**3GPP TSG-S4 Meeting #131 S4-250159R2**

**Geneva, 17-21 February 2025**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.2* | | | | | | | | |
| **PSEUDO CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.567** | **CR** |  | **rev** |  | **Current version:** | **0.4.1** |  |
|  | | | | | | | | |
| *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:*** | [SR\_IMS] Metadata formats for adaptive split rendering | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Nokia | | | | | | | | | |
| ***Source to TSG:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | SR\_IMS | | | | |  | ***Date:*** | | | 2025-02-11 |
|  |  | | | |  | |  | | |  |
| ***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:*** | | To add additional metadata for adaptive split rendering. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | Add metrics reporting format in Annex A.1. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Lack of the metadata format regarding interactive objects. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | Annex A.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:*** | |  | | | | | | | | |

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

7.3 Network support procedures

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 between the SR-DCMTSI client and the MF during a split rendering session. This split may be due to change in operating conditions of the split rendering session, for example 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.



* **Figure7.3-1: General procedures for adaptation of split rendering**

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:

NOTE 1: Split Adaptation refers to adaptation of split rendering operations in an ongoing split rendering session between the SR-DCMTSI client and MF, without impacting the MF resources provisioned by IMS AS in step 9 of clause 7.1.

NOTE 2: Adaptation of split rendering may be used for interactive objects that react to user actions, pose, eye gaze, eye status, etc. The metadata (e.g. level of interaction, and actions as well as status of the user) may change during the lifetime of the session, the rendering is appropriately adapted according to the updated metadata. An SR-DCMTSI client that supports the adaptive split rendering based on eye status information [shall] support the message format defined in Annex A.1.X.

Step 9: A trigger to adapt the split occurs at the SR-DCMTSI client in terminal; the trigger may be, for example, a change in available UE resources (e.g. battery, compute) or updated metadata, changes in QoE of the split rendering 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.

The meta-data messages exchanged shall follow the formats specified in clause 5.4

Step 15: The rendering loop (steps 4 through 8) 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.

|  |
| --- |
| **2nd Change** |

A.1.X Adaptive split rendering with eye status information interactive objects



If an SR-DCMTSI client that supports the adaptive split rendering with eyes status information, it shall support the message format defined in Table A.1.X-1. The eye status information shared by SR-DCMTSI client to MF during the adaptation procedure shall be in JSON format according to the Metadata Data Channel Message Format defined in clause 5.4.3. The message type shall be “urn:3gpp:split-rendering:v1:sr-split-eyeinfo”.

**Table A.1.X-1 Message format for eyes status information**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Cardinality** | **Description** |
| eyesInfo | Object | 1..1 | An array of eye information objects corresponding to past and current eyes status for the viewer. |
| eyesStatus | number | 1..1 | The current eyes status, e.g:  0: Eyes are open  1: Eyes are closing  2: Eyes are closed  3: Eyes are opening |
| eyesStatistics | Object | 1..1 | The current viewer eyes statistics |
| averageDuration | number | 1…3 | Array of average eyes duration for the viewer, e.g.:  [1] represents average closing time,  [2] represents average closed time  [3] represents average opening time |
| averageInterval | number | 1..1 | Average interval between two eye blinking. |
| elapsedTime | Number | 0…n | Time expressed in milliseconds since the last eyes open status. |

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