**Real-Time Fraud Detection System in Financial Transactions**

**Use Case: Real-Time Credit Card Fraud Detection**

**Objective:**  
A financial institution wants to monitor credit card transactions in real time to detect potential fraud based on high transaction amounts and unusual location patterns.  
Real-time dashboards and alerts are required to identify suspicious behavior and trigger investigations.

**Components & Their Roles**

| **Component** | **Purpose** |
| --- | --- |
| **Apache NiFi** | Simulates real-time credit card transactions, sends to Kafka |
| **Azure Event Hubs (Kafka)** | Message broker for ingesting streaming data |
| **Azure Databricks (PySpark)** | Processes Kafka stream and applies fraud rules |
| **Delta Lake (ADLS Gen2)** | Stores streaming data (raw and flagged) for historical audit and analysis |
| **Power BI** | Visualizes real-time fraud patterns and transaction activity |

**Workflow Steps**

**1. Apache NiFi → Event Hub (Kafka)**

**Simulates transactions with possible fraud scenarios:**

* High transaction amount > ₹75,000
* Foreign transactions (location outside India)

**NiFi Flow:**

GenerateFlowFile → UpdateAttribute → PutKafkaRecord\_2\_0

**Sample Transaction Payload:**

{

"transactionId": "TX8791",

"cardNumber": "9876-XXXX-XXXX-4321",

"amount": 88000,

"location": "London",

"timestamp": "2025-07-30T10:15:00Z",

"userId": "U101"

}

**2. Azure Event Hubs (Kafka-Compatible)**

* Namespace: fraud-detection-ns
* Topic: transactions
* Acts as a buffer between NiFi and Databricks

**3. Databricks PySpark (Streaming Pipeline)**

**Schema:**

schema = StructType([

StructField("transactionId", StringType()),

StructField("cardNumber", StringType()),

StructField("amount", DoubleType()),

StructField("location", StringType()),

StructField("timestamp", StringType()),

StructField("userId", StringType())

])

**Read from Kafka:**

df\_kafka = spark.readStream \

.format("kafka") \

.options(\*\*event\_hub\_config) \

.load()

df\_json = df\_kafka.selectExpr("CAST(value AS STRING)") \

.withColumn("data", from\_json(col("value"), schema)) \

.select("data.\*")

**Detect Fraudulent Transactions:**

from pyspark.sql.functions import when

df\_flagged = df\_json.withColumn("isFraud", when(

(col("amount") > 75000) | (col("location").isin("London", "NYC")), True

).otherwise(False))

**Write to Delta:**

df\_flagged.writeStream \

.format("delta") \

.outputMode("append") \

.option("checkpointLocation", "/mnt/checkpoints/fraud-check") \

.start("/mnt/delta/transactions\_processed")

**4. Power BI Dashboard (Real-Time Visualization)**

**Option: Use Azure Synapse or Databricks Connector**

* Connect Power BI to Delta Lake table via **Azure Synapse external table** or **Databricks SQL Endpoint**
* Visualize:
  + Fraud vs Non-Fraud Transactions
  + Top Locations by Amount
  + Transaction Heatmap
  + User-wise Transaction History

**Power BI Dashboard Pages:**

* **Overview Page**
  + Total Transactions
  + % Fraudulent Transactions
* **Fraud Analysis**
  + Bar chart: Fraud count by location
  + Map: High-risk geographies
* **User Monitoring**
  + Transactions by user
  + Time-series plot of high-value transactions

**Final Architecture Diagram (Real-Time Fraud Detection)**

