INTERNATIONAL TELECOMMUNICATION UNION
TELECOMMUNICATION STANDARDIZATION SECTOR
STUDY PERIOD 2022-2024

SG13-TD50/PLEN
STUDY GROUP 13
Original: English

Question(s): 20/13 Geneva, 4-15 July 2022

TD

Source: Editors

Title: Draft new Supplement 71 to ITU-T Y.3000 series (formerly Y.Supp-AN-Use Cases): "Use cases for Autonomous Networks" – for approval

Contact:	Leon Wong Rakuten Mobile	Email: leon.wong@rakuten.com
Contact:	Paul Harvey University of Glasgow	Email: paul.harvey@glasglow.ac.uk
Contact:	Hideki Yamamoto OKI	Email: yamamoto436@oki.com
Contact:	Xiaojia Song China Mobile	Email: songxiaojia@chinamobile.com
Contact:	Yongsheng Liu China Unicom	Email: liuys170@chinaunicom.cn
Contact:	Vijay Kumar Roy India	Email: vk.roy@gov.in
Contact:	Vishnu Ram OV India	Email: vishnu.n@ieee.org

Keywords: actor, artificial intelligence, autonomous networks, knowledge, component, controller, machine learning, requirement, use case

Abstract: This document is the output of the new draft Supplement 71 to Y.3000 series, Y.Supp-AN-Use Cases: "Use cases for Autonomous Networks", based on the changes agreed in Q20/13 meeting, Geneva, 4-15 July 2022 – for agreement.

NOTE - FG AN approved the technical specification (FGAN-O-013-R1) at its 5th meeting 3-5 November 2021 and sent it as the first deliverable of FG AN to SG13. SG13 (29 November – 10 December 2021, online) decided to create new work item on ITU-T Supplement Y.Supp-AN-Use Cases: "Use cases for Autonomous Networks." After that, Q20/13 Rapporteur meetings were held on 15-17 February 2022, virtual, and on 26-28 April 2022, virtual.

This document is based on SG13- TD25/WP1 of 2022-2024 study period (output of 26-28 April 2022 Q20/13 Rapporteur meeting) and according to the 4-15 July 2022 Q20/13 meeting's discussion and results on the following contributions:

No.	Source	Title	Qs	Discussion
C164	Ministry of Communications (India)	Y.Supp-AN-Use cases: Proposed updates for approval	Q20/13	Discussed and accepted with modifications (4-15 July 2022) according to the C164 proposals:

				1. update of definitions and descriptions for key concepts. 2. review and agree the use case tables. 3. review and agree the figures. 4. editorial changes.
C201	Rakuten Mobile, OKI, China Mobile, China Unicom, India	Y.Supp-AN-Use cases: Proposed updates for approval	Q20/13	Discussed and accepted in line with C164 agreements (the proposed changes are common to C164).

< See Annexure-1>

Annexure-1:

I n t e r n a t i o n a l   T e l e c o m m u n i c a t i o n   U n i o n

ITU-T Technical Specification

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE, INTERNET PROTOCOL ASPECTS, NEXT-GENERATION NETWORKS, INTERNET OF THINGS AND SMART CITIES

ITU-T Y.3000-series –
Use cases for autonomous networks

Draft new Supplement 71 to ITU-T Y-3000 series Recommendations (formerly Y.Supp-AN-Use Cases)

Use cases for autonomous networks

Summary

This Supplement discusses use cases for autonomous networks. The use cases are divided into two categories, and possible requirements, interactions among actors and possible key components are also discussed.

Various use cases are derived according to the key concepts behind autonomous networks of exploratory evolution, real-time responsive experimentation and dynamic adaptation to enable handling of hitherto unseen changes in network scenarios or inputs to reduce the human effort involved in managing the network.

Keywords

actor, artificial intelligence, autonomous networks, controller, component, knowledge, machine learning, requirement, use case

Table of Contents

Page

Page

Draft new Supplement 71 to ITU-T Y-3000 series Recommendations (formerly Y.Supp-AN-Use Cases)

Use cases for Autonomous Networks

1 Scope

This Supplement discusses use cases for autonomous networks (AN). The use cases are divided into two categories, and possible requirements, interactions among actors and possible key components are also discussed.

2 References

[ITU-T Y.3115] ITU-T Recommendation Y.3115 (2022), *AI enabled cross-domain network architectural requirements and framework for future networks including IMT-2020.*

[ITU-T Y.3172] ITU-T Recommendation Y.3172 (2019), *Architectural framework for machine learning in future networks including IMT-2020*.

[ITU-T Y.3173] ITU-T Recommendation Y.3173 (2020), *Framework for evaluating intelligence levels of future networks including IMT-2020*.

[ITU-T Y.3174] ITU-T Recommendation Y.3174 (2020), *Framework for data handling to enable machine learning in future networks including IMT-2020*.

[ITU-T Y.3176] ITU-T Recommendation Y.3176 (2020), *Machine learning marketplace integration in future networks including IMT-2020*.

[ITU-T Y.3179] ITU-T Recommendation Y.3179 (2021), *Architectural framework for machine learning model serving in future networks including IMT-2020*.

[ITU-T Y.Suppl. 55] ITU-T Supplement Y.Suppl. 55 (2019) ITU-T Y.3170-series – Machine learning in future networks including IMT-2020: use cases.

3 Terms and definitions

3.1 Terms defined elsewhere

This Supplement uses the following terms defined elsewhere:

3.1.1 network service [b-ITU-T Y.3515]**:** A collection of network functions with a well specified behaviour.

3.2 Terms defined here

This Supplement defines the following terms:

None.

4 Abbreviations and acronyms

This Supplement uses the following abbreviations and acronyms:

AI	Artificial Intelligence
AN	Autonomous Networks
CI/CD	Continuous Integration and Continuous Delivery
CLI	Command Line Interface
CN	Controller Node
DevOps	Development and Operations
ER	Emergency Response
Ev	Evolution
Ex	Experimentation
E2AP	E2 Application Protocol
GNN	Graph Neural Networks
GUI	Graphical User Interface
IDSA	Inter-domain Service Automation
KB	Knowledge Base
KPI	Key Performance Indicator
LCM	Life Cycle Management
MIMO	Multiple Input Multiple Output
ML	Machine Learning
MLFO	Machine Learning Function Orchestrator
mMTC	Massive Machine Type Communications
MNO	Mobile Network Operator
NAO	Network Application Orchestrator
NF	Network Function
NSSI	Network Slice Sub-net Instances
Op	Operational
OSS	Operational Support System
nRT RIC	Near Real Time RIC
QoE	Quality of Experience
QoS	Quality of Service
QPaaS	Quality of Experience (QoE) Prediction as-a-Service
RAN	Radio Access Network
RIC	RAN Intelligent Controller

RSRP	Reference Signal Received Power
SINR	Signal-to-Interference-plus-Noise Ratio
SLA	Service Level Agreement
TOSCA	Topology and Orchestration Specification for Cloud Applications
URLLC	Ultra-Reliable Low-Latency Communication
ZSM	Zero Touch Service Management

5 Conventions

In this Supplement, in alignment with the conventions of [ITU-T Y.Suppl. 55], possible requirements which are derived from a given use case, are classified as follows:

- The keywords "it is critical" indicate a possible requirement which would be necessary to be fulfilled (e.g., by an implementation) and enabled to provide the benefits of the use case.
- The keywords "it is expected" indicate a possible requirement which would be important but not absolutely necessary to be fulfilled (e.g., by an implementation). Thus, this possible requirement would not need to be enabled to provide complete benefits of the use case.
- The keywords "it is of added value" indicate a possible requirement which would be optional to be fulfilled (e.g., by an implementation), without implying any sense of importance regarding its fulfilment. Thus, this possible requirement would not need to be enabled to provide complete benefits of the use case.

6 Introduction

As the demand and expectation of communication networks is growing, so are user subscriptions and new service expectations. Network operators must find new ways to address these expectations while at the same time controlling operational cost. Autonomous networks are those that possess the ability to monitor, operate, recover, heal, protect, optimize, and reconfigure themselves in order to adapt to hitherto unseen changes in situations. These abilities are commonly known as the self-* properties. The impact of autonomy on the network touches many areas including planning, security, audit, inventory, optimisation, orchestration, and quality of experience. Hence, autonomous networks form an important part of future networks not only to provide value-added features to the user, but also to reduce the management overheads for the network operator. The use cases discussed in this document are based on considerations for enhancing end user experience, motivating autonomous behaviour in networks.

The main concepts behind autonomous networks which are elaborated here are exploratory evolution, real-time responsive experimentation and dynamic adaptation. To study and analyse use cases along these concepts in networks, a basic building block called “controller” is introduced. Controllers are used in the use cases to further elaborate autonomous networks and the key concepts required to enable them.

A controller is a system with workflows composed of modules, realizing or implementing specific functionalities in the network. A controller’s modules can be designed independently of the network architecture, implemented, and integrated into the network after verification.
A controller’s modules may themselves be workflows, open loops, or closed loops.

Examples of controllers are closed loops [ITU-T Y.3115] implemented as software modules that address functionalities like monitoring, analysis and optimization.

Dynamic changes in network scenarios and states may arise in the network as detected or identified via monitoring of the underlay network. Many of these situations may be hitherto unknown or unseen in the network. The concept of exploratory evolution introduces the mechanisms and processes of exploration and evolution to adapt a controller in response to changes in the underlay network. These processes generate new controllers or update (evolve) existing controllers to respond to such changes and solve the situation or task at hand more appropriately.
The process of exploratory evolution may potentially reduce the role of the human engineer by exploring alternate techniques or “creative” solutions.

Controllers may be tested using testing tools, data generators, or real data from the underlay network. Validation of controllers and their logic, using simulated and/or real data, may be done before the deployment of controllers in the network. A comparison of the results of exploratory evolution of controllers may result in the selection of the best variation of the solution or approach. This continuous process, based on monitoring and optimization of deployed controllers in the underlay network, is called real-time responsive experimentation.

Finally, the adapted and validated controllers are integrated at run-time to underlay networks. Dynamic adaptation is the final concept in equipping the network with autonomy and the ability to handle new and hitherto unseen changes in network scenarios.

With consideration of the above concepts, an autonomous network is a network which can generate, adapt, and integrate controllers at run-time using network-specific information and can realize exploratory evolution, real-time responsive experimentation and dynamic adaptation. Hence, networks which are able to handle hitherto unseen changes in network scenarios or inputs, thereby reducing the human effort involved in managing the network, are termed as “autonomous networks”. Use cases analysed in this document point to scenarios where the concepts of exploratory evolution, real-time responsive experimentation and dynamic adaptation can help realizing autonomous networks.

This Supplement provides various types of use cases. These use cases consist of those which directly describe various autonomous behaviours and those which describe the applications that benefit from autonomous behaviours. For each use case, there may be various entities which are involved in different steps of the use case. Such entities, termed here as actors, interact with each other as part of the use case scenario, contributing to use case requirements. Actors may be implemented physically or virtually in the network realising the use case. Analysis of use case requirements may suggest exemplar components which could realise the use case in the network.

NOTE 1 - The actors can be constituted of elements of different nature, including environment stakeholders such as operators and users, modules, domains and others.

This Supplement provides, where appropriate, actor-interaction diagrams and specific figures of exemplar components, which may help further analysis during later stages of standardization.

NOTE 2 – The components shown in the figures are exemplary in nature, pointing to a possible set of components which may be used in a realization of the use case. They are hereafter referred to as possible components.

The analysis of use cases includes the study of the relations to the concepts of exploratory evolution, real-time responsive experimentation and dynamic adaptation, as well as the derivation of possible requirements and their classification according to the conventions in clause 5.

NOTE 3 – From here on, for simplicity, the concepts of exploratory evolution, real-time responsive experimentation and dynamic adaptation are mentioned as, respectively, “evolution”, “experimentation” and “adaptation”.

7 Use cases and requirements

This clause describes the use cases, their requirements and, where appropriate, actor-interaction diagrams and specific figures of possible components are provided. Use cases are classified into two categories. Category 1 (cat 1) describes scenarios related to the autonomous behaviours themselves. Cat 1 use cases may have requirements related to the components enabling the key concepts of exploratory evolution, real-time responsive experimentation and dynamic adaptation. On the other hand, category 2 (cat 2) describes scenarios related to applications of autonomous behaviours in the network. They may point to the ways in which possible components in the AN interact to enable benefits to the various stakeholders like operators and end users.

7.1 Import and export of knowledge in an autonomous network

Use case id	AN-usecase-001
Use case description	To satisfy the key concepts of autonomous networks (evolution, experimentation and adaptation) while minimizing human intervention requires knowledge. This knowledge may include representation of data about the environment in which the autonomous system is operating, possible actions and consequences, key configuration options, possible measurement parameters and other elements of logic. This use case describes the scenarios where knowledge is accessed and used by the actors involved in the AN to realize the use case. General use case scenarios comprise of the following steps: 1. Knowledge is imported from outside or peer entities of the AN components 2. Knowledge is referred internally in the AN components, e.g., for driving evolution, driving exploration, configuration of automation loops. 3. Generate report for human consumption 4. Knowledge is stored and updated within the AN components 5. Knowledge is exported from the AN components to outside or peer entities.
Use case category	Cat 1: describes a scenario related to core autonomous behaviour itself.
Reference	[b-Clark], [b-AN2020], [b-Jimenez-Ruiz], [b-Myklebust], [b-Turing]

7.1.1 Use case requirements

Critical requirements

● AN-UC01-REQ-001: It is critical that AN enable the exchange of knowledge between the different involved AN components.

● AN-UC01-REQ-002: It is critical that AN enable the optimization of knowledge bases.

NOTE 1 – Examples of optimizations applied on the knowledge bases are access policies, granularity of storage, interconnection between various knowledge bases and relation between problems and solutions, addition of new knowledge.

● AN-UC01-REQ-003: It is critical that AN enable the creation of reports on the use of knowledge bases, for consumption by humans and machines.

NOTE 2 – Example of contents of reports are statistics on access of knowledge bases by various AN components as well as network services in the same or different administrative domains.

- AN-UC01-REQ-004: It is critical that AN enable the exchange of knowledge between the involved AN components and other entities in the same administrative domain.

NOTE 3 – Other entities include AN components and network services.

- AN-UC01-REQ-005: It is critical that AN use knowledge base(s) for mapping high level use case description(s) to controller specification.

NOTE 4 – Controller specification may use languages such as TOSCA [b-TOSCA-Simple], whereas use case descriptions may be unstructured. The high-level use case description is to be converted to a structured controller specification. In this process of "conversion", it may utilize the help of humans (using GUIs) who can better understand unstructured information, and/or automated generation techniques.

Expected requirements

- AN-UC01-REQ-006: It is expected that AN enable the exchange of knowledge between AN components and other entities in different administrative domains.

NOTE 5 – The entities in other administrative domains include AN components and network services.

Added value requirements

- AN-UC01-REQ-007: It is of added value that automated generation techniques are used by AN to produce controller specifications, using the stored controller descriptions and the knowledge base.

NOTE 6 – E.g., graph neural network (GNN)-based recommendation engine may be used to help automatic generation techniques.

7.1.2 Actor interactions and possible components

In line with the use case description, figure 1 illustrates the interactions between various actors for the access and use of knowledge in an autonomous network.

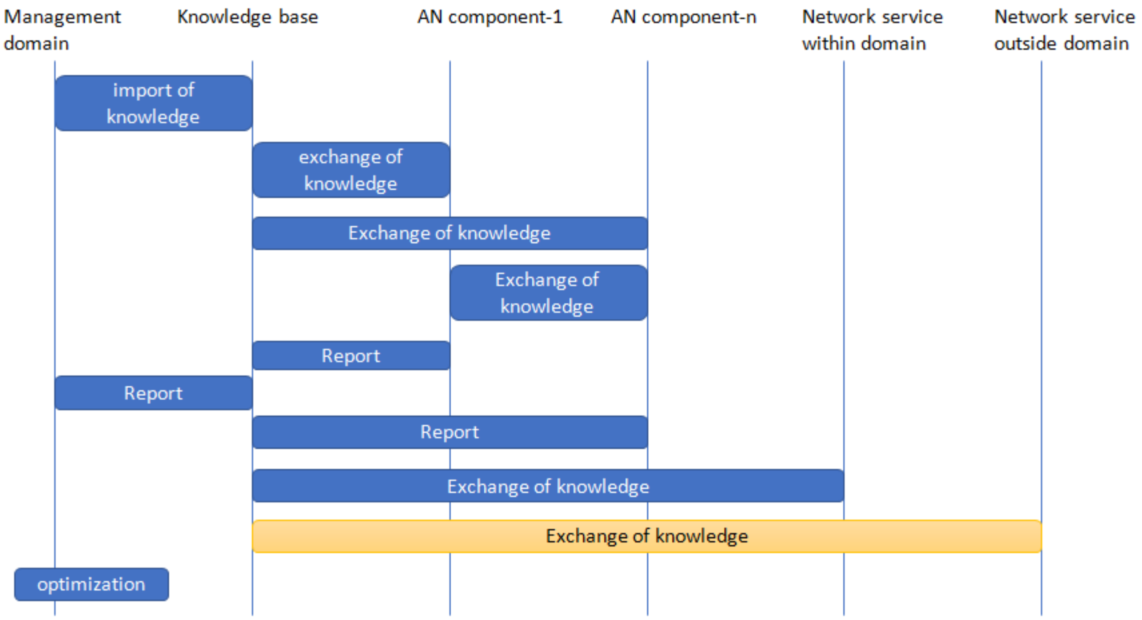


Figure 1: Actor interactions for access and use of knowledge in an autonomous network

NOTE 1 – Multiple AN components (1 to n) may be present in the system.

NOTE 2 – Actor interactions captured in figure 1 may not reflect the strict time sequence of activities.

Figure 1 shows the possible interactions between various actors enabling this use case. However, while instantiating this use case in the network, various components may be necessary. Figure 2 shows a possible set of components related to the use case. The possible functions and services exposed are depicted using connectors in the figure. In an implementation instance, these could be either directly connected or accessed via service-based architectures. The exact nature, requirements and implementations related to the components and their interactions are for future study and not in the scope of this document.

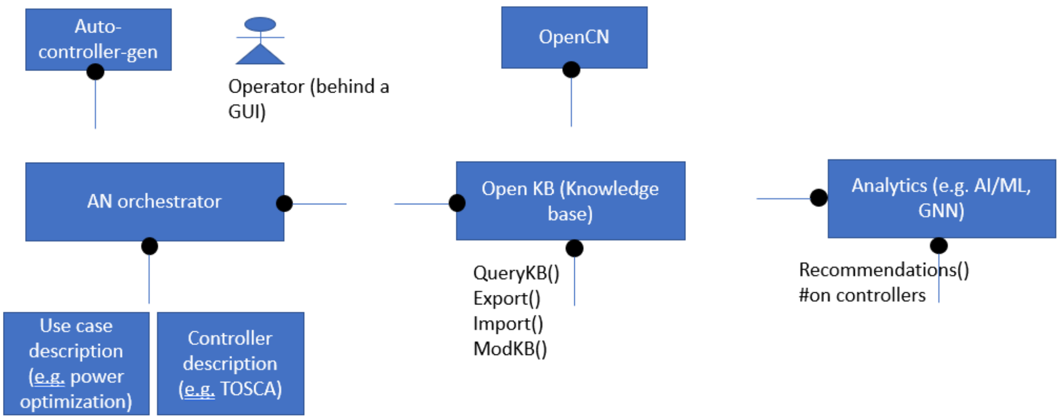


Figure 2: Possible components related to the access and use of knowledge in autonomous network

Actors involved (figure 1):

- Management domain: this actor includes entities in the network which configures, monitors and otherwise manages the AN components in the network.
- Knowledge base: this actor stores knowledge which may be queried, stored and updated by various AN components. This may be implemented in the form of an open knowledge base.
- Network services within domain: these actors are network services running within the same administrative domain of the AN. Similarly, there may be network services running outside the administrative domain of the AN.

Possible key components involved (figure 2):

- AN orchestrator: this component manages the other AN components in the network.
- Auto controller generator: this component may generate controllers.
- OpenCN: this component provides a repository of controllers.
- OpenKB: this component provides a repository to stores knowledge.
- Analytics: this component provides analytics services (including prediction, inferences) for the AN.

7.2 Configuring and driving simulators from autonomous components in the network

Use case id	FG-AN-usecase-002
Use case description	To explore and experiment with various autonomous behaviour scenarios, the AN component requires access to simulators. Simulators help evaluate the outcome of possible options without potential adverse fallouts in the real network. Long term study of simulation results is common by human researchers to understand the evolutionary needs of the network too. In this respect, the AN components need to interface with, configure and drive the different simulators. Following are related steps in this use case scenario: 1. The AN components decide the autonomous behaviour scenarios for exploration and experimentation. 2. The AN components interact with the AN sandbox to configure specific simulators which can perform the required experimentation. 3. The AN sandbox monitors the simulators and reports the completion of simulations. 4. The results are analysed by the AN components and further actions (like updating the knowledge base) are taken.
Use case category	Cat 1: describes a scenario related to core autonomous behaviour itself.
Reference	[ITU-T Y.3172]

7.2.1 Use case requirements

<u>Critical requirements</u>

● AN-UC02-REQ-001: It is critical that AN components arrive at autonomous behaviour scenarios, potentially usable for exploration and experimentation.

NOTE 1 – AN components may independently arrive at different autonomous behaviour scenarios usable for exploration and experimentation based on several factors like the functionalities they implement, current status of their knowledge, etc. For example, AN components may arrive at candidate strategies to be used for exploration and experimentation for access control. These strategies may be based on game theory approaches or combinatorial optimization approaches.

- AN-UC02-REQ-002: It is critical that AN components trigger experimentation in the AN sandbox.

NOTE 2 – AN components may independently trigger experimentation by configuring simulators in the AN sandbox.

- AN-UC02-REQ-003: It is critical that the AN sandbox collates, aggregates triggers for experimentation to form a coherent, experimentation pipeline, the execution of which is monitored and reported by the AN sandbox to the AN components.
- AN-UC02-REQ-004: It is critical that AN components analyse the reports from the AN sandbox while considering the steps in AN behaviour.

NOTE 3 – The steps in AN behaviour which depends on the analysis of reports from the AN sandbox include steps in evolution and update of knowledge.

7.2.2 Actor interactions and possible components

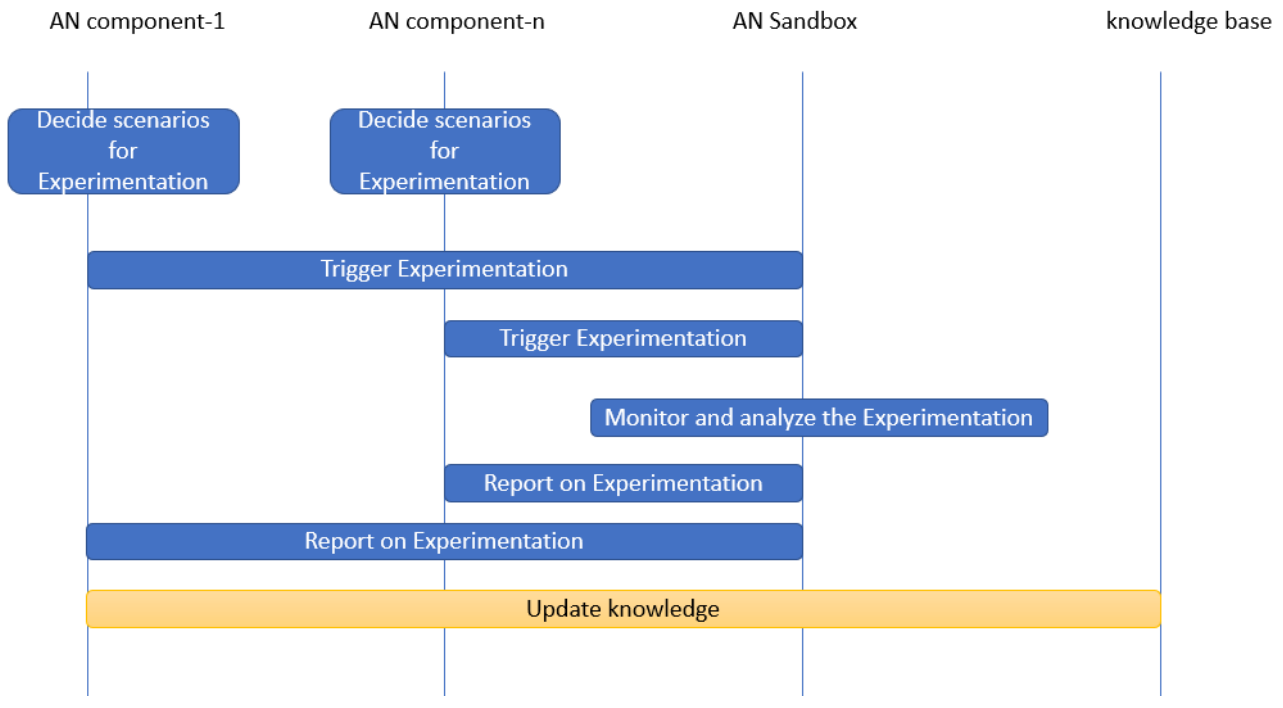


Figure 3: Actor interactions for configuring and driving simulators from autonomous components in the network

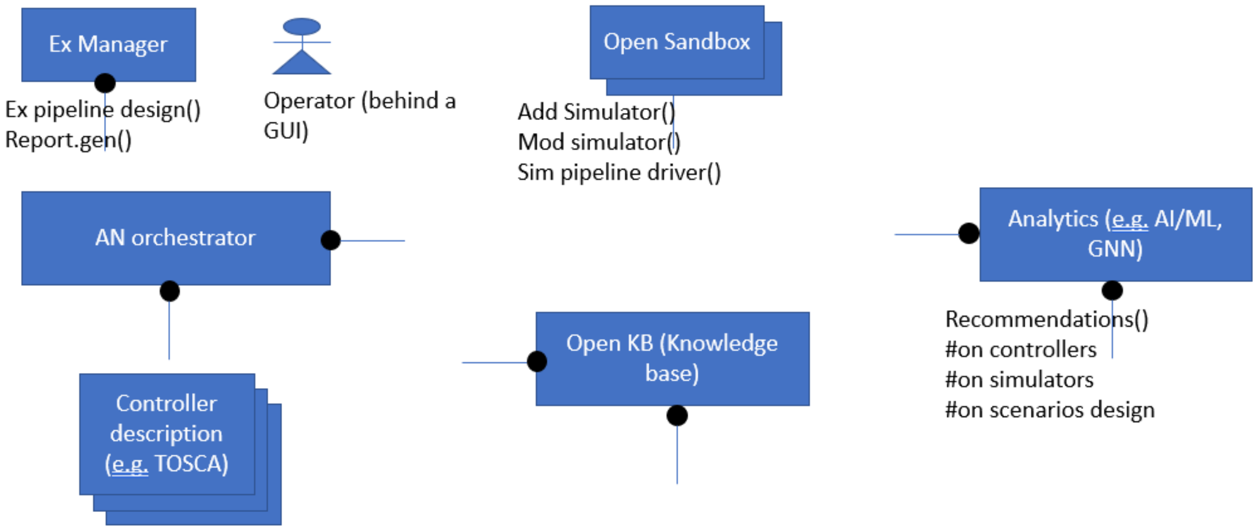


Figure 4: Possible components related to configuring and driving simulators from autonomous components in the network

Actors involved (figure 3):

- AN Sandbox: this actor provides an experimentation platform for controllers. This allows testing and verification of controllers. This may be implemented in the form of an openly accessible sandbox.
- Knowledge base: this actor stores knowledge which may be queried, stored and updated by various AN components. This may be implemented in the form of an open knowledge base.

Possible key components involved (figure 4):

- AN orchestrator: This component manages the other AN components in the network.
- Experimentation manager: This component which manages the experimentation stage of controllers, including testing and verification.
- Analytics: This component provides analytics services (including prediction, inferences) for the AN.
- OpenKB: This component provides a repository to stores knowledge.
- Open Sandbox: This component provides an experimentation platform that allows testing and verification of controllers

7.3 Peer-in-loop (including humans)

Use case id	FG-AN-usecase-003
Use case description	To guide the autonomous behaviour, AN components require access to peers. Peers include humans and other autonomous entities. Exchange of information with peers help in taking better decisions. In this respect, AN components need to interface with, exchange information with various other AN components and humans. Following are related steps in this use case scenario:

	1. The AN components decide to take guidance from other autonomous entities (peers like humans). 2. A message exchange with the peer is initiated. 3. The results of the exchange are analysed by the AN components and further actions (like updating knowledge base) are taken.
Use case category	Cat 1: describes a scenario related to core autonomous behaviour itself.
Reference	[Y.3172]

7.3.1 Use case requirements

<u>Critical requirements</u>

- AN-UC03-REQ-001: It is critical that AN components enable synchronous or asynchronous, interoperable exchange of feedback or information from peers regarding the decisions and choices related to the AN behaviour.

NOTE 1 – Peers may include humans and machines.

NOTE 2 – Feedback may include exchange of information regarding the AN behaviour such as evolution, experimentation and adaptation. Contents of the information exchanged may include capabilities and status of AN components, e.g., knowledge base, orchestration and simulators. The format used for information exchange is for further study.

7.3.2 Actor interactions and possible components

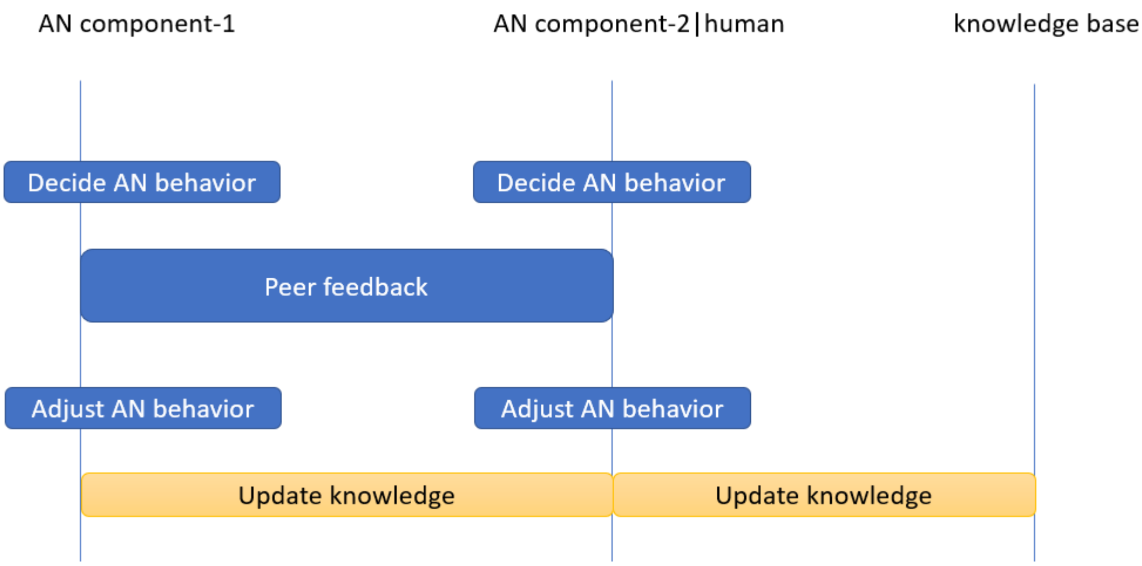


Figure 5: Actor interactions for Peer-in-loop (including humans)

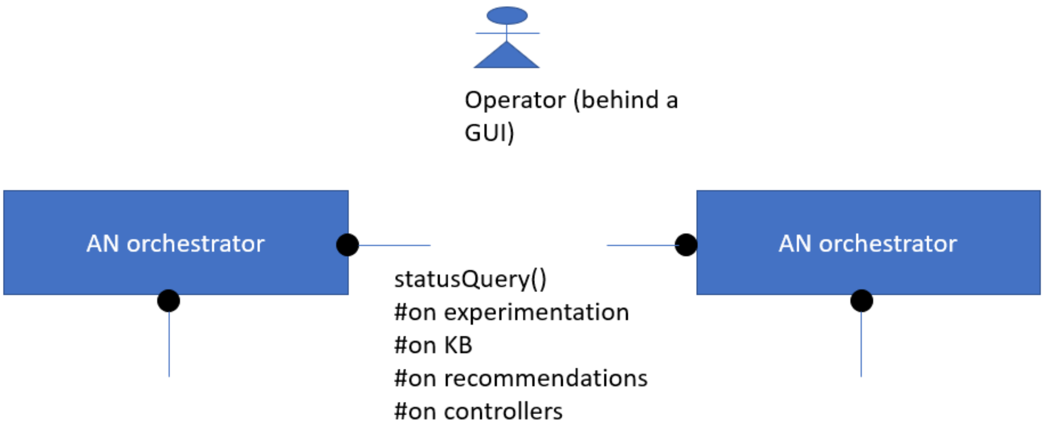


Figure 6: Possible components related to peer-in-loop (including humans)

Actors involved (figure 5):

Knowledge base: this actor stores knowledge which may be queried, stored and updated by various AN components. This may be implemented in the form of an open knowledge base.

Possible key components involved (figure 6):

AN orchestrator: This component manages the other AN components in the network.

7.4 Configuring and driving automation loops from autonomous components in the network

Use case id	FG-AN-usecase-004
Use case description	There are different automation loops in various domains of the network, already proposed by different standards bodies and industry bodies. To reflect the decisions of autonomous behaviour in the network, the AN components require access to automation loops. Automation loops help implement the decisions taken by the AN components in the network. Moreover, it is possible that automation loops provide valuable inputs for AN components to be considered for say further experimentation. In this respect, the AN components need to interface with, configure and drive the different automation loops. Following are related steps in this use case scenario: 1. The AN components decide the configurations of automation loops. 2. The AN components interact with the automation loops to configure specific scenarios which can perform the required automation. 3. The automation loops monitor the automation and reports the status of automation. 4. The results are analysed by the AN components and further actions (like updating knowledge base) are taken. NOTE- An example of configuring automation loops is, to select and provision the type of ML model to be used for domain specific analytics.
Use case category	Cat 1: describes a scenario related to core autonomous behaviour itself.
Reference	[ITU-T Y.3173]

7.4.1 Use case requirements

Critical requirements

- AN-UC04-REQ-001: It is critical that AN components decide the type of closed loops and manage the closed loops.

NOTE 1 – AN components may decide the type and structure of closed loops based on their analysis of reports, monitoring and other information exchanges. Management of closed loops may include instantiating, deletion, updating, and other operations on closed loops.

- AN-UC04-REQ-002: It is critical that AN components consider the capability and flexibility offered by closed loops to configure them to perform specific automation tasks.

- AN-UC04-REQ-003: It is critical that closed loops monitor the specific parameters of automation tasks and report them to the AN components.

NOTE 2 – Specific parameters of automation tasks may include data input to automation, analytics used in the closed loop, actions taken as part of automation, failures, error logs.

- AN-UC04-REQ-004: It is critical that AN components consider the reports from closed loops while deciding the AN behaviour.

NOTE 3 – Examples of AN behaviour are evolution, experimentation and adaptation.

7.4.2 Actor interactions and possible components

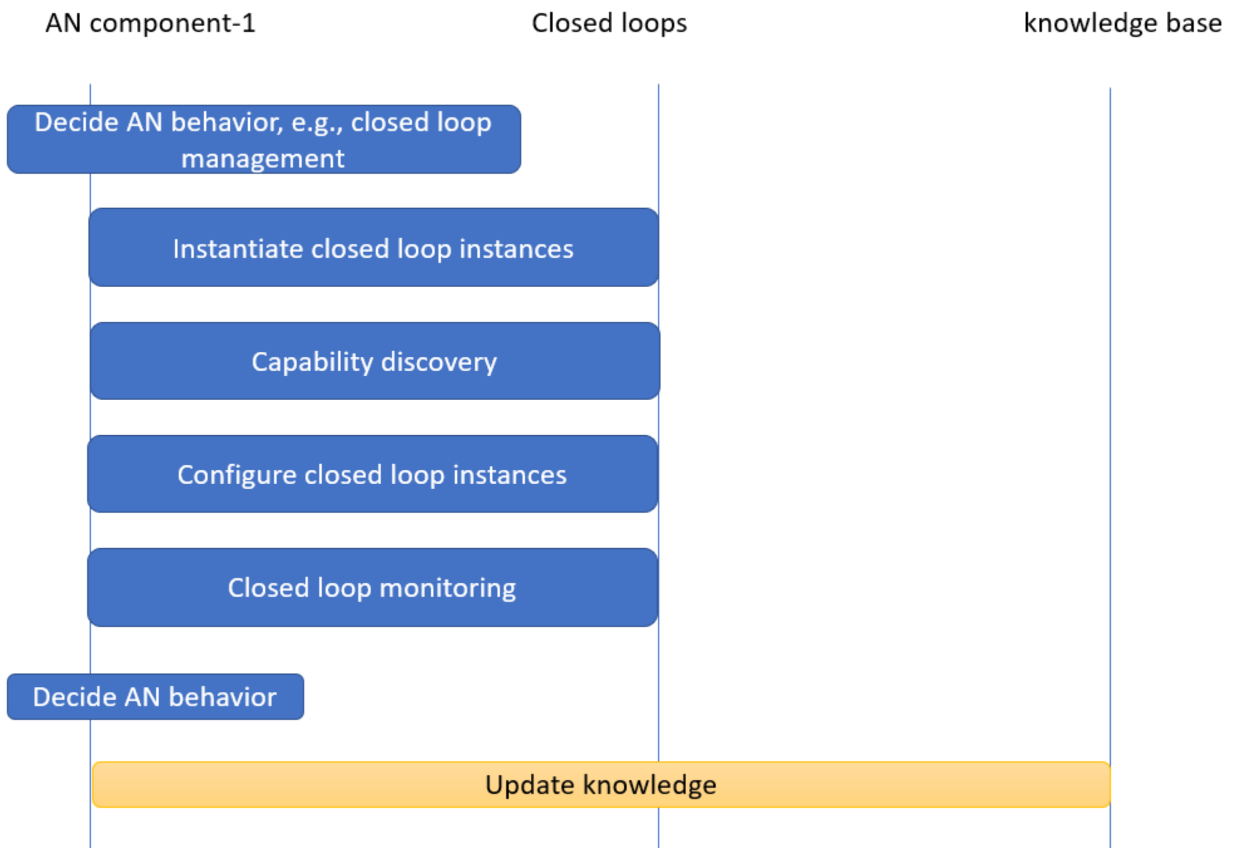


Figure 7: Actor interactions for configuring and driving automation loops from autonomous components in the network

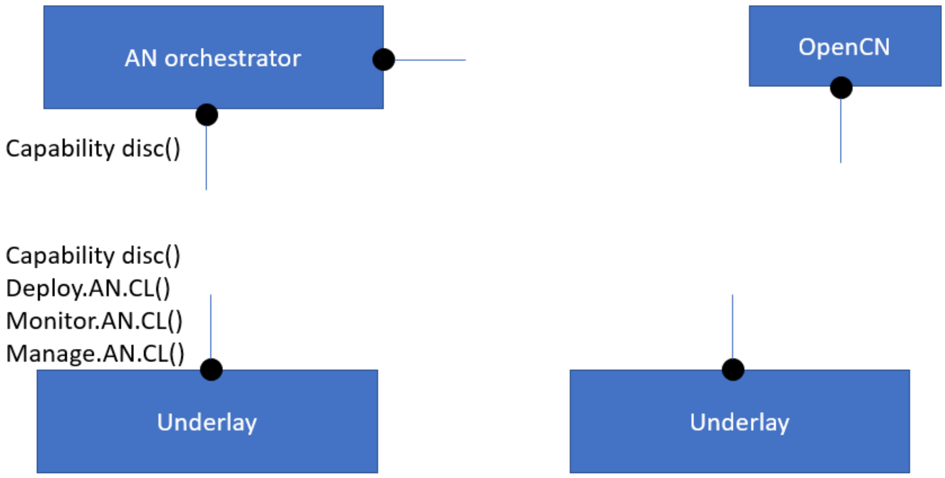


Figure 8: Possible components related to configuring and driving automation loops from autonomous components in the network

Actors involved (figure 7):

Knowledge base: This actor stores knowledge which may be queried, stored and updated by various AN components. This may be implemented in the form of an open knowledge base.

Possible key components involved (figure 8):

- AN orchestrator: This component manages the other AN components in the network.
- OpenCN: This component provides a repository of controllers.

7.5 Domain analytics services for E2E service management

Use case id	FG-AN-usecase-005
Use case description	Interoperability with components of external frameworks which provide domain analytics and predictions could assist autonomous networks in rapidly integrating with existing network components and reuse functionality which already exists in the network. This use case introduces the scenario where there are already such frameworks deployed in the network and autonomous network components are required to consume services provided by such frameworks. NOTE - For example, [b-ETSI GS ZSM 002] (section 6.5.3.2) describes the domain analytics services which provide domain-specific insights and generate domain-specific predictions based on data collected by domain data collection services and other data. Following are related steps in this use case scenario taking ZSM [b-ETSI GS ZSM 002] as reference of external framework. The AN components act as a ZSM service consumer. 1. ZSM framework components act as provider for closed loop (CL) management and other domain and cross domain services (including analytics) to the AN components. 2. Discovery of ZSM services is done by the AN components. 3. Each ZSM service performs the E2E service management based on the interaction with the AN components.

	a. Examples of interactions are: managing subscriptions, configuring analytics, request analysis results, etc. b. Other examples in the context of Zero Touch provisioning, workflow-based automation, provisioning and management of workflows are mentioned in clause 7.5 in [b-FGAN-O-013-R1].
Use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.
Reference	[b-ETSI GS ZSM 002]

7.5.1 Use case requirements

Critical requirements

- AN-UC05-REQ-001: It is critical that AN support discovery and consumption of the services provided by different types of closed loop service automation frameworks.

NOTE – Examples of different types of closed loop service automation frameworks are ETSI ZSM [b-ETSI GS ZSM 002] and FRINX machine [b-FGAN-O-013-R1]. Examples of actions taken by ANs after the consumption of services provided by closed loop service automation frameworks are managing subscriptions, configuring analytics, request analysis results, etc.

7.5.2 Actor interactions and possible components

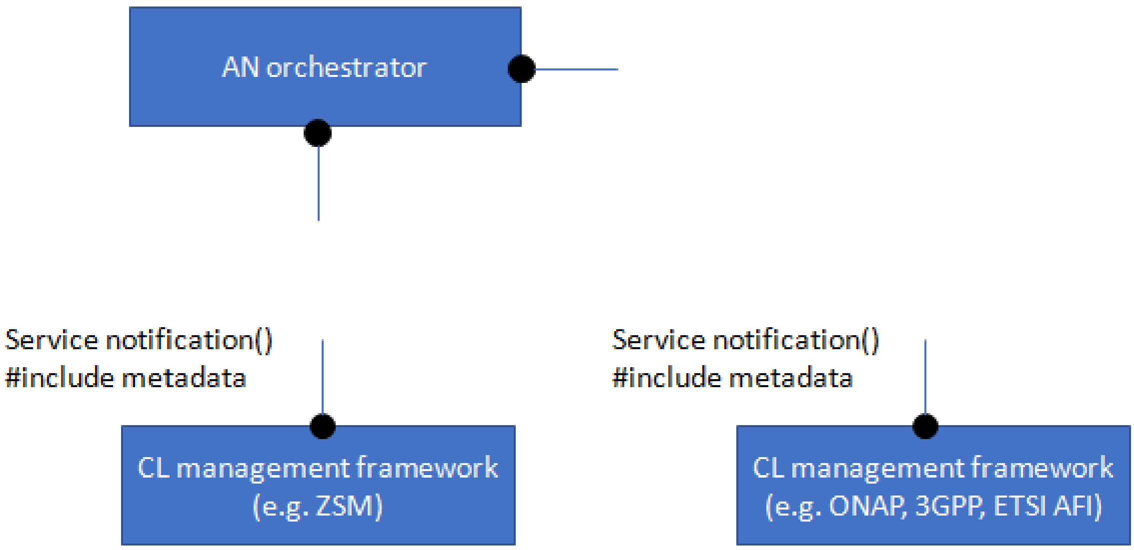


Figure 9: Possible components related to domain analytics services for E2E service management

Possible key components involved (figure 9):

- AN orchestrator: This component manages the other AN components in the network.
- CL management frameworks: Closed loop management frameworks e.g. provided by ETSI ZSM, manages closed loops in networks.

7.6 Automation and intelligent operation, maintenance and management (OAM) of radio network

Use case id	FG-AN-usecase-006

Description	Dynamic radio environment, network structure, user behaviour, and user distribution drive the network needs to be monitored and optimised continually. Currently, this consumes a lot of experts' time and labour work to discover problems, analyse root causes, and then formulating solutions for the radio networks. Therefore, AI and big data technology is necessary to achieve full process automation and intelligent management of wireless network. Some examples of autonomous management, applicable to OAM of wireless network are real-time monitoring of data quality, quasi-real-time diagnosis, root cause analysis, recommended solutions, and evaluation of impacts of recommended solutions in the radio network. ● Real-time monitoring of data quality There is a need to collect real-time data from the IMT-advanced and IMT-2020 integrated network management, then to compare the consistency of the number of network elements from the collected data, and to achieve data quality monitoring and warning through a visualization panel. ● Quasi-real-time diagnosis of abnormal condition in cells Using several categories of KPI performance indicators, an ML algorithm, e.g., Support Vector Machine (SVM)[b-Cortes], is used to diagnose network elements in these categories of scenarios such as residential and colleges on a daily/weekly basis, and distribute them to frontline managers. ● Root cause analysis and recommended solutions Through collecting tens of thousands of expert experiences, radio network knowledge bases have been established through knowledge graphs to develop intelligent recommendation algorithms and search engines, and to directly provide reasons and recommend solutions foreach network element with abnormal condition to first-line experts, thus reducing troubleshooting time and error rate. ● Evaluation of processing effects Through a mature evaluation system, the effectiveness of the solution to each abnormal condition is evaluated after the implementation, and then the knowledge base and recommendation algorithm are optimized, and the intelligence level of the entire process is continuously improved.
Category	Category 1 - Use case for autonomous behaviour
Reference	

7.6.1 Use case requirements

<u>Critical requirements</u>

● AN-UC06-REQ-001: It is critical that AN enable the discovery of problems in underlay networks, the analysis of root cause, and the formulation of solutions.

NOTE 1 – Examples of autonomous management, applicable to OAM of wireless network are real-time monitoring of data quality, quasi-real-time diagnosis, root cause analysis, recommendation of solutions, and evaluation of impacts of recommended solutions in the radio network.

● AN-UC06-REQ-002: It is critical that AN enable data quality, monitoring and visualization.

NOTE 2 – Data quality may need real-time monitoring, evaluation with respect to knowledge base, and reporting may be done using an online GUI or a report to human. Data quality may be useful to analytics services.

- AN-UC06-REQ-003: It is critical that AN enable the capturing and use of the knowledge from domain experts including use of AI/ML mechanisms for recommendation of solution based on root cause analysis.

NOTE 3 – Example of representation formats of knowledge is knowledge graphs.

Expected requirements

- AN-UC06-REQ-004: It is expected that AN use AI and big data technologies to achieve full process automation and intelligent management of networks.

Added value requirements

- AN-UC06-REQ-005: It is of added value that AN monitor varied sets of KPIs to identify faults.

- AN-UC06-REQ-006: It is of added value that ANolutions may be monitored, optimized and continuously improved.

NOTE 4 – The OpenKB [b-FGAN-O-013-R1] and recommendation algorithms are examples of AN solutions that may be optimized.

7.6.2 Actor interactions and possible components

None.

7.7 Intelligent energy saving for data centres

Use case id	FG-AN-usecase-007
Description	The rapid growth of mobile Internet, cloud computing and other business drives the need of large-scale data centres. Data centres consume large amounts of energy to run and maintain their cooling system and facilities, servers and other devices. Traditional methods cannot efficiently reduce the energy costs of data centres. Therefore, AI mechanisms are introduced to analyse the monitoring data and adjust the configurations automatically. Intelligent energy saving solution include a series of autonomous behaviour, such as automatic data acquisition, AI-based energy consumption modelling and inference, facilities parameters control policies decision, facilities adjustment actions implementation, energy saving result evaluation and control policies continuous optimization.
Category	Category 2 - Application of autonomous behaviour
Reference	None

7.7.1 Use case requirements

Critical requirements

- AN-UC07-REQ-001: It is critical that AN support data acquisition, representation, analysis of collected data and adaptation of configurations in underlay networks such as data centres.

NOTE 1 – Data acquisition and data representations may use industry standards. Analysis may use ML techniques. Adaptations may use underlay network’s specific APIs. Adaptations may be arrived at using controllers or workflows or closed loops.

Expected requirements

- AN-UC07-REQ-002: It is expected that AN support representation, autonomous analysis and continuous optimization of policies.

NOTE 2 – Policies may be related to domain specific workflows and decisions e.g. energy usage in data centres.

7.7.2 Actor interactions and possible components

None.

7.8 Autonomous massive MIMO

Use case id	FG-AN-usecase-008
Use case description	Massive MIMO is a key technology in IMT-2020, which can effectively improve the vertical coverage and system capacity in complex scenarios by using large-scale antenna array and three-dimensional beam-forming. Compared with traditional antenna, there are more dimensions of parameters to adjust for massive MIMO large-scale antenna array, including horizontal lobe width, vertical lobe width, azimuth, dip angle and beam number. Each dimension can be fine adjusted by setting a reasonable step size and theoretically and there may be tens of thousands possible combination of antenna parameter weights in a cell. Therefore, manual optimization and adjustment based on scenario/service changes can be very hard in consideration of multi-cell coordination. The autonomous massive MIMO use case is about helping operators quickly converge and achieve optimal adjustment of antenna parameters with AI capabilities of multi-dimensional analysis and prediction. The general workflow is: the IMT-2020base station collects position information from UE and sends it to the network management system, which then calculates the distribution of UE and finds the optimal weight combination with ML algorithms based on the target RSRP/SINR distribution in the current scenario, so as to maximize the utilization of system capacity and guarantee the user experience.
Use case category	Cat 2: application of autonomous behaviour
Reference	

7.8.1 Use case requirements

Critical requirements

- AN-UC08-REQ-001: It is critical that AN support identification of parameters which can be optimized, including ML parameters, based on the use case.

NOTE 1 – Example of use case is parameter optimization for massive MIMO large-scale antenna array, including horizontal lobe width, vertical lobe width, azimuth, dip angle and beam number.

- AN-UC08-REQ-002: It is critical that AN support identification of data which can be collected to analyse and infer, based on the use case.

NOTE 2 – Examples of data are distribution of UE, the target RSRP/SINR distribution in the current scenario.

- AN-UC08-REQ-003: It is critical that AN support identification of KPIs which need to be optimized.

NOTE 3 – Examples of KPIs are system capacity and QoE.

- AN-UC08-REQ-004: It is critical that AN support optimization of KPIs in distributed deployments which require multicell coordination.

7.8.2 Actor interactions and possible components

None.

7.9 Network resource allocation for emergency management based on closed loop analysis

Use case id	FG-AN-usecase-9
Use case description	Telecommunication systems are a critical pillar of emergency management. A set of hierarchical AI/ML based closed loops could be used to intelligently deploy and manage slice for emergency responders in the affected area. A higher closed loop in the OSS can be used for detecting which area is affected by the emergency and deploy a slice for emergency responders to that area. It can then set a resource arbitration policy for the lower closed loop in RAN. The lower loop can use this policy to intelligently share RAN resources between the public and emergency responder slice. It can also intelligently manage ML pipelines across the edge and emergency responder devices by using split AI/ML models or offloading of inference tasks from the devices to the edge. NOTE 1 - An instance of open RAN architecture is explained in [b-O-RAN.WG1.O-RAN-Arch], including near-RT RICs and non-RT RICs which host applications designed to run with different latency requirements (e.g. xApps and rApps). The applications may be implemented independent of the RIC implementations and may be provided by any third party. Following are related steps in this use case scenario: 1. The mobile network operator (MNO) may instruct the OSS to detect certain set of emergencies and provide connectivity to emergency responders according to predefined SLA. NOTE 2 - For example, this input may be provided using an operator intent. 2. The OSS might deploy a closed loop to achieve this. It might collect data from sources like network analytics data, social media scraping, input from emergency responders etc. NOTE 3 - For example, such inputs may be provided from nRT-RICs or other NFs in the network.

	3. The OSS might use AI/ML models to detect emergency and deploy an emergency response (ER) slice to the location. It might also create high level strategy/policy to reallocate resources among the slices. NOTE 4 - Such closed loops may be hosted in non-RT RIC and may be used for predictive resource allocations to specific edge locations based on predicted needs, in turn based on detected emergency. NOTE 5 - The policy to reallocate resources may depend, among other things, on the type of emergency e.g. a natural disaster, earthquake, a law and order situation, traffic accidents, etc. 4. The RAN domain might use this high-level strategy/policy and possibly other inputs from emergency responders to create a closed loop to arbitrate resources among RAN network slice sub-net instances (NSSI). NOTE 6 - Such closed loops may be hosted nearer to edge, e.g. nRT RIC. The policy input from higher loop may indicate, among other things, the different sources of data for the lower loop. 5. The RAN domain closed loop might also decide to offload inference tasks from ER devices to the edge or use split AI/ML model to run inference tasks on edge and ER device. This decision might be taken based on available network and compute resources. NOTE 7 - Some layers of the AI/ML model may be hosted in the wearable devices of the emergency responders, which will help in say locating of persons under distress using various inputs. Relation with autonomous behaviour- 1. Workflows for the closed loops are independent of each other. The only interaction between closed loops is via high level intents over the inter-loop interface. 2. Closed loops can create new closed loops in other network domains without human intervention. 3. Although loops are deployed in hierarchical fashion, each loop has the ability to evolve independently. It can use different models and ML pipelines as required. Each loop may move up or down the autonomy levels as defined in [ITU-T Y.3173]. 4. Closed loops have ability to split and provision AI/ML models to other closed loops in automated fashion. 5. By making closed loops in edge domain autonomous, the use case also enables lesser orchestration delay, better privacy and flexibility for verticals (e.g., industrial campus networks). 6. Higher loops can use historical knowledge available to them to optimize and generalize lower loops using high-level intent. This increases efficiency of lower loops while preserving their autonomy. (e.g., higher loop might know certain kind of ML models are good for cyclone emergency management based on previous cyclones.)

	NOTE 8 - This use case might be well aligned with the use case “Composable, hierarchical closed loops” in [b-FGAN-O-013-R1] and others above.
Use case category	Cat 1: describes a scenario related to core autonomous behaviour itself.
Reference	[b-ETSI GS ZSM 001]

7.9.1 Use case requirements

Critical requirements

- AN-UC09-REQ-001: It is critical that AN allow interaction between closed loops via high level intents.

NOTE 1 – Closed loops may create new closed loops in other network domains without human intervention.

- AN-UC09-REQ-002: It is critical that AN allow each closed loop to evolve independently, using different analytical, optimization mechanisms including ML models and ML pipelines as required.

NOTE 2 – Each closed loop may move up or down the autonomy levels as defined in [ITU-T Y.3173]

Expected requirements

- AN-UC09-REQ-003: It is expected that closed loops have the ability to provision or recommend AI/ML models to other closed loops in automated fashion.

- AN-UC09-REQ-004: It is expected that closed loops in edge domain may be autonomous, in order to enable lesser orchestration delay, better privacy and flexibility for verticals (e.g., industrial campus networks).

- AN-UC09-REQ-005: It is expected that higher closed loops use the knowledge base available to them to optimize and generalize lower closed loops using high-level intent.

NOTE 3 –This increases efficiency of lower closed loops while preserving their autonomy. (e.g., higher loop might know certain kind of ML models are good for cyclone emergency management based on previous cyclones.)

7.9.2 Actor interactions and possible components

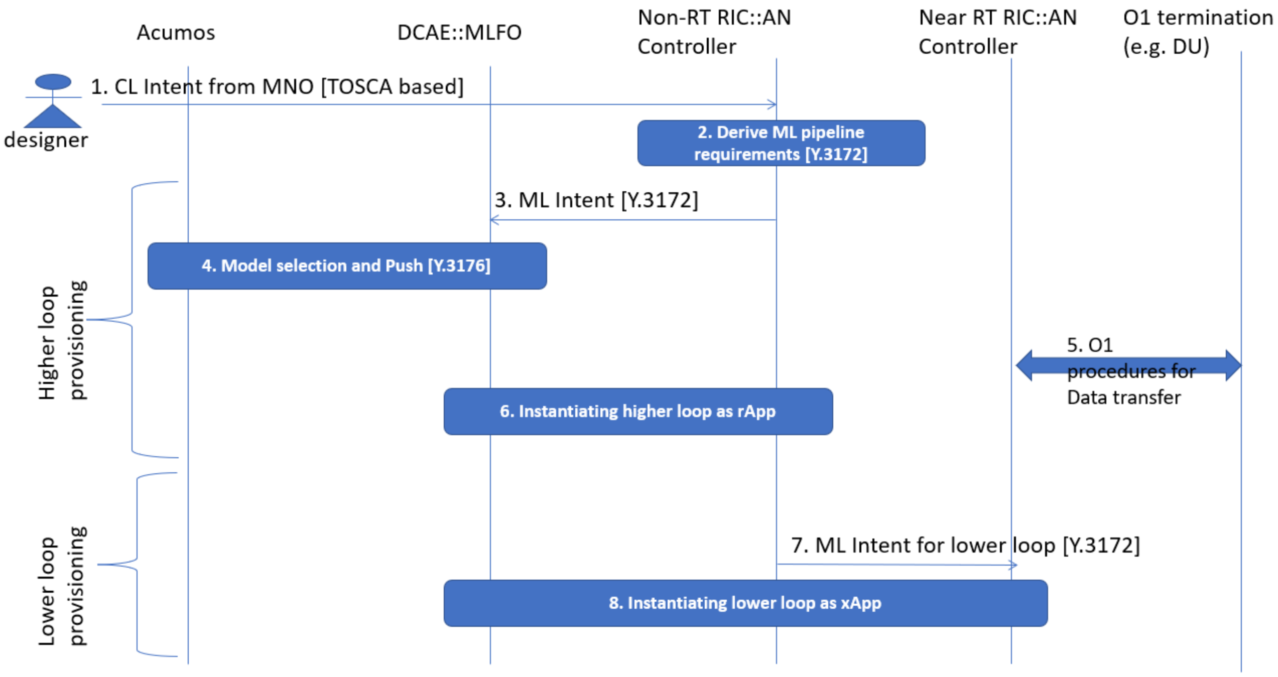


Figure 10: Actor interactions for network resource allocation for emergency management based on closed loop analysis

NOTE 1 – Create a high-level abstract model for closed loops, and then create declarative policies for that high-level model that express the "intent" of creating ML pipelines. The components ("nodes") of the high-level service are decomposed into more concrete services (possibly recursively). Declarative policies must be "translated" into more concrete declarative policies on the decomposed services in conjunction. For example, "non-RT" level service may impose certain closed loop requirements on a RIC that implements the ML pipeline. "nRT" level service may impose some other closed loop requirements on a RIC that implements that ML pipeline. This recursive decomposition coupled with recursive policy mapping happens all the way down until service components can get realized on the available resources. At that point, the low-level declarative policies must be translated somehow into imperative policies (e.g., if jitter exceeds a certain threshold, re-prioritize the traffic associated with the service).

NOTE 2 – "Imperative" policies that use the "event/condition/action" pattern, vs. declarative policies use a "capabilities/context/constraints" pattern. Declarative policies are more suitable for top-level "intent" statements, but they need to be translated (by the orchestrator) into corresponding "imperative" policies in order to be actionable. The "propagation" and "escalation" of intents: the "event/condition/action" statements are the control loops you're referring to that make sure that service components comply with desired behaviour at all times. By coupling "event/condition/action" control loops with TOSCA's substitution mapping feature, you can make these control loops "cascading", i.e. they can propagate down from high-level abstract "intent" statements to low-level device reconfigurations, and they can escalate back up if necessary.

NOTE 3 – The events are generated (using notifications) by nodes in the service topology model. The conditions are evaluated based on attribute values of nodes in the service topology model. The actions are performed on the service topology model first, and then propagated to the external world (the "resources")

7.10 Inter-domain service automation (IDSA) - for microfinance

Use case id	FG-AN-usecase-010

Use case description	Microfinance applications may be hosted by non-experts in IMT-2020 or any form of cloud / ICT technologies. The end-user requirements are domain-specific e.g. loan management, banking account/ledger management etc. The main stakeholders who are enterprises (e.g., banks) may be knowledgeable and would like to focus only in their business workflows (as against cloud / ICT technologies). The underlying cloud infrastructure (for that matter the application design) and the network infrastructure (IMT-2020 and beyond) is immaterial to a bank / finance manager. However, from a technology perspective the following rewards are desired to be reaped: 1. Give the best end-user experience: e.g., reduce down-time for services, reduced latencies for services, security and data privacy, intelligent services, by exploiting the best cloud service deployment for the microfinance application. e.g., edge, load balancing, secure messaging across multi-cloud, hybrid-cloud, AI/ML services via distributed cloud, etc. 2. Insulate the end-user from complexities of migration between generations of communication technologies: by providing interoperable, standard, backward compatible networking abstraction technologies. Integrating service lifecycle management pipeline provides agility to service development and testing. 3. Mitigate the risk of increased integration service costs: by using open source technologies, standards, benchmarking, automating in test beds. 4. Automate: reducing human involvement reduces training costs for banks, operational costs in networks and brings other benefits like intelligent fault isolation without depending on 3rd party service providers. NOTE - DevOps [b-ISO/IEC 23167] [b-ITU-T Y.3515], CI/CD [b-ITU-T Y.3525] are examples of service lifecycle management pipelines. Following are related steps in this use case scenario: 1. Intent based cloud service specification 2. processing of intent and development, validation in Sandbox/testbed. Testbed components (e.g., simulators, data models) are selected based on intent. 3. Evaluation and analysis of test results based on key parameter indices (KPI) specifications in the intent. 4. derivation of optimal configuration, cloud service deployment, management and orchestration. 5. intent based network service deployment, management and orchestration. 6. Single "cockpit" for monitoring the services 7. Autonomous, Intelligence-guided, technology-agnostic migration of services from one version of underlying technology to another and migrating applications from edge to fog. 8. Reports from various parts of the underlying technologies are provided to humans in regular intervals or event based.
Notes on use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.
Reference	[b-ISO/IEC 23167], [b-ITU-T Y.3515], [b-ITU-T Y.3525]

7.10.1 Use case requirements

Critical requirements

- AN-UC10-REQ-001: It is critical that AN consider inputs from industry vertical solution provider regarding the required service characteristics, using an intent-based mechanism, while deciding the development and deployment options for industry vertical applications and network services.

NOTE 1 – AN can autonomously decide the best possible development and deployment option for network services which can support the verticals. This has to be based on the requirements of the applications [ITU-T Y.3178]. E.g. for banking applications, service characteristics may include latency on banking transactions, mean time between service failures, level of privacy of each field in the customer profile, etc. Examples of deployment options may include edge, core cloud, enterprise network, using specific hardware etc.

- AN-UC10-REQ-002: It is critical that AN abstract the management (creation, deletion and update) of the industry vertical applications and network services, from the industry vertical solution provider.

NOTE 2 – underlying domain orchestration, network specific technologies and APIs are abstracted by AN towards the industry vertical solution provider. E.g. banking applications may be hosted as web applications (on popular web frameworks with or without an accompanying mobile component), enterprise applications (e.g. J2EE based). They may be instantiated as cloud-native applications, may use distributed architecture across private/public clouds etc. Service management infrastructure supporting the applications may include brokers, workflow managers and schedulers. Irrespective of such deployment and management variance, AN provides abstracted interfaces to verticals which hides such complexities.

- AN-UC10-REQ-003: It is critical that AN validate any changes to the application and network services in a sandbox environment before applying it in the network.

NOTE 3 – Autonomous behaviour may result in automated creation, deletion and update of applications and/or network services. The impact of such modified applications and/or network services has to be studied before they are applied in the network. This may be done by using a testbed or sandbox with simulators or even a digital twin-based environment. Specific emphasis may be applied on maintaining compatibility of the modifications with the applications and network services in the network.

- AN-UC10-REQ-004: It is critical that AN continuously monitor the application and network services in the network.

NOTE 4 –Monitoring may be done to find erroneous behaviour, faults, gaps in architecture, design, bugs, etc. Monitoring may identify the gaps in end-to-end service implementations with respect to changing intents of the verticals. Thus, monitoring may also be used to find the need for evolution in underlying network domains.

- AN-UC10-REQ-005: It is critical that AN produce regular and asynchronous reports for human consumption.

NOTE 5 – Reports to humans may summarize all monitored values, analysis, decision points and explanations for such decisions by the AN.

Expected requirements

- AN-UC10-REQ-006: It is expected that AN provide automated triggers to service lifecycle management pipeline for management (creation, deletion and update) of application and network services.

NOTE 6 – DevOps [b-ISO/IEC TS 23167], [b-ITU-T Y.3515], CI/CD [b-ITU-T Y.3525] are examples of service lifecycle management pipelines.

NOTE 7 – As part of management of applications, AN may analyse the gaps, faults and issues in the current design and implementation of an end-to-end service. Mitigation of such issues may be triggered to the DevOps pipeline. However, the level of automation of the solution may depend on the capabilities of the DevOps pipeline.

- AN-UC10-REQ-007: It is expected that the AN configuration includes the set of reference points which may be used for the integration into end-to-end network services and applications.

NOTE 8 – Even though AN exposes abstracted interfaces for application management to verticals, to achieve the integration of network services, AN may use open interfaces or closed black boxes. The availability of open interfaces and corresponding components is to be made known to AN via configurations. Such configurations may be dynamically changing based on availability of new components and interfaces.

- AN-UC10-REQ-008: It is expected that AN propose “recipes” of network services and applications which may satisfy a particular intent from the vertical.

NOTE 9 – Recipes may include a combination of existing application components, network service components, corresponding configuration options, etc.

- AN-UC10-REQ-009: It is expected that AN are updated at runtime by the underlying domain orchestration about the supported set of reference points in the domain, available set of network service and application components, which may be used for integration into end-to-end network services and applications.

NOTE 10 – Runtime changes, triggered by the operator or 3rd parties, in the underlying domains are made aware to the AN. Such updates may be abstracted and passed by the AN to the verticals, where relevant, for information or policy decisions.

<u>Added value requirements</u>

- AN-UC10-REQ-010: It is of added value that AN propose a modified “recipe” of network services and applications which may bridge a gap, fix a fault or solve issues in the current design and implementation of end-to-end services.

NOTE 11 – Modified recipe may be based on analysis of gaps, issues, or faults encountered while monitoring of network services and applications.

7.10.2 Actor interactions and possible components

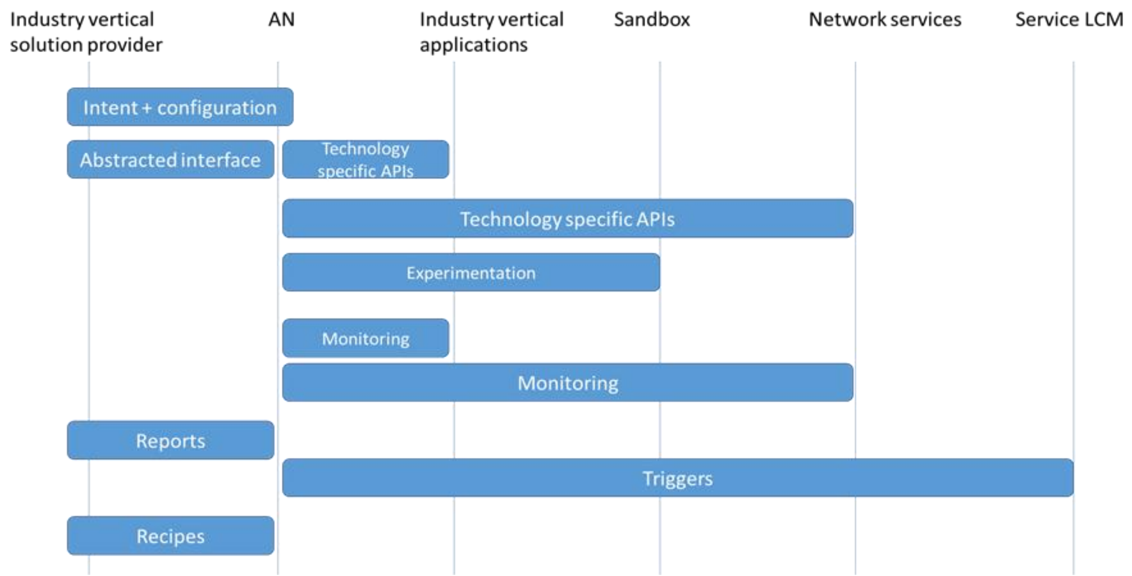


Figure 11: Actor interactions for inter-domain service automation (IDSA) - for microfinance

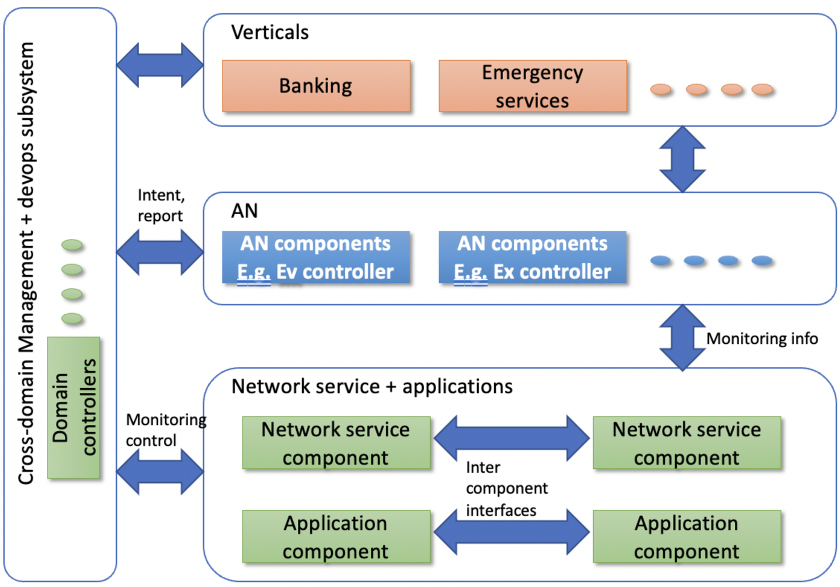


Figure 12: Possible components related to inter-domain service automation

Actors involved (figure 11):

- Industry vertical solution provider: this actor provides vertical related services which in turn uses network services in the underlay network. E.g., autonomous transport service which rely upon low latency services provided by 5G.
- Sandbox: This actor provides an experimentation platform for controllers. This allows testing and verification of controllers. This may be implemented in the form of an openly accessible sandbox.

Possible key components involved (figure 12):

- Network services and applications: includes services and applications deployed in the network, including their interfaces.
- Verticals: Includes applications used by end-users e.g. banking and emergency services.
- Domain controllers: part of the management subsystem, provides monitoring, configurations and supports reporting by various other components in the network.
- Ev controller: Evolution controller
- Ex controller: Experimentation controller

7.11 Autonomous vertical-driven edge service and middle-mile connectivity for rural financial inclusion (FI)

Use case id	FG-AN-usecase-011
Use case description	Digital Financial Inclusion in many geographies are limited because of lack of network availability and low reliability of connection. Frequently bank branches have to fall back on costly and complex connectivity through satellites. Other than the capital and operational aspects, the bank staff also needs to handle the link failover and maintenance activities in case of issues. Furthermore, such solutions do not allow local community to utilize the network. Autonomous last hop connectivity both through IMT-Advanced/IMT-2020 as well as other non 3GPP heterogeneous networks need to be seamlessly enabled to operate in an affordable manner. The solution is likely to provide the following benefits: 1. Based on the requirements from the verticals, provide connectivity for rural finance (FI)sites independent of network providers or large-Telcos. 2. Allow the connectivity to be shared with local community to ensure better return on investments. 3. Edge compute and related infrastructure can enable more compelling deployments for digital FI as well as other verticals (e.g. short-term tele-commute / interviews etc.). 4. Automate operations and security audits: reducing human involvement in maintaining and running the edge or last hop reduces training costs for banks, operational costs in networks and brings other benefits like intelligent fault isolation without depending on 3rd party service providers, continuous audit of deployed solution for security etc. Following are related sub-systems and associated steps in this use case scenario: 1. Deploy micro-servers/nano-data-centres for the edge. 2. Enable heterogeneous network connectivity. 3. Autonomous, Intelligence-guided handling of alignment or interference or mobility related challenges for various last hop approaches. 4. Automate on-boarding of community users and community specific apps, their billing/payments etc. 5. Single “cockpit” for monitoring the services and health of infra to local-bank-staff/managed-service-provider. 6. Reports from various parts of the underlying technologies are provided to humans in regular intervals or event based. This includes sharing of usage details with authorized management systems, These reports may be used for tracking the usage at a granular level, mapped to the vertical and the tracking the

	corresponding benefits from the infrastructure e.g. for the purpose of extending subsidies to such infrastructure.
Notes on use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.
Reference	

7.11.1 Use case requirements

<u>Critical requirements</u>

- AN-UC11-REQ-001: It is critical that AN utilize heterogeneous network connectivity options at the edge, in the last-mile, including the commissioning, provisioning, configuration, integration, maintenance and optimization, in a seamless, real-time and easy-to-use manner.

NOTE 1 – Especially in rural settings the technology of choice may vary considerably depending on various factors like availability of technology, ease of deployment, low power consumption, etc. Currently, in such deployments, it is invariably upon the industry vertical solution provider to also take on the responsibility of integrating and maintaining these varied last mile connectivity options. It is important that AN brings together various such technologies under one umbrella, at the edge, to provide seamless integration and maintenance. Some of the operations in the lifecycle of the last mile connectivity may need real-time interventions, some of it may need deep domain expertise and some of it may require training – all of which may not be possible in certain rural settings.

- AN-UC11-REQ-002: It is critical that AN enable sharing of the various network connectivity options at the edge, across the community driven industry vertical applications.

NOTE 2 – With an emphasis on providing maximum connectivity and application services to the local community, the best available option for connectivity need to be chosen, if needed, dynamically.

- AN-UC11-REQ-003: It is critical that AN enable onboarding of industry vertical applications at run-time.

NOTE 3 – Evolution of needs in a local community may result in changing application requirements.

- AN-UC11-REQ-004: It is critical that AN enable common, open, interoperable, adaptable mechanisms for managing end-users, based on the needs of the local community.

NOTE 4 – Onboarding, billing, problem-resolutions and other end-user management functions need to be agnostic, community-driven at the edge. Based on the use cases, the mechanisms for end-user management has to adapt. E.g. for low-mobility rural areas, a relevant tariff plan needs to be offered.

- AN-UC11-REQ-005: It is critical that AN enable a single-window of monitoring the heterogeneous underlying technologies.

NOTE 5 – Complexities of monitoring, administering, maintaining the complexities of the underlying technologies need to be hidden from the industry vertical solution provider as well as the local communities.

7.11.2 Actor interactions and possible components

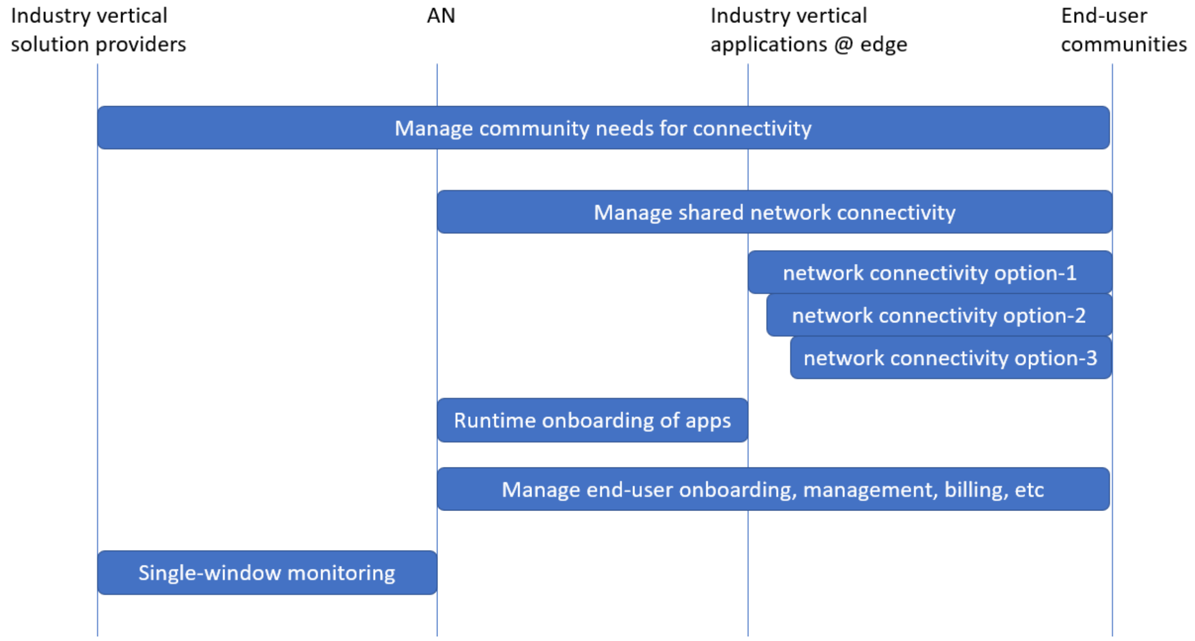
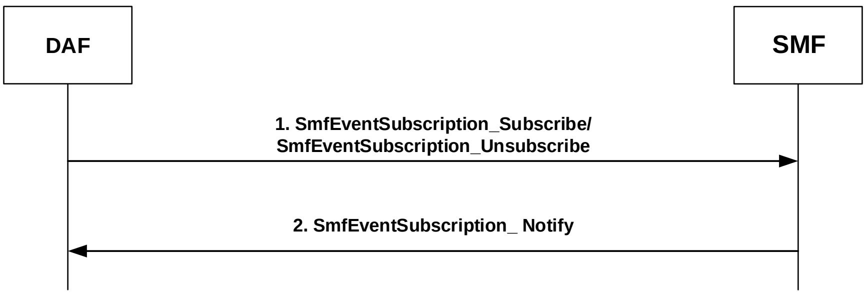


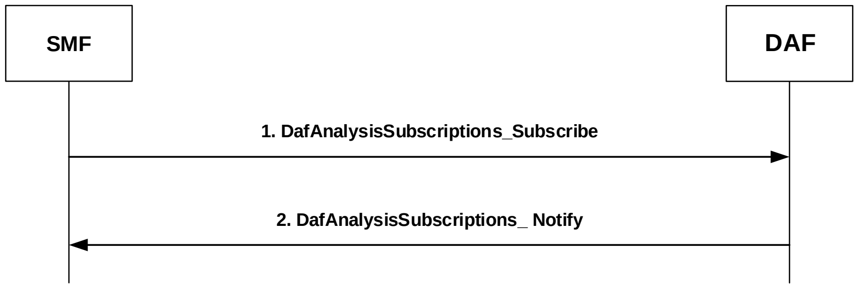
Figure 13: Actor interactions in Autonomous vertical-driven edge service

Actors involved (figure 13):

- Industry vertical solution provider: this actor provides vertical related services which in turn uses network services in the underlay network. E.g. Autonomous transport service which rely upon low latency services provided by 5G.
- End-user communities: this actor represents a set of end-users in a specific geographic location, with specific connectivity requirements.

7.12 Signalling flows for autonomous IMT-2020 network

Use case id	FG-AN-usecase-012
Use case description	Data analysis function (DAF) is introduced in IMT-2020 network [b-ITU-T Y.3104]. The signalling flow between DAF and other network functions (e.g. SMF, PCF, NACF and AF) describes data collection and analysis result providing. The procedure in figure 14 is used by DAF to collect data on event (s) related to SMF by invoking SmfEventSubscription service.  Figure 14: Signalling flow for Data Collection from SMF 1. The DAF subscribe to or unsubscribe from a (set of) Events (e.g. UE IP address, UP path change, PDU Session Establishment/ Release, and etc.) by invoking the SmfEventSubscription_Subscribe service operation.

	2. The SMF notifies the DAF (e.g. with the event report) by invoking SmfEventSubscription_Notify service operation. The procedure in figure 15 is used by NF service consumers (e.g. SMF) to request analytics information from DAF by invoking DafAnalysis_Request service.  Figure 15: Signalling flow for Analytics Subscribe/Unsubscribe from DAF 1. SMF subscribes to or unsubscribe from a (set of) data analytic events by invoking the DafAnalysisSubscriptions_Subscribe service operation. -Subscription requirements of data analytic events may include: -Load information of UPF 2. The DAF notifies the SMF about analysis events by invoking DafAnalysisSubscriptions _Notify service operation.
Use case category	Cat 1: autonomous behaviour in IMT-2020 network
Reference	[b-Q.IMT2020-PIAS]

7.12.1 Use case requirements

Critical requirements

- AN-UC012-REQ-001: It is critical that AN enable flexible provisioning and subscription of analysis parameters in network functions.

NOTE– Examples of analysis parameters are events, notifications, corresponding information and event handing controllers. Network functions may dynamically provision or subscribe to controllers and corresponding parameters.

7.12.2 Actor interactions and possible components

None.

7.13 Plug/play of network instance

Use case id	FG-AN-usecase-013
Use case description	Benefits of open architecture approach include: - reducing CAPEX through a prosperous multi-vendor ecosystem with scale economics. However, more the number of interfaces, more the effort in integration. This needs to be mitigated using automation. - Rich application space enabled using hierarchical controllers. The hierarchical control loops with varying time criticalities (<10ms (at edge) < 1s (at near edge) <

	multi-second (at orchestrator)) were discussed in [b-FGAN-O-013-R1]. However, provisioning of applications at various levels and corresponding coordination with capabilities of the network functions is a challenge. In this context, the use case "Plug/play of network instance" in open architecture is introduced. NOTE- The network instance can be a network resource, a network function, a network slice and network services [b-ETSI GS ZSM 001]. Following are related steps in this use case scenario: 1. Addition of SRCs [ITU-T Y.3172]: the network instance is plugged into the network. Data collection functions supported by this new SRCs are analysed. 2. Bottom-up bootstrapping of infrastructure layer (using cloud orchestration), network as a service (NaaS, using ONAP), services layer (using service orchestration), based on these new SRCs. OR Top down bootstrapping of apps, services, NaaS, infrastructure, based on these new SRCs.
Use case category	Cat 1: describes a scenario related to core autonomous behaviour itself.
Reference	[b-FGAN-O-013-R1], [b-ETSI GS ZSM 001], [ITU-T Y.3172]

7.13.1 Use case requirements

<u>Critical requirements</u>

- AN-UC013-REQ-001: It is critical that AN enable the plug and play of network functions in the underlay network and subsequent seamless participation of such network functions in the AN functions.

NOTE – Examples of AN functions are creation and hosting of controllers. Plug and play may be executed by manual or autonomous mechanisms.

7.13.2 Actor interactions and possible components

None.

7.14 Generative adversarial Sandbox (or hybrid closed loops)

Use case id	FG-AN-usecase-014
Use case description	In addition to open interfaces between various RAN components, a rich ecosystem of simulators is evolving. This allows implementation of various "hybrid" closed loops – part of the closed loop (e.g. data generation) is implemented in simulators whereas rest of the closed loop (e.g. analysis and action) are implemented in another part of the test network using real network functions (NF). This use case introduces "Generative adversarial Sandbox" (or "hybrid closed loops").

	Following are related steps in this use case scenario: 1. Based on the inputs from the NF (e.g. data from SRC) and existing closed loops, simulator configurations and capabilities are autonomously scripted. 2. Hybrid closed loops are autonomously composed – with parts of the closed loop in real NF and parts of it in simulators. 3. Similar to Generative adversarial Networks, hybrid closed loops are evaluated and tested using 2-part network – one simulated and another real network functions. 4. The results are analysed and ranked.
Use case category	Cat 1: describes a scenario related to core autonomous behaviour itself.
Reference	

7.14.1 Use case requirements

<u>Critical requirements</u>

● AN-UC014-REQ-001: It is critical that AN enable creation of hybrid closed loops with parts of the closed loops hosted in real network functions as against other parts of it in simulated network functions.

NOTE 1 – Examples of parts of closed loops are modules which generate data, modules which implement domain specific functions, modules which provide APIs for implementation of adapting decisions from controllers.

● AN-UC014-REQ-002: It is critical that AN enable testing and validation of closed loops using the parts of closed loops hosted in simulated network functions.

NOTE 2 – Examples of such testing are robustness related test scenarios, security and vulnerability testing scenarios.

7.14.2 Actor interactions and possible components

None.

7.15 Open, integrated, log analysis

Use case id	FG-AN-usecase-015
Use case description	Fault prediction and isolation based on log analysis is an important existing use case. Logs are generally implemented in unstructured text with no standard formats. With a disaggregated network service implementation, correlating logs from various vendors becomes a challenge. This complicates the fault prediction and fault isolation algorithms based on unstructured data from logs. This use case introduces “Open, integrated, log analysis”. Following are related steps in this use case scenario: 1. Collection of logs from various open interfaces and NFs 2. Correlation and analysis of the collected logs, across various open interfaces and NFs. 3. Identification of optimization mechanisms based on log analysis.
Use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.
Reference	

7.15.1 Use case requirements

Critical requirements

- AN-UC015-REQ-001: It is critical that AN enable correlation and identification of relevant logs, their access using open interfaces, analysis and resulting optimization of underlay networks to apply specific adaptations.

7.15.2 Actor interactions and possible components

None.

7.16 Compose-able, hierarchical closed loops

Use case id	FG-AN-usecase-016
Use case description	There are different automation loops in different levels of the architecture. High level use cases (like log-analysis based fault prediction) require access to capabilities of various network instances. This in turn may be provided by multiple vendors or opensource providers. Thus, provisioning and management of closed loops should be driven hierarchically. This use case introduces "Compose-able, hierarchical closed loops". Following are related steps in this use case scenario: 1. Declarative specifications decide the high-level aspects of closed loops. 2. They are in turn correlated with declarative specifications for network services. 3. These are then used to generate detailed declarative specifications for closed loops in different parts of the network. 4. Orchestrators at various levels generate commands to provision and manage the closed loops based on these generated declarative specifications. 5. The declarative specifications and/or closed loop components may be stored/updated for regeneration of closed loops at any point of time.
Use case category	Cat 1: describes a scenario related to core autonomous behaviour itself.
Reference	

7.16.1 Use case requirements

Critical requirements

- AN-UC016-REQ-001: It is critical that AN enable composition of hierarchical closed loops using declarative specifications.
- AN-UC016-REQ-002: It is critical that AN enable derivation of controllers at various levels of the network.
- AN-UC016-REQ-003: It is critical that AN enable management of declarative specifications of controllers.

NOTE – Management operations on declarative specifications may include creation, storage, update, delete, etc.

7.16.2 Actor interactions and possible components

None

7.17 Quality of Experience (QoE) Prediction as-a-Service (QPaaS)

Use case id	FG-AN-usecase-017

Use case description	Intelligent and autonomous troubleshooting is a crucial enabler for the current IMT-2020and beyond networks. Autonomous troubleshooting is challenging for several reasons, one of which is the availability of a wide range of applications that future networks will support.
	
	Traditionally, the methods to gain insight into the delivered quality of service and the users' experience have been through controlled laboratory experiments, where users' opinions have been collected. The results are then reported in Mean Opinion Scores (MOS), corresponding to the average of users' views. These methods are often referred to as subjective quality assessment, and there are standardized methods for conducting them.
	
	In this use case, an application or network service (NS) provider uses a QoE-Prediction-as-a-Service (QPaaS) autonomous system to conduct and follow-up QoE measurement and prediction.
	
	Firstly, the autonomous system conducts subjective tests to measure the user experience from participating users. The locations and specifications of which users are selected and how the users' responses affect the QoE depend on the application and is learned by the autonomous system. The autonomous system also measures relevant user parameters to map user opinions and application KPIs.
	
	Secondly, the autonomous system follows applicable network KPIs and map network and application KPIs.
	
	Thirdly, based on this mapping, the autonomous system enables the application provider to predict the QoE of its users based on network KPIs regardless of their participation. The autonomous system continuously (or periodically) improves the prediction accuracy by random subjective tests or user behaviour analysis.
	
	Related steps in this use case scenario are:
	• Application or network service (NS) provider demands and deploys a QoE prediction as-a-service (QPaaS) from a third-party server.
	  o Application or NS provider provides a mechanism to collect/use *user feedback* and *network metrics*.
	• Identify a method of measurement for QoE:
	  o Perform subjective tests, e.g., video streaming, two-way communications, etc. User opinions on a scale of 1-5 or thumbs up/down.
	  o Perform user behaviour analysis, e.g., gaming, AR/VR, driver assistance, etc. In a group of gamers connected via various CSP (communication service providers), if the gamers from a particular CSP face delays or a specific cell site (geographic area) is facing latency, the gaming scores and avatar-behaviour itself leave enough clues on the QoE. Similarly, on AR/VR, the level of engagement/interaction, or in assisted driving, the level of coordination between vehicles, can be measured.
	• The QPaaS server collects/processes network and application KPIs.
	• The QPaaS server determines a mapping between application KPIs and application QoE metric (MoS) using (supervised) machine learning.
	  o This mapping may be used by the application server for future objective testing of user QoE.
	• The QPaaS server collects/processes relevant network service KPIs and forms a mapping between network KPI and application KPI (or MoS) using (supervised) machine learning.

	○ The NS provider may use this mapping for future objective testing of network performance given the application. • Perform periodic verification with subjective tests/user feedback, and improve learning based on the results. Due to various applications, QoE measurement and prediction is a significant issue in future networks. The network should be able to autonomously perform QoE measurement and mapping of network KPI to QoE metrics. NOTE- As applications and network services evolve, so do their corresponding KPIs and the mappings (user satisfaction parameters, to application KPIs, and to network KPIs). See related open issues below which handles information exchange between evolving applications and NS and QPaaS.
Notes on use case category	Cat 2: describes a scenario related to the application of autonomous behaviour in the network.
Reference	[b-Jahromi], [b-Pierucci], [b-Bouraqia], [b-Liu]

7.17.1 Use case requirements

<u>Critical requirements</u>

- AN-UC017-REQ-001: It is critical that AN use both subjective information from users and QoE information derived and analysed from network services to arrive at the application QoE metric.

NOTE 1 – Subjective information from users may include user opinions and subject measures e.g. opinions on a scale of 1-5 or thumbs up/down about a streamed video. Examples of QoE information derived and analysed from network services are the level of engagement/interaction in an online game, or analysis of gaming scores and avatar behaviour.

- AN-UC017-REQ-002: It is critical that AN learn and update the process of information collection from users and derivation from network services.

NOTE 2 – For example the parameters collected, the mechanisms for collecting and the sample set for collection may be learnt and updated. Also the mapping between the application QoE metric and the information collected from users and derived from network services may evolve over a period of analysis.

NOTE 3 –The mapping between the application QoE metric and the information collected from users and derived from network services may be modelled using AI/ML techniques.

- AN-UC017-REQ-003: It is critical that AN evolve and update the mapping between application QoE metric, network KPIs and application KPIs.

NOTE 4 –The process of evolution and adaptation may be triggered by application feature additions, network service updates or user device updates.

<u>Expected requirements</u>

- AN-UC017-REQ-004: It is expected that AN enable the plugin of QoE prediction algorithms.

NOTE 5 - QoE prediction algorithms may be integrated based on abstract APIs exposed from AN, which are agnostic to the type of application and the specific underlying network technology.

7.17.2 Actor interactions and possible components

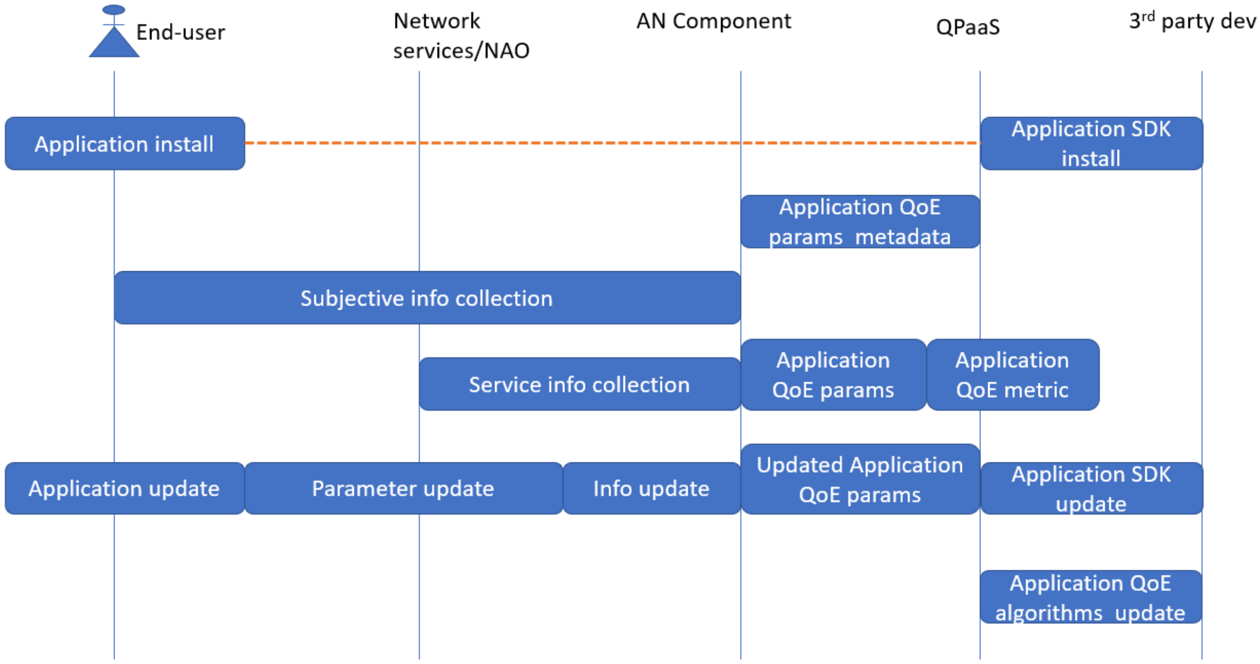


Figure 16: Actor interactions for Quality of Experience (QoE) Prediction as-a-Service (QPaaS)

Actors involved (figure 16):

QPaaS: QoE prediction as a Service is an actor which provides a platform to host third party applications which can predict the QoE experienced by end-users while using a specific application.

7.18 Autonomy applied to CDN

Use case id	FG-AN-usecase-18
Use case description	[b-FGAN-O-013-R1] introduced Autonomous content delivery networks (CDN), especially looking at a few key aspects of CDN and what makes them unique, focusing on several of their properties and approaches that can leverage to increase their autonomy. With increasing bandwidth of networks, proliferation in the connected devices, increasing demand of content (e.g. live video, cloud gaming, 360 video), build-your-own approach to CDN enabled by cloud services, cloud based CDNs are attractive but with several challenges. However caching based on data analysis remains unsolved while CDN providers struggle to provide rich content at high QoE. [b-FGAN-O-013-R1] called out specific aspects that need programmability- routing, caching and eviction. Importance of logging and metrics were called out on request/response metadata, timing information, internal logic decisions. Current implementations of closed loops for managing CDN are simplistic e.g. Standard auto scaling in the context of CDN, increasing the stream-per-node approach by hardware beef-up (better compute, more cores, more memory, L1 cache, networking including PCIe 4.0, cryptographic acceleration) The following considerations are important to note in the context of autonomous CDNs:

	1) metric for success: CDN usually define their success based on whether they can serve the user traffic. But this has a cyclic effect because the ability to attract traffic depends on how well CDNs process the traffic they currently have. The challenging part is the metric used to judge the CDN. NOTE - For example: a) incorporating the current number of requests into the score would be useful. b) if there is anonymised access to user quality of experience (QoE) data, how fast the page loaded, it would be a useful metric. c) another option is to measure the response time from the CDN (usually abridged to the hit ratio) d) include other overlays in the measurement e.g. control planes, service management, tiered storage design. e) going beyond auto scaling to healing, load balancing and edge compute, concurrency, 2) adapting the possible caching strategies (needed especially for memory intensive contents which require large memory in CDN): a) based on the treatment of various type of content and CDN use cases in the cache. e.g. live video that is cached for a brief period of time, VoD, live transcoding. To reduce the time to live (TTL) to avoid keeping infrequently popular objects in cache. b) bypass the cache for large objects, or for certain classes of users, or particular extensions c) use the disk, or explicitly forbid it d) take advantage of flexibility provided by virtual cache 3) decoupling the components help standardization: a) Open-caching is pushing to provide a subset of metrics b) Perhaps an opportunity to derive “upstream” gaps in standards and lead an opensource proof of concept (PoC). c) Study of an open, interoperable CDN components – e.g. caching, transcoding, analytics which can help independent evolution of the CDN pipeline, while taking advantage of the work in other bodies e.g. encode/decode, AI, graphics. and hardware evolution e.g. compute/mem/network/acceleration. d) Similarly, take advantage of the software deployment trajectory towards cloud native. Following are related steps in this use case scenario: 1. Outer-loop: Represent the “QoS/QoE requirements” in an intent, deployment considerations (e.g. hardware, cloud) are to be captured in the intent. software/CDN pipeline considerations are to be captured here too. 2. Based on analysis, derive the cache policy, action: auto scaling/traffic routing, geographic location (e.g. edge), decide storage configurations, APIs and concurrency mechanisms. 3. Experiment to determine a good combo of KPIs, data measurement, policies, action areas (e.g. scaling, positioning) 4. Inner-loop: Adapt the CDN and corresponding configurations based on the above, with tangible, demonstrable benefits in QoE. 5. Feedback to intent evolution – to step 1 above.
Notes on use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.

Reference	[b-FGAN-O-013-R1]

7.18.1 Use case requirements

<u>Critical requirements</u>

- AN-UC018-REQ-001: It is critical that AN enable representation of QoS/QoE requirements in an intent, and additionally deployment considerations (e.g. hardware, cloud) and software/CDN pipeline considerations are to be captured in the intent.

- AN-UC018-REQ-002: It is critical that AN enable adaptations based on analysis.

- AN-UC018-REQ-003: It is critical that AN enable experiment to determine a good combination of KPIs, data measurement, policies, action areas (e.g. scaling, positioning)

- AN-UC018-REQ-004: It is critical that AN enable the tracing of adaptations on configurations to tangible, demonstrable benefits in QoE.

- AN-UC018-REQ-005: It is critical that AN enable feedback to intent evolution.

NOTE – Feedback may include parameters for representation in the intent, additional deployment considerations, and adaptations.

7.18.2 Actor interactions and possible components

None.

7.19 Analysis-driven evolution in virtualized RAN based on DevOps

Use case id	FG-AN-usecase-19
Use case description	Open radio access network (e.g. O-RAN) architectures allow disaggregated evolution of RAN components. Programmability and interfaces exposed by RAN components in open RANs allow developers the opportunity to create applications (e.g. xApps) based on data from RAN. In parallel, development methodologies like DevOps are being applied to enable rapid introduction of services to networks. At the same time, technology evolution to the next generation of networks is in progress. This use case links the dev and ops cycle on one side to the programmability offered by new RAN architectures like O-RAN. An analysis of RAN services and applications (e.g. data, messages, interfaces, logs, etc from xApps and rApps) can provide valuable information regarding software evolution and technology evolution and deployment evolution. NOTE - [b- O-RAN.WG3.E2GAP] defines an interface, E2, connecting the Near-RT RIC and one or more O-CU-CPs, one or more O-CU-UPs, one or more O-DUs, and one or more O-eNBs. E2 Node is a logical node terminating E2 interface. The E2 enables a direct association between the xApp and the RAN functionality. Following are related steps in this use case scenario (with O-RAN as example architecture [b-O-RAN.WG1.O-RAN-Arch]):

	1. Analysis of heterogeneous RAN components, corresponding splits, capabilities, deployment options and interfaces and data models (e.g. E2 nodes and E2AP support). 2. Analyse the information in the near real time RAN intelligent controller (nRT RIC) 3. Discover the capabilities of various RAN nodes and instantiate (potentially cloud-native versions of) applications (e.g. xApps) based on RIC SDKs. 4. Provision and analyse the closed loops at near real time RIC. 5. In correlation with the non RT RIC, analyse the DevOps cycle at the near real time RIC to provision new types of CNFs in the near real time RIC and new types of E2 nodes (or new capability-needs in E2 nodes). 6. In the non RT RIC, analyse the DevOps cycles of near RT RIC, new capability needs of E2 nodes, arrive at new use cases (e.g., limitations of the current network with respect to user experience and the corresponding reasons) This use case is related to the evolution and experimentation aspects. It takes advantage of the increased data gathered from RAN via the open interfaces and the DevOps style of RIC application development to automate specific aspects of the evolution process.
Notes on use case category	Cat 1: describes a scenario related to core autonomous behaviour itself.
References	[b-DISH-AWS], [b-ONF-auto-service], [b-O-RAN.WG1.O-RAN-Arch], [b-O-RAN.WG3.E2GAP]

7.19.1 Use case requirements

Critical requirements

- AN-UC019-REQ-001: It is critical that AN enable the analysis of heterogeneous RAN components, corresponding splits, capabilities, deployment options and interfaces and data models

NOTE 1 – E.g. E2 nodes and E2AP support.

- AN-UC019-REQ-002: It is critical that AN enable the discovery of the capabilities of various RAN nodes and the instantiation of (potentially cloud-native versions of) applications.

NOTE 2 –Examples of applications are xApps.

- AN-UC019-REQ-003: It is critical that AN enable provisioning and analysis of closed loops at near real time locations.

NOTE 3 – Near real time RIC is an example of near real time location.

- AN-UC019-REQ-004: It is critical that AN in correlation with the orchestrator, analyse the DevOps cycle at the near real time locations to provision new types of network functions in the near real time locations

NOTE 4 – Further examples of new types of network functions are new types of E2 nodes (or new capability-needs in E2 nodes).

7.19.2 Actor interactions and possible components

None.

7.20 Evolving edge applications for verticals using private 5G

Use case id	FG-AN-usecase-20
Use case description	Vertical network applications like corrosion detection and intruder detection needs to be enabled at the edge using AI/ML. These applications enable inspection and surveillance services for critical industrial infrastructures. However, multi-domain (core and edge) e2e deployment of applications, on demand, is needed to achieve the required performance for these applications. In this context, 5G orchestration platform allows distributed deployment of applications, especially in exploiting the capabilities at an edge environment. This allows network operators to manage the unique KPIs of services at edge sites without exposing the network architecture. By providing an environment to develop and deploy edge applications, to serve specific needs of verticals, network operators are able to create an ecosystem for value creation, especially for domain-focussed small businesses. Following are related steps in this use case scenario: 1. Enterprises deploy private 5G network slices at edge 2. The applications and KPIs are analysed at the edge 3. Network management and optimization approaches are triggered based on this analysis 4. Tailor made applications which are specifically tuned for the needs of the enterprise are offered to the enterprise. NOTE - This aligns with the concept of NetApps and network application orchestrator (NAO) [b-FGAN-O-013-R1], decoupling the network operations logic from service provider logic and providing clear business roles. 5. The process facilitates experimentation and evaluation of candidate solutions. 6. Edge network evolution and adaptation is triggered based on the analysis. This use case is related to the evolution and experimentation aspects. It takes advantage of the increased deployment flexibility provided by private 5G networks. Edge-core information exchange is used to trigger experimentation and adaptation of the edge. There is also an expectation of alignment with domain specific experiments and matching KPIs based on innovations in the verticals.
Notes on use case category	Cat 1: describes a scenario related to core autonomous behaviour itself.
References	[b-FGAN-O-013-R1]

7.20.1 Use case requirements

<u>Critical requirements</u>

- AN-UC020-REQ-001: It is critical that AN interface with network application orchestration platforms at the edge to provide both local, vertical-specific, including real time analytics as well as non-real time analytics.

NOTE 1 –Network application orchestration platforms may coordinate with edge analytics and edge service management to abstract the edge network architecture to the AN. Analytics may be made available for remote access (e.g. away from the edge).

- AN-UC020-REQ-002: It is critical that AN provide network management and optimization, as well as application management and optimization services to application orchestration platforms at the edge.

NOTE 2 –While network management and optimization provide specific inputs to the edge about the network architecture, application management and optimization services may provide specific inputs on placement, functionalities and other aspects of applications.

Expected requirements

- AN-UC020-REQ-003: It is expected that AN provide tailor-made recipes for application management and optimization specific to verticals deployed at the edge.

NOTE 3 –These recipes may be the result of offline, generalized analytics at the AN. These recipes may be considered by NAO while designing, developing and deploying applications at the edge.

Added value requirements

- AN-UC020-REQ-004: It is of added value that AN consider input from NAO, and continuously optimize the tailor-made recipes for application management and optimization specific to verticals deployed at the edge.

NOTE 4 –The input from NAO may be agnostic to the details of network architecture or user details at the edge. This would further enhance the privacy and security of the system.

7.20.2 Actor interactions and possible components

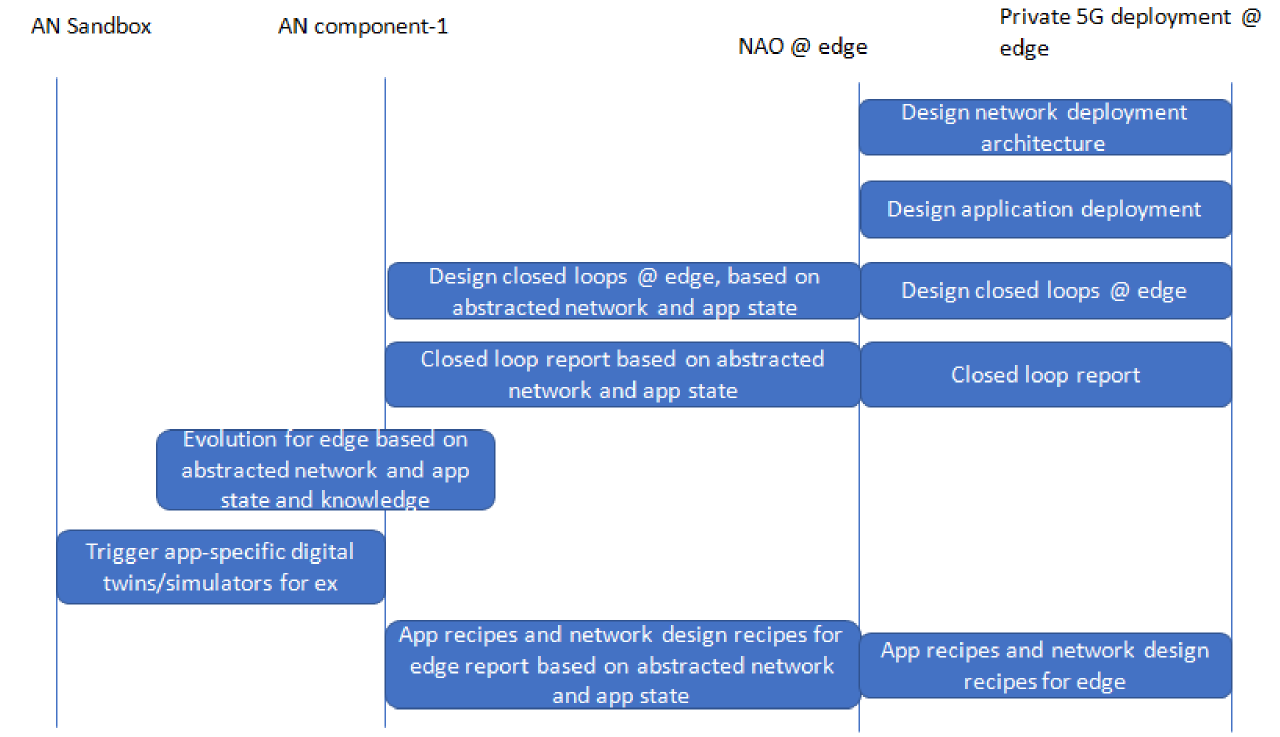


Figure 17: Actor interactions for Evolving Edge applications for verticals using Private 5G

Actors involved (figure 17):

NAO@edge: This is an actor which manages the network application and orchestrates their lifecycle at the edge locations.

7.21 Experimentation and “fire-drills” for public safety networks

Use case id	FG-AN-usecase-21
Use case description	Emergency response and public safety needs resilient, on-demand network setup and management. This will require inputs from verticals including emergency responders. Experimentation and trial runs may be mandated in certain regions. Based on the new technologies used in evolving the networks, there are different ways of deploying public safety networks. To validate the readiness of such networks, experiments need to be designed, even for the rare scenarios. In fact, for public safety networks, the design of rare scenarios is more important than the “sunny-day” success scenarios. It is also important that the experimentation matches step with the evolution of technologies used for implementing the public safety networks. Following are related steps in this use case scenario: - step 0: Continuous analysis of external inputs and creation of strategies for experiments (experiments are equivalent to "fire drills") - step 1: Closed loops are formed in sandboxes, "fire drills" are conducted and analysed. - step-2: Based on the “success” or failure of the rare scenarios, network optimization may be triggered. This use case is related to the evolution and experimentation aspects. It applies the principle similar to GAN for experimenting, validating the preparedness of public safety networks.
Notes on use case category	Cat 1: describes a scenario related to core autonomous behaviour itself.
Reference	

7.21.1 Use case requirements

<u>Critical requirements</u>

- AN-UC021-REQ-001: It is critical that AN enable continuous analysis of external inputs and creation of strategies for experiments.

- AN-UC021-REQ-002: It is critical that AN enable closed loops formation in sandboxes, where specific tests could be conducted and analysed on those closed loops.

- AN-UC021-REQ-003: It is critical that AN trigger network optimizations based on the “success” or failure of the rare scenarios in sandboxes.

7.21.2 Actor interactions and possible components

None.

7.22 Machine learning for network automation

Use case id	FG-AN-usecase-22
Use case description	[b-FGAN-O-013-R1] introduces Machine Learning-enabled network automation and the required "tailoring" of ML apps for networks. [b-FGAN-O-013-R1] calls out specific aspects that need consideration in network while applying ML: requirements from each domain, specificities (time, data and error) of each domain. Reference architecture is discussed including a ML orchestration layer. Considerations on algorithm design including trade-offs are discussed. The concept of "sub-problems" in networks and the limited role of ML is discussed too. The challenge in mix-match of data with ML (with respect to security, location, interoperability, etc) is discussed. A potentially top-down approach to service optimization is discussed. Following are the additional considerations in this use case: 1. Domain specific characteristics (called "specificities" in [b-FGAN-O-013-R1]) may not be known beforehand to the solution designer, especially given the loosely coupled architecture of future networks and ML. 2. In addition to domain-specificities, service-based specificities may be important too. Even in case of multi-domain services, specificities could be captured per-domain, E2E, at service level. So when new tenants are added, a way to dynamically capture their specificities + the domains which their corresponding specificities, is needed. 3. Agile dev and deployments in future networks may need dynamic discovery of trade-offs per service. Considering the service life cycle as day-0 (design), day-1 (deployment), day-2 (monitor), day-3 (optimization), day-4 (re-design), day-5 (evolution), feedback loops are important to enable rapid development and reduce time to market for new services. 4. run-time discovery of "sub-problems" Following are related steps in this use case scenario: 1. ML pipelines configure policies in the network based on the network QoS feedback. 2. Service metrics and related policies are provisioned in the ML pipelines based on the monitoring and analysis of errors. 3. Service redesign and optimization is triggered based on "sub-problems" and "specificities" discovered. 4. Network/domain specificities are tracked and similarly optimization problems are tracked. These are input to the service evolution.
Notes on use case category	Cat 2: Describes a scenario related to application of autonomous behaviour in the network.
Reference	[b-FGAN-O-013-R1], [ITU-T Y.3172]

7.22.1 Use case requirements

<u>Critical requirements</u>

● AN-UC022-REQ-001: It is critical that AN enable, in case of multi-domain services, specificities per-domain, E2E.

NOTE 1 – This includes a way to dynamically, autonomously capture their specificities + the domains which their corresponding specificities.

- AN-UC022-REQ-002: it is critical that AN enable agile dev and deployments in future networks dynamic discovery of trade-offs per service.

NOTE 2 – Considering the service life cycle as day-0 (design), day-1 (deployment), day-2 (monitor), day-3 (optimization), day-4 (re-design), day-5 (evolution), feedback loops are important to enable rapid development and reduce time to market for new services.

- AN-UC022-REQ-003: It is critical that AN enable run-time discovery of "sub-problems".

7.22.2 Actor interactions and possible components

None.

7.23 Autonomous agents (with varied competence) in networks

Use case id	FG-AN-usecase-23
Use case description	[b-FGAN-O-013-R1] introduces autonomous Systems in hostile environments. Estimation and judgement of competence as key criteria for determining the right level of autonomy is the focus. Levels of autonomy for unmanned systems are introduced, especially ranging from "sub-functions" to single functions to single system to teams. A framework for robot autonomy is discussed with corresponding guidelines. Based on these, the characteristics for operations in hostile environments are listed. [b-FGAN-O-013-R1] calls out the requirements for operations with autonomous systems. Relation between the needed level of autonomy depending on environment and type of task in contrast to capabilities of the system in combination with policies. Systems should provide the best possible support and hence the autonomy level has to be adjusted such that humans only have to intervene when it is necessary and makes sense. Due to dynamics during missions in hostile environments, the systems have to adapt their autonomy level at run-time (or humans have to do it) according to the situation and the corresponding requirements. The existing stage levels of autonomy have to be extended by the ability of switching the level in run-time (judgement by design vs. judgement in run time) A simple workflow scheme for autonomy with varying autonomy levels is discussed. This includes task specification by human and task understanding, feasibility check, task planning, task execution by system. Request for support to human and control by human can be to any of these workflow steps. Monitoring of performance levels by humans and learning by the system are added steps. Autonomy level and dependency on competence are discussed. Competence analysis as a weighted function of capabilities needed, capabilities existing, existing options for actions and existing constraints is described. Following are the additional considerations in this use case: 1. Taking telco service design, development, deployment and operations (ops) as an example - the levels of autonomy may be applied in as follows: (a) Service design is done by designers (100% designer interaction, no ops interaction)

	(b) Existing software development kits (SDKs) and application programming interfaces (APIs) are exercised to create applications (e.g. rApps or xApps) – (high level designer interaction, high code but low ops interaction) (c) Configuration of existing or new services in real time environments (e.g. distributed unit (DU) – mid level designer interaction, high ops involvement). (d) Service deployment and QoE measurement in customer premises (e.g. VoD – no code, low designer interaction, collaborative, high ops involvement). Thus, it may be relevant to consider the nature of the task in addition to the type of environment, e.g. for the ops engineers the design phase is a "difficult environment" (due to low involvement) and for the service designers the customer premises is a "difficult environment" (due to constraints in site visits). Thus, it may be relevant to consider a multi-agent system where the agents have varied competences (capabilities, options for actions and constraints). Following are related steps in this use case scenario: 1. Problem detected in the network: E.g. Video performance degradation for customers, agents collect debug data. analysis agents with matching capabilities are deployed at the nRT RIC and triggered to analyse the data. 2. Fault isolation: example of steps in the scenario are in CU, DU, user plane and control plane, collaborative analysis is used to pin-point the cause of failure. This may involve multi-agent team communication to do the steps in the workflow described in [b-FGAN-O-013-R1] 3. Fault correction: example of steps in the scenario are parameter configuration, service upgrade or software re-configuration. 4. Analysis of task performance by agents: example of steps in the scenario are collaborative analysis is used to collect data from the agents, including where the human interactions were needed. 5. Trigger creation of new agents with new capabilities: example of steps in the scenario are location and capabilities are selected based on the next higher level of autonomy to reduce human interaction in this scenario.
Notes on use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.
Reference	[b-Beyerer], [b-Hesse], [b-FGAN-O-013-R1]

7.23.1 Use case requirements

Critical requirements

- AN-UC023-REQ-001: It is critical that AN enable, autonomous agents to collect debug data.

NOTE 1 – For example analysis agents with matching capabilities are deployed at the nRT RIC and triggered to analyse the data.

- AN-UC023-REQ-002: It is critical that AN enable, fault isolation (e.g. in CU, DU, user plane, control plane) using collaborative analysis is used to pin-point the cause of failure.

NOTE 2 - This may involve multi-agent team communication to do the steps in the workflow.

- AN-UC023-REQ-003: It is critical that AN enable, fault correction (e.g. parameter configuration) using service upgrade or software reconfigurations.

- AN-UC023-REQ-004: It is critical that AN enable analysis of task performance by agents

NOTE 3 – Collaborative analysis is used to collect data from the agents – including where the human interactions were needed.

- AN-UC023-REQ-005: It is critical that AN enable creation of new agents with new capabilities.

NOTE 4 – Example location and capabilities are selected based on the next higher level of autonomy to reduce human interaction in this scenario.

7.23.2 Actor interactions and possible components

None.

7.24 Automated, adaptive acceleration for AI @ edge

Use case id	FG-AN-usecase-24
Use case description	[b-FGAN-O-013-R1] introduces spatial architectures which scale performance & resources to meet the application requirements and Scaling to fit into available resources – in the context of DNN. It also discussed that reduced precision can be highly effective to reach communication requirements. Spatial architectures can exploit custom arithmetic at a greater degree. Further, it discussed topologies fully co designed for hardware architecture, where the Circuit is the DNN [b-Umuroglu]. Adjust the parameters of DNN (lookup table (LUT) contents) while iterating on training dataset until accuracy. [b-FGAN-O-013-R1] shows results (with an example of intrusion detection) that spatial processing, customized arithmetic and learned circuits can help scale to communication throughput and latency requirements. [b-FGAN-O-013-R1] also talks about [b-Blott] and Providing tools and platforms for exploration of DNN compute architectures. ML engineers can create specialized hardware architectures on an FPGA with spatial architectures and custom precision. Design and runtime software tools (e.g. FINN) for DNN to FPGA development starting with training or learning reduced precision DNNs, using ONNX based intermediate representation, perform optimization on this intermediate representation, to create a DNN hardware IP, was discussed. Thus, it may be relevant to consider the following aspects for this specific use case: 1) AI-enabled applications are increasingly being deployed at the edge. Low latency, low power consumption and small footprint are considerations for AI applications at the edge. Accelerated, AI-enabled applications at the edge are important enablers for future networks. 2) As AI technology evolves, AI models evolve, the acceleration platform must also be adaptable and at the same time satisfying the requirements above. Also, reduced time to market, development time and cost to reach production readiness, are important factors influencing deployment decisions by network operators. fully customized circuit board is developed for each application may not fit this bill. 3) Pluggable solutions into a larger edge application, providing both the flexibility of a custom implementation with the ease-of-use and reduced time-to-market of an off-the-shelf solution, are needed. 4) Adaptive computing includes hardware that can be highly optimized for specific applications such as Field Programmable Gate Arrays (FPGAs). In addition to FPGAs, new types of adaptive hardware such as adaptive System-

	on-Chip (SoC) which contains FPGA fabric, coupled with one or more embedded CPU subsystems, have been introduced recently. 5) Prebuilt platforms and APIs, software tools enable full customization of the adaptive hardware, enabling even more flexibility and optimization. This can be used to design highly flexible, yet efficient systems at the edge. 6) Exploiting the development and adoption of standards in interface and protocols at the edge, different AI-enabled edge applications can use similar hardware components. Following are related steps in this use case scenario: 1. Given an AI/ML model layered architecture, the following considerations needs to be applied – (a) concurrency in processing of layers, (b) fragmentation/buffering between layers vs. offloading of layers into compute (c) precision vs. performance and energy efficiency. 2. Given the specific goals and constraints of the AI/ML model, consider the trade-off between complexity of target platform architecture and precision to explore the model architecture and layer compositions. 3. Transformation of an AI/ML model, going through the process of intermediate representation, optimization, hardware implementation, evaluation and back to training/modelling. 4. Derive feedback for hardware adaptation and design.
Notes on use case category	Cat 2: Describes a scenario related to application of autonomous behaviour in the network.
Reference	[b-Blott], [b-Xilinx], [b-Umuroglu], [b-SCALE-Sim], [b-FGAN-O-013-R1]

7.24.1 Use case requirements

Critical requirements

- AN-UC024-REQ-001: It is critical that AN enable, analysis of concurrency in processing of layers in a DNN, fragmentation/buffering between layers vs. offloading of layers into compute and analysis of precision vs. performance and energy efficiency.

- AN-UC024-REQ-002: It is critical that AN consider the trade-off between complexity of the target platform architecture and precision to explore the ML model architecture and layer compositions.

- AN-UC024-REQ-003: It is critical that Autonomous Networks enable a bi-directional pipeline to support the transformation of AI/ML models encompassing specification, optimisation, implementation, evaluation, and deployment.AN

- AN-UC024-REQ-004: It is critical that AN derive feedback for hardware adaptation and design.

7.24.2 Actor interactions and possible components

None.

7.25 Assistive networks: Adaptation of communication system based on changing user accessibility needs

Use case id	FG-AN-usecase-25

Use case description	For both disabled and the able-bodied, as the accessibility requirements evolve, adaptations need to be applied on the network, device and user profiles need to align with the changing needs of the user. The scope of assistive technologies needs to be broadened to "assistive networks". Assistive networks can be thought of E2E network slices that include assistive, adaptive, and rehabilitative connectivity for persons with specific needs. It also includes the automated mechanisms used in selecting, locating, using and customizing the networks. Assistive networks promote greater independence by enabling people to connect to the devices and network more autonomously. Environmental models exist for surroundings but building on top of such environment models to adapt the connectivity to the user with specific requirements is the need of the hour. User model and simulations are needed to provide inputs to AN. Development of standard definitions are needed for application model, assistive network, context modelling, environmental model and common user profile, metadata, simulation and virtual instance in the context of AN. Individualization needs to be added as an important dimension of AN for future networks. Reuse of common user profile (CUP) to automate the collection, analysis and adaptation of the network and applications is proposed here. Following are related steps in this use case scenario: 1. Environment model including the network environment is built for the user. e.g. Radio propagation models, signal strengths with respect areas, mobility prediction models. 2. User model is accessed and updated. e.g. User specific constraints, user inside a car wearing seat belt has limited mobility within the car. Similarly for elderly and children, persons under emergency needs. 3. Simulations are used (offline and/or real time) to determine the changes and adaptations needed in the network to satisfy the needs of the user. e.g. Digital twins which include environment simulations and user specific criteria. 4. Adaptations are applied to the network and the context. e.g. Drone based coverage is provided, reconfigurable intelligent surface (RIS) configurations or beam configurations to provide better coverage. 5. Generalizations and evolutions are studied for applicability in a larger context. e.g. continuous update of models, transfer of model parameters across domains for easy learning, evolution of network simulators and context for new encountered scenarios. This use case is related to adding environment sensing and adaptation to include inclusivity in the evolution and experimentation aspects.
Notes on use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.

References	[b-J.acc-us-prof], [b-Biswas], [b-AVA-1], [b-AVA-2], [b-ISO/IEC 24756], [b-ISO 9241-129], [b-KUKA]

7.25.1 Use case requirements

Critical requirements

● AN-UC025-REQ-001: It is critical that AN enable, creation and representation of environment model including the network environment for the user with assistive needs.

NOTE 1 – E.g. radio propagation models, signal strengths with respect areas, mobility prediction models.

● AN-UC025-REQ-002: It is critical that AN enable updating of user model.

NOTE 2 – E.g. user specific constraints, user inside a car wearing seat belt has limited mobility within the car. Similarly for elderly and children, persons under emergency needs.

● AN-UC025-REQ-003: It is critical that AN enable Simulations (offline and/or real time) to determine the changes and adaptations needed in the network to satisfy the needs of the user.

NOTE 3 – E.g. digital twins which include environment simulations and user specific criteria.

● AN-UC025-REQ-004: It is critical that AN enable Adaptations applied to the network and the context.

NOTE 4 – E.g. drone based coverage is provided, reconfigurable intelligent surface (RIS) configurations or beam configurations to provide better coverage.

● AN-UC025-REQ-005: It is critical that AN enable generalizations and evolutions which are studied for applicability in a larger context.

NOTE 5 – E.g. continuous update of models, transfer of model parameters across domains for easy learning, evolution of network simulators and context for new encountered scenarios.

7.25.2 Actor interactions and possible components

None.

7.26 Ev-as-a-service: Achieving zero touch evolution in a delegated autonomy case

Use case id	FG-AN-usecase-26
Use case description	Some usage scenarios assume a multi-domain architecture. Each domain may have its own orchestrator. NOTE 1 – Example in FG-AN-usecase-20, network application orchestrator (NAO) is mentioned. MANO/NFVO is used in the NFV domain, service orchestrator may be used similar to ONAP in the communication domain. Closed loops are assumed in each domain, managed by the corresponding orchestrators. (For the purposes of this use case), it is assumed that each closed loop enables autonomous behaviour in that domain for specific use cases e.g. resource scaling based on load. NOTE 2 - The autonomous behaviour enabled by use case specific closed loops and managed by domain orchestrators can be extended to any number of management domains. Current frameworks [ITU-T Y.3172], [ITU-T Y.3179], [b-ETSI GS ZSM 002] assume offline development and provisioning of services which form the closed loops. E.g. AI/ML model training based on data from the network, followed by model serving in the network.

	This use case introduces an evolution (Ev) function which analyses the inputs from the closed loops (and other context information in the domain orchestrator) to trigger creation of new services which can cater to the evolving needs of the domains. The triggers may be input to DevOps pipeline. This may result in creation of new framework services or new applications or new VNFs or new configurations or new AI/ML models etc These may then be tested and evaluated in an experimental setup (e.g. Digital twins, Sandbox, etc) and deployed in corresponding domains using the domain orchestrators. In summary, the use case aims to monitor, identify the need for ev, generate new function to support this need, “(re-)inject” that function through DevOps pipeline into the closed loops. Note that this may require multi-domain coordination to modify the closed loops and may be challenging from an implementation perspective. Implementation may depend on the capabilities provided by the underlying closed loop frameworks e.g. ZSM. Levels of “mutation” of CL: There can be a spectrum of adaptation changes to the closed loops (CL): a) No adaptation at all – same input to the CL, always leads to the same output. b) Limited adaptation – CL improves utility over time, so same input may not lead to the same output (after improvement). c) Full-fledged evolution, involving development and injection of new functions. The capability of underlying closed loop frameworks may be factor in deciding the level of adaptation possible in the AN. Division of responsibility between the controller and the CL: 1) “Unintelligent” vs. “intelligent” CLs: unintelligent CL may allow full re-configurations and re-injections of functions which may allow overall mutation of its functionality over time. Whereas an “intelligent” closed loop may use its intelligence to limit external influence by the controller. In any case, the domain orchestrator should know the mutation-capabilities of the CL. In case of limitations encountered for adaptations, the CL (or domain orchestrator) should be able to escalate the requirement to higher domains. 2) Timescale of Ev has to be agreed between the CL and the orchestrator. Following are related (example) steps in this use case scenario: 1. Enterprise/vertical provides intent for application/service 2. A corresponding slice is created by NAO. 3. Corresponding resources are allocated by NFV MANO 4. NSaaS may be instantiated using ONAP/SO. 5. use case specific closed loops are instantiated in each domain e.g. power optimization, interference management, resource utilization, self-x. 6. Ev as a Service is instantiated in the zero touch framework. 7. Based on the analysis of inputs from use case specific closed loops and domain orchestrators, Ev triggers configurations, updates, service instantiation, new closed loops, even new service development (using triggers to Devops pipeline). 8. After testing and validation, such updates are reflected in the domains and use case specific closed loops.

	This use case proposes Ev-as-a-service scenario in relation to zero-touch frameworks.
Notes on use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.
References	[b-ETSI GS ZSM 001], [b-ETSI GS ZSM 002], [b-ETSI GR ZSM 009-3], [b-Ciavaglia-1], [b-ETSI GS ZSM 013], [ITU-T Y.3172], [ITU-T Y.3179]

7.26.1 Use case requirements

Critical requirements

● AN-UC026-REQ-001: It is critical that evolution function (Ev) in AN analyses the inputs from domain specific closed loops (and other context information in the domain orchestrator) to trigger management (creation, update and delete) of network services, which may in turn participate in the closed loops.

NOTE 1 –The management (creation, update and delete) of network services, over a number of iterations, may result in evolution.

● AN-UC026-REQ-002: It is critical that the modifications to network services and applications may be tested and evaluated in an experimental setup and deployed in corresponding domains using the domain orchestrators.

NOTE 2 – Examples of experimental setups used for testing and evaluation of network services and applications are Digital twins, Sandbox, etc.

Expected requirements

● AN-UC026-REQ-003: It is expected that management of network services or applications or VNFs or configurations or AI/ML models is done at runtime in coordination with DevOps pipelines.

● AN-UC026-REQ-004: It is expected that domain specific closed loops allow management of network services or applications or VNFs or configurations or AI/ML models in coordination with AN components.

NOTE 3 – There can be a spectrum of adaptation changes (levels of "mutation") of network services:
a) No adaptation at all
b) Limited adaptation – improves utility over time
c) Full-fledged evolution, involving development and injection of new functions.
The capability of underlying closed loop frameworks may be factor in deciding the level of evolution and adaptation possible in the AN.

Added value requirements

● AN-UC026-REQ-005: It is of added value that Ev function in AN act as consumer of mutation functions provided by underlying service management frameworks and in turn provides evolution service to underlying service management frameworks.

NOTE 4 – Example of a service management framework is ETSI ZSM [b-ETSI GS ZSM 001].

7.26.2 Actor interactions and possible components

None.

7.27 Experimentation as a service: Digital twins as platforms for experimentation

Use case id	FG-AN-usecase-27
Use case description	[b-FGAN-O-013-R1] described a digital twin (DT) as a representation of a physical and/ or logical object. The contribution proposed to build Digital Twins of Computer Network infrastructures. Some examples of the (hypothetical) questions which could be answered using digital twins were listed as: Which is the best network upgrade given a budget? Which is the best link upgrade to accommodate a new customer? What is the method to support a new customer SLA with the current network capacity? etc. The impact of digital twins in Network Planning and Upgrading, Troubleshooting and Performance Analysis, What-if Analysis were described in [b-FGAN-O-013-R1]. This makes digital twins a perfect environment for experimentation in the context of autonomous networks. [b-FGAN-O-013-R1] took the approach of using neural networks (NN) to build digital twins. The approach using graph neural network (GNN) [b-Scarselli] was described. It generalizes to unseen topologies, routings and traffics. Specific example of RouteNet [b-Rusek-2] was described. RouteNet can generalize to unseen topologies, routings and traffic matrices. DRL+GNN looks as a promising technique for real-time network optimization was introduced in [b-FGAN-O-013-R1]. AN aims to remove the human from the control loop. This poses hard challenges to offer 100% guarantees once the AN products are deployed in networks. In order to achieve mature solutions for autonomous network control, it will be essential for AN vendors to validate in advance that their products will operate successfully in the target customer networks, before they are actually deployed. A DT can be used to estimate accurately the resulting network performance of an experimentation approach and the effect after applying the actions produced by the AN, thus determining what network scenarios are well-supported by the product. After a comprehensive validation test, the vendor can apply the adaptations to the network. Following are examples of scenarios in this use case: Scenario-1: **preparation of DT:** import network configurations (including closed loops) into digital twin. This sets the stage for preparation of simulations in the DT. Scenario-2: **trigger of DT for simulations**: update of network configurations in digital twin (if any, by engineer), followed by simulations in digital twin and generation of asynchronous events. These events are consumed by AN engine and may in turn result in experimental configurations/updates from the AN engine towards the DT. This cycle may continue based on the sequence of simulations and scenarios in the DT. The validation of KPIs in the DT as a result of experimentation and adaptations by the AN engine is an important step. Scenario-3: **trigger of AN engine by operator**: update of network policy/configurations by engineer, which triggers AN engine to corresponding

	experiments or configurations towards the digital twin. Experimentation may be configured in the digital twin and corresponding events and KPIs may be used to evaluation the result of the experimentation. This may result in selecting the best possible sequence of actions or adaptations towards the network. The validation of the AN engine actions is an important step here. Scenario-4: **trigger based on evolution**: Other triggers for experiments in AN engine may include inputs from evolution functionality. Experimentation and evaluation of actions or adaptations towards the network are same as above. Following are related steps in this use case scenario: 1. Import environment into DT, trigger simulations in DT and validate the results, especially the use case specific closed loops. 2. AN-triggered experiments and adaptations are tested using corresponding simulator settings in DT and evaluating the impact in simulations. Describe the relation with autonomous behaviour (if any). - this use case is related to the concept of experimentation.
Notes on use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.
References	[b-Rusek-1], [b-Rusek-2], [b-Almasan], [b-FGAN-O-013-R1]

7.27.1 Use case requirements

Critical requirements

- AN-UC027-REQ-001: It is critical that AN enable import of simulation environment into DT, trigger simulations in DT and validate the results, especially the use case specific closed loops.

- AN-UC027-REQ-002: It is critical that AN-triggered experiments and adaptations are tested using corresponding simulator settings in DT and the impact in simulated environment is evaluated.

7.27.2 Actor interactions and possible components

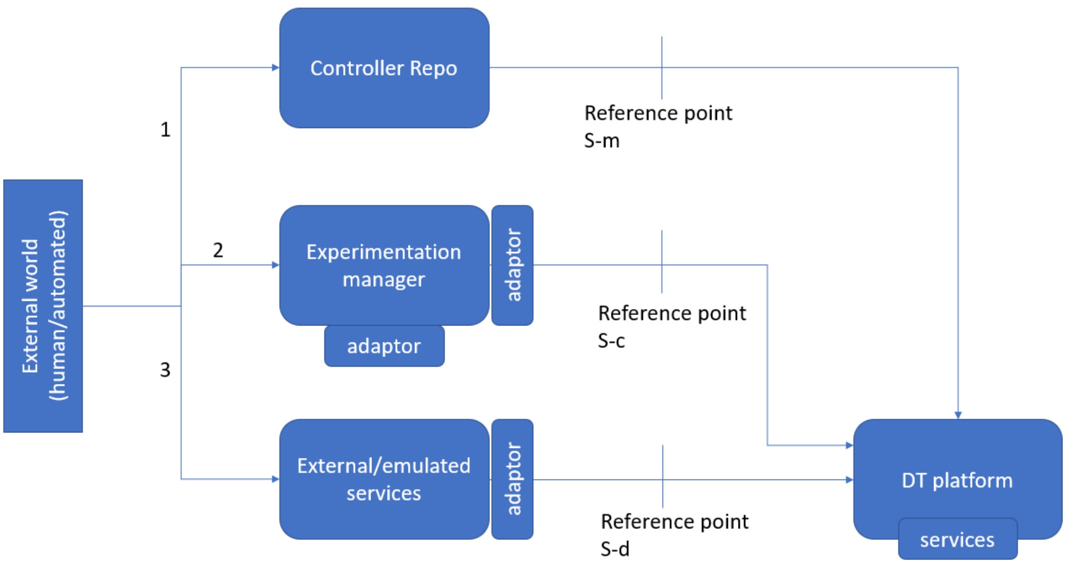


Figure 18: Possible components for experimentation as a service

Possible key components involved (figure 18):

- Experimentation manager: This is a component which manages the experimentation stage of controllers, including testing and verification.
- DT platform: This component provides a Digital Twin platform that allows testing and verification of controllers.
- Controller Repo: This component provides a repository of controllers.

7.28 Evolution from scenario-specific, explicit-coordination to coordination-free interoperability (achieved using data-driven approaches)

Use case id	FG-AN-usecase-28
Use case description	Deep learning is being used to address challenging problems in wireless communications such as modulation recognition, radio fingerprinting and many other scenarios. The advantages of this approach include the capability to address wide-range of scenarios, where a mathematical model is difficult to make (e.g. channel estimation, beam management for future networks). Existing solutions mostly rely on explicit coordination between the transmitter (TX) and the receiver (RX), introducing problems of interoperability and necessity for standards. Such signalling messages eat into the costly spectrum and complicate protocol design. A data-driven approach based on neural networks (NN) is an alternative to achieve coordination-free interoperability. An example scenario for the use case is as follows: Millimeter wave (mmWave) communication with large antenna arrays is a promising technique to enable extremely high data rates. Because of their highly directional transmissions, radios operating at millimeter wave (mmWave) frequencies need to perform beam management to establish and maintain reliable mmWave links. Transmitter (TX) and the receiver (RX) need to coordinate to select the beam pair that yields the highest beamforming gain. Currently 5G NR defines the stages of exhaustive beam sweep (EBS) as: initial access (IA) and beam tracking. For both IA and beam tracking, the 3rd Generation Partnership Project (3GPP) NR standard for 5G communications utilizes synchronization signal blocks (SSBs) [b-3GPP 36.331]. Beam management for the IA procedure in 3GPP NR involves: 1) Beam sweep: Base station transmits directional Synchronization Signals (SSs) to cover all the TXBs of a certain codebook. Each beam is swept with an SSB, which is a group of 4 OFDM symbols and 240 subcarriers in frequency. 2) Beam measurement: The User Equipment (UE) itself, if configured for directional reception, performs a directional scan, measuring the quality of each beam pair 3) Beam decision: The UE selects the beam to be used to perform initial access 4) Beam reporting: During the next SSB in the selected direction, the UE acquires information on the time and frequency resources in which the base station will be in receive mode for the random access message using the same TXB.

	Receiver can associate Signal-to-Noise-Ratio (SNR) levels to beams without explicit coordination with the transmitter using pilot-less estimation technique. The RX infers the Angle of Arrival (AoA) and the TXB by passively eavesdropping on data transmissions to other users in the network. This is enabled by leveraging a data-driven approach based on convolutional neural networks (CNNs) to achieve coordination-free beam management in mmWave networks based on a unique "signature" of the beam from the impairments. This method infers the Angle of Arrival (AoA) of the beam and the actual beam being used by the transmitter through waveform-level deep learning on ongoing transmissions between the TX to other receivers. Experimentation is achieved by experimental data collection campaign with two software-defined radio testbeds, and by using multiple antennas, codebooks, gains and locations and includes 3 different AoAs and multiple TX and RX locations. Evaluation criteria includes an upper bound on the expected search time of the proposed algorithm. The proposed technique reduces latency by up to 7x with respect to the 5G NR initial beam sweep in a default configuration and with a 12-beam codebook. Following are related steps in this use case scenario: 1. Based on the analysis of data from the network, reference points are selected by evolution where data-driven NN based approaches can be applied to reduce signalling. Evolution should cherry-pick the reference points which has the best trade-offs in terms of benefits (e.g. spectral efficiency, latency, etc) as against the cost of training. Evolution should also help in understanding the experimentation approaches to follow. 2. Based on the scenario under study (for evolution), experimentation is setup and data sources are provisioned and ML pipelines are setup [ITU-T Y.3172]. 3. Based on the evaluation of the AI/ML models in the sandbox, they are injected into the network functions (NFs). 4. Control and data flows are modified according to the evolved network.
Notes on use case category	Cat 1: Describes a scenario related to core autonomous behaviour itself.
Reference	[b-O'Shea-1], [b-O'Shea-2], [b-Jagannath], [b-Mao], [b-AI4Good-1], [b-AI4Good-2], [b-AI4Good-3]

7.28.1 Use case requirements

<u>Critical requirements</u>

- AN-UC028-REQ-001: It is critical that AN enable selection of reference points are based on evolution where data-driven NN based approaches can be applied to reduce signalling.

NOTE – Evolution should cherry-pick the reference points which has the best trade-offs in terms of benefits (e.g. spectral efficiency, latency, etc) as against the cost of training. Evolution should also help in understanding the experimentation approaches to follow.

- AN-UC028-REQ-002: It is critical that AN enable, based on the scenario under study (for evolution), experimentation setup and data sources provisioning and ML pipelines setup [ITU-T Y.3172].

● AN-UC028-REQ-003: It is critical that AN enable, injection of ML models into the network functions (NFs), Based on the evaluation of the AI/ML models in the sandbox,

● AN-UC028-REQ-004: It is critical that AN enable, modification of Control and data flows according to the evolved network.

7.28.2 Actor interactions and possible components

None.

7.29 Intelligent maintenance assistance system

Use case id	FG-AN-usecase-029
Use case description	The intelligent maintenance assistance system is an intelligent service system for network operation and maintenance. The system combines AI algorithms and AR capabilities to provide intelligent assistance for front line staff of operators in aspects of network operation and maintenance. The system includes backstage support system and AR glasses app. The backstage support system is deployed in cloud servers in the form of micro-service, and is connected with the network management system of operators to exchange data. The AR glasses app is used for staff's on-site work. The backstage support system provides the following functions: 1. AI algorithms and AR capabilities for system. Developers can also use these algorithms to develop applications. a) AI algorithms include Bar code / QR code recognition, OCR, device port recognition. b) AR capabilities include image recognition and tracking, 3D object recognition and tracking, visual simultaneous localization and mapping (SLAM). 2. The data management function can help implement equipment data transmission, storage, and management, including network resource data, equipment status data, equipment operation data, etc. 3. The system management function includes user management, authority management, system operation management, parameters configuration and other functions to ensure the stable and reliable operation of the system. The AR glasses app provides the following functions: 1. Information collection: Staff can collect pictures through the camera of AR glasses and upload them to the backstage support system. The AI algorithms in the backstage support system can recognize the collected pictures and save the recognition results. Here, the staff can collect the text information of the labels of devices to supplement and update the information in the backstage support system. 2. Data visualization: The data managed in the backstage support system can be displayed on the AR glasses. This function can provide the display of alarm information, base station information, equipment information, electricity consumption and network status, to assist the network operation and maintenance of staff. 3. Remote guidance of experts. Through the camera of AR glasses, experts get the situation of the work site and provide remote guidance.

	The steps in this use case are as below: 1. Using AR glasses (and other external sensors), collect data about the environment, which includes equipment label, port, electricity consumption etc. This step may include barcode / QR code recognition, OCR, Device port recognition etc. 2. AI based cognition analysis, perception visualization and other analysis algorithms are applied on the collected data to create a virtual model of planning and design to real environment to assist network designers. This step may include application of Image recognition and tracking, 3D object recognition and tracking, Visual SLAM. 3. This model is then used in conjunction with real data for maintenance and optimization by intelligence maintenance assistance system. This step may involve query of the virtual model, analysis of real alarms, cell data, along with the virtual model, to create intelligence assistance for frontline workers. This step may use network data management, system management and core algorithms for the whole system. The output from this step may include 3D models which can be rendered in AR glasses, AI processed network information for display, real-time remote guidance information. 4. As the network services evolve and new network functions are plugged in (virtual or physical), the following evolution steps are applied: a. AR app is updated to collect new data, including new equipment data, and new sensors and new environment information. b. backstage support system is updated with new data management systems, core algorithms etc 5. Periodic or asynchronous reports are produced for human consumption regarding the operation of intelligent maintenance assistant system. 6. A software development kit (SDK) may be exposed to 3rd party developers who may develop new applications to analyse the AR-collected data. This may in turn help operators to provide new value-added applications in the intelligent maintenance assistant system.
Notes on use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.
Reference	

7.29.1 Use case requirements

Critical requirements

- AN-UC029-REQ-001: It is critical that AN enable collection of environment data related to network operation and maintenance using automated techniques such as augmented reality (AR) glasses.
- AN-UC029-REQ-002: It is critical that AN enable analysis of environment data related to network operation and maintenance using cloud and AI techniques.
- AN-UC029-REQ-003: It is critical that AN provide intelligent assistance, rendered using automated techniques such as AR, for network operation and maintenance.

NOTE 1 – The intelligent assistance may be produced using analysis by AI/ML on the collected data from AR.

- AN-UC029-REQ-004: It is critical that AN update the data collection mechanisms and data analysis mechanisms along with the result rendering mechanisms based on the analysis by AI/ML on the collected data from AR and the evolution of the underlay networks.

- AN-UC029-REQ-005: It is critical that AN provide periodic and/or asynchronous updates to humans about the operation of the intelligent assistant system.

Expected requirements

- AN-UC029-REQ-006: It is expected that AN enable exposure of programming capabilities to 3rd party developers for creation of novel applications which can help automated operation and maintenance of network, including evolution and adaptation of network functions.

NOTE 2 – Such novel applications may analyse the data collected using AR, suggest new data collection mechanisms based on gaps in collected data, suggest new analytical methods, or suggest new targets for application of analysis.

7.29.2 Actor interactions and possible components

None.

7.30 Demand forecasting and live service migration methods in edge computing systems

Use case id	FG-AN-usecase-030
Use case description	Virtualization and cloudification of services have enabled automation, flexible placement and programmability to network topology. Efficiency of service delivery can be significantly improved using these techniques. However, there are significant challenges to host Ultra-reliable low-latency communication (URLLC) and massive machine type communications (mMTC) services in 5G, in centralized topologies. Monitoring of networks by telco operators have revealed that network topology is not static and load is not uniform over a long service time. This use case describes a dynamic network topology and service placement using the Genetic Algorithm to analyze and predict services. In addition, an efficient forecasting and live migration methods of service as an application to edge computing systems are introduced. This approach can enable intelligent allocation of operator equipment resources, for providing flexible and efficient topologies. Simulation based analysis of results proved that the network equipment efficiency can significantly be increased by these techniques. The optimization of mobile edge computing network performance for a service by addressing the service placement problem is described below. "Match-making" and analytics service is hosted by REx platform [b-AI4Good-4], exposed as APIs to 3rd party service providers. Services (e.g. gaming) can now utilize the resources at the edge efficiently. Edge network reports the resource status (and other metadata) to the REx platform via the "RExclient" deployed at the edge [b-FGAN-O-013-R1]. 3rd party service provider interfaces with the REx platform via "RExclient" deployed at the service provider [b-FGAN-O-013-R1] for service deployment at the edge. In addition, the following extensions are proposed:

	- REx platform will host "composition" service for controllers (closed loops). - Edge networks will expose APIs to deploy and manage controllers - REx platform will analyse the requirements from applications and manage the composition service towards the edge networks. - The sub/pub mechanism described in [b-FGAN-O-013-R1] is extended to include controller metadata. 1. Registration of MEC on the platform will include controller capabilities. 2. Pub MEC status information to the platform will include controller status. 3. Service provider subscribed to this platform receives information about available MECs along with the controllers. 1. Connection, Discovery and capability exchange (info exchange) between the hosted (prediction) service and clients (e.g.REx server and client) on the network operator side (edge) and the (application) service providers on the ISP side. 2. Data (including traffic characteristics and controller metadata) is measured and analysed to predict the resource utilization and automation at the edge. 3. Placement of services, migration, and composition of corresponding controllers are managed by REx platform.
Notes on use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.
Reference	[b-AI4Good-4], [b-FGAN-O-013-R1]

7.30.1 Use case requirements

<u>Critical requirements</u>

- AN-UC030-REQ-001: It is critical that AN integrate with Edge networks that will expose APIs to deploy and manage controllers

- AN-UC030-REQ-002: It is critical that AN enable analysis of the requirements from applications and manage the composition service towards the edge networks.

- AN-UC030-REQ-003: It is critical that AN enable sub/pub mechanism including controller metadata.

NOTE –Thus, Registration of MEC on the platform will include controller capabilities. Pub MEC status information to the platform will include controller status. Service provider subscribed to this platform receives information about available MECs along with the controllers.

- AN-UC030-REQ-004: It is critical that AN enable Connection, Discovery and capability exchange (info exchange) between the hosted (prediction) service and clients on the network operator side (edge) and the (application) service providers on the ISP side.

- AN-UC030-REQ-005: It is critical that Placement of services, migration, and composition of corresponding controllers are managed by domain specific orchestrators.

7.30.2 Actor interactions and possible components

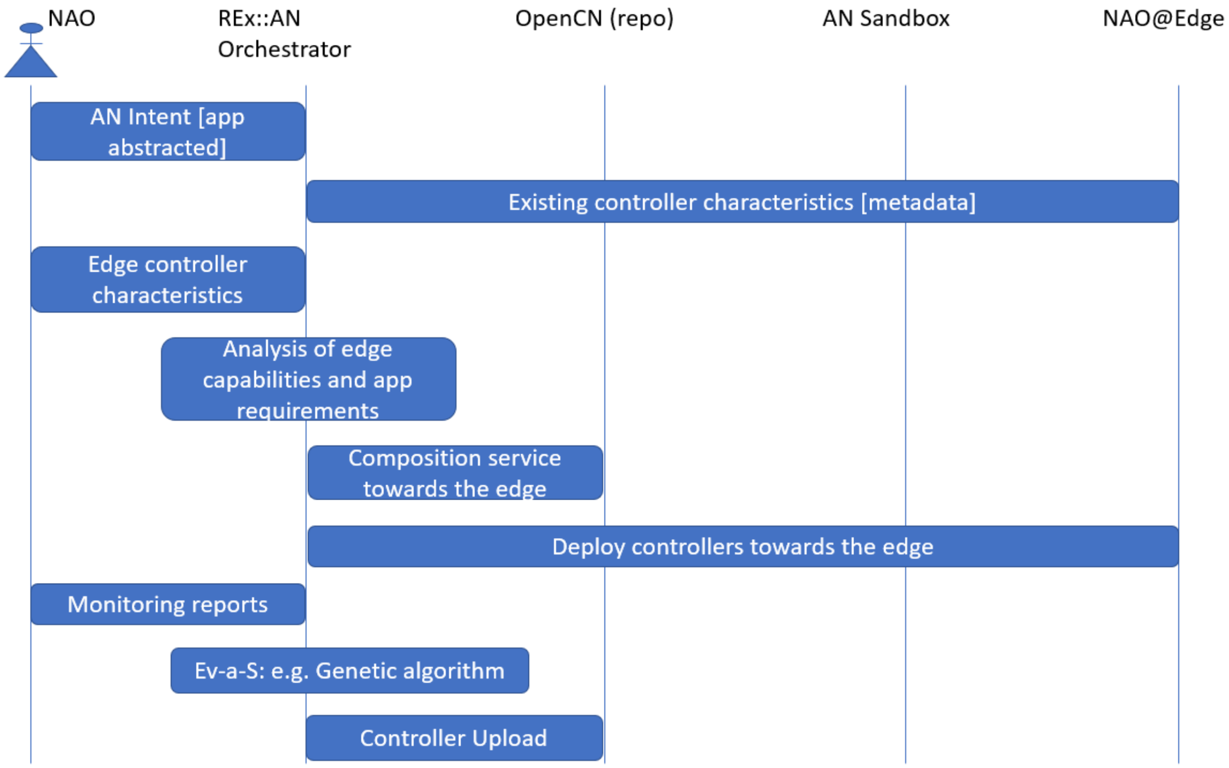


Figure 19: Actor interactions for demand forecasting and live service migration

Actors involved (figure 19):

- Network application orchestrator NAO: This is an actor which manages the network application and orchestrates their lifecycle.
- REx::AN Orchestrator: This actor is an implementation of AN Orchestrator that manages controllers in the network using REx platform.
- OpenCN: This actor stores the controllers.
- AN Sandbox: This actor provides an experimentation platform for controllers. This allows testing and verification of controllers. This may be implemented in the form of an openly accessible sandbox.
- NAO@edge: This is an actor which manages the network application and orchestrates their lifecycle at the edge locations.

7.31 OpenCN: An open repository of intents for controllers and modules

Use case id	FG-AN-usecase-031
Use case description	As controllers/closed loops evolve to solve practical problems in the networks, this use case aims to provide a baseline repository (called OpenCN) of intents for different forms of controllers. As in the case of various opensource repositories and AI/ML marketplaces, an open repository will form a baseline for reusable controllers, provide components for composing and chaining together controllers. In addition, open repo will increase trust in controllers. Metadata related to controllers which describes the controllers and related modules would be enables discovery and other related services like subscription/publication

	of new controllers etc. Metadata also allows to specify guidelines for integrating controllers with service-x and underlay networks. Not only does the baseline intents allow stakeholders to reuse, extend and interoperate controller implementations, but it also allows the development of an ecosystem of services around it – providing customizations (adaptations), integrations (post experimentations) and finally evolutions. NOTE- It is possible to integrate OpenCN with closed loop frameworks e.g. ETSI ZSM [b-ETSI GS ZSM 002]. <u>Steps in the use case are as follows:</u> - Initial version of controllers are formed from intent or composition from modules (by evolution controllers) - These may be stored in the repo labelled as “untested” or candidate controllers - Experimentation (Ex) manager pulls the candidates from the repo and (uses AN sandbox to) evaluate and test and compare the controllers - Evolution (Ev) manager uses the open repo to pull and apply ev strategies - Operational (Op) controllers are stored in the open repo and pulled and deployed in underlay networks by various closed loop automation frameworks. NOTE- Standardized intent formats may be used for storing controllers.
Notes on use case category	Cat 1: Describes a scenario related to core autonomous behaviour itself.
Reference	

7.31.1 Use case requirements

<u>Critical requirements</u>

- AN-UC031-REQ-001: It is critical that AN enable storage of controllers in an open repository.

NOTE – Experimentation (Ex) manager pulls the candidates from the repo and (uses AN sandbox to) evaluate and test and compare the controllers. Evolution (Ev) manager uses the open repo to pull and apply ev strategies. Operational (Op) controllers are stored in the open repo and pulled and deployed in underlay networks by various closed loop automation frameworks.

7.31.2 Actor interactions and possible components

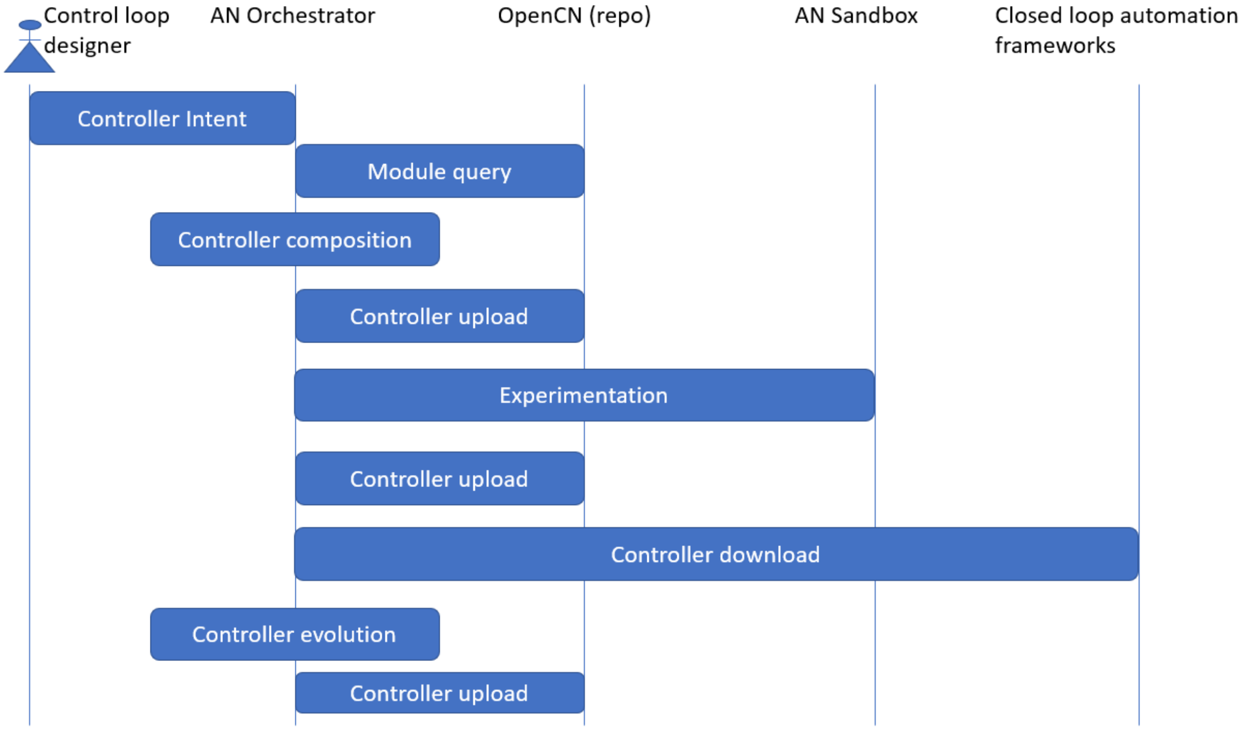


Figure 20: Actor interactions for open repository of intents

Actors involved (figure 20):

- Control loop designer: This actor designs a control loop. The actor can be a human or a machine.
- AN Orchestrator: This actor manages the other AN components in the network.
- OpenCN: This actor stores the controllers.
- AN Sandbox: This actor provides an experimentation platform for controllers. This allows testing and verification of controllers. This may be implemented in the form of an openly accessible sandbox.
- Closed loop automation frameworks: This actor manages closed loops in the networks.

7.32 AI enabled game theory-based mechanism for resource allocation

Use case id	FG-AN-usecase-032
Use case description	With the advent of Internet of Things, large number of devices may be trying to connect to the network. Macro base station may be capacity limited to serve these devices. An alternative is device to device communication, but complexity of algorithms and privacy concerns are limiting factors here. Another option is deploying small base stations known as small cells and each small cell try to serve set of users. Microcell, metro cell, Pico cell and femtocells are collectively known as small cells and network containing all these base stations are termed as "heterogeneous

	networks". To optimize the use of bandwidth, co-channel deployment mechanisms are assumed, i.e. macro cell and small cells will share the same set of sub-channels. Transmission among small cells may cause interference not only among them but also to the macro cell users. Hence, efficient resource allocation algorithms are considerd to mitigate interference among them so as to satisfy QoS constraints of all the devices/users.
	
	Distributed algorithms may be needed to mitigate the interference since there may not be coordination (e.g. X2 interface [b-3GPP 36.420]) among the small cells.
	In this scenario, distributed resource allocation (subchannel allocation and power allocation to each subchannel) are proposed so as to satisfy QoS requirements of all the devices. NOTE- Game theory based model is proposed and to reach equilibrium points both traditional based algorithms like ML algorithms are proposed.
	
	Story-1 for the use case: In an enterprise deployment of small cell, say in a factory floor, low latency is required for certain UEs, e.g. robotic arms or self-guided vehicles such as in Industry 4.0 use cases. Macro cell coverage is limited in such indoor factory floors and interference is a problem. Resource allocation for low-latency UEs in the presence of interference and low coverage by macro is the problem addressed in this "story".
	● Co-existence of UEs that require low latency service and that does not require low latency service
	    o Each small cell can allocate certain channels (power has to be allocated via game theory) to UEs that require low latency (high priority basis) and for other UEs it can allocate channel and power by playing game model
	    o Power allocation is based on UEs rate requirement
	● Serving UEs that all require low latency
	    o The small cell that serves these users can act as a leader and other cells can act as a follower.
	    o Here priority will go to the leader and after allocation of resource by this cell, all others will allocate resources accordingly by playing Stackelberg game model [b-Sankar-2].
	● Given the measurements such as channel gains observed by user on a particular subchannel along with interference plus noise on that channel, it is needed to allocate channel and power per subchannel so as to satisfy user requirements (e.g., rate, low latency)
	    o Here channel allocation matrix and power per subchannel (known as Nash Equilibrium-NE point [b-Sankar-1]) has to be computed iteratively via game theory approach. Three different kind of algorithms such as traditional optimised algorithm, reinforcement learning algorithm and genetic algorithm, are used.
	    o Comparison of the above algorithms for better usage.
	    o Once it is created the data set that consists of various measurements- channel gains, interference plus noise, set of sub channels, user per cell, user requirements along with NE point, the supervised learning (SL) model can be trained (this model sits at the small BSs to obtain subchannel allocation and power per subchannel depends on the local measurements along with other inputs at each small BS).
	    o Compare the output from SL with traditional methods.

<table>
<tr><td></td><td>

Supposing there are multiple users try to access the channel at a particular time and resources are scarce, then it s needed to schedule the users into different frame durations (msec). For this auctioning mechanism is considered to schedule the user in a particular time slot. Utility for each auction is a function of user requirements such as rate, latency etc.
Once the user scheduling is performed, then channel and power per channel according to algorithm described above, are performed.
In the above procedure both time domain and frequency domain scheduling are performed separately.

Story-2: surveillance videos– high UL rate is required for cameras – macro cell coverage and interference is problem. Allocation for rate-required UEs in the presence of interference and low coverage by macro. Priority allocation for UL intensive UEs to achieve QoS.
Story-3: power constrained – wearables etc – privacy – user specific data cannot be exposed to 3rd party – to do analytics, no data should be taken out of the trust zone (enterprise or private network). All AI/ML, analytics etc needs to be done within the private network.
The steps in the use case (related to autonomy) are:
1. Controllers are formed based on, e.g., Intents and/or Evolution (Ev) to optimize resource allocation with various considerations including latency, throughput or privacy preserving analytics.
2. Modelling of inter-controller interaction using game theory: “Players” in the game would be the equivalence classes of controllers. “players” may be selected from the evolvable population.
3. (Initial) Strategies/gains/payoffs are defined and initialized for each equivalence class and game is modelled.
4. Players [controllers] can be cooperating or non-cooperating. They may participate in the game based on trust.
5. Modelling of strategies/gains/payoff for each player (controller) may change based on Evolution.
6. Use experimentation (Ex) to study the model (strategies/gains) which is evolved and the elements of trust in this game can be studied. And the strategies for maximizing the gains can also be investigated, either by cooperation or non-cooperation.
7. Adaptations to be applied to the Controllers are arrived at – in the form of changes to strategies, gains/payoffs and trust. Formation of new “players” may be part of adaptation.
8. Outer loop: Collect the data from the set of solutions, train the AI/ML model, infer the equilibrium from the new input data using trained model.
</td></tr>
<tr><td>Notes on use case category</td><td>Cat 1: Describes a scenario related to core autonomous behaviour itself.</td></tr>
<tr><td>Reference</td><td>[b-Sankar-1], [b-Sankar-2], [b-Ahmad], [b-Al-Turjman], [b-Ciavaglia-2]</td></tr>
</table>

7.32.1 Use case requirements

Critical requirements

- AN-UC032-REQ-001: It is critical that AN enable characterisation of controllers using metadata, which may be updated dynamically based on monitoring of the controllers.

NOTE 1 –The metadata associated with controllers would be used for modelling controllers as players. Example – the closed loops aiming to optimize transmit power and those intending to optimize coverage may be modelled as players in a game.

Expected requirements

- AN-UC032-REQ-002: It is expected that AN enable experimentation with various gaming strategies, payoffs and equilibria with controllers are players.

NOTE 2 – Experimentation may be conducted in the Sandbox and coordinated e.g. by experimentation manager, may result in analysis of strategies, payoffs and equilibria.

- AN-UC032-REQ-003: It is expected that AN enable classification of controllers with respect to trustability of controllers.

NOTE 3 – Parameters related to measurement of trust as applied to controllers and the methods of classification may be out of scope of this particular use case.

- AN-UC032-REQ-004: It is expected that AN enable analysis of experimentation results with AI/ML based techniques.

NOTE 4 – Learnings from AI/ML may be used in optimizing the gaming strategies, payoffs and equilibria.

- AN-UC032-REQ-005: It is expected that AN enable adaptation of controllers with new strategies.

NOTE 5 – Learnings from AI/ML may be used in optimizing the gaming strategies, payoffs and equilibria.

- AN-UC032-REQ-006: It is expected that AN enable derivation and application of different combinations of game theory mechanisms such as auction theory based on the use case specification.

NOTE 6 – Use case specification may be captured and formalized in the form of Intent. Derivation of game theory mechanisms such as auctioneer may use human or automated mechanisms. Application or integration of such mechanisms in underlay networks may use specific architecture and interface considerations such as in Open RAN.

7.32.2 Actor interactions and possible components

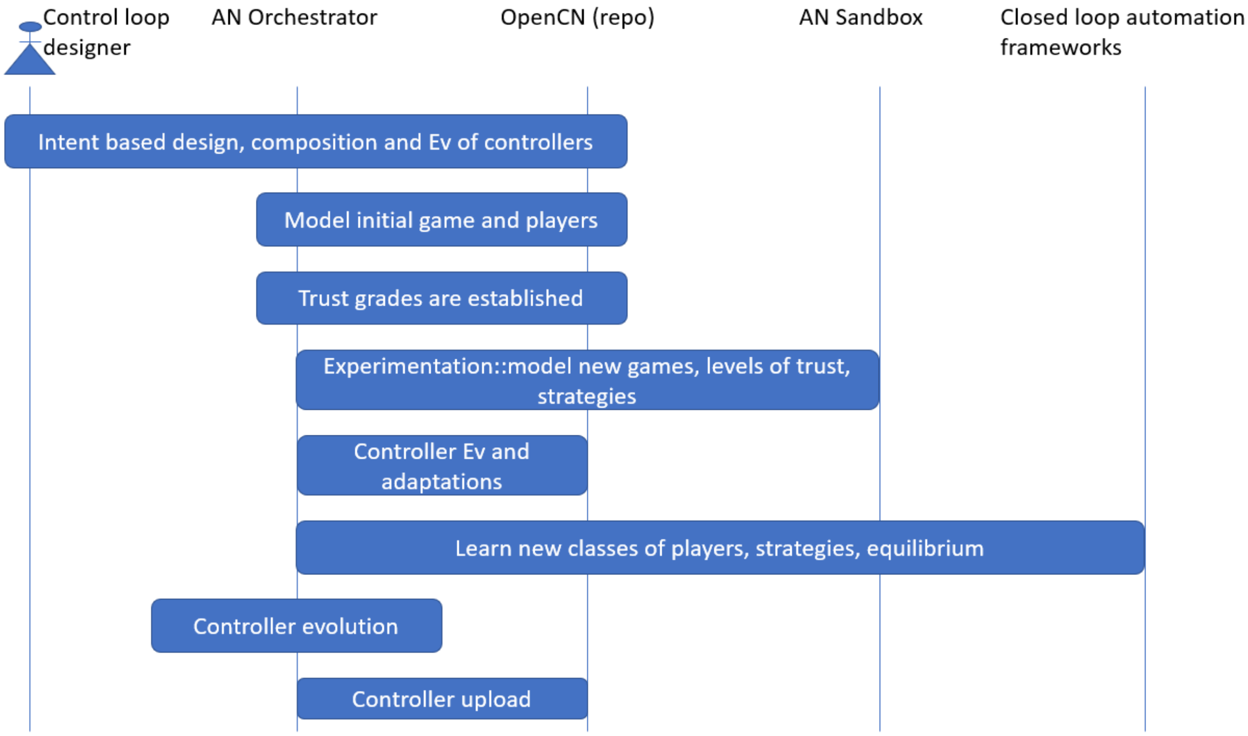


Figure 21: Actor interactions for AI enabled Game theory-based resource allocation

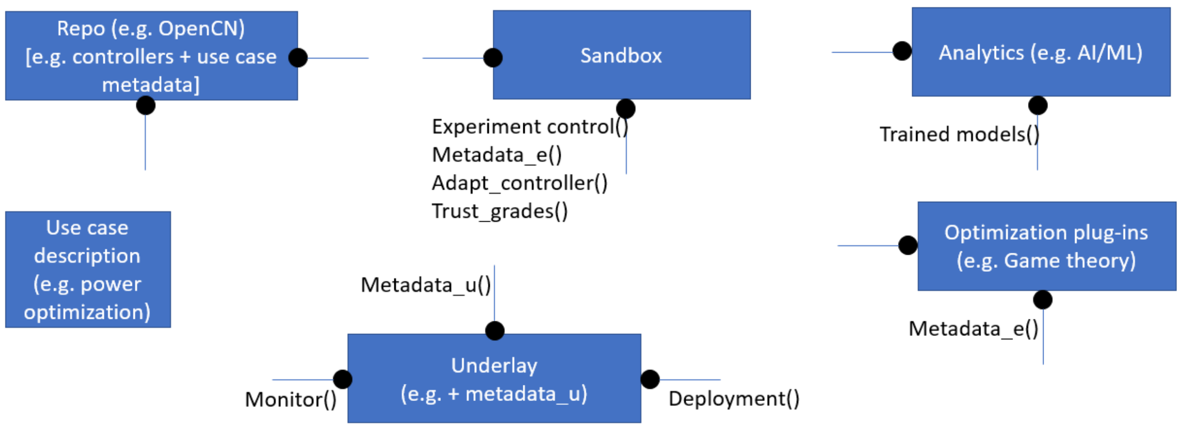


Figure 22: Possible components related to AI enabled Game theory-based resource allocation

Actors involved (figure 21):

- Control loop designer: This actor designs a control loop. The actor can be a human or a machine.
- AN orchestrator: This actor manages the other AN components in the network.
- OpenCN: This actor stores the controllers.
- AN Sandbox: This actor provides an experimentation platform for controllers. This allows testing and verification of controllers. This may be implemented in the form of an openly accessible sandbox.
- Closed loop automation frameworks: This actor manages closed loops in the networks.

Possible key components involved (figure 22):

- Repo: This component provides a repository of controllers.
- Sandbox: This component provides an experimentation platform that allows testing and verification of controllers.
- Analytics: This component provides analytics services (including prediction, inferences) for the AN.
- Optimization plug-ins: This component provides plug-ins which implement specific optimisation algorithms on top of basic schemes. E.g. Game theory based mechanisms for resource allocation.

7.33 Service automation using workflows

Use case id	FG-AN-usecase-033
Use case description	Network automation scenarios in future networks include large scale automation of management of network devices, services and retrieval of operational state data from a network. User specific workflows, along with modularized tasks are one of the mechanisms to achieve this automation. Combined with an interworking of NETCONF [b-IETF RFC 6241] and OpenConfig YANG models [b-Openconfig], vendor native models, and the command line interface (CLI), use specific, customized solutions can be created and dockerized containers can be designed and tested. The workflows are defined using a JSON based domain specific language (DSL) by wiring a set of tasks together. The tasks are either control tasks (fork, conditional, etc.) or application tasks (i.e. encoding a file) that are executed on a remote device. Atomic tasks are chained together into more complex workflows. NOTE 1 - The FRINX Machine [b-FRINX-1] distribution comes pre-loaded with a number of standardized workflows. <u>Steps in this use case are as follows:</u> Step-1: **Create or compose workflow:** Operations or functions on workflows include: In addition to create, the workflow designer can also - edit a workflow - delete a workflow These operations may be achieved using a workflow manager. This may include an API based interface or a GUI based interface. A graphical user interface may be used to create, edit or run workflows and monitor any open tasks. The GUI may also help in explainability. NOTE 2 - This would help to on-board new services (e.g. in underlay networks) and view their status. NOTE 3 - The composing step (from the GUI) may produce output in a generic form (e.g. TOSCA or YANG) and translating this representation of workflows into deployable instances can be using a generic and can take different types of inputs and produce different types outputs. Step-2: **Store in Resource database**: workflow specification and execution data are stored in a resource database.

	Step-3: **Link tasks**: A task corresponds to a worker utilized in the workflow. Tasks in our workflow may receive input parameters and execution logic of our task may be implemented in python functions called worker. Worker tasks may be registered in the main python file “main.py” in the same directory where you just created your worker. NOTE 4 - All workers which one want to use in FRINX Machine must be included in this file. Step-5: **Deploy** workflows: Workflows may be deployed on simulated underlay networks and their performance and benchmarking may be tested and monitored. Workflows may also be deployed on real underlay networks once their performance in the experimentation is satisfactory. Step-6: **Monitor:** The following service states may be mapped to workflows: - experimental state: workflows which are deployed in the sandbox are in experimental state. In combination with simulators, these are tested and experimented upon. - evolutionary state: workflows which are in ev state are selected for evolution, and based on ev strategies, various experiments may be designed for them. - deployed state: workflows which are in deployed state are in combination with service-x, acting upon real underlay networks. They may be monitored for performance and other parameters. - In the context of the use case, controllers (closed loops) are represented as workflows. Modules are modelled as tasks. - Controller specification and module specifications are created using designer, sanity checked and stored in the resource db. - Workflow manager is used to visualize the controllers, monitor and analyse - deploy will link service-x with controllers
Notes on use case category	• Cat 1: describes a scenario related to core autonomous behaviour itself.
Reference	[b-FRINX-1], [b-FRINX-2], [b-TIP 5G]

7.33.1 Use case requirements

- AN-UC033-REQ-001: It is critical that AN enable creation of controllers in a generic format agnostic to the type and characteristics of the underlay network.

NOTE 1 – The generic format may not include deployment specific details. This allows design time flexibility and abstraction.

- AN-UC033-REQ-002: It is critical that AN enable translation of controller specifications from generic format to include underlay network specific details.

NOTE 2 – The translation of controller specifications may include, for example, steps like replacing and augmenting abstracted parameters with underlay network specific parameters and such customizations.

- AN-UC033-REQ-003: It is critical that AN enable storage of generic and underlay network specific representation of controllers in a repository.

<u>Expected requirements</u>

- AN-UC033-REQ-004: It is expected that AN enable different implementations of translation from generic to underlay network specific representation of controllers.

NOTE 3 – Workflow managers which implement the translation may be provided by different vendors.

- AN-UC033-REQ-005: It is expected that AN enable storage of different classes of controllers in a repository.

NOTE 4 – Examples of different classes of controllers are (a) initial representation of controllers, (b) those which are experimented with corresponding metadata (results), (c) those which are deployed with corresponding metadata (from monitoring).

7.33.2 Actor interactions and possible components

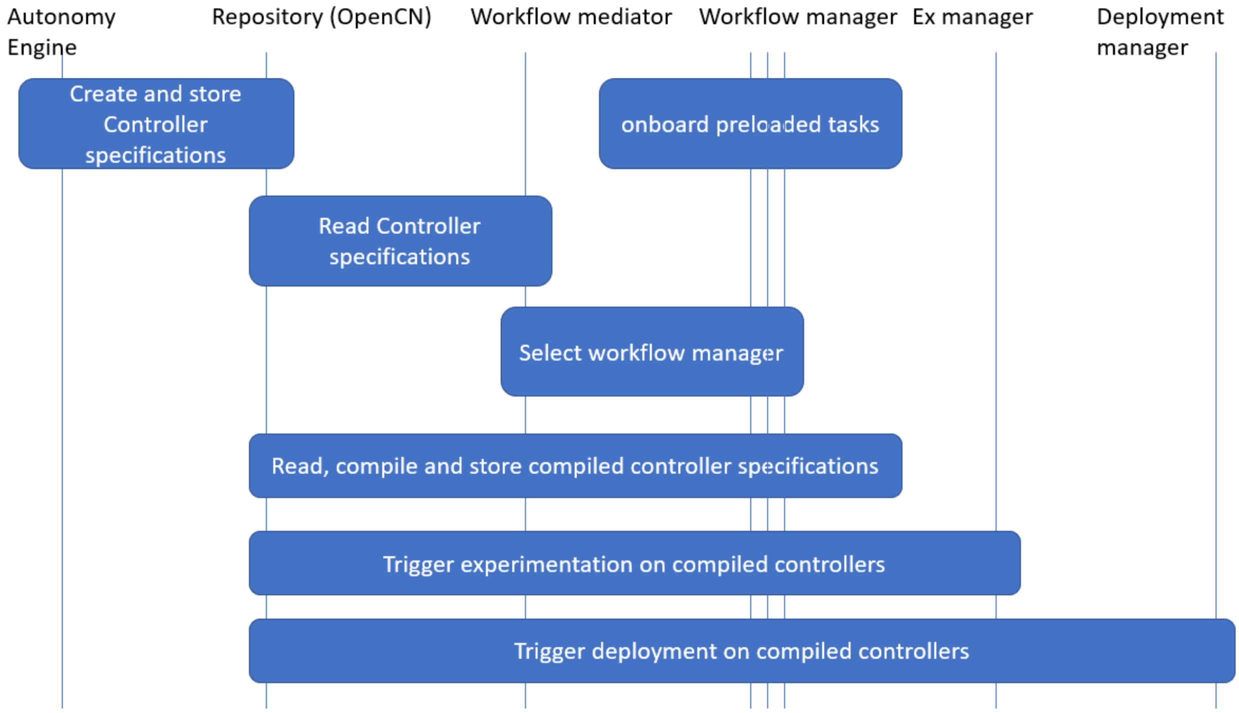


Figure 23: Actor interactions for service automation using workflows

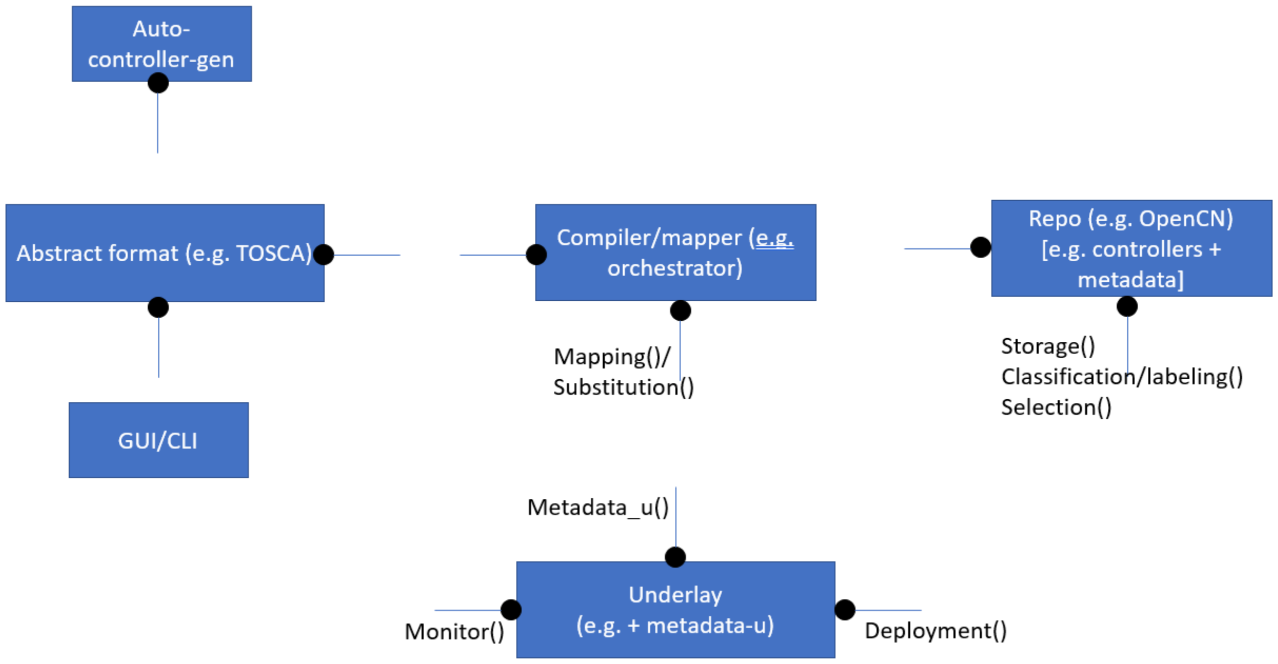


Figure 24: Possible components related to service automation using workflows

Actors involved (figure 23):

- Workflow mediator: This actor is responsible for enabling creation of controllers in a generic format agnostic and interacting with workflow managers adapt them to the type and characteristics of the underlay network.
- Workflow manager: This actor creates workflows from existing or newly created tasks and further triggers the validation and deployment of workflows in different types of underlay networks.
- Deployment manager: This actor deploys the workflows in underlay networks.

Possible key components involved (figure 24):

- Auto-controller-gen: This component is responsible for generating controllers, automated or assisting humans. This component may use inputs from the knowledge base, controller repositories and recommendation systems to generate controllers.
- Abstract format: A declarative format which allows to represent the design of controllers while abstracting the deployment details. The adaptations for specific underlay networks are applied at a later stage than design stage.
- Graphical user interface (GUI) or CLI: A graphical front-end which allows better experience for human users who are designing and managing controllers.
- Compiler or mapper: This component parses the controllers, validates the syntax as well as appropriately maps the abstracted notations and sub-components to their respective underlay network specific detailed representations.

7.34 Disaggregation and placement of in-network programs

Use case id	FG-AN-usecase-034
Use case description	Concepts and tools based on the overarching vision of Software-Defined Networking such as host based (packet processing), middleboxes, NFV (Network

	Function Virtualization), OpenFlow and configurable flow tables, and Programmable switch ASICs are well known. Support for P4 Language consortium [b-ONF P4] has grown over time and it has been central to enabling dataplane programmability. In-Network Programs such as Forward Error Correction (FEC), traffic compression, encapsulation/decapsulation, etc are typical applications in dataplanes. However, the current programming paradigms map dataplane programs 1:1 to devices and resource dedicated to the program, executing on a single target, limiting to the scope of "programmability of the network" and hence creating a mismatch with the overarching vision of SDN. Toolsets which decompose dataplane programs into a suitable mix of dataplanes are needed. Flightplan [b-Flightplan] is an example of such toolsets which helps splitting a P4 program into a set of cooperating P4 programs and maps them to run as a distributed system formed of several, possibly heterogeneous, dataplanes. <u>Steps in this use case are as follows:</u> <u>**Scenario A: "loosely coupled" integration of flightplan with AN**</u> Step-1: Given a controller specification, selection of "in-network controllers", e.g., dataplane programs, is made by the selection controller. Step-2: Flightplan helps in segmenting, planning and allocation/mapping to devices. Step-3: Underlay networks may use other toolsets, e.g. FRINX Machine, or form ZSM managed domains, and/or host their own ways of achieving the above mentioned integration of controllers in their service domains. This forms the "application" (or deployment) side of controller lifecycle. **Scenario B: "tightly coupled" integration of flightplan with AN** Step-1: Flightplan may interface and "collaborate" with AN orchestrator (or different controllers in the AN domain) to achieve the following: a) segmentation of dataplane programs based factors known to AN e.g. resource needs, overall E2E controller deployments across different domains like ZSM or FRINX. b) Diagnose and debug and analyse various "possibilities" for solutions. c) hand-over control between various dataplanes d) placement of controllers e) analysis of performance KPIs Step-2: Recommend, possibly evolutionary, changes to controllers based on flightplan's analysis of constraints and KPIs.
Notes on use case category	• Cat 1: describes a scenario related to core autonomous behaviour itself.
Reference	[b-Flightplan]

7.34.1 Use case requirements

● AN-UC034-REQ-001: It is critical that AN support both loosely coupled and tightly coupled integration mechanisms with underlay networks.

NOTE 1 – For example, underlay networks may integrate workflow management mechanisms like FRINX, closed loop automation frameworks like ETSI ZSM, and in-network dataplane program management mechanisms like Flightplan. Example of loosely coupled integration is passing down intents from AN to Flightplan (which further analyses it to derive program profile). Example of tightly coupled integration is exchange of granular information e.g. current resource allocation status, current KPIs monitored, providing control and visibility over placement of controllers in devices.

NOTE 2 – The loosely coupled integration allows for more autonomy in the underlay network.

- AN-UC034-REQ-002: It is critical that underlay networks support loosely coupled integration mechanisms with AN.

- AN-UC034-REQ-003: It is critical that AN support discovery of capability with respect to level of integration provided by the underlay network.

- AN-UC034-REQ-004: It is critical that AN decide the "application" and deployment of controllers to various underlay networks based on the E2E requirements of the use case.

- AN-UC034-REQ-005: It is expected that underlay networks support tightly coupled integration mechanisms with AN.

7.34.2 Actor interactions and possible components

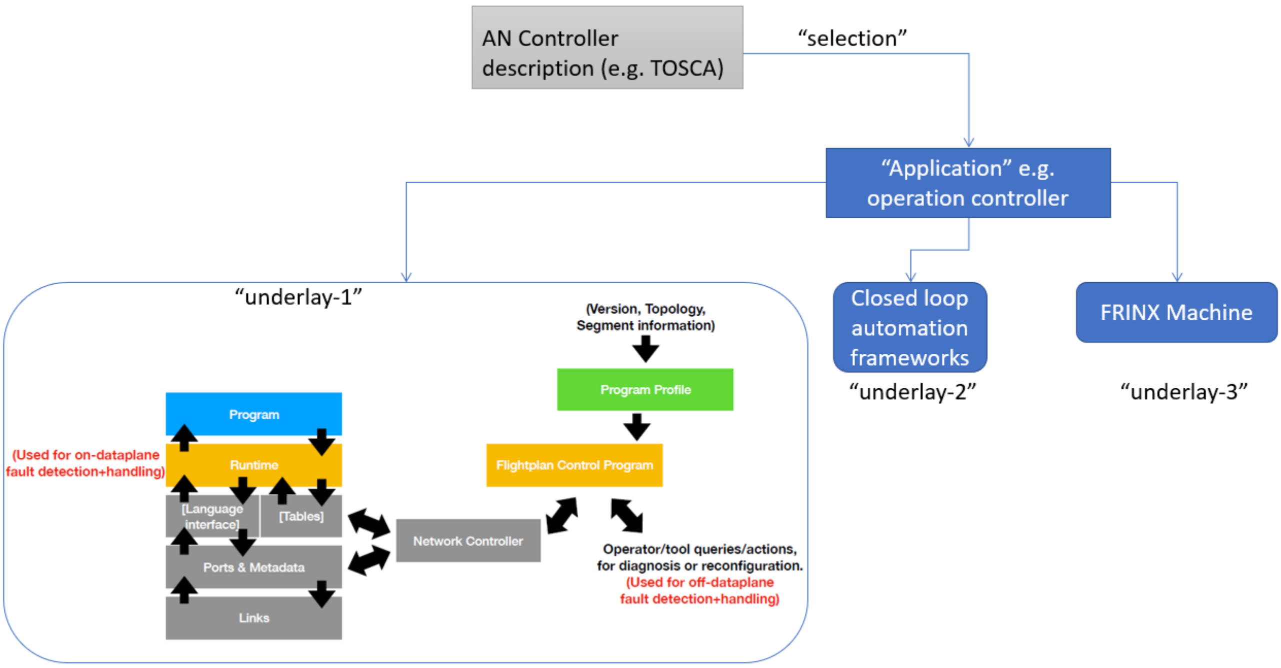


Figure 25: Actor interactions for disaggregation and placement of in-network programs

Actors involved (figure 25):

- Underlay-1: An instance of underlay network, which implements disaggregation and placement of in-network programs.
- Underlay-2: An instance of underlay network which implements closed loop automation frameworks.

- Underlay-3: An instance of underlay network which implements workflow orchestration, e.g., FRINX machine.
- Application: This actor interacts with underlay networks to instantiate and operate the controllers in the various underlay networks. For example, an operation controller which uses APIs exposed by various underlay networks.
- AN controller: This actor selects the underlay network to deploy the controllers as per the use case description.

7.35 A fast and lightweight autoML library (FLAML)

Use case id	FG-AN-usecase-035
Use case description	Currently, selecting learners and hyper parameters for each learner is a tedious and manual task. Fast and Lightweight AutoML (FLAML) [b-FLAML] is a lightweight Python library that finds accurate machine learning models automatically, efficiently and economically. FLAML leverages the structure of the search space to choose a search order optimized for both cost and error. For example, the system tends to propose cheap configurations at the beginning stage of the search, but quickly moves to configurations with high model complexity and large sample size when needed in the later stage of the search. FLAML integrates several simple but effective search strategies into an adaptive system. FLAML can: 1. Optimize with low latency 2. Can handle large datasets 3. Not restricted to fixed set of configs 4. Easy to add new/custom learners 5. Can cold start. Some of the optimizations done are related to: 1. The trials experimentation to be invoked 2. The order to invoke the trials Some of the proposed properties of optimizations in FLAML are: 1. Suitable sample size 2. Resample 3. Fair chance 4. Optimal trial <u>Steps related to use case:</u> 1. Overall use case for autonomous network is specified in the intent to AN. 2. The requirements for ML use case and hence ML intents are derived from the overall use case. 3. Based on the knowledge base [or human guidance], AN configures FLAML with the ML intent. 4. It is left to FLAML to select the learner, do trials, and adapt/optimize the ML model's parameters. 5. It is left to FLAML to tightly or loosely couple with the AN domain, to utilize its experimentation manager, analytics or Sandbox or knowledge base (KB).
Notes on use case category	• Cat 1: describes a scenario related to core autonomous behaviour itself.

Reference	[b-Wang-1], [b-FLAML], [b-Microsoft], [b-Wu-1], [b-Wu-2]

7.35.1 Use case requirements

- AN-UC035-REQ-001: It is critical that AN enable capturing of overall use case in an AN intent and derivation of requirements for ML use case and hence ML intents from the overall use case.
- AN-UC035-REQ-002: It is critical that Based on the knowledge base [or human guidance], AN configures ML optimization tools with the ML intent.

NOTE – FLAML is an example of ML optimization tool. It is left to FLAML to select the learner, do trials, and adapt/optimize the ML model's parameters.

- AN-UC035-REQ-003: It is critical that based on the capabilities of the ML optimization tool, AN tightly or loosely couples with the tool, to utilize its experimentation manager, analytics or Sandbox or knowledge base (KB).

7.35.2 Actor interactions and possible components

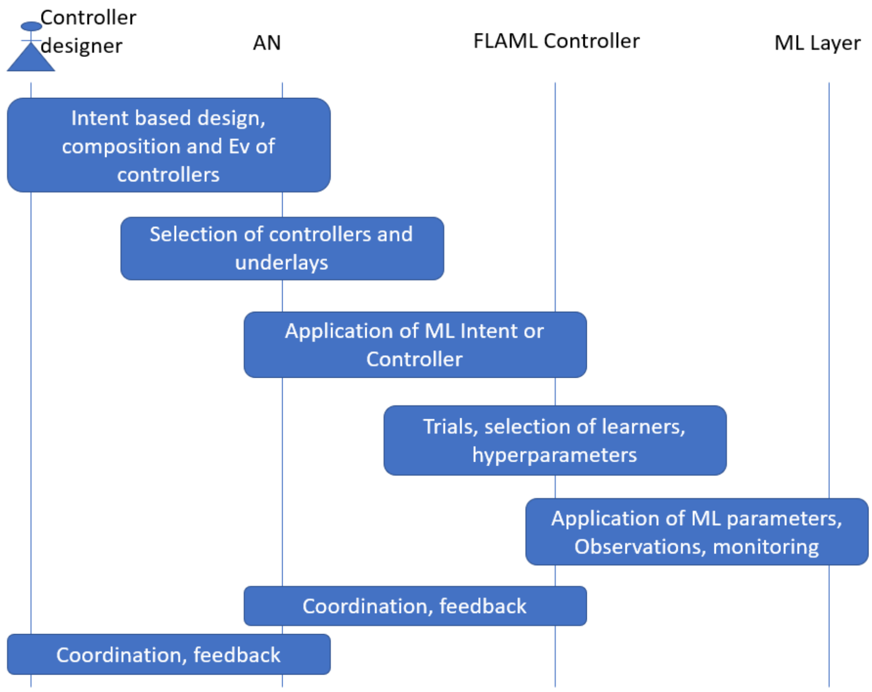


Figure 26: Actor interactions for FLAML

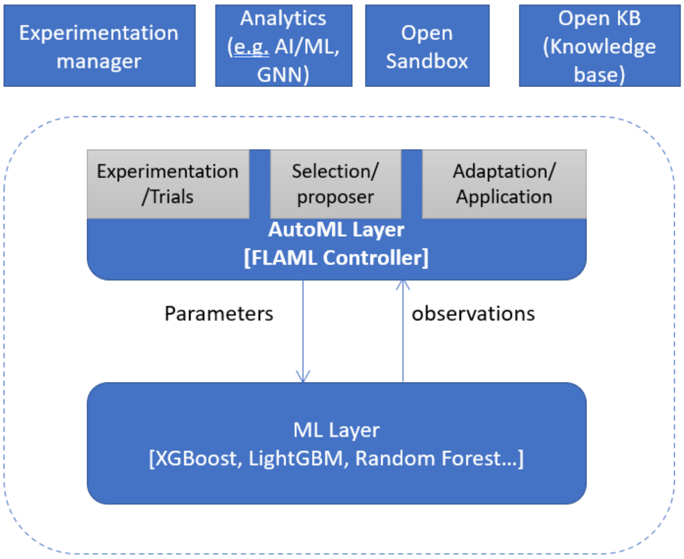


Figure 27: Possible components for FLAML (adapted from [b-FLAML])

Actors involved (figure 26):

- FLAML controller: This actor interacts with ML underlay networks to monitor and optimize the ML models.
- ML layer: This actor is the ML underlay network hosting various types of ML models.

7.36 Connected AI (CAI) testbed: Testbed for 5G connected artificial intelligence on virtualized networks

Use case id	FG-AN-usecase-036
Use case description	5G testbeds are an important alternative to simulators and many have been recently described, emphasizing aspects such as cloud functionalities, management and orchestration. This use case describes Connected AI testbed, which is a 5G mobile network testbed with a virtualized and orchestrated structure using containers, which focuses on integration to artificial intelligence (AI) applications. Two use cases are described, one for RAN slicing and another for the placement of VNFs according to application requirements. Focus is on application of AI to RAN and transport network, including fronthaul and backhaul. The SDN and RAN controllers work as information sources about the network. They also work as agents to dynamically change the mobile and the computer network. An AI agent performs different actions in the testbed according to the application, using the information provided by SDN and RAN controllers to train and execute in test stage its neural networks. The ML workloads are orchestrated along the cluster to provide the AI agent processes. <u>Steps related to the use case:</u> 1. Overall use case for autonomous network is specified in the intent to AN. 2. The requirements for AI Agent and hence ML intents are derived from the overall use case. 3. Based on the knowledge base [or human guidance], AN configures AI Agent with the ML intent. 4. It is left to AI Agent to select the model, and optimize the ML model's parameters. 5. It is left to AI Agent to tightly or loosely couple with the AN domain, to utilize its experimentation manager, analytics or Sandbox or knowledge base (KB).

Notes on use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.
Reference	[b-Nahum]

7.36.1 Use case requirements

● AN-UC036-REQ-001: it is critical that Based on the knowledge base or human guidance, AN configures AI agents.

NOTE – AI agents may in turn select the model and optimize the ML model's parameters in the ML pipeline to be deployed in the underlay network.

● AN-UC036-REQ-002: it is critical that based on the capabilities of the AI agent, AN tightly or loosely couples with the AI agent, to utilize its experimentation manager, analytics or Sandbox or knowledge base (KB).

7.36.2 Actor interactions and possible components

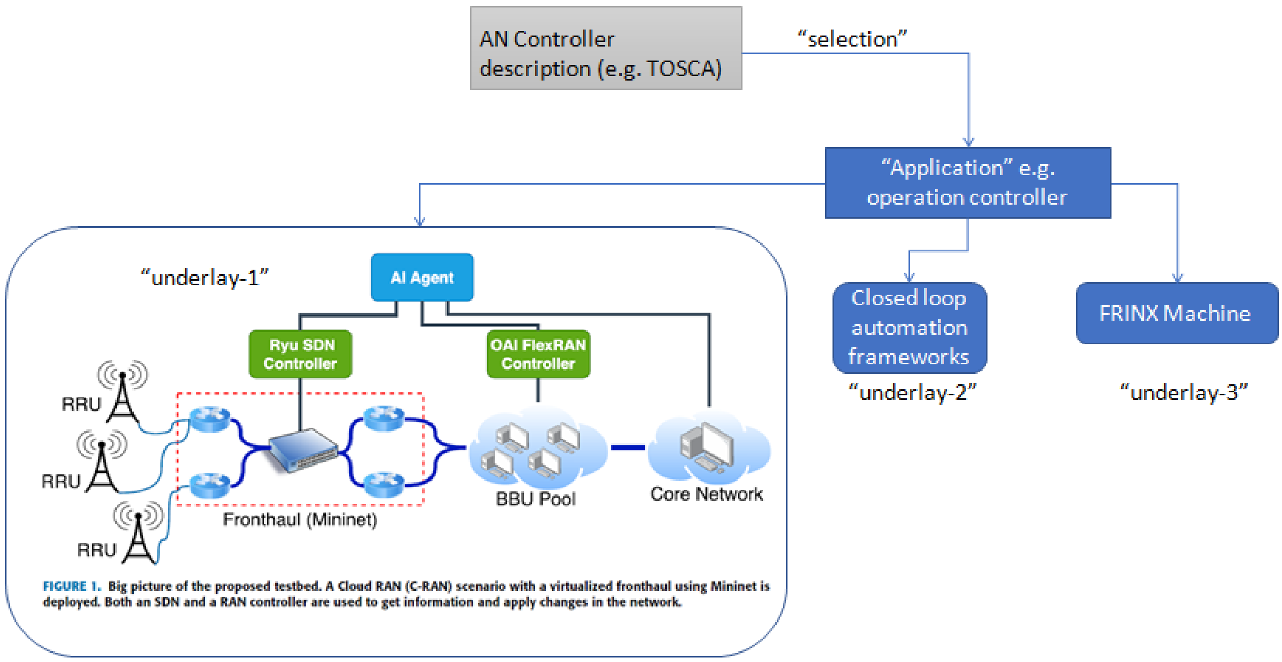


Figure 28: Actor interactions for Connected AI (CAI) testbed

Actors involved (figure 28):

- Underlay-1: An instance of underlay network, which implements which is a 5G mobile network testbed with a virtualized and orchestrated structure using containers, which focuses on integration to artificial intelligence (AI) applications.
- Underlay-2: An instance of underlay network which implements closed loop automation frameworks.
- Underlay-3: An instance of underlay network which implements workflow orchestration e.g. FRINX machine.

- Application: This actor interacts with underlay networks to instantiate and operate the controllers in the various underlay networks. For example, an operation controller which uses APIs exposed by various underlay networks.
- AN controller: This actor selects the underlay network to deploy the controllers as per the use case description.

7.37 Negotiated boundaries in AN for seamless network sharing

Use case id	FG-AN-usecase-037
Use case description	It is possible that various networks are operated and deployed by operators in shared settings. The operators or interested parties like Mobile Virtual Network Operators (MVNOs) share the network for various reasons like reduce CAPEX and OPEX (reduce TCO in general), providing coverage for the users. There are some underlying arrangements for sharing the network like RAN sharing, spectrum sharing, core network and transport network sharing. These arrangements are governed by the agreement between the interested parties. The arrangements possibly could be network slicing, 3GPP based network sharing or some custom vendor solution. The arrangements are rigid and not flexible because of a prior agreement between the parties. The prior agreement creates a boundary between the operators, and any change in the boundary requires a revisit to the agreement. NOTE – Forming new agreements with different operators, withdrawing the existing agreement, and changing service level agreements (SLA) are examples of changes in boundaries between operators. From the AN perspective, the strict boundary is a roadblock towards autonomy. The boundaries is required be flexible, scale in and out dynamically without involving any central authority (here operator/human). The AN decides the scaling of boundaries(in/out) based on the requirement. The control loops, intent or some prediction algorithm can generate requirements for the action of AN. The AN is required to adapt and provide the associated management like policing, billing and other configured elements related to the shared arrangement between the interested parties. The AN is required to have the capability to negotiate the boundaries through the adapted agreements between the interested parties. The negotiation of boundaries means the AN can independently change the agreements in runtime. The following describes an example of load balancing: 1. Consider a scenario of base stations deployed by different operators in a single coverage area. 2. The base station is loaded, and there is a need to balance the load for serving the users with the desired quality of service. 3. In the case of non-AN, the base station can balance the load to different operators base stations. However, it can only scale out to the pre-defined operators in the agreement, and those base stations can also be loaded. To scale out to the other base stations, it needs an operator(human) in the loop who then complete the sharing agreement and provision arrangements. 4. In AN case, the base station can identify the requirement to load balance the users to other base stations. The AN acts independently to adapt the agreement and scale its coverage via other base stations with reduced intervention of operator(human). The AN provision the associated agreements dynamically as per the requirement.

Use case category	Cat 2: Describes a scenario related to application of autonomous behaviour in the network.
Reference	

7.37.1 Use case requirements

Critical requirements

- AN-UC037-REQ-001: It is critical that autonomous networks support seamless autonomous behaviour in case of underlay networks that support scaling across shared resource pools and across administrative domains.

NOTE 1 –Examples of seamless autonomous behaviour may include inclusion (or exclusion) of new controllers or closed loops or services or devices into the scaled AN domain.

NOTE 2 –Examples of shared resource pools are CPU cores, memory, RAN resources like spectrum frequencies, or network resources like ports.

NOTE 3 –Examples of administrative domains are base stations, transport network or core network functions which are owned, maintained and operated by different network operators.

Expected requirements

- AN-UC037-REQ-002: It is expected that autonomous networks reuse existing mechanisms for dynamic management of agreements between different administrative domains of network operators to achieve seamless autonomous behaviour in case of underlay networks that support scaling across shared resource pools and across administrative domains.

NOTE 4 – Examples of existing mechanisms for dynamic management of agreements are block-chain mechanisms or smart contracts.

7.37.2 Actor interactions and possible components

Figure below is split into two parts, figure 29 and figure 30, depicting the scenarios of dynamic changes in base station load in terms of number of served users (part 1) and the corresponding network sharing and SLA negotiations (part 2).

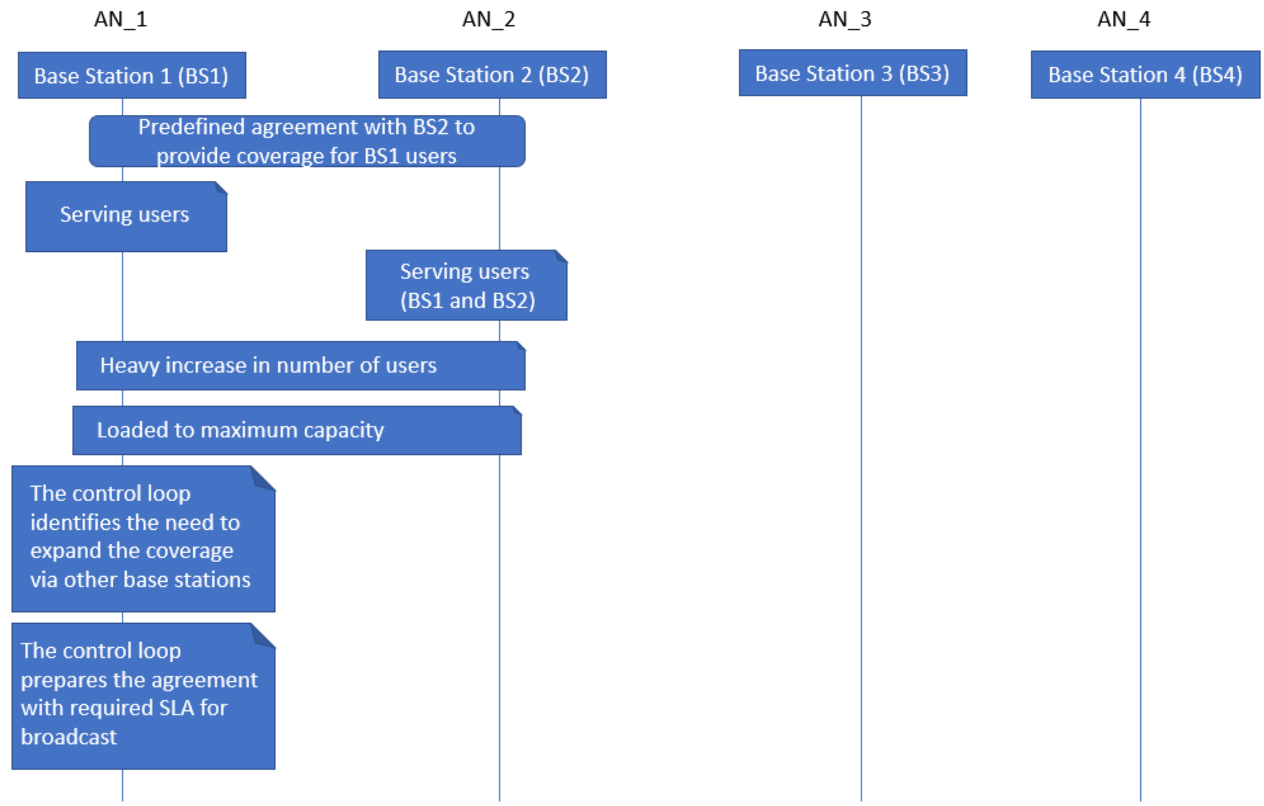


Figure 29: Actor interactions for negotiated boundaries in AN for seamless network sharing: Part-1

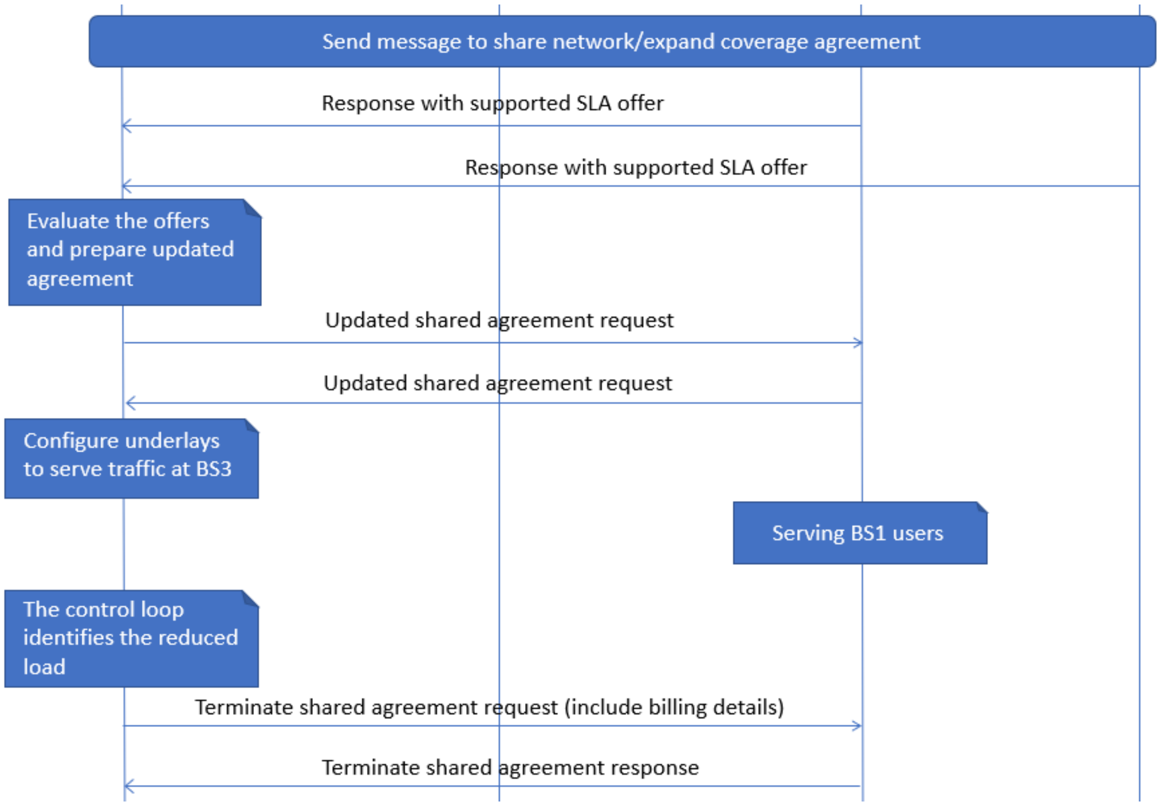


Figure 30: Actor interactions for negotiated boundaries in AN for seamless network sharing: Part-2

Actors involved (figure 30):

- Base station: An actor which provides connectivity to user equipment and hosts controllers to monitor and optimize the service load (in terms of number of users served).

7.38 AN enabled end-to-end supply chain

Use case id	FG-AN-usecase-038
Use case description	Today the supply chain for the networks is highly operator dependent. The operator drives the choice of equipment, procurement, testing and deployment. Planning tools can help to automate the deployment process. Modern advancements include deploying drones for inspection and monitoring of deployed base stations [b-Sharma]. Such inspection by drones is a step towards automation and autonomy. Mission critical applications require the on-demand deployment of the network, with rapid deployment. The supply chain's end-to-end orchestration stage helps in deployment of network equipment, virtual or physical, which brings the network equipment to up and running stage. In an ultra-dense deployment world, the scale of equipment in numbers is very large and hence creates challenges for an operator to manage millions of equipment deployed in different geographies. This brings to the fore, the importance of autonomous networks supported by an end-to-end orchestrated supply chain. In this use case, end-to-end orchestration of the supply chain is studied in the AN context. There is a need to delegate the control from the operator to AN and remove the dependency of the operator. The AN can act independently and decide the orchestration of the supply chain as per the requirements. All physical operations such as power-up of equipment, cabling, but it can certainly schedule those events, notify required teams, and monitor progress. Furthermore, the AN exposes interfaces for operators or orchestrators to feed the vendor lists, equipment lists for bootstrapping and update process. The AN drives the end-to-end supply chain orchestration with minimal intervention of the operator in run time. The following describes an example: 1. The operator feeds the list of vendors or procurement of equipment in AN. It is similar to the source of the image in the docker file. Over a period of time, the AN could learn which equipment from which vendor and rate the vendors based on the performance of their equipment. 2. AN identifies the need to provision network equipment for various reasons like replacement for fault equipment, expansion of the network's capacity, new sites, upgrade, emergency deployment. 3. AN understands the need possibly through some optimization algorithm, direct requirement through intents, other orchestration mechanisms, prediction-based control loops. 4. AN drives the supply chain by procuring, testing the equipment and finally deploying the equipment. The testing requires operator intervention for physical activities. The AN runs the automated test procedures and tallies the results to decide the deployment of equipment. 5. AN configures the underlying equipment and newly deployed equipment for integration to go live.
Use case category	Cat 2: Describes a scenario related to application of autonomous behaviour in the network.

Reference	[b-Sharma]

7.38.1 Use case requirements

Critical requirements

● AN-UC038-REQ-001: It is critical that autonomous networks support interface towards and orchestration of software and hardware inventory management systems in underlay networks.

● AN-UC038-REQ-002: It is critical that autonomous networks support interface towards respective teams (human or bot) or operations centre of the software and hardware components/equipment/systems.

NOTE –Examples of orchestration of inventory management systems are representing and triggering addition, deletion and changes in the inventory. Benefits of such orchestration mechanisms may include proactive and rapid adaptation of controllers and predictive management of supply chains.

7.38.2 Actor interactions and possible components

Figure below is split into two parts, figure 31 and figure 32, depicting the scenarios of monitoring of equipment a done in the network (part 1), and corresponding trigger of inventory management system and deployment, and validations (part 2).

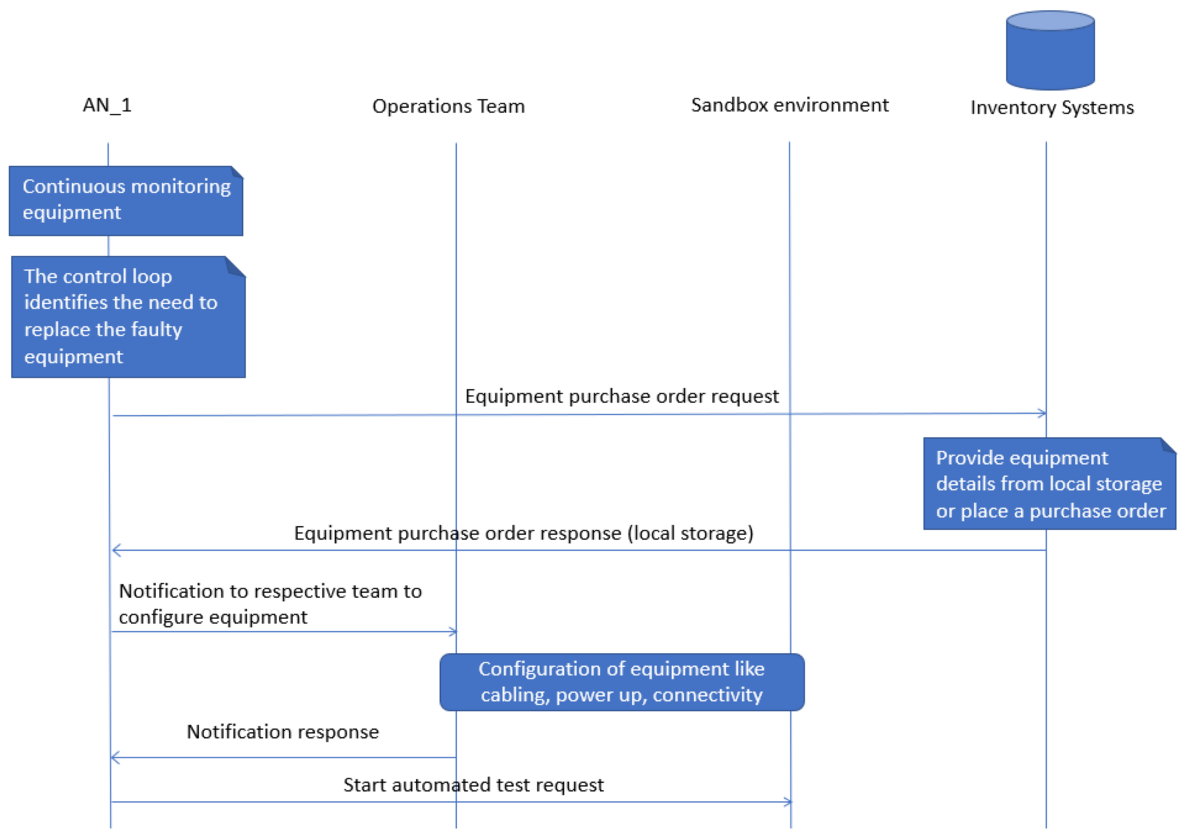


Figure 31: Actor interactions for AN enabled end-to-end supply chain: Part-1

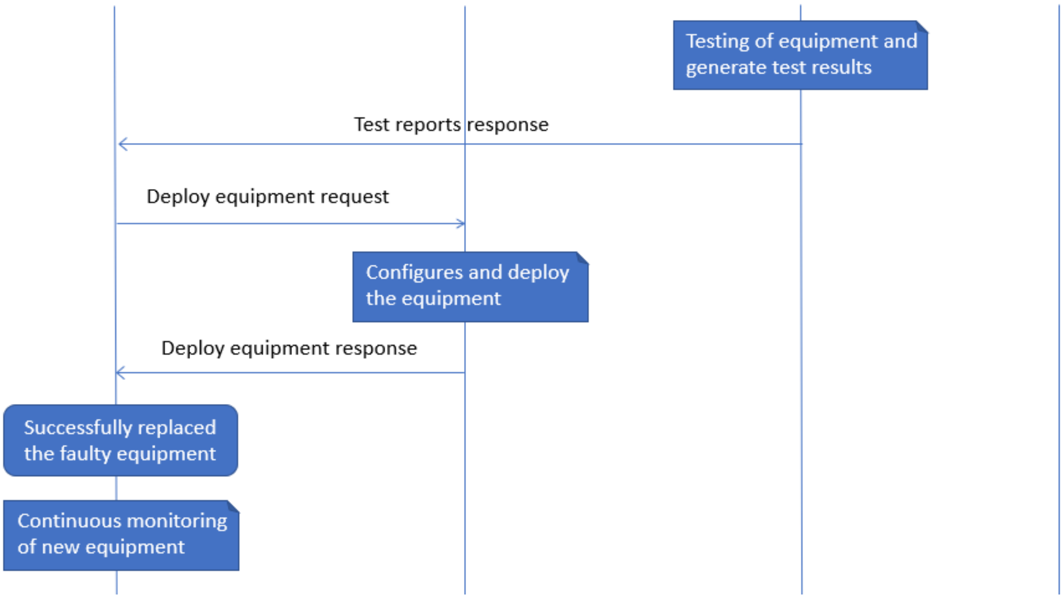


Figure 32: Actor interactions for AN enabled end-to-end supply chain: Part-2

Actors involved (figure 31, figure 32):

- AN_1: An actor or set of actors which provides autonomous network capabilities.
- Operations team: An actor which provides management and operations for equipment and other inventory in the underlay network.
- Inventory systems: An actor which stores and provides the list and details of equipment used in the underlay network.

7.39 Towards openness in AN

Use case id	FG-AN-usecase-039
Use case description	Future networks include heterogeneous AN and control loops are an essential part of AN. There could be various control loops deployed at various endpoints in a heterogeneous AN. The various endpoints could be the edge, cloud, devices like switches, routers, user terminals, customer premises equipment, in general, a control loop possibly in every network equipment. Different vendors develop various control loops for delivering specific functionality of the AN, focusing on improving cost, efficiency, performance, scalability factors. Single vendor solutions which develop the entire AN are rare. Most of the solutions would incorporate a plug and play design. In such a scenario, there is a need for AN to adopt an open design for inclusiveness. Inclusive design helps the AN to plug any control loop from any vendor based on its requirement. AN act independently without involving any central authority to include the third-party control loops. To enable such inclusive design, the AN is required to expose an interface to integrate control loops. The interface requires meta-data about control loops. AN independently manages the lifecycle of the control loops. AN requires the capability to test and configure the control loops. The vendors would use such an interface to develop the application (control loop), knowing that it will help integrate their application. Such a design helps the ecosystem to provide innovative solutions.

	This use case realization includes a control loop application store, where vendors can publish their control loops, and any AN can use it as a repository for control loops. A vendor does not need to publish the control loop in the application store. AN can fetch the control loop from the required location defined in the system. The following describes an example: 1. AN identifies the need to deploy a control loop for a particular service or update the existing control (maybe evolutionary) 2. Based on the requirement, AN search the control loop either in the application store or at a pre-configured location 3. AN download the control loop, test it in a sandbox environment and configures it. The configuration provides control loops providing access to resources. The resources could be data, storage, network, placement, security 4. Once configured, AN deploys the control loop and continuously monitors the performance of the control loop 5. AN also configures the SLA and billing settings for the vendor
Use case category	Cat 2: Describes a scenario related to application of autonomous behaviour in the network.
Reference	

7.39.1 Use case requirements

<u>Critical requirements</u>

- AN-UC039-REQ-001: It is critical that autonomous networks support identifying the need for a control loop, enable the selection, evaluation, integration and monitoring of control loops in different underlay networks.

NOTE – AN may use metadata based selection of control loops from open repositories. Evaluation of control loops may be done using Sandboxes. The control loops are local(internal) or third-party developed.

7.39.2 Actor interactions and possible components

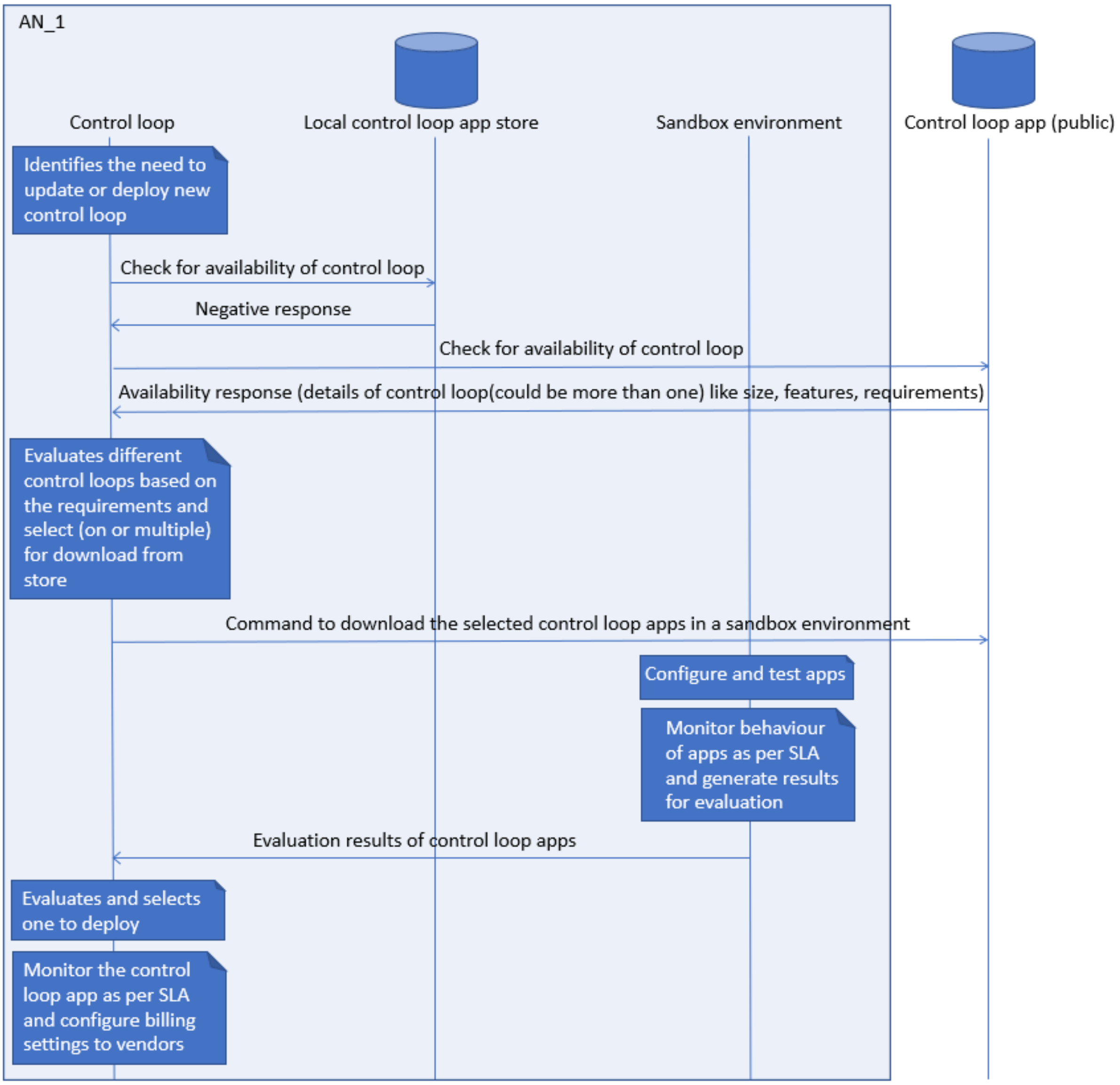


Figure 33: Actor interactions for openness in AN

Actors involved (figure 32, figure 33):

- Control loop: An actor which provides an instance of the controller.
- Local control loop app store: An actor which provides a local instance of openCN in a specific operator underlay network.
- Inventory systems: An actor which provides a public instance of openCN which may serve multiple operators' underlay networks.

7.40 Awareness in AN

Use case id	FG-AN-usecase-040
Use case description	There are heterogeneous networks (HetNet) with networks like RAN, Core, transport and convergence of wireless and wireline of different technologies.

	The evolution of HetNet technologies would comprise control loops in the journey towards AN. Multiple control loops from different vendors and operators surround AN at any given location or endpoint in such a HetNet scenario. AN need to be aware of the surrounding control loops or other AN in general. It could be for various reasons like collaboration, coordination, security, training machine learning model, split learning, offload. AN must have an interface to communicate and interact with other control loops in different AN and control loops within the AN. The interface is required to enable other control loops to be autonomous, discover each other, communicate directly with other control loops without involving any centralized entity. The control loops within the AN may have a centralized entity, but communication with the external control loop is without a centralized entity. The AN is self-sufficient to act independently. Consider the following example of usage of interface: 1. The discovery of nearby control loops and AN is available at edge nodes 2. An edge node AN_1 entity running some application and because of heavy burst load in the requests, it runs short of compute and storage to maintain the desired quality of service 3. The AN_1 node identifies the problem based on the feedback, monitoring, any logic/algorithm. It seeks out help in the form of collaboration to offload the requests using the interface 4. Another AN edge node AN_2 responds with its capability and SLA parameters. The AN_1 evaluates and decide to offload the requests to AN_2. 5. AN_1 and AN_2 uses the interface to configure the routing and policy to divert the traffic to AN_2 and initiate the offload process. 6. AN_1 uses the interface to release all the resources and bill the usage as per the SLA once the load reduces
Use case category	Cat 2: describes a scenario related to application of autonomous behaviour in the network.
Reference	

7.40.1 Use case requirements

<u>Critical requirements</u>

● AN-UC040-REQ-001: It is critical that autonomous networks support peer interaction between different controller instances.

NOTE – Examples of peer interaction are capability discovery and exchange, and resource pooling. Peer interaction may be achieved without involving a centralized entity.

7.40.2 Actor interactions and possible components

Figure below is split into two parts, figure 34 and figure 35, depicting the scenarios of monitoring of traffic load done in the network (part 1) and dynamic offload with corresponding configurations and reconfigurations (part 2).

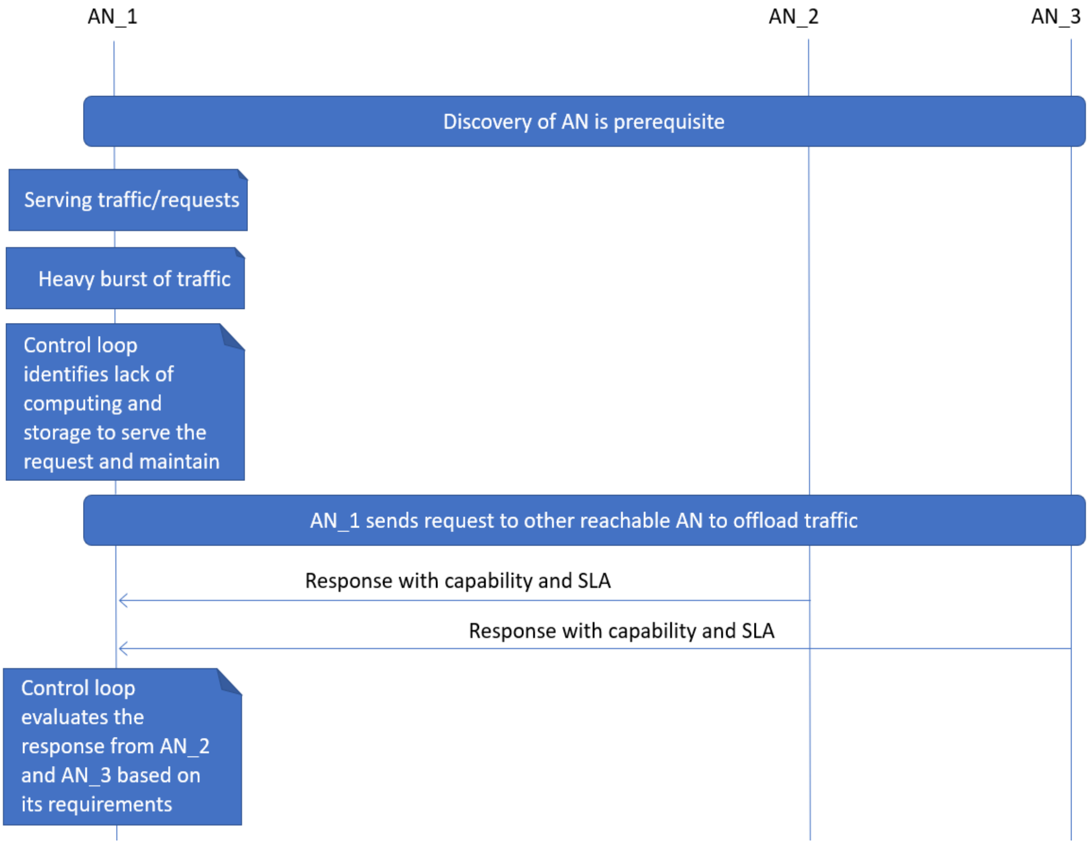


Figure 34: Actor interactions for awareness in AN: Part 1

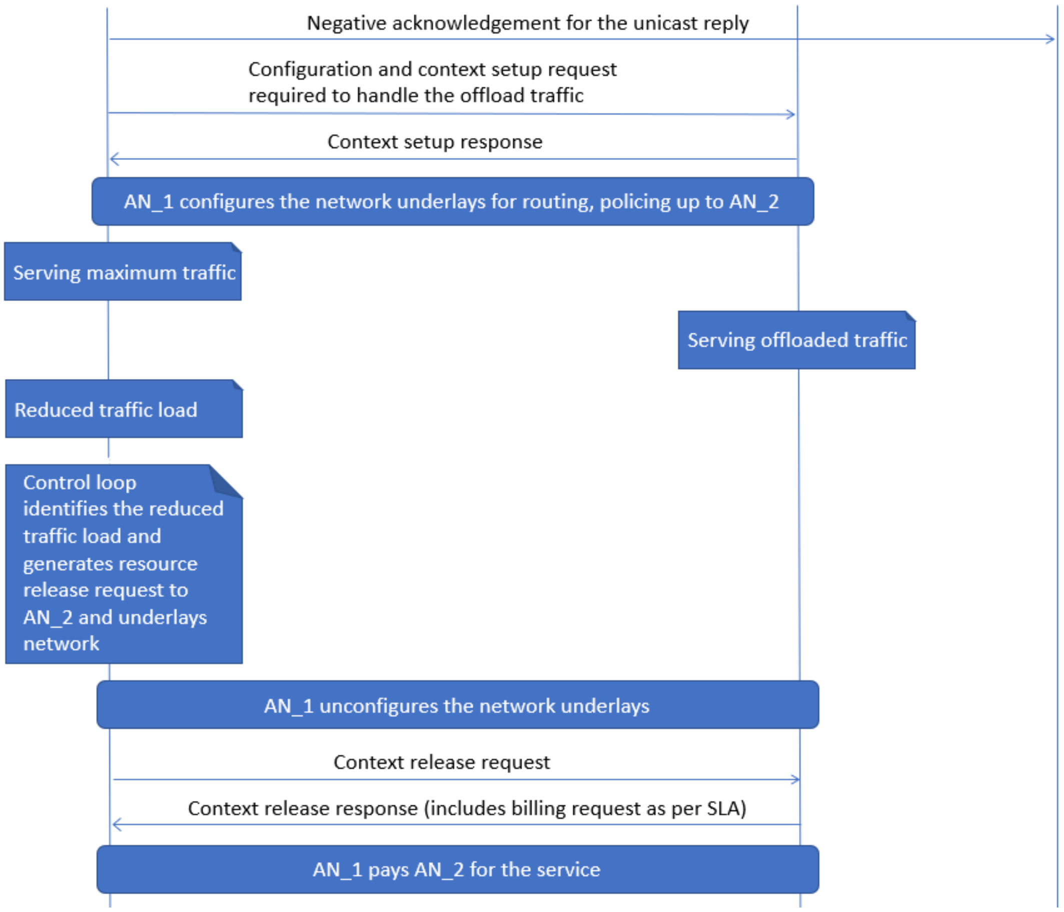


Figure 35: Actor interactions for awareness in AN: Part 2

Actors involved (figure 34, figure 35):

AN_1, AN_2 and other ANs are actors which serve traffic requests, hosts controllers which optimize the traffic load by making offload decisions to other ANs based on capability and SLA negotiations.

Bibliography

[b-3GPP 36.420] 3GPP 36.420 (2022), *Evolved Universal Terrestrial Radio Access Network (E-UTRAN); X2 general aspects and principles*.

[b-3GPP 36.331] 3GPP 36.331 (2022), *Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification*

[b-Ahmad] Ahmad, I., Narmeen, R., Nguyen, L. D. and Ha, D.-B. (2019),*Quality-of-Service Aware Game Theory-Based Uplink Power Control for 5G Heterogeneous Networks*. Springer Journal of Mobile Networks and Applications, Vol.24, No. 2, pp. 556–563.

[b-AI4Good-1] AI for Good Webinar: *Leveraging AI & machine learning to optimize today's 5G radio access network systems and to build the foundation of tomorrow's 6G wireless systems*, 27 November 2020 https://aiforgood.itu.int/events/leveraging-ai-machine-learning-to-optimize-todays-5g-radio-access-network-systems-and-to-build-the-foundation-of-tomorrows-6g-wireless-systems/

[b-AI4Good-2] AI for Good Webinar: *The road towards an AI-native air interface for 6G*, 18 November 2020. https://aiforgood.itu.int/events/the-road-towards-an-ai-native-air-interface-for-6g/

[b-AI4Good-3] AI for Good Webinar: *Machine learning for joint sensing and communication in future millimeter wave IEEE 802.11 WLANs*, 25 May 2021. https://aiforgood.itu.int/events/machine-learning-for-joint-sensing-and-communication-in-future-millimeter-wave-ieee-802-11-wlans/

[b-AI4Good-4] AI for Good Webinar: *Service migration algorithms in distributed edge computing systems*, 9 July 2021 https://aiforgood.itu.int/event/service-migration-algorithms-in-distributed-edge-computing-systems/

[b-Almasan] Almasan, P., Suárez-Varela, J., Badia-Sampera, A., Rusek, K., Barlet-Ros, P., &Cabellos-Aparicio, A. (2019). *Deep Reinforcement Learning meets Graph Neural Networks: An optimal routing use case*. arXiv: Networking and Internet Architecture.

[b-Al-Turjman] Al-Turjman, F., Ever, E. and Zahmatkesh, H. (2019),*Small Cells in the Forthcoming 5G/IoT: Traffic Modelling and*

	Deployment Overview.IEEE Communications Surveys & Tutorials, Vol. 21, No. 1, pp. 28-65.
[b-AN2020]	ITU Webinar on Towards a truly Autonomous Network, 3 November 2020: https://www.itu.int/en/ITU-T/webinars/20201103/Pages/default.aspx
[b-AVA-1]	ITU Technical Report (2013), *Part 2: Vocabulary for ITU-T Focus Group on Audiovisual Media Accessibility (FG AVA).*
[b-AVA-2]	ITU Technical Report (2013), *Part 3: Using audiovisual media – A taxonomy of participation*.
[b-Beyerer]	Beyerer, J., Deserno, T., Straube, S., Tchouchenkov, I. and Wedler, A. (2021), *Kompetent im Einsatz : Variable Autonomie Lernender Système in lebensfeindlichen Umgebungen*. Whitepaper aus derPlattformLernendeSysteme, München.
[b-Biswas]	Biswas, P., Halder, A. Maheshwary, K. and Arjun, S. (2017), *Inclusive Personlization of User Interfaces*. In proceedings of International Conference on Research into DesignICoRD 2017: Research into Design for Communities, Vol. 1, pp 295-306.
[b-Blott]	Blott, M., Preußer, T. B., Fraser, N. J., Gambardella, G., O'brien, K., Umuroglu, Y., Leeser, M. and Vissers, K. (2018), *FINN-R: An end-to-end deep-learning framework for fast exploration of quantized neural networks.* ACM Transactions on Reconfigurable Technology and Systems (TRETS) Vol. 11, No. 3, pp. 1-23.
[b-Bouraqia]	Bouraqia, K., Sabir, E., Sadik, M. and Ladid, L. (2020), *Quality of experience for streaming services: Measurements, challenges and insights*. IEEE Access, Vol. 8, pp. 13341–13361.
[b-Ciavaglia-1]	Ciavaglia, L. (2021), Next-Generation Closed-Loop Automation. IEEE Tutorial, May 17. https://www.slideshare.net/LaurentCiavaglia/nextgeneration-closedloop-automation-an-inside-view
[b-Ciavaglia-2]	Ciavaglia, L., Ghamri-Doudane, S., Smirnov, M., Demestichas, P., Stavroulaki, V.-A., Bantouna, A. and Sayrac, B.(2012), *Unifying Management of Future Networks With Trust*. Bell Labs Technical Journal, Vol. 17, No. 3, pp. 193-212.

[b-Clark] Clark, D. D., Partridge, C., Ramming, J. C. and Wroclawski, J. T. (2003), *A knowledge plane for the internet*. In Proceedings of the 2003 Conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '03, pp. 3-10.

[b-Cortes] Cortes, C., Vapnik, V. (1995), Support-vector networks. Mach Learn 20, 273–297

[b-DISH-AWS] Businesswire (2021), *DISH and AWS Form Strategic Collaboration to Reinvent 5G Connectivity and Innovation*. https://www.businesswire.com/news/home/20210421005315/en/DISH-and-AWS-Form-Strategic-Collaboration-to-Reinvent-5G-Connectivity-and-Innovation

[b-ETSI GR ZSM 009-3] Draft Group Report ETSI GR ZSM 009-3,*Zero-Touch Network and Service Management (ZSM) Closed-loopautomation: Advanced topics*.

[b-ETSI GS ZSM 001] Group Specification ETSI GS ZSM 001 (2019), *Zero-touch network and Service Management (ZSM); Requirements based on documented scenarios*.

[b-ETSI GS ZSM 002] Group Specification ETSI GS ZSM 002 (2019), *Zero-touch network and Service Management (ZSM);Reference Architecture*.

[b-ETSI GS ZSM 013] Draft Group Specification ETSI GS ZSM 013,*Zero-touch network and Service Management (ZSM);Automation of CI/CD for ZSM services and managed services*.

[b-FLAML] FLAML Demo (2021), *Fast and Lightweight AutoML - a brief overview*, https://youtu.be/YRbLmDjrjsc

[b-Flightplan] Flightplan, https://flightplan.cis.upenn.edu/

[b-FRINX-1] FRINX Machine Introduction, https://docs.frinx.io/frinx-machine/getting-started/getting-started-with-frinx-machine.html

[b-FRINX-2] FRINX Github repository, https://github.com/FRINXio.

[b-Hesse] Hesse, T., Peylo, C., Balhmann, C., Elbe, A. Camargo, I. N. and Slusallek, P. (2021), *Potenziale für industrieübergreifendes Flottenlernen KI-Mobilitätsdatenplattform zur Risikominimierung des automatisierten Fahrens*. Whitepaper aus der PlattformLernendeSysteme, München.

[b-IETF RFC 6241] IETF RFC 6241 (2011), *Network Configuration Protocol (NETCONF).*

[b-ISO 9241-129] ISO 9241-129 (2010), *Ergonomics of human-system interaction — Part 129: Guidance on software individualization*.

[b-ISO/IEC 23167] ISO/IEC TS 23167 (2020), *Information technology — Cloud computing — Common technologies and techniques*.

[b-ISO/IEC 24756] ISO/IEC 24756 (2009), *Information technology — Framework for specifying a common access profile (CAP) of needs and capabilities of users, systems, and their environments*.

[b-ITU-T X.1121] Recommendation ITU-T X.1121 (2004), *Framework of security technologies for mobile end-to-end data communications*.

[b-ITU-T Y.3104] Recommendation ITU-T Y.3104 (2018), *Architecture of the IMT-2020 network.*

[b-ITU-T Y.3515] Recommendation ITU-T Y.3515 (2017), *Cloud computing - Functional architecture of Network as a Service*.

[b-ITU-T Y.3525] Recommendation ITU-T Y.3525 (2020), *Cloud computing - Requirements for cloud service development and operation management*.

[b-J.acc-us-prof] Draft Recommendation J.acc-us-prof,*Common user profile format for audiovisual content*.

[b-Jagannath] Jagannath, J., Polosky, N., Jagannath, A., Restuccia, F., and Melodia, T. (2019), *Machine Learning for Wireless Communications in the Internet of Things: A Comprehensive Survey*. Ad Hoc Networks Vol. 93, p. 101913.

[b-Jahromi] Jahromi, H. and Delaney, D. (2018),*An Application Awareness Framework Based on SDN and Machine Learning: Defining the Roadmap and Challenges*. In 13th APCA International Conference on Control and Soft Computing (CONTROLO), pp. 411-416.

[b-Jimenez-Ruiz] Jimenez-Ruiz, E. (2021), *Semantic Web Technologies and Knowledge Graphs*. Teaching notes, University of London.https://github.com/turing-knowledge-graphs/teaching/tree/main/city

[b-KUKA]	KUKA - Robots & Automation (2021), *Robot AI makes robots safer for robot-human collaboration, even without programming*. https://www.youtube.com/watch?v=e0eKrM0_oyQ
[b-Liu]	Liu, L., Hu, H., Luo, Y. and Wen, Y. (2020*), When wireless video streaming meets AI: A deep learning approach*. IEEE Wireless Communications, Vol. 27, No. 2, pp. 127–133.
[b-Mao]	Mao, Q., Hu, F.and Hao Q. (2018), *Deep Learning for Intelligent Wireless Networks: A Comprehensive Survey*.In IEEE Communications Surveys & Tutorials, Vol. 20, No. 4, pp. 2595-2621.
[b-Microsoft]	Microsoft (2021), FLAML - Fast and Lightweight AutoML, https://github.com/microsoft/FLAML
[b-Myklebust]	Myklebust, E. B., Jiménez-Ruiz, E., Chen, J., Wolf, R. and Tollefsen, K. E. (2021), *Prediction of Adverse Biological Effects of Chemicals Using Knowledge Graph Embeddings*. Semantic Web Journal, 0 (0) 1, IOS Press.
[b-Nahum]	Nahum, C. V., Pinto, L., Tavares, V. B., Batista, P., Lins, S., Linder, N. and Klautau, A. (2020), Testbed for 5G Connected Artificial Intelligenceon Virtualized Networks. IEEE Access, Vol. 8, pp. 223202-223213.
[b-O-RAN.WG1.O-RAN-Arch]	O-RAN Architecture Description
[b-O-RAN.WG3.E2GAP]	O-RAN Working Group 3 Near-Real-time RAN Intelligent Controller Architecture & E2 General Aspects and Principles.
[b-Scarselli]	Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M. and Monfardini, G., (2009)*The Graph Neural Network Model*, in IEEE Transactions on Neural Networks, vol. 20, no. 1, pp. 61-80
[b-O'Shea-1]	O'Shea, T. J. and Hoydis, J. (2017), *An Introduction to Deep Learning for the Physical Laye*r. IEEE Transactions on Cognitive Communications and Networking, Vol. 3, No. 4, pp. 563–575.
[b-O'Shea-2]	O'Shea, T. J., Roy, T. and Clancy, T. C. (2018), *Over-the-Air Deep Learning Based Radio Signal Classification*. IEEE Journal of Selected Topics in Signal Processing, Vol. 12, No. 1, pp. 168–179.
[b-ONF]	The Open Networking Foundation (ONF) (2021), *The Open Networking Foundation, AirHop, Facebook and Telecom*

Infra Project Demonstrate First O-RAN Aligned, xApp-Powered Open RAN Solution. https://opennetworking.org/news-and-events/press-releases/the-open-networking-foundation-airhop-facebook-and-telecom-infra-project-demonstrate-first-o-ran-aligned-xapp-powered-open-ran-solution/?utm_source=ONF+Master+List&utm_campaign=b8631125bb-Edge+Spotl

[b-Openconfig] OpenConfig Data models and APIs, https://www.openconfig.net/projects/models/

[b-ONF P4] Open Networking Foundation, *P4 Open Source Programming Language*, https://p4.org/

[b-Pierucci] Pierucci, L. (2015), *The quality of experience perspective toward 5g technology*. IEEE Wireless Communications, Vol. 22, No. 4, pp. 10–16.

[b-Q.IMT2020-PIAS] Draft Recommendation ITU-T Q.IMT2020-PIAS, *Protocol for providing intelligent analysis services in IMT-2020 network*.

[b-Rusek-1] Rusek, K., Suárez-Varela, J., Mestres, A., Barlet-Ros, P. and Cabellos-Aparicio, A., (2019),*Unveiling the potential of Graph Neural Networks for network modeling and optimization in SDN*. In Proceedings of the 2019 ACM Symposium on SDN Research (SOSR '19), pp. 140-151

[b-Rusek-2] Rusek, K., Suárez-Varela, J., Almasan, P., Barlet-Ros, P. and Cabellos-Aparicio, A. (2020), *RouteNet: Leveraging Graph Neural Networks for network modeling and optimization in SDN*

[b-Sankar-1] Sankar, V. U. and Sharma, V. (2016), *QoS Provisioning for Multiple Femtocells via Game Theory*. Elsevier Journal of Computer Networks, Vol 102,pp.70-82.

[b-Sankar-2] Sankar, V. U. and Sharma, V. (2014), *Pricing and Power allocation in Femtocells using Stackelberg Game Theory*.In: International Conference on Signal Processing and Communications (SPCOM), Jul 22-25, Banaglore, India.

[b-SCALE-Sim] SCALE-Sim tutorial – ASPLOS 2021, https://scalesim-project.github.io/tutorials-2021-asplos.html

[b-Sharma] Sharma, R. (2020), *Rakuten Mobile to Use Drones to Carry Out Mobile Base Station Inspections*.

https://www.thefastmode.com/technology-solutions/16814-rakuten-mobile-to-use-drones-to-carry-out-mobile-base-station-inspections

[b-TIP 5G] TIP 5G Open Automation: Vision of Reality, July 2021. https://hopin.com/events/tip-5g-open-automation-vision-of-reality

[b-TOSCA-Simple] TOSCA Simple Profile in YAML Version 1.3, OASIS Standard, 26 February 2020

[b-Turing] The Alan Turing Institute (2021), *Knowledge graphs*. https://www.turing.ac.uk/research/interest-groups/knowledge-graphs

[b-Umuroglu] Umuroglu, Y., Akhauri, Y., Fraser, N. J. and Blott, M. (2020),*LogicNets: Co-Designed Neural Networks and Circuits for Extreme-Throughput Applications*. In 30th International Conference on Field-Programmable Logic and Applications (FPL), pp. 291-297

[b-Wang-1] Wang, C., Wu, Q., Weimer, M. and Zhu, E. (2021), *FLAML: A Fast and Lightweight AutoML Library*. In Fourth Conference on Machine Learning and Systems (MLSys 2021)

[b-Wang-2] Wang, C., Wu, Q., Huang, S. and Saied, A. (2021), *Economical Hyperparameter Optimization With Blended Search Strategy*. In the Ninth International Conference on Learning Representations (ICLR 2021).

[b-Wu-1] Wu, Q., Wang, C. and Huang, S. (2021), *Frugal Optimization for Cost-related Hyperparameters*. In Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, No. 12: AAAI-21.

[b-Wu-2] Wu, Q., Wang, C., Langford, J., Mineiro, P. and Rossi, M. (2021), *ChaCha for Online AutoML*. In Proceedings of the 38th International Conference on Machine Learning, PMLR 139, pp. 11263-11273

[b-Xilinx] Xilinx (2021), *Accelerate Your AI-Enabled Edge Solution with Adaptive Computing, Introducing Adaptive System-on-Modules (SOMs)*.

____