Data-Intensive Systems Project Part 1 Group 2

1. Description of the service/system, requirements (reasons) why it should use distributed databases

What is the YouTube system?

YouTube is one of the recognized platforms for sharing videos. The users can upload and view them, comment about those videos, and share a different range of content. This platform supports different types of media, such as music videos, educational tutorials, and gaming streams. Users can create accounts and subscribe to channels and receive personalized recommendations about their viewing history. In addition, YouTube is user-friendly and is accessible from different devices, such as computers, laptops, smartphones, and smart TVs.

The Reasons for YouTube's Use of Distributed Databases

There are many reasons for YouTube's use of distributed databases, but we have highlighted some of them here:

1. *Scalability*

YouTube manages many volumes of data, because the millions of videos uploaded daily, and they receive billions of views. The design of a distributed database enables integrated growth when more users join to the platform and more content is created. This approach decreases limitations related to the capacity of server and facilitates the management of increasing demand by distributing data between multiple servers.

2. *Performance*

Distributed databases allow YouTube to store copies of videos in different global locations. When a user requests a video, the system directs the request to the nearest server, so it reduces delay and guarantees quicker access. Because it is necessary for user satisfaction to load video fast.

3. *Reliability*

YouTube must be operational all the time, even in the event of server failures or maintenance. A distributed database system guaranties that if one server becomes inactive, the system can redirect traffic to other functional servers automatically. This redundancy significantly allows users to access content without interruption.

4. *Data Management*

YouTube generates and collects a lot of data, such as user behavior, video statistics, and comments. A distributed database system can organize this data efficiently and enabling improved analysis and reporting better.

5. *Global Domain*
YouTube is active on a global scale and attracts many users from different countries. Distributed databases facilitate storage of data for users with improving performance and loading times. YouTube can provide a more responsive experience, without paying attention to geographical locations of users by minimizing the distance that data must travel.

6. *Content Delivery and Caching*
YouTube also uses content delivery networks to facilitate efficient serving of video. Distributed databases work at the same time with content delivery networks (CDNs) to save popular videos near the users, which means the content that has been frequently watched, is rapidly accessible. This strategy reduced the load of primary database and increases overall speed.

a. Descriptions of users (client, provider, admin)

Descriptions of YouTube Users

YouTube has different user **clients, providers and admins** that are determined with different backgrounds, interests, and behaviors. They are:

1. *Normal Viewers as a client:*
Normal viewers are a part of YouTube's audience that have a relationship with the platform separately. These individuals access YouTube without a predetermined decision typically and looking for entertainment or attractive content.

This group is usually interested in more popular content such as fun clips, music videos, and viral challenges.

2. *Content Creators as a provider or admin:*
Content creators include a big range of individuals, from amateurs to professionals who produce, edit and upload videos to the YouTube platform.

Their focus is on making a connection with viewers for maintaining the sense of community about their channels.

3. *Subscribers as a client:*
Subscribers are a part of users that access specific channels to receive notifications about new content releases.

Subscribers are more active in interacting with their favorite channels than regular or normal viewers, participating in discussions and supporting creators with their likes, comments, and shares of the videos.

4. *Researchers and Learners as a client:*
A notable part of YouTube users is looking for educational content and using the platform to access educations, lectures, and informational videos and something like that.

This group prioritizes clarity and information in the videos that they use.

5. *Gamers as a client, admin or provider:*
Gamers are one of the most important subsets of YouTube users that watch gameplay videos, tutorials, and live streams regularly.

This group not only enjoys watching gaming content but also participates in discussions about strategies and games tips. Most of the gamers create their own content and interact with their audiences on different platforms.

6. *Influencers and Brands as an admin:*
Influencers and brands use YouTube as a marketing tool and advertise their products or services and engage with potential customers.

This group also emphasizes viewer engagement and brand presence, and they often collaborate with other creators and use advertising to reach broader audiences.

7. *Mobile Users as a client:*
Many YouTube users access the platform via smartphones or tablets.

Mobile users typically prefer fast and simple videos that show their lifestyles, and they often watch content during dead times or breaks.

8. *International Users as a client:*
YouTube has a global audience that includes users from different countries and cultural backgrounds.

These users may look for content in their native languages or explore international videos, which help enrich the content available on the platform.

9. *Commenters and Engagers as a client:*
A part of users participates in the comments section of videos actively.

These individuals share opinions, ask questions, and participate in discussions, that help a community surrounding specific channels.

So, we conclude that:

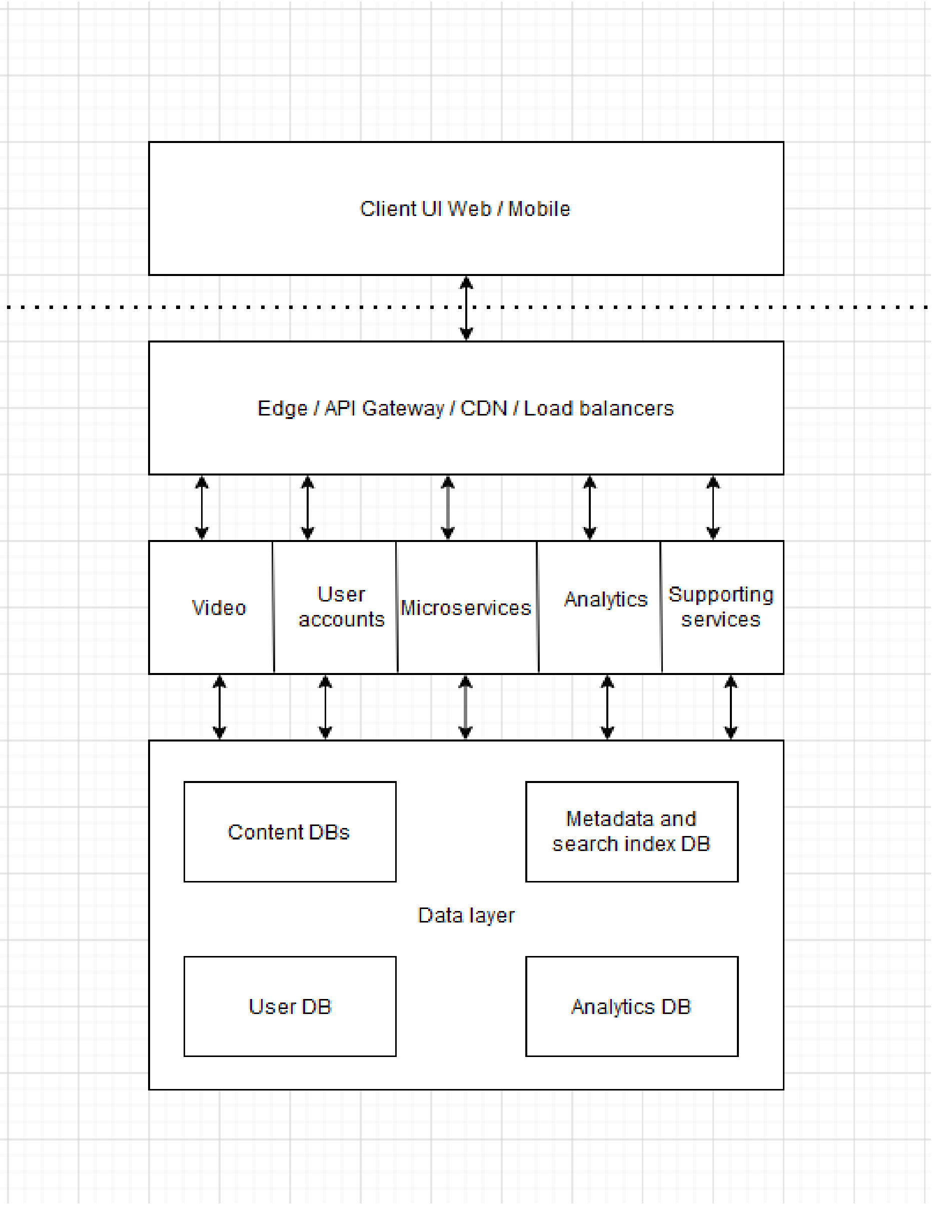
YouTube system is determined by a wide array of interests and interaction styles, that increase the platform's experience. These users’ categories help YouTube to improve its features.

2. Representation of your distributed database system (architecture figure), justifications of choices

Users interact with the platform through a web interface or mobile app. This layer communicates with the backend services using APIs.
The back end uses a microservice architecture, where each unit handles a specific function, such as video uploading, streaming, user management, or content recommendation. CDNs, load balancers and a modular shared-nothing distributed database optimize performance, usability and scalability.

On the data layer, content databases store the streamable media, and it’s required data. Metadata and search index databases are optimized for quick indexing containing data like titles descriptions and categories of the media. The analytics database is intended to improve user experience by storing user behavioral data for recommendations and storing related information to the user. The user account database stores information related to user accounts, such as login credentials, subscriptions, watch history, and preferences.

By distributing data across multiple servers located in different regions, we reduce latency and ensure high availability. In case of server failures, traffic is redirected to another server, maintaining service continuity. Load balancers enhance system performance by evenly distributing incoming requests across servers. This prevents overloading of any single server and ensures smooth operation, even under high traffic conditions (picture below).



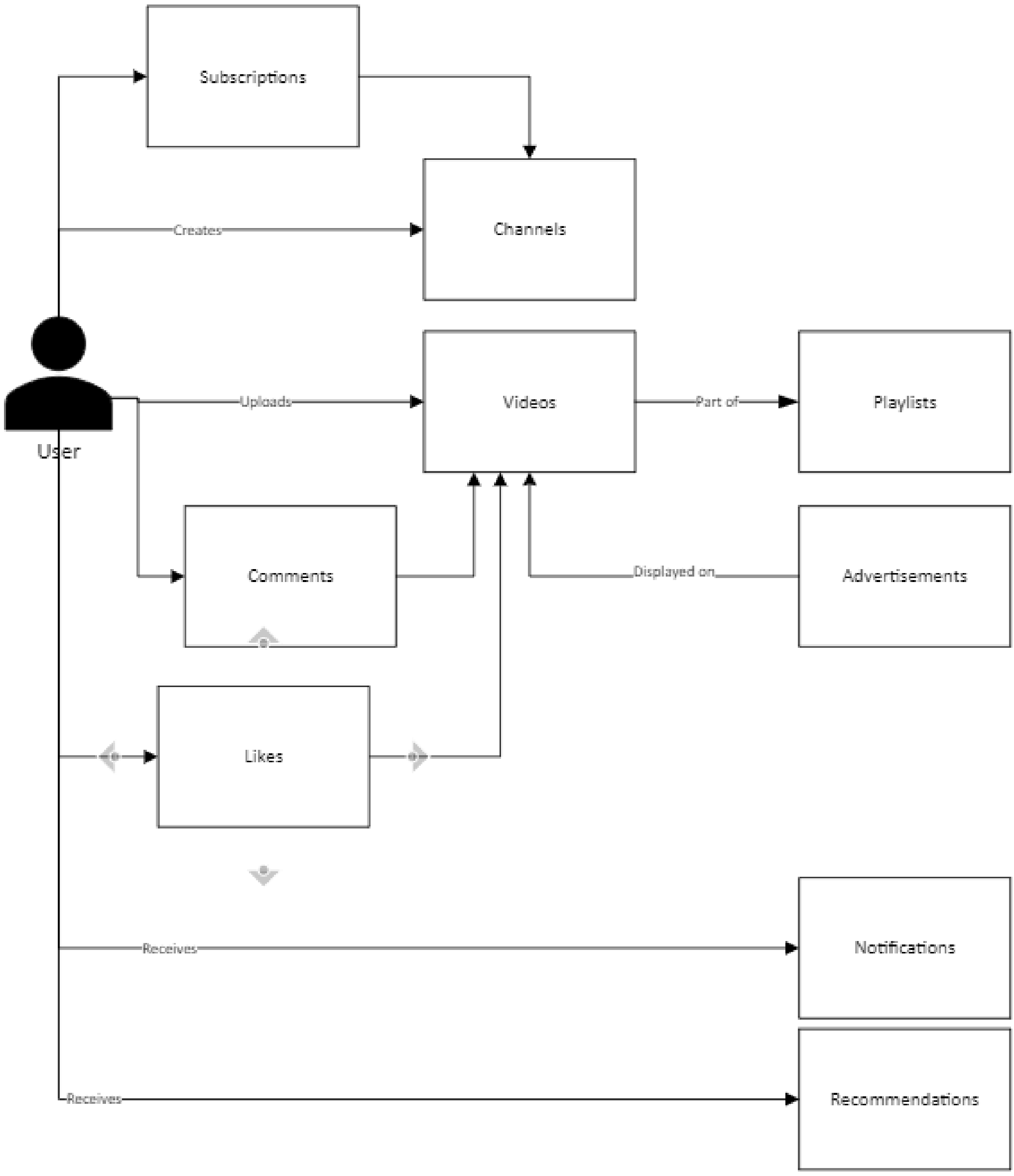
3. Global and local conceptual schemas

Key entities

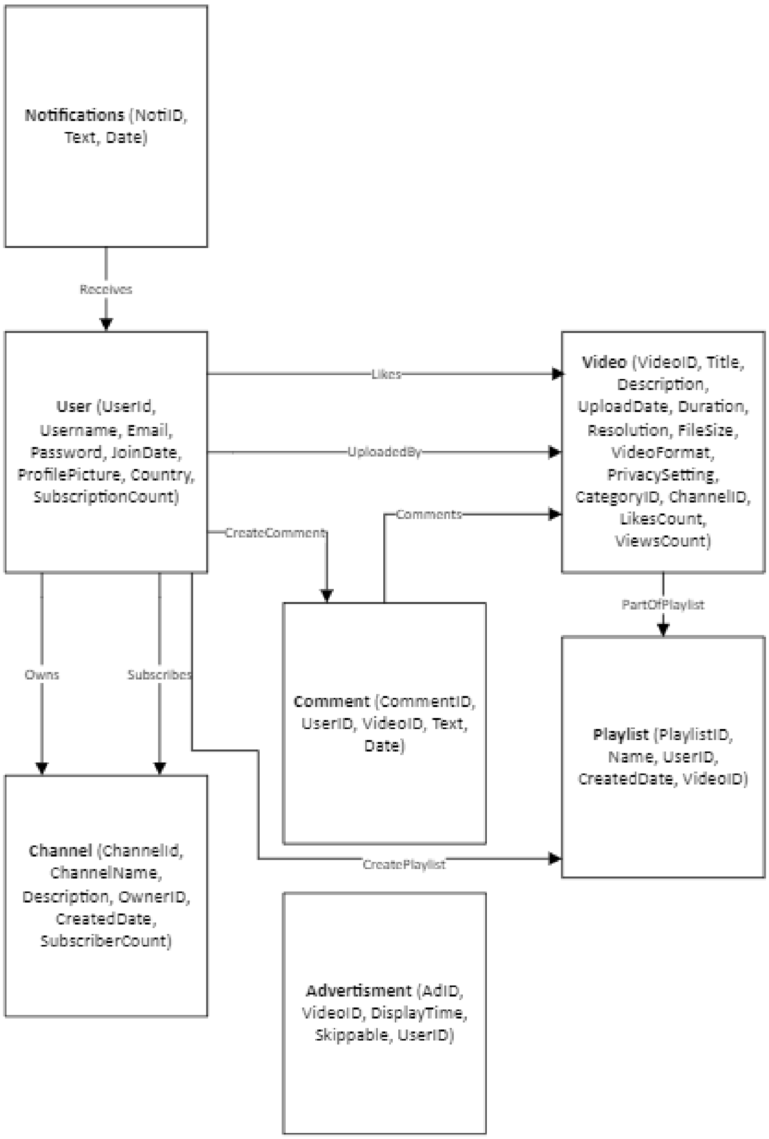
- Users: All users who interact with the system, including viewer, content creators and administrators
- Videos: The video content uploaded to the system
- Channels: Created by users to organize their content
- Playlists: Collection of videos created by users
- Comments: User-generated text related to videos
- Subscriptions: Users subscribe to channels to receive notifications
- Advertisements: Third-party content shown on/before/after videos
- Likes: Users feedback on videos

- Notifications: Alerts sent to user about new videos or activities
- Recommendations: User receives recommendations based on their subscriptions and earlier activity

GCS ER Model:



Local Conceptual Schema


Notifications (NotiID, Text, Date)
Receives
User (UserId, Username, Email, Password, JoinDate, ProfilePicture, Country, SubscriptionCount)
Likes
UploadedBy
Video (VideoID, Title, Description, UploadDate, Duration, Resolution, FileSize, VideoFormat, PrivacySetting, CategoryID, ChannelID, LikesCount, ViewsCount)
Comments
CreateComment
PartOfPlaylist
Owns
Subscribes
Comment (CommentID, UserID, VideoID, Text, Date)
Playlist (PlaylistID, Name, UserID, CreatedDate, VideoID)
Channel (ChannelId, ChannelName, Description, OwnerID, CreatedDate, SubscriberCount)
CreatePlaylist
Advertisment (AdID, VideoID, DisplayTime, Skippable, UserID)

4. Fragmentation (partitioning/sharding) and replication

a. Reasons and degree of fragmentation

As part of YouTube's fragmentation process, data can be distributed using three different sharding methods: user-based, content-based, and geographical sharding. Each method optimizes performance, scalability, and reliability in distinct ways. Here’s a closer look at the reasons and degree of fragmentation for each approach:

1. Geographical Sharding

- **Reason**: Geographical sharding divides data based on users' locations. This reduces latency by delivering content from the nearest data center, which is particularly important for a global platform like YouTube.
- **Degree of Fragmentation**: Sharding can be implemented on a regional level such as North America, Europe, or Asia and can be further divided by country or city if needed. Depending on user distribution, this approach could result in hundreds of shards.

2. User-Based Sharding

- **Reason**: In user-based sharding, data is divided based on user accounts or IDs. This ensures that user-specific data such as uploaded videos, watch history, likes, and comments is stored together. This setup speeds up read and write operations since accessing a specific user's data doesn't involve searching across multiple shards.
- **Degree of Fragmentation**: The number of shards is typically tied to the number of active users, potentially scaling up to thousands of shards to ensure the load is spread evenly. Each shard may hold data for millions of users, ensuring a balanced distribution and high performance.

3. Content-Based Sharding

- **Reason**: Content-based sharding organizes data according to video characteristics, such as categories (e.g. music, gaming, education) or video IDs. This approach helps distribute traffic by grouping data based on content type, making it easier to handle read-heavy operations for certain types of content.
- **Degree of Fragmentation**: Data can be spread across thousands of shards, with each shard dedicated to a specific content category or set of video IDs. This arrangement helps evenly distribute the load, improving the efficiency of queries and access.

b. Data replication strategy, reasons for it

YouTube, as a product, is part of a large ecosystem (Google) that offers a wide variety of features. However, in this assignment, we will narrow our focus to YouTube's core functionality as a video streaming and uploading service. This leads us to choose a replication strategy centred on the key priorities for such platforms:

- **Low Latency**: Ensuring videos load quickly to provide a seamless viewing experience.
- **Data Durability**: Guaranteeing 24/7 access to video content, even in the event of server failures.

Based on the previous points, YouTube requires a replication strategy that balances low latency and data durability. The most suitable approach is asynchronous distributed replication, which efficiently handles the global scale and the massive data volumes associated with video content.

This type of replication protocol is ideal, as it allows video data to be replicated across multiple geographically distributed nodes asynchronously providing the following characteristics for the system:

- Low Latency: Videos are stored in data centers around the world, enabling content delivery from the closest server to the user. This reduces buffering and ensures that videos load quickly.
- High Availability: With asynchronous distributed replication, video content is first uploaded to a regional node and then gradually replicated to other global nodes. Even if a server in one region fails, the content remains accessible from other replicas, ensuring continuous availability.
- Scalability: With the massive amount of video uploads and views daily. Asynchronous replication allows the system to scale efficiently, distributing data across the globe without waiting for synchronous updates. This is critical to managing the vast volume of data without compromising performance.
- Eventual Consistency: Since immediate consistency isn't required for video content as a slight delay in replication is acceptable, asynchronous replication optimizes performance while ensuring eventual consistency across all nodes.

AI use disclaimers:

- ChatGPT used to give tips to help design the database architecture and to improve grammar in 3. section