**Source: Samsung Electronics Co., Ltd.**

**Title: [FS\_AI4Media] Discussion on basic architecture for AI/ML**

**Agenda Item: 9.8**

**Document for: Agreement**

1. Introduction

This contribution provides a brief starting point for discussions on a basic architecture for AI/ML media services.

1. Scenarios

Considering the related use cases as documented in TR 22.874 and also as documented in the latest version of the Permanent Document (S4-220500), we can start from some basic scenarios for consideration of a basic architecture for AI/ML media services.

The basic starting scenarios are:

1. Delivery of a pre-trained AI/ML model from network to UE, typically at the start of an AI media service, but may also require updates during the service. At the most basic level AI/ML models can be delivered as a file (e.g. TensorFlow SavedModel, PDF5, ONNX file, NNEF file etc.) containing all the necessary information required for the UE to perform on device inference using the delivered model. For split scenarios, a (partial) AI model to be used in the UE may be delivered.
2. Split inference of a pre-trained AI/ML model(s) with two further sub-scenarios:
   1. Basic scenario with an inference in the network or in the UE.
   2. Split scenario with inferences between the network and the UE, where the intermediate data output from the network inference (resp. UE inference) is transferred to the UE (resp. network) to be used as the input for UE device inference (resp. network inference). Depending on the characteristics of the intermediate data, such as if the intermediate data is media content data, it may be practical to consider 5GMS architectures, procedures and/or protocols for the streaming delivery of such intermediate media data.
3. Service architectures

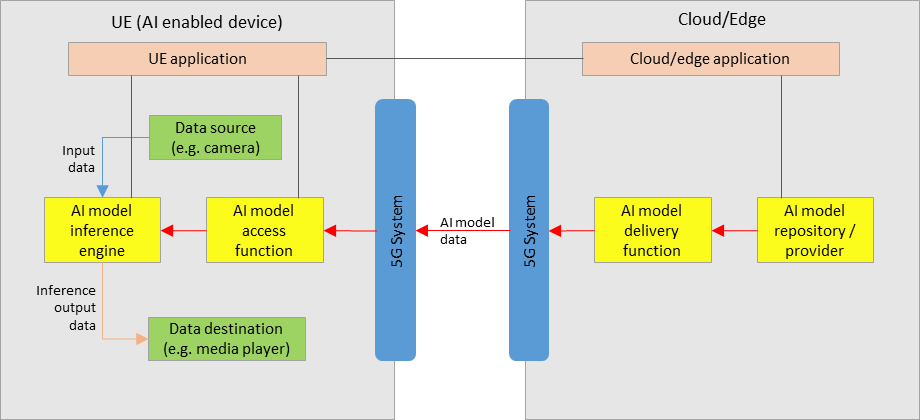


Figure 1: service architecture for AI/ML model delivery with inference in the UE

Figure 1 shows a simple service architecture for AI/ML model delivery, as described in step 1 of section 2, with an inference of a pre-trained AI/ML model in the UE, as described in scenario a) of section 2.

In the network:

* An AI model in the repository is selected for the service by the cloud/edge application, and sent to the delivery function for delivery to the UE.
* The delivery function sends the AI model data to the UE via the 5GS. The delivery function may also contain functionalities related to QoS requests and monitoring.

In the UE:

* An AI media service application provides an AI media service using the AI model inference engine and access function.
* The AI model access function receives the AI model data via the 5G system, and sends it to the inference engine.
* The AI model inference engine performs inference by using the input data from the data source (e.g. a camera, or other media source) as the input into the AI model received from the access function. The inference output data is sent to the data destination which consumes the output data (e.g. a media player).

Depending on the exact service scenario, AI model updates may be necessary during the service, and different AI model data delivery pipelines may be considered for such purposes.

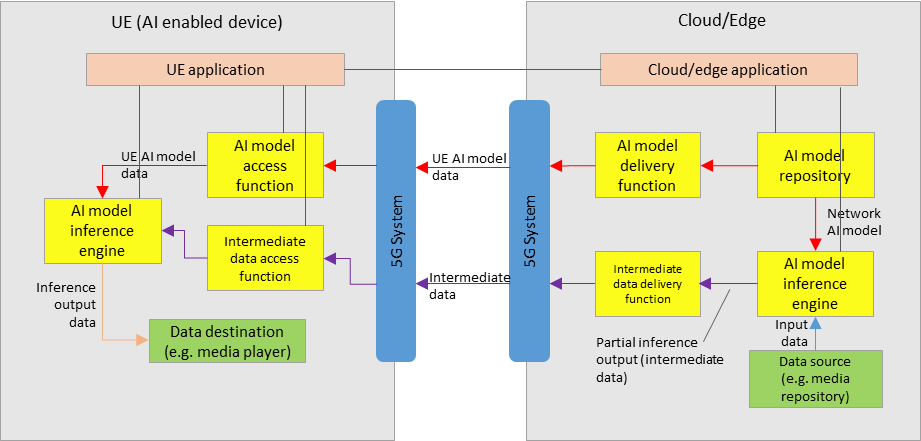


Figure 2: service architecture for split inference between the network and UE

Figure 2 shows a simple service architecture for split inferences between the network and the UE, with an AI model inference engine in the network performing network inference in the split inference, as described in scenario b) of section 2.

For the split inference scenario, additional components are required:

In the network:

* An AI model inference engine that receives both the network AI model, and input data, for network inference.
* An intermediate data delivery function which receives the partial inference output (intermediate data) from the network inference engine, and sends it to the UE via the 5GS. The delivery function may also contain functionalities related to QoS requests and monitoring.

In the UE:

* An intermediate data access function receives the intermediate data from the network, and send it to the UE inference engine for UE inference.
* The final inference output data is sent to the data destination (e.g. a media player).

Extra factors should be considered, including those such as:

* Configuration of the split inference between the network and UE
* Resource allocation and management for network inference, including ingestion of network AI model data and media data
* Intermediate data delivery pipelines between the network and UE, in particular considering the use of 5GMS defined pipelines to stream intermediate data that is media content data.
* The functionalities of certain components in figure 1 and figure 2 may overlap, and depending on the use case a combined architecture may also be considered FFS.
* Certain components may also overlap with functions defined in 5GMS, clarifications FFS.

1. Proposal

We propose to include section 3 of this contribution into the Permanent Document.