Self-Documentating API-Gateway

Solozobov Evgeny, Verhovod Nikita, Polienko Artem, Kochnev Victor

Task 6

Project statement

A typical ingress implementation in k8s forwards incoming traffic to a particular application. An improved API Gateway would integrate Single Sign-On, automatic OpenAPI schema generation, request validation and response caching. The goal of the project is to develop such a Kubernetes operator to extend existing ingress implementations to API Gateway. The project should provide means to define API schema using Custom Resource Definitions in K8s directly and generate them from the source code following a set of conventions.

Team: WR

Project repo: github.com/laym0n/wr

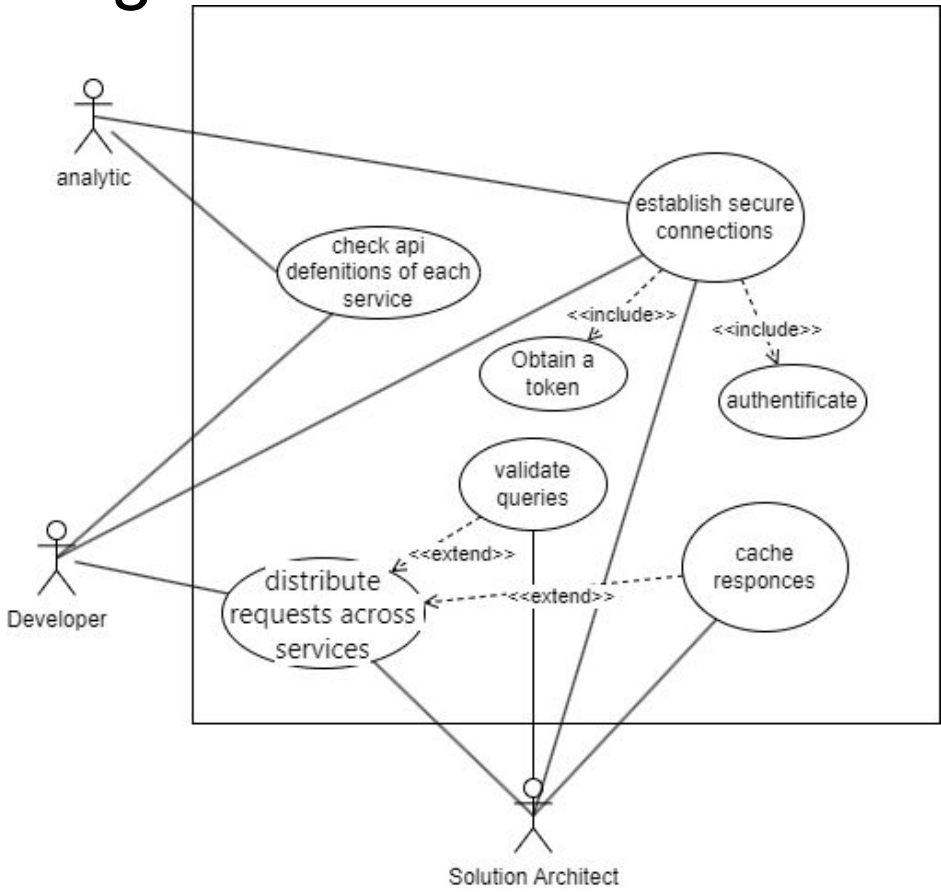
This report: tiny.cc/project-team-wr

Use case diagram

MVP

Prioritization of use cases

1) distribute requests across services
2) check api definitions of each service
3) establish secure connections
4) cache responses



Interaction analysis

Use case	Cooperation name	Used roles	Candidate classes
Check api definitions	Validate queries	Analytic Developer	Request Validator
Establish secure connection	Established	Analytic Developer	Autentification Provider
Distribute queries	Distributed by	Developer Solution architector	Distributer
Validate queries	Validated by	Developer Solution architector	Request validator
Cache responses	Cached by	Developer Solution architector	Response cacher

Updated class diagram

Responce Cacher	
know: upload Rules know: upload Config do: store response do: find Response	

Request Validator	
know: upload Rules know: upload Config do: get schema do: validate Request do: check token	OpenApi Generator Autentication provider OpenApi schema

OpenApi schema	
know: store info for openApi's file	

OpenApi Generator	
do: generate OpenApi schema do: get definitions	OpenApi Schema

Autentication Provider	
know: upload Config do: autentificate user do: validate token do: expire all token	

Distributor	
do: route Requests do: add Route do: get routes	Responce Cacher Request Validator