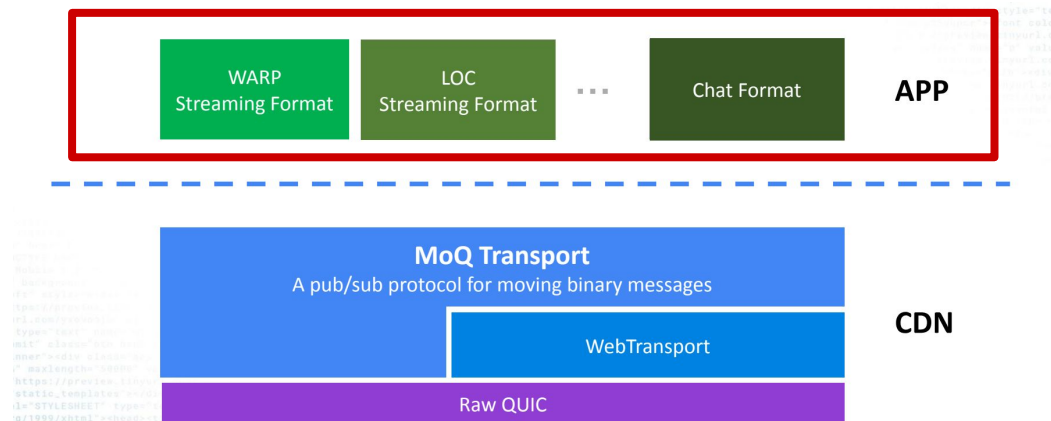


WARP・LOC

ストリーミングフォーマット概要

ストリーミングフォーマットの立ち位置



<https://datatracker.ietf.org/meeting/119/materials/slides-119-moq-sessa-warp-draft-update-for-ietf119-00>

MOQT: メディアの送り方

ストリーミングフォーマット: メディアの中身

WARP Streaming Format

- CMAF準拠のコンテンツをQUIC上で配信するためのフォーマット
 - 互換性を持たせることで既存プロトコルから MoQTへの移行を容易にする

> WARP leverages a simple prioritization strategy of assigning newer content a higher delivery order, allowing intermediaries to drop older data, and video over audio, in the face of congestion.

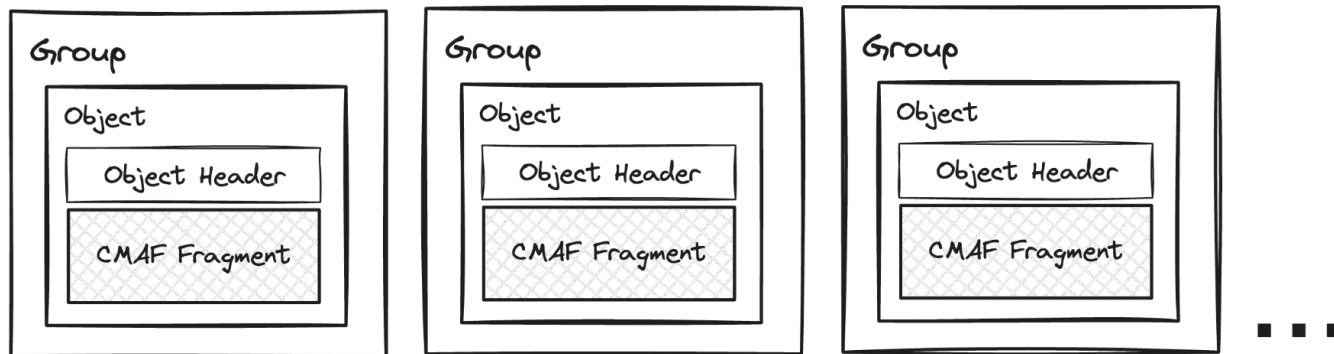
- 古いコンテンツより新しいコンテンツを優先する
- 輻輳発生時は映像より音声トラックを優先する
 - ビデオ会議など、映像が止まっても音声聞こえてれば耐えるケースは多い

パッケージングの方法

1. frag-per-group

CMAFフラグメント1つに対してMOQTグループ1つ。この場合MOQTグループは1つのMOQTオブジェクトを持つ

Track

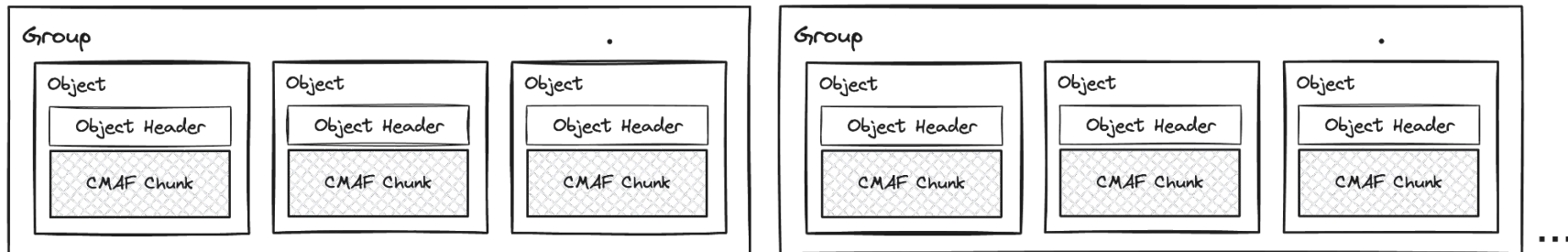


パッケージングの方法

2. chunk-per-object

CMAFチャンク(フラグメントより細かい単位)1つに対してMOQTオブジェクト1つ。同じ親フラグメントを持つCMAFチャンクは同じMOQTグループに入る

Track



WARPのその他

- MOQTオブジェクトのペイロードに対してAESアルゴリズム、CBCモードでの暗号化が推奨される
- Common Catalog Formatを使う
- CMAFの具体的なパッケージ方法については”CMAF Packaging for moq-transport”に移行(先月Expireしてる)

Low-Overhead Container

- WebCodecsのフォーマットをそのまま使う
 - WebCodecのEncodedVideoChunk, EncodedAudioChunkのペイロード部分を利用する

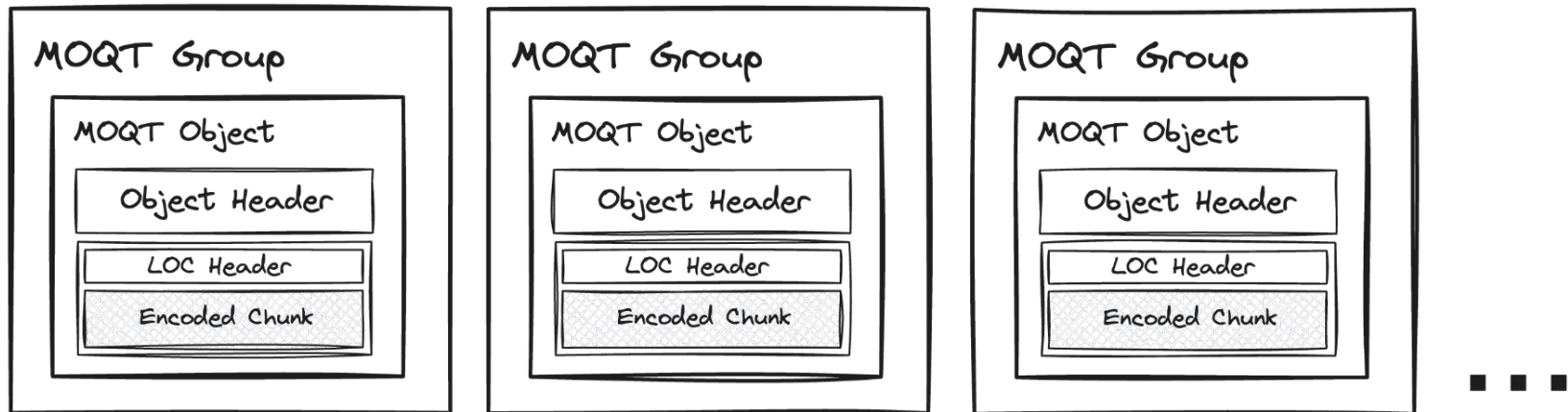
```
interface EncodedVideoChunk {  
  constructor(EncodedVideoChunkInit init);  
  readonly attribute EncodedVideoChunkType type;  
  readonly attribute unsigned long long timestamp; // microseconds  
  readonly attribute unsigned long long? duration; // microseconds  
  readonly attribute ArrayBuffer data;  
};
```

- WebCodecsによってエンコードされたチャンクのペイロード部分は生のデータ("raw elementary bitstream")
- CMAFなどはMOQTにおいて不必要な階層構造を持つが、LOCはこの点でオーバーヘッドが少ない

パッケージングの方法

WebCodecsでエンコードされたチャンク(EncodedVideoChunk / EncodedAudioChunk)に対して一つのMOQTオブジェクト

Track



LOC Header

音声・映像共通のプロパティ

- Sequence Number: オブジェクトごとに1増加する整数値
 - オブジェクト=フレームの場合これは省略できる
- Capture Timestamp: タイムスタンプ(マイクロ秒)
 - >Capture Timestamp in Microseconds: Captures the wall-clock time of the encoded media frame in a 64-bit unsigned integer.
- その他のプロパティは自由に設定できる

Metaの実装におけるLOC Frame定義

LOC Frame

dataType (varInt) 0 = Data, 1 = Audio, 2 = Video
chunkType (varInt) 0 = Delta, 1 = Key
seqId (varInt)
Timestamp (varInt) In nanoseconds
Duration (varInt) In nanoseconds
wallClock (varInt) In EPOCH milliseconds
metadataSize (varInt) In bytes, it can be 0
Metadata payload (metadataSize bytes) Format is the one delivered by WebCodecs
Encoded video / audio payload (X bytes)

```
SetData (mediaType, timestamp, duration, chunkType, seqId, firstFrameClkms, metadata, data) {  
  const pId = btoa(`${mediaType}-${timestamp}-${chunkType}-${seqId}-${Math.floor(Math.random * 100000)}`)  
  
  this.seqId = seqId  
  this.timestamp = timestamp  
  
  this.mediaType = mediaType  
  this.duration = duration  
  this.chunkType = chunkType  
  this.firstFrameClkms = firstFrameClkms  
  
  this.pId = pId // Internal  
  
  this.metadata = metadata  
  this.data = data  
}
```

参考文献

- <https://datatracker.ietf.org/doc/draft-law-moq-warpsstreamingformat/>
- <https://datatracker.ietf.org/doc/draft-mzanaty-moq-loc/>
- <https://datatracker.ietf.org/doc/html/draft-wilaw-moq-cmafpackaging-00>
- <https://datatracker.ietf.org/meeting/interim-2023-moq-08/materials/slides-interim-2023-moq-08-sessa-warp-streaming-format-00>
-