

ERSMS, Group F, Project Documentation

Hubert Dwornik, Michał Łezka, Jakub Mazur
Michał Sar, Krzysztof Rudnicki

June 18, 2024

1 System Architecture

We have designed and implemented 4 microservices, all of microservices are written in Python using Flask framework

AI recommendations Based on a list of movies ids, calculates and returns list of ids recommended for user who likes given movies

Analytics Holds information about number of ratings, number of users, average movie ratings, ratings of given user which are later used in webinterface to show data for admins concerning website usage

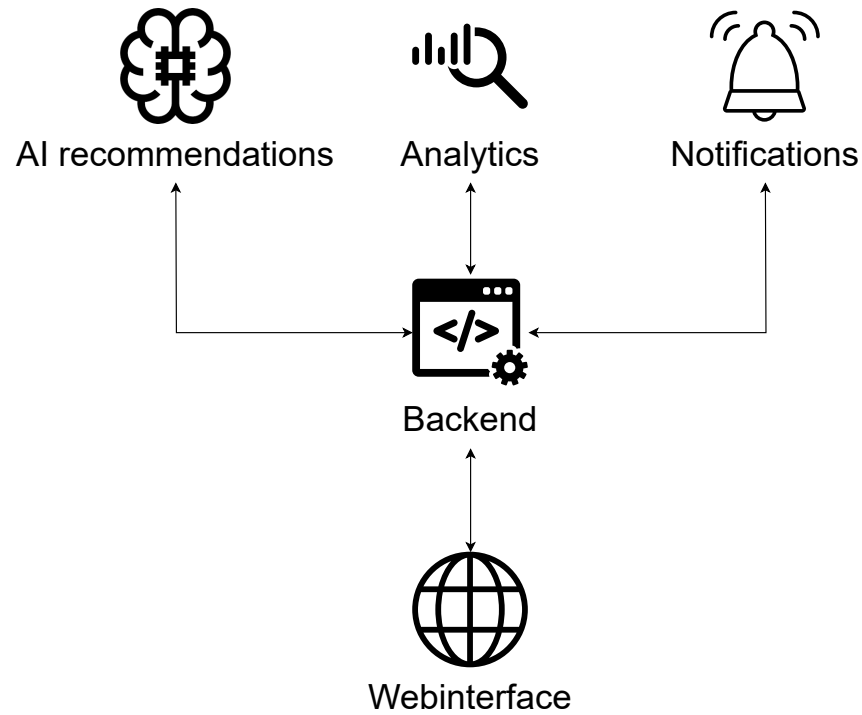
Notifications Notifies user whenever there is a new movie recommend by AI recommender for them

Backend Updates database and maintains all ongoing and incoming communication between all microservices, both between microservices and from microservices to the webinterface

Caching We implemented two caches

1. Backend is cached inside Analytics Service
Analytics service holds data about users and movies, in order to not pull all the data from backend every time we update analytics (for example every day), we keep the cache of backend data in our analytics service
2. AI recommendations are cached inside Notification Service
AI recommendations can change whenever new movie appears, notification service keeps the cache of ai recommendation service in order to not pull all the data from the ai recommendation every time it wants to notify the users about new movies to recommend

Figure 1: System architecture representation, webinterface although not part of microservices included to show relation with backend



- 2 Automated Infrastructure Management solution
- 3 Federated authorization and authentication management in the project
- 4 Threat model with mitigations