

John Kingola @Astralchemist

Quant engineer · Backtesting and trading-verification infrastructure · Algo engineer at the founding stage

Nairobi, Kenya · open to remote, hybrid, or on-site

I build the infrastructure that decides whether a trading strategy is real: backtesting engines, signal-verification systems, and the risk controls around them, such as pre-registered kill conditions, stop-loss discipline scored per channel, cache keys that survive strategy edits, and rate-limited APIs that hold up under load. I have done this since 2020, from MQL5 Expert Advisors built to client risk specifications to a backtest-as-a-service API with per-key rate limits and a cache keyed on the strategy hash. Currently technical advisor on algorithmic systems for a trading startup, and lead engineer behind the Change Intelligence Platform, which I built end to end and now maintain.

[github.com/Astralchemist](#)

[kingolajohn55@gmail.com](#)

[LinkedIn](#)

[kingola.dev](#)

- NOW

Four fixes merged into ffn and credited in its v1.3.0 release notes, one into skfolio v1.4.1; quantstats and Databento fixed what I reported. PRs are open at Goldman Sachs, Microsoft Qlib, barter-rs, and Alpaca.
- NEXT

A 90-day pre-registered paper-trading track, kill conditions committed on day 0.

EXPERIENCE

Dec 2025 to Present
Dubai, UAE · remote

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

- 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.
- 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.
- 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.

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.

SKILLS

- 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
- Data and backend
- Python, FastAPI, PostgreSQL and pgvector, TimescaleDB, Redis, RabbitMQ
- Web and product
- TypeScript, React, Next.js, Tailwind CSS, Node.js, Supabase
- Tooling and desktop
- Rust (Tauri, research engines), Kotlin (Compose Desktop), MCP servers, npm publishing, SQLite, Linux and VPS ops
- Delivery
- Docker Compose, GitHub Actions CI, Cross-platform credential stores, Release engineering, Latency and build-performance passes, Upstream PRs to quant libraries
- 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

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.

EDUCATION

Sep 2022 Mount Kenya University · Computer Science

PROJECTS

Signly Forex signal verification
Live

Verify Telegram trading signal channels before trusting them.

5 iterations · 11 mo to product · 5 what-if exits

Hard: Parsing free-text signals reliably, and defining a verdict that penalises stop-loss indiscipline instead of rewarding win rate.

[signlyanalytics.com](#)

Python · FastAPI · Polygon · PostgreSQL

Rig Semantic codebase graph over MCP
npm v0.1.5 · 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

Hard: About 100 commits in four days to v0.1.5, stable per-project ports, and keeping the map live during indexing.

[npm: rig-constellation](#) · Source

TypeScript · MCP · Node · SQLite

CreditWatch Cloud-credit runway from the menu bar
1.0.0 released 2026-09-21

Answers one question: how long does the prepaid GPU credit last at the current burn.

10% runway buffer · 72 h history kept · 1.0.0 not 0.1.0

Hard: The account schema documented balance. A live account proved the spendable figure is credit. Windows and Linux were built and unit tested but not claimed, because no human had run the installed binary.

[GitHub release 1.0.0 · CHANGELOG and ROADMAP](#)

Kotlin · Compose Desktop · SQLite · Keychain

HOW IT GREW

2020	Client Expert Advisors, Upwork. The client's risk rules are the real specification. The entry logic is only what they think they are paying for.
2024 Q4	MQL5 Expert Advisor. The risk rules were the product. The entries were not.
2025 Q1	Quant-Algos catalogue. A catalogue rots unless every backtest runs on a schedule. One broken clone URL costs more credibility than a missing strategy.
2025 Q1	Expert-Advisor v0.5.1 research suite. Research that is not a strategy must say so in its own README. It does.
2025 Q2	telegram_bot. The signals were the unreliable part, not the execution.
2025 Q3	fx-bot and fx-dash. Nobody wanted to run a bot. They wanted to know whether to trust the channel first.
2025 Q3	Backtest API. Cache on the strategy hash, not the request. The same strategy is asked for a hundred ways.
2025 Q4	TGX-bot. A verdict that rewards win rate is wrong. It has to penalise stop-loss indiscipline.
2026 Q1	Signly, live. Shipping is not maintaining. A live product with no number attached is a question a reviewer will ask.
2026 Q1	Kalshi orderflow study (data lost). Data written to a temporary directory is not data. The capture died at 3.2 hours. The conclusion survived; the evidence did not.
2026 Q2	Rig v0.1.5. One burst of commits reads as abandoned, however good the burst was.
2026 Q2	StickyInc. Zero-key auto-detection beats a settings screen. Detect claude, codex, gemini, or Ollama and ask nothing.
2026 Q3	CreditWatch 1.0.0. Trust the live system over the schema. Scope the promise, not the code. Alerts must not fire on the past.

Liquidity Engine BTC order-book research, pre-registered
Public · negative result

A deterministic Rust engine that records BTC perpetual order books on three venues, replays them exactly, and tested a market-making strategy that does not work at retail fees.

47-67% variance from OFI · 41-66 ticks drift at 1 s · 3 venues, exact replay

Hard: Measuring what a fill costs. Probes resting at the touch saw 41 to 66 ticks of adverse drift within one second on every venue, in every session, under every fill model.

[Evidence page](#) · [Study, eleven figures](#) · [Source](#)

[Rust](#) · [WebSocket feeds](#) · [L2 queue model](#) · [Python figures](#)

Trading Strategy Library Public quant work
Open source

A strategy catalogue plus research tools, from indicator to execution to statistics.

4 strategy families · 5% max risk per trade · 1:1 breakeven move

Hard: Keeping research honest. The research suite is labelled research-only in its own README, and that label stays.

[Quant-Algos](#) · [Expert-Advisor v0.5.1](#) · [MT5 Expert Advisor](#)

[Python](#) · [MQL5](#) · [NumPy](#) · [QuantConnect Lean](#)

Backtest API Trading infrastructure
Private, described architecturally

Backtest as a service: the boring parts, built so the same strategy is never computed twice.

5×3 pairs × timeframes · 60/min rate limit · hash cache key

Hard: Cache keys that survive strategy edits. The same strategy is asked for a hundred ways, so the key is the strategy hash, not the request.

Private repo, architecture only

[Python](#) · [FastAPI](#) · [TimescaleDB](#) · [Redis](#) · [Docker](#)

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

Hard: Detecting whichever agent CLI is installed without asking for a key or a config file.

[Releases](#)

[TypeScript](#) · [Rust](#) · [Tauri](#) · [MCP](#)

Also: Zoetrope (Rust, deterministic HTML-to-video, pre-alpha); Cernos (strategy research in Rust, gauntlet tests and verdicts against the market's drift, pre-alpha); Trading-System-CI (QuantConnect Lean in Docker with an Optuna rebate optimizer); Meridian, Ledgerly (client and SaaS platforms, Docker-first Next.js and Postgres).

UPSTREAM FIXES TO QUANT AND TRADING LIBRARIES

Bugs found by reference-value tests and benchmarks, reproduced with failing tests, and reported with a fix.

2026 Q4

Liquidity engine, pre-registered test.
The strategy failed: gated and ungated round trips lose the same, about 46 USDT per BTC. The fill study, which came late, would have said so first.

2026 Q4

Pre-registered paper-trading track (next). Not yet. A negative result with intact methodology passes. A positive result with hand-edited logs fails.

SHIPPED AND PLANNED

2026-10-27	This site, published at kingola.dev
2026-10-03	ffn v1.3.0 credits four fixes
2026-10-02	Liquidity engine published, with a negative result
2026-09-28	Bugs reported to Databento, Goldman Sachs, and Alpaca
2026-09-26	Metric bugs reported upstream
2026-09-22	This site, first build
2026-09-22	Portfolio spec and readiness plan
2026-09-21	CreditWatch 1.0.0
2026-05-26	Rig v0.1.5
target	Verified paper-trading track · Running by 2026-10-06, 90 days
target	Kalshi Orderflow, re-run · 2026-10-13
target	Rig 0.2 · 2026-10-27
target	CreditWatch 1.1.0 · 2026-11-10
target	Signly, maintained and measured · 2026-11-17
target	Backtest engine, public · 2026-12-01

<u>ffn</u>	The ulcer performance index was not annualised, rounding residue counted as tracking error, short lookbacks missed month-end anchors, and 3y, 5y and 10y returns were reported without enough history. Four PRs merged, credited by name in the v1.3.0 release notes.
<u>skfolio</u>	Drawdown measures ignored a loss on the first observation, so they disagreed with the optimizer. PR merged the same day, listed in the v1.4.1 release notes.
<u>quantstats</u>	The probabilistic Sharpe ratio used the wrong kurtosis and could return a probability above 1. All three bugs confirmed by the maintainer, fixed in v0.0.82.
<u>Databento dbn</u>	The symbol map stored one entry per day per mapping, 1.73 GB at 20,000 instruments, and the test suite failed without a feature flag. Both fixed by the maintainers, released in 0.71.0 and 0.72.0.
<u>gs-quant</u>	Goldman Sachs' toolkit: a correlation crash, Sortino computed over losing days only, Sharpe on two clocks, an RSI seed, and backtest P&L that counts the starting capital. Six issues and five fix PRs open; the first was reproduced independently by another contributor.
<u>Qlib</u>	Microsoft's risk_analysis max drawdown ignored a loss in the first period, the same bug as skfolio's. PR open.
<u>barter-rs</u>	The tear sheet showed the loss rate as the win rate, and got profit factor and Sortino wrong. PR open.
<u>Alpaca MCP</u>	Generated trading tools carried no read-only or destructive hints, so a client could not tell a quote from an order. PR open.

FINDINGS FROM LIVE VERIFICATION

Scope the promise, not the code

CreditWatch 1.0.0 was built and unit tested on Windows and Linux. It claims macOS only, because no human had run the installed binary on the other two. The code shipped. The claim waited.

Trust the live system over the schema

The provider's account schema documented a balance field. A live account proved the spendable figure is credit. The monitor now reads credit, and the finding is quoted in the CHANGELOG rather than hidden in a commit.

Alerts must not fire on the past

A 6-hour alert armed when runway is already 4 hours fires immediately and teaches the user to ignore it. CreditWatch refuses to arm any threshold below the current runway, and says so in the dialog.