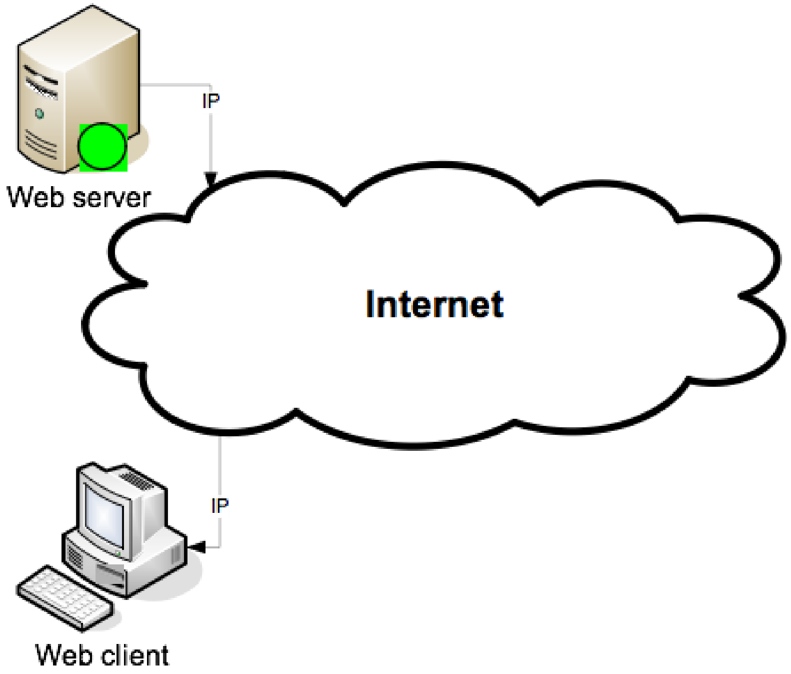
Connection-Oriented Streaming of Multimedia Content

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie
AGH University of Science and Technology

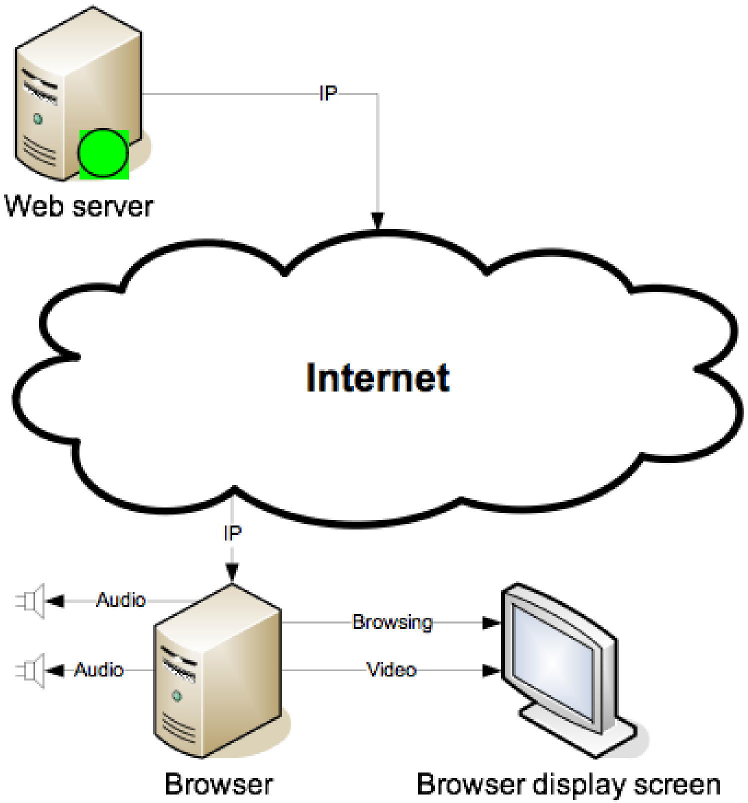
Mikołaj Leszczuk (AGH UST), Rafał Myszka (Akamai),
Jakub Nawała (AGH UST), Özlem Akkan (DEU)

Why Connectionless Streaming Is Bad?

- » Running out of **IP** addresses
- » Network Address Translation (**NAT**)
- » No **RTP/UDP** easily possible then… ☹
- » But **NAT** usually OK with **Web** traffic! ☺

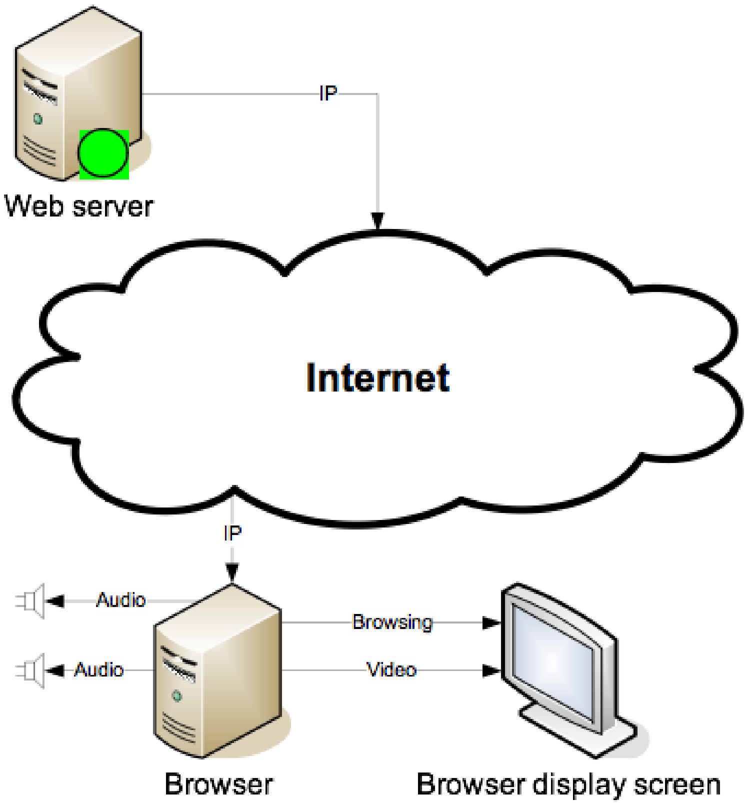


Why Connection-Oriented Streaming Is Good?



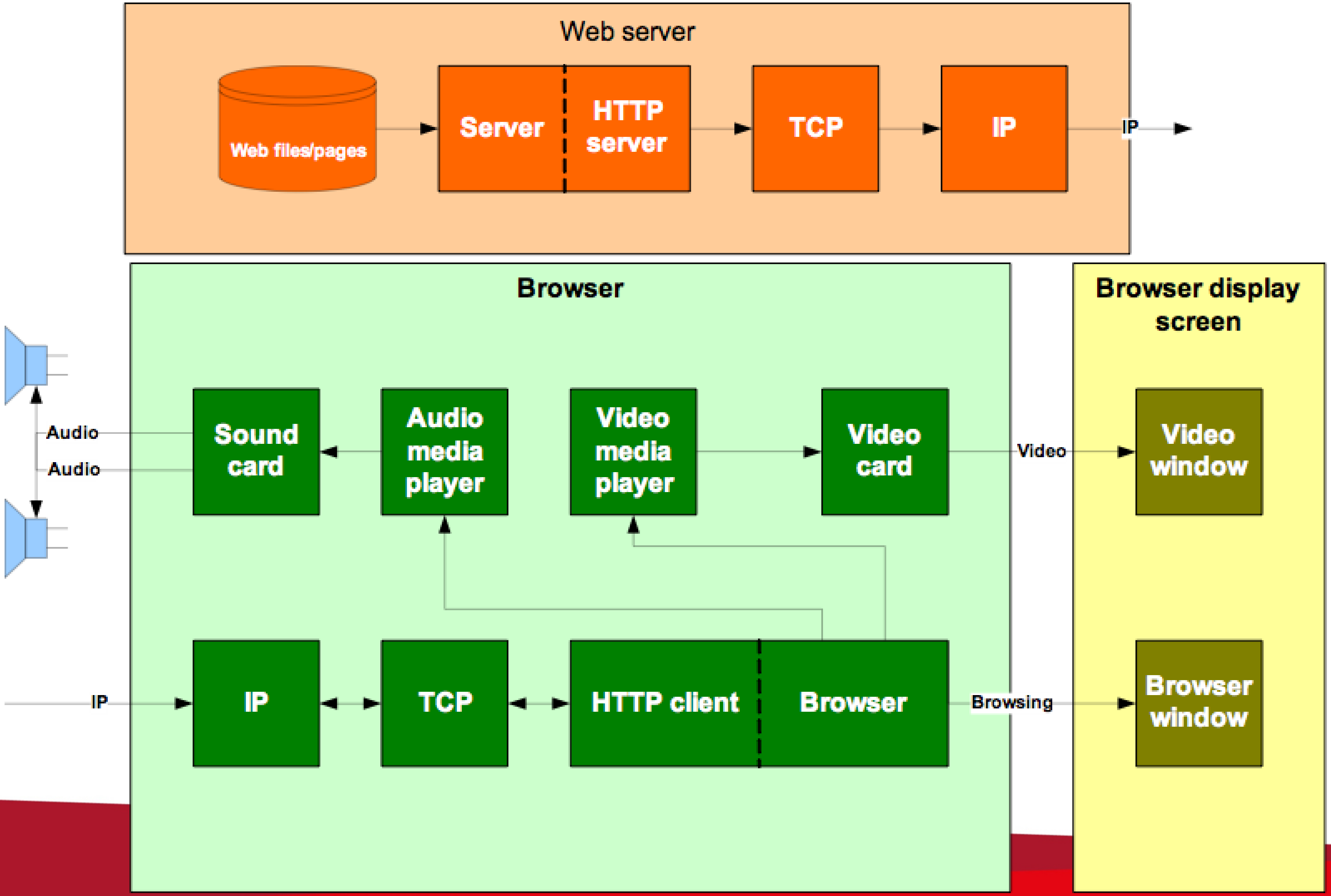
- **Web** means **HTTP/TCP/IP**
- Let’s encapsulate streaming as **Web**
- Progressive Media Download (**PMD**)
- Called as “streaming” even if technically not streaming
- What are technical details?

What is **PMD**?

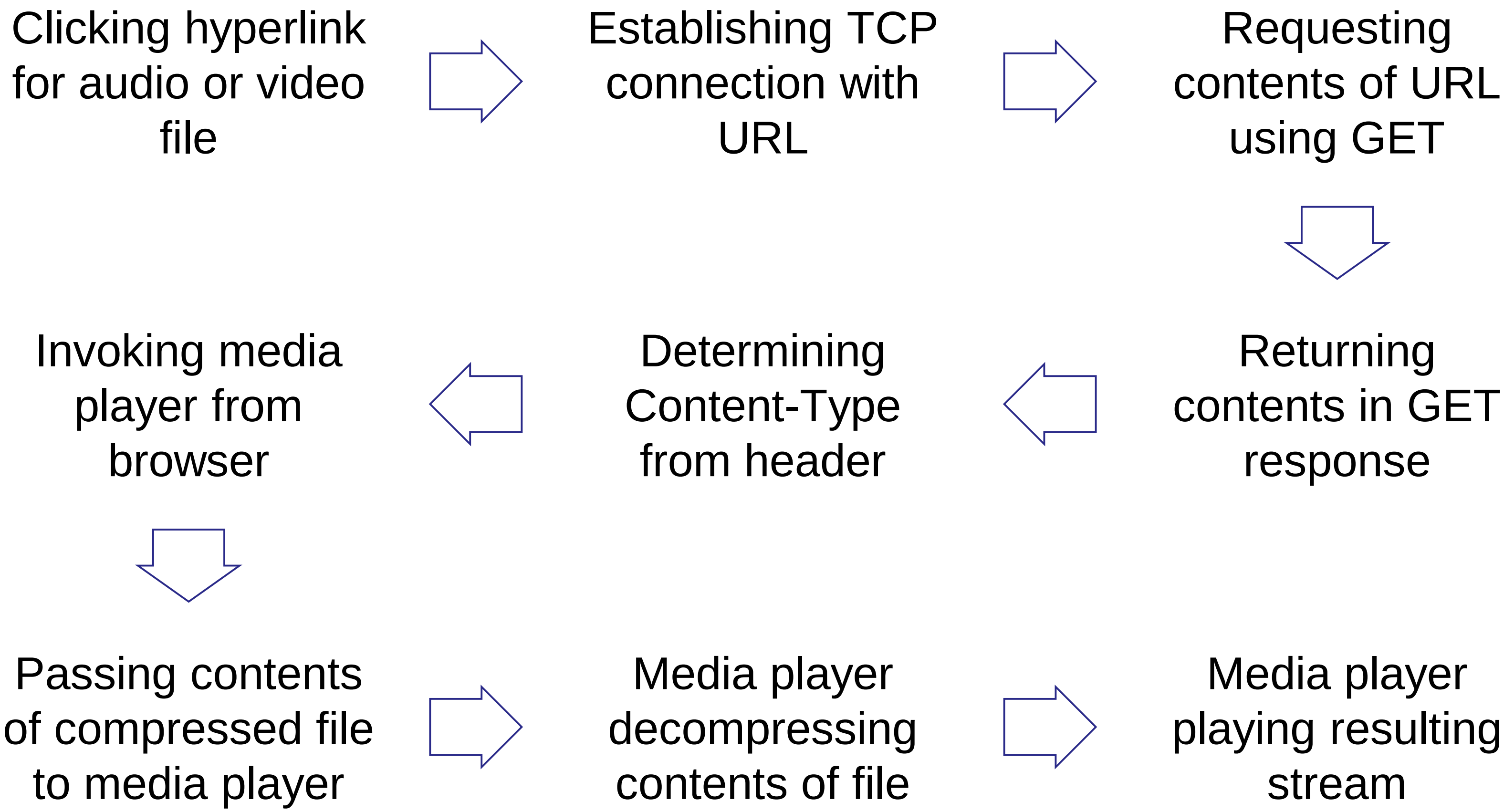


- » How do files are accessed?
- » Can plain **Web** server be used?
- » Is **A/V** accessed the same way?
- » how digital media data is received and stored

Components of **PMD**



HTTP Process for PMD



Clicking hyperlink for audio or video file

Establishing TCP connection with URL

Requesting contents of URL using GET

Returning contents in GET response

Determining Content-Type from header

Invoking media player from browser

Passing contents of compressed file to media player

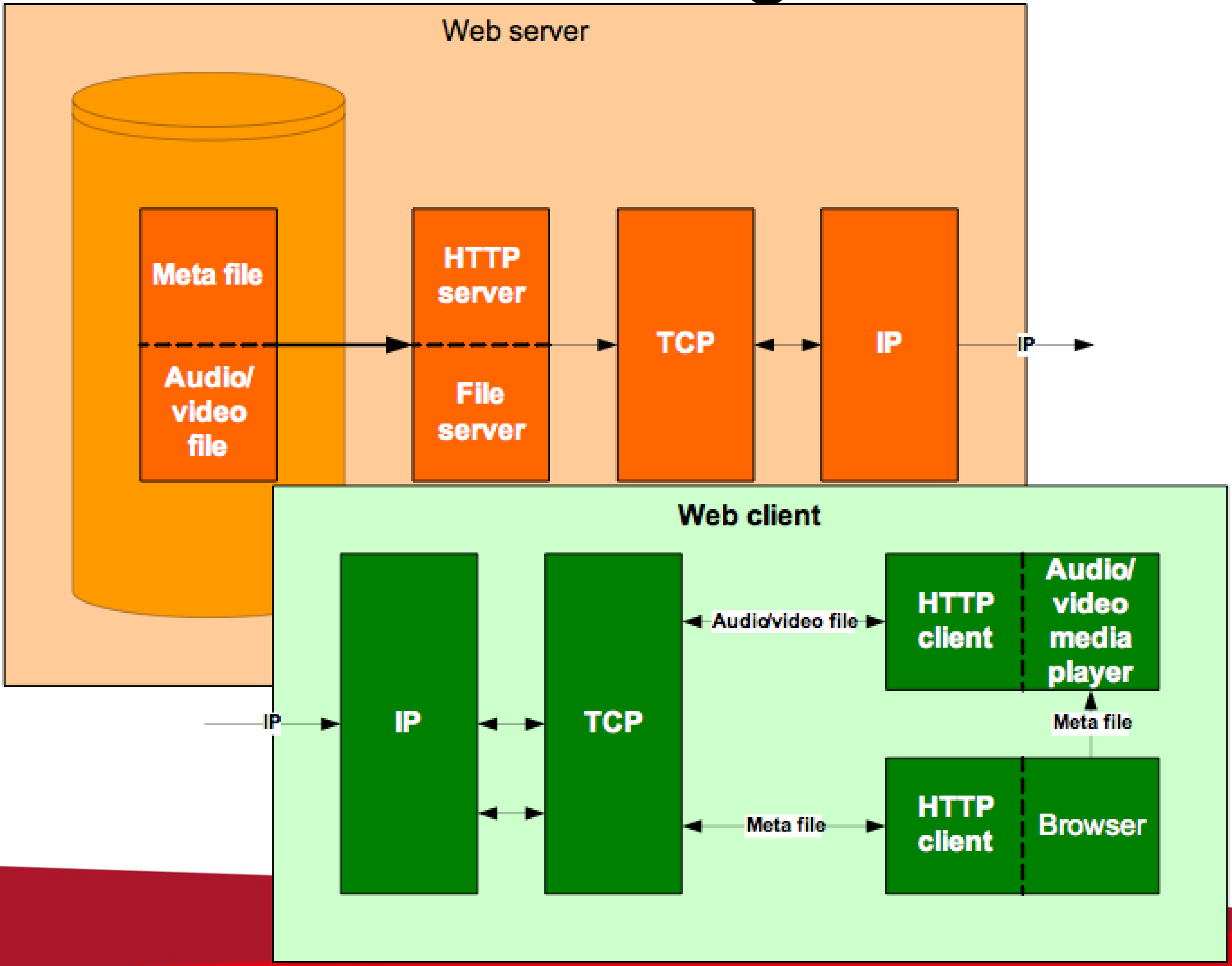
Media player decompressing contents of file

Media player playing resulting stream

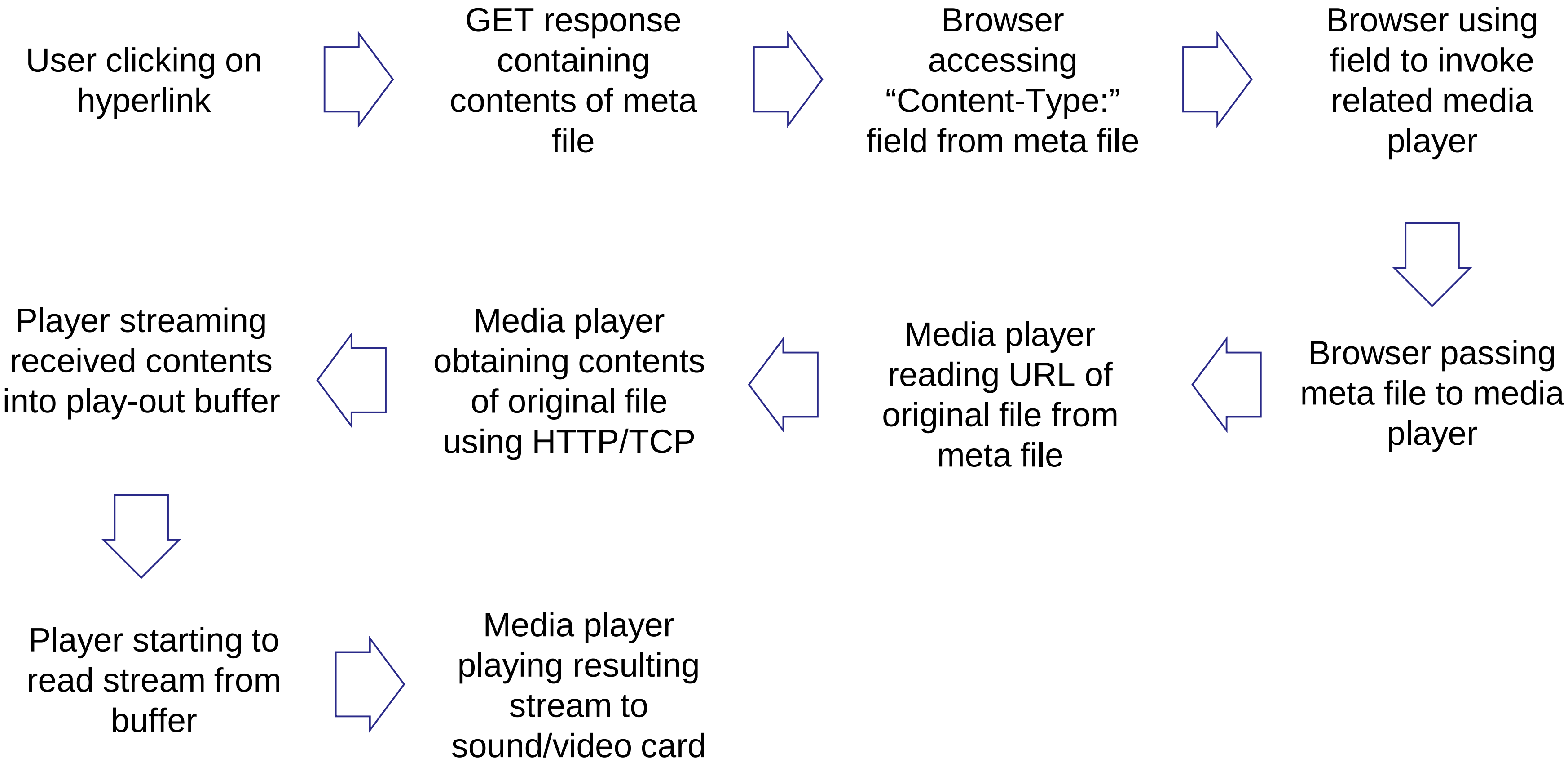
Problem. And Solution

- First: browser downloading entire media file
- Only then: passing it to media player
- **Long delay** if significant file size
- What if media player downloaded media file by itself?
- **Introducing “meta file”**

Components of **PMD** using Meta File



HTTP
Process for **PMD** using Meta File



User clicking on hyperlink

GET response containing contents of meta file

Browser accessing “Content-Type:” field from meta file

Browser using field to invoke related media player

Browser passing meta file to media player

Media player reading URL of original file from meta file

Media player obtaining contents of original file using HTTP/TCP

Player streaming received contents into play-out buffer

Player starting to read stream from buffer

Media player playing resulting stream to sound/video card

Header
Information Location

End of file
(AVI format)

- **First download, then play**
- **Playback start by**: total download time
- **Min bandwidth**: no (full pre-buffering)

Beginning of file
(streaming formats)

- **Simultaneous download and play**
- **Playback start by**: bandwidth statistics
- **Min bandwidth**: yes (smooth playback)

PMD Players

Browser

» Opening media file **URL** by browser

» Caching internally by browser

» Media file played directly by browser

Plug-in

» Redirecting media file **URL** to plug-in

» Caching externally by plug-in

» Media file played indirectly by plug-in

Software components

add specific functionalities

allowing customization &optimization

Pros and Cons of **PMD**

Pros

Clients can start the playback before the whole file gets downloaded

Part of the existing infrastructure

No configuration required

Cons

No dynamic flow control

» Media stops to play when playback rate exceeds download rate

No interactive streaming

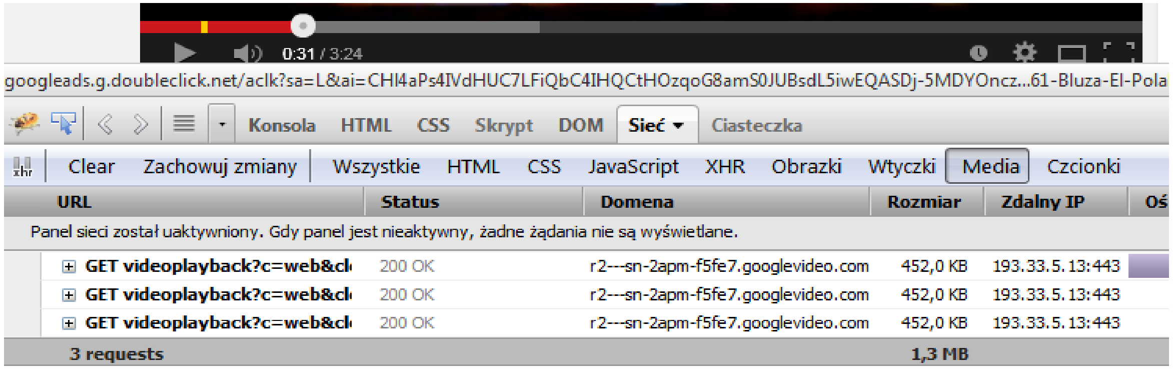
No support for multicast

Large overhead

What is Chunked **PMD**?

Progress Bar

Iteratively growing list of fragments (chunks) requested for download



googleads.g.doubleclick.net/aclk?sa=L&ai=CHl4aPs4IVdHUC7LFiQbC4IHQCtHOzqoG8amS0JUBsdL5iwEQASDj-5MDYOncz...61-Bluza-El-Pola

Panel sieci został uaktywniony. Gdy panel jest nieaktywny, żadne żądania nie są wyświetlane.

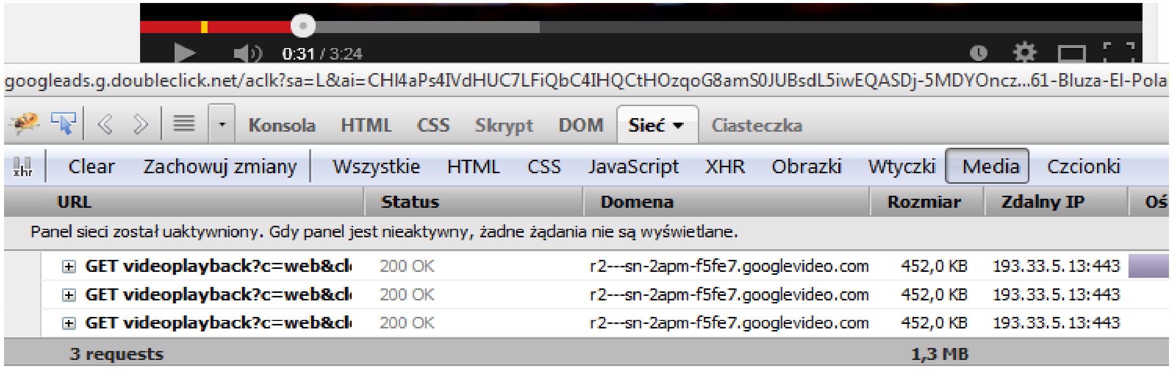
URL	Status	Domena	Rozmiar	Zdalny IP	Oś
GET videoplayback?c=web&cl	200 OK	r2---sn-2apm-f5fe7.googlevideo.com	452,0 KB	193.33.5.13:443	
GET videoplayback?c=web&cl	200 OK	r2---sn-2apm-f5fe7.googlevideo.com	452,0 KB	193.33.5.13:443	
GET videoplayback?c=web&cl	200 OK	r2---sn-2apm-f5fe7.googlevideo.com	452,0 KB	193.33.5.13:443	
3 requests			1,3 MB		

- Not requesting full content in **HTTP** request anymore
- Breaking content into sequence of small **HTTP-based** file segments
- Each segment containing short interval of playback time of content

What is Chunked **PMD**?

Progress Bar

Iteratively growing list of fragments (chunks) requested for download



googleads.g.doubleclick.net/aclk?sa=L&ai=CHl4aPs4IVdHUC7LFiQbC4IHQCtHOzqoG8amS0JUBsdL5iwEQASDj-5MDYOncz...61-Bluza-El-Pola

Panel sieci został uaktywniony. Gdy panel jest nieaktywny, żadne żądania nie są wyświetlane.

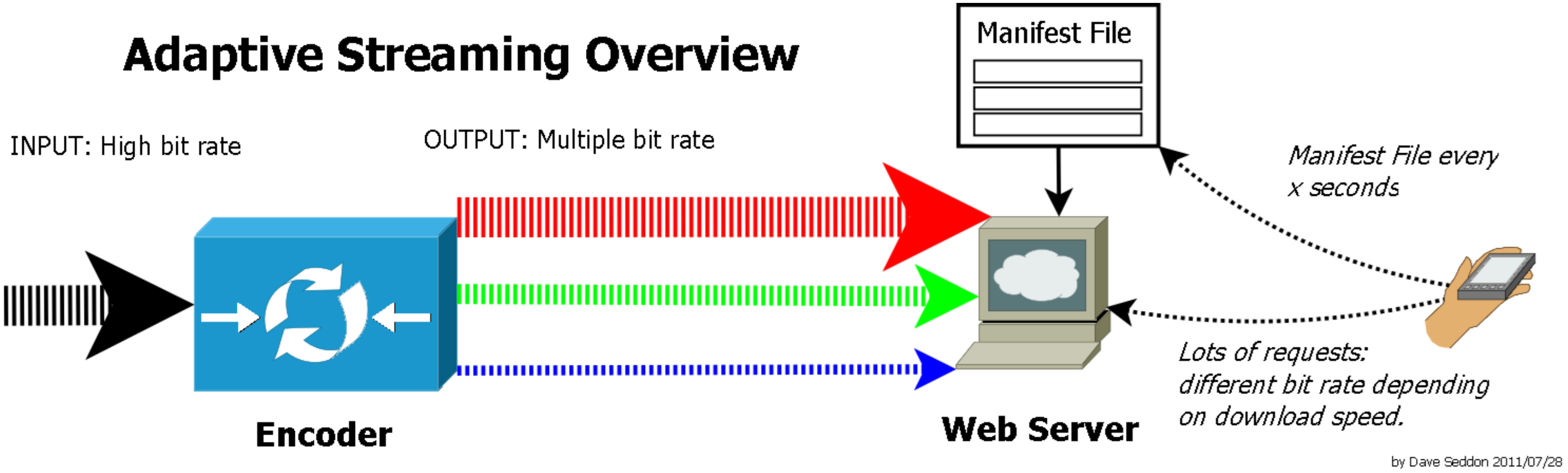
URL	Status	Domena	Rozmiar	Zdalny IP	Oś
GET videoplayback?c=web&cl	200 OK	r2---sn-2apm-f5fe7.googlevideo.com	452,0 KB	193.33.5.13:443	
GET videoplayback?c=web&cl	200 OK	r2---sn-2apm-f5fe7.googlevideo.com	452,0 KB	193.33.5.13:443	
GET videoplayback?c=web&cl	200 OK	r2---sn-2apm-f5fe7.googlevideo.com	452,0 KB	193.33.5.13:443	
3 requests			1,3 MB		

- Technology requesting (e.g.) each **Group of Pictures (GOP)** in a separate request
- Playback begins once chunk downloaded
- **Examples**: **YouTube, Vimeo**

Dynamic Adaptive Streaming over **HTTP**

- Continuous feedback from clients
  - Based on bandwidth and CPU usage
- Content made available at a variety of bit-rates
  - Each stream divided into short length (2 - 10 seconds) fragments
- Clients having the extensive control
  - Smooth changes of quality
  - Subtitles language change

Dynamic Adaptive Streaming over **HTTP** cont.


Adaptive Streaming Overview
Manifest File
INPUT: High bit rate
OUTPUT: Multiple bit rate
Manifest File every x seconds
Encoder
Web Server
Lots of requests: different bit rate depending on download speed.
by Dave Seddon 2011/07/28

Manifest File

- File containing **metadata** for group of accompanying files being part of
  - Set, or
  - Coherent unit
- Term from cargo shipping procedure, where **ship manifest** listing:
  - Crew of vessel, and/or
  - Cargo of vessel
- In **adaptive bitrate streaming**, client downloading manifest (playlist) file describing:
  - Available stream segments, and
  - Their respective bit rates

Manifest
Files for HTTP Streaming

https://www.menti.com/348xpctthx

Solutions for Dynamic Adaptive Streaming over **HTTP**

- Industry solutions (similar to each-other) – mystery of terms:
  - **HTTP Dynamic Streaming (HDS)** by **Adobe**
  - **HTTP Live Streaming (HLS)** by **Apple**
  - **HTTP Silverlight/Smooth Streaming (HSS)** by **Microsoft**
- International standard known as **MPEG-DASH**
- **MPEG-DASH i**s codec-agnostic, which means it can use content encoded with any coding format.

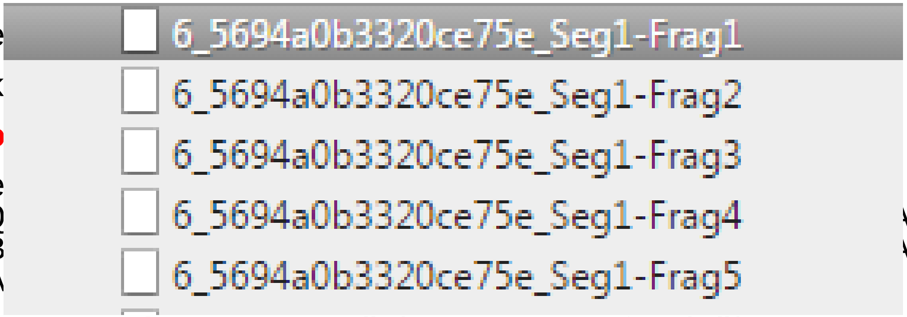
HDS

- Supports both: live and on-demand deliveries
- **Multi-Bit-Rate** (**MBR**) support
- Accepting **RTMP** feed (over port **1935**) as input signal from encoder
- Allows for parallel publishing to backup location – **redundancy** (**backup stream**)

To zdjęcie, autor: Nieznany autor, licencja: CC BY-SA

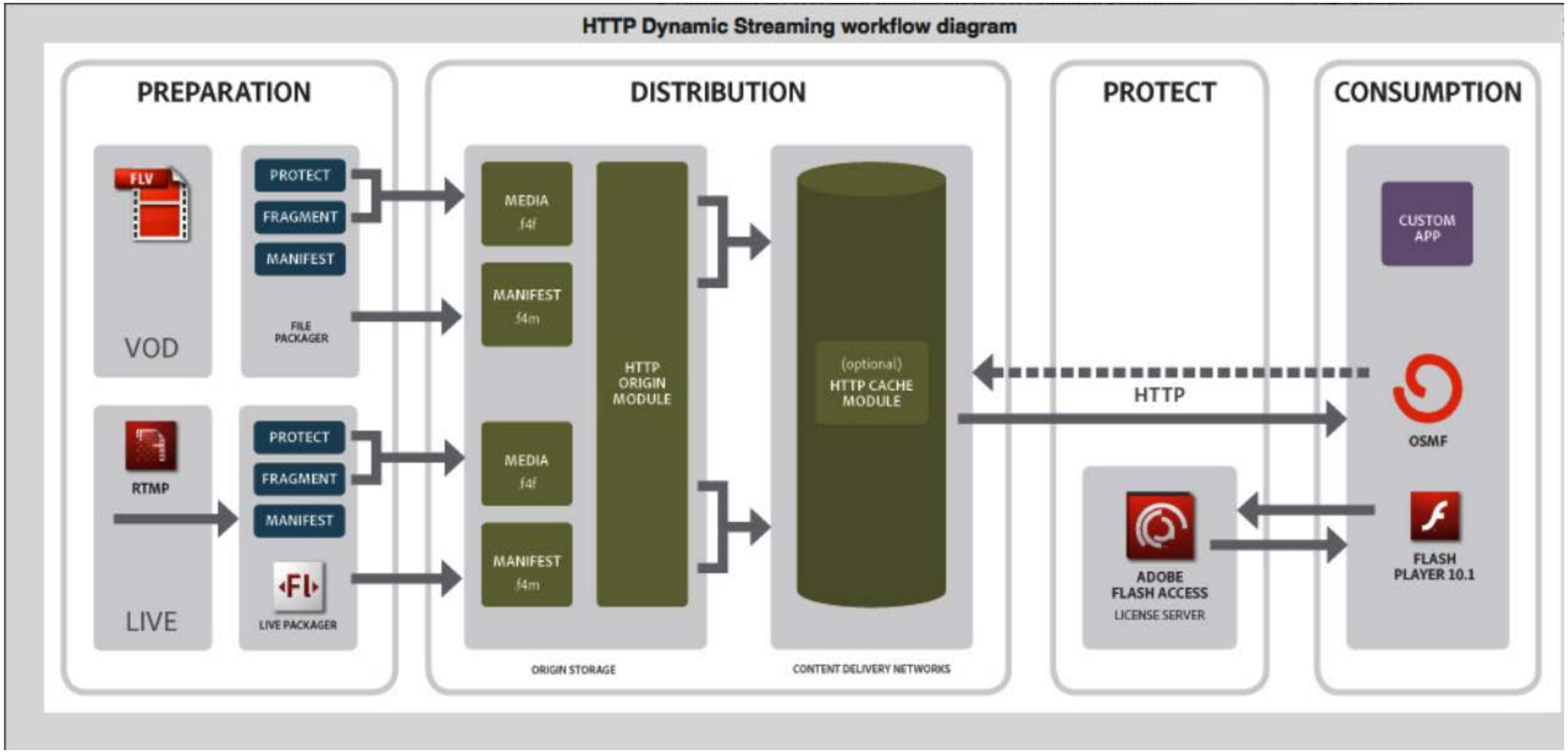
Syntax of **manifest.f4m** file – Encoded Suite of Details about Order of Fragments (**.f4**)

```
<?xml version="1.0" encoding="UTF-8"?>

<manifest xmlns="http://ns.adobe.com/f4m/1.0"
xmlns:akamai="uri:akamai.com/f4m/1.0">

<id>/multi/companion/nba_game/nba_game.mov_,300,600,800,1000,2500,4000,9000,k.mp4
.csmil_0</id>

  <streamType>recorded</stre

  <akamai:streamType>vod</ak

  <duration>306.093</duratio

<bootstrapInfo profile="name
id="bootstrap_6">AAAAi2Fic3Q                                               QAAABlh
c3J0AAAAAAAAAAABAAAAAQAAADMB                                               AXcAAAA
DMAAAAAAAST4AAAF80AAAAAAAAAA

<media bitrate="9326" url="6_5694a0b3320ce75e_" bootstrapInfoId="bootstrap_6">
<metadata>AgAKb25NZXRhRGF0YQgAAAAMAAhkdXJhdGlvbgBAcyF87ZFocwAFd2lkdGgAQJ4AAAAAAAA
ABmhlaWdodABAkOAAAAAAAAANdmlkZW9kYXRhcmF0ZQBAwheSP3JsVAAJZnJhbWVyYXRlAEA998/p06bg
AAx2aWRlb2NvZGVjaWQAQBwAAAAAAAAADWF1ZGlvZGF0YXJhdGUAQE/8AJQpGtMAD2F1ZGlvc2FtcGxlc
mF0ZQBA53AAAAAAAAAPYXVkaW9zYW1wbGVzaXplAEAwAAAAAAAAAAZzdGVyZW8BAQAMYXVkaW9jb2RlY2
lkAEAkAAAAAAAAAAhmaWxlc2l6ZQBBtUVpJAAAAAACQ==</metadata>
```



☐ 6_5694a0b3320ce75e_Seg1-Frag1
☐ 6_5694a0b3320ce75e_Seg1-Frag2
☐ 6_5694a0b3320ce75e_Seg1-Frag3
☐ 6_5694a0b3320ce75e_Seg1-Frag4
☐ 6_5694a0b3320ce75e_Seg1-Frag5

HDS Architecture Overview



HLS

Dedicated mostly for users of **iOS** and **macOS** (**Mac**)

» But other vendors implemented their clients based on the open specification (documented in RFC 8216)

Supports live and on-demand streaming

Multi Bit-Rate (**MBR**) support

Allows for parallel publishing to backup location – **redundancy** (**backup stream**)

Exemplary URL: https://bitdash-a.akamaihd.net/content/sintel/hls/playlist.m3u8

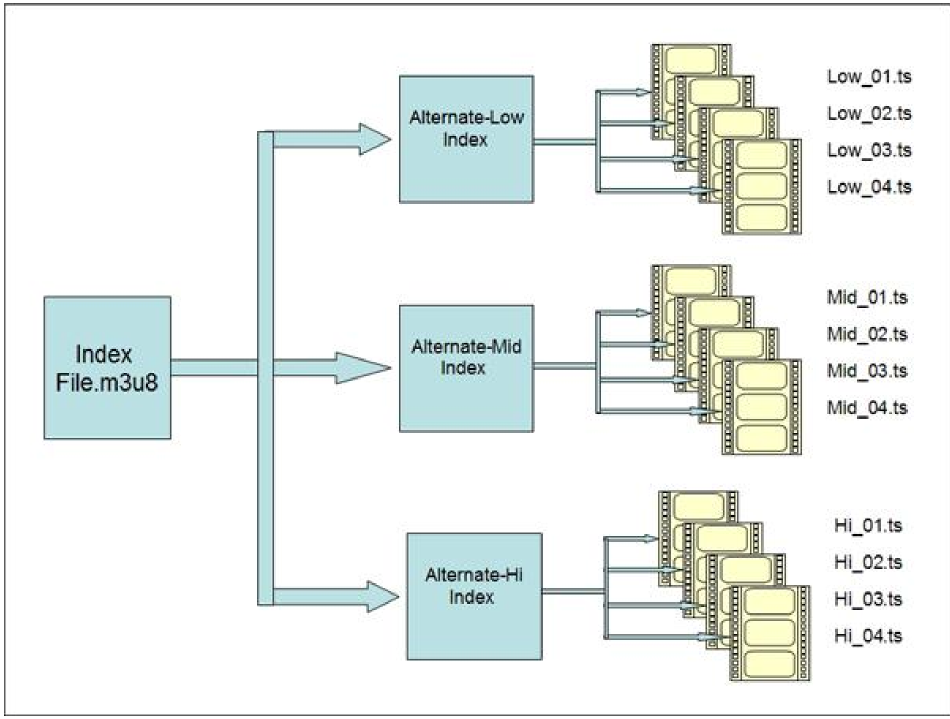
To zdjęcie, autor: Nieznany autor, licencja: CC BY-SA

Example of **master.m3u8** File – But Where Is Reference to Segments (**.ts**)?!

```
#EXTM3U
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=548000,RESOLUTION=320x240,CODECS="avc1.66.30, mp4a.40.34"
http://livetest_rm-lh.akamaihd.net/i/stream_1@143961/index_500_av-p.m3u8?sd=10&rebase=on
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=548000,RESOLUTION=320x240,CODECS="avc1.66.30, mp4a.40.34"
http://livetest_rm-lh.akamaihd.net/i/stream_1@143961/index_500_av-b.m3u8?sd=10&rebase=on
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1048000,RESOLUTION=320x240,CODECS="avc1.66.30, mp4a.40.34"
http://livetest_rm-lh.akamaihd.net/i/stream_1@143961/index_1000_av-p.m3u8?sd=10&rebase=on
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=1048000,RESOLUTION=320x240,CODECS="avc1.66.30, mp4a.40.34"
http://livetest_rm-lh.akamaihd.net/i/stream_1@143961/index_1000_av-b.m3u8?sd=10&rebase=on
```

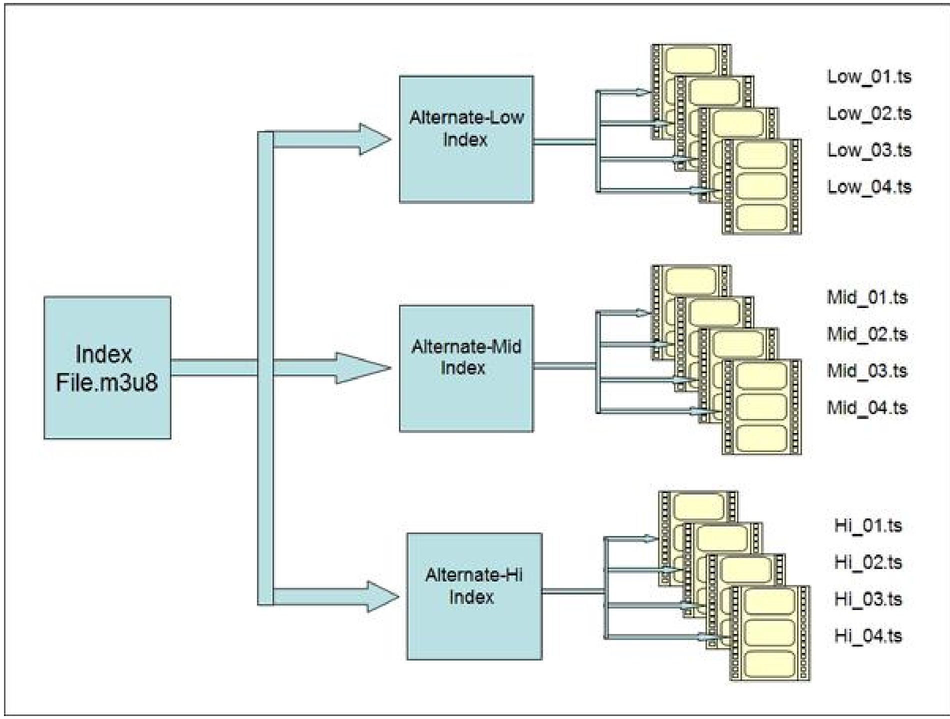
...One Level Deeper (Hierarchical Tree)

```
#EXTM3U
#EXT-X-TARGETDURATION:10
#EXT-X-ALLOW-CACHE:YES
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:75973
#EXTINF:3.249,
http://livetest_rm-lh.akamaihd.net/i/stream_1@143961/segment75973_500_av-p.ts
#EXTINF:10.000,
http://livetest_rm-lh.akamaihd.net/i/stream_1@143961/segment75974_500_av-p.ts
#EXTINF:10.000,
http://livetest_rm-lh.akamaihd.net/i/stream_1@143961/segment75975_500_av-p.ts
```



...One Level Deeper (Hierarchical Tree)

```
#EXTM3U
#EXT-X-TARGETDURATION:10
#EXT-X-ALLOW-CACHE:YES
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:75973
#EXTINF:3.249,
http://livetest_rm-lh.akamaihd.net/i/stream_1@143961/segment75973_500_av-p.ts
#EXTINF:10.000,
http://livetest_rm-lh.akamaihd.net/i/stream_1@143961/segment75974_500_av-p.ts
#EXTINF:10.000,
http://livetest_rm-lh.akamaihd.net/i/stream_1@143961/segment75975_500_av-p.ts
```



How to Convert "**Typical**" **.flv** or **.mp4** Video into **.m3u8**?

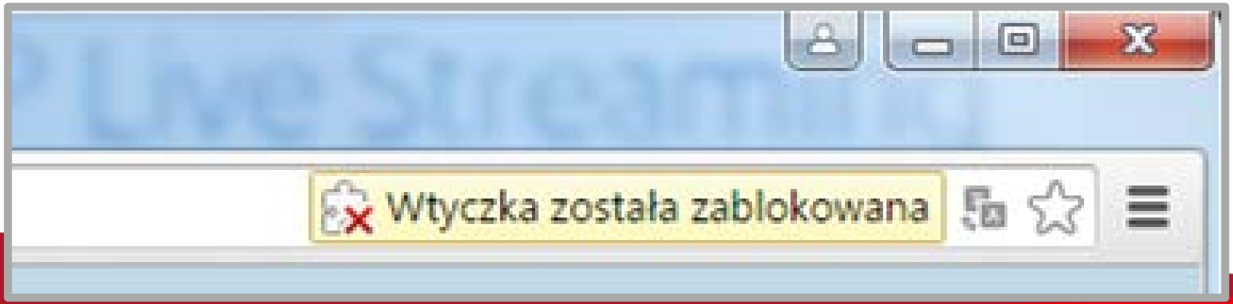
» Very easily ☺

» Using **FFMPEG**:

```
ffmpeg -i webcam.flv -vcodec
h264 -hls_list_size 0
-segment_list webcam.m3u8
-segment_format ts
test_webcam.m3u8
```

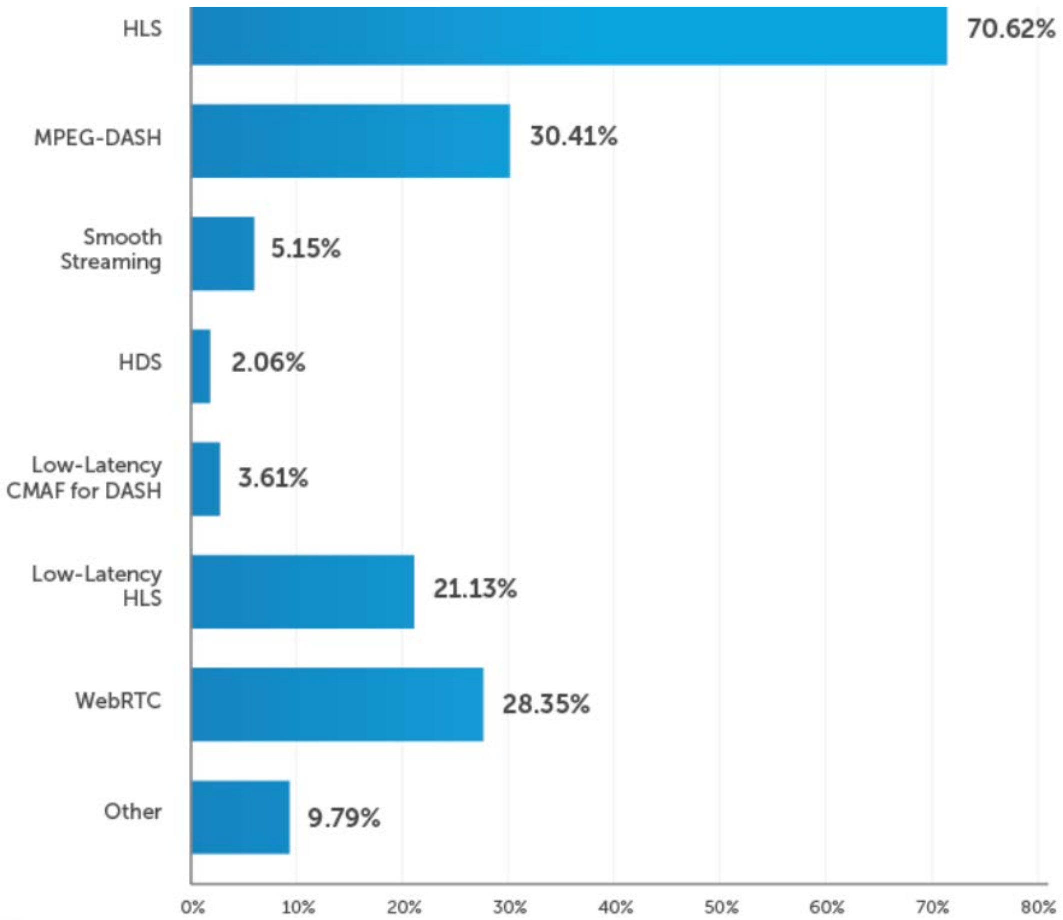
HSS

- Similar (tree-like) hierarchy of files:
  - Client manifest (**.ismc**)
  - Video fragment (**.ismv**)
  - Audio fragment (**.isma**)
- **No HSS support in browsers since 2015!**



Wtyczka została zablokowana

Which streaming formats are you currently using?



Comparison of Solutions

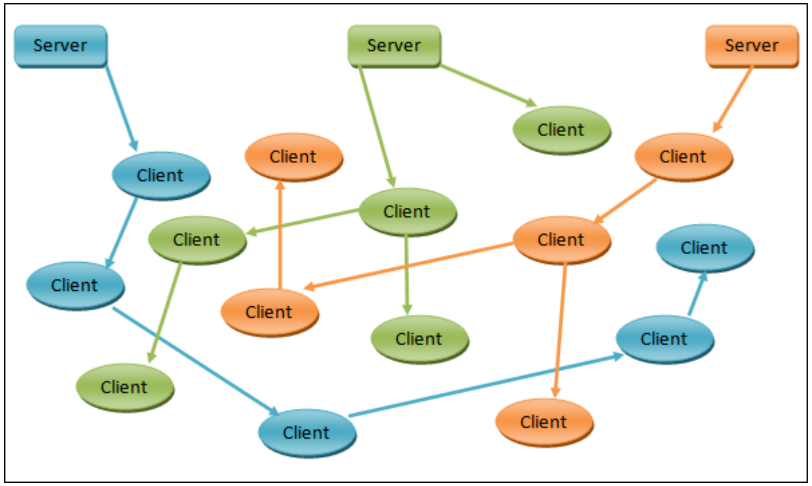
Streaming Protocol	Advantages	Trade-Offs
HTTP Live Streaming (HLS)	• Practical and flexible adaptive bitrate streaming protocol for streaming audio and video over the internet • Automatically adjust the quality of the stream based on the viewer's internet connection speed for better UX • Supported by most modern devices and platforms • Popular due to ease of use • Extremely cost effective for on-demand and large audience broadcast when paired with a CDN	• Not suited for real time communication or extremely low latency live streaming • Viewers of live events will experience a delay

Comparison of Solutions

Streaming Protocol	Advantages	Trade-Offs
Dynamic Adaptive Streaming over HTTP (DASH)	• Popular adaptive bitrate streaming protocol • International standard for streaming media • Vendor-independent alternative to HLS • Extremely cost effective for on-demand and large audience broadcast when paired with a CDN	• Not suited for real time communication or extremely low latency live streaming • Viewers of live events will experience a delay

Peer-to-Peer (P2P) Networks Serving Video Streams

- Used in connection with other streaming methods: **RTP** and **PMD**
- Increased geographical coverage
- Better bandwidth utilisation
- Saving costs to broadcaster
- Long latency times (even up to 90 seconds!)



Attribution: **Soumyasch** at **English language Wikipedia**

Recapitulation

PMD

- Multimedia content stored on a local machine
- No rate control
- Traditionally, the whole file is transmitted (even if it is not necessary)
- No need to keep many versions of the same file (on the server side)
- Uses existing HTTP architecture

Streaming

- No content stored on a local machine
- Adaptive rate control
- Only the chunk being watched is transmitted
- Built-in support for fast-forwarding
- Uses existing HTTP architecture

Reference

Pantos, R., & May, W. (2017). HTTP live streaming (No. RFC 8216). [Online] https://tools.ietf.org/html/rfc8216

Streaming Protocols

https://www.wowza.com/blog/streaming-protocols

The manifest (playlist) files for HTTP streaming can be built as…

https://www.mentimeter.com/s/39651b0f05220895431b3db039bf40e3/25de82744365