© Philipp Plum/ Fraunhofer FOKUS

Fraunhofer FOKUS

Fraunhofer FOKUS Institute for Open Communication Systems

HbbTV meets dash.js - Optimizing broadcast-broadband ad-insertion and WebVTT rendering

Daniel Silhavy

About me

- **Daniel Silhavy (Fraunhofer FOKUS)**

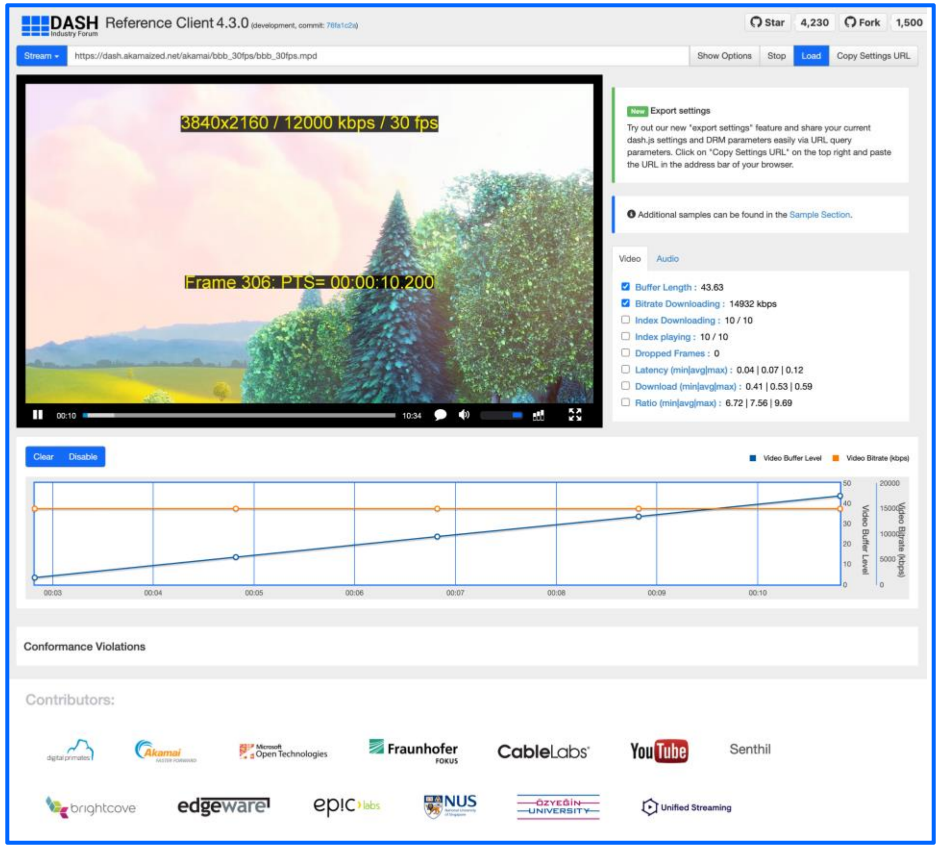
- **Area of expertise**
  - (5G) Media Streaming
  - Video Encoding,
  - Media Player Development
  - Standardization

- **Related Open-Source Projects**
  - Lead Developer of the dash.js project
  - 5G-MAG Reference Tools Development Team Coordinator
  - Joint Conformance Project (JCCP) Development Coordinator

- **Contact**
  - Email: daniel.silhavy@fokus.fraunhofer.de
  - LinkedIn: https://www.linkedin.com/feed/

dash.js : Open-Source DASH Player

- **dash.js** is the official reference player by the **DASH Industry Forum** for playback of MPEG-DASH content
- Maintained by Fraunhofer FOKUS, community driven development
- Used as a **reference client** for standardization, foundation for **production-grade** video applications and for **research** purposes e.g. implementing new ABR algorithms
- Included in the HbbTV Reference Application for Type-3 playback: https://github.com/HbbTV-Association/ReferenceApplication/tree/master/src/videoplayer
- Open-source project on Github - https://github.com/Dash-Industry-Forum/dash.js/ , last released version 4.7.2
- Written in JavaScript uses the W3C **Media Source Extensions (MSE**) and Encrypted Media Extensions (EME)
- Various features including support for ABR, multiperiod, DRM, MPD patching, Gap handling, CMCD, CMSD, Content Steering, CMAF low latency, various subtitle formats (TTML, IMSC1, WebVTT) and many more.



https://dash.akamaized.net/akamai/bbb_30fps/bbb_30fps.mpd

Export settings

Try out our new "export settings" feature and share your current dash.js settings and DRM parameters easily via URL query parameters. Click on "Copy Settings URL" on the top right and paste the URL in the address bar of your browser.

Additional samples can be found in the Sample Section.

☑ Buffer Length : 43.63
☑ Bitrate Downloading : 14932 kbps
☐ Index Downloading : 10 / 10
☐ Index playing : 10 / 10
☐ Dropped Frames : 0
☐ Latency (min|avg|max) : 0.04 | 0.07 | 0.12
☐ Download (min|avg|max) : 0.41 | 0.53 | 0.59
☐ Ratio (min|avg|max) : 6.72 | 7.56 | 9.69

Conformance Violations

MSE and EME on HbbTV Terminals

W3C Media Source Extensions API

- https://w3c.github.io/media-source/
- Extends the HTML5 Media Element to allow JavaScript applications to generate media streams for playback of adaptive streaming content
- Officially supported since HbbTV version 2.0.3
- But we observed **working MSE based playback on older devices e.g. Samsung 2017 with HbbTV 1.5**

W3C Encrypted Media Extensions API

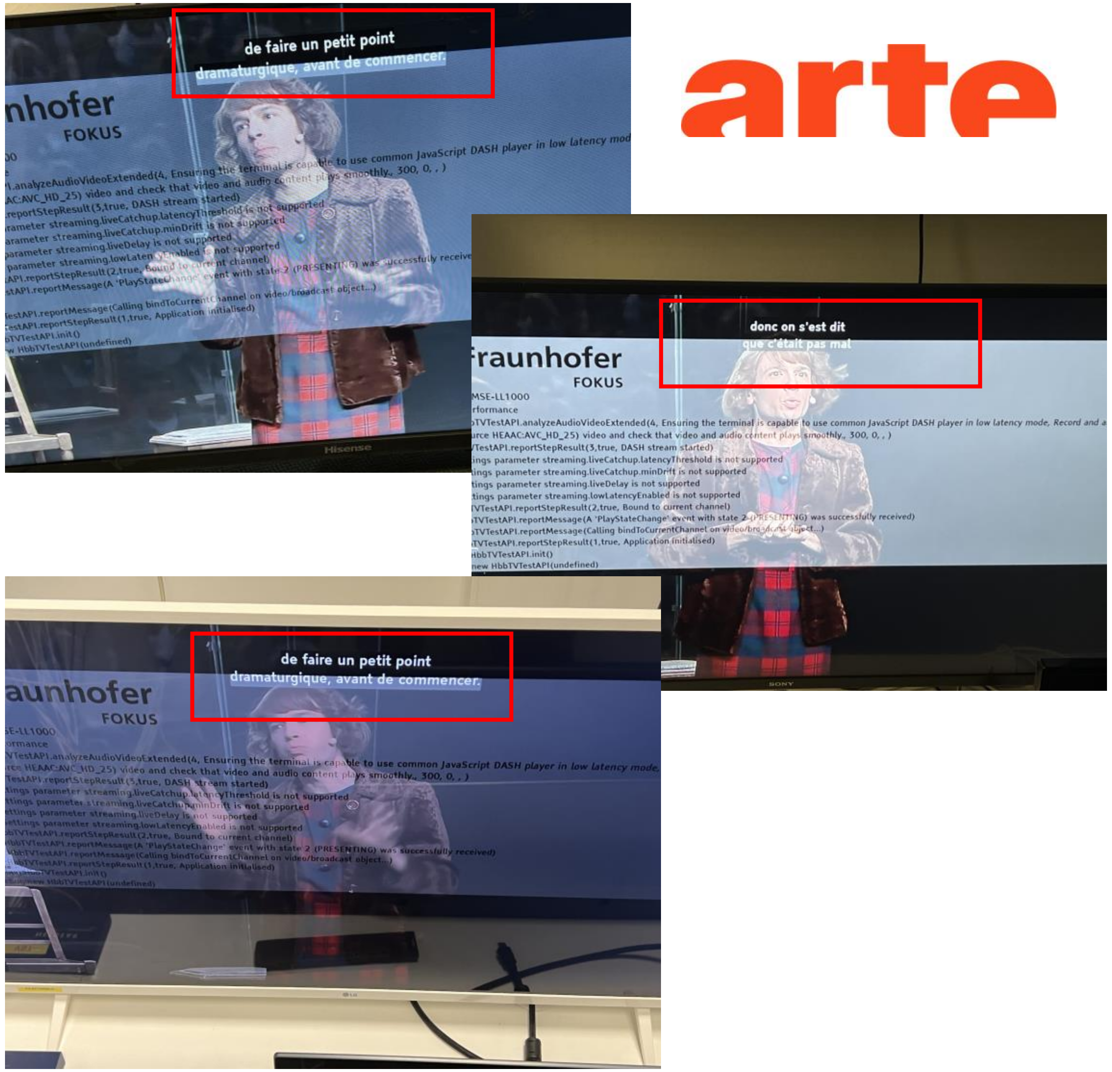
- https://www.w3.org/TR/encrypted-media/
- Extends the HTML5 Media Element providing APIs to control the playback of DRM protected content
- Officially supported since HbbTV version 2.0.1 (Type-1)
- Complements the MSE for Type-3 playback

Why use dash.js on HbbTV Terminals?

- ✓ Extensibility: Open-source codebase that can easily be extended (e.g. add support for additional subtitle formats).
- ✓ Consistency: Consistent behavior across different HbbTV Terminals. Type-1 player can differ in terms of supported DASH features and attributes.
- ✓ Future-proof: dash.js implements latest features such as CMCD and Content Steering. Updates for Type-1 players can be delayed.
- ✓ Controllability: Full control over crucial parts of the media player such as the ABR behavior.
- ✓ Customizable: Custom settings for specific devices are possible, e.g. limit playback to a certain resolution or bitrate.
- ✓ Robustness/Maintainability: Potential problems in the content can be solved on the client side.

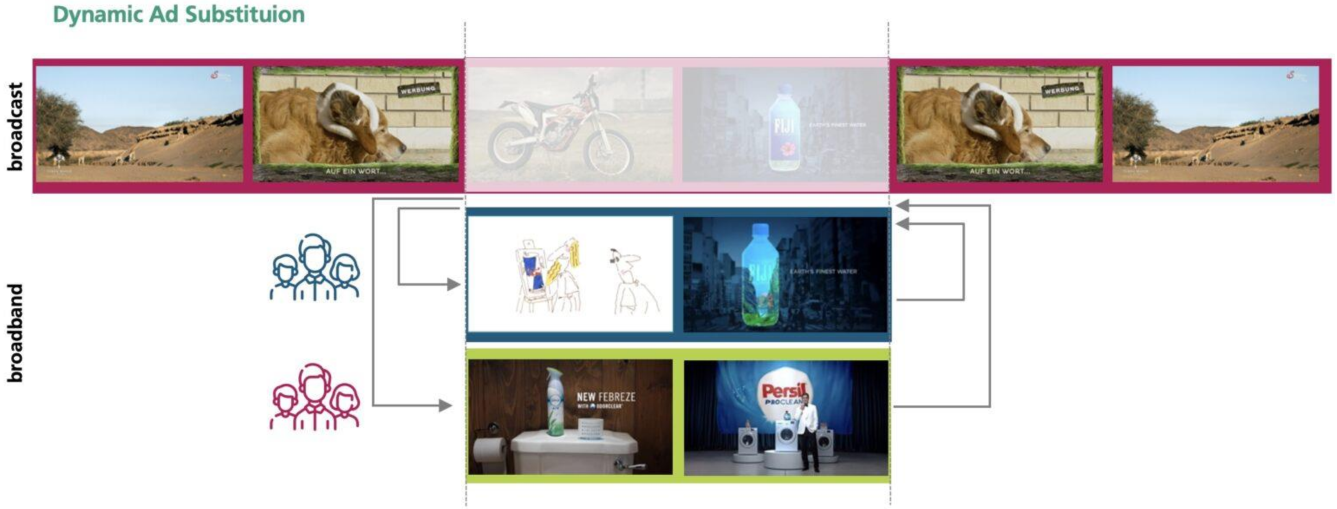
dash.js - WebVTT support on HbbTV terminals

- Joint project together with **ARTE**
- **Goal:** Playback of MPEG-DASH streams with WebVTT subtitles on HbbTV terminals using dash.js
- **Problem**:
  - dash.js was relying on native rendering of WebVTT subtitles.
  - Most HbbTV devices do not render WebVTT tracks natively (HbbTV mandates support for EBU-TT-D). Not even the necessary events are dispatched.
- **Solution**:
  - Trigger onCue events manually.
  - Manage active and hidden tracks.
  - Use library (vtt.js) for rendering/styling.
- Try it out yourself: https://reference.dashif.org/dash.js/nightly/samples/captioning/vttjs.html

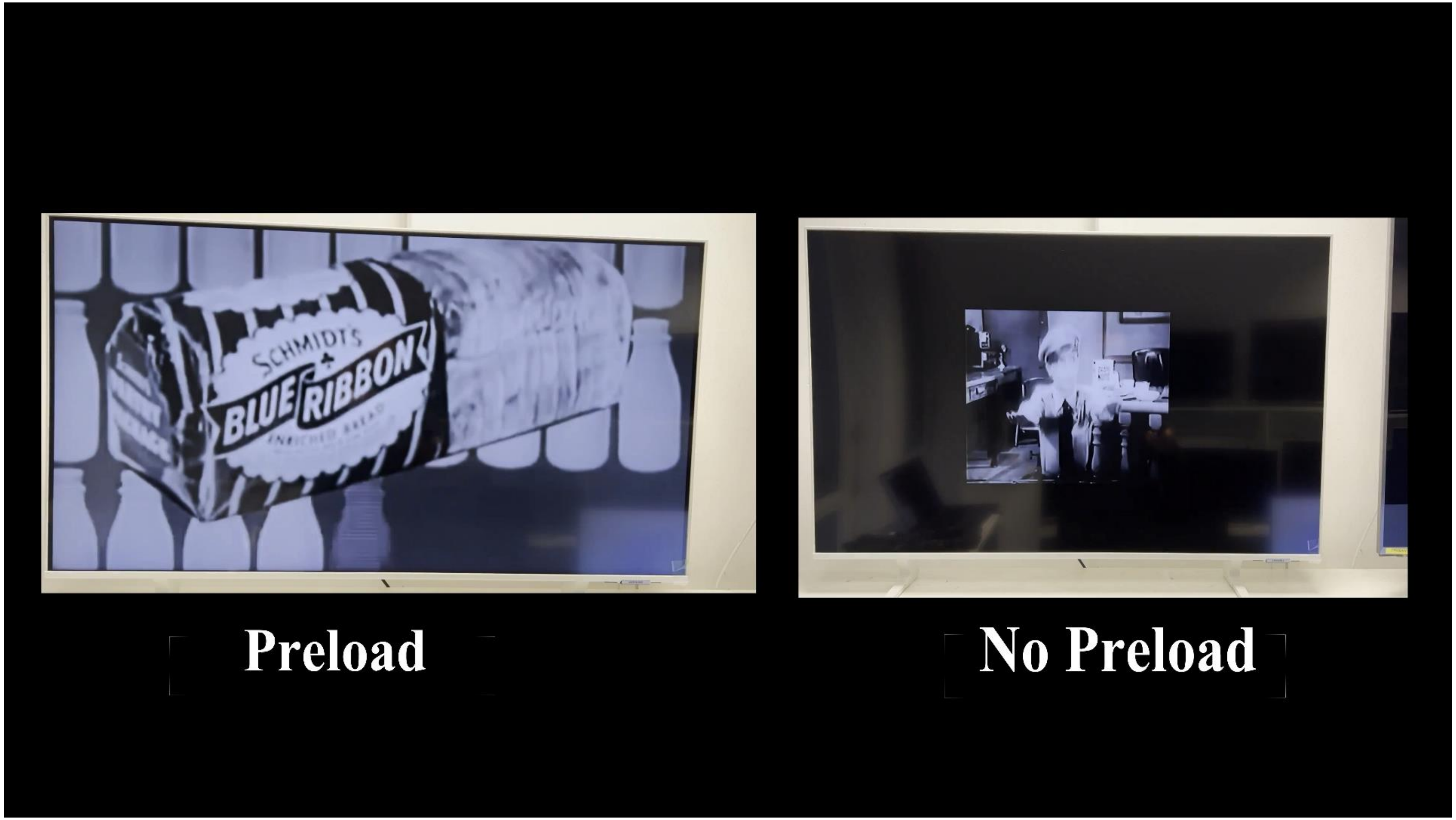


dash.js - Broadcast – Broadband Ad Substitution

- Joint project together with **Google** and **Mediaset Spain**
- **Problem**: Some platforms like HbbTV terminals have only a single decoder. It is not possible to initialize MSE based playback while the broadcast content is being decoded and rendered.
- **Goal**: Support Broadcast-Broadband ad insertion on HbbTV terminals. Segments should be prebuffered for a seamless transition between main content (broadcast) and ad content (broadband)
- **Solution**: Virtual buffer that is emptied once MSE is attached to video element
- Try it out yourself: https://reference.dashif.org/dash.js/nightly/samples/advanced/preload.html



dash.js - Preload Demo Video



Contact

Daniel Silhavy
Email: daniel.silhavy@fokus.fraunhofer.de
LinkedIn: https://www.linkedin.com/in/daniel-silhavy-21650a129/

Fraunhofer FOKUS
Institute for Open Communication Systems
Kaiserin-Augusta-Allee 31
10589 Berlin, Germany
info@fokus.fraunhofer.de
www.fokus.fraunhofer.de