**3GPP TSG-SA4 Meeting Sophia-Antipolis,**

**France, 29th Jan 2024 - 2nd Feb 2024 *S4-240123***

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *CR-Form-v12.2* | | | | | | | | |
| **Pseudo CHANGE REQUEST** | | | | | | | | |
|  | | | | | | | | |
|  | **26.119** | **CR** | - | **rev** | **-** | **Current version:** | **1.0.0** |  |
|  | | | | | | | | |
| *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:*** | [MeCAR] Improvements to TS 26.119 addressing general and editorial comments | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Source to WG:*** | Huawei, Hisilicon, Xiaomi, Tencent, Interdigital | | | | | | | | | |
| ***Source to TSG:*** | S4 | | | | | | | | | |
|  |  | | | | | | | | | |
| ***Work item code:*** | MeCAR | | | | |  | ***Date:*** | | | 2024-01-20 |
|  |  | | | |  | |  | | |  |
| ***Category:*** | D |  | | | | | ***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:*** | | TS 26.119 needs general and editorial improvements as agreed in [**S4aV230116**](https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/3GPP_SA4_AHOC_MTGs/SA4_VIDEO/Docs/S4aV230116.zip) | | | | | | | | |
|  | |  | | | | | | | | |
| ***Summary of change:*** | | * New Table of contents with more logical chapters * References fixed * Terms and definitions extended * Editorial improvements in clause 4, 5, 6, 7, 8, 9 and 10 * scene description in separate section * Metadata in separate section | | | | | | | | |
|  | |  | | | | | | | | |
| ***Consequences if not approved:*** | | Technical Specification is harder to read and understand, incorrect references and definitions | | | | | | | | |
|  | |  | | | | | | | | |
| ***Clauses affected:*** | | All except annex and scope | | | | | | | | |
|  | |  | | | | | | | | |
|  | | **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** |

Title of spec: UE Media Capabilities for AR services/applications/experiences (TBD)Contents

[Foreword 5](#_Toc156856797)

[Introduction 6](#_Toc156856798)

[1 Scope 7](#_Toc156856799)

[2 References 7](#_Toc156856800)

[3 Definitions of terms, symbols and abbreviations 8](#_Toc156856801)

[3.1 Terms 8](#_Toc156856802)

[3.2 Abbreviations 8](#_Toc156856803)

[4 XR Concepts and Device Types 10](#_Toc156856804)

[4.1 XR concepts 10](#_Toc156856805)

[4.1.1 XR Device 10](#_Toc156856806)

[4.1.2 XR Application 10](#_Toc156856807)

[4.1.3 XR Runtime 10](#_Toc156856808)

[4.2 Media pipelines and rendering loop 13](#_Toc156856809)

[4.3](#_Toc156856810) ~~[AR](#_Toc156856810)~~ [Device Types 13](#_Toc156856810)

[4.3.1 Device type 1: Thin AR glasses 13](#_Toc156856811)

[4.3.2 Device type 2: AR glasses 14](#_Toc156856812)

[4.3.3 Device type 3: XR phone 14](#_Toc156856813)

[4.3.4 Device type 4: XR Head Mounted Display (HMD) 14](#_Toc156856814)

[5 Device reference architecture and interfaces 14](#_Toc156856815)

[5.1 Architecture 14](#_Toc156856816)

[5.2 Description of the functional blocks 15](#_Toc156856817)

[5.3 Interfaces and APIs 15](#_Toc156856818)

[6 General system functions and capabilities 16](#_Toc156856819)

[6.1 XR system capabilities 16](#_Toc156856820)

[7 Visual functions and capabilities 20](#_Toc156856821)

[7.1 Decoding capabilities 21](#_Toc156856822)

[7.1.1 Single decoder instance 21](#_Toc156856823)

[7.1.2 Concurrent decoding capabilities 21](#_Toc156856824)

[7.2 Encoding capabilities 23](#_Toc156856825)

[7.2.1 Single encoder instance 23](#_Toc156856826)

[8 Audio functions and capabilities 23](#_Toc156856827)

[8.1 Audio/Speech Decoding 23](#_Toc156856828)

[8.2 Audio/Speech Encoding 24](#_Toc156856829)

[9 Scene processing capabilities 24](#_Toc156856830)

[9.1 General 24](#_Toc156856831)

[9.2 glTF-based Scene Description capabilities 24](#_Toc156856832)

[10 Device types and media profiles 26](#_Toc156856833)

[10.1 Introduction 26](#_Toc156856834)

[10.2 Device type 1: Thin AR glasses 26](#_Toc156856835)

[10.2.1 General 26](#_Toc156856836)

[10.2.2 XR System support 26](#_Toc156856837)

[10.2.3 Video capabilities support 27](#_Toc156856838)

[10.2.4 Audio/Speech capabilities support 27](#_Toc156856839)

[10.2.5 Scene Description capabilities support 27](#_Toc156856840)

[10.3 Device type 2: AR glasses 27](#_Toc156856841)

[10.3.1 General 27](#_Toc156856842)

[10.3.2 XR System support 28](#_Toc156856843)

[10.3.3 Video capabilities support 28](#_Toc156856844)

[10.3.4 Audio/Speech capabilities support 28](#_Toc156856845)

[10.3.5 Scene Description capabilities support 29](#_Toc156856846)

[10.4 Device type 3: XR phone 29](#_Toc156856847)

[10.4.1 General 29](#_Toc156856848)

[10.4.2 XR System support 29](#_Toc156856849)

[10.4.3 Video capabilities support 29](#_Toc156856850)

[10.4.4 Audio/Speech capabilities support 30](#_Toc156856851)

[10.4.5 Scene Description capabilities support 30](#_Toc156856852)

[10.5 Device type 4: XR HMD 31](#_Toc156856853)

[10.5.1 General 31](#_Toc156856854)

[10.5.2 XR System support 31](#_Toc156856855)

[10.5.3 Video capabilities support 31](#_Toc156856856)

[10.5.4 Audio/Speech capabilities support 32](#_Toc156856857)

[10.5.5 Scene Description capabilities support 32](#_Toc156856858)

[11 QoE Metrics 32](#_Toc156856859)

[11.1 Metrics and Observation Points 33](#_Toc156856860)

[11.1.1 Overview 33](#_Toc156856861)

[11.1.2 Observation Point 1: XR Runtime information 33](#_Toc156856862)

[11.1.3 Observation Point 2 34](#_Toc156856863)

[11.1.4 Observation Point 3 34](#_Toc156856864)

[11.1.5 Observation Point 4 34](#_Toc156856865)

[11.2 Metrics Definitions 34](#_Toc156856866)

[11.2.1 Latency metrics 34](#_Toc156856867)

[12 Metadata formats 36](#_Toc156856868)

[12.1 General 36](#_Toc156856869)

[12.2 Pose information format 36](#_Toc156856870)

[10.3 Action format 37](#_Toc156856871)

[10.4 Available Visualization Space format 38](#_Toc156856872)

[A.1 Introduction 40](#_Toc156856619)

[B.1 Introduction 40](#_Toc156856620)

[B.2 Capability mapping to OpenXR 40](#_Toc156856621)

[B.2.1 Mapping overview 40](#_Toc156856622)

[B.2.2 XR views and rendering loop 43](#_Toc156856623)

[B.2.3 Available Visualization Space implementation 45](#_Toc156856624)

[B.2.3.1 Using OpenXR\_XR\_FB 45](#_Toc156856625)

[B.2.3.2 Using xrComputeNewSceneMSFT 45](#_Toc156856626)

[[B.3 Capability mapping to WebXR] 45](#_Toc156856627)

# Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# Introduction

The present document provides technologies for the deployment of Augmented Reality (AR) and eXtended Reality services (XR) as defined in TR 26.928 [2] and the execution of Augmented Reality applications on targeted devices such as those identified in TR 26.998 [3].

On the spectrum of eXtended Reality (XR) experiences, Augmented Reality overlay virtual information on top of the user’s perception of the real environment. Those virtual and real components of the scene seamlessly blend together from the user’s perspective. Additionally, some AR experiences can enable interactivity between the user and the virtual components of the scene.

In the present document, the focus lies in the definition of the media capabilities for AR devices, including media format encapsulation capabilities, media codec capabilities, processing function capabilities. The related minimum required performances for different device types are also defined.

# 1 Scope

The present document defines the supported media formats, codecs, processing functions for XR Devices in UE per XR device type category. The present document addresses the interoperability gaps identified in the conclusions of TR 26.998 [3].

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TR 26.928: "Extended Reality (XR) in 5G".

[3] 3GPP TR 26.998: "Support of 5G glass-type Augmented Reality / Mixed Reality (AR/MR) devices".

[4] 3GPP TR 26.857: "5G Media Service Enablers".

[5] Khronos, "The OpenXR Specification", https://registry.khronos.org/OpenXR/specs/1.0/html/xrspec.html.

[6] 3GPP TS 26.506: "5G Real-time Media Communication Architecture (Stage 2)".

[7] ITU-T Recommendation H.264 (08/2021): "Advanced video coding for generic audiovisual services".

[8] ITU-T Recommendation H.265 (08/2021): "High efficiency video coding".

[9] 3GPP TS 26.117: "5G Media Streaming (5GMS); Speech and audio profiles".

[10] ISO/IEC 12113:2022 Information technology Runtime 3D asset delivery format Khronos glTF™2.0

[11] ISO/IEC 23090-14:2023 Information technology Coded representation of immersive media Part 14: Scene description

[12] ISO/IEC 23090-14:2023/Amd 1:2023 Information technology Coded representation of immersive media Part 14: Scene description

[13] ISO/IEC 23090-14:2023/DAmd 2 Information technology Coded representation of immersive media Part 14: Scene description

# 3 Definitions of terms, symbols and abbreviations

## 3.1 Terms

For the purposes of the present document, the terms given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1]

AR Glasses:

**Composition:** TBD

**Media Session Handler**: a set of functions responsible for handling all 5G control plane operations, such as requesting network assistance, discovering and allocating edge resources, etc.

**Presentation Engine**: a set of composite renderers, rendering the component of the scenes,

**Reference Points**: geometric points whose position is identified both mathematically and physically.

SD-Rendering: Scene Description Rendering

**Swapchain image:** a queue of images that are waiting to be presented to the screen

**XR Application**: a software application that integrates audio-visual content into the user’s real-world environment **XR Device**: a device capable of offering an XR experience.

**XR Head** **Mounted Display**:

**XR Runtime**: Set of functions provided by the XR Device to the XR Application in order to create XR experiences.

**XR Runtime API**: the API to communicate with an XR Runtime

**XR Scene Manager:** a set of functions that supports the application in arranging the logical and spatial representations

**XR Session**: an application’s intention to present XR content to the user.

**XR Source Management**: management of data sources provided through the XR runtime.

**XR Phone:** TBD

**XR System**: a collection of resources and capabilities from the XR Runtime exposed to the XR Application for the duration of the XR Session.

**XR View**: a rendered view of the scene generated by the XR Application and passed on to the XR Runtime during a running XR Session

**XR Space**: TBD

**Warping:** TBD

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

AR Augmented Reality

AAC Advanced Audio Coding

AVC Advanced Video Coding

CPB Coded Picture Buffer

DPB Decoded Picture Buffer

ELD Enhanced Low Delay

EVS Enhanced Voice Services

glTF graphics library Transmission Format

GLB glTF Binary

HEVC High Efficiency Video Coding

HMD Head-Mounted Display

HRD Hypothetical Reference Decoder

HSS Hypothetical Stream Scheduler

IVAS Immersive Voice and Audio Services

JSON JavaScript Object Notation

MPEG Moving Picture Experts GroupMR Mixed Reality

OP Observation Point

MPEG SD MPEG Scene Description

SLAM Simultaneous localisation and mapping

UE User Equipment

VCL Video Coding Layer

VR Virtual Reality

XR eXtended Reality

XR API XR Runtime Application Programming Interface

# 4 XR Concepts and Device Types

## 4.1 XR concepts

### 4.1.1 XR Device

Extended Reality (XR) refers to a continuum of experiences combine real-a and- virtual combined environments in which the user is immersed through one or more devices capable of audio, visual and haptics rendering generated by computers through human-machine interaction. XR encompasses technologies associated with Virtual Reality (VR), Augmented Reality (AR) and Mixed Reality (MR) which constitute the so-called XR continuum. A detailed overview of definitions, concepts and background on XR and AR is provided in TR 26.928 [2] and TR 26.998 [3], respectively.

Clause 4.1 of 26.928 [2] defines the use the terms Augmented Reality, Virtual Reality, Mixed Reality and eXtended Reality as used throughout this document.

An XR device is a device capable of offering an XR experience. An XR Device is assumed to have one or several displays, speakers, sensors, cameras, microphones, actuators, controllers and/or other peripherals that allow to create XR experiences, i.e. experiences for which the user interacts with the content presented in virtual world and/or augmented to the real-world. Example of XR Devices are AR Glasses, a VR/MR Head-Mounted Display (HMD) or a regular smartphone, etc.

### 4.1.2 XR Application

An application which offers an XR experience by making use of the hardware capabilities, including media capabilities, of the XR Device it runs on as well as the network connectivity to retrieve the asset being used by the application is referred to as an XR Application. In the context of this specification, it is primarily assumed that access to the network is provided by 5G System functionalities.

To enable XR experiences, the hardware on an XR Device typically offers a set of functions to perform commonly required XR operations. These operations include, but are not limited to:

- accessing controller/peripheral state,

- getting current and/or predicted tracking positions and pose information of the user,

- receiving or generating pre-rendered views of the scene for final presentation to the user, taking into account the latest user position and pose. Adaptation to the latest user position and pose is also referred to as warping.

### 4.1.3 XR Runtime

#### 4.1.3.1 General

XR Runtime provides a set of functionalities to XR applications including but not limited to peripheral management, runtime functions as tracking, SLAM, composition and warping etc. The functions are accessible to the XR Application via an API exposed by the XR Runtime referred to as the XR Runtime Application Programming Interface (XR API). The XR Runtime typically handles functionalities such as composition, peripheral management, tracking, Spatial Localization and Mapping (SLAM), capturing and audio-related functions. Further, it is assumed that the hardware and software capabilities of the XR Device are accessible through well-defined device APIs, and in particular the media capabilities are accessible through media APIs.

An overview of an XR Device logical components is shown in Figure 4.1.2-1.



Figure 4.1.2-1 Logical components of an XR Device

This specification relies on a hypothetical XR Runtime and its API in order to define the media capabilities. This way, different implementation of XR runtimes may be compatible with this specification. However, for the purpose of developing this specification, the minimal set of expected functionalities of the XR Runtime has been aligned with the core Khronos’ OpenXR specification [5]. Support for other XR Runtime environments is not precluded by this approach. Lastly, a mapping of general functionalities to OpenXR is provided in Annex B.

#### 4.1.3.2 XR session and rendering loop using XR Runtime (informative)

At startup, the XR Application creates an XR Session via the XR Runtime API and allocates the necessary resources from the available resources on the XR Device. Upon success, the XR Runtime begins the life cycle of the XR Session whose cycle is typically made of several states. The purpose of those states is to synchronise the rendering operations controlled by the XR Application with the display operations controlled by the XR Runtime. The rendering loop is thus a task jointly executed by the XR Runtime and the XR Application and synchronised via the states of the XR Session.

The XR Application is responsible of generating a rendered view of the scene from the perspective of the user. To this end, the XR Application produces XR Views which are passed to the XR Runtime at iterations of the rendering loop. The XR Views are generated for one or more poses in the scene for which the XR application can render images. From those views, the view corresponding to the viewer’s pose is typically called the primary view. There may be other XR Views defined in the scene, for instance for spectators.

The XR Views are configured based on the display properties of the XR Device. A typical head-mounted XR System has a stereoscopic view configuration, i.e. two views, while a handheld XR Device has a monoscopic view configuration, i.e. a single view. Other view configurations may exist. At the start of session, the XR Application configures the view type based on those device properties which remains the same for the duration of the XR Session.

A XR View may also comprise one more composition layers associated with an image buffer. Those layers are then composed together by the XR Runtime to form the final rendered images.

In addition to layers containing visual data, an XR View may be complemented with a layer provided depth information of the scene associated with this XR View. This additional information may help the XR Runtime to perform pose correction when generating the final display buffer. Another type of layer can be an alpha channel layer useful for blending the XR View with the real environment for video-see through XR devices, e.g. which is the case for AR applications running on smartphones.

For the XR Application to render the XR Views, the XR Runtime provides the viewer pose as well as projection parameters which are typically taken into account by applications to render those different XR Views. The viewer pose and projection parameters are provided for a given display time in the near future. The XR Runtime accepts repeated calls for prediction updates of the pose, which may not necessarily return the same result for the same target display time. Instead, the prediction gets increasingly accurate as the function is called closer to the given time for which a prediction is made. This allows an application to prepare the predicted views early enough to account for the amount of latency in the rendering while at the same time minimizing the prediction error when pre-rendering the views.

In addition, the XR Application communicates with input devices in order to collect actions. Actions are created at initialization time and later used to request input device state, create action spaces, or control haptic events. Input action handles represent ‘actions’ that the application is interested in obtaining the state of, not direct input device hardware.



Figure 4.1.4-1 Rendering loop for visual data

## 4.2 Media pipelines and rendering loop

[Editor’s note: Description of the pipelines, sensors, AR runtime, decoders… identify for what entities capabilities are defined]

In the context of this specification, media to be rendered and displayed by the XR Device through the XR Runtime is typically available in a compressed form on the device. In contrast, media is accessed using a 5G System, decoded in the device using media capabilities, and the decoded media is rendered to then be provided through swapchains to the XR Runtime as shown in Figure 4.2.1-1.

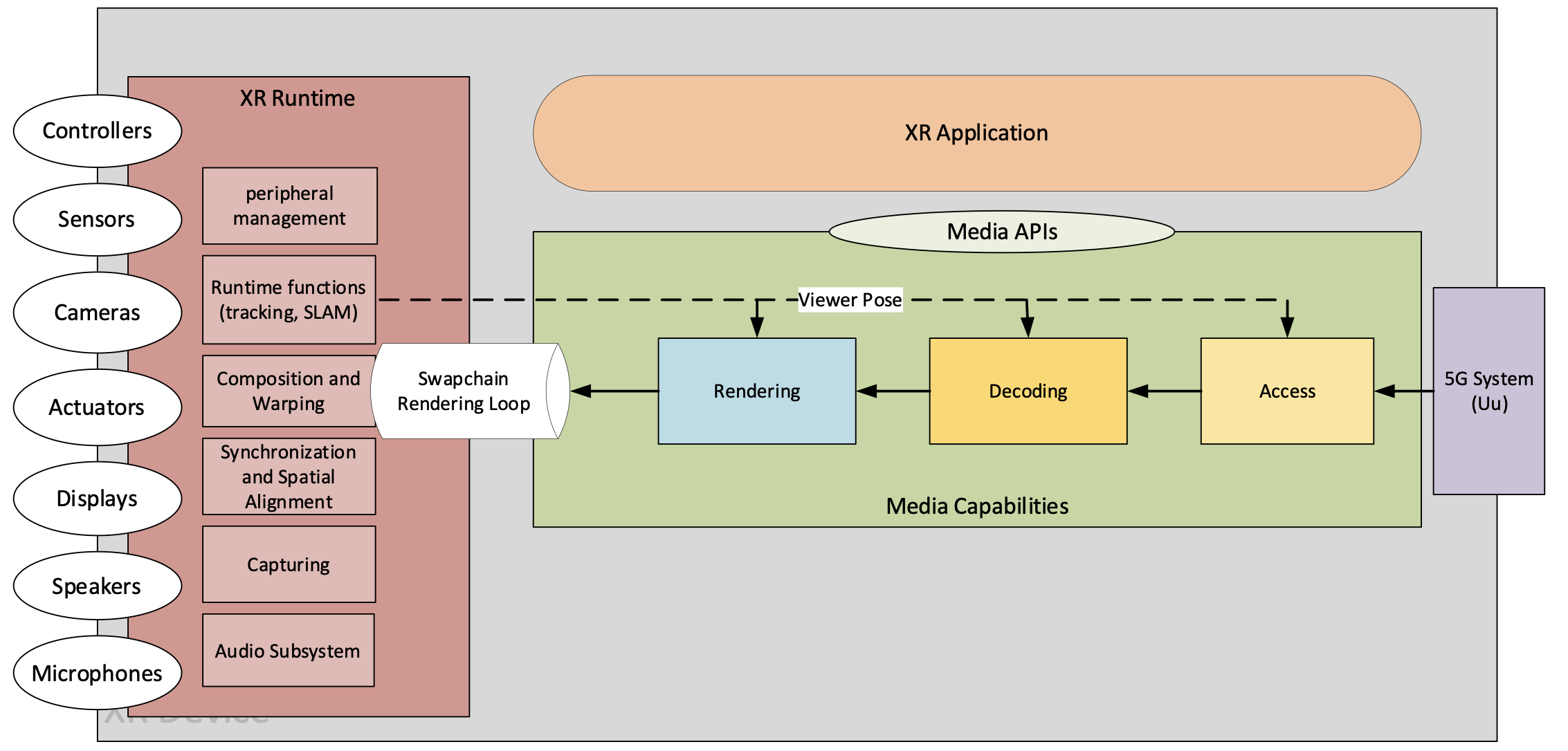


Figure 4.2.1-1 Media pipelines: Access, decoding and rendering

The rendering function is responsible to adapt the content to be presentable by the XR Runtime by making use of a rendering loop and using swapchains. The application configures pipeline of different processes, namely the media access, the decoding and the rendering. The static information provided to the rendering step needs to be sufficient to configure the number of layers as well as each layer appropriately including:

- View configuration

- Blend modes

- XR spaces

- swap chain formats and images

- projection layer types

[Frame rates: https://registry.khronos.org/OpenXR/specs/1.0/html/xrspec.html#XR\_FB\_display\_refresh\_rate

Rendering supported by the XR runtime

- Visual

- Audio]

## 4.3 AR Device Types

### 4.3.1 Device type 1: Thin AR glasses

The thin AR glasses device type represents a type of device which is considered to be power-constrained and with limited computing power with respect to the other device types. These limitations typically come from the requirement to design a device with a small and lightweight form factor. Regarding rendering capacity, this device type is expected to rely on remote rendering to be able display complex scenes to the user. For example, such device type may run a split rendering session where the split rendering server delivers pre-rendered views of the scene. However, devices in this category can still operate without external support for applications that do not require complex rendering capabilities, for instance, text messaging, 2D video communication, etc. Lastly, the thin AR glasses offers AR experiences to the user via optical see-through display.

### 4.3.2 Device type 2: AR glasses

The AR glasses device type represents a type of device which is considered to have higher computation power compared to the thin AR glasses device type. As a result, this AR device type has higher rendering capacities and is generally expected to be capable of rendering scenes without external support, even though remote rendering is not precluded to lower the power consumption on the device or enable the display of scenes beyond the device’s rendering capability. Lastly, the AR glasses offers AR experiences to the user via optical see-through display.

### 4.3.3 Device type 3: XR phone

The XR phone device type represents a type of device which corresponds to a smartphone with capacities and resources sufficient to offer AR experiences. As a result, this device type is capable of rendering scenes without external support. Lastly, the XR phone offers AR experiences to the user via video see-through display.

### 4.3.4 Device type 4: XR Head Mounted Display (HMD)

The XR HMD device type represents a type of device which corresponds to HMDs capable of offering at least AR experiences but not precluding other types of XR experiences. This device type is expected to be capable of rendering scenes without external support. Lastly, the XR phone offers AR experiences to the user via video see-through display.

# 5 Device reference architecture and interfaces

## 5.1 Architecture

The XR Baseline Client represents the functionalities, the peripherals, and the interfaces that are present on a generic XR Device. The XR Baseline Client reference architecture is shown in Figure 5.1-1 which reflects the architecture of XR Devices.

Une image contenant diagramme

Description générée automatiquement

Figure 5.1-1 XR Baseline clientarchitecture and interfaces

## 5.2 Description of the functional blocks

In terms of functionalities, an XR Baseline Client is composed of:

- An **XR Application**: a software application that integrates audio-visual content into the user’s real-world environment

- An **XR Runtime**: a set of functions that interface with a platform to perform commonly required operations, such as accessing the controller/peripheral state, getting current and/or predicted tracking positions, performing spatial computing, as well as submitting rendered frames to the display processing unit and rendered audio to the speakers with a late stage re-projection to the latest pose (see clause 4.1.3).

- An **XR Source Management**: management of data sources provided through the XR runtime such as microphones, cameras, trackers, etc. For instance, making the information available to the XR application or providing it to the MAF for sending in the uplink.

- A **Media Access Function**: A set of functions that enables access to media and other XR-related data that is needed in the Scene manager or XR Runtime to provide an XR experience as well to create delivery formats for information provided by the XR Source Management.

- A **Scene Manager**: a set of functions that supports the application in arranging the logical and spatial representation of a multisensorial scene based on support from the XR Runtime.

- A **Presentation Engine**: a set of composite renderers, rendering the component of the scenes, based on the input from the Scene Manager.

- A **Media Session Handler**: a set of functions responsible for handling all 5G control plane operations, such as requesting network assistance, discovering and allocating edge resources, etc. This may be realized as a 5G-RTC MSH, 5GMS Media Session Handler, or any other function. In addition, those functional blocks are integrated together via interfaces. Interfaces may be made of APIs and/or data formats and collectively act as a contract between the two sides of the interface.

In addition, those functional blocks are integrated together via interfaces. Interfaces may be made of APIs and/or data formats and collectively act as a contract between the two sides of the interface.

## 5.3 Interfaces and APIs

The XR Baseline Client contains the following interfaces:

- **IF-1** for the XR Runtime on one side and the Application (1a), the XR Source Management (1b) and the Presentation Engine (1c). IF-1a-c is implemented as an API (API-1) that exposes functions provided by the XR Runtime. An example of this API is the Khronos OpenXR API.

- **IF-2** describes the functions exposed by the XR Source Management that can be accessed and controlled by the XR application, or possibly other functions in the device. IF-2 is typically implemented as an API.

- **IF-3** lies between the XR Source Management and the Media Access Function and provides serialized information accessible on XR Runtime to the MAF.

- **IF-4** between the Media Access Function and the 5G System for user plane data, such as application data or other graphics data needed by the XR application.

- **IF-5** lies between the UE and the 5G System, implementing control sessions. An example instance of this interface is RTC-5 interface as defined by TS 26.506 [6].

- **IF-6** connects the Media Session Handler and the Application/MAF. It offers the tools for them to activate 5G media functionality such as network assistance and edge resource discovery. The IF-6 is realized through an API (API-6).

- **IF-7** lies between the XR Application and the Media Access function to configure Media Access. This is typically implemented as an API (API-7) that exposes functions of the MAF.

- **IF-8** is an interface that allows the XR application to make use of 5G System connectivity.

- **IF-9** between the Scene Manager and the Media Access Function.

- **IF-10** between the Scene Manager and the XR Application.

# 6 General system functions and capabilities

[Ed note: Description of general functions such as sensors, runtime and their different capabilities, same for system aspects including protocols…]

## 6.1 XR system capabilities

6device as defined in clause 4.3

6

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |
|  |  |  |  |  |

## 

### 

### 

### 

### 

# 7 Visual functions and capabilities

[Ed note: eg description of video formats and codecs, same for GPU capabilities and formats]

## 7.1 Decoding capabilities

### 7.1.1 Single decoder instance

The following video decoding capabilities are defined:

**- AVC-FullHD-Dec**: the capability to decode H.264 (AVC) Progressive High Profile Level 4.0 [7] bitstreams.

**- AVC-UHD-Dec:** the capability to decode H.264 (AVC) Progressive High Profile Level 5.1 [7] bitstreams with the following additional requirements:

- the maximum VCL Bit Rate is constrained to be 120 Mbps with cpbBrVclFactor and cpbBrNalFactor being fixed to be 1250 and 1500, respectively; and,

- the bitstream does not contain more than 10 slices per picture.

**- AVC-8K-Dec:** the capability to decode H.264 (AVC) Progressive High Profile Level 6.1 [7] bitstreams [TBD: possible additional requirements]

[Editor’s note: not present in TS 26.511].

**- HEVC-FullHD-Dec**: the capability to decode H.265 (HEVC) Main10 Profile, Main Tier, Level 4.1 [8] bitstreams that have general\_progressive\_source\_flag equal to 1, general interlaced\_source\_flag equal to 0, general\_non\_packed\_constraint\_flag equal to 1 and general\_frame\_only\_constraint\_flag equal to 1.

**- HEVC-UHD-Dec**: the capability to decode H.265 (HEVC) Main10 Profile, Main Tier, Level 5.1 [8] bitstreams that have general\_progressive\_source\_flag equal to 1, general interlaced\_source\_flag equal to 0, general\_non\_packed\_constraint\_flag equal to 1 and general\_frame\_only\_constraint\_flag equal to 1.

**- HEVC-8K-Dec**: the capability to decode H.265 (HEVC) Main10 Profile, Main Tier, Level 6.1 [8] bitstreams that have general\_progressive\_source\_flag equal to 1, general interlaced\_source\_flag equal to 0, general\_non\_packed\_constraint\_flag equal to 1, and general\_frame\_only\_constraint\_flag equal to 1 with the following additional requirements:

- the bitstream does not exceed the maximum luma picture size in samples of 33,554,432; and,

### - the maximum VCL Bit Rate is constrained to be 80 Mbps with CpbVclFactor and CpbNalFactor being fixed to be 1000 and 1100, respectively.7.1.2 Concurrent decoding capabilities

[Editor’s note: The technical details of the concurrent decoder instances definition in this clause will be further checked.]

#### 7.1.2.1 Definition

Concurrent video decoder instances are defined as follows.

For *N* video bitstreams encoded according to a video codec profile, decoding units flow into the coded picture buffer (CPB) for each stream according to a specified arrival schedule and are delivered by the common Hypothetical Stream Scheduler (HSS) that schedules the *N* bitstreams for decoding each of the units. For each access unit

**-** all data associated with an access unit is removed and decoded instantaneously by the instantaneous decoding process at CPB removal time of the access unit.

**-** Each decoded picture is placed in the Decoded Picture Buffer (DPB) for being referenced by the decoding process of this stream as well as for output and cropping.

**-** A decoded picture is removed from the DPB at the time that it becomes no longer needed for inter-prediction reference as well as the output time of the access unit is the largest of all decoded pictures remaining in the group of N decoders

Then at any point time,

**-** each of the individual streams conforms to the signaled profile/level/tier and HRD parameters of the individual stream.

**-** The sum of the CPB size conforms to common profile/level/tier signaling

**-** The aggregate decoder processing speed (samples per seconds) conforms to common profile/level/tier signaling.

**-** The sum of the DPB size conforms to common profile/level/tier signaling

**-** The common DPB size conforms to common profile/level/tier signaling

A set of *N* concurrent decoder instances conforms to a given capabilities (defined in clause 7.1.2.2), if a set of up to *N* bitstreams encoded to be decodable by the HRD above, is decodable within the timing limits.

#### 7.1.2.2 Capabilities

Based on the definition in clause 7.1.2.1, the following capabilities are defined:

**- AVC-FullHD-Dec-2**: The capability of supporting up to two (*N*=2) concurrent decoder instances with the aggregate capabilities of *AVC-FullHD-Dec*.

**- AVC-UHD-Dec-4**: The capability of supporting up to four (*N*=4) concurrent decoder instances with the aggregate capabilities of *AVC-UHD-Dec*.

**- HEVC-UHD-Dec-4:** The capability of supporting up to four (*N*=4) concurrent decoder instances with the aggregate capabilities of *HEVC-UHD-Dec*.

**- UHD-Dec-4**: The capability supporting up to four (*N*=4) concurrent decoder instances with either:

- the aggregate capabilities of *AVC-UHD-Dec-4*;

- the aggregate capabilities of *HEVC-UHD-Dec-4*; or,

- the capability of decoding up to 4 bitstreams for which each bitstream does not exceed the capability of being decodable either with *AVC-FullHD-Dec* or *HEVC-FullHD-Dec*.

**- AVC-8K-Dec-8:** The capability of supporting up to eight (*N*=8)concurrent decoder instances with the aggregate capabilities of *AVC-8K-Dec*.

**- HEVC-8K-Dec-8:** The capability of supporting up to eight (*N*=8)concurrent decoder instances with the aggregate capabilities of *HEVC-8K-Dec*.

**- 8K-Dec-8**: The capability supporting up to eight (*N*=8)concurrent decoder instances with either:

- the aggregate capabilities of *AVC-8K-Dec-8*;

- the aggregate capabilities of *HEVC-8K-Dec-8*; or,

- the capability of decoding up to:

- eight bitstreams for which each bitstream does not exceed the capability of being decodable either with *AVC-FullHD-Dec* or *HEVC-FullHD-Dec*; or,

- four bitstreams for which each bitstream does not exceed the capability of being decodable either with *AVC-UHD-Dec* or *HEVC-UHD-Dec*.

## 7.2 Encoding capabilities

### 7.2.1 Single encoder instance

The following video encoding capabilities are defined:

**- AVC-FullHD-Enc:** the capability to encode a video signal to a bitstream that is decodable by a decoder that is *AVC-FullHD-Dec* capable as defined in clause 7.1.1.1 with the following additional constraints:

- up to 245,760 macroblocks per second;

- up to a frame size of 8,192 macroblocks;

- up to 240 frames per second;

- the chroma format being 4:2:0; and

- the bit depth being 8 bit;

**- HEVC-FullHD-Enc:** the capability to encode a video signal to a bitstream that is decodable by a decoder that is *HEVC-FullHD-Dec* capable as defined in clause 7.1.1 with the following additional constraints:

- up to 133,693,440 luma samples per second;

- up to a luma picture size of 2,228,224 samples;

- up to 240 frames per second;

- the Chroma format being 4:2:0; and

- the bit depth being either 8 or 10 bit;

**- HEVC-UHD-Enc:** the capability to encode a video signal to a bitstream that is decodable by a decoder that is *HEVC-UHD-Dec* capable as defined in clause 7.1.1 with the following additional constraints:

- up to 534,773,760 luma samples per second;

- up to a luma picture size of 8,912,896 samples;

- up to 480 frames per second;

- the Chroma format being 4:2:0; and

- the bit depth being either 8 or 10 bit;

## 

### 

### 

# 8 Audio functions and capabilities

## 8.1 Audio/Speech Decoding

The following audio/speech decoding capabilities are defined:

- **EVS-Dec**: thedecodingcapability as defined in TS 26.117 [9] clause 5.2.

- **IVAS-[Editor’s note: IVAS level TBD]-Dec**: the decodingcapability as defined in TS 26.117 [9] clause 5.2.

- **EVS-Dec-2**: the decoding capability as defined TS 26.117 [9] clause 5.2.

- **EVS-Dec-4**: the decoding capability as defined TS 26.117 [9] clause 5.2.

- **AAC-ELDv2-Dec**: thedecoding capability as defined TS 26.117 [9] clause 5.2.

- **AAC-ELDv2-Dec-2**: thedecoding capability as defined TS 26.117 [9] clause 5.2.

## 8.2 Audio/Speech Encoding

The following audio/speech encoding capabilities are defined:

- **EVS-Enc**: the sender requirements for the **EVS** Operation Point as defined in TS 26.117 [9] clause 6.2.4.3.

- **IVAS-[Editor’s note: IVAS level TBD]-Enc**: the sender requirements for the **IVAS** Operation Point as defined in TS 26.117 [9] clause 6.3.5.3.

- **AAC-ELDv2-Enc:** the sender requirements for the **AAC-ELDv2** Operation Point as defined in TS 26.117 [9] clause 6.3.6.3.

# 9

### 9.1

9

### 9.2

p1011-13

0

11-13

11-13

-13

11-3

# 

## 

### 

### 

### 

### 

### 

## 

### 











# 10 Device types and media profiles

## 10.1 Introduction

AR experiences may be running on a variety of devices which have different characteristics and capabilities. Certain capabilities may be common to several devices while other capabilities may be unique to a specific device. Therefore, the present specification enables interoperability by collecting the media capabilities and profiles, defined in clauses 6 and 7. 8 and 9 per device type. The four device types defined are:

- Device type 1: Thin AR glasses (see clause 10.2)

- Device type 2: AR glasses (see clause 10.3)

- Device type 3: XR phone (see clause 10.4)

- Device type 4: XR HMD (see clause 10.5)

NOTE: A given physical device can correspond to more than one device type.

## 10.2 Device type 1: Thin AR glasses

[Editor’s note: For each device type, it is expected that rendering capabilities will also be added before completing the TS.]

### 10.2.1 General

As defined in 4.3.1, the thin AR glasses device type represents a type of device which is considered to be power-constrained and with limited computing power with respect to the other device types. These limitations typically come from the requirement to design a device with a small and lightweight form factor. Regarding rendering capacity, this device type is expected to rely on remote rendering to be able display complex scenes to the user. For example, such device type may run a split rendering session where the split rendering server delivers pre-rendered views of the scene. However, devices in this category can still operate without external support for applications that do not require complex rendering capabilities, for instance, text messaging, 2D video communication, etc. Lastly, the thin AR glasses offers AR experiences to the user via optical see-through display.

### 10.2.2 XR System support

An XR Device complying to the thin AR glasses device has an XR System with at least the following capabilities as based on clause 6.1:

- orientationTracking is 'true'

- positionTracking is 'true'

- Value 'additive' of the enumeration blendMode

- Values 'monoscopic' and 'stereoscopic' of the enumeration viewConfigurationPrimary

- Values 'view', 'local' and 'stage' of the enumeration referenceSpace

- If swapchainSupported is 'true', numberSwapchainImages is equal to 2

- Values 'projection' and 'quad' of the enumeration compositionLayer

NOTE: For the definition of those capabilities, please refer to clause 6.1.

### 10.2.3 Video capabilities support

An XR Device complying to device type 1 shall support at least the following decoding capabilities from clause 7:

- **AVC-FullHD-Dec**

- [**AVC-FullHD-Dec-2**]

An XR Device complying to device type 1 shall support at least the following encoding capabilities:

*-* **AVC-FullHD-Enc**

An XR Device complying to device type 1 should support at least the following decoding capabilities:

*-* **HEVC-FullHD-Dec**

An XR Device complying to device type 1 should support at least the following encoding capabilities:

*-* **HEVC-FullHD-Enc**

### 10.2.4 Audio/Speech capabilities support

[Editor’s note: The audio/speech capabilities for this device type are expected to be further adapted based on the characteristics of this device type.]

An XR Device complying to device type 1 device shall support the following decoding capabilities from clause 8:

- **EVS-Dec**

- **AAC-ELDv2-Dec**

An XR Device complying to device type 1 should support the following decoding capabilities:

- **IVAS-[Editor’s note: IVAS level TBD]-Dec**

- **EVS-Dec-2**

An XR Device complying to device type 1 may support the following decoding capabilities:

- **EVS-Dec-4**

- **AAC-ELDv2-Dec-2**

An XR Device complying to device type 1 shall support the following encoding capabilities:

- **EVS-Enc**

An XR Device complying to device type 1 should support the following encoding capabilities:

- **IVAS-[Editor’s note: IVAS level TBD]-Enc**

- **AAC-ELDv2-Enc**

### 10.2.5 Scene Description capabilities support

TBD

## 10.3 Device type 2: AR glasses

### 10.3.1 General

As defined in 4.3.2, The AR glasses device type represents a type of device which is considered to have higher computation power compared to the thin AR glasses device type. As a result, this device type has higher rendering capacities and is generally expected to be capable of rendering scenes without external support, even though remote rendering is not precluded to lower the power consumption on the device or enable the display of scenes beyond the device’s rendering capability. Lastly, the AR glasses offers AR experiences to the user via optical see-through display.

### 10.3.2 XR System support

An XR Device complying to the AR glasses device type has offers XR System with at least the following capabilities from clause 6.1:

- orientationTracking is ‘true’

- positionTracking is ‘true’

- Value ‘additive’ of the enumeration blendMode

- Value ‘stereoscopic’ of the enumeration viewConfigurationPrimary

- Values ‘view’, ‘local’ and ‘stage’ of the enumeration referenceSpace

- If swapchainSupported is ‘true’, numberSwapchainImages is equal to 2

- Values ‘projection’ and ‘quad’ of the enumeration compositionLayer

NOTE: For the definition of those capabilities, please refer to clause 6.1.

### 10.3.3 Video capabilities support

An XR Device complying to device type 2 shall support at least the following decoding capabilities from clause 7:

*-* **AVC-UHD-Dec**

**- [AVC-UHD-Dec-4]**

**- HEVC-UHD-Dec**

An XR Device complying to device type 2 shall support at least one of the following encoding capabilities:

- **AVC-FullHD-Enc**

**- HEVC-FullHD-Enc**

An XR Device complying to device type 2 should support at least the following decoding capabilities:

**- [HEVC-UHD-Dec-4]**

**- [UHD-Dec-4]**

### 10.3.4 Audio/Speech capabilities support

[Editor’s note: The audio/speech capabilities for this device type are expected to be further adapted based on the characteristics of this device type.]

An XR Device complying to device type 2 device shall support at least the following decoding capabilities:

- **EVS-Dec**

- **AAC-ELDv2-Dec**

An XR Device complying to device type 2 should support the following decoding capabilities:

- **IVAS-[Editor’s note: IVAS level TBD]-Dec**

- **EVS-Dec-2**

An XR Device complying to device type 2 may support the following decoding capabilities:

- **EVS-Dec-4**

- **AAC-ELDv2-Dec-2**

An XR Device complying to device type 2 shall support at least the following encoding capabilities:

- **EVS-Enc**

An XR Device complying to device type 2 should support the following encoding capabilities:

- **IVAS-[Editor’s note: IVAS level TBD]-Enc**

- **AAC-ELDv2-Enc**

### 10.3.5 Scene Description capabilities support

A device of type 2 should support glTF-based scene description as defined in clause 9.2.

If gltf-based scene description is supported, the following requirements and recommendation hold.

- The **SD-Rendering-gltf-Core** capabilities shall be supported

- The **SD-Rendering-gltf-ext1** capabilities should be supported

- The **SD-Rendering-gltf-ext2** capabilities may be supported

- The **SD-Rendering-gltf-interactive** capabilities may be supported

## 10.4 Device type 3: XR phone

### 10.4.1 General

As defined in 4.3.3, the XR phone device type represents a type of device which corresponds to a smartphone with capacities and resources sufficient to offer AR experiences. As a result, this device type is capable of rendering scenes without external support. Lastly, the XR phone offers AR experiences to the user via video see-through display.

### 10.4.2 XR System support

An XR Device complying to the XR phone device type offers an XR System with at least the following capabilities:

- orientationTracking is 'true'

- positionTracking is 'true'

- Values 'opaque' and 'alpha\_blend' of the enumeration blendMode

- Values 'monoscopic' and 'stereoscopic' of the enumeration viewConfigurationPrimary

- Values 'view', 'local' and 'stage' of the enumeration referenceSpace

- If swapchainSupported is 'true', numberSwapchainImages equal to 2

- Values 'projection' and 'quad' of the enumeration compositionLayer

NOTE: For the definition of those capabilities, please refer to clause 6.1.

### 10.4.3 Video capabilities support

An XR Device complying to device type 3 shall support at least the following decoding capabilities from clause 7:

**- AVC-UHD-Dec**

**- [AVC-UHD-Dec-4]**

**- HEVC-UHD-Dec**

**- [HEVC-UHD-Dec-4]**

**- [UHD-Dec-4]**

An XR Device complying to device type 3 shall support at least one of the following encoding capabilities:

*-* **AVC-UHD-Enc**

**- HEVC-UHD-Enc**

An XR Device complying to device type 3 should support at least the following decoding capabilities:

**- AVC-8K-Dec**

**- HEVC-8K-Dec**

**- [8K-Dec-8]**

### 10.4.4 Audio/Speech capabilities support

[Editor’s note: The audio/speech capabilities for this device type are expected to be further adapted based on the characteristics of this device type.]

An XR Device complying to device type 3 device shall support at least the following decoding capabilities from clause 8:

- **EVS-Dec**

- **AAC-ELDv2-Dec**

An XR Device complying to device type 3 should support the following decoding capabilities:

- **IVAS-[Editor’s note: IVAS level TBD]-Dec**

- **EVS-Dec-2**

An XR Device complying to device type 3 may support the following decoding capabilities:

- **EVS-Dec-4**

- **AAC-ELDv2-Dec-2**

An XR Device complying to device type 3 shall support at least the following encoding capabilities:

- **EVS-Enc**

An XR Device complying to device type 3 should support the following encoding capabilities:

- **IVAS-[Editor’s note: IVAS level TBD]-Enc**

- **AAC-ELDv2-Enc**

### 10.4.5 Scene Description capabilities support

A device of type 3 should support gltf-based scene description as defined in clause 9.2.

If gltf-based scene description is supported, the following requirements and recommendation hold.

- The **SD-Rendering-gltf-Core** capabilities shall be supported

- The **SD-Rendering-gltf-ext1** capabilities should be supported

- The **SD-Rendering-gltf-ext2** capabilities should be supported

- The **SD-Rendering-gltf-interactive** capabilities should be supported

## 10.5 Device type 4: XR HMD

### 10.5.1 General

As defined in clause 4.3.4, the XR HMD device type represents a type of device which corresponds to HMDs capable of offering at least AR experiences but not precluding other types of XR experiences. This device type is expected to be capable of rendering scenes without external support. Lastly, the XR phone offers AR experiences to the user via video see-through display.

### 10.5.2 XR System support

An XR Device complying to the XR HMD device type offers an XR System with at least the following capabilities from clause 6.1:

- orientationTracking is 'true'

- positionTracking is 'true'

- Value 'additive' or values 'opaque' and 'alpha\_blend' of the enumeration blendMode

- Values 'monoscopic' and 'stereoscopic' of the enumeration viewConfigurationPrimary

- Values 'view', 'local' and 'stage' of the enumeration referenceSpace

- If swapchainSupported is 'true', numberSwapchainImages is equal to 2

- Values 'projection' and 'quad' of the enumeration compositionLayer

NOTE: For the definition of those capabilities, please refer to clause 6.1.

### 10.5.3 Video capabilities support

An XR Device complying to device type 4 shall support at least the following decoding capabilities from clause 7

- **AVC-UHD-Dec**

- **[AVC-UHD-Dec-4]**

- **HEVC-UHD-Dec**

- **[HEVC-UHD-Dec-4]**

- **[UHD-Dec-4]**

An XR Device complying to device type 4 shall support at least one of the following encoding capabilities:

- **AVC-UHD-Enc**

- **HEVC-UHD-Enc**

An XR Device complying to device type 4 should support at least the following decoding capabilities:

- **AVC-8K-Dec**

- **HEVC-8K-Dec**

- **[8K-Dec-8]**

### 10.5.4 Audio/Speech capabilities support

[Editor’s note: The audio/speech capabilities for this device type are expected to be further adapted based on the characteristics of this device type.]

An XR Device complying to device type 4 device shall support at least the following decoding capabilities from clause 8:

- **EVS-Dec**

- **AAC-ELDv2-Dec**

An XR Device complying to device type 4 should support the following decoding capabilities:

- **IVAS-[Editor’s note: IVAS level TBD]-Dec**

- **EVS-Dec-2**

An XR Device complying to device type 4 may support the following decoding capabilities:

- **EVS-Dec-4**

- **AAC-ELDv2-Dec-2**

An XR Device complying to device type 4 shall support at least the following encoding capabilities:

- **EVS-Enc**

An XR Device complying to device type 4 should support the following encoding capabilities:

- **IVAS-[Editor’s note: IVAS level TBD]-Enc**

- **AAC-ELDv2-Enc**

### 10.5.5 Scene Description capabilities support

A device of type 4 should support gltf-based scene description as defined in clause 9.2.

If gltf-based scene description is supported, the following requirements and recommendation hold.

- The **SD-Rendering-gltf-Core** capabilities shall be supported

- The **SD-Rendering-gltf-ext1** capabilities should be supported

- The **SD-Rendering-gltf-ext2** capabilities should be supported

- The **SD-Rendering-gltf-interactive** capabilities should be supported

# 11QoE M

## 11

### 11

### 11

### 11

### 11

### 11

## 11

### 11

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

# 12 Metadata formats

### 12.1 General

TBD

### 12.2 Pose information format

The pose information is described by the poseInfo object.

The structure and the attributes of the poseInfo object are defined in Table 10.2.2-1. [Ed.note: table to be aligned with split rendering spec]

Table 10.2.2-1 - Pose information 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. [Ed: spec?] |
| 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. |
| confidence | number | 0..1 | This optional parameter provides a confidence score that reflects the probability for this pose prediction to be correct. For the current pose or a pose in the past, the confidence value would be 1. The confidence can take a value between 0 and 1.  If not provided by the XR runtime, this field may be estimated by the SRC or omitted. |
| 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. |

### 12.3 Action format

Actions are grouped into action sets which may be activated and deactivated during the lifetime of an XR session.

The action information is described by the actionSets object. The structure and the attributes of the actionSets object are defined in Table 6.2.3-1. [Ed.note: table to be aligned with split rendering spec]

Table 10.3-1 - 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 | number | 1..n | An array of objects that conveys information about the actions of the parent action set. |
| identifier | number | 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. |

### 12.4 Available Visualization Space format

The XR Application may define a three-dimensional space within the user’s real-word space that is suitable for rendering virtual objects called the Available Visualization Space. Such a space is defined with a shape which is either cube or sphere with the corresponding size and coordinates. In the case that the virtual scene is rendered by a remote entity (e.g. split rendering), this Available Visualization Space may be transmitted to this remote entity so that the composed AR objects remain within the defined Available Visualization Space. The method of calculating the Available Visualization Space is out of the scope of this document.

The content of the availableVisualizationSpace type shall follow the format defined in Table 10.4-1.

Table 10.4-1 – Available Visualization Space

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Cardinality | Description |
| availableVisualizationSpace | Object | 0..1 | An object defining the coordinate of the available visualization space. |
| cuboid | Object | 0..1\* | The available visualization space in form of cuboid |
| x | float | 1 | Offset of the available visualization space starting point in the x direction ias defined by the Open XR coordinate system in meters.  The value is in meters. |
| y | float | 1 | Offset of the available visualization space starting point in the y direction as defined by the Open XR coordinate system.  The value is in meters. |
| z | float | 1 | Offset of the available visualization space starting point in the z direction as defined by the Open XR coordinate system.  The value is in meters. |
| width | float | 1 | The width of available visualization space in the x direction as defined by the Open XR coordinate system.  The value is in meters. |
| height | float | 1 | The height of available visualization space in the y direction as defined by the Open XR coordinate system.  The value is in meters. |
| depth | float | 1 | The depth of available visualization space in the z direction as defined by the Open XR coordinate system.  The value is in meters. |
| sphere | Object | 0..1\* | The available visualization space in form of cuboid |
| x | float | 1 | Offset of the available visualization space center in the x direction as defined by the Open XR coordinate system.  The value is in meters. |
| y | float | 1 | Offset of the available visualization space center in the y direction as defined by the Open XR coordinate system.  The value is in meters. |
| z | float | 1 | Offset of the available visualization space center in the z direction as defined by the Open XR coordinate system.  The value is in meters. |
| radius | float | 1 | The radius of available visualization space as defined by the Open XR coordinate system.  The value is in meters. |
| \*Only one of cuboid or sphere object shall exists. | | | |

With this Available Visualization Space, a user may for instance avoid the virtual objects to be occluding other real objects in the scene, e.g. TV sets, people, etc., since an AR experience is achieved by the integration of visual objects into the user environment.

|  |
| --- |
| **End of First Change** |