System Design

CSCC01 - CS GANG

Sprint 4

Table of Contents

Software Architecture Diagram
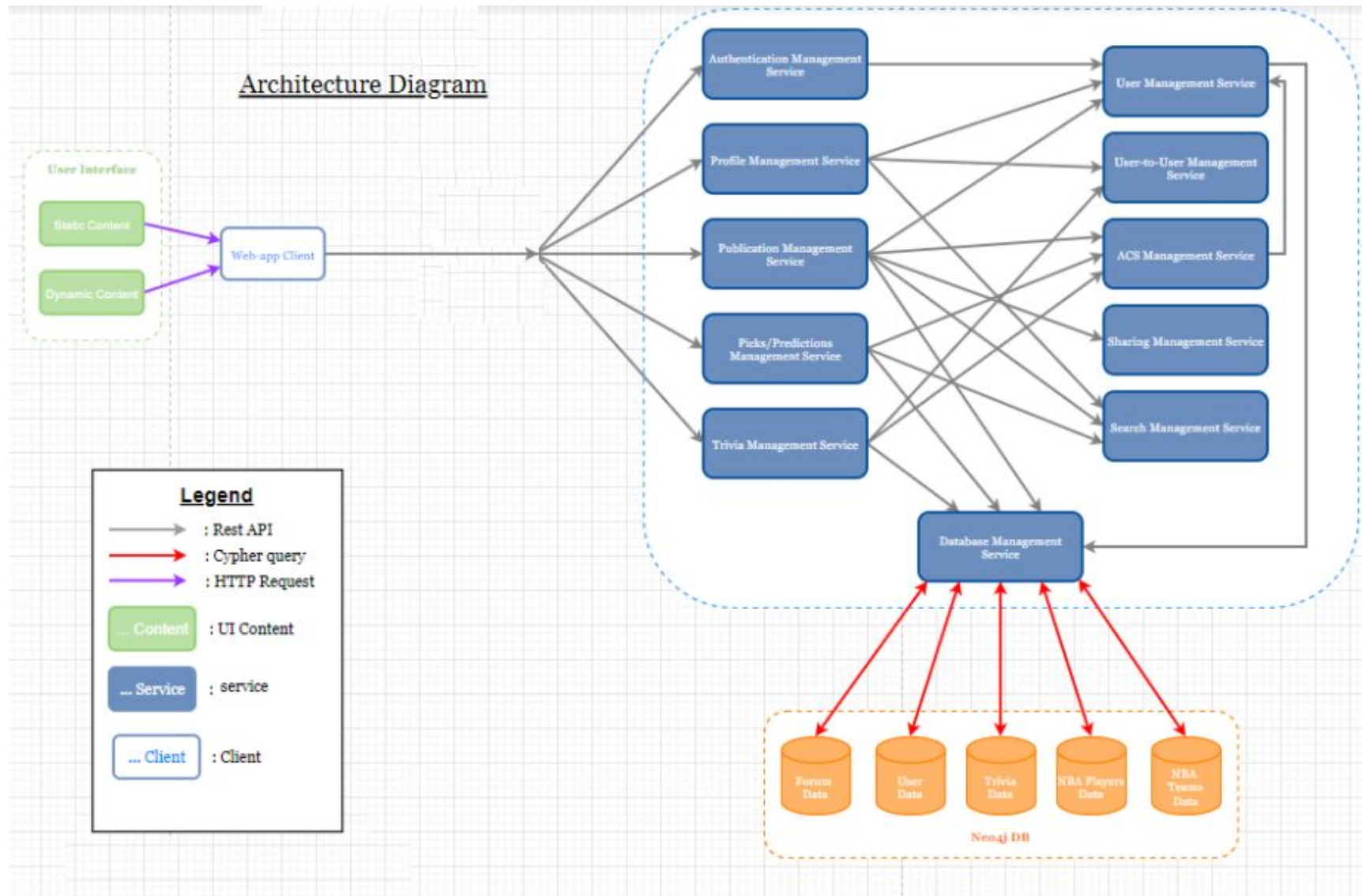
We have changed our architecture from Microservices to a monolithic Client-Server architecture. Doing this significantly lowered the latency on big requests (like fetching posts) and also made it much easier to implement long polling for notifications and multiplayer Trivia. We cannot use the common MVC pattern because in the React framework (which we are using for our frontend) has extremely tightly couple display and display logic, so the view and controller would essentially overlap.

Design patterns used in System Design (cannot be shown in architecture diagram):

Behavioral design pattern:
- Iterator: There will be an iterator interface that will be implemented by the Question, ConsumeRest, and Trivia class to sequentially traverse all the questions in the db to determine which questions the user is supposed to be shown
  - Collaborators: ConsumeRest, Trivia, Main

CRC Cards

Class name: Main
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Communicate with the neo4j database
- Send data to other microservices

Collaborators: Trivia, ConsumeRest, UserManagement, PicksPredictions, Publication, PrivateRoutes

Class name: TriviaQuestion
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Stores question author, answer choices and correct answer

Collaborators: ConsumeRest

Class name: Trivia
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Responsible for binding the backend to the frontend and handling all user events related to single player trivia

Collaborators: ConsumeRest, Question

Class name: DualTrivia
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Responsible for binding the backend to the frontend and handling all user events related to dual player trivia

Collaborators: ConsumeRest, Question, NotifBell, InviteButton

Class name: InviteButton
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Responsible for allowing a user to invite another user to a dual trivia game

Collaborators: ConsumeRest, NotifBell, Profile

Class name: NotifBell
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Responsible for fetching and updating backend processes related to notifications (ie, long-polling) and then displaying accordingly to front-end

Collaborators: DualTrivia, InviteButton, ConsumeRest, PicksPredictions

Class name: ConsumeRest
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Call database microservice to retrieve data from neo4j

Collaborators: Main, ConsumeRest, Trivia

Class name: JWTTokenAuthorizer
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Generate JWT Token given a user's email and password
- Check if a token is still valid (has not expired)

Collaborators: N/A

Class name: Authentication
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Ensure that session is correctly generated based on user's credentials
- Generate a token for user when it logs in
- Create a new user instance when new user signs up

Collaborators: Main, ConsumeRest, SignIn, User, JWTTokenAuthorizer

Class name: Profile
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Handling all the related functionality with updating user information(username, password, about)

Collaborators: Main, User, Search

Class name: PrivateRoutes
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Responsible for checking if user has valid credentials before allowing them access to a webpage

Collaborators: N/A

Class name: ChangePassword
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Responsible for communicating with the neo4j database when user updating the password

Collaborators: Main, User

Class name: ProfileTab
Parent class (if any): N/A

Classname Subclasses (if any): N/A
Responsibilities:
- Handling getting & displaying user's ACS score, uploaded posts and debate history from neo4j database

Collaborators: Main, User, ASC, Post, Debate

Class name: AcsTable
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Handling getting ACS of the user from neo4j database then display the ACS history table filtered by day, month, and year.

Collaborators: Main, User, ACS

Class name: PostaTable
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Handling getting uploaded posts of the user from neo4j then display the posts history table

Collaborators: Main, User

Class name: DebateTable
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Handling getting uploaded debate of the user from neo4j then display the debate history table

Collaborators: Main, User

Class name: Publication
Parent class (if any): N/A
Classname Subclasses (if any): Debate
Responsibilities:
- Responsible for functionality in the debate section
- Regulates interaction between users

Collaborators: Main, ConsumeRest, User, ACS, Sharing, Search

Class name: Carousel
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Display all games for Picks

Collaborators: PicksPredictions

Class name: Game
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Display teams facing each other
- Display date of game
- Allow user to predict winner
- Allow user to amend change

Collaborators: Carousel

Class name: PicksPredictions
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Responsible for allowing users to search for teams and players so that they can make their picks
- Also responsible for showing the results of the predictions

Collaborators: Main, ConsumeRest, Search, ACS

Class name: User
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Will upkeep anything related to the user

- This will be used as an object that can be easily loaded/unloaded

Collaborators: Main, ConsumeRest, ACS, Profile, Publication, Trivia

Class name: ACS
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Main responsibility is to handle all relevant regulations and calculations concerning the ACS score
- Connect with different microservices (primarily the ACS microservice) to update user ACS

Collaborators: User, Publication, PicksPredictions, Trivia

Class name: Sharing
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Responsible for sharing content from the app with third party apps
- Responsible for sharing SportsCred content to other third party apps

Collaborators: Publication

Class name: Search
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Handles all related functionality regarding searching teams, players, and other users

Collaborators: Profile, Publication, PicksPredictions

Class name: UserSearch
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Search users given either first name, last name or email

Collaborators: Search

Class name: PostSearch
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Search posts given content in post (words or hashtags)

Collaborators: Search

Class name: UploadPost
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Responsible to communicate with neo4j database when user uploading new posts

Collaborators: Main, User, Post

Class name: FaceBookEmbeds
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Responsible to integrate with facebook API to get facebook post embedded code by provide the post link

Collaborators: Main, User, Post

Class name: TwitterEmbeds
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Responsible to integrate with Twitter API to get facebook post embedded code by provide the post link

Collaborators: Main, User, Post

Class name: RedditEmbeds
Parent class (if any): N/A
Classname Subclasses (if any): N/A

Responsibilities:
- Responsible to integrate with Reddit API to get facebook post embedded code by provide the post link

Collaborators: Main, User, Post

Class name: InsEmbeds
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Responsible to integrate with Instagram API to get facebook post embedded code by provide the post link

Collaborators: Main, User, Post

Class name: Feed
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Create a feed of posts for an end user to consume
- Collect all posts by author, by time of publication etc.

Collaborators: User, Post, Main, Publication

Class name: PostReaction
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Allows user to interact with Post by liking or disliking

Collaborators: N/A

Class name: Post
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Store data of post (author, content, time of posting, likes, dislikes)

Collaborators: User, PostReaction

Class name: PostComment
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Provide comment column that can be collapsed and expanded by giving a list of JSON comment object

Collaborators: Post

Class name: ShareMenu
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Provide drop-down share button, gives functionality to post posts to other platforms (Facebook, Twitter, Reddit, Tumblr) ticket#: CSGAN-15
- input data = [parameters: url(String), content(String), hashtag(String)]

Collaborators: Post

Class name: OpenCourtPost
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Show detail of the post to users

Collaborators: User, Post

Class name: VisitorOpenCourtPost
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Show the detail of the post to users that don't log in and provide meta-information (provide post info to other platform's API when reposting the SportCred post

Collaborators: User, Post

Class name: SignUp
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Validate if user email already exists in db
- Create user account with user info (firstname, lastname, email, phone, password, birthday)

Collaborators: Main

Class name: SignUpScript
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Ensure that when a new user is added they are given the 52 trivia questions to do

Collaborators: Main, Trivia

Class name: ApiCalls
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- This class connects with the backend and regulates the authentication token for login and sign up

Collaborators: Main, SignUp, Login

Class name: Leaderboards
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- This class connects with the backend to the front end and allows users to view leaderboards within the app

Collaborators: Main, User

Class name: Debate

Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- This class is responsible for creating and rating Debate posts
- Get the day's debate post and ratings for user
- Update all user's ACS and send notifications

Collaborators: Main, User, Notifications

Class name: Analysis
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- This class is responsible for creating and rating Analysis posts
- Update all user's ACS and send notifications as users rate each other debate posts
- Pick 3 random debate posts for a user to rate
- Hide the poster of the debate one is rating until after their rating is submitted

Collaborators: Main, User, Notifications, Debate

Class name: UserPost
Parent class (if any): N/A
Classname Subclasses (if any): N/A
Responsibilities:
- Show/hide poster info
- Send user rating to post

Collaborators: Main, User, Debate