Source: Ericsson LM

**Title: pCR AR two-party conversational services**

**Agenda Item: 11.10 (FS\_5GSTAR)**

**Document for: Agreement**

# **Introduction**

3GPP TR 26.998 V0.9.0 (S4aV210731) clause 6.5 describes AR Conversational services for communication between two or more parties.

Annex 4 describes two use-cases for AR Conferencing to support 1:1 AR Conferencing and 1:many AR Conferencing. The two use-cases can further include:

1. 1-1 AR Conferencing, symmetrical use-case, where both parties are sending and receiving AR immersive media.
2. 1-1 or 1-many AR Conferencing, asymmetrical use-case, where one party is sending AR immersive media and the backchannel from other participants can be audio only, 2D video, etc.
3. 1-many or multi-party AR Conferencing, symmetrical use-case, where more than two parties are sending and receiving AR immersive media.

The asymmetrical vs. symmetrical differentiation is further highlighted in Figure 1. Some examples of asymmetrical cases include a product presentation by salesman to customers, remote medical consultation, grandfather watching grandkids. In all cases, a single party (salesman, doctor, grandkids) is filmed by a 3D camera and the stream is transmitted to the AR glasses of receiving participants. Some examples of symmetrical cases include team meetings, workshops, etc.



**Figure 1: Asymmetrical vs symmetrical AR conferencing**

The contribution fills the gap with respect to identified AR Conferencing use-cases for two-party AR calls in 3GPP TR 26.998.

# **Proposed change**

6.5 AR conversational services

6.5.1 Introduction

AR Conversational services are end-to-end use-cases that include communication between two or more parties. The following building blocks that can be used to realize AR conversational services can be identified as:

a) Call setup and control: this building block covers the

- signalling to setup a call or a conference.

- fetching of the entry point for the AR experience. The protocol shall support upgrading and downgrading to/from an AR experience. It shall also support adding and removing media. This also includes the device type (Type-1, Type-2, or Type-3) as well as non-AR experiences, e.g. tablet.

b) Formats: The media and metadata types and formats for AR calls should be identified. The format for the entry point, namely the scene description, and any extensions to support AR telephony need to be identified. Also, the format for media capturing, e.g., point clouds, colour attributes, etc. need to be identified. For AR telephony media types, the necessary and QoS characteristics need to be defined, as well as format properties and codecs.

c) Delivery: the transport protocols for the AR media need to be identified. AR telephony and conferencing applications require low latency exchange of real-time media. A protocol stack, e.g. based on RTP, will be required.

d) 5G system integration: offering the appropriate support by the 5G system to AR telephony and conferencing applications includes:

- signalling for QoS allocation,

- discovery and setup of edge resources to process media for AR telephony,

- usage of MBS,

- data collection and reporting.

The building blocks may have different instantiations and/or options. For example, the delivery may be mapped to a WebRTC protocol stack or to an MTSI protocol stack. Furthermore, a single session may combine several delivery methods to accommodate the different media types supported by an AR conversational service.

In addition, AR telephony and conferencing applications can support asymmetrical and symmetrical experiences. In an asymmetrical case, one party is sending AR immersive media and the backchannel from other participants can be audio only, 2D video, etc. In a symmetrical case, all involved parties are sending and receiving AR immersive media.

6.5.2 Basic architecture and call flows

There are different options for mapping to 5G system:

a) The MTSI architecture (TS 26.114) supports audio and 2D video conversational services. Extending the MTSI architecture to support AR signalling and immersive media. This includes both MTSI/RTP and MTSI/Data channel (DC) stack options.

b) Extending the 5GMS architecture (TS 26.501) to support AR conversational services by combining live uplink and live downlink. 5GMS offers basic functionality such as QoS support, reporting, and in the future also edge, which will be beneficial for all types of applications. The typical/expected QoS parameters (especially delay) need to be clarified.

c) An architecture based on something different than MTSI / IMS or 5GMS, for example, browser implementations such as WebRTC. WebRTC is widely deployed today for conversational services and is built on flexible ecosystem on the device side, which is important in this case since conversational AR will require significant device-side changes.

**Table 6.5.2-1: Comparison of different architecture options for supporting AR conversational services**

|  |  |  |  |
| --- | --- | --- | --- |
| **Component** | **MTSI: RTP and DC (TS 26.114)** | **5GMS/HTTP (26.501)** | **Other architecture (e.g. WebRTC-based)** |
| Protocol | SIP- and RTP-based or DC-based. SDP signalling and formats for AR are missing and need to be defined. Encoding and decoding at MTSI client needs to be extended beyond ITT4RT with improved support of AR immersive media formats (e.g. meshes, point clouds). | TCP- and HTTP-based streaming, using DASH/HLS and MPEG OMAF/CMAF technology. AR & immersive media content and signalling are assumed to work with HTTP-based streaming in the other use-case mappings. | For example, using non-IMS WebRTC data channel and/or extending WebRTC audio/video for AR media such as immersive media communications. AR signalling aspects to be studied.  |
| Connection establishment | Find and connect is solved through SIP and E.164 addressing in IMS. | Find-and-connect for the conversational, UE-to-UE, case is undefined. | WebRTC implementations offer dedicated APIs for connection establishment in various contexts such as social media platforms. Browser applications are widely available. |
| Performance | Technically possible, latency should in principle not be a problem to achieve, building on the existing QoS and policy framework in 5GC. | Low latency and QoS DASH support in 5GMS to be studied. | WebRTC is designed with low latency in mind but has no defined relation to QoS and policy framework in 5GC and use of that need to be studied.  |
| Deployments | Cross-operator interconnect aspects are included. Edge processing functions to be studied. | Cross-operator interconnect aspects are currently ignored. Edge processing functions to be studied (e.g. EMSA). | Cross-operator interconnect aspects are currently not applicable since WebRTC is used OTT today, but will become relevant and need study, especially if used with QoS. Edge processing functions to be studied. |
| Legal Intercept(only in scope of SA3, not SA4) | LI framework exist. Possible extensions to cover new AR media formats to be studied in SA3. | Not in scope since it is not a telephony service. | Not in scope if only OTT. |

NOTE: There is no support of WebRTC media stack in 3GPP today, except for the WebRTC data channel stack in MTSI. WebRTC access to IMS was studied in TR 23.701 and TR 23.706, and OTT WebRTC client access to 3GPP core network through a gateway is specified in TS 24.371.

1. Editor’s note: It is FFS whether the study of the other architecture is limited to OTT, where the subsequent scope would be to deal with interworking through a gateway, or a completely new thing with more native WebRTC support.

To describe the functional architecture for AR conversational use-cases such as clause Annex A.4 and identify the content delivery protocols and performance indicators an end-to-end architecture is addressed. The end-to-end workflow for AR conferencing (one direction) is shown in Figure 6.5.2-1. Camera is capturing the participant in an AR conferencing scenario. The camera is connected to a UE (e.g. laptop) via a data network (wired/wireless). Live camera feed, sensors and audio signals are provided to a UE/Edge node (or split) which processes, encodes, and transmits immersive media content to the 5G system for distribution. The immersive media processing function can include pre-processing of the captured 3D video, format conversion, and any other processing needed before compression. Immersive media content includes 3D representation, such as in form of meshes or point clouds, of participants in an AR conferencing scenario. After processing and encoding, the compressed 3D video and audio streams are transmitted over the 5G system. A 5G STAR UE decodes, processes and renders the 3D video and audio stream.

The use-case can be extended to bi-directional/symmetric case by adding a 3D camera on the receiver side and AR glasses on the sender side and applying a similar workflow. For the case of EDGAR UE, the immersive media is further pre-rendered in the 5G system and transmitted to the UE. In addition, further media processing such as main scene management and composition are processed in cloud/edge.

**Figure 6.5.2-1: Extensions to device architecture of conversational services for STAR UE**

6.5.2.1 Symmetrical case

We consider an immersive AR two party call between Alice and Bob. The end-to-end call flow is described:

1. [STAR UE Alice - STAR UE Bob]: Either one of the UEs can initiate an AR immersive call by starting an application on the phone or AR glasses.

2. [STAR UE Alice - STAR UE Bob]: Both UEs communicate with a signalling server to establish the AR call. During the session establishment, both parties agree on the format (e.g. point clouds, triangular/polygon meshes). The exact session type and configuration depends on the capabilities of STAR UE.

3. [STAR UE Alice]: Alice is captured by a depth camera embedded within the STAR UE or an external camera which generates an immersive 3D media stream (audio and video).

4. [STAR UE Alice]: The immersive 3D media is encoded and transmitted in real-time to Bob over the 5G system. Additional pre-processing can be applied before encoding such as format conversion.

5. [STAR UE Bob]: The immersive 3D media is received on Bob’s STAR UE. The immersive 3D media stream is decoded and rendered on AR glasses. Additional postprocessing can be applied before rendering such as format conversion, customization to match the stream to rendered environment e.g. filling holes.

6. [STAR UE Bob]: Bob is captured by a depth camera generating an immersive 3D media which is encoded and transmitted in real-time to Alice’s AR glasses.

7. [STAR UE Alice]: The immersive 3D media which is received, decoded and rendered on Alice’s AR glasses.

8. [STAR UE Alice - STAR UE Bob]: Both UEs terminate the service at the end of the call.

6.5.2.2 Asymmetrical case

We consider an immersive AR asymmetrical call between Alice and Bob, where Bob is transmitting immersive media to be consumed on the AR glasses of Alice (STAR UE). Bob (non-STAR UE) is receiving content from Alice via other means such as audio, 2D video, etc. The end-to-end call flow is described:

1. [STAR UE Alice – non-STAR UE Bob]: Alice initiates an AR immersive call by starting an application on the phone or AR glasses.

2. [STAR UE Alice - non-STAR UE Bob]: Alice communicates with a signalling server to establish the AR call. During the session establishment, the format is identified (e.g., point clouds, triangular/polygon meshes). The exact session type and configuration depends on the capabilities of STAR UE.

3. [non-STAR UE Bob]: Bob is captured by a depth camera embedded within the STAR UE or an external camera which generates an immersive 3D media stream (audio and video).

4. [non-STAR UE Bob]: The immersive 3D media is encoded and transmitted in real-time to Alice over the 5G system. Additional pre-processing can be applied before encoding such as format conversion.

5. [STAR UE Alice]: The immersive 3D media is received on Alice’s STAR UE. The immersive 3D media stream is decoded and rendered on AR glasses. Additional postprocessing can be applied before rendering such as format conversion, customization to match the stream to rendered environment e.g., filling holes.

6. [STAR UE Alice]: Alice is transmitting audio, 2D video or other media content as a back channel to Bob.

7.  [non-STAR UE Bob]: The 2D video or other media content which is received, decoded and rendered on Bob’s device.

8. [STAR UE Alice – non-STAR UE Bob]: Alice terminates the service at the end of the call.

NOTE: Additional call-flows that cover other AR conferencing use-cases listed in Table 6.1-1 can be added.

6.5.3 Instantiation #1: MTSI-based architecture extension

This instantiation provides the detailed architecture and procedures for the case of extending the current MTSI architecture. Figure 6.5.3-1 provides an MTSI-based architecture of conversational services for STAR UE.

An MTSI client specified in TS 26.114 [4.3.d] can be extended to an AR-MTSI client which supports AR immersive media and take a role of Media Access Functions. A data channel application, an HTML web page including JaveScript(s) provided by a data channel server through a bootstrap data channel, also can be used to provide rich user experiences such as sitting side by side on a bench. Support of data channel media is optional for an MTSI client. An AR-MTSI client supporting data channel is denoted as an AR-DCMTSI client. Note that the data channel server can be implemented in IMS core or outside of it.

**Figure 6.5.3-1: MTSI-based conversational service architecture for STAR UE**

Figure 6.5.3-2 illustrates the procedure diagram for an immersive AR two party call using STAR UEs including an AR-MTSI client.



**Figure 6.5.3-2: AR-MTSI client to AR-MTSI client call establishment (STAR UE)**

Assumptions:

- AR immersive media can be sent over RTP/UDP/IP.

- AR immersive media format (e.g. point clouds, triangular/polygon meshes) can be negotiated and configured using SDP.

Procedures:

1. A STAR UE initiates a SIP INVITE request, containing the SDP offer with AR media capabilities.
2. The call propagates to the terminating STAR UE.
3. The called party’s STAR UE returns an SDP answer in a SIP 183 progress message. The P-CSCF uses the SDP answer to allocate the required resources.
4. The originating STAR UE generate a PRACK which is transited to the terminating side of the call.
5. The originating STAR UE receives an associated 200 OK (PRACK).
6. The STAR UE reserves internal resources to reflect the SDP answer and configures mediapipelines.
7. The STAR UE sends a SIP UPDATE message with a new SDP offer confirming the selected media parameters.
8. The 200 OK (UPDATE) response is received for the terminating STAR UE containing the SDP answer.
9. The terminating STAR UE is now alerted and sends a SIP 180 Ringing response.
10. When the called party’s STAR UE has answered the call, it sends a 200 OK to the calling party STAR UE.
11. The STAR UE receives the 200 OK, and sends a SIP ACK message to acknowledge that the call has been established.
12. The STAR UE processes the immersive media to be transmitted.
13. The AR runtime function captures and processes the immersive media to be sent.
14. The AR runtime function passes the immersive media data to the AR-MTSI client.
15. The AR-MTSI client encodes the immersive media to be sent to the called party’s STAR UE.

NOTE: The capturing can be done by an external camera. In that case, the processing and encoding can be done outside STAR UE (i.e. AR-MTSI client)

1. The STAR UE has an AR call established with AR media traffic.
2. The STAR UE processes the received immersive media.
3. The AR-MTSI client decodes and process the received immersive media.
4. The AR-MTSI client passes the immersive media data to the Scene Manager.
5. The Scene Manager renders the immersive media, which includes the registration of the AR content into the real world accordingly.

Figure 6.5.3-3 illustrates the procedure diagram for an immersive AR two party call using STAR UEs including an AR-DCMTSI client.



**Figure 6.5.3-3: AR-DCMTSI client to AR-DCMTSI client call establishment (STAR UE)**

Assumptions:

- AR immersive media can be sent over RTP/UDP/IP.

- AR immersive media can be negotiated and configured using SDP.

- A data channel application can provide rich user experiences by utilizing both user’s underlying scene and pose of objects representing users in the scene.

Procedures:

1-14: Same as the procedures for AR-MTSI client to AR-MTSI client call establishment except that the SDP contains a data channel media description for the bootstrap data channel.

1. The STAR UE retrieve a data channel application through the bootstrap data channel.
2. Any additional data channels created and used by the data channel application itself are requested.
3. The AR-DCMTSI client initiate SIP re-INVITE request, containing an updated SDP offer to establish those data channels.
4. The data channels for the data channel application have been established.
5. The established data channel can be used by the data channel application JaveScript(s).

6.5.4 Instantiation #2: xxx

6.5.x Summary of AR conversational instantiations

Table 6.5.x-1 shows the list of potential instantiations and how they can be composed from each building block described in clause 6.5.1.

**Table 6.5.x-1: Summary of each instantiation for AR conversational services**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Building Block** | **Instantiation#1:****MTSI extension** | **Instantiation#2:** | **Instantiation#3:** | **Instantiation#4:** |
| Call setup and control | Conventional MTSI |  |  |  |
| Formats | TBD |  |  |  |
| Delivery | RTP/UDP/IP, SCTP/DTLS/UDP/IP |  |  |  |
| 5G system integration | Need policy exchange for AR-(DC)MTSI client (P-CSCF and PCF) |  |  |  |

# **Conclusion**

It is proposed to include Section 2 in the TR.