

MPEG DASH

ONE MANIFEST FOR ALL

COGICOM



Clement Duval

Previously on



COGICOM

PREREQUISITES

- ▶ Have followed every courses prior to this one
- ▶ Especially HLS for the bandwidth calculation, throughput, bitrate witching

ANALYZE AN MPD

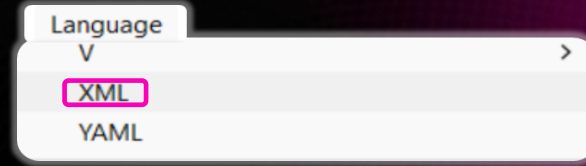
EFFECTIVELY

COGICOM

BROWSER CONSOLE

COGICOM

NOTEPAD++: OPEN THE MPD AS AN XML FILE



COGICOM

APPLE'S HLS: MOST POPULAR

	HLS	MPEG DASH
Technology	HTTP Adaptive Streaming	
Standardization		
OS native support		

APPLE'S HLS: MOST POPULAR

	HLS	MPEG DASH
Technology	HTTP Adaptive Streaming	
Standardization	Proprietary	Open Standard
OS native support		

APPLE'S HLS: MOST POPULAR

	HLS	MPEG DASH
Technology	HTTP Adaptive Streaming	
Standardization	Proprietary	Open Standard
OS native support	Any	Any browser/OS but... 

COGICOM

Most popular!

MPEG DASH: MOST EFFICIENT

	HLS	MPEG DASH
Manifest	One master & 2+ variants	One manifest
Addressing		

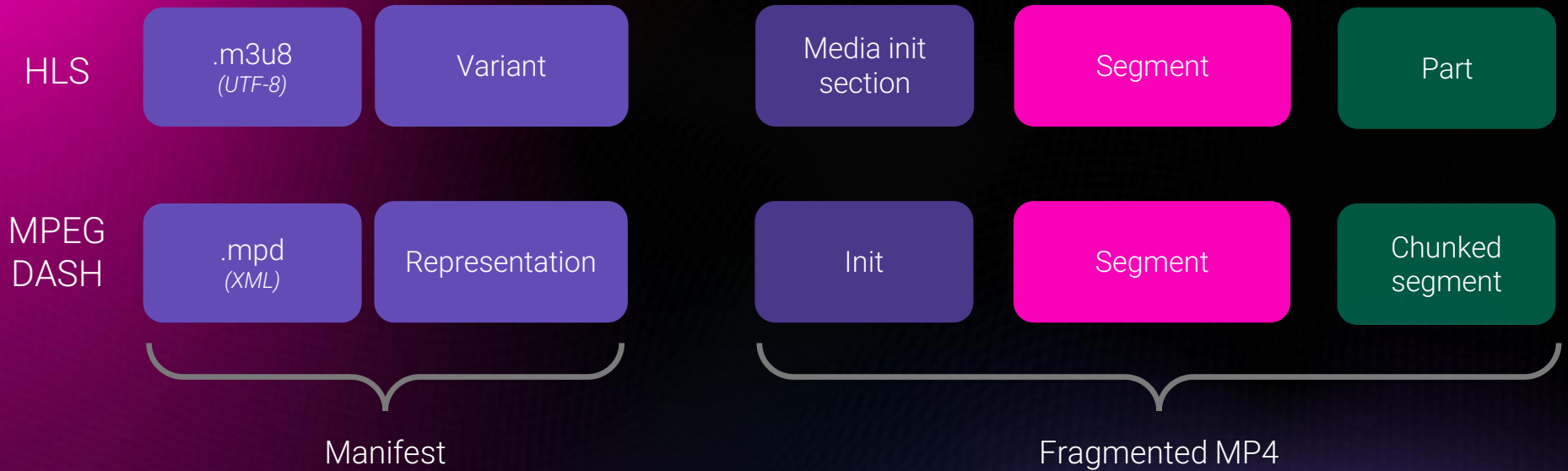
MPEG DASH: MOST EFFICIENT

	HLS	MPEG DASH
Manifest	One master & 2+ variants	One manifest
Addressing	Explicit	Implicit

Most efficient

COGICOM

HLS vs MPEG DASH: DATA MODEL

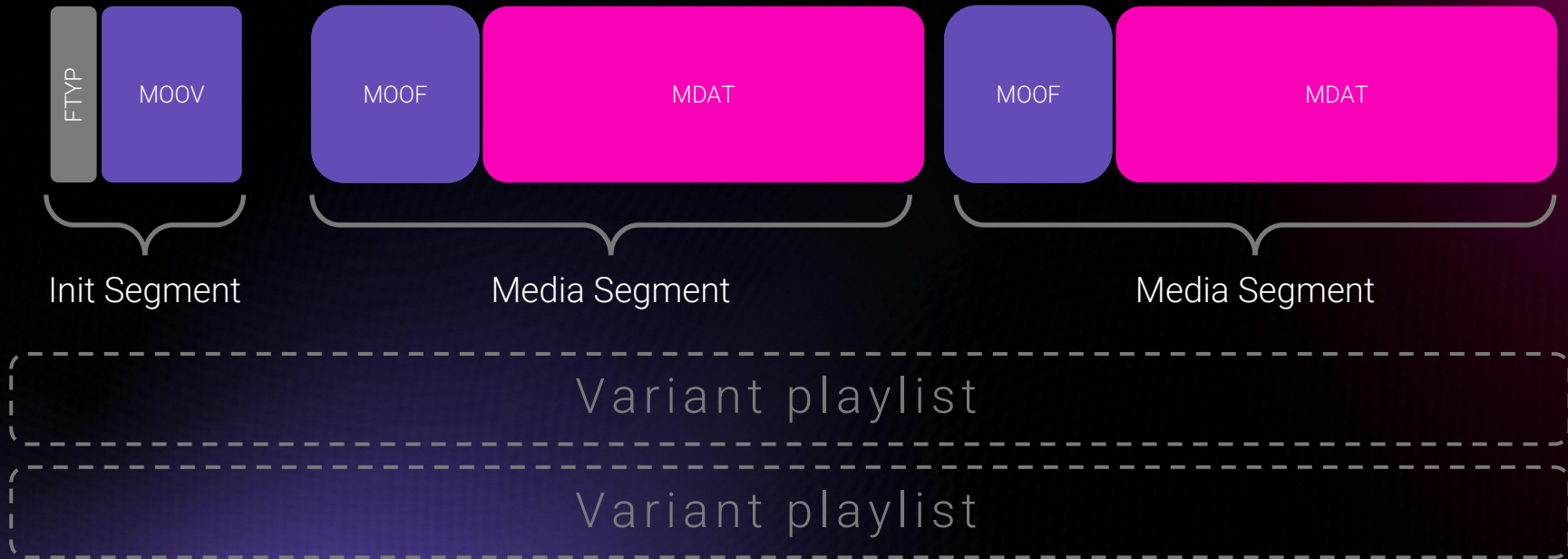


DASH IMPLICIT ADDRESSING

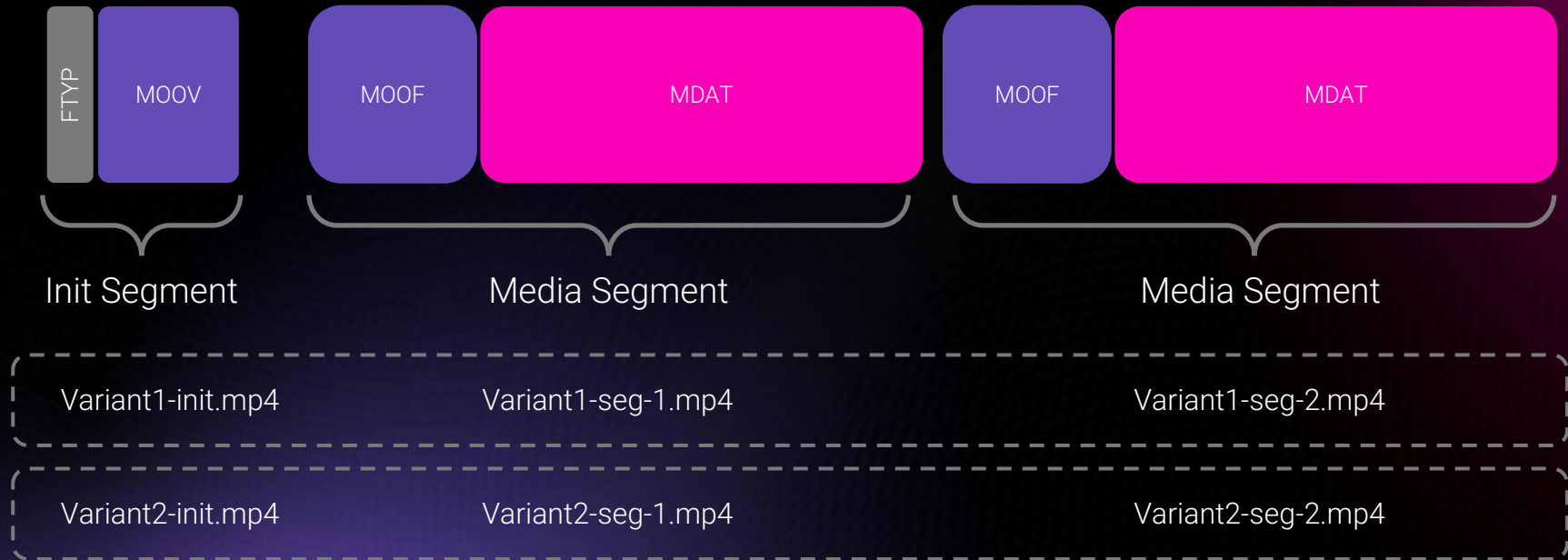
VS HLS EXPLICIT ADDRESSING

COGICOM

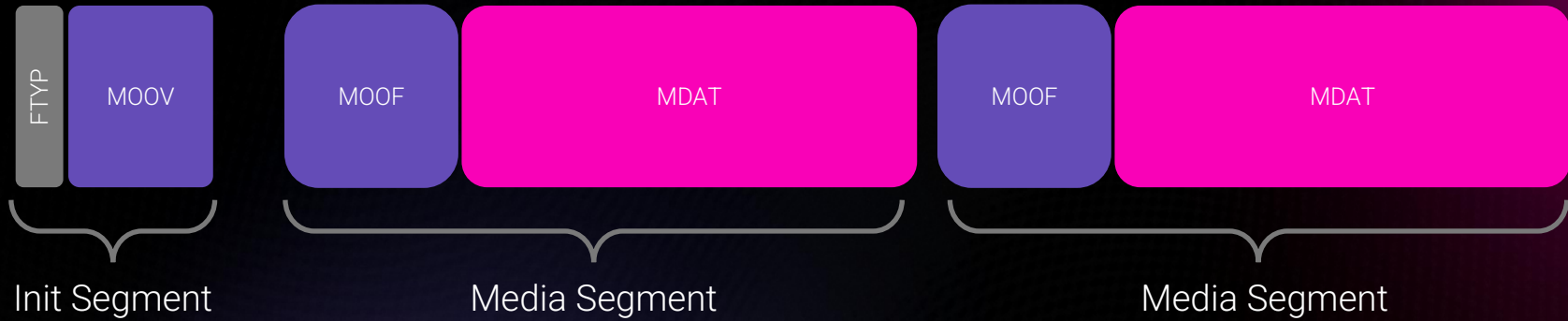
HOW HLS USES EXPLICIT ADDRESSING



HOW HLS USES EXPLICIT ADDRESSING



HOW HLS USES EXPLICIT ADDRESSING



What if MPEG DASH used the same explicit addressing for all segments from all representations into a single manifest?

HOW HLS USES EXPLICIT ADDRESSING

What if MPEG DASH used the same explicit addressing for all segments from all representations into a single manifest?

10 Video
Representations

2 Audio
Representations

2 second segments

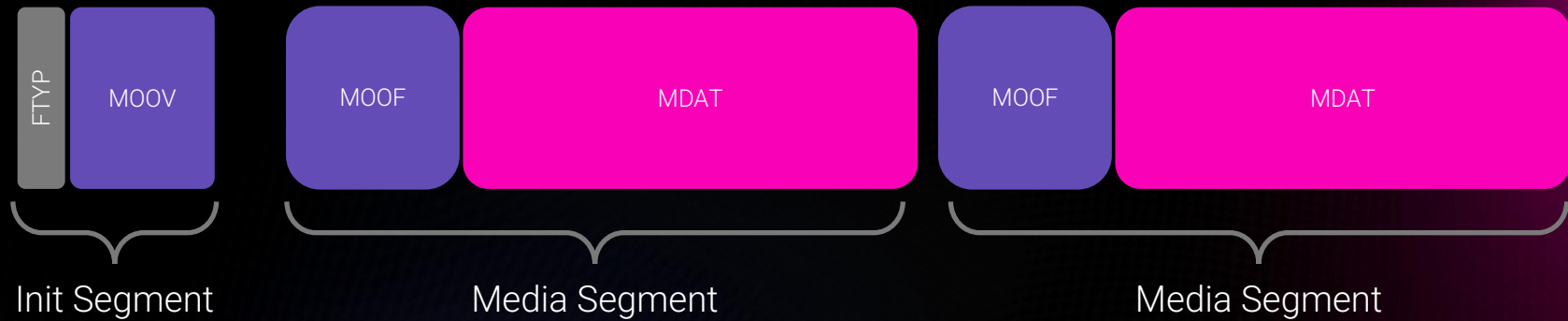
5 min live window size

Way too big !

=

1800 segment's URL
in the manifest!

MPEG DASH & IMPLICIT ADDRESSING



MPEG DASH uses **templates** & **variables**
to **compute/build** URLs

Implicit
addressing

THE DASH MANIFEST STRUCTURE

MEDIA PRESENTATION DESCRIPTION

COGICOM

THE MPD NESTED STRUCTURE



MPD XML STRUCTURE

Segment Template

Segment Timeline <S

Representation

Representation

Representation

THE DASH REPRESENTATION

DECLARING THE "VARIANTS"

COGICOM

THE DASH REPRESENTATION

```
<Representation id="720p" width=... height=... frameRate=... bandwidth=... codecs=.../>  
<Representation id="540p" width=... height=... frameRate=... bandwidth=... codecs=.../>  
<Representation id="270p" width=... height=... frameRate=... bandwidth=... codecs=.../>
```

THE DASH REPRESENTATION ID

```
<Representation id="720p"  
width="1280"  
height="720"  
frameRate="25/1"  
bandwidth="2499968"  
codecs="avc1.4D401F"/>
```

- ▶ ID (name) of the representation
- ▶ Could be anything !
- ▶ Usually long string of numbers!



REPRESENTATION ATTRIBUTES FOR VIDEO

```
<Representation id="720p"  
width="1280"  
height="720"  
frameRate="25/1"  
bandwidth="2499968"  
codecs="avc1.4D401F"/>
```

- ▶ High level information about the stream
- ▶ Player doesn't need to download/parse the init segment to know what type of media it is



THE BANDWIDTH ATTRIBUTE

```
<Representation id="720p"  
width="1280"  
height="720"  
frameRate="25/1"  
bandwidth="2499968"  
codecs="avc1.4D401F"/>
```

- ▶ Minimum download throughput to use this representation
- ▶ Contrary to HLS, DASH only declares the throughput of the current representation (video segments here)



THE CODECS ATTRIBUTE

```
<Representation id="720p"  
width="1280"  
height="720"  
frameRate="25/1"  
bandwidth="2499968"  
codecs="avc1.4D401F"/>
```

- ▶ Codec identifier
- ▶ Player doesn't need to download/parse the init segment to know if it is able to play this media



THE SEGMENT TEMPLATE

TEMPLATING THE URLs

COGICOM

```
<SegmentTemplate  
  timescale="10"  
  initialization="init_${RepresentationID$.mp4">  
  media="video_${RepresentationID$_$Time$.mp4"
```

DURATIONS & TIME EXPRESSED IN TIMESCALE UNITS

- ▶ 10 units per second
- ▶ 10 seconds becomes 100 time-units



```
<SegmentTemplate  
  timescale="10"
```

```
  initialization="init_${RepresentationID$.mp4">  
  media="video_${RepresentationID}_${Time$.mp4"
```

- ▶ Measures time inside the MPD
- ▶ Usually (way) more than the sample granularity of the media to measure

DURATIONS & TIME EXPRESSED IN TIMESCALE UNITS

```
<SegmentTemplate  
  timescale="10"  
  initialization="init_RepresentationID$.mp4">  
  media="video_RepresentationID$_Time$.mp4"
```

Coming up!

COGICOM

Live Demo

- ▶ Verify timescale used on *dash_cmaf_course1*



Quiz 1

COGICOM

BUILDING THE INIT SEGMENT URL

WITH INIT TEMPLATE & REPRESENTATION

COGICOM

```
<SegmentTemplate  
initialization="init_video_ $RepresentationID$.mp4"
```

```
<SegmentTemplate  
  initialization="init_video_<b>$RepresentationID$</b>.mp4"
```

```
<Representation id="<b>720p</b>" />
```

```
<Representation id="<b>540p</b>" />
```

Representation id
720p



http://BASE_URL/init_video_720p.mp4

Representation id
540p



http://BASE_URL/init_video_540p.mp4

Live Demo

- ▶ Verify the init segment URLs can be downloaded
- ▶ Verify that it is indeed an init segment (CMAF Header)



PATH NAVIGATION

SAVING SPACE IN THE MPD

COGICOM

LINUX LIKE PATH NAVIGATION



`./file` or `file:` → File is in the current directory

`./sub/file` → File is in the subdirectory of the current directory

`../file` → File is in the parent directory (1 level up)

`../../file` → File is in the parent directory (2 levels up)

`/file` → File is in the root directory

`http://other_webserver/file` → File is on a different web server's root directory

FETCHING THE MPD FROM THE ROOT DIRECTORY

`http://origin_server/dash1`

`http://origin_server/dash1/manifest.mpd`

- ▶ Relative paths in the .mpd are all relative to this **root directory**

```
<SegmentTemplate  
  media="video/video_${RepresentationID}_${Time$.mp4"  
  initialization="video/init_video_${RepresentationID$.mp4
```

http://origin_server/dash1/video

video/init_video_720p.mp4

video/video_720p_\${time}.mp4

- ▶ Video media & init segments are in the **video directory** inside the **root directory**

```
<SegmentTemplate  
  media="audio/audio_${RepresentationID}_${Time$.mp4"  
  initialization="audio/init_audio_${RepresentationID$.mp4
```

http://origin_server/dash1/audio

audio/init_audio_en.mp4

audio/audio_en_\${time}.mp4

- ▶ Video media & init segments are in the **audio directory** inside the **root directory**

Live Demo

- ▶ Identify the root directory
- ▶ Identify where audio & video segments are pulled from
- ▶ See the .mpd syntax used to provide relative paths



THE ADAPTATION SET

GROUPING WHAT CAN BE “BITRATE SWITCHED”

COGICOM

EVERYTHING INSIDE AN ADAPTATION SET **HAS THE SAME
CONTENT BUT WITH DIFFERENT BITRATE VERSIONS**
**PLAYERS CAN SWITCH BETWEEN THESE VERSIONS TO
ADAPT TO BANDWIDTH CHANGES**

COGICOM

THE ADAPTATION SET

► bitrate switching for the same content

Players can do bitrate switching within an adaptation set

- Same content
- Same codec
- Same audio layout

```
<Period
```

```
  <AdaptationSet ... mimeType="video/mp4"
```

```
    <Representation id="1" bandwidth="8000000"
    <Representation id="2" bandwidth="4000000"
    <Representation id="3" bandwidth="2000000"
```

```
  <AdaptationSet ... mimeType="audio/mp4" lang="es"
```

```
    <Representation id="4" bandwidth="129296"
    <Representation id="5" bandwidth="262000"
```

TWO AUDIO ENG FRENCH

COGICOM

GETTING THE RIGHT SEGMENTS..

...AT THE RIGHT TIME

COGICOM

REQUESTING THE RIGHT SEGMENT AT THE RIGHT TIME

```
<SegmentTemplate  
media="video_720p_$Time$.mp4"
```

OTT
PACKAGER

video_720p_0.mp4
video_720p_40.mp4
video_720p_80.mp4

How can DASH players get the right segment URL at the right time?

HTTP GET http://...video_720p_0.mp4

HTTP GET http://...video_720p_40.mp4

HTTP GET http://...video_720p_80.mp4



DASH Player

COGICOM

THE \$TIME PRINCIPLE

SYNCHRONIZATION PLAYERS & ORIGIN PACKAGERS



Network
Time
Protocol

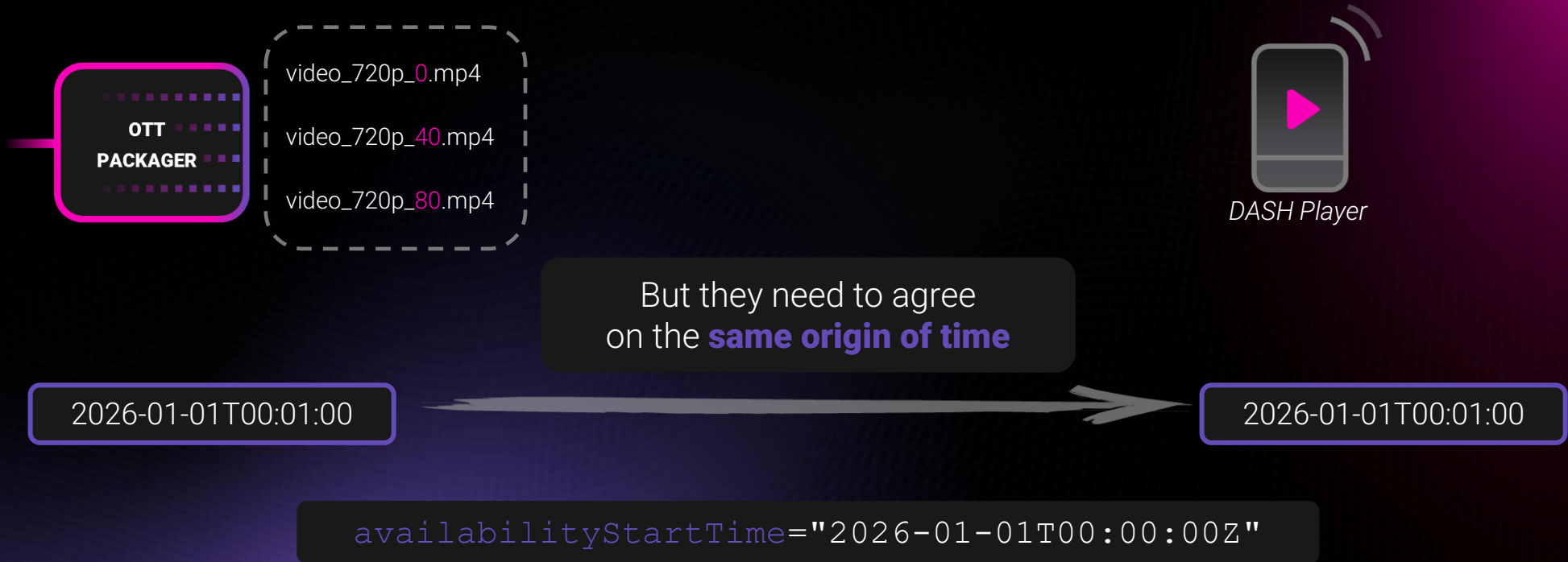
COGICOM

SYNCHRONIZATION PLAYERS & ORIGIN PACKAGERS



COGICOM

REQUEST THE RIGHT SEGMENT AT THE RIGHT TIME



COMPUTING SEGMENT START TIMES

Current Time

NTP

—

Origin of Time

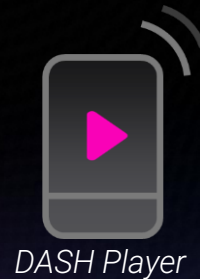
=

*Number of seconds elapsed since
the origin of time*

x Timescale

*Number of Timescale units
elapsed since the origin of time*

OTT
PACKAGER



DASH Player

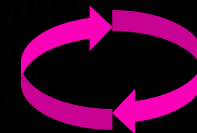
COGICOM

COMPUTING (FIXED) SEGMENT'S START TIME

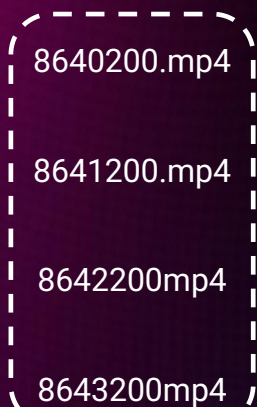
```
<SegmentTemplate timescale="100" media="DASH1TV-...$Time$.mp4">
```

```
<SegmentTimeline>
```

```
<S t="8640200" d="1000" r="3" />
```



At every segment duration (or more)



HTTP GET ../../8640200.mp4

HTTP GET ../../8641200.mp4

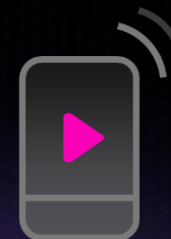
HTTP GET ../../8642200mp4

HTTP GET ../../8643200mp4

+ 1000

+ 1000

+ 1000



DASH Player

COGICOM

COMPUTING SEGMENT'S START TIME

Current Time

Origin of Time

2026-01-01T00:01:00

NTP

—

2026-01-01T00:00:00

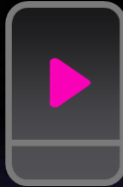
=

*0 h + 1 minute
60 sec*

x 100

*= 6000 Time units
elapsed since the origin of time*

OTT
PACKAGER



DASH Player

COGICOM

AVAILABILITY START TIME

COGICOM

AVAILABILITY START TIME: THE ORIGIN OF TIME

+00:00 (UTC time)

Zulu time (UTC time)



```
availabilityStartTime="2026-01-01T00:00:00Z"
```

- ▶ Arbitrary **UTC** date/time, **set in the past**
- ▶ Origin of time reference for the entire presentation/mpd
- ▶ Used for computing segment start times (\$time\$)

\$TIME SEGMENT ADDRESSING

COMPUTING SEGMENT START TIME

COGICOM

\$TIME SEGMENT ADDRESSING

```
availabilityStartTime="2026-01-01T00:00:00Z"  
<Period id="1" start="PT0S">  
  <SegmentTemplate timescale="10" media="video__$RepresentationID__$Time$.mp4"  
    <SegmentTimeline>  
      <S t="0" d="40" r="3" />
```

SEGMENT TIMELINE

$t="0"$ $d="40"$ $r="3"$

1 + 3

Start time of the 1st segment is at time = 0 relative to the Availability Start Time

This timeline consists of **segments duration** of 40 time-units

This timeline consists of:

- One initial segments ($t=0$)
- Repeated 3 more time

A total of **4 segments** repeated at 40 time-unit interval

`timescale="10"`

This timeline is made of 4 segments of 4 seconds each

COMPUTING SEGMENT'S \$TIME

Representation id

720p

```
availabilityStartTime="2026-01-01T00:00:00Z"  
<Period id="1" start="PT0S">  
  <SegmentTemplate timescale="10" media="video_RepresentationID_ $Time$.mp4"  
    <SegmentTimeline>  
      <S t="0" d="40" r="3" />
```



COGICOM

\$TIME SEGMENT ADDRESSING

COMPUTING SEGMENT'S ABSOLUTE START TIME

Representation id

720p

```
availabilityStartTime="2026-01-01T00:00:00Z"  
<Period id="1" start="PT0S">  
  <SegmentTemplate timescale="10" media="video__$RepresentationID__$Time$.mp4"  
    <SegmentTimeline>  
      <S t="0" d="40" r="3" />
```

\$time = 0

Segment starts at :
2026-01-01T00:00:00Z

\$time = 40

Segment starts at :
2026-01-01T00:00:04Z

\$time = 80

Segment starts at :
2026-01-01T00:00:08Z

\$time = 120

Segment starts at :
2026-01-01T00:00:12Z

COGICOM

\$TIME SEGMENT ADDRESSING

REAL WORLD EXAMPLE

Representation id

720p

```
availabilityStartTime="2026-05-25T16:07:32+00:00"  
<Period id="1" start="PT0S">  
  <SegmentTemplate timescale="50000" media="video_$RepresentationID$_$Time$.mp4"  
    <SegmentTimeline>  
      <S t="13515000"d="200000" r="4" />
```

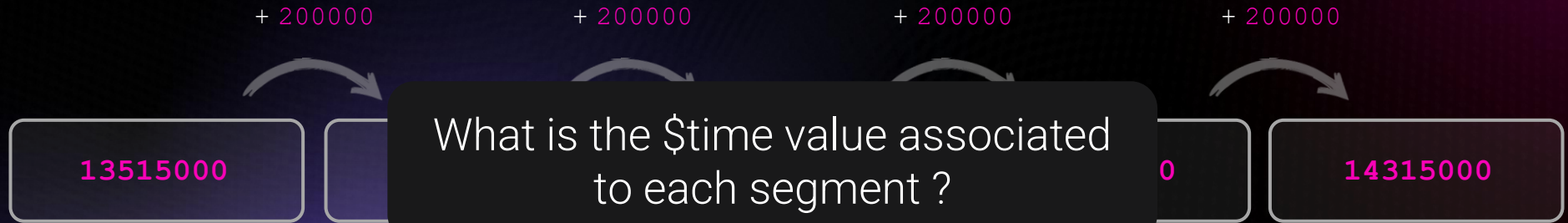
How many segment are contained in
this timeline?

REAL WORLD EXAMPLE

Representation id

720p

```
availabilityStartTime="2026-05-25T16:07:32+00:00"  
<Period id="1" start="PT0S">  
  <SegmentTemplate timescale="50000" media="video_$RepresentationID$_$Time$.mp4"  
    <SegmentTimeline>  
      <S t="13515000"d="200000" r="4" />
```



REAL WORLD EXAMPLE

Representation id

720p

```
availabilityStartTime="2026-05-25T16:07:32+00:00"  
<Period id="1" start="PT0S">  
  <SegmentTemplate timescale="50000" media="video_$RepresentationID$_$Time$.mp4"  
    <SegmentTimeline>  
      <S t="13515000"d="200000" r="4" />
```

13515000

13715000

13915000

14115000

14315000

What absolute time did the 1st segment of the timeline start?

REAL WORLD EXAMPLE

Representation id

720p

```
availabilityStartTime="2026-05-25T16:07:32+00:00"  
<Period id="1" start="PT0S">  
  <SegmentTemplate timescale="50000" media="video_$RepresentationID$_$Time$.mp4"  
    <SegmentTimeline>  
      <S t="13515000"d="200000" r="4" />
```

13515000

13715000

13915000

14115000

14315000

2026-05-25T16:07:32

+ 13515000 / 50000 = 270.3 sec
= 2026-05-25T16:12:02.3

The 1st segment of the timeline started on
2026-05-25 16:12:02.300

COGICOM

\$TIME SEGMENT ADDRESSING

REAL WORLD EXAMPLE

Representation id

720p

```
availabilityStartTime="2026-05-25T16:07:32+00:00"
<Period id="1" start="PT0S">
  <SegmentTemplate timescale="50000" media="video_$RepresentationID$_$Time$.mp4"
    <SegmentTimeline>
      <S t="13515000"d="200000" r="4" />
```

13515000

13715000

13915000

14115000

14315000

2026-05-25T16:07:32

+ 13715000 / 50000 = 274.3 sec
= 2026-05-25T16:12:06.3

The 2nd segment of the timeline started on
2026-05-25 16:12:06.300

COGICOM

\$TIME SEGMENT ADDRESSING

TIME SHIFT BUFFER DEPTH

THE LIVE WINDOW SIZE



Aka DVR window

COGICOM

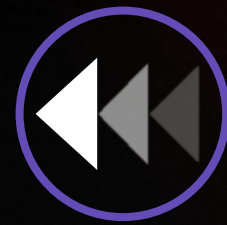
AMOUNT OF CONTENT KEPT ON THE ORIGIN SERVER

- NOT THE AMOUNT CONTENT VIDEO THE PLAYER SHOULD BUFFER
- IF THE PLAYER FETCHES A SEGMENT PAST THIS WINDOW, SERVER RETURNS "HTTP 404 FILE NOT FOUND"

Pause



Rewind



COGICOM

TIME SHIFT BUFFER DEPTH

```
...  
timeShiftBufferDepth="PT16.000S"  
<Period id="1" start="PT0S">  
  <SegmentTemplate timescale="10" media=...$Time.mp4  
    <SegmentTimeline>  
      <S t="0" d="40" r="3" />
```

This MPD contains 4 segments of 4 sec each
Total amount of content is 16 seconds

TIME SHIFT BUFFER DEPTH

```
...  
timeShiftBufferDepth="PT16.000S"  
<Period id="1" start="PT0S">  
  <SegmentTemplate timescale="10" media=...$Time.mp4  
    <SegmentTimeline>  
      <S t="0" d="40" r="3" />
```



It gives the **shortest** amount of content
among **all adaption sets**

Live Demo

- ▶ Identify the `TimeShiftBufferDepth` attribute
- ▶ Relate it to the number & duration of media segments inside the MPD
- ▶ Which one is shorter and reflected in the parameter?



Quiz 2

COGICOM

VARIABLE SEGMENT DURATIONS

WHY & WHEN

COGICOM

WHY DO VARIABLE SEGMENT DURATIONS

Dynamic Ad insertion

- ▶ Signaling ad break in live channel (SCTE-35)
- ▶ Cut the segment to ensure seamless splicing downstream

Variable GoP

- ▶ Create new segment on scene change
- ▶ Optimize packaging & VQ

Audio segment

- ▶ When audio segment can't be equal to video segment duration

COGICOM

Coming up!

VARIABLE SEGMENT DURATION

USING \$TIME

COGICOM

VARIABLE SEGMENT DURATION

```
<SegmentTemplate timescale="10"  
  <SegmentTimeline>  
    <S t="0" d="40" r="2" />  
    <S d="30" />  
    <S d="10" />  
    <S d="40" r="2" />
```

VARIABLE SEGMENT DURATION

```
<SegmentTemplate timescale="10"  
  <SegmentTimeline>  
    <S t="0" d="40" r="2"/>  
    <S d="30"/>  
    <S d="10"/>  
    <S d="40" r="2"/>
```

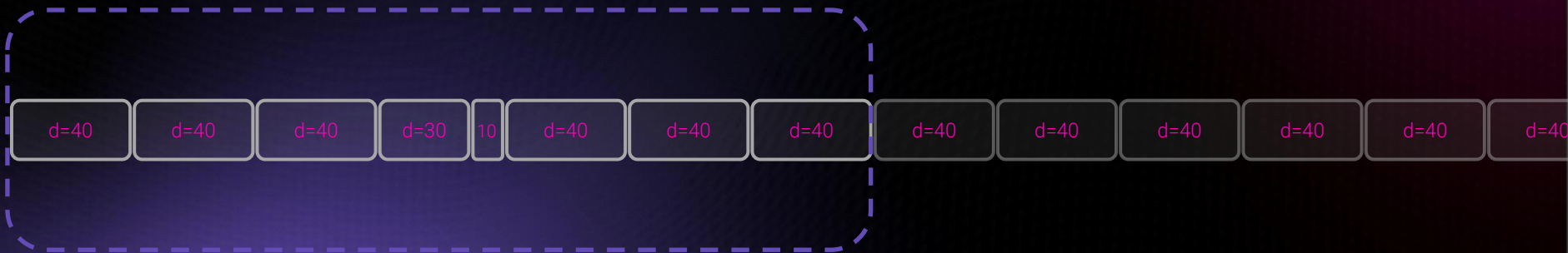


COGICOM

VARIABLE SEGMENT DURATION

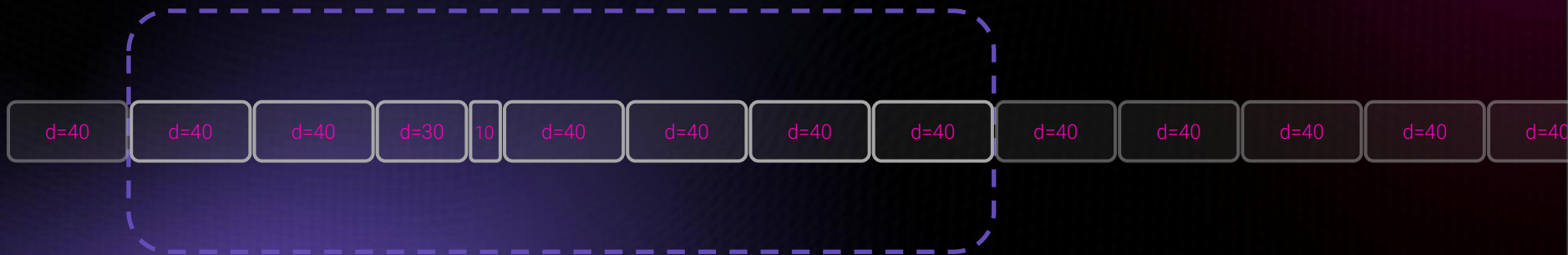
```
<SegmentTemplate timescale="10"  
  <SegmentTimeline>  
    <S t="0" d="40" r="2"/>  
    <S d="30"/>  
    <S d="10"/>  
    <S d="40" r="2"/>
```

timeShiftBufferDepth="PT28S"



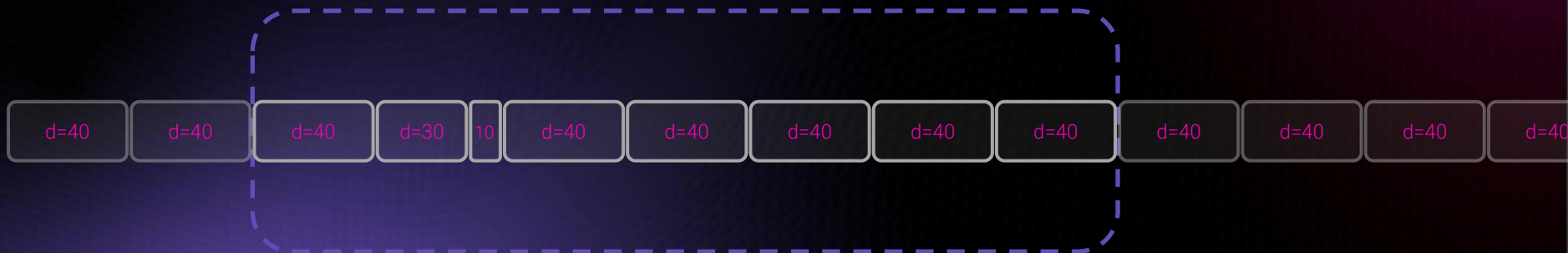
```
<SegmentTemplate timescale="10"  
  <SegmentTimeline>  
    <S t="40" d="40" r="1"/>  
    <S d="30"/>  
    <S d="10"/>  
    <S d="40" r="3"/>
```

timeShiftBufferDepth="PT28S"



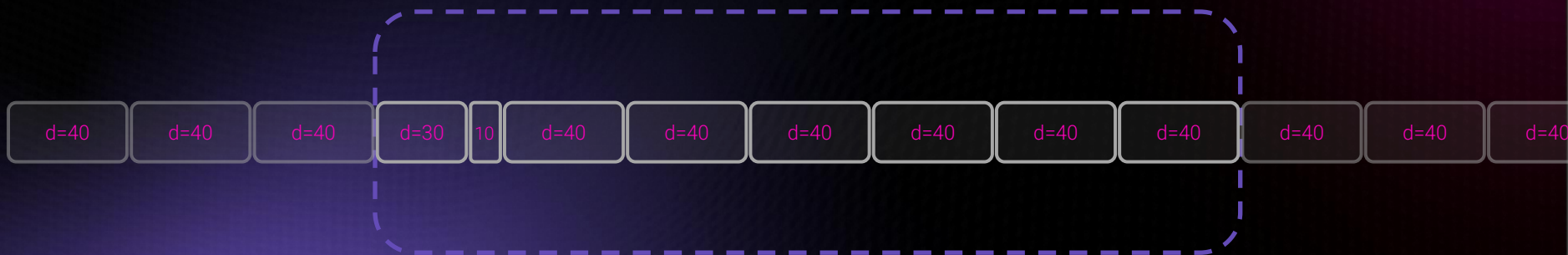
```
<SegmentTemplate timescale="10"  
  <SegmentTimeline>  
    <S t="80" d="40"/>  
    <S d="30"/>  
    <S d="10"/>  
    <S d="40" r="4"/>
```

timeShiftBufferDepth="PT28S"

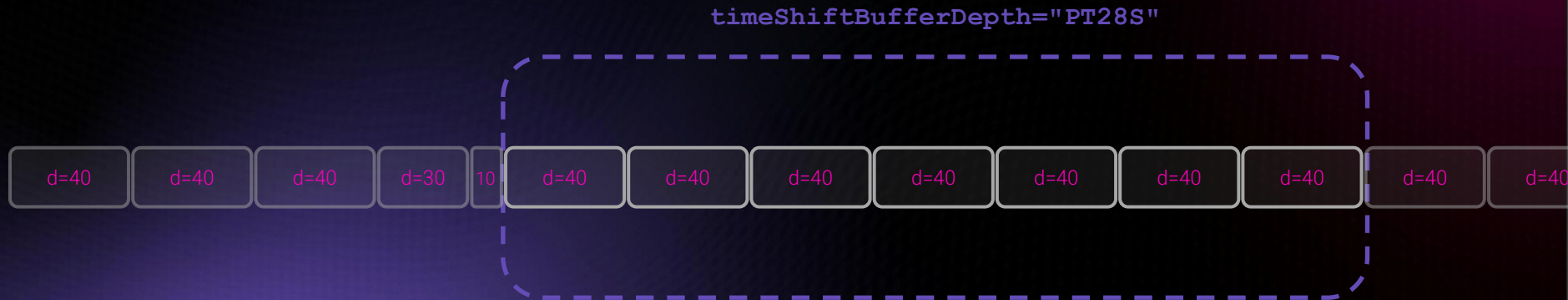


```
<SegmentTemplate timescale="10"  
  <SegmentTimeline>  
    <S t="120" d="30"/>  
    <S d="10"/>  
    <S d="40" r="5"/>
```

timeShiftBufferDepth="PT28S"



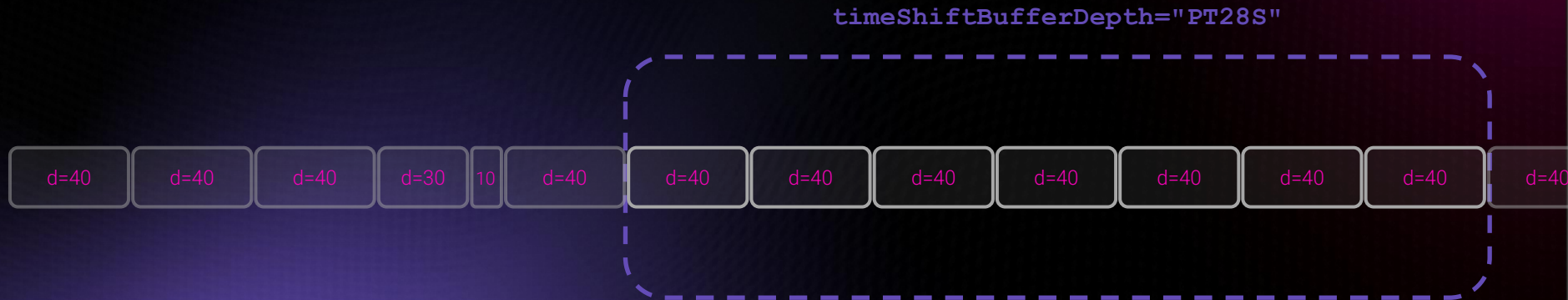
```
<SegmentTemplate timescale="10"  
  <SegmentTimeline>  
    <S t="160" d="40" r="6"/>
```



COGICOM

VARIABLE SEGMENT DURATION

```
<SegmentTemplate timescale="10"  
  <SegmentTimeline>  
    <S t="200" d="40" r="6"/>
```



COGICOM

VARIABLE SEGMENT DURATION

AUDIO & \$TIME ADDRESSING

WHEN VIDEO & AUDIO DURATION AREN'T EQUAL

COGICOM

```
<SegmentTemplate timescale="48000"  
  <SegmentTimeline>  
    <S t="494848" d="192512"/>  
    <S d="191488"/>  
    <S d="192512"/>  
    <S d="191488"/>  
    <S d="192512"/>  
    <S d="191488"/>  
    <S d="192512"/>
```



AUDIO & \$TIME ADDRESSING

We asked the packager to create
4-sec video & audio segments

COGICOM

AUDIO & \$TIME ADDRESSING

- ▶ making 4-sec segments for video & audio
- We asked the packager to create
4-sec video & audio segments

Can we make 4-sec **video** segments ?

Can we make 4-sec **audio** segments ?

COGICOM

MAKING 4-SEC VIDEO SEGMENTS

- ▶ At 25fps, a video frame lasts 0.04 sec or 40ms.
- ▶ How many frames can fit into one 4-sec segment?
 - ▶ $4 \text{ sec} / 0.04 = 100 \text{ frames}$

There will be **exactly** 100 video frames in each video segment

Video segment durations are **constant**

MAKING 4-SEC AUDIO SEGMENTS

It's more complicated

- ▶ With MPEG AAC-LC audio codec.
- ▶ Audio is transmitted as audio frames
- ▶ Each audio frame consists of 1024 audio samples
- ▶ One audio frame is $1024 / 48000 = 21.33\text{ms}$

In MPEG AAC-LC 48Khz,
one audio frame lasts **21.33ms**

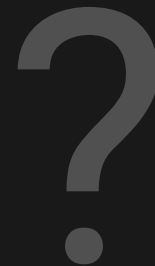
MAKING 4-SEC AUDIO SEGMENTS

In MPEG AAC-LC 48Khz,
one audio frame lasts **21.33ms**

Can we make 4-sec long **audio** segments?

- ▶ How many audio frames in a 4-sec segment?
 - ▶ $4000 / 21.33 = 187.5$ frames
- ▶ Segment can only be made of full frames, not fractional
- ▶ Segment durations must **fluctuate** between **187 frames & 188 frames**
 - ▶ Gives an average of 187.5 frames

```
<SegmentTemplate timescale="48000"  
  <SegmentTimeline>  
    <S t=" 192512 " d="192512"/>  
    <S d=" 191488 "/>  
    <S d=" 192512 "/>  
    <S d=" 191488 "/>  
    <S d=" 192512 "/>  
    <S d=" 191488 "/>  
    <S d=" 192512 "/>
```



timescale="48000"

Segment duration
(*timescale units*)

Segment duration
(sec)

Nb of frames
per segment

Average of 187.5
frames/segment

192512

4.0107

188 frames

191488

3.9893

187 frames

192512

4.0107

188 frames

191488

3.9893

187 frames

192512

4.0107

188 frames

191488

3.9893

187 frames

192512

4.0107

188 frames

≈

28 sec of audio content

≈

COGICOM

AUDIO & \$TIME ADDRESSING

THE SOLUTION : 1.92 SEC

COGICOM

\$NUMBER SEGMENT ADDRESSING

AN OVERVIEW

COGICOM

\$NUMBER PRINCIPLE

Number of segments generated
since the origin of time

```
availabilityStartTime="2026-01-01T00:00:00Z"  
<Period id="1" start="PT0S">  
  <SegmentTemplate timescale="50000" duration="200000" media="video_... _$Number.mp4"  
    <Representation ...  
    <Representation ...  
    <Representation ...
```

No more segment timeline

\$number template assumes that
all segments have equal duration

COMPUTING SEGMENT INDEXES

Current Time

NTP

—

Origin of Time

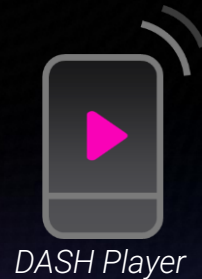
=

*Number of seconds elapsed since
the origin of time*

/ Segment duration

*of segments generated
since the origin of time*

OTT
PACKAGER



DASH Player

COGICOM

\$TIME vs \$NUMBER: USE CASES

	\$time\$	\$number\$
Segment Duration	Variable or fixed	Fixed only
Use cases		
Manifest Fetching		

\$TIME vs \$NUMBER: USE CASES

	\$time\$	\$number\$
Segment Duration	Variable or fixed	Fixed only
Use cases	VOD, Live , DAI & Low Latency	VOD & Live
Manifest Fetching		

\$TIME vs \$NUMBER: USE CASES

	\$time\$	\$number\$
Segment Duration	Variable or fixed	Fixed only
Use cases	VOD, Live , DAI & Low Latency	VOD & Live
Manifest Fetching	<i>Must be refreshed at every segment duration</i>	<i>Can be retrieved only once at session start</i>

COGICOM

Coming up!

\$TIME vs \$NUMBER: USE CASES

	\$time\$
Manifest Fetching	<i>Must be refreshed at every segment duration</i>

*Hands on
Lab*

COGICOM

Quiz 3

COGICOM