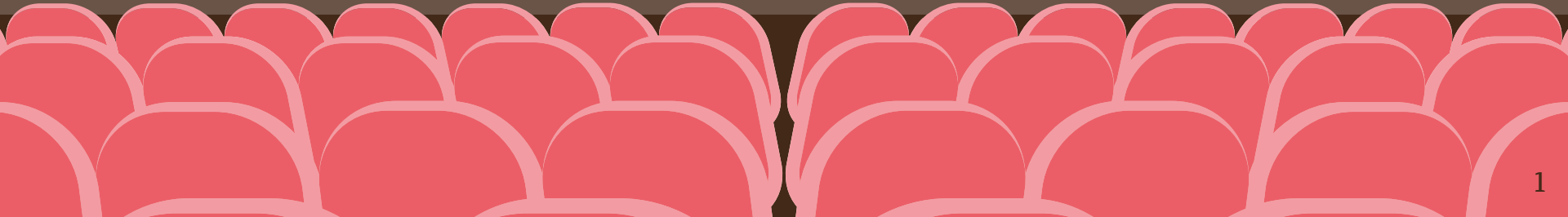


# ***SERVICE-ORIENTED SOFTWARE ENGINEERING***

***PROJECT: WHAT YOU WATCHED (WYW)***



# ***CONTENTS OF THIS PRESENTATION***



**SECTION 1** Introduction

**SECTION 2** Static view

**SECTION 3** Dynamic view

**SECTION 4** Software stack

# ***ABOUT THE PROJECT***



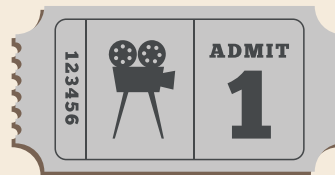
Develop a social platform where users keep track of all the films that they've watched and visualize the ones watched by friends (a simplified version of trak.tv).

Each film should have some information along with one or more ratings taken from specific site in the movie industry (for example IMDB).

# FUNCTIONAL REQUIREMENTS

## Profile

Allow the user to manage his information.



## List watched films

Allow the users to list all the watched movies.

## Add watched film

Allow the users to mark a movie as watched.



## Get films

Allow the users to list films.

## Get film rating

Allow the users to obtain the ratings for a specific film.



## Get people for film

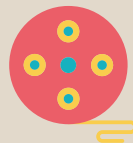
Get the set of people that has worked on a specific movie.

# ***MICROSERVICE-ORIENTED ARCHITECTURE***



## **Loose Coupling**

Less dependency on each other.



## **Service Abstraction**

Services hide the logic they encapsulate from the outside world.

# ***MICROSERVICE-ORIENTED ARCHITECTURE***

## **Service Reusability**

Logic is divided into services with the intent of maximizing reuse.



## **Service Autonomy**

Services should have control over the logic they encapsulate.

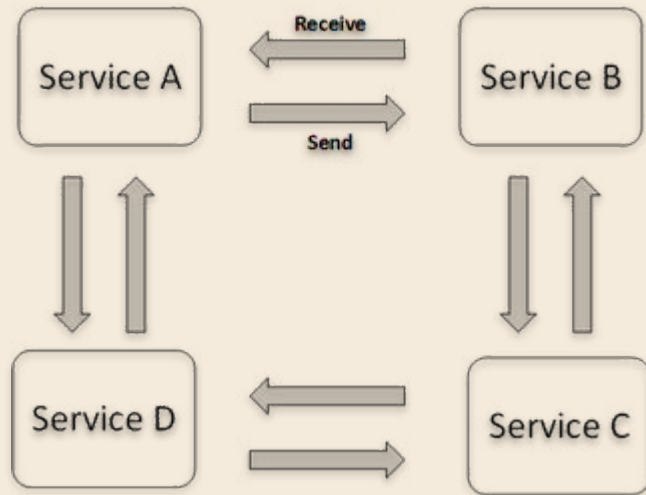


# ORCHESTRATION VS CHOREOGRAPHY

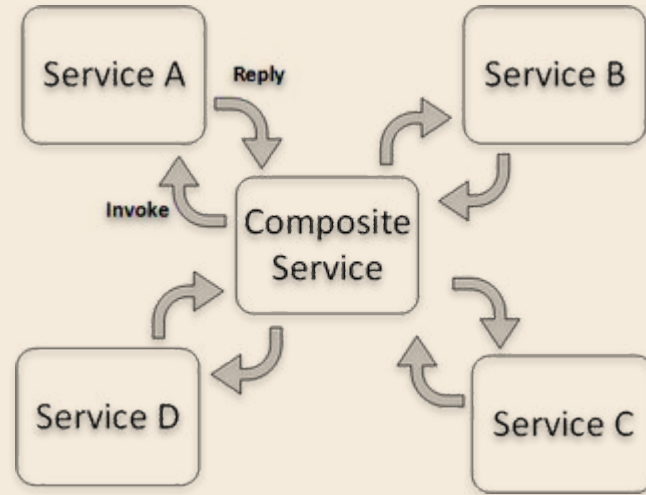
As the previous architecture suggests, the team has decided to opt for a **Choreography** as service composition pattern.

- Decentralized approach
- No Single-point-of-failure

## Choreography



## Orchestration



# ***STATIC ARCHITECTURE***





# ***(MICRO)SERVICES***



## **Film**

Manages films  
information.



## **User**

Manages users of  
the system.



## **Watched**

Manages watched  
film for a user logic.



## **Person**

Manages roles in a  
film production.



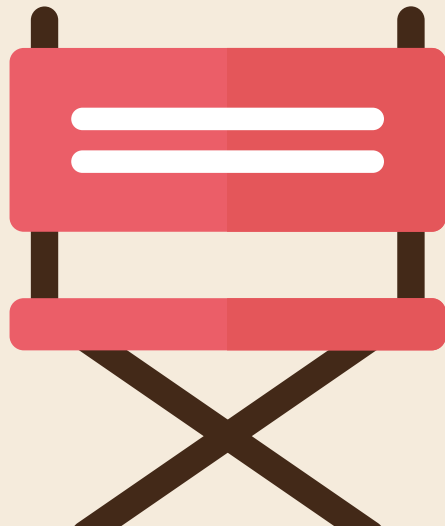
## **OMDB**

Used to retrieve  
film ratings.

# ***SERVICE PROXY, REGISTRY, LOAD BALANCER***

## **Service Proxy**

Ensures traffic is routed to the right destination service or container and to apply security policies.



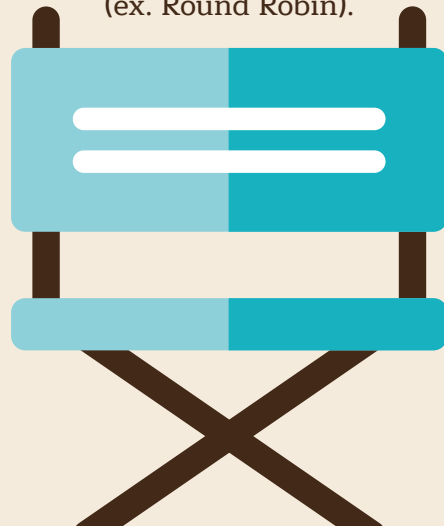
## **Service Registry**

Keeps track of all the available microservices in the cluster.

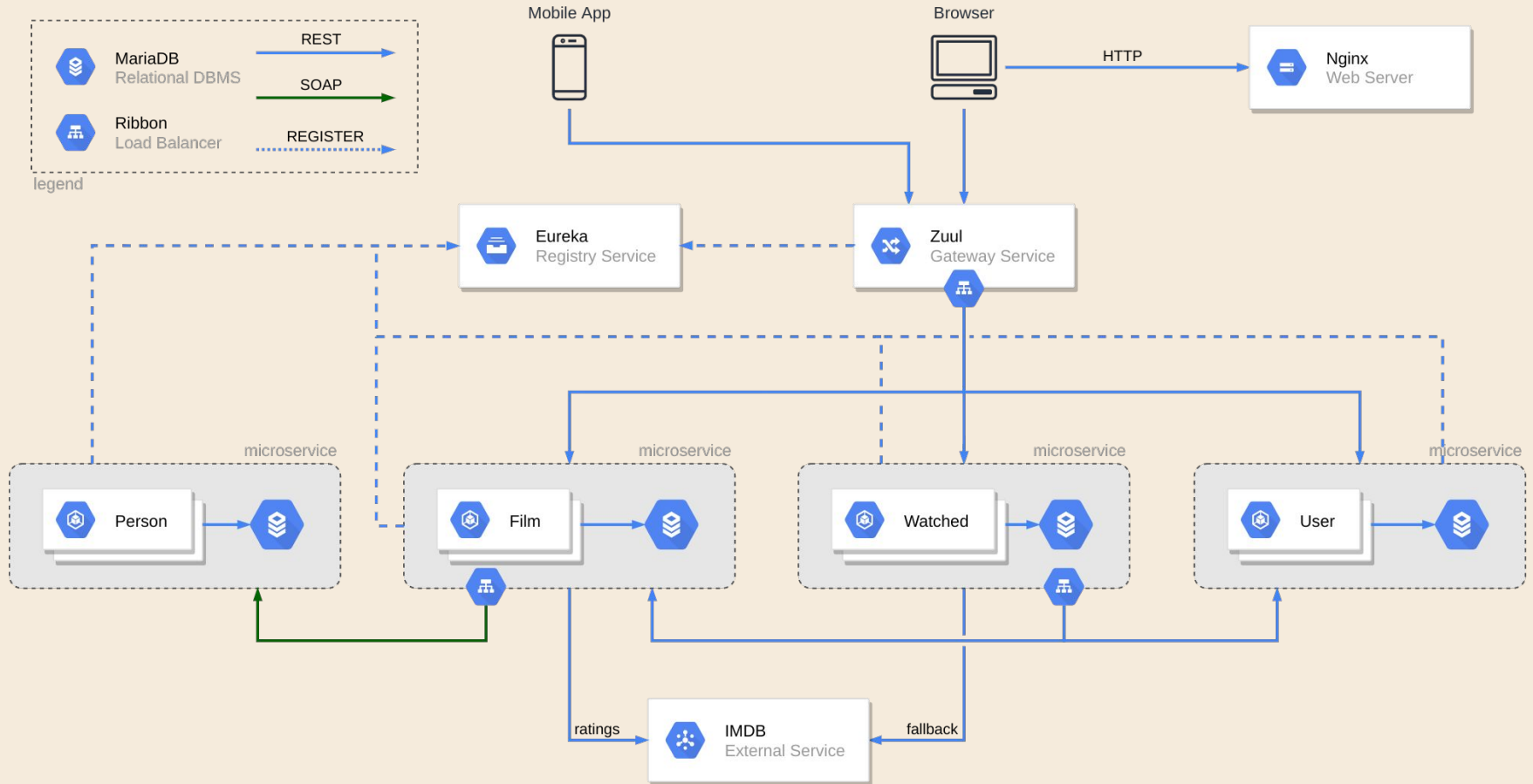


## **Service Load Balancer**

Distributes the working load based on the available instances and one balancing algorithm (ex. Round Robin).



# OUTLINE ARCHITECTURE



# ***DATABASE***

## **Watched-Film**

Stores information for watched films by users.

## **Film**

Stores the film information.

## **Person**

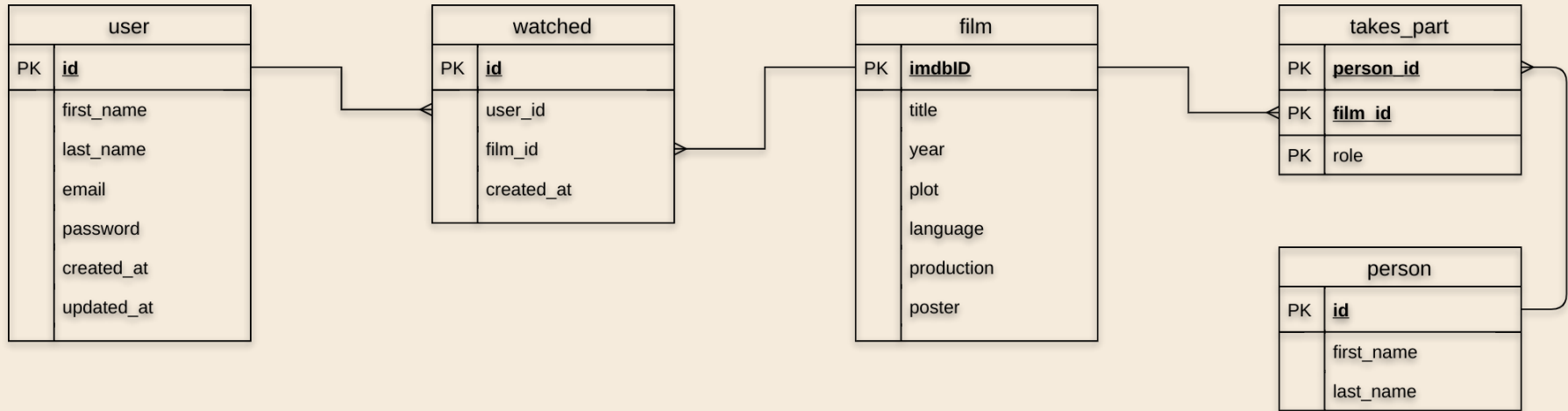
Stores information relatives to person comparing in films.

## **User**

Stores information about users of the platform.

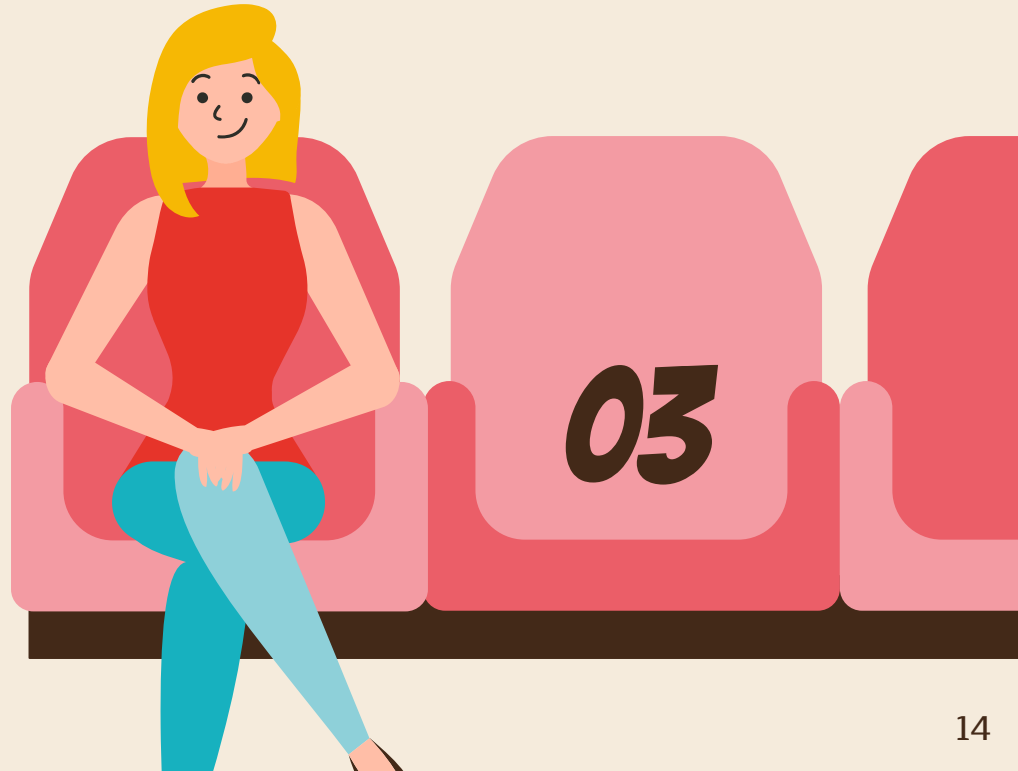


# ***DATABASE SCHEME***

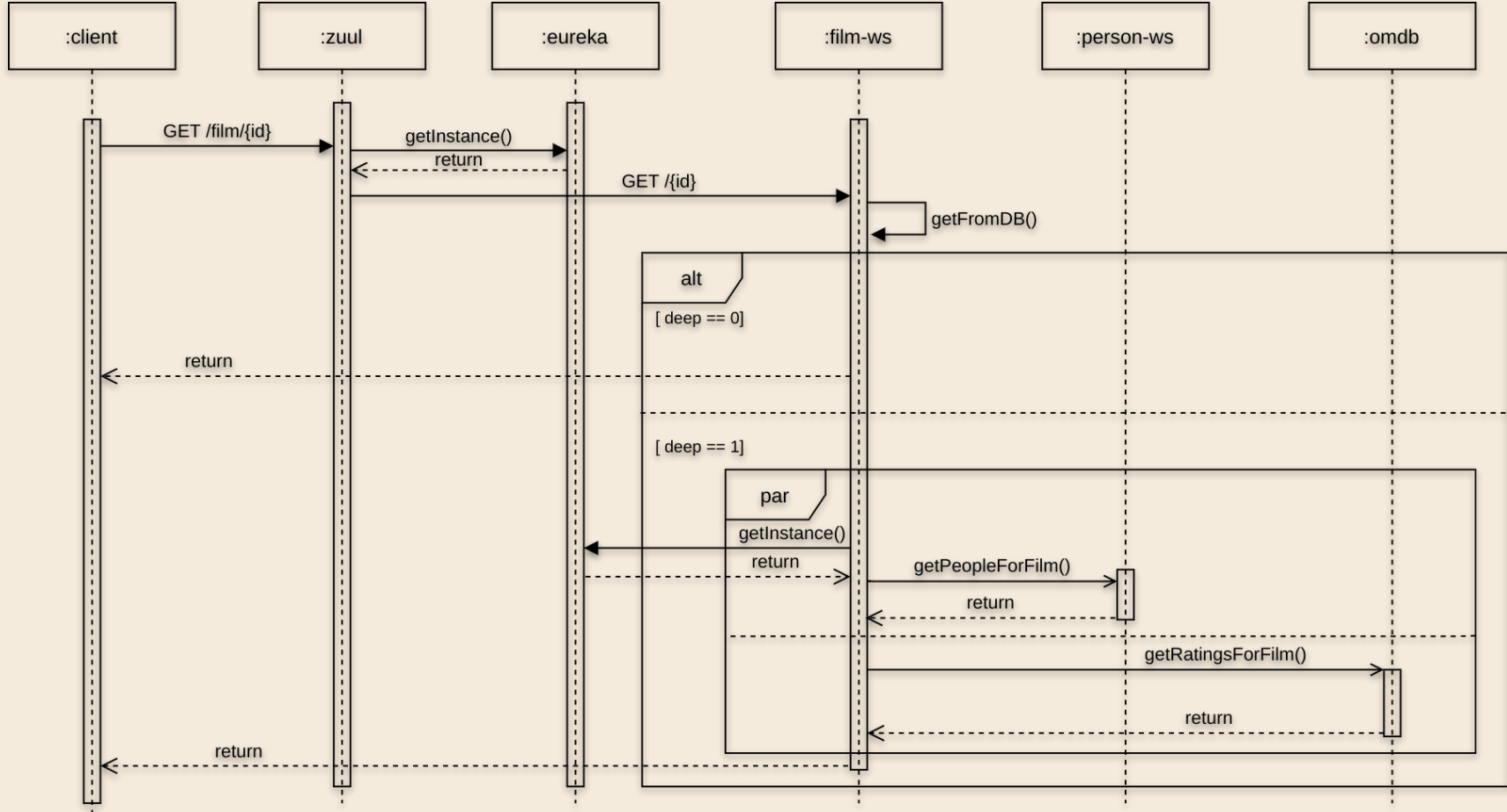


IMPORTANT: It's only the logic DB diagram. The microservices way don't allow a single monolithic relational database

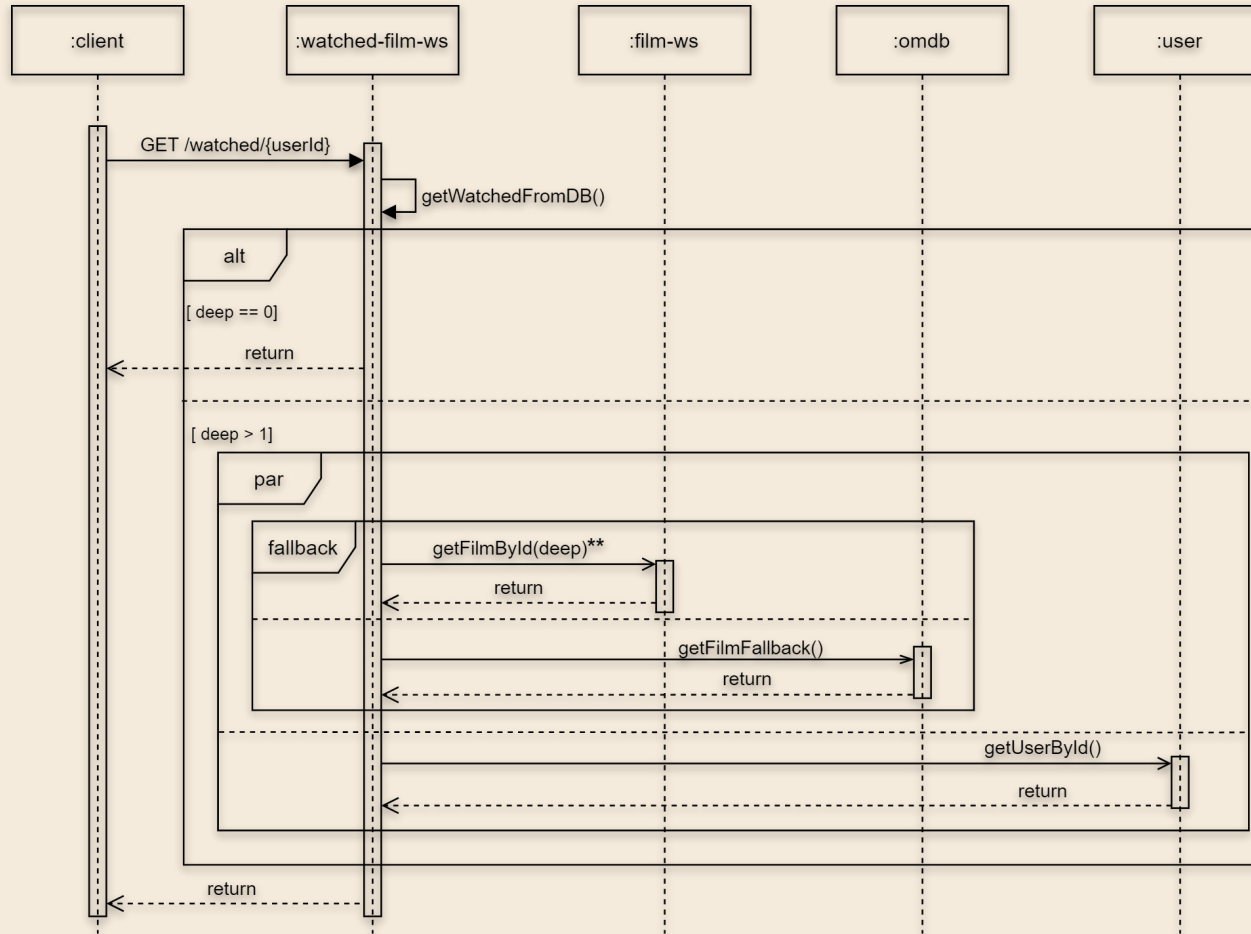
# ***DYNAMIC VIEW***



# SEQUENCE DIAGRAM - GET FILM BY ID



# SEQUENCE DIAGRAM - GET WATCHED BY USER

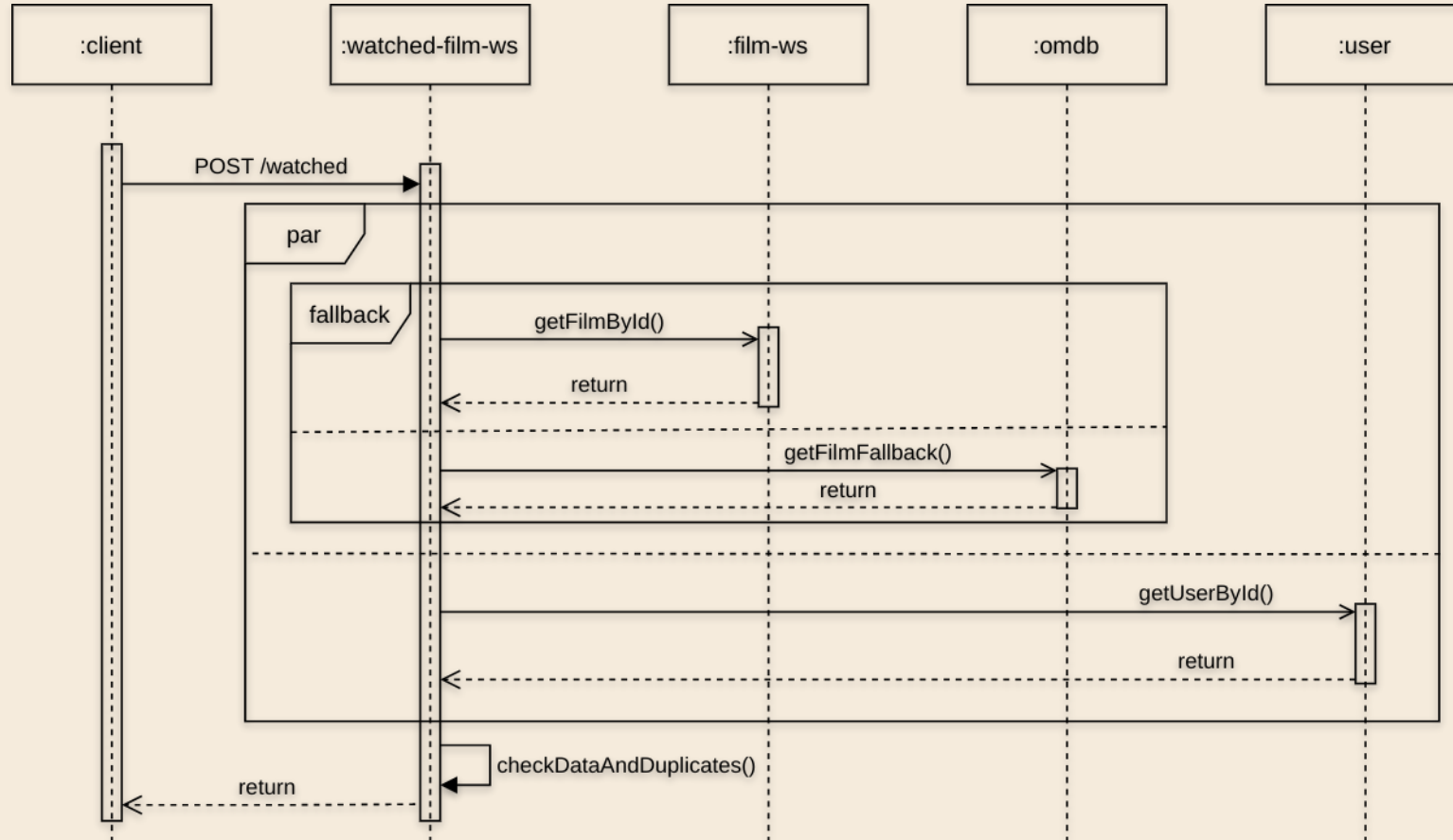


**Note:** In order to simplify the diagram the communication (as in previous slide) with proxy and registry has been omitted.

**\*\*:** This call will generate the scenario illustrated in the previous sequence diagram.



# SEQUENCE DIAGRAM - ADD WATCHED FILM



**Note:** In order to simplify the diagram the communication (as in previous slide) with proxy and registry has been omitted.



# ***SOFTWARE STACK***



# ***BACKEND***

## **Apache CXF**

Open source  
framework for web  
services

## **Spring Boot**

Framework for Java  
application

## **Netflix Stack**

Software stack for  
service-oriented  
architecture

## **Spring JPA**

ORM for  
persistence layer

## **MariaDB**

Relational DBMS

# NETFLIX OSS

## NETFLIX STACK

**FEIGN**  
Http client

**EUREKA**  
Service registry

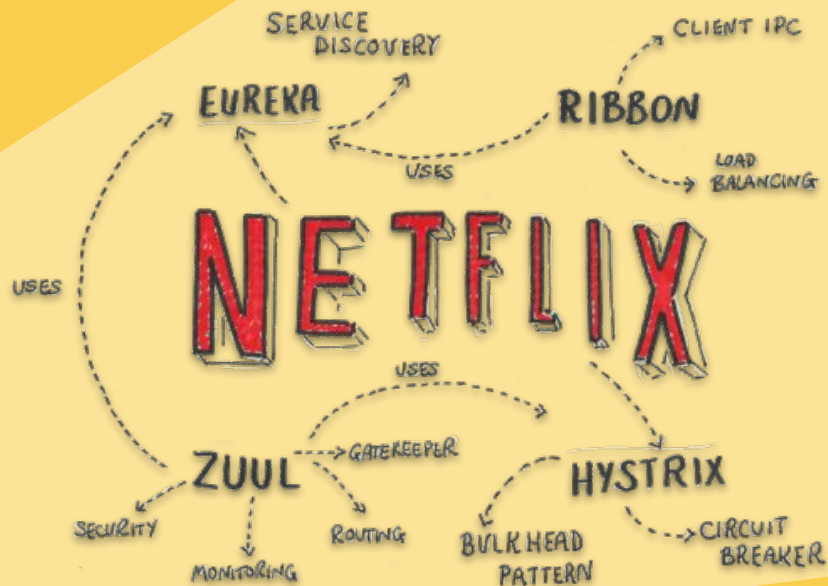
**HYSTRIX**  
Fault tolerance and  
fallback factory

**ZUUL**  
Service router/gateway

**RIBBON**  
Client-side load balancer

# NETFLIX OSS

## NETFLIX STACK



# FRONTEND



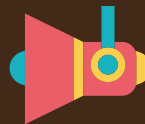
## VueJS

OS JS lib for building interactive web ui, focused on the ViewModel layer of the MVVM pattern



## BootstrapVue

Framework for building responsive, mobile-first sites



## Fetch

Javascript API for accessing and manipulating parts of the HTTP pipeline, such as requests and responses



## PWA

Allow an user experiences similar to native applications on desktop and mobile devices



# docker

## Docker

Containerized application

Images repository:

- Apache tomcat
- MariaDB
- DB adminer
- Ngnix

## Docker-compose(.yaml)

- Multi-container Docker application
- Scale single web service by spawning multiple instances for load-balancing

# FOLDERS STRUCTURE, MAVEN AND NPM

- **Three** layer **hierarchical** folders structure and pom.xml:
  - a. Root pom.xml
  - b. Microservices pom.xml
  - c. Single microservice pom.xml

notes: *<modules>* and *<parent>* tags

- **Archetype** generation
- **NPM** the frontend package manager

```
|-- README.md
|-- docker-compose.dev.yml
|-- docker-compose.prod.yml
|-- docker-compose.yml
|-- pom.xml
|-- frontend
|   |-- dist
|   |-- docker
|   |-- node_modules
|   |-- package.json
|   |-- public
|   |-- src
|   `-- vue.config.js
|-- microservices
|   |-- eureka-registry
|   |-- film
|   |-- person
|   |-- pom.xml
|   |-- rest-ws-archetype
|   |-- user
|   |-- watched-film
|   `-- zuul
|-- utils
|   |-- docker-disable
|   `-- imdb2sql
```



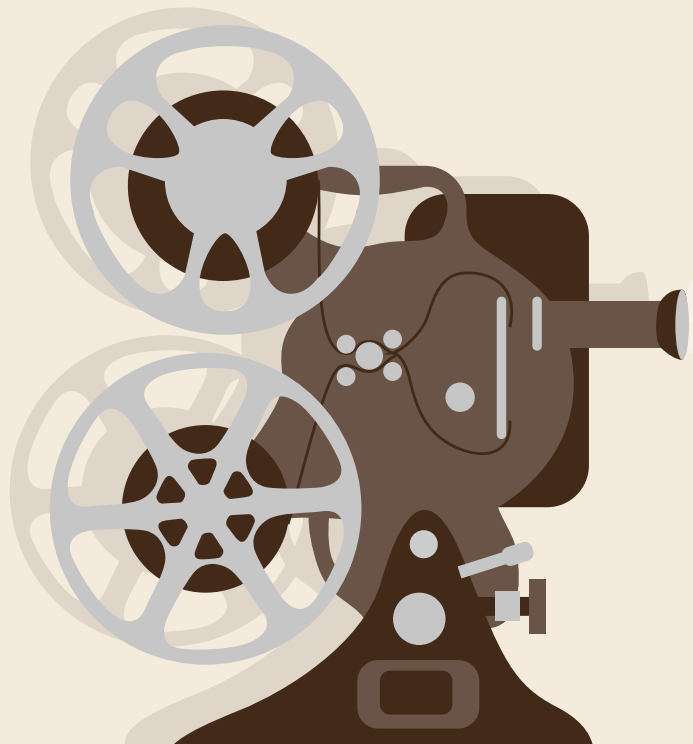


# ***THANKS***

Do you have any questions?



CREDITS: This presentation template was created by Slidesgo, including icons by Flaticon, and infographics & images by Freepik.



# ***RESOURCES***

[cxf.apache.org](https://cxf.apache.org)

[spring.io/projects/spring-boot](https://spring.io/projects/spring-boot)

[github.com/Netflix/zuul](https://github.com/Netflix/zuul)

[github.com/Netflix/eureka](https://github.com/Netflix/eureka)

[github.com/OpenFeign/feign](https://github.com/OpenFeign/feign)

[github.com/Netflix/Hystrix](https://github.com/Netflix/Hystrix)

[vuejs.org](https://vuejs.org)

[bootstrap-vue.org](https://bootstrap-vue.org)

[google.com/search?q=\\*](https://google.com/search?q=)

