**Source: HUAWEI**

**Title: [FS\_AVATAR] 2D Avatar architecture and call flow mapping**

**Spec: FS\_AVATAR PD v0.3.1**

**Agenda item: 9.8**

**Document for: Discussion and agreement**

**1. Introduction**

In last SA4#127-bis-e meeting, further clarification of 2D avatar was introduced by S4-240821 and agreed to be incorporated into the latest PD v0.3.1. During the previous discussion, it was asked to clarify the standardization content of 2D avatar. This contribution is to provide detailed information of the potential standard points regarding IMS architecture mapping and the call flow for 2D avatar.

**2. Reason for Change**

The contribution proposes to update the 2D avatar IMS architecture mapping and call flow to address the above issues.

**3. Proposal**

It is proposed to agree on the following changes to the FS\_AVATAR PD v0.3.1.

\* \* \* 1st Change \* \* \* \*

## 6.1 Potential Service Mappings



**Figure 6.1-1: Mapping Avatar Functions to IMS DC Architecture**

Figure 6.1-1 shows a mapping of avatar functions to the IMS DC architecture, specifically the possible avatar functions which may be supported by the MF.

**Avatar Storage:**

- Store and retrieve the base avatars. It may exist at the UE, DC AS, or the remote UE (subject to security constraints).

- Temporary (cache) storage of base avatars in MF for an avatar service session may also be supported, enabling the congregation and distribution of base avatars to multiple UEs in the session without the need for repeated delivery by the base avatar source. Base avatars used in recent calls may be identified by the sending UE and subsequently delivered from the DCAS and remote UEs to the MF.

**MF:**

- Base Avatar Generation: the MF generates base avatar by the user input and stores the base avatar to Avatar Storage. For 3D avatar, the base avatar can be a 3D model or an INR model. For 2D avatar, the base avatar is comprised of a DNN model and a base image/video.

- Animation Data Generation: the MF generates animation data using AI technologies based on the media received from the user.

- Avatar Animation: the MF downloads the base avatar, and animates the base avatar using the received animation data.

- Scene Management: For 3D avatar, the MF parses a scene description document to create a scene graph representation of the scene. For each node of the scene graph, it adds the associated media components for correct rendering of the corresponding object. For 2D avatar, the scene management is not needed.

**DC AS:**

- Support the subscription of avatar communication service and session control for avatar communication service.

Through such functions, the network may assist the UE with media processing related to the creation of avatar and animation data, as well the consumption of avatar data, in particular scene management/composition, and rendering.

For the support of avatar services based on the IMS DC architecture, media negotiation between the UE and network should include aspects related to:

* UE capability
* Network capability

\* \* \* End of 1st Change \* \* \* \*

\* \* \* 2nd Change \* \* \* \*

### 5.1.1 Avatar AR call

Figure 5-1 shows a generic call flow for an avatar-based call over the 5G Network. In addition to data transmission, when supported, the network provides network assistance for avatar generation, animation and retrieval. In the call flows the network entities providing avatar generation, animation and retrieval functions are generalized as a network media function, which may be for example, an RTC-AS, an MF, or an MCU. Further, an avatar storage entity is illustrated which may be an avatar repository, UE storage or a cloud storage.

The figure shows alternative flows which correspond to different possible mappings of functional blocks identified in the reference architecture. The mappings are highlighted with a call out box when applicable.



Figure 5-1 Avatar AR call flow

1. **Call Setup**
   1. A session is established between UE1, Media Function and UE2 and parameters of the session are negotiated. This may include exchanging capability information, media and metadata descriptions and formats, resource discovery etc. The involved entities agree on assignment of avatar generation, animation tasks and media requirements.
2. **Scene Description Retrieval**

The media function and the participating UEs retrieve scene descriptions, the scene description may be shared by the Media function with the UEs, or the UEs may have their own scene descriptions.

1. **Scene Description Update**

A scene update trigger occurs, e.g., if an object is added to or removed from a scene or if spatial information is updated. The update trigger may originate from the Media Function itself or the UEs. The UEs may update their scene descriptions independently or the MRF may generate an updated scene description and share it with the UEs.

NOTE: The step B and C are not needed for 2D avatar.

1. **Media and Metadata exchange**
   1. *Avatar Acquisition: In this step a base avatar is acquired by the media function.*

*Alternative #1: Network centric avatar generation*

* + 1. UE1 sends data for avatar generation to the media function. The data may be images (RGB or RGB-D), streamed to the media function as image or video stream(s). The media description of the streams may contain the camera configuration as well.
    2. The media function processes the received data to create a base avatar, for example, a rigged and/or skinned 3D model or a 2D model.

*Alternative #2: Network centric avatar loading*

* + 1. The media function loads an avatar for UE1 from an avatar storage which may be an avatar repository or storage on UE1.
  1. The media function delivers the base avatar to UE2.
  2. *Animation Data Generation*

*Alternative #1 Network centric animation data generation*

* + 1. UE1 sends source data to the media function. The source data may include images, video stream or voice stream, text. The media description of the streams may contain the camera configuration as well.
    2. The media function processes the received data to create animation data during the session. The animation data may include text, expression data and motion signals for joints.
    3. The media function delivers animation data to the UE animating the base avatar. In the diagram UE2 is shown as the recipient for clarity. The animation data may be delivered to UE1 as well.

*Alternative #2 UE centric animation data generation*

* + 1. UE1 creates animation data based on data like images, video, audio or text. The animation data may include text, expression data and motion signals for joints.
    2. UE1 delivers the animation data to the entity actuating avatar animation. The animating entity may be the media function or UE2.
  1. Avatar Animation

*Alternative #1 Network centric avatar animation*

* + 1. The media function animates the base avatar using animation data. The animation data may be generated by the media function, following step D.3.1 and D.3.2 or it may be received from UE1 following steps D.3.4 and D.3.5
    2. The media function delivers the animated avatar to the UEs. In the figure, delivery to UE1 is shown as example. The animated avatar may be delivered, for example, as 3D (e.g. video with depth and multi-view information) or 2D video.

*Alternative #2 UE centric avatar animation*

*Alternative #2a UE1 does avatar animation*

* + 1. UE1 animates the base avatar using animation data. The animation data may be generated by the media function, following steps D.3.1 and D.3.2 or it may be generated by UE1 in step D.3.4.
    2. UE1 delivers the animated avatar to UE2. The animated avatar may be delivered, for example, as 3D (e.g. video with depth and multi-view information) or 2D video.

*Alternative #2b UE2 does avatar animation*

* + 1. UE2 animates the base avatar using animation data. The animation data may be generated by the media function, following steps D.3.1 and D.3.2 and received by UE2 in step D.3.3 or it may be generated by UE1 in step D.3.4 and received by UE2 in step D.3.5.
  1. Avatar Rendering

UE2 renders the animated avatar, using for example, viewport and pose of the user. In case of 2D avatars, only decoding and display of the received 2d video may be needed.

\* \* \* End of 2nd Change \* \* \* \*