

Assignment: Netflix++

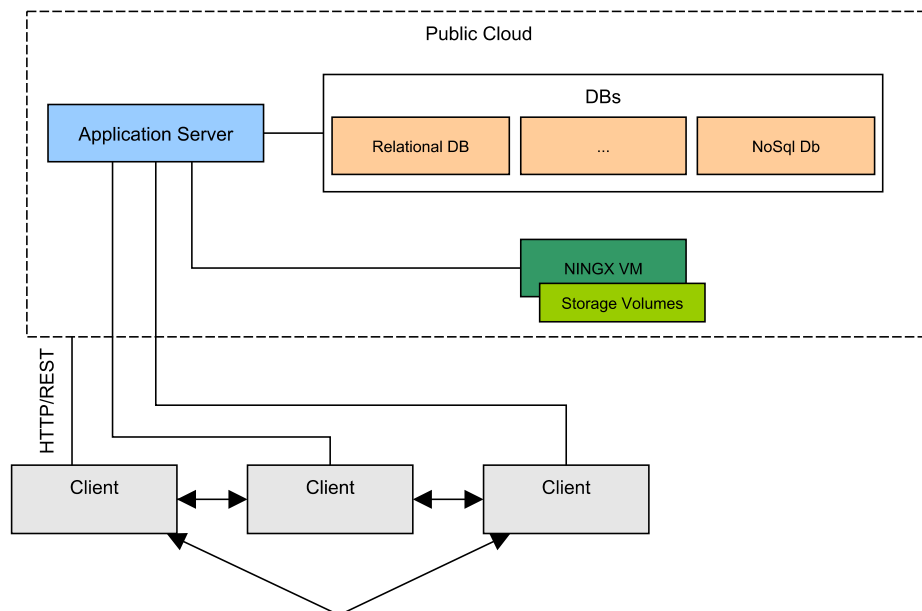
Mobile Device Programing 22/23

September 28, 2022

This assignment requires the implementation of a streaming facility, that extends normal Netflix capabilities. A user must be able to watch Video-on-Demand (VoDs) but the should get it from the nearest place (near mobile devices). A content management system (CMS) must be provided to handle content.

All the code must be written in Java (backend and frontend).

The system **must have exactly** the components depicted below:



This streaming service has to offer the same level of functionality that the regular Netflix provides. The backend features include the following (grade 10/20):

- Database with the entire catalog. Do not store the movies in the DB! The DB should have links to your storage.
- Different devices might have different video and audio codec support. You must provide both a high resolution (1080p) and a low resolution versions (360p) for each movie. Each movie must have different links to different movie formats/encodings (use **ffmpeg**).

Lastly, the system will require actual Android frontends (one for the streaming and the other to interact with the CMS). You are free to use all the APIs, libraries and resources available. However, the following must be supported:

- Both apps must support users and passwords.
- Use Jetty+Jersey for the Application Server and NINGX for the reverse proxy.
- For the DBs you can use a combination of MariaDB and Cassandra.
- The video streaming app must (grade 2.5/20):
 - A startup/loading screen showing your proud logo.
 - The one activity is responsible for video streaming. You should be able to navigate the contents stored in the backend, and select the stream you want to watch.
 - Implement a mesh protocol for transferring files direct from other clients.

The CMS app must (grade 2.5/20):

- Allow the upload and removal of movies.
- Automatically generate a low resolution version for each movie uploaded.
- Allow the creation and removal of streaming app users.

Mesh sub-subsystem (5pt)

- Implement a lightweight Bittorrent protocol, between seeders and clients.
 - Each file is divided in chunks, each chunk has a max of 10MB.
 - Each file, and every chunk, must have a hash (SHA256).
- Assume one seeder per file.
- Must support multiple clients per file.
- Use Netty (<https://netty.io/>) or implement using low-level network Android API.
 - If you want to use HTTP: <https://www.baeldung.com/java-netty-http-server>
 - Otherwise, build on own binary protocol:
 - * <https://github.com/reactor/reactor-netty/tree/main/reactor-netty-examples/src/main/java/reactor/netty/examples/tcp/echo>
 - * Serialization: <https://javamana.com/2022/117/202204270738551129.html>
- For the DBs, use open-source ones, such as SQLite.
- All clients have direct connections (no firewalls between them), i.e., all clients are in a local common network
- **All other components must be design and build by the student!**

Video Files:

Big Buck Bunny (Comedy, Animation)

<https://archive.org/details/PopeyeAliBaba>

The Letter, Lego movie [2003] (Comedy, Animation, LEGO), 6m30s

<https://archive.org/details/tl>

Charlie Chaplin's "The Vagabond" [1916] (Comedy), 24m43s

https://archive.org/details/CC_1916_07_10_TheVagabond

Night of the Living Dead [1968] (Sci-Fi / Horror), 95m17s

https://archive.org/details/night_of_the_living_dead

Popeye the Sailor Meets Ali Baba's Forty Thieves [1937] (Comedy, Animation), 16m58s

<https://archive.org/details/PopeyeAliBaba>