

LILY BOND: Welcome, everyone, and thank you for joining this webinar entitled HTML5 Video Accessibility - Updates, Features, and Guidelines. It's being held in collaboration with the Online Learning Consortium.

I'm joined today by John Foliot, who's a video accessibility expert and contributor to the W3C's new media accessibility user requirements. He's also a W3C co-facilitator on the HTML5 Accessibility Task Force, as well as the principal accessibility strategist at Deque. And with that, I'm going to hand it off to John who has a great presentation prepared for you today.

JOHN FOLIOT: All righty. How's that, everybody? Can everybody see that?

LILY BOND: Yes, looks good, John.

JOHN FOLIOT: So good afternoon and welcome. Today, we're going to talk about HTML5 video accessibility and about some of the work that happened at the W3C when we started looking at how to ensure that HTML5 video could, in fact, be accessible.

As sort of a precursor to all this, a requirement that the W3C's web content accessibility requirements is based on four basic principles, that the content is perceivable, operable, understandable, and robust. And for those of you that have done any kind of work in the accessibility space, I think this is a well-known mantra.

One of the things that sort of introduces some of the challenges, that multimedia is, in fact, also multi-modal. It includes sight, sound, and interactivity very often. And so it puts additional requirements or additional strains on meeting the WCAG requirements of perceivable, operable, understandable, and robust.

And so a bunch of us got together quite some time ago, and we started working on a Media Accessibility Checklist. The URL is there on the screen. The status of this document is that we're hoping to get it published as an official W3C NOTE later this year. But the reality is that it's a very stable document now, and you can go and reference it at the URL at any one time.

What we did is we started to look at accessibility, or least media requirements, by the types of disabilities that were out there. And so we looked at issues from the perspective of people that were blind or have low vision. Low vision also includes people that might have color blindness and whatnot.

We looked at it from the perception of deafness and hard of hearing. And what does it mean for someone who's deaf-blind to consume multimedia content? We also thought about things around physical impairments and cognitive and neurological disabilities.

And so through all of these different lenses, we started looking at the different types of requirements that were out there, and we compiled the master list that really sort of breaks down into two basic categories. There's the alternative content technologies, right, sort of the stuff that we're going to produce that is the companion pieces to the video that's being produced.

But we also looked at it from a system requirements perspective. What does the media player require in a perfect world? And some of the things we're going to look at today, we've outlined the requirement through that lens of in a perfect world-- this is what we'd like to see. Some of the things are a little difficult to deliver on today.

But one of the points of this document was that we also wanted it to be something of a roadmap. So that even if we can't achieve everything today, we've at least outlined what the user requirement is so that developers, as they continue to move forward, as the technology continues to mature and grow, they have a clear understanding of what it is that we need to do so that we don't have people going off reinventing a wheel that actually has four corners to it, because that doesn't help anybody.

And so as you can see here, there's a number of different things that we've sort of sussed out and went through, and I'm going to focus a little bit more closely on these requirements this afternoon. So on the first slide is the alternative content technologies, the sort of supplemental material that needs to be produced, the companion pieces that go along with your video.

The big one, the easy one, the one that I think everybody understands is the requirement for captions. And I have a quote here that I took from the captioning organization DCMP from "Captioning Key," and I think it's a really good quote because it really sort of defines exactly what we're looking for.

"Captioning is a process of converting the audio content of a television broadcast, webcast, film, video, live event, or other production into text and displaying the text on screen or monitor. Captions not only display words as the textual equivalent of a spoken dialogue or narration, but they also include speaker identification, sound effects, and when required, music

description."

That's a very specific and very precise definition. And I was talking with Lily and Tole before the podcast. We're not going to get into the legal requirements, but I would suggest to you that, from a legal perspective, that's a really good definition to use because it's well-socialized, and I think it really accurately describes what they mean when they refer to captions.

It's important that captions meet a couple of different requirements. One of the ones that I find really interesting and I get asked a lot about is about the synchronization and the timing. And it's important that the captions appear at approximately the same time as the audio on screen. There tends to be sometimes a bit of a confusion that it has to be exactly in sync to the millisecond to the audio that's being spoken in the audio track.

And in actual fact, that's not true because there are times when you might have a short audio burst or a very rapid-fire dialogue, where if you actually had the text appearing on screen at the same rate that the speakers were speaking at, there would be too much information on screen for the non-hearing user to be able to process in sufficient time.

So we want it to be approximately the same, and we want it to be really close. But there is an understanding and acceptance that it can spill over a little bit. Ideally, it can stay on screen longer than the actual utterance. But again, given the amount of content that's out there or that's in your particular video production, if it needs to actually come up on screen half a second or a second before it's spoken, because that's all the timing that you will have, that's deemed acceptable.

What's really important is that we give an equivalent and equal sort of delivery of the content that's in the audio, including the speaker identification and sound effects. And so this equivalent and equal really is the trump card, and it'll trump timing.

The other thing is we want to make sure that these captions are readily available to those who need or want them as required. And for those that may not be aware of the definitions or the different definitions, we have both open and closed captions.

Closed captions are the type of caption that you frequently see, where there's a button in the video player, a little CC button or equivalent, that you can turn on and off. And so you can turn the captions on or off on a user-demand requirement.

Open captions, on the other hand, are usually captions that are burned into the video or are constantly there. They're always present and cannot be turned off. In terms of the requirements for the WCAG requirements, either is absolutely acceptable.

One last note. People will use captions and subtitles sort of as generic terms interchangeable, but they really are not the same. And I prefer to be very specific and precise, and I always refer to them as captions.

Subtitles usually are used when we're dealing with foreign language content. So if a person's speaking German or Japanese or something like that, then we'll have the English equivalent on the screen. Subtitles are generally there for all users, including hearing users and non-hearing users. Whereas captions, while they're useful to hearing users, are more closely targeted towards non-hearing users.

In terms of the requirements or the WCAG Success Criteria, captions are mentioned in a couple of different places, both in WCAG, which is the "Web Content Accessibility Guidelines," but also finds its way into the "User Agent Accessibility Guidelines," which is another document produced by the W3C. And as you can see, there's a number of A and AA requirements within WCAG that reference or require captions.

One of the things that's interesting is Success Criteria 3.1.2, which is the Language of Parts. And it's important that when you're hosting your captions in your media player, that if you have multiple languages, they're identified as such. So just a data point to be aware of there.

There's a number of different type of captions or subtitle formats, and all of them must be able to deliver the following requirements. They need to be able to render text in a time-synchronized manner using the media resource as the time-base master.

There was a lot of discussion early on in HTML5 as to which resource actually owned the master timeline. And believe it or not, there was a fair amount of debate, but they did, in fact, resolve that the primary video, the master video, is also the time-base master.

We're going to talk about that a little bit later on because there was some ideas and some thoughts around what I kind of loosely refer to as time-shifting, which is kind of interesting. But right now, the requirement is that all captions are based on the master video.

Captions should be available in a text encoding, ideally UTF-8. I have seen, especially for some foreign language captions, where in older systems, they didn't support UTF-8 or

alternate or foreign character sets. And they're actually burning images into a caption file which, of course, was great for the non-hearing user.

But for the deaf-blind user who might be accessing this information using an assistive technology, they were just getting a series of pictures. And so they were basically shut out of it as well. So it's really important that our caption file is actually created in a text format.

One of the requirements that we've identified-- we're starting to see the ability to do this but it's still very nascent-- is the ability to support positioning on all parts of the screen, either inside of the media viewport, but also possibly in a predetermined space next to the media viewport.

Right now, what we're seeing is we're seeing instances where you'll have a side-by-side, and you'll have the video play, say for example, on a left-hand pane. And then a right-hand pane, we'll see either captions or dynamic transcripts, which we'll talk a little bit, where the dynamic transcript is sort of substituting for the caption file.

But we had also envisioned the ability to move the traditional on-screen caption binding box, which traditionally is in the lower third of the screen or lower quarter of the on-screen display, the ability to move that or to reposition it.

And we'll talk about the technology a little bit. But using technologies such as HTML5 and CSS, in theory, there's no reason why we can't do that. I've not seen of any actual examples of that yet. But again, the requirements document left room the idea that some of these things are possible. We just haven't seen it yet.

We also suggest-- and this is above and beyond WCAG, by the way-- that you be allowed, permitted, a range of font faces and font sizes, and that you could also be able to change the colors to support a full range of opacities. Right now, most captioning we see is what we consider to be high contrast, either black text on a white background or white text on a black background.

For some people with different types of cognitive disabilities, and specifically dyslexia, high-contrast lettering like that actually is one of one of the things that makes letters jump around on screen, a very common phenomena for people with dyslexia.

And experiments and tests have shown that when you actually reduce the color contrast a little bit, say for example, a dark brown on a beige background, still within sort of contrast

guidelines but a lesser contrast than a real stark black and white, that it actually has a very calming effect on the lettering as well for people with dyslexia.

So something to consider and something that, again, in theory, we should be able to do it with CSS, and in a perfect world, something that actually the end user could adjust on demand. That a user could go in and say, you know, I prefer to have a reduced contrast for my captions on-screen. And that could be a user setting either in the video player or in the browser directly.

We also identified requirements around supporting internationalization and visual display properties. For example, traditional Chinese and Japanese that is rendered top to bottom, or Hebrew, which is right to left. All standard requirements, but they were identified in that master document as well.

There is a number of different formats for caption files out there. Probably the best known today, or at least the one that seems to be most favored, is a file format known as WebVTT. Other formats out there include Timed Text Markup Language, which is a W3C standard, as well as SMPTE timed text, which is actually a superset of TTML.

TTML is XML-based. And the X in XML, of course, stands for extensible. And so the Society of Motion Picture and Television Engineers actually took that timed format and they created a superset where they added some additional authoring requirements, mostly around the area of stored metadata and whatnot.

Either of these formats are deemed acceptable. However, browser support is sporadic. You're going to find that the best browser support is with WebVTT, although Internet Explorer and Microsoft in general have also provided support for TTML and SMPTE timed text. When you get out in the world of captions, you'll find that there are numerous other formats, including Scenarist Closed Caption, or SCC, which is a binary file format that used to be the primary format for mobile platforms, as well as older formats like SAMI or SMIL, RealText, or SubRip, or SRT.

The really nice thing with all of these formats, with the exception of Scenarist, is that they're text-based files. The basics are essentially the same. It's more a syntax, authoring syntax, than anything else. And so converting from one file format to another is trivial at best. There's a number of low cost and free tools out there that can do that. And a lot of caption-providing companies will also deliver multiple formats to you on request.

So there are occasional needs for some of these other formats. But for the most part, if you're doing web-based delivery, my primary recommendation would be to consider using WebVTT and/or TTML or SMPTE timed text.

The second requirement for WCAG, and this one's serious because it is a WCAG requirement as well, is that you need to provide content description. WCAG refers to it to audio description. And if we're going to be extremely pedantic about it, that is a requirement for an audio track that basically is describing what's happening on-screen.

It includes actions as well as important visual things, costuming perhaps, or gestures, or key things on the set, as well as any changes. Traditionally, it's that little additional voice. I liken it to the little man that sits on your shoulder and whispers in your ear what's going on when you can't actually see it.

Up until now, creating audio descriptions has been considered extremely difficult. It requires an additional script. It requires additional voice talent. And it requires some very masterful editing to actually get those descriptions into the blocks of audio that's coming from the on-screen characters. And sometimes it's very, very difficult if you have a lot of rapid fire action back and forth.

You still need to be able to describe what's happening on-screen. And so finding the ability to insert that can be very difficult. We're starting to see now proof of concept and examples, where instead of actually producing it as audio, they're producing it as an additional timed text track that also one sort of in sync with the caption file. And one of the nice things about that is that it can also be rendered on-screen, along with the captions, and/or it could be sent to a synthesized voice tool, text-to-speech synthesizers.

If anybody on the webinar has ever experienced somebody using a screen reader, in particular an advanced user, you'll notice that they tend to turn the speed up quite quickly. I joke it's kind of like that voice at the end of a TV commercial or radio commercial, where they speed it up really fast. Because they have to get all that legal information in at the end.

Screen readers will do that, but even more so, upwards to 200 words a minute, as opposed to the traditional 120 words a minute. And so what that allows them to do is actually consume a lot more information in a short period of time. And so I've seen some proof of concepts out of NCAM, WGBH out of Boston, as well as IBM Japan, where they look to create content that is the description, but delivered as a text-based option instead.

So an absolute requirement there. And again, when we look at the success criteria of WCAG, you'll see that audio description for pre-recorded content is a AA requirement. So it's probably the biggest hole we have today in terms of delivering to full compliance of WCAG AA, to the point that I'm aware of some government entities up in Canada, specifically the Canadian federal government, as well as the provinces of Quebec and Ontario, have actually redlined this particular requirement, Success Criteria 1.2.5, with a provision that they'll address it down the road when the technology is more robust and more mature.

In terms of what to do if you're delivering content right now, the goal is to deliver the descriptive audio or descriptive text. But you may not be able to do so. And so it's a risk management thing that you should discuss with your risk management people. Again, there was also a number of UAAG requirements regarding the rendering of alternative content and retrieval process for those that are interested in looking at those more closely.

Finally, the third piece of content that's really critical, or that is part of the WCAG requirement, is the transcript. And a full transcript supports all kinds of different needs. It will provide-- where the caption is sort of the audio, the spoken dialogue, and the descriptions are the describing of what's happening on-scene, a transcript is sort of a marriage of the two.

And I best liken it to like a screenplay, where it describes both the actions and the audio that's happening. It can be presented either simultaneously with the media material-- and again, I mentioned earlier that we're starting to see sort of dynamic transcripts, where they're serving double duty because they're also using a technologies that kind of highlights the lines that are being spoken. The transcript is providing both descriptive material, as well as highlighting the content that's being spoken.

It doesn't necessarily have to be delivered that way. And in fact, we can also see transcripts that are completely downloaded and consumed offline. You know, I've even seen transcripts delivered in PDF. I don't recommend that. But any offline model would work.

And there are some requirements there as well. One of the things to consider is that transcripts will often be consumed external to the video for some users, specifically deaf-blind users. Multimedia is very difficult for them because they can neither see nor hear. But they can still consume a text transcript using braille output device or something like that.

So it should be made available as an independent download from your media resource. And

so again, even if you're doing that side by side sort of left pane-right pane delivery of a transcript, it's highly recommended that you provide the transcript as a download or as a hyperlink to that text transcript only for those users that may or need that requirement.

Again, the success criteria are there. There's requirements for both headings and audio and video presentation for pre-recorded that has a transcript. I called out headings here because, ideally, we want to see the transcript have some kind of structure as well, especially for a long form video. Short form videos less so, although structured HTML content would make the video transcript a lot easier to read or to navigate.

And we'll talk about that a little bit more. But providing headings and labels within your transcript will facilitate that for screen reader users. So those are your requirements there.

There's a couple of other things that we looked at and we envisioned that are kind of extending and enhancing existing technologies today. And one of them is extended descriptions, which allows us-- it's a combination of content creation, as well as your playback platform, where you could, in fact-- again, as we envisioned it-- you could, in fact, pause the video or video program or the audio program at key moments and allow an extended description to continue to give you a very rich and robust description.

For those of you that are in the educational vertical, that could be very useful, especially when you're doing video lectures where there's a lot of on-screen material that's being described or as part of the lesson, large charts or other complex graphics, where the user not only wants to listen to the dialogue-- you know, Professor Smith giving the lecture. But if there's a slide that has a lot of rich detail, they want the ability to really explore and understand that slide in a very rich detail as well.

That can be achieved using, again, these interactive transcripts, where you could create a hyperlink to an extended description. But if you're developing a custom playback platform from the get go, the ability to pause the audio and video and still allow the extended description, whether it's an audio track or a text-based track, to continue to render would be hugely beneficial for those users.

Similar to extended descriptions, we also looked at enhanced captions that would be also enriched with additional information. Imagine, if you will, a caption file that actually has a hyperlink embedded into it, or that can fire based on timing. Geoff Freed at NCAM, WGBH NCAM out of Boston, actually did a proof of concept about a year and a half, two years ago

using popcorn-js, a freely available JavaScript library, where he did just that.

I don't think Geoff would be offended if I said it's not a thing of beauty. It was very rudimentary in terms of design. But the functionality that he demonstrated was actually really cool, where he did both of these things. The ability to sort of pause the video and provide an on-screen text extended description, as well as providing glossary terms, and definitions for acronyms, and whatnot.

And so I urge you to seek it out. If you can't find it via Google, you can ping me. I'll provide my email address at the end of the webinar. And I'd be happy to send you the link to that. But it was very cool. And it was really exciting for me to see what Jeff had done as a proof of concept. And I look forward to the day for a developer to really sort of grab the bull by the horns and take that on and make it a real first class delivery platform.

Couple of other above and beyond technologies that we called out in the media requirements document is the requirements for clean audio, content navigation by content structure, and sign translation. Clean audio actually is based on a requirement that they have in the UK, where it benefits people that have reduced hearing by reducing or eliminating a lot of extraneous background noise and/or promoting the primary vocal track by boosting the audio, and really sort of reducing a lot of the hubbub that goes on so that the end user can really clearly focus on the important dialogue.

This is very useful, especially in longer form videos, where there might be a lot of-- imagine, for example, an action film, where there's a rock 'em sock 'em fighting kind of scene going on. And there's lots of guns shooting off and cars crashing and all kinds of background noise, and yet the principal actors are also trying to have a dialogue back and forth. That ends up being just a lot of audio noise for some users and it's very difficult for them to be able to follow along. And the ability to actually separate the dialogue track from the background noises makes it a lot easier for them to be able to comprehend it.

And so we can't see-- I mean, they're producing television content like this today in the UK. The BBC has a mandate to produce, I believe, up to 10% of their video programming using clean audio. And so the technology does exist. We haven't seen it yet on web-based delivery. But again, the requirements document wanted to be future looking, and so we called it out as something that exists and is being done in other formats. And there's no reason why we can't be doing it on the web.

Content navigation by content structure. Again, I mentioned in the terms of transcripts, we know that basic HTML headings and paragraphs and whatnot can be used by non-sighted users to navigate around. And again primarily in longer form videos, think of it kind of along the lines of chapters. And you should be able to navigate using either your transcript or your caption file to navigate through the content.

We see this already, again, for those of you that are working in the EDU vertical. We have a very similar mechanism with DAISY publishing right now, with EPUB publishing, where this is digital content that's being delivered to the end user, and yet when DAISY and with EPUB, you can navigate through the-- excuse me-- you can navigate content using these chapter markers. And so again, we called out that would be a really cool thing to be able to do with video.

There is a requirement for sign translation. It's really complicated right now because sign translation, again, it's very useful for non-hearing users, the deaf people. But signed translation is kind of also hamstrung by the fact that there are different types of sign language based on regional differences. There's American Sign Language, there's British sign language, there's Australian Sign Language, and those are just English sign languages.

And there's also an additional requirement for the production quality or production costs related to creating sign language translation. Ideally, we'd like to be able to see-- and the HTML5 has provided a means where you can, in fact, do picture in picture delivery of video. That becomes a little bit complicated on the mobile platform, where mobile devices have a really hard time dealing with multiple streams of content at one time. So again, we've noted them as being useful requirements, but it's the type of stuff that's not quite ready for prime time at this time.

So to summarize the authoring requirements, to be fully conformant to WCAG 2 AA, you require at a minimum captions, video descriptions, and transcripts.

The second half of the document that we looked at was really on system requirements. And so when it comes to system requirements, one thing that you have to understand is, what is an HTML5 media player? An HTML5 media player is in fact the browser. The browser is the video player and the video player is the browser.

One of the key things behind HTML5 was this desire to reduce the reliance or dependence on

plug-in architecture. And so one of the biggest plug-ins that a lot of the browser manufacturers were dealing with was the Flash plug-in. And depending on the platform and whatnot, it was problematic. It had a high overhead, there was security faces that were being exposed at the video that the browser vendors were not happy about. And there were a number of other issues around using embedded video players, whether it's Flash or Silverlight.

And so one of the first things they did is they actually created or added the code to the browser base so that it could render media file formats. There was a little bit of a discussion over which media file format, whether it would be MP4 or WebM. And in the early days MP4, there was concern around patent encumbrances. I think that's, for the most part, been addressed, and today MP4 can be rendered in all browsers. But the browser itself actually does the rendering. We don't need an embedded or a plugged in video player.

And so today an HTML5 media player is the browser. And really what HTML5 anticipated was that instead of creating a full-fledged media player, that instead the authors would simply be creating scripted and customized controls so that you could embed the video into a web page and have controls that could also meet design and branding requirements of the content owner.

So most of the HTML5 video players that we see today are actually doing just that. They're creating code that's embedded into the player. They'll use a fallback, and in the code sample that we have here I say fallback for legacy browsers. Many of the HTML5 video players that are in the market today, they'll fall back to Flash players for those devices that don't support HTML5, which today is increasingly a smaller and smaller pool.

So when we were looking at the requirements of the W3C, we looked at the ability to access interactive controls and menus. And so this is really important, right? Not everybody is going to be using a mouse. And in fact, many users today still are required to use keyboard and/or other alternate input devices. And so we noted that because HTML5 basically was anticipating the creation of these custom controls, that all the controls and menus must be available to all users, no matter which means they're using for input with their device. They need to be device independent so that you can use a keyboard or pointing device or even speech input today.

We also noted that these controls need to be discoverable, and the activation and deactivation needs to be, basically, in the hands of the end user. So it needs to be device agnostic content delivery. And the APIs and user agent controls should adhere to the UAAG guidance. So

earlier on I was referencing the User Agent Accessibility Guidelines. There's a wealth of information there about the requirements of when you're creating a user agent. And here, because the author is creating controls for a media player, which although it is the browser, it's kind of envisioned as embedded into a web page, because you're creating controls the UAAG requirements come to play as well.

So you need to be able to also find the alternative content using these ideas of agnostic tools. So most often we'll see a button of some sort in the controls that offers a drop down menu or something like that that would allow for multiple formats of subtitles or for closed captions or what have you. And so again, all these things need to be user agent agnostic or input device agnostic.

And so we have a number of requirements here. In cases where the alternative content has different dimensions than the original content, the user should have the ability to specify how the layout or reflow of the documents should be handled. So again, going back to that example that I had earlier of interactive transcripts, where you've got the video on the left pane and you've got your transcript on the right pane, the requirement for the end user is that for low-vision users they should also be able to enlarge the text that's in that right-hand pane.

And so one thing that we don't want to see is if the user uses zoom or in some other way enlarges the text that's in that right-hand pane, it needs to reflow properly so that we don't have horizontal scrolling. Because at that point you're going to introduce a huge cognitive load. I mean, it's hard enough to be watching the text on one side of the screen and the actual video on the other, but then to also have to start horizontally scrolling so you can follow along becomes something of a nightmare. So that reflow is critical.

Likewise on mobile devices, tablets and cell phones and whatnot. They also need to be able to browse alternatives and switch between them. So if, in fact, you have a simplified-- or let's say you have two audio streams. There's the standard audio stream and you're also providing a clean audio stream-- why not-- then the ability to choose between the two of those should be made available. Seems pretty obvious, but something that you might want to consider thinking about. Again, we don't have systems that do that today, but there's no reason why we couldn't.

And so all of these requirements here regarding synchronized alternatives and non-synchronized alternatives have been called out. These are the types of things that when you're developing your player you should be considering. So I mentioned your non-

synchronized alternatives can be rendered as replacements. That's your transcript that should be available to be downloaded so that you can actually pull that up instead.

The system requirements for granularity level control for structured navigation. So again, I've alluded to this a little bit, a real time control mechanism so that you can actually navigate content based on these alternative documents, your caption file, your transcript file. Having that structured and having the ability to go through your transcript file and find a chapter heading or a particular sentence, click on that, and actually fast-forward or skip the video to that place is a hugely beneficial thing. It would benefit people with cognitive disabilities. Heck, it would be just a great study aid.

And I'm aware that some companies are providing that service today. I believe our hosts today have offered that service for some time now. And so providing these kinds of things in your content will benefit. They're not absolute requirements, but they'll benefit end users, and so you'll find that when it comes to accessibility, I don't reach for the minimum bar, I reach the stars.

We talked about time-scale modification and this idea of time shifting. And so while we can't really do this with video players today, and I don't know if we'll be able to in the immediate future, a standard control API should be able to support the ability to speed up or slow down the content presentation without altering the audio pitch. So this is a requirement. This has been a longstanding requirement and actually is being delivered in audio books today, where you can actually slow down or speed up the delivery rate, and yet you don't get that deepening of the voice or Mickey Mouse voice as you speed it up. That the pitch can be maintained at a sort of normal human listenable rate, within reason, even though you've sped up or slowed down the audio.

And so again, because we've got this in devices today, we call it out as being a requirement. This is a "nice to have." But if you're designing a system for the future, we figured we'd call that out as well.

One of the most important things is that for user agents that are supporting accessibility APIs, any media control needs to be connected to that API. So because we're using-- traditionally we're using the web browser as the delivery device, the accessibility APIs are there for all of the major web browsers. And the easiest way to connecting to the accessibility API is by using ARIA.

So for all of the buttons and controls that you're creating in your custom controls, ensuring that you're creating them with ARIA will satisfy this requirement. Not much more to say to that. It's pretty simple. It's pretty straightforward. And for those people that are used to digging, you know, jumping into the code up to their elbows, I think this is a pretty straightforward requirement.

We also looked at the requirement around the use of the viewport. And what we discovered or what we sort of concluded was that the video viewport provides the bounding box that contains all of the information. But that there was no reason why it had to be that way. I mean, that was sort of the traditional way. When you had an embedded player, everything was inside of the viewport.

But along with this ability to move the captions around on screen, we also thought about the ability to sort move some of that alternative content outside of the viewport but still make it available. This is going to remain problematic on mobile devices for now because the mobile devices are the one tool where, for the most part, they're handing their videos off to the media players that are embedded on board in the system.

They're still not using the web browser. But for desktop delivery and for other types of delivery-- large-screen, sort of the lean-back delivery-- the ability to move this content outside of the viewport into a docked section or something like that is something that we recognize as being hugely useful.

So basically, the summary of the system requirements is that all the controls must be accessible via the platform APIs. And so-- rather via the platform accessibility APIs. And so again, the use of ARIA will satisfy that requirement. Alternative content must be discoverable and able to be modified by the end user.

So the discoverability-- usually again from your controls where you would have a button, or a drop-down box, or some other means of actually being able to discover the content. In the case of a transcript, it could be a simple link on page. Although we're looking to see a more elegant sort of design-friendly solution there-- still working on that particular piece of the W3C. And ideally the content navigation and content display should allow for personalization.

And again, we talked about using WebVTT, which again is-- the markup of WebVTT is very similar to HTML, and it was designed so that you could use CSS. And so a really robust web-based media player would allow, perhaps, if not a fine granular control where the end user can

go in and specify foreground and background color, certainly some kind of widget that would allow different types of color contrast or color combinations to suit different needs of different users. Again, and I mentioned that sort of brown on beige as being something useful for people with dyslexia.

One of things that I've talked about-- oops. What happened there? Oh, I got-- OK. One of the things that-- I don't want to spend too much time on today because we've only got about 10 minutes left, but one of things that's really kind of interesting and important is the infrastructure and production requirements. And so really there are three things that you need to be aware of when you're actually creating accessible video and hosting accessible video in your particular shop.

The first is the production of the accessible video and all the supporting pieces. The second requirement is the ability to stream your accessible videos and all of those accessible components. And a final requirement, something to be aware of, is the ability to manage your accessible video library. So I'll just zip through these slides very quickly. And again, this deck will be available after the webinar, and I've also got some notes if you want.

And so the first thing, the most important thing, is getting your captions and description files created. Many entities today are looking to outsource this as being the most efficient and cost effective as well as time effective way of doing it. Of course, captions-- 3Play Media, our hosts today, do an excellent job, and I've used them myself personally many times. And so I can certainly vouch for them.

Other caption providers that are doing a lot of work in the marketplace today include companies like Automatic Sync, or CaptionMax, or the National Captioning Institute. All of them seem to be focusing on different verticals. And I've not heard anything bad about any of them. So use 3Play Media. There you go. I put in a plug for them.

In terms of described video, a little harder to do. There's not as many people doing that. Bridge Multimedia is one such company, as well as Dicapta. CaptionMax will also provide described video. There is, again, as I said, a real art to creating described files, description files, whether it's producing of the audio file or whether it's just producing it as a text format.

The script, the actual text that's being delivered, there's an art and a skill to succinctly describe what's happening on screen in as most economical as few words as possible. And so it's not

for the faint of heart. I highly recommend that you look at professionals to do that. However, if you're on a shoestring budget or you only have one or two videos a year, there are a number of software tools out there that allow you to do that yourself.

In terms of captioning, it's been my experience over the years that if you're going to do the captioning yourself using one of these tools that it takes roughly six minutes for every one minute of video-- tends to be sort of the norm for non-expert caption producers. Obviously the professional companies could do it quicker than that. But for the average web guy that just does it once or twice a year, factor something six to one.

Again, there's a number of tools on the marketplace here. I've listed three of them. These are free or low cost. And basically, they allow you to play the video with a pause button. You play 10, 15 seconds of video. You hit the pause button. You type what you heard. And then it also does the timestamping. Very rudimentary, but effective. It gets it done.

In terms of video description software, there's a couple of online tools. YouDescribe is a web-based tool that's specifically, right now, used with YouTube videos. Cool little service. I'd like to see it extend beyond just YouTube videos. But it's a free web-based tool, and so they're limited by production capacity of the people that created the tool basically. And I think some injection of money would probably allow them to extend it beyond YouTube.

There's CapScribe, which is a Mac-based tool. There's MAGpie again, which can be used for both descriptions and captions. And there's Livedescribe, which was a prototype of the Center for Learning Technology. So all of these tools can be used for producing your description files.

When it comes to producing accessible videos, in HTML5 we have a situation here-- and really folks, if I'm going to give you something to take home today as a little nugget that you may or may not be aware of is the difference between in-band and out-of-band delivery of your caption files. So the traditional HTML5 delivered to the desktop-- HTML5 video-- the code, the HTML code, basically looks like what you see on screen right now where I've opened up my video element, I declare the movie, why there is a WebM file format or an MP4.

Using the track attribute, I've provided the English captions-- whoops, there's a bit of an error there. It shouldn't have been-- the second one, the french.vtt shouldn't have been captions. It should have actually been the kind equals subtitle. And then the language is French. And so I've met the requirement of I can identify the language, and I provided multiple alternatives. And then the fall back.

The problem here is that on the mobile platforms, they have a really hard time keeping all of these multiple files in sync simply because of the way the web works. And TCP/IP, where it's packets that they're being delivered. And especially on the mobile where you're also dealing with mobile networks, and distance to broadcast tower, and whatnot. And so what you need to do is you actually need to consider putting your caption files and your description files, if they're text-based, or even if they're audio-based, but all of your supporting files need to be contained in-band.

I have a screen capture here of a free software tool called Handbrake is one way of doing it, but you'll see on the screen capture, and I don't know if you can see right here, where there's the ability to actually include subtitles. Right now, Handbrake is only supporting the SRT file format, which I think is silly, but I've not tried importing WebVTT files. It may support them as well.

But essentially what happens is it brings all of these files into your MP4. So MP4 and WebM are what's known as media wrapper formats. And so your MP4 file will have your video that's been encoded in H.264. It'll probably have your audio in AAC audio format. And so all of these different formats and all of these different pieces are all wrapped up and bundled up into the MP4 file format.

In simple terms, I kind of describe it to some people as think of it as a special kind of zip file where everything-- you sort of contain everything-- or cabinet file is another file format-- where everything is sort of wrapped up into one file. And then that file is being delivered to the end user. And this is how mobile can ensure that the caption files and whatnot remain in sync with the master file of the video.

So there's a couple of tools to do that. With the exception of Handbrake, most of them tend to be commercial tools that you need to buy a license for. Most people I'm aware of are using Adobe Premiere Pro to do it. I think most of the companies that are producing caption files today can assist in this as well because it is a known issue, and it's a big issue. But for those of you that were unaware of it prior to today, it's really important in terms of being able to deliver your alternative content with the media that you included as an in-band file for the mobile delivery.

And speaking of mobile delivery, the other thing that you want to really be aware of is if you're looking to host the content yourself, the last thing you want to do is to be uploading your media

file on your standard web server. Content that's uploaded to a standard web server is using the HTTP file protocol, which is intended for text, hypertext transfer protocol, HTTP. And video is not. Video is a binary file format.

And what happens is that if you try to upload or if you try to deliver your video using HTTP, you're going to get a lot of buffering because of the way that HTTP works. It's waiting for all of the packets to arrive before it can render on screen. And so instead, what you want is you want to have a streaming media server. And today, most streaming media servers are also providing something called adaptive streaming.

And basically, adaptive streaming looks at a number of different things using various types of media queries. But it'll look at the screen resolution of the receiving device. So is it a cellphone, or is it a 72 inch monitor in your living room on the wall? It will also look at the delivery, the requests, and where is the request coming from? Is it coming from a mobile network, or is it coming over broadband? And it can then serve up an appropriate video encoding based on those requirements.

So for the large screen-- that 1080p delivery to your large screen TV-- you want a large file that's got a lot of rich video detail so that it can be delivered in high definition. Whereas you don't necessarily want to be delivering high definition movies to your iPhone, because you're just trying to push too many bits down the line that just aren't going to be used. So adaptive streaming will address that.

It will also address that issue of distance to your broadcast tower when you're using mobile delivery platforms, whether it's 4G or 3G, or God forbid, if you're still having to deal with Edge. And so you want to look at an adaptive streaming server or media server there.

Finally, when it comes time to managing your accessible video library, you're probably going to want to use a media CMS, and not all media CMS solutions are equal. I can't really give you any recommendations at this time. But when you're looking for a media CMS, you want to make sure that the CMS will allow for the integrated upload and tracking of not only your video file or your multimedia file, but also all the support files that you've created for your accessible media, whether it's a caption file or video or audio description, your transcripts, and any of the other support pieces that you've provided. Ideally, it's integrated with your adaptive streaming server so that if you have multiple copies of the same video, your high def and your low definition video, that they all go into the CMS, and they can all be managed from one common

user interface.

And ideally, the media CMS and/or the CMS in concert with your media server can convert to multiple codecs, whether it's mp4 or WebM as required. That's not a huge issue but something to consider. And so, again, I didn't want to spend a lot of time on that. But those are some of the other pieces that you need to consider as you're pulling together your entire media solution.

And so with that, I'm going to say thanks very much. Ran a couple minutes over 45 minutes, but actually, with the full hour, I can stick around a little bit, however, and take any questions. And so I'll stop now.

LILY BOND:

Great. Thank you, John. That was a fabulous presentation. As he said, we are right at 3:00 right now, but we are going to stick around for 10 to 15 more minutes for questions. Those will be included in the recording, so don't worry about it if you have to go. You'll still have access to them shortly.

Feel free to continue to type in your questions into the control panel. As people type in their last questions, I just wanted to mention that we have several upcoming webinars. One that we are particularly excited about will be presented by Arlene Mayerson, who led the legal team that secured a historic settlement in the case of NAD versus Netflix, which ensures 100% closed captioning for Netflix. And Arlene will discuss how she and the NAD brought Netflix under the ADA, as well as how the ruling impacts the legal landscape of web accessibility and closed captioning for education and other industries. That webinar will be on September 16, and you can register for that and other webinars on our website at 3playmedia.com/webinars.

So with that, let's get into some questions. John, someone is asking, has the HTML5 track element gotten widespread browser adoption?

JOHN FOLIOT:

The short answer to that is yes. As far as I know, all of the modern browsers today are supporting the track element.

LILY BOND:

Great. So someone else is asking whether you have any suggestions for properly marking up timestamp lengths in transcripts. They're trying to create-- they're able to create links in HTML5 transcripts that will actually jump the user to that particular spot in the podcast by an HTML5 player, but the time tag doesn't feel appropriate, and there doesn't appear to be any ARIA support for timestamps.

JOHN FOLIOT: So your audio was breaking up a little bit there, and I'm not sure if it's on my end or your end. Right now, the solutions that I'm seeing are really based on using the anchor tag. There's no reason, however-- and again, in the media requirements document that we created, we absolutely envisioned the ability to do that.

Again, for those of you who are interested, yeah. You've got my email address. It's on screen right there. If you send me an email, I can certainly point you to the link of what Geoff Freed did at WGBH NCAM, because I think what he was able to achieve with Popcorn.js really does kind of meet the requirement of what you're looking at, although it might be in reverse. But it certainly allows for the linking of content and having hyperlinks inside of your transcript that also allows for the pausing of the video and whatnot.

But to be able to use a transcript to navigate a video, again, I know 3Play offered that some time ago. I believe it's still part of your product offering. But that's really the best that you can achieve right now.

LILY BOND: Great. Thank you. John, do you have any idea whether search engines are indexing captions referenced by the track element?

JOHN FOLIOT: As a matter of fact, that is my understanding is that Google in particular is looking at that and are starting to index that. I think a lot of their focus also has been on caption files that are being uploaded to YouTube. And I didn't really talk about YouTube too much on this webinar, but all of the requirements-- WCAG requirements for accessibility that would be applied to self-hosted and self-published videos-- also apply to videos that you might be hosting or producing and delivering via YouTube.

With YouTube, they've got their automatic speech to text recognition, which is not very good, but it's better than nothing. But they allow for the upload of clean or accurate caption files. And so when you upload a caption file, my understanding is, yes, absolutely, Google is indexing that. And I believe you can now even search via captions. If you search for a key term in a caption file, it'll turn back YouTube videos.

LILY BOND: Great. Someone is asking, of the HTML5 video players you're aware of, is there one that stands out for learning platforms like Moodle or Blackboard as a good fit?

JOHN FOLIOT: To be honest with you, no, because again, as I mentioned in the webinar, the HTML5 video player is basically the browser. So the functionality that you're seeing is not in the actual video

player itself but rather in the controls that are associated to the video player. So when I pulled up that code example early on when we were talking about in-band versus out-of-band, you'll notice that it said angle bracket video, and then there was a space, and then I wrote controls. And so in that particular case, when you say controls, basically what you're saying in your source code is that you're providing-- or you're actually requiring the browser to use native controls. So all of the browsers will have native controls for the video element.

I got that wrong. When you say controls, it means you have scripted controls. And so, really, the way they envisioned it was that you would have scripted controls to do the start, stop, pause and all of the other functionality that we've talked about, and that they could also be styled to meet sort of the visual output. So really, it really has more to do with the themes and skins that you might be using than an actual media player.

I know some of the commercial media player folks out there have done a really good job on really making the scripted controls really cross platform and cross browser active. They work in all of the basic browsers and platforms. And they're all using some kind of fallback, usually Flash. But it reaches a point where, if you'll pardon the expression, it's kind of a Coke and Pepsi world, right? I mean, they're all basically doing the same thing.

LILY BOND: Thanks, John. I think we're going to do one more question here. Someone is asking if you could talk high level about the steps to switching one's content to an HTML5 player.

JOHN FOLIOT: As opposed to?

LILY BOND: To whatever player they're using now.

JOHN FOLIOT: Well, at a high level, you want to make sure that your video is in a supported file format. So your MP4, or rather your encoding, your H.264 encoding, is probably the first thing you want to make sure happens. There are other encodings out there, but they're not supported natively in the browser.

Google was working on Web 8 and WebM as the wrapper file format, but I think work on that has stalled. I've not heard any news in almost a year and a half, two years. Because the owners of the proprietary H.264 encoding technology-- basically, they relented and said that they weren't going to force the patent protection or the patent call on that particular encoding.

So you want to ensure your video is encoded in the H.264 and the MP4 wrapper so that it will

be well supported in all the browsers. I would recommend that if you have caption files in another file format and older caption formats-- we often saw SRT, .srt, as a common file format-- I would recommend converting it to WebVTT, simply because it's a more modern markup format. As I mentioned, they're all basically the same. That's just a bit of a syntax question. The nice thing with moving to WebVTT, however, is that you can now start to style it using CSS.

And so those would be the steps that I would do is I would look at my existing media library and make sure that it's all encoded and that the transcript files are in WebVTT. And then move into a good content management system and get good scripted controls in my media player.

LILY BOND: Great. Well, John, thank you so much for a great Q&A and a fantastic presentation. Everyone is writing in that they really appreciated it, and it was really valuable to them, so thank you so much.

JOHN FOLIOT: Well, thank you, Lily. If anybody has any further questions, if you want to link to that Popcorn example from the NCAM, WGBH, or if you have any other questions, my email address is there on screen. It's john.foliot@deque.com. I'm happy to answer questions. I mean, I can't go too in depth, but happy to answer some basic questions and provide whatever pointers I can, so don't be shy. And I want to thank 3Play Media for inviting me to come speak today and for providing these webinars. I think they're very useful, and they're great.

LILY BOND: Well, thank you. It was great having you present. And thank you to everyone who joined today. Just a reminder that we will be sending out an email with a link to view the recording, as well as John's slide deck tomorrow, so keep an eye out for that. And I hope that everyone has a great rest of the day. Thanks.