Project - Phase 2 Report

Group 2
André Santos `fc62754`
João Mangalhães `fc62546`
Luís Viana `fc62516`
Guilherme Santos `fc62533`
Afonso Santos `fc56900`

March 18, 2024

1 Use cases

We have 4 types of roles: an Admin and Artists which are logged-in users with special permissions, a User, which is a logged-in user, and a not logged-in user that we call Any. The roles defined follow a hierarchical pattern in the sense that for example, an Admin has access to all roles' functionalities and a User has access to all the Any´s role functionalities.

Role	Functionalities
Any	User Log in
	User Sign in
	Search for Tracks/Artists/Genres/Releases
User	Create Playlist
	Set Song as Liked/Disliked
	Delete Account
	See Recommendations
Artist	Add Releases/Tracks
Admin	Remove Releases/Tracks
	Ban User

Table 1: Use cases

In summary, the following list containing the defined API endpoints during Phase 2 shows which operations require authentication:

Endpoint	Requires Authentication
GET /artists/:artistId	No
GET /genres	No
GET /genres/:genreId	No
GET /tracks/:trackId	No
POST /tracks	Yes (Artist and above)
DELETE /tracks/:trackId	Yes (Admin)
GET /releases/:releaseId	No
POST /releases	Yes (Artist and above)
DELETE /releases/:releaseId	Yes (Admin)

Table 2: Functionalities for different types of users

2 Architecture

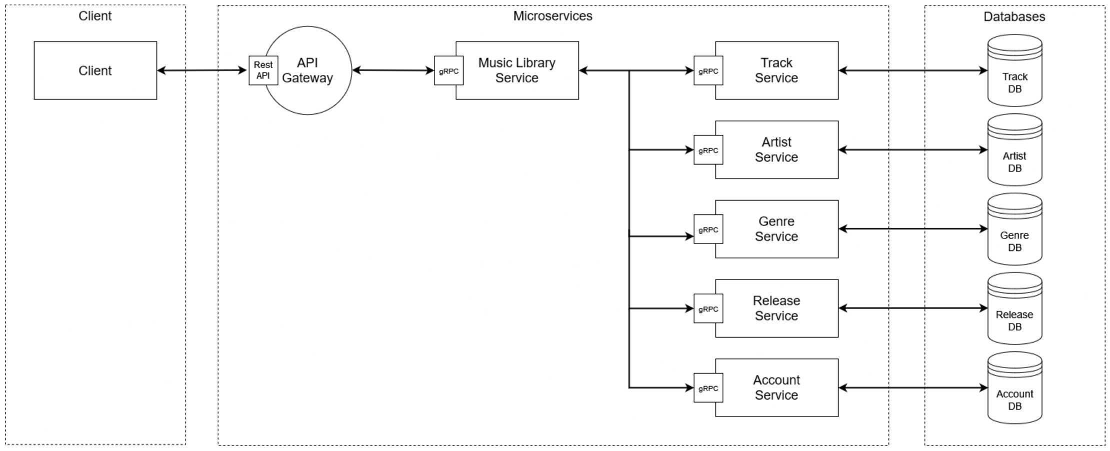

Figure 1: Application Architecture Diagram

We defined the following components (services and databases) that make up the application architecture:

- Track Service - Responsible for providing all operations related to Track´s data.This service also communicates with the Track DB to maintain consistency of said data.
- Artist Service - Responsible for providing all operations related to Artist´s data.This service also communicates with the Artist DB to maintain consistency of said data.
- Genre Service - Responsible for providing all operations related to Genre´s data.This service also communicates with the Genre DB to maintain consistency of said data.
- Release Service - Responsible for providing all operations related to Release´s data.This service also communicates with the Release DB to maintain consistency of said data.
- Profile Service - Responsible for providing all operations related to User Profile´s data such as recommendations based on user's profile. This data includes favourite tracks, user's playlists and liked/disliked tracks. This service also communicates with the Profile DB to maintain consistency of said data.
- Music Library Service - This service is responsible for supporting all functionalities provided by the system. This support requires coordination between all other micro-services since some functionalities may require the

execution of different operations provided by different micro-services. In this sense, this micro-service communicates with all other micro-services using gRPC to handle said coordination. Additionally, this micro-service communicates with an external authentication system to handle functionalities access control as well as profile creation through the Profile micro-service.

- Track DB - Relational database responsible for storing the system´s Track´s data.
- Artist DB - Relational database r for storing the system´s Artist´s data.
- Genre DB - Relational database r for storing the system´s Genre´s data.
- Release DB - Relational database r for storing the system´s Release´s data.
- Profile DB - Relational database r for storing the system´s Profile´s data.

The communication between a client application and the REST API uses HTTP while the communication between the REST API and the Music Library micro-service.