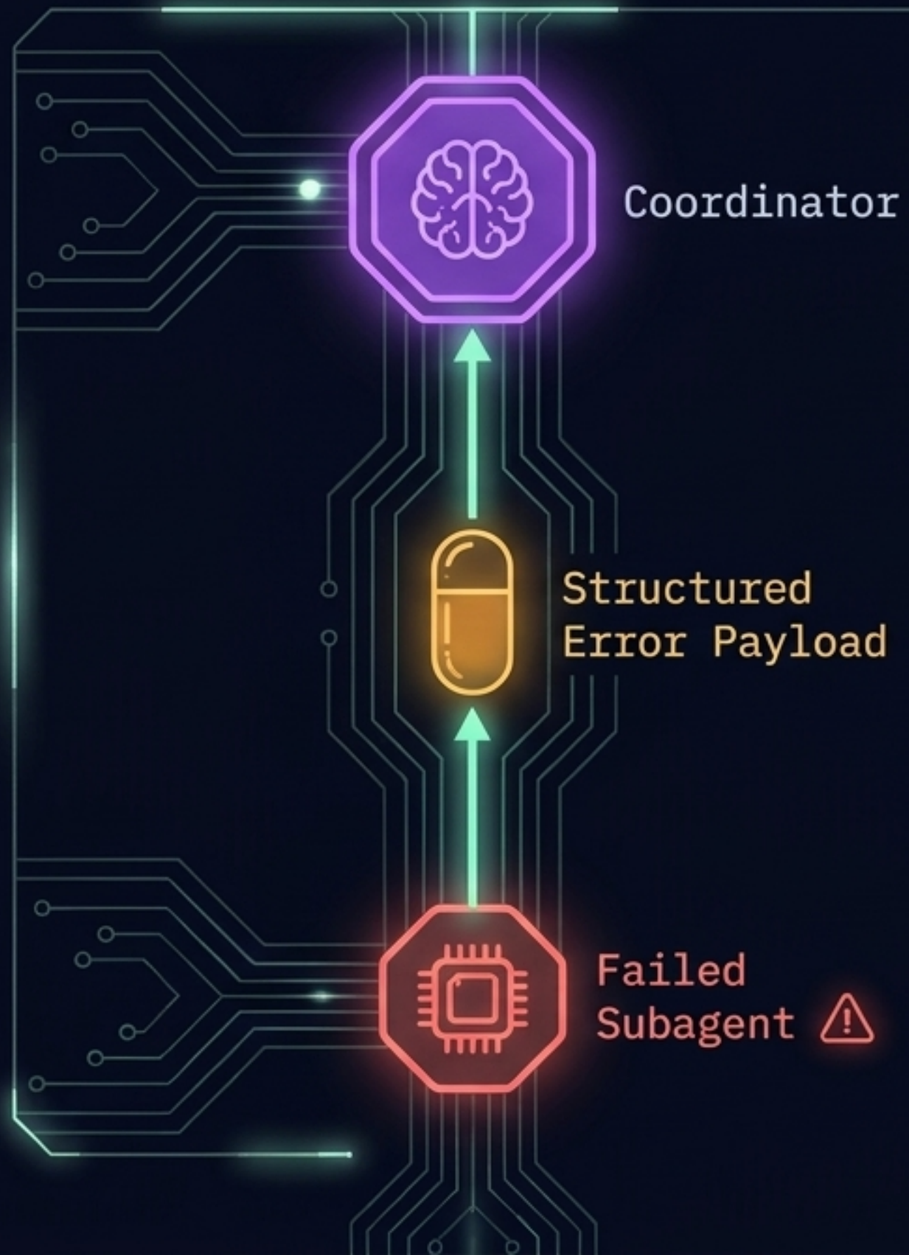
Sentiment (frustration does not equal complexity)

Self-reported model confidence (poorly calibrated)

Generic “looks complex” heuristics

Handling Multi-Concern Requests: Split message into distinct issues. Investigate efficiently in parallel. Provide one unified resolution addressing all items at once.

Error Propagation in Multi-Agent Systems



Anti-Patterns

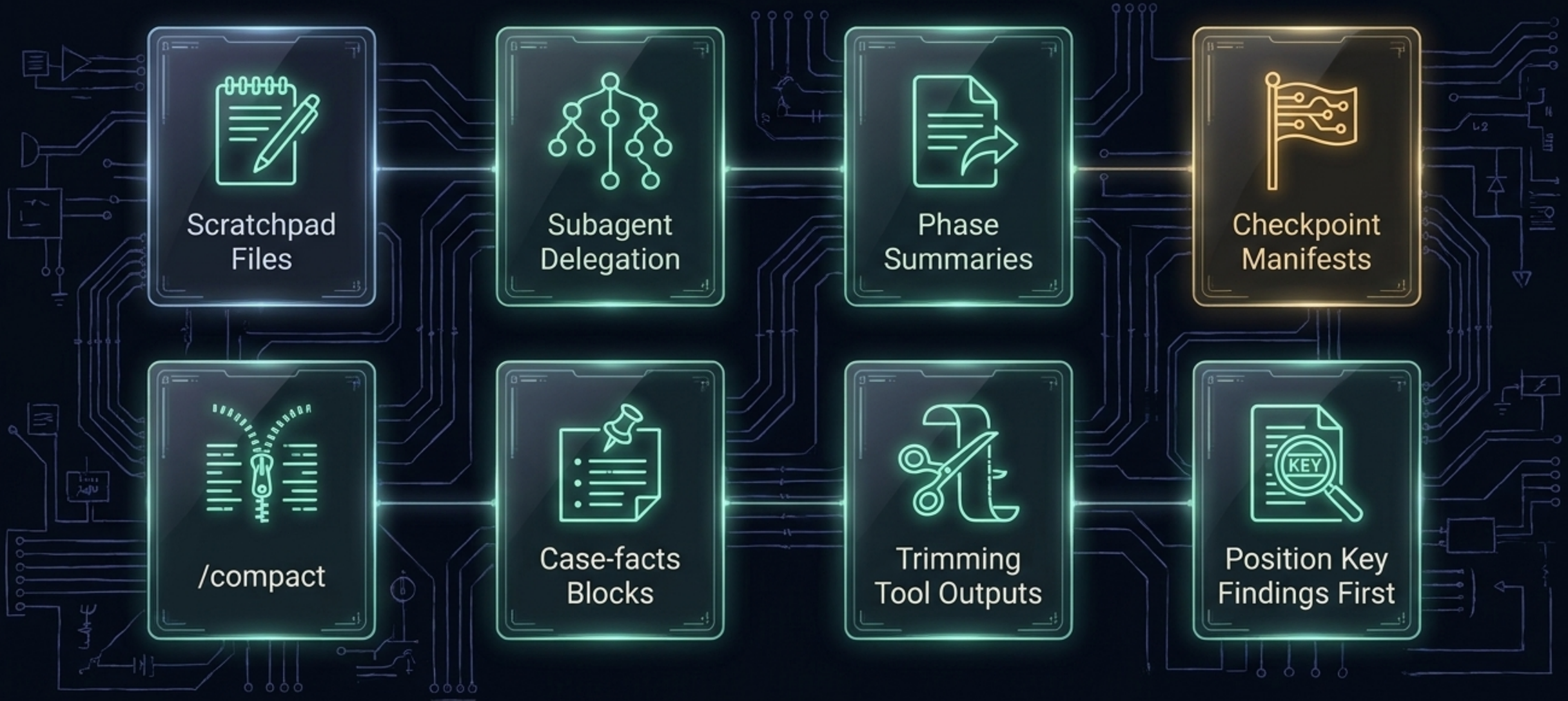
- Silent suppression: Coordinator assumes data doesn't exist.
- Killing the whole workflow: Throws away good partial work.
- Infinite local retry: Burns budget, hides outages.

The Fix

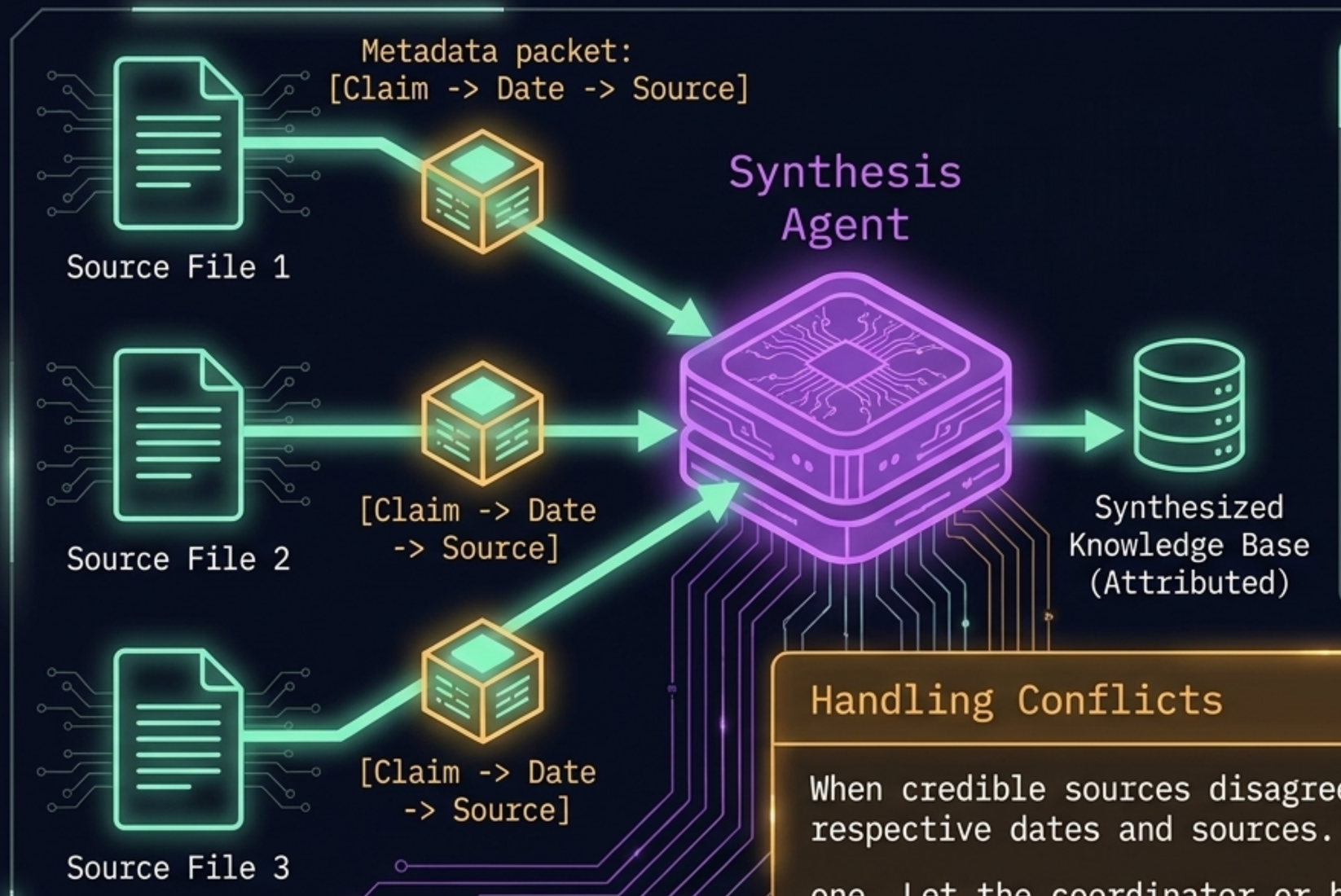
- Fail gracefully via bounded local retries.
- Propagate structured error context with partial results up to the coordinator.
- Let the coordinator map alternative paths.

Managing Long Codebase Context

Symptoms of context degradation: inconsistent answers, referencing generic patterns instead of found classes, forgotten constraints.



Provenance & Multi-Source Synthesis



The Requirement

Subagents must return every claim with its source URL, location, and publication date.

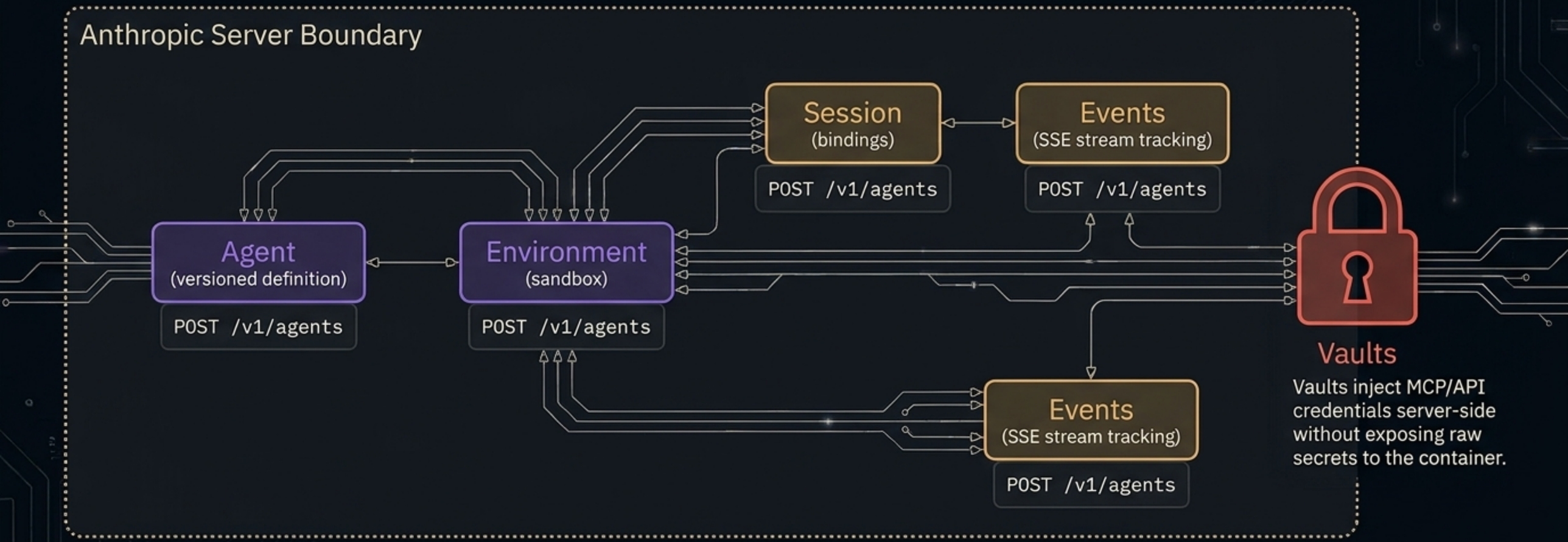
Synthesis Rule

The synthesis agent must preserve and merge these mappings.
Summarizing without attribution destroys utility.

Handling Conflicts

When credible sources disagree, report BOTH values with their respective dates and sources. Never silently average or pick one. Let the coordinator or human reconcile.

Claude Managed Agents Architecture



Compliance Trap: Managed Agents is stateful and NOT BAA/HIPAA/ZDR eligible. For regulated data, run the SDK loop locally.