**TSG-SA5 Meeting #162 *S5-253975d1***

**, , – Revision of S5-253473**

|  |
| --- |
| *CR-Form-v12.3* |
| **CHANGE REQUEST** |
|  |
|  |  | **CR** | **draftCR** | **rev** | **-** | **Current version:** | **19.2.0**  |  |
|  |
| *For* [***HE******LP***](http://www.3gpp.org/3G_Specs/CRs.htm#_blank)*on using this form: compr**ehensive instructions can be found at* [*http://www.3gpp.org/Change-Requests*](http://www.3gpp.org/Change-Requests)*.* |
|  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ***Proposed change affects:*** | UICC apps |  | ME |  | Radio Access Network | **X** | Core Network | **X** |

|  |
| --- |
|  |
| ***Title:***  | Input to draftCR Rel-19 TS 28.105 Add generic procedure for ML model training |
|  |  |
| ***Source to WG:*** |  |
| ***Source to TSG:*** | SA5 |
|  |  |
| ***Work item code:*** | AIML\_MGT\_Ph2 |  | ***Date:*** | 2025-08-14 |
|  |  |  |  |  |
| ***Category:*** | **B** |  | ***Release:*** | Rel-19 |
|  | *Use one of the following categories:****F*** *(correction)****A*** *(mirror corresponding to a change in an earlier release)****B*** *(addition of feature),* ***C*** *(functional modification of feature)****D*** *(editorial modification)*Detailed explanations of the above categories canbe found in 3GPP [TR 21.900](http://www.3gpp.org/ftp/Specs/html-info/21900.htm). | *Use one of the following releases:Rel-8 (Release 8)Rel-9 (Release 9)Rel-10 (Release 10)Rel-11 (Release 11)…Rel-17 (Release 17)Rel-18 (Release 18)Rel-19 (Release 19) Rel-20 (Release 20)* |
|  |  |
| ***Reason for change:*** | The training NRM fragment has different IOCs that are used in the provisioning interface between MnS consumer and producer. 3GPP WGs may read our AI/ML specifications to understand how the training procedure works. Sequence flows may help them to have a better understanding of this procedure.  |
|  |  |
| ***Summary of change:*** | Add sequence flow representing generic AI/ML procedure for ML model training.  |
|  |  |
| ***Consequences if not approved:*** | Procedures describing how to use IOCs belonging to training NRM fragment is missing. |
|  |  |
| ***Clauses affected:*** | 7.x (new), Annex A.x (new) |
|  |  |
|  | **Y** | **N** |  |  |
| ***Other specs*** |  | **X** |  Other core specifications  | TS/TR ... CR ...  |
| ***affected:*** |  | **X** |  Test specifications | TS/TR ... CR ...  |
| ***(show related CRs)*** |  | **X** |  O&M Specifications | TS/TR ... CR ... |
|  |  |
| ***Other comments:*** | No stage 3 |
|  |  |
| ***This CR's revision history:*** |  |

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

7.x Generic Procedures for AI/ML management

7.x.1 ML Model Training

Figure 7.x.1-1 illustrates the generic procedure for ML model training. For consumer-initiated training, steps 1-11 apply. For producer-initiated training, steps 4, 7, 8 and 9 apply.



**Figure 7.x.1-1: Procedure for ML Model Training**

1. ML Training MnS Consumer sends a createMOI request (see createMOI operation defined in TS 28.532 [11]) to ML Training MnS Producer, for MLTrainingRequest IOC. This request includes ML model training related information (see attributes of MLTrainingRequest IOC in clause 7.3.a.1.2.2).

2. Upon reception of the request, the ML Training MnS Producer creates a MLTrainingRequest MOI (i.e. an instance of MLTrainingRequest IOC) and configures it with the received ML model training related information.

3. The ML Training MnS Producer sends a createMOI response (see createMOI operation defined in TS 28.532 [11]) to ML Training MnS Consumer. This response includes the DN of the created MLTrainingRequest MOI.

From this moment on, the ML Training MnS Producer decides when to start the ML model training process (asynchronous process) fulfilling the training request.

4. Once the ML Training MnS producer decides to start the training process, the MnS producer creates a MLTrainingProcess MOI (i.e. an instance of MLTrainingProcess IOC, see clause 7.3a.1.2.4) and configures it. In case of consumer-initiated training, the MnS producer also configures the attribute trainingRequestRef of the MLTrainingProcess MOI with the DN of the MLTrainingRequest MOI.

5. The ML Training MnS producer sends a notification (see notifyMOICreation defined in TS 28.532 [11]) to the ML Training MnS consumer, to notify the DN of the created MLTrainingProcess MOI. See NOTE below.

NOTE: For step 5, the MnS consumer needs to have a subscription in place to receive notifications on instances of MLTrainingProcess IOC. The consumer will receive a notification for every MLTrainingProcess MOI that system creates. The consumer can check the mlTrainingRequestRef attribute of every MLTrainingProcess MOI that gets notified, to identify which is the MLTrainingProcess MOI that is in charge of processing MLTrainingRequest MOI. Once identified, the consumer can remove the subscription.

6. Based on the information received in the notification, the training MnS consumer can now check the status of the training process, by reading the value of the progressStatus attribute in the MLTrainingProcess MOI. To that end, the MnS consumer can use query (see steps 6a-1 and 6a-2) or subscribe-notify (see step 6b) mechanisms. The step 6 applies while the training process is being executed.

6-a1. The ML Training MnS consumer sends a getMOIAttributes request (see getMOIAttributes operation defined in TS 28.532 [11]) to ML Training MnS producer, for MLTrainingProcess MOI . The consumer sets attribute baseObjectInstance with the DN of the MLTrainingProcess MOI, and the attribute filter with progressStatus.

6-a2. The ML Training MnS producer sends a getMOIAttributes response (see getMOIAttributes operation defined in TS 28.532 [11]) to ML Training MnS consumer, returning the value of progressStatus attribute.

6-b. The ML Training MnS producer sends a notification to the MnS consumer when the progressStatus attribute in the MLTrainingProcess MOI changes. See NOTE below.

NOTE: For step 6-b, the MnS consumer needs to have a subscription in place scoping the the MLTrainingProcess MOI, with triggering events limited to changes on the value of the progressStatus attribute.

When the training process gets finished, the ML training MnS Producer performs a set of actions described in steps 7, 8 and 9. The execution of these steps is interchangeable.

7. The ML Training MnS producer changes the value of the progressStatus.status attribute to “FINISHED”. Upon this attribute value change, the consumer is aware of training process completion.

8. The ML Training MnS Producer creates a MLModel MOI (i.e. an instance of MLModel IOC), which represents the trained ML model.

9. The ML Training MnS producer creates a MLTrainingReport MOI (i.e. an instance of MLTrainingReport IOC, see clause 7.3a.1.2.3) and configures it. The MnS producer configures the attribute trainingProcessRef of the MLTrainingReport MOI with the DN of the MLTrainingProcess MOI, and the attribute mlModelGeneratedRef of the MLTrainingReport MOI with the DN of the created MLModel MOI.

From this moment on, the MnS consumer can access to the created report by reading the mLTrainingReportRef attribute in MLTrainingProcess MOI, as described in steps 10 and 11.

10. The ML Training MnS consumer sends a getMOIAttributes request (see getMOIAttributes operation defined in TS 28.532 [11]) to ML Training MnS producer, for MLTrainingProcess MOI . The consumer sets attribute baseObjectInstance with the DN of the MLTrainingProcess MOI, and the attribute filter with mLTrainingReportRef.

11. The ML Training MnS producer sends a getMOIAttributes response (see getMOIAttributes operation defined in TS 28.532 [11]) to ML Training MnS consumer, returning the value of mLTrainingReportRef attribute.

|  |
| --- |
| **Next Change** |

Annex A (informative):
PlantUML source code for NRM class diagrams

# A.X Generic Procedures for AI/ML management

## A.X.1 ML Model Training

@startuml

participant "ML Training MnS Consumer" as consumer

participant "ML Training MnS Producer" as producer

consumer -> producer: 1. Request to create an \nMLTrainingRequest MOI

producer -> producer:2. Create and configure \nMLTrainingRequest MOI

consumer <- producer: 3. Response to create \nMLTrainingRequest MOI

==Training process starts==

producer ->producer: 4. Create and configure \nMLTrainingProcess MOI

consumer <-producer: 5. Notify MLTrainingProcess MOI creation

loop While ML training process is running

alt MnS consumer uses query mechanism

consumer ->producer: 6a-1. Read MLTrainingProcess MOI \n(filter = "progressStatus")

consumer <-producer: 6a-2. Response ("progressStatus" value)

end

alt MnS consumer is subscribed to MLTrainingProcess MOI changes

consumer<-producer:6b. Notify MLTrainingProcess MOI change \n(scope = "progressStatus")

end

end

==Training process finishes==

producer->producer:7. Modify MLTrainingProcess MOI \n (set "progressStatus.status" = “FINISHED”)

producer->producer:8. Create and configure \nMLModel MOI

producer->producer:9. Create and configure \nMLTrainingReport MOI

consumer-> producer: 10. Read MLTrainingProcess MOI \n (filter = "mLTrainingReportRef")

consumer <-producer: 11. Response ("mLTrainingReportRef"  value)

@enduml

|  |
| --- |
| **End of Changes** |