

Use Cases and Requirements for MSE v2 and EME v2

CTA Wave Project

1 Introduction

This document captures the WAVE project use cases for MSE v2 and EME v2 that will be submitted in response to the “Call for MSE V2 and EME V2 Use Cases and Requirements” found at <https://lists.w3.org/Archives/Public/public-html-media/2016Jul/0017.html>.

Some of these use-cases and requirements will duplicate those already in the wiki pages however it is believed that multiple voices supporting them is useful to reinforce their importance.

Use cases and requirements have been separated into MSE and EME as the W3C have asked for them to be submitted separately.

2 MSE Use Cases

2.1 Client-side advert insertion into soft-parted on-demand content

A video content provider funds their on-demand content service through adverts. These are inserted before (“pre-roll”), during (“mid-roll”) and after (“post-roll”) the content. The ads are sourced from third parties and the video content provider has no control over their video format (i.e. spatial resolution and frame rate), encoding and packaging. Specifically, the content and the ads may differ by the following;

- Program is at 1920x1080/24p, ad insert is at 720x1280/59.94
 - Codec (e.g. content is HEVC but ads are AVC, e.g. content is Dolby but ads are HE-AAC, e.g. content is MPEG-H but ads are HE-AAC).
- Number of audio channels (e.g. content is 5.1 but ads are stereo)
 - Number of tracks (e.g. content contains multiple audio languages and/or accessibility tracks but ads only contain one audio language, e.g. content contains multiple subtitle/caption tracks but ads only contain one or even none)

The locations in the content where mid-roll ads are to be inserted are not fixed, the content item is a single file (or equivalent) and not split into separate files at ad insertion points (“hard-parted”).

The video content provider’s web app decides which ads are to be inserted by making a XMLHttpRequest call to an ad decision server. This returns information about appropriate ads to play (e.g. in the VAST format - https://en.wikipedia.org/wiki/Video_Ad_Serving_Template). This is equally applicable to pre-roll, mid-roll and post-roll ads.

The video content provider's web app will play a pre-roll ad and then the content. As the current playback position approaches a point where mid-roll ads could be inserted, the web app will decide which ads are to be inserted (if any). If ads are to be inserted then content playback will stop at the last frame of the content before the ad insertion point. The ad(s) will be played and then content playback will resume. User choices such as audio or subtitle/caption track selections are restored.

Ideally the transitions content to ads, between ads and ads back to content should be seamless. It is believed users will tolerate perhaps 250ms of transition but the solution should permit good quality implementations to be seamless. Since the content playback is on-demand, pausing playback for a short transition period has no technical consequences – only the negative user experience. (Note for on-demand content, the timeline can be paused for this ≤ 250 ms).

Content may or may not be encrypted. In this use case ads are not encrypted. In the future we may provide a scenario where ads are encrypted.

NOTE: In this use-case, the content is packaged in ISOBMFF as profiled by CMAF [1]

[1] <http://mpeg.chiariglione.org/sites/default/files/files/standards/parts/docs/w16186.zip>

2.2 Playback of Live content

Basic playback

A video content provider provides the same live content via the web as via broadcast and promotes it to consumers as such. The user starts the video content provider's web app and chooses which broadcast TV channel to watch. The web app starts presenting that channel from the live edge. During playback the user can;

- pause live content and resume (resuming may not be from the same place in the content if the content is paused for longer than it is buffered in the network)
- choose different languages for audio or subtitles
- change accessibility settings for audio or subtitles/captions
- ask for more information about the program currently showing

NOTE: In this use-case, the content is packaged in ISOBMFF as profiled by CMAF [1].

Ad insertion with live content

The live content being presented (see above) will contain adverts. The video content providers app has the opportunity to replace those adverts with other adverts. This part of the use-case is the same as for ad insertion into on-demand content except as follows;

1. Consumers have higher expectations of live TV than of on-demand and transitions from content to ads, between two ads and from ads to content that would be acceptable with on-demand content will not be acceptable in content that is promoted to the user as broadcast.
2. The timeline is always advancing for live content so any transition periods reduce the time available for adverts.

3. On-demand content has a clear media timeline starting at zero and running to the end of the content item. Locations where ads can be inserted can be expressed in terms of this timeline. Live content has a much less clear concept of a timeline. In the live use-case, locations where ads can be inserted will typically be announced by messages in the stream.

Synchronisation of Web Application to Live Content

The video content provider has an operator providing additional information about the live content (see above) and synchronised with it. Examples include pressing a button to send a pre-defined message when something happens and/or sending a message that contains additional information about something that has happened. This information is delivered to the video content provider's web application. The web application takes some action based on the information that is synchronised with the consumer seeing the corresponding video and hearing the corresponding audio. This action would typically be showing something to the user – something that may or may not include information from the message.

The content may be encrypted or not encrypted and may change between the two (see EME use-cases).

3 EME Use Cases

3.1 Encrypted live content

A video content provider provides the same live content via the web as via broadcast and promotes it to consumers as such (see MSE use-case for more details). The content is encrypted. The content may contain adverts (see MSE use-case for more details). These adverts will most likely not be encrypted. The decryption keys are periodically updated ('key rotation') and the entitlements may also change over time, e.g. at the boundary between two TV programmes.

NOTE: In this use-case, the content is packaged in ISO/BMFF as profiled by CMAF [1] and encrypted using common encryption also as profiled by CMAF [1].

[1] <http://mpeg.chiariglione.org/sites/default/files/files/standards/parts/docs/w16186.zip>

4 Derived Requirements

4.1 MSE

The following requirements are derived from the above use-cases and are believed not to be met with MSE v1. If entries already exist in GitHub that cover the requirement, then references to these are included.

1. It shall be possible to play one content item up to a particular position (not necessarily the end) and then (seamlessly or close to it) switch to a play a different content item (not necessarily from the start) where the two content items may differ by properties such as codec, number of tracks and number of audio channels.

4.2 EME

The following requirements are derived from the above use-cases and are believed not to be met with EME v1. If entries already exist in GitHub that cover the requirement then references to these are included.

1. It shall be possible to play content containing a mixture of encrypted and unencrypted content.

<https://github.com/w3c/encrypted-media/issues/251>

also <https://github.com/w3c/encrypted-media/issues/197>

2. It shall be possible to support key rotation

<https://github.com/w3c/encrypted-media/issues/132>

also <https://github.com/w3c/encrypted-media/issues/121>

MSE GitHub Issues (for cross-checking only)

Status = VNext

Title		
new feature - add remove sourcebuffer in every moment	feature request	https://github.com/w3c/media-source/issues/120
Please support for audio/wav (PCM) in MSE	feature request	https://github.com/w3c/media-source/issues/55
Needs mechanism to present texts and graphics accurately synchronized with video at low processing load.	feature request	https://github.com/w3c/media-source/issues/41
Needs event to notify when sourceBuffer needs more data	feature request	https://github.com/w3c/media-source/issues/40
More flexible error handling	feature request	https://github.com/w3c/media-source/issues/36
Support attachment (and inspection) of MediaSource to HTMLMediaElement via HTMLMediaElement.srcObject	feature request	https://github.com/w3c/media-source/issues/28
should buffering model be an option?	feature request	https://github.com/w3c/media-source/issues/21
Update SourceBuffer.appendStream() and related algorithms to use ReadableByteStream	feature request	https://github.com/w3c/media-source/issues/14
ISO BMFF Byte Stream format should support layered (scalable) encodings	feature request	https://github.com/w3c/media-source/issues/7
Refine TrackDefault spec and implementations		https://github.com/w3c/media-source/issues/140
Require (for at least webm and mp4, possibly others) "full" codec type in isTypeSupported() and addSourceBuffer()		https://github.com/w3c/media-source/issues/137
Need event to signal when decoder underflow has occurred		https://github.com/w3c/media-source/issues/101
Have appendBuffer and remove return promise.		https://github.com/w3c/media-source/issues/100
Clarify/remove 'stalled' and 'progress' events for MediaSource		https://github.com/w3c/media-source/issues/88
Clean-up the spec text around VideoPlaybackQuality		https://github.com/w3c/media-source/issues/77
VideoPlaybackQuality is underdefined, leading to interop concerns		https://github.com/w3c/media-source/issues/76
ISO BMFF bytestream: how can CEA 608 / 708 embedding be supported		https://github.com/w3c/media-source/issues/58
High-level overview of media specifications		https://github.com/w3c/media-source/issues/46
Support sample accurate audio splicing using timestampOffset/appendWindowStart/appendWindowEnd		https://github.com/w3c/media-source/issues/37
Report buffer changes in update events		https://github.com/w3c/media-source/issues/35
MSE does not work with Mixed Content, unlike video/audio.src=		https://github.com/w3c/media-source/issues/22

EME GitHub Issues (for cross-checking only)

To be completed