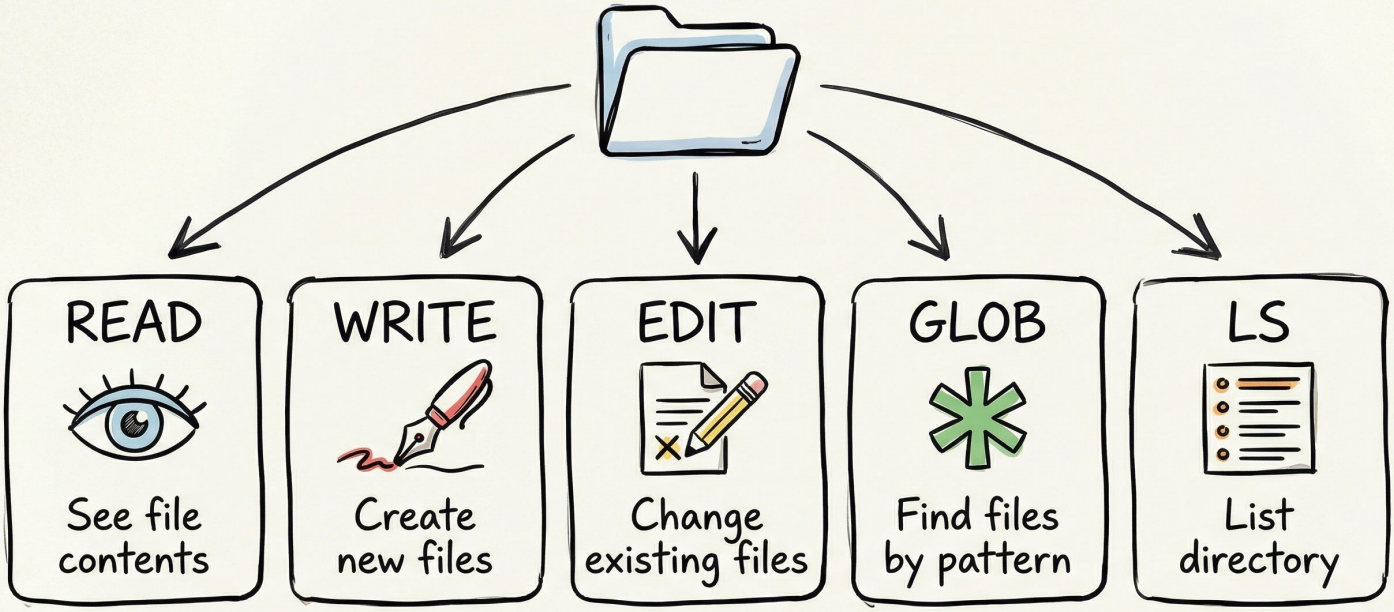
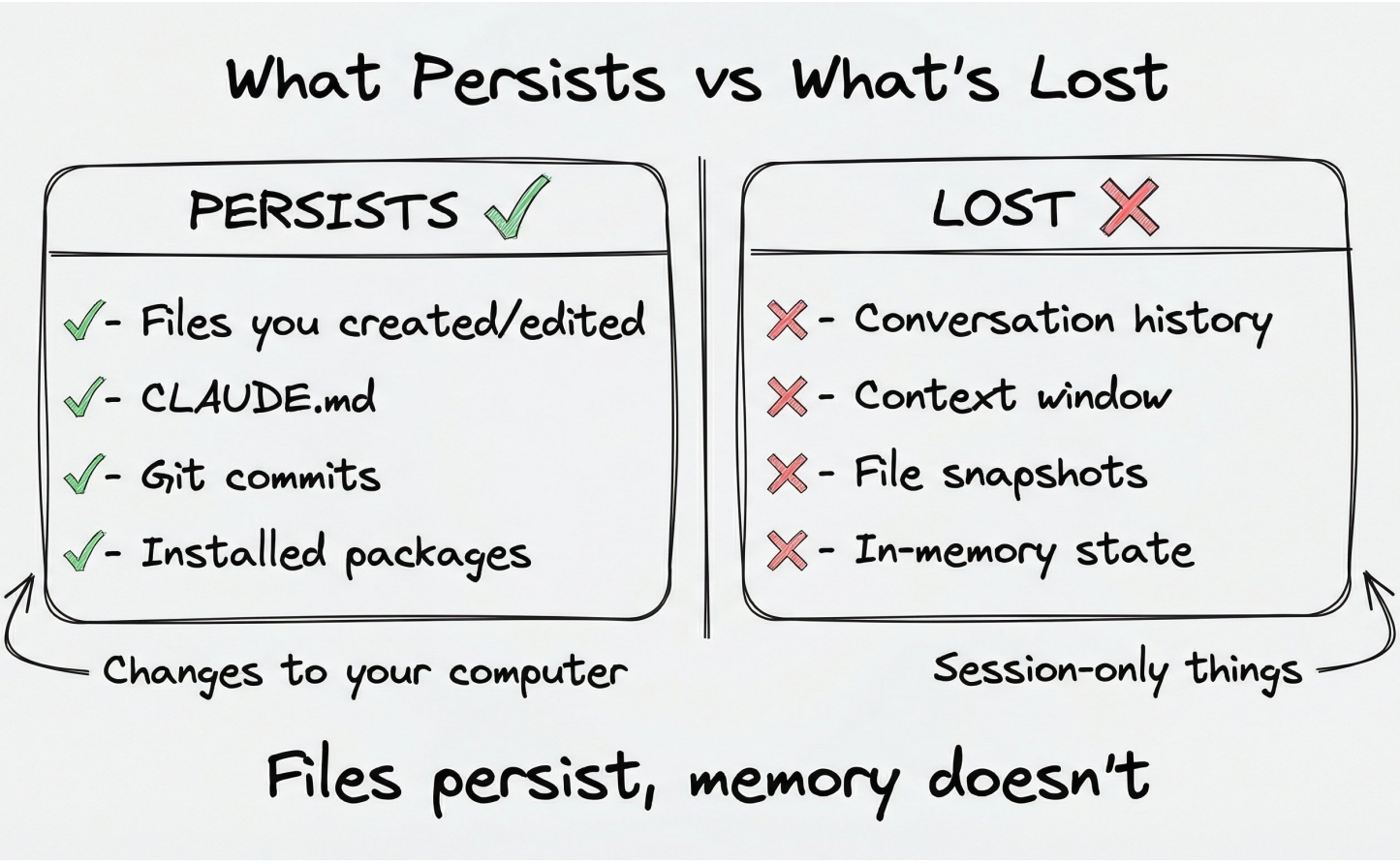
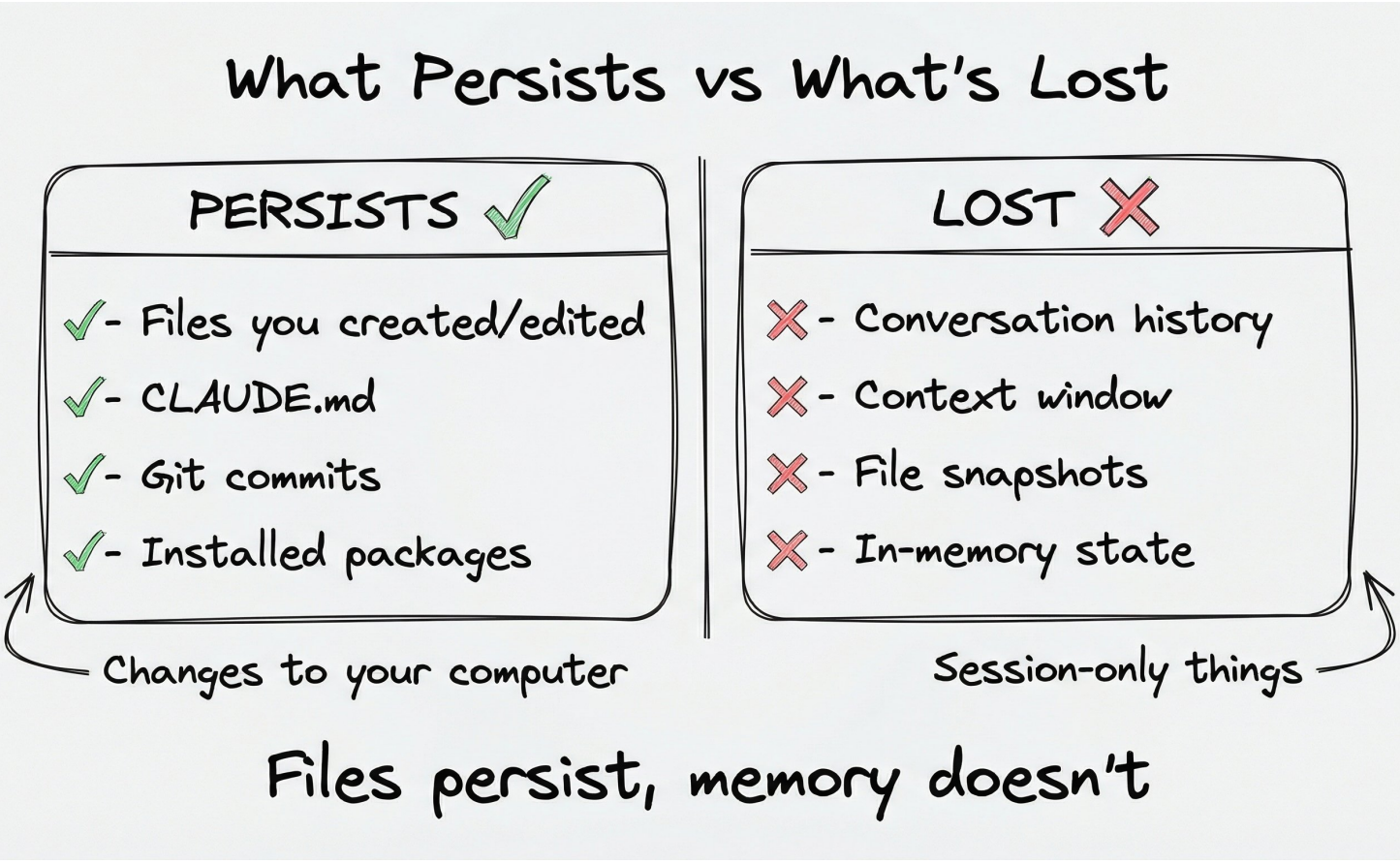
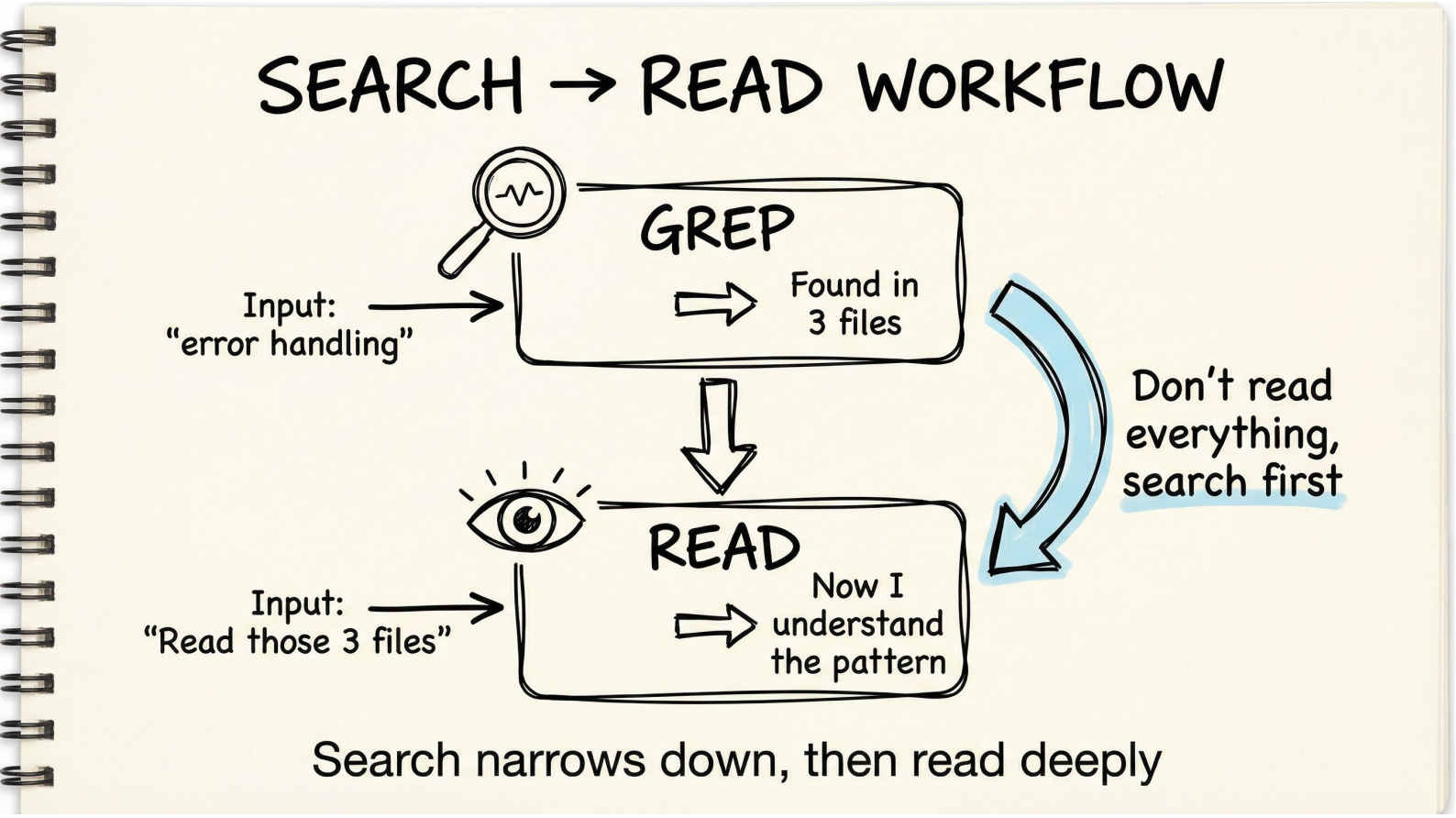
File Tools at a Glance

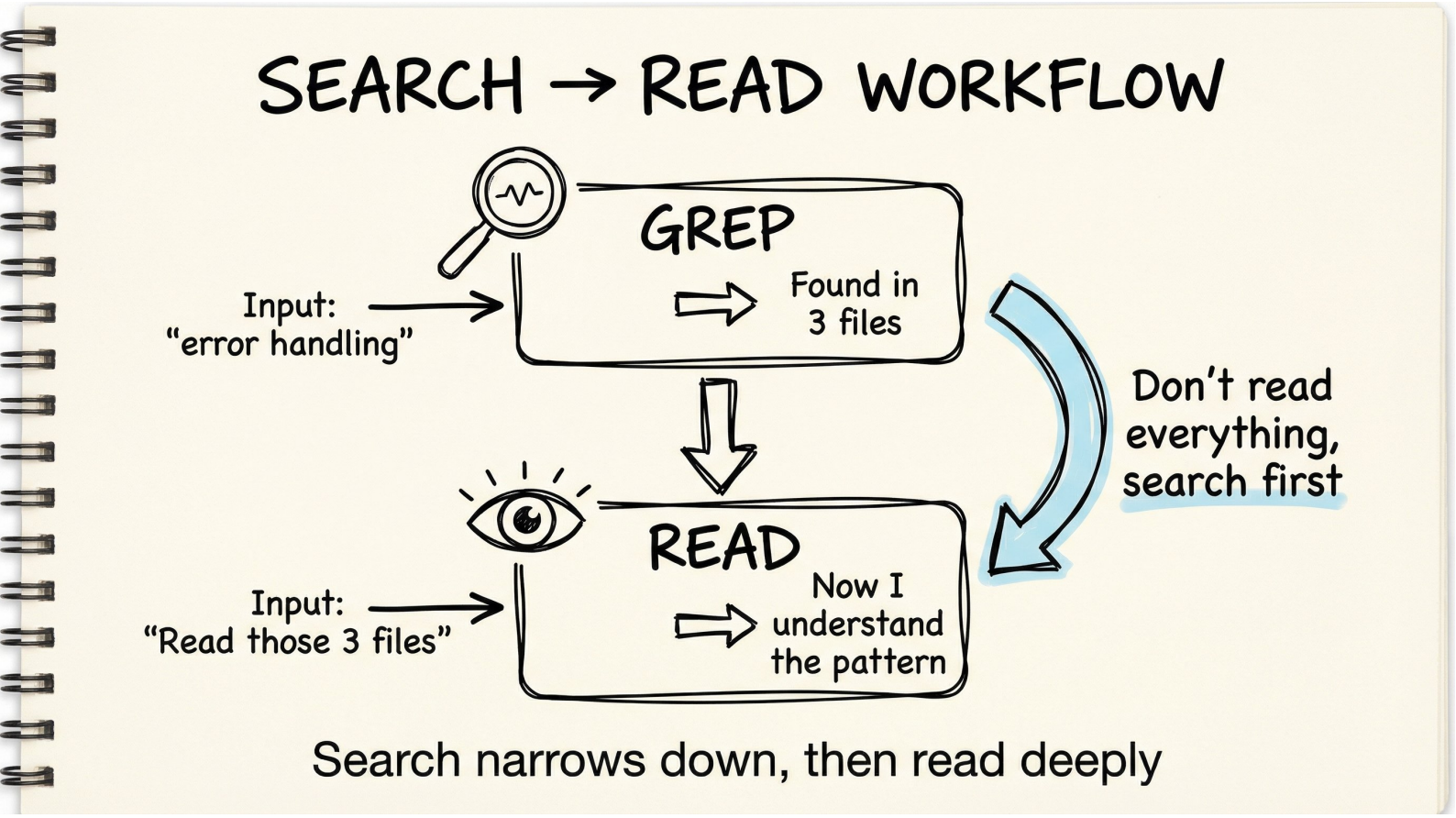


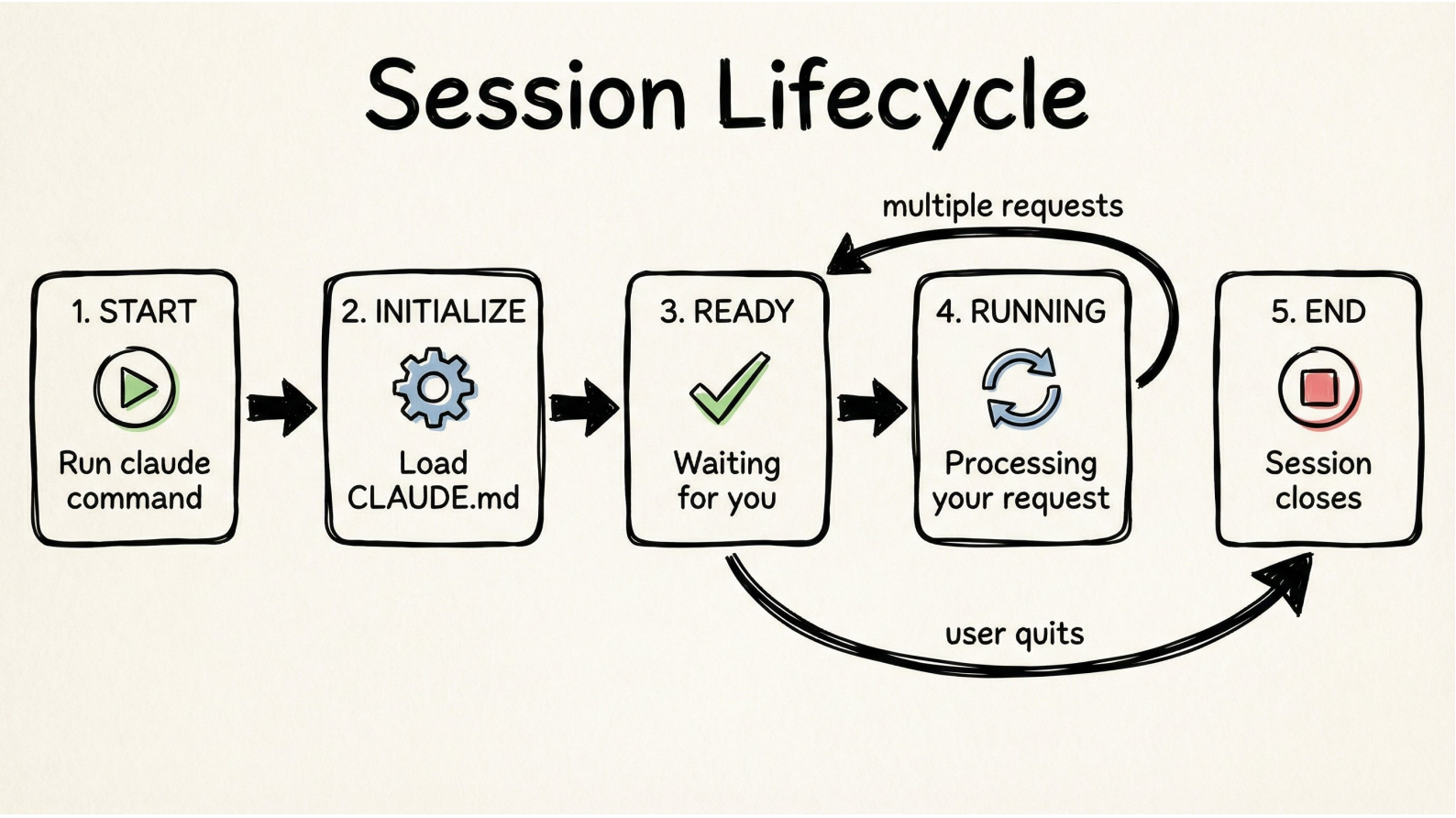
These are Claude's hands for working with your files

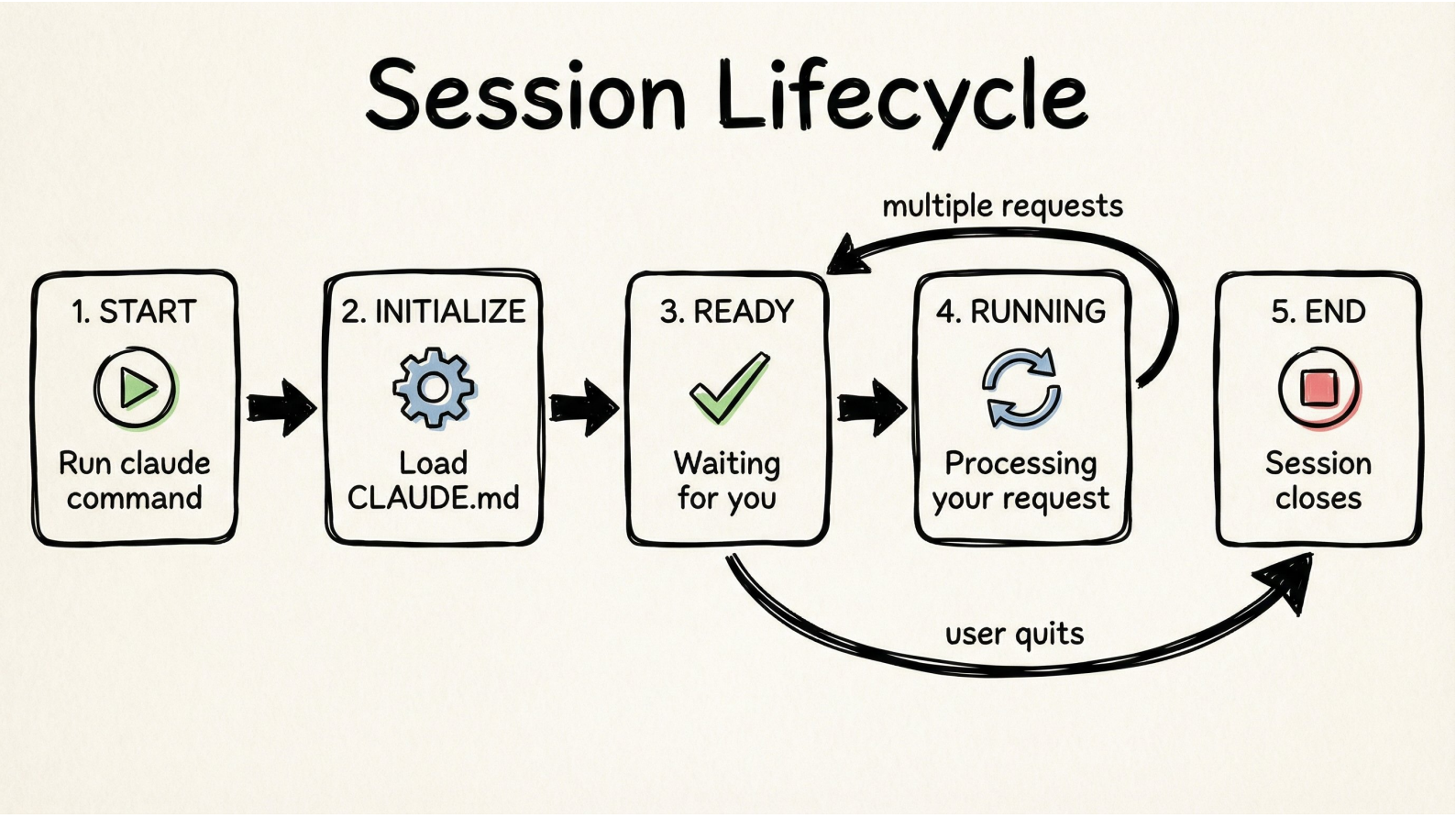

What Persists vs What's Lost
PERSISTS
✓ - Files you created/edited
✓ - CLAUDE.md
✓ - Git commits
✓ - Installed packages
LOST
✗ - Conversation history
✗ - Context window
✗ - File snapshots
✗ - In-memory state
Changes to your computer
Session-only things
Files persist, memory doesn't


What Persists vs What's Lost
PERSISTS
✓ - Files you created/edited
✓ - CLAUDE.md
✓ - Git commits
✓ - Installed packages
LOST
✗ - Conversation history
✗ - Context window
✗ - File snapshots
✗ - In-memory state
Changes to your computer
Session-only things
Files persist, memory doesn't

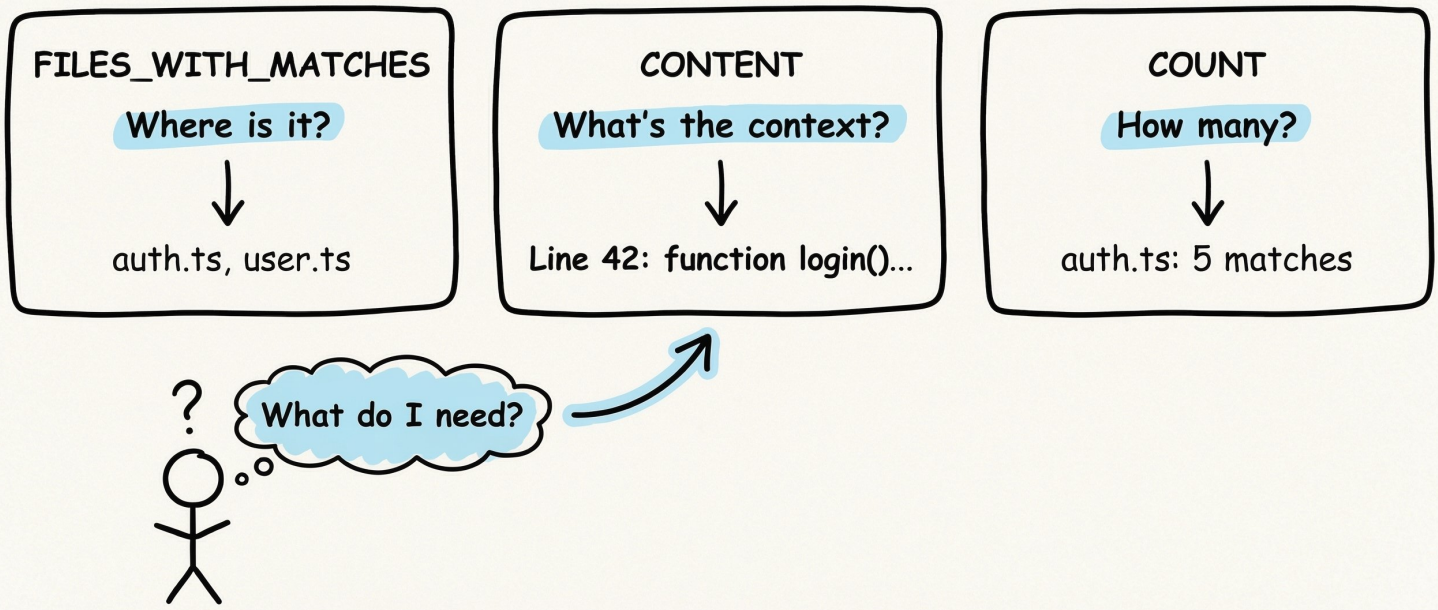

SEARCH → READ WORKFLOW
GREP
Input:
"error handling"
Found in
3 files
Don't read
everything,
search first
READ
Input:
"Read those 3 files"
Now I
understand
the pattern
Search narrows down, then read deeply


SEARCH → READ WORKFLOW
GREP
Input:
"error handling"
Found in
3 files
Don't read
everything,
search first
READ
Input:
"Read those 3 files"
Now I
understand
the pattern
Search narrows down, then read deeply

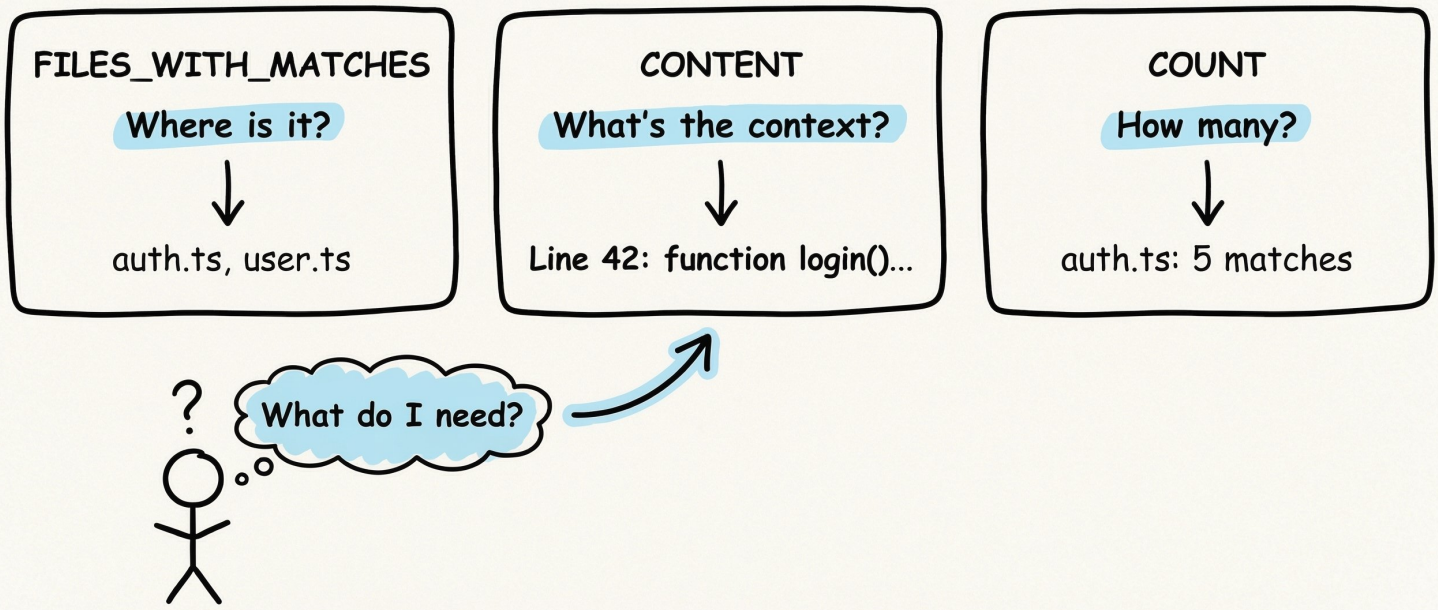

Session Lifecycle
multiple requests
1. START
Run claude command
2. INITIALIZE
Load CLAUDE.md
3. READY
Waiting for you
4. RUNNING
Processing your request
5. END
Session closes
user quits

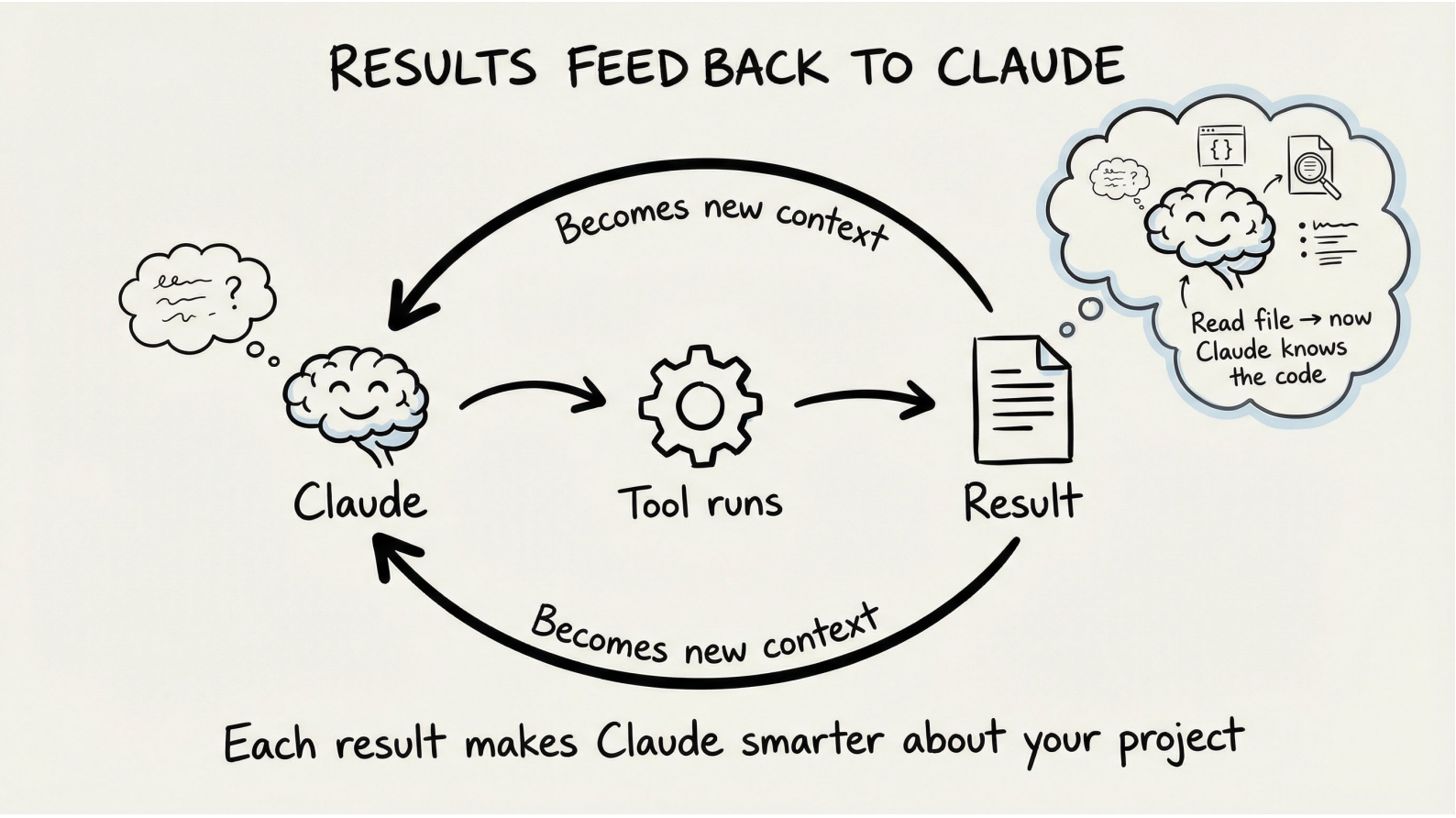

Session Lifecycle
multiple requests
1. START
Run claude command
2. INITIALIZE
Load CLAUDE.md
3. READY
Waiting for you
4. RUNNING
Processing your request
5. END
Session closes
user quits

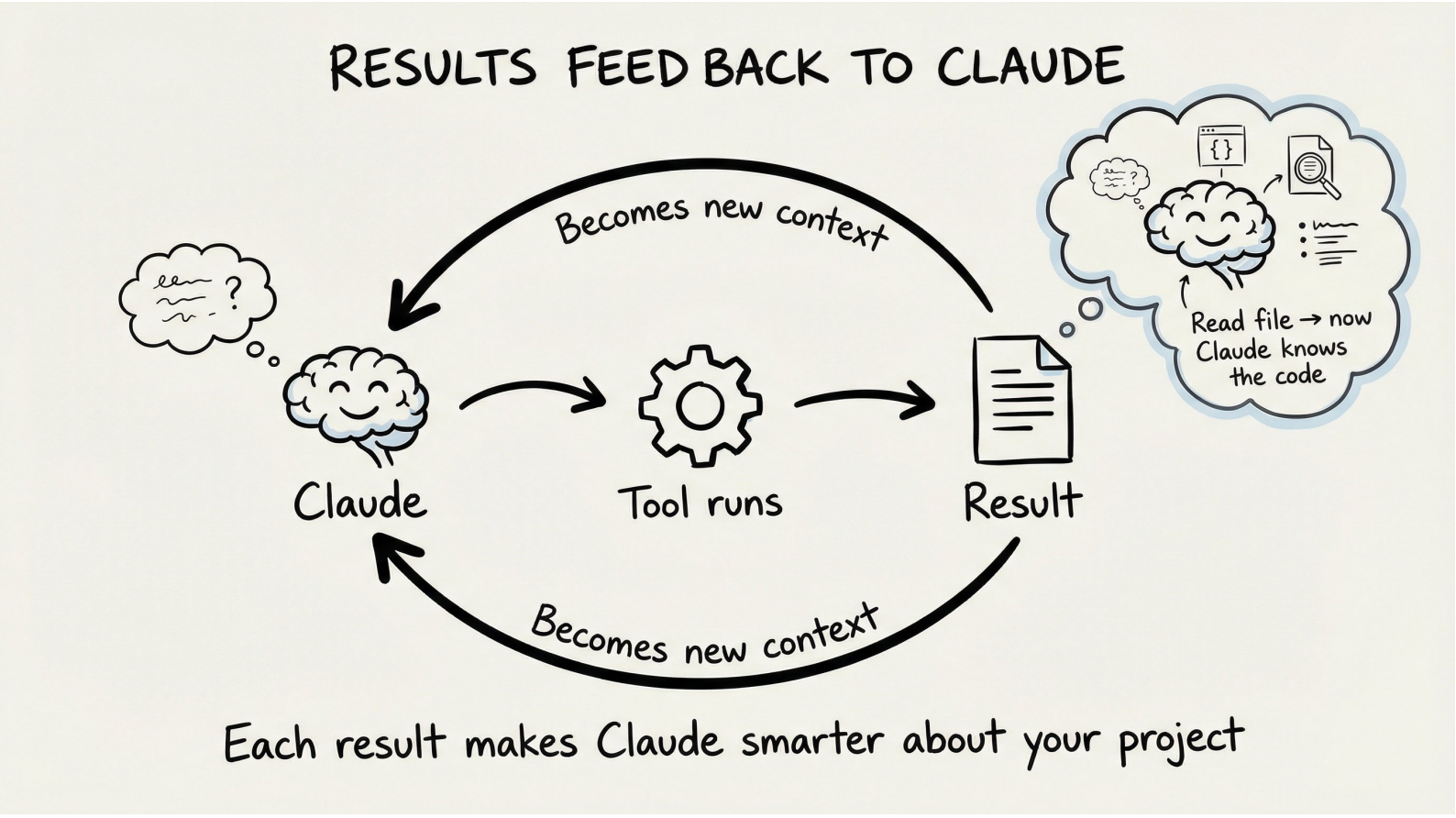
Three Search Modes



Three Search Modes

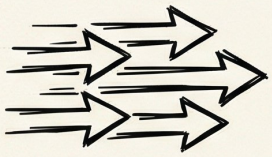



RESULTS FEED BACK TO CLAUDE
Becomes new context
Claude
Tool runs
Result
Read file → now Claude knows the code
Becomes new context
Each result makes Claude smarter about your project


RESULTS FEED BACK TO CLAUDE
Becomes new context
Claude
Tool runs
Result
Read file → now Claude knows the code
Becomes new context
Each result makes Claude smarter about your project

WHY RIPGREP IS FAST

PARALLEL



Searches many files at once

SMART



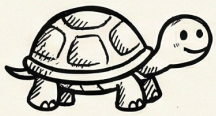
Skips node_modules, .git, etc.

EFFICIENT



Memory-mapped I/O

Regular search



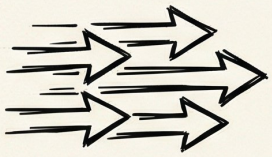
Ripgrep



This is why code search feels instant

WHY RIPGREP IS FAST

PARALLEL



Searches many files at once

SMART



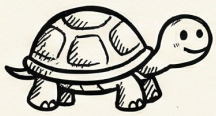
Skips node_modules, .git, etc.

EFFICIENT



Memory-mapped I/O

Regular search



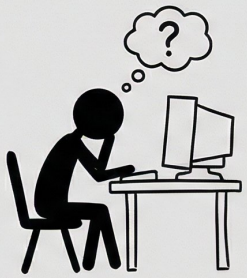
Ripgrep



This is why code search feels instant

Traditional Dev vs Claude Code

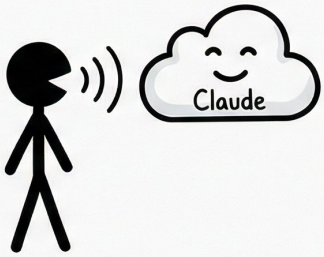
~~OLD WAY~~



- You read docs
- You write code
- You debug
- You test

TRANSFORMATION

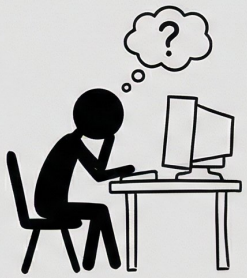
NEW WAY



- You describe what you want
- Claude reads & understands
- Claude writes code
- Claude tests & fixes

Traditional Dev vs Claude Code

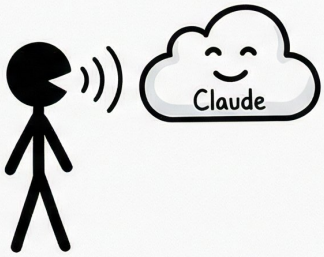
~~OLD WAY~~



- You read docs
- You write code
- You debug
- You test

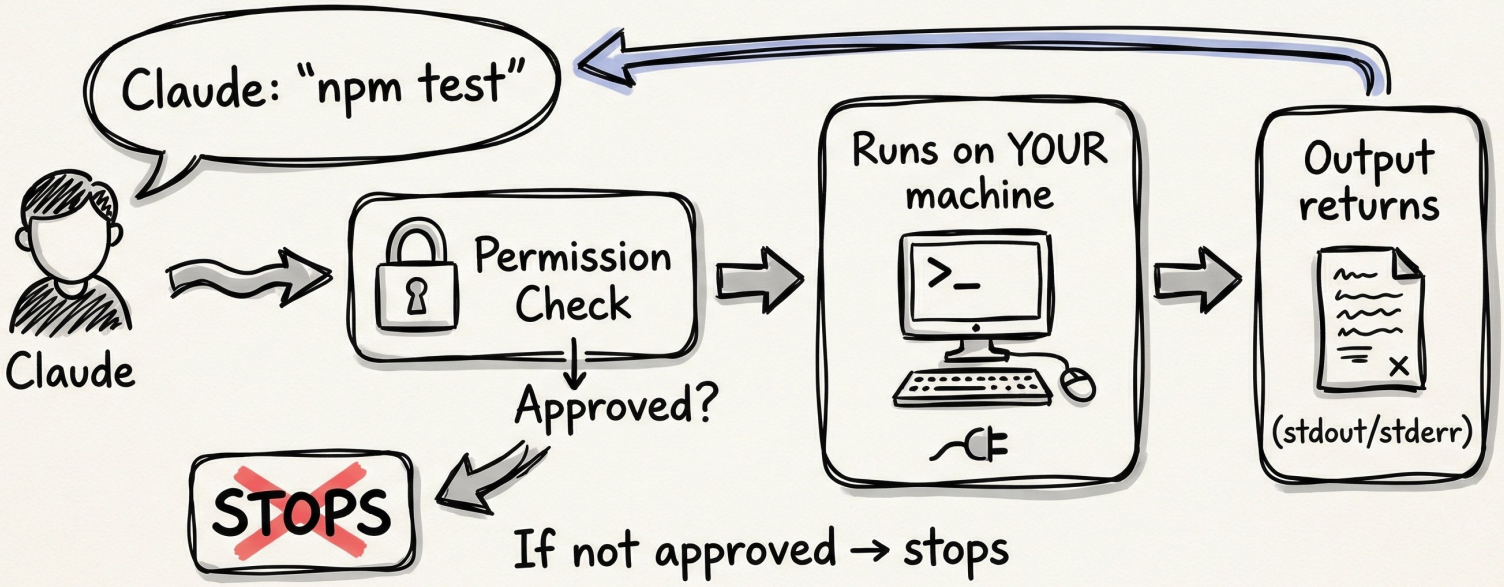
TRANSFORMATION

NEW WAY



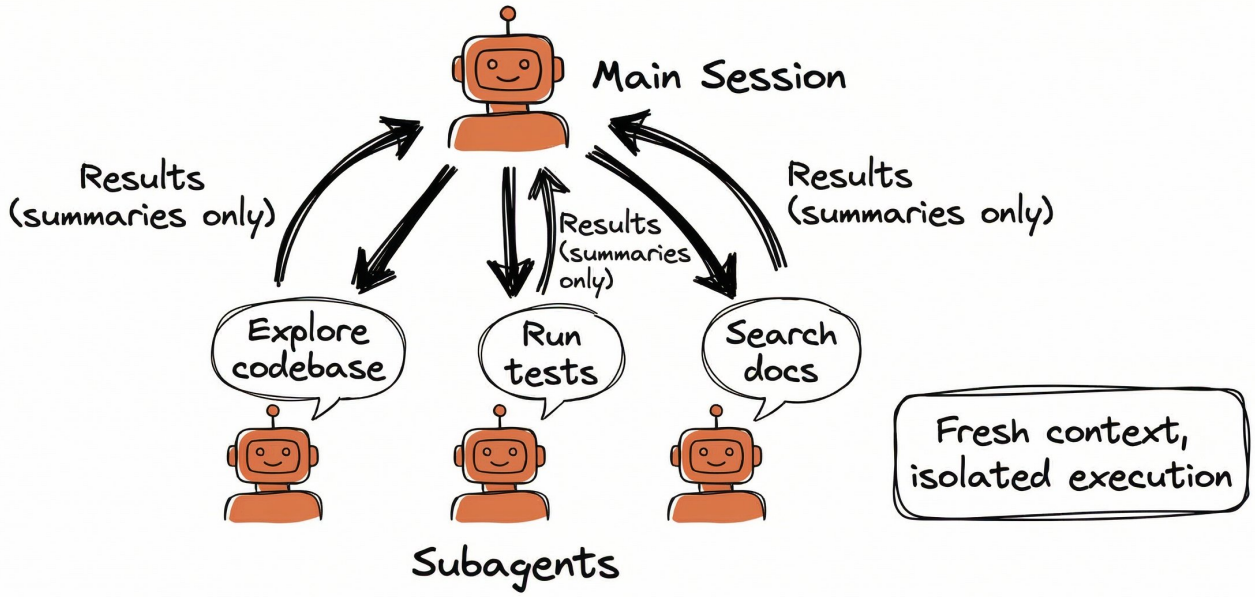
- You describe what you want
- Claude reads & understands
- Claude writes code
- Claude tests & fixes

COMMAND EXECUTION FLOW



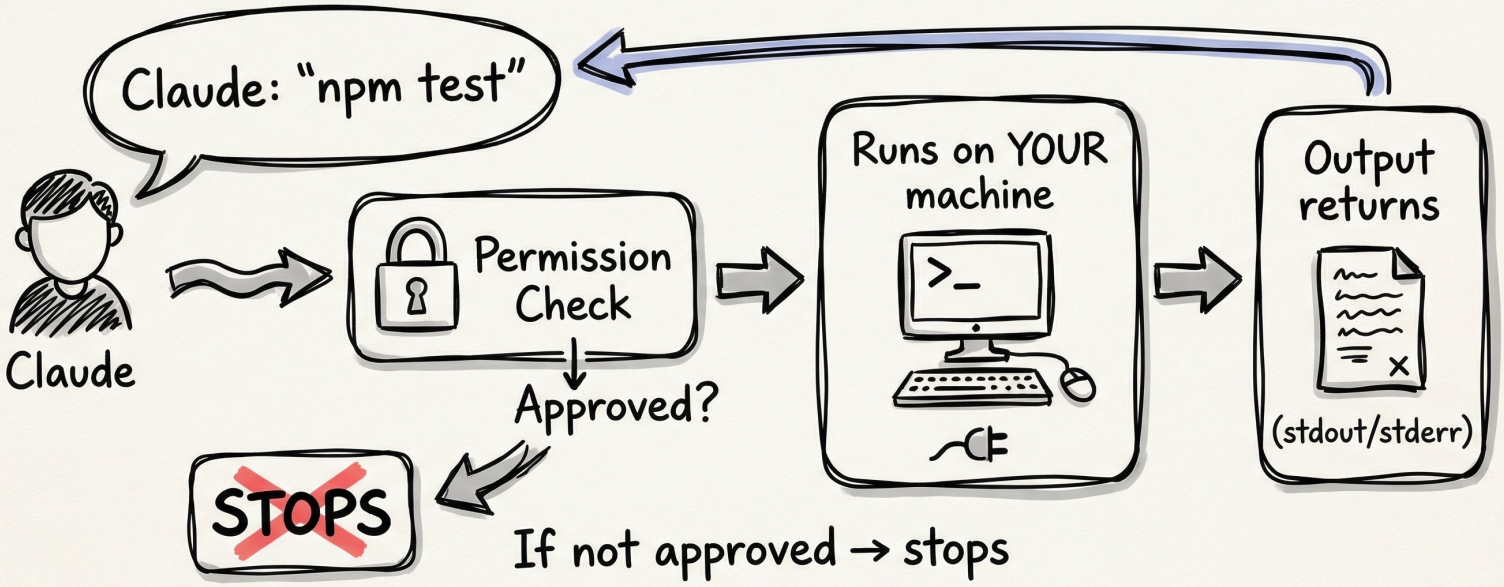
YOU CONTROL WHAT RUNS

Subagents: Parallel Workers



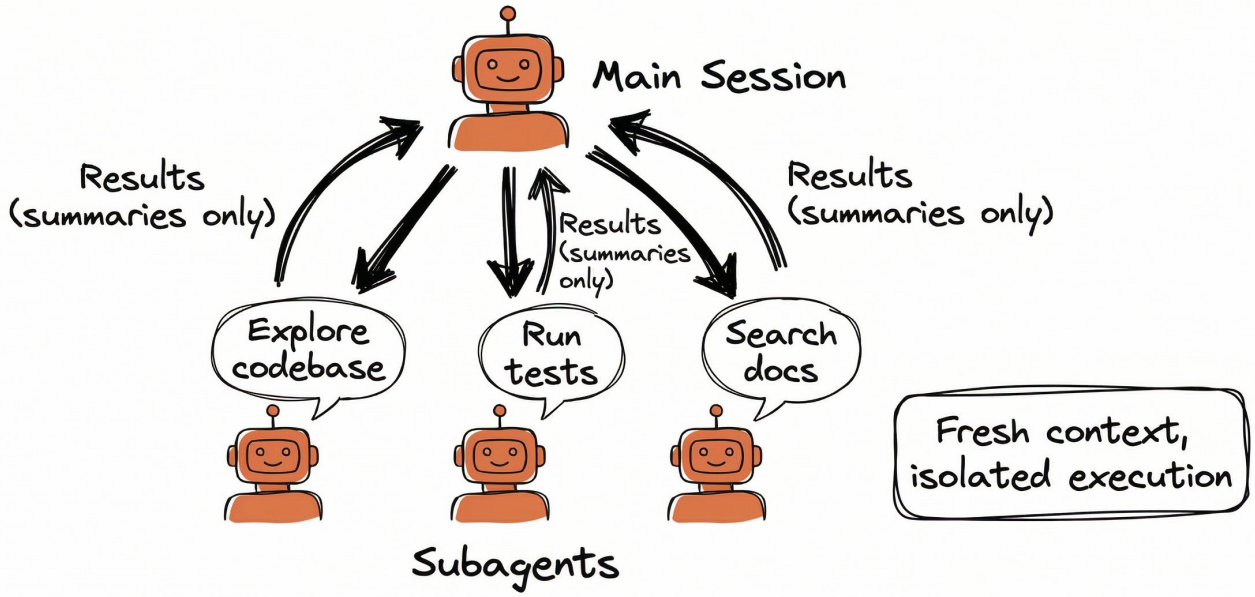
Parallel workers that keep main session clean

COMMAND EXECUTION FLOW

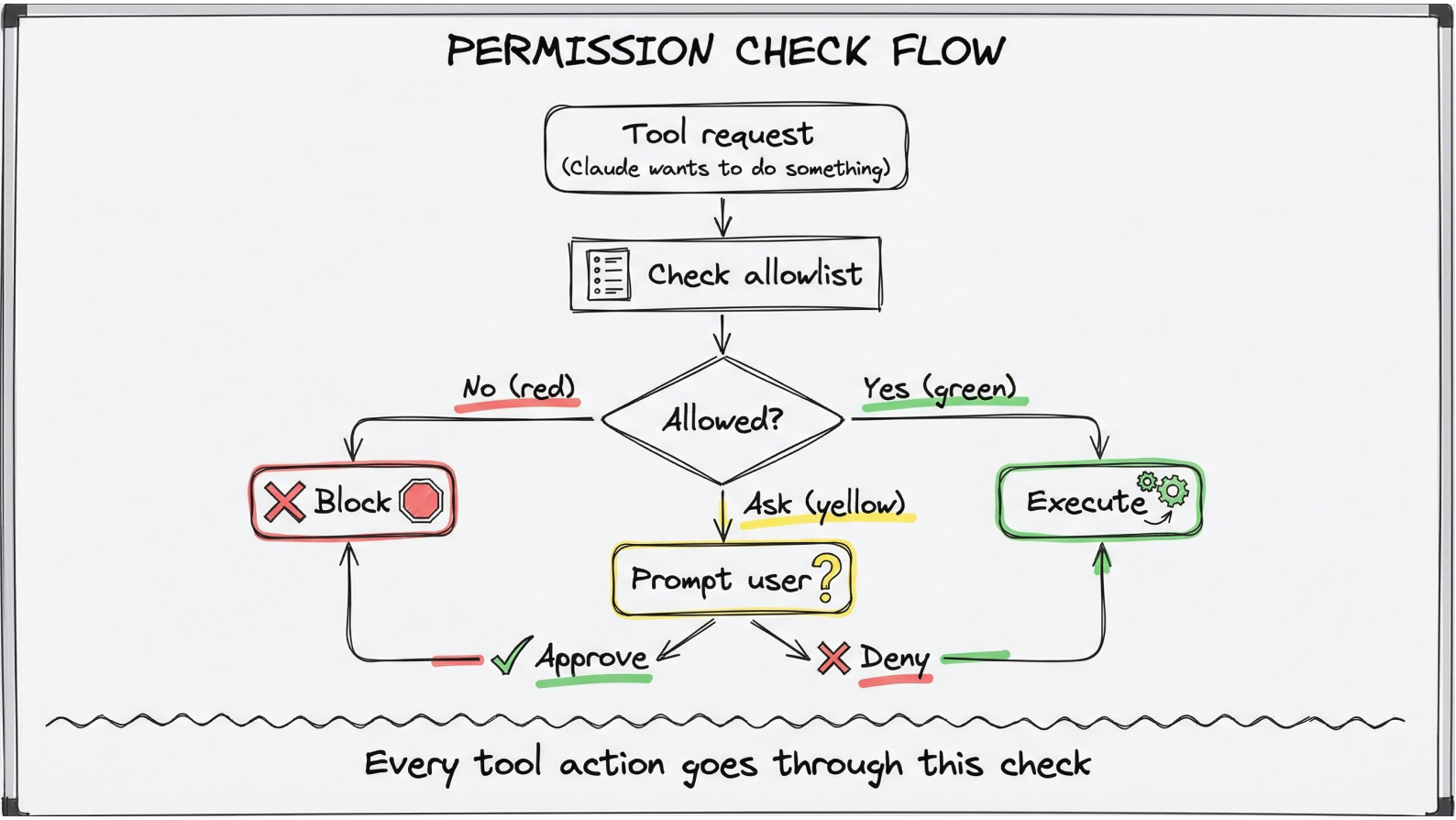


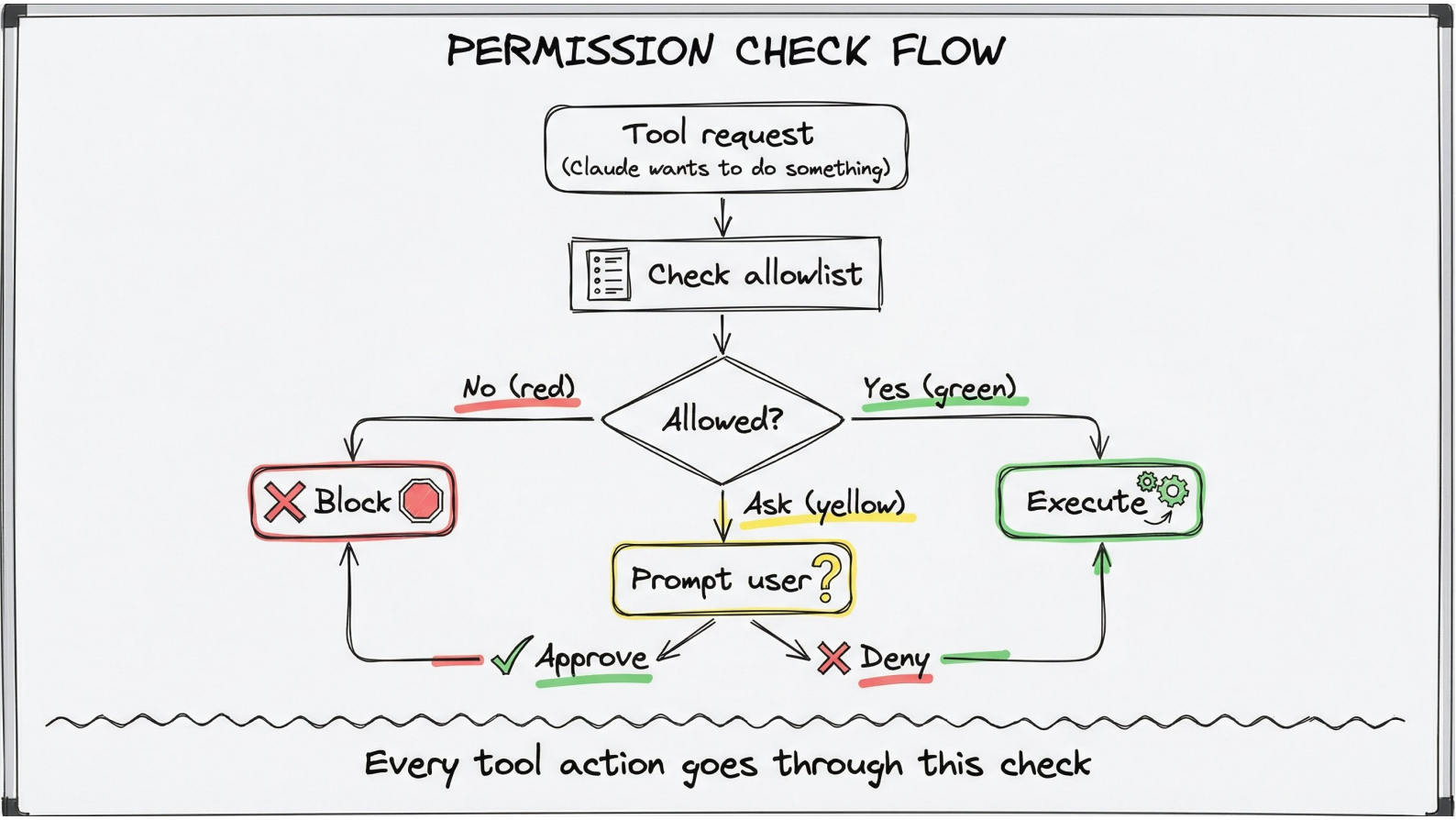
YOU CONTROL WHAT RUNS

Subagents: Parallel Workers



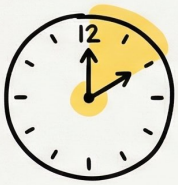
Parallel workers that keep main session clean


PERMISSION CHECK FLOW
Tool request
(Claude wants to do something)
Check allowlist
Allowed?
No (red)
Yes (green)
Block
Ask (yellow)
Execute
Prompt user?
Approve
Deny
Every tool action goes through this check


PERMISSION CHECK FLOW
Tool request
(Claude wants to do something)
Check allowlist
Allowed?
No (red)
Yes (green)
Block
Ask (yellow)
Execute
Prompt user?
Approve
Deny
Every tool action goes through this check

Timeouts & Background Jobs

SCENARIO 1: "Quick Command"

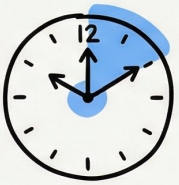


2 minutes

npm install → Done! ✓

Default timeout: 2 minutes

SCENARIO 2: "Long Command"



10 minutes

Big build job → runs in background

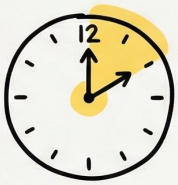
Can continue other work

Max timeout: 10 minutes

Long tasks run in background so you're not stuck

Timeouts & Background Jobs

SCENARIO 1: "Quick Command"

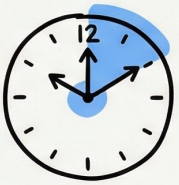


2 minutes

npm install → Done! ✓

Default timeout: 2 minutes

SCENARIO 2: "Long Command"



10 minutes

Big build job → runs in background

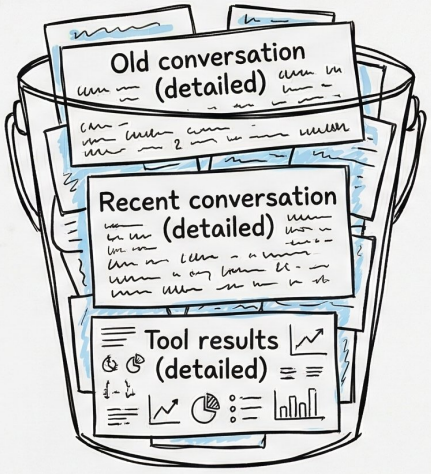
Can continue other work

Max timeout: 10 minutes

Long tasks run in background so you're not stuck

Automatic Compaction

BEFORE (full bucket)



Status: "Too full!"

Compaction

AFTER (room freed up)

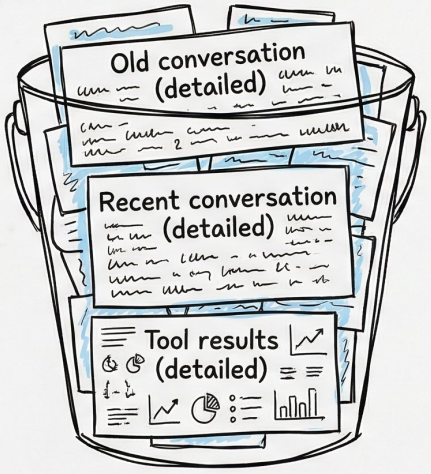


Status: "Room to continue!"

Old stuff gets summarized, recent stuff stays intact

Automatic Compaction

BEFORE (full bucket)



Status: "Too full!"

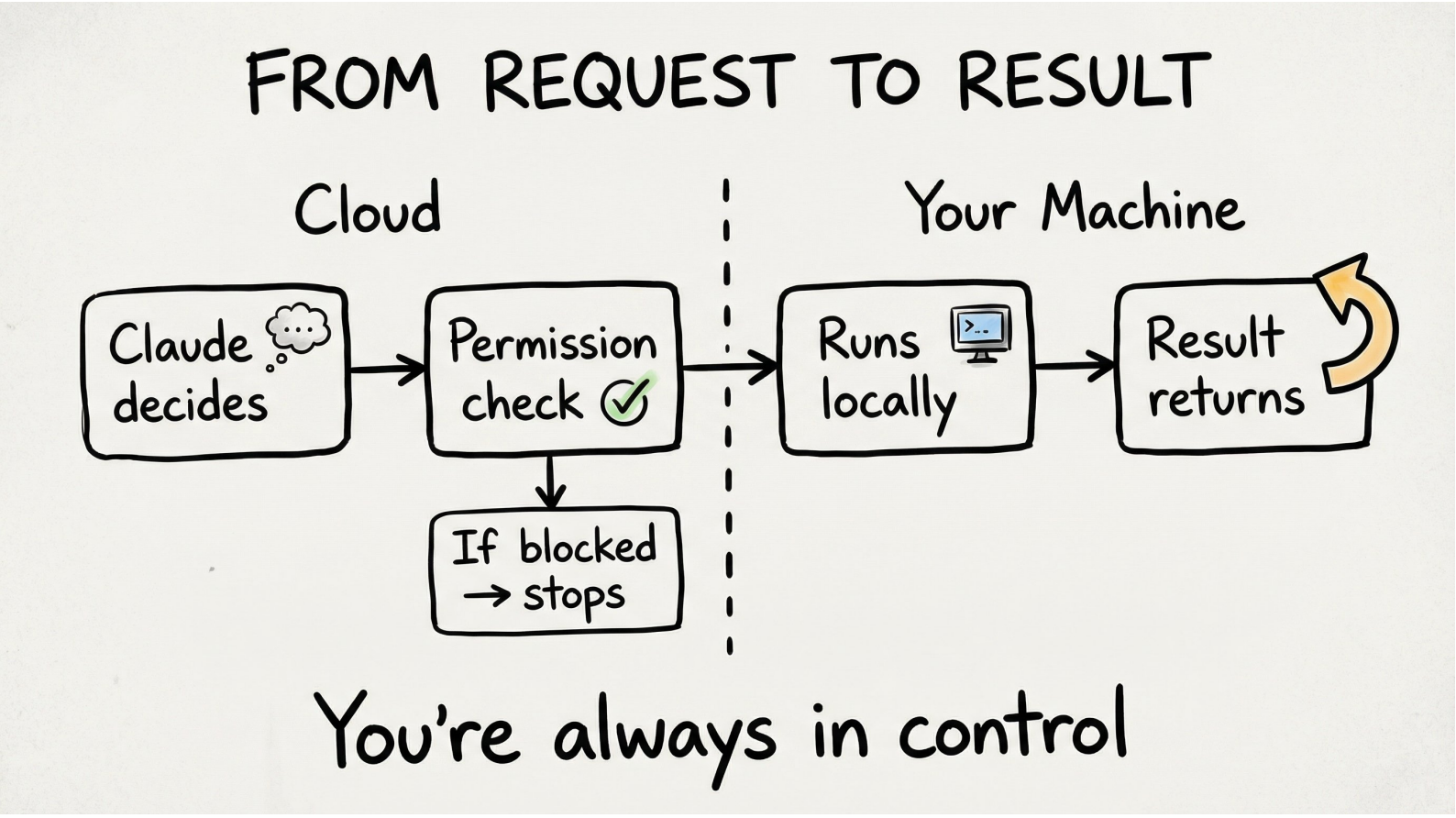
Compaction

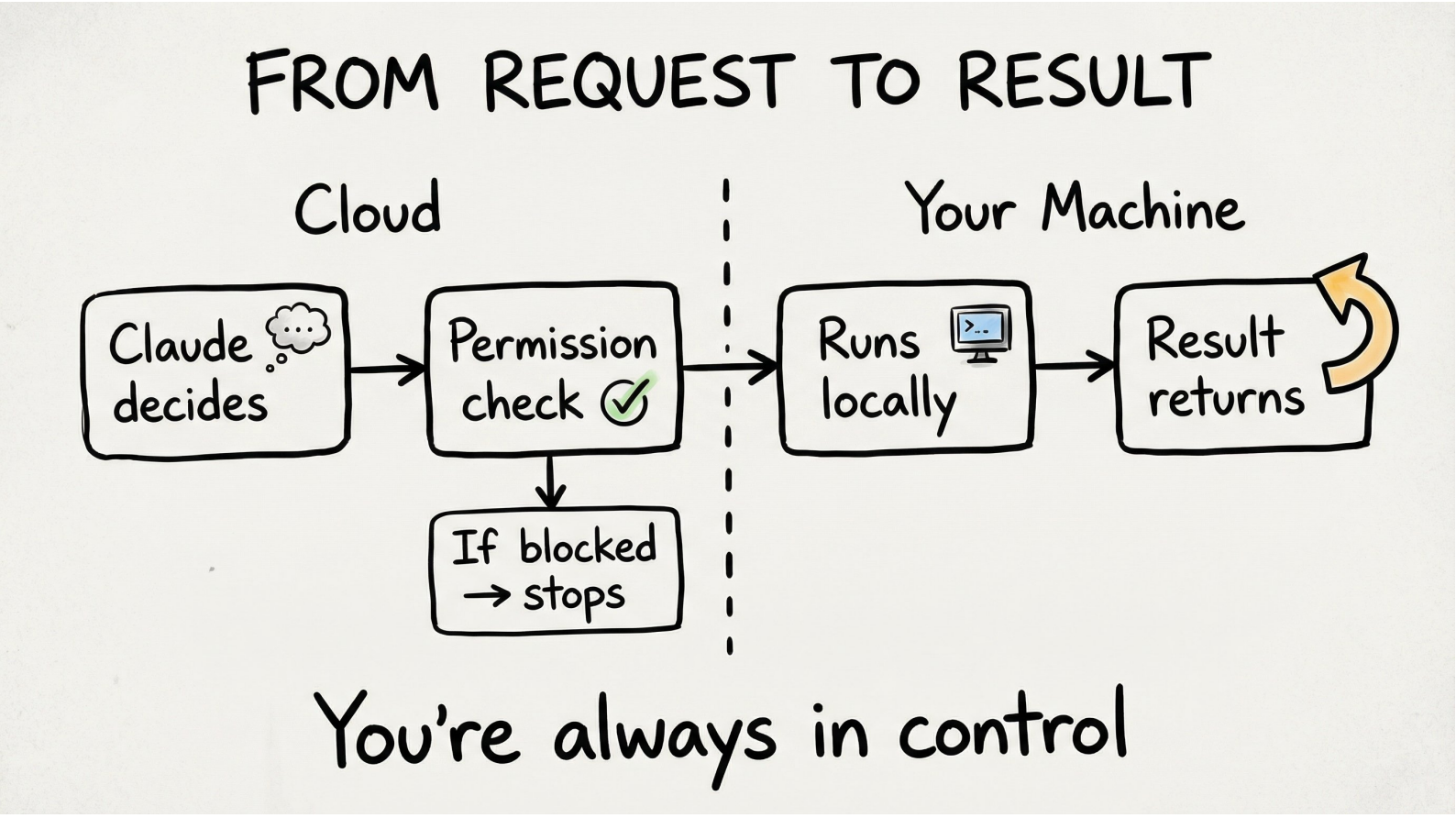
AFTER (room freed up)



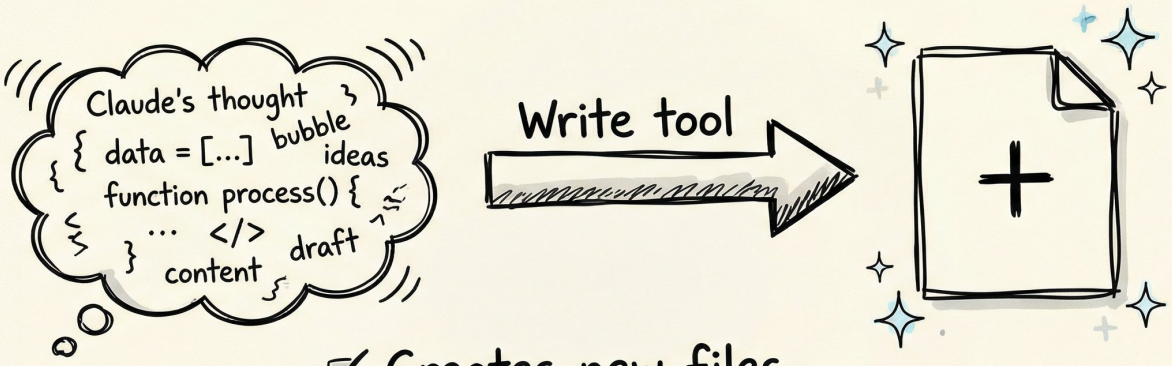
Status: "Room to continue!"

Old stuff gets summarized, recent stuff stays intact


FROM REQUEST TO RESULT
Cloud
Your Machine
Claude decides
Permission check
Runs locally
Result returns
If blocked → stops
You're always in control


FROM REQUEST TO RESULT
Cloud
Your Machine
Claude decides
Permission check
Runs locally
Result returns
If blocked → stops
You're always in control

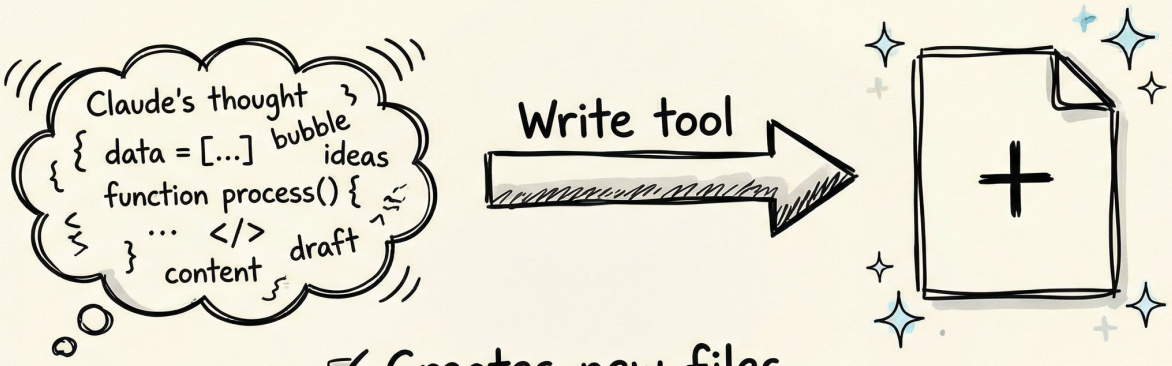
The Write Tool



- ☑ Creates new files
- ☑ Completely overwrites if exists
- ☑ You approve first

Warning note: Be careful - this replaces entire file!

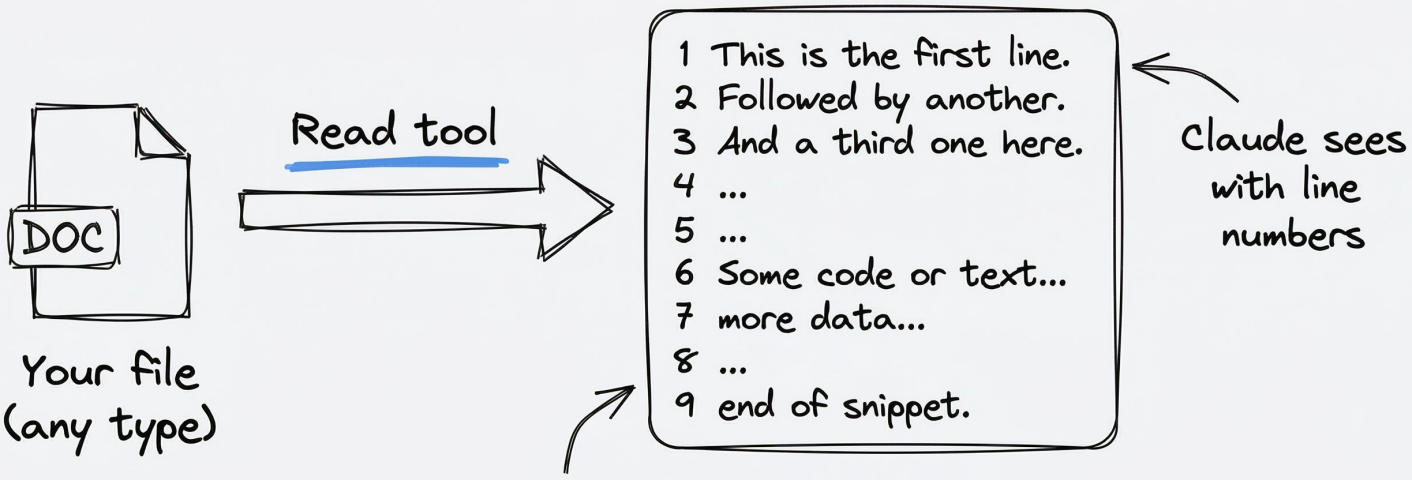
The Write Tool



- ☑ Creates new files
- ☑ Completely overwrites if exists
- ☑ You approve first

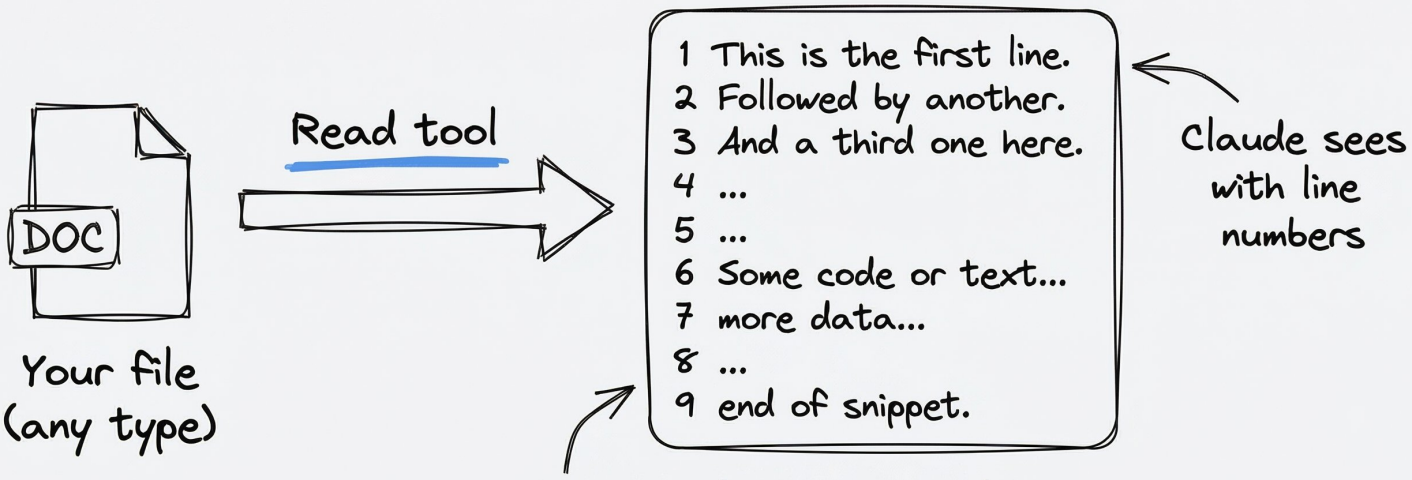
Warning note: Be careful - this replaces entire file!

The Read Tool



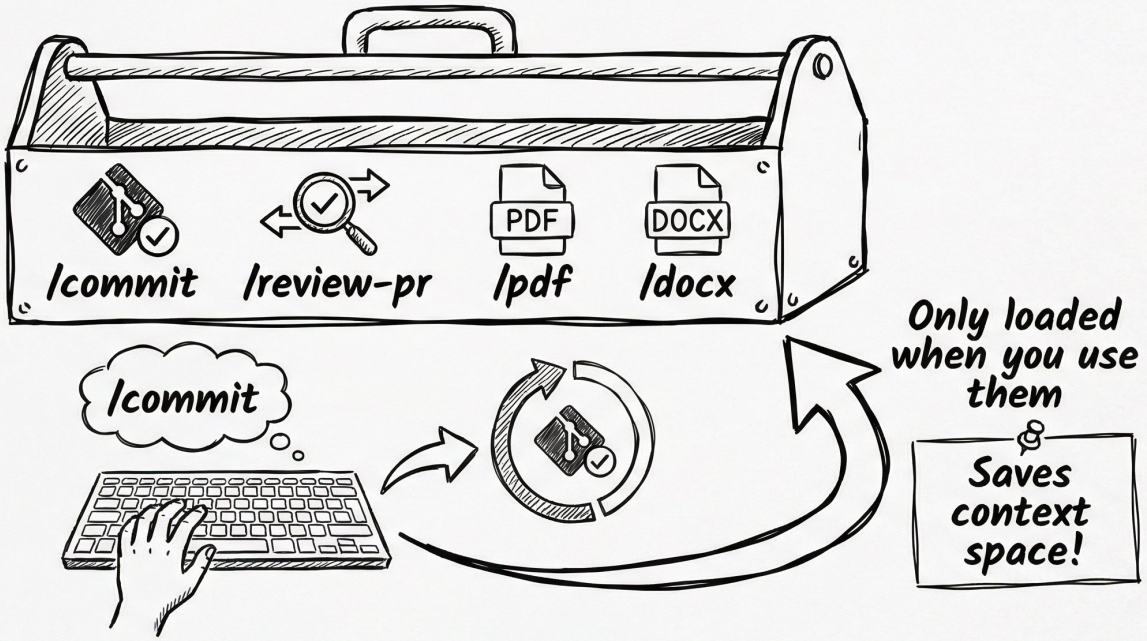
Claude reads files exactly as they are

The Read Tool



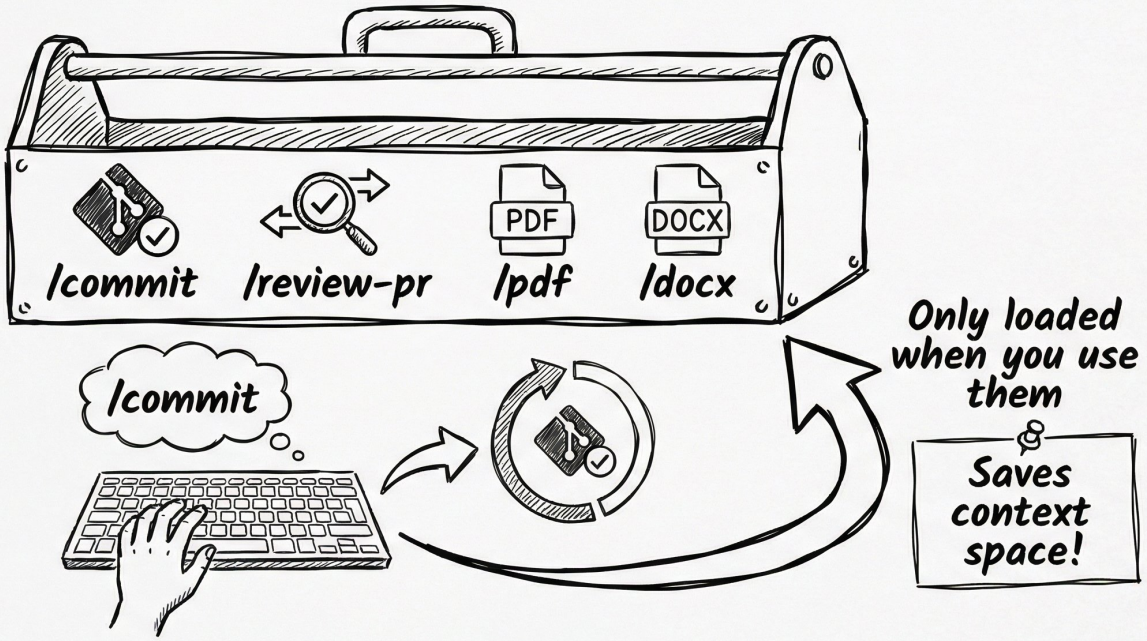
Claude reads files exactly as they are

SKILLS: ON-DEMAND WORKFLOWS

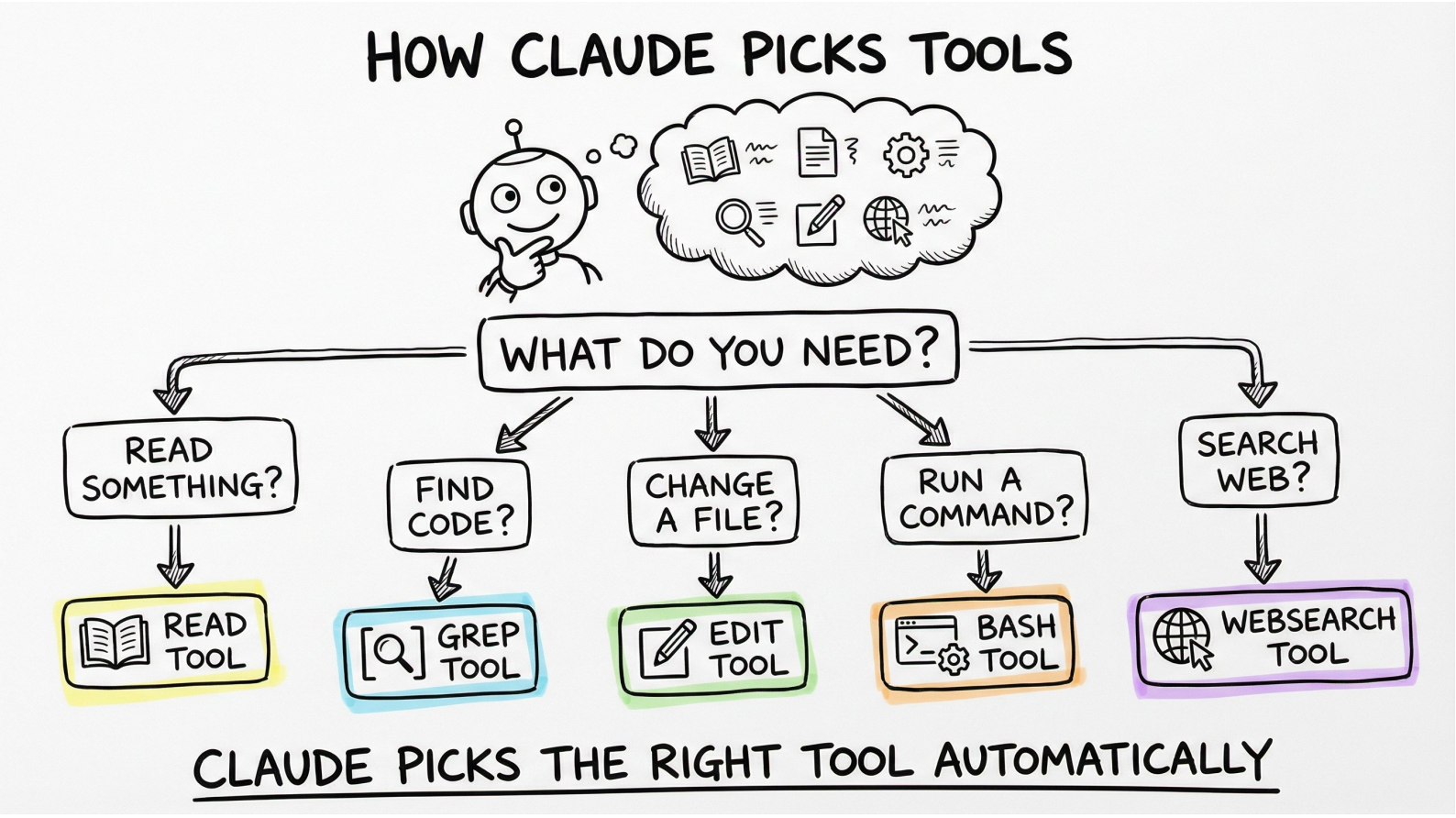


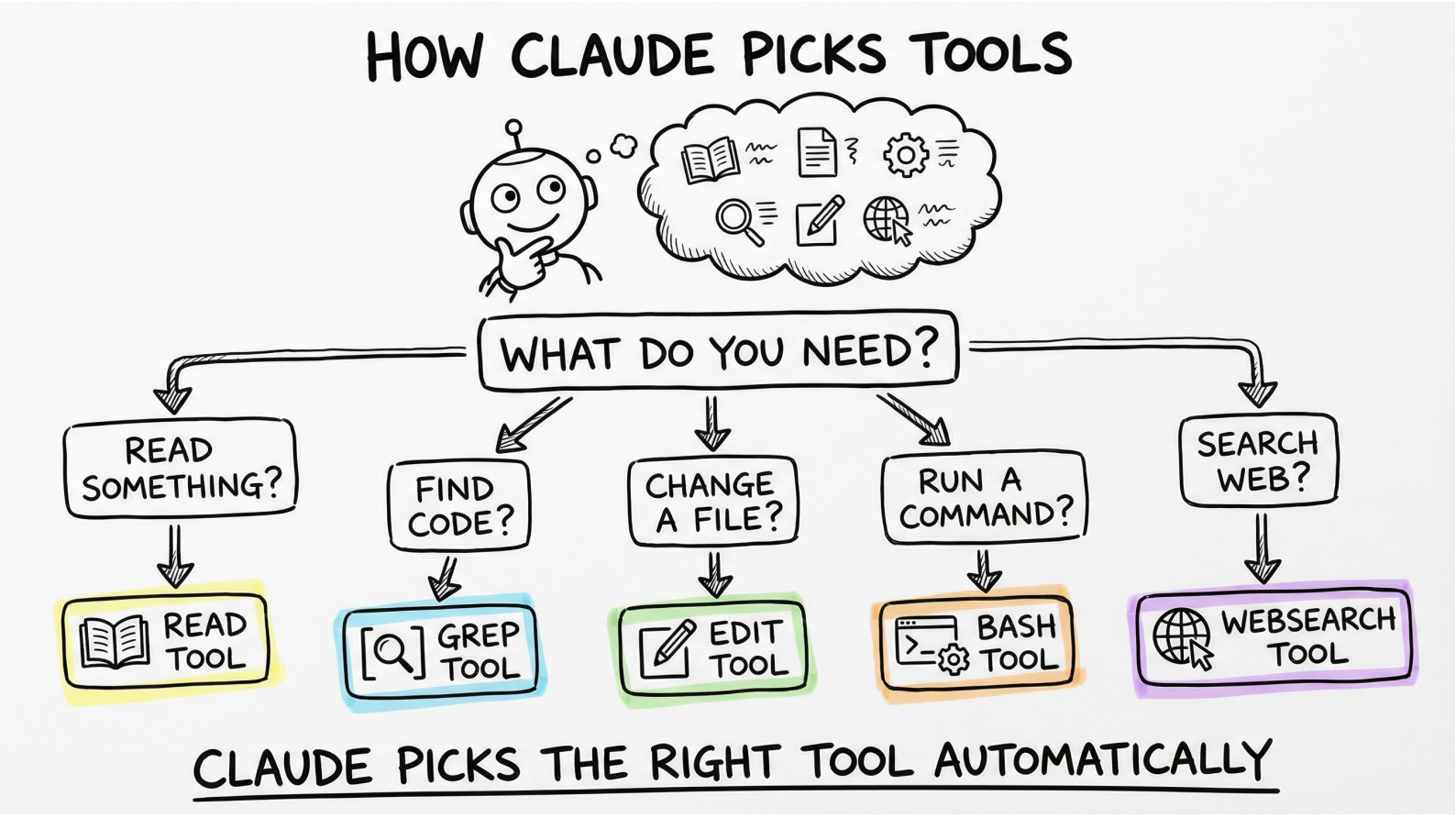
On-demand workflows - loaded only when needed

SKILLS: ON-DEMAND WORKFLOWS

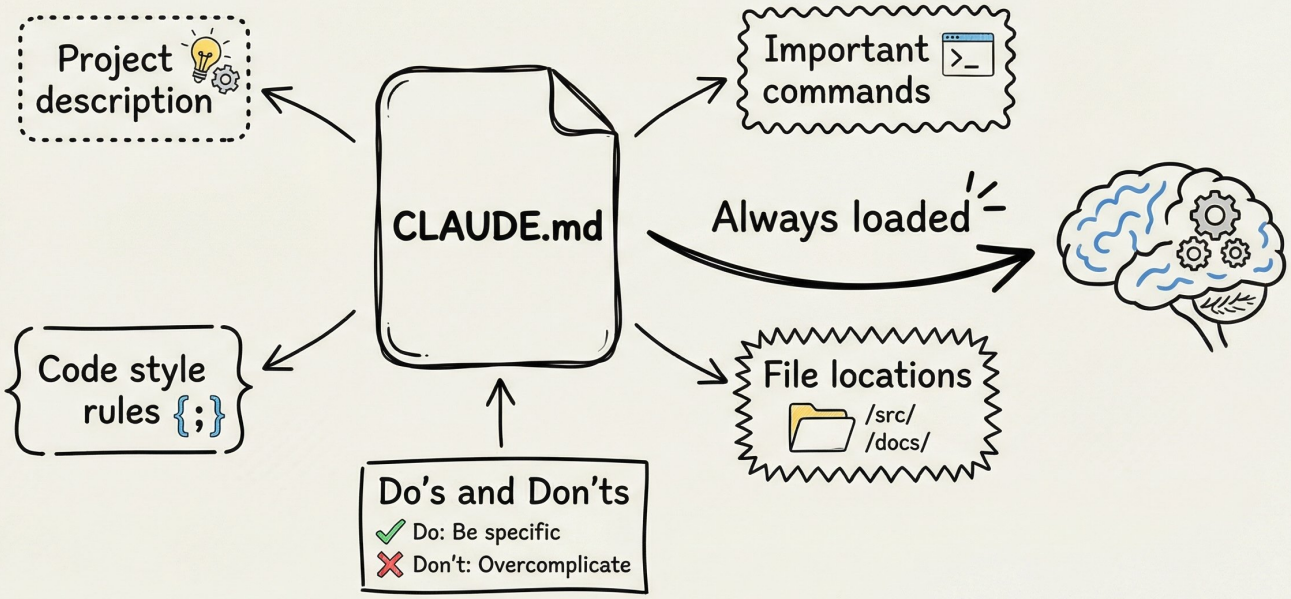


On-demand workflows - loaded only when needed


HOW CLAUDE PICKS TOOLS
WHAT DO YOU NEED?
READ SOMETHING?
FIND CODE?
CHANGE A FILE?
RUN A COMMAND?
SEARCH WEB?
READ TOOL
GREP TOOL
EDIT TOOL
BASH TOOL
WEBSEARCH TOOL
CLAUDE PICKS THE RIGHT TOOL AUTOMATICALLY

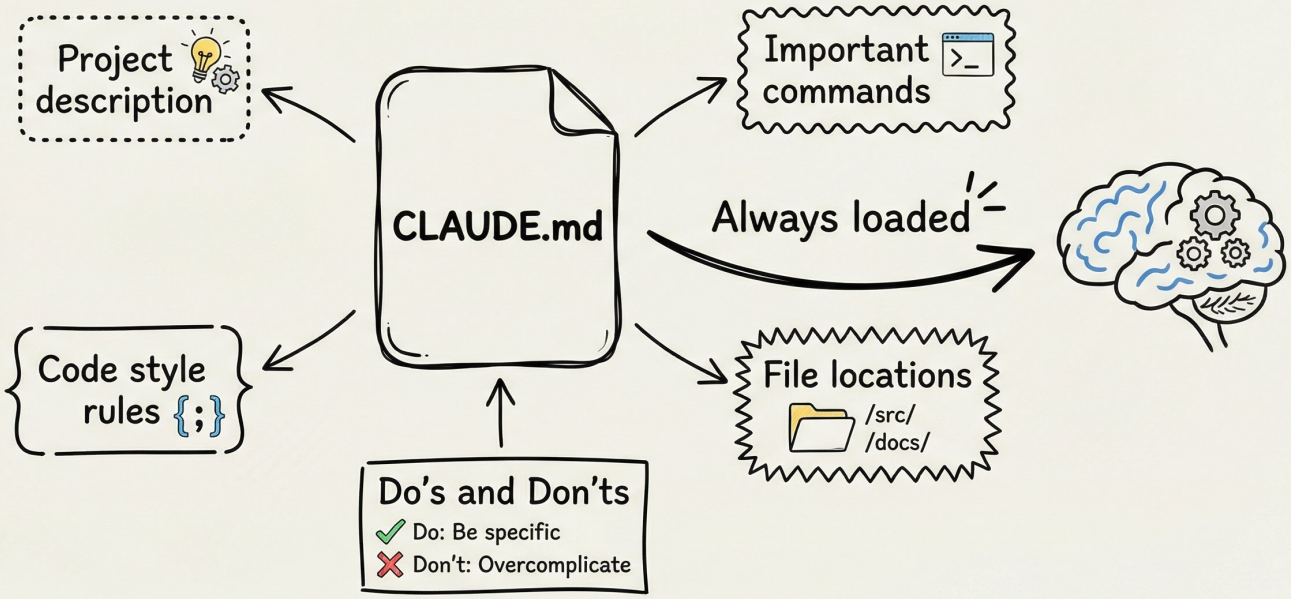

HOW CLAUDE PICKS TOOLS
WHAT DO YOU NEED?
READ SOMETHING?
FIND CODE?
CHANGE A FILE?
RUN A COMMAND?
SEARCH WEB?
READ TOOL
GREP TOOL
EDIT TOOL
BASH TOOL
WEBSEARCH TOOL
CLAUDE PICKS THE RIGHT TOOL AUTOMATICALLY

CLAUDE.md: Project Instructions



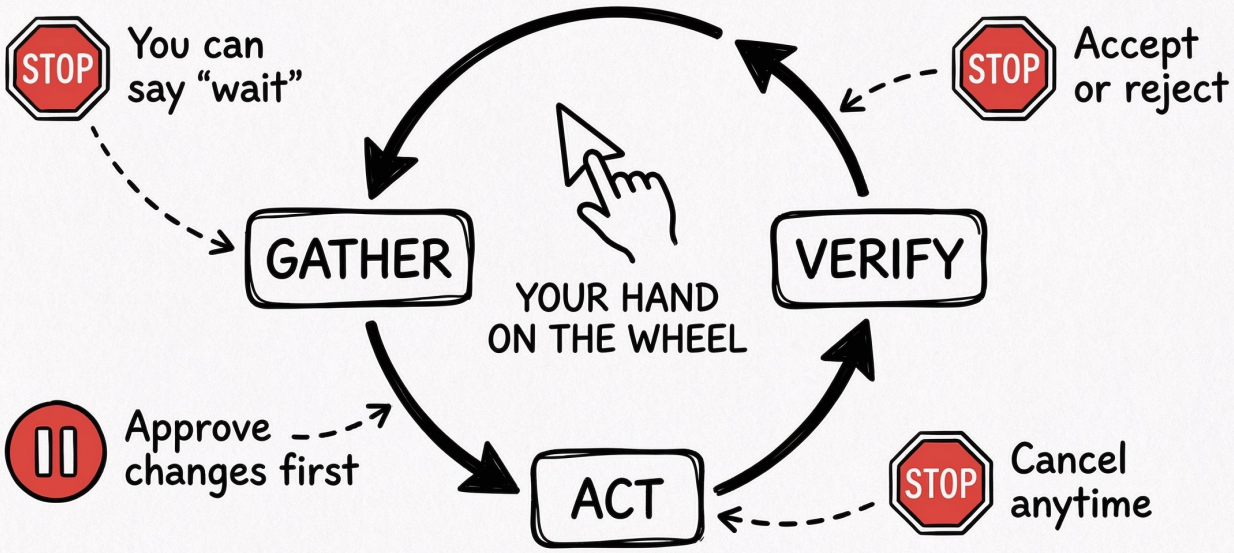
Your project's instruction manual for Claude

CLAUDE.md: Project Instructions



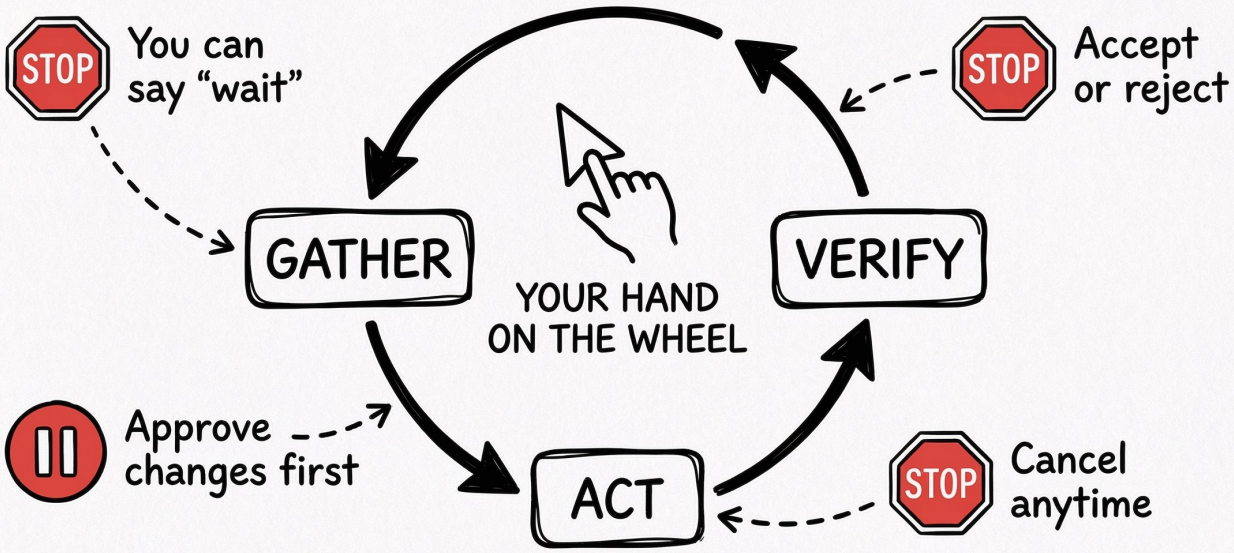
Your project's instruction manual for Claude

You're Always in Control



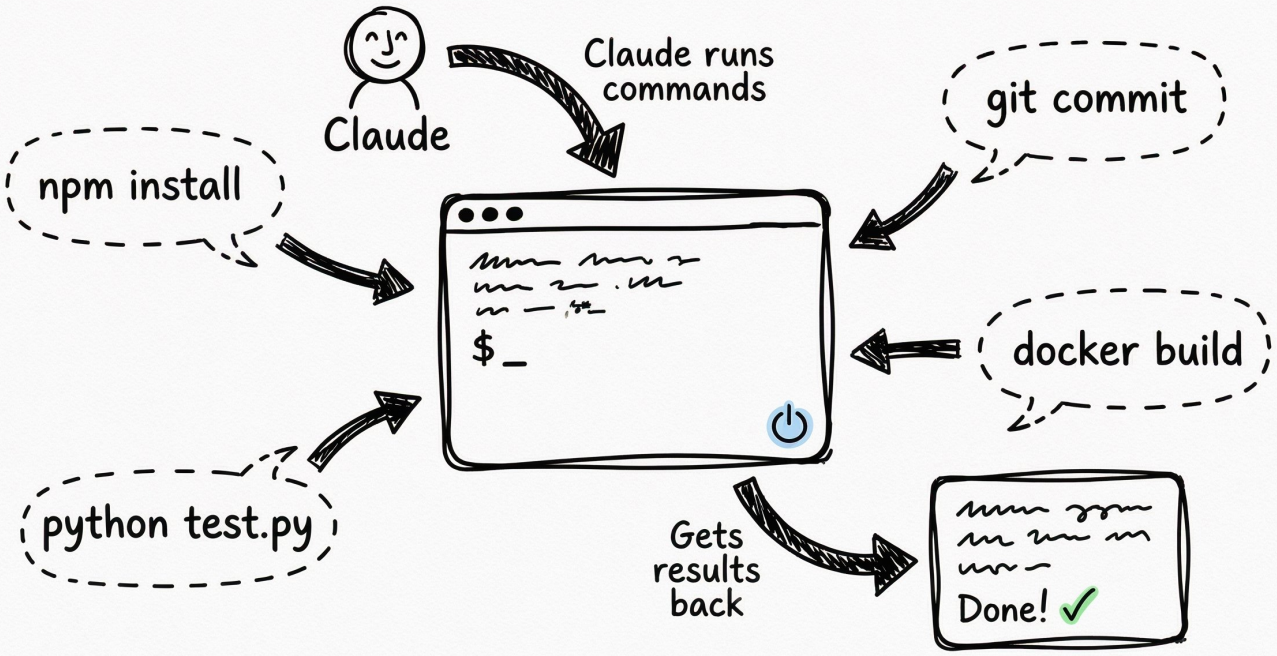
Claude suggests, you decide

You're Always in Control



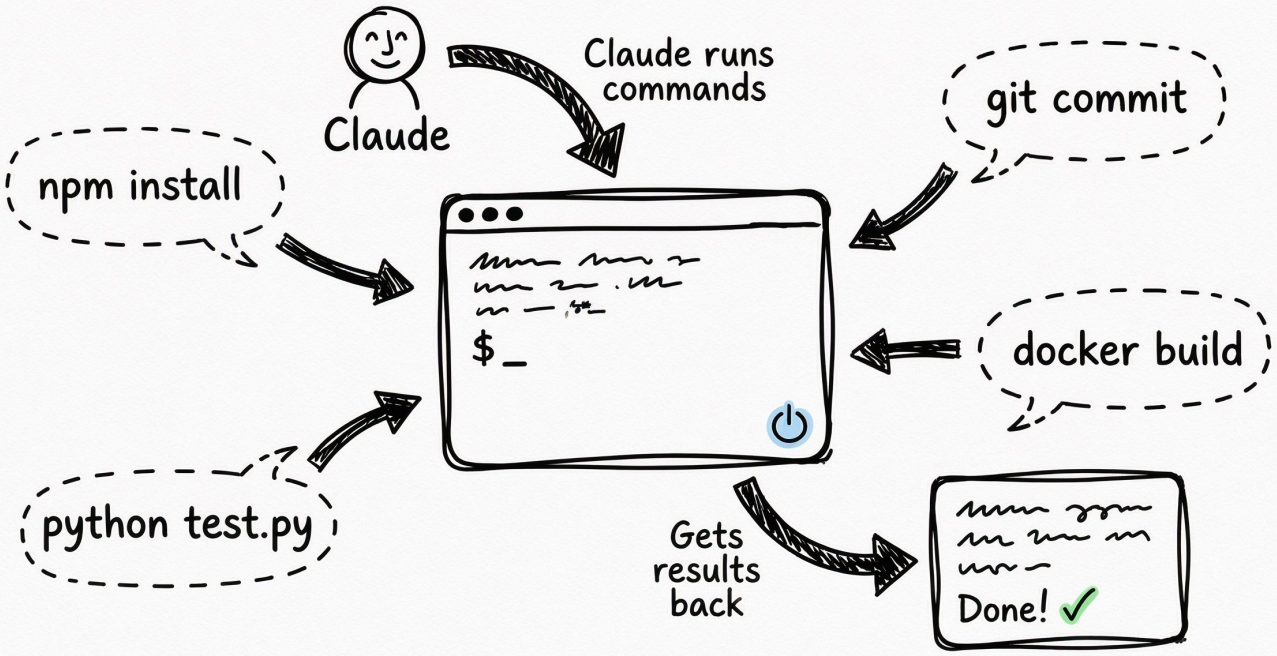
Claude suggests, you decide

THE BASH TOOL



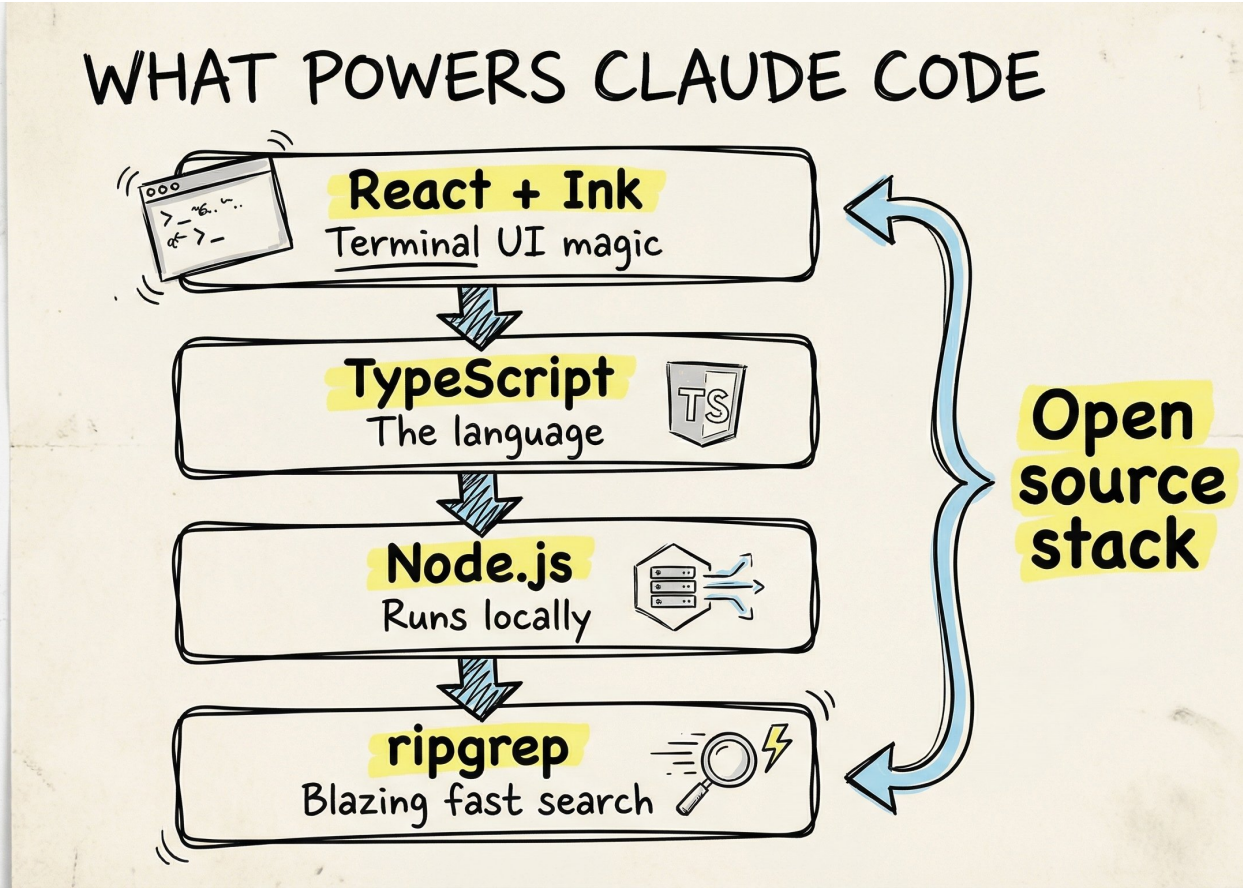
CLAUDE CAN RUN ANY COMMAND ON YOUR MACHINE

THE BASH TOOL

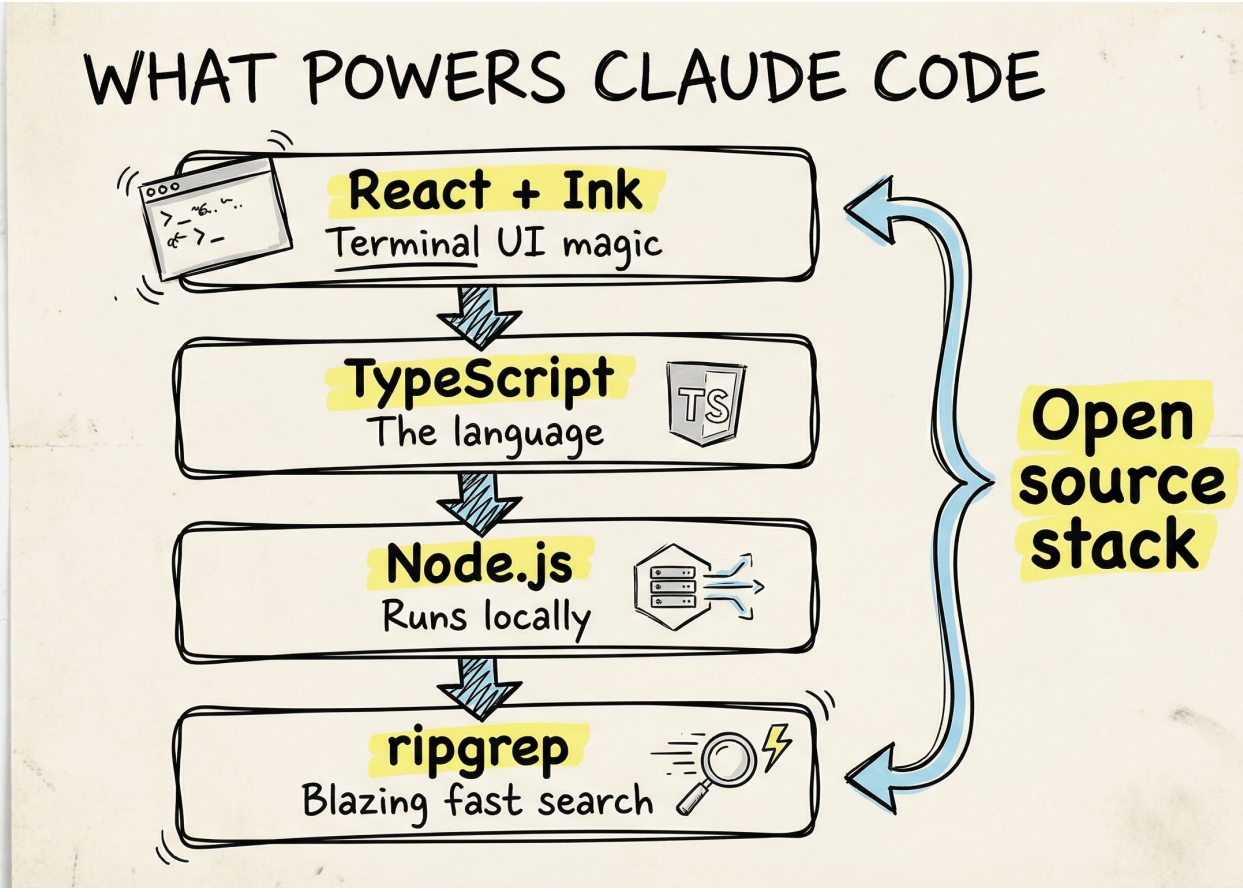


CLAUDE CAN RUN ANY COMMAND ON YOUR MACHINE

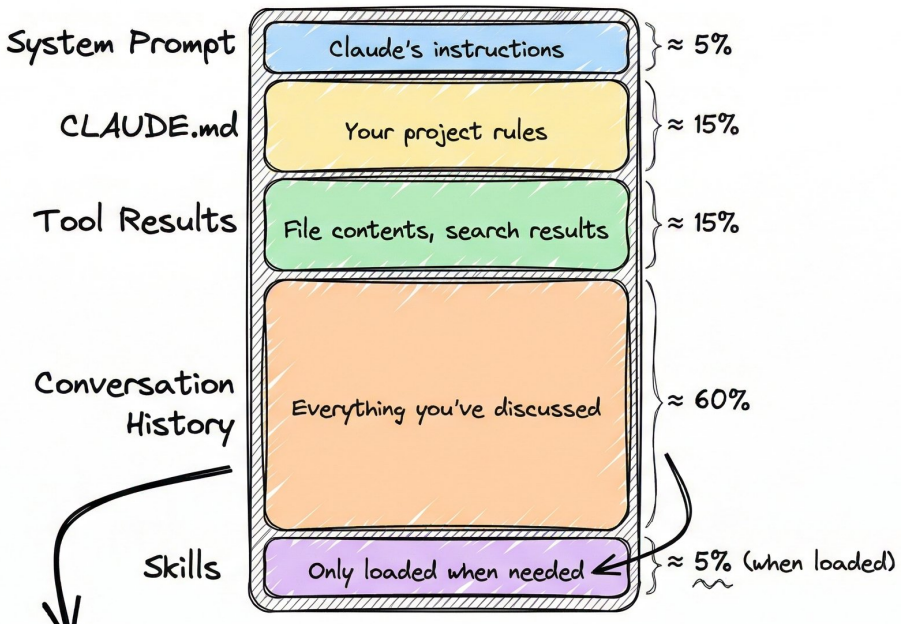
WHAT POWERS CLAUDE CODE



WHAT POWERS CLAUDE CODE

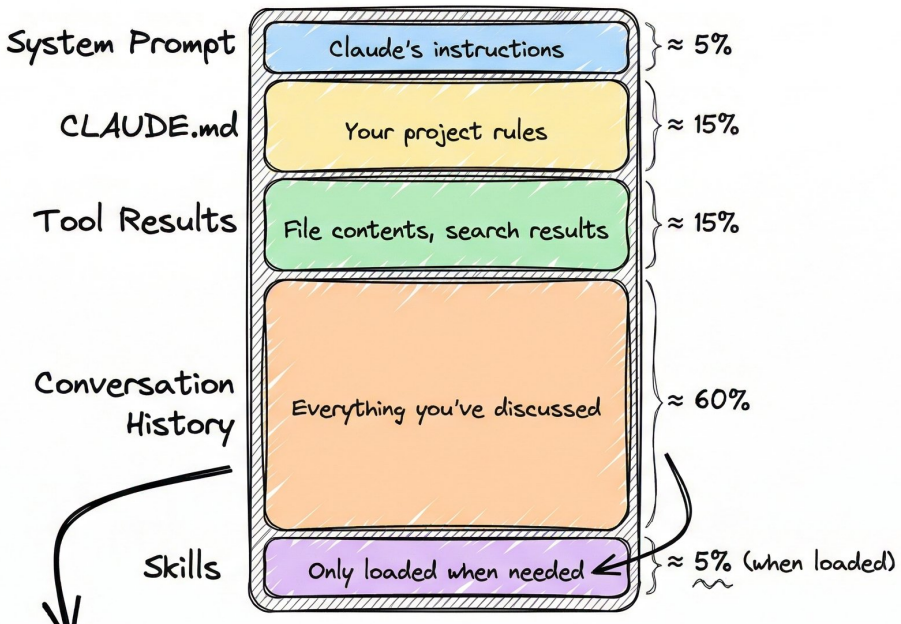


What Takes Up Space



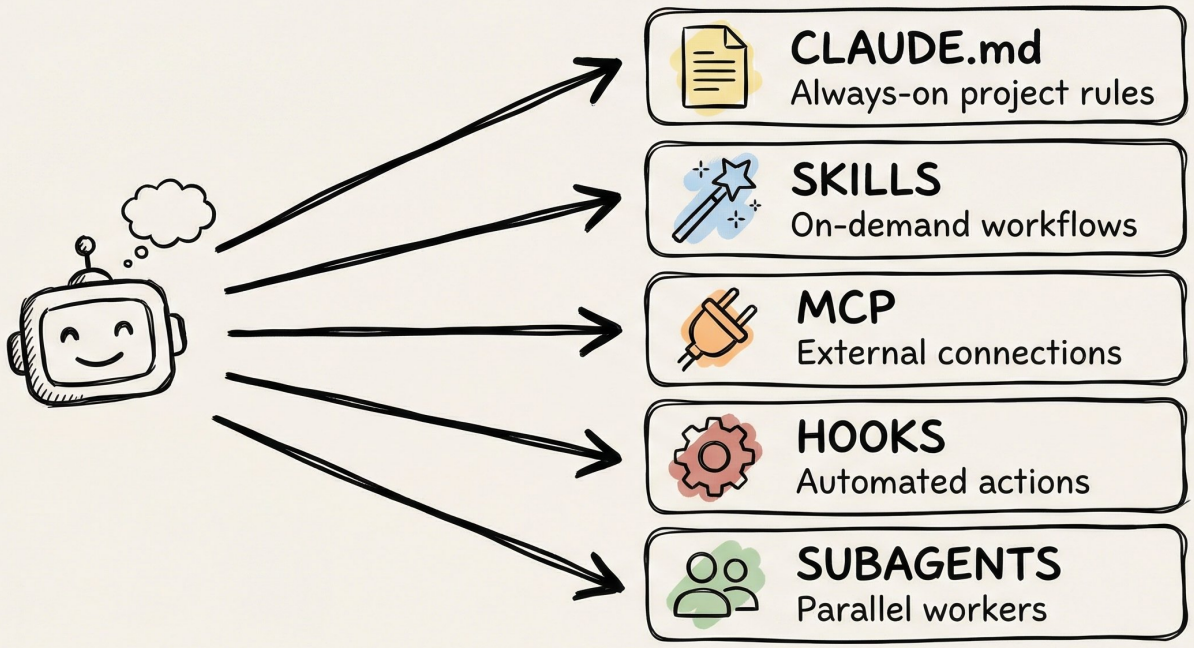
Long conversations = less room for tool results

What Takes Up Space



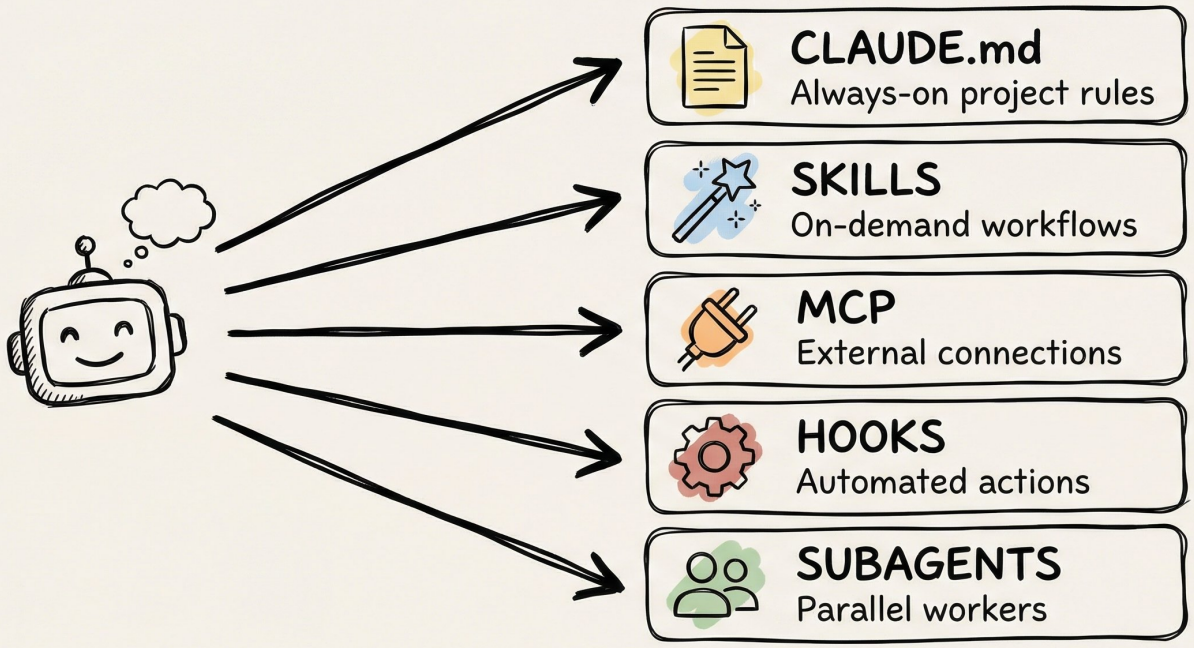
Long conversations = less room for tool results

5 WAYS TO EXTEND CLAUDE CODE

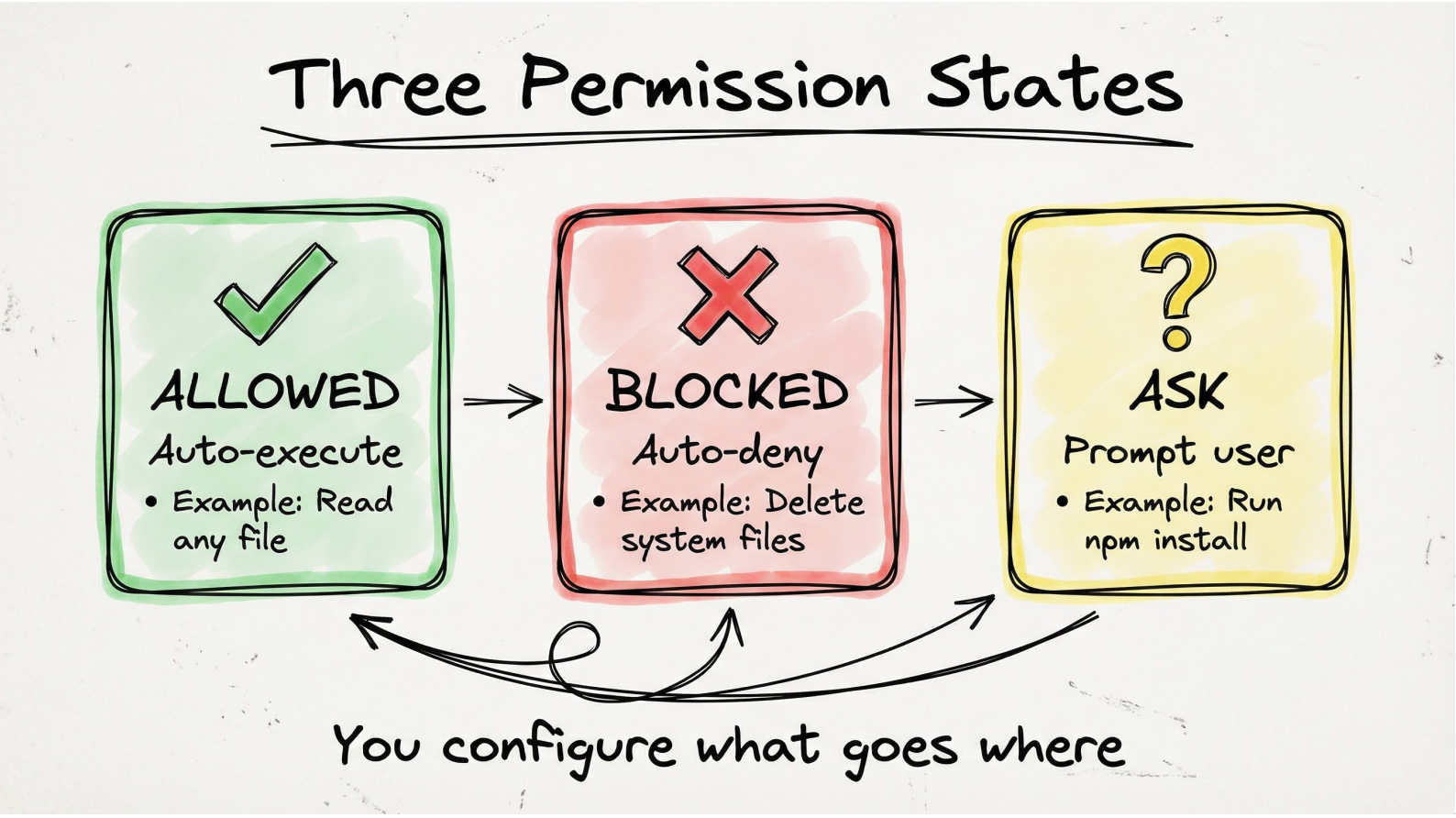


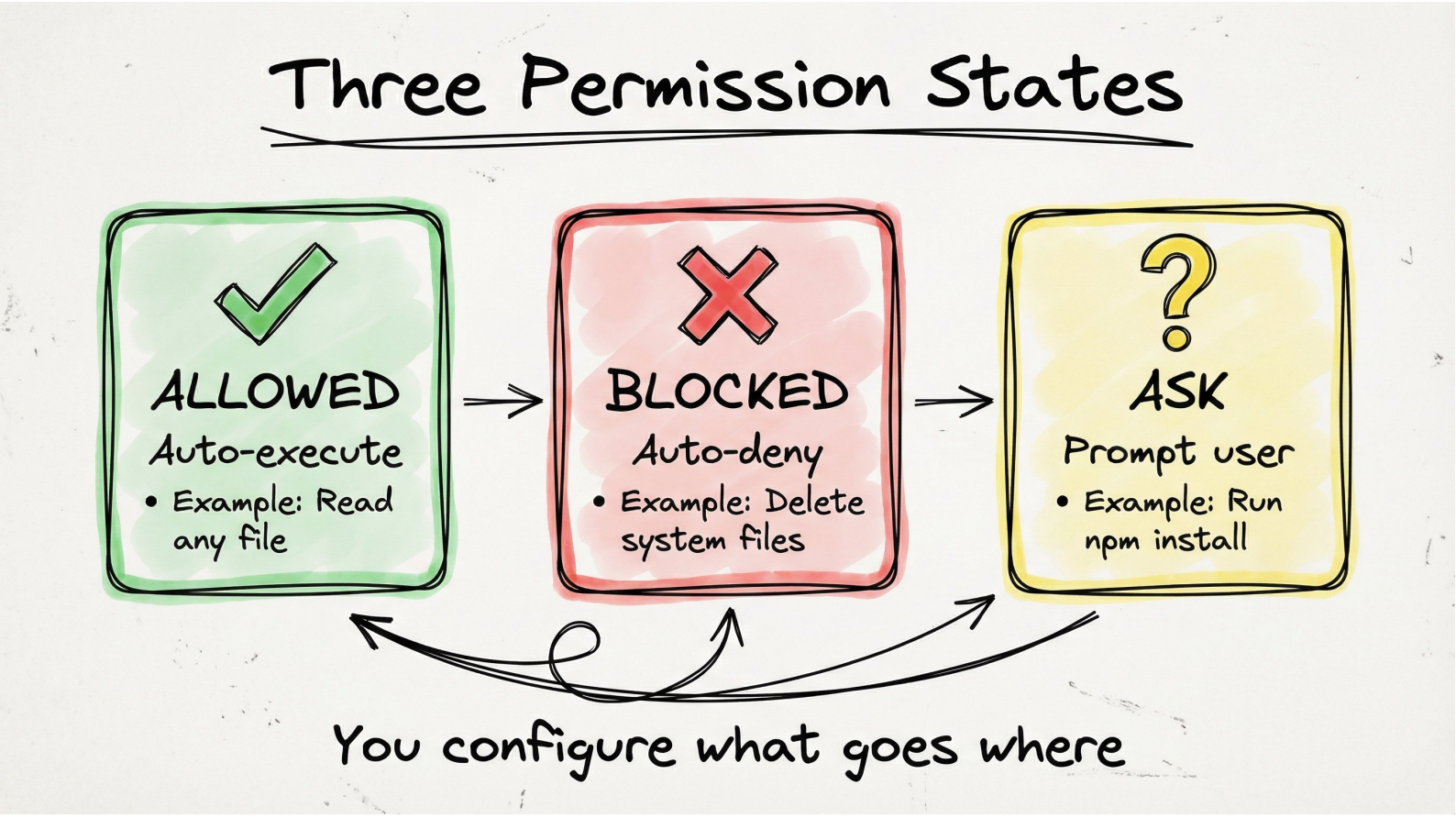
Customize Claude to fit your workflow

5 WAYS TO EXTEND CLAUDE CODE

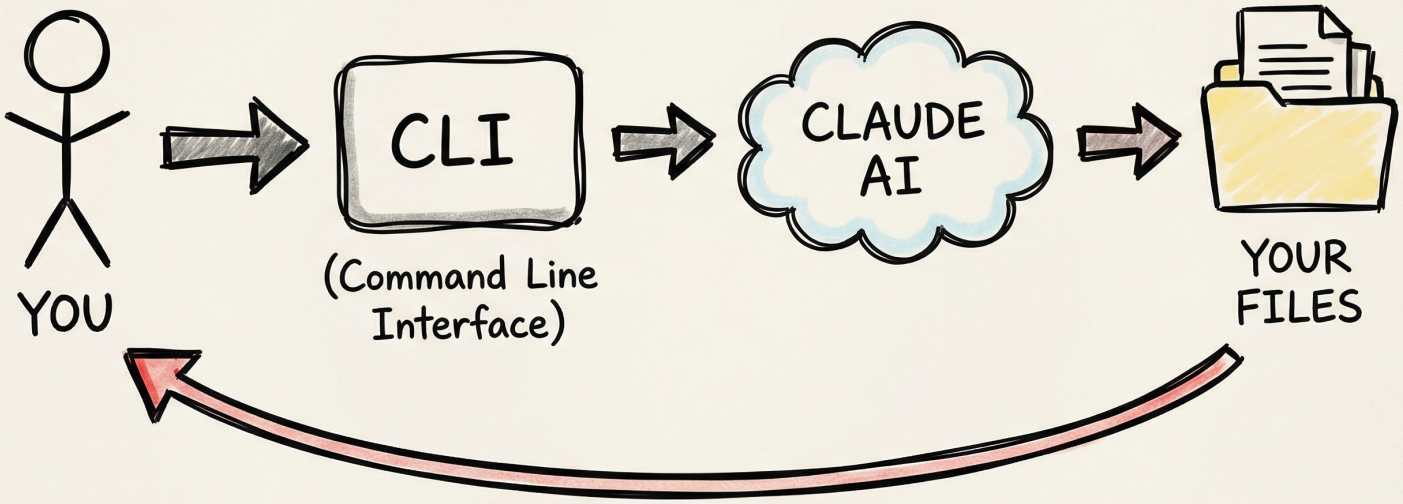


Customize Claude to fit your workflow


Three Permission States
ALLOWED
Auto-execute
• Example: Read any file
BLOCKED
Auto-deny
• Example: Delete system files
ASK
Prompt user
• Example: Run npm install
You configure what goes where

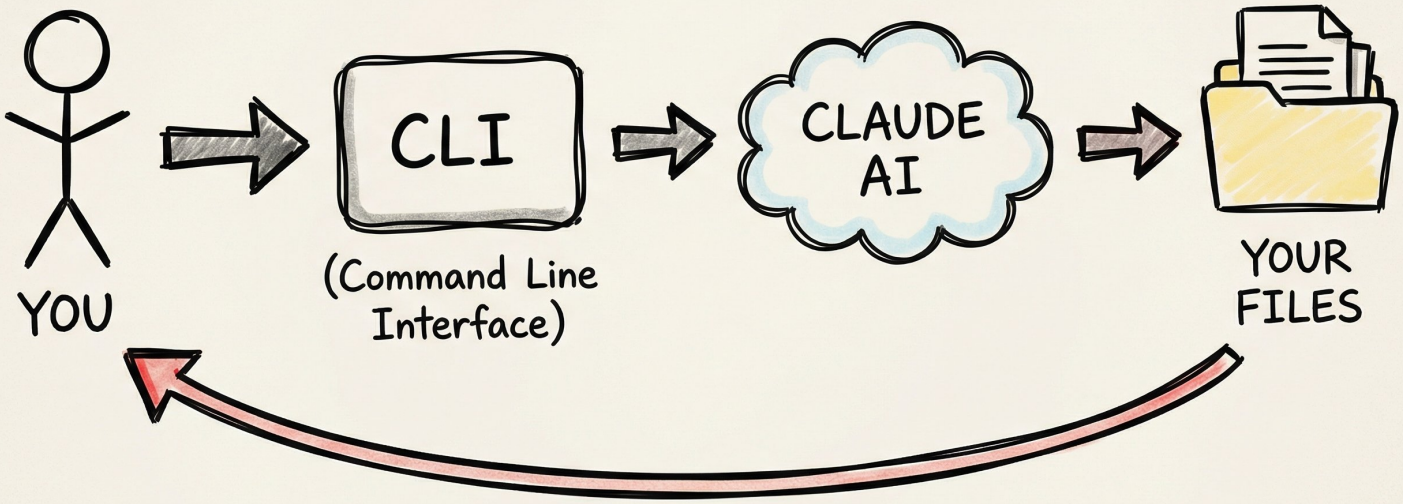

Three Permission States
ALLOWED
Auto-execute
• Example: Read any file
BLOCKED
Auto-deny
• Example: Delete system files
ASK
Prompt user
• Example: Run npm install
You configure what goes where

WHAT IS CLAUDE CODE?



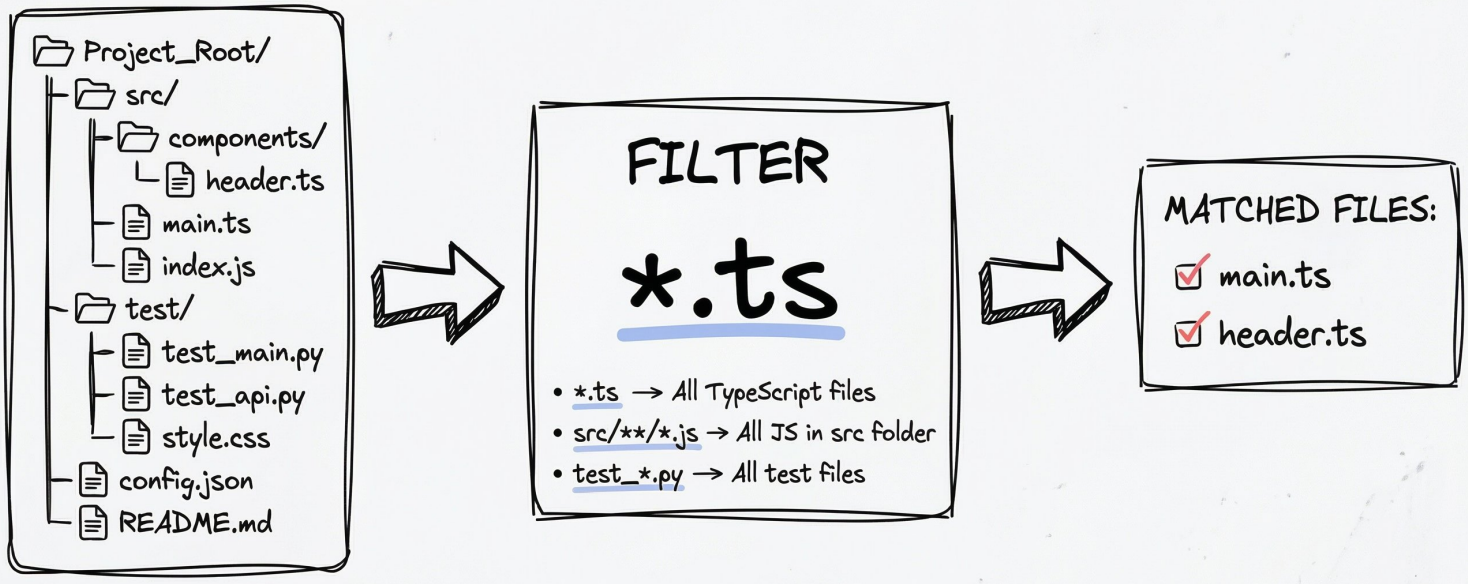
A CLI that connects you to Claude AI for coding

WHAT IS CLAUDE CODE?



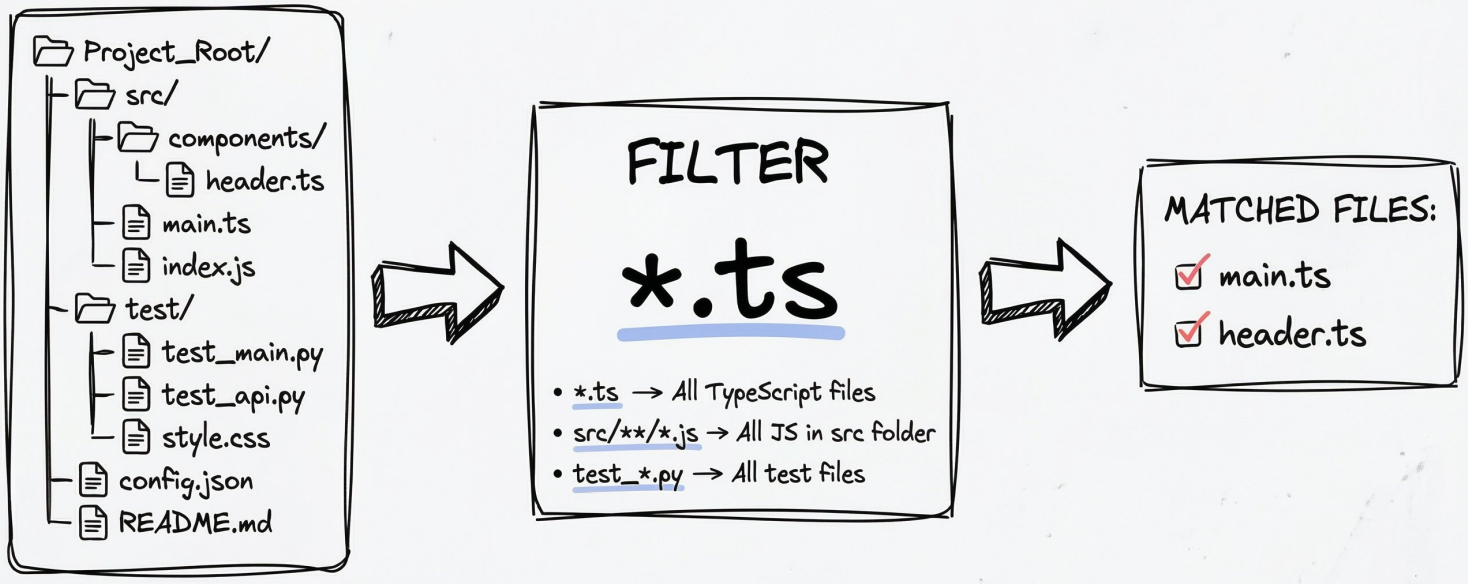
A CLI that connects you to Claude AI for coding

The Glob Tool



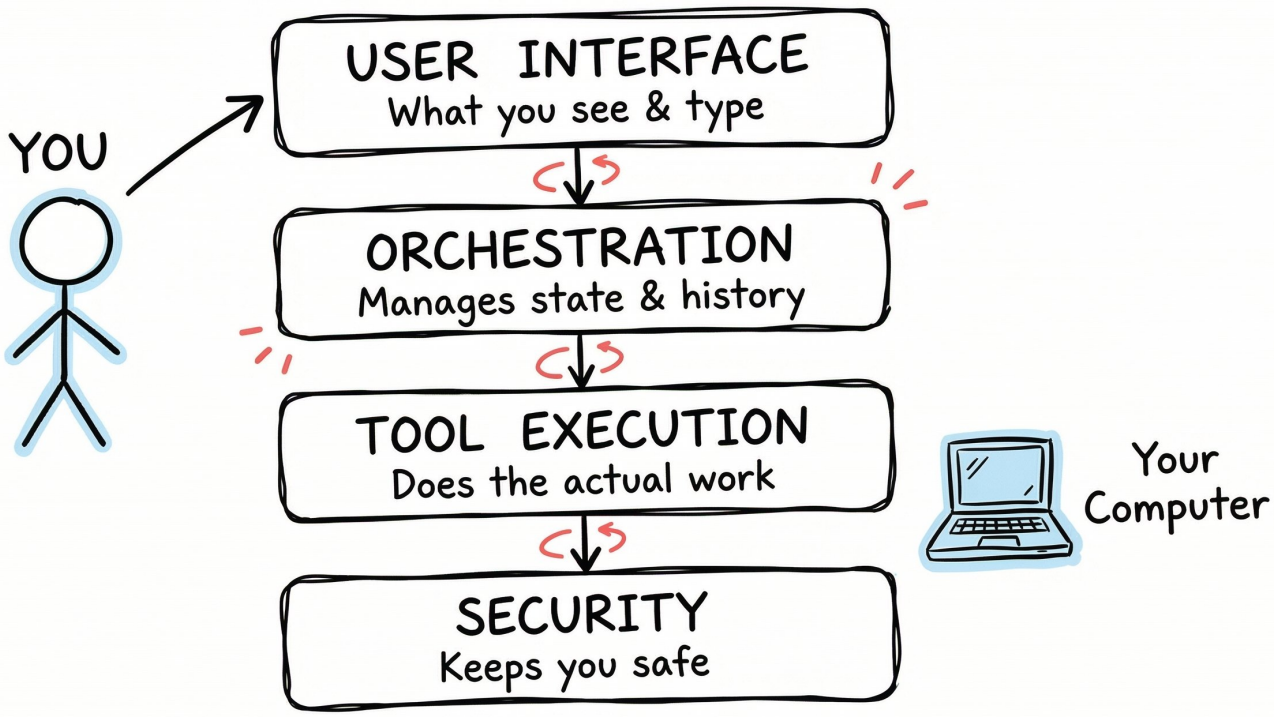
Find files by name patterns

The Glob Tool

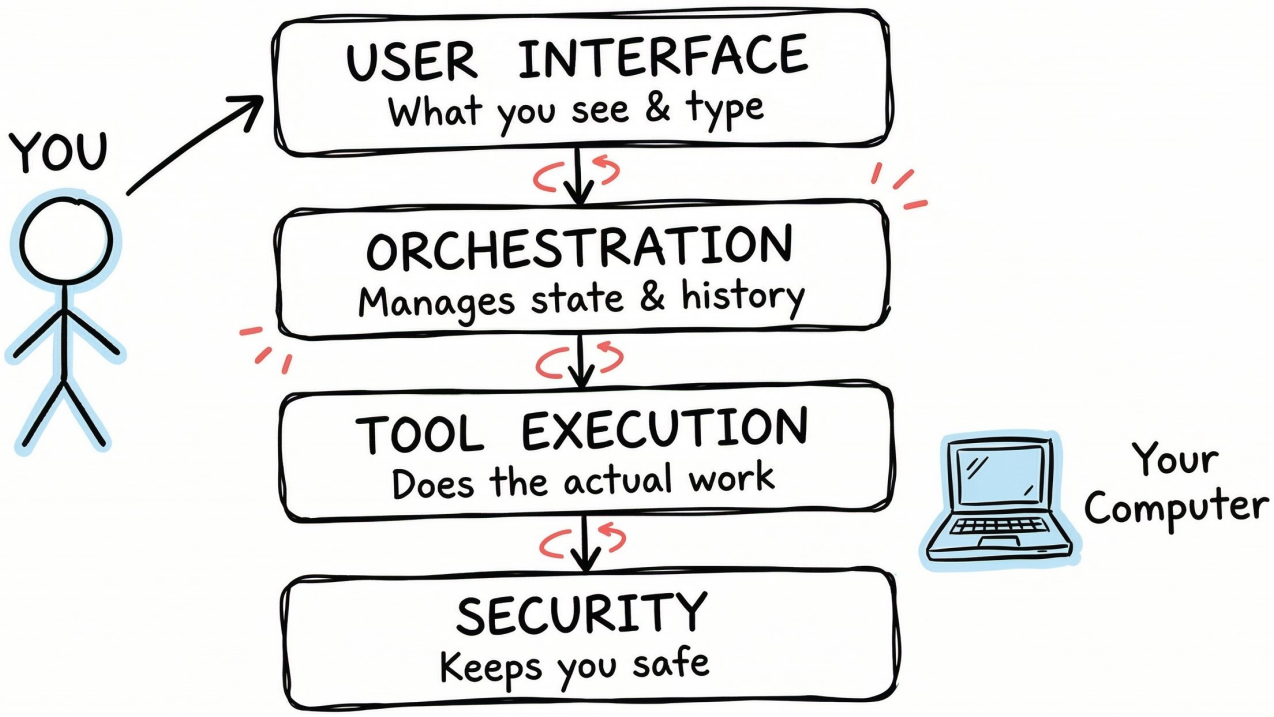


Find files by name patterns

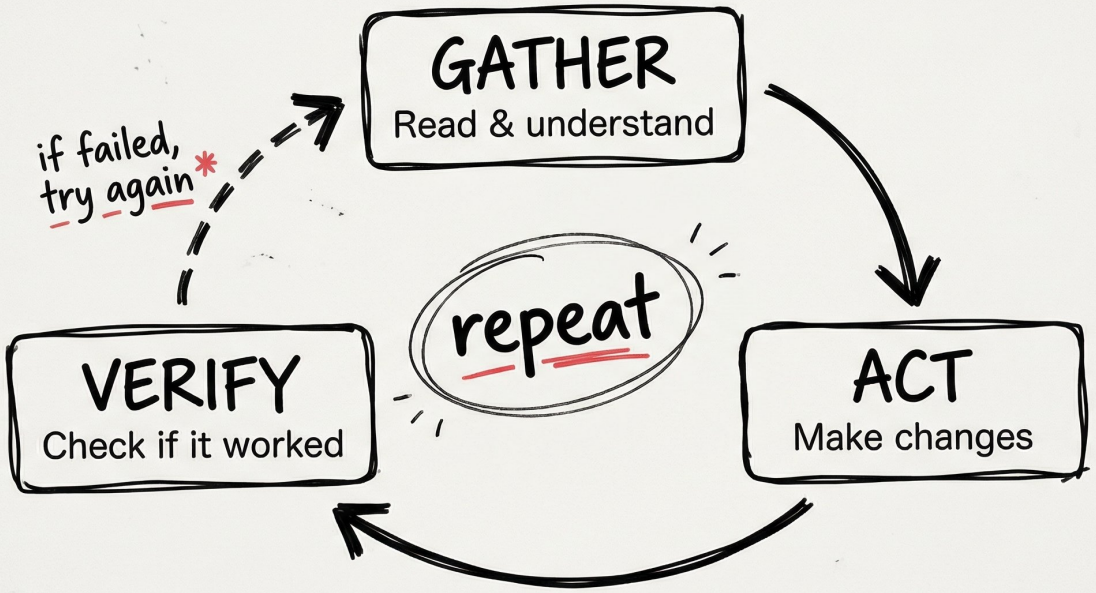
Claude Code: 4 Layers



Claude Code: 4 Layers

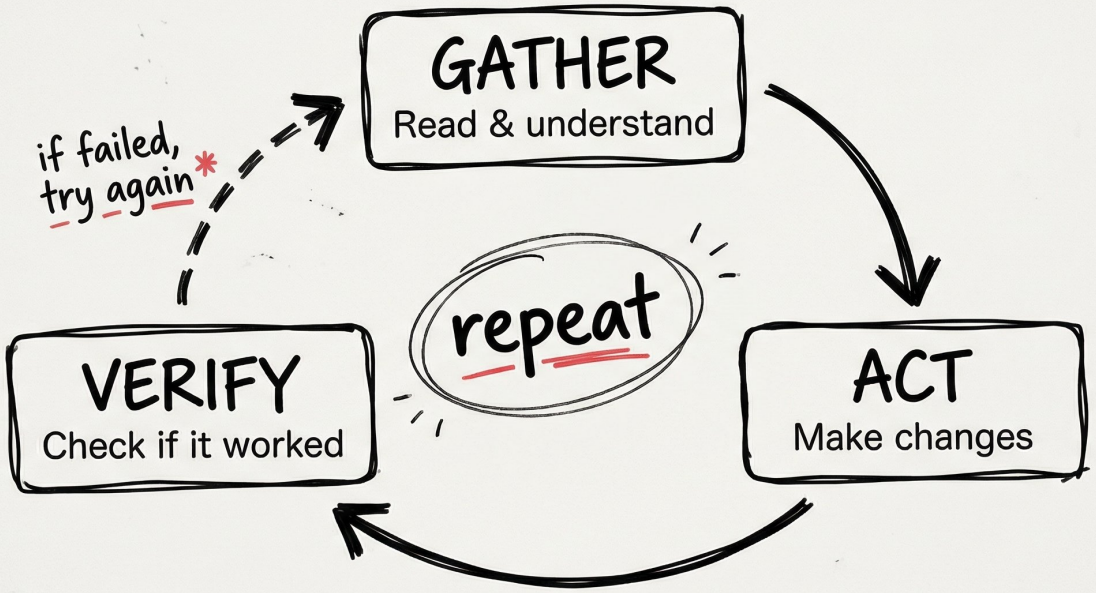


How Claude Code Thinks

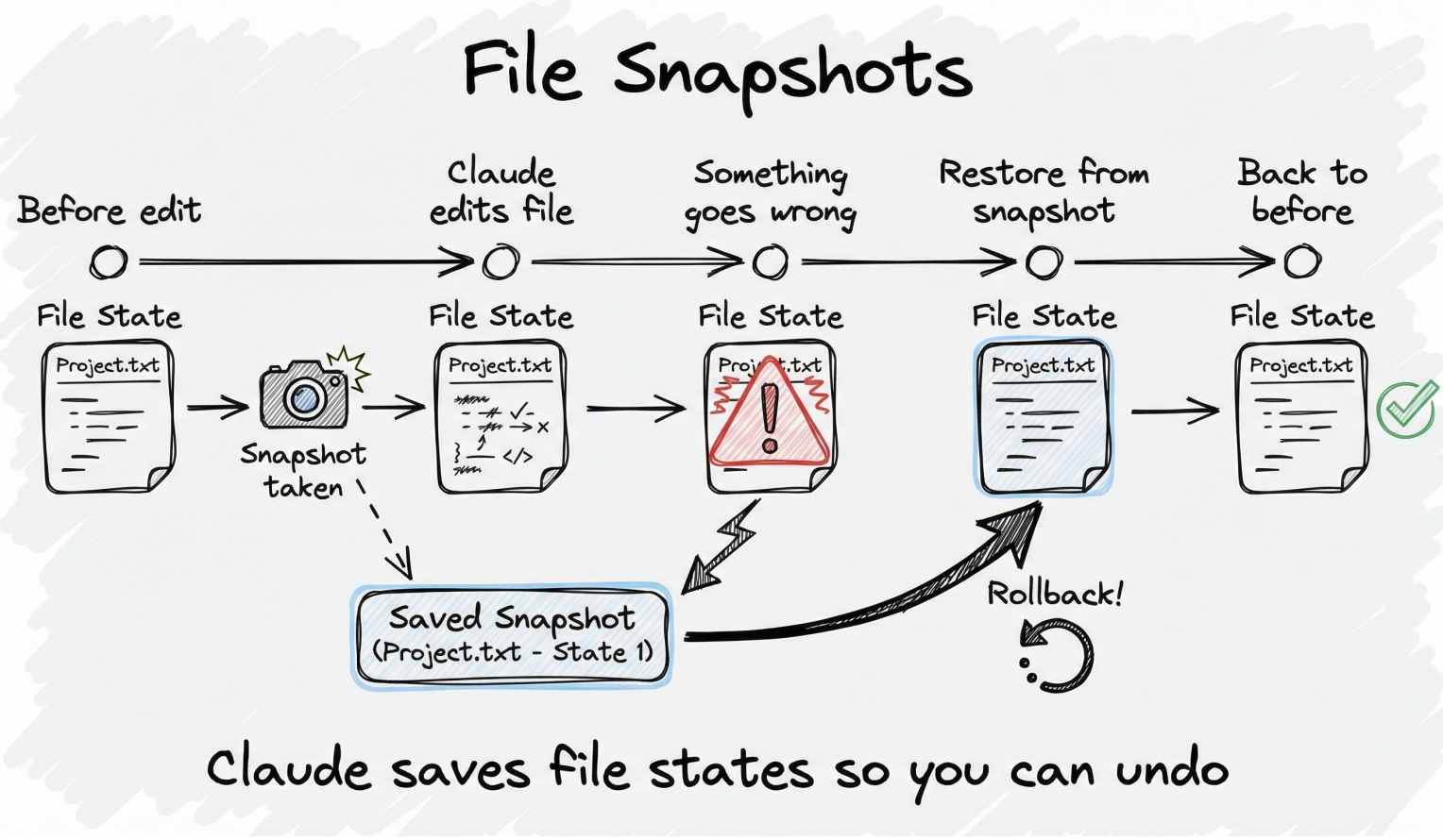


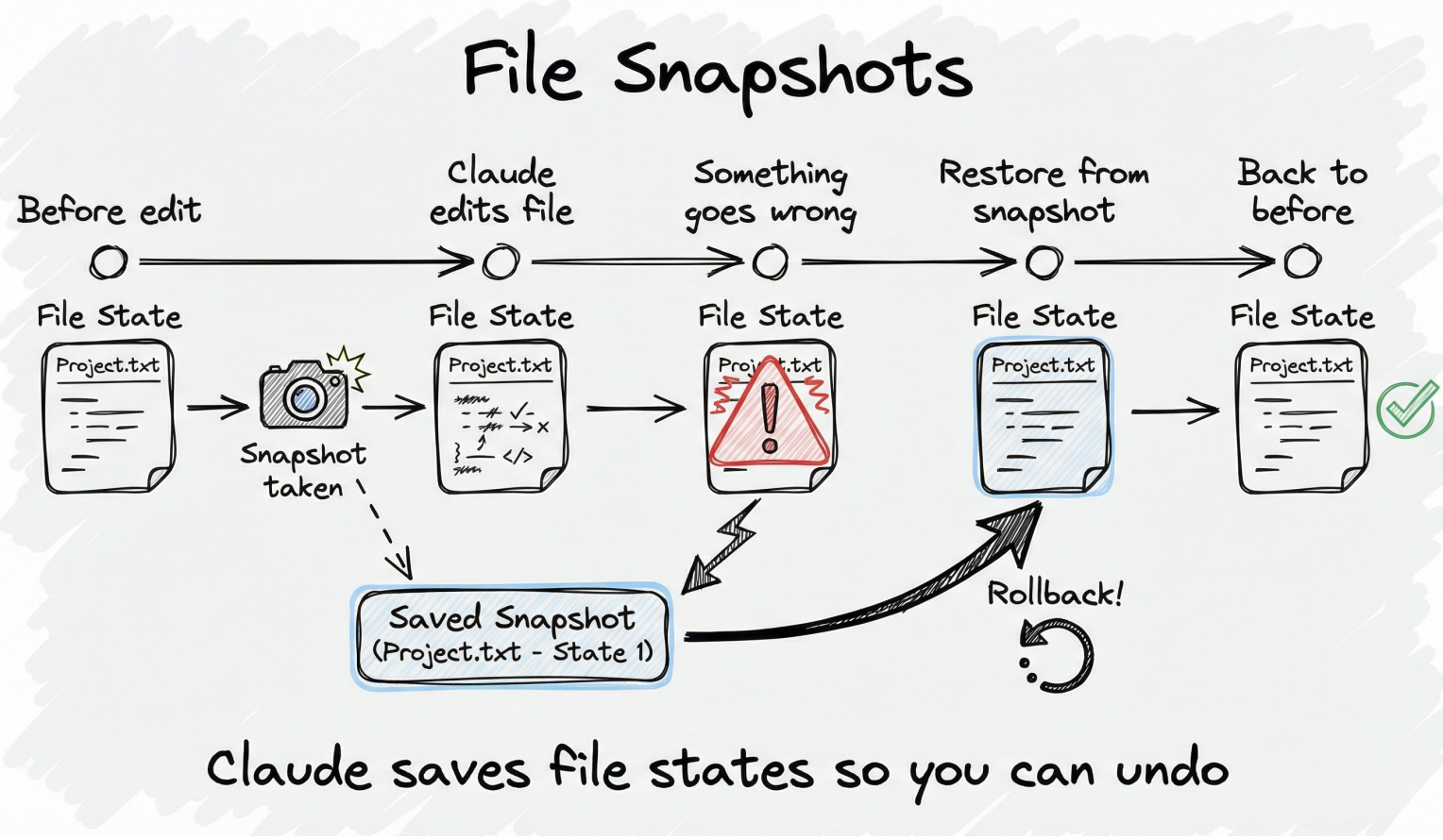
Every task follows this loop until complete ✓☺

How Claude Code Thinks

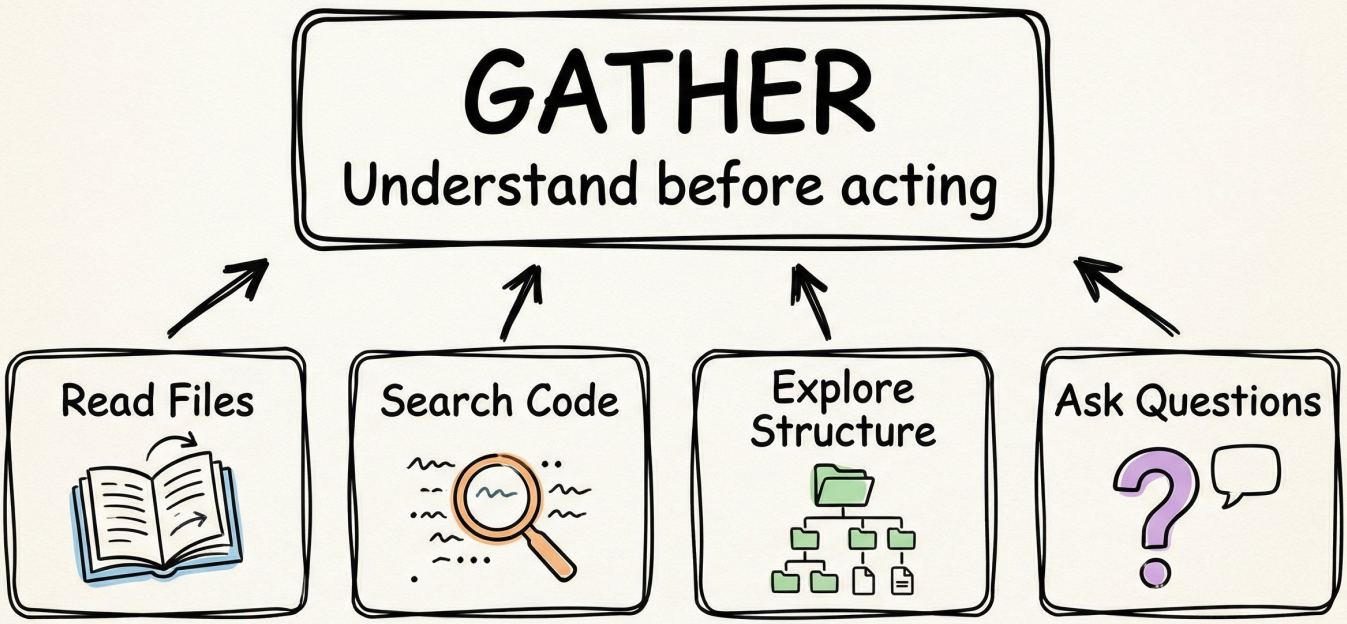


Every task follows this loop until complete ✓☺


File Snapshots
Before edit
Claude edits file
Something goes wrong
Restore from snapshot
Back to before
File State
File State
File State
File State
File State
Project.txt
Project.txt
Project.txt
Project.txt
Project.txt
Snapshot taken
Saved Snapshot
(Project.txt - State 1)
Rollback!
Claude saves file states so you can undo

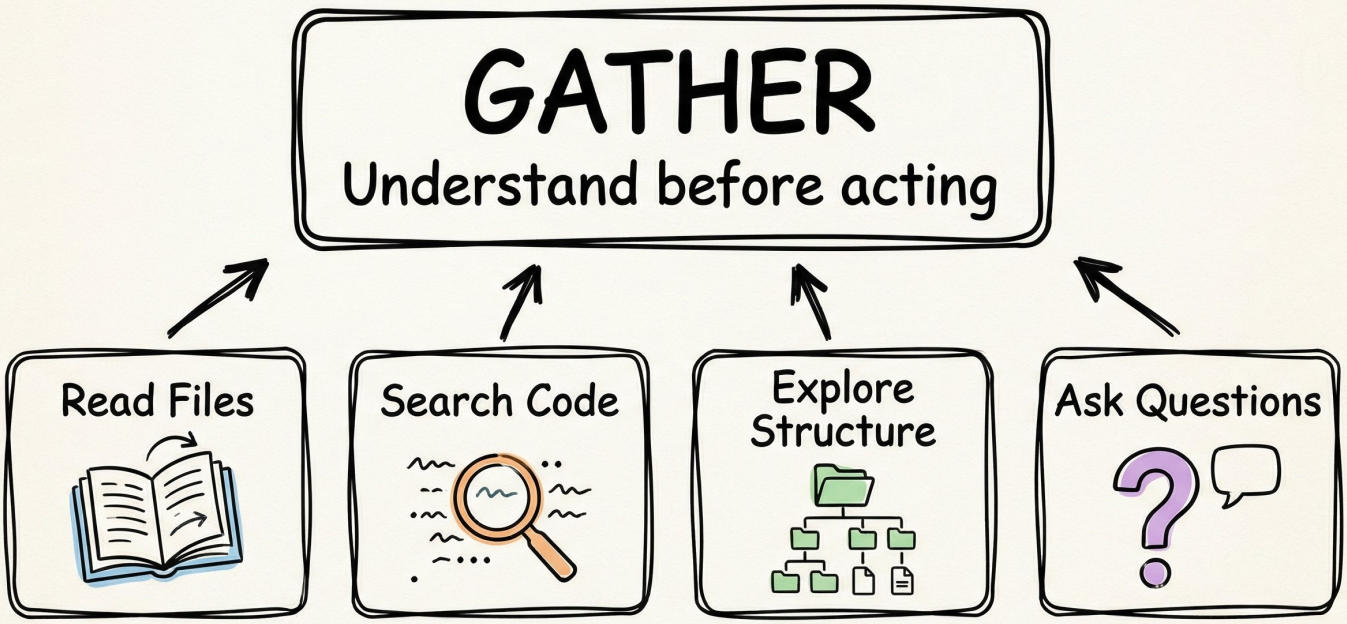

File Snapshots
Before edit
Claude edits file
Something goes wrong
Restore from snapshot
Back to before
File State
File State
File State
File State
File State
Project.txt
Project.txt
Project.txt
Project.txt
Project.txt
Snapshot taken
Saved Snapshot
(Project.txt - State 1)
Rollback!
Claude saves file states so you can undo

Phase 1: GATHER

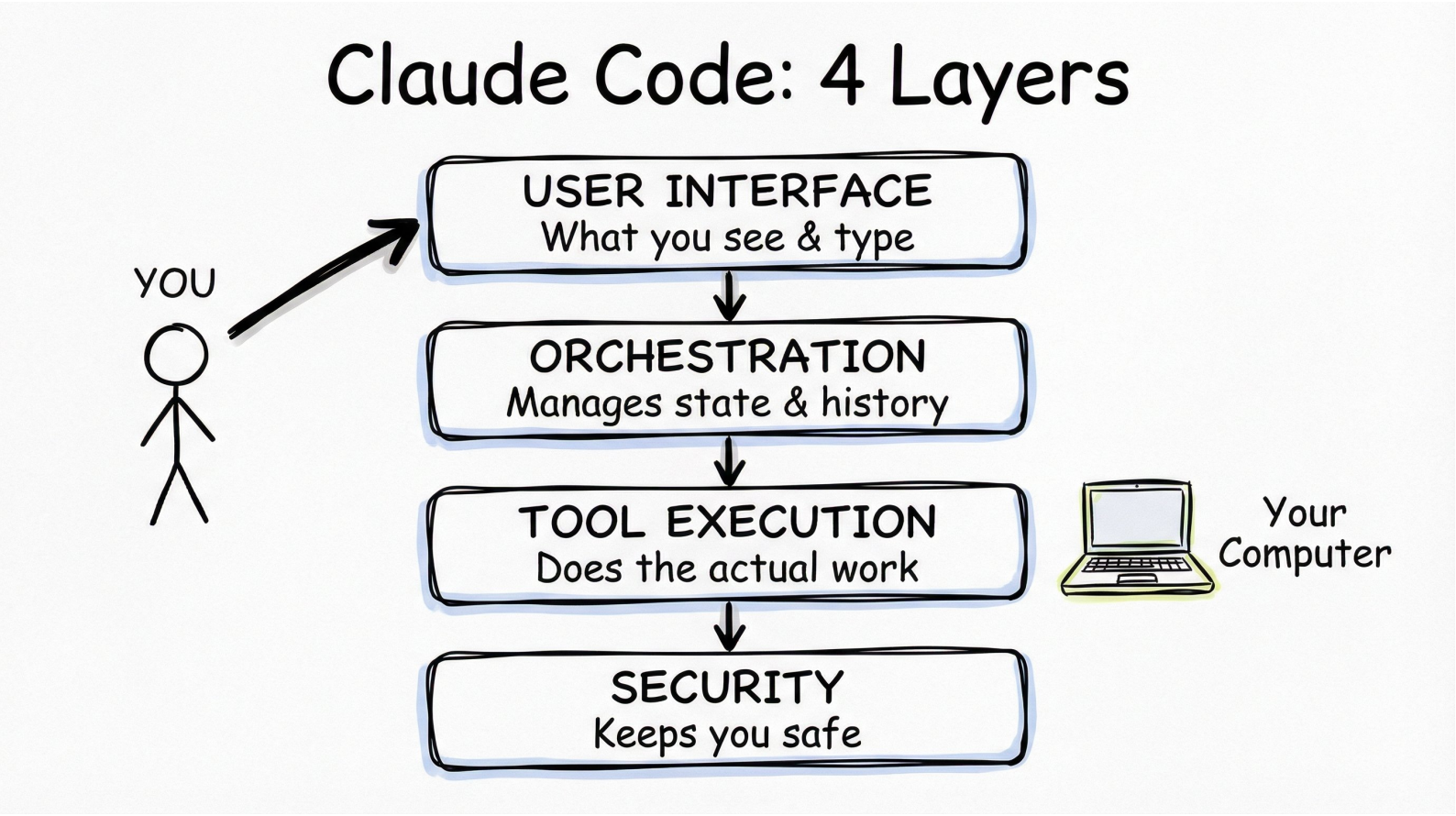


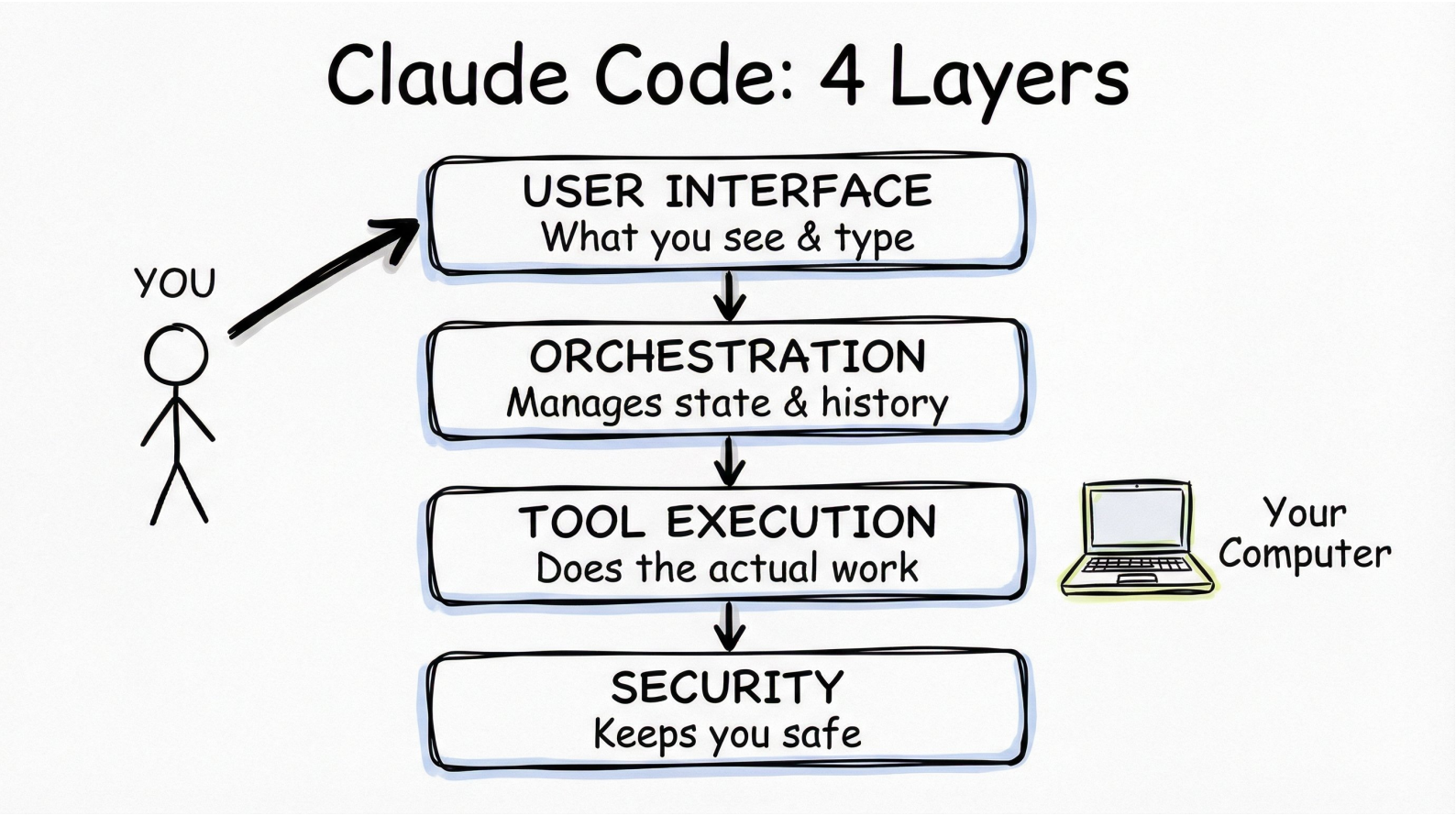
Claude never acts without understanding first

Phase 1: GATHER

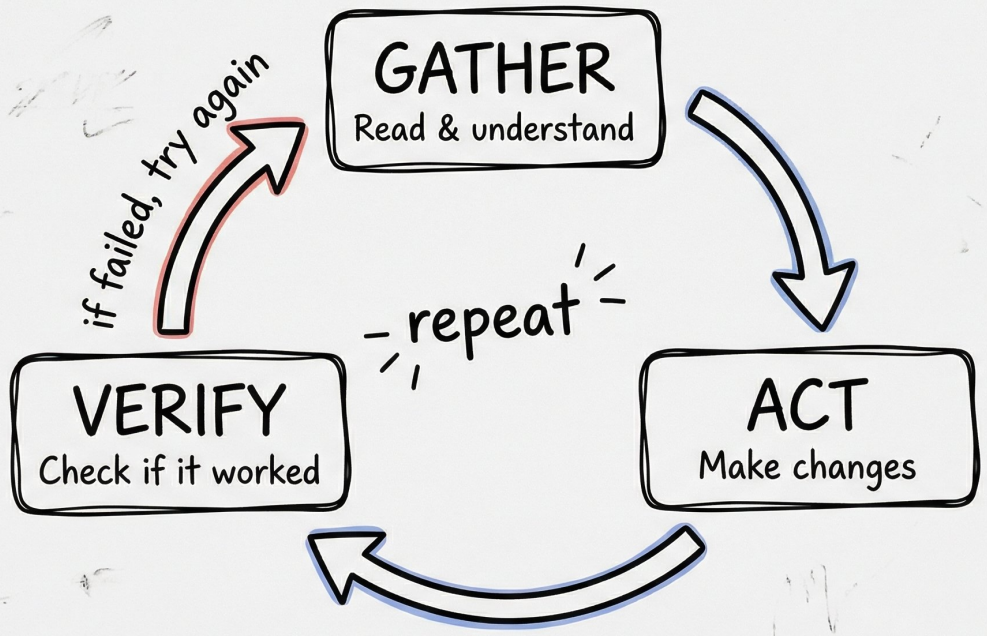


Claude never acts without understanding first


Claude Code: 4 Layers
USER INTERFACE
What you see & type
YOU
ORCHESTRATION
Manages state & history
TOOL EXECUTION
Does the actual work
Your Computer
SECURITY
Keeps you safe

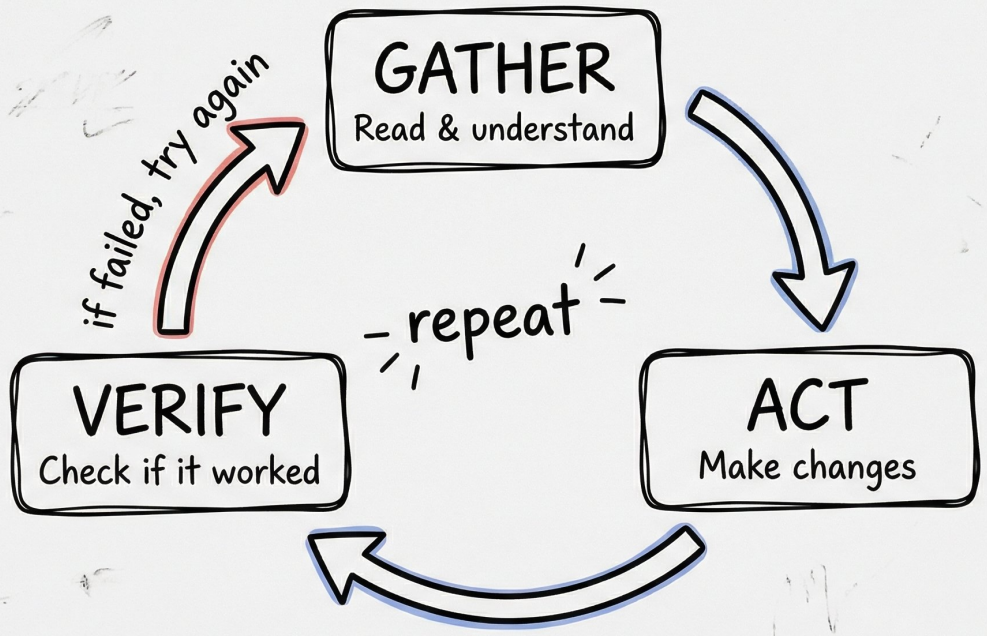

Claude Code: 4 Layers
USER INTERFACE
What you see & type
YOU
ORCHESTRATION
Manages state & history
TOOL EXECUTION
Does the actual work
Your Computer
SECURITY
Keeps you safe

HOW CLAUDE CODE THINKS



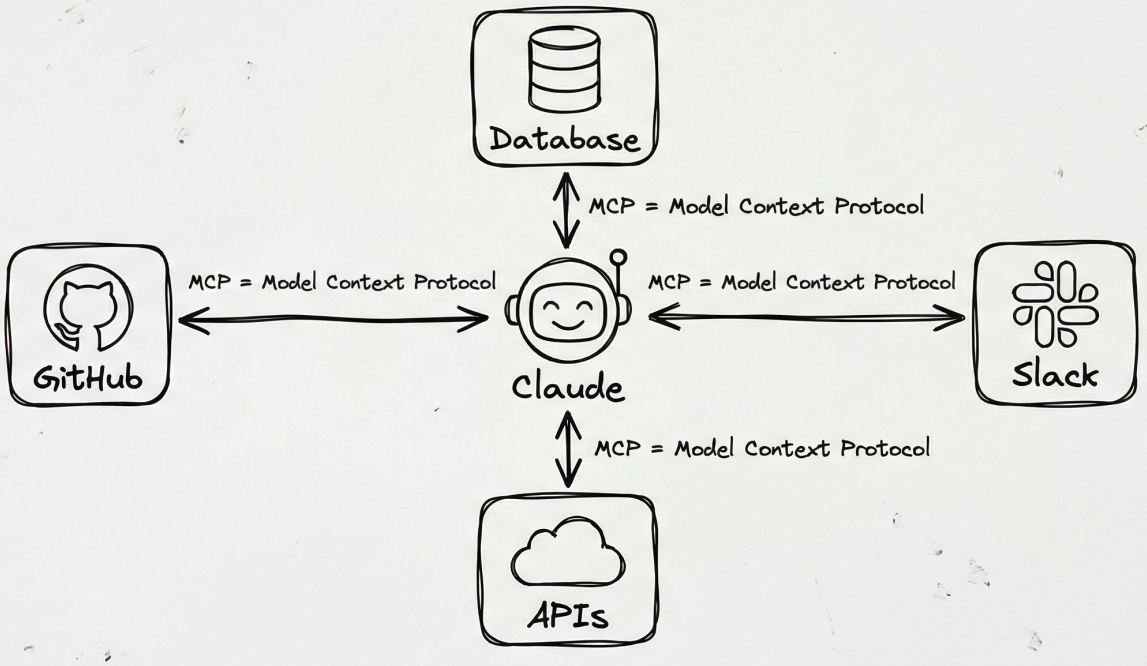
Every task follows this loop until complete

HOW CLAUDE CODE THINKS



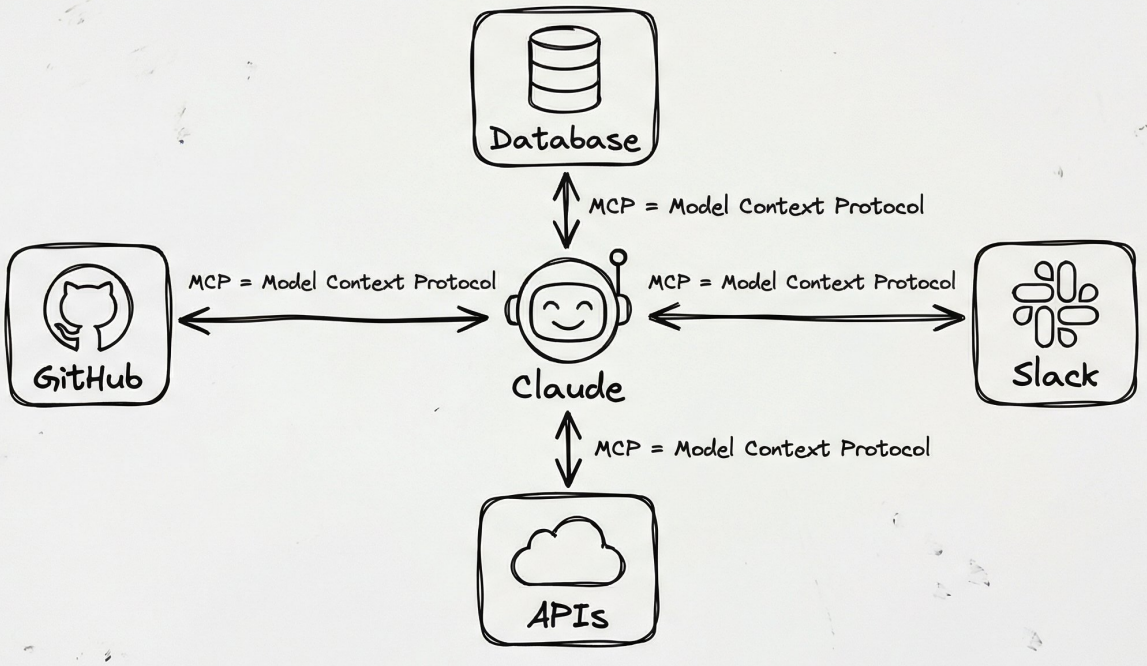
Every task follows this loop until complete

MCP: External Connections



Connect Claude to external services

MCP: External Connections



Connect Claude to external services

Common Permission Examples

Action	Decision	Why
Read any file	✓	Safe to read
Edit code files	✓	That's the job
Delete /etc	✗	System critical
npm install	?	Installs stuff
git push	?	Permanent action
rm -rf	✗	Too dangerous

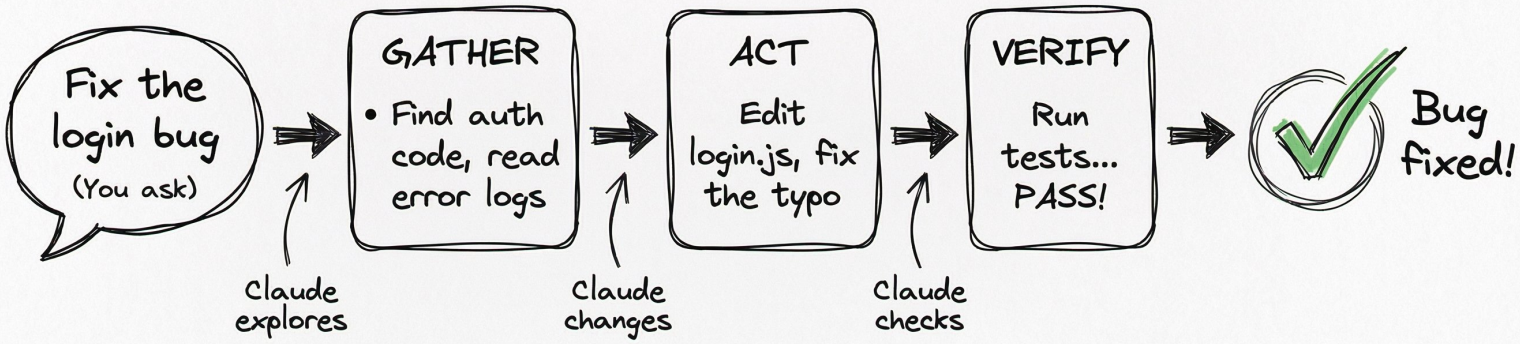
Sensible defaults, customizable by you

Common Permission Examples

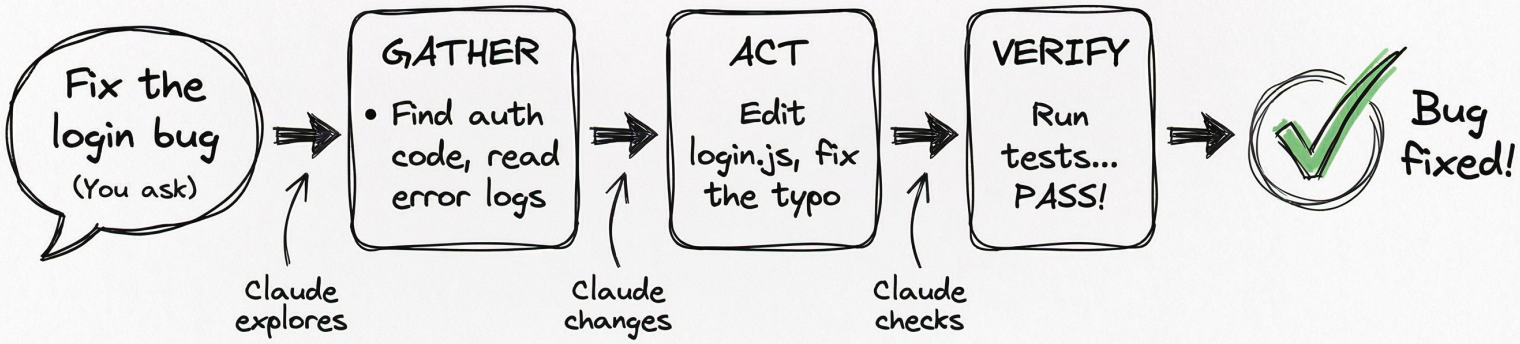
Action	Decision	Why
Read any file	✓	Safe to read
Edit code files	✓	That's the job
Delete /etc	✗	System critical
npm install	?	Installs stuff
git push	?	Permanent action
rm -rf	✗	Too dangerous

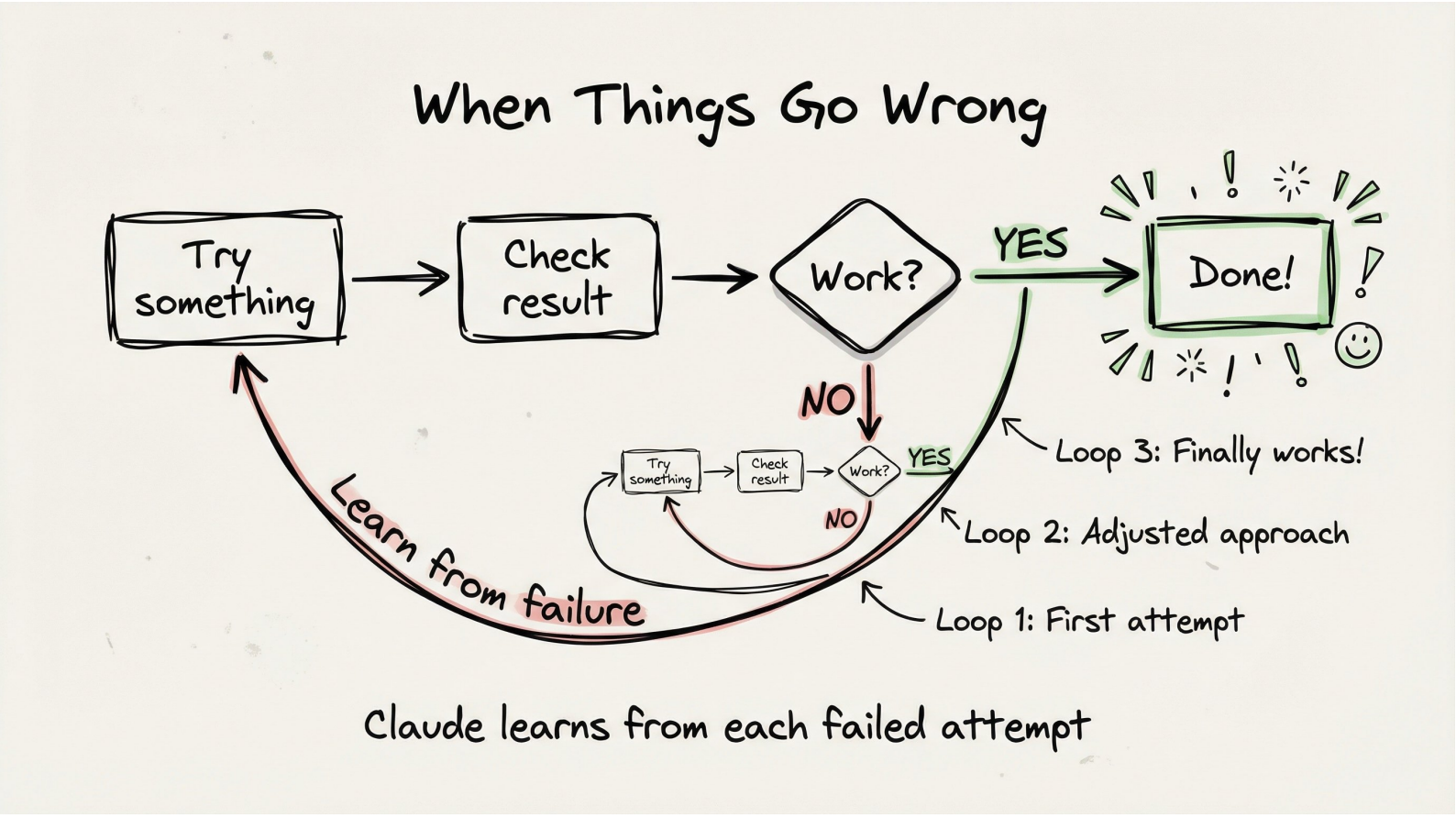
Sensible defaults, customizable by you

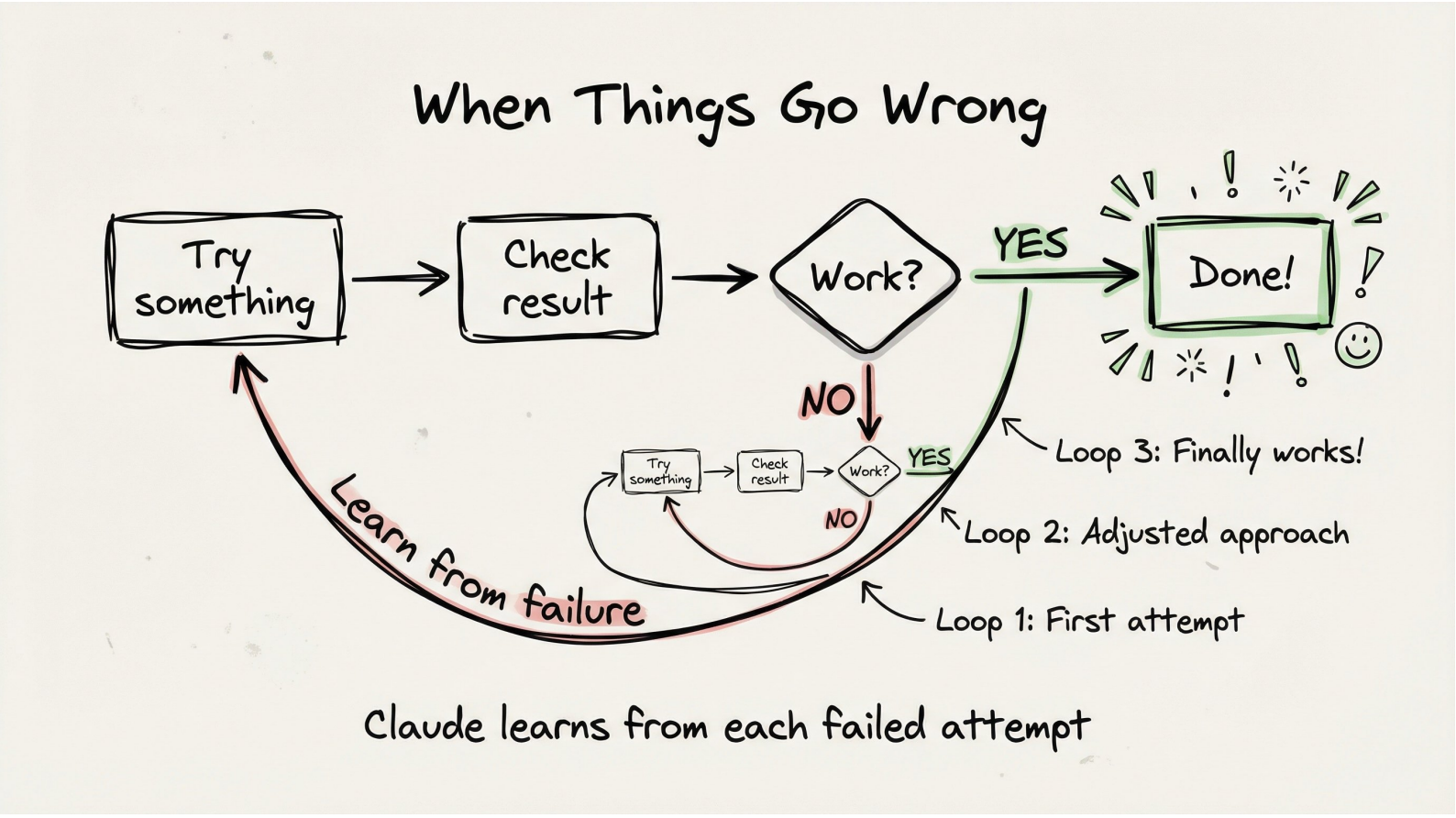
Example: Fixing a Bug



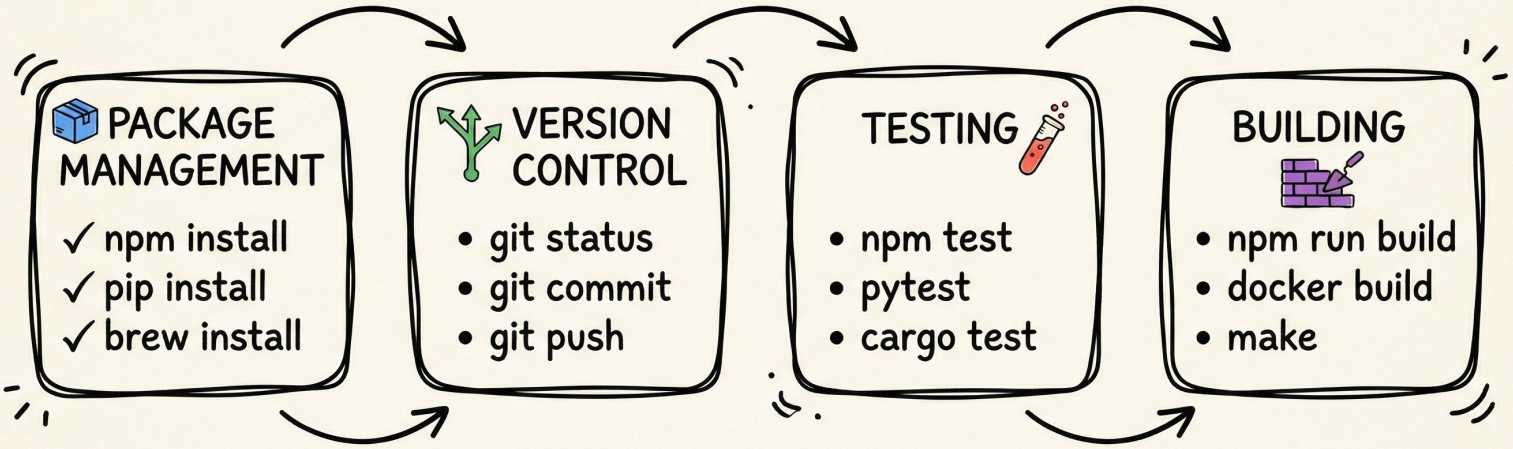
Example: Fixing a Bug




When Things Go Wrong
Try something
Check result
Work?
YES
Done!
NO
Try something
Check result
Work?
YES
NO
Learn from failure
Loop 3: Finally works!
Loop 2: Adjusted approach
Loop 1: First attempt
Claude learns from each failed attempt

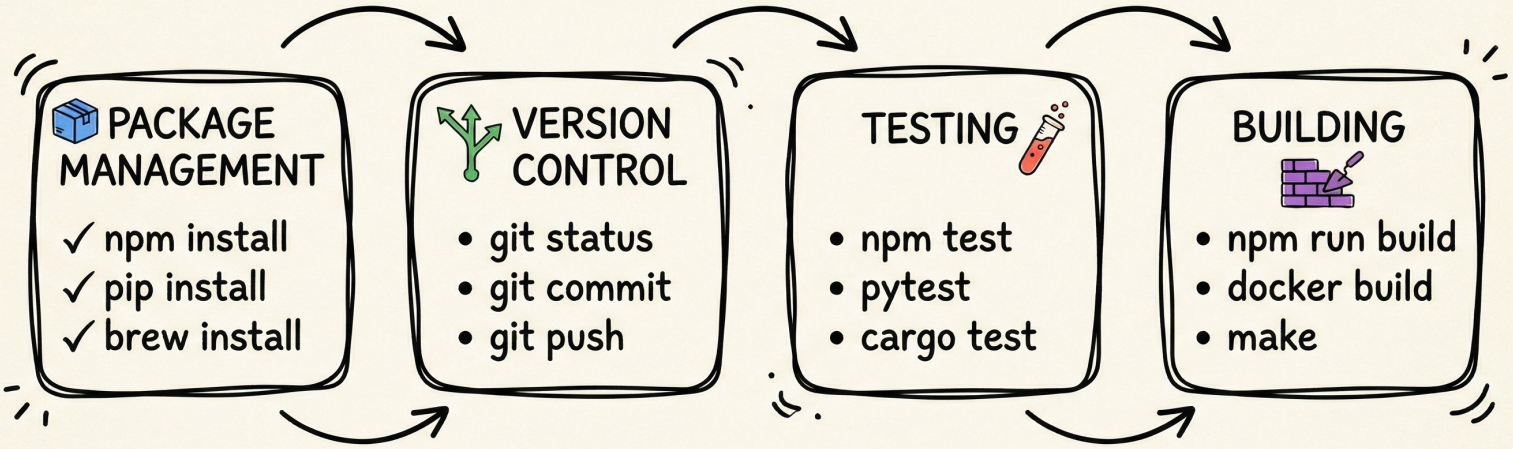

When Things Go Wrong
Try something
Check result
Work?
YES
Done!
NO
Try something
Check result
Work?
YES
NO
Learn from failure
Loop 3: Finally works!
Loop 2: Adjusted approach
Loop 1: First attempt
Claude learns from each failed attempt

Common Bash Patterns



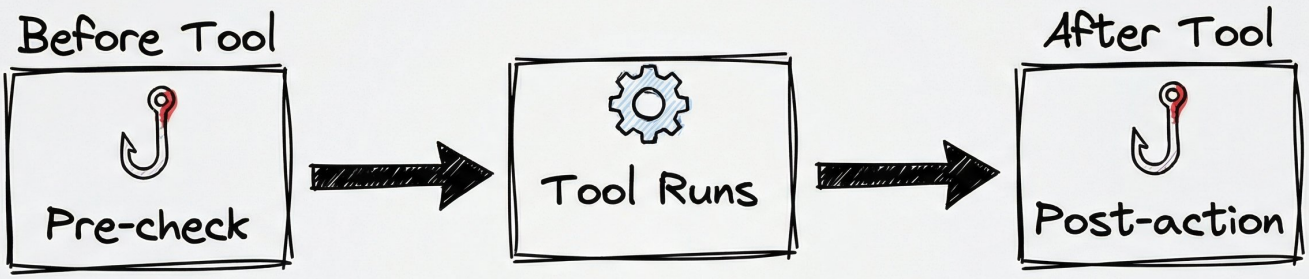
These are the commands Claude runs most often

Common Bash Patterns



These are the commands Claude runs most often

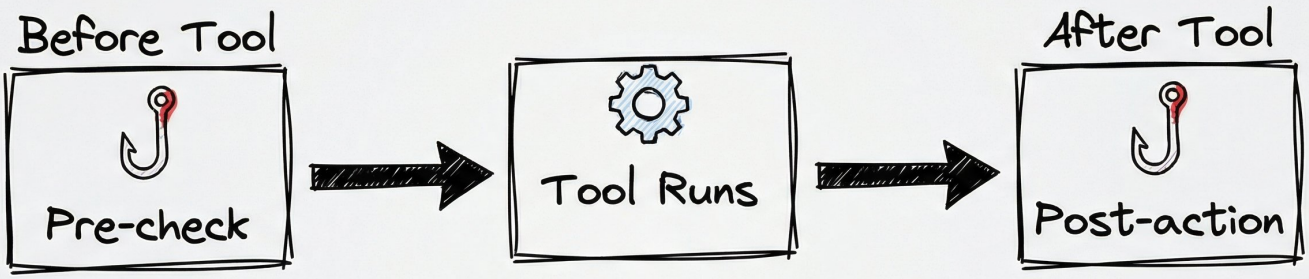
Hooks: Automated Actions



Examples:
Before write → Auto-lint
After commit → Run tests
Before bash → Security check

Automated actions triggered by events

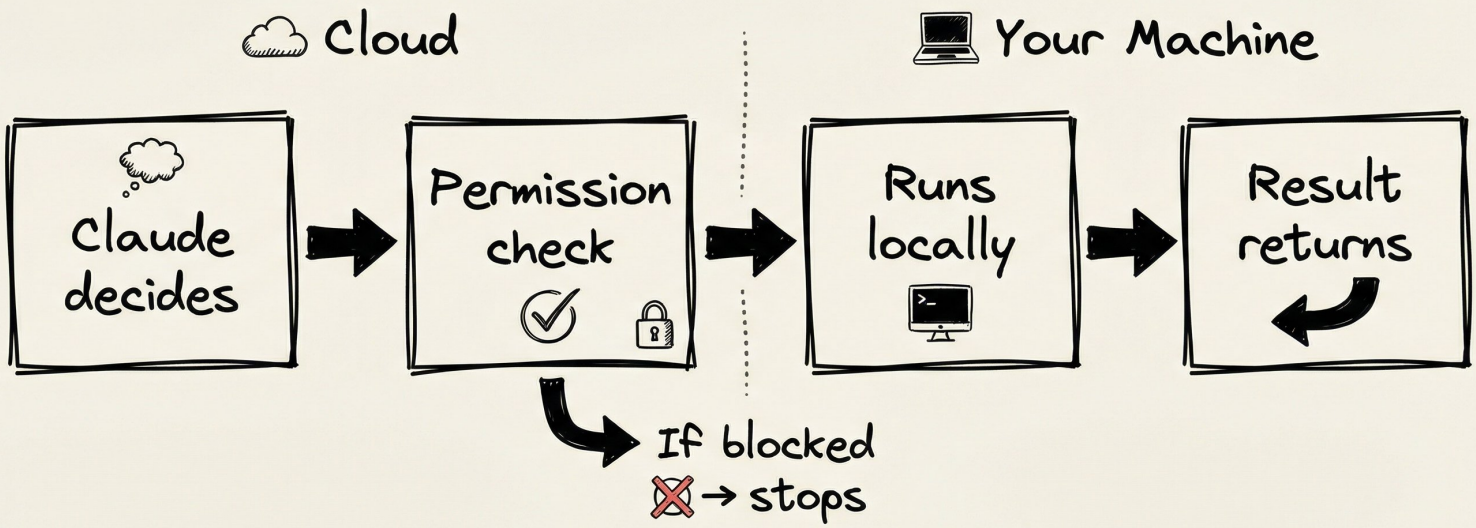
Hooks: Automated Actions



Examples:
Before write → Auto-lint
After commit → Run tests
Before bash → Security check

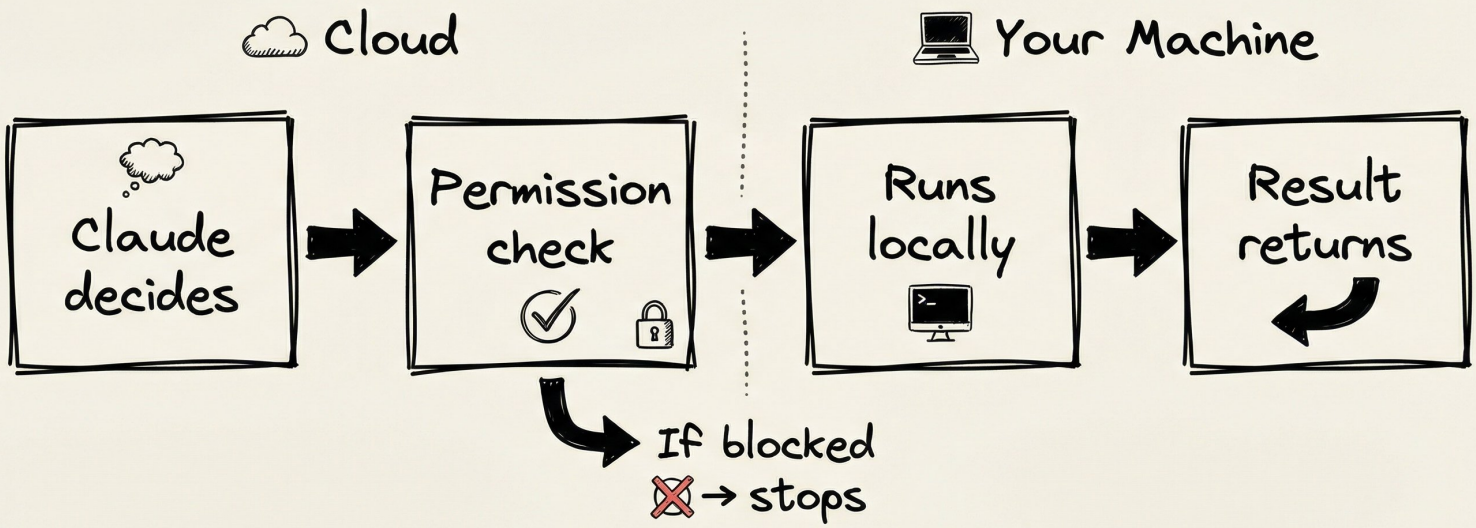
Automated actions triggered by events

From Request to Result



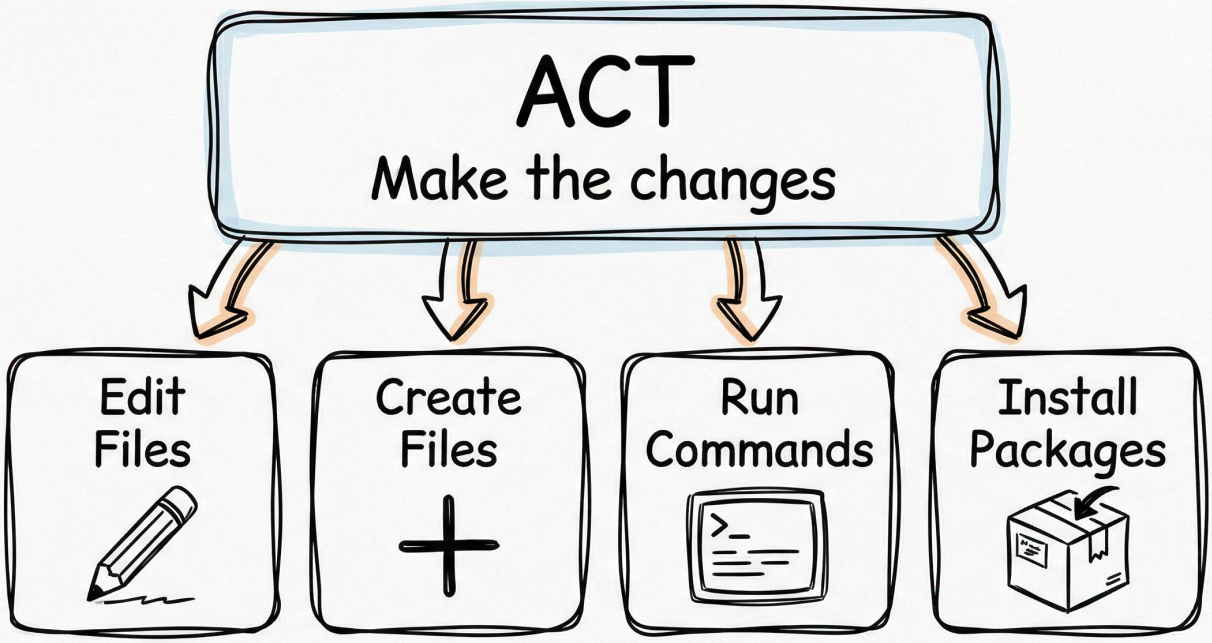
You're always in control

From Request to Result



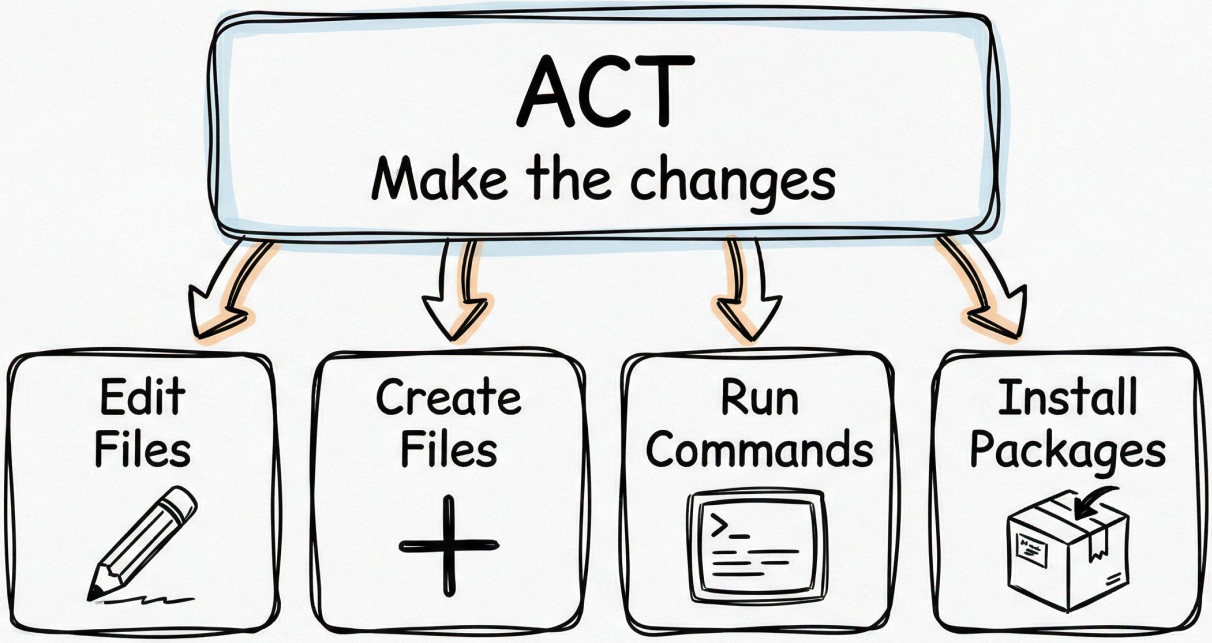
You're always in control

Phase 2: ACT



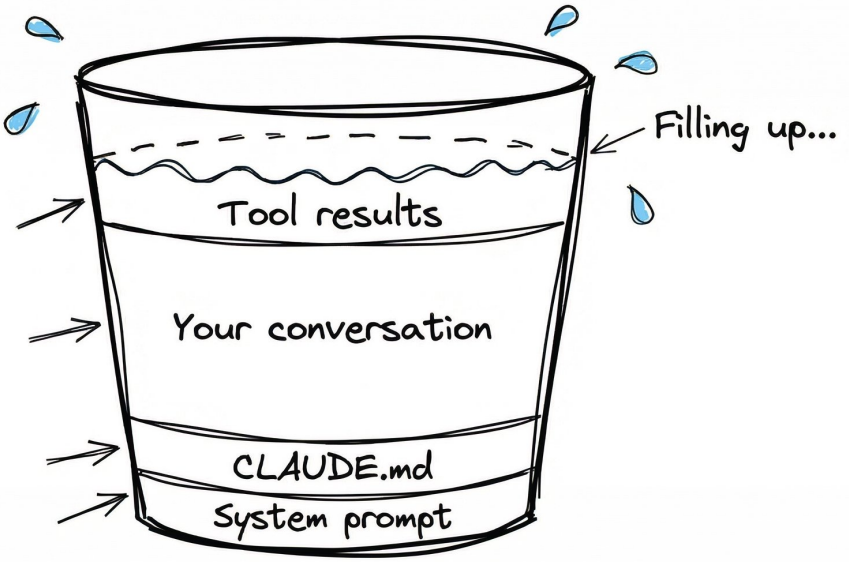
This is where things actually change

Phase 2: ACT



This is where things actually change

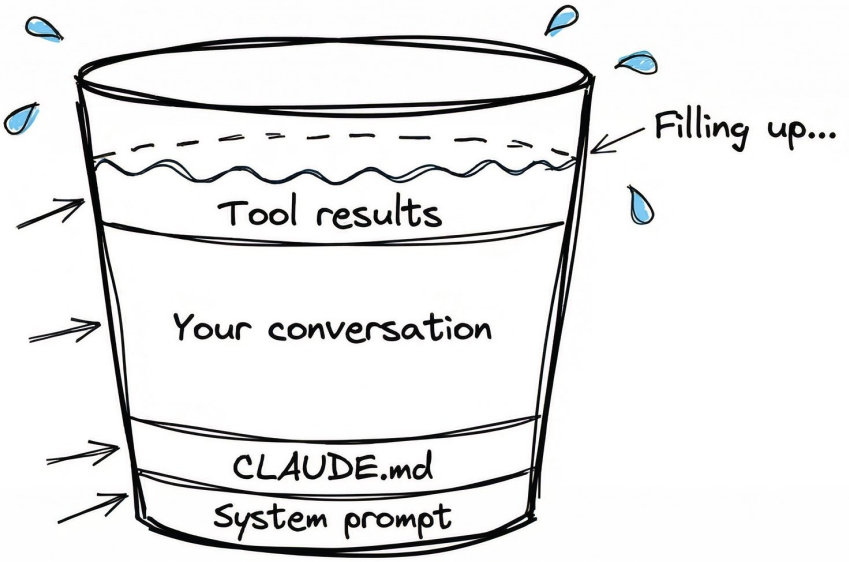
The Context Window



Claude's Working Memory

Like RAM - limited space, must manage carefully

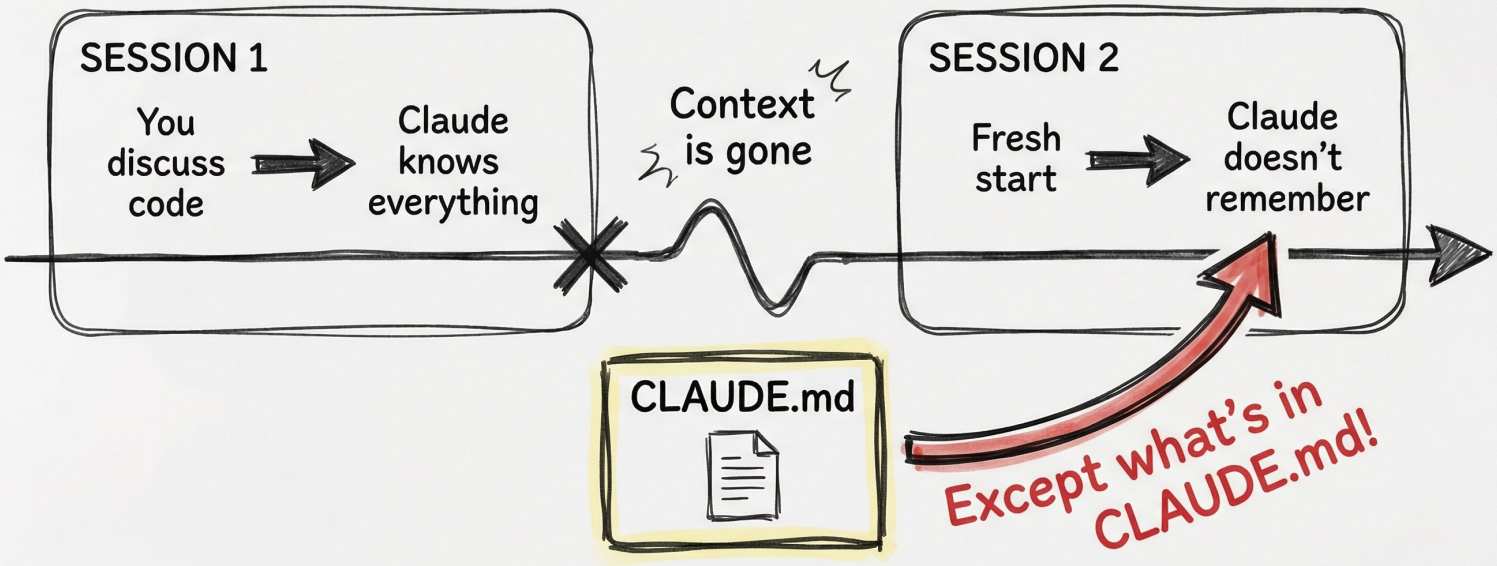
The Context Window



Claude's Working Memory

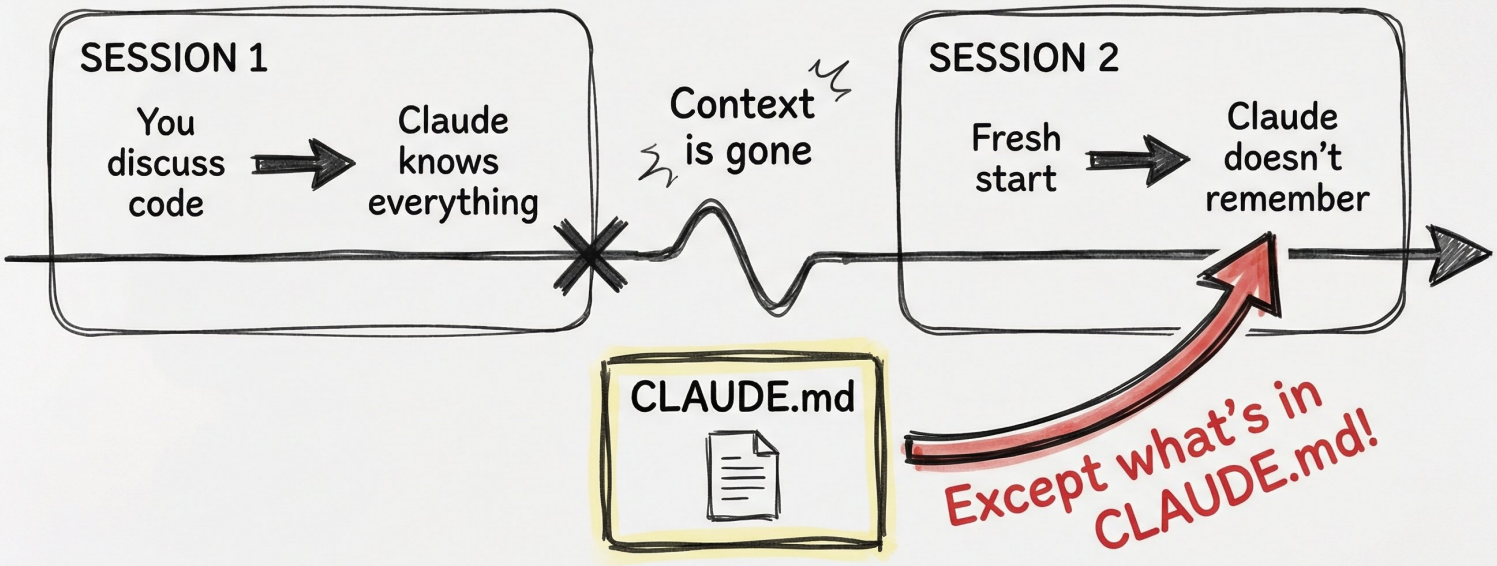
Like RAM - limited space, must manage carefully

Why Claude Forgets



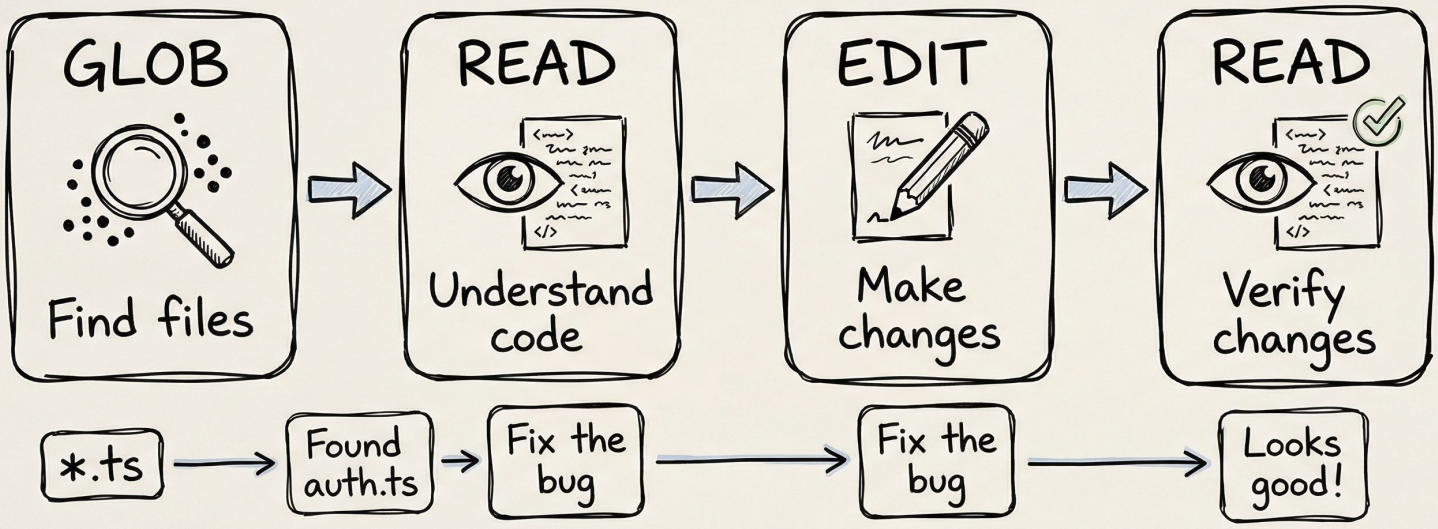
CLAUDE.md is your project's persistent memory

Why Claude Forgets



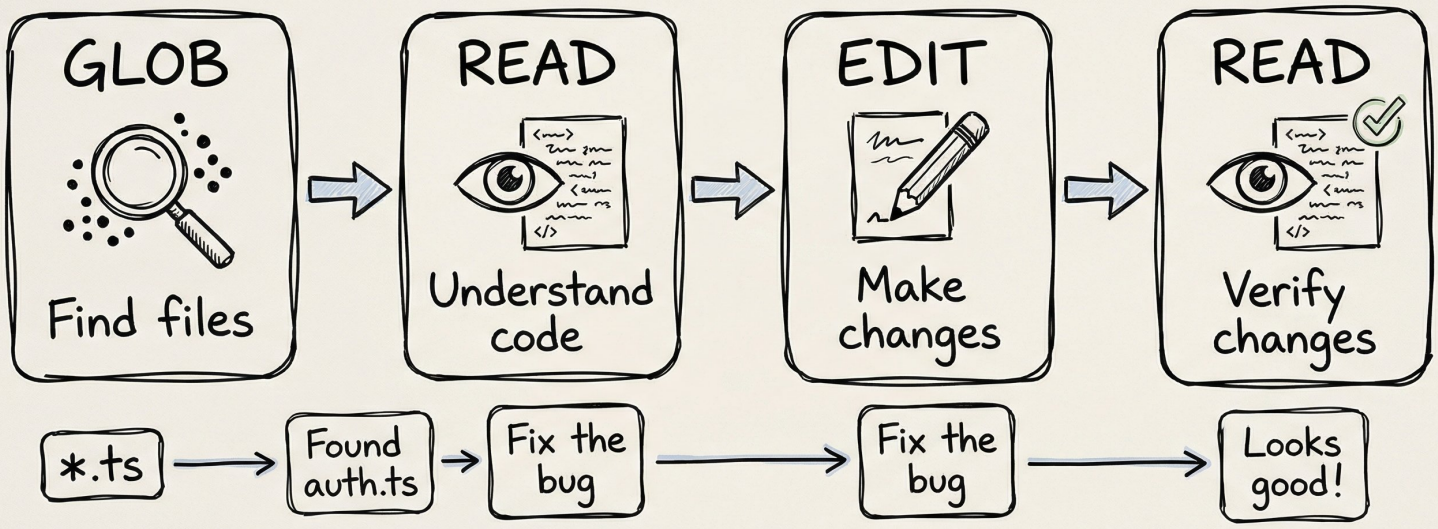
CLAUDE.md is your project's persistent memory

Typical File Workflow

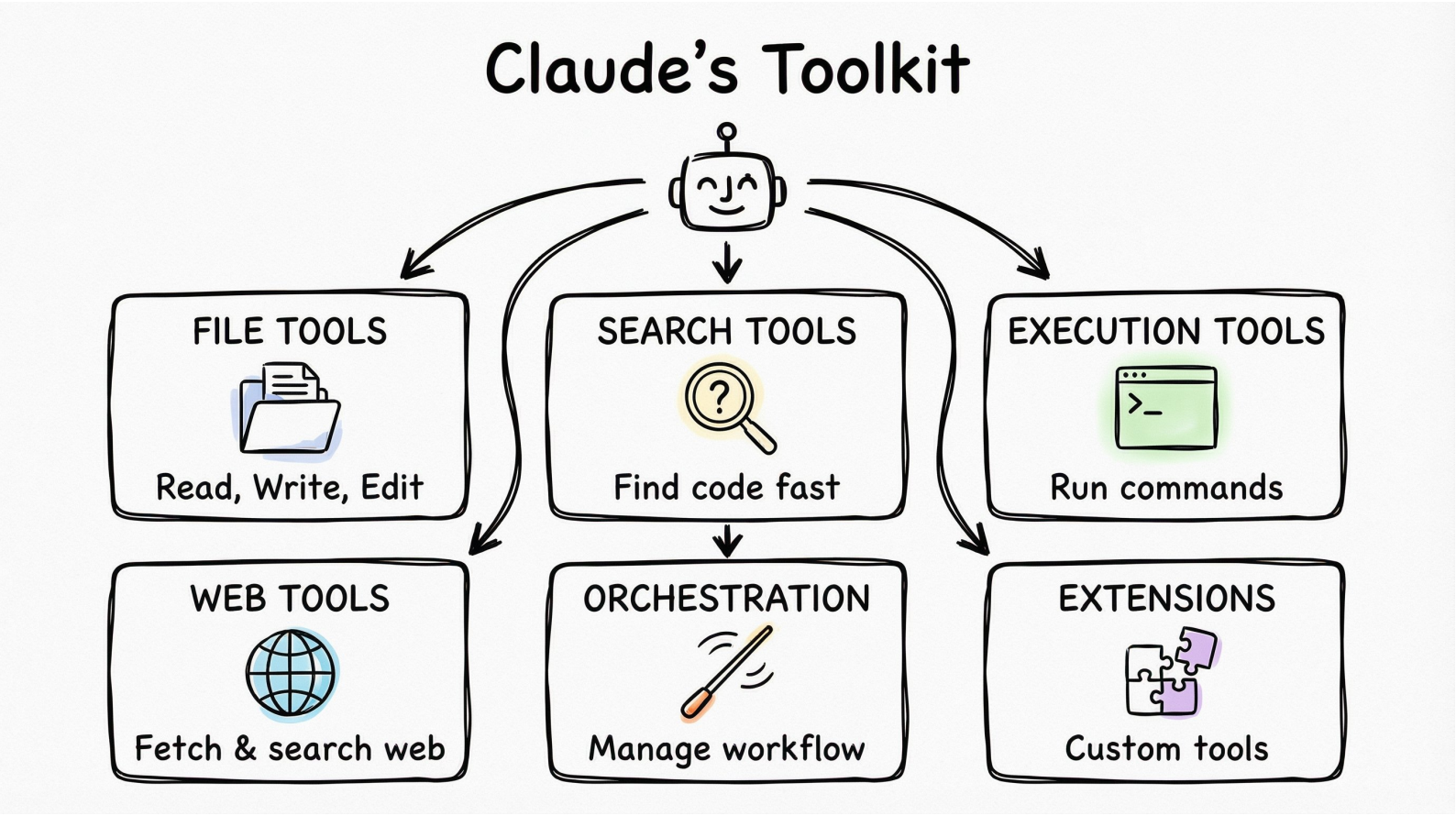


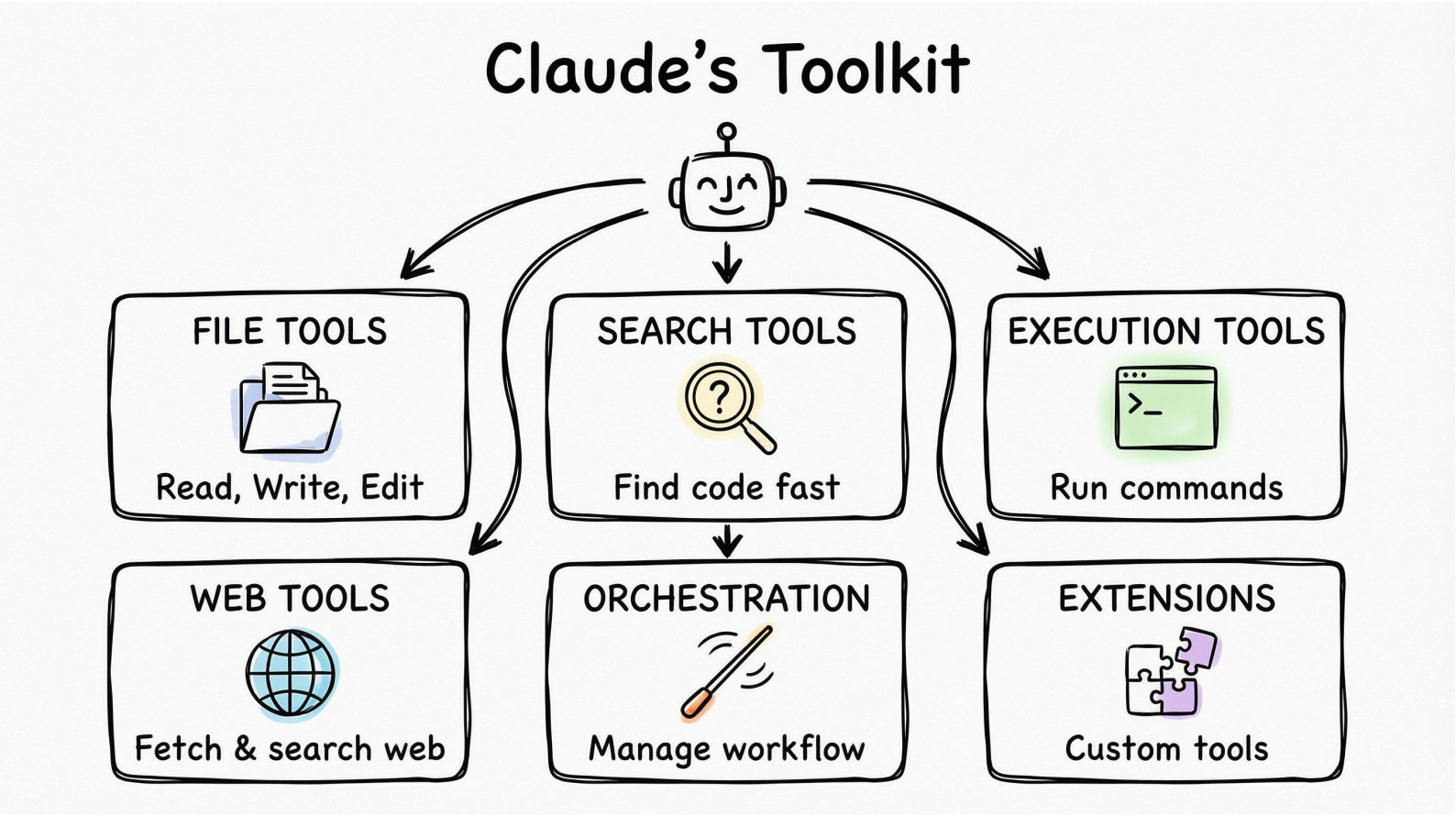
This is how Claude approaches most file tasks

Typical File Workflow



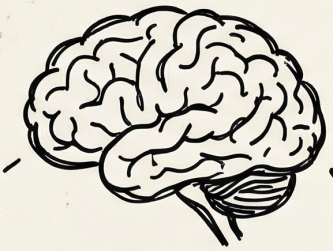
This is how Claude approaches most file tasks

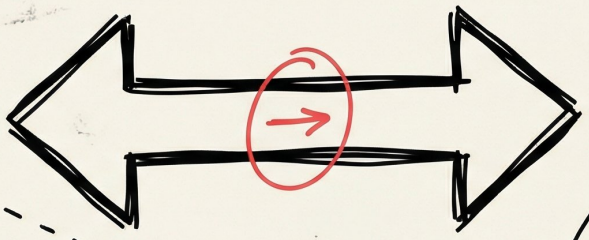

Claude's Toolkit
FILE TOOLS
Read, Write, Edit
SEARCH TOOLS
Find code fast
EXECUTION TOOLS
Run commands
WEB TOOLS
Fetch & search web
ORCHESTRATION
Manage workflow
EXTENSIONS
Custom tools


Claude's Toolkit
FILE TOOLS
Read, Write, Edit
SEARCH TOOLS
Find code fast
EXECUTION TOOLS
Run commands
WEB TOOLS
Fetch & search web
ORCHESTRATION
Manage workflow
EXTENSIONS
Custom tools

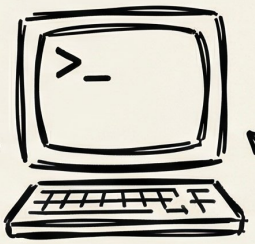
Radical Simplicity

MODEL





CLI



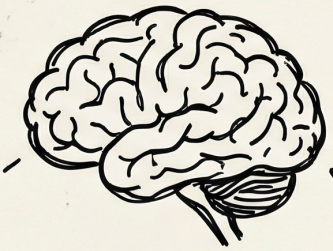
- Does ALL the thinking
- Decides what to do
- Reasons about problems

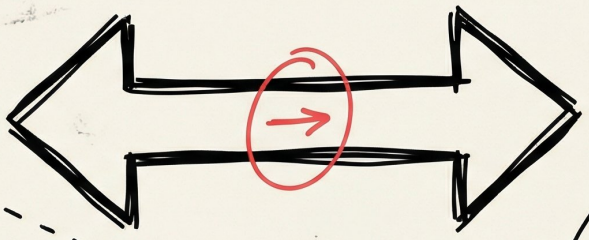
- Just executes
- Handles files
- Runs commands

90% of Claude Code was built BY Claude Code

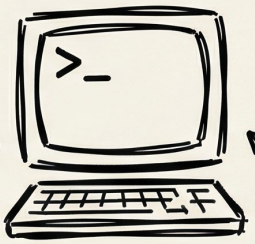
Radical Simplicity

MODEL





CLI

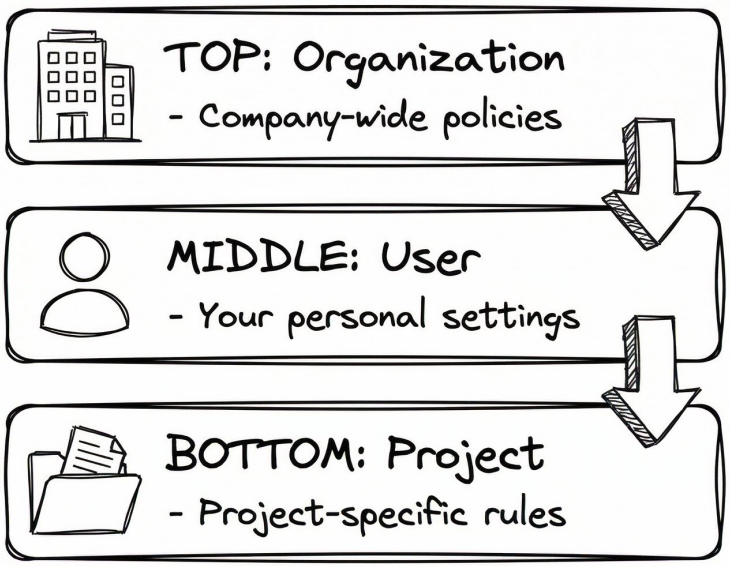


- Does ALL the thinking
- Decides what to do
- Reasons about problems

- Just executes
- Handles files
- Runs commands

90% of Claude Code was built BY Claude Code

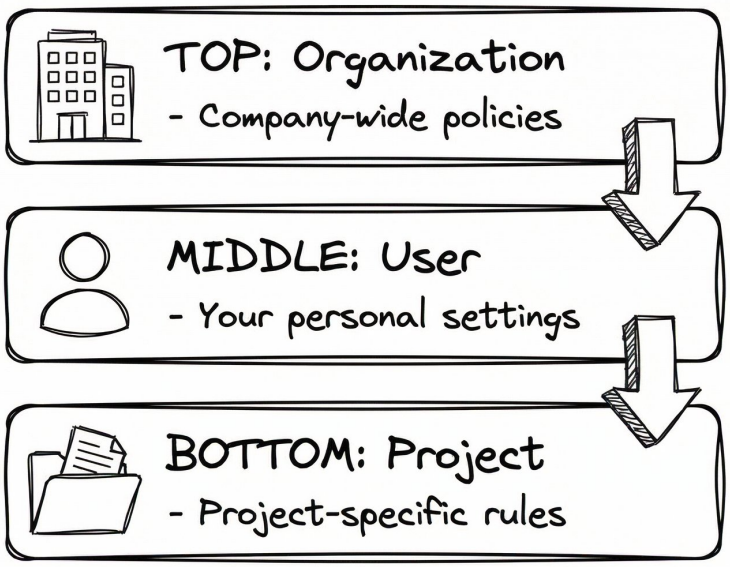
Permission Hierarchy



Note: "Most restrictive wins!"

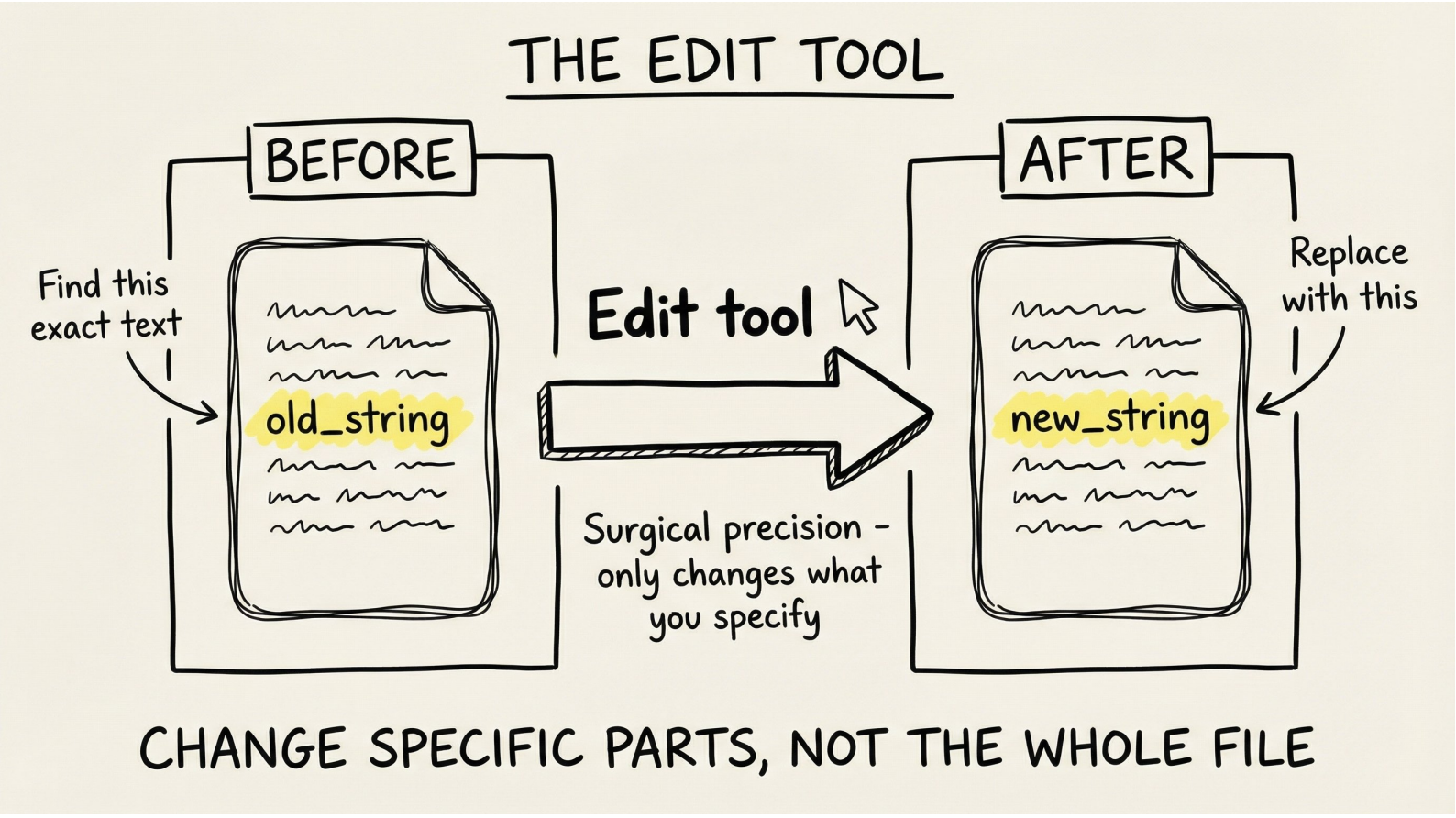
Example: "Org says no, User says yes → NO

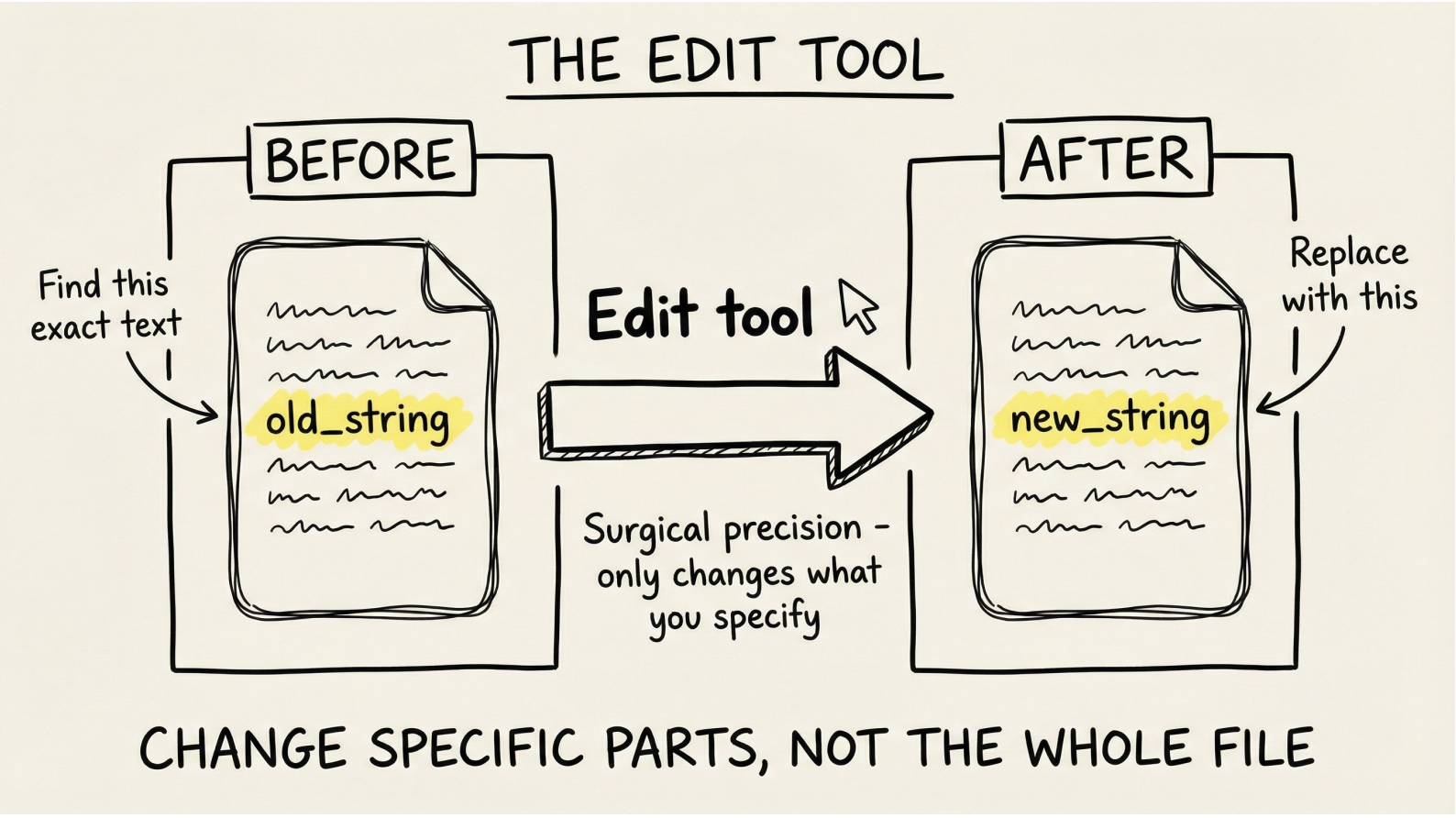
Permission Hierarchy



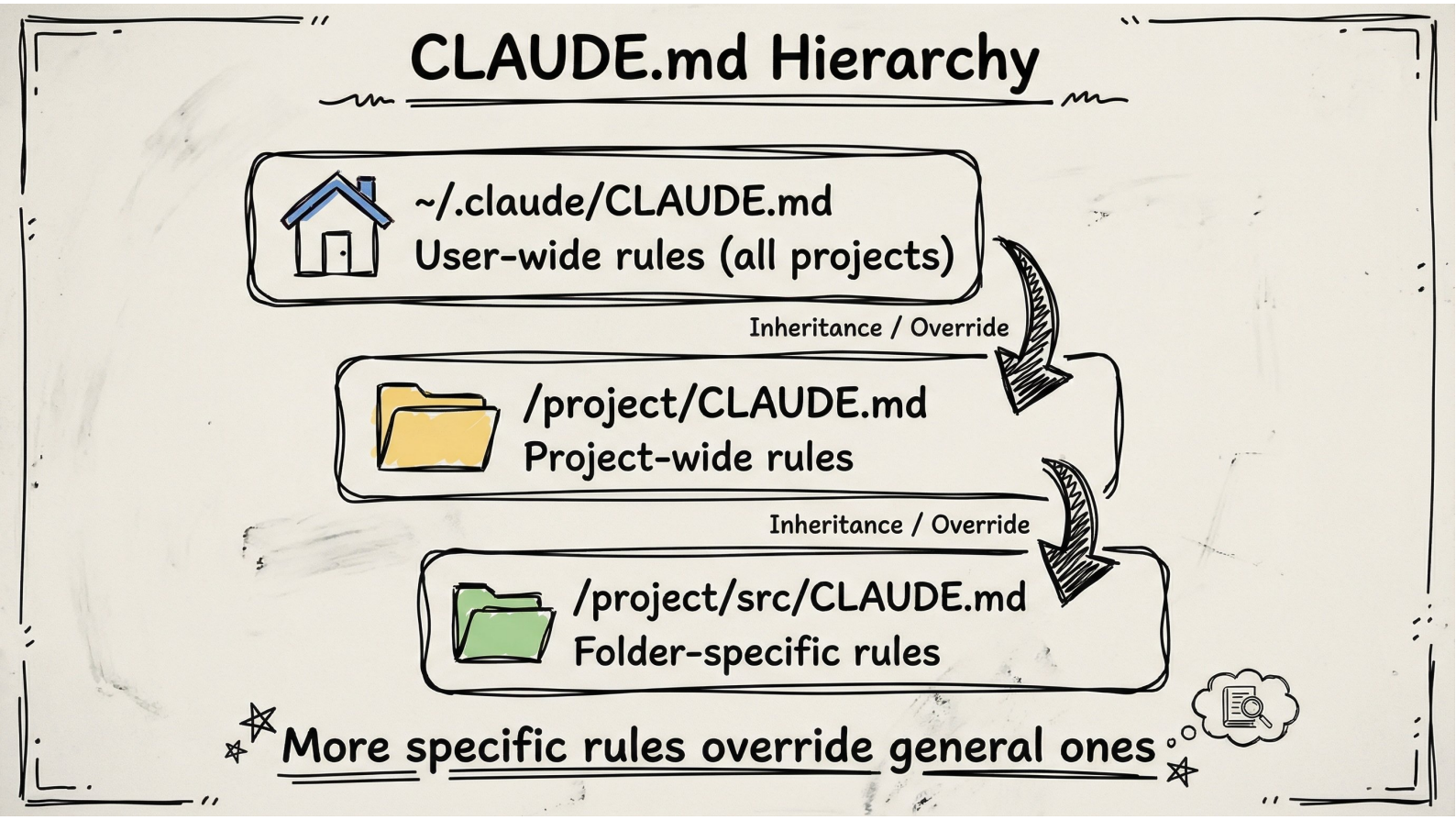
Note: "Most restrictive wins!"

Example: "Org says no, User says yes → NO


THE EDIT TOOL
BEFORE
AFTER
Find this exact text
old_string
Edit tool
Surgical precision - only changes what you specify
new_string
Replace with this
CHANGE SPECIFIC PARTS, NOT THE WHOLE FILE

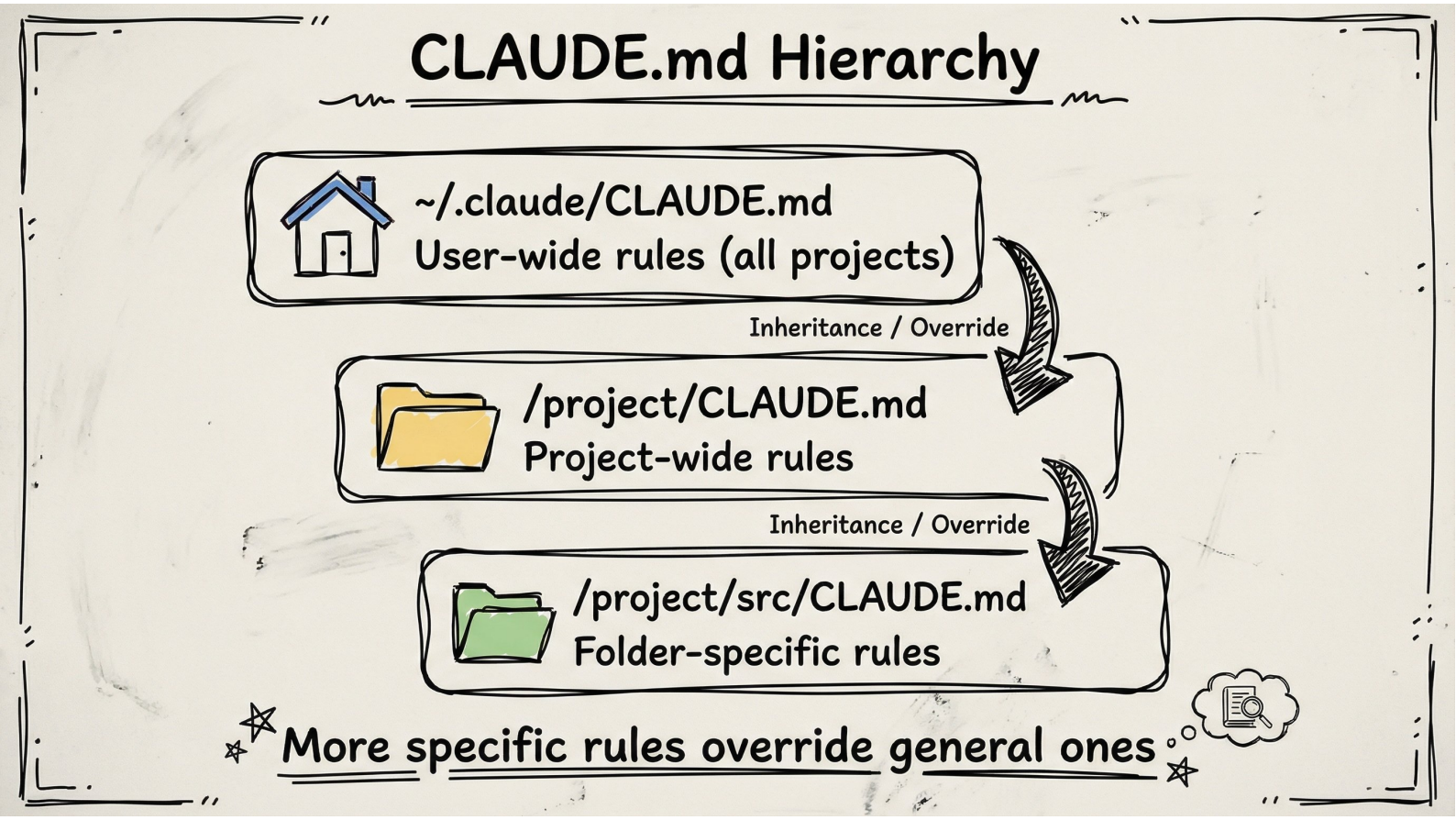

THE EDIT TOOL
BEFORE
AFTER
Find this exact text
old_string
Edit tool
Surgical precision - only changes what you specify
new_string
Replace with this
CHANGE SPECIFIC PARTS, NOT THE WHOLE FILE

CLAUDE.md Hierarchy



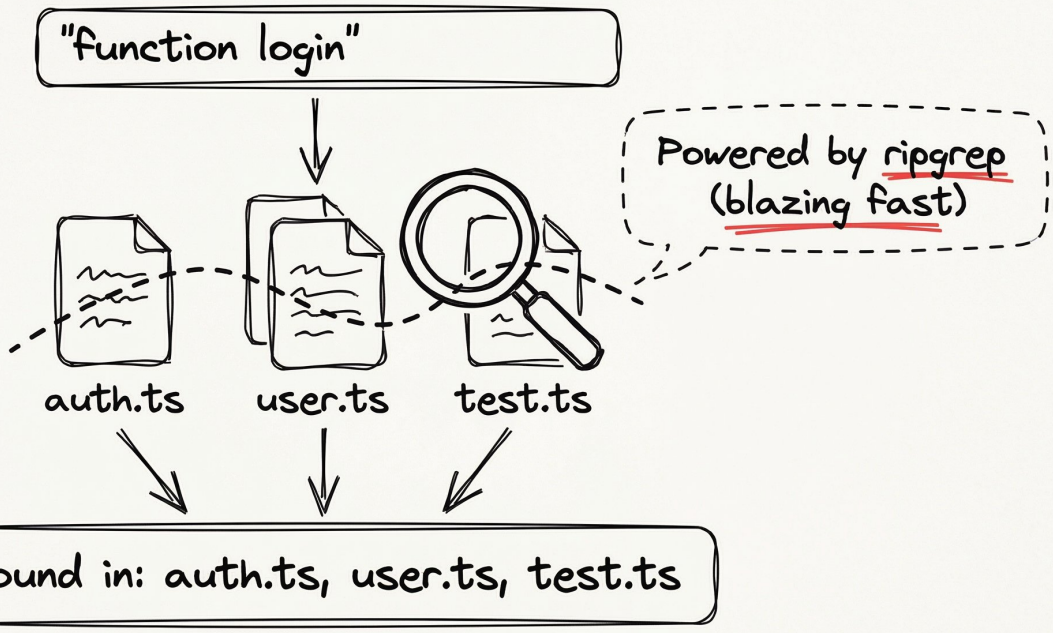
More specific rules override general ones

CLAUDE.md Hierarchy



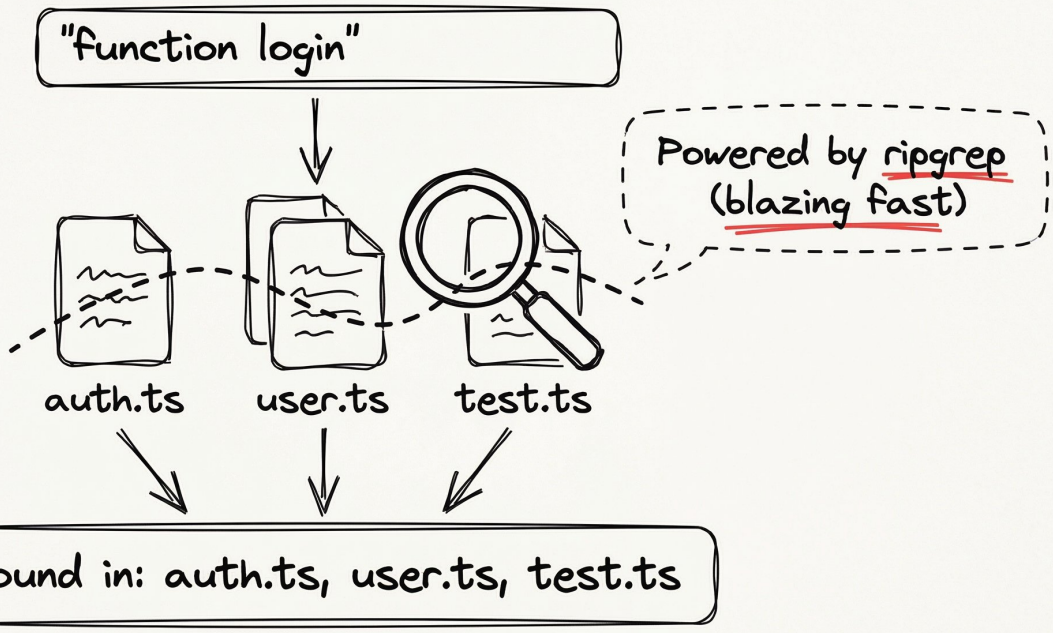
More specific rules override general ones

The Grep Tool



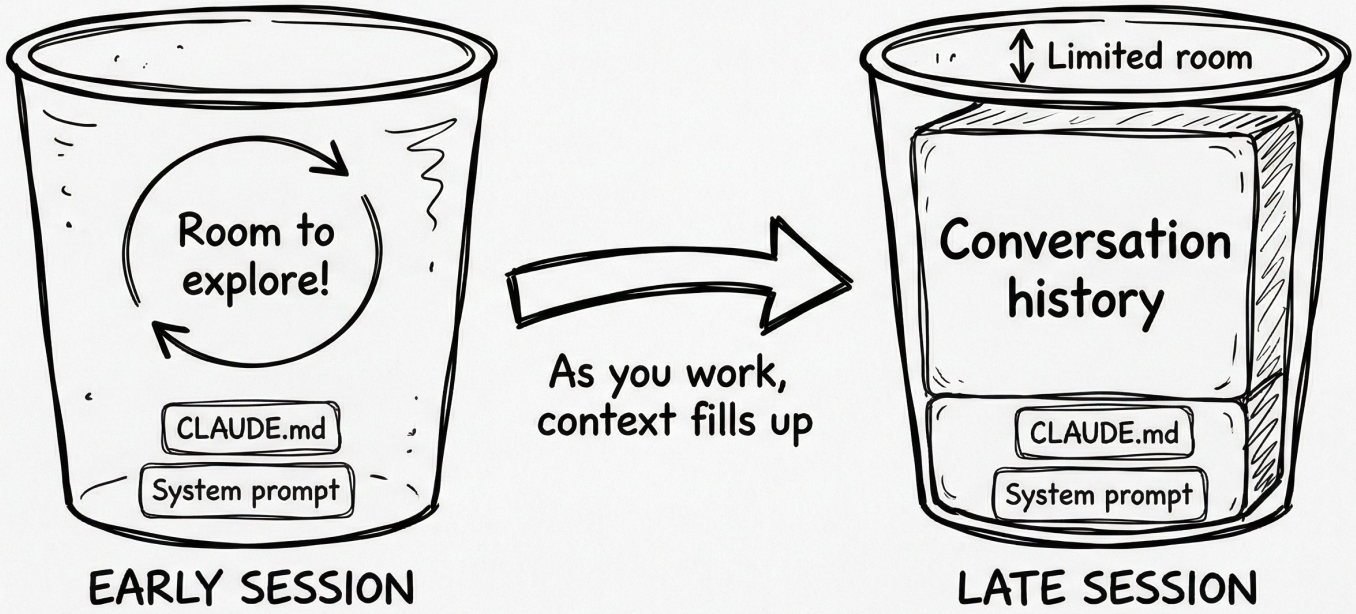
Find any text across your entire codebase

The Grep Tool



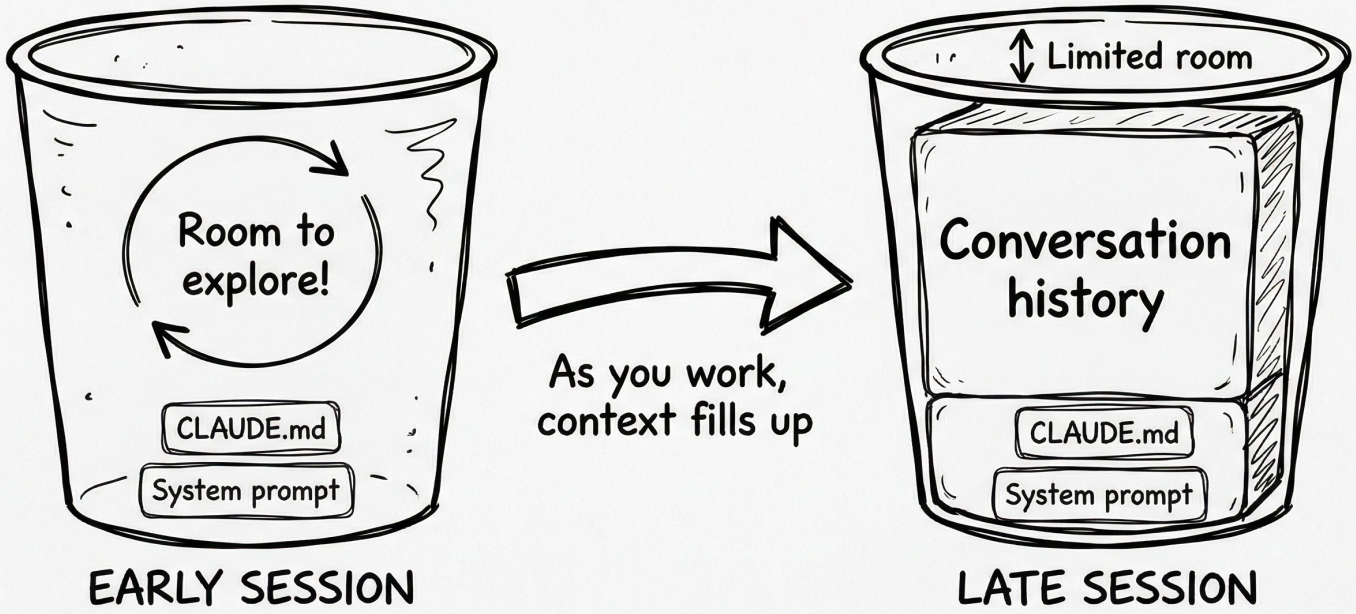
Find any text across your entire codebase

EARLY VS LATE SESSION

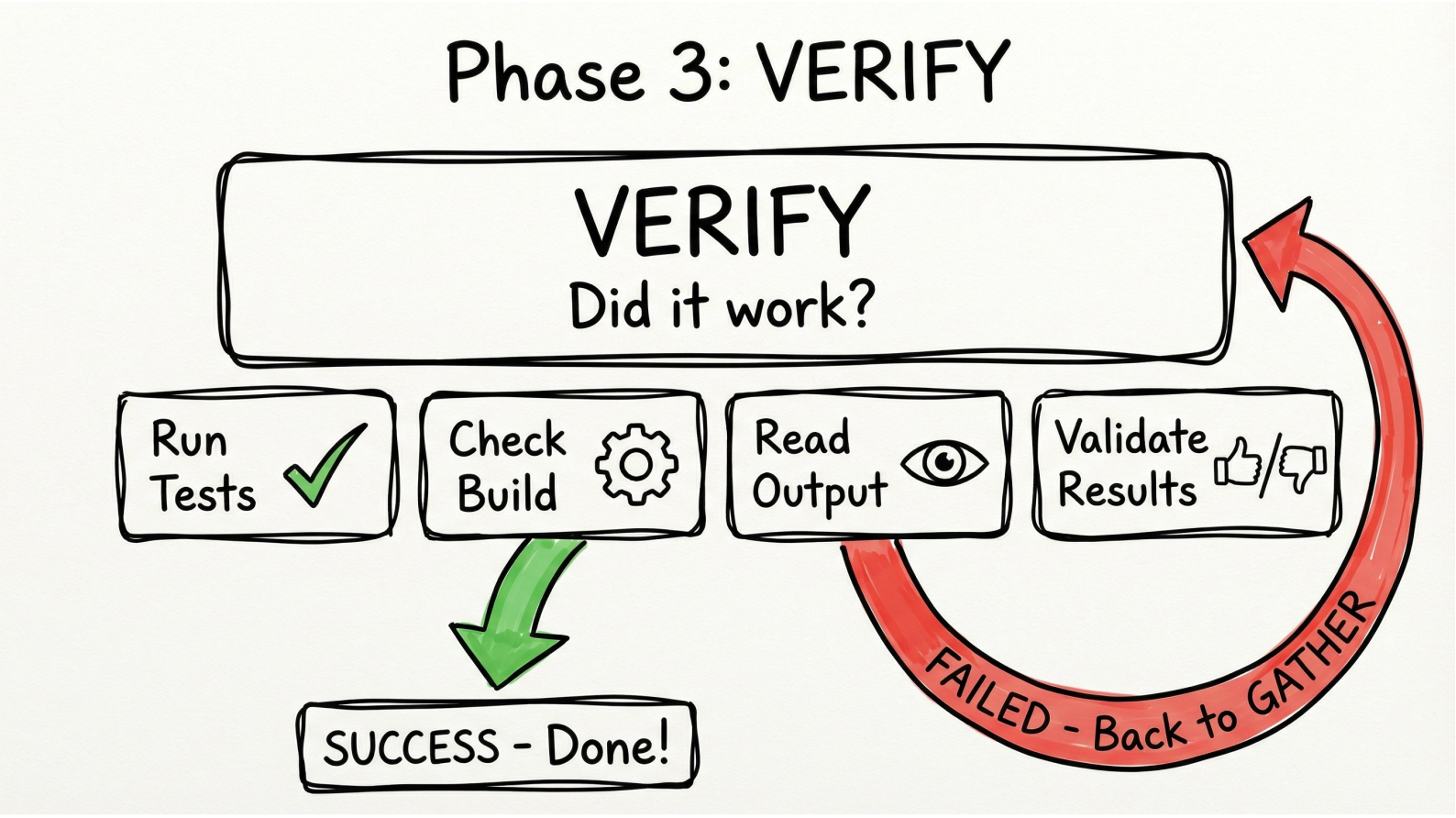


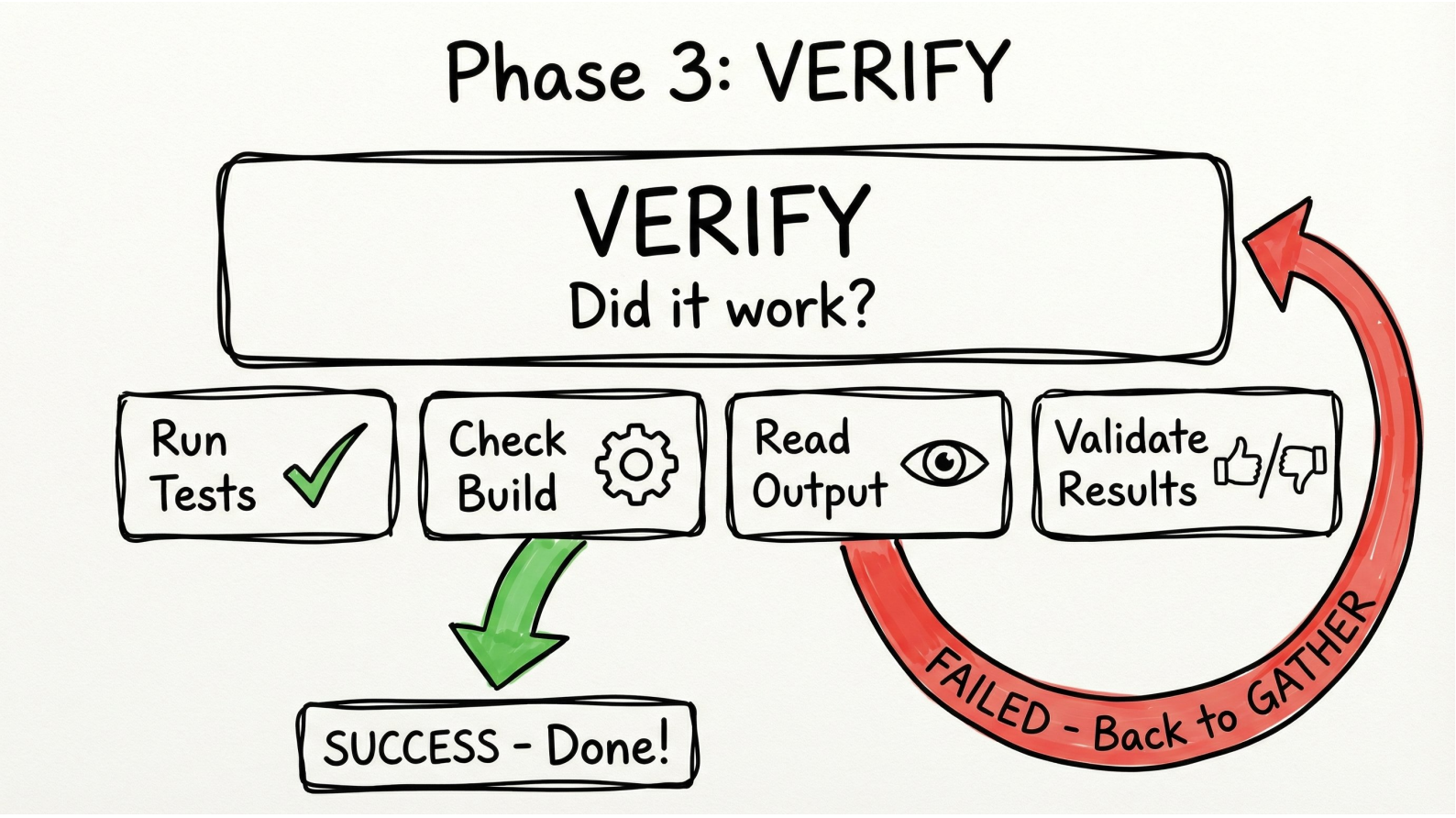
Fresh sessions have more room to work

EARLY VS LATE SESSION



Fresh sessions have more room to work


Phase 3: VERIFY
VERIFY
Did it work?
Run Tests
Check Build
Read Output
Validate Results
SUCCESS - Done!
FAILED - Back to GATHER


Phase 3: VERIFY
VERIFY
Did it work?
Run Tests
Check Build
Read Output
Validate Results
SUCCESS - Done!
FAILED - Back to GATHER

What is a Session?



What exists during a session:



A session = one continuous conversation

What is a Session?



What exists during a session:



A session = one continuous conversation