**Integrated Open-Source On-Premises ISO 8583 Monitoring & Alerting Architecture (100–200k TPS)**

پیشنهاد اولیه معماری سیستم مانیتورینگ با در نظر گرفتن شرایط زیر:

1. حذف نقاط single points of failure (SPOFs)
2. پیشنهاد تکنولوژی های جایگزین

**1. Data Collection & Ingestion**

**Tools & Agents**

* **Apache Kafka**: Distributed broker for high-throughput (200k+ TPS) ingestion; use ≥4 brokers with RF=3 in KRaft mode (no ZooKeeper).
* **Filebeat**: Collect ISO 8583 logs from switches/terminals.
* **Telegraf**: System metrics (CPU, memory, disk I/O).

**Parsing**

* Use **jPOS** (Java) or **iso8583‑python** to decode messages and extract MTI, response codes, PAN, amount.

**SPOF Mitigation**

* Deploy Kafka in active-active across 4 nodes; dual NICs & redundant load-balancers (e.g., HAProxy pair).

**Alternatives**

* **Apache Pulsar**: Native tiered storage, geo‑replication.
* **Redis Streams**: Ultra-low latency ingestion.

**2. Stream Processing**

**Primary Engines**

* **Apache Flink**: Stateful real-time metrics and fraud detection; RocksDB state backend, HA JobManagers (2 nodes).
* **Kafka Streams**: Lightweight enrichment (lookup merchant details in PostgreSQL).

**Key Workflows**

* Compute TPS, success/error rates by MTI.
* Velocity-based fraud: >5 transactions/PAN in 60 s.
* Track end‑to‑end latency (request→response).

**SPOF Mitigation**

* Flink: HA JobManagers; task managers scale horizontally.

**Alternatives**

* **Spark Structured Streaming** (micro‑batches).
* **Materialize**: Incremental SQL views.
* **ksqlDB**: SQL on Kafka topics.

**3. Storage Layer**

**Time-Series Data**

* **VictoriaMetrics**: Prometheus-compatible TSDB (1M+ samples/sec).
* **Apache Parquet + HDFS**: Cost-efficient raw metrics archive.

**Logs & Traces**

* **Elasticsearch**: Parsed ISO 8583 logs (30 d hot, cold to MinIO).
* **MinIO**: S3-compatible cold storage (3-node erasure coding).
* **PostgreSQL**: Reconciliation & reference tables.

**SPOF Mitigation**

* Elasticsearch: ≥3 master-eligible, dedicated nodes, time-based indices.
* PostgreSQL: Patroni-managed Primary+Replica.

**Alternatives**

* **OpenSearch**: Apache-licensed ES fork.
* **Loki**: Index-only log metadata.
* **TimescaleDB** or **ClickHouse** for TSDB & analytics.

**4. Visualization & UI**

* **Grafana**: Dashboards for TPS, error rates, latency (VictoriaMetrics).
* **Kibana**: Log exploration (Elasticsearch).

**Alternatives**

* **Superset**, **Redash** for SQL-native dashboards.

**5. Alerting**

* **Prometheus Alertmanager**: Threshold alerts
* **ElastAlert**: Elasticsearch anomaly detection (PAN velocity spikes).
* **Grafana Alerts**: Dashboard-based notifications.

**Alternatives**

* **Cortex**: Multi-tenant alerting.
* **Kapacitor**: Stream-based alerting.

**6. Infrastructure & Networking**

| **Role** | **Count** | **Spec (Per Node)** | **Notes** |
| --- | --- | --- | --- |
| Kafka Brokers | 4 | 16 vCPU, 64 GB RAM, 2 TB NVMe | RF=3, KRaft |
| Flink Task Managers | 4 | 32 vCPU, 128 GB RAM, 1 TB NVMe | HA JMs (2 nodes) |
| Elasticsearch Cluster | 5 | 64 vCPU, 256 GB RAM, 4 TB NVMe | 3 masters + 2 data |
| VictoriaMetrics | 2 | 32 vCPU, 128 GB RAM, 10 TB SSD | 30 d retention |
| PostgreSQL (HA) | 2 | 16 vCPU, 64 GB RAM, 2 TB SSD | Patroni |
| MinIO Erasure Nodes | 3 | 16 vCPU, 64 GB RAM, 50 TB HDD | 11+4 erasure profile |
| Support Services | 4 | 8 vCPU, 32 GB RAM, 1 TB SSD | Grafana, Kibana, Keycloak |

**Networking**

* 10 Gbps NICs, LACP on dual NICs, redundant leaf/spine switches.

**7. Security & Compliance**

* **Data Masking**: Flink tokenizes PAN → SHA‑256; pgcrypto in PostgreSQL.
* **Access Control**: Keycloak (clustered) for Grafana/Kibana RBAC; Auditbeat logs access.
* **Encryption**: TLS for all inter-node; LUKS at rest on metrics/log nodes.

**8. Scalability & Reliability**

* **Kafka**: ≥6 partitions/topic; dynamic broker addition if consumer lag >100k.
* **Flink**: Horizontal task manager scale; monitor CPU >75%.
* **Elasticsearch**: Auto-index rollover; add data nodes if disk >80%.

**9. Cost Optimization**

* **MinIO** on retired HDD servers; ZFS compression on ES/MinIO; Kafka tiered storage.

**10. Disaster Recovery**

* **Backups**: Daily ES→MinIO snapshots; hourly VM→NFS.
* **DR Site**: Async Kafka MirrorMaker 2 to secondary DC.