

Case 2

Data and Core Signals

The solution is centered on analyzing transactional status distribution over time, using minute-level aggregation as the primary monitoring granularity. Incoming transactions are classified into four mutually exclusive statuses: approved, denied, failed, and reversed.

This approach enables both volume-based and rate-based analysis while preserving sufficient temporal resolution for near real-time detection.

Architecture Overview

The system uses PostgreSQL as the source of truth, Grafana for visualization and alerting, and a Python FastAPI service for transaction ingestion and alert persistence.

High-level flow:

1. Transactions are sent to a REST ingestion endpoint.
2. Raw data is stored and aggregated in PostgreSQL.
3. SQL views compute rates, baselines, and anomaly indicators.
4. Grafana dashboards visualize volumes and rates in real time.
5. Grafana alert rules continuously evaluate anomaly conditions.
6. When alerts fire, Grafana sends a webhook to a FastAPI endpoint, which persists alert events for auditing and historical analysis.

API Endpoints

At minimum, the system exposes a transaction ingestion endpoint:

- POST `http://localhost:3002/ingest/transaction`
Receives transaction events and persists them in PostgreSQL, aggregating counts per minute by status and optional authorization code.

In addition, a Grafana webhook receiver was implemented:

- POST `http://localhost:3001/grafana/webhook`
Receives alert notifications from Grafana when anomaly conditions are met and persists alert metadata into an alerts table. This creates an auditable incident history and enables downstream integrations or post-incident analysis.

Database Modeling and SQL Layer

Transactional counts are stored in a transactions table, representing transaction volume per timestamp and status. For monitoring purposes, this data is aggregated into minute-level buckets through an intermediate SQL view (`v_tx_minute`), which computes:

- total transaction volume per minute
- counts per status (approved, denied, failed, reversed)
- corresponding rates (`denied_rate`, `failed_rate`, `reversed_rate`)

On top of these aggregates, a dedicated anomaly detection view (`v_tx_anomaly`) enriches each time bucket with baseline statistics derived from recent historical behavior and exposes explicit anomaly indicators. The view includes:

- per-minute totals (total, approved, denied, failed, reversed)
- per-minute rates (`denied_rate`, `failed_rate`, `reversed_rate`)
- rolling baseline mean and standard deviation for each rate, computed over the previous 30-minute window (excluding the current minute)
- boolean anomaly flags indicating when each metric exceeds its normal range:
 - `denied_above_normal`
 - `failed_above_normal`
 - `reversed_above_normal`

To complement status-based monitoring, transaction events may include an authorization code attribute. These values are stored and aggregated separately in the `transactions_auth_codes` table. A dedicated Grafana panel exposes this data, allowing operators to correlate detected anomalies with specific authorization codes and quickly identify issuer-side, integration, or downstream failures.

Anomaly Detection Method

The anomaly detection logic follows a baseline-based statistical approach using rolling windows. For each minute, the current denied, failed, and reversed rates are evaluated against a baseline computed from the preceding 30-minute window.

An anomaly is flagged for a given metric when all of the following conditions are met:

- sufficient transaction volume exists (total ≥ 30), preventing noise from low-sample fluctuations
- a valid baseline is available (non-null standard deviation)
- the current rate exceeds the baseline mean by more than three standard deviations ($\text{mean} + 3\sigma$)

This produces three independent anomaly signals:

- failed_above_normal
- denied_above_normal
- reversed_above_normal

Dashboards and Real-Time Visualization (Grafana)

Grafana dashboards provide real-time visibility into system health and transactional behavior. Key panels include:

- total transactions per minute
- failed, denied, and reversed rates over time
- alert history
- authorization code breakdown derived from transactions_auth_codes

Together, these panels allow operators to quickly identify deviations from normal behavior and to correlate incidents with time windows, traffic patterns, and specific authorization responses.