

# Catalog Specification for MoQ compliant Streaming Formats

draft-law-moq-catalog-00

File issues at <https://github.com/suhasHere/moq-catalog/issues>

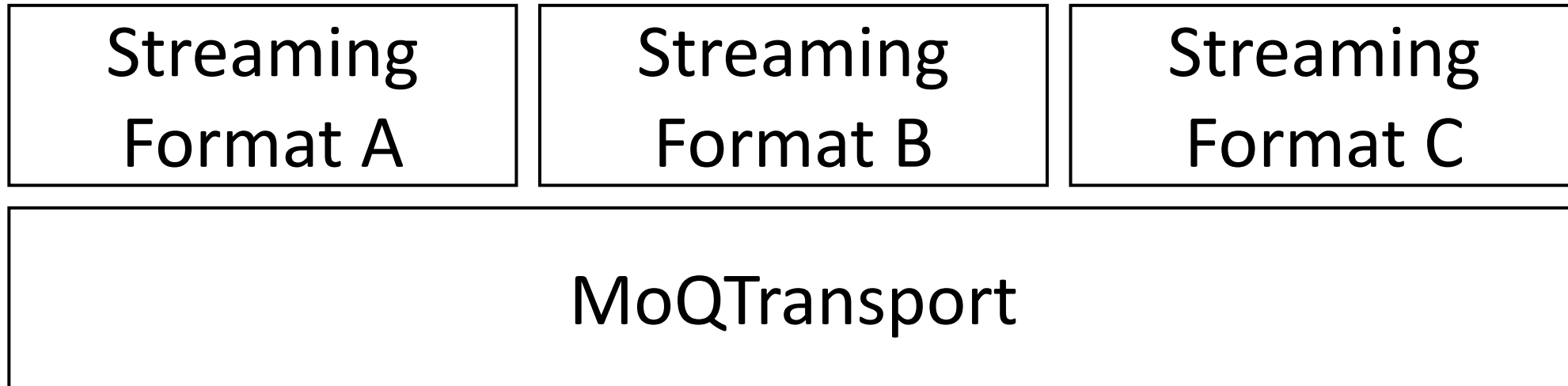
Will Law

Akamai

March 2023

# Assumption #1

There will be a common **MoQTransport Specification** which is then utilized by multiple **Streaming Format Specifications**



# Assumption #1

To satisfy our charter we need to define the MoQTransport and at least one Streaming Format

Streaming  
Format A

Streaming  
Format B

Streaming  
Format C

MoQTransport

# Separation of responsibilities

## MoQTransport (1)

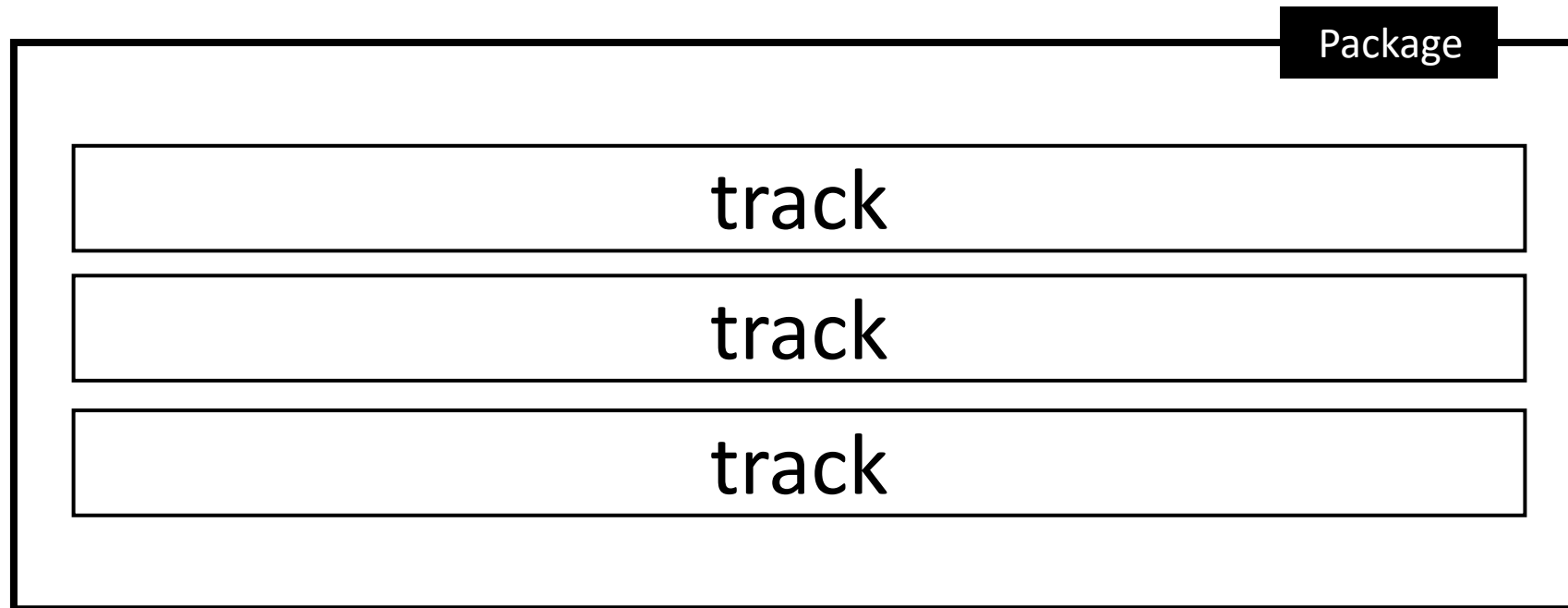
- Object model & structure (bundles, tracks, groups, objects)
- Addressing schemas
- Catalog requesting and type identification
- Rules for relays (priority, sendOrder, dropping)
- Carriage of MoQ over transport protocols (WebTransport and raw QUIC)

## Streaming Format (N)

- Catalog format and serialization
- Media container (likely reference some other container spec such as CMAF)
- How media is mapped to the MoQTransport object model (GOP per object, frame per object, datagrams etc)
- Workflow logic including
  - bitrate adaptation
  - content protection
  - access control
  - error handling etc.
  - synchronization between tracks
- Streaming format type registration

# Assumptions #2

- The subscribe-able entities within MoQTransport are **tracks**.
- There is an addressable **package of tracks** (name is TBD – could be bundle | emission | broadcast)
- One of the tracks is the "**catalog**". It describes the other tracks and gives the client the data it needs to select, initialize and subscribe to these tracks.
- Tracks have an **identifier**



# Problems

1. Given the address to a package of tracks, how does a client know how to retrieve the catalog?
2. Given that different streaming formats may use different binary serializations, how can a client know how to parse a catalog it has not seen before?

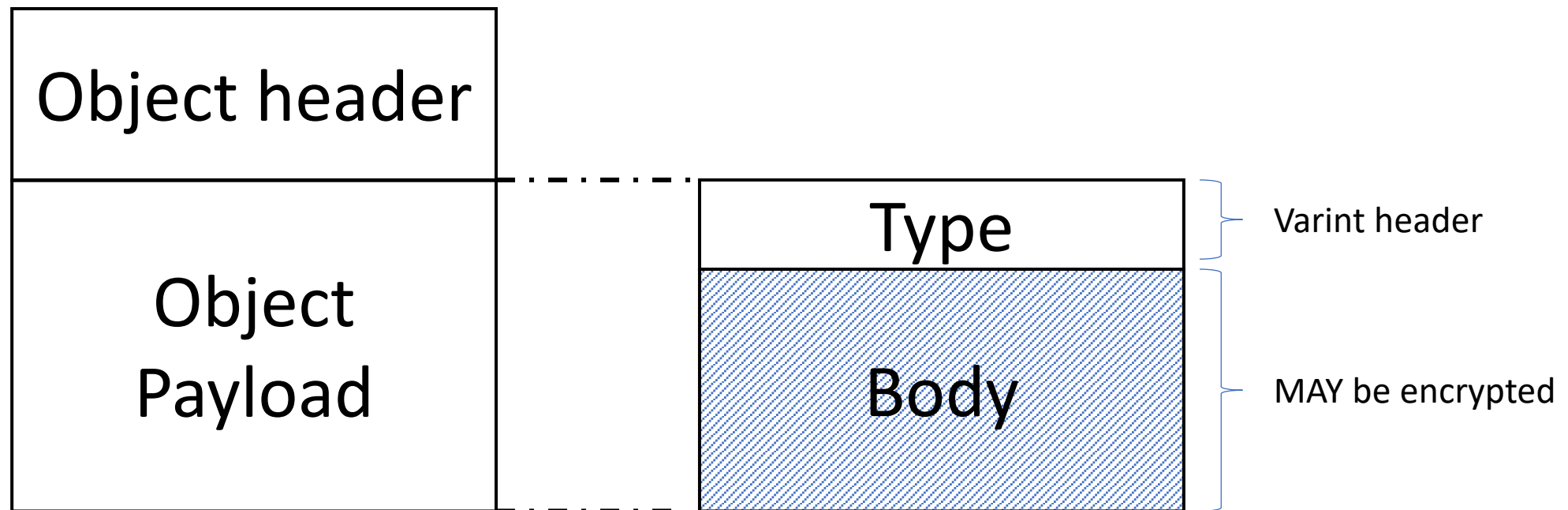
[draft-law-moq-catalog-00](#) proposes solutions to these two problems.

Problem#1: Given the address to a package of tracks, how does a client know how to retrieve the catalog?

1. The catalog SHALL have a reserved track ID of 0 if we have numeric identifiers or "catalog" if string.
2. There can only be one track with an ID of 0 | "catalog" in a package.
3. If applications demand a multi-level catalog description, there must still be a parent entry-point at 0 | "catalog" that references the tracks .

Problem#2: Given that different streaming formats may use different binary serializations, how can a client know how to parse a catalog it has not seen before?

A Catalog is a MoQ object, meaning it has a header and a payload.





# Registration

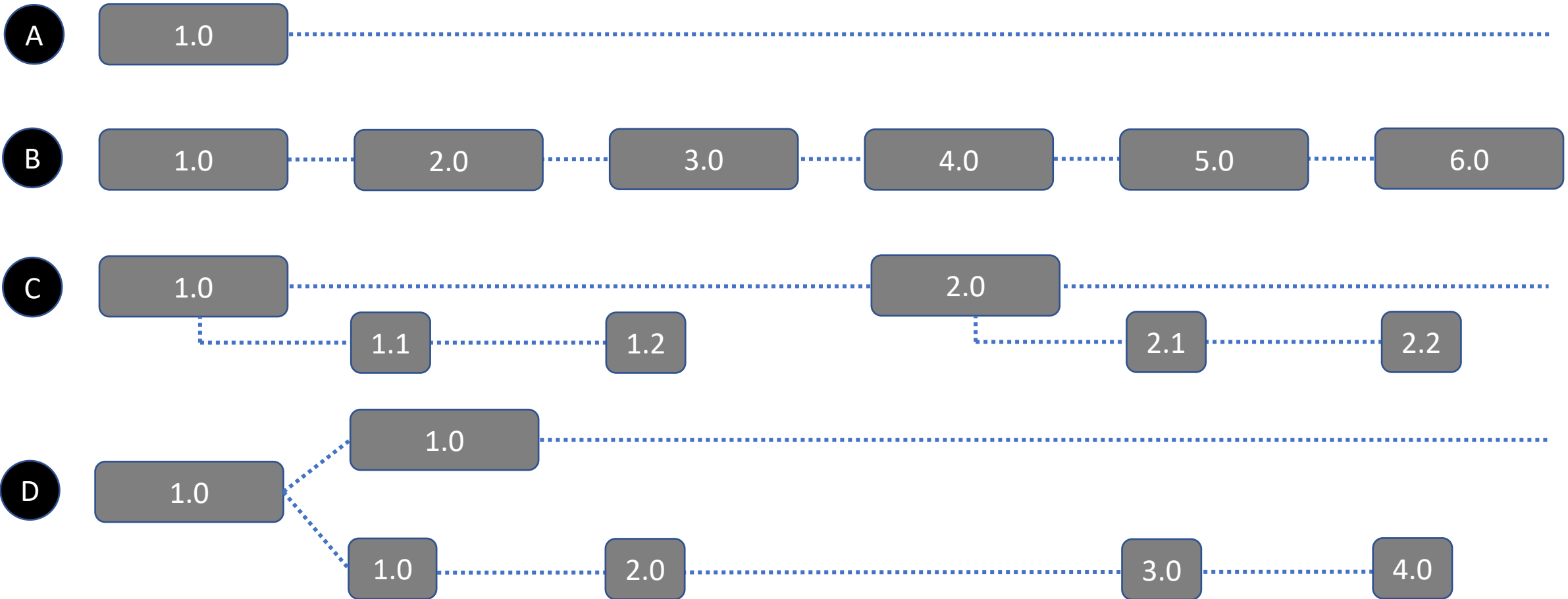
We create a new IANA registry, "*MoQTransport Catalog Type*". The registry policy is "RFC Required". The Type value is varint. The range is 0x0000-0xFFFF.

Type	Name	RFC
0x0000	Reserved	
0x0001	<i>WARP</i>	<i><a href="https://www.ietf.org/rfc/rfcXXXX.txt">https://www.ietf.org/rfc/rfcXXXX.txt</a></i>
0x0002	<i>QUICR</i>	<i><a href="https://www.ietf.org/rfc/rfcYYYY.txt">https://www.ietf.org/rfc/rfcYYYY.txt</a></i>
0x0003	<i>VR special etc.</i>	<i><a href="https://www.ietf.org/rfc/rfcZZZZ.txt">https://www.ietf.org/rfc/rfcZZZZ.txt</a></i>

Type 0 is reserved for experimentation. It allows a streaming format under development to be distributed by a delivery system without triggering behaviors associated with a registered type.

# Dependency examples

The catalog at the start of a Group MUST be independent and represent the latest view of the content. Subsequent objects within that Group MAY hold delta-encodes of the catalog



# What might a catalog look like #1?

varint + moov

# What might a catalog look like #2?

*Shown as JSON for clarity, but would be a binary encoding*

```
{
  "version":"1.00.68",
  "tracks":[
    {
      "id":1,
      "bitrate":2000000,
      "init":"HhyQJXIGpDv09tr0+0bzwlx16Av1yI4UDxHFfoP75/nGSOPmBIJU7raWKy578Jj8Xa5KeKFDbwuFhME8NQbVOKoGk/EX1
SPJRVkzNtuyaq/vruZav4SA4S+ImZ+hEQN2KvCm5itRy01LpZBsbnczQYgadHr3eQU/b0f1ihUjHM60uxq/FUMO/Oa5htNu67+nLZMoe6dv6X
qKeY4zf39RLW9s1TXNnb3lcEs5ysQI8KiPwmgD0OJUmcJWVtcpBOfT0puIYUiOBauqjfG5xWYQUSMnHysgPO7wbIDNNhC2vGg7CwVToACg
40haM3IHpRpKrxE7cMMhHrFhhtParil"
    },
    {
      "id":2,
      "bitrate":128000,
      "init":"iD9P8Ju1glXa94QXE6mpMoC4j4pvB/7V9a35RyCTxJTrdR4qr0wN7OHtX6IRg7WnmHkK2pbrHmuWw+NH5pHhEFmoTL
L2X9iwSel/jdLXEQcYCKqzeQf0Ar/HcHwClo5LDP6ATDJCi6mBZmvJAqEUreVFSXz+7AH7UbG9QiCpvgNhYELA4VVGmsS13LZ39PI3IPW6e1Q
3GRTYOXtclpXZ5Bgny/Z36L743yn/R6bl2Lt42c5kC4WKeIPnoIPjq61VavtXgb6ol5HXdPdQYSD5Bz8HWlay9SePlqaFmA0ezsGUZ7/vxFGJfyDg
m8tWbrZ2uUbP3KeV5U3SWZ6p/w4k"
    }
  ]
}
```

# What might a catalog look like #3?

```
{
  "version":"1.00.68",
  "label":"RERUN: FaZe vs. NaVi - Map 1 - EPL S17 – Semifinal",
  "tracks":[
    {
      "id":gamestreaming.com/abc/12345/video,
      "bitrate":2000000,
      "mimeType":"video/mp4",
      "codec":"avc1.64001f",
      "width": 1920,
      "height":1080,
      "frameRate":30,
      "label":"Primary game video feed",
      "initTrackID":gamestreaming.com/abc/12345/video-init,
    },
    {
      "id": gamestreaming.com/abc/12345/audio,
      "bitrate":128000,
      "mimeType":"audio/mp4",
      "codec":"mp4a.40.5",
      "sampleRate": 48000
      "label":"English commentary",
      "initTrackID":gamestreaming.com/abc/12345/audio-init,
    } ...
  ]
}
```

# What might a delta catalog look like #4?

```
{  
  "version": "2.4",  
  "sequence": 214,  
  "parent": 213,  
  "removeTracks": [2,3],  
  "addTracks": [{ "id": 4, "bitrate": 1000000, "type": "video", "name": "Bob", "dimensions": "1280x720" }],  
  "changeTrack": [{ "id": 1, "bitrate": 500000 }]  
}
```

# What might a DASH catalog look like #5?

```
<?xml version="1.0" encoding="utf-8"?>
<MPD xmlns="urn:mpeg:dash:schema:mpd:2011" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
availabilityStartTime="1970-01-01T00:00:00Z" maxSegmentDuration="PT2S" minBufferTime="PT2S" profiles="urn:mpeg:dash:profile:ietf-
moq:2023" publishTime="2023-03-30T07:25:34Z" timeShiftBufferDepth="PT5M" type="dynamic"
xsi:schemaLocation="urn:mpeg:dash:schema:mpd:2011 DASH-MPD.xsd">
<Period id="p0" start="PT0S">
  <AdaptationSet contentType="audio" lang="en" mimeType="audio/mp4" segmentAlignment="true" startWithSAP="1">
    <Role schemeldUri="urn:mpeg:dash:role:2011" value="main" />
    <SegmentTemplate duration="2" initializationTrackID="init/$RepresentationID$" mediaTrackID="audio/$RepresentationID$" />
    <Representation audioSamplingRate="48000" bandwidth="48000" codecs="mp4a.40.2" id="A48">
      <AudioChannelConfiguration schemeldUri="urn:mpeg:dash:23003:3:audio_channel_configuration:2011" value="2" />
    </Representation>
  </AdaptationSet>
  <AdaptationSet contentType="video" maxFrameRate="60/2" maxHeight="360" maxWidth="640" mimeType="video/mp4"
minHeight="360" minWidth="640" par="16:9" segmentAlignment="true" startWithSAP="1">
    <Role schemeldUri="urn:mpeg:dash:role:2011" value="main" />
    <SegmentTemplate duration="2" initializationTrackID="init/$RepresentationID$" mediaTrackID="video/$RepresentationID$" />
    <Representation bandwidth="300000" codecs="avc1.64001e" frameRate="60/2" height="360" id="V300" sar="1:1" width="640" />
  </AdaptationSet>
</Period>
</MPD>
```

Questions?