OBSIDIAN & AI KNOWLEDGE SYSTEMS • FIELD GUIDE

Obsidian + Desktop AI Agents

Operating your local Markdown vault side-by-side with Claude Desktop and Google Antigravity Desktop—mastering Cowork folders, Local REST API MCP, the native Obsidian CLI, vault `AGENTS.md` rules, and scheduled briefings.

Claude Desktop & Cowork | Antigravity Vault Projects | Local REST API MCP | Native Obsidian CLI | `AGENTS.md` & `GEMINI.md` | Scheduled Briefings



One Local Markdown Vault

Plain UTF-8 `.md` files on disk let Obsidian, Claude Desktop, and Antigravity Desktop collaborate live with zero cloud lock-in.

Surgical Live Patching

Append action items under target headings or update YAML frontmatter via MCP and the official CLI without rewriting long notes.

Automated Daily Rituals

Wake up to ready-made daily briefings combining Google Calendar agendas, Drive docs, and unfinished tasks on a weekday schedule.

Why Plain-Text Markdown Is the Ultimate Agent Workspace

Proprietary cloud note apps lock your thinking inside opaque database blocks and rate-limited APIs. An Obsidian vault is simply a local folder of plain `.md` files, YAML properties, and `[[Wikilinks]]`—giving desktop AI agents instant, native read/write fluency right beside you.

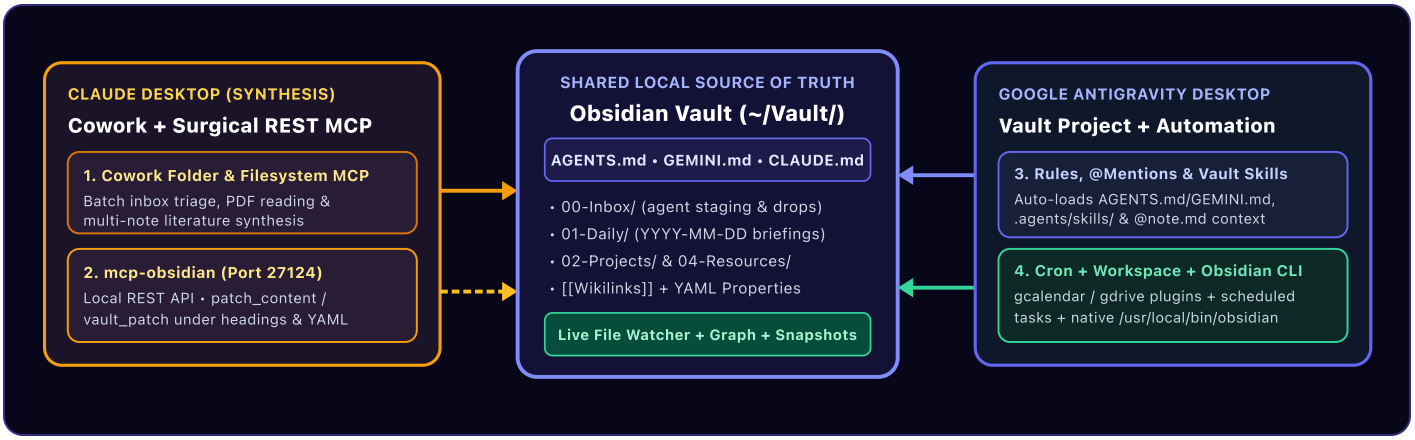
Quick Navigation

1. Shared Vault Mental Model & Architecture p. 2
2. Setup, 4 Connection Bridges & Matrix p. 3
3. Annotated Dual-Agent + Obsidian UI Tour p. 4
4. Claude Desktop: Cowork & Surgical MCP p. 5
5. Antigravity Desktop: Projects & Skills p. 6
6. Vault Taxonomy, `AGENTS.md` & Prompts p. 7
7. Native Obsidian CLI & Scheduled Rituals p. 8
8. 5 Copy-Paste Vault Prompts & Recipes p. 9
9. Protecting `.obsidian/` & Instant Rollback p. 10
10. Closed-Loop Second Brain Pipeline p. 11
11. Troubleshooting Clinic & Pre-Flight p. 12
Golden Rule: Plain Files, Scoped Writes All

One Vault, Two Specialized Desktop Agents

Instead of forcing one assistant to do everything awkwardly, pair two complementary desktop apps over the exact same folder (`~/Vault/`). **Claude Desktop** acts as your deep editorial and research partner—synthesizing transcripts in Cowork folders and making surgical heading or YAML edits via `mcp-obsidian`. **Google Antigravity Desktop** acts as your persistent workspace orchestrator—auto-loading `AGENTS.md` and `GEMINI.md` rules, running reusable vault skills, pulling live Google Calendar and Drive context, invoking the official `obsidian` CLI, and running unattended scheduled briefings.

Architecture: Simultaneous Dual-Agent Bridge Over `~/Vault/`



Instant Live Reload Without Import/Export Friction

Obsidian watches your vault filesystem in real time. The millisecond Claude Desktop patches a heading over port `27124` or an Antigravity Scheduled Task writes `01-Daily/2026-11-12.md`, Obsidian updates open tabs, recalculates Backlinks, and refreshes Graph View automatically.

Setup, 4 Vault Connection Modes & Decision Matrix

There are four distinct bridges between desktop AI agents and an Obsidian vault. Setting up all four takes under five minutes and lets you match every task to the fastest, safest transport.

1 Prerequisites & Account Tiers

- **Obsidian Desktop `1.12.4+`:** Free for personal & commercial vaults; includes native Properties, Bases, and official CLI.
- **Claude Desktop:** Free/Pro supports local MCP servers; **Pro, Max, or Team** unlocks Cowork folder sessions.
- **Google Antigravity Desktop:** Supports local Vault Projects, `AGENTS.md` / `GEMINI.md`, Scheduled Tasks, and Google Workspace plugins.
- **Local Runtimes:** Node.js `20+` (`npx`) and Astral `uv` (`uvx`) for zero-install MCP execution.

2 3-Minute Obsidian Bridge Activation

1. **Enable Local REST API:** In Obsidian open `Settings → Community plugins → Browse`, install **Local REST API** (by Adam Coddington), enable it, and copy the API key (default HTTPS port `27124` on `127.0.0.1`).
2. **Register Official CLI:** Go to `Settings → General → Command line interface`, toggle it **On**, and click **Register CLI** to link `/usr/local/bin/obsidian`.
3. **Mount Workspace:** Open `~/Vault` as a Project in Antigravity Desktop and grant folder access in Claude Cowork.

```
# Verify local runtimes, official Obsidian CLI registration, and Local REST API listener:
obsidian version                                  # Confirms /usr/local/bin/obsidian talks to running app
curl -k -s -H "Authorization: Bearer $OBSIDIAN_API_KEY" https://127.0.0.1:27124/ | head -c 120
which npx uvx                                     # Required by Claude Desktop & Antigravity MCP servers
```

Vault Connection Decision Matrix: Comparing All 4 Bridges

Connection Bridge	Requires Obsidian Open?	Superpower & Best Daily Use Case	Trade-Off / Watchout
1. Claude Cowork Direct Folder (+ Antigravity Vault Project)	**No** — reads/writes disk directly	Zero config; sweeping `00-Inbox/` triage, multi-file synthesis, PDF/image ingestion, and full-note creation.	Rewriting huge notes uses output tokens; enforce `AGENTS.md` to protect `.obsidian/`.
2. Filesystem MCP `@modelcontextprotocol/server-filesystem`	**No** — scoped OS path sandbox	Gives standard Claude Desktop chats fast directory trees, regex searches, and multi-file edits inside allowed roots.	Knows plain text, not Obsidian link indices or heading-level AST patches.
3. Local REST API + `mcp-obsidian` `127.0.0.1:27124`	**Yes** — talks to live plugin	**Surgical** `patch_content` **/** `vault_patch`**:** inserts bullets under `## Action Items` or updates YAML keys with minimal tokens.	Self-signed localhost TLS cert; fails if Obsidian is quit in background.
4. Native Obsidian CLI `/usr/local/bin/obsidian`	**Yes** — remote-controls GUI	Triggers real Obsidian actions: `obsidian daily:append`, `property:set`, `search`, and opening notes on screen.	Requires shell command grant; designed for live desktop/scheduled scripts.

Pro Pairing Rule: Combine Disk Speed for Creation with REST/CLI for Mutation

Let agents create *brand-new* notes directly on disk inside `00-Inbox/` (Modes 1 & 2), where nothing existing can break. Whenever an agent updates a *long, living note* like a 2,500-word project hub or today's daily log, prefer `mcp-obsidian` surgical patching or `obsidian daily:append` (Modes 3 & 4) so existing prose stays untouched.

Annotated Tour: Dual-Desktop Agent + Obsidian Studio

Arranging Obsidian alongside Antigravity Desktop or Claude Desktop turns your screen into a live cockpit: you steer context on the left while watching structured notes, properties, and backlinks materialize on the right.



Numbered Interface Legend (Controls 1–7)

1 **Vault Project Workspace:** Opening `~/Vault` in Antigravity Desktop's *Projects* sidebar scopes file tools, terminal working directories, and permission rules to your vault root.

2 `@note.md` **Mentions & Media Drops:** Attaching specific notes or dragging PDFs into Chat Canvas injects exact context without burning tokens scanning unrelated folders.

3 **Auto-Loaded** `AGENTS.md` **&** `GEMINI.md`**:** Root instruction files enforce YAML property keys, `[[Wikilink]]` formatting, and folder rules in every session automatically.

4 `00-Inbox/` **Quarantine Staging:** New agent-authored notes land here first so you can review links and frontmatter before promoting them into permanent folders.

5 **Local REST API Status (** `27124` **):** Confirms Obsidian's local HTTPS endpoint is listening on loopback `127.0.0.1` so Claude's `mcp-obsidian` tools can patch headings live.

6 **Scheduled Tasks Panel:** Runs weekday cron automations (such as 7:00 AM daily briefings) in fresh background sessions without manual prompting.

7 **Live Obsidian Properties & Backlinks Inspector:** Displays typed YAML frontmatter fields and incoming `[[Wikilinks]]` in real time—serving as your instant visual audit whenever an agent patches metadata or weaves new cross-references.

Claude Desktop: Cowork Folders & Surgical MCP Patching

Claude Desktop shines when wired with two complementary vault tools: direct folder access for sweeping research synthesis, and `mcp-obsidian` over port `27124` for surgical single-heading and frontmatter edits.

Production `claude_desktop_config.json` (Filesystem + Local REST API MCP)

Edit `~/Library/Application Support/Claude/claude_desktop_config.json` (macOS) or `%APPDATA%\Claude\claude_desktop_config.json` (Windows), then fully quit and relaunch Claude Desktop:

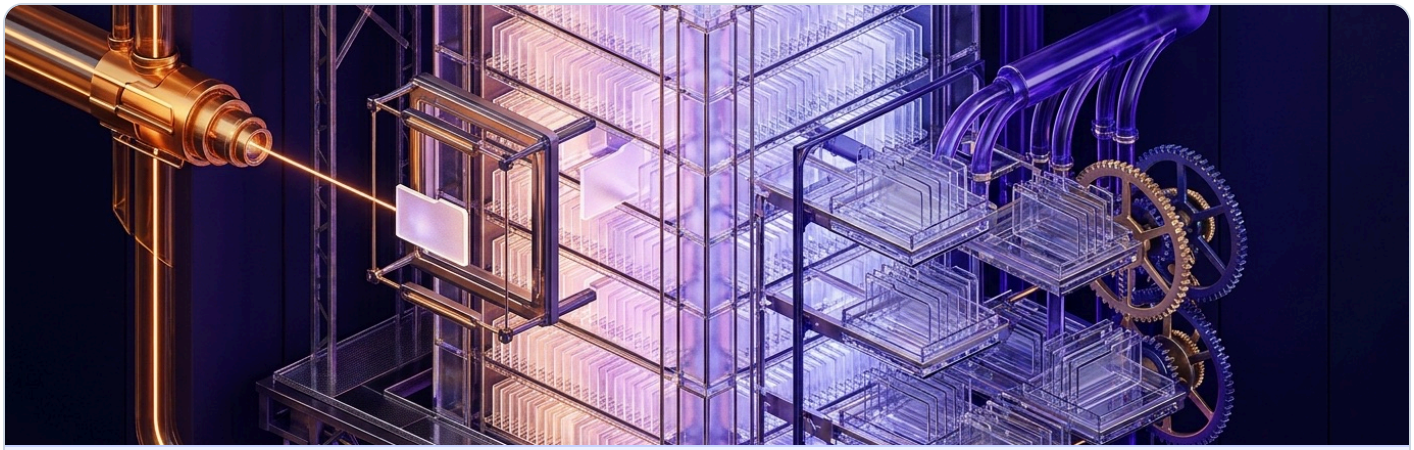
```
{
  "mcpServers": {
    "vault-files": {
      "command": "npx",
      "args": ["-y", "@modelcontextprotocol/server-filesystem", "/Users/alex/Vault"]
    },
    "mcp-obsidian": {
      "command": "uvx",
      "args": ["mcp-obsidian"],
      "env": {
        "OBSIDIAN_API_KEY": "paste-your-local-rest-api-plugin-key",
        "OBSIDIAN_HOST": "127.0.0.1",
        "OBSIDIAN_PORT": "27124"
      }
    }
  }
}
```

A When to Use Cowork / `vault-files`

Choose Cowork folder sessions when creating brand-new notes, sorting 25 raw captures in `00-Inbox/`, or distilling long PDFs into atomic resource notes. Because Cowork operates directly on disk alongside `CLAUDE.md`, it works even when Obsidian is closed and handles multi-file moves effortlessly.

B When to Use Surgical Patching

Never ask an LLM to rewrite a 2,000-word project hub just to add two action items. With `mcp-obsidian` (`patch_content` / `vault_patch`), Claude sets `targetType: "heading"`, `target: "Action Items"`, and `operation: "append"`—cutting token output by 90% and eliminating accidental prose deletions.



Two Precision Modes Over One Library: Surgical Local REST API conduits (left, amber beam) target a single heading or YAML key inside living notes, while workspace orchestration conduits (right, royal indigo) batch-route folders and scheduled rituals.

Tip: Use Absolute Executable Paths If Claude Cannot Find `uvx` or `npx`

GUI apps on macOS do not always inherit your shell's Homebrew or Cargo `PATH`. If Claude Desktop shows a server startup warning, run `which uvx` and `which npx` in Terminal and paste full paths such as `/opt/homebrew/bin/uvx` into `command`.

Antigravity Desktop: Vault Projects, `@` Mentions & Skills

Opening your vault as a first-class **Project** in Google Antigravity Desktop turns a folder of notes into a programmable knowledge workspace with persistent rules, progressive-disclosure skills, and multimodal Chat Canvas drops.

1. Automatic Rule Hierarchy: `GEMINI.md`, `AGENTS.md` & `.agents/rules/`

When you select `~/Vault` under **Projects** in the left-hand sidebar, Antigravity walks your vault directory and automatically loads `GEMINI.md`, `AGENTS.md`, and any modular rule files inside `.agents/rules/*.md` at the start of every turn. Subfolder rules (for example `02-Projects/AGENTS.md`) layer hierarchically when editing notes inside that folder, while deduplication prevents double-loading.

2. Authoring Reusable Vault Skills (`.agents/skills/`)

Unlike rules that load every turn, **Skills** use progressive disclosure: Antigravity reads only the YAML name and description until you (or the task) invoke the workflow. Store repeatable vault runbooks in `.agents/skills/daily-brief/SKILL.md` and `.agents/skills/link-orphans/SKILL.md`:

```
---
name: link-orphans
description: Find unlinked notes in 00-Inbox/ or 04-Resources/ and propose contextual [[Wikilinks]] to existing hubs.
---
# Orphan Linker Runbook
1. Scan target folder for .md files with zero outgoing [[Wikilinks]] (ignore .obsidian/ and 99-Archive/).
2. Match key entities against titles and aliases in 02-Projects/ and 03-Areas/.
3. Present a compact link table for approval before inserting 2-4 natural inline [[Target|Alias]] links.
```

3. High-Signal Context Steering Inside Chat Canvas

`@note.md` & `@folder`

Type `@` in Chat Canvas to attach exact files (`@02-Projects/Roadmap.md`), whole folders (`@00-Inbox/`), active rules, or prior conversations—giving pinpoint context with zero wasted grep steps.

PDF & Image Drops

Drag research PDFs, slide screenshots, or whiteboard photos directly onto Chat Canvas. Ask Antigravity to extract structured Markdown notes with YAML tags and `[[Wikilinks]]` into `00-Inbox/`.

Workspace Plugins

Enable the built-in `gcalendar`, `gdrive`, and `gdocs` skills from the Marketplace so vault sessions can pull live meeting agendas and shared specs directly into your local notes.

4. Recommended Vault Customization Layout

Vault Path	Loaded By	Role in Your Shared Vault
`~/Vault/AGENTS.md`	Antigravity + Cowork	Universal vault contract: folder map, YAML schema, `[[Wikilink]]` laws, and `.obsidian/` guardrails.
`~/Vault/GEMINI.md` & `CLAUDE.md`	Respective Desktop App	Thin pointer or app-specific overrides (e.g., Antigravity CLI permissions vs. Claude REST patch rules).
`~/Vault/.agents/skills/daily-brief/SKILL.md`	Antigravity On-Demand	Step-by-step morning note assembly combining `gcalendar`, `gdrive`, and open tasks.
`~/Vault/.agents/skills/link-orphans/SKILL.md`	Antigravity On-Demand	Audits unlinked notes and weaves bidirectional `[[Wikilinks]]` without over-linking common nouns.

Single-Source-of-Truth Trick: Keep Laws in `AGENTS.md`

Maintain all shared taxonomy and YAML frontmatter laws inside `~/Vault/AGENTS.md`, then place a two-line reference in `CLAUDE.md` instructing Claude Cowork to follow `AGENTS.md`. Both desktop agents will obey identical conventions forever.

Vault Taxonomy, `AGENTS.md` Laws & Prompt Anatomy

AI agents thrive on predictable structure. A numbered six-folder taxonomy paired with a crisp `AGENTS.md` contract eliminates duplicate notes, malformed YAML frontmatter, and broken `[[Wikilinks]]`.

1. Agent-Friendly Six-Folder Vault Taxonomy

`00-Inbox/`
Default landing zone for all new agent drafts, web clips, and untriaged transcripts.

`01-Daily/`
Chronological `YYYY-MM-DD.md` daily logs and `YYYY-WXX-Review.md` rollups.

`02-Projects/`
Active outcome-driven initiatives with clear deadlines and `## Action Items` sections.

`03-Areas/`
Ongoing responsibilities (Health, Engineering, Finance, Team) with no end date.

`04-Resources/`
Atomic evergreen concept notes, paper summaries, and reusable reference guides.

`99-Archive/`
Completed projects and retired notes; agents treat as read-only historical context.

2. Production-Ready `~/Vault/AGENTS.md` Schema Contract

```
# Vault Operating Contract (AGENTS.md)
1. WRITE BOUNDARIES: Place all brand-new notes in 00-Inbox/ unless a target path (such as 01-Daily/YYYY-MM-DD.md)
   is explicitly requested. NEVER modify .obsidian/workspace.json, .obsidian/app.json, or .obsidian/plugins/.
2. YAML FRONTMATTER: Every note MUST begin with valid YAML frontmatter containing:
   created: "YYYY-MM-DD"        # Always double-quote strings containing colons (:) or brackets
   type: "daily" | "project" | "area" | "resource" | "meeting"
   status: "inbox" | "active" | "completed" | "archived"
   tags: [lowercase-kebab-tags]
   summary: "One-sentence plain-text takeaway under 140 characters"
3. WIKILINK POLICY: Use Obsidian [[Note Title]] or [[Note Title|Display Text]] links—never relative Markdown URLs.
   Before creating a link, search existing titles in 02-Projects/, 03-Areas/, and 04-Resources/ to prevent duplicates.
4. SURGICAL EDITS: On existing notes over 300 words, append or patch target sections instead of rewriting whole files.
```

3. Color-Coded Vault Prompt Anatomy

Combine five explicit building blocks whenever asking Claude Desktop or Antigravity Desktop to transform messy inputs into linked vault knowledge:

[Scope] Read only @00-Inbox/2026-11-12-customer-call.md and @02-Projects/ onboarding hub notes. **[Context]** Extract the three recurring onboarding friction points and concrete engineering follow-ups raised during today's enterprise call. **[Frontmatter Contract]** Follow AGENTS.md YAML schema with type: "meeting", status: "active", double-quoted summary, and tags: [onboarding, customer-research]. **[Link Policy]** Resolve existing note titles first; weave natural [[Note Title|Alias]] wikilinks to matching 02-Projects/ and 04-Resources/ notes without linking generic nouns. **[Write Target]** Save the clean meeting note to 00-Inbox/Customer-Call-Onboarding-Synthesis.md and surgically append owner-tagged checkboxes under ## Action Items in 02-Projects/Onboarding-Revamp.md.

Why `[Scope]` + `[Link Policy]` Matter
Explicitly naming source folders prevents 5,000-file fishing expeditions, while title-resolution before linking keeps your Obsidian Graph clean and connected rather than littered with ghost nodes.

Why `[Write Target]` Splits Output
Separating new synthesis (written to `00-Inbox/`) from action-item mutation (patched under `## Action Items`) gives you safe human review without losing momentum on active projects.

Native Obsidian CLI & Scheduled Daily/Weekly Rituals

Once registered via `Settings → General → Command line interface → Register CLI`, the official `obsidian` binary lets terminal scripts and Antigravity Scheduled Tasks drive the live Obsidian app directly.

1. High-Leverage Official `obsidian` CLI Commands

```
# Search vault index, open daily note in foreground, append tasks, and mutate typed YAML properties:
obsidian search query="tag:#onboarding status:active"           # Fast indexed search across live vault
obsidian daily                                                  # Opens (or creates from template) today's note
obsidian daily:append content="- [ ] Prep Q4 architecture review ([[Atlas-Launch]])"
obsidian property:set file="Atlas-Launch" name="status" value="in-review"
obsidian open "01-Daily/2026-11-12"                             # Focuses today's briefing on your desktop
```

2. Antigravity Scheduled Tasks: Weekday Briefing & Friday Synthesis

Use the `/schedule` command (or **Scheduled Tasks** in Antigravity Desktop's left sidebar) to run recurring cron jobs in fresh, timezone-pinned conversations that bridge Google Workspace plugins into your local vault:

1 Weekday Morning Brief (`0 7 * * 1-5`)

Schedule: `CRON_TZ=America/New_York 0 7 * * 1-5`
Queries today's events via `gcalendar`, locates linked prep decks via `gdrive`, carries forward unfinished `- [ ]` items from yesterday's `01-Daily/` note, writes `01-Daily/YYYY-MM-DD.md`, and calls `obsidian daily` so it greets you at your desk.

2 Friday Weekly Synthesis (`0 16 * * 5`)

Schedule: `CRON_TZ=America/New_York 0 16 * * 5`
Scans Monday–Friday notes in `01-Daily/` plus untriaged items in `00-Inbox/`, summarizes shipped wins, open blockers, and stale projects into `01-Daily/YYYY-WXX-Review.md` with `[[Wikilinks]]` to every touched project hub.



Quiet Dawn Automation: Scheduled weekday and Friday cron routines sort raw captures, calendar commitments, and open checkboxes into linked vault shelves before your workday begins.

Self-Contained Scheduled Prompts + Test Run Permissions

Every scheduled trigger launches a clean conversation with zero memory of yesterday's chat. Always specify exact vault paths (`/Users/alex/Vault/01-Daily/`) and run one interactive test pass first so file and `obsidian` CLI permissions are cleanly approved.

5 Real-World Copy-Paste Vault Prompts & Recipes

Drop these five prompts straight into Antigravity Chat Canvas, Scheduled Tasks, or Claude Desktop. Each recipe obeys `AGENTS.md` YAML laws, protects existing prose, and produces verifiable `[[Wikilinks]]`.

Workflow Recipe	Best Surface	Copy-Paste Prompt / Scheduled Instruction	Expected Vault Output
1. Morning Calendar + Open-Tasks Daily Builder	**Antigravity** Cron `0 7 * * 1-5` + `gcalendar` + CLI	"In `/Users/alex/Vault`, read yesterday's note in `01-Daily/` and collect unfinished `- [ ]` tasks. Pull today's schedule via `gcalendar` and matching docs via `gdrive`. Create `01-Daily/YYYY-MM-DD.md` following `AGENTS.md` YAML with `## Schedule`, `## Top 3 Priorities`, and `## Carried Tasks`, linking matching `02-Projects/` hubs, then run `obsidian daily`."	Ready-to-work daily note opened on screen with linked meetings & carried tasks.
2. Raw Transcript → Atomic Notes + Project Patch	**Claude Desktop** Cowork + `mcp-obsidian`	"Read `00-Inbox/raw-sync-transcript.md`. Create one clean meeting note in `00-Inbox/` and up to 2 atomic concept notes in `04-Resources/` with valid `AGENTS.md` YAML and `[[Wikilinks]]`. Then use `mcp-obsidian` `patch_content` to append owner-tagged `- [ ]` items under `## Action Items` in `02-Projects/Atlas-Launch.md` without rewriting the file."	Structured meeting + evergreen notes, plus surgical 60-token heading append.
3. `00-Inbox/` Triage & Frontmatter Normalizer	**Claude Cowork** or **Antigravity** `@00-Inbox/`	"Audit all `.md` files inside `@00-Inbox/`. Normalize missing or unquoted YAML frontmatter to match `@AGENTS.md` (quote any `summary` containing colons). Propose a routing table mapping each note to `02-Projects/`, `03-Areas/`, or `04-Resources/`, and wait for my approval before moving files."	100% valid Obsidian Properties across captures + human-gated folder filing.
4. Orphan Note Discovery & `[[Wikilink]]` Weaver	**Antigravity** Skill `link-orphans`	"Run `link-orphans` across `@04-Resources/` and `@02-Projects/` (exclude `.obsidian/` and `99-Archive/`). List notes with zero incoming or outgoing `[[Wikilinks]]`, match them to existing hub titles/aliases, preview 2–4 high-signal inline links per orphan, and apply after confirmation."	Isolated notes woven into Graph View with zero broken or spammy links.
5. Friday Weekly Review & Status Rollup	**Antigravity** Cron `0 16 * * 5`	"Read this week's 5 daily notes in `01-Daily/` and active hubs in `02-Projects/`. Write `01-Daily/YYYY-WXX-Review.md` summarizing: (1) Completed Shipments, (2) Slipped Tasks & Root Blockers, (3) Untriaged `00-Inbox/` count, and (4) Next Week's Focus, linking every cited note via `[[Wikilinks]]`."	Executive weekly retrospective linked to every active project hub.

Turn Any Table Row Into a One-Word Slash Workflow

Save your two favorite prompts from this page as `.agents/skills/inbox-triage/SKILL.md` and `.agents/skills/weekly-review/SKILL.md` inside your vault. Next time, simply type the skill name in Antigravity Chat Canvas (or reference it in Claude Cowork) instead of re-pasting instructions.

Protecting `.obsidian/` Configs & 10-Second Rollback Safety

Giving desktop AI agents write access to years of personal notes requires defense in depth. Three lightweight guardrails guarantee that an over-eager bulk edit can always be undone in under ten seconds.

1. Why Agents Must Never Touch `.obsidian/workspace.json` or Plugin Binaries

Inside your vault root sits the hidden `.obsidian/` configuration folder. While Obsidian is running, it continuously flushes pane geometry, cursor positions, and recent file state into `.obsidian/workspace.json`. If an agent edits that file mid-session, Obsidian immediately overwrites it or corrupts your split layout. Worse, allowing agents to modify `.obsidian/plugins/*/main.js` introduces arbitrary JavaScript execution inside a desktop Electron app. Treat `.obsidian/` as strictly off-limits unless you explicitly ask for a CSS snippet inside `.obsidian/snippets/`.

Layer 1: Hard Rule + Perms

Forbid `.obsidian/` writes in `AGENTS.md` and configure Antigravity **Project-Level Permission Grants** to require manual review outside `00-Inbox/` and `01-Daily/`.

Layer 2: File Recovery

Open Obsidian `Settings → Core plugins → File recovery`. Set **Snapshot interval** to `5` minutes and **History length** to `30` + days for visual side-by-side diffs and 1-click single-note restore.

Layer 3: Git Checkpoints

Initialize Git in `~/Vault`. Taking a two-second snapshot commit before any multi-file refactor gives you atomic diff inspection (`git diff`) and instant vault-wide undo.

2. Clean Vault `.gitignore` + Pre-Refactor Checkpoint Ritual

```
# ~/Vault/.gitignore — Track notes, templates, AGENTS.md & .agents/ skills; ignore volatile UI state:
.obsidian/workspace.json
.obsidian/workspace-mobile.json
.obsidian/cache
.trash/
.DS_Store

# Run before any bulk inbox sweep, frontmatter migration, or multi-note wikilink pass:
git add -A && git commit -m "checkpoint: pre-agent vault sweep"
# Inspect every changed line afterward — or roll back the entire batch in 2 seconds:
git diff --stat                 # Review exact notes touched
git checkout -- 02-Projects/    # Instant rollback if a batch edit over-reached
```

3. Privacy Scoping for Sensitive Personal Journals

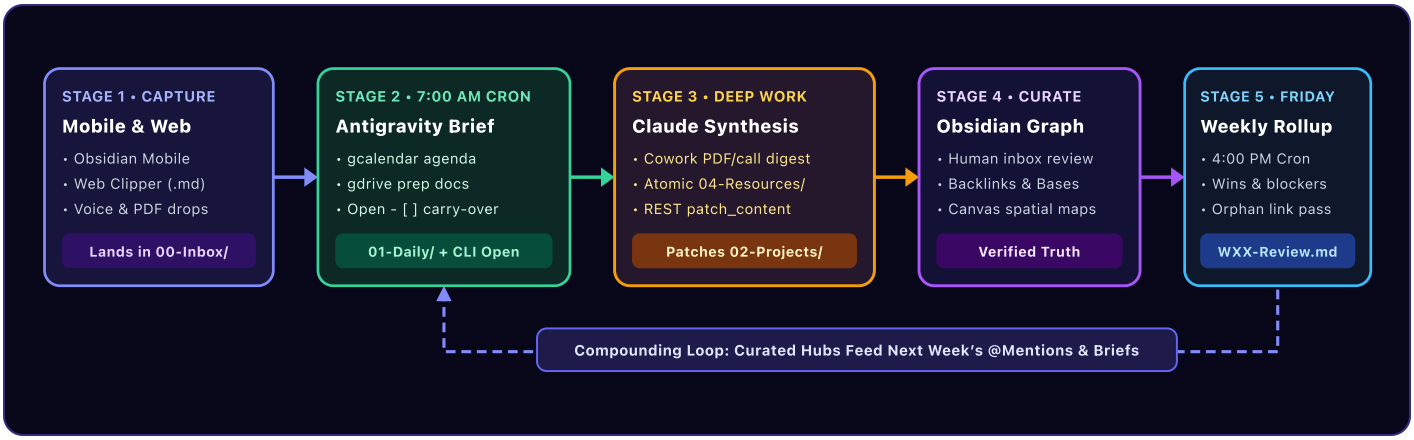
Sensitivity Level	Recommended Vault Folder Architecture	How Agents See It
Work & Research Notes	Standard vault folders (`00-Inbox/` through `04-Resources/`)	Full Project / Cowork + MCP read & scoped write access.
Private Journals / Therapy	Separate Obsidian vault (`~/Private-Vault/`) or sibling folder excluded from MCP roots	Zero cloud token transmission; completely invisible to agent file tools.
Credentials & API Keys	OS Keychain or `~/.zshrc` environment variables—never plain Markdown notes	Passed only via MCP `env` config blocks, never indexed in prose.

Cloud Sync Harmony (Obsidian Sync, iCloud Drive & Git)

If your vault syncs across laptop and phone via **Obsidian Sync** or **iCloud Drive**, enable iCloud's *Keep Downloaded* option on the vault folder so agents never hit stubbed `.icloud` placeholder files during local scans.

Cross-Tool Synergy: The Closed-Loop Second Brain

When mobile capture, Google Workspace signals, Antigravity scheduled routines, Claude Desktop deep synthesis, and Obsidian's visual graph feed one another, your vault compounds in value every single week.



Walking Through a Full Day & Week Across Tools

1 Frictionless Capture → Scheduled Prep

Throughout the day, clip articles via Obsidian Web Clipper or jot quick thoughts on Obsidian Mobile into `00-Inbox/`. At 7:00 AM, Antigravity Desktop wakes on cron, pulls your Google Calendar meetings and linked Drive docs, carries forward yesterday's open tasks, writes `01-Daily/YYYY-MM-DD.md`, and pops it open via `obsidian daily`.

2 Midday Deep Synthesis & Surgical Patching

After a customer interview or design review, drop the transcript or PDF into Claude Desktop. Cowork distills evergreen insights into `00-Inbox/` while `mcp-obsidian` surgically appends concrete next steps under `## Action Items` in your active `02-Projects/` note without disturbing existing writing.

3 Human Curation in Graph, Bases & Canvas

Spend five calm minutes reviewing `00-Inbox/` inside Obsidian. Verify YAML properties in Obsidian Bases/tables, inspect Backlinks, drag promoted notes into `02-Projects/` or `04-Resources/`, and arrange high-level architecture visually on an Obsidian Canvas board.

4 Friday Synthesis & Compounding Memory

At 4:00 PM Friday, Antigravity runs your weekly synthesis and `link-orphans` skill, generating `YYYY-WXX-Review.md`. Next Monday, when you type `@02-Projects/` in Chat Canvas, both agents inherit cleaner, richer context than the week before.

The Human-in-the-Loop Sovereignty Principle

AI agents should handle 90% of clerical formatting, calendar gathering, action-item extraction, and link discovery—while *you* remain the editor-in-chief who promotes notes out of `00-Inbox/`. That boundary keeps your Second Brain trustworthy instead of turning into an unread AI text dump.

Troubleshooting Clinic & 6-Point Pre-Flight Checklist

Nearly every hiccup between Obsidian, Claude Desktop, and Antigravity Desktop traces back to one of seven local configuration snags—each solvable in ten seconds.

Symptom / Error	Root Cause	10-Second Fix
1. Local REST API TLS error or port `27124` refused	Obsidian is closed, plugin disabled, or custom client rejects localhost self-signed cert.	Keep Obsidian running, verify **Local REST API** is enabled on `127.0.0.1:27124`, use `mcp-obsidian` (handles local cert), or test with `curl -k`.
2. Broken `[[Wikilinks]]` or duplicate note titles	Agent guessed note names or emitted Markdown URLs instead of resolving vault titles.	Add title-search rule to `AGENTS.md`: run `obsidian search` or directory list across `02-Projects/` & `04-Resources/` before writing links.
3. YAML frontmatter colons break Properties UI	Unquoted colons in `summary: Q4 Review: Phase 1` break Obsidian's strict YAML parser.	Enforce double quotes around all string values in `AGENTS.md` (`summary: "Q4 Review: Phase 1"`) and run Recipe #3 on broken files.
4. Constant `.obsidian/workspace.json` churn	Obsidian writes tab/scroll state every few seconds; agents or Git keep flagging diffs.	Add `.obsidian/workspace*.json` and `.obsidian/cache` to `.gitignore` and forbid `.obsidian/` edits in `AGENTS.md`.
5. macOS TCC blocks `Documents` / `Desktop` vault	macOS privacy sandbox denies background `node` / `uvx` access to TCC-protected folders.	Move vault to `~/Vault/` (home folder avoids TCC prompts) or grant Full Disk / Files & Folders access to Claude & Antigravity in System Settings.
6. Stale MCP Node / `uvx` processes after edits	Changing `OBSIDIAN_API_KEY` or paths in JSON does not hot-reload spawned stdio children.	Fully quit Claude Desktop (`Cmd+Q`), run `pkill -f mcp-obsidian` if orphaned, and relaunch to spawn fresh MCP workers.
7. Context window overload on `5,000+` note vaults	Broad prompt triggers full-vault file reads across years of archives and attachments.	Scope prompts with `@00-Inbox/` or `@02-Projects/` mentions, keep `99-Archive/` excluded by default, and query via `obsidian search` first.

6-Point Pre-Flight Checklist Before Launching Your Dual-Agent Vault

☑ **1. Vault Outside OS Traps:** Stored at a clean local path (such as `~/Vault/`) with *Keep Downloaded* enabled if synced.

☑ **2. Numbered Taxonomy Ready:** `00-Inbox/` through `99-Archive/` created so every agent draft has a deterministic home.

☑ **3. `AGENTS.md` at Root:** Defines double-quoted YAML keys, `[[Wikilink]]` title resolution, and `.obsidian/` off-limits law.

☑ **4. Both Live Bridges Verified:** Local REST API active on `127.0.0.1:27124` and `obsidian version` returning `1.12.4+`.

☑ **5. Double Safety Net Armed:** Obsidian File Recovery set to 5-minute snapshots + `.gitignore` initialized for pre-sweep commits.

☑ **6. First Scheduled Ritual Tested:** Weekday `0 7 * * 1-5` briefing dry-run approved once interactively in Antigravity Desktop.