**Agenda item:** 10.6

**Source:** Qualcomm Inc.

**Title: [IBACS] Transcoding for Network Rendering**

**Document for** Discussion andAgreement

# Introduction

TS23.228 extends the IMS architecture to add support for data channel service in IMS. AR calls are enabled through the exchange of scene description and scene description updates over data channels. AR calls are supported by an MRF or a Media Function. The Media Function uses a service-based interface to interact with the Data Channel AS. The architecture is depicted in the following figure:



The call setup is described by the following call flow diagram:



AR calls may require significant processing resources to render the content of the AR call scene, especially when multiple participants contribute to the creation of a complex scene. These scenes may consist of a virtual environment that may be anchored to the user’s real world, as well as content from all participants in the call. Content from a participant may for example be their Avatars, slide materials, 3D virtual objects, etc.

Furthermore, PBR rendering would require emulating the light transmission in that virtual world to recreate things like user shadows and object reflections on specular surfaces.

These advanced rendering capabilities may not be available to AR glasses and HMDs or may be too power hungry to run on these devices.

TS23.228 suggest the definition of a Network-centric Procedure that enables remote rendering of the AR content based on a request from the sender of AR content. This call setup is depicted by the following figure, where step 2 shows the decision by the UE-A to trigger network rendering:



In this contribution, we propose to handle network rendering as a transcoding operation that is triggered by the IMS AS for clients that do not possess the required AR processing capabilities. The MF may act on behalf of that endpoint to produce corresponding AR content, e.g. place a 2D overlay video of that participant on a 3D screen in the virtual 3D scene.

# Transcoding AR calls

The service uses the data channel to distribute the scene description for the AR call. The scene description is used to compose the scene that will serve as the shared space for all participants in the AR call.

Each participant shall declare support for the AR call as well as its rendering capabilities in the invite. The scene description that it receives will be tailored to the rendering capabilities, potentially offering a few alternative representations that the UE needs to choose from.

The procedure for the call setup is described in the following call flow:



The steps are as follows:

1. UE1 wants to start an AR call with UE2 or join an ongoing AR conference. UE1 generates the SDP for the offer that indicates that it can only receive 2D content. The offer may still indicate that it can send pose information and other 2D/3D media. UE1 sends the invite to the I/S/P-CSCF, which forwards it to the IMS AS.
2. The IMS AS identifies the capabilities of UE1 and decides to invoke network rendering functionality for the AR call. It rewrites the SDP to redirect the 3D media and scene content to the MF.
3. The IMS AS negotiates the data channel resources for the session with the MF/MRF.
4. If no network rendering is needed, the IMS AS forwards the invitation to the I/S/P-CSCF, including the data channel information.
5. The IMS AS allocates via the DCSF the resources for the transcoding of the 3D content.
6. The IMS AS then forwards the updated invite to the I/S/P-CSCF. This invitation indicates that the 3D content should be routed to the MF.
7. The I/S/P-CSCF forwards the invite that was generated in step 6 to UE2.
8. UE2 informs the IMS AS that it accepts the call invite.
9. The AR AS sends scene updates to UE2 and to the MF.
10. The MF transcodes the 3D scene by performing network rendering.
11. The MF streams the rendered media to UE1.
12. Both AR AS and UE2 establish the connection to the eMRF to receive the scene description for the call.
13. The connections between UE1 and the AR AS and between AR AS and UE2 are established.

This procedure allows the network to invoke remote rendering for the AR call, without explicit intervention of the UE. The IMS AS is responsible for selecting the appropriate network MF that will perform the network rendering operation for the session.

The AR AS is the central entity for the AR call. It manages the scene description for the session and performs scene composition. The endpoints send their scene updates and pose information to the AR AS over the data channel.

# Signaling

UE1 may be aware that it is joining an AR call or it may not be aware of that. If it is aware of that, it may want to share information about its display capabilities and may share information about its current viewer pose for XR remote rendering.

To share this information with the IMS AS, UE1 uses new defined SDP attributes. The IMS AS will detect the signaling and use that information to configure the MF remote rendering session.

The SDP signaling should include:

* Display configuration, e.g. the access to an HMD
* OpenXR support: the supported view configuration and projection layers
* GPU capabilities such as support for different rendering pipelines as well as the supported scene complexity

A new SDP session level attribute is proposed to indicate these capabilities for the session setup phase.

The ABNF syntax for the attribute is provided here:

display-attr=”a=display-capabilities:” view-config SP \*(projection-param / render-param)

view-config=”view=” “2D” / “Stereo” / “Stereo+Depth” / “Stereo+Transparency” / “Stereo+Depth+Transparency”

projection-param=”projection=” 1\*((“Quad” / “Equirectangular” / “Depth”) \*1(“;”))

The IMS AS detects this attribute and decides to invoke remote rendering capabilities for the AR call.

The absence of the attribute may indicate that the device has no AR capabilities. If the other participant is offering 3D content, the IMS AS should also invoke network rendering capabilities to convert from 3D to 2D.

# Proposal

We propose to agree the call flow into the IBACS PD.