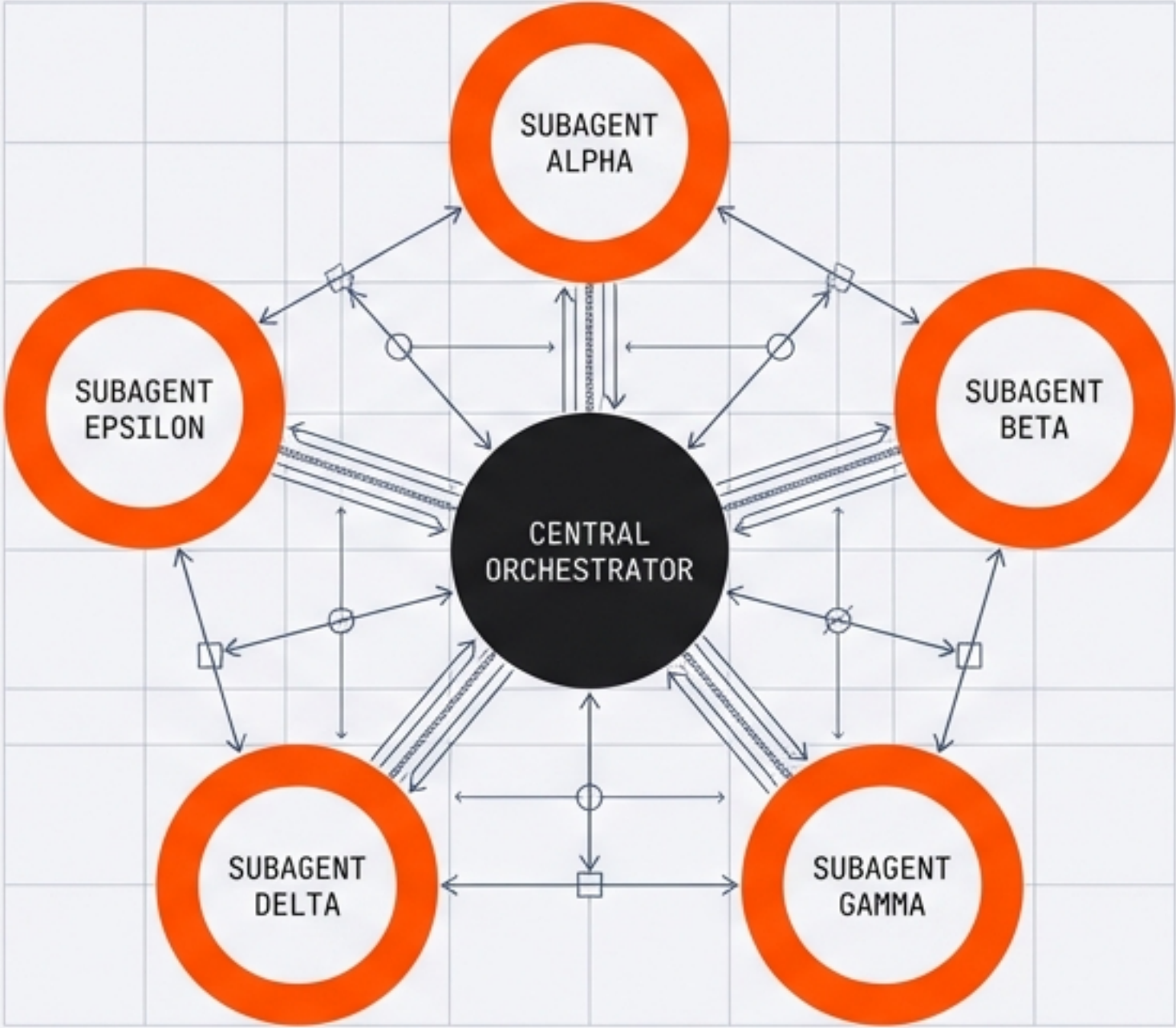


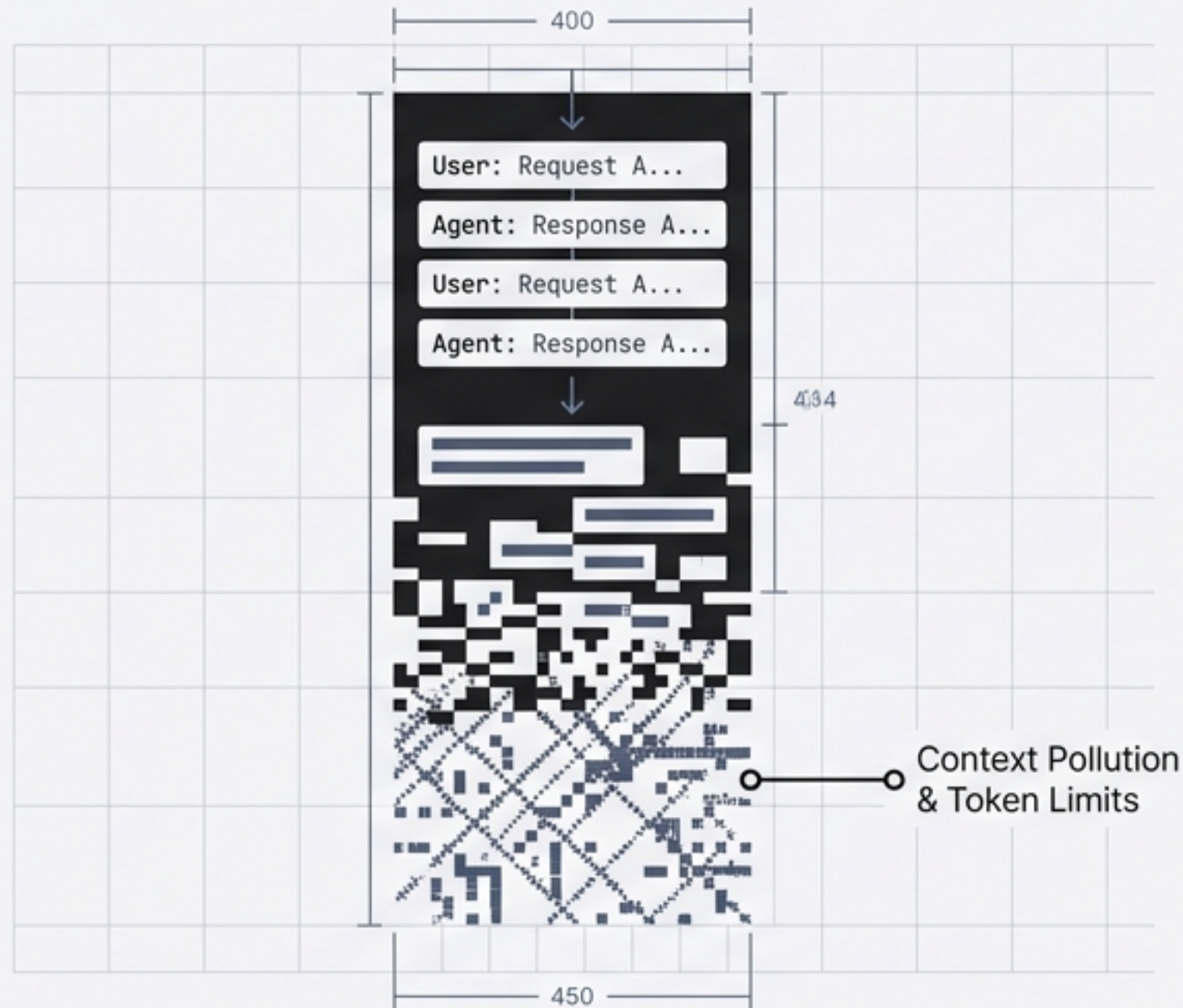
CLAUDE CODE ORCHESTRATION

Mastering Subagents, GSD Frameworks, and Parallel Architecture

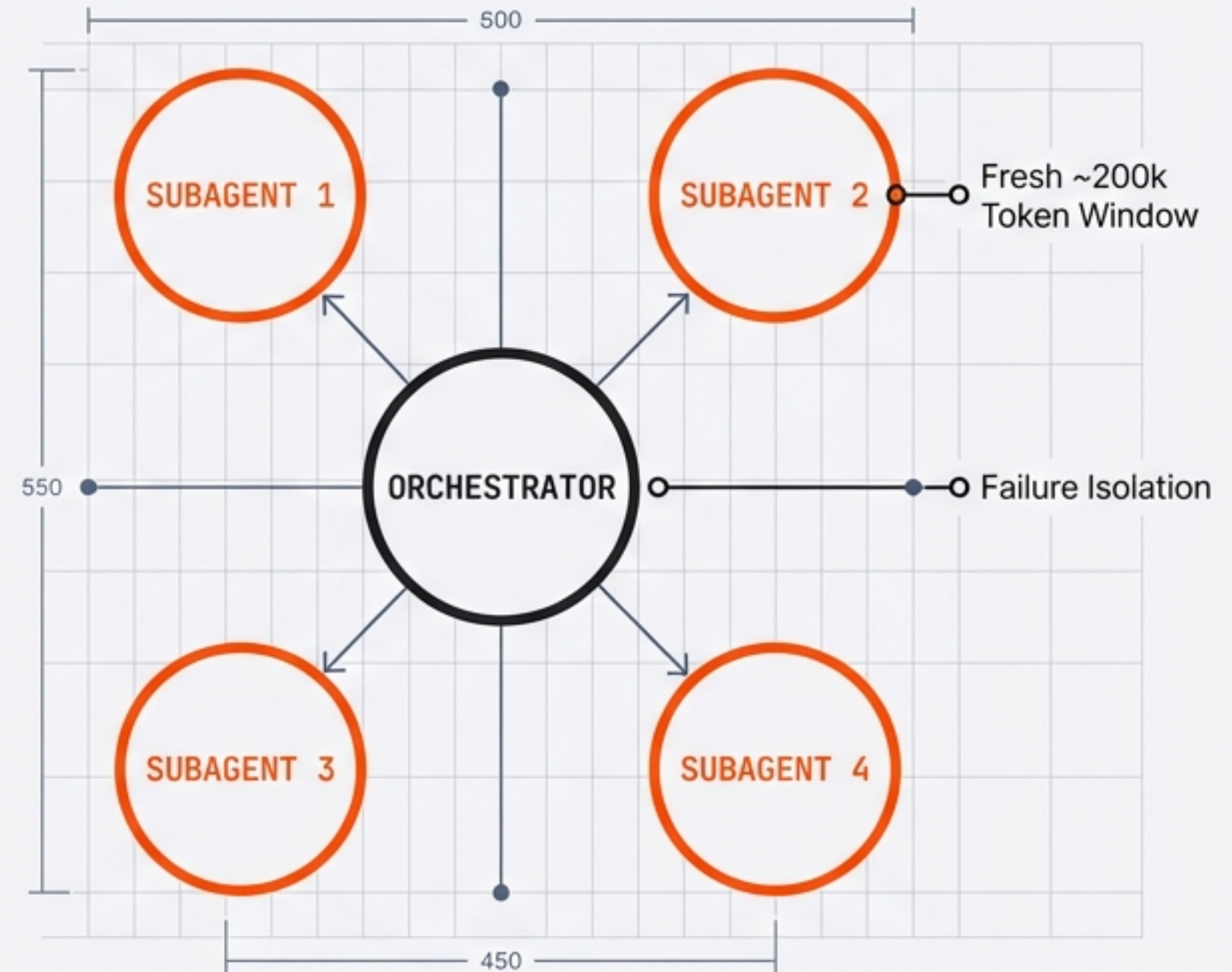


The constraints of linear chat require a distributed architecture

LINEAR CHAT (DEGRADING)

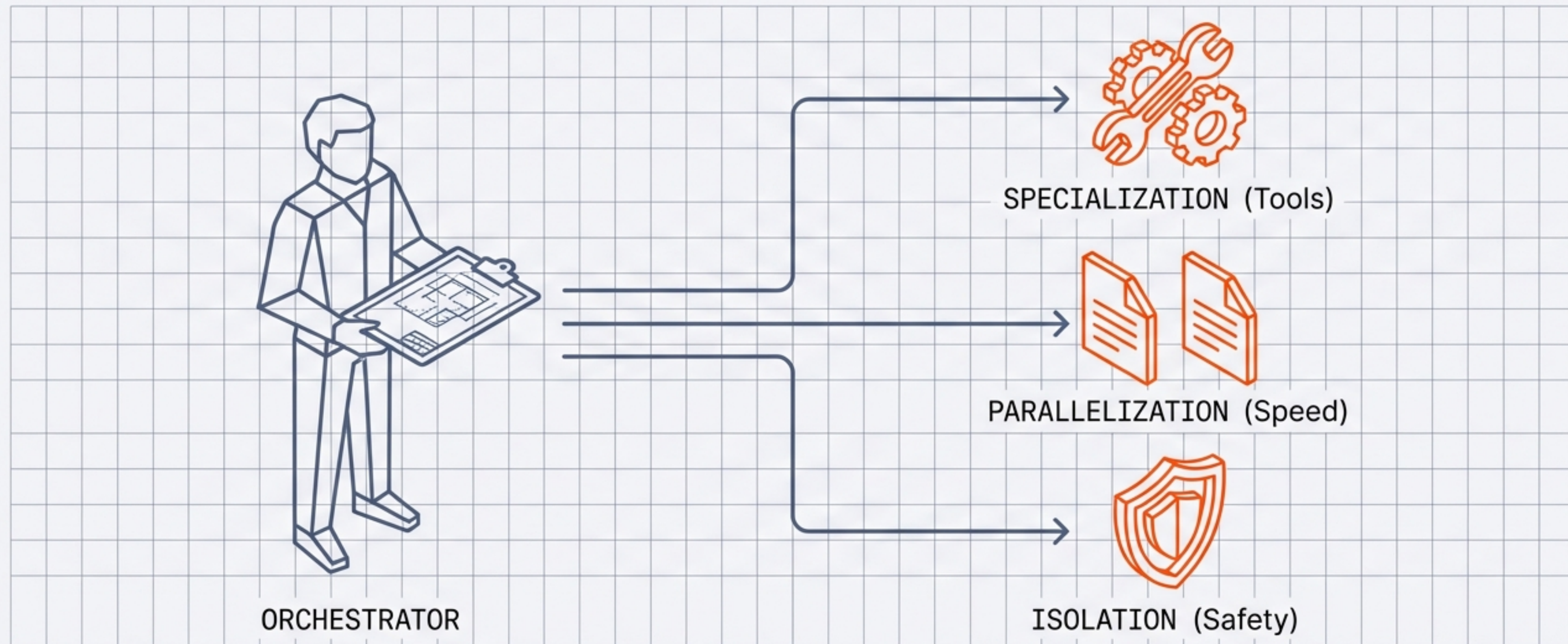


DISTRIBUTED SUBAGENTS (ISOLATED)



Shift your mental model from Chatter to Orchestrator

You are no longer just prompting; you are dispatching expert contractors for specific jobs.



The Core Agent Roster



Bash

Role:
The Hands

Tools:
Bash CLI Only
JetBrains Mono

Use Case:
Git operations,
terminal execution
Inter

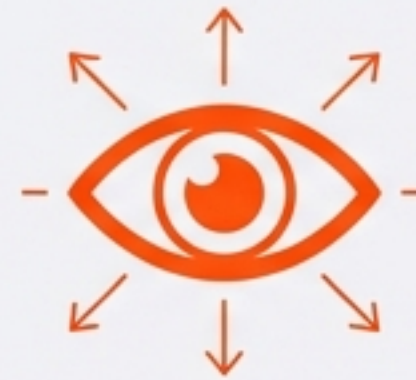


General

Role:
The Brain

Tools:
All Tools
JetBrains Mono

Use Case:
Complex multi-step
tasks, content editing
Inter



Explore

Role:
The Eyes

Tools:
Read-Only (No Edit)
JetBrains Mono

Use Case:
Codebase mapping,
"Where is X?",
Architecture
Inter



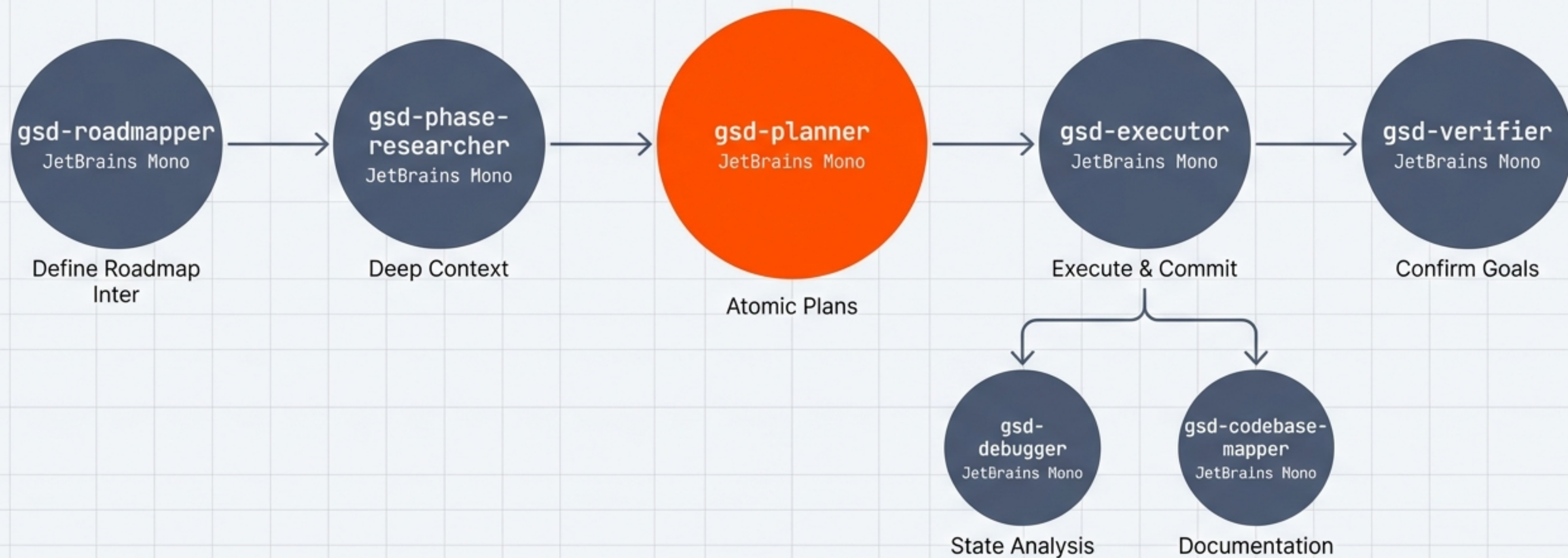
Plan

Role:
The Architect

Tools:
Read-Only
JetBrains Mono

Use Case:
System design,
feature strategy
Inter

The GSD Framework: A lifecycle for autonomous execution



Automatic triggers vs. Manual orchestration

AUTOMATIC TRIGGERS (Intent Detection)

➤ "Where is X?"



Explore Agent

➤ "Large Pattern Search"



General Agent

➤ Architecture Query



Explore (Very Thorough)

➤ Non-trivial Request



Plan Agent

MANUAL GOVERNANCE (CLAUDE.md)

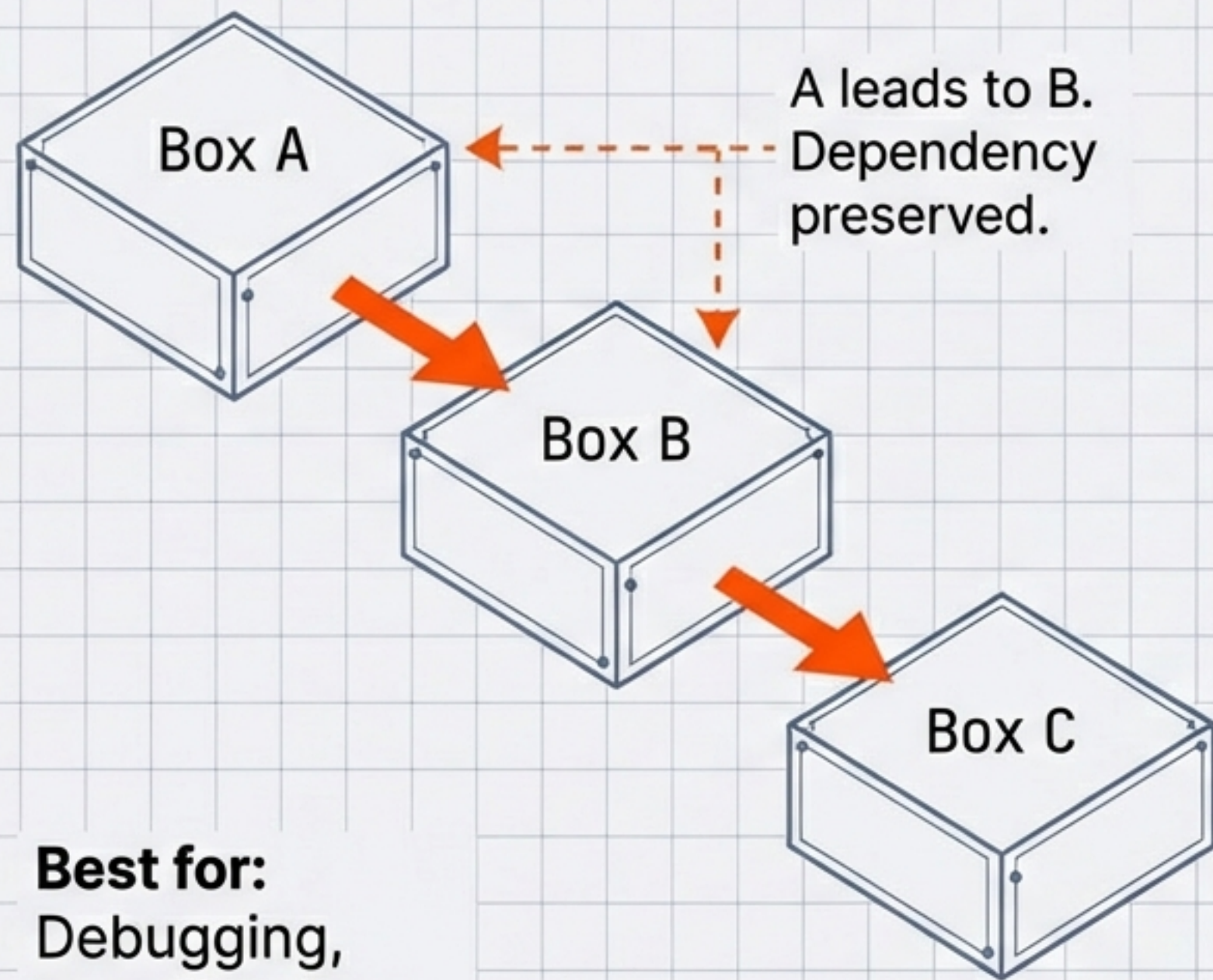
CLAUDE.md

Rule: For complex tasks (3+ steps), ask me first.

This looks complex. Sequential, Parallel, or Direct?

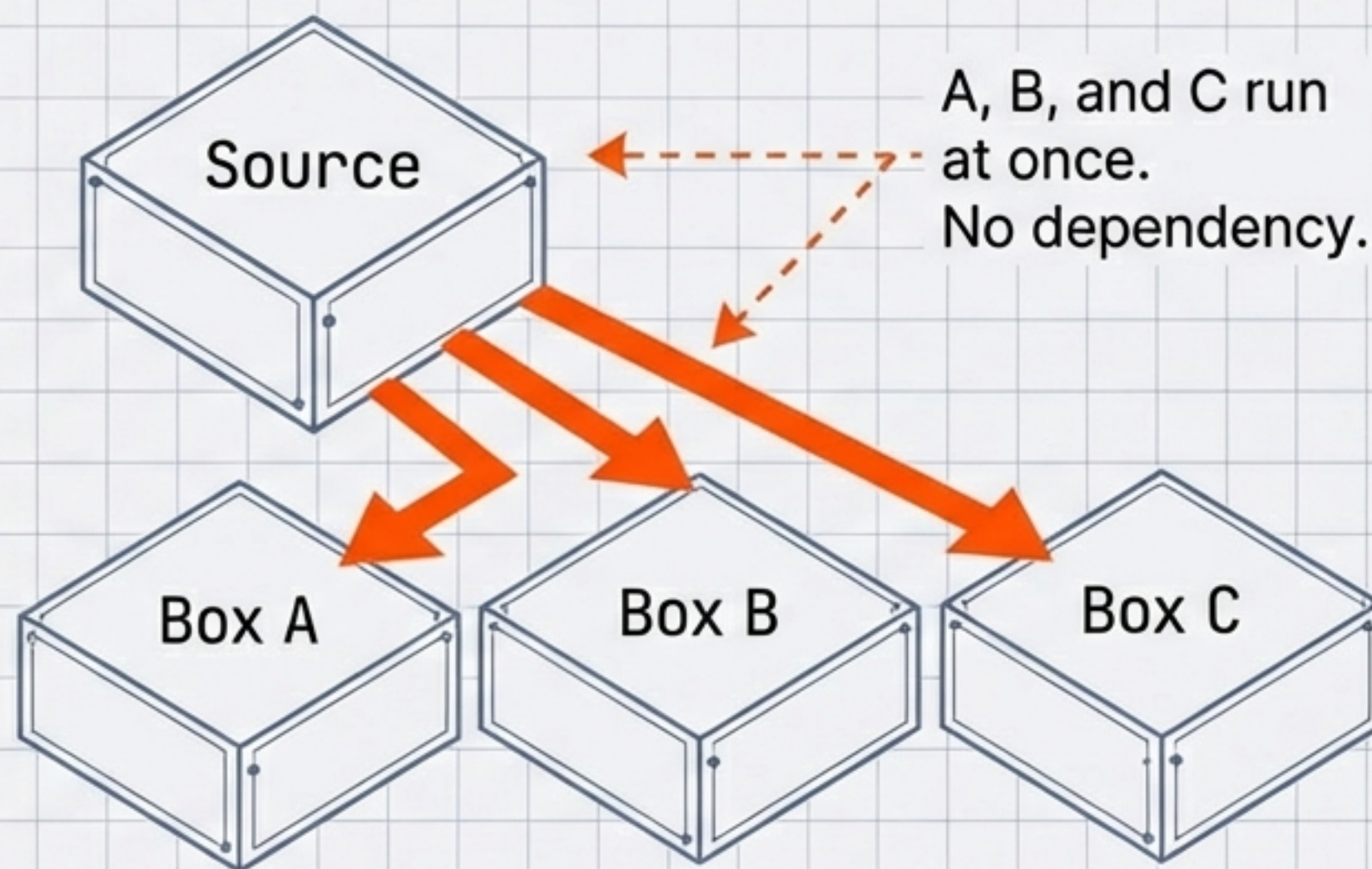
Execution Strategy: Sequential vs. Parallel

SEQUENTIAL (Accuracy First)



Best for:
Debugging,
Investigations

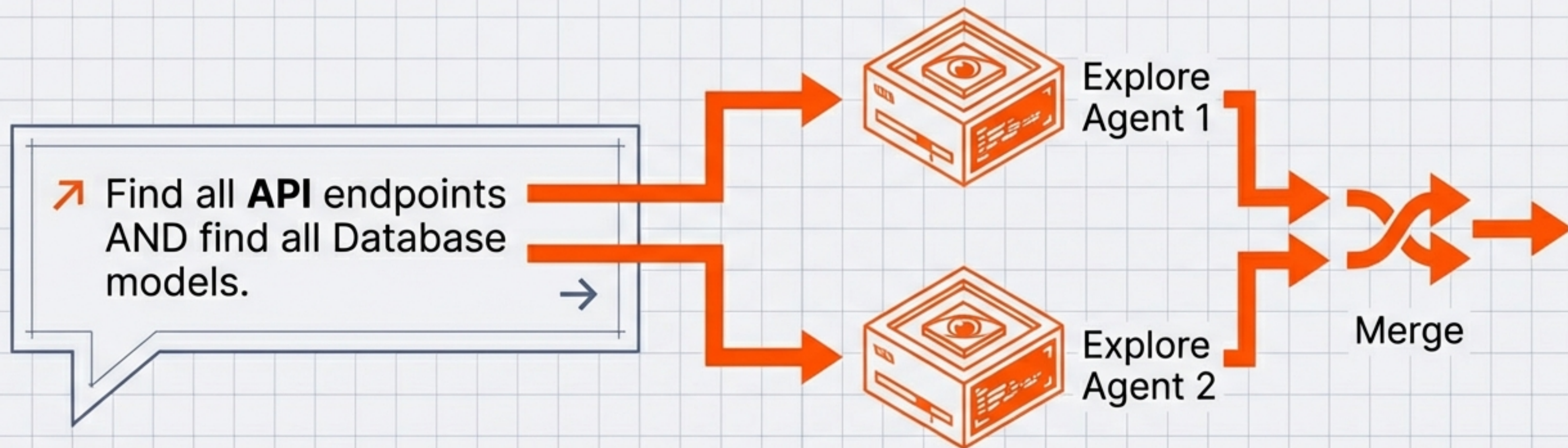
PARALLEL (Speed First)



Best for: Independent searches,
Discovery

Deep Dive: Optimizing for Parallel Speed

Drastically reduces wait time for research phases.



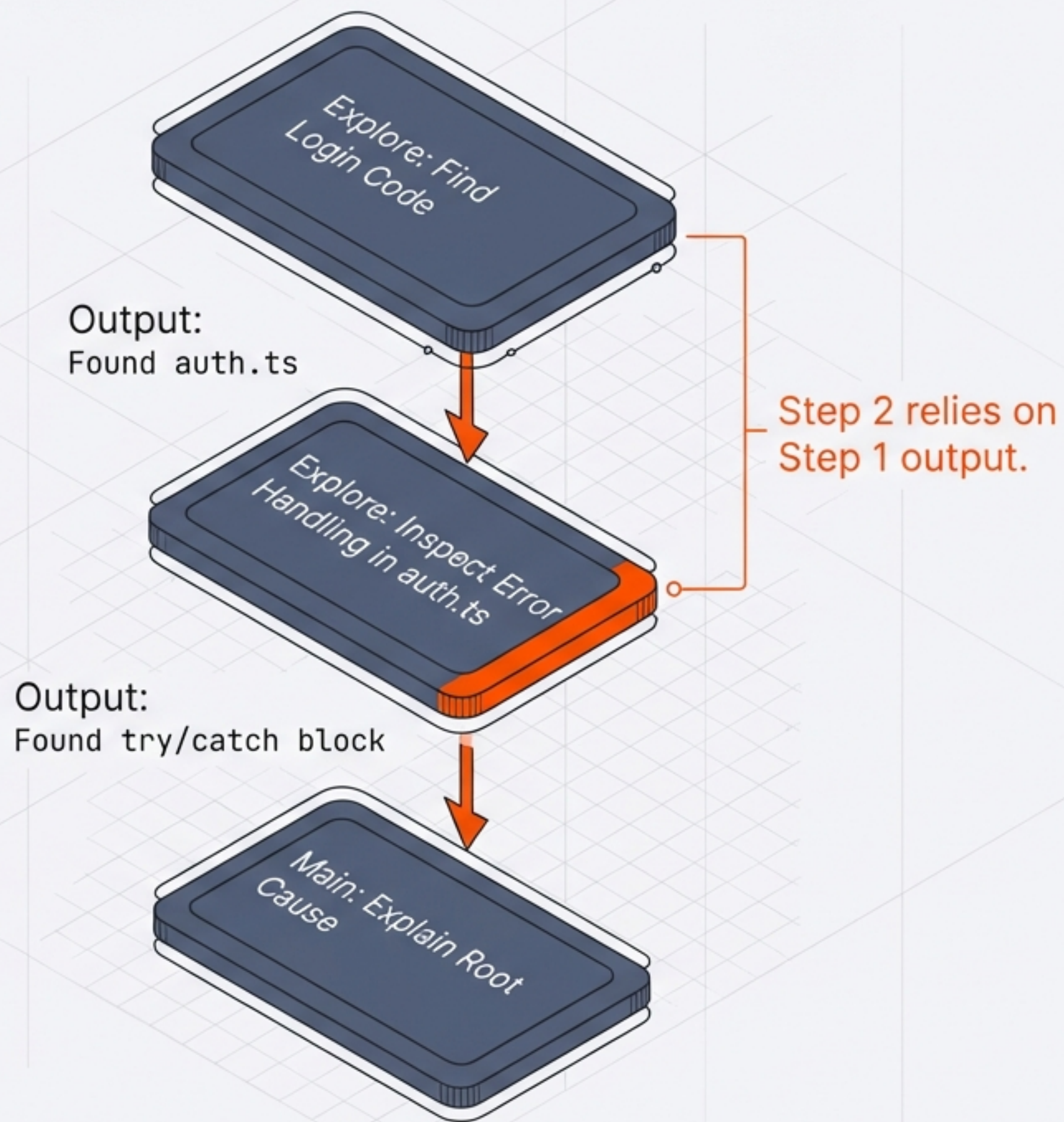
THE CRITICAL RULE

Parallel agents must be spawned in a single tool message. Splitting requests across multiple messages forces sequential default.

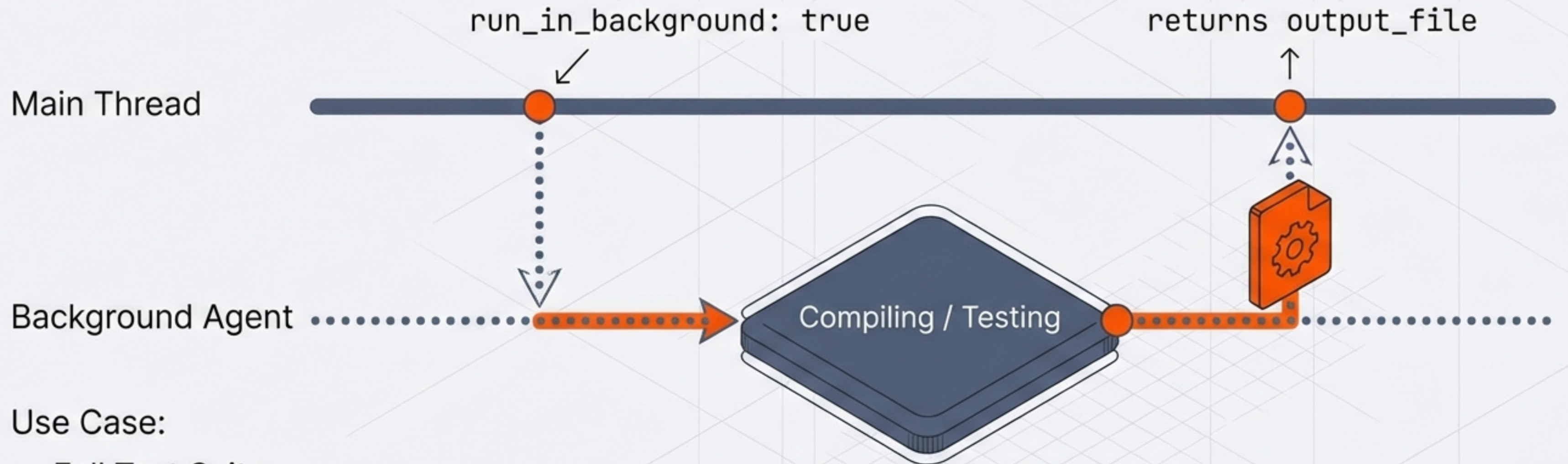
Deep Dive: Ensuring Accuracy with Sequential Logic

Preserves logical chain of thought to prevent hallucination.

Use Case: "Why is login failing?"



Asynchronous productivity with Background Agents



Use Case:

- Full Test Suites
- Project Builds
- Massive Grep Searches

Codifying governance in CLAUDE.md

```
# Exploration Rules
- thoroughness: "medium" (General Questions)
- thoroughness: "very_thorough" (Architecture)

# Mandatory Governance
- rule: Always use Explore for "Where is..."
- rule: Ask before spawning for tasks > 3 steps

# Strategy Defaults
- independent_tasks: "parallel"
- debugging: "sequential"
```

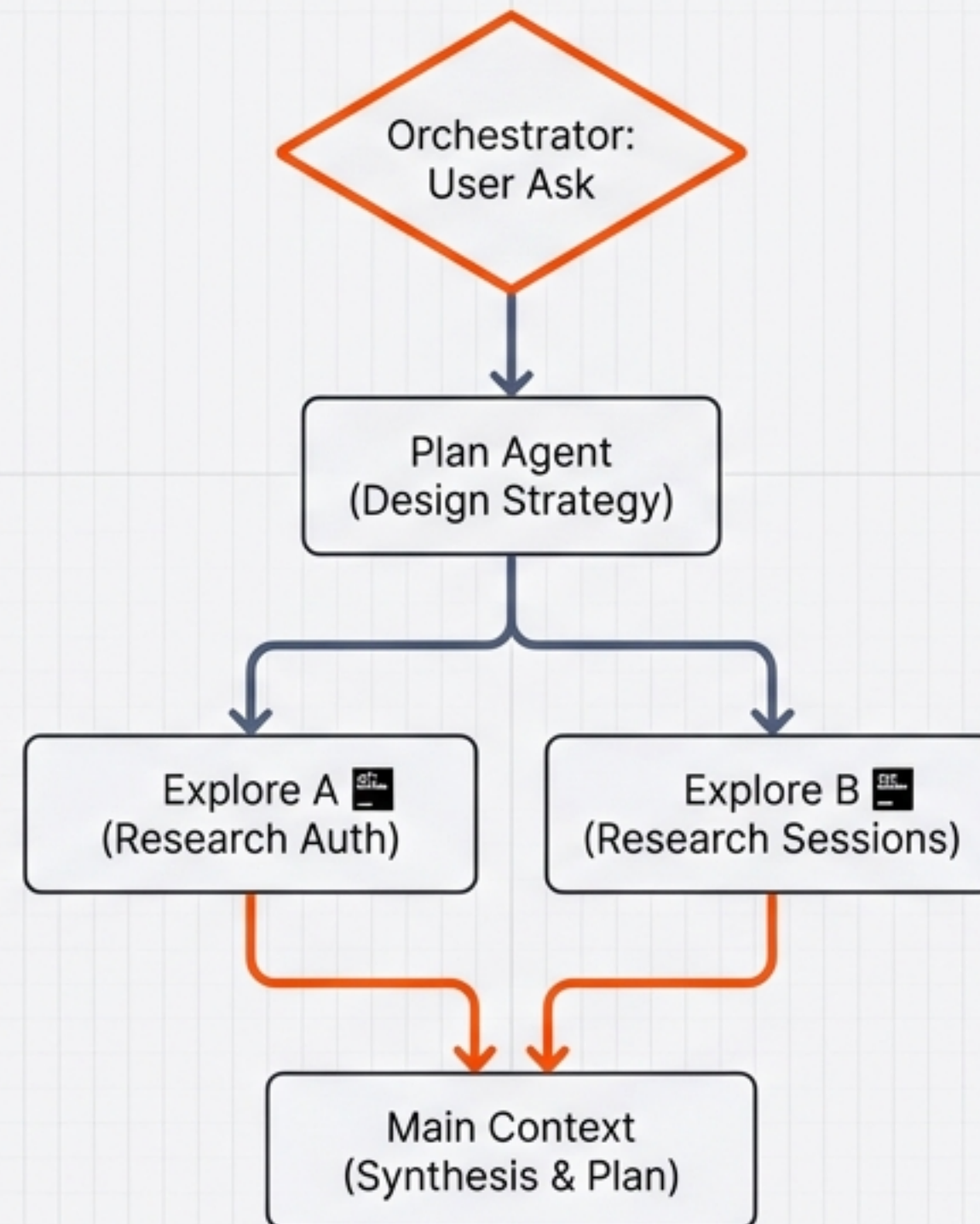
Precise configuration rules define Claude's behavior and ensure adherence to team policies.

Agent Selection Cheat Sheet

TASK TYPE	AGENT	SETTING
Where is X?	Explore (JetBrains Mono)	Quick / Medium
How does Y work?	Explore (JetBrains Mono)	Medium
Architecture Overview	Explore (JetBrains Mono)	Very Thorough
Bug Fixing	General (JetBrains Mono)	-
Feature Planning	Plan (JetBrains Mono)	-
Git / CLI Work	Bash (JetBrains Mono)	-
Search & Replace	General (JetBrains Mono)	-

Workflow Pattern: Refactoring Authentication

A Mixed Strategy: Plan -> Parallel Research -> Implementation



Rules of Engagement: Best Practices

MINIMAL CONFIGURATION



1. Always use **Explore** for codebase questions.



2. Ask before complex tasks.

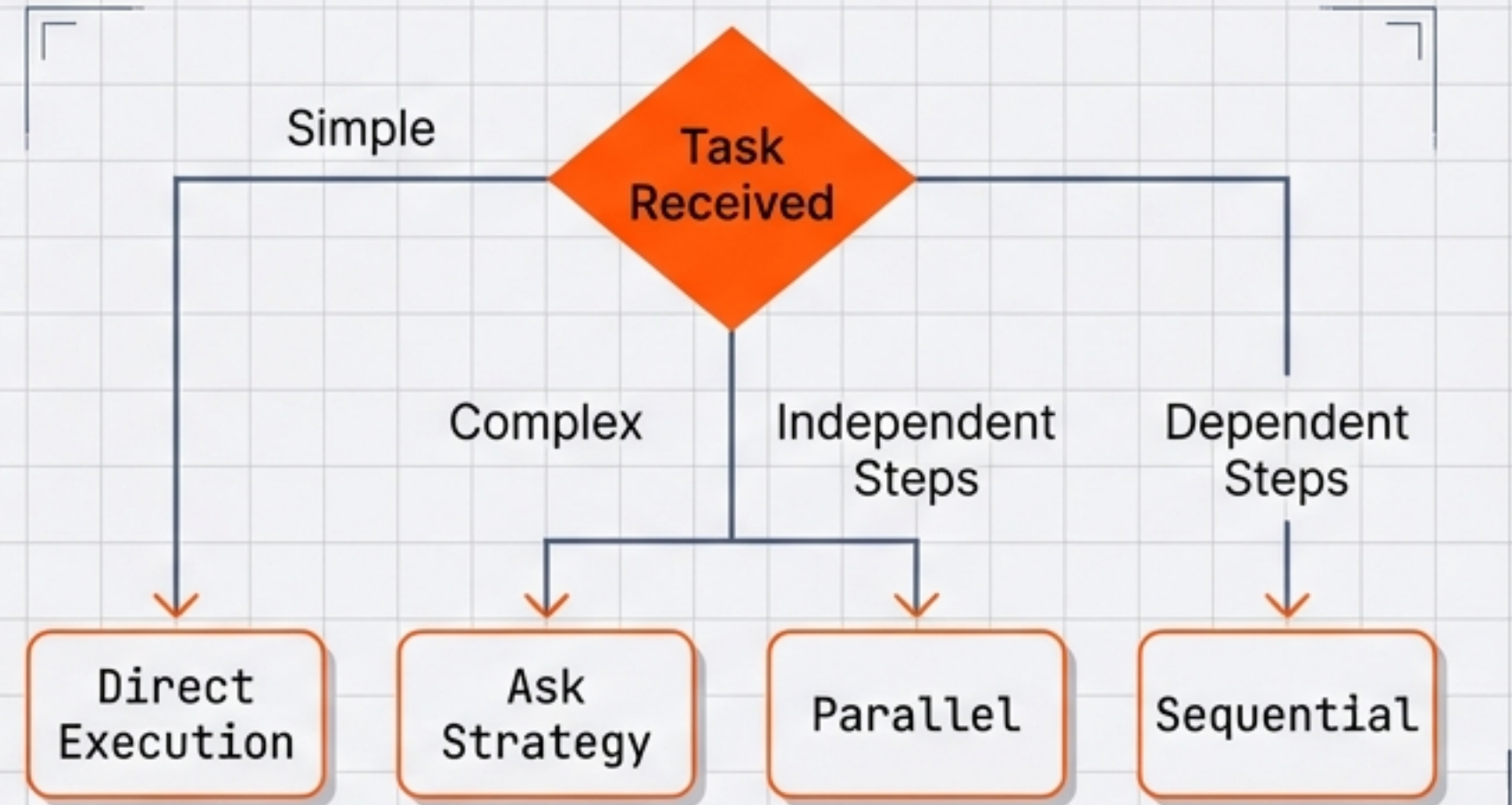


3. Default to **Parallel** for independent work.



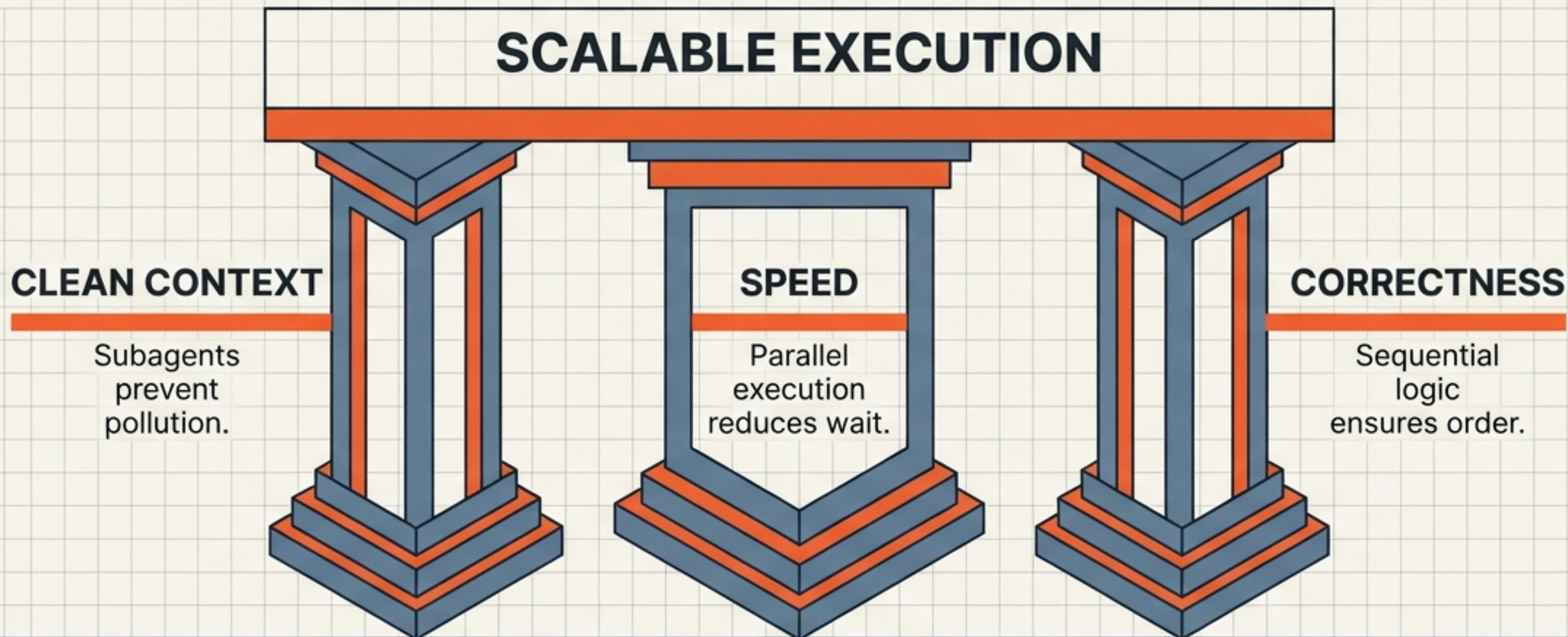
4. Use **Background Agents** for slow tasks.

DECISION LOGIC



Summary: The Pillars of Orchestration

A Mixed Strategy: Plan → Parallel Research → Implementation



Mastery comes from configuration. Tune your CLAUDE.md to treat the AI not as a chatbot, but as an orchestrated workforce.