|  |
| --- |
| 3GPP TS 26.264 V1.1.0 (2024-04) |
| Technical Specification |
| 3rd Generation Partnership Project;Technical Specification Group Services and System Aspects;IMS-based AR Real-Time Communication (Release 18) |
|  |
|  |  |
|  |
| The present document has been developed within the 3rd Generation Partnership Project (3GPP TM) and may be further elaborated for the purposes of 3GPP.The present document has not been subject to any approval process by the 3GPPOrganizational Partners and shall not be implemented.This Specification is provided for future development work within 3GPPonly. The Organizational Partners accept no liability for any use of this Specification.Specifications and Reports for implementation of the 3GPP TM system should be obtained via the 3GPP Organizational Partners' Publications Offices. |

|  |
| --- |
|  |
| ***3GPP***Postal address3GPP support office address650 Route des Lucioles - Sophia AntipolisValbonne - FRANCETel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16Internethttps://www.3gpp.org |
| ***Copyright Notification***No part may be reproduced except as authorized by written permission.The copyright and the foregoing restriction extend to reproduction in all media.© 2024, 3GPP Organizational Partners (ARIB, ATIS, CCSA, ETSI, TSDSI, TTA, TTC).All rights reserved.UMTS™ is a Trade Mark of ETSI registered for the benefit of its members3GPP™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational PartnersLTE™ is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational PartnersGSM® and the GSM logo are registered and owned by the GSM Association |

Contents

Foreword 4

1 Scope 6

2 References 6

3 Definitions of terms, symbols and abbreviations 6

3.1 Terms 6

3.2 Symbols 7

3.3 Abbreviations 7

4 System description 7

4.1 General 7

4.2 Terminal architecture 8

4.3 End-to-End Reference Architecture 9

5 Immersive AR Media 11

5.1 General 11

5.2 Speech 11

5.3 Video 11

5.4 Real-time text 11

5.5 Still images 11

6 AR Metadata 11

6.1 General 11

6.2 Metadata data channel message format 12

6.3 Spatial descriptions 13

6.3.1 Spatial description format 13

6.3.1.1 General 13

6.3.1.2 Available visualization space and user position 13

6.4 Scene descriptions 13

6.5.1 General 14

6.5.2 Pose Format 14

6.5.3 Action Format 14

7 Media configurations 14

8 AR Data Transport 14

8.1 General 14

8.2 RTP transport 15

8.3 RTCP usage 15

8.4 Data channel usage 15

9 Quality of Experience 15

Annex A (normative): Call flows for IBACS 16

A.1 IMS AR Call Flow 16

A.1.1 General 16

A.1.2 AR Call Session Setup 16

A.1.3 Split Rendering Negotiation 16

A.1.4 Scene Description Processing 18

A.1.5 AR Media Processing 19

Annex <X> (informative): Change history 21

# Foreword

This Technical Specification has been produced by the 3rd Generation Partnership Project (3GPP).

The contents of the present document are subject to continuing work within the TSG and may change following formal TSG approval. Should the TSG modify the contents of the present document, it will be re-released by the TSG with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

x the first digit:

1 presented to TSG for information;

2 presented to TSG for approval;

3 or greater indicates TSG approved document under change control.

y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.

z the third digit is incremented when editorial only changes have been incorporated in the document.

In the present document, modal verbs have the following meanings:

**shall** indicates a mandatory requirement to do something

**shall not** indicates an interdiction (prohibition) to do something

The constructions "shall" and "shall not" are confined to the context of normative provisions, and do not appear in Technical Reports.

The constructions "must" and "must not" are not used as substitutes for "shall" and "shall not". Their use is avoided insofar as possible, and they are not used in a normative context except in a direct citation from an external, referenced, non-3GPP document, or so as to maintain continuity of style when extending or modifying the provisions of such a referenced document.

**should** indicates a recommendation to do something

**should not** indicates a recommendation not to do something

**may** indicates permission to do something

**need not** indicates permission not to do something

The construction "may not" is ambiguous and is not used in normative elements. The unambiguous constructions "might not" or "shall not" are used instead, depending upon the meaning intended.

**can** indicates that something is possible

**cannot** indicates that something is impossible

The constructions "can" and "cannot" are not substitutes for "may" and "need not".

**will** indicates that something is certain or expected to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**will not** indicates that something is certain or expected not to happen as a result of action taken by an agency the behaviour of which is outside the scope of the present document

**might** indicates a likelihood that something will happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

**might not** indicates a likelihood that something will not happen as a result of action taken by some agency the behaviour of which is outside the scope of the present document

In addition:

**is** (or any other verb in the indicative mood) indicates a statement of fact

**is not** (or any other negative verb in the indicative mood) indicates a statement of fact

The constructions "is" and "is not" do not indicate requirements.

# 1 Scope

This clause shall start on a new page.

The present document focuses on IMS-based conversational AR (Augmented reality) services. AR services can overlay media (e.g., video, audio, text, etc.) on top of the user’s real perception. Conversational AR services as described by the present document typically include a bidirectional conversational A/V connection in addition to other non-real-time AR media for collaboration or communication between two or more users.

# 2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non‑specific.

- For a specific reference, subsequent revisions do not apply.

- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[2] 3GPP TS 26.114: "IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction".

[3] 3GPP TS 26.119: "Media Capabilities for Augmented Reality".

[4] 3GPP TS 23.228: "IP Multimedia Subsystem (IMS); Stage 2".

[5] 3GPP TS 24.229: "IP multimedia call control protocol based on Session Initiation Protocol (SIP) and Session Description Protocol (SDP); Stage 3".

[6] 3GPP TS 26.565: "Split Rendering Media Service Enabler".

[X] Khronos glTF 2.0, [glTF™ 2.0 Specification (khronos.org)](https://registry.khronos.org/glTF/specs/2.0/glTF-2.0.html)

[X2] ISO/IEC 23090-14 AMD 2, Information technology — Coded representation of immersive media — Part 14: Scene description — Amendment 2: Support for haptics, augmented reality, avatars, Interactivity, MPEG-I audio, and lighting

…

[x] <doctype> <#>[ ([up to and including]{yyyy[-mm]|V<a[.b[.c]]>}[onwards])]: "<Title>".

It is preferred that the reference to TR 21.905 be the first in the list.

# 3 Definitions of terms, symbols and abbreviations

This clause and its three (sub) clauses are mandatory. The contents shall be shown as "void" if the TS/TR does not define any terms, symbols, or abbreviations.

## 3.1 Terms

For the purposes of the present document, the terms given in TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in TR 21.905 [1].

Definition format (Normal)

**<defined term>:** <definition>.

**AR data:** Collection of information to be exchanged among participants in a call with AR experience. It includes AR media and AR metadata.

**AR media:** Media (e.g., audio, video, text or image) that will be rendered by the AR-MTSI client as an overlay over the user’s real perception. This includes traditional 2D media (e.g., a 2D audio stream rendered to be perceived by the user to originate from their left side) and 3D media (e.g., spatial audio and volumetric video).

**AR metadata:** Data that provides information on AR media and its rendering. This includes pose, spatial descriptions and scene descriptions.

**AR-MTSI client:** A DCMTSI client supporting AR capabilities as defined by this specification.

**AR MRF:** An AR-MTSI client implemented by functionality included in the MRFC and the MRFP.

**AR-MTSI client in terminal:** An AR-MTSI client that is implemented in a terminal or UE. The term "AR-MTSI client in terminal" is used in this document when entities such as AR MRF is excluded.

**Split rendering**: The procedure in which a UE offloads some of the media processing related to rendering tasks to a media function as considered for network centric AR IMS session procedures in TS 23.228 [4]

## 3.2 Symbols

For the purposes of the present document, the following symbols apply:

Symbol format (EW)

<symbol> <Explanation>

## 3.3 Abbreviations

For the purposes of the present document, the abbreviations given in TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in TR 21.905 [1].

ADC Application Data Channel

BDC Bootstrap Data Channel

AS Application Server

DC Data Channel

DCSF Data Channel Signalling Function

I‑CSCF Interrogating‑CSCF

IMS IP Multimedia Core Network Subsystem

MF Media Function

MRF Multimedia Resource Function

P‑CSCF Proxy‑CSCF

S‑CSCF Serving‑CSCF

# 4 System description

## 4.1 General

Typical conversational AR scenarios as envisioned in this document consist of an immersive AR call that may include the following conversational components:

- Real-time speech/audio that can comprise mono, stereo, and/or spatial audio.

- Real-time 2D video or 360-degree video that can be rendered as rectangular or spherical overlay in the AR experience.

- A real-time volumetric video of the user or an object that can be rendered in AR or MR.

In addition to the above conversational media, non-real-time objects may be exchanged over the data channel as well.

Both two-party and multiparty calls are possible. The AR experience may be unidirectional, i.e., only one party receives AR media and renders it, or it may be bidirectional, i.e., both parties receive and transmit AR media. The term AR-MTSI client includes both:

- an AR-MTSI client in terminal which is an AR device as defined in [3] e.g., AR glasses, phone, Head Mounted Display (HMD) that has an XR Runtime for rendering an AR experience.

- AR MRF that provides support for AR conversational services.

As an AR-MTSI client in terminal is a DCMTSI client in terminal with additional features for AR communication, the following requirements for a MTSI client terminal also apply for an AR-MTSI client in terminal:

- the interworking requirements in clause 12 of TS 26.114 [2],

- the jitter buffer management requirements in clause 8 of TS 26.114 [2],

- the packet loss handling requirements in clause 9 of TS 26.114 [2],

- the media and rate adaptation requirements in clause 10 and 17 of TS 26.114 [2], and

- the network preference management object in clause 15 of TS 26.114 [2],

NOTE: If an AR-MTSI client in terminal supports functionalities for MSMTSI client in terminal as specified in Annex S of TS 26.114 [2], the media and rate adaptation requirements in Annex S.8 of TS 26.114 [2] also apply for an AR-MTSI client in terminal.

## 4.2 Terminal architecture

The detailed XR client architecture is not in the scope of this specification. The XR baseline client architecture can be found in [3]. The pre/post-processor component in terminal provides AR capabilities for processing output of peripherals and the input/output of encoders/decoders, which may include:

- XR runtime

- Scene manager

- Presentation engine

- XR source management

The AR-MTSI client has XR Runtime capabilities for rendering AR experience, e.g., spatial localization and mapping, etc., and can support local AR rendering and network-assisted split-rendering based on client’s capabilities. A UE may support multiple microphones, cameras or sensors.

An AR-MTSI client supports the protocol stack of a basic MTSI client as described in clause 4.2 of [2]. For the specific AR communication instance, AR-MTSI client can select different IMS media channel to delivery AR data to IMS network or peer UE. In general, AR media components with real-time characteristics are transported via RTP session and AR metadata is transported via data channel or RTP session with AR media.



Figure 4.2.1: Functional components of an AR-MTSI client in terminal

## 4.3 End-to-End Reference Architecture

The end-to-end architecture to support AR communication over IMS can be found in TS 23.228 Annex AC [4]. The following Figure 4.3.1 is a simplified version showing the media functions within the scope of this specification.



Figure 4.3.1: Generalized IMS DC Architecture to support AR communication

NOTE 1: General control-related elements over Gm interface, such as SIP signalling (TS 24.229 [5]), fall outside the scope of this specification, albeit parts of the session setup handling and session control for AR conversational media over Gm interface, such as the usage of SDP and setup and control of the individual media streams between clients, are defined in this specification.

NOTE 2: DC Application Repository may be in external DN but can also be in operator domain. The DC Application Repository holds the application(s) that can be used in AR communication sessions and is out of scope of 3GPP.

AR Application Server (AR AS):

- AR Application Server is responsible for AR service control related to AR communication, including AR session media control and AR media capability negotiation with the UE.

NOTE 3: AR Application Server is a specific DC Application Server and is out of scope of 3GPP.

NOTE 4: The UE can download the AR metadata from AR AS through application data channel.

DCSF:

- The DCSF receives event reports from the IMS AS, and decides whether AR communication service is allowed to be provided during the IMS session. Additionally, the DCSF interacts with the AR AS for DC resource control.

MF/MRF:

- Support AR conversational service by providing transcoding for terminals with limited capabilities. Additionally, the MF/MRF may collect spatial and media descriptions from UEs and create scene descriptions for symmetrical AR call experiences.

- Provide remote rendering for AR-MTSI clients in terminals with limited capabilities based on rendering negotiation. For remote rendering the AR-MTSI client provides AR metadata, e.g., pose data, as defined in clause 6 of this specification.

IMS AS:

- The IMS AS receives the media control instructions from the DCSF and accordingly interacts with the UE for connecting the UE's audio/video media termination to the MF/MRF [4], and interacts with MF/MRF for data channel media resource management for AR media processing.

# 5 Immersive AR Media

## 5.1 General

An AR-MTSI client supports simultaneous transfer of multiple media components with real-time characteristics. An AR-MTSI client supports the core media components in [2] for a conversational AR scenario including text, image, video and speech (also referred to as audio).

## 5.2 Speech

[LINK TO IVAS]

## 5.3 Video

An AR-MTSI client in terminal may display video content based on AR metadata (Clause 6) and media configuration (Clause 7). The AR real-time communication aspects of this specification are not restricted to specific codecs. However, in order to support minimum service interoperability, an AR-MTSI client in terminal shall implement the UE codec and media handling requirements as specified in TS 26.114 [2].

Specifically, the AR-MTSI client in terminal may support Overlays and Scene Description-Based Overlays (as described in TS 26.114 [2] in clause Y.6.4 and Y.6.9) to render video elements in parts of the AR environment. This may result into rendering the video stream (or parts of the video stream) in a sub-area of the display device. Further, the UE may negotiate a stream characteristic most suitable for the sub-area and may renegotiate the stream characteristics in case the sub-area changes.

## 5.4 Real-time text

An AR-MTSI client in terminal may support real-time text as specified in TS 26.114 [2] and may render it as defined in the AR metadata (Clause 6)

## 5.5 Still images

An AR-MTSI client in terminal may support still images as specified in TS 26.114 [2] and may render it as defined in the AR metadata (Clause 6)

# 6 AR Metadata

## 6.1 General

Real-time scene creation for an AR conference with two or more participants may be done by the MRF to create a symmetric experience for all participants. For an MRF to create a scene, it may request the following information from the UEs:

- spatial description of the space surrounding the UE e.g., the occlusion-free space around the user in which the AR media will be rendered.

- media properties indicating the AR media that the UE will be sending, and thus have to be incorporated in the scene.

- receiving media capabilities of the UEs, which may include

- UE media decoding capabilities

- UE hardware capabilities (e.g., the display resolution)

- information based on detecting the location, orientation, and capabilities of physical world devices, eligible for usage in an audio-visual communications session

Based on this information the MRF creates a scene which includes:

- defining the placement of the user and the AR media in that scene, including e.g., the position, size, depth from the user, anchor type, and recommended resolution (or quality)

- specific rendering properties for the AR media, e.g., for a 2D object to be rendered with a billboarding effect

The MRF can then share the scene with the participant UEs using a supported scene description format. This scene description may be different for different UEs.

NOTE: The scene as sent by the MRF allows the UE to 1) select and request any related media (for example, in a quality and bitrate based on the rendering characteristics or network connection), 2) render the complete scene on a (virtual) display device, and 3) update the rendering and requested media dynamically (e.g., according to the movement and view orientation of the user).

## 6.2 Metadata data channel message format

For the carriage of metadata defined in this clause the AR-MTSI clients shall use the data channel. The data channel sub-protocol shall be identified as “3gpp-ar-metadata”, which shall be included in the dcmap attribute of the SDP.

The transmission order for the data channel shall be set to in-order and the transmission reliability shall be set to reliable.

The metadata message format shall be set to text-based and the messages shall be UTF-8 encoded JSON messages.

A data channel message may carry one or more AR metadata messages as defined in Table 6.2-1.

Table 6.2-1 AR Metadata Messages Format

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Cardinality** | **Description** |
| messages | Array(Message) | 1..n | A list of AR metadata messages. Each message shall be formatted according to the Message data type as defined in Table 6.2-2 |

Each metadata message shall follow the format specified in Table 6.2-2.

Table 6.2-2 AR Metadata Message Data Type

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Type** | **Cardinality** | **Description** |
| id | string | 1..1 | A unique identifier of the message in the scope of the data channel session. |
| type | string | 1..1 | A urn that identifies the message type.  |
| payload | object | 1..1 | The message payload depends on the message type. |
| sendingAtTime  | number | 0..1 | The wall clock time when the AR metadata message is transmitted. (ref. T1’) |

Editor’s Note: T1’ is used in MeCAR, it should be updated with correct reference once TS 26.119 defines it.

## 6.3 Spatial descriptions

### 6.3.1 Spatial description format

#### 6.3.1.1 General

A spatial description format is used for defining the physical space around a UE or trackable in which. This clause includes the supported formats and the method for exchanging the information between AR-MTSI clients.

#### 6.3.1.2 Available visualization space and user position

An AR-MTSI client in terminal may send available visualization space, user position and other trackable poses to AR MF/MRF for scene creation and update. Trackable is a real-world object (e.g., the UE, floor, controllers, table etc.) that the UE can detect, which can be used as a reference to anchor virtual objects to the real world..

The available visualization space defines an occlusion-free space around the user for rendering the AR scene as a geometric primitive. The format for available visualization space is defined in clause 6.2.4 of [3]. The type of the message containing visualization space as a payload shall be “**urn:3gpp:ar:v1:visualization-space**”. The availableVisualizationSpace object [3] shall contain a xrSpaceId. The xrSpaceId is used for determining the local coordinate axis of the visualization space. The xrSpaceId shall be a unique identifier for an XR space of one AR-MTSI client in terminal.

The AR-MTSI client in terminal that sends the available visualization space may also send at least one pose for a trackable e.g., to determine the position of the user within the visualization space. The AR-MTSI client in terminal may send additional poses for anchoring virtual objects. The poses shall be sent using the format defined in clause 6.2.2 of [3]. The poseInfo (as defined in Table 6.2.2-1 [3]) shall contain an xrSpaceId that is the same as the one used for visualization space. The poses may additionally contain a label string to identify the type of anchor. The labels are application-dependent, but for example, user, floor, left controller etc., can be used as labels. The type of the message for a pose sent for scene creation shall be set to “**urn:3gpp:ar:v1:initial-pose**”.The format for initial user pose shall be as defined in the table below. If an initial pose is not signalled with a visualization space, the space is assumed to be anchored to the viewer pose, assumed to be the center of the visualization space, i.e., orientation [x,y,z,w] = [0,0,0,1] and position [x,y,z] =[0,0,0].

## 6.4 Scene descriptions

An AR-MTSI client in terminal [shall] comply with the capabilities requirements for scene description as described in clause 10 of TS26.119 for their respective device type.

A scene description of an AR session is sent from the AR MF/MRF to the AR-MTSI clients in terminal. An MF that supports exchange of scene description shall support the following:

- glTF 2.0 scenes as specified in [X].

- MPEG\_media extension as defined in [X2] to refer to RTP media streams. The external media shall be RTP media streams supported by an AR-MTSI client and signalled in the SDP.

- MPEG\_anchor\_extension as defined in [X2] to anchor objects in the real-world.

The scene description shall be sent by the AR MF/MRF to the AR-MTSI client in terminal over the data channel. The type of the message shall be set to “**urn:3gpp:ar:v1:sd**”.

An AR MF/MRF that supports scene descriptions should create and distribute the scene for an AR call with audio and video streams based on the visualization space, viewer position and AR media properties. The AR MF/MRF should create the scene description for each participant (AR-MTSI client in terminal) such that the shared experience is symmetrical for the different users in the call, e.g., to maintain relative position of users and objects.

Editor’s Note: To be aligned with Mecar.

6.5 Network media rendering

### 6.5.1 General

The AR-MTSI client in terminal supporting network media rendering shall support metadata formats for split rendering as specified in clause 8.2.2 of TS 26.565 [6].

[Editor’s Note: It is FFS whether AR-MTSI client in terminal specific extension is required.]

[Editor’s Note: The following clauses will potentially reference the corresponding format in TS 26.119]

### 6.5.2 Pose Format

When the network media rendering is activated, the AR-MTSI client in terminal periodically transmits a set of pose predictions to the AR AS. The pose prediction format shall conform to the payload of the message whose type is "**urn:3gpp:split-rendering:v1:pose**" as specified in clause 8.2.2.2 of TS 26.565 [6].

### 6.5.3 Action Format

The action sets and actions are negotiated during the AR media rendering negotiation. The AR-MTSI client in terminal reports any changes to action state as it occurs by sending updated actions to the AR AS after the network media rendering is activated. When the AR-MTSI client in terminal sends updated actions to the AR AS, the action format shall conform to the payload of the message whose type is "**urn:3gpp:split-rendering:v1:action**" as specified in clause 8.2.2.3 of TS 26.565 [6].

Editor’s Note: The clauses should be aligned with MeCAR.

# 7 Media configurations

7.1 General

The media configuration requirements for MTSI clients in terminals specified in TS 26.114 [2], clause 6, also apply for AR-MTSI client in terminal.

An SDP framework for AR data exchange for AR communication is presented to negotiate codec support for AR media, AR metadata, as well as RTP/RTCP signalling necessary for AR media rendering processing.

AR-MTSI client in terminal shall use RTP for the real-time transport of AR media for AR communication. Any AR media as an overlay may refer to the overlay configuration described in clause Y 6.4.3 of TS 26.114 [2]*.*

AR-MTSI client in terminal shall use the data channels for exchange of AR metadata and rendering negotiation. The SDP attribute *3gpp\_armetadata\_types* should be used to indicate the types of AR metadata which defined in clause 6 (e.g. pose, action and scene description) within the data channel.

The syntax for the SDP attribute is:

a=3gpp\_armetadata\_types: <metadata-1> / … / <metadata-N>

Editor’s Note: The attribute is under “a=dcmap” line with subprotocol=" 3gpp-ar-metadata" , and the ABNF syntax for the attribute may be further specified somewhere in the document if needed.

The poses as part of AR metadata may be transmitted via RTP session as a RTP header extension which specified in clause 4.3 of TS 26.522.

7.2 Network media rendering configuration

As specified in Annex AC.9 of TS 23.228 [4], the AR application server can provides network assisted rendering. An AR-MTSI client in terminal can decide to request network media rendering based on user selection and its status such as power, signal, computing power, internal storage, etc. The AR-MTSI client in terminal shall complete an AR media rendering negotiation with the AR AS before it initiates subsequent procedures to activate the network media rendering. The data channel should be established for rendering negotiation with SDP offer/answer between AR-MTSI client in terminal and MF/MRF with the sub-protocol “3gpp-sr-conf”, and continue to be used for rendering re-negotiation until the end of the AR communication.

The AR media rendering negotiation between the AR-MTSI client in terminal and the AR AS shall determine the split-rendering configuration which exchanged via the data channel of rendering negotiation. The split-rendering configuration shall be in JSON format as specified in clause 8.4.2 of TS 26.565 [6].

Editor’s Note: We need to consider to specify a minimum set of network rendering configuration for AR communication refer to TS 26.565.

# 8 AR Data Transport

## 8.1 General

[Editor’s Note: RTP-based media transport will be defined in 5G\_RTP and subsequently integrated/adopted in IBACS.]

The data transport requirements for MTSI clients in terminals specified in TS 26.114 [2], clause 7, also apply for AR-MTSI clients in terminals.

## 8.2 RTP transport

## 8.3 RTCP usage

## 8.4 Data channel usage

# 9 Quality of Experience

[Editor’s Note: It is expected that the Quality of Experience requirements for MTSI clients in terminals specified in TS 26.114 also apply for terminals to be specified by this specification]

Annex A (normative):
Call flows for IBACS

# A.1 IMS AR communication Call Flows

## A.1.1 General

Figure A.1.1-1 illustrates a high-level call flow for AR communication.


Figure A.1.1-1: High-level call flow for AR communication

The following steps are performed:

A. AR Call Session Setup: UE1 initiates an AR call, and the AR call session is established between UE1 and UE2, data channels along with audio and video channels are established between UE and network

B. Split Rendering Negotiation: When the UE has poor rendering capability which is not able to satisfy the requirements of AR communication, the split rendering negotiation is involved between the UE and IMS. The split may be adjustable during a session. This split rendering negotiation step can be executed/re-executed during the session when the UE’s status changed and/or user selection.

C. Scene Description Processing: Prepare and generate scene description updates for the AR call session, this procedure can be done either on UE or IMS network.

D. AR Media Processing: AR media & Metadata transmission and AR media rendering for the AR call session, if split rendering is enabled, this procedure can be done by both UE and IMS network.

## A.1.2 AR Call Session Setup

The AR call session procedure in clause AC.7 of TS 22.228[4] shall be followed [4].

## A.1.3 Split Rendering Negotiation

Figure A.1.3-1 illustrates a detailed call flow for split rendering procedure.



Figure A.1.3-1: Split Rendering Negotiation

The steps are as follows:

1. The UE1 initiates an AR communication session and establishes audio and video session connections with the UE2. Then the bootstrap and application data channels are established for the UE1 and UE2.

2. When the UE1 discovers that its media capabilities cannot meet the AR communication related media rendering requirements, the UE1 decides to start split rendering call flow. Then the UE1 calculates which AR objects can be rendered by itself based on its status, and decides which part of the AR objects to be rendered in the UE1 and the others to be rendered in the IMS network.

3. The UE1 initiates the application data channels between the UE1 and the MF/MRF, for the split rendering request and metadata transmission.

4-5. The UE1 sends a Split Rendering Request to the MF/MRF through the established application data channel, the request includes the information of the AR objects to be rendered in IMS network, the MF/MRF transfers the request to the AR AS.

Note: The use of data channel for split rendering procedure (steps 2, 4 and 5) is optional. Alternatively, the MF/MRF or AR AS may decide to use split rendering for media delivery based on, e.g., device capability and media properties.

6. The AR AS may decide whether to provide AR media rendering function based on the request message received from the UE1, the media rendering resource available on the MF/MRF and the split rendering provisioning status for the UE1.

7. The AR AS sends a request to the DCSF to create required AR media rendering resource on the MF/MRF for the AR objects that should be rendered in IMS network, the DCSF transfers the request to the IMS AS.

NOTE: The requested AR media rendering resources are such that they can accommodate expected changes in scene description for the session.

8. The IMS AS sends the media resource allocation request to the MF/MRF, to reserve AR media rendering resource for the UE1.

9. When the MF/MRF resources are allocated successfully, the IMS AS returns a successful response to the DCSF, the DCSF transfers the response to the AR AS.

10-11 The AR AS returns a successful Split Rendering Response carrying the result to the MF/MRF. The MF/MRF then transfers the response/notification message with split rendering configuration information to the UE1 through the application data channel.

12. Subsequent procedures continue.

## A.1.4 Scene Description Processing

Figure A.1.4-1 illustrates a detailed call flow for scene description processing procedure.



Figure A.1.4-1: Scene Description Processing

The steps are as follows:

1. The UE1 initiates an AR communication session and establishes audio and video session connections with the UE2. Then the bootstrap and application data channels are established for the UE1 and UE2.

2. The split rendering negotiation procedure has been finished.

3. MF/MRF prepares the scene description based on media descriptions and assets for the call.

4. MF/MRF delivers the scene description to the UEs.

5. A UE may trigger a scene update e.g., when a new object is added/removed in the scene, or a spatial information update is sent or UE capabilities change. The figure shows the update is triggered by UE1, but this can be either UE.

6. The MF/MRF will process the new information and creates a scene description update. It is also possible for the MF/MRF to initiate an update without an update from the UEs, for example if link conditions change.

7. MF/MRF distributes scene description update to all UEs.

NOTE: Spatial data related updates may be required for collaborative AR calls, e.g., when multiple users are physically collocated and also part of the same AR experience. The type of spatial description updates is FFS.

8. Subsequent procedures continue.

## A.1.5 AR Media Processing

Figure A.1.5-1 illustrates a detailed call flow for AR media processing procedure.



Figure A.1.5-1: AR Media Processing

The steps are as follows:

1. The UE1 initiates an AR communication session and establishes audio and video session connections with the UE2. Then the bootstrap and application data channels are established for the UE1 and UE2.

2. The split rendering negotiation procedure has been finished.

3. The scene description processing has been finished.

4a. The UE1 captures the AR metadata, performs AR media rendering locally and then encodes rendered AR media, e.g., as audio/video media stream.

5a. The UE1 sends the rendered AR media and/or AR metadata to the peer through the established media connection(s).

6a. The UE1 receives the AR media and/or AR metadata from the peer through established media connection(s).

7a. The UE1 decodes the AR metadata and performs AR media rendering locally and displays it on its screen.

Network Centric Procedure:

4b. The UE1 retrieves AR metadata such as pose and user input locally, and renders the part of AR objects that should be done in the UE1 according to the result in step 2.

5b. The UE1 sends AR media to the MF/MRF, the AR media can include the audio/video media stream rendered by the UE1, and the AR metadata which is used for the AR objects rendering that should be done in the IMS network.

6b. The UE2 may also send AR media and/or AR metadata to the MF/MRF, e.g., the viewport.

7b. The MF/MRF renders the part of AR objects based on the AR metadata received. Then the MF/MRF decodes the audio/video media stream received from the UE1, and combines with the AR media rendered by the MF/MRF.

8b. The MF/MRF sends the rendered audio/video media stream to UE2.

9b. The MF/MRF sends the rendered audio/video media stream to the UE1.

10. A UE may trigger a scene update as described in clause 9.1.3. for example, if objects are added/removed from the scene or if UE capabilities change and it can no longer render some AR objects as negotiated in step 2.

11. The MF/MRF processes the new information and creates a scene description update. It is also possible for the MF/MRF to initiate an update without an update from the UEs, for example if network conditions change.

12. The MF/MRF distributes the scene description update to all UEs.

Annex <X> (informative):
Change history

|  |
| --- |
| Change history |
| Date | Meeting | TDoc | CR | Rev | Cat | Subject/Comment | New version |
| 2022-08 | SA4#120-e | S4-221017 |  |  |  | Initial skeleton | 0.0.1 |
| 2022-08 | SA4#120-e | S4-221202 |  |  |  | Version agreed during SA4#120-e | 0.1.0 |
| 2023-08 | SA4#125 | S4-231474 |  |  |  | Scope, references, definitions, and general descriptions | 0.2.0 |
| 2023-08 | SA4#125 | S4-231476 |  |  |  | Spatial description format | 0.2.0 |
| 2023-11 | SA4#126 | S4-231783 |  |  |  | Draft TS 26.264 v0.2.1 implemented S4aR230099 on end-to-end reference architecture and S4aR230121 on the usage of a scene | 0.3.0 |
| 2023-11 | SA4#126 | S4-231784 |  |  |  | Add general requirements applied both AR-MTSI and MTSI clients and remove unnecessary clauses | 0.3.0 |
| 2023-11 | SA4#126 | S4-231785 |  |  |  | Reuse Split Rendering Formats for session setup and negotiation and metadata defined in TS 26.565 | 0.3.0 |
| 2023-11 | SA4#126 | S4-231967 |  |  |  | Generalized reference architecture based on IMS DC architecture in TS 23.228 | 0.3.0 |
| 2024-02 | SA4#127 | S4-240393 |  |  |  | Agreements in SA4#127: S4-240163, S4-240165, S4-240370, S4-240323 | 0.4.0 |
| 2024-03 | SA#103 | SP-240024 |  |  |  | Version 1.0.0 created by MCC | 1.0.0 |
| 2024-04 | SA4#127-bis-e | S4-240804 |  |  |  | Agreements in SA4#127-bis-e: S4-240655, S4-240763, S4-240823 | 1.1.0 |