

NGINX for Commercial Quality Streaming Services

**Seungyeob Choi | schoi@verizon.com
Manager, Software Engineering
Verizon Digital Media Services**

Agenda

1. Introduction to streaming
2. Turn NGINX into a streaming server
3. HLS/DASH VOD by NGINX – How?
4. Supporting HLS 7
5. Encrypting HLS
6. High-bitrate streaming (4K/8K UHD) by NGINX
7. Low-latency live by NGINX

Introduction to Streaming

Streaming by HLS & MPEG-DASH

- **HLS and MPEG-DASH are the two most widely accepted formats.**
- **The content is divided to small chunks, each containing a few seconds of video/audio.**
- **A playlist file provides the list of all chunks.**
- **Playlist and chunks can either be pre-processed, or be prepared upon request.**

Pre-packaging to HLS using ffmpeg

```
ffmpeg -i video.mp4 -codec copy
-vbsf h264_mp4toannexb
-f segment
-segment_list video.m3u8
-segment_time 10
-segment_format mpeg_ts
-segment_list_type m3u8
video%04d.ts
```

video.m3u8

```
#EXTINF:10,
video0000.ts
#EXTINF:10,
video0001.ts
#EXTINF:10,
video0002.ts
#EXTINF:10,
video0003.ts
...
```

**List all segments
of 10 seconds**

**Cut mp4 into a 10
second segment
and repackage in
MPEG2 TS format.**

MP4

.ts

.ts

.ts

...

Pre-packaging to MPEG-DASH using gpac (MP4Box)

```
MP4Box -dash 10000 -rap  
-out video.mpd  
-segment-name video_  
video.mp4
```

video.mpd

```
<SegmentURL media="video_1.m4s"/>  
<SegmentURL media="video_2.m4s"/>  
<SegmentURL media="video_3.m4s"/>  
<SegmentURL media="video_4.m4s"/>  
<SegmentURL media="video_5.m4s"/>  
<SegmentURL media="video_6.m4s"/>...
```

List all
segments of
10 seconds

MP4

.m4s

.m4s

.m4s

...

Cut mp4 into a 10
second segment
and repackage in
fragmented-mp4
format.

Turn NGINX to a streaming server

Download and install NGINX RTMP module

- **Developed by Roman Arutyunyan**
- **Git: <https://github.com/arut/nginx-rtmp-module>**
- **Blog: <http://nginx-rtmp.blogspot.com/>**

```
sudo apt-get install -y unzip build-essential libpcre3 libpcre3-dev libssl-dev
cd /opt
wget http://nginx.org/download/nginx-1.10.1.tar.gz
wget https://github.com/arut/nginx-rtmp-module/archive/master.zip
tar -zxvf nginx-1.10.1.tar.gz
unzip -o master.zip
cd nginx-1.10.1

./configure --with-http_ssl_module --add-module=../nginx-rtmp-module-master
make
make install
```


HLS Live

- A live application can have HLS mode on.
- HLS playlist (.m3u8) and segment (.ts) files are generated in real time from ingested stream.

```
rtmp {  
    server {  
        application mylive {  
            live on;  
            hls on;  
            hls_path /usr/local/nginx/html/hls;  
            hls_fragment 10s;  
        }  
    }  
}
```

DASH Live

- A live application can have DASH mode on.
- DASH playlist (.mpd) and segment (.m4v and .m4a) files are generated in real-time from ingested stream.

```
rtmp {  
    server {  
        application mylive {  
            live on;  
  
            hls on;  
            hls_path /usr/local/nginx/html/hls;  
            hls_fragment 10s;  
  
            dash on;  
            dash_path /usr/local/nginx/html/dash;  
            dash_fragment 10s;  
        }  
    }  
}
```

VOD streaming by NGINX Plus

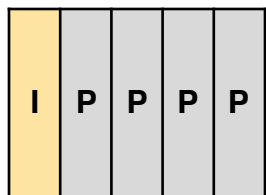
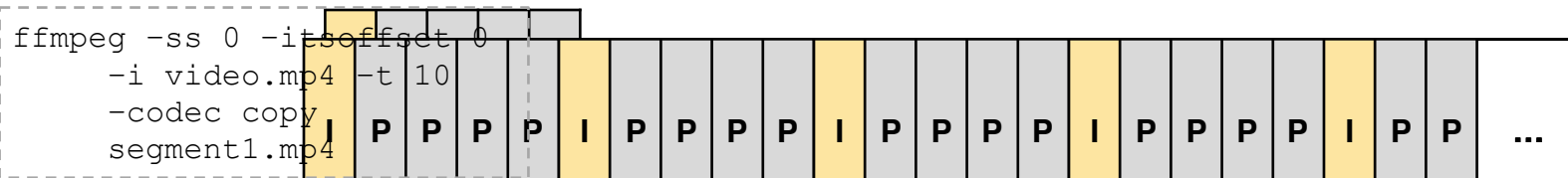
- **hls directive turns on dynamic HLS segmentation.**
- **NGINX Plus can prepare HLS manifest (.m3u8) and segment (.ts) files on the fly.**
- **f4f directive turns on HDS mode.**
- **NGINX Plus can prepare HDS manifest (.f4m) and segments.**
- **But, mp4 file needs to be pre-processed to f4f using Adobe's f4fpackager.**
- **Not much benefit if compared with pre-packaging.**

```
http {
    server {
        application myhls {
            hls;
            hls_fragment 10s;
        }
    }
}
```

```
http {
    server {
        application myhds {
            f4f;
        }
    }
}
```

**HLS/DASH
VOD by
NGINX
- How?**

How to segment



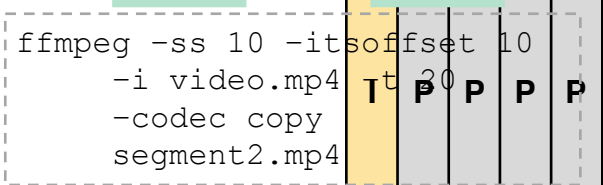
10s

20s

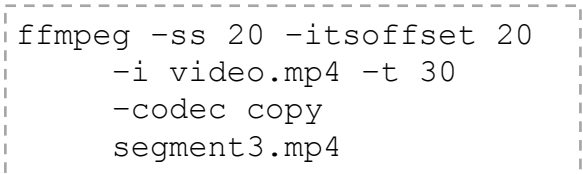
20s

30s

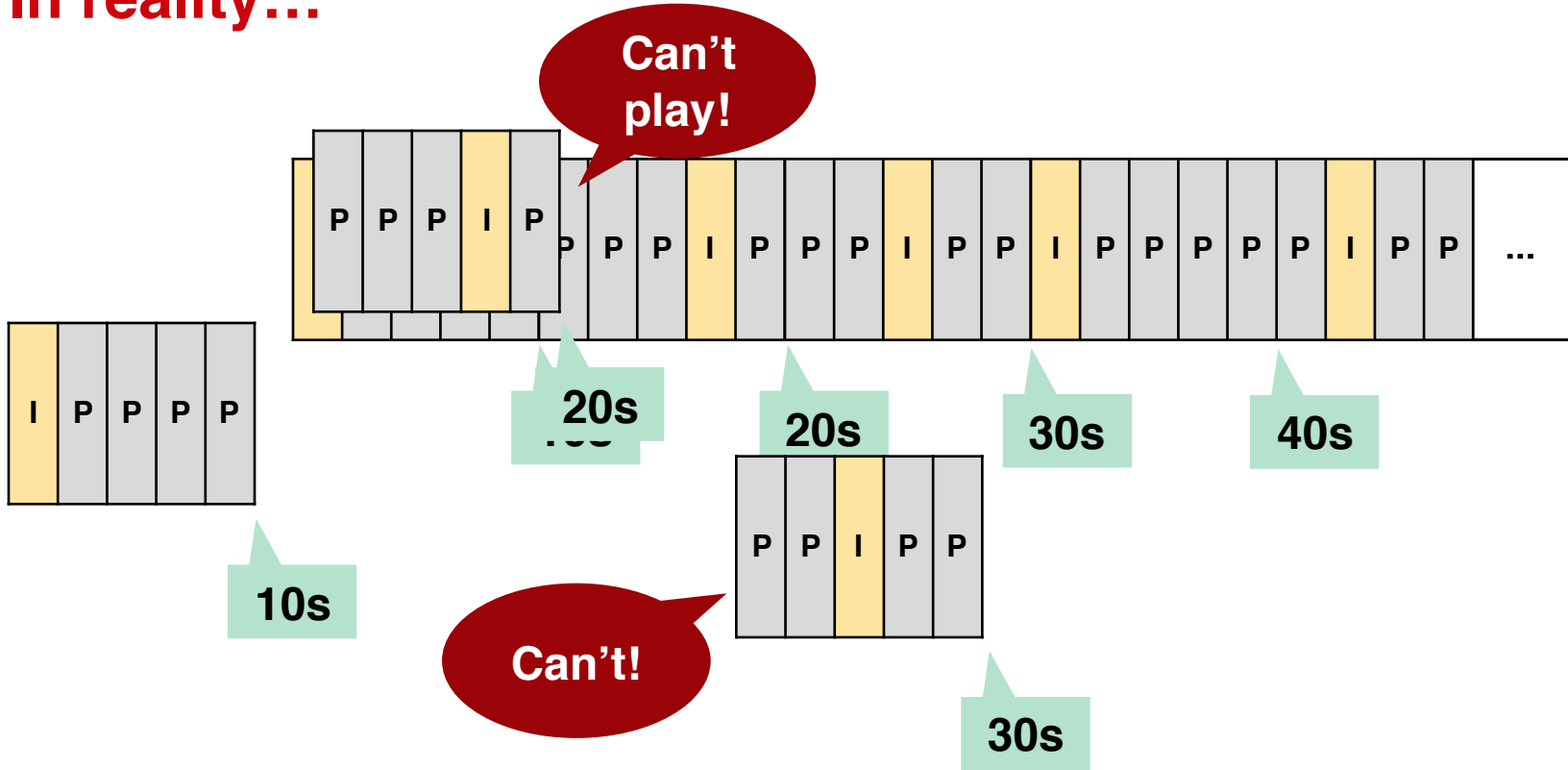
40s



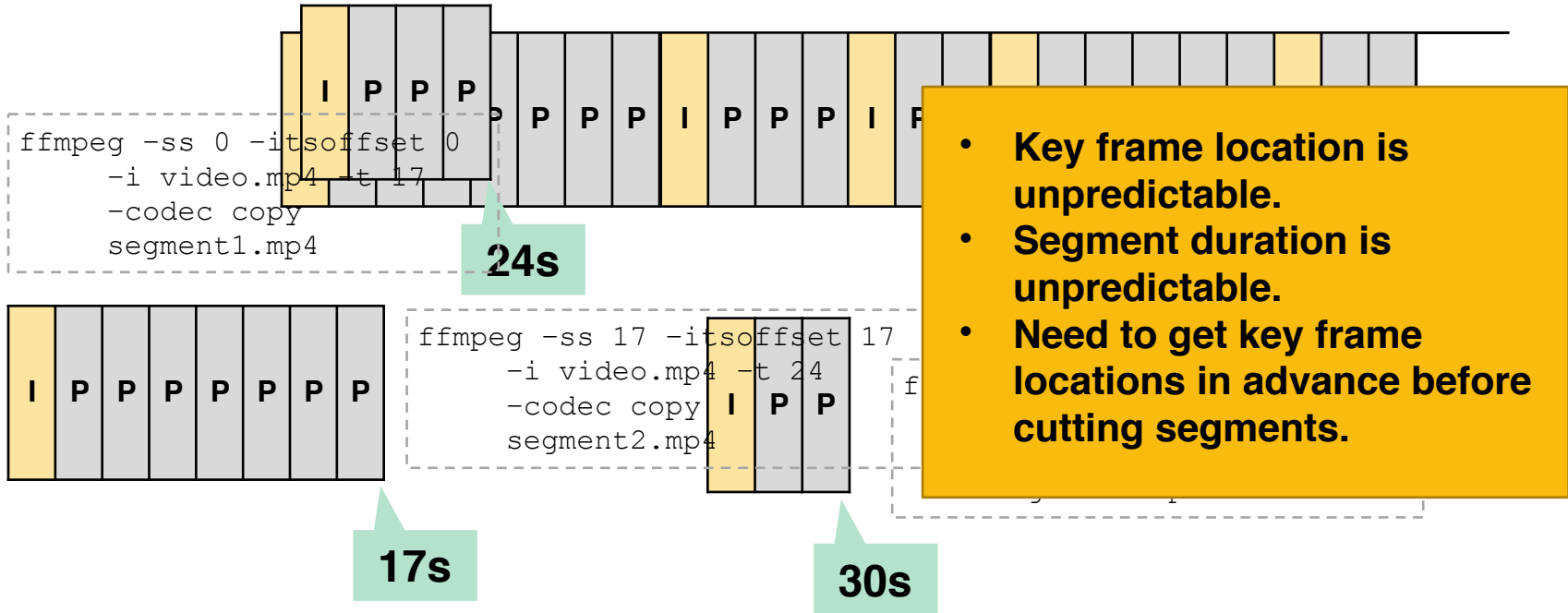
30s



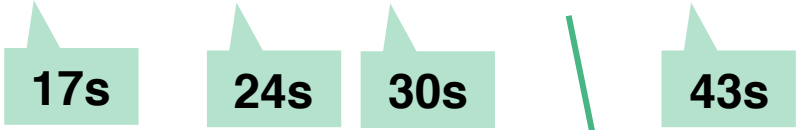
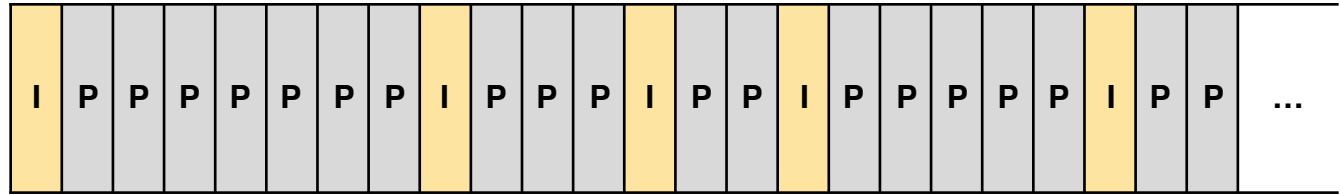
In reality...



Then, how to cut real mp4?



The art of getting segment boundaries



Get the location of key frames

```
ffprobe -select_streams v -skip_frame nokey  
-show_frames -v quiet video.mp4  
| grep '^pkt_pts_time'  
| sed 's/pkt_pts_time=//'
```

0.000000
17.000000
24.000000
30.000000
43.000000
...

Generate HLS playlist

- The playlist can be generated from a template using key frame locations.

Template

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-ALLOW-CACHE:YES
#EXT-X-TARGETDURATION:sizeof($N)

FOR EACH $N[i] {

#EXTINF:($N[i+1] - $N[i]),
$file.$N[i].ts

}
```

Segment boundaries

```
$N={
0.000000,
17.000000,
24.000000,
30.000000,
43.000000,
...
}
```

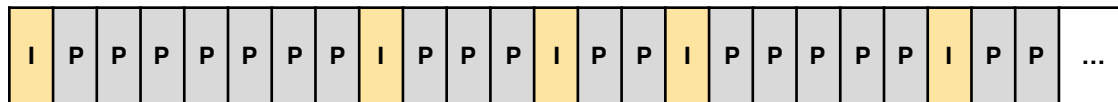
File name

```
$file=
video.mp4
```

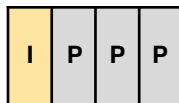
video.mp4.m3u8

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:0
#EXT-X-ALLOW-CACHE:YES
#EXT-X-TARGETDURATION:10
#EXTINF:17.000000,
video.mp4.0.000000.ts
#EXTINF:7.000000,
video.mp4.17.000000.ts
#EXTINF:6.000000,
video.mp4.24.000000.ts
#EXTINF:13.000000,
video.mp4.30.000000.ts
#EXTINF:8.000000,
video.mp4.43.000000.ts
...
```

Generate HLS segment



```
ffmpeg -ss 17 -itsoffset 17  
-i video.mp4 -t 24  
-codec copy  
segment2.mp4
```



video.mp4.17000.ts

MP2TS

```
ffmpeg -i segment2.mp4 -codec copy  
-vbsf h264_mp4toannexb  
-flags -global_header  
-segment_format mpegts  
segment2.ts
```

DASH playlist and segment

video.mp4.mpd

Segment boundaries

```
$N={  
  0.000000,  
  17.000000,  
  24.000000,  
  30.000000,  
  43.000000,  
  ...  
}
```



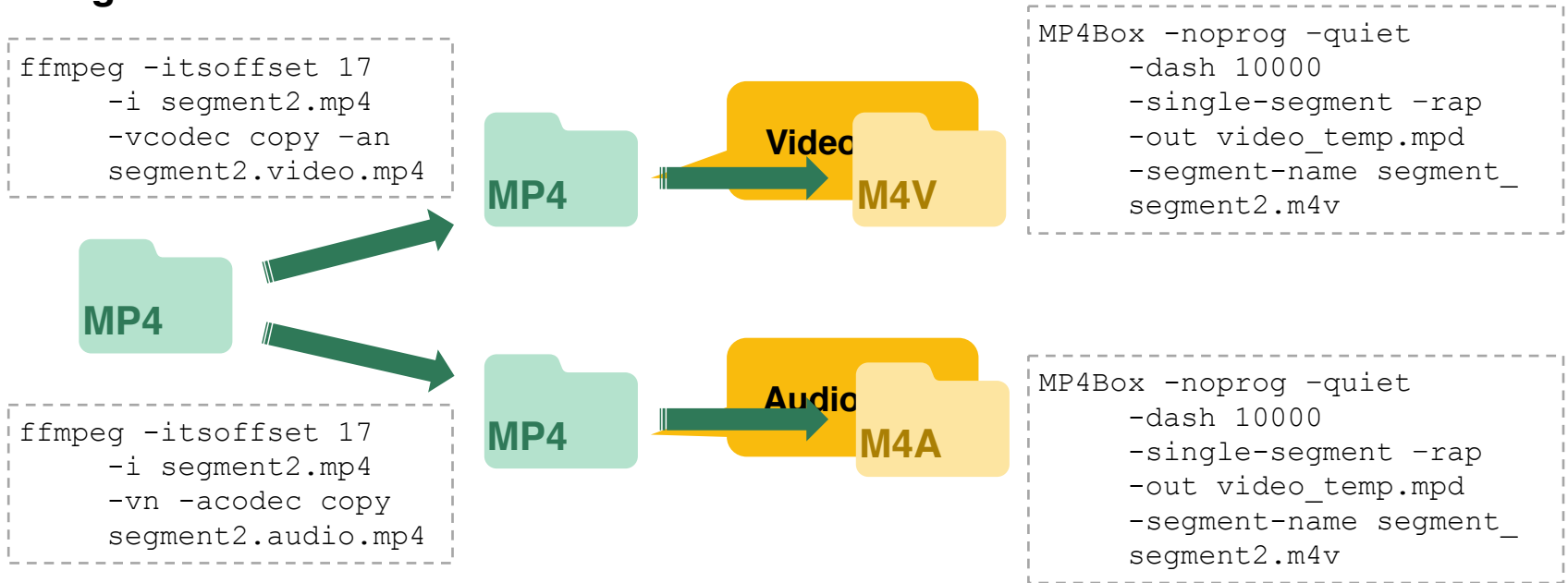
Template



```
<MPD type="static" xmlns="urn:mpeg:DASH:schema:MPD:2011"  
  profiles="urn:mpeg:dash:profile:isoff-on-demand:2011"  
  minBufferTime="PT1.5S"  
  mediaPresentationDuration="PT0H0M40S">  
  <Period start="PT0S" duration="PT0H0M40S">  
    <AdaptationSet segmentAlignment="true" bitstreamSwitching="true">  
      <Representation id="avc1:9938805"  
        mimeType="video/mp4" codecs="avc1.42c01e"  
        width="640" height="264"  
        startWithSAP="1" bandwidth="144000">  
        <SegmentTemplate presentationTimeOffset="0" timescale="1000"  
          initialization="video.mp4.m4v" media="video.mp4+$Time$.m4v">  
          <SegmentTimeline>  
            <S t="0" d="17000"/>  
            <S t="17000" d="7000"/>  
            <S t="24000" d="6000"/>  
            <S t="30000" d="13000"/>  
  
            ...  
          </SegmentTimeline>  
        </SegmentTemplate>  
      </Representation>  
    </AdaptationSet>  
  
    ...  
  </Period>  
</MPD>
```

Track split and repackage (MP4Box)

- Most DASH players don't like video and audio tracks combined in a single segment.



Supporting HLS 7 and up

HLS 7 revision 20 – What's new?

- **HLS now support segments in fMP4.**
 - Can share segments with MPEG-DASH and Microsoft Smooth Streaming.
 - Can split video and audio (multiple audio tracks are now cheaper than before).
 - Can save CPU and storage!
- **Master playlist is now mandatory, even if the stream is a single-bitrate.**

Playlist format

HLS 3 Playlist

```
#EXTM3U
#EXT-X-VERSION:3
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-PLAYLIST-TYPE:VOD
#EXT-X-ALLOW-CACHE:NO
#EXT-X-TARGETDURATION:12
#EXTINF:10.677,
vide.mp4+0.ts
#EXTINF:9.801,
vide.mp4+10677.ts
#EXTINF:9.551,
vide.mp4+20478.ts
#EXTINF:11.261,
vide.mp4+30030.ts
#EXTINF:10.344,
vide.mp4+41291.ts
#EXTINF:8.425,
vide.mp4+51634.ts
#EXT-X-ENDLIST
```

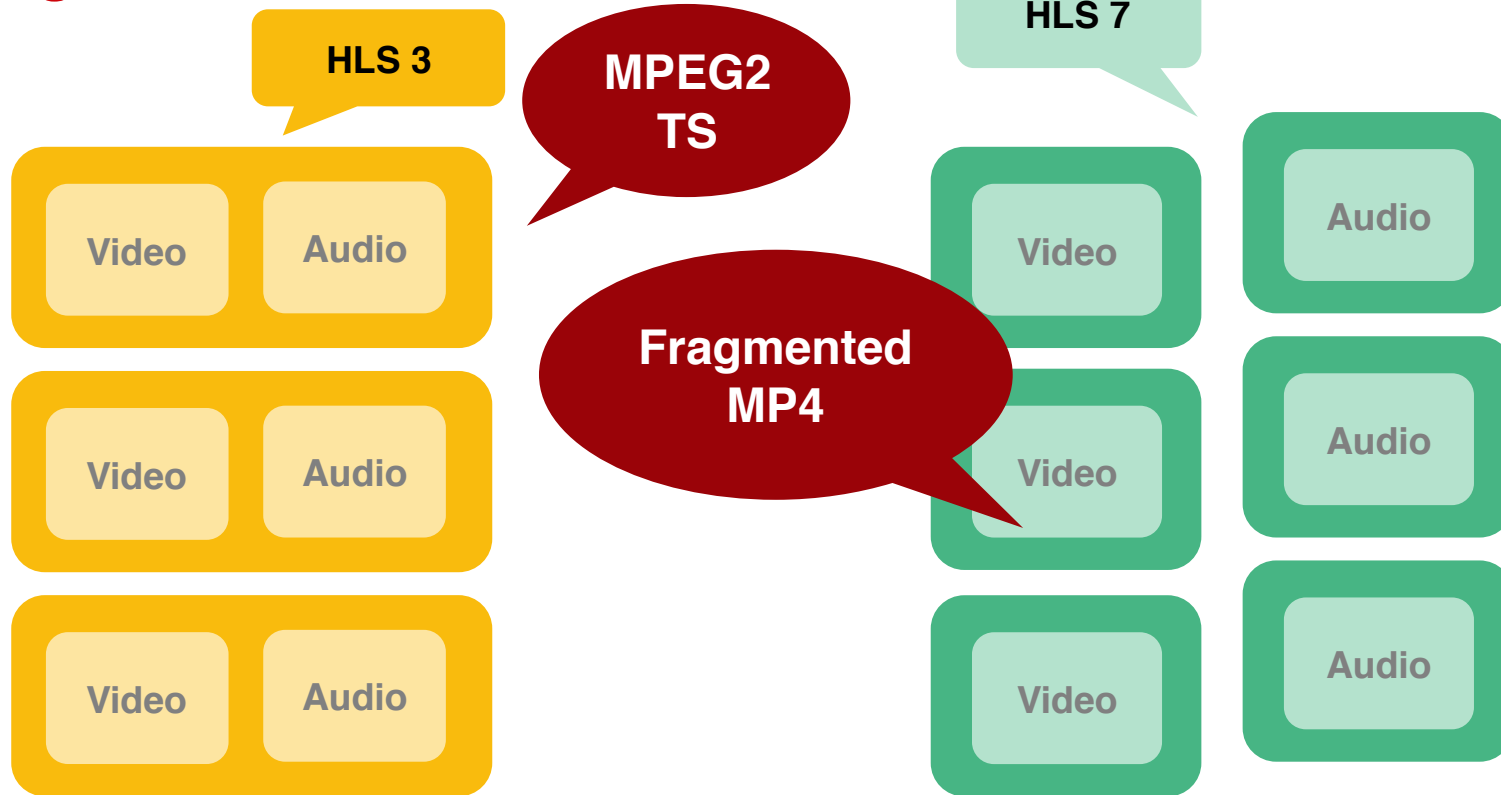
```
#EXTM3U
#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=10130832,CODEC="avc1.640033,mp4a.40.2",RESOLUTION=3840x2160
vide.mp4_1.m3u8
#EXT-X-MEDIA:TYPE=CLOSED-CAPTIONS,GROUP-ID="cc1",LANGUAGE="en",NAME="English",AUTOSELECT=YES,DEFAULT=YES,INSTREAM-ID="CC1"
```

HLS 7 Master playlist

HLS 7 Media playlist

```
#EXTM3U
#EXT-X-VERSION:7
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-PLAYLIST-TYPE:VOD
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-MAP:URI="1.mp4"
#EXT-X-TARGETDURATION:12
#EXTINF:10.67733,
1+0.mp4
#EXTINF:9.80146,
1+512512.mp4
#EXTINF:9.55121,
1+982982.mp4
#EXTINF:11.26125,
1+1441440.mp4
#EXTINF:10.34367,
1+1981980.mp4
#EXTINF:8.42508,
1+2478476.mp4
#EXT-X-ENDLIST
```

Segments



How to support HLS 7 r20 using NGINX RTMP

```
#EXTM3U

#EXT-X-STREAM-INF:PROGRAM-ID=1,BANDWIDTH=10130832,CODEC="avc1.640033,mp4a.40.2",RESOLUTION=3840x2160
vide.mp4_1.m3u8

#EXT-X-MEDIA:TYPE=CLOSED-CAPTIONS,GROUP-ID="cc1",LANGUAGE="xx",NAME="Undefined",AUTOSELECT=YES,DEFAULTSELECT=NO,URI="cc1.m3u8",PROGRAM-ID="CC1"
```

HLS 7 Master playlist

HLS 7 Media playlist

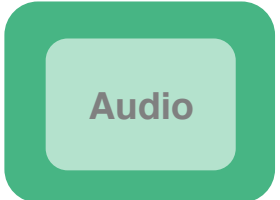
```
#EXTM3U
#EXT-X-VERSION:7
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-PLAYLIST-TYPE:VOD
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-MAP:URI="1.mp4"
#EXT-X-TARGETDURATION:12
#EXTINF:10.67733,
1+0.mp4
#EXTINF:9.80146,
1+512512.mp4
#EXTINF:9.55121,
1+982982.mp4
#EXTINF:11.26125,
1+1441440.mp4
#EXTINF:10.34367,
1+1981980.mp4
#EXTINF:8.42508,
1+2478476.mp4
#EXT-X-ENDLIST
```

Create by rules

Convert from DASH mpd

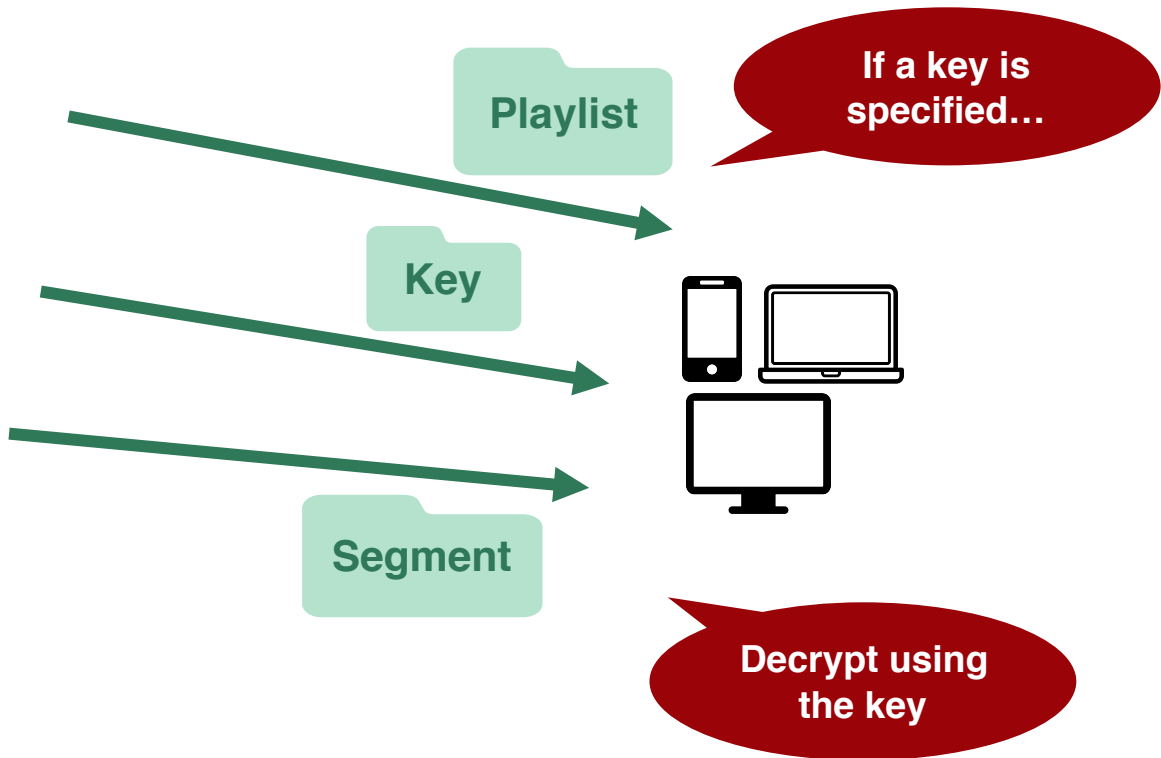
Reuse DASH segments

HLS 7 Segments



Encrypting HLS

Protecting HLS – How does it work?



Protecting HLS – How to make it work?

```
#EXTM3U
#EXT-X-VERSION:7
#EXT-X-KEY:METHOD=AES-128,URI="video.key",IV=0x00000000000000000000000000000000
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-PLAYLIST-TYPE:VOD
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-MAP:URI="1.mp4"
#EXT-X-TARGETDURATION:12
#EXTINF:10.67733,
1+0.mp4
#EXTINF:9.80146,
1+512512.mp4
#EXTINF:9.55121,
1+982982.mp4
...
```

```
openssl enc -aes-128-cbc -in "original.mp4" -K "video.key" -p -iv 0
-out "encrypted.mp4"
```

**High-bitrate
streaming
(4K/8K UHD)
by NGINX**

4K/8K UHD – How big are they?

When horizontal and vertical resolutions are doubled, the amount of data are quadrupled. As a result, the number of pixels are growing exponentially in bigger resolution.

- SD 640x480 = 300K pixels.
- HD 1280x720 = 900K pixels (3x SD)
- UHD 4096x2160 = 8.44M pixels (9.4x HD, 28x SD)

Vimeo recommendation

<https://vimeo.com/help/compression>

Quality	Bitrate (Mbps)
SD	2-5
720p	5-10
1080p	10-20
2K	20-30
4K	30-60
8K	50-80

Optimization for smooth 4K/8K experience

- Segment duration
- Key frame interval
- Multi-bitrate
- Network optimization
- HEVC/H.265

2-4 seconds is recommended.
Longer segments require longer
second

Make sure to connect to is

(Roughly) half-sized data for the
same quality, or (also roughly)
twice quality from the same
bitrate.

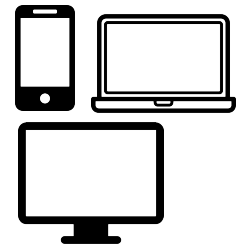
**Low-latency
live
streaming by
NGINX**

Low latency live – Why is it a such big challenge?

HTTP-based streaming was originally not designed for live streaming.



Total 45-60 seconds?



Optimization for sub-10 seconds latency

- Smaller segment duration (2-3 seconds)
- Shorter key frame interval
- Smaller playlist window (3x segment)
- Network optimization
- Chunk Transfer Encoding

Segments start being sent before they are complete.

Player buffer is beyond control. We can enforce to be small, by a small playlist window.

The smaller, the better, but should not be too small. (Too small make the player take up after disturbance.)

Thank you.