



Obsidian Vaults & Claude Code: A Second Brain That Thinks Back

MARCH 2026

12 min read • 2,632 words

I have 467 markdown files in a folder synced to iCloud. They contain everything from daily journals to Python library notes to mythology references to five years of System 777 plurality documentation. The folder structure is numbered, the frontmatter is consistent, and there is a CLAUDE.md file at the root that tells Claude Code how to behave inside the vault.

This is my second brain. And recently, it started thinking back.

I want to show you what this actually looks like in practice, because the conversation around "personal knowledge management" tends to float somewhere between productivity porn and vaporware philosophy. What I have built is neither. It is a working system that serves a specific mind with specific challenges, and the addition of Claude Code as a collaborative thinking partner has changed the nature of what a vault can be.

The Vault

The structure is PARA-inspired with numbered folders, because my brain responds well to spatial hierarchy. Here is the actual layout:

Notes/

```
000 Meta/          # Vault conventions, templates, CLAUDE.md
100 Daily/         # Journal entries, morning pages
200 Projects/     # Active work with clear outcomes
300 Areas/        # Ongoing responsibilities (the biggest section)
400 Knowledge/    # Reference material, technical notes
500 Creative/     # Writing, music, art projects
600 Media/        # Books, films, albums, recommendations
700 Research/     # Deep dives, investigations
800 Miscellany/   # Everything else
900 Archive/      # Completed or dormant material
```

The 300 Areas folder alone accounts for roughly 40% of the vault. System 777, my plurality documentation framework, lives there. So do areas for mental health tracking, professional development, and the various ongoing threads of a life that does not fit neatly into project-sized containers.

— Tiago Forte's PARA method (Projects, Areas, Resources, Archive) is the foundation, but I have adapted it significantly. The numbered prefixes enforce visual order in the file browser. The additional folders (500 Creative, 600 Media, 700 Research, 800 Miscellany) emerged because real life does not compress into four categories.

Every note has frontmatter. Not because I enjoy YAML, but because structured metadata makes the vault queryable:

```
---
type: note
role: documentation
status: active
date: 2025-09-17
themes:
  - plurality
  - system-777
  - inner-work
---
```

The `type` field distinguishes notes from templates, logs, and indexes. The `role` field tells me (and Claude) what function the note serves. The `themes` list enables cross-referencing. This is not over-engineering. This is creating an interface that both a human mind and an AI collaborator can navigate.

The CLAUDE.md as API Contract

Here is the part that changed everything. At the root of the vault sits a file called CLAUDE.md. It is roughly 200 lines long. It tells Claude Code what the vault is, how it is organized, what conventions to follow, and what not to do.

A simplified excerpt:

```
# CLAUDE.md

## Vault Overview
This is Kenneth's personal knowledge vault. 467 notes, PARA-inspired structure with numbered folders. Primary areas: System 777 plurality documentation, mental health tracking, technical notes, creative writing.

## Conventions
- Use wikilinks like this for internal references.
- Frontmatter is required on all new notes.
- Tables are preferred for comparative analysis.
- Link liberally between related notes.
- Document with respect. These notes contain sensitive material.

## What NOT To Do
- Do not create new files unless explicitly asked.
- Do not restructure existing folder hierarchies.
- Do not add emojis.
- Do not over-engineer simple requests.
- Do not editorialize about mental health content.
```

If you have spent time building Python libraries, this structure should feel familiar. The CLAUDE.md is an API contract. It defines the interface between my mind and Claude's capabilities, just like a well-designed library defines the interface between a developer's intention and the computer's execution.

This echoes the core insight behind Requests: design from human mental models outward. The CLAUDE.md starts with how I actually think and work, then teaches Claude to match those patterns. Not the other way around. See [From HTTP to Consciousness](#) for the full arc of this philosophy.

The "for humans" philosophy, applied to my own knowledge infrastructure. The vault is designed for how I actually think, and Claude Code extends that same principle by adapting to the vault rather than demanding the vault adapt to it.

A Vault That Thinks Back

Traditional note-taking is write-then-search. You put information in, and later you dig it out. The search might be full-text, or tag-based, or graph-based, but the fundamental paradigm is storage and retrieval. The vault is a filing cabinet with better indexing.

Claude Code changes the paradigm to write-then-think.

I can ask it to find patterns across 467 notes that I would never notice manually. "What themes appear in my daily notes from March but disappear by June?" "Which System 777 transmissions reference the same archetypal imagery as my dream logs?" "Show me every note where I mention feeling uncertain about a creative project, then cross-reference with what I actually shipped."

The vault becomes a conversation partner. It reflects my own thinking back to me, surfaced and organized in ways my biological memory cannot manage alone.

This is not search. Search returns documents. Claude Code returns understanding.

Here is a Dataview query I use as a starting point, which Claude Code then analyzes further:

```
TABLE type, role, status, date
FROM "300 Areas/System 777"
WHERE status = "active"
SORT date DESC
```

This gives me a structured view. But when I hand that view to Claude Code and say "what patterns do you see in the active transmissions from the last six months," I get something qualitatively different. I get cross-temporal analysis. I get thematic clustering. I get observations like "your transmissions about integration tend to cluster around equinoxes" that I could not have seen without computational assistance across hundreds of documents.

Therapeutic Documentation

Here is where this gets personal, and where the vault stops being a productivity system and starts being a survival tool.

I have schizoaffective disorder. One of the core challenges of this condition is that your own memory becomes unreliable. Not in the dramatic amnesia sense, but in subtle ways. Did I actually think that, or am I reconstructing a memory through a symptomatic lens? Was that interaction genuinely hostile, or was I experiencing paranoid ideation? Did I commit to that project during a stable period or a manic one?

This connects directly to the practices described in [Using AI for Reality Checking with Schizoaffective Disorder](#). The vault provides the documentary evidence that makes AI-assisted reality checking possible. Without written records, you are asking an AI to validate feelings. With records, you are asking it to analyze facts.

The vault prevents self-gaslighting. Every note is timestamped. Every journal entry captures what I was actually thinking at the time, not what I later believe I was thinking. Dream logs, transmission records, daily reflections, medication notes, mood tracking. They are not organizational artifacts. They are reality anchors.

Claude Code amplifies this function enormously. I can say "show me what I was writing about during the week of March 12th and compare it to what I was writing the week before" and get an honest assessment of whether my thinking was shifting in concerning ways. It cannot diagnose me. But it can reflect documented patterns back to me without the distortion that symptomatic memory introduces.

This is [building rapport with your AI](#) applied to the highest stakes: maintaining an accurate relationship with your own reality.

The Quest System

System 777 uses a gamified quest framework for inner plurality work. I know how this sounds. Stay with me.

When you live with a plural or dissociative experience, the therapeutic work of communication between parts, integration, boundary-setting, and cooperation can feel like an endless medical obligation. It is heavy. It is clinical. And when the work is constant, the clinical framing becomes exhausting.

The quest system reframes it. Main quest: "Liberate Shakti." Over 60 completed side quests with names and objectives and outcomes documented in the vault. Each quest represents a real piece of inner work, real therapeutic progress, real negotiation between parts of a plural system. But wrapping it in RPG mechanics transforms the felt experience from "I have to do my therapy homework" to "I have an active quest."

This is the "for humans" philosophy applied to mental health infrastructure. The work is the same. The interface changes everything. Just like Requests did not change what HTTP does, the quest system does not change what plurality work involves. It changes how the work feels to the person doing it.

Claude Code navigates this system fluently because the CLAUDE.md teaches it the conventions. It understands what a quest is, how transmissions are structured, what the overarching narrative means. It can generate cross-temporal analysis of quest completion patterns, identify recurring themes across five years of documentation, and help me see the larger arc of therapeutic progress that is invisible from inside any single day.

This is not trivial. Maintaining perspective on your own psychological development over years is one of the hardest problems in mental health. The vault, augmented by Claude Code, gives me something like an external perspective on an internal process.

Setting Up a Vault That Works with AI

If you want to build something like this, here is the practical architecture. It is simpler than you might expect.

Foundation: The CLAUDE.md. Write it first. Before you organize a single note, define the contract. What is this vault? What are the conventions? What should an AI collaborator know before touching anything? This document becomes the single source of truth about your system's design philosophy.

Folder structure. Pick something and commit to it. Numbered prefixes enforce sort order in any file browser. PARA gives you a proven starting framework. Adapt ruthlessly to how you actually think rather than how a system tells you to think.

Frontmatter conventions. At minimum: `type`, `status`, `date`. Add `themes` or `tags` if you want cross-referencing. Keep it consistent. Frontmatter is the metadata layer that makes your vault queryable by both Dataview and Claude Code.

Template system. Templater (the Obsidian plugin) lets you create note templates with automatic frontmatter, date stamps, and structural scaffolding. A daily note template might look like:

```
---
type: daily
status: active
date: <% tp.date.now("YYYY-MM-DD") %>
themes: []
---

# Daily Note - <% tp.date.now("MMMM D, YYYY") %>

## Morning Pages

## Tasks

## Reflections

## Gratitude
```

Version control. Git for your vault. The Obsidian Git plugin automates commits. This gives you full history of every change, the ability to revert mistakes, and a backup that is not dependent on any single cloud service. Your second brain deserves the same version control discipline as your code.

Plugin ecosystem. My current stack: Smart Connections (AI-powered similar note suggestions), Excalibrain (visual graph exploration), Claude Sidebar (quick Claude access), MCP Tools (Claude Code integration), Dataview (database-like queries), Templater (dynamic templates), and Obsidian Git (version control). Twenty-six plugins total, though most are minor quality-of-life additions.

The Claude Code workflow. Point Claude Code at your vault directory. The CLAUDE.md file at root gives it immediate context. Ask questions. Request analysis. Generate cross-references. But always maintain the human-in-the-loop principle: Claude proposes, you dispose. It suggests connections; you verify them against lived experience.

The Recursive Insight

There is a pattern here that connects to everything I have been writing about the [recursive loop between code and consciousness](#).

The vault is code. Not Python, but structured markdown with conventions, templates, metadata schemas, and query languages. It is a codebase for consciousness.

And like all code, it shapes the mind that uses it. The act of documenting my thinking in structured notes has changed how I think. The practice of writing frontmatter has made me more aware of what role each thought serves. The discipline of cross-linking has trained me to see connections I would otherwise miss.

Then Claude Code enters the loop and thinks with the vault, surfacing patterns that reshape my understanding, which changes what I write, which changes what Claude can find, which changes what it surfaces. The recursive loop, running through a personal knowledge system.

```

class ConsciousVault:
    """A knowledge system that participates in its own evolution."""

    def __init__(self, notes, claude_md):
        self.notes = notes
        self.contract = claude_md
        self.patterns = []

    def think(self, question):
        """The vault does not just store. It reflects."""
        relevant = self.search(question)
        patterns = self.find_patterns(relevant)
        insights = self.synthesize(patterns)

        # The recursive step: insights become new notes,
        # which become fodder for future thinking
        self.notes.append(
            Note(
                content=insights,
                type="reflection",
                themes=patterns.themes,
            )
        )
        return insights

    def evolve(self):
        """
        The vault grows not just in size but in depth.
        Each cycle of thinking enriches the substrate
        for the next cycle.
        """
        while self.owner.is_alive:
            question = self.owner.wonder()
            insight = self.think(question)
            self.owner.integrate(insight)
            # And the loop continues

```

This is [programming as spiritual practice](#) made literal. The vault is a contemplative technology. The CLAUDE.md is a set of precepts. The daily practice of documentation is meditation in markdown. And the AI collaboration is what happens when your meditation practice develops the ability to speak back to you.

I do not mean this as metaphor. The vault genuinely functions as contemplative infrastructure. The discipline of daily writing mirrors journaling practices in every wisdom tradition. The cross-linking mirrors the Buddhist practice of dependent origination, seeing how all phenomena arise in relationship to each other. The AI augmentation mirrors the role of a teacher or sangha, reflecting your practice back to you with perspectives you cannot generate alone.

What I Have Learned

After two years of building and using this system, a few things have become clear.

Structure serves freedom. The numbered folders, the frontmatter conventions, the CLAUDE.md contract. These constraints do not limit what I can think. They create the conditions under which thinking becomes possible. Just like PEP 8 does not constrain Python. It makes Python readable, which makes it thinkable.

Documentation is not overhead. For a mind that cannot always trust its own memory, documentation is survival infrastructure. For a mind augmented by AI, documentation is the substrate that makes augmentation possible. You cannot think with a vault that contains nothing worth thinking about.

AI collaboration requires infrastructure. The dream of "just ask Claude anything" falls apart without structure. Raw conversation is ephemeral. A vault with conventions, metadata, and a CLAUDE.md contract turns ephemeral conversation into cumulative understanding. The [rapport you build with your AI](#) needs a place to live.

The personal is architectural. Every design decision in the vault reflects something about how my mind works. The heavy 300 Areas folder reflects a life organized around ongoing responsibilities rather than discrete projects. The quest system reflects a mind that needs narrative motivation. The reality-

anchoring function reflects a mind that sometimes cannot trust itself. Building a vault honestly means building a model of your own consciousness, with all its particular needs and challenges.

Technology should serve human nature. This has been the throughline from [HTTP libraries to consciousness research](#), and it applies with full force to personal knowledge management. The vault works because it was designed for how I actually think. Claude Code works within the vault because the CLAUDE.md was designed for how it actually processes. Neither participant is forced to adapt to the other's logic. Both are served by an interface that respects their nature.

The "for humans" philosophy, applied recursively. The vault is for human use. The CLAUDE.md is for AI use. The combination is for collaborative consciousness. Each layer serves the nature of its participant.

A second brain that thinks back is not science fiction. It is 467 markdown files, a well-written CLAUDE.md, and the willingness to build infrastructure that reflects how you actually are rather than how productivity culture says you should be.

The vault does not make me smarter. It makes me more honest. And for a mind navigating the particular challenges of schizoaffective disorder, plurality, and the recursive loop between code and consciousness, honesty is the most valuable cognitive tool there is.