Movie and Book Recommendation System

Design Document

Application architecture

The application is designed using a microservices-based architecture for efficient scaling and building loosely coupled modules for fast and efficient development. The application shall include the following high-level microservices:

- Eureka Discovery Service
- Config Service
- User Service
- Movie Service
- Recommendation Service
- Application gateway Service
- Client View Service

Technology Stack

Backend

- Spring Boot and Spring cloud for application configuration
- Maven for building, testing and running the application
- MongoDB

Frontend

- ReactJS
- Javascript
- HTML

Services Overview

Config Service

Config server is where all configurable parameters of all microservices are stored and maintained.

It is more like externalizing properties/resource files out of the project codebase to an external service altogether so that any changes to any given property does not necessitate the re-deployment of a service which is using that property.

Eureka Discovery Service

Discovery service is the one of the key tenets of a microservice based architecture.
Eureka is the Netflix Service Discovery Server and Client.
Discovery service creates a registry which allows services to find and communicate with each other without hard-coding hostname and port.

User Service

The user service contains all users for the system. This service interacts with the Mongo database to manage user data.

Movie / Book Service

Movie service handles the operations regarding the Movie and Book database. It interacts with the Mongo database to manage the movie/book data.

Recommendation Service

It Is in charge of recommendation logic. Its database contains basic information about users(id), movies(id) and books(id) and as well as information about which user liked which movie/book.

Application Gateway Service

This application gateway allows any browser, mobile app or other user interfaces to consume services from multiple hosts. This is the front door of the application and connects the Client View with the multiple backend services that are running independently via a single entry-point.

Client View Service

This is the React service that will be running on the user's browser, it handles the views and fetching responses from the Recommendation

Services Interaction Diagram