|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.2* | | | | | | | | |
| **PSEUDO CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.565** | **CR** |  | **rev** |  | **Current version:** | **1.2.0** |  |
|  | | | | | | | | |
| *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 |  | Radio Access Network |  | Core Network |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |
| ***Title:*** | QoE metrics reporting for Split Rendering Client | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | InterDigital France R&D, SAS | | | | | | | | | |
| ***Source to TSG:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | SR\_MSE | | | | |  | ***Date:*** | | | 20th May 2024 |
|  |  | | | |  | |  | | |  |
| ***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:*** | | The TS 26.565 does not address the metrics reporting protocol, format and the scheme used for reporting the QoE/latency metrics from a split-rendering client (SRC) to the metrics collection server. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | This pCR proposes the list of QoE metrics that needs to be reported by a split rendering client and the quality metrics reporting protocol, format and the updates to the metrics scheme. | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | |  | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | Update to clause 9 | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **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:*** | | The 5GMS protocols specification TS 26.510 discusses the procedures and APIs for metric measurement, collection and reporting in the 5G system networks for 5GMS and RTC services.  The TS 26.565 specifications describes the list of Quality of Experience (QoE) metrics used for split-rendering client. The TS 26.565 does not address the metrics reporting protocol, format and the scheme used for reporting the QoE/latency metrics from a split-rendering client (SRC) to the metrics collection server.  The TS 26.113 specification defines the Metrics Reporting Provisioning API which allows the Application provider to configure the metrics collection and reporting procedures for a real-time media delivery session and defines the Quality reporting scheme for configuring the UE to send the QoE metrics reports and Quality reporting protocol for reporting the QoE metrics from the UE to the Metrics server.  This contribution proposes the QoE metrics to be reported to the metric collection server by an SRC or endpoint. This contribution also proposes the QoE metrics reporting protocol for sending the latency metrics by the SRC to the metrics collection server. This contribution also proposes to extend the metrics reporting configuration resource defined in TS 26.510 and the Quality reporting scheme defined in TS 26.113 for configuring an SRS to measure and report the QoE/latency metrics. | | | | | | | | |
|  | |  | | | | | | | | |
| ***This CR's revision history:*** | |  | | | | | | | | |

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

# References

[17] 3GPP TS 26.247: "Transparent end-to-end Packet-switched Streaming Services (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH)".

|  |
| --- |
| **End of first Change** |

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

# 9 Split Rendering Client

## 9.3 Split Rendering Metrics

### 9.3.1 QoE Metrics reporting protocol

The Metrics Reporting API allows the Media Session Handler to send QoE metrics reports to the metrics collection server.

A SR UE supporting Quality of Experience shall report QoE metrics according to the QoE configuration. QoE reporting is optional, but if an MSH reports QoE metrics, it shall report all requested metrics.

### 9.3.2 QoE metrics definition

#### 9.3.2.1 Introduction

This clause extends clause 15.2 of TS 26.113 that provides the general QoE metric definitions and measurement framework. A SR UE supporting the QoE metrics feature shall support the reporting of the metrics in this clause.

The metrics are calculated for each measurement resolution interval "measureinterval" as described in clause 15.2.1 of TS 26.113. They are reported to the server according to the reporting interval "reportinginterval" and after the end of the session as described in clause 15.2.1 of TS 26.113.

#### 9.3.2.2 Pose to render to photon metric

The *PoseToRenderToPhoton* duration is the time duration between the time at which the pose information is available from the XR runtime to the renderer and the display time of the rendered frame. The unit of this metric is expressed in milli seconds.

The average pose to render to photon is equal to the sum of *PoseToRenderToPhoton* duration of each frame during the measurement resolution period divided by the time duration, in seconds, of the measurement resolution period. The unit of this metric is expressed in milli seconds and can be a fractional value.

The minimum pose to render to photon duration is equal to the lowest value of *PoseToRenderToPhoton* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The maximum pose to render to photon duration is equal to the highest value of *PoseToRenderToPhoton* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The syntax for the metric "*PoseToRenderToPhoton*" metric is as defined in Table 9.3.2.2-1.

Table 9.3.2.2-1: Pose to render to photon metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| PoseToRenderToPhoton | | Object |  |
|  | @avgPoseToRenderToPhoton | DoubleVectorType | An unordered list of all average pose to render to photon delay measured within each measurement resolution period. |
|  | @minPoseToRenderToPhoton | UnsignedIntVectorType | The minimum pose to render to photon duration is equal to the lowest value of PoseToRenderToPhoton duration measured during each measurement resolution period.  Provides an unordered list of minimum pose to render to photon delay measured during a metric reporting period. |
|  | @maxPoseToRenderToPhoton | UnsignedIntVectorType | The maximum pose to render to photon duration is equal to the highest value of PoseToRenderToPhoton duration measured during each measurement resolution period.  Provides an unordered list of maximum pose to render to photon delay measured during a metric reporting period. |

#### 9.3.2.3 Render to photon metric

The *renderToPhoton* duration is the time duration between the time at which the presentation engine started rendering to the display time of the rendered frame. The unit of this metric is expressed in milli seconds.

The average render to photon metric is equal to the sum of *renderToPhoton* duration of each frame during the measurement resolution period divided by the time duration, in seconds, of the measurement resolution period. The unit of this metric is expressed in milli seconds and can be a fractional value.

The minimum render to photon duration is equal to the lowest value of *renderToPhoton* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The maximum render to photon duration is equal to the highest value of *renderToPhoton* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The syntax for the metric "*renderToPhoton*" metric is as defined in Table 9.3.2.3-1.

Table 9.3.2.3-1: Render to Photon metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| renderToPhoton | | Object |  |
|  | avgRenderToPhoton | DoubleVectorType | An unordered list of all average render to photon delay measured within each measurement resolution period. |
|  | @minRenderToPhoton | UnsignedIntVectorType | The minimum render to photon duration is equal to the lowest value of renderToPhoton duration measured during each measurement resolution period.  Provides an unordered list of minimum render to photon delay measured during a metric reporting period. |
|  | @maxRenderToPhoton | UnsignedIntVectorType | The maximum render to photon duration is equal to the highest value of renderToPhoton duration measured during each measurement resolution period.  Provides an unordered list of maximum render to photon delay measured during a metric reporting period. |

#### 9.3.2.4 Round-trip interaction delay metric

The *roundtripInteractionDelay* duration is the time duration between the time a user action is initiated to the time the action is presented to the user. The unit of this metric is expressed in milli seconds.

The average round trip interaction delay metric is equal to the sum of *roundtripInteractionDelay* duration of each action during the measurement resolution period divided by the number of actions in the measurement resolution period. The unit of this metric is expressed in milli seconds and can be a fractional value. Within each resolution period the number of user actions are summed up and stored in the vector @*numberOfInteractionEvents.*

The minimum round trip interaction delay duration is equal to the lowest value of *roundtripInteractionDelay* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The maximum round trip interaction delay duration is equal to the highest value of *roundtripInteractionDelay* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The identifier of a user action with minimum, and maximum, round trip interaction delays within each measurement resolution period are provided in the @*minActionIDs*, @*maxActionIDs* respectively, as an unordered list of user action identifiers.

The syntax for the metric "*roundtripInteractionDela*y" metric is as defined in Table 9.3.2.4-1

Table 9.3.2.4-1: Round-trip interaction delay metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| roundtripInteractionDelay | | Object |  |
|  | @avgRoundTripInteractionDelay | SoubleVectorType | An unordered list of all average round trip interaction delays measured within each measurement resolution period. |
|  | @numberOfUserActions | UnsignedIntVectorType | The number of user actions within each measurement resolution period are summed up and stored in the vector. Provides an unordered list of user actions (occurred within each measurement period) measured during a metric reporting period. |
|  | @minRoundTripInteractionDelay | UnsignedIntVectorType | The minimum round trip interaction delay duration is equal to the lowest value of roundtripInteractionDelay duration measured during each measurement resolution period.  Provides an unordered list of minimum round trip interaction delay measured during a metric reporting period. |
|  | @minActionIDs | UnsignedIntVectorType | The identifier of a user action with minimum round trip interaction delay within each measurement resolution period. Provides an unordered list of user action identifiers (occurred within each measurement period) with minimum round trip interaction delay measured during a metric reporting period. |
|  | @maxRoundTripInteractionDelay | UnsignedIntVectorType | The maximum round trip interaction delay duration is equal to the highest value of roundtripInteractionDelay duration measured during each measurement resolution period.  Provides an unordered list of maximum round trip interaction delay measured during a metric reporting period. |
|  | @maxActionIDs | UnsignedIntVectorType | The identifier of a user action with maximum round trip interaction delay within each measurement resolution period. Provides an unordered list of user action identifiers (occurred within each measurement period) with maximum round trip interaction delay measured during a metric reporting period. |

#### 9.3.2.5 User interaction delay metric

The *userInteractionDelay* duration is the time duration between the time a user action is initiated to the time the action is taken account by the content creation engine in the scene manager. The unit of this metric is expressed in milli seconds.

The average user interaction delay metric is equal to the sum of *userInteractionDelay* duration of each action during the measurement resolution period divided by the number of actions in the measurement resolution period. The unit of this metric is expressed in milli seconds and can be a fractional value.

The minimum user interaction delay duration is equal to the lowest value of *userInteractionDelay* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The maximum user interaction delay duration is equal to the highest value of *userInteractionDelay* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The identifier of a user action with minimum and maximum user interaction delays within each measurement resolution period are provided in the @*minActionIDs*, and @*maxActionIDs* respectively as an unordered list of user action identifiers.

The syntax for the metric "*userInteractionDelay*" metric is as defined in Table 9.3.2.5-1.

Table 9.3.2.5-1: User interaction delay metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| userInteractionDelay | | Object |  |
|  | @avgUserInteractionDelay | DoubleVectorType | An unordered list of all average user interaction delays measured within each measurement resolution period. |
|  | @numberOfUserActions | UnsignedIntVectorType | The number of user actions within each measurement resolution period are summed up and stored in the vector. Provides an unordered list of user actions (occurred within each measurement period) measured during a metric reporting period. |
|  | @minUserInteractionDelay | UnsignedIntVectorType | The user interaction delay duration is equal to the lowest value of userInteractionDelay duration measured during each measurement resolution period.  Provides an unordered list of user interaction delay measured during a metric reporting period. |
|  | @minActionIDs | UnsignedIntVectorType | The identifier of a user action with minimum round trip interaction delay within each measurement resolution period. Provides an unordered list of user action identifiers (occurred within each measurement period) with minimum round trip interaction delay measured during a metric reporting period. |
|  | @maxUserInteractionDelay | UnsignedIntVectorType | The maximum user interaction delay duration is equal to the highest value of userInteractionDelay duration measured during each measurement resolution period.  Provides an unordered list of maximum user interaction delay measured during a metric reporting period. |
|  | @maxActionIDs | UnsignedIntVectorType | The identifier of a user action with maximum round trip interaction delay within each measurement resolution period. Provides an unordered list of user action identifiers (occurred within each measurement period) with maximum round trip interaction delay measured during a metric reporting period. |

#### 9.3.2.6 Age of contents metric

The *ageOfContent* is the time duration between the time the content is created in the scene by the scene manager and the time it is presented to the user. The unit of this metric is expressed in milli seconds. Within each measurement resolution period the number of scene creations and updates are counted and stored in the vector @*numberOfSceneEvents.*

The minimum age of content duration is equal to the lowest value of *ageOfContent* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The maximum age of content duration is equal to the highest value of *ageOfContent* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The syntax for the metric "*ageOfContent*" metric is as defined in Table 9.3.2.6-1.

Table 9.3.2.6-1: Age of Content metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| ageOfContent | | Object |  |
|  | @ageOfContent | UnsignedIntVectorVectorType | An unordered list of all age of content duration metric measured within each measurement resolution period. |
|  | @numberOfSceneEvents | UnsignedIntVectorType | The number of scene creation and/or scene updates within each measurement resolution period are stored in the vector. Provides an unordered list of scene creation and/or scene updates (occurred within each measurement period) measured during a metric reporting period. |
|  | @minAgeOfContent | UnsignedIntVectorType | The minimum age of content duration is equal to the lowest value of ageOfContent measured during each measurement resolution period.  Provides an unordered list of minimum age of content measured during a metric reporting period. |
|  | @maxAgeOfContent | UnsignedIntVectorType | The maximum age of content is equal to the highest value of ageOfContent measured during each measurement resolution period.  Provides an unordered list of maximum age of content measured during a metric reporting period. |

#### 9.3.2.7 Scene update delay metric

The *sceneUpdateDelay* duration is the time duration spent by the scene manager to update the scene graph. The unit of this metric is expressed in milliseconds. Within each measurement resolution period the number of scene updates are counted and stored in the vector @n*umberOfSceneUpdates.*

The minimum scene update delay is equal to the lowest value of *sceneUpdateDelay* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The maximum age of content duration is equal to the highest value of *sceneUpdateDelay* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The syntax for the metric "*sceneUpdateDelay*" metric is as defined in Table 9.3.2.7-1.

Table 9.3.2.7-1: Scene update delay metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| sceneUpdateDelay | | Object |  |
|  | @sceneUpdateDelay | UnsignedIntVectorVectorType | An unordered list of all scene update delays measured within each measurement resolution period. |
|  | @numberOfSceneupdates | UnsignedIntVectorType | The number of scene updates within each measurement resolution period are stored in the vector. Provides an unordered list of scene updates (occurred within each measurement period) measured during a metric reporting period. |
|  | @minSceneUpdateDelay | UnsignedIntVectorType | The minimum scene update delay is equal to the lowest value of sceneUpdateDelay measured during each measurement resolution period.  Provides an unordered list of minimum scene update delay duration measured during a metric reporting period. |
|  | @maxSceneUpdateDelay | UnsignedIntVectorType | The maximum age of content is equal to the highest value of sceneUpdateDelay measured during each measurement resolution period.  Provides an unordered list of maximum scene update delay duration measured during a metric reporting period. |

#### 9.3.2.8 Metadata delay metric

The *metadataDelay* duration is the time duration between the time the split rendering metadata is sent from the SRC and the time the split rendering server start to render using that metadata. The unit of this metric is expressed in milliseconds.

The average metadata delay metric is equal to the sum of *metadataDelay* duration of each metadata message during the measurement resolution period divided by the number of metadata messages in the measurement resolution period. The unit of this metric is expressed in milli seconds and can be a fractional value. Within each measurement resolution period the number of metadata messages are summed up and stored in the vector *@numberOfMetadataMessages.*

The minimum metadata delay is equal to the lowest value of *metadataDelay* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The maximum metadata delay is equal to the highest value of *metadataDelay* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The syntax for the metric "*metadataDelay*" metric is as defined in Table 9.3.2.8-1.

Table 9.3.2.8-1: Metadata delay metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| metadataDelay | | Object |  |
|  | @avgmetadataDelay | UnsignedIntVectorType | An unordered list of average metadata delays measured within each measurement resolution period. |
|  | @numberOfMetadataMessages | UnsignedIntVectorType | The number of metadata messages within each measurement resolution period are stored in the vector. Provides an unordered list of total number of metadata messages (occurred within each measurement period) measured during a metric reporting period. |
|  | @minMetadaDataDelay | UnsignedIntVectorType | The minimum metadata delay is equal to the lowest value of metadataDelay measured during each measurement resolution period.  Provides an unordered list of minimum metadata delay duration measured during a metric reporting period. |
|  | @maxMetadataDelay | UnsignedIntVectorType | The maximum metadata message delay is equal to the highest value of metadataDelay measured during each measurement resolution period.  Provides an unordered list of maximum metadata delay duration measured during a metric reporting period. |

#### 9.3.2.9 Data frames delay metric

The *dataFrameDelay* duration is the time duration between the time the media rendered frame is transmitted from the split rendering server to the time the split rendering client received the data frame. The unit of this metric is expressed in milliseconds.

The average data frame delay metric is equal to the sum of *dataFrameDelay* duration of each transmitted frame during the measurement resolution period divided by the number of frames transmitted in the measurement resolution period. The unit of this metric is expressed in milli seconds and can be a fractional value. Within each measurement resolution period the number of data frames transmitted are summed up and stored in the vector @n*umberOfDataFrames.*

The minimum data frame delay is equal to the lowest value of *dataFrameDelay* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The maximum data frame delay is equal to the highest value of *dataFrameDelay* duration measured during the measurement resolution period. The unit of this metric is expressed in milli seconds and is an integer value.

The syntax for the metric "*dataFrameDelay*" metric is as defined in Table 9.3.2.9-1.

Table 9.3.2.9-1: Data frame delay metric information for Quality Reporting

|  |  |  |  |
| --- | --- | --- | --- |
| Key | | Type | Description |
| dataFrameDelay | | Object |  |
|  | @avgdataFrameDelay | UnsignedIntVectorType | An unordered list of average data frame transmission delays measured within each measurement resolution period. |
|  | @numberOfDataFrames | UnsignedIntVectorType | The number of data frames transmitted within each measurement resolution period are stored in the vector. Provides an unordered list of total number of data frames transmitted (within each measurement period) during a metric reporting period. |
|  | @minDataFrameDelay | UnsignedIntVectorType | The minimum data frame delay is equal to the lowest value of dataFrameDelay measured during each measurement resolution period.  Provides an unordered list of minimum data frame delay duration measured during a metric reporting period. |
|  | @maxdataFrameDelay | UnsignedIntVectorType | The maximum data frame delay is equal to the highest value of dataFrameDelay measured during each measurement resolution period.  Provides an unordered list of maximum data frame delay duration measured during a metric reporting period. |

### 9.3.3 Quality metrics reporting

#### 9.3.3.1 General

The quality metrics report follows the XML-based report format defined in clause 9.3.5.2.

The MIME type of an XML-formatted QoE report shall be "application/3gprtc-qoe-report+xml".

The metrics reporting protocol is as defined in clause 9.5.3 of TS 26.510. Split rendering UEs shall use the above MIME content type. The metrics report format is defined in the following sub clause.

#### 9.3.3.2 Report format

The QoE report is formatted as an XML document that complies with the XML schema in listing 10.6.2‑1 of TS 26.247 [17].

The schema in listing 9.3.5.2-1 is an extension to allow additional QoE metrics for SR UE to be reported using the QoE report specified in clause 10.6.2 of TS 26.247 [17].

Listing 9.3.3.2-1: SR\_MSE QoE Metrics XML schema

|  |
| --- |
| <?xml version="1.0"?> <xs:schema version="TSG104-Rel18" xmlns:xs="http://www.w3.org/2001/XMLSchema"  targetNamespace="urn:3gpp:metadata:2024:RTC:SR\_MSEQoEMetrics"  xmlns:sv="urn:3gpp:metadata:2016:PSS:schemaVersion"  xmlns="urn:3gpp:metadata:2024:RTC:SR\_MSEQoEMetrics" elementFormDefault="qualified">    <xs:element name="QoeMetric" type="QoeMetricType"/>    <xs:complexType name="QoeMetricType">  <xs:sequence>  <xs:choice>  <xs:element name="poseToRenderToPhoton" type="PoseToRenderToPhotonType"/>  <xs:element name="renderToPhoton" type="RenderToPhotonType"/>  <xs:element name="roundTripInteractionDelay" type="RoundTripInteractionDelayType"/>  <xs:element name="userInteractionDelay" type="UserInteractionDelayType"/>  <xs:element name="ageOfContent" type="AgeOfContentType"/>  <xs:element name="sceneUpdateDelay" type="SceneUpdateDelayType"/>  <xs:element name="metadataDelay" type="MetadataDelayType"/>  <xs:element name="dataFrameDelay" type="DataFrameDelayType"/>  </xs:choice>  <xs:element ref="sv:delimiter"/>  <xs:any namespace="##other" processContents="skip" minOccurs="0" maxOccurs="unbounded"/>  </xs:sequence>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="PoseToRenderToPhotonType">  <xs:attribute name="avgPoseToRenderToPhoton" type="DoubleVectorType" use="required"/>  <xs:attribute name="minPoseToRenderToPhoton" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="maxPoseToRenderToPhoton" type="UnsignedIntVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="RenderToPhotonType">  <xs:attribute name="avgPoseToRenderToPhoton" type="DoubleVectorType" use="required"/>  <xs:attribute name="minRenderToPhoton" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="maxRenderToPhoton" type="UnsignedIntVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="RoundTripInteractionDelayType">  <xs:attribute name="avgRoundTripInteractionDelay" type="DoubleVectorType" use="required"/>  <xs:attribute name="numberOfUserActions" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="minRoundTripInteractionDelay" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="minActionIds" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="maxRoundTripInteractionDelay" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="maxActionIds" type="UnsignedIntVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="UserInteractionDelayType">  <xs:attribute name="avgUserInetractionDelay" type="DoubleVectorType" use="required"/>  <xs:attribute name="numberOfUserActions" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="minUserInetractionDelay" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="minActionIds" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="maxUserInteractionDelay" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="maxActionIds" type="UnsignedIntVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="AgeOfContentType">  <xs:attribute name="ageOfContent" type="UnsignedIntVectorVectorType" use="required"/>  <xs:attribute name="mumberOfSceneEvents" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="minageOfContent" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="maxAgeOfContent" type="UnsignedIntVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="SceneUpdateDelayType">  <xs:attribute name="sceneUpdateDelay" type="UnsignedIntVectorVectorType" use="required"/>  <xs:attribute name="numberOfSceneUpdates" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="minsceneUpdateDelay" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="maxsceneUpdateDelay" type="UnsignedIntVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="MetadataDelayType">  <xs:attribute name="avgMetadataDelay" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="numberOfMetadataMessages" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="minMetadataDelay" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="maxMetadataDelay" type="UnsignedIntVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:complexType name="DataFrameDelayType">  <xs:attribute name="avgDataFrameDelay" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="numberOfDataFrames" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="minDataFrameDelay" type="UnsignedIntVectorType" use="required"/>  <xs:attribute name="maxDataFrameDelay" type="UnsignedIntVectorType" use="required"/>  <xs:anyAttribute processContents="skip"/>  </xs:complexType>  <xs:simpleType name="UnsignedIntVectorVectorType">  <xs:list itemType="UnsignedIntVectorType"/>  </xs:simpleType>  <xs:simpleType name="unsignedLongVectorType">  <xs:list itemType="xs:unsignedLong"/>  </xs:simpleType>  <xs:simpleType name="DoubleVectorType">  <xs:list itemType="xs:double"/>  </xs:simpleType>  <xs:simpleType name="StringVectorType">  <xs:list itemType="xs:string"/>  </xs:simpleType>  <xs:simpleType name="UnsignedIntVectorType">  <xs:list itemType="xs:unsignedInt"/>  </xs:simpleType>  </xs:schema> |

#### Quality Reporting Scheme and Metrics reporting configuration for SRC

A SR UE shall use the metrics reporting scheme defined in clause 6.7 of TS 26.113 [?]. The URN to be used for the **Scheme**@schemeIdUri shall be "urn:3GPP:ns:PSS:RTC:QM1". The Metrics Reporting Provisioning API allows an RTC Application Provider to configure the Metrics Collection and Reporting procedure for a particular split rendering session at reference point RTC-1. The Service Access Information API allows an RTC Application Function to configure the metrics collection and reporting process for a particular split rendering session at reference point RTC-5.

A SR UE shall use the data model for metrics reporting provisioning API defined in clause 8.10.3.1 of TS 26.510 [?] and the Service Access Information API defined in clause 9.2.3 of TS 26.510 [?]. The metrics element present in the MetricsReportingConfiguration resource and the ServiceAccessInformation resource shall include zero or more metrics defined in clause 9.3.4 of this document in addition to the quality metrics defined in clause 15.2 of TS 26.113.

|  |
| --- |
| **End of second Change** |