Get started

WelcomeIntroductionQuickstartInstall helpOnboardingChangelogTroubleshooting

The app

UI overviewEditor and terminalARIA and agentsAI agentsGrind modeCapability packsWalletsSolana developmentRobinhood ChainLaunchReceiptsAgent economyGit integrationKeyboard shortcuts

Integrations

MCP bridgeDaemon SeekerStreamlockClawPumpDegenToolsJuiceVoight observabilitySpawn agentsSAID protocol

Trust

Security modelPolicies and approvalsArchitectureToken policyOpen source

More

Daemon ProRoadmapBrand kit

▶ DocumentationIntroduction

Introduction

What is DAEMON, why it exists, and what makes it different.

What is DAEMON?#

DAEMON is **where agents do Solana**: the execution venue that sits between an AI agent and your keys. Agents - ARIA built in, or your own from Cursor or Claude Code over the bridge - request on-chain actions: wallets, launches, swaps, deploys. DAEMON checks each request against **your**

Archived copy · cryp2nova.com

policy, signs approved actions from a vault the agent can't read, and leaves a **signed receipt** for everything. Your agent can ask. It can't sign.

Around that boundary, DAEMON is a full workspace: a purpose-built Electron application with its own Monaco editor, real terminals, state management, and database layer, not a VS Code fork, not a plugin. Keep your editor for code; graduate here when agents touch money.

Why DAEMON?#

AI agents are initiating on-chain actions through tools that can read your private keys. The default Solana keypair is a plaintext file; every editor extension, agent skill, and npm postinstall script you run can read it, and the incident record keeps compounding. See the incident tracker.

DAEMON closes that gap and consolidates the workflow around it:

- **Policy-gated execution** - spend caps, allowlists, and cluster rules run on every request; approval cards with simulation diffs for anything outside policy
- **An OS-encrypted vault** - keys never serialize across the agent boundary, not to the agent, not to us
- **Signed receipts** - a verifiable record of every action an agent takes
- **Agent Economy** - profiles, spend policy, paid resource checks, and receipts for agents that buy APIs, compute, data, or other agent work
- **Bring your own agent** - one MCP install connects Cursor, Claude Code, or VS Code to the same boundary
- **ARIA, the built-in operator** - runs the whole workspace from natural language
- **Agent swarms** - parallel sandboxed lanes in isolated git worktrees, push disallowed, keys stripped
- **The full cockpit** - Monaco editor, real terminals, wallet state, token launches, swaps, and deploys in one venue

The contrast#

The comparison that matters isn't editor versus editor: it's the same agent with and without a boundary.

Your agent today	Your agent behind DAEMON
Reads id.json like any other file	Keys sealed in an OS-encrypted vault
One malicious skill from a drained wallet	Unknown programs blocked by policy
Signs anything, instantly, silently	Simulation diff before any mainnet write
No record of what it did	A signed receipt for every action

Open Source#

Archived copy · cryp2nova.com

DAEMON is free and open source under the MIT License.

- **GitHub:** github.com/nullxnothing/daemon
- **Twitter:** @DaemonTerminal
- **Telegram:** t.me/daemonide
- **Built by:** nullxnothing

PreviousWelcomeNextQuickstart

© 2026 Daemon

ProductPricingDocsSecurityRoadmapTeamToken
GitHubXDiscordTelegramDEX Screener
PrivacyTermsToken policy

CRYP2NOVA

Archived copy · cryp2nova.com