## **3GPP TSG-SA WG4 Meeting #129-e *S4-241534***

**Online, 19-23 August 2024 (revision of S4al240085)**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.3* | | | | | | | | |
| **CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.804** | **CR** | **0018** | **rev** | **-** | **Current version:** | **18.1.0** |  |
|  | | | | | | | | |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: comprehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* | | | | | | | | |
|  | | | | | | | | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME | **X** | Radio Access Network |  | Core Network | **X** |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | and a sample solution | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Huawei Hisilicon | | | | | | | | | |
| ***Source to TSG:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | FS\_AMD | | | | |  | ***Date:*** | | | 2024-05-23 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | **B** |  | | | | | ***Release:*** | | | Rel-19 |
|  | *Use one of the following categories:* ***F*** *(correction)* ***A*** *(mirror corresponding to a change in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)*  Detailed explanations of the above categories can be found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | | | | | | | | *Use one of the following releases: Rel-8 (Release 8) Rel-9 (Release 9) Rel-10 (Release 10) Rel-11 (Release 11) … Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)  Rel-20 (Release 20)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | Introduce the key issue of dynamic content generation in redundant workflows that is common in the media industry. There are some benefits in deploying this closer to the user in the mobile network. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Dynamic content generation from distributed sources is common in the media industry, by adopting dynamic content generation in the network more targeted and personalized experiences can be achieved. Examples include forensic watermarking, advertisement insertion, additional codec or DRM support and many other use cases could be enabled. The baseline case of resilience and redundancy is also significantly beneficial.  As of today, 5G media streaming is not (widely) adopted in the industry.  This additional feature may make it more attractive to enable media streaming related functionality in the network using 5GMS. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | FS\_AMD will not enable new and improved user experiences, potential of 5G media streaming is not utilized. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | 2, 5.x (NEW) | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **Y** | **N** |  | | | |  | | |
| ***Other specs*** | |  | **X** | Other core specifications | | | | TS/TR ... CR ... | | |
| ***affected:*** | |  | **X** | Test specifications | | | | TS/TR ... CR ... | | |
| ***(show related CRs)*** | |  | **X** | O&M Specifications | | | | TS/TR ... CR ... | | |
|  | |  | | | | | | | | |
| ***Other comments:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

## FIRST CHANGE

## 2 References

[DASH9] Draft Text of ISO/IEC FDIS 23009-9 Information technology - Dynamic adaptive streaming over HTTP (DASH) – Part 9: Redundant Encoding and Packaging for segmented live media (REaP), ISO/IEC JTC 1/SC 29/WG 3 NO 1165, Jan. 26, 2024. [Online]. Available: <https://www.mpeg.org/standards/MPEG-DASH/9/>

[UNPKG24] Emma Roth, "A popular open-source content delivery network went down for hours", The Verge, Apr. 12, 2024. [Online]. Available: <https://www.theverge.com/2024/4/12/24128276/open-source-unpkg-cdn-down> (accessed May 9, 2024).

[NET23] Sebastian Moss, "Cloudflare recovers from service outage after power failure at core North American data center", Data Center Dynamics, Nov. 3, 2023. [Online]. Available: <https://www.datacenterdynamics.com/en/news/cloudflare-recovers-from-service-outage-after-power-failure-at-core-north-american-data-center/> (accessed May 9, 2024).

[FSLY21] Brian Barrett, "How an Obscure Company Took Down Big Chunks of the Internet", Wired, Jun. 8, 2021. [Online]. Available: <https://www.wired.com/story/fastly-cdn-internet-outages-2021/> (accessed May 9, 2024).

[AKAM21] Josh Fomon, "CDN Provider Akamai Takes Down Popular Internet Services During Outage", Ookla, Jul. 22, 2021. [Online]. Available: <https://www.ookla.com/articles/akamai-outage-july-22-2021> (accessed May 9, 2024).

[NET22] Charlotte Trueman, "Cloudflare outage brings hundreds of sites, services temporarily offline", Computer World, Jun. 21, 2022. [Online]. Available: <https://www.computerworld.com/article/1627967/cloudflare-outage-brings-hundreds-of-sites-services-temporarily-offline.html> (accessed May 9, 2024).

[VZ19] Jim Salter, "The Internet broke today: Facebook, Verizon, and more see major outages", Ars Technica, Jul. 3, 2019. [Online]. Available: <https://arstechnica.com/information-technology/2019/07/facebook-cloudflare-microsoft-and-twitter-suffer-outages/> (accessed May 9, 2024).

[DEMX01] Marc Hoppner, "A content owner, a CDN and a player walk into a bar", (Jan. 6, 2023). Accessed: May 9, 2024. [Online Video]. Available: <https://www.youtube.com/watch?v=S9EdoQFOQ9I&list>  
=PLkyaYNWEKcOf98lZxnCcL6y7ZIVU3oSYO&index=12

[DEMX02] Guillaume du Pantavice, "Improving streaming experience with Bayesian optimization, from AB to AZ test", (Dec. 25, 2021). Accessed: May 9, 2024. [Online Video]. Available: https://www.youtube.com/  
watch?v=t4nRrLygVwo&list=PLkyaYNWEKcOfD1GYFxFbZXDP03XM-cZPg&index=19

[IEEE01] E. Ghabashneh and S. Rao, "Exploring the interplay between CDN caching and video streaming performance", IEEE INFOCOM 2020 – IEEE Conference on Computer Communications, Toronto, ON, Canada, 2020, pp. 516-525.

[ACM01] K. Vermeulen, L. Salamatian, S. H. Kim, M. Calder, and E. Katz-Bassett, "The central problem with distributed content: common CDN deployments centralize traffic in a risky way", In Proceedings of the 22nd ACM Workshop on Hot Topics in Networks (HotNets ’23). Association for Computing Machinery, New York, NY, USA, 70-78.

[MHV01] A. Bentaleb, R. Farahani, F. Tashtarian, C. Timmerer, H. Hellwagner, and R. Zimmermann, "Which CDN to Download From? A Client and Server Strategies", (Jan. 6, 2024). Accessed: May 9, 2024. [Online Video]. Available: <https://www.youtube.com/watch?v=xCZmCnWgQRE>

[VAS01] Will Law, "Content steering with MPEG DASH", (May 4, 2023). Accessed: May 9, 2024. [Online Video]. Available: <https://www.youtube.com/watch?v=o9Pa5y-Usxw>

[MWS23] W. Law and Y. Reznik, "DASH content steering at scale", Media Web Symposium (MWS’23), June 2023.

[DIFCS] ETSI TS 103 998: "DASH-IF: Content steering for DASH".

[CMMF] ETSI TS 103 973: "Coded multisource media format (CMMF) for content distribution and delivery".

[TS26510] 3GPP TS 26.510: "Interactions and APIs for provisioning and media session handling (Release 18)".

[TS26512] 3GPP TS 26.512: "5G Media streaming (5GMS) protocols (Release 18)".

[MHV02] Roberto Ramos-Chavez, Espen Braastad, Jamie Fletcher, and Arjen Wagenaar. 2024. Standards based OTT A/B Watermarking at Scale. In Proceedings of the 3rd Mile-High Video Conference (MHV '24). Association for Computing Machinery, New York, NY, USA, 94. <https://doi.org/10.1145/3638036.3640251>

## SECOND CHANGE (ALL NEW TEXT)

## 5.x KI#X Multi-AS Dynamic content generation

### 5.x.1 Description

#### 5.x.1.1 Introduction

Media streaming applications traditionally obtain content from a multiple sources over different paths in a network. This imposes several limitations:

1. content from different sources needs to be interchangeable (not necessarily identical), this is not a problem for static content, but for dynamically generated content such as live media this can be problematic as differences may occur during various system configuration such as clock synchronization issues, delay in different parts of the workflow, but also general configuration settings related to the media processing to dynamically generate the presentation.

2 When content is generated closer to the user, in the edge of the network more dynamic and interesting media presentations can be realized as more localized version of the media streaming generation can be generated, i.e. dynamic content generation in the mobile network in the AS can be attractive for operators.

This study considers integration of different technologies into the 5G Media Streaming System that addresses the case of dynamic content generation at different AS and sources, and similar, issues by allowing media streaming applications to efficiently generate and access content that may be changing dynamically at the edge of the network from different application servers. Different client implementations may then beneficially use the content from these multiple sources or networks concurrently, potentially guided by service or network provider.

This both improves redundancy (resilience to failures) and flexibility (extending media services).

In addition to be being able to dynamically generate content from multiple AS for redundancy/resilience, it can also enable use cases such as content replacement, introducing new codecs and/or DRM’s (as separate AS generates representations of a certain codec or DRM). Another application is watermarking where multiple sources generate identical content, but the embedded watermark is different, and both segments and playlists need to be interchangeable.

The key aspect addressed are redundant workflows with multiple packager, encoder steps, in this case a single contribution sources passes through multiple distribution encoders and packagers that may be distributed throughout the network.

#### 5.x.1.2 Challenges Multi-AS dynamic content generation

Dynamic content generation usually includes different components such as contribution encoders, distribution encoders, DRM etc, this is to be able to create an optimized and targeted experience to the end user. By generating content closer to the end user in the mobile network, more targeted and personalized experience can be achieved.

Challenges in deployments with multi-AS deployments and architectures with dynamic content generation may include:

1. *Sustained CDN-/network-wide service disruptions* where network access, connectivity or QoS is severely degraded. Examples may include cases where an entire CDN’s network is degraded because of a network-wide misconfiguration or power failure. In this case a client should switch to another CDN for retrieving the content

2. *Intermittent or short-term disruptions or delay between upstream components.* Examples include short periods of congestion within the network, isolated HTTP request/response failures or delays caused by application servers that may lead to different state at different downstream application servers.

3. Timeline issues with the content, due to slight time differences the playlists or manifest’s may list a different number of segments or have a different live edge, when a player switches from one AS to another AS it may therefore result in retroactive timeline changes and may potentially crash. These changes may happen because of different delays on different path or slightly different configurations on a path.

4. Segment replacement and substitution at the edge requires aligned timelines and segment durations to be successful, i.e. guidelines for appropriate content formatting and generalized content formatting may be needed, this case can enable redundant segment generation for resilience, watermarking, content replacement etc.

### 5.x.2 Collaboration scenarios

#### 5.x.2.1 Multi-AS media delivery with dynamic content generation

In this scenario, the 5GMSd Client requests adaptive media streaming content from two or more 5GMSd Application Servers. The Client may choose one 5GMSd AS or use multiple simultaneously. This allows the client to distribute network load across Application Servers and M4 downlink transports, optimize costs, as well as improve QoS.

The retrieved content over M4 includes both media segments and playlist/MPD, in some cases different AS may generate different groups of segments and/or different media playlists/manifests.

The client’s Media Session Handler discovers the URLs of these Application Servers from the 5GMSd Application Function (AF), either through a Media Entry Point or from a separate piece of metadata. QoE metrics from the client may be used by the AF to determine the best Application Server(s) for each client to use when streaming media. Figure 5.x.2.1-1 shows the client communicating with multiple Application Servers. Each AS has no direct communication with its peers; rather it communicates (minimally) with the Application Provider and with the 5GMSd AF (not depicted) via reference point M3d.

In this case the AS includes dynamic content generation and generates, based on the content retrieved on M2 new media segments and/or representations.

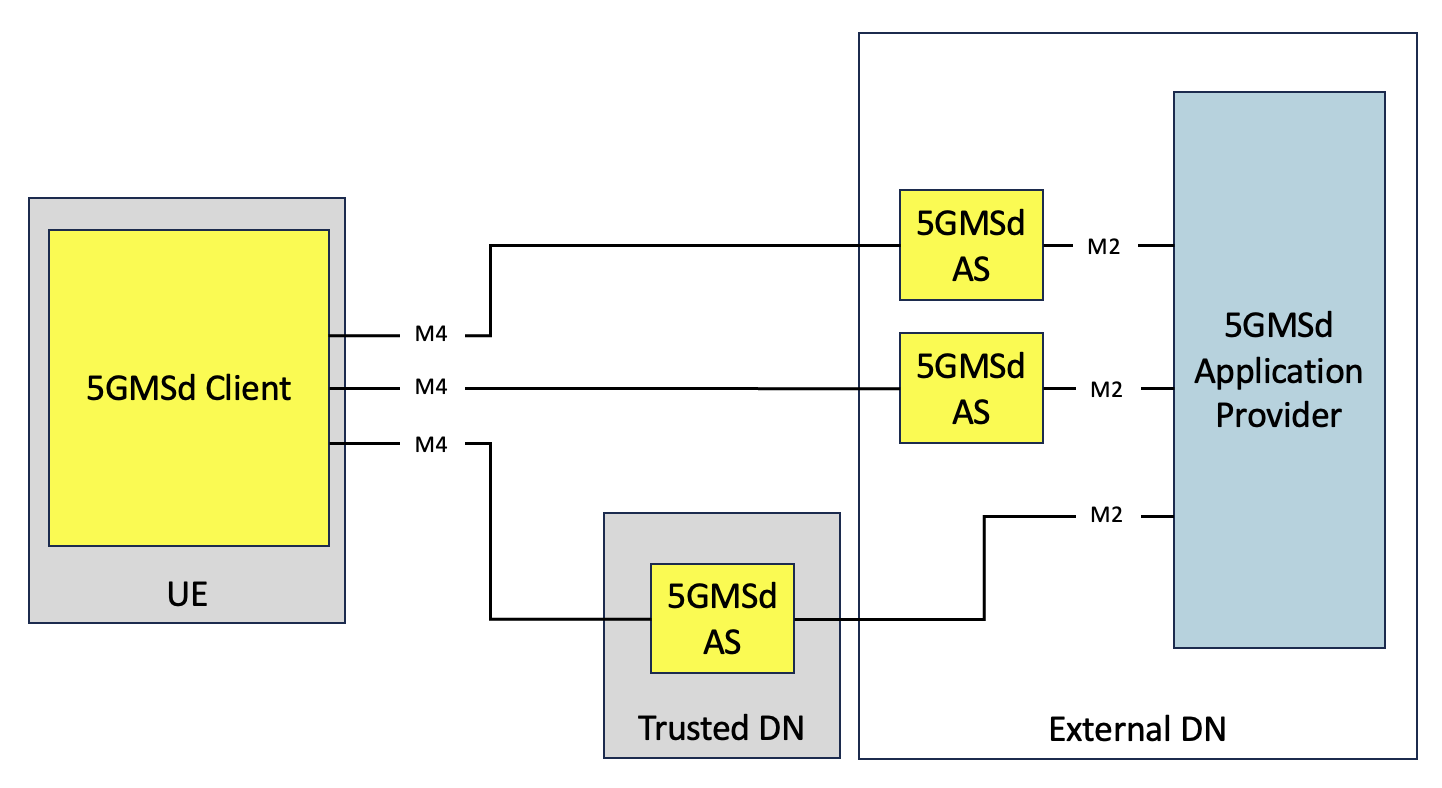


Figure 5.x.2.1-1: Multi-AS media delivery within 5G system with dynamic content generation

### 5.x.3 Architecture mapping

Editors Note: An architecture mapping will be provided once this CR is endorsed.

### 5.x.4 High-level call flow

Editors Note: A High level call flow will be provided once this CR is endorsed.

### 5.x.5 Gap analysis and requirements

#### 5.x.5.1 Dynamic Content Publishing from multiple sources (AS)

When streaming media presentations are originated from different sources, consistency of the media presentations is important.

This is trivial to achieve for the case of static non-changing content.

However, live media streaming presentation get updated frequently over time, resulting in different update media presentation descriptions (MPD), updated media playlists (for HTTP Live Streaming case), and different segments becoming available or no longer being available.

The support for such live streaming cases is important for multi-AS scenarios, as it is desirable that both media segments and media presentation descriptions are interchangeable even when they originate from different AS sources.

In practice, AS/CDN source A may have a slightly delayed version of a presentation compared to AS/CDN source B resulting in different segment availability, and potentially conflicting media presentation descriptions and media playlists at between the different AS/CDN sources.

Such timing differentiations may happen upstream in the workflow due to different delays or differently configured upstream components. Even if the delays are relatively modest, still discrepancies between sources may occur that are disruptive to the user experience.

Practice has shown, that players of streaming media presentations are sensitive to such issues. For example, retroactive changes to the media timeline may cause playback failure.

Figure YY shows the example case of a retroactive timeline changes caused by switching to an AS 2 that has a delayed version of the dynamic media presentation.

In step 1 the media presentation description is received with given MPD@publishTime, earliest media presentation time EPT, last segment presentation time LPT in the DVR window. The playback starts and after an MPD update request the DVR window is updated without problems as the update is consistent as it comes from the same sources. In a subsequent request the player switches to an alternative AS 2 and this has delayed input, now an updated MPD is received with a later publish time, but the earliest presentation time of the newest segment is decreased and the timeline is changed retro-actively.

As a consequence the playback is interrupted as players can usually not handle such cases.

Other related issues that may occur due to different source versions may be different timeline gaps in the media presentation and overall differences in the segment availability between subsequent updates.

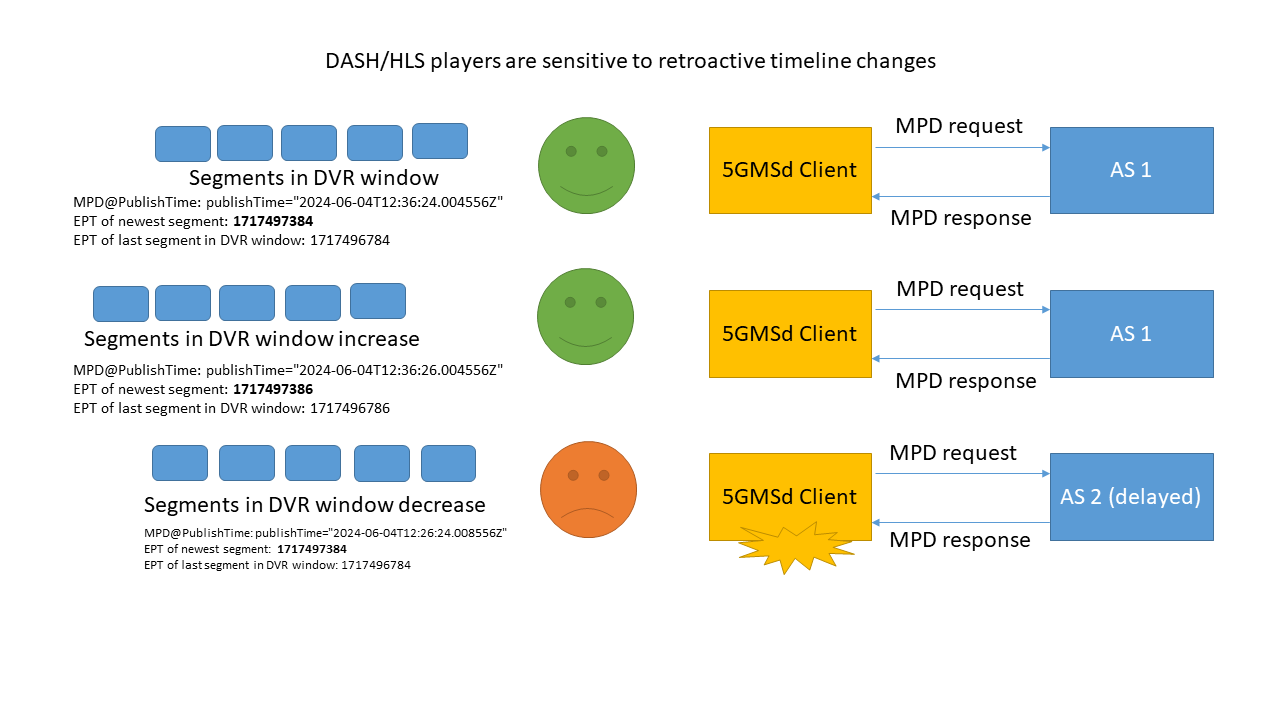


Figure YY Sample case of playback failure in multi-CDN delivery

#### 5.x.5.2 Examples of inconsistent Content Publishing from multiple sources (CDN)

In this clause we provide some examples how inconsistent timelines can occur in dynamic content publishing, even if the segments are interchangeable and both source contents have an identical timeline. The cause can be that content publishing origin B gets a delayed input compared to content publishing origin A.

Each dynamic content publishing origin generates the media playlist description based on its input and sets the publishTime based on the system clock time that is synchronized to global timing systems.

In this case the first MPD is retrieved from origin A that is ahead, in this case the newest segment time is

1031634228096 + 1152 x (312 + 1) = 1031634588672

Then the player receives the MPD from content publishing origin B in a delayed data center it gets an MPD with a later publish time, but the start time of the newest segment is:

1031634226944 + 1152 x (312 + 1) = 1031634587520

Which is earlier, this implies that the newest segment from the MPD from origin A is not the MPD, but the publishtime is later, leading to a retro-active timeline change interpreted by the player.

Table 1 Snapshot of publishTime and segmentTimeline in MPD retrieved from origin A

|  |
| --- |
| <?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 http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-DASH\_schema\_files/DASH-MPD.xsd"  type="dynamic"  availabilityStartTime="1970-01-01T00:00:00Z"  publishTime="2024-06-26T08:36:24.332711Z"  minimumUpdatePeriod="PT2S"  timeShiftBufferDepth="PT10M"  maxSegmentDuration="PT2S"  minBufferTime="PT10S"  profiles="urn:mpeg:dash:profile:isoff-live:2011,urn:com:dashif:dash264">  <Period  id="1"  start="PT0S">  <AdaptationSet  ……  <SegmentTemplate  timescale="600"  initialization="live-$RepresentationID$.dash"  media="live-$RepresentationID$-$Time$.dash">  <!-- 2024-06-26T08:26:20.160000Z / 1719390380 - 2024-06-26T08:36:21.120000Z -->  <SegmentTimeline>  <S t="1031634228096" d="1152" r="312" />  </SegmentTimeline>  ……  </MPD> |
|  |

Table 2 Snapshot of publishTime and segmentTimeline in MPD retrieved from origin B

|  |
| --- |
| <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 http://standards.iso.org/ittf/PubliclyAvailableStandards/MPEG-DASH\_schema\_files/DASH-MPD.xsd"  type="dynamic"  availabilityStartTime="1970-01-01T00:00:00Z"  publishTime="2024-06-26T08:36:25.378162Z"  minimumUpdatePeriod="PT2S"  timeShiftBufferDepth="PT10M"  maxSegmentDuration="PT2S"  minBufferTime="PT10S"  profiles="urn:mpeg:dash:profile:isoff-live:2011,urn:com:dashif:dash264">  <Period  id="1"  start="PT0S">  <AdaptationSet  ….  <SegmentTemplate  timescale="600"  initialization="live-$RepresentationID$.dash"  media="live-$RepresentationID$-$Time$.dash">  <!-- 2024-06-26T08:26:18.240000Z / 1719390378 - 2024-06-26T08:36:19.200000Z -->  <SegmentTimeline>  <S t="1031634226944" d="1152" r="312" />  </SegmentTimeline>  ….  </MPD> |

### 5.x.6 Candidate solutions

#### 5.x.6.Y Content Formatting at multiple AS sources following MPEG-DASH part-9 REaP

This solution proposed to solve the issue of dynamic content generation from multiple sources 5.x.5.y is comprised of using additional format constraints that can be applied to DASH and HLS streaming media presentations (both media segments and media playlist or media presentation description).

This solution is based on Redundant Encoding and Packaging as defined in [DASH9]. This solution defines formats for usage in the streaming head-end that typically resides in the application server enabling generation interchangeable media presentations at different sources.

In MPEG-DASH part 9 REaP this output is defined as the Delivery Media presentation Description (D-MPD), it adds constraints on the formatting and output generation of the media presentation description to make sure that outputs form different distributed origins application servers are interchangeable by a player (i.e. no retroactive timeline changes). These constraints apply to the representations, the setting of MPD@publishTime and some other attributes/elements in the MPD. A corresponding version for HTTP Live Streaming playlists is defined as well.

The solution starts from defining a reference/example head-end architecture, that in the case of 5G Media streaming may be the AS or even components before the AS. The key assumption is that a common contribution source (encoder) with a common timeline is used that can be converted back to a timeline relative to Unix Epoch, that is 00:00 1-1-1970 UTC. This assumption can hold in many cases where a common contribution signal is used in the streaming head-end. This way all tracks and streams have the same timeline origin. Secondly a regular media segmentation strategy is proposed resulting in aligned media segment boundaries of the different segments.

This is used in the output of 1 or more distribution encoders, that produce ISO BMFF tracks with aligned segment boundaries and a common shared timeline relative to Unix Epoch. In addition REaP defines the I-MPD, a constrained version of the media presentation description to announce the streams that is for example used for live ingest or distribution encoder egress.

The critical part are the distributed packagers in combination with the origin as this corresponds to the output generated at the AS or multiple AS’s in the multi-source dynamic content generation case.

For this case REaP defines the Delivery MPD (D-MPD) that constrains the formatting of the output MPD by linking some of the fields in DASH MPD that can cause retroactive changes to the media timeline. For example, by linking the MPD@publishTime explicitly to the earliest presentation time of the newest segment, and constraining the way the set of available segments are updated, it is possible to generate consistent media presentation description or HTTP live streaming playlists.

The Approach in REaP can work in modestly well synchronized AS’s up to 100 ms out of sync. A key aspect for a REaP setup is to configure the expected maximum delay incurred in the workflow/AS to account for this setting MPD@publishTime as a function of the earliest media presentation time of the latest segment at the live edge.

The approach has been shown using open source based implementation demo at popular conference in [MHV02] to implement a pseudo watermarking workflow, in pseudo watermarking the 2 AS sources create interchangeable content that only differs by an embedded watermark. In this demo the watermark was shown visually on the screen, showing how the player is able to seamlessly play content generated dynamically from different sources.

The advantage of using REaP as a solution is that changes will be limited, and the output format from the AS can be easily checked for conformance using existing conformance tooling. In addition, very limited or up to no signalling is needed between AS’s making the solution very easy to adopt and maintain in practice.

In REaP this problem case shown in 5.X.2 is overcome by coupling the publish time to the media timeline in the output to enable consistent dynamic content publishing between sources. If the publish time was coupled to the media time this specific case of retro-active timeline change could have been avoided, but it requires some further configuration.

Therefore some configurations from the REaP may be considered in 5GMS, such as segment duration, the epoch used, the synchronization time stamp for linking the publish time and media timeline.

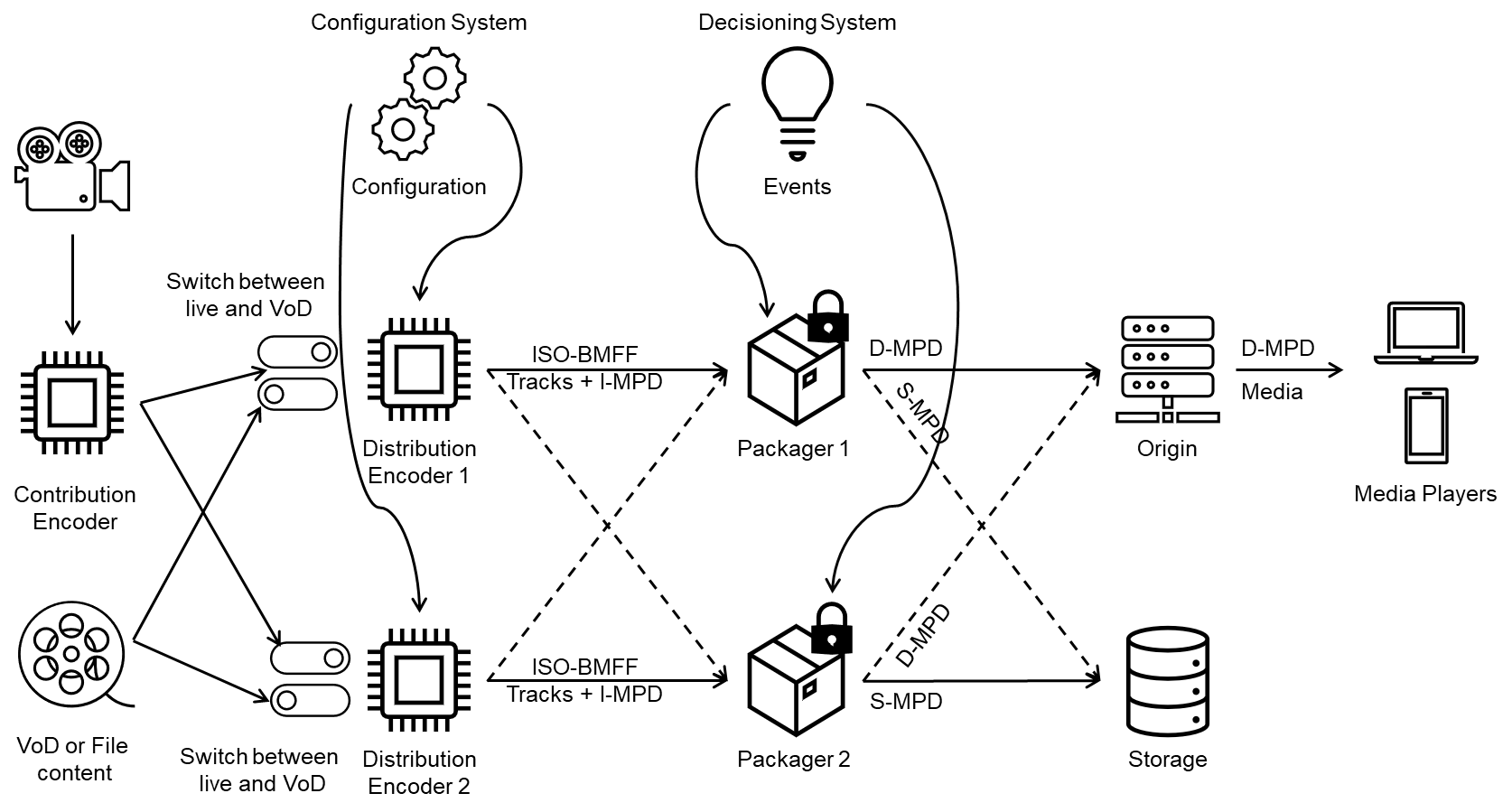


Figure X architecture for redundant encoding and packaging

### 5.x.7 Summary and Conclusions

## END OF CHANGES