

DRONE BASSAN

Toronto, ON • 647-202-2900 • dronebassan@gmail.com • github.com/dronebassan

I am a Computer Science and Economics student at McGill who builds trading systems end to end. I designed and shipped a low-latency algorithmic trading app to the Google Play Store that streams live market data over WebSockets, runs strategies and execution on a server I built, and settles trades through the Hyperliquid perpetuals API. I also wrote a systematic perps bot with real risk controls and validated it in paper trading. I care about the parts of a system where latency, correctness, and risk actually cost money.

EXPERIENCE

Software Engineer — Talis Trade (Low-Latency Algorithmic Trading App)

May 2026 – Present

iOS + Android • Toronto, ON

- Built and launched a self-custodial algorithmic trading app to the **Google Play Store**, taking Android from nothing to full feature parity with the existing iOS app. Both clients compile from a single **Kotlin Multiplatform** shared module, so strategy models, order types, and market-data parsing live in one place and cannot drift between platforms.
- Streamed live market data over **WebSockets** straight into the client, so prices, order book updates, and fills land on screen as they happen instead of on a polling interval. Built the reconnect and resubscribe path so a dropped socket does not leave a user staring at a stale price while a position is open.
- Wired the clients to the backend through **Ktor**, which handles every network request the app makes: authentication, strategy configuration, order submission, and portfolio state.
- Ran strategies, execution, and the AI chat layer on an **AWS** server rather than on the phone, so strategy logic keeps evaluating and orders keep firing whether or not the app is open, and so the execution path sits close to the exchange instead of behind a mobile connection.
- Settled every trade through the **Hyperliquid API**, where the funds actually live. Orders are signed with EIP-712 against Privy embedded wallets, so the app is fully self-custodial and never holds user keys.
- Caught and fixed a money-critical trading-risk defect during a pre-deployment review, before any real capital was exposed to it.

Stack: Kotlin, Kotlin Multiplatform, Jetpack Compose, Ktor, WebSockets, AWS, Hyperliquid API, Privy embedded wallets, EIP-712 signing.

PROJECTS

Systematic Perpetuals Trading Bot on Hyperliquid (Hummingbot)

Dec 2025 – Jan 2026

- Built a systematic perps strategy from scratch: it ranks BTC/ETH/SOL by liquidity, takes direction from EMA momentum, and filters out bad regimes using funding rate, realized volatility, and trend strength before it will take a trade.
- Implemented the risk stack a real desk would expect: position sizing scaled to account risk and stop distance, stop-loss and take-profit, time-based and signal-flip exits, and a daily-loss kill-switch. Internal state only advances on confirmed fills, so a rejected or partial order can never desync the bot from the exchange.
- Logged every trade to CSV with an auto-generated markdown report, and validated the whole system through paper trading before running it against live markets.

Stack: Python, Hummingbot, Hyperliquid API, pandas.

Intelligent PDF Parsing System

2025

- Built a Python service that extracts clean text from any PDF, including scanned and low-quality ones, using a **three-stage fallback**: pdfplumber first, Tesseract OCR when that fails, and Gemini 2.5 Flash for the layouts neither can handle. Useful for turning filings and reports into structured, queryable data.

Stack: Python, pdfplumber, Tesseract OCR, Google Gemini 2.5 Flash.

SKILLS

Quant / Trading: Systematic strategy design, perpetual futures, EMA momentum & trend filters, funding and volatility regime filters, position sizing, stop/target and kill-switch risk logic, backtesting & paper trading, execution and fill handling

Languages: Python, Kotlin, JavaScript, SQL basics, HTML/CSS

Systems: WebSocket market-data streaming, REST/HTTP clients (Ktor), AWS server deployment, Kotlin Multiplatform, Jetpack Compose, EIP-712 signing, Git/GitHub

Libraries / Tools: Hummingbot, Hyperliquid API, pandas, Claude Code

EDUCATION

McGill University

Montreal, QC • Expected 2029

B.A. & Sc., Double Major in Computer Science and Economics

Bill Crothers Secondary School

Toronto, ON

Ontario Secondary School Diploma • 96% overall average