**Agenda item:** 9.5

**Source:** Ericsson LM

**Title: Media capability for RTC**

**Document for** Discussion andAgreement

# Introduction

S4-221272 (MeCAR Permanent Document) is describing the architecture and APIs for AR glasses. Three device APIs have been defined: XR run-time API, XR scene API, and MAF API.

In this contribution, we describe the device APIs for Real-time Communications (RTC) considering AR Conferencing use-case.

# Changes

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

5.7 Media capability for RTC

TR 26.998 defined several real-time communication use-cases. Use cases 19 and 22 in TR 26.998 refer to AR conferencing scenarios. The media capabilities (APIs) for RTC (AR conferencing use-cases) should be defined based on the device architecture (Section 4.2.1).The follow-up actions to address the specification work regarding the aspects described in this table will not be handled as part of the MeCAR Work Item. They may however be considered in other SA4 work Items like iRTCW and SmarTAR.

|  |  |  |  |
| --- | --- | --- | --- |
| API | Mapping to use-case (AR conferencing) | List of APIs | Consumer of the API |
| Media Access Functions (MAF) | APIs for uplink media transmission such as compressed RGB and depth.  APIs for downlink media transmission such as point cloud, 3D mesh, RGB-D, or 2D video (split rendering). | * Authentication between devices (e.g., using tokens) * Create a session between device and server * Transmit uplink streams using publish or producer patterns * Connect to the session – both peers (sender and receiver) * Receive downlink streams using subscribe or consumer patterns * Download 3D objects, scene description, or other data | Environment can be browser or game engine or stand-alone app (Windows, iOS, Linux) for sender (camera client). Environment is game engine for receiving data, because at the moment this is no browser support for glasses yet.  MAF is tightly coupled with WebRTC libraries/SDKs. |
| AR Scene manager | APIs for rendering of 3D objects. | * Create an AR scene * Update a scene * Create a node in the scene * Assign decoded 3D data to the node * Render a scene | Game Engine for XR glasses or native development environment for ARCore/ARKit. No browser support for AR glasses is available at the moment. However, three.js can create a scene, render objects as well. |
| AR Run-time | APIs for positioning of 3D objects in the scene. | * Rendered frame is provided to AR run-time * Position of user (e.g., 6DoF) | Glasses SDK. In future, it could be potentially done through game engine only. |

# Proposal

We propose to include the changes in Section 2 to permanent document.