Alice in Kerneland — Portable Arch + Obsidian USB

1) Overview

A single USB boots **Arch anywhere** UEFI/BIOS and auto-mounts a **VAULT** partition holding your **Obsidian vault**, portable **Firefox/Chromium** profiles, and a lean **SQLite** DB. A minimal **archiso** brings up networking, mounts VAULT, sets `XDG_*` to the stick, launches Obsidian on the vault, and exposes browser launchers that keep all data on VAULT. Optional **LUKS** on VAULT. Target **1.5 GB RAM idle**; keep daemons to a minimum.

> **Build items:** archiso recipe; VAULT partition/encryption script; systemd-user units to auto-open Obsidian; portable browser launchers.
>
> **Stack:** Arch/archiso, systemd-boot + syslinux, i3 or Sway, Obsidian AppImage, SQLite, systemd-user, udev. *No Rust; tiny x86-64 asm only if truly needed.)*

2) Members

Mark Evgenev (mevgenev) - Engineer, focused on **Obsidian Vaults**

Matt Daly (mdaly22) - Engineer, focused on **Linux environment**

3) Capture & Data Model

One **SQLite** file on VAULT with two tables:

> **pages(url, title, text_md, captured_at, links_json, source_browser)**
> **notes(id, title, tags, text_md, links_json, updated_at)**

A cross-browser **WebExtension** adds a toolbar button that runs **Readability.js Turndown** to capture clean Markdown into `00_Inbox/` and inserts a row into SQLite via a tiny **Native-Messaging** host Python; optional micro-C helper). Inside Obsidian, a plugin (**VaultPipe**) exposes DB-backed views Inbox, URL index, backlinks), a minimal **task tracker** (checkboxes + due dates in frontmatter), and an in-app **browser pane** using Electron's BrowserView for quick clip-and-annotate.

> **Build items:** WebExtension Chromium Firefox); Python native host; SQLite schema/migrations; Obsidian plugin Node/Electron); optional DuckDB read-only analytics.
>
> **Stack:** WebExtensions Readability Turndown, Python 3 `sqte3` , Node/Electron plugin API, tiny C/asm utility if necessary.

4) Operations & Everyday Use

Plug in VAULT mounts Obsidian opens your vault. Use **chromium-usb**/**firefox-usb** for portable browsing (profiles on VAULT. A small **CLI** ("`ace` ") provides: `cp` (clipboard → note+DB, `task` (add/list), `sync` Obsidian-Git or rsync), `vey` (integrity/free space), `eject` (clean power-off). Works as a live desktop or as a pure data stick on any host; add Ventoy later for multi-ISO without touching VAULT.

> **Hygiene:** tune journaling/logging to reduce writes; default offline; everything local.
>
> **Backups:** Obsidian-Git or `sync` from VAULT to any destination.

5) Install / Flash Plan (simple)

> **Build ISO** minimal archiso with NetworkManager, i3/Sway, Obsidian AppImage, Chromium/Firefox, `sqte3` , `gt` , and your user-services.
>
> **Partition stick:** keep the Arch ISO (or full install), add a second **VAULT** partition (ext4 or exFAT; optionally LUKS. **First boot:** systemd-user path unit detects `/un/meda/$USER/VAULT` and launches Obsidian; browser launchers point `-- poe` / `--use-data-d` to VAULT.

Validate: ensure UEFI/BIOS boot, network OK, Obsidian opens vault, WebExtension can save to DB, CLI commands run.

5) Security, Resource Constraints & Portability

Resource targets: 1.5 GB idle, minimal services, lightweight WM (i3/Sway), no background indexers. **IO discipline:** move caches/profiles to VAULT, compress logs or rotate aggressively, avoid write-heavy apps.

Security: optional LUKS on VAULT; all capture scripts run locally; no cloud dependency by default.

Portability: boots on Intel/AMD, UEFI/Legacy; can coexist with Ventoy; use exFAT on VAULT if you need read/write on non-Linux hosts.

No Rust: keep builds in Bash/Python/Node; allow small, surgical x86-64 asm only for hardware checks or micro-optimizations.

Quick Build Checklist (bullet summary)

ISO minimal archiso + systemd-user units Obsidian AppImage.

VAULT second partition (ext4/exFAT; optional LUKS, contains `obsdan-vaut/` , `db/` , `.poes/` for browsers. **DB** SQLite with `pages` and `notes` ; mirror key rows to Markdown files.

Capture: WebExtension Native-Messaging Python) → write `.md` + insert into SQLite.

Obsidian plugin: “VaultPipe” for DB-backed views, task tracker, in-app browser pane.

CLI `ace cp|task|sync|vey|eject` .

Ops: offline-first, low-writes, Git/rsync backup, Ventoy optional.