**3GPP TSG-WG SA4 Meeting #124**

**Berlin, DE, 22nd – 26th May 2023**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.2* | | | | | | | | |
| **PSEUDO CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.565** | **CR** |  | **rev** |  | **Current version:** | **0.4.0** |  |
|  | | | | | | | | |
| *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:*** | [SR\_MSE] Pixel Streaming Media Profile | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Qualcomm Inc. | | | | | | | | | |
| ***Source to TSG:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | SR\_MSE | | | | |  | ***Date:*** | | | 16th May 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 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-16 (Release 16) Rel-17 (Release 17) Rel-18 (Release 18) Rel-19 (Release 19)* | |
|  |  | | | | | | | | | |
| ***Reason for change:*** | | Moves the SR profiles to Annex C and defines SR-related metadata (already in MeCAR PD). | | | | | | | | |
|  | |  | | | | | | | | |
| ***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:*** | |  | | | | | | | | |

|  |
| --- |
| **First Change** |

### 8.2.1 General

This clause defines media and metadata formats that are common to one or more split rendering profiles.

#### 8.2.2 Metadata Formats

#### 8.2.2.1 General

In the “3gpp-sr” data channel sub-protocol, the message content depends on the type of the message. The data channel sub-protocol is defined in clause X.

Message types shall be unique identifiers in the URN format. This clause defines a set of message types and their formats. The messages are derived from the OpenXR API to ensure smooth operation with AR devices that support OpenXR. In case other XR APIs are used, mapping the message payload to the appropriate XR API structures shall be performed by the split rendering client.

Editor’s Note: This following sections will potentially reference the corresponding formats in 26.119.

#### 8.2.2.2 Pose Format

The split rendering client on the XR device periodically transmits a set of pose predictions to the split rendering server. The type of the message shall be set to “**urn:3gpp:split-rendering:v1:pose**”.

Each predicted pose shall contain the associated predicted display time and an identifier of the XR space that was used for that pose.

Depending on the view configuration of the XR session, there could be different pose information for each view.

The payload of the message shall be as follows:

Table 8 - Pose Prediction Format

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Cardinality** | **Description** |
| poseInfo | Object | 1..n | An array of pose information objects, each corresponding to a target display time and XR space. |
| displayTime | number | 1..1 | The time for which the current view poses are predicted. |
| xrSpace | number | 0..1 | An identifier for the XR space in which the view poses are expressed. The set of XR spaces are agreed on between the split rendering client and the split rendering server at the setup of the split rendering session.  The set of XR spaces is negotiated as part of the split rendering configuration as defined in clause 8.4.2.2. |
| viewPoses | Object | 0..n | An array that provides a list of the poses associated with every view. The number of views is determined during the split rendering session setup between the split rendering client and server, depending on the view configuration of the XR session. |
| pose | Object | 1..1 | An object that carries the pose information for a particular view. |
| orientation | Object | 1..1 | Represents the orientation of the view pose as a quaternion based on the reference XR space. |
| x | number | 1..1 | Provides the x coordinate of the quaternion. |
| y | number | 1..1 | Provides the y coordinate of the quaternion. |
| z | number | 1..1 | Provides the z coordinate of the quaternion. |
| w | number | 1..1 | Provides the w coordinate of the quaternion. |
| position | Object | 1..1 | Represents the location in 3D space of the pose based on the reference XR space. |
| x | number | 1..1 | Provides the x coordinate of the position vector. |
| y | number | 1..1 | Provides the y coordinate of the position vector. |
| z | number | 1..1 | Provides the z coordinate of the position vector. |
| fov | Object | 1..1 | Indicates the four sides of the field of view used for the projection of the corresponding XR view. |
| 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. |

#### 8.2.2.3 Action Format

Actions are grouped into action sets, which may be activated and deactivated during the lifetime of an XR session. The action sets and actions are negotiated at the start of the split rendering session.

The split rendering client reports any changes to action state as soon as it occurs by sending a message of the type “**urn:3gpp:split-rendering:v1:action**”.

The content of the action message type shall follow the following format:

Table 9 - Action Format

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Cardinality** | **Description** |
| actionSets | Object | 1..n | An array of active action sets, for which there is at least an action that has a state change. |
| actions | Object | 1..n | An array of objects that conveys information about the actions of the parent action set. |
| identifier | string | 1..1 | A unique identifier of the action that was agreed upon during split rendering session setup. |
| subactionPath | string | 1..1 | The sub-action path for which the state has changed. It abstracts a binding between an action and the hardware input associated to it by the XR runtime. |
| state | object | 1..1 | The state of the action that had a change in state. |
| lastChangeTime | number | 1..1 | The timestamp of the last change to the state of this action. |
| currentStateBool | Bool | 0..1 | The current Boolean state of the action |
| currentStateNum | number | 0..1 | The current numerical state of the action. |
| currentStateVec2 | Array | 0..1 | An array of numerical state values for the action. |

|  |
| --- |
| **Second Change** |

Annex C (normative):  
Split Rendering Profiles

## C.1 Pixel Streaming Profile

### C.1.1 Overview

The full-prerendering profile is restricted to the support of 2D content exclusively. The capabilities of the receiving UE are shared with the split rendering server prior to the start of the split rendering session. These capabilities and configurations would indicate the audio-visual output setup on the UE. For example, it would indicate that the output device is an HMD that supports 2 views and stereo audio.

### C.1.2 Downlink Formats

**HEVC-Dec-1:** the SRC shall support the decoding of H.265 (HEVC) Main Profile level 4 bitstreams. The SRC should support the decoding of H.265 (HEVC) Main 10 Profile level 5 bitstreams.

**HEVC-DEC-2:** the SRC shall support the simultaneous decoding of two bitstreams that comply with media capability HEVC-DEC-1. The SRC should support the simultaneous decoding of three bitstreams that comply with media capability HEVC-DEC-1.

**HEVC-ENC-1:** the SRS shall support the simultaneous encoding of 2 H.265 (HEVC) Main Profile level 4 bitstreams.

**HEVC-ENC-2:** the SRS shall support the encoding of monochrome video for alpha and depth information using H.265 (HEVC) Main Profile level 4.

AUDIO-DEC-1: the SRC shall support the decoding of MPEG-4 Low Delay AAC v2 Profile (AAC-ELDv2) Level 2 bitstreams [AAC-ELDv2].

AUDIO-ENC-1: the SRS shall support the encoding of a stereo audio signal to a bitstream that is decodable by an AUDIO-DEC-1 decoder using MPEG-4 Enhanced Low Delay AAC v2 (AAC-ELDv2).

NOTE: AAC-ELDv2 without SBR has an algorithmic delay (including framing delay) < 16ms (depending on frame size)

**AUDIO-ENC-2:** the SRS shall support the encoding of an audio stream using EVS with a total algorithmic latency no higher than 32 milliseconds.

**AUDIO-DEC-2**: the SRC should be able to decode an EVS stream with input sampling rate of up to 48kHz, operating on full band audio bandwidth at bitrates up to 128kbps.

Editor’s Note: This section will potentially reference the corresponding capabilities in 26.119.

### C.1.3 Uplink Formats

**XR-Pose-Cap 1:** the SRC shall be able to retrieve one or more pose predictions for each view and for every frame to be rendered. The pose predication shall be formatted according to clause 8.2.2.2.

**XR-Pose-Cap 2:** the SRC shall be able to retrieve and collect the user actions that occurred during an identified time interval. The action information shall be formatted according to clause 8.2.2.3.