

John Zachary Fitch

Agent tooling | systems performance | privacy-first infrastructure

I build production-grade tooling for agents and the substrate they depend on: deterministic retrieval, verifiable edits, structured tool APIs, and execution environments you can reason about. I work across Rust, Python, and web platforms with an evidence-first style: measure, reproduce, fix, and ship.

Recent Highlight (Jan 2026)

OPENAI CODEX

Ghost in the Codex Machine

Investigated and helped fix an "invisible" release-only regression where a pre-main constructor stripped `LD_*` / `DYLD_*` environment variables, breaking CUDA/MKL library discovery inside tool subprocesses. Shipped upstream with release-notes credit.

PROOF

 [Issue #8945 - Fix PR #8951](#)

 [Shipped + credited in rust-v0.80.0](#)

Representative impact

- MKL/BLAS (repro harness): ~2.71s -> ~0.239s (11.3x)
- CUDA workflows: restored expected library discovery to avoid CPU fallback in affected setups

Release notes excerpt:

"Special thanks to @johnzfitch for the detailed investigation and write-up in #8945."

Core Skills

Rust

Performance-critical systems, CLI tools, data structures, correctness-oriented engineering

Python

Tooling, analysis pipelines, automation, reproducible experiments

Web

WebGPU/WASM applications, client-side ML inference, offline-first UX

Systems

Linux, NixOS, DNS, TLS automation, containerized services, security hardening

Agent Integration

MCP servers, skill/plugin packaging, tool-driven workflows

Selected Projects (Public)

- [llmx](#) (Rust/WASM) - local-first codebase indexing + semantic chunk exports for agents
- [codex-xtreme](#) (Rust) - optimized, patched Codex builds (includes [codex-patcher](#))
- [burn-plugin](#) - Claude Code plugin + skills for the Burn deep learning framework
- [pyghidra-lite](#) - token-efficient MCP server for tool-driven program analysis
- [claude-cowork-linux](#) - run the official Claude Desktop app on Linux with sandboxing
- [Observatory](#) (WebGPU) - client-side AI image detection (live)
- [SpecHO v2](#) (Python) - 161D linguistic fingerprinting for AI text detection
- [definitelynot.ai](#) (PHP/JS) - Unicode-security-aware sanitizer + API
- [Iconics](#) (Python) - semantic icon library for professional docs (8k+ icons)

Infrastructure (Self-Hosted)

I operate production infrastructure on bare metal with:

- Declarative NixOS configuration (reproducible, atomic upgrades, rollbacks)
- Authoritative DNS and automated wildcard certificates (DNS-01 / RFC2136)
- Post-quantum security layers (hybrid SSH KEX, WireGuard + Rosenpass)

Education

UC Berkeley - Mathematics

What I'm Looking For

Roles building agent runtimes and developer tools, retrieval systems, and security/privacy foundations. I work best on teams that value measurable results, clear ownership, and high engineering standards.