Introductory Tutorial for ITU ML5G Build-a-thon

Presented at ITU FGAN 3rd Virtual meeting

Abhishek Dandekar, Fraunhofer HHI, Germany
<abhishek.girish.dandekar@hhi-extern.fraunhofer.de>

17 June 2021

Key concepts

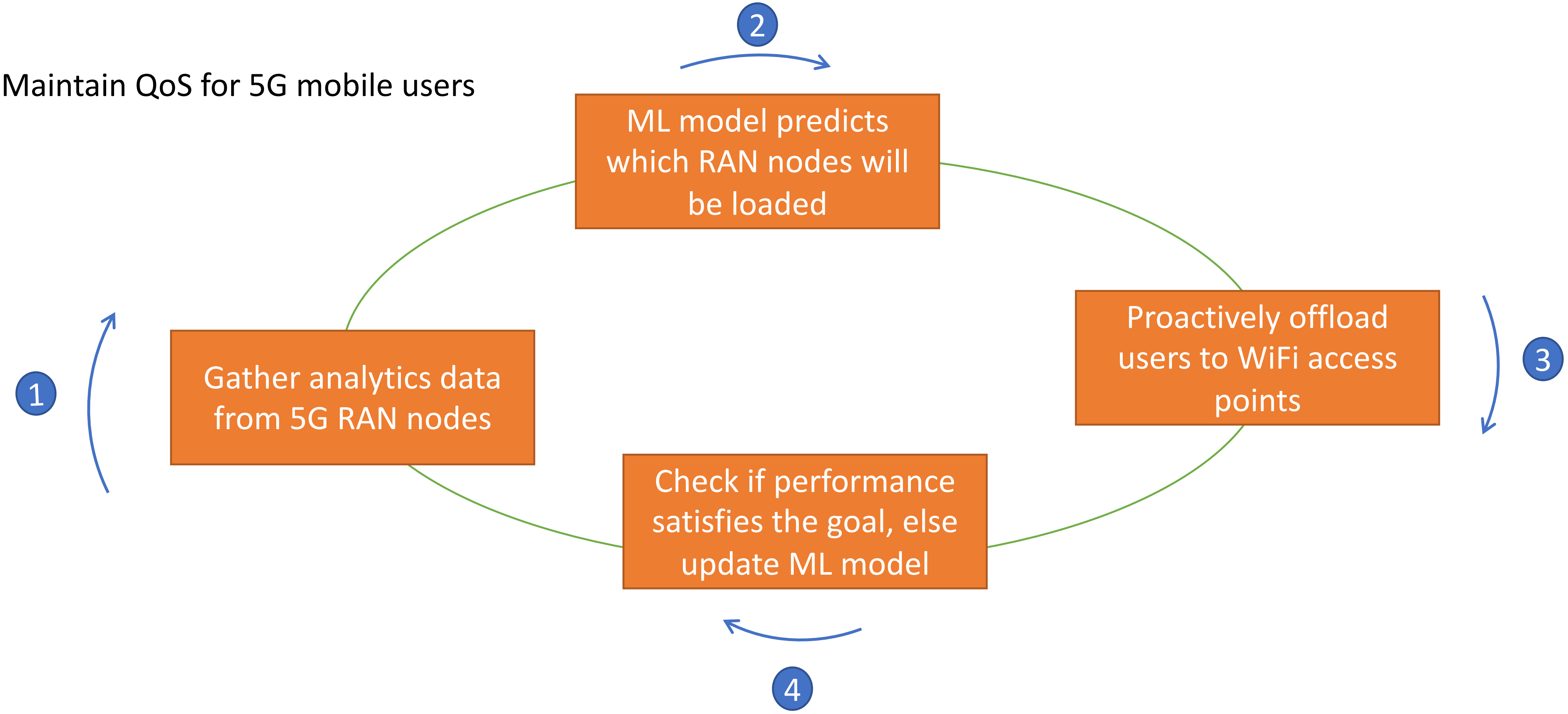
Closed loop

- A closed loop is a type of control mechanism that monitors and regulates a set of managed network entities with the objective of achieving a specific goal[1].
- Autonomous closed loops are those which are able to achieve this without any external intervention(apart from the input goal).
- Autonomous closed loops might be achieved using methods based on AI/ML models.

[1] ETSI GS ZSM 009-1 V0.10.5 (WIP)

Autonomous control loop example

Goal: Maintain QoS for 5G mobile users



Orchestration

- Orchestration is generally used to indicate management of physical or virtual network functions.
- However closed loop orchestration refers to all the activities required to manage the closed loop until it achieves the goal.
- These might include interpreting the goal, fetching data from data sources, fetching AI/ML models, function chaining, deployment of functions in the network etc.

Intent

- Intent describes a high level goal which needs to be achieved.
  - e.g. Maintain QoS for 5G mobile users
- It only describes *what* needs to be achieved rather than *how.*
- Intent can be created by a human user or machine
  - e.g. human network admin, control loop etc.
- Intent can be consumed by machine
  - e.g. closed loop orchestrator

The Challenge

Orchestration of hierarchical autonomous closed loops for emergency management in 6G networks

Description:
Future mobile networks including 6G are expected to use AI/ML based closed loops for performing various network operations. Our goal is to orchestrate a set of hierarchical autonomous AI/ML based closed loops in the mobile network. These closed loops should be able to perform operations like disaster detection, ML Pipeline management etc. Participants are expected to use relevant simulators as well as the 5G Berlin testbed for this purpose.

Participants will get opportunity to-

1. Get hands on experience with state-of-the-art 5G Berlin testbed
2. Contribute towards global telecom standardization activities in ITU
3. Win great prizes!
4. Use AI for Good! (Emergency management)

Further information:
https://github.com/ITU-build-a-thon/challenge-resources

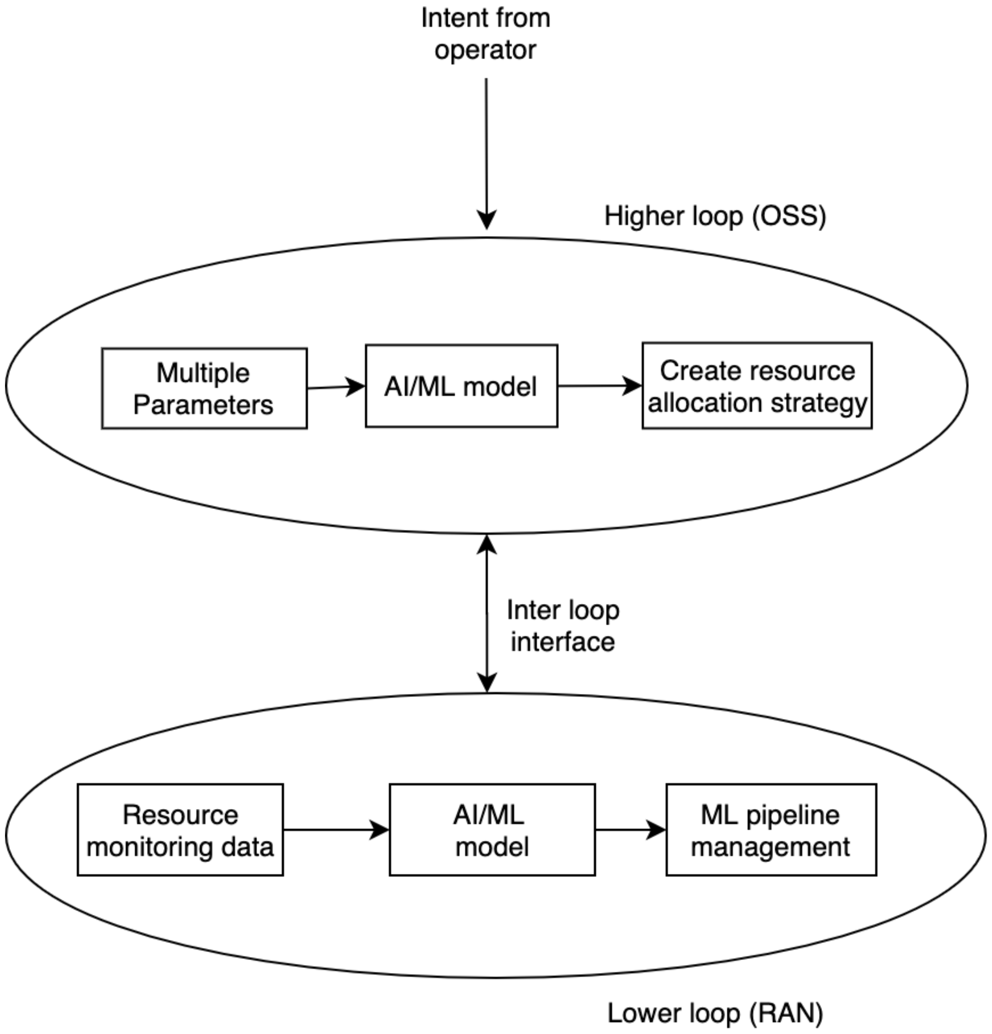
Use case details

1. A mobile operator wants to detect certain set of emergencies and provide connectivity to emergency responders according to predefined SLA. It provides these requirements to OSS via a user generated intent.

2. OSS might deploy a closed loop to detect emergencies. It might collect data from sources like network analytics data, social media scraping, input from emergency responders etc.

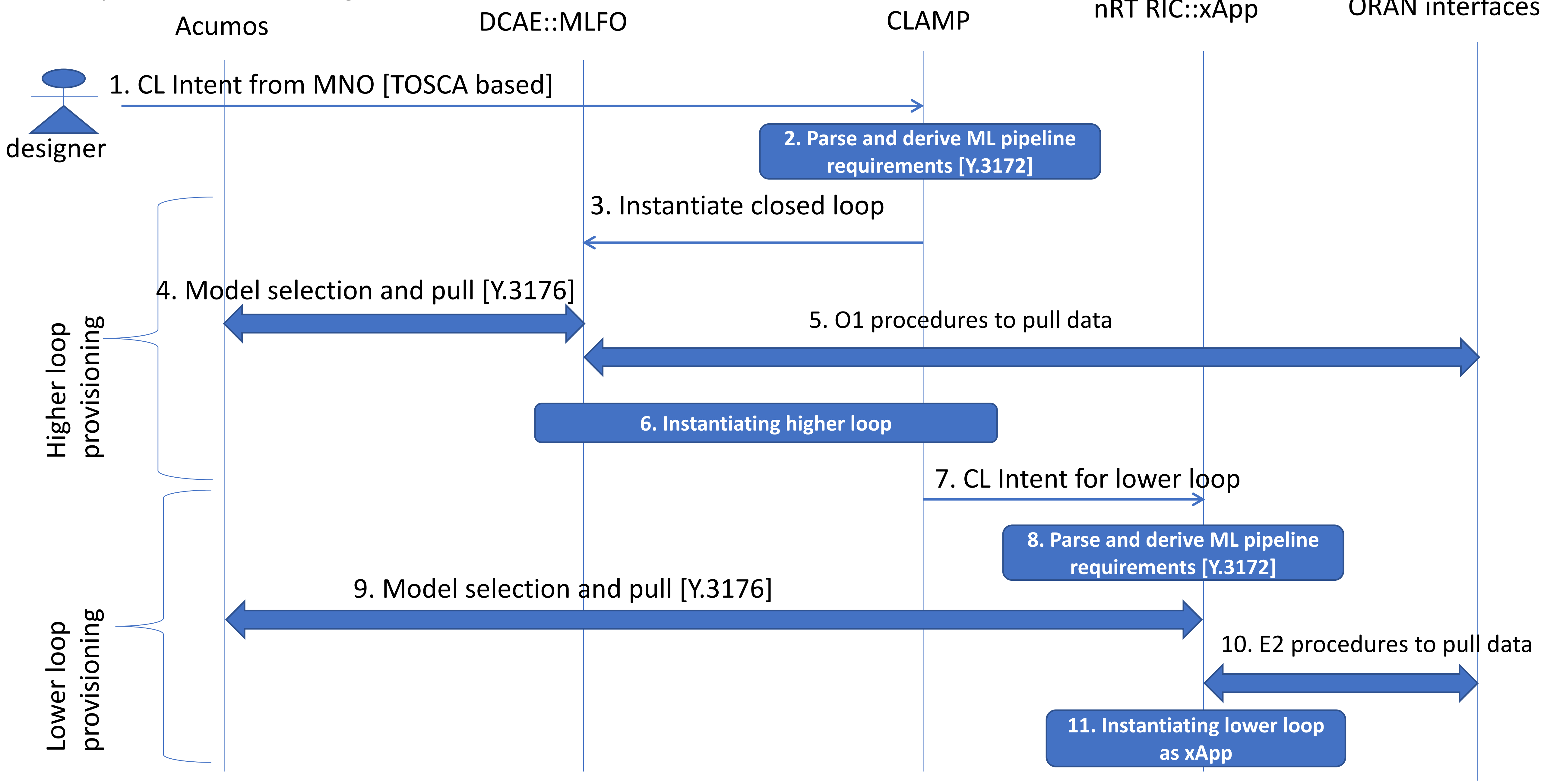
3. This closed loop might use AI/ML models to detect emergency. It creates a high level strategy to reallocate RAN resources for Emergency Responders (ER). It then applies this strategy to the lower RAN loop in form of machine generated intent over inter-loop interface.

4. RAN domain closed loop might use this high-level strategy (and other inputs) to offload inference tasks from ER devices to the edge or use split AI/ML model to run inference tasks jointly on edge and ER device. This decision might be taken based on available network and compute resources.



Ref: FGAN-I-83-R2 , FGAN-I-090

Sequence diagram



Activity 1- Intent handling

- Goal 1: Design a intent (e.g. using TOSCA)
- Goal 2: Develop a mechanism for parsing intent and deriving requirements for the autonomous system.
- Goal 3: Develop mechanism for generation of intent from given set of requirements.
- Goal 4: Integrate all of these with industry grade orchestrator (e.g. ONAP)
- Note: Activity-1 corresponding to steps 1, 2 and 7 in the sequence diagram.

Getting started (Activity 1)

- Look at TOSCA specifications[1]
- Look at how TOSCA is parsed[2]

[1] https://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.3/os/TOSCA-Simple-Profile-YAML-v1.3-os.html
[2] https://puccini.cloud/tosca/parser/

Activity 2- Control loop instantiation

- Goal 1: According to the given requirements create a pipeline or control loop in DCAE MOD or CLAMP.
- Goal 2: Fetch relevant model from Acumos and deploy the model into ONAP in DCAE. A pre-trained model might be used for this purpose.
- Note: Activity-2 corresponding to steps 3,4 in the sequence diagram.

Getting started (Activity 2)

- Look at Acumos DCAE integration[1]
- Explore DCAE MOD APIs[2]

[1] https://wiki.onap.org/display/DW/Acumos+DCAE+Integration
[2] https://docs.onap.org/projects/onap-dcaegen2/en/latest/sections/design-components/DCAE-MOD/DCAE-MOD-Architecture.html#

Activity 3- Data handling

- Goal 1: Configure sources of data over O1 interface to receive appropriate data.
- Goal 2: Receive the data over O1 interface and apply it to the deployed AI/ML model in ONAP.
- Goal 3: Demonstrate successful inference and apply it to the sink target (e.g. intent to ORAN)
- Note: Activity-3 corresponding to steps 5, 6 in the sequence diagram.

Getting started (Activity 3)

- Explore how DCAE handles data[1]
- Explore NTS simulator/O1 interface[2]
- Take a look at ORAN architecture[3]

[1] *https://docs.onap.org/projects/onap-dcaegen2/en/latest/sections/services/serviceindex.html*
[2] https://docs.o-ran-sc.org/projects/o-ran-sc-sim-o1-interface/en/latest/overview.html
[3] https://www.youtube.com/watch?v=otlUOgwitmU

Activity 4- ORAN Control loop instantiation

- Goal 1: According to the given requirements fetch model from Acumos.
- Goal 2: Deploy the model as xApp in ORAN. A pre-trained model might be used for this purpose.
- Note: Activity-4 corresponding to steps 8, 9 in the sequence diagram.

Getting started (Activity 4)

- Explore how Acumos model can be pulled and deployed based on requirements from the intent in nRT-RIC [1]

[1] https://wiki.o-ran-sc.org/display/RICA/ML-based+xApp

Activity 5- ORAN Data handling

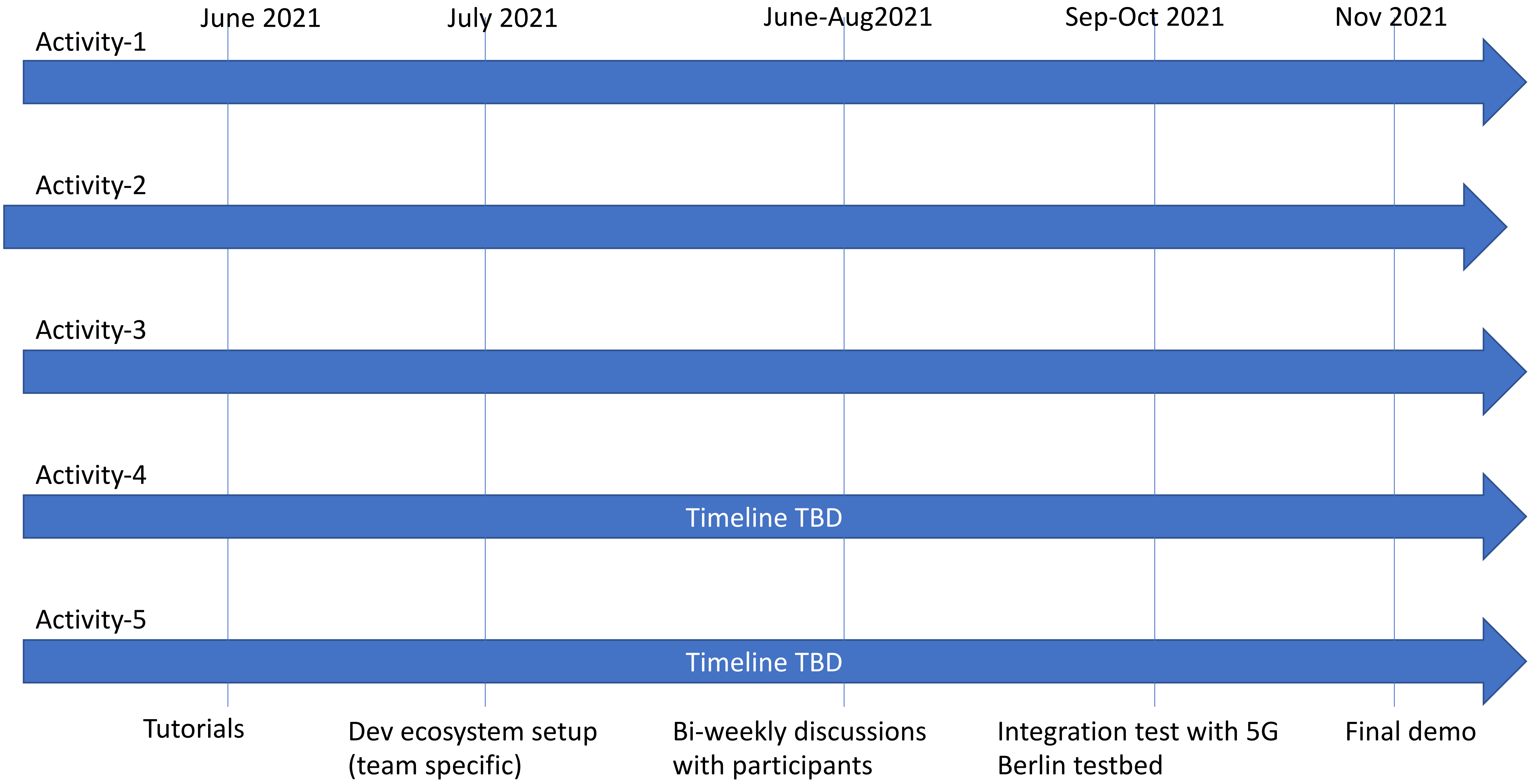
- Goal 1: Configure sources of data over E2 interface to receive appropriate data.
- Goal 2: Receive the data apply it to the deployed AI/ML model in xApp.
- Goal 3: Demonstrate successful inference and apply it to the sink target.
- Note: Activity-5 corresponding to steps 10, 11 in the sequence diagram.

Getting started (Activity 5)

- Explore E2 interface in nRT-RIC[1]
- Explore how to carry out inference in xAPP[2]

[1] https://wiki.o-ran-sc.org/display/RICP
[2] https://wiki.o-ran-sc.org/display/RICA/ML-based+xApp

Timeline


June 2021
July 2021
June-Aug2021
Sep-Oct 2021
Nov 2021
Activity-1
Activity-2
Activity-3
Activity-4
Timeline TBD
Activity-5
Timeline TBD
Tutorials
Dev ecosystem setup (team specific)
Bi-weekly discussions with participants
Integration test with 5G Berlin testbed
Final demo

Join

- Github: https://github.com/ITU-build-a-thon/challenge-resources
- Slack: https://join.slack.com/t/itu-challenge/shared_invite/zt-eql00z05-CXelo7_aL0nHGM7xDDvTmA

5G Berlin testbed details will be revealed in the next presentation….

Stay tuned!!