RESUME · AI TOOLING

John Kingola @Astralchemist

AI tooling engineer · Models as extensions to coding tools · MCP servers and CLI design

> I check that something is true before anyone relies on it: trading strategies, AI agents, and the systems around them.

Nairobi, Kenya (UTC+3) | remote as a contractor or through an EOR | no visa sponsorship needed

I build the layer between AI models and the tools developers already use. Rig plugs into coding agents such as Claude Code, Cursor, and Codex over MCP and gives them a semantic map of the codebase instead of grep; it is at v0.3.0 on npm and launched on Show HN. StickyInc finds whichever agent CLI is installed (claude, codex, gemini, or Ollama) without asking for a key, and turns a tool call into a checkbox on screen. Cernos is a terminal tool with its own command and TUI that runs an LLM loop on OpenRouter to translate trading strategies from eight languages, then checks every translation against the original code. On the Change Intelligence Platform I fine-tuned Llama 3.1 8B and served it with vLLM on a Modal L4 GPU, behind an OpenAI-compatible API with an OpenRouter fallback.

github.com/Astralchemist LinkedIn kingolajohn55@gmail.com kingola.dev

v0.3.0	3	8	0
Rig on npm, launched on Show HN	coding agents Rig wires into	languages Cernos translates and verifies	API keys StickyInc asks for

NOW Rig is at v0.3.0 on npm. On retainer for the Change Intelligence Platform, including its self-hosted Llama 3.1 8B on vLLM.

// why hire me: decisions, and how i checked them

What does a coding agent need from a codebase that grep can't give it?

- THOUGHT Grep returns strings. To change code safely an agent needs callers, callees, and what a change will break.
- TESTED Rig indexes a repo in six stages into a typed graph and exposes search, callers and callees, impact and hubs as MCP tools, wired into Claude Code, Cursor and Codex.
- RESULT On npm at v0.3.0 and launched on Show HN. Local-first: no key, nothing uploaded.
- PROOF npm: rig-constellation

Can you trust an LLM's translation of trading code?

- THOUGHT An LLM can turn a strategy into clean JSON that still trades differently from the original.
- TESTED Cernos runs the original code in a locked-down sandbox and requires its signals to match the translation before anything is backtested. The sandbox runs Python originals today.
- RESULT A translation that drifts is rejected, not accepted as mostly right.
- PROOF Private repo, described

// ai tools i build with

Claude Code · Anthropic | Cursor · Anysphere | Codex · OpenAI

// skills

AI and GPU inference

vLLM on CUDA GPUs (Modal L4) | PyTorch, Hugging Face Transformers, Accelerate | KV-cache and GPU memory tuning

Llama 3.1 8B fine-tuning and serving | Claude, OpenRouter, Ollama fallback | Tool-calling loops, embeddings, and RAG

Tooling and desktop

Rust (Tauri, research engines) | Kotlin (Compose Desktop) | MCP servers | npm publishing | SQLite | Linux and VPS ops

Data and backend

Python | FastAPI | PostgreSQL and pgvector | TimescaleDB | Redis | RabbitMQ

Web and product

TypeScript | React | Next.js | Tailwind CSS | Node.js | Supabase

Delivery

Docker Compose | GitHub Actions CI | Cross-platform credential stores | Release engineering | Latency and build-performance passes | Upstream PRs to quant libraries

Trading and research

Backtesting engines | Signal extraction | L2 order books, markouts, and adverse selection | Power laws and EVT | MQL5, MT5, and Pine Script | QuantConnect Lean

// experience

Dec 2025 to Present
Dubai, UAE · remote

The Change Intelligence Platform · Lead engineer (contract), then maintainer

- Multi-tenant change-management SaaS: Next.js and FastAPI on a Docker Compose stack with Supabase. Fine-tuned Llama 3.1 8B and served it with vLLM on a Modal L4 GPU, behind an OpenAI-compatible API with an OpenRouter fallback.
- Built and shipped the platform solo in six months, Dec 2025 to Jun 2026, from architecture to production, under the direction of change-management practitioners. On retainer since for maintenance and feature work.
- Four role-based tiers, including a delegated Champion portal with magic-link auth, PDF export via Puppeteer, a seven-tag sentiment model, and an automated LinkedIn outreach pipeline.
- Led a latency and build-performance pass after an audit surfaced I/O-wait bottlenecks and N+1 query patterns.

Next.js | FastAPI | Python | Supabase | PostgreSQL | Docker Compose | Llama 3.1 8B | vLLM | Modal | OpenRouter | Puppeteer

Oct 2025 to Present
Dubai, UAE · remote

Early-stage stealth startup · Technical advisor, algorithmic systems

- Joined as lead algorithmic engineer at the founding stage, then moved to an advisory role while funding closes.
- Designed the backtest-as-a-service API: FastAPI, TimescaleDB, a Redis cache keyed on the strategy hash, and per-key rate limits.

Python | FastAPI | TimescaleDB | Redis

2024 to 2025
Nairobi · hybrid

ConsultHasha · Principal systems engineer

- Early engineer at a startup. Designed and built the REST API layer and the production service endpoints the product and its client integrations ran on.
- Owned architecture, deployment, and reliability.

2020 to 2025
Remote

Upwork · Freelance quant and full-stack developer

- MQL5 and MT5 Expert Advisors built to client risk specifications, then Python backtesting tools, Telegram signal bots, and execution dashboards.
- Also Docker-first Next.js and Postgres client platforms.

MQL5 | MetaTrader 5 | Python | Telegram bots | Next.js | PostgreSQL | Docker

// education

Sep 2022 — **Mount Kenya University** · Computer Science

// projects

Rig — Semantic codebase graph over MCP

● npm v0.3.0 · Show HN

npx rig-constellation gives a coding agent a map of the codebase instead of grep. Local-first, no key, nothing uploaded.

~100 commits, 4 days · 6 index stages · 3 agents wired

npm: rig-constellation · Source

TypeScript · MCP · Node · SQLite

StickyInc — Commitment pane fed by MCP

● v0.5.1 · v0.6 in review

Tell an LLM to call the dentist Friday, an MCP tool call fires, and a checkbox appears on screen. No backend, ever.

0 backends · 4 agent CLIs detected · every tag released

Releases

TypeScript · Rust · Tauri · MCP

Cernos — Strategy research engine in Rust, LEAN indicators ported

● Pre-release · private repo

A terminal tool that finds trading strategies on the web, translates them into one JSON format with an LLM, checks each translation against the original code in a sandbox, backtests and breeds them, and exports the winner as standalone Python or C++.

241 indicator types ported from LEAN · 129 testing methods · 8 source languages translated

Private repo, architecture only

Rust · OpenRouter LLM loop · Sandboxed containers · Python and C++ export

Numbers as of 2026-11-03. No live P&L is claimed.