**Agenda item:** 9.6

**Source:** Qualcomm Inc.

**Title: [FS\_AI4Media] Mapping to IMS using DC Applications**

**Document for** Discussion andAgreement

# Introduction

In this contribution, we describe a potential mapping of AI4Media to IMS, which leverages all existing IMS architecture and procedures.

# Background

The IMS data channel feature was introduced in Rel-17 to support the enhancement of multimedia telephony with more advanced application logic through web applications. A bootstrap data channel is established as part of the multimedia telephony session, then a list of available web applications is downloaded and offered to the user. Once a user has made a selection, the selected web application is downloaded locally. The remote UE is informed, through a re-INVITE, about the selected web application, so that both endpoints are using the same web application. An application data channel is then established between the endpoints to exchange application specific data. The UE uses the HTTP protocol for the communication over the bootstrap channel.

The architecture extensions to IMS to support data channels is shown in the following figure:



The DC Application Server acts as the endpoint for application data channels. It communicates with the DCSF for resource control and traffic forwarding and also supports the interaction with multiple UEs for simultaneous data channel applications.

The DCSF manages the signaling control for data channels. It implements data channel control and manages resources for both bootstrap and application data channels. It also manages the download and configuration of data channel applications from the DCAR.

The Media Function (MF) manages media resources and forwards data channel traffic. It terminates the bootstrap data channel from the UE and forwards HTTP traffic to the DCSF. The MF provides the media resources to anchor application data channels and relays the traffic between the UEs. The MF may terminate the application data channel by acting as an HTTP proxy or it may simply relay traffic by acting as UDP proxy.

The Data Channel Application Repository (DCAR) stores and manages verified data channel applications. These can then be downloaded by the UE through the DCSF and MF.

The IMS AS is enhanced to support data channel functionalities and manage interactions between the different entities.

# Mapping AI/ML Media Processing to IMS

Several AI/ML use cases were discussed and documented in TR26.927. Among these use cases are the NLP use cases, which describe a wide range of media processing tasks that can be applied to a multimedia call. To support these use cases, it should be possible to integrate the media processing with the media of the MTSI call. This is best done by leveraging the already existing data channel infrastructure, where AI/ML media processing is treated as a special category of web applications.

The supported media processing may be discovered as part of the data channel application discovery process over the bootstrap channel. The AI/ML media processing itself may be run in the UE, MF, or on both (split processing). The following flow chart depicts the process to trigger AI/ML media processing:



The steps are as follows:

1. The UE registers with the SIP registrar and indicates its ability to support AI-based media processing, e.g. through the inclusion of a dedicated parameter of the Contact header field,
2. The UE invites UE2 for a call by sending a SIP INVITE. Alternatively, the UE receives an invite to join a call.
3. The IMS-AS sets up the data channel resources for the connection,
4. The session is established by forwarding the INVITE to the remote UE and receiving the acknowledgement,
5. Setting up the AI/ML inference task using the data channel procedures:
   1. The UE establishes a bootstrap data channel to the DCSF through the MF
   2. The UE downloads a list of the DC applications enhanced with the list of available inference tasks,
   3. The telephony application displays a selection screen with the list of available inference tasks to the user for selection,
   4. Once the user has selected a inference task, the UE downloads a web application associated with the selected AI/ML inference task over the bootstrap channel,
   5. The telephony application renders the web application to the user and collects user’s input,
   6. The UE then establishes an application data channel to the MF/MRF and and shares any information acquired from the user’s input to the MF/MRF,
   7. The MF/MRF may use the exchanged application information to configure the selected AI/ML inference task accordingly,

[Editor’s note: Needs to study how the MF processes the application information for configuration ]

1. The UE sends the audio and video streams of the call through the MF/MRF for processing,
2. The MF/MRF runs the selected inference task with the call media streams as input,
3. The MF/MRF forwards the processed media streams to UE2,
4. The user may update or change the AI/ML inference task. The updates are sent to the MF/MRF via the application data channel,
5. The MF/MRF updates the AI/ML inference task accordingly and continues the inference.

Note that the AI/ML inference task may create additional media streams that are then included in the updated SDP offer automatically by the IMS AS.

As described by the flow chart diagram, launching and applying AI/ML inference tasks on the media streams of a call may be realized through the usage of data channels. Bootstrap data channel is used to discover available AI/ML inference tasks. The application data channel may then be established to select, configure, and manage the AI/ML inference tasks.

Split inference may be configured by downloading the DNN model part as part of the web application resources and then using web technologies such as WebCodes and WebNN on the UE side to do the UE inference.

# Proposal

We propose to document the IMS mapping in this contribution as part of TR26.927.