Commercial Content in Browsers

Jan Lindquist jan.lindquist@ericsson.com
Mats Cedervall, mats.m.cedervall@ericsson.com

Abstract

The current and future situation for service providers that offer commercial, DRM protected, content via browsers is discussed

Background

Content Providers require DRM to protect their content. Service Providers that deliver the content have to adopt a media player and DRM combination that the Content Providers approve of. Traditionally the approach that Service Providers have used is one of the following:

- Use a Multimedia framework that has support for an approved DRM scheme. Examples of this is Silverlight with PlayReady and Adobe® Access in Flash®
- Use a browser plugin that handles both DRM and media playback, either self-developed or using a 3'rd party. Plugin architectures that have been used for this include NPAPI and ActiveX
- Native client

Legacy plugin architectures have suffered from poor security and stability and at the same time W3C technologies like HTML-5 offer similar capabilities that multimedia plugins have. Because of this several major browser vendors have announced that they are dropping or replacing plugin support [1], [2].

Recently W3C has defined EME (Encrypted Media Extensions [3]). EME is a proposal that extends HTMLMediaElement providing APIs to control playback of protected content. EME requires the presence of one or more Content Decryption Modules (CDMs) that are integrated in some way with the browser. The CDM is trusted by the copyright holders to hide certain secret data and prevent manipulation from the user. The integration of the CDM is however not defined by W3C and it is up to the browser vendor what DRM Scheme is used and how it is integrated.

New, more secure, plugin architectures have been developed, e.g. NACL/PPAPI [4], but currently it is questionable if it meets multimedia requirements and only one browser vendor currently supports it

The options for the Service Providers in the future include:

- Deploy Multi-vendor DRM support tailored for each browser with no plugin support but with EME support with DRM choices made by browser vendors. Plugins for browsers that support plugin
- Use native app potentially based on HTML (e.g. Webkit) and application provided DRM.
- Support only one or a few browsers with either EME or plugin support.

None of these approaches are attractive or cost effective to Service Providers and the end users experience will likely suffer as a result.

Browser Positions

Chrome
NPAPI support will be dropped September 2014. Will support new API PPAPI instead, which will be reachable from Native Client framework or from integrated components (e.g. Flash). EME support with Widevine DRM.

Internet Explorer
Future versions of internet explorer will not have support for plugins, in particular Metro. EME support with PlayReady DRM [2].

Firefox
No statement about future NPAPI support. Will require extra user interaction to use plugins.

Safari and Opera
No statement has been found at the time of writing this article. In Safari when using plugins it asks for user consent.

Chromecast
DRM support for PlayReady and Widevine.

Conclusion

The future situation for commercial content service providers is anti-competitive because of the nature of DRM support in different browsers. In addition the end-user experience is likely to suffer since the user may either have to use a non-preferred browser or a native application. Also a vision to view prime content on any browser may not become mainstream except those integrating multiple DRM systems at a costly factor.
Application based content viewing will continue to be the preferred choice.
In order to avoid the above issue it is suggested that it is further discussed and a clear industry direction is set.

References

[1] Saying Goodbye to Our Old Friend NPAPI, http://blog.chromium.org/2013/09/saying-goodbye-to-our-old-friend-npapi.html

[2] Get ready for plug-in free browsing, http://msdn.microsoft.com/en-us/library/ie/hh968248(v=vs.85).aspx

[3] Encrypted Media Extensions, https://dvcs.w3.org/hg/html-media/raw-file/default/encrypted-media/encrypted-media.html

[4] Native Client, https://developers.google.com/native-client/dev/