Consumer Technology Association™

CTA Specification

Web Application Video Ecosystem – DASH-HLS Interoperability Specification

CTA-5005-A

June 2023

NOTICE

Consumer Technology Association (CTA)™ Standards, Bulletins and other technical publications are designed to serve the public interest through eliminating misunderstandings between manufacturers and purchasers, facilitating interchangeability and improvement of products, and assisting the purchaser in selecting and obtaining with minimum delay the proper product for his particular need. Existence of such Standards, Bulletins and other technical publications shall not in any respect preclude any member or nonmember of the Consumer Technology Association from manufacturing or selling products not conforming to such Standards, Bulletins or other technical publications, nor shall the existence of such Standards, Bulletins and other technical publications preclude their voluntary use by those other than Consumer Technology Association members, whether the standard is to be used either domestically or internationally.

Standards, Bulletins and other technical publications are adopted by the Consumer Technology Association in accordance with the American National Standards Institute (ANSI) patent policy. By such action, the Consumer Technology Association does not assume any liability to any patent owner, nor does it assume any obligation whatever to parties adopting the Standard, Bulletin or other technical publication.

This document does not purport to address all safety problems associated with its use or all applicable regulatory requirements. It is the responsibility of the user of this document to establish appropriate safety and health practices and to determine the applicability of regulatory limitations before its use.

(Formulated under the cognizance of the CTA **WAVE Project**; for information please see cta.tech/WAVE.)

Published by
CONSUMER TECHNOLOGY ASSOCIATION
Technology & Standards Department
www.cta.tech

Table of Contents

Figures

FOREWORD

This document was developed by the Web Application Video Ecosystem (WAVE) Project of the Consumer Technology Association. The WAVE Project is a broad industry initiative of content, technology, infrastructure and device companies, all working together towards commercial Internet video interoperability based on industry standards.

Web Application Video Ecosystem – DASH-HLS Interoperability Specification

2023 Edition

1 Scope

The scope of this specification is the encoding and packaging of segmented media in adaptive multimedia presentations conforming to the MPEG Common Media Application Format [CMAF] such that ISO MPEG Dynamic Adaptive Streaming over HTTP [DASH] and IETF HTTP Live Streaming [HLS] versions of those multimedia presentations are fully interoperable.

2 References

2.1 Normative References

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the dated edition applies. For undated references, the latest edition of the referenced document (including any amendments and corrigenda) applies.

[CMAF] ISO/IEC 23000-19:2020, *Information technology — Coding of audio-visual objects — Part 19: Common media application format (CMAF) for segmented media*.

[DASH] ISO/IEC 23009-1:2021, Information technology — Dynamic adaptive streaming over HTTP (DASH) – Part 1: Media presentation description and segment formats.

[HLS] Draft-pantos-hls-rfc8216bis "HTTP Live Streaming 2nd Edition", April 2020, https://tools.ietf.org/html/draft-pantos-hls-rfc8216bis

[RFC8673] HTTP Random Access and Live Content, November 2019, https://tools.ietf.org/html/rfc8673

[ID3-EMSG] Carriage of ID3 Timed Metadata in the Common Media Application Format (CMAF), April 2020, https://aomediacodec.github.io/id3-emsg/

[TV-ANYTIME] ETSI TS 102 822-3-1 V1.11.2 (2019-06), Broadcast and On-line Services: Search, select, and rightful use of content ("TV-Anytime"); Part 3: Metadata; Sub-part 1: Phase 1 – Metadata schemas,

https://www.etsi.org/deliver/etsi_ts/102800_102899/1028220301/01.11.02_60/ts_1028220301v011102p.pdf

2.2 Informative References

The following documents contain provisions that, through reference in this text, constitute informative provisions of this document. At the time of publication, the editions indicated were valid. All documents are subject to revision, and parties to agreements based on this document are encouraged to investigate the possibility of applying the most recent editions of the documents listed here.

[CENC] ISO/IEC 23001-7:2016, *Information technology – MPEG systems technologies – Part 7: Common encryption in ISO base media file format files*, https://www.iso.org/standard/68042.html

[HLS-CMAF] About the Common Media Application Format with HTTP Live Streaming, *https://developer.apple.com/documentation/http_live_streaming/about_the_common_media_application_format_with_http_live_streaming*

2.3 Document Notation and Conventions

The following terms are used to specify conformance elements of this specification. These are adopted from the ISO/IEC Directives, Part 2. For more information, please refer to those directives.

- SHALL and SHALL NOT indicate requirements strictly to be followed in order to conform to the document and from which no deviation is permitted.
- SHOULD and SHOULD NOT indicate that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited.
- MAY and NEED NOT indicate a course of action permissible within the limits of the document.

Terms defined to have a specific meaning within this specification will be capitalized, e.g. “Track”, and should be interpreted with their general meaning if not capitalized.

2.4 Known Issues

It is recognized that at the time of publication for this specification, new technologies are being discussed and added to the [CMAF], [DASH] and [HLS] specifications that define new functionality and create new points deserving of an interoperability description. Examples of these technologies are:

- [DASH] MPD Patching and [HLS] Partial Updates
- [DASH] Remote Resolution and [HLS] Interstitials

- [DASH] Multiple Base URLs and [HLS] Content Steering
- Usage of priority mechanisms for HTTP2/3 connections in [DASH] and [HLS] Low-Latency operation modes
- Usage of root/leaf hierarchies to rotate encryption keys
- [DASH] Annex I Flexible Insertion of URL Parameters and [HLS] EXT-X-DEFIN with QUERYPARAM attribute

Future editions of this specification will aim to include interoperability information about these new technologies while also expanding considerations for established use-cases.

2.5 Definitions

[CMAF], [DASH] and [HLS] all use different terminology when referring to conceptual pieces of an adaptive media presentation. In some cases, there is a direct mapping from one term to another, but in most cases the mapping is not quite exact. In this specification, terms defined in [CMAF] are used for common language and terms defined in both [DASH] and [HLS] are used for specification specific language.

As an informational reference to the reader, a collection of definitions from the various specifications is provided in the table below, relating them to their parallel terms in other specifications where applicable:

Term	Reference	Definition
Adaptation Set	DASH	(from [DASH]) A set of interchangeable encoded versions of one or several media content components. See also, Aligned CMAF Switching Set, CMAF Switching Set, and Multivariant Playlist (HLS).
Aggregating Response Transfer Method	N/A	A streaming data transfer mechanism in which data is sent as it becomes available and where the transfer continues until the object is complete. The receiver of the data may act upon any available portion of the data prior to the transfer being completed. HTTP/1.1 Chunked Transfer Encoding is a form of this method. Similar capabilities can be achieved through the frame mechanisms of HTTP2 and HTTP3.

Term	Reference	Definition
Aligned CMAF Switching Set	CMAF	(from [CMAF]) Set of CMAF Switching Sets, the CMAF Tracks of which all contain alternative encodings of the same source content in time-aligned CMAF Fragments, but all CMAF tracks do not conform to a single CMAF Switching Set. See also, Adaptation Set (DASH) and Multivariant Playlist (HLS).
Average Bitrate	N/A	The sum of bits observed in a defined time window divided by the duration of the defined time window.
CMAF Addressable Media Object	CMAF	(from [CMAF]) CMAF media object packaged for storage or delivery. Examples: CMAF Track File, CMAF Segment, and CMAF Chunk.
CMAF Chunk	CMAF	(from [CMAF]) CMAF media object that contains a consecutive subset of the media samples of a CMAF fragment, where only the first CMAF chunk of a CMAF fragment is constrained to be an adaptive switching point. See also, Delivery Unit Media Segment (DASH) and Partial Segment (HLS).
CMAF Fragment	CMAF	(from [CMAF]) Encoded ISOBMFF media segment conforming to CMAF constraints. See also, Delivery Unit Media Segment (DASH) and Partial Segment (HLS).
CMAF Header	CMAF	(from [CMAF]) Sequence of CMAF constrained ISOBMFF boxes that do not reference any media samples, but are associated with a CMAF Track and necessary for decoding the CMAF fragments. See also, Initialization Segment (DASH) and Media Initialization Section (HLS).

Term	Reference	Definition
CMAF Presentation	CMAF	(from [CMAF]) Set of one or more CMAF Selection Sets that can be simultaneously decoded to produce a multimedia user experience, potentially including synchronized audio, video, and subtitles. See also, Media Presentation (DASH) and Presentation (HLS).
CMAF Principal Header	CMAF	(from [CMAF]) A CMAF Header that includes sufficient information to initialize the media decoder and decryption for all CMAF Tracks of the CMAF Switching Set.
CMAF Segment	CMAF	(from [CMAF]) CMAF Addressable Media Object consisting of one or more consecutive CMAF Fragments from the same CMAF Track. See also, Media Segment (DASH) and Media Segment (HLS).
CMAF Selection Set	CMAF	(from [CMAF]) Set of one or more CMAF Switching Sets, where each CMAF Switching Set encodes an alternative aspect of the same presentation over the same time period, only one of which is intended to be played at a time, e.g. an alternative language or codec. See also, Group (DASH) and Rendition (HLS).
CMAF Switching Set	CMAF	(from [CMAF]) Set of one or more CMAF Tracks, where each track is an alternative encoding of the same source content, and are constrained to enable seamless track switching. See also, Adaptation Set (DASH) and Multivariant Playlist (HLS).
CMAF Track	CMAF	(from [CMAF]) Sequence of CMAF Fragments that are consecutive in presentation time, contain one media stream, conform to at least one structural CMAF brand, including an associated CMAF Header that can initialize playback. See also, Representation (DASH) and Variant Stream (HLS).

Term	Reference	Definition
CMAF Track File	CMAF	(from [CMAF]) One CMAF Track stored consecutively in a single ISOBMFF file with the earliest CMAF Fragment constrained to start at decode time zero. See also Self-Initializing Media Segment (DASH).
Delivery Unit Media Segment	DASH	A Media Segment containing one or more whole self-contained movie fragments. A whole, self-contained movie fragment is a movie fragment (`'moof'`) box and a media data (`'mdat'`) box that contains all the media samples that do not use external data references referenced by the track runs in the movie fragment box. See also CMAF Chunk, CMAF Fragment, and Partial Segment (HLS).
EXT-X-DISCONTINUITY	HLS	A tag which indicates a discontinuity between the Media Segment that follows it and the one that preceded it. See also, Period (DASH).
Group	DASH	A set of one or more Adaptation Sets where only one Representation from the entire group may be active at any given time. See also, CMAF Selection Set and Rendition (HLS).
Initialization Segment	DASH	(from [DASH]) Segment containing metadata that is necessary to present the media streams encapsulated in Media Segments. See also, CMAF Header and Media Initialization Section (HLS).
Multivariant Playlist	HLS	A Multivariant Playlist provides a set of Variant Streams, each of which describes a different version of the same content. Multivariant Playlist tags define the Variant Streams, Renditions and other global parameters of the presentation. This term was previously defined as Master Playlist and re-defined in [HLS]. See also Media Presentation Description (MPD) (DASH).
Media Content Component	DASH	(from [DASH]) A single continuous component of the media content with an assigned media component type.

Term	Reference	Definition
Media Initialization Section	HLS	Segment containing metadata that is necessary to present the media streams encapsulated in Media Segments. See also CMAF Header and Initialization Segment (DASH).
Media Playlist	HLS	A Media Playlist contains a list of Media Segments, which, when played sequentially, will play the multimedia presentation. See also Media Presentation Description (MPD) (DASH).
Media Presentation	DASH	(from [DASH]) A collection of data that establishes a bounded or unbounded presentation of media content. See also CMAF Presentation and Presentation (HLS).
Media Presentation Description (MPD)	DASH	An XML document containing information about Media Segments, different qualities of the media content. See also Multivariant Playlist (HLS), Media Playlist (HLS).
Media Segment	DASH	(from [DASH]) Segment that complies with media format in use and enables playback when combined with zero or more preceding segments, and an Initialization Segment (if any). See also CMAF Segment and Media Segment (HLS).
Media Segment	HLS	Segment containing an encoded bitstream of the media. See also CMAF Segment and Media Segment (DASH).
Partial Segment	HLS	A subset of a Media Segment that can be delivered independently to reduce publishing latency. See also CMAF Chunk, CMAF Fragment, and Delivery Unit Media Segment (DASH).
Period	DASH	(from [DASH]) Interval of the Media Presentation, where a contiguous sequence of all Periods constitutes the Media Presentation. See also EXT-X-DISCONTINUITY (HLS).

Term	Reference	Definition
Playlist	HLS	A generic term for either a Media Playlist or a Multivariant Playlist. A Playlist is a Media Playlist if all URI lines in the Playlist identify Media Segments. A Playlist is a Multivariant Playlist if all URI lines in the Playlist identify Media Playlists.
Presentation	HLS	A multimedia content experience encoded into one or more series of Media Segments for presentation to end clients. See also, CMAF Presentation and Media Presentation (DASH).
P95 RTT	N/A	The maximum Round-Trip Time to the server that 95% of the client's experience.
Rendition	HLS	Renditions are alternate versions of the content, such as audio produced in different languages or video recorded from different camera angles. See also Group (DASH) and CMAF Selection Set.
Representation	DASH	(from [DASH]) Collection and encapsulation of one or more media streams in a delivery format and associated with descriptive metadata. See also, Variant Stream (HLS).
Segment	DASH	(from [DASH]) Unit of data associated with an HTTP-URL and optionally a byte range that are specified by an MPD, or with a data URL. See also Media Segment (DASH), Initialization Segment (DASH), Media Initialization Section (HLS) and Media Segment (HLS).
Segment Index	DASH	(from [DASH]) Compact index of the time range to byte range mapping within a Media Segment (DASH) separately from the MPD.
Self-Initializing Media Segment	DASH	A single media segment which contains both initialization information as well as media data. See also CMAF Track File.
Subsegment	DASH	(from [DASH]) Unit within Media Segments (DASH) that is indexed by a Segment Index.

Term	Reference	Definition
Track Constraints	N/A	Media authoring constraints that are applied during the creation or reprocessing of a CMAF Track and associated structures to enable interoperability.
Variant Stream	HLS	A Variant Stream, or HLS Variant, includes a Media Playlist that specifies media encoded at a particular bit rate, in a particular format, and at a particular resolution for media containing video. It can also specify a set of Renditions. See also Representation (DASH).

2.6 Edition Changes

2.6.1 2021 Edition

This was the first publication of this specification that established the overall specification format and included the original interoperability use-cases:

- Basic On-Demand and Live Streaming
- Low Latency Live Streaming
- Encrypted Media Presentations
- Presentation Splicing

2.6.2 2023 Edition

This was the second publication of this specification that included the additional use-cases:

- Carriage of Timed Event Data (In-Stream Based and Manifest Based)
- Rotation of Encryption Keys
- Carriage of Track Roles

This version provided the following corrections:

- Throughout – Renamed “Master Playlist” to “Multivariant Playlist” to align with HLS terminology change.
- 5.4.1.2 – Added missing final paragraph that described the generation of individual HLS Media Playlists from discovered track alignments.

This version did not modify any statements of explicit conformance SHALL or MAY from the previous.

3 Defining DASH-HLS Interoperability using CMAF (Informative)

In the context of this document, we consider an architecture in which a service provider application performs playout of one or more ISO MPEG Common Media Application Format [CMAF] presentations. The player addresses the CMAF Presentations on a content delivery network utilizing the information provided by a manifest and performs playout utilizing a reference playback platform. An overview of this architecture with identified interfaces is documented in Figure 3.1 – Considered Deployment Architecture.

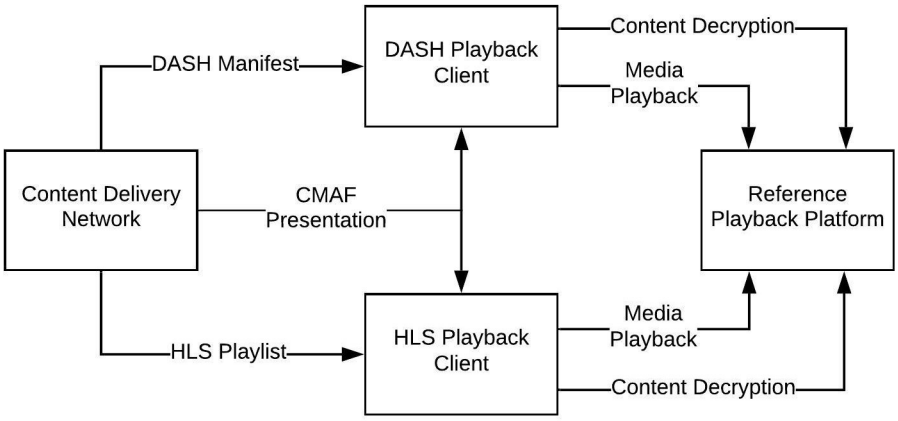

Figure 3.1 – Considered Deployment Architecture

This document will focus on the form of the CMAF Presentations through a variety of use cases and how these presentations can be equivalently described through the two popular manifest formats in use today – HTTP Live Streaming [HLS] and Dynamic Adaptive Streaming over HTTP [DASH]. While [DASH] and [HLS] define compatibility profiles for [CMAF], their differences in data model impose different constraints on CMAF Presentations, which require additional rules to ensure full interoperability between the formats.

The remainder of this section will provide an informational overview of the CMAF data model and how the CMAF compatibility profiles that have been defined by [DASH] and [HLS] map these concepts to their respective formats.

3.1 The CMAF Media Object Model

While this document assumes reader familiarity with the [CMAF] specification, a brief informational overview of concepts is provided to help contextualize the data model mappings made by the [DASH] and [HLS] constraint profiles.

The CMAF media object model is made up of three media object types:

- *Encoded media objects* are the base objects that all others derive from. They are specified as CMAF Headers and CMAF Fragments that together form CMAF Tracks.

- *Addressable media objects* are derived from CMAF Tracks and CMAF Fragments and can be used for storage and delivery of a CMAF Presentation. They are specified as CMAF Headers, CMAF Track Files, CMAF Segments, and CMAF Chunks.
- *Logical media objects* are defined by the logical grouping of encoded media objects based on content and encoding constraints. They are specified as CMAF Switching Sets, Aligned CMAF Switching Sets, CMAF Selection Sets, and CMAF Presentations.

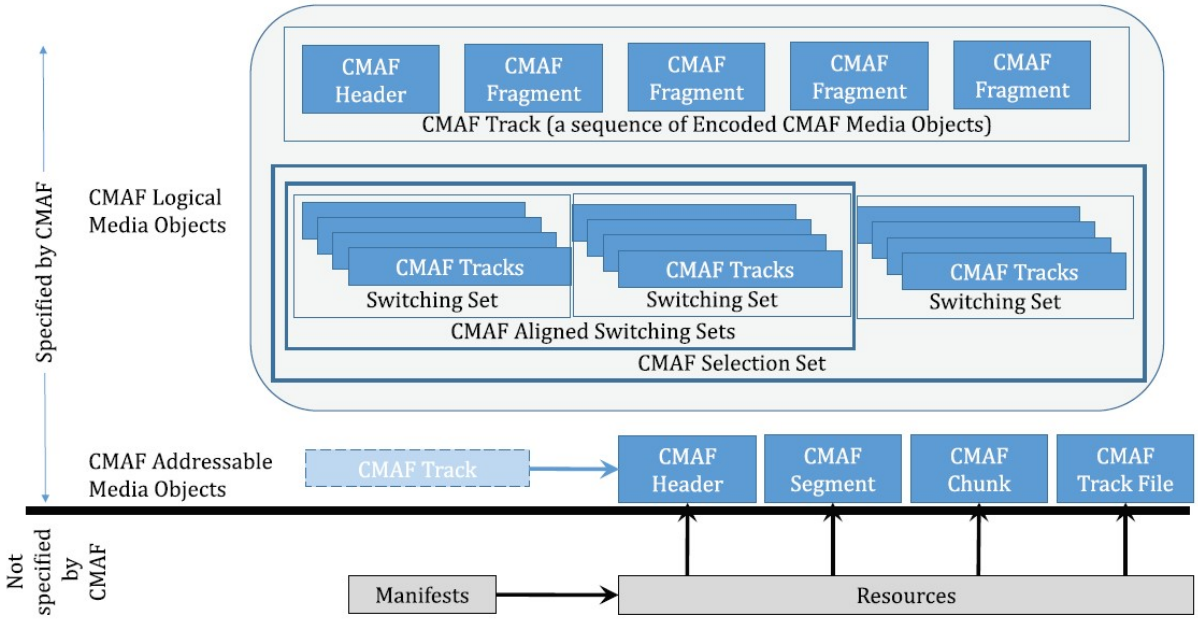

Figure 3.2 – CMAF Object Overview from [CMAF] Specification

The [CMAF] specification defines a conceptual manifest which contains references to the addressable media objects and describes the logical media objects. However, [CMAF] explicitly defers the definition of these manifests to the existing adaptive streaming standards, such as [DASH] and [HLS].

3.2 The DASH CMAF Profile

For the [DASH] specification, the CMAF conceptual manifest role is fulfilled by a single document called a Media Presentation Description (MPD) which contains various DASH structures for describing a streamable presentation.

The DASH 5th Edition introduces a DASH Profile for CMAF Content, `"urn:mpeg:dash:profile:cmaf:2019"`, that defines a normative mapping of CMAF structures to DASH structures. For each level within the MPD structure, the profile declares the CMAF constraints that apply, additional constraints on the CMAF structures for conformance with DASH structures, and the appropriate signaling of CMAF structures in the DASH structures.

The following is an informational summary of mappings and constraints:

- In the context of [DASH], CMAF Segments are the only addressable units – i.e., units with assigned URLs.

    - CMAF Fragments and CMAF Chunks may be contained in DASH Media Segments.
- CMAF Tracks are equivalent to DASH Representations where:
    - Each CMAF Segment in the CMAF Track is a DASH Segment in the DASH Representation.
    - The CMAF Header in the CMAF Track is equivalent to the DASH Initialization Segment.
    - CMAF Track Files are equivalent to DASH Indexed Self-Initializing Media Segments.
- CMAF Switching Sets are equivalent to DASH Adaptation Sets where:
    - The CMAF Principal Header information is used to set the appropriate values for Adaptation Set attributes `@contentType`, `@mimeType` and `@codecs`.
    - Each CMAF Track in the CMAF Switching Set is a Representation in the Adaptation Set.
    - The `@bitstreamSwitching` attribute of the Adaptation Set is set to `true` if the switching set further conforms to CMAF Switching Set single initialization constraints.
- Aligned CMAF Switching Sets are equivalent to DASH Adaptation Sets where:
    - There is one DASH Adaptation Set for each CMAF Switching Set in the aligned set.
    - All Adaptation Sets of the aligned set share the same integer value for either the `@segmentAlignment` or `@subsegmentAlignment` attribute.
- CMAF Selection Sets are equivalent to DASH Groups.
- CMAF Presentations are equivalent to DASH Periods where:
    - The `@presentationTimeOffset` in DASH is equal to the first CMAF presentation time.
    - The duration of the Period is at most the duration of the contained CMAF Presentation.
- CMAF Presentations may be split across multiple Periods. When this occurs, continuity signaling for Adaptation Sets should be used.
- A DASH Manifest may contain one or more CMAF Presentations as a series of Periods.

3.3 The HLS CMAF Profile

For the [HLS] specification, the CMAF conceptual manifest role is split between the Multivariant Playlist and Media Playlist. Conceptually, the responsibility split that HLS defines associates all logical and addressable media objects with Multivariant Playlist concepts.

Guidelines for the mapping of [CMAF] to [HLS] are provided in the “About the Common Media Application Format with HTTP Live Streaming” [HLS-CMAF] document which details the split of responsibilities between the Multivariant and Media Playlists. The Multivariant Playlist defines

the tiers of the primary media type of the presentation, distinguished by bitrate, codecs, resolutions, and other attributes. The primary tiers are linked to additional Renditions of secondary media types that are meant to be selected for playback alongside the main tiers. Both the main tiers and additional Rendition descriptions provide references to Media Playlists that contain the associated addressable media objects.

An informational summary of mappings and constraints is provided:

- CMAF Tracks are equivalent to HLS Media Playlists where:
    - CMAF Segments are equivalent to HLS Segments.
    - CMAF Headers are referenced by `EXT-X-MAP` tags associated with HLS Segments.
    - CMAF Fragments and CMAF Chunks are equivalent to HLS Parts.
- CMAF Switching Sets are contained in an HLS Multivariant Playlist where:
    - Each CMAF Track in Switching Sets for primary media has an `EXT-X-STREAM-INF` tag referencing the HLS Media Playlist defining the track.
    - Each CMAF Track in Switching Sets for secondary media has an `EXT-X-MEDIA` tag referencing the HLS Media Playlist defining the track.
    - GROUP-ID attributes are used to relate `EXT-X-STREAM-INF` and `EXT-X-MEDIA` tags containing tracks that can be selected for playback together.
- Aligned CMAF Switching Sets are not differentiated by HLS and have no explicit signaling, instead these sets are contained in an HLS Multivariant Playlist in the same way as CMAF Switching Sets.
- CMAF Selection Sets are contained in an HLS Multivariant Playlist where:
    - CMAF Tracks of Switching Sets within the Selection Set that describe the primary variants of each media type are denoted as described for CMAF Switching Sets.
    - CMAF Tracks of Switching Sets within the Selection Sets that describe alternate variants of media are denoted with an `EXT-X-MEDIA` tag referencing the HLS Media Playlist defining the track.
    - Tracks within a Selection Set that define alternate variants of the same media type have a `GROUP-ID` attribute with equivalent value.

4 DASH and HLS Interoperability Use Cases

This section describes a variety of streaming use-cases for CMAF Presentation(s) and provides the following information for each one:

- A general description of the use case and the general form of the CMAF Presentation(s) that are utilized in it.
- What constraints are needed when authoring the CMAF Presentation(s) to produce media objects that can be addressed in both the [DASH] and [HLS] manifest formats.

- What Track Constraints are needed on existing [DASH] CMAF Profile content to produce content interoperable with [HLS].
- What Track Constraints are needed on existing [HLS] CMAF Profile content to produce content interoperable with [DASH].

This section does not specify the Manifest or Playlist level specifics. It details the CMAF Track constraints such that the same media object could be used for [DASH] or [HLS] applications.

A basic use case is described initially with subsequent use cases adding in additional constraints to enable added features.

4.1 Basic On-Demand and Live Streaming

4.1.1 Use-Case Overview

For this use-case:

- Content is either:
  - Prepared ahead of time and made available on demand to clients in its entirety (On-Demand).
  - Prepared and made available to clients in real-time (Live Streaming).
- The content is a single CMAF presentation that may have multiple CMAF conformant video, audio, and text components that make up CMAF defined media objects.
- The content is intended to be played with no auxiliary content before, during, or after the main presentation.
- The media components are provided without the use of encryption.

4.1.2 Constraints on CMAF Authoring for Manifest Interoperability

When authoring [CMAF] content for this use-case the following constraints SHALL apply:

- Text components SHALL be packaged in of the following [CMAF] compliant formats:
  - IMSC1 Text Track
  - IMSC1.1 Text Track
  - WebVTT Track

  Note: [CMAF] provides both Text Track and Image Track types to represent text components. Here we are stating that only Text Track types are interoperable.

  Note: Generally, IMSC1 (1.0 and 1.0.1) Text Tracks will provide a higher level of interoperability as of the time of this specification's publication.
- CMAF Track Files used as a CMAF Addressable Media Object (i.e., those that are delivered to the client) SHALL contain a single SegmentIndexBox (`'sidx'`) following the CMAF Header and preceding any CMAF Fragments.

Note: The requirement of a SegmentIndexBox being generated for Track Files is due to [DASH] byte-range addressing requiring the segment index to properly address individual DASH Subsegments. This box is not required by or used in [HLS], nor if CMAF Track Files are only used internally to generate CMAF Segments for delivery.

To maximize general device compatibility when authoring [CMAF] content for this use-case the following constraints SHOULD apply:

- For presentations presented in an on-demand environment:
    - The Average Bitrate of a CMAF Fragment within a CMAF Track SHOULD be within 10% of the Average Bitrate calculated over the full duration of the Track.
- For presentations presented in a live streaming environment:
    - The Average Bitrate of any CMAF Fragment within a 1-hour slice of a CMAF Track or the maximum duration of the Track, whichever is smaller, SHOULD be no more than 110% of the Average Bitrate calculated over the same time slice.

4.1.3 Constraints on DASH Content for HLS Interoperability

Existing [DASH] CMAF Profile conforming content MAY be used with [HLS] following these changes to the content:

- Any text components not available in a CMAF Text Track type as outlined in section 4.1.2 SHALL have a conforming track type produced that SHALL be used in HLS Playlists.

4.1.4 Constraints on HLS Content for DASH Interoperability

Existing [HLS] CMAF Profile conforming content MAY be used with [DASH] following these changes to the content:

- CMAF Track Files without a SegmentIndexBox (`'sidx'`) SHALL have a SegmentIndexBox generated and inserted following the CMAF Header and preceding any CMAF Fragments.

4.2 Low Latency Live Streaming

4.2.1 Use-Case Overview

For this use-case we consider content that matches the Basic On-Demand and Live Streaming use-case and is used with these additional considerations:

- Content is prepared and made available to clients in real-time.
- The content is intended to be presented to end users with an end-to-end system latency that is less than three times the average duration of a CMAF segment.
- Partially generated CMAF Segments are intended to be accessed by clients before they are fully produced.

4.2.2 Constraints on CMAF Authoring for Manifest Interoperability

When authoring [CMAF] content for this use-case the following constraints SHALL apply:

- Each CMAF Segment SHALL be made available as a single addressable object containing a series of one or more CMAF Chunks.

  Note: Producing a single addressable object per CMAF Segment and utilizing byte range addressing of substructures where necessary, produces the greatest level of operational efficiency as clients playing in both regular and low latency playback modes will utilize the same segments in network caches.

- All `emsg` boxes inserted into the CMAF Segment after the start of the first CMAF Chunk SHALL be repeated before the first Chunk of the next Segment.

  Note: Some playback environments may not support parsing of `emsg` boxes that appear within a Segment, they can only be discovered by these players by repeating at the start of the next segment, but at the cost of message latency.

To maximize general device compatibility when authoring [CMAF] content for this use-case the following constraints SHOULD apply:

- CMAF Chunks SHOULD have a minimum target duration of approximately 500ms or three times the P95 RTT experienced by clients, whichever is greater.

  Note: Compatibility with LL-HLS authoring guidelines can be maximized by using a Chunk target duration of one second.

In addition to general authoring requirements, systems delivering CMAF Segments that will be used in a low-latency environment have operational constraints placed on them to fulfill the serving requirements of both manifest formats. Delivery systems SHALL conform to the following constraints:

- CMAF Chunks for an addressable segment:
  - SHALL be byte-range addressable within their containing CMAF Segments prior to the segment being fully produced,
  - SHALL be transmitted via an Aggregating Response Transfer Method, and
  - SHALL transmit Chunks only after they have been fully produced.

    Note: This requirement is to ensure Chunks are transmitted at line speed as required by the [HLS] specification.

When operating in a distribution environment where clients might request non-zero byte-start positions within a segment being actively produced, delivery systems SHALL conform to the following:

- Systems SHALL understand and abide by the conventions set forth in [RFC8673].
- Systems SHALL use a `last-byte-pos` of $2^{53}-1$ (`9007199254740991`) as an indication that the client is requesting an aggregated response.

  Note: This behavior may be used to enable partial download of a multi-GOP segment or perform mid-segment switching for HLS described parts, but in all cases the client is

expected to use these semantics to request an aggregated response from a non-zero start byte instead of an open-ended range request.

- Systems SHALL make objects available via both HTTP1.1 and HTTP2.
- Systems MAY make objects available via HTTP3.

  Note: The requirement of HTTP1.1 and HTTP2 is due to [DASH] and [HLS] specifying these transfer protocols explicitly for Low-Latency operation. Generally, the lowest specification compliant transfer protocol can be used but higher efficiency operation can be achieved using new transfer protocols.

4.2.3 Constraints on DASH Content for HLS Interoperability

Existing [DASH] CMAF Profile content MAY be used with [HLS] with no further changes to the content.

The [DASH] delivery systems SHALL provide the following additional functionality:

- CMAF Chunks for an addressable segment SHALL be byte-range addressable within the object.

The DASH delivery systems SHALL provide the following additional functionality:

- Systems SHALL understand and abide by the conventions set forth in [RFC8673].
- Systems SHALL use a `last-byte-pos` of $2^{53}-1$ (`9007199254740991`) as an indication that the client is requesting an aggregated response.

4.2.4 Constraints on HLS Content for DASH Interoperability

Existing [HLS] CMAF Profile content MAY be used with [DASH] following these changes to the content:

- CMAF Chunks stored as independent objects SHALL be combined to create a single addressable CMAF segment.

Additionally, existing [HLS] delivery systems SHALL provide the following functionality:

- CMAF Segments requested during active processing SHALL be transmitted via an Aggregating Response Transfer Method.

4.3 Encrypted Media Presentations

4.3.1 Use-Case Overview

For this use-case we consider content that matches any other streaming use case and is used with these additional constraints:

- Content provided to clients contains video or audio components that make use of encryption.

4.3.2 Constraints on CMAF Authoring for Manifest Interoperability

When authoring [CMAF] content for this use-case the following constraints SHALL apply:

- The Common Encryption `cbcs` scheme ([CENC] 10.4) SHALL be used for encryption with the following constraints:
    - Constant 16-byte Initialization Vectors SHALL be used.
    - When not stated by a codec media profile, the encrypt:skip pattern SHALL be set as follows:
        - Video components SHALL use a 1:9 pattern.
        - Audio components SHALL use a 10:0 pattern.
- Sample auxiliary information, if present, SHALL be addressed by a CMAF conforming SampleAuxiliaryInformationOffsetsBox (`'saio'`).
- Any individual CMAF Segment SHALL have a single encryption key and Initialization Vector.

  Note: [CMAF] does allow changes of key at CMAF Fragment boundaries, but for maximum interoperability this is restricted to the segment level.

For any service provider targeting a sufficiently diverse device ecosystem, there may be devices that are incapable of consuming content encrypted with the `cbcs` scheme. For these devices, the following constraints MAY be utilized:

- For every `cbcs` encrypted component, an alternative component MAY be produced following the previous constraints with the following modification:
    - The Common Encryption `cenc` scheme ([CENC] 10.1) SHALL be used for encryption.
    - Initialization Vectors SHALL be used as stated in the CMAF specification, section 8.2.3.1
- The `cenc` encrypted components SHALL only be addressed by a [DASH] manifest.

4.3.3 Constraints on DASH Content for HLS Interoperability

Existing [DASH] CMAF profile conforming content MAY be used with [HLS] following these changes to the content:

- For any components encrypted with the `cenc` scheme:
    - An equivalent `cbcs` encrypted component SHALL be produced using the common constraints outlined above.
- Any components encrypted with the Common Encryption `cbcs` scheme that do not conform to the common constraints outlined above SHALL be re-encrypted to align with the provided constraints.

  Note: Re-encryption is preferred over duplication in this case as all known client implementations of the `cbcs` scheme can handle the constrained definition of `cbcs` described, but not all client `cbcs` implementations can handle all optional behavior of the `cbcs` scheme.

4.3.4 Constraints on HLS Content for DASH Interoperability

Existing [HLS] CMAF Profile conforming content MAY be directly used with [DASH].

For any service provider targeting a sufficiently diverse device ecosystem, there may be devices that are incapable of consuming content encrypted with the `cbcs` scheme. See 4.3.2 for constraints to increase device reach.

4.4 Presentation Splicing

4.4.1 Use-Case Overview

For this use-case we consider presentations that match any other use case and are used with these additional considerations:

- Multiple CMAF Presentations are intended to be presented together as one presentation experience.
- Presentations are combined in either:
    - Linear timeline concatenations, or
    - Mid-presentation splice inserts
- Availability of track features such as languages and roles may vary across presentations (i.e., missing dubs / translations in advertisements).
- Encoded versions of tracks may vary across presentations (e.g., codec changes, bitrate ladder differences, encryption usage).

4.4.2 Constraints on CMAF Authoring for Manifest Interoperability

When authoring [CMAF] content for this use-case the following constraints SHALL apply:

- CMAF Fragment boundaries SHALL be created at all splice points.
- CMAF Fragments of different media types NEED NOT have exactly equivalent durations but SHALL be within one ISOBMFF sample duration.

When performing cross-presentation playout, the quality of the cross-boundary playout experience is causally related to the robustness of the playback platform implementation. To maximize general device ability to provide seamless cross-boundary playout for this use-case the following constraints SHOULD apply:

- Every media type in every considered Presentation SHOULD have at least one CMAF Switching Set with equivalent encoding constraints.
- Each CMAF Track in a CMAF Switching Set in a Presentation SHOULD have at least one Track in an equivalent Switching Set in all other Presentations with:
    - An equivalent target duration, and
    - A bit rate computation following the constraints outlined in 4.1.2.

4.4.3 Constraints on DASH Content for HLS Interoperability

Existing [DASH] CMAF profile-conforming content used in a presentation splicing scenario in [DASH] MAY be used with [HLS] following these changes to the content:

- CMAF Fragment boundaries SHALL be created at all splice points.
- CMAF Fragments of different media types NEED NOT have exactly equivalent durations but SHALL be within one ISOBMFF sample duration.

4.4.4 Constraints on HLS Content for DASH Interoperability

Existing [HLS] CMAF profile-conforming content used in a presentation splicing scenario in [HLS] MAY be directly used with [DASH].

4.5 Carriage of Timed Event Data

4.5.1 Use-Case Overview

For this use-case we consider content that matches any other streaming use case and is used with these additional considerations:

- Auxiliary event data related to content is prepared for presentation to clients.
- The auxiliary event data is timed to content and intended to be surfaced relative to the content playout timeline.
- The auxiliary event data may be carried in either the CMAF Segments directly or the content manifest.

4.5.2 Constraints on CMAF Authoring for Manifest Interoperability

When authoring [CMAF] content for this use-case, the following constraints SHALL apply:

- All `emsg` boxes inserted into the CMAF Segment after the start of the first CMAF Chunk SHALL be repeated before the first Chunk of the next Segment.

Note: Some playback environments may not support parsing of `emsg` boxes that appear within a Segment; they can only be discovered by these players by repeating at the start of the next segment, but at the cost of message latency.

4.5.3 Constraints on DASH Content for HLS Interoperability

Existing [DASH] CMAF profile conforming content MAY be used with [HLS] with no further changes to the content.

4.5.4 Constraints on HLS Content for DASH Interoperability

Existing [HLS] CMAF Profile conforming content MAY be used with [DASH] following these changes to the content:

- Any CMAF Fragments containing ID3 tags SHALL have the ID3 tags converted to `emsg` boxes as defined by the [ID3-EMSG] carriage specification.

4.6 Rotation of Encryption Keys

4.6.1 Use-Case Overview

For this use-case we consider content that matches the Encrypted Media Presentations uses case and is used with these additional considerations:

- The encryption key used for encrypting content provided to clients rotates over the course of a presentation.

4.6.2 Constraints on CMAF Authoring for Manifest Interoperability

When authoring [CMAF] content for this use-case the following constraints SHALL apply:

- Each CMAF Segment whose encryption key differs from the `default_KID` declared in the original CMAF Header Track Encryption Box (`'tenc'`) box SHALL provide:
    - A Sample Group Description Box (`'sgpd'`) containing a CencSampleEncryptionInformationGroupEntry (`'seig'`) sample group description structure that describes the new encryption key information.
    - A Sample To Group Box (`'sbgp'`) that maps the samples of the fragment to the CencSampleEncryptionInformationGroupEntry (`'seig'`) containing the new encryption key information.
    - If also present in the original CMAF Header, Protection System Specific Header Boxes (`'pssh'`) containing the new encryption key information for each expected DRM system.

  Note: The use of Sample Group Description Boxes for key rotation has been identified by industry leaders as viable across the ecosystem for all major devices of the 2017 or later model years.

To maximize general device compatibility when authoring [CMAF] content for this use-case, the following constraints SHOULD apply:

- For each new encryption key, a new CMAF Header SHALL be created with the `default_KID` set to the new encryption key in the Track Encryption Box (`'tenc'`).

This maximized approach is generally only required for distributors looking to support some pre-2017 devices and other smaller footprint ecosystems but can be generally used in conjunction with the Sample Group Description Box recommendation above, as the Sample Groups are ignored by these older devices. This allows for a single CMAF resource creation to occur and be shared across the full deployment with differentiated manifest signaling.

Note: A CMAF Track may only have one CMAF Header in a CMAF Presentation. This means that the creation of a new CMAF Header causes the creation of an entirely new CMAF Presentation that starts at the time point represented by the new CMAF Header. This is important, as the semantics for representing multiple disjoint CMAF Presentations to an ecosystem can be directly re-used in this case.

4.6.3 Constraints on DASH Content for HLS Interoperability

Existing [DASH] CMAF Profile conforming content MAY be directly used with [HLS].

4.6.4 Constraints on HLS Content for DASH Interoperability

Existing [HLS] CMAF Profile conforming content MAY be directly used with [DASH].

4.7 Carriage of Track Roles

4.7.1 Use-Case Overview

For this use-case we consider content that matches any other use-case and is used with these additional considerations:

- The content tracks have defined roles that should be presented to clients.

4.7.2 Constraints on CMAF Authoring for Manifest Interoperability

When authoring [CMAF] content for this use-case, the following constraints SHALL apply:

- Track roles SHALL be stored in one or more KindBox (‘`kind`’) within the UserDataBox (‘`udta`’) of the TrackBox (‘`trak`’) in the CMAF Header.
- Track roles SHALL be represented by the DASH Role scheme ([DASH] 5.8.5.5) when possible.

  Note: When multiple role values of a scheme are required to fully describe a track’s role, a KindBox is inserted for each role value. This means there may be multiple boxes for a kind scheme as well as multiple schemes.

4.7.3 Constraints on DASH Content for HLS Interoperability

Existing [DASH] CMAF Profile conforming content MAY be directly used with [HLS].

4.7.4 Constraints on HLS Content for DASH Interoperability

Existing [HLS] CMAF Profile conforming content MAY be directly used with [DASH].

5 DASH-HLS Presentation Conversion Details

This section describes how to convert between the descriptive structures of [DASH] and [HLS] manifests with the goal of enabling the generation of one format from the other. Where one format lacks the descriptive information necessary to perform the conversion, care will be taken to detail alternative ways to derive the necessary information.

The structure conversion descriptions are grouped by the same streaming use-cases defined as part of Section 4. The use-case overviews established in that section for each use-case should be considered equivalent for this section.

As there are numerous structures within each format, this section addresses only a subset of the applicable structures set per use-case. Future editions of the specification will add additional structure descriptions to further describe conversions for each use-case.

5.1 Basic On-Demand and Live Streaming

This section focuses on basic signaling components of [DASH] and [HLS] manifests that can be found in both on-demand and live streaming use-cases as defined in Section 4.1. Due to the type of CMAF presentations in each use-case, not all sections apply to both, notably:

- 5.1.1 CMAF Track File Addressing applies only to on-demand use-cases.
- 5.1.2 Segmented CMAF Track Addressing applies to both on-demand and live use-cases.

5.1.1 CMAF Track File Addressing

5.1.1.1 DASH / HLS Signaling Overview

CMAF Track Files are addressed by client players utilizing byte-range addressing. There are three types of ranges important to enabling utilization of the track:

- Initialization Data Range: This range addresses the CMAF Header within the Track File.
- Index Range: This range addresses the ISOBMFF `sidx` box within the Track File.
- Segment Range(s): These ranges describe the individual CMAF Segments within the CMAF Track File.

Within [DASH] manifests, a combination of `BaseURL` and `SegmentBase` elements provide the information necessary for clients to make byte-range requests. The `BaseURL` element provides the HTTP resource location and the `SegmentBase` element explicitly describes the initialization data and index ranges of the CMAF Track File. With this signaling, the CMAF Segment ranges are not directly described, instead the player utilizes the index range to download the `sidx` box which contains the byte information necessary to compute all CMAF Segment ranges within the CMAF Track File. For further information refer to [DASH] 5.3.9.2 Segment base information.

```
<Representation id="foo" mimeType="video/mp4" ...>
    <BaseURL>foo-video.mp4</BaseURL>
    <SegmentBase indexRange="a-b" indexRangeExact="true">
        <Initialization range="0-a" />
    </SegmentBase>
</Representation>
```

Figure 5.1 – DASH Representation with SegmentBase Example

Within [HLS] Media Playlists, `EXT-X-MAP` and `EXT-X-BYTERANGE` tags are used with Media Segment entries to provide the information necessary for clients to make byte-range requests. The Media Segment entries provide the HTTP resource location for each Media Segment while the `EXT-X-BYTERANGE` preceding the entry describes the byte offset and byte length for the Media Segment within the CMAF Track File. The `EXT-X-MAP` tag, preceding a set of Media Segment entries, describes the HTTP resource location and the initialization data range for the CMAF Header segment within the Track File. HLS Media Playlists provide no explicit description of the Track File index range.

```
#EXTM3U
#EXT-X-TARGETDURATION:8
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-MAP:URI="foo-video.mp4",BYTERANGE="0-a"
#EXTINF:8.008,
#EXT-X-BYTERANGE:<length>@<offset>
foo-video.mp4
#EXTINF:8.008,
#EXT-X-BYTERANGE:<length>@<offset>
foo-video.mp4
...
```

Figure 5.2 – HLS Media Playlist with Byte Ranges

5.1.1.2 Derivation of HLS Signaling from DASH Signaling

When producing [HLS] Media Playlists from [DASH] Manifests:

- The HTTP resource location provided by the `BaseURL` element is directly used for both Media Segment entry and `EXT-X-MAP` tag generation.
- The `SegmentBase.Initialization@range` attribute value is directly used in the `EXT-X-MAP` tag `BYTERANGE` attribute.
- Every Media Segment is explicitly listed in order in the HLS Media Playlist, as the DASH manifest does not contain this data, it must be derived by acquiring the `sidx` box utilizing the `BaseURL` and `SegmentBase@indexRange` in the manifest. For every entry in the `sidx` box, create a Media Segment entry with at least:
    - An `EXTINF` tag set to the `sidx` described Media Segment duration.
    - An `EXT-X-BYTERANGE` tag set to the `sidx` described Media Segment byte length and a byte offset equal to the sum of the byte lengths in preceding `sidx` entries.

5.1.1.3 Derivation of DASH Signaling from HLS Signaling

When producing [DASH] Content from [HLS] Content:

- The HTTP resource location provided by all Media Segments should be equivalent and is directly used for the `BaseURL` element resource location.
- The `EXT-X-MAP` tag `BYTERANGE` attribute value is directly used in the `SegmentBase.Initialization@rang`e attribute
- As the index range is not explicitly addressed in HLS content, an alternative `SegmentBase` signaling is utilized to signal that the player needs to download and scan a range of bytes for the `sidx` box:
    - `SegmentBase@indexRange` is set to a range starting at 0 and continuing to the start of the first Media Segment.
    - `SegmentBase@indexRangeExact` is set to `false`.

Note: This utilizes the CMAF requirement that the `sidx` box occurs before any CMAF Fragments for the track file to be a properly conforming CMAF Track File.

5.1.2 Segmented CMAF Track Addressing

5.1.2.1 DASH / HLS Signaling Overview

Segmented CMAF Tracks are addressed by client players utilizing explicit URIs for the CMAF Header and CMAF Segments. As every URI is explicitly described, a service would not technically need to provide related URIs between Segments; however, interoperable operation requires that URIs are sequentially derivable with respect to sequence number or presentation time.

Within [DASH] manifests, the `SegmentTemplate` and `SegmentTimeline` elements provide the explicit URI information to client players. The [DASH] format supports two forms of templated URI specification:

- Number based:
    - In this mode of operation, the client substitutes a running sequential number into a provided template to derive explicit URIs for Segments.
    - Individual segment durations may be signaled with a `SegmentTimeline` structure, but it is not required, as the `SegmentTemplate@duration` attribute provides an approximate duration that can be utilized for timeline management.

```
<Representation id="foo" mimeType="video/mp4" ...>
    <SegmentTemplate
        timescale="1000000"
        duration="8008000"
        startNumber="1"
        initialization="foo-video.init.mp4"
        media="foo-video.$Number$.m4s" />
</Representation>
```

Figure 5.3 – DASH SegmentTemplate Example

- Time Based:
    - In this mode of operation, the client substitutes the presentation start time of the desired Segment into a provided template to derive an explicit URI.
    - Individual segment start times are signaled with a `SegmentTimeline` structure.

```
<Representation id="foo" mimeType="video/mp4" ...>
    <SegmentTemplate
        timescale="1000000"
        initialization="foo-video.init.mp4"
        media="foo-video.$Time$.m4s">
        <SegmentTimeline>
            <S t="0" d="8008000" r="100" />
        </SegmentTimeline>
    </SegmentTemplate>
```

```
</Representation>
```

Figure 5.4 – DASH SegmentTemplate with SegmentTimeline Example

The `SegmentTemplate@initialization` and `SegmentTemplate@media` attributes provide the URI templates for the CMAF Header and CMAF Segments, respectively. For further details refer to the [DASH] 5.3.9.4 Segment Template.

Within the [HLS] Media Playlists, Media Segment entries directly provide the explicit URIs for each CMAF segment and the `EXT-X-MAP` tag's URI attribute provides the explicit URI for the CMAF Header.

```
#EXTM3U
#EXT-X-TARGETDURATION:8
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-MAP:URI="foo-video.init.mp4"
#EXTINF:8.008,
foo-video.1.mp4
#EXTINF:8.008,
foo-video.2.mp4
...
```

Figure 5.5 – HLS Media Playlist with Explicit Number-Based Media URIs

```
#EXTM3U
#EXT-X-TARGETDURATION:8
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-MAP:URI="foo-video.init.mp4"
#EXTINF:8.008,
foo-video.0.mp4
#EXTINF:8.008,
foo-video.8008000.mp4
...
```

Figure 5.6 – HLS Media Playlist with Explicit Time-Based Media URIs

5.1.2.2 Derivation of HLS Signaling from DASH Signaling

When producing [HLS] Media Playlist from [DASH] Manifests:

- Media Segment entries are derived from the template evaluation of the `SegmentTemplate` and the appropriately signaled start and end boundaries of the containing Period.
- The duration of the Media Segment entries signaled by their preceding `EXTINF` tag should be either:
    - The `S@d` attribute value of the related `S` element in the source `SegmentTimeline` if one is present.

- The `SegmentTemplate@duration` attribute value if explicit per segment durations are not signaled.

  Note: In either case the duration value is divided by the `SegmentTemplate@timescale` value to convert from ticks per second to seconds.

  Note: DASH signaling permits up to 50% variance of segment duration when `SegmentTemplate@duration` is utilized while HLS signaling requires duration to be accurate enough to avoid perceptible errors when segment durations are accumulated. In practice, if segment durations will vary frequently it is better to produce [DASH] with `SegmentTimeline` addressing and derive [HLS] from the explicit timing.

5.1.2.3 Derivation of DASH Signaling from HLS Signaling

When producing [DASH] manifests from [HLS] Playlists:

- A `SegmentTemplate` element MAY be created per Representation where:
  - The `SegmentTemplate@initialization` attribute is set to the URI attribute of the `EXT-X-MAP` tag in the source HLS Media Playlist.
  - The `SegmentTemplate@timescale` attribute is set to the timescale present in the CMAF Header.
  - The `SegmentTemplate@media` attribute is set to a templated URI derived from analysis of the Media Segment entries utilizing either a `$Number$` or `$Time$` signaling semantics.
  - Where accurate timeline signaling is required or desired, and segment durations are not constant, a `SegmentTimeline` element is derived by creating an `S` element for each Media Segment entry signaling `S@t` and `S@d` as appropriate.
  - Where accurate timeline signaling is not required or desired, or the segment duration is constant, `SegmentTemplate@duration` attribute is set to the value computed by averaging the values of the `EXTINF` tags of the source HLS Media Playlist.

  Note: A sufficiently advanced conversion may be able to heuristically discover initialization and media templates containing `$RepresentationID$` signals allowing for a single `SegmentTemplate` entry to be generated per `AdaptationSet` rather than per `Representation`. As all Representations are CMAF Tracks in the same CMAF Switching Set their individual `SegmentTimeline` descriptions should always be equivalent.

  Note: Attributes representing media time of the `SegmentTemplate`, `SegmentTimeline`, and `S` elements have units of ticks per second. Any derived value for an attribute in seconds must be multiplied by the value of the `SegmentTemplate@timescale` attribute to achieve the final attribute value.

5.2 Low Latency Live Streaming

This section introduces signaling components of [DASH] and [HLS] manifests that are utilized by the low-latency live streaming use-case as defined in Section 4.2.

5.2.1 CMAF Chunk Signaling

5.2.1.1 DASH / HLS Signaling Overview

In live streaming contexts it may be desirable to have players perform playback on the very latest segments, which may even not be fully produced yet, to produce a low latency operation mode. For clients to operate in this mode, they need an additional set of signaling to enable CMAF Chunk addressing and utilize Aggregated Response Transfer Methods.

Within [DASH] manifests, low latency operation information is described through attributes and elements describing a Representation:

- The `SegmentTemplate@availabilityTimeOffset` attribute for a Representation is set to make segments available earlier than their availability time in the [DASH] timing model.
- The `SegmentTemplate@availabilityTimeComplete` attribute for a Representation is set to false indicating that segments may not be complete at their adjusted availability time.
- The `Resync` element is used to signal segment substructure bounding information to enable player operation on structures smaller than a full segment.

General [DASH] manifests do not provide explicit description of CMAF Chunks; instead the [DASH] timing model adjustments along with Aggregated Response Transfer Methods are used to enable low latency playback with basic DASH `SegmentTemplate` addressing modes:

```
<Representation id="foo" mimeType="video/mp4" ...>
    <SegmentTemplate
        timescale="1000000"
        duration="8008000"
        startNumber="1"
        initialization="foo-video.init.mp4"
        media="foo-video.$Number$.m4s"
        availabilityTimeOffset="2.0"
        availabilityTimeComplete="false" />
</Representation>
```

Figure 5.7 – DASH Low Latency SegmentTemplate Example

```
<Representation id="foo" mimeType="video/mp4" ...>
    <SegmentTemplate
        timescale="1000000"
        initialization="foo-video.init.mp4"
        media="foo-video.$Time$.m4s"
        availabilityTimeOffset="2.0"
        availabilityTimeComplete="false">
        <SegmentTimeline>
            <S t="0" d="8008000" r="100" />
        </SegmentTimeline>
    </SegmentTemplate>
</Representation>
```

Figure 5.8 – DASH Low Latency SegmentTemplate with SegmentTimeline Example

Where CMAF Chunk descriptions are desired, bounding details of chunks may be signaled with a Resync element. This provides hints to the players about where they might find appropriate decode start points with container level parsing. For further details refer to [DASH] 5.3.13 Resynchronization:

```
<Representation id="foo" mimeType="video/mp4" ...>
    <Resync
        type="0"
        dT="500000"
        dImin="0.03125"
        dImax="0.09375" />
    <SegmentTemplate
        timescale="1000000"
        duration="8008000"

        startNumber="1"
        initialization="foo-video.init.mp4"
        media="foo-video.$Number$.m4s"
        availabilityTimeOffset="2.0"
        availabilityTimeComplete="false" />
</Representation>
```

Figure 5.9 – DASH Low Latency Resync Example

Individual chunks can be described in the [HLS] Media Playlist as what [HLS] calls a Partial Segment. The relevant [HLS] signaling is as follows:

- The `EXT-X-PART-INF` tag signals the target duration of Partial Segments as the value of the `PART-TARGET` attribute.
- The `EXT-X-PART` tag identifies one or more Partial Segments for a parent segment along with various properties of the Partial Segment.
- The `EXT-X-PRELOAD-HINT` tag provides a URI and optional byte range start that a player can use to make a speculative blocking request that results in an aggregated response from the server.

- The `EXT-X-SERVER-CONTROL` tag provides the suggested part hold back duration, which is the minimum distance behind the live point that the player should operate.
- The `EXT-X-PROGRAM-DATE-TIME` tag associates the first sample of a Media Segment with an absolute date and/or time allowing more-precise mapping between Segments across Renditions.

Partial Segment signaling may be used to signal CMAF Chunks within a CMAF Segment where that information is required to meet operational constraints. Where CMAF Chunks are made available as independent files they may be directly addressed and hinted at:

```
#EXTM3U
#EXT-X-TARGETDURATION:8
#EXT-X-VERSION:7
#EXT-X-PART-INF:PART-TARGET=2.002
#EXT-X-SERVER-CONTROL:PART-HOLD-BACK=6.006
...
#EXT-X-PART:DURATION=2.002,INDEPENDENT=YES,URI="foo-
video.1.1.mp4"
#EXT-X-PART:DURATION=2.002,URI="foo-video.1.2.mp4"
#EXT-X-PART:DURATION=2.002,URI="foo-video.1.3.mp4"
#EXT-X-PART:DURATION=2.002,URI="foo-video.1.4.mp4"
#EXTINF:8.008,
foo-video.1.mp4

#EXT-X-PART:DURATION=2.002,INDEPENDENT=YES,URI="foo-
video.2.1.mp4"
#EXT-X-PART:DURATION=2.002,URI="foo-video.2.2.mp4"
#EXT-X-PRELOAD-HINT:TYPE=PART,URI="foo-video.2.3.mp4"
```

Figure 5.10 – HLS Media Playlist with Partial Segment Signaling for Independent Files per Substructure

Where CMAF Chunk addressing is necessary, but a single file per CMAF Segment is produced, the CMAF Chunks may be addressed and hinted at utilizing byte range requests:

```
#EXTM3U
#EXT-X-TARGETDURATION:8
#EXT-X-VERSION:7#EXT-X-PART-INF:PART-TARGET=2.002
#EXT-X-SERVER-CONTROL:PART-HOLD-BACK=6.006
...
#EXT-X-PART:DURATION=2.002,INDEPENDENT=YES,URI="foo-
video.1.mp4",BYTERANGE="20000@0"
#EXT-X-PART:DURATION=2.002,URI="foo-
video.1.mp4",BYTERANGE="15000@20000"
#EXT-X-PART:DURATION=2.002,URI="foo-
video.1.mp4",BYTERANGE="15000@35000"
#EXT-X-PART:DURATION=2.002,URI="foo-
video.1.mp4",BYTERANGE="15000@50000"
#EXTINF:8.008,
foo-video.1.mp4
```

```
#EXT-X-PART:DURATION=2.002,INDEPENDENT=YES,URI="foo-
video.2.mp4",BYTERANGE="20000@0”
#EXT-X-PART:DURATION=2.002,URI="foo-
video.2.mp4",BYTERANGE="15000@20000"
#EXT-X-PRELOAD-HINT:TYPE=PART,URI="foo-video.2.mp4",BYTERANGE-
START=35000
```

Figure 5.11 – HLS Media Playlist with Partial Segment Signaling for Single File per CMAF Segment

5.2.1.2 Derivation of HLS Signaling from DASH Signaling

When producing [HLS] Media Playlists from [DASH] Manifests:

- All fully available segments in the [DASH] manifest time shift buffer utilize the translation mechanics outlined in 5.1.2.2.

  Note: As `Resync` elements do not describe accurate byte boundaries within the segment resource it cannot be used to directly generate `EXT-X-PART` entries without further segment inspection or secondary information sources. Due to this, the HLS translation will not be low latency enabled. We are continuing to investigate this functionality and will provide updates in further publications.

5.2.1.3 Derivation of DASH Signaling from HLS Signaling

When producing [DASH] Manifests from [HLS] Playlists:

- Segment addressing per Representation follows the same mechanics outlined in 5.1.2.3.
- The `SegmentTemplate@availabilityTimeOffset` attribute is set sufficiently large enough, ideally to the segment duration minus the duration of one chunk, to allow players to request segments as they are being produced without encountering 404’s for segments not yet started.
- The `SegmentTemplate@availabilityTimeComplete` attribute is set to false, indicating the segments at the live head may not be completely produced at their offset availability time.

5.2.2 Service Latency Parameters

5.2.2.1 DASH / HLS Signaling Overview

Service providers may desire to specify parameters around the latency a client player operates in and can do so with Service Latency Parameter signaling.

Within [DASH] manifests, the desired latency parameters are described through the `Latency` element of the `ServiceDescription` element at the top level of the `MPD`. The `Latency` element allows for the description of multiple aspects of latency including:

- `Latency@target`, which describes the service provider’s preferred latency in milliseconds when compared to a producer reference time also signaled within the manifest.

- `Latency@max`, which describes the maximum amount of latency the service provider desires to have in the stream, preferring the stream to not be played if latency below this threshold cannot be maintained.
- `Latency@min`, which describes the minimum amount of latency the service provider expects to have in the stream, preferring the stream to not be played at a lower latency, even if the client player is capable of reaching a lower one.

The [DASH] Specification does not constrain the values that can be specified in the Latency attributes; however, targets that are not achievable by players as a result of the chunk sizes and availability times should be avoided. As different client players and scenarios may have different latency targets for the same presentation, multiple `ServiceDescription` elements may be present with `Scope` elements that limit the application of the description to clients that understand the specified scopes. For further information refer to [DASH] Annex K DASH Service Description.

```
<ServiceDescription id="1">
    <Latency
        referenceId="ref-1"
        target="1000"
        min="500"
        max="2000" />
</ServiceDescription>
...
    <ProducerReferenceTime
        id="ref-1"
        type="captured"
        wallClockTime="6839535273404006400"
        presentationTime="300000" />
```

Figure 5.12 – DASH Service Description with Latency Example

Within [HLS] Media Playlists, the desired latency parameters are described through the `EXT-X-SERVER-CONTROL` tag. This tag allows the specification of two latency targets:

- The `HOLD-BACK` attribute allows the specification of the target latency in a normal client playback mode.
    - This attribute is optional in any Media Playlist.
    - When specified, it is defined to be at least three times the target duration signaled by the `EXT-X-TARGETDURATION` tag in the same Media Playlist.
- The `PART-HOLD-BACK` attribute allows the specification of the target latency in a low latency client playback mode.
    - This attribute is required in any Media Playlist that contains the `EXT-X-PART-INF` tag.
    - When specified, it is defined to be at least twice the target part duration signaled by the `PART-TARGET` attribute of the `EXT-X-PART-INF` tag.

For further information, refer to [HLS] 4.4.3.8 EXT-X-SERVER-CONTROL.

```
#EXTM3U
#EXT-X-TARGETDURATION:8
#EXT-X-VERSION:7#EXT-X-PART-INF:PART-TARGET=2.002
#EXT-X-SERVER-CONTROL:HOLD-BACK=24.024,PART-HOLD-BACK=4.004
...
#EXT-X-PART:DURATION=2.002,INDEPENDENT=YES,URI="foo-
video.1.mp4",BYTERANGE="20000@0"
#EXT-X-PART:DURATION=2.002,URI="foo-
video.1.mp4",BYTERANGE="15000@20000"
#EXT-X-PART:DURATION=2.002,URI="foo-
video.1.mp4",BYTERANGE="15000@35000"
#EXT-X-PART:DURATION=2.002,URI="foo-
video.1.mp4",BYTERANGE="15000@50000"
#EXTINF:8.008,
foo-video.1.mp4
```

Figure 5.13 – HLS Media Playlist with EXT-X-SERVER-CONTROL

5.2.2.2 Derivation of HLS Signaling from DASH Signaling

When producing [HLS] Media Playlists from [DASH] Manifests:

- The segment duration provided by either the `SegmentTemplate@duration` attribute or derived from the explicit `SegmentTimeline.S` elements is used to generate the `EXT-X-TARGETDURATION` tag.
- If present, the value of the `Resync@dT` attribute normalized to seconds using the appropriate time scale is used to generate the `EXT-X-PART-INF` tag.
- The latency related attributes of the `EXT-X-SERVER-CONTROL` tag are set where:
    - The `PART-HOLD-BACK` attribute is set based on the service desired low latency target, if known, otherwise the attribute is omitted.

      Note: The value of the `Latency@target` attribute of the `ServiceDescription` element can be considered an upper bound for the `PART-HOLD-BACK` attribute, but it may be larger than the intended latency. This difference is due to the `PART-HOLD-BACK` attribute being relative to the end of the Media Playlist and the `Latency@target` attribute being relative to the wall clock time associated with the presentation.
    - The `HOLD-BACK` attribute is set to the value of the `MPD@suggestedPresentationDelay` if present, otherwise the attribute is omitted.
    - In both cases, the minimum values the attributes may be set to are limited by their relation to the generated target duration descriptions per the HLS specification.

5.2.2.3 Derivation of DASH Signaling from HLS Signaling

When producing [DASH] Manifests from [HLS] Playlists:

- If present, the `HOLD-BACK` attribute of the `EXT-X-SERVER-CONTROL` tag is set as the value of the `MPD@suggestedPresentationDelay` attribute.
- If known, the desired low latency target is used to:
    - Generate a `ServiceDescription` element with a `Latency` element where `Latency@target` is set to the desired target converted to milliseconds.
    - Generate a `ProducerReferenceTime` element with `@type`, `@wallClockTime`, and `@presentationTime` attributes set to capture the relation of stream time to wall clock time that is known to the conversion service.

  Note: The value of the `PART-HOLD-BACK` attribute of the `EXT-X-SERVER-CONTROL` tag can be considered a lower bound for the `Latency@target` attribute, but it may be smaller than the intended latency. This difference is due to the `PART-HOLD-BACK` attribute being relative to the end of the Media Playlist and the `Latency@target` attribute being relative to the wall clock time associated with the presentation.

  Note: As `EXT-X-SERVER-CONTROL` is per Media Playlist for [HLS] and `ServiceDescription` elements are per MPD or `Period` element for [DASH], the basic expectation here is that all Media Playlists supply equivalent details and that only one needs to be sourced. If this assumption is not true, the conversion needs to use a heuristic for choosing the `ServiceDescription` elements to place in the MPD.

5.3 Encrypted Media Presentations

This section introduces signaling components of [DASH] and [HLS] manifests that are related to the encrypted media presentation use-case as defined in Section 4.3.

5.3.1 Encryption Key Signaling

5.3.1.1 DASH / HLS Signaling Overview

Encrypted tracks require signaling of the encryption within the manifests to enable the client players to initialize the DRM pipelines on the devices as part of the media pipeline instantiation. There are three types of data commonly associated with encryption signaling:

- The Key Identifier (KID)
- The Common Encryption Mode
- Protection System Specific Initialization Data

Within [DASH] manifests, the `ContentProtection` descriptor at the `AdaptationSet` or `Representation` level is used to provide encryption signaling. Multiple descriptors are provided with different DRM vendor SystemIDs in the `@schemeIdUri` attributes to signal system specific data:

- Common Encryption details, defined by [CENC], are provided in a descriptor where:
    - The `@schemeIdUri` attribute is set to `"urn:mpeg:dash:mp4protection:2011"`.

  - The `@value` attribute is set to the common encryption mode utilized for the track, either "`cenc`" or "`cbcs`" for CMAF conforming content.
  - An optional `@cenc:default_KID` attribute is present and set to the UUID form of the KID.
- DRM specific initialization data is contained in a descriptor where:
  - The `@schemeIdUri` attribute is set to the UUID defined as the vendor SystemID.
  - The remaining attributes are set based on the vendor system semantics.
  - The descriptor may contain a `cenc:pssh` element containing the base64 encoding of a `pssh` box that would otherwise be found in a track file.

For further information refer to [DASH] 5.8.5.2 Content protection as well as [CENC] 11 XML representation of Common Encryption parameters.

```
<AdaptationSet mimeType="video/mp4" ...>
    <ContentProtection
        schemeIdUri="urn:mpeg:dash:mp4protection:2011"
        value="cbcs"
        cenc:default_KID="939f032e-cd6d-4fbc-9ad0-
9d082ba37420"/>
    <ContentProtection
        schemeIdUri="urn:uuid:4b25f294-46df-494b-aed8-
efb9848648b7"
        value="FooBar DRM 1.0">
        <cenc:pssh>...</cenc:pssh>
    </ContentProtection>
</AdaptationSet>
```

Figure 5.14 – DASH Manifest with Content Protection Elements

Within [HLS] manifests, individual Media Playlists provide basic encryption details with the `EXT-X-KEY` tag that proceeds the segments with encryption applied to them. Multiple tags may be present to provide different DRM vendor specific system information. The attributes of `the EXT-X-KEY` tag are set as follows:

- The `METHOD` attribute is set to:
  - `SAMPLE-AES` for `cbcs` mode encrypted files.
  - `SAMPLE-AES-CTR` for `cenc` mode encrypted files.

    Note: `SAMPLE-AES-CTR` is not an HLS compliant value, but it is an industry convention that has been widely adopted and documented here for consistency.
- The `KEYFORMAT` attribute is set to the identifier of a specific DRM vendor system that can be used to interpret the `KEYFORMATVERSIONS` and `URI` attributes.
- The remaining attributes: `KEYFORMATVERSIONS`, `URI`, and `IV` are set based on the DRM vendor system semantics.

```
#EXTM3U
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-KEY:METHOD=SAMPLE-
AES,KEYFORMAT="com.apple.streamingkeydelivery",KEYFORMATVERSIONS="1",U
RI="skd://939f032e-cd6d-4fbc-9ad0-9d082ba37420"
#EXT-X-KEY:METHOD=SAMPLE-
AES,KEYFORMAT="com.foobar.drm",KEYFORMATVERSIONS="1",URI="data:text/pl
ain;charset=UTF-16;base64,..."
#EXT-X-MAP:URI="foo-video.init.mp4"
#EXTINF:8.008,
foo-video.0.mp4
#EXTINF:8.008,
foo-video.1.mp4
...
```

Figure 5.15 – HLS Media Playlist with EXT-X-KEY tags

In addition to the details provided in the individual Media Playlists, a summary of encryption details for all Media Playlists may also be provided in the Multivariant Playlist with the `EXT-X-SESSION-KEY` tag. The attributes of the `EXT-X-SESSION-KEY` tag are the same as the `EXT-X-KEY` tag and meant to have the values mirror the values within the Media Playlists.

```
#EXTM3U
#EXT-X-VERSION:7

#EXT-X-STREAM-
INF:CODECS="avc1.640020,mp4a.40.2",RESOLUTION=960x540,FRAME-
RATE=30.000,...
foo-video.m3u8

#EXT-X-SESSION-KEY:METHOD=SAMPLE-
AES,KEYFORMAT="com.apple.streamingkeydelivery",KEYFORMATVERSIONS="1",U
RI="skd://939f032e-cd6d-4fbc-9ad0-9d082ba37420"
#EXT-X-SESSION-KEY:METHOD=SAMPLE-
AES,KEYFORMAT="com.foobar.drm",KEYFORMATVERSIONS="1",URI="data:text/pl
ain;charset=UTF-16;base64,..."
```

Figure 5.16 – HLS Multivariant Playlist with EXT-X-SESSION-KEY tags

5.3.1.2 Derivation of HLS Playlists from DASH Manifests

When producing [HLS] Playlists from [DASH] Manifests:

- The `ContentProtection` descriptors related to the targeted Representation for the Media Playlist are used to generate `EXT-X-KEY` tags where:
    - The common encryption descriptor informs the value of the `METHOD` attribute for all tags.
    - An `EXT-X-KEY` tag is generated for each specified vendor system based on their individual signaling requirements.

- The groups of `ContentProtection` descriptors MAY be used to generate a full set of `EXT-X-SESSION-KEY` tags in the Multivariant Playlist where:
    - The common encryption descriptor of the group informs the value of the `METHOD` attribute for all tags.
    - An `EXT-X-SESSION-KEY` tag is generated for each specified vendor system in the descriptor group based on their individual signaling requirements.

5.3.1.3 Derivation of DASH Manifests from HLS Playlists

When producing [DASH] Manifests from [HLS] Playlists:

- One set of `EXT-X-KEY` tags per CMAF Switching Set should be used to generate `ContentProtection` descriptors for the `AdaptationSet` elements generated for the Switching Set.
- The common encryption descriptor MAY be generated from the `EXT-X-KEY` tags of a Switching Set where:
    - The `@value` attribute is set to:
        - `"cbcs"` if the `METHOD` attribute is `SAMPLE-AES`.
        - `"cenc"` if the `METHOD` attribute is `SAMPLE-AES-CTR`.
    - The `@cenc:default_KID` is set to:
        - The default KID known from an `EXT-X-KEY` tag if it can be derived from a vendor value.
        - The value of the `default_KID` field in the CMAF Header of one of the tracks in the switching set.
- DRM vendor system specific descriptors MAY be generated from the system `specific EXT-X-KEY` tags using the semantics outlined by the DRM vendor.

5.4 Presentation Splicing

This section introduces signaling components of [DASH] and [HLS] manifests that are related to the presentation splicing use-case as defined in Section 4.4.

5.4.1 Alignment of CMAF Switching Sets and Tracks across Presentations

5.4.1.1 DASH / HLS Signaling Overview

When a service deployment desires to play multiple CMAF Presentations together as a single logical stream, the manifest is used to describe how the presentations should be combined to create a single playback timeline.

For this overview we will utilize a common example to exhibit parallel descriptions of the same concepts, in this common example we have the following features:

- Three CMAF Presentations:
    - `"ad-1"` a 15 second advertisement containing:

        - An English audio Switching Set with a single AAC encoded Track.
        - An English video Switching Set with a set of H264 encoded Tracks at the resolutions:
            - 960x540
            - 1280x720
    - "content" a 10-minute piece of content containing:
        - An English audio Switching Set with a single AAC encoded Track.
        - A Spanish audio Switching Set with a single AAC encoded Track.
        - An English video Switching Set with a set of H264 encoded Tracks at the resolutions:
            - 960x540
            - 1280x720
        - An English video Switching Set with a set of H265 encoded Tracks at the resolutions:
            - 1280x720
            - 1920x1080
    - "ad-2" a 30 second advertisement containing:
        - An English audio Switching Set with a single AAC encoded Track.
        - A Spanish audio Switching Set with a single AAC encoded Track.
        - An English video Switching Set with a set of H264 encoded Tracks at the resolutions:
            - 960x540
- The desired ordering of the presentations is:
    - "ad-1"
    - "content"
    - "ad-2"

  Note: We are depicting here a set of presentations that do not follow the consistency constraints outlined in Section 4.4. We do this to illustrate the extent of matching heuristics necessary in manifest conversion but recommend the constraints of 4.4 are followed to simplify implementation and deployment.

Within [DASH] manifests, each CMAF presentation is defined as one or more `Period` elements and the `Period` elements are ordered in the MPD based on the desired playout order. As Periods are independent elements, no manifest level constraints are placed on the availability of different CMAF Selection Sets and Switching Sets between CMAF Presentations:

```
<MPD id="combined-foo" ...>
    <Period id="ad-1" duration="15.0S">
        <AdaptationSet contentType="video" lang="en-US" ...>
            <Representation
                codecs="avc1.640020"
                width="960" height="540"
                frameRate="30" .../>
            <Representation
                codecs="avc1.640020"
                width="1280" height="720"
                frameRate="30" .../>
        </AdapationSet>
        <AdaptationSet contentType="audio" lang="en-US" ...>
            <Representation codecs="mp4a.40.2" .../>
        </AdapationSet>
    </Period>
    <Period id="content" duration="600.0S">
        <AdaptationSet contentType="video" lang="en-US" ...>
            <Representation
                codecs="avc1.640020"
                width="960" height="540"
                frameRate="30" .../>
            <Representation
                codecs="avc1.640020"
                width="1280" height="720"
                frameRate="30" .../>
            </AdapationSet>
        <AdaptationSet contentType="video" lang="en-US" ...>
            <Representation
                codecs="hvc1.2.4.L123.B0"
                width="1280" height="720"
                frameRate="30" .../>
            <Representation
                codecs="hvc1.2.4.L123.B0"
                width="1920" height="1080"
                frameRate="30" .../>
            </AdapationSet>
        <AdaptationSet contentType="audio" lang="en-US" ...>
            <Representation codecs="mp4a.40.2" .../>
            </AdapationSet>
        <AdaptationSet contentType="audio" lang="es" ...>
            <Representation codecs="mp4a.40.2" .../>
            </AdapationSet>
    </Period>
    <Period id="ad-2" duration="30.0S">
        <AdaptationSet contentType="video" lang="en-US" ...>
            <Representation
                codecs="avc1.640020"
                width="960" height="540"
                frameRate="30" .../>
            </AdapationSet>
```

```
        <AdaptationSet contentType="audio" lang="en-US" ...>
            <Representation codecs="mp4a.40.2" .../>
            </AdapationSet>
        <AdaptationSet contentType="audio" lang="es" ...>
            <Representation codecs="mp4a.40.2" .../>
            </AdapationSet>
    </Period>
</MPD>
```

Figure 5.17 – DASH Manifest combining Multiple CMAF Presentations

Within [HLS] Playlists, the Multivariant Playlist defines the set of CMAF Selection Sets and Switching Sets that will be made available in the combined presentation. The CMAF Selection Sets, Switching Sets, and Tracks across Presentations will be heuristically aligned to generate the descriptive attributes of the variants listed in the Multivariant:

```
#EXTM3U
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="audio-
en",NAME="English",LANGUAGE="en-
US",AUTOSELECT=YES,DEFAULT=YES,CHANNELS="2",URI="combined-foo/aac-
en.m3u8"
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="audio-
es",NAME="Spanish",LANGUAGE="es",AUTOSELECT=YES,DEFAULT=YES,CHANNELS="
2",URI="combined-foo/aac-es.m3u8"

#EXT-X-STREAM-
INF:CODECS="avc1.640020,mp4a.40.2",RESOLUTION=960x540,FRAME-
RATE=30.00,AUDIO="audio-en",...
combined-foo/h264-540.m3u8
#EXT-X-STREAM-
INF:CODECS="avc1.640020,mp4a.40.2",RESOLUTION=1280x720,FRAME-
RATE=30.00,AUDIO="audio-en",...
combined-foo/h264-720.m3u8
#EXT-X-STREAM-
INF:CODECS="hvc1.2.4.L123.B0,avc1.640020,mp4a.40.2",RESOLUTION=1280x72
0,FRAME-RATE=30.00,AUDIO="audio-en",...
combined-foo/h265-720.m3u8
#EXT-X-STREAM-
INF:CODECS="hvc1.2.4.L123.B0,avc1.64002a,mp4a.40.2",RESOLUTION=1920x10
80,FRAME-RATE=30.00,AUDIO="audio-en",...
combined-foo/h265-1080.m3u8

#EXT-X-STREAM-
INF:CODECS="avc1.640020,mp4a.40.2",RESOLUTION=960x540,FRAME-
RATE=30.00,AUDIO="audio-es",...
combined-foo/h264-540.m3u8
#EXT-X-STREAM-
INF:CODECS="avc1.640020,mp4a.40.2",RESOLUTION=1280x720,FRAME-
RATE=30.00,AUDIO="audio-es",...
```

```
combined-foo/h264-720.m3u8
#EXT-X-STREAM-
INF:CODECS="hvc1.2.4.L123.B0,avc1.640020,mp4a.40.2",RESOLUTION=1280x72
0,FRAME-RATE=30.00,AUDIO="audio-es",...
combined-foo/h265-720.m3u8
#EXT-X-STREAM-
INF:CODECS="hvc1.2.4.L123.B0,avc1.64002a,mp4a.40.2",RESOLUTION=1920x10
80,FRAME-RATE=30.00,AUDIO="audio-es",...
combined-foo/h265-1080.m3u8
```

Figure 5.18 – HLS Multivariant Playlist combining Multiple CMAF Presentations

The Media Playlists defines the segments of the heuristically aligned tracks in the desired playout order with `EXT-X-DISCONTINUITY` tags used to signal the boundaries of CMAF Presentations:

```
#EXTM3U
#EXT-X-TARGETDURATION:5
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS

#EXT-X-MAP:URI="ad-1-h264-720.mp4",BYTERANGE="0-a"
#EXTINF:5,
#EXT-X-BYTERANGE:<length>@<offset>
ad-1-h264-720.mp4
...
#EXT-X-DISCONTINUITY
#EXT-X-MAP:URI="content-h264-720.mp4",BYTERANGE="0-a"
#EXTINF:5
#EXT-X-BYTERANGE:<length>@<offset>
content-h264-720.mp4
...
#EXT-X-DISCONTINUITY
#EXT-X-MAP:URI="ad-2-h264-540.mp4",BYTERANGE="0-a"
#EXTINF:5
#EXT-X-BYTERANGE:<length>@<offset>
ad-2-h264-540.mp4
...
```

Figure 5.19 – HLS Variant Playlist for Heuristically Aligned 720p H264 Variant

Notice that even though `ad-2` does not contain a 720p variant the next largest rendition is included instead as Media Playlists cannot contain gaps.

```
#EXTM3U
#EXT-X-TARGETDURATION:5
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS

#EXT-X-MAP:URI="ad-1-h264-720.mp4",BYTERANGE="0-a"
#EXTINF:5,
#EXT-X-BYTERANGE:<length>@<offset>
ad-1-h264-720.mp4
```

```
...
#EXT-X-DISCONTINUITY
#EXT-X-MAP:URI="content-h265-1080.mp4",BYTERANGE="0-a"
#EXTINF:5
#EXT-X-BYTERANGE:<length>@<offset>
content-h265-1080.mp4
...
#EXT-X-DISCONTINUITY
#EXT-X-MAP:URI="ad-2-h264-540.mp4",BYTERANGE="0-a"
#EXTINF:5
#EXT-X-BYTERANGE:<length>@<offset>
ad-2-h264-540.mp4
...
```

Figure 5.20 – HLS Variant Playlist for Heuristically Aligned 1080p HEVC Variant

Notice that we are mixing both codecs and resolutions due to HEVC providing a resolution unavailable in both ads.

```
#EXTM3U
#EXT-X-TARGETDURATION:5
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS

#EXT-X-MAP:URI="ad-1-aac-en.mp4",BYTERANGE="0-a"
#EXTINF:5,
#EXT-X-BYTERANGE:<length>@<offset>
ad-1-aac-en.mp4
...
#EXT-X-DISCONTINUITY
#EXT-X-MAP:URI="content-aac-es.mp4",BYTERANGE="0-a"
#EXTINF:5
#EXT-X-BYTERANGE:<length>@<offset>
content-aac-es.mp4
...
#EXT-X-DISCONTINUITY
#EXT-X-MAP:URI="ad-2-aac-es.mp4",BYTERANGE="0-a"
#EXTINF:5
#EXT-X-BYTERANGE:<length>@<offset>
ad-2-aac-es.mp4
...
```

Figure 5.21 – HLS Variant Playlist for Heuristically Aligned Spanish Audio Variant

Notice here that we are not mixing codecs or encoding parameters but are mixing the logical track property of language. This would be an example of Selection Set mixing to ensure a complete timeline when an appropriate match does not exist across presentations.

5.4.1.2 Derivation of HLS Playlists from DASH Manifests

When producing [HLS] Playlists from [DASH] Manifests, the individual CMAF Presentations MAY be rediscovered by directly examining the individual Period elements.

- From the individual CMAF Presentations, the set of CMAF Selection Sets, Switching Sets, and Tracks to be included in the combined Multivariant Playlist need to be decisioned:
    - If the primary Presentation in the experience is known by the conversion logic, the Selection Sets, Switching Sets, and Tracks provided by the primary presentation can be used as the base.
    - Otherwise, a superset of Selection Sets, Switching Sets, and Tracks needs to be generated by looking across all present Presentations.

With the set of CMAF Selection Sets, Switching Sets, and Tracks to be included in the Multivariant Playlist decided, each CMAF Presentation needs to assign a Selection Set, Switching Set, and Track to be used in variant alignment:

- For every target Selection Set in the Multivariant Playlist:
    - If the Presentation has an equivalent Selection Set, the Presentation Selection Set is assigned directly.
    - If the Presentation does not have an exact equivalent, an alternate for the same media type is chosen based on operator defined precedence of defining attributes such as language, role, and track characteristics.

      Note: If a Presentation has no Selection Sets at all for a media type the process needs to choose to either omit the Selection Set from the Multivariant Playlist or error out of the conversion.
- For every target Switching Set in the target Selection Sets:
    - If the Presentation has an equivalent Switching Set for the Selection Set assigned to the target Selection Set, the Presentation Switching Set is assigned directly.
    - If the Presentation does not have an exact equivalent, an alternate is chosen based on operator defined precedence of defining attributes such as encoding.
- For every target Track in the target Switching Sets:
    - For video tracks:
        - If the Presentation has an equivalent Track for the Switching Set assigned to the target Switching Set, the Presentation Track is assigned directly.
        - Otherwise, an alternate is chosen based on an operator defined fallback heuristic. A suggested fallback heuristic for video tracks selects:
            - A Track with equivalent resolution, frame rate, and dynamic range, but lower bandwidth.
            - A Track with equivalent resolution and frame rate, but alternate dynamic range
            - A Track with equivalent resolution, but lower frame rate.
            - A Track with equivalent frame rate, but lower resolution.

- For audio tracks:
  - If the Presentation has an equivalent Track for the Switching Set assigned to the target Switching Set, the presentation Track is assigned directly.
  - Otherwise, an alternate is chosen based on an operator defined fallback heuristic. A suggested fallback heuristic for audio tracks selects:
    - A Track with equivalent channel configuration, but lower bandwidth.
    - A Track with an alternative channel configuration.

With the alignment of CMAF Selection Sets, Switching Sets, and Tracks across Presentations known, the HLS Multivariant Playlist can now be generated based on the target information. Where a mixture of sets and tracks across presentations occurs, proper usage defined by the HLS specification for `EXT-X-STREAM-INF` and `EXT-X-MEDIA` tags should be followed.

The HLS Media Playlists can then be generated by outputting the segment descriptions for each track aligned to the HLS Multivariant Playlist variant description in the original order of the Periods in the DASH manifest. The segment description outputs of the individual CMAF presentations are separated by an `EXT-X-DISCONTINUITY` tag to signal the presentation boundary point.

5.4.1.3 Derivation of DASH Manifests from HLS Playlists

When producing [DASH] manifests from [HLS] Playlists, the individual CMAF presentations need to be rediscovered by combining information from the Multivariant Playlist, Media Playlists, and Media Initialization Sections:

- From the Multivariant Playlist, a set of target CMAF Selection Sets, Switching Sets, and Tracks can be discovered.
- The individual CMAF Presentations are discovered by combining the target structure data and an examination of the Variant Playlists:
  - The `EXT-X-DISCONTINUITY` tags signal the separation of individual Tracks and the ordering of the Presentations.
  - The assigned Track of individual Presentations for the target Track of the HLS Variant is equal to the Track described between the `EXT-X-DISCONTINUITY` tags
  - Where Tracks were duplicated across HLS Variants due to fallback assignment heuristics, deduplication can be done by eliminating Tracks referencing the same Media Segments.
- After examining every Variant Playlist, each Presentation should have a known set of assigned Tracks for the target Selection Sets and Switching Sets:
  - As the Multivariant Playlist Track description is lossy with spliced presentations, explicit Track descriptions can be discovered by examining the headers for each Track.

- Explicit track descriptions should allow for the target Switching Set descriptions to be made explicitly into the Switching Sets of the Presentation and further deduplicated based on equivalency.
- Further deduplication of Selection Sets is possible only if the conversion logic has an ability to rediscover the attributes that differentiate Selection Sets from the contained Track Files.

With the individual CMAF Presentations known, the [DASH] manifest can be created by:

- Generating a `Period` describing each Presentation.
- Ordering and splicing the `Period` elements based on the ordering and splicing discovered from the HLS Variant Playlists.

5.5 Carriage of Timed Event Data

This section introduces signaling components of [DASH] and [HLS] manifest that are related to the signaling of timed event data as defined in Section 4.5.

5.5.1 In-band Timed Event Data Signaling

5.5.1.1 DASH / HLS Signaling Overview

When inserting timed event signals directly into the CMAF Presentation media it is desirable to predefine the signals available in the stream to the client so that it may properly listen for them and inform higher application structures of their potential presence.

Within [DASH] manifests, in-band timed event data is described through an `InbandEventStream` element associated with the Representation containing the messages. The `InbandEventStream` element is primarily defined by two attributes based on the in-band event data:

- The `InbandEventStream@schemeIdUri` attribute identifies the message scheme and should match the `schemeIdUri` field of the `emsg` boxes embedded in the media.
- The `InbandEventStream@value` attribute semantics are defined by the owners of the scheme identified by the `InbandEventStream@schemeIdUri` attribute and should match the value field within the `emsg` boxes embedded in the media.

```
<Representation id="foo" mimeType="video/mp4" ...>
    <InbandEventStream
        schemeIdUri="urn:com:example:inband:2023"
        value="1" />
    <BaseURL>foo-video.mp4</BaseURL>
    <SegmentBase indexRange="a-b" indexRangeExact="true">
        <Initialization range="0-a" />
    </SegmentBase>
</Representation>
```

Figure 5.22 – DASH Representation with InbandEventStream Elements

Where a set of Representations all contain the same in-band timed event data and are aligned in an `AdaptationSet` with `@segmentAlignment=true`, the `InbandEventStream` element may be present at the `AdaptationSet` level instead.

Within [HLS] playlists, in-band timed event data has no explicit signaling available at either the HLS Multivariant Playlist or the HLS Media Playlist level.

5.5.1.2 Derivation of HLS Signaling from DASH Signaling

Due to the lack of DASH Signaling, HLS derivation cannot be performed.

5.5.1.3 Derivation of DASH Signaling from HLS Signaling

Due to the lack of HLS Signaling, DASH derivation cannot be performed.

5.5.2 Manifest Timed Event Data Carriage

5.5.2.1 DASH / HLS Signaling Overview

When inserting timed event signals directly into the CMAF Presentation media is not possible or not desired, such as late-bound sourcing or session personalized data streams, event data may be alternatively carried directly in the DASH manifests and HLS playlists.

Within [DASH] manifests, timed event data is carried in an `EventStream` element present at the `Period` level of the manifest. The `EventStream` attributes define the shape of the event stream data:

- The `EventStream@schemeIdUri` attribute identifies the message scheme that applies to all data related to the event stream.
- The `EventStream@value` attribute semantics are defined by the owners of the scheme identified by the `EventStream@schemeIdUri` attribute.
- The `EventStream@timescale` attribute specifies the timescale in units per seconds to be used for deriving real-time duration values of events.
- The `EventStream@presentationTimeOffset` specifies the offset used to align the event stream data to the start of the containing Period.

The `EventStream` represents the actual event timing and data as `Event` child elements where:

- The `Event@presentationTime` is the presentation time, in timescale units, of the event relative to the start of the Period when taking into account the value of `EventStream@presentationTimeOffset`.
- The `Event@duration` is an optional duration, in timescale units, of the event, when not present the duration is unknown.
- The `Event@id` is an optional identifier for the event whose scope is defined by the `EventStream@schemeIdUri` and `EventStream@value` pair.
- The data of the event is carried within the body of the `Event` element and may be optionally encoded for delivery by the encoding signaled in `Event@contentEncoding`.

Further semantics and abilities of the `EventStream` and `Event` elements can be found in [DASH] 5.10.

```
<Period id="1">
    <EventStream
        schemeIdUri="urn:com:example:event:2022"
        timescale="100"
        value="1">
        <Event
            id="1"
            presentationTime="0"
            duration="500"><![CDATA[
            { "data": "foobar" }
        ]]></Event>
        <Event
            id="2"
            presentationTime="1000" ><![CDATA[
            { "data": "bazqux"}
        ]]></Event>
    </EventStream>
</Period>
```

Figure 5.23 – DASH Manifest with EventStream Example

Within [HLS] playlists, timed event data may be carried in `EXT-X-DATERANGE` tags in HLS Media Playlists. The `EXT-X-DATERANGE` tag associates a set of attribute/value pairs with a specific Date Range in the Playlist:

- The `ID` attribute uniquely identifies the Date Range within the Playlist.
- The `CLASS` attribute specifies the attributes and semantics of the Date Range.
- The `START-DATE`, `END-DATE`, `CUE`, `DURATION`, and `PLANNED-DURATION` attributes are used to define the Date Range.
- Custom attributes are any attribute with a preceding `X-` prefix.

As Date Ranges are tied to specific start and end dates, an `EXT-X-PROGRAM-DATE-TIME` must be present in the HLS Media Playlist to provide a timing anchor. This also means that their relative location in the HLS Media Playlist has no explicit meaning, they may be presented interleaved with segment data, apart from segment data, and sometimes in non-linear order.

The full semantics of `EXT-X-DATERANGE` definition are available within [HLS] Section 4.4.5.1, but generally provides broader capability and usage than timed event data mappings in [DASH].

```
#EXTM3U
#EXT-X-TARGETDURATION:8
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-PROGRAM-DATE-TIME:1970-01-01T00:00:00.000+00:00
```

```
#EXT-X-MAP:URI="foo-video.init.mp4"
#EXT-X-DATERANGE:ID="1",CLASS="example-event-class",START-DATE="1970-
01-01T00:00:00.000+00:00",DURATION="5.0",X-
DATA="eyJkYXRhIjoiZm9vYmFyIn0="
#EXTINF:8.008,
foo-video.1.mp4
#EXTINF:8.008,
foo-video.2.mp4
#EXT-X-DATERANGE:ID="2",CLASS="example-event-class",START-DATE="1970-
01-01T00:00:10.000+00:00",X-DATA="eyJkYXRhIjoiYmF6cXV4In0="
...
```

Figure 5.24 – HLS Media Playlist with EXT-X-DATERANGE Example

As illustrated by the [DASH] and [HLS] example above, both mechanisms can carry event data, but they are not explicitly equivalent. Specifically, the lack of equivalency between the `CLASS` attribute in [HLS] and the `@schemeIdUri` / `@value` attribute pair in [DASH] means a direct derivation is not available.

To enable interoperability, the HLS EMSG Event Data Binding specification [DASH-HLS, Annex A] is used to equally represent our [DASH] example as [HLS].

```
#EXTM3U
#EXT-X-TARGETDURATION:8
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-PROGRAM-DATE-TIME:1970-01-01T00:00:00.000+00:00
#EXT-X-MAP:URI="foo-video.init.mp4"
#EXT-X-DATERANGE:ID="00001",CLASS="urn:cta:wave:dash-hls:event-
daterange",START-DATE="1970-01-
01T00:00:00.000+00:00",DURATION="5.0",X-EVENT-SCHEME-ID-
URI="urn:com:example:event:2022",X-EVENT-VALUE="1",X-EVENT-ID="1",X-
EVENT-MESSAGE-DATA="eyJkYXRhIjoiZm9vYmFyIn0="
#EXTINF:8.008,
foo-video.1.mp4
#EXTINF:8.008,
foo-video.2.mp4
#EXT-X-DATERANGE:ID="00002",CLASS="urn:cta:wave:dash-hls:event-
daterange",START-DATE="1970-01-01T00:00:10.000+00:00",X-EVENT-SCHEME-
ID-URI="urn:com:example:event:2022",X-EVENT-VALUE="1",X-EVENT-
ID="1",X-EVENT-MESSAGE-DATA="eyJkYXRhIjoiYmF6cXV4In0="
...
```

Figure 5.25 – HLS Media Playlist with Annex 1 EXT-X-DATERANGE Binding Example

This data binding may be used to bring equivalent manifest level signaling representation to any event data scheme, though some schemes may provide their own explicit [DASH] and [HLS] binding specifications to better accommodate their scenarios.

5.5.2.2 Derivation of HLS Signaling from DASH Signaling

When producing [HLS] Playlist from [DASH] Manifests the HLS EMSG Event Data Binding specification [DASH-HLS, Annex A] is used where:

- Each `Event` in an `EventStream` is used to generate an `EXT-X-DATERANGE` where:
    - The `START-DATE` attribute is computed based on the value of the `EventStream@timescale`, `EventStream@presentationTimeOffset`, and `Event@presentationTime`, and containing `Period` time signaling.
    - The `DURATION` attribute is set to the value of `Event@duration` normalized by `EventStream@timescale`, if present.
    - The `ID` attribute is generated to provide a playlist unique value.
    - The `CLASS` attribute is set to the value `"urn:cta:wave:dash-hls:event-daterange"`.
    - The `X-EVENT-SCHEME-ID-URI` attribute is set to the value of the `EventStream@schemeIdUri` attribute.
    - The `X-EVENT-VALUE` attribute is set to the value of the `EventStream@value` attribute.
    - The `X-EVENT-ID` attribute is set to the value of the `Event@id` attribute, if present.
    - The `X-EVENT-MESSAGE-DATA` attribute is set to a base64 encoding of the data in the `Event` element body.

5.5.2.3 Derivation of DASH Signaling from HLS Signaling

When producing [DASH] Manifests from [HLS] Playlists, we assume the HLS EMSG Event Data Binding specification [DASH-HLS, Annex A] was used such that:

- For each `EXT-X-DATERANGE` with a `CLASS` attribute of `"urn:cta:wave:dash-hls:event-daterange"`:
    - A target `Period` for data representation is picked based on the `START-DATE` attribute.
    - An `EventStream` element is created if no existing `EventStream@schemeIdUri` and `EventStream@value` pair match the `X-EVENT-SCHEME-ID-URI` and `X-EVENT-VALUE` pair.

      Note: `EventStream@timescale` and `EventStream@presentationTimeOffset` attributes should be set appropriately, but no specific value is required as part of this conversion process.
    - An `Event` element is created in the matching `EventStream` element where:
        - `Event@id` is set to the value of the `X-EVENT-ID` attribute.
        - `Event@presentationTime` is set to represent the `START-DATE` attribute when considering the target `Period` timing, `EventStream@presentationTimeOffset` value, and `EventStream@timescale` value.

- The `Event` body should be set to the encoded base64 value of the `X-EVENT-MESSAGE-DATA` attribute, and `Event@contentEncoding` should be set to the value "`base64`".

5.6 Rotation of Encryption Keys

This section introduces signaling components of [DASH] and [HLS] manifests that are related to the rotation of encryption keys use-case as defined in Section 4.6.

5.6.1 KID Sample Group Description Changes

5.6.1.1 DASH / HLS Signaling Overview

When using sample group description-based rotation, the primary change communication mechanism is the box structures within individual segments as outlined in Section 4.6.

Within [DASH] manifests there is no explicit signaling provided for these segment-level changes.

Within [HLS] manifests, all Media Segments must be preceded by an `EXT-X-KEY` tag describing the encryption applied to them. This means that key rotations using sample group descriptions still require an `EXT-X-KEY` tag prior to the effect segment(s). There is no requirement for the `EXT-X-KEY` tag to be repeated prior to all changed segments, only prior to the applicable first segment.

```
#EXTM3U
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-KEY:METHOD=SAMPLE-
AES,KEYFORMAT="com.apple.streamingkeydelivery",KEYFORMATVERSIONS="1",U
RI="skd://939f032e-cd6d-4fbc-9ad0-9d082ba37420"
#EXT-X-MAP:URI="foo-video.init.mp4"
#EXTINF:8.008,
foo-video.0.mp4
#EXTINF:8.008,
foo-video.1.mp4
…
#EXTINF:8.008,
foo-video.499.mp4
#EXT-X-KEY:METHOD=SAMPLE-
AES,KEYFORMAT="com.apple.streamingkeydelivery",KEYFORMATVERSIONS="1",U
RI="skd://047ae84d-6b05-4afc-af70-b99aa46945a1"
#EXTINF:8.008,
foo-video.500.mp4
#EXTINF:8.008,
foo-video.501.mp4
…
```

Figure 5.26 – HLS Media Playlist with EXT-X-KEY tags before in-segment rotation

5.6.1.2 Derivation of HLS Signaling from DASH Signaling

When producing [HLS] Media Playlists from [DASH] Manifests:

- The default encryption signaling is derived as described in Section 5.3.1.2.
- As the segment-level key changes are not signaled in the DASH manifest, the translator must derive the encryption data from one of the following methods and create the appropriately positioned `EXT-X-KEY` tags:
    - Retrieve and parse the CMAF boxes of each segment looking for a `seig` box that overrides the encryption parameters.
    - Consume a secondary data stream that describes what encryption information exists for a presentation and to which segments the information applies.

  Note: An alternative method of DASH Period creation is not suggested here as there would be no signaling difference between the two periods since they only signal the `default_KID` and that value does not change.

5.6.1.3 Derivation of DASH Signaling from HLS Signaling

When producing [DASH] Manifests from [HLS] Playlists:

- The default encryption signaling is derived as described in Section 5.3.1.3.
- The segment-level key changes signaled by `EXT-X-KEY` tags are not included in the DASH output as they have no representation at the manifest level.

5.6.2 Track default_KID Changes

5.6.2.1 DASH / HLS Signaling Overview

As outlined in Section 4.6, a broader range of device compatibility can be achieved by creating a new CMAF Header whenever the encryption key changes. As also noted in Section 4.6, the creation of a new CMAF Header causes the creation of an entirely new CMAF Presentation, meaning the manifest and playlist representation semantics are like those outlined in Section 5.4 Presentation Splicing with key changes across boundaries.

Within [DASH] manifests, the new CMAF Header is signaled by the creation of a new Period at the key rotation point. The new Period would mirror most the semantics of the preceding Period with the notable exceptions of:

- `SegmentBase` / `SegmentTemplate` CMAF Initialization references updated to the new header location and proper presentation timing offsets.
- Encryption `default_KID` signaling for the new encryption key as described in Section 5.3.

```
<Period id="1">
    <AdaptationSet id="1" mimeType="video/mp4" ...>
        <ContentProtection
            schemeIdUri="urn:mpeg:dash:mp4protection:2011"
            value="cbcs"
            cenc:default_KID="939f032e-cd6d-4fbc-9ad0-9d082ba37420"/>
        <Representation id="foo" mimeType="video/mp4" ...>
            <SegmentTemplate
```

```
                timescale="1000000"
                duration="8008000"
                startNumber="1"
                initialization="foo-video.init.mp4"
                media="foo-video.$Number$.m4s" />
        </Representation>
    </AdaptationSet>
</Period>
<Period id="2">
    <AdaptationSet id="1" mimeType="video/mp4" ...>
        <ContentProtection
            schemeIdUri="urn:mpeg:dash:mp4protection:2011"
            value="cbcs"
            cenc:default_KID="476b55c1-3503-4eed-ba45-80a99b44c091"/>
        <Representation id="foo" mimeType="video/mp4" ...>
            <SegmentTemplate
                timescale="1000000"
                startNumber="500"
                initialization="foo-video.init-2.mp4"
                media="foo-video.$Number$.m4s" />
        </Representation>
    </AdaptationSet>
</Period>
```

Figure 5.27 – DASH Manifest with CMAF Header Based Key Rotation

Note: If a production system is not able to align the segments across video and audio where key rotation takes effect a Period boundary will have to be done for video and audio independently. This could have unintended consequences on manifest rendering such as Periods missing Adaptation Sets or Periods with very small durations.

Within [HLS] playlists, the new CMAF Header is signaled prior to the effected segments by the inclusion of an:

- `EXT-X-KEY` tag describing the new encryption information.
- `EXT-X-MAP` tag describing the new CMAF Header.

```
#EXTM3U
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS
#EXT-X-KEY:METHOD=SAMPLE-
AES,KEYFORMAT="com.apple.streamingkeydelivery",KEYFORMATVERSIONS="1",U
RI="skd://939f032e-cd6d-4fbc-9ad0-9d082ba37420"
#EXT-X-MAP:URI="foo-video.init.mp4"
#EXTINF:8.008,
foo-video.0.mp4
#EXTINF:8.008,
foo-video.1.mp4
…
#EXTINF:8.008,
foo-video.499.mp4
```

```
#EXT-X-KEY:METHOD=SAMPLE-
AES,KEYFORMAT="com.apple.streamingkeydelivery",KEYFORMATVERSIONS="1",U
RI="skd://047ae84d-6b05-4afc-af70-b99aa46945a1"
#EXT-X-MAP:URI="foo-video.init-2.mp4"
#EXTINF:8.008,
foo-video.500.mp4
#EXTINF:8.008,
foo-video.501.mp4
…
```

Figure 5.28 – HLS Media Playlist with CMAF Header Based Key Rotation

5.6.2.2 Derivation of HLS Signaling from DASH Signaling

When producing [HLS] Playlists from [DASH] Manifests, the heuristics outlined in Section 5.4.1.2 for presentation splicing derivation are re-used due to the presence of multiple `Period` elements.

5.6.2.3 Derivation of DASH Signaling from HLS Signaling

When producing [DASH] Manifests from [HLS] Playlists, the heuristics outlined in Section 5.4.1.3 for presentation splicing derivation are re-used due to the presence of `EXT-X-DISCONTINUITY` tags.

5.7 Carriage of Track Roles

This section introduces signaling components of [DASH] and [HLS] manifests that are related to the signaling of track roles as defined in Section 4.7.

5.7.1 Track Role Signaling

5.7.1.1 DASH / HLS Signaling Overview

Track roles may be signaled directly within the DASH manifest and HLS Playlists in addition to the track file carriage of the values.

Within [DASH] manifests, roles are described using one or more scheme/value pairs that relate to the track's role. These pairs are signaled by the `Role` descriptor element usually present at the AdaptationSet and/or Representation level of the manifest. The `Role` element contains two attributes that define the role:

- The `Role@schemeIdUri` attribute is used to identify the role scheme.
- The `Role@value` attribute is used to carry the role value being signaled.

When multiple role values must be used to fully describe a track role, multiple `Role` elements are present as siblings in the [DASH] manifest.

The primary role descriptor described by the [DASH] specification is the DASH Role scheme `"urn:mpeg:dash:role:2011"`, which was also adopted as the recommended scheme of the [CMAF] specification and established as the expected scheme for this specification in 4.7.2. For convenience, the values of the DASH Role scheme are repeated non-authoritatively:

Value	Description
caption	Captions.
subtitle	Subtitles.
main	Main media component(s) that is/are intended for presentation if no other information is provided.
alternate	Media content component(s) that is/are an alternative to (a) main media content component(s) of the same media component type.
supplementary	Media content component that is supplementary to a media content component of a different media component type.
commentary	Experience that contains a commentary (e.g., director's commentary) (typically audio).
dub	Experience that contains an element that is presented in a different language from the original (e.g., dubbed audio, translated captions).
description	Textual or audio media component containing a textual description (intended for audio synthesis) or an audio description describing a visual component.
sign	Visual media component representing a sign-language interpretation of an audio component.
metadata	Media component containing information intended to be processed by application specific elements.
enhanced-audio-intelligibility	Experience containing an element for improved intelligibility of the dialogue.
emergency	Experience that provides information, about a current emergency, that is intended to enable the protection of life, health, safety, and property, and may also include critical details regarding the emergency and how to respond to the emergency.
forced-subtitle	Textual information meant for display when no other text representation is selected. It is used to clarify dialogue, alternate languages, texted graphics or location/person IDs that are not otherwise covered in the dubbed/localized audio.

Value	Description
easyreader	Simplified or reduced captions as specified in [United States Code Title 47 CFR 79.103(c)(9)].
karaoke	Textual representation of a song's lyrics, usually in the same languages as the associated song. See SMPTE ST 2067-2.

Where necessary multiple schemes providing the same role description may be used in tandem.

```
<Period id="1">
    <AdaptationSet id="1" mimeType="application/mp4" ...>
        <Role
            schemeIdUri="urn:mpeg:dash:role:2011"
            value="main"/>
        ...
    </AdaptationSet>
    <AdaptationSet id="2" mimeType="audio/mp4" ...>
        <Role
            schemeIdUri="urn:mpeg:dash:role:2011"
            value="alternate"/>
        <Role
            schemeIdUri="urn:mpeg:dash:role:2011"
            value="description"/>
        ...
    </AdaptationSet>
</Period>
```

Figure 5.29 – DASH Manifest with AdaptationSet Role Descriptors

In addition to the `Role` element, the `Accessibility` descriptor element is equivalently defined and used to provide accessibility intention to a role scheme/value pair. The `Accessibility` element may be used with the DASH role scheme, but it is often also used with additional schemes to provide distinctive intention that cannot be captured by the DASH role scheme alone. Such an example is the AudioPurposeCS scheme "`urn:tva:metadata:cs:AudioPurposeCS:2007`" defined in [TV-ANYTIME], which provides additive purpose description to a role. For convenience, the name and values of AudioPurposeCS are repeated below non-authoritatively:

Value	Name
1	Audio description for the visually impaired
2	Audio description for the hard of hearing
3	Supplemental commentary

Value	Name
4	Director's commentary
5	Educational notes
6	Main programme audio
7	Clean feed – no FX mix

For more information on usage and additional placement locations refer to [DASH] 5.8.4.2 Role / [DASH] 5.8.4.3 Accessibility and for details on the DASH Role scheme values, refer to [DASH] 5.8.5.5 DASH role scheme.

Within [HLS] playlists, roles are described through a combination of Media Characteristic Tags (MCTs) and `EXT-X-MEDIA` tag attributes:

- The `CHARACTERISTICS` attribute optionally contains one or more comma separated MCTs that describe the characteristics of the associated track.
- The `FORCED` attribute is used for signaling forced subtitle renditions.

Several public Media Characteristic Tags are defined by [HLS] and the Apple AVFoundation library, for convenience they are repeated non-authoritatively:

MCT	Description
`public.subtitles.forced-only`	Indicates that a track presents only forced subtitles.
`public.accessibility.describes-music-and-sound`	Indicates that a track includes legible content, such as transcriptions of spoken dialog and descriptions of music and sound effects, in the language of its specified locale.
`public.accessibility.describes-video`	Indicates that a track includes audible content that describes the visual portion of the presentation.
`public.translation.dubbed`	Indication that a track contains audio language or dialect translation of the original content.
`public.easy-to-read`	Indicates that a track provides legible content that's edited for easy reading.

MCT	Description
public.auxiliary-content	Indicates that a track includes content its author indicates is auxiliary to the asset's presentation.
public.main-program-content	Indicates that a track includes content its author indicates is essential to the asset's presentation.
public.accessibility.transcribes-spoken-dialog	Indicates that a track includes legible content that transcribes spoken dialog.

In addition to public MCTs, private MCTs may be defined and used within the CHARACTERISTICS attribute.

```
#EXTM3U
#EXT-X-VERSION:7
#EXT-X-INDEPENDENT-SEGMENTS

#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="audio-
en",NAME="English",LANGUAGE="en-
US",AUTOSELECT=YES,DEFAULT=YES,CHANNELS="2",URI="aac-en.m3u8"
#EXT-X-MEDIA:TYPE=AUDIO,GROUP-ID="audio-es",NAME="English
(Description)",LANGUAGE="es",AUTOSELECT=YES,DEFAULT=YES,CHANNELS="2",U
RI="aac-es.m3u8",CHARACTERISTICS="public.accessibility.describes-
video"

#EXT-X-STREAM-
INF:CODECS="avc1.640020,mp4a.40.2",RESOLUTION=960x540,FRAME-
RATE=30.00,AUDIO="audio",...
h264-540.m3u8
#EXT-X-STREAM-
INF:CODECS="avc1.640020,mp4a.40.2",RESOLUTION=1280x720,FRAME-
RATE=30.00,AUDIO="audio",...
h264-720.m3u8
```

Figure 5.30 – HLS Multivariant Playlist with EXT-X-MEDIA Characteristics

For more information on usage refer to [HLS] 4.4.6.1 EXT-X-MEDIA and the AVMediaCharacteristic struct of AVFoundation.

As the [CMAF] specification recommends the usage of the DASH Role scheme for in media carriage of roles, this specification assumes the usage of it, and no official mapping exists from the DASH Role scheme to the well-known HLS Media Characteristics Tags, one is provided:

DASH Role Scheme value	Media Characteristic Tag	Notes
caption	public.accessibility.transcribes-spoken-dialog	None
subtitles	N/A	Signaled by TYPE=SUBTITLE on the EXT-X-MEDIA tag instead of MCT.
main	public.main-program-content	While equal values exist, they are generally not included and instead implicit from the lack of value.
alternate	N/A	Implicitly signaled by other MCTs or attributes present on the EXT-X-MEDIA tag.
supplementary	public.auxiliary-content	Further description provided by additional MCTs or attributes on the EXT-X-MEDIA tag.
commentary	N/A	Explicit commentary signal not available, but NAME attribute can aide UI based identification.
dub	public.translation.dubbed	None.
description	public.accessibility.describes-music-and-sound, public.accessibility.describes-video	MCT(s) used depends on track type.
sign	N/A	No equivalent.

DASH Role Scheme value	Media Characteristic Tag	Notes
`metadata`	N/A	No equivalent.
`enhanced-audio-intelligibility`	`public.accessibility.enhances-speech-intelligibility`	MCT value is proposed and not officially designated yet.
`emergency`	N/A	No equivalent.
`forced-subtitle`	N/A	Signaled by `FORCED=YES` on `EXT-X-MEDIA` tag instead of MCT.
`easyreader`	`public.easy-to-read`	None.
`karaoke`	N/A	No equivalent.

5.7.1.2 Derivation of HLS Signaling from DASH Signaling

When producing [HLS] Multivariant Playlists from [DASH] Manifests, the signaling for each track in the [DASH] Manifest, based on its immediate Representation element and its containing switching hierarchy, is evaluated for one or more [HLS] signaling attributes to attach:

- For any main tracks:
    - Defined by:
        - A `Role` descriptor with `Role@value` equal to `main`, or
        - The lack of an explicit `Role` descriptor
    - If the main track represents the primary media component type, no additional signaling is required on the `EXT-X-STREAM-INF` tag.
    - If the main track represents an additional media component type,
        - Set the `DEFAULT` attribute value to `YES` on the `EXT-X-MEDIA` tag, and
        - Optionally add the MCT `public.main-program-content` to the `CHARACTERISTICS` attribute of the `EXT-X-MEDIA` tag.
- For any text tracks associated with a `Role` descriptor:
    - Where `Role@value` is `caption`,

    - Add the MCT `public.accessibility.transcribes-spoken-dialog` to the `CHARACTERISTICS` attribute of the `EXT-X-MEDIA` tag.
- For any text tracks associated with a `Role` descriptor:
    - Where `Role@value` is `subtitles`,
    - Set the `TYPE` attribute value to `SUBTITLE` on the `EXT-X-MEDIA` tag.
- For any tracks associated with a `Role` descriptor:
    - Where `Role@value` is `supplementary`,
    - Add the MCT `public.auxiliary-content` to the `CHARACTERISTICS` attribute of the `EXT-X-MEDIA` tag.
- For any tracks associated with a `Role` descriptor:
    - Where `Role@value` is `dub`,
    - Add the MCT `public.translation.dubbed` to the `CHARACTERISTICS` attribute of the `EXT-X-MEDIA` tag.
- For any text tracks associated with a `Role` descriptor:
    - Where `Role@value` is `description`,
    - Add the MCT `public.accessibility.describes-music-and-sound` to the `CHARACTERISTICS` attribute of the `EXT-X-MEDIA` tag.
- For any audio tracks associated with a `Role` descriptor:
    - Where `Role@value` is `description`,
    - Add the MCT `public.accessibility.describes-video` to the `CHARACTERISTICS` attribute of the `EXT-X-MEDIA` tag.
- For any tracks associated with a `Role` descriptor:
    - Where `Role@value` is `enhanced-audio-intelligibility`,
    - Add the MCT `public.accessibility.enhances-speech-intelligibility` to the `CHARACTERISTICS` attribute of the `EXT-X-MEDIA` tag.
- For any tracks associated with a `Role` descriptor:
    - Where `Role@value` is `forced-subtitle`,
    - Set the `FORCED` attribute value to `YES` on the `EXT-X-MEDIA` tag.
- For any tracks associated with a `Role` descriptor:
    - Where `Role@value` is `easyreader`,
    - Add the MCT `public.easy-to-read` to the `CHARACTERISTICS` attribute of the `EXT-X-MEDIA` tag.

- For any tracks associated with a `Role` descriptor:
  - Where `Role@value` is one of:
    - `alternate`
    - `commentary`
    - `sign`
    - `metadata`
    - `emergency`
    - `karaoke`
  - No well-known equivalent [HLS] signaling is available.

    Note: Where well-known MCT values and track attributes do not fully describe the roles of a track, private MCT values and/or custom NAME attribute values may be utilized to facilitate the translation as long as all components within the deployment ecosystem are aware of the private and/or custom values.

Here it is assumed all originating Role descriptors utilize the "`urn:mpeg:dash:role:2011`" scheme. Alternate schemes would need to define their own MCT and attribute signaling equivalents. The descriptors for a track MAY match one or more statements above, in this case multiple MCT values will be added to each track description.

5.7.1.3 Derivation of DASH Signaling from HLS Signaling

When producing [DASH] Manifests from [HLS] Playlists, the signaling for each track in the [HLS] Multivariant Playlist is evaluated for one or more Role descriptors to attach:

- For any track associated with an `EXT-X-STREAM-INF` tag, either:
  - Add a `Role` descriptor with `Role@value="main"`, or
  - Add no explicit descriptor to imply main media.
- For any track associated with an `EXT-X-MEDIA` tag:
  - Where:
    - The `DEFAULT` attribute value is `YES`, or
    - the `CHARACTERISTICS` attribute value contains the MCT `public.main-program-content`
  - Either:
    - Add a `Role` descriptor with `Role@value="main"`, or
    - Add no explicit descriptor to imply main media.
- For any track associated with an `EXT-X-MEDIA` tag:
  - Where the `FORCED` attribute value is `YES`,
  - Add a `Role` descriptor with `Role@value="forced-subtitles"`.
- For any track associated with an `EXT-X-MEDIA` tag:

    - Where:
        - the `TYPE` attribute value is `CLOSED-CAPTIONS`, or
        - the `CHARACTERISTICS` attribute value contains the MCT `public.accessibility.transcribes-spoken-dialog`
    - Add a `Role` descriptor with `Role@value="captions"`.
- For any track associated with an `EXT-X-MEDIA` tag:
    - Where the `TYPE` attribute value is `SUBTITLES`.
    - Add a `Role` descriptor with `Role@value="subtitles"`.
- For any track associated with an `EXT-X-MEDIA` tag:
    - Where the `CHARACTERISTICS` attribute value contains the MCT `public.auxiliary-content`.
    - Add a `Role` descriptor with `Role@value="supplementary"`.
- For any track associated with an `EXT-X-MEDIA` tag:
    - Where the `CHARACTERISTICS` attribute value contains the MCT `public.translation.dubbed`.
    - Add a `Role` descriptor with `Role@value="dub"`.
- For any track associated with an `EXT-X-MEDIA` tag:
    - Where the `CHARACTERISTICS` attribute value contains either MCT:
        - `public.accessibility.describes-music-and-sound`, or
        - `public.accessibility.describes-video`
    - Add a `Role` descriptor with `Role@value="description"`
- For any track associated with an `EXT-X-MEDIA` tag:
    - Where the `CHARACTERISTICS` attribute value contains the MCT `public.accessibility.enhances-speech-intelligibility`.
    - Add a `Role` descriptor with `Role@value="enhanced-audio-intelligibility"`
- For any track associated with an `EXT-X-MEDIA` tag:
    - Where the `CHARACTERISTICS` attribute value contains the MCT `public.easy-to-read`.
    - Add a `Role` descriptor with `Role@value="easyreader"`.

Here it is assumed all generated `Role` descriptors will have `Role@schemeIdUri="urn:mpeg:dash:role:2011"`. Alternate schemes would need their own appropriate `Role@values` but could follow similar tag signal derivations. The signaling for a track MAY match one or more statements; in this case, multiple Role elements will be

generated. In addition to the `Role` element, the `Accessibility` descriptor element is equivalently defined and used to provide accessibility intention to a role scheme/value pair.

While these evaluations establish descriptors at a track level for individual `Representation` elements, a sufficiently advanced conversion process could hoist signaling to the `AdaptationSet` element level assuming all tracks belong to the same CMAF Switching Set.

Annex A HLS Event Data Binding

To enable better interoperability of event data carriage at the DASH manifest and HLS Playlist level, this binding specification is defined for event data whose top-level field structure can be represented in the terms of a CMAF EventMessageBox (‘emsg’) semantics, namely:

- A defined scheme_id_uri.
- An optional value and message_data[] that are defined by the scheme.
- An associated presentation_time.
- An optional id that can be used to deduplicate event messages.
- An optional duration.

Event data matching this structure SHALL be carried in HLS Media Playlist EXT-X-DATERANGE tags where:

- The START-DATE attribute is aligned to the event presentation_time.
- The DURATION attribute is set if the event has a duration.
- The ID attribute is generated to provide a playlist unique value unrelated to the event scheme.
- The CLASS attribute is set to the value “urn:cta:wave:dash-hls:event-daterange”
- The X-EVENT-SCHEME-ID-URI attribute is set to the event scheme_id_uri.
- The X-EVENT-VALUE attribute is set to the event value if one exists.
- The X-EVENT-ID attribute is set to the event id if one exists.
- The X-EVENT-MESSAGE-DATA attribute is a base64 encoding of the event message_data[].

  Note: The application of base64 encoding to the message data is to ensure safe carriage of information within the HLS Playlist which has limited character availability within the attributes of the EXT-X-DATERANGE tag. It is possible that an event scheme may define its message payload to be base64 encoded already, causing the data to be double encoded. While potentially undesirable, this is the intended behavior as this carriage binding must work agnostic to any particular event scheme, allowing interoperable implementations to operate on any scheme without understanding specifics of it.

Consumer Technology Association Document Improvement Proposal

If in the review or use of this document a potential change is made evident for safety, health or technical reasons, please email your reason/rationale for the recommended change to standards@CTA.tech.

Consumer Technology Association
Technology & Standards Department
1919 S Eads Street, Arlington, VA 22202
FAX: (703) 907-7693 standards@CTA.tech

Consumer Technology Association™