**Source: Xiaomi (Rapporteur)**

**Title: MeCAR Permanent Document v2.0**

**Version: 2.0.0**

**Agenda Item: 15.4**

**Document for: Agreement**

**Work Item information:**

|  |  |
| --- | --- |
| **Work Item Name** | Media Capabilities for Augmented Reality |
| **Acronym** | MeCAR |
| **Unique ID** | 950015 |
| **Targe release** | 18 |

**Source details:**

|  |  |  |
| --- | --- | --- |
| **Rapporteur** | **Name** | Emmanuel THOMAS |
| **Company** | Xiaomi |
| **Email** | thomase@xiaomi.com |
| **Editor** | **Name** | Gilles TENIOU |
| **Company** | Tencent |
| **Email** | teniou@tencent.com |

**Revision history:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Meeting** | **Subject/Comment** |
| 1.0.0 | 2022-04-14 | SA4#118e | * Added
	+ Working assumptions
	+ Example media capabilities
 |
| 2.0.0 | 2022-05-20 | SA4#119e | * Added
	+ Clause 2 on definitions, including Optical see-through device and Video see-through device
	+ Clause 4 on device categories, including draft EDGAR-1 architecture (4.2)
	+ Clause 5.3 on media capability validation framework, including Khronos 3D Commerce conformance example (5.3.1), a possible media capability evaluation framework (5.3.2) and its MeCAR scope (5.3.3)
	+ Clause 6 on sensor and user environment data types including the view-related information (6.2)
	+ Clause 7 on activities in external organizations, including the AVTCORE WG in IETF (7.1)
 |

# Contents

[Contents 2](#_Toc103878074)

[1 Introduction 3](#_Toc103878075)

[2 Definitions, symbols and abbreviations 3](#_Toc103878076)

[2.1 Definitions 3](#_Toc103878077)

[3 Working assumptions 4](#_Toc103878078)

[3.1 Prioritization of AR optical see-through 4](#_Toc103878079)

[3.2 Device design 4](#_Toc103878080)

[3.3 General functional architecture 5](#_Toc103878081)

[3.4 5G\_STAR EDGAR-type device architecture 6](#_Toc103878082)

[3.5 Media Access Function for AR 6](#_Toc103878083)

[4 Device categories 7](#_Toc103878084)

[4.1 General 7](#_Toc103878085)

[4.2 External Display Glasses for AR version 1 (EDGAR-1) 7](#_Toc103878086)

[4.2.1 Device architecture 7](#_Toc103878087)

[5 Media capabilities 8](#_Toc103878088)

[5.1 Categories of media capabilities 8](#_Toc103878089)

[5.2 Examples of media capabilities 9](#_Toc103878090)

[5.3 Media capability validation framework 10](#_Toc103878091)

[5.3.1 Example framework by Khronos on 3D Commerce conformance (glTF viewer) 10](#_Toc103878092)

[5.3.2 Possible capability evaluation framework 12](#_Toc103878093)

[5.3.3 Possible scope of media capability 13](#_Toc103878094)

[6 Sensor and user environment data types 13](#_Toc103878095)

[6.1 General 13](#_Toc103878096)

[6.2 View-related information 13](#_Toc103878097)

[7 Relevant activities in external organizations 14](#_Toc103878098)

[7.1 IETF AVTCORE WG 14](#_Toc103878099)

[8 Technical status 14](#_Toc103878100)

[8.1 List of elements open for work 14](#_Toc103878101)

[8.2 List of completed elements 15](#_Toc103878102)

[8.3 List of remaining elements 15](#_Toc103878103)

[8.4 List of open issues 15](#_Toc103878104)

[9 References 15](#_Toc103878105)

# 1 Introduction

During SA4#117-e the New Work Item on “Media Capabilities for Augmented Reality” (MeCAR) in S4-220332 was agreed and afterwards approved in by SA#95e in SP-220242.

The media capabilities of AR devices typically contribute to three main functionalities that are simple media rendering, split-rendering, for which a pre/scene-rendering of the scene and views is carried out in the cloud/edge and uplink streaming of sensor and device data to the network in order to support network-based processing of device sensor information.

To support basic interoperability for AR applications in the context of 5G System based delivery, a set of well-defined media capabilities are essential to create the conditions of a successful ecosystem. Therefore, MeCAR work item defines those media capabilities for AR devices in a service-independent manner. In particular, the following objectives are considered:

* Define at least one AR device category that addresses the constraints of an EDGAR-type AR glass
	+ Note: Additional device categories may be defined, but with lower priority.
* For each AR device category
	+ Define a reference terminal architecture regarding media capability aspects for this AR device category
	+ Define media types and formats produced and consumed by the AR device, including basic scene descriptions, audio, graphics and video as well as sensor information and metadata about user and environment.
	+ Define the integration of the relevant existing 3GPP codecs into the reference terminal architecture
	+ Define decoding capabilities, including support for multiple parallel decoders
	+ Define encoding capabilities
	+ Define security aspects related to the media capabilities
	+ Define the required, recommended and optional media capabilities for this AR device category
* Integrate IVAS into suitable AR device categories, once IVAS is available
* Define capability exchange mechanisms based on complexity of AR media and capability of device to support EAS KPIs for provisioning of edge/cloud resources
	+ Note: Identify a suitable existing capability framework, or if it does not exist, we need to work with the broader industry (e.g., IETF, KHRONOS, W3C, etc.) to get this done.
* Identify which QoE metrics from VR QoE metrics can be reused or enhanced for AR media (e.g., resolution per eye, Field of view (FOV), round-trip interaction delay, etc.) and define relevant KPIs that are dedicated to AR/MR
* Specify additional relevant KPIs and simple QoE Metrics for AR media
* Specify encapsulations into RTP, ISOBMFF and CMAF
* Specify the relevant codec-level parameters for session setup and negotiation of the media delivery and provide instantiations for SDP and DASH MPD
* Enable AR media in 5G Media Streaming by defining suitable 5GMS profiles based on AR media capabilities
* Define typical traffic characteristics for AR media

# 2 Definitions, symbols and abbreviations

## 2.1 Definitions

**Optical see-through device**:Device providing a view of the surrounding world by letting the light from the real world directly reaches the user’s eyes through an optical system.

**Video see-through device**: Device providing a view of the surrounding world by capturing the light from the real world and then presenting it through a display system to the user.

# 3 Working assumptions

## 3.1 Prioritization of AR optical see-through

Optical see-through devices have the advantage of not requiring any additional video streams for rendering the surrounding environment. The natural light passing through the lenses of the device provides the user with a clear and natural view of the surrounding environment and does not add additional rendering latencies since the only media data that needs to be streamed by the device is the AR data. Therefore, even display systems with limited capabilities can offer the user a high level of experience, as demonstrated by the Everysight system.

Prioritizing optical see-through devices will allow us to focus on the overlaid AR data that is displayed on the glasses and synchronized (time and space) with the real word. It will make possible to offer a good level of user experience relying on relatively lower KPIs, in terms of FoV, resolutions, etc., and leads to lighter constraints on the design of the glasses, making easier the emergence of near to mid-term solutions.

## 3.2 Device design

Looking at existing AR Glasses, based on the study in TR 26.998 [1] and based on information from chipset manufacturers on existing and emerging devices, an AR Glass designed for AR experiences does integrate complex functionalities and many of those relate to capabilities. Figure 1 is a picture providing an overview of an AR glass.

Hinge

SoC Media

Connectivity

Eye Tracking + Camera/Sensor Aggregator

Figure 1 - Overview of an AR glass

 Typical functions of such a AR glass consists of:

* Peripheries including
	+ Displays
	+ Cameras
	+ Microphones
	+ Sensors
	+ Camera/Sensor Aggregators
	+ Perception functionality: Eye Tracking, Face Tracking, etc.
* SoC Media
	+ Display Processing
	+ GPU functionalities: Composition/Reprojection
	+ Decoding
	+ Decryption
	+ Camera Front ends
	+ Perception functionality: 6DoF, etc.
	+ Encoding
* Connectivity
	+ Wi-Fi, Bluetooth, 5G, etc.

An interesting aspect to consider from the above is that the device consists of different thermal islands, hence division in multiple chips in the headset is highly desirable. This means that both minimizing the power consumption per thermal island as well as minimizing the overall power consumption is an essential design constraint for the device battery life.

In addition, such type of devices require to partition workloads to remote devices or the cloud to some extent to balance the power load. Based on this, media capabilities are also possibly required on UE that acts as a hub for a tethered glass. Architectures and processing for this will be discussion SmartAR. The main target device in the MeCAR work item remains glasses as shown above.

It should be noted that such AR glasses are predominantly served with media that can directly be rendered by the peripheries, or produce media captured on the device and sent to remote processing.

It is considered that for media capabilities related to this primary AR category, only capabilities of the SoC media are to be part of the media capability definitions. We also note that the XR experience observed by the user depends on more aspects than the media capabilities, such as the display, the optics, the quality of the sensors, the stability of the connection and so on. However, such aspects are not considered to be part of the media capabilities for AR.

Initial System-on-Chip (SoC) media will likely rely on existing hardware, for example from lower end mobile chipsets. Some people consider XR even a hack that uses existing components in a smart manner. However, a core aspect of XR experiences different from traditional mobile devices is the concurrent operation of multiple encoders and/or decoders to address different sensors, eye buffers, layers and so on, as well as the rendering to GPU instead of directly going to the display.

Only over time, such hardware will get added specific functionalities, but not in the near and mid-term. Expected in the future are higher render and display resolutions, multi-layer composition, etc.

Given that many functionalities are defined through Khronos OpenXR, defining capabilities for example by mandating or recommending support of certain APIs or parameter settings on API may be relevant. In some cases it may not even be possible to define capabilities, but for example rely on test signals and benchmarking requirements that estimate the performance of a device.

Based on these observations, an initial main objective of a standard is to create near to mid-term interoperability for media capabilities based existing and emerging media SoCs.

## 3.3 General functional architecture

For any type of AR devices targeted by MeCAR, the functional architecture depicted in Figure 2 is applicable.



Figure 2 - General functional architecture of AR device

## 3.4 5G\_STAR EDGAR-type device architecture

From TR 26.998 [1], the architecture of the EDGAR device type was defined as illustrated in Figure 3. Note that EDGAR in TR 26.998 stands for EDGe-dependent AR (EDGAR) UE.



Figure 3 - Architecture of 5G\_STAR EDGAR-type device

## 3.5 Media Access Function for AR

The Media Access Function defined in TR 26.998 [1] supports the AR UE to access and stream media. Figure 4 depicts its different functions and buffer elements.



Figure 4 - Media Access function for AR as defined in TR 26.998 [1]

# 4 Device categories

## 4.1 General

This clause collects the work carried out in the MeCAR Work Item related to the various device categories.

## 4.2 External Display Glasses for AR version 1 (EDGAR-1)

### 4.2.1 Device architecture

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

Figure 5 provides the technical architecture of EDGAR-1 UE.



Figure 5 - 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.

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.

# 5 Media capabilities

## 5.1 Categories of media capabilities

Media capabilities may be defined for those categories:

* Audio
	+ Capture
	+ Playback
	+ Codec
	+ Formats
	+ Framework (multiple codecs, etc.)
* Camera
	+ RGB
	+ Depth
* Display
	+ Processing
	+ Number of Displays
	+ Bit depth
	+ Color format
* GPU
	+ Functionalities/APIs
	+ Performance
* Security
	+ Content Protection
	+ Cryptography
	+ Key Management
* Non-media sensors
	+ Types: Accelerometer, Magnetometer, Gyroscope, ambient light
	+ Access for example through OpenXR APIs
* Video
	+ Playback/Decoding
	+ Processing
	+ Recording/Encoding
	+ Formats (bit depth, color components, chroma subsampling, etc.)
	+ Framework (multiple codecs, etc.)
* Runtime
	+ APIs
	+ Performance

## 5.2 Examples of media capabilities

Given the categories listed in clause 3.1, the following are examples of media capabilities for those categories:

* Video
	+ Playback/Decoding
		- H.264 High, Main and Baseline profile
		- H.265 Main and Main 10 Profile
		- Maximum processing: Up to 8,294,400 Macroblocks per second (corresponding to 8192x4320 @ 60fps)
		- HEIF
	+ Processing
	+ Recording/Encoding
		- H.264 High, Main and Baseline profile
		- H.265 Main and Main 10 Profile
		- Maximum processing: Up to 3,888,000 Macroblocks per second (corresponding to 3840x2160 @ 120fps)
		- Low-latency encoding
		- Error-robustness, slicing, intra refresh, long term prediction
	+ Formats
		- 8-Bit: NV12, UBWC, YV12, RGBA888
		- 10-Bit: UBWC TP10, P010
	+ Framework (multiple codecs, etc.)
		- Maximum number of combined encoding and decoding instances: 16
* GPU
	+ Functionalities
		- tbd
	+ Performance
		- Examples
			* 3D Triangle Rate
			* 3D Pixel Draw Rate
			* Texture Fetch Rate
			* Z reject rate (pixels/sec)
		- The issue is that GPU capabilities are more defined through benchmarks. A way to address is to define a set of test signals that a GPU needs to be able to handle in real-time.
* Audio
	+ Capture
	+ Playback
	+ Codec
	+ Formats
	+ Framework (multiple codecs, spatial audio support etc.)
		- Low-Latency: input, output, roundtrip
		- Game Audio Playback up to 8/16/32 simultaneous streams

## 5.3 Media capability validation framework

### 5.3.1 Example framework by Khronos on 3D Commerce conformance (glTF viewer)

#### 5.3.1.1 General

The Khronos group defines many specifications that rely on hardware capabilities and, in particular, its specifications are largely powered by Graphics Processing Units (GPU). As a result, the deployment of Khronos specification depends significantly on the ability for a vendor to evaluate whether its products meets the requirement of those specifications.

To this end, Khronos offers the Khronos 3D Commerce Viewer Certification Program which “enables any company to demonstrate that their viewer is capable of accurately displaying 3D Products that have been created using the 3D Commerce asset creation guidelines”.

The relevant part in the context of MeCAR is the certification process described in [2].



Figure 6 - Khronos' 3D commerce certification process

#### 5.3.1.2 Relevant steps in the MeCAR context

From this certification process only a subset of those steps are relevant for us which are:

* Viewer Test Package
	+ What does it contain? What are the file formats?
* Run Certifications Test
	+ How are those test described? Are the test objective or subjective? On which criteria and/or metrics do they rely on?
* Generates Results packages
	+ How are expressed, in format, the performance of a 3D viewer against the tests? Is the result binary, i.e. passed/not passed? Or a score on a given scale with a minimum threshold?

To answer, those questions more documentation is available at the Khronos Group 3DC Certification repository [3]. The following was found based on the available documentation.

* Viewer Test Package
	+ The package contains a list of glTF models [4]:
		- AnalyticalCubes
		- AnalyticalGrayscale
		- AnalyticalSpheres
		- GreenChair
		- Mixer
		- Shoe
		- TennisRacquet
		- WickerChair
* Run Certifications Test
	+ The test plan defines how the tested viewer must operate to render the test models:
		- “The Certification Program Test Plan document defines the detailed requirements for generating the certification images.”
	+ Some test are verified by mathematical functions some by humans.
		- “Certification renders will be evaluated programmatically and through human checks”
		- Example of subjective test:
			* “Strings should appear translucent outside of the blue star area”
		- Example of objective test:
			* “When scored by the evaluation tool included in the repository an SSIM or PSNR lower than their respective thresholds will automatically flag the image for review.”
* Generates Results packages
	+ To evaluate whether a glTF viewer is conformant, the tested renders must generate images from the glTF model and those images are programmatically verified against reference renders.
		- “All certification images must be 1024x1024 and displayed according to the embedded cameras. The five retail models have three cameras each. One of the analytical models (spheres) is displayed in four different IBLs. All certification images need to be created according to the rules specified in the Test Plan document.”
	+ How are expressed, in format, the performance of a 3D viewer against the tests? Is the result binary, i.e. passed/not passed? Or a score on a given scale with a minimum threshold?

#### 5.3.3.3 Takeaways from the certification process

Here are some takeaways from the certification test:

* A set of test models is essential for defining the test and the evaluation criteria.
* Objective tests are a minimum to pass but subjective tests via human verification are here to confirm for hard cases, e.g. transparency, reflection, etc.
* For objective tests, PSNR or SSIM is used to evaluated the rendered images from the test models.
* The tests are limited to static images and not rendering of the models over time.

### 5.3.2 Possible capability evaluation framework

In the context of MeCAR, the goal is not to certify a device but to define the media capabilities that are required at minimum for a given device category. The figure below depicts a possible workflow for implementing the evaluation of graphics capabilities in rendering glTF models and scenes.



Figure 2 - Possible framework for defining media and graphics capabilities

The first type of requirements is the playback of the test vectors. The test vectors are composed of a set of glTF tests models and scenes as well as pose traces. The MeCAR UE is supposed to render views of those glTF test models under the given poses coded in the test pose traces). The second type of requirements is whether the playback of the test vectors is correct. To this end, the generated views could be considered as a rendered videos (similar to the rendered image in the Khronos example). Such videos could be then checked against a reference video for the given test vector. The video validator could verify for the entire video:

* correct number of frames
* correct frame rate
* correct coded resolution of frames
* correct chroma sampling
* correct bit depth
* correct disparity between left and right views
* correct timing with respect to real-time rendering constraints

For each frame, the video validator could verify that each rendered image does not deviate too much from the reference image in the reference video. To validate the real-time nature of the rendering, the test run environment should also limit the time allowed to run the test scene.

### 5.3.3 Possible scope of media capability

In contrast to the Khronos example, the goal in MeCAR is not to establish a certification process. As a result, we would define the scope of the MeCAR graphics capability that does not fully cover the framework described in clause 3.1. The possible scope would cover he following elements:

* The glTF test models (possibly included media assets).
* The test pose traces associated with the glTF test models. The pose traces could be specific to each glTF test model.
* The test plan that defines the criteria to evaluate the rendered video (resolution, number of frames, etc…)
* Optionally, the generation of the reference rendered videos could be included to facilitate the reuse of this framework. However, since MeCAR may not define the reference scene render, providing these reference rendered video may actually go beyond MeCAR scope. This should be further discussed.

# 6 Sensor and user environment data types

## 6.1 General

This clause collects data types that can be consumed or produced by a MeCAR device and relates to Sensor and user environment data types.

## 6.2 View-related information

The device may generate static or dynamic data streams related to view information. The different types of data are listed as follows.

* Device pose information
	+ Position in user’s volumetric space (maybe relative coordinate to user’s space origin)
	+ Position in global space (maybe absolute coordinate)
	+ Viewing direction in user’s volumetric space (maybe relative direction to user’s space origin)
	+ Viewing direction in global space
	+ Device pose timestamp
* Device frustum information
	+ Field of view
	+ Aspect ratio
	+ Z-near
	+ Z-far
	+ Resolution of display
	+ Screen to eye distance
	+ Eye gaze point (on display or space)

The device may also receive media content representing pre-rendered views of a scene along with descriptive information. The different types of data are listed as follows.

* Pre-rendered media
* Codec, resolution, and profile of the pre-rendered media
* Estimated pose used to generate the pre-rendered media
* Estimated presentation time used to generate the pre-rendered media
* Frustum information of the pre-rendered media (in case device does not provide its frustum information)

# 7 Relevant activities in external organizations

## 7.1 IETF AVTCORE WG

MPEG has published a group of standards, under the umbrella of Visual Volumetric Video-based Coding (V3C). The V3C family of standards covers the aspects of encoding, storage, and transport of volumetric video and consists of three documents:

1. ISO/IEC 23090-5 Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC)
2. ISO/IEC 23090-12 MPEG Immersive video, which specifies the compression of volumetric video content captured by multiple cameras; and
3. ISO/IEC 23090-10 Carriage of visual volumetric video-based coding (V3C) data.

A V3C encoder converts volumetric video frames, i.e., 3D volumetric information, into a collection of 2D images, and associated data, known as atlas data. The converted 2D images are subsequently encoded using existing video or image/video codecs, while the atlas data is encoded with mechanisms specified in ISO/IEC 23090-5.

The RTP payload format for V3C atlas component is under development in the IETF AVTCORE WG (see Internet draft <https://datatracker.ietf.org/doc/draft-ilola-avtcore-rtp-v3c/>). The draft provides information on how the association between the V3C atlas component and the V3C video components can be done on SDP level, e.g., by defining groups of RTP streams to contain V3C encoded data (RFC 5888), or by defining a way to bundle multiple RTP streams in a single transport (RFC 8843).

The authors of the RTP payload format for V3C keep a public repository of the project where the latest status of the work can be followed: <https://github.com/laurilo/draft-ilola-avtcore-rtp-v3c>.

# 8 Technical status

## 8.1 List of elements open for work

The work-in-progress elements are:

* Reference terminal architecture for EDGAR-type
* Encoding/Decoding capabilities for EDGAR-type
* Media types and formats for EDGAR-type
* AR Audio Capabilities

## 8.2 List of completed elements

The completed elements thus far are:

* None

## 8.3 List of remaining elements

The following elements listed in the work plan remains to be started:

* To be started at SA4#119
	+ Media types and formats for EDGAR-type
* To be started at SA4#121
	+ Capability exchange mechanisms to support edge provisioning
	+ Typical traffic characteristics for AR media
	+ Addition of AR Media Capabilities for 5G Media Streaming
* To be started at SA4#122
	+ Integration of 3GPP codecs in the EDGAR-type architecture
	+ Security aspects related to the media capabilities of the EDGAR-type
	+ Encapsulations into RTP, ISOBMFF and CMAF
	+ Codec-level parameter for SDP and DASH
	+ Advanced Media Capabilities for AR media
* To be started at SA4#124
	+ KPIs and simple QoE Metrics for AR media

## 8.4 List of open issues

The current open issues that are identified are:

* None

# 9 References

1. 3GPP TR 26.998, “Support of 5G Glass-type Augmented Reality / Mixed Reality (AR/MR) devices”
2. 3D Commerce Viewer Certification Program, <https://www.khronos.org/3dcommerce/certification/>
3. Khronos Group 3DC Certification documents, <https://github.com/KhronosGroup/3DC-Certification/>
4. Khronos Group 3DC Certification models, <https://github.com/KhronosGroup/3DC-Certification/tree/main/models>