Architecture Diagram

Collaborative Music Player

Web Client

- User Registration
- User Login

- Song Playback
- Song Progress
- Queue Interaction

Logic Server

- User Authentication
- User Interaction

- Song Uploading
- Parallel Playback
- Credit Processing

Web Server

- Access Database

Database

- User Account Details
- User Connections

- Song Queue

- Room List
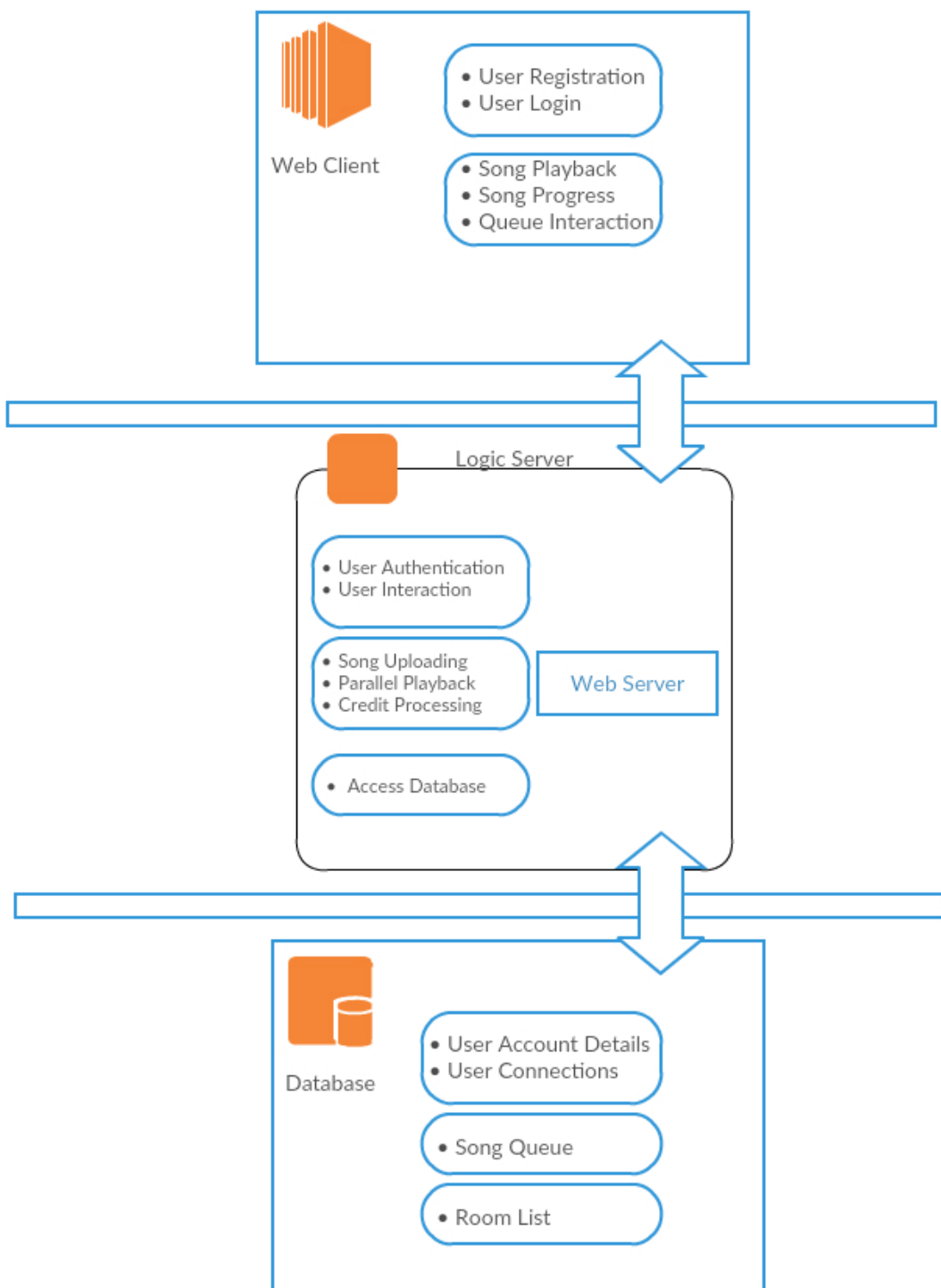
Overview

The Model-View-Controller (MVC) design pattern assigns objects in an application one of three roles: model, view, or controller. The pattern defines not only the roles objects play in the application, it defines the way objects communicate with each other. Each of the three types of objects is separated from the others by abstract boundaries and communicates with objects of the other types across those boundaries. The collection of objects of a certain MVC type in an application is sometimes referred to as a layer.

In our case the view client is the web interface that users interact with. The logic server handles the various process and communications between the view and the controller which is the database. The controller object (the database) acts as an intermediate between the web client and the logic server. This relationship is therefore well defined by a 2 layer architecture and more specifically the MVC design pattern.

Our System Architecture follows two main Architecture Styles. The 2 Layer Client-Server and the Model View Controller. The Diagram above shows three main layers that each communicate with one other layer. This design shows a Web Client that connects to a Logic Server which interfaces with a Database. These components are the View, Controller and Model respectively. Because the Client is hosted as a Web Client and the Logic Server and Database are accessed through that web client we have a clear Client Server relationship.