



HTTP STREAMING

DASH

MPEG-DASH ISO/IEC 23009-1





Content

- **Introduction**
- **Service description**
- **MPD description**
- **Segment description**
- **Profiles**
- **Functionalities**
- **References**



introduction



Terminology

- HTTP adaptive bit rate streaming **HABS**
- Apple HTTP Live Streaming **HLS**
- Adobe HTTP Dynamic Streaming **HDS**
- *Windows Media HTTP Streaming Protocol* **WMSP**
- Dynamic Adaptive Streaming over HTTP **DASH**
 - *Supported by Microsoft, Adobe*



HTTP streaming

- HTTP & TCP not efficient: required bandwidth = 2 x bitrate of media due to TCP overhead
- HTTP + consumption w/o downloading the entire content
- HTTP Streaming ::= delivery of continuous media



HTTP Streaming why

- Reasonable Internet connectivity (i.e., in terms of bandwidth for media content)
- Available anywhere, anytime, and almost on any device
- HTTP does not cause any NAT/firewall issues as it is the case with other media transport protocols like RTP/RTSP
- SERVER simplicity ()



Akamai's principal architect for media engineering, commented:

“We've spent the past five years delivering a variety of adaptive video formats—SmoothHD, HDNI, HLS and HDS—all of which are **80 percent the same but 100 percent incompatible.**”



HTTP Streaming solutions

Apple HTTP Live Streaming HLS

draft-pantos-http-live-streaming-03

MPEG2 TS

Microsoft IIS Smooth Streaming

Appelée technologie Smooth Streaming (silverlight plugin)

MP4

Adobe HTTP Dynamic Streaming HDS

Adobe Flash Media Streaming Serveur

Mp4 (F4V F4F)

Dynamic Adaptive Streaming over HTTP DASH

MPEG standard

MP4 TS



HTTP STREAMING DIFFERENCIATION

All HTTP-based adaptive streaming technologies use a combination of **encoded media files** and **manifest files** that identify alternative streams and their respective URLs.



HTTP STREAMING DIFFERENCIATION

The respective **players** monitor *buffer status* (HLS) and **CPU utilization** (Smooth Streaming and HTTP Dynamic Streaming) and **change streams** as necessary, locating the alternate stream from the URLs specified in the manifest file.



HTTP STREAMING DIFFERENTIATION

HLS uses MPEG-2 Transport Stream (**M2TS**) segments, stored as thousands of tiny M2TS files, while Smooth Streaming and HDS use time-code to find the necessary fragment of the appropriate **MP4** elementary streams.



Streaming HTTP Adaptatif

**Multiple niveaux en fichiers
(n niveaux, n connexions)**

Commutation de Chunk

Closed GOP

GOP de longueur fixe

De 2 sec à 10 sec

Démo : smoothstreaming

<http://www.iis.net/media/experiencesmoothstreaming1080p>



DASH Definition

- **Dynamic Adaptive Streaming over HTTP (DASH)
Standart in 2012**

MPEG-DASH ISO/IEC 23009-1

**Specifies formats that enable delivery of media content
from standard HTTP servers to HTTP clients and
enable caching of content by standard HTTP caches**



DASH Standard

■ Superset for system specifications

- 3GPP Release-9 AHS in TS 26.234 and 3GPP Release-10 DASH TS26.247
- Open IPTV Forum HTTP Adaptive Streaming (HAS)

System specifications may define more: codecs, DRM, etc.

■ International Standard 2012



MPEG-DASH Design Principles

DASH is an enabler

- **provides formats to enable efficient and high-quality streaming over the Internet**
- **System definition left to other organizations (SDOs, Fora, Companies, etc.)**

DASH is not:

- **system, protocol, presentation, codec, middleware, client specification**



HTTP Streaming 3GP

Manifest File (MF): Playlist, Media Presentation Description, ... mostly XML-based
3gm format

Delivery Format (DF)

3gp format

as extensions/specializations of

- ISO Base File Format (ISOBMFF)
- MPEG-2 Transport Stream (M2TS)



Highlighted Features 1

- **Supports adaptive on demand and live streaming such as MPEG-4 file format and MPEG-2TS.**
- **Efficient and ease of use of existing CDNs, proxies, caches, NATs and firewalls.**
- **Control of entire streaming session by the client.**
- **Support of seamless switching of tracks.**
- **The concept of switching and selectable streams.**



Highlighted Features 2

- **Signaling, delivery, utilization of multiple DRM schemes.**
- **Supports ad-insertion.**
- **Segments with variable durations.**
- **Sub-segment alignment indication to simplify switching and avoiding overlapping fragments.**



Highlighted Features 3

- **Manifest fragmentation and assembly for external inclusion of elements.**
- **Content Descriptors for accessibility, roles, rating and camera views.**
- **Multiple base URLs for the same content.**
- **Clock drift control for live sessions.**



Highlighted Features 4

- **Scalable Video Coding (SVC) and Multiview Video Coding (MVC)**
- **Subsetting of representation groups according to the content author's guidance.**
- **Quality metrics for reporting the session experience.**



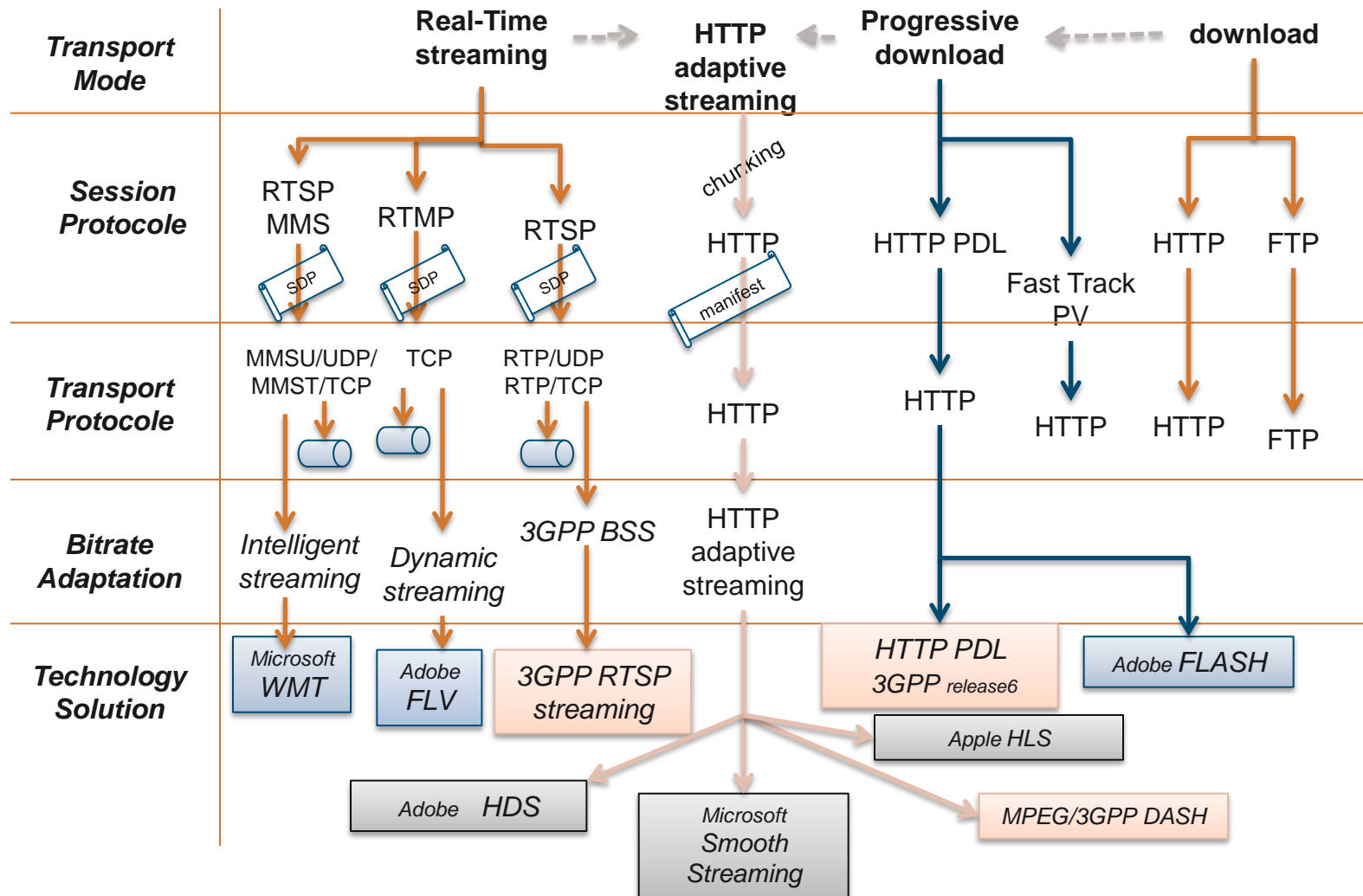
Drawback

- **Does not Standardize the player and server**
- **Bitrate regulation let to the client decision! Peak bitrate risk**
- **Lot of chunk, server risk : file listings ,log processing capacity,**

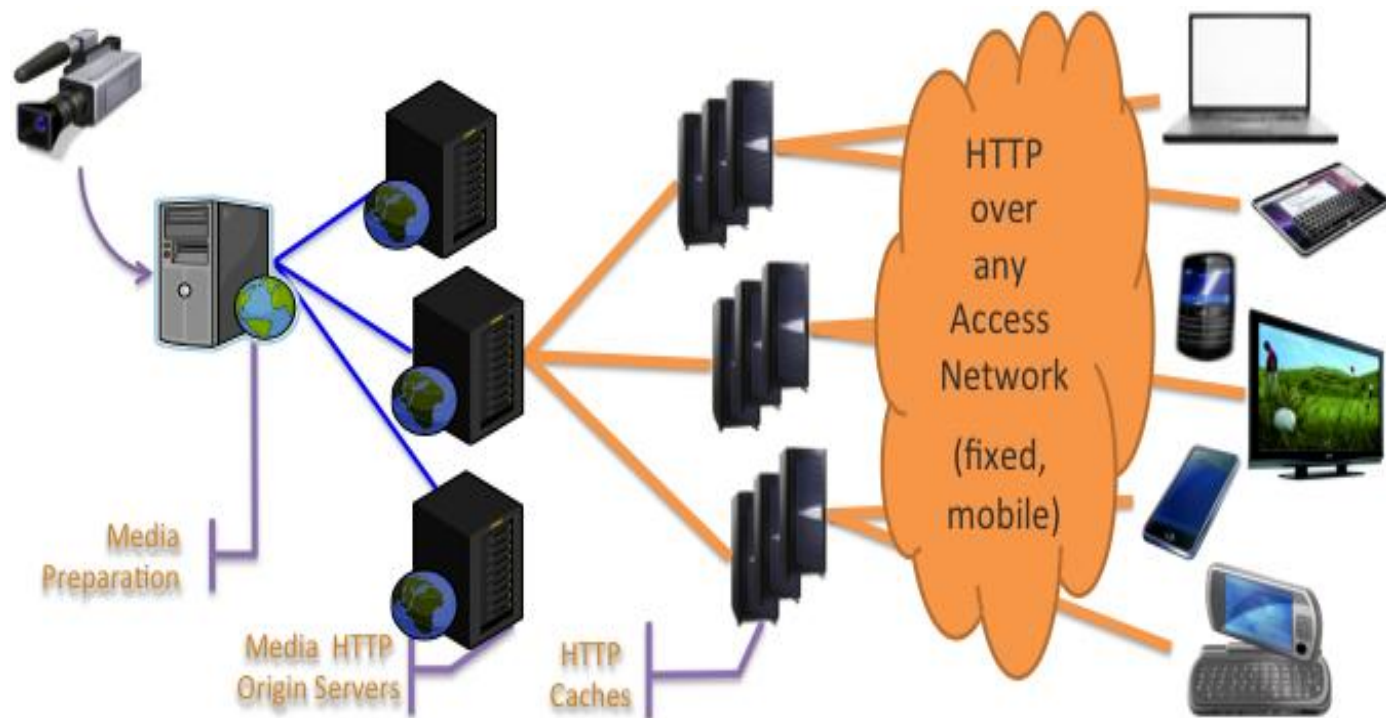


Service Description

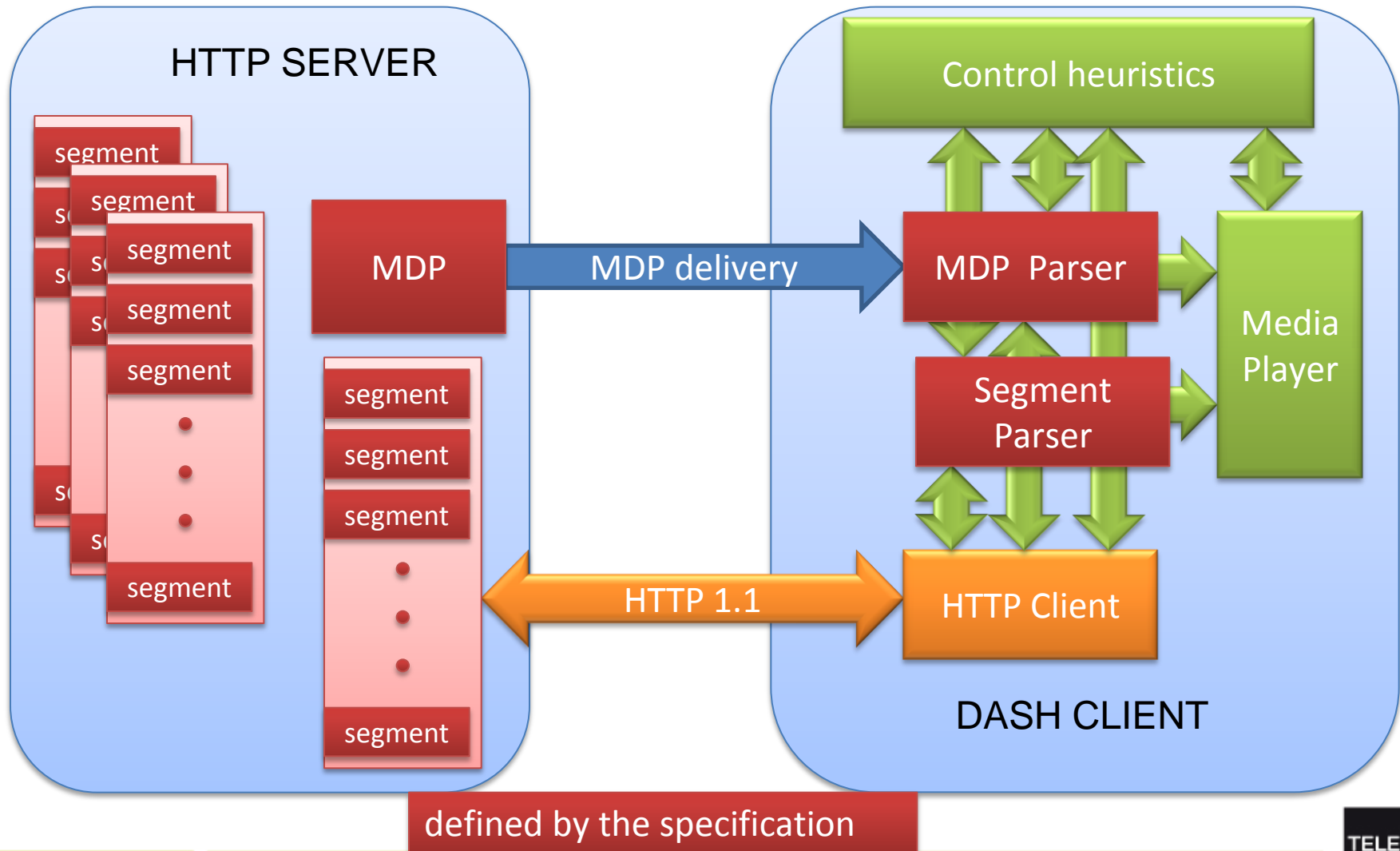
Techniques



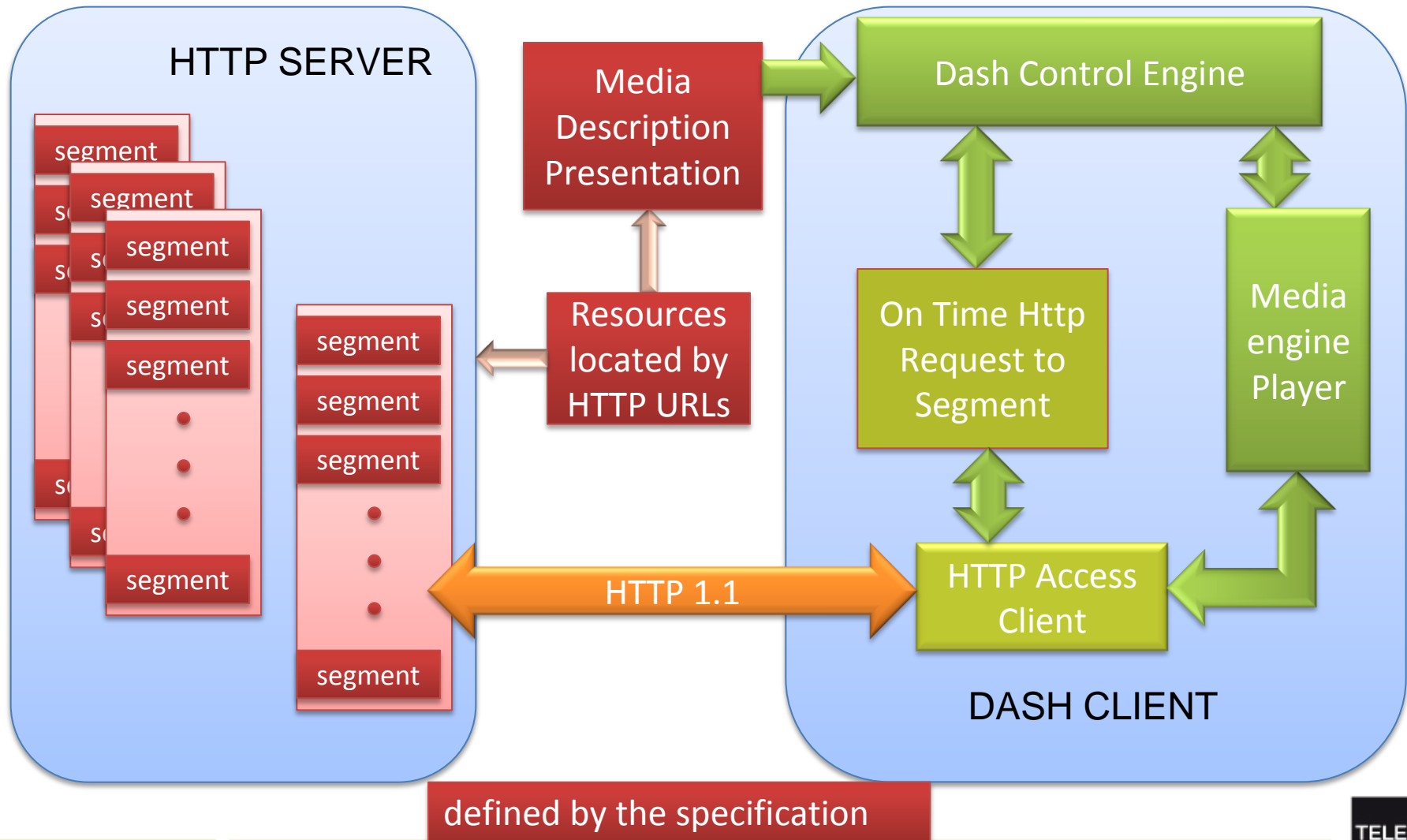
HTTP Scheme



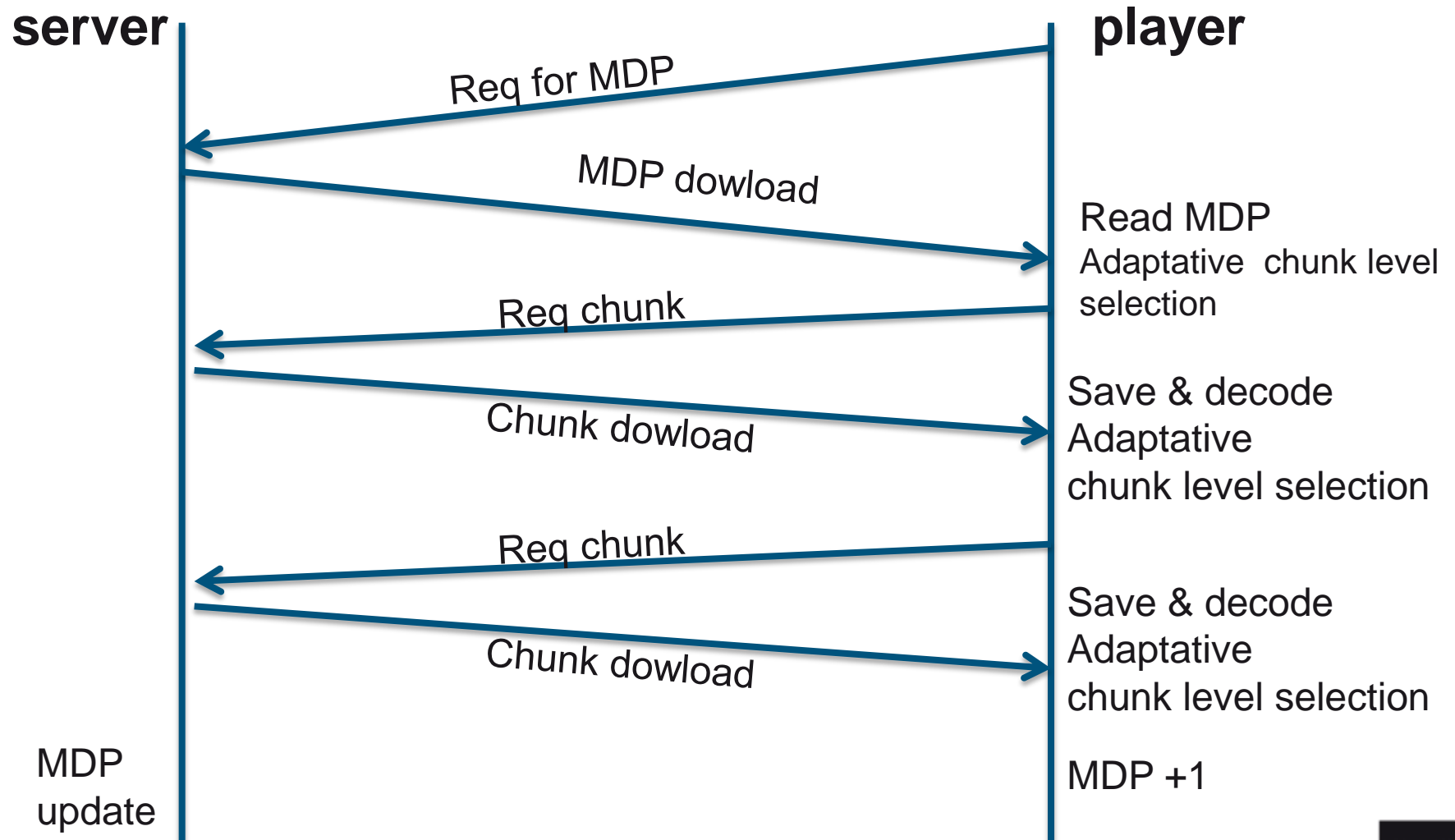
Scope of the MPEG-DASH Standard



Scope of the MPEG-DASH Standard



DASH How To



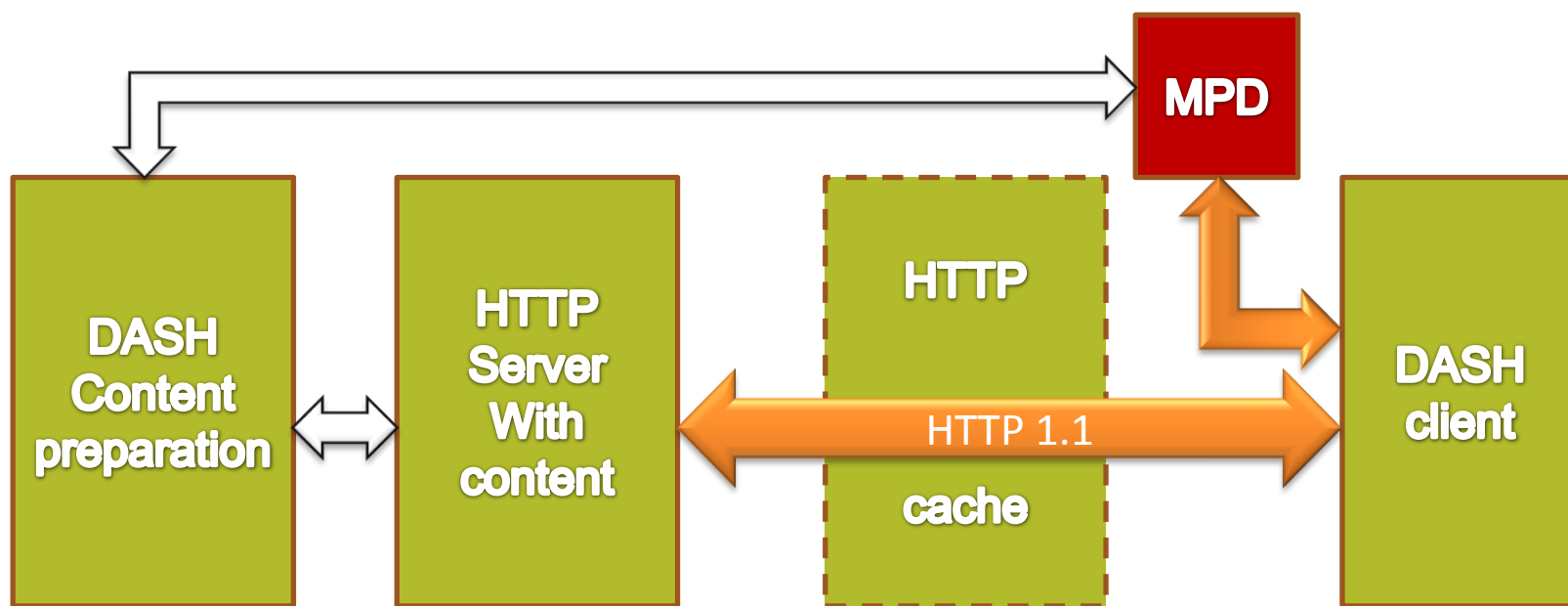


Defines two formats

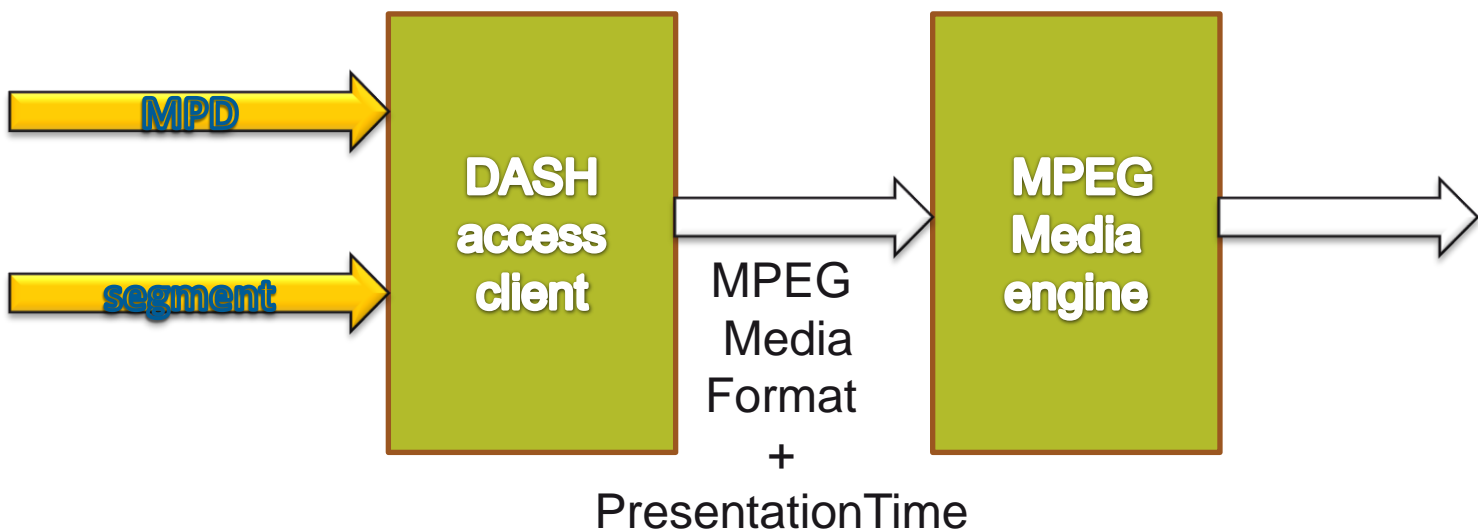
- **Media Presentation Description (MPD)**
- **The Segment formats identified in the MPD (Chunk)**

- **DASH Client is not standardized.**
- **MPD provides sufficient information for a client to provide a streaming service to the user by accessing the Segments through the protocol HTTP/1.1 specified in the scheme of the defined resources.**

Possible DASH Architecture



DASH Client Model





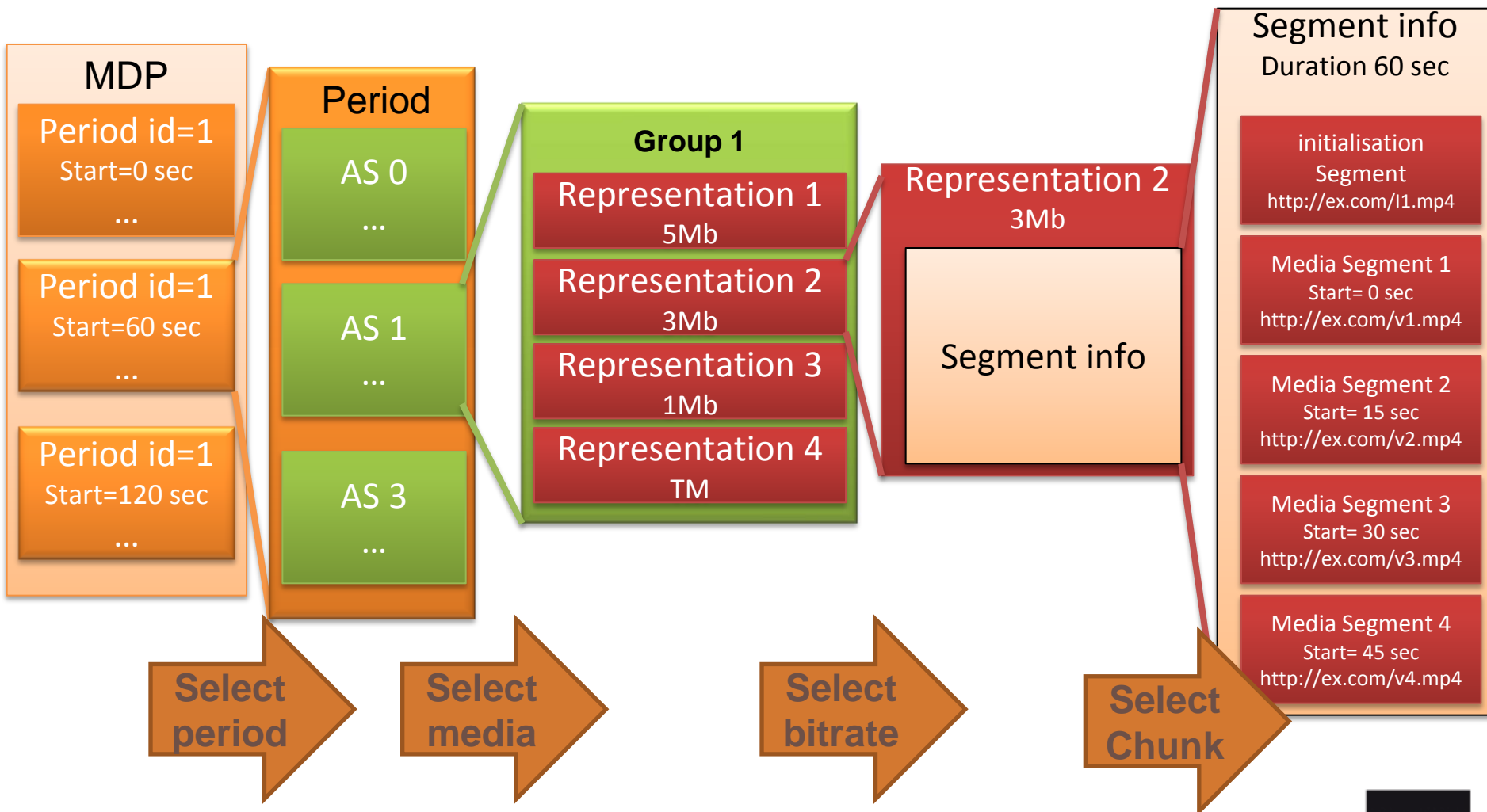
MDP description



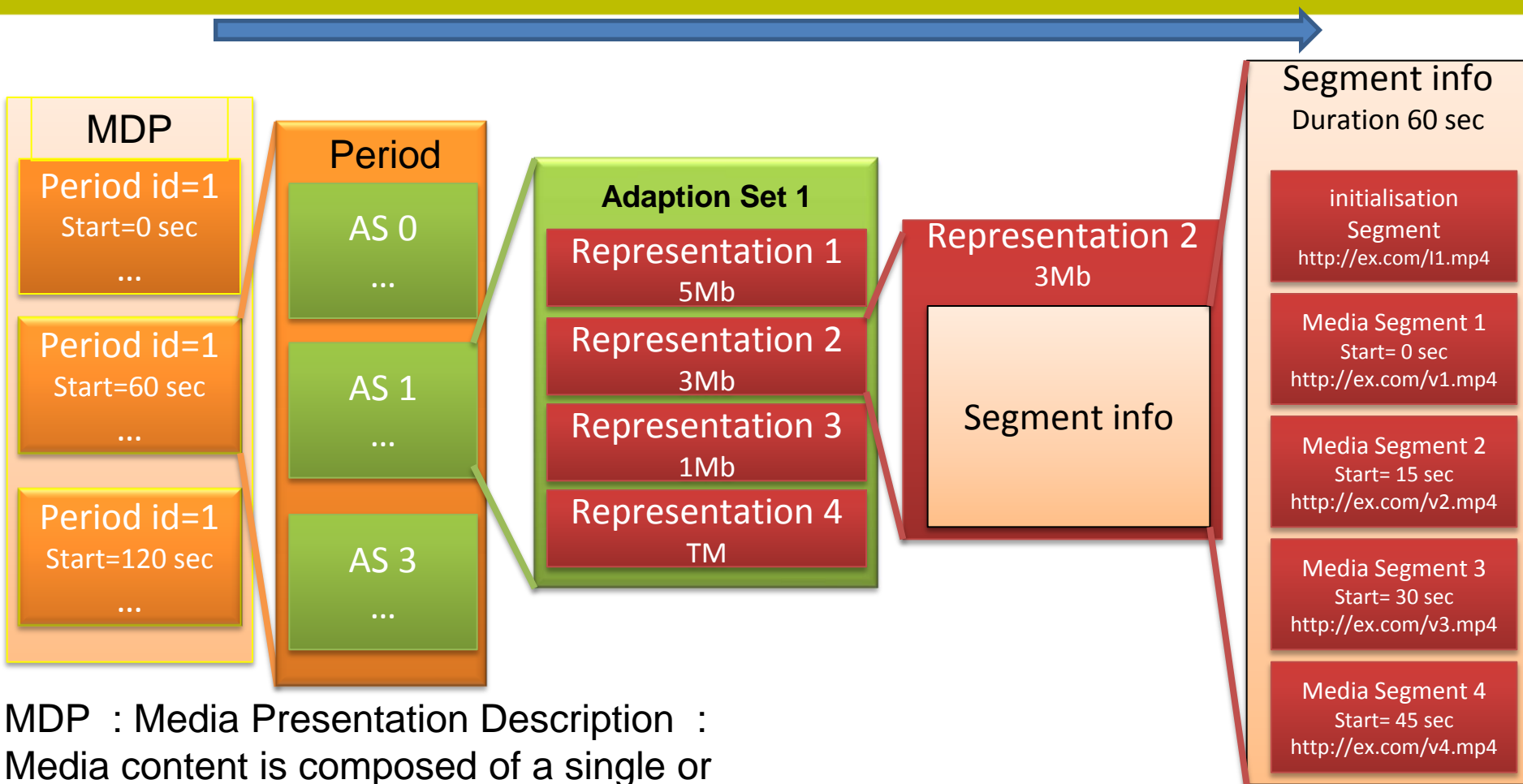
MPD Description

- **MPD is an XML-document**
- **collection of data that is accessible via URL to a DASH Client**
- **HTTP-URLs may be absolute or relative**
- **updates of the MPD possible**

MPD hierarchical data model

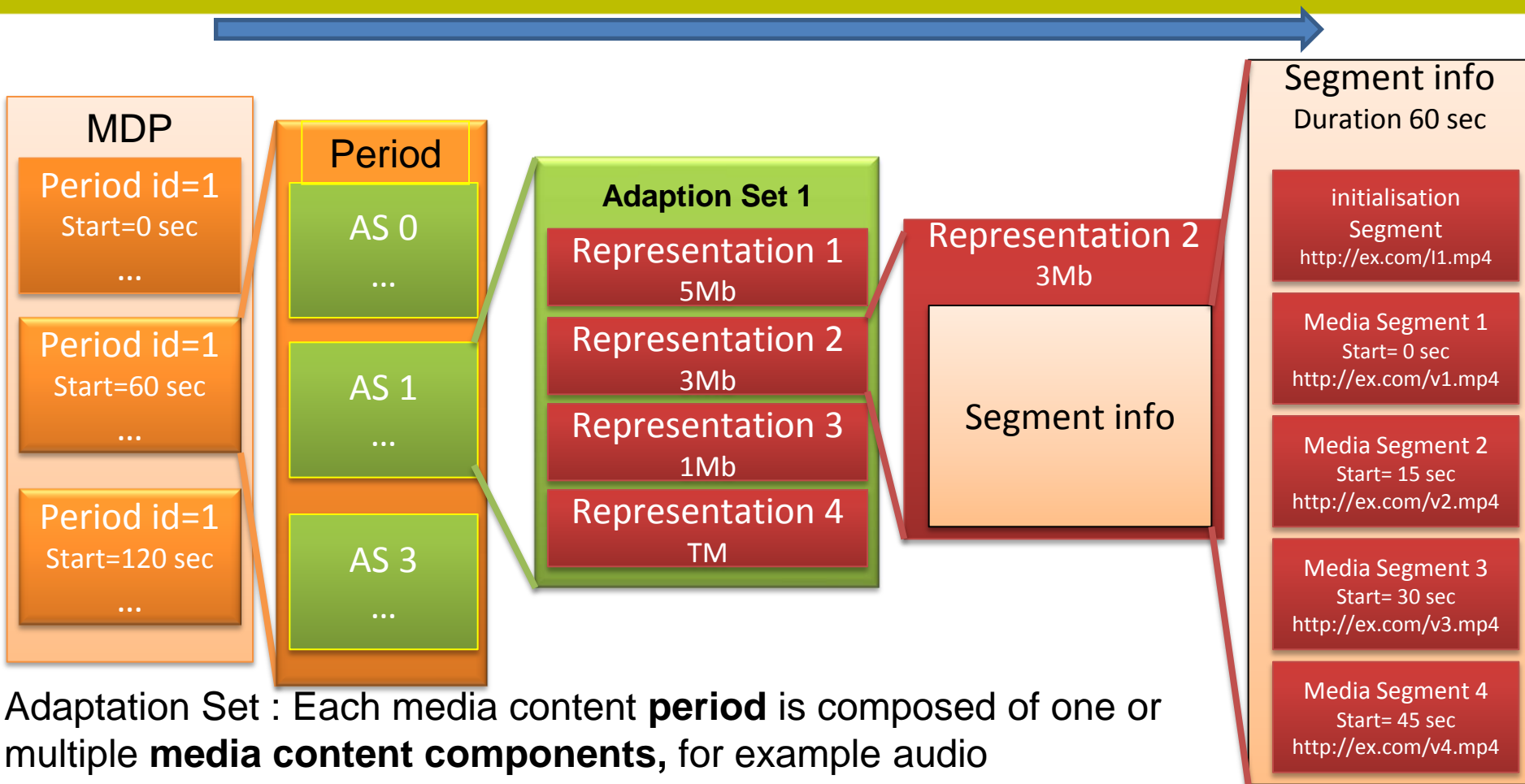


MPD hierarchical data model



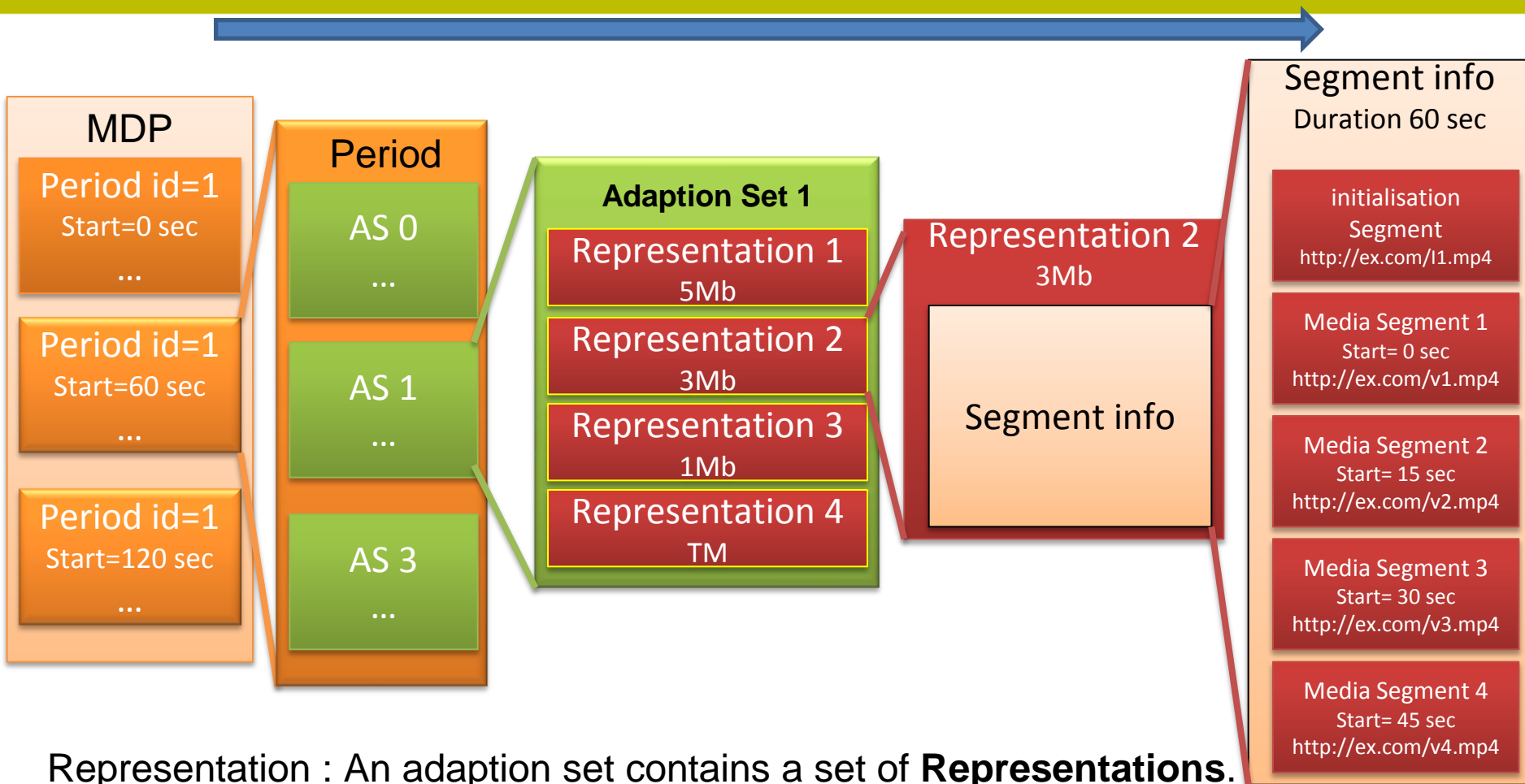
MDP : Media Presentation Description :
Media content is composed of a single or
multiple contiguous media content **periods**
in time.

MPD hierarchical data model



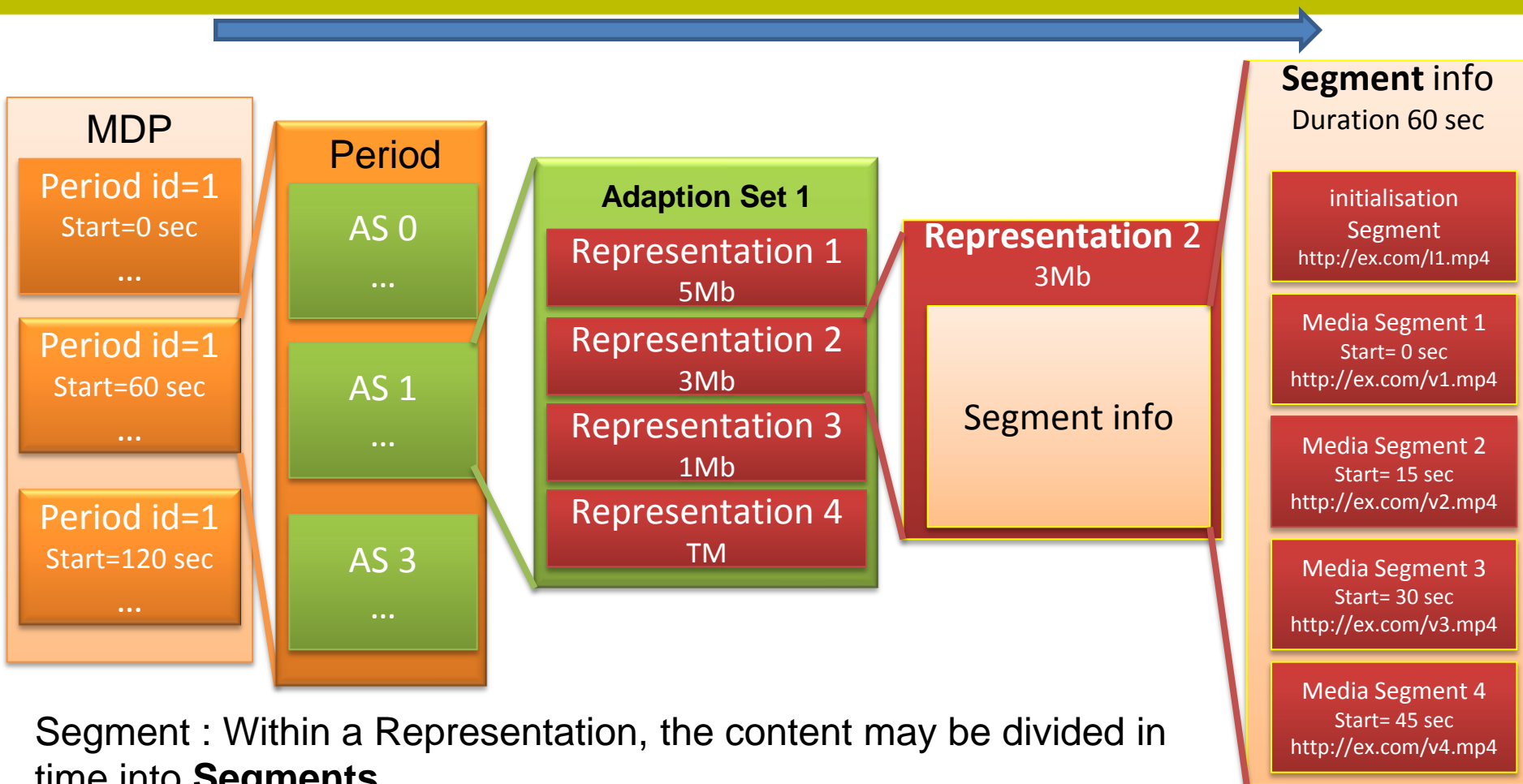
Adaptation Set : Each media content **period** is composed of one or multiple **media content components**, for example audio components in various languages and a video component. Each media content component has an assigned **media content component type**, for example audio or video.

MPD hierarchical data model



Representation : An adaption set contains a set of **Representations**. A Representation describes a *deliverable encoded version* of one or several media content components.

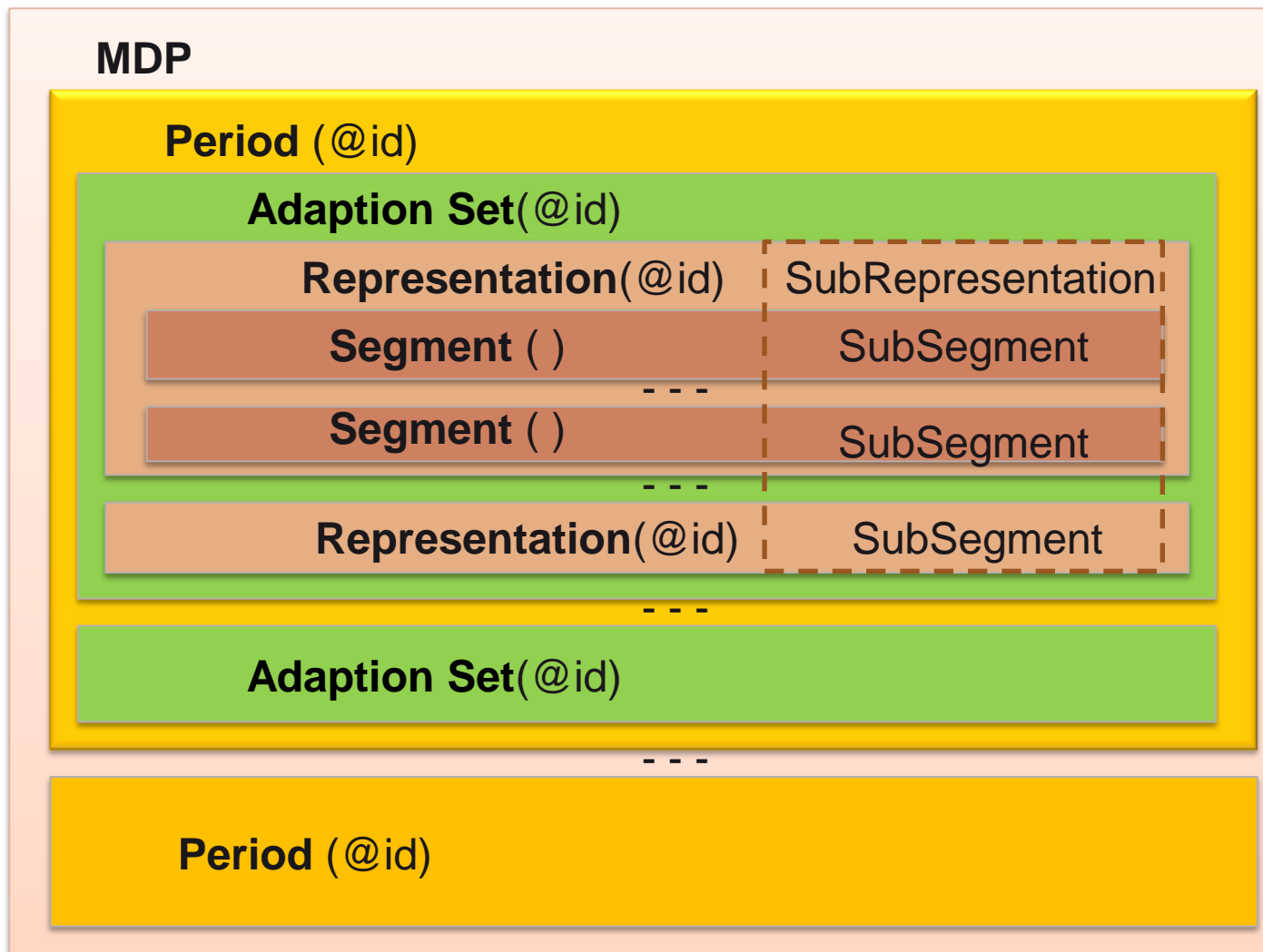
MPD hierarchical data model



Segment : Within a Representation, the content may be divided in time into **Segments**

MPD may also include a byte range with the URL : Segment is contained in the provided byte range of some larger resource

DASH High-Level Data Model





Segment description



Media Segments

- **ISO Base Media File Format as defined in ISO/IEC 14496-12**
- **MPEG-2 Transport Stream as defined in the ISO/IEC 13818-2 Format**

HTTP Live Streaming how to Encoding

Chunk length

2 sec for short sequence. (> 5mn).

6 sec for long sequence

Key frame (intra)

Every 2 sec

Independent chunk

Closed GOP

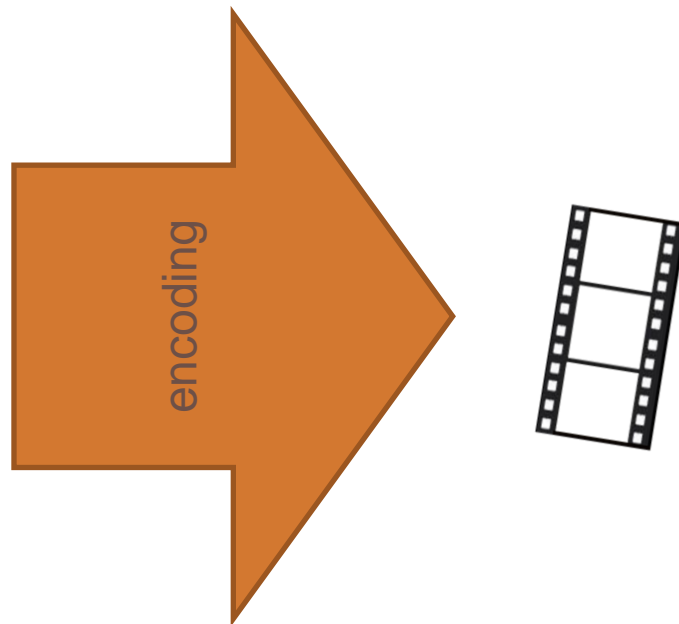
Rates example

512 kbps 288P

768 kbps 360P

1200 kbps 480P

2000 kbps 720P





Apple HTTP coding recommendation

Figure 1.

Apple provides very specific configuration recommendations in Technical Note TN2224.

HTTP Streaming Encode Recommendations 16:9 Aspect Ratio

	Dimensions	Total Bitrate	Video Bitrate	Keyframe	Restrict Profile to
CELL	480x320	64	NA	NA	NA
CELL	416x234	150	110	30	Baseline, 3.0
CELL	416x234	240	200	45	Baseline, 3.0
CELL	416x234	440	400	90	Baseline, 3.0
WI-FI	640x360	640	600	90	Baseline, 3.0
WI-FI	640x360	1240	1200	90	Main, 3.1
WI-FI	960x540	1840	1800	90	Main, 3.1
WI-FI	1280x720	2540	2500	90	Main, 3.1
WI-FI	1280x720	4540	4500	90	Main, 3.1



Consistent Audio Parameters

- 40Kbps audio for all streams. Most authorities recommend the same,
- audible **pops may occur when switching** between streams with different audio parameters.
- =>One set of parameters for audio.
- Concert example two sets of audio parameters,
 - one mono, one stereo
 - same sample rate and bit depth
 - Lower-quality 64Kbps/44 kHz/16-bit mono
 - Higher quality 128Kbps/44 kHz/16-bit stereo

HTTP Live Streaming how to mux A

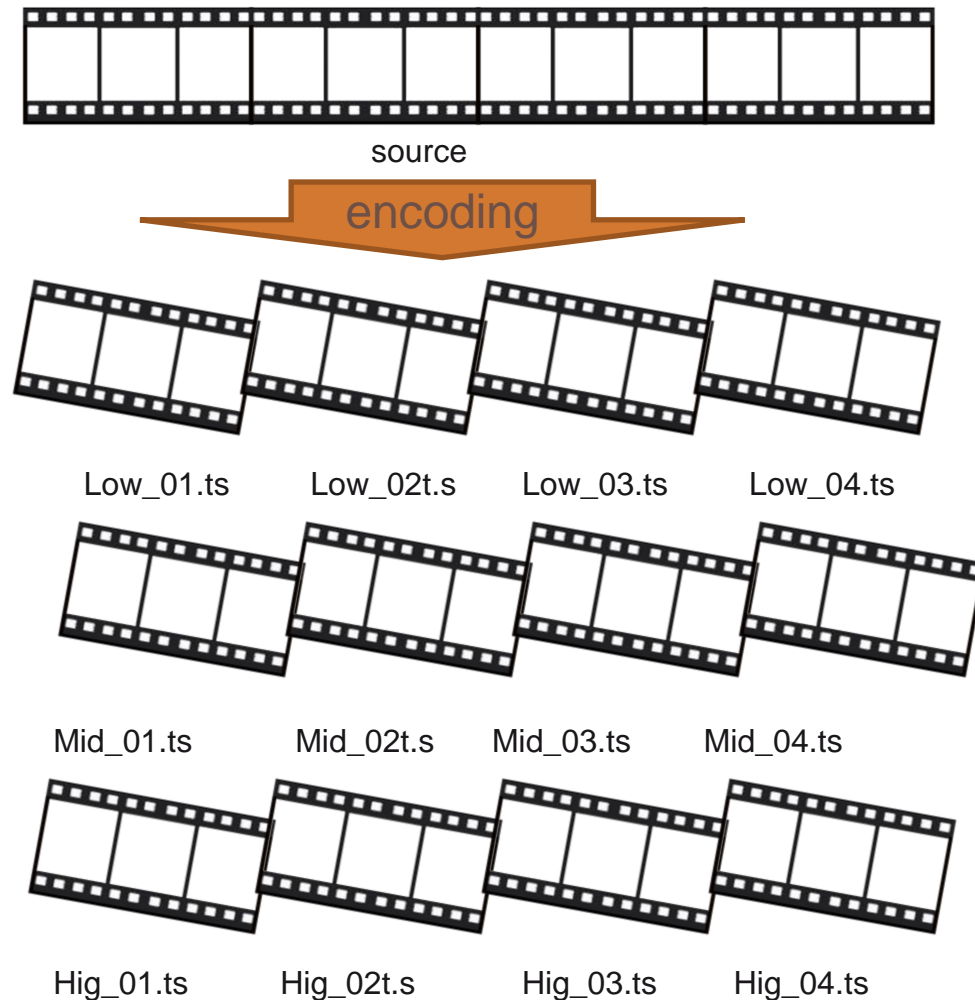
Encode the source into multiple files at different data rates

Ex 3: low midle high

Divide them into short chunks, usually between 5-10 seconds long.

Low_01.ts Low_02.ts
In closed GOP
(independent)

Here MPEG2 TS is used as format container *.ts



HTTP Live Streaming how to mux B

Encode the source into multiple files at different data rates

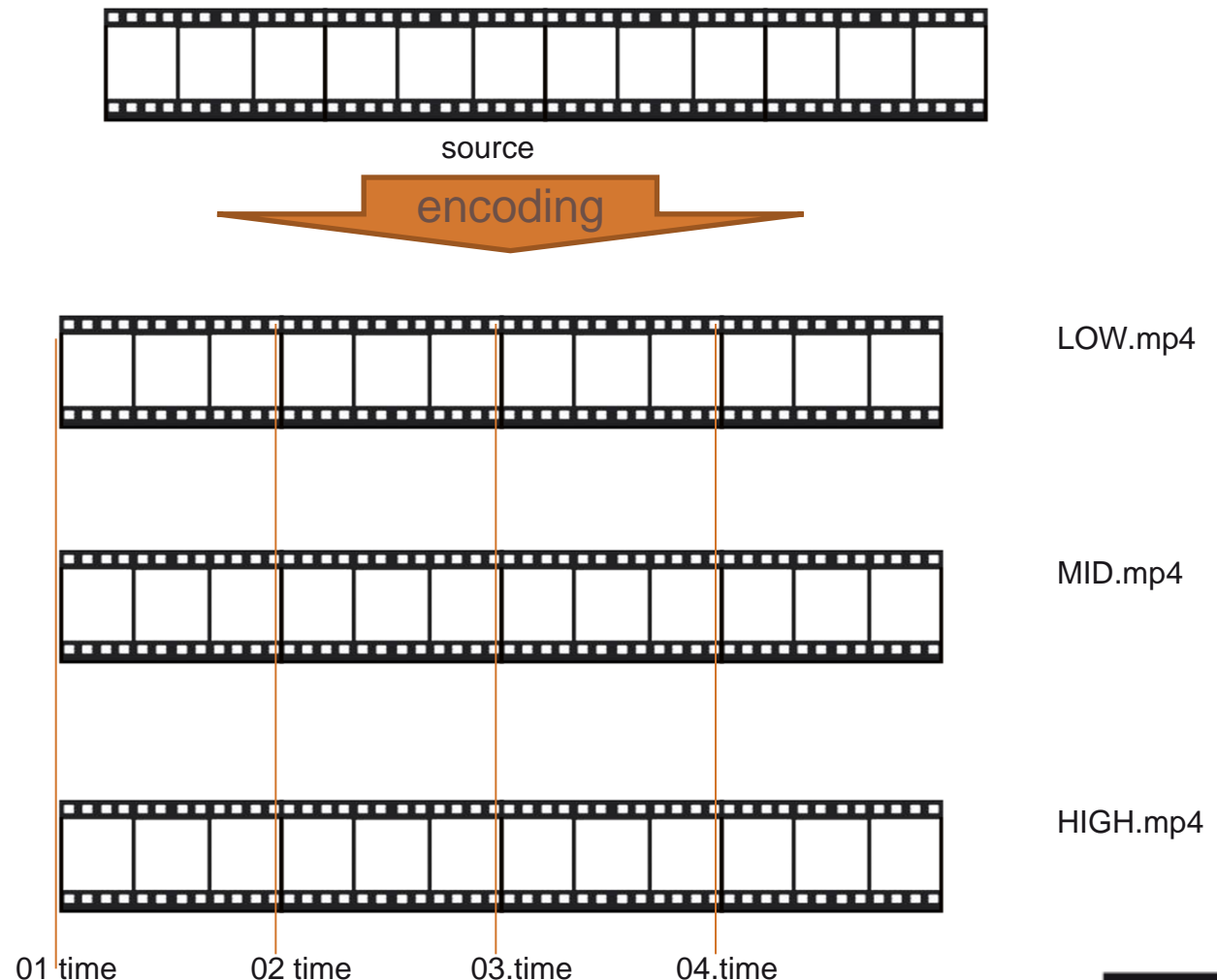
Ex 3: low middle high

Divide them into short chunks, usually between 5-10 seconds long.

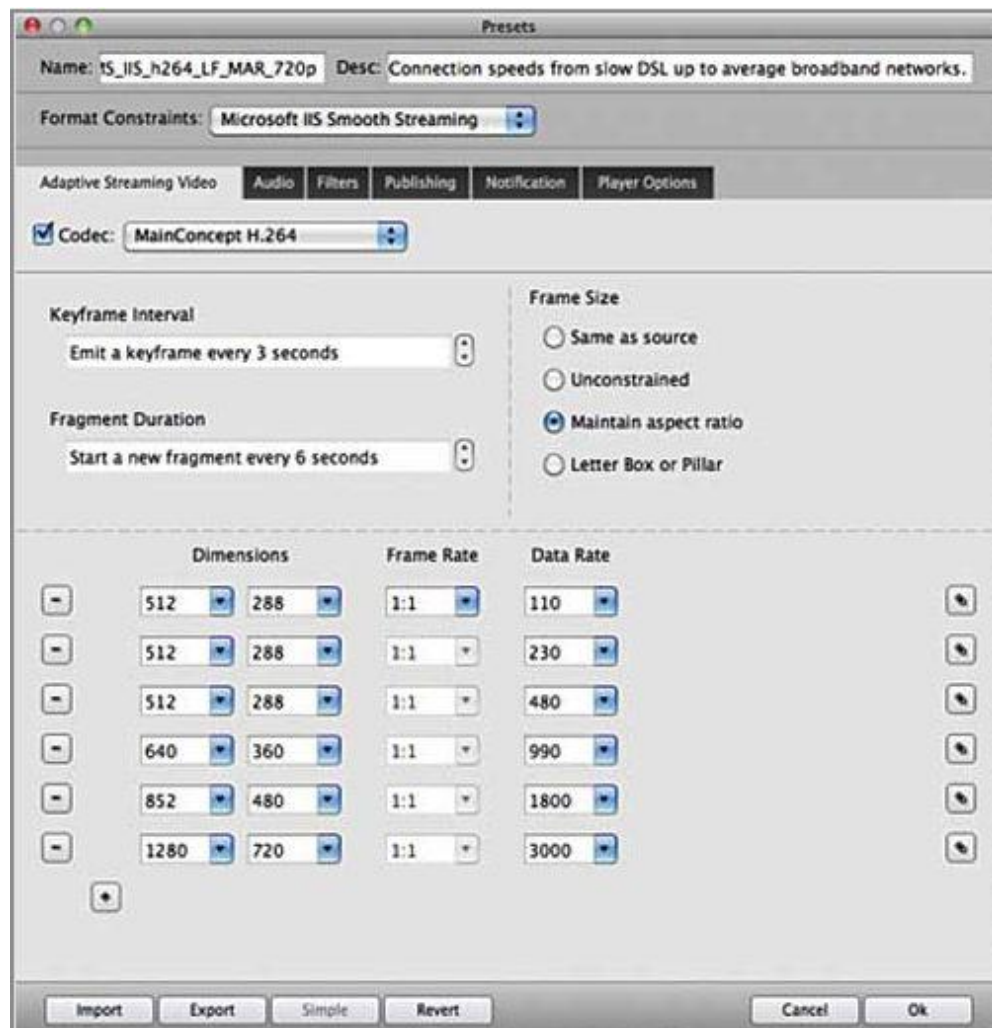
In closed GOP (independent)

Insert chunk with time in a file

File as MP4



Example configuration

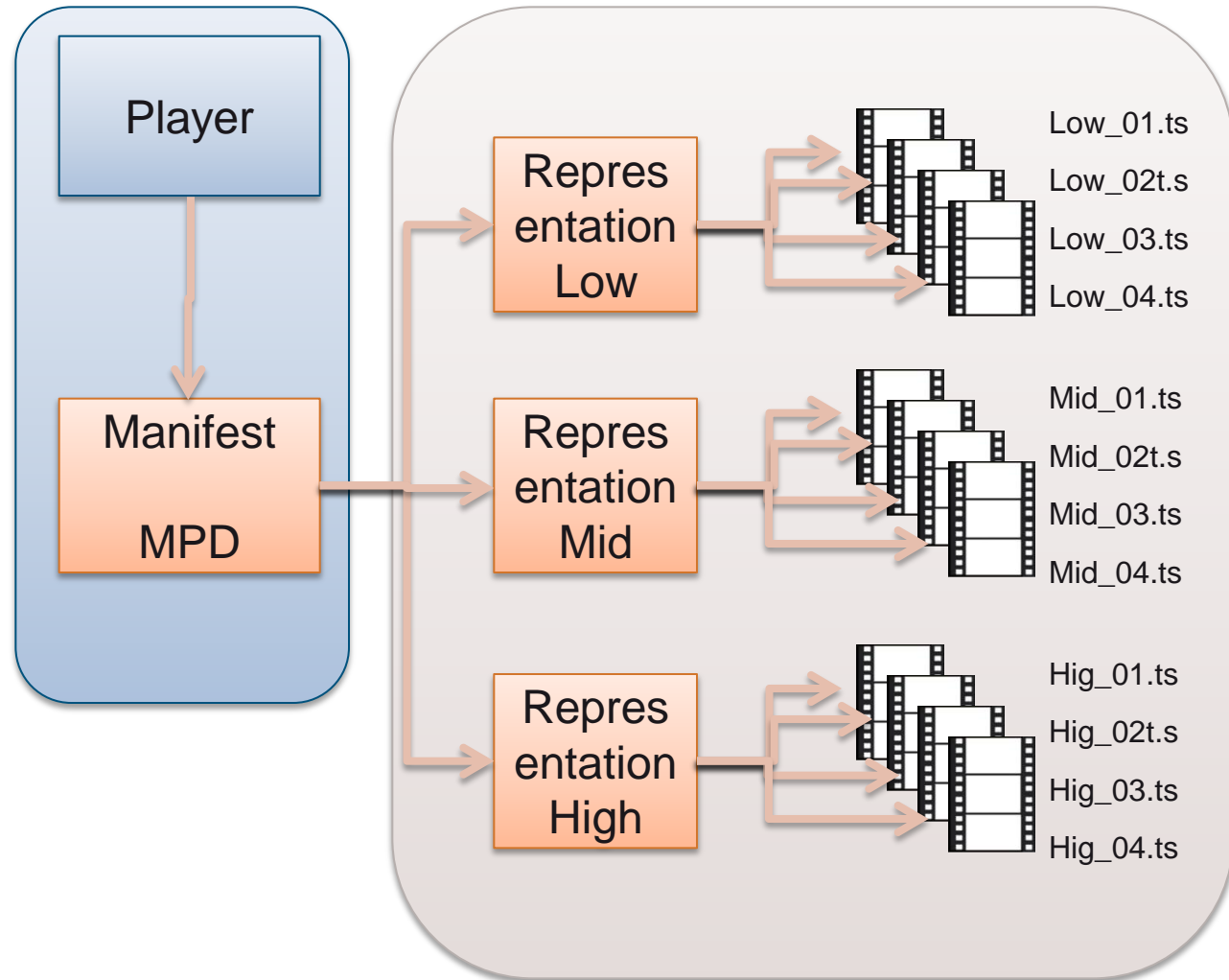


<http://www.streamingmediaglobal.com/Articles/Editorial/Featured-Articles/How-to-Produce-for-Adaptive-Streaming-81020.aspx>

HTTP Live Streaming MPD

Loaded chunk on an HTTP server along with a text-based **manifest file** :
Media Presentation Description, XML file
MPD

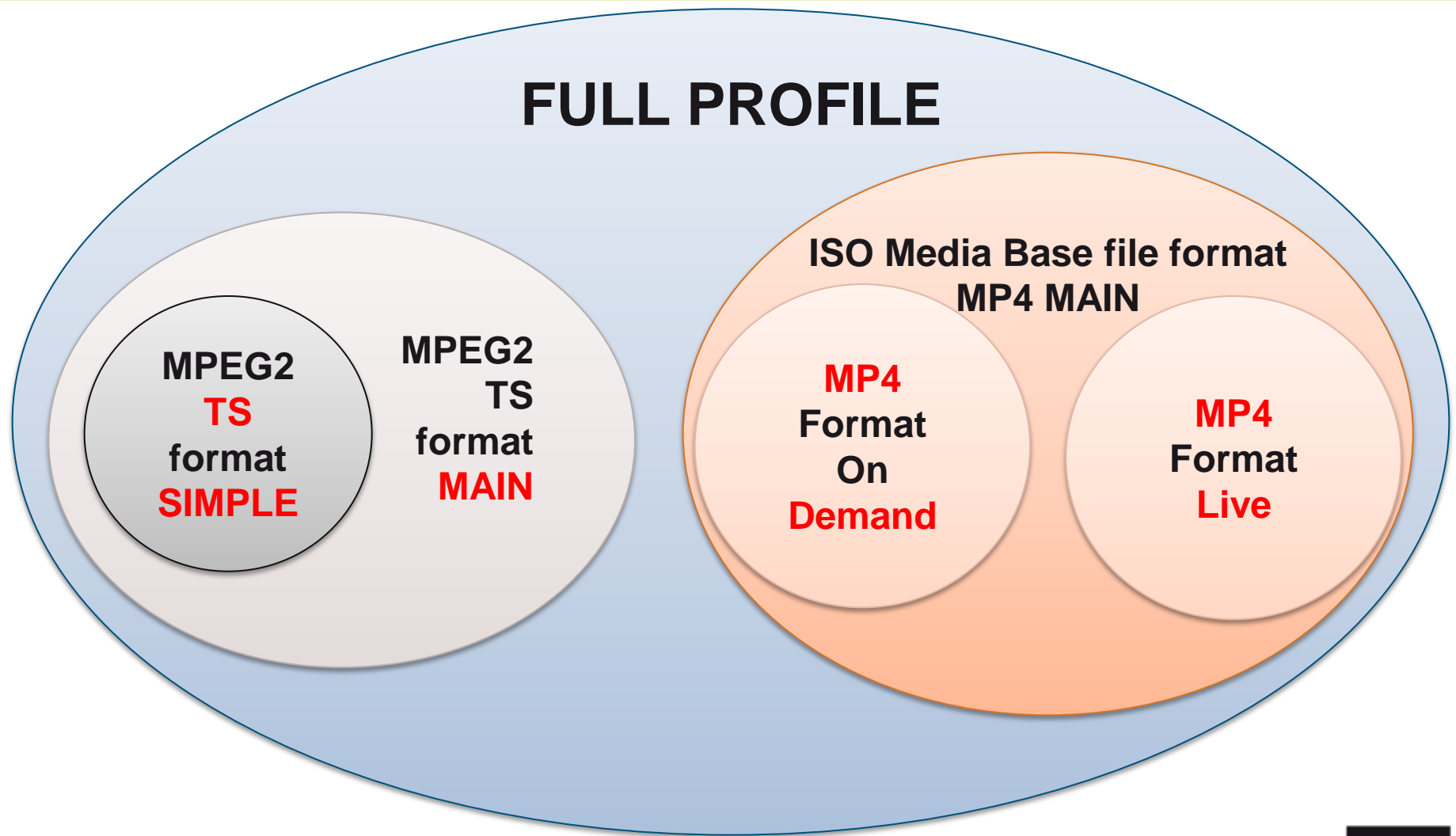
that directs the player to URL of each of the encoded chunk ***.ts**





Profiles

Profiles





Profile

- **Three profiles are defined relying on the ISO base media FF as Segment formats**
 - Main
 - On demand subset of Main
 - Live subset of Main
- **Two profiles are defined for MPEG-2 TS based Media Segment formats**
 - Main
 - Simple subset of main



Profile ISO base media FF on demand

- **One segment / representation**
- **Subsegment aligned in across representation within an adaptatation set**
- **Subsegment begin by Stream Access Points**
- **MPD@type shall be “static”**



Profile ISO base media FF Live

- **Small segment**
- **MPD@type shall be “Live”**

Example MPD for ISO Base media file format On Demand profile **MDP**

<?xml version="1.0"?>

<MPD

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns="urn:mpeg:DASH:schema:MPD:2011"

xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011 DASH-MPD.xsd"

type="static"

mediaPresentationDuration="PT3256S"

minBufferTime="PT1.2S"

profiles="urn:mpeg:dash:profile:isoff-on-demand:2011">

<BaseURL>http://cdn1.example.com/</BaseURL>

<BaseURL>http://cdn2.example.com/</BaseURL>

<Period>

</Period>

</MPD>

On-Demand **period**

<Period>

<!-- English Audio -->

<AdaptationSet mimeType="audio/mp4" codecs="mp4a.0x40" lang="en"
subsegmentAlignment="true" subsegmentStartsWithSAP="1">

</AdaptationSet>

<!-- French Audio -->

<AdaptationSet mimeType="audio/mp4" codecs="mp4a.40.2" lang="fr"
subsegmentAlignment="true" subsegmentStartsWithSAP="1">

</AdaptationSet>

<!-- Timed text -->

<AdaptationSet mimeType="application/ttml+xml" lang="de">

</AdaptationSet>

<!-- Video -->

<AdaptationSet mimeType="video/mp4" codecs="avc1.4d0228"

</AdaptationSet>

</Period>



On-Demand **audio**

<!-- English Audio -->

```
<AdaptationSet mimeType="audio/mp4" codecs="mp4a.0x40" lang="en"
  subsegmentAlignment="true" subsegmentStartsWithSAP="1">
  <ContentProtection schemeldUri="urn:uuid:706D6953-656C-5244-4D48-
656164657221"/>
  <Representation id="1" bandwidth="64000">
    <BaseURL>7657412348.mp4</BaseURL>
  </Representation>
  <Representation id="2" bandwidth="32000">
    <BaseURL>3463646346.mp4</BaseURL>
  </Representation>
</AdaptationSet>
```




On-Demand **text**

```
<!-- Timed text -->  
  <AdaptationSet mimeType="application/ttml+xml" lang="de">  
    <Role schemeldUri="urn:mpeg:dash:role" value="subtitle"/>  
    <Representation id="5" bandwidth="256">  
      <BaseURL>796735657.xml</BaseURL>  
    </Representation>  
  </AdaptationSet>
```

On-Demand video

```
<AdaptationSet mimeType="video/mp4" codecs="avc1.4d0228"
  subsegmentAlignment="true" subsegmentStartsWithSAP="2">
  <ContentProtection schemeldUri="urn:uuid:706D6953-656C-5244-4D48-
656164657221"/>
  <Representation id="6" bandwidth="256000" width="320" height="240">
    <BaseURL>8563456473.mp4</BaseURL>
  </Representation>
  <Representation id="7" bandwidth="512000" width="320" height="240">
    <BaseURL>56363634.mp4</BaseURL>
  </Representation>
  <Representation id="8" bandwidth="1024000" width="640" height="480">
    <BaseURL>562465736.mp4</BaseURL>
  </Representation>
  <Representation id="9" bandwidth="1384000" width="640" height="480">
    <BaseURL>41325645.mp4</BaseURL>
  </Representation>
  <Representation id="A" bandwidth="1536000" width="1280" height="720">
    <BaseURL>89045625.mp4</BaseURL>
  </Representation>
  <Representation id="B" bandwidth="2048000" width="1280" height="720">
    <BaseURL>23536745734.mp4</BaseURL>
  </Representation>
</AdaptationSet>
```



ISO Base media file format Live profile

```
<?xml version="1.0"?>
```

```
<MPD
```

```
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xmlns="urn:mpeg:DASH:schema:MPD:2011"
```

```
  xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011 DASH-  
MPD.xsd"
```

```
  type="dynamic"
```

```
  minimumUpdatePeriod="PT2S"
```

```
  timeShiftBufferDepth="PT30M"
```

```
  availabilityStartTime="2011-12-25T12:30:00"
```

```
  minBufferTime="PT4S"
```

```
  profiles="urn:mpeg:dash:profile:isoff-live:2011">
```

```
    <BaseURL>http://cdn1.example.com/</BaseURL>
```

```
    <BaseURL>http://cdn2.example.com/</BaseURL>
```

```
      <Period>...</Period>
```

```
</MPD>
```

ISO Base media file format Live profile

PERIOD

<Period>

<!-- Video -->

<AdaptationSet mimeType="video/mp4" codecs="avc1.4D401F"
frameRate="30000/1001" segmentAlignment="true"
startWithSAP="1">

</AdaptationSet>

<!-- English Audio -->

<AdaptationSet mimeType="audio/mp4" codecs="mp4a.0x40" lang="en"
segmentAlignment="0" startWithSAP="1">

</AdaptationSet>

<!-- French Audio -->

<AdaptationSet mimeType="audio/mp4" codecs="mp4a.0x40" lang="fr"
segmentAlignment="0" startWithSAP="1">

</AdaptationSet>

</Period>

ISO Base media file format Live profile

video

```
<!-- Video -->
<AdaptationSet mimeType="video/mp4" codecs="avc1.4D401F"
  frameRate="30000/1001" segmentAlignment="true" startWithSAP="1">
  <BaseURL>video/</BaseURL>
  <SegmentTemplate timescale="90000"
    initialization="$Bandwidth%/init.mp4v"
    media="$Bandwidth%/$Time$.mp4v">
    <SegmentTimeline> <S t="0" d="180180" r="432"/>
    </SegmentTimeline>
  </SegmentTemplate>
  <Representation id="v0" width="320" height="240"
    bandwidth="250000"/>
  <Representation id="v1" width="640" height="480"
    bandwidth="500000"/>
  <Representation id="v2" width="960" height="720"
    bandwidth="1000000"/>
</AdaptationSet>
```

ISO Base media file format Live profile

audio

<!-- English Audio -->

```
<AdaptationSet mimeType="audio/mp4"
codecs="mp4a.0x40" lang="en" segmentAlignment="0"
startWithSAP="1">
```

```
  <SegmentTemplate timescale="48000"
initialization="audio/en/init.mp4a"
media="audio/en/$Time$.mp4a">
```

```
    <SegmentTimeline>
```

```
      <S t="0" d="96000" r="432"/>
```

```
    </SegmentTimeline>
```

```
  </SegmentTemplate>
```

```
    <Representation id="a0" bandwidth="64000" />
```

```
</AdaptationSet>
```



Functionalities

Acces point
MDP Update



Functionalities

Regulation

HTTP Bitrate Regulation



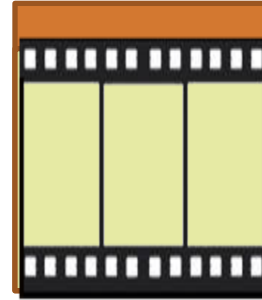
Chunk
Level 3



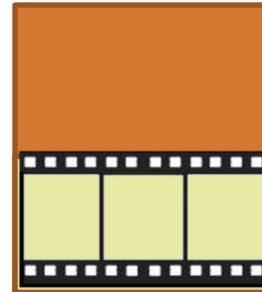
Chunk
Level 2



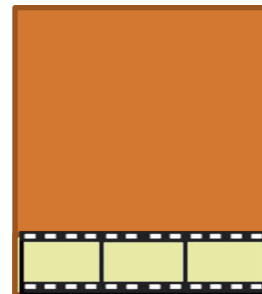
Chunk
Level 2



Fast reception
Go lower level



Fine reception
Keep level



slow reception
Go upper level

Decoder Buffer



Functionalities

Access Point

Resynchronization
Zapping

- **Stream Access Point are defined to enable seamless stream switching**
- **The type of SAP is dependent only on which Access Units are correctly decodable**
- **Type 1 => “Closed GoP random access point” low delay**
- **Type 2 => “Closed GoP random access point” classical**
- **Type 3 => “Open GoP random access point”,**
- **Type 4 => “Gradual Decoding Refresh (GDR) random access point”**
- **Type 5 ...**
- **Type 6 ...**

SAP Media stream access points

- **Type 1 « closed GOP »**
- $I1_{Isap}$ P2 P2
- **First Decoding order Isap = first presentation order**
- Decoding order assume good continuous presentation
- No ref to anterior access unit
- **Type 2 « closed GOP »**
- Dec order : $I1_{Isau}$ B0 P3 B2 P4
- Pres order : B0 I1 B2 P3
- **First Decoding order Isau \neq first presentation order**
- Decoding order assume good presentation
- No ref to anterior access unit for B0

SAP Media stream access points

■ Type 3 « open GOP »

■ Dec order : $I1_{Isau}$ B0 P3 B2 P4

■ Pres order : B0 $I1_{Tsap}$ B2 P3

■ First Decoding order $Isau \neq$ first presentation order

■ Not correct Decoding of B0

■ Ref to anterior access unit for B0

■ Type 4 « Gradual Decoding Refresh (GDR) » multiple reference frame HEVC

■ Dec order : $I1_{Isau}$ B0 P3 B2 P4

■ Pres order : B0 $I1_{Tsap}$ B2 P3

■ First Decoding order $Isau \neq$ first presentation order

■ Not correct Decoding of B0 P3...

■ Ref to anterior access unit for B0



Functionalities

MDP Update

Modification of content location



- **The server may update the MPD document during the Media Presentation**
- **Media Presentation is described by a MPD including any possible updates of the MPD**



MDP update Parameters

- **MPD@type** set to 'dynamic', the MPD may be updated during the Media Presentation
- **MPD@minimumUpdatePeriod** :
 - Provides the minimum period the containing MPD document is updated.



MDP update client side

- If MPD@type set to 'dynamic'
- If the MPD@minimumUpdatePeriod attribute in the client is provided, then the check time is defined as the sum of the fetch time of this operating MPD and the value of this attribute, i.e.
- $CheckTime = FetchTime + MPD@minimumUpdatePeriod$



Functionalities

alternatives sources

content available from differents sources



Alternatives sources

- URLs at each level of the MPD are resolved according to RFC3986 :

MPD Period AdaptationSet Representation

- Ex : content available from two sources (cdn1 and cdn2)

<BaseURL><http://cdn1.example.com/></BaseURL>

<BaseURL><http://cdn2.example.com/></BaseURL>



Functionalities

Protection



Protection

- **Content Descriptors for Protection, Accessibility, Rating, etc.**
- **Signaling, delivery, utilization of multiple DRM schemes.**



Protection MP4 and TS Chunk

■ MP4

Representations based on ISO/IEC14496-12

urn:mpeg:dash:mp4protection:2011

@value attribute shall be the 4CC contained in the
Scheme Type Box of the **Protection Scheme**
Information Box

■ TS

■ Representations based on ISO/IEC 13818-1 (MPEG-2 Transport Stream),

indicate the **Conditional Access System** used:

urn:mpeg:dash:13818:1:CA_descriptor:2011

????????

■ MP4 PROTECTION

Representations based on ISO/IEC14496-12 a content protection scheme using the Protection System Specific Header Box defined in ISO/IEC 23001-7 may be identified in the **ContentProtection element**.

UUID URN as defined in RFC 4122 indicating the UUID specified in the SystemId field of the Protection System Specific Header Box shall be used. This does not imply that such schemes cannot define alternative URNs, or that all UUID URNs refer to schemes of this type.



Some problems...?

1. Chunk size / Number of objects / Number of log entries
2. Stateless delivery loss of: session reporting / streams reporting / denying access by token to chunk URL
3. Caching servers were never designed for live streaming
4. No QoS guarantee
5. Overloading networks
6. regulation





References

<http://dashpg.com/>

VLC DASH Player

- <http://www-itec.uni-klu.ac.at/dash/>



- <http://gpac.wp.institut-telecom.fr/>

<http://www.videolan.org/>





References

- **Short Tutorial on MPEG-DASH ISO/IEC JTC1/SC29/WG11 N12340 Iraj Sodagar Microsoft Corporation**
- **MPEG-DASH: The Standard for Multimedia Streaming Over Internet Iraj Sodagar Microsoft Corporation**
- **ISO/IEC DIS 23009-1 Information technology — Dynamic adaptive streaming over HTTP (DASH) — Part 1: Media presentation description and segment formats**
- **ISO/IEC FCD 23001-6 Information technology — MPEG systems technologies — Part 6: Dynamic adaptive streaming over HTTP (DASH)**

http streaming reference

- *RTMP Flash*
- Kapoor, Abhinav. ["Live Dynamic Streaming With Flash Media Server 3.5."](#) Adobe.com.
Bouthillier, Larry. ["How to Do Dynamic Streaming With Flash Media Server."](#) StreamingMedia.com.
["Tutorial: On-Demand HTTP Dynamic Streaming."](#) Adobe.com.
["Encoding Guidelines Dynamic Streaming for Flash Over HTTP."](#) Akamai.com.*HTTP Live Streaming*
- Apple Tech Note: ["Best Practices for Creating and Deploying HTTP Live Streaming Media for the iPhone and iPad."](#) Apple.com.
Apple Tech Note: ["HTTP Live Streaming Overview."](#) Apple.com.
Apple Tech Note: ["Using HTTP Live Streaming."](#) Apple.com.*Smooth Streaming*
- ["Akamai HD for Microsoft Silverlight On-Demand Encoding Recommendations."](#) Akamai.com.
Zambelli, Alex. ["IIS Smooth Streaming Technical Overview."](#) Microsoft.com.
Zambelli, Alex. Webinar: ["Transcoding Methodology 101."](#) StreamingMedia.com.*HTTP Flash*
- ["Encoding Guidelines for Dynamic Streaming for Flash Over HTTP."](#) Akamai.com.
Levkov, Maxim. ["Video Encoding and Transcoding Recommendations for HTTP Dynamic Streaming on the Flash Platform."](#) Adobe.com.