**📦 Overview of Components**

1. **Data-Preprocessing Pipeline** (ETL jobs run offline, ahead of time)
2. **Screener App** (Streamlit or similar, reads only pre-computed data)

**🔄 1. Data-Preprocessing Pipeline**

This is **not** part of your live UI—it’s a standalone set of scripts (or orchestrated tasks) that you run on a schedule (e.g. nightly, weekly, or on demand) to build and maintain the database your screener will consume.

**A. Universe Definition (Step 1)**

* **Input:** Your two CSVs (NSE, BSE + ISIN→Ticker map)
* **Output:** qualified\_tickers\_list.csv (all tickers you might ever consider)
* **Frequency:** One-off or “whenever you update your CSVs” (e.g. quarterly)

**B. 5-Year Data Validation (Step 2)**

* **Input:** qualified\_tickers\_list.csv
* **Process:**
  1. In batches, fetch each ticker’s annual financials via yfinance
  2. Count distinct year-ends, keep only those ≥ 4 years
  3. Cache HTTP calls + throttle to avoid Yahoo limits
  4. Log any failures/shortages to a “quarantine” list
* **Output:** qualified\_4yr\_tickers.csv (tickers with true ≥4 years data)
* **Frequency:** Rerun whenever you want to re-validate (e.g. quarterly, after new filings)

**C. Metrics Ingestion & Percentile Precompute (Step 3 & 4)**

* **Input:** qualified\_4yr\_tickers.csv
* **Process:**
  1. Batch-fetch for each ticker:
     + Income statement (revenue, netIncome)
     + info['industry']
  2. Compute per-ticker metrics:
     + Sales Growth (CAGR, YoY, or however you define)
     + Net Profit Margin (netIncome / revenue)
  3. Group by industry → compute percentiles (1st–10th, or user-chosen)
  4. Persist to your data store:
     + Lightweight database (SQLite/Postgres) **or**
     + Partitioned Parquet/CSV files
* **Output:**
  1. **Metrics table**: ticker, industry, sales\_growth, npm, as\_of\_date
  2. **Cutoffs table**: industry × percentile → metric threshold
* **Frequency:** Nightly or weekly (depending on how fresh you want results)

**🎛 2. Screener App**

This is your **“thin client”**—e.g. a Streamlit or Dash app which:

1. **On startup** loads the latest metrics + cutoffs from your data store (no yfinance calls).
2. **Presents UI controls**:
   * Dropdown for industry (or “All”)
   * Slider for percentile choice (1–10) for Sales Growth and/or NPM
3. **Filters** the pre-computed metrics table by “metric ≥ cutoff” for selected percentile(s).
4. **Displays** resulting tickers (with their metrics, links, charts, etc.).

**Key:** the app itself does **zero** HTTP/API calls to Yahoo. Everything it needs is sitting in your database or files, ready to be queried.

**🕘 Scheduling & Orchestration**

* Use a simple cron **or** a lightweight orchestrator (e.g. **Airflow**, **Prefect**, **Dagster**, or even a scheduled GitHub Action) to chain:

text

CopyEdit

[quarterly] → Universe Definition

↓

[quarterly] → 4-Year Validation

↓

[nightly/weekly] → Metrics Ingestion & Percentile Precompute

* The Screener App can either be redeployed after each pipeline run or simply be coded to always read “latest” from your store.

**✅ Why This Works**

* **Separation of Concerns** keeps your UI fast and reliable.
* **Pre-computed data** means no surprises, no rate-limit errors in front of users.
* **Modular pipelines** let you tweak validation, ingestion, or cutoffs independently.
* **Scalable**—if you need more tickers or more metrics tomorrow, you just adjust the ETL, not the UI.