

Second Mind. Obsidian × Claude Code.

Your vault becomes the memory. Claude Code becomes the agent. Everything that goes through your head — captured, cross-linked, searchable, ready to be acted on.

```
$ cd ~/vault && claude  
> /init # Claude writes CLAUDE.md at your vault root
```

The rule: Agents read. Humans write.
Sources: obsidian.md/help/cli · community.obsidian.md · starmorph.com Q2 2026

Every ChatGPT session forgets you. Your vault doesn't.

01 · CAPTURE IS HUMAN

You write the notes. In your voice. Half-formed. Contradictory. Personal. This is the *signal*. Claude reads it — it never fabricates it.

Rule: "Agents read. Humans write." (Isenberg & InternetVin, 59-min Q2 2026 workflow video)

02 · CLAUDE IS THE AGENT

Claude Code reads your vault as context. Follows YOUR conventions (from **CLAUDE.md**). Creates daily notes, links, syntheses.

Everything in ~/.claude/ — plans, memory, skills. Never pollutes your knowledge with generated fluff.

03 · GIT IS THE MEMORY

Your vault is a git repo. Every change tracked. Every mistake reversible. Sync across machines without Obsidian Sync subscription.

Bonus: Claude Code respects git — asks before commits, warns on large diffs.

Four layers. Each replaceable. That's the whole point of open formats.

LAYER 1 · STORAGE

Markdown files.

plain .md + wikilinks

What lives here: your notes.

Why it matters: no lock-in. If Obsidian dies tomorrow, your knowledge is still readable in TextEdit / VSCode / cat.

Format: Markdown + [[wikilinks]] + YAML frontmatter + #tags. All plain text.

Rule: nothing proprietary. If a tool needs a database → it's not a note tool.

LAYER 2 · SYNC

Git repo.

GitHub / GitLab · private repo

What lives here: the whole vault, versioned.

Why NOT Obsidian Sync (\$10/mo): free, versioned, works from any machine, and Claude Code can read commit history.

Setup:

- git init in vault
- Obsidian Git plugin (auto-commit every N min)
- .gitignore: .obsidian/workspace*, .trash/

Bonus: Claude Code sees when you last changed a note — powerful context signal.

LAYER 3 · AGENT

Claude Code.

CLI · runs in vault dir

What lives here: the reader/writer.

Why Claude Code (not Desktop): real filesystem access. Reads notes. Creates daily notes. Runs Obsidian CLI. Custom slash commands.

Config lives in:

- CLAUDE.md — vault root
- .claude/ — skills, commands

Sandbox: workspace-write. Never danger-full-access on your vault.

LAYER 4 · UI

Obsidian.

the human-facing view

What lives here: graph view, backlinks, search, editor.

Why still Obsidian: Markdown alone isn't enough. Graph view reveals connections. Backlinks reveal orphans. Search reveals patterns.

Optional plugins for the loop:

- Dataview (SQL over notes)
- Templater (dynamic templates)
- Claude Sidebar / Claudian / Agent Client (embedded Claude Code)

Also optional: the Obsidian CLI (v1.12+) — 54× faster than grep.

Structure your vault so Claude Code can navigate blind. This is the file tree.

```
~/VAULT · THE FILE TREE

~/vault/
├── CLAUDE.md           # Claude's operating manual · ~200 lines
├── VAULT-INDEX.md     # live dashboard · what's active now
├── .claude/
│   ├── commands/      # /day, /trace, /close, /ghost
│   ├── skills/         # obsidian-daily, research-synth
│   └── settings.json   # sandbox: workspace-write
├── daily/              # YYYY-MM-DD.md · Templater-driven
├── projects/
│   ├── active-app/
│   │   └── index.md    # per-folder README for Claude
│   └── side-quest/
├── areas/              # ongoing responsibilities
├── resources/          # reference · articles · book notes
├── zettels/            # permanent notes · atomic ideas
├── templates/         # Templater .md files
├── .gitignore          # excludes .obsidian/workspace*
└── .git/               # init'd on day 1
```

4 RULES · WHY THIS SHAPE

- 01 · CLAUDE.md at root — always.** Claude reads it first every session. Include: what this vault is, how it's organized, naming conventions, "never touch X", tone rules. Aim ~200 lines (Kenneth Reitz's actual number).
- 02 · IPARAG-style folders, not free-form.** Inbox / Projects / Areas / Resources / Archives / Galaxy (Zettels). Claude can't guess where a note goes if *you* can't say. Structure prompts good filing.
- 03 · Every folder has an index.md.** Human writes the first version, Claude auto-updates when files change. Solves the "where do I look?" problem when you re-enter a folder in 3 months.
- 04 · .gitignore the noise.** Exclude .obsidian/workspace*, .trash/, .obsidian/plugins/*/data.json. Everything else — commit. Your knowledge is now version-controlled forever.

Ranked from **simplest to most powerful**. Start with #1 — most solo users never need more.

BRIDGE 1 · ★ START HERE

Vault-as-workdir.

```
$ cd ~/vault
$ claude
> /init # writes CLAUDE.md
> what's in daily/2026-07-13.md?
```

How it works: Claude Code reads/writes files directly. No plugin. No MCP. Just filesystem access.

Best for: your personal vault, 90% of solo workflows. Simplest to reason about.

Trade-off: Claude Code lives in the terminal. Not in Obsidian's UI.

BRIDGE 2 · MCP SERVER

Vault from anywhere.

```
# in ~/code/my-app/ (NOT vault)
$ claude mcp add obsidian
# configure vault path
> query my Obsidian notes on X
```

How it works: An MCP server exposes vault as a queryable tool. Claude in any repo can query it.

Options (pick one): obsidian-mcp-tools + Local REST API (community); Claudesidian MCP (Ollama embeddings); mcp-obsidian.

Trade-off: Obsidian must be running. Extra moving part.

BRIDGE 3 · IN-OBSIDIAN

Claude in the sidebar.

```
# Obsidian Community Plugins →
# pick ONE:
• Claude Code IDE (official)
• Claudian
• Agent Client
• Claude Sidebar
```

How it works: Plugin embeds Claude Code inside Obsidian. Same terminal, same commands, one window.

Best for: writers who never leave Obsidian. Note → Claude → note without alt-tab.

Trade-off: plugins evolve fast. Pin versions. Read the plugin's README before enabling YOLO mode.

Obsidian 1.12 shipped a CLI. 54× faster than grep on a 4,663-file vault.

ENABLE IT · 30 SECONDS

Settings → General → Advanced → toggle "Command-line interface". Obsidian binary appears on your \$PATH.

THE 7 COMMANDS THAT MATTER

```
obsidian daily      # open today's daily note
obsidian search "topic"  # full-text · returns paths + snippets
obsidian create path/file # new note w/ template
obsidian open path/file  # bring note to front
obsidian move old.md new.md # auto-updates ALL wikilinks
obsidian ls --orphans    # unlinked notes · 0.26s vs grep 15.6s
obsidian export file.md  # → PDF / HTML / plain md
```

WHY CLAUDE CODE LOVES IT

01 · Token savings. Claude used to `grep -r "topic" ~/vault` and pipe 200KB of raw hits. With CLI: gets a ranked, structured list. Kevin Lee reports 60%+ token reduction across all skills.

02 · Semantic operations. `obsidian move` updates wikilinks. Plain `mv` leaves broken links. Claude uses the CLI, your graph stays whole.

03 · Daily note idempotency. `obsidian daily` reads `.obsidian/daily-notes.json`. Claude gets today's file at the right path with the right template — no guessing.

04 · Works whether Obsidian is running. The `notesmd-cli` community variant runs even without the app open — useful for cron jobs and headless machines.

Rule: add "Prefer the `obsidian` CLI over grep/find" to your `CLAUDE.md`. Every session.

One markdown file = one slash command. Ship 6, iterate from there.

COMMAND	WHAT IT DOES	PROMPT SKELETON (SAVED IN .CLAUDE/COMMANDS/)
/day morning · brain-dump	Opens today's daily note. Reads yesterday's for unfinished tasks. Prompts you for a brain-dump.	Run obsidian daily . Read yesterday's daily/*.md. List unfinished tasks. Ask "what's on your mind?" and append to Today section with a timestamp.
/trace idea archaeology	Track how an idea evolved. Where did it first appear? What notes reference it? Timeline of your thinking.	Args: \$TOPIC. Search vault for \$TOPIC (obsidian search). List hits chronologically (by file.mtime). Summarize how the idea shifted across dates.
/connect wikilink suggestions	Reads current note. Suggests wikilinks to related zettels. Asks before inserting — no auto-pollution.	Read current note. For each key concept, search zettels/. Propose ≤5 wikilinks as a list. Wait for approval before editing the file.
/close evening · reflection	Closes the day. What did I ship? What's blocked? What goes to tomorrow? Writes a summary to the daily note.	Read today's daily. Extract checked tasks, open loops, decisions. Append 4-block reflection: Shipped / Blocked / Learned / Tomorrow.
/research synthesis · sourced	Cross-vault synthesis on \$TOPIC. Every claim carries a wikilink to the source note.	Args: \$TOPIC. Search resources/ + zettels/. Write a synthesis note. EVERY claim must cite [[wikilink]]. If <3 sources, refuse and say what's missing.
/ghost answer in your voice	Draft an email / doc / reply using your vault's voice — vocabulary, sentence rhythm, values.	Args: \$CONTEXT. Sample 10 recent daily/*.md for tone. Draft response. Do NOT fabricate opinions – flag anything the vault doesn't back with "[uncertain]".

Six commands are the InternetVin/Isenberg core set + community additions. Setup: each command = one .md file in .claude/commands/. First line is the description; body is the prompt. Sources: youtube Isenberg 59-min video · reddit r/ObsidianMD /day workflow · github ashish141199/obsidian-claude-code template.

Four moments where the stack **saves you an hour a day.**

WORKFLOW 1 · 08:00

Daily note · brain-dump → structured

\$ `claude` in vault. `/day`.

Voice-dictate 3 min of "what's on my mind". Claude opens today's daily note (via `obsidian daily`), parses your dump, splits into: Focus / Meetings / Ideas / Blockers. Preserves your exact words.

Vs. old way: 15 min of typing + structuring. Now: 3 min of talking.
Time saved: ~12 min/day.

WORKFLOW 2 · MIDDAY

Meeting notes → action items

During the call: raw notes into a new file under `meetings/YYYY-MM-DD-topic.md`. After: Claude, `extract action items from this file, add to today's daily under Tasks, create followups in projects/ with wikilinks back`.

The trick: Claude links the actions back to the meeting file — you always have context on WHY.
Time saved: ~10 min per meeting.

WORKFLOW 3 · AFTERNOON

Research synthesis · sourced only

`/research vector databases`. Claude walks `resources/` + `zettels/`, writes a synthesis note. Every claim = a wikilink to the source note. If <3 sources — Claude refuses and lists what's missing.

The trick: zero fabrication. If it's not in your vault, it's not in the synthesis.
Time saved: hours of re-reading.

WORKFLOW 4 · 22:00

Close the day · index maintenance

`/close`. Claude reads today's daily, updates `VAULT-INDEX.md` with what changed, refreshes any folder `index.md` that got new files, finds orphan zettels (`obsidian ls --orphans`), suggests wikilinks.

The trick: vault stays organized without you doing gardening. *You* just wrote today; Claude filed.
Time saved: ~15 min/week on maintenance alone.

If any of these matches your workflow — **fix it this week.**

#01

Letting Claude write into your zettels

Symptom: your permanent notes fill with Claude-flavored prose. Six months in, you can't tell what YOU thought.

Fix: "Agents read. Humans write." Claude writes into `daily/`, `drafts/`, `indexes`. Zettels are yours.

Cost: your voice erodes. The whole point of the vault dies.

#02

danger-full-access on the vault

Symptom: "It kept asking permission so I disabled the sandbox." One bad prompt → half your notes rewritten, no diff.

Fix: keep `sandbox_mode = "workspace-write"`. Commit before letting Claude touch anything. Git IS your safety net.

Cost: years of notes lost. Not hypothetical — it's happened.

#03

Ignoring the Obsidian CLI

Symptom: Claude runs `grep -r` across the vault. Sessions cost 3× the tokens. Wikilinks break when Claude uses `mv`.

Fix: enable the CLI (Settings → General → Advanced). Add "Prefer obsidian CLI" to your `CLAUDE.md`.

Cost: daily quota gone by lunch. Broken graph.

#04

CLAUDE.md as a wishlist

Symptom: `CLAUDE.md` has 40 vague rules ("write clean notes", "be smart"). Claude follows none of them.

Fix: only rules Claude actually breaks. When it makes a mistake twice → add the specific rule. ~200 lines max.

Cost: the file becomes decoration. Claude behaves like there's no config at all.

#05

No git · trusting one machine

Symptom: "I'll add git later." Later = when Claude accidentally deletes 40 notes and there's no history.

Fix: `git init` on day 1. Obsidian Git plugin, auto-commit every 30 min. Private GitHub repo.

Cost: unrecoverable data loss. No CTRL-Z on the filesystem.

#06

Chasing every new plugin

Symptom: 4 different Claude plugins installed. Two MCP servers. Semantic search + REST API + sidebar. Nothing quite works.

Fix: start with Bridge 1 (vault-as-workdir). Add tools when you HIT a real limit. Not before.

Cost: analysis paralysis. You configure instead of think.

Ten items. Every one is a command or a file — not a concept.

DAY 1 · FOUNDATIONS · 4 ☐ 01 · Vault at ~/vault. git init. Folders: daily/ projects/ areas/ resources/ zettels/ templates/. ☐ 02 · .gitignore: .obsidian/workspace*, .trash/. First commit. ☐ 03 · Obsidian Settings → General → Advanced → toggle Command-line interface. Test: obsidian daily. ☐ 04 · In vault: claude, then /init. Edit the generated CLAUDE.md — cut to ~200 lines of real rules.	DAY 2-4 · COMMANDS · 3 ☐ 05 · Write .claude/commands/day.md. Use it every morning for a week. ☐ 06 · Write .claude/commands/close.md. Use it every evening. Notice which VAULT-INDEX changes stick. ☐ 07 · Add one more command from your actual friction — /trace, /connect, or /research. Don't ship all six on day one — earn them.	DAY 5-7 · HABIT · 3 ☐ 08 · Install Obsidian Git plugin. Auto-commit every 30 min. Verify commits appear on GitHub. ☐ 09 · Retrospective: when Claude makes the same mistake twice, add the rule to CLAUDE.md. Not before. ☐ 10 · One end-to-end run: brain-dump at 08:00 → do work → /close at 22:00. Read tomorrow's daily. If yesterday's context is <i>right there</i>, you've built it.
---	--	--