FAWRY PAY

Data Engineering Case Study

Real-Time Fraud Detection & Financial Reconciliation

Presented By:
Mahmoud Ali AbdelMaksoud Salem
ID: 21139149

Group:
MNF4_AIS5_S1

1. Company Profile

About Fawry

Fawry is the leading digital transformation & e-payment platform in Egypt, processing **financial transactions** for millions of customers daily.

Core Operations:

- **Bill Payments:** Utilities, Telecom, Education.
- **Banking:** Wallets, ATM integration, B2B settlements.
- **Retail:** Merchant POS systems and supply chain.

Scale

4M+
Daily Transactions

Reach

300K+
POS Terminals

2. The Engineering Challenge

1. Speed (Latency)

Problem: Fraud must be stopped *before* the payment is approved. **Requirement:** Process transaction rules in $<$**500ms**.

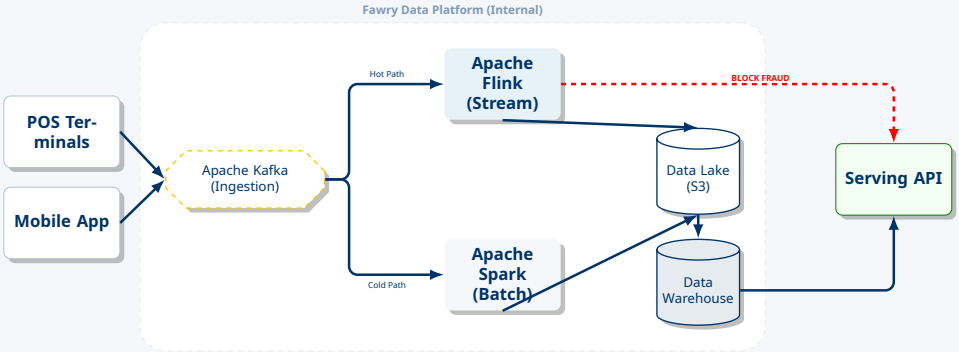
2. Accuracy (ACID)

Problem: Financial data cannot be "approximately" correct. **Requirement: Zero Data Loss** and exact ledger matching.

3. Scalability

Problem: Traffic spikes during "Salary Days" and holidays. **Requirement:** Auto-scaling pipeline to handle **10x load**.

3. Data Pipeline Architecture



4.1 Ingestion Layer (The Buffer)

Technology: Apache Kafka

Kafka acts as a high-throughput event bus.

- **Why Kafka?** To handle ”Backpressure”. If the database is slow, Kafka holds the data so no transaction is lost.
- **Topics:** Transactions are split by type (e.g., `BillPayment`, `WalletTransfer`).

Sample Transaction Event

```
{
"txn_id": "992381",
"merchant_id": "M_772",
"amount": 500.00,
"loc": "Cairo, EG",
"status": "INITIATED"
}
```

4.2 Processing: Hot Path (Real-Time)

Real-Time Security Layer

Technology: Apache Flink (Stateful Stream Processing)
This layer processes transactions *before* they are finalized.

- **State Management:** Flink keeps a "State" of the user's last 5 locations/transactions in memory.
- **Fraud Rule Example:**

$$\text{If } (\text{Location}_{\text{Current}} \neq \text{Location}_{\text{Last}}) \text{ AND } (\text{TimeDiff} < 5 \text{ mins}) \rightarrow \textbf{BLOCK}$$

- **Outcome:** The transaction is blocked instantly.

4.3 Processing: Cold Path (Batch)

Batch Reconciliation Layer

Technology: Apache Spark
This layer runs heavily during the night (Off-Peak Hours).

- **Aggregation:** Runs every night at 2:00 AM to process millions of records.
- **Reconciliation Logic:**

$$\sum \text{Money Collected} == \sum \text{Billers Owed}$$

- **Output:** Generates settlement files (CSV) for partners like Vodofone, Orange, and Electricity Co.

4.4 Storage Layer (Data Lakehouse)

1. Data Lake (S3 / HDFS)

- **Role:** Stores raw, immutable logs forever.
- **Format: Apache Parquet** (Columnar) for compression and fast scanning.
- **Use Case:** Audit trails (Regulatory Requirement) & retraining Fraud ML models.

2. Data Warehouse

- **Role:** Stores "Cleaned" and modeled data.
- **Schema:** Star Schema (Fact Tables for Transactions, Dimension Tables for Merchants).
- **Use Case:** Fast SQL queries by the finance team for monthly reports.

4.5 Serving Layer (Value Delivery)

How Data is Consumed

- **Operational APIs (REST):**
  - Used by the Mobile App to show ”Credit Limit”.
  - Used by Merchant App to show ”Today’s Earnings”.
- **BI Dashboards (Tableau / PowerBI):**
  - **CEO Dashboard:** Total Revenue Live ticker.
  - **Ops Dashboard:** Heatmap of failed transactions across Egypt regions.

5. Conclusion

Summary

Secure

End-to-end Encryption & Tokenization protects user data.

Fast

Real-time Fraud Checks (<500ms) ensure safety without delay.

Reliable

Decoupled architecture ensures Zero Data Loss during outages.

References: Fawry Investor Relations, Confluent Financial Services Architectures.