**CSE 586 DISTRIBUTED SYSTEMS PROJECT 1 - PHASE 3**

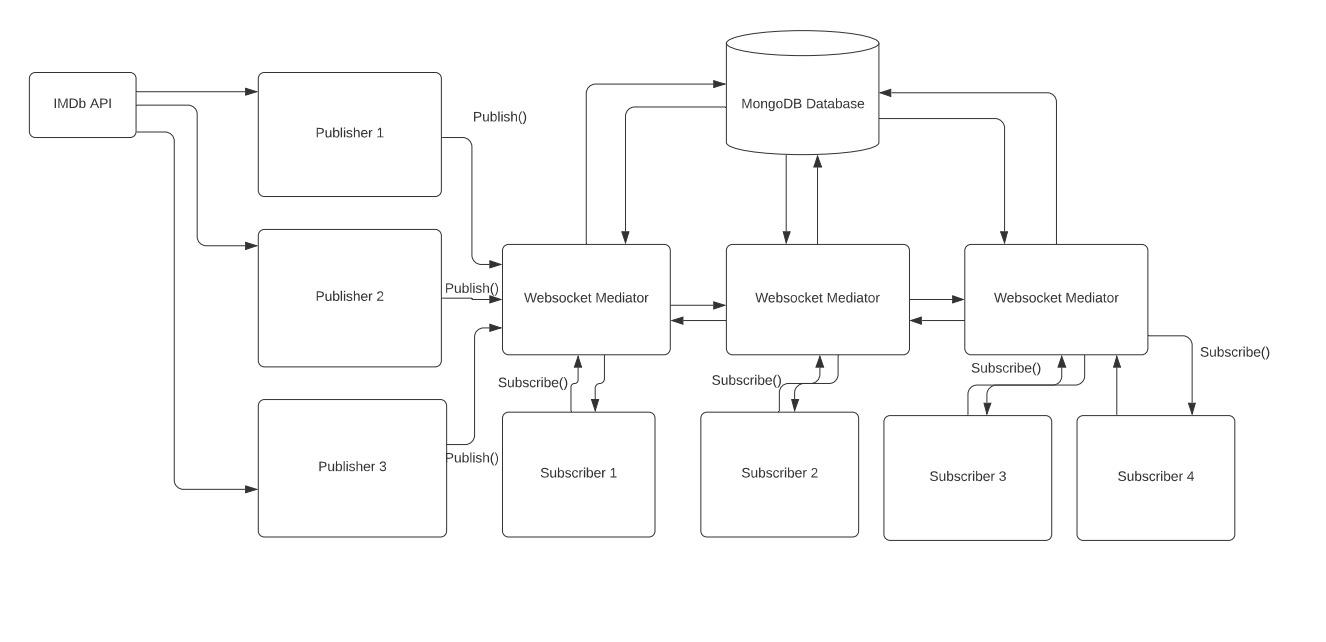
**Authors: -** Tanush Tripathi (50411177) and Rishi Joshi (50365611) (Team No 27)

**Date of Submission: -** 6th November 2021

**Introduction**

Phase 3 of the project deals with the implementation of a distributed version of the Publish Subscribe system. It consists of multiple mediators. A publish-subscribe system is a system where publishers publish structured events to an event service and subscribers’ express interest in particular events through subscriptions which can be arbitrary patterns over the structured events.

**Architectural Diagram**



**Broker Network Diagram**

**Diagram

Description automatically generated**

**Technologies Used**

We have used the following technologies for this phase: -

* Python
* Flask
* MongoDB
* WebSocket
* Docker
* IMDb API

**Implementation**

We have taken the following steps as our implementation approach:-

1. Use the IMDb API for accessing the IMDb movies database
2. Write a python implementation for the publisher that uses the IMDb API to get the data
3. Use MongoDB which is a NoSQL database to store our data
4. Create a collection that stores the subscriber information about the topic that they are subscribed to
5. Implement three WebSocket mediators using python and flask. Similarly implement subscribers.
6. Each subscriber is connected to a mediator with regards to the top of its choice.
7. The mediators are also interconnected via a network of mediators.
8. The subscribers receive the requested updates from the mediators.
9. Docker is used for creating containers.
10. For this project given below are the publishers, mediators, and subscribers :-

* Publishers

1. Publisher 1:- Gets the Hindi(Bollywood) Movies Data from the IMDB API
2. Publisher 2:- Gets the English (Hollywood) Movies Data from the IMDB API
3. Publisher 3:- Gets the Tamil Movies Data from the IMDB API

* Mediators

1. Mediator 1 :- Gets the data corresponding to the the Topics:- Action and Drama Movies
2. Mediator 2 :- Gets the data corresponding to the Topics:- Comedy Movies
3. Mediator 3 :- Gets the data corresponding to the Topics:- Crime Movies

* Subscribers

1. Subscriber 1:- Subscribes to the Topic :- Crime Movies
2. Subscriber 2 :- Subscribes to the Topic :- Comedy Movies
3. Subscriber 3:-Subscribes to the Topic :- Action Movies
4. Subscriber 4:-Subscribes to the Topic :- Drama Movies

**How to Deploy**

The following steps are used to deploy the application

Deploying Publisher:-

Through Command Prompt, go to the Project Folder “27\_50411177\_p13->SourceCode ->Publisher -> Publisher1

1. Run the following command to build the publisher 1 docker container

docker build .

1. Run the following command to create a docker image

docker-compose build

1. Run the following command to create a docker image

docker-compose up

1. Repeat the Steps 2, 3 and 4 for publishers 2 and 3 as well.
2. The data from the web api is then inserted into the mongodb database after all images for publishers are created.

Deploying Subscriber and Mediator: -

1. Open two Command Prompt Side by Side
2. In the First Command Prompt terminal, go to the Project Folder “27\_50411177\_p13” SourceCode ->Mediator
3. In the Second Command Prompt terminal, go to the Project Folder “27\_50411177\_p12” -> SourceCode ->Subscriber
4. In the First Command Prompt Terminal, go to the Project Folder “27\_50411177\_p13” -> SourceCode ->Mediator -> Mediator1 and run the below command:-

docker build .

1. Repeat the Step4 for mediators 2 and 3 as well.
2. In the First Command Prompt terminal, go to the Project Folder “27\_50411177\_p13” -> SourceCode ->Mediator and run the below commands in the same order: -

docker-compose build

docker-compose up

1. In the Second Command Prompt terminal, go to the Project Folder “27\_50411177\_p13” -> SourceCode ->Subscriber and run the below commands in the same order: -

docker build .

docker-compose build

docker-compose up

1. After executing docker-compose up on both terminals, one can observe flask servers started at the client and the server terminal. The mediator flask application is hosted on the localhost:8081
2. Open the URL localhost:8081 on the browser and then click on the **Subscribe** Button below the movie topic you want to subscribe to.
3. After clicking go to the Mediator Terminal and you can see the latest notifications regarding the details of the movies and their respective ratings corresponding to the topic subscribed by the user (The mediators are internally connected to each other and once a request is received it displays the data after looking in all the three mediators)
4. To again subscribe to a topic, repeat the steps 6-10, you can then view the movie information corresponding to that topic.

**Project Contribution**

**Rishi Joshi :-**50 %

**Tanush Tripathi** :- 50%