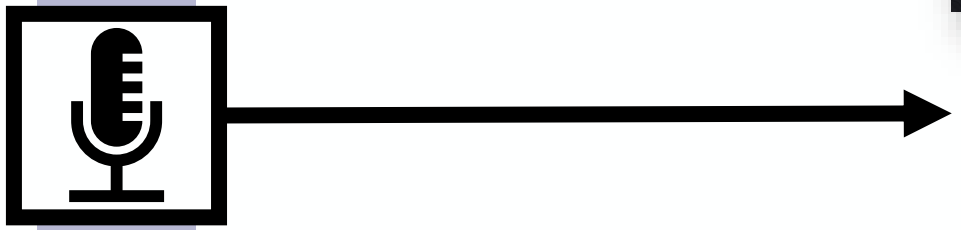
# DASH Content Steering

**Will Law (Akamai), Daniel Silhavy (Fraunhofer FOKUS)**



PRESENTER_PRIORITY:
[“Will”, “Daniel”]

Presenter Steering Server

DASH Industry Forum

# Content steering is a bit of a misnomer 😯

We're not actually steering the content.

In fact, there is only one version of the content.

We are steering between **sources** **CDNs** **that host and serve the content**.

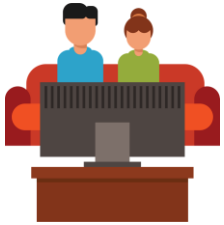
So content steering **is the art and science of switching CDNs**

DASH Industry Forum

# Why do content distributors want to switch CDNs?

- **Performance**
  - CDN performance and capacity varies dramatically with AS and time.
  - Don't start new users and don’t keep existing users on a poorly performing CDN
- **Capacity**
  - Switch users away midstream if a CDN is developing capacity problems due to competing traffic.
- Volumetric **contractual commits**
  - CDN A gets 35% of traffic
  - CND B gets 65% of traffic
- **Cost**
  - Price can vary by region and even time of day.

# Real Life Example



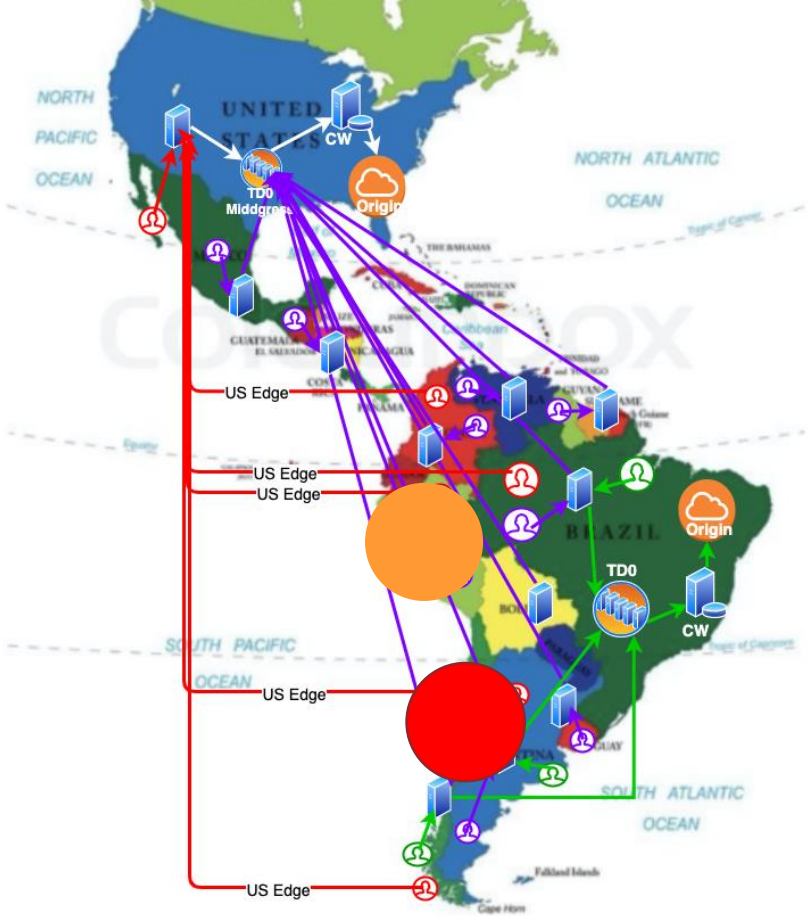
End user in Santiago Chile via Telefonica

- 51ms latency to Shield BR
- 157ms latency to Shield US-East

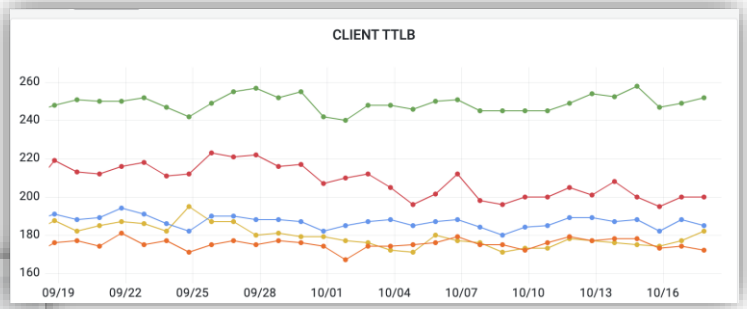
End user in Lima Peru via Claro

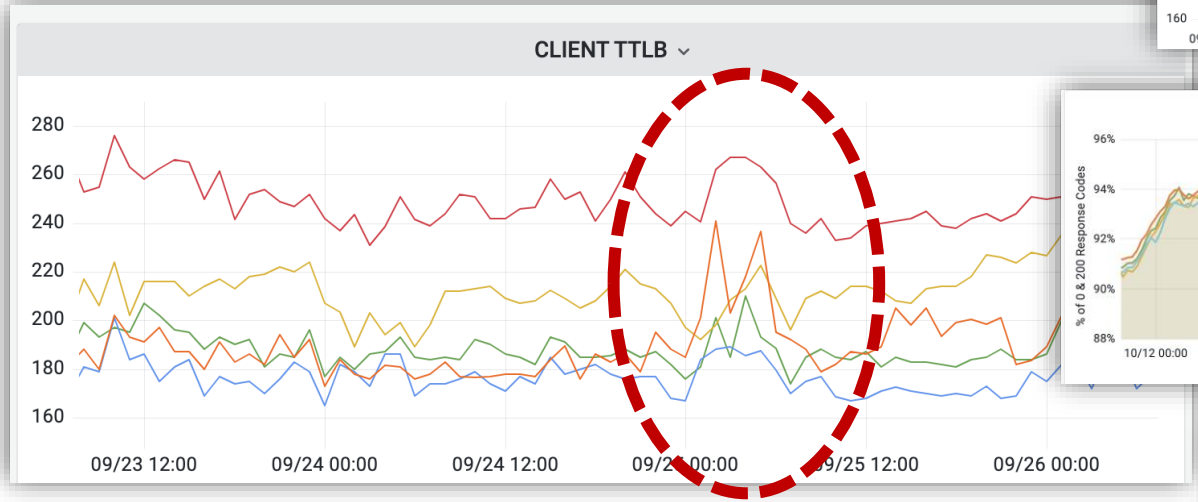
- 127ms latency to Shield BR
- 71ms latency to Shield US-East

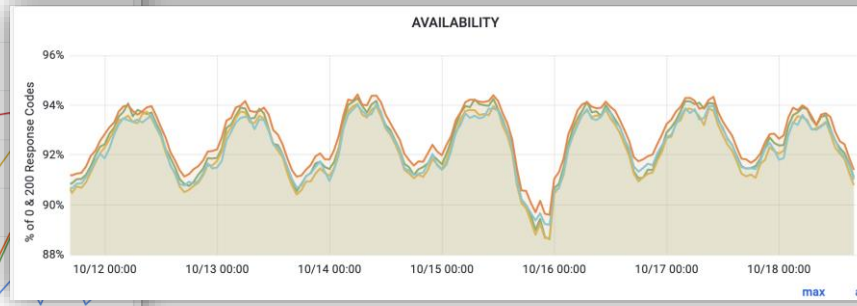
*latency represented by ping times
*Image credit Dylan Armajani

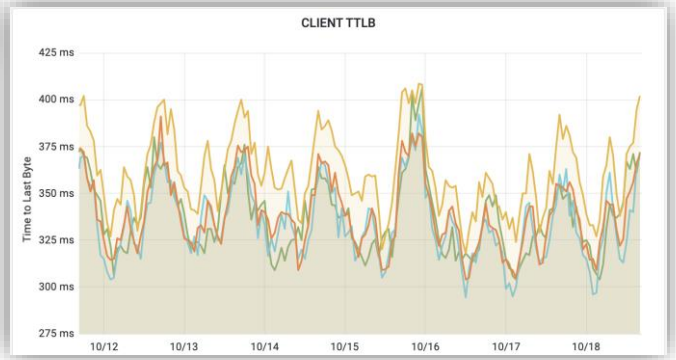


# CDN performance does vary by time



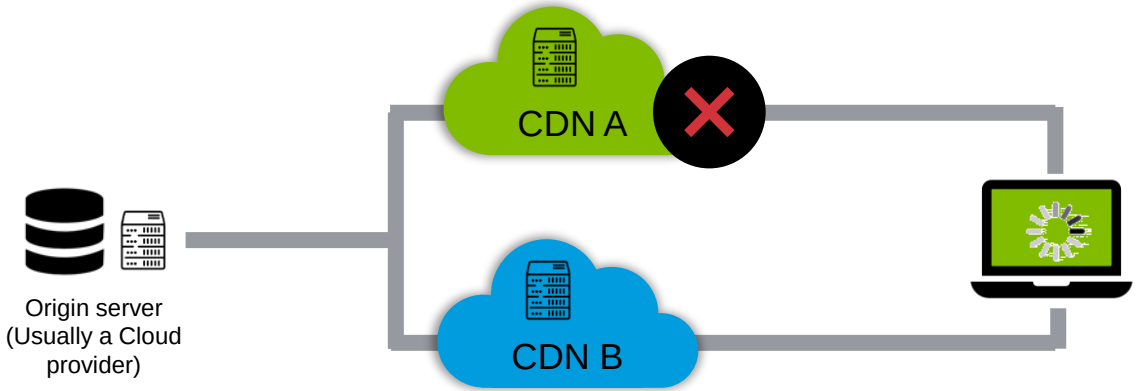




[Image data courtesy of Paramount]

For best performance during this 10-minute window, switch to blue CDN and away from orange

# But DASH already has BaseURL elements ..




https://content.com/manifest.mpd

```
<MPD … >
<Period>
<BaseURL>
https://cdnA.com
</BaseURL>
<BaseURL>
https://cdnB.com
</BaseURL>
...
</Period>
</MPD>
```

DASH Industry Forum

# DASH Player failover logic is indeterminate

When a player should switch to an alternate BaseURL is left to the discretion of the player.

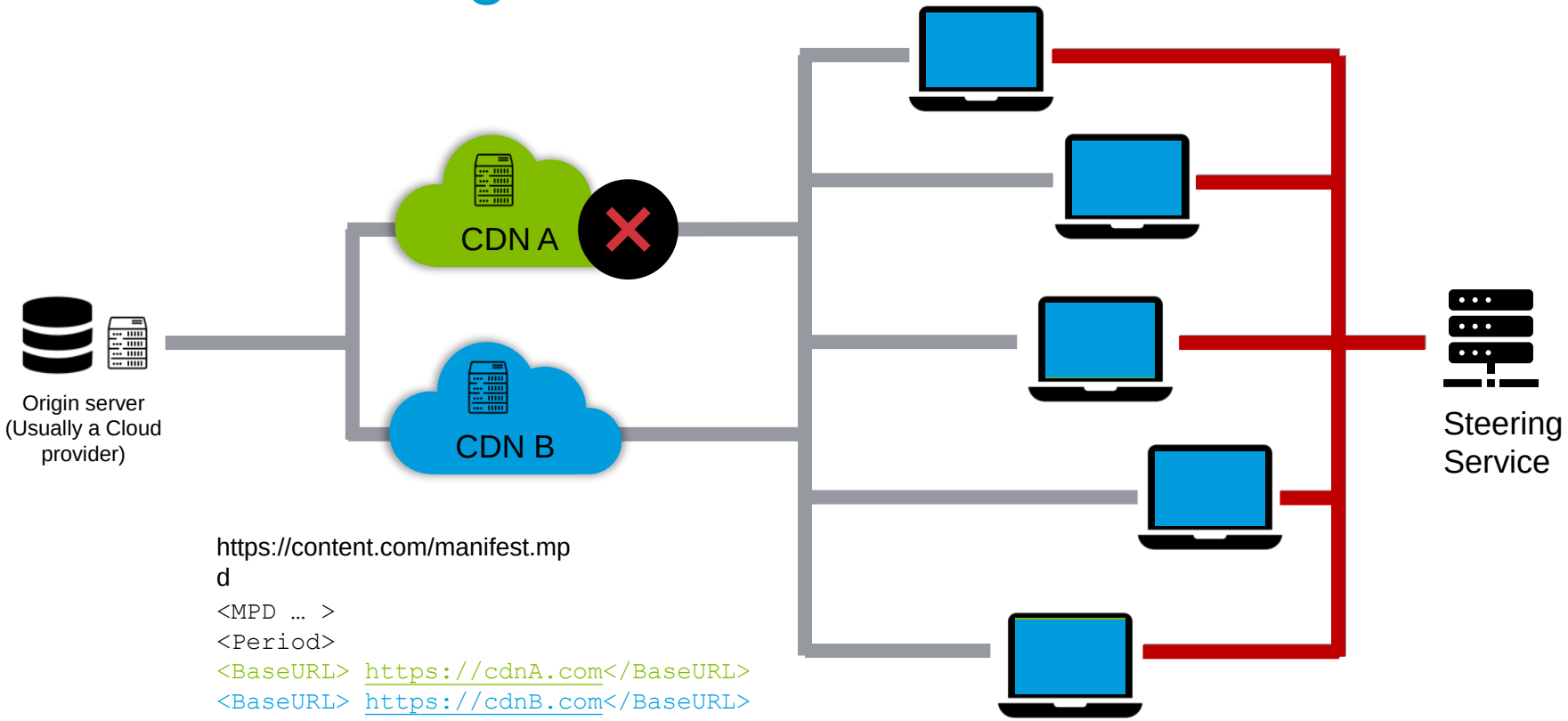
All sorts of behaviors exist in the wild:

- After receiving a 404, retry 3 times at the same bitrate then switch down to a lower bitrate
- After receiving a 404, switch bitrates immediately
- After receiving a 404, switch to an alternate BaseURL immediately at the same bitrate,
- After receiving a 404, re-request content indefinitely as long as buffer is above threshold.

**Behavior in response to slowness and lower buffer levels is completely undefined.**

**Large player populations are not deterministically controllable in the face of CDN slowness.**

# Content Steering




https://content.com/manifest.mpd

```
<MPD … >
<Period>
<BaseURL> https://cdnA.com</BaseURL>
<BaseURL> https://cdnB.com</BaseURL>
...
</Period>
</MPD>
```

DASH Industry Forum

# Who did the work to standardize DASH Content Steering?

DASH Industry Forum

- Founded in 2012 to promote and catalyze the adoption of MPEG-DASH and transition it from a standard to a deployed ecosystem.
- With more than 60 members, DASH-IF represents a large footprint of the ecosystem:
  - Service providers and broadcasters
  - CDN and client implementors
  - Technology providers
- DASH-IF serves as the point of contact/coordinator for other standards organizations using DASH-based distribution.

# How do we do it?

```
<MPD … >
<Period>
<BaseURL serviceLocation="alpha">https://cdnA.com</BaseURL>
<BaseURL serviceLocation="beta">https://cdnB.com</BaseURL>
...
</Period>
<ContentSteering defaultServiceLocation="beta"
queryBeforeStart="false">https://steeringservice.com/app/inst
ance1234</ContentSteering>
</MPD>
```

*Note that the ContentSteering element is the last element in the manifest*

DASH Industry Forum

# JSON Steering server response for DASH

```
{
"VERSION": number,
"TTL": number,
"RELOAD-URI": string;
"PATHWAY-PRIORITY": [".CDN
"PATHWAY-CLONES": [{"BA
}
```

REQUIRED – must be an integer

REQUIRED – number of seconds

OPTIONAL – URI

REQUIRED, array of serviceLocation identifiers

OPTIONAL, instructions to clone new serviceLocations

DASH Industry Forum

# Player workflow by example – step 1

```
<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="PT30S" timeShiftBufferDepth="PT30M" availabilityStartTime="2022-02-
25T12:30:00" minBufferTime="PT4S" profiles="urn:mpeg:dash:profile:isoff-live:2011">

<BaseURL serviceLocation="alpha">https://cdn1.example.com/</BaseURL>

<BaseURL serviceLocation="beta">https://cdn2.example.com/</BaseURL>

<Period id="1">

<AdaptationSet mimeType="video/mp4" codecs="avc1.4D401F" frameRate="30000/1001"

segmentAlignment="true" startWithSAP="1">

<BaseURL>video/</BaseURL>

…

</AdaptationSet>

</Period>

<ContentSteering defaultServiceLocation="beta" queryBeforeStart="true">
https://steeringservice.com/app/instance1234></ContentSteering>

</MPD>
```

DASH Industry Forum

# Player workflow by example – step 2

Player makes a request to

`https://steeringservice.com/app/instance1234`

Note that the _DASH_ parameters are not attached to this request, since the player has not yet started playback. The server responds with:

```
{
"VERSION": 1,
"TTL": 300,
"RELOAD-URI": "https://steeringservice.com/app/instance1234?session=abc"
"PATHWAY-PRIORITY": ["alpha","beta"]
}
```

The player would recognize that the highest priority serviceLocation specified is "alpha", so it would use the BaseURL construct of `https://cdn1.example.com/` as it begins to request content. It also sets a timer for 300s.

DASH Industry Forum

# Player workflow by example – step 3

After 300s, the player makes a request to

`https://steeringservice.com/app/instance1234?session=abc&_DASH_pathway=alpha&_DASH_throughput=5140000`

Note the two special reserved query args

`_DASH_pathway`: specifies the current serviceLocation

`_DASH_throughput`: specifies the current throughput in bps.

The use of _DASH_ differentiates the response from HLS players, which use _HLS_.

```
{
"VERSION": 1,
"TTL": 250,
"RELOAD-URI": "https://steeringservice.com/app/instance1234?session=abc"
"PATHWAY-PRIORITY": ["beta","alpha"]
}
```

The player would then switch to loading the next media objects using the BaseURL of https://cdn2.example.com/. 250s later it would again request the steering service and the cycle would continue until end-of-stream was reached.

DASH Industry Forum

# Can we steer ads separately from content? – Yes!

```
{
   "VERSION": 1,
   "TTL": 300,
   "RELOAD-URI":
"https://steeringservice.com/app/1234"
   "PATHWAY-PRIORITY":
["segments2","ad3","ad2","segments1","ad1","ad4"
]
}
```

As it enters each period, the player simply re-evaluates the BaseURL priorities.

```
<MPD ... >
  <BaseURL serviceLocation="segments1">http://cdn1.com/</BaseURL>
  <BaseURL serviceLocation="segments2">http://cdn2.com/</BaseURL>
  <Period ... (primary content)/>
  <Period ...(ad)>
      <BaseURL serviceLocation="ad1">http://adcdn1.com/</BaseURL>
      <BaseURL serviceLocation="ad2">http://adcdn2.com/</BaseURL>
      ...
  </Period>
  <Period ... (primary content)/>
  <Period ...(ad)>
      <BaseURL serviceLocation="ad1">http://cdn1.com/</BaseURL>
      <BaseURL serviceLocation="ad2">http://cdn2.com/</BaseURL>
      ...
  </Period>
  <Period ... (primary content)/>
  <Period ...(ad)>
      <BaseURL serviceLocation="ad3">http://adcdn1.com/</BaseURL>
      <BaseURL serviceLocation="ad4">http://adcdn4.com/</BaseURL>
      ...
  </Period>
   <ContentSteering>https://steeringservice.com/app/1234</ContentSteering>
</MPD>
```

DASH Industry Forum

# Can we steer the manifest refreshes themselves?

**Yes – by adding a serviceLocation identifier to the existing <Location> element.**

```
<MPD ... >
   <BaseURL serviceLocation="segments1">http://cdn3.com/</BaseURL>
   <BaseURL serviceLocation="segments2">http://cdn4.com/</BaseURL>
   <Location serviceLocation="mpd1">https://ssai1.com/manifest.mpd</Location>
   <Location serviceLocation ="mpd2">https://ssai2.com/manifest.mpd</Location>
<ContentSteering defaultServiceLocation="alpha">https://steeringservice.com/app</ContentSteering>
   <Period .../>
</MPD
```

**Steering server would return**

```
{
  "VERSION": 1,
  "TTL": 300,
  "RELOAD-URI": "https://steeringservice.com/app/instance12345?session=abc"
  "SERVICE-LOCATION-PRIORITY": ["mpd2","segments2"]
}
```

DASH Industry Forum

# Useful things to know about DASH Content Steering

1. If the player encounters a delivery error, it's first job is to try and maintain the best QoE for the end-user.
2. DASH reports the active serviceLocation and throughput using the reserved _DASH_pathway and DASH_throughput parameters. Unlike HLS, these allow the reporting of multiple values.
3. DASH has the ability to force a player to query the steering server before starting playback. This allows steering to be used to update old manifests in the field that point to content sources that no longer exist.

DASH Industry Forum

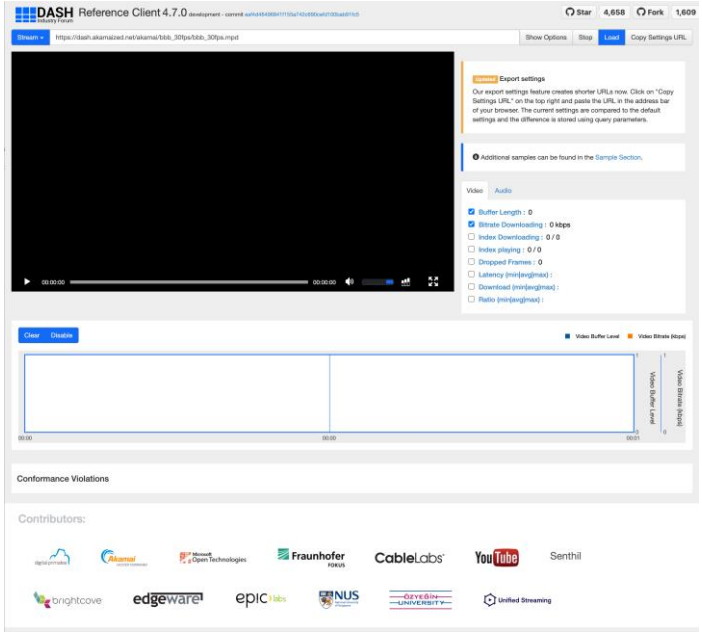
PRESENTER_PRIORITY:
[“Daniel”, “Willl”]

Presenter Steering Server

DASH Industry Forum

# dash.js - DASH-IF Reference Client

- dash.js is the official reference player by the **DASH Industry Forum** for playback of MPEG-DASH content
- Started in 2012, maintained by Fraunhofer FOKUS, community driven development
- Open-source project on Github with various features such as ABR, multiperiod, DRM, MPD patching, CMCD, CMSD, CMAF low latency, content steering and many more
- Application Areas:
  - **Reference platform**: Verify new features, also used by other organizations such as CTA-WAVE, DVB, HbbTV, 5G-MAG
  - **Production**: Used by the industry (e.g. BBC, Deutsche Telekom)
  - **Research**: Test and compare new ABR algorithms (Twitch challenge)



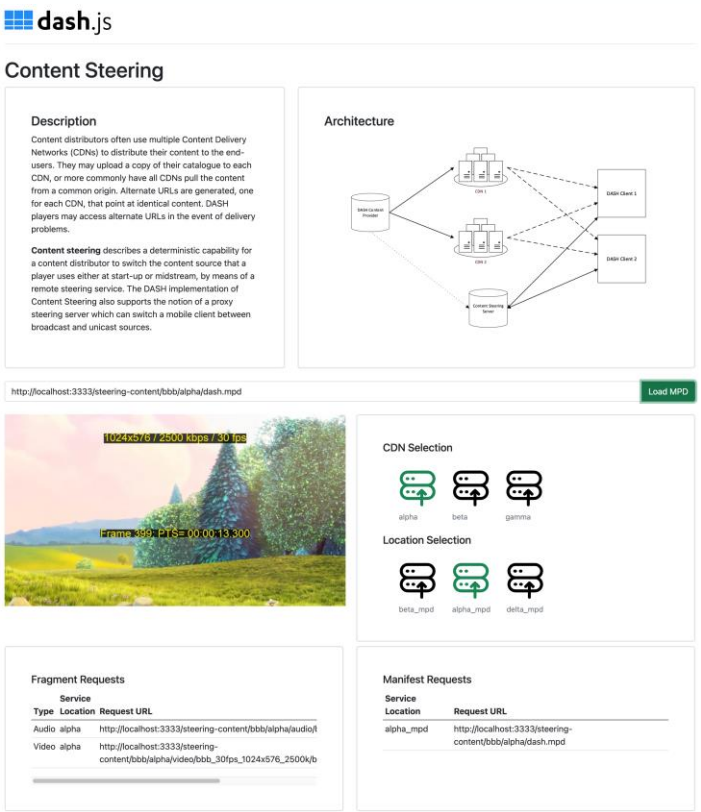

https://reference.dashif.org/dash.js/nightly/samples/dash-if-reference-player/index.html

https://github.com/Dash-Industry-Forum/dash.js

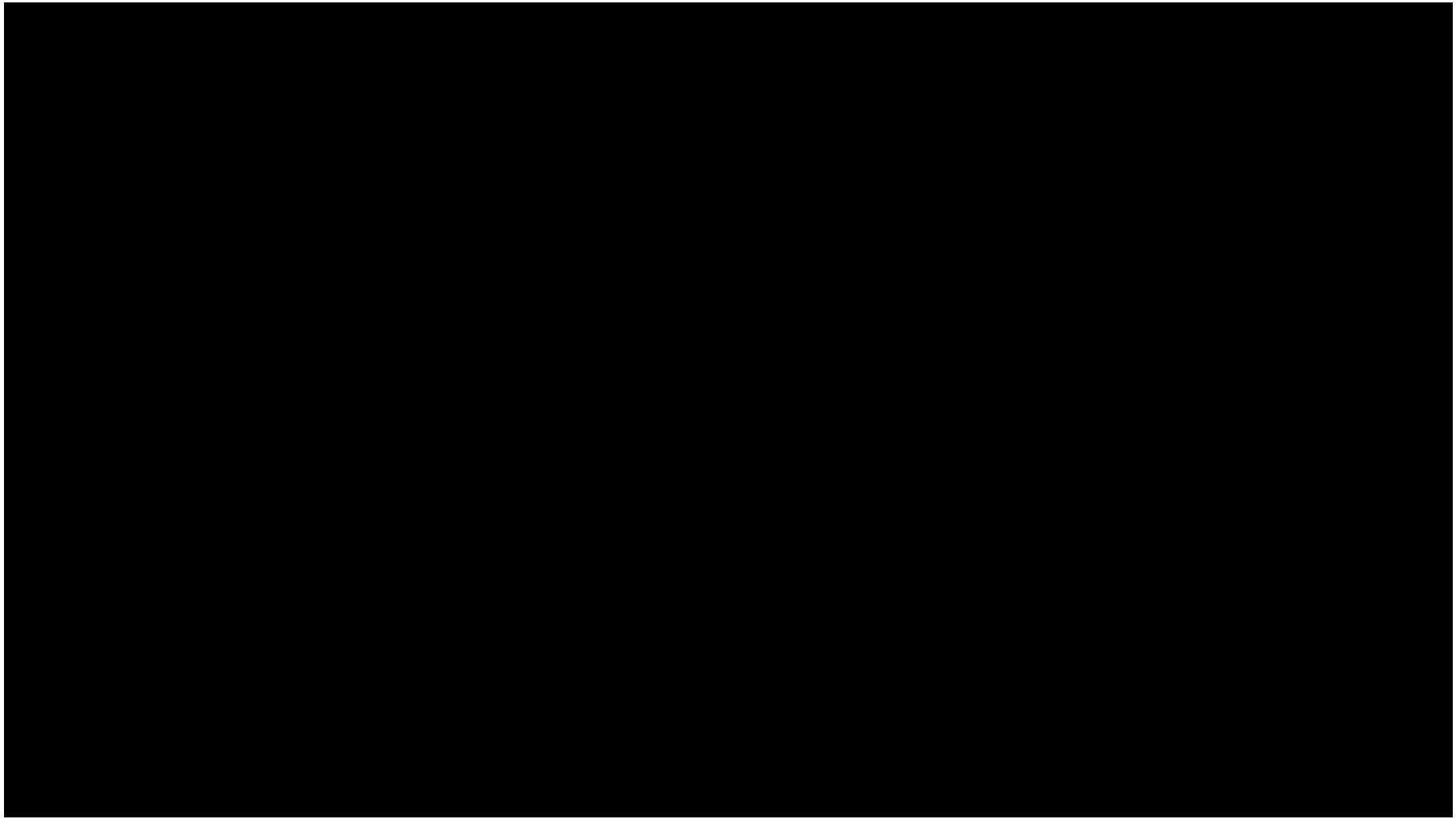
DASH Industry Forum

# Content Steering in dash.js

- An early version of the content steering specification was added in dash.js v4.5.0 (reference implementation)
- Version 4.7.0 updates the implementation including support for
  - Pathway cloning
  - Steering <BaseURL>, <*Location*> and <*PatchLocation*>
  - Comma separated list of the throughput values for all service locations using *DASH_throughput* and *DASH_pathway*
  - Content Steering via <ServiceDescription> elements
  - Handling server-side errors



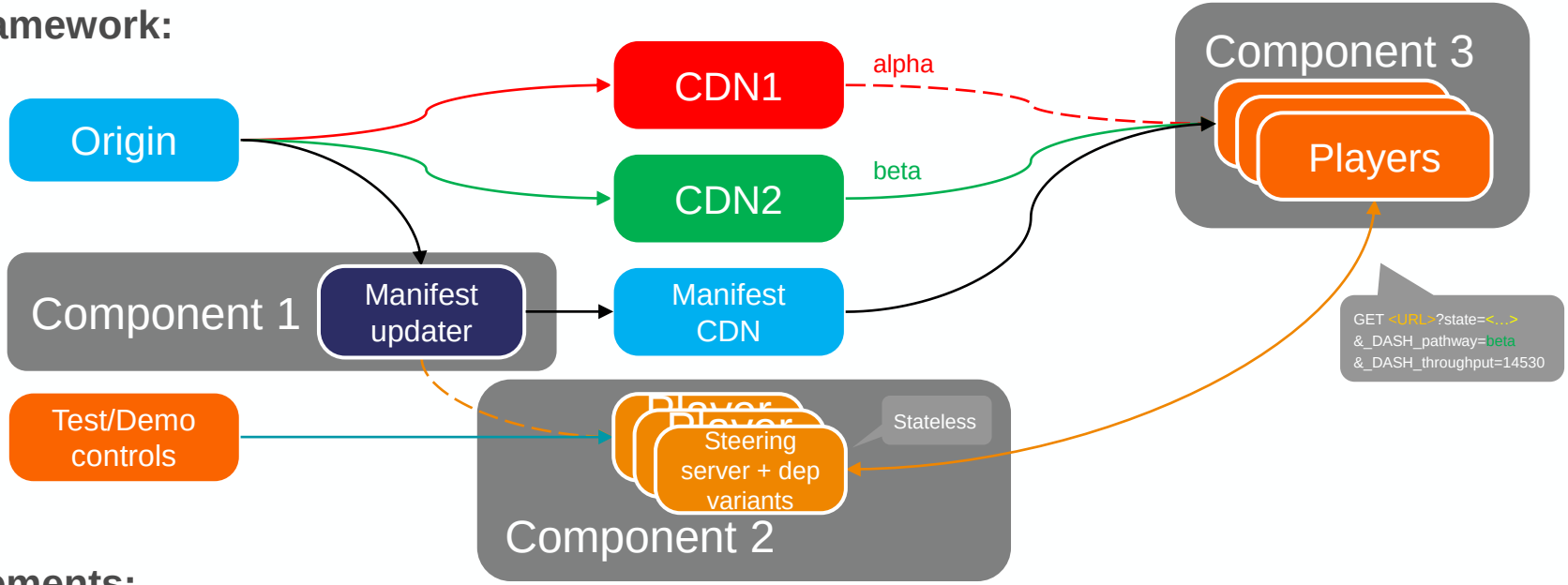

https://reference.dashif.org/dash.js/nightly/samples/advanced/content-steering.html



BRIGHTCOVE

# Open Source Proposal for SVTA

## Framework:




## Elements:

- Component 1: manifest updater inserting content steering information in the manifests (Golang)
- Component 2: steering server implementation with several deployment variants supported:
  - standalone server, using AWS Lambdas, Lambda @ Edge, etc. (JS)
- Component 3: DASH and HLS players supporting content steering: DASH.js and HLS.js – existing open source projects

DASH Industry Forum

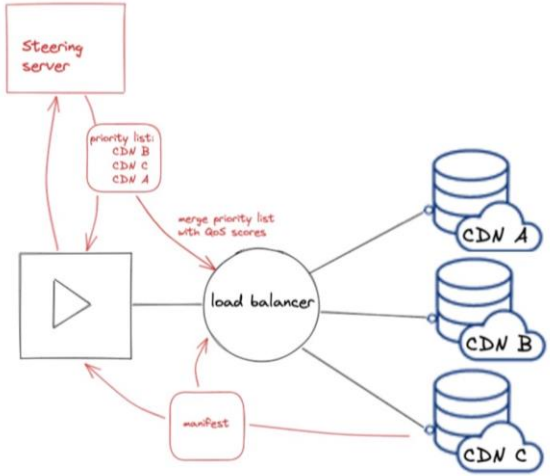
(Slide credit: Guillem Cabrera and Yuriy Reznik)

© Brightcove Inc. All Rights Reserved.

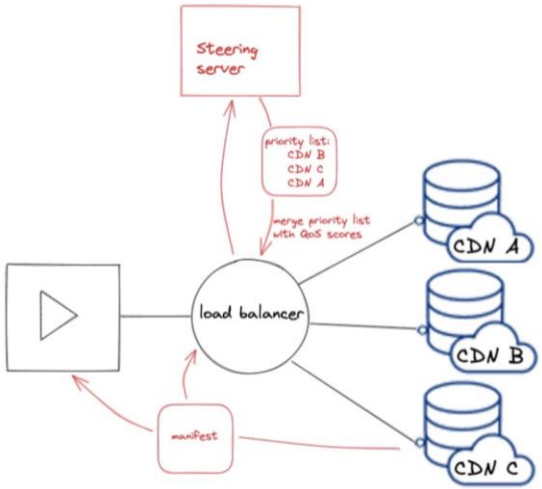
# SVTA – CDN load balancer – Harmonizing client and server-side approaches

LUMEN®

- SVTA Players & Playback WG is working on a CDN load balancer
- Load balancer complements steering server by adding micro-switching on the individual viewer level
- Merges steering server priorities with client-side QoS scores for the final decision




*Player communicates with load balancer and steering server*




*Load balancer can also communicate with the steering server even if the player does not support steering.*

DASH Industry Forum

(Slide credit: Sergey Arsenyev)

# dash.js face-to-face

- Next dash.js face-to-face on June 12th in Berlin
- Co-located with the Media Web Symposium 2023
- Join us in Berlin!
  - https://www.eventbrite.de/e/dashjs-face-to-face-meeting-2023-tickets-597581391027
  - https://mws10.eventbrite.com/

DASH Industry Forum

# Contact

- Will Law
- Email: wilaw@akamai.com
- LinkedIn: https://www.linkedin.com/in/wilaw/

- Daniel Silhavy
- Email: daniel.silhavy@fokus.fraunhofer.de
- LinkedIn: https://www.linkedin.com/in/daniel-silhavy-21650a129/
- Twitter: https://twitter.com/dsilhavy
- Blog: https://websites.fraunhofer.de/video-dev/

DASH Industry Forum