

CHAPTER SIX

App, Terminal, and Cowork

The real differences between the five places Claude Code lives — and what Cowork actually is, now that it's a third tab next to Chat and Code.

CHAPTER SIX

Where you actually are

YOUR SPLIT, FROM YOUR OWN TRANSCRIPTS

1,262 prompts in the desktop app. 56 in the terminal. Ninety-six to four.

That is a defensible choice — the desktop app is genuinely the better daily driver for what you build. But a small number of things are terminal-only, and one of them is currently blocking a project you've already built.

CHAPTER SIX

The five surfaces

Claude Code runs on the terminal CLI, the desktop app, the web (claude.ai/code), the VS Code extension, and the JetBrains plugin. They share one engine — CLAUDE.md, settings, skills and MCP servers carry across all of them.

THE THING THAT SURPRISES EVERYONE: SESSIONS DO NOT SYNC

The desktop app, the web app, and the IDE extensions **each maintain their own separate session history**. You cannot start something in the terminal and pick it up in the app. This is architectural, not a bug — and it's why your work is effectively siloed in whichever surface you started it.

■ What only the terminal can do

CAPABILITY	WHY IT MATTERS TO YOU
<code>claude setup-token</code>	This is what's blocking your AI Brief right now. See below.
Headless mode — <code>claude -p "prompt" ,</code> with <code>--output-format json</code>	Run once, capture structured output, exit. The foundation of every automation you've wanted to build. Your cron jobs need this.
Piping — <code>cat build.log claude -p</code> <code>"find the root cause"</code>	Stream data through Claude without writing files
<code>--worktree / -w</code>	Isolated git worktrees for parallel agents — you juggle six live projects daily
<code>--bare</code>	Skip all hooks, skills, plugins and MCP for reproducible CI runs

CAPABILITY	WHY IT MATTERS TO YOU
Interactive dialogs — <code>/permissions</code> , <code>/config</code> , <code>/agents</code> , <code>/doctor</code> , <code>/hooks</code>	Some of these open panels that simply don't exist elsewhere

YOUR AI BRIEF IS BLOCKED ON A TERMINAL COMMAND

Your memory file says it plainly: "BLOCKED on user running `claude setup-token` → `.env` for headless auth." The launchd timer fires every morning at 07:12 **into a wall**.

`claude setup-token` is **terminal-only**. It cannot be run from the desktop app. This is the clearest possible example of why the 4% matters: a finished, deployed project has been silently failing every single day because the one command that unblocks it lives in a surface you don't open.

Open a terminal. Run it. The project starts working.

■ What only the desktop app can do

To be fair to your choice — these are real, and they're why you're there:

- **The preview pane.** A live dev server beside your code. You used `preview_start` 186 times; this is load-bearing for you.
- **Graphical diff review** with accept/reject, rather than a terminal pager.
- **The Cowork tab** — see below.
- **Automatic worktree isolation per session**, without needing the flag.
- **Drag-and-drop files**, and a visible agent panel for background work.

THE RECOMMENDATION

Stay in the desktop app. It suits how you work — visual, preview-heavy, screenshot-driven. Don't migrate.

But keep a terminal open for four things: `claude setup-token` (once, today), headless `claude -p` for anything scheduled, `--worktree` when you're juggling projects, and the `/doctor` and `/hooks` dialogs when you set up the automation from Chapter Five.

This is not "the terminal is better." It's "four specific capabilities live there, and one of them is costing you a working product."

CHAPTER SIX

What Cowork actually is

You asked about "Claude co-work." It's **Cowork**, and it is a genuinely separate product — **a third tab in the desktop app, alongside Chat and Code.**

Chat is conversational. **Code** is Claude Code. **Cowork** is autonomous task execution for work that isn't coding — it runs unattended, on a schedule or in the background, and you review the results afterwards.

THE STATISTIC THAT DEFINES IT

Anthropic's own figure: **more than 90% of Cowork usage is not software development.** It's business operations and content creation — reports, documents, spreadsheets, research, content pipelines.

This is why it exists as a separate tab rather than a Claude Code feature. It is not aimed at you-the-developer. It is aimed at you-the-founder.

	CLAUDE CHAT	CLAUDE CODE	CLAUDE COWORK
For	Conversation, drafting	Coding, files, terminal	Business automation, async work
Execution	Interactive	Interactive, or headless	Autonomous — you set it, it runs
Runs	Anthropic's servers	Your machine (or cloud)	Anthropic's cloud sandbox
Survives you closing it?	No	No	Yes
Scheduled	No	Via Routines	Yes, natively
Tools	Search, artifacts	Full dev stack, MCP, local files	Docs, Sheets, Slack, email, cloud storage
Typical user	Everyone	Engineers	Ops, marketing, PMs

■ Where Cowork would genuinely earn its place in your week

Here's the honest read. If 90% of your work is building, Cowork mostly sits unused — Claude Code is your tool. **But you are not only a developer**, and a specific slice of what you do is exactly what Cowork is for:

- **The Instagram collection transcriptions.** You've already had Claude digest saved collections into PDF summaries — and the Quotes collection is *still outstanding* in your memory. That is a Cowork job: it runs unattended, you read the output later.
- **The AI Brief.** You built a daily news page and wired a launchd timer to refresh it. **Cowork does this natively, in the cloud, without needing `setup-token` or your laptop to be awake.** This is arguably the correct home for that entire project.

- **Client-facing documents** — the Aqsati deck, proposals, the Toyota spec sheets. Long, unattended, document-shaped work.
- **Recurring business reporting** — anything you'd otherwise do by hand on a Sunday.

WHERE COWORK IS THE WRONG TOOL

It cannot reach your local machine, your private MCP servers, or your databases. It's a cloud sandbox. Real-time debugging, production deploys, and anything needing your local filesystem stay in Claude Code — no exceptions.

VERDICT ON COWORK

It's **already included in your plan.** There is no extra cost and no setup — it's a tab you have never clicked.

Try exactly one thing on it: move the AI Brief refresh there. It's the project most obviously shaped like a Cowork job, it's currently broken in a way Cowork would sidestep entirely, and it would tell you in one afternoon whether the tool deserves a permanent place in your week.

CHAPTER SIX

A note on what I could not verify

Claude Code ships fast, and surface-by-surface feature parity is the single hardest thing to pin down accurately — the docs lag the changelog, and third-party write-ups are frequently wrong. Treat the fine-grained parity claims in this chapter as **directionally right but worth checking against the docs before you build a workflow on them.**

The four things I am confident about, because they are load-bearing here: **sessions do not sync across surfaces; `claude setup-token` and headless `-p` are terminal-only; the preview pane is a desktop-app feature; and Cowork is a separate product included in your existing plan.**

Sources. Claude Code docs (code.claude.com) — desktop, web, VS Code, JetBrains, sessions, permission modes, headless, worktrees, costs · Claude Cowork product page and July 2026 web/mobile launch announcement · TechCrunch. Surface parity details change frequently; verify against the docs before depending on them.