



VIDEO INFRASTRUCTURE FOR WIKIMEDIA

Now and in the future

Brion Vibber
Wikimedia Foundation



October 10, 2015

VIDEO CONCERNS

- Playback
- Archiving
- Ingest
- Creation



VIDEO CONCERNS

- **Playback**
- Archiving
- Ingest
- Creation



The thing that affects the most people the most directly is playback. Without working playback, literally nothing else we do with video matters.

WAY BACK IN 2007...



Håkon Wium Lie

Published on 17 April 2007 in [Blog](#). [Edit this article on GitHub](#).

Licensed under a [Opera Software ASA](#) license.

[open-web](#)

A Call for Video on the Web

It's time to make video a first-class citizen of the web. We, the users, have video cameras in our pockets and the bandwidth to transfer more clips and streams than we can watch. What's missing is an easy way to integrate video into web pages, and native support for video in browsers. We, the web community, should address this by adding a [video element](#) to HTML.

In addition to giving video an HTML element, we must also agree on a baseline video format that will be universally supported, just like the GIF, JPEG and PNG image format are universally supported. It's

In 2007, Opera proposed a <video> tag for HTML5, with a sample implementation using the free Ogg Theora codec. The theory was simple: make video a first-class citizen of HTML web pages, just like images. And just like images, a baseline compatible format was envisioned as vital.

``



`<video src=galaxy.ogg>`



And for basic uses, it works pretty well with that original simple vision. An HTML document can embed a standalone video or audio file and have it available for playback without jumping through too many hoops.

WHAT'S HARD?

- Lack of baseline format
- Huge variances in bandwidth
- Processing time for size/bandwidth variants

So problem solved right? Let's pack our bags and go home. Nope, not yet.

WHAT'S HARD?

- **Lack of baseline format**
- Huge variances in bandwidth
- Processing time for size/bandwidth variants

First problem is that HTML5 video still lacks a baseline format — unlike images where the standard JPEG, PNG, and GIF will always work, you can't quite rely on a single format to work in all browsers.

FORMAT WARS: PART I



Early drafts of the HTML5 <video> spec recommended Ogg Theora as a baseline format. It wasn't the top of the line codec even then, but was free, implemented in open source software, and all known patents were freely licensed by On2 which had developed the VP3 codec it was based on.

FORMAT WARS: PART 2



The idea of a baseline codec died in committee, however. Apple, Microsoft and Nokia objected to Theora on both quality and concern over potential unknown third-party patent liability, while Mozilla objected to the MPEG-4 formats over the very much-known liability of the MPEG-LA patent pool organization.

FORMAT WARS: PART 2



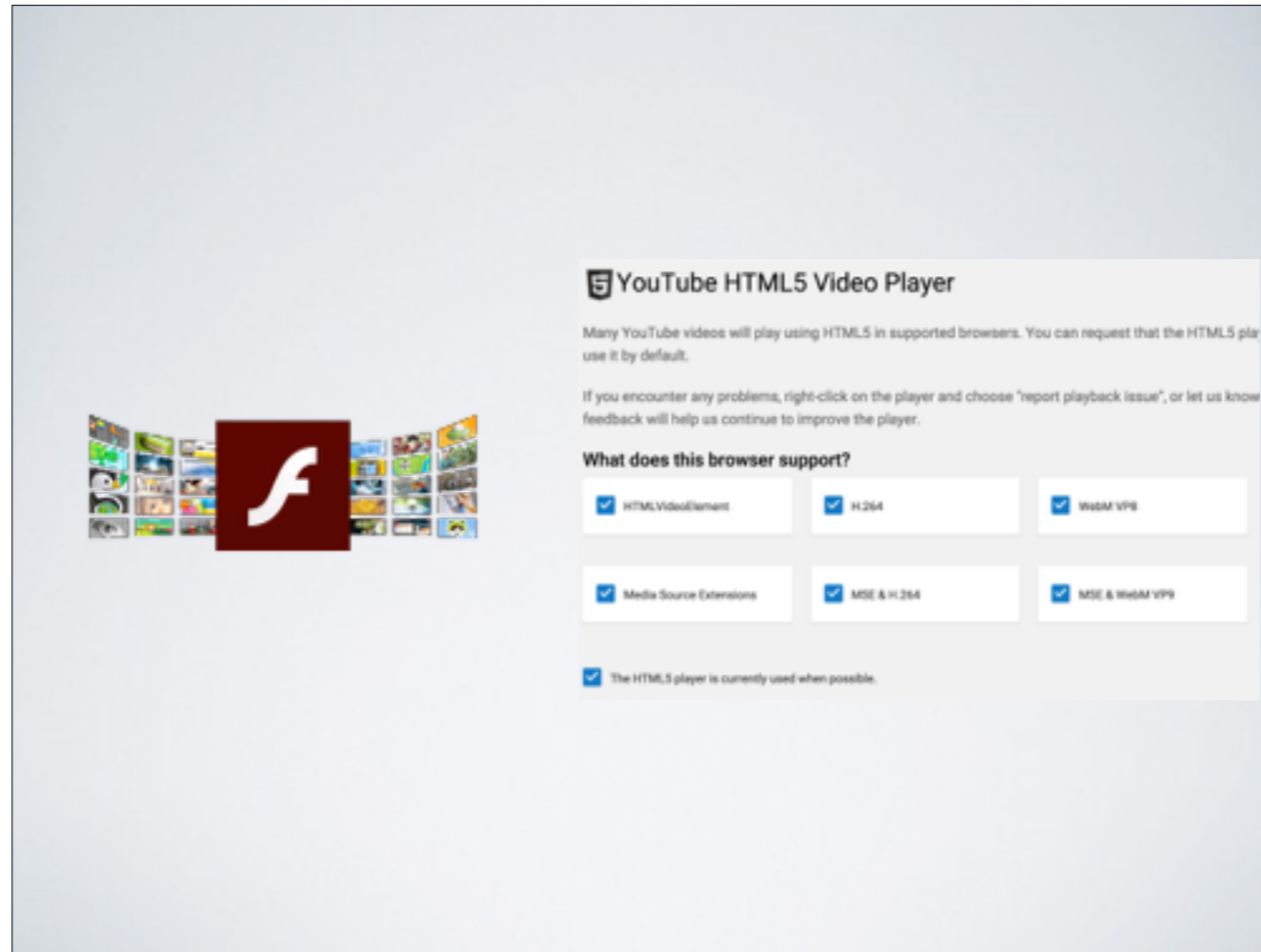
Licensing of patented formats is easy to deal with if you're a manufacturer of devices like TVs or cameras: you just factor it into your costs like any other part or process, and pass it on to the consumer. Free/open source software is another story; organizations like Mozilla could in theory license the patents for their distribution — as Google does with Chrome — but anyone else building and distributing the software, such as Linux distributors, would not be able to. To this day, Firefox and some open-source versions of Chromium lack full H.264 support as the standoff continues.

FORMAT WARS: PART 3

web▶m



In 2010, Google launched the WebM format using the VP8 codec. Like Theora, it was based on tech from On2, but a later generation that was more comparable to MPEG-LA's H.264. Unlike Theora, the royalty-free patent license was backed by a big company. Mozilla and Opera signed on immediately; Microsoft and Apple continued to rebuff the idea and considered the patent grants insufficient.



The image shows a screenshot of the YouTube HTML5 Video Player interface. On the left, there is a grid of video thumbnails with a large red square containing a white 'f' logo (the Adobe Flash logo) overlaid on top. To the right, the text reads: 'YouTube HTML5 Video Player', 'Many YouTube videos will play using HTML5 in supported browsers. You can request that the HTML5 player use it by default.', and 'If you encounter any problems, right-click on the player and choose "report playback issue", or let us know feedback will help us continue to improve the player.' Below this is a section titled 'What does this browser support?' with a list of supported features, each with a blue checkmark icon: HTMLVideoElement, H.264, WebM VP8, Media Source Extensions, MSE & H.264, MSE & WebM VP8, and 'The HTML5 player is currently used when possible.'

For the most part, end-users have been insulated from this by the use of the Adobe Flash plugin to add H.264 to Firefox, and/or dual H.264 + WebM stacks on consolidated providers like YouTube.



5 YEARS PASS

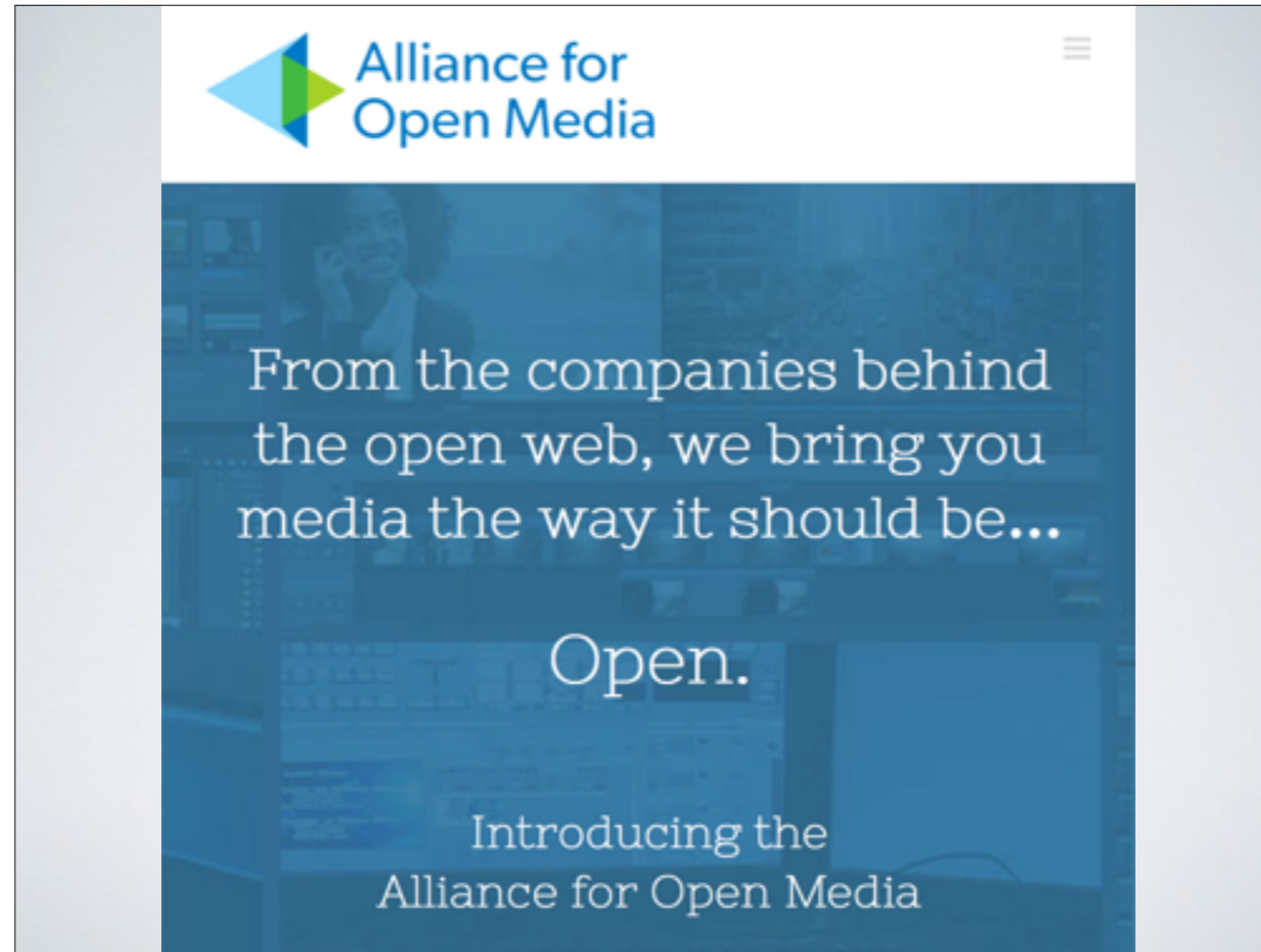
5 years pass. Google keeps improving WebM, releasing VP9 with improved compression. The biggest user of WebM remains Google's own YouTube. Hardware VP8 and VP9 decoding starts to creep into devices thanks to Google lobbying smartphone and smart TV manufacturers. Mozilla funds additional codec research just in case.

This discussion is now closed and archived.

Please do not modify it.

Closing with consensus **prefer no MP4 support.**

Wikimedia remained picky about the openness of the formats, keeping us from using the dual format technique that YouTube does. Commons contributors in 2014 rejected an RFC to pursue even limited MP4 support, leading WMF's Multimedia Team to abandon further video-related work for the foreseeable future.



In 2015, some interesting developments. In addition to Google and Mozilla, traditionally H.264-heavy shops like Cisco, Netflix, and Microsoft join in the Alliance for Open Media to help smooth out any IP issues around a next-generation royalty-free video codec standard. On the technical end, Google's VP10, Mozilla/Xiph's Daala, and Cisco's Thor codecs are all contributing ideas to the IETF NetVC working group for a next-next-gen standard.

SEPTEMBER 8, 2015 10:00 AM / by [Microsoft Edge Team](#)

Announcing VP9 support coming to Microsoft Edge

75 SHARES

 SHARE TWEET SHARE

Today, we're excited to announce that [WebM/VP9](#) support is now [in development](#) in Microsoft Edge. VP9 is an open source video format that offers efficient compression to stream HD content at lower bitrates, and is well suited to UHD streaming. Initial support for VP9 will be available in Windows Insider Preview builds soon. This is part of our continuing effort to expand codec offerings in Windows. We continue to evaluate other formats and look forward to receiving feedback as we work on implementing them.

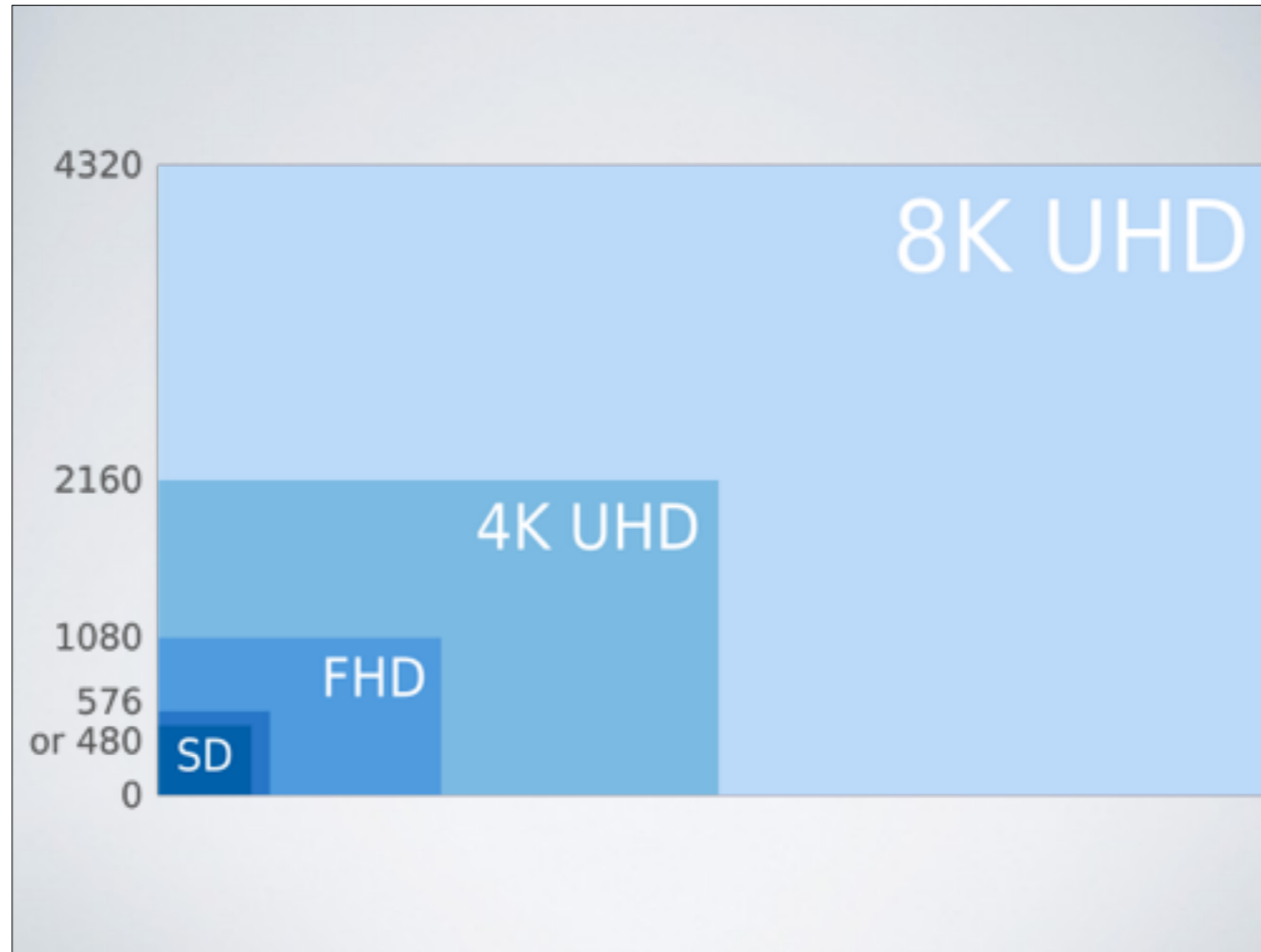
VP9 Support

Our initial VP9 implementation in Microsoft Edge will support adaptive streaming using [Media Source Extensions](#), and will be detectable using the `MediaSource.isTypeSupported()` API. It will be specifically targeted to meet the needs of websites that use VP9 to deliver video in combination with MP4/AAC or other audio codecs already supported by Microsoft. We are working on future support for VP9 for media tags and local playback, as well as considering support for additional audio formats likely to be used with

Microsoft has even started work on adopting WebM VP9 support — not just through a hidden third-party plugin, but natively and standard.

WHAT CHANGED?

Why the change of heart? What's different in 2015 from 2010 or 2007?



First, displays are getting sharper, bringing a demand for higher-resolution video. As higher-end computers and TVs move from HD to Ultra-HD, next-generation codecs can save huge amounts of bandwidth. If you're Netflix or YouTube or Xbox Video or handling lots of video conferencing... that's money in the bank.

The screenshot shows the HEVC Advance website's licensing page. The navigation bar at the top includes the HEVC Advance logo and links for overview, technology, licensing, goals, press, careers, and a contact link with an envelope icon. The main heading is "licensing". Below this, there are four download links, each with an icon and a "download .pdf" button:

- HEVC Advance Royalty Rate Summary** (tag icon) - download .pdf
- HEVC Patent License Summary** (document icon) - download .pdf
- HEVC Advance Tier 1 and 2 Country List** (list icon) - download .pdf
- HEVC Advance Brand Guidelines** (cube icon) - download .pdf

Additionally, H.264's presumed successor, HEVC, managed to somehow gain a *second* industry patent licensing organization this year. Now not only is MPEG-LA shaking you down, so is "HEVC Advance"... this largely destroyed the old claim that having an industry patent pool made users and developers safer versus the explicitly royalty-free formats.



Apple is still holding out so far, with only H.264 supported in Safari and iOS, and no commitment to AOMedia or NetVC.

MAD WORKAROUNDS!

6:22 AM 98%
brionv.com

ogv.js decoding Theora and WebM video in JavaScript

Time lapse of Monument Valley, Navajo Tribal Park.webm

Source: 360p Ogg Player: JavaScript (auto)

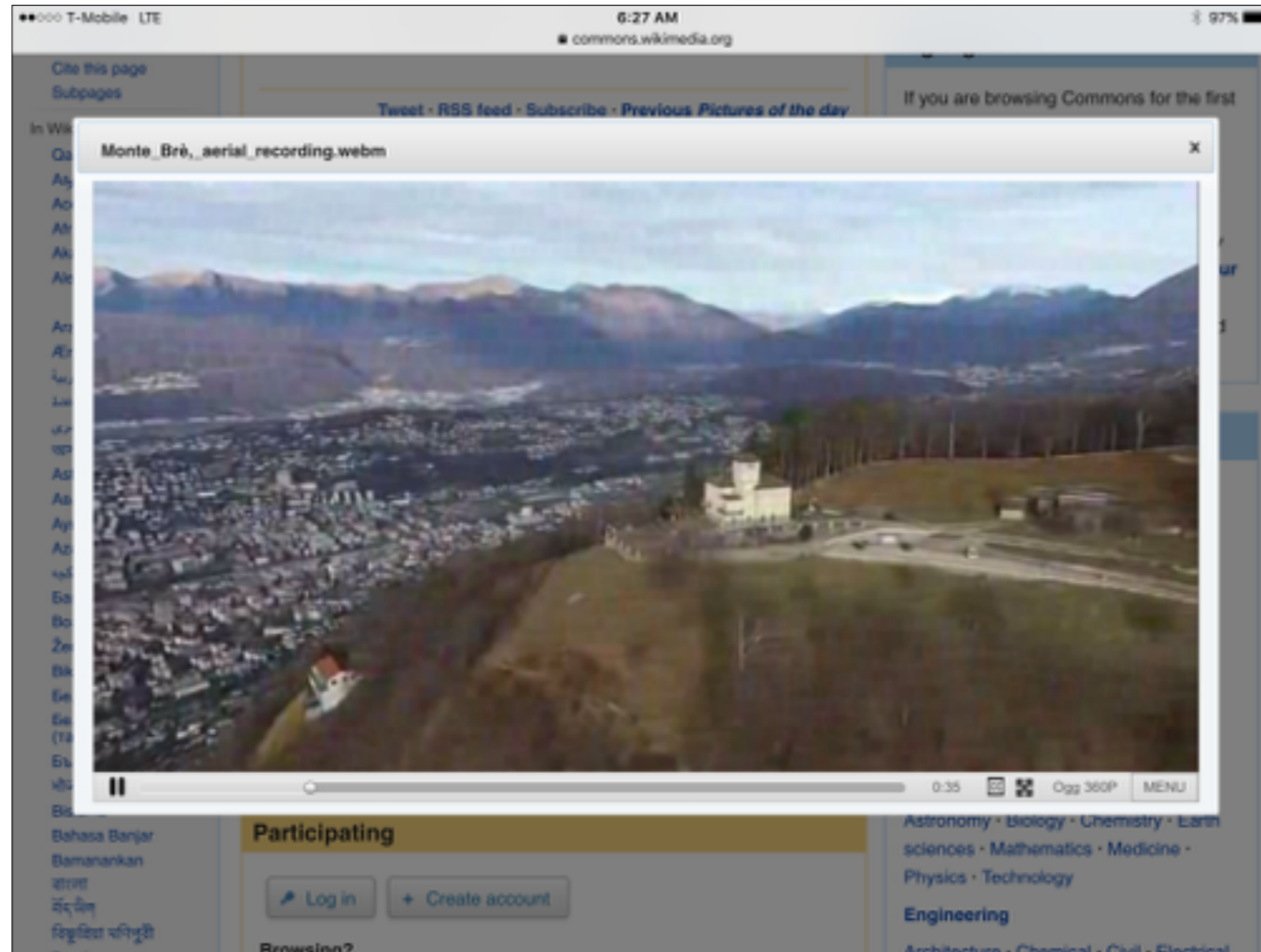
Video 23.98fps 640x360 (jitter 1.97ms)

Audio 44100 Hz, 2 ch (13 drops, 3.5s delay)

target	clock	cpu	mux	vid	aud	buf	draw
41.7	41.6	16.4	0.5	10.6	4.3	0.1	0.8

red: target frame rate
blue: frame-to-frame clock time
black: per-frame CPU time

In 2013 I started an experimental project “ogv.js”, a plugin-free media player built in JavaScript. Thanks to huge improvements in recent years to JavaScript performance, this actually runs well enough in current versions of Safari and Internet Explorer/Edge to play back modest-resolution files on most laptops and newer 64-bit iPhones.



We've integrated it into the media player on Wikipedia and Wikimedia Commons; currently this uses the older Ogg Theora format for best performance, but I'm working on WebM support too which will improve picture quality in scenes with heavy motion.

<http://brionv.com/misc/ogvjs-gstreamer-conf-2015.pdf>

Please enjoy some technical slides on JavaScript media playback!

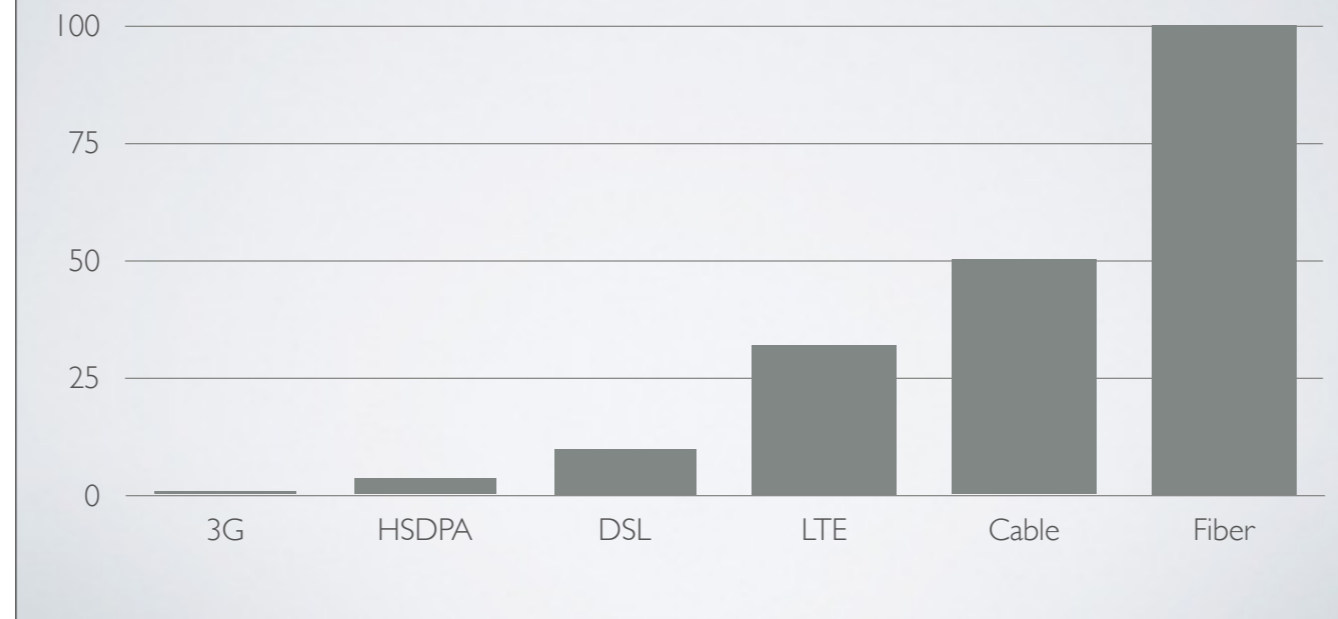
For those interested in more technical details on the JavaScript playback, please check out the slides from my talk at GStreamer Conference 2015!

WHAT'S HARD?

- Lack of baseline format
- **Huge variances in bandwidth**
- Processing time for size/bandwidth variants








Ok so we've got native playback on Chrome and Firefox, and JavaScript playback in Safari and IE. What's next?

BANDWIDTH IS NOT EQUAL



Internet bandwidth varies dramatically from place to place, provider to provider, etc. Creating a single video file that is suitable for streaming over both slow mobile networks and fast cable or fiber is impossible!

MY NEW YEARS' RESOLUTION IS 640X480

Format	Bitrate	Download	Status	Encode time
WebM 720P	840 kbps		Completed 07:54, 19 August 2015	58 s
WebM 480P	620 kbps		Completed 07:54, 19 August 2015	1 min 4 s
WebM 360P	341 kbps		Completed 07:54, 19 August 2015	42 s
Ogg 480P	1.81 Mbps		Completed 07:54, 19 August 2015	27 s
Ogg 360P	945 kbps		Completed 07:53, 19 August 2015	15 s
Ogg 240P	498 kbps		Completed 07:53, 19 August 2015	9.0 s
Ogg 160P	183 kbps		Completed 07:53, 19 August 2015	5.0 s

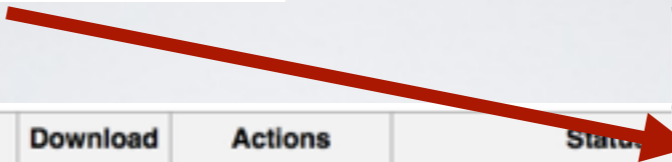
So, whether you're Wikipedia or YouTube, as a video service provider you end up having to produce multiple versions of your file at different resolutions to fit different network bitrates and processing speeds.







WHAT'S HARD?

- Lack of baseline format
- Huge variances in bandwidth
- **Processing time for size/bandwidth variants**

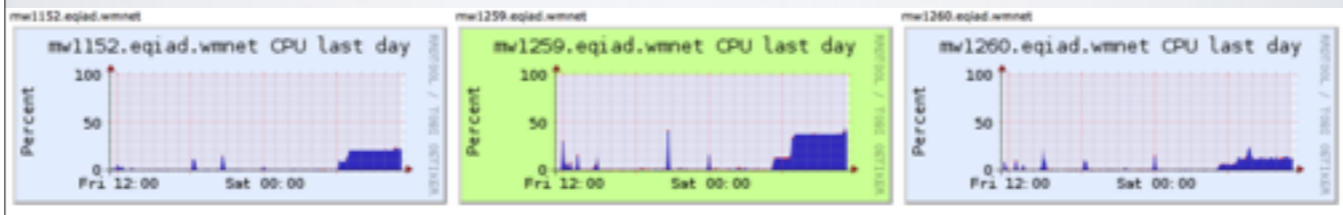
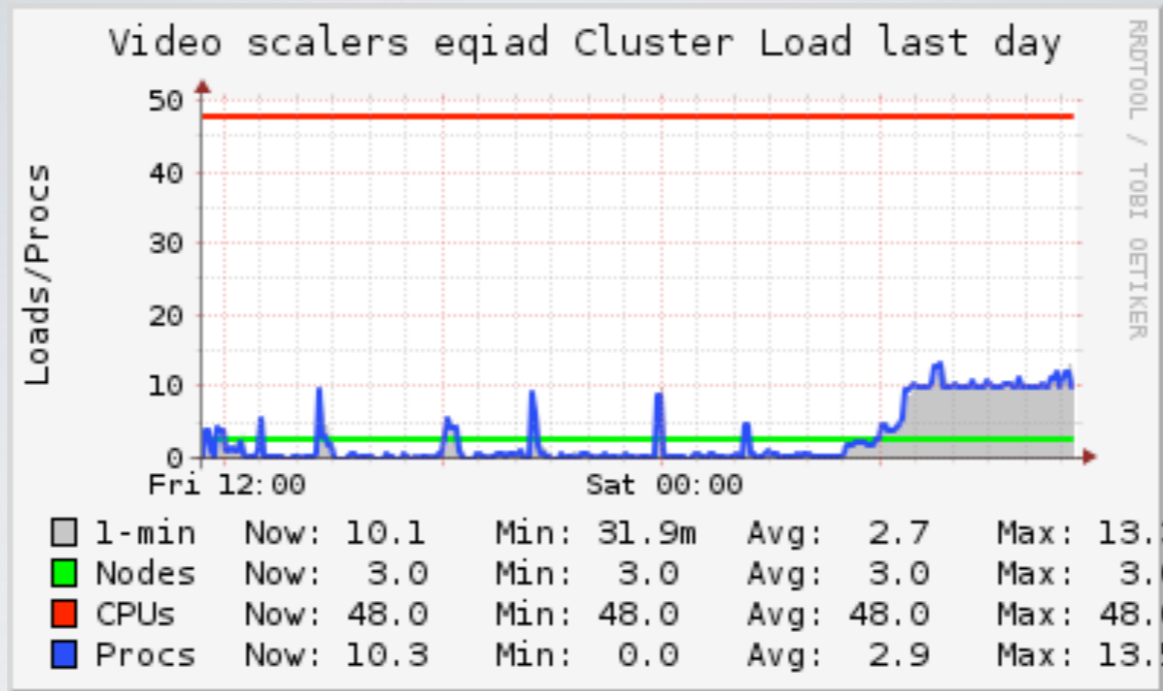
This leads us to the server-side resources needed to support these multiple formats and sizes.

length 12 min 44 s



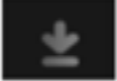
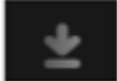
Format	Bitrate	Download	Actions	Status	Encode time
WebM 480P	1.09 Mbps		Reset transcode	Completed 18:24, 9 October 2015	22 min 17 s
WebM 360P	575 kbps		Reset transcode	Completed 18:14, 9 October 2015	11 min 42 s
Ogg 480P	2.01 Mbps		Reset transcode	Completed 18:22, 9 October 2015	20 min 13 s
Ogg 360P	1.04 Mbps		Reset transcode	Completed 18:14, 9 October 2015	12 min 17 s
Ogg 240P	525 kbps		Reset transcode	Completed 20:43, 8 October 2015	6 min 45 s
Ogg 160P	190 kbps		Reset transcode	Completed 18:05, 9 October 2015	2 min 46 s

Re-encoding video can be very slow... Unlike images which we can rescale on-demand relatively easily, it may take minutes — or hours — to encode a video of non-trivial length. With only a handful of servers dedicated to the task, long HD videos can literally take hours to fully process.



We currently have just 3 machines dedicated to video transcoding, and the load is not shared as well as it could be among them.

VP9 ~4x slower than VP8

Format	Bitrate	Download	Status	Encode time
VP9 1080P	2.2 Mbps		Completed 16:05, 14 August 2015	43 min 1 s
WebM 1080P	2.45 Mbps		Completed 10:42, 14 August 2015	10 min 48 s

As we start upgrading our WebM output to the newer VP9, we'll save bandwidth but it can take about 4x as long to encode. In the long term we'll have to invest in a better transcode pipeline to keep things moving fast... especially if we want to be able to take bulk uploads of material from archives. Which we do!

USE THE CORES, LUKE



The total encoding time is dominated by the highest-resolution version or two of the file. VP9 has a “tiled” encoding mode which allows using multiple CPU cores more efficiently, which will help speed up HD encoding.

MULTISERVER PIPELINE



A smarter transcode pipeline could divide up a large file into chunks which can be simultaneously processed on separate machines, then combined into a single output file. This has less direct benefit for bulk uploads, but gets each individual file done more quickly — good for individual uploads of timely news footage, for instance.

LEAVE IT TO HARDWARE?



Hardware encoders for VP9 may be a big help for faster encoding as well — especially if we ever want to do realtime streaming & recording of events. We're still waiting on something productized and purchasable, but I'm in contact with folks at Google who are more in the know as to what's in the works.

PLAYBACK SOLVED!
NOW WHAT?

Ok, consider playback issues “mostly solved” for now. What’s next?

VIDEO CONCERNS

- Playback
- **Archiving**
- Ingest
- Creation



Wikipedia and Commons are not just for the web readers of today; it's for the future, and for re-users and remixers and republishers. We have to learn some lessons from the archival world.

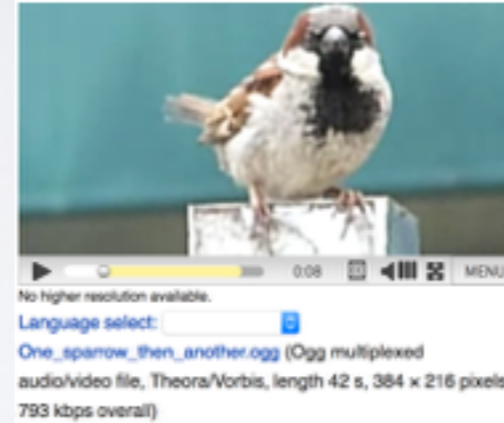
WIKIPEDIA AS ARCHIVE

- **Long-term** preservation
 - Need standard, documented formats implemented by open source
 - Use highest quality source available

Computing history is littered with obsolete data formats that are difficult to read and use today. For the long term it's important to use formats that will be usable in the future; well-documented open standards are very, very good here. Open-source implementations are vital. And if converting will be lossy, using high-quality source materials whenever possible is important.

AGING FORMATS

384x216 :(



A lot of the early video clips uploaded to Wikipedia in the 2000s were low resolution, and relatively low quality to fit our strict file upload size limits. Some can be upgraded to higher quality versions from other sources, but original content may be stuck at low quality forever.

(RE)COMPRESSION BLUES



Size of this resource: 400 x 160 pixels. Other resolutions: 500 x 160 pixels, 640 x 160 pixels, 1,024 x 160 pixels.
Original file (Ogg multiplexed audio/video file, Theora/VP8, length 4 min 27 s, 1,280 x 720 pixels, 3.85 Mbps overall)

Summary [edit]

Description:	English. The Hamilton Montage, performed live 12 May 2009 at the White House by Lin-Manuel Miranda (with Alex Lacamoire).
Date:	1 December 2015, 02:20:08
Source:	http://www.whitehouse.gov/photos-and-videos/video/lin-manuel-miranda-performs-white-house-poetry-jam-8-8 of White House copyright notice (http://www.whitehouse.gov/copyright/) allows free use under a Creative Commons Attribution 3.0 License .
Author:	Lin-Manuel Miranda (uploaded by Gobbit3)
Camera location:	38° 52' 51.72" N , 77° 02' 11.47" W View this and other nearby images on: OpenStreetMap - Google Earth



Size of this resource: 400 x 160 pixels. Other resolutions: 500 x 160 pixels, 640 x 160 pixels, 1,024 x 160 pixels, 1,280 x 160 pixels.
Original file (H.264 authorized file, VP8/VP9, length 1 min 42 s, 1,600 x 1,080 pixels, 6.17 Mbps overall)

Summary [edit]

Description:	English. The Optical Payload for Lasercomm Science will beam video via laser from the International Space Station back to Earth. Here is animation showing how the technology works, with an explanation from the OPALS Mission Manager, Matt Abrahamson of JPL, plus the video NASA stated for OPALS' first official transmission.
Date:	6 June 2014
Source:	http://www.youtube.com/watch?v=1abMTCnCR0k
Author:	NASA/Jet Propulsion Laboratory

Even when we get high-definition files, they usually have to be transcoded into Ogg or WebM from already-compressed sources, introducing some generational loss.

WIKIPEDIA AS ARCHIVE

- Long-term **preservation**
 - Source videos may be long, awkward to use directly
 - Need to be able to break down and remix

Of course when you're trying to preserve data as an archive, you may have long, awkward files that take up lots of space. So we need to be able to take those big source videos and break them down into trimmed clips that are useful in an encyclopedia, a dictionary, a news site, etc. We'll get to more of that later!

WE ARE NOT ALONE



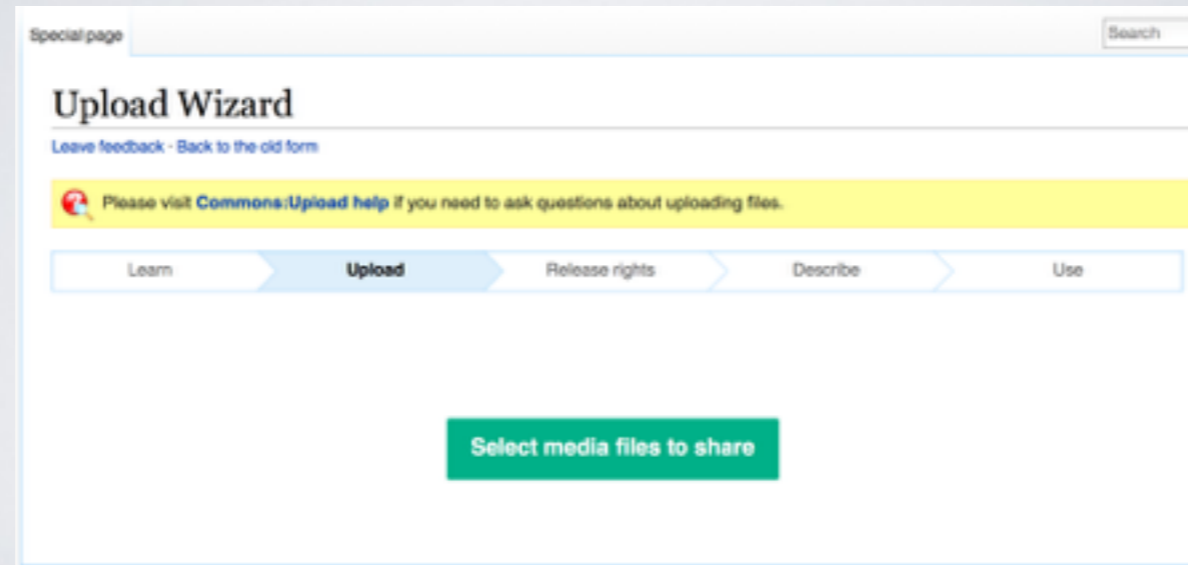
Luckily, we are not alone. The Internet Archive has been doing online video archives for some time, and for better or worse they've avoided falling into paralysis over formats and licenses. This makes them a great data source and potential partner if we can get a bridge hooked up from their system to ours!

VIDEO CONCERNS

- Playback
- Archiving
- **Ingest**
- Creation

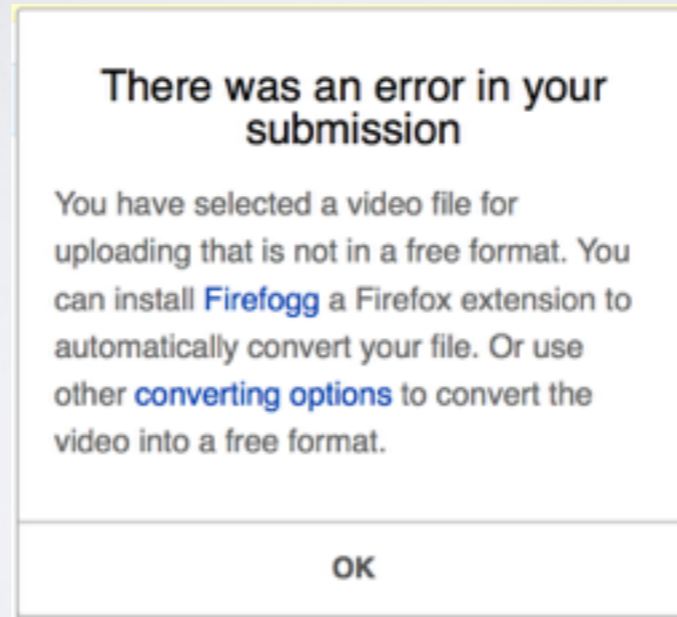


WEB UPLOAD



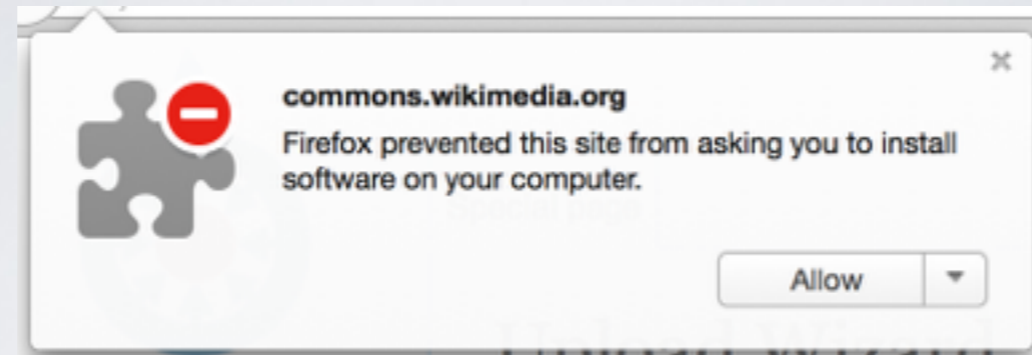
Uploading a few files to Commons isn't **too** awful, but for media files you currently have to convert them into either Ogg or WebM manually before upload. This introduces some data loss in the recompression, but more importantly it's a huge barrier step for potential contributors.

BACK TO THE FORMAT WARS



Most consumer & prosumer cameras these days natively record to some MP4/H.264 flavor; that's also the most common output format from desktop video editing workflows. To avoid alienating contributors, we need to either eliminate that transcode step, or automate it more reliably.

BACK TO THE FORMAT WARS

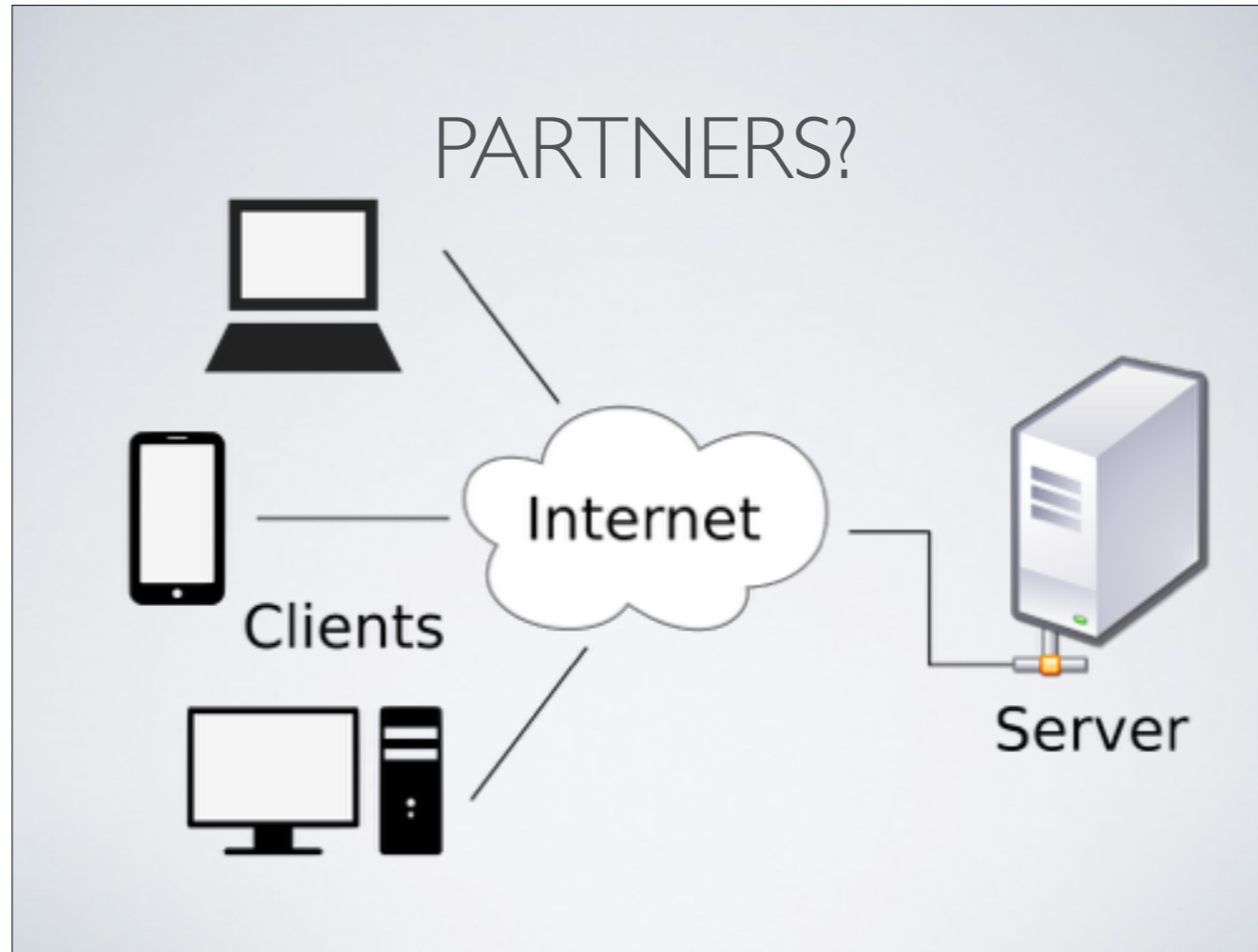


For instance if you click that “Install Firefogg” link, Firefox comes up with a big security warning. And if you’re not using Firefox, you’re thrown to the wolves with a list of ffmpeg command lines!

LARGE & BULK UPLOADS



Uploads of large files remains a bit buggy, and we have a hard limit of a few hundred megabytes. Uploading dozens or hundreds of large files? You may be on your own figuring out how to transcode them, **and** will then have to arrange some awkward side-loading of the files either over the internet or by mailing a hard drive.



If the Wikimedia community continues to reject even server-side import of MP4-based files, we'll have to either arrange some sort of partnership with another organization to do transcoding or figure out how to build a transcode+upload manager program that folks can run on a desktop.

LEVERAGE THE DESKTOP



Since most desktop operating systems include MP4 support in the system, we can pull the same trick that Mozilla does with Firefox and use the system's native decoder, letting us distribute the rest of the program freely without worrying about licensing. But, as an organization we have little experience with building desktop software.

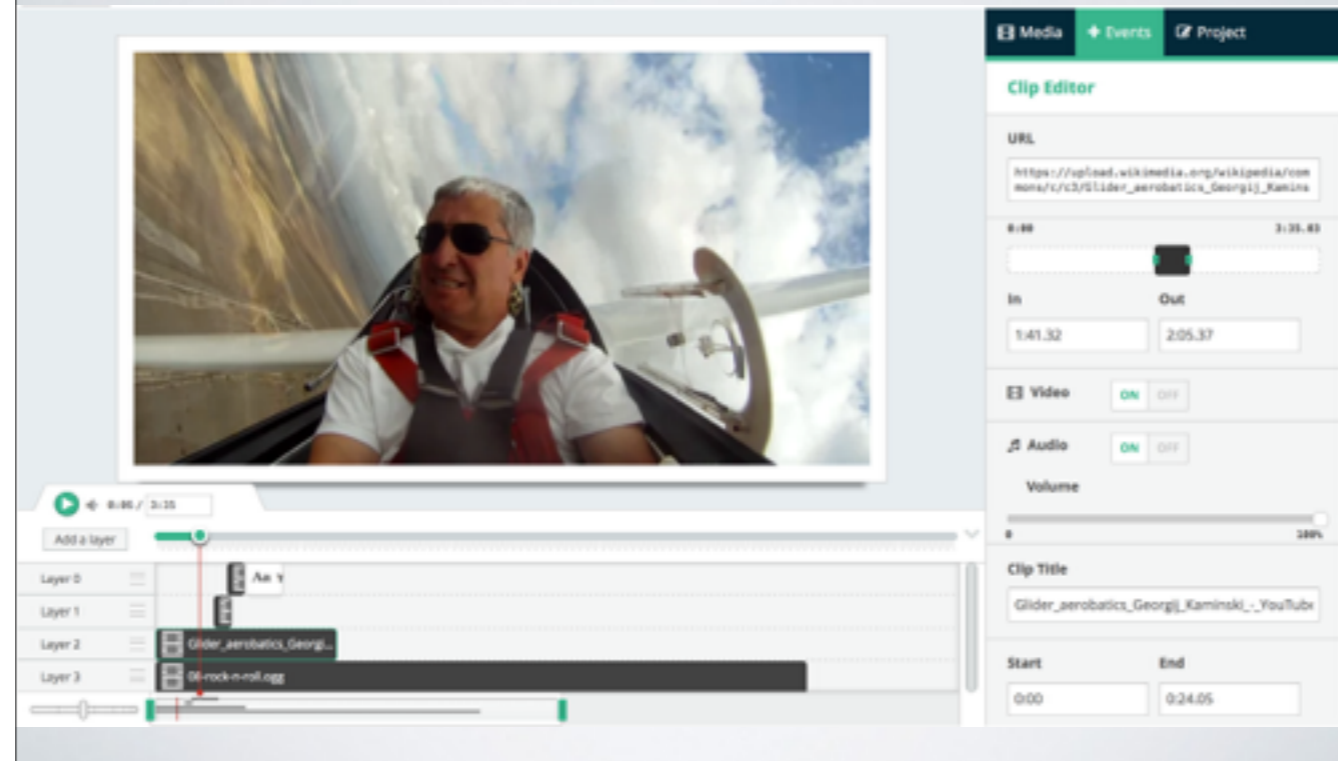
VIDEO CONCERNS

- Playback
- Archiving
- Ingest
- **Creation**



Wikimedia Commons is an archive of sorts, but Wikipedia and its other sister sites have more direct goals of reference & education. We need to be able to take source videos and trim, edit, and remix them into clips that are useful in that context.

MOZILLA'S POPCORN MAKER



But we don't have to invent the wheel... One of the projects I was really excited to see come out of Mozilla's "Webmaker" initiative was "Popcorn Maker", an in-browser video editor that let you assemble clips and do basic non-linear editor operations.



Hello there!

You were looking for something, but, unfortunately, it's no longer here.

To keep our tools and curriculum as relevant as possible, the Webmaker team has deprecated some services. Most of these services are being replaced by new and improved ones, but a few of them are not. You can read more about this on [our blog](#).

Need help, or have questions or comments about our processes? Send us some feedback [here!](#)

Looking for something to do? Why not checkout our new [Webmaker Android app?](#)

Unfortunately Mozilla has recently discontinued Popcorn Maker as the Webmaker team consolidates resources, and it's vanished from the web.

This repository Search Pull requests Issues Gist

mozilla / popcorn-editor Unwatch 30 Unstar 21 Fork 4

Fork of Webmaker's Popcorn Editor for local use

60 commits 3 branches 1 release 2 contributors

Branch: master popcorn-editor / +

Merge pull request #12 from mozilla/travis-ci-deploy

Nolski authored on Aug 17 Latest commit #22b6ba5dc

PopcornEditor	Dynamically set videoUrl key	2 months ago
.gitignore	need to dist css for bower	3 months ago
.travis.yml	set up auto-deploy to s3	2 months ago
README.md	Added travis badge	2 months ago
bower.json	Updated to v 0.0.1	3 months ago
gulpfile.js	added travis	2 months ago
index.html	Refactored createTemplate to just return the template	3 months ago
package.json	add gulp to project dependencies	2 months ago

Code

- Issues 27
- Pull requests 0
- Wiki
- Pulse
- Graphs

SSH clone URL

git@github.com:mozilla/popcorn-editor

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

Fortunately it's open source! The core in-browser editor component has been made to work standalone, and we'll be working with some of the fine folks from Mozilla to integrate it into MediaWiki. If you're interested, please join us during the unconference sessions tomorrow!

(POPCORN EDITOR DEMO)

Attempt demo if wifi permits. If following at home, try out <http://popcorn-editor.wmflabs.org/> !

SCHNITTSERVER



There's also been some great work done on an editing server with some German & Swiss funding, known informally as the "Schnittserver". This is experimentally hosted by Internet Archive and, though the user interface is very bare, can connect to Commons to upload output files in the formats that Wikimedia wants.



In addition to format transcoding, the server supports sharing of video editing projects in the 'MLT' format used by Kdenlive (on Linux) and Shotcut (cross-platform). This allows one user to upload source files - in whatever format - while others work with the editing project using freely-distributable WebM 'proxy clips'.

File: Lone Star Geyser

Author	Dschwen
Keywords	
Date	2015-09-09
Language	English
License	Own work, copyleft, attribution required (Multi-license GFDL, CC-BY-SA all versions)
MIME Type	application/x-kdenlive
Size	13.6 KiB
Status	Uploaded => Validated => Rendered => Published

Other files used by this project

	Title	Author	
	Lone Star Geyser 1	Dschwen	Download
	Lone Star Geyser 2	Dschwen	Download
	Lone Star Geyser 3	Dschwen	Download

[Edit](#) | [Internet Archive](#) | [Download](#) | [Download Project XML](#)

[Upload New Version](#)

The final project can then be rendered out from the original files — minimizing generational loss in transcoding — and uploaded directly to Commons, Internet Archive, etc.

WORDS WORDS WORDS



Editing TimedText:Folgers.ogv.de.srt

Adding [Universal Subtitles](#). Formatting template for the [SubRip](#) format:

```
1
00:00:20,000 --> 00:00:24,400
Text that should be displayed from second 20 to second 24

2
00:00:24,600 --> 00:00:27,800
Text that should be displayed from second 24 to second 27
```

B I > [Advanced](#) > [Special characters](#) > [Help](#) > [EditDoc](#)

```
0
00:00:01,707 --> 00:00:04,154
Harold schmeckt der Kaffee?

1
00:00:04,169 --> 00:00:05,185
uh-uhmm

2
00:00:05,005 --> 00:00:07,785
Du meinst, er ist so schlecht wie gestern?

3
00:00:07,785 --> 00:00:08,962
uh-huh
```

Localization is also a key concern for Wikimedia projects. We already support subtitles on-wiki, but they're very awkward to edit. Better tools are needed.

Questions?

<https://brionv.com/misc/video-wikiconusa-2015.pdf>

Copy of slides at <https://brionv.com/misc/video-wikiconusa-2015.pdf> for your convenience. Would put on Commons but have to track down URLs for all the pictures I included first. :D