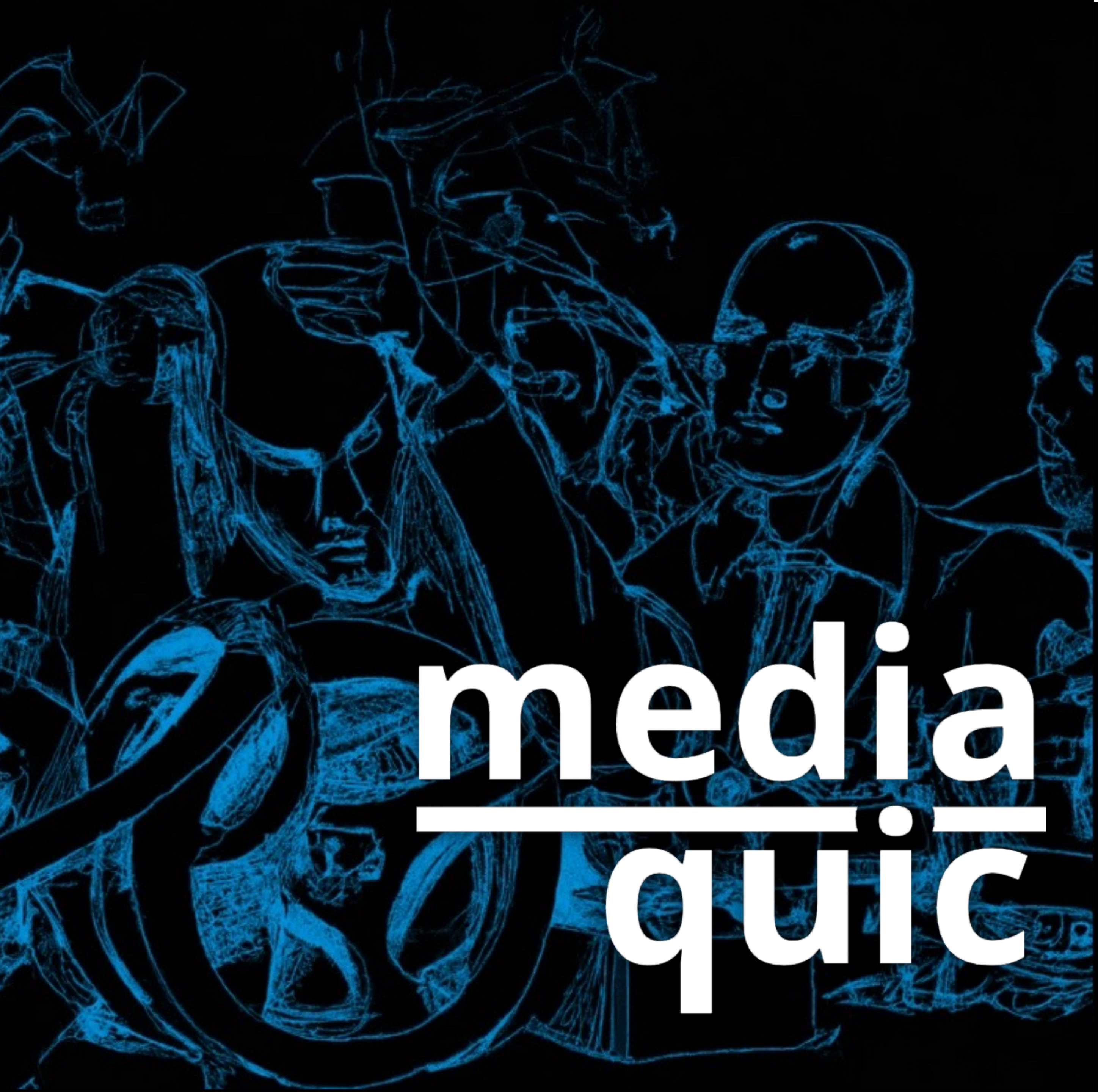


MoQ: Unifying Real-Time Communications and Content Delivery

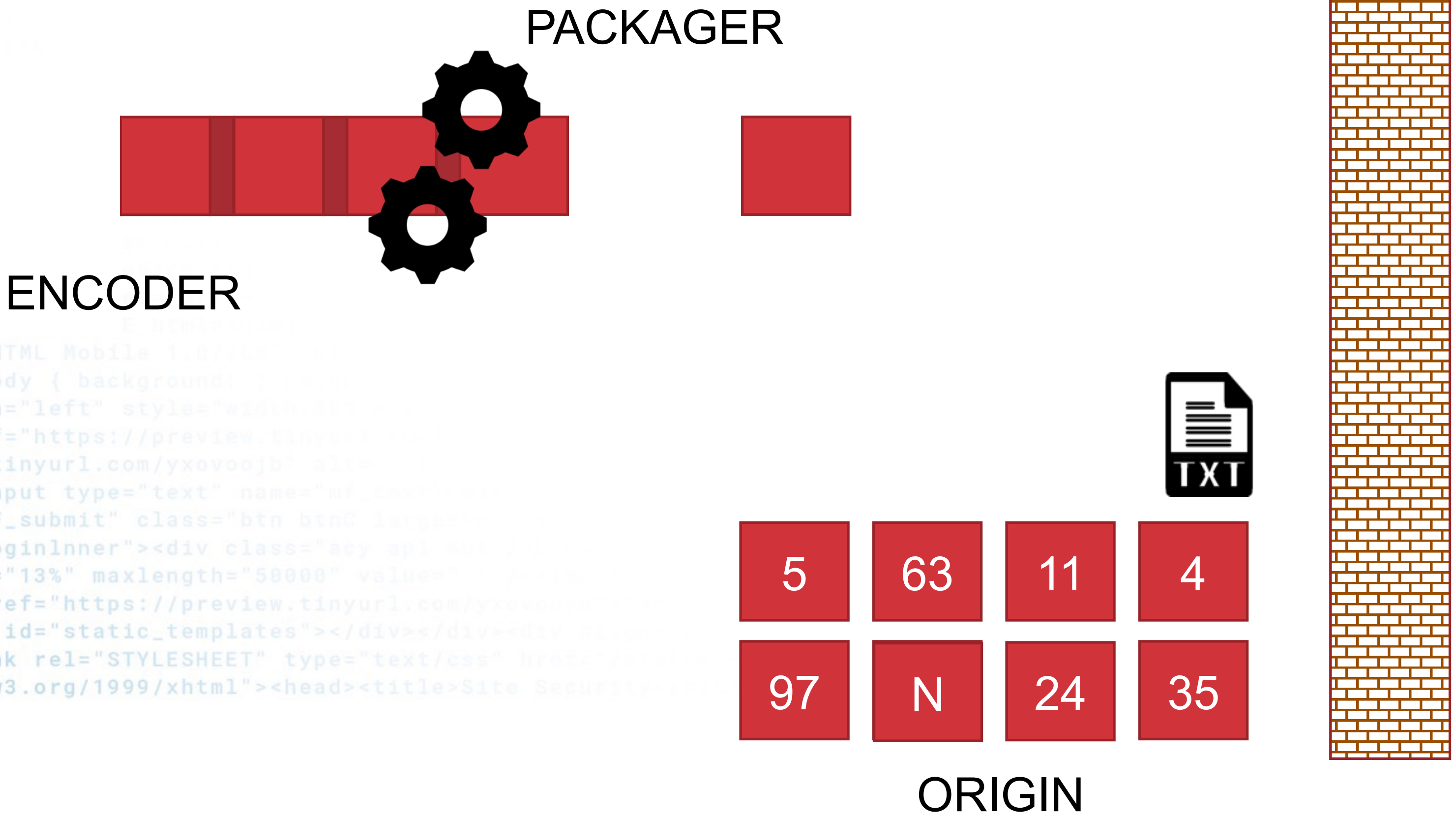
Will Law
Akamai

Ali C. Begen
Özyeğin University

Dec 2023

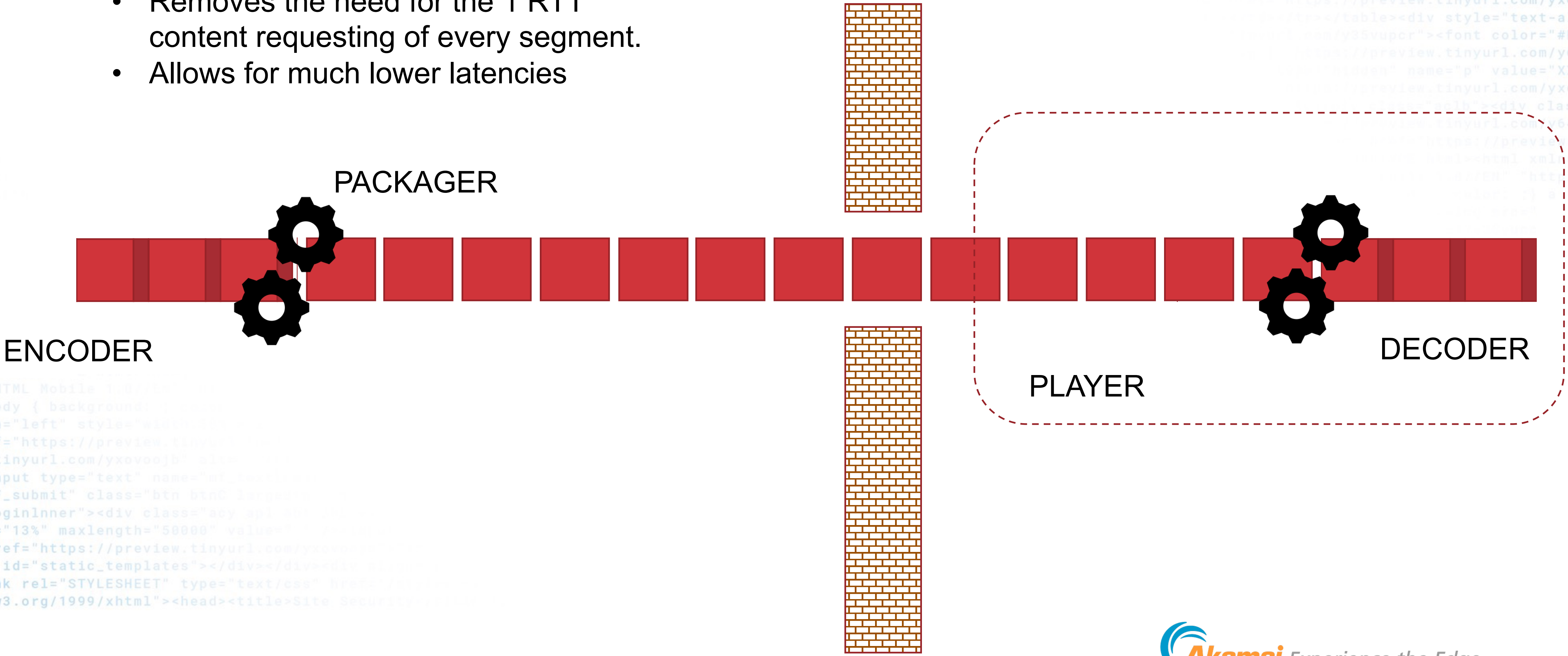
ONE DOES NOT SIMPLY TURN IT ON
AND ACHIEVE LOW LATENCY

The curious case of HTTP Streaming and the Lost Sequence Information



Pushing the content directly to the receiver

- Removes the need for the 1 RTT content requesting of every segment.
- Allows for much lower latencies



Akamai Experience the Edge

Why did Pub/Sub get replaced by HAS?

1. Not designed for **distribution via multi-tenant 3rd party networks** (CDNs)
2. **Live edge only,** with no support for behind-live and VOD playback use-cases.
3. Focused on **contribution or distribution**, but not both.
4. **Vendor proprietary** solutions versus open global standards
5. Tight **binding of codecs and media formats** to the transport solution.

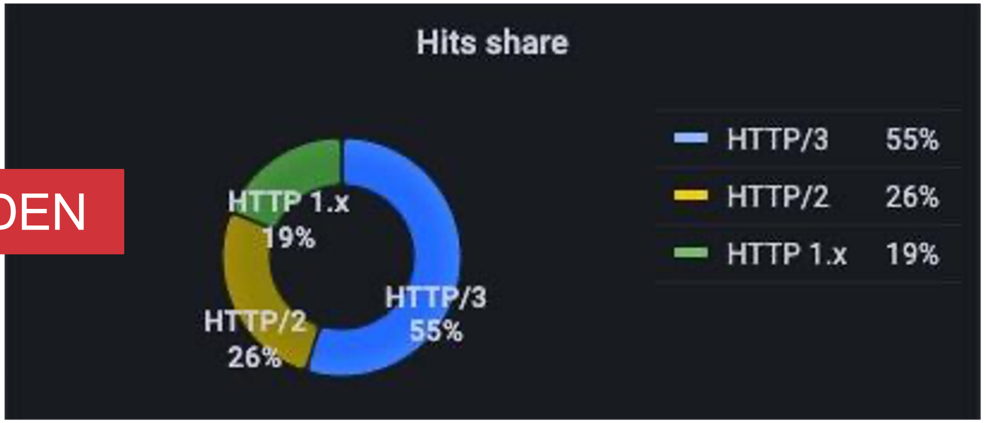
Akamai Experience the Edge

If we want QUIC, why not just use HTTP/3 with HLS/DASH?

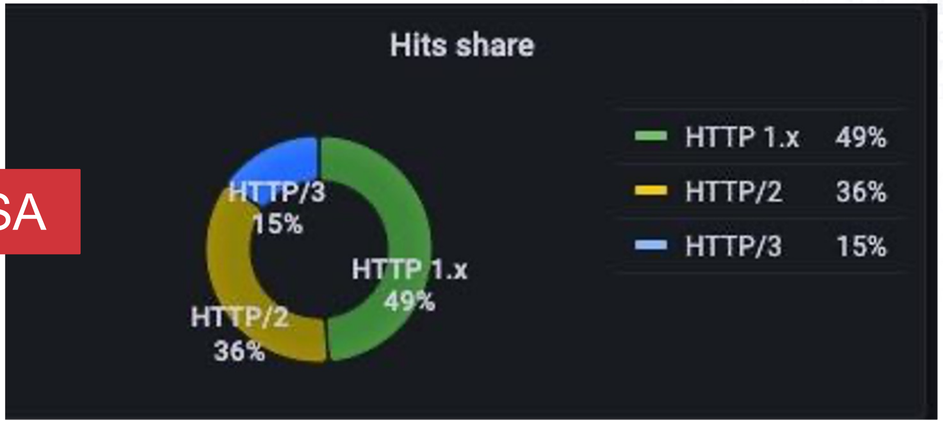
HTTP/3 Perf - real world data from Akamai network

Data taken on Akamai AMD network, March 7-20 for a large media conglomerate.

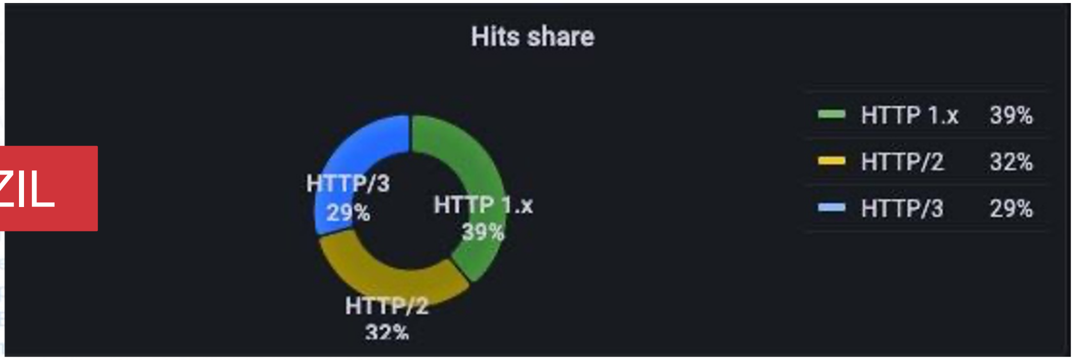
SWEDEN



USA



BRAZIL

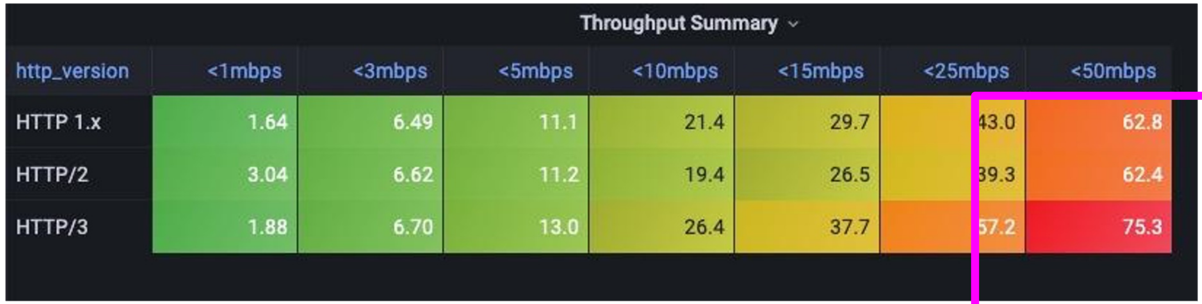


HTTP/3 Perf - real world data from Akamai network

Data taken on Akamai AMD network, March 7-20 for US media conglomerate.

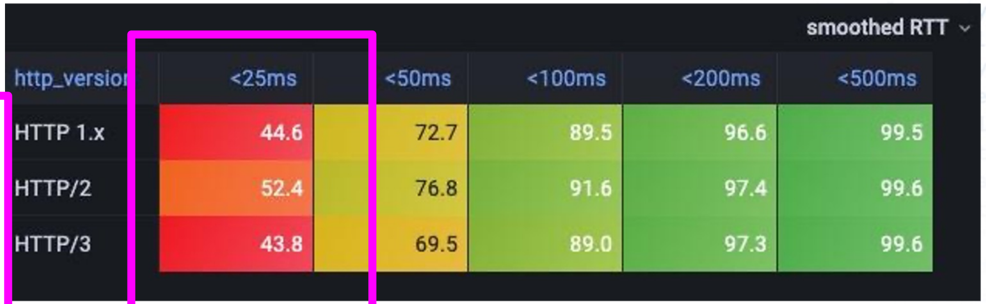
Note – we constantly update our HTTP stack and these results are not replicable or transferable to other delivery properties.

SWEDEN



Throughput Summary

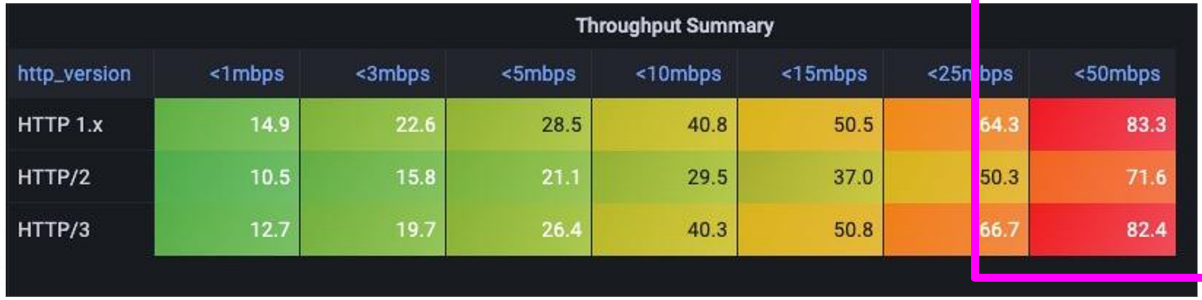
http_version	<1mbps	<3mbps	<5mbps	<10mbps	<15mbps	<25mbps	<50mbps
HTTP 1.x	1.64	6.49	11.1	21.4	29.7	43.0	62.8
HTTP/2	3.04	6.62	11.2	19.4	26.5	39.3	62.4
HTTP/3	1.88	6.70	13.0	26.4	37.7	57.2	75.3



smoothed RTT

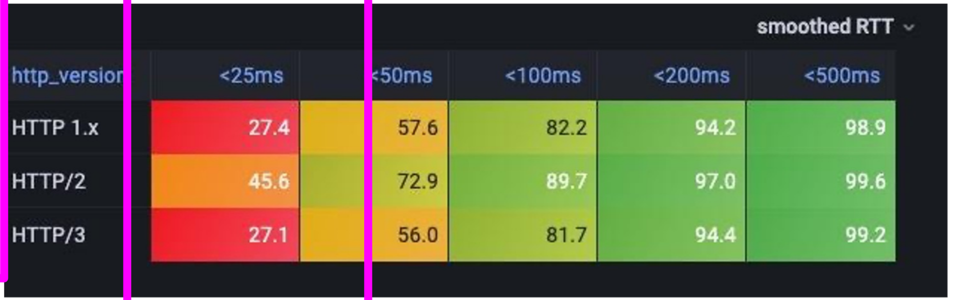
http_versior	<25ms	<50ms	<100ms	<200ms	<500ms
HTTP 1.x	44.6	72.7	89.5	96.6	99.5
HTTP/2	52.4	76.8	91.6	97.4	99.6
HTTP/3	43.8	69.5	89.0	97.3	99.6

BRAZIL



Throughput Summary

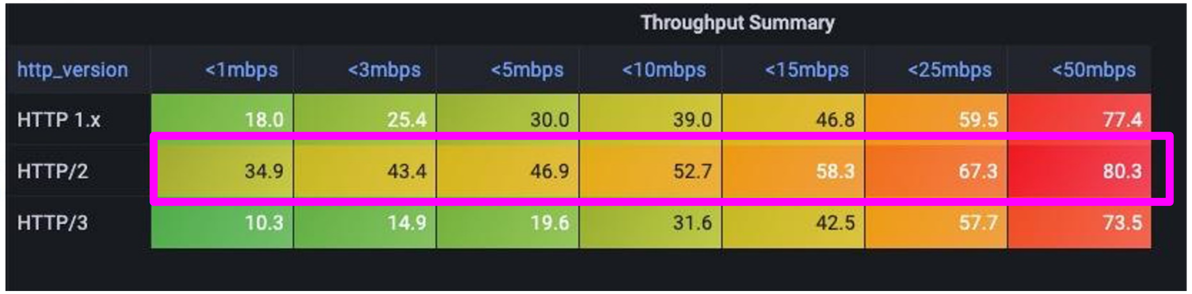
http_version	<1mbps	<3mbps	<5mbps	<10mbps	<15mbps	<25mbps	<50mbps
HTTP 1.x	14.9	22.6	28.5	40.8	50.5	64.3	83.3
HTTP/2	10.5	15.8	21.1	29.5	37.0	50.3	71.6
HTTP/3	12.7	19.7	26.4	40.3	50.8	66.7	82.4



smoothed RTT

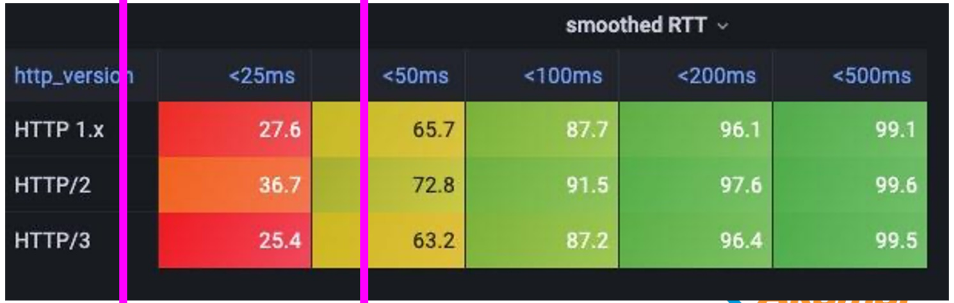
http_versior	<25ms	<50ms	<100ms	<200ms	<500ms
HTTP 1.x	27.4	57.6	82.2	94.2	98.9
HTTP/2	45.6	72.9	89.7	97.0	99.6
HTTP/3	27.1	56.0	81.7	94.4	99.2

USA



Throughput Summary

http_version	<1mbps	<3mbps	<5mbps	<10mbps	<15mbps	<25mbps	<50mbps
HTTP 1.x	18.0	25.4	30.0	39.0	46.8	59.5	77.4
HTTP/2	34.9	43.4	46.9	52.7	58.3	67.3	80.3
HTTP/3	10.3	14.9	19.6	31.6	42.5	57.7	73.5



smoothed RTT

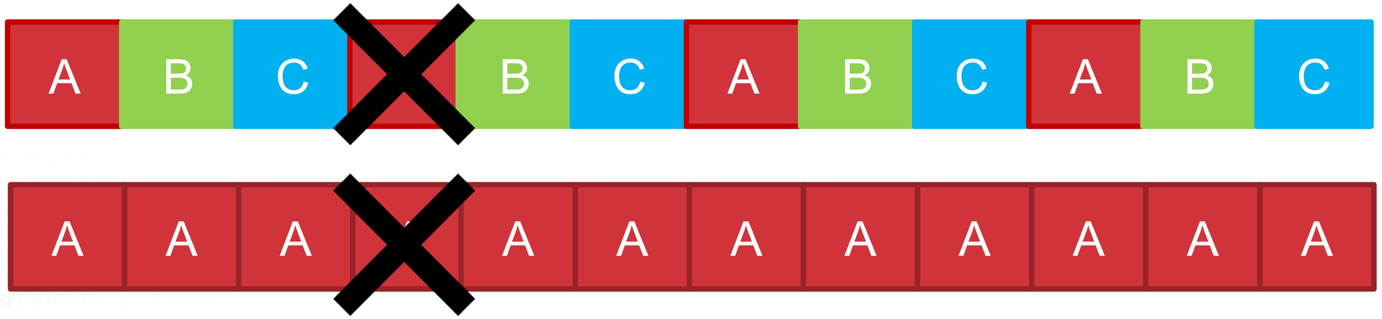
http_version	<25ms	<50ms	<100ms	<200ms	<500ms
HTTP 1.x	27.6	65.7	87.7	96.1	99.1
HTTP/2	36.7	72.8	91.5	97.6	99.6
HTTP/3	25.4	63.2	87.2	96.4	99.5

How to optimally benefit from QUIC?

Clearly, generic QUIC + HTTP/3 usage only provides marginal benefit over H1.1 and H2 when used with existing HAS players.

In many situations, they behave very similarly to TCP + HTTP/2

Single stream QUIC is still HEAD-OF-LINE blocked



Multi-stream QUIC allows flow on B and C

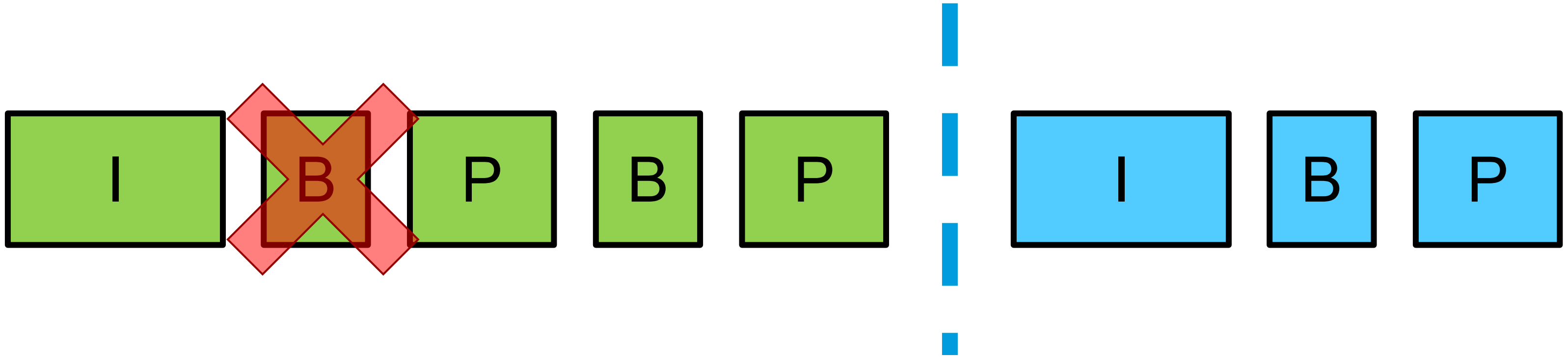
Single stream QUIC is still HEAD-OF-LINE blocked

We will get better performance from QUIC

- IF the connection has loss
- IF multiple streams are in progress at the same time.

Original slide credit: Robin Marx

Options for flexible loss recovery



What should the sender do? Three main options:

1. Retransmit B frame, then new frames
2. Send new frames first, then retransmit B
3. Send **only** new frames
4. Repair B using FEC data

What TCP does

What QUIC can do

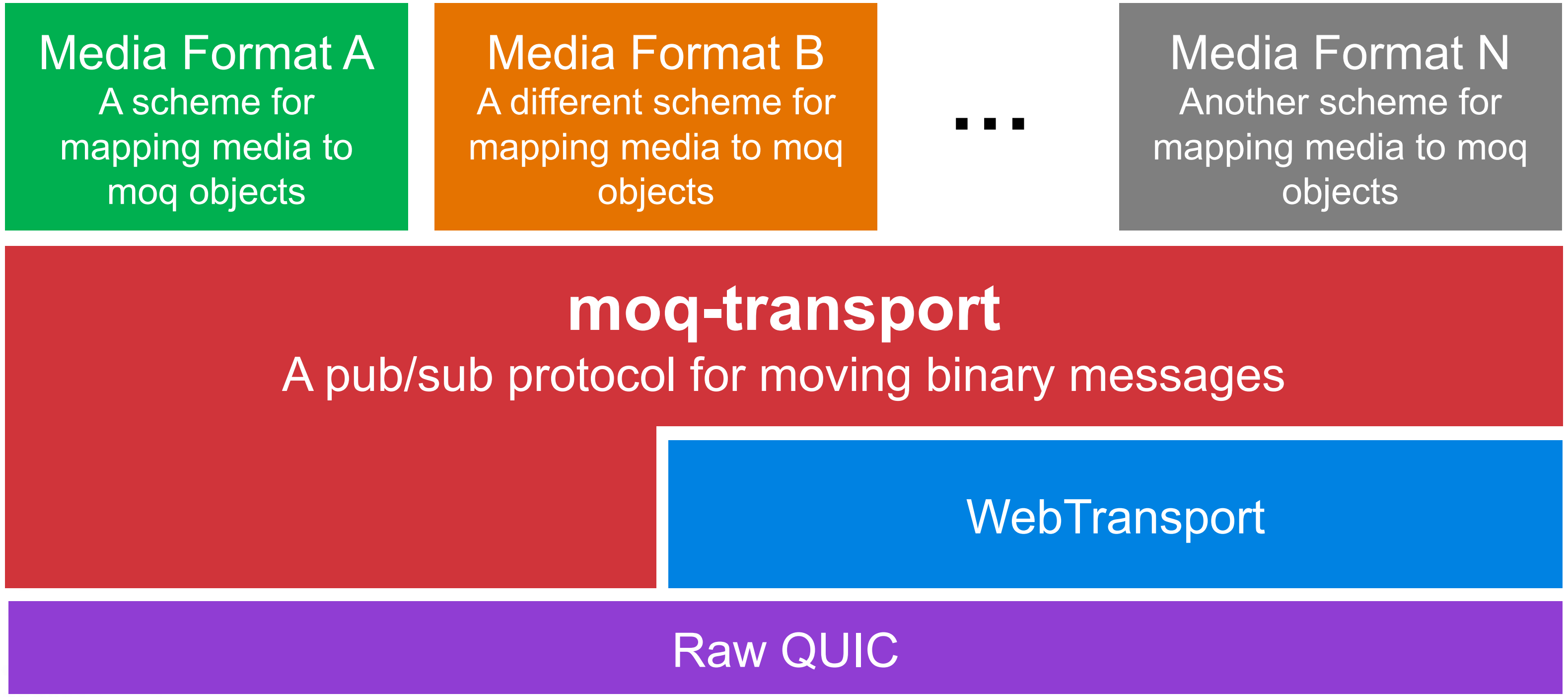
What application can do

Over to Ali

IETF MoQ – Media over QUIC

- Media over QUIC (MoQ) will develop a simple low-latency media delivery solution for ingest and distribution of media.
- Use cases including live streaming, gaming, and media conferencing and will scale efficiently.
- Implementable in both browser and non-browser endpoints.
- The common protocol for publishing media for ingest and distribution will support:
  - one or more media formats,
  - an interoperable way to request media and encodings, including audio, video, and timed metadata, such as captions and cue points.
  - rate adaptation strategies based on changing codec rates, changing chosen media encoding/qualities, or other mechanisms
  - cache friendly media mechanisms
- Can be used over raw QUIC or WebTransport.
- Chartered in Sept 2022 - https://datatracker.ietf.org/doc/charter-ietf-moq/01/

What is IETF MoQ?



Media Format A
A scheme for mapping media to moq objects

Media Format B
A different scheme for mapping media to moq objects

…

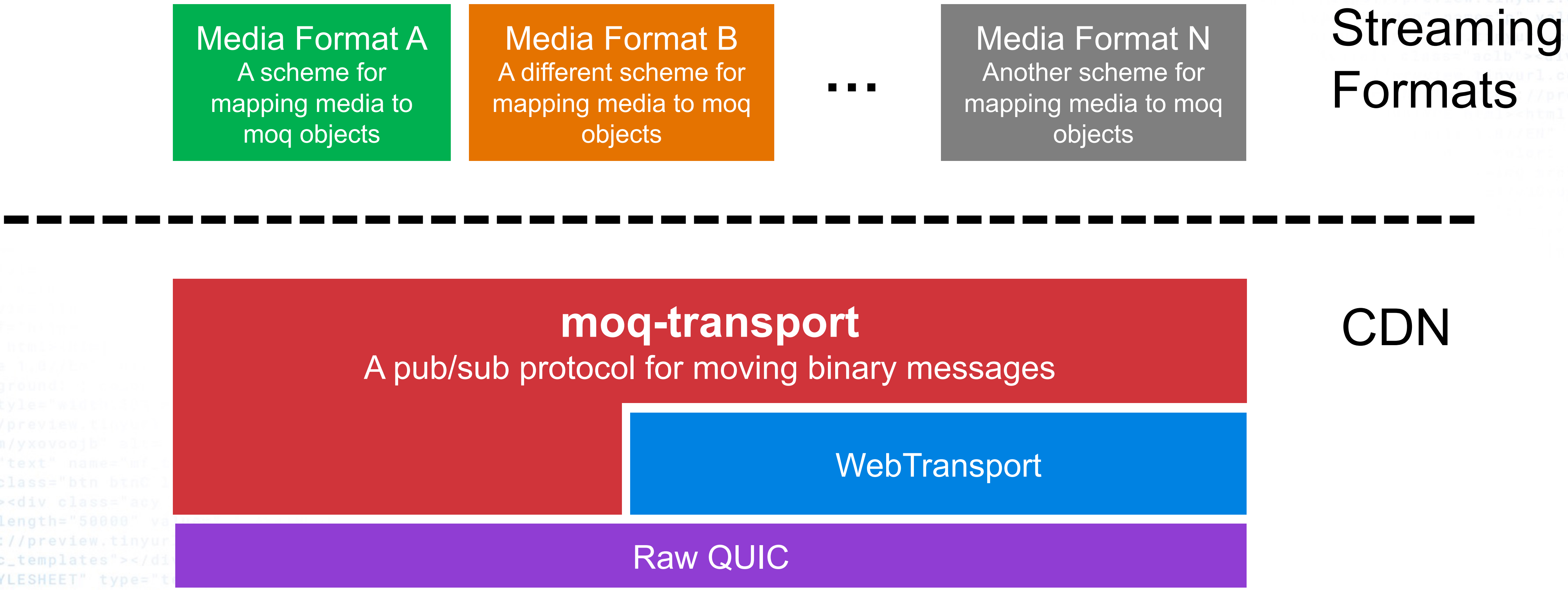
Media Format N
Another scheme for mapping media to moq objects

moq-transport
A pub/sub protocol for moving binary messages

WebTransport

Raw QUIC

What is IETF MoQ?



Media Format A

A scheme for mapping media to moq objects

Media Format B

A different scheme for mapping media to moq objects

…

Media Format N

Another scheme for mapping media to moq objects

Streaming Formats

moq-transport

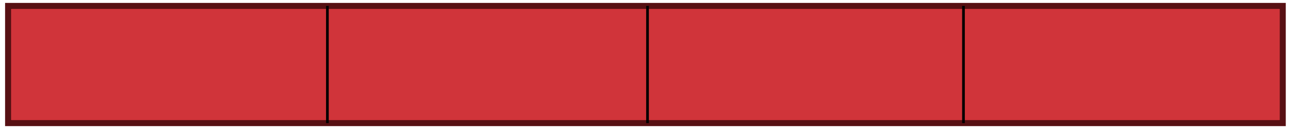
A pub/sub protocol for moving binary messages

WebTransport

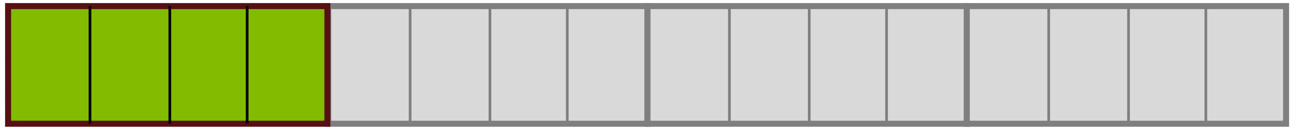
Raw QUIC

CDN

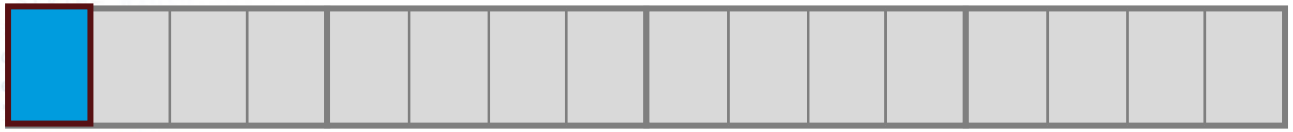
MoqTransport Object Model



Track – a temporal sequence of Groups. The entity against which a consumer issues a subscribe request.



Group – a sequence of Objects. Objects within a group SHOULD NOT depend on objects in other groups. A group behaves as a join point for subscriptions.

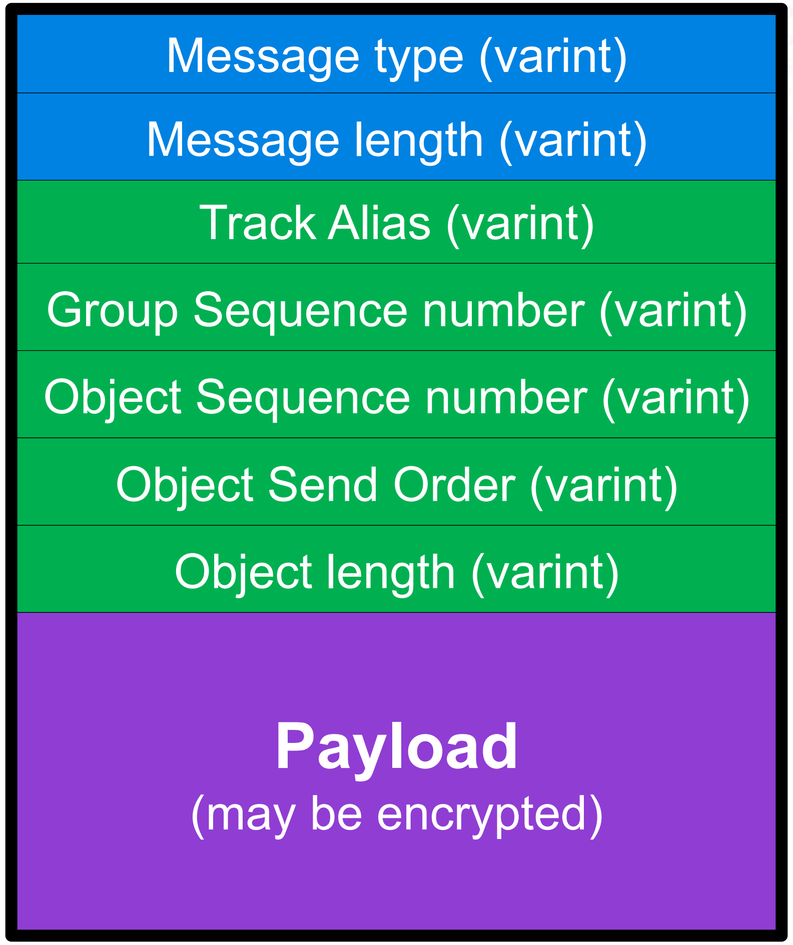


Object – an object is an addressable unit whose payload is a sequence of bytes. This is the atomic unit of transmission.

MoqTransport message types

- SUBSCRIBE
- SUBSCRIBE_OK
- SUBSCRIBE _ERROR
- ANNOUNCE
- ANNOUNCE_OK
- ANNOUNCE_ERROR
- UNANNOUNCE
- UNSUBSCRIBE
- SUBSCRIBE_FIN
- SUBSCRIBE_RST
- GOAWAY
- CLIENT_SETUP
- SERVER_SETUP
- OBJECT (with payload length)
- OBJECT (without payload length)

*all of these are subject to change ☺



Message type (varint)

Message length (varint)

Track Alias (varint)

Group Sequence number (varint)

Object Sequence number (varint)

Object Send Order (varint)

Object length (varint)

Payload

(may be encrypted)

Object message structure

WARP - a streaming format

Catalog draft

CATALOG

Defines versioning, catalog naming, track operations, track relationships, packaging declarations.

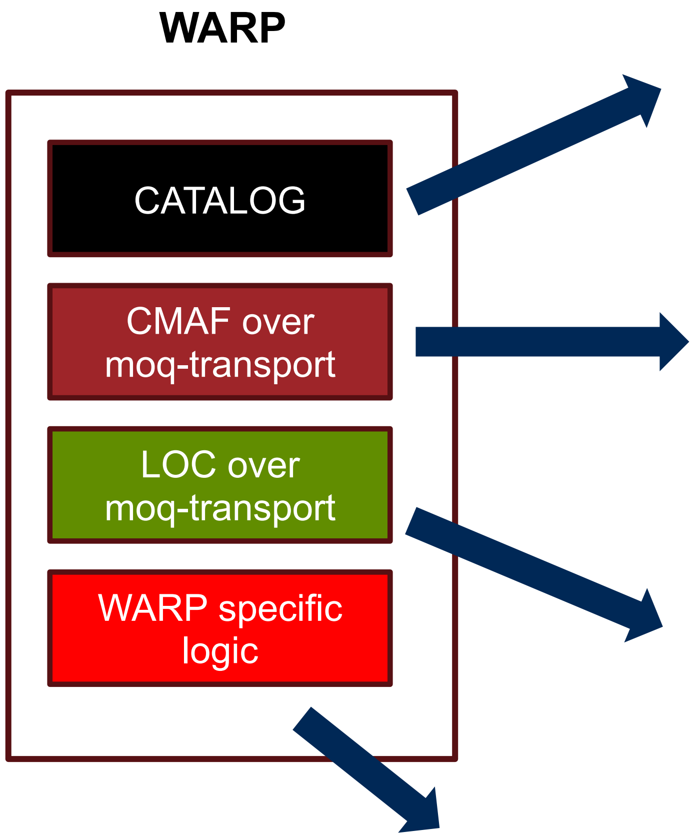
Packaging drafts

CMAF over moq-transport

Specifies how to package CMAF content for carriage over a moq-transport/catalog environment

LOC over moq-transport

Specifies how to package LOC content for carriage over a moq-transport/catalog environment



Done and available at https://datatracker.ietf.org/doc/draft-wilaw-moq-catalogformat/

Done and available at https://datatracker.ietf.org/doc/draft-wilaw-moq-cmafpackaging/

Need this TBD

2 PRs: https://github.com/moq-wg/warp-streaming-format/pulls
8 issues: https://github.com/moq-wg/warp-streaming-format/issues

CMAF Packaging for moq-transport

https://datatracker.ietf.org/doc/draft-wilaw-moq-cmafpackaging/

Defines an interoperable method of transmitting CMAF [CMAF] compliant media content over Media Over QUIC Transport (MOQT) [MoQTransport].

CMAF Track === MOQT Track

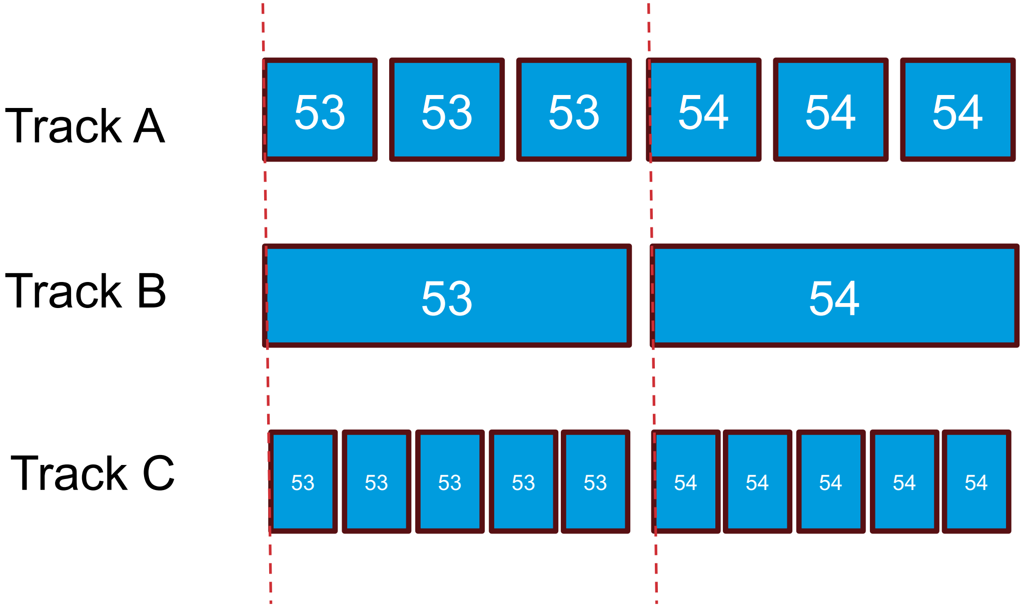
CMAF Switching Set === time-aligned MOQT Tracks

This draft maps CMAF objects to MOQT objects. The mapping of MOQT Objects to MOQT Streams is defined by the Streaming Format.

Time-aligned Packaging

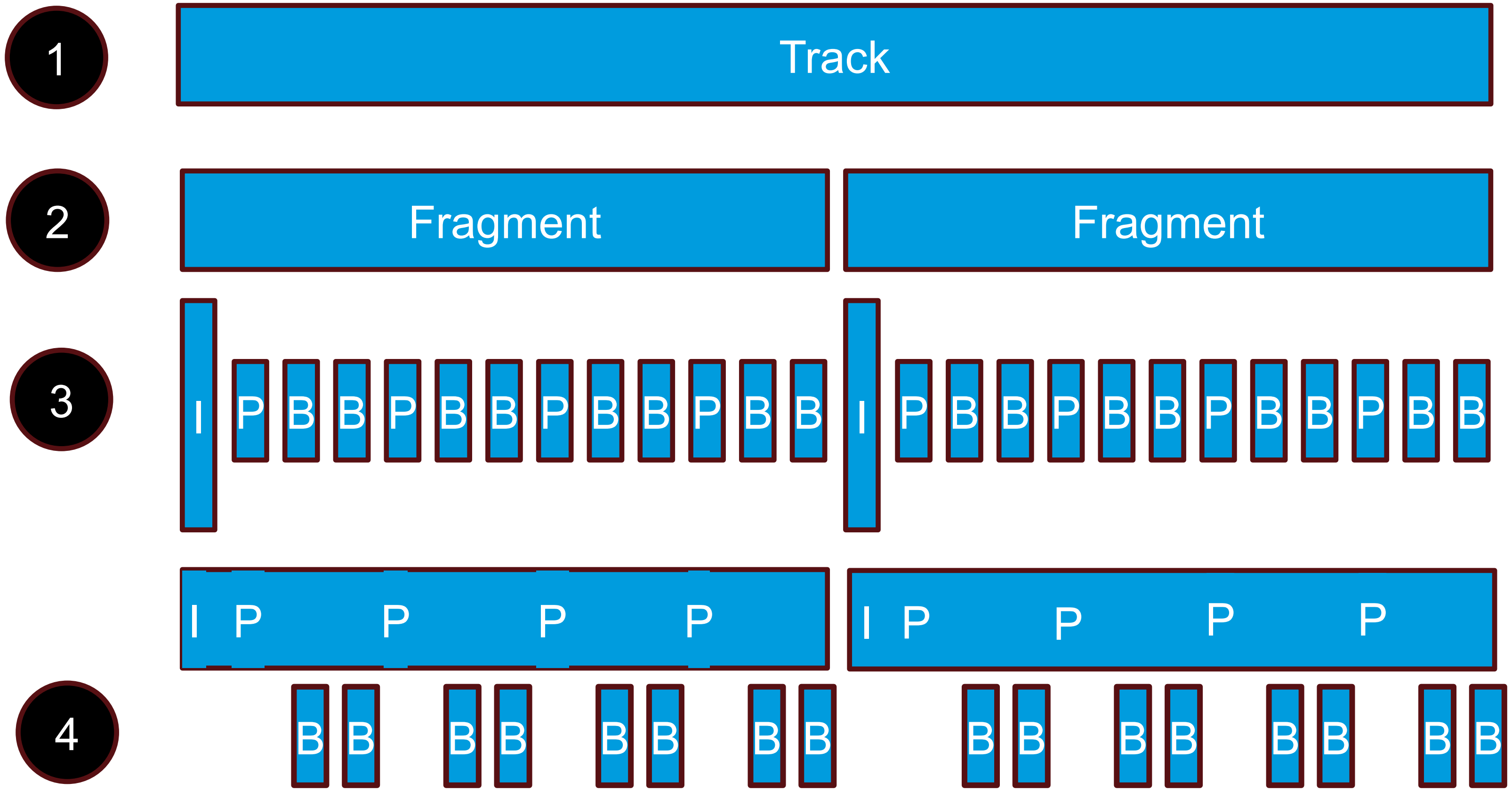
Equivalent Group Numbers across time-aligned tracks MUST hold media content with equivalent presentation time.

Group boundaries provide clean switch points.



Group and object durations do not need to match as long as the group boundaries align.

4 different modes of mapping CMAF objects to moq-transport streams



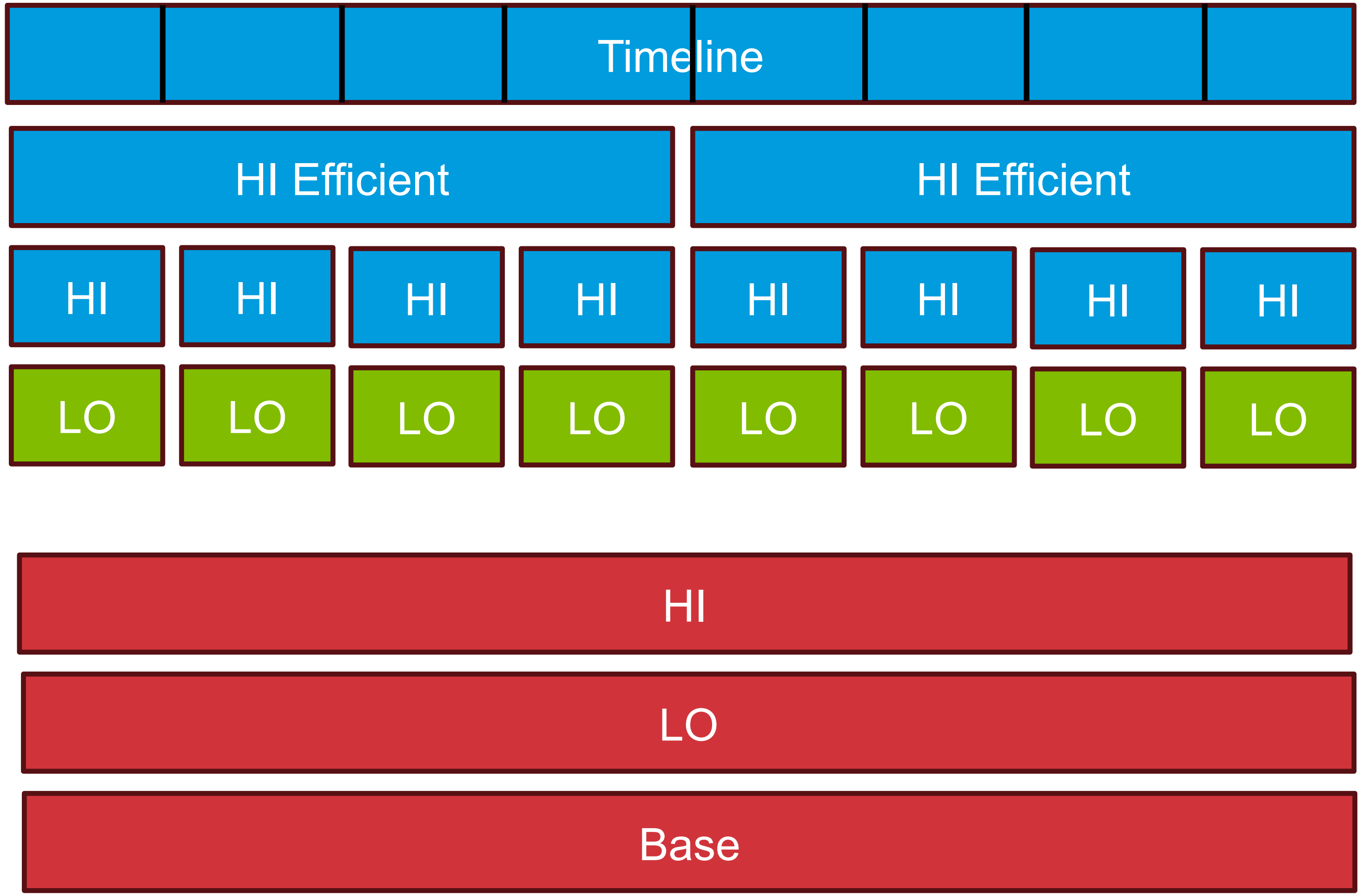
One track per stream

One fragment per stream

One frame per stream

Streams per frame type

ABR options



Can add a timeline track to inform receivers of group time/byte offsets.

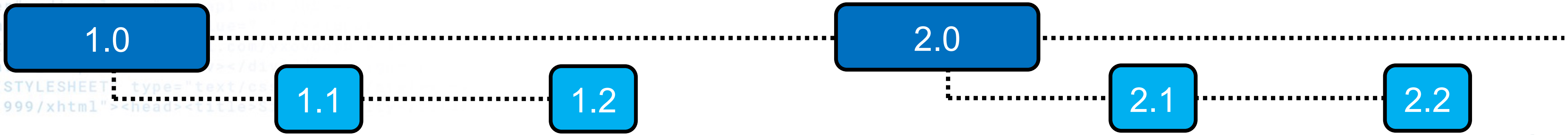
We can add tracks with longer GOPS for bandwidth efficiency

Traditional ABR – GOP boundaries match across tracks

SVC

What is a CATALOG ?

- A catalog is a **special track**.
- It has a **reserved name**
- Its purpose is to provide
  - the **names of all tracks** being produced by the publisher
  - **metadata** (bitrate, codec, resolution, frame rate etc) for each track to help with client selection.
  - **initialization data** for each track
  - **updates** about track additions and deletions.
- Catalogs can leverage **delta updates**, to enable lightweight propagation of track changes.



Example #1: Time-aligned Audio/Video Tracks with single quality

```
{
  "version": 1,  version of this catalog format
  "sequence": 0,  catalog update sequence number
  "streamingFormat": 1,  streaming format using this catalog
  "streamingFormatVersion": "0.2",  streaming format version
  "namespace": "conference.example.com/conference123/alice",  Track namespace (inherited)
  "packaging": "loc",  Track packaging format (inherited)
  "renderGroup": 1,  Track render group  - indicates tracks that are time-aligned and designed to be rendered
  "tracks": [        together (inherited)
    {   Track array - holds all tracks available from the publisher
      "name": "video",  Track name
      "selectionParams":{"codec":"av01.0.08M.10.0.110.09","width":1920,"height":1080,"framerate":30,"bitrate":1500000}
    },                                          Parameters describing the media characteristics of the track
    {
      "name": "audio",  Track name
      "selectionParams":{"codec":"opus","samplerate":48000,"channelConfig":"2","bitrate":32000}
    }                             Parameters describing the media characteristics of the track
  ]
}
```

Example #2: Simulcast video tracks - 3 alternate qualities along with audio

altgroup1 defines a group of alternative track. The player should subscribe to one from this group at a time

```
{
  "version": 1,
  "sequence": 0,
  "streamingFormat": 1,
  "streamingFormatVersion": "0.2",
  "namespace": "conference.example.com/conference123/alice",
  "renderGoup": 1,
  "codec": "av01",
  "tracks":[
    { "name": "hd", "selectionParams": {"width":1920,"height":1080,"bitrate":5000000,"framerate":30}, "altGroup":1 },
    { "name": "md", "selectionParams": {"width":720,"height":640,"bitrate":3000000,"framerate":30}, "altGroup":1 },
    { "name": "sd", "selectionParams": {"width":192,"height":144,"bitrate":500000,"framerate":30}, "altGroup":1 },
    { "name": "audio", "selectionParams":{"codec":"opus","samplerate":48000,"channelConfig":"2","bitrate":32000}
   ]
}
```

The audio track overwrites the inherited av01 codec

Example #3: Patch update adding a track

```
[
   { "op": "add", "path": "/tracks/-", "value": {
    "name": "slides",
    "selectionParams": {
            "codec":"av01.0.08M.10.0.110.09",
            "width":1920,
            "height":1080,
            "framerate":15,
            "Bitrate":750000
            },
     "renderGroup":1
     }
   }
 ]
```

Note that namespace and packaging were all declared in the parent.

Example #4: Patch update removing 3 tracks

```
[
   { "op": "remove", "path": "/tracks/2"},
   { "op": "remove", "path": "/tracks/1"},
   { "op": "remove", "path": "/tracks/0"},
]
```

Example #5: A catalog referencing catalogs for two different formats

```
{
  "version": 1,
  "sequence": 0,
  "catalogs": [    catalogs and tracks arrays are mutually
    {              exclusive
      "name": "catalog-for-format-one",
      "namespace": "sports.example.com/games/08-08-23/live",
      "streamingFormat":1,      identifies the streaming format
      "streamingFormatVersion": "0.2"   identifies this format’s version
    },
    {
      "name": "catalog-for-format-five",
      "namespace": "chat.example.com/games/08-08-23/chat",
      "streamingFormat":5,     identifies the streaming format
      "streamingFormatVersion": "1.6.2"   identifies this format’s version
    }
  ]
}
```

Key issues being debated right now

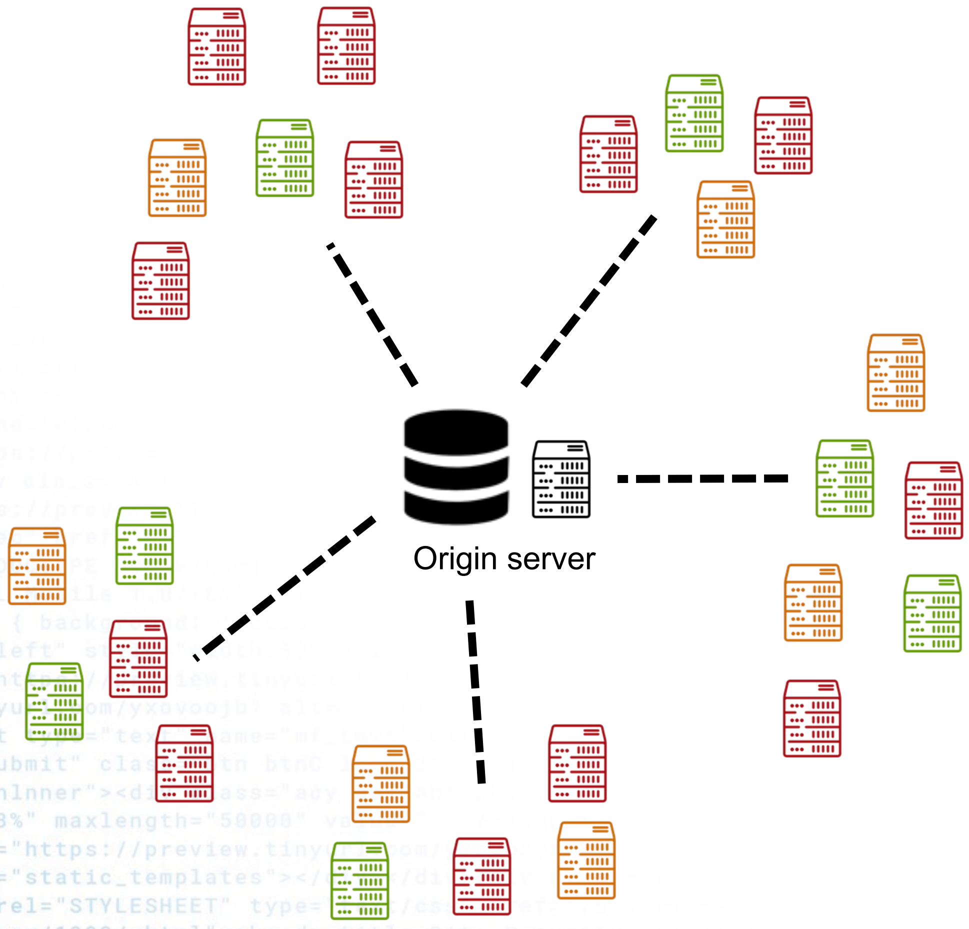
- How **PUBLISHING** should work
  - Publish only after subscription
  - ANNOUCE origin locations?
- **Priority schemes** and Congestion response
- **Relay** interactions
  - How to implement relative prioritization at relays across different vendors?
- How will **variable quality** (rate adaptation) be achieved?
  - SS-ABR, CS-ABR, SVC,dynamic encoding
- **Advertising insertion** (MOQT dependency)
- **Content protection** – define and add Schema and pssh data to catalog as track properties. (catalog dependency)
- And **many more**!!

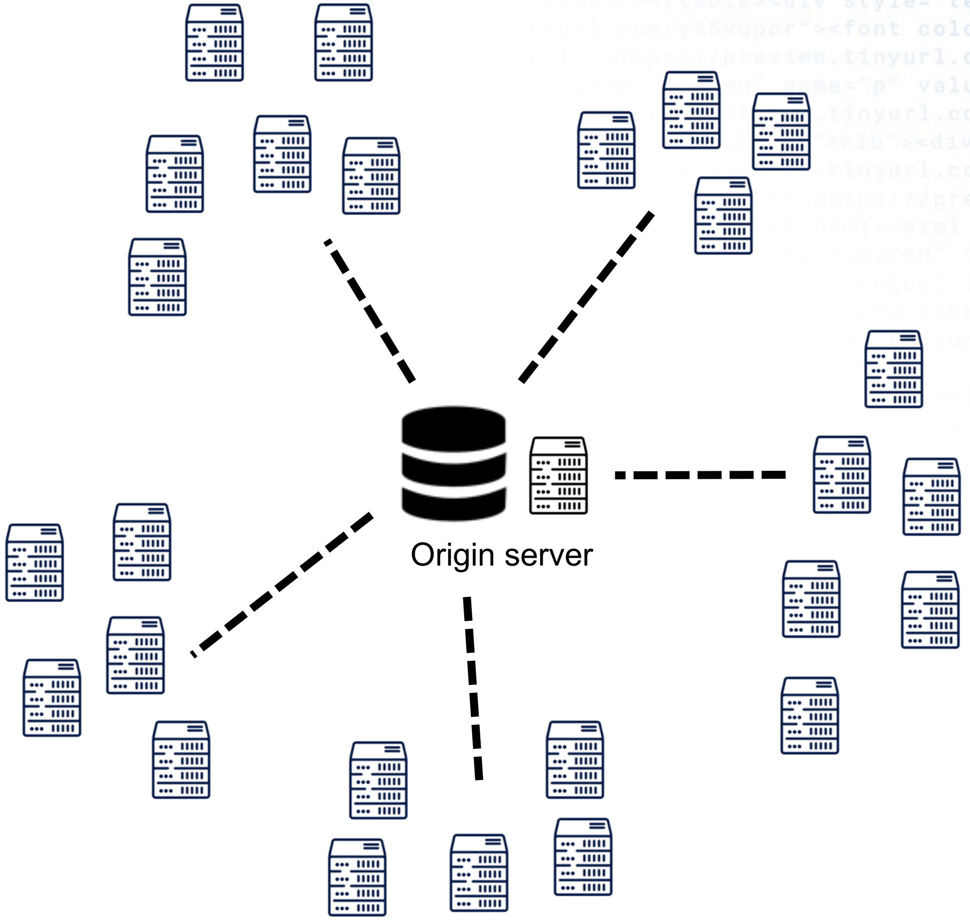
Comparison of low-latency formats – Dec 2023

	MoQ	WebRTC	LL-HLS	LL-DASH	HESP
Supports real-time latency (200-400ms)	🟩	🟩	🟥	🟥	🟥
Supports interactive latency (800-4000ms)	🟩	🟥	🟥	🟥	🟩
Supports interactive latency (3000-4000ms)	🟩	🟥	🟩	🟩	🟩
Supports stable latency (8000-20000ms)	🟩	🟥	🟩	🟩	🟩
Can be cached	🟩	🟥	🟩	🟩	🟩
Broad player support	🟥	🟩	🟩	🟩	🟥
Broad advertising support	🟥	🟥	🟥	🟥	🟥
DRM support	🟥	▨	🟩	🟩	🟩
Supports playback from browser-based clients	🟩	🟩	🟩	🟩	🟩
Operates in networks without QUIC support	🟨	🟩	🟩	🟩	🟩

The economics of CDN distribution

A homogeneous network provides greater capacity and lower COGS.





VOD

LL-HLS/DASH

WebRTC

MoQ

Headwinds for MoQ

- Some networks block QUIC traffic today
- QUIC is far more (>100%) CPU intensive to deliver than TCP
- Congestion response still unproven
- ABR still unproven
- WebRTC and HLS/DASH work sufficiently well for many use-cases.
- Resistance to change media workflows
- Lack of advertising support

MoQ timelines

- **IETF #117** July 22-28, San Francisco
- **Virtual Interim Meeting** - Boston – October 3-5
- **IETF #118** Nov 4-10, Prague.
- Interim Meeting - Denver(?) – Feb 8-9
- IETF #119 Brisbane – March 16-22
- IETF #120 July 20-26

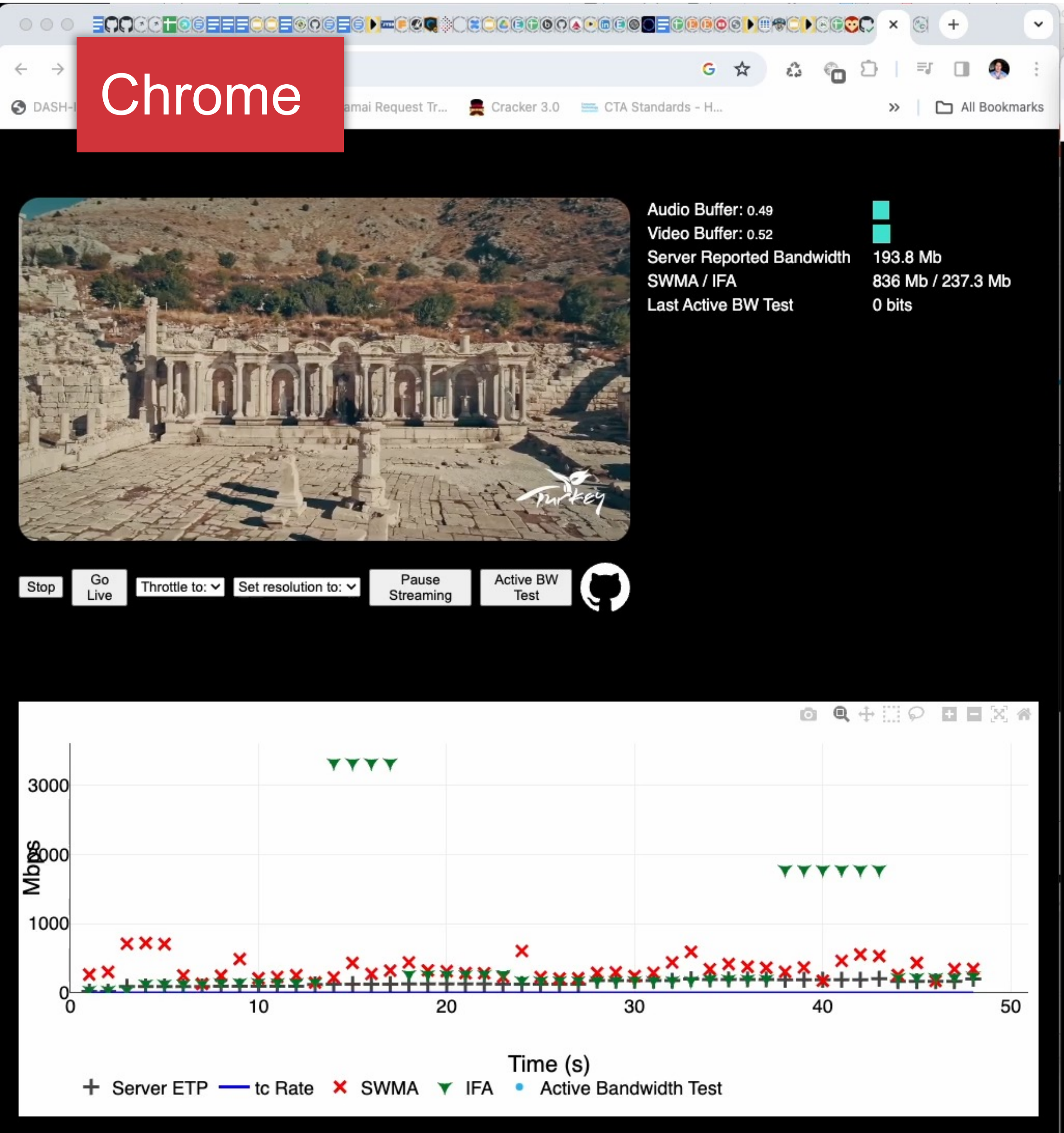
When will MoQ specification be "ready"? Late 2024?

Can you get involved? Absolutely. See

- WorkGroup: https://datatracker.ietf.org/group/moq/about/
- Mailing list: https://www.ietf.org/mailman/listinfo/moq

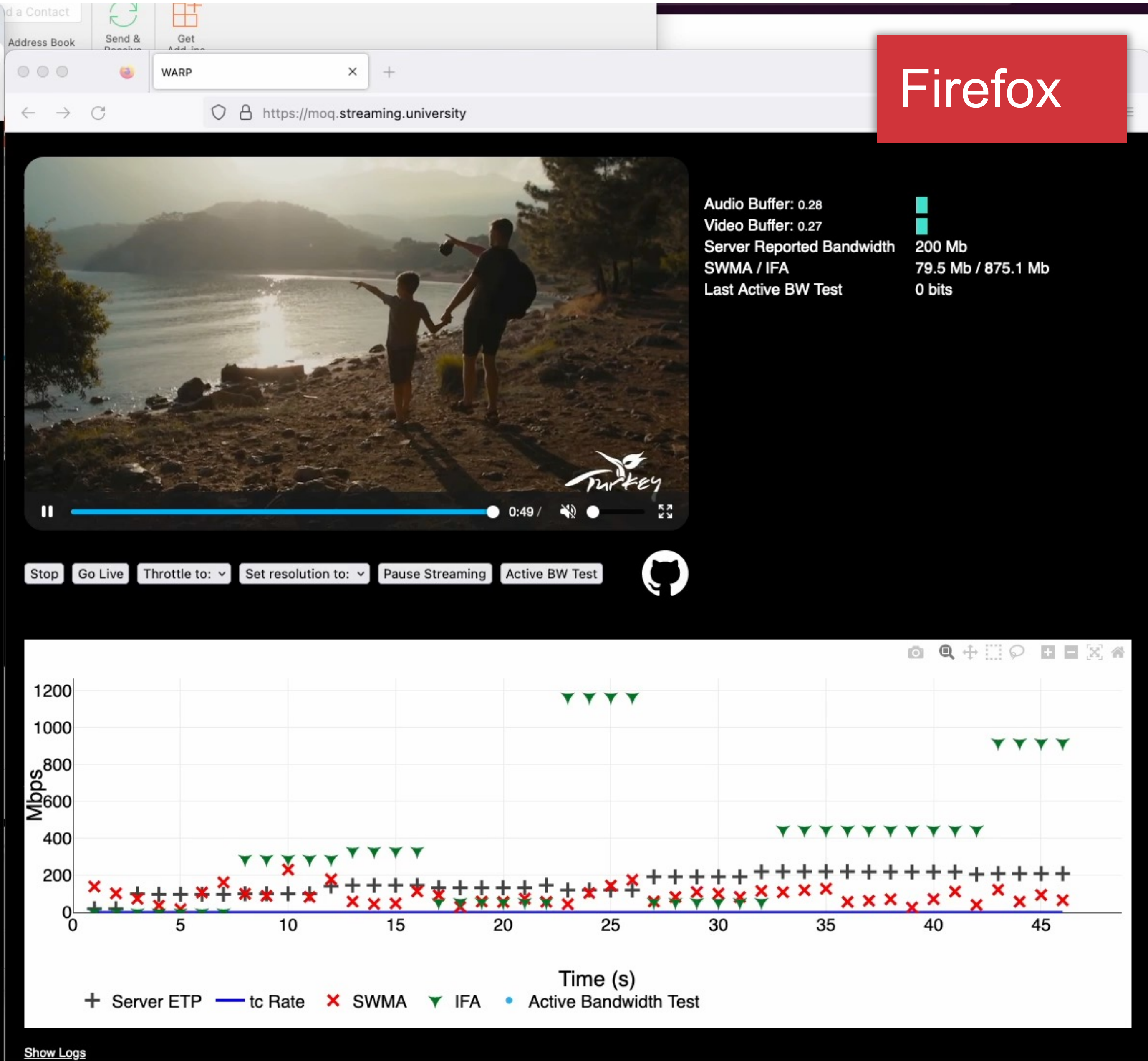
moq.streaming.university

Chrome



Audio Buffer: 0.49	
Video Buffer: 0.52	
Server Reported Bandwidth	193.8 Mb
SWMA / IFA	836 Mb / 237.3 Mb
Last Active BW Test	0 bits

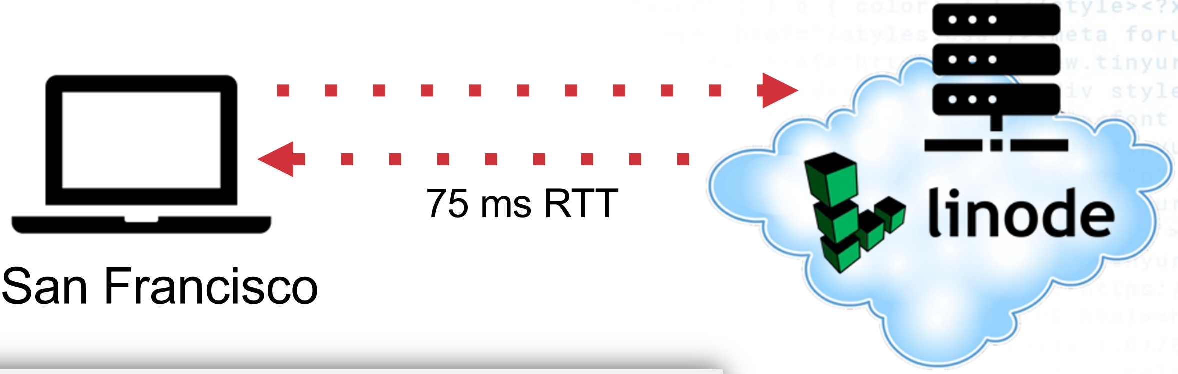
Firefox



Audio Buffer: 0.28	
Video Buffer: 0.27	
Server Reported Bandwidth	200 Mb
SWMA / IFA	79.5 Mb / 875.1 Mb
Last Active BW Test	0 bits

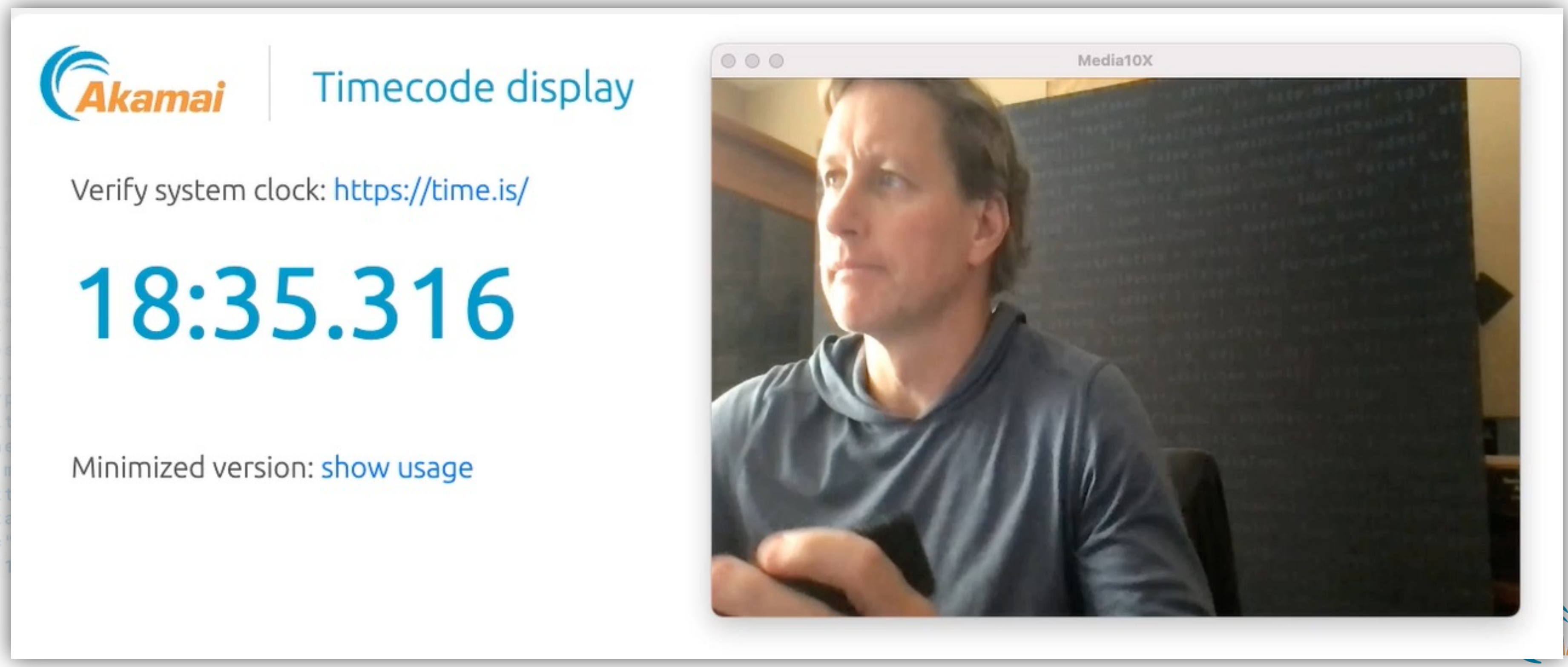
QUICR Demo – San Francisco to Akamai Linode in Atlanta and back again.

A very alpha version of the cisco QUICR protocol (using datagrams over QUIC)



San Francisco

Atlanta



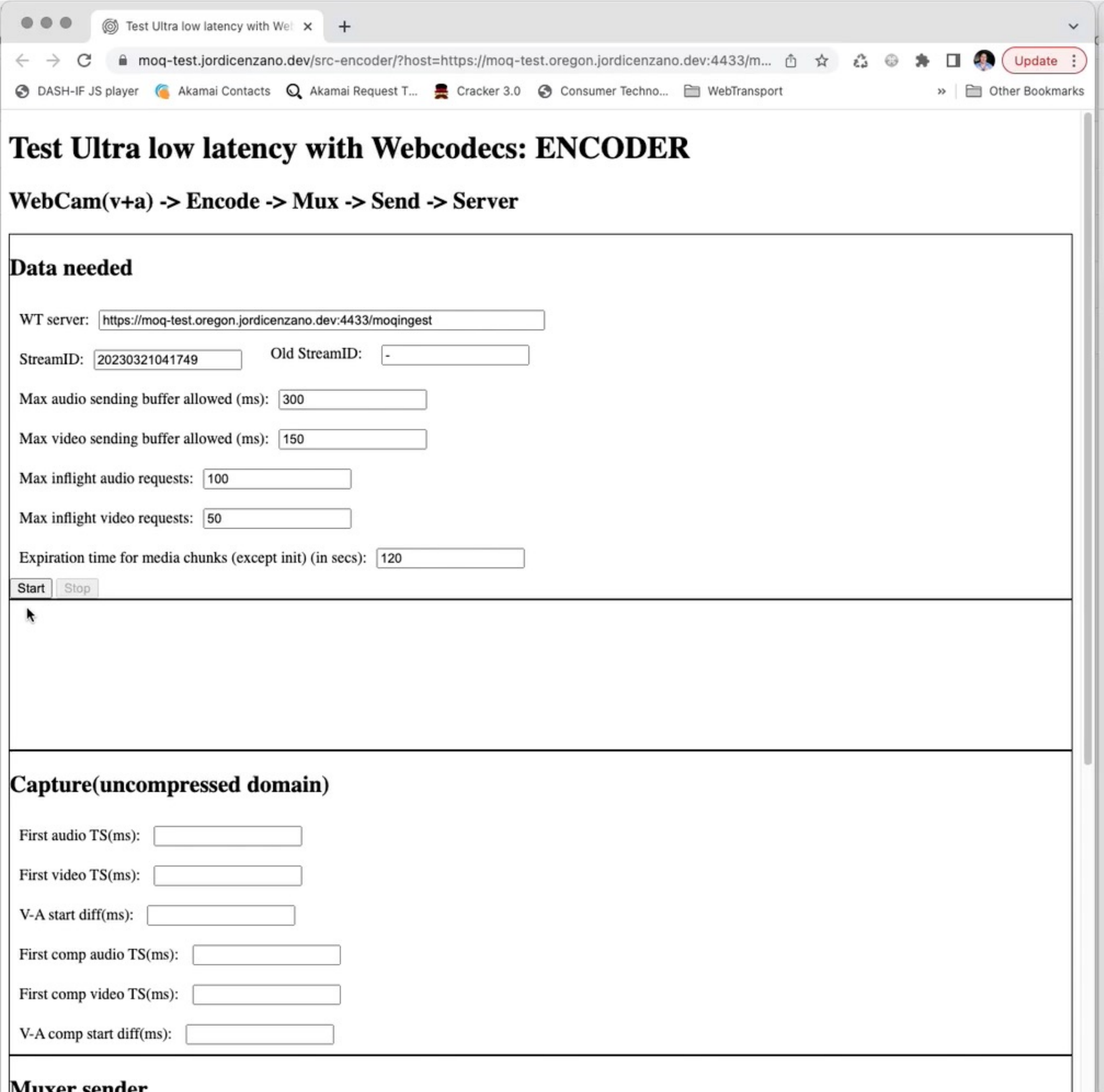
Timecode display

Verify system clock: https://time.is/

18:35.316

Minimized version: show usage

Demo - META implementation of MoQ (by Jordi Cenzano)



Test Ultra low latency with We
moq-test.jordicenzano.dev/src-encoder/?host=https://moq-test.oregon.jordicenzano.dev:4433/m...
Update
DASH-IF JS player
Akamai Contacts
Akamai Request T...
Cracker 3.0
Consumer Techno...
WebTransport
Other Bookmarks

Test Ultra low latency with Webcodecs: ENCODER

WebCam(v+a) -> Encode -> Mux -> Send -> Server

Data needed

WT server: https://moq-test.oregon.jordicenzano.dev:4433/moqingest

StreamID: 20230321041749 Old StreamID: -

Max audio sending buffer allowed (ms): 300

Max video sending buffer allowed (ms): 150

Max inflight audio requests: 100

Max inflight video requests: 50

Expiration time for media chunks (except init) (in secs): 120

Start Stop

Capture(uncompressed domain)

First audio TS(ms):

First video TS(ms):

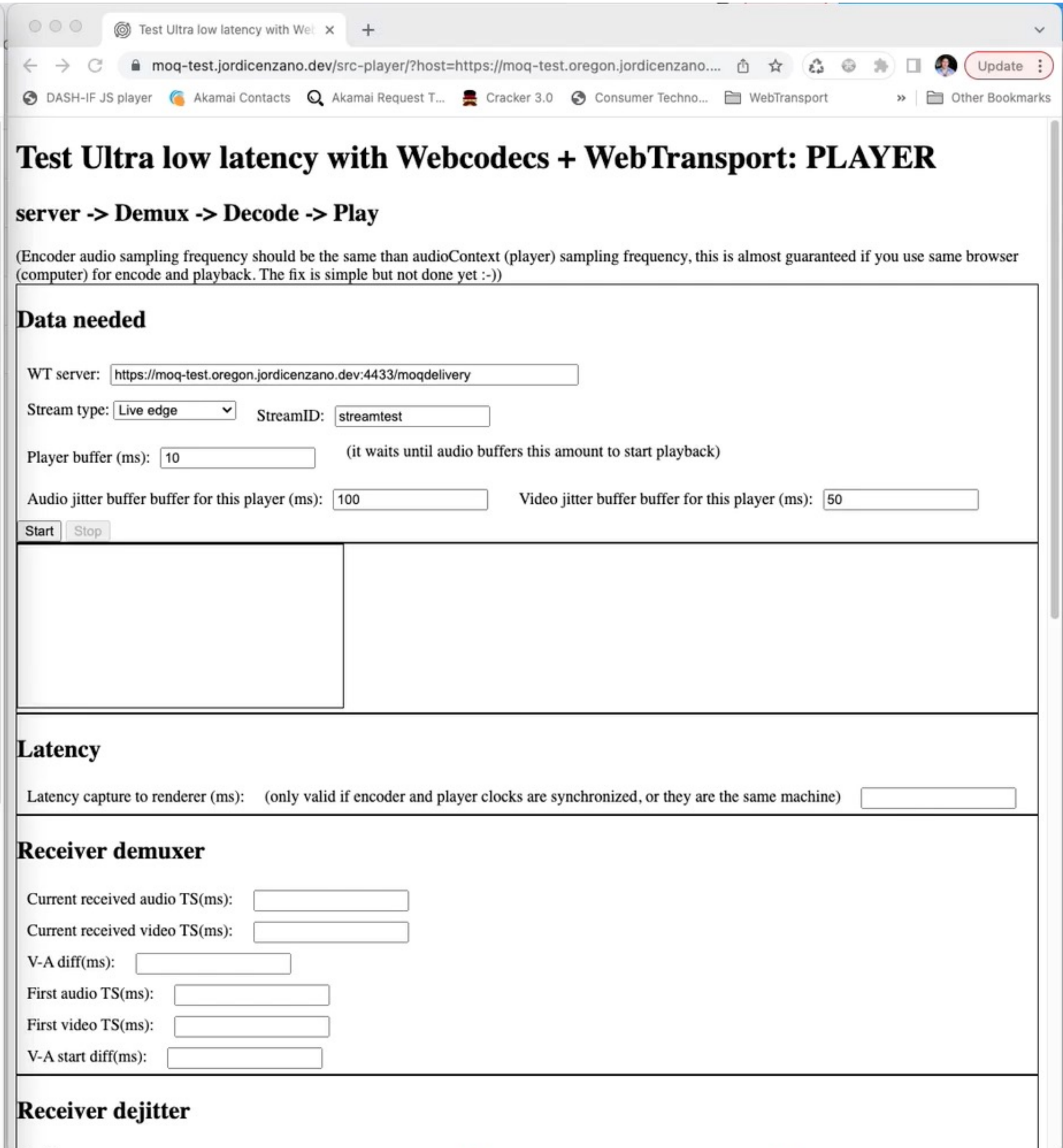
V-A start diff(ms):

First comp audio TS(ms):

First comp video TS(ms):

V-A comp start diff(ms):

Muxer sender



Test Ultra low latency with We
moq-test.jordicenzano.dev/src-player/?host=https://moq-test.oregon.jordicenzano....
Update
DASH-IF JS player
Akamai Contacts
Akamai Request T...
Cracker 3.0
Consumer Techno...
WebTransport
Other Bookmarks

Test Ultra low latency with Webcodecs + WebTransport: PLAYER

server -> Demux -> Decode -> Play

(Encoder audio sampling frequency should be the same than audioContext (player) sampling frequency, this is almost guaranteed if you use same browser (computer) for encode and playback. The fix is simple but not done yet :-))

Data needed

WT server: https://moq-test.oregon.jordicenzano.dev:4433/moqdelivery

Stream type: Live edge StreamID: streamtest

Player buffer (ms): 10 (it waits until audio buffers this amount to start playback)

Audio jitter buffer buffer for this player (ms): 100 Video jitter buffer buffer for this player (ms): 50

Start Stop

Latency

Latency capture to renderer (ms): (only valid if encoder and player clocks are synchronized, or they are the same machine)

Receiver demuxer

Current received audio TS(ms):

Current received video TS(ms):

V-A diff(ms):

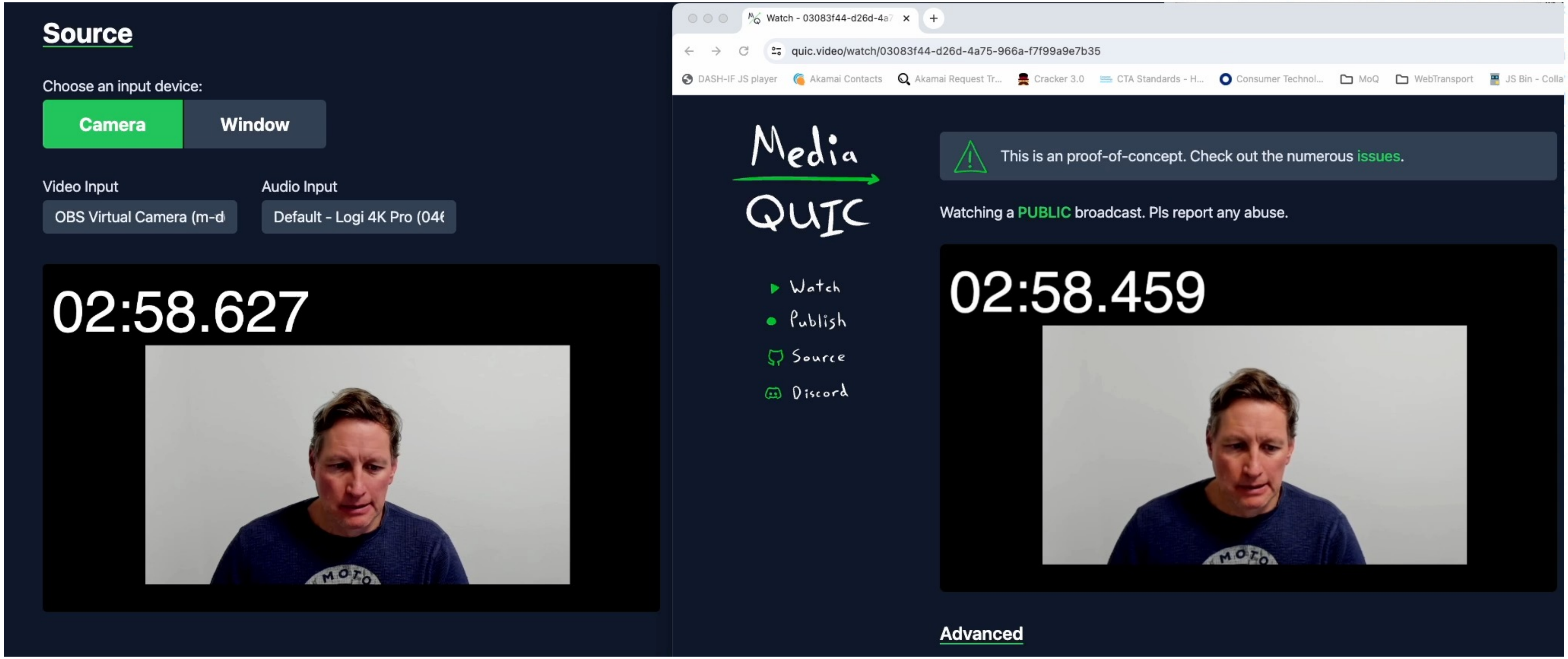
First audio TS(ms):

First video TS(ms):

V-A start diff(ms):

Receiver dejitter

Quic.video



Source

Choose an input device:

Video Input: OBS Virtual Camera (m-d

Audio Input: Default - Logi 4K Pro (04€

02:58.627

Watch - 03083f44-d26d-4a7

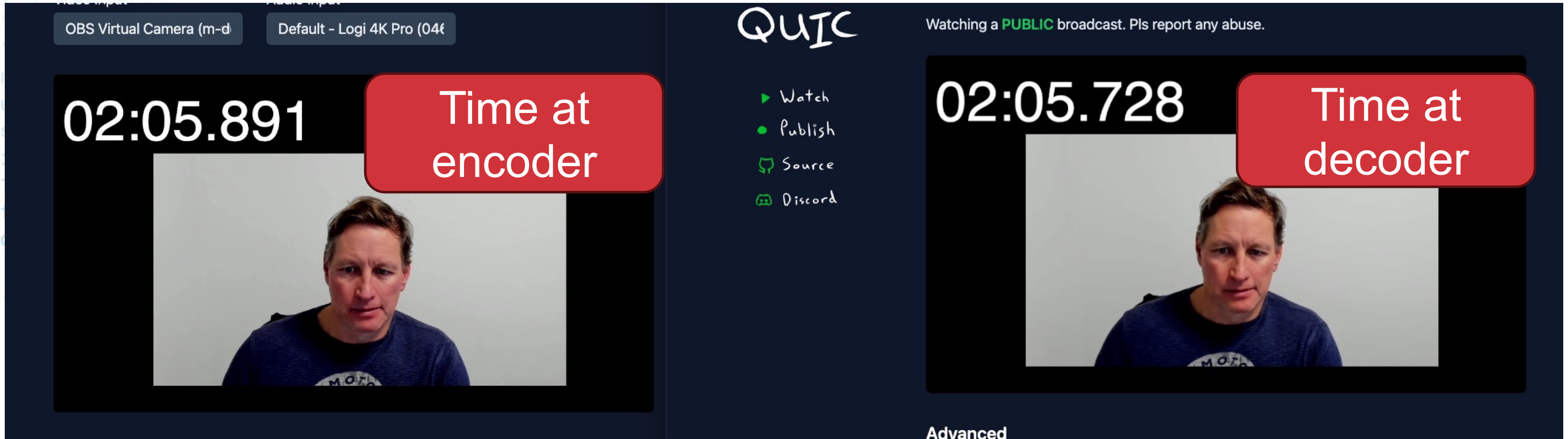
quic.video/watch/03083f44-d26d-4a75-966a-f7f99a9e7b35

This is an proof-of-concept. Check out the numerous issues.

Watching a PUBLIC broadcast. Pls report any abuse.

02:58.459

Advanced



Video Input: OBS Virtual Camera (m-d

Audio Input: Default - Logi 4K Pro (04€

02:05.891

Time at encoder

Watching a PUBLIC broadcast. Pls report any abuse.

02:05.728

Time at decoder

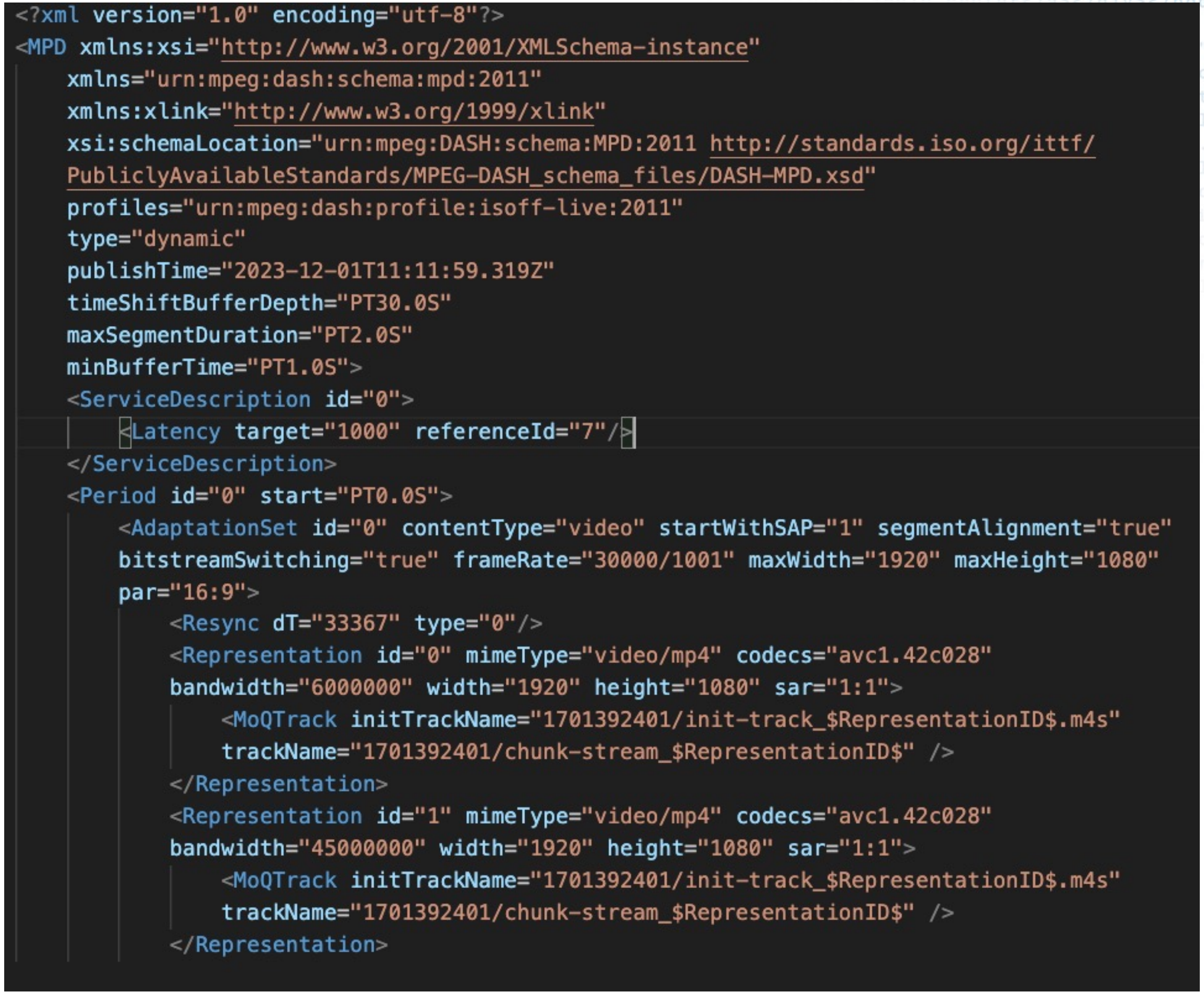
Advanced

163ms video latency
Zurich – Des Moines – Zurich
(120ms RTT)

Akamai Experience the Edge

Recommendations for DASH IF

1. Update DASH Manifest (.mpd) so that it can be used over MOQT. (DASM)
2. Extend dash.js to support this new DASH playback over MOQT.



```
<?xml version="1.0" encoding="utf-8"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="urn:mpeg:dash:schema:mpd:2011"
    xmlns:xlink="http://www.w3.org/1999/xlink"
    xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011 http://standards.iso.org/ittf/
    PubliclyAvailableStandards/MPEG-DASH_schema_files/DASH-MPD.xsd"
    profiles="urn:mpeg:dash:profile:isoff-live:2011"
    type="dynamic"
    publishTime="2023-12-01T11:11:59.319Z"
    timeShiftBufferDepth="PT30.0S"
    maxSegmentDuration="PT2.0S"
    minBufferTime="PT1.0S">
    <ServiceDescription id="0">
        <Latency target="1000" referenceId="7"/>
    </ServiceDescription>
    <Period id="0" start="PT0.0S">
        <AdaptationSet id="0" contentType="video" startWithSAP="1" segmentAlignment="true"
        bitstreamSwitching="true" frameRate="30000/1001" maxWidth="1920" maxHeight="1080"
        par="16:9">
            <Resync dT="33367" type="0"/>
            <Representation id="0" mimeType="video/mp4" codecs="avc1.42c028"
            bandwidth="6000000" width="1920" height="1080" sar="1:1">
                <MoQTrack initTrackName="1701392401/init-track_$RepresentationID$.m4s"
                trackName="1701392401/chunk-stream_$RepresentationID$" />
            </Representation>
            <Representation id="1" mimeType="video/mp4" codecs="avc1.42c028"
            bandwidth="45000000" width="1920" height="1080" sar="1:1">
                <MoQTrack initTrackName="1701392401/init-track_$RepresentationID$.m4s"
                trackName="1701392401/chunk-stream_$RepresentationID$" />
            </Representation>
```

Thank you for your time.

Questions?