

KEA

Data design

Product description

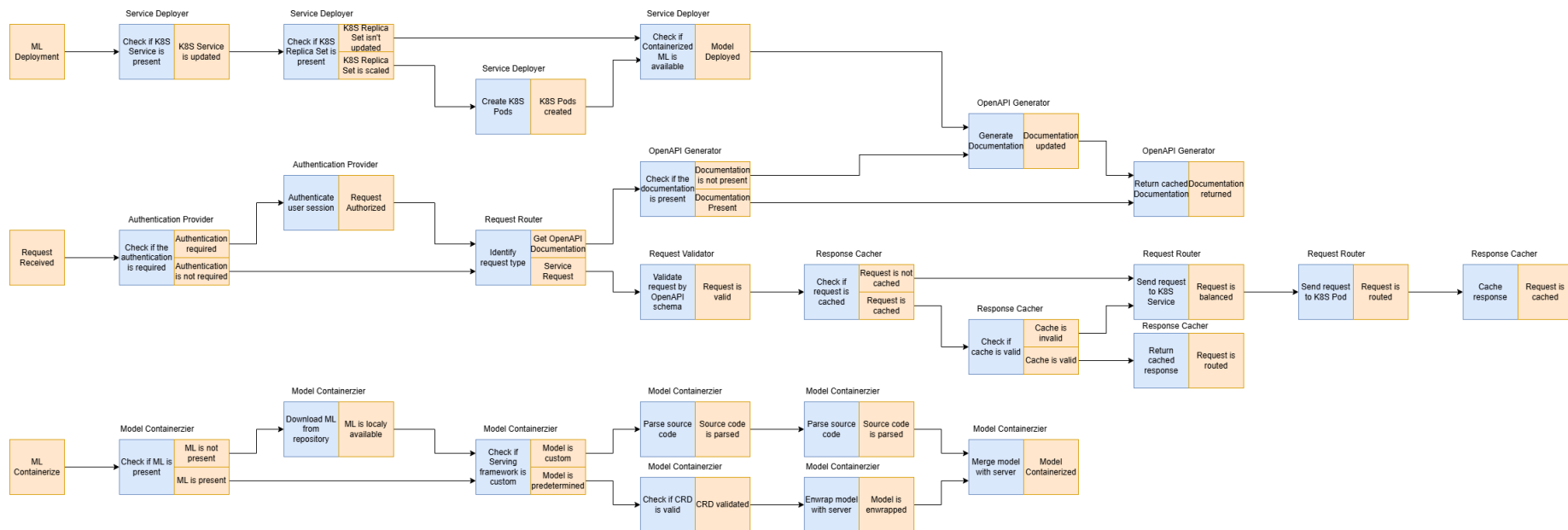
The product is a platform for deploying, managing, and scaling machine learning models in production. It offers a secure, flexible environment for automating ML tasks like model versioning, routing, and monitoring. With Kubernetes integration and containerization support, it's designed for developers, ML engineers, and enterprises needing scalable, reliable ML infrastructure.

Team K8C: Tsurkan Daniel; Dandamaev Gadji; Tsaturyan Konstantin; Smolkin Mikhail

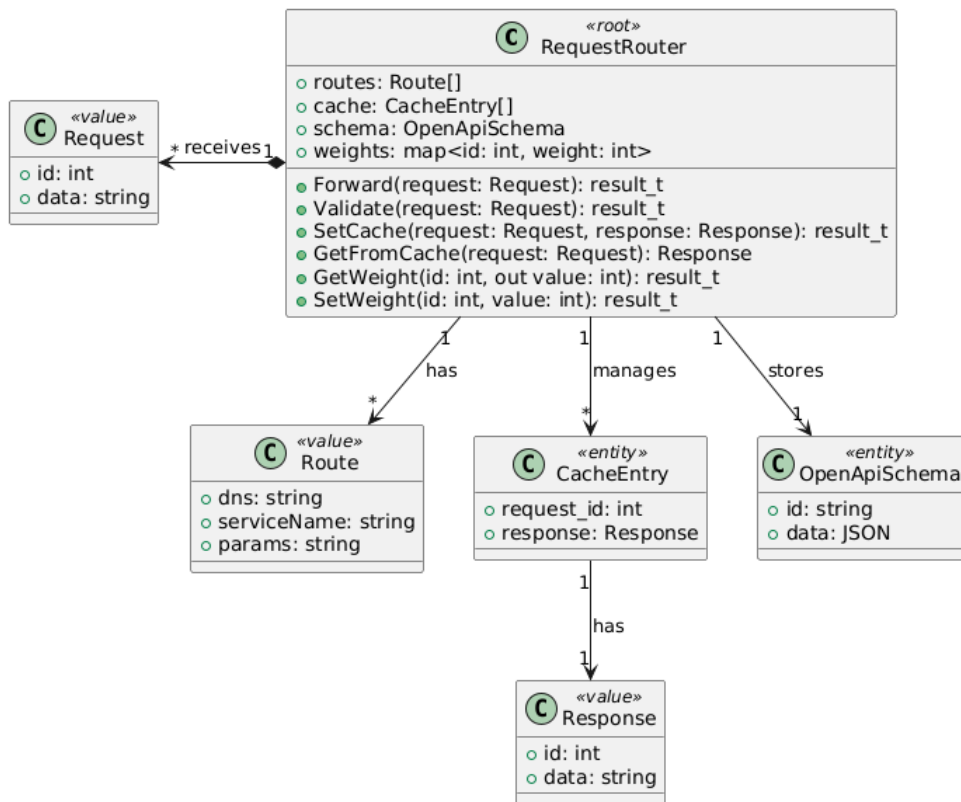
Project repo: <https://github.com/fanglores/Advanced-Software-Design>

This report: [https://github.com/fanglores/Advanced-Software-Design
/blob/master/Practice%20Tasks/Module2/Task_11/Task_11.pdf](https://github.com/fanglores/Advanced-Software-Design/blob/master/Practice%20Tasks/Module2/Task_11/Task_11.pdf)

Event flow



Logical data model RequestRouter



API usage RequestRouter

Use Case: Forward Request

Scenario:

User sends request to a service

Request is being validated by OpenAPI schema

Request is being forwarded to a specific K8s service

RequestRouter Routes and validates API requests, with caching support. ^

POST

/router/loadbalancer Configure load balancer for a deployed ML service



GET

/router/loadbalancer List all load balancer configurations



GET

/router/loadbalancer/{serviceId}
Get load balancer configuration for a specific ML service



DELETE

/router/loadbalancer/{serviceId} Remove load balancer configuration



POST

/router/validate Validate an API request against OpenAPI schema

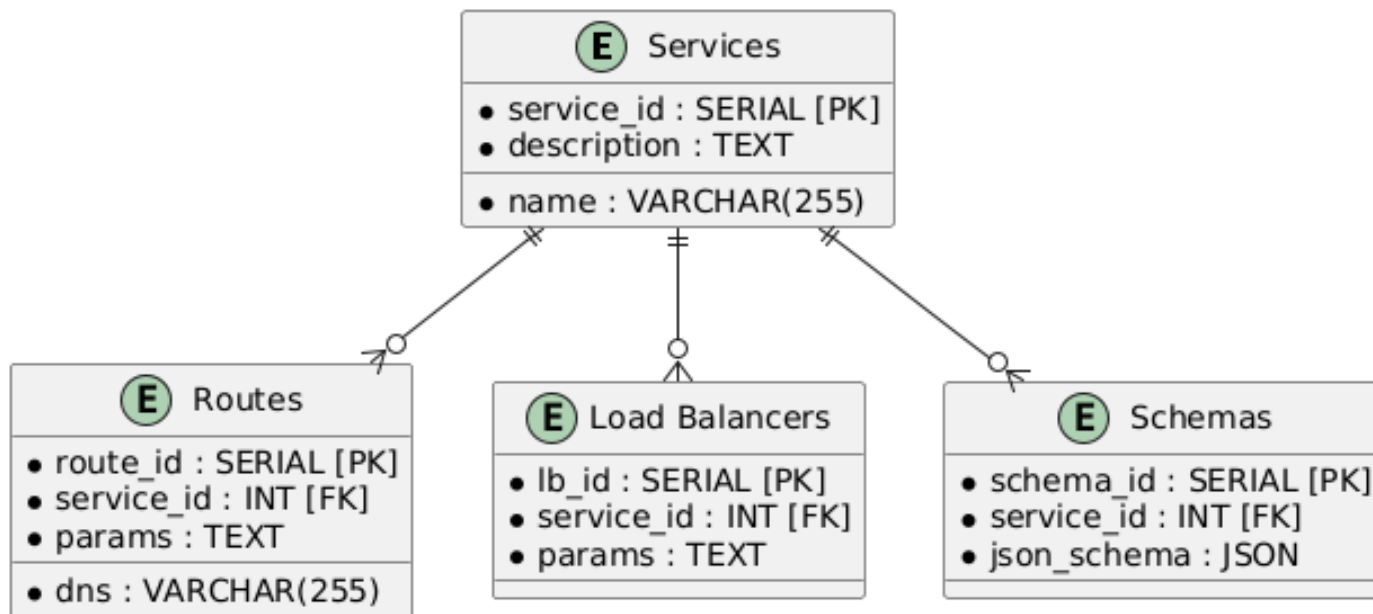


GET

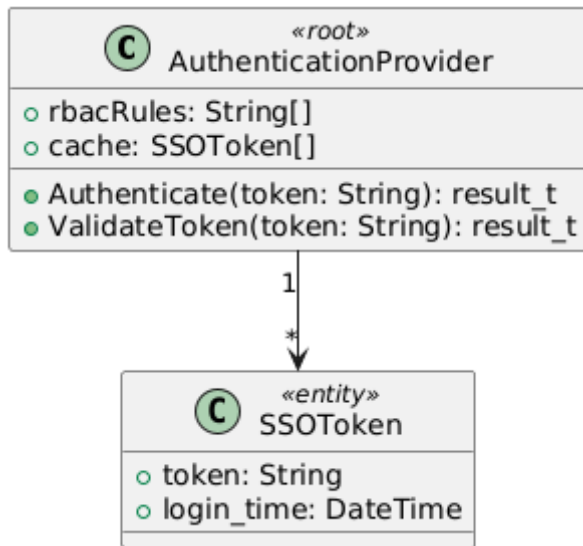
/router/cache Fetch cached API response



Physical schema **RequestRouter**



Logical data model **Authenticator**



API usage Authenticator

Use Case: Authenticate

Scenario:

User sends request

Request is sent to
authentication

Request is checked for
SSO authentication
possibility

Authenticator Handles authentication and RBAC validation.

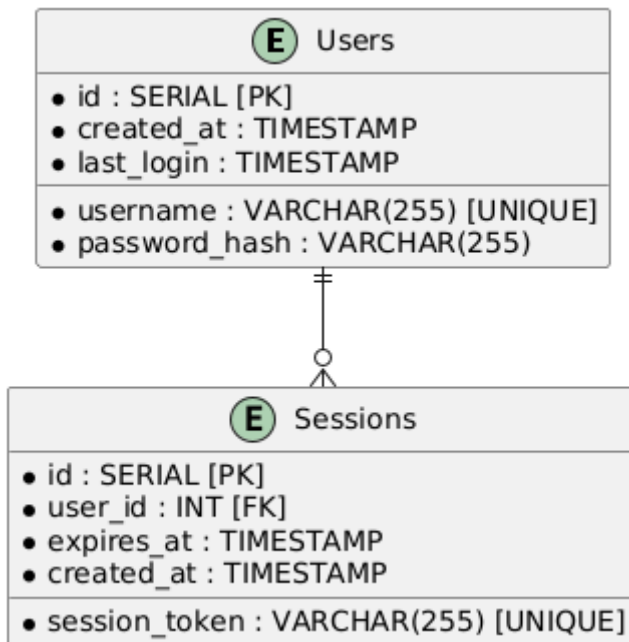


POST

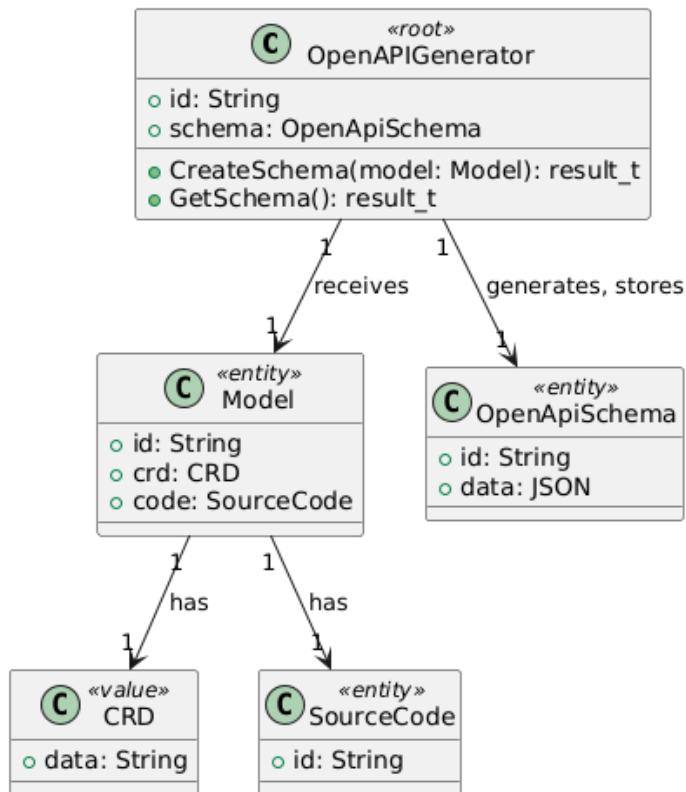
/auth/login Perform SSO authentication



Physical schema **Authenticator**



Logical data model **OpenAPIGenerator**



API usage OpenAPIGenerator

Use Case: Generate
OpenAPI Schema

Scenario:

User sends request to get
schema

OpenApi Generator checks
if schema is present, if not
– creates it, returns actual
schema

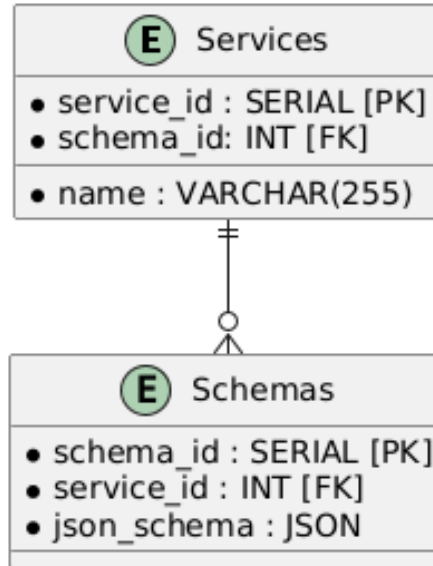
OpenAPIGenerator Generates OpenAPI schemas for ML services. ^

GET

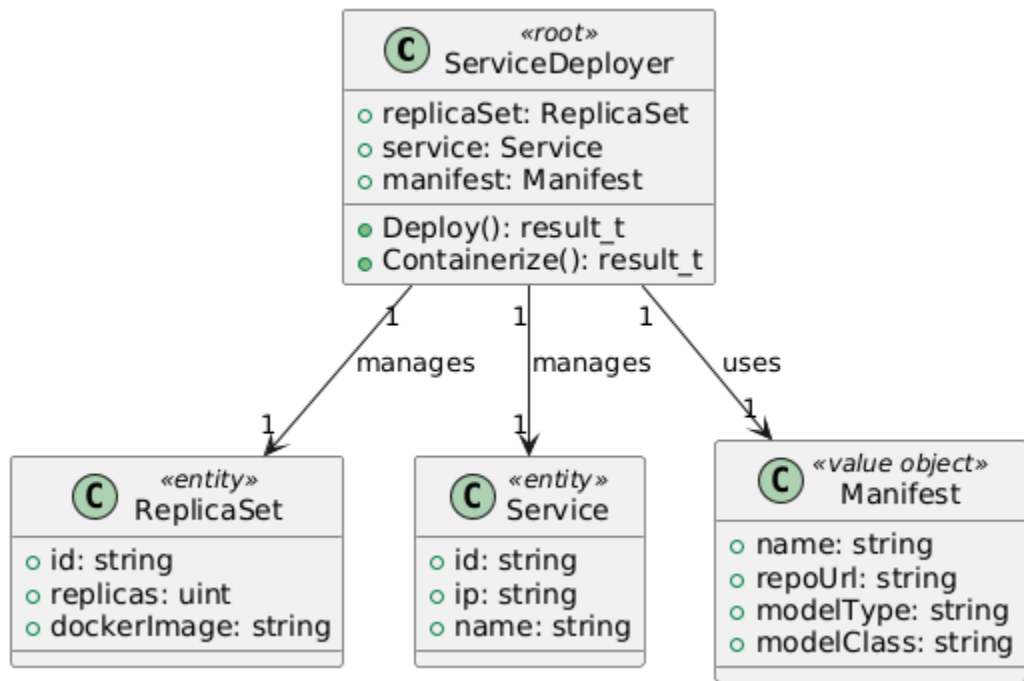
/openapi/{serviceId} Get the OpenAPI schema of a deployed service

✓

Physical schema OpenAPIGenerator



Logical data model **ServiceDeployer**



API usage **ServiceDeployer**

Use Case: Deploy Service

Scenario:

User sends request to
deploy ML Model

ML Wrappers creates
docker container

Docker container is
deployed via service into
K8s

ServiceDeployer Deploys and manages ML services. ^

POST

/services Deploy an ML service



GET

/services List all deployed services



GET

/services/{serviceId} Get service details



PUT

/services/{serviceId} Update an ML service



DELETE

/services/{serviceId} Delete an ML service

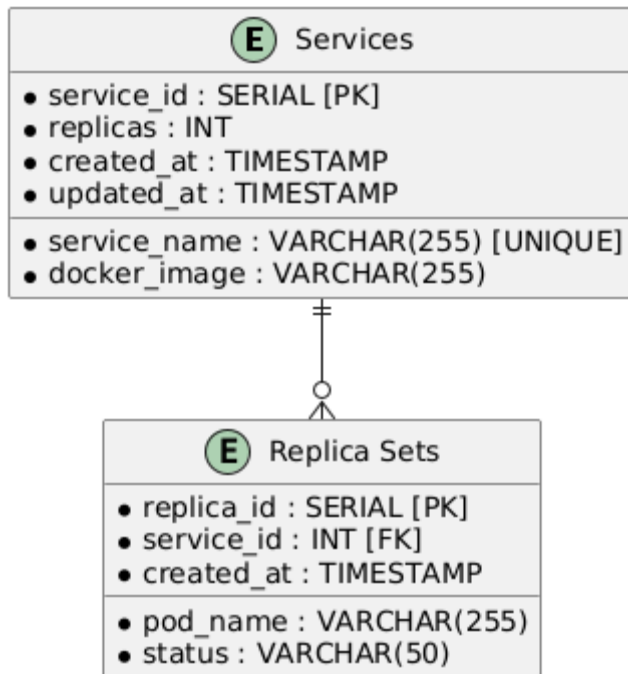


POST

/services/{serviceId}/predict Make a prediction



Physical schema **ServiceDeployer**



Team work

Request Router – Tsaturyan Konstantin

Authenticator – Dandamaev Gadji

OpenAPI Generator – Tsurkan Daniel

Service Deployer – Smolkin Mikhail