hulu

DASH Live Streaming at Scale

Portland Video Meetup, December 2019
ZACHARY CAVA

Video Platform Architect

DASH Live Streaming at Scale
Hulu is TV
Topics Today

• Basics of a Live Stream
  • Live Streaming with DASH
  • Optimizing for Scale
  • MPD Patch Updates

• Dynamic Ad Replacement
  • Requirements of Ad Targeting
  • Server-Guided Ad Insertion
  • Returning from a Replacement

• Q&A
Basics of a Live Stream
DASH Live Streaming at Scale

Basics of a Live Stream

Time

Distribution Feed

Video

Audio
DASH Live Streaming at Scale

Basics of a Live Stream

Time

Distribution Feed

Encoded Variants

Video

Audio

Video

Audio
DASH Live Streaming at Scale

Basics of a Live Stream

Time

Distribution Feed
- Video
- Audio
- SCTE Markers

Encoded Variants
- Video
- Audio
Basics of a Live Stream

DASH Live Streaming at Scale

Time

Distribution Feed

Video

Audio

SCTE Markers

Encoded Variants

Video

Audio
DASH Live Streaming at Scale

Basics of a Live Stream

Time

Distribution Feed

Video
Audio
SCTE Markers

Encoded Variants

Video
Audio

Periods

| In-Stream Content | In-Stream Ad Opportunity | In-Stream Content |
Live Streaming with DASH
The Timing Model

Wall Clock

Periods

- In-Stream Content
- In-Stream Ad Opportunity
- In-Stream Content

Video

Audio

Timing Model Components

- Effective Time Shift Buffer
- Delay
- Time Shift Buffer

$t=now$
Live Streaming with DASH

The Timing Model

Wall Clock

Periods

In-Stream Content | In-Stream Ad Opportunity | In-Stream Content

Video

Audio

Timing Model Components

Effective Time Shift Buffer | Delay

Time Shift Buffer

\[ t=\text{now} \]
Live Streaming with DASH

Segment Addressing

channel/init/video.mp4
channel/video/1.m4s
channel/video/2.m4s
channel/video/3.m4s
channel/video/4.m4s
channel/video/5.m4s
channel/video/6.m4s
channel/video/7.m4s
channel/video/8.m4s
channel/video/9.m4s
Live Streaming with DASH

Segment Addressing: HLS

```
#EXTM3U
#EXT-X-TARGETDURATION:4
#EXT-X-VERSION:4
#EXT-X-MEDIA-SEQUENCE:1
#EXT-X-MAP:URI="channel/init/video.mp4"
#EXTINF:4.004
channel/video/1.m4s
#EXTINF:4.004
channel/video/2.m4s
#EXTINF:4.004
channel/video/3.m4s
#EXTINF:4.004
channel/video/4.m4s
#EXTINF:4.004
channel/video/5.m4s
#EXTINF:4.004
channel/video/6.m4s
#EXTINF:4.004
channel/video/7.m4s
#EXTINF:4.004
channel/video/8.m4s
#EXTINF:4.004
channel/video/9.m4s
...```
Live Streaming with DASH

Segment Addressing: SegmentList

channel/init/video.mp4
channel/video/1.m4s
channel/video/2.m4s
channel/video/3.m4s
channel/video/4.m4s
channel/video/5.m4s
channel/video/6.m4s
channel/video/7.m4s
channel/video/8.m4s
channel/video/9.m4s

<Representation...
  <SegmentList timescale="90000" duration="360360">
    <Initialization sourceURL="channel/init/video.mp4"/>
    <SegmentURL media="channel/video/1.m4s"/>
    <SegmentURL media="channel/video/2.m4s"/>
    <SegmentURL media="channel/video/3.m4s"/>
    <SegmentURL media="channel/video/4.m4s"/>
    <SegmentURL media="channel/video/5.m4s"/>
    <SegmentURL media="channel/video/6.m4s"/>
    <SegmentURL media="channel/video/7.m4s"/>
    <SegmentURL media="channel/video/8.m4s"/>
    <SegmentURL media="channel/video/9.m4s"/>
  </SegmentList>
</Representation>
Live Streaming with DASH

Segment Addressing: SegmentTemplate with $Number$

channel/init/video.mp4
channel/video/1.m4s
channel/video/2.m4s
channel/video/3.m4s
channel/video/4.m4s
channel/video/5.m4s
channel/video/6.m4s
channel/video/7.m4s
channel/video/8.m4s
channel/video/9.m4s

```xml
<SegmentTemplate timescale="90000"
    duration="360360"
    startNumber="1"
    media="channel/video/$Number$.m4s"
    initialization="channel/init/video.mp4">
</SegmentTemplate>
```
Live Streaming with DASH

Segment Addressing: SegmentTemplate with $\$Time\$

channel/init/video.mp4
channel/video/1.m4s
channel/video/2.m4s
channel/video/3.m4s
channel/video/4.m4s
channel/video/5.m4s
channel/video/6.m4s
channel/video/7.m4s
channel/video/8.m4s
channel/video/9.m4s

```
<SegmentTemplate timescale="90000"
presentationTimeOffset="3054136803"
media="channel/video/$\$Time\$.m4s"
initialization="channel/init/video.mp4">
  <SegmentTimeline>
  <S t="3054139806" d="360360"/>
  <S t="3054500166" d="360360"/>
  <S t="3054860526" d="360360"/>
  <S t="3055220886" d="360360"/>
  <S t="3055581246" d="360360"/>
  <S t="3055941606" d="360360"/>
  <S t="3056301966" d="360360"/>
  <S t="3056662326" d="360360"/>
  <S t="3057022686" d="360360"/>
  <S t="3057383046" d="360360"/>
  <S t="3057743406" d="360360"/>
  </SegmentTimeline>
</SegmentTemplate>
```
<Period id="1" start="...">
  <BaseURL serviceLocation="cdn-1">https://www.cdn1.com/</BaseURL>
  <BaseURL serviceLocation="cdn-2">https://www.cdn2.com/</BaseURL>
  <BaseURL serviceLocation="cdn-3">https://www.cdn3.com/</BaseURL>
  ...
</Period>
Basic Manifest Example

• Let’s look at a 5 minute manifest
• Live Streaming Details
  • Multi-Period
  • 8 Video Representations
  • 1 Audio Representation
  • CEA-608/708 Captions
  • 3 Segment CDNs
Live Streaming with DASH

5 Minute Manifest (1 of ?)

```xml
<?xml version="1.0" encoding="UTF-8"?>
<MPD type="dynamic" minimumUpdatePeriod="PT3.0S" suggestedPresentationDelay="PT8S"
 availabilityStartTime="2017-05-01T07:00:00+00:00" ...>
<Period id="1" start="PT70406325.063S">
  <BaseURL serviceLocation="cdn-1">https://cdn-1.net/channel/</BaseURL>
  <BaseURL serviceLocation="cdn-2">https://cdn-2.net/channel/</BaseURL>
  <BaseURL serviceLocation="cdn-3">https://cdn-3.net/channel/</BaseURL>
  <AdaptationSet id="2" mimeType="video/mp4" segmentAlignment="true"
    bitstreamSwitching="true" maxWidth="1280" maxHeight="720" maxFrameRate="60000/1001">
    <Accessibility schemeIdUri="urn:scte:dash:cc:cea-608:2015" value="CC1=eng"/>
    <ContentProtection schemeIdUri="urn:mpeg:dash:mp4protection:2011" value="cenc" cenc:default_KID="..."/>
    <ContentProtection schemeIdUri="urn:uuid:9a04f079-9840-4286-ab92-e65be0885f95"/>
    <ContentProtection schemeIdUri="urn:uuid:edef8ba9-79d6-4ace-a3c8-27dcd51d21ed"/>
    <Representation id="VIDEO_7" codecs="avc1.4d400d" width="352" height="198"
      startWithSAP="1" bandwidth="474092" frameRate="30000/1001">
      <SegmentTemplate timescale="90000" presentationTimeOffset="3054136803"
        media="channel/VIDEO_7/$Time$.m4s"
        initialization="channel/init/VIDEO_7.mp4">
        <SegmentTimeline>
          <S t="3054139806" d="360360"/>
        </SegmentTimeline>
      </SegmentTemplate>
    </Representation>
  </AdaptationSet>
</Period>
</MPD>
```
5 Minute Manifest (2 of ?)

Live Streaming with DASH

```xml
<S t="3054500166" d="360360"/>
<S t="3054860526" d="360360"/>
<S t="3055220886" d="360360"/>
<S t="3055581246" d="360360"/>
<S t="3055941606" d="360360"/>
<S t="3056301966" d="360360"/>
<S t="3056662326" d="360360"/>
<S t="3057022686" d="360360"/>
<S t="3057383046" d="360360"/>
<S t="3057743406" d="360360"/>
<S t="3058103766" d="360360"/>
<S t="3058464126" d="360360"/>
<S t="3058824486" d="360360"/>
<S t="3059184846" d="360360"/>
<S t="3059545206" d="360360"/>
<S t="3059905566" d="360360"/>
<S t="3060265926" d="360360"/>
<S t="3060626286" d="360360"/>
<S t="3060986646" d="360360"/>
<S t="3061347006" d="360360"/>
<S t="3061707366" d="360360"/>
```
5 Minute Manifest (3 of ?)

Live Streaming with DASH

```xml
<S t="3062067726" d="360360"/>
<S t="3062428086" d="360360"/>
<S t="3062788446" d="360360"/>
<S t="3063148806" d="360360"/>
<S t="3063509166" d="360360"/>
<S t="3063869526" d="360360"/>
<S t="3064229886" d="360360"/>
<S t="3064590246" d="360360"/>
<S t="3064950606" d="360360"/>
<S t="3065310966" d="360360"/>
<S t="3065671326" d="360360"/>
<S t="3066031686" d="360360"/>
<S t="3066392046" d="360360"/>
<S t="3066752406" d="360360"/>
<S t="3067112766" d="360360"/>
<S t="3067473126" d="360360"/>
<S t="3067833486" d="360360"/>
<S t="3068193846" d="360360"/>
<S t="3068554206" d="360360"/>
<S t="3068914566" d="360360"/>
<S t="3069274926" d="360360"/>
```
Live Streaming with DASH

5 Minute Manifest (4 of ?)

<S t="3069635286" d="360360"/>
<S t="3069995646" d="360360"/>
<S t="3070356006" d="360360"/>
<S t="3070716366" d="360360"/>
<S t="3071076726" d="360360"/>
<S t="3071437086" d="360360"/>
<S t="3071797446" d="360360"/>
<S t="3072157806" d="360360"/>
<S t="3072518166" d="360360"/>
<S t="3072878526" d="360360"/>
<S t="3073238886" d="360360"/>
<S t="3073599246" d="360360"/>
<S t="3073959606" d="360360"/>
<S t="3074319966" d="360360"/>
<S t="3074680326" d="360360"/>
<S t="3075040686" d="360360"/>
<S t="3075401046" d="360360"/>
<S t="3075761406" d="360360"/>
<S t="3076121766" d="360360"/>
<S t="3076482126" d="360360"/>
<S t="3076842486" d="360360"/>
<SegmentTimeline>
  <S t="3077202846" d="360360"/>
  <S t="3077563206" d="360360"/>
  <S t="3077923566" d="360360"/>
  <S t="3078283926" d="360360"/>
  <S t="3078644286" d="360360"/>
  <S t="3079004646" d="360360"/>
  <S t="3079365006" d="360360"/>
  <S t="3079725366" d="360360"/>
</SegmentTimeline>
</SegmentTemplate>
</Representation>

<Representation id="VIDEO_6" codecs="avc1.4d400d" width="384" height="216"
startWithSAP="1" bandwidth="741797" frameRate="30000/1001">
  <SegmentTemplate timescale="90000" presentationTimeOffset="3054136803"
media="channel/VIDEO_6/$Time$.m4s"
initialization="channel/init/VIDEO_6.mp4">
  <SegmentTimeline>
    <S t="3054139806" d="360360"/>
    <S t="3054500166" d="360360"/>
    <S t="3054860526" d="360360"/>
    <S t="3055220886" d="360360"/>
  </SegmentTimeline>
</SegmentTemplate>
</Representation>
Moments Later...
Live Streaming with DASH

5 Minute Manifest (36 of 36)

```xml
<SegmentTimeline>
  </SegmentTemplate>
</Representation>
</AdaptationSet>
</Period>
</MPD>
```
Live Streaming with DASH

Only 5 Minutes of Content

- By The Numbers
  - 694 XML Nodes
  - 46KB Uncompressed Size
  - ~30ms Parse Time
- Simply Does Not Scale
  - 1 hour => 8665 Nodes, 573KB
  - 3 hours => 26512 Nodes, 1.9MB
- What Makes The Manifest So Big?
  - Explicit segment addressing
  - Usage of multiple periods
Optimizing for Scale
Focusing Optimizations

- Concurrency is The Challenge
  - Clients constantly polling
  - Need highly cacheable responses
  - Avoid sacrificing quality for scale
- Optimize for Network and Clients
  - Ensure high cache hit ratio
  - Minimize cost of updates for clients
  - Service resources aren’t endless
- Important to Optimize Holistically
Compress the Timelines: Omit $t$ for Consecutive Segments

```
<SegmentTemplate timescale="90000"
    presentationTimeOffset="3054136803"
    media="channel/video/$Time$.m4s"
    initialization="channel/init/video.mp4">
    <SegmentTimeline>
        <S t="3054139806" d="360360"/>
        <S t="3054500166" d="360360"/>
        <S t="3054860526" d="360360"/>
        <S t="3055220886" d="360360"/>
        <S t="3055581246" d="360360"/>
        <S t="3055941606" d="360360"/>
        <S t="3056301966" d="360360"/>
        <S t="3056662326" d="360360"/>
        <S t="3057022686" d="360360"/>
        <S t="3057383046" d="360360"/>
        <S t="3057743406" d="360360"/>
    </SegmentTimeline>
</SegmentTemplate>
```
Compress the Timelines: Utilize S@r Attribute

```
<SegmentTemplate timescale="90000"
    presentationTimeOffset="3054136803"
    media="channel/video/$Time$.m4s"
    initialization="channel/init/video.mp4">
    <SegmentTimeline>
        <S t="3054139806" d="360360"/>
        <S d="360360"/>
        <S d="360360"/>
        <S d="360360"/>
        <S d="360360"/>
        <S d="360360"/>
        <S d="360360"/>
        <S d="360360"/>
        <S d="360360"/>
    </SegmentTimeline>
</SegmentTemplate>
```

```
<SegmentTemplate timescale="90000"
    presentationTimeOffset="3054136803"
    media="channel/video/$Time$.m4s"
    initialization="channel/init/video.mp4">
    <SegmentTimeline>
        <S t="3054139806" d="360360" r="10"/>
    </SegmentTimeline>
</SegmentTemplate>
```
Compress the Timelines: Amazon Pattern Proposal

<SegmentTemplate timescale="90000"
presentationTimeOffset="3054857523"
media="channel/audio/$Time$.m4s"
initialization="channel/init/audio.mp4">
  <SegmentTimeline>
    <Pattern t="3054861448"
pattern_repeat="12">
      <P d="360960"/>
      <P d="359040"/>
      <P d="360960" r="1"/>
      <P d="359040"/>
      <P d="360960" r="2"/>
      <P d="359040"/>
      <P d="360960" r="1"/>
      <P d="359040"/>
    </Pattern>
    ....many lines...
    <S d="360960" r="1"/>
    <S d="359040"/>
  </SegmentTimeline>
</SegmentTemplate>

<SegmentTemplate timescale="90000"
presentationTimeOffset="3054857523"
media="channel/audio/$Time$.m4s"
initialization="channel/init/audio.mp4">
  <SegmentTimeline>
    <S t="3054861448" d="360960"/>
    <S d="359040"/>
    <S d="360960" r="1"/>
    <S d="359040"/>
    <S d="360960" r="2"/>
    <S d="359040"/>
    <S d="360960" r="1"/>
    <S d="359040"/>
    ....many lines...
    <S d="360960" r="1"/>
  </SegmentTimeline>
</SegmentTemplate>
Optimizing for Scale

Compress the Timelines: One Timeline per AdaptationSet

```xml
<AdaptationSet id="1" mimeType="video/mp4" segmentAlignment="true"
   bitstreamSwitching="true">
   <SegmentTemplate timescale="90000" presentationTimeOffset="3054136803"
      media="channel/$RepresentationID$/Time$.m4s"
      initialization="channel/$RepresentationID$/video.mp4">
      <SegmentTimeline>
         <S t="3054139806" d="360360" r="10"/>
      </SegmentTimeline>
   </SegmentTemplate>
   <Representation id="VIDEO_7" />
   <Representation id="VIDEO_6" />
   <Representation id="VIDEO_5" />
   <Representation id="VIDEO_4" />
   <Representation id="VIDEO_3" />
   <Representation id="VIDEO_2" />
   <Representation id="VIDEO_1" />
   <Representation id="VIDEO_0" />
</AdaptationSet>
```
By The Numbers

- 78 XML Nodes
- 8.1KB Uncompressed Size
- ~20ms Parse Time
- Smaller Magnitude, Scales The Same
- 1 hour => 986 Nodes, 100KB
- 3 hour => 3351 Nodes, 373KB
- Cost Accumulates Over Time
- Updates increase in cost over time
- Parsing during playback is expensive
- What is really changing in updates?
Optimizing for Scale

Change Across Updates
Optimizing Update Requests

- Solution Constraints
  - Send only new information to clients
  - Maintain response caching
  - Minimize request client processing
  - Keep server in control of playout
- Created the Patch Manifest concept
  - Special server update response
  - Leverages MPD structure semantics
  - Uses client persisted breadcrumbing
  - Relies on client timeline persistence
MPD Patch Updates
MPD Patch Updates

Patch Manifest Playout

Received Manifests → Join Manifest → Patch 1

- Big Buck Bunny
- Rabbit
- Landscape

Patch 1 Question Mark
MPD Patch Updates

Patch Manifest Playout

Received Manifests

Join Manifest

Patch 1

In-Memory Manifest

MPD Patch Updates

Patch Manifest Playout

Received Manifests

Join Manifest

Patch 1

In-Memory Manifest
MPD Patch Updates

Patch Manifest Playout

Received Manifests

Join Manifest

Patch 1

Patch 2

In-Memory Manifest

Patch Manifest Playout
MPD Patch Updates

Patch Manifest Playout

Received Manifests

Join Manifest

Patch 1

Patch 2

In-Memory Manifest

Patch Manifest Playout
Anatomy of a Join Manifest

<?xml version="1.0" encoding="UTF-8"?>
<MPD type="dynamic" id="channel" minimumUpdatePeriod="PT3.0S" publishTime="2019-10-18T22:06:14" >
  <PatchLocation ttl="60">/patch/channel.mpd?publishTime=2019-10-18T22:06:14&s=5095823234</PatchLocation>
  <Period id="1" start="PT70715052.772S">
    <BaseURL serviceLocation="cdn-1">https://cdn-1.net/channel/</BaseURL>
    <BaseURL serviceLocation="cdn-2">https://cdn-2.net/channel/</BaseURL>
    <BaseURL serviceLocation="cdn-3">https://cdn-3.net/channel/</BaseURL>
    <AdaptationSet id="2" mimeType="video/mp4" segmentAlignment="true" bitstreamSwitching="true" >
      <Accessibility schemeIdUri="urn:scte:dash:cc:cea-608:2015" value="CC1=eng"/>
      <ContentProtection schemeIdUri="urn:mpeg:dash:mp4protection:2011" value="cenc" cenc:default_KID="..."/>
      <ContentProtection schemeIdUri="urn:uuid:9a04f079-9840-4286-ab92-e65be0885f95"/>
      <ContentProtection schemeIdUri="urn:uuid:edef8ba9-79d6-4ace-a3c8-27dcd51d21ed"/>
      <SegmentTemplate timescale="90000" presentationTimeOffset="5069874311" media="channel/$RepresentationID$/S$Time$.m4s"
                       initialization="channel/init/$RepresentationID$.mp4">
        <SegmentTimeline>
          <S t="5069877314" d="360360" r="71"/>
        </SegmentTimeline>
      </SegmentTemplate>
    </AdaptationSet>
  </Period>
</MPD>
Anatomy of a Patch Manifest

```xml
<?xml version="1.0" encoding="UTF-8"?>
  <p:replace sel="/MPD/PatchLocation[0]">
    <PatchLocation ttl="60">/patch/channel.mpd?publishTime=2019-10-18T22:06:17&s=5095823234</PatchLocation>
  </p:replace>
  <p:add sel="/MPD/Period[@id='1']/AdaptationSet[@id='1']/SegmentTemplate/SegmentTimeline">
    <S t="5095823752" d="360960" />
  </p:add>
  <p:add sel="/MPD/Period[@id='1']/AdaptationSet[@id='2']/SegmentTemplate/SegmentTimeline">
    <S t="5095823234" d="360360" />
  </p:add>
</Patch>
```
MPD Patch Updates

Patch Manifests

- By The Numbers
  - 9 XML Nodes
  - ~1KB Uncompressed Size
  - ~2ms Parse Time
- Successfully Deployed Solution
  - Constant cost addition over time
  - Efficient for network and client
  - No reduction in cache efficiency
  - Server fully in control of timeline
Dynamic Ad Replacement
Performing Ad Replacement

**Dynamic Ad Replacement**

<table>
<thead>
<tr>
<th>Periods</th>
<th>In-Stream Content</th>
<th>In-Stream Ad Opportunity</th>
<th>In-Stream Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audio</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Dynamic Ad Replacement

Performing Ad Replacement

Time

<table>
<thead>
<tr>
<th>Periods</th>
<th>In-Stream Content</th>
<th>Slot 1</th>
<th>Slot 2</th>
<th>S3</th>
<th>In-Stream Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Decisioned Ad Pod
Performing Ad Replacement

Dynamic Ad Replacement

Time

<table>
<thead>
<tr>
<th>Periods</th>
<th>In-Stream Content</th>
<th>Slot 1</th>
<th>Slot 2</th>
<th>S3</th>
<th>In-Stream Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Audio</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Events</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Decisioned Ad Pod
DASH Live Streaming at Scale

Requirements of Ad Targeting
Requirements of Ad Targeting

Targeting Drives Stream Uniqueness
Requirements of Ad Targeting

Targeting Drives Stream Uniqueness

Letter A

Letter A

Letter B

Letter B
Requirements of Ad Targeting

Targeting Drives Stream Uniqueness

1. Letter A
2. Letter A
3. Letter B
4. Letter B
Requirements of Ad Targeting

Targeting Drives Stream Uniqueness

Letter A
Number 1

Letter A
Number 1

Letter B
Number 2

Letter B
Number 3
Requirements of Ad Targeting

Targeting Drives Stream Uniqueness

1. Letter A
   Number 1

2. Letter A
   Number 1

3. Letter B
   Number 2

4. Letter B
   Number 3
Targeting Drives Stream Uniqueness

- **1.** Letter A
  - Number 1
  - Color Red

- **2.** Letter A
  - Number 1
  - Color Yellow

- **3.** Letter B
  - Number 2
  - Color Green

- **4.** Letter B
  - Number 3
  - Color Yellow
### Requirements of Ad Targeting

**Targeting Drives Stream Uniqueness**

<table>
<thead>
<tr>
<th>Letter</th>
<th>Number</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>Red</td>
</tr>
<tr>
<td>A</td>
<td>1</td>
<td>Yellow</td>
</tr>
<tr>
<td>B</td>
<td>2</td>
<td>Green</td>
</tr>
<tr>
<td>B</td>
<td>3</td>
<td>Yellow</td>
</tr>
</tbody>
</table>
Requirements of Ad Targeting

More Granularity, Greater Uniqueness
Requirements of Ad Targeting

More Granularity, Greater Uniqueness
Requirements of Ad Targeting

More Granularity, Greater Uniqueness
Requirements of Ad Targeting

User Based Targeting

- Individualized Ad Targeting
  - Simplifies Ad Sales Flow
  - Keeps Ad Experience Consistent
  - Requires Stream Unique Decisions

- Common Live Streaming Requirements
  - Polling for Changes
  - Stream Update Consistency
Requirements of Ad Targeting

Players Requesting Unique Updates
Requirements of Ad Targeting

Per User Stream Updates
Per User Stream Updates

Requirements of Ad Targeting
Requirements of Ad Targeting

Per User Stream Updates
Requirements of Ad Targeting

Per User Stream Updates
Requirements of Ad Targeting

Per User Stream Updates
Requirements of Ad Targeting

Per User Stream Updates
Requirements of Ad Targeting

Per User Stream Updates
Requirements of Ad Targeting

Per User Stream Updates
Server-Guided Ad Insertion
Server-Guided Ad Insertion

Revisiting Targeted Streams

1.

2.

3.

4.
Revisiting Targeted Streams

Server-Guided Ad Insertion
## Keeping Responses Generic

Server-Guided Ad Insertion

<table>
<thead>
<tr>
<th>Element or Attribute Name</th>
<th>Use</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UrlQueryInfo</td>
<td></td>
<td>Provides URL query string information</td>
</tr>
<tr>
<td>@queryTemplate</td>
<td>O</td>
<td>Provides URL parameters template information</td>
</tr>
<tr>
<td>@useMPDUrlQuery</td>
<td>O</td>
<td>Indicates the URL parameters of the MPD URL are used</td>
</tr>
<tr>
<td>ExtendedUrlInfoType</td>
<td></td>
<td>Provides information for derivation of parameter string. This is an extension of <code>UrlQueryInfo</code></td>
</tr>
<tr>
<td>@includeInRequests</td>
<td>OD</td>
<td>Specifies which HTTP GET requests shall carry parameters. White-spaced concatenated list of these keys: <code>“segment”, “xlink”, “mpd”, “callback”, “chaining”, “fallback”</code></td>
</tr>
</tbody>
</table>
<EssentialProperty schemeIdUri="urn:mpeg:dash:urlparam:2016">
    <up:ExtUrlQueryInfo useMPDUrlQuery="true" includeInRequests="mpd">
        queryTemplate="target_params=${query:target_params}$" />
</EssentialProperty>
Server-Guided Ad Insertion

Shared Responses

1

2

3

4
Server-Guided Ad Insertion

Shared Responses

1. [Image]
2. [Image]
3. [Image]
4. [Image]
Separation of Scale Concerns

Server-Guided Ad Insertion

Content Patch Cluster

Advertisement Patch Cluster
Returning from a Replacement
Normal Patch Playout

Returning from a Replacement

Time

In-Memory MPD

Periods
In-Stream Content 1

Segments

Patch Update

Periods
1

Segments
Returning from a Replacement

Advertisement Patch Occurs

Time

In-Memory MPD

Periods

In-Stream Content 1

Segments

Patch Update

<table>
<thead>
<tr>
<th>Ad 1</th>
<th>Ad 2</th>
<th>A3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

82
Early Segment Announcement

Time

In-Memory MPD

Periods

Segments

In-Stream Content 1  Ad 1  Ad 2  A3

Patch Update
# Early Segment Announcement

<table>
<thead>
<tr>
<th>Time</th>
<th>In-Memory MPD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Periods</td>
<td>In-Stream Content 1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Segments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Availability</th>
<th>Active Window</th>
</tr>
</thead>
</table>

| Patch Update |
Early Segment Announcement

Returning from a Replacement

<table>
<thead>
<tr>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>

In-Memory MPD

<table>
<thead>
<tr>
<th>Periods</th>
<th>In-Stream Content 1</th>
<th>Ad 1</th>
<th>Ad 2</th>
<th>A3</th>
</tr>
</thead>
</table>

Segments

<table>
<thead>
<tr>
<th>Availability</th>
<th>Active Window</th>
</tr>
</thead>
</table>

Patch Update

No New Content
Return to Main Stream

In-Memory MPD

<table>
<thead>
<tr>
<th>Periods</th>
<th>In-Stream Content 1</th>
<th>Ad 1</th>
<th>Ad 2</th>
<th>A3</th>
</tr>
</thead>
</table>

Segments

| Availability | Active Window |

Patch Update

C2
Return to Main Stream

In-Memory MPD

Periods

Segments

Availability

Patch Update

Active Window

In-Stream Content 1 | Ad 1 | Ad 2 | A3 | C2

C2

C2
Early Cutback Scenario

Returning from a Replacement

Time

---

In-Memory MPD

Periods

| In-Stream Content 1 | Ad 1 | Ad 2 | A3 |

Segments

Availability

Active Window

Patch Update

C2
Early Cutback Scenario

Returning from a Replacement

Time

In-Memory MPD

Periods

Segments

Availability

Active Window

Patch Update
Late Cutback Scenario

Returning from a Replacement

In-Memory MPD

<table>
<thead>
<tr>
<th>Periods</th>
<th>In-Stream Content 1</th>
<th>Ad 1</th>
<th>Ad 2</th>
<th>A3</th>
</tr>
</thead>
</table>

Segments

Availability

Active Window

Patch Update

IA
## Late Cutback Scenario

### Time

<table>
<thead>
<tr>
<th>Periods</th>
<th>In-Stream Content 1</th>
<th>Ad 1</th>
<th>Ad 2</th>
<th>A3</th>
<th>IA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Segments</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Availability</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Patch Update</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### In-Memory MPD

- **In-Stream Content 1**
- **Ad 1**
- **Ad 2**
- **A3**
- **IA**
Server-Guided Ad Insertion

The Best of Both Worlds

- Not quite client side or server side
  - Client only uses manifest concepts
  - Server doesn’t perform stitching
- Exists in both LIVE and VOD
  - LIVE handled as described here
  - VOD handled with X-Link
- Strongly separates content and ads
  - Enables robust scalability
  - Requires a structured timing model
Server-Guided Ad Insertion

Today / Forward Looking

- Active in production today
  - Enabled across variety of channels
  - Internal and partner player support
- Biggest setback is SCTE-35 quality
  - Each broadcaster has own flavor
  - Working with partners to correct
- DASH Industry Forum Updates
  - Ad Insertion Interoperability Points
  - Reference Implementations
hulu

THANK YOU