**3GPP TSG-SA WG4 Meeting #122 S4-230341**

**Athens, Greece, 20 – 24 February 2023**

|  |
| --- |
| *CR-Form-v12.2* |
| **PSEUDO CHANGE REQUEST** |
|  |
|  | **26.565** | **CR** |  | **rev** |  | **Current version:** | **1.0.0** |  |
|  |
| *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 |  | Radio Access Network |  | Core Network |  |

|  |
| --- |
|  |
| ***Title:***  | CR on Harmonizing the SR reference architectures |
|  |  |
| ***Source to WG:*** | Qualcomm Incorporated, Tencent Cloud |
| ***Source to TSG:*** | S4 |
|  |  |
| ***Work item code:*** | SR\_MSE |  | ***Date:*** | 14-02-2023 |
|  |  |  |  |  |
| ***Category:*** | B |  | ***Release:*** | Rel-18 |
|  | *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-16 (Release 16)Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19)* |
|  |  |
| ***Reason for change:*** |  |
|  |  |
| ***Summary of change:*** |  |
|  |  |
| ***Consequences if not approved:*** |  |
|  |  |
| ***Clauses affected:*** |  |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  |  |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  |  |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  |  |  O&M Specifications | TS/TR ... CR ...  |
|  |  |
| ***Other comments:*** |  |
|  |  |
| ***This CR's revision history:*** | This PCR is a merge of the following documents:Architecture: 230160, 230133 Call flows: 230161, 230188API: 230162, 230134 |

|  |
| --- |
| 1st Change |

## 5.1 Reference Architecture

### 5.1.1 Introduction

In this clause, different variants of the reference architecture for the split rendering MSE are defined, each representing a different perspective and level of details.

The following functions are introduced:

* Split-Rendering Client (SRC): This function is responsible to acquire the UE media capabilities and negotiates with the 5GMS AS to agree on the split-rendering process at the 5GMS AS.
* Split-Rendering Server (SRS): This function is responsible for negotiation of SR session with SRC, monitoring the server’s edge resource usage, and managing/running the split rendering process.
* Application Function (AF): responsible for provisioning, QoS allocation, and edge resource discovery.
* Application Service Provider: The application provider that offers the service.
* Application: The application running on UE
* Media Session Handler (MSH): is the entity on UE that is responsible for the control plane communication with the AF.

*Editor’s Note: All interfaces are currently defined as SR-interfaces. However, if no SR-specific functionality is identified for an interface, the 5G-RTC interface reference will be used instead.*

### 5.1.2 Client Architecture

Figure 5.1.2-1 depicts the reference client architecture.



Split Rendering Client

Figure 5.1.2-1 – Device architecture of AR UE

The split rendering client consists of the following components:

* The Media Access Functions: allow for fetching and processing of the pre-rendered media in preparation of final display. The MAF is also responsible for the carriage of any metadata or local media to the split rendering server.
* The scene manager and thin Presentation Engine: is responsible for the negotiation of the split rendering session and the parsing of the description of the rendered media as provided by the SR server. It is also responsible for setting up and managing the XR session with the XR runtime.
* The XR source management is responsible for gathering timed metadata such as pose and action information and sending it to the SR server.

### 5.1.3 End-to-End Architecture



Figure 5.1-3 – Split management architecture

As shown in Figure 5.1.3:

1. The 5G Application Providers (AP) provisions the split-rendering through SR-1.

2. In the use cases in which the AP is involved in the media delivey, the SR-2 interface is used for this purpose.

3.The communication between AF and SRS is through SR-3. This interface is out of the scope of this document.

4. The communication as well as the media delivery between SRC and SRS is though SR-4.

5. The SRC discovers the client media capabilities through the SR-7 interface. This interface is out of the scope of this document.

6. The 5G Application and AP interactive through SR-8. This interface is out of the scope of this document.

### 5.1.5 User Plane Architecture

Figure 5.1.5-1 depicts the user plane architecture for split rendering.



The SR interfaces are considered to be specializations of their parent RTC interfaces as defined in TS26.506.

In the context of split rendering, the SR-4 interface is further classified as SR-4s and SR-4m sub-interfaces. The SR-4s interface covers all user-plane signaling, including WebRTC and ICE signaling. The SR-4m serves for media and metadata exchange between the split rendering client and the split rendering server.

## 5.2 Procedures and Call Flows

### 5.2.1 Call flow for Split Rendering instance discovery

#### 5.2.1.1 Client-driven procedures and call flows

Figure 5.2.1-1 demonstrates a call flow for setting up the split rendering by the client.



Figure 5.2.1-1: High-level call flow for initiating a split by client

Steps:

1. The Application Service Provider requests the provisioning a split management session.
2. The split management session is announced to the Application as part of the Service Access Information.
3. The Application requests a split of the client media functions from the SRC.
4. The SRC inquires the Media Session Handler about the client’s media capabilities.
5. The SRC and SRF negotiate on the acceptable capabilities for the device and agree on the split option.
6. The SRC starts the split-rendering process.
7. The SRF acknowledges the SRC that the split-rendering on edge is running and provides its access information.
8. The SRC acknowledges the Application that the split-rendering on edge is running.
9. The Application requests SRC to start the media delivery.
10. The SRC starts the media delivery.

#### 5.2.1.2 Network-driven procedures and call flows

Figure 5.2.1.2-1 demonstrates a call flow for setting up the split rendering by the network.



Figure 5.2.1.2-1: High-level call flow for the network-driven split management

Steps:

1. The Application Service Provider requests the provisioning of the split management session.
2. The split management session is announced to the Application as part of the Service Access Information.
3. The SRS offers the SRC the split the client media functions.
4. The SRC inquires the Media Session Handler about the client’s media capabilities.
5. The SRC and SRS negotiate on the acceptable capabilities for the device and agree on the split option.
6. The SRS starts the split-rendering process on the edge.
7. The SRS acknowledges the SRC that the split-rendering on edge is running and provides its access information.
8. The SRC announces to the Application that the split-rendering on edge is running.
9. The Application requests the SRC to start media delivery.
10. The SRC starts the media delivery.

### 5.2.2 Call flow for Split Rendering session setup

The split rendering operation can be described by the following call flow:

 ![Msc-generator~|version=8.2.0~|lang=signalling~|size=857x485~|text=#This is the default signalling chart.~n#Edit and press F2 to see the result.~n#You can change the default chart~n#with the leftmost button on the Preferences pane of the ribbon.~n~nnumbering=yes;~nhscale=auto;~n~nsrc: Split REndering Client (SRC) {~nxr[label=~qXR\nRuntime~q], sm[label=~qScene\nManager\n(thin PE)~q], xrsrc[label=~qXR source\nManagement~q], maf[label=~qMedia\nAccess\nFunction~q];~n};~nsrs[label=~qSplit\nRendering\nServer~q];~n~nsm-~gsrs: create a split rendering session;~nsrs-~gsm: send description of split rendering output;~nsm-~gmaf-srs: establish transport connections\ne.g. WebRTC session;~nxr..srs:Rendering Loop [number=no] {~n~2xr~l-~gxrsrc: receive pose information and user actions;~n~2xrsrc-~gsrs: transmit pose infomration and user actions;~n~2srs..srs: perform rendering\nfor requested posse;~n~2srs-~gmaf: send next buffer frame(s);~n~2maf..maf: decode and process\nbuffer frame;~n~2maf-~gsm-xr: pass raw buffer frames for display;~n~2xr..xr: compose and \nrender frame;~n};~|]()The steps are:

1. The Presentation Engine discovers the split rendering EAS and sets up a connection to it. It provides information about its rendering capabilities and the OpenXR configuration.
2. In response, the split rendering EAS creates a description of the split rendering output and the input it expects to receive from the UE.
3. The Presentation Engine requests the buffer streams from the MAF, which in turn establishes a connection to the split rendering EAS to stream pose and retrieve split rendering buffers.
4. The Source Manager retrieves pose and user input from the XR runtime.
5. The Source Manager shares the pose predictions and user input actions with the split rendering EAS
6. The split rendering EAS uses that information to render the frame
7. The rendered frame is encoded and streamed down to the MAF

### 5.2.2 Simple split-rendering negotiation

In Figure 5.2.1-1 and 5.2.1-2, step 5 defines the negotiation between the SRC and SRS for the split-rendering configuration. In most simple case, the SRC provides SRS the capabilities of the device and if SRS can accommodate the split-rendering processing that addresses the device, it confirms the configuration. In such scheme, the SRS is responsible to make the decision and no back-and-forth negotiation occurrs.

|  |
| --- |
| **Next Change** |

# 7 Split Rendering Application Function

## 7.1 Functionality

## 7.2 RESTful APIs

### 7.2.1 Split-Rendering Provisioning procedures (SR-1)

#### 7.2.1.1 General

These procedures are used by the Application Provider and the AF on SR-1 to provision for the split-rendering process.

#### 7.2.1.2 Create Split-Rendering Configuration

This procedure is used by the AP to create a new Split-Rendering Configuration. The AP shall use the HTTP POST method for this purpose and the request message body shall include a SplitRenderingConfiguration resource.

If the procedure is successful, the AF shall generate a resource identifier representing the new Split-Rendering Configuration. In this case, the AF shall respond with a 201 (Created) HTTP response message and shall provide the URL to the newly created resource in the Location header field. The response message body may include a SplitRenderingConfiguration resource that represents the current state of the Split Rendering Configuration, including any fields set by the AF.

If the procedure is not successful, the AF shall provide a response code as defined in clause XXYY.

#### 7.2.1.3 Read Split-Rendering Configuration properties

This procedure is used by the AP to obtain the properties of an existing Split-Rendering resource from theAF. The HTTP GET method shall be used for this purpose.

If the procedure is successful, the AF shall respond with a 200 (OK) response message that includes the SplitRenderingConfiguration resource in the response message body.

If the procedure is not successful, the AF shall provide a response code as defined in clause XXYY.

#### 7.2.1.4 Update Split-Rendering Configuration properties

The update operation is invoked by the AP to modify the properties of an existing SplitRenderingConfiguration resource. All writeable properties except domainNameAlias may be updated. The HTTP PATCH or HTTP PUT methods shall be used for the update operation.

If the procedure is successful, the AF shall respond with a 200 (OK) and provide the content of the resource in the response, confirming the successful update operation.

If the procedure is not successful, the AF shall provide a response code as defined in clause XXYY.

#### 7.2.1.5 Destroy Split-Rendering Configuration

This operation is used by AP to destroy a Split-Rendering Configuration resource and to terminate the related distribution. The HTTP DELETE method shall be used for this purpose. As a result, the AF will release any associated network resources, purge any cached content, and delete any corresponding configurations.

If the procedure is successful, the AF shall respond with a 200 (OK) response message.

If the procedure is not successful, the A.F shall provide a response code as defined in clause XXYY.

### 7.2.2 Split-Rendering Provisioning API

### 7.2.2.1 Overview

This clause specifies the API that the AP uses at interface SR-1 to provision and manage the SRC though the AF. Each such configuration is represented by a SplitRenderingConfiguration, for which the resource structure is specified in 7.2.2.2 and the data model is specified in clause 7.2.2.3.

### 7.2.2.2 Resource structure

The Split-Rendering Provisioning API is accessible through this URL base path:

{apiRoot}/3gpp-m1*/*{apiVersion}*/*provisioning-sessions/{provisioningSessionId}/

Table 7.2.2-1 below specifies the operations and the corresponding HTTP methods that are supported by this API. In each case, the Provisioning Session identifier shall be substituted into {provisioningSessionId} in the above URL template and the sub-resource path specified in the second column shall be appended to the URL base path.

Table 7.2.2‑1: Operations supported by the Split-Rendering Provisioning API

|  |  |  |  |
| --- | --- | --- | --- |
| Operation | Sub‑resource path | Allowed HTTP method(s) | Description |
| Create Split-Rendering Configuration | content-hosting-configuration | POST | Used to create a Split-Rendering Configuration resource. |
| Retrieve Split-Rendering Configuration | GET | Used to retrieve an existing Split-Rendering Configuration. |
| Update Split-Rendering Configuration | PUT,PATCH | Used to modify an existing Split-Rendering Configuration. |
| Delete Split-Rendering Configuration | DELETE | Used to delete an existing Split-Rendering Configuration. |
| Purge Split-Rendering Configuration cache | content-hosting-configuration/purge | POST | This operation is used to invalidate some or all cached media resources associated with this Split-Rendering Configuration. |

### 7.2.2.3 Data model

#### 7.2.2.3.1 SplitRenderingConfiguration resource

The data model for the SplitRenderingConfiguration resource is specified in table 7.2.2-2 below:

Table 7.2.2-2: Definition of SplitRenderingConfiguration resource

| Property name | Data Type | Cardinality | Description |
| --- | --- | --- | --- |
| Name | String | 1..1 | A name for this Split Rendering Configuration. |
|  Status | Boolean | 1..1 | Indicates whether to the SRS shall use Split-Rendering if possible |
|  Configuration | Object | 1..1 | Describes the split-rendering configuration currently used by the SRS |

|  |
| --- |
| **Next Change** |

# 10 Security and Privacy Aspects

## 10.1 Security

Signaling for session establishment and exchange of application-specific messages shall use a secure transport channel based on WebSockets as defined in TS26.113.

Media transport shall be secured by the usage of WebRTC.

## 10.2 Privacy

Editor’s Note: Privacy considerations are FFS.