**3GPP TSG-SA WG4 Meeting #129-eS4-241591\_rev01**

**Online, 19 – 23 August 2024**

**Source: InterDigital Canada**

**Title: [FS\_AVATAR] pCR on Reference Architecture Mapping to Non-IMS Services**

**Spec: 3GPP TS 26.813**

**Agenda item: 9.8**

**Document for: Agreement**

**1. Introduction**

The 3GPP SA4 Video SWG has been working on a study of avatars in real-time communication services (FS\_AVATAR). The latest version of the technical report of the study TR 26.813 v0.5.0 documents a reference avatar architecture with a description of the main functions in this architecture, including: base avatar generation, animation data generation, avatar animation, scene management, and avatar storage. This architecture is shown in Figure X below for completeness.



Figure X - Avatar Reference Architecture

The objectives of the FS\_AVATAR study include the following two objectives:

* Study the integration of Avatars into the RTC services (including WebRTC and IMS).
* Document the network procedures and the impact on the 5G-RTC architecture.

This document provides a mapping of the avatar reference architecture to the General RTC Architecture in TS 26.506 for non-IMS-based RTC services.

**2. Reason for Change**

To complete the mapping of the avatar reference architecture to non-IMS-based 5G media services.

**4. Proposal**

It is proposed to agree the following changes and to integrate them into 3GPP TS 26.813.

\* \* \* Begin Changes \* \* \* \*

## 8.7 Mapping to non-IMS-based Services

## 8.7.1 Architecture Mapping

The architecture shown in Figure 12 demonstrates a mapping of the avatar reference architecture described in clause 7 to the General RTC Architecture defined in TS 26.506 and based on collaboration scenario 3 of that specification.



**Figure 12 - RTC Architecture for Avatar-based Communication**

In this architecture an application for avatar-based communication interacts with an RTC Client to establish WebRTC-based RTC sessions with a Media AS that supports avatar functions. Note that while the Avatar Storage function is not shown in Figure 12, this function can be part of the UE or the RTC AS. The avatar functions supported by the Media Function include:

* Base Avatar Generation: the MF may generate the base avatar from captured sensor data on the participant’s UE and made available to the MF.
* Animation Data Generation: the MF can generate an animation data stream based on the media received from the user that can be applied to the base avatar to animate it.
* Avatar Animation: the MF can apply an animation stream that is either generated locally in the MF itself or received from the UE to a base avatar to animate it.
* Scene Management: the MF may create and maintain a scene graph representation of the scene in which the avatars and other media objects may be placed.

\* \* \* Next Change \* \* \* \*

## 8.7.2 Call flows

## 8.7.2.1 Call setup and capability negotiation

Figure 13 demonstrates a call flow diagram for call setup and capability negotiation between the avatar application running on UE1, the MF, and a remote endpoint (UE2).



**Figure 13 - Call flow for call setup and capability negotiation**

The steps shown in Figure 13 are as follows:

* 1: Service provisioning and announcement of an RTC service on the network side, in particular between the Media AF (application function) and the Media AP (application provider).
* 2: The RTC MSH requests configuration information from the RTC AF relating to RTC sessions.
* 3: The RTC AF provides the requested configuration information to the RTC MSH.
* 4: The application queries the RTC MSH through the RTC Access Function for the WebRTC Signalling Function information and retrieves this information. In some cases, where the signalling is handled by the RTC Access Function, the RTC Access Function directly queries the RTC MSH for this information.
* 5, 6: Two or more RTC endpoints exchange signalling information related to the RTC session, such as ICE candidates and SDP offer/answer through the SWAP protocol using the trusted WebRTC Signalling Function in the RTC AS. The SDP offer/answer messages may also be used to negotiate avatar-related capabilities such as an avatar representation format or an animation data format as well as support for data channels.

## 8.7.2.2 Avatar Media and Metadata Exchange

Figure 14 demonstrates a call flow diagram for the exchange of avatar media and associated metadata between relevant entities in the architecture mapping presented in clause 8.7.1 after call setup has been complete and an RTC session has been established between the MF, the RTC endpoint on UE1, and the remote endpoint (UE2).



**Figure 14 - Call flow for Avatar Media and Metadata Exchange**

The description of the steps shown in Figure 5 is as follows:

**B. Scene Description Retrieval**

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

**C. 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 MF itself or the UEs. The UEs may update their scene descriptions independently or the MF may generate an updated scene description and share it with the UEs.

**D.1 Avatar Acquisition**

* D.1a.1: The RTC application on UE1 sends captured data needed to generate the base avatar to the Media Function in the RTC AS through one or more RTP streams over the RTC session.
* D.1a.2: The MF uses the captured data sent by UE1 to generate the base avatar for the user.

Note: When the base avatar is generated in the network by the MF, the generated base avatar may be stored in the Avatar Storage for future loading.

* D.1b.1: The MF loads the base avatar for UE1, identified by the negotiation step, from the Avatar Storage.

**D.2 Avatar Delivery**

* D.2.1: The MF creates a reliable Data Channel for the delivery of the base avatar to UE2.
* D.2.2: The MF creates a reliable Data Channel for the delivery of the base avatar to UE1.

**D.3 Animation Data Generation**

* D.3a.1: UE1 sends source animation data (e.g., sensor data, audio, video, text) to the MF through media or data channels.
* D.3a.2: The MF processes the received source data to generate an animation data stream for the session.
* D.3a.3: The MF delivers the generated animation data through a media or data channel to UE2.

Note: In network-centric avatar animation data generation, the animation data generated by the MF may also be delivered to UE1.

* D.3b.1: UE1 uses data captured by its sensors to generate an animation data stream.
* D.3b.2: UE1 sends the generated animation data stream to UE2 through the MF over a media or data channel.

**D.4 Avatar Animation**

* D.4a.1: The MF uses the animation data (generated by the MF itself in step D.3a.2 or received from UE1 in step D.3b.2) to animate the base avatar.
* D.4a.2: The MF sends an animated avatar stream to UE1 and UE2.
* D.4b.1: UE1 animates the base avatar using the animation data stream generated in step D.3b.1.
* D.4b.2: UE1 sends an animated avatar stream directly to UE2 through a media channel.

**D.5 Avatar Rendering and Display**

* UE2 renders the animated avatar based on the target viewport and pose.

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