NETCELL EDA PLATFORM

Event-Driven AI Platform Services

NETCELL-EDA Architecture

Platform Services

Purpose: Features for entities (Contact, Account, etc.), run the right model, return scores fast enough to be used in event processing.

Used by:

Contact Events Processor

Account/Billing/Security processors

Analytics services (on-demand scoring)

NETCELL-EDA Architecture

Platform Services

1. API design

Endpoint example (REST):

POST /api/inference/contact

Request:

json

```
{
  "TenantId": "tenantA",
  "ContactId": 12345,
  "Features": {
    "state_change_count_last_30_days": 4,
    "time_since_last_state_change_hours": 2.5,
    "total_logins_last_7_days": 12,
    "billing_incidents_last_90_days": 0,
    "security_incidents_last_90_days": 1
  }
}
```

Response:

json

```
{
  "ContactId": 12345,
  "Scores": {
    "Risk": 0.18,
    "Churn": 0.07,
    "Engagement": 0.82
  },
  "ModelInfo": {
    "RiskModelVersion": "risk_v3",
    "ChurnModelVersion": "churn_v2",
    "EngagementModelVersion": "eng_v1"
  }
}
```

NETCELL-EDA Architecture

Platform Services

2. Internal structure

Core components:

Model Loader

- Loads models from Model Registry (e.g. files, DB, S3, Cloudera storage).
- Keeps them in memory for fast inference.
- Supports multiple versions per tenant if needed.

Feature Validator

- Checks that required features are present.
- Applies default values or normalization (scaling, encoding).

Inference Engine

- Runs models (e.g. XGBoost, LightGBM, neural nets).
- Combines outputs into final scores.

Tenant Router

- Chooses which model set to use based on TenantId.
- Allows per-tenant customization in the future.

NETCELL-EDA Architecture
Platform Services

3. Performance & scalability

Stateless service:

No local state per request.

Easy to scale horizontally on Kubernetes.

Autoscaling:

Use HPA (Horizontal Pod Autoscaler) based on CPU or request rate.

Latency target:

Aim for <50–100 ms per inference.

Enough for real-time event processing.

NETCELL-EDA Architecture
Platform Services

4. Integration with event processors

Flow for ContactStateChanged**:**

Contact Events Processor receives event from Kafka.

It:

- loads ContactProfile
- updates state & history
- builds feature vector.

Calls Inference Service:

- POST /api/inference/contact with features.

Receives scores.

Writes scores into ContactProfile.Scores.

Upserts profile to Document DB.

This keeps **profiling logic** centralized and reusable.

NETCELL-EDA Architecture
Platform Services

5. Model registry & versioning

You'll want a simple **Model Registry**:

Stores:

- model files (e.g. risk_v3.pkl, churn_v2.onnx)
- metadata (version, training date, metrics, tenant).

Inference Service:

- loads latest approved version per model type.
- exposes /api/models endpoint for introspection if needed.

NETCELL-EDA Architecture

Platform Services

Thank you