**Agenda item:** 9.7

**Source:** Qualcomm Incorporated

**Title: Multimedia Application Context Definition**

**Document for** Discussion andAgreement

# Introduction

In this contribution, we provide a discussion of the definition of multimedia application context.

# Background

An EAS is typically an instance that runs in the edge and processes client requests. These requests may result in short-lived or long-lived interactions with or without possible state storage. Recent developments in cloud computing have offered a set of options for the deployment of the server:

* Bare metal servers: running the EAS on a bare metal server allows for full access to the server hardware resources but it suffers significantly from issues like scalability, cost efficiency, and fault tolerance.
* Virtual machines: virtual machines operate on a hypervisor, which runs on top of a bare metal host server. A hypervisor allows the creation and maintenance of VMs. A VM runs its own operating system in a completely isolated manner. It allows the virtualization of the underlying host hardware, making it available to applications and services running on that VM. VMs are typically resource hungry, due to the overhead required for running the guest operating system. They also suffer from long instantiation and launch delays.
* Containers: unlike VMs, contains provide operating system level virtualization. Containers still run in their private space, having separate processing, file systems, and networking. However, containers share the host’s operating system, thus reusing as much as possible of the components of the host operating system. This makes them more lightweight than VMs, which materializes in low overhead and fast launch times.
* Serverless: serverless computing is also known as Function as a Service (FaaS) architecture. Upon reception of an incoming request, the request gets forwarded to a suitable server (based on load balancing) and a function is invoked on that server to process the request. Provisioning and managing the servers is fully taken care of by the cloud service provider. Serverless allows for extremely fast launch times and low footprint. It is also very scalable.

Multimedia applications may differ significantly in their needs. Some applications may require full access to the underlying hardware, such as the GPUs, which limits the deployment options to bare metal or VMs. Others are transient and only require lightweight processing so that serverless computing is more appropriate.

In order to support application context relocation for multimedia applications, it is important to classify these applications based on criteria that affect the deployment and relocation procedures.

The following criteria are proposed to classify the multimedia applications:

* Stateful or stateless: a multimedia application may require the server side to maintain a state for the application. The state is usually maintained in memory for quick access for applications with continuous requests. The EAS should be able to serialize an application state into a representation that can be shared/transferred to another EAS.
* Persistent or transient connectivity: a multimedia application may require a persistent connection with the EAS to function properly. Examples are applications that rely on real-time communication or streaming. This may also cover backend connectivity of the EAS to the cloud.
* Downtime tolerance: some multimedia applications may be very sensitive to downtime. Other applications may tolerate downtime and are rather fault tolerant. This depends mostly on the previous two factors.

The relocation may be performed with or without application involvement. In the latter case, the container state is dumped, and a new container is instantiated starting from the dump state. Applications that are aware of the relocation procedure may offer better relocation/migration experience by avoiding a cold migration with long downtimes. The application servers will connect and transfer state to the target EAS before terminating the source EAS.

To facilitate this procedure, the application running on the server should have access to APIs to transfer its state and handover the session to the application running on the target EAS. The EAS will invoke the collected application context from the running applications and then will invoke Eees\_AppContextRelocation\_Request to trigger the transfer.

The application context and the ways to collect it should be configurable as part of the provisioning step. The options are:

* no application context as the application is stateless. If the application relies on transient connectivity, the relocation may be transparent to the application.
* a set of application context descriptors, where each describes an encrypted binary blob that contains the application context. That data may only be interpreted by the application on T-EAS. It should also contain an identifier for the target application running on the T-EAS.
* Notification hooks to inform the Application Provider about an imminent relocation operation.

# Proposal

We propose to define the provisioning extensions that describe the desired level of relocation support and to specify the APIs for the applications running on the source EAS to provide the application state.