[(↦ /author/stefanlederer)](/author/stefanlederer)

ABOUT THE AUTHOR

[Stefan Lederer](https://twitter.com/slederer) (↦ https://twitter.com/slederer) is co-founder of [bitmovin](http://www.bitmovin.com) (↦ http://www.bitmovin.com), which is providing online streaming solutions for high-quality adaptive bitrate video, with a focus on MPEG-DASH & … [More about Stefan Lederer …](/author/stefanlederer) (↦ /author/stefanlederer)

HTML5 Media Source Extensions: Bringing Production Video To The Web

QUICK SUMMARY ↬ *In the last decade, plugins such as Flash and Silverlight have enabled a rich consumption of video in browsers, powering popular services such as YouTube and Netflix. However, this approach has shifted towards HTML5 over the last few years. Almost two years ago, the W3C published the final recommendation of the HTML5 spec, which came with a new set of HTML elements and APIs, especially for video. Some of them aim for more semantics in web pages but don't introduce new features. Others extend the possibilities of the web and enhance the possibilities for developers* ***without the need for plugins*** *such as Adobe Flash, Microsoft Silverlight or Java.*

IN THE LAST DECADE, PLUGINS SUCH AS FLASH AND SILVERLIGHT HAVE enabled a rich consumption of video in browsers, powering popular services such as YouTube and Netflix. However, this approach has shifted towards HTML5 over the last few years.

Almost two years ago, the W3C published the [final recommendation of the HTML5 spec](https://www.w3.org/TR/html5/) (↦ https://www.w3.org/TR/html5/), which came with a new set of HTML elements and APIs, especially for video. Some of them aim for more semantics in web pages but don't introduce new features. Others extend the possibilities of the web and enhance the possibilities for developers **without the need for plugins** such as Adobe Flash, Microsoft Silverlight or Java.

Further Reading On SmashingMag:

- Providing A Native Experience With Web Technologies (↦ https://www.smashingmagazine.com/2014/10/providing-a-native-experience-with-web-technologies/)
- Making A Complete Polyfill For The HTML5 Details Element (↦ https://www.smashingmagazine.com/2014/11/complete-polyfill-html5-details-element/)
- A Beginner's Guide To Progressive Web Apps (↦ https://www.smashingmagazine.com/2016/08/a-beginners-guide-to-progressive-web-apps/)
- The HTML5 Logo: What Do You Think? (↦ https://www.smashingmagazine.com/2011/01/the-html5-logo-what-do-you-think-opinion-column/)

This is especially important because, for example, Google has announced (↦ http://blog.chromium.org/2013/09/saying-goodbye-to-our-old-friend-npapi.html) the removal of NPAPI (an API used by these plugins), as has Firefox (↦ https://blog.mozilla.org/futurereleases/2015/10/08/npapi-plugins-in-firefox/), and Microsoft advocates for plugin-free browsing (↦ http://msdn.microsoft.com/en-us/library/ie/hh968248.aspx). Although these vendors still provide a Flash player, it is probably only a matter of time before they don't. Furthermore, browsers on mobile devices are a step beyond because most of them do not support plugins and have no Flash player.

Let's look at some of the new HTML5 elements and what they improve for video:

- `<canvas>` provides scripts to render graphs, game graphics and more. This is sometimes referred to as the Canvas JavaScript API. The `canvas` element can also be used with WebGL to render 2D and 3D graphics, using the graphics card's GPU.
- `<video>` enables out-of-the-box playback of video, which is really awesome. This finally makes plugin-free multimedia on the web a reality. In fact, browser vendors seem to agree on a single format — MPEG-4/H.264 (↦ http://caniuse.com/#search=mp4), which is universally supported in modern browsers, with a notable exception of Opera Mini.
- `<audio>` enables out-of-the-box playback of audio content on a web page. As with video, the decision of which container formats and codecs to support is left to browser vendors.
- `<track>` can be used for timed text tracks, such as subtitles and captions in a video. WebVTT files are supported out of the box.

Most of the new elements have been known about and used for a while in HTML5 video player code (↦ https://www.streamingvideoprovider.com/streaming-video-players.html) because they are implemented in all modern browsers. The specification is stable. Still, the W3C has a lot of work to do left.

For me, the most important standard the W3C is working on is the "Media Source Extensions (↦ http://www.w3.org/TR/media-source/)" (MSEs) standard, which currently has the status of "Candidate

Recommendation." This JavaScript API allows us to generate media streams for the `<video>`, `<audio>` and other elements, enabling adaptive streaming standards such as MPEG-DASH in pure HTML5 and JavaScript.

Another interesting one is the "Encrypted Media Extensions *(↦http://www.w3.org/TR/encrypted-media/)*" standard, which allows playback of protected content in HTML5 and JavaScript. However, this is currently a "Working Draft" and will take a while to finalize.

We welcome the new standard and look forward to the time when we don't need a Flash player or plugin, when multimedia can be viewed on virtually any device with a single implementation.

Why MPEG-DASH?

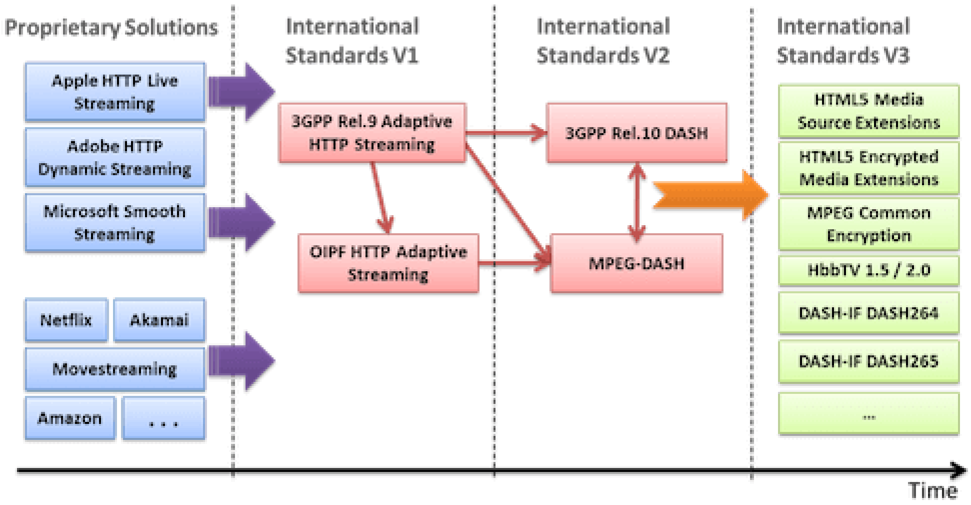
Let's look into the MPEG-DASH *(↦https://www.bitcodin.com/blog/2015/04/mpeg-dash/)* streaming format and why it is used in HTML5. MPEG-DASH (the DASH being short for dynamic adaptive streaming over HTTP) is an international, vendor-independent standard ratified by MPEG and ISO (ISO/IEC 23009-1). Previous adaptive streaming technologies — such as Apple HLS *(↦https://www.bitcodin.com/blog/2015/07/apple-http-live-streaming-hls/)*, Microsoft Smooth Streaming *(↦https://www.bitcodin.com/blog/2015/05/microsoft-smooth-streaming/)* and Adobe HDS — were released by vendors with limited support for vendor-independent streaming servers or for playback clients. A vendor-dependent situation was clearly not desirable, and so standardization bodies started a harmonization process, resulting in the ratification of MPEG-DASH in 2012.

These are the goals and benefits of MPEG-DASH in a nutshell:

- Reduce startup delays as well as buffering and stalls during video playback.
- Continue adaptation to the bandwidth situation of the client.
- Use client-based streaming logic to enable the highest scalability and flexibility.
- Use existing and cost-effective HTTP-based CDNs, proxies and caches.
- Efficiently bypass NATs and firewalls through the use of HTTP.
- Enable common encryption through the signaling, delivery and use of multiple concurrent DRM schemes from the same file.
- Enable simple splicing and (targeted) ad insertion.
- Support "trick mode" efficiently.
- And much more! *(↦https://www.bitcodin.com/blog/2015/03/mpeg-dash-vs-apple-hls-vs-microsoft-smooth-streaming-vs-adobe-hds/)*

In recent years, MPEG-DASH has been integrated in new standardization efforts — such as the HTML5 MSEs, which enable DASH playback via HTML5's `video` and `audio` tags, as well as the HTML5 Encrypted Media Extensions, which enable DRM-protected playback in web browsers. Furthermore, DRM-protection with MPEG-DASH is harmonized across different systems with MPEG-CENC (for common encryption); and MPEG-DASH playback on different smart TV platforms is enabled via integration with Hybrid Broadcast Broadband TV (HbbTV 1.5 (↦ *http://www.etsi.org/deliver/etsi_ts/102700_102799/102796/01.02.01_60/ts_102796v010201p.pdf*) and HbbTV 2.0 (↦ *https://www.hbbtv.org*)).

Also, usage of the MPEG-DASH standard has been simplified by industry efforts around the DASH Industry Forum and its DASH-AVC/264 recommendations, as well as by forward-looking initiatives such as the DASH-HEVC/265 recommendation on the usage of H.265/HEVC within MPEG-DASH.


Proprietary Solutions
International Standards V1
International Standards V2
International Standards V3
Apple HTTP Live Streaming
Adobe HTTP Dynamic Streaming
Microsoft Smooth Streaming
Netflix
Akamai
Movestreaming
Amazon
...
3GPP Rel.9 Adaptive HTTP Streaming
OIPF HTTP Adaptive Streaming
3GPP Rel.10 DASH
MPEG-DASH
HTML5 Media Source Extensions
HTML5 Encrypted Media Extensions
MPEG Common Encryption
HbbTV 1.5 / 2.0
DASH-IF DASH264
DASH-IF DASH265
...
Time

(↦https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/c58723c3-f932-4904-af47-56e0253496ef/ecosystem-standards-opt.png)

Ecosystem of video streaming standards (Image: Bitcodin (↦https://www.bitcodin.com/blog/2015/04/mpeg-dash/)) (View large version (↦https://cloud.netlifyusercontent.com/assets/344dbf88-fdf9-42bb-adb4-46f01eedd629/c58723c3-f932-4904-af47-56e0253496ef/ecosystem-standards-opt.png))

Today, MPEG-DASH is being deployed more and more (↦*https://bitmovin.com/status-mpeg-dash-today-youtube-netflix-use-html5-beyond/*), accelerated by services such as Netflix and Google, which have recently switched to this new standard. With these two major sources of traffic, MPEG-DASH already accounts for 50% of total Internet traffic.

How Do The MSEs Work?

Now let's look into the details of the MSEs and how they can be used by developers. The MSEs are a specification that extend the `HTMLMediaElement` to allow JavaScript to dynamically construct media streams for `audio` and `video` tags. This wasn't possible before because these tags were only able to access full files (i.e. MP4 files). This approach is also called progressive streaming, or progressive downloading, because the media files are downloaded and played at the same time, enabling pseudo-streaming.

However, this brings with it **poor seeking capability** and no possibility to adapt the video and audio quality to the bandwidth situation of the user. By constructing media streams in JavaScript as inputs for the `audio` and `video` tags, developers can now dynamically adapt a media stream to the user's context, thus improving the streaming experience.

As mentioned, MPEG-DASH is the streaming format of choice for the MSEs. So, let's look at the steps necessary to build an HTML5 MSE-based video player:

01 Download and parse the manifest file — called an MPD in MPEG-DASH — which describes the video stream's details, such as the number of the video stream's quality levels and resolutions, audio languages and subtitles, as well as the name of the media segments and files on the HTTP-based origin server or CDN.

02 Estimate the available bandwidth on the client device, choose the appropriate video quality to achieve buffer-less streaming, and download the media segments in JavaScript.

03 Hand over the downloaded media segments to the MSE buffer in JavaScript.

04 Decode and render the video via the `video` tag, typically in the hardware.

This is how HTML5-based adaptive streaming players work, as used by Netflix and YouTube. There are already quite mature solutions out there, which makes it easy for developers and content providers to switch to adaptive bitrate streaming in HTML5, such as the DASH-IF open-source project dash.js *(↦ https://github.com/Dash-Industry-Forum/dash.js/wiki)* and the Bitdash HTML5 player *(↦ http://www.dash-player.com/)*.

MPEG-DASH content generation is also straightforward and supported by open-source tools such as x264 and MP4Box *(↦ http://www.dash-player.com/blog/2014/11/mpeg-dash-content-generation-using-mp4box-and-x264/)*, as well as by commercial encoding services such as Bitcodin *(↦ http://www.bitcodin.com/)*

However, use of the MSEs is not limited to MPEG-DASH. More and more projects (including hls.js *(↦ https://github.com/dailymotion/hls.js/tree/master)*) and players (including Bitdash *(↦ http://www.dash-player.com/)*) support Apple's HLS format in HTML5 using the MSEs. They do this by trans-multiplexing the HLS media segments — which are MPEG2-TS containers — to the ISO Base Media File Format required by HTML5 and MPEG-DASH.

Encrypted Media Extensions For DRM

Major changes are currently happening in the DRM market, caused by the imminent drop of NPAPI plugins — such as Silverlight, which led to the drop of the leading DRM system, PlayReady — from Chrome and Firefox. This puts nearly all premium content providers in a tough situation because they will have to switch technologies and find a future-proof solution.

Premium streaming media publishers will not be able to rely on Microsoft's PlayReady DRM to secure their content in Chrome and Firefox on PC and Android devices. They will have to re-evaluate their content protection and streaming platform strategy and will have to find a future-proof solution and then switch technologies shortly.

For many content providers, MPEG-DASH has emerged as the technology of choice. DASH projects have rolled out at an accelerating pace, and MSEs and Encrypted Media Extensions (EMEs) with Widevine DRM look to be the most viable alternative. Also, MPEG-CENC makes it possible to support separate DRM systems with just one version of protected content, and EMEs are based on the MSEs for MPEG-DASH-based content.

So, this combination of different DRM systems — for example, Widevine Modular for Chrome and Android, Microsoft PlayReady for Internet Explorer and Edge, and Adobe's Primetime for Firefox — for one version of a piece of content gives content providers additional incentive to move towards MPEG-DASH as an international standard, given its flexibility with streaming, DRM and CDNs.

Browser Support For MSEs And EMEs

After a couple of years of slow adoption by browser vendors of HTML5 and the MSEs in particular, we are now seeing a majority of them supporting it. This also goes for EMEs, although in this case each vendor is going for a different DRM system, and the ecosystem is a bit more differentiated.

However, to reach 99% of users, we have to have a video streaming setup that also supports browsers that don't support the MSEs, in particular old browser versions and Safari on iOS. Old browsers can be served easily using a Flash-based player, which can play back the same MPEG-DASH content that is used by the MSEs, as shown by the Bitdash player *(↦ http://developer.dash-player.com/supported-formats-devices)*. To support iOS devices, we have to use Apple's streaming format, called HLS, which Apple is mandating for HTML5. Open standards such as the MSEs are not supported by Apple, although they are supported on Safari on OS X.

The following matrix gives an overview of the status of MSE and EME support across browsers and platforms today (courtesy of Bitmovin *(↦ http://www.dash-player.com/browser-capabilities/)*):

Environment	Player Technology	Media	DRM

Environment	Player Technology	Media	DRM
Chrome	HTML5 MSE	MPEG-DASH	Widevine Modular
Internet Explorer 11 Windows 8.1	HTML5 MSE	MPEG-DASH	PlayReady
Internet Explorer (other)	Flash, Silverlight	MPEG-DASH	ClearKey, PlayReady
Edge	HTML5 MSE, HTML5 HLS	MPEG-DASH, HLS	PlayReady, AES HLS
Firefox	HTML5 MSE	MPEG-DASH	Adobe
Safari	HTML5 MSE, HTML5 HLS	MPEG-DASH, HLS	Fairplay, AES
Android: Web > v4.1	HTML5 MSE, HTML5 HLS	MPEG-DASH, HLS	Widevine Modular
Android: app	Google's Exoplayer	MPEG-DASH, HLS	Widevine Modular
iOS: web	HTML5 HLS	HLS	AES
iOS: app	native HLS support	HLS	Fairplay, AES
smart TV	Native MPEG-DASH support or HTML5 MSE (e.g. Tizen)	MPEG-DASH or HLS	Device-dependent
HbbTV (1.5)	native MPEG-DASH support	MPEG-DASH	device-dependent

The Future Of HTML5 Video

New media codecs are pushing into the market, making video compression even more efficient, which is especially important for higher-quality formats such as 4K and UHD and for streaming to mobile devices. The most common codec is **HEVC/h.265**, which could be the default codec in a couple of years from now (if the patent situation doesn't mess that up). And it will also utilize the browser's built-in MSEs for playback and use MPEG-DASH as the streaming format, which shows the flexibility of this open standard.

Developers of video players just have to perform some simple adaptations, like changing the codec's attribute when creating the SourceBuffer; and, if the underlying browser supports HEVC decoding (most likely done by a hardware decoder), then you will be able to watch your HEVC MPEG-DASH streams in HTML5! We've successfully tested with browsers, such as Microsoft Edge (↦ *http://www.windowscentral.com/microsoft-windows-10-will-support-hevc-video-standard)*, which comes with HEVC support. Also, Google recently announced support *(↦ https://code.google.com/p/chromium/issues/detail?id=454948#c14)* in its Chromium browser.

Nevertheless, HEVC is not yet available for the vast majority of Internet video assets, and only a few devices are capable of decoding it. And, of course, it is not the only codec in town. The open and royalty-free video-encoding format VP9 (the successor to VP8) aims to have an even better encoding efficiency and is already supported by popular browsers such as Google Chrome and Microsoft Edge, and this codec is compatible with MSE, too. However, we cannot foresee which codecs will find their way into our daily streaming routine. But whether it's VP8/9, AVC or HEVC, the MSEs and MPEG-DASH are ready!

An upcoming trend is 360-degree video, which is pretty straightforward to use in HTML5. Developers could make use of the adaptive streaming support of MSEs and just add a JavaScript or WebGL rendering layer for a 360-degree experience on top of it. Recently, I gave a talk *(↦ http://www.slideshare.net/bitmovin/build-a-netflix-for-360-vr-video-using-html5-dash-javascript-webgl)* about this topic and on how to build a Netflix-like service for virtual reality using HTML5, JavaScript, DASH and WebGL.

Conclusion

I hope this article has given you a good overview of the state and future of video on the web. The MSEs and EMEs are big steps toward an ecosystem of open standards for video on the web, replacing plugins such as Flash and Silverlight. Furthermore, HTML5 is getting on the platforms of choice in today's multi-platform world, including desktop, mobile and smart TV environments.

Along with streaming standards such as MPEG-DASH, content providers can have a **unified video solution across platforms and devices**. They can enhance the user experience through adaptive streaming formats, which prevent buffering, decrease loading times and provide the best possible quality for each user's bandwidth and device situation.

(rb, vf, al, il)

Front page image credits: Accessible HTML5 Video Player (↦ http://paypal.github.io/accessible-html5-video-player/)

Founded by Vitaly Friedman and Sven Lennartz. 2006–2020.

Smashing is proudly running on Netlify.

Fonts by Latinotype.

Cats can be forgetful, but we are not.
Thanks for being truly smashing — yet again.
www.smashingmagazine.com