INT-L

Centralizing growth.

v1.0

Data Sources

1. **Stock Market Data (Primary Response Variable) <** Polygon.IO Developer Plan
    a. Historical and 15-minute delayed OHLCV (daily and intraday)
    b. Index data: SPY, NASDAQ, QQQ, DIA, VIX
    c. Corporate Actions: Splits/Dividends
2. **Macroeconomic Indicators (Predictor)** < FRED API
    a. CPI (inflation)
    b. Fed Funds Rate
    c. GDP
    d. Unemployment Rate
    e. PMI/ISM indices
3. **Company Events (Predictor)**
    a. Polygon API: Earnings, Calendar, Financials
    b. SEC EDGAR API: 1-K, 10-Q, 8-K, Press Releases

ALL PREDICTORS SERVE TO MODEL THE PRICE OF SELECTIONS AT CERTAIN PERIODS. CERTAIN SELECTIONS WILL HAVE EXTENUATING WEIGHTS BASED ON WHAT PREDICTORS PLAY THE MOST PIVOTAL ROLES IN PRICING.

Database Schemas

prices_daily: *ticker, date, open, high, low, close, adj_close, volume*

prices_intraday: *ticker, datetime, open, high, low, close, volume*

macro_indicators: *indicator_name, date, value*

company_events: *ticker, event_type (earnings, filing, guidance), date, details*

benchmarks: *index_ticker, date, close, volume*

portfolio_allocations: *ticker, allocation_pct, sleeve (core/opportunity), effective_date*

SCHEMA IS SUBJECT TO CHANGE ON AN AS-NEEDED BASIS TO MEETS THE NEEDS OF InteL V1.0 SYSTEM REQUIREMENTS.

DATA STORAGE WILL BE PERFORMED IN AN *AMAZON WEB SERVICES (RDS POSTGRESS)* DATABASE, WITH CLEANING AND TRANSFORMATION PROCESSES TAKING PLACE IN PARTNERED *AMAZON WEB SERVICES LAMBDA* ENVIRONMENTS.

Processing Pipeline

Ingestion Layer:

- Nightly ETL job via orchestrated
    - Ingest new OHLCV from Polygon
    - Ingest latest macro releases from FRED
    - Pull company events/filings from Polygon + EDGAR

Feature Engineering:

- Price-derived: daily returns, rolling volatility, moving averages
- Macro-derived: inflation surprises (current CPI versus rolling mean)
- Event-derived: binary flag for earnings week, sentiment score from filings (text parse)

Storage:

- Write to Postgres (append-only for prices/events, upserts for macro series)

NOTE: IF LP ACCESS IS GRANTED, MORE SCHEMA DESIGN IS REQUIRED. THIS WILL BE DONE WITH EXTRA PRECAUTION TO PROTECT THE PRIVACY OF LP PERSONALLY IDENTIFIABLE INFORMATION (PII) AND INFORMATION RELATED TO FUND-INVOLVEMENT. IF TLWEF ERECTS A PLATFORM FOR LP ACCESS, ALL PERTINENT LPs WILL BE NOTIFIED IMMEDIATELY OF SUCH CHANGES TO THE PLATFORM AND WILL BE GIVEN THE CHOICE TO OPT-IN FREE-OF-CHARGE OR OPT OUT, WITH THE ABILITY TO OPT IN AT ANY TIME.

AWS-Specific Architecture

1. **Core Database Layer (AWS RDS Postgres)** db.t3.medium, 100GB storage $70
    a. **Central warehouse for cleaned and normalized data**
    b. **Tables for prices, intraday, macro, events, and benchmarks**
    c. **Scalable (vertical/horizontal), automated backups, snapshots**
    d. **Enforce schema integrity**
    e. **Indexing for saved runtime costs**
2. **Data Ingestion and Cleaning (AWS Lambda, serverless)** 128MB, avg 1s runtime $5
    a. **Each Lambda = one ingestion task (e.g. fetching prices, macros, etc)**
    b. **Trigger daily via CloudWatch Events (e.g. 7pm ET after market close)**
    c. **Python scripts inside Lambda:**
        i. **Call Polygon/FRED/EDGAR APIs**
        ii. **Clean and transform data (adjust splits, parse filings, align timestamps)**
        iii. **Write to staging tables in RDS**
    d. **AWS S3 (raw data storage)** t3.medium, 100GB storage $5
        i. **Store JSON/CSV dumps from APIs before cleaning**
        ii. **Provides audit trail and recovery if ingestion breaks**
3. **Processing Pipeline**
    a. **Staging Tables in RDS:**
        i. **Raw data loaded from Lambda**
        ii. **Transformation functions clean (adjust dividends, merge corporate actions)**
    b. **Clean Tables in RDS:**
        i. **Final canonical datasets (used for analytics and dashboards)**
4. **Analytics Layer**
    a. **AWS EC2 / SageMaker** $30
        i. **For heavier ML backtests or modeling**
        ii. **SageMaker optional for scalability and prebuilt ML infrastructure**
    b. **Direct Querying from Python (statsfjfmodels, sklearn) connected to RDS**
5. **Visualization**
    a. **Plotly Dash/Streamlit app hosted on EC2** (internal use).
    b. **Quarto Reports rendered and stored in S3 for client-facing access.**
    c. **S3 and CloudFront is an option for a private client portal**
6. **Security and Ops**
    a. **IAM roles for access (AM/admin versus other permissions)**
    b. **Encrypt RDS at rest, enforce SSL in transit**
    c. **CloudWatch monitoring for Lambda and RDS health**
    d. **Automated alerts if data ingestion fails.**

Data Requirements

Must ingest **daily OHLCV** for 50+ tickers (with ability to expand to 1,000).
Must ingest **intraday OHLCV (1m bars)** for capacity of up to 30 tickers.
Must ingest **macroeconomic indicators:** CPI, Fed Funds Rate, GDP, Unemployment, PMI.
Must ingest **company events:** earnings calendar, guidance, SEC filings.
All raw data must be archived to **S3.**
All cleaned data must be stored in **RDS Postgres.**

Processing Requirements

Daily ETL must complete **within 2 hours of market close.**
Intraday updates must be processed with **<1 minute lag.**
All corporate actions (splits, dividends) must be applied before storage in clean table.
Staging to clean transformations must be automated (Lambda and SQL procedures).

Analytics Requirements

Must calculate: **daily returns (**simple/log**), rolling volatility, and Sharpe ratio.**
Must support at least **one regression model that predicts return percentage.**
Must support at least **one classification model that predicts movement up/down.**
Must provide **attribution output**: macro, events, price momentum contribution.

Reporting Requirements

Internal dashboard must display:
Core and Opportunity performance
Forecast versus realized return (per ticker)
Macro and event overlays
Client-facing quarterly reports must include:
Account NAV versus S&P benchmark
Attribution breakdown (macro/events/momentum)
Performance versus 30% annualized growth target
Reports must be generated in **PDF/HTML** (Quarto) and archived in **S3.**

PLEASE SEE NEXT PAGE FOR NON-FUNCTIONAL REQUIREMENTS.

Non-Functional Requirements

- **SCALABILITY:** Support up to 1,00 tickets with <5% performance degradation.
- **RELIABILITY:** > 99% successful ingestion job completion per year.
- **LATENCY:**
    - Daily refresh: completed by 2 hours after market close.
    - Intraday refresh: < 1 minute lag
- **SECURITY:**
    - All data encrypted at rest (RDS, S3) and in transit (SSL/TLS)
    - IAM roles enforced for Account Manager and others with access
- **AUDITABILITY:** Raw dumps and processing logs must be retained for **24 months.**
- **COST EFFICIENCY:** AWS infrastructure must remain < $230 per month baseline.

Start Date: 20251003

Checkpoint 1 (20251004 5:00pm CST)

- Configure RDS and deploy schema
- Configure S3 buckets (raw dumps and reports)
- Set IAM roles and CloudWatch monitors

Checkpoint 2 (20251008 7:00pm CST)

- Build Lambdas for:
    - Polygon OHLCV (daily and intraday)
    - FRED macro indicators
    - SEC EDGARD events
- Write Raw to S3 and Clean to Staging

Checkpoint 3 (20251010 7:00pm CST)

- ETL transformations for Staging to Clean
- Derived metrics: daily returns, volume, Sharpe, event flags, macro surprises
- Validation scripts (split/dividend adjustments, nulls)

Checkpoint 4 (20251012 9:00pm CST)

- Bulk pull 10-year historical OHLCV from Polygon (daily, intraday)
- Bulk load macro history (FRED) and company filings archive (SEC)
- Transform, clean, insert into RDS clean tables
- QC time alignment and event tagging

Checkpoint 5 (20251015 9:00pm CST)

- Implement regression (returns) and classification (up/down)
- Run backtests on historical data to validate ingestion
- Attribution framework (macro, events, momentum factors)

Checkpoint 6 (20251016 9:00pm CST)

- Stand up dashboard in EC2, generate and store reports, test