NETCELL-EDA Architecture

Security architecture

NETCELL-EDA Architecture
Security architecture

Core principles

Least privilege: All services, users, and tenants get only the permissions they need.

Defense in depth: Network, identity, data, and application layers all enforce controls.

Tenant isolation: Strong logical isolation via topics, namespaces, and storage boundaries.

Auditability: Every sensitive action is logged and traceable.

NETCELL-EDA Architecture
Security architecture

Identity & access

Service identities:

Each microservice uses its own service account.

Access to Kafka, Document DB, Feature Store, and Model Registry is via scoped credentials.

User identities:

Admins, data scientists, and operators authenticate via SSO/IdP.

RBAC defines roles: tenant-admin, data-scientist, ops, viewer.

NETCELL-EDA Architecture
Security architecture

Network & perimeter

Ingress:

Event Gateway and Dashboard/API behind an API gateway or ingress controller.

TLS everywhere (external and internal).

Segmentation:

Separate namespaces per tenant.

NetworkPolicies restrict cross-namespace traffic.

Kafka, DB, Feature Store accessible only from whitelisted services.

NETCELL-EDA Architecture
Security architecture

Data security

At rest:

Encryption for Kafka logs, Document DB, Feature Store tables, and backups.

In transit:

TLS for all service-to-service and client-to-service communication.

Tenant isolation:

Tenant-specific Kafka topics (tenantA.*, tenantB.*).

Tenant-specific DB collections or schemas.

Tenant-aware access checks in Profiling and Inference.

NETCELL-EDA Architecture
Security architecture

Kafka & ACLs

Topic ACLs:

- Producers/consumers restricted per tenant.
- Admin operations limited to ops/architecture roles.

Schema Registry:

- Only approved schemas can be registered.
- Backwards/forwards compatibility enforced.

NETCELL-EDA Architecture
Security architecture

Application security

Input validation:

Event Gateway validates payloads and schemas.

Normalizer enforces canonical formats.

Secrets management:

All credentials stored in a secure secrets manager.

No secrets in code or config repos.

NETCELL-EDA Architecture
Security architecture

Logging & audit

Audit logs:

Model deployments, ACL changes, tenant onboarding, and schema changes are logged.

Access logs:

All API calls, dashboard access, and admin actions recorded.

NETCELL-EDA Architecture
Security architecture

Data flow diagram (textual)

NETCELL-EDA Architecture
Security architecture

Main flow (per event)

External system (CRM/Billing/Security)

Emits event (e.g., ContactUpdated).

Event Gateway

Receives HTTP/streaming event.

Authenticates source.

Performs basic validation.

Normalizer

Converts raw payload to canonical event schema.

Adds tenant ID, correlation ID, timestamps.

NETCELL-EDA Architecture
Security architecture

Kafka (EventsHub)

- Writes event to tenant-specific topic (tenantA.contact.updated).
- Replicates across brokers.

Event Processor (e.g., Contact Processor)

- Consumes from Kafka topic.
- Enriches event, derives features.
- Emits enriched event or feature payload.

Profiling Service

- Reads enriched data.
- Updates profile in Document DB.
- Writes features to Feature Store (offline + online).

NETCELL-EDA Architecture
Security architecture

Inference Service

- Receives request (sync or async).
- Loads features from online Feature Store.
- Loads model from Model Registry.
- Runs inference and returns prediction.
- Optionally writes prediction back to Document DB.

Analytics / Dashboards

- Read profiles, features, and predictions from Document DB and Feature Store.
- Present tenant-specific views.

NETCELL-EDA Architecture
Security architecture

Side flows

Model lifecycle:

Data Science pulls training data from Offline Feature Store.

Trains model, registers in Model Registry.

Inference Service picks up new version via deployment pipeline.

Monitoring & logs:

All services emit metrics and logs to centralized observability stack.

NETCELL-EDA Architecture
Security architecture

Monitoring & observability plan

Objectives

Know when something breaks before tenants do.

Trace every event from source to profile to prediction.

Measure latency, throughput, errors, and drift.

NETCELL-EDA Architecture
Security architecture

Metrics

Kafka:

Label: Broker health

Leader/follower status, partition under-replication.

Label: Topic metrics

Messages/sec, lag per consumer group, retention.

Event Gateway & Normalizer:

Label: Request metrics

Requests/sec, error rate, latency percentiles.

Label: Validation failures

Count and rate per source system.

NETCELL-EDA Architecture
Security architecture

Processors:

Label: Consumption metrics

Lag, messages/sec, processing time.

Label: Business metrics

Events per type, per tenant.

Profiling Service:

Label: Profile update latency

Label: DB write success/error rate

Label: Feature write metrics (to Feature Store).

NETCELL-EDA Architecture
Security architecture

Inference Service:

- Label: Inference latency (p50/p95/p99)
- Label: Error rate (per model, per tenant)
- Label: Throughput (requests/sec).

Feature Store:

- Label: Read/write latency
- Label: Batch job success/failure
- Label: Data freshness.

NETCELL-EDA Architecture
Security architecture

Model lifecycle:

Label: Drift metrics

Feature distribution vs training baseline.

Label: Model performance

Accuracy, precision, recall (offline evaluation).

Label: Throughput (requests/sec).

NETCELL-EDA Architecture
Security architecture

Logs & traces

Structured logs:

Correlation ID, tenant ID, event type, service name.

Distributed tracing:

Trace from Event Gateway → Normalizer → Kafka → Processor → Profiling → Inference.

Use OpenTelemetry or similar.

NETCELL-EDA Architecture
Security architecture

Dashboards

Ops dashboard:

Cluster health, Kafka, Kubernetes, DB, Feature Store.

Tenant dashboard:

Events processed, profiles updated, predictions generated, latency.

Model dashboard:

Performance, drift, version usage, error rates.

NETCELL-EDA Architecture
Security architecture

Alerts

Critical alerts:

- Kafka under-replication
- DB unavailable
- Inference latency > threshold
- Error rate spike in Gateway/Processors/Inference
- Drift above configured threshold

Warning alerts:

- Lag increasing
- Resource saturation (CPU/memory)
- Slow Feature Store reads

NETCELL-EDA Architecture
Security architecture

Observability stack

Metrics: Prometheus (or equivalent)

Dashboards: Grafana (or equivalent)

Logs: Centralized log store (ELK/OpenSearch/etc.)

Tracing: OpenTelemetry + tracing backend

NETCELL-EDA Architecture
Security architecture

Thanks