Second W3C Web and TV Workshop

Statement of interest

Jeroen Wijering, LongTail Video

Participant's interest

LongTail Video is the company behind the JW Player, the world's most popular open source embeddable video player. Roughly 10% of all online video streams run through a JW Player.

Additionally, we offer an online video platform for transcoding and delivery to multiple devices. About 2500 clients leverage this platform.

With about 5,000,000 page views, 5,000 forum posts and hundreds of code commits to our site every month, our community of mostly SMB publishers and web developers truly represents the "long tail" of online video.

Historically, the JW Player has been available for the Adobe Flash and Microsoft Silverlight platforms. Over the past few months, the player has taken large strides. We have added a fully featured HTML5 player, abstracted out the actual video playback (Flash/HTML5/Silverlight) from the user interface (CSS/HTML), and built a new API (JavaScript). We have done this because:

a) We are strong believers in the HTML5 <video> tag as an emerging standard for IP-based video delivery.

b) As both mobile devices and connected TV's become relevant to video publishers, we want to offer a single tool for delivering their videos to all relevant platforms and devices.

On the video platform side, we have taken similar steps, introducing WebM transcoding; iOS and Android SDKs; and integrations with Boxee, iTunes and Google TV.

We are interested in participating with this workshop to push forward the development of web video standards, particularly with regards to HTTP adaptive streaming. We believe these web standards should be the framework for delivering video to the big screen.

For our (1M+) customers, continued standardization is critical for making the publishing of their content to connected TV's affordable. At the same time, the success of a connected TV platform may very well depend upon the availability of the "long tail" content these publishers provide.

View points

Standardization of HTTP Adaptive Streaming

Existing HTTP adaptive streaming solutions from Microsoft, Adobe and Apple have proven it to be an extremely suitable technology for IP-based delivery of video to various devices. Standardization is now needed, so HTTP adaptive streaming can be implemented across as many devices and platforms as possible.

MPEG DASH may very well become this standard. In its current incarnation, it is still biased towards “Big Media”, containing a plethora of features not needed by smaller video publishers. A well-defined baseline format will be an excellent solution. At the same time, MPEG DASH should actively support the WebM format (and vice versa) so the current H264/VP8 codec standoff does not also result in multiple adaptive streaming standards.

Uptake of HTTP adaptive streaming much depends upon the requirements and availability of server side tools. Adaptive streaming should be possible through either today’s web servers (HTTP range requests?) or through small plugins to existing web servers (for rendering manifests and returning video fragments).

Advances in HTML5 <video>

The HTML5 <video> tag is, in its current state, mostly useful for displaying short clips. Several enhancements are required to make it suitable for a wider range of applications, and large-scale usage on connected TV’s.

A mechanism for (temporarily) blocking scripting access to a certain <video> tag is important for ad-supported content, to prevent any pre, mid, or post roll advertisement from being skipped due to a few lines of JavaScript.

A baseline video encryption mechanism (e.g. at the video frame / audio sample level) and decryption key transmit protocol would be appreciated by publishers (PIFF?). Our customers do not require extensive, proprietary, and “unbreakable” DRM applications. Most solely want to avoid right-click-save-as scenarios and are not interested in running rights management software.

For both the streaming of long-form content, live events, and streaming to connected TV and mobile devices; adaptive streaming support in (or compatibility with) HTML5 <video> would be an instant solution. There are various ways to getting there:

a) Exposing QOS metrics and a Stream API to build adaptive streaming into JavaScript frameworks and “feed” fragments into a <video> element.

b) Allowing an adaptive streaming manifest file to be the @src of a video tag. An API for retrieving QOS metrics and level switches and forcing quality levels should be provided.

c) Offering <track> and <level> elements to build an adaptive streaming manifest in HTML. This can be married to the existing work on <track> for metadata tracks. The same API as mentioned under b) should be added.

There are pros and cons to each method. Backing (by word or code) or a prototype from at least one browser vendor is probably needed to move forward here.