Design File: Project Cartesian

Core Concept & Philosophy

The Metaphor: "Cartesian Theater" / Homunculus.
The Goal: A custom Linux Distribution where the Desktop Environment is a single cohesive Rust binary managing both AI and UI.
Differentiation: The AI is not a sidebar; it is the System Manager. However, the user retains full manual sovereignty via traditional UI paradigms (files, windows, settings).

Core Philosophy:

- Anything the AI can do the user can do too.
- The AI can only manage the computer through an interface of human designed scripts.

Features

AI Manager

The "Two-Mind" Architecture: The AI is not a single entity, but a split consciousness to balance performance and intelligence.

- **The Manager (GPU):** A heavy, intelligent model (Gemma 9B) that handles complex reasoning, script generation, and system maintenance. It runs only when the system is idle or under light load.

- **The Sidekick (CPU):** A lightweight, low-latency model (Gemma 2B) that handles quick queries, chat, and wiki lookups. It resides in System RAM to leave the GPU free for gaming.

The Interface: Macros, Not Commands To ensure safety, the AI cannot run raw terminal commands (e.g., rm -rf). Instead, it interacts via a **Macro Registry:**

- **Pre-defined Skills:** A library of safe, human-verified scripts (e.g., clean_cache, organize_files).
- **The "Permission Gate":** High-risk actions (Red Blocks) require explicit physical confirmation from the user (a "Launch Key" UI element).

Personality Profiles:

- Users can select the “vibe” of the AI (eg. Professional, Sassy, Robotic, Stoic).
- **Implementation:** Swappable System Prompts injected into the context window.

The "Witness" System (Context & Recording)

Dual-Pipeline Architecture: To ensure zero performance impact on gaming, the vision system is split:

- **The Replay Buffer:** Uses dedicated Hardware Encoders (NVENC) to write compressed video to RAM for user clipping.

- **The AI Eye:** Uses DMA-BUF (Zero Copy) to read raw GPU textures directly for context sampling, bypassing the encoding/decoding overhead entirely.

AI Context Sampling:

- **Routine:** The Vision AI samples a frame every 30 seconds to understand user activity.
- **Event-Driven:** Switching applications triggers an immediate sample.
- **On-Demand:** Asking the AI a question triggers an immediate sample.

Privacy Lock: This buffer is locked in RAM and inaccessible to filesystems until the user or AI explicitly "Clips" it.

Memory Architecture (The Hippocampus)

Vector Database: All interactions and context samples are stored as vector embeddings.

The Decay Algorithm: A heuristic function mimics human forgetfulness.

- **Short-term:** High detail, fades quickly (days).
- **Long-term:** Summarized concepts, fades slowly (months).
- **Core Memories:** Users or the AI can flag specific events/facts as "Core," protecting them from decay (e.g., "User is a Python developer").

Encrypted Neural Backup:

- Ability to export the Vector DB and User Preferences as a single AES-256 encrypted file for backup or migration between machines.

Audio Channels

Philosophy: Windows handles audio as a single "Mix." Project Cartesian handles audio as a "Studio Console."

Technical Implementation (PipeWire):

- **Virtual Sinks:** The OS creates persistent virtual audio cables: Game, Voice, Media, System, and Aux.
- **AI Routing:** When an app launches, the Sidekick analyzes its metadata (e.g., spotify.exe, discord, cs2). It automatically "patches" the audio output to the correct channel.
    - **Example:** You launch a new indie game. The AI recognizes it as a game and routes it to the Game channel.
- **The Mixing Board:** The "Control Room" UI features a persistent 5-fader mixer. The user can mute "Music" globally without Alt-Tabbing into Spotify.

Built In Recording Software

The Goal: A zero-setup recording suite that leverages the **Audio Channels** to create professional-grade clips without post-processing.

Features:

- **Iso-Track Recording:** Because audio is pre-split (see above), the recorder captures Game Audio and Microphone onto separate tracks, while completely ignoring Music (copyright safety).

- **"Witness" Mode (AI Clipping):**
    - The **Vision Sidekick** (Moondream model) watches the screen buffer.
    - When the user presses the "Clip That" button, the AI saves the last 30 seconds.
    - AI Enhancement: The AI auto-generates a title and tags for the clip based on what it saw (e.g., "Triple Kill in Valorant").

Universal Store & Gaming

The Meta-Store: A unified interface aggregating content from Steam, Epic (via Heroic), Flatpak, and System Packages.

Gaming Compatibility: Pre-integrated **Valve Proton** and **DXVK** layers. The OS detects Windows executables and automatically configures the compatibility environment, borrowing optimization profiles from the Steam Deck community.

The Macro Forge

The Goal: Democratize system automation. Users shouldn't need to know Python to teach the OS new tricks.

Functionality:

- **Visual Editor:** A "Scratch-like" block coding interface where users drag-and-drop actions (e.g., [Find File], [Wait], [Move to Folder]).
- **The Store:** A community repository where users upload Macros.
- **Safety Rating:**
    - **Green Tier:** Safe, read-only macros.

    - **Red Tier:** Macros that modify files or system settings. These trigger a warning and require manual review by the user before running.

The Resource Governor

The Goal: Ensure the AI never causes lag in games or heavy applications.

Logic:

- **State Detection:** The OS monitors the process list for Fullscreen 3D applications.
- **Dynamic Lobotomy:**
    - **Gaming Detected:** The Heavy "Manager" AI is instantly unloaded from VRAM. The Lightweight "Sidekick" is pinned to the CPU Efficiency Cores (E-Cores).
    - **Panic Switch:** If the OS detects a frametime spike (>20ms variance), it sends a SIGSTOP signal to the AI, freezing it instantly to prioritize the game.

The “Black Box” Audit Log

The Goal: Total accountability for the AI's actions.

Functionality:

- **Immutable Log:** A read-only text file that records every action the AI takes.
- **Format:** [TIMESTAMP] [AI_AGENT] [ACTION] [TARGET].
    - **Example:** 2025-12-14 14:00:01 [MANAGER] [MOVED_FILE] /home/user/desktop/image.png -> /home/user/pictures/

- **Rollback:** The user can right-click any log entry to "Undo" that specific action (using the send2trash recovery logic).

Fail-Safes & Accessibility

Safe Mode (Lobotomy Boot):

- A bootloader option to start the OS with the AI Daemon completely disabled. Useful if the AI configuration becomes corrupted or hallucinated settings prevent normal usage.

Subtitles & Logs:

- All AI speech output is simultaneously displayed as a subtitle overlay and logged to a chat history panel for accessibility and auditability.

User Experience & Lifecycle

First Contact (The Interview):

- **Optional Onboarding:** Upon first boot, the user can choose to undergo an "Interview" where the AI learns their preferences (games, workflow, privacy tolerance).
- **Skip & Learn:** Alternatively, the user can skip this. The AI will learn passively (building context from usage) or prompt for the interview later.

Update Channels:

- **Stable (Default):** Quarterly updates. Tested, reliable, "boring."
- **Bleeding Edge (Opt-In):** Rolling updates. Gets new AI features immediately but carries higher breakage risk.

Project Mode (Context-Aware Workspaces):

- The AI can initialize structured project environments (Design, Code, State files).
- **Isolation:** Changes made by the AI in "Project Mode" are scoped strictly to that project's file tree.
- **Collaboration:** Users can edit these files manually, and the AI respects the changes as "Hard Commits."

The Model Zoo (Pluggable Brains):

- **Open Architecture:** Users are not locked into Gemma/Moondream.
- **Drag-and-Drop:** Users can drop a .gguf file into the /models folder, and select it in the Control Room settings to swap out the "Manager" or "Sidekick."