CLIENT-SERVER ARCHITECTURE FOR YOUTUBE SERVICE

Abstract:

Implementation of the client-server pattern has been proposed to structure the YouTube program. Which is a huge library for requesting and viewing videos

Introduction:

define software architecture as a system structure or structures, which comprise software elements, externally visible properties of those elements, and relationships among them. The software architecture is the basis on which the fundamental guidelines for the system construction are defined.

Capturing the rationale generated in the architectural design allows stakeholders to answer the questions that arise along the software system life, and what assumptions made before decision, alternatives and trade-offs used to select an approach and reject others.

Assumptions:

The YouTube application is based on several main features, and is mainly based on request and reception, as well as needs a very large space as a base that stores videos, and an appropriate presentation method that helps the user to access the videos he needs

When choosing the appropriate pattern, we are making assumptions to take into account these above-mentioned features, meaning that the selected pattern must be suitable for high storage spaces and to provide several services at the same time, and not have a central point of failure so that each service is separate by itself, based on the many services it provides The YouTube . We also expect the presence of huge requests at the same time from several users, so it is assumed that the chosen style is able to schedule requests and receive them in a short time.

Some alternative patterns for our YouTube web application are:

• ***Model View Controller*** *because it is considered as a web-based application that has a component which manages the system's data and a component which manages the data presentation to user and the controller that manages user interaction through the web. It supports YouTube where it needs a pattern that supports presentation of same data in different ways independently as the MVC do.*

• ***Layered****: because for providing content on the platform we have to follow organized steps and move from an interface layer to the core database layer sequentially. It supports maintaining the content where it increases the dependability through layers.*

• ***Event Bus Pattern****: it's a pattern where the publisher sends content to the receivers through the event bus web server and it's suitable for the YouTube where each creator can view his content to the subscribers.*

Trade-offs or criteria used to select an approach and reject others:

The reason for using the **Client-server pattern***: It is one of the most appropriate patterns and the most pattern that shows the way the YouTube application works, in which the client and server applications interact directly with the transport layer protocol, which means the communication process that enables entities to send and receive information. This pattern is effective for describing YouTube because there is Clients and servers have many periodic tasks that they constantly have to perform.*

Reason not to use alternate styles:

The reason not to use **Model View Controller***: is that it requires a lot of periodic updates, and its views may take some time and this causes the speed and quality of the application to decrease.*

The reason for not using **Layered architecture pattern***: is that in the YouTube application () does not apply to all its parts, but rather only a few specific parts of it, and () is difficult to cleanly separate between layers.*

The reason not to use **Event Bus Pattern***: is that the method of this pattern is similar to notifications in that it is only sent in one direction and no interaction is allowed, which means that it describes a very small part of the YouTube app and cannot be used to describe the entire application because YouTube is basically an interactive application.*

Client-server architecture :

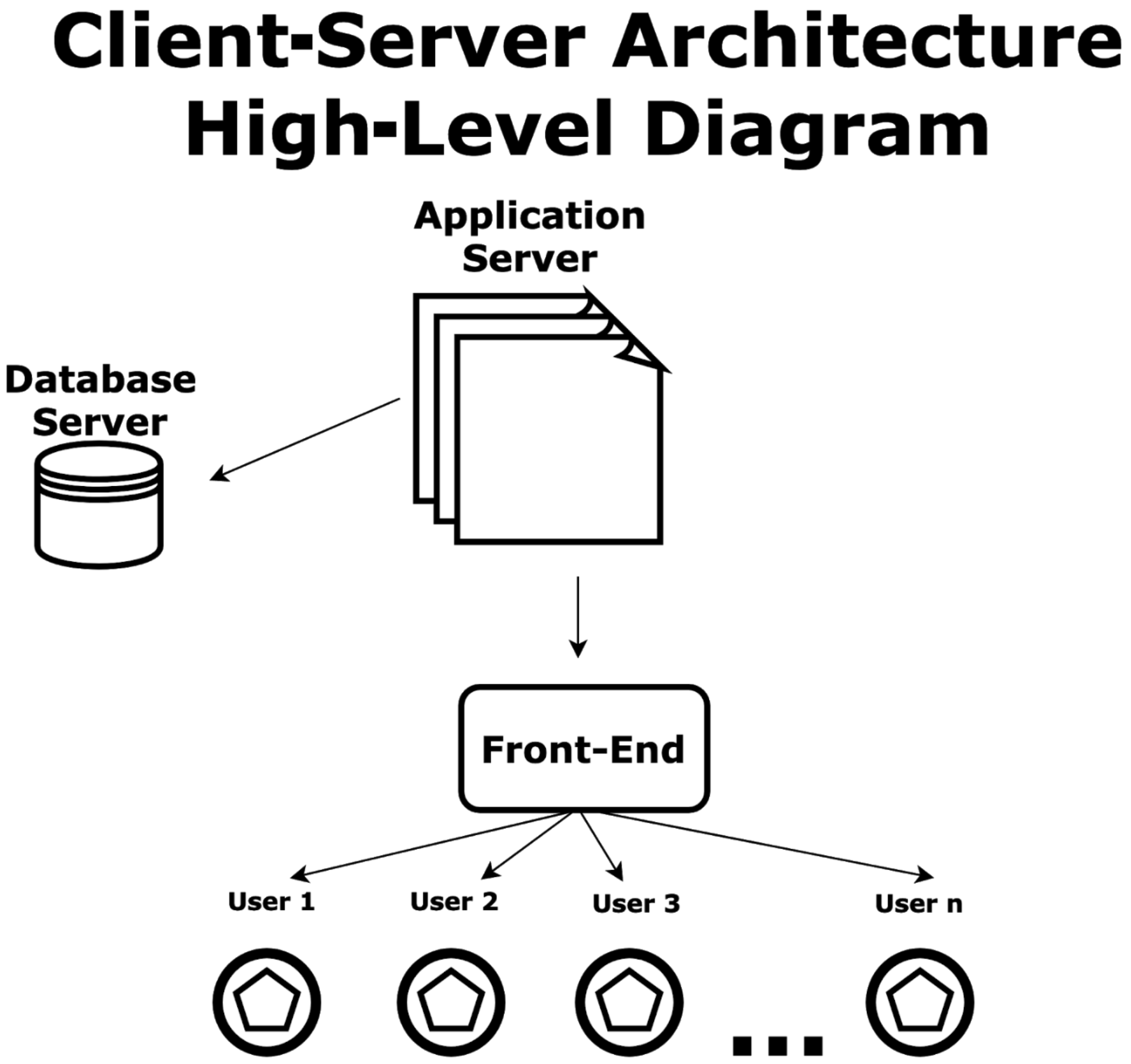

Figure 1

How to apply the client-server architecture pattern for the you-tube app:

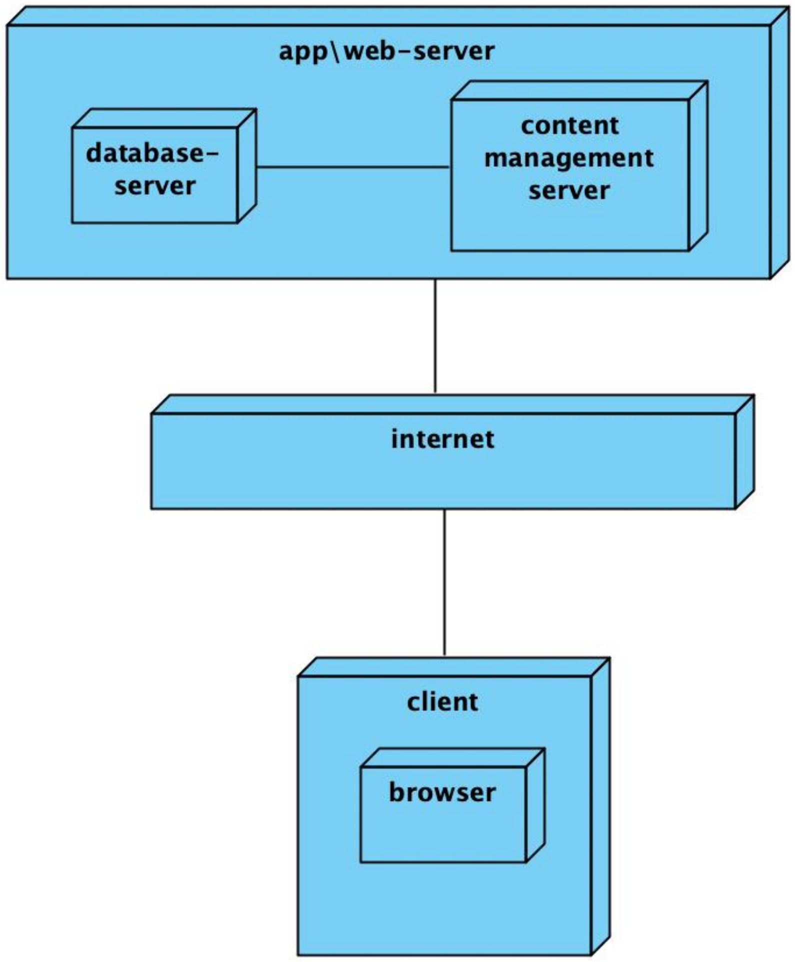

Figure 2 you-tube (clint-server architecture)

The functional requirements:

1. *The system shall allows user to sign-in.*
    - Through the browser on the client side, the user sends a request to the server to log in, the server verifies the entered data and matches it with the data in the database, then a response is sent to the client with acceptance or rejection.

2. *The system shall displays home page for user (this is the main page).*
    - Every login from the client to YouTube will send a request to the server to display the main page, the content management server will submit the request to the client.

3. & 4. The system shall allows user to search any video through entering keyword.

The system shall displays the videos related to keywords that user enter while search.

- The client sends a request with keywords to the server, the server collects all the related videos from the database, the content server arranges them and sends them to the client.(figure3)

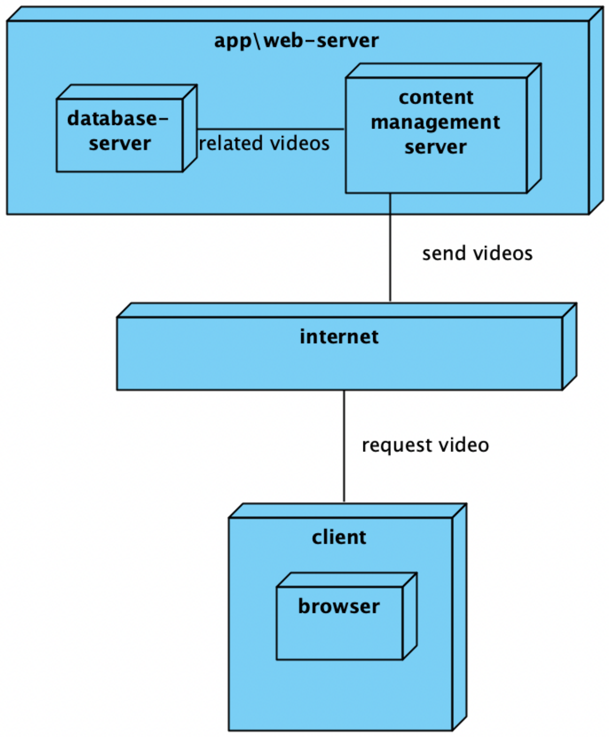

Figure 3

5. & 6. The system shall allows user to share videos with friends and family.

The user should access share button to send videos to someone from friends list.

- The client sends a request to the server to send the video to a specific address, the media server sends the video to the desired address.(figure 4)

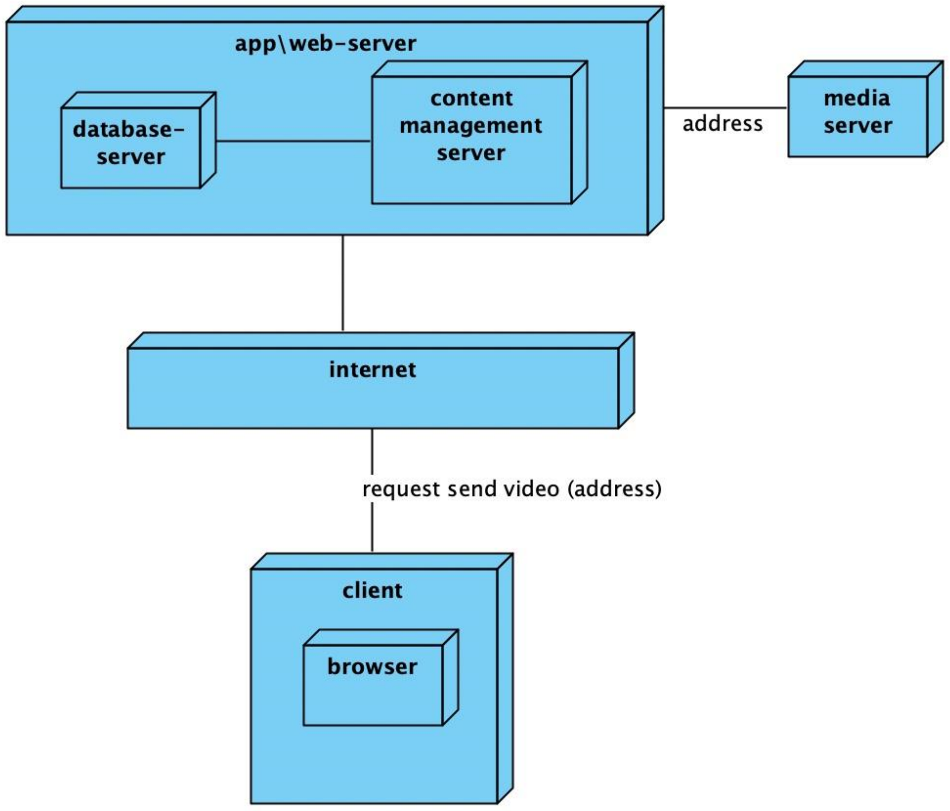

Figure 4

7. The system should allows user to download videos.

- The client asks the server to download the video, the server downloads the video to the client device.

8. The system shall allows user to upload their videos into his account.

- The client sends the video that it wants to upload to the server. The server saves the video in the database and the content server displays the video in the client's account.(figure5)

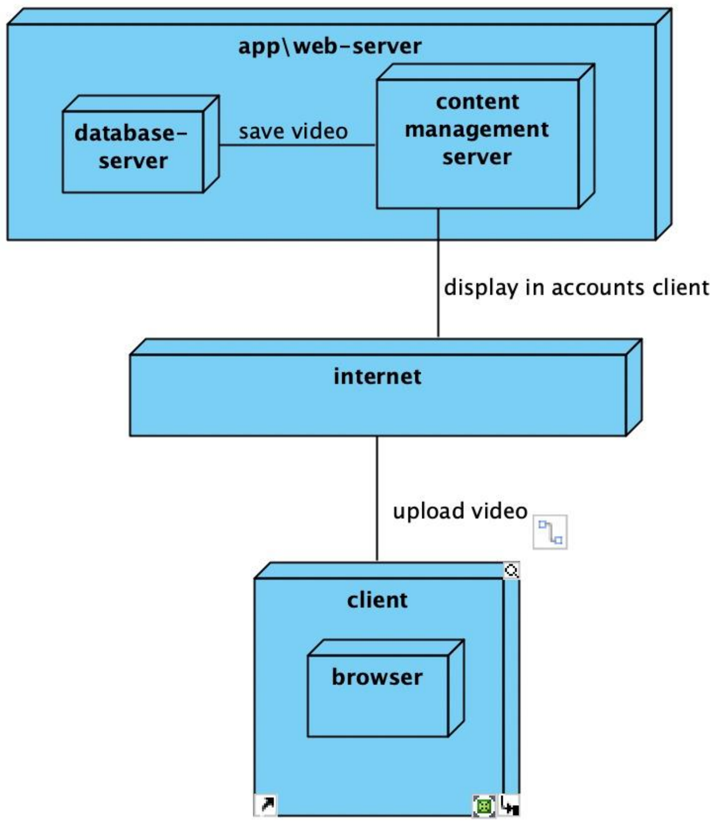

Figure 5

9. The system shall displays all the channel subscribed by user.

- The client requests to display the subscribed channels, the content server collects data from the database and displays the subscribed channels.(figure6)

10. The system shall shows all the history of all the videos that the user watched so far in the his library section.

- The client requests to display the recently viewed videos, the content server collects the videos from the database and displays them to the client.(figure6)

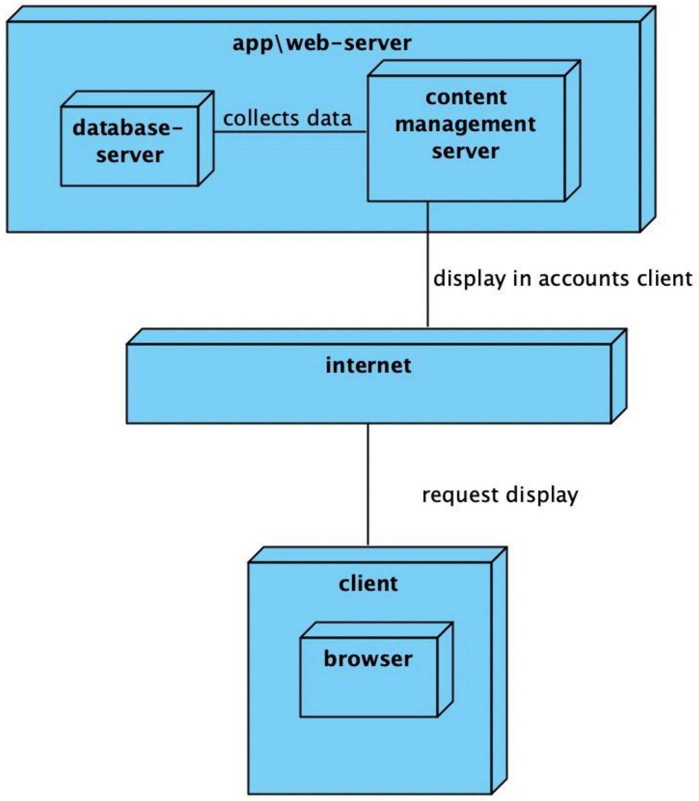

Figure 6

11. *The system should allow companies to display their ads for a fee*

- The company is the client and it have to register in YouTube server by creating a channel then upload the advertising short video into it. The company must have a business account in the google ads server and link the advertising video on their channel to it, now they have to search for popular content creators through browsing server and choose the videos they want to view their ads on, after that they request it to the YouTube approvals server and pay the fee after it's approved.

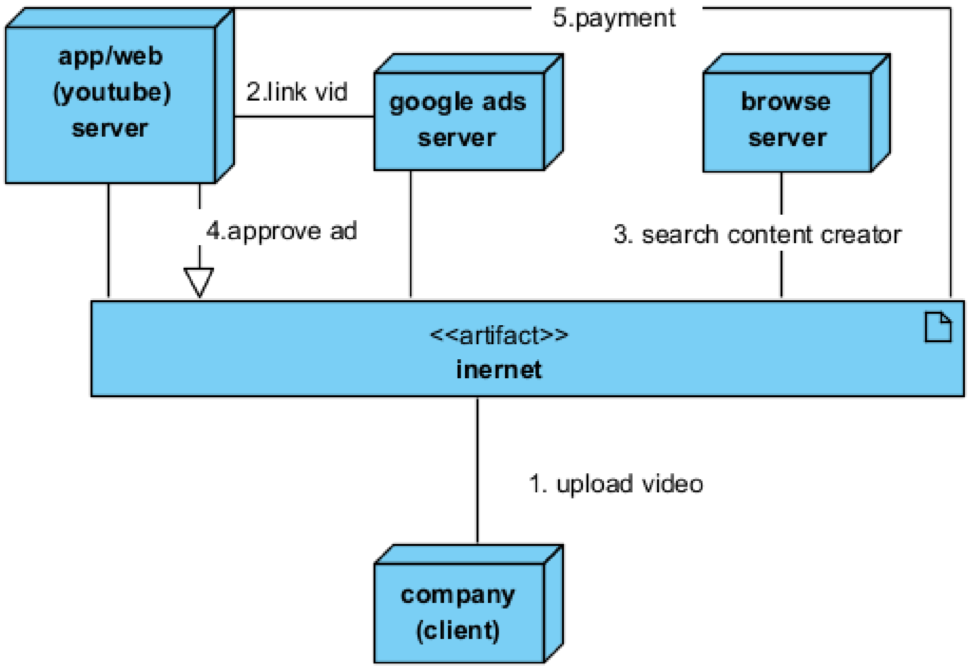

Figure 7

12. *The system should allow users to sign in youtube premium for a fee*

- The client have to be registered by having a channel in the YouTube server and he have to send a request to subscribe in YouTube premium to the server then his membership to will update his info in the database to a "premium user" .

13. *The system shall not display advertisements to the youtube premium users.*

- All the clients saved as "premium users" in the data base will have a feature that the YouTube server will block ads from them.

14. *The system shall allow the users to write comments on the videos.*

- The clients can upload their feedback to the videos provided by the server and they are shared to all other clients.

15. *The system shall display the number of views to the users*

- The server shows the clients a numerical indication made by a counter in the server that counts how many times the video have been played by all clients.

16. *The system shall allow the user to close the comments section for their own videos.*
- A client can request the server to block the comments section written by users.

17. *The system shall allow users to upload a display picture and description to each video .*
- The systems server gives the client opportunity to upload an image for each uploaded video and a description including key words about the video’s topic.

18. *The system shall divide the videos in terms of content into age groups.*
- The server management divides all the received videos by clients into age groups with accessing limitations

19. *The user shall have an account to subscribe to channels.*
- The server gives the option of subscribing to channels only for the clients who are registered in their database

20. *The system shall allow users to edit or delete their uploaded videos.*
- The YouTube app server gives the clients the option to delete or edit or update their videos from the server.

21. The system shall allow the user to specify who has access to the videos on their own channel.

- *The client chooses who can access the videos on his channel and the content management server applies the access permissions according to the customer's choices.*

22. The system should display videos that match the user's interests on the main interface.

- *When the client enters the application, the content management server displays videos that suit the interests of the client and which are stored on the database server.*

23. The system shall allow the user to manage app alerts (disable or allow notifications).

- *The client opens the application and goes to the settings to control alerts and notifications and then the content management server saves the new changes.*

24. The system shall have a library that displays playlists

25. The system shall allow the user to add videos that he wants to watch later to the playlist called "watch later".

26. The system should have a list called "Liked Videos" that shows the videos that the user liked.

- *When the client logs in to the system, the content management server displays a section on the main interface called “Library” which contains many lists such as “Liked videos” and “Watch later” lists, and the videos saved in these lists are stored in the database server.*

27.The system shall display on its main interface a section called "shorts" that displays short videos.

- *When the client enters the application, the content management server displays a section on the main interface called "Shorts" which displays short videos.*

28. The system should allow users to send their feedback and opinions by clicking on "Send Feedback" on the main page.

- *The client sends his views and comments to the content management server, which sends them to the database server for storage.*

29. The system shall maintain the security and copyright policies, remove any videos that violate the laws.

30. The system shall be enable users to inform the competent authorities when their rights are violated or they are exposed to a problem.

- *The client sends the report to the server if it encounters any problem or violation, then the content management server will send the data to the database server to show the included laws and then the content management will solve the problem and block the violating content.*

The non-functional requirements:

1. The user shall have at least 500kbps of net connection in order to play the video without buffering . (product req. => performance req. => speed)

- The client must have a dependable internet connection to access the YouTube services.

*2. The system shall be available for the users 24*7. (product req. => availability.)*

- The clients have the opportunity to use the app server all the time.

3. The system easy to use and access to videos in no more than 5 seconds. (usability).

- The server is well implemented where it's being accessible by the client by less than 5 sec.

4. System recovery shall take place within 2 minutes if fails to run it. (Maintainability)

- The server can be maintained within 2 min. after a failure.