Fraunhofer FOKUS
Institut für Offene Kommunikationssysteme

Technische Universität Berlin

AWT PJ
Video Streaming Mixer Library

Poonam Kumari Roy, Mohamed Mesto, Yuni Quintero | AWT PJ | Workshop 3

Fraunhofer FOKUS

Agenda

1. Wrap up: Problem Statement and Schedule
2. Output Manifest
3. Library Component
4. Final Demo

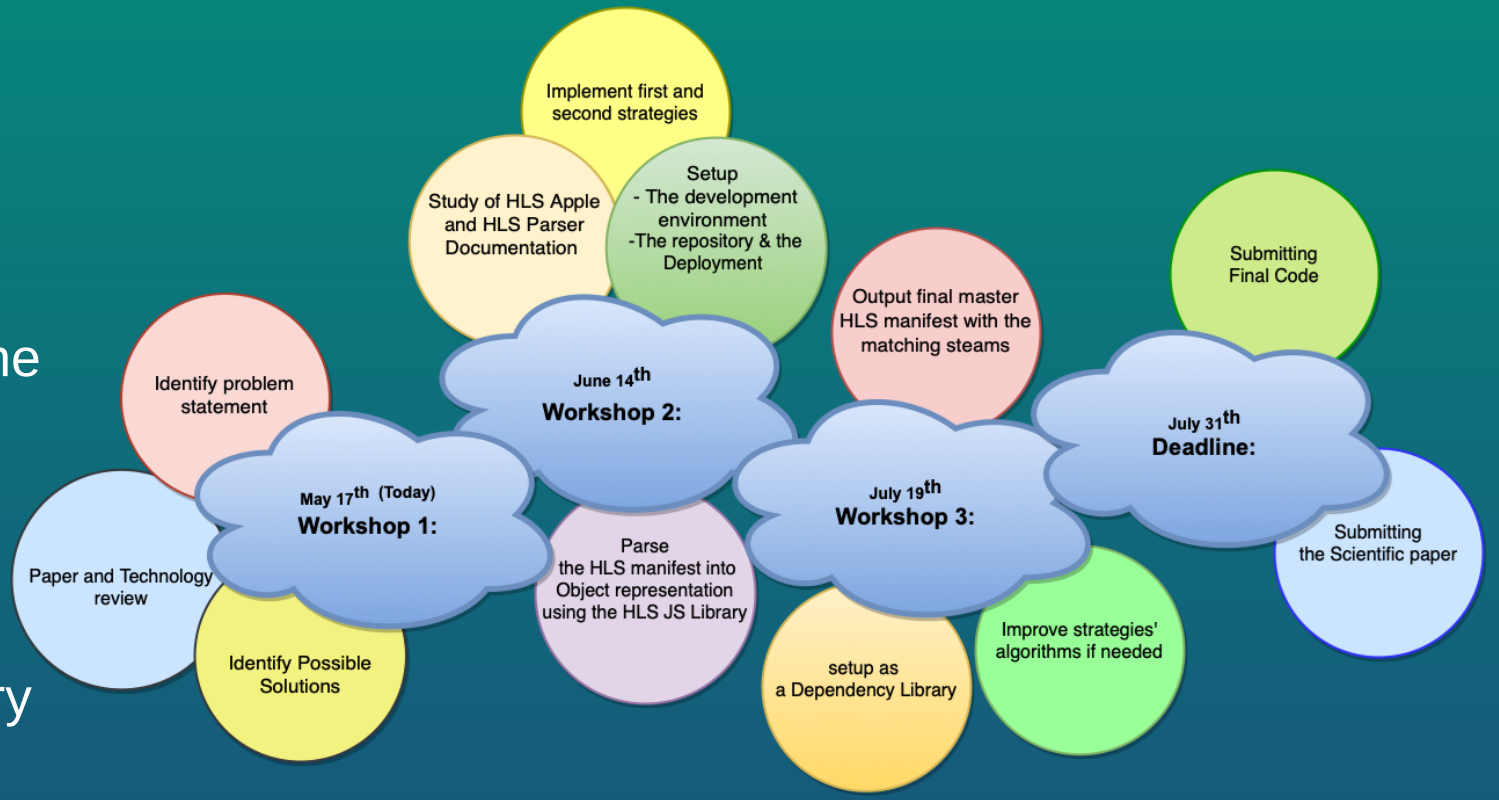
Wrap up: Problem Statement and Schedule

We have implemented two strategies that select compatible streams in regards to resolution in order to join these streams into a single master playlist.

The first strategy consists of matching resolutions against the first element of the input array.

The second strategy consists of an intersection of resolutions.

We use as tools the hls-parser js library and node js as a development environment.



Poonam Kumari Roy, Mohamed Mesto, Yuni Quintero | AWT PJ | Workshop 2

Output Manifest

After implementing both strategies, we have as a result an array with the compatible streams and an array with the resolutions that matched.

We have to obtain the matching variants for each resolution of the final output.

In order to remove these duplicates, we choose the variant with the maximum bandwidth value.

For each matching resolution, we fetch the media playlist of all the variants to obtain their segments which will be joined in an array to create a new Media Playlist object.

Then, we create a new Variant object that has as uri the filename of the media playlist m3u8 file.

Finally, we create a new Master Playlist object joining all the variants in the dictionary in an array. We convert this playlist into string representation too and write it on a master.m3u8 file.

Library Component

We created an npm package that has in the file utils.js two methods algorithmA(urls: String) and algorithmB(urls: String), along with other utility functions like joinSegments(), makeRepresentationsDict(), createMasterPlaylist().

These two methods are then exported in the index.js of the package, from which an external node app can import as a dependency with the following command:

```
npm i awt-pj-ss22-video-streaming-mixer-library-1
```

Final Demo

Poonam Kumari Roy, Mohamed Mesto, Yuni Quintero | AWT PJ | Workshop 3

Resources

- https://bitmovin.com/adaptive-streaming/
- https://www.wowza.com/blog/adaptive-bitrate-streaming
- HLS documentation https://datatracker.ietf.org/doc/html/rfc8216
- hls-parse js library https://www.npmjs.com/package/hls-playlist-parser
- R. Seeliger, D. Silhavy, Dr. S. Arbanowski “Dynamic ad-insertion and content orchestration workflows through manifest manipulation in HLS and MPEG-DASH” https://ieeexplore.ieee.org/document/8228708
- https://developer.apple.com/documentation/http_live_streaming/about_apple_s_http_live_streaming_tools
- https://docs.npmjs.com/creating-and-publishing-private-packages
- https://mp4.to/results/?conversion=5bf78dd7b37b4c6cb8cd65892e10a895