Marc Kendal

Builder & Technologist — Python · AI-Assisted Engineering · Real-Time Systems London, UK
marckendal@gmail.com · +44 7939 130 443 marckendal.com · github.com/MKToronto

---

Summary

Python engineer shipping AI-assisted expert tooling. Architected and directed two Claude Code plugins — one **published on the Claude Code marketplace + Codex CLI** (`python-clean-architecture` : 12 commands + skill pack for FastAPI clean architecture; installable by any Claude Code user), one **private in personal research** (`quant-finance` : 17 skills + a dedicated quant-expert subagent operationalising Marcos López de Prado's three quant-ML books). Claude-assisted authorship with architecture, curation, and validation owned directly — a distinctive modern skill worth naming honestly. Works spec-first with AI coding agents day-in, day-out — plans before prompts, reviews every change before commit. Full-time since May 2025 on a solo-built **algorithmic trading platform built to optimise personal savings returns** (~200 Python / asyncio modules, live across four broker APIs). Before that, five years owning the Python image-processing layer inside Fortress Technology's flagship ICON X-ray inspection machine, deployed at scale across Fortress's installed base. **Every piece of code shipped at Fortress went out with zero post-ship bugs and no customer complaints.**

Experience

Founding Engineer — Automated Trading Platform + Claude Code Plugins (independent R&D) · May 2025 – Present

Sole architect and engineer of a production-grade algorithmic trading platform in Python / asyncio, built to optimise personal savings returns — ~200 modules covering execution across four broker APIs, risk management, strategy iteration — plus two Claude Code plugins built alongside it. Live against real capital; ML-ops layer filtering signals before they become orders under active development. *(Following paternity leave, Sep 2024 – April 2025.)*

- **Shipped the published `python-clean-architecture` Claude Code plugin** on the Claude Code marketplace, dual-distributed to Codex CLI. Installable in any developer's workflow via `/plugin marketplace add` . 12 commands (`scaffold-api` , `refactor-legacy` , `extract-god-class` , `decouple` , `make-pythonic` , `review-architecture` , `add-endpoint` , `scaffold-tests` , `check-quality` , `diagnose-smells` , `suggest-patterns` , `review-api-design`) + skill pack covering 7 design principles with refactoring recipes, 22 code-quality rules, 25 Pythonic patterns, plus a working FastAPI hotel-booking example project. Honest attribution to Arjan Codes for the underlying teaching.
- **Built and maintain the private `quant-finance` Claude Code plugin** used inside the trading research — 17 skills + 5 commands + a dedicated `quant-expert` subagent operationalising López de Prado's trilogy. Skills cover triple-barrier labelling, fractional differentiation (with ADF stationarity testing + weight-formula derivation), covariance denoising (Marcenko-Pastur bounds), causal-factor validation (three Monte Carlo experiments: confounder / collider / mediation), sample weighting, microstructural features (Kyle/Amihud/Hasbrouck lambdas, PIN/VPIN), seven-sins backtest validator (CPCV + embargo + deflated Sharpe + PBO).
- **Honest authorship framing** on both plugins: architecture, skill-boundary curation, and validation against source material are owned directly; Claude wrote most of the prose under iterative direction.

This is the **agentic-engineering skill** in practice — designing, shipping, and distributing installable tooling that other engineers install — named honestly rather than as first-principles authorship.
- Owned **execution** end-to-end on the trading platform: four-broker Protocol abstraction (Interactive Brokers, IG, OANDA, Alpha Vantage, mock venue), order lifecycle management, position reconciliation against broker reality (retry-with-backoff → query original order's fill state → corrective `PositionAdjustment`).
- Built the async event-driven core — candle aggregator with atomic-pointer-swap ring buffer (CPython GIL), event bus with mixed sync/async dispatch, timezone-aware overnight guard, shadow / paper-trade mode, adaptive rate limiting against broker API pressure.
- Iterated strategies through five generations against real trade logs using registry-pattern signal composition and per-strategy balance tracking.
- Built a separate FastAPI monitor sub-app: live PnL / Sharpe / drawdown dashboards, WebSocket updates, Pushover critical alerts, runtime tuning endpoints exposing `tracemalloc` memory diagnostics.

Research Engineer — Fortress Technology (Toronto) · Oct 2019 – Aug 2024

Owned the Python computer-vision and data-ingestion layers inside Fortress's industrial-inspection product line. Flagship ICON X-ray inspection machine deployed at scale across Fortress's installed base. **Zero post-ship bugs and no customer complaints across shipped work.**

- Owned the real-time image-processing pipeline on Raspberry Pi edge hardware, alongside legacy C / OpenGL / Perl: C detector code writing into SysV shared memory; Python reading via `sysv_ipc` with byte offsets computed programmatically from a numpy dtype mirror of the C struct (`extract_constants.py` parses the C header); numba-JIT hot paths (`@jit(nopython=True, parallel=True, fastmath=True)` with `prange`) including adaptive-geometry rendering with sqrt-scaled brush radius and recursive hole-filling; atomic `hardlink_to` → `.replace()` file publishing; FastAPI / WebSocket layer with 9 asyncio locks over disjoint state regions; ThreadPoolExecutor / asyncio bridge bringing SHM polling into the async server. *(product demo)*
- Made the strategic call to invest in labelled-data acquisition at R&D-leadership level, then designed and built `image_dev_tool_fastapi` — FastAPI + Svelte + Redis-queue data-ingestion tool pulling time-windowed image batches from fielded machines over SSH, giving operators a UI to categorise contaminants.
- Built the SvelteKit operator UI running on every shipped ICON X-ray (`xray_streaming`) — kiosk-mode Chromium served via `sirv` on Pi, with a full offline-deployable installation pipeline. *(Fortress ICON product page)*
- Founding engineer on the Knapp pill-dispensing product — Flask control system shipped through external-customer handoff on a first Fortress project with zero post-ship bugs.
- Owned the multilingual HMI translations pipeline (`Translations_v3`) — bidirectional JSON ↔ Excel tool synchronising 40+ languages.
- Mentored junior engineers on the React + Python auth stack through pair programming and code review; 150+ bugs closed together with measurable skill growth.

Software Engineer — British Airways (London, UK) · 2014 – 2016

- Built live-data interception code for ba.com in Python (aspect-oriented programming / AOP) — captured real production data streams so tests ran against realistic live inputs instead of synthetic stubs.
- Built Selenium / Cucumber / Concordion BDD suites for desktop and mobile releases.
- Co-authored a "date-rolling" Python service that kept stubbed test data valid as calendar dates advanced — adopted across all teams.

Selected Projects

- `python-clean-architecture` — Published Claude Code marketplace plugin, dual-distributed to Codex CLI. 12 commands + skill pack (50+ reference files, 25 Pythonic pattern references) + FastAPI hotel-booking example. Primary artefact of the agentic-engineering skill.
- `quant-finance` **Claude Code plugin** — Private; 17 skills + 5 commands + dedicated `quant-expert` subagent operationalising López de Prado's trilogy into installable research tooling.
- **Automated trading platform** — ~200 Python / asyncio modules; four-broker execution abstraction; live.
- `localytics` — Published self-hosted alternative to GitHub Insights with **function-level cyclomatic complexity** (via Radon), weekly / monthly / yearly activity heatmaps, and per-file churn. Two FastAPI apps — a local scanner + a Render-hosted dashboard (FastAPI + Redis) — exchanging only aggregated numbers; source code never leaves the machine, a deliberate counter to hosted platforms that pull private source into training sets. Built to track the trading platform's development; MIT-licensed. Live demo.
- `docugit` — Published CLI extracting detailed git code-changes over configurable time-ranges for R&D tax-credit report generation via LLMs. Another example of AI-tooling-meets-business-problem.
- **Shiurim transcription pipeline** — Whisper + SpeechBrain speaker identification + Claude Code CLI classification over a large Dropbox audio corpus; ChromaDB-backed natural-language query plugin (`triebitz-halacha`).

Tech

Primary: Python · C (ctypes + shared memory) · JavaScript / Svelte / SvelteKit **AI / Agent tooling: Claude Code plugin architecture · Anthropic SDK · Codex CLI · spec-driven, agent-orchestrated workflow · prompt + skill + subagent design · ChromaDB · Whisper · SpeechBrain Systems / Infra:** FastAPI · asyncio · numba · Docker · SQLite · Redis / RQ · WebSockets · systemd · mamba / conda · Raspberry Pi · paramiko / SCP · memray profiling · SysV shared memory **Scientific Python:** NumPy · SciPy · Pandas · scikit-learn · FastAI · PyTorch · OpenCV **Working practice:** Agile · CI/CD · code review · unit testing · refactoring · spec-driven, agent-orchestrated development · review every change before commit · Claude Code + Codex CLI day-to-day

---

Education

MSc Data Science · City, University of London · 2016 – 2019 Dissertation (Computer Vision): *Improving the Optical Character Recognition of Middle Eastern Languages* — PyTorch CNNs.

PGDip Philosophy · The Heiden Institute

BSc (Hons) Economics, 2:1 · University of Nottingham · 2005 – 2008
