

# Http Live Streaming

Gary Chen 2014/04/08

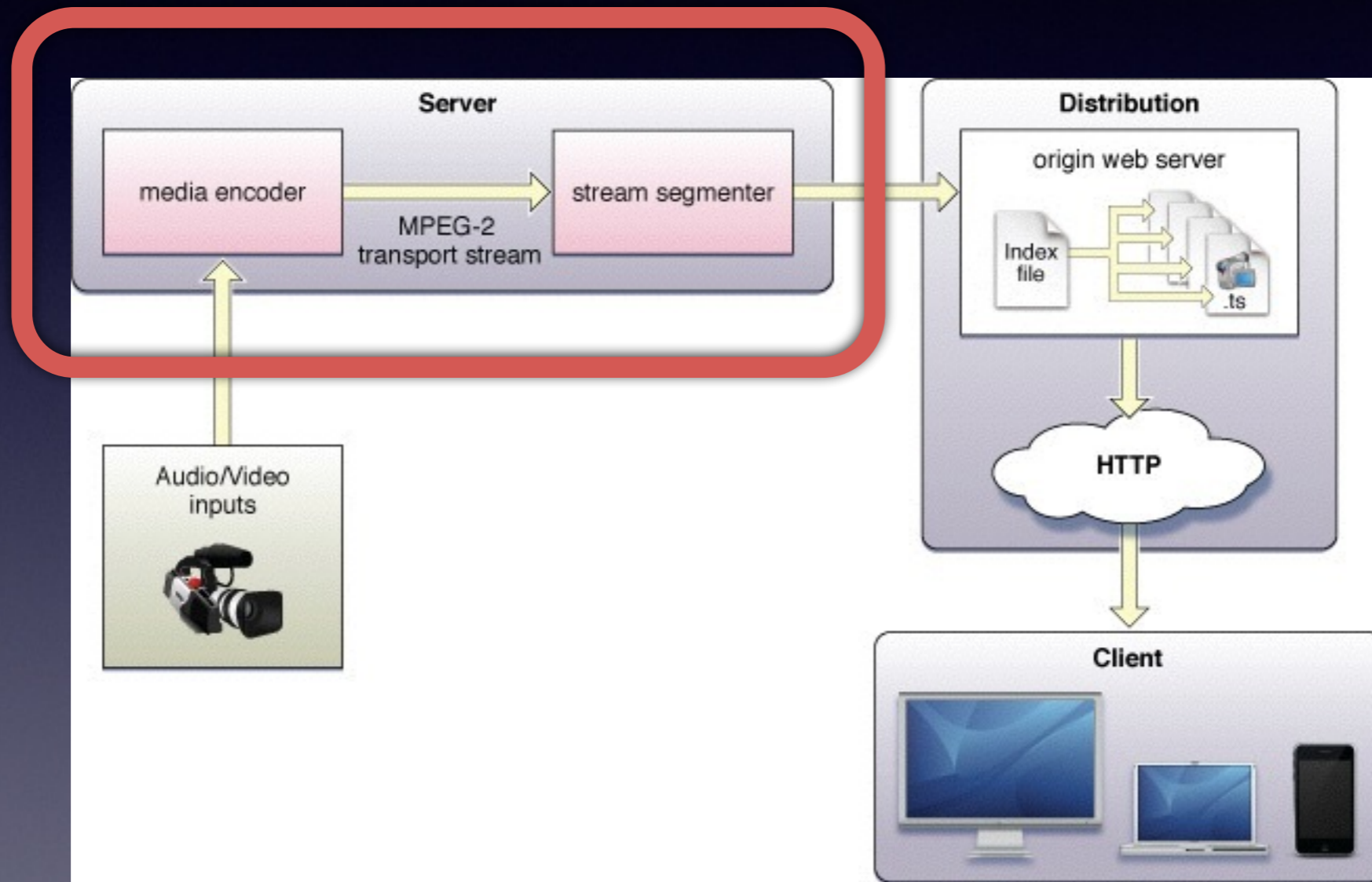
# Outline

- HLS Overview
- m3u8 format
- WebVTT
- Media Source Web API
- How to implement?

# What is HLS?

- HTTP Live Streaming allows you to send/receive **live** or **prerecorded** **<audio>** and **<video>**.
- HLS supports multiple alternate streams at **different bit rates**, and the client software can switch streams intelligently as **network bandwidth changes**.

# HLS Architecture

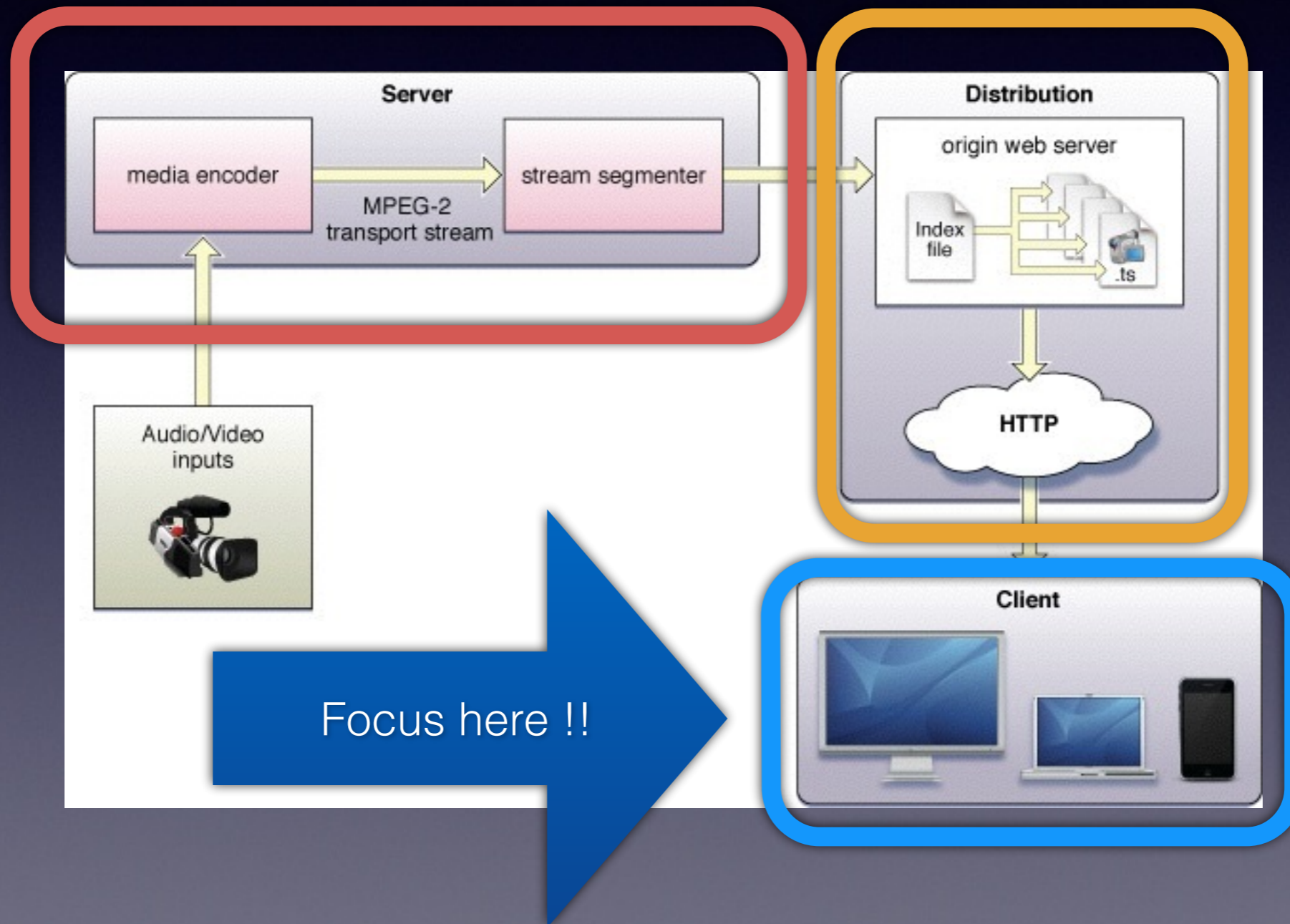




# MPEG2-TS

- MPEG transport stream (MPEG-TS, MTS or TS) is a standard format for transmission and storage of **audio, video, and Program and System Information Protocol (PSIP)** data.
- Transport Stream is specified in **MPEG-2 Part 1**.

# HLS Architecture



# Client Software

Responsible for **determining the appropriate media** to request, downloading those resources, and then **reassembling** them so that the media can be presented to the user in a continuous stream.



# What is m3u8?

- The Unicode version of "m3u" is "m3u8", which uses **UTF-8** Unicode characters.

```
#EXTM3U
```

```
#EXT-X-PLAYLIST-TYPE:VOD
```

```
#EXT-X-TARGETDURATION:10
```

```
#EXT-X-VERSION:3
```

```
#EXT-X-MEDIA-SEQUENCE:0
```

```
#EXTINF:10.0, title
```

```
http://example.com/movie1/fileSequenceA.ts
```

```
#EXTINF:10.0, title
```

```
http://example.com/movie1/fileSequenceB.ts
```

```
#EXTINF:10.0, title
```

```
http://example.com/movie1/fileSequenceC.ts
```

```
#EXT-X-ENDLIST
```



# VOD and Event(Live Playlist)

#EXTM3U

#EXT-X-PLAYLIST-TYPE:**VOD**

#EXT-X-TARGETDURATION:10

#EXT-X-VERSION:3

#EXT-X-MEDIA-SEQUENCE:0

#EXTINF:10.0, title

<http://example.com/movie1/fileSequenceA.ts>

#EXTINF:10.0, title

<http://example.com/movie1/fileSequenceB.ts>

#EXTINF:10.0, title

<http://example.com/movie1/fileSequenceC.ts>

#EXT-X-ENDLIST

#EXTM3U

#EXT-X-PLAYLIST-TYPE:**EVENT**

#EXT-X-TARGETDURATION:10

#EXT-X-VERSION:3

#EXT-X-MEDIA-SEQUENCE:0

#EXTINF:10.0, title

<http://example.com/movie1/fileSequence0.ts>

#EXTINF:10.0, title

<http://example.com/movie1/fileSequence1.ts>

.

.

.

# Event(Live Playlist)

#EXTM3U

#EXT-X-PLAYLIST-TYPE:EVENT

#EXT-X-TARGETDURATION:10

#EXT-X-VERSION:3

#EXT-X-MEDIA-SEQUENCE:0

#EXTINF:10.0, title

<http://example.com/movie1/fileSequence0.ts>

#EXTINF:10.0, title

<http://example.com/movie1/fileSequence1.ts>

.  
. .  
. .

#EXTM3U

#EXT-X-PLAYLIST-TYPE:EVENT

#EXT-X-TARGETDURATION:10

#EXT-X-VERSION:3

#EXT-X-MEDIA-SEQUENCE:1

#EXTINF:10.0, title

<http://example.com/movie1/fileSequence1.ts>

#EXTINF:10.0, title

<http://example.com/movie1/fileSequence2.ts>

.  
. .  
. .

#EXT-X-ENDLIST

# Basic Variant Playlist

#EXTM3U

#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=150000,RESOLUTION=416x234, \

CODECS="avc1.42e00a,mp4a.40.2"

<http://example.com/low/index.m3u8>

#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=240000,RESOLUTION=416x234, \

CODECS="avc1.42e00a,mp4a.40.2"

[http://example.com/lo\\_mid/index.m3u8](http://example.com/lo_mid/index.m3u8)

#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=440000,RESOLUTION=416x234, \

CODECS="avc1.42e00a,mp4a.40.2"

[http://example.com/hi\\_mid/index.m3u8](http://example.com/hi_mid/index.m3u8)



# Encryption Keys

#EXTM3U

#EXT-X-TARGETDURATION:10

#EXT-X-VERSION:3

#EXT-X-MEDIA-SEQUENCE:0

#EXT-X-KEY:METHOD=AES-128,URI="https://priv.example.com/key.php?r=52"

#EXTINF:10.0,

movieA.ts

#EXT-X-KEY:METHOD=AES-128,URI="https://priv.example.com/key.php?r=53"

#EXTINF:10.0,

movieB.ts



# Closed Captions

```
#EXTM3U
#EXT-X-MEDIA:TYPE=CLOSED-CAPTIONS,GROUP-
ID="cc",NAME="CC1",LANGUAGE="en",DEFAULT=YES,AUTOSELECT=YES,INSTREAM-ID="CC1"

#EXT-X-MEDIA:TYPE=CLOSED-CAPTIONS,GROUP-
ID="cc",NAME="CC2",LANGUAGE="sp",AUTOSELECT=YES,INSTREAM-ID="CC2"

#EXT-X-STREAM-INF:BANDWIDTH=1000000,SUBTITLES="subs",CLOSED-CAPTIONS="cc"
closedCaptions.m3u8
```

**Support WebVTT**

# WebVTT

00:11.000 --> 00:13.000

<v Roger Bingham>We are in New York City

00:13.000 --> 00:16.000

<v Roger Bingham>We're actually at the Lucern Hotel, just down the street

00:16.000 --> 00:18.000

<v Roger Bingham>from the American Museum of Natural History

```
<video width="640" height="480" controls>  
  <source src="video.webm" type="video/webm" />  
  <track src="subtitles.vtt" kind="subtitles" srclang="en" label="English" />  
</video>
```

# What is Media Source Extension?

- W3C Candidate Recommendation (CR)
- This specification extends `HTMLMediaElement` to allow JavaScript to generate media streams for playback. Allowing JavaScript to generate streams facilitates a variety of use cases like `adaptive streaming` and `time shifting live streams`.



# Introduction to MSE

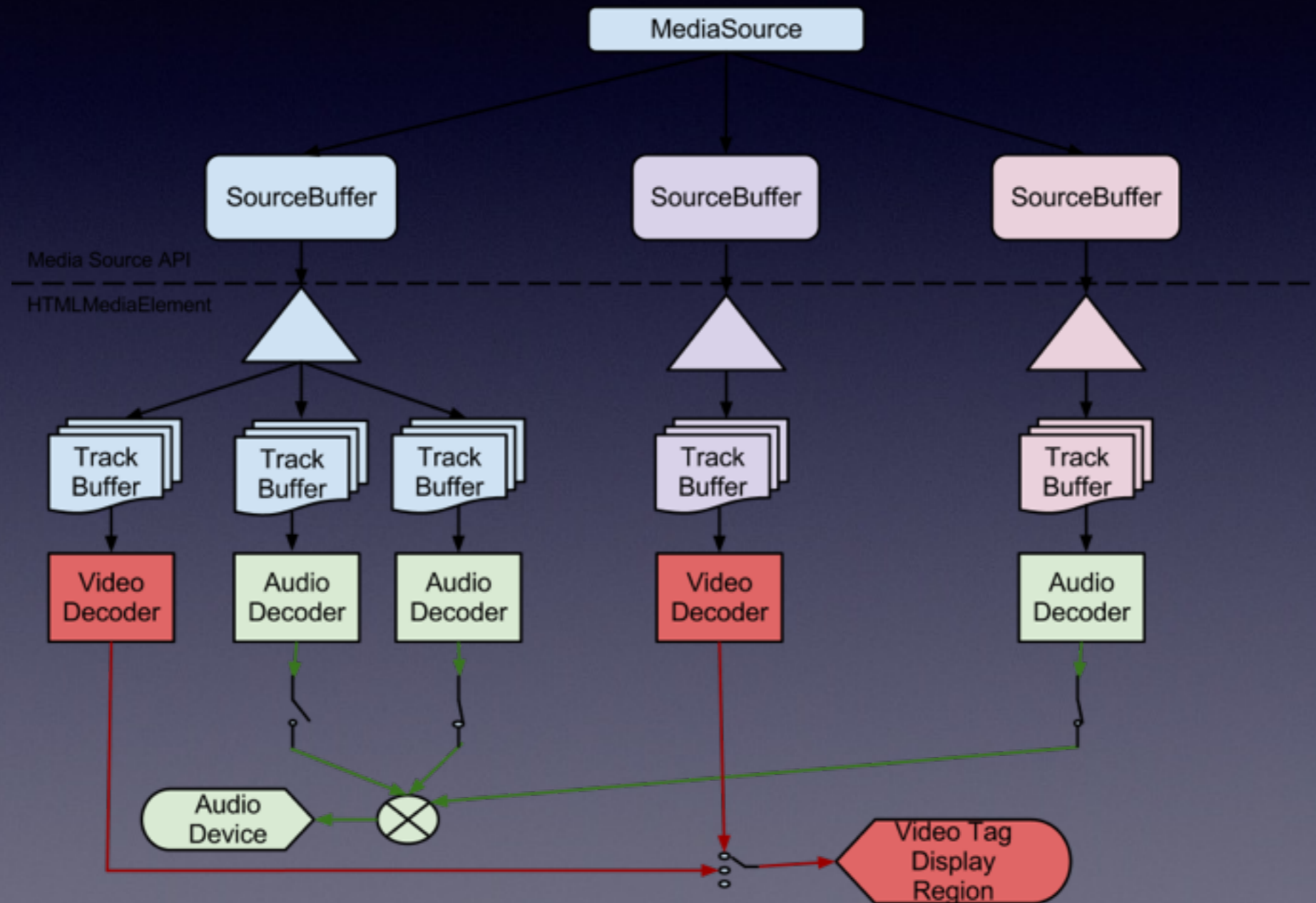
- This specification allows JavaScript to dynamically construct media streams for `<audio>` and `<video>`.
- Defines objects that allow JavaScript to pass media segments to an HTMLMediaElement [HTML5]. A **buffering model** is also included to describe how the user agent acts when different media segments are appended **at different times**.



# MSE Byte Stream Format Registry

<b>MIME type/subtype</b>	<b>Public Specification(s)</b>
audio/webm video/webm	<a href="#">WebM Byte Stream Format</a>
audio/mp4 video/mp4	<a href="#">ISO BMFF Byte Stream Format</a>
audio/mp2t video/mp2t	<a href="#">MPEG-2 Transport Streams Byte Stream Format</a>

# MSE Diagram



# MSE Sample Code

```
2 var ms = new MediaSource();
3 var video = document.querySelector('video');
4 video.src = window.URL.createObjectURL(ms);
5 // get source
6 var VideoWebMChunks = getSource();
7 ms.addEventListener('sourceopen', function(e) {
8     // assign mime type
9     var sourceBuffer = ms.addSourceBuffer('video/webm; codecs="vorbis, vp8"');
10
11     VideoWebMChunks.forEach(function(oneVideoWebMChunk, index, array){
12         // console.log("add VideoWebMChunks[" + index + "] into Video buffer");
13         sourceBuffer.appendBuffer(oneVideoWebMChunk);
14         if (index == (VideoWebMChunks.length-1) ) {
15             // endOfStream
16             ms.endOfStream();
17         }
18     });
19 }, false);
20
```



# Browser compatibility

	Desktop	Mobile			
Feature	Chrome	Firefox (Gecko)	Internet Explorer	Opera	Safari (WebKit)
Basic support	(Yes)	25.0 (25.0) [1]	Not supported	Not supported	Not supported

[1] Available after switching the `about:config` preference `media.mediasource.enabled` to true.

[2].IE11 introduces support for MPEG-DASH media streaming through (MSE).

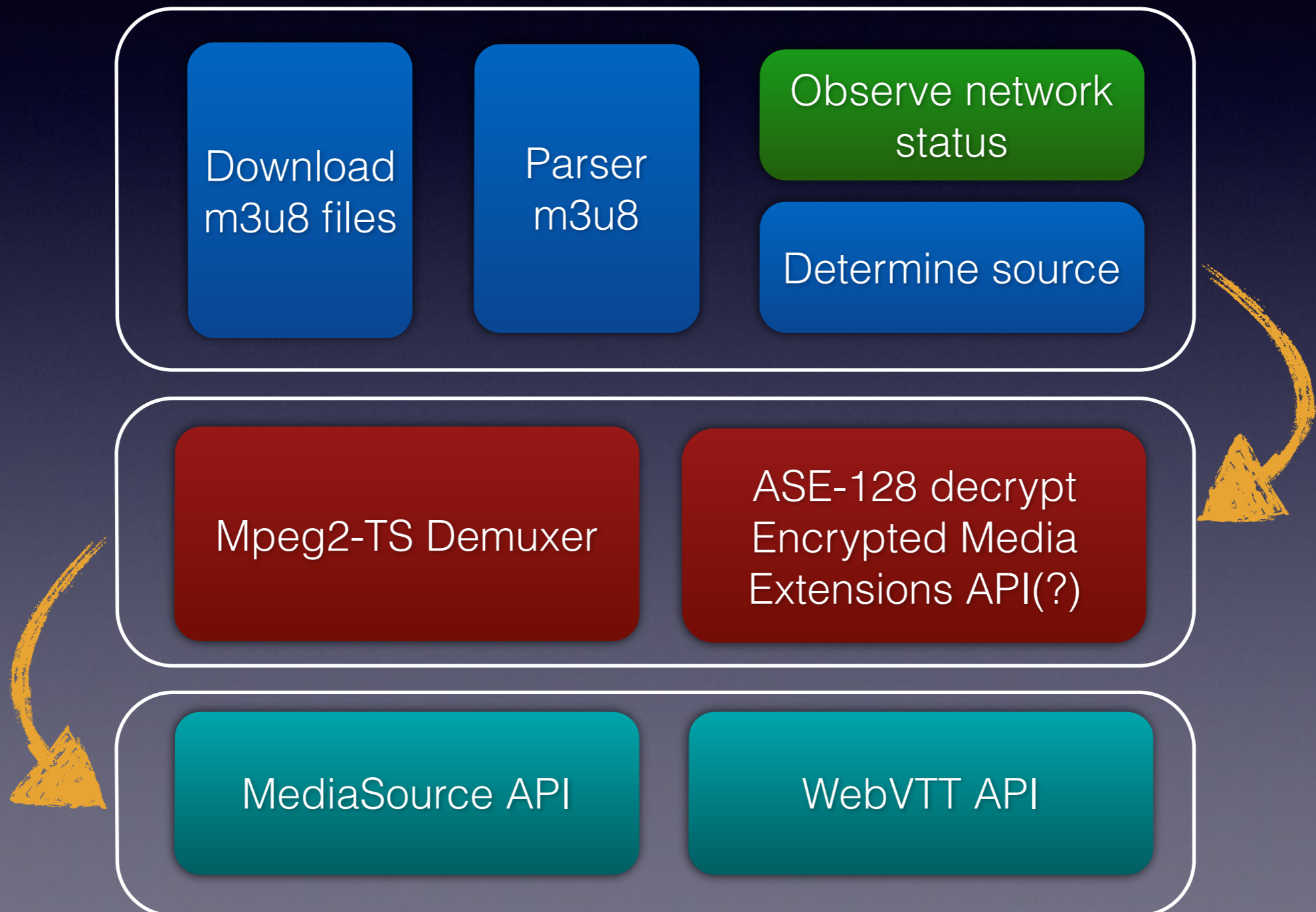
<http://msdn.microsoft.com/en-us/library/ie/bg182646%28v=vs.85%29.aspx>



# How does Safari implement?

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4   <title>Basic Stream - HTTP Live Streaming Examples - Apple Developer</title>
5   <meta name="omni_page" content="HTTP Live Streaming Examples - Basic Stream" />
6   <meta name="viewport" content="width=640; initial-scale=1.0; maximum-scale=1.0; user-
  scalable=0;" />
7 </head>
8 <body>
9   <video src="https://devimages.apple.com.edgekey.
  net/streaming/examples/bipbop_4x3/bipbop_4x3_variant.m3u8"
10   controls autoplay >
11   </video>
12 </body>
13 </html>
```

# How to implement?



# Bug 577084

- JS Demuxer  
<http://github.com/jDataView/jBinary>  
<http://rreverser.github.io/mpegts>
- <http://mae.localhost/mpegts/>



# Reference

- <https://developer.apple.com/>
- <http://tools.ietf.org/html/draft-pantos-http-live-streaming-08>
- <http://msdn.microsoft.com/en-us/library/ie/bg182646%28v=vs.85%29.aspx>
- <https://dvcs.w3.org/hg/html-media/raw-file/tip/media-source/media-source.html>
- <https://dvcs.w3.org/hg/html-media/raw-file/tip/media-source/byte-stream-format-registry.html>
- <https://developer.mozilla.org/en-US/docs/Web/API/MediaSource>
- <https://developer.mozilla.org/en-US/docs/HTML/WebVTT>
- [https://bugzilla.mozilla.org/show\\_bug.cgi?id=577084](https://bugzilla.mozilla.org/show_bug.cgi?id=577084)
- Blake