

**ISO/IEC JTC 1/SC 29/WG 11**

**Coding of moving pictures and audio**

**Convenorship: UNI (Italy)**

**ISO/IEC JTC 1/SC 29/WG 11 N 19256**

**Document type: Approved WG 11 document**

**Title: Exploration on Multi-stream Support for CMAF**

**Status: Approved**

**Date of document: 2020-04-24**

**Source: Systems**

**Expected action:**

**No. of pages: 14**

**Email of convenor: leonardo@chiariglione.org** <mailto:qianheng@sdis.cn>

**Committee URL: mpeg.chiariglione.org**

**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC1/SC29/WG11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC1/SC29/WG11 MPEG 130/N19256**

**Alpbach, AT – April 2020 (virtual)**

|  |  |
| --- | --- |
| **Source** | **Systems** |
| **Title** | **Exploration on Multi-stream Support for CMAF** |

Exploration on Multi-stream Support for CMAF

# Introduction

This document responds to the exploration on multistream in N19017.

CMAF defines constraints on ISOBMFF files and in particular restricts the number of ISOBMFF tracks in CMAF Track to one. This document gathers uses cases for *Multistream* applications from WG11 members as well as external organizations. A Multistream application is defined in this document as an application using multiple single-track CMAF files, whose content is processed at the client side by a single decoder. This document also identifies initial requirements on extensions to CMAF to address these use cases.

# Scenarios

## Audio Multistream Scenarios

For NGA audio codecs, there is the possibility of kinds of audio being in separate tracks. Additionally, audio codecs and delivery systems that support advanced layering may result in additional tracks from those listed in the section below.

### Definitions:

**Track** = DASH ISOBMFF track

**Complete Main** = Music & Effects (M&E) plus Dialog

**Alternate Dialog** (in the same Language) =  e.g. Director’s commentary, Sports Home city commentary, Sports Away city commentary, (NASCAR) In-car audio, etc.

**Overlay Dialog** (in the same language) = e.g. Video Description Service (VDS) aka Audio Description.

**Alternate Complete Main** (in the same language) = e.g. Emergency Information (EI)

### Track Combinations:

1. Complete Main
   1. One track
   2. Two tracks
      1. M&E
      2. Dialog
2. M&E with Alternate Dialog
   1. One track
   2. Two tracks
      1. M&E
      2. Alternate Dialog
3. Complete Main with Overlay Dialog
   1. One track
   2. Two tracks
      1. Complete Main
      2. Overlay Dialog
   3. Three tracks
      1. M&E
      2. Dialog
      3. Overlay Dialog
4. Alternate Complete Main
   1. One track

### MPEG-H Audio Multistream Applications

MPEG-H 3D Audio enables a variety of additional use-cases beyond the capabilities of traditional audio codecs. One of these use-cases that is especially relevant for http streaming environments is the “multi-stream” use case: The audio data that makes up one audio scene is not delivered in one single data stream but is distributed to a set of multiple streams than can be separately requested by the client.

Dividing the audio scene into a main stream (“channel bed”) and several dialogue streams is one example of the multi-stream use-case. This is more bit-rate efficient than creating full mixes for each language since only the main stream and one language are transmitted to the receiver. These streams are the input to a single decoder instance that decodes multiple streams and creates a single PCM output.

In order to enable separate delivery of those streams, each stream is stored in a single ISOBMFF track in a separate file.

## Video Multistream Scenarios

Video can be coded with both spatial and temporal layering. And, full presentations might also be used to deliver a “picture in picture” video overlay, where for example the overlay might show a sign language interpreter ("closed signing"). When picture in picture video is present, then there is an additional video track (up to 3 total) that needs to be decoded concurrently with those listed below.

### Temporal scalability

For many video codecs, temporal scalability and ISOBMFF can be used such that a base track contains frames resulting in a low frame rate while if additional frames from another track are used, the resulting frame rate is increased.

### AVC/HEVC Scalability and Multiview

AVC and HEVC define a set of multi-layer extensions to satisfy the needs of scalable applications or multiview applications. In such cases, the video elementary stream is composed of different sets of NAL units each associated to a given layer. Storage in the ISOBMFF enables distributing different layers to different tracks for separate delivery, and in some cases relies on NAL Unit Extractors to provide directions on how to reconstruct a complete bitstream.

### HEVC Tiling

HEVC defines track types to store parts of a coded pictures called tile. This is extensively used by Virtual Reality applications such as in OMAF. The tiles are delivered separately and combined at the client side and decoded by the same decoder.

## Use cases requiring encryption

We can distinguish at least three types of use cases based on encryption. Consideration for encryption use cases in w17548, section 2.3 creates the scenario where there is the additional use case where one track is encrypted and one track is not.

### Multi-tracks, one encrypted and one not (single key)

In some broadcast applications it is envisaged to use multistream content but not to encrypt one track but encrypt (with a single key) another track.

### Multi-tracks, single key

In some broadcast applications it is envisaged to use multistream content but to encrypt all streams, with one key.

### Multi-tracks, multiple keys

In other applications, it can be envisaged that alternative components can be charged for. For example, an unusual language dialog track that would not normally justify the translation cost.  Therefore, the base tracks are encrypted with one key and the additional dialog track would necessarily be with another key.

Another example of this category of use cases is third party dialog translation. An additional dialog track is free but delivered over broadband and encrypted by a third-party service without access to the base services key or crypto infrastructure.

## Auxiliary Tracks

HEIF and MIAF define image sequences (and items) possibly linked to auxiliary image sequences (and items) using multiple tracks. This can be used to provide an alpha channel or a depth channel.

In this use case, it may be possible under some constraints to use the same decoder. However, even if multiple decoders may be used, a single output could be produced, and therefore could be considered as *Multistream*.

## Track Combinations

1. No Spatial Scaling or Temporal Sub-layering:
   1. One Track
2. Only Temporal Sub-layering
   1. One track – both temporal sub-layers
   2. Two tracks
      1. Temporal sub-layer 0
      2. Temporal sub-layer 1
3. Only Spatial Scaling
   1. Two Tracks
      1. Base spatial
      2. Enhancement spatial
4. Both Spatial Scaling and Temporal Sub-layering
   1. Two Tracks
      1. Base spatial + both temporal sub-layers
      2. Enhancement spatial + both temporal sub-layers

# DASH Preselections

## Introduction

In the 4th edition of ISO/IEC 23009-1, Preselections in were updated to addressed different use cases including single stream and multistream interop. Please check clause 5.3.11. In the following, the definitions and relevant use cases are reviewed.

The text s reviewed for:

* Relevant for CMAF on container level
* Irrelevant for CMAF due to file format restrictions
* An open issue to be discussed in the context of CMAF
* Manifest signaling

## Relevant Specification Text

### Definitions

### Main Adaptation Set *Adaptation Set* (3.1.3) in a *Preselection* (3.1.35) that contains the *Initialization Segment* (3.1.19) for the complete experience

### Preselection set of *media content components* (3.1.22) that are intended to be consumed jointly

### General

The concept of Preselections was initially considered for the purpose of enabling Next Generation Audio (NGA) codecs to signal suitable combinations of audio elements that are offered in different Adaptation Sets. However, the Preselection concept is introduced in a generic manner such that it can also be applicable to and used by other media types and codecs.

Preselections define user experiences that can be selected by the DASH Client. Each Preselection is uniquely identifiable and distinguishable, e.g. by language. A Preselection encompasses a subset of media components such that the media components can be selected and combined into a complete experience.

Preselections can be used to reference a set of Representations from multiple Adaptation Sets in order to produce a complete experience. Preselections can also be used to indicate a pre-defined experience at the elementary-stream level, i.e. the DASH Client can select a pre-defined experience and provides the selection to the media engine.

Preselections may be uniquely identified by a Preselection Tag. Users/Codecs using this Tag functionality are encouraged to provide more information on how tags defined in the MPD map to functionality in the specific codec.

Preselections have equivalent annotation parameters to Adaptation Sets and are always assigned exactly one media type.

Media components can be mapped to Adaptation Sets in multiple ways:

1. by a one-to-one mapping between media components and Adaptation Sets;
2. by the inclusion of multiple media components in a single Adaptation Set where all encoded versions of the media components are multiplexed on the file-container level;
3. by the inclusion of multiple media components in a single Adaptation Set where all encoded versions of the media components are multiplexed on the elementary-stream level.

If the Adaptation Set contains a single media component, then the media component can be referenced by the @id of the Adaptation Set.

If the Adaptation Set contains multiple media components multiplexed on the file-container level, then each media component is mapped to a Content Component as defined in 5.3.4.

For example, in the ISO BMFF case, a Representation contains multiple tracks and each track is mapped to a Content Component. Therefore, media components can be referenced by the @id of an Adaptation Set or the @id of a Content Component. When Preselections reference Content Components, the @id of Adaptation Sets and Content Components shall be unique within the scope of a Period.

If the Adaptation Set contains multiple media components multiplexed at the elementary-stream level, then a pre-defined experience is referenced by the Preselection. For example, in the ISO BMFF case, a Representation contains a single track of multiple media components that is referenced by the @id of the Adaptation Set. Multiple Preselections can reference the Adaptation Set and select a pre-defined experience by passing the Presentation Tag to the media engine along with the media stream.

The Main Adaptation Set is the Adaptation Set that contains the Initialisation Segment for the complete experience. Each Preselection shall reference a Main Adaptation Set and may reference zero, one or more other Adaptation Sets.

NOTE   In the context of Preselection, the term "Main Adaptation Set" is used. This term is not to be confused with an Adaptation Set that has assigned the main Role.

Within a Preselection, two types of Adaptation Sets are differentiated:

* Main Adaptation Set: A Representation of this Adaptation Set is needed for playback of the Preselection. In particular for ISO BMFF, the Initialization Segment of such a Representation is needed for playback of the Preselection.
* Partial Adaption Set: A Representation of this Adaptation Set is only consumable together with the Main Adaptation Sets within this Preselection. Again, in particular for ISO BMFF, the Initialization Segment of a Representation of the Main Adaptation Set is needed for playback.

Preselections, main Adaptation Set and partial Adaptation Sets may be defined by one of the two means:

* A preselection descriptor
* A preselection element

The Preselection Descriptor is used for two purposes.

1. To indicate that an Adaptation Set is part of a Preselection.
2. Optionally, to provide instructions on how to combine the Adaptation Set with other Adaptation Sets to form a Preselection.

As an alternative to the Preselection descriptor, Preselections may also be defined through the Preselection element as provided in Table 25. The selection of Preselections is based on the contained attributes and elements in the Preselection element.

Table 25 — Semantics of PreSelection element

| **Element or Attribute Name** | | | | **Use** | **Description** |
| --- | --- | --- | --- | --- | --- |
|  |  | Preselection | |  |  |
|  |  |  | @id | OD  default=1 | specifies the id of the Preselection. This shall be unique within one Period. |
|  |  |  | @preselectionComponents | M | specifies the ids of the contained Adaptation Sets or Content Components that belong to this Preselection as white space separated list in processing order. The first id defines the Main Adaptation Set. |
|  |  |  | @lang | O | same semantics as in Table 5 for @lang attribute. |
|  |  |  | @order | OD  Default: 'undefined' | specifies the conformance rules for Representations in Adaptation Sets within the Preselection.  When set to 'undefined', the Preselection follows the conformance rules for Multi-Segment Tracks in subclause 5.3.11.5.1.  When set to 'time-ordered', the Preselection follows the conformance rules for Time-Ordered Segment Tracks in subclause 5.3.11.5.2.  When set to 'fully-ordered', the Preselection follows the conformance rules for Fully-Ordered Segment Tracks in subclause 5.3.11.5.3. In this case, order in the @preselectionComponents attribute specifies the component order. |

### Conformance rules

#### Conformance rules for Multi-Segment Tracks

Where multiple Adaptation Sets indicate this type of ordering, each Adaptation Set and the contained Representations follow the regular conformance rules for multi-segment tracks as defined in subclause 5.3.5.1.

No additional conformance rules are defined for the Representations in different Adaptation Sets within Preselections.

#### Conformance rules for Time-Ordered Segment Track

Where multiple Adaptation Sets indicate this type of ordering, each Adaptation Set and the contained Representations follow the conformance rules for multi-segment tracks as defined in subclause 5.3.11.5.1.

In addition, the concatenation of the following shall represent a conforming Segment track as defined in subclause 4.5.4 and conforming to the media type as specified in the @mimeType attribute for the Representation of the Main Adaptation Set:

* An Initialization Segment of one Representation of the Main Adaptation Set (specified by the first id in the @preselectionComponents attribute or the Preselection Descriptor), and
* media segments/subsegments of one Representation from each Adaptation Set referenced in the Preselection ordered by non-decreasing first decode times.

Note: This does not constrain the order of segments with the same first decode time.

If Adaptation Sets within a Preselection are time-ordered as defined above, they should share the same value for the @segmentAlignment or @subsegmentAlignment attributes, i.e. the Representations of all Adaptation Sets referenced by the Preselection are segment/subsegment aligned.

#### Conformance rules for Fully-Ordered Segment Track

Where multiple Adaptation Sets indicate this type of ordering, each Adaptation Set and the contained Representations follow the conformance rules for multi-segment tracks as defined in subclause 5.3.11.5.1.

In addition, the concatenation of the following shall represent a conforming Segment track as defined in subclause 4.5.4 and conforming to the media type as specified in the @mimeType attribute for the Representation of the Main Adaptation Set:

* An Initialization Segment of one Representation of the Main Adaptation Set (specified by the first id in the @preselectionComponents attribute or the Preselection Descriptor), and
* media segments/subsegments of one Representation from each Adaptation Set referenced in the Preselection ordered first by non-decreasing decode times and then by position in the list given in @preselectionComponents.

If Adaptation Sets referenced by a Preselection are fully ordered as defined above, they shall share the same value for the @segmentAlignment or @subsegmentAlignment attributes, i.e. the Representations of all Adaptation Sets referenced by the Preselection shall be segment/subsegment aligned.

# Proposed Text for CMAF

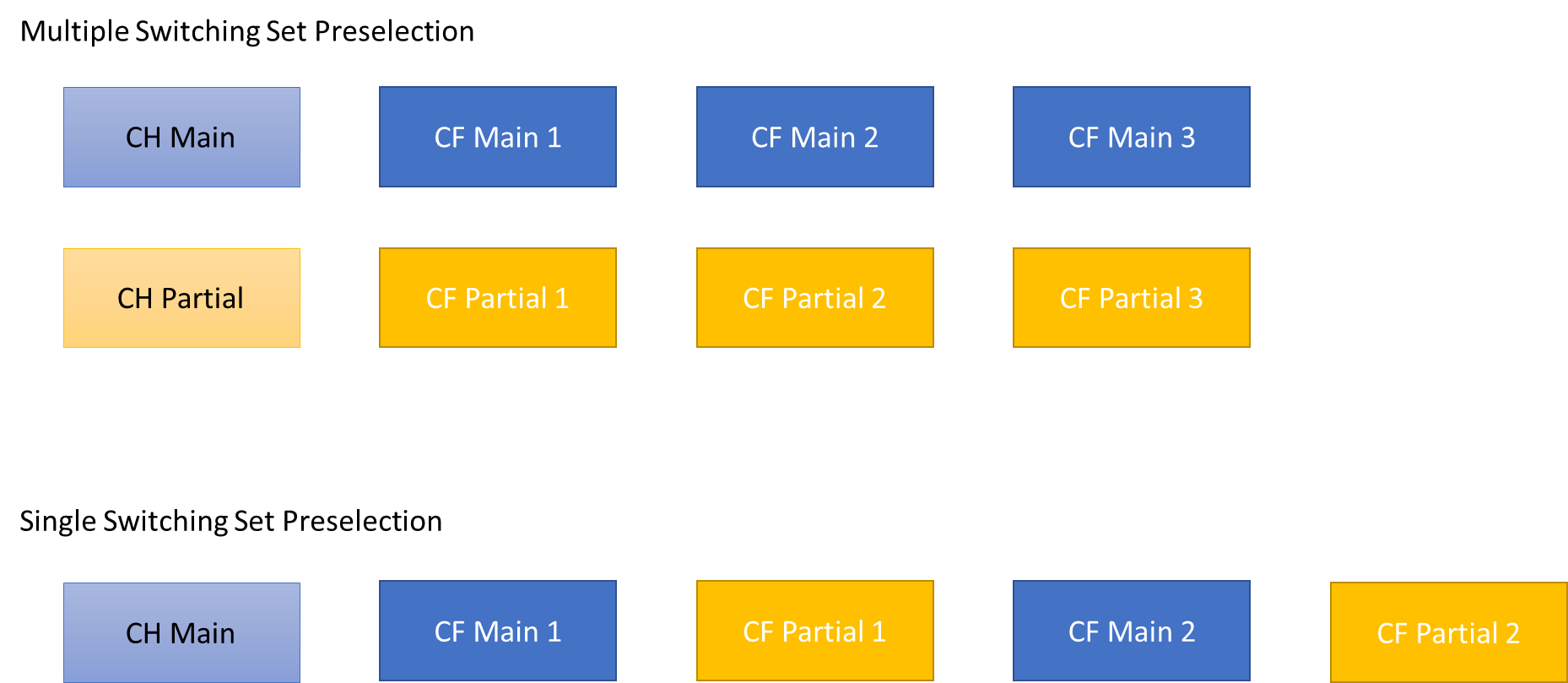
## 6.6.11 Preselections

### 6.6.11.1 Introduction

Preselections are used to combine a set of CMAF Tracks in order to produce a complete experience.

Two cases are differentiated:

1. The CMAF Tracks are contained in multiple CMAF Switching Set, and one track from each Switching Set is picked. For more details refer to 6.6.11.2.
2. The CMAF Tracks are contained in a single CMAF Switching Set, creating CMAF Tracks with multiple ISO BMFF Tracks in a single CMAF Track. A Preselection is a subset of ISO BMFF Tracks in the CMAF Track. For more details refer to 6.6.11.3.



### 6.6.11.2 Multiple Switching Set Preselections

For the case of Multiple Switching Set Preselections the following holds.

Within a Preselection, two types of Switching Sets are defined:

* Main Switching Set: A CMAF Track of this Switching Set is needed for playback of the Preselection. In particular, the CMAF Header of such a Switching Set is needed for playback of the Preselection.
* Partial Switching Set: A CMAF Track of this Switching Set is only consumable together with the Main Switching Set within this Preselection. Again, in particular, the CMAF Header of a CMAF Track of the Main Switching Set is needed for playback.

The combination of CMAF Structures from different CMAF Tracks from different Switching Sets may follow certain conformance rules that can be used for playback.

Manifest Signaling is out of scope for CMAF. However, an external manifest must provide an ability to signal the following:

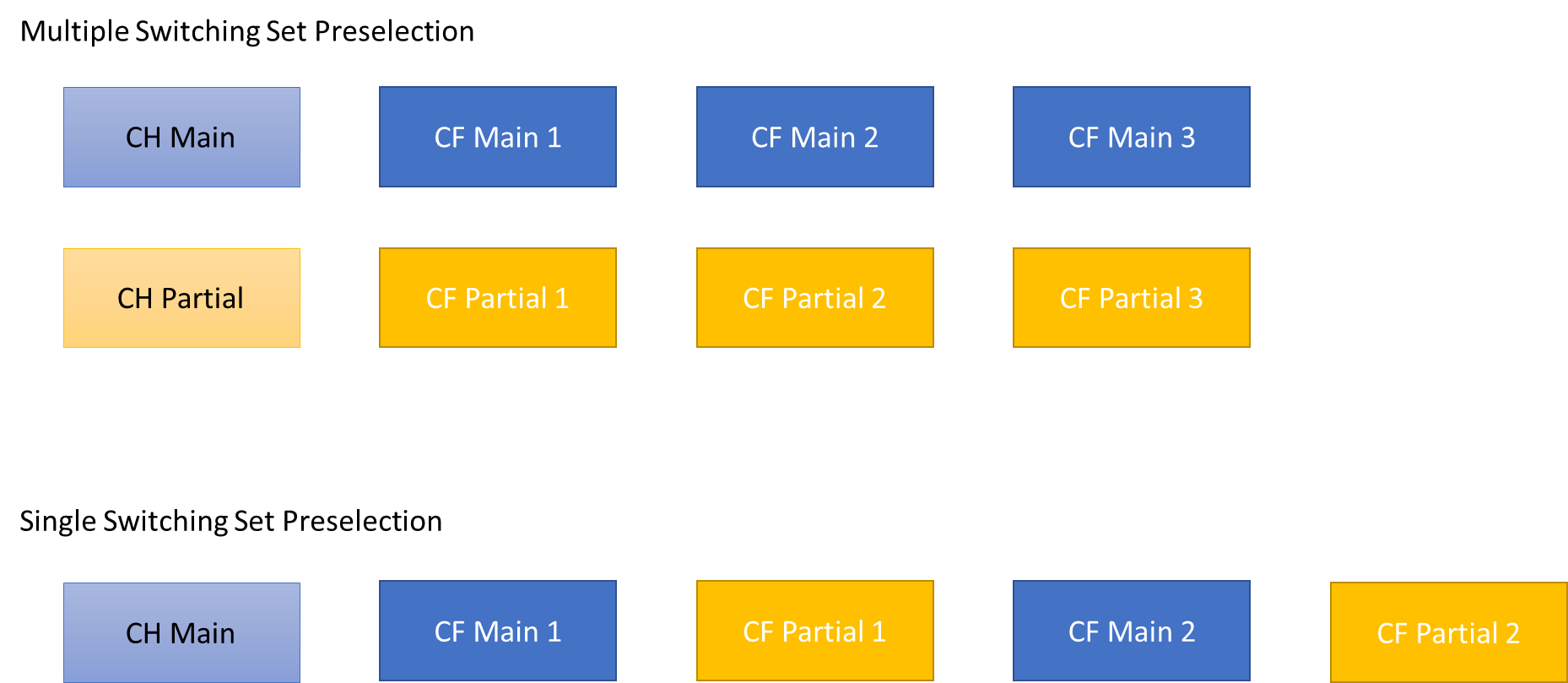
1. The CMAF Switching Sets that are included in a Preselection
2. The signaling, which Switching Set is the Main Switching Set in the Preselection
3. The signaling, if a partial Switching Set can be consumed independent of the main Adaptation Set
4. The conformance rules that apply for the combination of CMAF Structures from different Switching Sets within a Preselection.

### 6.6.11.3 Single Switching Set Preselections

<This issue needs more discussion, as now multiple ISO BMFF Tracks would be stored in a single CMAF Track/Switching Set. Preselection only selects a subset of the tracks for playback>

## 7.X Single Media Type Multi-Track CMAF Track

<We need to define a CMAF Track, that contains multiple ISO BMFF track, that can be consumed and played jointly.>



## 7.3.4a CMAF Preselections

### 7.3.4a.1 Introduction

Preselections are used to combine a set of CMAF Tracks in order to produce a complete experience.

Two cases are differentiated:

1. The CMAF Tracks are contained in multiple CMAF Switching Set, and one track from each Switching Set is picked. For more details refer to 7.3.4a.2.
2. The CMAF Tracks are contained in a single CMAF Switching Set, creating CMAF Tracks with multiple ISO BMFF Tracks in a single CMAF Track. A Preselection is a subset of ISO BMFF Tracks in the CMAF Track. For more details refer to 6.6.11.3.

### 7.3.4a.2 Multiple Switching Set Preselections

For the case of Multiple Switching Set Preselections the following holds.

Within a Preselection, two types of Switching Sets are defined:

* Main Switching Set: A CMAF Track of this Switching Set is needed for playback of the Preselection. In particular, the CMAF Header of such a Switching Set is needed for playback of the Preselection.
* Partial Switching Set: A CMAF Track of this Switching Set is only consumable together with the Main Switching Set within this Preselection. Again, in particular, the CMAF Header of a CMAF Track of the Main Switching Set is needed for playback.

The combination of CMAF Structures from different CMAF Tracks from different Switching Sets may follow certain conformance rules that can be used for playback.

Manifest Signaling is out of scope for CMAF. However, an external manifest must provide an ability to signal the following:

1. The CMAF Switching Sets that are included in a Preselection
2. The signaling, which Switching Set is the Main Switching Set in the Preselection
3. The signaling, if a partial Switching Set can be consumed independent of the main Adaptation Set
4. The conformance rules that apply for the combination of CMAF Structures from different Switching Sets within a Preselection.

Preselections are used to combine a set of CMAF Tracks from multiple Switching Sets in order to produce a complete experience.

Within a Preselection, two types of Switching Sets are defined:

* Main Switching Set: A CMAF Track of this Switching Set is needed for playback of the Preselection. In particular, the CMAF Header of such a Switching Set is needed for playback of the Preselection.
* Partial Switching Set: A CMAF Track of this Switching Set is only consumable together with the Main Switching Set within this Preselection. Again, in particular, the CMAF Header of a CMAF Track of the Main Switching Set is needed for playback.

A CMAF Preselection conforms to the following constraints:

* A CMAF Preselection shall contain one or more CMAF Switching Sets.
* All CMAF switching sets within a CMAF Preselection set shall be of the same media type, e.g. audio, video, or subtitles.
* All CMAF switching sets in a Preselection shall be of equal duration
* All CMAF switching sets in a Preselection shall contain the same number of CMAF fragments in every CMAF track
* All CMAF switching sets in a Preselection shall contain CMAF fragments in every CMAF track with matching earliest presentation time and duration, for each CMAF fragment in a CMAF track

A Time-Ordered CMAF Preselection conforms to the following constraints:

* A Timed ordered CMAF Preselection shall conform to a CMAF Preselection
* The ordering of the following shall represent a conforming Single Media Type Multi-Track CMAF Track:
  + A CMAF Header of one CMAF Track of the Main Switching Set, and
  + CMAF Fragments of one CMAF Track from each Switching Set included in the Preselection ordered by non-decreasing first decode times.

A Fully-Ordered CMAF Preselection conforms to the following constraints:

* A Fully ordered CMAF Preselection shall conform to a Time-ordered CMAF Preselection
* In addition, the ordering for CMAF Fragments from each Switching Set included in the Preselection ordered first by non-decreasing decode times and then by position in order of a well defined list.

### 7.3.4a.3 Single Switching Set Preselections

<This issue needs more discussion, as now multiple ISO BMFF Tracks would be stored in a single CMAF Track/Switching Set. Preselection only selects a subset of the tracks for playback>

# Evaluation of Requirements

A *Multistream* CMAF approach should define:

* Constraints permitting the use of multiple CMAF files, each containing a single track, but the set of files being related, this includes constraints:
  + On the relationships between the track types (e.g. targeted to the same decoder or not; ability to produce a single decoder output; combination of sample entries (hvc1+hvt1)…);

***[Response: The relationship between tracks is defined in the sample entry and information in the CMAF Header of the Main Switching Set.]***

* + On the relationships between tracks (decoding dependencies or not). This could be achieved by describing how those tracks would be signaled if hypothetically they were in the same file (e.g. use of track references, of track groups, signaling of a master track…)

***[Response: The relationship between tracks is defined in the sample entry and information in the CMAF Header of the Main Switching Set.]***

* + On the time-alignment of samples between tracks

***[Response: The relationship between tracks is defined in the sample entry and information in the CMAF Header of the Main Switching Set.]***

* + On the use of encryption

***[Response: The relationship between tracks is defined in the sample entry and information in the CMAF Header of the Main Switching Set.]***

* Concepts/Data Model permitting the use of these multiple single-track files when the application has out-of-band signaling of the relationship between tracks, e.g. for use by external specifications (DASH, …)

***[Response: The requirements for manifest signaling are provided.]***

* Constraints enabling the use of multistream content in low latency delivery scenarios

***[Response: this is an open question on how chunk interleaving can be done and is permitted. However, we should first complete the conceptual approach and then move into the refinement.]***

* Constraints when each single-track file is offered in different bitrates/quality/… for switching purposes

***[Response: No changes to what exists in CMAF.]***

Some special cases of multistream content could simplify the specification by CMAF of the above constraints:

* A) When an existing specification already indicates how to store the multistream content in the form of a multi-track ISOBMFF file.
* B) When an existing specification already indicates how to store the multitream content in a single track.

# Example workaround solutions

The form of a specification for the support of multistream content depends on the client capabilities. The following illustrates different options for the support by clients in for example, the HTML5 MSE environment.

## Native decoding of the multiple files

Obviously, the first solution to enable the above use cases is for client to natively support multiple input files. In that case, the client would, if needed, do the merge operations described above (merge of multiple files in a single file or merge multiple tracks in a single track). This is in particular applicable to browsers with the Media Source Extension.

## JavaScript multiplexing

Considering the special case of clients supporting JavaScript and the Media Source Extension API, an approach could be as described in m41705 to perform the merge operations described above in JavaScript without requiring native extension of the Media Source Extension API. Preliminary implementations (as reported in m42794) seem to indicate that such approach would suffice to satisfy some use cases but would not be sufficient for use cases requiring encryption.