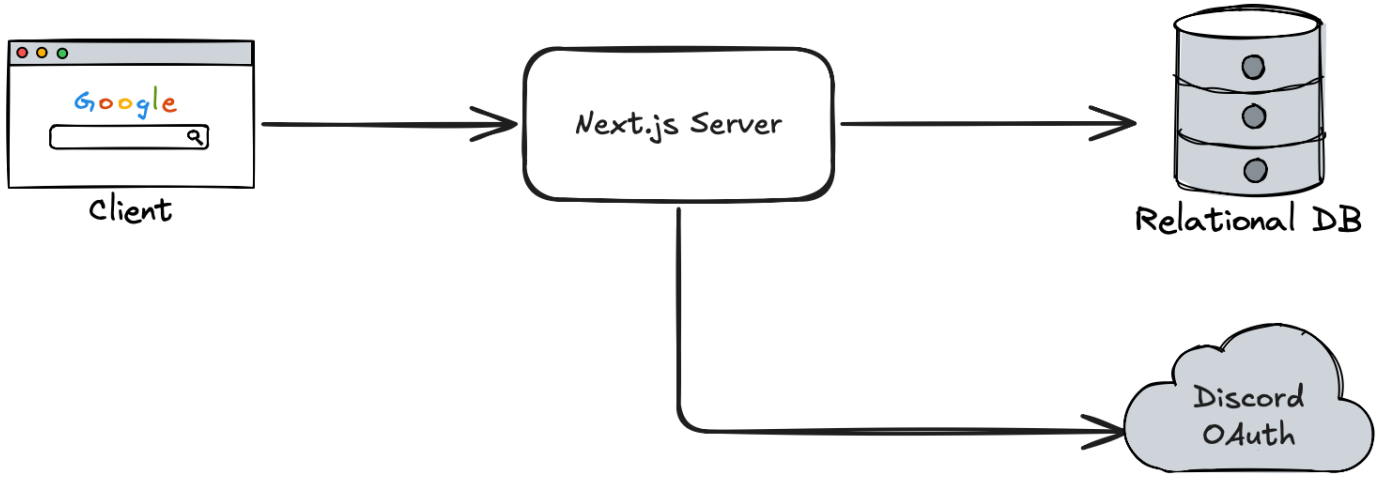
System Architecture

> *NOTE: this is a living document and may not always be up to date.*

Documentation created with AI support.

System Overview



The scouting site is currently comprised of the following components:

- A Next.js Server
- Relational Database (PostgreSQL)

Component Overviews

Next.js Server

Framework documentation: https://nextjs.org/docs

Next.js is a full-stack React framework that enables server-side rendering and seamless client-server communication. The Next.js server acts as the central hub for handling client requests. It authenticates users via Discord OAuth, interacts with the PostgreSQL database to fetch or store scouting data, and serves the frontend UI to users. This setup ensures that only authorized users can access or submit data, while also maintaining a clean separation between the client, backend logic, and persistent storage.

PostgreSQL Database

PostgreSQL Documentation: https://www.postgresql.org

The relational database in this architecture serves as the persistent storage layer for the scouting site. It stores all essential data such as team information, match statistics, user profiles, and scouting submissions. The Next.js server acts as an intermediary, handling data validation, access control, and business logic before reading from or writing to the database. By using a relational database, the system benefits from structured schemas, efficient querying, and strong data integrity—ensuring that all scouting information is reliable and organized.

DrizzleORM

DrizzleORM Documentation: https://orm.drizzle.team/docs/overview

To interact with the database and manage migrations, we utilize DrizzleORM within the Next.js application (ORM - "Object-relationship manager"). Through Drizzle, we are able to easily manage database migrations, and write type-safe queries in our Next.js server. Drizzle also provides Drizzle Studio, an easy way to view and interact with your data locally.

Other Core Technologies

Docker

Docker Documentation: https://docs.docker.com

Docker is a platform that allows developers to package applications and their dependencies into lightweight, portable containers. These containers run consistently across different environments—whether on a developer’s laptop, a test server, or in production. With Docker, you can define your app’s environment in a Dockerfile, ensuring that everything from the operating system to libraries and runtime is standardized.

Our services are deployed using Docker Compose. We currently publish a Docker image of the Next.js server to the GitHub Container Registry. In development, we encourage using Docker to work with Postgres. This reduces "works on my machine" issues, and more closely simulates the production environment. The repository comes with scripts to properly start up a Docker container running Postgres locally.

Revision #5
Created 25 April 2025 07:59:41 by Drew Beamer

Updated 25 April 2025 08:53:35 by Drew Beamer