Encrypted Media Extensions(EME)

These are notes from public-html mailing list that has happened till 1st March 2012.

Various participants and stakeholders in the EME debate

- **Digital Content Providers:**
  These companies provide digital media to users usually as streaming services. Content owners(Hollywood) usually want the providers to place restrictions on the media to reduce illegal copying of media. Often there are legal constraints to the kind of content protection they can and cannot use.
  - Participants in the debate:
    - Glenn Adams works for Cox Communications
    - Bob Lund works for Cable Labs
    - Mark Watson works for Netflix[+]
    - Mark Vickers works for Comcast.
    - Clark Stevens works for CableLabs

- **DRM platform Providers**
  The stakes of DRM platform providers aren’t very direct, the public statement supporting EME talks about benefits to the user and benefit to developers and not directly to the platform providers[1] [2]. According to Ross Anderson and Hal Varian platform providers benefit from lack of interoperability that is inherent to DRM systems.[3] [4]
  - Participants in the debate:
    - Adrian Bateman works for Microsoft[+].
    - Eric Carlson works for Apple.
    - Carr, Wayne works for Intel
    - Leonard Rosenthol works for Adobe
    - David Dowin works for Google[+]

- **FOSS browser developers**

  - Participants in the debate:

[1] Adobe Support for Encrypted Media Extensions https://blogs.adobe.com/standards/tag/eme/

[2] https://blogs.windows.com/msedgedev/2015/10/27/using-encrypted-media-extensions-for-interoperable-protected-media/

[3] Section 5.2, Chapter 22, “Security Engineering”

[4] Economics of DRM, Hal Varian https://www.law.berkeley.edu/files/Varian.pdf

[+] They are the editors of the EME draft

        - Chris Pearce works for Mozilla
        - Henri Sivonen works for Mozilla
        - Robert O'Callahan used to work with Mozilla till March 2016, he was rr (record and replay) developer.
        - David Baron works for Mozilla
        - Boris Zbarsky works for Mozilla.

- **Individuals**
  Various individuals have participated in the EME debate regardless of the stakes of their employers, these individual have debated about various issues including ethics, privacy of users, accessibility of content and EME process and the need for it in the W3C.
    - Participants in the debate
        - Tab Atkins works for Google as a web standard's hacker http://www.xanthir.com
        - Ian Hickson: He works for Google and is a part of CSS working group.
        - John C. Vernaleo worked at Conformal System at the time of EME discussion, a security and open source firm. http://www.netpurgatory.com/web_stuff/media/cv_jcvernaleo.pdf
        - Andreas Kuckartz is a german software developer.
        - Charles Pritchard works for Jumis, which is a web applications and services provider
        - Kornel Lesiński works at Financial Times
        - Benjamin Hawkes-Lewis
        - David Singer works for Multimedia and Software Standards, Apple. He has written on identity management and privacy principles in W3C. https://www.w3.org/2011/track-privacy/papers/Apple.pdf https://www.w3.org/2011/identity-ws/papers/idbrowser2011_submission_51.pdf
        - Smylers is a programmer from Leeds, UK
        - John Foliot is the principal accessibility strategist, for the last 15 year he has worked to make make websites more accessible, prior to that he worked for the record industry for 15 year.

Timeline

- 21st Feb 2012 Adrian Bateman announced on public-html mailing list that he, Mark Watson, and David Dorwin have been collaborating on an API to enable encrypted media in HTML.
- Heated discussion followed the announcement

- April 2012: Paul Cotton proposed to a set up an media task force to complete EME[5] [6], which saw significant opposition and also support.
- January 2013: Paul Cotton Call for Consensus (CfC) to publish as a First Public Working Draft (FPWD)[7]
- February 2013: W3C announced EME is within the scope of HTML
- March 2013: EFF publishes its Formal Objection against EME
- May 2013: Working Group decides to publish EME as first public working draft[8]
- At least by September 2013 IE had implemented EME[9]
- May 2014: Mozilla decides to integrate Adobe CDM in Firefox.[10] [11]
- May 2014: FSFE submits an open letter to European Commission

Major themes of the Debate

The editors of the EME draft have stated that the aim of EME is API to play encrypted media through <audio> and <video> tags.

DRM in HTML5

- Tab Atkins argued that it is technically impossible to prevent users from copying the media they consume and DRM is practically useless, it imposes unnecessary costs on legitimate users while doing very little to actually stop copyright infringement. While it talking about other forms media like movies, pictures and books, he says that commercial video can be done without DRM and there are Indie producers on youtube and the comedian Louis CK who make livings without DRM.[12] [13] It is on its on its way out in other media industries and he see why video should be given a special privilege, given the history elsewhere.[14]
- Henri Sivonen argues that failing to describe the specs of the protection system EME doesn't provide benefits of interoperability and level playing field for competition. Only lowering the R&D cost for proprietary DRM systems shouldn't be a good enough reason for W3C to work on EME.[15]

[5] https://lists.w3.org/Archives/Public/public-html/2012Mar/0275.html

[6] https://lists.w3.org/Archives/Public/public-html/2012Apr/0007.html

[7] https://lists.w3.org/Archives/Public/public-html-admin/2013Jan/0102.html

[8] https://lists.w3.org/Archives/Public/public-html-admin/2013May/0030.html

[9] https://blogs.msdn.microsoft.com/ie/2013/09/05/online-professional-quality-video-premium-media-experiences-without-plug-ins-in-internet-explorer-11/

[10] https://blog.mozilla.org/blog/2014/05/14/drm-and-the-challenge-of-serving-users/

[11] https://hacks.mozilla.org/2014/05/reconciling-mozillas-mission-and-w3c-eme/

[12] http://lists.w3.org/Archives/Public/public-html/2012Feb/0317.html

[13] http://lists.w3.org/Archives/Public/public-html/2012Feb/0326.html

[14] https://lists.w3.org/Archives/Public/public-html/2012Feb/0456.html

[15] http://lists.w3.org/Archives/Public/public-html/2012Feb/0333.html

- Ian Hickson's strongly rejected EME on ethical grounds saying[16],
  - *There is no value to the user for you to use HTML rather than Flash if you're still going to require a proprietary plugin. The value of an open standard is that anyone can write an interoperable user agent without needing to coordinate with anyone else, and get full access to all the content on the Web. DRM completely undermines this. It compromises the very purpose of having a standard.*

  - *The shame is that people's rights are being taken away by paranoid content producers who don't trust their users, and that there are any software engineers willing to do this for them. And shame is absolutely the right word for this.*

    It is morally wrong to deprive users the ability to time-shift or format-shift content they have paid for.

    It is morally wrong to deprive users the ability to reuse content they have paid for purposes of parody.

    It is morally wrong to deprive users the ability to criticise content they have paid for.

    DRM doesn't stop copyright violations, and is unnecessary for the purposes of selling media (indeed it seems to actually reduce total sales). The music industry figured this out a few years ago. Your industry will figure it out eventually too.

- Mark Watson argued that DRM is not supposed to be perfect, it acts like a speed bumps for copyright infringement. By not standardizing DRM, millions of dollars worth of engineering effort goes waste and the user bears an unnecessary and large cost of fragmentation.[17] [18] He later said "*This list* (public-html) *is not the place to argue the ethics of that. W3C needs to decide whether to work on making that a possibility, or whether HTML5 is simply not going to be a suitable technology for our segment of the industry, which would be a shame.*"[19]
- Glenn Adams asked W3C to remain neutral on the fact that DRM should be used or not.[20] He then argues the users of web include both the content producers and the content consumers, protecting users should mean protecting the interests of both,

[16] https://lists.w3.org/Archives/Public/public-html/2012Feb/0497.html
[17] http://lists.w3.org/Archives/Public/public-html/2012Feb/0324.html
[18] http://lists.w3.org/Archives/Public/public-html/2012Feb/0329.html
[19] https://lists.w3.org/Archives/Public/public-html/2012Feb/0494.html
[20] https://lists.w3.org/Archives/Public/public-html/2012Feb/0369.html

also W3C shouldn't be biased against the users who are willing to pay for protected content.[21]

Support walled gardens

- Glenn while arguing to support existence of performance constraints on SmartTVs he said “*SmartTVs do not try to support browsing the web at large. Rather, they support specific walled garden content that has been specifically tested against the device.”*[22]
- Smylers said that if SmartTV work in walled garden and doesn’t implement HTML5 as whole. W3C shouldn’t be concerned with it.[23] Later he said that W3C should focus on the needs of the web first and then look at the requirements of others without causing significant determinant to the web requirements. [24]
- Glenn replied “*these issues are relevant to this WG because these service providers and their end users are both customers and members of this WG;*”[25]
- Henri said “*Personally, I disagree and think that the World Wide Web Consortium should be focused on the World Wide Web and not put effort into catering to user agents that don't try to support browsing the Web at large.”*[26]
- Leonard disagreed with Henri, giving example of HTML based email client and EPUB-based readers, he said the needs of other standards (and standards bodies) that have based their work on HTML are quite relevant to this working group.[27]
- Glenn said there is no browser which implements the full HTML spec and whether a smart TV employs walled garden HTML in some cases or arbitrary HTML in others does not mean it is not part of the web. Using web technology makes it part of the web.[28]
- Mark Watson said that services like SmartTV are considered a part of the World Wide Web and W3C mission refers to “web of everything” which explicitly mentions TVs.[29]

[21] https://lists.w3.org/Archives/Public/public-html/2012Feb/0375.html
[22] https://lists.w3.org/Archives/Public/public-html/2012Feb/0480.html
[23] https://lists.w3.org/Archives/Public/public-html/2012Feb/0499.html
[24] https://lists.w3.org/Archives/Public/public-html/2012Feb/0507.html
[25] https://lists.w3.org/Archives/Public/public-html/2012Feb/0503.html
[26] https://lists.w3.org/Archives/Public/public-html/2012Feb/0504.html
[27] https://lists.w3.org/Archives/Public/public-html/2012Feb/0505.html
[28] https://lists.w3.org/Archives/Public/public-html/2012Feb/0509.html
[29] https://lists.w3.org/Archives/Public/public-html/2012Feb/0512.html

Comparison with the existing HTML5 specification and alternate implementation

Obfuscation

- Charles Pritchard argued obfuscation should be enough for EME's intended purpose as only programmers can grab the content who is the 99.99% of world's population. He says" *"consumer" content protection is just a silly cat and mouse game of make-work opportunities for corporate attorneys and 501c structures. It's well-established that content protection does not stop any consumer products from being pirated. Obfuscation works just fine for stopping the average consumer. Everything else is just extra work, unfortunately necessary to keep contracts from getting wet*."[30] He said that there is no need of EME as `<video>` is codec agnostic and the media can be streamed using an arbitrary codec which only supported by the media provider.[31]
- Tab Atkins pointed out that the legal difference between encryption and using an arbitrary codec is that the former is covered by DMCA but the later isn't.[32]

- Henri asked why "*content descrambling using a site-supplied JS program running in the general purpose interoperable JS execution environment*" isn't being proposed?[33]
- Mark Watson responded by saying that JS is not a secure enough and it might turn out to be to heavy to carry out decryption.[34]

Encryption using existing standard tag

- According to David Singer encrypted video can be played through the existing <video> tags where the content file says its content-ID and is marked as protected, someone who has the DRM to play the content installed and has brought the keys to play it can watch the video.[35] for concrete example he talked about protected .m4p audio files from iTunes library, which plays just fine on Safari.[36]
- Mark Watson responded by arguing that many browsers have entire media pipeline in the browser's code and hence David's method cannot use the content protection

[30] http://lists.w3.org/Archives/Public/public-html/2012Feb/0325.html
[31] https://lists.w3.org/Archives/Public/public-html/2012Feb/0328.html
[32] https://lists.w3.org/Archives/Public/public-html/2012Feb/0348.html
[33] https://lists.w3.org/Archives/Public/public-html/2012Feb/0414.html
[34] https://lists.w3.org/Archives/Public/public-html/2012Feb/0426.html
[35] https://lists.w3.org/Archives/Public/public-html/2012Feb/0422.html
[36] https://lists.w3.org/Archives/Public/public-html/2012Feb/0433.html

facilities available on the OS, he also said that there isn't a standard way to use the method.[37]

Using HTTPS or JS encryption for EME's purpose

- Henri said if it is okay reveal unscrambled content to the user and the aim is to only hide it from the third parties then https is also sufficient [38].
- Mark Watson responded that the content providers might wish to hide content when it is stored in servers and that http service with CDNs are cheaper than https and is operationally simpler.[39] On which Henri said *"if the CDN is treated as an adversary but the user isn't, there's no need for open-ended vendor-specific CDMs, to address this case. Instead, it would make more sense to standardize one general-purpose HTTP payload decryption layer"*[40]

- Tab Atkins said in case where server is untrusted DRM is an unnecessary baggage and JS encryption between the browser and the video tag should suffice[41]
- According to Glenn Adams JS encryption will be unacceptable from a performance point of view in constrained devices like TV and Set Top Boxes.[42]
- Tab Atkins clarified that decryption in JS doesn't necessarily mean written in JS and encryption/decryption module written in C++ with a JS API will satisfy.[43] Charles also said that he see no reason why stream ciphers written in JS cannot meet the performance requirements.[44]
- Glenn agreed that exposing a block decryption API at JS level might make performance a non issue, but the more important issue is where plaintext will be exposed to the JS and whether content providers will find it acceptable. [45] [46]

Henri's proposal

"

This feature adds a decryption layer to the browser's HTTP stack and an API for initializing decryption keys from a different origin. Also, the Same Origin Policy is extended to block obvious access to decrypted data from JavaScript.

[37] https://lists.w3.org/Archives/Public/public-html/2012Feb/0424.html
[38] https://lists.w3.org/Archives/Public/public-html/2012Feb/0411.html
[39] https://lists.w3.org/Archives/Public/public-html/2012Feb/0438.html
[40] https://lists.w3.org/Archives/Public/public-html/2012Feb/0498.html
[41] https://lists.w3.org/Archives/Public/public-html/2012Feb/0456.html
[42] https://lists.w3.org/Archives/Public/public-html/2012Feb/0460.html
[43] https://lists.w3.org/Archives/Public/public-html/2012Feb/0469.html
[44] https://lists.w3.org/Archives/Public/public-html/2012Feb/0470.html
[45] https://lists.w3.org/Archives/Public/public-html/2012Feb/0471.html
[46] https://lists.w3.org/Archives/Public/public-html/2012Feb/0506.html

The browser maintains a key storage that holds tuple of key, sha1(key), origin of key, list of authorized origins and time to live. There's a JavaScript API navigator.addKey(keyUrl, arrayOfAuthorizedOrigins, timeToLiveSeconds, doneCallback). keyUrl is a URL of the same origin as the caller of the API. The payload retrieved from the URL is key material to be added to the key storage. arrayOfAuthorizedOrigins is an array of origins serialized as strings that are authorized to serve content to be decrypted using the key. (This is a privacy mechanism against other origins probing the key store in case an untrusted CDN has leaked key hashes. More on hashes later.) timeToLiveSeconds is the number of seconds after which the browser purges this keystore entry. doneCallback is a JavaScript function that the browser calls after it has retrieved and processed keyUrl. Upon success, a single argument true is passed. Upon failure, a single argument false is passed. (Note: The key material is not exposed to JS.) The browser generates an id for the key by hashing the key material with SHA-1. Origin of key is set to the origin of the caller of the API which has to be the same as the origin of keyUrl.
When an HTTP response includes the response header Content-Encoding: AES256, the processing happens as follows (if any step fails, treat as like other HTTP errors):

The HTTP implementation gets the value of another response header called Key-SHA1 and base64-decodes it. Then, the browser's key storage is searched for a key whose sha1(key) entry matches this value and whose list of authorized origins containst the origin of the HTTP response and decrypts the response payload using AES256 using the located key as the decryption key. The decrypted payload is exposed to the other parts of the browser as having origin E(origin of key).

Origins of the type E(Origin) have the following properties:

 * A resource of origin E(Origin) can be included as embedded content (<img>, <video>, <audio>) in (and only in) a document whose origin is Origin.

 * For the purpose of JavaScript access to the data of the resource (be it raw bytes or pixel data), E(Origin) is considered to be different-origin with every origin including the origin(s) representing the authority of browser extensions.

 * Browsers disable the "Save As..." context menu for embedded content whose origin is of the form E(Origin)

When the layer above the HTTP layer request the HTTP stack to perform a range request on a Content-Encoding: AES256 resource, the HTTP stack must extend the range such that the range consists of full AES256 blocks in both directions. (The lack of block chaining is deliberate so that seeking is possible.)
“

- Mark Responds:
  If the keys are available to the browser code, then this probably wouldn't work for most of our content, since a browser could be trivially modified to release the keys.

  If the keys are not exposed to the browser code (i.e. if the mechanism above were implemented within something like the CDM we propose and this met our security requirements) then another issue is that you (presumably) rely on standard HTTP authentication to authenticate the user.

  Presently we use our own application protocol within which we have our own authentication mechanisms. We provided details of this in the discussion on the proposed web cryptography working group [1] . This avoids the need to stand up front-end servers for each different key delivery protocol (of which your proposal would be just one).

  This is one reason why we propose to proxy the key delivery messages through the Javascript layer.

  Performing the decryption at the HTTP layer has the following problems:
  - HTTP servers have to support the additional headers, which means CDN changes are necessary. Even trivial CDN changes take time to roll out across the world and CDNs generally want some compensation. Anything which restricts our choice of CDNs increases our costs and reduces user quality of experience (due to reduced diversity). So there's value to us if the solution works with existing HTTP services.
  - existing standard encryption mechanisms (for example common encryption for mp4 files) cannot be used, meaning that existing files would have to be re-encrypted and also that separate copies of the media would be needed for browsers based on this system and devices following the existing standards. Requiring separate copies has both storage costs and cache efficiency costs (which impact quality of experience).

  It's possible the second point could be addressed by making the HTTP stack file-format-aware, so that it could apply the appropriate kind of decryption. But that seems architecturally problematic.

- Henri responds
  > If the keys are available to the browser code, then this probably wouldn't
  > work for most of our content, since a browser could be trivially modified to
  > release the keys.

  The keys would be exposed to browser code in the straw feature I described, yes. Note that if the CDM exposes clear pixel and audio sample data to the browser, the browser could be modified to dump the pixel and audio sample data in which case releasing the keys would be moot.

> another issue is that you (presumably) rely on
> standard HTTP authentication to authenticate the user.

HTTP-only cookies over HTTPS could be used. It's very secure if the browser hasn't been modified and the straw feature I outlined didn't attempt to deal with browsers instrumented specifically to defeat the feature, since instrumented browsers could dump content in the case where a CDM provides clear pixels and audio samples to the browser anyway.

The key loading HTTP request could include a Sec-Foo header that can't be faked with XHR.

> Performing the decryption at the HTTP layer has the following problems:
> - HTTP servers have to support the additional headers, which means CDN
> changes are necessary. Even trivial CDN changes take time to roll out across
> the world and CDNs generally want some compensation. Anything which
> restricts our choice of CDNs increases our costs and reduces user quality of
> experience (due to reduced diversity). So there's value to us if the
> solution works with existing HTTP services.

So adding a couple of static-per-file HTTP headers to CDN content is too onerous for you, but you consider it reasonable to ask browsers to deal with all the complications of supporting CDMs. Wow.

Comparison of EME with the existing plugin system

Mark Watson has repeatedly claimed that EME doesn’t try to be *perfect* a standard and what it mostly does is that that it improves upon the existing plugin based system to provide protected content.[47] This view is countered on two levels, one is that only being better than the plugin based system isn’t a good enough reason for EME to be part of W3C, on the second level Boris Zbarsky have argued that EME isn’t actually better than the plugin based system.

[47] https://lists.w3.org/Archives/Public/public-html/2012Feb/0350.html

- Boris pointed out that the existing Flash based plugin system doesn't entry barriers for new browsers because a correct implementation of NPAPI is enough for the Flash plugin to work with the browser[48], whereas the API for interaction with CDMs is not specified in EME and new browser would have to go into contracts with the CDM providers and they can restrict the new players from coming in.

- The legal difficulties with EME can be worse because the CDM providers can make it illegal for the browser developer to release the CDM interaction API through contracts [49] and they can make it illegal to reverse engineer the API using the anti-circumvention provision of DMCA.[50] (As a side note reverse engineering can be more difficult with EME with respect to secondary copyright infringement liability because Flash is also used for games and other stuff where the intention to use Flash is not to prevent copying, whereas the primary intention of EME is to ease DRM. Existence of Flash alternatives Gnash, Ligthspark and Shumway shows that DMCA is isn't much of an issue with Flash.)

- Mark Watson argues that there are practical difficulties for new browser with flash even if they have an implementation of NPAPI according to the specification, he says CDMs provide more control to browsers because they are smaller and have much less functionality.[51]

- Ian said "*There is no value to the user for you to use HTML rather than Flash if you're still going to require a proprietary plugin. The value of an open standard is that anyone can write an interoperable user agent without needing to coordinate with anyone else, and get full access to all the content on the Web. DRM completely undermines this. It compromises the very purpose of having a standard*."[52]

- Henri said "*A single mechanism that doesn't have secret parts of implementation is superior to pluggable CDMs, because a single non-secret standard mechanism avoids vendor lock-in.*"[53]

- According to Henri plug-ins are different from CDMs, because[54]:
  1) There is only one plug-in (Flash Player) to deal with
  2) It's possible to code your way around most .swf content on the
  Web: You can write an independent implementation of .swf player that

[48] https://lists.w3.org/Archives/Public/public-html/2012Feb/0427.html
[49] https://lists.w3.org/Archives/Public/public-html/2012Feb/0448.html
[50] https://lists.w3.org/Archives/Public/public-html/2012Feb/0429.html
[51] https://lists.w3.org/Archives/Public/public-html/2012Feb/0428.html
[52] https://lists.w3.org/Archives/Public/public-html/2012Feb/0497.html
[53] https://lists.w3.org/Archives/Public/public-html/2012Feb/0498.html
[54] https://lists.w3.org/Archives/Public/public-html/2012Mar/0025.html

> deals with restaurant menus. If you license the codec patents, you can support video, too. Full compatibility by coding around the problem stops at the CDM inside Flash Player, so at the limit, the situation reduces to one CDM. (Though even one CDM that your product isn't compatible with might be one too many.)
>
> In the case of launching a new browser for an existing OS that already has NPAPI Flash Player, the situation is different from CDMs, because you can implement an NPAPI host royalty-free, without asking permission and without making an bizdev deals. (You get Silverlight and Java as a bonus with some more bug workarounds.)

Implementation of EME in Free and Open Source Browsers

- Tab Atkins said that DRM is based on encryption, and all good encryption is open-sourced. But it requires the consumer's software to have the decryption key without giving the consumer the key. This is obviously impossible in open-source software.[55]
- Mark Watson argued that EME is not an issue with Firefox from specification point of view because they don't specify the content Decryption Module.[56] Also a Content Decryption Module implementing the 'clearkey' keysystem can be implemented as Open Source.[57]

- Glenn Adams argues that Firefox can use a OS API like mechanism of implement EME where they don't have to reveal the source code the decryption module.[58] He implementations where complete source is available he said "*The issue isn't whether it can be implemented in open source software (it can), the issue is whether encrypted content can be decoded and presented to a user on a device that uses such implementation while simultaneously satisfying further constraints imposed by licensing terms by content providers who insist on using content protection with acceptable impediments to unauthorized access*.

  If such content providers cannot be assured of such protection, then they may not make the content available through such means."[59] On which John C.

Vernaleo said "*Saying that something can be implemented but not be used seems to be a very odd definition of implementing to me.*"

[55] https://lists.w3.org/Archives/Public/public-html/2012Feb/0326.html
[56] https://lists.w3.org/Archives/Public/public-html/2012Feb/0324.html
[57] https://lists.w3.org/Archives/Public/public-html/2012Feb/0377.html
[58] https://lists.w3.org/Archives/Public/public-html/2012Feb/0322.html
[59] https://lists.w3.org/Archives/Public/public-html/2012Feb/0375.html

- Henri Sivonen said a more interesting to ask is if Netflix et al. are willing to use open source implementations of CDMs to provide their content.[60]

- EME needs browser media stack to have some secret which cannot be easily obtained, Andreas see it as a proposal “*to create the foundation for "specific user agent solutions" which "open source browsers" can not "natively support.*"[61]

- Mark Watson agreed that fully open source browsers might not be able to support DRM within the browser itself but they can use CDMs, he also pointed out that EME might be incompatible with GPLv3.[62]

- Henri puts out some requirements from (open source) browser point of view [63]:
    - The system is fully specified and doesn't involve any implementation-side secrets.
    - The system can be implemented by anyone and in Open Source software.
    - The system doesn't require browsers to interface with 3rd-party black boxes that the browser vendors don't control
- Henri said that an attacker on a open source CDM will need to pull out the source of the browser, instrument it to dump decrypted frames and audio samples, compile and run the instrumented browser. The cases where the restrictions aren’t inbuilt in the hardware, EME doesn’t not provide substantially more secure content protection.*“GPLv3-compatibility is a useful shorthand. If a solution couldn't be implemented under GPLv3, chances are it is encumbered in ways that are harmful to competition when it comes to launching new browsers or new Web-capable devices.”* [64]

Interoperability and CDM specification

- Kornel stated the EME spec does very little to fulfil its stated goal which is to provide easy and interoperable content protection[65]. Specifically
    - It leaves the interaction between CDM and the browser undefined, so each CDM provider will have to cooperate with every single browser vendor it's willing to support, and browser vendors may need to implement proprietary CDM APIs several times.

[60] https://lists.w3.org/Archives/Public/public-html/2012Feb/0379.html
[61] https://lists.w3.org/Archives/Public/public-html/2012Feb/0482.html
[62] https://lists.w3.org/Archives/Public/public-html/2012Feb/0483.html
[63] https://lists.w3.org/Archives/Public/public-html/2012Feb/0500.html
[64] https://lists.w3.org/Archives/Public/public-html/2012Mar/0003.html
[65] https://lists.w3.org/Archives/Public/public-html/2012Feb/0444.html

    - Very few companies have enough market power to establish a new DRM approved by "Hollywood", so it's quite possible that the current plugin problem will just morph into an identical CDM problem
    - The spec only tries to tackle relatively easy client-side part and a unified server-side API which is much important to enable diversity of protection systems is left out scope.
- Vickers, Mark supported having a standard CDM API saying it will be useful and it would make sense to work on through W3C or some other group.[66]
- Glenn Adams said "*defining an ABI between browsers and CDM implementations is a reasonable task, but *not* a task for the W3C.*"[67]
- Mark Watson said there could be an API if people want it to be there but there is no such api for media or plugin codecs.[68] Charles Pritchard had also made this observation about codecs earlier.[69]
- Henri said "*A single mechanism that doesn't have secret parts of implementation is superior to pluggable CDMs, because a single non-secret standard mechanism avoids vendor lock-in.*"[70]
- Andreas demanded examples of content providers who "*demands using a "Content Decryption Module" but does not demand such "platform-specific capabilities to protect the rendering path*"[71] on which Mark Watson said that "*the requirements of content providers vary by provider, by content and by device (and probably on other axes too). You should not expect to see an enumeration of them in this discussion*"[72]
- Boris said that making it easy for consumers to watch TV and movie online is a shared goal but one caveat he would have *"is that in an ideal world a user should be able to do this on a device of their choosing, including a device the user built himself (or something like OLPC, etc; devices that happen to be produced by someone who is not an established player in the device-making space).*

  Or put another way, we want to avoid creating barriers to entry into the device-making space, if we can. That means avoiding unnecessary restrictions on both the hardware and the software of potential new devices."[73]
- Mark responds saying the EME reduces the that it should be possible for the device manufacturers to support CDMs and unless we are talking about GPLv3 software stack reducing entry barrier for unestablished manufacturers shouldn't be a problem.[74]

[66] https://lists.w3.org/Archives/Public/public-html/2012Feb/0439.html
[67] https://lists.w3.org/Archives/Public/public-html/2012Feb/0442.html
[68] https://lists.w3.org/Archives/Public/public-html/2012Feb/0437.html
[69] https://lists.w3.org/Archives/Public/public-html/2012Feb/0337.html
[70] https://lists.w3.org/Archives/Public/public-html/2012Feb/0498.html
[71] https://lists.w3.org/Archives/Public/public-html/2012Feb/0482.html
[72] https://lists.w3.org/Archives/Public/public-html/2012Feb/0483.html
[73] https://lists.w3.org/Archives/Public/public-html/2012Feb/0508.html
[74] https://lists.w3.org/Archives/Public/public-html/2012Feb/0516.html

- Boris says “unnecessary requirements” includes requiring licensing for a CDM.[75]
- Mark said “*I think we have to find models in which CDMs can get onto the device without any payment from the device vendor. This list isn't probably isn't the right place to resolve that.*”[76]

- Henri said “*Without knowing the nature of the CDMs, the impact of the proposal can't be evaluated. It can't even be evaluated if the proposal proposes a sensible API without having a good idea of what kind of things CDMs would be.*”[77] Specifically “*What does the CDM do? Who has the permission to write, ship and execute code that does what the CDM is required to do? How are these permissions given to others? Are there parts of the implementation (i.e. stuff that isn't downloaded from the site as part of content; e.g. keys that don't arrive from the site) that are secret (i.e. cannot be put in a public source code repository)? If there's a secret component, who decides who gets to know that secret and incorporate it in products? Under what terms?”* He further asserts that “*Generally with W3C specs, there are no deliberately secret parts and an implementer doesn't need to ask for permission, since the specs are royalty-free to implement.*” and it should remain the same with EME.[78]
- Mark clarified that a browser can have multiple CDMs with multiple keysystems and what they propose is to standardize is the discovery, selection and interaction with CDMs, not the CDMs themselves (like codecs). Anyone can create and ship a CDM without a permission and Henri’s other questions are about the commercial choices of the browser and CDM manufacturers. Having a browser side of CDM API which can be implemented in a DMCA-safe way is that they need to work on in the proposal.[79]
- Tab Atkins said that the current codec situation is horrible and *“we (W3C) should not add another codec-war-style situation to the web platform unless we absolutely can't avoid it and the payoff is sufficiently great.”*[80]
- Charles said the codec situation is present because they the vendors don’t want a standard codec mainly because they don’t want to struct in technology which will be obsolete in few years. Given the existing not specifying CDMs seems reasonable.[81] In support Clarke said that industry is full of outdated standards that gets dragged along, a solution can suggesting a baseline CDM implementation and encouraging the implementers to use it.[82]

[75] https://lists.w3.org/Archives/Public/public-html/2012Feb/0517.html
[76] https://lists.w3.org/Archives/Public/public-html/2012Feb/0518.html
[77] https://lists.w3.org/Archives/Public/public-html/2012Feb/0500.html
[78] https://lists.w3.org/Archives/Public/public-html/2012Mar/0003.html
[79] https://lists.w3.org/Archives/Public/public-html/2012Mar/0012.html
[80] https://lists.w3.org/Archives/Public/public-html/2012Mar/0013.html
[81] https://lists.w3.org/Archives/Public/public-html/2012Mar/0014.html
[82] https://lists.w3.org/Archives/Public/public-html/2012Mar/0016.html

- Mark Watson said that a codec like situation is unavoidable given that IPR issues disallows a CDM which is fully specified by W3C.[83]
- Responding to Baron he lists out possible types of CDMs[84]
  - CDMs embedded into device firmware, most likely for TVs and similar devices
  - CDMs shipped with an OS
  - CDMs shipped with a browser
  - CDMs shipped with a browser in 'inactive' form and 'activated' by a service which needs them
  - CDMs installed by the user
  - CDMs whose installation is triggered by a particular service (and presumably oked by the user)
- Fully specified CDM specs might also be problematic because of patents[85]
- Henri said that CDM is different from codecs because[86]:

  1) So far, the list of CDMs isn't down to a very small known number (3 in the case of codecs at the moment, though H.265 or Daala could increase the number)

  2) In addition to being encumbered by patents, CDMs could be encumbered by trade secrets. (If there aren't secret algorithms, there might be secrets keys that an implementation needs to incorporate in order to be compatible.)

  3) In addition to being encumbered by patents, it seems (IANAL, though) that being compatible with a CDM without permission might run afoul with anti-circumvention laws.

Privacy

- In context of browser extensions Charles once argued that EME can help protect privacy of users by ensuring that the content consumed by users doesn't get intercepted by third parties.[87]

Accessibility

- W3C’s principal accessibility John Foliot strategist asked to strip way the philosophical arguments about EME and think about from a purely technical point of view and then

[83] https://lists.w3.org/Archives/Public/public-html/2012Mar/0018.html
[84] https://lists.w3.org/Archives/Public/public-html/2012Mar/0020.html
[85] https://lists.w3.org/Archives/Public/public-html/2012Mar/0015.html
[86] https://lists.w3.org/Archives/Public/public-html/2012Mar/0025.html
[87] https://lists.w3.org/Archives/Public/public-html/2012Feb/0393.html

he justifies EME by saying it what the economy wants. About DRM he said don’t use DRM if you don’t like it.[88]

- Ian Hickson draws a satire of John's "Don't use it if you don't like it" argument about DRM, argues that EME intentionally makes content inaccessible hence it is unethical.[89] Glenn Adams called Ian's argument non-sense and says DRM/Content Protection has nothing to do with impaired users.[90]

- Supporting Ian, Benjamin Hawkes-Lewis posted weblinks describing various accessibility hurdles that come up due to DRM.[91] To which Glenn Adams said that DRM/CP doesn't intentionally discriminates against accessibility features. He then points out that HTML5 text track facility offer workarounds in case where accessibility features are missing.[92]

- Henri said "*Even if one accepted the notion that DRM discriminates equally against non-accessibility and accessibility features, discriminating against accessibility features at all deserves special attention*", DRM foils the exceptions provided by copyright law for accessibility purposes, an example can be case where the DRM proprietor places a contractual requirements on implementors prohibiting sending the audio to a speech recognition system for generating captions on the fly on the client side.[93]

- Mark Vickers said that accessibility is a motivation for EME and it improves upon many of the accessibility issues in plugin systems used to play protected content by providing standard ways to add accessibility features.[94]

Security

[88] https://lists.w3.org/Archives/Public/public-html/2012Feb/0338.html
[89] https://lists.w3.org/Archives/Public/public-html/2012Feb/0354.html
[90] https://lists.w3.org/Archives/Public/public-html/2012Feb/0362.html
[91] https://lists.w3.org/Archives/Public/public-html/2012Feb/0405.html

[92] https://lists.w3.org/Archives/Public/public-html/2012Feb/0408.html
[93] https://lists.w3.org/Archives/Public/public-html/2012Feb/0412.html
[94] https://lists.w3.org/Archives/Public/public-html/2012Feb/0420.html

Support and Opposition to EME proposal

Support

- Eric Carlson said EME proposal is workable.[95]

- Carr, Wayne supports the proposal without justification.[96]

- Bob Lund supports the proposal by saying it is a good first step to support problems of content providers and device manufacturers.[97]

- Mark Vickers, representing Comcast, strongly supported the EME proposal claiming that EME will go a long way towards moving most of the will be a vital step enabling commerce and communications through web browser.[98]

- Glenn Adams on the behalf of Cox Communications supported "*the W3C defining a reasonable solution to enable the playback of DRM and/or Copy Protected media via HTML5."*[99]

Opposition

- Boris Zbarsky said that EME is bad for the web if it requires new browsers to coordinate with companies while coming up.[100]

- Andreas Kuckartz opposed the proposal if it cannot be implemented by Open Source Software.[101]

- Ian Hickson called the proposal unethical and said "*the* (proposal) *does not provide robust content protection, so it would not address this use case even if it wasn't unethical.*"[102]

[95] https://lists.w3.org/Archives/Public/public-html/2012Feb/0314.html
[96] http://lists.w3.org/Archives/Public/public-html/2012Feb/0341.html
[97] https://lists.w3.org/Archives/Public/public-html/2012Feb/0347.html
[98] https://lists.w3.org/Archives/Public/public-html/2012Feb/0391.html
[99] https://lists.w3.org/Archives/Public/public-html/2012Feb/0369.html
[100] https://lists.w3.org/Archives/Public/public-html/2012Feb/0352.html
[101] https://lists.w3.org/Archives/Public/public-html/2012Feb/0374.html

[102] https://lists.w3.org/Archives/Public/public-html/2012Feb/0274.html