NETCELL EDA PLATFORM

Event-Driven AI Cloudera Integration

NETCELL-EDA Architecture

Cloudera Integration

Cloudera Integration Blueprint (Enterprise Architecture)

This is how entire platform plugs into Cloudera's ecosystem.

This blueprint is designed for **large organizations**, multi-tenant, high-scale, secure.

NETCELL-EDA Architecture

Cloudera Integration

1. Cloudera Components Will Used

Streaming

Cloudera Kafka (Streams Messaging Manager)

Schema Registry

Ranger for Kafka ACLs

Data Lake

Cloudera Data Lake (Iceberg tables)

Hive/Impala for analytics

Ranger for table-level security

Compute

Spark (batch feature generation, model training)

Flink (optional for streaming feature pipelines)

Storage

Cloudera Object Storage (S3/HDFS)

- Model files
- Raw event archives
- Training datasets

Governance

Ranger (security policies)

Atlas (data lineage)

NETCELL-EDA Architecture

Cloudera Integration

Kubernetes

Cloudera Data Flow or external K8s cluster

Runs microservices:

- Event processors
- Profiling service
- Inference service
- Config service
- Model registry
- Feature store API

NETCELL-EDA Architecture

Cloudera Integration

3. Cloudera Security Model (Tenant Isolation)

Kafka:

Ranger policies:

- TenantA can only read/write events with TenantId = A
- Admins can read all topics

Schema Registry:

- Tenant-specific schemas if needed

Data Lake:

Iceberg tables partitioned by TenantId

Ranger policies restrict table partitions

Atlas tracks lineage per tenant

Model Registry:

Tenant-specific model versions

Ranger controls access to model files

Microservices:

Each service uses service accounts

mTLS between services

TenantId enforced at API layer

NETCELL-EDA Architecture

Cloudera Integration

4. Cloudera Deployment Flow (end-to-end)

Step 1 — Event arrives

Goes into Cloudera Kafka

Schema validated

Ranger enforces tenant permissions

Step 2 — Event Processor consumes

Runs in Kubernetes

Uses Schema Registry

Calls Profiling Service

Step 3 — Profiling Service updates profile

Writes to Document DB

Updates Online Feature Store

Calls Inference Service

Step 4 — Inference Service loads models

From Cloudera Object Storage

Metadata from Hive Model Registry

Step 5 — Offline Feature Store updated

Spark jobs compute daily features

Stored in Iceberg tables

Step 6 — Training pipeline runs

Spark trains new models

Writes new versions to Model Registry

Stores model files in Object Storage

Step 7 — Analytics layer reads data

Dashboards query Iceberg via Impala/Hive

Ranger enforces tenant isolation

NETCELL-EDA Architecture

Cloudera Integration

Thank you