KEA

API Design

Product description

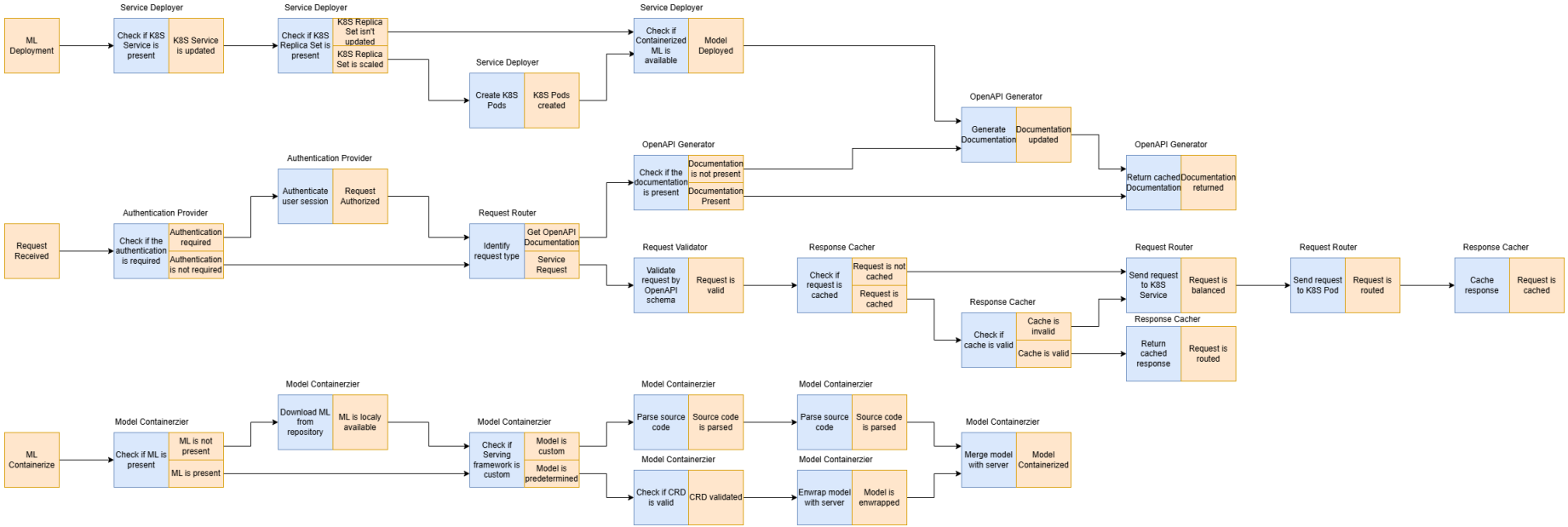
The product is a platform for deploying, managing, and scaling machine learning models in production. It offers a secure, flexible environment for automating ML tasks like model versioning, routing, and monitoring. With Kubernetes integration and containerization support, it's designed for developers,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

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

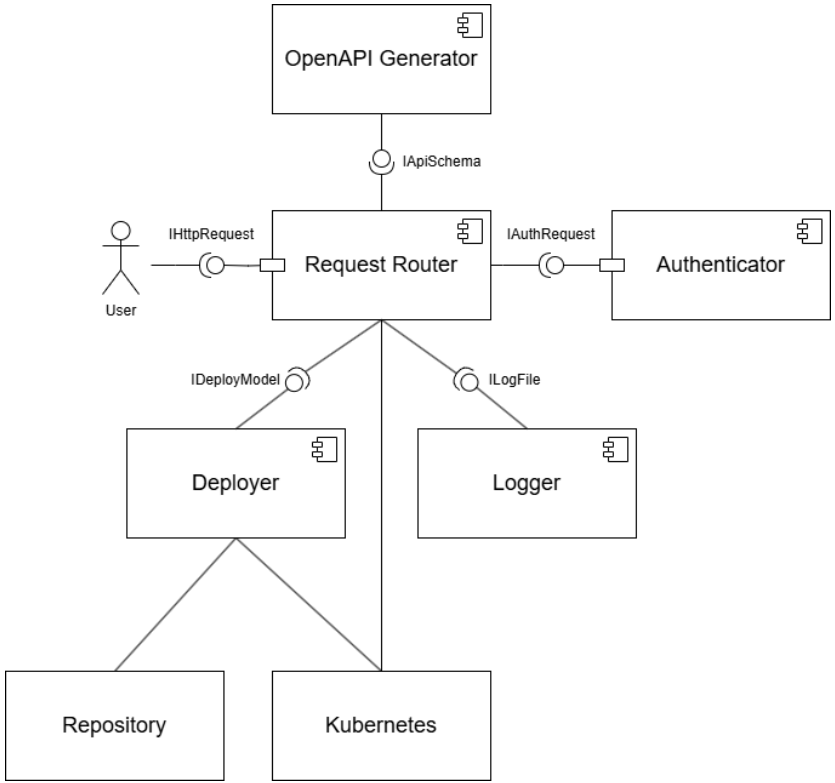
Project repo: https://github.com/fanglores/Advanced-Software-Design

This report: https://github.com/fanglores/Advanced-Software-Design/blob/master/Practice%20Tasks/Module2/Task_10/Task_10.pdf

Event flow


ML Deployment
Service Deployer
Check if K8S Service is present
K8S Service is updated
Service Deployer
Check if K8S Replica Set is present
K8S Replica Set isn't updated
K8S Replica Set is scaled
Service Deployer
Create K8S Pods
K8S Pods created
Service Deployer
Check if Containerized ML is available
Model Deployed
OpenAPI Generator
Generate Documentation
Documentation updated
OpenAPI Generator
Return cached Documentation
Documentation returned
Request Received
Authentication Provider
Check if the authentication is required
Authentication required
Authentication is not required
Authentication Provider
Authenticate user session
Request Authorized
Request Router
Identify request type
Get OpenAPI Documentation
Service Request
OpenAPI Generator
Check if the documentation is present
Documentation is not present
Documentation Present
Request Validator
Validate request by OpenAPI schema
Request is valid
Response Cacher
Check if request is cached
Request is not cached
Request is cached
Request Router
Send request to K8S Service
Request is balanced
Request Router
Send request to K8S Pod
Request is routed
Response Cacher
Cache response
Request is cached
Response Cacher
Check if cache is valid
Cache is invalid
Cache is valid
Response Cacher
Return cached response
Request is routed
ML Containerize
Model Containerzier
Check if ML is present
ML is not present
ML is present
Model Containerzier
Download ML from repository
ML is localy available
Model Containerzier
Check if Serving framework is custom
Model is custom
Model is predetermined
Model Containerzier
Parse source code
Source code is parsed
Model Containerzier
Check if CRD is valid
CRD validated
Model Containerzier
Parse source code
Source code is parsed
Model Containerzier
Enwrap model with server
Model is enwrapped
Model Containerzier
Merge model with server
Model Containerized

Service diagram



Open API

Paths:

- /router/loadbalancer
- /router/loadbalancer/{serviceId}
- /services
- /services/{serviceId}
- /services/{serviceId}/predict
- /auth/login
- /openapi/{serviceId}
- /logs/{serviceId}
- /router/validate
- /router/cache

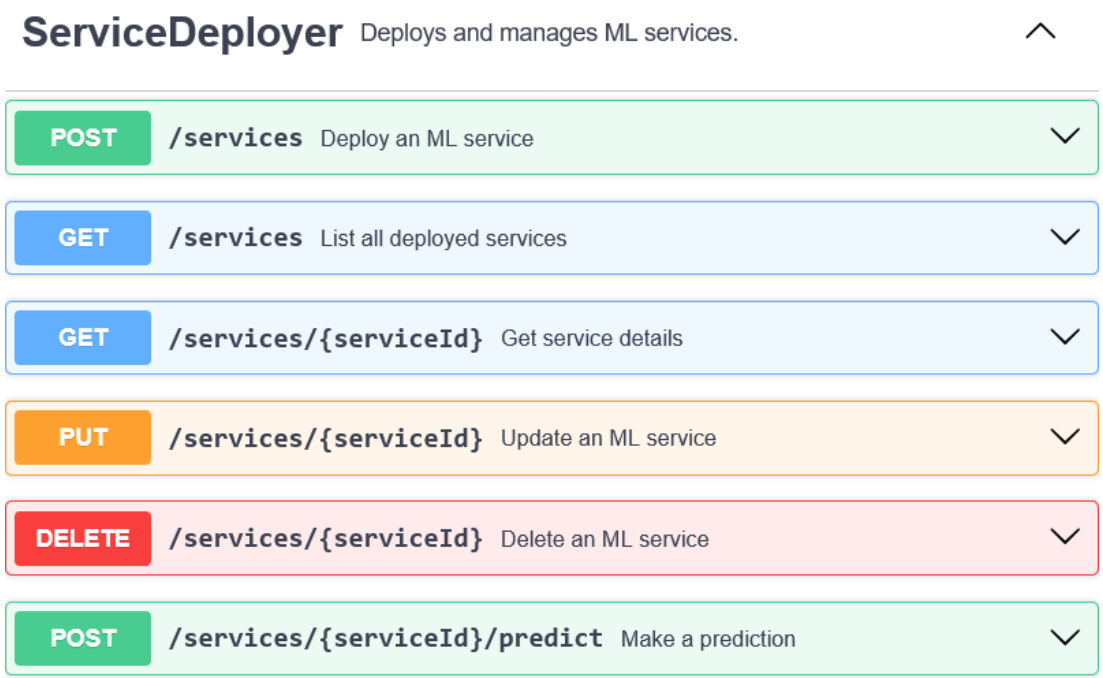
Schemas:

- LoadBalancerConfig
- LoadBalancerResponse
- LoadBalancerInfo
- LoadBalancerDetails
- DeployRequest
- DeployResponse
- SenriceInfo
- ServiceDetails
- UpdateRequest
- UpdateResponse
- PredictRequest
- PredictResponse
- LoginRequest
- LoginResponse
- Logs
- Validation Request

https://github.com/fanglores/Advanced-Software-Design/blob/master/PracticeTasks/Module2/Task_10/openapi.yaml

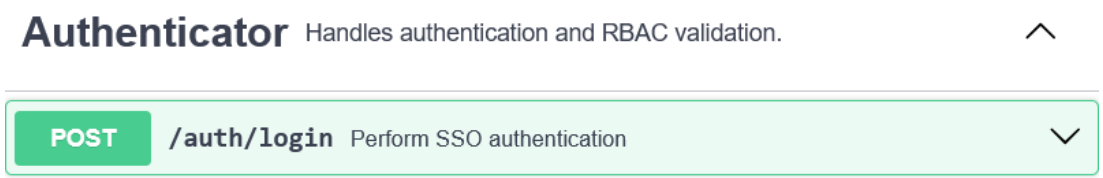
API usage **ServiceDeployer**

Use Case: DeployService
Scenario:
User sends request to deploy ML Model
ML Wrappers creates docker container
Docker container is deployed via service into K8s



API usage **Authenticator**

Use Case: Authenticate
Scenario:
User sends request
Request is sent to authentication
Request is checked for SSO authentication possibility



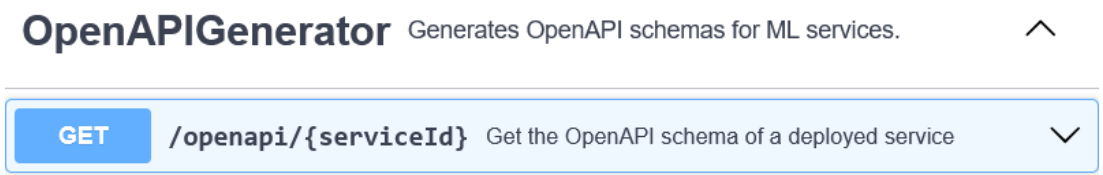
API usage **OpenAPIGenerator**

Use Case: Generate OpenAPI Schema

Scenario:

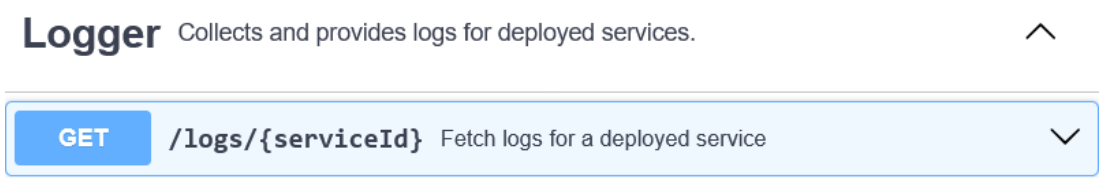
User sends request to get schema

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



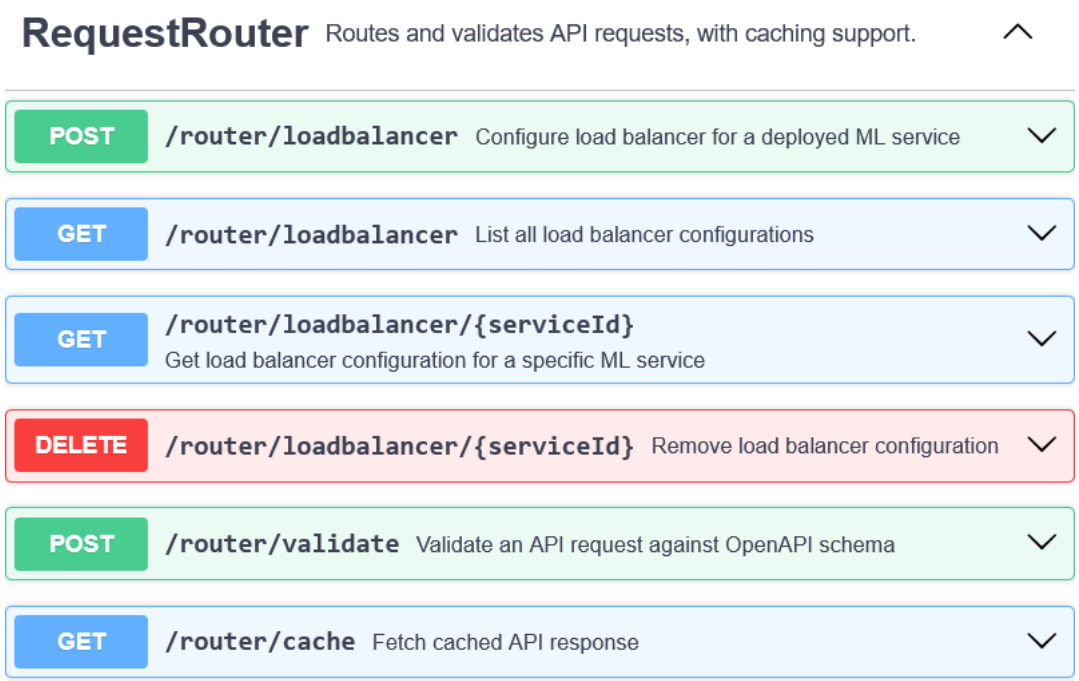
API usage **Logger**

Use Case: Logging
Scenario:
User send request to fetch logs
Logger retrieves logs for a service



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



Solution stack (prepare)

Implementation

- API definition: OpenAPI
- Connection server for API: python gunicorn
- App framework: python flask
- Serialization/state format: json

Asynchronous interactions (optional)

- Message queue: (e.g. rabbitmq, kafka, redis streams…)
- Messaging client library: (e.g. celery, spring stream …)

Testing tools pytest

Operations

- App initializer: Spring Initializer
- Code build: makefile
- CI/CD pipeline: github
- Delivery method: docker
- Logging & monitoring (logrotate, prometheus, ELK, grafana, …) (optional)

Some references

https://github.com/mfornos/awesome-microservices

https://awesomeopensource.com/projects/microservices-architecture

https://www.redhat.com/en/blog/comparing-openapi-grpc

https://cloud.google.com/apis/design/resources