**Agenda item:** 10.7

**Source:** Tencent Cloud

**Title: [GA4RTAR] Improvements on EAS discovery in edge-enabled 5G RTC Architecture**

**Document for:** Discussion andAgreement

# Introduction

This contribution improves the recent updates on the edge computing support for 5G-RTC. Mainly, it uses the mechanism proposed in SR-MSE to provide network driven edge server instantiations ot the client to use for split-rendering.

# 5G-RTC Architecture extension

|  |
| --- |
| **Change 1** |

4.4.2 Extended 5G-RTC architecture for Edge Computing

The 5G-RTC architecture can be extended to add support for media processing in the edge. The extended architecture is an integration of the 5G-RTC architecture defined in TS 26.506 with the architecture for enabling Edge Applications defined in TS 23.558 and TS 26.501.

The extended 5G-RTC architecture supports both client-driven as well as Application Function-driven management of the edge processing session.

The 5G-RTC Application Provider may request the deployment of edge resources as part of the Provisioning Session.

* The 5G-RTC Application Provider provisions the edge provisioning through RTC-1, a similar fashion as defined in TS 26.512 subclause 7.10, enabling client-driven and/or Application Function driven edge configuration.
* In the client-driven approach, the WebRTC Application becomes aware of the support of edge processing in the network and takes steps, such as using the EDGE-5 APIs, to discover and locate a suitable 5G-RTC AS instance in the Edge DN, similar to the process defined in TS 26.501 subclause 8.1
* In the Application Function driven approach, the 5G-RTC Application Provider requests 5G-RTC AF to deploy edge processing for the media sessions of the corresponding Provisioning Session, similar to the process defined in TS 26.501 subclause 8.2. The WebRTC Application may get aware of the deployed EAS through the Application Service Provider through RTC-8 or through the Media Session Handler through RTC-5 (and possibly RTC-6). The EAS and split-rendering information is provided together such that the associated can be made by UE between the two set of data.. Additionally, the EAS may also be discovered through other means, such as DNS resolution with support from the DNS server (e.g., EASDF/DNS resolver) as specified in 3GPP TS 23.548 [1].

When the WebRTC application is a web application, the implementation of the EDGE-5 interface to discover the 5G-RTC AS/EAS location by accessing the EEC is difficult as the Web browser providers may not implement interfaces necessary for supporting edge enabled 5G-RTC applications/services. Also, in the Application Function-driven approach the Application Client (AC) and EEC are not used to discover the 5G-RTC AS/EAS location.

To resolve the above EAS discovery issue in the Application Function-driven approach and when the WebRTC application is a web application, the EAS information can be shared with the Media Session Handler by the 5G-RTC AF using RTC-5 interface.

NOTE: Other methods that can be used for sharing EAS information (e.g., sharing EAS hostname to the WebRTC application by RTC-8 or by other means and then using DNS resolution) are FFS.

The extended architecture is as shown in Figure 1.



Figure 1. Edge-enabled 5G-RTC architecture

NOTE: This architecture diagram is an example for CS-2 scenario.

|  |
| --- |
| **Change 2** |

4.4.3 Extended 5G-RTC architecture for Split Rendering

In the case of edge resource availability, the WebRTC application may request the edge server to run split-rendering process. In this case, the WebRTC can use TS 26.565 to establish a split-rendering process:

1. Optionally the Application Service Provider uses the procedure defined in 4.4.2 to instantiate or allocate EAS for the WebRTC application.

2. The Application Service Provider provisions the split-rendering process for the WebRTC application.

3. The WebRTC Application requests a split-rendering session set up through RTC-4 as defined in TS 26.565.NOTE: When the WebRTC application is a web-based application, it does not have access to MSH. The interaction between web-based application and MSH is outside the scope of this document.