DASH-IF IOP-5 V5.0.0 (2021-11)

DASH-IF Interoperability Points;
Part 5: Ad Insertion in DASH

DASH Industry Forum

3855 SW 153rd Dr.
Beaverton, OR 97003 - USA

Email : admin@dashif.org

Important notice

The present document can be downloaded from:
http://www.dashif.org/guidelines

Contents

Tables

Intellectual Property Rights

Disclaimer

This is a document made available by DASH-IF. The technology embodied in this document may involve the use of intellectual property rights, including patents and patent applications owned or controlled by any of the authors or developers of this document. No patent license, either implied or express, is granted to you by this document. DASH-IF has made no search or investigation for such rights and DASH-IF disclaims any duty to do so. The rights and obligations which apply to DASH-IF documents, as such rights and obligations are set forth and defined in the DASH-IF Bylaws and IPR Policy including, but not limited to, patent and other intellectual property license rights and obligations. A copy of the DASH-IF Bylaws and IPR Policy can be obtained at http://dashif.org/.

The material contained herein is provided on an "AS IS" basis and to the maximum extent pe mitted by applicable law, this material is provided AS IS, and the authors and developers of this material and DASH-IF hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of workmanlike effort, and of lack of negligence.

In addition, this document may include references to documents and/or technologies controlled by third parties. Those third-party documents and technologies may be subject to third party rules and licensing terms. No intellectual property license, either implied or express, to any third-party material is granted to you by this document or DASH-IF. DASH-IF makes no warranty whatsoever for such third-party material.

Note that technologies included in this document and for which no test and conformance material is provided, are only published as candidate technologies, and may be removed if no test material is provided before releasing a new version of this guidelines document. For the availability of test material, please check https://www.dashif.org.

Foreword

This Technical Specification (TS) has been produced by the DASH-IF Technical Working Group.

Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in deliverables except when used in direct citation.

Executive summary

Part 5 of DASH-IF IOP v5 provides guidelines for advertisement insertion in a MPEG CMAF based content serving workflow utilizing MPEG DASH as the delivery protocol. This part details the general architecture of an ad enabled content serving workflow along with the conditioning, packaging, and signalling requirements to enable both SSAI and SGAI insertion forms. This part defines guidelines for the creation, storage, and serving of ad content. In addition, this part provides informative references and recommendations for auxiliary content and ad systems involved in the ad insertion architecture.

Introduction

This document is Part 5 of a multipart set of documents, collectively called “IOP V5.0.0”. All the parts are:

1. Overview, architectures and interfaces
2. Core principles and CMAF mapping
3. On-demand services
4. Live and low-latency live services

5. Ad insertion (this document)
6. Content protection
7. Video
8. Audio
9. Text
10. Events
11. Additional functionalities
12. Conformance and reference tools

1 Scope

This document supersedes clause 5 “Ad Insertion in DASH” of the DASH-IF IOP Guidelines version 4.3. The scope remains the same, providing recommendations and guidelines for implementing ad insertion in DASH. To enable greater industry utilization and align better with MPEG CMAF content authoring, the following updates and additions have been made:

- Ad insertion terminology has been aligned with industry standard terminology. The architectures have been updated to better detail and support Server-Side Ad Insertion and Server-Guided Ad Insertion.
- Client-Side Ad Insertion has been explicitly excluded from the scope of the architecture.
- Content conditioning has been aligned with MPEG CMAF requirements.
- Information has been organized around interfaces and functionality within the architecture for readability.

2 References

2.1 Normative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

The following referenced documents are necessary for the application of the present document.

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

[2] DASH-IF IOP-1, "DASH-IF Interoperability Points; Part 1: Overview, Architectures and Interfaces"

[3] ANSI/SCTE 35 2015, "Digital Program Insertion Cueing Message for Cable"

[4] ANSI/SCTE 104 2019A, "Automation System to Compression System Communications Applications Programming Interface (API)"

[5] ISO/IEC 23000-19, "Information technology — Multimedia application format (MPEG-A) — Part 19: Common media application format (CMAF) for segmented media"

[6] "DASH-IF Live Media Ingest Protocol", https://dashif-documents.azurewebsites.net/Ingest/master/DASH-IF-Ingest.html

[7] CTA-5003, "Web Application Video Ecosystem – Device Playback Capabilities"

[8] SMPTE RP 2079-2013, "Digital Object Identifier (DOI) Name and Entertainment ID Registry (EIDR) Identifier Representations", Society of Motion Picture and Technology Engineers

[9] EIDR ID Format – IEDR: ID Format, v1.2, March 2014, http://eidr.org/documents/EIDR_ID_Format_v1.2.pdf

[10] ANSI/SCTE 214-1, "MPEG DASH for IP-Based Cable Services, Part 1: MPD Constraints and Extensions"

[11] ANSI/SCTE 214-3, "MPEG DASH for IP-Based Cable Services, Part 3: DASH/FF Profile"

[12] ANST/SCTE 130-3 2013, "Digital Program Insertion-Advertising Systems Interface Part 3"

[13] SMPTE RP 2092-1-2015, "Advertising Digital Identifier (AD-ID®) Representations", Society of Motion Picture and Television Engineers

2.2 Informative references

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

NOTE: While any hyperlinks included in this clause were valid at the time of publication, DASH-IF cannot guarantee their long-term validity.

The following referenced documents are not necessary for the application of the present document, but they assist the user with regard to a particular subject area.

[i.1] CableLabs Video-On-Demand Content Specification, Version 1.1, available at https://specification-search.cablelabs.com/cablelabs-video-on-demand-content-specification-version-1-1
[i.2] ISO/IEC 1318-1, MPEG-2 Part 1, Systems
[i.3] Reliable Internet Stream Transport (RIST), TR-06-1, October 17, 2018, available at https://www.videoservicesforum.org/download/technical_recommendations/VSF_TR-06-1_2018_10_17.pdf
[i.4] W3C Media Source Extensions, available at https://www.w3.org/TR/media-source/
[i.5] W3C Encrypted Media Extensions, available at https://www.w3.org/TR/encrypted-media/
[i.6] IAB Video Ad Serving Template (VAST), available at https://www.iab.com/guidelines/digital-video-ad-serving-template-vast/
[i.7] IAB Lab Tech Open Measurement SDK, available at https://iabtechlab.com/standards/open-measurement-sdk/
[i.8] ITU-R BS.1770-4, Algorithms to measure audio programme loudness and true-peak audio level

3 Definition of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the following terms apply:

ABR encoder – live encoder that converts a broadcast stream or mezzanine into a ladder of different bit-rate tracks

ad avail processor – logical service that, given cue data, determines the placement of advertisement content within a stream and describes the necessary ad decision service communications

ad content server – server storing the ad content and serving it on a per request basis

ad creative – linear visual and auditory asset that represents the content of an advertisement

ad decision service – functional entity that decides which ad(s) will be shown to the user

ad insertion MPD manipulator – functional entity that proxies a DASH MPD and may change it to insert the ad creative in the streaming presentation. It may also embed other ad related metadata, or remove ad related metadata in the MPD

ad pod – location or point in time where one or more ad slots may be scheduled for delivery; same as ad break, avail, and placement opportunity; pre-, mid-, and post- prefix may be used to denote pod location relative to content as before, during, and after respectively

ad reporting server – functional entity for collecting viewer impressions of advertisement content

ad slot – single ad creative that is one of possibly many others that make up an ad pod

CDN node – functional entity returning a segment on request from DASH client. There are no assumptions on location of the node

client-side ad insertion – ad serving architecture which has the client application load advertisements in a secondary player at externally described ad opportunity points

CMAF packager – functional entity, often residing with the ABR Encoder, which packages the adaptive bit-rate tracks into CMAF tracks

DASH ad resolver – functional entity which returns one or more remote elements on request from DASH client

DASH access client – client consuming the DASH stream, possibly also contains functionality for client-side ad insertion and viewer impression reporting.

DASH ad resolver – functional entity which returns one or more ad creatives in a dash formatted construct on request from a DASH Access client

DASH packager – functional entity that processes conditioned content and produces media segments suitable for consumption by a DASH client. This entity is also known as fragmenter, encapsulater, or segmenter

DASH-IF ad content – Content that follows specific restrictions and requirements according to this specification to be independently produced and inserted into well-formated main content by simple MPD manipulation processes

default content – placeholder content within main content that is shown when a paid placement is not inserted

MPD generator – functional entity returning an MPD on request from DASH client. It may be generating an MPD on the fly or returning a cached one.

origin – functional entity that contains all media segments indicated in the MPD, and is the fallback if CDN nodes are unable to provide a cached version of the segment on client request

placement opportunity – marker within content that denotes where paid placements may be inserted

reference playback platform – reference platform for playback (e.g. HTML-5 MSE/EME)

server-side ad insertion – ad serving architecture that interleaves content and ad assets prior to the stream reaching the client

server-guided ad insertion – ad serving architecture that fully describes ad opportunities within content prior to the stream reaching the client, but has the client resolve opportunities as needed

service provider – entity that actively distributes a streaming service to an end consumer

splice point – point in media content where its stream may be switched to the stream of another content, e.g. to an ad

tracking event – data payload associated with an ad creative that is emitted by an application when a specific time point or criteria is met during the creative playout

3.2 Symbols

For the purposes of the present document, the following symbols apply:

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ABR	Adaptive Bit Rate
ADS	Ad Decision Service
CSAI	Client Side Ad Insertion
CDN	Content Delivery Network
CMAF	Common Media Application Format
CTA	Consume Technology Association
DASH	Dynamic Adaptive Streaming over HTTP
DASH-IF	DASH Industry Forum
EIDR	Entertainment ID Registry
HTTP	HyperText Transport Protocol
IAB	Interactive Advertising Bureau
IF	Interface
IOP	InterOperability Point
ISO	International Standards Organization
MPD	Media Presentation Description
MPEG	Motion Pictures Experts Group
RIST	Reliable Internet Stream Transport
SCTE	Society of Cable Telecommunications Engineers
SGAI	Server-Guided Ad Insertion
SMPTE	Society of Motion Picture and Technology Engineers
SSAI	Server-Side Ad Insertion
TS	Transport Stream
VAST	Video Ad Serving Template
VOD	Video On-Demand
WAVE	Web Application Video Ecosystem

4 Introduction

4.1 Use Cases and Scenarios

4.1.1 Overview

This clause provides an overview of guiding use cases considered in the context of ad insertion for DASH. The initial focus is on live use cases addressed in clause 4.1.3 together with the transition issues in clause 4.1.7.

In future version of this document, the remaining use cases will be addressed. However, the tools documented in this clause may very well be used for ad insertion for all documented use cases.

4.1.2 VoD

In this case content is statically defined and made available on demand to clients. Ad insertion takes place at pre-defined placement opportunities within the content. Opportunities are located at conventional pre-, mid-, and post-roll positions within the content.

No restriction is placed on the duration of the inserted ads. Service providers may choose to fill the opportunities when the client first requests content and/or when the client playout approaches the opportunity location. Service providers may also choose to skip an opportunity, in which case content will seamlessly continue.

If possible, content should be preconditioned such that segment boundaries are created at placement opportunities.

4.1.3 Live

In this case content is being made periodically available to clients as part of a live event. Placement opportunities are signalled by the content author via in-band cues such as SCTE-35. Service providers may have the right to replace a subset or all of the placement opportunities that occur.

Opportunities will have an explicit expected duration announced with them and may come with little to no pre-warning. Inserted advertisements will replace in stream content and should exactly match the expected opportunity duration to avoid delaying the main content.

While opportunities are generally expected to match the announced duration, in practice opportunities may be terminated early by the content author in response to the occurring event. In this case, the main content will take priority and the inserted advertisement will be cut short at the point of in-stream opportunity termination.

In addition to early termination, opportunities may be extended by the content author in response to the occurring event. In this case, the service provider may elect to return to the main stream and use the original in stream content for the remainder of the break or treat the extension as a new opportunity and fill the announced extended duration.

Service providers may choose to skip a replacement opportunity entirely, in which case the original in stream content will be played instead.

If in-band cues are used to signal opportunities, the content encoding should produce exact segment boundaries at the cue points.

4.1.4 Recorded Live

In this case content is a capture of a live stream that is made available on demand to clients. Placement opportunities are the same that occurred during the original live event. Service providers may have the right to replace a subset or all of the placement opportunities that occur.

Opportunities have an explicit duration and default content associated with them. Inserted advertisements will replace the default content and may vary in duration from the original content.

Service providers may choose to skip a replacement opportunity, in which case the default content will be played instead. Service providers may also choose to remove a placement opportunity, in which case the content before the opportunity will seamlessly transition to the content after the opportunity.

4.1.5 Pre-Roll into Live

In this case a service provider desires to present an advertisement prior to entering a live stream. The advertisement is a static asset that is available on demand to clients and the live stream is being made periodically available to clients as part of a live event.

The advertisement may be of any duration desired and is not associated with any conditioning or markers in the live stream.

Following the playout of the advertisement, the client will join the live stream and no longer be able to access the original advertisement.

4.1.6 Obfuscation of Inserted Ads

In this case a service provider wishes to present an advertisement with a content stream but does not wish for the advertisement to be detectable by the client. To accomplish this the advertisement may be stitched into content prior to packaging and manifest generation such that there is a single asset produced containing the stitched assets.

This use case is not currently in the scope of this document as the single asset will result in less interoperability challenges. However, the DASH-IF Working Group is continuing to study effective obfuscation methods and practices within DASH and will provide information in future editions of this document.

4.1.7 Changes that may happen at Transitions of Main Content and Ads

A complex issue in the playback of ad content in combination with main content is the transition between the two contents. This transition should happen in a smooth and seamless manner such that the user does not observe discontinuities, quality changes, audio glitches, rebuffering or other artefacts. DASH provides different signaling mechanisms to indicate how content is offered. In many times, it is then a question on the capability of the underlying playback platform, whether it can handle such content in a smooth and seamless manner or if issues and problems are expected to occur. At such splice points, different issues may happen in general, some of them listed below:

- Timeline discontinuities: in order to avoid rewriting content, the inserted content may not follow the timeline of the main content. However, this can be handled in DASH at Period boundaries.
- Overlaps or possibly even gaps or of content on the main media timeline: The content may not exactly be matching the envisaged insertion instructions and hence content may be overlapping at the splice points or there may be gaps. DASH permits signalling of such properties and playback platforms can handle the playback of content with such properties.
- Encryption and key changes: In case of DRM protected content, changes of the encryption or of keys may happen at splice points. DASH permits signalling of such properties and playback platforms can handle the playback of content with such properties.
- Codecs changes: Ads may be prepared with different codecs than the main content. This may result in complex codec change operations and not all platforms can handle such operations. DASH permits signalling of such changes, but also allows playback platforms to identify their capabilities, whether they can handle such changes or not.
- Codec profile/level changes: Similarly, to the above, ads may be prepared with different codec profiles or levels. Again, DASH permits signalling of such changes, but also allows playback platforms to identify their capabilities, whether they can handle such changes or not.
- Signal changes (HDR/SDR, 4K/HD, Stereo/5.1): Ad content may be offered with different signal properties, for example the resolution of the video may changes, the color space or transfer characteristics may changes, or in audio, the channel configuration may change. Again, DASH permits signalling of such changes, but also allows playback platforms to identify their capabilities, whether they can handle such changes or not.
- Addition or removal of a track (e.g. a language, subtitle): At ad or program boundaries, certain tracks or sub-assets may not be available, for example a specific language may not be available, the content may not provide subtitles, or even the offering in a certain format or codec may not be available. Again, DASH permits signalling of such changes, but also allows playback platforms to identify their capabilities, whether they can handle such changes or not.

- Audio volumes may change. Audio volume may be measured in units such as K-weighted relative to full scale [i.8] Regulations may determine that certain volumes may not be exceeded for broadcast. In addition, transitions in loudness may affect the overall user experience. These aspects should be accounted for when inserting advertisement content and at period boundaries.

This specification addresses three aspects in the context of the above:

1) The signalling of the DASH formatted content at splice points on what changes may happen

2) Certain requirements on DASH formatted content in order to support playback on a majority of devices

3) The ability of signalling the required capabilities for a playback platform in order to seamlessly playback the content.

4.2 Architectures

In the context of DASH-IF guidelines, primarily two architectures are considered. In the Server-Side Ad Insertion (SSAI) architecture, the ad is inserted in the network before reaching the DASH Client. In the Server-Guided Ad Insertion (SGAI) architecture, information about ad placement and resolution is inserted in the network, but final resolution is done on demand by the DASH client. The architectures share a significant amount of the functions and interfaces documented in Figure 1.

NOTE: Client-Side Ad Insertion (CSAI) is a third architecture for ad insertion which has the ad opportunities described and loaded external from the main player in a secondary player. These guidelines will not cover this architecture as the operations in it occur completely outside of the DASH Client.

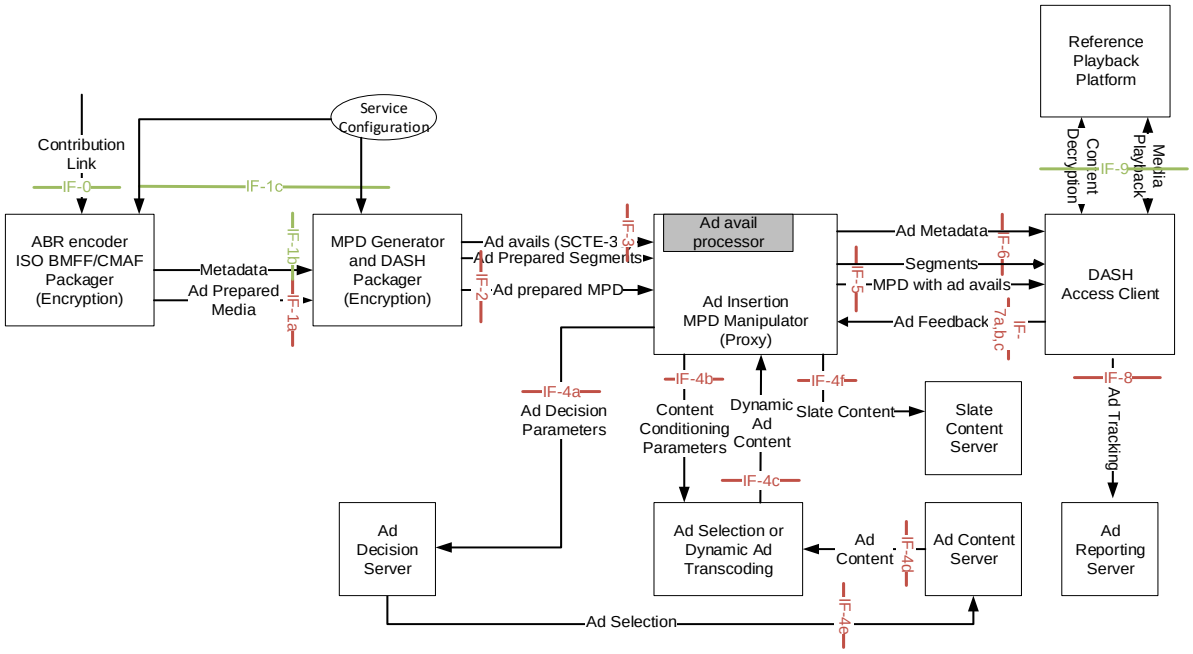

Figure 1 DASH-IF Ad Insertion Architecture

In this document, requirements and recommendations are provided for different interfaces. The main focus of the work are the interfaces to and from the DASH client. However, network interfaces and functions are also discussed as they impact the processing in certain functions.

An overview of the functions and interfaces are provided in clause 4.3.

4.3 Overview on Interfaces and Functions

The Ad Insertion architectures start with the ingest of an input stream over IF-0 which is processed by an ABR Encoder and output as well-formed CMAF content over the IF-1 interface. A DASH Packager / MPD Generator uses IF-1 input to generate a conformant DASH content presentation that is sent over IF-2 and additional opportunity metadata that is sent (IF-3).

An Ad Insertion MPD Manipulator uses the inputs of IF-2 and IF-3 to generate a DASH presentation that is a mixture of content and advertisements. In the SSAI architecture, the manipulator uses IF-4 to ask an Ad Decisioning / Content Server to provide advertisement placements for the content stream, possibly utilizing client sourced parameters from IF-7, before generating the final DASH MPD for IF-5 which contains metadata about the inserted ads via IF-6. In the SGAI architecture, the manipulator does not immediately use IF-4, instead it embeds opportunity information from IF-3 into the DASH MPD IF-5 output so that the DASH Client may later use IF-7 to retrieve the proper ad placements by sending this opportunity information along with client source parameters to the manipulator.

The DASH Client utilizes the reference media pipeline provided by IF-9 to perform seamless playout of the mixed content and ad presentation obtained via IF-5. Ad measurement and tracking is enabled in the client by IF-8 utilizing the ad metadata embedded as part of IF-6.

In Table 1 the interfaces defined are detailed with clause references and some example instantiations. Each interface clause will provide an informative overview of said interface and where aspects of the interface fall within the scope of this document, normative requirements will be provided.

Table 1 Interfaces identified in the ad insertion architecture, example instantiations and references within the document

Interface	Function	Example Instantiations	Reference
IF-0	ABR Stream Source	MPEG-2 TS, RIST	5.1**Error! Reference source not found.**
IF-1a	Packager Ingest Media	DASH Ingest interface 1, azure smooth ingest, CMAF	5.2
IF-1b	Packager Ingest Metadata	DASH Ingest interface 1 metadata, azure smooth ingest metadata	5.2
IF-1c	Configuration Parameters	See for example DASH-IF IOP v4.3 and LL-DASH extensions	5.2
IF-2	Content Preparation	MPEG DASH, IOP v4.3.	5.4
IF-3	Ad Avail Signalling	SCTE-214.X, CableLabs	5.5
IF-4a	Ad Decisioning Parameters	This specification	5.6.2
IF-4b	Ad Content Conditioning Parameters	This specification	5.6.3
IF-4c	Dynamic Ad Content Format	This specification	5.6.7
IF-4d	Ad Storage Format	This specification	5.6.5
IF-4e	Ad Selection Result	VAST, SCTE-130	5.6.4
IF-4f	Slate Content	This specification	5.6.6
IF-5	MPD and Segments with Ad Placement	MPEG DASH, IOP v4.3, this specification	5.7
IF-6	Ad Metadata Signalling	MPEG DASH, IOP v4.3	5.8
IF-7a	Client Sourced Ad Decisioning Parameters	MPEG DASH, IOP v4.3	5.9.2
IF-7b	Client Sourced Content Conditioning Parameters	Request Headers/Queries, undefined	5.9.3
IF-7c	Late Bound Remote Resolution	MPEG DASH, IOP v4.3	5.9.4
IF-8	Ad Tracking and Measurement	VAST, Open Measurement SDK	5.10
IF-9	Reference Media Playback and Content Decryption	HTML-5 video, MSE, EME, CTA WAVE Device Playback Platform	5.3

5 Interfaces

5.1 IF-0: ABR Stream Source

5.1.1 General

The formatting and delivery of media input to the ABR encoder is described by IF-0. The ad insertion architectures in this document are agnostic to the choice of this interface instantiation and as such information in this clause shall be considered informational.

Example interface instantiations may differ depending on the type of media input being supplied to the architecture. For example, a VOD workflow may utilize a mezzanine delivery format such as the CableLabs Video-On-Demand Content Specification [i.1], while a Live workflow may utilize a contribution feed delivery format such as MPEG-2 TS [i.2], RIST [i.3].

For any instantiation, it is usually beneficial for the media input to contain descriptive metadata about the media input such that the ABR encoder may provide conditioning of the encoded output and pass-through said information to components later in the ad insertion streaming architecture. As the format of descriptive metadata may be workflow specific, the examples provided below should be considered informational only.

5.1.2 Live Workflow Descriptive Metadata

In a Live workflow, the descriptive metadata may consist of program, segmentation, and splicing information, we will refer to this information as *broadcast events*. Examples of what *broadcast events* signal are program start/end, chapter start/end, interstitial, distributor start/end, provider break start/end, content identification, and many others. SCTE-35 [3] and/or SCTE-104 [4] are examples of standards used to insert such broadcast events aligned with the media presentation in IF-0. In Figure 3, we show a segmentation of a live input based on SCTE-104/35.

Figure 3 shows a live broadcast with segmented broadcast information based on broadcast events. In this case broadcast events are used to segment and can optionally be used to signal ad breaks. Nevertheless, more information is carried about the broadcast streams. The placement opportunities are shown in green. For more information relating to the commands supported we refer to SCTE-35 [3].

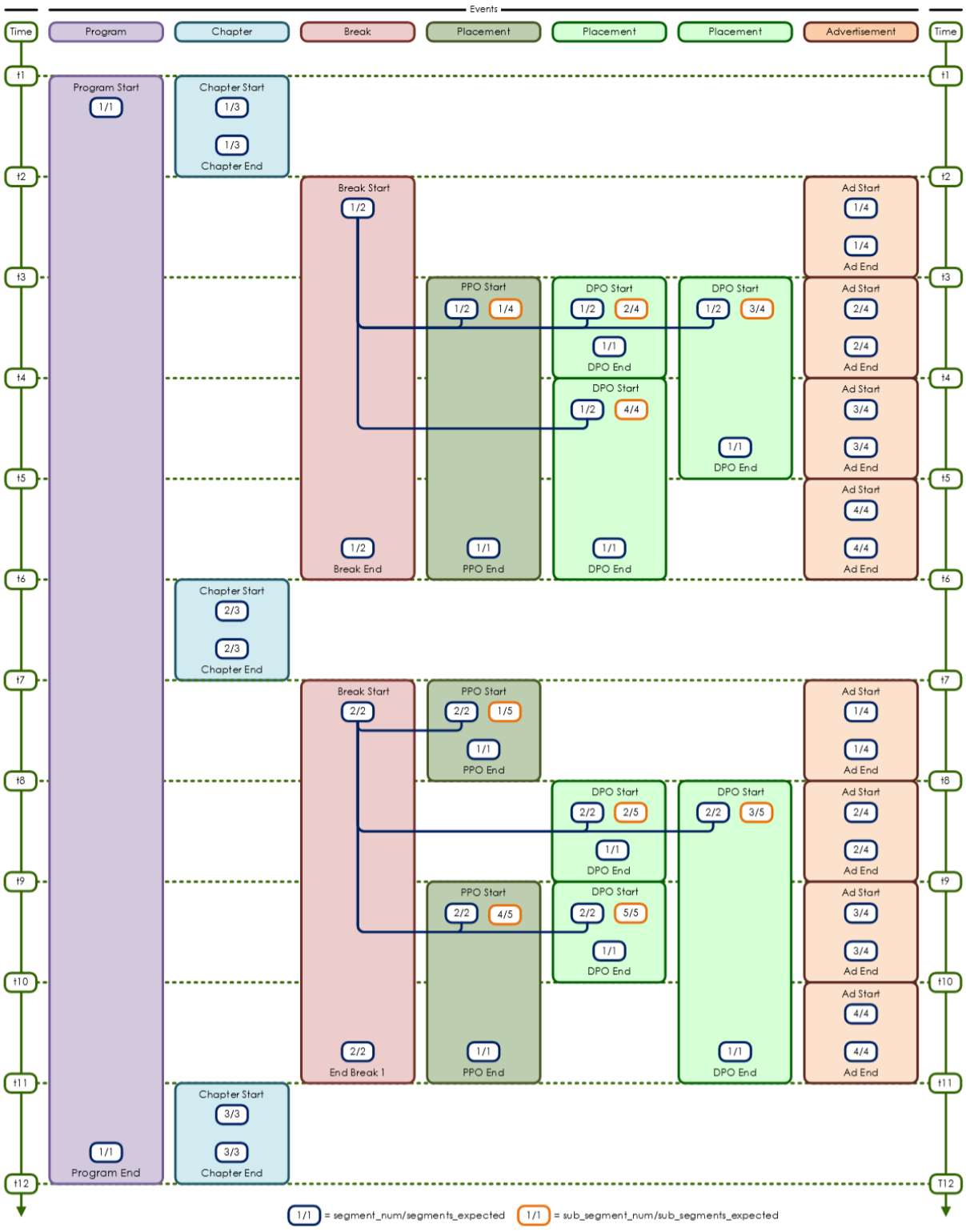

Figure 2 Segmented Live Broadcast with Broadcast Events from SCTE-35 [3]

5.1.3 VOD Workflow Descriptive Metadata

In a VOD workflow, content is delivered to a service provider by a content provider as a package of various assets and metadata that make up the full description of the content. This package contains mezzanine assets that streamable assets may be produced from, but may also contain still image cover art, promotional assets, and preview trailer assets. The metadata provided alongside assets include basic information such as title, genre, and rating, but also includes advanced metadata such as chapter locations, distribution subscriber requirements, and distributor ad preservation requirements.

One format of this package is described by the CableLabs Video-On-Demand Content Specification [i.1], which we defer to for further information about package structure and data.

5.1.4 Abstracted Model

Media provided in mezzanine or ingest is assumed to have a continuous media time and the timestamp of the media carries through the ABR encoder for each media type as shown in Figure 3. In addition, splice points are defined and at these splice points, at a specific media time t_{splice}, the ABR encoder is expected to prepare the content accordingly in order to permit splicing. The reason and details of each splice point and the conditions may be carried through but are irrelevant for the media preparation.

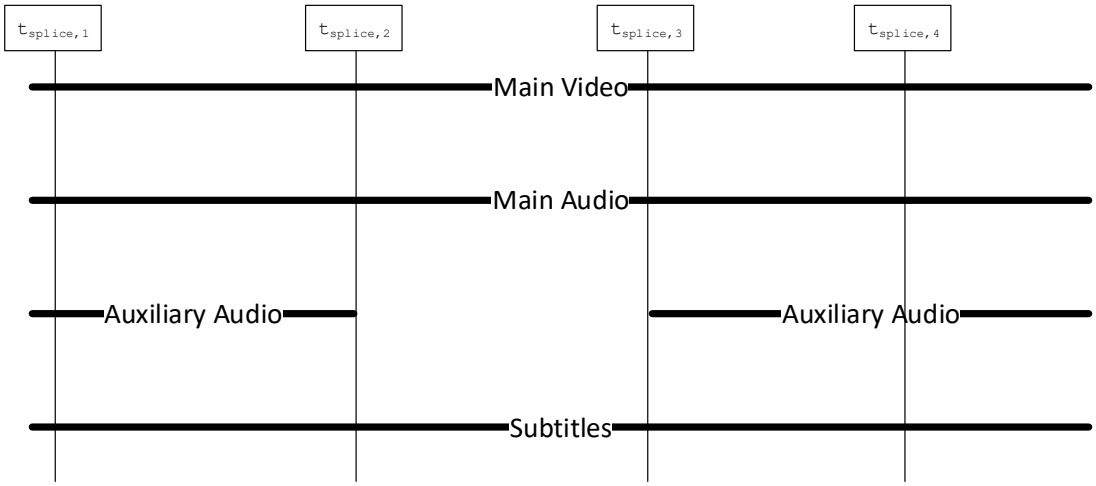

Figure 3 Abstracted Media Model with Splice Points

5.2 IF-1: Packager Ingest

The ABR encoder provides encoded variants of the media input and prepares CMAF conforming headers, chunks and fragments as defined in ISO/IEC 23000-19 [5], organized in CMAF structures such as CMAF Tracks and Switching Sets. The content may also be provided together with an MPD that follows the DASH Core Profile for CMAF content as defined in ISO/IEC 23001-9 [1]. This reflects what is documented with IF-1a in Figure 1.

Those CMAF prepared content is assumed to be properly annotated through metadata. The metadata carries information that can be used by the DASH packager for specific information. This reflects what is documented with IF-1a in Figure 1. A recommended protocol for the combination of the two, IF-1a and IF-1b interface, is the DASH-IF Ingest Spec [6] (CMAF ingest interface).

In addition, the service follows certain service configuration options that are provided by external means. The configuration may include information such as the nominal CMAF fragment duration (DASH segment duration), CMAF chunk duration, number and bitrates in a CMAF Switching Set, codec configurations and media profiles, etc. This reflects what is documented with IF-1c in Figure 1.

The definition of this interface IF-1 is outside of the scope of this document, but in the following clauses several assumptions on the generated media being provided to the DASH packager are taken, pre-dominantly that the encoder produces well-formatted CMAF ISO/IEC 23000-19 [5] conforming content. Note that these assumptions are not a requirement for this specification, but a service provider should understand the downstream system effects if the packager ingest does not follow these assumptions. For example, a transcoding or timeline corrections needs to be done in the DASH packager to meet the output requirements for following interfaces, or a specific addressing scheme may have to be used.

The following assumptions are taken:

- The ABR encoder produces continuous content with a single CMAF Header for each CMAF Track. There may be instances that in between two potential splice points at media times $t_{splice,i}$ and $t_{splice,i+1}$ not all Tracks/Switching Sets are provided. However, at least a minimum set of Switching Sets are always present.

- For those CMAF tracks that are present for the entire program, the media time is continuous, also across splice points. This means that the subset of continuously present CMAF Switching sets of the entire program conforms to a CMAF presentation as defined in ISO/IEC 23000-19, clause 7.3.6.

NOTE: This assumption may be relaxed, but if done, there needs to be a signalling for such a discontinuity. Input on this subject would be welcome. Discontinuities may be signalled in DASH.

- There are three options for content in between two potential splice points at media times $t_{splice,i}$ and $t_{splice,i+1}$.
  - For Option 1 referred to as *"Splice-Conditioned Packaging"*, the following holds:
    - The output of the ABR encoder in between two potential splice points at media times $t_{splice,i}$ and $t_{splice,i+1}$ is CMAF conforming, i.e. it conforms to a CMAF presentation as defined in ISO/IEC 23000-19 [5], clause 7.3.6.
    - The first splice point at $t_{splice,i}$ is the *timeline origin* of all CMAF tracks in the CMAF presentation as defined in ISO/IEC 23000-19 [5], clause 7.3.6.

      NOTE: This does not imply that each splice point resets the timeline. Indeed this would contradict the first assumption above that media is time-continuous.
    - The ABR encoder provides content that can be converted to conforming DASH content, for example consistent CMAF Fragment duration to enable proper usage of DASH Segment duration signalling, bitrate characteristics for signalling in the MPD, event messages, etc.
    - The ABR encoder creates a CMAF Fragment boundary for all CMAF Tracks at $t_{splice,i}$ and resets the CMAF Fragment duration from here on.

      NOTE: This permits Period boundary insertion at $t_{splice,i}$ without modification of the CMAF content.
    - The ABR encoder creates media for *all* samples of *all* CMAF tracks in between $t_{splice,i}$ and $t_{splice,i+1}$ with $t_{splice,i}$ being included and $t_{splice,i+1}$ being excluded.

      NOTE: This permits to create a Period that is fully covered by content.
    - The ABR encoder creates a CMAF Fragment boundary for all CMAF Tracks at $t_{splice,i+1}$ and resets the CMAF Fragment duration from here on.

      NOTE: This permits Period boundary insertion at $t_{splice,i+1}$ without modification of the CMAF content.
  - For Option 2 referred to as *"Splice-Conditioned Encoding"*, the following holds:
    - The ABR encoder creates a SAP type 1 or 2 at $t_{splice,i}$ with T_{SAP} set to $t_{splice,i}$. The placement of the SAP type 1 or 2 may not and typically does not coincide with a CMAF Fragment boundary.
    - The ABR encoder creates a SAP type 1 or 2 at $t_{splice,i+1}$ with T_{SAP} set to $t_{splice,i+1}$. The placement of the SAP type 1 or 2 typically does not coincide with a CMAF Fragment boundary.
  - For Option 3 referred to as *"Splice Point Signalling"*, no specific encoding and packaging is done at the splice points.
    - It may be the case that an exact alignment of a SAP type with the splice point may not be possible, for example due to the codec or format properties. However, additional SAP types may be available, or the media can be accessed quickly by other means, for example by accelerated decoding.
- The ABR encoder passes through timed metadata (from contribution/production feed IF-0) related to the provided descriptive metadata and content conditioning. The following information is needed by the MPD proxy for each splice point:
  - The media time of the splice point $t_{splice,i}$.
  - For each Switching Set that is available in the media stream after the splice point, i.e. it continues or is newly added.

  - The type of the splice point

- The presentation time offset of the splice point, i.e. the media time that aligns with the splice point in scale of the value of the `@timescale` of the Adaptation Set.

- The segment address (number, time) of the Segment that contains the media time of at the splice point.

- The content may be provided at once, for example as part of a VoD Asset generation, or the content may be provided by the ABR encoder on a continuous timeline, for which real-time and media time advance in concurrently.

NOTE: Splice points are defined independent whether you enter or exit the content. It may be also the case that within one content generation workflow, certain media encoding follows option 1 whereas others may follow option 2 or option 3. For example, video may follow option 1, and audio may be encoded based on option 3.

The three options for encoder and packager configuration are shown in Figure 4. In option 1, CMAF Fragment boundaries are aligned with splice points, and in option 2, splice points may occur in the middle of a CMAF Fragment, but are supported by a SAP type 1/2 for random access. In option 3, no SAP type 1 or 2 is necessarily provided at the splice point.

NOTE: As an example, please note that CMAF Fragment #3 in Option 1 may be shorter, or it may be even longer than CMAF Fragment #2 in order to align Splice Points with CMAF Fragment Boundaries.

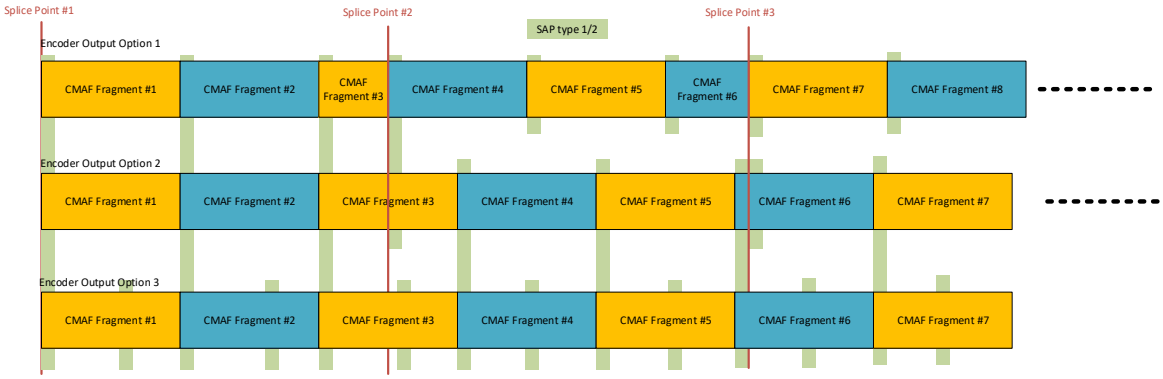

Figure 4 CMAF Encoder and Packager Options

5.3 IF-9: Reference Media Playback and Decryption

Another important assumption in the context of this specification is the availability of a reference playback platform that enables a DASH client to use for media playback and decryption. Without limiting the usage of any DASH player, this assumption permits that content is authored such that platforms with certain restrictions can be used.

The DASH Client interacts with the media pipeline on the reference platform via the IF-9 interface. The definition of this interface is out of the scope of this document, but the general assumption of the DASH-IF IOP is an MSE [i.4] / EME [i.5] reference pipeline.

Furthermore, it is assumed, for interoperability and/or robustness of this interface, that the reference playback platform supports the playback requirements defined by the Consumer Technology Association Web Application Video Ecosystem Project (CTA Wave) Device playback specification [7].

Specifically, in the context of this specification, a playback platform is expected to support playback requirements as documented in clause 8 of CTA-WAVE 5003 [7] for any content conforming to a CMAF Switching Set according to CMAF media profile included in an MPD, namely:

- 8.2 Sequential Track Playback
- 8.3 Random Access to Fragment
- 8.4 Random Access to Time

- 8.5 Switching Set Playback
- 8.8 Playback over WAVE Baseline Splice Constraints
- 8.13 Restricted Splicing of Encrypted Content
- 8.14 Sequential Playback of Encrypted and Non-Encrypted Baseline Content

If a playback platform wants to consume content authored according to encoding and packaging option 2 or 3, ("Splice-Conditioned Encoding" and "Splice Signalling", respectively) as defined in clause 8.2.2 for content conforming to a CMAF Switching Set according to CMAF media profile included in an MPD, is expected to support the following playback requirements as documented in clause 8 of CTA-WAVE 5003 [7].

- 8.9 Out-Of-Order Loading
- 8.10 Overlapping Fragments

Finally, it is assumed that the reference playback platform can be used in order to query proper capabilities such that MPD information can be transformed into capability queries, e.g. if a codec is supported. Device Capability queries is discussed in clause 6.4 of CTA-WAVE 5003 [7].

5.4 IF-2: Content Preparation

The format and requirements of the DASH manifests and segments output by the DASH Packager / MPD Generator for use later in the ad insertion architecture is described by IF-2. The DASH IOP Guidelines provide the general normative requirements on the DASH output and we assume those as a baseline set of requirements. Here we provide further normative requirements for the ad insertion architectures.

Generally, for each known ad splice point, the DASH Packager/MPD Generator should either:

1) Insert a Period boundary at point $t_{splice,i}$. and provide the properties of the splice point for each splice point.
2) Insert sufficient signalling, for example in an Event stream, such that the splice point with its properties is identified and a Period can be easily added at the splice point without reading the Segments. The signalling semantics are provided in clause 5.2.

The recommendation of providing a Period boundary at splice points within the DASH Packager / MPD Generator is made such that the downstream Ad Insertion MPD Manipulator can perform replacements and insertions on the MPD-level only without accessing the content segments.

Should the DASH Packager / MPD Generator not be aware of what splice points are appropriate for ad insertion, the Period boundaries may be omitted and instead be created by the downstream Ad Insertion MPD Manipulator, further details of this operation are provided as part of IF-5.

In the following it assumed that at least one media type (typically video) follows the content generation according to clause 5.2, option 1 ("Splice-Conditioned Packaging") as defined in the CMAF Profile for DASH and furthermore it is assumed:

Each CMAF Fragment generates one DASH Segment

- The content is provided in a live session, i.e. CMAF fragments or CMAF chunks are made available to the DASH packager once completed.
- The minimum splice point advance notice time is known, i.e. the DASH packager gets a pre-notification or ad avail for a splice point that will be added to the media. This allows the DASH Packager to configure the minimum update period of the MPD properly. By this, the DASH client or MPD proxy requests the MPD in high enough frequency such that none of the announced Periods in the MPD are missed. Other workflows are possible, for example MPDs are regularly pushed to the MPD Proxy. For example, in SCTE-35 [3], the advance notice is recommended to provide an advance notice of at least 4s.

Then a DASH packager produces content by (i) generating an initial MPD, and (ii) dynamic operation of the packager including MPD processing/updates and Segment offering.

The initial MPD is generated as follows:

- The CMAF data is mapped to the MPD using the DASH core profile for CMAF content as defined in ISO/IEC 23009-1 [1]. Note that quite many details are defined in this profile.
- For every CMAF Switching Set that is known to be offered in the MPD, an Initialization Set as defined in ISO/IEC 23009-1 ([1], clause 6.3.3) should be added that describes all known static parameters for the CMAF Switching Set, preferably based on the information in the CMAF Master Header (i.e. a CMAF Header that is sufficient to initialize the media pipeline for continuous playback, see CTA WAVE 5003 [7] for details) for this CMAF Switching Set.
  - Every Initialization Set gets assigned a unique id.
  - For every CMAF Switching Set that is not known to be offered on a continuous basis, the `@inAllPeriods` of the Initialization Set is set to `false`.
  - For every CMAF Switching Set that is known to be offered on a continuous basis, the `@inAllPeriods` of the Initialization Set is set to `true` or the attributed is omitted. This is typically the case for a live content that a main audio and video CMAF switching set is provided in a continuous basis.
- The **`MPD`**`@availabilityStartTime` is set to an arbitrary value, for example to 01/01/1970.
- The `@mininumUpdatePeriod` is set sufficiently small such that DASH clients and MPD proxies do not miss Periods created for announced splice points taking into account the minimum splice point advance notice time.
- For each Period, a unique ID for the main content should be carried in an **`AssetIdentifier`** descriptor. Examples for identification schemes are as follows:
  - An EIDR identification scheme, defined by SMPTE RP 2079-2013 [8], is signalled by setting:
    - `@schemeIdUri` to "`urn:eidr`"
    - `@value` to a valid canonical EIDR entry as defined in SMPTE RP 2079-2013 [8]
  - In the absence of other identification schemes, the DASH-IF scheme may be used defined as follows:
    - `http://dashif.org/guidelines/v5/asset-id` (previously `@schemeIdUri` to "`urn:org:dashif:asset-id:2014`")
    - `@value` to a MovieLabs ContentID URN ([9], 2.2.1)

  NOTE: Based on the MPEG MPD restrictions, at most a single Asset Identifier may be added per Period.
- The initial MPD follows the main content in Table 2.

For every splice point *i* at time $t_{splice,i}$, if a Period is generated, the following applies:

- If it is the first Period in the presentation and the media is "starting" to be produced:
  - `@start` of the Period is set to *NOW* - `@availabilityStartTime` with some possible margins to address different Segment availability times, for example due to publication delay on a CDN.
- If it is not the first Period in the presentation:
  - `@start` of the Period is set to the sum of the value of **`Period`**`@start` of the previous period and the interval between the two splice points ($t_{splice,i}$ - $t_{splice,i-1}$)
  - Period continuity is signaled across all Adaptation Sets that are continuing across the Period boundary. Preferably the same signaling and track structure is used.
- Every available CMAF Switching Set in the CMAF Presentation is mapped to one Adaptation Set using the DASH Core Profile for CMAF content as defined in ISO/IEC 23009-1 [1]. In addition, the following restrictions apply for each Adaptation Set:
  - The `@presentationTimeOffset` is set to $t_{splice,i}$ normalized by the `timescale`.

- For Option 1 *"Splice-Conditioned Packaging"*:
  - The value of `@eptDelta` is 0 and should therefore be absent.
  - The `@startNumber` or **`SegmentTimeline`** is set such that it references the first Segment in the Period after the splice point.
- For Option 2 and 3 (which is basically a generalization to option 1), all details are provided in the DASH Core Profile for CMAF content.
- Period continuity should be signaled to indicate, which Adaptation Set follows continuously the previous one and for Representations with the same `@id` within continuous Adaptation Sets, the continuity is signaled.
- If the CMAF Switching Set is identical to one for which an Initialization Set was set, then all parameters from the Initialization Set are copied into this Adaptation Set and the `@initializationRefId` is set to the one referred to.

NOTE: This operation does not consider aspects such as inconsistent/variable segment durations within a CMAF Presentation, upstream losses or errors, etc. Any of such occurrences may result in additional Periods that may be added according to the DASH-IF IOP guidelines.

The mapping is shown in Figure 5.

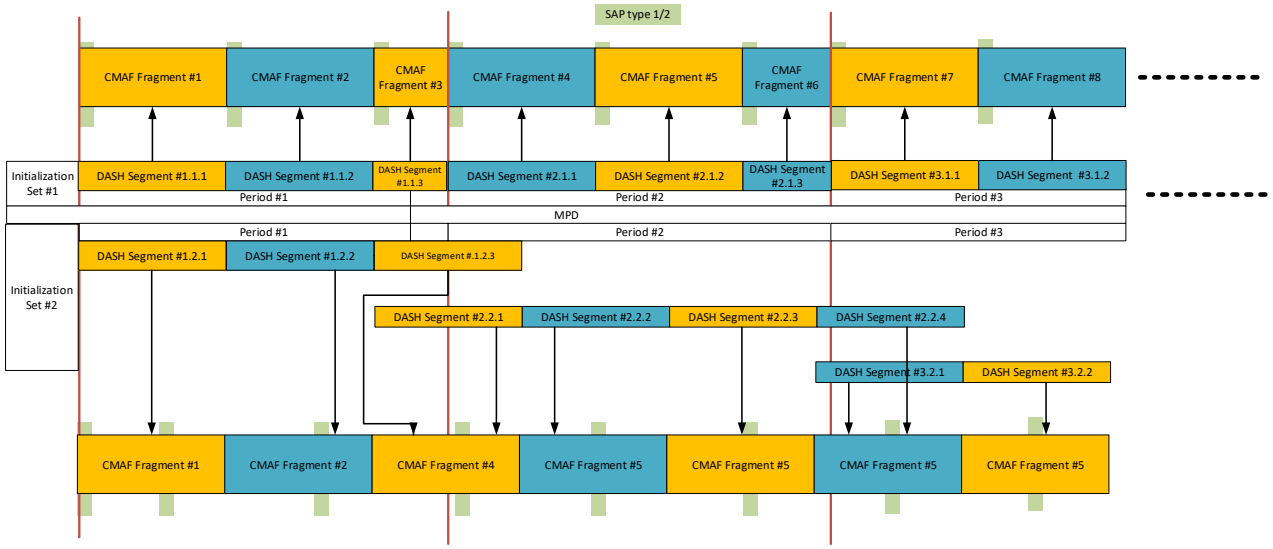

Figure 5 CMAF Fragment to DASH Mapping for Option 1 and 3

Table 2 defines the relevant parameters for the main content MPD.

Table 2 DASH-IF Main live content MPD

Element or Attribute Name	Use	Description
`MPD`		Provides the requirements for DASH-IF main content. Any not specified value is identical to what is provided in ISO/IEC 23009-1 ([1], clause 5.3.1).
`ServiceDescription`	0 … N	
`Latency``@target`	O	A target latency may be provided
`@profiles`	M	should include a profile indicator for the DASH CMAF profile `"urn:mpeg:dash:profile:cmaf:2019"` .
`InitializationSet`	0 … N	may be present and the `@inAllPeriods` may be set to true in order to express the continuity of content across Period boundaries.

Element or Attribute Name	Use	Description
ProgramInformation	0…N	This should be used to describe information about the main content. More details may be provided.
Period	1 … N	One or more Periods shall be present. Provides the requirements for main live content. Any not specified value is identical to what is provided in ISO/IEC 23009-1 ([1], clause 5.3.2).
@xlink:href	R	Shall be absent.
@xlink:actuate	R	Shall be absent.
@start	M	Shall be present.
AssetIdentifier	0…1	This should be used to provide an explicit main content identifier
EventStream	0...N	specifies an event stream. There are two types of Events: • Event Streams terminating in MPD Proxy/Ad processor. • Events Streams continuing even if an ad is inserted. The handling of these Events is decided by the proxy, for more details see below.
@presentationTimeOffset	OD	is needed for multi-period split events across Period boundaries. For details refer to ISO/IEC 23009-1 [1].
AdaptationSet	1...N	At least one Adaptation Set shall be present.
@xlink:href	R	Shall be absent
@xlink:actuate	R	Shall be absent
SegmentBase@presentationTimeOffset	OD default: 0	shall be set to the correct value of the presentation time of the Adaptation Set at the start of the Period, if the presentation time is not equal to 0.
@contentType	M	Shall be present
SegmentList	0	Shall be absent
Representation	1 … N	specifies a Representation. At least one Representation element shall be present in each Adaptation Set. Any not specified value is identical to what is provided in ISO/IEC 23009-1 ([1], clause 5.3.3).
EmptyAdaptationSet	0	Shall be absent
UTCTiming	1 … N	At least one shall be present

Key
For attributes: M=mandatory, O=optional, R=removed
For elements: <minOccurs>…<maxOccurs> (N=unbounded)
Elements are **bold**; attributes are non-bold and preceded with an **@.**

5.5 IF-3: Ad Avail Signalling

5.5.1 Introduction

Opportunity metadata is made up of the original descriptive metadata of the input media related to signalling of ad opportunities and the content segmentation information generated by the DASH Packager. Carriage of opportunity metadata in the presentation output by the DASH Packager / MPD Generator is done via IF-3.

The following normative statements on opportunity metadata carriage are made:

- Opportunity metadata shall be carried through DASH MPD Events.

The requirement of MPD Events over other carriage mechanisms is made such that the downstream Ad Insertion MPD Manipulator can perform insertions without accessing the content segments.

NOTE: This does not exclude the option that equivalent metadata is present inband, but it is not expected that the MPD Proxy make use of such inband information.

While the carriage method is considered normative, the format of the metadata is workflow dependent. Examples of known schemes are provided in the subsequent sub-clauses of this interface. For the purposes of this document, we will assume the carriage of opportunity metadata via SCTE-35 signalling in DASH MPD Events. Other methods and formats may be used, but a service provider should understand the downstream system effects if the packager does not follow this assumption.

Any opportunity metadata used in the context of this specification MUST provide:

- the presentation time (in media time) of the splice point that corresponds to the start of an opportunity.
- either:
  - the guaranteed accurate duration of the opportunity, or
  - an expected duration of the opportunity, if known, and
  - the identifier of a later metadata event that will signal the accurate end of the opportunity

As an example, clause 5.5.2 provides an overview how SCTE-35 can be used to fulfil the requirements of a DASH-IF external opportunity signalling.

5.5.2 Opportunity Signalling via SCTE-35

SCTE-35 describes a set of command messages that can be utilized to describe ad opportunities within a presentation. Typically, the broadcast events for a LIVE presentation are already signaled as SCTE-35 commands which may be directly used, but a VOD workflow may optionally synthesis a series of SCTE-35 commands to describe the conditioning and opportunities in a VOD presentation as well.

The SCTE 214-1 [10] / 214-3 [11] specification defines a set of event schemes for carrying SCTE-35. The appropriate event scheme to use depends on the utilized DASH mechanism, DASH MPD Events may use either `urn:scte:scte35:2013:xml` or `urn:scte:scte35:2014:xml+bin`.

An example of carrying SCTE-35 with MPD Events using the `urn:scte:scte35:2014:xml+bin` scheme is shown in Table 2. The timing described within the SCTE-35 payload provide the **`Event`** properties `@id`, `@presentationTime`, and `@duration` properties. The id may be used to filter out duplicate events. In this example the payload is encoded using Base64 enclosed in the **`Binary`** element per SCTE 35.

For signalling opportunity metadata, it is common place to use a SCTE-35 `time_signal()` splice command that carries a `segmentation_descriptor()` splice descriptor. Alternatively the legacy `splice_insert()` command may be used. In this case the Event duration equals the break duration in the splice, which is the expected duration of the ad break. The `out_of_network` indicator is set to `true` when the splice into an advertisement slot occurs. A break can be terminated early by an additional MPD Event carrying a `splice_insert()` command with the `out_of_network` indicator set to 0 and the splice immediate flag set.

The presentation time of the splice point is provided by the `splice_time()` structure present within the command and is represented in the Event Stream as the value of the `@presentationTime` attribute of the Event.

Opportunity boundaries are signalled by pairs of `segmentation_descriptor()` `segment_type_id` values, for full details on valid pairings see SCTE-35 [3], clause 10.3.3.5. The `segmentation_descriptor()` pair will contain the same value for the `segmentation_event_id` field allowing for them to be properly matched.

If an expected opportunity duration is known, the segment start `segmentation_descriptor()` may provide the expected duration in the `segmentation_duration` field which may be represented in the Event Stream as the value of the `@duration` attribute of the **`Event`** until the final duration is known.

With SCTE-35, the final accurate duration is only established when the segment end `segmentation_descriptor()`, identified by the matching `segmentation_event_id`, is encountered. The segment end descriptor may occur at, before, or after the duration specified by the segment start descriptor. The final duration of the opportunity is calculated as the difference between the splice times of the segmentation pairs and is represented in the Event Stream as the value of the `@duration` attribute of the Event.

Table 3 Example of a SCTE-35 message embedded as an MPD event using SCTE 214

```
<EventStream
  schemeIdUri="urn:scte:scte35:2014:xml+bin"
  timescale="1">
  <Event presentationTime="1540809120" duration="24" id="1999">
    <scte35:Signal>
       <scte35:Binary>
           /DAhAAAAAAAAAP/wEAUAAAfPf+9/fgAg9YDAAAAAAAA/APOv
       </scte35:Binary>
    </scte35:Signal>
  </Event>
</EventStream>
```

NOTE: While the presentation time is 48 years, the presentation time may and typically is adjusted by the `@presentationTimeOffset` to align with the Period start time.

5.6 IF-4: Ad Decisioning and Exchange Interfaces

5.6.1 Introduction

Information about ad content to insert into a presentation is retrieved from the Ad Decision and Ad Content server(s) via the IF-4 interfaces. The request from the Ad Insertion MPD Manipulator for ad content provides all the information needed to perform ad decisioning, including content metadata and opportunity descriptions. The response is then translated by the Ad Insertion MPD Manipulator into the DASH structures detailed in IF-5.

There are many details corresponding to the functions of ad requests and ad decisioning, as such this document identifies 6 different sub-interfaces that outline primary interactions of ad requests and decisioning.

The identified 6 sub-interfaces are:

- Interface IF-4a: Ad Decision request parameters. For details see 5.6.2.
- Interface IF-4b: Content Conditioning request parameters. For details see 5.6.3.
- Interface IF-4c: Recommended Dynamic Ad Content response format. For details see 5.6.7.
- Interface IF-4d: Recommended Ad Content Storage format. For details see 5.6.5.
- Interface IF-4e: Ad Selection Result format. For details see 5.6.4.
- Interface IF-4f: Slate Content. For details see 5.6.6.

These sub-interfaces are described independently to better define the data and interactions that occur as part of an ad insertion architecture, but no assumption is made about the actual architecture and deployment form of the Ad Decision and Ad Content server(s). Instead, this document assumes the following:

- There exists an Ad Decision entity that accepts request parameters of the form described by IF-4a and provides a decision of the form described by IF-4e.
- There exists an Ad Content entity that accepts conditioning parameters of the form described by IF-4b and:
  - Provides the ad content of the form described by IF-4c
  - May also provide filler slate content of the form described by IF-4f
- The Ad Decision and Ad Content entity may be the same system or independent systems.
- The ad content server may or may not be stored in the storage format described by IF-4d. However, this document defines a recommended storage format to interop with CMAF workflows.

5.6.2 IF-4a: Ad Decision request parameters

5.6.2.1 Decisioning Parameters

This document does not specify a specific format, but only discusses some principle approaches. Additional details on specific formats are for further study.

A *decisioning parameter* is a piece of information about the content stream, consumption medium, or end user that is used by the Ad Decisioning Server as part of the advertisement qualification and selection process. The Ad Insertion MPD Manipulator collects and sends this information to the Ad Decisioning Server as part of IF-4a.

The transmission of decisioning parameters is highly integration dependent, but examples of commonly used industry parameters are:

- Content Unique Identifier
- Content Genre
- Content Language
- Service Provider Identifier
- Device Type (TV, SetTop, Mobile, Computer, etc)
- Device Manufacturer
- Device Model
- End User IP Address
- End User Zip Code

The methods for carrying decisioning parameters in the request from the Ad Insertion MPD Manipulator to the Ad Content Server may follow the same guidelines as methods used on requests between the DASH Client and the Ad Insertion MPD Manipulator. Refer to interface IF-7a for further details on these methods.

5.6.2.2 Decisioning Modes

The *decisioning mode* of an Ad Decisioning Server dictates how the server chooses to fulfil ad requests made by a caller. The Ad Insertion MPD Manipulator must specify the decisioning mode for the Ad Decisioning Server to use via IF-4a based on the implemented ad insertion architecture. There are two general modes of ad decisioning that the SSAI and SGAI architectures respectively enable: *stream level decisioning* and *pod level decisioning*.

With *stream level decisioning*, all advertisement opportunities are decided prior to DASH client receiving the stream. A SSAI architecture accomplishes this by having the Ad Insertion MPD Manipulator send the IF-3 supplied opportunity metadata to the Ad Decision server via IF-4a. The result of the ad decision request will contain advertisements for the entirety of the stream which the Ad Insertion MPD Manipulator transforms into an IF-5 manifest a mixture of content and advertisements.

After a DASH client receives a stream produced from an SSAI architecture, the stream will remain fixed for the duration of the playback session, e.g. the same advertisements will play again should the user choose to rewind the stream.

With *pod level decisioning*, advertisement opportunities are decided just as the DASH client reaches the opportunity within the stream. A SGAI architecture accomplishes by having the Ad Insertion MPD Manipulator use the IF-3 supplied opportunity metadata to generate an IF-5 manifest with a mixture of content and remote entities that represent opportunities. As the client reaches remote entities during playout, the client utilizes IF-7 to return the opportunity metadata to the Ad Insertion MPD Manipulator which then sends the data to the Ad Decision server via IF-4a. The result of the ad decision request will contain advertisements for this single opportunity which the Ad Insertion MPD Manipulator transforms into an IF-7 response for the client to consume.

After a DASH client receives a stream produced from an SGAI architecture, the stream can continue to change for the duration of the playback session, e.g. the advertisements can be re-decisioned should the user choose to rewind the stream.

5.6.3 IF-4b: Content Conditioning request parameters

A *conditioning parameter* is a piece of information about the encoding/packaging of the content stream or a client player capability that is used by the Ad Content Server to ensure an ad creative is compatibly encoded for inclusion in the generated presentation. The Ad Insertion MPD Manipulator collects and sends this information to the Ad Content Server as part of IF-4b.

The transmission of conditioning parameters is highly integration dependent, but examples of commonly used industry parameters are:

- Video / Audio Codecs
- Player Splice Condition Robustness
- Encryption Schemes

NOTE: The information that should be added is the one that was used to initialize the playback for the main content. We review the CMAF/DASH Core Profile. Examples are: CMAF Master Headers, width and height, etc.

It is expected that an Ad Content Server will use a best effort approach to utilize the provided conditioning parameters, but no requirement for real-time transcoding of ad content to fit all parameters is made by this document. Further recommendations on the usage of conditioning parameters by the Ad Content Server are made by interface IF-4c.

The methods for carrying conditioning parameters in the request from the Ad Insertion MPD Manipulator to the Ad Content Server may follow the same guidelines as methods used on requests between the DASH Client and the Ad Insertion MPD Manipulator. Refer to interface IF-7b for further details on these methods.

5.6.4 IF-4e: Ad Selection Result format

5.6.4.1 Overview

The response of the Ad Decision Server identifies the advertisements decisioned by the server and provides information associated with the advertisement such as general metadata, viewability requirements, media files, mezzanines, and tracking events. The actual ad content is provided by the Ad Content Server, preferably following the DASH-IF Ad Content format as defined in clause 5.6.7. Depending on the decisioning mode, the decision response may optionally contain the placement and ordering of advertisements as well.

While the general information carried in the response is described in this interface, the explicit format of this response is workflow dependent. Examples of known industry formats are given in the subsequent sub-clauses of this interface.

For an ad selection result format to be used in the context of this specification, the result format MUST include the following information:

- A reference to one or multiple well-defined ad content, each with a well-defined duration.

The result format may include:

- Tracking beacons
- Identifiers
- Campaign information
- Tracking information

For the purposes of this document we only assume the existence of this described information, but make no requirement or recommendation on the format used to specify this information.

The following sub-clauses provide an informational overview of well-known industry formats for ad decisioning and how each format can be utilized to carry the described information.

5.6.4.2 IAB VAST

An instantiation of IF-4e is standardized by the Interactive Advertising Bureau (IAB) as the Digital Video Ad Serving Template (VAST) [i.6] specification.

The VAST specification provides structure definitions for representing a variety of ad types, including linear, non-linear, and companion. A single VAST response may contain a stand-alone ad slot or a whole pod of ad slots, and each ad structure can provide general metadata, viewability requirements, media files, mezzanines, and tracking events.

In VAST, a reference to DASH-IF ad content (as defined in clause 5.6.7) may be provided for each of the Ad element contained in the response. Specifically, each `Ad` element may contain a **`MediaFile`** element, under the path **`Ad.InLine.Creatives.Linear.MediaFiles`**, where:

- `@type` is compatible to `"application/dash+xml profiles='http://dashif.org/guidelines/dashif-ad-content'"`
- The value of **`MediaFile`** is a URI that resolves to the DASH-IF ad content MPD.

5.6.4.3 SCTE-130

Another instantiation of IF-4e is standardized by the Society of Cable Telecommunications Engineers (SCTE) as the response of the Ad Decision Service (ADS). The ADS is responsible for determining how advertising content is combined with non-advertising content. The exact format and schema of the ADS response is normatively defined within SCTE-130 Part 3 [12], which we defer to for detailed information.

In SCTE-130, a reference to DASH-IF ad content (as defined in clause 5.6.7) may be derived for each **`Placement`** element contained in the **`PlacementResponse`** structure. Specifically, each **`Placement`** may contain a **`core:AssetRef`** element, under the path(s) **`Placement.coreContent`** or **`Placement.CoreRotationList.core:Content`**, where the supplied `@assetID` may be used to retrieve the content from the Ad Content Server specified by `@providerID`.

NOTE: The description above is provided by a basic understanding of SCTE 130-3, should an expert have corrections or additional examples please feel free to provide feedback.

5.6.5 IF-4d: DASH-IF Ad Content Storage format

This interface provides a recommended content format for ad content that is expected to be dynamically inserted into a DASH Live or On-Demand Media Presentation.

Ad content is recommended to follow the DASH-IF Ad content format as defined in the following. This specification does not exclude the use of other content, but the content author should be aware of any differences to the DASH-IF Ad Content format.

DASH-IF Ad follows the restrictions and requirements according of this specification and may be produced independently of the main content for insertion into well-formatted main content by simple MPD manipulation processes.

If content is offered conforming to the DASH-IF Ad content format and follows the following requirements and recommendations, then it may be annotated with a `@profiles` parameter: `"http://dashif.org/guidelines/dashif-ad-content"`.

The following requirements for DASH-IF Ad Content apply:

The content shall be provided as a DASH Media Presentation, i.e. a complete MPD with referenced Segments and shall follow the semantics in

- Table 4.
- The DASH Media Presentation shall conform to the DASH Core Profile for CMAF content as defined in ISO/IEC 23009-1 [1].
- The presentation time shall be anchored to 0. Specifically, this means:
  - The `@presentationTimeOffset` shall be absent, i.e. it is assumed to be zero.
  - The `@eptDelta` shall be absent, i.e. the earliest presentation time of each Segment is 0.

  NOTE: An important assumption for the above profile is the availability of content for CMAF Tracks over the entire Period. Content may be overlapping at the end of the Period.
- The DASH Media Presentation shall contain exactly one Period.
- The **`MPD`**`@type` shall be set to `'static'`.

The following recommendations for DASH Ad Content apply:

- The MPD should contain a profile indicator signalling "`http://dashif.org/guidelines/dashif-ad-content`"
- The content should be offered in segmented format using the Segment timeline.
- The Segment durations within one Adaptation Set should be approximately identical.
- The content may and typically should include multiple variants for the same ad, for example different codecs, formats and resolutions in order for the Dynamic Conditioning, the MPD proxy or a DASH client to adjust the ad to the current playback conditions.
- An **`AssetIdentifier`** descriptor should be present to carry a globally unique content ID for the ad content. Examples for asset identifier schemes are:
  - An Ad-ID identification scheme, defined by SMPTE 2092-1 [13], is signalled by setting:
    - `@schemeIdUri` to "`urn:smpte:ul:060E2B34.01040101.01200900.00000000`"
    - `@value` to the canonical full Ad-ID identifier as defined in SMPTE RP 2092-1 [13].
  - In the absence of other identification schemes, DASH-IF defines scheme may be signalled by setting:
    - `@schemeIdUri` to "`http://dashif.org/guidelines/v5/asset-id`"
    - `@value` to a MovieLabs ContentID URN ([9], 2.2.1)

Table 4 DASH-IF Ad content MPD

Element or Attribute Name	Use	Description
MPD		Provides the requirements for ad insertion content. Any not specified value is identical to what is provided in ISO/IEC 23009-1 ([1], clause 5.3.1).
@profiles	M	should include a profile indicator signalling `http://dashif.org/guidelines/dashif-ad-content` and shall include a profile identifier for the DASH CMAF profile "`urn:mpeg:dash:profile:cmaf:2019`" profile. This also means that the content follows the CMAF Profile.
@type	M	Shall be set to `static`
@mediaPresentationDuration	R	Shall not be present.
@minimumUpdatePeriod	R	Shall not be present, implied by type static.
@minBufferTime	M	Shall be present
@timeShiftBufferDepth	R	Shall not be present
@suggestedPresentationDelay	R	Shall not be present
@maxSegmentDuration	R	Shall not be present
@maxSubsegmentDuration	R	Shall not be present
ProgramInformation	0…N	This should be used to describe information about the ad.
BaseURL	0	Shall not be present. If a Base URL is present, then it is as part of the Period.
Period	1	Exactly one Period shall be present. Provides the requirements for ad insertion content. Any not specified value is identical to what is provided in ISO/IEC 23009-1 ([1], clause 5.3.2).
@xlink:href	R	Shall be absent.
@xlink:actuate	R	Shall be absent.
@start	R	Shall be absent, i.e. assumed to be `0`.
@duration	M	Shall be present and provide the duration of the ad content
BaseURL	1…N	At least one shall be present and refer to the BaseURL of the ad content.
AssetIdentifier	0…1	This should be used to provide an explicit identifier for the ad content. For details see above.
EventStream	0...N	Event Streams are permitted in ad content, for example for beaconing.
AdaptationSet	1...N	At least one Adaptation Set shall be present.
@xlink:href	R	Shall be absent
@xlink:actuate	R	Shall be absent
InbandEventStream	0...N	Inband Event Streams are permitted in ad content, for example for beaconing.
CommonAttributesElements	—	specifies the common attributes and elements (attributes and elements from base type **RepresentationBaseType**). For details, see subclause
SegmentBase@presentationTimeOffset	OD default: 0	shall be absent.
SegmentBase@eptDelta	O	shall be absent.
SegmentBase@pdDelta	O	May be present for non-video tracks, and if present, it shall be non-negative and should be as small as possible.
@contentType	M	Shall be present
SegmentList	0	Shall be absent

Element or Attribute Name	Use	Description
Representation	1 … N	specifies a Representation. At least one Representation element shall be present in each Adaptation Set. Any not specified value is identical to what is provided in ISO/IEC 23009-1 ([1], clause 5.3.3).
EmptyAdaptationSet	0	Shall be absent
UTCTiming	0	Shall not be present
LeapSecondInformation	0	Shall not be present
Key For attributes: M=mandatory, O=optional, R=removed For elements: <minOccurs>…<maxOccurs> (N=unbounded) Elements are **bold**; attributes are non-bold and preceded with an @.		

Figure 6 provides an overview of the DASH-IF Ad content format.

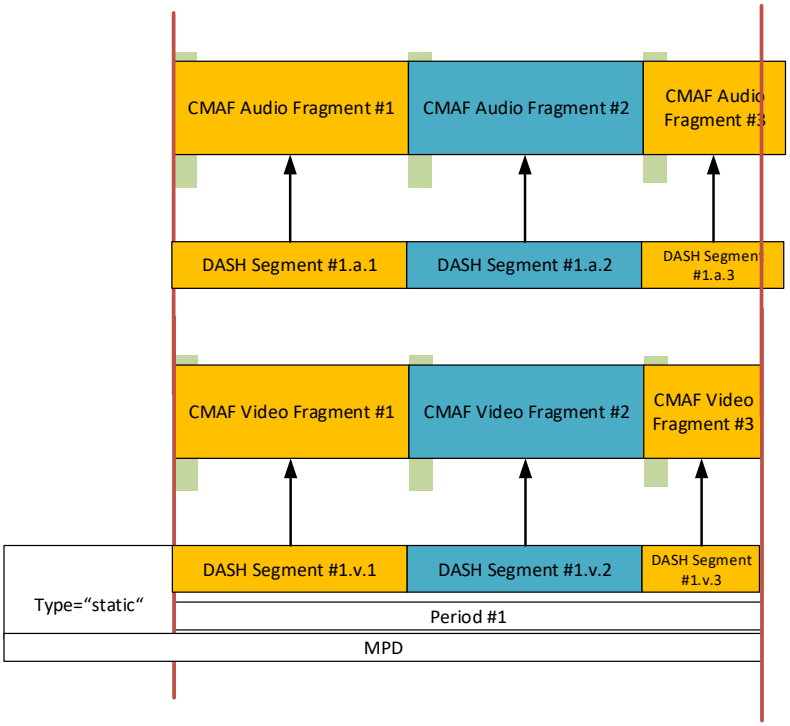

Figure 6 Recommended Ad Content Format

5.6.6 IF-4f: DASH-IF Recommended Slate Content

This interface provides a recommended content format for slate content that is expected to be dynamically inserted into a DASH Live or On-Demand Media Presentation by an Ad Insertion Proxy to fill timeline gaps introduced by ad durations that do not exactly match the placement opportunity duration.

Slate content is recommended to follow the requirements and recommendations of DASH-IF ad content format as defined in this clause 5.6.5. In addition, the following recommendations apply:

- The Segment durations should be short for Live presentations, preferably in the range not exceeding one second. Short segments aid in small acquisition time and cutback accuracy.
- The slate content should be of low complexity, but not represent a still image. Movement is important in order to avoid perception of stalling.
- When no sound is desired or necessary, the slate content should still provide a silent audio track. Providing both video and audio is important to ensure seamless playout by clients.

5.6.7 IF-4c: Recommended Dynamic Ad Content Response format based on conditioning parameters

The response format should follow the DASH-IF Ad Content format as defined in clause 5.6.5.

If ***no*** conditioning parameters are provided, the response should include multiple content variants, e.g. multiple codecs, resolutions, etc..

If codec conditioning parameters are provided, the response should include content options including at least one of the codecs.

If format conditioning parameters are provided, the response should include content options including at least one of the supported formats.

If encryption conditioning parameters are provided, the response should include content options including at least one of the supported encryption modes.

If the content needs to be obfuscated/blocked, then the response should be adjusted to the main content.

If the content needs to be served through xlink with Period, then only the Period of the main content is extracted.

If the content is used on live and especially low-latency live services, then the content in the Period should be adjusted to enable consistent playback including consistent join times.

5.7 IF-5: MPD and Segments with Ad Placements

5.7.1 Overview

The format and requirements of the DASH manifest output by the Ad Insertion MPD Manipulator is described by IF-5 as shown in Figure 7. The Ad Insertion MPD Manipulator operates under the following assumptions:

- It takes the DASH content provided by IF-2 as defined in clause 5.4.
- Opportunity metadata is provided by IF-3 as defined in clause 5.5.
    - The splice point timing is known as $t_{splice\text{-}out}$ when the opportunity starts on media time. $t_{splice\text{-}out}$ matches the **`Period`**`@start` of the main content at the opportunity
    - The end of the ad insertion opportunity, $t_{splice\text{-}in}$, is either
        - known at the time when $t_{splice\text{-}out}$ occurs and the ad decision can be accurately made, or
        - unknown at the time when $t_{splice\text{-}out}$ occurs and the ad decision may either assume a duration or use a signalled planned duration and will adjust the inserted content when $t_{splice\text{-}in}$ actually occurs
- Based on the above, the MPD proxy requests an ad pod for a certain duration and the MPDs of the ad content are returned. The content is properly conditioned ad content and is received via communication on IF-4 as defined in clause 5.6.
- The MPD proxy inserts the ad content into the MPD and makes it part of the media presentation timeline. For details on the operation, refer to clause 0. The **`MPD`**`@minimumUpdatePeriod` is set such that clients return to the proxy at the end of the ad pod at the very latest. If the end point of the opportunity, $t_{splice\text{-}in}$, is unknown, **`MPD`**`@minimumUpdatePeriod` needs to be set such that the client checks back regularly.
- While serving the ad, three options exist
    - The duration of the selected ad content matches the ad opportunity duration in the main content. The return to the main Period is simple.
    - The duration of the select ad content exceeds the ad opportunity duration in the main content. In this case, the ad content needs to be cut short by adding the main Period again. Typically, the ad content overlaps at the start of the main Period, this may cause the removal and/or truncation of multiple ad content Periods.

- The duration of the select ad content is shorter than the ad opportunity duration in the main content. In this case, additional content can be added, either a new ad (if the ad slot is long enough) or a slate, i.e. short piece of content.

- At $t_{splice\text{-}in}$, another splice point exists in the main media to rejoin the main content and $t_{splice\text{-}in}$ matches the **`Period`**`@start` of the main content when transitioning back.

- Each splice point is signalled with a Period boundary (or at least sufficient information is available that the MPD Proxy can add a Period).

Based on this input, the MPD manipulator produces a Media Presentation such that the content on IF-5 can be played back by the DASH client. The transport of additional ad metadata along with linear ad creatives is described as part of IF-6.

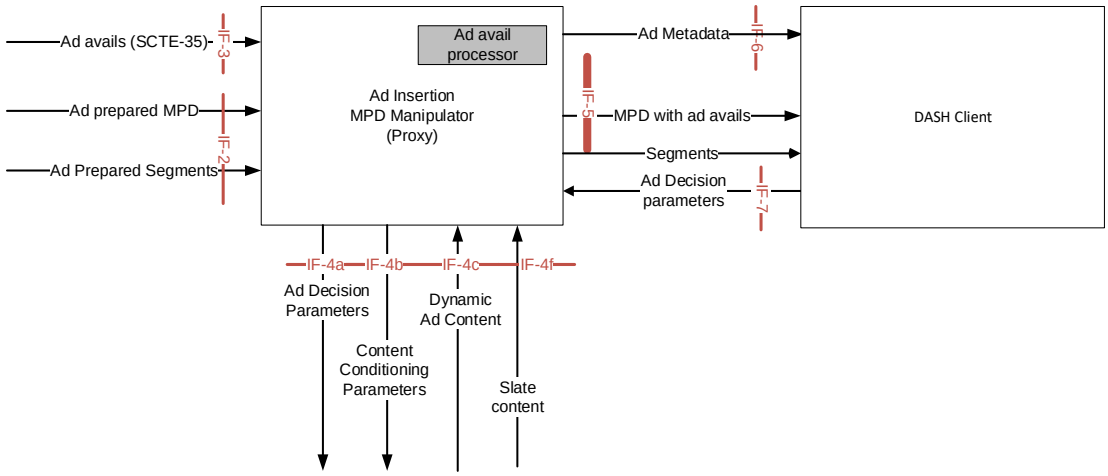

Figure 7 MPD Manipulator Operation: Conforming IF-5 output

In this first version of the document the following requirements are made for a DASH-IF content in the context of ad insertion:

3) The main content on IF-2 follows the definitions in clause 5.4 with the assumption of using option 1 for at least one continuous video Switching set and option 3 for the remaining ones on IF-1. The main content shall follow the DASH Core Profile for CMAF content, i.e. video is aligned with Period boundaries.

4) The ad content on IF-4c should follow the DASH Ad insertion content as defined in clause 5.6.7. In addition, the content may be conditioned, for example such that for any Initialization/Adaptation Set present in the MPD on IF-2, at least one Adaptation Set is provided in the ad content on IF-4b.

5) Ad content is available that is properly formatted

6) Slate content is available that can used in case the ad content does not match the pod.

NOTE: The above assumptions are not a requirement for this specification, but recommended practices. However, MPD proxy implementation/implementer should be aware that if the content is not following the above requirements on what the consequences are for the operation of the service.

5.7.2 Media Presentation Requirements on IF-5

The DASH IOP Guidelines are built on to of the DASH Profiles for CMAF content.

In particular, the content on IF-5 shall conform to the DASH Extended Profile for CMAF content taking into account the multi-period requirements.

If used in the context of Ad Insertion, this document provides further normative requirements for the ad insertion architectures for multi-Period content from inserted ads. However, in general the client should not be able to differentiate between the main content Periods and ad content Periods.

The ad content is spliced into the main content following what is documented in a Period following Table 5.

Table 5 Ad Content spliced into main content

Element or Attribute Name	Use	Description
MPD		Provides the requirements for content that is combined between main content an ad content. Any not specified value is identical to what is provided in ISO/IEC 23009-1 ([1], clause 5.3.1).
ServiceDescription		
Latency@target	0	Target Latency is provided
@profiles	M	Set to "urn:mpeg:dash:profile:cmaf-extended:2019" or "urn:mpeg:dash:profile:cmaf:2019" The proxy can change to "urn:mpeg:dash:profile:cmaf-extended:2019" if it cannot exactly condition the ads to the Period durations.
@minimumUpdatePeriod		Is adjusted accordingly to the operation. For details see clause 0.
InitializationSet	0 … N	The MPD proxy is expected to check the **InitializationSet** parameters if the @inAllPeriods of the Initialization Set is set to true: • If it can provide ad content conditioned to the **InitializationSet** parameters it should leave the Initialization Set included. • If it can not provide ad content conditioned to the **InitializationSet** parameters, it must set the @inAllPeriods of the Initialization Set is set to false or remove the Initialization Set from the MPD.
Period (Main content)		specifies the information of a Period. The information from the main content Period is reused except specified differently
@duration	R	Shall not be present, the duration is determined by the @start of the Ad Period.
EventStream	0...N	re-used from the main content.
AdaptationSet	1...N	re-used from the main content.
AssetIdentifer	0…1	re-used from the main content
Period (Ad Content)	0 … N	specifies the information of a Period. The information from the Ad content Period is reused except specified differently
@id	M	A unique identifier, preferably re-used the one already present in the main content.
@start	M	is set to $t_{splice\text{-}out}$ from the main content
@duration	O (remove)	is typically removed. For detailed operations see clause 0.
BaseURL	1…N	re-used from the remote Ad content Period unless the ad content is moved elsewhere.
@availabilityTimeOffset	M	is set such that the client can download content according to the schedule of the live service. For detailed operations see clause 0.
EventStream	0...N	re-used from the remote slate content Period unless proxy decides to remove based on business rules.

Element or Attribute Name	Use	Description
AdaptationSet	1...N	re-used from the remote slate content Period, but only a subset may be picked based on the main content or information from the client. For more details on how Adaptation Sets are added, please refer to Table 3. In addition, if the Adaptation Set is compatible to an Initialization Set, it should include `@initializationSetRef` attribute with value referencing the compatible Initialization Set.
AssetIdentifier	0…1	re-used from the remote Ad content Period
Period (Slate Content)	0 … N	specifies the information of a Period. The information from the slate content Period is reused except specified differently
`@id`	M	A unique identifier, preferably using a unique slate content identifier.
`@start`	M	is typically set to the sum of $t_{splice\text{-}out}$ and the duration of the previous ad. For detailed operations see clause 0.
`@duration`	O (remove)	is typically removed. For detailed operations see clause 0**Error! Reference source not found.**.
BaseURL	1…N	re-used from the remote slate content Period unless the slate content is moved elsewhere.
`@availabilityTimeOffset`	M	is set such that the client can download content according to the schedule of the live service. For detailed operations see clause 0.
EventStream	0...N	re-used from the remote slate content Period unless proxy decides to remove based on business rules. Slate content is not expected to carry Events.
AdaptationSet	1...N	re-used from the remote slate content Period, but only a subset may be picked based on the main content or information from the client. For more details on how Adaptation Sets are added, please refer to Table 3. In addition, if the Adaptation Set is compatible to an Initialization Set, it should include `@initializationSetRef` attribute with value referencing the compatible Initialization Set.
AssetIdentifier	0…1	re-used from the slate content Period
Period (Main Content)		specifies the information of a Period. The information from the Ad content Period is reused except specified differently
`@start`	M	is set to $t_{splice\text{-}in}$ from the main content
`@duration`	R	shall not be present
EventStream	0...N	Re-used from main content.
AdaptationSet	1...N	re-used from main content.
AssetIdentifier	0…1	re-used from the main content

Key
For attributes: M=mandatory, O=optional
For elements: <minOccurs>...<maxOccurs> (N=unbounded)
The conditions only hold without using `xlink:href`. If linking is used, then all attributes are "optional" and <minOccurs=0>.
Elements are **bold**; attributes are `non-bold` and preceded with an `@`.

5.7.3 MPD Proxy Operation Guidelines

In a SSAI architecture the Ad Insertion MPD Manipulator may utilize IF-4 to request an advertisement decision for part or all of the content stream. For some embodiments, the ad placements provided by an ad decision will be represented as a set of ad pods positioned within the content stream with each pod containing a series of one or more ad slots. In this case, the manipulator will create a new Period for each ad slot with a sequence of Periods representing a complete ad

pod. In other embodiments, the ad placements for a stream may have only one creative for the entire pod, in this case the manipulator will create a single new Period for each pod.

The manipulator inserts the Period element(s) representing the pod(s) at the content stream position specified by the ad decisioning response. If the content MPD provided to the manipulator already has Period boundaries created at the desired ad insertion points, it may directly insert the Period(s) representing the ad placements.

NOTE: The case when the manipulator is provided a content MPD without Period boundaries at the desired insertion points is not covered in the document. In the case, the manipulator would have to split the content Period(s). However, splitting operations may require the manipulator to access the content segments in addition to the content MPD and for this reason it is recommended that the content MPD is provided with Period boundaries already generated. Additional input on this case is welcome.

If the Ad Insertion MPD Manipulator does not perform an insertion at a conditioned Period boundary, and for all Adaptation Sets in the new Period exactly one Adaptation Set is referenced in the Supplemental descriptor for Period continuity, it should remove this Period element and recombine the Periods.

NOTE: This may also be done by the manipulator to provide responses to DASH clients that cannot handle Multi-Period responses.

For scenarios where original in-stream ads are being replaced by the Ad Insertion MPD Manipulator instead of inserted into a clean stream, the manipulator would create the same Period(s) for ad creatives and replace sections of the content stream. If the ads are already delineated in the content MPD then the in-stream Periods are replaced by the generated ones, if the ads are not delineated in the content MPD then the manipulator must perform splitting operations prior to replacement.

A possible detailed operation flow for the proxy is:

1) The MPD proxy passes through the DASH content provided by IF-2 as defined in clause 5.4 for Period that is not an Ad Period. The MPD proxy leaves the value of `@minimumUpdatePeriod` unchanged as it receives information from the main server.

2) When opportunity metadata is provided through IF-3 for a replacement content, the MPD proxy prepares for content replacement.

    a. The splice point timing is known as $t_{splice\text{-}out}$ when the opportunity starts on media time.

    b. $t_{splice\text{-}out}$ matches the **`Period`**`@start` of the main content at the opportunity

    c. the MPD proxy knows or estimates the duration of the opportunity and asks through IF-4a and IF-4b for an appropriate ad content.

3) One or more MPDs for the selected ad content selected are returned via IF-4e. Assuming the ad content follows the Ad content format as defined by IF-4d in clause 5.6.5, the MPD proxy inserts the ad into the MPD and makes it part of the media presentation timeline as follows:

    a. A sequence of insertion Periods is created by collecting all Periods from the selected set of ad content MPDs such that:

        i. The order of the Periods follows the order of the ads declared in the decisioning response.

        ii. The **`Period`**`@start` for each Period is set such that:

            1. The first inserted **`Period`**`@start` is equal to $t_{splice\text{-}out}$.

            2. All subsequent **`Period`**`@start` are equal to the previous **`Period`**`@start` + **`Period`**`@duration`.

        iii. The **`Period`**`@duration` of each period is removed.

        iv. For presentations where early termination of opportunities will not occur, `@availabilityTimeOffset` for the **`BaseURL`** of each insertion Period is set to the duration of the ad opportunity (in order to signal to the client that the ad content is fully available)

v. Where the MPD Proxy has information that the client only needs a subset of Adaptation Sets, all Adaptation Sets unnecessary for client operation are removed from each insertion Period.

vi. For all Adaptation Sets in each insertion Period, for which a compatible CMAF Header is available compared to the main content, period connectivity signalling is added.

b. All client targeted Event Streams that exist during the opportunity time in the main content are copied into the sequence of insertion Periods. Note that this may require splitting the Event Stream across multiple periods with appropriate signalling updates.

c. All Periods within the main content that span the opportunity are removed

d. The sequence of insertion Periods is inserted at $t_{splice\text{-}out}$ in the main content.

e. The **`MPD`**`@minimumUpdatePeriod` is unmodified from the main content

4) The ad is served automatically to the client as the client will start requesting the Segments according to the information and timeline in the MPD from the Ad content server. While serving the ad, the MPD proxy continues to update from the main content server and so do the clients (following the `@minimumUpdatePeriod`). A few aspects may happen:

a. The ad opportunity is terminated through opportunity metadata. This also means that a new Period is received in the main content with **`Period`**`@start` set to $t_{splice\text{-}in}$. If the difference of $t_{splice\text{-}out}$ and $t_{splice\text{-}in}$ is

i. exactly identical to the duration of the inserted ad content. Then the MPD just takes the content directly from the main content. Period connectivity may be signalled if it applies for Adaptation Sets

ii. smaller than the duration of the inserted ad content. In this case the ad needs to be early terminated and the following is done.

1. Each inserted Period with **`Period`**`@start` greater than $t_{splice\text{-}in}$ may be removed.

2. For the inserted Period that contains $t_{splice\text{-}in}$

a. The segment URLs that go beyond the new termination point of the Period may removed

b. For each Adaptation Set, `@pdDelta` is set to indicate the overlap of the last signalled Segment over the Period boundary.

3. Period connectivity may be signalled in the main content if it applies for Adaptation Sets

iii. larger than the duration of the ad content. Then a new decision is taken, see issue 4b.

b. The ad is running out of segments over time and no new splice back opportunity occurs. In this case, a new decision needs to be taken whether to serve a new ad or a slate. There is no difference if one or the other is served. The newly selected content is added following the same process as 3 where the insertion point is the end of the previously inserted content instead of $t_{splice\text{-}out}$.

c. The process may continue with either a or b being doing repeatedly.

In a SGAI architecture the Ad Insertion MPD Manipulator utilizes the opportunity metadata provided by IF-3 to embed signals into the MPD response that allow the DASH client to defer ad opportunity resolution until it is actively needed for the presentation.

NOTE: The DASH-IF Working Group is actively studying the DASH mechanisms appropriate for enabling these signals and will provide further information in future IOP updates.

5.7.4 DASH Client Operation Requirements and Guidelines for Playback on Reference Platform

A DASH client is expected to be able to consume content offered with the DASH Extended Profile for CMAF Content and DASH-IF multi-period ad content. Specifically, this includes:

1) For playback within a single Period the DASH client shall support playback of a single period of content according to DASH profile for CMAF content.

2) If (i) a Representation is selected for playback in the new Period, (ii) this Representation signals Period continuity, and (iii) this Representation is already loaded to the playback buffer, then the DASH client should continue to load the segments into the playback buffer w/o any timeline adjustment or CMAF Header switching.

3) If (i) a Representation is selected for playback in the new Period, (ii) the containing Adaptation Set signals Period continuity to an Adaptation Set in the previous Period, (iii) another Representation other than the one selected is already loaded to the playback buffer, then the DASH client should

    a. first append the Initialization Segment/CMAF Header for the new Representation and

    b. then continue to load the segments of the new Representation into the playback buffer w/o any time adjustment, i.e. only CMAF Header *Switching* is applied.

4) If (i) a Representation is selected for playback in the new Period, (ii) this Representation signals Period connectivity, (iii) this Representation is already loaded to the playback buffer in the previous Period, (iv) the Period start time of the new Period is provided as `PST`, and (v) the presentation time offset normalized by the time scale in the new Representation is `PTO`, then the DASH client should

    a. Load all Segments/CMAF Fragments of the old Period into the playback buffer.

    b. Once completed, *time adjust* the playback by setting the offset to `PST - PTO`.

    c. continue to load the Segments of the new Representation into the playback buffer w/o CMAF Header *switching*.

5) If (i) a Representation is selected for playback in the new Period, (ii) this Representation signals Period connectivity, (iii) another Representation from the same Adaptation Set other than the one already loaded to the playback buffer in the previous Period, (iv) the Period start time of the new Period is provided as `PST`, and (v) the presentation time offset normalized by the time scale in the new Representation is `PTO`, then the DASH client should

    a. Load all Segments/CMAF Fragments of the old Period into the playback buffer.

    b. Then append the CMAF header for the new Representation

    c. Once completed, *time adjust* the playback by setting the offset to `PST - PTO`.

    d. Continue to load the Segments of the new Representation into the playback buffer.

6) If (i) a Representation is selected for playback in the new Period with start time `PST` (ii) no specific signalling is provided on the relation of the current Period and the next one (iii) the presentation duration of the old Representation is provided as `PD`, and (iv) the presentation time offset in the new Representation is set to `PTO` signalled then the DASH client should

    a. either

        i. apply a *change type* call, if this method is supported, or if not

        ii. void the buffer presentation of the old Representation

    b. Then append the CMAF header for the new Representation

    c. *Time adjust* the playback by setting the presentation time at time `PD` to `PST - PTO`.

    d. Continue to load the segments into the playback buffer CMAF Header switching.

7) Playback of overlapping content at Period boundaries can be optimized by using *append window start* and *append window end* signalling in order to exactly define the portion of the segment that needs to be played. Media content outside this range is truncated and not played. More details are for further study.

5.8 IF-6: Ad Metadata Signalling

5.8.1 Introduction

Ad metadata, such as creative descriptions, viewability requirements, and tracking events, is provided to the Ad Insertion MPD Manipulator as part of the IF-4 ad decisioning response. To enable client usage of this metadata, the manipulator may provide the metadata in the presentation via IF-6. Similar to IF-3, the following normative statements on ad metadata carriage are made:

- Ad metadata must be carried through one of the following mechanisms:
  - DASH MPD Events, see ISO/IEC 23009-1 ([1], clause 5.10.2).
  - DASH Inband Event Messages see ISO/IEC 23009-1 ([1], clause 5.10.3).
- DASH MPD Events should be the preferred carriage mechanism

The recommendation of MPD Events over other carriage mechanisms is made such that the Ad Insertion MPD Manipulator can provide ad metadata to the DASH client without modifying the ad segments.

For both mechanisms, the DASH client aligns the surfacing of the event data to the client application with the timed playout of the Period, see DASH-IF Event Processing in ISO/IEC 23009-1 [1], Annex A for further information.

The format and usage of ad metadata is integration specific and is therefore out of the scope of this document. Known metadata event schemes are provided in the subsequent sub-clauses of this interface as informational examples.

5.8.2 DASH Callback Event

MPEG-DASH devices a basic callback event scheme denoted by the scheme id `"urn:mpeg:dash:event:callback:2015"` and `value=1`. When a DASH client encounters this scheme it will treat the message data payload of each event as a URI and perform a GET request ignoring the response. This functionality was designed to directly facilitate the requirements of basic timed tracking events, such as those establish in the IAB VAST specification.

5.9 IF-7: Ad Decisioning Parameters and Remote Resolution

5.9.1 Introduction

In both an SSAI and SGAI architecture, the ad decisioning call of IF-4 requires decisioning parameters about the available opportunity and information about the client which will present the decisioned advertisement. The IF-7 interface can be utilized to directly carry information from clients to the MPD Proxy that is needed for the ad decisioning.

In a SGAI architecture, instead of providing ad placements directly, the MPD provided via IF-5 contains information on how to resolve the ad placements as the DASH client needs them for playout of the presentation. For this architecture, the IF-7 interface can additionally be utilized to perform this late resolution function.

The DASH-IF Working Group is actively studying the DASH mechanisms appropriate for enabling this interface and will provide normative information in a future update of the IOP. The subsequent sub-clauses of this interface detail initial thoughts on this interface and should be considered informational only.

5.9.2 IF-7a: Decisioning Parameters via URL Parameters

Just as in a SSAI architecture, an SGAI architecture must provide the decisioning and conditioning parameters to the Ad Decisioning / Ad Content Server via IF-4, but unlike SSAI, the usage of IF-4 is done as needed instead of pre-emptively. This means that the service entity handling the late resolution of the ad opportunity must be made aware of the parameters that were previously known to the Ad Insertion MPD Manipulator.

MPEG-DASH defines the Flexible Insertion of URL Parameters mechanism to enable the dynamic creation of request parameters by the DASH client, combining information provided in the manifest and information available from other requests. As the Ad Insertion MPD Manipulator is constructing the MPD provided via IF-5, it may utilize the Flexible Insertion mechanism to embed the content decision and conditioning parameters known to it such that they are properly transmitted to the late resolution handler without the need for server-side state.

In addition to the parameters known at original MPD generation time, further investigation is being conducted by the DASH-IF Working Group to determine if player runtime conditions could be additionally included via this mechanism to provide greater detail around active seamless playout requirements.

5.9.3 IF-7b: Content Conditioning via URL Parameters

This aspect is for further study.

5.9.4 IF-7c: Late Binding via Remote Periods

MPEG-DASH defines the XLink mechanism for enabling remote elements within an MPD. The Remote Period variant of remote elements can be used in a SGAI architecture to delay the resolution of ad opportunities. In particular a Remote Period with `@xlink:actuate="onRequest"` can be inserted into an MPD as an ad pod placeholder and the DASH client will perform resolution as the portion of the timeline containing the Remote Period is approached during presentation playout. The response for the resolution can contain one or more Periods to represent one or more ad slots within the ad pod.

Remote elements must be further studied for interoperability guidelines, in particular the usage of remote elements within a dynamic MPD has not been studied sufficiently to understand restrictions and constraints that such an MPD would impose on remote elements.

5.10 IF-8: Ad Tracking and Measurement

5.10.1 Introduction

It is common practice for advertisements to utilize impression tracking from the client to report and measure the number of times an ad is viewed. IF-6 provides the carriage of metadata to enable tracking scenarios that are described by IF-8. Ad tracking and measurement integrations can be very workflow and client platform dependent and are therefore out of the scope of this document. Known tracking mechanisms are provided as informational references.

5.10.2 VAST View Tracking

The IAB VAST specification [i.6] describes a **`Tracking`** element which provides a URI that should be requested when a named event occurs within a creative. A subset of these events describe progression through a linear creative by tracking points of time within the creative, such as start, end, first quartile, second quartile, etc. As these events are directly timed with the playout of the media, the DASH event mechanisms can be used to convey and align these events with the linear creative.

When translating these named events to timed events, the presentation time of the event should correlate to the same logical position as the VAST named event, for instance the start event should be a presentation time of 0 relative to the start of the Period containing the ad creative. If no client application processing is required of the event, the DASH Callback Event scheme may be used to have the DASH client directly perform requests, see clause 5.8.2 for further details. Should a service provider wish to handle the request on their own, to further process the URI or provide it to a third-party library, a custom event scheme may be established by the provider and utilized by the client application. DASH-IF defined a scheme uri for VAST 3.0 https://dashif.org/identifiers/vast30.

NOTE: The https://dashif.org/identifiers/vast30 scheme was initially defined in DASH IOP v4.3 Section 5.3.3.7, describing the carriage of an entire VAST response as part of a single Event element which can then be picked up by a library within the application client that parses out and handles the eventing. As part of this document, we do not carry through the recommendation of full payload insertion, instead describing the breakdown of a VAST document into tracking events as part of IF-8 where we describe how a VAST tracking event can be mapped to timed presentation events.

5.10.3 Open Measurement SDK

The IAB Tech Lab has produced the Open Measurement SDK [i.7] as a way of facilitating third-party viewability and verification measurements without requiring SDKs from individual measurement providers. Service providers integrate the Open Measurement SDK into their client applications and the SDK facilitates the execution of measurement provider defined tracking parameters for each ad creative.

For each ad creative played the SDK must be instantiated with ad metadata describing the measurement providers and their parameters for the creative. This metadata may be carried via IF-6 using a custom event scheme and surfaced to the client application which can then initialize the SDK for the creative.

5.10.4 Alternative Tracking Methods

Other tracking services could be used to track viewer impressions, such as proprietary or open solutions, that enable reliable tracking of viewer impressions.

Annex A (Informative): Change History

Date	Version	Information about changes
2021-11-30	5.0.0	Version published as part 5 v5.0.0