ORAN-WG2.AIML.v01.00

Technical Report

O-RAN Working Group 2
AI/ML workflow description and requirements

Revision History

Date	Revision	Author	Description
2019.03.12	01.00.01	Rittwik Jana, Reuben Klein, Dhruv Gupta, Qi Sun	Draft for template
2019.03.15	01.00.02	Rittwik Jana, Reuben Klein	First draft with Scope, references, acronyms
2019.03.22	01.00.03	C. Santori, RaviKanth	Common terminology Comments/clarifications
2019.04.01	01.00.04	R. Jana, C. Santori, Yang Jiaolong	Edited common terminology, added pipeline definitions, modified ML lifecycle figure
2019.04.05	01.00.05	All	Addressed comments from team call
2019.04.08	01.00.06	AT&T, Nokia, Orange, Aricent	Updated scope, figures and comments
2019.04.22	01.00.07	RaviKanth	Moved AL/ML terminology to S2, updated fig in S5.2 and added details for figures in S3 updated figure-2 based on updated fig uploaded by Nokia
2019.04.23	01.00.08	R. Jana	Merged inputs, generate clean version for approval
2019.05.03	01.01.00	R. Jana	Approved version, final formatting
2019.06.02	01.01.02	AT&T, CMCC, Nokia, Altran	Updating draft with Approved CR 2019.05.27-WG2-C-AI_ML Procedure_Interface and Requirements_v12
2019.07.05	01.01.03	CMCC, AT&T	Updating draft as per WG2 f2f meeting comments
2019.07.12	01.01.03	Manoop Talasila	Updated ML model resource discovery requirements and ML based xApp design details
2019.07.20	01.01.04	R. Jana, D Gupta, Q Sun	Updated draft 01.01.04 with corrections suggested by CR matrix in CMCC-ATT-WG2-ML Spec-v01.01.03-review-comments.xls
2019.08.03	01.01.05	R. Jana, D. Gupta	Updated to draft 01.01.05 reflecting changes suggested in WG2-ML Spec-v01.01.04-review-comments, 2019-07-30-ATT-CR -AIMLv01.01.04-Phases_description, new figures from WG2_2019.07.20_Nokia-O-RAN ML Workflow INTERFACES_v8.pptx, AIML_07_20_19-draftspec-figures_v2.pptx
2019.08.09	01.01.06	R. Jana, D. Gupta, Q. Sun	Updated sequence figure as per 2019-07-30-ATT-CR -AIMLv01.01.04-PlantUML_figure; cleaned up extraneous text
2019.08.19	01.01.07	J. Power, R.Jana, D Gupta, Q. Sun	Incorporating comments from John Power
2019.09.01	01.01.08	R.Jana, D.Gupta, Q. Sun, J. Power	O-RAN Adopter license agreement, final edits
2019.09.11	01.01.09	R.Jana	Final edits, removed comments resolved

2019. 09.14	01.01. 10	Q.Sun, R.Jana	Final edits, removed all the comments, capture the unresolved comments for the next version.
2019. 10.08	01.00	R. Jana	Spec renamed to 01.00 for publishing (as per O-RAN guidelines)

1

Contents

Chapter 1 Introduction

1.1 Scope

This Technical Specification has been produced by O-RAN Alliance.

The contents of the present document are subject to continuing work within O-RAN WG2 and may change following formal O-RAN approval. In the event that O-RAN Alliance decides to modify the contents of the present document, it will be re-released by O-RAN Alliance with an identifying change of release date and an increase in version number as follows:

Release x.y.z

where:

- x the first digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc. (the initial approved document will have x=01).
- y the second digit is incremented when editorial only changes have been incorporated in the document.
- z the third digit included only in working versions of the document indicating incremental changes during the editing process.

The current document addresses the overall architecture and solution for AI/ML related requirements for the use-cases described in O-RAN WG2 UCR doc [ORAN-WG2.UCR.01.01.00]. The document provides the terminology, workflow, and requirements, related to AI/ML model training, and its distribution and deployment in the Radio Access Network (RAN).

1.2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document in Release 15.

[1] 3GPP TR 21.905: “Vocabulary for 3GPP Specifications”

[2] 3GPP TS 38.401: "NG-RAN; Architecture description".

[3] "O-RAN: towards an Open and smart RAN", O-RAN white paper, *https://www.o-ran.org/s/O-RAN-WP-FInal-181017.pdf*

[4] ORAN-WG2.UCR.01.01.00, ORAN WG2 use case requirements

Informative references

[i.x1] ETSI GR NFV IFA 023: "Network Functions Virtualisation (NFV); Management and Orchestration; Report on Policy Management in MANO; Release 3".

[i.x2] ETSI GS ZSM 001: "Zero-touch Network and Service Management (ZSM); Requirements based on documented scenarios".

[i.x3] OPNFV Wiki, copper project, https://wiki.opnfv.org/display/copper

[i.x4] OpenStack Wiki, congress project, https://wiki.openstack.org/wiki/Congress

[i.x5] ONAP Wiki, Policy Framework, https://wiki.onap.org/display/DW/The+ONAP+Policy+Framework

[i.x6] B. Moore, E. Ellesson, J. Strassner, A. Westerinen, "Policy Core Information Model," RFC 3060, IETF, February 2001

[i.x7] A. Westerinen et.al. "Terminology for Policy-Based Management", RFC 3198, IETF, November 2001

[i.x8] Morris Sloman, "Policy driven management for distributed systems" in Journal of Network and Systems Management, Plenum Press, Vol. 2, No. 4, 1994, pp. 333-360

[i.x9] Acumos ML model schema -

[i.x10] Acumos ML model signature - http://acumosr.research.att.com/#/marketSolutions?solutionId=cdb34d2d-f46a-4658-97e8-8b30efb9451c&revisionId=d3bb02db-3079-4b00-9114-0f0693bbc635&parentUrl=marketplace#md-model-detail-template

[i.x11] O-RAN WG1.OAM Architecture -v01.00 - ACC-2019.07.02-oRAN.WG1.OAM Draft Arch Spec-v01.00.015.docx

1.3 Definitions and Abbreviations

1.3.1 Abbreviations

For the purposes of the present document, the terms and definitions given in 3GPP TR 21.905 [1] and the following apply. A term defined in the present document takes precedence over the definition of the same term, if any, in 3GPP TR 21.905 [1].

NMS: A Network Management System

O-DU: O-RAN Distributed Unit: a logical node hosting RLC/MAC/High-PHY layers based on the 7-2x fronthaul split defined by O-RAN.

O-RU: O-RAN Radio Unit: a logical node hosting Low-PHY layer and RF processing based on the 7-2x fronthaul split defined by O-RAN.

Non-RT RIC: O-RAN non-real-time RAN Intelligent Controller: a logical function that enables non-real-time control and optimization of RAN elements and resources, AI/ML workflow including model training and updates, and policy-based guidance of applications/features in Near-RT RIC.

Near-RT RIC: O-RAN near-real-time RAN Intelligent Controller: a logical function that enables near-real-time control and optimization of RAN elements and resources via fine-grained data collection and actions over E2 interface.

O1: Interface between orchestration & management entities (Orchestration/NMS) and O-RAN managed elements, for operation and management, by which FCAPS management, Software management, File management and other similar functions shall be achieved.

A1: Interface between Non-RT RIC and Near-RT RIC to enable policy-driven guidance of Near-RT RIC applications/functions, and support AI/ML workflow.

E2: Interface between Near-RT RIC and underlying RAN functions (CU-CP, CU-UP, and DU).

1.3.2 Abbreviations

For the purposes of the present document, the abbreviations given in 3GPP TR 21.905 [1] and the following apply. An abbreviation defined in the present document takes precedence over the definition of the same abbreviation, if any, in 3GPP TR 21.905 [1].

eNB	eNodeB (applies to LTE)
gNB	gNodeB (applies to NR)
O-DU	O-RAN Distributed Unit
O-RU	O-RAN Radio Unit
O-CU	O-RAN Central Unit
RIC	[O-RAN] RAN Intelligent Controller
Non-RT RIC	Non-real-time RIC
Near-RT RIC	Near-RT RIC
QoE	Quality of Experience
KQI	Key Quality Indicator
KPI	Key performance indicator
CNN	Convolutional neural network
PCA	principal components analysis
RL	reinforcement learning
DRL	deep reinforcement learning
GPU	graphics processing unit
KNN	k nearest neighbors
LSTM	long short-term memory
ML	machine learning

NN	neural network
RL	reinforcement learning
RNN	recurrent neural network
SMO	service management and orchestration
SVM	support vector machine

Chapter 2 Machine Learning

Machine learning is a field of study that provides computers the ability to learn without being explicitly programmed. The ability to learn useful information from input data can help improve RAN or network performance. For example, convolutional neural networks and recurrent neural networks can extract spatial features and sequential features from time-varying signal strength indicators (e.g., RSSI).

This chapter introduces some of the common terminology related to AI/ML based use-cases development in context of O-RAN architecture.

2.1 Common terminology and Definitions

Table 1 - Common terminology

Definitions	Note/example
Application: An application is a complete and deployable package, environment to achieve a certain function in an operational environment. An AI/ML application is one that contains some AI/ML models.	Generally, an AI/ML application should contain a logically top-level AI/ML model and application-level descriptions
ML-assisted solution: A solution which addresses a specific use case using Machine-Learning algorithms during operation.	As an example, video optimization using ML is an ML-assisted solution.
ML model: The ML methods and concepts used by the ML-assisted solution. Depending on the implementation a specific ML model could have many sub-models as components and the ML model should train all sub-models together.	ML models include supervised learning, unsupervised learning, reinforcement learning, deep neural network, and depending on use-case, appropriate ML model has to be chosen. Separately trained ML models can also be chained together in a ML pipeline during inference.
ML workflow: A ML workflow is the process consisting of data collection and preparation, model building, model training, model deployment, model execution, model validation, continuous model self-monitoring and self-learning/retraining related to ML-assisted solutions	Based on ML model chosen, some or all of the phases of workflow will be included.
ML (model) life-cycle: The life-cycle of the ML model includes deployment, instantiation and termination of ML model components.	These are operational phases: the initial training, inference, possible re-training

ML pipeline: The set of functionalities, functions, or functional entities specific for an ML-assisted solution.	a ML pipeline may consist of one or several data sources in a data pipeline, a model training pipeline, a model evaluation pipeline and an actor.
ML training host: The network function which hosts the training of the model	Non-RT RIC can also be a training host. ML training can be performed offline using data collected from the RIC, O-DU and O-RU.
ML inference host: The network function which hosts the ML model during inference mode (which includes both the model execution as well as any online learning if applicable).	The ML inference host often coincides with the Actor. The ML-host informs the actor about the output of the ML algorithm, and the Actor takes a decision for an action.
Actor: The entity which hosts an ML assisted solution using the output of ML model inference.	
Action: An action performed by an actor as a result of the output of an ML assisted solution.	
Subject of action: The entity or function which is configured, controlled, or informed as result of the action.	
Model training information: Information needed for training the ML model.	This is the data of the ML model including the input plus optional labels for supervised training
Model inference information: Information needed as input for the ML model for inference.	The data needed by an ML model for training and inference may largely overlap, however they are logically different.

Figure 1 depicts the use of the ML components and terminologies as described in Table 1.

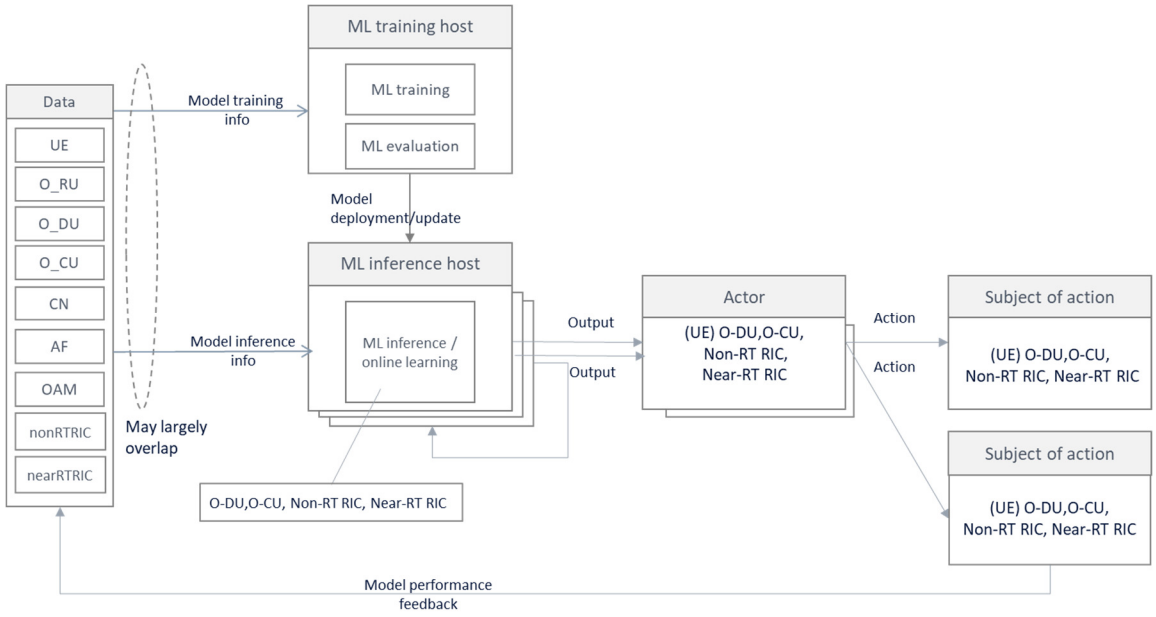

Figure 1 - ML modeling terminology

2.2 General principles

Principle 1: In O-RAN we will always have some offline learning as a proposed best practice (even for reinforcement learning type of scenarios). In the current document, offline training means a model is first trained with offline data, and trained model is deployed in the network for inference. Online training refers to scenarios such as reinforcement learning, where the model 'learns' as it is executing in the network. However, even in the latter scenario, it is possible that some offline training may happen.

Principle 2: A model needs to be trained and tested before deploying in the network. A completely untrained model will not be deployed in the network.

Chapter 3 Types of Machine Learning algorithms

This section provides a view of how the different ML algorithms can be deployed and realized in O-RAN architecture. It does not detail or recommend the various machine learning algorithms available or recommend specific ML algorithms that should be applied to the use-cases realized in O-RAN architecture.

3.1 Supervised learning

Input data is called training data and has a known label or result. Supervised learning is a machine learning task that aims to learn a mapping function from the input to the output, given a labeled data set.

1. Regression: Linear Regression, Logistic Regression
2. Instance-based Algorithms: k-Nearest Neighbor (KNN)
3. Decision Tree Algorithms: CART
4. Support Vector Machines: SVM
5. Bayesian Algorithms: Naive Bayes
6. Ensemble Algorithms: Extreme Gradient Boosting, Bagging: Random Forest

Supervised learning can be further grouped into Regression and Classification problems. Classification is about predicting a label whereas Regression is about predicting a quantity.

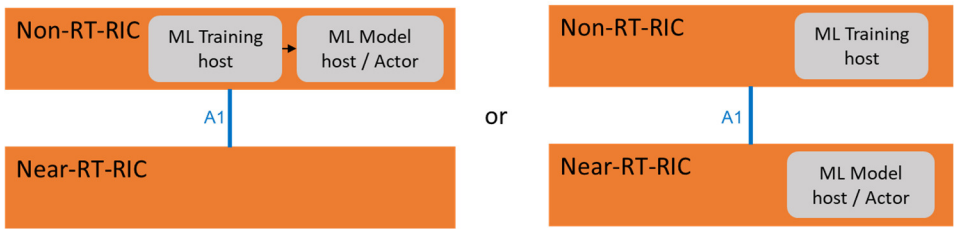

Figure 2 - Supervised learning model training and actor locations

In supervised learning (see Figure 2), Non-RT RIC is part of the SMO and thus is part of the management layer. ML training host and ML model host/actor can be part of Non-RT RIC or Near-RT RIC.

3.2 Unsupervised learning

Input data is not labeled and does not have a known result. Unsupervised learning is a machine learning task that aims to learn a function to describe a hidden structure from unlabeled data. Some examples of unsupervised learning are K-means clustering and principal component analysis (PCA).

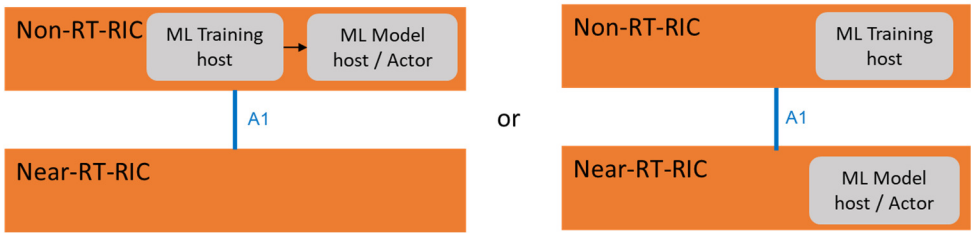

Figure 3 - Unsupervised learning model training and actor locations

In unsupervised learning (see Figure 3), ML training host and ML model host/actor can be part of Non-RT RIC or Near-RT RIC.

3.3 Reinforcement learning

A goal-oriented learning based on interaction with environment. In reinforcement learning (RL), the agent aims to optimize a long-term objective by interacting with the environment based on a trial and error process. There are several RL algorithms

- Q-learning
- Multi-armed bandit learning
- Deep RL

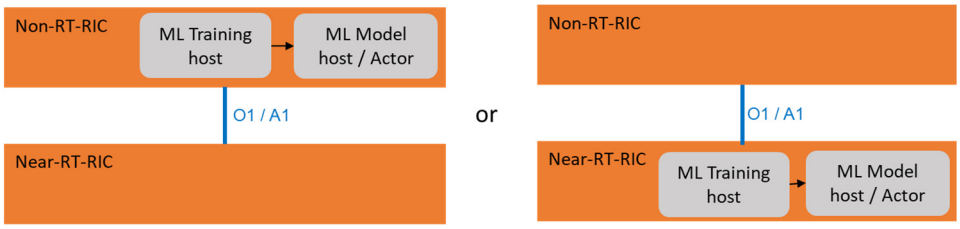

Figure 4- Reinforcement learning model training and actor locations

In reinforcement learning (see Figure 4), ML training host and ML model host/actor shall be co-located as part of Non-RT RIC or Near-RT RIC.

3.4 Mapping AI/ML functionalities into O-RAN control loops

There are three types of control loops defined in O-RAN. ML assisted solutions fall into the three control loops. Time scale of O-RAN control loops depend on what is being controlled, e.g. system parameters, resources or radio resource management (RRM) algorithm parameters. For example, if O-RAN control loop adapts the parameters of RRM algorithms, its time scale is slower than that of the RRM algorithm.

Loop 1 deals with per TTI msec level scheduling and operates at a time scale of the TTI or above. Loop 2 operates in the near RT RIC operating within the range of 10-500 msec and above (resource optimization). Loop 3 operates in the Non-RT RIC at greater than 500 msec (policies, orchestration). It is not expected that these loops are hierarchical but can instead run in parallel.

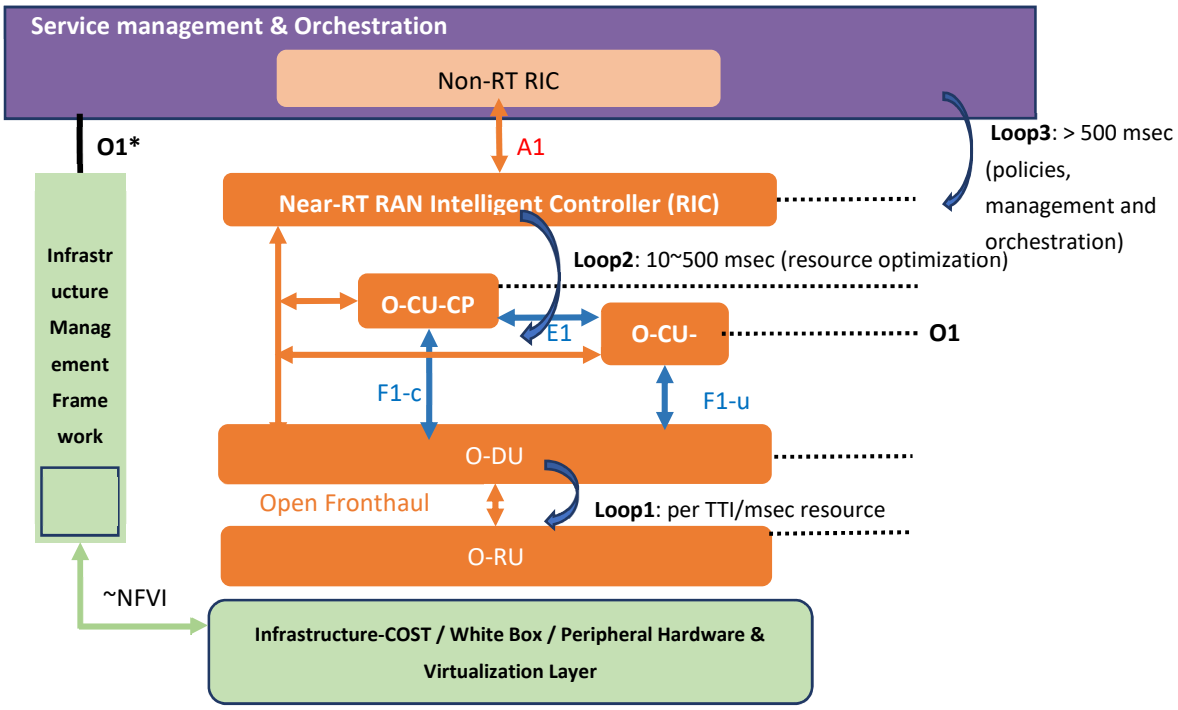

Figure 5 - Control loops in O-RAN

Figure 5 shows the three control loops in O-RAN architecture. AI/ML related functionalities can be mapped into the three loops. The location of the ML model training and the ML model inference for a use case depends on the computation complexity, on the availability and the quantity of data to be exchanged, on the response time requirements and on the type of ML model. For example, online ML model for configuring RRM algorithms operating at the TTI time scale could run in O-DU, while the configuration of system parameters such as beamforming configurations requiring a large amount of data with no response time constraints can be performed in the Non-RT RIC and Orchestration and management layer where intensive computation means can be made available.

In the first phase of O-RAN, ML model training will be considered in the Non-RT RIC and ML model inference will be considered in loops 2 and 3. For loop2, the ML inference is typically running in Near-RT RIC For Loop 1, the ML model inference is typically running in an O-DU. ML workflow on loop 1 is FFS. While ML model implementation in O-RU could be envisaged, it is presently not supported in O-RAN.

Chapter 4 Procedure/Interface framework, Data/Evaluation pipelines

4.1 AI/ML General Procedure and Interface Framework

This chapter first provides the general framework of AI/ML procedure and interfaces, which addresses the ML components rather than network functions (non/Near-RT RIC, etc.). The potential mapping relationship between the ML components and network functions, interfaces defined in O-RAN are also illustrated in Figure 6.

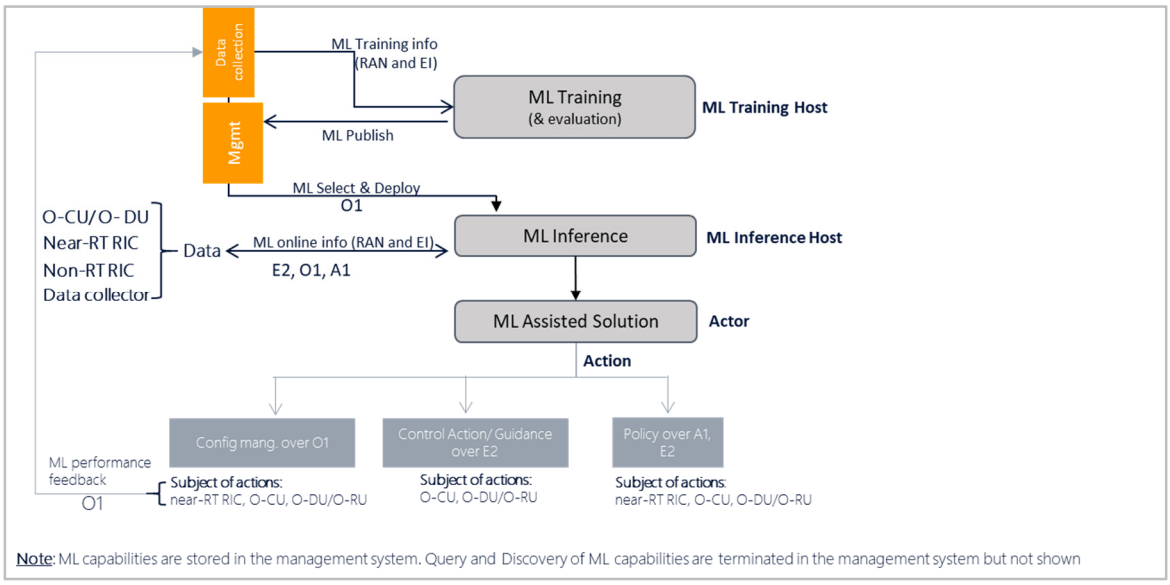

Figure 6 - ML training host and inference locations

Note: *ML capabilities shall be stored in the management system (FFS). Query and Discovery of ML capabilities are terminated in the management system but not shown. ML inference host often coincides with the actor*

The deployment scenarios that are considered for ML architecture/framework in O-RAN architecture are

1. Deployment Scenario 1.1: Non-RT RIC acts as both the ML training and inference host
2. Deployment Scenario 1.2: Non-RT RIC acts as the ML training host and the Near-RT RIC as the ML inference host
3. Deployment Scenario 1.3: Non-RT RIC acts as the ML training host and the O-CU/O-DU as the ML inference host (for FFS)

In addition, for reinforcement learning based ML model-based deployment, both ML training and ML interference host shall be co-located on same MF.

Table 2 shows the various deployment scenarios and interfaces.

Table 2 - AI/ML deployment scenarios

Deployment Scenario	ML Training Host	ML Inference Host	Interface for ML model deployment / update	Subject of Action	Action from inference host to subject		Enrichment data for inference
					Config Mgmt. (CM)	Policy / Control	
Scenario 1.1	SMO/Non-RT RIC	Non-RT RIC	SMO internal	Near-RT RIC	O1	A1 (policy)	SMO internal
				O-CU, O-DU, O-RU	O1	N/A	SMO internal
Scenario 1.2	SMO/Non-RT RIC	Near-RT RIC	O1, O1*	Near-RT RIC	near-RT RIC internal	near-RT RIC internal	A1
				O-CU, O-DU, O-RU	N/A	E2 (control/policy)	E2 (if applicable)
Scenario 1.3 (FFS)	SMO/Non-RT RIC	O-CU / O-DU	O1, O1*	O-CU, O-DU, O-RU	FFS	FFS	FFS

Note: Configuration management for scenario 1.2 via E2 is FFS;

O1^ - Non-RT RIC can use SMO internal interfaces to trigger configuration changes over O1

Based on the framework, some key phases of machine learning are expected to be applied to any ML-assisted solution planned in O-RAN architecture. Any use case defined for ML-assisted solution shall have one or more phases (as applicable) and the phases are defined below:

1. **ML model capability query/discovery**

This procedure shall be executed whenever AI/ML model is to be used for ML-assisted solution. This procedure can be executed at start-up or run-time (when a new ML model is to be executed or existing ML model is to be updated). The SMO will discover various capabilities and properties of the ML inference host, such as:

a) Processing capability of HW where ML model is going to be executed (for example: resources available such as CPU/GPU, memory etc. that can be allocated for ML model inference).

b) Properties such as supported ML model formats and ML engines (for example: Protobuf, JSON, or any ONAP specific VES data formats).

c) NFVI based architecture support in MF to run ML model(s)

d) Data-sources available to run ML-pipeline (for example: support for data streams, data lake, or any specific database access)

This discovery of the capabilities shall be used to check if a ML model can be executed in the target ML inference host (MF), and what number and type of ML models can be executed in the MF.

Note: Exact mechanism and contents of capabilities discovery is FFS.

2. **ML model Selection and Training**

This procedure corresponds to design time selection and training of a ML model in relation with a specific ML-assisted solution (use case) to be executed. The ML designer will select and onboard the ML model and relevant meta data into the SMO environment. Utilizing on the ML training data collection, the ML training host will initiate the model training. Once the model is trained and validated, it is published back in the SMO catalogue.

At this stage, the ML designer can check whether the trained model can be deployed in the ML inference host, by mapping the ML model requirements to HW and performance properties discovered from Step 1. Upon successful validation, ML designer will inform the SMO to initiate model deployment.

3. **ML model Deployment and Inference**

The AI/ML model that is selected for the use case can be deployed via containerized image to MF where ML model shall be executing. This also includes configuration of ML inference host with AI/ML model description file.

Note: The O1 interface mechanism for ML model deployment is being specified by WG1.

Once the ML model is deployed and activated, ML online data shall be used for inference in ML-assisted solutions, which includes:

a) 3GPP specific events/counters (across all different Managed Elements) over O1/E2 interface
    a. Events: 3GPP 32.423
    b. Counters: 3GPP 32.425
b) Non-3GPP specific events/counters (across all different Managed Elements) over O1/E2 interface (to be defined in ORAN WGs)

c) Enrichment information from non-RT RIC over A1 interface (to be defined in ORAN WGs)

Based on the output of the ML model, the ML-assisted solution will inform the Actor to take the necessary actions towards the Subject. These could include CM changes over O1 interface, policy management over A1 interface, or control actions or policies over E2 interface, depending on the location of ML inference host and Actor.

4. **ML model performance monitoring**

The ML inference host is expected to feedback or report the performance of the ML model to the ML training host so that the ML training host can monitor the performance of the ML model and potentially update the model. Based on the use-case, specific set of data as applicable for use-case shall be used for ML model re-training. Based on the performance evaluation, either some guidance can be provided to use a different model in the ML inference host, or a notification can be sent indicating the need for retraining the model.

Note: Feedback mechanism and how the ML model switching can occur at runtime is FFS.

5. **ML model redeploy/update**

Based on the feedback and data received from various MFs, the ML performance evaluation module can inform the ML designer that an update is required to the current model. The ML designer will initiate the model selection and training step, but with the existing trained model. Once a new model has been trained, it will be deployed as described in Step 3, and the updated model will be used for ML inference.

4.2 Model Design and Composition

ML model design is the first step to conceiving the initial model. This requires connecting to data sources, parsing messages and tokenizing to create and select features. This activity is offline and requires data exploration mechanisms to help the model designer.

4.3 Data, Model Training and Model Evaluation pipeline

This section describes the data, model training and evaluation pipelines.

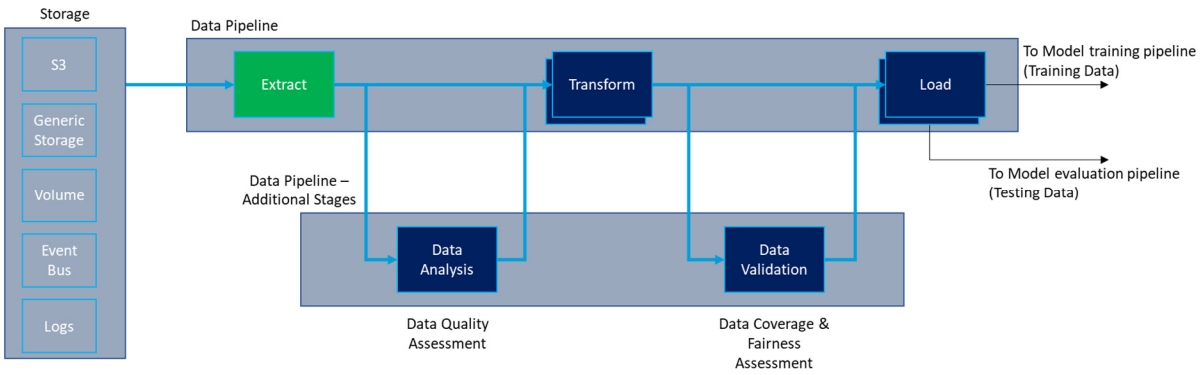

Figure 7 - Data pipeline

Figure 77 defines the data pipelines. The “extract-transform-load” (ETL) process describes how data can be extracted from storage, transformed and loaded into training and testing sets. Additional data quality and validation stages can be inserted into the ETL pipeline. Data cleaning can also be part of the Transform block. This is outside the scope of WG2.

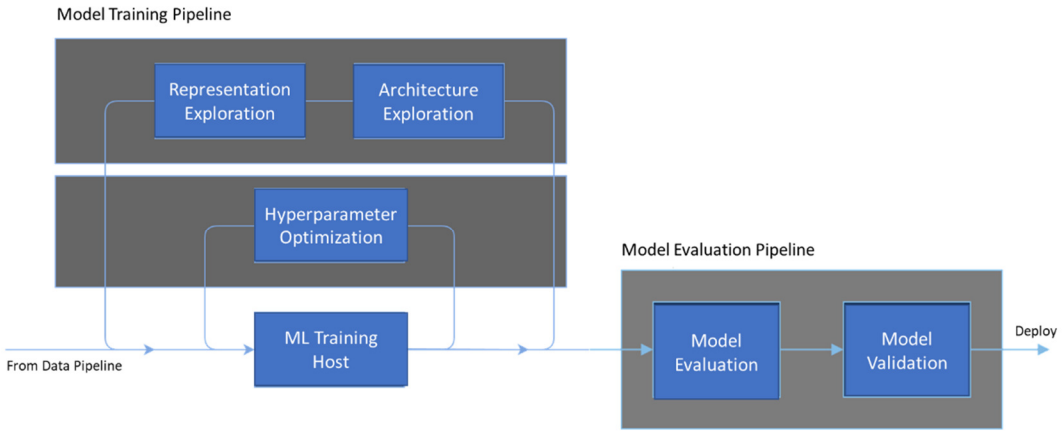

Figure 8 - Model training and evaluation pipelines

Figure 88 shows the model training and evaluation pipelines. Model training pipeline may change with model types. Model evaluation pipeline, however, is a more generic process. Model evaluation can be used to evaluate a single model or extended to select the best model from a range of models.

The end-to-end training process includes the following

- Fulfills the data requirements of the model (format, sample distribution, extent)
- Connects with requisite data via Data Pipeline (simplest ex: a data broker)
- Partitions data into appropriate sets (training, validation, testing, sample)
- Keeps Training data cached for error recovery and subsequent usage
- Manages model training though all phases
- Implements training by invoking a Model Training Pipeline
- Training Client tunes model parameters during training phases
- Scoring client monitors performance in order to declare training phase complete
- Communicates with license manager for usage and versioning

4.4 ML Model Lifecycle Implementation Example

The section provides an example (see Figure 9) of ML model lifecycle implementation example and key phases involved in the design and deployment in O-RAN architecture.

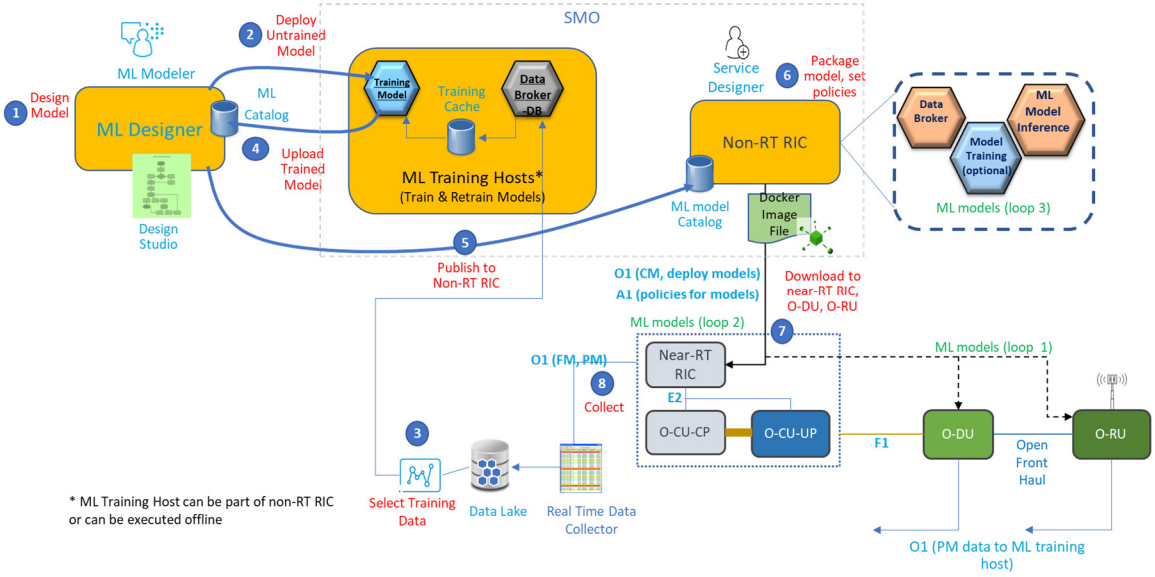

Figure 9 - ML model lifecycle (an implementation example)

Note: ML Model capability query and discovery can occur in ML designer and Non-RT RIC.

The typical steps involved in AL/ML based use-case application in O-RAN architecture is shown in Figure 910 considering supervised/unsupervised learning ML models. The steps for reinforcement model could vary with respect to ML training host and the related interaction flows.

1. ML Modeler uses a designer environment along with ML toolkits (e.g., scikit-learn, R, H2O, Keras, TensorFlow) to create the initial ML model
2. The initial model is sent to training hosts for training
3. The appropriate data sets are collected from the Near-RT RIC, O-CU and O-DU to a data lake and passed to the ML training hosts.
4. The trained model/sub models are uploaded to the ML designer catalog (one such open source catalog platform is AcumosAI). The final ML model is composed.
5. The ML model is published to Non-RT RIC along with the associated license and metadata.
6. Non-RT RIC creates a containerized ML application containing the necessary model artifacts (when using AcumosAI, the ML model's container is created in Acumos catalog itself).
7. Non-RT RIC deploys the ML application to the Near-RT RIC, O-DU and O-RU using the O1 interface. Policies are also set using the A1 interface.
8. PM data is sent back to ML training hosts from Near-RT RIC, O-DU and O-RU for retraining.

Note that Near-RT RIC can also update ML model parameters at runtime (e.g., gradient descent) without going through extensive retraining. Training hosts and ML designers can also be part of Non-RT RIC.

Chapter 5 Deployment Scenarios

This chapter describes the high-level architecture of deployment scenarios defined in Section 5.1 and also captures the sequence diagrams to show end-to-end flows.

The current version captures the deployment scenarios 1.1 (see Figure 10) and 1.2 (see Figure 11) only, and scenario 1.3 is not in current scope of document and are FFS.

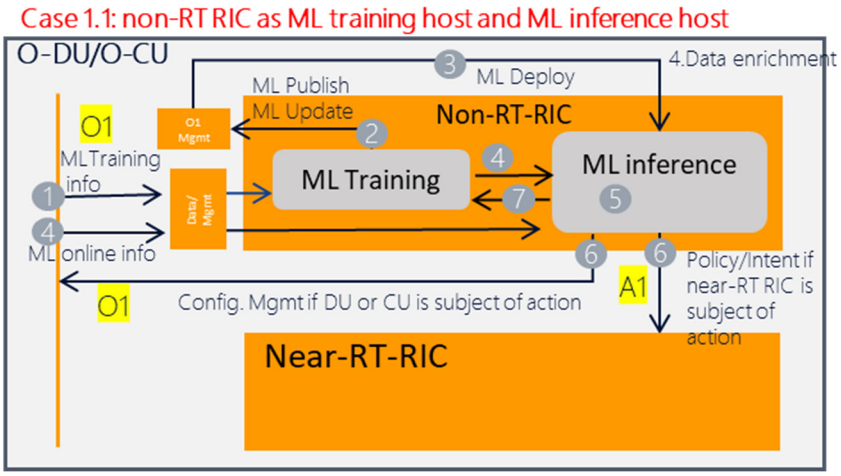

Figure 10 – Deployment scenario 1.1 - ML training and inference host locations

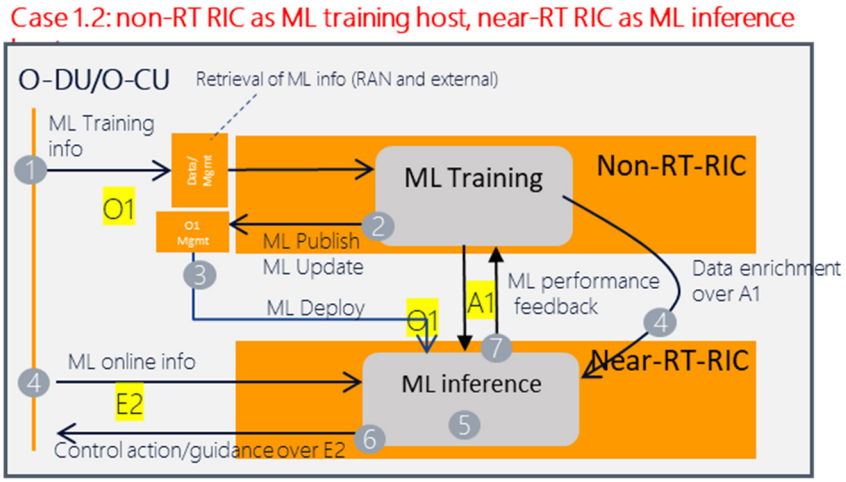

Figure 11 - Deployment scenario 1.2 - ML training and inference host locations

5.1 Sequence Diagram for Deployment Scenarios 1.1 and 1.2

The sequence diagram (see Figure 12) for Deployment Scenario-1.1 (SMO/Non-RT RIC for model training, Non-RT RIC for model inference host) and Deployment Scenario-1.2 (SMO/Non-RT RIC for model training, Near-RT RIC as model inference host) is captured below

Figure 12 - Non-RT RIC as ML training host, Non-RT RIC or Near-RT RIC as ML inference host (Note that the SMO components are defined in WG1 OAM architecture, Appendix B [i.x11])

Chapter 6 Requirements

6.1 Functional Requirements

[Editor notes]: This section describes the functional requirements for A1 interface and Non-RT RIC.

[REQ-Non-RT RIC-FUN1]	Non-RT RIC may request/trigger ML model training in training hosts.

Notes: Regardless of where the model is deployed and executed, non-RT RIC should request/trigger ML model training. Note that ML models may be trained and not currently deployed. Implicitly, model re-training and model performance/evaluation.

[REQ-Non-RT RIC-FUN2]	Non-RT RIC shall provide a query able catalog for ML designer to publish/install trained ML models (executable software components) and Non-RT RIC will provide discovery mechanism if a particular ML model can be executed in the target ML inference host (MF), and what number (and type) of ML models can be executed in the MF.

Notes: Non-RT RIC is a component of the SMO framework, i.e., one component of the NMS. The catalogue is not only for external ML market place or platform to publish the models, but also the source for any internal models as well. Non-RT RIC can also connect to external ML catalogues via SMO specific interfaces (interface specification is not in scope of document). There are three types of catalogs namely (design-time catalog (outside non-RT RIC in other ML platforms), training/deployment-time catalog (inside non-RT RIC), and run-time catalog (inside near-RT RIC for scenario 1.2)). In scenario 1.1 where ML models are trained, deployed and executed in non-RT RIC.

[REQ-Non-RT RIC-FUN3]	Non-RT RIC shall support necessary capabilities (enable executable software to be installed, e.g., containers) for ML model inference in support of ML assisted solutions running in non-RT RIC

Notes: ML engines are packaged s/w executable libraries that provide the necessary routines to run the model.

Note: As an example, policies to switch and activate ML model instances under different operating conditions (busy hour vs non-busy hour or seasonal changes, etc.)

[REQ-Non-RT RIC-FUN5]	Non-RT RIC shall be able to access feedback data over O1 interface on ML model performance and perform necessary evaluation.

Note: PM and FM stats for ML model are relayed over O1. If the ML model fails during runtime an alarm can be generated as feedback to non-RT RIC. How well the ML model is performing in terms of accuracy of prediction or other operating statistics it produces can be sent to non-RT RIC over O1.

Table 4.1.0-1

REQ	Description
[REQ-O1-FUN1]	O1 interface shall support deployment and update of the ML models as a packaged s/w executable (e.g., in a container).
[REQ-O1-FUN2]	O1 interface shall support PM and FM data collection for ML models, including fine-grained events/counters needed for ML training and inference.
[REQ-O1-FUN3]	O1 interface shall support collection of ML relevant capabilities of the managed function where the model is to be deployed for inference.

6.2 Non-Functional Requirements

[Editor notes]: This section describes the non-functional requirements for A1 interface and Non-RT RIC, e.g., security.

REQ	Description
[REQ-O1-NONFUN1]	O1 interface shall support scaling ML model instances running in target ML inference host (MF) by observing resource utilization in MF.

Note: The environment where the ML model instance is running will monitor resource utilization (e.g., in ORAN-SC there is a component call ResourceMonitor in near-RT RIC; similarly, in non-RT RIC there needs to be a ResourceMonitor that continuously monitors resource utilization). If resources are low or fall below a certain threshold, the runtime environment in near-RT RIC and non-RT RIC needs to provide a scaling mechanism to add more ML instances. K8s runtime environments typically provide auto-scaling feature.

REQ	Description
[REQ-O1-NONFUN2]	ML model instances running in target ML inference hosts shall be automatically scaled by observing resource utilization in MF.

Annex A (Informative)

A.1 Discussion on A1/O1 clarification

The following table tries to summarize WG2 involved information exchange over O1 and A1 interface based on the UCR doc and AI/ML workflow discussion.

Table 3 - A1 vs O1 information exchange

Information	Interface	Management Services	Remarks

Policy	A1		
Enrichment information	A1		
Policy feedback	A1		Feedback for model state
Non-RT RIC performance data collection	O1	Performance measurement	SMO internal interface to access O1 data
Non-RT RIC Fault data collection	O1	Fault measurement	SMO internal interface to access O1 data
Network parameter configuration	O1	Provisioning management	O1-CM
AI/ML model deployment	O1	Software management	
AI/ML model update	O1	Software management	Containerized, same for xApp update/revision control as per OAM
AI/ML model performance monitoring	O1		Enhancement is needed. How to model the AI/ML in the information model needs further study.

A.2 Examples of ML model capabilities/descriptors

- ML capabilities may include performance aspects of the target network function (e.g. CPU, memory, etc.), support for ML engines, supported libraries, [i.x10, i.x9].

- These capabilities need to be matched against an ML model descriptor to decide whether a model can be deployed in the target network function.

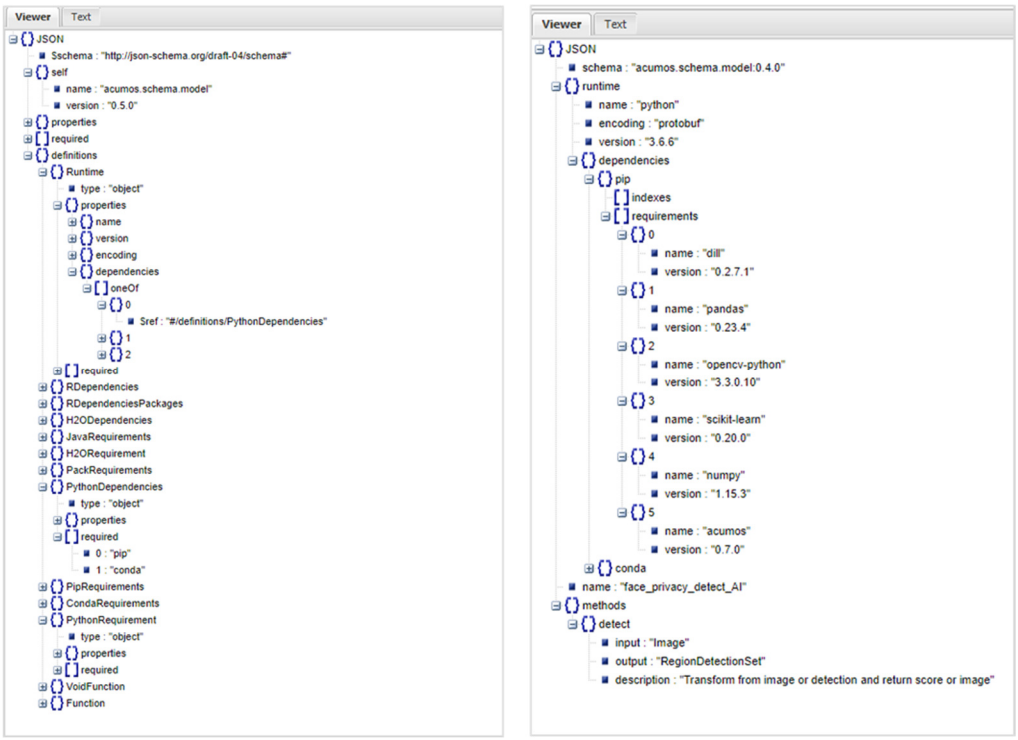

Figure 13 - Example ML model descriptor schema

Figure 14 provides an example schema for ML models and an illustration for a face_privacy_detection use case. It shows the input/output mapping and a set of ML model runtime dependencies.

Annex Z

O-RAN ADOPTER LICENSE AGREEMENT

BY DOWNLOADING, USING OR OTHERWISE ACCESSING ANY O-RAN SPECIFICATION, ADOPTER AGREES TO THE TERMS OF THIS AGREEMENT.

This O-RAN Adopter License Agreement (the "Agreement") is made by and between the O-RAN Alliance and the entity that downloads, uses or otherwise accesses any O-RAN Specification, including its Affiliates (the "Adopter").

This is a license agreement for entities who wish to adopt any O-RAN Specification.

SECTION 1: DEFINITIONS

1.1 "Affiliate" means an entity that directly or indirectly controls, is controlled by, or is under common control with another entity, so long as such control exists. For the purpose of this Section, "Control" means beneficial ownership of fifty (50%) percent or more of the voting stock or equity in an entity.

1.2 "Compliant Portion" means only those specific portions of products (hardware, software or combinations thereof) that implement any O-RAN Specification.

1.3 "Adopter(s)" means all entities, who are not Members, Contributors or Academic Contributors, including their Affiliates, who wish to download, use or otherwise access O-RAN Specifications.

1.4 "Minor Update" means an update or revision to an O-RAN Specification published by O-RAN Alliance that does not add any significant new features or functionality and remains interoperable with the prior version of an O-RAN Specification. The term "O-RAN Specifications" includes Minor Updates.

1.5 "Necessary Claims" means those claims of all present and future patents and patent applications, other than design patents and design registrations, throughout the world, which (i) are owned or otherwise licensable by a Member, Contributor or Academic Contributor during the term of its Member, Contributor or Academic Contributorship; (ii) such Member, Contributor or Academic Contributor has the right to grant a license without the payment of consideration to a third party; and (iii) are necessarily infringed by implementation of a Final Specification (without considering any Contributions not included in the Final Specification). A claim is necessarily infringed only when it is not possible on technical (but not commercial) grounds, taking into account normal technical practice and the state of the art generally available at the date any Final Specification was published by the O-RAN Alliance or the date the patent claim first came into existence, whichever last occurred, to make, sell, lease, otherwise dispose of, repair, use or operate an implementation which complies with a Final Specification without infringing that claim. For the avoidance of doubt in exceptional cases where a Final Specification can only be implemented by technical solutions, all of which infringe patent claims, all such patent claims shall be considered Necessary Claims.

1.6 "Defensive Suspension" means for the purposes of any license grant pursuant to Section 3, Member, Contributor, Academic Contributor, Adopter, or any of their Affiliates, may have the discretion to include in their license a term allowing the licensor to suspend the license against a licensee who brings a patent infringement suit against the licensing Member, Contributor, Academic Contributor, Adopter, or any of their Affiliates.

SECTION 2: COPYRIGHT LICENSE

2.1 Subject to the terms and conditions of this Agreement, O-RAN Alliance hereby grants to Adopter a non-exclusive, non-transferable, irrevocable, non-sublicensable, worldwide copyright license to obtain, use and modify O-RAN Specifications, but not to further distribute such O-RAN Specification in any modified or unmodified way, solely in furtherance of implementations of an O-RAN Specification

2.2 Adopter shall not use O-RAN Specifications except as expressly set forth in this Agreement or in a separate written agreement with O-RAN Alliance.

SECTION 3: FRAND LICENSE

3.1 Members, Contributors and Academic Contributors and their Affiliates are prepared to grant based on a separate Patent License Agreement to each Adopter under Fair, Reasonable And Non-Discriminatory (FRAND) terms and conditions with or without compensation (royalties) a nonexclusive, non-transferable, irrevocable (but subject to Defensive Suspension), non-sublicensable, worldwide license under their Necessary Claims to make, have made, use, import, offer to sell, lease, sell and otherwise distribute Compliant Portions; provided, however, that such license shall not extend: (a) to any part or function of a product in which a Compliant Portion is incorporated that is not itself part of the Compliant Portion; or (b) to any Adopter if that Adopter is not making a reciprocal grant to Members, Contributors and Academic Contributors, as set forth in Section 3.3. For the avoidance of doubt, the foregoing license includes the distribution by the Adopter's distributors and the use by the Adopter's customers of such licensed Compliant Portions.

3.2 Notwithstanding the above, if any Member, Contributor or Academic Contributor, Adopter or their Affiliates has reserved the right to charge a FRAND royalty or other fee for its license of Necessary Claims to Adopter, then Adopter is entitled to charge a FRAND royalty or other fee to such Member, Contributor or Academic Contributor, Adopter and its Affiliates for its license of Necessary Claims to its licensees.

3.3 Adopter, on behalf of itself and its Affiliates, shall be prepared to grant based on a separate Patent License Agreement to each Members, Contributors, Academic Contributors, Adopters and their Affiliates under FRAND terms and conditions with or without compensation (royalties) a nonexclusive, non-transferable, irrevocable (but subject to Defensive Suspension), non-sublicensable, worldwide license under their Necessary Claims to make, have made, use, import, offer to sell, lease, sell and otherwise distribute Compliant Portions; provided, however, that such license will not extend: (a) to any part or function of a product in which a Compliant Portion is incorporated that is not itself part of the Compliant Portion; or (b) to any Members, Contributors, Academic Contributors, Adopters and their Affiliates that is not making a reciprocal grant to Adopter, as set forth in Section 3.1. For the avoidance of doubt, the foregoing license includes the distribution by the Members', Contributors', Academic Contributors', Adopters' and their Affiliates' distributors and the use by the Members', Contributors', Academic Contributors', Adopters' and their Affiliates' customers of such licensed Compliant Portions.

SECTION 4: TERM AND TERMINATION

4.1 This Agreement shall remain in force, unless early terminated according to this Section 4.

4.2 O-RAN Alliance on behalf of its Members, Contributors and Academic Contributors may terminate this Agreement if Adopter materially breaches this Agreement and does not cure or is not capable of curing such breach within thirty (30) days after being given notice specifying the breach.

4.3 Sections 1, 3, 5 - 11 of this Agreement shall survive any termination of this Agreement. Under surviving Section 3, after termination of this Agreement, Adopter will continue to grant licenses (a) to entities who become Adopters after the date of termination; and (b) for future versions of O-RAN Specifications that are backwards compatible with the version that was current as of the date of termination.

SECTION 5: CONFIDENTIALITY

Adopter will use the same care and discretion to avoid disclosure, publication, and dissemination of O-RAN Specifications to third parties, as Adopter employs with its own confidential information, but no less than reasonable care. Any disclosure by Adopter to its Affiliates, contractors and consultants should be subject to an obligation of confidentiality at least as restrictive as those contained in this Section. The foregoing obligation shall not apply to any information which is: (1) rightfully known by Adopter without any limitation on use or disclosure prior to disclosure; (2) publicly available through no fault of Adopter; (3) rightfully received without a duty of confidentiality; (4) disclosed by O-RAN Alliance or a Member, Contributor or Academic Contributor to a third party without a duty of confidentiality on such third party; (5) independently developed by Adopter; (6) disclosed pursuant to the order of a court or other authorized governmental body, or as required by law, provided that Adopter provides reasonable prior written notice to O-RAN Alliance, and cooperates with O-RAN Alliance and/or the applicable Member, Contributor or Academic Contributor to have the opportunity to oppose any such order; or (7) disclosed by Adopter with O-RAN Alliance's prior written approval.

SECTION 6: INDEMNIFICATION

Adopter shall indemnify, defend, and hold harmless the O-RAN Alliance, its Members, Contributors or Academic Contributors, and their employees, and agents and their respective successors, heirs and assigns (the "Indemnitees"), against any liability, damage, loss, or expense (including reasonable attorneys' fees and expenses) incurred by or imposed upon any of the Indemnitees in connection with any claims, suits, investigations, actions, demands or judgments arising out of Adopter's use of the licensed O-RAN Specifications or Adopter's commercialization of products that comply with O-RAN Specifications.

SECTION 7: LIMITATIONS ON LIABILITY; NO WARRANTY

EXCEPT FOR BREACH OF CONFIDENTIALITY, ADOPTER'S BREACH OF SECTION 3, AND ADOPTER'S INDEMNIFICATION OBLIGATIONS, IN NO EVENT SHALL ANY PARTY BE LIABLE TO ANY OTHER PARTY OR THIRD PARTY FOR ANY INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES RESULTING FROM ITS PERFORMANCE OR NON-PERFORMANCE UNDER THIS AGREEMENT, IN EACH CASE WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, AND WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

O-RAN SPECIFICATIONS ARE PROVIDED "AS IS" WITH NO WARRANTIES OR CONDITIONS WHATSOEVER, WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE. THE O-RAN ALLIANCE AND THE MEMBERS, CONTRIBUTORS OR ACADEMIC CONTRIBUTORS EXPRESSLY DISCLAIM ANY WARRANTY OR CONDITION OF MERCHANTABILITY, SECURITY, SATISFACTORY QUALITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, ERROR-FREE OPERATION, OR ANY WARRANTY OR CONDITION FOR O-RAN SPECIFICATIONS.

SECTION 8: ASSIGNMENT

Adopter may not assign the Agreement or any of its rights or obligations under this Agreement or make any grants or other sublicenses to this Agreement, except as expressly authorized hereunder, without having first received the prior, written consent of the O-RAN Alliance, which consent may be withheld in O-RAN Alliance's sole discretion. O-RAN Alliance may freely assign this Agreement.

SECTION 9: THIRD-PARTY BENEFICIARY RIGHTS

Adopter acknowledges and agrees that Members, Contributors and Academic Contributors (including future Members, Contributors and Academic Contributors) are entitled to rights as a third-party beneficiary under this Agreement, including as licensees under Section 3.

SECTION 10: BINDING ON AFFILIATES

Execution of this Agreement by Adopter in its capacity as a legal entity or association constitutes that legal entity's or association's agreement that its Affiliates are likewise bound to the obligations that are applicable to Adopter hereunder and are also entitled to the benefits of the rights of Adopter hereunder.

SECTION 11: GENERAL

This Agreement is governed by the laws of Germany without regard to its conflict or choice of law provisions.

This Agreement constitutes the entire agreement between the parties as to its express subject matter and expressly supersedes and replaces any prior or contemporaneous agreements between the parties, whether written or oral, relating to the subject matter of this Agreement.

Adopter, on behalf of itself and its Affiliates, agrees to comply at all times with all applicable laws, rules and regulations with respect to its and its Affiliates' performance under this Agreement, including without limitation, export control and antitrust laws. Without limiting the generality of the foregoing, Adopter acknowledges that this Agreement prohibits any communication that would violate the antitrust laws.

By execution hereof, no form of any partnership, joint venture or other special relationship is created between Adopter, or O-RAN Alliance or its Members, Contributors or Academic Contributors. Except as expressly set forth in this Agreement, no party is authorized to make any commitment on behalf of Adopter, or O-RAN Alliance or its Members, Contributors or Academic Contributors.

In the event that any provision of this Agreement conflicts with governing law or if any provision is held to be null, void or otherwise ineffective or invalid by a court of competent jurisdiction, (i) such provisions will be deemed stricken from the contract, and (ii) the remaining terms, provisions, covenants and restrictions of this Agreement will remain in full force and effect.

Any failure by a party or third party beneficiary to insist upon or enforce performance by another party of any of the provisions of this Agreement or to exercise any rights or remedies under this Agreement or otherwise by law shall not be construed as a waiver or relinquishment to any extent of the other parties' or third party beneficiary's right to assert or rely upon any such provision, right or remedy in that or any other instance; rather the same shall be and remain in full force and effect.