

WARP Draft Overview

Updates & Proposals

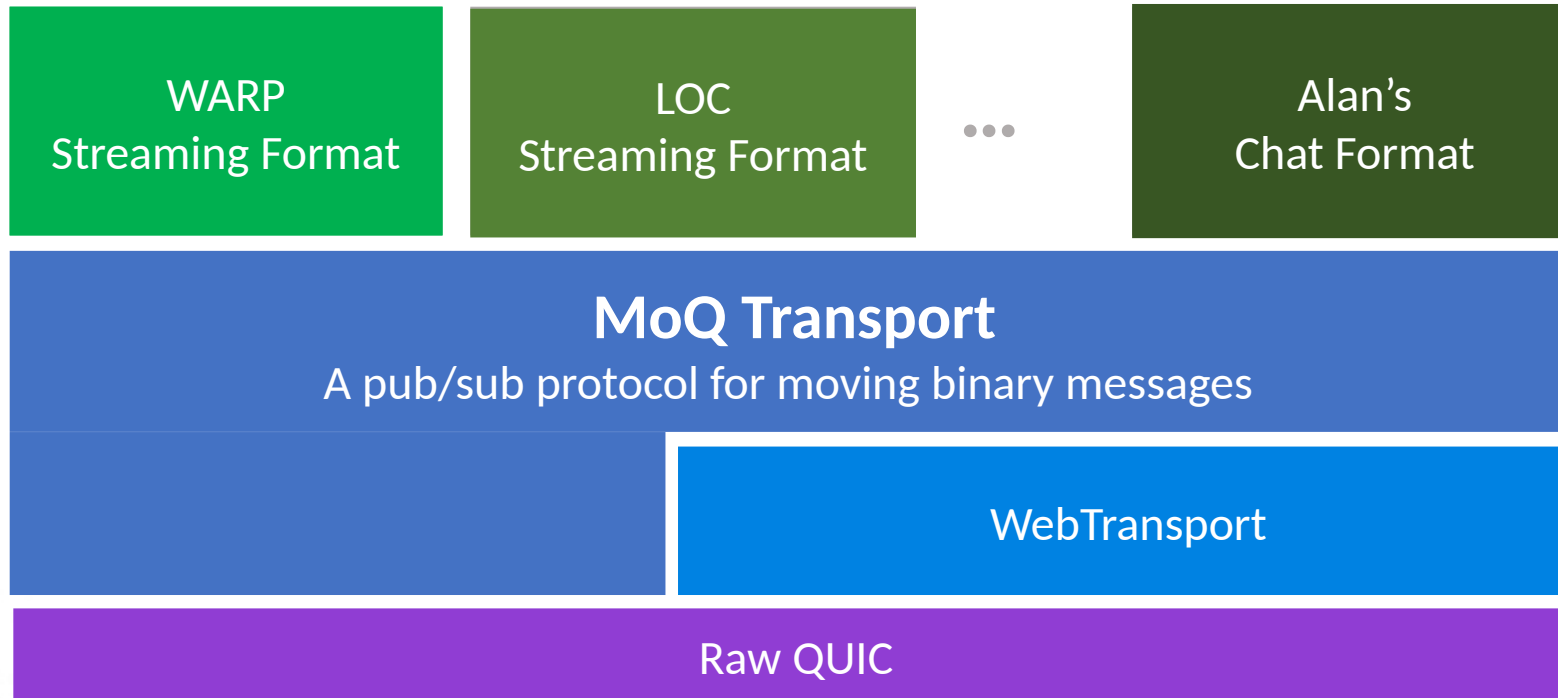
IETF #117, July 2023

Will Law

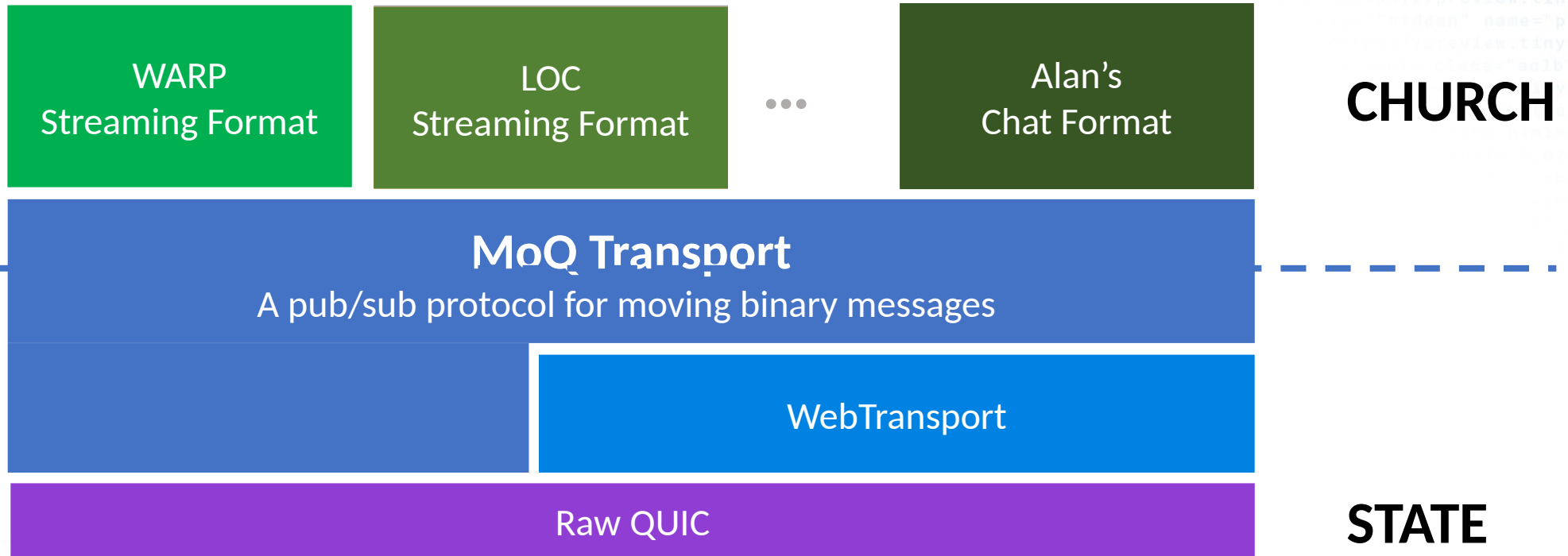
Repo: <https://github.com/moq-wg/warp-streaming-format>

Issues: <https://github.com/moq-wg/warp-streaming-format/issues>

What is a MoQ Streaming Format?



What is a MoQ Streaming Format?



WARP features - high level

- Designed to deliver CMAF packaged content with good interoperability with existing encoders, packaging systems and player decoders.
- Focus on live and near-live streaming
- WARP works by fragmenting the bitstream into objects that can independently be transmitted via QUIC streams.
- Congestion response - intra-track stream prioritization combined with track adaptation.
- Other details already described during interim and available here: <https://datatracker.ietf.org/meeting/interim-2023-moq-08/materials/slides-interim-2023-moq-08-sessa-warp-streaming-format-00>

Catalog objects

- A catalog is a track that describes the availability of other tracks. It is a directory of what is being published under a content "bundle".
- Each catalog object describes changes which add or delete a track.
- Catalog objects can represent deltas from prior objects.
- The init header serves dual purposes:
 - information the client needs to select the track (resolution, codec, bitrate etc)
 - initializing the decoder.

```
CATALOG payload {
  media format type (i),
  version (i),
  parent object sequence (i),
  track change count (i),
  track change descriptors (..)
}
```

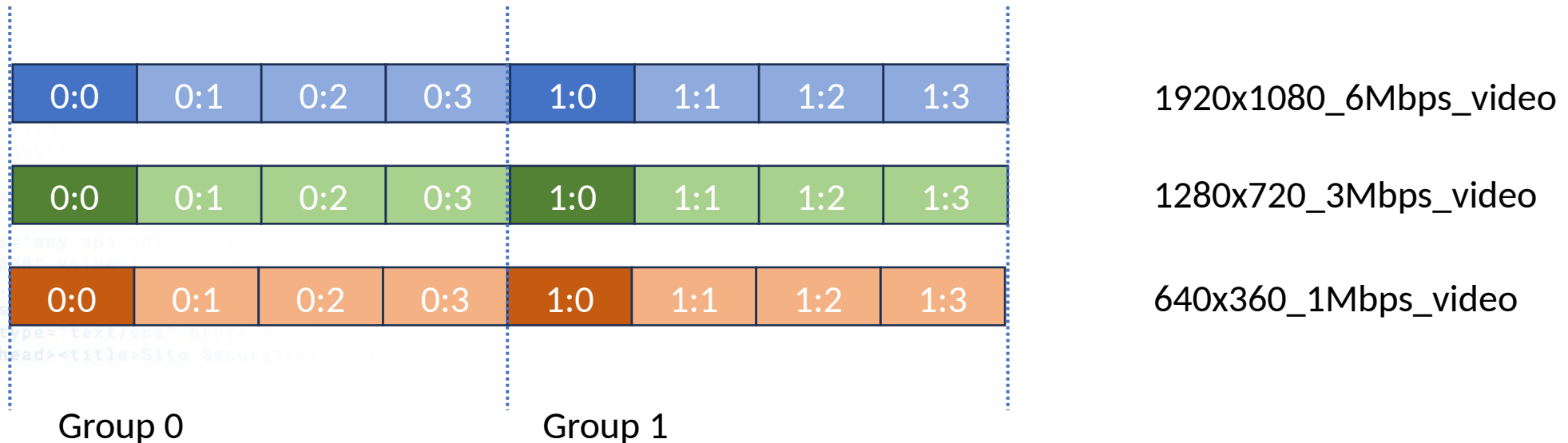
```
Track Change Descriptor {
  full track name length (i),
  full track name (..),
  operation (1),
  change payload(..)
}
```

```
Initialization Header {
  init length (i),
  init payload (..)
}
```

```
Deletion Header {
  Last group: (i),
  Last object: (i)
}
```

Packaging

- Each codec bitstream **MUST** be packaged into a sequence of **Objects** within a separate track.
- Media tracks **SHOULD** be media-time aligned. CMAF Switching Sets meet this requirement. A receiver **SHOULD** be able to cleanly switch between media tracks at group boundaries.
- Each group **MUST** be independently decodable.

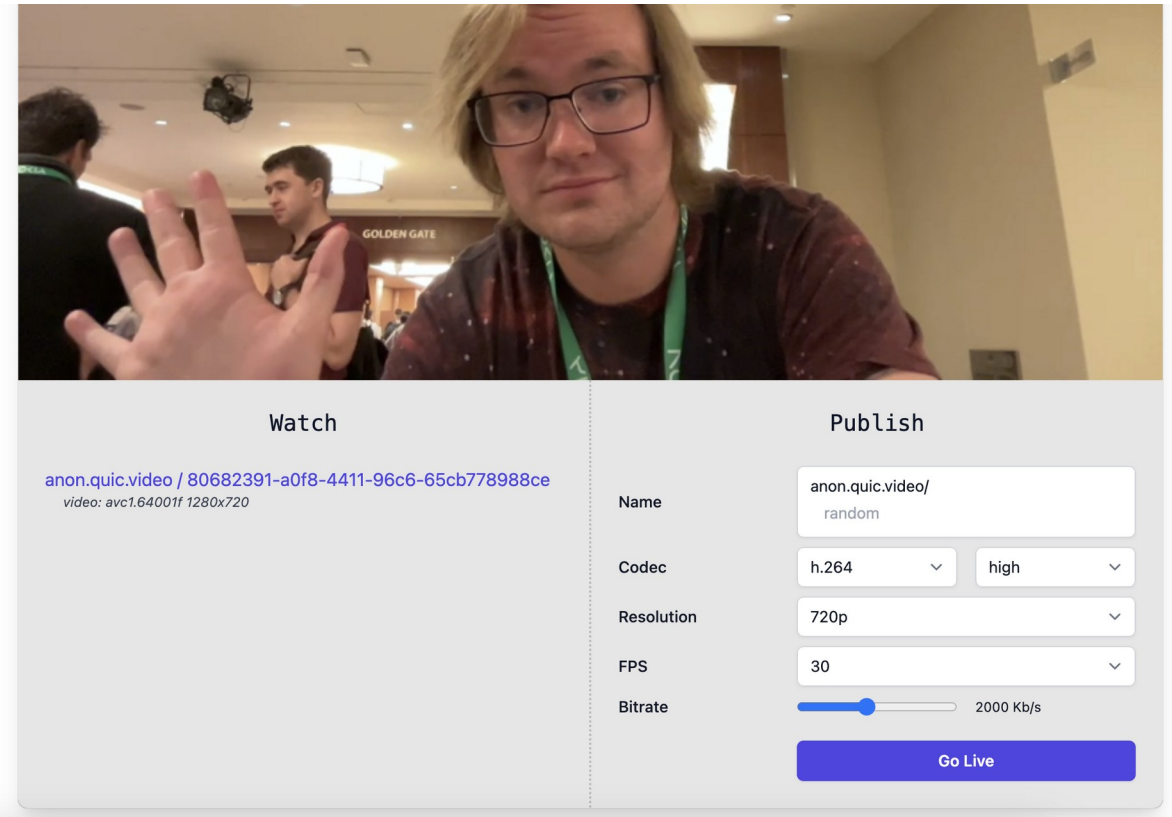


Does WARP Streaming Format work?

Yes. Luke Curley built a [Rust](#) (server+native) and a [Javascript](#) (web) library implementing moq-transport and warp-streaming format.

Modifications necessary to WARP

- Catalog renamed from 'catalog' to '0' (track names in mp4 are ints).
- Versioning not implemented.



Are we done?

No. There are multiple issues which need to be addressed with our architecture for moq streaming formats.

I will lay these out over the next several slides, and also propose a new catalog structure that addresses some of these problems.

We will then use our second slot on Wednesday to discuss and ideally achieve consensus on these issues before committing solutions to PRs.

Issue #1: How does a publisher simultaneously publish multiple different streaming formats?

If a publisher wants to output WARP content and Alan's Chat at the same time, they can't both produce a catalog called "catalog". On the receiver side, how does a client ask for a specific streaming format? Possible solutions:

1. Each streaming format defines a unique name or number for its catalog. We register these in an IANA table to ensure uniqueness.

```
example.com/path/catalog-warp  
example.com/path/catalog-achat
```

```
example.com/path/warp  
example.com/path/achat
```

```
example.com/path/0  
example.com/path/3
```

2. We allow hierarchical catalogs which can reference other catalogs. Thus, the first catalog loaded would be a description of the other available catalogs,

```
example.com/path/catalog  
{  
  type:0, example.com/path/alpha  
  type:3, example.com/path/beta  
}
```

3. We have a single catalog that describes all available streaming formats produced by that publisher.

```
example.com/path/catalog  
{  
  { type:0,  
    ...  
  },  
  { type:3,  
    ...  
  },  
}
```

4. We allow the application to transfer the format knowledge

```
example.com/path/warp/catalog  
example.com/path/achat/catalog
```

Problem #2: Should WARP continue to support only CMAF?

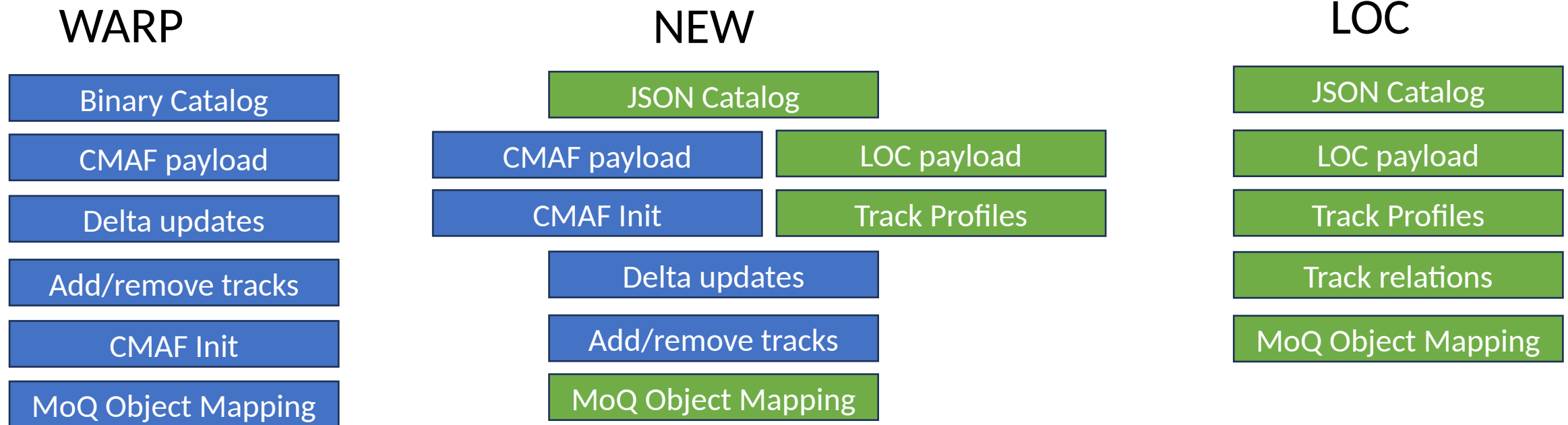
Arguments for only supporting CMAF:

- A focus on CMAF allows for constrained endpoint development.
- Well developed packager and decoder ecosystem
- Simple conversion (or alternate distribution path) for HLS/DASH content

Arguments for allowing alternate audio/video containers

- Do we really want to make a new streaming format every time we have a new container?
- CMAF is heavyweight encapsulation for real-time audio.
- CMAF is not natively supported by WebCodecs.
- People would like to mix CMAF for video with audio in a lighter-weight container, within the same catalog.
- Most components of a catalog are common. We can break out containers into separate specs and then just reference them from a common catalog specification.

Problem #3: Are WARP and LOC sufficiently different that we need separate streaming formats?



Streaming format architecture

Base drafts for catalog and packaging

Catalog draft



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

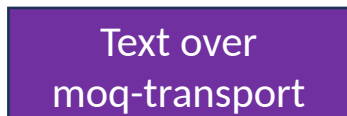
Packaging drafts



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

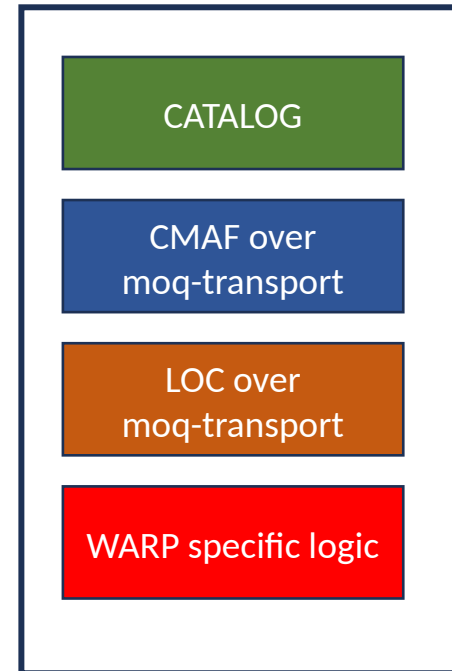


Specifies how to package LOC content for carriage over a moq-transport/catalog environment

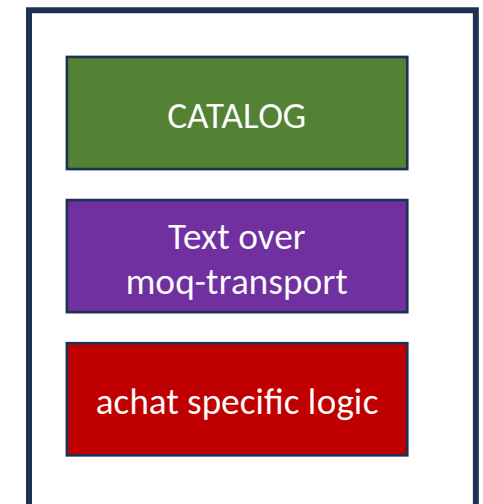


Specifies how to package text content for carriage over a moq-transport/catalog environment

WARP



Alan's chat



Proposed new harmonized catalog format

Example #1: Sports broadcast, CMAF, HD/SD video and audio track.

```
{
  "format": 1, // specifies the WARP streaming format in the IANA registry
  "version": 1, // the WARP version, to allow future extensions
  "namespace": "sports.example.com/games/12345", // track namespace, inherited by all tracks.
  "timeAligned": ["hd","sd","a"], // specifies that these tracks are time-aligned
  "tracks":[ // array of track change operations. Add is default
    {
      "name": "hd", // track name
      "packaging": "cmf", // packaging type, an enumerated string specifying CMAF
      "init": "ZGZkZiBvYWRmZyA7YWtkamZ ... tvYWl3dWVybnR2YWVydA==" //base64 encoded init
    },
    {
      "name": "sd",
      "packaging": "cmf",
      "init": "bW5id2VydGtqd2N1cXl3ZXJ ... MzQ5NTg2NzIzOTQ3ODU="
    },
    {
      "name": "a",
      "packaging": "cmf",
      "init": "CjUycDZveTU0aXUyb3BjNnU1NDI2OD ... bm1rbDY1NDk2LTU0cnI="
    }
  ]
}
```

Proposed new harmonized catalog format

Example #2: Terminating the sports broadcast and removing all tracks.

```
{
  "format": 1,
  "version": 1,
  "namespace": "sports.example.com/games/12345",
  "tracks": [
    {
      "name": "hd", // track name
      "operation": 0 // an enumerated track operation - remove
    },
    {
      "name": "sd",
      "operation": 0 // remove
    },
    {
      "name": "a",
      "operation": 0 // remove
    }
  ]
}
```

Proposed new harmonized catalog format

Example #3: Conference user Alice, capable of sharing lip-synced audio and video

```
{
  "format": 1, // specifies the WARP streaming format in the IANA registry
  "version": 1, // the WARP version, to allow future extensions
  "namespace": "conference.example.com/conference123/alice", // track namespace, inherited by all tracks.
  "lipsync": ["audio", "video"], // specifies that these tracks are lip-sync'd
  "tracks": [ // array of track change operations.
    {
      "name": "video", // track name
      "packaging": "loc", // packaging type, an enumerated string specifying loc packaged content
      "qualityProfile": "cs=av01.0.08M.10.0.110.09,wd=1920,ht=1080,fr=30" // quality profile parameters
    },
    {
      "name": "audio",
      "packaging": "loc",
      "qualityProfile": "cs=opus,sr=48000,cc=2"
    }
  ]
}
```

Proposed new harmonized catalog format

Example #4: Conference user Alice, issuing a delta update to add slide share

```
{
  "format": 1,
  "version": 1,
  "parentSequenceNumber": 1, // parent object sequence number, specifying that this is a delta update
  "lipsync": ["audio", "video", "slides"],
  "tracks": [
    {
      "name": "slides",
      "packaging": "loc",
      "qualityProfile": "cs=av01.0.08M.10.0.110.09,wd=1920,ht=1080,fr=15"
    }
  ]
}
```

Proposed new harmonized catalog format

Example #5: Mixed broadcast, using CMAF for video and OPUS/LOC for audio

```
{
  "format": 1,
  "version": 1,
  "namespace": "hybrid.example.com/12345",
  "timeAligned": ["video","audio"],
  "tracks":[
    {
      "name": "video", // track name
      "packaging": "cmf", // packaging type, an enumerated string specifying CMAF packaged content
      "init": "ZGZkZiBvYWVmZyA7YWtkamZ ... tvYWl3dWVybR2YWVydA==" //base64 encoded init
    },
    {
      "name": "audio",
      "packaging": "loc",
      "qualityProfile": "cs=opus,sr=48000,cc=2"
    }
  ]
}
```

Proposed new harmonized catalog format

Example #6: Sports broadcast, mixing in CMAF, LOC and a captions track from a different namespace.

```
{
  "format": 1,
  "version": 1,
  "namespace": "sports.example.com/games/12345", // track namespace, inherited by all tracks.
  "timeAligned": ["video","audio"],
  "lipsynced": ["captions","audio"], // captions are lip-synced with audio
  "tracks":[
    {
      "name": "video",
      "packaging": "cmf",
      "init": "ZGZkZiBvYWRmZyA7YWtkamZ ... tvYWl3dWVybnR2YWVydA=="
    },
    {
      "name": "audio",
      "packaging": "loc", // different packaging format
      "qualityProfile": "cs=opus,sr=48000,cc=2"
    }
  ]
  {
    "namespace": "captions.example.net/external/12345", // new namespace, overrides inheritance
    "name": "captions",
    "packaging": "cmf",
    "init": "bW5id2VydGtqd2N1cXl3ZXJ ... MzQ5NTg2NzIzOTQ3ODU="
  }
}
```

Suggested additional Catalog optimizations

- Only add Parent Sequence number for delta updates
- Collapse streaming format type and version to a tuple e.g. "1:2.1"
- Should quality parameters be described for CMAF tracks? The reason to do this is for Javascript clients which don't have mp4 parsers and which need to make track selection decisions.
- Listing all INIT segments makes for a larger CMAF catalog. What if each init was in its own track? The client would then subscribe only to the inits that it needed. Init track can be subscribed-to in parallel with the media segments track to reduce receipt delay.
- JSON offers a human readable catalog which is useful for comprehension & debugging. We can put JSON inside binary, or binary inside JSON. But let's pick one.
- Track descriptors should also carry these optional attributes:
 - 'label' attribute for UI display
 - a 'role' designator (caption, subtitle, dub, description, forced-subtitle etc)
 - an 'accessibility' designator,
 - a 'ratings' designator for broadcast compliance