Architecture Diagram of the Real-Time Trivia Platform

Santiago Suaza Montalvo, David Mateo Ballesteros Ballesteros,
Juan Sebastián Gordillo Medina, Santiago Bejarano Ariza
Department of Systems Engineering
Universidad Nacional de Colombia
{ssuazam, dmballesterosb, jsgordillom, sbejaranoa}@unal.edu.co

I. Architecture Diagram

A. Overview

The architecture is divided into four main layers:

- **Frontend (Presentation Layer):** Implemented in *Angular*, it provides the user interface for players and hosts. It connects to the backend using HTTP requests and maintains a WebSocket connection through *Socket.IO* for real-time communication.
- **Backend (Application Layer):** Built with *Node.js* and *Express*, it handles all business logic, session management, and data exchange with the database and external APIs. It includes a *JWT authentication module* for secure user login and validation.
- **Database (Data Layer):** The *MySQL* database stores user information, session data, and the global leaderboard.
- **External API Integration:** The backend uses the *Open Trivia DB* API to fetch question sets in JSON format.

B. Communication Flow

The frontend sends login and registration requests to the backend, which verifies credentials in the MySQL database. After successful authentication, a JSON Web Token (JWT) is issued for secure future requests. A WebSocket channel allows real-time bidirectional communication for sending questions, receiving answers, and updating the leaderboard. The backend retrieves questions from the external API and saves session results to the database.

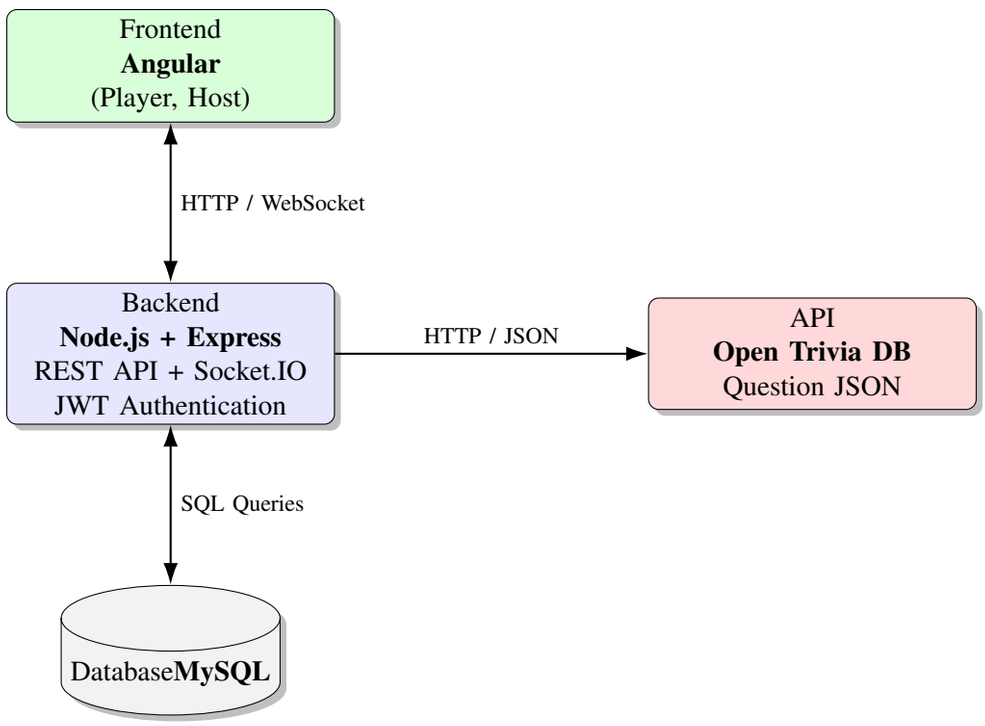

Figure 1. System architecture for the real-time trivia platform, showing bidirectional communication, authentication flow, and external API integration.