

The Claude Code Blueprint

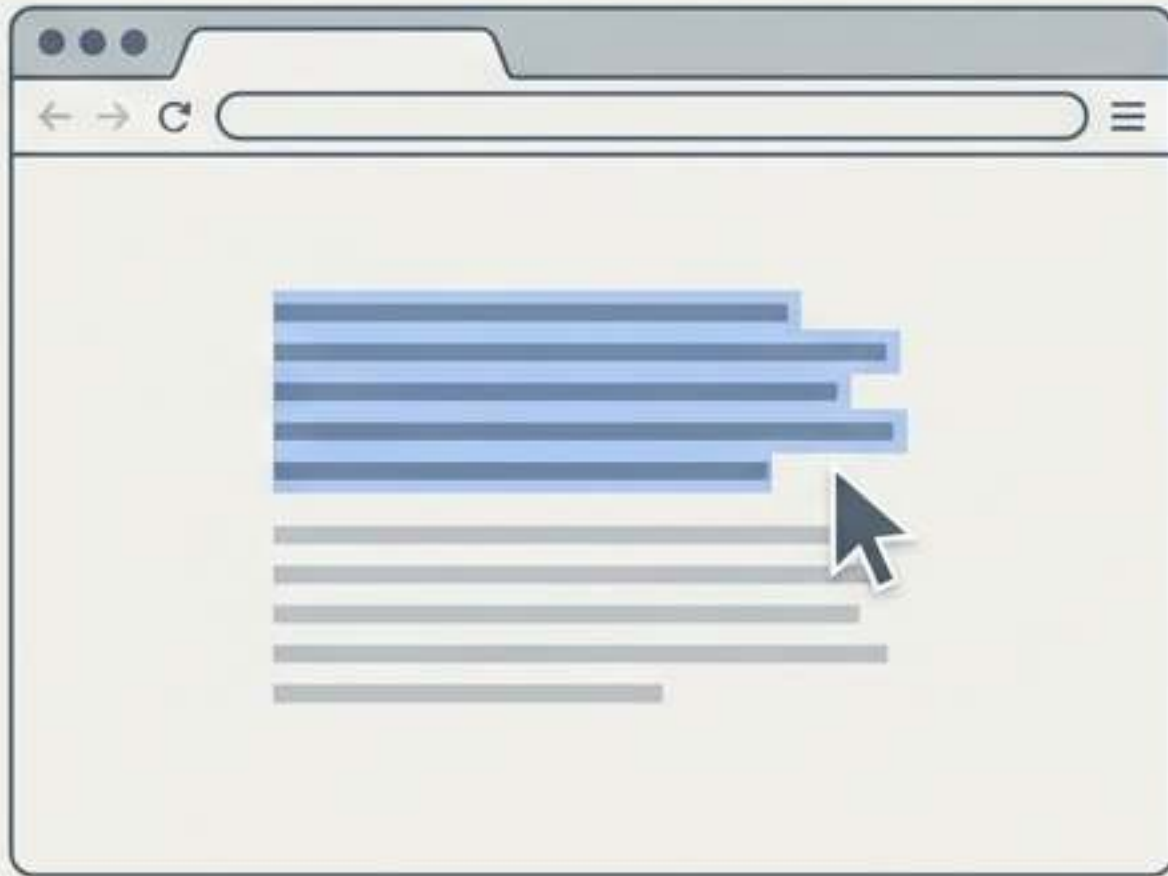
12 Features Every Engineer Should Know

Moving from AI assistants to autonomous coding agents.

```
> claude start █
```

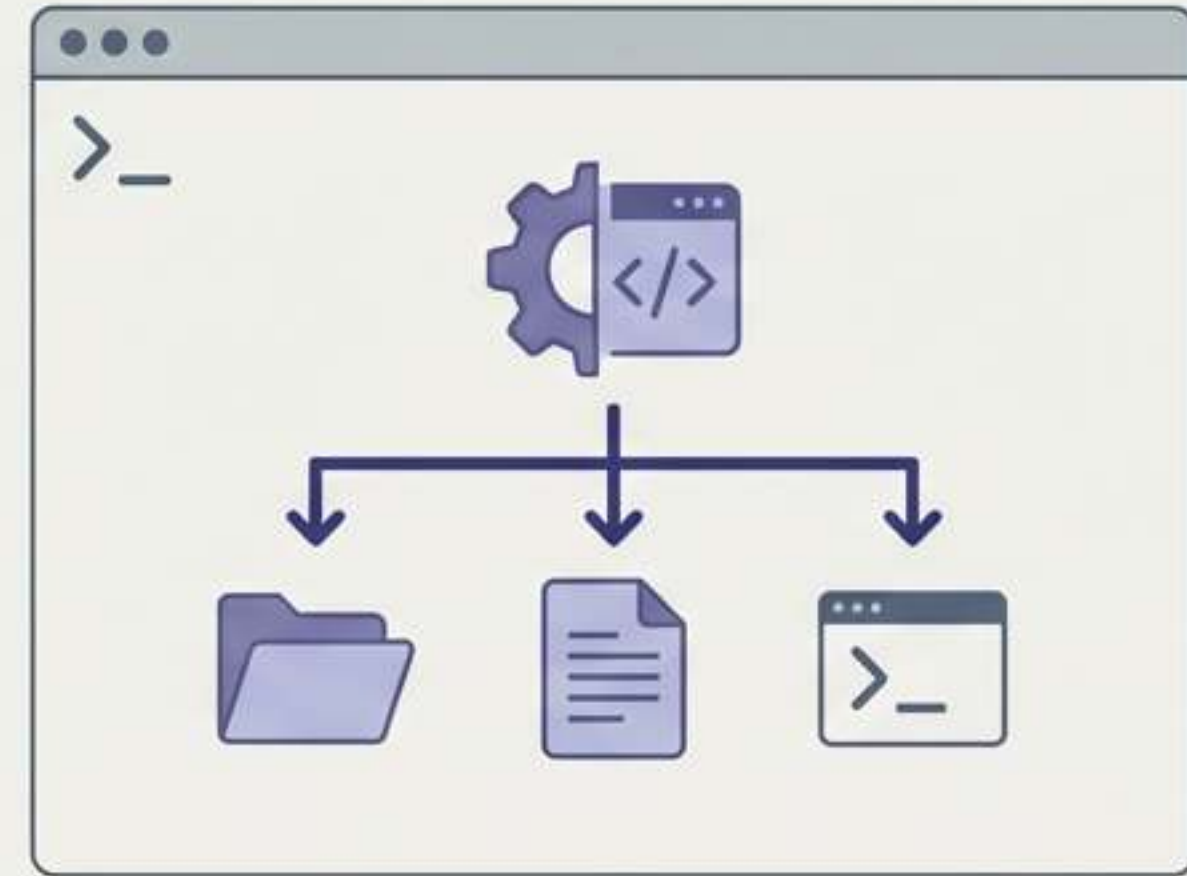
From Chatbots to Autonomous Agents

Traditional Chat



- Sandboxed in a browser.
- Relies on human copy-paste.

Claude Code

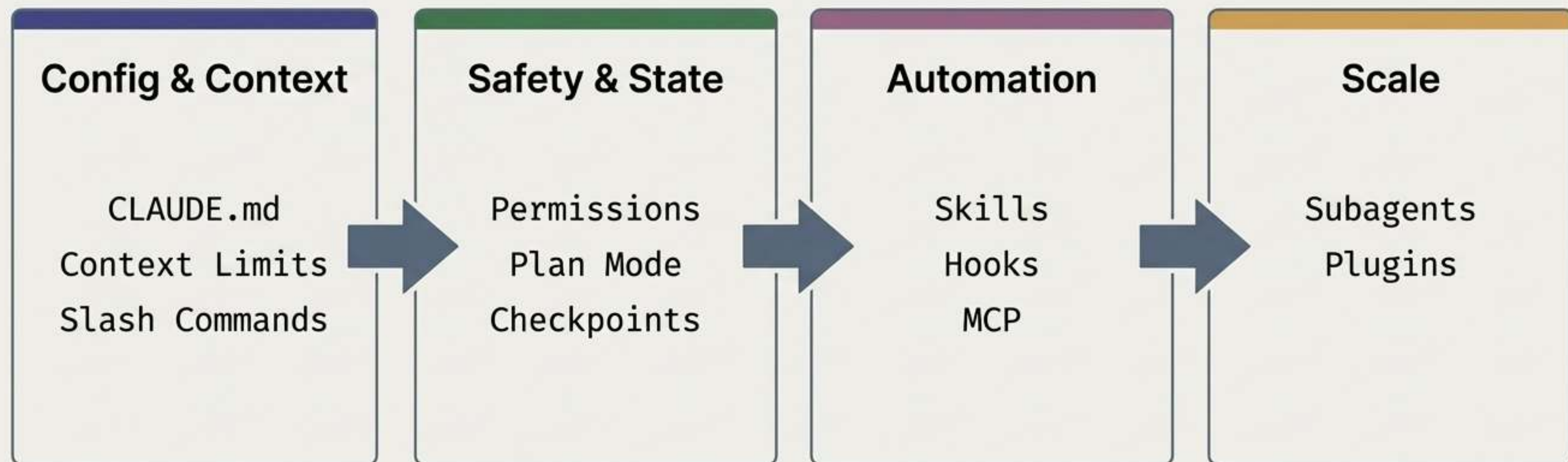


- Lives directly in your CLI.
- Reads local project files and runs bash commands.

Takeaway: This structural shift creates massive leverage but requires a new mental model for state, safety, and context.

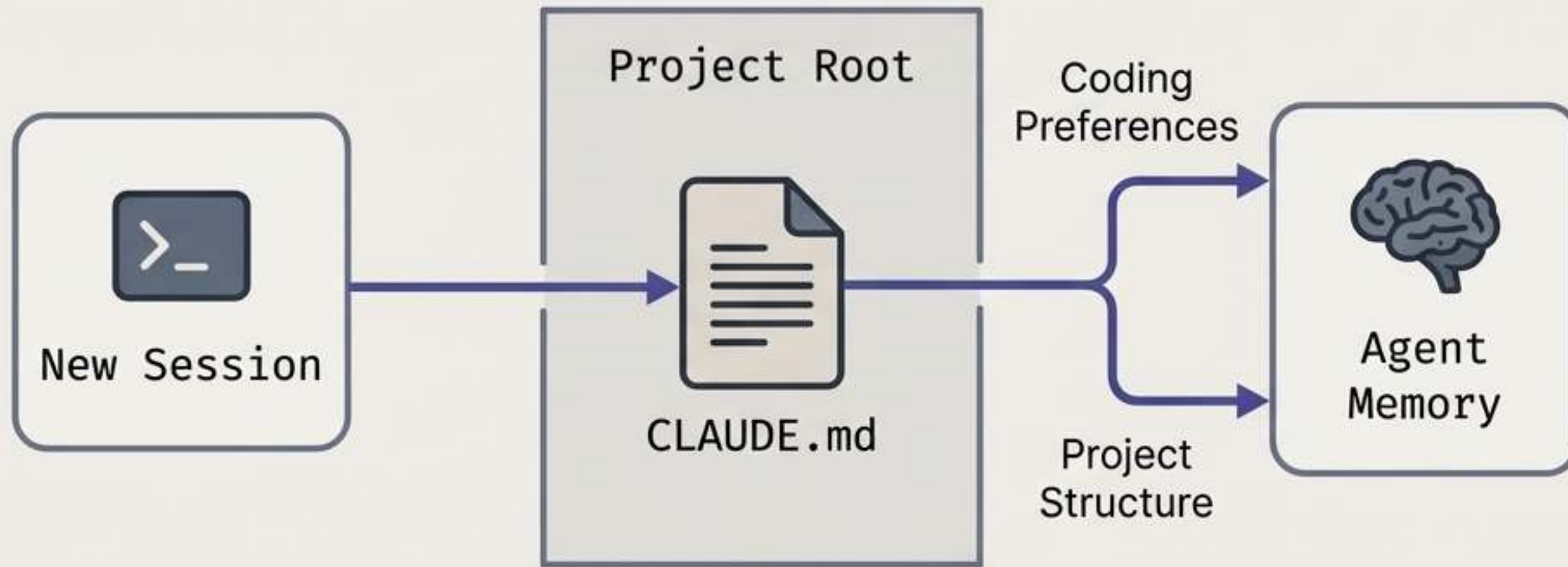
The Developer Maturity Model

Adopting autonomous agents is a progression. You don't configure everything on day one. You start by establishing context, move to securing your local state, automate repetitive loops, and finally scale the architecture to your team.



CLAUDE.md: Curing AI Amnesia

Every time you start a session, Claude knows nothing about your project. CLAUDE.md fixes this.



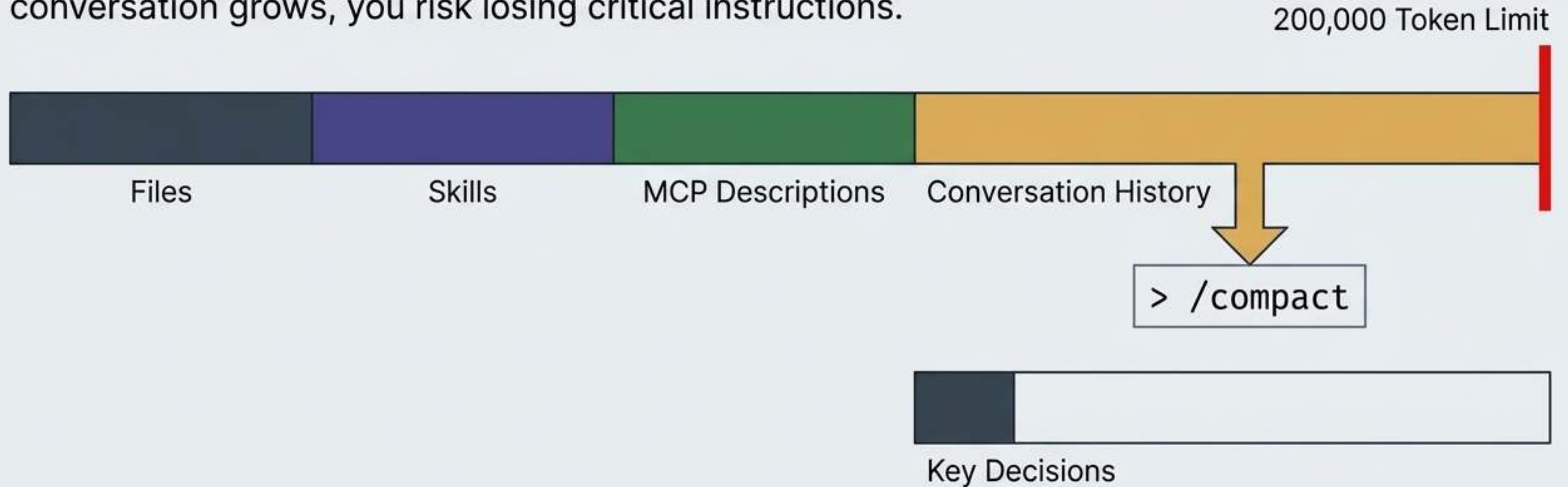
Format: Short, human-readable markdown file.

Function: Read at the start of every session to inherit rules.

Creation: Type `/init` to auto-generate.

Defending the 200k Token Window

Claude Code works within a fixed context window. As the conversation grows, you risk losing critical instructions.



> /context : See exactly what is consuming your token limit.

> /compact : Compress the history into key decisions, freeing up space automatically.

Balancing Speed vs. Safety

By default, Claude asks for approval before editing files or running commands. The interactive permissions menu lets you customize this behavior.

/permissions		
Run Tests	Pre-approved	<input checked="" type="checkbox"/>
Commit Code	Pre-approved	<input checked="" type="checkbox"/>
Delete Files	Blocked / Requires Prompt	<input type="checkbox"/>

Start conservative. Gradually open up your allow-lists as you build trust with specific agent workflows.

Plan Mode: Think Before You Edit

Coding right away is tempting, but on complex tasks, Claude can edit the wrong files and waste tokens. Plan Mode separates planning from doing.

Plan Mode

Shift + Tab



Status: Read-Only Tools Allowed

Action: Propose step-by-step plan. Ask questions and read files.

Normal Mode



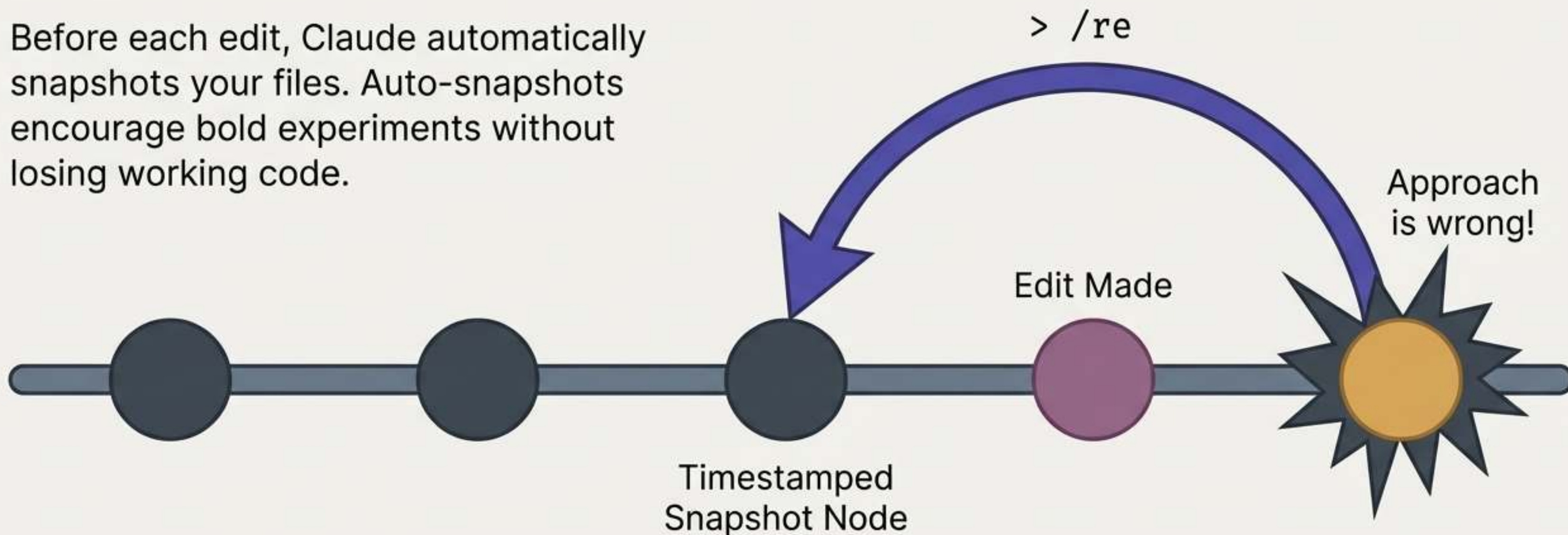
Status: Read/Write Execution

Action: Edit files and run commands.

Once you review and approve the read-only plan, switch back to normal mode to execute.

/re: The Ultimate Undo Button

Before each edit, Claude automatically snapshots your files. Auto-snapshots encourage bold experiments without losing working code.



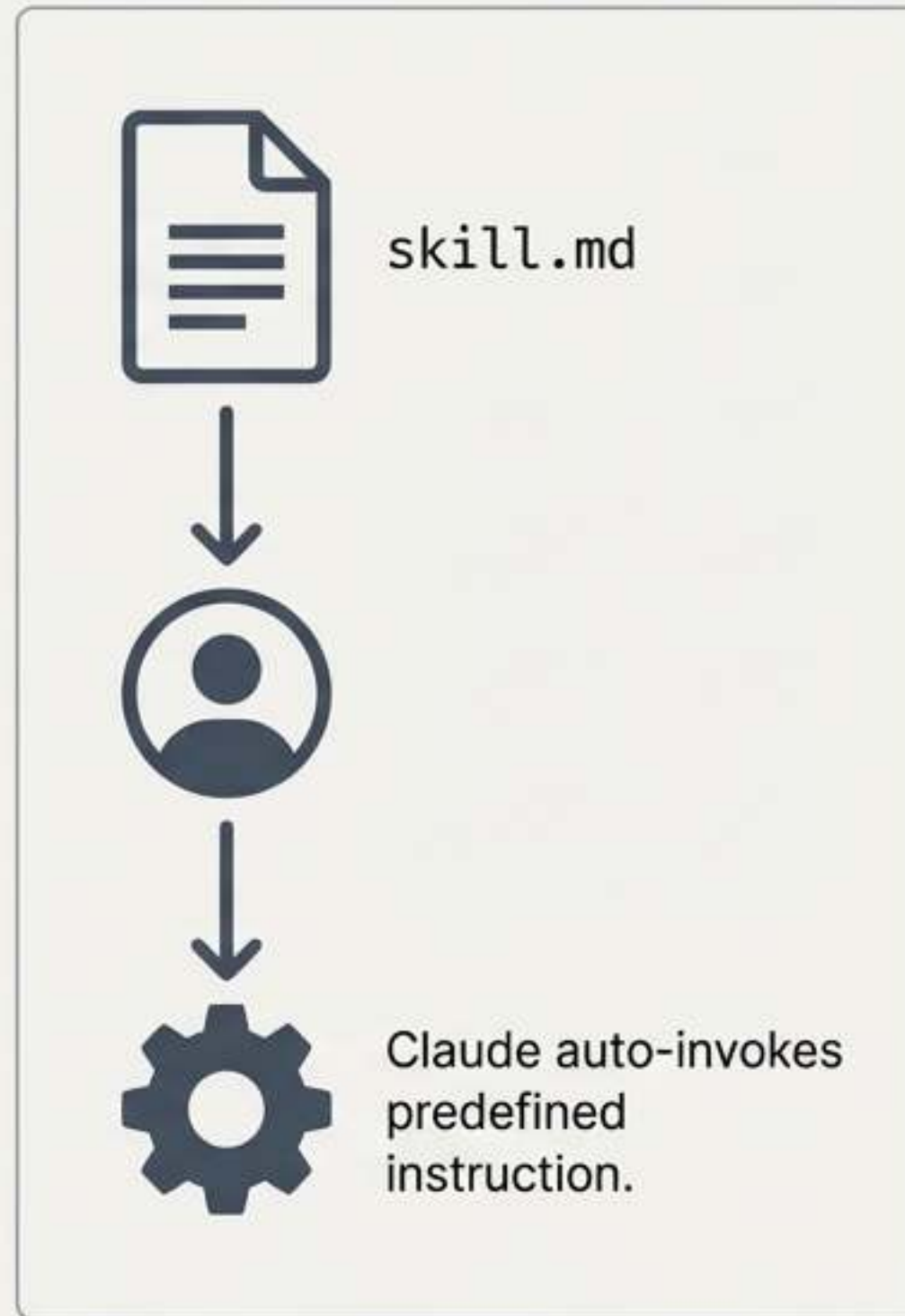
Step 1 Type `/re`

Step 2 View checkpoints with timestamps

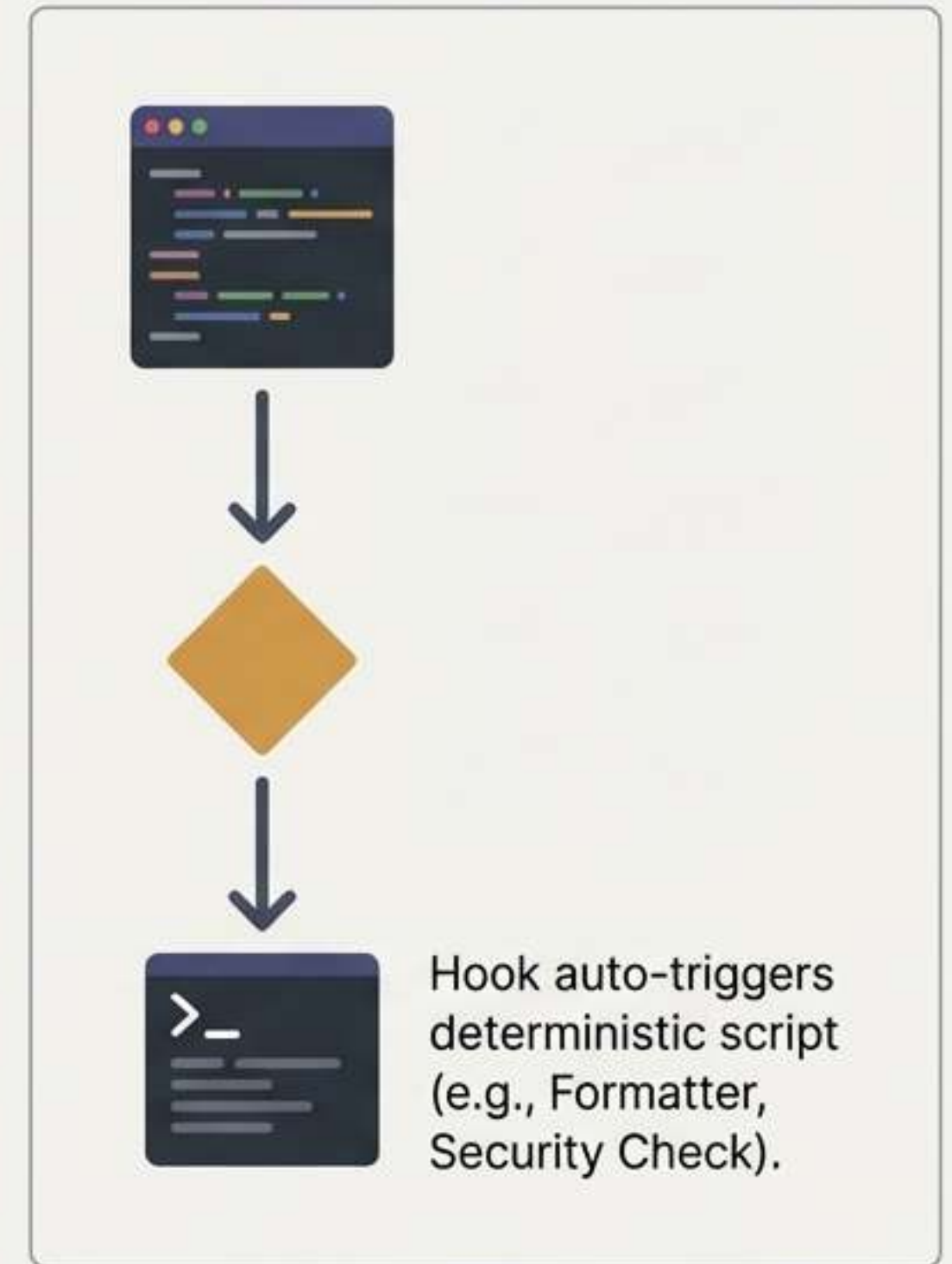
Step 3 Instantly restore exact state

Automating the Repetitive

Stop typing detailed prompts repeatedly. Automate common workflows using on-demand Skills or automatic Hooks.



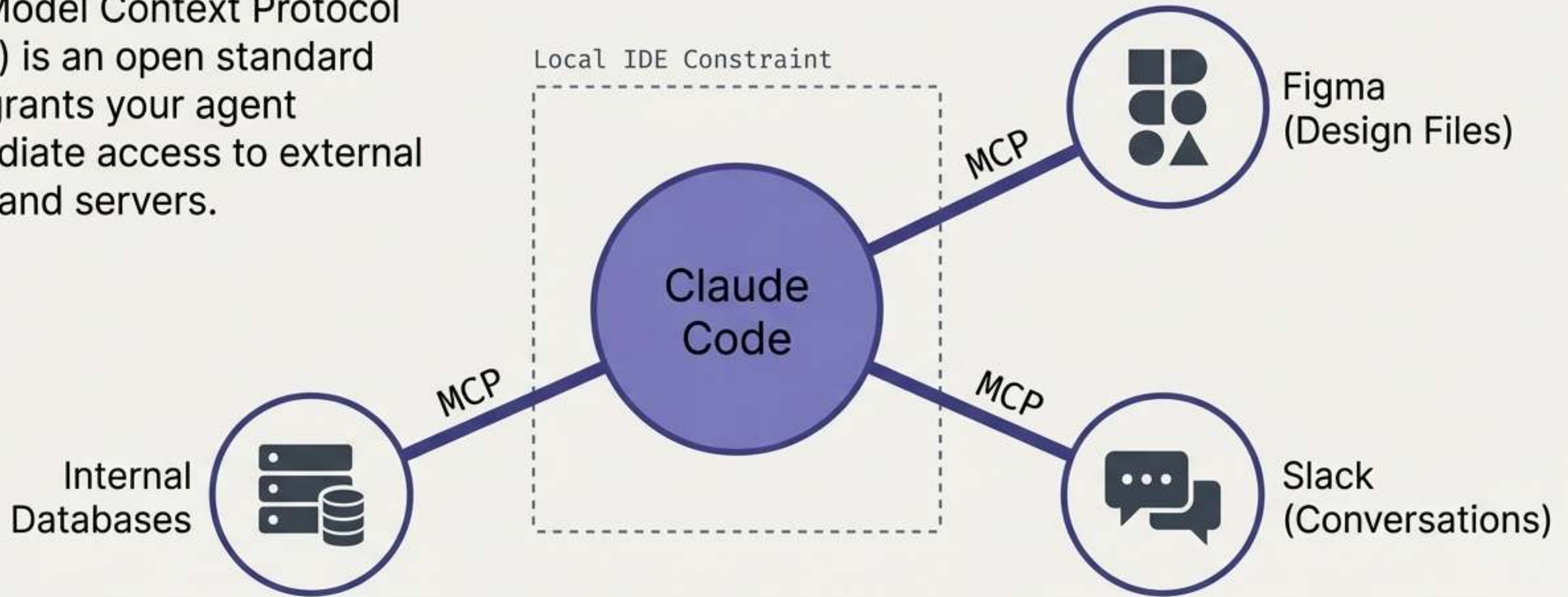
Skills: Turn workflows into repeatable, shareable automation.



Hooks: Scripts that run automatically at specific points in the workflow.

Breaking Out of the IDE with MCP

The Model Context Protocol (MCP) is an open standard that grants your agent immediate access to external tools and servers.



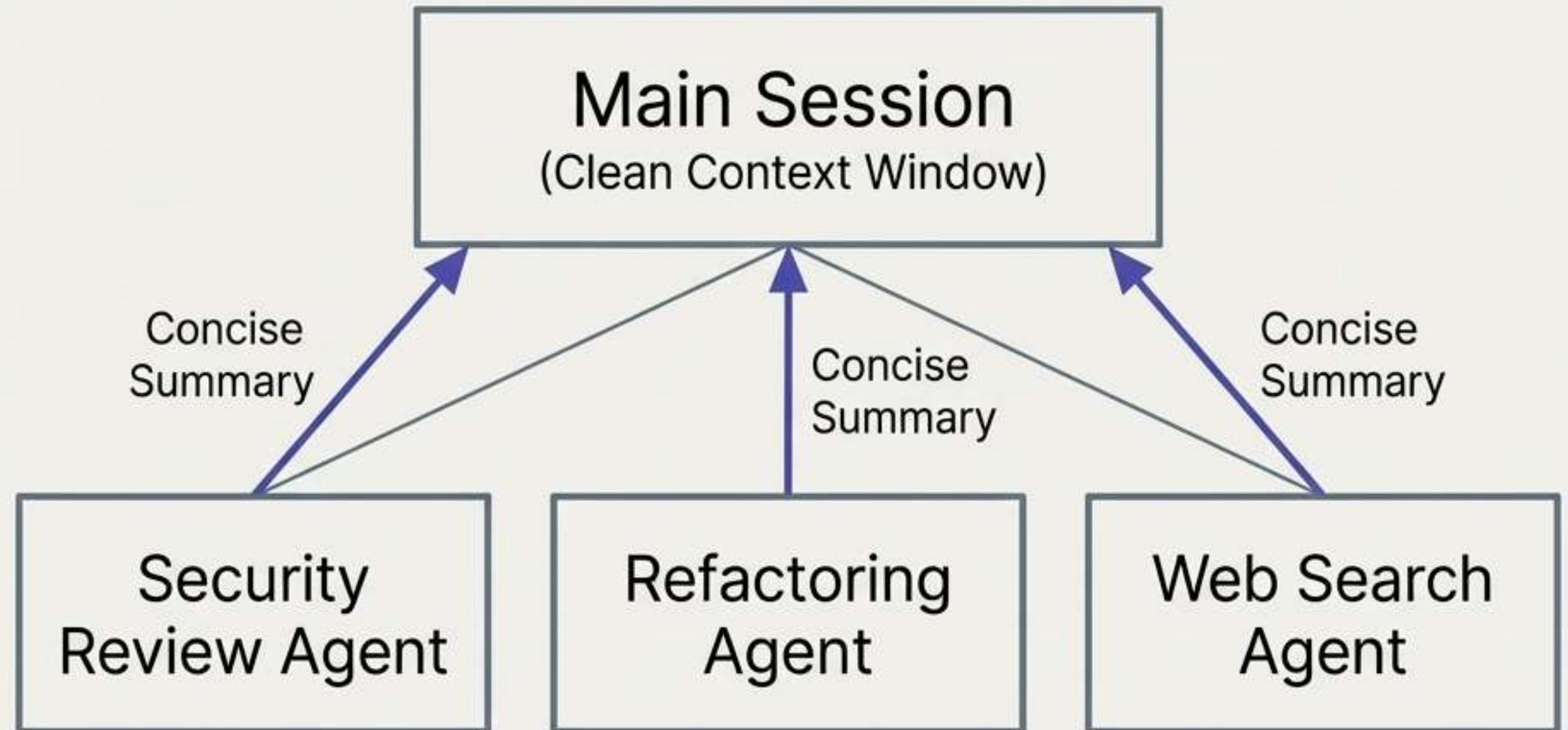
Connect your tools, expose the data, and Claude integrates it directly into your CLI session.

Which Automation Tool Should I Use?

Feature	Trigger Mechanism	Primary Use Case & Example
Skills (<code>skill.md</code>)	On-Demand (Prompt-driven)	Shareable workflows. Example: Run my standard deployment checklist.
Hooks	Event-Driven (Before/After tool runs)	Guaranteed execution. Example: Run code formatter after every file edit.
MCP	External Data Access	Bridging walled gardens. Example: Read the latest Figma spec or Slack thread.

/agents: Dividing the Labor

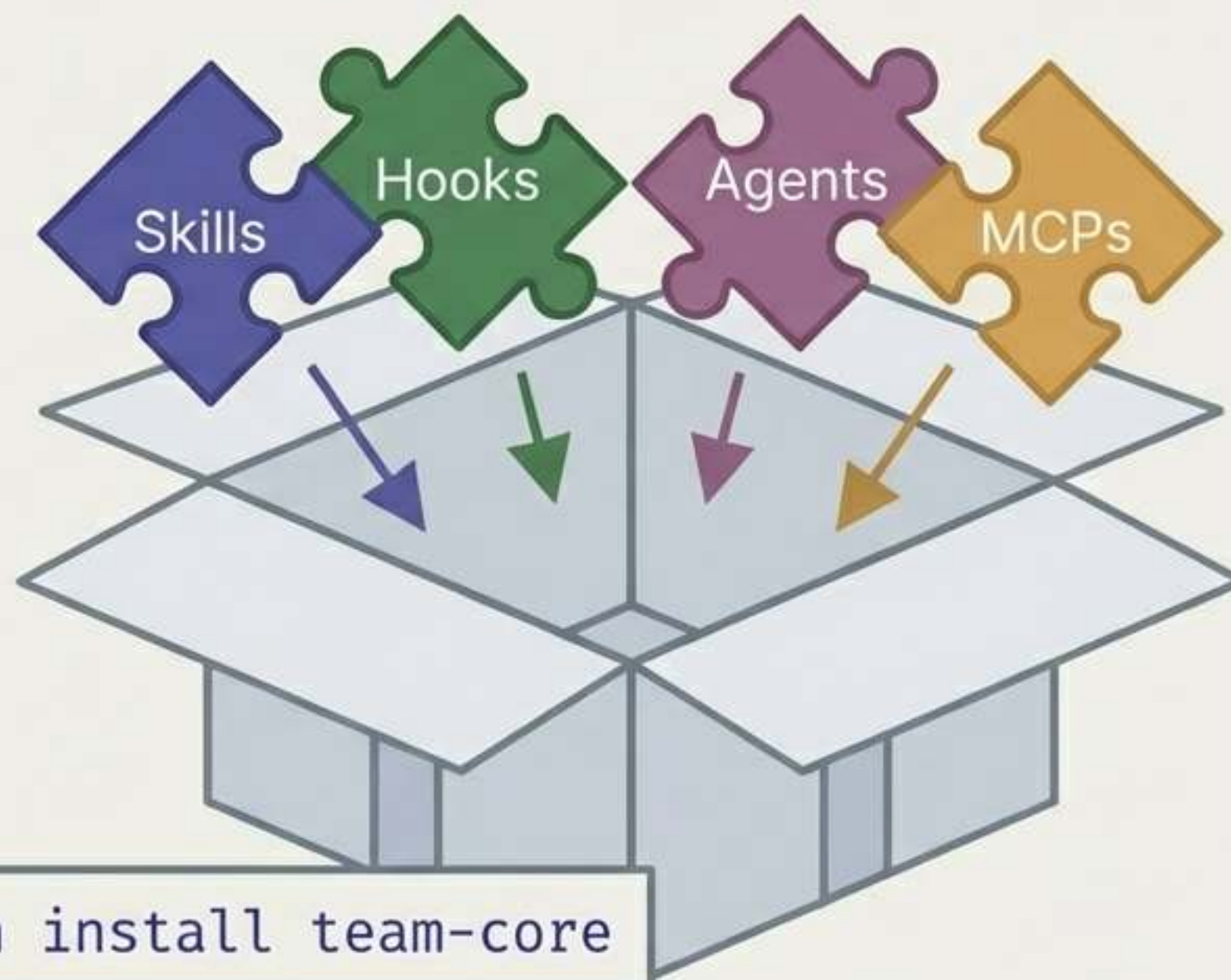
For complex tasks, doing everything in one session gets messy fast. Subagents are separate Claude sessions spun up for specific, focused jobs.



The subagents work independently and return only a short summary, summary, keeping your main context window lean and preventing messy single-session context overload.

Plugins: Packaging for the Team

You've spent hours configuring the perfect setup. Plugins bundle your **architecture into a single installable unit.**



```
> claude plugin install team-core
```

Publish to a marketplace or Git repository, and your team installs the entire custom architecture with one command instead of manual configuration.

Slash Commands: Your Keyboard Shortcuts

<code>/init</code>	Auto-generates CLAUDE.md for project memory.
<code>/permissions</code>	Opens interactive menu to manage tool access.
<code>/re</code>	Rewinds timeline to a previous file checkpoint.
<code>/context</code>	Analyzes token consumption and limits.
<code>/compact</code>	Compresses conversation history to free up space.
<code>/agents</code>	Spawns specialized subagents to divide labor.

Start Building from Day One

Claude Code brings simplicity and structure to autonomous agents. The best way to learn is to open a terminal and start building.

```
> claude █
```

AI Engineering Cohort

Reasoning LLMs

Deep Research

RAG

MCP Agents

Looking to build real systems? Join thousands of learners in our project-based AI engineering curriculum.

[Link in description](#)