Phase 7 - Revision

**Grupo 2:**

Diogo Lança – fc53495

Meuri Canhanga – fc42559

Miguel Silva – fc58974

Pedro Santos – fc46417

Renato Ribeiro – fc57433

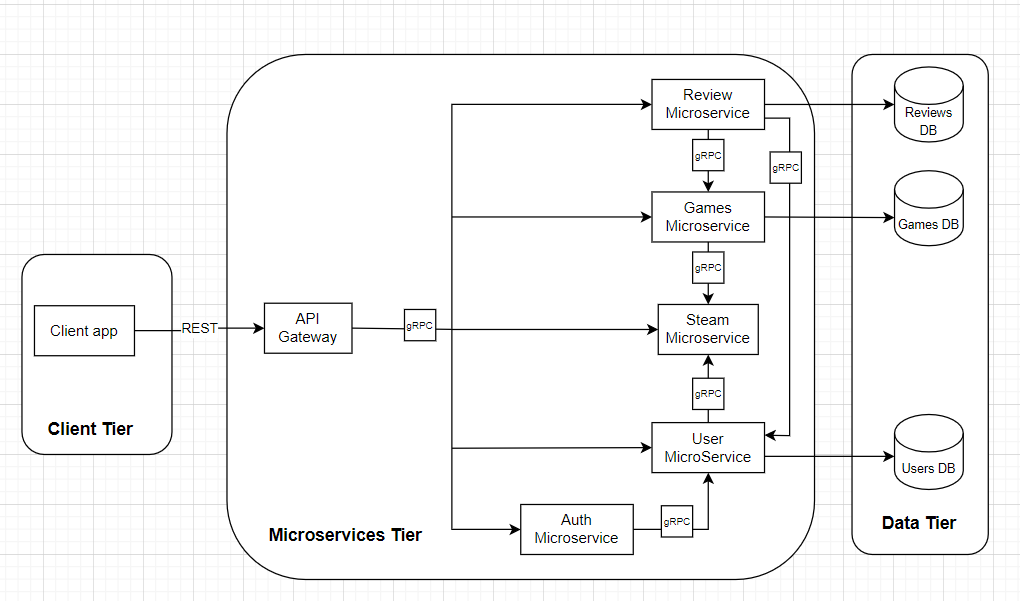
1. Phase 1 summary

For the project, we selected a subset of a **dataset** of around 21 million **user reviews** in multiple languages of around 300 different games on Steam (video game digital distribution service). This dataset has **1,07 GB** worth of user reviews, **last updated on 25 January of 2021**. Worth noting that the dataset was downloaded from Kaggle[[1]](#footnote-1) in a **CSV** **format**.

1. Use Cases

|  |  |  |
| --- | --- | --- |
| **Services** | **Role** | **Functionalities** |
| Normal | Anyone | User sign in |
| User login |
| Suggest games on the main page based on recommendations |
| Search for a game |
| User | Review a game |
| Edit a review |
| Set a review of another user has helpful |
| Delete Review |
| Filter Reviews |
| See all Reviews |
| Set a game has recommended/not recommended |
| Delete account |
| See the total number of games in user possession |
| See his user data |
| User logout |
| Admin | Add games |
| Remove games |
| Remove Reviews (case of wrong/ inappropriate content) |
| Remove users (case they don’t comply with the laws reserved to the service) |

In the current implementation the roles are still not being applied, this means that anyone can use these operations. Nonetheless, we think it’s important to show our vision for the project.

1. Architectural design

On our architecture the client will use the operations with the swagger User Interface (UI). Like mentioned before, the roles on the use cases are still not being applied so anyone using our current implementation can “try out” all the operations.

The client will access our project by a public IP address on a browser – for example ‘[35.233.14.122/ui/](http://35.233.14.122/ui/)’. The IP is obtained on the Google Cloud Platform (GCP) after deployment. On this moment forward the client can “try out” all our implemented operations.

Since leaving a machine deployed uses resources on the GCP, so it’s possible that now the machine is down, and the IP address won’t work. So, if by any chance the client has the source code, it’s possible to access it locally – on the bash 🡪 go to the script’s directory 🡪 and run the build\_dockers.sh file (not forgetting to have the docker running). Then on a browser the client can access the project with ‘localhost:5000/ui/’ and “try out” the operations.

Our API gateway communicates with the microservices through grpc (reminding that the authorize microservice is still not implemented), and since the architecture might be confusing, the communications are as follows:

* The reviews microservice communicates with games and users microservices due to the ‘games/{id}/reviews’ and ‘/users/{user\_id}/games/{app\_id}/reviews’ operations respectively. The rest of the review’s operations are passing directly through the reviews microservice.
* Steam microservice communicates with games and users microservices to suggest the recommended games and active users.
* The other operations (shown on the swagger UI) are being called directly from their respective microservices.

1. Deployment Plan

On the current deployment we use our resources on the GCP and on a Kubernetes machine we deploy our project and give the public IP to let the client try our implementation. But this is only temporary, where the plan/goal is to let the client run a shell file and being able to create a Kubernetes machine and deploy our project being able to access it anytime.

1. Dataset source: <https://www.kaggle.com/najzeko/steam-reviews-2021> [↑](#footnote-ref-1)