**Source: Meta Ireland (Rapporteur)**

**Title: Functional Components for iRTC Client in the Terminal**

**Agenda Item: 10.5**

**Document for: Discussion and Agreement**

1. **Introduction**

A key figure whose role is similar to figure 4.2 of MTSI is proposed to illustrate the media, data, and signal flow of an iRTC client in terminal.



Figure 4.2: Functional components of a terminal including an MTSI client in terminal using 3GPP access

Notable differences of iRTC’s functional components from those of MTSI include:

* Text telephony based on [1] is not supported (UI enables text input)
* Media pre/post-processors are present but are not within the scope of standards
* Sensor output is input to media post-processors (e.g., for spatial processing) or sent
* Device information is input to media pre-processors (e.g., for describing microphones or converting RGBD to point cloud [2])
* There are paths (e.g., for metadata [3] or animation codes for avatars) from media encoder to data channel, and from data channel to media decoder)

1. **Proposal**

It is requested to review the following changes in TS 26.113 and revise them if necessary. A Visio document including both figures are attached for facilitating revisions.

## 4.2 iRTC client in terminal

The functional components of a terminal including an iRTC client using 3GPP access are shown in figure 4.2.x. The scope of the present document is to specify the handling and interaction of media and data, which includes their capture and generation, pre/post-processing, and compression. Transport of media and data consists of the encapsulation of the coded media and data in one or more transport protocols, which is shown in figure 4.2.x as the "packet-based network interface" and is illustrated in more detail in the user-plane and control-plane protocol stacks shown in figure 5.5.x.



Figure 4.2.x: Functional components of a terminal including an iRTC client in terminal using 3GPP access

The 3GPP Layer 2 protocol to be interfaced with iRTC client is PDCP [x14] for EPC and SDAP [x15] for 5GC, which is used on top of PDCP as shown in clause 4.4.1 of [x16]. It is assumed that the SDAP would be configured without header for both directions in the typical iRTC cases, effectively interfacing with PDCP, as SDAP header would be needed only when more than one QoS flows are multiplexed in a DRB or reflective mapping is enabled. An architecture for XR baseline client can be found in [x4].

NOTE 1: Functional components in the grey box, except audio and video pre/post-processors, are within the scope of present document, which also specifies output formats of sensor, microphone, and camera.

NOTE 2: In certain codec types, e.g., avatars, at least the decoders need to be personalized with the information of those who will participate in the communication before or during session setup.

NOTE 3: Device information is assumed to be stored in the UE and loaded to the iRTC client during session setup.

NOTE 4: The iRTC client may exchange media and data with external devices tethered over wired links such as USB-C, 3GPP PC5 [x17], or non-3GPP radio access technologies such as Wi-Fi or Bluetooth.

NOTE 5: Text can be entered via user interface, typically available on display.

1. **References**

[1] ITU-T Recommendation [T.140](https://www.itu.int/rec/T-REC-T.140/en), Protocol for multimedia application text conversation

[2] [S4aR220014](https://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/3GPP_SA4_AHOC_MTGs/SA4_RTC/Docs/S4aR220014.zip), 3D video capture description for iRTC client in the terminal

[3] [S4aR220053](https://urldefense.com/v3/__https:/www.3gpp.org/ftp/TSG_SA/WG4_CODEC/3GPP_SA4_AHOC_MTGs/SA4_RTC/Docs/S4aR220053.zip__;!!Bt8RZUm9aw!9G964HWg3FfZTdT6OwRuGZr5KXZgK8vXvMCR4qAq1j1tgOnyv2oKfdG8XETuIH5bAmGt-2BNUQarHjv4PA$), Real-time metadata transport over data channel