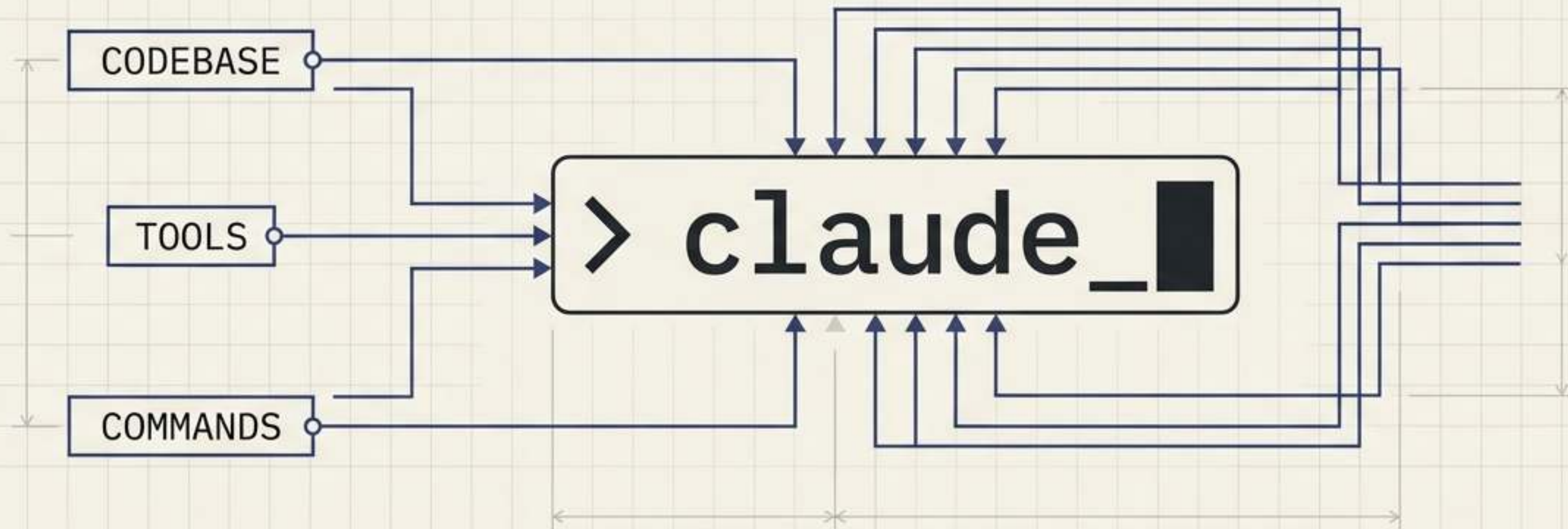


MASTERING CLAUDE CODE

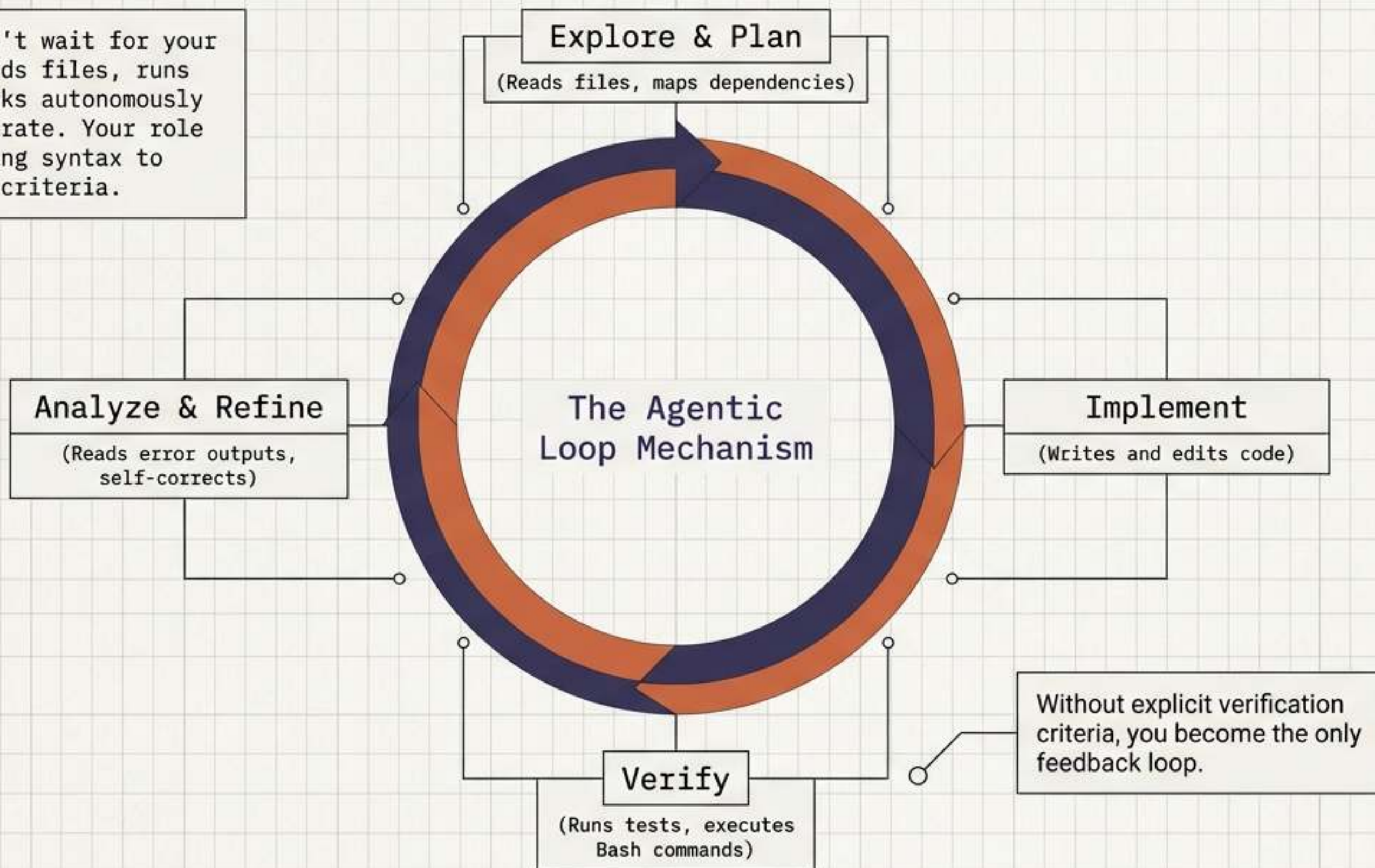
The Operator's Manual for Agentic Environments



TARGET AUDIENCE: ENGINEERS & TECH LEADS | SYSTEM: ANTHROPIC CLAUDE CODE | MODE: ADVANCED ORCHESTRATION

Transitioning from Chatbot to Agentic Loop

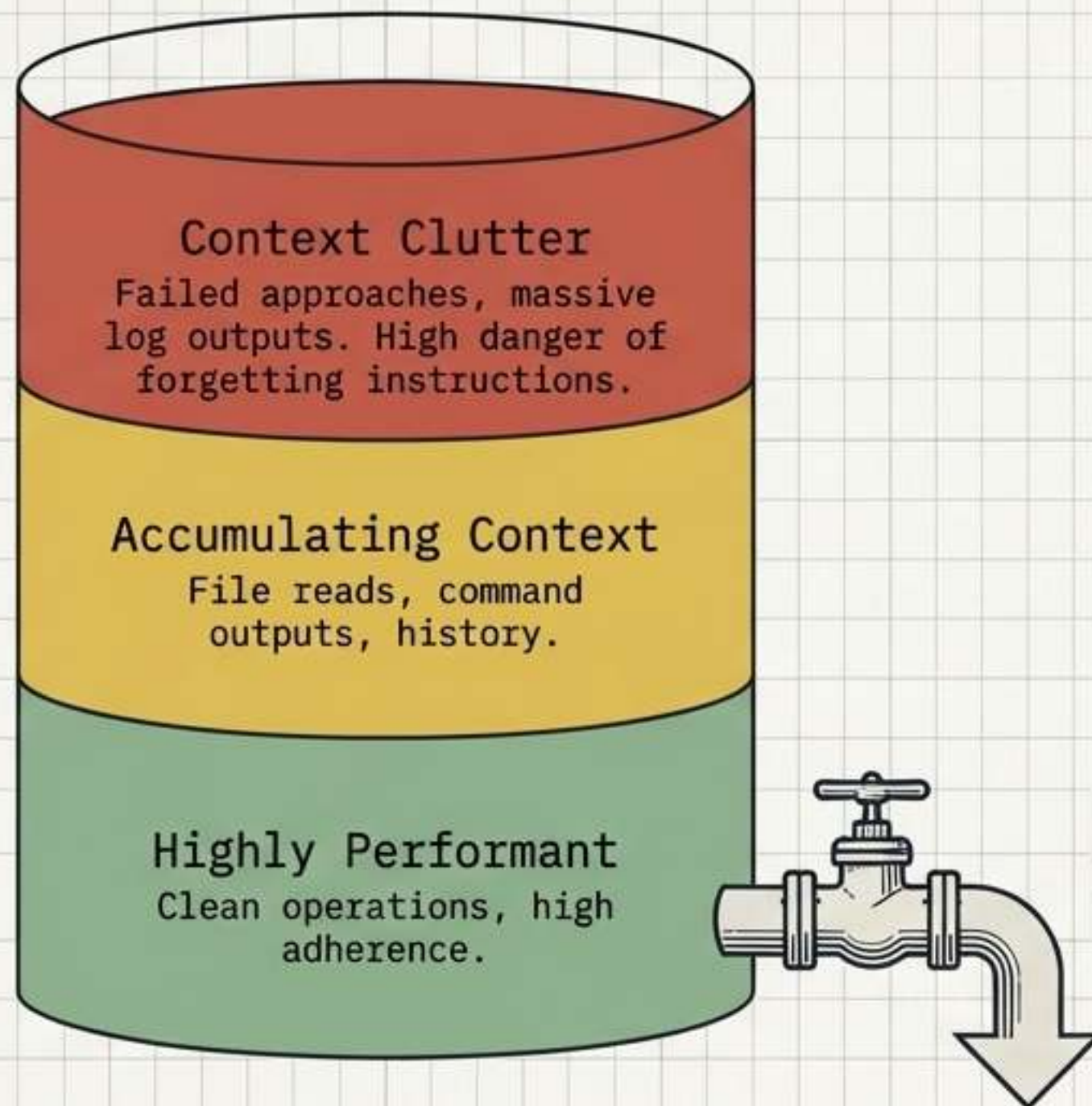
Claude Code doesn't wait for your questions. It reads files, runs commands, and works autonomously while you orchestrate. Your role shifts from writing syntax to defining success criteria.



The Core Constraint: Managing the Context Fuel Tank

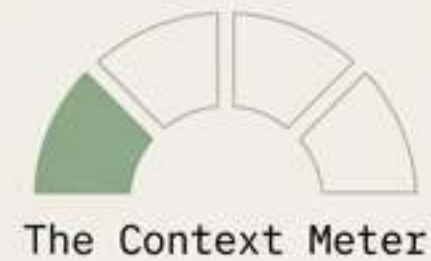
LLM performance degrades as context fills. Every message, file read, and command output consumes this limited resource.

The Context Fuel Tank

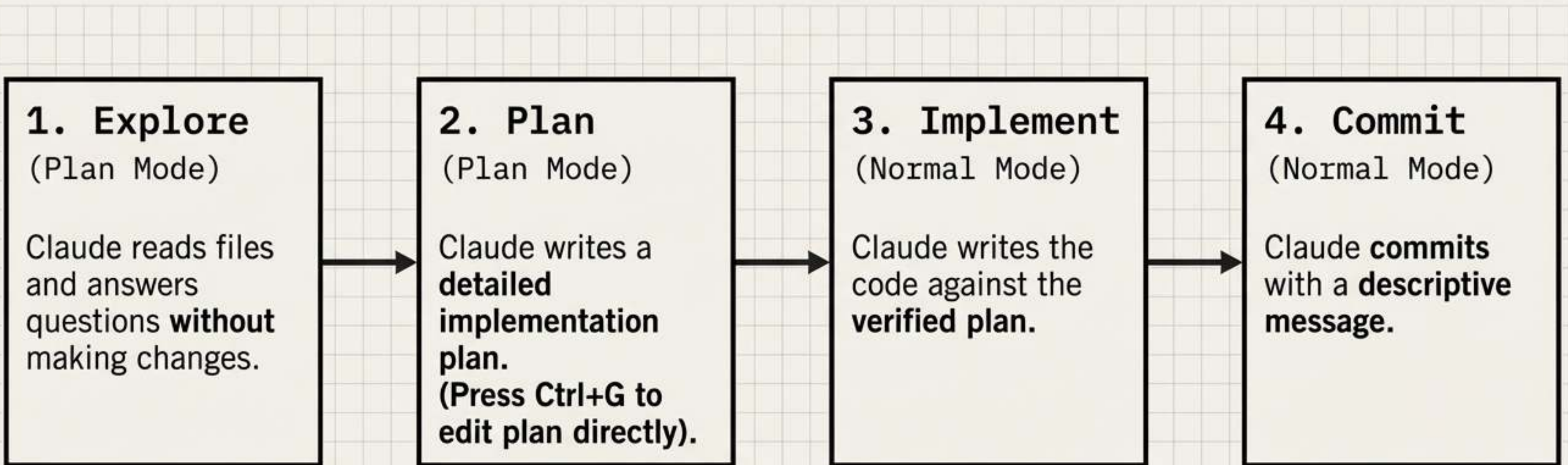


Use `/clear` or `/compact` to drain the tank back to the green zone.

Separate Research from Execution Stop Mode to Prevent Hallucinations

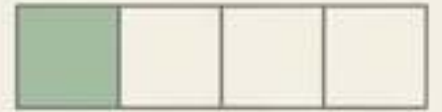


Jumping straight to code produces solutions for the wrong problems.
Enforce a deliberate phase-gate approach.



Pro-Tip: If you can describe the exact diff in one sentence, skip the plan. Otherwise, separate the modes.

Prompting for Autonomy requires Scoping and Verification

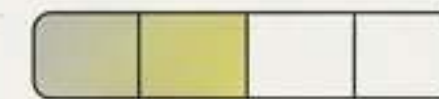


The Context Meter

Prompt Transformation Matrix

Strategy	Rookie Prompt	Pro Prompt
Scope the Task	add tests for foo.py	write a test for foo.py covering the logged-out edge case. avoid mocks.
Reference Patterns	add a calendar widget	look at HotDogWidget.php. follow this pattern to build a calendar widget without external libraries.
Symptom & Fix	fix the login bug	users report timeout failures. check src/auth/token refresh. write a failing test, then fix it.
Visual Verification	make the dashboard look look better	[paste screenshot] implement this design. take a screenshot of the result and compare. list differences and fix.

Feed the Engine: Piping Rich Context



The Context Meter

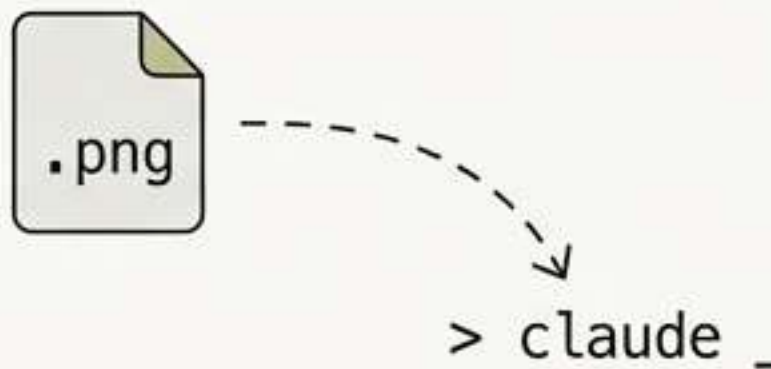
Claude can infer intent, but it cannot read your mind. Use operators and pipes to inject precise, bounded context directly into the agent.

```
● ● ● The @ Operator  
  
fix the routing bug in  
@src/router.ts
```

Limits context window burn by targeting specific files.

```
● ● ● Piping Data  
  
cat error.log | claude
```

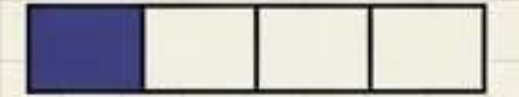
Feeds exact stack traces directly to the agent.

```
● ● ● Image Drag-and-Drop  
  

```

Visually verifies UI logic and fixes.

```
● ● ● Delegated Fetching  
  
Use 'gh issue view 123' to  
understand the bug, then  
fix it.
```

Teaching Claude to use CLI tools to pull its own context.



Pre-Flight: Architecting a Lean CLAUDE.md

CLAUDE.md loads into the context window *every session*. Bloated config files cause Claude to ignore actual instructions. **Treat it like code: review, prune, test.**

✓ KEEP (System Constraints)

- Bash commands Claude can't guess
- Project-specific architectural decisions
- Testing instructions and preferred runners
- Unique developer environment quirks

✗ CUT (Context Bloat)

- Standard language conventions
- Detailed API docs (link to them instead)
- File-by-file descriptions
- Self-evident practices (“write clean code”)

Pro-Tip: Use modular imports to keep the root file small:

```
<!-- @import ./docs/style-guide.md -->
```



Pre-Flight: The Ecosystem Matrix

Extend Claude's capabilities using targeted tools rather than bloated global instructions.

Skills

`.claude/skills/`

Domain knowledge and reusable workflows loaded *on-demand*. Examples: API conventions, repeatable workflows.

Subagents

`.claude/agents/`

Specialized assistants running in isolated context windows. Example: Security Reviewer exploring large codebases safely.

Hooks

`.claude/settings.json`

Deterministic scripts running at specific workflow triggers. Example: Forcing ESLint after every file edit.

MCP Servers

`claude mcp add`

External tool connections for external services. Examples: Querying databases, reading from Figma, updating issue trackers.

In-Flight: The Context Management Toolkit

Cockpit Controls

The Brake (Course-Correct)

`Esc` `[Esc]`

Stop Claude mid-action to prevent context pollution from a bad trajectory.

The Time Machine (Rewind)

`[Esc]` + `[Esc]` or `/rewind`

Restore conversation, code state, or summarize from a checkpoint.

The Compactor (Distill)

`/compact`

Forces Claude to summarize history, preserving key decisions while dumping raw token bloat.

The Nuke (Reset)

`/clear`

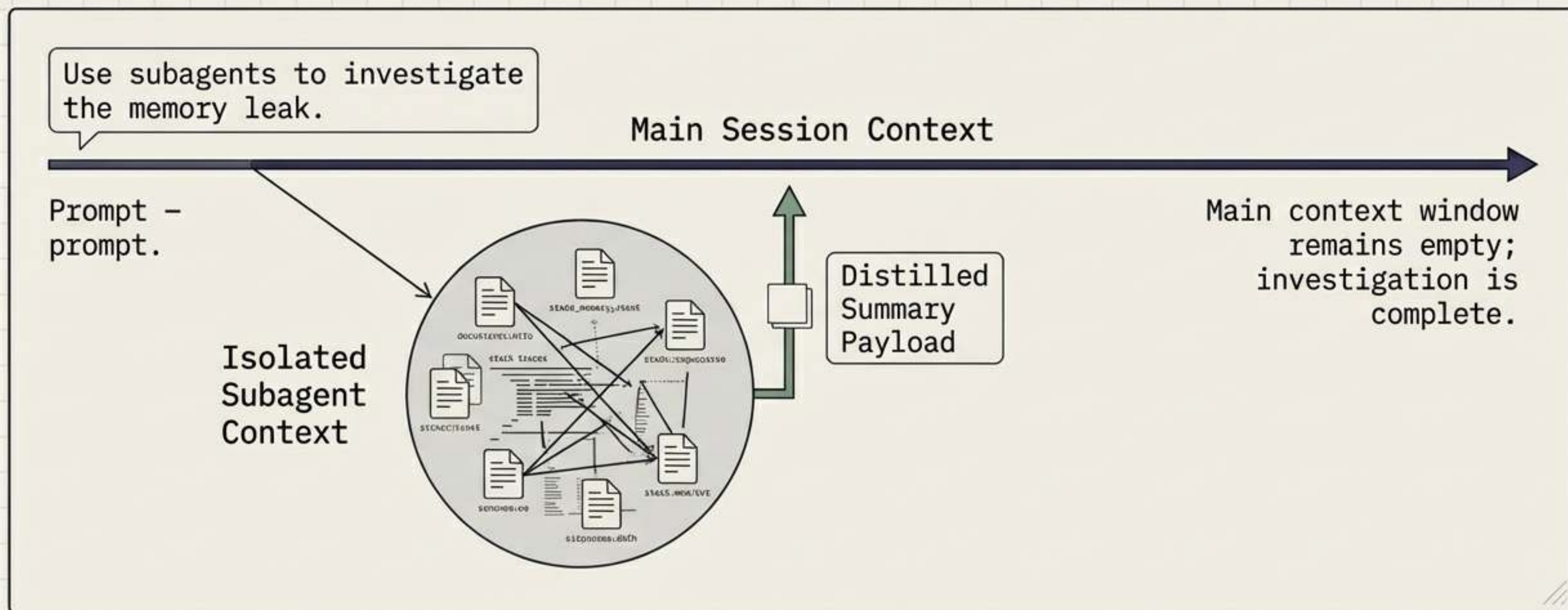
Completely resets the context window between unrelated tasks. A clean prompt beats accumulated corrections.

In-Flight: The Subagent Context Shield

The Context Meter



"Investigate this" is a dangerous prompt—it causes Claude to read hundreds of files, permanently polluting your context window. Subagents solve this.

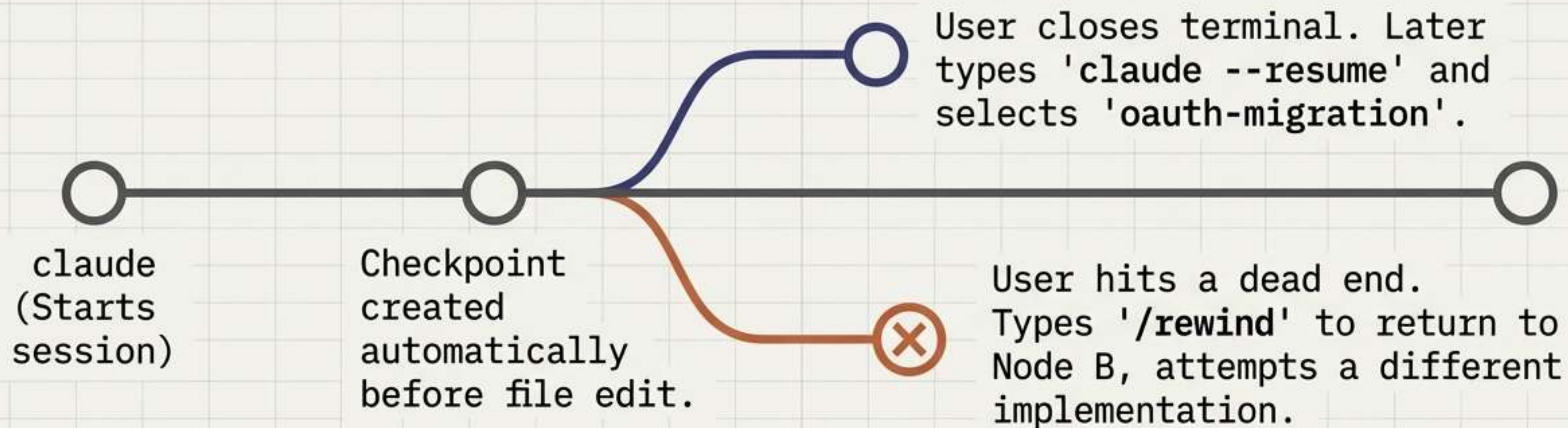




The Context Meter

Fleet Management: Treat Sessions Like Git Branches

Conversations are persistent and reversible. When a task spans multiple sessions, you don't have to re-explain the context.



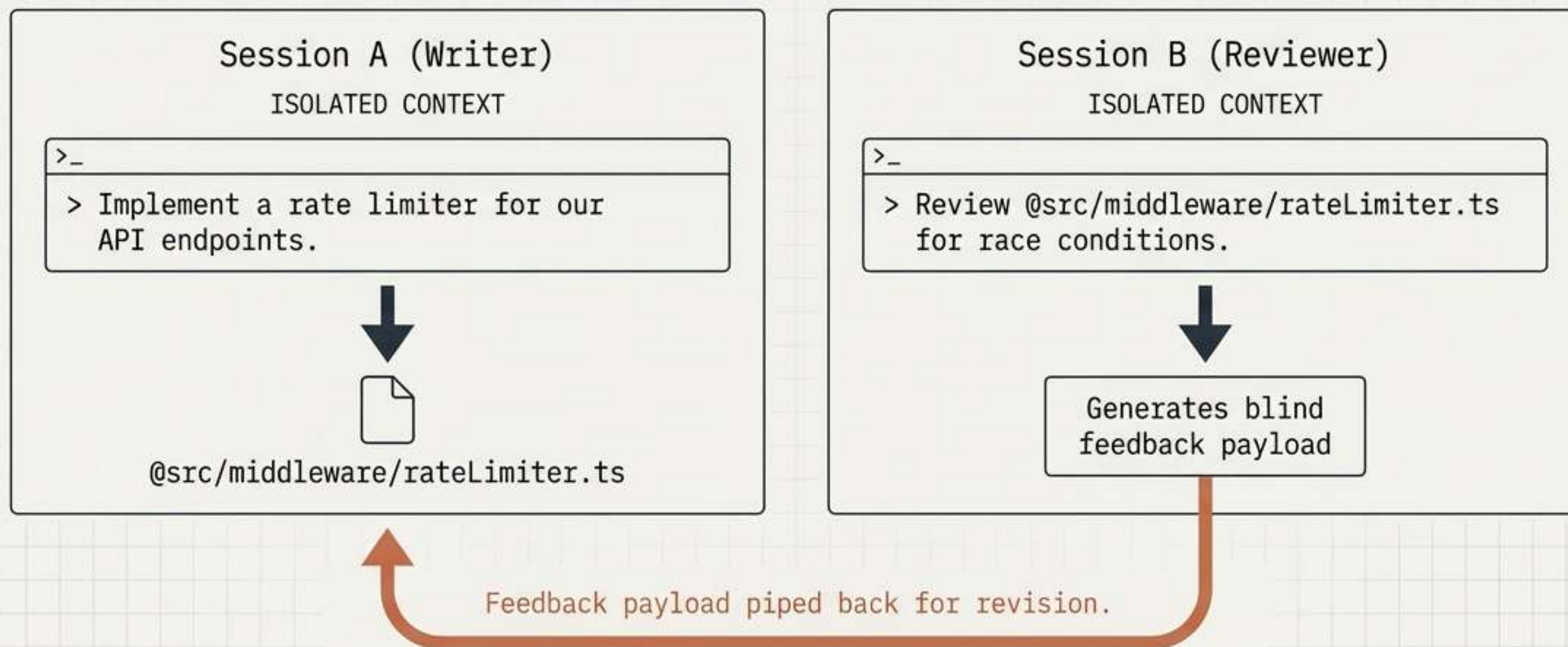
Pro-Tip: Use `/rename` to give sessions descriptive names. Workstreams should have separate, persistent contexts.

Fleet Management: Parallel Architectures



Multiply output by running isolated Claude sessions concurrently.
Fresh context removes bias.

The Writer/Reviewer Architecture

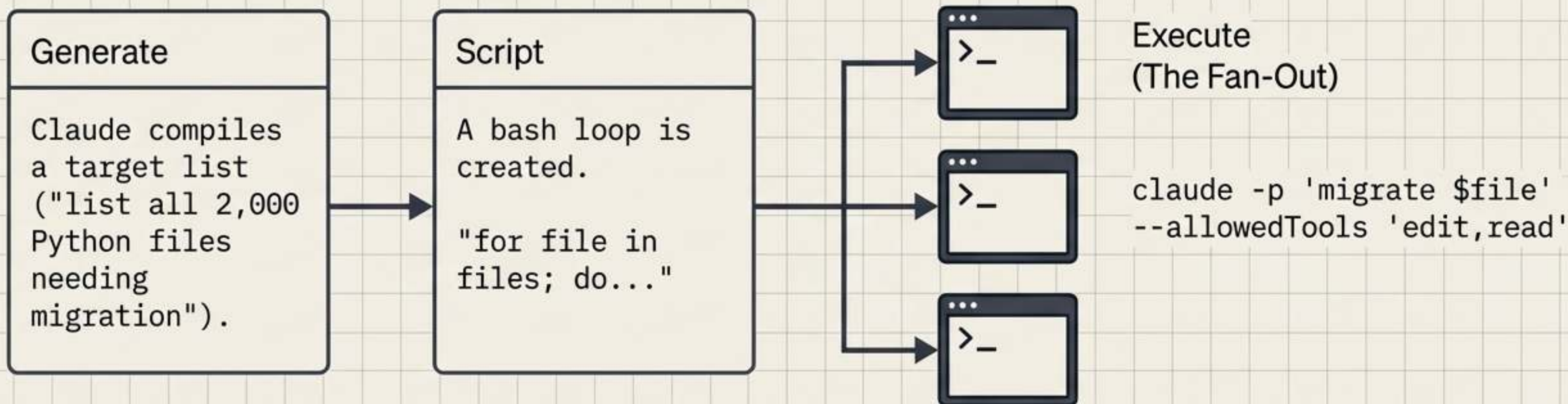




The Context Meter

Fleet Management: The Fan-Out Pipeline

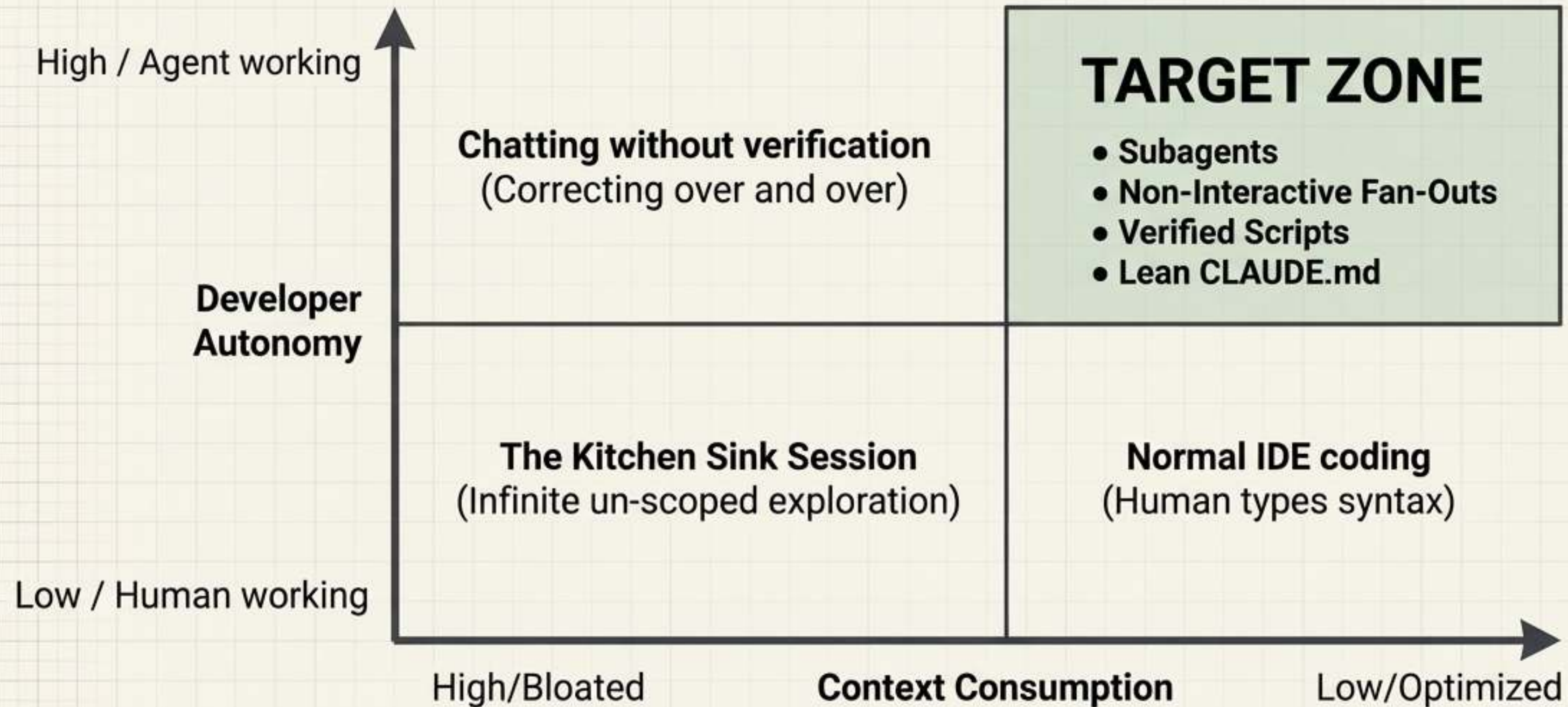
For large codebase migrations, use **Non-Interactive Mode (-p)** to **distribute** work across parallel invocations.



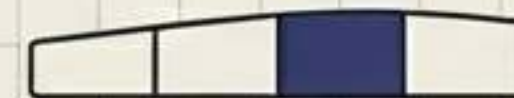
Use `--allowedTools` to tightly scope permissions for batch operations running unattended.

Synthesis: The Context Economy











The highest level of orchestration achieves maximum agentic autonomy while consuming the least amount of primary context.



The Operator's Diagnostic Checklist



The Context Meter

Symptom	Fix
 Context full of irrelevant information.	 Run <code>/clear</code> between unrelated tasks (Avoid "Kitchen Sink" sessions).
 Claude keeps repeating the same mistake.	 <code>/rewind</code> or <code>/clear</code> + new prompt. Stop polluting context with failed approaches.
 Claude ignores global project rules.	 Prune <code>CLAUDE.md</code> . Move specific workflows to <code>SKILL.md</code> or Hooks.
 Code looks right but breaks edge cases.	 The Trust-Then-Verify Gap. Enforce test-running or visual screenshots in the initial prompt.
 Infinite exploration consumes all tokens.	 Shield the main timeline by prompting: "Use subagents to investigate."

Over time, intuition replaces the manual. Observe context. Verify always. Clear often.