

Thomas Wells

thomas.j.wells@proton.me | linkedin.com/in/thomas-j-wells | github.com/eelixir | London Based

Education

University of Surrey

2025 - 2028

BSc (Hons) Computer Science: Predicted Upper Second-Class Honours

- Activities: Boxing Society (Treasurer), Computer Science Society

Skills

Languages: Python, Java, C++ 17, SQL

Platforms & Data: Azure, Linux, PostgreSQL, SQLite, REST APIs

Tools: Git/Github/GitLab, CMake, Google Test

Work Experience

Cyris

March 2026 - June 2026

Software Engineer Intern

US, Remote

- Partner directly with the CTO and founding team to deliver the technical roadmap, strengthen core infrastructure, and prepare an automated trading execution platform for public launch.
- Optimised Azure CI/CD pipelines by removing redundant remote builds and introducing artifact bundling, **reducing average deployment time by 45%** and accelerating developer iteration.
- Led the end-to-end redesign of the mobile and desktop interfaces, improving usability and simplifying access to advanced trading analytics and automated execution features.

Projects

Mercury – C++ Trading Engine

github.com/eelixir/mercury

C++ 17, CMake, Google Benchmark

- Engineered a high-performance limit order book and matching engine from scratch, achieving a sustained throughput of **3.2M+ orders/second** and an average insertion latency of **~320ns**.
- Implemented custom data structures designed to keep order lookup, insertion, and cancellation consistently fast, helping the engine handle large volumes of orders efficiently.
- Developed a robust ecosystem featuring pre-trade risk management, FIFO P&L tracking, and a backtesting framework capable of simulating diverse market conditions, enabling the validation of strategies against historical data.

LLaMA 3 CoT Enforcer Fine-Tune Model

github.com/eelixir/llama3-cot-enforcer

Python, Jupyter Notebooks, Google Colab

- Engineered a low-latency inference proxy to strictly enforce **Chain-of-Thought (CoT) reasoning protocols** on local LLaMA 3 deployments, ensuring structured, verifiable, and deterministic logic paths.
- Implemented custom logistic processors and token-level constrained decoding mechanisms, guaranteeing structural adherence to complex JSON schemas and **eliminating format-related hallucinations**.
- Designed a robust parsing and extraction pipeline to separate intermediate reasoning traces from final outputs, allowing the enforced logic to be seamlessly integrated into broader applications.

Onyx Terminal – Prediction Markets Trading Terminal

onyxterminal.xyz

Python (FastAPI), TypeScript (React 18), PostgreSQL (TimescaleDB)

- Designed a high-performance trading terminal (selected for the Polymarket Builders Program) integrating directly with the CLOB (Central Limit Order Book) via async WebSockets for low-latency market data aggregation.
- Developed algorithmic signal generation systems to detect "Smart Money" flow, implementing a statistical classifier that filters market participants based on trailing volume and historical profitability to automate whale wallet tracking.
- Built a full-depth Limit Order Book aggregator that compiles raw tick data into a consolidated DOM Ladder, providing granular visibility into liquidity walls, bid-ask spreads, and immediate market microstructure.