Video editing in the browser

- Yash Khandelwal (Software developer, Microsoft)
- Steve Becker (Principal Software Eng Manager, Microsoft)
- Harneet Sidhana (Principal Program Manager, Microsoft)

Topics:

- Current workarounds/issues
- MediaBlob
- WebCodecs

Video editing on the web is painful

Client-side workarounds

Client-side

JavaScript/WASM libraries

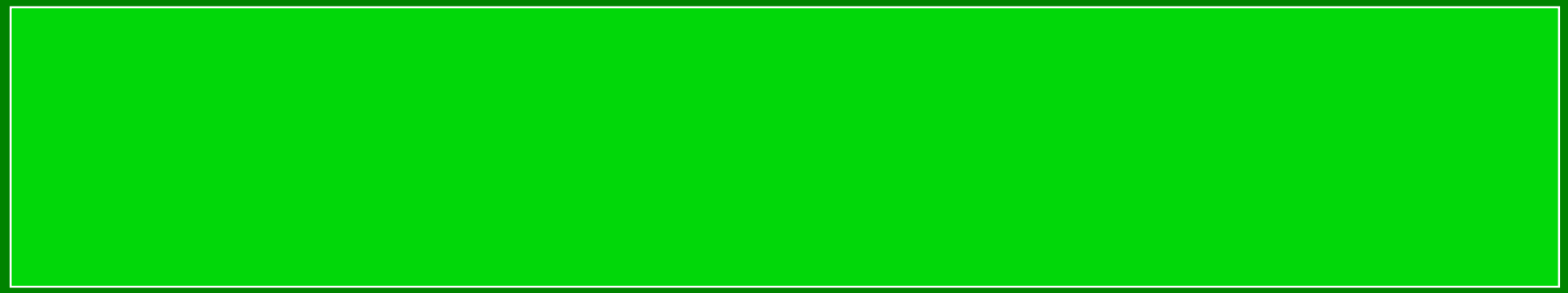
- Usually in several MBs
- Affects bandwidth
- May not be optimized

Client-side

Using MediaRecorder

Client-side

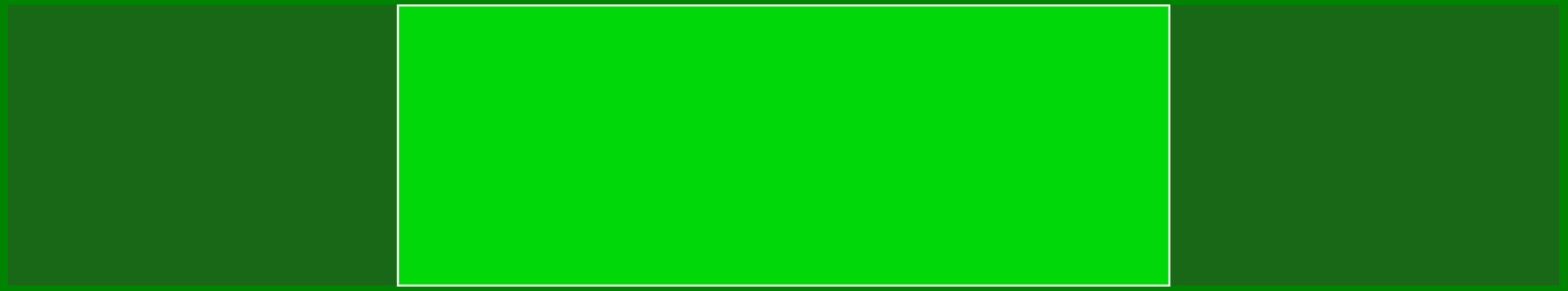
1x playback



30 minute video

Client-side

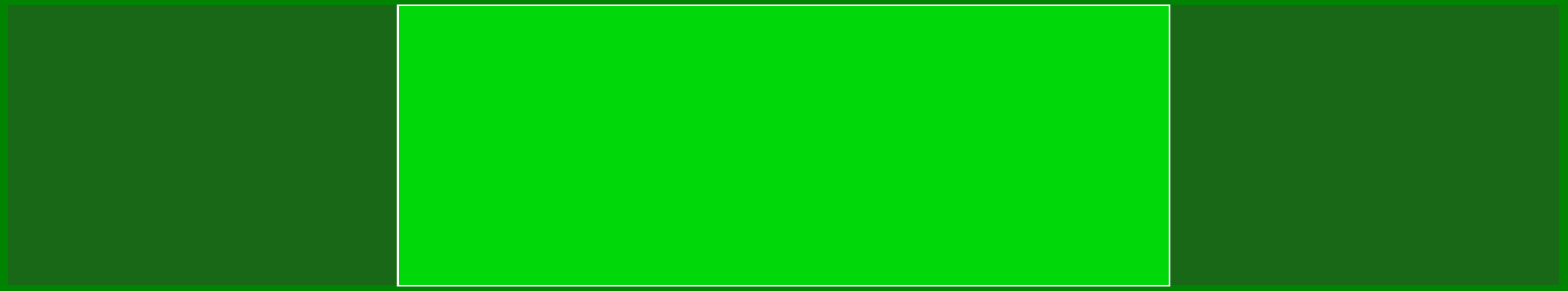
1x playback



Trim by 15 minutes

Client-side

1x playback



Takes 15 minutes

Server-side issues

Server-side

Increased bandwidth

Upload -> Edit -> Download

Server-side

Compute cost

Dedicated servers for doing the video edits

Server-side

Queue Delay

Bottlenecks on the server resulting in an editing queue

Studio beta

Example of NFL.com Gamecenter

UNDO REDO

DISCARD CHANGES SAVE

Video

Details

Analytics

Editor

Comments

Transcriptions

Other features

03:17 EDIT TI

45:00 56:08

ADD AN END SCREEN

ADD BLUR

Save changes?

It might take a few hours for your changes to apply. During that time:

- Viewers will see the current version of the video
- You won't be able to make other changes to the video

While you wait, you can leave this site.

CANCEL SAVE

MediaBlob

High level video editing API that inherits from Blob

- new MediaBlob(blob); // Mime sniffing to detect whether the blob is a valid media object or not.

MediaBlob properties:

MediaBlob.duration

readonly attribute long long duration;

Example:

```
let mediaBlob = new MediaBlob(blob);
console.log(mediaBlob.duration)  // print the duration in ms
```

Proposed editing operations (MediaBlobOperation):

trim

```
void trim(long long startTime, long long endTime);
```

concat

```
void concat(<Sequence<MediaBlob>);
```

split

```
void split(long long time);
```

finalize

```
Promise<Sequence<MediaBlob>> finalize(optional DOMString mimeType);
```

Example:

```
let mbo = new MediaBlobOperation(new MediaBlob(blob1));
mbo.trim(4000, 10000);
mbo.concat(new MediaBlob(blob2));
mbo.finalize().then(function(mediaBlobs) {
    // mediaBlobs[0] will be a concatenated MediaBlob of blob1 (which will be trimmed) and blob2
});
```

728ms

Time taken to trim a 30 min video to 15 mins using MediaBlob

- Easy to use with existing MediaRecorder and File API
- No detailed knowledge of media concepts needed by developers
- Developers need not worry about codecs licensing
- Provides hardware optimizations when available

External libraries
1x record time
Queue Delay
Increased bandwidth
Compute Cost

~~External libraries~~

~~1x record time~~

~~Queue Delay~~

~~Increased Bandwidth~~

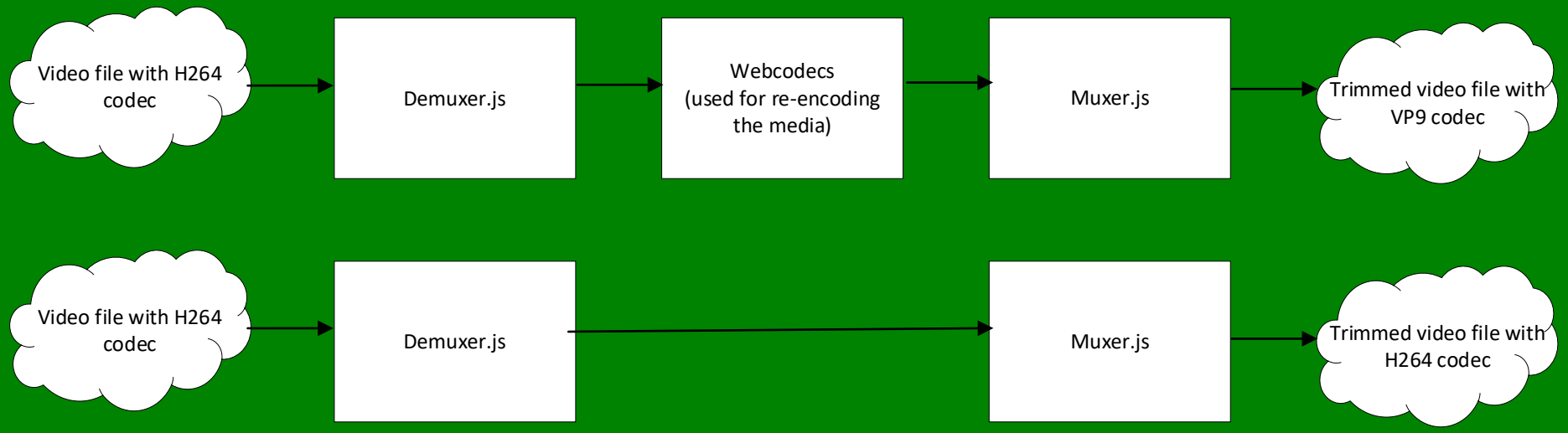
~~Compute Cost~~

WebCodecs

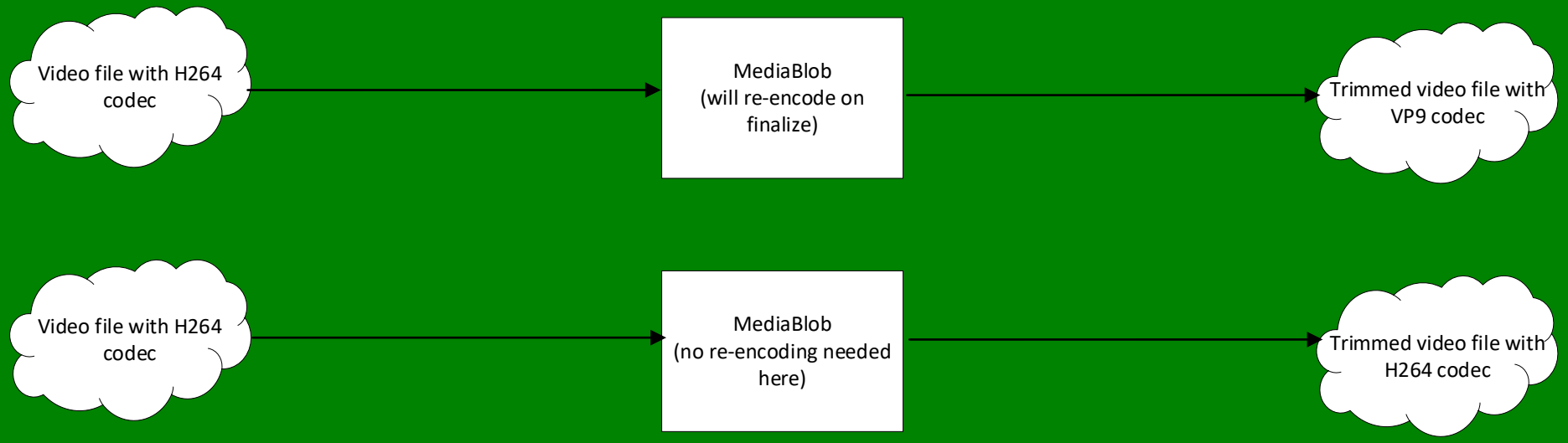
Low level API that allows web apps to encode and decode media

- Powerful API that provides support for live gaming, real time communication among other things
- For video editing, it provides transcoding
- Does not provide a way to demux/mux media file. Must rely on external demuxers/muxers

With WebCodecs



With MediaBlob



Current status:

- MediaBlob exists on the Edge browser and can be used with Origin Trials
- Intent to prototype has been posted to chromium blink-dev
  - Ongoing discussion about whether we need MediaBlob when WebCodecs is already being implemented

Thank you !

Links:

WebCodecs: https://github.com/WICG/web-codecs/blob/master/explainer.md

MediaBlob: https://github.com/WICG/video-editing/blob/master/readme.md

Intent to prototype: https://groups.google.com/a/chromium.org/forum/#!topic/blink-dev/3eac-HVygFY

Update 7/6/20

- After discussion on the intent-to-prototype, it was concluded that the JavaScript implementation of muxer is efficient.

- Based on the feedback from the group here and on the explainer, we are looking into how to adapt MediaBlob to a “playlist” model.