**3GPP TSG-SA WG4 Meeting #127S4-240433**

**Sophia-Antipolis, France, 29 January - 2 February 2024**

**Source: Samsung Electronics Co., Ltd.**

**Title: [FS\_AVATAR] pCR on Avatar reference architecture details**

**Spec: 3GPP TR 26.813 v0.2.1**

**Agenda item: 9.10**

**Document for: Agreement**

**1. Introduction**

Our current reference architecture in clause 7 depicts blocks of avatar data, as well as blocks of avatar functions, and the workflows between these blocks. This contribution proposes further details, including:

1. ~~For the sake of completion, adding a renderer function block and a rendered scene data block to the reference architecture.~~

* ~~Considering an end-to-end functional flow for avatar, scene data is passed onto a renderer which renders the scene according to the requirements of the user/UE’s display. Depending on this requirement, the renderer outputs data in the form of a rendered scene.~~

1. Specifying where certain blocks may exist between the UE and network, depending on the avatar service and configuration.

* The current architecture does not specify the location of each data/function block. Depending on each specific avatar service and its configuration(s), certain avatar data and avatar functions may exist either in a UE, or in a network function in the 5GS.
* For example, depending on UE capability, one sending UE may send Captured Data directly to the network where Animation Data Generation is performed by a network media function for processing, whereas another sending UE may perform Animation Data Generation and Avatar Animation processes on device, sending only Animated Avatar to the network for further processing or delivery to a receiving UE.
* Animation processes on device, sending only Animated Avatar to the network for further processing or delivery to a receiving UE.

**2. Reason for Change**

Refinement of the reference architecture.

**3. Proposal**

It is proposed to agree the following changes to 3GPP TR 26.813.

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

## 7 Reference Architecture

The following figure depicts the reference Architecture for Avatar:



Figure 11. Avatar Reference Architecture

The identified Avatar functions are:

* **Avatar Storage**: an entity that offers storage of base Avatars. This entity may be offered by the 5G System, a 3rd party entity, or the local storage of the user’s devices. The Avatar Storage ensures proper access to the base Avatar and any related data.
* **Avatar Animation:** depending on the Avatar representation format, this entity retrieves the base Avatar, receives representation format-specific animation data streams, and performs the Avatar animation to produce the animated Avatar that will be used in the rendering process.   
  [Note that some animation approaches may not need to rely on the 3D base avatar, instead they directly produce rendered 2D view of the Avatar.]
* **Scene Management**: creates and composes the shared 3D scene for all participants. It integrates a description of the user’s Avatar and updates its position and orientation based on the user’s pose. The updated scene is shared with all participants.
* **Animation data generation:** generating animation data from raw signals. The raw signals may come from cameras, microphones, and specialized motion capturing devices, etc. For example, through the current functional element, the video captured by the camera can be converted into facial feature points, and the audio captured by the microphone can be converted into text, etc.
* **Base Avatar Generation:** generates the Base Avatar from the inputs such as captured video from camera and other sensors information. Note that this might be done online or offline.

Figure 12 shows several examples of workflows where avatar functions are differently distributed between the sending UE (UE-A), the network, and the receiving UE (UE-B). The type of avatar data delivered from UE-A (highlighted by the blue arrows) is inherently different depending on which avatar functions are performed on the device by UE-A.



Figure 12. Avatar workflows showing avatar functions performed by different entities

The decision to use a certain avatar workflow depends on the requirements of the specific avatar service, as well as the configurations between the sending (UE-A) and receiving (UE-B) UEs, and the network. More specifically:

* The avatar functions supported by the sending UE (UE-A)
* The avatar functions supported by the receiving UE (UE-B)
* Negotiation of the avatar workflow configuration to request certain avatar functions to be performed by the network

\* \* \* End of Changes \* \* \* \*