ETSI TR 126 928 V18.0.0 (2024-05)

5G;
Extended Reality (XR) in 5G
(3GPP TR 26.928 version 18.0.0 Release 18)

Reference
RTR/TSGS-0426928vi00

Keywords
5G

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - APE 7112B
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° w061004871

Intellectual Property Rights

Essential patents

IPRs essential or potentially essential to normative deliverables may have been declared to ETSI. The declarations pertaining to these essential IPRs, if any, are publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (https://ipr.etsi.org/).

Pursuant to the ETSI Directives including the ETSI IPR Policy, no investigation regarding the essentiality of IPRs, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Trademarks

The present document may include trademarks and/or tradenames which are asserted and/or registered by their owners. ETSI claims no ownership of these except for any which are indicated as being the property of ETSI, and conveys no right to use or reproduce any trademark and/or tradename. Mention of those trademarks in the present document does not constitute an endorsement by ETSI of products, services or organizations associated with those trademarks.

DECT™, **PLUGTESTS™**, **UMTS™** and the ETSI logo are trademarks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are trademarks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners. **oneM2M™** logo is a trademark of ETSI registered for the benefit of its Members and of the oneM2M Partners. **GSM**® and the GSM logo are trademarks registered and owned by the GSM Association.

Legal Notice

This Technical Report (TR) has been produced by ETSI 3rd Generation Partnership Project (3GPP).

The present document may refer to technical specifications or reports using their 3GPP identities. These shall be interpreted as being references to the corresponding ETSI deliverables.

The cross reference between 3GPP and ETSI identities can be found under https://webapp.etsi.org/key/queryform.asp.

Modal verbs terminology

In the present document "**should**", "**should not**", "**may**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the ETSI Drafting Rules (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

Contents

Foreword

This Technical Report 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.

Introduction

This Technical Report collects information on eXtended Reality (XR) in the context of 5G radio and network services. Extended reality (XR) refers to all real-and-virtual combined environments and associated human-machine interactions generated by computer technology and wearables. It includes representative forms such as augmented reality (AR), mixed reality (MR), and virtual reality (VR) and the areas interpolated among them. In this Technical Report, baseline technologies for XR type of services and applications are introduced outlining the QoE/QoS issues of XR-based services, the delivery of XR in the 5G system, and an architectural model of 5G media streaming defined in TS 26.501. In addition to the conventional service category, interactive, streaming, download, and split compute/rendering are identified as new delivery categories. A survey of 3D, XR visual and audio formats is also provided.

Use cases and device types are classified, and processing and media centric architectures are introduced. This includes viewport independent and dependent streaming, as well as different distributed computing architectures for XR. Core use cases of XR include those unique to AR and MR in addition to those of VR discussed in 3GPP TR 26.918, ranging from offline sharing of 3D objects, real-time sharing, multimedia streaming, online gaming, mission critical applications, and multi-party call/conferences. Based on the details in the report, proposals for potential standardisation areas are documented.

1 Scope

The present document collects information on eXtended Reality (XR) in the context of 5G radio and network services. The primary scope of the present document is the documentation of the following aspects:

- Introducing Extended Reality by providing definitions, core technology enablers, a summary of devices and form factors, as well as ongoing related work in 3GPP and elsewhere,
- Collecting and documenting core use cases in the context of Extended Reality,
- Identifying relevant client and network architectures, APIs and media processing functions that support XR use cases,
- Analysing and identifying the media formats (including audio and video), metadata, accessibility features, interfaces and delivery procedures between client and network required to offer such an experience,
- Collecting key performance indicators and Quality-of-Experience metrics for relevant XR services and the applied technology components,
- Drawing conclusions on the potential needs for standardisation in 3GPP.

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 TR 26.918: "Virtual Reality (VR) media services over 3GPP".

[3] 3GPP TS 26.118: "3GPP Virtual reality profiles for streaming applications".

[4] ARCore, https://developers.google.com/ar/

[5] ARKit, https://developer.apple.com/arkit/

[6] 3GPP TR 22.842: "Study on Network Controlled Interactive Service in 5GS".

[7] 3GPP TR 26.247: "Transparent end-to-end Packet-switched Streaming Service (PSS); Progressive Download and Dynamic Adaptive Streaming over HTTP (3GP-DASH)".

[8] 3GPP TS 23.501: "System Architecture for the 5G System".

[9] Schuemie, Martijn J., Peter Van Der Straaten, Merel Krijn, and Charles A.P.G. Van Der Mast. “Research on Presence in Virtual Reality: A Survey.” CyberPsychology & Behavior, Vol. 4, No. 2. April 2001.

[10] Ching, Teo Choong. "The Concept of Presence in Virtual Reality." Medium. 27 August 2016.

[11] Sparks, Matt. "Don't Break the Spell: Creating Presence in Virtual Reality" Learning Solutions Magazine, 17 July 2017.

[12] 3GPP TS 26.501: "5G Media Streaming Architecture".

[13] 3GPP TS 22.173: "IP Multimedia Core Network Subsystem (IMS) Multimedia Telephony Service and supplementary services; Stage 1".

[14] 3GPP TS 26.114: "IP Multimedia Subsystem (IMS); Multimedia Telephony; Media handling and interaction".

[15] 3GPP TR 22.891 "Feasibility Study on New Services and Markets Technology".

[16] Khronos, "The OpenXR Specification", Jan 25, 2020, https://www.khronos.org/registry/OpenXR/specs/1.0/html/xrspec.html#introduction

[17] W3C, "WebXR Device API", https://www.w3.org/TR/webxr/

[18] Rolland, Jannick & Holloway, Richard & Fuchs, Henry. (1994). Comparison of optical and video see-through, head-mounted displays. Proceedings of SPIE - The International Society for Optical Engineering. 10.1117/12.197322.

[19] "Cloud Gaming: Architecture and Performance", Ryan Shea and Jiangchuan Liu, Simon Fraser University; Edith C.-H. Ngai, Uppsala University; Yong Cui, Tsinghua University; IEEE Network-July/August 2013.

[20] M. Claypool and K. Claypool. Latency and player actions in online games. Communications of the ACM, 49(11):40–45, 2006.

[21] Quax, P., Monsieurs, P., Lamotte, W., De Vleeschauwer, D., and Degrande, N. Objective and subjective evaluation of the influence of small amounts of delay and jitter on a recent first person shooter game. In Proceedings of 3rd ACM SIGCOMM workshop on Network and system support for games (New York, NY, USA, 2004), NetGames '04, ACM, pp. 152–156.

[22] Chen, K.-t., Huang, P., Wang, G.-s., Huang, C.-y., and Lei, C.-l. On the Sensitivity of Online Game Playing Time to Network QoS. Proceedings of IEEE INFOCOM 2006 00, c (2006).

[23] 3GPP TS 23.501: "System architecture for the 5G System (5GS)".

[24] 3GPP TS 38.300: "NR; Overall description; Stage-2".

[25] 3GPP TS 22.173: "IP Multimedia Core Network Subsystem (IMS) Multimedia Telephony Service and supplementary services; Stage 1".

[26] 3GPP TS 26.114: "IP Multimedia Subsystem (IMS); Multimedia telephony; Media handling and interaction".

[27] 3GPP TR 23.758: "Study on application architecture for enabling Edge Applications"

[28] 3GPP TR 23.748: "Study on enhancement of support for Edge Computing in 5G Core network (5GC)".

[29] 3GPP TS 23.558: "Architecture for enabling Edge Applications (EA)".

[30] Recommendation ITU-T H.264 (04/2017): "Advanced video coding for generic audiovisual services" | ISO/IEC 14496-10:2014: "Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding".

[31] Recommendation ITU-T H.265 (12/2016): "High efficiency video coding" | ISO/IEC 23008-2:2015: "High Efficiency Coding and Media Delivery in Heterogeneous Environments – Part 2: High Efficiency Video Coding".

[32] 3GPP TS 26.116: "Television (TV) over 3GPP services; Video profiles". [33] Jens-Rainer Ohm, Gary J. Sullivan, Heiko Schwarz, Thiow Keng Tan, and Thomas Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards—Including High Efficiency Video Coding (HEVC)" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 22, NO. 12, DECEMBER 2012.

[34] T.K. Tan, M. Mrak, R. Weerakkody, N. Ramzan, V. Baroncini, G.J. Sullivan, J.-R. Ohm, K.D. McCann, "HEVC subjective video quality test results", IBC2014 Conference, 2014.

[35] Thiow Keng Tan ; Rajitha Weerakkody ; Marta Mrak ; Naeem Ramzan ; Vittorio Baroncini, Jens-Rainer Ohm, Gary J. Sullivan, "Video Quality Evaluation Methodology and Verification Testing of HEVC Compression Performance" IEEE Transactions on Circuits and Systems for Video Technology, Volume: 26 , Issue: 1 , Jan. 2016.

[37] ISO/IEC 23090-2: "Information technology — Coded representation of immersive media — Part 2: Omnidirectional media format"

[38] S. Schwarz et al., "Emerging MPEG Standards for Point Cloud Compression," in IEEE Journal on Emerging and Selected Topics in Circuits and Systems, vol. 9, no. 1, pp. 133-148, March 2019.

[39] Khronos, "The GL Transmission Format (glTF)", Jun 9, 2017, https://github.com/KhronosGroup/glTF/blob/master/specification/2.0/README.md

[40] Long Qian, Alexander Barthel, Alex Johnson, Greg Osgood, Peter Kazanzides, Nassir Navab, and Bernhard Fuerst, "Comparison of optical see-through head-mounted displays for surgical interventions with object-anchored 2D-display", Int J Comput Assist Radiol Surg. 2017 Jun; 12(6): 901–910, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5891507/.

[41] 3GPP TS 22.261: "Service requirements for the 5G system".

[42] 3GPP, VR-IF and the Advanced Imaging Society's 2ND VR ECOSYSTEMS & STANDARDS WORKSHOP, Culver City, CA, US, https://www.vr-if.org/events/3gpp-vrif-ais-workshop/.

[43] Slater, Mel and Martin Usoh. "Body Centred Interaction in Immersive Virtual Environments." Body centred interaction in immersive virtual environments. In N. M. Thalmann & D. Thalmann (Eds.), Artificial Life and Virtual Reality (pp. 125-148). New York: John Wiley. 1994.

3 Definitions of terms, symbols and abbreviations

3.1 Terms

For the purposes of the present document, the terms given in 3GPP 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 3GPP TR 21.905 [1].

3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP 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 3GPP TR 21.905 [1].

3DoF	Three Degrees of Freedom
5QI	5G QoS Identifier
6DoF	Three Degrees of Freedom
AI	Artificial Intelligence
API	Application Programming Interface
AR	Augmented Reality
ARP	Allocation and Retention Priority
ASIC	Application-Specific Integrated Circuit
ASTC	Adaptive Scalable Texture Compression
ATW	Asynchronous TimeWarp
AVC	Advanced Video Coding
BC1	Block Compression for RGB
CAD	Computer-Aided Design
CBR	Constant BitRate
CDN	Content Delivery Network
CPU	Compute Processing Unit
CTC	Call for TeChnologies
DASH	Dynamic Adaptive Streaming over HTTP

DL	DownLink
DNS	Domain Name System
DoF	Degrees of Freedom
EAC	Ericsson Alpha Compression
ERP	Equi-Rectangular Projection
ETC2	Ericsson Texture Compression version 2
EVC	Essential Video Coding
FFS	For Further Study
FLUS	Framework for Live Uplink Streaming
FOV	Field-Of-View
FPS	Frames Per Second
GBR	Guaranteed BitRate
GFBR	Guaranteed Flow Bit Rate
GNSS	Global Navigation Satellite System
G-PCC	Geometry-based Point Cloud Compression
GPS	Global Positioning System
GPU	Graphics Processing Unit
HEVC	High-Efficiency Video Coding
HMD	Head-Mounted Display
HRTF	Head-Related Transfer Function
HTTP	Hyper-Text Transfer Protocol
HUD	Heads-Up Display
IDMS	Inter-destination Multimedia Synchronization
IMU	Inertial Measurement Unit
IOD	Inter-aural Output Difference
IVAS	Immersive Voice and Audio Services
JPEG	Joint Photographic Experts Group
JVET	Joint Video Exploration Team
LIDAR	Light Detection and Ranging
MCPTT	Mission Critical Push To Talk
MCU	Multipoint Control Unit
MEC	Multi-access Edge Computing
MFBR	Maximum Flow Bit Rate
MMS	Multimedia Messaging Service
MOBA	Multiplayer Online Battle Arena
MPEG	Moving Pictures Expert Group
MR	Mixed Reality
NBMP	Network-Based Media Processing
NCIS	Network Controlled Interactive Service
NEF	Network Exposure Function
PBR	Physically-Based Rendering
PCC	Point Cloud Compression
PCF	Policy Control Function
PDB	Packet Delay Budget
PDU	Packet Data Unit
PER	Packet Error Rate
PLY	PoLYgon
PNG	Portable Network Graphics
PPI	Pixels Per Inch
PQI	PC5 QoS Identifier
PSS	Packet-Switched Streaming
PTT	Push To Talk
PVRTC	PowerVR Texture Compression
QCI	QoS Class Identifier
QFI	QoS Flow ID
QoE	Quality of EXperience
QoS	Quality of Service
RCS	Rich Communication Service
RGB	Red-Green-Blue colour space
RGBD	Red-Green-Blue-Depth
RPG	Role Playing Game
RQA	Reflective QoS Attribute

RTP	Real-Time Protocol
RTS	Real-time Strategy
RTT	Round Trip Time
SCS	Spatial Compute Server
SDP	Session Description Protocol
SIP	Session Initiation Protocol
SLAM	Simultaneous Localization and Mapping
SWB	Super WideBand
TCP	Transmission Control Protocol
ToF	Time of Flight
TPU	Tensor Processing Unit
UL	UpLink
USB	Universal Serial Bus
VCL	Video Coding Layer
V-PCC	Video-based Point Cloud Compression
VPS	Visual Positioning System
VR	Virtual Reality
VVC	Versatile Video Coding
XR	Extended reality
YUV	Luminance-Bandwidth-Chrominance

4 Introduction to Extended Reality

4.1 XR Terms and Definitions

4.1.1 Different Types of Realities

The scope of this clause is the introduction of eXtended Reality (XR) to 3GPP services and networks. eXtended Reality (XR) is an umbrella term for different types of realities as shown in Figure 4.1-1. The figure also shows different application domains of XR such as entertainment, healthcare, education, etc. The different terms are defined in the following, reusing and extending some definitions from 3GPP TR26.918 [2].

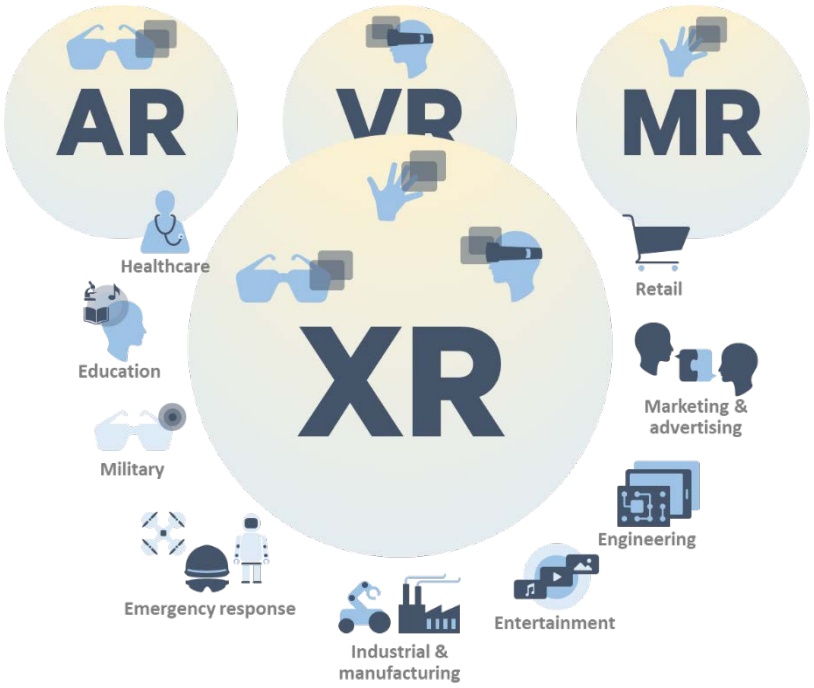


Figure 4.1-1: Different Types of Realities and some applications

Virtual reality (VR) is a rendered version of a delivered visual and audio scene. The rendering is designed to mimic the visual and audio sensory stimuli of the real world as naturally as possible to an observer or user as they move within the limits defined by the application. Virtual reality usually, but not necessarily, requires a user to wear a head mounted display (HMD), to completely replace the user's field of view with a simulated visual component, and to wear headphones, to provide the user with the accompanying audio. Some form of head and motion tracking of the user in VR is usually also necessary to allow the simulated visual and audio components to be updated in order to ensure that, from the user's perspective, items and sound sources remain consistent with the user's movements. Additional means to interact with the virtual reality simulation may be provided but are not strictly necessary.

Augmented reality (AR) is when a user is provided with additional information or artificially generated items or content overlaid upon their current environment. Such additional information or content will usually be visual and/or audible and their observation of their current environment may be direct, with no intermediate sensing, processing and rendering, or indirect, where their perception of their environment is relayed via sensors and may be enhanced or processed.

Mixed reality (MR) is an advanced form of AR where some virtual elements are inserted into the physical scene with the intent to provide the illusion that these elements are part of the real scene.

Extended reality (XR) refers to all real-and-virtual combined environments and human-machine interactions generated by computer technology and wearables. It includes representative forms such as AR, MR and VR and the areas interpolated among them. The levels of virtuality range from partially sensory inputs to fully immersive VR. A key aspect of XR is the extension of human experiences especially relating to the senses of existence (represented by VR) and the acquisition of cognition (represented by AR).

Other terms used in the context of XR are *Immersion* as the sense of being surrounded by the virtual environment as well as *Presence* providing the feeling of being physically and spatially located in the virtual environment. The sense of presence provides significant minimum performance requirements for different technologies such as tracking, latency, persistency, resolution and optics. For more details, refer to clause 4.2.

Other relevant terms in the context of XR experiences are:

- **Parallax** is the relative movement of objects as a result of a change in point of view. When objects move relative to each other, users tend to estimate their size and distance.
- **Occlusion** is the phenomena when one object in a 3D space is blocking another object from being viewed.

This document uses the acronym XR throughout to refer to equipment, applications and functions used for Virtual Reality, Augmented Reality, and other related technologies. Examples include, but are not limited to:

- Head-mounted displays for Virtual Reality,
- Optical see-through glasses and camera see-through HMDs for Augmented and Mixed Reality,
- Mobile devices with positional tracking and camera.

All in common with them is the ability that they offer some degree of spatial tracking and the spatial tracking results in an interaction to view some form of virtual content. More details on XR devices are provided in clause 4.8.

4.1.2 Degrees of Freedom and XR Spaces

A user acts in and interacts with extended realities as shown in Figure 4.1-2. Actions and interactions involve movements, gestures, body reactions. Thereby, the *Degrees of Freedom (DoF)* describe the number of independent parameters used to define movement of a viewport in the 3D space.

Any consistent interaction for an XR application with XR hardware is assumed to be restricted to an XR session. Once an XR session has been successfully established, it can be used to poll the viewer pose, query information about the user’s environment, and present imagery to the user.

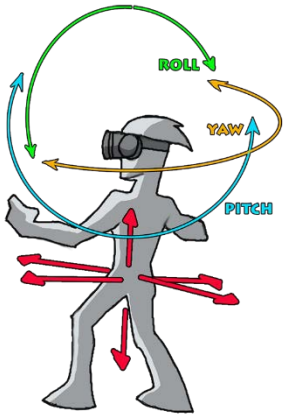
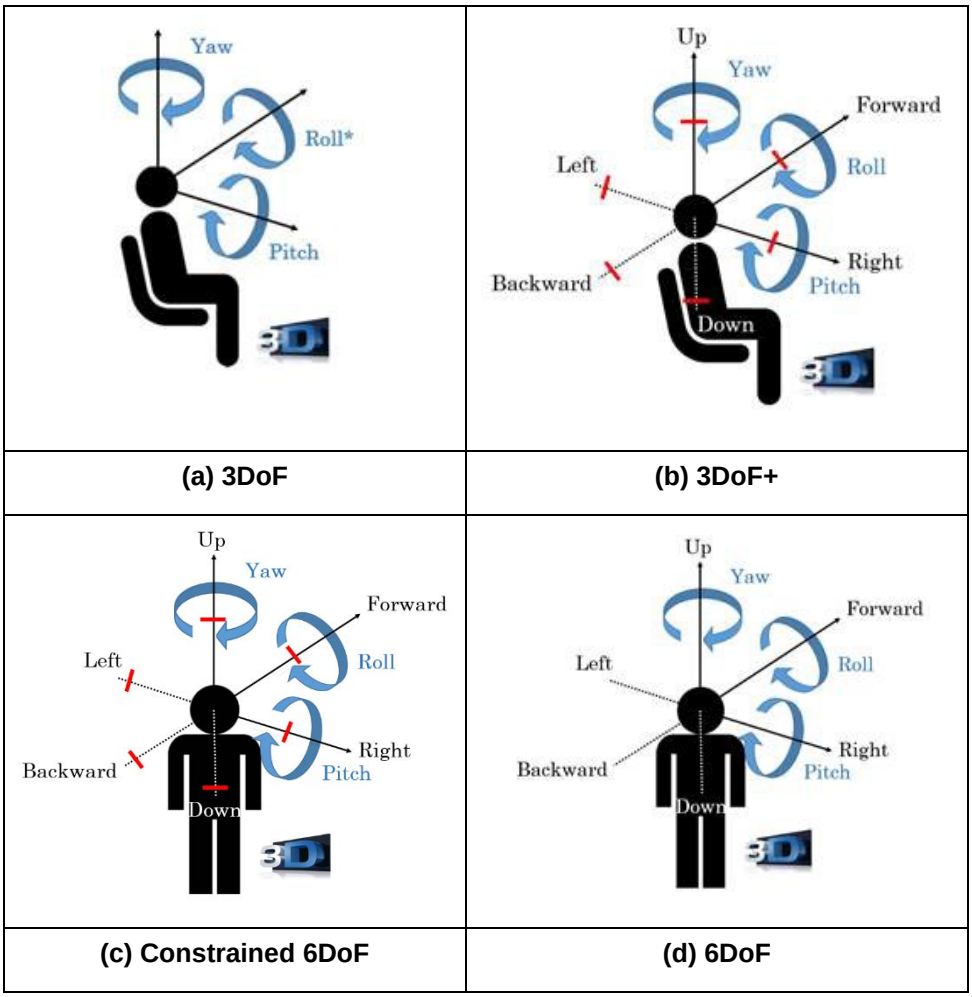


Figure 4.1-2: Different degrees of freedom for a user in extended realities

Typically, the following different types of Degrees-of-Freedom are described (and also shown in Figure 4.1-3).

- **3DoF**: Three rotational and un-limited movements around the X, Y and Z axes (respectively pitch, yaw and roll). A typical use case is a user sitting in a chair looking at 3D 360 VR content on an HMD (see Figure 4.1-3 (a)).
- **3DoF+**: 3DoF with additional limited translational movements (typically, head movements) along X, Y and Z axes. A typical use case is a user sitting in a chair looking at 3D 360 VR content on an HMD with the capability to slightly move his head up/down, left/right and forward/backward (see Figure 4.1-3 (b)).
- **6DoF**: 3DoF with full translational movements along X, Y and Z axes. Beyond the 3DoF experience, it adds (i) moving up and down (elevating/heaving); (ii) moving left and right (strafing/swaying); and (iii) moving forward and backward (walking/surging). A typical use case is a user freely walking through 3D 360 VR content (physically or via dedicated user input means) displayed on an HMD (see Figure 4.1-3 (d)).
- **Constrained 6DoF**: 6DoF with constrained translational movements along X, Y and Z axes (typically, a couple of steps walking distance). A typical use case is a user freely walking through VR content (physically or via dedicated user input means) displayed on an HMD but within a constrained walking area (see Figure 4.1-3 (c)).



(a) 3DoF	(b) 3DoF+
(c) Constrained 6DoF	(d) 6DoF

Figure 4.1-3: Different degrees of freedom

Another term for Constrained 6DoF is *Room Scale VR* being a design paradigm for XR experiences which allows users to freely walk around a play area, with their real-life motion reflected in the XR environment.

Note: Constrained 6DoF is not intended to describe multi-room spaces, areas with uneven floor levels, or very large open areas. Content that handles those scenarios is better categorized as (unconstrained) 6DoF.

The Degrees of Freedom may also be used to describe the tracking capabilities of an XR device. For more details on tracking, refer to clauses 4.1.3 and 4.1.4. Content and tracking capabilities of a device do not necessarily have to match. However, the user is preferably informed by the application on any differences between content and tracking capabilities in terms of number/types of degrees of freedom. .

Spaces provide a relation of the user's physical environment with other tracked entities. An *XR Space* represents a virtual coordinate system with an *origin* that corresponds to a physical location. The world coordinate system is the coordinate system in which the virtual world is created. Coordinate systems are essential for operating in 3-dimensional virtual and real worlds for XR applications.

As an example, a coordinate system is defined by OpenXR [16] in clause 2.15 as well as for WebXR [17], both using a Cartesian right-handed coordinate system as shown in Figure 4.1-4. This coordinate system is right-handed in sense that, where +X is considered "Right", +Y is considered "Up", and -Z is considered "Forward".

A coordinate system is expected to be a rectangular Cartesian in which all axes are equally scaled.

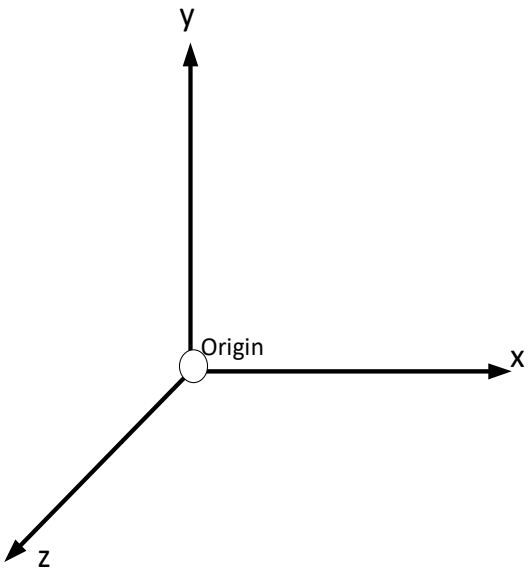


Figure 4.1-4: Right-Handed Coordinate system

A three-dimensional vector is defined by the `(x,y,z)` coordinates. If used to represent physical distances (rather than e.g. velocity or angular velocity) and not otherwise specified, values are in meters. A *position* in the XR space is a 3D-vector representing a position within a space and relative to the origin.

An XR reference space is one of several common XR Spaces that can be used to establish a spatial relationship with the user's physical environment. An XR reference space may be restricted, determining the ability by the user to move. This aligns with the definitions above as well as Figure 4.1-3, namely an XR reference space providing the degrees of freedom for a user.

- For 3DoF, the XR reference space is limited to a single position.
- For 3DoF+, the XR reference space is limited to a small space centered around a single position, a small bounding box, limited to positions attainable with head movements only, around a single position is provided.
- For constrained 6DoF, the XR reference space has a native bounds geometry describing the border around the space, which the user can expect to safely move within. Such borders may for example be described by polygonal boundary given as an array representing a loop of points at the edges of the safe space. The points describe offsets from the origin in meters.
- For 6DoF, the XR reference space is unlimited and basically includes the whole universe.

A simplified diagram (mapped to 2D) on XR Spaces and and their relation to the scene is provided in Figure 4.1-5.

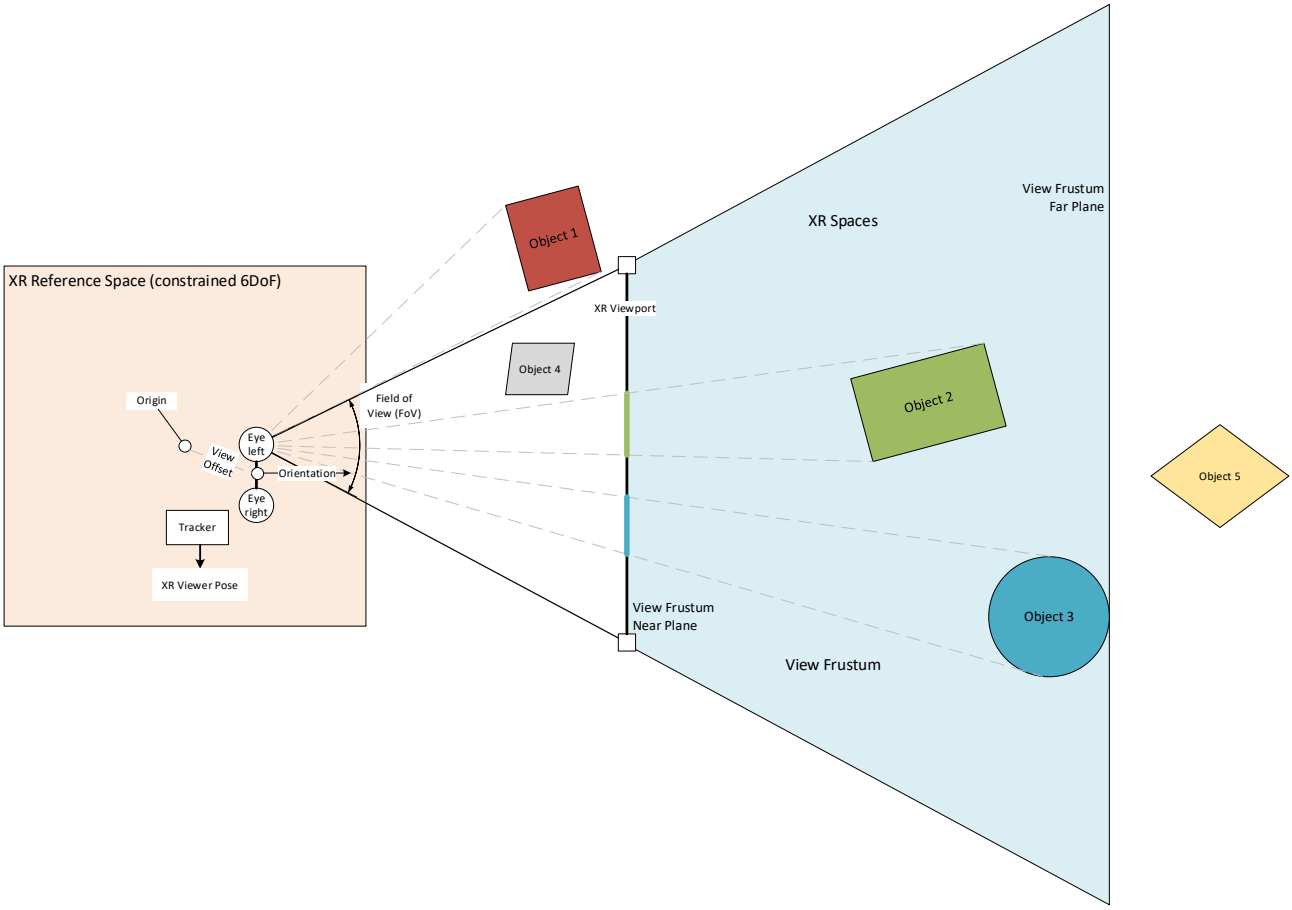


Figure 4.1-5: Simplified Illustration of XR Spaces

Unless the user does a reconfiguration, XR reference spaces within an XR session are static, i.e. the space the user can move in is restricted by the initial definition.

An *XR View* describes a single view into an XR scene for a given time. Each view corresponds to a display or portion of a display used by an XR device to present the portion of the scene to the user. Rendering of the content is expected to be done to well align with the view's physical output properties, including the field of view, eye offset, and other optical properties. A view, among others, has associated

- a *view offset*, describing a *position and orientation* of the view in the XR reference space,
- an *eye* describing which eye this view is expected to be shown. Displays may support stereoscopic or monoscopic viewing.

An *XR Viewport* describes a viewport, or a rectangular region, of a graphics surface. The XR viewport corresponds to the projection of the XR View onto a target display. An XR viewport is predominantly defined by the width and height of the rectangular dimensions of the viewport. In 3D computer graphics, the view frustum is the region of space in the modeled world that may appear on the screen, i.e. it is the field of view of a perspective virtual camera system. The planes that cut the frustum perpendicular to the viewing direction are called the *near plane* and the *far plane*. Objects closer to the camera than the near plane or beyond the far plane are not drawn. Sometimes, the far plane is placed infinitely far away from the camera so all objects within the frustum are drawn regardless of their distance from the camera.

Generally, an *XR Pose* describes a *position* and *orientation* in space relative to an XR Space.

- The *position* in the XR space is a 3D-vector representing the position within a space and relative to the origin defined by the `(x,y,z)` coordinates. If used to represent physical distances, `x`, `y`, and `z` are in meters.
- The orientation in the XR space is a quaternion representing the orientation within a space and defined by a four-dimensional or homogeneous vector with `(x,y,z,w)` coordinates, with `w` being the real part of the quarternion and `x`, `y` and `z` the imaginary parts.

Unit quaternions are used to document spatial rotations in three dimensions. Roll, pitch, and yaw as for example used in TS 26.118 [3] have limitations(for example due to the well-known gimbal lock). Hence, in computer science, engineering and XR applications, they are replaced with the more robust quaternion.

An *XR Viewer Pose* is an XR Pose describing the state of a viewer of the XR scene as tracked by the XR device. XR Viewer Poses are documented relative to an XR Reference Space.

The views array is a sequence of XR Views describing the viewpoints of the XR scene, relative to the XR Reference Space the XR Viewer Pose was queried with.

4.1.3 Tracking and XR Viewer Pose Generation

In XR applications, an essential element is the use of spatial tracking. Based on the tracking and the derived XR Viewer Pose, content is rendered to simulate a view of virtual content.

XR viewer poses and motions can be sensed by *Positional Tracking*, i.e. the process of tracing the XR scene coordinates of moving objects in real-time, such as HMDs or motion controller peripherals. Positional Tracking allows to derive the *XR Viewer Pose*, i.e. the combination of position and orientation of the viewer. Different types of tracking exist:

- *Outside-In Tracking*: a form of positional tracking and, generally, it is a method of optical tracking. Tracking sensors placed in a stationary location and oriented towards the tracked object that moves freely around a designated area defined by sensor coverage.

- *Inside-out Tracking*: a method of positional tracking commonly used in virtual reality (VR) technologies, specifically for tracking the position of head-mounted displays (HMDs) and motion controller accessories whereby the location of the cameras or other sensors that are used to determine the object's position in space are located on the device being tracked (e.g. HMD).

- *World Tracking*: a method to create AR experiences that allow a user to explore virtual content in the world around them with a device's back-facing camera using a device's orientation and position, and detecting real-world surfaces, as well as known images or objects.

- *Simultaneous Localization and Mapping* (SLAM) is the computational problem of constructing or updating a map of an unknown environment while simultaneously keeping track of the user's location within an unknown environment. For more details refer on SLAM, refer to clause 4.1.4.

If not mentioned otherwise, it is assumed that devices in the context of the document are able to track the XR Viewer Pose in 6DoF with sufficient accuracy as defined in clause 4.2.1.

To maintain a reliable registration of the virtual world with the real world as well as to ensure accurate tracking of the XR Viewer pose, XR applications require highly accurate, low-latency tracking of the device at about 1kHz sampling frequency. An XR Viewer Pose consists of the orientation (for example, 4 floating point values in OpenXR [16])) and the position (for example, 3 floating point values in OpenXR [16]). In addition, the XR Viewer Pose needs to have assigned a time stamp. The size of a XR Viewer Pose associated to time typically results in packets of size in the range of 30-100 bytes, such that the generated data is around several hundred kbit/s if delivered over the network.

4.1.4 XR Spatial Mapping and Localization

Spatial mapping, i.e. creating a map of the surrounding area, and localization, i.e. establishing the position of users and objects within that space, are some of the key areas of XR and in particular AR. Multiple sensor inputs are combined to get a better localization accuracy, e.g., monocular/stereo/depth cameras, radio beacons, GPS, inertial sensors, etc.

Some of the methods involved are listed below:

1) **Spatial anchors** are used for establishing the position of a 3D object in a shared AR/MR experience, independent of the individual perspective of the users. Spatial anchors are accurate within a limited space (e.g. 3m radius for the Microsoft® Mixed Reality Toolkit). Multiple anchors may be used for larger spaces.

2) **Simultaneous Localization and Mapping (SLAM)** is used for mapping previously unknown environments, while also maintaining the localization of the device/user within that environment.

3) **Visual Localization**, e.g., vSLAM, Visual Positioning System (VPS), etc., perform localization using visual data from, e.g., a mobile camera, combined with other sensor data.

Spatial mapping and localization can be done on the device. However, network elements can support the operations in different ways:

1) Cloud services may be used for storing, retrieving and updating spatial data. For larger public spaces, **crowdsourcing** may be used to keep the data updated and available to all.

2) A **Spatial Computing Server** that collects data from multiple sources and processes it to create spatial maps including, but not limited to, visual and inertial data streamed from XR devices. The service can then provide this information to other users and also assist in their localization based on the data received from them.

Indoor and outdoor mapping and localization are expected to have different requirements and limitations. Privacy concerns need to be explored by the service provider when scanning indoor spaces and storing spatial features, especially when it is linked to global positioning.

Tracking adds the concept of continuous localisation over time.

4.2 Quality-of-Experience for XR

4.2.1 Immersiveness and Presence

For providing XR experiences that make the user feel *immersed* and *present*, several relevant quality of experience factors have been collected (https://xinreality.com/wiki/Presence). Immersion is an objective description of aspects of the system such as field of view and display resolution. Presence is the feeling of being physically and spatially located in an environment. According to Slater and Usoh [43], “immersion is a necessary rather than a sufficient condition for presence - immersion describes a kind of technology, and presence describes an associated state of consciousness.” Presence is divided into 2 types: Cognitive Presence and Perceptive Presence.

Cognitive Presence is the presence of one's mind. It can be achieved by watching a compelling film or reading an engaging book. Cognitive Presence is important for an immersive experience of any kind.

Perceptive Presence is the presence of one's senses. To accomplish perceptive presence, one's senses, sights, sound, touch and smell, have to be tricked. To create perceptive presence, the XR Device has to fool the user's senses, most notably the audio-visual system. XR Devices achieve this through positional tracking based on the movement. The goal of the system is to maintain your sense of presence and avoid breaking it.

Perceptive Presence is the objective to be achieved by XR applications and is what is referred in the following.

In a paper [9] titled "Research on Presence in Virtual Reality: A Survey", the authors quote Matthew Lombard’s slightly more scientific definition of presence: "Presence (a shortened version of the term “telepresence”) is a psychological state of subjective perception in which even though part or all of an individual’s current experience is generated by and/or filtered through human-made technology, part or all of the individual’s perception fails to accurately acknowledge the role of the technology in the experience. Except in the most extreme cases, the individual can indicate correctly that s/he is using the technology, but at some level, and to some degree, her/his perceptions overlook that knowledge and objects, events, entities, and environments are perceived as if the technology was not involved in the experience." In other words, feeling like you’re really there.

Presence is achieved when the involuntary aspects of our reptilian corners of our brains are activated. When the user reaches out to grab the virtual apple, becomes unwilling to step off a plank or feel nervous when walking on rooftops. According to Teo Choong Ching [10], there are four components relevant for feeling present, namely the

1. The Illusion of being in a stable spatial place
2. The Illusion of self-embodiment.
3. The Illusion of Physical Interaction
4. The Illusion of Social Communication

The most relevant component from a the technical aspect in the context of this Technical Report is the first one. This part of presence can be broken down into three broad categories, listed in order of most important to least important for their impact on creating presence: Visual presence, Auditory presence, and sensory or haptic presence.

Technical Requirements for visual presence have been formulated by Valve's ™ R&D Team Brendan Iribe (https://www.roadtovr.com/oculus-shares-5-key-ingredients-for-presence-in-virtual-reality/) from Oculus ™ as well as from experience collected from 3GPP members product development teams:

- Tracking
  - 6 degrees of freedom tracking - ability to track user's head in rotational and translational movements.
  - 360 degrees tracking - track user's head independent of the direction the user is facing.
  - Sub-centimeter accuracy - tracking accuracy of less than a centimeter.
  - Quarter-degree-accurate rotation tracking
  - No jitter - no shaking, image on the display has to stay perfectly still.
  - For room-scale games and experiences, comfortable tracking volume - large enough space to move around and still be tracked of roughly 2m cubes. For seated games/experiences a smaller tracking volume is sufficient.
  - Tracking needs to be done frequently to always be able to operate with the latest XR Viewer Pose. Minimum update rates as discussed above are 1000Hz and beyond. Especially rotational tracking requires high frequency.
- Latency
  - Less than 20 ms motion-to-photon latency - less than 20 milliseconds of overall latency (from the time you move your head to when you see the display change).
  - Minimize the time of pose-to-render-to-photon. Rendering content as quickly as possible. Less than 50ms for render to photon in order to avoid wrongly rendered content. For more details refer to clause 4.2.2.
  - Fuse optical tracking and inertial measurement unit (IMU) data –
  - Minimize loop: tracker → CPU → GPU → display → photons.
  - Minimize interaction delays and age of content depending on the application. For more details see 4.2.2.
- Persistence
  - Low persistence - Turn pixels on and off every 2 - 3 ms to avoid smearing / motion blur. Pixel persistence is the amount of time per frame that the display is actually lit rather than black. “Low persistence” is simply the idea of having the screen lit for only a small fraction of the frame. The reason is that the longer a frame goes on for, the less accurate it will be compared to where you’re currently looking. The brain is receiving the same exact image for the entire frame even as you turn your head whereas in real life your view would constantly adjust.
  - 90 Hz and beyond display refresh rate to eliminate visible flicker.
- Resolution
  - Spatial Resolution: No visible pixel structure - you cannot see the pixels. Low resolution and low pixels per inch (PPI), can cause the user to feel pixelation and feel like he or she is looking through a screen door.
    - In 2014, It was thought at least 1k by 1k pixels per eye would be sufficient.
    - However, in theory, in our fovea, we need about 120 pixels per degree of view to match reality, possibly requiring significantly more than the 1k by 1k, all the way to 8k.
    - In 2019, it is commonly accepted that 2k by 2k per eye provides acceptable quality. Increasing the horizontal resolution to 4k is considered a next step.

        - According to Plamer Luckey, founder of Oculus® Rift™, pixelation will not go away completely until at least 8K resolution (8196 x 4096) per eye is achieved (https://arstechnica.com/gaming/2013/09/virtual-perfection-why-8k-resolution-per-eye-isnt-enough-for-perfect-vr/).

    - Temporal Resolution: According to https://developer.oculus.com/blog/asynchronous-timewarp-examined/, to deliver comfortable, compelling VR that truly generates presence, developers will still need to target a sustained frame rate of 90Hz and beyond, despite the usage of asynchronous time warping.

- Optics

    - Wide Field of view (FOV) is the extent of observable world at any given moment and typically 100 - 110 degrees FOV is needed. For details on FoV, see 3GPP TR 26.918 [2], clause 4.2.2.

    - Comfortable eyebox - the minimum and maximum eye-lens distance wherein a comfortable image can be viewed through the lenses.

    - High quality calibration and correction - correction for distortion and chromatic aberration that exactly matches the lens characteristics. For details on optics, see 3GPP TR 26.918 [2], clauses 4.2.3 and 4.2.4.

For requirements on auditory presence, refer to 3GPP TR 26.918 [2] and [11].

For requirements on sensory and haptics presence, refer for example to [11].

The sense of presence is not only important to VR experiences, but equally so to immersive AR experiences. To achieve Presence in Augmented Reality, seamless integration of virtual content and physical environment is required. Like in VR, the virtual content has to align with user's expectations. For truly immersive AR and in particular MR, it is expected that users cannot discern virtual objects from real objects.

Also relevant for VR and AR, but in particular AR, is not only the awareness for the user as shown in Figure 4.1-5, but also for the environment. This includes:

- Safe zone discovery
- Dynamic obstacle warning
- Geometric and semantic environment parsing
- Environmental lighting
- World mapping

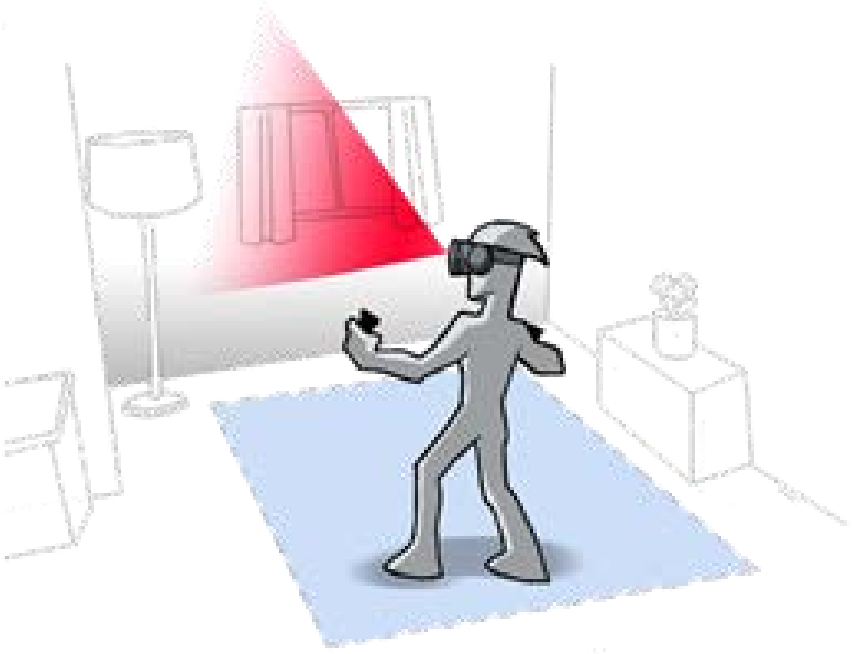


Figure 4.1-5: Environmental Awareness in XR Applications

For AR, to obtain an enhanced view of the real environment, the user may wear a see-through HMD to see 3D computer-generated objects superimposed on his/her real-world view. This see-through capability can be accomplished using either an *optical see-through* or a *video see-through* HMD. Tradeoffs between optical and video see-through HMDs with respect to technological, perceptual, and human factors issues are for example discussed in [18].

4.2.2 Interaction Delays and Age of Content

Beyond the sense of presence and immersiveness, the age of the content and user interaction delay are of the uttermost importance for immersive and non-immersive interactive experiences, i.e. experiences for which the user interaction with the scene impacts the content of scene (such as online gaming).

User interaction delay is defined as the time duration between the moment at which a user action is initiated and the time such an action is taken into account by the content creation engine. In the context of gaming, this is the time between the moment the user interacts with the game and the moment at which the game engine processes such a player response.

Age of content is defined as the time duration between the moment a content is created and the time it is presented to the user. In the context of gaming, this is the time between the creation of a video frame by the game engine and the time at which the frame is finally presented to the player.

The **roundtrip interaction delay** is therefore the sum of the *Age of Content* and the *User Interaction Delay*. If part of the rendering is done on an XR server and the service produces a frame buffer as rendering result of the state of the content, then for raster-based split rendering (as defined in clause 6.2.5) in cloud gaming applications, the following processes contribute to such a delay:

- User Interaction Delay
  - capture of user interaction in game client,
  - delivery of user interaction to the game engine, i.e. to the server (aka network delay),
  - processing of user interaction by the game engine/server,
- Age of Content
  - creation of one or several video buffers (e.g. one for each eye) by the game engine/server,

- encoding of the video buffers into a video stream frame,
- delivery of the video frame to the game client (a.k.a. network delay),
- decoding of the video frame by the game client,
- presentation of the video frame to the user (a.k.a. framerate delay).

For gaming, for example, references [19] and [20] provide interaction delay tolerance thresholds per game type illustrated in Table 4.2.2-1. Note that this Interaction delay refers to the roundtrip interaction delay as defined above.

Table 4.2.2-1: Interaction delay tolerance in traditional gaming (from [19]).

Example game type	Perspective	Delay tolerance
First person shooter (FPS)	First person	100 ms
Role playing game (RPG)	Third person	500 ms
Real-time strategy (RTS)	Omnipresent	1000 ms

In [21], the authors set up a 12 players match of Unreal Tournament 2003™ in a controlled environment. Each player is assigned a specific amount of latency and jitter for the duration of the match. After the match, the players answer a questionnaire about their experience in the game. This study still uses relatively few players, but they are able to conclude that more than 60ms of latency noticeably reduces both performance and experience of this game.

In general, it seems that 60 ms [18], or even 45 ms [22] are better estimates at how much latency is acceptable in the most fast-paced games than the traditionally quoted 100ms value.

In other cases the latency of the content is for example determined by conversational delay thresholds. Typically, around 200ms of latency is acceptable.

Overall, different applications and use cases require different delay requirements and this phenomena should be considered.

The four following categories are considered with respect to roundtrip interaction delay:

- Ultra-Low-Latency applications: roundtrip interaction delay threshold of at most 50ms latency.
- Low-Latency applications: roundtrip interaction delay threshold of at most 100ms latency.
- Moderate latency applications: roundtrip interaction delay threshold of at most 200ms latency.
- Non- critical latency applications: roundtrip interaction delay threshold higher than 200ms latency.

4.3 XR Delivery in 5G System

4.3.1 General Delivery Categories

For the purpose of classifying use cases, this clause defines delivery categories for XR experiences. The following categories are defined:

- *Download*: An XR experience is downloaded and consumed offline without requiring a connection. All media and experience related traffic is downlink.
- (Passive) *Streaming*: The experience is consumed in real-time from a network server. The user does not interact with the XR experience, or if interacting with the XR experience, the interaction is not triggering any uplink traffic. All media related traffic is downlink.
- *Interactive* (Streaming): The experience is consumed in real-time from a network server. The user (or the device automatically) interacts with the XR experience and the interaction changes the delivered content. The traffic is predominantly downlink, but certain traffic is uplink, e.g. XR Viewer Pose information. Different flavours of

interaction exist, for example viewport adaptation, gaming events, etc. Interaction delay requirements may be different, ranging from immersive latency requirements to more static selection interactions.

- *Conversational*: The experience is generated, shared and consumed in real-time from two or more participants with conversational latency requirements.

- *Split* Compute/Rendering: Network functions run an XR engine to support processing and pre-rendering of immersive scenes and the delivery is split into more than one connection, e.g. Split rendering, Edge Computing, etc. The latency and interaction requirements again depend on the use case and the architecture implementation.

A more detailed analysis of architectures in the context of 5G is provided in clause 6.

4.3.2 5G System and Radio Functionalities for XR

The integration of XR applications within the 5G System is approached following the model of 5G Media Streaming as defined in 3GPP TS 26.501 [8]. Assume a 5G-XR Application Provider being an XR Application provider that makes use of 5G System functionalities for its services. For this purpose, it provides a 5G-XR Aware Application on the UE to make use of a 5G-XR client and network functions using network interfaces and APIs, potentially defined in 5G-XR related specifications.

The architecture in Figure 4.3.2-1 represents potential 5G-XR functions within the 5G System (5GS) as defined in 3GPP TS 23.501 [23]. Three main functions are defined:

- 5G-XR AF: An Application Function similar as defined in 3GPP TS 23.501 [23], clause 6.2.10, dedicated to 5G-XR Services.

- 5G-XR AS: An Application Server dedicated to 5G-XR Services.

- 5G-XR Client: A UE internal function dedicated to 5G-XR Services.

In the context of this Technical Report, 5G-XR AF and 5G-XR AS are initially considered Data Network (DN) functions and communicate with the UE via N6, N3 and Uu as defined in 3GPP TS 23.501 [23].

Communication through sidelink PC5 may be an alternative to Uu based communication.

Functions in trusted DNs are trusted by the operator's network as illustrated.. Therefore, AFs in trusted DNs may directly communicate with all 5G Core functions.

Functions in external DNs may only communicate with 5G Core functions via the NEF using N33.

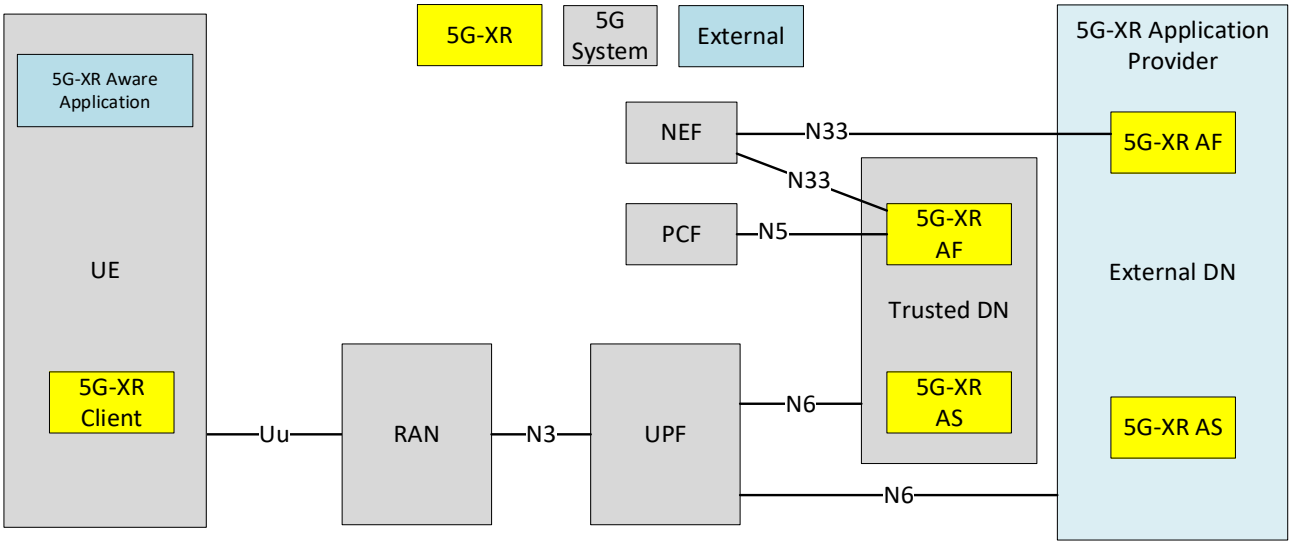


Figure 4.3.2-1: 5G-XR functions integrated in 5G System

NOTE 1: The functions indicated by the yellow filled boxes are in potential scope of stage 3 specifications for 5G-XR. The functions indicated by the grey boxes are defined in 5G System specifications. The functions indicated by the blue boxes are assigned to the applications.

The above architecture is used as a starting point. With XR related functions exclusively assigned to either DN or UE. However, architectural extensions may be identified for the 3GPP system that may benefit from XR applications. Examples include the use of network slicing, edge computing or usage of 5G quality of service.

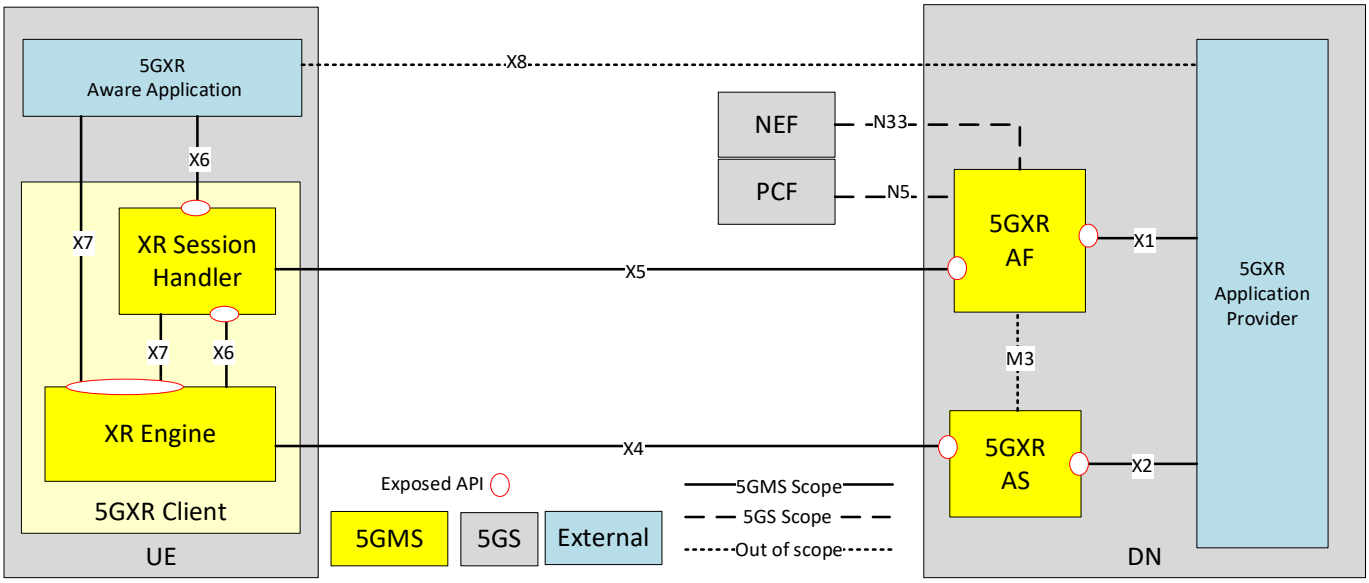


Figure 4.3.2-2: 5G-XR Interfaces and Architecture

A basic XR architecture integrated in 5G is shown in Figure 4.3.2-2.

The following functions may be considered to be defined:

- 5G-XR Client on UE: Receiver of 5G-XR session data that may be accessed through well-defined interfaces/APIs by the 5G-XR Aware Application.
- The 5G-XR Client contains two sub-functions
  - XR Session Handler: A function of the UE that communicates with the 5G-XR AF in order to establish, control and support the delivery of an XR session. The XR Session Handler exposes APIs that can be used by the 5G-XR Aware Application.
  - XR Engine: A function of the UE that communicates with the 5G-XR Application Server in order to get access to XR related data, includes XR relevant functionalities such as sensors and tracking, processes this data and communicates with the XR Session Handler for XR session control.
- 5G-XR Aware Application: The 5G-XR Client is typically controlled by an external XR aware application, e.g. an App, which implements the external application service provider specific logic and enables establishing an XR session. The 5G-XR Aware Application makes use of 5G-XR Client and network functions using interfaces and APIs.
- 5G-XR AS: An Application Server which hosts 5G-XR media and media functions.
- 5G-XR Application Provider: External XR application provider that makes use of 5G-XR client and network functionalities to provide an XR experience to the 5G-XR Aware applications.
- 5G-XR AF: provides various control functions to the XR Session Handler on the UE and/or to the 5G-XR Application Provider. It may relay or initiate a request for different Policy or Charging Function (PCF) treatment or interact with other network functions.

In the context of the above, 5G radio may also be differentiated between 5G Uu and 5G Sidelink/PC5. Uu is the interface between User Equipement (UE) and Radio Access Network (RAN) as defined in 3GPP TS 38.300 [24]. Sidelink is a mode of communication whereby UEs can communicate with each other directly as defined in 3GPP TS 38.300 [24].

4.3.3 Quality-of-Service in 5G

Clause 5.7 of 3GPP TS 23.501 [8] explains the QoS model for 5G. The 5G QoS model is based on QoS Flows. The 5G QoS model supports both:

- QoS Flows that require guaranteed flow bit rate (GBR QoS Flows)
- and QoS Flows that do not require guaranteed flow bit rate (Non-GBR QoS Flows).

The 5G QoS model also supports Reflective QoS (see clause 5.7.5 of 3GPP TS 23.501 [8]).

A QoS Flow ID (QFI) is used to identify a QoS Flow in the 5G System. User Plane traffic assigned to the same QoS Flow within a Protocol Data Unit (PDU) Session receives the same traffic forwarding treatment (e.g. scheduling, admission threshold).

The QFI may be dynamically assigned or may be equal to the 5G QoS Identifier (5QI). A QoS Flow may either be 'GBR', 'Non-GBR' or “Delay Tolerant GBR” depending on its QoS profile and it contains QoS parameters as follows:

- For each QoS Flow, the QoS profile includes the QoS parameters:
  - 5G QoS Identifier (5QI); and
  - Allocation and Retention Priority (ARP).
- For each Non-GBR QoS Flow only, the QoS profile can also include the QoS parameter:
  - Reflective QoS Attribute (RQA).
- For each GBR QoS Flow only, the QoS profile also include the QoS parameters:
  - Guaranteed Flow Bit Rate (GFBR) - uplink (UL) and downlink (DL); and
  - Maximum Flow Bit Rate (MFBR) - UL and DL; and
- In the case of a GBR QoS Flow only, the QoS profile can also include one or more of the QoS parameters:
  - Notification control;
  - Maximum Packet Loss Rate - UL and DL

The one-to-one mapping of standardized 5QI values to 5G QoS characteristics is specified in table 5.7.4-1 of 3GPP TS 23.501 [8] and shown below in Table 4.3.3-1.

5QI values potentially relevant for XR applications in the context of this Technical Report are highlighted in *italics*.

Table 4.3.3-1: Standardized 5QI to QoS characteristics mapping (identical to Table 5.7.4.1-1 in 3GPP TS 23.501 [10])

5QI Value	Resource Type	Default Priority Level	Packet Delay Budget	Packet Error Rate	Default Maximum Data Burst Volume (NOTE 2)	Default Averaging Window	Example Services
1	*GBR (NOTE 1)*	*20*	*100 ms*	*10^{-2}*	*N/A*	*2000 ms*	*Conversational Voice*
2		*40*	*150 ms*	*10^{-3}*	*N/A*	*2000 ms*	*Conversational Video (Live Streaming)*
3		*30*	*50 ms*	*10^{-3}*	*N/A*	*2000 ms*	*Real Time Gaming, V2X messages Electricity distribution – medium voltage, Process automation - monitoring*
4		*50*	*300 ms*	*10^{-6}*	*N/A*	*2000 ms*	*Non-Conversational Video (Buffered Streaming)*
65		7	75 ms	10^{-2}	N/A	2000 ms	Mission Critical user plane Push To Talk voice (e.g., MCPTT)
66		20	100 ms	10^{-2}	N/A	2000 ms	Non-Mission-Critical user plane Push To Talk voice
67		15	100 ms	10^{-3}	N/A	2000 ms	Mission Critical Video user plane
75		25	50 ms	10^{-2}	N/A	2000 ms	V2X messages
5	Non-GBR (NOTE 1)	10	100 ms	10^{-6}	N/A	N/A	IMS Signalling
6		*60*	*300 ms*	*10^{-6}*	*N/A*	*N/A*	*Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive video, etc.)*
7		*70*	*100 ms*	*10^{-3}*	*N/A*	*N/A*	*Voice, Video (Live Streaming) Interactive Gaming*
8		*80*	*300 ms*	*10^{-6}*	*N/A*	*N/A*	*Video (Buffered Streaming) TCP-based (e.g., www, e-mail, chat, ftp, p2p file sharing, progressive*
9		90					video, etc.)
69		5	60 ms	10^{-6}	N/A	N/A	Mission Critical delay sensitive signalling (e.g., MC-PTT signalling)
70		55	200 ms	10^{-6}	N/A	N/A	Mission Critical Data (e.g. example services are the same as QCI 6/8/9)
79		65	50 ms	10^{-2}	N/A	N/A	V2X messages
80		*68*	*10 ms*	*10^{-6}*	*N/A*	*N/A*	*Low Latency eMBB applications Augmented Reality*
82	Delay Critical GBR	19	10 ms (NOTE 4)	10^{-4}	255 bytes	2000 ms	Discrete Automation (see TS 22.261 [2])
83		22	10 ms (NOTE 4)	10^{-4}	1358 bytes (NOTE 3)	2000 ms	Discrete Automation (see TS 22.261 [2])
84		24	30 ms (NOTE 6)	10^{-5}	1354 bytes	2000 ms	Intelligent transport systems (see TS 22.261 [2])
85		21	5 ms (NOTE 5)	10^{-5}	255 bytes	2000 ms	Electricity Distribution- high voltage (see TS 22.261 [2])

NOTE 1: A packet which is delayed more than PDB is not counted as lost, thus not included in the PER.
NOTE 2: It is required that default MDBV is supported by a PLMN supporting the related 5QIs.
NOTE 3: This MDBV value is set to 1354 bytes to avoid IP fragmentation for the IPv6 based, IPSec protected GTP tunnel to the 5G-AN node (the value is calculated as in Annex C of TS 23.060 [56] and further reduced by 4 bytes to allow for the usage of a GTP-U extension header).
NOTE 4: A delay of 1 ms for the delay between a UPF terminating N6 and a 5G-AN should be subtracted from a given PDB to derive the packet delay budget that applies to the radio interface.
NOTE 5: A delay of 2 ms for the delay between a UPF terminating N6 and a 5G-AN should be subtracted from a given PDB to derive the packet delay budget that applies to the radio interface.
NOTE 6: A delay of 5 ms for the delay between a UPF terminating N6 and a 5G-AN should be subtracted from a given PDB to derive the packet delay budget that applies to the radio interface.

The applicability of 5QI and potential gaps for XR Services over 5G are analysed further in this Technical Report.

4.3.4 5G Media Delivery

In the context of this Technical Report and the delivery options identified in clause 4.3.3.1, the first three basic delivery types – download, passive streaming and interactive streaming – are most suitably mapped to 5G Media Streaming as defined in 3GPP TS 26.501 [8] and associated stage 3 specifications. The applicability of 5G Media Streaming for XR applications and potential necessary extensions are identified in clauses 5, 6 and 7 of this Technical Report.

Conversational services are most suitably mapped to the Multimedia Telephony Service for IMS (MTSI) as defined in 3GPP TS 22.173 [24] with focus on XR media handling (e.g. signalling, transport, codecs, formates) when using 3GPP access, in particular 5G radio technologies. It is expected that the media handling of MTSI clients as defined in 3GPP TS 26.114 [25] may be suitably extended in order to support XR applications and services.

4.3.5 Edge Computing

Beyond the use of Application Servers as defined in 5G Media Streaming today, the 5G-XR application may benefit from additional processing in the edge. In an example, as shown in Figure 4.3.5-1, an edge platform may be offered by the 5G network operator to support XR services served from the content provider or from the cloud.

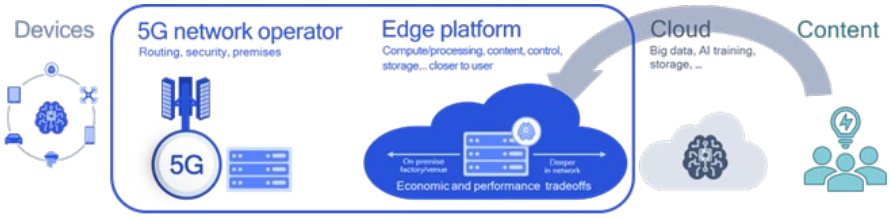


Figure 4.3.5-1 Cloud and Edge Processing

In the context of Release-17, 3GPP work is ongoing in order to identify the integration of edge processing in 5G systems. 3GPP TR 23.748 [28] defines the necessary modifications to 5G system architecture to enhance Edge Computing. This work is currently in study phase, defining key issues and scope for Release-17. Specifically, this study is investigating mechanisms to discover connectivity to available Edge Computing resources (e.g. using DNS), mobility improvements for both UE consuming Edge Computing services and for Edge Application Servers, and for network capability exposure towards the Edge Application Server.

In addition, in 3GPP TR 23.758 [27] and 3GPP TS 23.558 [29] a new set of application layer interfaces for Edge Computing are identified that may potentially be useful for integration of Edge Computing. Specifically, the interfaces will enable application-layer discovery of Edge Application Servers, capability exposure towards the Edge Application Server, and procedures for onboarding, registration, and lifecycle management of Edge Applications.

The activities detailed in the present clause are intended to be application-neutral (i.e. to provide generic solutions for any use of Edge Computing platforms). The media aspects for using Edge Computing are not identified in these studies and information in the present Technical Report may be beneficial to contribute to Edge Computing for media processing. In particular, split Compute/Rendering architectures are not yet specified in the 5G System architecture beyond those being part of a 5G-XR aware application. Integration of computational resources into the 5G System as part of Edge Computing functionalities are currently under study in 3GPP. The present Technical Report serves to identify potentially relevant functions for XR applications when using Edge Computing and rendering.

4.4 XR Engines and Rendering

4.4.1 Introduction

XR engines provide a middleware that abstracts hardware and software functionalities for developers of XR applications. In the market as understood when initially writing this report, such engines are predominantly based on proprietary and commercial solutions, but with a trend towards providing standardized abstraction layers and APIs, notably provided by Khronos' OpenXR [16] as well as W3C's WebXR [17]. An overview of the landscape as seen by the OpenXR community is shown in Figure 4.4.1-1.

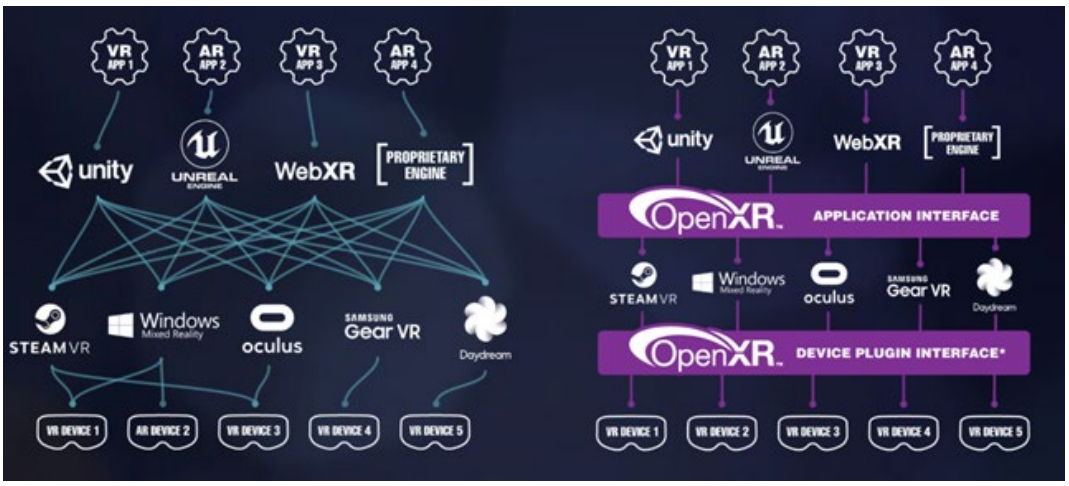


Figure 4.4.1-1: XR engine and ecosystem landscape today and in the future as seen by OpenXR © Khronos

An XR engine is a software-development environment designed for people to build XR experiences such as games and other XR applications. The core functionality typically provided by XR engines include a rendering engine ("renderer") for 2D or 3D graphics, a physics engine or collision detection (and collision response), sound, scripting, animation, artificial intelligence, networking, streaming, memory management, threading, localization support, scene graph, and may include video support.

Typical components are summarized below

- Rendering engine: This engine basically provides the functionalities as documented in clause 4.2.2 with a set of well defined APIs. In summary, the rendering engine generates animated 3D graphics by any of a number of methods (rasterization, ray-tracing etc.). Instead of being programmed and compiled to be executed on the CPU or GPU directly, most often rendering engines are built upon one or multiple rendering application programming interfaces (APIs), such as Direct3D, OpenGL, or Vulkan which provide a software abstraction of the graphics processing unit (GPU).

- Audio engine: The audio engine is the component which consists of algorithms related to the loading, modifying and output of sound through the client's speaker system. At a minimum it is able to load, decompress and play sound files. More advanced audio engines can calculate and produce such configurations as Doppler effects, echoes, pitch/amplitude adjustments, oscillation, etc. The audio engine can perform calculations on the CPU, or on a dedicated ASIC. Abstraction APIs, such as OpenAL, SDL audio, XAudio 2, Web Audio, etc. are available.

- Physics engine: The physics engine is responsible for emulating the laws of physics realistically within the XR application. Specifically, it provides a set of functions for simulating physical forces and collisions, acting on the various objects within the scene at run time.

- Artificial intelligence *(AI)*: AI is usually outsourced from the main XR engine into a special module. XR applications may implement very different AI systems, and thus, AI is considered to be specific to the particular XR application for which it is created.

One of the major game engines used to create several notable games such as Fortnite ™, PlayerUnknown's Battlegrounds ™, and Life is Strange 2 ™, is the *Unreal Engine 4* ™. Another game engine with significant share is the *Unity* ™ engine. This engine is the one behind games such as Rust ™, Subnautica ™, and Life is Strange Before the Storm ™. Unity™ is a cross-platform XR engine developed by Unity Technologies, first announced and released in June 2005. A component of Unity are scriptable rendering pipelines for developers to create high-end graphics including high-end ones for consoles and PC experiences, as well as the lightweight ones for mobile, virtual reality, augmented reality, and mixed reality.

The Unreal Engine was first showcased in the 1998 first-person shooter game Unreal. Although initially developed for first-person shooters, it is used in a variety of other genres, including platformers, fighting games, MMORPGs, and other RPGs.

Figure 4.4.1-2 provides an overview of typical CPU and GPU operations for XR applications.

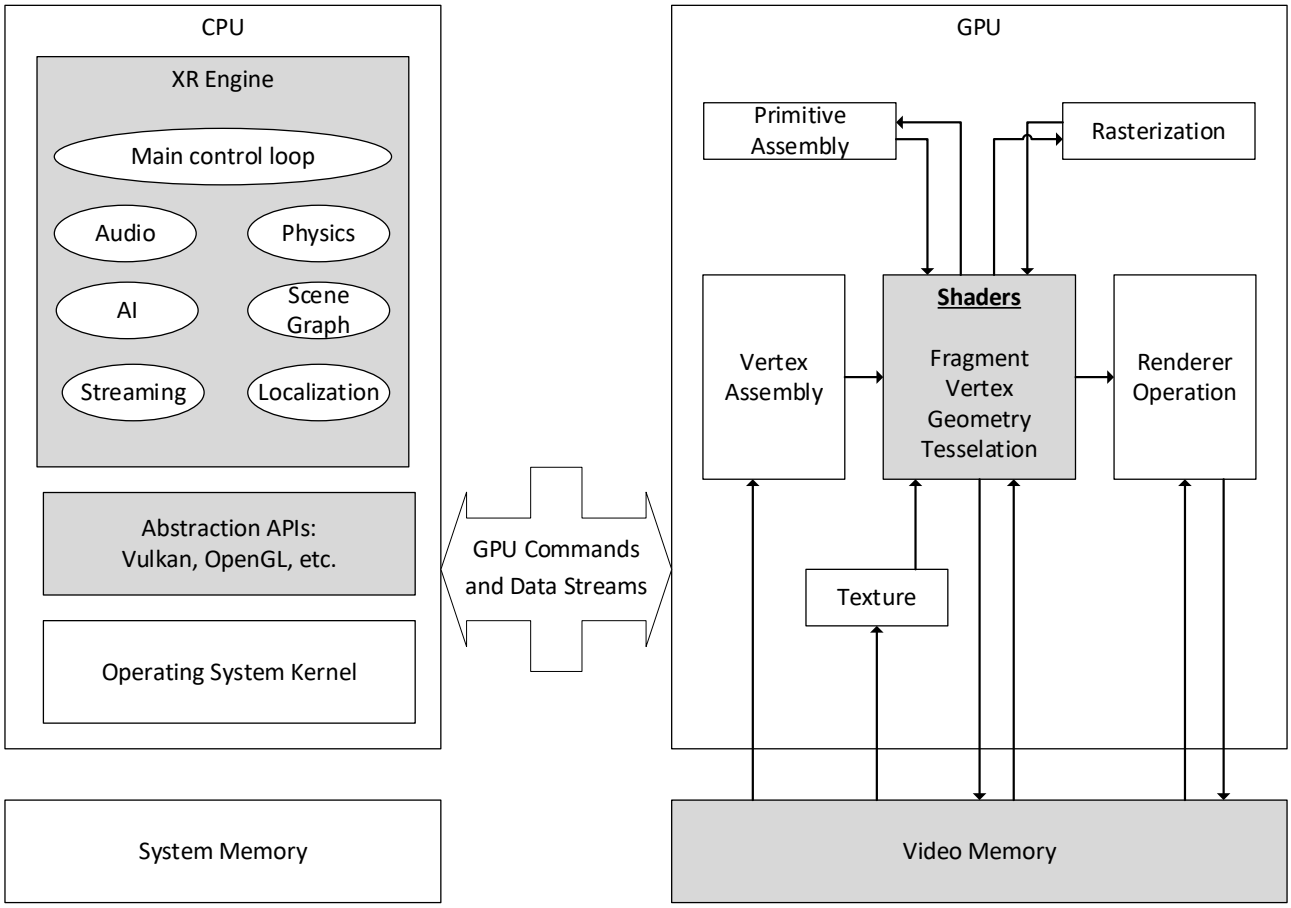


Figure 4.4.1-2: CPU and GPU operations for XR applications

As mentioned above, key aspects of such XR engines and abstraction layers is the integration of advanced functionalities for new XR experiences including video, sound, scripting, networking, streaming, localization support, and scene graphs. By well-defined APIs, XR engines may also be distributed, where part of the functionality is hosted in the network on an XR Server and other parts of the functionality are carried out in the XR device.

GPU operations and rendering is dealt with in clause 4.4.2.

In the remainder of this Technical Report, the term *XR engine* is used to provide any type of typical XR functionalities as mentioned above. A key issue is the functional integration of potentially 3GPP defined technologies, including well defined APIs and interfaces for the usability and benefit of XR application developers.

4.4.2 Briefly on Rendering Pipelines

Rendering or graphics pipelines are basically built by a sequence of shaders that operate on different buffers in order to create a desired output. Shaders are a type of computer programs that were originally used for shading (the production of appropriate levels of light, darkness, and colour within an image), but which now perform a variety of programmable functions in various fields of computer graphics as well as image and video processing.

The input to the shaders is handled by the application, which decides what kind of data each stage of the rendering pipeline should operate on. Typically this data is 3D assets consisting of geometric primitives and material components. The application controls the rendering by providing the shaders with instructions describing how models should be transformed and projected on a 2D surface.

Generally there are 2 types of rendering pipelines (i) rasterization rendering and (ii) ray-traced rendering (a.k.a. ray-tracing) as shown in Figure 4.2-1.

- *Rasterization* is the task of taking an image described in a vector graphics format (shapes) and converting it into a raster image (a series of pixels, which, when displayed together, create the image which was represented via

shapes). The rasterized image may then be displayed on a computer display, video display or printer, stored in a bitmap file format, or processed and encoded by regular 2D image and video codecs.

- *Ray tracing* heavy pipelines are typically considered as more computationally expensive thus less suitable for real-time rendering. However, in recent years real-time ray tracing has leaped forward due to improved hardware support and advances in ray tracing algorithms and post-processing.

However, there are several flavors to each type of rendering and they may in some cases be combined to generate hybrid pipelines. Generally both pipelines process similar data; the input consists of geometric primitives such as triangles and their material components. The main difference is that rasterization focuses to enable real-time rendering at a desired level of quality, whereas ray tracing is likely to be used to mimic light transmission to produce more realistic images.

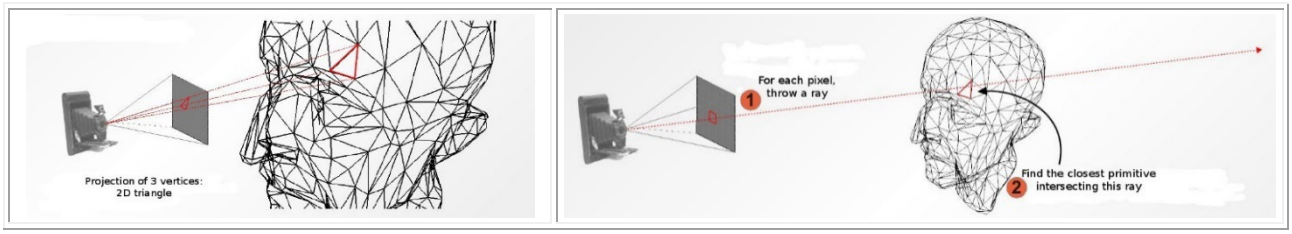


Figure 4.4.2-1 Rasterized (left) and ray-tracing based (right) rendering

4.4.3 Real-time 3D Rendering

3D rendering is the process of converting 3D models into 2D images to be presented on a display. 3D rendering may include photorealistic or non-photorealistic styles. Rendering is the final process of creating the actual 2D image or animation from the prepared scene, i.e. creating the viewport. Rendering ranges from the distinctly non-realistic wireframe rendering through polygon-based rendering, to more advanced techniques such as ray tracing. The 2D rendered viewport image is simply a two dimensional array of pixels with specific colours.

Typically, rendering needs to happen in real-time for video and interactive data. Rendering for interactive media, such as games and simulations, is calculated and displayed in real-time, at rates of approximately 20 to 120 frames per second. The primary goal is to achieve a desired level of quality at a desired minimum rendering speed. The impact of the frame rate for the rendering pipeline is discussed in details in clause 4.2.2. The rapid increase in computer processing power and in the number of new algorithms has allowed a progressively higher degree of realism even for real-time rendering. Real-time rendering is often based on rasterization and aided by the computer's GPU.

Animations for non-interactive media, such as feature films and video, can take much more time to render. Non real-time rendering enables use of brute-force ray tracing to obtain a higher image quality.

However, in the context of XR in this Technical Report, the assumption of rendering is to be real-time to react to updated XR pose information, updates in the scene as produced, and so on.

4.4.4 Network Rendering and Buffer Data

In several applications, rendering or pre-rendering is not exclusively carried out in the device GPU, but assisted or split across the network. If this is the case, then the following aspects matter:

- The type of buffers that are pre-rendered/baked in the network. Typical buffer formats are summarized below.
- The format of the buffer data. Again, some data is collected below.
- The number of parallel buffers that are handled
- Specific delay requirements of each of the buffers
- The dimensions of the buffers in terms of size and time
- The ability to compress the buffers using conventional video codecs.

Typical buffers are summarized in the following:

- *Vertex Buffer*: A rendering resource managed by a rendering API holding vertex data. May be connected by primitive indices to assemble rendering primitives such as triangle strips.
- *Depth Buffer*: A bitmap image holding depth values (either a Z buffer or a W buffer), used for visible surface determination, during rasterization of 3D scenes.
- *Texture Buffer*: A region of memory (or resource) used as both a render target and a texture map. A texture map is defined as an image/rendering resource used in texture mapping, applied to 3D models and indexed by UV mapping for 3D rendering. Texture/Image represents a set of pixels. Texture buffers have assigned parameters to specify creation of an Image. It can be 1D, 2D or 3D, can have various pixel formats (like R8G8B8A8_UNORM or R32_SFLOAT) and can also consist of many discrete images, because it can have multiple array layers or MIP levels (or both). As an example, detailed formats for Vulkan are provided here:
  - https://www.khronos.org/registry/vulkan/specs/1.0/html/chap33.html
  - https://vulkan.lunarg.com/doc/view/1.0.30.0/linux/vkspec.chunked/ch31s03.html
- *Frame Buffer*: a region of memory containing a bitmap that drives a video display. These frame buffers are in raster formats, i.e. they are the result of rasterization. It is a memory buffer containing a complete frame of data. Frame buffers are supported by swap chains being a series of virtual frame buffers utilized by the graphics card and graphics API for frame rate stabilization and several other functions. In every swap chain there are at least two buffers. The first frame buffer, the screen buffer, is the buffer that is rendered to the output of the video card. The remaining buffers are known as backbuffers. Each time a new frame is displayed, the first backbuffer in the swap chain takes the place of the screenbuffer, this is called presentation or swapping. A variety of other actions may be taken on the previous screenbuffer and other backbuffers (if they exist). The screen buffer may be simply overwritten or returned to the back of the swap chain for further processing. The action taken is decided by the client application and is API dependent.
- *Uniform Buffer*: A Buffer Object that is used to store uniform data for a shader program. It can be used to share uniforms between different programs, as well as quickly change between sets of uniforms for the same program object. A uniform is a global Shader variable declared with the "uniform" storage qualifier. These act as parameters that the user of a shader program can pass to that program. Uniforms are so named because they do not change from one shader invocation to the next within a particular rendering call.

In MPEG, work has started on integration of timed media into XR scenes in order to provide input to rendering buffers through network APIs, for example by retrieving 2D or 3D compressed data.

4.5 2D Compression Technologies

4.5.1 Core Compression Technologies

This clause provides an overview of core 2D video compression technologies that are available on mobile platforms as well as their performance. For power-efficient and best performance, encoding and decoding is preferably exclusively carried out in hardware. This clause reviews the 3GPP specifications, actual hardware availability as well as the performance of codecs.

As of today, two codecs are prominently referenced and available, namely H.264/AVC [30] and H.265/HEVC [31]. Both codecs are defined as part of the TV Video Profiles in 3GPP TS 26.116 [32] and are also the foundation of the VR Video Profiles in 3GPP TS 26.118 [3]. The highest defined profiles are:

- H.264/AVC Progressive High Profile Level 5.1 [30] with the following additional restrictions and requirements:
  - the maximum VCL Bit Rate is constrained to be 120Mbps with cpbBrVclFactor and cpbBrNalFactor being fixed to be 1250 and 1500, respectively.
  - the bitstream does not contain more than 10 slices per picture
- H.265/HEVC Main-10 Profile Main Tier Profile Level 5.1 [31] without any restrictions

These profiles and levels basically permit the delivery of video formats up to 4K at 60 frames per second. In modern mobile CPUs, the above profile/level combinations are supported, and recently even extended to support 8K video.

An overview of typical coding performance is provided in Table 4.5-1.

Table 4.5-1: Expected Video coding standards performance and bitrate target

Codec	Coding performance (Random-Access)		Targeted bitrate (Random Access)
	Objective	Subjective	
AVC			4k: ▪ Statmux: 20-25 Mbps ▪ CBR: 35 - 50 Mbps 8k: ▪ CBR: 80 - 100 Mbps ▪ High quality: 100 - 150 Mbps [33][34][35]
HEVC	-40% vs AVC [33][34][35]	-60% vs AVC [33][34][35]	4k: ▪ Statmux: 10-13 Mbps ▪ CBR: 18-25 Mbps 8k: ▪ CBR: 40-56 Mbps ▪ High quality: 80-90 Mbps [33][34][35]

A more detailed analysis of video codec performance is FFS.

Work on video compression technologies beyond the capabilities of HEVC [31] are continued by the MPEG/ITU. For example, the Joint Video Exploration Team (JVET) initiated the work on the development of a new video coding standard, to be known as Versatile Video Coding (VVC). In addition, MPEG started working on a new video coding standard to be known as MPEG-5 Essential Video Coding (EVC) in January 2019. Also noteworthy is the improvement of encoders over time even for existing standards which also leads to bitrate reductions at the same quality.

Based on this information it can be expected that within the time frame until 2025, video compression technology will permit bitrate reductions by a factor of 50% compared to what is today possible with HEVC [31].

On top of regular lossy video compression algorithms, low-latency, low-complexity and near lossless codecs are important for certain applications. As an example, JPEG XS is a recent standard for visually lossless low-latency lightweight image coding. According to https://jpeg.org/static/whitepapers/jpeg-xs-whitepaper.pdf, such a codec permits simple yet efficient coding, keeps latency and complexity very low and at the same time achieves visually lossless quality at compression ratios up to 10:1.

Furthermore, for XR formats beyond regular 2D, two different approaches are taken in the compression

1) usage of existing 2D codecs and providing pre- and post-processing in order to convert the signals to 3D signals
2) usage of dedicated compression technologies for specific formats.

More details on these issues are discussed in clause 4.6.

4.5.2 Format and Parallel Decoding Challenges

In XR type of applications, when buffers are processed by rendering engines, existing video codecs may be used to efficiently compress them when they need to be transmitted over the network. As typically a huge amount of data is exchanged and operation needs to be done in a power-efficient manner in constraint environments (see clause 4.8), XR applications rely on existing video codecs on mobile platforms, for example those codecs defined in 3GPP specifications (see clause 4.5.1). While serving an immediate need and providing a kickstart for XR type of services, such video codecs may not be fully suitable for XR applications for different reasons, some of them listed below.

First of all, the formats of the buffers in XR and graphics applications may be different and more variety exists, see clause 4.4.4. Also in certain case, not only textures need to be supported, but also 3D formats, see clause 4.6.

Beyond this, XR applications may require that multiple buffers are served and synchronized in order to render an XR experience. This results in requirements for parallel decoding of multiple streams for multiple buffers (texture, geometry, etc.) as well as multiple objects. In many cases these buffers need to be made available to the rendering engine in a synchronized manner to ensure the highest quality of the rendered scene. Furthermore, the amount of streams and data to be processed may vary heavily over the period of an XR session and requires flexible video decoding architectures, also taking into account efficient and low-latency processing.

As an example, MPEG is addressing several of these challenges as part of their MPEG-I project on immersive media coding. In particular, for the variety of applications, a flexible and powerful hardware based decoding and processing architecture is desirable.

4.6 3D and XR Visual Formats

4.6.1 Introduction

This clause introduces 3D and XR visual formats. Both, static images as well as video formats are introduced. In all cases it is assumed that the visual signal is provided as a sequence of pictures with a specific frame rate in frames per second. The chosen frame rate may be a matter of the production of the video, or it may be based on requirements due to interactions with the content, for example in case of conversational applications or when using split rendering.

4.6.2 Omnidirectional Visual Formats

4.6.2.1 Introduction

Omnidirectional formats have been introduced in 3GPP TS 26.118 [3], clause 4.1, as well as in 3GPP TR 26.928 [2], clause 4.2.5.

4.6.2.2 Definition

Omnidirectional visual signals are represented in a spherical coordinate space in angular coordinates (ϕ,θ). The viewing is from the origin of the sphere, looking outward. Even though a spherical coordinate is generally represented by using radius, elevation, and azimuth, it assumes that a unit sphere is used for capturing and rendering. Thus, a location of a point on the unit sphere is identified by using the sphere coordinates azimuth (ϕ) and elevation (θ).

For video, such a centre point may exist for each eye, referred to as *stereo* signal, and the video consists of three colour components, typically expressed by the luminance (Y) and two chrominance components (U and V).

According to 3GPP TS 26.118 [3], clause 4.1.3, mapping of a spherical picture to a 2D texture signal is illustrated in Figure 4.6.2-1. The most commonly used mapping from spherical to 2D is the equirectangular projection (ERP) mapping. The mapping is bijective, i.e. it may be expressed in both directions.

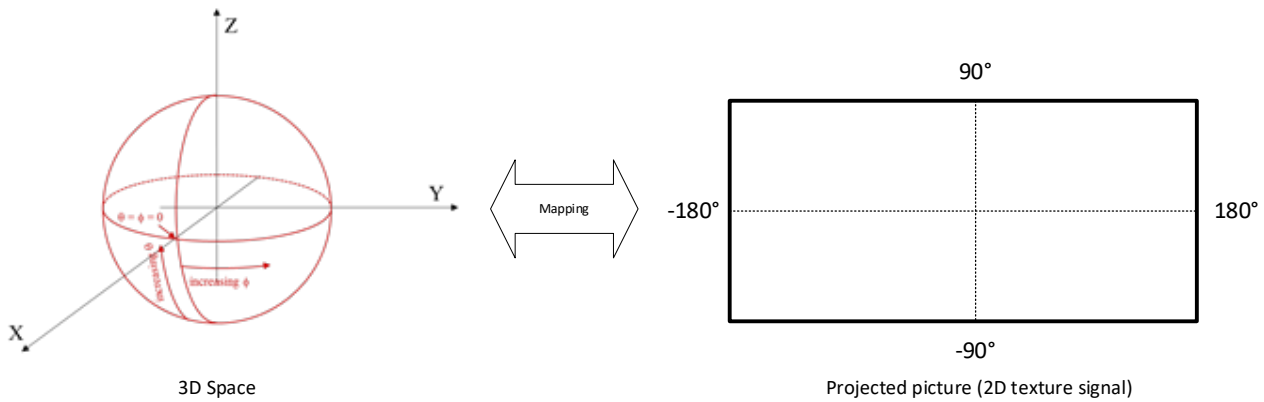


Figure 4.6.2-1: Examples of Spherical to 2D mappings

Assume a 2D texture with `pictureWidth` and `pictureHeight`, being the width and height, respectively, of a monoscopic projected luma picture, in luma samples and the center point of a sample location `(i,j)` along the horizontal and vertical axes, respectively, then for the *equirectangular* projection the sphere coordinates (ϕ,θ) for the luma sample location, in degrees, are given by the following equations:

$$\phi = (0.5 - i \div pictureWidth) * 360$$
$$\theta = (0.5 - j \div pictureHeight) * 180$$

Whereas ERP is commonly used for production formats, other mappings may be applied, especially for distribution. For more details on projection formats, refer to 3GPP TR 26.918 [2], clause 4.2.5.4.

4.6.2.3 Production and Capturing Systems

For production, capturing and stitching of spherical content, refer to 3GPP TR 26.918 [2], clauses 4.2.5.2 and 4.2.5.3.

4.6.2.4 Rendering

Rendering of spherical content depends on the field of view (FoV) of a rendering device. The pose together with the field of view of the device enables the system to generate the user viewport, i.e., the presented part of the content at a specific point in time. According to 3GPP TS 26.118 [3], the renderer uses the projected texture signals and rendering metadata (projection information) and provides a viewport presentation taking into account the viewport and possible other information. With the pose, a user viewport is determined by identifying the horizontal and vertical fields of view of the screen of a head-mounted display or any other display device to render the appropriate part of decoded video or audio signals. For video, textures from decoded signals are projected to the sphere with rendering metadata received from the file decoder. During the texture-to-sphere mapping, a sample of the decoded signal is remapped to a position on the sphere.

Related to the generic rendering approaches in clause 4.2.2, the following steps are part of the rendering spherical media:

- Generating a 3D Mesh (set of vertexes linked into triangles) based on the projection metadata. The sphere is mapped to a mesh and the transformation of the mesh is dynamically updated based on the updated projection metadata.
- Mapping each vertex to a position on a 2D texture. This is again done using the available projection metadata.
- Rotating the camera to match the user's head orientation. This is based on the available pose information.
- Computing the viewport by using computer graphic algorithms as discussed in details in clause 4.2.2

4.6.2.5 Compression, Storage and Data Formats

According to 3GPP TS 26.118 [3], clause 4.1.3, commonly used video encoders cannot directly encode spherical videos, but only 2D textures. However, there is a significant benefit to reuse conventional 2D video encoders. Based on this, Figure 4.6.2-2 provides the basic video signal representation in the context of omnidirectional video in the context of the present document. By pre-processing, the spherical video is mapped to a 2D texture. The 2D texture is encoded with a regular 2D video encoder and the VR rendering metadata (i.e. the data describing the mapping from the spherical coordinate to the 2D texture) is encoded and provided along with the video bitstream, such that at the receiving end the inverse process can be applied to reconstruct the spherical video.

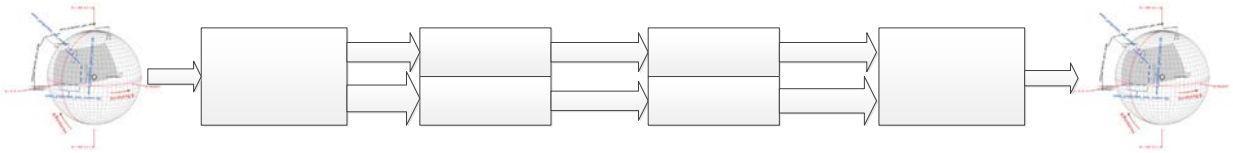


Figure 4.6.2-2: Video Signal Representation

Compression, storage and data formats are defined for example in 3GPP TS 26.118 [3] as well as in ISO/IEC 23090-2 [37]. This includes viewport-independent and viewport-dependent compression formats. A principle overview of different approaches is documented in 3GPP TR 26.918 [2], clause 4.2.5.6.

4.6.2.6 Quality and Bitrate considerations

According to clause 4.2.1, 1k by 1k per eye is a minimum for the signal in the viewport, and for stereoscopic rendering this results in basically a signal for 2k by 1k, typically at a frame rate of 50 or 60fps.With current codecs according to clause 4.5.1, the pure viewport data can be represented with around 4-10 Mbit/s. However, a viewport typically only covers around 100 degree horizontal and 60 degree vertical. Hence, to present a full omnidirectional presentation, about 20 times more data may be necessary, leading to 80 – 200 Mbit/s. Viewport-dependent coding and delivery, in particular tiling, can support to reduce the required bitrates.

4.6.2.7 Applications

For use cases and applications, see 3GPP TR 26.918 [2], clause 5.

4.6.3 3D Meshes

4.6.3.1 Introduction

A polygon mesh is a collection of vertices, edges and faces that defines the shape of a polyhedral object in 3D computer graphics and solid modeling. The faces usually consist of triangles (triangle mesh), quadrilaterals (quads), or other simple convex polygons (n-gons), since this simplifies rendering, but may also be more generally composed of concave polygons, or even polygons with holes.

4.6.3.2 Definition

Objects created with polygon meshes are represented by different types of elements. These include vertices, edges, faces, polygons and surfaces as shown in Figure 4.6.3-1. In many applications, only vertices, edges and either faces or polygons are stored.

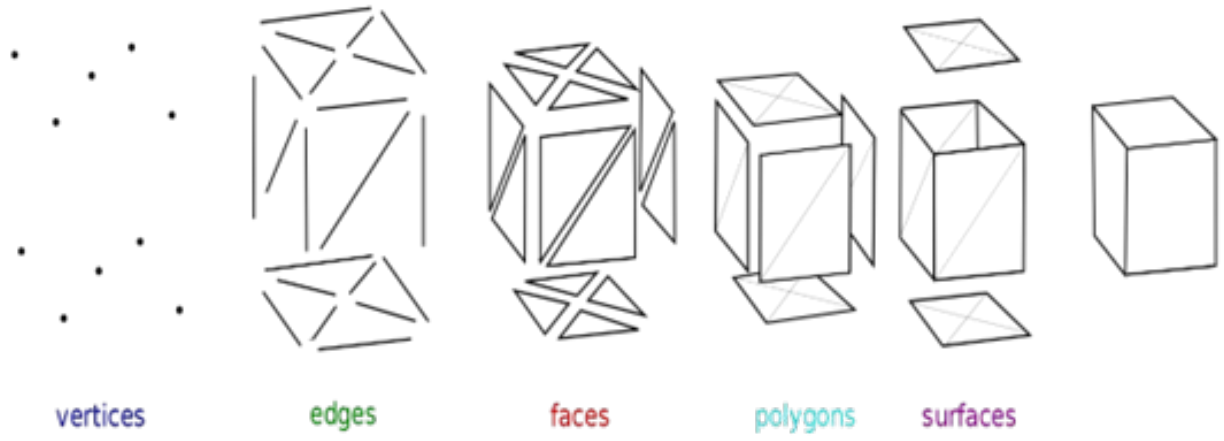


Figure 4.6.3-1: Elements necessary for mesh representations ©Wikipedia (Mesh_overview.jpg: The original uploader was Rchoetzlein at English Wikipedia.derivative work: Lobsterbake [CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)])

Polygon meshes are defined by the following elements:

- ***Vertex***: A position in 3D space defined as `(x,y,z)` along with other information such as colour (r,g,b), normal vector and texture coordinates.
- ***Edge***: A connection between two vertices.
- ***Face***: A closed set of edges, in which a triangle face has three edges, and a quad face has four edges. A polygon is a coplanar set of faces. In systems that support multi-sided faces, polygons and faces are equivalent. Mathematically a polygonal mesh may be considered as an unstructured grid, or undirected graph, with additional properties of geometry, shape and topology.

- ***Surfaces***: or smoothing groups, are useful, but not required to group smooth regions.
- ***Groups***: Some mesh formats contain groups, which define separate elements of the mesh, and are useful for determining separate sub-objects for skeletal animation or separate actors for non-skeletal animation.
- ***Materials***: defined to allow different portions of the mesh to use different shaders when rendered.
- ***UV coordinates***: Most mesh formats also support some form of UV coordinates which are a separate 2D representation of the mesh "unfolded" to show what portion of a 2-dimensional texture map to apply to different polygons of the mesh. It is also possible for meshes to contain other such vertex attribute information such as colour, tangent vectors, weight maps to control animation, etc (sometimes also called channels).

4.6.3.3 Production and Capturing Systems

Meshes are commonly produced by many different graphics engines, computer games, and so on. For more details, see also clause 4.6.7.

4.6.3.4 Rendering

Meshes can be rendered directly on GPUs that are highly optimized for mesh-based rendering.

4.6.3.5 Storage and Data Formats

4.6.3.5.1 Introduction

Polygon meshes may be represented in a variety of ways, using different methods to store the vertex, edge and face data such as face-vertex, winged, half or quad-edge meshes, corner-table or vertex-vertex meshes. Different formats also store other per vertex and materials related data in different ways. Each of the representations have particular advantages and drawbacks. The choice of the data structure is governed by the application, the required performance, the size of the data, and the operations to be performed. For example, it is easier to deal with triangles than general polygons. For certain operations it is necessary to have a fast access to topological information such as edges or neighboring faces; this requires more complex structures such as the winged-edge representation. For hardware rendering, compact, simple structures are needed and are as such commonly incorporated into low-level rendering APIs such as DirectX and OpenGL.

Many different formats for storage and data formats exist for storing polygon mesh data, as an example PLY-format is introduced below, because it provides 3D data in a human readable format. In practice PLY-format is rarely used in real-time rendering. Typically the data format is configured to the need of the 3D engine and as such the landscape is littered with proprietary 3D formats. 3D formats may be roughly divided in two categories, run time friendly formats such as glTF and formats which enable transferring 3D assets between systems like PLY.

4.6.3.5.2 PoLYgon (PLY) File Format

The PoLYgon (PLY) format (see http://paulbourke.net/dataformats/ply/) is used to describe a 3D object as a list of vertices, faces and other elements, along with associated attributes. A single PLY file describes exactly one 3D object. The 3D object may be generated synthetically or captured from a real scene. Attributes of the 3D object elements that might be stored with the object include: colour, surface normals, texture coordinates, transparency, etc. The format permits one object to have different properties for the front and back of a polygon.

The PLY does not intend to act as a Scene Graph, so it does not include transformation matrices, multiple 3D objects, modeling hierarchies, or object sub-parts. A typical PLY object definition is a list of `(x,y,z,r,g,b)` triples for vertices and their colour attributes `(r,g,b)`, so they represent a point cloud. It may also include a list of faces that are described by indices into the list of the vertices. Vertices and faces are primary elements of the 3D object representation.

PLY allows applications to create new attributes that are attached to the elements of an object. New attributes are appended to the list of attributes of an element, in a way to maintain backwards compatibility. Attributes that are not understood by a parser are simply skipped.

Furthermore, PLY allows for extensions to create new element types and their associated attributes. Examples of such elements could be materials (ambient, diffuse and specular colours and coefficients). New elements can also be discarded by programs that do not understand them.

A PLY file is structured as follows:

```
Header
Vertex List
Face List
(lists of other elements)
```

The header is a human-readable textual description of the PLY file. It contains a description of each element type, including the element's name (e.g. "vertex"), how many of such elements are in the object, and a list of the various attributes associated with the element. The header also indicates whether the file is in binary or ASCII format. A list of elements for each element type follows the header in the order described in the header.

The following is an example PLY in binary format with 19928 vertices and 39421 faces:

```
ply
format binary_little_endian 1.0
comment generated by 3GPP
element vertex 19928
property float x
property float y
property float z
property float nx
property float ny
property float nz
property int flags
property uchar red
property uchar green
property uchar blue
property uchar alpha
element face 39421
property list uchar int vertex_indices
property int flags

end_header


…
```

This example demonstrates the different components of a PLY file header. Each part of the header is a carriage-return terminated ASCII string that begins with a keyword. In case of binary representation, the file will be a mix of an ASCII header and binary representation of the elements, in little or big endian, depending on the architecture on which the PLY file has been generated. The PLY file must start with the characters "ply".

The vertex attributes listed in this example are the `(x,y,z)` floating point coordinates, the (nx,ny,nz) representation of the normal vectors, a 32 bit flag mask, `(r,g,b)` 8-bit representations of the colour of each vertex, an 8-bit representation of the transparency alpha channel. Faces are represented as a list of vertex indices with a flags attribute associated with each face.

4.6.3.6 Texture Formats

Different GPUs may support different texture formats, both raw and compressed. Raw formats include different representations of the RGB colour space, e.g. 8/16/32 bit representations of each colour component, with or without alpha channel, float or integer, regular or normalized, etc.

Typical GPU texture compression formats include BC1, PVRTC, ETC2/EAC, and ASTC. Other image compression formats such as JPEG and PNG need to be decompressed and passed to the GPU in a format that it supports.

Recently, the Basis Universal GPU texture format has been defined. This format also supports video texture compression. As decoding happens on the GPU, the application will benefit from reduced CPU load and CPU to GPU memory copy delay.

4.6.3.8 Bitrate and Quality Considerations

Bitrates and quality considerations for meshes are FFS.

4.6.3.7 Applications

Meshes are used in many applications.

4.6.4 Point Clouds

A point cloud is a collection of data points defined by a given coordinates system. In a 3D coordinates system, for example, a point cloud may define the shape of some real or created physical system. Point clouds are used to create 3D meshes and other models used in 3D modeling for various fields including medical imaging, architecture, 3D printing, manufacturing, 3D gaming and various XR applications.

Point clouds are often aligned with 3D models or with other point clouds, a process known as point set registration. In computer vision and pattern recognition, point set registration, also known as point matching, is the process of finding a spatial transformation that aligns two-point sets. The purpose of finding such a transformation includes merging multiple data sets into a globally consistent model, and mapping a new measurement to a known data set to identify features or to estimate its pose. Point set registration is used in augmented reality.

An overview on point cloud definitions, formats, production and capturing systems, rendering, bitrate/quality considerations and applications is for example provided in [38]. According to this document, media-related use cases may usually contain between 100,000 and 10,000,000 point locations and colour attributes with 8-10 bits per colour component, along with as some sort of temporal information, similar to frames in a video sequence. For navigation purposes, it is possible to generate a 3D map by combining depth measurements from a high-density laser scanner, e.g. LIDAR, camera captured images and localization data measured with GPS and an inertial measurement unit (IMU). Such maps can further be combined with road markings such as lane information and road signs to create maps to enable autonomous navigation of vehicles around a city. This use case requires the capture of millions to billions of 3D points with up to 1 cm precision, together with additional attributes, namely colour with 8-12 bits per colour component, surface normals and reflectance properties attributes. According to the paper, depending on the sequence, compression factors between 1:100 to 1:500 are feasible for media-related applications. According to the same paper, bitrates for single objects with such compression methods are in the range of 8 to 20 Mbit/s.

For production of point clouds, see also clause 4.6.7.

4.6.5 Light Fields

An overview on light-field technology is for example provided in https://mpeg.chiariglione.org/sites/default/files/events/7.%20MPEG127-WS_MehrdadTeratani.pdf.

4.6.6 Scene Description

Scene Descriptions are often called scene graphs due to their representation as graphs. A scene graph is a directed acyclic graph, usually just a plain tree-structure, that represents an object-based hierarchy of the geometry of a scene. The leaf nodes of the graph represent geometric primitives such as polygons. Each node in the graph holds pointers to its children. The child nodes can among others be a group of other nodes, a geometry element, a transformation matrix, etc.

Spatial transformations are represented as nodes of the graph and represented by a transformation matrix. Other Scene Graph nodes include 3D objects or parts thereof, light sources, particle systems, viewing cameras, …

This structure of scene graphs has the advantage of reduced processing complexity, e.g. while traversing the graph for rendering. An example operation that is simplified by the graph representation is the culling operation, where branches of the graph are dropped from processing, if deemed that the parent node’s space is not visible or relevant (level of detail culling) to the rendering of the current view frustum.

Scene descriptions permit generation of many different 3D scenes for XR applications. As an example, glTF from Khronos [39] is a widely adopted scene description specification and is now adopted by MPEG as the baseline for their extensions to integrate real-time media into scenes.

4.6.7 Production and Capturing Systems for 3D Mesh and Point Clouds

4.6.7.1 Overview

In order to capture a 3D mesh in a production facility using stereo cameras, an array of multiple cameras is placed around a recording space. A subject (for example an actor) is recorded inside the recoding space. Figure 4.6.7-1 shows an outline of one of the prototype studios as well as some example images from a production plant. In the left of Figure 4.6.7-1 a rotunda-like recording space is shown, in which multiple cameras are placed around the peripherals of the space.

Figure 4.6.7-1: Example of a capture and production facility for Point Cloud/ 3D meshes

A multi-camera or multi-stereo-camera pairs setup serves as a stereo camera base unit. Each stereo camera pair records the subject(s) from a different viewpoint. For example, Figure 4.6.7-2 shows different image captures from different multi-camera setup taken at one time instance. The volumetric capture studio has an integrated illuminated background for lightning.

Figure 4.6.7-2: Example of the 32 images captured simultaneously at one time instance in a studio

Figure 4.6.7-3 illustrates the 3D mesh production workflow. After the capture, a foreground and background segmentation process is performed in the 'keying' stage. A 'depth estimator' is applied to the captured images from each stereo pair to generate depth information with high accuracy for each pixel. From each stereo camera pair, the 3D information is stored as a colour-coded depth value in a 2D image. For example, a resulting depth map image from a stereo camera pair is shown in Figure 4.6.7-4.

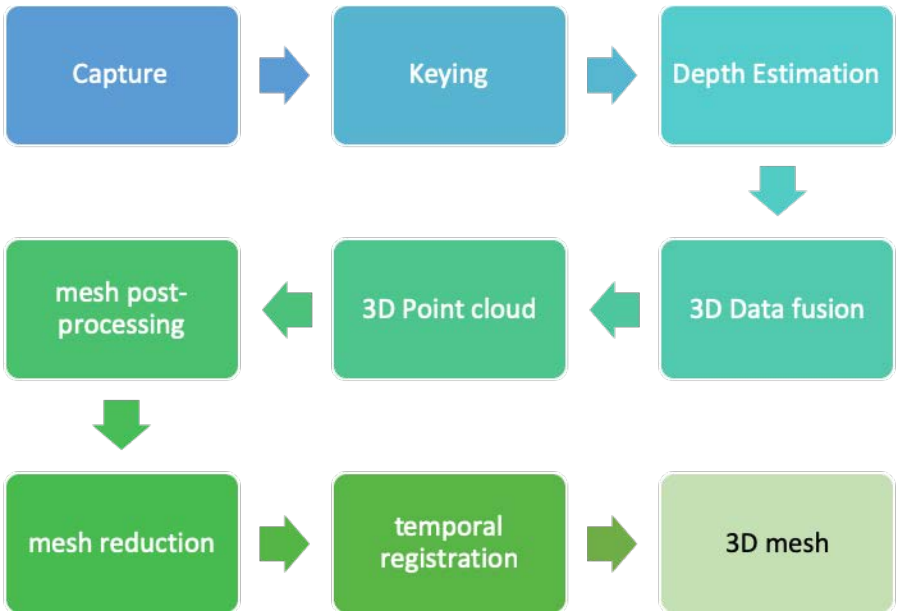


Figure 4.6.7-3: 3D mesh generation and production workflow

Figure 4.6.7-4: Example of a dense depth map calculated per frame for each stereo camera pair

In the following stage, the depth information from every stereo camera pair is merged using the initial camera calibration and a related 3D fusion process. Any 3D point which is occluded by others is filtered out, resulting in an advanced foreground segmentation.

The result of the 3D fusion process is a 3D Point cloud. The 3D Point cloud is further processed by different post-processing steps such as meshing, mesh reduction and texturing, etc as shown in the figure 4.6.7-5. The depth-based surface reconstruction results in a high-density mesh which consists of a high number of vertices and faces. In order to simplify the resulting high-density mesh to a single consistent mesh, a geometric simplification is performed. This process is called the mesh reduction. The simplified meshes are texturized using a 2D texture map in a common 2D image file format. In the final stage of the post-processing, the resulting meshes are temporally registered to obtain animated meshes.

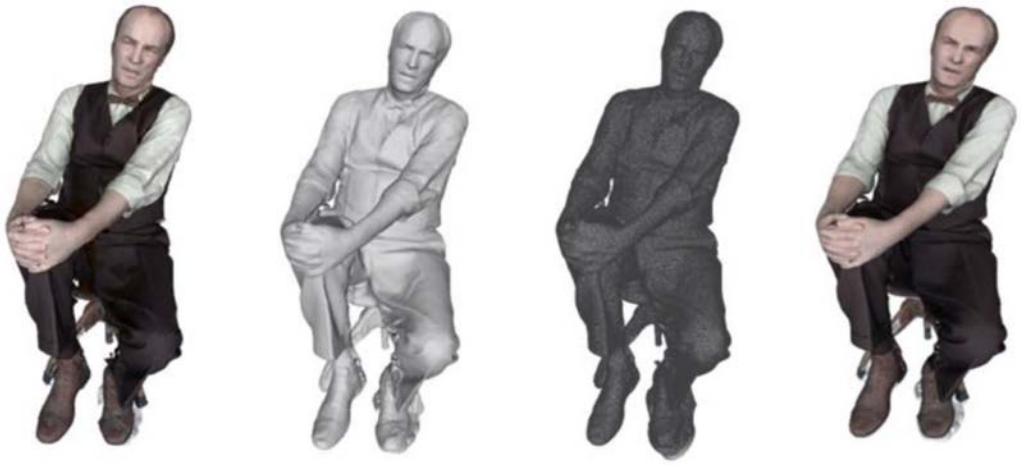


Figure 4.6.7-5: Example of resulting point cloud (left), and 3D models such as meshing, simplification and texturing (from second left to right)

4.7 3D and XR Audio Formats

For 3D and XR audio formats and systems, refer to 3GPP TR 26.918 [2], clause 4.3 as well as to 3GPP TS 26.118 [3], clause 6.

4.8 Devices and Form Factors

4.8.1 Device Types

Extended reality devices are of different form factors as shown in Figure 4.8-1. These form factors may differ in processing capabilities, communication types and power consumption.

The majority of Virtual Reality (VR) and Augmented Reality (AR) devices are head-mounted displays (HMDs).. In AR, the display is usually transparent and digital information is superimposed onto real life objects. In VR, the display is not transparent and only virtual information and images are displayed in front of wearer's eyes. Head-mounted display (HMD) is a device worn over the head. It features a display in front of one or both theeyes. The display streams data, images and other information in front of the wearer's eye(s). Certain HMDs have displays over both of their users' eyes, others only have a display over one of the users' eyes.

Typical components of HMDs are listed as follows:

- Optical systems: Display and lenses
- Tracking sensors and possibly additional sensors
- Cameras
- XR related processing (summarized as XR engine) including GPUs, CPUs, ASICs (e.g. dedicated media encoding and decoding), etc.
- Communication functionalities, as for example provided by a 5G System

A *smartphone* (defined as ***XR5G-P1*** device type) may be used both for AR as well as for VR (together with a card-board). In both cases, typically an XR engine/runtime is available to support processing of sensor data, viewport rendering as well as SLAM processing. In this case, the 5G modem and all media/XR processing is integrated in the device. Power consumption of such devices is important, but not ultimately critical due to the onboard battery.

VR HMDs typically provide the a large field of view, stereoscopic 3D imagery as well as rotational, translational and positional tracking for full 6DoF experiences. For VR, the following device types are identified:

- ***XR5G-V1*** - *Simple VR Display wired*: Such device types are commonly available in 2019. They only include sensors for tracking as well as a display. The remaining processing is done on a remote device, e.g. a puck or a smartphone. XR/Media Processing, connectivity and power supply are provided through wired tethering.
- ***XR5G-V2*** - *Simple VR Display wireless*: Such device types are not yet available in 2019. They include sensors for tracking, a display, a wireless connection (which could be WiFi based or 5G Sidelink based), and a power supply. The remaining processing is done on a remote device, e.g. a puck or a smartphone.
- ***XR5G-V3*** - *Smart VR Viewer wireless tethering*: Such device types are not yet available in 2019. They include sensors for tracking, a display, a wireless connection (which could be WiFi based or 5G Sidelink based), at least some XR processing, as well as a power supply. The remaining processing is done on a remote device, e.g. a puck or a smartphone.
- ***XR5G-V4*** - *VR HMD standalone*: Such device types are commonly available in 2019, except 5G connectivity. For such devices, the 5G modem, power supply as well as all media/XR processing is expected to be integrated into a single device.

Note that XR5G-V1 and XR5G-V2 are similar in terms of functions, but the wireless connectivity creates additional challenges and may even be done by a 5G sidelink communication.

An optical head-mounted display is a type of head-mounted display that projects images and allows the user to see through its display and is used in augmented reality (AR). Unlike Virtual Reality HMDs that obscures the vision of the real world, AR devices allow to see the surroundings while streaming data and image overlays in front of the eyes. Optical head-mounted displays may cover only 1 eye or both eyes. Wearers can interact with the projected digital content through input methods such as voice commands, gestures and controllers.

For AR Glasses, design constraints are significantly more important. In particular, design constraints apply in terms of sleekness, weight and power. The processing power is expected to be low to avoid battery consumption and thermal dissipation. Wireless AR glasses are commercially compelling. AR is typically associated to using glasses, also referred

to as optical see-through. However, AR experiences may be achieved using HMDs with video-see through functionalities. A comparison of different types is for example provided in [40]. The following device types are identified:

- ***XR5G-A1*** - *Simple AR Wearable Glass wired*: Such device types are available in 2019. They include a minimum number of sensors, possibly cameras for AR localization, as well as a display. Power, XR processing and connectivity is supplied from an external source.

- ***XR5G-A2*** - *Simple AR Wearable Glass wireless*: Such device types are not available in 2019 and are still far out. They would include a minimum number of sensors, possibly cameras for AR localization, power supply, as well as a wireless modem for connectivity. XR processing, AR localization as well as network connectivity is provided by external means, e.g. a puck or a smartphone.

- ***XR5G-A3*** - *Smart AR HMD video see-through*: Such device types are an initial entry for AR applications. There are much closer to XR5G-V1 type of devices, but by having cameras projecting the real world on the screen, the VR device can operate as an AR device.

- ***XR5G-A4*** - *AR Wearable Glass standalone*: Such device types are not available in 2019 but are under consideration. For such devices, the 5G modem, power supply, as well as all media/XR processing is expected to be integrated in a single device.

- ***XR5G-A5*** - *Smart AR Wearable Glass wireless*: Such device types are not available in 2019 and are still far out. In addition to an XR5G-A2 device, such a device would include at least a certain amount of XR/Media processing capabilities such as encoders/decoders and XRprocessing.

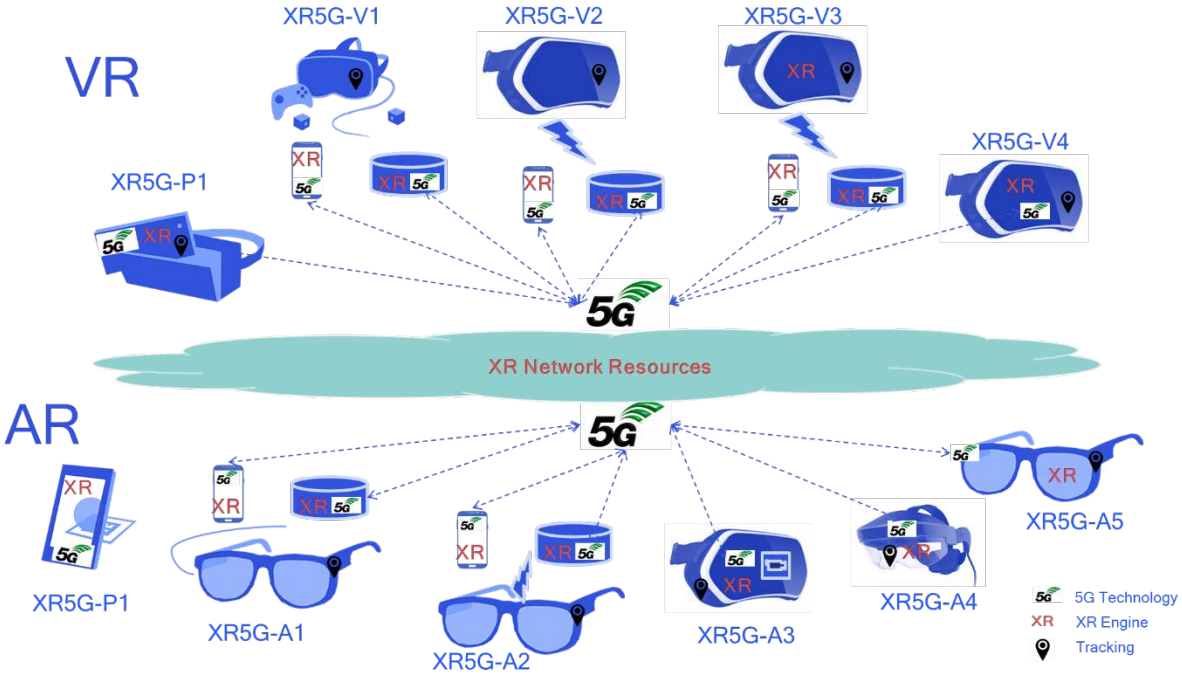


Figure 4.8-1: XR Form Factors

One of the most important issues when considering form factors and processing is the ability of the device to dissipate power, especially when no external cooling is available. Figure 4.8-2 shows the temperature rise depending on the surface power density. As example, two points on the figure can be considered:

- At 5C rise over ambient, power density that can be dissipated is 0.023 W/square inch. A smart phone whould have a surface area from 20 to 30 square inch, i.e. the power that can be dissipated is 0.5 to 0.75 Watt.

- AT 25C rise over ambient, power density that can be dissipated is 0.18 W/square inch. For a smartphone this would allow around 4 to 5 W continuous power dissipation. However, for an AR glass, the surface area is much smaller and so much less power can be dissipated, somewhere in the range of 1W. As an example, in a 25C room, the device enclosure surface temp would be 50C, which is already on the higher end of what is generally considered acceptable.

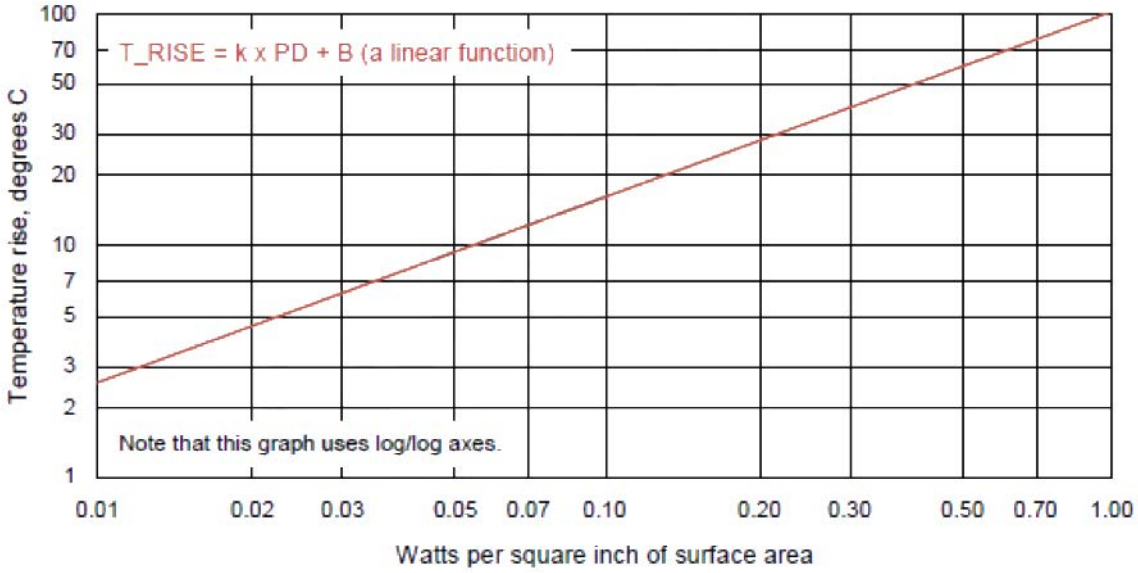


Figure 4.8-2: Temperature rise vs. power density

A summary of the different device types is provided in Table 4.8-1 along with tethering examples, placement of 5G Uu modem, XR engine and localization support, power supply and typical maximum available power. In all device types, the sensors are on the device. The table also addresses the options applicable for tethering between the device carrying the 5G Uu Modem, and the XR device, if applicable. The table also addresses options for the XR engine that includes scene recognition and viewport rendering. The following definitions for the XR engine are used:

- *External*: the device only supports display and receives a fully rendered viewport data that can be displayed directly. Any scene recognition, if applicable, is not on the device.
- *Split*: the external device does a pre-rendering of the viewport based on sensor information and the device does the final rendering considering the latest sensor information. Different degrees of split exist, as discussed before. Similalrly, scene recognition can be subject to split computation.
- *XR device*: that device does the full rendering of the viewport in the device, sensor information is only processed locally. Any scene recognition, if applicable, is on the device.

Table 4.8-1: XR Device Types

XR Type Number	XR Device Type Name	Tethering Examples	5G Uu Modem	XR Engine Localization	Power Supply	Typical Max Avail Power
XR5G-P1	Phone	n/a	XR device	XR device or split	Internal	3-5 W
XR5G-V1	Simple VR Viewer wired tethering	USB-C	External	External	External	2-5 W
XR5G-V2	Simple VR Viewer wireless tethering	802.11ad/y, 5G sidelink, etc.	External	External	Internal	2-3 W
XR5G-V3	Smart VR Viewer wireless tethering	802.11ad/y, 5G sidelink, etc.	External	XR device or Split	Internal	2-3 W
XR5G-V4	VR HMD Standalone	n/a	XR device	XR device or Split	Internal	3-7 W
XR5G-A1	Simple AR Wearable Glass wired tethering	USB-C	External	External	External	1-3 W
XR5G-A2	Simple AR Wearable Glass wireless tethering	802.11ad/y, 5G sidelink. etc.	External	External	Internal	0.5 – 2 W
XR5G-A3	Smart AR HMD see-through standalone	n/a	XR device	XR device or Split	Internal	3-7 W
XR5G-A4	AR Wearable Glass standalone	n/a	XR device	XR device or Split	Internal	2 - 4 W
XR5G-A5	Smart AR Wearable Glass wireless tethering	802.11ad/y, 5G sidelink. etc.	External	XR device or Split	Internal	0.5 – 2 W

4.8.2 Power Consumption

This clause addresses the available power in different XR Device types as well as the power consumption of typical XR processing functions as identified in the context of XR services.

When designing media processing, XR functionality and 5G connectivity, it is important to understand the power consumption of different components that are possibly integrated in XR devices. The following should be considered:

- Tracking and Sensing
  - 3DoF tracking may be done with low power consumption, e.g., below 1 Watt
  - 6DoF tracking involving for example, capturing cameras, LEDs for eye and hand tracking, etc. are more power-consumption intense
- Display
  - Display power consumption is critical and depends on the device.
  - Display power consumption can be in the range of 0.3W up to 1W
- Render (GPU)
  - The power consumption of the GPU depends on frame rates, resolution, display technology
  - The power consumption can be from several mWatt to several Watt depending on the use case
- Compute and Media Processing (CPU)
  - Similar observation as for the GPU
  - If encoding is involved, power consumption is typically higher.
- Connectivity
  - The power consumption of wireless contection such as 5G depends on several factors including bitrates, distance from radio access network, channel conditions, frequeny range, etc.

- The power consumption can be from several mWatt to several Watt depending on the use case.

It is expected that each of the components will undergo improvements to address power savings. It is important that in the development of technical specifications of XR devices, the power consumption of each component is considered.

4.9 Ongoing Standardisation Work

4.9.1 Related Work in 3GPP

4.9.2.1 Introduction

This clause summarizes relevant 3GPP activities efforts in the context of XR.

- 3GPP TR 26.918 [2] provides an introduction to Virtual Reality and 3GPP TS 26.118 [3] defines Virtual Reality Media Profiles for omnidirectional 3DoF media.
- 3GPP TR 22.842 [6] on Network Controlled Interactive Service (NCIS) analyses several use cases of NCIS as follows: NCIS Service Supporting
    - New Requirements for VR Based NCIS Service
    - Cloud Rendering for Games
    - High Speed Scenario
    - IoE Based Social Networking
    - Communication within NCIS group

  Based on the TR, several requirements are identified for new requirements in 3GPP TS 22.261 [41]. Also, KPIs for such services mentioned above are documented in clause 6.2 of 3GPP TR 22.842 [6], requiring additional input including some information from this TR.
- In context of Release-17, 3GPP work is ongoing in order to identify the integration of edge processing in 5G systems. 3GPP TR 23.748 [28] defines modifications to 5GS system architecture to enhance Edge Computing. This work is currently in study phase, defining Key Issues and scope for Release-17. In addition, in 3GPP TR 23.758 [27] a new set of application layer interfaces for Edge Computing is identified that may potentially be useful for integration edge computing.

4.9.2 Related Work External of 3GPP

4.9.2.1 Introduction

This clause summarizes relevant external standardisation efforts in the context of XR that may provide certain functionalities being of benefit for 5G-based XR applications.

4.9.2.2 MPEG

4.9.2.2.1 Introduction

In October 2016, MPEG initiated a new project on “Coded Representation of Immersive Media”, referred to as MPEG-I. The proposal was justified by the emergence of new devices and services that allow users to be immersed in media and to navigate in multimedia scenes. It was observed that a fragmented market exists for such devices and services, notably for content that is delivered “over the top”. The project is motivated by the lack of common standards that do not enable interoperable services and devices providing immersive and navigable experiences. The MPEG-I project is expected to enable existing services in an interoperable manner and to support the evolution of interoperable immersive media services. Enabled by the Parts of this Standard, end users are expected to be able to access interoperable content and services, and acquire devices that allow them to consume these.

After the launch of the project, several phases, activities, and projects have been launched that enable services considered in MPEG-I.

The project is divided in *tracks* that enable different core experiences. Each of the phases is supported by key *activities* in MPEG, namely in systems, video, audio and 3D graphics-related technologies.

Core technologies as well as additional enablers are implemented in *parts* of the MPEG-I standard. Currently the following 14 parts are under development:

- Part 1 – Immersive Media Architectures
- Part 2 – Omnidirectional MediA Format
- Part 3 – Versatile Video Coding
- Part 4 – Immersive Audio Coding
- Part 5 – Video-Based Point Cloud Coding (V-PCC)
- Part 6 – Immersive Media Metrics
- Part 7 – Immersive Media Metadata
- Part 8 – Network-Based Media Processing
- Part 9 – Geometry Point Cloud Coding (G-PCC)
- Part 10 – Carriage of Video-based Point Cloud Coding Data
- Part 11 – Implementation Guidelines for Network-based Media Processing
- Part 12 - Carriage of Geometry-based Point Cloud Coding Data
- Part 13 – Multi-Decoder Video Decoding Interface for Immersive Media
- Part 14 – Scene Description for MPEG Media

In addition, other technical components may be provided in existing MPEG specifications outside of MPEG-I (e.g., HEVC and AVC) in order to create interoperable immersive experiences.

4.9.2.3 Khronos

Khronos creates open standards for 3D graphics, Virtual and Augmented Reality, Parallel Computing, Neural Networks, and Vision Processing. Specifically relevant for the work on XR are the following activities:

- *OpenGL*® is the most widely adopted 2D and 3D graphics API in the industry, bringing thousands of applications to a wide variety of computer platforms. It is window-system and operating-system independent as well as network-transparent. OpenGL enables developers of software for PC, workstation, and supercomputing hardware to create high-performance, visually compelling graphics software applications, in markets such as CAD, content creation, energy, entertainment, game development, manufacturing, medical, and virtual reality. OpenGL exposes all the features of the latest graphics hardware.
- *Vulkan* is a new generation graphics and compute API that provides high-efficiency, cross-platform access to modern GPUs used in a wide variety of devices from PCs and consoles to mobile phones and embedded platforms.
- *OpenXR* [16] is an open standard that provides high-performance access to Augmented Reality (AR) and Virtual Reality (VR)—collectively known as XR—platforms and devices.
- *glTF*™ (GL Transmission Format) [39] is a specification for the efficient transmission and loading of 3D scenes and models by applications. glTF minimizes both the size of 3D assets, and the runtime processing needed to unpack and use those assets. glTF defines an extensible, common publishing format for 3D content tools and services that streamlines authoring workflows and enables interoperable use of content across the industry.

4.9.2.4 W3C WebXR

The WebXR Device API Specification (https://immersive-web.github.io/webxr/) [17] provides interfaces to VR and AR hardware to allow developers to build compelling, comfortable VR/AR experiences on the web. The latest “WebXR Device API, Editor’s Draft, 10 February 2020” is available here https://immersive-web.github.io/webxr/ and provides an interface to VR/AR hardware. It is no longer marked as “UNSTABLE API”. It also provides a link to WebXR Device API Explained.

4.10 XR Use Cases

In Annex A of this document, a significant amount of use cases are collected that serve for identifying potential interfaces, formats, protocols and requirements for the 5G system in order to support XR applications.

Table 4.10 provides an overview of the use cases and their characterization.

In addition, this table more explicitly adds the device types that have been developed in clause 4.8.

Table 4.10: Overview of Use cases as documented in Annex A

No	Use Case	Type	Experience	Delivery	Device Types
1	3D Image Messaging	AR	3DoF+, 6DoF	Upload and Download	XR5G-P1 XR5G-AX
2	AR Sharing	AR, MR	6DoF	Local, Messaging Download and Upload	XR5G-P1 XR5G-AX
3	Streaming of Immersive 6DoF	VR	3DoF+, 6DoF	Streaming Interactive Split	XR5G-V3 or XR5G-V4 with controller
4	Emotional Streaming	2D, AR and VR	2D, 3DoF+, 6DoF	Streaming Interactive, Split	XR5G-P1 XR5G-V3 XR5G-V4
5	Untethered Immersive Online Gaming	VR	6DoF	Streaming, Interactive, Split	XR5G-V3 XR5G-V4 with gaming controller
6	Immersive Game Spectator Mode	VR	6DoF	Streaming, Split	XR5G-P1 XR5G-V3 XR5G-V4
7	Real-time 3D Communication	3D, AR	3DoF+	Conversational	XR5G-P1 XR5G-AX
8	AR guided assistant at remote location (industrial services)	2D video with dynamic AR rendering of graphics	6DoF (2D + AR)	Local, Streaming, Interactive, Conversational	XR5G-P1 XR5G-AX
9	Police Critical Mission with AR	AR, VR	3DoF to 6DoF	Local, Streaming, Interactive, Conversational, Group Communication	XR5G-A3 XR5G-A4
10	Online shopping from a catalogue – downloading	AR	6DoF	Download	XR5G-P1 XR5G-AX
11	Real-time communication with the shop assistant	AR	6DoF	Interactive, Conversational	XR5G-P1 XR5G-AX
12	360-degree conference meeting	AR, MR, VR	3DoF	Conversational	XR5G-P1 XR5G-V3 XR5G-V4
13	3D shared experience	AR, MR, VR	3DoF+ 6DoF	Conversational	XR5G-P1 XR5G-V3 XR5G-V4
14	6DOF VR conferencing	VR	6DoF	Interactive, Conversational	XR5G-V3 XR5G-V4
15	XR Meeting	AR, VR, XR	6DoF	Interactive Conversational	XR5G-P1 XR5G-A1 XR5G-A2 XR5G-A5
16	Convention / Poster Session	AR, VR, MR	6DoF	Interactive Conversational	XR5G-P1 XR5G-A1 XR5G-A2 XR5G-A5

No	Use Case	Type	Experience	Delivery	Device Types
17	AR animated avatar calls	AR	2D, 3DoF	Conversational	XR5G-P1 XR5G-A1 XR5G-A2 XR5G-A5
18	Online shopping from a catalogue – downloading	AR	6DoF	Download	XR5G-P1 XR5G-A1 XR5G-A2 XR5G-A5
19	Front-facing camera video multi-party calls	AR	3DoF	Conversational	XR5G-P1 XR5G-AX
20	AR Streaming with Localization Registry	AR, Social AR	6DoF	Streaming, Interactive, Conversational	XR5G-A1 XR5G-A2 XR5G-A5
21	Immersive 6DoF Streaming with Social Interaction	VR and Social VR	3DoF+, 6DoF	Streaming Interactive Conversational Split	XR5G-V3 XR5G-V4
22	5G Online Gaming Party	VR	6DoF	Streaming, Interactive, Split, D2D	XR5G-V3 XR5G-V4
23	Spatial Shared Data	AR	6DoF	Streaming Interactive Conversational Split	XR5G-AX

The use cases are summarized in clause 5 into several core use cases and scenarios.

4.11 Summary of Remaining Issues addressed in this Document

Based on these introduced technologies and the use cases, the remainder of this Technical Report is addresses the following:

- identify the mapping of different XR use cases to the 5G System and 5G Media Delivery services according to clause 4.3.
- identify functions, interfaces and APIs for different delivery scenarios.
- define high-level call flows and parameter exchange for the different service scenarios.
- for the different scenarios, identify the formats as well as traffic requirements/properties
- identify technical requirements on formats, processing functions, interfaces, and protocols in order to achieve adequate Quality of Experience based on the considerations in clause 4.2.
- identify potential standardisation areas and their potential timeline.

5 Core Use Cases and Scenarios for Extended Reality

5.1 Introduction

This clause documents core consolidated use cases and scenarios for extended reality based on the underlying offered functionalities, the nature of the communication, interactivity and real-time requirements. These have been derived from the use cases collected in Annex A. Table 5.1-1 lists the core use cases and the list of the use cases in Annex A covered by each of them.

The following categories and their accompanying illustrations are drawn for clarity. In actual implementations, elements from more than one system may be used together. However, it is attempted to ensure that the functionalities included from any particular use case in Annex A are fully covered within a single core use case diagram.

The term UE is used to define a 5G-enabled user equipment that meets the capability requirements of a particular use case. A UE may for example be a mobile handset, AR glasses or an HMD with or without controllers as documented in clause 4.3.

Note that depending on the actual UE, the usage scenario may differ slightly. Some examples of such differences are given in the accompanying text within each category. Furthermore, in certain cases the capabilities of the devices may be present on each UE and not restricted to only one side.

Table 5.1-1: Core use case mapping to Annex A

Core Use Cases and Scenarios	Clause	Use Case from Annex A
Offline Sharing of 3D Objects	5.2	Use Case 1: 3D Image Messaging Use Case 2: AR Sharing Use Case 10: Online shopping from a catalogue – downloading
Real-time XR Sharing	5.3	Use Case 7: Real-time 3D Communication Use Case 8: AR guided assistant at remote location (industrial services) Use Case 11: Real-time communication with the shop assistant Use Case 17: AR animated avatar calls Use Case 23: 5G Shared Spatial Data
XR Multimedia Streaming	5.4	Use Case 3: Streaming of Immersive 6DoF Use Case 4: Emotional Streaming Use Case 20: AR Streaming with Localization Registry Use Case 21: Immersive 6DoF Streaming with Social Interaction
Online XR Gaming	5.5	Use Case 5: Untethered Immersive Online Gaming Use Case 6: Immersive Game Spectator Mode Use Case 22: 5G Online Gaming party
XR Mission Critical	5.6	Use Case 9: Police Mission Critical with AR
XR Conference	5.7	Use Case 12: 360-degree conference meeting Use Case 13: 3D shared experience Use Case 14: 6DOF VR conferencing Use Case 15: XR Meeting Use Case 16: Convention / Poster Session
Spatial Audio Multiparty Call	5.8	Use Case 18: AR avatar multi-party calls Use Case 19: Front-facing camera video multi-party calls

5.2 Offline Sharing of 3D Objects

5.2.1 Summary of Use cases

This clause summarizes and integrates the following use cases from Annex A in a single core use case referred to as "Offline Sharing of 3D Objects":

- Use Case 1: 3D Image Messaging (see Annex A.2)
- Use Case 2: AR Sharing (see Annex A.3)
- Use Case 10: Online shopping from a catalogue – downloading (see Annex A.10)

5.2.2 Description

Offline sharing is used for sharing 3D models/objects and 3D MR scenes amongst UEs. In Figure 5.2-1, UE A shares a 3D static/dynamic object with UE B. The 3D object can be a stored object downloaded by UE A from the cloud, or captured by the device using for example a depth camera. It may include additional information such as colour, texture, size, etc. of the 3D object, which is referred to as *effects* in the figure. Upon receiving, UE B can render this object (and/or 3D objects it has downloaded from the cloud) in the surrounding reality using an MR rendering engine; it can choose the desired effects for the 3D object. It can then capture the rendered MR scene and send it back to UE A. MMS is used for sharing the 3D object and the captured MR scene between the UEs. Note that the diagram is drawn for clarity and in reality, the capabilities of the devices may be present on each UE and not limited only to one side.

The rendering functionality depends on the type of device. For instance, flat images on a mobile phone vs. 3D rendering on AR glasses.

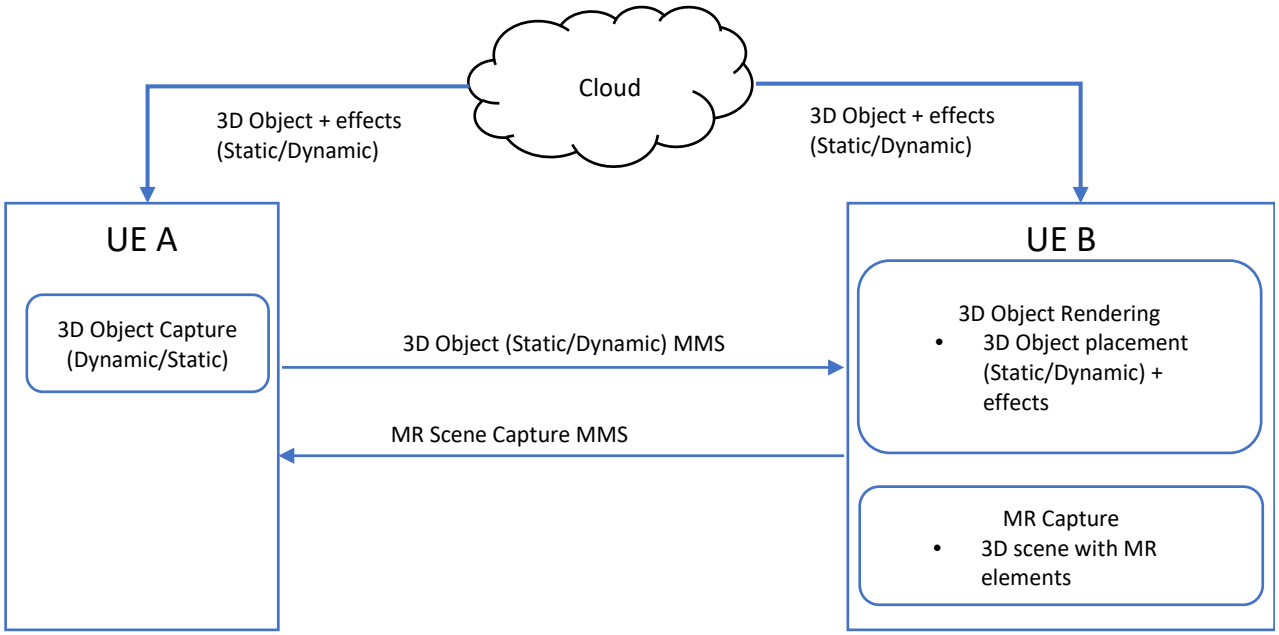


Figure 5.2-1: Offline Sharing 3D Objects and MR Scenes

5.2.3 Potential Normative Work

Table 5.2-1 provides an overview of potential normative work that has been collected as part of the use case collection in Annex A and maps the potential work area to one or several use cases in Annex A.

Table 5.2-1: Overview of potential normative work linked to different offline sharing use-cases in Annex A

Potential Normative Work	Link to Use Case
1. Standardized format for 3D objects	Use Case 1, 3D Image Messaging: "*The phone captures a set of images and builds a 3D model of the object.*" Use Case 10: Online shopping from a catalogue – downloading: "*This online shop provides for each selling product, 2D images, 3D objects models and detailed information on size, colour, materials."*
2. Standardized format for mixed reality 3D scenes	Use Case 2, AR Sharing: *"Bob places the virtual model of the couch on a plane surface in the living room"* Use Case 10: Online shopping from a catalogue – downloading: "*This online shop provides for each selling product, 2D images, 3D objects models and detailed information on size, colour, materials."*
3. Delivery protocol for 3D objects and scenes (MMS extension and/or other)	Use Case 1, 3D Image Messaging: "*Alice sends the image to Bob as an MMS message.*" Use Case 2, AR Sharing: "*Alice scans a QR code with her phone to download a 3D model of the couch and sends it to Bob via MMS."* Use Case 10: Online shopping from a catalogue – downloading: "*This online shop provides for each selling product, 2D images,*

	3D objects models and detailed information on size, colour, materials."
4. Decoding, rendering, composition API for 3D objects in an AR scene.	Use Case 2, AR Sharing: *"Bob likes how the couch fits in their living room and captures a 3D picture of the room with the couch and shares it with Alice"* Use Case 10: Online shopping from a catalogue – downloading: "*The sofa is then rendered on his AR glasses and John continue to use his smartphone in order to control the location of the sofa within the living room*."

5.3 Real-time XR Sharing

5.3.1 Summary of Use Cases

This clause summarizes and integrates the following use cases from Annex A in a single core use case referred to as "Real-time XR Sharing":

- Use Case 7: Real-time 3D Communication
- Use Case 8: AR guided assistant at remote location (industrial services)
- Use Case 11: Real-time communication with the shop assistant
- Use Case 17: AR animated avatar calls
- Use Case 23: 5G Shared Spatial Data

5.3.2 Description

UE B is a device capable of XR rendering, such as AR glasses, or a mobile phone that is sending a real-time video stream of the XR experience to UE A, as illustrated in Figure 5.3-1. The XR experience is being captured and rendered by the UE B. The experience includes capture of 3D objects in a 2D scene. The rendering experience includes (real-time) 3D (static/dynamic) object placement in a 2D scene, overlay video, avatars, etc., that may be downloaded from the cloud, or sent by UE A to UE B. A bidirectional or unidirectional A/V channel may be open between the devices depending on the use case. The received objects by UE B can be influenced by UE B directly as well, based on direct or indirect input from the user. UE A also sends a real-time stream of motion signals and effects that influence the rendering of the 3D object model on UE B. Examples include: 1) head/body motion or facial expressions resulting in corresponding changes in a dynamic 3D object, e.g. an avatar, and 2) positioning and size of the 3D object within the XR scene, and subsequent changes in these parameters such as moving the object closer to a wall or making it larger. Motion data may be collected using direct input from the user interface of the device or implicitly using data from camera, gyroscopes, accelerometer, etc., including gestures. Other predefined effects for the 3D objects that can be placed on or around it can also be shared from UE A to UE B or downloaded from the cloud. Network based XR media processing may be used where required.

In a subset of this scenario where XR is not used, a 3D object may be rendered within the received video stream, e.g., a 3D representation of the head of a video call participant.

Shared AR experiences can be realized using a Spatial Computing Server (SCS), as shown in Figure 5.3-1. Collocated users with AR devices can view and interact with AR objects in a time synchronized and spatially correct manner using the SCS. Devices will send positional data updates (e.g. GPS, Wifi, visual scans etc.) to the SCS. The SCS provides spatial maps, spatial anchors and localization services to the devices. Spatial data may be crowd-sourced from multiple devices for higher accuracy in spatial mapping. It should also be possible to then assign spatial information to AR objects so they can be be dropped by users at specific locations for later discovery by other visitors.

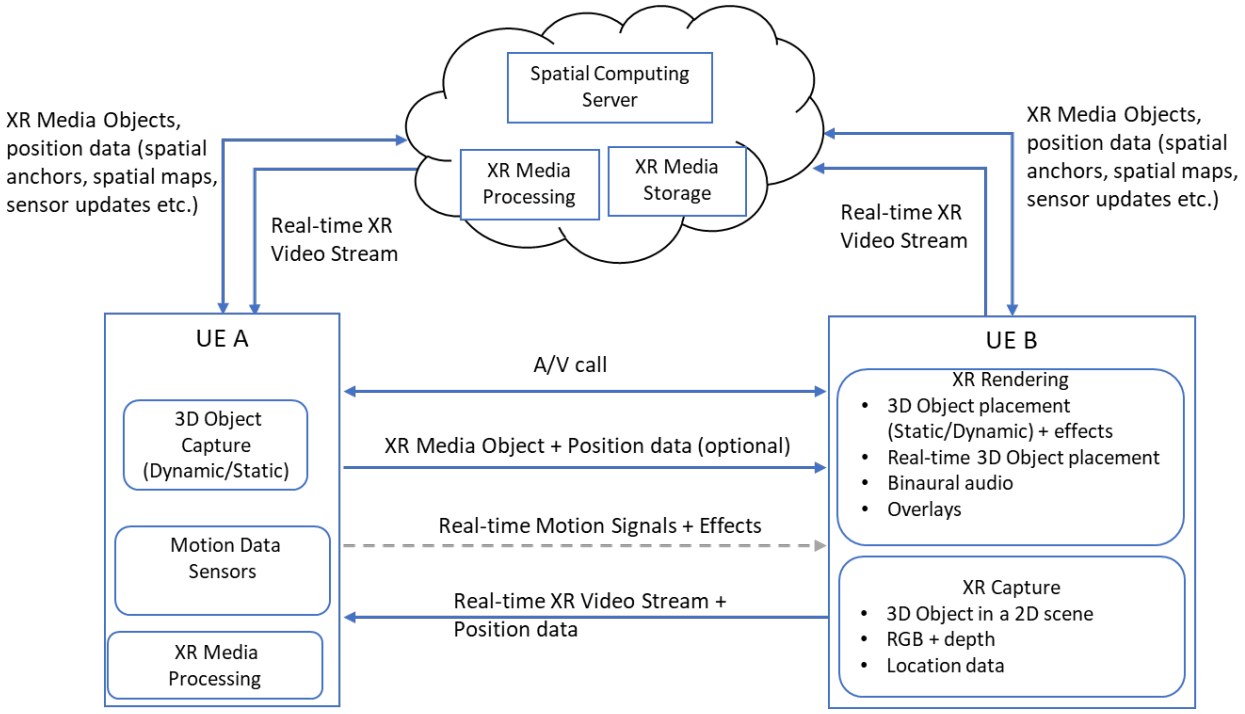


Figure 5.3-1 Real-time sharing of XR content

5.3.3 Potential Normative Work

Potential normative work that has been collected as part of the use case collection in Annex A is provided in the following:

- Support of static/dynamic 3D objects in MTSI (formats and transport).)
- Overlaid video rendering in the network or locally.
- Overlaid video rendering in the network or locally.
- Coded representation of 3D depth signals and transport in MTSI.
- Coded representation of XR scenes and transport in MTSI.
- MTSI/FLUS uplink of XR video.
- Downlink XR video with local/cloud computation and rendering.
- Visual coding and transmission of avatars or cut-out heads, alpha channel coding.
- Transports and potentially coding of motion data to show attentiveness.
- Scalable formats for storing and sharing spatial information.
- Data representation for spatial information.
- Collected sensor data to be streamed.
- Content delivery protocols to access AR maps and content items.

5.4 XR Multimedia Streaming

5.4.1 Summary of Use Cases

This clause summarizes and integrates the following use cases from Annex A in a single core use case referred to as "XR Multimedia Streaming":

- Use Case 3: Streaming of Immersive 6DoF
- Use Case 4: Emotional Streaming
- Use Case 20: AR Streaming with Localization Registry
- Use Case 21: Immersive 6DoF Streaming with Social Interaction

5.4.2 Description

This category covers live and on-demand streaming of XR multimedia streams, which include 2D or Volumetric video (i.e., Constrained 6DoF) streams that are rendered in XR with binaural audio as well as 3DOF and 3DOF+ immersive A/V streams. It is illustrated in Figure 5.4-1, where UE is a device capable of receiving and rendering the type of stream in use. It is also capable of controlling the playback of these streams using input from handheld controllers, hand gestures, biometric readings, body and head movements etc., which is communicated to the content server. Control signals include pause, rewind, viewpoint selection or, in case of emotional streaming, adaptive content selection.

In another system instance, the content server can provide Inter-destination Multimedia Synchronization (n) for a group experience. A Spatial Computing Server is used by XR capable devices to register, compute, update and recall the spatial configuration of their surroundings. The service is meant for indoor spaces like a shared room or building. Appropriate surfaces may be selected for display of XR streams and saved in the Spatial Computing Server. The configuration can be shared amongst authorized users to enhance group experience when multiple users are physically sharing the same space.

A social aspect may be added to XR multimedia streaming by receiving live social media feeds, in addition to the XR media stream, that can be rendered as an overlay. Furthermore, the users may be able to see the avatars of other remote located users consuming the same media (additionally from the same viewpoint in case of multiple viewpoints) and have audio conversations with them by establishing one or more communication channels. The social aspects are added by the cloud in the Figure 5.4-1. The XR Media storage in the cloud is for fetching XR objects, e.g., avatars. The location is shared with the cloud to establish colocation within the XR. For instance, users viewing from the same viewpoint may be considered collocated when consuming synchronized media.

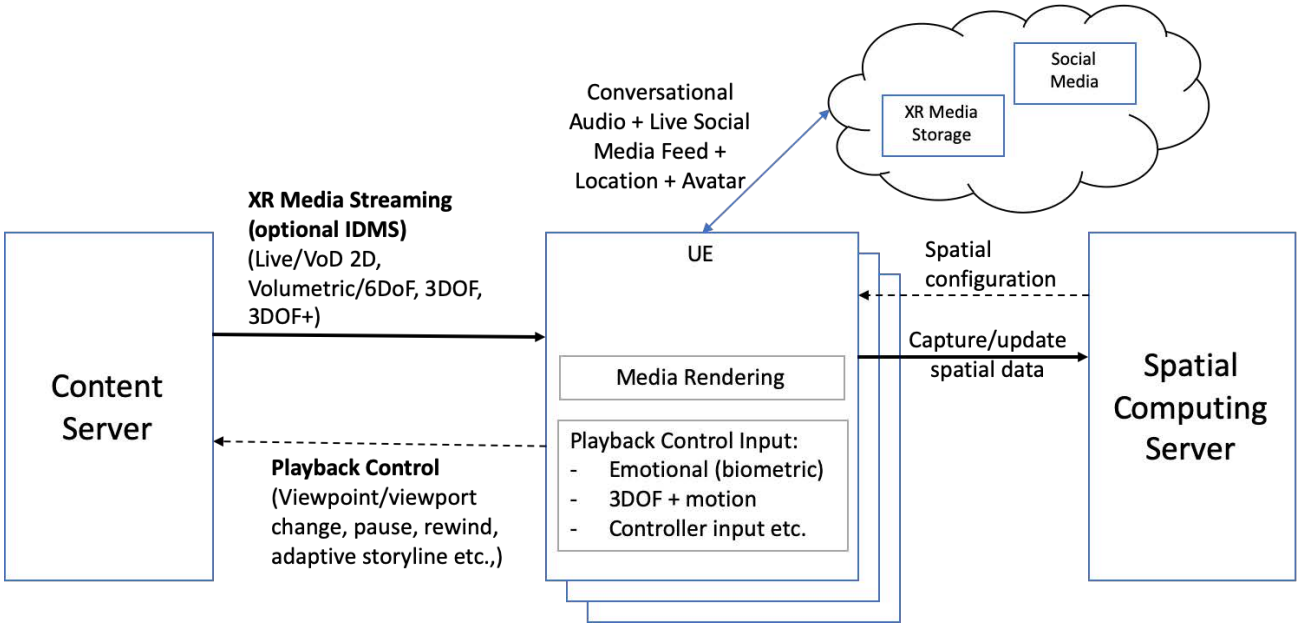


Figure 5.4-1: XR Multimedia Streaming

5.4.3 Potential Normative Work

For this use case, the potential normative work may cover:

- Coded representation of Audio/Video Formats as well as geometry data for XR (volumetric, 3DoF+)
- Scene composition and description
- Storage and Cloud Access Formats
- Transport protocols to support any media beyond 2D streaming
- Decoding, rendering and sensor APIs
- Biometrics and Emotion Metadata definition and transport
- Seamless splicing and smooth transitions across storylines
- Format for storing and sharing indoor spatial information.
- Inter-destination multimedia synchronization for group/social experience.
- Social VR Components – Merging of avatar and conversational streams to original media (e.g., overlays, etc.)

5.5 Online XR Gaming

5.5.1 Summary of Use Cases

This clause summarizes and integrates the following use cases from Annex A in a single core use case referred to as "Online XR Gaming":

- Use Case 5: Untethered Immersive Online Gaming
- Use Case 6: Immersive Game Spectator Mode
- Use Case 22: 5G Online Gaming party

5.5.2 Description

The system as illustrated in Figure 5.5-1 consists of a game server capable of serving several online gamers. Each player UE receives a stream of the live game from the game server and sends control signals back to the server that influence the game play. Control signals include handheld controller inputs, biometric readings and 3DOF+ motion as required by the gameplay.

Other users may join the live game in spectator mode and can receive the stream from the perspective of an active player or a spectator view independent of other players; viewpoint changes may be possible. The spectator may enjoy an immersive experience or 2D view depending on the device. Optionally, the spectators may interact with the players through cheering or game reward systems. In a different instance, cloud rendering could be present.

In an extension, the players may be co-located at a gaming party/session using device-to-device or centralized (via gaming server) communication to enhance the user experience due to improved QoS parameters. A geofenced area may be used as the game arena for an AR gaming session, with support from a Spatial Computing Server for location registration and update and a Group discovery protocol for identifying and matching players. AI methods may be required for Image & Object Recognition, XR Lighting, Occlusion Avoidance, Shared Persistence. Spatial Computing Server receives continuous updates from the players in the arena, which are used for spatial mapping and localization. The updated spatial data (e.g., maps and player locations) is shared with the other devices and also the game server.

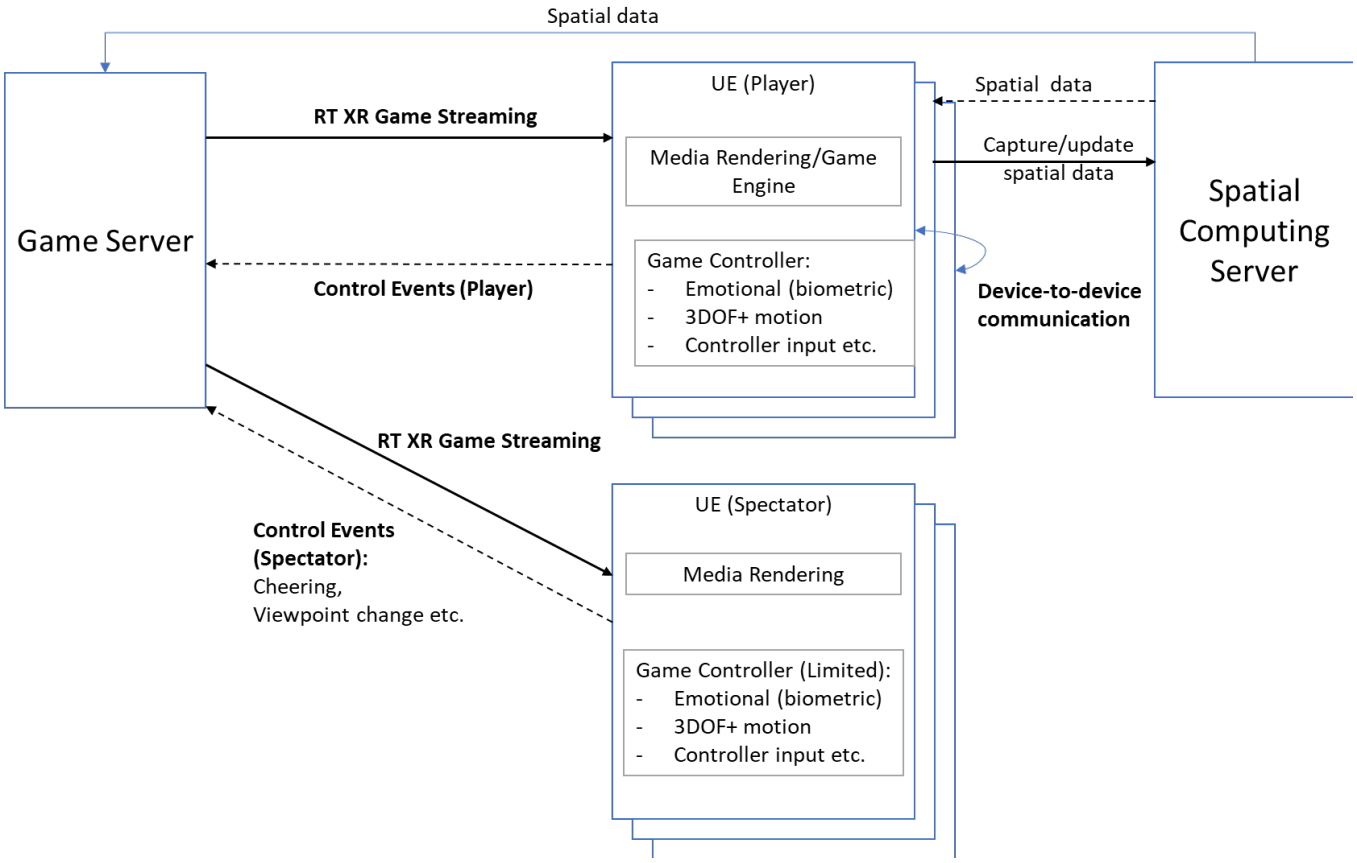


Figure 5.5-1: Online XR Gaming

5.5.3 Potential Normative Work

For this use case, the potential normative work may cover:

- Streaming protocols for online gaming
- Decoding, rendering and sensor APIs
- Architectures for computing support in the network (see gap analysis in TR 22.842, clause 5.3.6)
- Coded Representation of Audio/Video Formats for gaming.
- Device-to-device communication support for XR gaming.
- Format for storing and sharing location information.
- Network support for group discovery and authentication, shared persistence and spatial mapping.

5.6 XR Mission Critical

5.6.1 Summary of Use Cases

This clause summarizes and integrates the following use cases from Annex A in a single core use case referred to as "XR Mission Critical":

- Use Case 9: Police Mission Critical with AR

5.6.2 Description

The system shown in Figure 5.6-1 is for critical missions using XR support. In this use case a team in an indoor space is equipped with mission gear that is connected to a centralized control center. The UE in the figure represents a single team member and there may be more than one device included in the misson gear, such as AR glasses, a 360 degree helmet-mounted camera, a microphone, binaural audio headphones and other sensors. The control center/conference server performs the role of mission support by providing XR graphics such as maps, text, location pointers of other team members or other objects/people in the surrounding, etc. The mixed audio of the team members as well as audio from control center is also delivered to the UE to aid team communication and coordination. One or more drone-mounted cameras may also be used, which can be controlled by the control center or one of the members of the mission team. The control center is equipped with A/V processing capabilities to extract important information from the multitude of 360 degree video feeds, for instance, for identifying moving objects. All devices at the site(UEs and drones) use MCPTT to communicate.

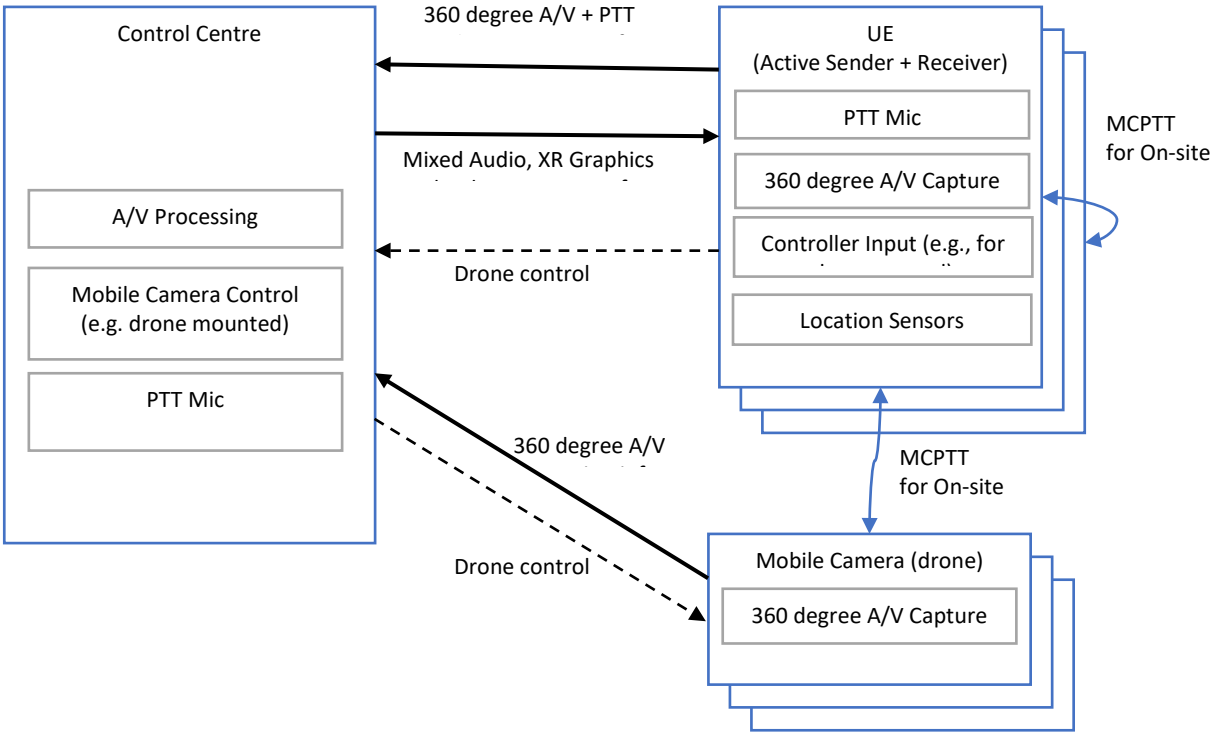


Figure 5.6-1: XR Critical Mission

5.6.3 Potential Normative Work

For this use case, the potential normative work may cover:

- MTSI/FLUS uplink 3D audio
- MTSI/FLUS/MCVideo uplink XR streams
- Downlink XR video with overlaid graphics with local/cloud computation and rendering
- Downlink XR audio with mixed-in 3D audio objects with local/cloud computation and rendering.
- MTSI/MCPTT SWB/FB voice communication

5.7 XR Conference

5.7.1 Summary of Use Cases

This clause summarizes and integrates the following use cases from Annex A in a single core use case referred to as "XR Conference":

- Use Case 12: 360-degree conference meeting
- Use Case 13: 3D shared experience
- Use Case 14: 6DOF VR conferencing
- Use Case 15: XR Meeting
- Use Case 16: Convention / Poster Session

5.7.2 Description

This system caters for an XR conference with multiple physically co-located and remote participants using XR to create telepresence. The shared conference space can be, 1) a physical space that is shared by local participants and sent as an immersive stream to the remote participants, 2) a virtual space that has the same layout as the physical space so that the physically present (local) and remote participants have a similar experience while moving in the space (e.g. captured via a 360-degree camera), and 3) a virtual space that is retrieved from an application server (AS). In any of the 3 options the conference space might be extended with other Media Objects (e.g. a presentation video) retrieved from an application server (AS). The UE functionality can be split into two parts one for user media capture and one for rendering. In practice, these functions may (and usually will) be integrated within a single device (e.g. a smartphone) possibly augmented by peripheral devices like a wireless camera. Another option is that they are indeed separated, e.g. using a dedicated capture device (like a 360-degree camera) and a XR rendering device (like AR Glasses, mobile phone, VR HMD, holographic display, etc.). However, it should also be considered that some UEs will render the immersive communication experience on traditional displays.

Figure 5.7-1 illustrates the system. Virtual spaces and avatars are retrieved from the Application Server by the Conference Server. A Spatial Computing Server is used for storing the layout of the physical space, when used. Remote participants would be seen as avatars within the XR experience located at their relative position in the shared space. Alternatively, they may be represented as themselves using a live video feed and video extraction to isolate the person from the background and using a photo-realistic representation of their face in place of the HMD. The required video processing is located in the conference server in Figure 5.7-1. For example, a Network Media Processing function may perform any media and/or metadata processing that is required to place, a certain user and multiple other users consistently into a virtual environment. A typical example would be a Multipoint Control Unit (MCU) function that combines and processes the video captured from the various users in order to reduce the resource requirements of the clients. Finally, the Conference Server takes care of all the session signalling to set up channels for the exchange of media data and metadata. It performs Session Control functions for the scene configuration, network media processing and common scene media.

Remote participants are free to move around when 6DOF motion is used. The conference server processes the audio streams from the participants to create an XR experience. Participants will hear binaural audio of all participants according to their position and a 360-degree surround sound, if needed. If a physical space is used, the conference server would also receive and process input from one or multiple 360-degree A/V capture devices and RGB+depth camera. Note that when 6DOF is supported, all remote participants can move freely within the confines of the designated space, moving from one room to another when there are multiple rooms defined in the space. Using motion signals, relative positioning and location information, it would be possible for participants (local + remote) to form smaller groups for discussion within the XR space as would happen in a real space. The conversation/real-time XR stream shown in the figure is a mix of VR (remote user) or AR (local user) media, room layout (virtual/physical) and mixed binaural audio. The presentation pointer data may be sent from one of the UEs while presenting a shared presentation/poster for highlighting specific parts.

A top-view of the conference space showing its layout and the current positions of the participants can be viewed by participants and is indicated as part of the XR stream label in the figure (but as separate physical stream). The conference server should also provide IDMS.

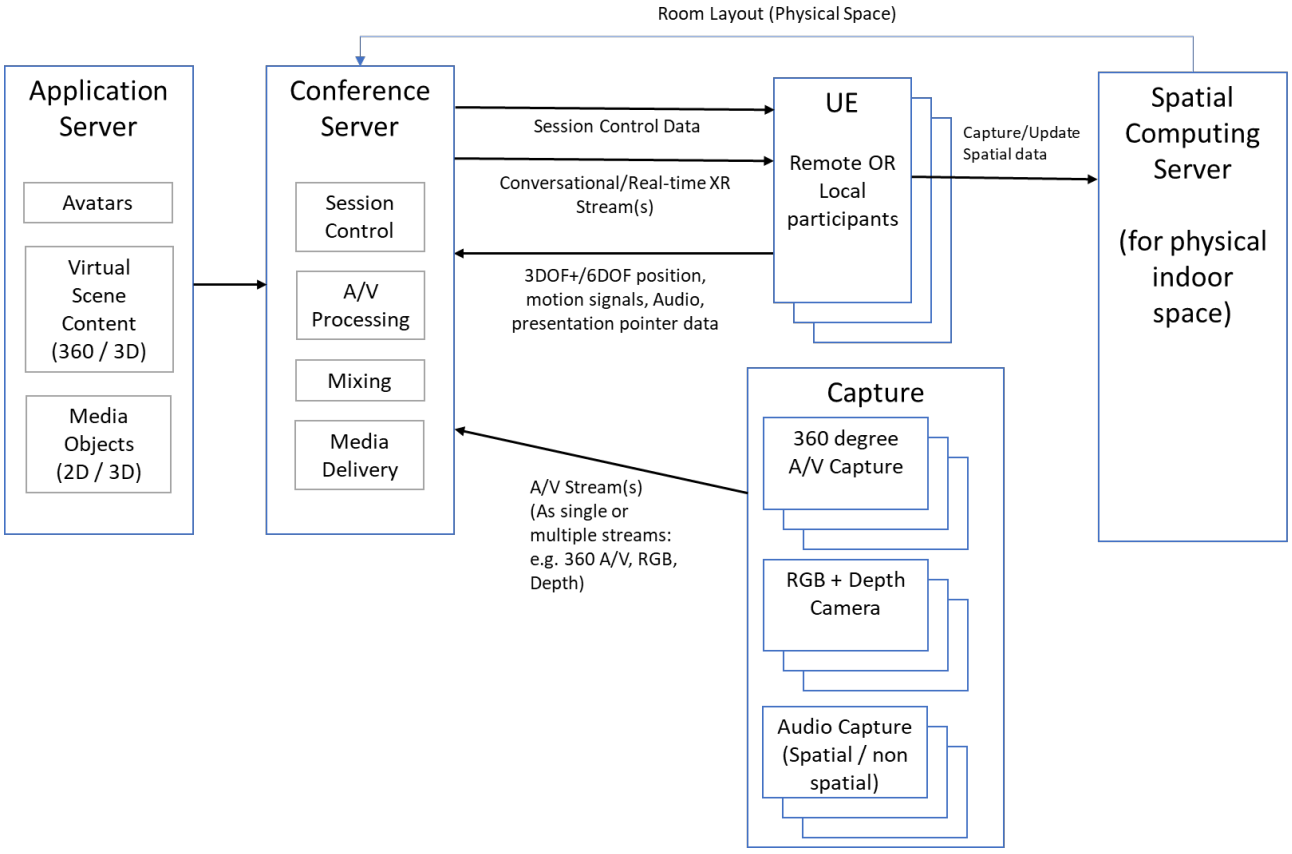


Figure 5.7-1: XR Conference

5.7.3 Potential Normative Work

Table 5.7-1 provides an overview of potential normative work that has been collected as part of the use case collection in Annex A and maps the potential work area to one or several use cases in Annex A.

Table 5.7-1: Overview of potential normative work linked to different conversational/conferencing use-cases in Annex A

Potential Normative Work	Link to Use Case
1. Position and scaling of people	Use Case 12, 360-degree conference meeting: *"Users, need to be scaled and positioned in the AR/VR environment in a natural way"*
2. Background (spacial) audio / picture / video	Use Case 12, 360-degree conference meeting: *"Audio playback needs to match the spatial orientation of the user"*
3. media processing in the network/edge (e.g., cloud rendering, foreground/background segmentation of the user capture, replace HMD of the user with a photo-realistic representation of their face, calibration and synchronization of different camera capture data, etc.)	Use Case 12, 360-degree conference meeting: *"The Data from the rgb+depth camera needs to be acquired and further processed to remove the user from its background"*

4. Support for RGB+Depth video data (in signalling profiles, transmission and associated metadata)	Use Case 12, 360-degree conference meeting: *"each user is captured with an RGB+Depth camera"*
5. DOF metadata framework and a 6DOF capable renderer for immersive voice and audio: • including the Spacial Audio orientation of a speaker • 360 image metadata for associated audio	Use case 15, XR Meeting: *"spatial render of multiple received audio streams according to their associated 6DOF attributes."* Use case 16, Convention / Poster Session: *"the audio and the avatars associated with the virtual users support directivity"*
6. appropriate codec to address XR conversational audio aspects (including directionality of users) and related metadata (e.g. IVAS)	See 5.
7. Synchronized and share state of the shared XR scene or space	Use case 14, 6DOF VR conferencing: *"At the same time the UE of U5 sends its changing position to the conferencing server, which updates the virtual conferencing space with the new coordinates of U5. As the virtual conferencing space is shared, users U1–U4 become aware of moving user U5 and can accordingly adapt their respective renders.*

5.8 Spatial Audio Multiparty Call

5.8.1 Summary of Use Cases

This clause summarizes and integrates the following use cases from Annex A in a single core use case referred to as "Spatial Audio Multiparty Call":

- Use Case 18: AR avatar multi-party calls
- Use Case 19: Front-facing camera video multi-party calls

5.8.2 Description

The system shown in Figure 5.8-1 illustrates an AR multiparty call. Each party can see the other parties using 2D video streams captured by the front facing camera of a mobile phone. Alternatively, they can be displayed as avatars, for instance, when a pair of AR glasses are used as UE. Each party hears spatial audio with the audio of the other parties originating from where their avatar/video is placed on the display. Motion such as head turns are tracked to create a realistic impression that the audio is originating from that virtual direction.

In a special case, the avatars and the audio of the other parties on party A's display is based on their actual geolocation and the relative direction they are with respect to party A. The same would be true for all parties. UEs also have the ability to switch to PTT to suppress surrounding sound if they wish. They may use the "hear what I hear" function to send a 3D audio of their surroundings to the other parties.

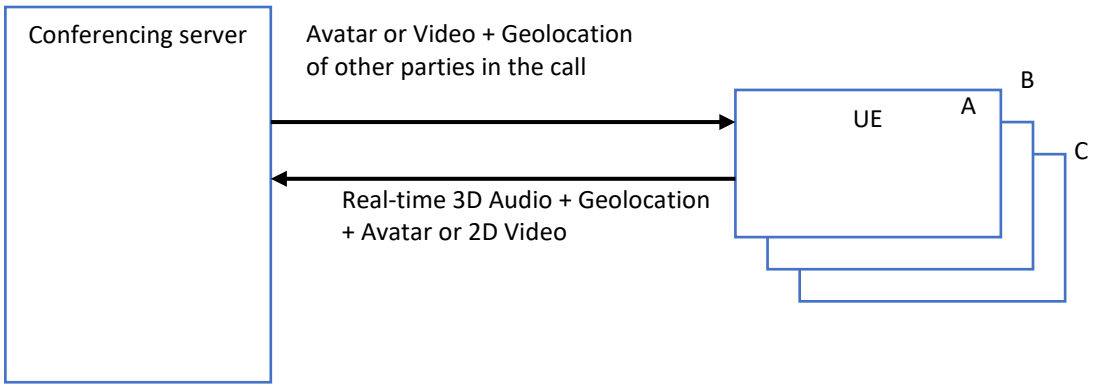


Figure 5.8-1: Spatial Audio Multiparty Call

5.8.3 Potential Normative Work

For this use case, the potential normative work may cover:

- Visual coding and transmission of avatars
- Coding of cut-out heads, alpha channel coding
- Audio coding and transmission of mono objects and 3D audio for streams from all participants.

6 Mapping Extended Reality to 5G

6.1 Introduction

Based on the technologies introduced in clause 4 as well as the core use cases and scenarios introduced in clause 5, this clause maps a set of core technologies to 5G media centric architectures.

6.2 XR Processing and Media Centric Architectures

6.2.1 Introduction

This clause focuses on rendering and media centric architectures. The architectures are simplified and illustrative, they only consider an XR server and an XR device to identify the functions in XR servers and XR devices that communicate and exchange information, possibly over a 5GS communication. The baseline technologies are introduced in clause 4. These architectures focus on processes where the following main tasks are carried out:

- Display
- Tracking and Pose Generation
- Viewport Rendering
- Capture of real-world content
- Media encoding
- Media decoding
- Media content delivery
- 5GS communication
- Media Formats, metadata and other data delivered on communication links

- Spatial Location Processing

The section also discusses benefits and challenges of the different approaches in terms of required bitrates, latencies, reliability, etc. A main aspect to be addressed in the following are the processes that are involved in the motion-to-photon/sound latency and how the processed may impact the XR viewport rendering.

6.2.2 Viewport-Independent delivery

6.2.2.1 Architecture

In the viewport independent delivery case, following the architecture in 3GPP TS 26.118 [3], clause 4.3, tracking and sensor information is only processed in the XR device as shown in Figure 6.2.2-1. This means that the entire XR scene is delivered and decoded.

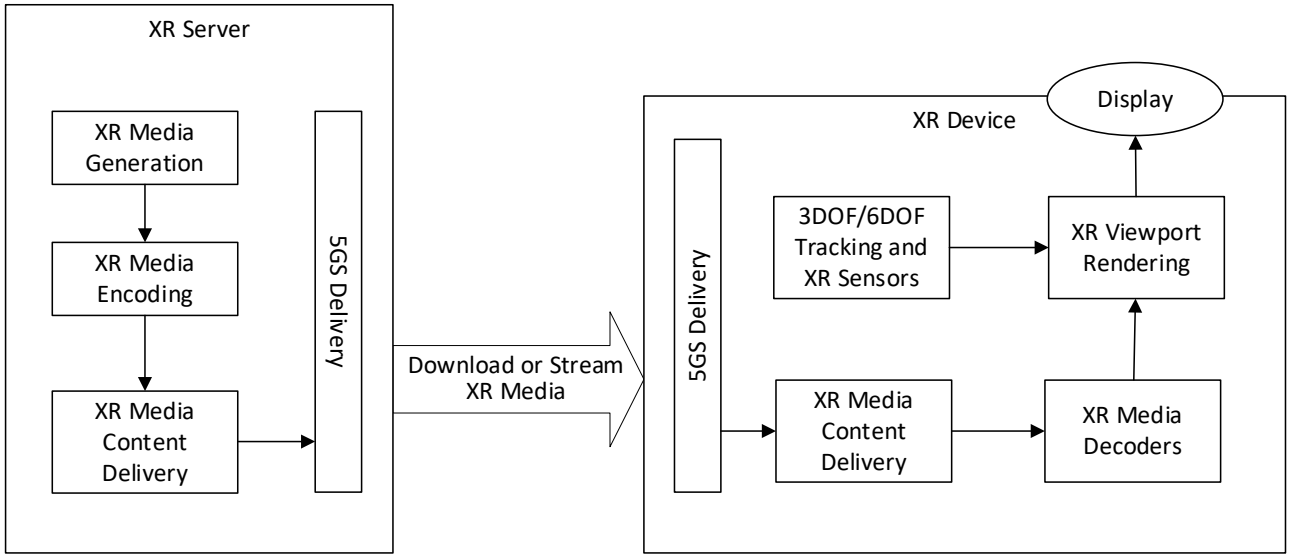


Figure 6.2.2-1: Viewport Independent Delivery

6.2.2.2 Use Cases in Context

Use cases that may be addressed partially or completely by this delivery architecture are summarized in clause 5.4.

6.2.2.3 Basic Procedures

The basic procedures follow the procedures of 5G Media Streaming in 3GPP TS 26.501 [12], clause 5. Both, on-demand and live streaming may be considered.

6.2.2.4 Content Formats and Rendering

No content format for 6DoF streaming is fully defined yet, but the content may for example be a scene for which foreground 3D objects, for example represented by point-clouds, are mixed with background content, for example a omnidirectional scene. The combination of the content may be provided by a scene description that places the object in the 6DoF scene. Typically, the data needs to be streamed into buffers that are jointly rendered.

Due to the significant amount of data that needs to be processed in the device, hardware supported decoding and rendering is necessary.

Such an approach typically requires the delivery and decoding of several video and audio streams in parallel.

6.2.2.5 Relevant QoS and QoE parameters

In the case of viewport-independent streaming, processing of updated pose information is only done locally in the XR device. Delivery latency requirements are independent of the motion-to-photon latency.

In order to provide sufficient content quality, the video material is referably encoded such that the QoE parameters as defined as defined in clause 4.2 can be fulfilled. The necessary QoS and bitrates on the 5G System depend depend on the type of the XR media as well as on the streaming protocol. Based on information from the workshop "Immersive Media meets 5G" in April 2019 as well as from publicly announced demos, that based on today's equipment and the one available over the next 2-3 years, around 100 Mbps are sufficient bitrates to address high-quality 6DOF VR services. This is expected to allow 2k per eye streaming at 90 fps (see clause 4.2) using existing video codecs (see clause 4.5). The QoE requirements may increase further, for example higher resolution and frame rate, but with the advance of new compression tools, this is expected to be compensated.

The XR media delivery are typically built based on download or adaptive streaming such as DASH (see for example TS 26.118 [3] and TS 26.247 [7]), such that one can adjust quality to the available bitrate to a large extent.

Suitable 5QI values for adaptive streaming over HTTP are 6, 8, or 9 as defined in clause 4.3.3.

If other protocols are applied for streaming, then suitable 5QIs would be for FFS.

In the context of the present document, relevant 3D media formats, efficient compression, adaptive delivery as well as the perceived quality of the XR media is of key relevance.

6.2.2.6 Potential Standardisation Needs

In the context of this delivery scenario, the following potential standardisation needs are identified:

- Very high-bitrate and efficient/scalable streaming protocols
- 6DoF Scene Description and XR media integration
- Video and audio codec extensions to efficiently code and render graphics centric formats (2D, meshes, point clouds)
- Support of decoding platforms that support the challenges documented in clause 4.5.2.

6.2.3 Viewport-dependent Streaming

6.2.3.1 Architecture

In the viewport dependent delivery case, following the architecture in TS 26.118 [3], clause 4.3, the tracking information is predominantly processed in the XR device, but the current pose information is provided to the XR delivery engine in order to include the pose information in the adaptive media requests. In an extension to this in the case of XR and 6DoF, the XR pose and additional information may be shared with the XR content delivery in order to only access the information that is relevant for the current viewports. According to Figure 6.2.3-1, the tracking and sensor data is processed in the XR device for XR rendering, and the media is adaptively delivered/requested based on the XR viewport. A reduced or a viewport optimized scene is delivered and also only a reduced scene is processed. Examples include an object that is not visible is not delivered, or only delivered in low quality, or that only the part of the object that is in the viewport is delivered with the highest quality.

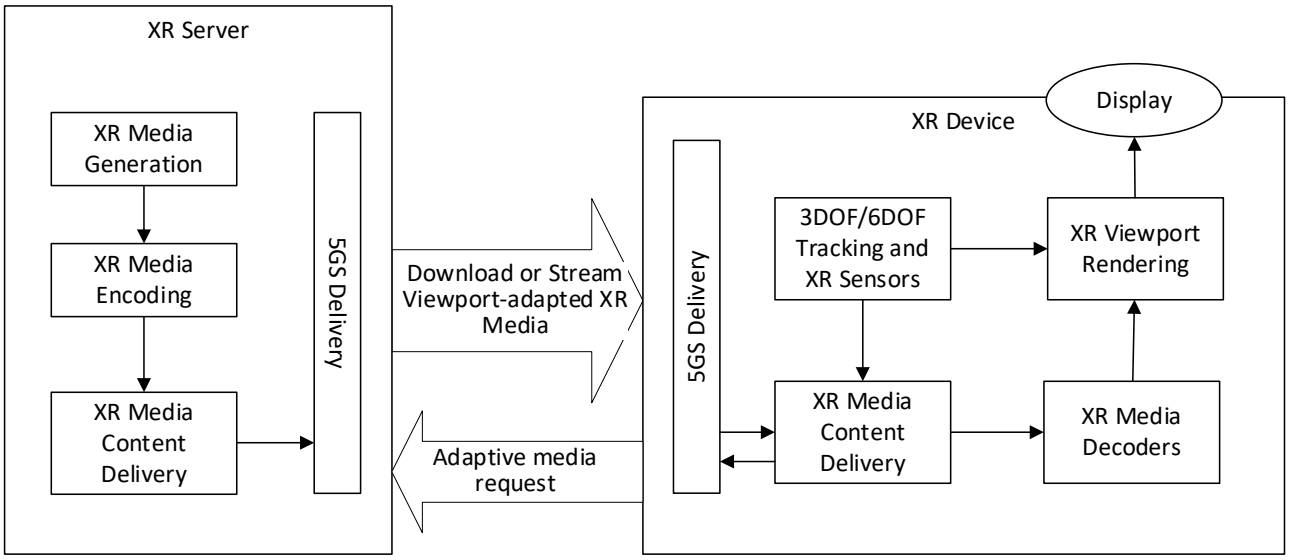


Figure 6.2.3-2 Viewport-dependent Streaming

6.2.3.2 Use Cases in Context

Use cases that may be addressed partially or completely by this delivery architecture are summarized in clause 5.4.

6.2.3.3 Basic Procedures

The basic procedures follow the procedures of 5G Media Streaming in TS 26.501 [12], clause 5. Both, on-demand and live streaming may be considered.

In addition, the request for data is accompanied with information from the XR engine.

6.2.3.4 Content Formats and Rendering

The same formats as discussed in clause 6.2.2.4 apply.

6.2.3.5 Relevant QoS and QoE parameters

Compared to the viewport independent delivery in clause 6.2.2, for viewport dependent streaming, updated tracking and sensor information impacts the network interactivity. Typically, due to updated pose information, HTTP/TCP level information and responses are exchanged every 100-200 ms in viewport-dependent streaming.

From analysis in TR 26.918 [2] and other experience as for example documented the workshop "Immersive Media meets 5G" in April 2019" [42], such approaches can reduce the required bitrate compared to viewport independent streaming by a factor of 2 to 4 at the same rendered quality.

It is important to note that viewport-dependent streaming technologies are typically also built based on adaptive streaming allowing to adjust quality to the available bitrate. The knowledge of tracking information in the XR Delivery receiver just adds another adaptation parameter. However, generally such systems may be flexibly designed taking into account a combination/tradeoff of bitrates, latencies, complexity and quality.

6.2.3.6 Potential Standardisation Needs

In the context of the this architecture, the following potential standardisation needs are identified:

- The same aspects as defined in clause 6.2.2.6
- In addition, more flexible data structures and access to these data structures as well as concurrentdecoding and streaming of smaller units of data, such as tile-based structures, may be defined.

- If other protocols than adaptive streaming over HTTP would be applied, then suitable 5QIs would be for FFS.

6.2.4 Viewport Rendering in Network

6.2.4.1 Overview

In a architecture as shown in figure 6.2.4-1 below, the viewport is entirely rendered in the XR server. The XR server generates the XR Media on the fly based on incoming Tracking and Sensor information, for example a game engine. The generated XR media is provided for the viewport in a 2D format (flattened), encoded and delivered over the 5G network. The tracking and sensor information is delivered in the reverse direction. In the XR device, the media decoders decode the media and the viewport is directly rendered without using the viewport information.

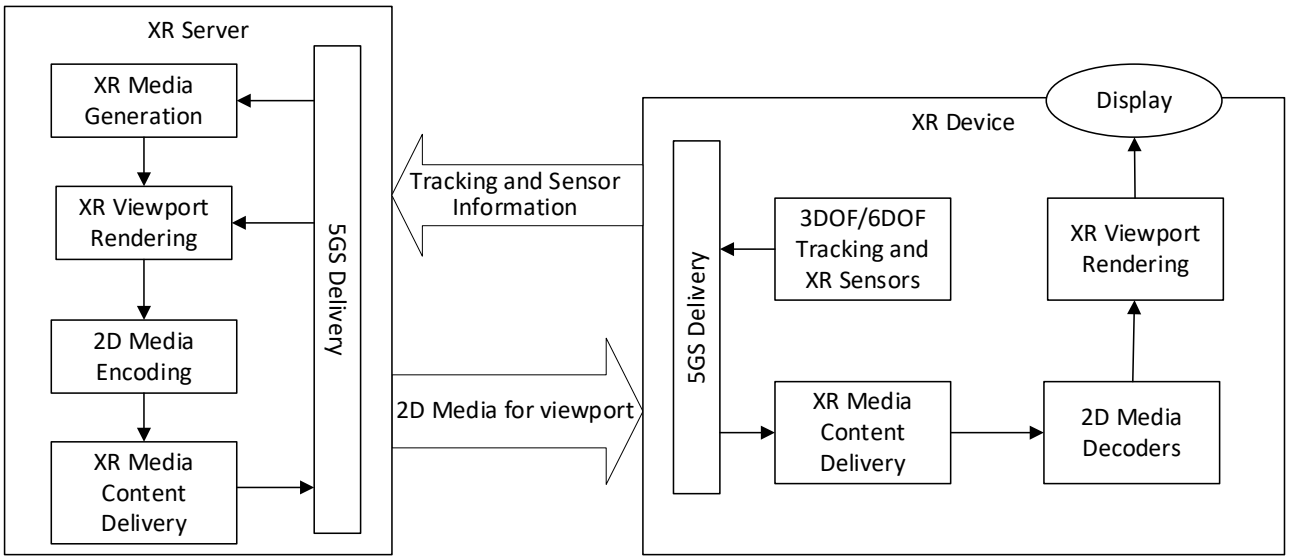


Figure 6.2.4-1 Viewport rendering in Network

The following call flow highlights the key steps:

1) An XR device connects to the network and XR Media Generation application
   a) Sends static XR device information and capabilities (supported decoders, viewport)
2) Based on this information, XR server sets up encoder and formats
3) Loop
   a) XR device collects pose (or a predicted pose)
   b) XR Pose is sent to XR Server
   c) The XR Server uses the pose to generate/compose the viewport
   d) XR Viewport is encoded with regular media encoders
   e) The compressed video is sent to XR Device
   f) The XR device decompresses video and directly renders viewport

Such an architecture enables simple clients, but has significantly challenges on compression and transport to fulfill the latency requirements. Latencies should be kept low for each processing step including delivery, to make sure that the cumulative delay for all the processing steps (including tracking, pose delivery, viewport rendering, media encoding, media delivery, media decoding and display) is within the immersive motion-to-photon latency upper limit of 20ms.

6.2.4.2 Relevant QoS and QoE parameters

The following three cases, with different media delivery bitrates, are considered:

1) Around 100 Mbps: In this case, the XR device needs to perform certain amount of processing and decoding.

2) Around 1 Gbps: In this case, only lightweight and low-latency compression (e.g. intra only) may be used to provide sufficiently high quality (4k or even 8k at sufficiently high frame rates above 60 fps) and sufficiently low latency (immersive limits of less than 20ms for motion to photon) for such applications. It is still expected that some processing (e.g. decoding) by the XR device is needed.

3) Around 10 Gbps or even more: A full "USB-C like" wireless connection, providing functionalities that currently can only be provided by cable, possibly uncompressed formats such as 8K. The processing requirements for the XR device in this case may be minimal.

Note that the lightweight compression or no compression in cases 2) and 3) can help to reduce processing delays.

In addition, the formats exported from Game engines needs to be supported by the respective media encoders.

Note that in this case for XR-based services, the motion-to-photon latency determines the maximum latency requirements for the content. This means that 20ms as defined in clause 4.5.1 are the end-to-end latency targets, including the uplink streaming of the pose information.

This is different, if the content is rendered on a flat device (for example for cloud/edge gaming applications on smartphones), for which not motion-to-photon latency determines the latency requirements, but the roundtrip interaction delay. In this case, the requirements from clause 4.5.2 apply, typically a 50ms latency is a requirement for most advanced games.

6.2.4.3 Potential Standardisation Needs

On Formats and codecs:

- From the analysis, for case 1, similar aspects as defined in clause 6.2.2.6 apply for the formats.
- For cases 2 and 3, formats are of less relevance for 3GPP as such formats are typically defined by other consortia, if at all.

On network support:

- Network rendering for cloud gaming on flat screens is expected to be of significant relevance. In this case the end-to-end latency (action to photon) is determined by the roundtrip interaction delay, i.e. 50ms (see 4.5.2). 5QIs to support such latencies as well as guaranteed bitrates are considered of relevance. Required bitrates follow case 1) from above.
- Network rendering for XR services would require an end-to-end latency including motion-to-photon (including network rendering, encoding, delivery and decoding) of 20ms to meet the immersive limits and it is expected that the bitrates would be higher due to low-complexity and low-latency encoding, following case 2) and 3) from above. Hence,
    - 5QIs and QoS would be necessary, that provides significantly lower latency than 10ms in both directions and the same time provides a stable and high bitrate in the range of 0.1 – 1 Gbps according to case 2).
    - It is not expected to be practical for Uu-based communication to achieve such low-latencies at very high bitrates (mostly case 3, e.g. 1Gbps and higher) in the short term, but final studies on this matter are FFS.
    - However, sidelink-based based communication addressing network rendering is expected to be feasible in the 5G architecture and is subject to active work in 3GPP.

6.2.5 Raster-based Split Rendering

6.2.5.1 Architecture

Raster-based split rendering refers to the case where the XR Server runs an XR engine to generate the XR Scene based on information coming from an XR device. The XR Server rasterizes the XR viewport and does XR pre-rendering.

According to Figure 6.2.5-1, the viewport is pre-dominantly rendered in the XR server, but the device is able to do latest pose correction, for example by asynchronuous time-warping (see clause 4.1) or other XR pose correction to address changes in the pose.

- XR graphics workload is split into rendering workload on a powerful XR server (in the cloud or the edge) and pose correction (such as ATW) on the XR device
- Low motion-to-photon latency is preserved via on device Asynchronous Time Warping (ATW) or other pose correction methods.

As ATW is applied the motion-to-photon latency requirements (of at most 20 ms) are met by XR device internal processing. What determines the network requirements for split rendering is time of pose-to-render-to-photon and the roundtrip interaction delay. According to clause 4.5, the latency is typically 50-60ms. This determines the latency requirements for the 5G delivery.

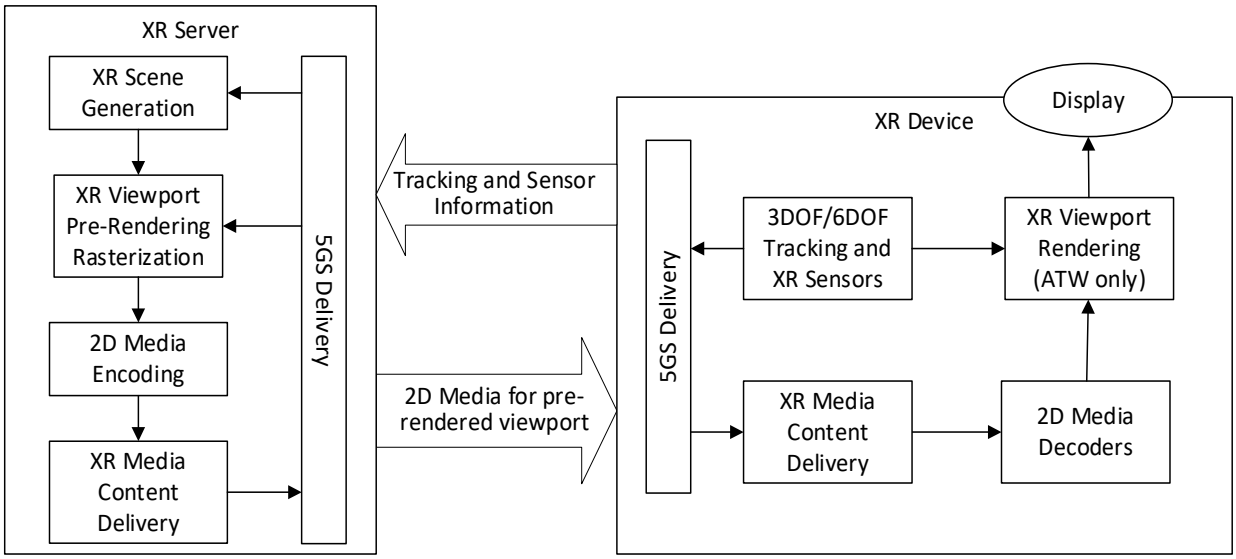


Figure 6.2.5-1: Split Rendering with Asynchronous Time Warping (ATW) Correction

6.2.5.2 Use Cases in Context

The use cases in clause 5.5 may be addressed by this architecture.

6.2.5.3 Basic Procedures

The following call flow highlights the key steps:

1) An XR Device connects to the network and joins XR application
   a) Sends static device information and capabilities (supported decoders, viewport)
2) Based on this information, the XR server sets up encoders and formats
3) Loop
   a) XR Device collects XR pose (or a predicted XR pose)
   b) XR Pose is sent to XR Server
   c) The XR Server uses the pose to pre-render the XR viewport
   d) XR Viewport is encoded with 2D media encoders
   e) The compressed media is sent to XR device along with XR pose that it was rendered for
   f) The XR device decompresses video
   g) The XR device uses the XR pose provided with the video frame and the actual XR pose for an improved prediction using and to correct the local pose, e.g. using ATW.

6.2.5.4 Content Formats and Rendering

Rasterized 3D scenes available in frame buffers (see clause 4.4) are provided by the XR engine and need to be encoded, distributed and decoded. According to clause 4.2.1, relevant formats for frame buffers are 2k by 2k per eye, potentially even higher. Frame rates are expected to be at least 60fps, potentially higher up to 90 fps.The formats of frame buffers are regular texture video signals that are then directly rendered. As the processing is graphics centric, formats beyond commonly used 4:2:0 signals and YUV signals may be considered.

6.2.5.5 Relevant QoS and QoE parameters

With the use of time warp, the latency requirements follow those documented in clause 4.2.2, i.e. the end-to-end latency between the user motion and the rendering is 50ms.

It is known from experiments that with H.264/AVC the bitrates are in the order of 50 Mbps per eye buffer. It is expect that this can be reduced to lower bitrates with improved compression tools (see clause 4.5) but higher quality requirements may absorb the gains. It is also known that this is both content and user movements dependent, but it is known from experiments that 50 - 100 Mbps is a valid target bitrate for split rendering.

Regular stereo audio signals are considered, requiring bitrates that are negligible compared to the video signals.

5QI values exist that may address the use case, such 5QI value number 80 with 10ms, however this is part of the non-GBR bearers (see clause). In addition, it is unclear whether the 10ms with such high bitrates and low required error rates may be too stringent and resource consuming. Hence, for simple split rendering in the context of the requirements in this clause, suitable 5QIs may have to be defined addressing the latency requirements in the range of 10-20ms and bitrate guarantees to be able to stream 50 to 100 Mbps consistently.

The uplink is predominantly the pose information, see clause 4.1 for details. Data rates are several 100 kbit/s and the latency should be small in order to not add to the overall target latency.

6.2.5.6 Potential Standardisation needs

In the context of this architecture, the following potential standardisation needs are identified:

- Regular 2D video encoders and decoders that are capable encode and decode 2K per eye as well as 90 fps and are capable to encode typical graphics frame buffer signals.
- Pose information in the uplink at sufficiently high frequency
- Content Delivery protocols to support the delivery requirements
- Edge computing discovery and capability discovery
- A simple XR split rendering application framework for single buffer streaming
- New 5QIs and QoS support in 5G System for split rendering addressing latency requirements in the range of 10-20ms and bitrate guarantees to be able to stream 50 to 100 Mbps consistently.

6.2.6 Generalized XR Split Rendering

6.2.6.1 Architecture

In Figure 6.2.6-1, an architecture is shown for which the XR server pre-renders the 3D scene into a simpler format to be processed by the device (e.g. it may provide additional metadata that is delivered with the pre-rendered version). The device recovers the baked media and does the final rendering based on local correction on the actual pose.

- XR graphics workload is split into rendering workload on a powerful XR server and simpler XR processing on the XR device
- This approach enables to relax the latency requirements to maintain a full immersive experience as time-critical adjustment to the correct pose is done in the device.
- this approach may provide more flexibility in terms of bitrates, latency requirements, processing, etc. than the single buffer split rendering in clause 6.2.5.

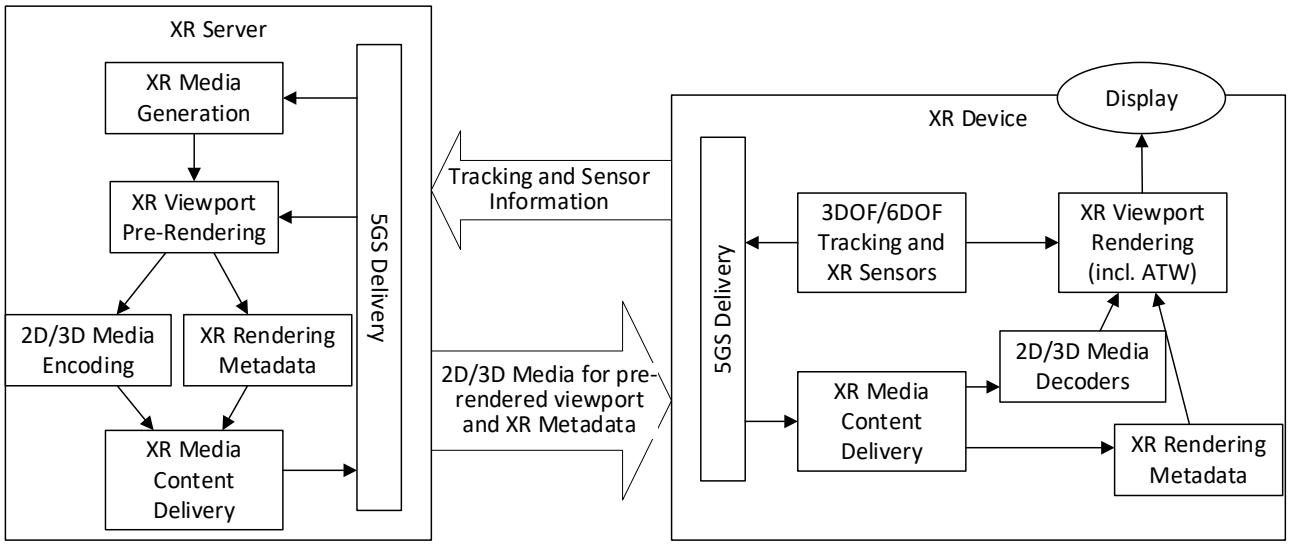


Figure 6.2.6-1: VR Split Rendering with XR Viewport Rendering in Device

Such an approach needs careful considerations on the formats of projected media and their compression with media decoders. Also important is distribution of latencies to different components of the system. More details and breakdown of the architectures is necessary. The interfaces in the device however are aligned with the general structure defined above.

In general, the similar requirements and considerations as in clause 6.2.5 apply, but a more flexible framework may be considered by providing not only 2D frame buffers, but different buffers that are split over the network.

6.2.6.2 Use Cases in Context

The use cases in clause 5.5 may be addressed by this architecture.

6.2.6.3 Basic Procedures

The following call flow highlights the key steps:

1) An XR Device connects to the network and joins XR application
   a) Sends static device information (supported decoders, viewport, supported formats)
2) Based on this information, network server sets up encoder and formats
3) Loop
   a) XR Device collects XR pose (or a predicted XR pose)
   b) XR Pose is sent to XR Server
   c) The XR Server uses the pose to pre-render the XR viewport by creating one or multiple rendering buffers, possibly with different update frequencies
   d) The rendering buffers are encoded with 2D and 3D media encoders
   e) The compressed media is sent to XR device along with additional metadata that describes the media
   f) The XR device decompresses the multiple buffers and adds this to the XR rendering engine.
   g) The XR rendering engine takes the buffers, the rendering pose assigned to the buffers and the latest XR pose to create the finally rendered viewport.

6.2.6.4 Content Formats and Rendering

In this context, the buffers may not only be 2D texture or frame buffers as in case of clause 6.2.5, but may include geometric data, 3D data, meshes and so on. Also multiple objects may be generated. The content formats discussed in clause 4.6 apply.

6.2.6.5 Relevant QoS and QoE parameters

With the use of different buffers, the latency requirements follow those documented in clause 4.5.2, i.e. the end-to-end latency between the user motion and the rendering is 50ms. However, it may well be that the update frequency of certain buffers is less. This may result in differentiated QoS requirements for different encoded media, for example in terms of latency, bitrates, etc.

More details are FFS.

6.2.6.6 Potential Standardisation needs

In the context of the this architecture, the following potential standardisation needs are identified:

- Similar aspects as defined clause 6.2.5.6
- Flexible 2D and 3D formats that can be shared over the network to serve device rendering buffers
- Formats and decoding capabilities as defined in clause 4.5.2
- Edge computing discovery and capability discovery for Generalized Split rendering
- A generalized XR split rendering application framework
- More flexible 5QIs and QoS support in 5G System for generalized split rendering addressing differentiated latency requirements in the range of 10ms up to potentially several 100ms and with bitrate guarantees.

6.2.7 XR Distributed Computing

This clause provides the architecture for extended reality applications which supports the XR split rendering. The workload for XR processing is split into workloads on XR server and the device. The below Figure 6.2.7-1 shows a high-level structure of the XR distributed computing architecture which describes their components and interfaces.

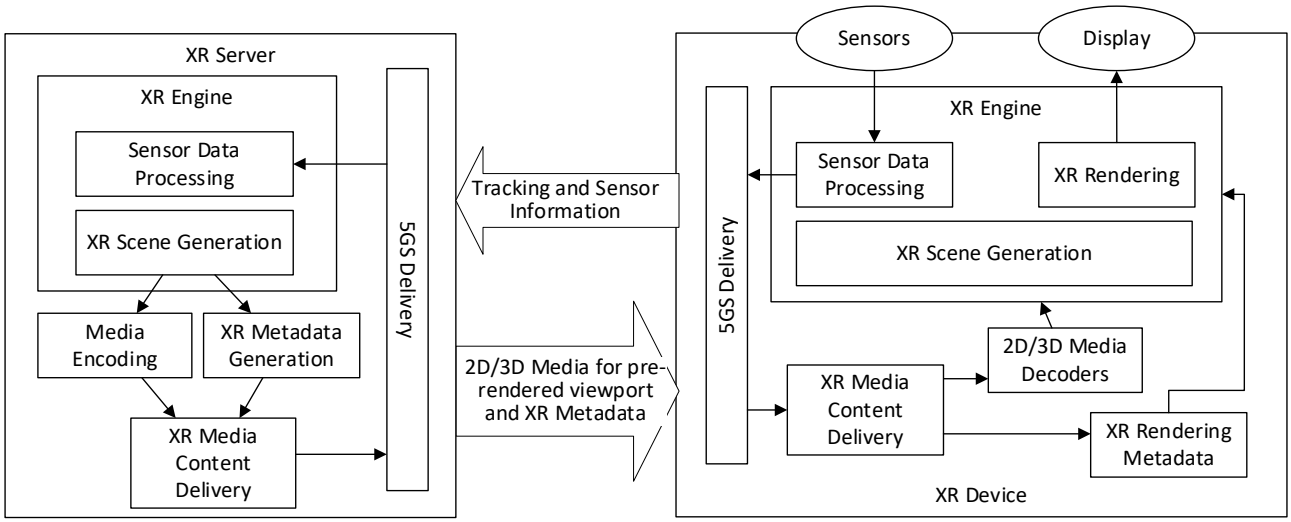


Figure 6.2.7-1: XR Distributed Computing Architecture

An XR client may have following capabilities.

- XR capture
- Sensor data processing (e.g., AR pose tracking)

- XR scene generation
- XR rendering.
- 2D or 3D Media decoding
- Metadata (including scene description) processing
- 5G delivery

An XR edge server may have following capabilities.

- Sensor data processing
- XR scene generation
- 2D or 3D media encoding
- Metadata (including scene description) generation
- 5G delivery

An XR client connects to the network and joins XR rendering application. The XR client sends static device information (e.g., sensors, supported decoders, display configuration) to the XR edge server. Based on this information, the XR edge server sets up encoder and formats.

When the XR client has a set of sensors (e.g., trackers and capturing devices, it collects sensor data from sensors. The collected sensor data is processed either locally or at the XR edge server. The collected sensor data or locally processed information (e.g., a current AR pose) is sent to the XR edge server. The XR edge server uses the information to generate the XR scene. The XR edge server converts the XR scene into a simpler format as 2D or 3D media with metadata (including scene description). The media component is compressed, and the compressed media stream and metadata are delivered to the XR client. The XR client generates the XR scene by compositing locally generated or received media and metadata and renders the XR viewport via the XR display (e.g., HMD, AR glass).

For example, the XR client captures the 2D video stream from a camera and sends the captured stream to the XR edge server. The XR edge server performs the AR tracking and generates the AR scene which a 3D object is overlaid over a certain position in the 2D video based on the AR tracking information. The 3D object or 2D video for the AR scene are encoded with 2D/3D media encoders, and the scene description or the metadata is generated. The compressed media and metadata are sent to the XR client. The XR client decodes the media or metadata and generates an AR scene which overlays the 3D object in the 2D video., A user viewport is determined by horizontal/vertical field of view of the screen of a head-mounted display or any other display device. The appropriate part of AR scene for the user viewport is rendered and displayed.

6.2.8 XR Conversational

In Figure 6.2.8-1, a general architecture for XR conversational and conference services is depicted. As stated, these services are an extension on the current MTSI work, using the IMS for session signalling. In order to support XR conversational services (in 5G), extensions are needed in the signalling to enable VR/AR specific attributes, and the media and metadata need to support the right codecs, profiles and metadata. A new interface is the interface to the network (based) media processing. This is an interface similar to that to an SRF, but is expected to be much more extensive to support various types of media processing. This interface can be based on the work in MPEG on Network Based Media Processing.

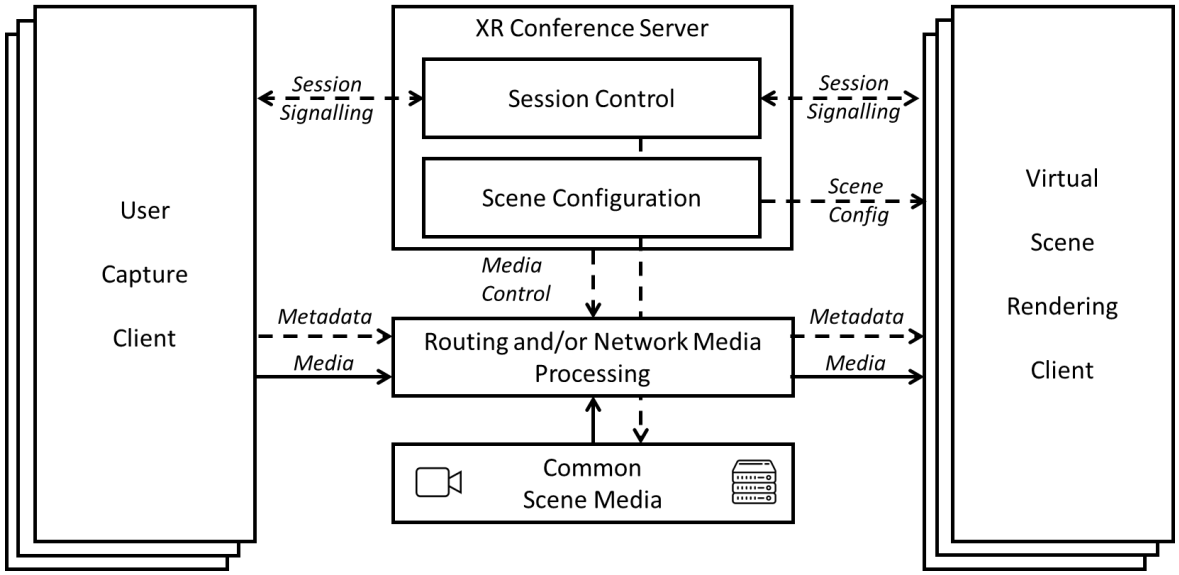


Figure 6.2.8-1 General architecture for XR conversational and conference services

Typical steps for call session setup follow normal IMS procedures, in case the clients have a simple peer-2-peer call and also do all processing themselves (simplified procedure as follows):

1. The first client initiates call setup (SIP INVITE);
2. The IMS (i.e. central session control) routes the call setup to the second client, ensuring proper bandwidth reservations in the network;
3. The second client, after call acceptance, responds to the call setup (200 OK);
4. The network controls all bandwidth reservations;
5. Call setup is completed.

But, given mobile clients, their limited processing capabilities, battery capacity and potentially problems with heat dissipation, processing might be moved to the network. Typical processing for XR conferencing:

- Foreground/background segmentation;
- HMD removal, i.e. replacing a users HMD with a 3D model of the actual face, possibly including eye tracking / reinsertion;
- 3D avatar reconstruction, i.e. using RGB + depth cameras or multiple cameras to create 3D user video avatars;
- Support for multiple users with a (centralised or distributed) VR conferencing bridge, stitching multiple user captures together;
- Creating a self-view, i.e. local 3D user avatar from the user's own perspective.

In such network-processing scenario, setup is somewhat extended:

1. First a client initiates the call setup;
2. Based on the call setup, the session control triggers network based media processing, reserves resources in the network, incl. media processing resources;
3. Session control forwards call setup to the second client;
4. After call acceptance, both first and second client are connected to the network processor.
5. Session control instruct the network processor on the actual processing and the stream forwarding, i.e. which input streams go to which clients.

Specific details here are for further study. Routing media streams can be performed in various ways, using existing methods for stream transcoding, or more centralised conferencing signalling. The interface to the media processor can be based on the existing MRF interface. But, given the developments within MPEG on Network Based Media Processing, maybe a new interface should be defined.

For the routing of signalling, many options allready exist within the IMS. Re-use and perhaps slight modifications are expected to be sufficient to cover the various use cases defined here. For the SDP within the signalling, more modifications are expected. Besides support for new types of media and thus new types of codecs (point cloud streaming, mesh streaming, depth streaming) and profiles for those codecs, new types of metadata also need to be supported. Calibration data (i.e. calibration of HMD position vs camera position), HMD information (position,

orientation, characteristics), camera information (position/orientation, lens/sensor characteristics, settings), user information (3D model, IOD, HRTF, etc) can all be used to perform or improve the media processing.

Also, there are different media types, both for the environment and for the user avatars. A virtual environment can consist of a rendered environment, a 360 photo or video, or some hybrid. User avatars can be graphical avatars, video based, 3D video avatars, rendered avatars. Devices can be a single device (one mobile in an HMD enclosure, potentially with a separate bluetooth camera) or can be multiple devices (separate stand-alone VR HMD, multiple (smartphones as cameras).

Additional aspects to be taken into account are:

- placement of media processor: central vs edge, centralised vs distributed.
- delay aspects for communication purposes. Ideally, delay is kept to a minimum, i.e. <150 ms one-way delay. Given the required processing, this is a challenge, and will effect e.g. codec choices and rendering choices.

For XR Conversational services, we can consider 3 bandwidth cases according to the type of capture/user representation transmitted (with almost constant bandwidth on the upload):

- 2D+/RGBD: >2.7Mbit/s (1 camera), >5.4Mbit/s (2 Camera)
- 3D Mesh: ~30 Mbit/s
- 3D VPCC / GPCC: 5-50 Mbit/s (CTC MPEG)

Furthermore we can assume that joining an communication experience session will result in a download peek at the beginning of the session to download the environment and associated media objects within the XR application. Throughout a XR communication experience session the download might vary depanding on the amound of users represented and the upload format of those users.

6.3 Summary of Traffic Characteristics

This clause summarizes initial typical traffics characteristics for different architectures, based on the findings in this clause. The parameters relate to the 5G PDU session QoS as introduced in clause 4.3.

Table 6.3-1 provides tnitial traffic characteristics for different architectures based on the findings in clause 6.2. Whereas some initial aspects are collected, many issues are FFS.

Legend:

- DL: Downlink,
- UL: Uplink,
- PDB: Packet delay budget,
- PER: Packet Error Rate,
- RTT: Round Trip Time

Note: Either RTT applies, or UL and DL PDB applies separately, but RTT and UL/DL PDB cannot apply simultaneously

Table 6.3-1: Initial Traffic Characteristics for different architectures

Architecture	DL Rate range	UL Rate range	DL PDB	UL PDB	RTT	DL PER range	UL PER range	Traffic periodicity range	Traffic file size distribution
Viewport independent streaming	100 MBPs	HTTP requests every second. TCP handshake	See adaptive streaming	See adaptive streaming	See adaptive streaming and TCP equation	10e-6	10e-6	Almost constant	Almost constant
Viewport dependent streaming	25 MBPs	More frequent HTTP	See adaptive streaming	See adaptive streaming	See adaptive streaming and TCP equation	10e-6	10e-6	Almost constant	Almost constant

		requests every 100ms. TCP handshake							
Viewport Rendering in Network case 1	100 MBit/s	FFS	FFS	FFS	FFS	FFS	FFS	FFS	FFS
Viewport Rendering in Network case 2	1 GBit/s	FFS	FFS	FFS	FFS	FFS	FFS	FFS	FFS
Viewport Rendering in Network case 3	10 Gbit/s	FFS	FFS	FFS	FFS	FFS	FFS	FFS	FFS
Raster-based Split Rendering with Pose Correction	100 Mbit/s	500 kbit/s	20ms	10ms	50ms	FFS	FFS	Almost constant	FFS
Generalized Split Rendering	FFS	FFS	FFS	FFS	FFS	FFS	FFS	FFS	FFS
XR Distributed Computing	FFS	FFS	FFS	FFS	FFS	FFS	FFS	FFS	FFS
XR Conversational	FFS	FFS	FFS	FFS	FFS	FFS	FFS	FFS	FFS
XR Conferencing Details are FFS	3Mbit/s up to 50Mbit/s per user	3Mbit/s up to 50Mbit/s	Allowing real time communication	Allowing real time communication	Allowing real time communication	FFS	FFS	almost constant (with peek during start-up)	> 50Mb at the beginning, depending on media consumption no or almost constant

6.4 Analysis of existing 5QIs

As a summary of the above, existing 5QIs may be used for adaptive streaming over HTTP applications as defined in clause 6.2.2 and 6.2.3.

For other types of services, new 5QIs for Uu-based communication are considered beneficial, among others

- If other protocols than adaptive streaming over HTTP would be applied, then suitable 5QIs would be for FFS.
- New 5QIs and QoS support in 5G System for network and split rendering addressing latency requirements in the range of 10-20ms and bitrate guarantees to be able to stream 50 to 100 Mbps consistently
- More flexible 5QIs and QoS support in 5G System for generalized split rendering addressing differentiated latency requirements in the range of 10ms to several 100ms and with bitrate guarantees.
- Error rates are FFS.
- The data rate, latency and PER for different architectures as introduced in clause 6.3 are FFS.

For sidelink based communication, new PQI/QoS parameters may be defined as well. Details are FFS.

7 Potential Standardisation Areas

7.1 General

This clause documents and clusters potential standardisation areas in the context of this Technical Report.

7.2 XR-Centric Device Types and Architectures

As documented in clause 4.5, XR centric devices are the key enablers for XR services. Key aspects for XR devices are:

- Rendering-centric device architectures using sophisticated GPU functionalities, see clause 4.4.
- Support for Tracking, in particular inside-out tracking in the device
- Heavily power-constrained at least for certain form factors
- Support for multiple decoding formats and parallel decoding of media streams following the challenges documented in clause 4.5.2

In addition, device characteristics can be quite different. Hence, the device types developed in clause 4.8 serve as a starting point for different device types. A more formal definition of XR devices types is considered useful.

In any work considered in 3GPP, end-points compatible to Khronos-based graphics and XR functionalities should be considered. A framework for interfacing device centric XR functionalities with 5G System and radio functionalities is a relevant standardisation effort.

7.3 Extensions to 5G Media Streaming for XR/6DoF Media

Streaming XR and 6DoF is considered in several use cases evaluated in this Technical report. With the establishment of 5G Media Streaming in Release-16 in TS 26.501 and the stage-3 specifications, extensions of 5G Media Streaming to support XR experiences is useful. MPEG develops several new formats and codecs specifically addressing 3D and XR, but also proprietary formats exist that make use of regular hardware supported standardized media codecs and rendering functionalities. All of them have in common that they rely on existing and emerging device architectures that make use of existing video codecs and GPU rendering. In addition, the use of multiple video codecs in parallel is commonly applied. XR/6DoF Streaming is based on CDNs and HTTP delivery, however new functionalities are required.

Extensions to 5G Media streaming may be done in ordered to support the delivery of different XR/DoF media in 5G Systems. Relevant aspects are:

- Additional media decoding capabilities including higher profile and levels, the use of multiple decoders in parallel to interface with rendering architectures, as well more flexible formats following clause 4.5.2
- Support for more flexible delivery protocols allowing parallel download of different objects and parts of objects
- Viewport-dependent streaming as documented in clause 6.2.3
- Potentially new 5QIs and radio capabilities to support higher bitrate streaming
- Biometrics and Emotion Metadata definition and transport

7.4 Raster-based Split Rendering with Pose Correction

Split Rendering is a promising technology to support online gaming in power- and resource constrained devices. Split rendering requires the use of edge computing as the pose-to-render-to-photon is expected to be below 50ms. Rendering of rasterized frame buffers in the network allows to support XR devices with existing codecs using 5G System and radio capabilities. Relevant aspects for single-buffet split rendering include:

- A simple XR split rendering application framework where a single frame buffer per media per eye is shared and final pose correction is done in the XR device

- 2D video encoders and decoders that are capable encode and decode 2K per eye as well as 90 fps as well as typical 2D rasterized graphics centric formats
- Integration of audio into split rendering architectures
- Formats and protocols for XR Pose information delivery and possibly other metadata in the uplink at sufficiently high frequency
- Content Delivery protocols that support split rendering
- 5QIs and other 5GS/Radio capabilities that support split rendering
- Edge computing discovery and capability discovery based on work in SA2 and SA6 (see clause 4.3.6)

7.5 XR conference applications

XR conversational applications within real or computer generated virtual environments is considered in several use cases evaluated in this Technical report. Some work is already ongoing in IVAS, ITT4RT, however, potential additional normative work includes;

- Study the mapping of XR conference applications to the 5G system architecture including Media streaming and MTSI.
- Support for media processing in the network (e.g. NBMP) (e.g. foreground/background segmentation of the user capture, replacement of the user with a photo-realistic representation of their face, etc.)
- 6DOF metadata framework and a 6DOF capable renderer for immersive voice and audio.
- Support of static/dynamic 3D objects' formats and transport for real-time sharing
- Transport of collected data from multiple sensors (e.g. for spatial mapping)
- Format for storing and sharing spatial information (e.g. indoor spatial data).
- Content Delivery protocols that support XR conversational cases
- 5QIs and other 5GS/Radio capabilities that support XR conversational cases
- Edge computing discovery and capability discovery based on work in SA2 and SA6 (see clause 4.3.6)

7.6 Augmented Reality for New Form Factors

Augmented reality was discussed in details in this report. Glass-type AR/MR UEs with standalone capability, i.e., that can be connected directly to 3GPP networks are interesting emerging devices. Such a type is classified as XR5G-A5 in Table 4.3-1. However, it would also be necessary to consider situations where XR5G-A5 UEs have to fall back to XR5G-A1 or XR5G-A2, i.e., to the wired or wirelessly tethered modes, e.g., from NR Uu to 5G sidelink or IEEE 802.11ad/y. Furthermore, an evolution path for devices under the category XR5G-A3 and XR5G-A4 should be considered. Further studies are encouraged, among others

- Basic use cases: a set of use cases relevant for XR5G-A5 can be selected from Table 5.1-1. The preferred cases will be those capable of delivering experiences previous or existing services could not support, e.g., real-time sharing or streaming 3D objects. They also have to be easier to realize in the environments of glasses that are more limited than those of phones.
- Media formats and profiles: for the selected use cases, available formats and profiles of the media/data can be discussed. Sharing of XR and 3D Data is of interest for personal and enterprise use cases as documented in scenario 5.2. Properly generated XR data can be used in AR applications on smartphone devices as well as on AR glasses. Exchange formats for AR-based applications are relevant, for example for services such as MMS.
- Transport technologies and protocols: in case the selected use cases include relocation or delivery of 3D or XR media/data over 3GPP networks, combinations of transport protocols, radio access and core network technologies that support the use cases at relevant QoS can be discussed. If existing technologies and protocols

cannot serve the use cases properly, such gaps can be taken into account in the consideration of normative works.

- Form factor-related issues: in Table 4.3-1, typical maximum transmit power of XR5G-A5 is 0.5-2W while phone types transmit at 3-5 W. However, if XR5G-A5 is implemented in a form factor of typical glasses, i.e., smaller than goggles or HMDs and with a weight less than 100 g, its cellular modems and antennas are located near the face. In this case, XR5G-A5 UEs can have more constraints on transmit power and it would be necessary to develop solutions to overcome it, e.g. considering situations where XR5G-A5 UEs have to fall back to XR5G-A2 from NR Uu to 5G sidelink or IEEE 802.11ad/y. Furthermore, an evolution path for devices under the category XR5G-A3 (3-7W) and XR5G-A4 (2-4W) should be considered.

7.7 Traffic Characteristics and Models for XR Services

As identified in the course of the development of this report, there is significant interest in 3GPP radio and system groups on the traffic characteristics for XR services. This effort should be a prime work for 3GPP to collect realistic traffic characteristics for typical XR services. Of specific interest for other groups in 3GPP is a characterization of traffic of an XR service in the following domains:

- Downlink data rate ranges
- Uplink data rate ranges
- Maximum packet delay budget in uplink and downlink
- Maximum Packet Error Rate,
- Maximum Round Trip Time
- Traffic Characteristics on IP level in uplink and downlink in terms of packet sizes, and temporal characteristics.

Such characteristics are expected to be available for at least the following applications

- Viewport independent 6DoF Streaming
- Viewport dependent 6DoF Streaming
- Single Buffer split rendering for online cloud gaming
- XR conversational services

The Technical Report on Typical Traffic Characteristics in TR 26.925 should be updated to address any findings to support the 3GPP groups.

7.8 Social XR

Social XR is used as an umbrella term for combining, delivering, decoding and rendering XR objects (avatars, conversational, sound sources, streaming live content, etc.) originating from different sources into a single user experience. Social XR may be VR centric, but also may apply to AR and MR.

Social XR is expected to integrate multiple XR functionalities such a 6DoF streaming with XR conversational services. Some normative work may include:

- Social XR Components – Merging of avatar and conversational streams to original media (e.g., overlays, etc.)
- Parallel decoding of multiple independently generated sources.
- Proper annotation and metadata for each object to place it into scene.
- Description and rendering of multiple objects into a Social XR experience.

Details are FFS.

7.9 Generalized Split and Cloud Rendering and Processing

Edge/Cloud processing and rendering is a promising technology to support online gaming in power- and resource constrained devices. Relevant aspects for generalized cloud/split rendering include:

- A generalized XR cloud and split rendering application framework based on a scene description
- Support for 3D formats in split and cloud rendering approaches
- Formats and protocols for XR Pose information delivery and possibly other metadata in the uplink at sufficiently high frequency
- Content Delivery protocols that support generalized split/cloud rendering
- Distributions of processing resources across different resources in the 5G system network, in the application provider domain (cloud) and the XR device.
- Supporting the establishment of Processing Workflows across distributed resources and managing those
- 5QIs and other 5GS/Radio capabilities that support generalized split/cloud rendering by coordination with other groups
- Edge computing discovery and capability discovery based on work in SA2 and SA6 (see clause 4.3.6)

It is recommended that this area is studied in more details to identify key issues.

8 Conclusions and Proposed Next Steps

In this study, frameworks for eXtended Reality (XR) have been analysed. XR referes a larger concept for representing reality that includes the virtual, augmented, and mixed realities. After defining key terms and outlining the QoE/QoS issues of XR-based services, the delivery of XR in the 5G system is discussed, following an architectural model of 5G media streaming defined in TS 26.501. In addition to the conventional service categories, conversational, interactive, streaming, and download, split compute/rendering is identified as a new delivery category. A survey of 3D, XR visual and audio formats was provided.

Use cases and device types have been classified, and processing and media centric architectures are introduced. This includes viewport independent and dependent streaming, as well as different distributed computing architecture for XR. Core use cases of XR include those unique to AR and MR in addition to those of VR discussed in TR 26.918, ranging from offline sharing of 3D objects, real-time sharing, multimedia streaming, online gaming, mission critical applications, and multi-party call/conferences.

Based on the details in the report, the following is proposed:

In the short-term:

Develop a flexible XR centric device reference architecture as well as a collection of device requirements and recommendations for XR device classes based on the considerations in clause 7.2. Device classes should include VR device for 6DoF streaming and XR online gaming (XR5G-V4), as well as AR devices (XR5G-A1, XR5G-A4 and XR5G-A5).

Develop a framework and basic functionalities for raster-based Split Rendering for Online Gaming according to the considerations in clause 7.4.

Document typical XR traffic characteristics in 3GPP TR 26.925 based on the initial considerations in this report, in particular clause 7.7 and support other 3GPP groups in designing systems for XR services and applications.

Address simple extensions to MTSI to support XR conversational services based on the considerations in clause 7.5

Study detailed functionalities and requirements for glass-type AR/MR UEs with standalone capability according to clause 7.6 and addresses exchange formats for AR centric media, taking into account different processing capabilities of AR devices.

In the mid-term:

Based on the work developed in the short-term for raster-based split rendering, an extended Split and Cloud Rendering and Processing should be defined based on the considerations in clause 7.9, preferably preceded by a dedicated study

Address simple extensions to 5G Media Streaming to support 6DoF Streaming based on considerations in clause 7.3. Stage-2 aspects related to TS26.501 should be considered first before starting detailed stage-3 work.

Based on the work developed in the shorter time frame above, address the considerations in more detailed considerations inclause 7.5 and clause 7.8 on Social XR

The work should be carried out in close coordination with other groups in 3GPP on XR related matter, edge computing and rendering as well in communication with experts in MPEG on the MPEG-I project as well as with Khronos on their work on OpenXR, glTF and Vulkan/OpenGL.

Annex A:
Collection of XR Use Cases

A.1 Introduction and Template

In order to collect relevant service scenario and core use cases in the context of XR, this Annex documents collected individual use cases and the established processes to collect those use cases.

The following procedure was applied for adding to the present document:

- There is consensus that the use case is understood, relevant and in scope of the Study Item
- A feasibility study is provided and considered sufficient. Some examples on what is expected on feasibility is provided below.
  - How could the use case be implemented based on technologies available today or expected to be available in a foreseeable timeline, at most within 3 years?
  - What are the technology challenges to make this use case happen?
  - Do you have any implementation information?
    - Demos
    - Proof of concept
    - Existing services
    - References
  - Could a reduced experience of the use case be implemented in an earlier timeframe or is it even available today?
- Beyond use case description and feasibility, the template includes sufficient information on
  - Categorization: Type, Degrees of Freedom, Delivery Type, Device
  - Preconditions: What is necessary to make this work?
  - QoS Considerations: What network capabilities are needed, e.g. bitrate, latency, etc.?
  - QoE Considerations: What is expected that the user is satisfied with the quality?
  - Potential Standardisation Status and Needs: This may include 3GPP relevant standards or external standards

For use cases that are moved to the present document, in the course of the study item, is expected that the following aspects are addressed:

1) The use case is mapped to one or multiple architectures.
2) For each use case the functions and interfaces are defined, and the requirements are developed to address the use case.
3) Specific requirements include
   a) Architectural requirements
   b) Network and QoS requirements
   c) Media Processing requirements
   d) More detailed QoE requirements

The template provided in Table A.1-1 is recommended to be used for this collection.

Table A.1-1 Proposed Use Case Collection Template

Use Case Name
Description
Categorization
Type: AR, VR, XR, MR **Degrees of Freedom: 2D, 3DoF, 3DoF+, OD 6DoF, 6DoF** **Delivery: Download, Streaming, Interactive, Conversational, Split** **Device: Phone, HMD, Glasses, Automotive Heads-up, others**
Preconditions
<provides conditions that are necessary to run the use case, for example support for functionalities on the end device or network>
Requirements and QoS/QoE Considerations
<provides a summary on potential requirements as well as considerations on KPIs/QoE as well as QoS requirements>
Feasibility

<How could the use case be implemented based on technologies available today or expected to be available in a foreseeable timeline, at most within 3 years?
- What are the technology challenges to make this use case happen?
- Do you have any implementation information?
  - Demos
  - Proof of concept
  - Existing services
  - References
- Could a reduced experience of the use case be implemented in an earlier timeframe or is it even available today?
>

Potential Standardization Status and Needs
<identifies potential standardization needs>

Table A.1-2 provides an overview of the use cases and their characterization.

Table A.1-2: Overview of Use cases

No	Use Case	Type	Experience	Delivery	Device
1	3D Image Messaging	AR	3DoF+, 6DoF	Upload and Download	Phone
2	AR Sharing	AR, MR	6DoF	Local, Messaging	Phone

No	Use Case	Type	Experience	Delivery	Device
				Download and Upload	
3	Streaming of Immersive 6DoF	VR	3DoF+, 6DoF	Streaming Interactive Split	HMD with a controller
4	Emotional Streaming	2D, AR and VR	2D, 3DoF+, 6DoF	Streaming Interactive, Split	Phone and HMD
5	Untethered Immersive Online Gaming	VR	6DoF	Streaming, Interactive, Split	HMD with a Gaming controller
6	Immersive Game Spectator Mode	VR	6DoF	Streaming, Split	2D screen or HMD with a controller
7	Real-time 3D Communication	3D, AR	3DoF+	Conversational	Phone
8	AR guided assistant at remote location (industrial services)	2D video with dynamic AR rendering of graphics	6DoF (2D + AR)	Local, Streaming, Interactive, Conversational	5G AR Glasses, 5G touchscreen computer or tablet
9	Police Critical Mission with AR	AR, VR	3DoF to 6DoF	Local, Streaming, Interactive, Conversational, Group Communication	5G AR Glasses/Helmet, VR camera/microphone, Audio stereo headset, 5G accurate positioning
10	Online shopping from a catalogue – downloading	AR	6DoF	Download	AR Glasses, Rendering system, Tablet (or smartphone), Capture device
11	Real-time communication with the shop assistant	AR	6DoF	Interactive, Conversational	AR Glasses, Rendering system, Tablet (or smartphone), Capture device
12	360-degree conference meeting	AR, MR, VR	3DoF	Conversational	Mobile / Laptop
13	3D shared experience	AR, MR, VR	3DoF+ 6DoF	Conversational	Mobile / Laptop
14	6DOF VR conferencing	VR	6DoF	Interactive, Conversational	VR gear with binaural playback and HMD video playback, Call server
15	XR Meeting	AR, VR, XR	6DoF	Interactive Conversational	Phone, HMD, Glasses, headphones
16	Convention / Poster Session	AR, VR, MR	6DoF	Interactive Conversational	Phone, HMD, AR Glasses, VR controller/pointing device, headphones
17	AR animated avatar calls	AR	2D, 3DoF	Conversational	Phone, HMD, Glasses, headphones
18	Online shopping from a catalogue – downloading	AR	6DoF	Download	AR Glasses, Rendering system, Tablet (or

No	Use Case	Type	Experience	Delivery	Device
					smartphone), Capture device
19	Front-facing camera video multi-party calls	AR	3DoF	Conversational	Smartphone with front-facing camera, headset
20	AR Streaming with Localization Registry	AR, Social AR	6DoF	Streaming, Interactive, Conversational	AR glasses with binaural audio playback support
21	Immersive 6DoF Streaming with Social Interaction	VR and Social VR	3DoF+, 6DoF	Streaming Interactive Conversational Split	HMD with a controller
22	5G Online Gaming Party	VR	6DoF	Streaming, Interactive, Split, D2D	HMD with a Gaming controller
23	Spatial Shared Data	AR	6DoF	Streaming Interactive Conversational Split	HMD, AR Glasses

A.2 Use Case 1: 3D Image Messaging

Use Case Description: 3D Image Messaging
Alice uses her phone that is equipped with a depth camera to capture an image of a statue in 3D. The phone captures a set of images and builds a 3D model of the object. After a few seconds the 3D image is ready to share and Alice sends the image to Bob as an MMS message.
Categorization
Type: AR **Degrees of Freedom: 3DoF+ or 6DoF** **Delivery: Upload and Download** **Device: Phone, AR glasses**
Preconditions
- Phone is equipped with 3D capture capabilities, such as depth camera or a stereo camera on the back of the phone, possibly supported by an app for processing multiple images. - Phone is equipped with a 3D image viewer
Requirements and QoS/QoE Considerations
- QoS: Reliable delivery of a File of a few MByte distributed over MMS - QoE: Quality of the 3D object representation, level of details
Feasibility
Some new smartphone releases are equipped with a Time of Flight (ToF) depth camera (see for example https://en.wikipedia.org/wiki/Time-of-flight_camera) that can be used to build accurate 3D models of objects of interest. Compared to structured light cameras, ToF do not require a large baseline to achieve good depth accuracy.

Applications such as 3D Photo are using the stereo camera on the back of some phone models to generate a 3D model using a set of pictures taken consecutively. To compensate for the small baseline, complex processing (e.g. deep model to reconstruct the depth map) may be required.

The 3D image can be stored as a point cloud, a mesh, or a layered image. The content maybe compressed to reduce the message size. The content is identified through its mime type and can be embedded with other content such as text.

Potential Standardization Status and Needs

The following aspects may require standardization work:

- Standardized formats for 3D images, e.g. meshes, point clouds, and/or depth-layered images
- Extensions to MMS to support 3D images

A.3 Use Case 2: AR Sharing

Use Case Description: AR Sharing

Alice is shopping for a new couch at the furniture store close to her. Alice finds a couch that she likes and wants to check Bob's opinion who sits back home. Alice scans a QR code with her phone to download a 3D model of the couch and sends it to Bob via MMS. Bob places the virtual model of the couch on a plane surface in the living room. Bob likes how the couch fits in their living room and captures a 3D picture of the room with the couch and shares it with Alice.

Categorization

Type: AR, MR

Degrees of Freedom: 6DoF

Delivery: Local, Messaging Download and Upload

Device: Phone, AR glasses

Requirements and QoS/QoE Considerations

- QoS: Reliable Delivery (Upload and Download) of a File of a few or several MByte
- QoE: Quality of the 3D object representation, level of details

Preconditions

- Bob's smartphone has support for AR technology

Feasibility

Modeling of sale items in 3D will be increasing. This will facilitate purchase decisions for millions of customers. Texture of the 3D models may vary to reflect available choices for the item.

A user can use ARCore [4] or ARKit [5] to detect flat surfaces and place the 3D model on it. The AR scene can be captured with the real scene in the background and the 3D object in the foreground.

To achieve physically-based rendering (PBR), additional characteristics of the 3D object's texture are stored. These may include properties such as specular, diffuse, transparency, reflectivity, etc.

Potential Standardization Status and Needs

The following aspects may require standardization work:

- Standardized format for 3D objects is needed

- Standardized format for mixed reality 3D scenes is needed
- Extensions to MMS to support sharing of 3D objects and scenes

A.4 Use Case 3: Streaming of Immersive 6DoF

Use Case Description: Streaming of Immersive 6DoF

Alice consumes a recorded highlight of a basketball match being seated close to the court by using an application on the 5G enabled HMD. For this, Alice wears an HMD together with a 6DoF manual controller. The HMD is connected to a 5G network but has no other tethered connection. The 6DoF controller allows to change the viewing position (i.e. the seat) and looking at the action from different angles. In addition, restricted local 6DoF movement of Alice at a location enables to locally interact with the scene based on HMD sensors. Even more the controller allows to rewind, slow mo and pause the scene. In the pause or slow-motion mode, the scene can be viewed from different angles using the controller and head motion. The scene is overlaid with information that helps Alice to navigate through the scene. Alice feels *present* in the scene.

In an extension to the use case, the game is consumed in a live mode.

Categorization

Type: VR

Degrees of Freedom: 3DoF+, 6DoF

Delivery: Streaming, Interactive, Split

Device: HMD with a controller

Preconditions

- Application is installed that permits to consume the scene
- The application uses existing HW capabilities on the device, including A/V decoders, rendering functionalities as well as sensors. Inside-out Tracking is available.
- Media is captured properly and accessible on a server, preferably on a CDN.

Requirements and QoS/QoE Considerations

- Required QoS:
  - Bitrates and Latencies that are sufficient to render the viewport within the immersive limits.
  - Some numbers are provided here: https://www.roadtovr.com/nextvr-latest-tech-is-bringing-new-levels-of-fidelity-to-vr-video/
    - in the best case scenario with 8 Mbps bandwidth, the company can now stream 20 pixels per degree. Keep in mind, that's also in stereo and at 60 FPS
    - plans to roll out this higher-res playback
  - If full 6DoF with presence needs to be enabled, up to 100 Mbit/s may be necessary.
  - However, with viewport adaptive streaming and/or split rendering architectures, the requirements on bitrates may be lower, but the latency requirements may increase. A more detailed study is necessary.
- Required QoE:
  - Fast startup of the service,
  - fast reaction to manual controller information,

- reaction to head and limited body movement within immersive limits,
- seamless experiences when moving across positions
- providing sufficient AV experience to enable *presence*. https://xinreality.com/wiki/Presence
  - highest image quality, stereoscopy
  - should also work in slow motion

Feasibility

Content generated in 6DoF

- 6DoF content is generated by companies such as NextVR ™:
  - https://www.roadtovr.com/nextvr-latest-tech-is-bringing-new-levels-of-fidelity-to-vr-video/
  - https://www.digitaltrends.com/home-theater/nextvr-nba-league-pass-writing-future-of-vr/
  - https://www.vrfocus.com/2019/02/nextvr-and-qualcomm-to-demo-5g-6dof-vr-streaming-at-mwc19/
- "Fearless is designed to play on a 5G enabled handset powered by the Qualcomm Snapdragon 855 Mobile Platform and features six-degrees-of-freedom (6DoF) streaming."
  - https://www.benzinga.com/pressreleases/19/02/r13233697/nextvr-to-demonstrate-6dof-vr-streaming-over-5g-and-new-ar-portal-at-m
  - To create Fearless ™, NextVR™ used a state-of-the-art, proprietary camera that generates the 6DoF volume in ultra-high resolution.
  - This includes 6DoF captured audio and video

Selected Devices/XR Platforms supporting this:

- Oculus Quest ™ is announced https://www.oculus.com/quest/
- Vive Cosmos ™
  - https://uploadvr.com/vive-cosmos-everything-we-know/
- Qualcomm ™ reference design:
  - https://www.vrandfun.com/the-qualcomm-snapdragon-855-will-be-able-to-deliver-up-to-8k-360-video-playback/
  - https://www.roadtovr.com/qualcomm-reference-headset-2x-pixels-vive-pro-ces-2018/
  - https://venturebeat.com/2019/02/25/qualcomms-5g-xr-viewers-will-stop-the-wave-of-mediocre-ar-headsets/
  - https://www.i4u.com/2019/02/130947/qualcomm-pushes-5g-connected-ar-and-vr-viewers

Potential challenges to make this happen within 3 years

- Broadly available high-quality 6DoF and volumetric capturing systems. There are still not enough variety of volumetric content to get a feel for how it would handle more challenging scenes like those with closer and/or faster moving objects
- Broad availability of HMDs and end devices supporting the playback
- Availability of access bandwidth to stream such services

Potential Standardization Status and Needs

The following aspects may require standardization work:

- Coded Representation of Audio/Video Formats as well as geometry data
- Scene composition and description
- Storage and Cloud Access Formats
- Delivery Architectures to support 6DoF Streaming
- Content Delivery and Streaming Protocols
- Decoding, rendering and sensor APIs
- Network conditions that fulfill the QoS and QoE Requirements

A.5 Use Case 4: Emotional Streaming

Use Case Description: Emotional Streaming
Bob is watching a horror movie using a 5G connected HMD. He is fascinated, but his body reaction, eye rolling, and other attributes are collected and are used to create a personalized story line. Movie effects are adjusted for personal preferences while reactions are collected when watching the movie. Bob's emotional reactions determine the story-line. Alice is watching the same story on her newest 5G connected smart phone.
Categorization
Type: 2D interactive, VR and AR **Degrees of Freedom: 2D, 3DoF+, 6DoF** **Delivery: Streaming, Interactive, Split** **Device: Phone and HMD**
Preconditions
- Application is installed that permits to consume the story - The application uses existing HW capabilities on the device, including A/V decoders, rendering functionalities as well as sensors - The application uses AI functionalities to extract personalized reactions based on sensor tracking
Requirements and QoS/QoE Considerations

- QoS:
  - Bitrates and Latencies that are sufficient to render the viewport within the immersive limits or at least to react to the emotions
- QoE:
  - Sufficiently fast reaction to body emotion feedback,
  - for HMD, reaction to head movement within immersive limits,
  - providing sufficient AV experience to enable presence.
  - Streaming with seamless transitions from one scene to either of the choices

Feasibility

https://www.cnet.com/news/with-5g-you-wont-just-be-watching-video-itll-be-watching-you-too/

Interactive and branching content

- Netflix's Bandersnatch ™ provides an example for content interactive streaming.
- Also games use similar decision making trees. Examples are provided here:
  - http://skipabeatgame.com/
  - https://www.digitaltrends.com/cool-tech/bring-to-light-heart-rate-vr/
  - https://vrscout.com/news/vr-horror-game-tracks-heart-rate/

Device Features

- Facial expression tracking with AI is available on mobile devices
- Eye Tracking combined with AI is available on mobile devices
- IoT/Wearable devices provide the ability to measure biometric metrics such as heart beat and other stress detecting factors (skin changes, etc.) and may be connected with app

Biometric and Emotion Tracking Technologies are summarized:

- https://blog.therachat.io/emotion-tracking/
- https://www.aplanforliving.com/6-wearables-to-track-your-emotions/
- https://www.inc.com/magazine/201607/tom-foster/lightwave-monitor-customer-emotions.html

Potential Standardization Status and Needs

The following aspects may require standardization work:

- Coded Representation of Audio/Video Formats
- Seamless splicing and smooth transitions across storylines
- Scene composition and description
- Storage and Cloud Access Formats
- Content Delivery Protocols
- Decoding, rendering, sensor and emotion tracking APIs
- Biometrics and Emotion Metadata definition and delivery

A.6 Use Case 5: Untethered Immersive Online Gaming

Use Case Description: Untethered Immersive Online Gaming

100 friends play Fortnite Battle Royal ™. Of the the 100 friends, several are on travel and connect on a stand-alone HMD. The HMD has a with 5G connection.

Fortnite Battle Royale ™ is a free-to-play battle royale video game. As a battle royale game, Fortnite Battle Royale features up to 100 players, alone, in duos, or in squads of up to four players, attempting to be the last player or group alive by killing other players or evading them, while staying within a constantly shrinking safe zone to prevent taking lethal damage from being outside it. Players start with no intrinsic advantages, and scavenge for weapons and armor to gain the upper hand on their opponents. The game features cross-platform play between the platforms that was limited for the first five seasons, before the restrictions were eased.

Other popular VR games are here:

- https://veer.tv/blog/30-best-vr-games-for-playstation-vr-oculus-rift-htc-vive-in-2018/
- https://uploadvr.com/best-psvr-games/
- https://www.digitaltrends.com/gaming/best-psvr-games/
- Population: One ™
  - https://www.ign.com/articles/2019/01/11/population-one-isnt-quite-fortnite-vr-but-its-pretty-convincing
  - http://www.populationonevr.com/
  - https://uploadvr.com/ces-population-one-preview/
- https://vrgames.io/game/

Categorization

Type: VR

Degrees of Freedom: 6DoF

Delivery: Streaming, Interactive, Split

Device: HMD with a Gaming controller

Preconditions

- Gaming client is installed that permits to consume the game
- The application uses existing HW capabilities on the device, including game engines, rendering functionalities as well as sensors. Inside-out Tracking is available.
- Connectivity to the network is provided.

Requirements and QoS/QoE Considerations

- Collected Statistics:
  - https://www.zdnet.com/article/how-fortnite-approaches-analytics-cloud-to-analyze-petabytes-of-game-data/
  - Fortnite ™ processes 92 million events a minute and sees its data grow 2 petabytes a month
  - Akamai ™ said Fortnite set a game traffic record on its network July 12 with 37 terabytes
  - https://www.techadvisor.co.uk/feature/game/how-much-data-does-fortnite-use-3683618/per second delivered across its platform.
    - Techadvisor IDG © Copyright 2019 IDG UK. All Rights Reserved has checked our data usage, and according to the tool, the 15-minute session used 12.4MB of mobile data. That may sound like a lot, but it's the equivalent of streaming a one- or two-minute video on YouTube ™. It may vary slightly depending on a number of factors, but Techadvisor IDG © estimate that Fortnite uses between 10-15MB per 15 minutes of gameplay, or around 50-60MB per hour.
- Required QoS:
  - https://broadbandnow.com/guides/best-internet-service-setup-serious-gamers
    - Any connection over 2 mbps with less than 75ms ping should work well for 99% of games.
    - the main factors affecting your gameplay are:
      - Efficiency of your network

- Distance to other players in multiplayer games
- QoS and network prioritization might not matter much for the average Internet user, but for gamers it can make a big difference in network lag.
- Ping is king.

- Different scenarios need to be looked at, for example where the rendering is happening.

- Required QoE:
  - fast reaction to manual controller information,
  - reaction to head movement within immersive limits,
  - providing sufficient gaming rendering experience to enable *presence*.
    - https://xinreality.com/wiki/Presence
  - supporting frame rate not lower than 60 FPS and resolution not lower than 8K

The TR 22.842 [6] provides some information as well, please refer to clause 5.3.1. Summary of some discussions:

- Latency requirements for online games may be very tight. Examples
  - Current mainstream FPS (First Person Shooter) game requires 60 frames per second, which means frame interval is 16.67ms. If rendering is done in the cloud and taking out the delay for rendering and encoding/decoding processing, the network round trip time (RTT) delay should be less than 5ms.
  - MOBA (Multiplayer Online Battle Arena) game requires 20ms RTT.
- Resolutions and frame rates need to be sufficiently high: higher than 60 FPS and 8K resolution
- Packet loss rates should be low as game experiences degrade quickly

And some references from TR 22.842

- O. Abari, D. Bharadia, A. Duffield, and D. Katabi, "Cutting the Cord in Virtual Reality," in Proceedings of the 15th ACM Workshop on Hot Topics in Networks. ACM, 2016, pp. 162–168.
- E. Bastug, M. Bennis, M. Médard, and M. Debbah, "Toward Interconnected Virtual Reality: Opportunities, Challenges, and Enablers," IEEE Communications Magazine, vol. 55, no. 6, pp. 110–117, 2017.
- Athul Prasad, Mikko A. Uusitalo, David Navrátil, and Mikko Säily, "Challenges for Enabling Virtual Reality Broadcast Using 5G Small Cell Network", IEEE Wireless Communications and Networking Conference Workshops, pp. 220-225, 2018.
- Mohammed S. Elbamby, Cristina Perfecto, Mehdi Bennis, and Klaus Doppler, "Toward Low-Latency and Ultra-Reliable Virtual Reality", IEEE Network, March/April, 2018.
- Orlosky, Jason & Kiyokawa, Kiyoshi & Takemura, Haruo, "Virtual and Augmented Reality on the 5G Highway", Journal of Information Processing, 25. 133-141. 10.2197/ipsjjip.25.133.
- J. Huang, Z. Chen, D. Ceylan and H. Jin, "6-DOF VR videos with a single 360-camera", 2017 IEEE Virtual Reality (VR), Los Angeles, CA, 2017, pp. 37-44. doi: 10.1109/VR.2017.7892229.
- Impact of Packet Losses on the Quality of Video Streaming, https://www.diva-portal.org/smash/get/diva2:831420/FULLTEXT01.pdf
- New Study from GSMA and CAICT Forecasts That China Will Be the World's Largest 5G Market by 2025, https://www.webscalenetworking.com/news/2017/06/27/8571173.htm

Feasibility

Available Games

- Fortnite ™ is available as a game and can be downloaded

- Other VR games are also available or in beta:
  - Population: One ™
    - https://www.ign.com/articles/2019/01/11/population-one-isnt-quite-fortnite-vr-but-its-pretty-convincing
    - http://www.populationonevr.com/
    - https://uploadvr.com/ces-population-one-preview/
  - https://vrgames.io/game/

Selected Devices/XR Platforms supporting this:

- Oculus Rift ™, Playstation VR ™, HTC Vive ™
  - These are tethered and connected devices
  - Specifications are here: https://www.digitaltrends.com/virtual-reality/oculus-rift-vs-htc-vive/
    - Oculus Go ™
    - Oculus Quest ™ is announced https://www.oculus.com/quest/

An important aspect is that the processing power of untethered devices is typically lower as all processing needs to be done on the device. The feasibility is likely improved by supporting the device with additional network processing.

Demos and Architectures are provided that show cloud and split rendering:

- NVIDIA ™ Cloud Rendering: https://www.nvidia.com/object/gpu-cloud-rendering.html
- Google ™ Cloud Rendering: https://www.zyncrender.com/
- Split Rendering: https://www.qualcomm.com/news/onq/2018/09/18/boundless-xr-new-era-distributed-computing

Potential Challenges:

- Getting end-to-end workflow in place
- Operational costs

Potential Standardization Status and Needs

The following aspects may require standardization work:

- Network conditions that fulfill the QoS and QoE Requirements
- Content Delivery Protocols
- Decoding, rendering and sensor APIs
- Architectures for computing support in the network
- TR 22.842 provides a gap analysis in clause 5.3.6 that is in line with these needs

A.7 Use Case 6: Immersive Game Spectator Mode

Use Case Description: Immersive Game Spectator Mode

The world championship in Fortnite ™ are happening and the 100 best players meet. Millions of people want to follow the game online and connect to the live game streaming. Many of them connect over a 5G connected HMD

and follow the game. The users can change their in-game position by using controllers and body movement. Two types of positions are possible:

- Getting the exact view of one of the participants
- A spectactor view independent of the player view

Other users follow on a 2D screen.

In an extension of the game, the spectators "interact" with the players and the scene in a sense that the players hear cheering, get rewarded by presence of spectators, similar to a stadium experience.

The Twitch.TV ™ experience is also available for standalone 5G connected devices.

Categorization

Type: VR

Degrees of Freedom: 6DoF

Delivery: Streaming, Split

Device: 2D screen or HMD with a controller

Preconditions

- Application is installed that permits to follow the game
- The application uses existing HW capabilities on the device, including A/V decoders, rendering functionalities as well as sensors. Inside-out Tracking is available.
- A serving architecture is available that provides access to the game
- The game is rendered in the network

Requirements and QoS/QoE Considerations

- Required QoS:
  - Depends on the architecture, but similar considerations as for the Use Case 5 in A.6
- Required QoE:
  - Being timely close to the live gaming experience, in the extension, presence needs to provide a live participation experience.
  - fast reaction to manual controller information,
  - reaction to head movement within immersive limits,
  - providing sufficient AV experience to enable *presence*. https://xinreality.com/wiki/Presence

Feasibility

Twitch shows that games are watched live with incredible statistics (https://sullygnome.com/):

- Fortnite ™ has 1,412,048,240 watching hours over 365 days, this means it is more than 160,000 years

Spectator Mode in VR Games

- https://techcrunch.com/2018/11/09/can-the-startup-building-a-fortnite-for-vr-become-the-fortnite-of-vr/ , see towards the end

Similar considerations as for use case 5 in clause A.6.

Potential Standardization Status and Needs

The following aspects may require standardization work:

- Coded Representation of Audio/Video Formats
- Content Delivery Protocols
- Decoding, rendering and sensor APIs
- Network conditions that fulfill the QoS and QoE Requirement
- Architectures and interfaces that permit such experiences

A.8 Use Case 7: Real-time 3D Communication

Use Case Description: Real-time 3D Communication

Alice uses her mobile phone to start a video call with Bob. After the call starts, Alice sees a button on her screen that reads "3D". Alice clicks on the button to turn on the 3D mode on the video call app. Bob is able to see Alice's head in 3D and he uses his finger to rotate the view and look around Alice's head. Bob may not be able to see the full head or may see a reconstructed model of it (e.g. based on a pre-captured model). Alice is able to apply a selected set of 3D AR effects to her 3D head (e.g. putting a hat or glasses).

Categorization

Type: 3D Real-time communication, AR

Degrees of Freedom: 3DoF+

Delivery: Conversational

Device: Phone, AR glasses

Preconditions

- Alice's phone is equipped with 3D capture capabilities, such as front depth camera
- Bob's phone can receive a proper 3D object in real-time and apply the facial expressions during the rendering

Requirements and QoS/QoE Considerations

- QoS:
  - conversational QoS requirements
  - sufficient bandwidth to delivery compressed 3D objects, e.g. point cloud compression
- QoE:
  - Quality of the 3D object representation, level of details
  - Quality of facial expressions

The following requirements are considered:

- High quality, very low delay 3D reconstruction of Head/Face, e.g. resolution of the 3D head representation measured in number of points or polygons

Feasibility

Advances in image and video processing together with the proliferation of front-facing depth sensors are going to enable real-time reconstruction of the call participants. To run in real-time, extensive hardware capabilities are

required, such as multi-GPU or Tensor Processing Unit (TPU) processing. These operations may be performed in the network, e.g. by a media gateway or a dedicated processing engine.

The representation of the call participant's head can be done in Point Cloud format to avoid the expensive Mesh reconstruction operation.

Potential Standardization Status and Needs

The following aspects may require standardization work:

- Extension of the MTSI service to support dynamic 3D objects and their formats

A.9 Use Case 8: AR guided assistant at remote location (industrial services)

Use Case Description: AR guided assistant at remote location (industrial services)

- Pedro is sent to fix a machine in a remote location.
- Fixing the machine requires support from a remote expert.
- Pedro puts his AR 5G glasses on and turns them on. He connects to the remote expert, who uses a tablet or a touch-screen computer, or uses AR glasses, headphones, as well as a gesture acquisition device that is connected and coordinated with his glasses.
- The connection supports conversational audio and Pedro and the expert start a conversation.
- Pedro's AR 5G glasses support accurate positioning and Pedro's position is shared live with the expert such that he can direct Pedro in the location.
- The AR 5G glasses are equipped with a camera that also has depth capturing capability.
- Pedro activates the camera such that the expert can see what Pedro is viewing.
- The expert can provide guidance to Pedro via audio but also via overlaying graphics to the received video content, by activation of appropriate automatic object detection from his application, and via drawing of instructions as text and/or graphics and via overlaying additional video instructions. In the case that the expert uses AR glasses, the expert can also identify the depth of the video sent by Pedro and more accurately place the overlay text or graphics.
- The overlaid text and/or graphics are sent to Pedro's glasses and they are rendered to Pedro such that he receives the visual guidance from the expert on where to find the machine and how to fix it.
- Note: the video uplink from Pedro's glasses might be "jumpy" as Pedro moves his head. A second camera and corresponding video uplink to show an overview video of Pedro and the machinery or alternatively a detailed video of the machinery functioning, is a help to the expert when performing this type of service.

Categorization

Type: AR

Degrees of Freedom: 2D video with dynamic AR rendering of graphics (6DoF)

Delivery: Local, Streaming, Interactive, Conversational

Device: 5G AR Glasses, 5G touchscreen computer or tablet

Preconditions

Pedro has AR Glasses with the following features

- 5G connectivity
- Support for conversational audio
- Positioning (possibly even indoor)
- Camera with depth capturing
- Rendering of overlay graphics
- Rendering of overlay video

The remote expert has a tablet or touch-screen device (with peripheries) with the following features

- Securily connected to Pedro
- Headphones
- Gesture acquisition
- Composition tools to support Pedro
- Access to a second stationary camera that is provides synchronized video to Pedro's uplink traffic

Requirements and QoS/QoE Considerations

QoS:

- conversational QoS requirements
- sufficient bandwidth to delivery compressed 3D objects, e.g. point cloud compression
- Accurate user location (indoor/outdoor) (to find machine or user location)

QoE:

- For Pedro:
  - Fast and accurate rendering of overlay graphics and video
  - Synchronized rendering of audio and video/graphics
- For remote expert:
  - High-quality depth video captured from Pedro's device
  - Synchronized and good video signal from second camera
  - Synchronized voice communication from Pedro
  - Accurate positioning information

Feasibility

- Vuzix Blade ™ AR glasses with WiFi connectivity to a smartphone with 4G connectivity
- Specific applications. For example:
  - https://www.vuzix.com/appstore/app/gemvision (Remote assistance for hands-on workforce)
  - https://play.google.com/store/apps/details?id=com.utilityar.workflow (Remote Adviser from Utility AR)
  - https://www.youtube.com/watch?v=d3YT8j0yYl0 (Dynamics 365 Remote Assist + MS HoloLens 2™)
  - https://www.youtube.com/watch?v=lzYg32ngWmU&t=9s (Assistance from virtual AR trainer)

Potential Standardization Status and Needs

- 5G connectivity: Release-15 and Release-16 3GPP standardization

- 5G positioning: ongoing 3GPP standardization – API required for sharing with low latency
- MTSI regular audio between Pedro and expert
- MTSI 2D video call from Pedro to expert, potentially a second video source as help for the expert.
- Pedro received video + graphics (manuals, catalogs, manual indications from the expert, object detection) + overlaid video rendering either in the network or locally
- Synchronization of different capturing devices
- Coded Representations of 3D depth signals and delivery in MTSI context

A.10 Use Case 9: Police Critical Mission with AR

Use Case Name: Police Critical Mission with AR

- A squad team of police officers (Hugo, Paco and Luis) are sent to a dangerous location to perform a task, for instance, a rescue mission
- Each team member is equipped with a helmet with:
  - AR displays (or AR Glasses),
  - stereo headphones with embedded microphones for capturing the surrounding sound and a microphone for conversational purposes (see audio sub- use case below)
  - VR360 camera, e.g. double fish eye or a more advance camera array in such way that are located in surface of the helmet (for safety reasons)
  - 5G connectivity and very accurate 5G location
- Each team member can talk each other via PTT or duplex communication
- Each team capture and deliver VR video with extremely low latency to central police.
  - A lower quality may be sent to lower the latency requirement
  - A high quality is stream up for recording purposes
  - Surround sound maybe capture as well.
- The squad team can be backed up by one or more drones relaying 360 VR video, hyper-sensorial data, and enabling XR haptics.
  - Squad team members can augment their surroundings with drone data.
  - Squad team members can extend their physical presence by taking over control of one or more drones.
  - Police central operations can extend their physical presence by taking over control of one or more drones.
- At the police central facilities, they can see each VR360 camera and have communication to all members of the team
  - Each squad team may have a counterpart (person) who is monitoring VR360 camera using HMD so can assist for dangerous situation outside of its field of view. This may be an automated process too that signal Graphics information of an incoming danger.
- The central facilities may share additional information to every team member such maps, routes, location of possible danger and additional information via text or simple graphics

- Each team member shared their accurate positioning to each team and can be displayed/indicated in the AR display (e.g. showing that someone is behind a wall)
- Each camera VR capture is analyzed in real time to identify moving objects and shared to others team members (as point above)

Audio

- Each team communicates via microphone, and automatic Speech to text can be generated so it is rendered in AR display in case of noisy conditions
- Stereo communication is needed to enhance the intelligibility
- Since each team is wearing stereo headset
  - Microphones are place near speakers to capture the surround noise and it is feedback (with no latency) to each earpiece.
  - The receiving audio of each team member is 3D spatially placed (e.g. in front or in the direction where the other team members are located) so the user does not get distracted from the surround sound environment. (this audio is mixed with the microphone feedback)

Categorization

Type: AR, VR

Degrees of Freedom: 3DoF to 6DoF

Delivery: Local, Streaming, Interactive, Conversational, Group Communication

Device: 5G AR Glasses/Helmet, VR camera/microphone, Audio stereo headset, 5G accurate positioning

Preconditions

- AR 5G Glasses/Helmet
- VR camera and microphone capture
- 5G connectivity and positioning
- Real time communication
- One or more drones relaying 360 VR video, hyper-sensorial data, and enabling XR haptics

Requirements and QoS/QoE Considerations

- Accurate user location (indoor/outdoor)
- Low latency
- High bandwidth

Feasibility

- There are a few devices available today that target some of the requirements described in this Use Case, e.g. "HUD 3.0", a military HMD that projects critical data to the soldier's field of view (https://www.popularmechanics.com/military/a19635016/us-troops-to-test-augmented-reality-by-2019/). With the new announcement of the HoloLens 2 from Microsoft with more advance technology for AR applications and better rendering quality makes it easy to create a proof of concept for this use case (ignoring form factors and security requirements for police helmet). The HoloLens 2 features are described here: https://pureinfotech.com/microsoft-hololens-2-tech-specs/. The device can use WIFI connectivity to connect to a 5G device. A VR camera can easily be mounted to the for proof of concept.

Potential Standardization Status and Needs

- 5G connectivity with dedicated slices for high resilience on critical communications

- 5G positioning
- MTSI/MCPTT SWB/FB voice communication
- MTSI/FLUS uplink 3D audio
- MTSI/FLUS uplink VR
- Downlink AR video with overlaid graphics with local/cloud computation and rendering
- Downlink AR audio with mixed-in 3D audio objects with local/cloud computation and rendering

A.11 Use Case 10: Online shopping from a catalogue – downloading

Use Case Description: Online shopping from a catalogue – downloading
In order to purchase a new sofa for his living room, John connects to an online shop offering the ability to virtually insert items in his home place. This online shop provides for each selling product, 2D images, 3D objects models and detailed information on size, colour, materials. John chooses his favourite sofa from the item list via the shop application on his smartphone or tablet. Option1: John is only equipped with a smartphone. The sofa is added his living room on his smartphone thanks to the onboard camera and depth sensor from the device. John can then try different locations in the living room, select the colour that better fits with his home place. Option 2: John is also equipped with a pair of AR glasses When connected to the online store via his smartphone, John also connects his AR glasses to his smartphone. The sofa is then rendered on his AR glasses and John continue to use his smartphone in order to control the location of the sofa within the living room.
Categorization
Type: AR **Degrees of Freedom: 6DoF** **Delivery: Download** **Device: AR Glasses, Rendering system, Tablet (or smartphone), Capture device**
PreCondition
Tablet (or smartphone) with the following features - 4G/5G connectivity - 3D capture capabilities with depth capturing - rendering of overlay 3D model in the captured scene/video Capture device (video and depth camera). AR glasses with connectivity to the tablet/smartphone. Application with 3D model representation of selling items.
QoS and QoE considerations

QoS:

- Accurate and low latency rendering
- Reliable and fast download of the 3D model to be rendered.

QoE:

- Fast and accurate rendering of 3D object of items (such as proper lightening and reflectance in AR scenes)
- Accurate placement of the 3D object in AR scene.
- Less heterogeneity through AR glasses

Feasibility

AR services of furniture planning are already available. For example,

- IKEA™: https://www.youtube.com/watch?v=vDNzTasuYEw
- Amazon™: https://www.amazon.com/adlp/arview

In such applications, the chosen item can be placed in the AR scene. Therefore, it would be possible that the item is represented through AR glasses if it has information of the 3D model. Rendering device is capable of rendering a 3D object in the captured scene or in the field of view of the user's AR glasses.

Potential Standardization Status and Needs

The following aspects may require standardization work:

- Standardized format for 3D object such as point clouds
- Delivery protocols for 3D object
- Decoding, rendering, composition API for 3D object in AR scene

A.12 Use Case 11: Real-time communication with the shop assistant

Use Case Description: Real-time communication with the shop assistant

In addition to the above use case for online shopping from a catalogue, the remote assistant is available for products on sale. John can seek advice from the online shop assistant on which colour the sofa better matches with the living room.

John chooses his favourite sofa from the item list via the shop application on his smartphone or tablet and can add 3D representation of the sofa into his living room scene captured by the camera. John can try different locations in the living room, select the colour that better fits with his home place.

John captures the AR scene with 3D representation of the sofa in his living room and transmits the captured scene of the living room is transmitted in real time to the online assistant who can make suggestions to John.

Use case extension:

The shop assistant is able to place virtual furniture, e.g., a lamp into John's captured scene in real time and transmit to suggest for John to also buy a lamp that nicely fits with the rest of the living room.

Categorization

Type: AR

Degrees of Freedom: 6DoF

Delivery: Interactive, Conversational

Device: AR Glasses, Rendering system, Table (or smart phone), audio headset

PreCondition

AR glasses equipped or connected with capture device (depth camera), positioning system and rendering system. Capture device supports to save the captured scenes in point cloud format.

Tablet (or smartphone) with 4G/5G connection

Headset (headphones with embedded microphones) is used for conversation.

Online shopping mall supports all of the items in point clouds.

QoS and QoE considerations

QoS:

- In case of sufficient bandwidth, the user and assistant should be able to transmit and receive the scene and the voice streams simultaneously (if necessary, simultaneous instant messaging service should be possible). For HD video quality, at least over 1 Mbit/s is needed.
- conversational QoS requirements

QoE:

- No disconnection or interruption in the middle of the conversation between the user and the assistant even in the environment where the captured scenes are sharing.
- high-quality AR scene with accurate placement and rendering of 3D object in real environment
- Synchronized AR scene between user and assistant

Feasibility

- real time AR communication or assistance, for example:
  - https://www.youtube.com/watch?v=GFhpAe10qnk9 (Live remote support with 3D annotation to the Microsoft™ HoloLens™)
  - In this application, field technicians can use the Microsoft™ HoloLens™ to connect to a remote expert with an unprecedented clarity of communication, as well as receive assistance and perform tasks with unmatched speed and accuracy
  - https://chalk.vuforia.com/ (Vuforia™ chalk)

It provides a remote guidance and collaboration app designed for technicians and experts to more effectively communicate to solve problems.

Potential Standardization Status and Needs

The following aspects may require standardization work:

- Coded representations of AR scene and delivery in MTSI context
- MTSI/FLUS uplink AR video
- Downlink AR video with local/cloud computation and rendering
- MTSI regular audio between John and assistant

A.13 Use Case 12: 360-degree conference meeting

Use Case Description: 360-degree conference meeting

In this 360-degree conferencing use case three co-workers (Eilean, Ben and John) are having a virtual stand-up giving a weekly update of their ongoing work. Ben is dialing into the VR conference from work with a VR headset and a powerful desktop PC. Eilean is working from home and dialing in with a VR headset attached to a VR capable laptop with a depth camera. John is traveling abroad and dialing in with a mobile phone used as VR HMD and a bluetooth connected depth camera for capture. Thus, each user is captured with an RGB+Depth camera.

Figure 1, example image of a photo-realistic 360-degree communication experience

In virtual reality all 3 of them are sitting together around a round table (See Figure 1). The background of the virtual environment is a prerecorded 360-degree image or video making it seem they are in their normal office environment. Each user sees the remote participants as photo realistic representations blended into the virtual office environment (in 2D). Optionally, a presentation or video can be displayed on the middle of the table or on a shared screen somewhere in the environment.

AR alteration: A possible AR alteration to this use case can be that Ben and Eilean are sitting in a real meeting room at work using AR headsets, while John is attending remotely using a mobile as VR HMD. John is then blended as an overlay into the real environment of Ben and Eilean, rather then a virtual office.

Categorization

Type: AR, MR, VR

Degrees of Freedom: 3DoF

Delivery: Conversational

Device: Mobile / Laptop

Preconditions

The above use case results into the following hardware requirements:

- Each user needs a AR or VR HMD (mobile, stand alone, wired/wireless VR HMD).
- Each user needs a depth camera to be captured (based on Bluetooth, integrated into a mobile phone or wired)
- Each user needs a microphone and audio headset for audio upload and spatial audio playback
- Each user needs to be connected and registered to the network to facitilate the end-to-end audio/video call.

Requirements and QoS/QoE Considerations

The following QoS requirements are considered:

- Bandwidth: As minimal bandwidth it is expected at least 3Mbit/s (this is for a single 2D user stream with chroma background), however this requirement can increase with more complex and higher resolution streams.
- Delay: The delay has to be suitable for real-time communication.

The main goal of this use case is to create shared presence and immersion. Thus foresee the following QoE Considerations as relevant:

- Capture & Processing:
  - The resolution of the rgb+depth camera needs to be sufficient.
  - The foreground / background extraction needs to result into an accurate cut-out of a user
- Transmission:
  - The compression of audio and video data should follow similar constraints as traditional video conferencing.
- Rendering:
  - Users, need to be scaled and positioned in the AR/VR environment in a natural way
  - Audio playback needs to match the spatial orientation of the user

Feasibility

Demos & Technology overview:

- M. J. Prins, S. N. B. Gunkel, H. M. Stokking, and O. A. Niamut. TogetherVR: A Framework for photorealistic shared media experiences in 360-degree VR. SMPTE Motion Imaging Journal 127.7:39-44, August 2018.
- S. N. B. Gunkel, H. M. Stokking, M. J. Prins, O. A. Niamut, E. Siahaan, and P. S. Cesar Garcia. Experiencing Virtual Reality Together: Social VR Use Case Study. In *Proceedings of the 2018 ACM International Conference on Interactive Experiences for TV and Online Video*. ACM, 2018
- S. N. B. Gunkel, M. J. Prins, H. M Stokking, and O. A. Niamut. Social VR platform: Building 360-degree shared VR spaces. In Adjunct Publication of the 2017 ACM International Conference on Interactive Experiences for TV and Online Video, ACM, 2017.

In summary:

- Users are captured with an RGB+depth device, e.g. Microsoft Kinect or Intel Realsense Camera
- This capture is processed locally for foreground/background segmentation WebRTC is used for transmission of streams to the other call participants.
- A-Frame / WebVR is used for rendering the virtual environment

Existing Service:

- http://www.mimesysvr.com/

Summary of steps:

Figure 2, Functional blocks of end-to-end communication

Furthermore, to realize this use case it can be mapped it into the following functional blocks:

- Capture & Processing: The Data from the rgb+depth camera needs to be acquired and further processed to remove the user from its background to be ready for transmission. It is foreseen that many end-user devices will not be capable of doing this themselves, and that processing will need to be offloaded to the network. (Optionally) there can be audio processing and enhancements like removal of background noise and reverberation of the capture environment.

- Transmission: There needs to be a two-way end to end link between individual participants to transmit audio and video data. The video data should include a cut-out of the user on a chroma background in order to place a user representation into the 360-degree image background. Instead of chroma background, alpha channel (for transparancy) is also an option.
- Rendering: Rendering on the end user device, preferably on a single decoding platform/chipset with efficient simultaneous decoding of different media streams. Further, the transferred user representation has to be blended into a VR or AR environment and any audio needs to be played according to its spatial origin within the environment.
- Cloud processing (optional): by adding a (pre-) rendering function into the cloud, processing and resource usage will shift from the end user device into the edge (or cloud) and thus imply a less scalability system but lower processing load for the end user device

Please note that this is a functional diagram and this is not mapped to physical entities yet.

Potential Standardization Status and Needs

The following aspects may require standardization work:

- System
  - Architecture
  - Communication interfaces / signalling
- Media Orchestration (i.e. metadata)
  - Position and scaling of people
  - Spatial Audio (e.g. including audio directionality of users)
  - Background audio / picture / video
  - Shared content (i.e. video background), i.e. multi-device media synchronization
  - Allow Network based processing (e.g. cloud rendering, foreground /background segmentation of user capture, replace HMD of user with a photo-realistic representation of there face, etc.)
- Transmission
  - The end-to-end system (including the network) needs to support the RGB+Depth video data.

A.14 Use Case 13: 3D shared experience

Use Case Description: 3D shared experience

In this shared 3D use case two friends (Eilean and Bob) are sharing a virtual experience. The experience builds around a crime investigation showing an investigation of two murder suspects and allowing the users to discuss and identify who committed the murder. Both Eileen and Bob are joining from home wearing a VR HMD and being captured via an RGB+depth camera. In VR they experience a 3-dimensional room (6DOF, police station), being represented in 3D and including a self-representation that allows them to point at items in the room and at each other. This representation can be based on the same capture that is made with the RGB+depth camera for communication purposes. Further, in the virtual police station each one of them has a window to follow a different interrogation (windowed 6DOF / 3DOF+), allowing them to collect information to solve the murder together (see figure 2).

Figure 2, example image of a virtual 3D experience with photo-realistic user representations

Categorization

Type: AR, MR, VR

Degrees of Freedom: 3DoF+ / 6DOF

Delivery: Conversational

Device: Mobile / Laptop

Preconditions

The above use case results into the following hardware requirements:

- Each user needs a VR HMD (mobile, stand alone, wired/wireless VR HMD).
- Each user needs a depth camera to be captured (based on Bluetooth, integrated into a mobile phone or wired)
- Each user needs a microphone and audio headset for audio upload and spatial audio playback
- Each user needs to be connected and registered to a network that is able to facitilate the end-to-end audio/video call.

Requirements and QoS/QoE Considerations

The following QoS requirements are considered:

- Bandwidth: As minimal bandwidth it is expected at least 6Mbit/s (this is for a single 2D+ user stream with RGB + depth video), however this requirement can increase with more complex and higher resolution streams.
- Delay: suitable for real-time communication
- Delay (self-view): suitable for feeling of embodiment

The main goal of this use case is to create a shared presence and immersion in a 3DOF+/6DOF experience. Thus the following QoE Considerations are relevant:

- Capture & Processing:
  - The resolution of the rgb+depth camera needs to be sufficient.
  - The foreground / background extraction needs to result into an accurate cut-out of a user
- Transmission:
  - The compression of audio and video data should follow similar constraints as traditional video conferencing.
- Rendering:
  - Users, needs to be scaled and positioned in the AR/VR environment in a natural way

- Audio playback needs to match the spatial orientation of the user
- A self view needs to be properly aligned with the actual body movement to align proprioceptive and visual experience. Also, delay for this needs to be kept to a minimum.

Feasibility

Demos & Technology overview:

- M. J. Prins, S. N. B. Gunkel, H. M. Stokking, and O. A. Niamut. TogetherVR: A Framework for photorealistic shared media experiences in 360-degree VR. SMPTE Motion Imaging Journal 127.7:39-44, August 2018.
- S. N. B. Gunkel, H. M. Stokking, M. J. Prins, N. van der Stap, F.B.T. Haar, and O.A. Niamut, 2018, June. Virtual Reality Conferencing: Multi-user immersive VR experiences on the web. *In Proceedings of the 9th ACM Multimedia Systems Conference* (pp. 498-501). ACM.
- 2018, IBC Demo: https://vrtogether.eu/2018/09/14/ibc-show-2018/

In summary:

- Users are captured with an RGB+depth device, e.g. Microsoft Kinect or Intel Realsense Camera
- This capture is processed locally for foreground/background segmentation and optionally for creation of a self-view.
- WebRTC is used for setting up streams to the other call participants.
- A-Frame / WebVR is used for rendering the virtual environment.

Existing Service:

- http://www.mimesysvr.com/

Summery of steps:

Figure 2, Functional blocks of end-to-end communication

Furthermore to realize this use case it is mapped into the following functional blocks:

- Capture & Processing: The Data from the rgb+depth camera needs to be acquired and further processed (to remove the user from its background), particularly the depth information might need further possessing before transmission
- Transmission: There needs to be a two-way end to end link between individual participants to transmit audio and video data. The video data should include a both the rgb colour and depth information.
- Rendering: The transferred user representation has to be blended into the VR environment (according to its geometrical properties based on the RGB + Depth data) and any audio needs to be played according to its special origin within the environment. Further the self-representation of the user has to be displayed aligned so that the view of the user and its physical position match.

Please not that all 3 functional blocks can be executed either on one device, multiple devices or the network.

Potential Standardization Status and Needs

The following aspects may require standardization work:

- System

- Architecture
- Communication interfaces (signalling)

- Media Orchestration (i.e. metadata)
  - Position and scaling of people
  - Spatial Audio (e.g. including audio directionality of users)
  - Background audio
  - Shared content, i.e. multi-device media synchronization
  - Allow Network based processing (e.g. cloud rendering, foreground /background removal of user capture, image enhancements like hole filling, replace HMD of user with a photo-realistic representation of there face, etc.)

- Transmission
  - The end-to-end system (including the network) needs to support the RGB+Depth video data.

A.15 Use Case 14: 6DOF VR conferencing

Use Case Name
6DOF VR conferencing
Description
The use case was initially described in TR 26.918 as Virtual Meeting Place: *The main idea here is to create a virtual world where people can meet and interact anonymously through their avatars with other people. A user would be able to move freely in the virtual world (6 DOF) and mingle with different groups of people depending for example on the discussion they are having. In this scenario, the user would be able to speak to other users in his/her immediate proximity and obtain a spatial rendering of what the other users in his/her immediate proximity are saying and would hear them from the same relative positions they have to him/her in the virtual world.* Below follows a more detailed description both of the physical scenario underlying the use case and the created virtual scenario. 1. Physical scenario The physical VR conference scenario is illustrated in Fig. 1. Five VR conference users from different sites are virtually meeting. Each of them is using VR gear with binaural playback and video playback using an HMD. The equipment of all users supports movements in 6DOF with corresponding headtracking. The UEs of the users exchange coded audio up- and downstream with a VR conference call server. Visually, the users are represented through their respective avatars that can be rendered based on information related to relative position parameters and their rotational orientation.

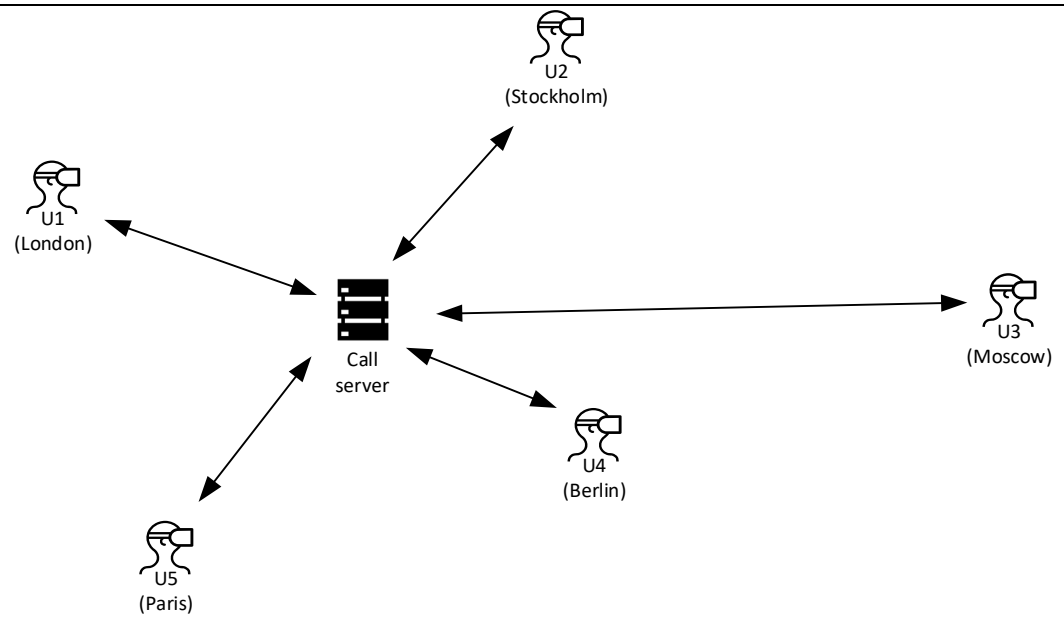


Figure 1: Physical scenario

2. Virtual scenario

Fig. 2 illustrates the virtual conferencing space generated by the conference call server. Initially, the server places the conference users Ui, i=1…5, at virtual position coordinates $K_i = (x_i, y_i, z_i)$. The virtual conferencing space is shared between the users. Accordingly, the audio-visual render for each user takes place in that space. For instance, from user U5's perspective, the rendering will virtually place with the other conference participants at the relative positions $K_i - K_5$, $i \neq 5$. For example, user U5 will perceive user U2 at distance $|K_i - K_5|$ and under the direction of the vector $(K_i - K_5)/|K_i - K_5|$, whereby the directional render is done relative to the rotational orientation of U5. Also illustrated in Fig. 2 is the movement of U5 towards U4. This movement will affect the position of U5 relative to the other users, which will be taken into account while rendering. At the same time the UE of U5 sends its changing position to the conferencing server, which updates the virtual conferencing space with the new coordinates of U5. As the virtual conferencing space is shared, users U1–U4 become aware of moving user U5 and can accordingly adapt their respective renders. The simultaneous movement of user U2 works according to corresponding principles.

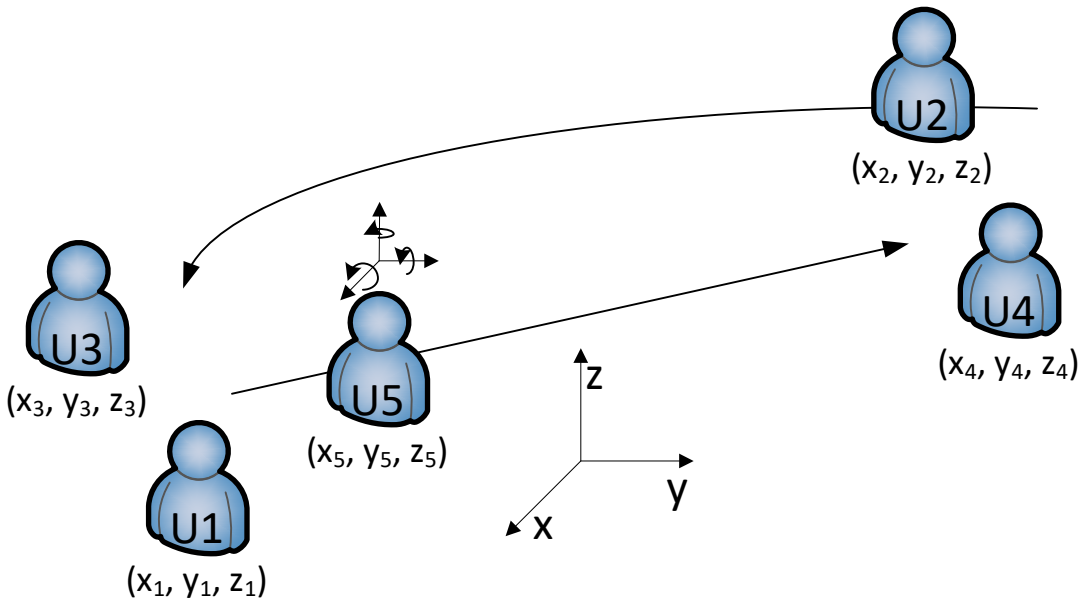


Figure 2: Virtual scenario

Categorization

Type: VR

Degrees of Freedom: 6DoF

Delivery: Interactive, Conversational

Media Components: Audio-only, Audio-Visual

Devices: VR gear with binaural playback and HMD video playback, Call server

Preconditions

The described scenario relies on a conference call server.

Similar scenarios can be realized without a server. In that case, the UEs of all users need to be configured to share their encoded audio and their 6DOF positional and rotational information with the UEs of all other users. Each UE will further allow simultaneous reception and decoding of audio bitstreams and 6DOF attributes from the UEs of all other users.

Specific minimum preconditions

- UE with render capability through connected HMD supporting binaural playback.
- Mono audio capture.
- 6DOF Position tracking.

Conference call server:

- Maintenance of participant position data in shared virtual meeting space.

Media preconditions:

Audio:

- The capability of simultaneous spatial render of multiple received audio streams according to their associated 6DOF attributes.
- Adequate adjustments of the rendered scene upon rotational and translational movements of the listener's head.

Video/Graphics:

- Support of simultaneous graphics render on HMDs of multiple avatars according to their associated 6DOF attributes, including position, orientation, directivity.

Media synchronization and presentation format control:

- Required for controlling the flow and proper render of the various used media types.

System preconditions:

- A metadata framework for the representation and transmission of positional information of an audio sending endpoint, including 6DOF attributes, including position, orientation, directivity.

Requirements and QoS/QoE Considerations

QoS: conversational requirements as for MTSI, using RTP for Audio and Video transport.

- Audio: relatively low bit rate requirements, that will meet conversational latency requirements.
- Video/Graphics: no particular QoS requirements since graphics synthesis can be done locally at each rendering UE based on the received 6DOF attributes of the audio elements corresponding to the participants.

QoE: Immersive voice/audio and visual graphics experience.

The described scenario provides the users with a basic 6DOF VR meeting experience. Quality of Experience of the audio aspect can be enhanced if the user's UEs not only share their position coordinates but also their rotational orientation. This will allow render of the other virtual users not only at their positions in the virtual conference space but additionally with proper orientation. This is of use if the audio and the avatars associated with the virtual users support directivity, such as specific audio characteristics related to face and back.

Feasibility

The following capabilities and technologies are required:

- UE with render capability through connected HMD supporting binaural playback.
- Mono audio capture.
- 6DOF position tracking.

It is concluded that a service offering an experience as the described scenario is feasible with today's technology. The identified preconditions as well as the provided considerations on QoS/QoE do not suggest a feasibility barrier, given the technologies widely available and affordable today.

Potential Standardization Status and Needs

- Requires standardization of at least a 6DOF metadata framework and a 6DOF capable renderer for immersive voice and audio.
- The presently ongoing IVAS codec work item will provide an immersive voice and audio codec/renderer and a metadata framework that may meet these requirements.
- Also required are suitable session protocols coordinating the distribution and proper rendering of the media flows.

A.16 Use Case 15: XR Meeting

Use Case Name

XR Meeting

Description

This use case is a mix of a physical and a virtual meeting. It is an XR extension of the virtual meeting place use case described in 3GPP TR 26.918. The use case is exemplified as follows:

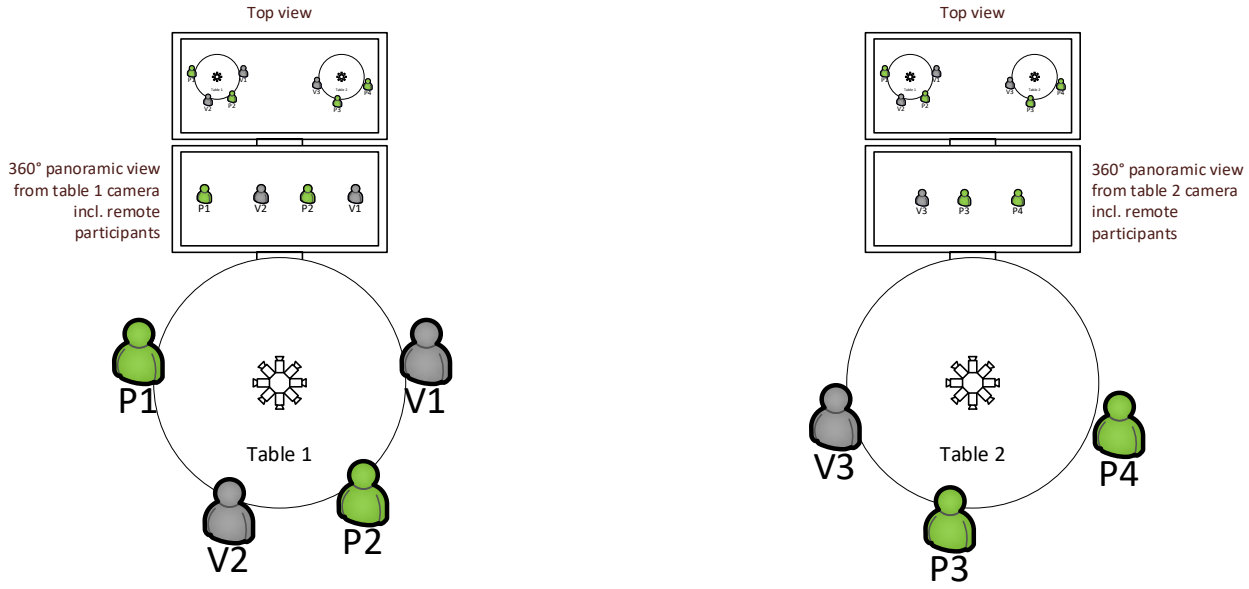
Company X organizes a workshop with discussions in a couple of smaller subgroups in a conference room, as for instance shown in the figure below. Each subgroup gathers around dedicated spots or tables and discusses a certain topic and participants are free to move to the subgroup of their interest. Remote participation is enabled.

The main idea for the remote participants is to create a virtual, 3D-rendered space where they can meet and interact through their avatars with other people. This 3D-rendered virtual space is a simplified representation of the real conference room, with tables at the same positions as in the real world. Remote participants are equipped with HMD supporting binaural playback. A remote participant can move freely in the virtual conference room and interact with the different subgroups of people depending, for example, on the discussion they are having. A remote participant can speak to other participants in their immediate proximity and obtain a spatial audio rendering of what the other participants are saying. They can hear the real participants from their relative positions in the virtual world, and they can freely walk from one subgroup to another to seamlessly join different conversations that may happen concurrently in the meeting space. Consistent with the auditory scene, the remote participant will be able to see on the HMD a rendered "Scene view" of the complete virtual meeting space from their viewpoint, i.e. relative to position and viewing direction. As options, the remote participant may also select to see a "Top view" of the

complete meeting space with all participants (or their avatars) or a "Table view". The latter is generated from a 360-degree video capture at the relevant table. The audio experience remains in any case as during "Scene view".

The physical participants see and hear avatars representing the remote participants through AR Glasses supporting binaural playback. They interact with the avatars in the discussions as if these were physically present participants. For physical participants, the interactions with other physical and virtual participants happen in an augmented reality. In addition, at each subgroup meeting spot, a video screen displays a 360-degree panoramic "Table view" taken from the middle of the respective table, including the overlaid avatars of the remote participants taking part in the subgroup discussion. Also displayed is the complete meeting space with all participants (or their avatars) in a top view.

A schematic of the configuration at the physical meeting space is shown in the following figure. In that figure, P1 through P4 represent the physical participants while V1 through V3 are the remote participants. Also shown are two subgroup meeting spots (tables), each with a 360-degree camera mounted on its center. Further, at each table the two video screens are shown for the 360-degree panoramic "Table view" and for the "Top view".



Categorization

Type: AR, VR, XR

Degrees of Freedom: 6DoF

Delivery: Interactive, Conversational

Device: Phone, HMD with binaural playback support, AR Glasses with binaural playback support

Preconditions

On a general level the assumption is that all physical attendees (inside the meeting facilities) wear a device capable of binaural playback and, preferably, AR glasses. Remote participants are equipped with HMDs supporting binaural playback. The meeting facility is a large conference room with a number of spatially separated spots (tables) for subgroup discussions. Each of these spots is equipped with at least one video screen. At each of the spots a 360-degree camera system is installed.

Specific minimum preconditions

Remote participants:

- UE with render capability through connected HMD supporting binaural playback.
- Mono audio capture.
- 6DOF Position tracking.

Physical participants:

- UE with render capability through a non-occluded binaural playback system and preferably, but not necessarily, AR Glasses.
- Mono audio capture of each individual participant e.g. using attached mic or detached mic with suitable directivity and/or acoustic scene capture at dedicated subgroup spots (tables).
- 6DOF Position tracking.

Meeting facilities:

- Acoustic scene capture at dedicated subgroup spots (tables) and/or mono audio capture of each individual participant.
- 360-degree video capture at dedicated subgroup spots (tables).
- Video screens (connected to driving UE/PC-client) at dedicated subgroup meeting spots visualizing participants including remote participants at a subgroup spot ("Table view") and/or positions of participants in shared meeting space in "Top view".

Conference call server:

- Maintenance of participant position data in shared virtual meeting space.
- (Optional) synthesis of graphics visualizing positions of participants in shared meeting space in "Top view".
- (Optional) generation of overlay/merge of synthesized avatars with 360-degree video to "Table view".

Media preconditions:

Audio:

- The capability of simultaneous spatial render of multiple received audio streams according to their associated 6DOF attributes.
- Adequate adjustments of the rendered scene upon rotational and translational movements of the listener's head.

Video/Graphics:

- 360-degree video capture at subgroup meeting spots.
- Support of simultaneous graphics render of multiple avatars according to their associated 6DOF attributes, including position, orientation, directivity:
  - Render on AR glasses.
  - Render on HMDs.
- Overlay/merge synthesized avatars with 360-degree video to "Table view":
  - Render as panoramic view on video screen.
  - VR Render on HMD excluding a segment containing the remote participant itself.
- Synthesis of "Top view" graphics visualizing positions of participants in shared meeting space.

Media synchronization and presentation format control:

- Required for controlling the flow and proper render of the various used media types.

System preconditions:

- A metadata framework for the representation and transmission of positional information of an audio sending endpoint, including 6DOF attributes, including position, orientation, directivity.
- Maintenance of a shared virtual meeting space that intersects consistently with the physical meeting space:

Real and virtual participant positions are merged into a combined shared virtual meeting space that is consistent with the positions of the real participant positions in the physical meeting space.

Requirements and QoS/QoE Considerations

QoS: conversational requirements as for MTSI, using RTP for Audio and Video transport.

- Audio: Relatively low bit rate requirements, that will meet conversational latency requirements.
- 360-degree video: Specified in TS 26.118, and will meet conversational latency requirements. It is assumed that remote participants will at each time receive only the 360-degree video stream of a single subgroup meeting spot (typically the closest).
- Graphics for representing participants in shared meeting space may rely on a vector-graphics media format, see e.g. 26.140. The associated bit rates are low. Graphics synthesis may also be done locally in render devices, based on positional information of participants in shared meeting space.

QoE: Immersive voice/audio and visual experience, Quality of the mixing of virtual objects into real scenes.

The described scenario provides the remote users with a 6DOF VR meeting experience and the auditory experience of being physically present in the physical meeting space. Quality of Experience for the audio aspect can further be enhanced if the user's UEs not only share their position but also their orientation. This will allow render of the other virtual users not only at their positions in the virtual conference space but additionally with proper rotational orientation. This is of use if the audio subsystem and the avatars associated with the virtual users support directivity, such as specific audio characteristics related to face and back.
The "Scene view" for the remote participants allows consistent rendering of the audio with the 3D-rendered graphics video of the meeting space. However, that view obviously compromises naturalness and "being-there" experience through the mere visual presentation of the participants through avatars. The optional "Table view" may improve the naturalness as it relies on a real 360-degree video capture. However, QoE of that view is compromised since the 360-degree camera position does not coincide with virtual position of remote user. Viewpoint correction techniques may be used to mitigate this problem.

The physical meeting users experience the remote participants audio-visually at virtual positions as if these were physically present and as if they could come closer or move around like physical users. The AR glasses display the avatars of the remote participants at positions and in orientation matching the auditory perception. Physical participants without AR glasses get a visual impression of where the remote participants are located in relation to the own position through the video screens at the subgroup meeting spots with the offered "Table view" and/or the "Top view".

Feasibility

Under "Preconditions" the minimum preconditions are detailed and broken down by all involved nodes of the service, such as remote participants, physical participants, meeting facilities and conference call server. In summary, the following capabilities and technologies are required:

- UE with render capability through connected HMD supporting binaural playback.
- UE with render capability through a non-occluded binaural playback system and preferably, but not necessarily, AR Glasses.
- Mono audio capture and/or acoustic scene capture.
- 6DOF position tracking.
- 360-degree video capture at dedicated subgroup spots.

- Video screens (connected to driving UE/PC-client) at dedicated subgroup meeting spots visualizing participants including remote participants at a subgroup spot ("Table view") and/or positions of participants in shared meeting space in "Top view".
- Maintenance of participant position data in shared virtual meeting space.
- (Optional) synthesis of graphics visualizing positions of participants in shared meeting space in "Top view".
- (Optional) generation of overlay/merge of synthesized avatars with 360-degree video to "Table view".

While the suggested AR glasses for the physical meeting participants are very desirable for high QoE, the use case is fully feasible without glasses. Immersion is in that case merely provided through the audio media component. Thus, none of the preconditions constitute a feasibility barrier, given the technologies widely available and affordable today.

Potential Standardization Status and Needs

- Requires standardization of at least a 6DOF metadata framework and a 6DOF capable renderer for immersive voice and audio.
- The presently ongoing IVAS codec work item may provide an immersive voice and audio codec/renderer and a metadata framework that may meet these requirements.
- Other media (non-audio) may rely on existing video/graphics coding standards available to 3GPP.
- Also required are suitable session protocols coordinating the distribution and proper rendering of the media flows.

A.17 Use Case 16: Convention / Poster Session

Use Case Name

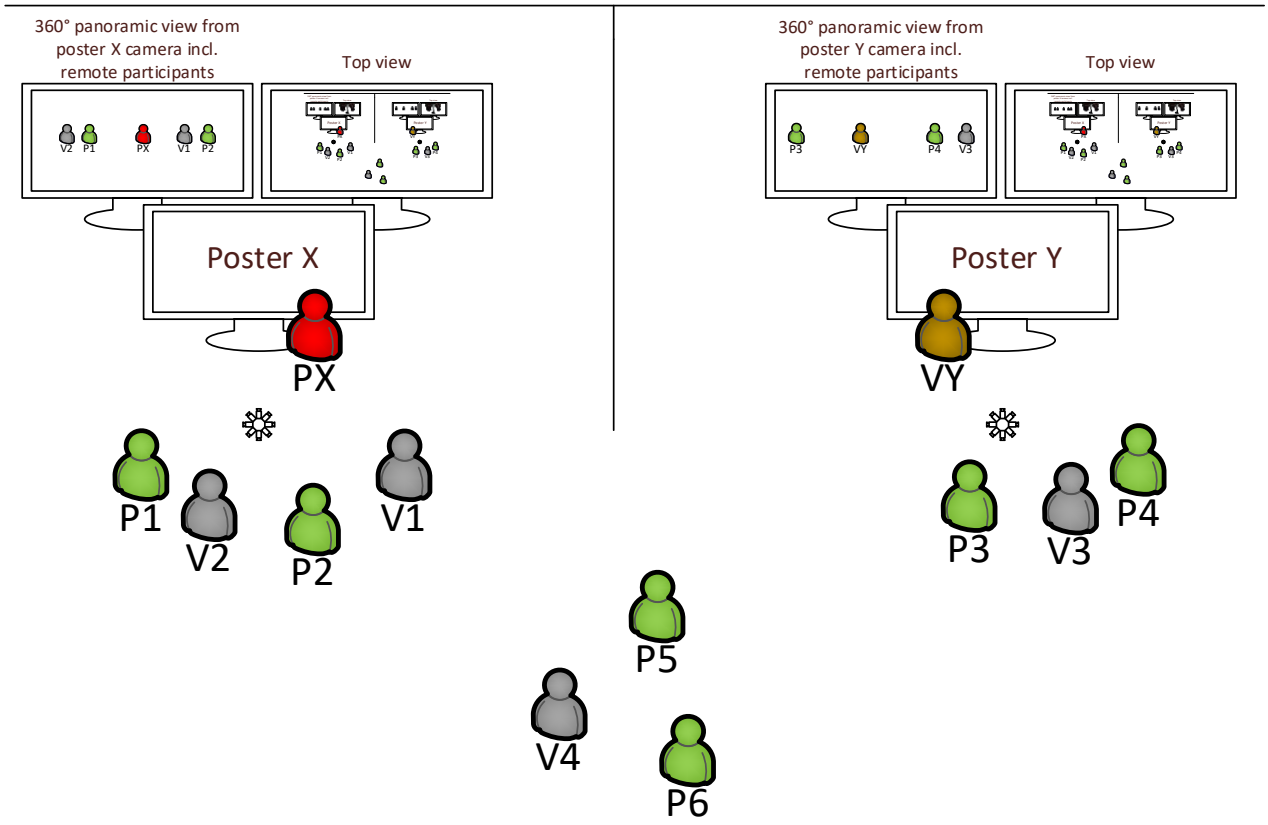
Convention / Poster Session

Description

This use case is exemplified with a conference with poster session that offers virtual participation from a remote location.

It is assumed that the poster session may be real, however, to contribute to meeting climate goals, the conference organizers are offering a green participation option. This is, a virtual attendance option is offered to participants and presenters, as an ecological alternative avoiding travelling.

The conference space is organized in a few poster booths, possibly separated by some shields. In some of the booths, posters are presented by real presenters, in some other booths, posters are presented by remote presenters. The audience of the poster presentations may be a mix of physically present and remote participants. Each booth is equipped with a first video screen for the poster display and one or two additional video screens for the display of a "Top view" and/or the display of a panoramic "Poster presentation view". Each booth is further equipped with a 360-degree camera system capturing the scene next to the poster. The conference space is visualized in the following figure, which essentially corresponds to the "Top view". In this figure, P1-P6 represent physical attendees, V1-V4 are remote attendees, PX and VY are real and, respectively, remote presenters. There are two poster presentations of posters X and Y. Participants V4, P5 and P6 are standing together for a chat.



Physical attendees and presenters have the experience of an almost conventional poster conference, with the difference that they see remote persons through their AR glasses, represented as avatars. They hear the remote persons through their binaural playback systems. They can also interact in discussions with remote persons like they were physically present. Physical presenters use a digital pointing device to highlight the parts of their poster that they want to explain. The physical audience attends the poster presentation of a remote presenter in some dedicated physical spots of the conference area that is very similar to the poster booth of a physical presenter. The participants see and hear the virtual presenter through their AR glasses supporting binaural playback. They also see and hear the other audience that may be physically present or just be represented though avatars.

Remote participants are equipped with HMD supporting binaural playback. They are virtually present and can walk from poster to poster. They can listen to ongoing poster presentations and move closer to a presentation if they think the topic or the ongoing discussion is interesting. A remote participant can speak to other participant in his/her immediate proximity and obtain a spatial rendering of what the other participants in his/her immediate proximity are saying. He/she can hear them from the relative positions they have to him/her in the virtual world. Consistent with the auditory scene, the remote participant will be able to see on the HMD a synthesized "Scene view" of the complete conference space (including the posters) from his/her viewpoint, i.e. relative to position and viewing direction. The remote participant may also select to see a "Top view", which is an overview of the complete conference space with all participants (or their avatars) and posters or to see a "Poster presentation view". The latter is a VR view generated from the 360-degree video capture at the relevant poster but excluding a segment containing the remote participant itself. The audio experience remains in any case as during "Scene view". In order to give the remote participants the possibility to interact in the poster discussions, they also have the possibility to use their VR controller as a pointing device to highlight certain parts of the poster, for instance when they have a specific question.

Remote presenters are equipped with HMD supporting binaural playback and a VR controller. Most relevant for them is the "Scene view" in which they see (in their proximity) their audience represented by avatars. This view is overlaid with their own poster. They use their VR controller as a pointing device to highlight a part of the poster that they want to explain to the audience. It may happen that a remote presenter sees some colleague passing by and, to attract her/him to the poster, they may take some steps towards that colleague and call out to her/him.

The remote participants are represented at the real event through their avatars, which the real participants and presenters see and hear through their AR glasses supporting binaural playback. The real and virtual participants and the presenter interact in discussions as if everybody was physically present.

Categorization

Type: AR, VR, XR

Degrees of Freedom: 6DoF

Delivery: Interactive, Conversational

Device: Phone, HMD with binaural playback support, AR Glasses with binaural playback support, VR controller/pointing device

Preconditions

On a general level the assumption is all physical attendees (inside the conference facilities) wear a device capable of binaural playback. Remote participants are equipped with HMD supporting binaural playback. The meeting facility is a large conference room with a number of spatially separated booths for the different poster presentations. Each of these spots is equipped with a video screen for the poster and at least one other video screen. At each of the poster spots a 360-degree camera system is installed.

Specific minimum preconditions

Remote participant:

- UE with connected VR controller.
- UE with render capability through connected HMD supporting binaural playback.
- Mono audio capture.
- 6DOF Position tracking.

Remote presenter:

- UE with connected VR controller.
- UE with render capability through connected HMD supporting binaural playback.
- UE has document sharing enabled for sharing of the poster.
- Mono audio capture.
- 6DOF Position tracking.

Physical attendees/presenters:

- UE with render capability through a non-occluded binaural playback system and AR Glasses.
- Mono audio capture of each individual participant e.g. using attached mic or detached mic with suitable directivity and/or acoustic scene capture at dedicated subgroup spots (poster booths).
- 6DOF Position tracking.
- UE has a connected pointing device.
- UE of presenter has document sharing enabled for display of the poster on video screen and for sharing it with remote participants.

Conference facilities:

- Acoustic scene capture at dedicated subgroup spots (poster booths) and/or mono audio capture of each individual participant.
- 360-degree video capture at dedicated spots, at the posters.
- Video screens at dedicated spots (next to the posters), for poster display and for visualizing participants including remote participants at a poster ("Poster presentation view") and/or positions of participants in shared meeting space in "Top view".

- Video screens are connected to driving UE/PC-client.

Conference call server:

- Maintenance of participant position data in shared meeting space
- Synthesis of graphics visualizing positions of participants in conference space in "Top view".
- Generation of overlay/merge of synthesized avatars with 360-degree video to "Poster presentation view".

Media preconditions:

Audio:

- The capability of simultaneous spatial render of multiple received audio streams according to their associated 6DOF attributes.
- Adequate adjustments of the rendered scene upon rotational and translational movements of the listener's head.

Video/Graphics:

- 360-degree video capture at subgroup meeting spots.
- Support of simultaneous graphics render of multiple avatars according to their associated 6DOF attributes, including position, orientation, directivity:
  - Render on AR glasses.
  - Render on HMDs.
- Overlay/merge synthesized avatars with 360-degree video to "Table view":
  - Render as panoramic view on video screen.
  - VR Render on HMD excluding a segment containing the remote participant itself.
- Synthesis of "Top view" graphics visualizing positions of participants in shared meeting space.

Document sharing:

- Support of sharing of the poster from UE/PC-client as bitmap/vector graphics or as non-conversational (screenshare) video.

Support of sharing of pointing device data and VR controller data, potentially as real-time text.

Media synchronization and presentation format control:

- Required for controlling the flow and proper render of the various used media types.

System preconditions:

- A metadata framework for the representation and transmission of positional information of an audio sending endpoint, including 6DOF attributes, including position, orientation, directivity.
- Maintenance of a shared virtual meeting space that intersects consistently with the physical meeting space:
  - Real and virtual participant positions are merged into a combined shared virtual meeting space that is consistent with the positions of the real participant positions in the physical meeting.

Requirements and QoS/QoE Considerations

QoS: conversational requirements as for MTSI, using RTP for Audio and Video transport.

- Audio: Relatively low bit rate requirements, that will meet conversational latency requirements.

- 360-degree video: Specified in TS 26.118, and will meet conversational latency requirements. It is assumed that remote participants will at each time receive only the 360-degree video stream of a single poster spot (typically the closest).
- Graphics for representing participants in shared meeting space may rely on a vector-graphics media format, see e.g. TS26.140. The associated bit rates are low. Graphics synthesis may also be done locally in render devices, based on positional information of participants in shared meeting space.
- Document sharing: Relatively low bit rate. No real-time requirements.
- Pointing device/VR controller data: Very low bit rate. Real-time requirements.
- Media synchronization and presentation format: Low bit rate. Real-time requirements.

QoE: Immersive voice/audio and visual experience, Quality of the mixing of virtual objects into real scenes.

The described scenario provides the remote users in "Scene view" with a 6DOF VR conferencing experience and the feeling of being physically present at the conference. The remote participants and the real poster session / conference audience are able to hear the remote attendee's verbalized questions and the presenter's answers in a way that their audio impression matches their visual experience and which provides a high degree of realism. Quality of Experience can further be enhanced if the user's UEs not only share their position but also their orientation. This will allow render of the other virtual users not only at their positions in the virtual conference space but additionally with proper rotational orientation. This is of use if the audio and the avatars associated with the virtual users support directivity, such as specific audio characteristics related to face and back. The experience is further augmented through the virtual sharing of the posters and the enabled interactions using the pointing devices.

However, the "Scene view" compromises naturalness and "being-there" experience through the mere visual presentation of the participants through avatars. The optional "Poster presentation view" may improve the naturalness as it relies on a real 360-degree video capture. However, QoE of that view is compromised since the 360-degree camera position does not coincide with virtual position of remote user. Viewpoint correction techniques may be used to mitigate this problem.

The physical meeting users experience the remote participants audio-visually at virtual positions as if these were physically present and as if they could come closer or move around like physical users. The AR glasses display the avatars of the remote participants at positions and in orientation matching the auditory perception. Physical participants without AR glasses receive a visual impression of where the remote participants are located in relation to the own position through the video screens at the poster booths.

Feasibility

Under "Preconditions" the minimum preconditions are detailed and broken down by all involved nodes of the service, such as remote participants, physical participants, meeting facilities and conference call server. In summary, the following capabilities and technologies are required:

- UE with connected VR controller/pointing device.
- UE with render capability through connected HMD supporting binaural playback.
- UE with render capability through a non-occluded binaural playback system and AR Glasses.
- Mono audio capture and/or acoustic scene capture.
- 6DOF Position tracking.
- UE supporting document sharing (for sharing of the poster).
- 360-degree video capture at dedicated subgroup spots, at the posters.
- Video screens (connected to driving UE/PC-client) at dedicated spots (next to the posters), for poster display and for visualizing participants including remote participants at a poster ("Poster presentation view") and/or positions of participants in shared meeting space in "Top view".
- Maintenance of participant position data in shared virtual meeting space.
- Synthesis of graphics visualizing positions of participants in conference space in "Top view".

- Generation of overlay/merge of synthesized avatars with 360-degree video to "Poster presentation view".
- Poster sharing and sharing of pointing device data.

While the suggested AR glasses for the physical meeting participants are very desirable for high QoE, the use case is fully feasible even without glasses. Immersion is in that case merely provided through the audio media component. Thus, none of the preconditions constitute a feasibility barrier, given the technologies widely available and affordable today.

Potential Standardization Status and Needs

- - Requires standardization of at least a 6DOF metadata framework and a 6DOF capable renderer for immersive voice and audio.
- - The presently ongoing IVAS codec work item may provide an immersive voice and audio codec/renderer and a metadata framework that may meet these requirements.
- - Other media (non-audio) may rely on existing image/video/graphics coding standards available to 3GPP.
- - Also required are suitable session protocols coordinating the distribution and proper rendering of the media flows.

A.18 Use Case 17: AR animated avatar calls

Use Case Name

AR animated avatar call

Description

This use case is about a call scenario between one user wearing AR glasses and the other user using a phone in handset mode. The AR glasses user sees an animated avatar of the phone user. Movements of the phone user are used to control the animation of his avatar. This improves the call experience of the user of the AR glasses.

A potential user experience is described as a user story:

Tina is wearing AR glasses while walking around in the city. She receives an incoming call by Alice, who is using her phone, and who is displayed as an overlay ("head-up display") on Tina's AR glasses. Alice doesn't have a camera facing at her, therefore a recorded 3D image of her is sent to Tina as the call is initiated. The 3D image Alice sent can be animated, following Alice's actions. As Alice holds her phone in handset mode, her head movements result in corresponding animations of her transmitted 3D image, giving Tina the impression that Alice is attentive.

NOTE: An option for this use case is a "mute animations" control. Note that Alice didn't press the "mute animations" button that would have disabled all animations of her 3D image for Tina.

As Tina's AR glasses also include a pair of headphones, Alice' mono audio is rendered binaurally at the position where she is displayed on Tina's AR glasses. Tina also has interactivity settings, allowing to lock Alice's position on her AR screen. Therefore, her visual and auditory appearance moves when Tina rotates her head. As Tina disables the position lock, the visual and auditory appearance of Alice is placed within Tina's real world and thus Tina's head rotation leads to compensation on the screen and audio appearance, requiring visual and binaural audio rendering with scene displacement.

Type: AR

Degrees of Freedom: 2D, 3DoF

Delivery: Conversational

Device: Phone, HMD, Glasses, headphones

Preconditions
AR participants: Phone with tethered AR glasses and headphones (with acoustic transparency). Phone participant: Phone with motion sensor and potentially proximity sensor to detect handset mode.
Requirements and QoS/QoE Considerations
QoS: QoS requirements like MTSI requirements (conversational, RTP), e.g. 5QI 1. QoE: Immersive voice/audio and visual experience, Quality of the mixing of virtual objects (avatars) into real scenes and rendering an audio overlaid to the real acoustic environment.
Feasibility
AR glasses in various form factors exist, including motion sensing and inside-out tracking. This allows locking of avatars and audio objects to the real world. Smart phones typically come with built-in motion sensing, using a combination of gyroscopes, magnetometers and accelerometer. This allows extraction of the head's rotation, when the phone is used in handset mode, which could be motion data sent to the other endpoint to animate/rotate the avatar/3D image.
Potential Standardization Status and Needs
Visual coding and transmission of avatars or cut-out heads, alpha channel coding Transmission and potentially coding of motion data to show attentiveness

A.19 Use Case 18: AR avatar multi-party calls

Use Case Name
AR avatar multi-party call
Description
This use case is about multi-party communication with spatial audio rendering, where avatars and audio of each participant are transmitted and spatially rendered in the direction of their geolocation. Each participant is equipped with AR glasses with external or built-in head phones. 3D audio can be captured and transmitted instead of mono, which leads to enhancements when sharing the audio experience. A potential user experience is described as a user story: Bob, Jeff, and Frank are in Venice and walking around the old city sightseeing. They are all wearing AR glasses with a mobile connection via their smartphone. The AR glasses support audio spatialization, e.g. via binaural rendering and playback over the built-in headphones, allowing the real world to be augmented with visuals and audio. They start a multi-party call, where each of them gets the other two friends displayed on his AR glasses and can hear the audio. While they walk around in the silent streets, they have a continuous voice call with the avatars displayed on their AR glasses, while also other information is displayed to direct them to the secret places of Venice. Each of them transmits his current location to his friends. Their AR glasses / headphones place the others visually and acoustically (i.e. binaurally rendered) in the direction where the others are. Thus, they all at least know the direction of the others. As Jeff wants to buy some ice cream, he switches to push-to-talk to not annoy his friends with all the interactions he has with the ice cream shop. As Bob gets closer to Piazza San Marco the environment gets noisier with sitting and flying pigeons surrounding him. Bob turns on the "hear what I hear" feature to give them an impression on the fascinating environment, sending 3D audio of the scene to Frank and Jeff. As they got interested, they also want to experience the pigeons around them and walk through the city to the square. Each of the friends is still placed on the AR glasses visually and acoustically in the direction where the friend is, which makes it easy for them to find Piazza San Marco and for Frank to just walk

across the square to Bob as he approaches him. Jeff, who still eats his ice cream is now also coming closer to Piazza San Marco and just walks directly to Bob and Jeff. As they get close to each other they are no longer rendered (avatars and audio), based on the positional information, and they simply chat with each other.
Type: AR **Degrees of Freedom: 3DoF** **Delivery: Conversational** **Device: AR glasses, headphones**
Preconditions
Connected AR glasses or phone with tethered AR glasses and headphones (with acoustic transparency). Positioning support (e.g. using GNSS) to derive geolocation, allowing calculation of relative position. 3D audio capturing (e.g. using microphone arrays) and rendering.
Requirements and QoS/QoE Considerations
QoS: QoS requirements like MTSI requirements for voice/audio and avatars (conversational, RTP), e.g. 5QI 1 for audio. QoE: Immersive voice/audio and visual experience, Quality of the capturing and rendering of avatars, the different participants and 3D audio.
Feasibility
AR glasses in various form factors exist. Those usually include motion sensors, e.g. based on accelerometers, gyroscopes, and magnetometers, but also cameras are common, allowing inside-out tracking and augmentation of the real world. 3D audio capturing and rendering are available, e.g. using spherical or arbitrary microphone arrays for capturing and using binaural rendering technologies for audio spatialization. An audio-only solution using headphones and head-tracking is easier to implement, this would however remove the possibility to visually augment the real world and display avatars.
Potential Standardization Status and Needs
Visual coding and transmission of avatars Audio coding and transmission of mono objects and 3D audio for streams from all participants NOTE: scene composition is usually a differentiating factor

A.20 Use Case 19: Front-facing camera video multi-party calls

Use Case Name
Front-facing camera video multi-party call
Description
This use case is based on front-facing camera calls, i.e. a user is having a video call, seeing the other participants on the display of e.g. a smartphone he holds at arm's length. The use case has some overlap with UC 6 (AR face-to-face calls) and UC 10 (Real-time 3D Communication), extended by spatial audio rendering for headphones/headsets. The

spatial audio rendering is based on the head-tracker data extracted from the smartphones front-facing camera, giving a user the impression, even with movements, that the voice of the other participants originates from a virtual stage in the direction of the phone with the video of the other's faces.

A potential user experience is described as a user story:

Bob, Jeff, and Frank are back in New York City and each of them is walking to work. They just have their smart phones with a front-facing camera and a small headset, allowing the real world to be augmented with audio.

They start a multi-party video call to discuss the plans for the evening, where each of them gets the other two friends displayed on the phone and can hear the audio, coming from the direction on the horizontal plane where the phone is placed in their hand and some small spread to allow easy distinction. While they walk around in the streets of New York, they have a continuous voice call with their phones at arm's length, with the, potentially cut-out, faces of their pals displayed on their phones. For Bob the acoustic front is always in the direction of his phone, thus the remote participants are always in the front. When Bob rotates his head though, the front-facing camera tracks this rotation and the spatial audio is binauralized using the head-tracking information, leaving the position of the other participants steady relative to the phone's position. As Bob turns around a corner with the phone still at arm's length for the video call using the front-facing camera, his friends remain steady relative to the phone's position.

Type: AR

Degrees of Freedom: 3DoF

Delivery: Conversational

Device: Smartphone with front-facing camera, headset , AR glasses

Preconditions

Phone with front-facing camera, motion sensors, and headset (more or less acoustically transparent). Motion sensors to compensate movement of the phone, front-facing camera to capture the video for the call and potentially track the head's rotation.

Requirements and QoS/QoE Considerations

QoS: QoS requirements like MTSI requirements (conversational, RTP), e.g. 5GQI 1 and 2.

QoE: Immersive voice/audio and visual experience, Quality of the capturing, coding and rendering of the participant video (potentially cut out faces), Quality of the capturing, coding and rendering of the participant audio, including binaural rendering taking head tracking data into account.

Feasibility

Several multi-party video call applications using the front-facing camera exist, e.g. https://www.cnet.com/how-to/how-to-use-group-facetime-iphone-ipad-ios-12/ , https://faq.whatsapp.com/en/android/26000026/?category=5245237

Head tracking using cameras exists, e.g. https://xlabsgaze.com

Binaural rendering with head-tracking also exists (see also TS26.118)

Potential Standardization Status and Needs

Visual coding and transmission of video recorded by front-facing camera; potentially cut-out heads, alpha channel coding

Audio coding and transmission for streams from all participants

NOTE: scene composition is usually a differentiating factor

A.21 Use Case 20: AR Streaming with Localization Registry

Use Case Description: AR Streaming with Localization Registry

A group of friends has arrived at a museum. The museum provides them with an AR guide for the exhibits. The museum's exhibition space has been earlier scanned and registered via one of the museum's AR devices to a Spatial Computing Service. The service allows storing, recalling and updating of spatial configuration of the exhibition space by a registered AR device. Visitors' AR devices (to be used by museum guests as AR guides) have downloaded the spatial configuration upon entering the museum and are ready to use.

The group proceeds to the exhibit together with their AR guides, which receive a VoD stream of the museum guide with the identifier Group A. Registered surfaces next to exhibits are used for displaying the video content (may be 2D or 3D content) of the guide. In the case of spatial audio content, this may be presented in relation to the registered surfaces. The VoD stream playback is synched amongst the users of Group A. Any user within the group can pause, rewind or fast forward the content, and this affects the playback for all the members of the group. Since all users view the content together, this allows them to experience the exhibit as a group, and discuss during pauses without affecting the content streams for other museum visitors that they are physically sharing the space with. Other groups in the museum use the same spatial configuration, but their VoD content is synched within their own group.

The use case can be extended to private spaces, e.g., a group of friends gathered at their friend Alice's house to watch a movie. Alice's living room is registered already under her home profile on the Spatial Computing Service; the saved information includes her preferred selection of the living room wall as the movie screening surface. She shares this configuration via the service with her guests.

In this use case, the interaction with a travel guide avatar may also occur in a conversational fashion.

Categorization

Type: AR and Social AR

Degrees of Freedom: 6DoF

Delivery: Streaming, Interactive, Conversational

Device: AR glasses with binaural audio playback support

Preconditions

The use case requires technical solutions for the following functions:

Spatial Computing Service

A 5G service that registers users and stores their indoor spatial configuration with the following features:

- Reception of a stream of visual features for a space to be registered. The input may be from a mobile phone camera, an AR device or a combination of data from multiple sensors and cameras located in the space.
- Usage of a localization algorithm such as SLAM (Simultaneous Localization and Mapping) for indoor spatial localization, and the storage of special configurations, such as the selection of particular surfaces for special functions (e.g., wall for displaying a video stream).
- Distribution of previously stored information to other devices belonging to the same user or to other authorized users.
- Updating of localization information and redistribution when required.

Content synchronization

A streaming server that distributes content and ensures synchronized content playback for multiple AR users. The server does not need to have the content stored locally. It can, for example, get the content stream from a streaming

service and then redistribute it. For the museum guests, the functionality may be part of the XR client or embedded in a home gateway or console-like device.

QoS/QoE Considerations

- Required QoS:
  - Sufficiently low latency for synchronized streaming playback and conversational QoS.
- Required QoE:
  - Synchronization of VoD content for multiple users within acceptable parameters. This requires ensuring the streams' playback occurs near simultaneously for all users, so that user reactions to specific scenes such as jump scares in a horror movie or a goal in a sport sequence are also synced within the group. Furthermore, playback reaction time to user actions such as pause, fast forward and rewind should be low and similar for all users within the group. Conversational low-delay QoE is also expected.

Feasibility

The use case is feasible within a timeframe of 3 years. Required hardware, AR glasses, are available in the market, and network requirements are no more or less than existing streaming services.

The feasibility of the use case depends on the accuracy of the localization registration and mapping algorithm. Multiparty AR experiences, such as a shared AR map annotation demo from Mapbox (https://blog.mapbox.com/multi-user-ar-experience-1a586f40b2ce?gi=60ceb3226701) and the Multiuser AR experience exhibition at the San Fransisco Museum of Modern Art by Ubiquity6 (https://www.youtube.com/watch?v=T-I3YG_w-Z4), provide good examples for proof of concept of already available technology for creating a shared AR experience.

Potential Standardization Status and Needs

The following aspects may require standardization work:

- Standardized way of sharing and storing indoor spatial information with the service and other devices.
- Mixing VoD streams may require some additional functions for social AR media control playback. This would relate to allowing users to control the playout of the VoD stream (pause, rewind, fast-forward) for all users in a synchronized manner.

A.22 Use Case 21: Immersive 6DoF Streaming with Social Interaction

Use Case Description: Immersive 6DoF Streaming with Social Interaction

In an extension to the use case 3 in clause 6.4 for which Alice is consuming the game in live mode, Alice is now integrated into social interaction:

- She virtually watching the game with other friends who are geographically distributed and whose avatars are sitting in the stadium next to her. She has voice conversations with those friends while watching the game.
- While she moves through the stadium to another location, she make friends with other folks watching the same game in the virtual environment.
- She gets overlaid contextually relevant twitter feeds

Categorization

Type: VR and Social VR

Degrees of Freedom: 3DoF+, 6DoF

Delivery: Streaming, Split, Conversational, Interactive

Device: HMD with a controller

Preconditions

- Application is installed that permits to consume the scene
- The application uses existing HW capabilities on the device, including A/V decoders, rendering functionalities as well as sensors. Inside-out Tracking is available.
- Media is captured properly and accessible on cloud storage through HTTP access
- One or multiple communication channels across users can be setup

Requirements and QoS/QoE Considerations

- Same as use case in clause 6.3. In addition, the following applies
  - Required QoS:
    - Sufficient low latency for the communication channel
  - Required QoE:
    - Sufficiently low communication latency
    - Synchronization of user communication with action
    - Synchronized and context-aware twitter feeds

Feasibility

See use case 3 in clause A.4.

The addition of social aspects can be addressed by apps.

Some discussion on this matter:

- https://www.roadtovr.com/nextvr-latest-tech-is-bringing-new-levels-of-fidelity-to-vr-video/, see the second page. However, still no publicly announced details.

Social VR is used in different context. See for example here:

- https://www.juegostudio.com/infographic/various-social-vr-platforms
- https://www.g2crowd.com/categories/vr-social-platforms

Some example applications are provided

- Facebook Spaces™
  - https://www.facebook.com/spaces
- VRChat
  - https://www.vrchat.net/
  - https://en.wikipedia.org/wiki/VRChat
  - https://youtu.be/5cpElonP33k
- Oculus Venues ™
  - https://www.engadget.com/2018/05/30/oculus-venues-hands-on
  - https://www.esquireme.com/oculus-headset-will-let-you-watch-live-sport-in-virtual-reality

Optimizations can be done by integrating social A/V with main content (rendering, blending, overlay).

Additional pointers to deployments and use cases:

- https://www.nextvr.com/nextvr-gets-social-with-oculus-venues-now-fans-can-enjoy-live-vr-experiences-together/
- https://www.oculus.com/blog/go-behind-the-scenes-of-the-oc5-oculus-venues-livestream-with-supersphere/?locale=en_US
- Verizon presentation at XR Workshop
  - Virtual Live Events w/Friends
    - Virtually attend live events with friends in 4K/8K 360°3D video (aka 'VR')
    - Technical Requirements
      - 4K, 8K+ (6DoF) real time (volumetric)streaming, Immersive 360°Video (stereoscopic, 90+ FPS) ➔ MEC for video stitching is optional on 4K
      - Directional audio, user point of view ➔ For real time chat, selectable viewpoints
      - Integrated Videos and Communications ➔ RCS-based communication, supports delivery to all deployed smartphones as well as VR devices

Potential Challenges:

- Quality of avatars
- Synchronization of scene
- Quality of interactions

Potential Standardization Status and Needs

The following aspects may require standardization work:

- same as use case 6.3
- Social VR component and conversation
- Synchronized Playout of users in the same room

A.23 Use Case 22: 5G Online Gaming party

Use Case Description: 5G Online Gaming party

In an extension to use case 5 in Annex A.6 on Online Immersive Gaming experience, the users join a Gaming Party either physically or virtually in order to experience maximum and controlled user experience. There are two setups for the party:

- The friends connect to a common server through 5G that provides managed resources and access guarantees to meet their high-demand requirements for gaming.
- The friends meet physically and connect to an infrastructure using wireless 5G connection. The setup explores all options, including connecting to a centralized infrastructure, but also possibly connecting HMDs using device to device communication.

The experience is improved and especially very consistent compared to best effort connections they had been used to before.

In a similar use case as presented during the 2nd XR Workshop, it is referred to as "City-wide multiplayer, immersive AR gaming action/adventure experience"

- User enters an outdoor geo-fenced space including parks & other activation sites for an AR gaming play experience.
- Once inside the geolocation, user opens app on 5G phone & goes through local matchmaking to join with other nearby players for co-operative experience.
- Players use AR wayfinding to head to the first dead drop.
- User scans environment using AR Lens to uncover first clue and work alongside other players to solve AR puzzle to unlock the next level.
- The winners from the battle unlock AR Wayfinding for next location and next battle.
- At the final location, the remaining users confront final opponent and play AR combat mini game to defeat him and unlock exclusive content.

Categorization

Type: VR, AR

Degrees of Freedom: 6DoF

Delivery: Streaming, Interactive, Split, device-to-device

Device: HMD with a Gaming controller, AR glasses

Preconditions

- Gaming client is installed that permits to consume the game
- The application uses existing HW capabilities on the device, including game engines, rendering functionalities as well as sensors. Inside-out Tracking is available.
- Connectivity to the network is provided.
- Connectivity can be managed properly
- Devices may connect using device-to-device communication
- Wayfinding and SLAM is provided to locate and map to the venue in case of AR
- AR and AI functionalities are provided for example for Image & Object Recognition, XR Lighting, Occlusion Avoidance, Shared Persistence

Requirements and QoS/QoE Considerations

The requirements are similar to what is discussed in use case 6.25.

Feasibility

Feasibility follows the previous discussions. However, a 5G Core Architecture that would provide such functionalities, would be needed. In addition, authentication for such "5G parties" is needed.

Potential Standardization Status and Needs

The following aspects may require standardization work:

- Network conditions that fulfill the QoS and QoE Requirements
- Content Delivery Protocols

- Decoding, rendering and sensor APIs
- Architectures for computing support in the network
- TR 22.842 [6] provides a gap analysis in clause 5.3.6 that is in line with these needs
- Authentication to such groups
- Possible support for device-to-device communication

A.24 Use Case 23: 5G Shared Spatial Data

Use Case Description: Shared Spatial Data

Consider as an example people moving through Heathrow airport. The environment is supported by spatial map sharing, spatial anchors, and downloading/streaming location based digital content. The airport is a huge dynamic environment with thousands of people congregating. Spatial maps and content will change frequently. Whereas base maps have been produced by professional scanners, they are continuously updated and improved by crowd sourced data. Semi-dynamic landmarks such a growing tree, a new park bench, or holiday decorations are incorporated into the base map via crowd sourced data. Based on this individuals have their own maps and portions of those maps may be shared with friends nearby. One could imagine spatial content will consume as much bandwidth as permitted, be it a high resolution volumetric marketing gimmick with virtually landing Concorde in Heathrow or a simple overlay outside a lounge showing the current wait time for getting access.

As people walk through 1km+ size spaces like the airport, they'll be progressively downloading updates and discarding map information that is no longer relevant. Similar to data flows in Google maps, smartphones continually send location and 3D positioning data (GPS, WiFi, scans, etc…) to the cloud in order to improve and augment 3D information. AR maps and content will in all likelihood be similarly layered, dynamic, and progressively downloaded. Spatial AR maps will be a mixture of underlying living spatial maps and digital content items.

The use case addresses several scenarios:

- Co-located people wearing an XR HMD collaboratively interact with a detailed 3D virtual model from their own perspective into a shared coordinate system (using a shared map).
- One person wearing an XR HMD places virtual objects at locations in 3D space for later discovery by other's wearing an XR HMD. This requires a shared map and shared digital assets.
- XR clients continuously send sensing data to a cloud service. The service constructs a detailed and timely map from client contributions and provides the map back to clients.
- An XR HMD receives a detailed reconstruction of a space, potentially captured by a device(s) with superior sensing and processing capabilities.

Categorization

Type: AR

Degrees of Freedom: 6DoF

Delivery: Streaming, Interactive, Split, device-to-device, different types

Device: HMD, AR Glasses

Preconditions

- Application is installed on an HMD or phone with connected AR glass
- The application uses existing HW capabilities on the device, rendering functionalities as well as sensors. Inside-out Tracking is available. Also a global positioning system for anchoring is available

- Connectivity to the network is provided.
- Wayfinding and SLAM is provided to locate and map in case of AR
- AR and AI functionalities are provided for example for Image & Object Recognition, XR Lighting, Occlusion Avoidance, Shared Persistence

Requirements and QoS/QoE Considerations

5G's low-latency high-bandwidth capabilities, as compared to 4G's capabilities, make 5G better suited for sending dense spatial data and associated 3D digital assets over a mobile network to XR clients.

This data could be transferred as discrete data downloads or streamed and may be lossy or lossless.

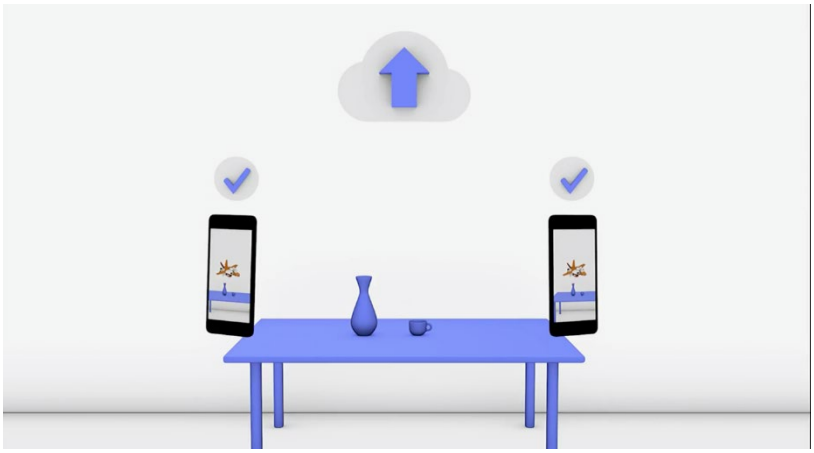
Continuous connectivity is important, sharing local information to improve maps.

The underlying AR maps should be accurate and should be up to date.

The content objects should be realistic.

The data representation for the AR maps and the content objects is scalable.

Feasibility

- Microsoft Spatial Anchors: https://azure.microsoft.com/en-us/services/spatial-anchors/
- Co-located people wearing an XR HMD collaboratively interact with a detailed 3D virtual model from their own perspective into a shared coordinate system (using a shared map).
- Google: Shared AR Experiences with Cloud Anchors: https://developers.google.com/ar/develop/java/cloud-anchors/overview-android
  - One person wearing an XR HMD places virtual objects at locations in 3D space for later discovery by other's wearing an XR HMD. This requires a shared map and shared digital assets
  - 
- Google Visual Positioning Service: https://www.roadtovr.com/googles-visual-positioning-service-announced-tango-ar-platform/
  - XR clients continuously send sensing data to a cloud service. The service constructs a detailed and timely map from client contributions and provides the map back to clients. Example is Google's Visual Positioning Service
- Drivenet Maps – Open Data real-time road Maps for Autonomous Driving from 3D LIDAR point clouds: https://sdi4apps.eu/2016/03/drivenet-maps-open-data-real-time-road-maps-for-autonomous-driving-from-3d-lidar-point-clouds/
  - An XR HMD receives a detailed reconstruction of a space, potentially captured by a device(s) with superior sensing and processing capabilities. An example of navigation is given in the MPEG-I use case document for point cloud compression (w16331, section 2.6)

Potential Standardization Status and Needs
The following aspects may require standardization work: - Data representations for AR maps - Collected sensor data to be streamed up streams - Scalable streaming and storage formats for AR maps - Content delivery protocols to access AR maps and content items Network conditions that fulfill the QoS and QoE Requirements

Annex B:
Change history

Change history							
Date	**Meeting**	**TDoc**	**CR**	**Rev**	**Cat**	**Subject/Comment**	**New version**
2019-09	SA#85	SP-190644				Presented to TSG SA#85 for information	1.0.0
2019-11	SA4#106	S4-191289				Agreements during SA4#106	1.1.0
2020-01	SA4#107	S4-200213				Agreements during SA4#107	1.2.0
2020-02	SA4#87	SP-200342				Agreements during SA4#107	1.3.0
2020-03	SA#87	SP-200046				Presented to TSG SA#87-e for approval	2.0.0
2020-03	SA#87	SP-200046				Approved by TSG SA#87-e	16.0.0
2020-12	SA#90-e	SP-200936	0001		F	Use Case Update for AR/MR Device Type	16.1.0
2022-04	-	-	-	-	-	Update to Rel-17 version (MCC)	**17.0.0**
2023-03	SA#99	SP-230260	0002	5	F	CR to TR 26.928 Add Clarification of the difference between Immersion and Presence	**18.0.0**

History

Document history		
V18.0.0	May 2024	Publication