**3GPP TSG SA WG4 Meeting 132S4-250791**

**Fukuoka, JP, 19 - 23 May 2025**

**Source: Qualcomm Incorporated**

**Title: [FS\_MeMe] Manifest-based Rich Media Messages**

**Type: pseudo Change Request**

**Spec: 3GPP TR26.841v1.2.0**

**Agenda item: 8.6**

**Document for: Decision**

**1. Introduction and Discussion**

The key issue on external body content and late binding was not yet sufficiently progressed.

**2. Reason for Change**

This progresses the work.

**3. Conclusions**

Please accept.

**4. Proposal**

It is proposed to agree the following changes to 3GPP TR26.841v1.2.0.

**5. Revision**

This revision takes into account the comments made during the presentation during SA4#132.

|  |  |
| --- | --- |
| TDoc | [S4-250791](https://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_132_Fukuoka/Docs/S4-250791.zip) |
| Title | [FS\_MeMe] Manifest-based Rich Media Messages |
| Source | Qualcomm Sweden |
| Contact | Thomas Stockhammer |
| Agenda Item | 8.6 |
| E-mail Discussion | No e-mail discussion. |
| Revisions | No revisions available. |
| Minutes | 21/05/25  Presented by Thomas.   * Iraj: Is it possible to do late binding?   + Thomas: That is not explicitly said. But if you have packaged content, it is aligned.   + Iraj: I can suggest a text. * Rufael: Is it only for static MPD or also dynamic?   + Thomas: Only to static, single period… very simple. * Thomas: We need to look at Rufael’s contribution in 828. |
| Disposition |  |
| Status | parked |

|  |  |
| --- | --- |
| TDoc | [S4-250828](https://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_132_Fukuoka/Docs/S4-250828.zip) |
| Title | [FS\_MEME] Thoughts and Issues with using DASH for messaging service |
| Source | HUAWEI TECH. GmbH |
| Contact | Rufail Mekuria |
| Agenda Item | 8.6 |
| E-mail Discussion | No e-mail discussion. |
| Revisions | No revisions available. |
| Minutes | 21/05/25  Presented by Rufael.   * Thomas: Interoperability is already solved with format profiles in 143. * Thomas: About b), I agree this has not to be packaged on the device. * Thomas: About d), I don’t want to push for DRM. Encryption like clear key could also be used.   + Rufael: Maybe we need more studies to do that.   + Thomas: Yes. I just wanted to say there is an opportunity there. * Waqar: There are already specifications for online content with reliable MPD or M3U8. Maybe some security is needed. * Iraj: Is video and audio are also provided with the manifest? We need to check if local content (with a local HTTP server) is OK.   + Thomas: We can add this. * Thomas: I can take these comments into account in my other contributions. |
| Disposition | Agree to take this into account. |
| Status | agreed |

\* \* \* First Change \* \* \* \*

## 5.3 Key Topic #3: Support of external body content and late binding

### 5.3.1 Description

MIMI as an example requires clients to support the external body content that can be accessed with http and https URLs. The functionality is clearly of relevance. However, in context of 3GPP and MPEG, additional options exist, for example using media/external media type, using external data references in file formats, usage of HTML-5 presentations, glTF scenes or streaming manifests with inline references. A study of suitable support of different options is relevant.

In 3GPP TS 26.143 [26143], clause 4.1, it is started that container formats may support different functionalities, among others it may have body parts that reference external content via URI that will be processed automatically. In this case, it includes a media type and may optionally include the size of the data, an expiration timestamp and other parameters. The content may be rendered with the other parts of the message, or a be downloaded or rendered separately. However, in 3GPP TS 26.143 [26143], clause 5.2.1, it is explicitly noted that specification does not define mechanisms for referencing external content. This aspect is for further study.

As for example stated in clause 4.2.3, MIMI supports ExternalPart as a convenient way to present both "attachments" and (possibly inline rendered) content which is too large to be included in a Messaging Layer Security (MLS) application message. One may differentiate whether the content is rendered inline, or as attachment if the sender intends the content to be downloaded or rendered separately. Examples of use cases include to store bulky content (ex: videos, images, recorded sounds) outside the messaging infrastructure, or to access a specific service URI, for example, a media forwarding service for conferencing.

In another use case, content may be available in different content versions, and content versions may differ for each media type (examples to follow). Clients then select a version that is most suitable. The selection is based on client capabilities, client preferences, user preferences and possibly also interactive signalling with the user.

Multiple content versions may be offered to provide the same content in different encodings, for example different codecs; or different source formats, for example one content version encoded from a standard dynamic range master and another encoded from a high dynamic range video master. Alternatively, the content may differ in languages (applies for audio and subtitles), or different camera views of the same scene, etc.. This aspect gets even more prominent with the advance of advanced transcoding, possibly including AI/ML-based functions: Content generated on a device may be provided as original content, but may also be improved, for example into superresolution, generation of stereo content, relighting and HDR content, translated into different languages, etc.

A content model, similar as is presented for DASH-IF IOP Guidelines [IOP4.3], clause 3.9, may be considered to express different content options that are provided to an MMBP Player, documented in Figure 5.3.1-1.



Figure 5.3.1-1 Content Model of Media Content and MMBP Selection

Proper labelling of content allows consistent selection.

In order to support the content author in providing content in a consistent manner, Figure 2.3-1 provides a conceptual content model. The content may be described as a whole and may contain different media types, video, audio, subtitle and application types.

Within each media type, the content author may want to offer different alternative content that are time-aligned, but each alternative represents different content. The selection is expected to be done by communication with an application or the user, typically using a user interface appropriate for selection. In the absence of this external communication, or at startup or if inline rendering is identified, the MMBP still needs to playback content and therefore benefits from information of what is the default content. Such default content is referred to as main content, whereas any content that is not main is referred to as alternative. There may be multiple alternatives which may need to be distinguished. Examples for such are synchronized camera views of one master content, different perspectives on the content, etc.

Furthermore, it may be that content of different media type is linked by the content author, to express that two content of different media type are preferably played together. As an example, there may be directional audio coming from one view, but for a different camera view, a different audio sound is recorded. In addition to semantical content level differentiation, each alternative content may be prepared with different target versions, based on content preparation properties (downmix, subsampling, translation, etc.), client preferences (decoding or rendering preferences, e.g. codec), client capabilities (decoding capabilities, rendering capabilities) or user preferences (accessibility, language, etc.).

In addition, the content author may also provide priorities for target versions. Typical examples are that the content is prepared for H.264/AVC and H.265/HEVC capable receivers, and the content author prefers the selection of the H.265/HEVC version as its distribution is more efficient. A device supporting both decoders may then choose the one with higher priority signalled by the content author. In a similar version, the same content may be provided in different languages. In this case, it can still be expected that the language can be automatically selected by the client, so it is assigned to a target version. Again, a content author may express priorities on languages, for example preferring the native language over a dubbed one. Languages may be considered as alternative content as well, but as long as automatic selection can be provided, it may be considered as different target versions. Hence for each content of one media type, different target versions may exist and the annotation of the content expressed that it is expected that automated selection can be done. Finally, in the content model, each of the target version typically has multiple versions may be prepared to enable quality and rate-based selections, primarily using the bandwidth and possibly abstract quality information. In another variant, switching across different versions may even be considered.

In summary, the discussion in this clause addresses the following issues:

- Ability to pre-download information about available content that is only referenced, but not included. Download happens only once the download is triggered by the user or the application.

- Rich content offering (for example by transcoding or AI/ML-based processing) with different options to be selected by application, user, or system

- providing content in different bitrates and quality to address network bandwidth availability.

- generic ability to refer to a content or service externally.

### 5.3.2 Gap Analysis and Requirements

Based on the basic call flow in clause 4.1 and in Figure 4.1-3, the description in 5.3.2-1 is extended in the call flow to address the above functionality.

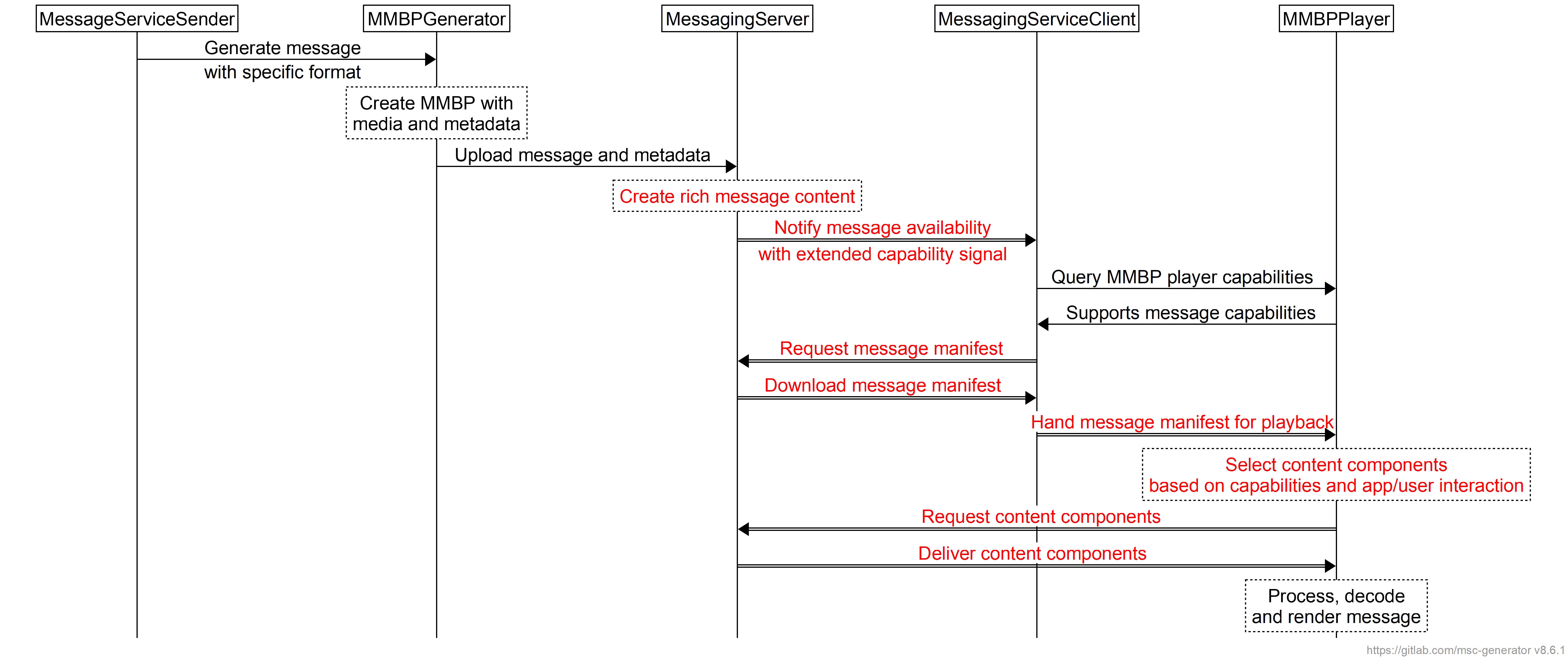


Figure 5.3.2-1 High-Level Call flow for messaging with external referencing and late binding

The following high-level call flow is executed to address messaging:

1) A Message Service Sender instructs and MMBP generator to generate a message with the requested formats where the format is defined through a set of parameters handed over on an MMBP-GEN-API.

2) The MMBP generator creates an MMBP with relevant media data, and adds relevant static and dynamic metadata.

3) The generated file with associated relevant static and dynamic metadata is provided in a container message that is uploaded to a messaging server, together with a **signalling of the capability of the message**.

**3a) The messaging server creates a rich message, possibly including different versions, languages, codecs. Whether the messaging server creates this a priori, or just creates a manifest, is implementation specific.**

4) A messaging service client is informed about the availability of a message in this format using the media type signalling together with appropriate sub-parameters. **The signalling may include different capability profiles.**

5) A messaging service client uses the MMBP-PLAY-API to query the MMBP player with it can process, decode and render the message.

6) If the MMBP player indicates that it supports the capability, the messaging service clients selects and downloads the **message manifest** and hands it to the MMBP player for processing playback**, including downloads of relevant components**.

**6a) The MMBP player selects the subset of the content components, request these from the Messaging Server, decodes and renders the message based on its decoding and rendering capabilities.**

Based on the above analysis, the following gaps are identified:

1) Support for a richer message format that can include different versions of the content, targeting different users, device capabilities, etc.

2) The ability to have a two-stage process, where external references are included in the content (for example in a manifest) and only in the second stage, the selected content components are downloaded.

3) the ability to either do binding of the content components, either early binding (combine the content on the server in a single asset) or permit late-binding (download of individual components and synchronized playback on the server).

4) A client implementation to consistently select content based on consistent annotation in a manifest.

### 5.3.3 Potential Solutions

#### 5.3.3.1 Solution 1: External Referencing in Multipart MMBP

##### 5.3.3.1.1 General

Container types for messages permit the ability to reference external content. Typically, this content is accompanied with additional metadata, such as the type of the content, the size of the content, some expiration time stamps, as well as other descriptive and security related parameters. Reference may be to a single media file, or to service. A restricted set of functionalities may be supported, namely the number 2) in clause 5.3.2, the ability to have a two-stage process. Late binding and rich content items are not supported.

##### 5.3.3.1.2 RFC 2046

RFC 2046 [RFC2046], clause 5.2.3, defines the external-body subtype that indicates that the actual body data are not included, but merely referenced. In this case, the parameters describe a mechanism for accessing the external data. When a MIME entity is of type "message/external-body". The parameters include access-type, expiration timestamps, size of the external body and permissions.

##### 5.3.3.1.3 IETF MIMI

IETF MIMI Message Content [IETF-MIMI], clause 4.5 defines an ExternalPart that provides a convenient way to reference external content and provides a similar function to the message/external-body media type as defined in RFC 2046. Information may include the size of the data in octets, an optional timestamp after which the external content is invalid and information about the content type. Encryption and descriptive information may also be provided. The external URL may be a service.

#### 5.3.3.2 Solution 2: Streaming Manifest-based solution

##### 5.3.3.2.1 General

In another option, streaming manifests provide reference to content components for streaming/download. Streaming manifests also describe in details content properties such as languages, media types, allow to offer different codecs and many different content description formats. They also include all means to support URLs for referencing and many other options that are relevant for messaging services. Also, streaming manifests allow late-binding. Hence, they may serve as well supported format to annotate and download of different components. At the end, the MMBP player may just be a streaming client that communicates with the Messaging Service Client using the APIs of the player.

##### 5.3.3.2.2 DASH Media Presentation

A prominent and well-studies approach is a DASH Media Presentation. Different indications to use DASH as a download and storage format have been discussed and progressed:

- In TR 26.938 [26938], DASH as a download format was introduced. An example was provided using the basic-on-demand profile. In this case, the DASH Media Presentation perfectly describes a format the can also be used for download services. The MPD permits to offer DVD-like content as download content.

- In DASH-IF IOPv5 part 5 [IOP5-5], Ad Insertion, a content storage format for ads is defined in clause 5.6.5 that can be used as inserted and downloaded ads.

- In the sixth edition of ISO/IEC 23009-1, a new profile for storing simple on-demand assets is defined.

- In ISO/IEC 23009-9, an S-MPD is defined that allows to store.

In addition, MPEG-DASH defines a full content annotation scheme, and a mapping of the media annotation to the MPD is defined in ISO/IEC 23009-1, as well as DASH-IF IOP Guidelines [IOP4.3], clause 3.9. The content model shown in Figure 5.3.1-1 can be mapped to a DASH MPD.

Based on the above analysis, the identified gaps can be addressed by a proper MPEG-DASH profile:

1) Support for a richer message format that can include different versions of the content, targeting different users, device capabilities, etc.

- A DASH MPD can include many different content versions, a mapping of the media annotation to the MPD is defined in ISO/IEC 23009-1, as well as DASH-IF IOP Guidelines [IOP4.3], clause 3.9. This also includes

2) The ability to have a two-stage process, where external references are included in the content (for example in a manifest) and only in the second stage, the selected content components are downloaded.

- By adding the MPD in the MMBP message, the content components can be externally referenced. The MPD permits signaling of profiles and capabilities, and one or multiple profiles can be created to signal MMBP content capabilities in DASH MPD.

- It may be even beneficial to signal the URL to the MPD in an external content reference aligned with the solution in clause 5.3.3.1, in order to permit a 3-stage download.

3) the ability to either do binding of the content components, either early binding (combine the content on the server in a single asset) or permit late-binding (download of individual components and synchronized playback on the server).

- MPEG-DASH in its deployed format uses late binding, i.e. each media component is stored separately and the client can combine the downloaded Representations.

4) A client implementation to consistently select content based on consistent annotation in a manifest.

- DASH-IF IOP Guidelines [IOP4.3], clause 3.9 describe a detailed client model on how to select the content.

- The client selection process is implemented in DASH reference client dash.js

- dash.js or a general DASH player may be viewed as an MMBP player with all functionalities including appropriate APIs.

In summary, a DASH-based Media Presentation and DASH player fulfil all requirements.

##### 5.3.3.2.3 DASH/HLS Combined Presentation

DASH/HLS interoperability is a key issue in supporting highly scalable distribution systems for CDN-based distribution as well as for MBS/MBMS distribution. Offering common CMAF segments that can be consumed by both DASH and HLS media players promises to address these issues.

The CTA WAVE DASH-HLS Interoperability Specification [CTA-5005-A] specifies how to generate CMAF content that can be delivered using both a DASH MPD and an HLS m3u8 manifest. These guidelines cover the following use cases:

1. *Basic on-demand and live streaming:* The CMAF content is provided without encryption for on-demand or live consumption.

2. *Low-latency live streaming:* The CMAF content is provided to be consumed in a low-latency fashion with an end-to-end delay less than 3 times that of the CMAF segment duration.

3. *Encrypted media content:* The content of case 1, but MPEG Common Encryption is applied.

4. *Presentation splicing:* The content is similar to case 1, but consists of multiple CMAF presentations, either concatenated or spliced in the middle from one to another.

5. *Carriage of timed event data:* Timed metadata is delivered with the CMAF content, either as part of the media segments, or as part of the presentation manifest, and is expected to be delivered and processed along the media timeline.

6. *Carriage of track roles:* Content is annotated with the role of each track/switching set and these annotations need to be delivered to the client to be used for the selection process.

In particular use case 1 and 6 address the needs to support a common content with two manifests.

With the nesting of alternate content and providing an MPD and HLS manifest URL in the message, a fully compliant messaging system for DASH and HLS players can be supported.

##### 5.3.3.2.4 Example

Listing 5.3.3.2.4-1 implements an MMBP container that includes a link to a 3GP file, an MPD as well as link to an HLS M3U8 and the client can choose which option. Within the manifests, the content options can be selected. In addition, the profile included in the MPD can be signalled on content level.

Listing 5.3.3.2.4-1 Example MMBP container to include two options for rich content

|  |
| --- |
| MIME-Version: 1.0  Content-Type: multipart/alternative; boundary="boundary42"  --boundary42  Content-Type: message/external-body; access-type=URL;  URL="https://www.example.com/video4messaging.3gp"  Content-Type: video/3gpp  --boundary42  Content-Type: message/external-body; access-type=URL;  URL="https://www.example.com/dash4messaging.mpd"  Content-Type: application/dash+xml profile="urn:3GPP:org:26143:baseline"; charset="UTF-8"  --boundary42  Content-Type: message/external-body; access-type=URL;  URL="https://www.example.com/hls4messaging.m3u8"  Content-Type: application/vnd.apple.mpegurl; charset="UTF-8"  --boundary42-- |

An example MPD on TR 26.938 is provided in Listing 5.3.3.1.4-2. The content includes two languages, and two video Adaptation Sets and one for time text. Preferably, each of the content conforms to the requirements in TS 26.143.

Listing 5.3.3.2.4-2 Example MPD container to include richer content offering

|  |
| --- |
| <?xml version="1.0" encoding="UTF-8"?> <MPD   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  xmlns="urn:mpeg:DASH:schema:MPD:2011"  xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011"  type="static"  mediaPresentationDuration="PT3256S"  minBufferTime="PT1.2S"  profiles="urn:mpeg:dash:profile:isoff-on-demand:2011, urn:3GPP:org:26143:baseline">   <BaseURL>http://cdn1.example.com/</BaseURL>  <BaseURL>http://cdn2.example.com/</BaseURL>   <Period>  <!-- English Audio -->  <AdaptationSet mimeType="audio/mp4" codecs="mp4a.0x40" lang="en" subsegmentAlignment="true">  <Representation id="1" bandwidth="64000">  <BaseURL>7657412348.mp4</BaseURL>  </Representation>  <Representation id="2" bandwidth="32000">  <BaseURL>3463646346.mp4</BaseURL>  </Representation>  </AdaptationSet>  <!-- French Audio -->  <AdaptationSet mimeType="audio/mp4" codecs="mp4a.40.2" lang="fr" subsegmentAlignment="true">  <Role schemeIdUri="urn:mpeg:dash:role" value="dub"/>  <Representation id="3" bandwidth="64000">  <BaseURL>3463275477.mp4</BaseURL>  </Representation>  <Representation id="4" bandwidth="32000">  <BaseURL>5685763463.mp4</BaseURL>  </Representation>  </AdaptationSet>  <!-- Timed text -->  <AdaptationSet mimeType="text/mp4" codecs="3gp.text" lang="fr" lang="de">  <Role schemeIdUri="urn:mpeg:dash:role" value="subtitle"/>  <Representation id="5" bandwidth="256">  <BaseURL>796735657.mp4</BaseURL>  </Representation>  </AdaptationSet>  <!—Video AVC -->  <AdaptationSet mimeType="video/mp4" codecs="avc1.4d0228" subsegmentAlignment="true" selectionPriority="2">  <Representation id="6" bandwidth="256000" width="320" height="240">  <BaseURL>8563456473.mp4</BaseURL>  </Representation>  <Representation id="7" bandwidth="512000" width="320" height="240">  <BaseURL>56363634.mp4</BaseURL>  </Representation>  <Representation id="8" bandwidth="1024000" width="640" height="480">  <BaseURL>562465736.mp4</BaseURL>  </Representation>  <Representation id="9" bandwidth="1384000" width="640" height="480">  <BaseURL>41325645.mp4</BaseURL>  </Representation>  <Representation id="A" bandwidth="1536000" width="1280" height="720">  <BaseURL>89045625.mp4</BaseURL>  </Representation>  <Representation id="B" bandwidth="2048000" width="1280" height="720">  <BaseURL>23536745734.mp4</BaseURL>  </Representation>  </AdaptationSet>  <!—Video HEVC -->  <AdaptationSet mimeType="video/mp4" codecs="hev1.1.6.L150.90" subsegmentAlignment="true" selectionPriority="1">  <Representation id="C" bandwidth="128000" width="320" height="240">  <BaseURL>8563456473-h.mp4</BaseURL>  </Representation>  <Representation id="D" bandwidth="256000" width="320" height="240">  <BaseURL>56363634-h.mp4</BaseURL>  </Representation>  <Representation id="E" bandwidth="512000" width="640" height="480">  <BaseURL>562465736-h.mp4</BaseURL>  </Representation>  <Representation id="F" bandwidth="695000" width="640" height="480">  <BaseURL>41325645-h.mp4</BaseURL>  </Representation>  <Representation id="G" bandwidth="760000" width="1280" height="720">  <BaseURL>89045625-h.mp4</BaseURL>  </Representation>  <Representation id="H" bandwidth="1024000" width="1280" height="720">  <BaseURL>23536745734-h.mp4</BaseURL>  </Representation>  </AdaptationSet>  </Period> </MPD> |

##### 5.3.3.2.5 Potential Open Issues

The following aspects are identified as potential open issues:

- Is there an ability to package DASH content including segments in a multi-part MIME container to not download the information, but provide all information to the client within the message?

- For full interoperability, a very restricted subset of DASH/CMAF needs to be defined, for example using a single Period, specific codecs and so on.

- It needs to be clarified if the content following the manifest-based format can be generated on devices, or only on messaging servers. If the earlier, what would be the requirements for producing richer content. If done on the server, an end-to-end encryption may prevent of transcoding the information.

- Messages may be available only for a certain period of time on the messaging server. Some consistency between message availability and the signalling in the manifest is needed.

- It should also be clarified if native support of a player is needed or possibly the playback can be supported by downloading a plugin, or JavaScript code as for example dash.js

- Another open issue results from URLs in messages that may be prone to security and/or privacy attacks.

### 5.3.4 Summary and Conclusions

Generally, richer content is expected to be distributed in messaging services. Also with the availability of AI-based processing, content may be transformed to provide richer experiences for different users, device types and situations. Providing, annotating and accessing such rich content is important and concepts such as external referencing, late binding, streaming access, and other aspects are of relevance.

Based on the discussion in this clause, it is recommended to address richer content formats in media messaging and permit external referencing. The following additions to TS 26.143 are recommended:

1. Add in clause 5.2.1 in the Player and Decoding capabilities the functionality to support message/external-body as defined in RFC 2046.

2. Add a new Manifest-based container format in a generic manner in a new subclause of clause 5 in TS 26.143 and reference the clause from clause 5.2.

3. Provide an instantiation for a DASH-based MPD based on the existing DASH-based storage formats identified in clause 5.3.3.2.2.

4. Define a reference client implementation based on DASH-IF IOP v4.3 and dash.js for content selection in an Annex of TS 26.143.

5. Create a new Manifest-based MMBP Player Profile that permits the following options

- external referencing

- referencing of DASH MPDs

- referencing of HLS M3U8

- the included content in the presentations conforms to the content defined in TS 26.143

- a DASH profile identifier is provided

- add the call flow in 5.3.2 to the profile

6. Address the open issues summarized in clause 5.3.3.2.5

7. Add an example to the Annex aligned with the one in clause 5.3.3.2.4.

In addition to the above benefits, the usage of a streaming client in such messaging applications as a player would permit to also inherit additional functionalities, such as DRM (see key topic #4 in clause 5.4.), metrics, and other reporting functionalities.

\* \* \* End of Changes \* \* \* \*