



360° VIDEO PLAYOUT FRAUNHOFER FOKUS

Stephan Steglich <stephan.steglich@fokus.fraunhofer.de>

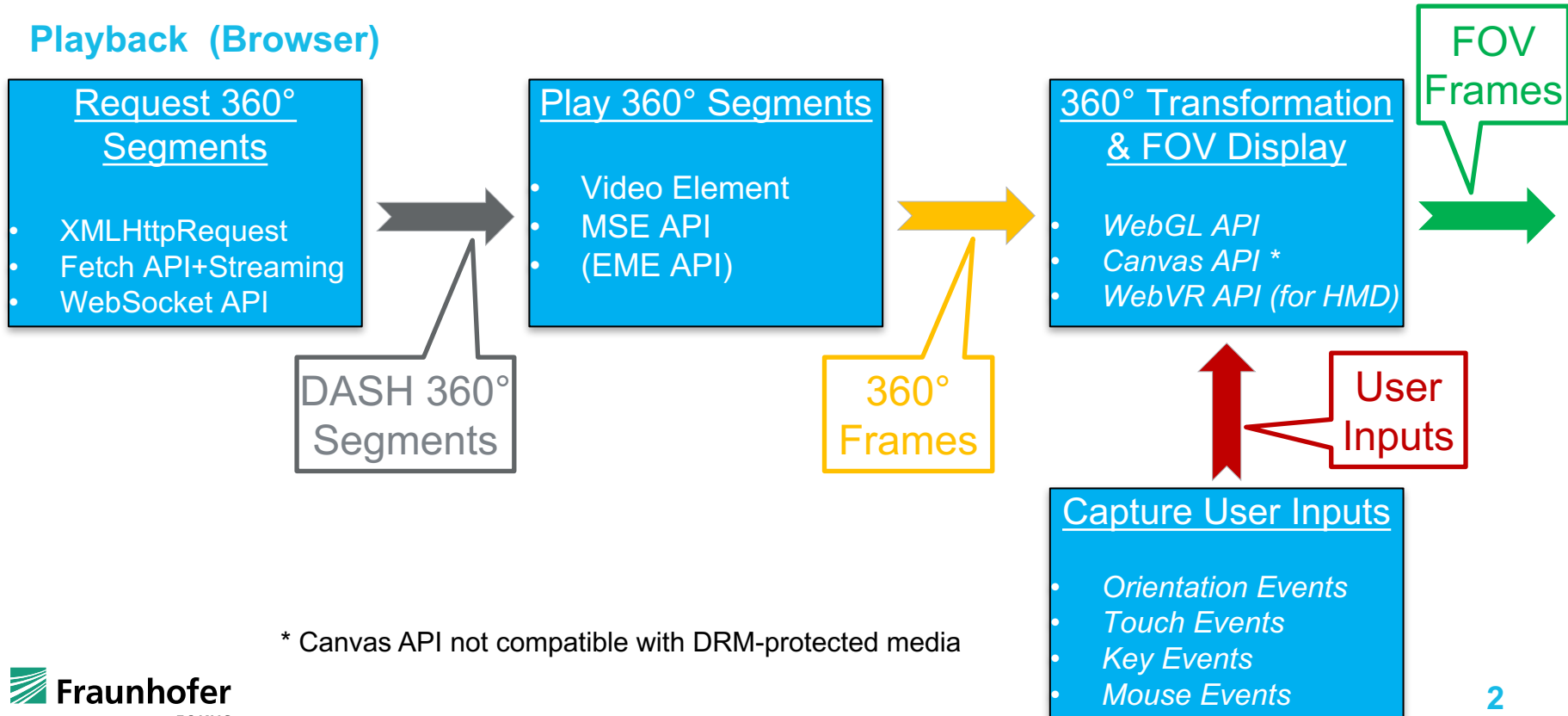
Louay Bassbouss louay.bassbouss@fokus.fraunhofer.de

Stefan Pham stefan.pham@fokus.fraunhofer.de

www.fokus.fraunhofer.de/go/360

360° CLIENT SIDE PROCESSING

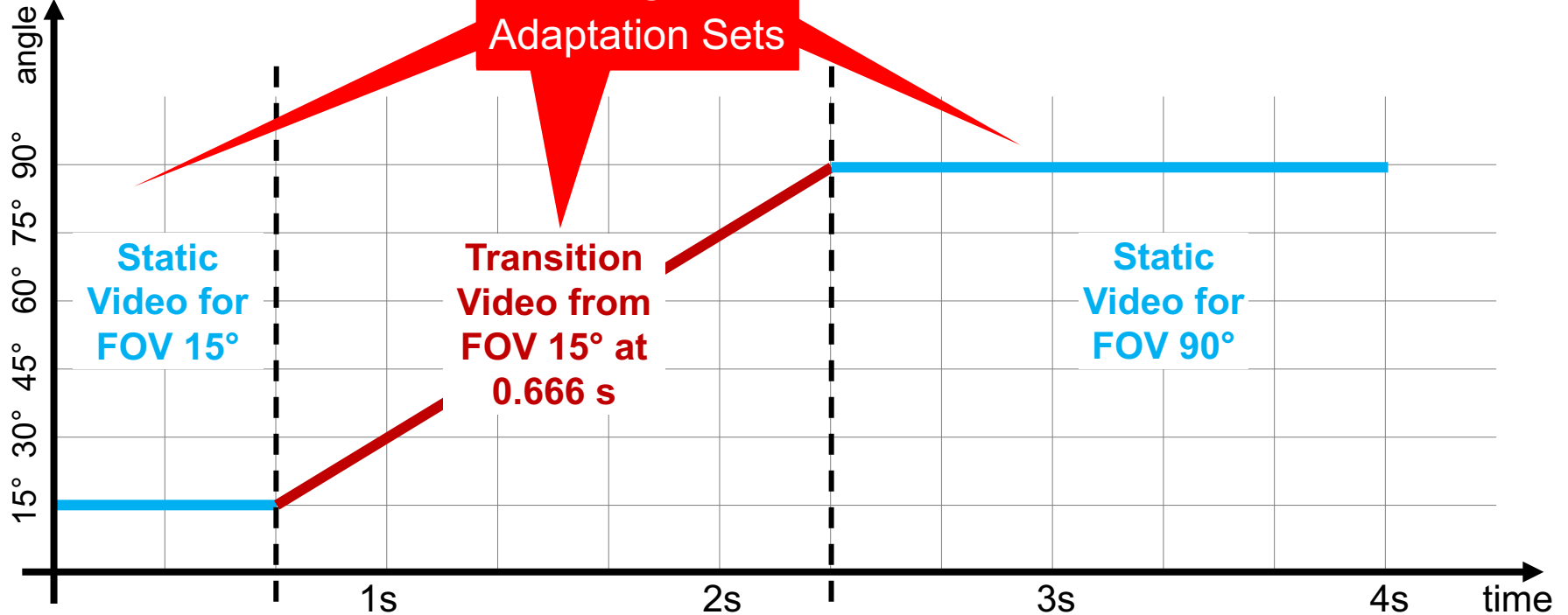
Playback (Browser)



* Canvas API not compatible with DRM-protected media

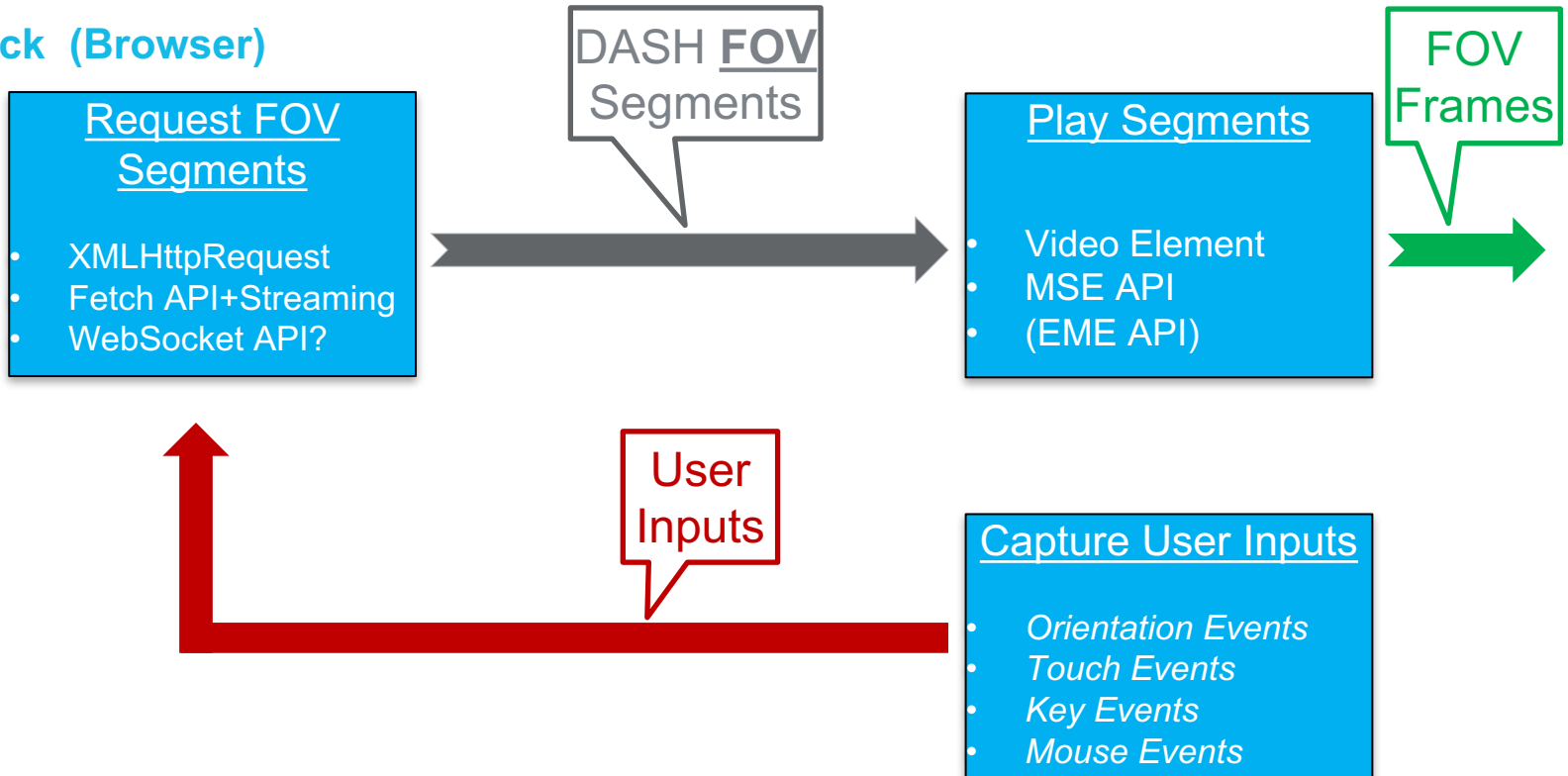
360° SERVER SIDE PRE-RENDERING

How it works?



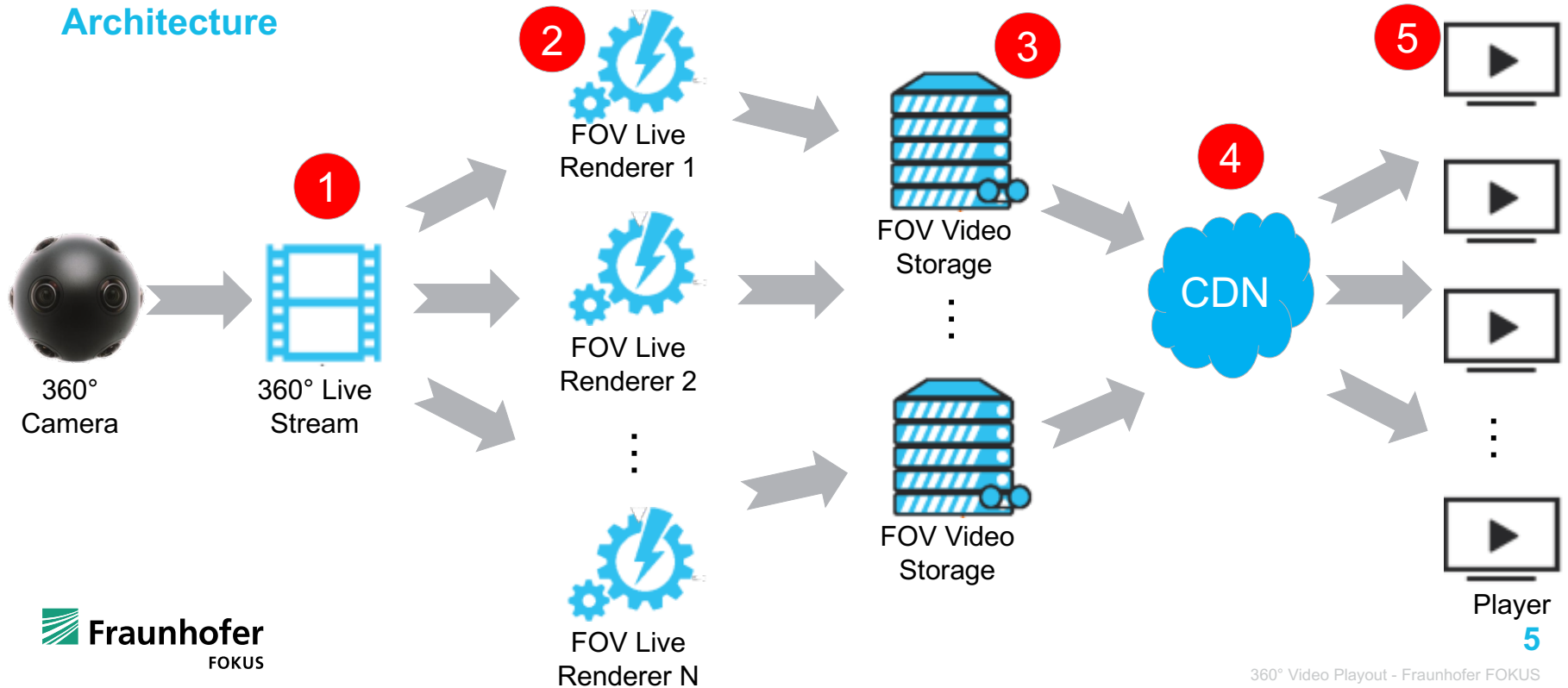
360° SERVER SIDE PROCESSING

Playback (Browser)



360° SERVER SIDE PROCESSING

360° Server Side Pre-rendering (Fraunhofer Cloud Payout) – Live Streaming Architecture



360° SERVER SIDE PROCESSING

360° Server Side Pre-rendering (Fraunhofer Cloud Playout) – Live Streaming Architecture

1. The input is a Equirectangular Video Stream coming from the camera after stitching e.g. as RTMP stream (or any other suitable live streaming protocol).
2. The Camera stream will be sent to a number of “FOV Renderers” running on GPU machines in the cloud. Supposing that each machine can render **N** FOV videos in parallel and in real-time, and the total number of FOVs is **M**, then the number of required Rendering machines is **M/N**. For example an AWS [p2.xlarge instance](#) can render **N=4** FOVs from a 4K input equirectangular video with 30FPS in parallel and real-time. If the number of FOV videos (static and dynamic) is **M=45** (For **30°** steps between two horizontal FOVs), then the number of required Rendering instances is around **12**
3. The Output FOV video segments (output of FOV renderers) are stored together with a manifest file somewhere in a cloud storage locations.
4. The FOV videos and will be delivered to the Client via CDN (low latency is important)
5. The client implements the entire player logic and requests the FOV video segments over the CDN using HTTP and plays back the segment using MSE (in Browser Environment).

IMPROVEMENTS FOR W3C APIS

- MSE deficiencies:
 - Internal buffer (the buffer of the underlying Media Player) expects an undefined # frames across browsers, which causes additional delay when appending new frames/segments
 - Even after clearing the SourceBuffer (e.g. replacing segments), old frames/segments get played back
 - Multiple SourceBuffers attached to single MediaSource and switching between the SourceBuffers causes issues
 - Replacing segments (segment append mode) in a single SourceBuffer does not work reliably
 - As a workaround we buffer segments into JS and copy just-in-time into SourceBuffer
- WebVR/EME:
 - Using WebGL and Canvas for video transformations is not possible with DRM-protected content => need a secure media path for transformations and possibly API extensions

More information and Demos at:
www.fokus.fraunhofer.de/go/360

Thank you for listening!
Questions?



*Fraunhofer Institute for Open
Communication Systems*

*Kaiserin-Augusta-Allee 31
10589 Berlin, Germany*

www.fokus.fraunhofer.de



Dr.-Ing. Stephan Steglich

Director
Future Applications and Media (FAME)
Fraunhofer Institute for
Open Communication Systems FOKUS

Kaiserin-Augusta-Allee 31 | 10589 Berlin | Germany
Phone +49 30 3463-7373 | Mobile +49 175 527 11 46
stephan.steglich@fokus.fraunhofer.de