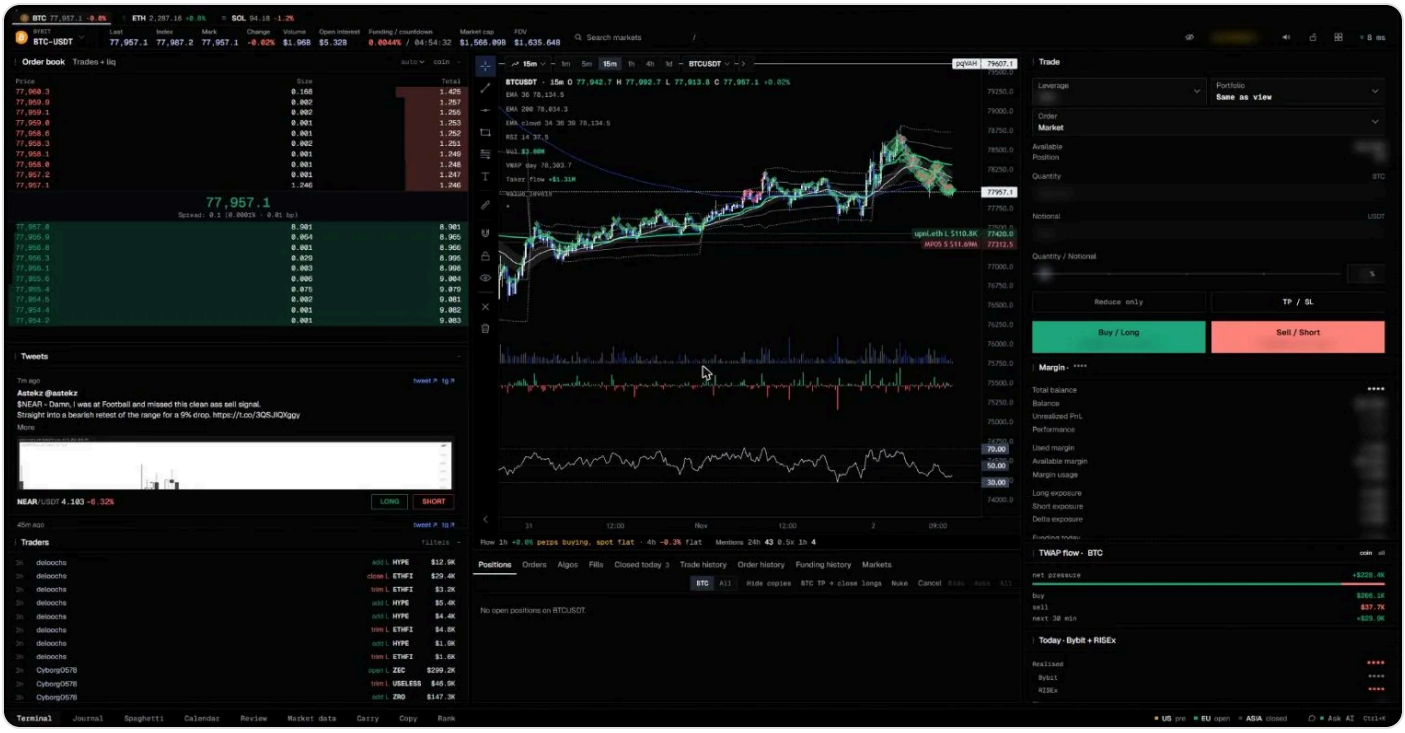
PERSONAL BUILD · REAL TIME DATA · RUNS LOCALLY

Real Time Market Data Terminal

A one screen trading cockpit that merges live feeds from six exchange venues, news and social, with a read only AI analyst on top.

Built with Claude Code · Python, WebSockets, JavaScript, SQLite, Claude API

~30k	6	20	10
LINES PYTHON AND JS	EXCHANGE VENUES LIVE	TOOLS THE AI CAN CALL	VIEWS



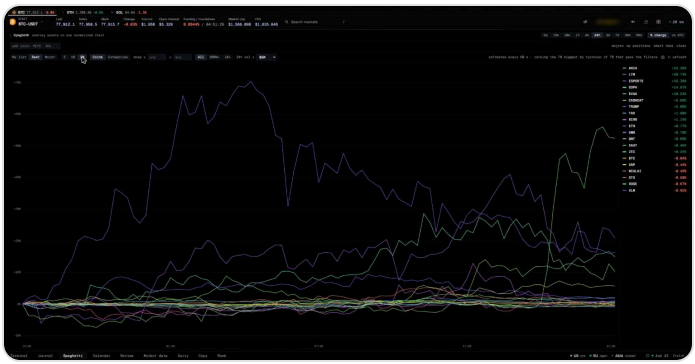
Main screen with private mode on (balances hidden)

Python · WebSockets · Server sent events · JavaScript · SQLite · Claude API · Real time data · Claude Code

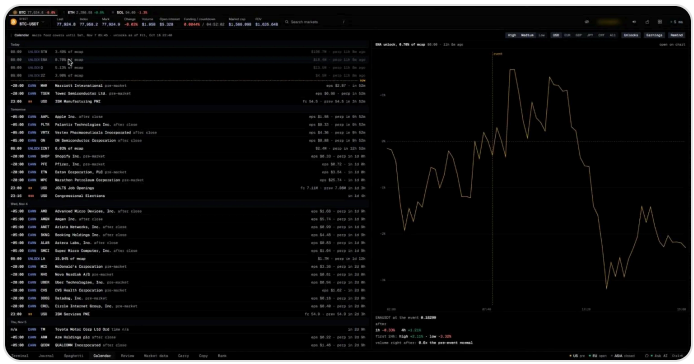
01 Overview

A local cockpit for crypto perpetual futures. One Python process holds the exchange connections and keys and streams account state to a single page browser UI. Around the chart and order ticket it gathers liquidations, order flow, large trader positioning, news, social posts and a calendar, plus journaling, review tools and an AI analyst. The design goal: fast, safe and few clicks over feature count.

02 Ten views



Spaghetti: best 20 coins on one chart



Calendar with the price reaction to an event

Terminal
The main screen on a drag and resize grid layout with saved, lockable presets.

Journal
The trade journal, embedded and prompting for entry and exit reasons.

Spaghetti
Many assets on one normalized percent chart, from 5 minutes to 90 days, with a versus BTC option and best and worst lists.

Calendar
Macro releases with impact filters, earnings and token unlocks, synced to the charts with reminders.

Review
What you did right and wrong, studied from your own trades by day, week, month or all time.

Market data
Twelve cards including a liquidation map, taker flow, TWAP flow, price and open interest, funding, mentions and liquidation levels.

Carry
Every coin listed on two venues, ranked by what a delta neutral funding hedge earns, with a calculator.

Copy
A window onto a remote copy trading bot, plus a screener ranking traders by how copyable they are.

Rank
A game style ladder where every closed trade is a game, scored in risk multiples with no look ahead.

Ask AI
The analyst chat described below.

03 Panels on the main screen

Order book and trades
Depth with grouping plus a 60 second volume and delta strip.

Liquidation tape
Streaming liquidations from three venues, filtered by size, venue and side.

Traders
Live fills and positions of tracked large wallets and public smart money accounts.

Tweets and news
Live feeds with urgency, coin and confirmed or rumor filters.

Chart
Draggable take profit and stop lines, EMAs, VWAP, volume, taker buy minus sell bars and liquidation bubbles.

Trade ticket
Market, limit, scale (2 to 20 order ladders), chase, TWAP and hedge, sized by quantity or notional, with margin and liquidation estimates.

Margin and exposure
Balance, used and available margin, long, short and delta exposure, funding today and leverage.

Bottom tabs
Positions, orders, algos, fills, closed today, trade, order and funding history, with a close all action behind a confirm step.

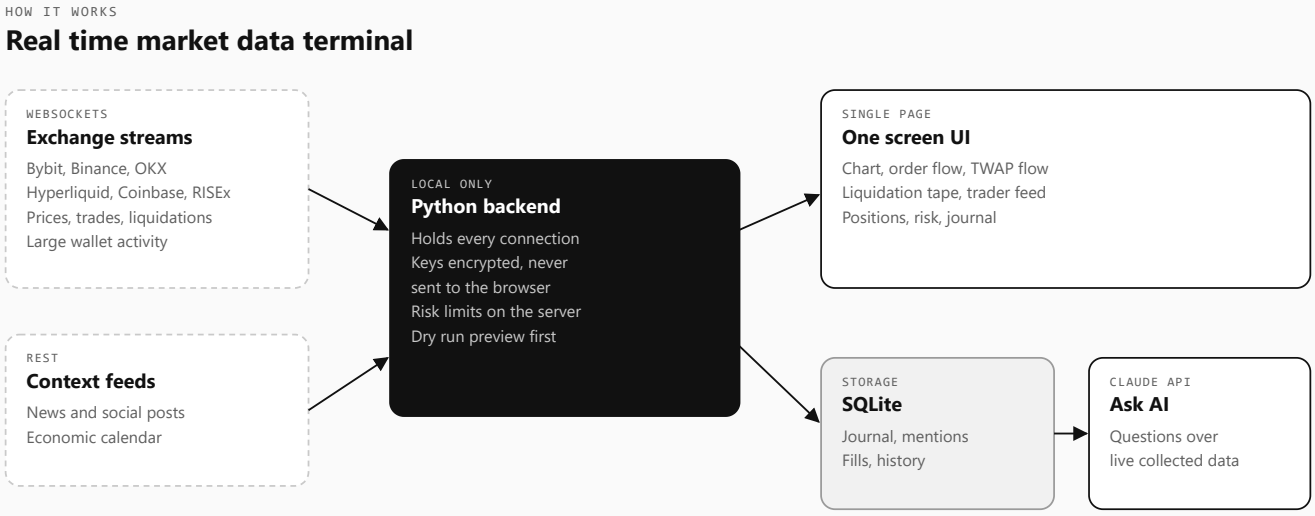
Keyboard first
13 rebindable hotkeys, a market search palette, and a private mode that hides names and sizes for screen sharing.

Health bar
Feed age and socket round trip time always visible, with sound alerts.

04 Ask AI: a read only analyst

- **Right model for the job:** Claude Haiku tags news and trades, Claude Sonnet answers questions and writes reviews and risk reads, Claude Opus writes the weekly review.
- **20 tools** over the terminal's own data: positions, exposure scenarios, news search, liquidations, large trader moves, outlier finder, market overview, narratives, calendar, trade history, journal stats and memory.
- **Numbers are computed in code,** and a post check flags any figure in an answer that did not come from a tool result.
- **Read only by design:** the AI can never place, change or cancel an order.
- **Cost controlled:** prompt caching, retries, a monthly hard cap and per feature caps, with every call logged.
- **Channels** for book, risk, journal, market, plays, news and calendar, each with its own focus prompt and long term memory.
- Social posts are always treated as data, never as instructions.

05 How it works



- Exchange WebSockets for prices, books, trades, liquidations and private account events, plus signed REST calls.
- State is pushed to the page over server sent events, batched to at most about 40 updates a second without delaying a lone update.
- About 30 background workers: feeds, liquidation collectors, news and announcement loops, algo sweeps, journaling and housekeeping.
- Three SQLite databases hold AI call costs, clustered news, 14 days of liquidations, trades, reviews, social mentions and graded calls.
- News is cleaned, de duplicated by similarity and tagged once per story.

06 Safety by design

- API keys are encrypted with Windows DPAPI and never sent to the browser. The server checks key permissions at start and daily and warns if a key could withdraw.
- Risk rails on the server, not the UI: order size limits, stop distance limits, a far limit price guard, an optional exposure cap and a burst limit on new orders.
- Every order shows a dry run preview with expected loss at stop and profit at target before it is sent.
- Algos only send the next child order once the previous one is confirmed, stop on unknown states and sweep orphans after a restart.
- Paper mode blocks every live order route.
- Local only web security: loopback binding, host header checks against DNS rebinding, origin checks and a strict content security policy.

07 Tech stack

Backend	Python 3.13, mostly standard library, websocket client, Telethon, SQLite
Frontend	Vanilla JavaScript in one page, lightweight charts, gridstack layout
AI	Anthropic Messages API with tool use and prompt caching
How I built it	Claude Code, iterating in small versioned steps with a backup before every change

---

Charles Lorenz Almenanza · Land Data Analyst and Claude Code builder · lorenzalmenanza@gmail.com · charles-almenanza-portfolio.pages.dev
Personal project. No balances, results or keys are shown.