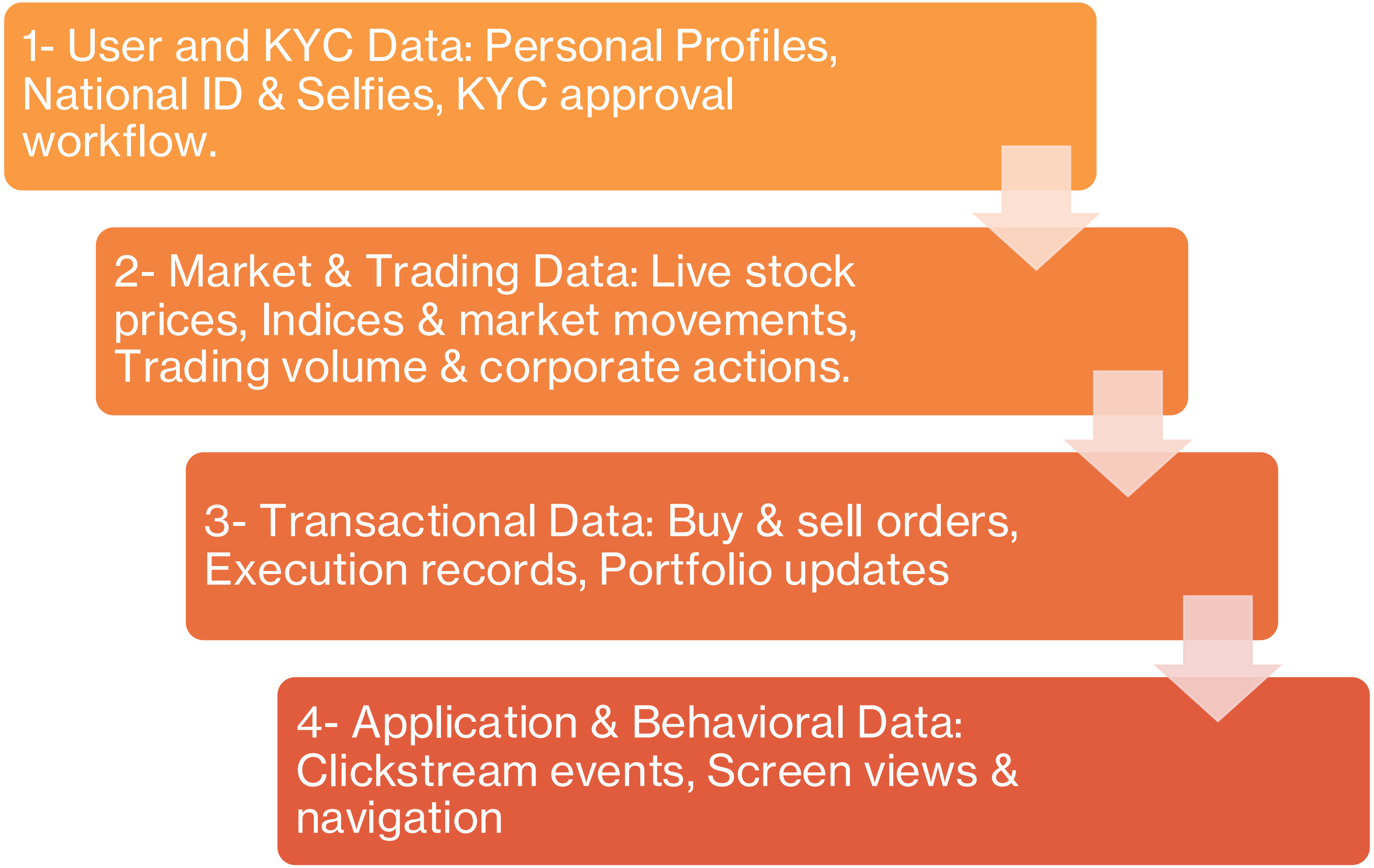
Thndr Data Engineering Use Case

Digital investment platform working on real-time trading environment, secure financial transactions, KYC verification, user activity tracking, and analytical workloads.

Data Sources



1- User and KYC Data: Personal Profiles, National ID & Selfies, KYC approval workflow.

2- Market & Trading Data: Live stock prices, Indices & market movements, Trading volume & corporate actions.

3- Transactional Data: Buy & sell orders, Execution records, Portfolio updates

4- Application & Behavioral Data: Clickstream events, Screen views & navigation

Storage Layer

- 1- Data Lake:
- types of data: like Market feed streams (JSON / streaming events)
- Purpose: a- Acts as the single source of truth.
- b-Enables scalable, low-cost storage.
- c-Supports machine learning workloads.
- possible technologies: AWS S3 · Azure Data Lake Gen2
- 2- Data Warehouse:
- Data stored: Normalized user profiles and Cleaned transactions & order history
- Purpose: a- High-performance SQL analytics
- b- Fast reporting for operations & finance
- possible tech: BigQuery & Azure Synapse.

Processing Layer

1- Real-Time Processing: used for: a- Live stock price ingestion

b- Real-time trade execution system

c- Real-time notifications

Techniques & Tools: Kafka · Spark Streaming · Flink · Redis caching

2- Batch Processing: used for: a- Daily account valuation

b- Transaction reconciliation

c- ETL data cleansing and normalization

Techniques & Tools: Apache Spark · SQL pipelines

Serving Layer

1- Business Intelligence Dashboards:

for internal teams:

a- Risk: fraud patterns, suspicious activity

B- Product team: feature usage, user funnels

C- Operations: daily trade volume, failure rates

D- Compliance: regulatory reporting

Tools: Power BI · Tableau

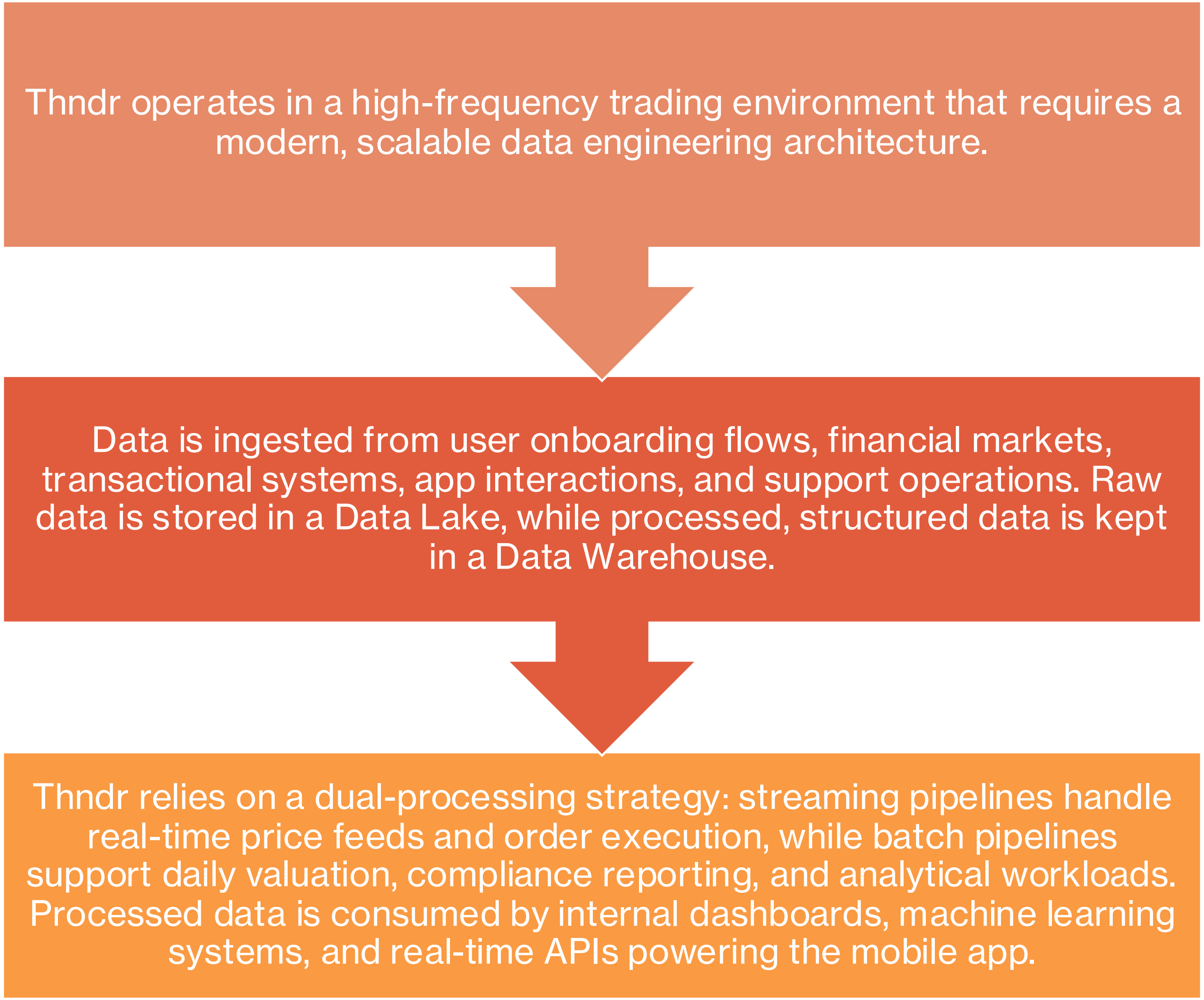
2- Machine Learning Models:

a- Fraud detection

B- User segmentation

C- Predictive user scoring

Executive Summary

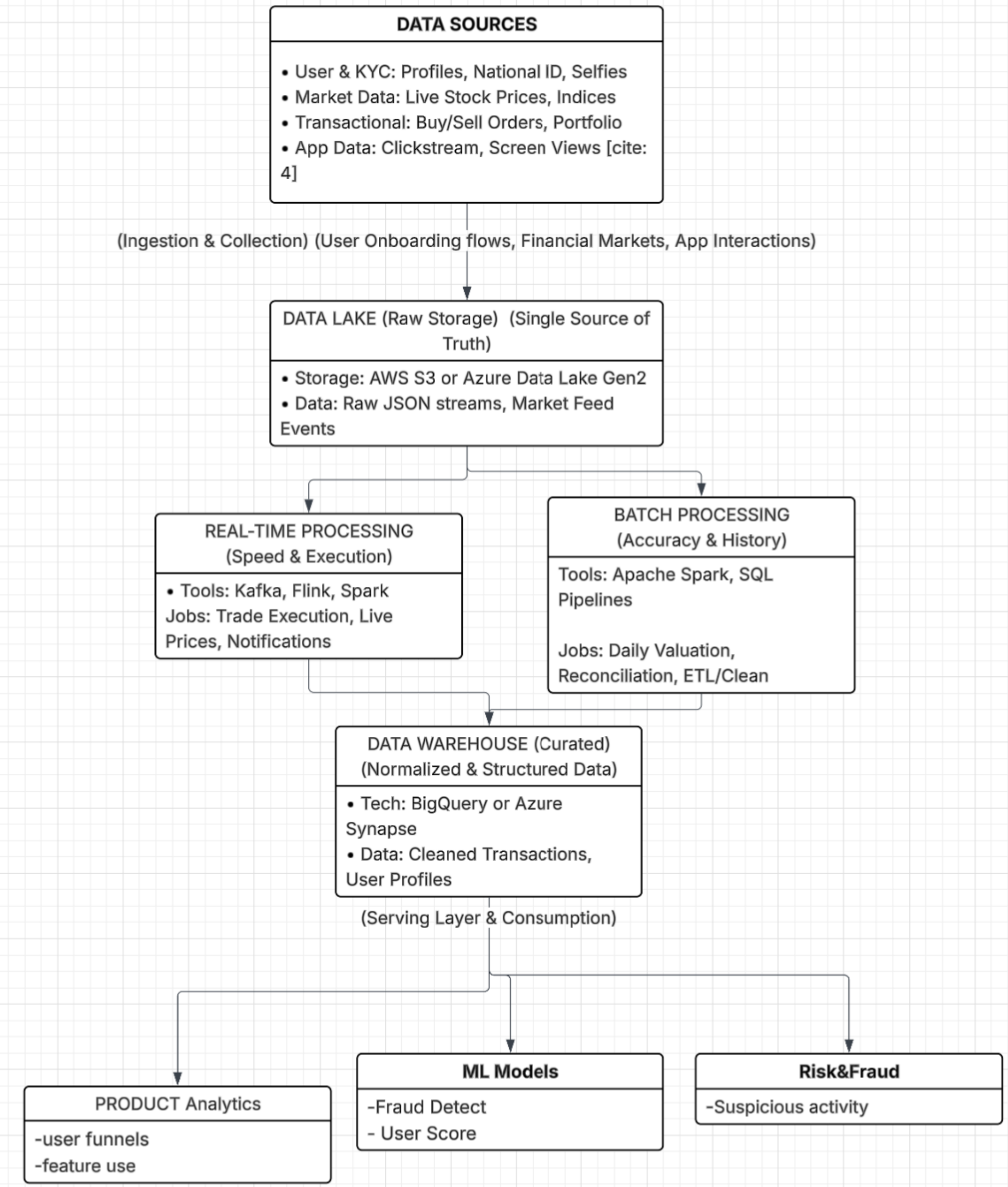


Thndr operates in a high-frequency trading environment that requires a modern, scalable data engineering architecture.

Data is ingested from user onboarding flows, financial markets, transactional systems, app interactions, and support operations. Raw data is stored in a Data Lake, while processed, structured data is kept in a Data Warehouse.

Thndr relies on a dual-processing strategy: streaming pipelines handle real-time price feeds and order execution, while batch pipelines support daily valuation, compliance reporting, and analytical workloads. Processed data is consumed by internal dashboards, machine learning systems, and real-time APIs powering the mobile app.

Pipeline Architecture Diagram



DATA SOURCES

- User & KYC: Profiles, National ID, Selfies
- Market Data: Live Stock Prices, Indices
- Transactional: Buy/Sell Orders, Portfolio
- App Data: Clickstream, Screen Views [cite: 4]

(Ingestion & Collection) (User Onboarding flows, Financial Markets, App Interactions)

DATA LAKE (Raw Storage) (Single Source of Truth)

- Storage: AWS S3 or Azure Data Lake Gen2
- Data: Raw JSON streams, Market Feed Events

REAL-TIME PROCESSING (Speed & Execution)

- Tools: Kafka, Flink, Spark Jobs: Trade Execution, Live Prices, Notifications

BATCH PROCESSING (Accuracy & History)

Tools: Apache Spark, SQL Pipelines

Jobs: Daily Valuation, Reconciliation, ETL/Clean

DATA WAREHOUSE (Curated) (Normalized & Structured Data)

- Tech: BigQuery or Azure Synapse
- Data: Cleaned Transactions, User Profiles

(Serving Layer & Consumption)

PRODUCT Analytics

-user funnels
-feature use

ML Models

-Fraud Detect
- User Score

Risk&Fraud

-Suspicious activity