Contents

[I. History 2](#_Toc128121886)

[II. Introduction 3](#_Toc128121887)

[III. High-Level Design 4](#_Toc128121888)

[1. Component Model 4](#_Toc128121889)

[2. Deployment Model 5](#_Toc128121890)

[IV. Detailed Design 5](#_Toc128121891)

[1. Search Matches 5](#_Toc128121892)

[2. Get Calendar 8](#_Toc128121893)

[V. Database Design 10](#_Toc128121894)

# History

|  |  |  |  |
| --- | --- | --- | --- |
| # | Date | Author | Description |
| 1 | 2023/02/23 | Bach | Init document |

# Introduction

FixtureService is a backend system designed to support the frontend web and mobile applications. The application provides the main following functionality:

* View a list of fixtures for a given tournament.
* Calendar view of fixtures with dates that have matches.

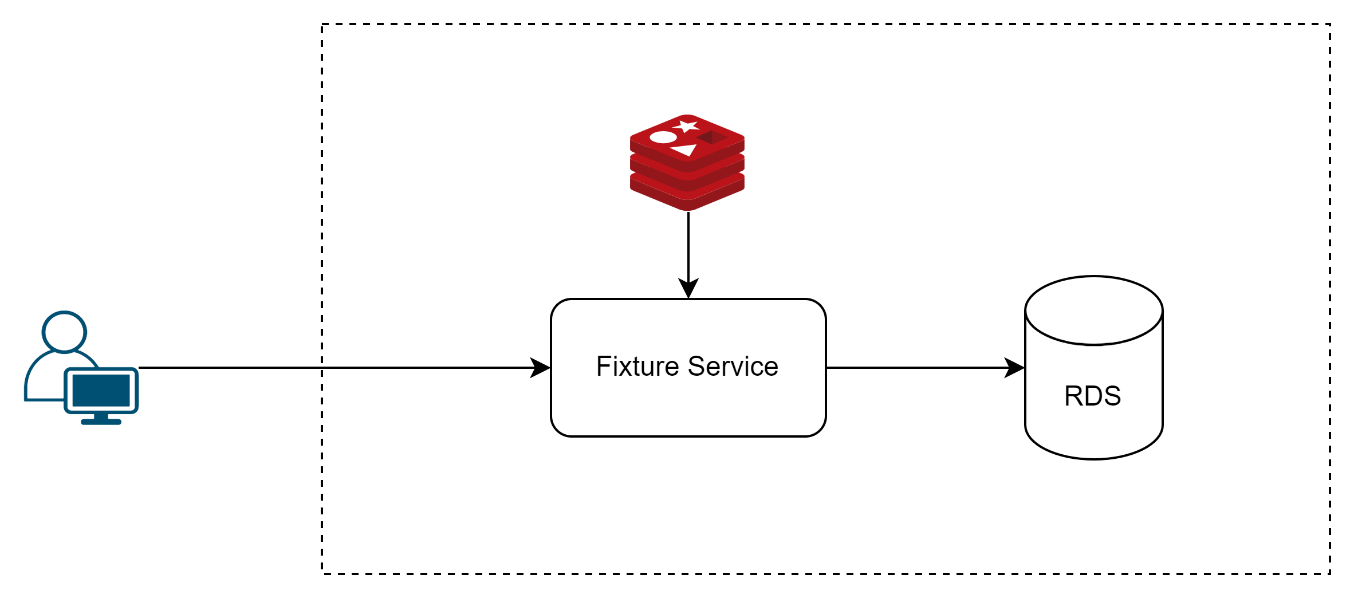
The design needs to consider factors such as:

* Security
* High Availability
* Scalability
* Flexibility & Easy to use

We are not focusing on authentication and authorization as they are out of scope.

# High-Level Design

## Component Model



**Fixture Service**

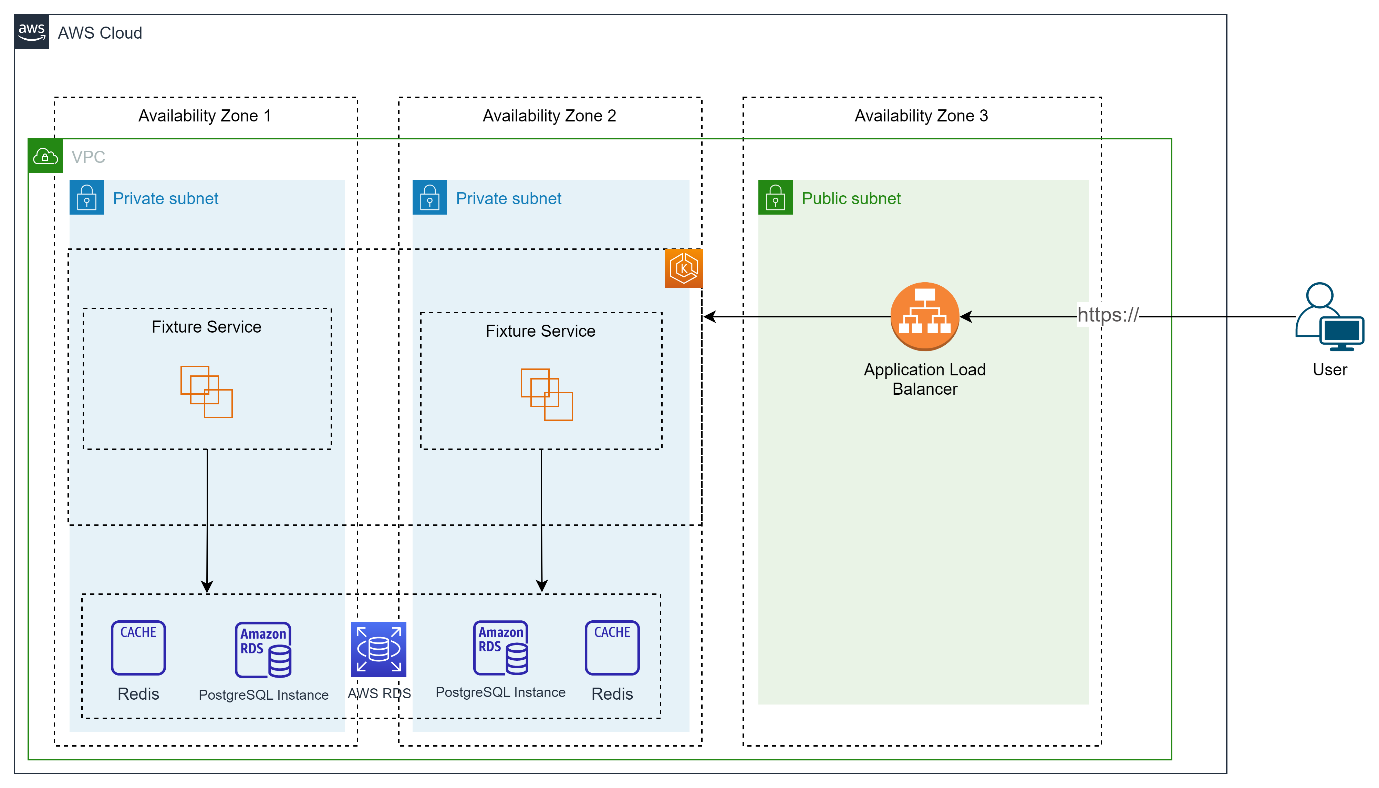
This service is responsible for handling user requests.

**Database**

This RDS stores the data for Matches, Teams and Tournaments.  
  
**Cache**

The cache can be used to store the results of frequent query to improve the application's performance.

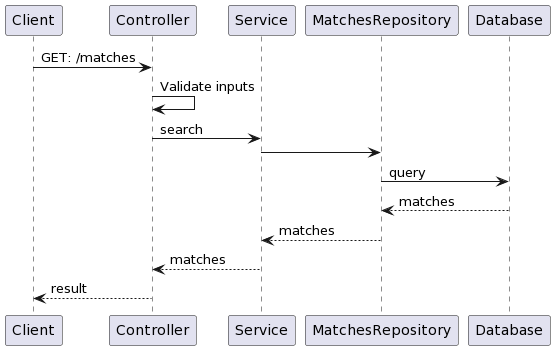
## Deployment Model

  
  
**FixtureService** is designed to be highly scalable and available to ensure it can handle a large number of concurrent users. To achieve this, we have chosen to deploy FixtureService on Amazon Elastic Kubernetes Service (EKS)

**RDS**: For the database, we have chosen Amazon RDS with Aurora as the database engine for scalability and high availability. We also use database proxy.

# Detailed Design

## Search Matches

This API enables users to search for fixtures by specifying various search parameters. The API returns a list of fixtures matching the search criteria, which includes details such as tournament name, home and away team names, match scores, and match date/time. The API supports pagination, allowing users to retrieve fixtures in chunks, and also supports sorting.  
  
Sequence diagram  
  
  
API  
**Endpoints**: GET /matches

**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Required | Description |
| tournamentId | string | TRUE | The ID of the tournament for which to search fixtures. |
| from | Date | TRUE | The start date of the search period |
| to | Date | TRUE | The end date of the search period |
| page | number | FALSE | The page number of the search results to retrieve. Defaults to 1. |
| pageSize | number | FALSE | The number of search results to return per page. Defaults to 20. |
| sortField | string | FALSE | The name of the field to sort the search results by. Defaults to 'startAt'. |
| sortDirection | enum (ASC/DESC) | FALSE | The direction to sort the search results in. Defaults to 'ASC'. |

**Response**

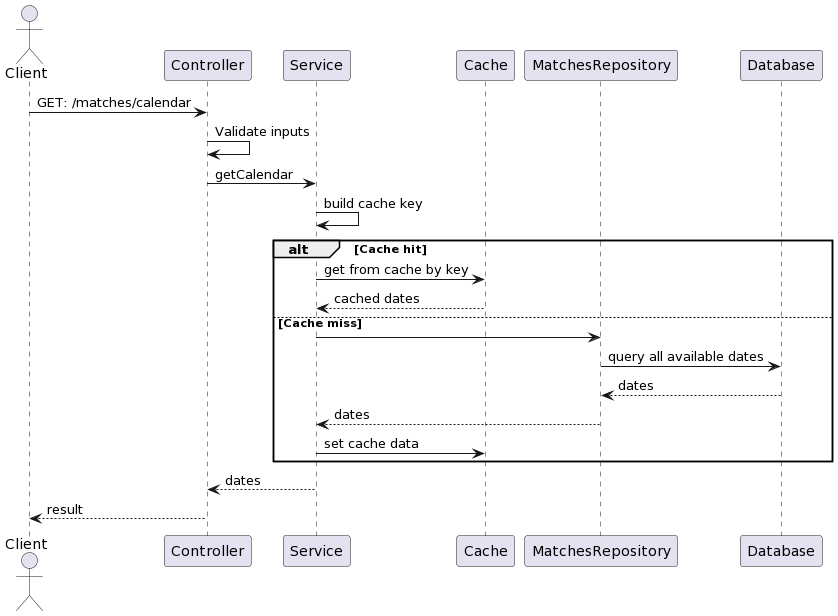
|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Required | Description |
| matches | array | Yes | An array of match objects. See [Match Object] |
| totalMatches | number | Yes | The total number of matches matching the search criteria |
| hasNext | boolean | Yes | A flag indicating if there are more matches available for the current search criteria |

*Match Object*

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Required | Description |
| id | string | Yes | The unique identifier of the match |
| startAt | string | Yes | The start time of the match in ISO 8601 format |
| endAt | string | Yes | The end time of the match in ISO 8601 format |
| homeTeamScore | number | Yes | The score of the home team |
| awayTeamScore | number | Yes | The score of the away team |
| homeTeam | object | Yes | An object representing the home team |
| homeTeam.name | string | Yes | The name of the home team |
| homeTeam.image | string | No | The image URL of the home team |
| awayTeam | object | Yes | An object representing the away team |
| awayTeam.name | string | Yes | The name of the away team |
| awayTeam.image | string | No | The image URL of the away team |

## Get Calendar

This API enables users to view fixtures in a calendar format, where the calendar displays only dates that have matches scheduled. The API returns a list of dates with scheduled matches.

Sequence diagram  
  
  
API  
**Endpoints**: GET: /matches/calendar

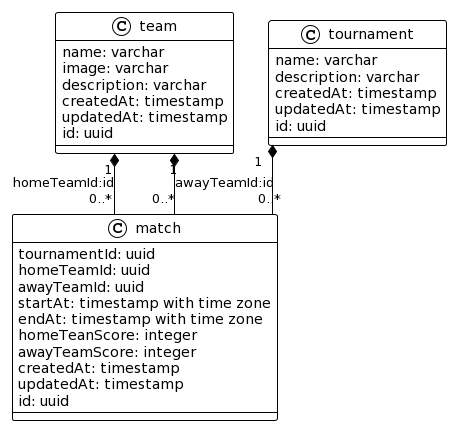
**Parameters**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Required | Description |
| tournamentId | string | Yes | The ID of the tournament for which to retrieve fixtures. |
| from | Date | Yes | The start date of the date range for which to retrieve fixtures. |
| to | Date | Yes | The end date of the date range for which to retrieve fixtures. |
| timezone | string | Yes | The timezone in which to retrieve fixture dates. |

**Response**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Required | Description |
| dates | Array | Yes | List of dates |

# Database Design



1. **Team table**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Constrains | Description |
| id | uuid | Primary Key | Unique identifier for the team |
| name | varchar | Not Null | Name of the team |
| image | varchar | Nullable | URL to the team's logo image |
| description | varchar | Nullable | Short description of the team |
| createdAt | timestamp | Not Null | Timestamp when the team was created |
| updatedAt | timestamp | Not Null | Timestamp when the team was last updated |

**Index**: [id, name, image]

1. **Tournament table**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Constrains | Description |
| id | uuid | Primary Key | The unique identifier of the tournament |
| name | varchar | Not Null | The name of the tournament |
| description | varchar | Nullable | A brief description of the tournament |
| createdAt | timestamp | Not Null | The timestamp when the tournament was created |
| updatedAt | timestamp | Not Null | The timestamp when the tournament was last updated |

**Index**: [id, name]

1. **Match table**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Constrains | Description |
| id | uuid | Primary Key | The unique identifier of the match. |
| tournamentId | uuid | not null | The ID of the tournament the match belongs to. |
| homeTeamId | uuid | not null | The ID of the home team. |
| awayTeamId | uuid | not null | The ID of the away team. |
| startAt | timestamp with time zone | not null | The start time of the match. |
| endAt | timestamp with time zone | Nullable | The end time of the match. |
| homeTeamScore | integer | not null | The score of the home team. |
| awayTeamScore | integer | not null | The score of the away team. |
| createdAt | timestamp | not null | The timestamp when the match was created. |
| updatedAt | timestamp | not null | The timestamp when the match was last updated. |

**Index**: [tournamentId, startAt]