3GPP TSG-SA WG4 Meeting #133-eS4-251287r01

Online, 18th July 2025 - 25th Jul 2025

**Source: Nokia**

**Title: [FS\_ARSpatial] pCR on mapping of spatial computing to 5G services**

**Spec: 3GPP TR 26.819**

**Agenda item: 9.7**

**Document for: Agreement**

**1. Introduction**

This contribution proposes various changes related to the mapping of spatial computing to 5G services.

**2. Reason for Change**

Description of some functional blocks of the baseline UE architecture in clause 6.2 is missing/incomplete. Further, it would be more accurate to use the term “XR baseline terminal” instead of “XR baseline client”, as the provided architecture also includes other entities (e.g. XR Runtime, Scene Manager) in addition to the Media Client (Media Session Handler, Media Access Function).

Spatial Computing Client (SCC) is a specialized type of Media Client that establishes communication for spatial computing purposes. Therefore, its current representation in clause 6.3 as a sub-block within the Media Access Function is not accurate. As a Media Client, the Spatial Computing Client should encompass both a Media Access Function and a Media Session Handler.

The call flow for spatial computing operations (clause 6.4) and its description contain unclear and inaccurate points; revisions are proposed to address these issues. Further, a Spatial Computing Client may, in addition to sensor data, also transmit control data while invoking spatial computing functions, e.g. to request a spatial region of interest from the Media AS. This needs to be reflected in the call flow.

**3. Proposal**

It is proposed to agree the following changes to 3GPP TR 26.819 v1.0.0.

\* \* \* First Change \* \* \* \*

# 6 Mapping of spatial computing to 5G services

## 6.1 Introduction

Using the Reference Architecture for Media Delivery defined in clause 4.1.2.2 of TS 26.506 [65] as a reference architecture, it is also possible to directly map specific spatial computing functions into the generalized functions in order to support spatial computing services. Spatial computing functions are located in the Media AS and in the XR Runtime. These functions generate and process XR Spatial Description data.

## 6.2 Terminal Architecture

The terminal architectural breakdown is based on the XR baseline terminal architecture in clause 5.1 of TS 26.119 [25]. The figure depicting the XR baseline terminal architecture is replicated here as Figure 6.2-1 for convenience.

NOTE: The term “XR baseline client” from TS 26.119 [25] is changed here to “XR baseline terminal”, as the provided architecture also includes other entities (e.g. XR Runtime, Scene Manager) in addition to the Media Client (Media Session Handler and Media Access Function).

A diagram of a software application

AI-generated content may be incorrect.

Figure 6.2-1 - XR Baseline terminal architecture.

The XR baseline terminal consists of the following components:

* The Media Access Function is responsible for the delivery of any metadata or local media to the spatial computing functions in the AS.
* The Scene Manager is responsible for the parsing of the description provided by the AS as output from the spatial computing functions (XR Spatial Description). It is also responsible for setting up and managing the XR session with the XR Runtime.
* The Presentation Engine is responsible for rendering the scene components based on the input from the Scene Manager.
* The XR Source Management is responsible for gathering timed metadata such as sensor data and pre-processed sensor information and making it available to the XR Application or the Media Access Function for sending in uplink.
* XR Runtime is a set of functions provided by the XR Device to the XR Application to support XR experiences. It may include some spatial computing functions that can be performed locally on the device.
* XR Application is a software application that runs on an XR device and offers an XR experience based on an XR Runtime. XR Application may invoke the spatial computing functions included in the XR Runtime.

\* \* \* Second Change \* \* \* \*

## 6.3 General Architecture for Spatial Computing

A diagram of a computer system

AI-generated content may be incorrect.

Figure 6.3-1 – Spatial computing architecture.

The architecture includes the following network functions and UE entities:

* Media Application Provider: Offers the spatial computing service and provisions support for spatial computing sessions offered by it.
* Media Application Function (Media AF): This function is responsible for QoS allocation in the 5G Core and providing service configuration information to the Media AS and Media Session Handler.
* Media Application Server (Media AS): This function is responsible for establishing the spatial computing session with the Media Client and monitoring the server’s edge resource usage. It hosts a Spatial Computing Function which can manage and run the spatial computing functions.
* Media-aware Application: The application running on the UE that makes use of 3GPP-defined APIs to invoke the Media Session Handler and/or the Media Access Function.
* Media Session Handler (MSH): This entity on the UE is responsible for the control plane signalling with the Media AF to establish a spatial computing session.

Spatial Computing Client (SCC): This function is responsible for discovering the UE spatial computing capabilities and negotiating with the Media AS to agree on the spatial computing session. NOTE: The SCC is a specialized Media Client and includes a Media Access Function and a Media Session Handler.

* Media Access Function: This function is as defined in TS 26.506 with the capabilities to send sensor data and receive XR Spatial Descriptions from the Media AS.
* Scene Manager: a set of functions that supports the application in arranging the logical and spatial representation of a multisensorial scene based on support from the XR Runtime. The Scene Manager composes the scene using the XR Spatial Description data.
* XR Source Management: management of data sources provided through the XR Runtime. It retrieves the sensor data from the XR Runtime and provides them to the Media Access Function.
* XR Runtime: Set of functions provided by the XR Device to the Media Application to create XR experiences. It may include some spatial computing functions.

The relevant interfaces shown in Figure 6.3-1 are:

* M1: The Media Application Provider provisions the spatial computing service through M1.
* M4: The signaling as well as the data delivery between Media Access Function and Media AS is through M4.
* M5: The MSH and the Media AF may exchange spatial computing configuration related information through the M5 interface.
* M6: Reference point between the Media-aware Application and the MSH for the purpose of configuring the Media Session Handler.
* M7: The Spatial Computing Client discovers the UE spatial computing capabilities through the M7 interface.
* M11: This interface may be used to convey QoS allocation and QoE information related to spatial computing functions between the Media Session Handler and the Media Access Function.

## 6.4 Call flow for spatial computing session setup and operation

The spatial computing operation can be described by the call flow in Figure 6.4.1.

Msc-generator~|version=8.6.2~|lang=signalling~|size=1422x717~|text=hscale=1.2;~n~nUE {~n~4RT: XR\nRuntime;~n~4PE: Presentation\nEngine;~n~4SM: Scene\nManager;~n~4XRSM: XR Source\n Management;~n~4App: Media-aware\nApplication;~n~4SCC: Spatial Computing Client {~n~8MSH: Media Session\nHandler;~n~8MAcF: Media Access\nFunction;~n~4};~n};~nMAF: Media\nAF;~nMAS: Media\nAS;~nbox SM--MAS: 0: Scene Description acquisition;~nbox UE++MAS: Session setup and negotiation~n{~n~4App-~gMAcF: 1. Initiate session;~n~4MAcF-~gMSH: 2. Inform MSH;~n~4MSH-~gMAF: 3. Invoke Dynamic Policy and\n AS discovery (for edge computing);~n~4MSH-~gMAcF: 4. Configure Media\n Access Function;~n~4MAcF~l-~gMAS: 5. Establish session;~n~4App-~gRT: 6. Configure XR Runtime;~n};~nRT-~gXRSM-~gMAcF: 7. Retrieve sensor and control data;~nMAcF-~gMAS: 8. Transmit sensor and control data;~nbox MAS--MAS: 9. Compute XR Spatial\nDescription;~nMAS-~gMAcF: 10. Send XR Spatial\nDescription;~nbox MAcF--MAcF: 11. Complete XR Spatial\nDescription;~nMAcF-~gSM: 12. Send XR Spatial Description;~nbox SM--SM: 13. Compose\nscene;~nSM-~gPE: 14. Send composed\nscene;~nbox PE--PE: 15. Render\ncomposed scene;~n~n~n/* CDNA: CDN A;~nCDNB: CDN B;~nO: Origin {~n~4CG: CMMF Bitstream\nGenerator/Source;~n~4S: Server;~n};~nAP: Application Service\nProvider;~n~n...: Service Configuration and Provisioning;~nAP~l-~gCDNA: Configure CDN A Service;~nAP~l-~gCDNB: Configure CDN B Service;~nAP-~gCG: CMMF Bitstream Generator/Source\nConfiguration Information;~nAP-~gS: Publish MPEG-DASH content\nto Origin Server;~n~n...: Client Initialization;~nbox App--AP: Service and Content Discovery~n{~n~4App-~gAP: Request application information;~n~4AP-~gApp: Provide application information\n\-(includes URLs to content manifests and cmmf_config.json);~n};~n~nApp-~gCR: Configure CMMF Reciever; ~n~nAP~l-CR: GET https://example.com/cmmf_config.json;~nAP-~gCR: cmmf_config.json\n\-(list of base URLs/domain names for CDN A and CDN B);~n~n...: Media Player Initialization;~n~nApp..App: Select\nContent;~nApp-~gMP: Configure Media Player;~n~nAP~l-MP: GET https://example.com/manifest.mpd;~nAP-~gMP: manifest.mpd;~n~n...: Content Delivery;~nMP-~gMP: Select\n100kb/seg01.mp4 [weak];~n~nMP=~gCR: GET 100kb/seg01.mp4;~nhspace MP-CR: GET 100kb/seg01.mp4;~nCR=~gCDNA: GET https://cdn-a.example.com/100kb/seg01.mp4;~nhspace CR-CDNA: GET https://cdn-a.example.com/100kb/seg01.mp4;~nCR=~gCDNB: GET https://cdn-b.example.com/100kb/seg01.mp4;~nhspace CR-CDNA: GET https://cdn-b.example.com/100kb/seg01.mp4;~n~nbox CDNA--CDNA: Check Cache;~nbox CDNA--S: Cache Miss [tag=~qAlt\#1~q]~n{~n~4CDNA=~gCG: GET 100kb/seg01.mp4;~n~4CG=~gS: GET 100kb/seg01.mp4;~n~4S=~gCG: 100kb/seg01.mp4;~n~4box CG--CG: Encode\nCMMF\nBitstream;~n~4CG=~gCDNA: 100kb/seg01.mp4 (CMMF A 1);~n}~n..: Cache Hit [tag=~qAlt\#2~q]~n{~n~4CDNA-~gCDNA: Read\nCache [weak];~n};~nCDNA=~gCR: 100kb/seg01.mp4 (CMMF A 1);~n~nbox CDNB--CDNB: Check Cache;~nbox CDNB--S: Cache Miss [tag=~qAlt\#1~q]~n{~n~4CDNB=~gCG: GET 100kb/seg01.mp4;~n~4CG=~gS: GET 100kb/seg01.mp4;~n~4S=~gCG: 100kb/seg01.mp4;~n~4box CG--CG: Encode\nCMMF\nBitstream;~n~4CG=~gCDNB: 100kb/seg01.mp4 (CMMF B 1);~n}~n..: Cache Hit [tag=~qAlt\#2~q]~n{~n~4CDNB-~gCDNB: Read\nCache [weak];~n};~nCDNB=~gCR: 100kb/seg01.mp4 (CMMF B 1);~n~nbox CR--CR: CMMF\nDecode;~nCR=~gMP: 100/seg01.mp4;~n~nMP-~gMP: Select\n200kb/seg02.mp4 [weak];~nMP=~gCR: GET 200/seg02.mp4;~nCR=~gCDNA: GET https://cdn-a.example.com/200kb/seg02.mp4;~nhspace CR-CDNA: GET https://cdn-a.example.com/200kb/seg02.mp4;~nCR=~gCDNB: GET https://cdn-b.example.com/200kb/seg02.mp4;~nhspace CR-CDNA: GET https://cdn-b.example.com/200kb/seg02.mp4;~n...: Continue; */~n~|

Figure 6.4.1 - High-level call flow for spatial computing session setup and operation.

The steps are:

0. The Scene Manager acquires the scene description and discovers the needed spatial computing functions for the XR experience.

1. The Media-aware Application initiates a session by configuring the Media Access Function at reference point M7.

2. The Media Access Function informs the Media Session Handler about the session and retrieves the relevant policy and configuration information at reference point M11.

3. The Media Session Handler communicates with the Media AF for the AS discovery (in the case of edge computing) and the configuration of quality-of-service parameters at reference point M5.

4. The Media Session Handler passes the information configured in step 3 to the Media Access Function via the interfaces M6/M11.

5. The Media Access Function in the Spatial Computing Client establishes the session with the Media AS at reference point M4.

6. The Media-aware Application configures the XR Runtime. This configuration enables the XR Runtime to provide sensor and control data needed for the spatial computing functions.

NOTE: Sensor data may include e.g. pose of the AR device. Control data may include e.g. coordinates of a bounding volume to request from the Media AS a spatial region of interest, i.e. a subset of 3D world model.

7. The Source Manager retrieves sensor and control data from the XR Runtime and provides them to the Media Access Function.

8. The Media Access Function sends sensor and control data to the Media AS.

9. The Media AS uses that data to invoke the spatial computing functions and compute the XR Spatial Description.

10. The Media AS transmits the generated XR Spatial Description to the Media Access Function in the Spatial Computing Client.

11. The Spatial Computing Client may complete the XR Spatial Description using local sensor data obtained by the XR Runtime.

12. The Spatial Computing Client provides the XR Spatial Description to the Scene Manager.

13. The Scene Manager composes the scene using the XR Spatial Description.

14. The Scene Manager provides the composed scene to the Presentation Engine.

15. The Presentation Engine renders the composed scene based on the input from the Scene Manager.

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