

Architecture proposal for project MUXar

This document is a summary of the proposed software architecture for the *MUXar* project. For more details about the project, consult the *docs* folder in the project's *Github* repository and the development blog.

The project repository can be found here : https://github.com/wadeproject/muxar_wade

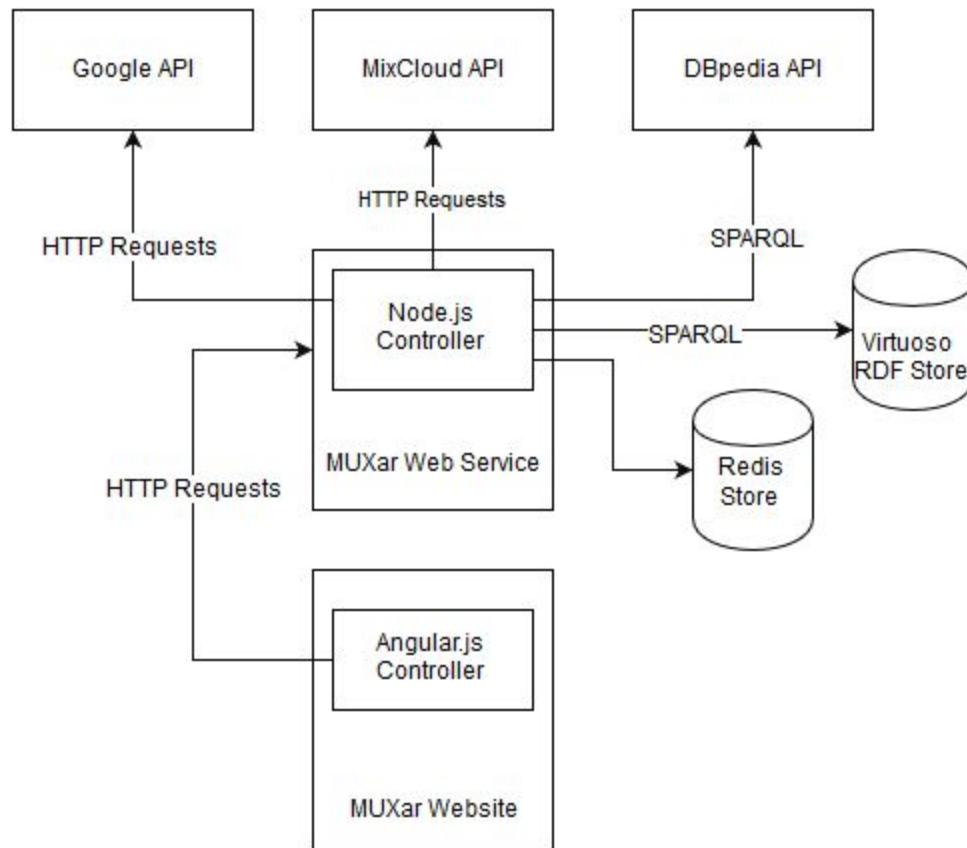
The development blog can be found here : <http://muxar.blogspot.it/>

Architecture Components

As described in the project's Technical Report, there are five main components :

- The web server which consists of a **Node.js controller** which is the main hub of the application. It's role is to serve content to the end-user, communicate with external APIs and internal data stores.
- The client component which is an **Angular.js controller** that runs on the client's side. This controller is served with the webpage that the user requests from the web server.
- The **external services** with which the main controller communicates via different protocols (HTTP, SPARQL). They are used to authenticate users and gather data.
- The **Redis store** which is an internal server where the main controller stores session keys for authenticated users and other ephemeral data.
- The **Virtuoso store** which will hold information in an RDF format and will communicate with the main controller.

These relationships are represented in the following components diagram :



The Back-end

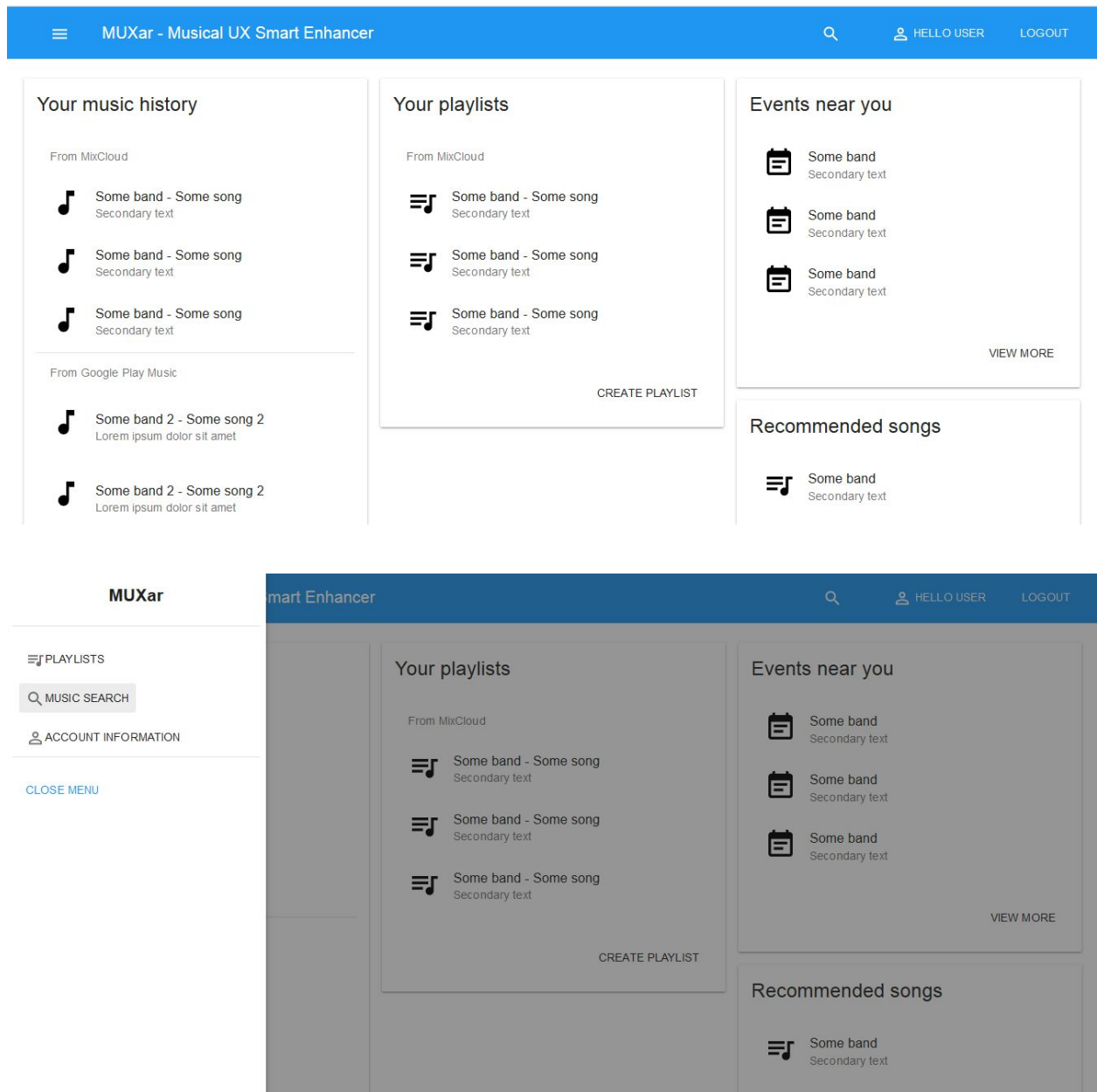
The main controller will have several API endpoints that the front-end controller will send requests to. They are documented using *swagger* and can be viewed using the *swagger.yaml* file available in the *docs* folder of the project's repository.

The development of the main controller will be based on several existing modules for *Node.js*. Some provide functionality for user authentication, others provide means of communication over the *SPARQL* protocol. A complete list of dependencies will be available in the *package.json* file placed in the root of the project's repository. This file is used by *NPM* to deploy the necessary dependencies.

The Front-end

The web pages served to the end-user will contain an *Angular.js* controller which will retrieve data from the main controller's API endpoints. The webpages server are formatted using the *Angular Material* library.

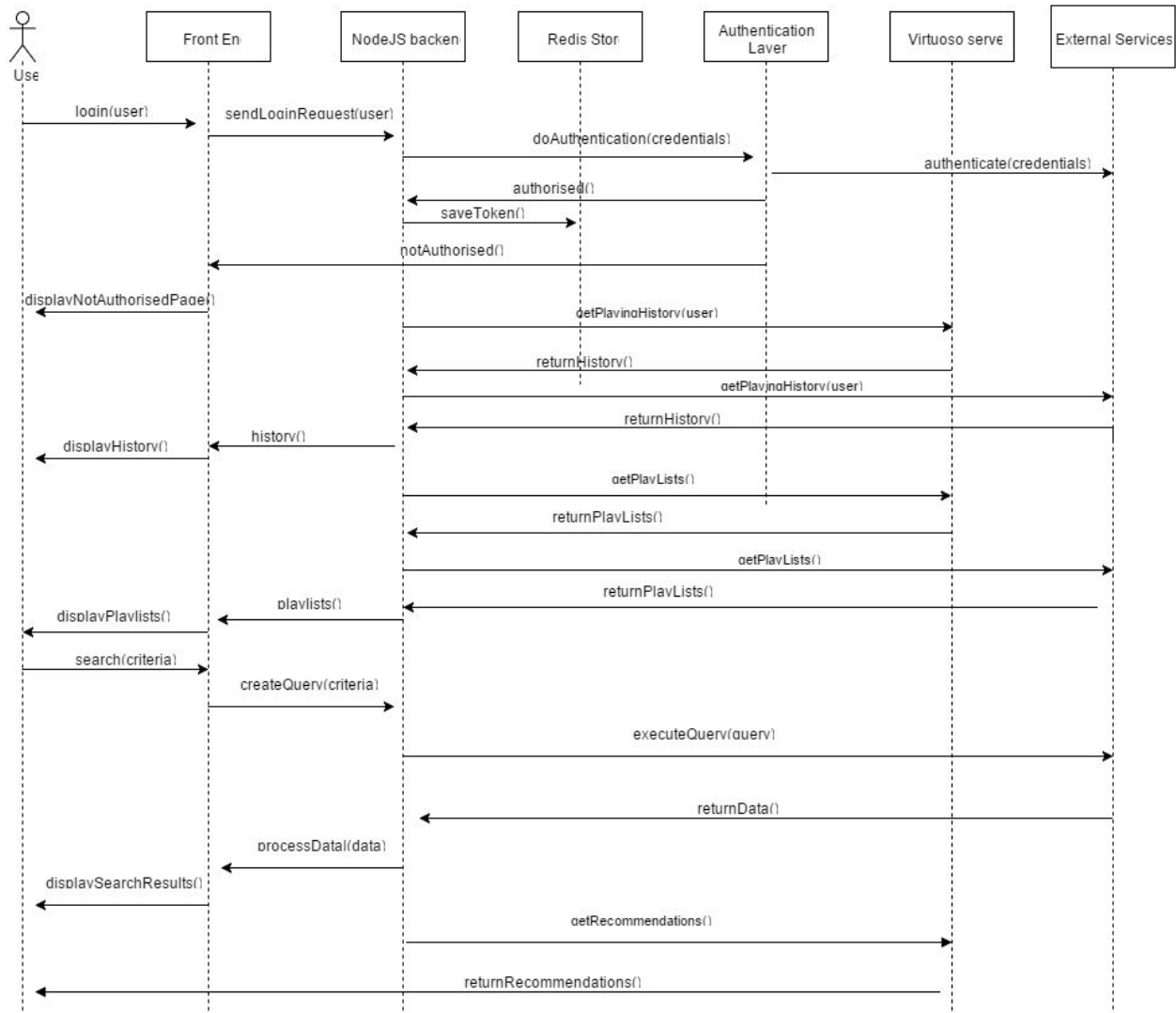
Here are a few images representing preliminary mock-ups of the user interface :



An html version of the mock-ups represented in the above images can be found in the *docs* folder of the project's repository.

Communication

The operations that reflect how the front-end and back-end components communicate with themselves and the rest of the architecture is reflected in the following sequence diagram :



In order to support these operations, the entire application as a system goes through the following states :

