Premium Video Content Protection Strategies with Common Media Application Format (CMAF)

Table of contents

Executive Summary

Will there be a single streaming format for protected video content? The industry is getting close, but we're not there yet. In this article, we explore the progress that's been made and what's holding the industry back from something that would make premium video delivery so much easier.

The Promise and the Challenge

The HTML5 video standards have been evolving for the last years to a point where they can be leveraged for **premium video experiences**. It evolved from the early struggle of finding the right codec to work across all browsers, to the gradual adoption and the maturity of Media Source Extensions (MSE) and Encrypted Media Extensions (EME) for DRM protected content.

Digital rights management at scale is essential for premium video content, which means we need a single streaming format. For instance, companies can save big on storage costs with a single streaming format because storing the content in a premium video library in more than one format increases storage costs. As an example, a video catalog of 200,000 titles highly benefits from a single format purely because of storage costs. The same is true for live streaming, where a single format leads to higher caching efficiencies inside and outside of the content delivery network (CDN), which then results in an overall higher video quality with content cached closer to the end user. In addition, the live encoding / packaging requirements are higher with multiple formats. This is why the industry as a whole is looking for the single format that will play across all screens.

The Single Streaming Format

For non-DRM protected content the single format has been a reality, and it is Apple's HTTP Live Streaming (HLS). The reason why the format prevailed is because it is the only streaming format that traditionally played on iOS / TVOS devices and is approved for the cellular networks, combined with Apple's dominance in the devices market (nobody can afford to exclude iOS from their device strategy), and the public release of the **HLS specifications** to the Internet Engineering Task Force (IETF) early on. The specs allowed every device manufacturer to support HLS streaming and play the same content as iOS.

Since the specs can be implemented differently, devices have quite a bit of quality variance for HLS playback. This is where solutions like Adobe Primetime come in to normalize the implementation quality and make it work for premium video.

In parallel, **MPEG-DASH** evolved as an independent standard with the promise to unify the industry around one format, which succeeded everywhere except Apple devices.

The Obstacle of DRM

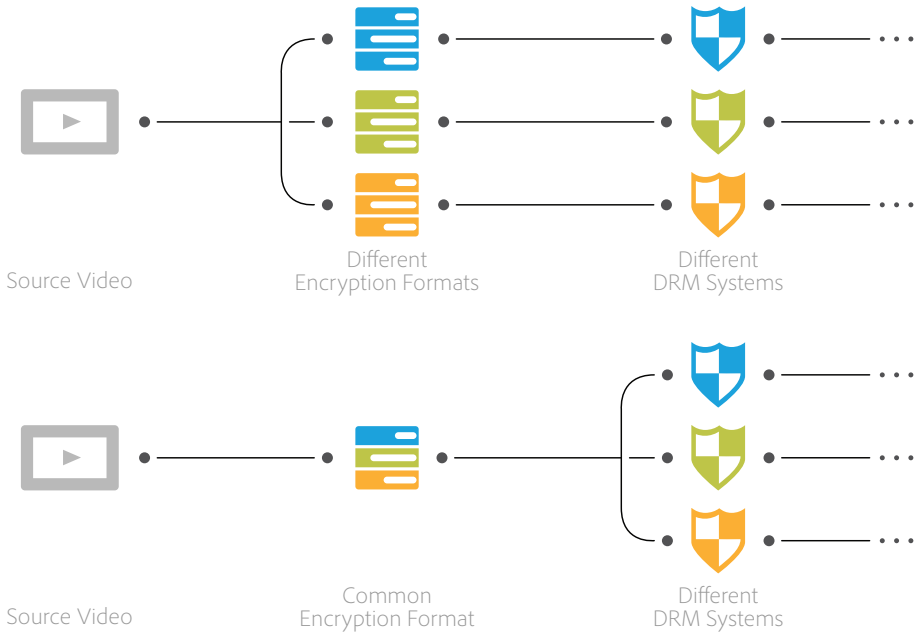
There are two main challenges with the single format and Digital Rights Management:

1. The use of multiple DRM systems is a studio requirement to reach all HTML5 browsers and devices.

DRM fulfills a critical purpose: to protect content from illegal redistribution and against violations to the terms of use of content owners, which are often movie studios. Unfortunately there is not a single-vendor DRM solution available across all HTML5 browsers and devices. Even though the HTML5 EME interface is standardized, it does not include the ability to use a DRM of choice. The DRM technology is included as part of the browser itself, and each browser supports a different DRM. For instance:

- Chrome supports Widevine
- Internet Explorer supports PlayReady
- Firefox supports Adobe Access and Widevine
- Safari supports FairPlay

To avoid having to package a different stream for each DRM system, the concept was introduced to use multi DRM systems with a single streaming format such as Common Encryption (CENC) for MPEG-DASH, and the unofficial HLS equivalent, HLS with sample-based encryption.



2. DRM systems in the market are not unified on the segment container format and the encryption mode.

The following simplified visualization demonstrates the layers involved with modern HTTP streaming formats whereby the adaptive protocol layer is dependent on the container layer, which is dependent on the encryption layer, which is dependent on the video codec layer.

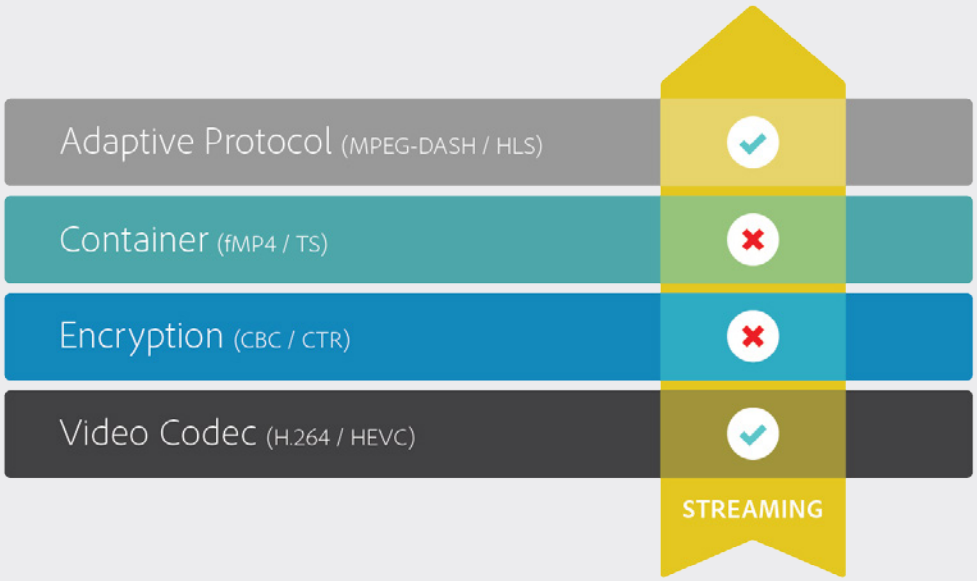


Figure 1: Layers of modern HTTP streaming formats prior to CMAF.

1. **Adaptive Protocol**

   This can be often parsed by the player itself and is not tied to a specific platform, with the main exception of iOS (HLS)

2. **Container**

   This can be often parsed by the player itself and is not tied to a specific platform, with the main exception of iOS (Only TS until recently, now fMP4 as well)

3. **Encryption**

   The two common encryption modes are CBC (Cipher Block Chaining) and CTR (Counter Mode). In case of a hardware supported DRM, the mode is tied to a specific DRM on a platform.

4. **Video Codec**

   In most cases H.264 today, with HEVC and VP9 emerging for 4k content. For best performance, the underlying hardware should support the codec directly.

While the container + streaming formats can be often rewritten by the client, and platforms have unified around H.264, the DRM systems have chosen different encryption modes, which leads to a big challenge: streaming providers must support two encryption formats because there is not an agreed upon encryption mode.

The modes supported in hardware are either CTR or CBC, as shown here:

- PlayReady -> Dash/fMP4 -> CTR
- Widevine -> Dash/fMP4 -> CTR
- FairPlay -> HLS/TS -> CBC

Since the encryption mode is hardwired into the underlying DRM system, it is not possible to transmux the formats on the client securely in the application layer/player. This then impacts the container layer and the protocol layer due to the dependencies outlined above.

The Introduction of Common Media Application Format (CMAF)

The challenge of at least requiring two different container formats, and therefore streaming protocols, has been identified as a problem by MPEG, and efforts have to be started to standardize on a single fMP4 based format, CMAF.

The core mission of CMAF is to evolve toward a common format. **The requirements document on CMAF** further explains, "Several MPEG technologies have been adopted for the majority of video delivered over the internet and other IP networks (cellular, cable, broadcast, etc.). Various organizations have taken MPEG's core coding, file format and system standards, and combined them into their own specifications for their specific applications. While these specifications share major common parts, their differences result in both unnecessary duplication of engineering effort, and duplication of identical content in slightly different formats. The industry would benefit if application consortia could reference a single MPEG specification (a "common format") that would allow a single media encoding to work across many applications and devices."

CMAF is being worked on by Apple, Adobe, and many others in MPEG. Apple's new container can be seen as a preview of what CMAF could achieve. CMAF defines an ISOBMFF container that is compatible both with Apple's new container and with MPEG-DASH, using CENC for encryption. It also includes media profiles that define a common set of video and audio codecs, as well as closed caption formats. Finally it includes a model for mapping container constructs into a generic adaptive switching framework. This last part is intended to allow CMAF to fit easily into both the HLS and MPEG-DASH streaming protocols.

CMAF creates a recipe for what is required by a service provider when delivering video services.
Service providers must:

- Provide video in H.264/AVC in both CTR and CBC encryptions.
- Provide audio in stereo AAC (and optionally in multi-channel).
- Provide captions in both WebVTT and ISMC1 formats.
- Provide an HLS and a DASH manifest.

This is not the simple, single-format solution the industry has been looking for, but it is actually a big step forward. CMAF is the first time that some of the tectonic divisions in the streaming ecosystem (CBC versus CTR, HLS versus DASH) have been addressed in a single framework. It's a little complicated, but it's also consistent.

Apple's Big Move: HLS + fragmented MP4 (fMP4)

Clearly, the streaming ecosystem has been trying to unify around a common format with MPEG-DASH, but the exclusion of Apple's support was a challenge. This changed with the WWDC 2016 announcement of Apple starting to support fMP4 as an additional container format for HLS, and their involvement in CMAF, which is trying to develop a standard fMP4 container.

fMP4 can support both CBC and CTR, and therefore is compatible with most hardware DRM systems.

Apple's recent announcement of support for HLS with a fragmented MP4 media container in iOS and MacOS has led many people to ask whether the streaming industry has finally reached the goal of "one format to reach all devices". While the new container support is welcome, and will certainly simplify workflows for video service providers, there are still many challenges for those building out streaming video systems.

To understand some of these challenges, it is useful to look at Apple's new container support, and how it relates to long-running standardization efforts such as MPEG-DASH. Apple's fMP4 container is based on what is known as the ISO Base Media File Format (ISOBMFF), as defined in ISO/IEC 14496-12 (commonly called just "part 12"). MPEG-DASH also includes support for an ISOBMFF container, meaning that Apple's container and standard DASH containers are built on the same core technology. This being said, there are some key differences (such as support for timed metadata in the form of 'emsg' boxes) that make Apple's container and the DASH container different.

Apple's container also utilizes CENC as defined in the **ISO 23001-7** standard. Specifically, Apple supports a mode of CENC, an AES-CBC mode known as 'cbcs'. This differs from the most common profiles of MPEG-DASH (such as those proposed by SCTE, 3GPP, and DVB), which are all based on AES-CTR. So while Apple's container and the common DASH container are based on the same encryption specification, they use that specification in different ways that are not compatible.

On the surface, these differences seem to limit the usefulness of Apple's new container support. In truth, Apple's container is a huge first step in a broader industry movement. And while this initiative may not result in the ideal of "one format to reach all devices", it promises to provide a reasonable blueprint for service providers and device manufacturers, allowing the streaming industry to scale.

As of today, neither Apple, Microsoft, or Google have publicly made any commitments around ***when*** they will converge on CBC versus CTR. So, a truly single format for DRM is not the reality yet.

What will CMAF mean for DASH?

As described above, CMAF is designed to work with DASH, and to define a model where DASH and HLS can be used simultaneously with the same set of packaged content. It will be practical to use them both because the manifest layer of these protocols is lightweight (a text file, versus high bitrate encoded media). By allowing DASH to coexist with HLS, rather than competing with or supplanting it, CMAF gives service providers an on-ramp to DASH. Adobe continues to invest heavily in DASH, both by working on the standard and by implementing it in our products. We see CMAF as a positive extension of that investment.

What does CMAF mean for Adobe Primetime?

CMAF, with all its promise and all its complexity, shows why robust technology stacks will continue to be required to support video service providers. For example, cross-device video playback technology such as Adobe Primetime can manage the details of which encryption, closed caption format, etc. is best for a given device.

One key benefit of the Adobe Primetime player architecture is that it handles media container format parsing, rather than relying on device facilities. This means that Adobe Primetime could in a future release provide HLS + fragmented MP4 support on non-Apple devices, without having to wait for the device vendor to provide this capability. This would be a low level video engine providing support for DASH / HLS in combination with CMAF across most devices. In addition, while a common encryption is helpful when preparing content for delivery, it is still necessary to manage the platform-specific DRM license management for each device. Adobe Primetime also offers a DRM service that handles this for you.

Overall, we are getting pretty close to a truly single DRM protected format - with one exception: the encryption mode, which is tied to the DRM system of the hardware.

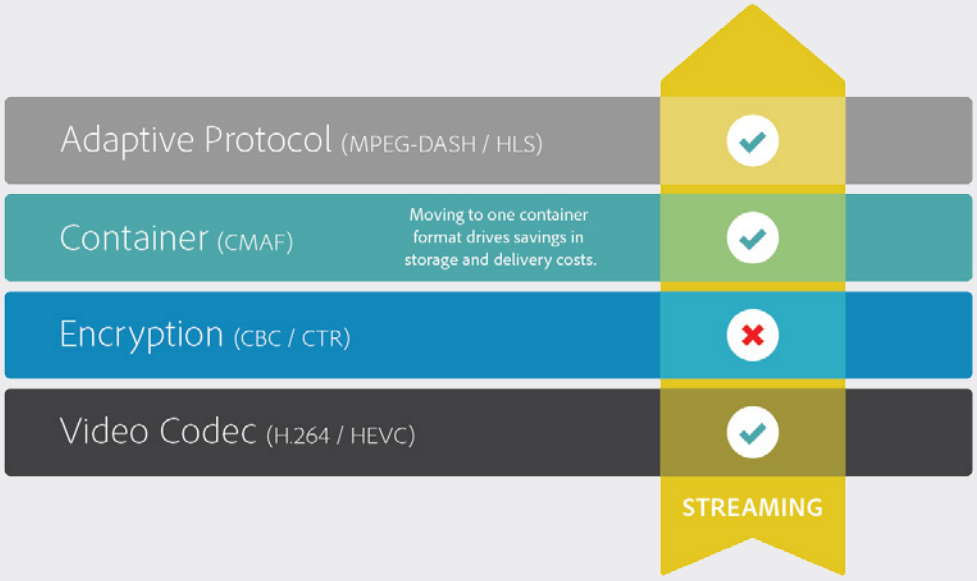


Figure 2: Layers of modern HTTP streaming formats with CMAF. Once everything is compatible, a single format will be possible.

Perhaps the most important benefit we see in CMAF for Adobe Primetime is that it is creating a clear path from HLS to DASH. Many of our customers have invested heavily in HLS-based workflows, particularly around ad insertion. Spinning up the same for DASH is painful. Further Adobe Primetime investments in CMAF could allow those customers to first switch to a DASH-compatible packaging, while keeping their ad monetization processes in place. Later, they can choose to switch to a DASH-based ad workflow.

Conclusion

A single streaming format without DRM has been a reality with HLS in the past, and will now also be possible with fMP4 with Apple's support. Unfortunately it is not yet possible when DRM content protection is required due to incompatible encrypted modes. Nevertheless, it's a promising new initiative in the industry and heading in the right direction, and deserves close attention going forward.

For more information
www.adobe.com

Adobe

Adobe Systems Incorporated
345 Park Avenue
San Jose, CA 95110-2704
USA
www.adobe.com