

RFID Anomaly Detection System Architecture

Author: Meeranath Sai Salendra | Version: 2.0

Edge Layer

Processing Layer

Cloud Layer

Impinj R700 RFID Reader

- Reads RFID tags from zones
- Publishes tag events via MQTT (EPC, Zone, Timestamp, RSSI)

MQTT

Python Processing Service

MQTT Subscriber

Receives tag events
Every ~100ms

Redis Cache (Redisearch)

- Last read timestamp
- Zone history per tag
- Tag status (normal/missing)
- 30-second TTL tracking
- Fast lookup & updates
O(1) Access

Anomaly Detector

Rule 1: Missing Detection
No read in 30s = Missing

Rule 2: Zone Hopping
≥3 zones = Anomaly

read
update
check

Cloud API

Cloud Database (BigQuery/PostgreSQL)

rfid_events	anomalies	tag_registry
<ul style="list-style-type: none">• tag_id• zone_id• timestamp	<ul style="list-style-type: none">• anomaly_type• detection_time• zone_sequence	<ul style="list-style-type: none">• registered_tags• expected_zones• metadata

React Dashboard

- Real-time anomaly alerts
- Missing tag notifications (Cloud Run / Firebase)

System Characteristics

Performance:

- MQTT: ~100ms latency
- Redis: <5ms cache operations
- Detection: Real-time processing

Scalability:

- Handles 1000s of tags
- Horizontal scaling ready

Reliability:

- Redis persistence enabled
- Batch writes to cloud DB

Detection Rules

1. Missing Tag Detection:

- If no RFID read for tag in 30 seconds
- Mark as MISSING in Redis
- Trigger alert & store anomaly

2. Zone Hopping Detection:

- Track zone history in Redis
- If tag reads from ≥3 different zones