**3GPP TSG-SA WG4 Meeting #129** **S4-241532**

**eMeeting, 19th - 23th  August 2024**

|  |
| --- |
| *CR-Form-v12.2* |
| **CHANGE REQUEST** |
|  |
|  | **26.567** | **pCR** |  | **rev** |  | **Current version:** | **0.2.0** |  |
|  |
| *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:***  | Network Centric Procedures |
|  |  |
| ***Source to WG:*** | Nokia  |
| ***Source to TSG:*** | S4 |
|  |  |
| ***Work item code:*** | SR\_IMS |  | ***Date:*** | 13 August 2024 |
|  |  |  |  |  |
| ***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 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:*** | Minor editorial corrections and addition of generic network assistance procedure |
|  |  |
| ***Summary of change:*** | Addition of text in clause 7.3 and minor editorial corrections in clause 7.1 |
|  |  |
| ***Consequences if not approved:*** | No text in clause 7.3 |
|  |  |
| ***Clauses affected:*** | 7.3, 7.1 |
|  |  |
|  | **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:*** |  |

|  |
| --- |
| Change 1 |

7 Procedures

7.1 General procedures for session establishment

*Editor’s note: session establishment procedures*



Figure7.1-1: high level call flows for split rendering over IMS.

The steps are as follows:

Step 1: The UE1 initiates a media session and establishes audio and video session connections with the UE2. Then the bootstrap and application data channels are established for the UE1 and UE2.

Step 2: UE1 sends a request to create a split rendering session leveraging the IMS network entities:

 When the UE1 discovers that its media capabilities cannot meet the related media rendering requirements, the UE1 decides to start split rendering call flow. Then the UE1 calculates which objects can be rendered by itself based on its status and decides which part of the objects to be rendered in the UE1 and the others to be rendered in the IMS network.

 The UE1 initiates the application data channels between the UE1 and the IMS AS, for the split rendering request and metadata transmission.

Step 3: The IMS AS interacts with the DCSF via DC1 for event notifications.

Step 4: The DCSF receives event reports from the IMS AS and decides whether data channel service is allowed to be provided during the IMS session. The DCSF manages bootstrap data channel and (if applicable) application data channel resources at the MF via the IMS AS;

Step 5 and 6: The IMS AS receives the data channel control instructions from the DCSF and accordingly interacts with the MF via DC2

Step 7: The IMS AS sends a Split Rendering Request to the DC AS via the MF through the established application data channel, the request includes the information of the objects to be rendered in IMS network.

Step 8: The DC AS sends a description of the split rendering output to the IMS AS via the MF.

Step 9: The IMS AS sends the media resource allocation request to the DCSF, to reserve XR media rendering resource for the UE1.

Step 10: When the resources are allocated successfully, the DCSF returns a successful response to the IMS AS.

Step 11: The IMS AS returns a successful response to the UE1.

Step 12: Successful SR session is established between UE1and MF through the application data channel.

Step 13: Subsequent procedures continue for the UE2.

|  |
| --- |
| Change 2 |

7.3 Network support procedures

*Editor’s note:* *procedures for adaption of split rendering client and server based on network support.*

An SR-DCMTSI client or an MF may trigger further procedures during a split rendering, One such procedure may be to adapt the split of rendering operations due to change in operating conditions of the split rendering session, including operating conditions of the UE, the MF or changes in the application or scene being rendered, for example changes in the scene description. Split adaptation may include data exchange, for example, exchange of adaptation messages, application state information and assets needed for the split rendering of an DC application. The following generic procedure shall apply, while the exact details may depend on the DC-application being rendered.



The steps are as follows:

Step 1: The IMS session is established between the SR-DCMTSI client in terminal and a terminating SR-DCMTSI Client which may be in a terminal. For Person to Person calls, procedures in clause 7.1 are followed.

Step 2: A split rendering session is set up between the SR-DCMTSI client and a serving MF.

Step 3: Assets related to the application being split rendered may be delivered to participants of the split rendering session. The asset delivery may include javascript assets, scene descriptions and graphical objects needed for the session.

Rendering Loop:

The rendering loop is executed continuously during the duration of the split rendering session, for each frame.

Step 4: The SR-DCMTSI client in terminal sends metadata required for rendering to the MF. The metadata may include pose, pose predictions, user inputs etc.

Step 5 and 6: The SR-DCMTSI client in terminal and the MF render the frame.

Step 7: The frame rendered by the MF is transmitted to the SR-DCMTSI client in terminal as well as possible metadata.

Step 8: The SR-DCMTSI client in terminal composes a display frame from the received rendered media and media rendered locally.

NOTE: Steps 5,6,7 although ordered above, may occur in any order. Step 8 may include pose-correction. Step 8 and 6 may be executed as a single step.

Further Procedures:

 Split Adaptation:

Step 9: A trigger to adapt the split occurs at the SR-DCMTSI client in terminal; the trigger may e.g. be a change in available UE resources, changes in QoE of the SR session, changes in the scene/application being rendered.

Step 10: The SR-DCMTSI client in terminal decides if a new split of the rendering operations is needed and determines the new split.

Step 11: The SR-DCMTSI client in terminal sends a request to the MF to adapt the split to the new split.

Step 12: The MF actuates the new split of the rendering operations.

Step 13: The MF sends an acknowledgment of the new split to the SR-DCMTSI client in terminal.

Step 14: The MF and UE may exchange messages and data to support the new split of operations. This may include exchange of messages, for example, for synchronization of the state of the scene being split rendered or exchange of assets, for example, those in Step 3.

Step 15: The rendering loop (steps 4 through 12) continues

Note:  Split adaptation is shown to be initiated by the SR-DCMTSI client in terminal for clarity, the procedure may be triggered by the MF. Further, other procedures to actuate the new split may be executed during the split rendering session.

|  |
| --- |
| End of Changes |