SOLUTION DOCUMENT

Real-Time Fraud Detection System

(Architecture + Data Flow + ML + Rules + UI + API + Components)

1. Overview

This solution implements a **real-time fraud detection platform** combining:

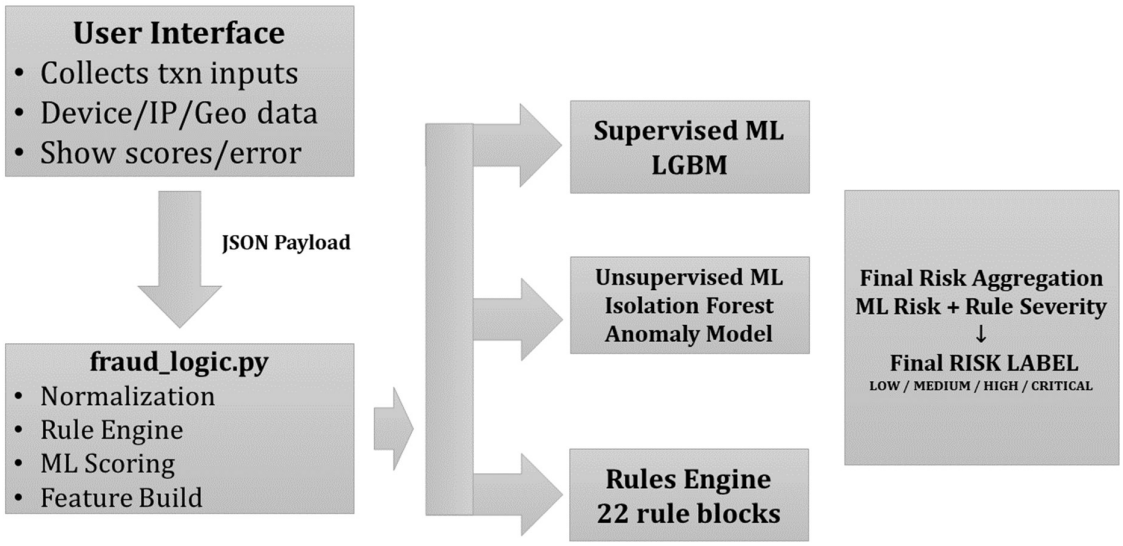
- **Supervised ML model** (LightGBM)
- **Unsupervised anomaly model** (Isolation Forest)
- **Deterministic rule engine**
- **Behavioral scoring**
- **Geo-risk scoring**
- **Device & velocity rules**
- **Time-of-day and pattern risk**
- **User interface (Streamlit)**
- **API interface (FastAPI)**

The platform assigns a **fraud risk score (0–100)** and a final label:

LOW / MEDIUM / HIGH / CRITICAL

The system is fully explainable, auditable, and suitable for production.

2. Architecture Diagram



3. Data Flow

1. **User enters transaction details**
2. Payload is built with:
    - Amount, location, currency
    - Device/browser
    - Time-of-day
    - Transaction city/country
    - Home city/country
    - Telemetry (velocity, averages)
3. **ip_country is automatically derived from txn_country**
4. ML model receives 8 features:
    - Amount
    - TransactionType
    - Location (txn_city)
    - DeviceID
    - Channel
    - hour
    - day_of_week
    - month
5. Isolation Forest produces anomaly magnitude
6. Rule engine evaluates:
    - 22 deterministic rules
    - Geo mismatch
    - Time of day
    - Velocity
    - Beneficiary risk
    - Card/ATM risk
7. ML + Rules fused into final severity.

4. Machine Learning Logic

4.1 Supervised Model (LightGBM)

Predicts probability of fraud:

- Output range: `0.0 → 0.02` generally
- Converted to **Fraud Risk Score (0–100)**

4.2 Unsupervised Model (Isolation Forest)

Measures unusualness of the transaction:

- Raw score inverted
- Normalized to **Anomaly Score (0–100)**

4.3 Risk Labeling

```
If fraud_prob >= 0.0173 → CRITICAL
If fraud_prob >= 0.00023 → HIGH
If fraud_prob >= 0.00005 → MEDIUM
Else LOW
```

4.4 Score Normalization

Fraud Score (0–100) = (fraud_prob / 0.02) * 100
Anomaly Score (0–100) = (anomaly_score / 0.10) * 100

Ensures business interpretability.

5. Rule Engine Overview (22 rules)

Categories:

- **Extreme Amount Rules**
- **Geo-Risk Rules**
- **Device Risk**
- **Impossible Travel**
- **Velocity Rules**
- **Card Behavior**
- **ATM behavior**
- **Beneficiary Risk**
- **Daily & Monthly Pattern Risk**
- **Time-of-Day Rules**

Rules produce severity levels:

- LOW
- MEDIUM
- HIGH
- CRITICAL

Highest severity becomes the rule score.

6. Final Risk

```
Final_Risk = MAX( ML_Risk_Label , Rules_Severity )
```

Example:

- ML = MEDIUM
- Rules = HIGH
  → **Final Risk = HIGH**

7. Response Time

Total end-to-end scoring time measured:

```
start_time = time.perf_counter()
end_time   = time.perf_counter()
response_time = end_time - start_time
```

Displayed to the user.

8. API Layer (FastAPI)

The API receives the same payload as UI.

Response includes:

- FraudProbabilityRaw
- AnomalyScoreRaw
- FraudRiskScore (0–100)
- AnomalyRiskScore (0–100)
- ML Risk Label
- Rule triggers
- Final Risk

9. Deliverables Provided

- Full source code
- Optimized app.py
- fraud_logic.py
- FastAPI app
- Normalized scoring
- Response time integration
- Documentation
- KT-ready explanation
- Example transactions dataset