



viewWard

Certify Devices

Specification for Devices 2019

Version: 4.13-r1

Date: 2019-02-01

CONTENTS

[1. REVISION HISTORY](#)

[2. INTRODUCTION](#)

[2.1. Scope](#)

[2.2. Versions for requirements and software](#)

[2.2.1. Backward compatibility](#)

[2.3. Definitions](#)

[2.4. Compliance terminology used in this document](#)

[2.4.1. REQUIRED and CONDITIONALLY REQUIRED features](#)

[2.4.1.1. DRM](#)

[2.4.1.2. Codecs and media formats](#)

[2.4.1.3. Keys on the remote control](#)

[2.4.1.4. Resolution](#)

[3. TECHNICAL REQUIREMENTS](#)

[3.1. HTML5 <video> and <audio>](#)

[3.1.1. Media element](#)

[3.1.1.1. Requirements for video and audio media elements](#)

[3.1.1.2. Requirements for video media elements](#)

[3.1.1.3. Codec support](#)

[3.1.2. Track element](#)

[3.1.2.1. Requirements for all track elements](#)

[3.1.2.2. Requirements for text track elements](#)

[3.2. Media streaming](#)

[3.2.1. Transport protocols](#)

[3.2.2. Progressive download](#)

[3.2.3. Adaptive Bitrate streaming protocols](#)

[3.2.3.1. Apple HTTP Live Streaming \(HLS\)](#)

[3.2.3.1.1. Restrictions for HLS content](#)

[3.2.3.2. MPEG-DASH](#)

[3.2.3.2.1. Restrictions for MPEG-DASH content](#)

[3.2.3.3. Microsoft Smooth Streaming \(MSSS\)](#)

[3.2.3.3.1. Restrictions for Smooth Streaming content](#)

[3.3. Media Source Extensions \(MSE\)](#)

[3.4. Subtitles and Closed Captioning](#)

[3.5. DRM](#)

[3.5.1. Content Decryption Modules \(CDMs\)](#)

[3.5.1.1. ClearKey](#)

[3.5.1.2. PlayReady](#)

[3.5.1.2.1 PlayReady: License Acquisition](#)

[3.5.1.3. Widevine](#)

[3.5.1.4. AES-128 encryption](#)

[3.5.2. DRM Invocation Methods](#)

[3.5.2.1. WebInitiator](#)

[3.5.2.2. Encrypted Media Extensions \(EME\)](#)

[3.6. Video and audio formats](#)

[3.6.1. Media container formats](#)

[3.6.2. Video codecs](#)

[3.6.3. Audio codecs](#)

[3.7. Input handling](#)

[3.7.1. Key mappings](#)

[3.7.1.1. Navigation and Select keys](#)

[3.7.1.2. Back key](#)

[3.7.1.3. Exit/Close key](#)

[3.7.1.4. window.close\(\)](#)

[3.7.1.5. Entering text](#)

[3.8. User Agent string](#)

[3.9. Screen resolution](#)

[3.10. Security](#)

[3.10.1. Same-Origin policy](#)

[3.10.2. Mixed content](#)

[3.10.3. Root certificates](#)

[3.10.4 Cipher suites](#)

[3.10.5 Debugging tools](#)

[3.11. Performance](#)

[3.11.1. General performance](#)

[3.11.2. Video transition \(Advert insertion\) performance](#)

[3.12. Settings](#)

[3.12.1. Fonts and languages](#)

[3.12.2. Date and time](#)

[3.12.3. Default colors](#)

[3.12.4. Memory](#)

[3.12.4.1. Out of memory](#)

[3.12.5. Storage](#)

[3.12.6. Cookies](#)

[3.12.7. Disconnect from network](#)

[3.12.8. Cannot open URL](#)

[3.13. Accessibility](#)

[3.13.1 Text-to-speech](#)

[4. ABBREVIATIONS](#)

[5. REFERENCES](#)

[6. ANNEX A \(informative\). Media type strings for video and audio codecs](#)

[6.1. MP4 video and audio](#)

[6.1.1. H.264 profiles](#)

[6.1.2. H.265/HEVC profiles](#)

[6.1.3. AV1 profiles](#)

[6.1.4. AAC profiles](#)

[6.1.5. MP3 \(MPEG-1 Layer III\) profiles](#)

[6.1.6. Dolby profiles](#)

[6.1.7 Combination examples of media type strings](#)

[6.2. WebM video and audio](#)

[6.3. Audio only](#)

1. REVISION HISTORY

Version Number	Chromium	Date	Comment
4.9.0	53.0	2016-08-18	Publication release
4.9.0-r1	53.0	2016-11-04	Changes: <ul style="list-style-type: none"> • Default background/text color: section 3.12.3 • Key Back is remapped to JavaScript VK_BACK, not VK_BACK_SPACE: section 3.7.1 • The default behavior of Back key: section 3.7.1.2 • OPR/ component added to the UserAgent String: section 3.8 • Device screen resolution: section 2.4.1.4 and 3.9 • Optional feature 'persistent licences' for Widevine was removed: section 3.5.1.3
4.10	56.0	2017-03-16	Changes: <ul style="list-style-type: none"> • Added a few MPEG-DASH MIME types profiles to be recognized by MPEG-DASH client: section 3.2.3.2 • Clarified <SegmentList> support as optional: section 3.2.3.2 • Clarified scope of h.264 and h.265 video codec support: section 3.6.2 • EME is only available via https protocol: section 3.5.2.2 • EME updated to spec July 2016: section 3.5.2.2 • Dropped support of progressive VP9 Profile 2 streaming: section 3.6.2
4.11	59.0	2017-09-15	Changes: <ul style="list-style-type: none"> • Media Source Extensions is required in newer revision - W3C Candidate Recommendation 05 July 2016: section References • Added performance requirements for CSS Transforms: section 3.11 • Clarified requirements regarding no Internet connection: section 3.12.7 • Clarify HDR support for specific codecs and streaming protocols: section 3.6.2
4.12	63.0	2018-04-03	Changes: <ul style="list-style-type: none"> • Added cipher suites requirements: section 3.10.4 • Added Advert Insertion performance requirements: section 3.11.2 • Devtools must be disabled: section 3.10.5 • Added requirement for MPEG-DASH period in section 3.2.3.2.1 • Added requirement for PlayReady automatic license acquisition: 3.5.1.2
4.13	67.0	2018-07-18	Changes: <ul style="list-style-type: none"> • Added text-to-speech requirements in a new 3.13. Accessibility chapter.

4.13-r1	67.0	2019-02-01	Changes: <ul style="list-style-type: none">• Added AV1 codec requirements: section 3.6.2• Added OGG Vorbis audio codec requirements: section 3.6.3• Added Microsoft PlayReady CustomData and License Acquisition override feature optional requirement: 3.5.1.2.1• Removed VP8 requirement.
---------	------	------------	--

2. INTRODUCTION

2.1. Scope

This document specifies the requirements that must be met by devices in order to be part of the Vewd Certify Devices.

The Certify Devices defines a complete OTT platform suitable for any OTT content globally, with rich features for media streaming, content protection, and security. As the document is based on feedback from content owners and app developers worldwide, failure to support the features specified below will place limits on the apps and OTT content that a device will be able to support.

Note that requirements for specific apps (such as YouTube TV) or regional standards (such as HbbTV or Freeview Play) are out of the scope of this document.

2.2. Versions for requirements and software

For each update of the Vewd Core, the version number of this specification and the Specification for Devices [1] are also updated to match.

2.2.1. Backward compatibility

New versions of these requirements may not be backward compatible with previous versions. Changes are planned to be introduced in two phases which both last approximately 6 months.

Features that are planned to be removed will be marked as [DEPRECATED] in this specification.

- After a deprecation period, [DEPRECATED] items will be removed in later versions of the specification.
- Items that are introduced in later versions of the specification will be marked as [INTRODUCED in <version>].
- The [Specification revision history](#) table will be updated accordingly.

2.3. Definitions

Chromium and **Google Chrome** - Chromium is the open-source project that forms the basis of the Google Chrome browser.

Certify Devices - A program run by Vewd to certify platforms according to this specification. The goal of the program is to ensure that a platform can run all OTT apps that are certified for Vewd.

Vewd Core - An embeddable browser and streaming engine with an extensible API, based on the Chromium open-source project, which implements a set of international and industry standards to download and render webpages, execute web apps, and stream video and audio content.

Vewd - A set of products and solutions developed by Vewd for TV and Set-Top Box (STB) manufacturers to enable HTML5 rendering and adaptive streaming in their devices.

(Also used as a short form for the company Vewd)

Vewd Application - A Vewd-compliant web app that is certified to run on Vewd Devices. Also referred to as App or Application in this document.

Vewd Device - A TV or STB device running software based on the Vewd Core, and certified to meet the requirements defined in the Vewd Specification for Devices.

Vewd Specification for Devices or Specification for Devices - This document, which specifies the platform requirements that Vewd Devices must meet to be officially certified, and the features that an app can expect to have available when running on a Vewd Device.

See also: [5. Abbreviations](#)

2.4. Compliance terminology used in this document

The following keywords used in this specification to indicate the level of compliance needed for the requirements are sourced from RFC2119 [\[30\]](#). In essence:

- MUST, REQUIRE or SHALL indicates that you need to comply with the requirement absolutely.
- SHOULD or RECOMMENDED indicates that while sometimes there could be valid reasons to ignore the requirement, you need to fully understand and accept the implications and risks to optimal end-user experience.
- MAY or OPTIONAL indicates that you can decide to implement an item at your discretion.

2.4.1. REQUIRED and CONDITIONALLY REQUIRED features

We also use an additional compliance term not described in RFC2119:

- **CONDITIONALLY REQUIRED** - an item or a feature that must be supported in the browser if the underlying platform supports this capability. Examples could include a specific audio or video codec, or support for Ultra HD/4K resolution.

This specification defines one platform, and care has been taken to ensure that there are no choices or variants of this platform. However, there are a few features where the functionality available to an app is directly dependent on the capabilities of the device. These features are marked as **CONDITIONALLY REQUIRED** in this specification.

Below is an overview of requirements that are dependent on underlying platform capabilities. For more information about each item see the corresponding sections in this document.

2.4.1.1. DRM

Support for content protection with Microsoft PlayReady is mandatory for Vewd Devices, while content protection with Google's Widevine is mandatory only if the device has Widevine installed.

2.4.1.2. Codecs and media formats

Support for video codecs and media formats is only mandatory if the codecs or formats are supported by the underlying platform, as these are dependent on the chipset powering the device, and on external licenses. some codecs.

Audio:

- Opus
- Dolby AC3/E-AC3
- Vorbis Ogg

Video:

- H.265 (HEVC)
- VP9
- AV1

Container:

- WebM (Only required when AV1 and / or VP9 is supported)

2.4.1.3. Keys on the remote control

The design of the remote control is up to each manufacturer, and may vary in size and functionality. Some keys are marked as mandatory (such as. 4-way navigation keys), while others are marked as mandatory only for devices where there is such a key on the remote control.

2.4.1.4. Resolution

The maximum resolution of a device is defined by the manufacturer. All devices must support either HD (720p) or Full HD (1080p), but some features may be mandatory if the device supports Full HD (1080p) or Ultra HD/4K (2160p) resolutions.

3. TECHNICAL REQUIREMENTS

3.1. HTML5 <video> and <audio>

Vewd Devices MUST support HTML5 <video> and <audio> elements according to the HTML 5.1 specification [\[9\]](#).

3.1.1. Media element

3.1.1.1. Requirements for video and audio media elements

All devices MUST support the following requirements for the handling of <video> and <audio> media elements:

- Initial state of the object's attributes ([4.7.14.2](#), [4.7.14.6](#), [4.7.14.8](#), [4.7.14.9](#))
- Handling of error states during playback ([4.7.14.1](#))
- Audio and video elements MUST be able to play both types of sources ([4.7.14](#))
- Proper handling of events during the playback ([4.7.14.8](#), [4.7.14.9](#), [4.7.14.15](#))
- Proper handling of *playbackRate* and *defaultPlaybackRate* ([4.7.14.8](#))
- The resource selection algorithm MUST work properly in the case of multiple source elements ([4.7.14.5](#))
- The *preload* attribute of the media element MUST be respected ([4.7.14.5](#))

3.1.1.2. Requirements for video media elements

All devices MUST pass the following requirements for the handling of <video> media elements:

- Handling the poster attribute ([4.7.10](#))
 - Proper aspect ratio MUST be preserved during resizing of the containing element of the video ([4.7.20](#))
 - Video elements MUST be able to play a stream containing only audio and text tracks ([4.7.10](#))
 - Proper rendering of video sources
- PLANNED INTRODUCTION:** Future versions of this document will require three video streams: one currently playing and two pre-loaded.

3.1.1.3. Codec support

All devices MUST respond truthfully to codec support inquiries:

- *canPlayType* response for specified video and audio codecs ([4.7.14.3](#))

3.1.2. Track element

Devices MUST support rendering of subtitles and closed captions as specified in HTML 5.1, "The track element" ([4.7.13](#)). Supported track formats are specified in the [Subtitles and Closed Captioning](#) section.

3.1.2.1. Requirements for all track elements

A device MUST meet the following requirements for the handling of track elements:

- Initial state values of media tracks and text tracks ([4.7.14.10.1](#), [4.7.14.11](#))
- *readyState* attribute handling for out-of-band text tracks ([4.7.14.11.3](#))
- Proper handling of enabling and disabling audio tracks ([4.7.14.10.2](#))
- Switching of video and text tracks ([4.7.14.10.2](#), [4.7.14.11](#))

3.1.2.2. Requirements for text track elements

- Devices MUST support the synchronization of the text track and the audio track.
- Text tracks MUST be rendered properly on video media elements.

3.2. Media streaming

3.2.1. Transport protocols

- The device MUST support the retrieval of any media content either by HTTP or HTTPS using HTTP protocol v1.1 and Range requests.
- The device MUST support Transport Layer Security (TLS) version 1.2 with forward security.
- TLS key sizes MUST be at least 2048 bits for RSA and 256 bits for EC.
- TLS MUST NOT use any known insecure cryptographic primitives (e.g., RC4 encryption, SHA-1 certificate signatures).

3.2.2. Progressive download

The device MUST support the following combinations:

Container	Audio codecs	Video codecs	DRM	DRM trigger	In-band subtitles
ISO BMFF	AAC-LC HE-AAC v1 HE-AAC v2 MP3 Dolby AC3 Dolby E-AC-3	H.264 H.265	None	None	Not supported
MPEG2-TS	AAC-LC HE-AAC v1 HE-AAC v2 MP3 Dolby AC3 Dolby E-AC-3	H.264	None	None	Not supported
WebM	Opus	VP9 AV1	None	None	Not supported
ADTS / AAC MP3	AAC-LC HE-AAC v1 HE-AAC v2 MP3	None	None	None	Not supported

Note 1: All rules and restrictions for the support of media formats and codecs applies as outlined in the [Video and audio formats](#) section.

3.2.3. Adaptive Bitrate streaming protocols

The following Adaptive Bitrate (ABR) streaming protocols MUST be supported:

Streaming Type	MIME-Types	Notes
Apple HTTP Live Streaming (HLS)	application/vnd.apple.mpegurl application/x-mpegURL	VoD (append-mode window) and Event (sliding window)
MPEG-DASH	application/dash+xml	Main and Live profiles of MPEG-DASH
Microsoft Smooth Streaming (MSS)	application/vnd.ms-sstr+xml application/vnd.ms-playready.initiator+xml	

3.2.3.1. Apple HTTP Live Streaming (HLS)

The device MUST support HTTP Live Streaming Protocol version 3, as defined in [2]. It MUST support both Live and On-Demand streams. Support for the following M3U8 playlist tags is REQUIRED:

- #EXTM3U (section 3.1 in [2])
- #EXTINF (section 3.1 in [2])
- #EXT-X-TARGETDURATION (section 3.3.1 in [2])
- #EXT-X-MEDIA-SEQUENCE (section 3.3.2 in [2])
- #EXT-X-KEY (section 3.3.3 in [2])
- #EXT-X-ENDLIST (section 3.3.7 in [2])
- #EXT-X-STREAM-INF (section 3.3.8 in [2])
- #EXT-X-DISCONTINUITY (section 3.3.9 in [2])
- #EXT-X-VERSION (section 3.3.10 in [2])

Container	Audio codecs	Video codecs	Encryption	Decryption trigger	In-band subtitles	MIME type
MPEG2-TS	AAC-LC HE-AAC v1 HE-AAC v2 MP3 Dolby AC3 Dolby E-AC-3	H.264 H.265	None		Not supported	application/vnd.apple.mpegurl application/x-mpegURL
MPEG2-TS	AAC-LC HE-AAC v1 HE-AAC v2 MP3 Dolby AC3 Dolby E-AC-3	H.264 H.265	AES-128	Manifest	Not supported	application/vnd.apple.mpegurl application/x-mpegURL
ADTS	AAC-LC HE-AAC v1 HE-AAC v2	None	None		Not supported	application/vnd.apple.mpegurl application/x-mpegURL
ADTS	AAC-LC HE-AAC v1 HE-AAC v2	None	AES-128	Manifest	Not supported	application/vnd.apple.mpegurl application/x-mpegURL
MP3	MP3	None	None		Not supported	application/vnd.apple.mpegurl

						application/x-mpegURL
MP3	MP3	None	AES-128	Manifest	Not supported	application/vnd.apple.mpegurl application/x-mpegURL

Note 1: All rules and restrictions for the support of media formats and codecs applies as outlined in the [Video and audio formats](#) section.

3.2.3.1.1. Restrictions for HLS content

A device **MUST** be able to handle streams with the following limitations:

Parameter	Requirements
Frame rate	Up to 60fps
Audio sampling rate	Up to 48000 Hz
Number of audio channels	Up to 8 (7+LFE)
Media segment file size	Up to 15MB
Segment duration	In range 1s - 12s
Average bitrate over one segment	Up to 8 Mbit/s (for up to 1080p)
Manifest file size	Up to 2MB
Number of tracks in one M3U8 manifest file	Up to 36

Devices **MAY** fail gracefully on streams that do not abide by the following restrictions:

1. Audio/video encoding
 1. The same codec **MUST** be used across all variant streams (all quality levels).
 2. Audio parameters (number of channels and sample rates) **MUST** be the same across all variant streams.
2. Media segments
 1. All media segments **MUST** be independently decodable. Consequently, the first video frame in every segment that contains video **MUST** be an IDR frame.
 2. Discontinuities in timestamps, frame rate, encoding profiles, or audio/video parameters **MUST NOT** occur within segments.
3. Playlist files (M3U8)
 1. Audio and video playlists **MUST** use the same target duration, and **MUST** contain the same duration of content.
 2. A playlist **MUST NOT** contain invalid URLs.
 3. Media sequence numbers **MUST** be aligned across all variant streams (quality levels), so that media sequence numbers can be used to identify matching content.
 4. For live streams, media segments **MUST** remain available on the server for at least one target duration after the segment disappears from the playlist.
 5. Playlists **MUST** use sufficiently accurate segment durations to ensure that the sum of the #EXTINF durations of any contiguous group of segments is within one video frame duration of the actual duration.
 6. Playlists **MUST** provide at least 6 segments in live/linear streams.
 7. Discontinuities in timestamps, frame rate, encoding profiles, or audio/video parameters **MAY** occur between segments, but such discontinuities **MUST** be indicated using the #EXT-X-DISCONTINUITY tag.
 8. EXT-X-STREAM-INF tags **MUST** always provide CODECS and RESOLUTION attributes.
4. Subtitles
 1. The device **MUST** support subtitles that conform to the [Subtitles and Closed Captions](#) section.

5. DRM

1. The decryption key **MUST** be directly downloadable via an HTTP or HTTPS URLs
2. Note that with AES-128 encrypted HLS, segments are completely encrypted.

3.2.3.2. MPEG-DASH

The device **MUST** support the following MPEG-DASH profiles:

Profile	Identifier	Reference
ISO Base Media File Format Live	urn:mpeg:dash:profile:isoff-live:2011	[3], section 8.4
ISO Base Media File Format Main	urn:mpeg:dash:profile:isoff-main:2011	[3], section 8.5
DASH-AVC/264	http://dashif.org/guidelines/dash264 urn:com:dashif:dash264	[29], section 6.3
DASH-AVC/264 SD	http://dashif.org/guidelines/dash264#sd	[29], section 7.3
DASH-AVC/264 HD	http://dashif.org/guidelines/dash264#hd	[29], section 8.3
DASH-AVC/264 main	http://dashif.org/guidelines/dash264main	[5], section 8.2
DASH-AVC/264 high	http://dashif.org/guidelines/dash264high	[5], section 8.3

Note that the DVB Profile of MPEG-DASH ([4], section 4.1), identified as “urn:dvb:dash:profile:dvb-dash:2014”, is required by HbbTV 2.x and Freeview Play (UK), and that support for this profile is planned to become mandatory in a later version of this specification.

The following combinations **MUST** be supported by the device:

Container	Audio codecs	Video codecs	DRM	DRM Trigger	In-band subtitle	MIME type
ISO BMFF	AAC-LC HE-AAC v1 HE-AAC v2 MP3 Dolby AC3 Dolby E-AC-3	H.264 H.265	None	None	Supported	application/dash+xml
ISO BMFF	AAC-LC HE-AAC v1 HE-AAC v2 MP3 Dolby AC3 Dolby E-AC-3	H.264 H.265	ClearKey PlayReady Widevine	EME	Supported	application/dash+xml

Note 1: All rules and restrictions to the support of media formats and codecs applies as outlined in the [Video and audio formats](#) section.

Note 2: All rules and restrictions for the support of DRM applies as outlined in the [DRM](#) and [EME](#) sections.

3.2.3.2.1. Restrictions for MPEG-DASH content

A device **MUST** be able to handle streams, with the following limitations:

Parameter	Requirements
Frame rate	Up to 60fps
Audio sampling rate	Up to 48000 Hz

Number of audio channels	Up to 8 (7+LFE)
Media segment file size	Up to 15MB
Segment duration	In range 1s - 12s
Average bitrate over one segment	Up to 10Mbit/s (for 1080p content)
Manifest file size	Up to 2MB
Number of tracks in one MPD file	Up to 36

Clients may fail gracefully on streams that do not abide by the following restrictions and are not compliant with the [\[5\] DASH-IF Interoperability Points](#) documentation:

- The media segment container format **MUST** be the ISO Base Media File Format (aka MP4).
- All media segments **MUST** be independently decodable. Consequently, the first video frame in every segment that contains video **MUST** be an IDR frame.
- Manifest URLs **MAY** include the MPD anchor, but **MUST NOT** use any other than the 't' parameter ([\[3\]](#) section C.4)
- The device **MUST** support multiple audio tracks associated with multiple Adaptation Sets defined in an MPD of MPEG-DASH.
- Each audio track **MUST** be a separate media stream.
- Support for the <SegmentList> element is **NOT REQUIRED**
- The device **MUST** support multiple Periods defined in an MPD of MPEG-DASH

3.2.3.3. Microsoft Smooth Streaming (MSSS)

Devices **MUST** support Microsoft Smooth Streaming Transport Protocol v2.2 as defined in [\[2\]](#), and **MUST** support both Live and On-Demand streams.

NOTE: The version number refers to the MajorVersion and MinorVersion attributes in the manifest, *not* the Smooth Streaming Protocol Specification version. As of revision 6.0 of the specification, the only valid values are 2.0 and 2.2 (see section 2.2.2.1 in [\[3\]](#)).

Container	Audio codecs	Video codecs	DRM	DRM Trigger	In-band subtitle	MIME type
PIFF v1.1 [13]	AAC-LC HE-AAC v1 HE-AAC v2	H.264	None	None	Supported	application/vnd.ms-sstr+xml
PIFF v1.1 [13]	AAC-LC HE-AAC v1 HE-AAC v2	H.264	PlayReady	Manifest	Supported	application/vnd.ms-sstr+xml
PIFF v1.1 [13]	AAC-LC HE-AAC v1 HE-AAC v2	H.264	PlayReady	WebInitiator	Supported	application/vnd.ms-playready.initiator+xml

Note 1: all rules and restrictions to the support of media formats and codecs applies as outlined in the [Video and audio formats](#) section.

Note 2: All rules and restrictions to the support of DRM applies according to the [DRM](#) and [WebInitiator](#) sections.

3.2.3.3.1. Restrictions for Smooth Streaming content

A device **MUST** be able to handle streams within the following limitations:

Parameter	Requirements
Frame rate	Up to 60fps

Audio sampling rate	Up to 48000 Hz
Number of audio channels	Up to 8 (7+LFE)
Media segment file size	Up to 15MB
Segment duration	In range 1s - 12s
Average bitrate over one segment	Up to 10Mbit/s (for 1080p content)
Manifest file size	Up to 2MB

Clients MAY fail gracefully on streams that do not abide by the following restrictions:

- All media segments MUST be independently decodable. Consequently, the first video frame in every segment that contains video MUST be an IDR frame.
- For live streams that use FragmentLookahead, segments MUST remain available on the server for one DVRWindowLength after they disappear from the Manifest file.

3.3. Media Source Extensions (MSE)

Media Source Extensions MUST be supported according to the MSE specification [7]. The following combinations of containers and codecs MUST be supported:

Container	Audio codecs	Video codecs
MP4	AAC / MP3	H.264 / H.265
WebM	Opus	VP9
MP4	AAC / MP3	<no video>
WebM	Opus	<no video>
MP4	<no audio>	H.264 / H.265
WebM	<no audio>	VP9
MP4	<no audio>	AV1
MP4	AAC / MP3	AV1

Note: All rules and restrictions to the support of media formats and codecs in MSE applies as outlined in the [Video and audio formats](#) section.

3.4. Subtitles and Closed Captioning

To display subtitles or Closed Captions, the device MUST support WebVTT according to [15] to the extent that it is supported by the Chromium engine, and MUST support the EBU-TT-D text track profile, as specified in [17], which is a subset of the TTML text track format.

Support for in-band and out-of-band text tracks is REQUIRED according to the table below:

Media delivery method	In-band subtitles	Out-of-band subtitles
Progressive playback	Not supported	Mandatory

HLS	Not supported	Mandatory
MPEG-DASH	Mandatory	Mandatory
Smooth Streaming	Mandatory	Mandatory
MSE	Not supported	Mandatory

3.5. DRM

3.5.1. Content Decryption Modules (CDMs)

3.5.1.1. ClearKey

- MUST be supported with EME

3.5.1.2. PlayReady

- MUST be supported with EME
- MUST be supported with WebInitiator
- PlayReady Header Object v4.0.0.0 MUST be supported [8]
- PlayReady Header Object v4.1.0.0 SHOULD be supported [8]
- MUST be supported with security level “2000” or higher

3.5.1.2.1 PlayReady: License Acquisition

- Device MUST support automatic license acquisition, supporting the following:
 - License Acquisition URLs provided in the manifest MUST override those provided in the PSSH boxes
 - License Acquisition URL provided in the WebInitiator MUST override those provided in the manifest and the PSSH boxes
- Reactive License Acquisition (Post-delivery) MAY support CustomData and License Acquisition override mechanism using LA_URL and drm_custom_data variables passed as GET parameters of video source URL to override global settings

3.5.1.3. Widevine

- Widevine is CONDITIONALLY REQUIRED if the platform has Widevine DRM installed
- MUST be supported with EME
- MUST be supported with security level “L1”
- SHOULD support “server certificate” and “privacy mode” features

3.5.1.4. AES-128 encryption

- MUST be supported for Apple HLS streams

3.5.2. DRM Invocation Methods

3.5.2.1. WebInitiator

PlayReady MUST be supported with the features below:

- Licence Pre-Acquisition for PlayReady
 - Available only together with Microsoft Smooth Streaming (MSSS) content
 - Mime type: "application/vnd.ms-playready.initiator+xml"

3.5.2.2. Encrypted Media Extensions (EME)

Encrypted Media Extensions (EME) MUST be supported un-prefixed according to the W3C Working Draft 05 July 2016 [\[6\]](#).

Note: EME MUST only be used on secure contexts, it can not be used on any pages served over HTTP.

The following EME features MUST be supported for the CDMs:

- ClearKey according to EME specification “9.1 Clear Key”
 - MUST support key system ID: “org.w3.clearkey”
 - MUST support initialization data types: CENC [\[10\]](#), WEBM [\[11\]](#), KEYIDS [\[12\]](#)
- Widevine (CONDITIONALLY REQUIRED)
 - Widevine MUST be supported when available on the device
 - MUST support key system ID: “com.widevine.alpha”
 - MUST support initialization data type: CENC [\[10\]](#), KEYIDS [\[12\]](#)
 - Robustness “HW_SECURE_DECODE” MUST be supported for video
 - Minimal robustness “SW_SECURE_CRYPTO” MUST be supported for audio
 - Features “server certificate” and “privacy mode” SHOULD be supported
- PlayReady
 - MUST support key system ID “com.microsoft.playready”
 - MUST support initialization data type: CENC [\[10\]](#)

EME MUST work with MSE and adaptive streaming.

- The `HTMLMediaElement.onencrypted` event MUST be triggered if encrypted content has been detected.
- After a successful call to `HTMLMediaElement.setMediaKeys()` with valid `MediaKeys`, content MUST be fully playable.

Detecting encrypted content and triggering licence acquisition MUST be supported from:

- Manifest - DRM initialization data are stored in an adaptive streaming manifest
- Media container - DRM initialization data are stored in a video container (CENC [\[14\]](#) or PIFF [\[13\]](#))

Not all DRM invocation methods and DRM systems are available with all transfer protocols. This is specified in detail in the section for each protocol. Every CDM MUST be supported with all combinations of MSE-supported formats and codecs.

NOTE: Either Widevine or PlayReady is required to play encrypted content on YouTube TV. Widevine is required in order to support Ultra HD/4K resolution videos on YouTube TV.

3.6. Video and audio formats

3.6.1. Media container formats

The following media container formats MUST be supported:

- ISO Base Media File Format ISO/IEC 14496-12:2012
 - Streaming-optimized MP4 (moov box before the mdat box)
 - Unoptimized MP4 (mdat box before the moov box)
- WebM (CONDITIONALLY REQUIRED)
 - MUST be supported when VP9 video codecs are available on the device
- MPEG2-TS ISO/IEC 13818-1:2000
- ADTS / AAC (audio elementary stream)
- MPEG-1 Layer III (audio elementary stream)

3.6.2. Video codecs

The following video codecs formats MUST be supported:

- H.264 as specified in [\[20\]](#)

- The device **MUST** support all profile/level configurations up to High Profile Level 4.1 included.
- H.265 as specified in [19] (CONDITIONALLY REQUIRED)
 - H.265/HEVC **MUST** be supported when the codec is available on the device
 - The device **MUST** support all profile/level configurations up to High Profile Level 4.1 included.
 - HEVC Main Level 5 and 5.1 (CONDITIONALLY REQUIRED)
 - These two levels **MUST** be supported when the device supports Ultra HD/4K resolution (2160p)
 - HEVC Main 10 Level 4.1 and 5.1 (CONDITIONALLY REQUIRED)
 - These two levels **MUST** be supported when the device supports HDR, either HDR10 or HLG10
 - It is **ONLY** required in the case of MPEG-DASH streaming use-case
- VP9 as specified in [22] (CONDITIONALLY REQUIRED)
 - VP9 **MUST** be supported when the codec is available on the device
 - Profile 0 (CONDITIONALLY REQUIRED)
 - When device supports VP9 then VP9 profile 0 **MUST** be supported
 - The following VP9 levels **MUST** be supported (described in [18]): 1, 1.1, 2.1, 3, 3.1, 4, 4.1
 - When the device supports video in Ultra HD/4K resolution (2160p) then it **MUST** support the VP9 levels (described in [18]): 5, 5.1
 - Profile 2 (CONDITIONALLY REQUIRED)
 - When the device supports HDR, either HDR10 or HLG10, then VP9 profile 2 **MUST** be supported
 - It is **ONLY** required in the case of Media Source Extension(MSE) and MPEG-DASH streaming use-cases
 - The device **MUST** support the VP9 levels (described in [18]): 1, 1.1, 2.1, 3, 3.1, 4, 4.1
 - When the device supports video in Ultra HD/4K resolution (2160p), it **MUST** support the VP9 levels (described in [18]): 5, 5.1
- AV1 as specified in [21] (CONDITIONALLY REQUIRED)
 - AV1 **MUST** be supported when the codec is available on the device
 - Profile 0 (CONDITIONALLY REQUIRED)
 - When device supports AV1 then AV1 profile 0 (MAIN) **MUST** be supported
 - The following AV1 levels **MUST** be supported (described in [21]): 2.0, 2.1, 3.0, 3.1, 4.0, 4.1

VP9 Profile 2 is **ONLY** required in the case of Media Source Extension(MSE) streaming use-case.

3.6.3. Audio codecs

The following audio codecs **MUST** be supported:

- HE-AAC v1
- HE-AAC v2
- LC-AAC
- MP3
- Opus (CONDITIONALLY REQUIRED)
 - Opus **MUST** be supported when the codec is available on the device
- Dolby AC3/E-AC3 (CONDITIONALLY REQUIRED)
 - Dolby AC3/E-AC3 **MUST** be supported when the codecs are available on the device
- Ogg Vorbis (CONDITIONALLY REQUIRED)
 - Ogg Vorbis **MUST** be supported when the codec is available on the device

[Annex A](#) specifies the mime-types for audio and video encoding schemes.

3.7. Input handling

3.7.1. Key mappings

The device **MUST** provide standardized key codes which are mapped from a remote control input to a platform key which **MUST** be sent to the Vewd Core. All “JavaScript key code” constants **MUST** be available to the web apps in the global JavaScript context.

Hardware key	Linux key code	Android key code	JavaScript key code	Requirement
←	OMI_KEY_LEFT	KEYCODE_DPAD_LEFT	VK_LEFT	Mandatory
→	OMI_KEY_RIGHT	KEYCODE_DPAD_RIGHT	VK_RIGHT	Mandatory
↑	OMI_KEY_UP	KEYCODE_DPAD_UP	VK_UP	Mandatory
↓	OMI_KEY_DOWN	KEYCODE_DPAD_DOWN	VK_DOWN	Mandatory
Confirm / Select / OK	OMI_KEY_ENTER	KEYCODE_DPAD_CENTER / KEYCODE_ENTER	VK_ENTER	Mandatory
Back / Return	OMI_KEY_BACK	KEYCODE_BACK	VK_BACK	Mandatory
Exit/Close	N/A	N/A	N/A	CONDITIONALLY REQUIRED*
BLUE	OMI_KEY_BLUE	KEYCODE_PROG_BLUE	VK_BLUE	CONDITIONALLY REQUIRED*
RED	OMI_KEY_RED	KEYCODE_PROG_RED	VK_RED	CONDITIONALLY REQUIRED*
GREEN	OMI_KEY_GREEN	KEYCODE_PROG_GREEN	VK_GREEN	CONDITIONALLY REQUIRED*
YELLOW	OMI_KEY_YELLOW	KEYCODE_PROG_YELLOW	VK_YELLOW	CONDITIONALLY REQUIRED*
Menu	OMI_KEY_MENU	KEYCODE_MENU	VK_MENU	CONDITIONALLY REQUIRED
0	OMI_KEY_0	KEYCODE_0	VK_0	CONDITIONALLY REQUIRED*
1	OMI_KEY_1	KEYCODE_1	VK_1	CONDITIONALLY REQUIRED*
2	OMI_KEY_2	KEYCODE_2	VK_2	CONDITIONALLY REQUIRED*
3	OMI_KEY_3	KEYCODE_3	VK_3	CONDITIONALLY REQUIRED*
4	OMI_KEY_4	KEYCODE_4	VK_4	CONDITIONALLY REQUIRED*
5	OMI_KEY_5	KEYCODE_5	VK_5	CONDITIONALLY REQUIRED*
6	OMI_KEY_6	KEYCODE_6	VK_6	CONDITIONALLY REQUIRED*
7	OMI_KEY_7	KEYCODE_7	VK_7	CONDITIONALLY REQUIRED*
8	OMI_KEY_8	KEYCODE_8	VK_8	CONDITIONALLY REQUIRED*
9	OMI_KEY_9	KEYCODE_9	VK_9	CONDITIONALLY REQUIRED*
PLAY	OMI_KEY_PLAY	KEYCODE_MEDIA_PLAY	VK_PLAY	CONDITIONALLY REQUIRED*
PAUSE	OMI_KEY_PAUSE	KEYCODE_MEDIA_PAUSE	VK_PAUSE	CONDITIONALLY REQUIRED*
STOP	OMI_KEY_STOP	KEYCODE_MEDIA_STOP	VK_STOP	CONDITIONALLY REQUIRED*
NEXT	OMI_KEY_TRACK_NEXT	KEYCODE_MEDIA_NEXT	VK_TRACK_NEXT	CONDITIONALLY REQUIRED*
PREV	OMI_KEY_TRACK_PREV	KEYCODE_MEDIA_PREVIOUS	VK_TRACK_PREV	CONDITIONALLY REQUIRED*

FF (Fast-Forward)	OMI_KEY_FAST_FWD	KEYCODE_MEDIA_FAST_FORWARD	VK_FAST_FWD	CONDITIONALLY REQUIRED*
REWIND	OMI_KEY_REWIND	KEYCODE_MEDIA_REWIND	VK_REWIND	CONDITIONALLY REQUIRED*
SUBTITLE	OMI_KEY_SUBTITLE	KEYCODE_CAPTIONS	VK_SUBTITLE	CONDITIONALLY REQUIRED*
INFORMATION	OMI_KEY_INFO	KEYCODE_INFO	VK_INFO	CONDITIONALLY REQUIRED*

CONDITIONALLY REQUIRED* - MUST be supported if this particular key is available on the remote control.

3.7.1.1. Navigation and Select keys

Navigation and Select keys MUST result in sending corresponding key codes with key events into the web app according to the DOM Level 3 Events specification ([\[23\]](#)).

3.7.1.2. Back key

The Back/Return button is a mandatory button on the remote control to go back or close the app. The JavaScript window.VK_BACK is passed to the app, so it can be handled by JavaScript. The caption for this button should be "Back" or "Return" or similar, as long as it remains clear to the end user. The caption should also be consistent with the entire device UI.

3.7.1.3. Exit/Close key

The Exit/Close button is an optional but recommended key on the remote control. The device firmware handles the key, and it should close the app immediately with no event exposed to the app. Its caption should be clear so the end-user understands what they are pressing (example: Exit/Close), and it should be on par with how other use-cases and device functionality is closed.

3.7.1.4. window.close()

When a web app calls `window.close()` in JavaScript, the device MUST handle the call by closing the specific window (app) immediately.

3.7.1.5. Entering text

The device MUST provide a method to input characters into edit or password fields on web pages, either via an on-screen keyboard, or a hardware device (keyboard) that supports entering text.

3.8. User Agent string

The User Agent string MUST contain the following components:

Component	Comment
Mozilla/5.0 (<OS> <Architecture>)	<ul style="list-style-type: none"> The browser states it's "Mozilla-compatible" OS - operating system, e.g. <i>Linux</i> or <i>Android</i> Architecture - CPU architecture, e.g. <i>MIPS</i> or <i>ARM</i>
AppleWebKit/537.36 (KHTML, like Gecko)	WebKit version
Chrome/53.0.*	The Chrome version
Safari/537.36	Safari version

OPR/40.0.2207.0	Opera Desktop version
OMI/4.9.0.*	The Vewd Core version
Model/<CustomerName>-<DeviceModel>	<ul style="list-style-type: none"> • “CustomerName” MUST represent the name of the OEM company • “DeviceModel” MUST represent generalized or particular model name of the OEM device

Note that minor versions of Chrome/ and OMI/ components MAY differ between devices, as it depends on number of builds and fixes delivered in particular project.

The entire User Agent string MAY resemble the following example:

```
Mozilla/5.0 (Linux MIPS) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2774.3 Safari/537.36
OPR/40.0.2207.0 OMI/4.9.0.41.E107811.5 Model/MyManufacturerName-MyModelName
```

3.9. Screen resolution

The target device MUST be capable of rendering at one of the following resolutions:

- 1280 x 720
- 1920 x 1080
 - CONDITIONALLY REQUIRED if the device hardware and platform is capable of Full HD (1080p) or higher rendering
 - CONDITIONALLY REQUIRED if the video plane has Ultra HD/4K (2160p) or better resolution

When a generic browser window is opened, the graphics plane MUST be set up so that the browser can see the best resolution the device is capable of rendering. Note that this MAY be lower than the actual number of pixels in the physical display due to limitations in rendering capabilities or performance.

Depending on the resolution of the graphics plane, logical resolution and CSS pixel resolutions ([\[28\]](#)) WILL be supported as outlined in the following table:

Graphics plane resolution	Window resolution	
	1280 x 720	1920 x 1080
1280 x 720	MUST support at resolution 1dppx	-
1920 x 1080	MUST support at resolution 1.5dppx	MUST support at resolution 1dppx

Note that the resolution seen in a generic browser window, and specified in this document, may differ from the resolution seen by apps running in a HbbTV Window or under control of the Vewd App Store, as follows:

Application window type	Resolution
Generic	According to the table above
TV Store	Always 1280 x 720
HbbTV	Always 1280 x 720
TV Browser UI	According to the table above

3.10. Security

3.10.1. Same-Origin policy

The device MUST apply Same-Origin policy with no exceptions.

3.10.2. Mixed content

The device MUST NOT allow any active mixed content in:

- `<script>` (src attribute)
- `<link>` (href attribute) (this includes CSS stylesheets)
- `<iframe>` (src attribute)
- XMLHttpRequest requests
- All cases in CSS where a URL value is used (@font-face, cursor, background-image, and so on)
- `<object>` (data attribute)

Passive/Display mixed content MUST be allowed on the device but the browser engine SHOULD send a warning to the developer console about mixed content usage.

Passive/Display mixed content types:

- `` (src attribute)
- `<audio>` (src attribute)
- `<video>` (src attribute)
- `<object>` subresources (when an `<object>` performs HTTP requests)

3.10.3. Root certificates

On Linux, the device MUST provide all root certificates vetted according to the Mozilla Root Certificate Program [24] as included by default in Mozilla Network Security Services (NSS) library. The device MAY include additional trusted root certificates according to customer-specific certificate programs.

The Online Certificate Status Protocol (OCSP) for checking the revocation status of X.509 digital certificates MUST be enabled on the device.

On Android, by default, the System Credentials Storage is used to validate server certificates during SSL handshakes. The device MUST provide a reliable implementation of the storage in order to trust given servers.

3.10.4 Cipher suites

The cipher suite requirements are specified in table below and are aligned with HbbTV technical specification ([31]). The cipher suites are defined in IETF RFC 5246 ([32]) and IETF RFC 5289 ([33]). Devices should prioritize these cipher suites in the order shown. The device shall implement all cipher suites marked mandatory and shall not implement any cipher suites marked forbidden.

Cipher suite	Status
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (0xc02b)	Mandatory
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (0xc02f)	Mandatory
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (0xc02c)	Recommended

TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (0xc030)	Recommended
TLS_RSA_WITH_AES_128_CBC_SHA (0x002f)	Mandatory: this is a fallback and should be after all other mentioned ciphers
Cipher suites with anonymous key exchange (TLS_DH_anon_* suites)	Forbidden
Cipher suites with NULL encryption	Forbidden
Cipher suites using RC4 encryption, for example: <ul style="list-style-type: none"> ● TLS_RSA_WITH_RC4_128_SHA (0x0005) ● TLS_RSA_WITH_RC4_128_MD5 (0x0004) 	Forbidden
Cipher suites using encryption or signing algorithms offering less than 112 bits of security	Forbidden

3.10.5 Debugging tools

Vewd Core supports the development debugging tool (Chrome DevTools) as specified in [25]. This tool can be used only in the development phase and MUST be disabled in the final version of the Vewd Integration. The status of the debugging tool in the Vewd Integration can be checked by visiting “chrome://about” URL.

3.11. Performance

3.11.1. General performance

Vewd reserves the right to review the performance of the Vewd integration running on the target device, including how quickly and smoothly the device handles animations, transitions (scale, rotate, skew, 2d, 3d), rendering full-screen images (jpg, png), 2d/3d canvas, remote control input latency, and video playback response (play, seek, pause).

- The device is RECOMMENDED to maintain at least 48fps animating one object using CSS Transforms.
- The device is RECOMMENDED to maintain at least 26fps animating 20 objects using CSS Transforms.

3.11.2. Video transition (Advert insertion) performance

The device SHOULD provide support for dynamic insertion of advertising using two HTML5 media elements (<video> or <audio>) as specified in [34] [Annex J \(informative\): Advert insertion guidance for content providers](#). This requirement will probably be made MANDATORY in the future versions of this specification

At any given time only one HTML media element can be playing. The other element intended for dynamic insertion should be explicitly hidden using CSS property and should start prefetching the media data. Once enough data has been loaded, the second media element should appear and play whereas the first one should be hidden and paused. The delay between the end of the presentation of an HTML5 media element and starting presentation of another HTML5 media element is RECOMMENDED to be less than 250 ms. This time requirement is applicable if the first media element refers to either MPEG DASH content or ISOBMFF (MP4) content and the second media element refers to either MPEG DASH content or ISOBMFF (MP4) content. Advert insertion SHOULD work with two encrypted streams; the 250ms performance, however, is required only in cases of one encrypted stream and one unencrypted stream or two unencrypted streams. The transition between two video streams, one playing, and one preloaded, for any other content than aforementioned, SHOULD be seamless (within 2s).

3.12. Settings

3.12.1. Fonts and languages

The device MUST provide a Sans Serif font containing all the characters required to properly render the text in the languages available on the device. Web Fonts in WOFF / WOFF2 (Web Open Font Format) and TTF (TrueType Font) file format MUST be supported. Right-to-left rendering MUST be supported if required by any language available on the device.

Whenever a language is specified by the device:

- The `window.navigator.language` JavaScript object MUST be set according to specification [\[9\]](#), section “7.1.6.2 Language preferences”.
- *Accept languages* MUST be consistent between HTTP Accept-Language header and the `window.navigator.languages` JavaScript object and SHOULD include the current locale.

3.12.2. Date and time

The device MUST provide the current date and time, either set automatically, or set manually by the end user, for example, via the Settings menu, in order to successfully open secure web pages.

3.12.3. Default colors

The device MUST open web pages using the following default colors:

Property	Default color
Background color	White
Font color	Black

3.12.4. Memory

The target device MUST enable and define a browser memory limit of at least 200MB. This includes memory available to the browser to render an app and to allocate the memory the application requests.

3.12.4.1. Out of memory

The device MUST handle out-of-memory situations gracefully by releasing unused memory to make it available to the browser engine. If no memory can be released, then the device MUST show a warning message that there is insufficient memory on the device.

3.12.5. Storage

The device MUST ensure that the following minimum storage is available for apps:

- 64MB for localStorage ([\[26\]](#))
- 16MB for Temporary Storage ([\[26\]](#))
- 32MB for HTTP cache

The memory for the localStorage SHOULD be shared between all apps, and each app MAY use up to 10MB. The localStorage and sessionStorage SHOULD have a limit of 5MB per origin.

The memory for temporary storage SHOULD be shared between all apps, and each app MAY use up to 20% of the shared pool (3.2MB).

The device is NOT REQUIRED to support persistent storage from the Quota Management API ([27]).

The in-memory backend for the HTTP cache SHOULD be used if the access time to the file system on the device introduces the measurable performance penalty.

3.12.6. Cookies

The device MUST provide the possibility of storing at least 2000 cookies,; at least 180 cookies per domain, with at least 4096 bytes (for example, a total of 8MB cookie storage). The device MUST NOT remove any persistent cookies that have been accessed within the last 30 days.

3.12.7. Disconnect from network

The device MUST present a user-friendly message when an internet connection is not available. The network connection should be working again after the connection is restored (cable plugged-in or WiFi network available).

3.12.8. Cannot open URL

The device MUST present a user-friendly message when a URL is unreachable.

3.13. Accessibility

3.13.1 Text-to-speech

Text-to-Speech (TTS) is an accessibility feature enabling audible spoken words of the text strings shown in an application. Users with impaired vision disability benefit from using Text-to-Speech. Devices MAY choose to support TTS using WAI-ARIA web standards [35]. WAI-ARIA is a set of standards specifying how web applications can define the semantics of a web page, e.g. define that a link is a command, that an area is content, and navigation sequences between areas.

The hints provided by WAI-ARIA are used by the screen reader to enrich the experience and work accordingly to the expectations of web page authors building accessibility-enabled pages.

4. ABBREVIATIONS

AAC	Advanced Audio Coding
AC-3	(Dolby Digital) Audio Compression 3
ADTS	Audio Data Transport Stream
CENC	Common Encryption
CDM	Content Decryption Module
DASH	Dynamic Adaptive Streaming over HTTP
DASH-IF	DASH Industry Forum
DRM	Digital Rights Management

E-AC-3	(Dolby Digital) Enhanced Audio Compression 3
EBU-TT-D	EBU Timed Text format, part D - the format for the distribution of subtitles over IP
EME	Encrypted Media Extensions
HbbTV	Hybrid Broadcast Broadband TV
HDR	High Dynamic Range
HE-AAC	High-Efficiency Advanced Audio Coding
HLS	HTTP Live Streaming
HLG	Hybrid Log-Gamma
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDR	Instantaneous Decoder Refresh
ISO BMFF	ISO base media file format
KEYIDS	Stream-independent format for specifying a list of key ID(s) for DRM initialization
LC-AAC	Low-Complexity Advanced Audio Coding
LFE	Low Frequency Effects
MPD	DASH Media Presentation Description
MPEG	Moving Picture Experts Group
MPEG2-TS	MPEG-2 Transport Stream
MSS	Microsoft Smooth Streaming
PIFF	Protected Interoperable File Format
SHA	Secure Hash Algorithm
TLS	Transport Layer Security
TTF	TrueType Font
TTML	Timed Text Markup Language
WebM	WebM Stream Format
WebVTT	Web Video Text Tracks format
WOFF	Web Open File Format
WOFF2	WOFF File Format 2.0

5. REFERENCES

- [1] HLS, HTTP Live Streaming, RFC 8216 August 2017 <https://tools.ietf.org/html/rfc8216>
- [2] Smooth Streaming Protocol: <https://msdn.microsoft.com/en-us/library/ff469518.aspx>
- [3] DASH, Information technology — Dynamic adaptive streaming over HTTP (DASH): ISO/IEC 23009-1:2014
- [4] DVB MPEG-DASH Profile for Transport of ISO BMFF:
https://www.dvb.org/resources/public/standards/a168_dvb-dash.pdf
- [5] DASH-IF Guidelines for Implementation - Interoperability Points v3.2:
<http://dashif.org/wp-content/uploads/2015/12/DASH-IF-IOP-v3.2.pdf>
- [6] Encrypted Media Extensions - W3C Candidate Recommendation Working Draft 05 July 2016
<https://www.w3.org/TR/2016/CR-encrypted-media-20160705/>
- [7] Media Source Extensions - W3C Candidate Recommendation 05 July 2016 ,
<https://www.w3.org/TR/2016/CR-encrypted-media-20160705/>
- [8] PlayReady documents <https://www.microsoft.com/playready/documents/>
- [9] HTML 5.1 W3C Working Draft, 2 June 2016: <https://www.w3.org/TR/html51/>
- [10] "cenc" DRM Initialization Data Format <https://www.w3.org/TR/eme-initdata-cenc/>
- [11] "webm" DRM Initialization Data Format <https://www.w3.org/TR/eme-initdata-webm/>
- [12] "keyids" DRM Initialization Data Format <https://www.w3.org/TR/eme-initdata-keyids/>
- [13] Protected Interoperable File Format
<http://www.iis.net/learn/media/smooth-streaming/protected-interoperable-file-format>
- [14] Common Encryption Scheme [ISO/IEC 23001-7](https://www.iso.org/obp/ui/#iso:code:38100:23001-7)
- [15] Web Video Text Tracks Format (WebVTT) <https://w3c.github.io/webvtt/>
- [16] TTML text track format <https://www.w3.org/TR/ttml-imsc1/>
- [17] EBU-TT-D text track format <https://tech.ebu.ch/docs/tech/tech3380.pdf>
- [18] Definition of video VP9 levels <http://www.webmproject.org/vp9/levels>
- [19] High efficiency video coding - International Telecommunication Union (ITU-T) Recommendation for h.265 -
<https://www.itu.int/rec/T-REC-H.265>
- [20] Advanced video coding for generic audiovisual services - International Telecommunication Union (ITU-T) Recommendation for h.264 - <https://www.itu.int/rec/T-REC-H.264>
- [21] AV1 Bitstream & Decoding Process Specification <https://aomediacodec.github.io/av1-spec/av1-spec.pdf>
- [22] (Draft) VP9 Bitstream and Decoding Process Specification - <http://www.webmproject.org/vp9/>
- [23] DOM Level 3 Events specification - <https://www.w3.org/TR/DOM-Level-3-Events/>
- [24] Mozilla CA Certificate Policy -
<https://www.mozilla.org/en-US/about/governance/policies/security-group/certs/policy>
- [25] Chromium Remote Debugging protocol (DevTools) -
<https://developer.chrome.com/devtools/docs/debugger-protocol>
- [26] Web Storage API - https://developer.mozilla.org/en-US/docs/Web/API/Web_Storage_API

- [27] HTML5 Persistent Storage - https://developer.chrome.com/apps/offline_storage#persistent
- [28] CSS Values and Units Module Level 3, Chapter 6.4: Resolution units - <https://www.w3.org/TR/2016/CR-css-values-3-20160929/#resolution>
- [29] Guidelines for Implementation: DASH-AVC/264 Interoperability Points - <http://dashif.org/wp-content/uploads/2015/04/DASH-AVC-264-v2.00-hd-mca.pdf>
- [30] Key words for use in RFCs to Indicate Requirement Levels - <https://www.ietf.org/rfc/rfc2119.txt>
- [31] ETSI TS 102 796 V1.4.1 Hybrid Broadcast Broadband TV Technical Specification http://www.etsi.org/deliver/etsi_ts/102700_102799/102796/01.04.01_60/ts_102796v010401p.pdf
- [32] IETF RFC 5246 <https://tools.ietf.org/html/rfc5246>
- [33] IETF RFC 5289 <https://tools.ietf.org/html/rfc5289>
- [34] HbbTV 2.0.1 Specification with Errata #2 Integrated and Changes Marked https://www.hbbtv.org/wp-content/uploads/2018/02/HbbTV-SPEC20-00061-000-HbbTV_201_with_errata_2_integrated.pdf
- [35] Accessible Rich Internet Applications (WAI-ARIA) 1.1 <https://www.w3.org/TR/wai-aria/>

6. ANNEX A (informative). Media type strings for video and audio codecs

This section specifies codecs and media identifiers for audio and video codecs that MUST be supported with the HTML5 <video> tag. It also provides identifiers for some examples of combinations of those codecs and media containers. The Codec ID strings' list is not comprehensive, because not all the rules from section [4.6. Video and audio formats](#) are shown below.

6.1. MP4 video and audio

6.1.1. H.264 profiles

Profile	Level	Codec ID string [rfc6381]
Baseline	3.1	avc1.42E01F avc3.42E01F
Main	3.1	avc1.4D401F avc3.4D401F
Main	4	avc1.4D4028 avc3.4D4028
High	4	avc1.640028 avc3.640028

6.1.2. H.265/HEVC profiles

Profile	Level	Constraints	Codec ID string
Main	3.1	None	hev1.1.6.L93.00 hvc1.1.6.L93.00

	4	None	hev1.1.6.L120.00 hvc1.1.6.L120.00
	4.1	None	hev1.1.6.L123.00 hvc1.1.6.L123.00
	5.1	None	hev1.1.6.L153.00 hvc1.1.6.L153.00
Main 10	4.1	None	hev1.2.4.L123.00 hvc1.2.4.L123.00
	5.1	None	hev1.2.4.L153.00 hvc1.2.4.L153.00

6.1.3. AV1 profiles

Profile	Level	Constraints	Codec ID string
Main	2.0	None	av01.0.00M.08
	2.1	None	av01.0.01M.08
	3.0	None	av01.0.04M.08
	3.1	None	av01.0.05M.08
	4	None	av01.0.08M.08
	4.1	None	av01.0.09M.08

6.1.4. AAC profiles

Profile name	Codec ID string
AAC-LC	mp4a.40.2
HE-AAC v1 (SBR)	mp4a.40.5
HE-AAC v2 (SBR+PS)	mp4a.40.29

6.1.5. MP3 (MPEG-1 Layer III) profiles

Codec ID string
mp4a.69
mp4a.6B

6.1.6. Dolby profiles

Profile name	Codec ID string
AC-3	ac-3

	mp4a.a5
E-AC-3	ec-3
	mp4a.a6

6.1.7 Combination examples of media type strings

Video codec	Video profile	Audio codec	Audio profile	Media type string
H.264 level 3.1	baseline	AAC	aac_he	video/mp4; codecs="avc1.42E01F, mp4a.40.5"
			aac_lc	video/mp4; codecs="avc1.42E01F, mp4a.40.2"
		MP3		video/mp4; codecs="avc1.42E01F, mp4a.69"
				video/mp4; codecs="avc1.42E01F, mp4a.6B"
H.264 level 3.1	main	AAC	aac_he	video/mp4; codecs="avc1.4D401F, mp4a.40.5"
			aac_lc	video/mp4; codecs="avc1.4D401F, mp4a.40.2"
		MP3		video/mp4; codecs="avc1.4D401F, mp4a.69"
				video/mp4; codecs="avc1.4D401F, mp4a.6B"
H.264 level 4.0	main	AAC	aac_he	video/mp4; codecs="avc1.4D4028, mp4a.40.5"
			aac_lc	video/mp4; codecs="avc1.4D4028, mp4a.40.2"
		MP3		video/mp4; codecs="avc1.4D4028, mp4a.69"
				video/mp4; codecs="avc1.4D4028, mp4a.6B"
H.264 level 4.0	high	AAC	aac_he	video/mp4; codecs="avc1.640028, mp4a.40.5"
			aac_lc	video/mp4; codecs="avc1.640028, mp4a.40.2"
		MP3		video/mp4; codecs="avc1.640028, mp4a.69"
				video/mp4; codecs="avc1.640028, mp4a.6B"

6.2. WebM video and audio

Video format	Video codec	Audio codec	Media type string
	VP9 profile 0	Opus	video/webm; codecs="vp9, opus"

			video/webm; codecs="vp9.0, opus"
	VP9 profile 2	Opus	video/webm; codecs="vp9.2, opus"

6.3. Audio only

Audio codec	Audio profile	Media type string
AAC	aac_he	audio/mp4; codecs="mp4a.40.5"
	aac_lc	audio/mp4; codecs="mp4a.40.2"
MP3		audio/mp4; codecs="mp4a.69"
		audio/mp4; codecs="mp4a.6B"
OGG Vorbis		audio/ogg; codecs="vorbis"