# AI Trading Agent – Plan (v2)

Date: 21/08/2025

## Overview

Refined plan based on feedback: stronger data integrity, explicit LLM→signal rules, more robust risk controls, switch to alpaca-py, and clearer ops/monitoring. This is the version we'll follow going forward.

## Environment & Dependencies

• Python 3.11, venv  
• requirements.txt with version pinning  
• Core: numpy, pandas, matplotlib, yfinance, backtrader, requests, python-dotenv  
• Trading/API: alpaca-py (not alpaca-trade-api), tiingo  
• Storage/Perf: duckdb (lightweight DB), optional: polars  
• Caching: requests-cache (optional)

## Data Sources & Integrity

• Market data: Polygon (intraday), Tiingo (EOD). Start EOD, add intraday later.  
• Fundamentals/filings: SEC EDGAR. Optionally Tiingo Fundamentals / IEX Cloud.  
• Integrity: adjust for corporate actions (splits/dividends). Align timestamps across sources.  
• Keys: .env (local-only). In production, use a secret manager.

## Research LLM (Cloud)

• Tasks: summarize earnings, classify sentiment, extract guidance/risks.  
• Output schema (validated JSON): {sentiment:[-1..1], guidance\_up:bool, risk\_tags:[...]}  
• LLM→Signal rule (initial): go long if sentiment ≥ 0.7 AND guidance\_up==True. Reduce/avoid entries on negative risk tags (e.g., 'liquidity', 'accounting').  
• Log every LLM call and result; human review optional pre‑trade.

## Backtesting

• Use vectorbt for wide parameter sweeps; Backtrader for event-driven realism (later).  
• Include transaction costs & slippage (bps). Guard against look‑ahead (shift signals by 1 bar).  
• Validation: walk‑forward + OOS; regime tests (2008/2020); Monte‑Carlo noise on returns.

## Risk Management

• Per-trade: Stop-Loss, Take-Profit. Add optional ATR-based stop (volatility aware).  
• Portfolio: max exposure per position (2–5%), sector caps, daily drawdown kill‑switch.  
• Sanity checks before orders; automatic circuit breaker on anomalies.

## Execution

• Paper trade via Alpaca (alpaca-py). Only after ≥3 months of paper success → IBKR live.  
• Implement order throttling/retries and broker rate-limit awareness.

## Monitoring & Logging

• Store trades/signals/metrics in DuckDB or SQLite. Separate error log channel.  
• Dashboards for PnL, drawdown, win‑rate, exposure. Alerts on exceptions/kill-switch.

## Checklist Before Live

✔ No future leakage; ✔ Slippage/fees modeled; ✔ Walk‑forward & OOS passed; ✔ Robustness (param sensitivity, noise tests); ✔ Paper ≥ 3 months with thresholds; ✔ Risk limits in code; ✔ Full audit trail.

## Roadmap

Immediate (Phase 1):  
 1) Pin requirements.txt, switch to alpaca-py.  
 2) Backtest: add fees/slippage + optional ATR stop + trade CSV (done).  
 3) Sanity checks & error logging channel.  
 4) Paper-trading glue (alpaca-py) with tiny sizing.  
  
Phase 2:  
 • LLM pipeline with validated JSON & signal rules; correlation testing to market reaction.  
 • Vectorbt sweep, regime tests, Monte-Carlo.  
 • Portfolio limits & dashboards.