Source: Samsung Electronics Co., Ltd.

**Title: FS\_5GSTAR: Generic reference device functional structures pCR**

**Agenda Item: 10.9**

**Document for: Discussion and Agreement**

# **Introduction**

This contribution follows from S4aV200541 that was agreed to be included into the permanent document in the Video SWG call on 29th September. S4aV200541 included generic device configurations based on the three device types, describing the different functions and interfaces in the configurations. In this contribution, we provide:

* Updated text to that agreed into the PD (from S4aV200541) for discussion and agreement as text to be included into the relevant clauses of the TR 26.998.

# **Changes**

\*\*\* Start change 1 \*\*\*

# **4.2 Device functional structures**

# **4.2.1 Device functions**

AR glasses contain various functions that are used to support a variety of different AR services as highlight by the different use cases in clause X.

The various functions that are essential for enabling AR glass-related services within an AR device functional structure include:

* Tracking and sensing
	+ Inside-out tracking for 6DoF user position
	+ Eye Tracking
	+ Hand Tracking
	+ Sensors
* Capturing
	+ Vision camera: capturing (in addition to tracking and sensing) of the user’s surroundings for vision related functions
	+ Media camera: capturing of scenes or objects for media data generation where required

NOTE: vision and media camera logical functions may be mapped to the same physical camera, or to separate cameras. Camera devices may also be attached to other device hardware (AR glasses or smartphone), or exist as a separate external device.

* AR/MR media processing
	+ Codecs: encoder and decoders for the encoding and decoding of media data
	+ Vision engine: engine which performs processing for AR related localisation, mapping, object detection etc., i.e. SLAM and objecting tracking
	+ Composition: displaying layers of images at different levels of depth in a display.
	+ Renderer: the generation of one (monoscopic displays) or two (stereoscopic displays) eye buffers from the media
* Tethering and network interfaces for AR/MR content delivery
	+ The AR glasses may be tethered through non-5G connectivity (wired, WiFi)
	+ The AR glasses may be tethered through different flavours for 5G connectivity
* Display: Optical see-through displays allow the user to see the real world “directly” (through a set of optical elements though). AR displays add virtual content by adding additional light on top of the light coming in from the real-world.
	+ Some good reads:
		- https://kguttag.com/2019/10/21/fov-ar-and-the-view-of-the-real-world/
		- https://www.linkedin.com/pulse/why-making-good-ar-displays-so-hard-daniel-wagner/

# **4.2.2 Generic reference device functional structure device types**

Depending on the terminal device configuration, the functions identified in clause 4.2 may exist exclusively in different physical entities, or may be duplicated between the different entities.

Generic reference device functional structures are shown below.

Device type #1



Editor’s Note:

* A better name for device type #1 will be considered
* It should be clarified that the 5G Modem is in the AR glasses
* We need different architectures as a refinement of the above architecture
	+ There are different types of connectivity
		- Uu to gNB
		- FR (sidelink) – device to device
	+ There are different glass types currently in the market, for example
		- Design glasses: AR glasses designed as regular glasses not weighing more than 70 grams (Michael Abrash, Oculus Connect 5 in 2018, the human head can comfortably carry significantly more weight than those 70g - if that weight is well distributed.)
			* As an example, in such glass types rendering can not be done in the device. Available power is very different.
		- Enterprise glasses: hololens like glasses for which design is less relevant.

Device type #2



Editor’s Note:

* A better name for device type #2 will be considered
* 5G Modem is in the smartphone – tethered connectivity is WiFi or USB-C
* Both AR glasses and smartphone should be placed to a box to represent device type #2 UE

Device type #3



Editor’s Note:

* A better name for device type #3 will be considered
* 5G Modem is in the smartphone – tethered connectivity is WiFi or USB-C. Put a box and UE.
* Both AR glasses and smartphone should be placed to a box to represent device type #3 UE
* Device type #2 and #3 are similar and the detailed differences between them are for further study.

# **4.2.3 Interfaces**

The following interfaces between the different functions of the device functional structure for device type #1 are defined:



1. Tracker/sensor output interface

Outputs from various tracking and sensor device components, namely IMUs (inertial measurement unit). These outputs are typically sent as inputs to the vision engine, which includes 6oF pose data.. The 6DoF pose data may also be sent to acloud entity when cloud-based processing is utilised.

AR/MR devices might contain the following related components:

* Accelerometer – used by the system to determine linear acceleration along the X, Y and Z axes and gravity
* Gyro – used by the system to determine rotations
* Magnetometer – used by the system to estimate absolute orientation
1. Camera output to Vision engine interface

Depending on the device hardware, different types of camera outputs may be available as inputs into the vision engine or the camera output may be sent to the network. Possible camera signals include:

* Visible light environment tracking cameras – typically gray-scale cameras used by a system for head tracking and map building
* Depth cameras (ToF)– these may be short-throw (near-depth) or long-throw (far-depth), depending on the application desired (e.g. hand tracking, or spatial mapping)
* Infrared (IR cameras) – used for eye tracking
* World facing RGB camera – used for image processing tasks and locating the camera’s position in and perspective on the scene
1. Vision engine to AR/MR renderer interface

The vision engine provides all the information required for the AR/MR renderer to adapt the rendering for a consistent combination of virtual content with the real world[Y]. This information may be the output of vision engine processes such as spatial mapping, scene understanding, and room scan visualization. Vision engine processes are typically SLAM related, differing depending on the specific device and/or platform implementation.

One typical output of the vision engine is the device/user pose information, which may include estimation properties depending on the service and application.

Editor’s note: in the case of other (tethered) device types, the split of functions between entities are FFS.

1. Vision engine to AR/MR media processor interface

The vision engine provides all the information required for the AR/MR media processor to perform processes such as 3D modelling. Information from the vision engine as used by the AR/MR renderer may also be sent to and used by the AR/MR media processor for relevant media processing.

1. AR/MR renderer to Display interface

The AR/MR rendering typically outputs a rendered 2D frame (per eye) for a given time instance according to the device/user’s current pose in his/her surrounding environment. This rendered 2D frame is sent as an input to the device display. In the future, it can be predicted that non-2D displays will require a different output from the AR/MR renderer in order to support 3D displays e.g., Light Field 3D display.

1. Camera output to Media processor interface

RGB and depth cameras are be used to capture RGB/depth images and videos, which can be consumed as regular images and videos, or may be used as inputs to a media processor for further media processing.

1. Media processor to AR/MR media codec interface

A media processor performs processes such as 3D modelling, in order to output uncompressed media data into the AR/MR media codec for encoding. One example of the media data at this interface is raw point cloud media data.

1. AR/MR media codec to AR/MR renderer interface

Compressed AR/MR media content intended for rendering and display is decoded by the AR/MR media codec, and fed into the AR/MR renderer. This is typically a decoded video bitstream (for 2D AR/MR media which may have been rendered on a remote device), or a decoded 3D media bitstream (for 3D AR/MR media).

1. AR/MR media codec to Network delivery interface

Media contents that are captured or generated by the device (from interface 5 and 6) are encoded by the AR/MR media codec before being passed onto the network delivery entity for packetization and delivery over the 5G network. For 2D AR/MR media contents, this is typically a compressed video bitstream that is conformant to the video codec used by the AR/MR media codec.

Network delivery to AR/MR media codec interface

Media contents that are received through the network delivery interface over the 5G network are depacketized by the network delivery entity and fed into the AR/MR media codec. The subsequent decoded bitstream is handled through interfaces 7 and 4.

1. Network delivery interface

Network interface for AR/MR content delivery.

\*\*\* End change 1 \*\*\*

\*\*\* Start change 2 \*\*\*

# **2 References**

[Y]ETSI GS ARF 003 V1.1.1 “Augmented Reality Framework (ARF); AR framework architecture”

\*\*\* End change 2 \*\*\*

\*\*\* End of changes \*\*\*

# **Proposal**

We propose to include the changes as presented in this contribution as a pCR for TR 26.998.