

AWS COMMUNITY DAY DAVAO 2026 · HANDS-ON WORKSHOP

# Build Your Own Context

A Personal MCP Server with S3, Lambda,  
API Gateway and Cognito

---

PRESENTED BY

**Arlou A. Beloria**

OPEN THE ATTENDEE GUIDE

[www.arloubeloria.com/workshop/vault-brain](http://www.arloubeloria.com/workshop/vault-brain)

DID YOU KNOW?

# A Markdown file can carry a world

Build Your Own X organizes practical knowledge in a Markdown file, not a complex application. Vault Brain follows the same idea for your context, with MCP as the access layer.

## Build your own **Distributed Systems**

- [Java: Building Your Own Kafka-like System From Scratch: A Step-by-Step Guide](#)

## Build your own **3D Renderer**

- [C++: Introduction to Ray Tracing: a Simple Method for Creating 3D Images](#)
- [C++: How OpenGL works: software rendering in 500 lines of code](#)
- [C++: Raycasting engine of Wolfenstein 3D](#)
- [C++: Physically Based Rendering: From Theory To Implementation](#)
- [C++: Ray Tracing in One Weekend](#)

THE PROBLEM

# The context tax

You explain your projects to one AI client. When you switch, the next client starts without that history.

**Repeated explanation costs time.**

# What is a Vault Brain?

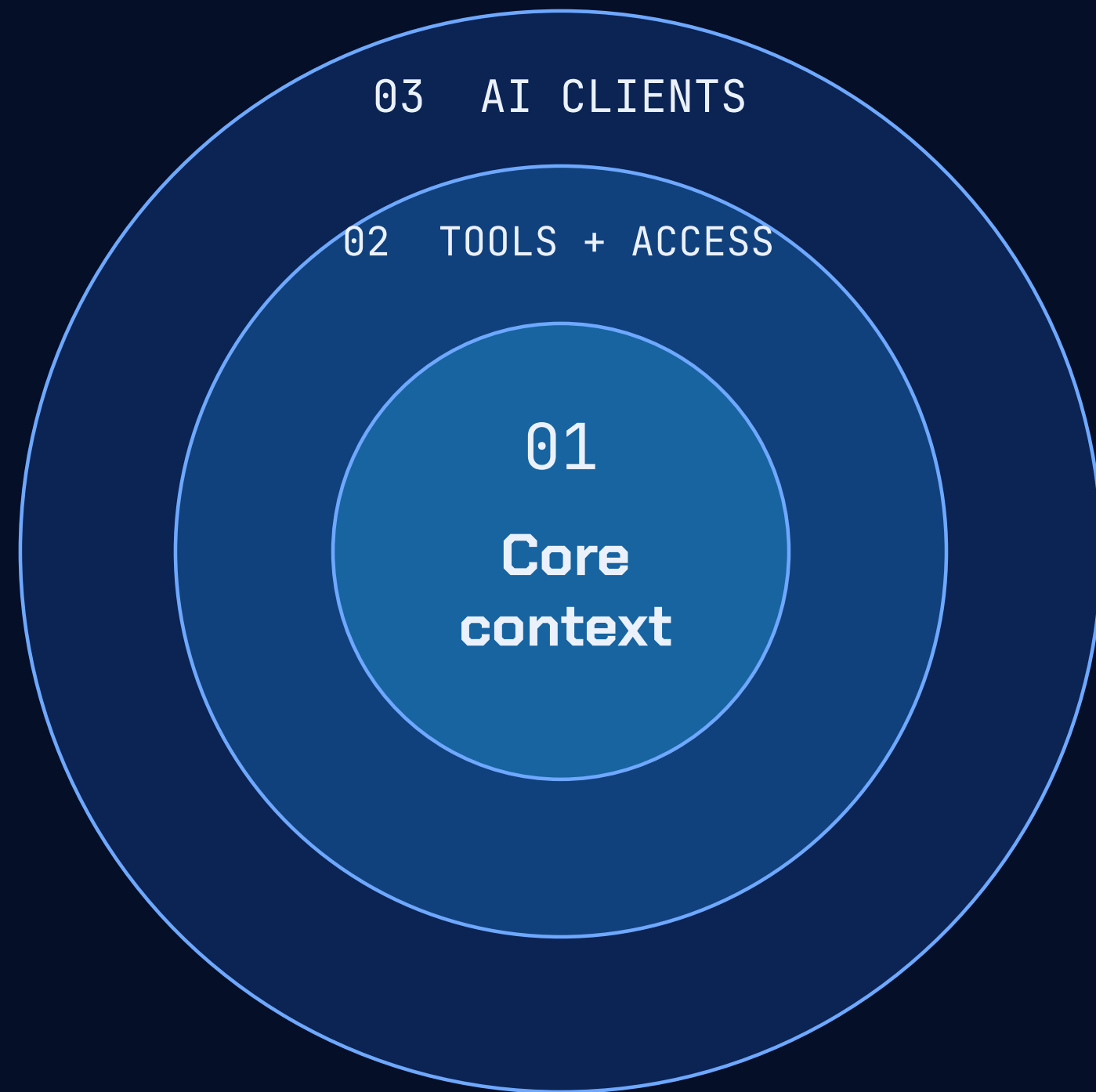
**Your durable context lives as Markdown in your own S3 bucket.**

---

MCP gives AI clients controlled access.

Obsidian lets you read and edit the same notes yourself.

# Three layers, three jobs



## AI clients

Claude web, ChatGPT, future MCP clients

---

## Tools and access

Vault Brain MCP and Cognito

---

## Core context

Readable Markdown in private Amazon S3

Obsidian syncs directly with the same S3  
Markdown.

# Switch clients, keep the context

---

01      **Keep the vault in your AWS account**

---

02      **Connect another MCP-capable client**

---

03      **Continue from the same Markdown**

---

I moved from Claude to OpenAI without rebuilding my notes.

THE ANALOGY

# The MCP server is the librarian

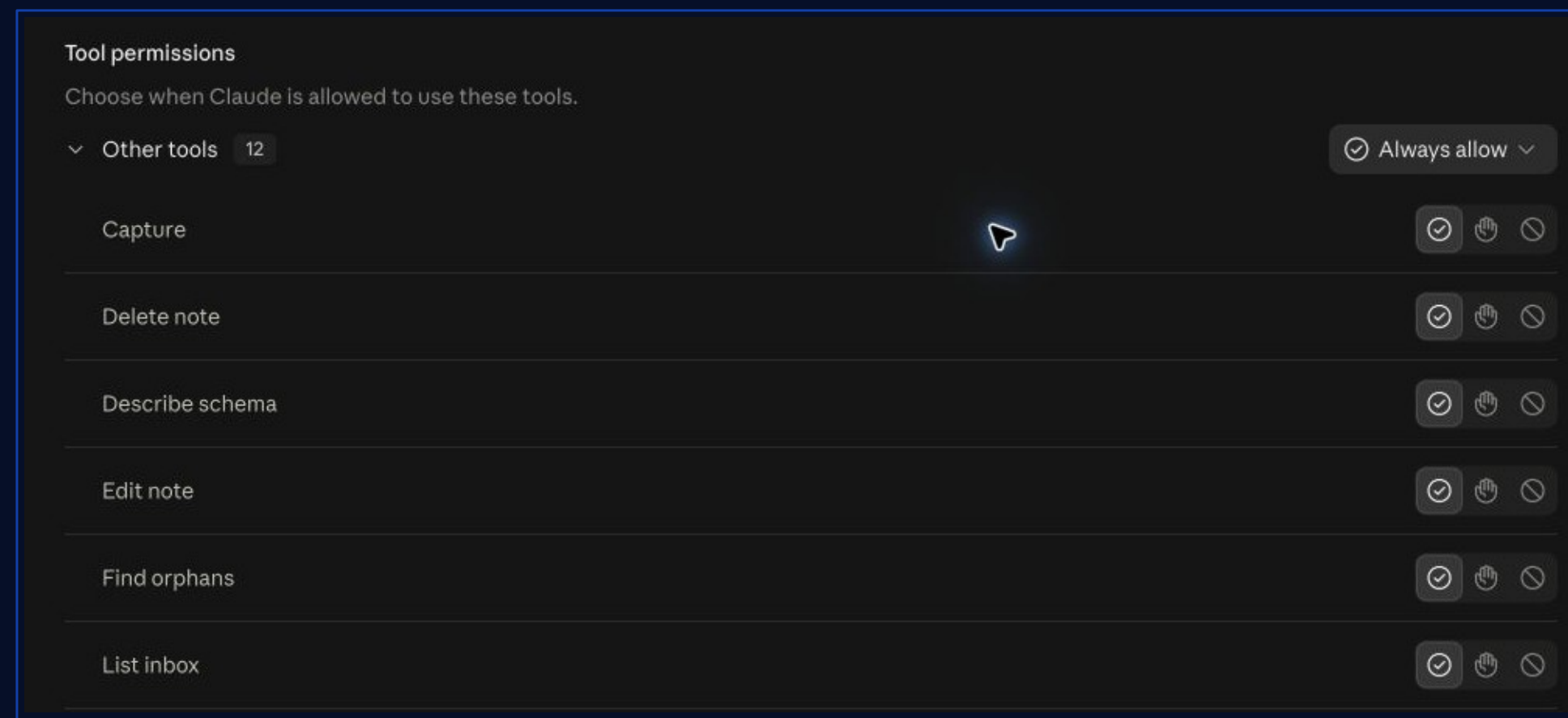
The vault holds the books. The AI client asks the server to find, read, or file something.

MCP is the common language for that request.

# The client discovers tools

## tools/list

The server describes the tools it offers.



REAL CLAUDE WEB CAPTURE • 12 CONNECTED TOOLS

Review write and delete permissions before use.

ONE TOOL CONTRACT

## search\_vault

description

Find text inside notes

input schema (simplified)

```
{ query: string }
```

The client supplies this contract alongside your message.

Tool contracts contain no AWS keys or handler code.

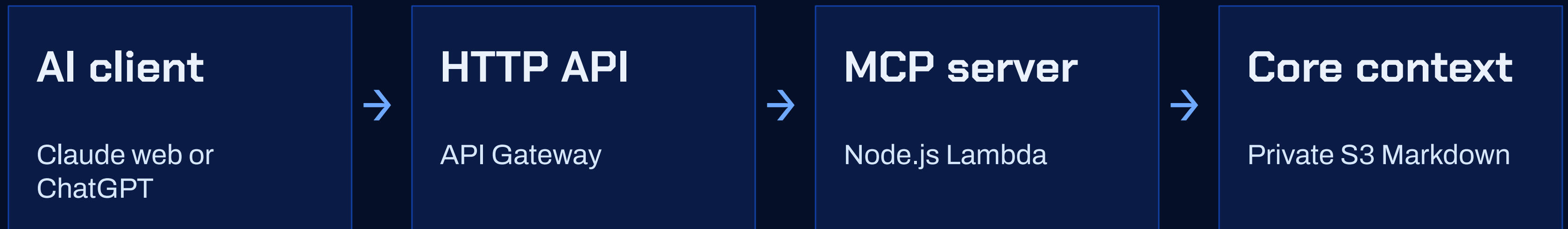
# Under the hood of one tool call

“Find blue mango 42 in my notes.”

|    |                        |                                                       |
|----|------------------------|-------------------------------------------------------|
| 01 | <b>Client provides</b> | <code>your message + tool contracts</code>            |
| 02 | <b>Model selects</b>   | <code>search_vault({ query: "blue mango 42" })</code> |
| 03 | <b>Client sends</b>    | <code>tools/call</code> with those arguments          |
| 04 | <b>Server returns</b>  | matching Markdown snippets                            |
| 05 | <b>Model continues</b> | tool result joins conversation context                |

Structured inputs, not one pasted prompt.

# The deployed architecture



---

## Cognito

OAuth sign-in and bearer access

## Obsidian

Remotely Save syncs directly with S3

Terraform state lives in a separate protected S3 bucket.

# Today's route

START OF SESSION

01

**Preflight**

Check tools and AWS

THEN BUILD LIVE

02

**Deploy**

Your AWS stack

03

**Connect**

Claude web or ChatGPT

04

**Sync**

Obsidian and S3

STEP-BY-STEP GUIDE

[www.arloubeloria.com/workshop/vault-brain](http://www.arloubeloria.com/workshop/vault-brain)

# What you will leave with

- 
- 01 A private Markdown vault in your AWS account
  - 02 One web AI client connected to Vault Brain tools
  - 03 A note your AI saves to the vault and reads back
  - 04 The same files visible and editable in Obsidian
-

# Prepare before the workshop

Have these ready. We clone and run preflight together during the session.

|    |                      |                                                                                                                                            |
|----|----------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 01 | <b>Software</b>      | Git, AWS CLI, Terraform 1.10+, and Node.js 22+.                                                                                            |
| 02 | <b>AWS access</b>    | A personal account and named CLI profile with IAM, S3, Lambda, API Gateway, Cognito, and CloudWatch Logs access.                           |
| 03 | <b>Template</b>      | Keep <a href="https://github.com/Arlovzki/vault-brain">github.com/Arlovzki/vault-brain</a> handy. We clone it during the session.          |
| 04 | <b>AI + Obsidian</b> | Sign in to Claude web or ChatGPT; confirm you can add a custom MCP connection. Install Obsidian; confirm Community plugins can be enabled. |

USE SAFE SAMPLE DATA

Use one small, non-sensitive note. No passwords, API keys, customer data, or private company material.

GUIDE [www.arloubeloria.com/workshop/vault-brain#pre-flight](https://www.arloubeloria.com/workshop/vault-brain#pre-flight)

CHECK

# Starting checkpoint

Run this check together before building. Use claude-web or chatgpt for CLIENT.

---

|    |                    |                                                                    |
|----|--------------------|--------------------------------------------------------------------|
| 01 | Clone the template | <code>git clone https://github.com/Arlovzki/vault-brain.git</code> |
|----|--------------------|--------------------------------------------------------------------|

---

|    |                  |                             |
|----|------------------|-----------------------------|
| 02 | Enter the folder | <code>cd vault-brain</code> |
|----|------------------|-----------------------------|

---

|    |               |                                                                                    |
|----|---------------|------------------------------------------------------------------------------------|
| 03 | Run preflight | <code>npm run preflight -- --fix --profile YOUR_AWS_PROFILE --client CLIENT</code> |
|----|---------------|------------------------------------------------------------------------------------|

---

Checkpoint: Git, AWS, Terraform, Node.js 22, one web AI client, and Obsidian are ready.

GUIDE [www.arloubeloria.com/workshop/vault-brain#clone-template](http://www.arloubeloria.com/workshop/vault-brain#clone-template)

# How the codebase is organized

The repository has four jobs: context, tools, infrastructure, and the workshop runner.

## REPOSITORY MAP

---

|                      |                                         |
|----------------------|-----------------------------------------|
| vault-brain/         |                                         |
| ├── vault-starter/   | Markdown notes and vault rules          |
| ├── mcp-server/      |                                         |
| │   └── src/         |                                         |
| │       ├── index.ts | MCP request routing                     |
| │       ├── tools.ts | 12 tool contracts and S3 handlers       |
| │       ├── auth.ts  | Cognito and bearer access               |
| │       ├── dcr.ts   | OAuth client registration               |
| │       └── day.ts   | Vault-local dates                       |
| ├── state-bootstrap/ | Protected Terraform state bucket        |
| ├── terraform/       | Vault stack and remote S3 backend       |
| ├── package.json     | Cross-platform workshop commands        |
| └── scripts/         |                                         |
| └── workshop.mjs     | Configure, preflight, deploy, and smoke |

# The code behind a tool call

GITHUB SOURCE · [mcp-server/src/tools.ts](#)

```
20
21  ✓ export interface Tool {
22      name: string;
23      description: string;
24      inputSchema: Record<string, unknown>;
25      handler: (args: Record<string, unknown>) => Promise<string>;
26  }
```

ONE TOOL CALL

[index.ts](#)

MCP routes



[tools.ts](#)

Contracts and handlers



[S3 vault](#)

Markdown files

[auth.ts](#) checks access

[dcr.ts](#) handles /register

[day.ts](#) supplies local dates

## ACTIVITY

# Configure your stack and state

Create local config, verify your AWS account, then bootstrap remote state.

- |    |                     |                                                   |
|----|---------------------|---------------------------------------------------|
| 01 | Create config       | <code>npm run configure</code>                    |
| 02 | Set personal values | <code>Edit terraform/terraform.tfvars</code>      |
| 03 | Validate setup      | <code>npm run preflight -- --client CLIENT</code> |
| 04 | Bootstrap state     | <code>npm run bootstrap-state</code>              |

Checkpoint: The main Terraform stack is connected to its remote S3 backend.

GUIDE [www.arloubeloria.com/workshop/vault-brain#configure-stack](http://www.arloubeloria.com/workshop/vault-brain#configure-stack)

## ACTIVITY

# Deploy and verify

Build the Lambda and deploy S3, API Gateway, Cognito, IAM, and CloudWatch.

---

|    |                   |                             |
|----|-------------------|-----------------------------|
| 01 | Deploy your stack | <code>npm run deploy</code> |
|----|-------------------|-----------------------------|

---

|    |                      |                                   |
|----|----------------------|-----------------------------------|
| 02 | Set Cognito password | Follow the hidden terminal prompt |
|----|----------------------|-----------------------------------|

---

|    |                    |                            |
|----|--------------------|----------------------------|
| 03 | Run the smoke test | <code>npm run smoke</code> |
|----|--------------------|----------------------------|

---

Checkpoint: The MCP endpoint reports all 12 tools and the smoke test passes.

GUIDE [www.arloubeloria.com/workshop/vault-brain#run-deployment](https://www.arloubeloria.com/workshop/vault-brain#run-deployment)

# How sign-in protects the vault

Your AI client signs in before it can use a vault tool.

- 
- |    |                            |                                   |
|----|----------------------------|-----------------------------------|
| 01 | <b>You sign in</b>         | Cognito checks your owner account |
| 02 | <b>Client gets access</b>  | Cognito issues an access token    |
| 03 | <b>A tool is requested</b> | Lambda verifies the token first   |
- 

PKCE protects the sign-in code exchange. Your client handles it.

## ACTIVITY

# Connect one web AI client

Choose Claude web or ChatGPT. Both connect through the same authenticated /mcp URL.

- |    |                   |                                             |
|----|-------------------|---------------------------------------------|
| 01 | Choose one path   | Claude web or ChatGPT                       |
| 02 | Add the connector | Paste the deployed HTTPS URL ending in /mcp |
| 03 | Complete sign-in  | Use Cognito, then confirm the tools appear  |

Checkpoint: Your chosen client can see and use the Vault Brain tools.

GUIDE [www.arloubeloria.com/workshop/vault-brain#choose-client](https://www.arloubeloria.com/workshop/vault-brain#choose-client)

## ACTIVITY

# The four-prompt test

Use the full prompts in the guide to capture, find, and read one test note.

---

|    |     |                                    |
|----|-----|------------------------------------|
| 01 | Ask | What is waiting in my vault inbox? |
|----|-----|------------------------------------|

---

|    |         |                                      |
|----|---------|--------------------------------------|
| 02 | Capture | Save a note containing blue mango 42 |
|----|---------|--------------------------------------|

---

|    |      |                                     |
|----|------|-------------------------------------|
| 03 | Find | Find the exact phrase blue mango 42 |
|----|------|-------------------------------------|

---

|    |      |                                   |
|----|------|-----------------------------------|
| 04 | Read | Read the complete note back to me |
|----|------|-----------------------------------|

---

Checkpoint: Your client reports a saved path, finds the phrase, and reads the note.

GUIDE [www.arloubeloria.com/workshop/vault-brain#core-prompt-loop](http://www.arloubeloria.com/workshop/vault-brain#core-prompt-loop)

## ACTIVITY

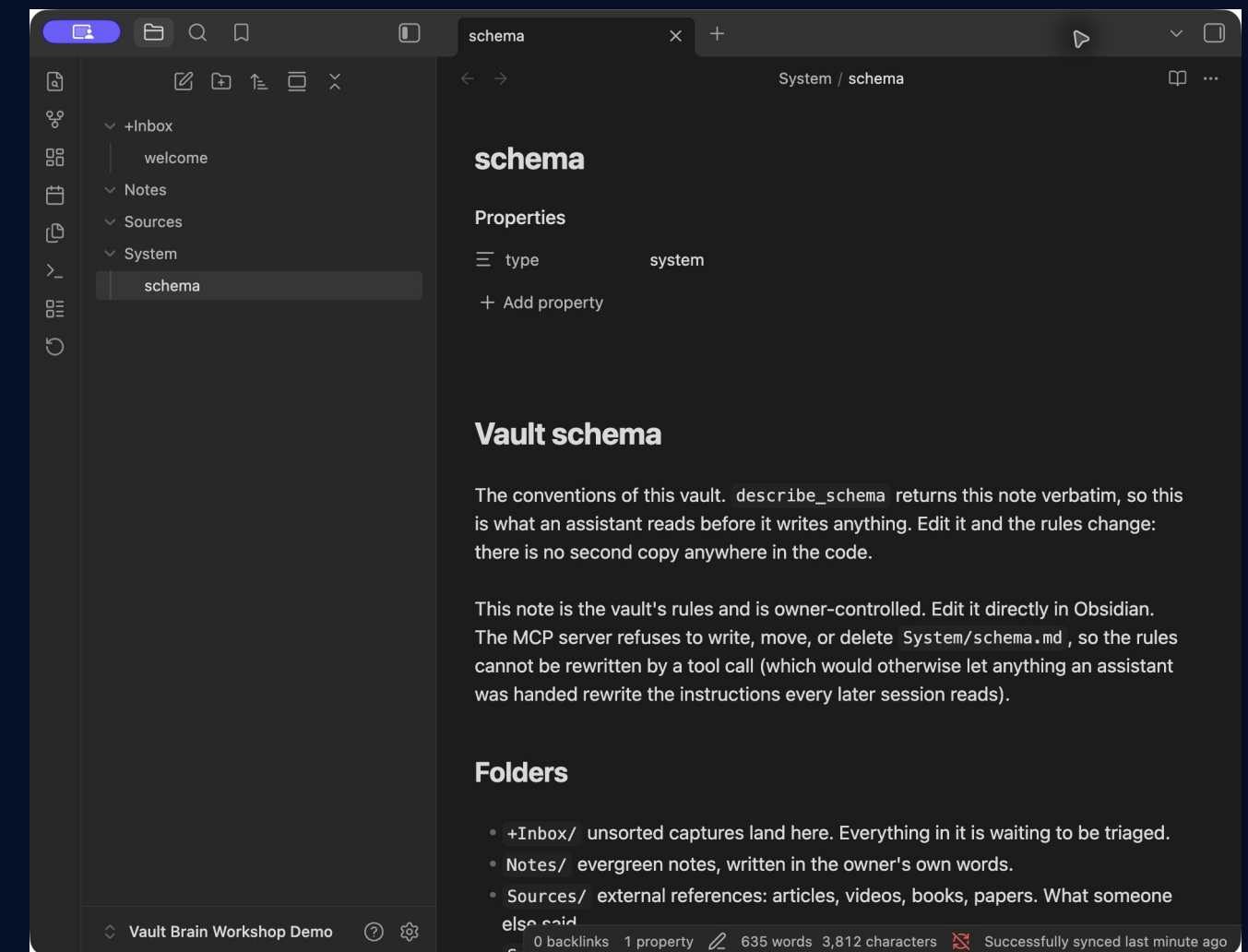
# Obsidian closes the loop

Sync the same Markdown in both directions.

01 Create an empty vault

02 Enable Community plugins

03 Set up Remotely Save for S3



REAL SYNCED VAULT

Checkpoint: the AI finds your Obsidian note, and Obsidian shows the AI-created note.

GUIDE [www.arloubeloria.com/workshop/vault-brain#obsidian-guide](http://www.arloubeloria.com/workshop/vault-brain#obsidian-guide)

# Keep going

Your memory has an address you control.

[www.arloubeloria.com/workshop/vault-brain](http://www.arloubeloria.com/workshop/vault-brain)

---

BONUS AFTER THE WORKSHOP

**Capture Claude Code or Codex session summaries in the vault.**

Questions?