360° VIDEO CLOUD STREAMING & HTMLVIDEOELEMENT EXTENSIONS

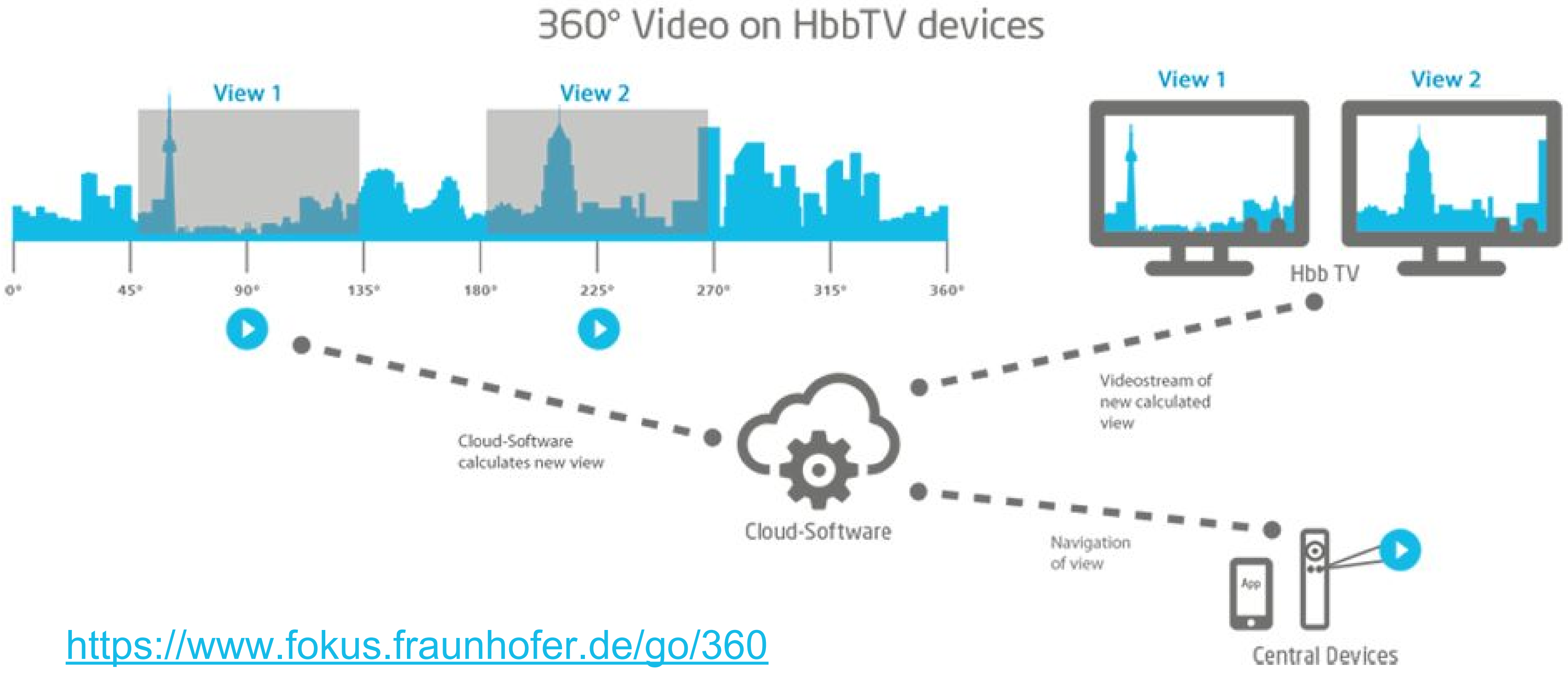
Louay Bassbouss | Fraunhofer FOKUS

W3C Workshop on Web & Virtual Reality, October 19-20, 2016; San Jose, CA, USA

making cities smart

Fraunhofer
FOKUS

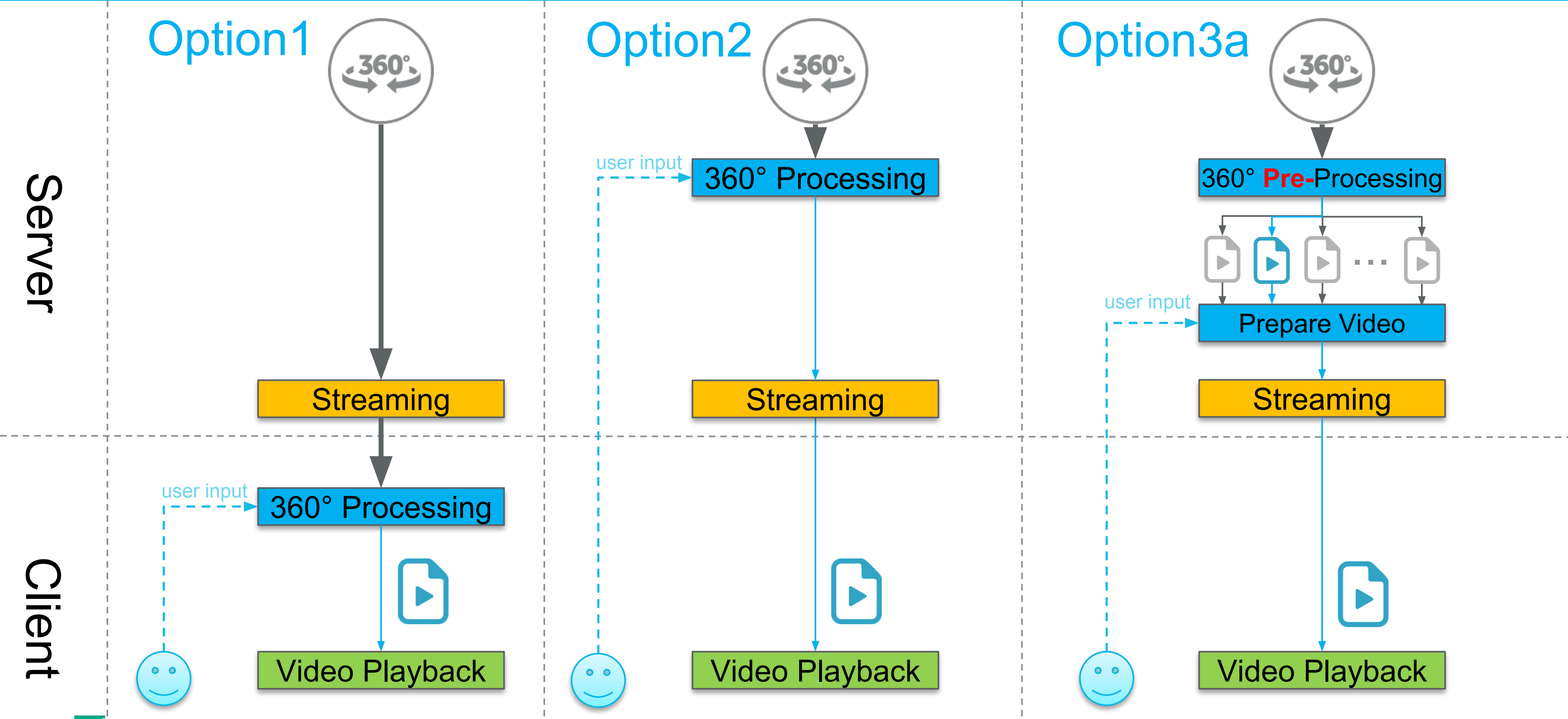
360° video cloud streaming



https://www.fokus.fraunhofer.de/go/360

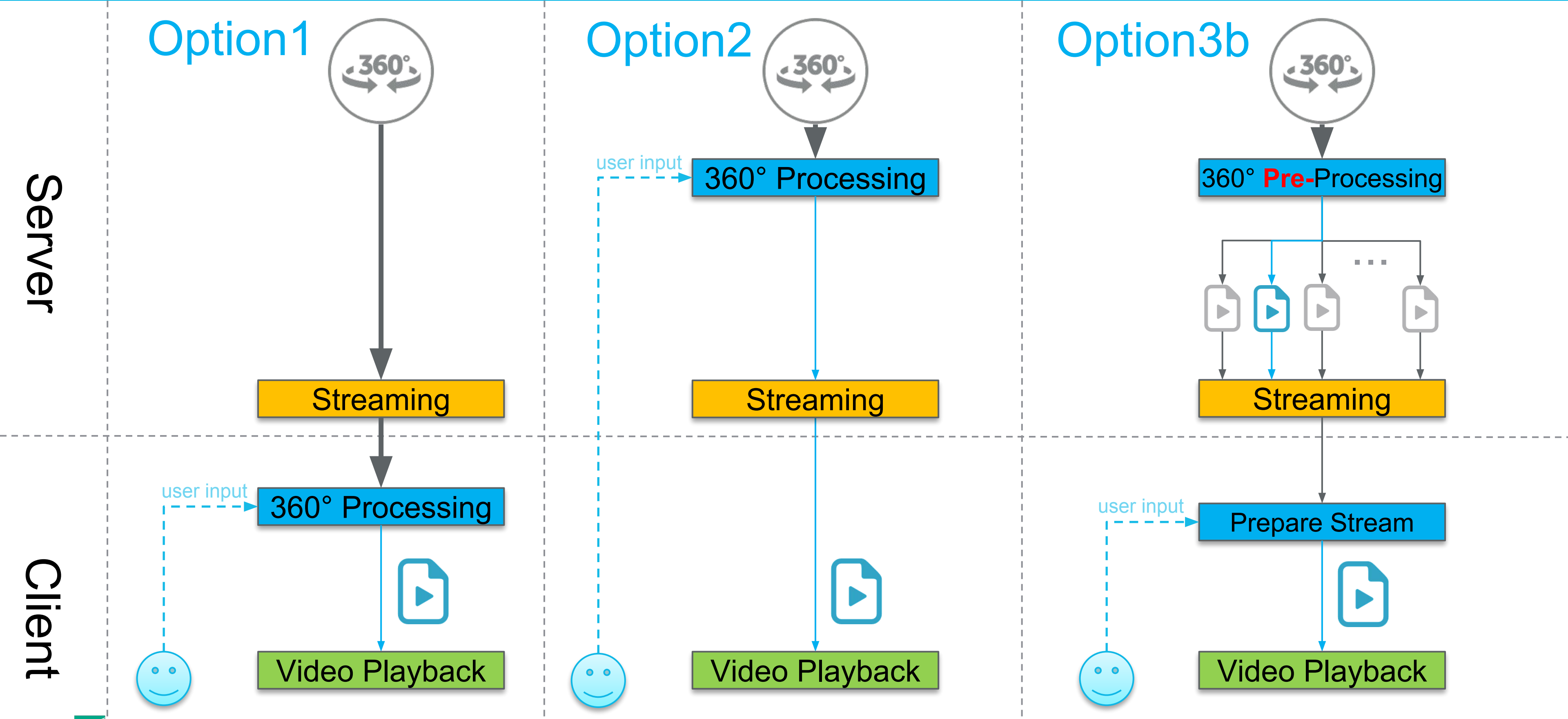
Fraunhofer FOKUS

360° STREAMING AND VIDEO PROCESSING OPTIONS



Fraunhofer
FOKUS

360° STREAMING AND VIDEO PROCESSING OPTIONS



ADVANTAGES AND DISADVANTAGES

	Option1	Option2	Option3a	Option3b
Additional Storage	No	No	Yes	Yes
360° Video Processing on Client	Yes	No	No	No
360° Video Processing on Server	No	Yes	No[1]	No[1]
Bandwidth	High	Low	Low	Low[2]
Motion-to-Photon Delay	Low	Medium[3]	Medium[3]	Medium[4]
CDN usage	Yes	No[5]	No[5]	Yes
Example Target Devices	Head Mounted Displays	Low Capability Devices e.g. HbbTV	Low Capability Devices e.g. HbbTV	Medium Capability Devices e.g. Chromecast
Interaction Types	- Motion Sensors - Touch/Mouse	- TV RC - Keyboard - (Touch/Mouse)	- TV RC - Keyboard - (Touch/Mouse)	- TV RC - Keyboard - Touch/Mouse

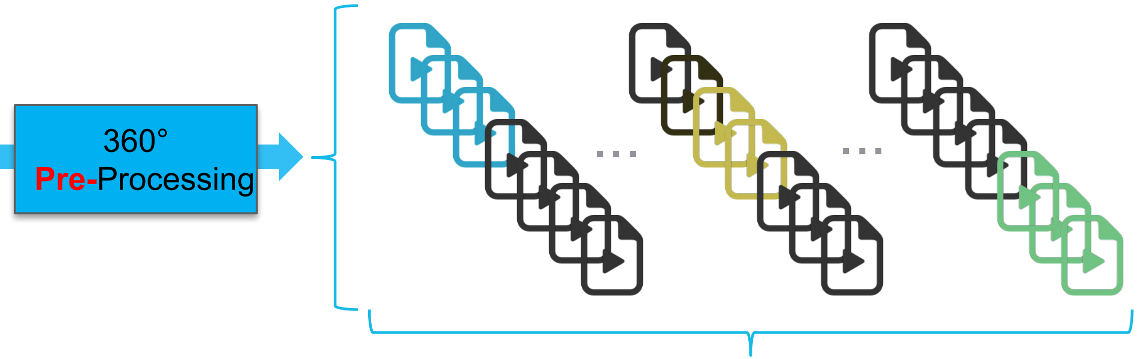
Fraunhofer FOKUS

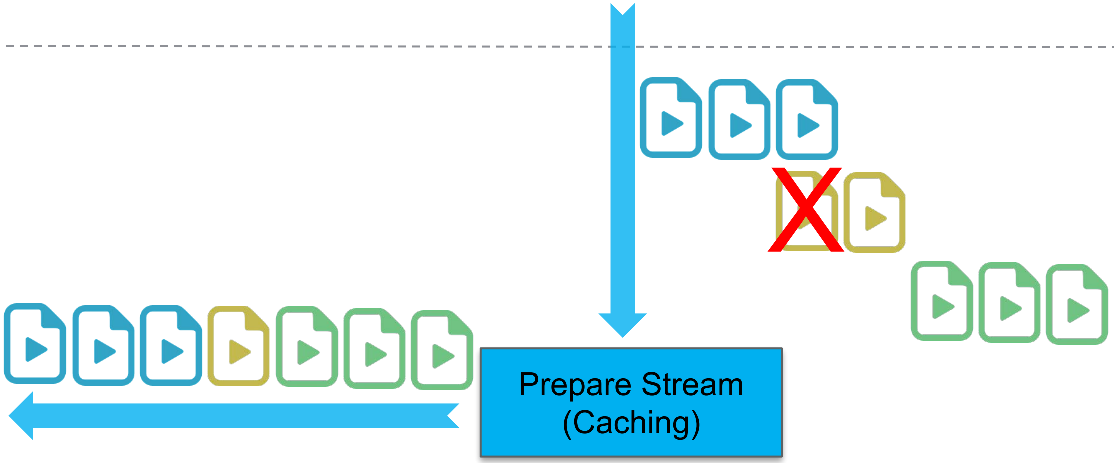
Notes for previous slide

1) The original 360° video will be pre-processed and FOVs are stored in separate files. There will be an overlap between the FOVs this is why there is a need for more storage but on the other side no video processing is needed.
2) since only one FOV is streamed to the client, no additional bandwidth is needed comparing to traditional video streaming. But it is still possible to pre-cache neighboring FOVs e.g. in lower quality to enable fast switch between FOVs in this case additional bandwidth is needed
3) Motion to Photon delay depends on network latency and protocol used to stream a single FOV (and Buffering on the Player).
4) Motion to Photon Delay depends on the caching strategy of the player.
5) it is difficult to use CDN since a persistent connection between client and server is needed (there is a session for each client)

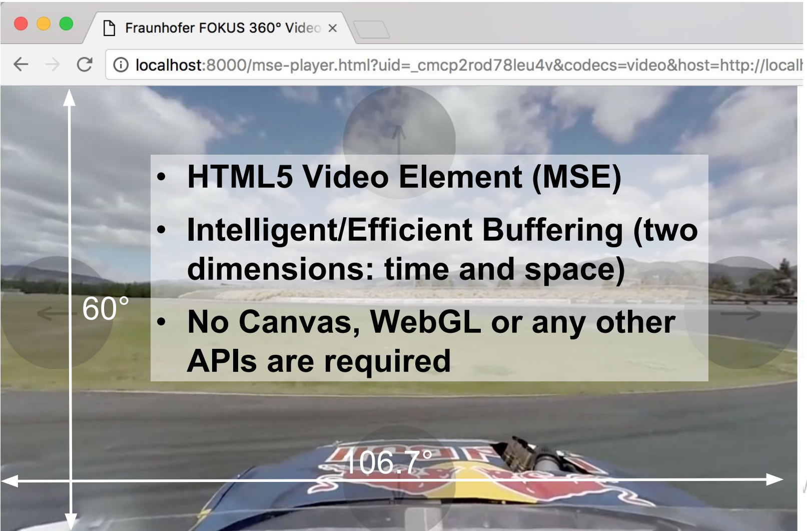
DEMONSTRATION (Option 3b)

4k origin 360° Video, 30fps, bitrate 40053 kb/s



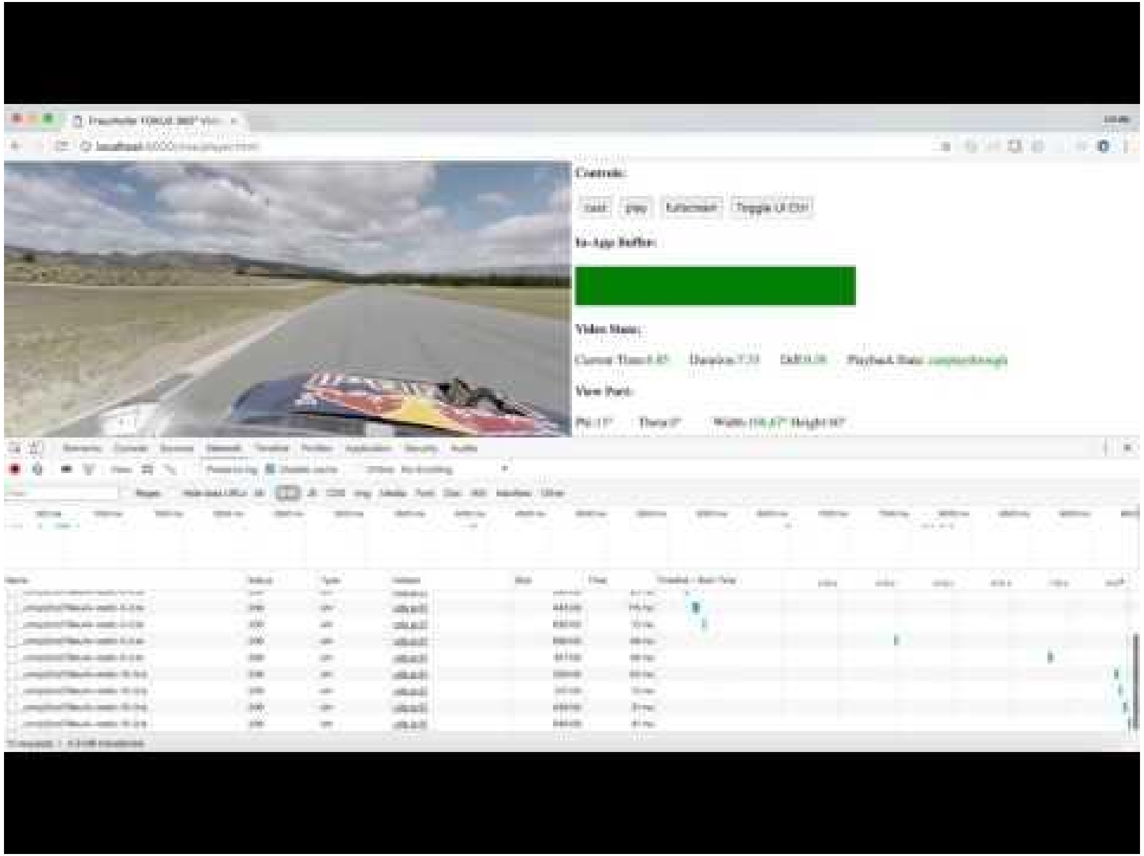


HD view port, 30fps, bitrate 2435 kb/s, segment=333ms


Fraunhofer FOKUS 360° Video
localhost:8000/mse-player.html?uid=_cmcp2rod78leu4v&codecs=video&host=http://localh
60°
106.7°

- **HTML5 Video Element (MSE)**
- **Intelligent/Efficient Buffering (two dimensions: time and space)**
- **No Canvas, WebGL or any other APIs are required**

Demo video



Fraunhofer FOKUS

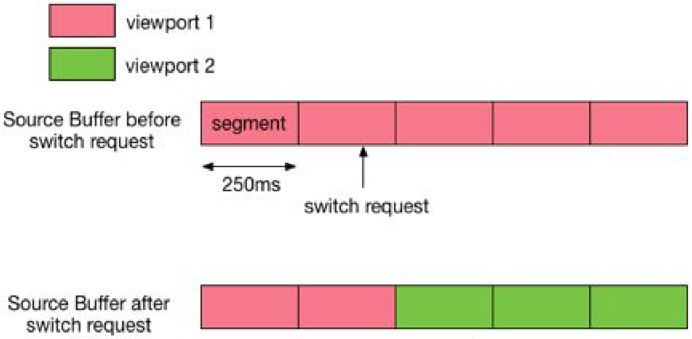
HTMLVIDEOELEMENT EXTENSIONS

- Two types of players:
  - Native 360° Video Player
  - Using MSE → do we need extensions for the MSE API?
- Native 360° Video Player:
  - The HTMLVideoElement plays 360° video natively. Set video.src={360_video_url} (or use <source element>)
  - The HTMLVideoElement needs to get all the metadata in order to render the view correctly.
  - New functions to set and get the FOV are needed
  - New events on start, during and after changing the FOV are needed
  - (Maybe also functions and events for Zoom in/out.)
  - Example:
    - video.setFOV(phi, theta, width, height)
    - video.onfovstart, video.onfovend, ...

Fraunhofer FOKUS

HTMLVIDEOELEMENT EXTENSIONS

- MSE 360° Player
  - Allows to implement different player algorithms similar for DASH on top of MSE
  - Available viewports can be described in a manifest (e.g. DASH SRD fields)
  - At the start of the playback the currently selected viewport is buffered. When the user triggers a switch request for a different viewport, already buffered segments are removed/replaced by segments of the new viewport.
  - Challenge:
    - How to reduce delay by switching between two viewports?


viewport 1
viewport 2
Source Buffer before switch request
segment
250ms
switch request
Source Buffer after switch request

Fraunhofer FOKUS