**3GPP SA4#114-e *S4-210764***

**19-28 May 2021 revision of S4aI211184**

|  |
| --- |
| *CR-Form-v12.0* |
| **Pseudo CHANGE REQUEST** |
|  |
|  | **26.804** | **CR** | **<CR#>** | **rev** | **-** | **Current version:** | **0.2.1** |  |
|  |
| *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:***  | [FS\_5GMS-EXT] Updated text for Content Preparation |
|  |  |
| ***Source to WG:*** | Tencent |
| ***Source to TSG:*** | SA4 |
|  |  |
| ***Work item code:*** | FS\_5GMS-EXT |  | ***Date:*** | 2021-01-25 |
|  |  |  |  |  |
| ***Category:*** | **B** |  | ***Release:*** | Rel-17 |
|  | *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 canbe 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-12 (Release 12)**Rel-13 (Release 13)Rel-14 (Release 14)Rel-15 (Release 15)Rel-16 (Release 16)* |
|  |  |
| ***Reason for change:*** | The study item description identifies the key topic “Content Preparation”. |
|  |  |
| ***Summary of change:*** | Adding more challenging potential open issues. |
|  |  |
| ***Consequences if not approved:*** | Key topic not addressed |
|  |  |
| ***Clauses affected:*** |  |
|  |  |
|  | **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:*** |  |
| ***56***  |  |
| ***This CR's revision history:*** |  |

### 5.2.7 Potential open issues

#### 5.2.7.1 TS 26.512 current support

Table 7.6.3.1‑1 in TS 26.512 [16] defines the content hosting configuration resource.

Each Content Hosting Configuration may include one or more Distribution Configurations, each one with an optional Content Preparation Template. However, the Content Hosting Configuration is currently allowed to include only one Ingest Configuration.

The Ingest Configuration defines the input format for content preparation. Currently, two ingest protocols are specified in clause 8.2 of [16]: HTTP pull-based ingest and DASH-IF push-based. In the case of HTTP pull, if a request is received in M4d that can’t be satisfied, an HTTP pull request is made through M2.

#### 5.2.7.2 Open issues in collaboration scenario 1: Content preparation before downlink streaming

##### 5.2.7.2.1 Open issue 1: More complex requests

The following figures show two examples for content distribution.

5.2.7.2.1.1 Transcoding one-to-one video segments from one encoding format to another encoding format

Figure 5.2.7.2-1 shows the case in which the content preparation for every input segment has a single output segment. When the 5GMSd AS receives a media segment request from the 5GMSd Client, if the content is not already cached, it translates the M4d request URL to a corresponding M2d request URL using the rewrite rule specified in clause 7.6.3 TS 26.512 [16]. Then the corresponding media segment is requested, transcoded, and delivered to the 5GMSd Client. (The transcoded media segment also becomes available from the 5GMSd AS cache to satisfy future M4d requests.)

5GMSd AS

Content Hosting

**M2d**

Content Preparation

PathRewriteRules

Cache

**M4d**

5GMSd Client

5GMSd Application Provider

Request for Segment 246

Segment 246

Request for Segment 246

Segment 246

Figure 5.2.7.2-1: Example 1 of a media distribution by pull

Steps:

1. The 5GMSd Client requests a media segment <http://cdn.com/segment246.mp4> through M4d.

2. The request URL is mapped according to the PathRewriteRules in the Content Hosting distribution configuraiton (“cdn.com” => “originprovider.com/video/”) and <http://originprovider.com/video/segment246.mp4> is requested from the 5GMSd Application Provider through M2d.

3. The 5GMSd Application Provider provides the requested media segment to the Content Preparation subfunction of the 5GMSd AS, which transcodes the segment and delivers the output segment to the Cache subfunction.

4. The transcoded segment is delivered to the UE via M4d.

###### 5.2.7.2.1.2 Combining two video segments into one to increase segment duration before distribution

Figure 5.2.7.2-2 shows the case in which the content preparation takes segments *n* and *n+1* from the upstream 5GMSd Application Provider and combines them into one segment with double duration for delivery to the 5GMSd Client. The goal could be to achieve higher compression efficiency for distribution.

5GMSd AS

Content Hosting

**M2d**

Content Preparation

PathRewriteRules

Cache

**M4d**

5GMSd Client

5GMSd Application Provider

Request for Segment 246

Segment 246

Segment 247

Request for Segment 247

Segment 123

Request for Segment 123

Figure 5.2.7.2-2: Example 2 of a media distribution by pull

Steps:

1. The 5GMSd Client requests a media segment <http://cdn.com/segment123.mp4> through M4d.

2. The requested URL is mapped according to the PathRewriteRules in the Content Hosting distribution configuration (“cdn.com” => “originprovider.com/video/” and transformation of request URL using the rule {n} => {n, n+1}). Two separate requests are made via M2d to the 5GMSd Application Provider for the following media segments:

- <http://originprovider.com/video/segment246.mp4>

- <http://originprovider.com/video/segment247.mp4>

3. The 5GMSd Application Provider provides the two requested media segments to the Content Preparation subfunction, which in turn merges the two segments into one, delivers the output segment to the Cache.

4. The Cache subfunction delivers the merged media segment to the UE via an M4d response.

The PathRewriteRules currently defined in clause 7.6.3 of [16] do not address step 2 of this example (i.e. transforming one request into two different requests).

There are three possibilities for addressing this gap:

1. Keep PathRewriteRules as is, and therefore exclude content preparation cases with more than one input and/or with more complex translation to be used with push protocol(s) only.

2. Extend the current request URL transformation capabilities of the 5GMSd Content Hosting feature.

3. Keep PathRewriteRules as is and include an additional (and more sophisticated) address translation function as a part of the Content Preparation subfunction of the 5GMSd AS.

Figure 5.2.7.2-3 shows the architecture for the third option above.

5GMSd AS

Content Hosting

**M2d**

Content Preparation

PathRewriteRules

Cache

**M4d**

5GMSd Client

5GMSd Application Provider

Request for Segment 246

Segment 246

Segment 247

Request for Segment 247

Segment 123

Request for Segment 123

Address Translator

Figure 5.2.7.2-3: Example 2 of a media distribution by pull

Steps:

1. The 5GMSd Client requests a media segment <http://cdn.com/segment123.mp4> through M4d.

2. The requested URL is mapped according to the PathRewriteRules in the Content Hosting distribution configuration and provided to the address translator of the Content Preparation module.

3. The address translator module transforms the received request into two separate requests via M2d to the 5GMSd Application Provider for the following media segments:

- <http://originprovider.com/video/segment246.mp4>

- <http://originprovider.com/video/segment247.mp4>

4. The 5GMSd Application Provider provides the two requested media segments to the Content Preparation subfunction, which in turn merges the two segments into one, delivers the output segment to the Cache.

5. The Cache subfunction delivers the merged media segment to the UE via an M4d response.