360 video streaming across the network

Quality and Usability Lab SS22

Maurizio Vergari, Prof. Dr. Stefan Schmid, Max Franke

SoSe 2022

Final presentation

Emirali Caferzade, Hanneng Hu,
Christian Gumprecht, Yining Cong

Agenda

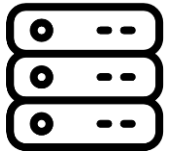
- **Motivation**
- Related literature
- Solution/Teams
  - camera
  - network (Emirali & Hanneng)
  - oculus/website (Christian & Yining)
- Presentation of prototype idea (demo-video)
- Time schedule
- Outlook
- Appendix

Motivation

How to stream 360 live video from camera to oculus?

Motivation

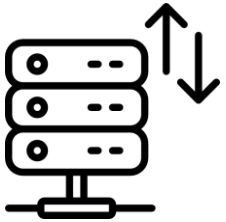
Advantages



Doesn't need much available space on server side

Possibility to broadcast to several headsets simultaneously

Challenges



Bandwidth intensive

Requires sub-millisecond

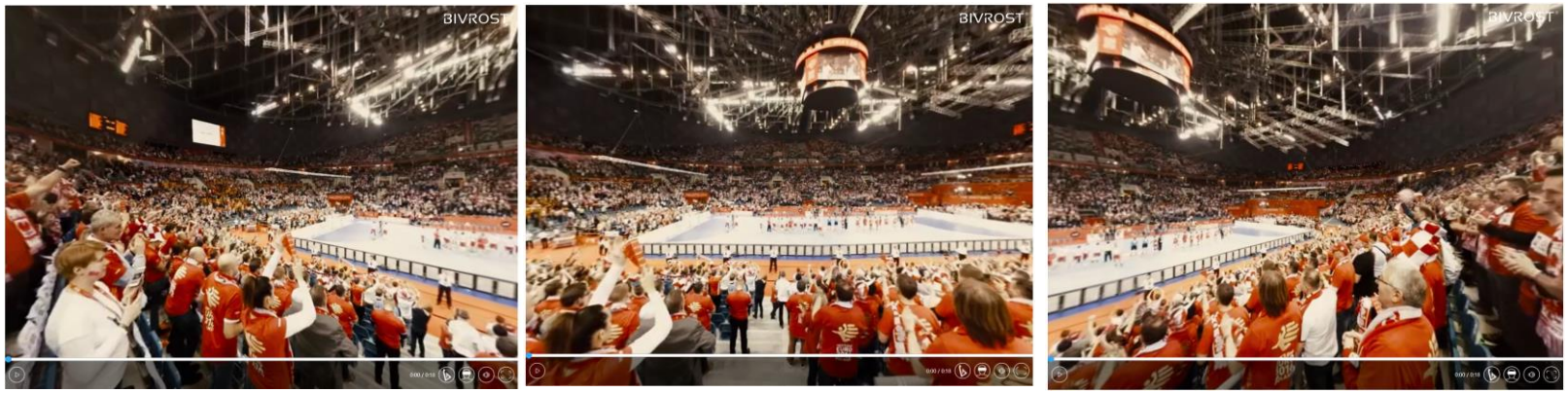
https://vr-sync.com/stream-360-video-to-virtual-reality-headsets/

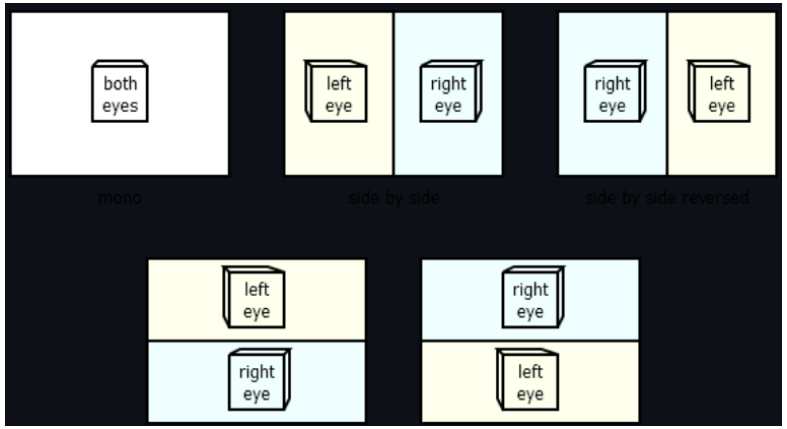
Agenda

- Motivation
- **Related literature**
- Solution/Teams
  - camera
  - network (Emirali & Hanneng)
  - oculus/website (Christian & Yining)
- Presentation of prototype idea (demo-video)
- Time schedule
- Outlook
- Appendix

Related literature

BIVROST 360WebPlayer

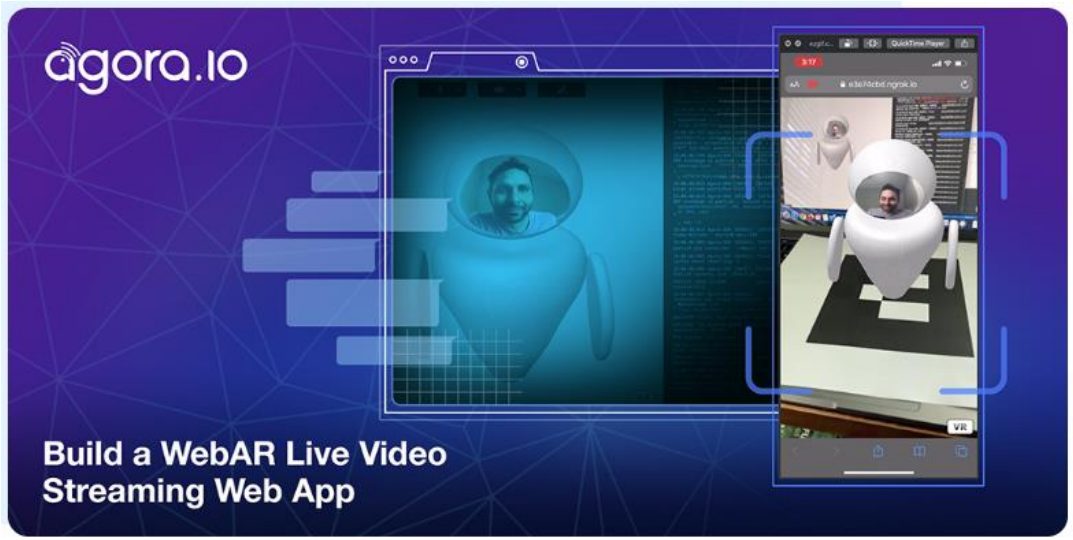




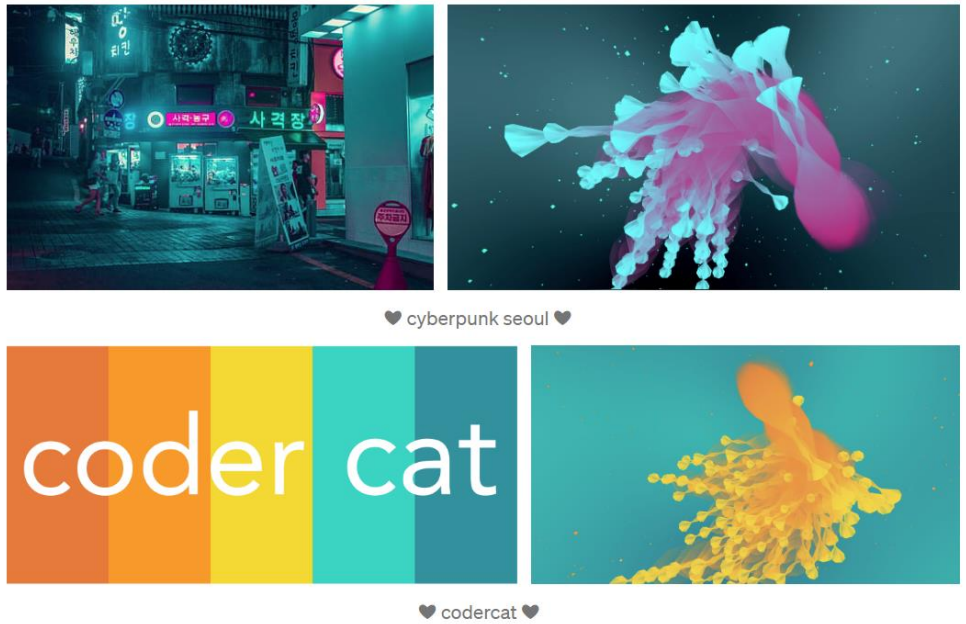
https://github.com/BIVROST/360WebPlayer
https://beprosto.github.io/webxr-tutorial/

Related literature

Agora WebXR & A-Frame



Agora

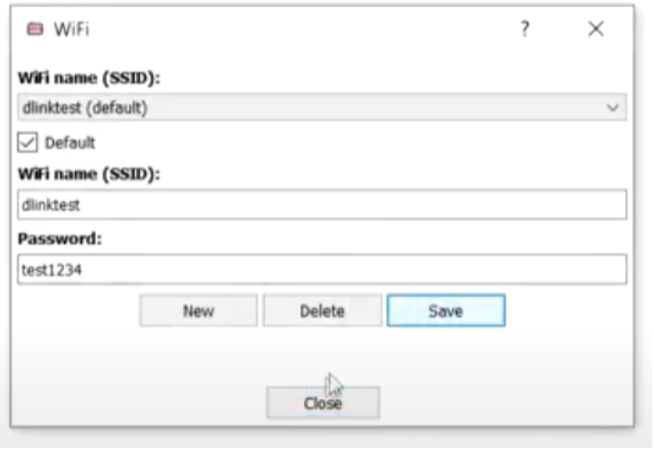


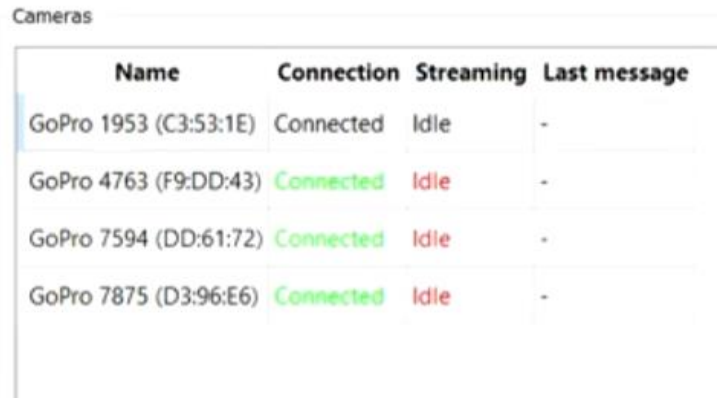
A-Frame

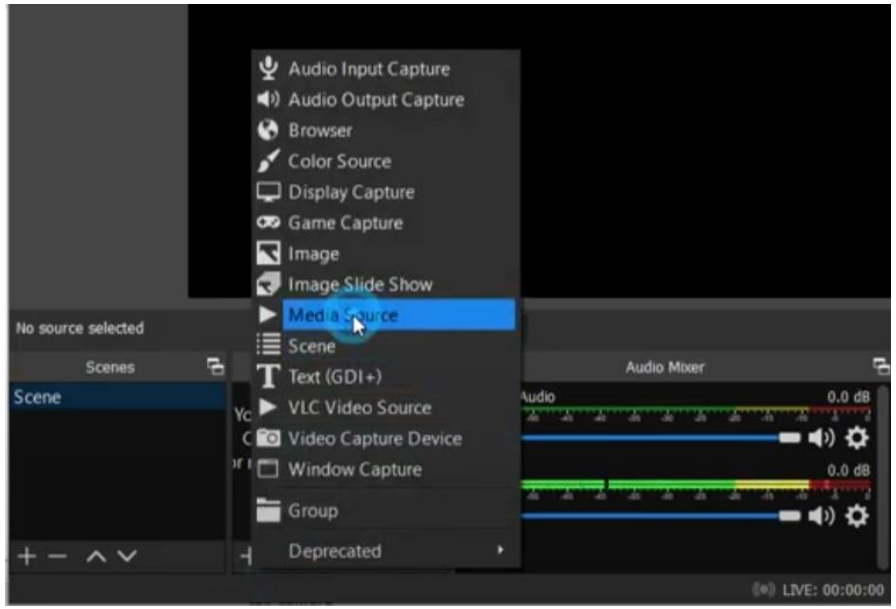
https://www.agora.io/en/blog/build-a-webar-live-video-streaming-web-app/
https://github.com/digitallysavvy/AgoraWebXR
https://webvr.directory/
https://github.com/aframevr/aframe

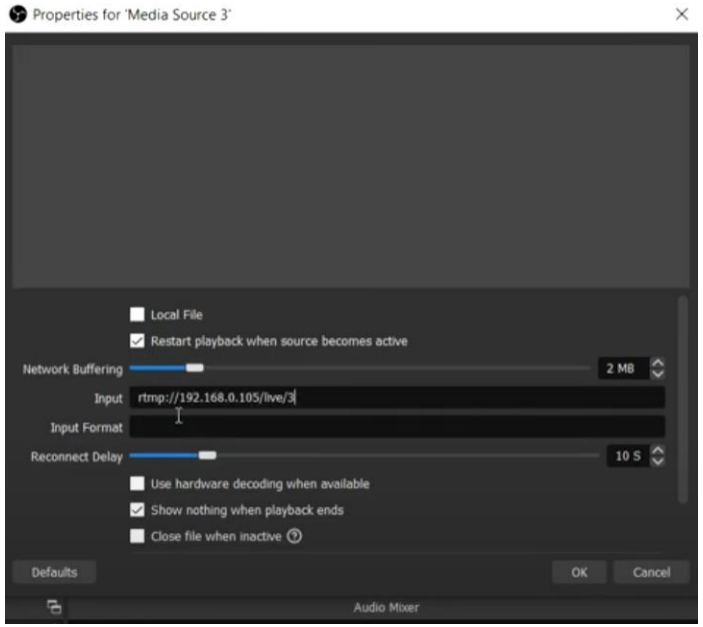
Related literature

live stream from multiple GoPro cameras to OBS studio on Windows







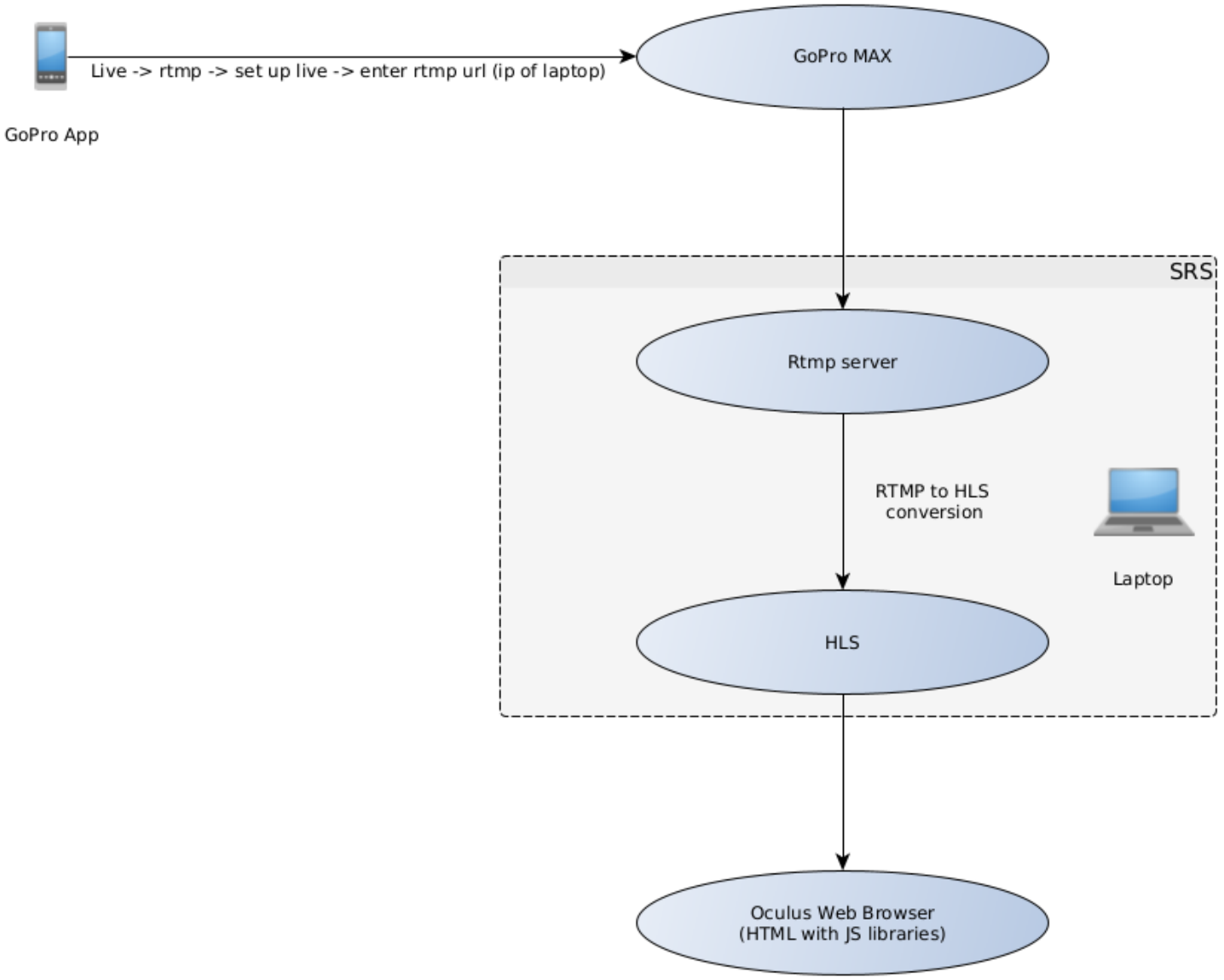


https://www.youtube.com/watch?v=e328xxdbRyk
https://www.youtube.com/watch?v=6sVs4PFdxPc

Agenda

- Motivation
- Related literature
- **Solution/Teamwork**
  - camera
  - network(Emirali & Hanneng)
  - oculus/website (Christian & Yining)
- Presentation of prototype idea (demo-video)
- Time schedule
- Outlook
- Appendix

Solution/Teams (flow chart)



Solution/ camera

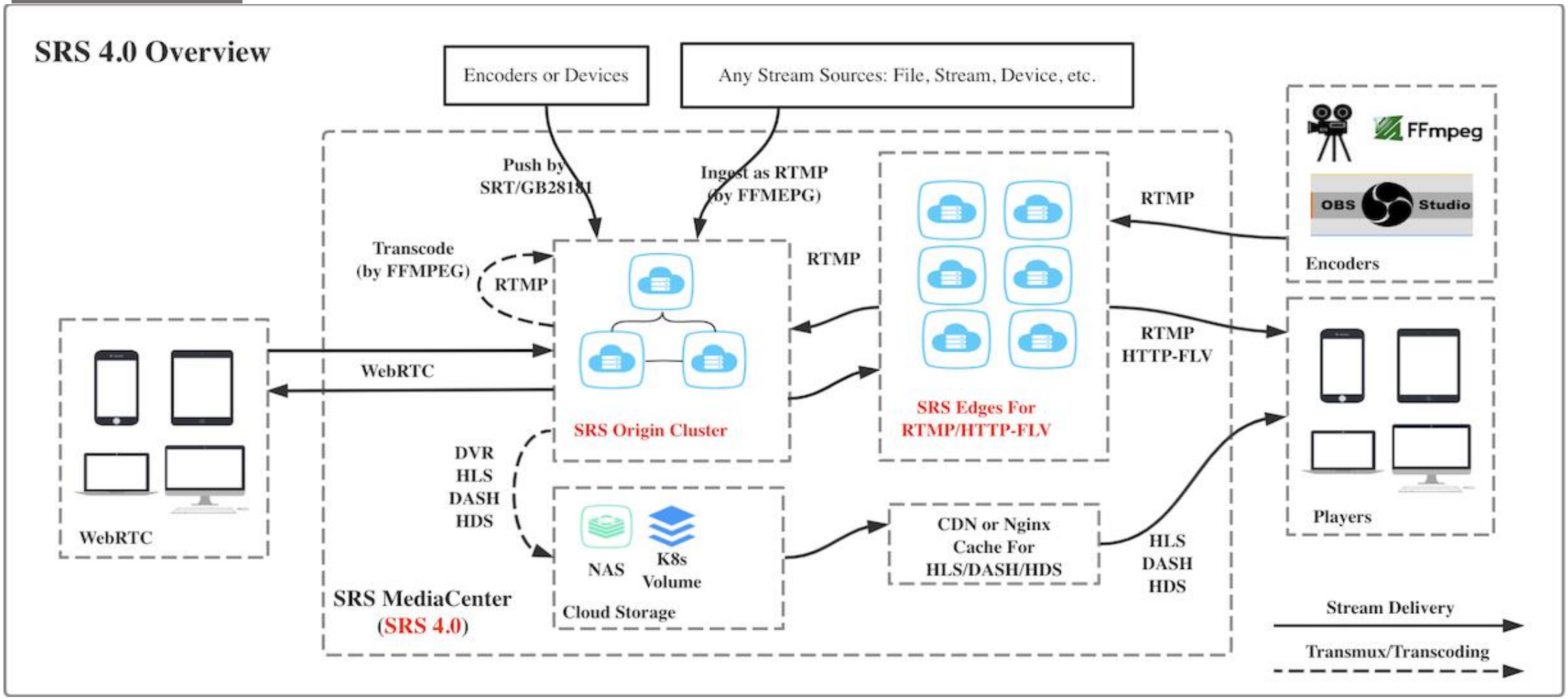
GoPro

MAX

Quik

Solution/Team(network)

SRS Server Arch



Solution/Team(network)

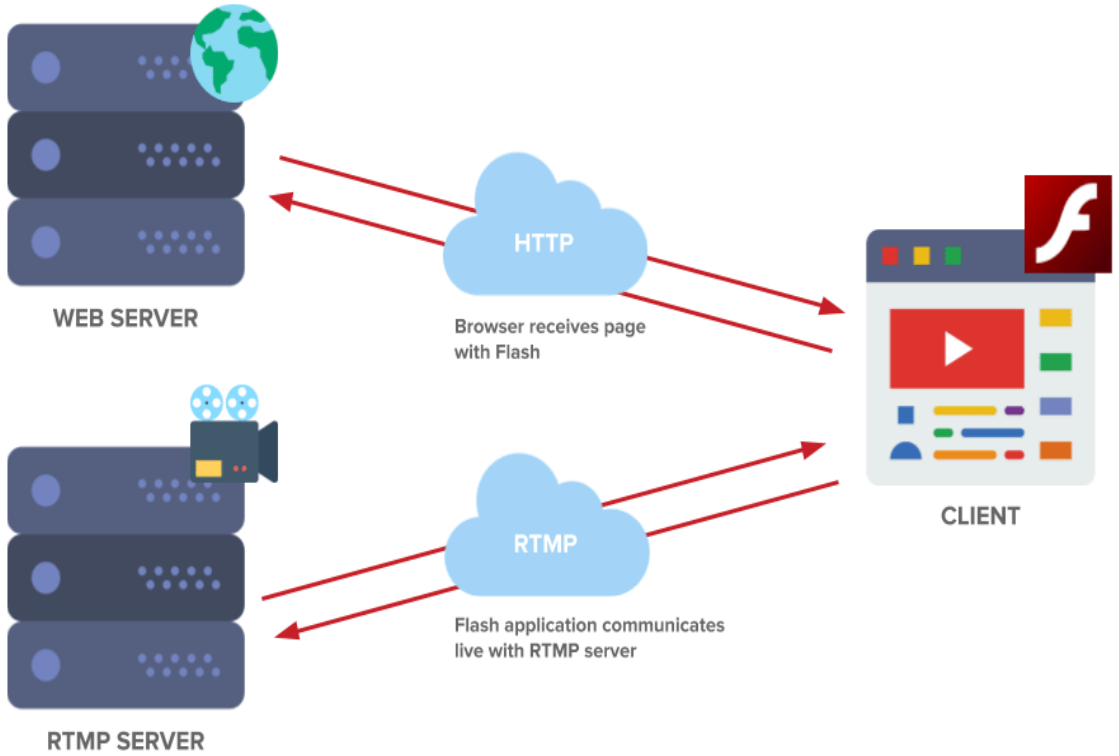
SRS/ RTMP

- **R**eal-**T**ime **M**essaging **P**rotocol
- TCP based protocol
- Audio, video and data over the internet
- Low latency in streams

- Approach:

  - Record with RTMP-compatible camera or encoder

  - Transcode the media stream into another protocol (HLS)

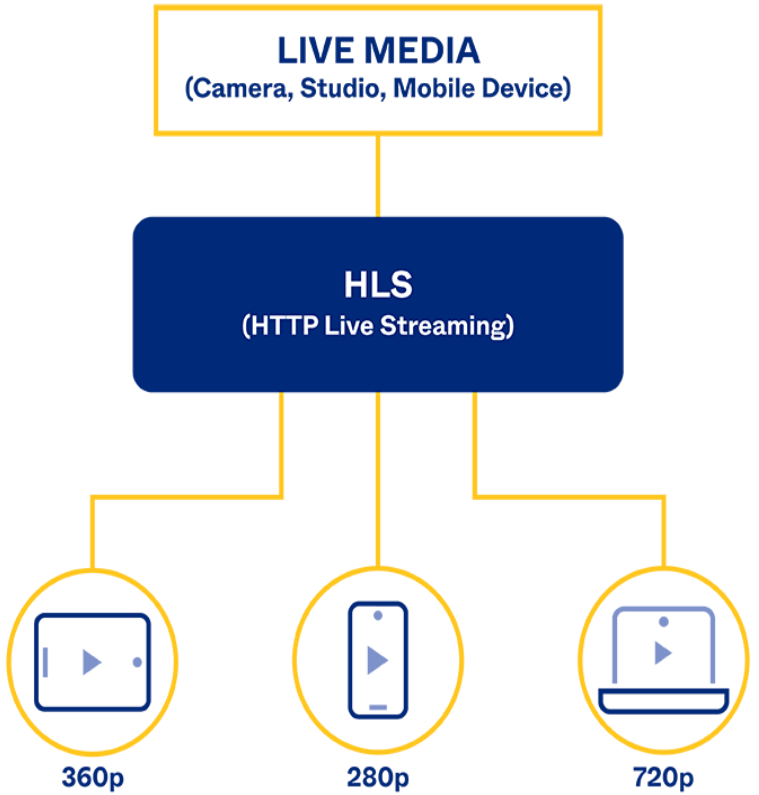


Youtube Use Case

Solution/Team(network)

SRS/ HLS

- About HLS:
  - HTTP based
  - Developed by Apple
  - Adaptive bitrate streaming
  - For both live and VOD
  - Breaks overall stream into sequence of file chunks (ts file)
  - A list of available streams are served to the client as a playlist
  - -> HTTP server sending m3u8 livestream playlist files



Solution/Team(network)

SRS/ HLS

- SRS:
  - Demuxes RTMP stream and remuxes it to livestream playlist (encoded at different bit rates)
  - Hosts the HTTP server
  - Outputs livestream URL `http://179.67.84.227:8080/live/livestream.m3u8`
- Important data parameters configurable using the hls.config:
  - e.g hls_window (in seconds): determines m3u8 playlist size
  - e.g hls_fragment (in seconds): determines number of ts files in playlist

- Limitation of SRS: stream must be in H.264+AAC format

- Why HLS ?
  - We chose frame rate over image quality
  - Widely supported
  - Easy to implement using SRS

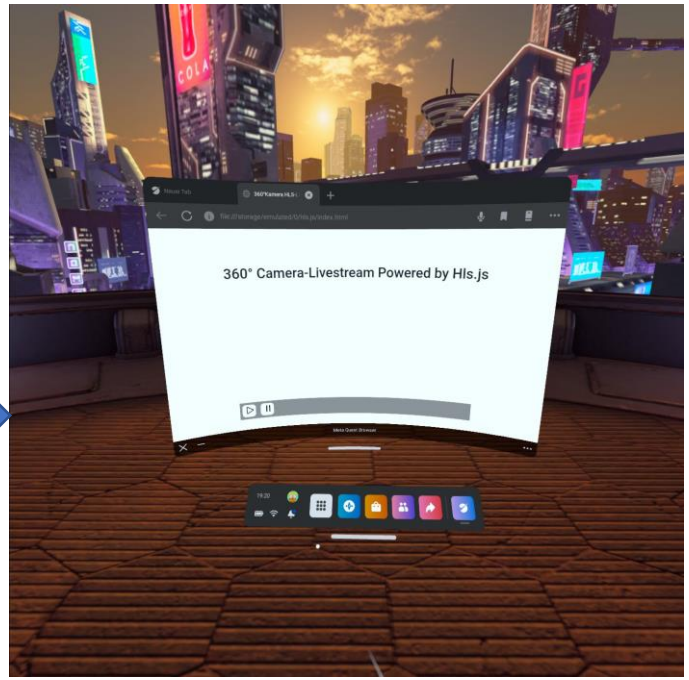
Solution/Team(oculus/website)

Meta Quest Browser

- Meta Quest2 Client: Meta Quest Browser
- Already installed on all Meta VR-headsets
- Browser based on Chromium
- HTML5, Css3 & JavaScript capable
- 360° video -> only 3Dof content (no 6Ddof)

COMMAND

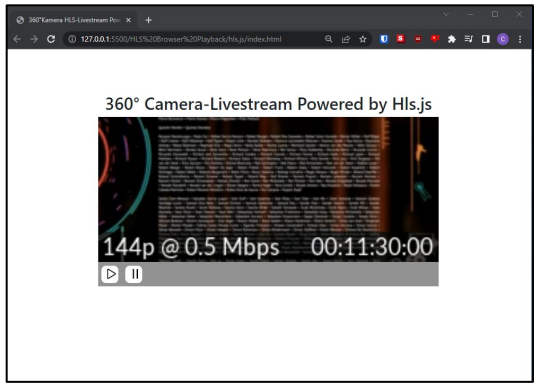
```
adb -s _ODH_CONNECTED_DEVICE_SERIAL_ID_ shell am start \-n
com.oculus.os.vrbrowserlauncher/.MainActivity \-a
android.intent.action.VIEW -d
'file:///storage/emulated/0/hls.js/index.html'
```



Solution/Team(oculus/website)

HTML, HLS.js & Javascript

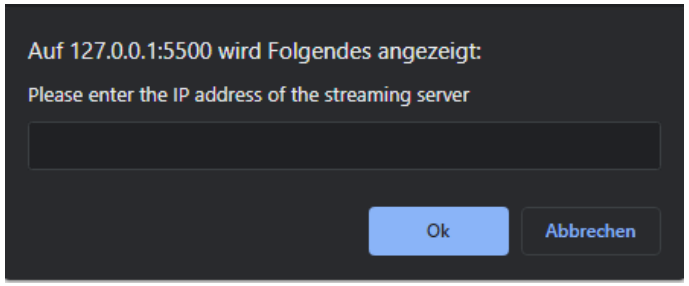
HTML video tag



JavaScript library hls.js

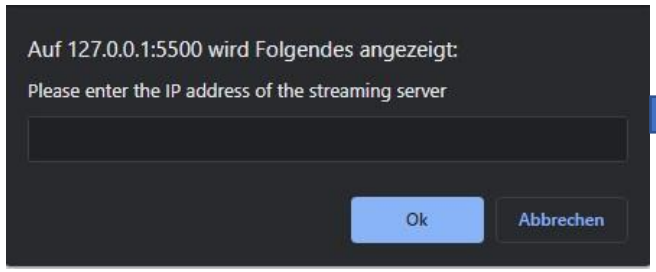
```
<script>

let IPAddr = prompt("Please enter the IP address of the streaming server");



    var test_stream = 'https://cph-p2p-msl.akamaized.net/hls/live/2000341/test/master.m3u8'
    var video = document.getElementById('video');
    if (Hls.isSupported()) {
      var hls = new Hls({
        debug: true,
      });
      hls.loadSource('http://'+IPAddr+'/live/livestream.m3u8');
      hls.attachMedia(video);
      hls.on(Hls.Events.MEDIA_ATTACHED, function () {
        video.muted = false;

      });
    }
```

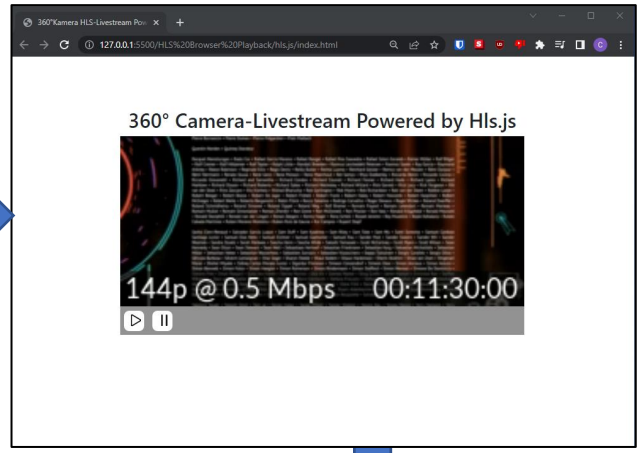
Javascript "prompt()"



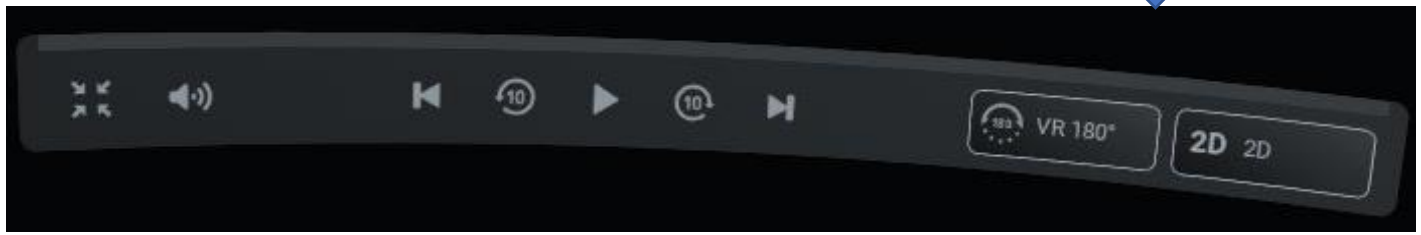
Solution/Team(oculus/website)

Viewing video in VR





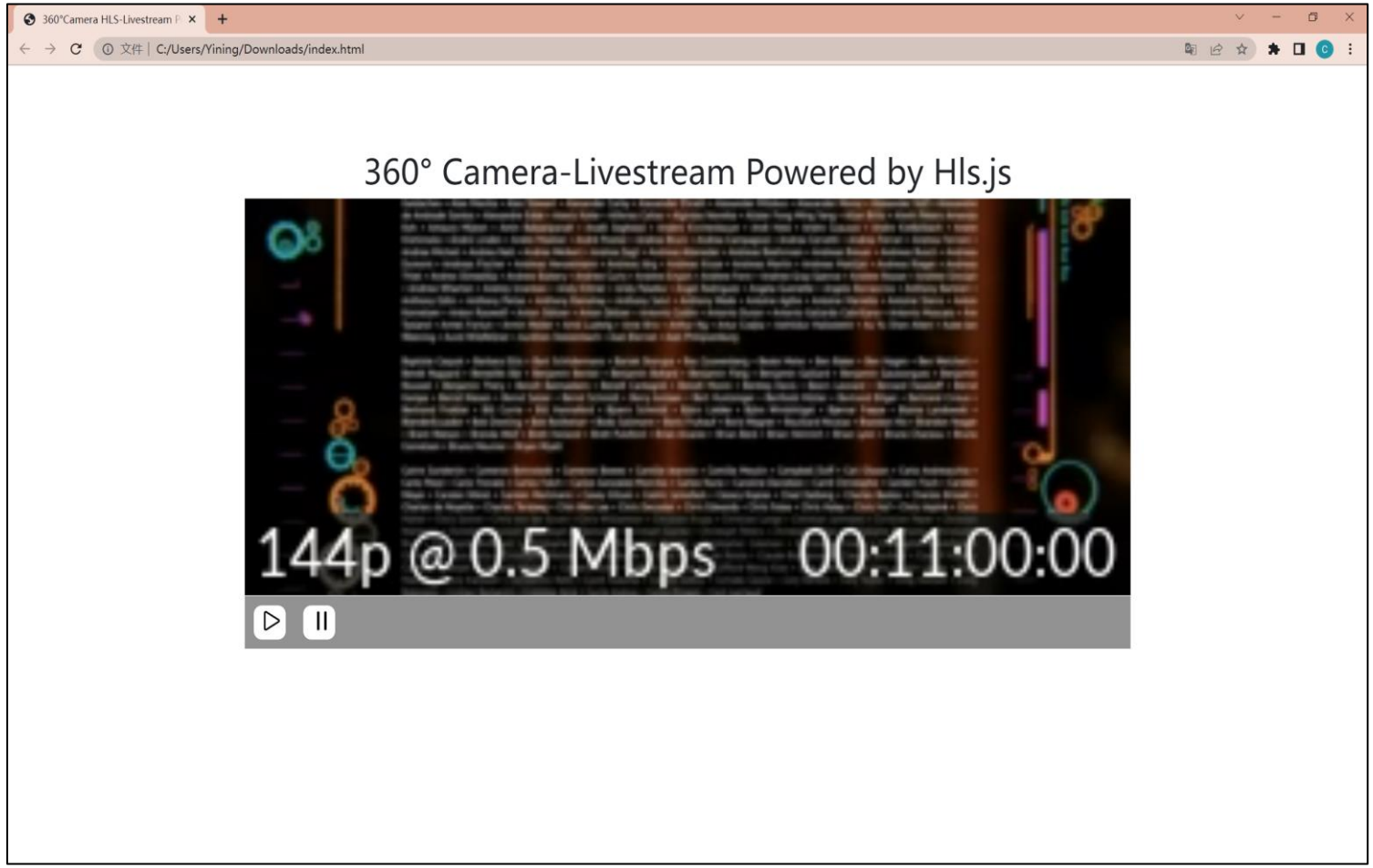
- Use of browsers native VR capabilities
- Browser's natively able to view 180° & 360° video
- Built in media control



Solution/Team(Website)

- First attempt: Unity and WebXR,
- Second attempt: Use AVPro package to live stream 360 video in unity, but had challenges when exporting video to WebXR

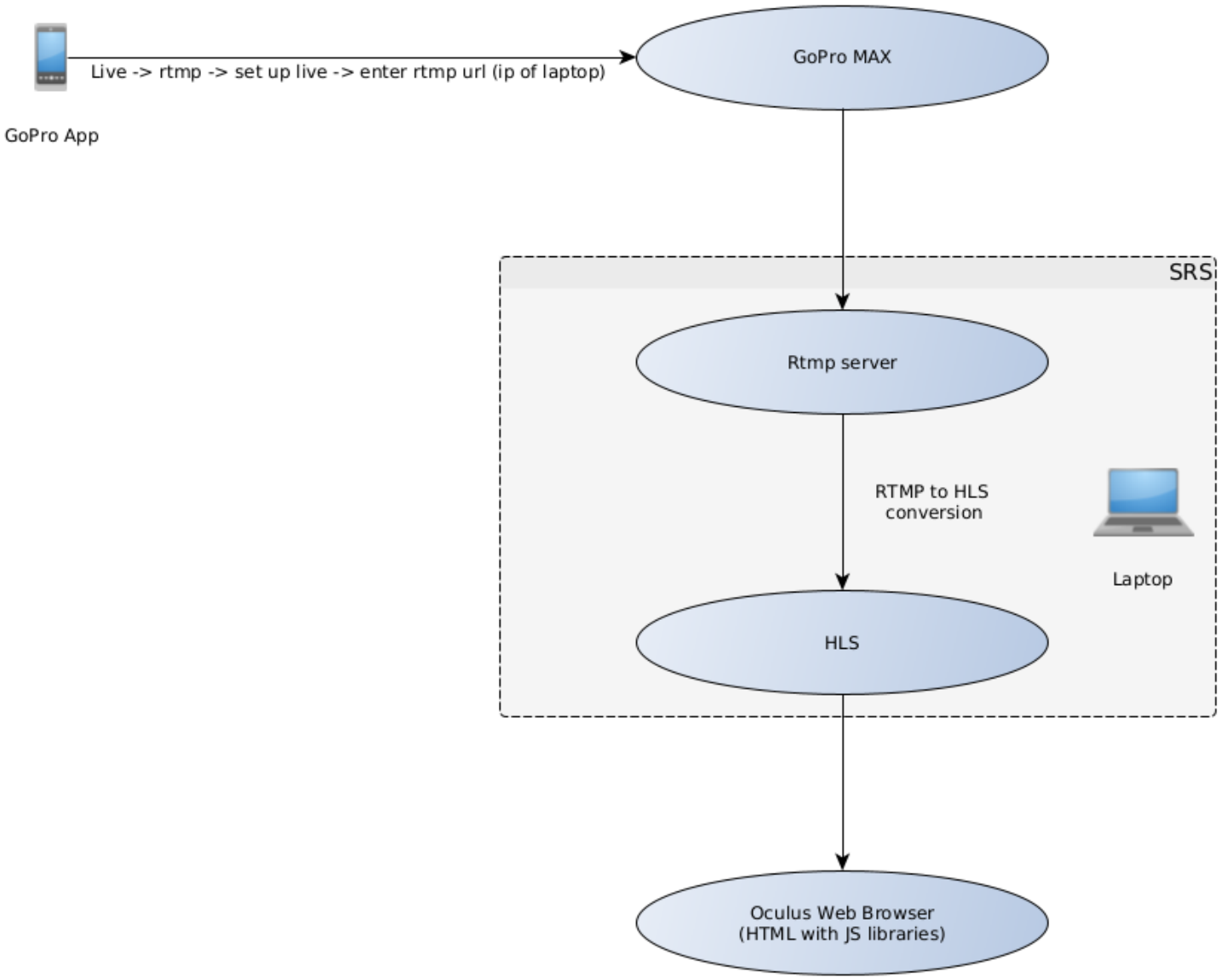
Then try to do UI interface in html and css



Agenda

- Motivation
- Related literature
- Solution/Teams
  - camera
  - network (Emirali & Hanneng)
  - oculus/website (Christian & Yining)
- **Presentation of prototype idea (demo-video)**
- Time schedule
- Outlook
- Appendix

Presentation of prototype idea



Agenda

- Motivation
- Related literature
- Solution/Teams
  - camera
  - network(Emirali & Hanneng)
  - oculus/website (Christian & Yining)
- Presentation of prototype idea (demo-video)
- **Time schedule**
- Outlook
- Appendix

Time schedule

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1		week 1	week 2	week 3	week 4	week 5	week 6	week 7	week 8	week 9	week 10	week 11	week 12	week 13	week14	week 15	week 16	week 17	
2		Wed 11.05	Wed 18.05	Wed 25.05	Wed 01.06	Wed 08.06	Wed 15.06	Wed 22.06	Wed 29.06	Wed 06.07	Wed 13.07	Wed 20.07	Wed 27.07	Wed 03.08	Wed 10.08	Wed 17.08	Wed 24.08	Wed 31.08	
3	**Researching part**																		
4	Researching the existing situation																		
5	Researching the existing tools																		
6	Researching the existing libraries																		
7	researching about streaming software																		
8																			
9																			
10																			
11																			
12																			
13	**Learn about Devices**																		
14	camera stream testing																		
15	connect camera with phone																		
16	connect camera with pc																		
17	get familiar with oculus device																		
18																			
19																			
20																			
21																			
22	**Sending data**																		
23	send data from camra to server																		
24	using library to encode video																		
25	encoding																		
26	decoding																		
27	send data from server to oculus																		
28																			
26																			
27	**Prepare for mid-term presentation**																		
28	**Prepare for final presentation**																		
29	**prepare for final report**																		

Agenda

- Motivation
- Related literature
- Solution/Teams
  - camera
  - network(Emirali & Hanneng)
  - oculus/website (Christian & Yining)
- Presentation of prototype idea (demo-video)
- Time schedule
- **Outlook**
- Appendix

Outlook

- Enhance framework and design for users.
- Add/Implement security measures for server and website.
- Create APK for oculus, for standalone access.
- Overhaul the website (function&design)
- Assembling all final packages on Github
- Final Report

Agenda

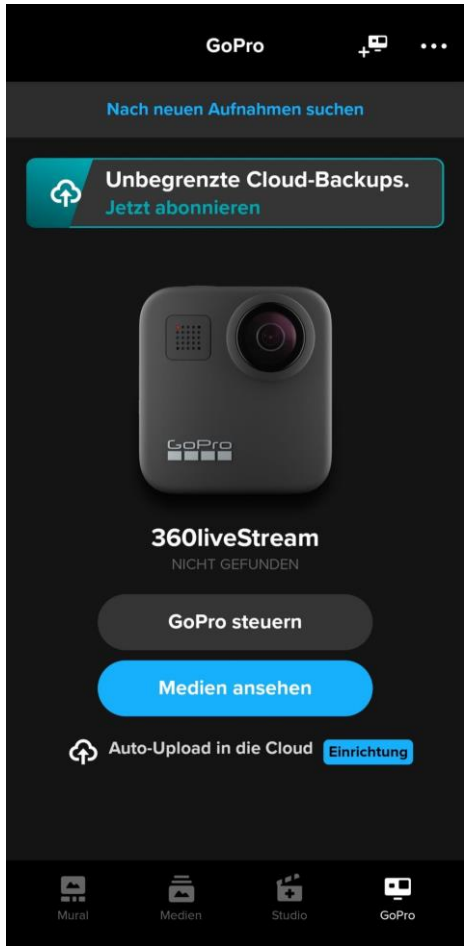
- Motivation
- Related literature
- Solution/Teams
  - camera
  - network (Emirali & Hanneng)
  - oculus/website (Christian & Yining)
- Presentation of prototype idea (demo-video)
- Time schedule
- Outlook
- **Appendix**

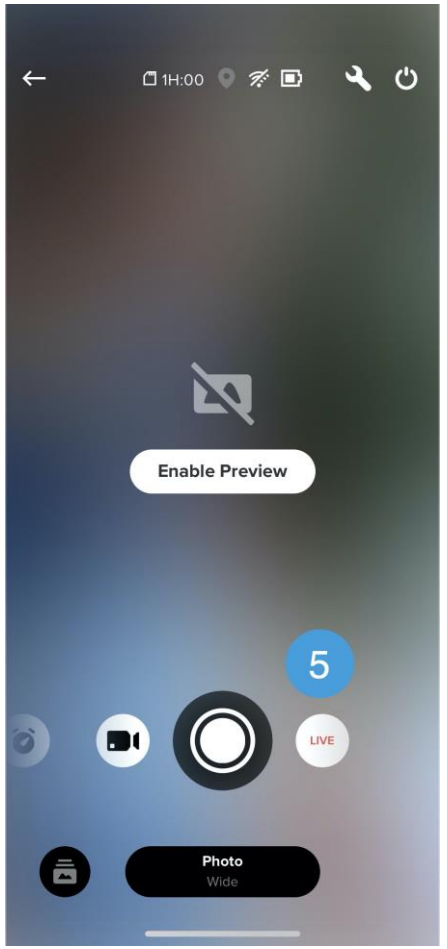
Appendix

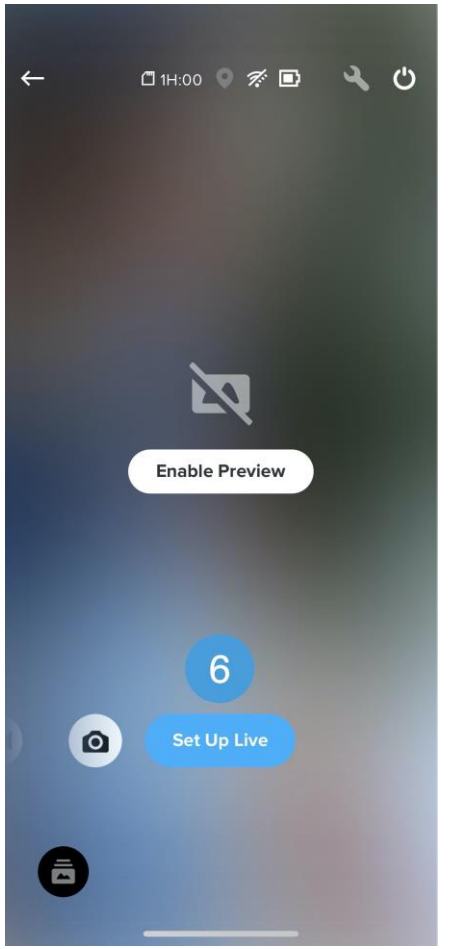
- 1. Set-up the SRS (Server) (using Python-script)
- 2. Copy given rtmp-URL

Appendix

- 3. Open the "Quik" GoPro App
- 4. Tap the camera icon in the bottom left corner of the screen.
- 5. Scroll through the icons on the bottom of the screen, and tap on [Live].
- 6.Now tap on [Set Up Live].

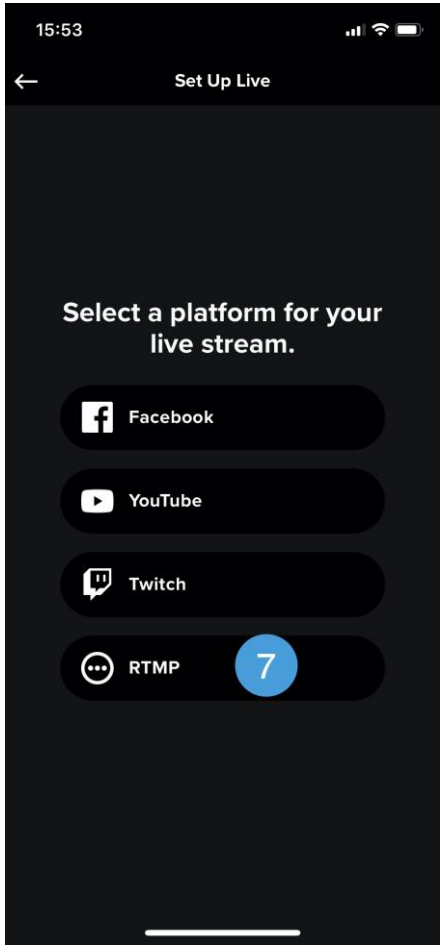


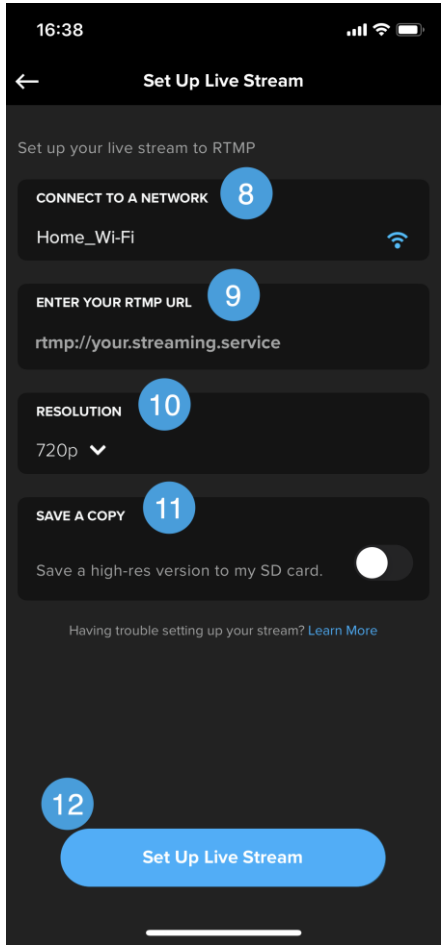




Appendix

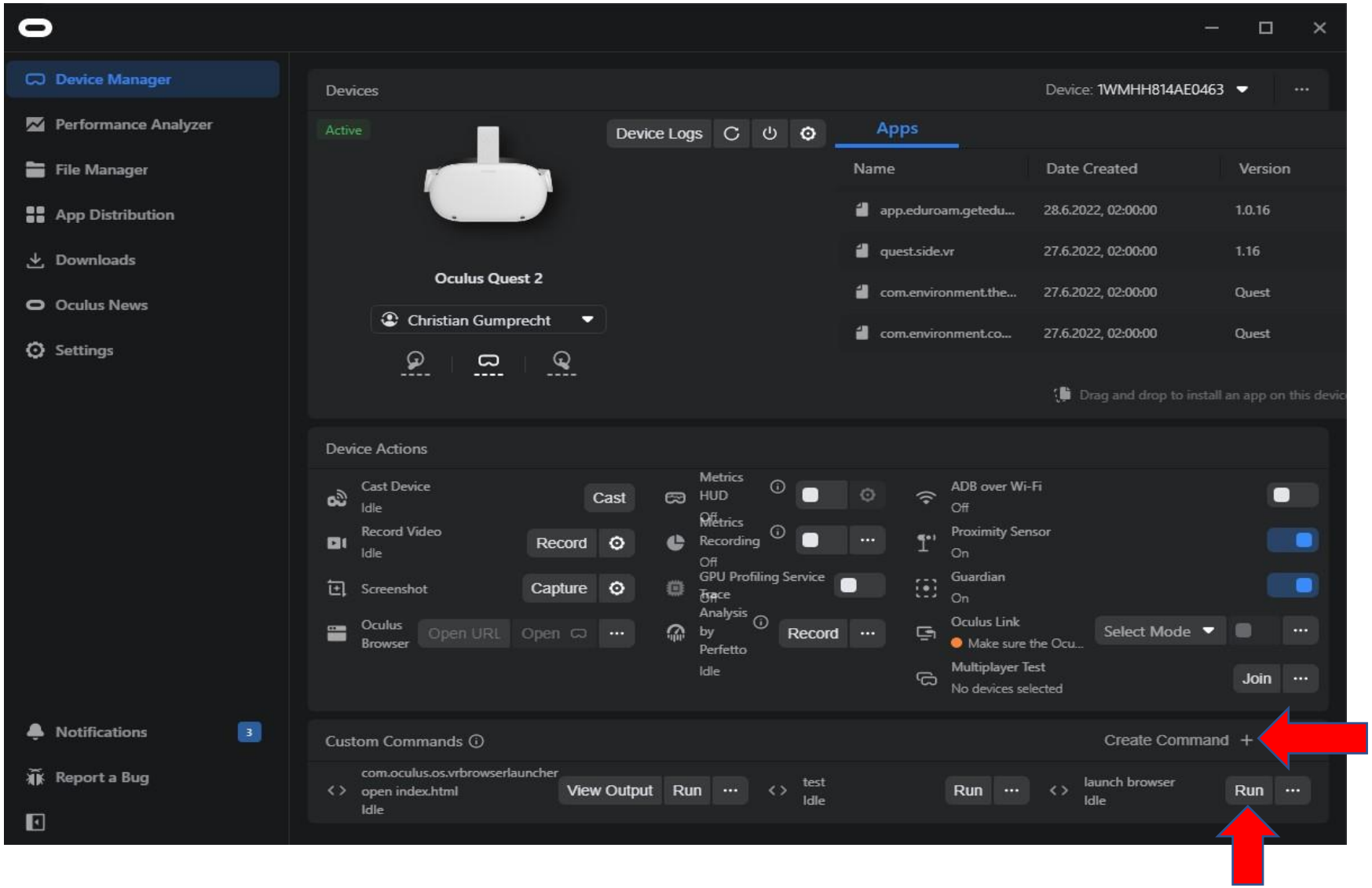
- 7. A new screen appears with a list of streaming platforms. Tap [RTMP].
- 8.Another screen appears to set up live stream. At the top, choose to connect a Wi-Fi network or your mobile hotspot.
- 9. Enter or paste your URL in the required field
- 10. Choose the resolution you want: 480p, 720p or 1080p
- 11. Choose your save settings
- 12.Tap [Set Up Live Stream].





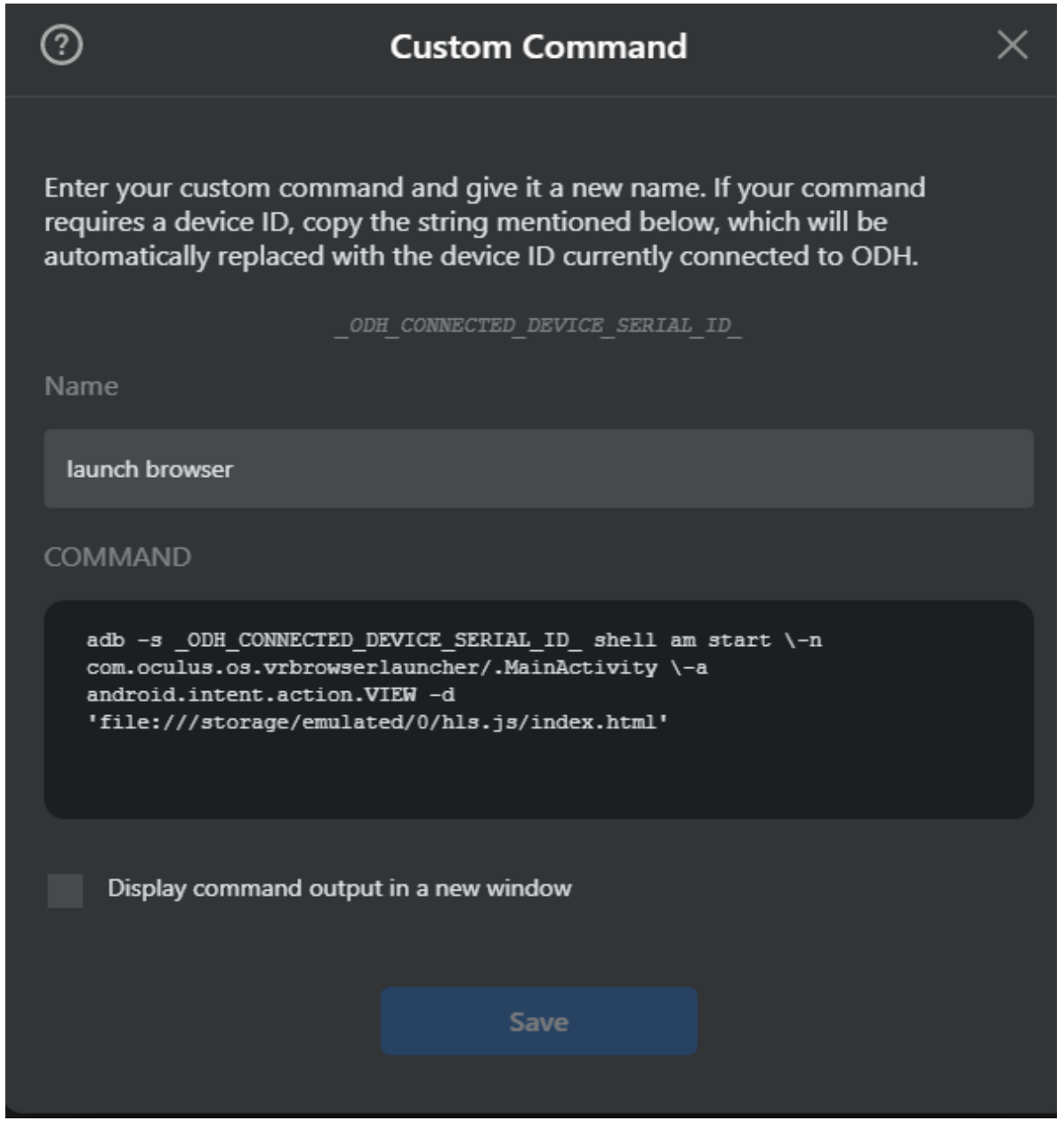
Appendix

- 13. Press [Create Command] on the Oculus Dev Hub



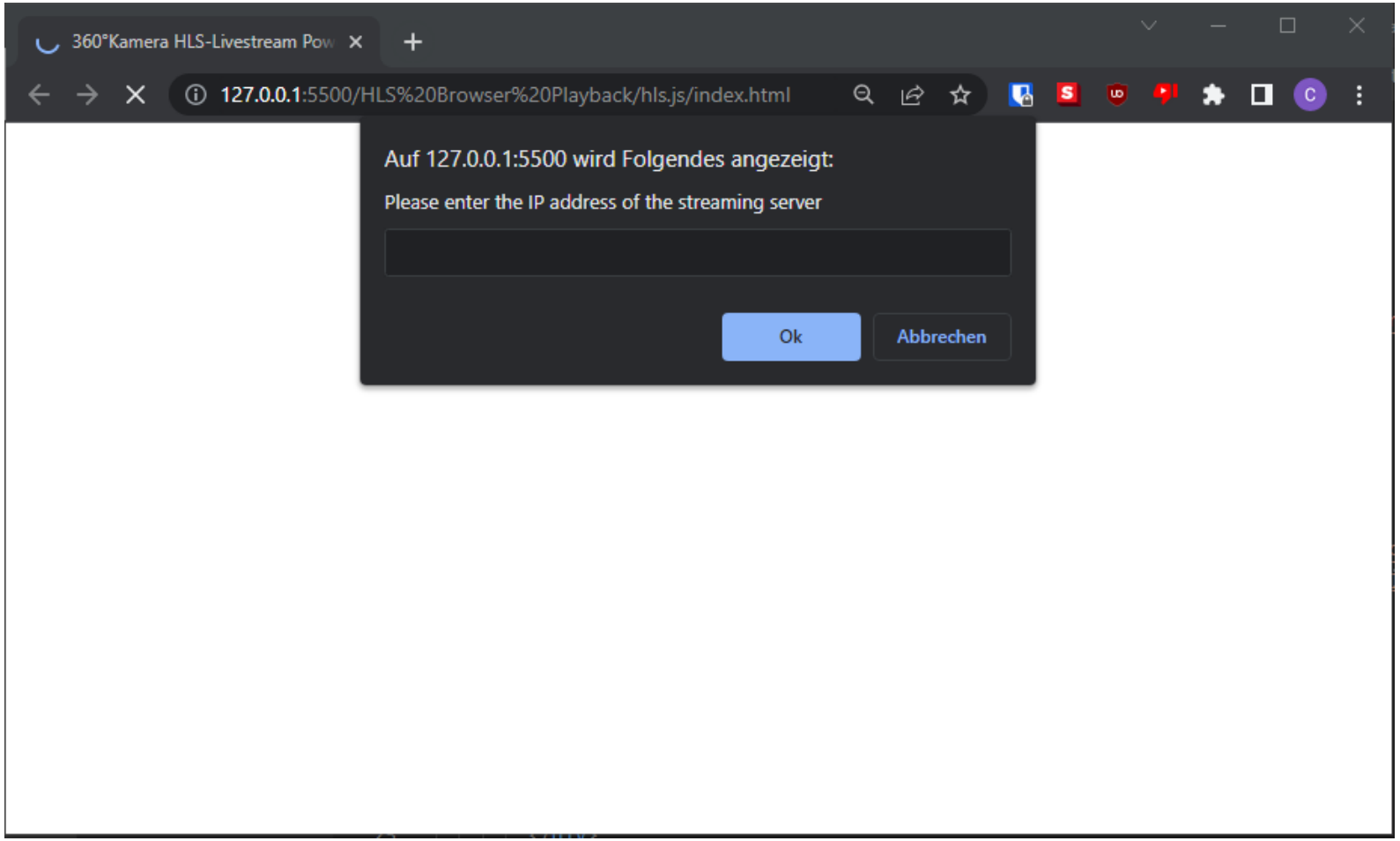
Appendix

- 14.
  Run Custom adb command with the right path to open the website on the Meta Quest2.



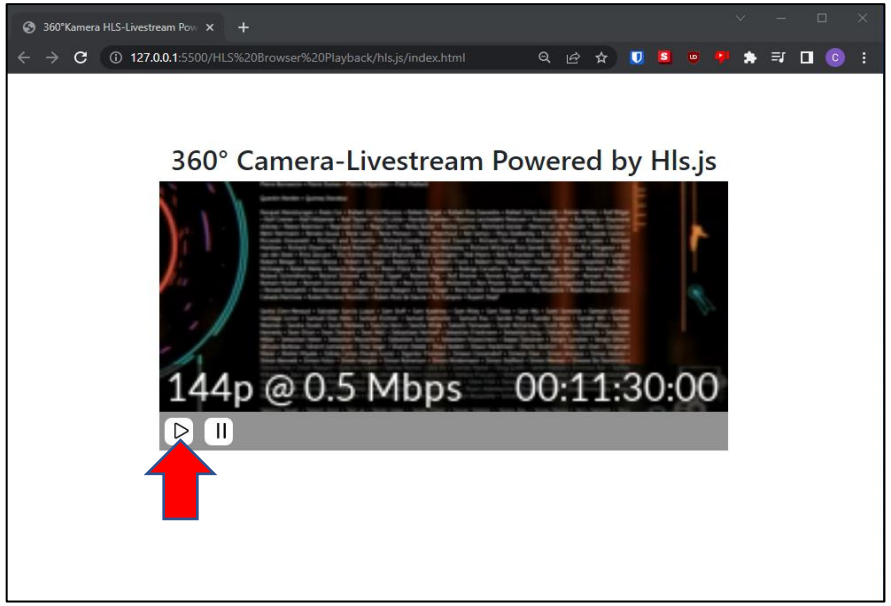
Appendix

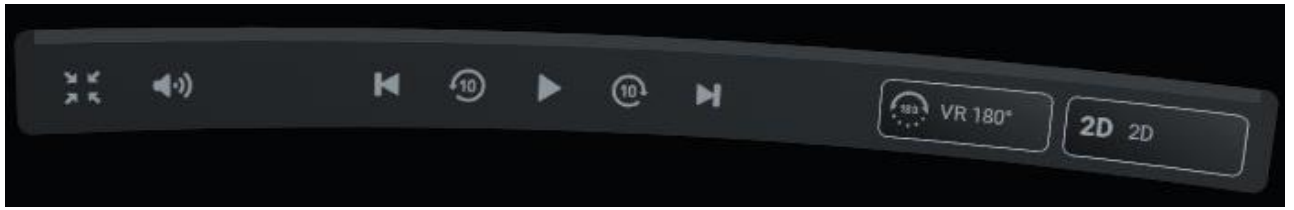
- 15. Enter the IP Adress of the stream server, so the website can acess the HLS stream



Appendix

- 16. Press on play to start stream and to go into Fullscreen VR-Mode
- 17. Press on play to start stream and to go into Fullscreen VR-Mode
- 18. Chose desired view format, depending u pon the video





Appendix

Our project is about how to stream 360° video from a 360° camera to oculus browser. 360° video is one of the most powerful content formats for user engagement. It transports viewers to the center of a narrative, giving them a highly immersive experience of a real-world or custom environment. One of our goals is to provide people with an opportunity to view in all directions (360 degrees) from the camera's point of view, using a head mounted device. The other goal is to make it possible, that people can enjoy a live broadcast with families, friends and thousands of strangers at the same time.

Therefore, we developed this prototype: First, we connect 360° camera with a phone, then we set up a live stream on the phone and entered a RTMP-URL. Using a real time video server supporting RTMP->HLS conversion, we then send the live stream to the network. Then Oculus quest could then access that stream (HLS with JS library).

After testing our product, we found out that our product works pretty well. We have a high resolution and good quality for the video. But there are still problems with it: when we enter a live stream mode, GoPro Max doesn’t allow us to do a 360 video. Therefore, our next step is to find a 360° live camera and stream 360° live video with our product.

Sources

Agora WebXR & A-Frame

- https://www.agora.io/en/blog/build-a-webar-live-video-streaming-web-app/
- https://github.com/digitallysavvy/AgoraWebXR
- https://webvr.directory/
- https://github.com/aframevr/aframe

GoPro Pictures

- https://de.gopro.com/help/articles/block/getting-started-with-live-streaming?sf92792173=1
- https://www.proshop.de/Videorekorder/GoPro-MAX/2799261

HLS(image)

- https://www.okta.com/identity-101/hls-streaming/

Hls.js (image)

- https://hls-js.netlify.app/api-docs/

Hls.js test-site/demo

https://hls-js.netlify.app/demo/

Hls.js (Javascript library)

- https://github.com/video-dev/hls.js/

HTML(image)

- https://ru.w3docs.com/uploads/media/book_gallery/0001/02/849d4286475e04155fd5f21861f16f53db95

HTML & Javascript

- https://www.w3schools.com/

Sources

Javascipt (image)

- https://marcas-logos.net/wp-content/uploads/2020/11/JavaScript-logo-1024x640.png

Live stream from GoPro cameras to OBS studio

- https://www.youtube.com/watch?v=e328xxdbRyk
- https://www.youtube.com/watch?v=6sVs4PFdxPc

Meta Quest Browser (image)

- https://www.oculus.com/experiences/quest/1916519981771802/

Oculus for developers

- https://developer.oculus.com/documentation

RTMP Use case example (Youtube)

- https://www.pubnub.com/learn/glossary/what-is-rtmp/

SRS Arch (image)

- https://ossrs.io/lts/en-us/

360WebPlayer

- https://github.com/BIVROST/360WebPlayer
- https://beprosto.github.io/webxr-tutorial/

Thank you for your attention!