**Source: Xiaomi**

**Title: On frame submission to the AR Runtime in EDGAR-1 architecture**

## Document for: Agreement

## Agenda Item: 9.5

# 1 Introduction

At the 3GPP meeting #119-e, SA4 published the Permanent Document (PD) for MeCAR v2.0 [1]. During the post 119-e period, discussion on MeCAR took place during scheduled Video SWG telcos. This contribution proposes updates to the clause 4.2.1 Device architecture for EDGAR-1 device category based on these discussions around the interface between AR Scene Manager and the AR Runtime.

# 2 Background on OpenXR rendering

2.1 Rendering cycles

As described in the OpenXR Reference Guide [2], an OpenXR application is composed of different cycles as depicted in Figure 1.

Diagram

Description automatically generated

Figure 1 - OpenXR application lifecycle [2]

In terms of rendering operation, the relevant part is located between the call to xrBeginFrame and the call to xrEndFrame on the bottom right part of the diagram.

When the application calls the xrEndFrame function, the application provides the structure XrFrameEndInfo which contains all necessary information to render the frame that is:

* The time at which this frame should be displayed.
* The mode to be used for blending the user’s envriromnent with the submitted frame
* One or more layers which composes the submitted frame

As documented in the OpenXR specification:

“XrFrameEndInfo may reference swapchains into which the application has rendered for this frame. From each XrSwapchain only one image index is implicitly referenced per frame, the one corresponding to the last call to xrReleaseSwapchainImage.”

This describes how the runtime and the application can exchange visual data, i.e. via the use of swapchains.

2.2 Swapchains

Swapchains are a generic mechanism for computer systems to manage the generation and the display of images. As commonly defined “a swap chain (also swapchain) is a series of virtual framebuffers utilized by the graphics card and graphics API for frame rate stabilization and several other functions. The swap chain usually exists in graphics memory, but it can exist in system memory as well.” [3].

The OpenXR API allows an application to request the creation of swapchains using the xrCreateSwapchain function according to a specific format supported by the platform. The supported formats can be queried by the xrEnumerateSwapchainFormats function.

# 3 Proposal

3.1 General

The proposal is as follows :

* Add an arrow from the AR Scene Manager to the AR Runtime where frames are submitted.
* Add a Swapchains API seating on the AR runtime to allow submitting the images in the Swapchains.
* Add text describing the usage of Swapchains.

3.2 Proposed updated clause

4.2.1 Device architecture

[Editor’s note] At SA4#119, this section was added while further improvements were improved.

Figure 2 provides the technical architecture of EDGAR-1 UE.

Diagram

Description automatically generated

**Figure 2 - Device architecture of EDGAR-1 device**

The EDGAR-1 is regular 5G UE with 5G connectivity provided through an embedded 5G modem and 5G system components. The EDGAR-1 UE also features several sensors and user controllers relevant for AR experiences that are cameras, microphones, speakers, display and generic user input. The AR/MR Application is responsible for orchestrating the various device resources to offer the AR experience to the user. In particular, the AR/MR Application can leverage three main internal components on the device which are:

* The Media Access Functions (MAF)
* The AR Runtime
* The AR Scene Manager

The AR/MR Application can communicate with those three components via dedicated APIs called the MAF-API, the AR Scene Manager API and the AR Runtime API. Among other functionalities, those APIs enables the AR/MR Application to discover and query the media capabilities in terms of support as well as available resources at runtime. Regarding rendering, the AR/MR application obtains the head pose information from the AR Runtime which is then provided to the AR Scene Manager. Based on this information, the AR Scene Manager determines the objects visible to the user at a given point in time or more generally the objects that may be needed to be rendered in the next rendering cycles. The AR Scene Manager then submits the rendered views to the AR Runtime as frames written to the images of the Swapchains which formats where configured beforehand by the AR/MR Application using the information provided by the AR Runtime API. From those images in the Swapchains, the AR Runtime then generates the left and right eye buffers possibly based on late adjustment techniques using updated head pose information, if available, commonly known as late stage reprojection (LSR).

Once the AR/MR application is running, the downlink media flows from the 5G System to the MAF in compressed form and then from The MAF to the AR Scene Manger in a decoded form. In parallel, the EDGAR-1 UE is capable of establishing an uplink data flow from the AR Runtime to the MAF wherein the data may be in an uncompressed form and then from the MAF to the 5G System wherein the MAF may have compressed the data in order to facilitate the expected transmission over the network.

# References

1. S4-220760, MeCAR Permanent Document v2.0, 3GPP TSG SA WG4 119-e Meeting, 11th – 12th May 2022
2. OpenXR 1.0 Reference Guide, <https://www.khronos.org/files/openxr-10-reference-guide.pdf>
3. Wikipedia contributors, "Swap chain," *Wikipedia, The Free Encyclopedia,* <https://en.wikipedia.org/w/index.php?title=Swap_chain&oldid=1053599560> (accessed August 11, 2022).