**3GPP TSG-SA WG4 Meeting** **#131** ***S4-250197***

**Geneva, Switzerland, 17th – 21st Feb, 2025**

**Source: Nokia**

**Title:** **[SR\_IMS] pCR Split Rendering Configuration**

**Spec:** **3GPP TS 26.567 v0.4.1**

**Agenda item: 10.5**

**Document for: Discussion and agreement**

1. **Introduction**

Draft TS 26.567 specifies split rendering session establishment procedures in clause 7.1. This includes media capability exchange over SDP. However, split rendering session configuration information needs to be communicated to the MF or DC AS for successful operation of a split rendering session between the SR-DCMTSI client and MF or DC AS.

1. **Reason for Change**

Addition of a split rendering configuration format.

1. **Proposal**

It is proposed to agree the following changes to TS 26.567 v.0.4.1

\* \* \* First Change \* \* \* \*

5.4.2 Metadata Formats

5.4.2.1 General

SR-DCMTSI client and Media Function shall support the usage of the IMS data channel for the exchange of split rendering metadata with the MF. The data channel shall declare “3gpp-sr” as the data channel sub-protocol. The message content format depends on the type of the message. The data channel sub-protocol is defined in clause 8.3.3 of TS 26.565 [5].

5.4.2.2 Pose Format

For XR services, the pose information format that is used for IMS-based split rendering [shall] comply with the format defined in clause 12.2 of TS 26.119 [6]. The pose information [shall] be carried as part of the data channel messaging mechanism. The metadata data channel message format is as defined in clause 8.3.3 of TS 26.565 [5]. The message type [shall] be “urn:3gpp:split-rendering:v1:pose”.

5.4.2.3 Action Format

The action information format that is used for IMS-based split rendering [shall] comply with the format defined in clause 12.3 of TS 26.119 [6]. The action information [shall] be carried as part of the data channel messaging mechanism. The metadata data channel message format is as defined in clause 8.3.3 of TS 26.565 [5]. The message type [shall] be “urn:3gpp:split-rendering:v1:action”.

5.4.2.4 Split Rendering Configuration Format

The SR-DCMTSI client and Media Function shall support the split rendering session configuration defined in clause Annex A.1.X.

If DC AS is in the media path performing split rendering, it shall support the split rendering session configuration defined in clause Annex A.1.X.

The split rendering configuration message shall be identified as “urn:3gpp:split-rendering:v2:sr-configuration”

\* \* \* First Change \* \* \* \*

A.1 Supported Metadata Formats and Message Types

This annex defines the metadata and metadata message types supported by this specification. Metadata formats and meta data channel message types and formats supported in this specification re-use or modify formats and message types defined in other 3GPP specifications as defined in Table A.1-1.

**Table A.1-1 Formats and relationship with 3GPP specifications**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Format | Source specification | Clause in source specification | Modified for this specification | Clause in this specification | urn |
| Split Rendering Configuration | TS 26.565 | 8.4.2 | Yes | A.1.X | urn:3gpp:split-rendering:v2:sr-configuration |
| Pose | TS 26.119 | 12.2 | No | 5.4.2.2 | urn:3gpp:split-rendering:v1:pose |
| Action | TS 26.119 | 12.3 | No | 5.4.2.3 | urn:3gpp:split-rendering:v1:action |
| Metadata Data Channel Message | TS 26.565 | 8.3.3 | No | 5.4.3 | NA |
| Split Adaptation Message | TS 26.565 | C.2.3.2 | Yes | A.2.X | urn:3gpp:split-rendering:v2:asrp:sr-split |
| Seamless Adaptive Split | TS 26.565 | C.2.3.2 | Yes | A.2.Y | urn:3gpp:split-rendering:v1:asrp:sr-split-seamless |
| State Synchronization Message | TS 26.565 | C.2.3.3 | No | To be filled | urn:3gpp:split-rendering:v2:sr-state |

*Editor’s Note: The clause may contain all message types for XR and other services. The suitable message types from TS 26.565 need to be referred or modified and imported to this spec as appropriate. If there is a need to further define profiles with support for specific messages as mandatory/optional is FFS.*

\* \* \* 2nd Change All NEW TEXT \* \* \* \*

A.1.X Split Rendering Configuration

The SR-DCMTSI client shall send a split rendering session configuration information to the MF and if applicable, to the DC AS after successful establishment of a split rendering session and before starting the rendering loop. The session configuration shall be in JSON format and shall follow the format in Table A.1.X-1

**Table A.1.X-1 Split Rendering Configuration Format**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Cardinality** | **Description** |
| renderingFlags | Array(SR\_CONFIG\_FLAGS) | 0..1 | Provides a set of flags to activate/deactivate selected rendering functions. The defined SR\_CONFIG\_FLAGS are:   * FLAG\_ALPHA\_BLENDING * FLAG\_DEPTH\_COMPOSITION * FLAG\_EYE\_GAZE\_TRACKING |
| splitRenderingFeatures | array( SR\_FEATURE\_FLAGS) | 0..1 | A list of split-rendering features supported by the SR-DCMTSI client. The supported features are:   * ADAPTIVE * SEAMLESS\_ADAPTIVE * DELAY\_ADAPTIVE |
| deviceCapabilities | Object | 0..1 | If the SR-DCMTSI client is implemented by a device defined in TS 26.119 [4], clause 6.2.5., device capabilities may be listed here |
| spaceConfiguration | Object | 0..1 | The space configuration is typically sent by the split rendering server to the split rendering client. Upon reception of this information, the SR client uses this information to create the reference and action spaces as well as to agree on common identifiers for the XR spaces. |
| referenceSpaces | Array | 0..1 | An array of reference spaces and their identifiers. |
| id | number | 1..1 | A unique identifier of the XR space in the context of the split rendering session. |
| refSpace | enum | 1..1 | One of the defined reference spaces in OpenXR. These may be: XR\_REFERENCE\_SPACE\_TYPE\_VIEW, XR\_REFERENCE\_SPACE\_TYPE\_LOCAL, or XR\_REFERENCE\_SPACE\_TYPE\_STAGE. |
| actionSpaces | Array | 0..1 | An array of action spaces that need to be defined by the split rendering client in the XR session. |
| id | number | 1..1 | A unique identifier of the XR space in the context of the split rendering session. |
| actionId | number | 1..1 | Provides the unique identifier of the action. |
| subactionPath | string | 1..1 | The subaction path identifies the action, which can then be mapped by the XR runtime to user input modalities. |
| initialPose | Pose | 0..1 | Provides the initial pose of the new XR space’s origin. |
| viewConfiguration | Object | 0..1 | Conveys the view configuration that is configured for the XR session. |
| type | Enum | 1..1 | The type indicates the view configuration. Defined values are MONO and STEREO. Other values may be added. |
| width | number | 1..1 | The recommended width of the swapchain image. |
| height | number | 1..1 | The recommended height of the swapchain image. |
| compositionLayer | string | 1..1 | An identifier of the selected composition layer. |
| minPoseInterval | number | 0..1 | The minimum time interval between two consecutive pose information instances sent to the network, in milliseconds. |
| fovs | Array | 0..1 | An array that provides a list of the field of views (FoV) associated with each view. |
| fov | Object | 1..n | Indicates the four sides of the field of view used for the projection of the corresponding XR view. The number of views n is determined by the *type* enum of the *viewConfiguration*. Both the *viewPoses* in the Pose Format and the *fovs* arrays shall be ordered in a consistent way (i.e., a same index can be used to retrieve the view pose and the related FoV information). |
| angleLeft | number | 1..1 | The angle of the left side of the field of view. For a symmetric field of view this value is negative. |
| angleRight | number | 1..1 | The angle of the right side of the field of view. |
| angleUp | number | 1..1 | The angle of the top part of the field of view. |
| angleDown | number | 1..1 | The angle of the bottom part of the field of view. For a symmetric field of view this value is negative. |
| environmentBlendMode | enum | 1..1 | The type indicates the environment blend mode configuration. Defined values are OPAQUE, ADDITIVE and ALPHA\_BLEND. Other values may be added. |
| actionConfiguration | Array | 0..1 | This contains a list of the actions that are to be defined by the SR client. |
| action | Object | 1..n | A definition of a single action object. |
| id | number | 1..1 | A unique identifier of the action. |
| actionType | enum | 1..1 | The type of the action state. This can be a Boolean, float, vector2, pose, vibration output, etc. |
| subactionPaths | string | 1..n | An array of subaction paths associated with this action. The split rendering client will provide the state of all defined sub-action paths. |
| extraConfigurations | Object | 0..1 | A placeholder for addition configuration information. |

\* \* \* End Changes \* \* \* \*