

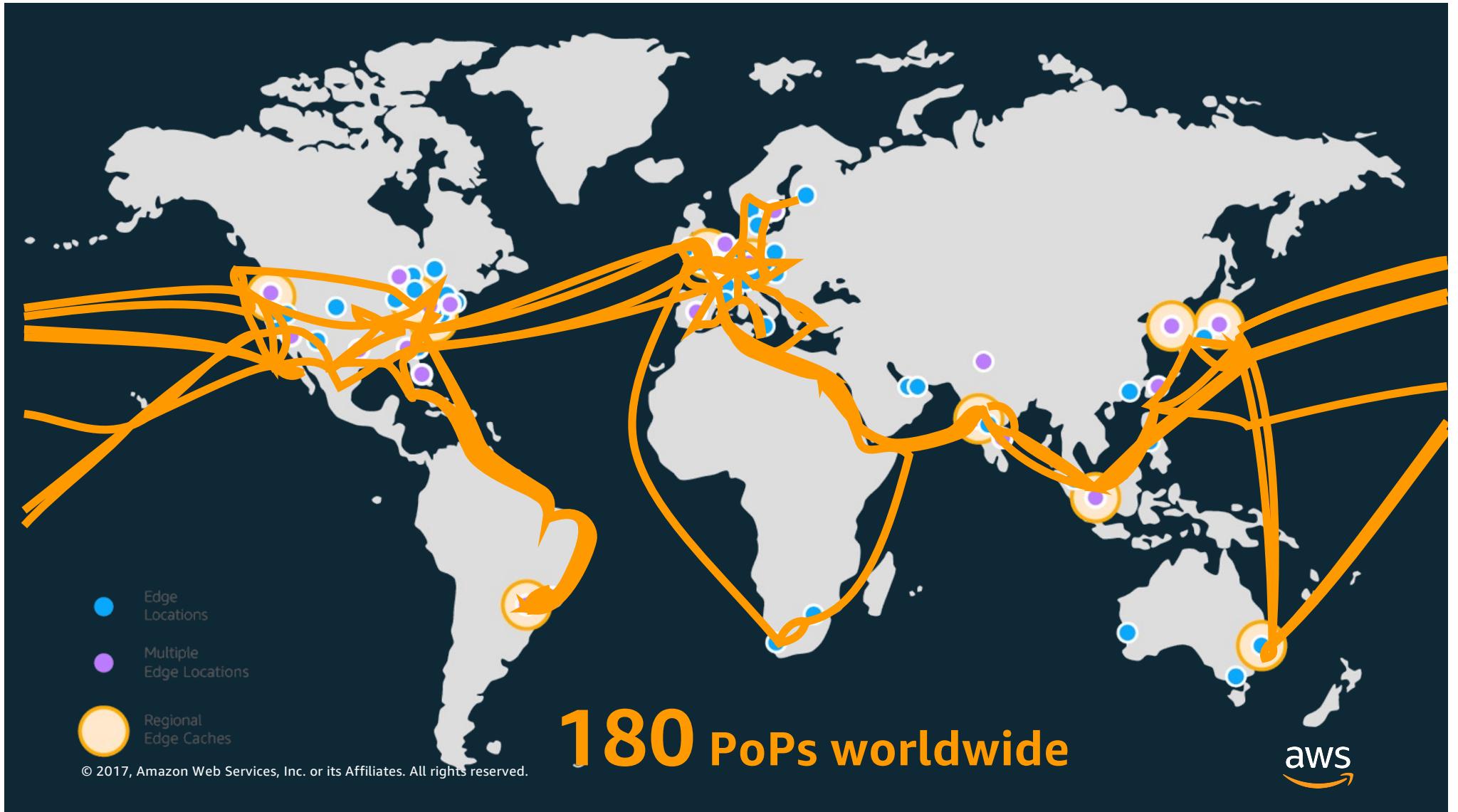


# Video Streaming using CloudFront

## Chalk Talk – M&E Symposium, Stockholm

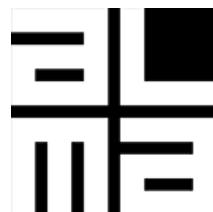
Achraf Souk, Solutions Architect, Edge Services

12/06/2019



**a**cast®

s a n o m a



**BONNIER**  
Broadcasting

 **Readly**

 **xstream**  
GET AHEAD  
A SeaChange Company

 **bam  
bus  
er**



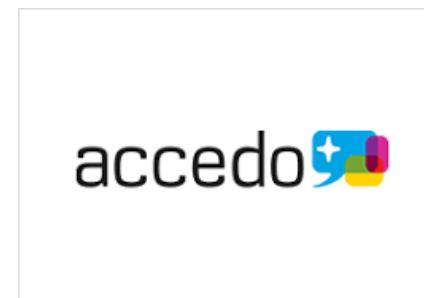
**MaineTV**

**screen9**

 **SCHIBSTED**  
MEDIA GROUP

**VIMOND** 

 **YOUSICIAN**

 **accedo** 

 **BLOCKBUSTER®**

# Lambda@Edge

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

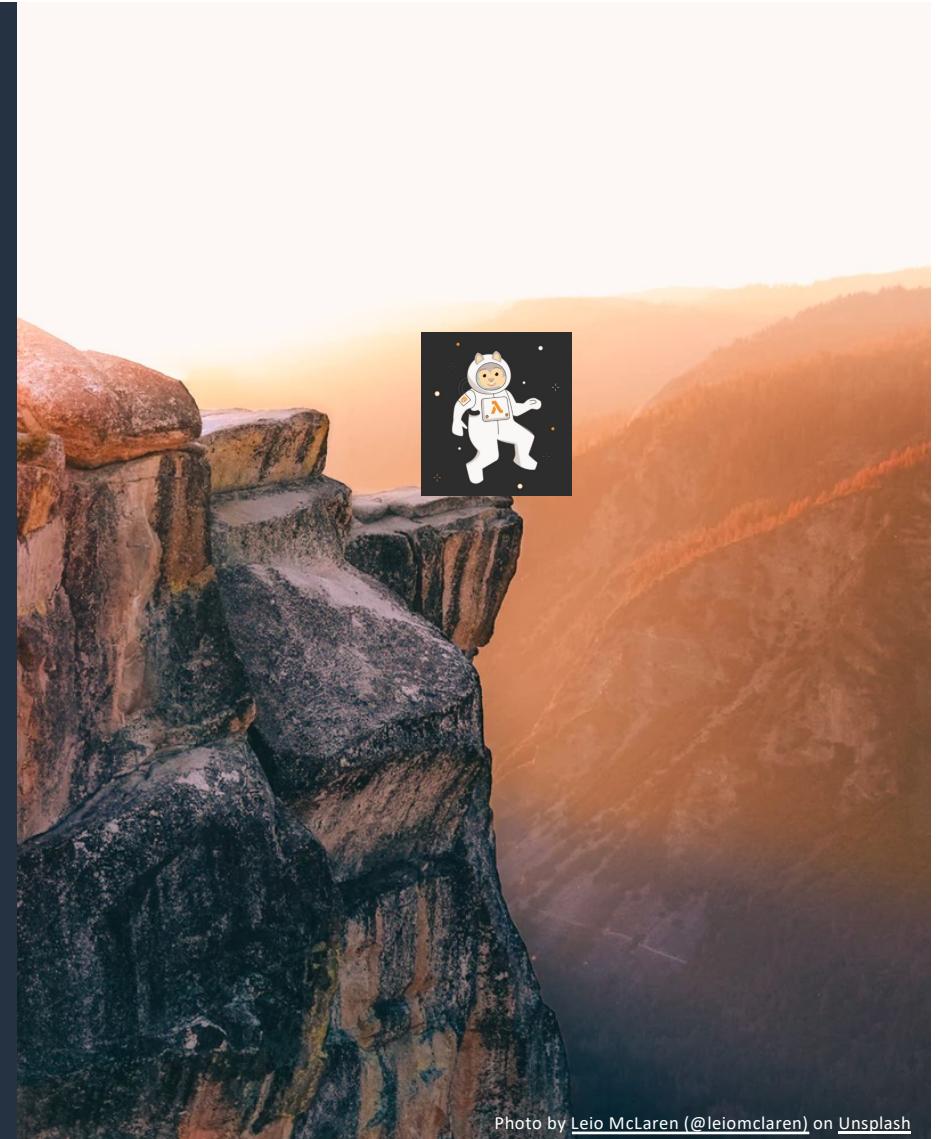
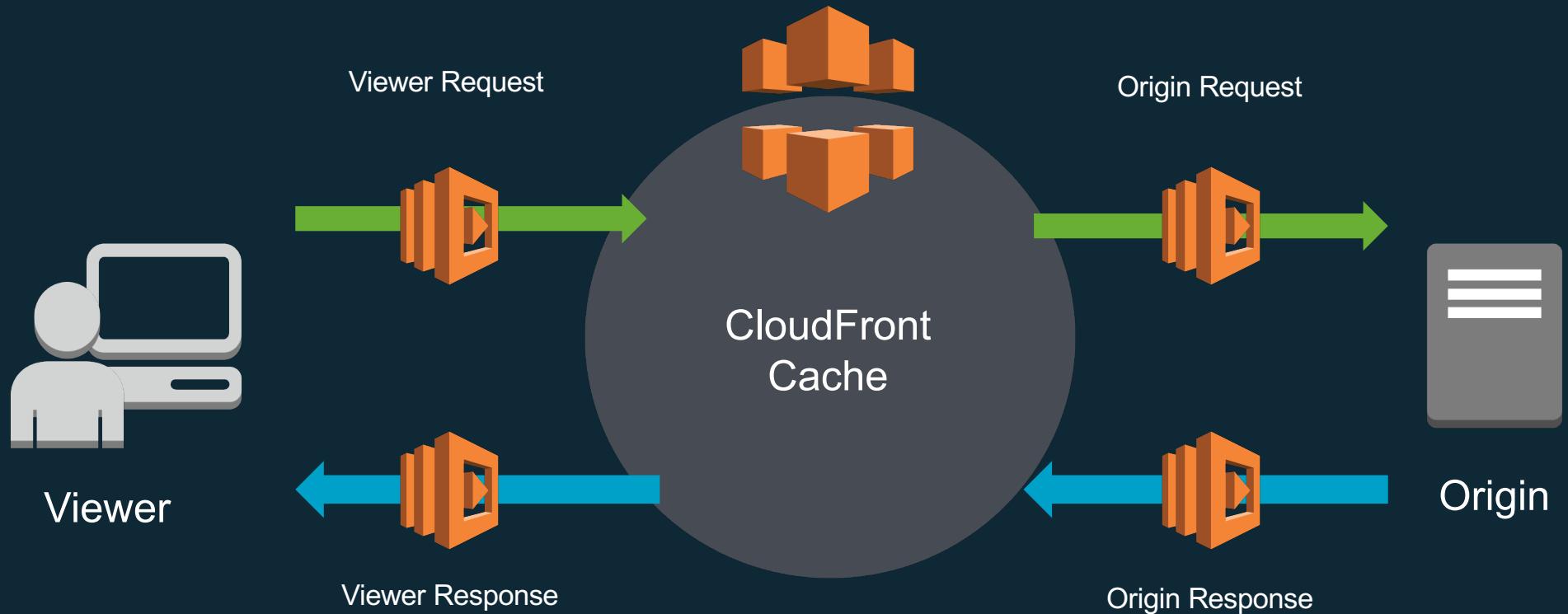


Photo by [Leio McLaren \(@leiomclaren\)](#) on [Unsplash](#)

# You said Lambda@Edge?



# Manifest manipulation

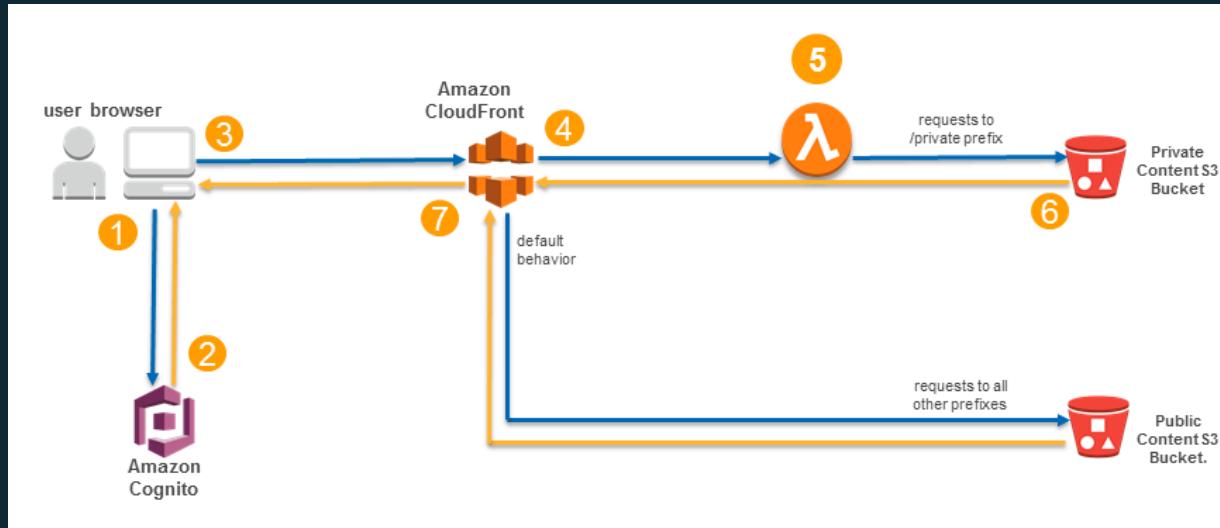
BLOCKBUSTER®

- TVOD provider with audience in the Nordics.
- Legacy packager that produced MSS manifests that were not digested by Smart TVs
- Temporary solution based on Lambda@Edge:
  - Triggered on MSS manifest requests
  - Remove high bitrates (>5MBps)
  - Splits multi-quality audio in to multiple tracks
  - Strip undesired elements "<F"
  - <https://github.com/blockbuster/manifest-proxy>

## Custom authorization (1/2)

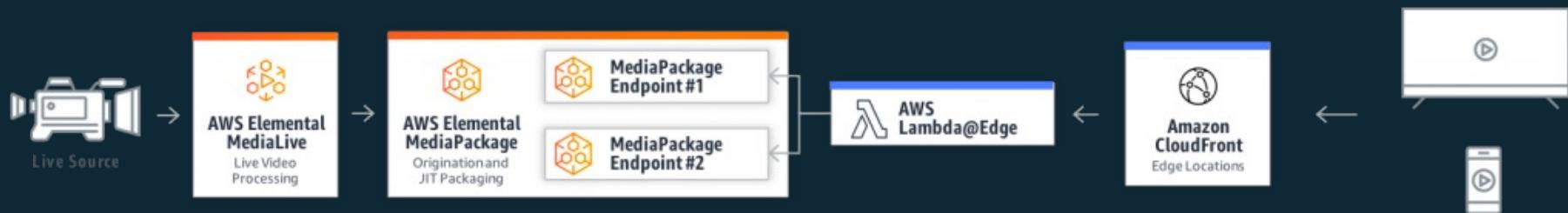
- *Broadcaster* in Finland provides Live Streaming for local audience.
- Looking to implement custom authentication based on JWT token to avoid using cookies.
- Solution based on Lambda@Edge:
  - Validate JWT token on manifests and rewrite manifest with tokenized URLs
  - Leverage native CloudFront URL tokenization for video segments

# Custom authorization – Amazon Cognito (2/2)



<https://aws.amazon.com/blogs/networking-and-content-delivery/authorizationedge-how-to-use-lambdaedge-and-json-web-tokens-to-enhance-web-application-security/>

# How to filter live streaming renditions by device type at the edge (1/2)



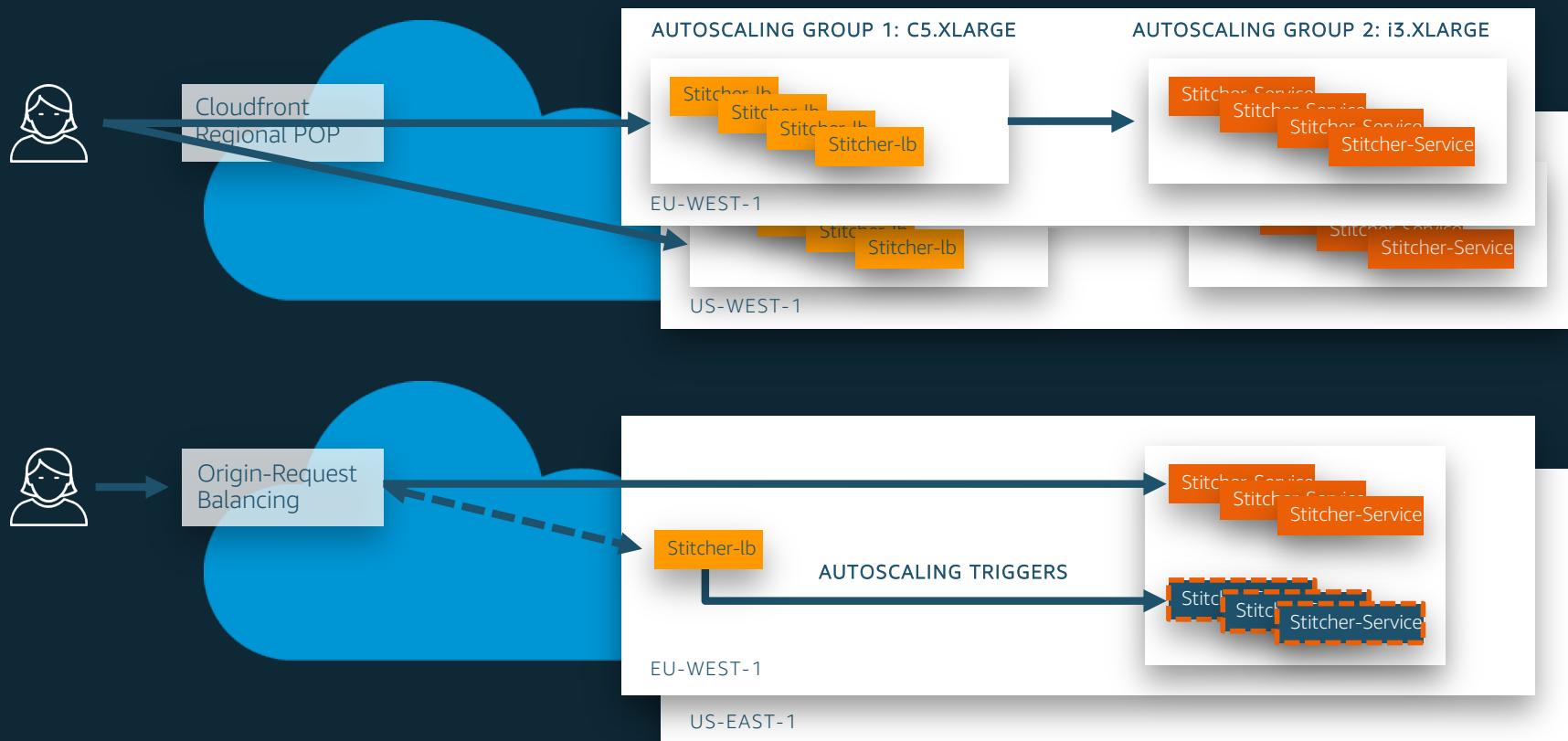
```
exports.handler = (event, context, callback) => {
  const request = event.Records[0].cf.request;
  const headers = request.headers;

  const bigScreenManifest = "8ebc56efb8ab4b86bb27b76126b4c8de"
  const smallScreenManifest = "7360a56565a1455b92424aaf8a3dc4aa"
```

# How to filter live streaming renditions by device type at the edge (2/2)

```
var reqSplit = request.uri.split("/")
if ((headers['cloudfront-is-desktop-viewer']
&& headers['cloudfront-is-desktop-viewer'][0].value === 'true') ||
(headers['cloudfront-is-smarttv-viewer']
&& headers['cloudfront-is-smarttv-viewer'][0].value === 'true'))
{
    reqSplit[3] = bigScreenManifest
    request.uri = reqSplit.join("/")
} else {
    reqSplit[3] = smallScreenManifest
    request.uri = reqSplit.join("/")
}
callback(null, request);
};
```

# Loadbalancing to origin



# Performance

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Photo by [Bill Jelen](#) on [Unsplash](#)

# What CloudFront already does on your behalf

- Consistent hashing in Edge locations
- Regional Edge Caches
- AWS backbone
- Persistent connections
- Forward collapsing
- Object streaming

# Low latency streaming

- Simplestream Ltd
- Encoder -> Media Store (chunked object transfer) -> CloudFront
- "Chunked CMAF allows us to offer our customers another cutting-edge solution to help solve their problems. From live sports to teleshopping – latency is an issue. Now we can we eliminate this headache with an end-to-end solution, from ingest through to the end user; **with sub 3 second latency!** »
- <https://aws.amazon.com/blogs/media/lower-latency-with-aws-elemental-mediastore-chunked-object-transfer/>

# How you can increase performance

- Improve Cache Hit Ratio:
  - Cache Control headers on origin or/and Minimum TTL on CloudFront – Multiple behaviors – Value forwarding
  - Enable CORS intelligently
  - Override Accept-Encoding header
  - Consider prefetching
- Replicate long tail content closer to users
- Leverage connection reuse/coalescing

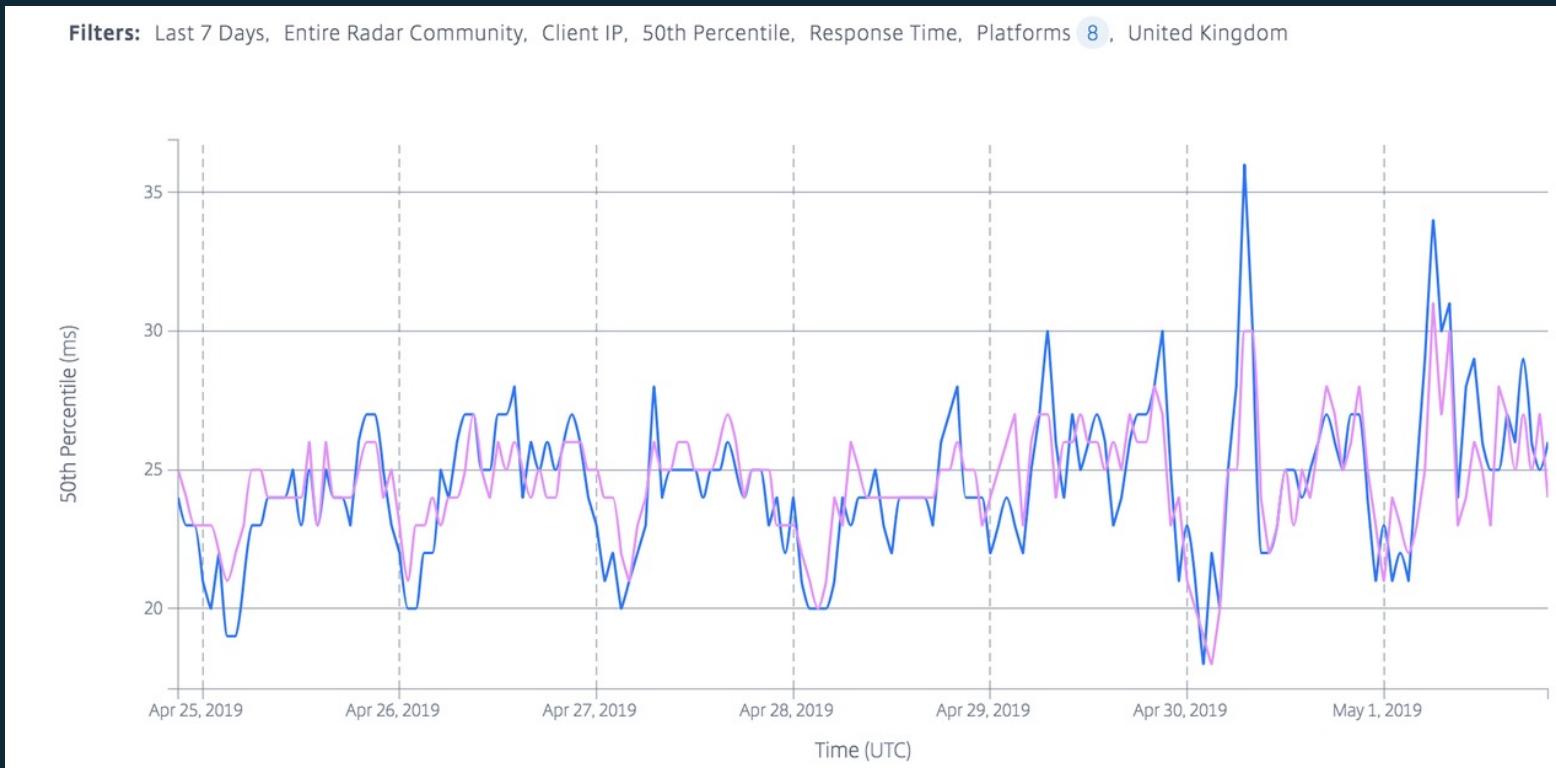
# Multi-CDN

© 2017, Amazon Web Services, Inc. or its Affiliates. All rights reserved.



Photo by [Vadim Fomenok](#) on [Unsplash](#)

# Why Multi-CDN?



# Balance benefits with drawbacks

- Multi-Vendor overhead (technical & non-technical)
- Cost reduction myth
- Feature parity
- Investment in Load-balancing
- Lower Cache Hit Ratio

# Step 1 : Measuring performance

- Third party Real User Measurements (RUM) like Cedexis, easy to set up but biased:
    - Different configuration (not necessarily same maps, optimizations..)  
-> Private Measurements
    - Hot objects
    - 100KB size test
    - Availability with 2 secs timeout
  - Client side measurements (QoE or QoS) but requires integration. <https://fr.slideshare.net/AmazonWebServices/build-an-aws-analytics-solution-to-monitor-the-video-streaming-experience-mae309-aws-reinvent-2018>
- Use QoE for routing & RUM when not enough data

## Step 2: Scoring

- Common metrics: Buffering, play failures, bitrates and start-up time.
- Dimensions: at least ASN
- Aggregation: median & P90
- Aggregation period: minutes & tens of minutes

## Step 3: Routing

- HTTP vs DNS
- Failover
- Keep CDNs warm
- Switching thresholds & Incremental traffic shifts
- CDN capacity & Commits
- Pattern prediction?

# Case Study: Amazon Prime Video



- Long tail SVOD/TVOD catalog (10/90)
- Own client side analytics
- Metrics: percentage of sessions without re-buffering, the percentage of sessions without fatal errors, bitrates, response times and time to first frame.
- Dimensions: ASN, Geo & Device
- Aggregation: 5 mins, 1 hour.
- <https://www.youtube.com/watch?v=2YAz20284FI>



Thank you

Achraf Souk

[achrsouk@amazon.com](mailto:achrsouk@amazon.com)