**3GPP TSG-SA WG4 Meeting #131S4-250122\_r03**

**Geneva, Switzerland, 17 - 21 February 2025**

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

**Title: [FS\_AVATAR] pCR on reference architecture and procedures**

**Agenda item: 9.7**

**Document for: Agreement**

**1. Introduction**

A potential call flow on scene management in calls with avatars exists in clause 5.1.1 of the PD v0.6.0. This contribution provides several updates on the existing procedures for clarification, and for the updated of the same to be included into the latest version of TR 26.813.

**2. Discussion**

A reference architecture for scene management in calls with avatars exists in the PD, but not in the TR.

Updates include:

* Introducing SA2 defined avatar IDs for identification of avatars, including an avatar list
* Alignment changes on use of terms such as User #1 Device, 3D scene graph, node(s) and shared scene description
* Separating call setup and scene management/update steps for clarification

**3. Proposal**

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

Change marks are made against the pre-existing text in the PD.

\* \* \* First Change \* \* \* \* (change marks on top of text in PD)

### x.x.x Scene Management in Calls with Avatars

The following call flow depicts the process for setting up and managing a common 3D scene for an AR call with Avatars from 2 or more participants.



**Figure x.x.x-1: Call flow for scene management in calls with avatars**

The steps are described as follows:

1. Prerequisite: In order to use Avatars in communication and shared experience sessions, the user needs to generate and upload their base avatar model:

* 1. The user may use local or cloud-based avatar generation tools and services to create a personalized avatar base model
  2. The user uploads its base avatar model to a central accessible storage server that will offer download of that user’s base avatar model to authorized users.

NOTE: Secure handling of base avatar models is expected but not included in this call flow.  
NOTE: Base avatar generation is not necessary if the UE already has a pre-generated base avatar model.

Avatar call setup:

2.a UE1 establishes a communication session with UE2 (only for one-to-one calls between two participants).

2.b Alternatively, UE1 establishes or joins a communication/shared space session with the AS. As part of the session establishment process, tand avatar-related capabilities (e.g., supported avatar representation and animation formats and supported animation features) that UE1 supports If UE1 is the first to join the session,a

3. The avatar list associated with the user of UE1 is requested (by UE1 or the Scene Manager) from the Avatar Storage and sent to UE1 via the Scene Manager.

4. UE1 selects an avatar to be used for the avatar call from the avatar list. Metadata related to each avatar in the avatar list (received in step 3) may also be used in the selection process.

5. UE1 offers the selected avatar to the Scene Manager as a scene update with the avatar ID corresponding to the selected avatar. This may be done by offering to the Scene Manager a scene update for inserting a new node into the shared scene description. This newly added node contains a description of how UE1’s avatar can be reconstructed and animated by other participants in the session.

6. The Scene Manager adds a new node (or set of nodes) to the shared scene description that represents UE1’s selected avatar and its related assets. It locally assigns the ownership of this node(s) to UE1, thus only allowing UE1 to update the status of these nodes. A camera node is also inserted and assigned to UE1. This node includes information about the camera used by UE1 to render the 3D scene of the AR call.

7. The participants in the session receive the shared scene description that contains the users’ avatar descriptions.

8. Based on the information in the shared scene description, each participant (e.g., UE2) identifies the required base avatar model of UE1. If a previously downloaded base avatar model is available, it is reused, otherwise the participant’s UE downloads UE1’s base avatar model in accordance with the granted level of access (i.e. which assets and at which level of detail) during that session. The access may for instance be limited to a predetermined level of detail or to a subset of the digital assets that are stored as part of the base avatar model.

9. In a direct call between only two UEs, UE2 informs UE1 about the available animation functionalities for the base avatar model. Based on this information, UE1 generates animation streams and sends those to UE2. Note that the animation streams may be relayed by a central entity, such as an IMS MF.

In-call avatar animation:

10. UE1 generates an animation data stream for the user’s avatar (e.g., based on tracking sensor data, such as a camera capturing the user’s face, or based on user interaction) and sends the stream to either: a) UE2 directly in the case of a direct call with UE2, or b) the AS over an established data channel that is part of the session.

11. Upon receiving the animation data, in a multi-party communication session, the AS may perform a processing step (e.g., transcoding or conversion) on the data via the Media Function prior to forwarding the stream to other participants (e.g., based on the participant’s supported avatar animation capabilities and features).

12. The AS distributes the animation stream of UE1 to other participants in the communication session.

13. Each participant uses the downloaded base avatar model and the received animation streams for UE1 to reconstruct and animate the avatar of UE1. The avatar is then rendered as part of the scene.

Scene management and update:

14. During the session, the participant UEs may trigger an update of the shared scene description in the Scene Manager. This can be done by offering an individual scene change to the Scene Manager, this may be in the form of a patched partial scene (e.g., JSON patch).

15. The Scene Manager receives the individual scene change offers and creates an updated version of the shared scene description, if acceptable.

16. The Scene Manager may periodically check the matching of the version of its latest shared scene description with that of the version(s) in the UEs.

17. The Scene Manager delivers the latest version of the shared scene description to corresponding UEs according to the outcome decision of step 12. Alternatively, instead of the whole shared scene description, patches (individual scene change updates) may be delivered to UEs to create the latest version of the shared scene description.

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