Deliverable 4.1

State-of-the-Art Review and Initial Definition of BeGREEN O-RAN Intelligent Plane, and AI/ML Algorithms for NFV User-Plane and Edge Service Control Energy Efficiency Optimization

December 2023

Co-funded by
the European Union

6GSNS

Contractual Date of Delivery:	**November 30, 2023**
Actual Date of Delivery:	**December 31, 2023**
Editor(s):	**Juan Sánchez-González (UPC)**
Author(s)/Contributor(s):	**Juan Sánchez-González, Jordi Pérez-Romero, Oriol Sallent, Anna Umbert (UPC)** **Miguel Catalan-Cid, Esteban Municio, Jorge Pueyo, Pau Tomas (i2CAT)** **German Castellanos, Revaz Berozashvili, Simon Pryor (ACC)** **J. Xavier Salvat, Jose A. Ayala-Romero, Lanfranco Zanzi (NEC)** **Joss Armstrong (LMI)** **Jesús Gutiérrez (IHP)** **Mir Ghoraishi (GIGASYS)**
Work Package	**WP4**
Target Dissemination Level	**Public**

This work is supported by the Smart Networks and Services Joint Undertaking (SNS JU) under the European Union's Horizon Europe research and innovation programme under Grant Agreement No 101097083, BeGREEN project. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or SNS-JU. Neither the European Union nor the granting authority can be held responsible for them.

Revision History

Revision	Date	Editor / Commentator	Description
0.1	2023-05-29	Juan Sánchez-González (UPC)	Initial version and initial ToC
0.2	2023-07-29	Juan Sánchez-González (UPC), Miguel Catalan-Cid (i2CAT)	Revised ToC and initial assignments. Initial contributions (chapter 2)
0.3	2023-09-29	I2CAT, LMI, UPC, BT	Initial contributions (chapter 2)
	2023-10-10	German Castellanos (ACC)	Contributions of i) dRAX solution, ii) TeraVM, iii) EU projects related, iv) Near-RT RIC
	2023-10-10	NEC	Contributions section 2.3.4
0.4	2023-10-22	REL	Contribution section 2.3.2.2
0.5	2023-10-23	I2CAT, LMI, UPC, NEC	Initial contributions chapter 3
0.6	2023-10-24	All	Complete version for internal review
0.7	2023-12-04	Review chapter 2 (ACC) Review chapter 3 (NEC, LMI)	Chapter review.
0.8	2023-12-11	UPC, I2CAT	Top to bottom review
0.9	2023-12-13	Keith (BT)	Style and grammar review
0.91	2023-12-30	Jesús Gutiérrez (IHP) Mir Ghoraishi (GIGASYS)	Final review and proof reading
1.00	2023-12-31	Simon Pryor (ACC)	Submission to the EC

Table of Contents

List of Figures

List of Tables

List of Acronyms

3GPP	3rd Generation Partnership Project
5GC	5G Core
5GNR	5G New Radio
AAL	Acceleration Abstraction Layer
AI	Artificial Intelligence
AIF	AI Functions
AIMM	AI-enabled Massive MIMO
AMF	Access and mobility Management Function
AP	Average Precision
API	Application Programming Interface
ASM	Advanced Sleep Mode
ASN	Abstract Syntax Notation
AWS	Amazon Web Services
B5G	Beyond 5G
BaaS	Backend-as-a-Service
BS	Base Station
BSC	Base Station Controller
BYOD	Bring Your Own Device
CN	Core Network
CNN	Convolutional Neural Network
COTS	Commercial Off-The-Shelf
CP	Control Plane
CPU	Central Processing Unit
CQI	Channel Quality Indicator
CU	Central Unit
CU-CP	Central Unit Control Plane
CUPS	Control User Plane Separation
CU-UP	Central Unit User Plane
CVE	Common Vulnerability and Exposure
D2D	Device to Device
DBSCAN	Density Based Spatial Clustering of Applications with Noise
DCCF	Data Collection Coordination Function
D-DQN	Decaying DQN
DL	Downlink
DME	Data Management and Exposure
DMS	Deployment Management Services
DNN	Deep Neural Network
DPD	Digital Pre-Distortion
DQN	Deep Q Network
DRL	Deep Reinforcement Learning
DSCH	Downlink Shared Channel
DSS	Dynamic Spectrum Sharing
DU	Distributed Unit
DUT	Devices Under Test
DV	Data Volume
E2AP	E2 Application Protocol

E2SM	E2 Service Model
EC	Energy Consumption
EI	Enrichment Information
EMF	Electromagnetic Field
eNB	eNodeB
EPC	Evolved Packet Core
ET	Envelope Tracking
FaaS	Function as a Service
FAPI	Function Application Platform Interface
FCAPS	Fault, Configuration, Accounting, Performance, Security
FEC	Forward Error Correction
FNN	Feedforward Neural Network
FOCOM	Federated O-Cloud Orchestration and Management
FR	Frequency Range
gNB	gNodeB
GNSS	Global Navigation Satellite System
GPR	Gaussian Process Regression
GPU	Graphics Processing Unit
GTP	General Packet Radio Service Tunneling Protocol
GUI	Graphic User Interface
HA	High Availability
HARQ	Hybrid Automatic Repeat request
HOF	Handover Failure
HPO	Horizontal Pod Autoscaling
HTTP	HyperText Transfer Protocol
HW	Hardware
IAB	Integrated Access and Backhaul
ICS	Information Coordinator Service
IMS	Infrastructure Management Services
IoT	Internet of Things
IoU	Intersection over Union
IPC	Instructions Per Cycle
ITU	International Telecommunication Union
JSAC	Joint Sensing and Communications
KPI	Key Performance Indicator
KPM	Key Performance Metrics
L2	Layer 2
LPU	Logical Processing Unit
LSTM	Long Short-Term Memory
LTE	Long Term Evolution
MAC	Medium Access layer
mAP	Mean Average Precision
MAPE-K	Monitor-Analyze-Plan-Execute over a shared Knowledge
MCS	Modulation and Coding Scheme
MDT	Minimisation of Drive Tests
MEC	Multi-Access Edge Computing

MFAF	Messaging Framework Adapter Function
MISE	Memory-Interference Induced Slowdown Estimation
ML	Machine Learning
MLOps	Machine Learning Operations
MME	Model Management and Exposure
mMIMO	Massive Multiple Input Multiple Output
mmWave	Millimetre Wave
MNO	Mobile Network Operator
MPKI	Misses per 1000 instruction
MSE	Mean Squared Error
MVA	Mobile Video Analytics
MVNO	Mobile Virtual Network Operator
NAS	Non-Access Stratum
Near-RT	near Real-Time
NF	Network Function
NFO	Network Function Orchestrator
NFV	Network Function Virtualisation
NGAP	Next Generation Application Protocol
NGMN	Next Generation Mobile Networks
NI	Network Intelligence
NIC	Network Intelligent Component
NIC	Network Interface Card
NIF	Network Intelligent Function
NIO	Network Intelligence Orchestrator
NIP	Network Intelligence Plane
NIS	Network Intelligent Service
N-MAPE-K	Network MAPE-K
non-RT	Non-Real-Time
NR	New Radio
ns-3	Network Simulator 3
NSA	Non-Stand Alone
NSAP	Network and Service Automation Platform
NSSMF	Network Slice Subnet Management Function
NUC	Next Unit of Computing
NWDAF	Network Data Analytics Function
OAI	Open Air Interface
OAIC	Open AI Cellular
OAM	Operation and Maintenance
OFDM	Orthogonal Frequency Division Multiplexing
O-FH	O-RAN Front-Haul
OMEC	Open Mobile Evolved Core
ONAP	Open Network Automation Platform
ONF	Open Networking Foundation
ONOS	Open Network Operating System
O-RAN	Open RAN
OS	Operating System
OSA	OAI Software Alliance

OSC	O-RAN Software Community
OTIC	Open Testing & Integration Centre
PAWR	Platforms for Advanced Wireless Research
PDCCH	Physical Downlink Control Channel
PDCP	Packet Data Convergence Protocol
PDSCH	Physical Downlink Shared Channel
PDU	Packet Data Unit
PGW	Packet Data Network Gateway
PoE	Power over Ethernet
PHY	Physical layer
PLMN	Public Land Mobile Network
PM	Performance Management
PoC	Proof of Concept
PoE	Power over Ethernet
PRB	Physical Resource Block
PUCCH	Physical Uplink Control Channel
PUSCH	Physical Uplink Shared Channel
PRB	Physical Resource Block
QoS	Quality of Service
RACH	Random Access Channel
rApp	Radio access network Applications
RAN	Radio Access Network
RAT	Radio Access Technology
RIC	RAN Intelligent Controller
RIS	Reconfigurable Intelligent Surface
RISA	RSI Actuator
RISC	RIS Controller
RISO	RIS Orchestrator
RL	Reinforcement Learning
RLC	Radio Link Control
RLF	Radio Link Failure
RLP	RU Load Predictor
RN	Relation Network
RNC	Radio Network Controller
RNN	Recurrent Neural Network
RRC	Radio Resource Control
RRM	Radio Resource Management
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RU	Radio Unit
RUE	Relay UE
SA	Standalone
SCTP	Stream Control Transmission Protocol
SCTP-C	Stream Control Transmission Protocol Client
SCTP-S	Stream Control Transmission Protocol Server
SDAP	Service Data Adaptation Protocol

SD-CN	Software-Defined Core Network
SDK	Software Development Kit
SDR	Software Defined Radio
SD-RAN	Software-Defined Radio Access Networking
SEP	Service Enablement Platform
SHAP	Shapley Additive Explanation values
SINR	Signal to Interference and Noise Ratio
SISO	Single Input Single Output
SLA	Service Level Agreement
SM	Service Model
SME	Service Management and Exposure
SMF	Session Management Function
SMO	Service Management and Orchestration
SNR	Signal to Noise Ratio
SQL	Structured Query Language
SRB	Signalling Radio Bearer
TCO	Total Cost of Ownership
TD	Time Difference
TDD	Time Division Duplex
TDoA	Time Difference of Arrival
TGW	Telemetry Gateway
TIP	Telecom Infra Project
TRL	Technology Readiness Level
TSDB	Time Series Database.
UE	User Equipment
UL	Uplink
ULP	UPF Load Predictor
UMAP	Uniform Manifold Approximation and Projection
UP	User Plane
UPF	User Plane Function
vBS	Virtual Base Station
vCU	Virtual Centralized Unit
vDU	Virtual Distributed Unit
VNF	Virtual Network Function
VPN	Virtual Private Networks
vRAN	Virtual RAN
WLAN	Wireless Local Area Network
WP	Work Package
xApp	Cross-Functional Application
XAI	eXplainable AI
XDP	eXpress Data Path
XGBoost	Extreme Gradient Boost

Executive Summary

BeGREEN is proposing an evolved Radio Access Network (RAN) for Beyond 5G (B5G) communication networks with the aim of accommodating increased traffic and service demands and of improving energy efficiency. BeGREEN covers a wide range of mechanisms to reduce energy consumption at hardware, link, and system levels.

This deliverable presents an exhaustive SotA review focusing on relevant specifications, developments and projects related to RAN Intelligent Controllers (RIC) and their utilisation for the implementation of intelligent and automated control loops. After this, open-source, commercial and simulated/emulated implementations are described. This document presents the design principles of a BeGREEN O-RAN (Open-RAN) "Intelligent Plane" that allows to introduce Artificial Intelligence (AI) and Machine Learning (ML) control and management plane functions to reduce the overall energy consumption of the RAN infrastructure. It also covers the relationship of the BeGREEN Intelligent Plane with the rest of BeGREEN components and the O-RAN architecture, extending the previous work done in BeGREEN D2.1 [1].

The proposed framework enables the development of new AI/ML procedures that recognise time-space patterns in the system (e.g. evolution of traffic, UE –User Equipment- mobility, etc.) and learn appropriate network configuration or reconfiguration actions to improve the network performance and improve energy efficiency. A description of the state of the art, design principles and an initial design of the proposed AI/ML-assisted procedures is provided. In particular, the proposed solutions cover the use of advanced AI/ML methodologies based on Explainable AI (XAI) that allow the identification of entities and areas of the network where energy savings are achievable and, consequently, improve the energy efficiency. The use of AI/ML algorithms is also proposed to dynamically dimension and allocate the computing resources needed for each vBS (virtual Base Station) in an O-RAN O-Cloud Computing Platform, again to improve the network performance and the energy efficiency. Other proposed AI/ML solutions aim to have an intelligent control of the Radio Unit (RU), Reconfigurable Intelligent Surfaces (RIS) and Relays according to the RAN status and traffic and UE mobility predictions with the aim of reducing the energy consumption and improving the network performance. Moreover, other AI/ML solutions aim to enhance the energy efficiency of edge services hosting User Plane Functions (UPF) Network Functions (NF) by properly tuning the CPU (Central Processing Unit) frequency of the edge server. Finally, a joint orchestration of vRANs (virtual RAN) and Edge AI services is proposed to minimize the overall power consumption subject to the performance constraints of the service.

This deliverable serves as the reference document to BeGREEN D4.2, where an initial implementation and evaluation of the BeGREEN Intelligent Plane and proposed AI/ML solutions will be presented.

1 Introduction

The evolution from 5G to beyond 5G (B5G) and 6G mobile communication networks brings a paradigm shift not only in terms of higher speed rates, lower latency, and increased connectivity but also in addressing critical issues related to the environmental implications associated to a higher energy consumption. The way in which B5G and 6G mobile networks are planned, deployed, and managed needs to be improved to reverse the increased trend of energy consumption. These challenges require innovative architectural transformations and novel algorithmic solutions to ensure sustainability and minimise the environmental impact of cellular networks. The RAN consumes more than 70% of the total energy of a 5G system, making it a prority for optimisation.

Cellular networks are undergoing a revolutionary transform with the advent of O-RAN architectures. O-RAN deployments are based on disaggregated, virtualized and software-based components, connected through open and standardised interfaces, and interoperable across different vendors. The O-RAN architecture includes two RAN Intelligent Controllers (RICs), namely, the Non-Real-Time RAN Intelligent Controller (non-RT RIC), and Near-Real Time RAN Intelligent Controller (Near-RT RIC). These controllers, coupled with AI/ML[1] applications developed in xApps (cross-functional application) and rApps (radio access network applications), provide powerful instruments to intelligently manage the RAN with the aim of improving the network performance, and reduce the RAN energy consumption. The non-RT RIC facilitates long-term optimisation of the network, while the Near-RT RIC addresses almost real-time decision-making. The integration of AI/ML xApps and rApps introduces a cognitive layer that can learn from historical data, adapt to evolving network dynamics and make adequate decisions for improving the network performance and the energy efficiency.

In this context, an “Intelligent Plane” is proposed as an additional plane, along with the user plane and the data plane, that allows to introduce AI/ML control and management plane functions to reduce the overall energy consumption of the RAN infrastructure. The proposed BeGREEN Intelligent Plane includes the SMO, the non-RT RIC, the Near-RT RIC and the developed rApps and xApps, which will be empowered by an AI Engine and a datelake. The proposed AI Engine will provide a serverless execution environment hosting the AI/ML models, offering inference and training services to the rApps/xApps. Also, it will manage the lifecycle of the AI/ML models. The datelake is a large data repository which will include metrics from different domains (such as RAN, Core Network, Edge applications, etc.).

The proposed framework allows the development of new AI/ML solutions to recognise time-space patterns in the system (e.g., evolution of traffic, UE mobility, etc.) and learn the adequate network configuration/reconfiguration actions to improve the network performance and improve energy consumption. The proposed AI/ML solutions include:

- ***Energy usage measurement****:* The basis for a holistic optimisation strategy that can help to reach energy efficiency targets is measuring the energy usage of entities in the network. In this context, the use of XAI will allow to identify entities and areas of the network where energy savings are achievable and, consequently, improve energy efficiency.
- ***Virtualized resource allocation in vRAN****:* The use of AI/ML algorithms to dynamically dimension and allocate the computing resources needed for each vBS in an O-RAN O-Cloud Computing Platform to improve the network performance and reduce energy consumption is proposed.
- ***AI/ML-based algorithmic solutions for non-RT RU control****:* The intelligent and dynamic energy-efficient control of RUs is one of the main challenges and opportunities of the O-RAN architecture due to its impact on the global energy consumption of the network. Different approaches such as switching

[1] AI and ML terms denote related and overlapping concepts. In fact, ML can be seen as a subset of AI. In this document, the term AI/ML will be used to denote AI and/or ML techniques.

on/off the cells can be considered to orchestrate RU status according to the network status and the targeted energy efficiency, while AI/ML can be incorporated to enhance decision making through predictions (for example load, mobility, interference...).

- ***AI/ML RIS control**:* This technology can enhance wireless communications by controlling the propagation environment as desired, manipulating the radio waveforms to avoid interferences or extend communication links passively. On the other hand, the use of Joint Sensing and Communications (JSAC) technology to obtain information such as UEs location, combined with a smart use of the RIS may be used to power off certain RUs at specific time periods and save energy without significantly impacting the user performance.
- ***Relay control**:* The deployment of relays can lead to energy savings through the reduction of transmit powers in mobile networks thanks to the better propagation conditions in the involved links. Several challenges arise for exploiting the use of relays for the optimisation of both energy efficiency and spectral efficiency in B5G networks, such as determining the adequate place to locate fixed relays, the power and resource allocation at the BSs and the relays, user to cell association and relay selection, relay activation and deactivation, etc.
- ***Traffic-aware CPU state management to reduce energy consumption of NFV (Network Function Virtualisation) user-plane functions**:* The energy consumption of software-based implementations of user-plane functions such as the UPF or the Central Unit User Plane (CU-UP) is highly dependent on network traffic workload. The dynamic tuning of NFVs can be empowered with AI/ML based predictions (for example expected load or energy consumption) and/or with AI/ML-based algorithms inferring the best configurations.
- ***Joint orchestration of vRANs and Edge AI services**:* The widespread adoption of AI services mandates a fundamental reconfiguration of mobile network management. In this context, the network's role must directly optimize service performance. This optimisation revolves around key criteria, notably accuracy, low end-to-end latency, and high task throughput, all achieved in a resource-efficient manner. This shift is of paramount importance due to the substantial data volumes, computational intensity, and energy consumption associated with these services.

In this context, BeGREEN D4.1 structure is as follows:

Chapter 2 presents a state-of-the-art (SotA) review and an initial design of the O-RAN based BeGREEN Intelligent Plane and its relationship with the rest of BeGREEN components. Firstly, an introduction of the O-RAN architecture is made, focusing on the relevant aspects of the non-RT RIC and the Near-RT RIC as specified by O-RAN, and the intelligent RAN control. Then, a general description of open-source, commercial and simulated/emulated implementations of the RICs and the intelligent planes is presented. Next, we describe relevant projects in the context of RICs and their relationship with BeGREEN. After this detailed review of the state of the art, an initial design of the architecture, interfaces and main functionalities of BeGREEN Intelligent Plane is described. Additionally, the expected interactions with other elements inside and outside of the O-RAN architecture, which are needed to provide energy efficiency in the RAN, are described.

Chapter 3 presents a SotA analysis and an initial design of the proposed AI/ML-based solutions for improving the energy efficiency in different entities of the BeGREEN RAN infrastructure. On the one hand, some solutions propose the use of XAI to identify entities and areas of the network where energy savings are achievable and, consequently, improve energy efficiency. On the other hand, the use of AI/ML algorithms to dynamically dimension and allocate the computing resources needed for each vBS in an O-RAN O-Cloud Computing Platform to improve the network performance and reduce energy consumption is also proposed. Other AI/ML solutions devoted to reducing energy consumption and improve the network performance are based on a smart control of the RU, RIS or Relays according to the RAN status, traffic predictions, UE mobility

predictions, etc. Additionally, AI/ML solutions for CPU state management according to traffic predictions are proposed to reduce energy consumption of NFV user-plane functions. Finally, Chapter 3 also proposes service-aware AI/ML solutions to optimize the overall system energy efficiency of RAN and edge infrastructure.

Chapter 4 presents the final summary and conclusions.

2 O-RAN based Intelligent Plane for System Level Energy Efficiency

This chapter is devoted to the definition of the BeGREEN Intelligent Plane and its relationship with the rest of the BeGREEN components and the O-RAN architecture, extending the previous work done in BeGREEN D2.1 [1]. First, Section 2.1 exhaustively surveys relevant specifications, developments and projects related to RICs. Secondly, Section 2.2 presents the architecture, interfaces, and main functionalities of the BeGREEN Intelligent Plane. Additionally, we describe the expected interactions with other elements inside and outside of the O-RAN architecture, which are needed to provide energy efficiency in the RAN.

2.1 State-of-the-art

This section presents relevant architectures, developments and projects related to RICs and their utilisation to implement intelligent and automated control loops. First, we briefly introduce the non-RT RIC and the Near-RT RIC as specified by O-RAN. Then, we focus on open-source, commercial and simulated or emulated implementations. Finally, we describe relevant projects implementing RICs and/or intelligent planes.

2.1.1 O-RAN specification

O-RAN networks can be built with multi-vendor, interoperable components, and can be programmatically optimised through a centralised abstraction layer and data-driven closed-loop control [2]. In this regard, the RICs are a key element of the architecture to implement RAN intelligence through non-RT and near-RT control loops, which are respectively implemented by the so-called rApps and xApps. These Apps are software applications designed to automate the management and optimisation of the RAN and may be empowered by AI/ML algorithms.

Figure 2-1 illustrates the baseline O-RAN architecture including its main components and interfaces. A detailed description of the baseline architecture is included in BeGREEN D2.1 [1], while in this section we will focus on the RICs components and the intelligent RAN control.

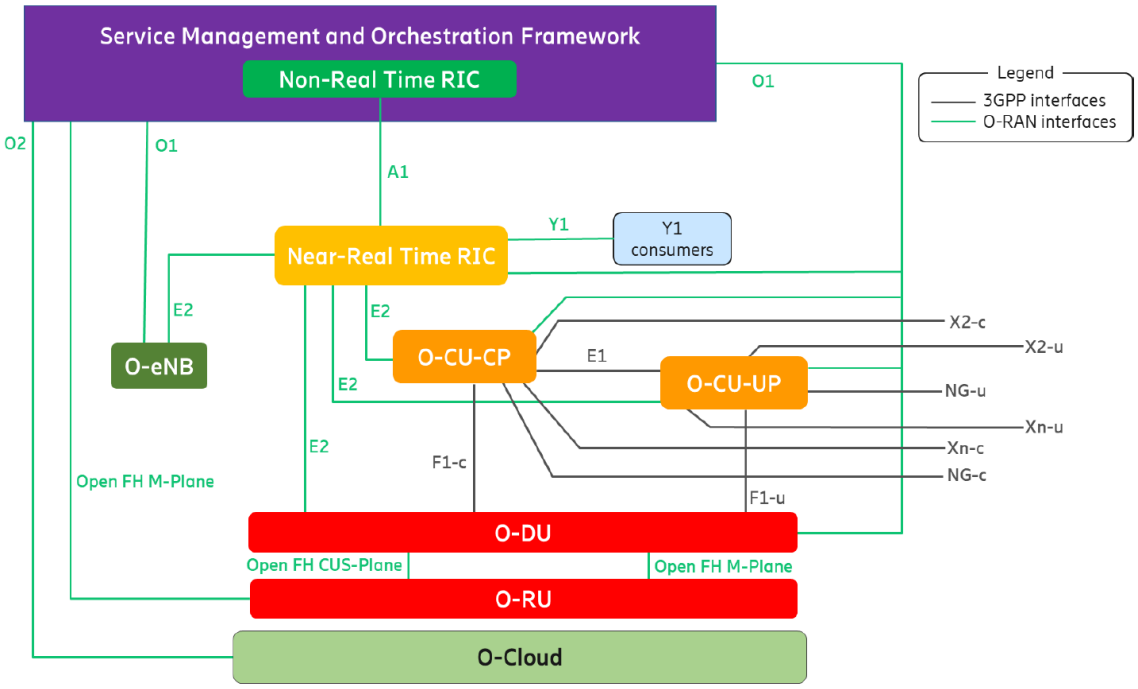


Figure 2-1 O-RAN architecture overview

2.1.1.1 non-RT RIC

The non-RT RIC is a logical function that enables non-RT control and optimisation of RAN elements and resources, and policy-based guidance of applications/features for the Near-RT RIC, targeting control loop periods higher than 1 second. As described in [3], non-RT RIC shall support the main requirements listed below:

- Functionality to register services and allow service consumers to discover them and subscribe/unsubscribe.
- Send and receive messages from the Near-RT RIC via the A1 interface.
- Functionality to expose, produce and consume data, if not implemented in the Service Management and Orchestration (SMO) framework.
- Functionality to support AI/ML workflow functions (e.g., training, monitoring, inference), if not implemented in the SMO.
- Collect analytical data from the Near-RT RIC (Y1 interface), if not implemented in the SMO framework.

One of the key interfaces in the non-RT RIC is the R1 interface, which was recently specified in [4]. Its main objective is to expose R1 services to rApps and between rApps through Service Management and Exposure (SME) operations, which may be related to Data Management and Exposure (DME) services, non-RT RIC services (e.g., A1 services), SMO services (e.g., Operation and Maintenance (OAM)/O1 or O2 services) or even external services (e.g., access to external data or to AI/ML workflow services in case they are not provided by the SMO or the non-RT RIC).

Figure 2-2 depicts the non-RT RIC reference architecture, as specified in [3]. Note that some elements can be anchored inside or outside the non-RT RIC depending on the implementation. Also, some services, such as AI/ML, can be implemented by external components outside of the O-RAN architecture.

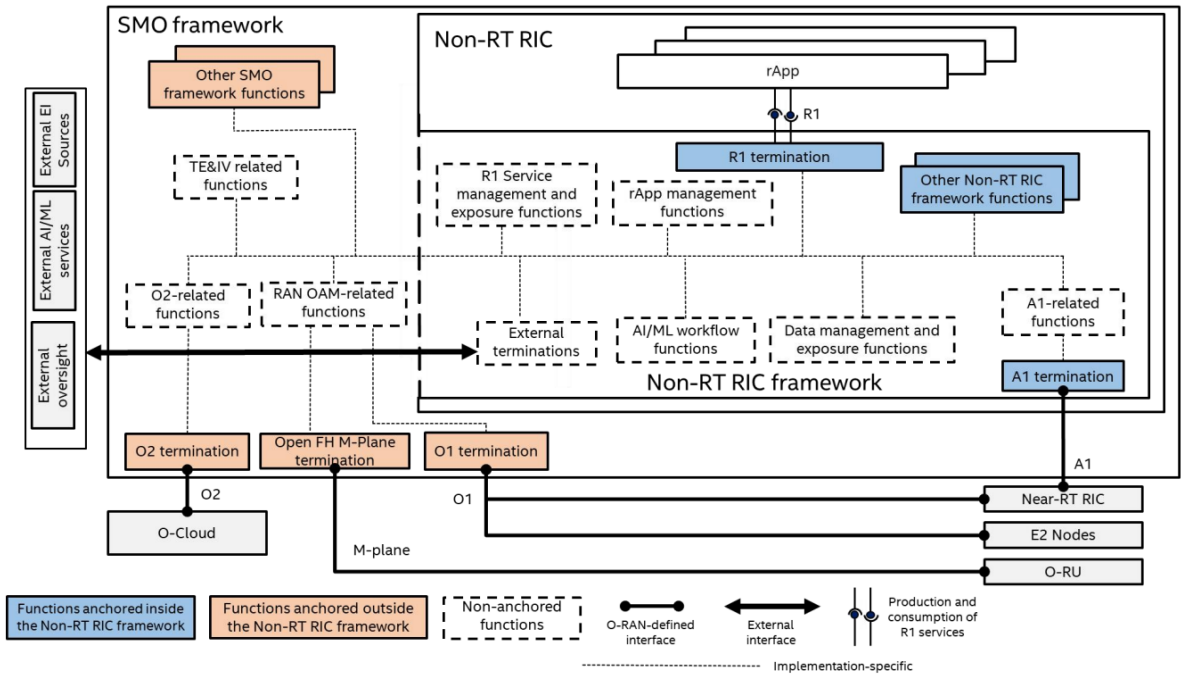


Figure 2-2 - non-RT RIC reference architecture [3]

The role of the A1 interface is to enable the non-RT RIC and the hosted rApps to provide to the Near-RT RIC and the hosted xApps: (i) policy-based guidance, (ii) enrichment information and, (iii), ML model management [5]. A1 policies, which are managed through A1-P service, are mainly focused on providing RAN performance goals and non-RT feedback to the Near-RT RIC to manage the near-RT RAN operations. The A1 Enrichment Information (A1-EI) service is used to enrich the decisions taken by the Near-RT RIC and its xApps with external data not available in their domain, such as application data or core analytics. Finally, the A1-ML services, which are still under definition, may be used to manage AI/ML workflows between the RICs, such as the exchange of model parameters in Federated Learning scenarios. In the context of Energy Saving uses cases, A1-P may be used to control the policies of xApps performing cell on/off switching: e.g., controlling aggressiveness of the algorithm depending on day/night cycles or adding/deleting some specific cells according to slicing or Quality of Service (QoS) policies. In addition, A1-EI may enrich xApp energy saving decisions with external data such as load or user mobility predictions. Although O1 and O2 interfaces correspond to the SMO domain, its exposure to the non-RT RIC and the rApps is key to create intelligent and automated control loops managing the O-nodes and the O-Cloud. O1 is devoted to OAM-related services, such as monitoring the O-nodes, provision configuration changes, or managing the RAN slices. Similarly, O2 allows to monitor and manage the O-Cloud infrastructure and deployed Virtual Network Functions (VNF). In the context of Energy Saving use cases, O1 may be used to manage the RU on/off switching (directly or through the Distributed Unit –DU-), while O2 may allow to manage the vRAN resources according to energy efficiency criteria.

2.1.1.2 Near-RT RIC

The Near-RT RIC serves as a logical function enabling precise control and optimisation of E2 node functions and resources in near real-time, operating with control loops at speeds ranging from 10 milliseconds to 1 second. The main requirements for the Near-RT RIC, as provided by O-RAN Alliance in [6][7], are that the Near-RT RIC shall:

- Provide a database function that stores the configurations relating to E2 nodes, cells, bearers, flows, UEs, and the mappings between them.
- Provide ML tools that support data pipelining.
- Provide a messaging infrastructure.
- Provide logging, tracing, and metrics collection from Near-RT RIC framework and xApps to SMO.
- Provide security functions.
- Support conflict resolution to resolve the potential conflicts or overlaps which may be caused by the requests from xApps.

Based on the above requirements, the O-RAN Alliance specified the Near-RT RIC internal architecture and building blocks as shown in Figure 2-3.

The Near-RT RIC encompasses a range of critical functions, including managing databases and related Shared Data Layer services for reading and writing RAN/UE information. It handles xApp subscription management, conflict resolution among multiple xApps, and facilitates internal messaging. Security protocols for xApps, as well as fault, configuration, and performance management services, are provided. Logging, tracing, and metrics collection capabilities are in place for monitoring and evaluation. The Near-RT RIC also manages interface terminations, including E2, A1, O1, and Y1, connecting with various external systems. The Near-RT RIC acts as a host for one or more xApps, utilizing the E2 interface for gathering near real-time data on a per-UE or per-Cell basis and offering additional value-added services. Control of the E2 Nodes by the Near-RT RIC is guided by policies and enriched data supplied via A1 from the non-RT RIC. Leveraging available data, the Near-RT RIC generates RAN analytics information, accessible through the Y1 interface.

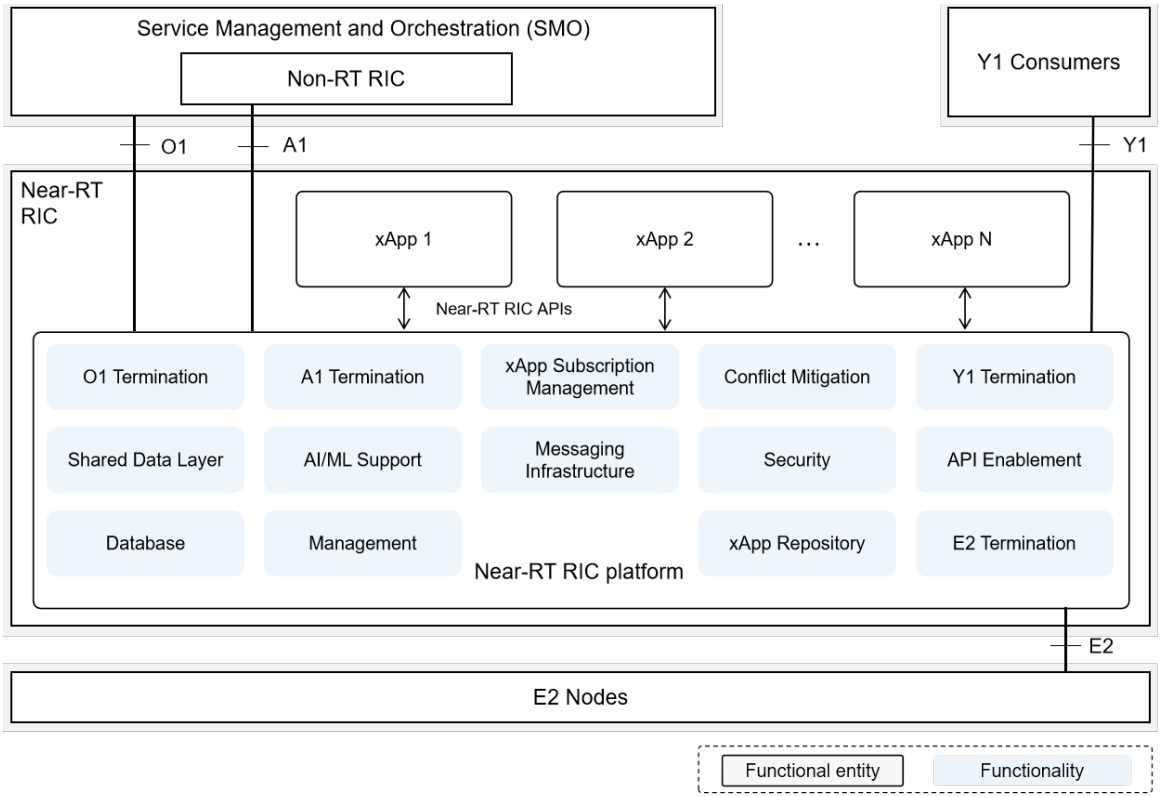


Figure 2-3 Near-RT RIC architecture [8]

Additionally, xApps hosted by the Near-RT RIC execute services and communicate outcomes to E2 Nodes via the E2 interface. The platform features Application Programming Interface (API) enablement functions, supporting operations related to Near-RT RIC API, such as repository and registry management, authentication, discovery, and event subscriptions. The Near-RT RIC also provides support for AI/ML operations, including data pipelining, training, and performance monitoring for xApps. Moreover, it handles the selection and access control of xApps for A1 message routing based on operator policies and policy types, showing its capacity to orchestrating critical functions, ensuring efficient communication and management within the RIC ecosystem. It plays a pivotal role in facilitating seamless interactions between xApps, E2 Nodes, and external systems [8].

The allocation of Radio Resource Management (RRM) functionalities between the Near-RT RIC and the E2 Node is contingent on the capabilities exposed by the E2 Node over the E2 interface, as outlined in the E2 Service Model (E2SM). This model delineates which functions within the E2 Node may be controlled by the Near-RT RIC, establishing a function-specific RRM division between the E2 Node and the Near-RT RIC. For functions specified in the E2 Service Model, the Near-RT RIC may, for instance, monitor, suspend/stop, override, or regulate the behaviour of the E2 Node based on policies. In the event of a Near-RT RIC failure, the E2 Node retains its ability to provide services, although certain value-added services that rely on the Near-RT RIC may experience temporary disruption [9].

The E2 interface, a critical component in O-RAN, is designed with specific requirements and principles. It serves to uniquely identify each E2 Node responsible for delivering RIC Services to the Near-RT RIC. Furthermore, a Near-RT RIC can handle E2 connections from multiple E2 Nodes, each catering to a specific Radio Access Technology (RAT) type. This interface also facilitates addressing specific RAN functions in a given E2 Node, and it ensures the E2 Node's continued operation even if the E2 interface or Near-RT RIC encounters a failure. The E2 interface supports latency requirements ranging from 10 milliseconds up to 1 second, essential for near-real-time optimisation [8][10].

The E2 functionalities are supported by two interfaces protocols, i.e., the E2 Application Protocol (E2AP), and

the E2SM [10]:

- The E2AP establishes the procedural framework for communication between the Near-RT RIC and E2 nodes, offering a fundamental set of services. E2AP messages encapsulate various E2SMs that execute specific functionalities, such as reporting RAN metrics and controlling RAN parameters. Operating atop the Stream Control Transmission Protocol (SCTP) protocol, each E2 node exposes distinct RAN functions, determined by its tuning capabilities, parameter adjustments, and performance metric reporting proficiency. Utilizing publish-subscribe mechanisms, E2 nodes share data, enabling xApps on the Near-RT RIC to selectively subscribe to specific RAN functions. This separation of capabilities allows for precise interaction between xApps and the RAN. At its core, E2AP manages essential interface procedures including setup, reset, indication, error reporting, and Near-RT RIC service updates, facilitating the exchange of supported RAN functions by the E2 node.
- The E2SM encompasses four core services facilitated by the E2AP protocol. These services, which can be customized and combined to establish a comprehensive E2SM, include: E2 Report, E2 Insert, E2 Control, and E2 Policy. E2 Report involves the transmission of E2 RIC Indication messages from an E2 node, conveying critical data and telemetry. E2 Insert notifies an xApp in the Near-RT RIC about specific events in the E2 node, like a UE signalling a potential handover. E2 Control allows for autonomous initiation by the RIC or in response to an insert message, influencing parameters of the E2 node's RAN functions. Lastly, E2 Policy entails a subscription message detailing an event trigger and a policy for autonomous radio resource management execution by the E2 node. Each of these procedures plays a crucial role in orchestrating efficient communication and operations within the E2SM framework.

2.1.1.3 AI/ML support

As was introduced in BeGREEN D2.1 [1], the concrete definition of AI/ML workflows in O-RAN specification is still on-going [11]. Nevertheless, AI/ML support is being considered in both non-RT and Near-RT RICs.

In the case of the non-RT RIC, [3] defines required AI/ML workflow services, which may be provided by the SMO, the non-RT RIC or/and producer rApps, and consumed by consumer rApps. These services include capabilities such as model training, model management and exposure, and model monitoring. At the time of writing this document, model inference services were not included in the specification, although, as discussed in [11], the SMO, non-RT RIC or the rApps may act as inference host.

Regarding the Near-RT RIC, as specified in [7], it may support AI/ML functionalities such as data pipelining, model management, model training and model inference. Again, depending on the implementation and the scenario, some of these functionalities may be implemented and offered by the SMO, the non-RT RIC or the rApps.

In conclusion, although AI/ML support is being considered within the domain of the RICs and AI/ML workflow services should be exposed to rApps and xApps, the O-RAN specification allows flexibility in its implementation.

2.1.2 RAN intelligent controller

This section introduces relevant projects related to the implementation of RICs. First, we describe the main efforts of O-RAN software community (OSC) from the O-RAN Alliance. Then, we detail the RIC implementation in two relevant projects, open air interface (OAI), and open network foundation's (ONF) software-defined radio access networking (SD-RAN). Next, we list relevant commercial RIC realisations. Finally, we present the main open-source simulators and commercial emulators.

2.1.2.1 O-RAN software community (OSC)

OSC is a collaboration between the O-RAN Alliance and Linux Foundation supporting the creation of an open-source community to explore, develop, and mature the concepts and specification developed by O-RAN. As with any open-source community it is based on contributions to its different projects. Companies and developers around the world contribute in various ways. To support them, there are several labs [12] that can be used for pairwise and end-to-end integration testing.

A. OSC non-RT RIC.

The non-RT RIC is an Orchestration and Automation function described by the O-RAN Alliance for non-real-time intelligent management of RAN functions. The primary goal of the non-RT RIC is to support non-real-time radio resource management, higher layer procedure optimisation, policy optimisation in RAN, and providing guidance, parameters, policies, and AI/ML models to support the operation of Near-RT RIC functions in the RAN to achieve higher-level non-real-time objectives. Non-RT functions include service and policy management, RAN analytics and model-training for the Near-RT RICs.

The OSC implementation provides an A1 interface that allows communication with any Near-RT RIC. Using the A1 interface, the non-RT RIC can create and manage policy types and instances in the Near-RT RICs, which provide guidance according to a higher level. It can also be used to provide additional information not available in the Near-RT RIC via the A1-EI service. The RIC also acts as a hosting platform for rApps, while implementing the R1 interface, allowing the communication between non-RT RIC and rApps and the exposure of said rApps towards the SMO or other rApps functions.

OSC non-RT RIC Release H[2] defines the interfaces and components shown in Figure 2-4. One key component of OSC non-RT RIC is the information coordinator service (ICS), which operates as a data subscription platform that simplifies the relationship between data creators and subscribers. Subscribers, or data consumers, can set up a data subscription, called an Information Job, without needing to know who is generating the data. These Information Jobs could draw from multiple data sources.

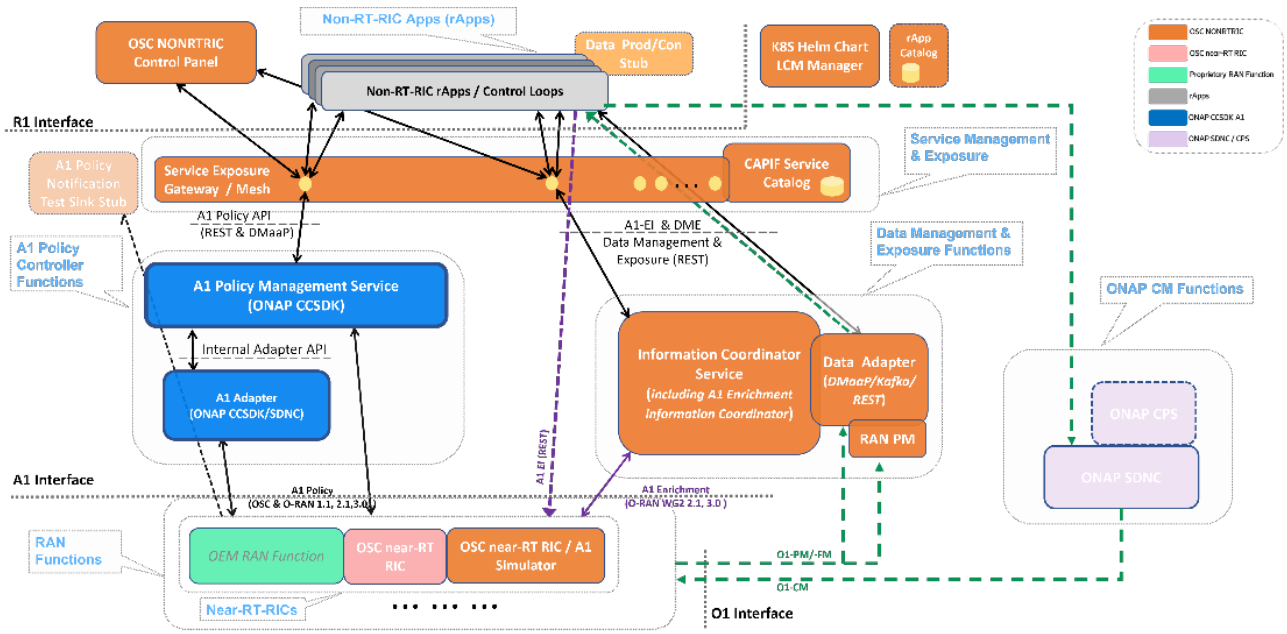


Figure 2-4 OSC non-RT RIC architecture

[2] https://wiki.o-ran-sc.org/display/REL/H+Release

On the other hand, data providers or producers have the capacity to offer a variety of data forms, known as information types. Each information type might be supplied by any number of providers, and the end-users of this data can be diverse, including rApps that utilise R1 APIs or Near-RT RICs employing A1-EI APIs. Each Information Job details the specific data required and any additional parameters that might differ based on the data category involved.

B. OSC Near-RT RIC

The OSC provides an implementation of the Near-RT RIC as a platform built on a microservices architecture, as shown in Figure 2-5. It serves as a host for xApps, which operate on the platform but are developed independently from the Near-RT RIC itself. The platform equips xApps with the necessary tools to manage an array of RAN stations (including eNB, gNB, Central Unit -CU-, DU) within a specific region. This is achieved through the E2 protocol endorsed by the O-RAN alliance, often referred to as the southbound interface.

Moreover, the platform offers northbound interfaces for operators, namely the A1 and O1 interfaces. Through the A1 interface it can communicate with the non-RT RIC, which, as described above, is used to set strategic goals for the network and receive updates on the execution of these objectives. The platform is also tasked with incorporating the O-RAN alliance's O1 management interface, which principally integrates fault, configuration, accounting, performance, security (FCAPS) functions with open network automation platform (ONAP).

The Near-RT RIC platform acts as a mediator for interactions between xApps and the RAN elements via the E2 interface, as well as between xApps and network operators through the A1 and O1 interfaces. xApps utilize the RIC platform's services to function effectively.

Some of the components of the Near-RT RIC platform, such as xApps lifecycle management, E2 status updates, basic RAN information from the fundamental E2 protocol, logging, configuration management, statistics gathering, microservice orchestration, and security, are considered integral parts of the platform.

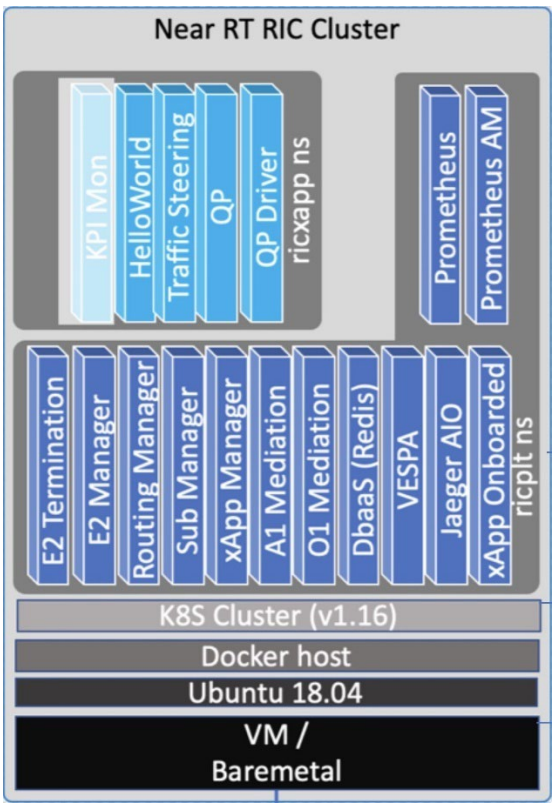


Figure 2-5 OSC Near-RT RIC architecture

C. OSC machine learning operations (MLOps)

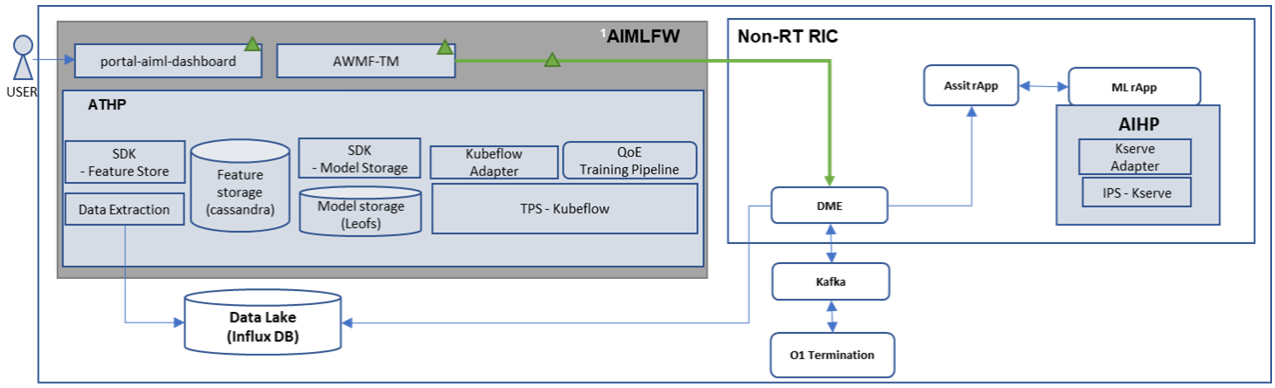
The AI/ML Framework implementation of the OSC is designed within the RAN ecosystem. The Framework's capabilities are divided into several key services and functionalities, each tailored to support different stages of the AI/ML lifecycle.

Firstly, the model management and exposure (MME) service is in charge of managing the lifecycle of AI/ML models. It includes a variety of operations such as registering and deregistering models, updating, storing, and deleting them as well. Additionally, it provides the ability to discover models and manage subscriptions to changes in models, including the notification of such changes. The MME also handles the registration and deregistration of AI/ML model training capabilities.

The training service is another critical component of the framework, focusing on the AI/ML training process. It allows for the initiation of training jobs, querying the status of these jobs, and cancelling them if necessary. It also notifies users of any status changes in the ongoing training jobs.

Alongside the training service, the framework includes a generic training pipeline, which comes with a default generic Kubeflow pipeline established during installation. This pipeline leverages model information provided during the creation of training jobs. The Framework features advanced capabilities such as feature selection, the ability to dynamically change data sources, and the initiation of training only after the necessary data is prepared in the database. To ensure the reliability and efficiency of the AI/ML Framework, automated testing is integral. This includes scripts for installation and testing of all Framework functions, ensuring a smooth and consistent operational experience.

The AI/ML Framework is not static, it is designed to adapt and evolve. It integrates use cases for both non-RT RIC and Near-RT RIC, ensuring that new use cases can be validated and incorporated effectively. Figure 2-6 depicts the current considered architecture.



AIMFLW	AI/ML Framework
AWMF	AI Workflow Management Function
TM	Training Manager
TPS	Training Platform Service
ATHP	AI Training Host Platform
IPS	Inference Platform Service
AIHP	AI Inference Host Platform

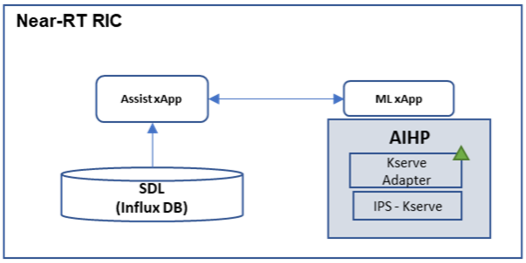


Figure 2-6 AI/ML framework design diagram as specified in O-RAN Release-H[3]

[3] https://wiki.o-ran-sc.org/display/AIMLFEW/Design

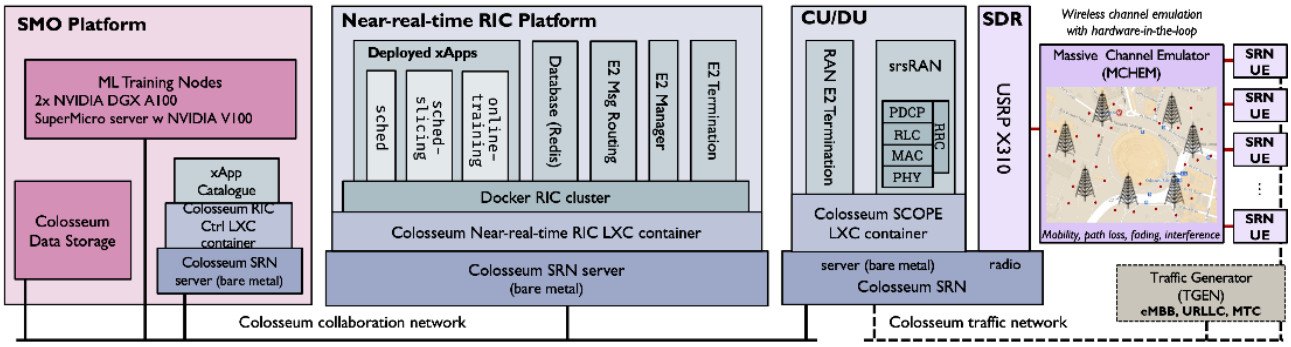


Figure 2-7 ColO-RAN architecture

D. OSC-based projects

This section introduces two relevant projects using OSC RICs: OpenRAN Gym and Open AI Cellular (OAIC).

OpenRAN Gym[4] is an open research platform dedicated to data-centric O-RAN experiments on a larger scale. It is grounded on robust frameworks that manage data collection and RAN control. This platform facilitates comprehensive design and evaluation of data-driven xApps by incorporating an O-RAN-compatible Near-RT RIC and E2 termination. ColO-RAN[5] is currently the only available O-RAN framework for OpenRAN Gym. It is an extensive framework for O-RAN testing, incorporating software-defined radios within its loop. It harnesses the expansive scale and computational prowess of the Colosseum wireless network emulator. Through ColO-RAN, it is possible to perform large-scale ML research, employing O-RAN components, adaptable Base Station (BS), and a "wireless data factory". A notable feature of ColO-RAN is its Near-RT RIC, grounded in the O-RAN Software Community RIC and hosted on Colosseum, which manages a software-centric RAN founded on a SCOPE framework. This comes with a standard aligned E2 termination. The baseline architecture of ColO-RAN is depicted in Figure 2-7.

ColO-RAN introduces a Software Development Kit (SDK) to facilitate rapid integration and assessment of AI/ML-oriented xApps for real-time RAN analysis and management. Broadly speaking, ColO-RAN xApps comprise two primary elements: the Service Model (SM) connector which oversees communications with the Near-RT RIC, such as liaising with BSs, handling Abstract Syntax Notation 1 (ASN.1) message transformation, and accessing the RIC Redis database, and a data-centric logic unit which carries out functions based on real-time Key Performance Metrics (KPMs) from the RAN, such as forecasting traffic or managing the BSs.

Regarding the RAN domain, OpenRAN Gym offers two main options:

1. **Open Air Interface:** OAI is a 3GPP-compliant software implementation of the full 5G New Radio (NR) stack (more details in Section 2.1.2.2). OpenRAN Gym offers the option of deploying an OAI gNB on Colosseum[6], equipped with a standard compliant E2 agent for the E2AP component and custom Service Models based on protocol buffers, together with a Near-RT RIC and a sample xApp.
2. **NS-O-RAN**[7] is the first open-source simulation platform that combines a functional 4G/5G protocol stack in ns-3 (Network Simulator 3) with an O-RAN-compliant E2 interface [13]. It has been designed and implemented to enable the integration of the OSC Near-RT RIC (Bronze release) and ns-3 simulator through an adaptation of the e2sim[8] library from OSC. This way, it allows to run large-scale 5G simulations grounded on 3GPP channel models and a comprehensive representation of the 3GPP

[4] https://openrangym.com/
[5] https://www.northeastern.edu/colosseum/openran-gym/
[6] https://openrangym.com/tutorials/oai-o-ran
[7] https://openrangym.com/tutorials/ns-o-ran
[8] https://github.com/o-ran-sc/sim-e2-interface

RAN protocol stack. This design aids in gathering extensive data on RAN KPMs across diverse simulated scenarios and application contexts such as multimedia streaming, web browsing, and wireless virtual reality. In addition to supporting an O-RAN-compliant E2 interface, ns-O-RAN incorporates two E2SM: E2SM-KPM monitoring and E2SM-RAN control (RC). These models empower real-time control functionalities, exemplified by traffic direction and mobility.

3. **OAIC**[9] is an open-source software architecture and toolset that provides both the AI controllers and the AI testing framework. The project is meant to provide an infrastructure that helps with the research and development of AI-enabled cellular radio networks, by leveraging existing open-source 5G software to integrate the AI controllers into 5G processing blocks, effectively extending the scope of the OSC framework. Figure 2-8 illustrates the main building blocks of its architecture. OAIC includes its own Near-RT RIC based on the OSC release E, a cellular stack based on srsRAN[10] and with support for 5G Non-Stand Alone (NSA), an E2 agent capable of interacting with the standard OSC RIC and support for xApps. srsRAN is an open-source implementation of 4G and 5G RAN, compliant with 3GPP and O-RAN Alliance specifications. Recently, the srsRAN project has released a new version featuring O-RAN CU/DU. Nevertheless, currently OAIC only supports non-disaggregated RANs. Also, it allows to use zeroMQ library to emulate the RAN side (RAN nodes and UEs). OAIC is being used in the POWDER[11] platform within the platforms for advanced wireless research (PAWR) program. POWDER aims to support software-programmable experimentation on O-RAN using Software Defined Radio (SDR) platforms.

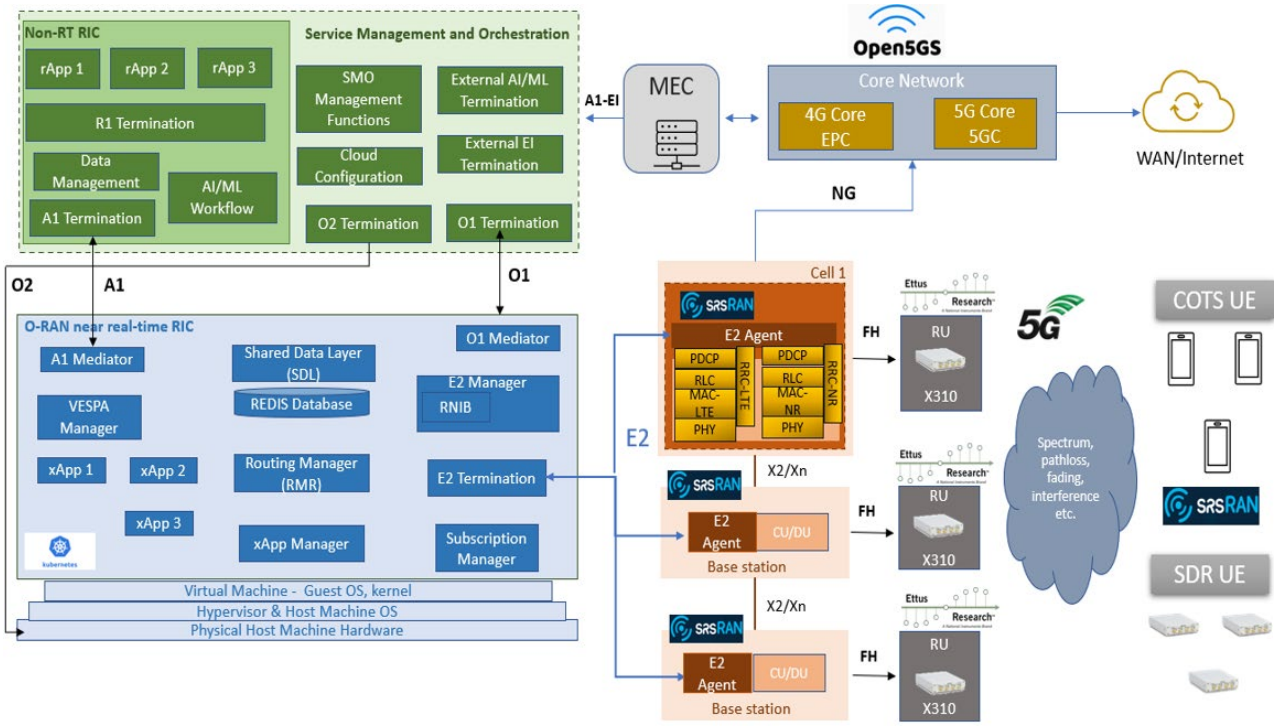


Figure 2-8 OAIC architecture

2.1.2.2 Open Air Interface

The Open Air-Interface Software Alliance (OSA), created in collaboration with the French research institution EURECOM, is a non-profit organisation dedicated to advancing open-source software and hardware

[9] https://www.openaicellular.org/
[10] https://www.srslte.com/
[11] https://powderwireless.net/

development in the fields of Core Network (CN) and RAN technologies. The software infrastructure provided by OAI[12] includes all the essential components needed for the seamless deployment of RANs (including eNB, gNB, 4G UE, and 5G UE) and core networks (4G-CN and 5G-CN), with individual parts distributed under different licenses.

The primary goal of OSA is to transform RAN and CN into adaptable and open platforms for delivering network services. This platform enables the exploration of new use cases across various industries. The project group within OSA is working on developing a set of flexible control and management frameworks with extendable interfaces on top of OAI RAN and CN. These frameworks allow for precise monitoring and adaptability of the CN infrastructure. Additionally, they enable the use of a customisable 4G/5G service delivery platform, enhanced by a range of intelligent network applications (xApps) and service development kits.

MOSAIC5G (M5G) project group, within the OSA, is an organisation dedicated to transform RAN and CN into agile and open network-service delivery platforms. This project group's primary focus is on creating adaptable control and orchestration systems, featuring expandable interfaces, which will be integrated with OAI's RAN and CN. These systems will enable precise monitoring and programmable control of the foundational network infrastructure. Figure 2-9 illustrates the M5G architecture and its connection with the OAI stack, while its components are described as:

- Trirematics is a cloud-native orchestration and management framework designed to facilitate the management of various RAN and CN deployment scenarios through blueprint-based processes. It is equipped to accommodate various extensions as value-added features, including:
    - Kubernetes, which serves as a container orchestration technology.
    - Kubeflow, an advanced ML Toolkit compatible with Kubernetes.
    - Operator for efficient system management.
- FlexCN is a flexible and programmable core network controller, with the capabilities of Software-Defined Core Networks (SD-CN). It facilitates the supervision and management of the foundational 4G and 5GC networks by making use of the standardised 3GPP subscription and notification mechanism. It also allows to create various service models, such as key performance monitoring, as well as a range of xApps and SDKs, including traffic control and slicing capabilities.

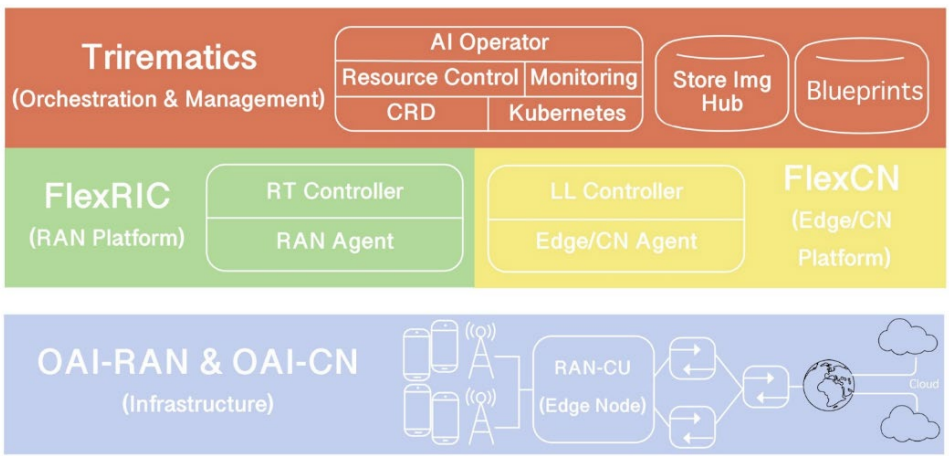


Figure 2-9 MOSAIC5G schematic architecture

[12] https://openairinterface.org/

- FlexRIC [14] is a flexible and programmable RAN intelligent controller for SD-RAN. It interfaces with the OAI radio stack over the O-RAN-defined E2-interface to monitor and control the RAN in real-time. It is a flexible and efficient SDK, a pivotal element in the landscape of SD-RANs, fostering the development of specialised controllers tailored to specific use cases while adhering to a lean and resource-efficient approach. It is designed with several fundamental principles in mind, addressing critical design challenges:
    - RAT Agnostic and Vendor-Independent SD-RAN Design.
    - Ultra-Lean Design: low-latency and resource-restricted use cases.
    - Flexibility and Forward-Compatibility.
    - As shown in Figure 2-10, FlexRIC architecture consists of an agent library and a server library:
        - The FlexRIC agent library serves as the foundation for extending BSs with intelligent agent functionalities. It is designed for seamless integration into various BS implementations and deployment scenarios while providing additional capabilities for handling multiple controllers within a multi-service environment.
        - The server library, together with internal applications (iApps) [14] and optionally a communication interface, build the controller. It manages agent connections, and multiplexes messages between iApps and the agents. iApps implement and abstract some common control operations which can be further exploited by xApps via northbound communication interfaces without needing to program again the control logic (e.g., RAN slicing control). They also allow the implementation of faster control-loops in the FlexRIC.
    - As depicted in Figure 2-11, FlexRIC enables the seamless development of specialised, service-specific, E2-compatible controllers, consisting of:
        - Controller Specialisation: Specialised controllers are realized through iApps, communication interfaces, and xApps. This modular approach allows for the development of controllers tailored to specific use cases, such as traffic control, slicing, and recursive slicing through network virtualisation.
        - FlexRIC Server Library: The server library handles agent connections, message multiplexing, and dispatching E2AP messages. It follows an event-driven/callback-driven model, minimizing overhead and invoking iApps only when new messages are received. This lightweight approach ensures efficient operation.

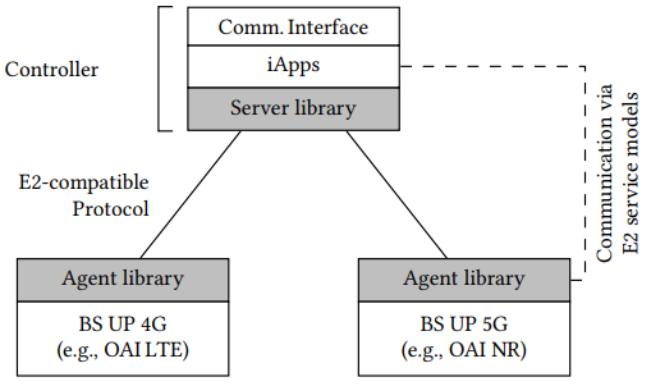


Figure 2-10 FlexRIC SDK architecture: agent and server library

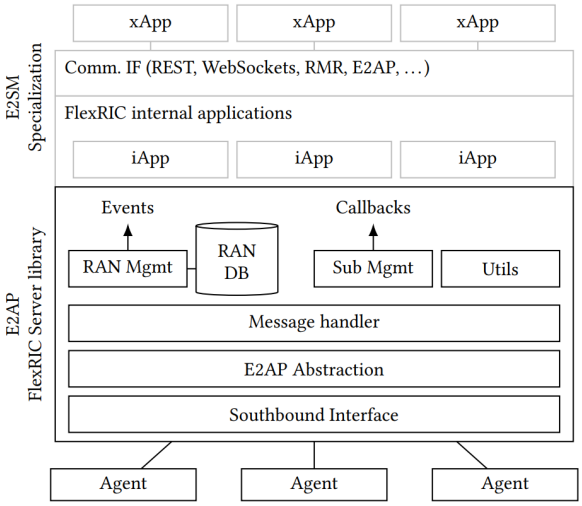


Figure 2-11 FlexRIC iApps and xApps

2.1.2.3 Software-defined RAN (SD-RAN)

SD-RAN[13] is the ONF platform for the development of SD-RAN, compliant with the O-RAN architecture. It is designed for modern, cloud-based systems and builds on other tools and platforms like Open Network Operating System (ONOS) and Aether. The main goal of the project is to develop open-source components for the control and user planes of the CU/DU of a disaggregated RAN. To do so, they stay in close coordination with the O-RAN Alliance and O-RAN Software Community. Figure 2-12 depicts the main architecture of SD-RAN. At its core, the SD-RAN architecture starts with a micro-ONOS based Near-RT RIC. As any micro-ONOS system, it is designed around the use of different micro-services with very delineated roles and responsibilities. The SD-RAN RIC can be thought of as a chassis that includes a series of key subsystems that constitute the platform. xApps can then be installed into this chassis and use the provided API to interact with the rest of services. The RIC comes with all the necessary terminations for the main standard O-RAN interfaces, i.e., E2, A1 and O1. Each termination runs as a separate micro-service, and each will allow running multiple instances to provide scalability and high availability. Each one also provides an external interface to interact with the outside environment using the protocols and encodings specified by the O-RAN standards.

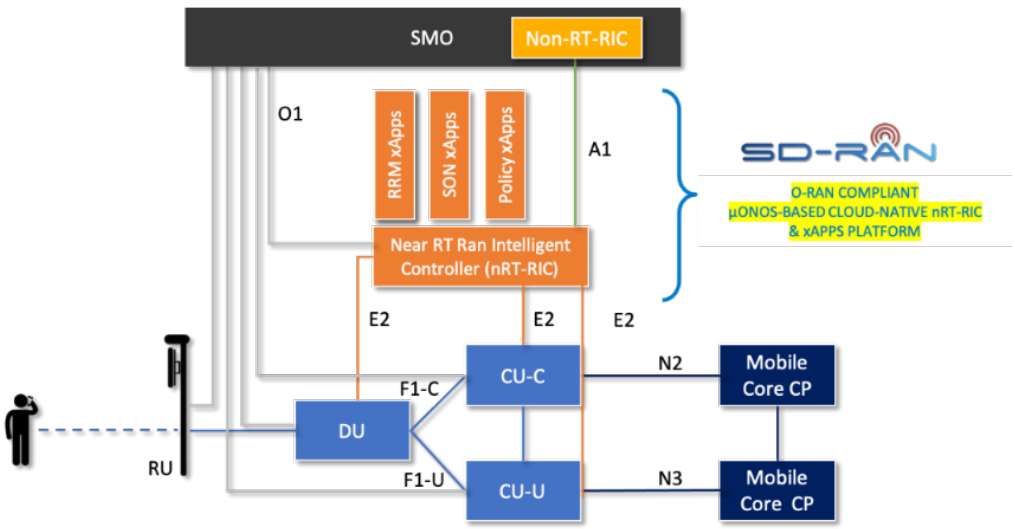


Figure 2-12 SD-RAN main architecture

[13] https://opennetworking.org/open-ran/

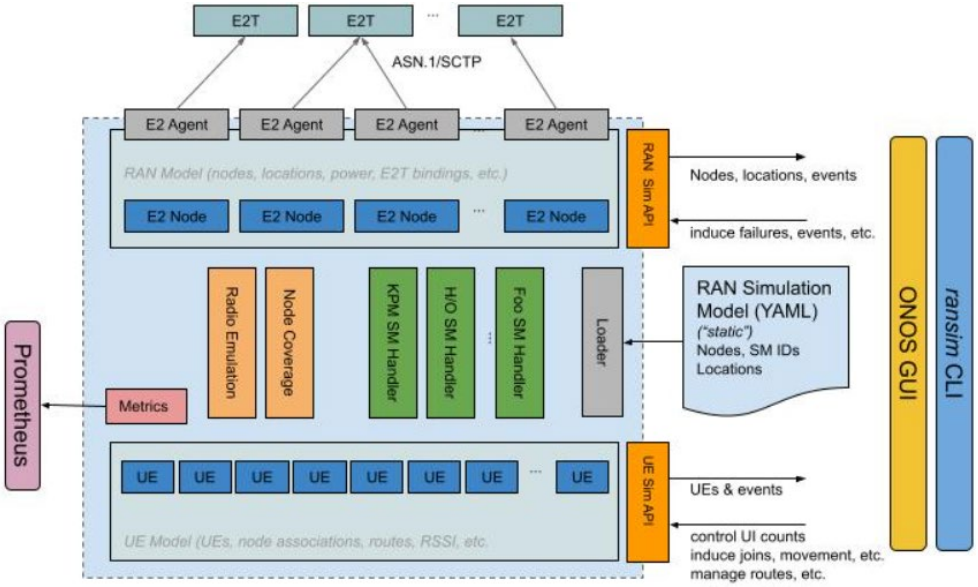


Figure 2-13 SD-RAN RAN simulator

To simplify the development of RIC applications, an SDK is also provided. The role of the SDK is to encapsulate some of the complexities of dealing with individual services and to provide implementation of client-side logic, which is likely to be shared by many applications.

The easiest way to install the platform is using SD-RAN-in-a-Box[14] (RiaB), a SD-RAN cluster able to operate within a single host machine. It provides a development/test environment for developers/users in ONF SD-RAN community, as well as deploying the whole SD-RAN infrastructure on Kubernetes: the ONOS RIC, an Open Mobile Evolved Core (OMEC) and the RAN (CU/DU/UE). Regarding the RAN, SD-RAN makes use of OAI CU, DU and RU O-RAN compliant disaggregated baseband units.

RiaB also comes with the option to install a RAN simulator, depicted in Figure 2-13 , which allows simulating several RAN CU/DU nodes and RU cells via the O-RAN E2AP standard. The simulated RAN environment is described using a YAML model file loaded at start-up. The simulator offers an API that can be used at run-time to induce changes in order to simulate a dynamically changing environment. Some of the simulation parameters include the number of E2 nodes, the service models to be used, the topology of the RAN and the number of UEs. All the data generated by the simulations can be sent to different data stores for their posterior retrieval.

2.1.2.4 Commercial and industrial implementations

O-RAN core principles of intelligence and openness have fuelled the emergence of commercial RIC solutions. In this subsection, a description of different commercial RIC solutions from relevant vendors is provided. Then, the Accelleran's dRAX is described, which is the RIC framework that will be integrated in the BeGREEN Intelligent Plane.

2.1.2.4.1 Commercial RIC solutions

Airhop is a cloud-native RAN software technology company. This organisation initiated the SD-RAN project in 2020 with the aim of creating components suitable for disaggregated RAN deployments. Since its affiliation with the ONF, Airhop has been engaged in the development of a Near-RT RIC and a suite of xApps for the management of the RAN. In April 2021, in collaboration with Facebook and the Telecom Infra Project (TIP), they successfully showcased the inaugural instance of multi-vendor RAN automation, integrating the ONF's

[14] https://docs.sd-ran.org/master/sdran-in-a-box/README.html

SD-RAN RIC with Airhop's xApps.

Altiostar is a prominent participant in the RIC domain, which is specialised in open vRAN technology provisioning. The company has successfully formulated its own cloud-native microservice-driven solutions, encompassing a virtual CU and DU, and third-party O-RAN compliant RUs and antennas. These components interface with Commercial Off-The-Shelf (COTS) hardware platforms through a virtualised infrastructure. Additionally, Altiostar is presently engaged in the development of its Near-RT RIC platform and associated xApps. It has also outlined plans to extend support for third-party RICs, thereby enabling functionalities such as traffic steering, optimisation of radio channel resources, strategic resource allocation, and QoS monitoring and enforcement for network slicing.

Capgemini's RATIO includes a Near-RT RIC, a non-RT RIC and the SMO. RATIO supports a fully disaggregated RIC architecture and different vendors for CU/DU and xApps/rApps. Additionally, it includes built-in O-RAN-compliant xApps and rApps, such as Traffic Steering or Energy Saving, that can run on any RIC platform and are pre-integrated with the Capgemini RIC and 5G CU/DU nodes. Both RICs support integration with the Capgemini Net Anticipate platform, which has support for ML models and workflows.

Juniper RIC is an open and interoperable platform based on a cloud-native microservices architecture and is fully compliant with the O-RAN specifications and interfaces. Juniper RIC includes the non-RT RIC and Near-RT RIC with their associated rApps and xApps to control the RAN network functions. It supports both an open API and an SDK for integration with any third-party O-RAN-compliant xApps or rApps and integrates with any O-RAN-compliant SMO solution. Juniper RIC incorporates some built-in rApps/xApps such as RAN Slice Service Level Agreement (SLA) assurance, Traffic steering or Energy Efficiency (switching off cell, network element, and network function during periods of low traffic). The non-RT RIC implements AI/ML workflow services.

Mavenir's RIC solution also comprehends both non-RT RIC and Near-RT RIC components. Regarding AI/ML, the non-RT RIC supports offline model training and inference through rApps, while in the Near-RT RIC the AI/ML inference is performed through containerized applications hosting trained AI/ML applications. Mavenir Near-RT RIC allows control RAN activity at both the cell and individual user level.

Nokia introduced its Service Enablement Platform (SEP) in March 2021, offering network programmability and integrating AI/ML into the Open RAN ecosystem. Notably, Nokia's SEP is distinctive for combining Near-RT and Multi-Access Edge Computing (MEC) capabilities within a single platform, adaptable to specific enterprise needs. Collaborating with AT&T, Nokia conducted trials of various xApps on AT&T's 5G millimetre wave (mmWave) network edge, showcasing spectrum efficiency improvements and rapid feature integration in the RAN. The partners have intentions to open-source the RIC, enabling third-party contributions for further code development.

Parallel Wireless has developed its OpenRAN Controller, facilitating Open RAN solutions. Distinguished as the first to offer a comprehensive "all G" RAN, it encompasses virtualised 2G Base Station Controller (BSC), 3G Radio Network Controller (RNC), 4G eNB, and X2/S1 Gateway configurations. This 5G native platform also offers a seamless migration path to 5G technology. Additionally, Parallel Wireless offers both Near-RT and non-RT controllers, each serving distinct functionalities in RAN orchestration, configuration, optimisation, fault management, and network intelligence application.

Sterlite Technologies (STL) and ASOCS collaborated to present their RIC during the Global O-RAN Alliance Plugfest in India, hosted by Bharti Airtel in 2020. The demonstration spotlighted the Near-RT RIC's efficacy in mobility load balancing on the O-RAN E2 interface, with future potential to manage escalating network demands through SDN.

2.1.2.4.2 Accelleran's dRAX solution

Accelleran's dRAX is a cloud-native O-RAN aligned product, comprising a loosely coupled Control User Plane Separation (CUPS) CU, a distributed Near-RT/non-RT RIC, and a RAN-oriented SMO/Dashboard with xApp/rApp SDK. dRAX offers an open software framework for 5G RAN control plane functions, closely aligning with Open RAN principles. Its architecture, shown in Figure 2-14, is rooted in cloud-native principles, utilising containerised microservices that communicate asynchronously. Each key RAN component, whether it is the Central Unit Control Plane (CU-CP), Central Unit User Plane (CU-UP), Near-RT-RIC/non-RT-RIC, SMO, or xApp/rApp SDK, is disaggregated into distinct service entities. dRAX integrates with several OpenRAN ecosystem DUs/RUs, accommodating both Split 7.2 and Split 2 architectures [15][16].

dRAX presents an open software framework specially designed for the control plane functions of 5G RAN, ensuring adherence to Open RAN architectural norms. Its cloud-native framework, built on containerised microservices, guarantees efficient communication through an asynchronous messaging system. Key RAN components, such as CU-CP, CU-UP, Near-RT-RIC, non-RT-RIC, SMO, and the xApp/rApp SDK, are segmented into individual service entities for detailed management. dRAX smoothly interfaces with diverse OpenRAN ecosystem Dus and RUs, endorsing both the Split 7.2 (decoupled DU and RU) and Split 2 (integrated DU and RU) configurations. It can be divided in the following O-RAN aligned products.

- CUPS.
- Distributed Near-RT/non-RT RIC.
- xApp/rApp SDK.
- RAN-focused SMO/Dashboard.

Accelleran's dRAX™-RIC simplifies the integration and management of xApps and rApps, offering essential services through containerized deployment. It is a go-to platform for operators and developers, utilizing real-time RAN data to create advanced AI-driven applications for enhanced RAN intelligence and automation. With easy-to-use templates and APIs, dRAX™-RIC empowers third-party developers to effortlessly incorporate RAN intelligence and control. Aligned with the Open RAN Alliance's vision, it provides a robust SDK for seamless integration of third-party xApps/rApps into the RAN. The Near-RT/non-RT RIC serves as a platform for xApps/rApps as microservices, seamlessly integrated into the dRAX ecosystem.

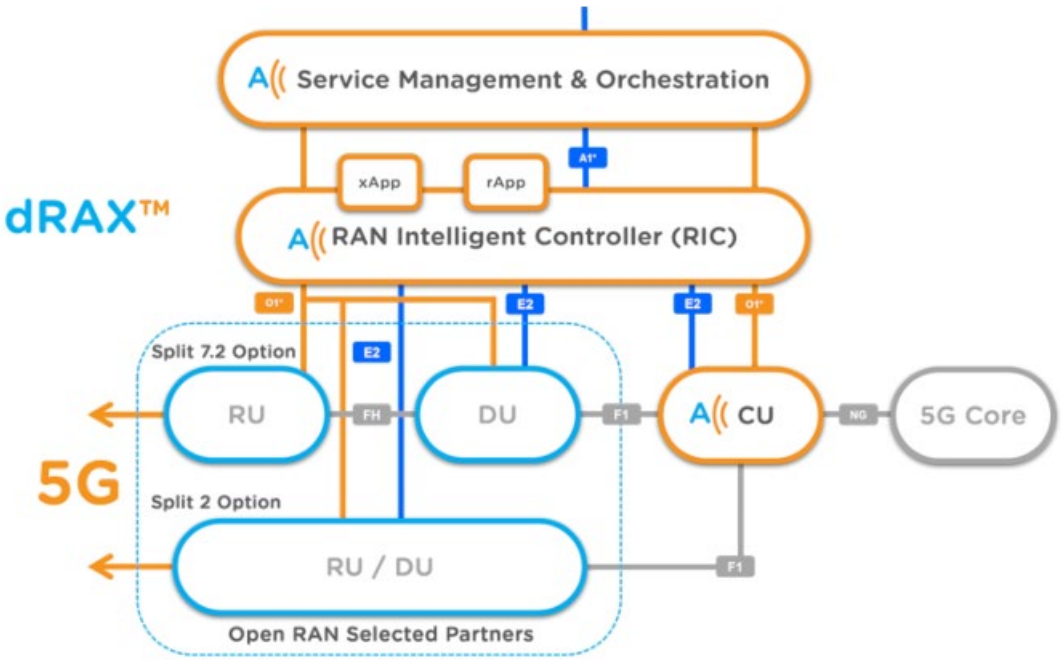


Figure 2-14 Accelleran dRAX components model

Below is a selection of x/rApps developed by Accelleran and third parties, deployable on the dRAX platform:

- Network Optimisation: Fine-tunes resource allocation in real time, optimising network performance during peak usage.
- Advanced Analytics: Utilises data analytics to offer valuable insights into network trends, facilitating proactive decision-making.
- Service Orchestration: Facilitates the introduction of new services and features into the network, enabling service orchestration and delivery.
- Network Slicing: Automates the network slicing process and optimises the allocation of network resources for distinct applications, services, and users. It aids operators in delivering bespoke network services, enhancing network efficiency, and trimming operating costs.
- Network Capacity Planning: Streamlines the network capacity planning process and perfects the allocation of network resources. It scrutinises network traffic patterns, anticipates future demand, and optimises the network's resource capacity.
- Network Monitoring: Automates the network monitoring process, offering real-time insights into OpenRAN network performance. It identifies network anomalies, highlights performance bottlenecks, and alerts network operators in real time.
- Self-Organising Networks: Enables the network to self-organise and self-optimise, diminishing the necessity for manual intervention. It automates the configuration and management of network elements, enhancing network performance.
- Energy Saving: This xApp harnesses the collective power of three distinct xApps, all created via our SDK and orchestrated on the RAN Intelligent Controller. In a simulated environment, this combined system showcased a remarkable capability of achieving energy savings for the network.

2.1.2.5 Simulators and emulators

This subsection presents a brief description of relevant open-source simulators and commercial emulators. Recent 5G developments require simulation and/or emulation of the new RIC component. Since the RIC can host arbitrary operator-specific algorithms, in a simulation environment this is more just a software interface allowing plug-in of new code, rather than an active component itself.

The word “simulator” traditionally means a pure-software system, whereas “emulators” include at least some hardware. Network simulators are traditionally classified as either link-level or system-level. The former are typically very precise in their simulation at the physical layer but become too slow if many connected links need to be simulated. The latter deliberately sacrifice some of this precision in order to achieve fast simulation of large networks.

The AI-enabled Massive MIMO (AIMM) simulator below is in the latter category. The well-known ns-3 to some extent tries to cover both simulation domains, but this often makes it too slow for large systems. On the other hand, TeraVM and Keysight offer emulation solutions to address O-RAN compliant testing of RICs and its associated rApps and xApps.

Recent 5G developments require simulation and/or emulation of the new RIC component. Since the RIC can host arbitrary operator-specific algorithms, in a simulation environment this is more just a software interface allowing plug-in of new code, rather than an active component itself.

2.1.2.5.1 AIMM simulator[15]

Recent developments in the field of AI provide new capabilities of generating automated solutions for network management functions. Specifically, Reinforcement Learning (RL) is an approach for dynamically controlling and solving Markov Decision Processes. A RL intelligent agent learns to make sequential decisions by interacting with the environment. Other options include neural network and deep learning methods. To gather information and train any of these intelligent agents, it is necessary to have an accurate simulation framework for network management functionalities, activities, processes and use cases.

In the AIMM Simulator, these processes will live in a simulated RIC. This component can host arbitrary AI/ML algorithms, typically linked in as external processes which execute in parallel with the main simulator. This is different from a RIC emulator, since a RIC emulator typically interacts with real hardware.

A. AIMM Simulator general design considerations

AIMM Sim is a system-level simulator which emulates a full cellular radio system following 5G concepts and channel models. The intention is to have an easy-to-use and fast system simulator written in pure Python with minimal dependencies. It is especially designed to be suitable for interfacing to AI engines such as 'TensorFlow' or 'pytorch', and it is not a principal aim for it to be extremely accurate at the level of the radio channel. For the latter task, pre-computed look-up tables (based on simulated channel models) are used to obtain fast run-times. If a more precise link-level model is required, a simulator such as ns-3 can be used. The code has a structure as shown in Figure 2-15.

The output of simulation runs are logfiles in a standard format (by default, tab-separated columns) and doesn't include plotting or post-simulation analysis. The lines in the logfile are constructed and formatted by an instance of the Logger class. For testing and debugging purposes, a real time plotter is provided as a separate program. This reads and plots the logfile as it is generated, through a shell pipeline. Extensive online documentation, with a full set of tutorial examples, is provided at [17].

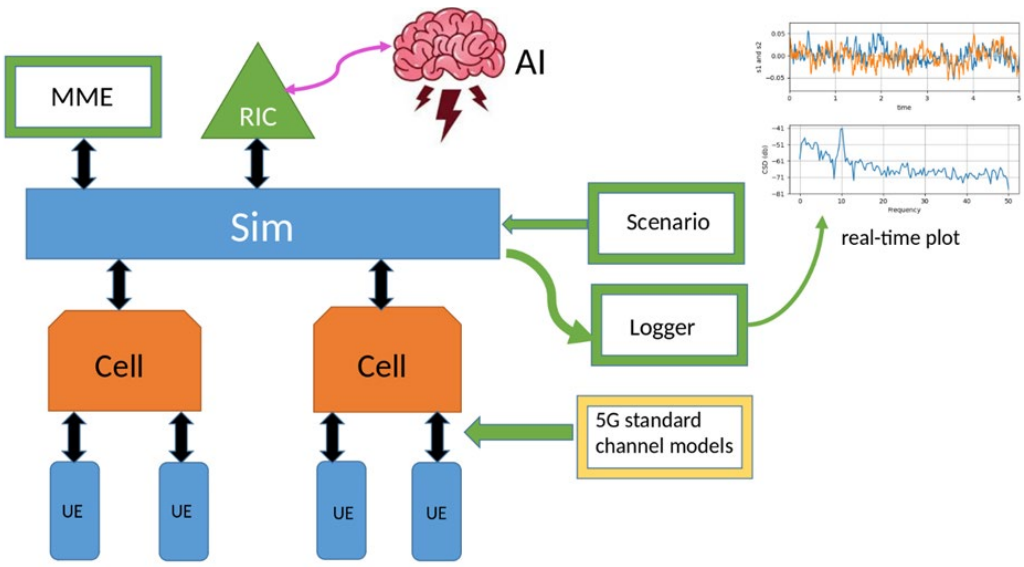


Figure 2-15 AIMM simulator block structure

B. AIMM simulator detailed design

The following factors influence the overall software architecture:

1. The software architecture should closely mimic the real system, with a software class for each type

[15] https://aimm-simulator.readthedocs.io/en/latest/

of network component.

2. The components should exchange traffic in a similar way to the real system. However, "traffic" here is an abstraction; there is no concept, for example, of IP packets, or of resource blocks at the physical layer. These constraints are imposed to get sufficient speed from the simulator, and thus to get as many ML training episodes in a given time as possible.
3. There should be a RIC module, at the top level of management. The AI or ML methods will operate solely in the RIC, effectively as xApps and rApps.
4. The simulation technique should be the discrete-event method. In the core of the simulator, a queue of pending events is maintained. Most events will be periodic (such as UE reporting), and an easy-to-use framework is provided to automate this. The discrete-event method has negligible overheads and allows easy mapping from simulated time to real time.
5. Sub-banding (division of the channel into sub-channels which may be dynamically reallocated between cells) is implemented on all Cell objects, but the number of sub-bands may be set to 1, effectively switching off this feature.
6. All simulations take place in three spatial dimensions, for example, to allow modelling of high office buildings. Some simple capabilities for accounting for wall losses in indoor scenarios are provided.
7. Dynamic features of a specific simulation are handled by a Scenario class. This can, for example, move users according to some mobility model.
8. UE handovers between cells will be handled internally by a heuristic based on Reference Signal Received Power (RSRP), as in a real system. However, for research into smart or AI-based handover strategies, this default heuristic can be overridden.
9. In fact, all modules can be overridden or have their default behaviour modified if desired, using the usual subclassing technique.

C. Outline of usage principles

The basic steps required to build and run a simulation are:

1. Create a Sim instance, which represents the complete simulation.
2. Create one or more cells. Cells are automatically given a unique index, starting from 0.
3. Create one or more UEs. UEs are automatically given a unique index, starting from 0.
4. Attach UEs to cells.
5. Create a Scenario, which typically moves the UEs according to some mobility model, but in general can include any events which affect the network.
6. Create one or more instances of the Logger class.
7. Optionally create a RIC, possibly linking to an AI engine.
8. If necessary, create a custom Logger class by subclassing.
9. Start the simulation.
10. Plot or analyse the results in the logfiles.

The AIMM system-level simulator allows easy construction of large-scale 5G network simulations, with a clean interface (through the RIC class) into standard AI software packages. Because the RIC class has privileged access to internal cell data, as well as permission to change settings operating parameters in cells, it is the right place to run any AI or ML components.

Furthermore, current developments such as implementing xApps and rApps with communication via Google 'protobuf' can be accommodated by putting a simple translation layer in the RIC. Thus, the current design is essentially agnostic regarding messaging protocols.

2.1.2.5.2 ns-3 simulator

The ns-3 is an open-source discrete-event network simulator predominantly used for research and educational purposes. It offers realistic simulations of network protocols and devices, granting the ability to craft and simulate complex network models. One of its major strengths is its extensibility, achieved by promoting the creation of third-party modules and by allowing users to implement their protocols or modify existing ones. Another interesting feature of ns-3 is its capacity to merge simulated network nodes with real-world applications and traffic generators. ns-3 also provides Python bindings, enabling users to script their simulations in Python, which makes it very easy to integrate with third-party software.

As mentioned above, ns-3 heavily relies on the modules created by the community to enhance its capabilities. One of these modules is the ns3-mmWave[16], which can simulate 5G cellular networks operating at mmWave. The module includes several detailed statistical channel models as well as the ability to incorporate real measurements or raytracing data. The physical (PHY) and Medium Access Control (MAC) layers are modular and highly customizable, making it easy to integrate algorithms or compare orthogonal frequency division multiplexing (OFDM) numerologies.

An evolution of the mmWave module is 5G-LENA[17]. It is more advanced than its predecessor in terms of beamforming, Time Division duplex (TDD), 3GPP channel model, and operation at Frequency Range 2 (FR2). It incorporates fundamental PHY-MAC and NR features aligned with 3GPP NR Release-15.

Additionally, some interesting projects related to O-RAN have been integrated with ns-3:

- **ns-3 O-RAN[18]** is a module that implements the classes required to model a network architecture based on the O-RAN specifications. These models include Near-RT RIC that is functionally equivalent to OSC's, and reporting modules that attach to simulation nodes and serve as communication endpoints with the RIC in a similar fashion as the E2 Terminators in O-RAN.
- **ns-3 near-RIC[19]** is an extension of the mmWave module. It enables support for running multiple terminations of an O-RAN compliant E2 interface, effectively allowing the connection of ns-3 with any Near-RT RIC that implements an E2 termination. For instance, it can be used to connect ns-3 with the OSC Near-RT RIC by using the E2 simulator[20], available since release E.

2.1.2.5.3 Commercial Emulators

A. TeraVM

TeraVM is a versatile solution for application emulation and security performance, offering extensive test coverage for both wired and wireless networks. It operates as a software-based L2-7 test tool, compatible with x86 servers and major cloud platforms like Azure, Amazon, and Google. This virtualized tool allows testing and securing devices, networks, and services across different environments, including labs, data centers, and the cloud, with consistent performance levels. This flexibility ensures that optimized networks and services can be deployed with minimal risk [18].

[16] https://github.com/nyuwireless-unipd/ns3-mmwave
[17] https://5g-lena.cttc.es/
[18] https://github.com/usnistgov/ns3-oran
[19] https://github.com/o-ran-sc/sim-ns3-o-ran-e2
[20] https://wiki.o-ran-sc.org/display/SIM/E2+Simulator

One of TeraVM's notable features is its flow-based tool, providing real-time per-flow statistics and the ability to emulate and measure individual endpoint and application performance for various services. Its adaptive engine dynamically determines the maximum capacity of Devices Under Test (DUT), enabling the same test profile to be used for multiple platforms.

TeraVM excels in wireless mobility validation, supporting various generations (5G, 4G, 3G, 2G) with realistic testing scenarios. It enables highly scalable user and control plane traffic, allowing for testing capabilities that exceed 100 Gbps of traffic. With a focus on realism and adaptability, TeraVM provides a comprehensive solution for application emulation, security validation, and performance testing, ensuring the robustness and reliability of networks and services across different environments and technologies. It has several emulators and testers that are summarized next:

- **TeraVM Core Emulator**: The TeraVM Core Emulator from VIAVI helps RAN engineers overcome 5GC Network dependencies. It offers a controlled and repeatable test environment that facilitates the rapid implementation of 3GPP standards. This simplifies the development process for 5G gNBs and the introduction of 5G services to the market. The emulator complements the capabilities of the TM500 test mobile, enabling comprehensive testing of a 5G gNB in both Non-Stand Alone (NSA) and Stand Alone (SA) modes.
- **TeraVM Core Test:** TeraVM 5GC Test provides fully configurable emulation of thousands of base stations, millions of UE's and user applications to create the most realistic 5G RAN environment to stress the core. It full supports 3GPP interfaces such as N1, N2, N3 and N6 enables the tester to accurately emulate UE applications and mobility behaviour for inter-5G and inter-RAT scenarios.
- **TeraVM O-CU Tester/DU Simulator:** The TeraVM O-CU/O-DU Testers provides a comprehensive validation test suite for the O-CU/O-DU elements of the disaggregated gNB.
- **TeraVM Classic**: TeraVM Classic offers comprehensive cybersecurity and resilience capabilities. It provides scalable real-world application and threat emulation using actual internet threats sourced from well-known Common Vulnerability and Exposure (CVE) repositories. Additionally, TeraVM supports both clientless (SSL/TLS/DTLS) and client (IPsec IKEv1/ v2) oriented Virtual Private Networks (VPNs), enabling performance validation through various algorithms. For those transitioning to NFV, TeraVM's NFV test solution ensures performance, reliability, and predictability for virtualized network functions.

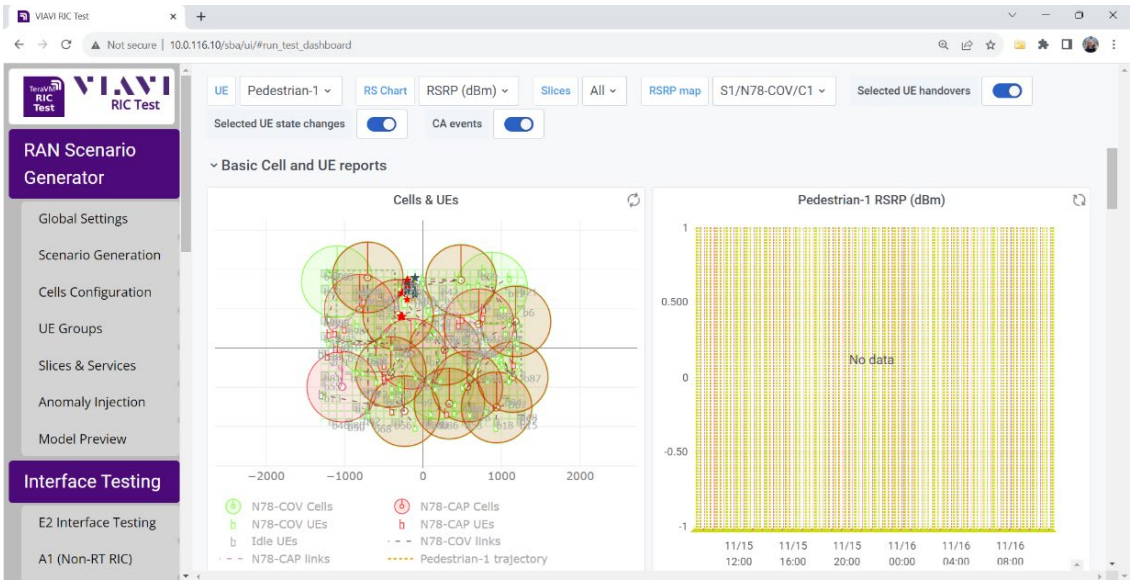


Figure 2-16 Viavi TeraVM RIC tester for O-RAN testing

- **TeraVM Security:** TeraVM is a versatile application and security performance validation solution used for functional and performance testing of network functions and service deployments. It provides comprehensive coverage for assessing vulnerability exposure, security perimeter performance with the latest threats, access/authentication/authorisation with support for a wide range of third-party VPN clients, Single Sign-On (SAML), and endpoint remote access connection (EAP-MD5). Additionally, TeraVM enables validation of various policy management solutions, including Bring Your Own Device (BYOD) and blacklisted URL/DNS, through per-flow emulation of endpoints and applications.
- **TeraVM RIC Tester:** TeraVM RIC Test facilitates emulated RAN measurements for testing RIC efficiency. It allows for testing of proposed changes and assesses if they lead to improvements. This ensures effective testing for RIC developers. TeraVM is a virtualized test tool that can quickly adapt inputs for testing. VIAVI is actively contributing to the expedited deployment of open RAN networks by enabling operators to thoroughly test RIC performance. They've developed a method to test the E2 interface, simulating the volume and scale of messages that would be transmitted. Extending the capabilities of TeraVM is one example of how VIAVI supports network evolution, working closely with operators to ensure cost-effective and efficient testing of networks in preparation for subscriber use [19]. Figure 2-16 shows an example of the VIAVI RIC Tester Graphic User Interface (GUI), showing the radio environment, including Cells and UEs and other metrics for result analysis. The integration of the BeGREEN Intelligent Plane and TeraVM Tester will be analysed withing the context of Work Package 5 (WP5).

B. Keysight

Keysight commercializes a cloud-native solution called RICtest[21] which allows testing of the non-RT RIC (plus rApps) and the Near-RT RIC (plus xApps) by emulating O-RAN network nodes and traffic profiles. It emulates E2 nodes, supporting the different E2 service models and the E2 application protocol standardised by O-RAN, and the O1 interface for managing O-nodes from the non-RT RIC. The testing platform enables the emulation of thousand E2 nodes, eNBs/gNBs and active devices, being able to provide emulations of large 5G NSA and 5G SA topologies. It also provides a GUI which allows topology, UE ranges, traffic profiles and traffic patterns definition.

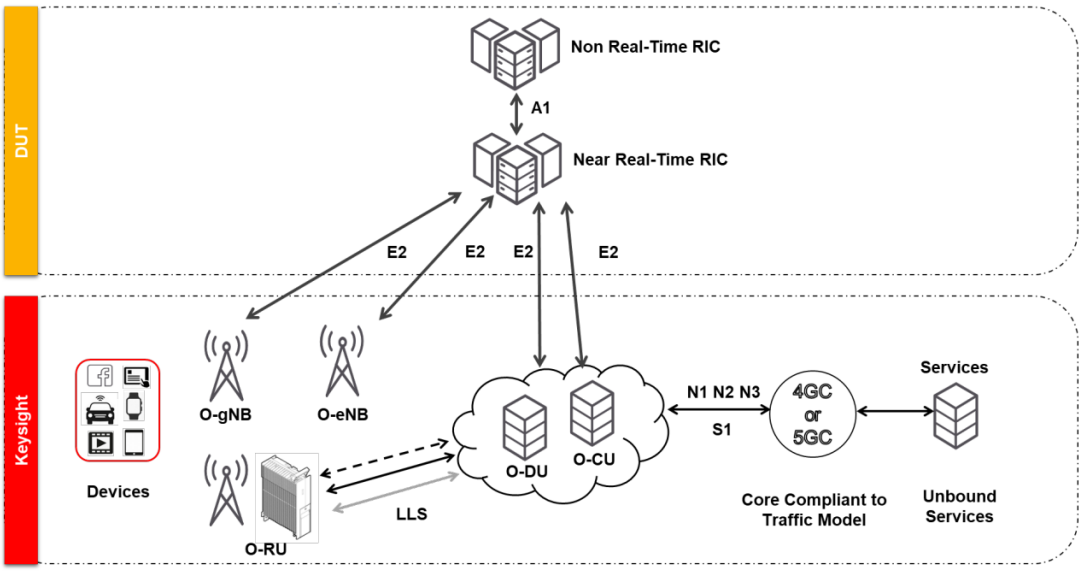


Figure 2-17 Keysight RICtest architecture

[21] https://www.keysight.com/us/en/product/P8828S/rictest-ran-intelligent-controller-test-solutions.html

The RICtest has been used to test O-RAN features in several O-RAN PlugFest and is also being used in different OSC Open Testing & Integration Centres[22] (OTIC). Additionally, it is also part of the infrastructure of one of the Pilot 6G Sites of the SNS project 6G-SANDBOX[23].

2.1.3 Relevant projects

This section presents relevant European projects in which the implementation of an Intelligent Plane or the RICs has a significant role. The main focus is put in H2020 or SNS projects with participation of any partner also involved in BeGREEN, but other pertinent projects are also considered.

- **DAEMON (H2020)**

DAEMON project[24] defines a Network Intelligence Plane (NIP), a collection of modules and interfaces responsible for managing the Network Intelligence (NI) within the network. The NIP is defined in DAEMON as a unified framework based on the MAPE-K (Monitor-Analyse-Plan-Execute over a shared Knowledge) control model to integrate and orchestrate NI components and algorithms at different time scales. Building on top of the N-MAPE-K representation, the DAEMON project dissects NI algorithms into common elements that have different characteristics and introduces original training and closed control loops to define the Network MAPE-K (N-MAPE-K) model (as shown in Figure 2-18).

The DAEMON approach is to move the NIP design from a purely separate plane to a more orthogonal approach where NI Algorithms can effectively be integrated into the traditional planes (data, control, and management) for easy adoption in the industry. To achieve such integration of the NIP, DAEMON also introduces a reference representation of complex NI algorithms to hierarchically organize them in Network Intelligent Services (NISs) that can be broken down into one or more Network Intelligence Functions (NIFs), which, in turn, are composed of atomic NIF Components (NIF-Cs). Among the different types of NIF-Cs, DAEMON defines fundamental NIF-Cs classes such as Sensor NIF-Cs, Monitor NIF-Cs, Analyze NIF-Cs, Plan NIF-Cs, etc. Besides creating a taxonomy of NI algorithms, the DAEMON consortium also brings a set of novel developed NI Algorithms to be integrated in the DAEMON NIP to illustrate how real algorithms would map into the proposed architecture. A few of them address energy consumption at different levels, such as energy-driven vRAN orchestration [20], energy-aware VNF placement [21] or energy-aware scheduling in virtual BS [22].

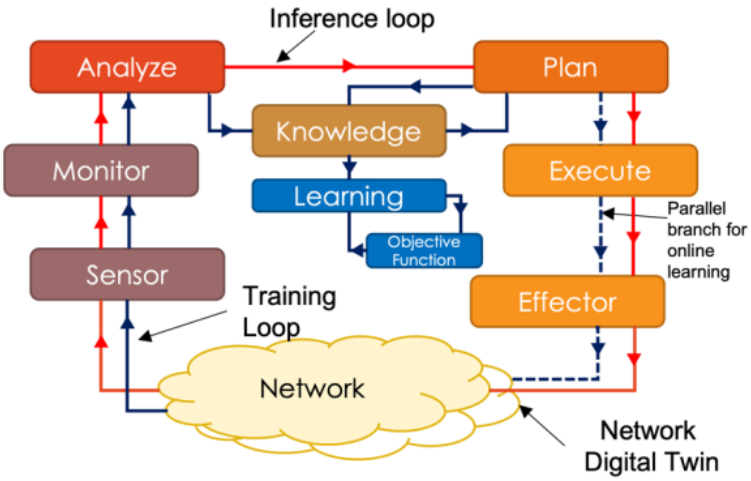


Figure 2-18 Extended N-MAPE-K abstractions for NI algorithms

[22] https://www.o-ran.org/testing-integration#learn-otic
[23] https://6g-sandbox.eu/pilot-6g-sites/malaga/
[24] https://h2020daemon.eu/

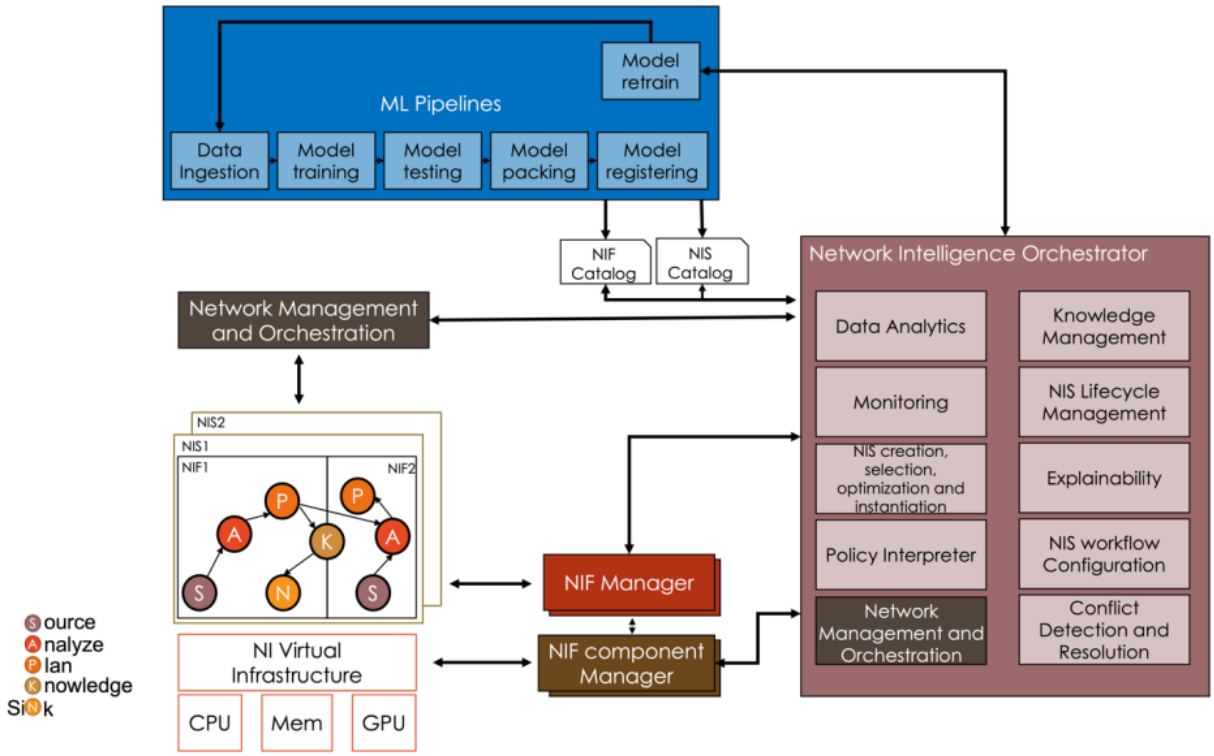


Figure 2-19 The NIP and the functional blocks of the Network Intelligence Orchestrator and ML pipelines

Such a variety of NIF and NIS that can be deployed at the network generates new challenges which may not be possible to address with current management frameworks. DAEMON defines its own Network Intelligence Orchestrator (NIO) based on NFV MANO to address the aforementioned challenges as illustrated in Figure 2-19. While outside the network domain lifecycle management of intelligent algorithms are usually controlled through MLOps, when transferring this view into the mobile network realm, these items cannot be transferred as is, mostly because of the very different timescales that are usually involved in network environments. Therefore, the NIO splits ML items, such as Feature Engineering, Model Training or Model Engineering, into elements that are only related to pure ML tasks in an external ML Pipeline. On the other hand, other elements that need to directly interact with the NIFs in the network, continuously evaluating the quality of the NIS and performing fine-grained lifecycle management of the NIF-Cs, are directly integrated in the NIO and NIP.

- **AI@EDGE (H2020)**

AI@EDGE[25] project aims to develop a serverless connect-compute fabric for creating and managing resilient, elastic, and secure end-to-end slices. Such slices will be capable of supporting a diverse range of AI-enabled applications. AI/ML is also being considered for empowering closed-loop automations focused on orchestrating application and RAN domains. At the RAN side, AI@EDGE leverages O-RAN architecture to provide cross-layer, multiconnectivity, and disaggregated radio access. The project also studies and implements hardware acceleration services at the serverless platform. Figure 2-20 depicts the main building blocks of the AI@EDGE architecture including the closed loops.

[25] https://aiatedge.eu/

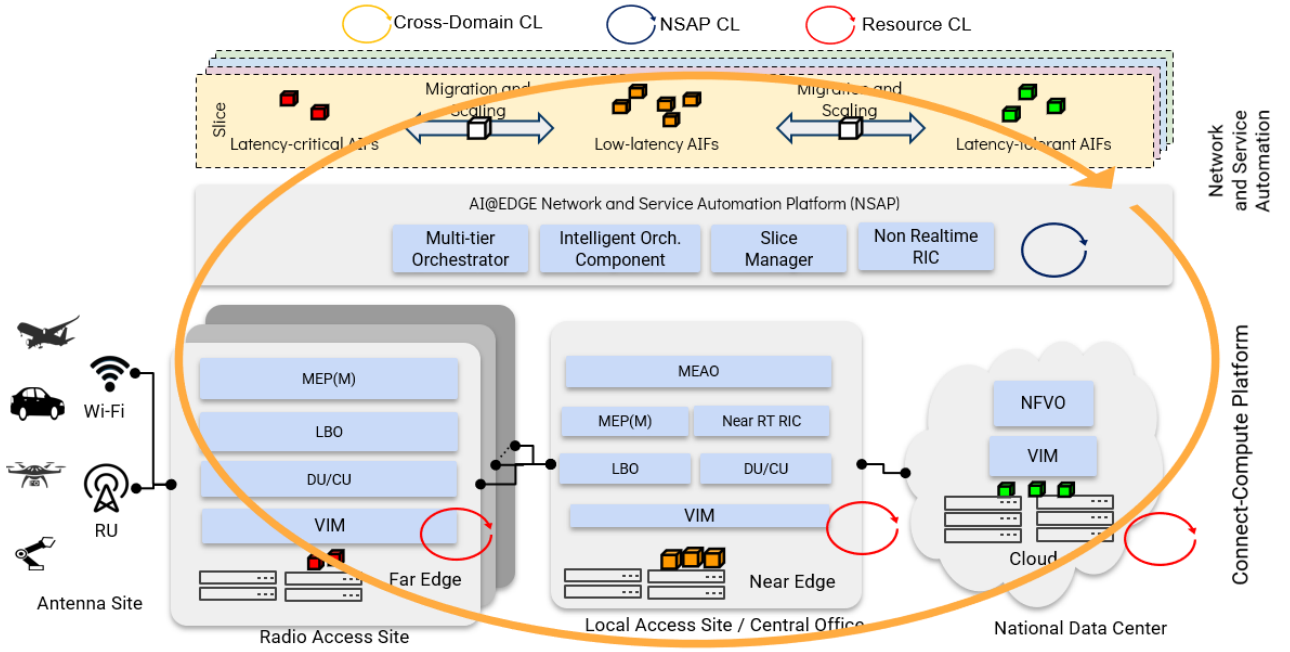


Figure 2-20 AI@EDGE system architecture including closed loops [23]

The Network and Service Automation Platform (NSAP) is conceptually similar to O-RAN's SMO, incorporating also functions to intelligently orchestrate applications and AI Functions (AIF) at the MEC domain. Figure 2-21 shows the main architecture of the NSAP, which includes O-RAN SMO components such as the non-RT RIC, the rApps, R1/A1/O1 interfaces, RAN OAM, etc. It also includes a data pipeline for both data and AI models used by AIFs to train, retrain, and update models according to the AIF descriptor.

Regarding the AIFs, AI@EDGE proposes a conceptual model which defines control and data plane interfaces for allowing, among others, AI model reconfiguration, the exchange of model parameters to support distributed and federated learning scenarios, or the exchange of data on which the ML model is applied. This model is captured through an AIF descriptor, which is exploited by the orchestration and life-cycle management functions and aims at targeting the functional and non-functional requirements related to the AI and ML Ops, distribution, data management performance, and Hardware (HW) acceleration. Finally, the required workflows to manage the AIFs, such as model updating, replacing, or retraining, are also being specified and implemented.

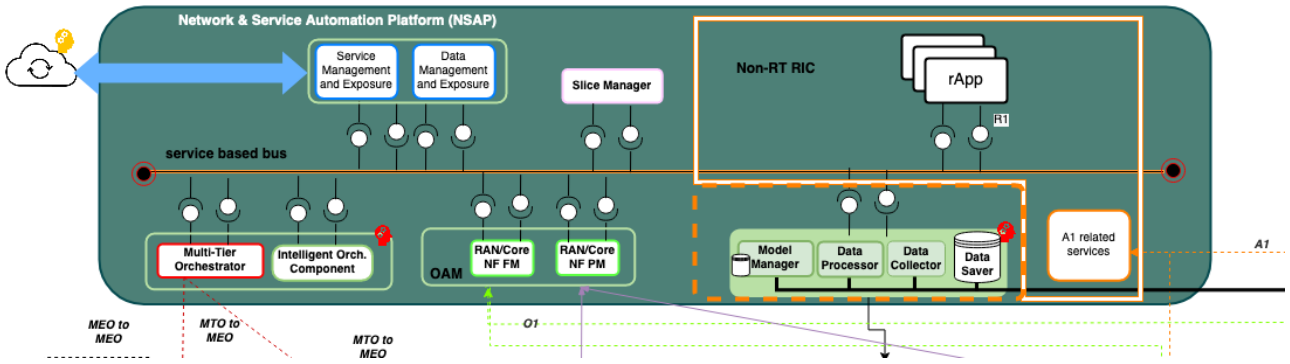


Figure 2-21 AI@EDGE NSAP architecture

- **Affordable 5G**[26]

The large-scale deployment of 5G networks became a reality, as major vendors rolled out 5G equipment and Mobile Network Operators (MNOs) geared up for commercialisation. Concurrently, there was a growing need for specialised network solutions offering widespread coverage with high data rates, dense access points, and enhanced system capacity to complement these high-performing 5G networks. Affordable5G embarked on creating an inclusive 5G network (see Figure 2-22), leveraging technical innovations spanning various aspects of the 5G network. This encompassed cell densification, RU/DU/CU split, hardware acceleration, edge computing, and core network virtualisation, seamlessly integrating open-source RAN, MEC, and MANO solutions for cloud-native, microservice-based deployments.

To realize its ambitious vision, the consortium united ten European Small and Midsize Enterprises, backed by Mobile Virtual Network Operators (MVNO), system integrators, and research institutes. This collaboration offered these entities the opportunity to enhance their products in line with their individual roadmaps while fostering cooperation among them. Affordable5G played a pivotal role in positioning European Small and Midsize Enterprises at the forefront of the global 5G competition, providing support in their commercialisation endeavours and strategies, particularly in niche market segments like neutral hosting, private networks, and MVNOs with new entrant actors.

The innovative solution underwent evaluation and validation through two vertical pilots focused on emergency communications and smart cities. These pilots were meticulously selected for their high representativeness in terms of system performance, scalability, mobility patterns, slice types, deployment requirements, and their anticipated impact on the future 5G market. Within the scope of this project, i2CAT and Accelleran developed an initial integration approach among their RIC platforms.

- **RISE-6G**[27]

Reconfigurable Intelligent Surface (RIS) technology represents a turning-point in the next-generation wireless network design. They offer great versatility when deployed in various objects within the signal propagation environment, including walls, mirrors, ceilings, and appliances. They exhibit remarkable properties, functioning as unconventional reflectors for incoming radio waves and analogue processors for handling multipath scattering. When equipped with the necessary active radio-frequency components, they can fulfil roles as transmitters, receivers, or sensors. These surfaces support a wide range of functionalities, including beamforming, range and position estimation, radio-frequency mapping, and sensing, as well as the detection of obstacles and activities. They are particularly well-suited for minimising electromagnetic field (EMF) exposure, controlling wave propagation and channel geometry, and reducing the transmission power requirements of existing BSs and access points.

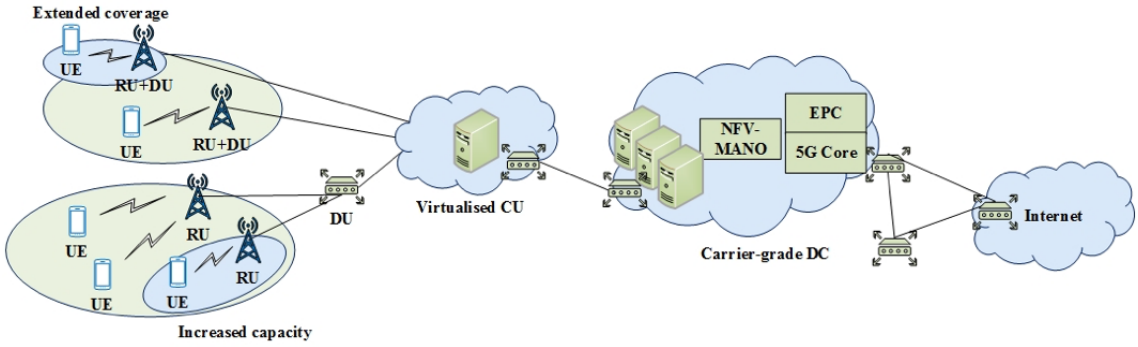


Figure 2-22 Affordable 5G architecture

[26] https://www.affordable5g.eu/
[27] https://rise-6g.eu/

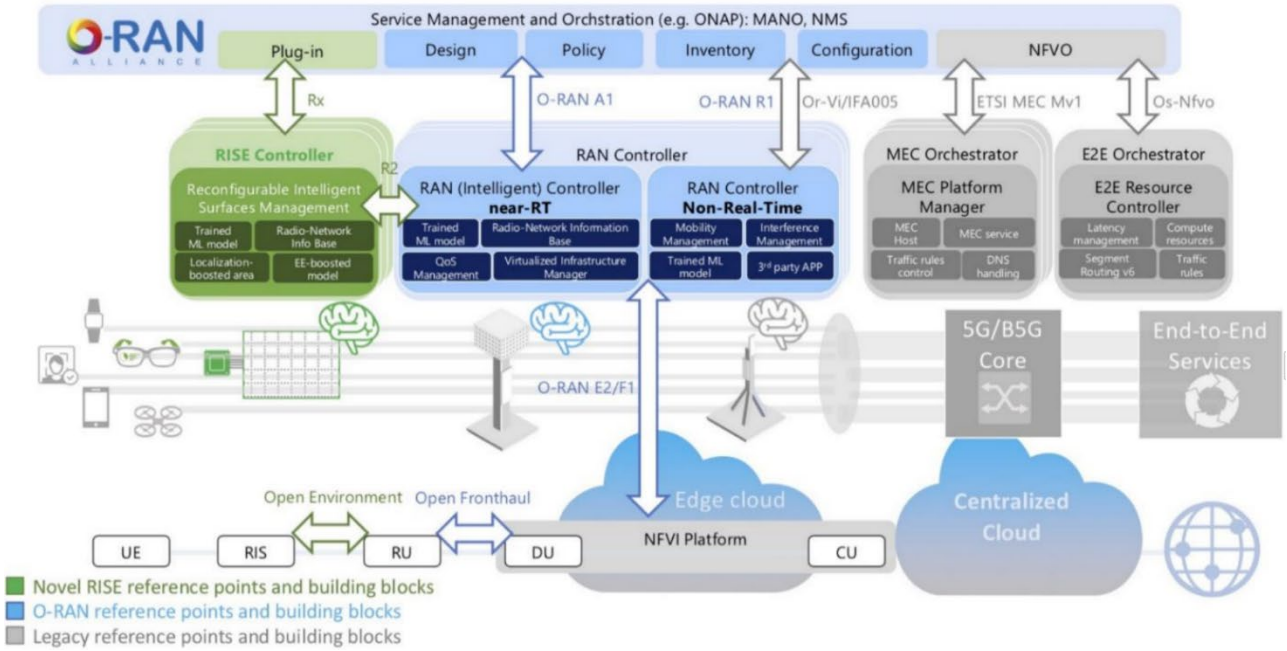


Figure 2-23 RISE-6G architecture integrated within the O-RAN/3GPP/ETSI network architecture

However, the design and implementation of such intelligent and sustainable environments is a major challenge. RISE-6G introduces a groundbreaking concept of Wireless-as-a-Service, where the wireless environment transforms into a service, providing dynamic control over wave propagation in wireless communication. This novel connectivity paradigm combines energy efficient RIS systems with conventional network nodes. Among the considered use cases are enhanced connectivity and reliability, enhanced localisation and sensing, and enhanced sustainability and security. In order to realise this vision, RISE-6G defines the RIS based O-RAN enabled architecture as shown in Figure 2-23.

BeGREEN will leverage the knowledge generated in the RISE-6G project to integrate RIS into the O-RAN/3GPP/ETSI architecture and enable energy-efficient use cases. Finally, RISE6G is also studying how AI can be jointly used with RIS systems. Among the identified approaches are i) supervised training and deployment, ii) unsupervised deployment, and iii) online training and decision making on already deployed RIS systems.

- **NANCY (SNS-JU**)

NANCY[28] project introduces a secure and intelligent architecture for B5G wireless networks. By means of AI and blockchain, NANCY enables secure and intelligent resource management, flexible networking, and orchestration. It considers novel architectures, namely point-to-point connectivity for device-to-device connectivity, mesh networking, and relay-based communications, as well as protocols for medium access, mobility management, and resource allocation.

One of the main objectives of the project is to develop an AI-based wireless RAN orchestration which allows to maximize energy efficiency and trustworthiness, supports ultra-high availability, and optimizes network topology. To this objective, it will leverage O-RAN architecture focusing on device collaboration, RAN sharing and cell-free radio access.

- **VERGE (SNS-JU)**

VERGE[29] will tackle evolution of edge computing from three perspectives: “Edge for AI”, “AI for Edge” and

[28] https://nancy-project.eu/
[29] https://www.verge-project.eu/

security, privacy, and trustworthiness of AI for Edge. “Edge for AI” defines a flexible, modular, and converged Edge platform that is ready to support distributed AI at the edge. This is achieved by unifying lifecycle management and closed-loop automation for cloud-native applications, MEC and network services, while fully exploiting multi-core and multi-accelerator capabilities for ultra-high computational performance. “AI for Edge” enables dynamic function placement by managing and orchestrating the underlying physical, network, and computation resources. Application-specific network and computational Key Performance Indicators (KPI) will be assured in an efficient and collision-free manner, taking Edge resource constraints into account. Security, privacy, and trustworthiness of AI for Edge are addressed to ensure security of the AI-based models against adversarial attacks, privacy of data and models, and transparency in training and execution by providing explanations for model decisions improving trust in models. VERGE will verify the three perspectives through delivery of 7 demonstrations across two use cases: XR-driven Edge-enabled industrial B5G applications across two separate Arçelik sites in Turkey, and Edge-assisted Autonomous Tram operation in Florence. VERGE will disseminate results to academia, industry and the wider stakeholder community through liaisons and contributions to relevant standardisation bodies and open sources, a series of demonstrations showing progression through Technology Readiness Levels (TRL) and by creating an open dataspace for enabling public access to the datasets generated by the project.

- **ARI-5G**[30]

ARI-5G is aimed at implementing, testing, and demonstrating a standards-based RIC platform incorporating specific software solutions through xApps. The primary goal of this project is to validate the deployment of multi-vendor solutions, focusing on advancements in power conservation, energy efficiency, and spectrum management. This includes optimising mMIMO (massive Multiple Input Multiple Output) technology tailored for 5G networks, with the ultimate aim of expediting the successful commercialisation of these innovations.

The main objective of this project is to validate the effectiveness of multi-vendor solutions, particularly in areas of power consumption, energy efficiency, and spectrum management. This validation process will play a pivotal role in ensuring the seamless integration and compatibility of diverse technologies within the RIC platform.

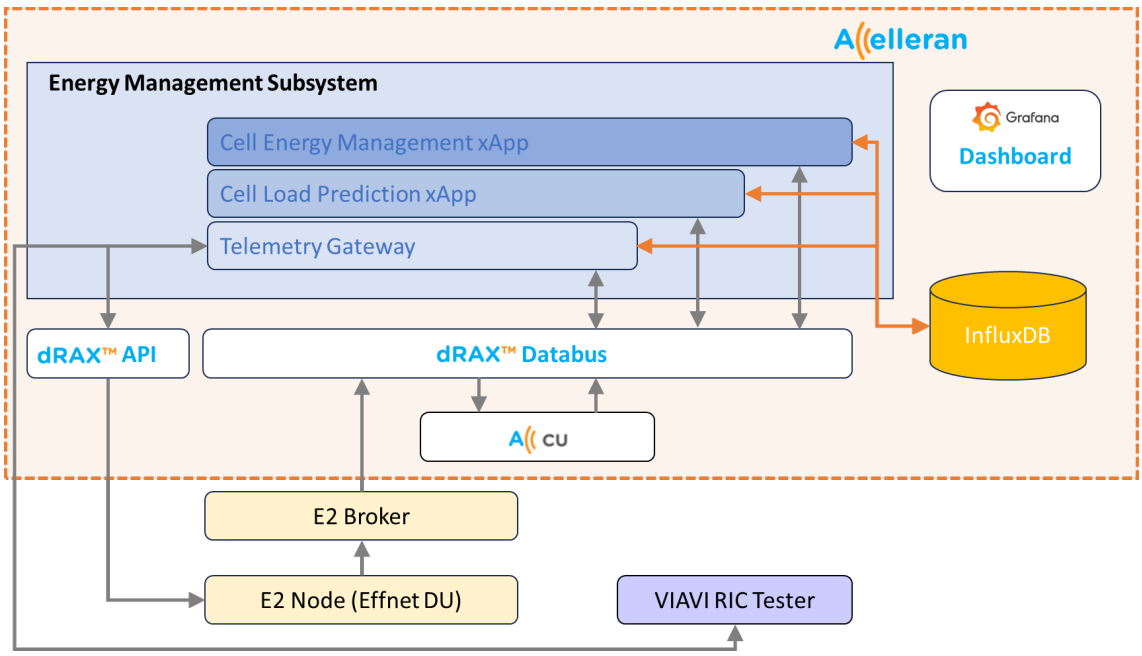


Figure 2-24 General energy management subsystem presented on ARI-5G

[30] https://telecominfraproject.com/accelerating-ran-intelligence-5g/

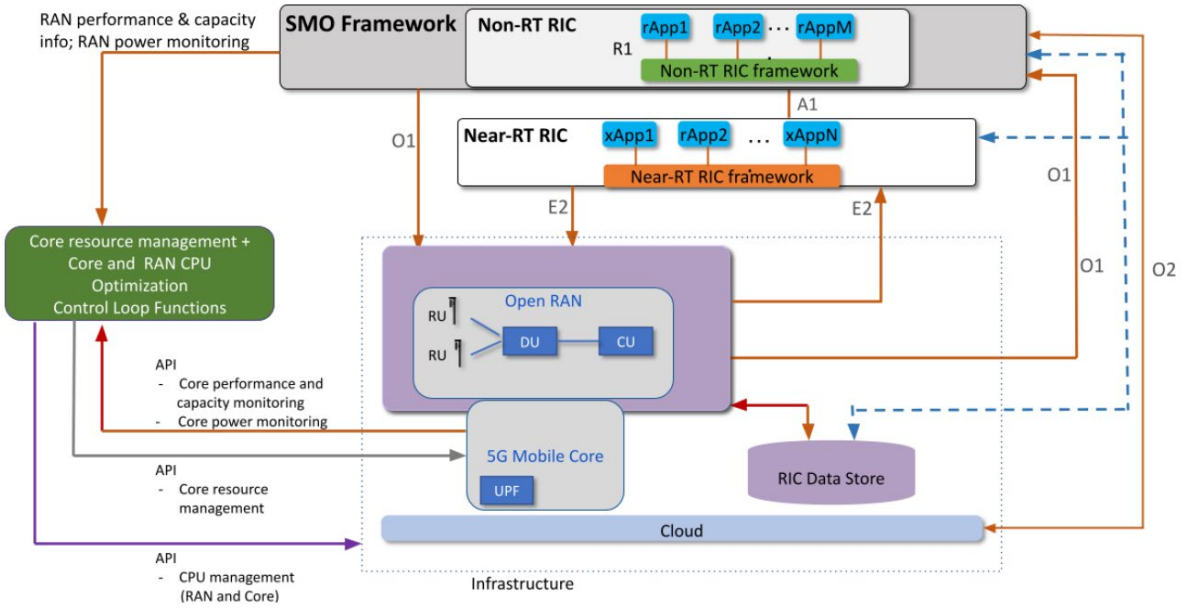


Figure 2-25 ONF SMART-5G PoC target architecture

ARI-5G places a strong emphasis on energy management. The project endeavours to implement solutions that reduce power consumption and enhance energy efficiency, contributing to more sustainable and cost-effective 5G network operations. In Figure 2-24, the general architecture of the Energy Management Subsystem developed is described. A multi xApp system provided energy control by interfacing with diverse RAN environments, and, while predicting future traffic profiles, it controls the status (power on/off) of the cells in the system. Preliminary results show that the target of 10% energy savings is achieved with the proposed algorithms.

- **SMART-5G**[31]

The Sustainable Mobile and RAN Transformation 5G (SMaRT-5G) project is a collaborative effort from the ONF focused on developing, demonstrating and open sourcing ML-driven, intelligent energy savings solutions for mobile networks. The project explores two main approaches to reduce overall power consumption: (i) Optimizing RAN power consumption and (ii) Optimizing compute utilisation for both RAN and Core. RAN power consumption optimisation approaches are focused on Cell/Carrier On/Off switching and Advanced Sleep Modes (ASM). On the other hand, compute power optimisations mainly aim to dynamically adjusting CPU utilisation according to real-time network needs. Both approaches, which are also being considered in BeGREEN project, are aligned with O-RAN energy saving use cases [24].

The project targets the realisation of Proof-of-Concept (PoC) implementations, levering software from other ONF projects such as SD-RAN, SD-Core, Ather and ONAP. Also, it will consider OSC releases. RAN simulator from SD-RAN and ns-3 will be integrated to simulate/emulate the RAN domain in the PoC evaluations. Figure 2-25 shows the presumed PoC architecture, which incorporates the RICs, the rApps/xApps and the resource management functions.

- **OPEN6G**[32]

OPEN6G builds on the Open RAN efforts ongoing today to tackle existing challenges on network automation, Joint Sensing and Communication (JSAC), and the integration of RIS within the Open RAN systems. To do so, OPEN6G sets three main goals: i) exploring the limits of AI-driven network automation in future 6G systems, ii) designing cost-efficient JSAC solutions and iii) designing cost-efficient RIS-enabled JSAC solutions. To

[31] https://opennetworking.org/sustainable-5g/
[32] https://i2cat.net/unico/open6g/

prototype and test the aforementioned solutions, OPEN6G will also design and develop an experimentation platform for novel AI-powered Open RAN 6G JSAC+RIS applications. Through these three main goals, OPEN6G aligns with the future Open RAN 6G research challenges defined the European Commission 6G Vision [25], which includes network of networks, connecting intelligence, sustainability, trustworthiness, extreme experience, and global service coverage.

Finally, Table 2-1 summarizes the list of relevant projects, the involved partners and describes the relationship with the activities carried out in BeGREEN.

Table 2-1 Relevant Projects and Their Relationship with BeGREEN

Project Name	Involved Partners	Relationship with BeGREEN
DAEMON (Network intelligence for aDAptive and sElf-Learning MObile Networks)	NEC, I2CAT	Network Intelligence (NI) at different time scales and NI native architecture. AI/ML at the RICs for orchestration.
AI@EDGE (A Secure and Reusable Artificial Intelligence Platform for Edge Computing in Beyond 5G Networks)	I2CAT	AI/ML for empowering closed-loop automations, joint edge-RAN optimisations, i2CAT's non-RT RIC development, AI/ML pipelines
Affordable5G (High-tech and affordable 5G network roll-out to every corner)	I2CAT, ACC, UPC, RUNEL	Initial integration between ACC's Near-RT RIC and i2CAT's non-RT RIC, RAN telemetry exposure, RAN disaggregation
RISE-6G (Reconfigurable Intelligent Sustainable Environments for 6G Wireless Networks)	NEC	RIS for dynamic and goal-oriented radio wave propagation control and integration of RIS into O-RAN
NANCY (A secure and intelligent architecture for the beyond the fifth generation (B5G) wireless network)	I2CAT	I2CAT's non-RT RIC developments, AI/M-based RAN optimisations
VERGE (AI-powered eVolution towards opEn and secuRe edGe archit Ectures.)	UPC	Development of AI/ML and an edge platform to support distributed AI/ML at the edge.
ARI-5G (Accelerating RAN Intelligence in 5G)	ACC, BT	ARI-5G defines the basis for energy savings management on the RIC. Also, it has presented the Telemetry framework as an interface to transport metrics across all the O-RAN components.
SMART-5G (The Sustainable Mobile and RAN Transformation 5G)	-	Evaluation of Energy Saving UCs using O-RAN RICs: RU on/off, CPU control
OPEN6G (Redes RAN abiertas para los sistemas 6G revolucionarios)	I2CAT, TID, NEC	Development of a O-RAN platform perform network automation, JSAC and RIS-enabled ISAC

2.2 BeGREEN O-RAN based intelligent plane architecture

This section describes the architecture of the Intelligent Plane and details the functionalities of its main components. As shown in Figure 2-26, the Intelligent Plane mainly includes the AI Engine and the RICs, plus rApps and xApps. However, to better understand Intelligent Plane operations and the required interfaces, this section also presents how the Intelligent Plane will interact with the different domains being targeted in BeGREEN energy efficiency optimisations, that is the RAN, including relays and RIS; the 5GC; and the Edge. As depicted in Figure 2-26, the proposed architecture leverages O-RAN Alliance baseline architecture and extends it by including the AI Engine to support the application of AI/ML. The services of the AI Engine will be exposed to the RICs and the rApps/xApps through additional interfaces not included in the O-RAN specification.

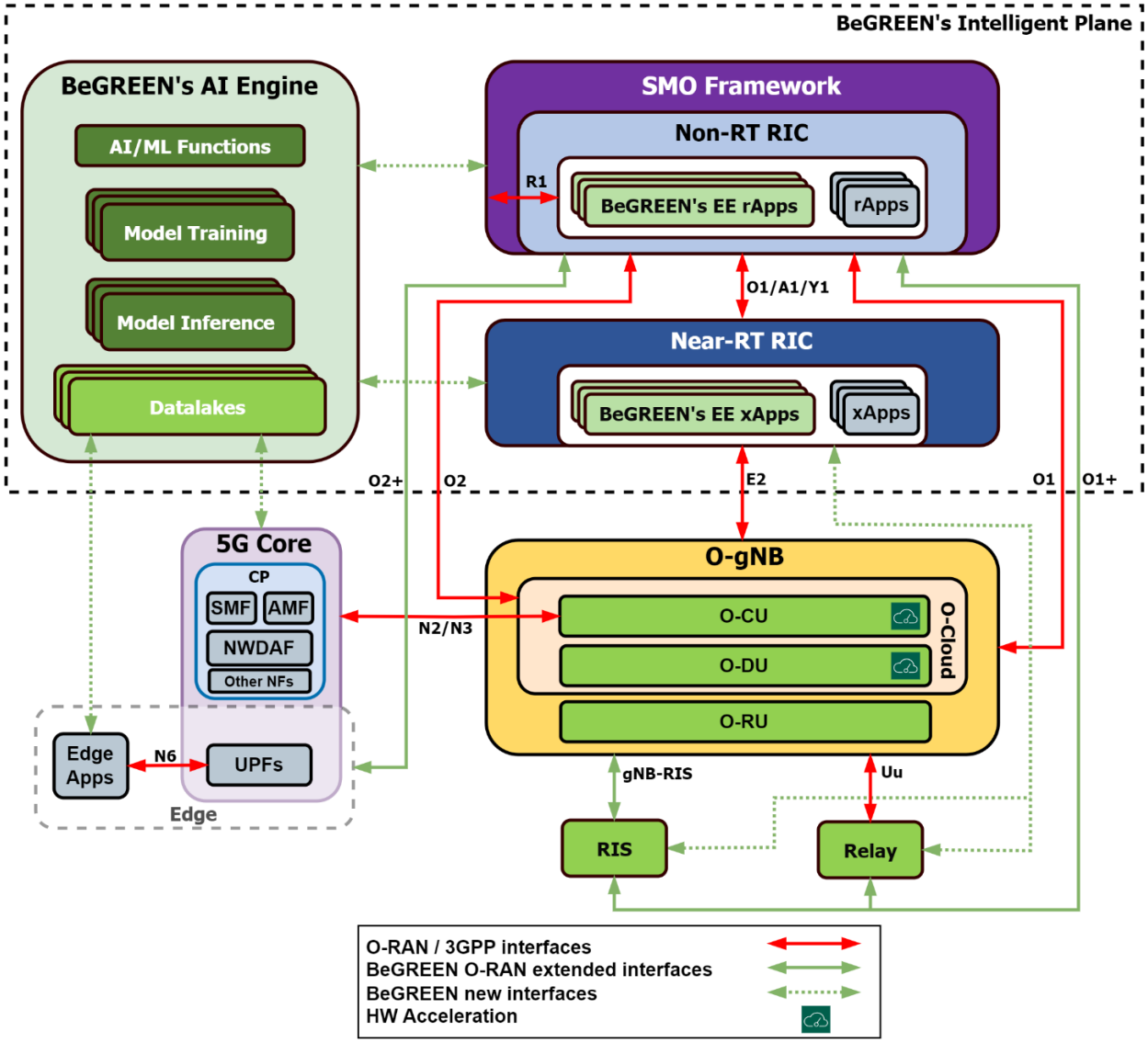


Figure 2-26 BeGREEN architecture including the Intelligent Plane

Similar approach is taked for the interfaces to/from the Edge, the RIS and Relays, since nowadays these components are not considered within the standard O-RAN architecture; however, we envision that some functionalities related to these components will leverage existent O-RAN interfaces such as O1 or O2 (denoted as O1+ and O2+ in Figure 2-26). Additional details on the different components and interfaces can be found in the next sections. Note that, compared to the architecture provided in BeGREEN D2.1 [1], on the one hand, this version simplifies the representation of the RICs and the rApp/xApps in order to relate it with the common representation of the O-RAN architecture; nevertheless, concrete details on the RIC components can be found in the subsequent subsections. On the other hand, the components and functionalities of the AI Engine are more detailed compared to the previous version in BeGREEN D2.1 since the specification of its main functionalities and its relationship with the other components of the BeGREEN architecture has been one of the main outputs of this first phase of WP4.

2.2.1 Intelligent Plane components

This section provides a comprehensive description of the architectural components of the BeGREEN Intelligent Plane: AI Engine, non-RT, and Near-RT RICs.

2.2.1.1 AI Engine

The AI Engine, whose architecture and integration with the RICs is depicted in Figure 2-27, is a key component in the Intelligent Plane. First, it implements the necessary AI/ML workflow services to provide AI/ML support to rApps/xApps. Secondly, it offloads workloads from the RICs by hosting the ML models, which are trained and served within the AI Engine framework, and other functions which may be used intensively by the rApps/xApps, such as the Energy Score calculation. As illustrated in Figure 2-27, these functions may be implemented as serverless. Finally, the AI Engine will also host the AI datalake(s) to provide the required data to the hosted AI/ML models and offloaded functions.

As introduced in the previous paragraph, the proposed AI Engine follows a loosely coupled approach, where the AI/ML models are hosted in the AI Engine instead of being embedded in the rApps/xApps needing their outputs. This approach, which is being considered as one of the options to implement AI/ML in O-RAN architecture [11], enables ML models that are managed independently by rApps/xApps, facilitating operations like monitoring or retraining. Indeed, any rApp/xApp can have access to the ML model outputs which are exposed as services or data types (e.g., access to a load prediction for a specific cell), allowing model reutilisation. Additionally, this approach also allows deploying the ML models for training or inference in servers or clusters different of the RICs, enabling offloading through serverless computing or through hardware acceleration.

In order to expose the AI/ML workflow services of the AI Engine to the rApps/xApps, in BeGREEN we will introduce the concept of "AI Engine Assist rApps/xApps". These rApps/xApps will be linked to a ML model, exposing its outputs to other rApps/xApps (i.e., working as data producers) and assisting the communication between the AI Engine and the RICs for other procedures such as ML monitoring or retraining. The communication between these rApps/xApps and the AI Engine will be based on the definition of a common interface, although each one will be individually in charge of its ML model needs, e.g., required input parameters and data to do training or inference, triggered pipeline, etc.

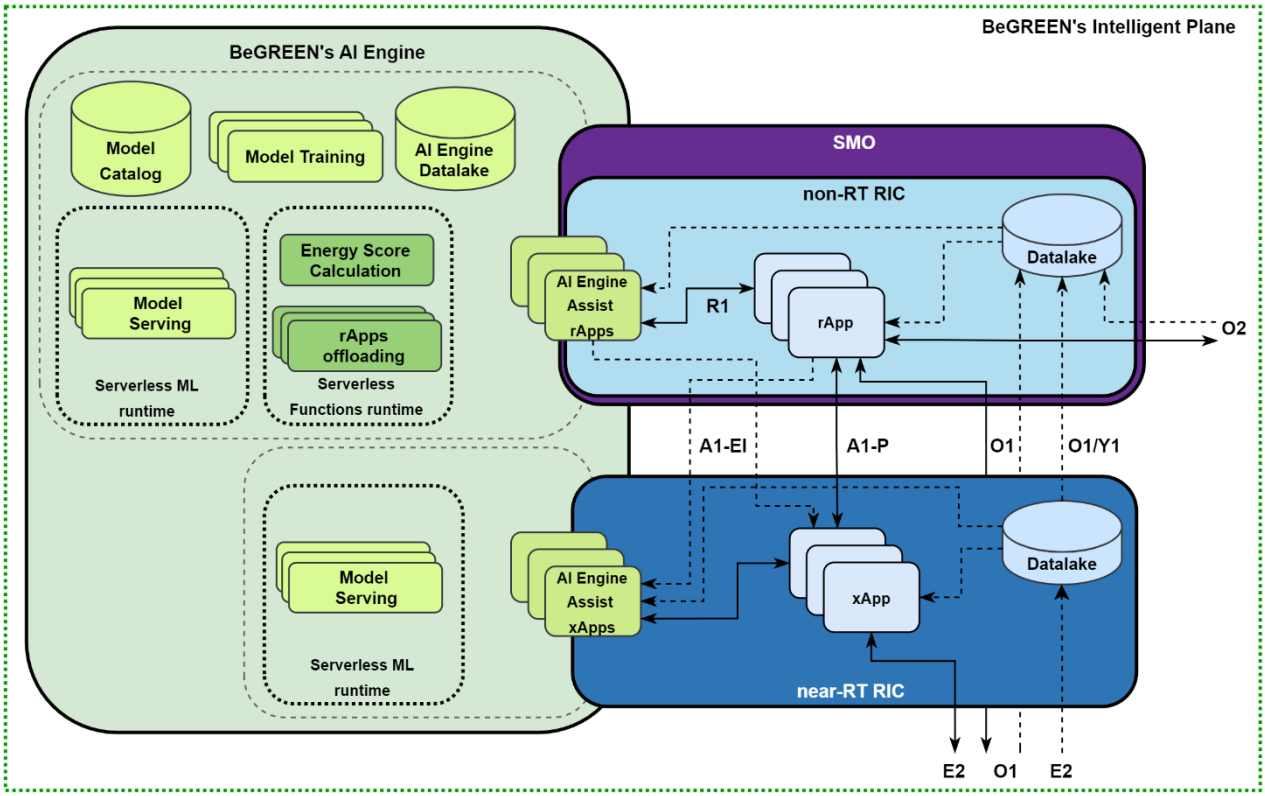


Figure 2-27 Detailed architecture of the AI Engine and its relationship with the RICs

We envision four main scenarios or AI/ML workflows:

- ML model creation and training: The model developer obtains the required data in the AI Engine datalake by interacting with the RICs (e.g., deploying a rApp or xApp producing and exposing the data). After analysing the data, it generates the model and trains it. Once trained, the developer publishes the model in the catalog and creates the inference pipeline plus, optionally, monitoring, and retraining pipelines. Finally, the developer creates the "AI Engine Assist rApp/xApp" which can interact with the created pipelines in the AI Engine through the AI Engine-RIC interface and acts as ML model producer.
- ML model inference (non-RT RIC level): rApp developer/deployer creates/deploys a control rApp which subscribes to the ML model outputs exposed by the "AI Engine Assist rApp" through the R1 interface. The "AI Engine Assist rApp" generates the inference data through the inference pipeline of the ML model in the AI Engine, which is deployed in a serverless way. The rApp gets the ML output and generates an action through O1/O2/A1/R1 interfaces. Alternatively, the xApp developer/deployer can create/deploy a control xApp which subscribes to the ML model outputs exposed by the "AI Engine Assist rApp" through the A1-EI interface. In this case, the xApp gets the prediction and generates an action through E2 interface.
- ML model inference (Near-RT RIC level): xApp developer/deployer creates/deploys an xApp which subscribes to the of ML model outputs exposed by the "AI Engine Assist xApp" through the Near-RT RIC. The "AI Engine Assist xApp" gets the inference data through the inference pipeline of the ML model in the AI Engine, which is deployed in the same server or in a near server (e.g., same edge resources) to assure near-RT runtime. Finally, the xApp gets the ML model output and generates an action through E2 interface. Alternatively, inference outputs may be exposed to northbound (e.g., the RIC) or to external components (e.g., application servers) through the Y1 interface.
- ML model monitoring and retraining: According to the monitoring pipeline, the AI Engine triggers the retraining of a ML model. The retraining is supported by data from the RICs, which is obtained through the "AI Engine Assist rApp/xApp".

In future WP4 deliverables, we will refine and detail these workflows and the required operations by the involved components. Some of these components that comprise the AI Engine will be based on existent open-source frameworks or solutions devoted to MLOps, serverless platforms, and datalakes. An initial list of the main considered solutions is listed below. Additionally, as illustrated in Figure 2-27, the Energy Score calculation will be one of the main offloaded tasks in the AI Engine due to its relevance within the Intelligent Plane. Therefore, a detailed description of its definition is also included.

A. Machine Learning Operations (MLOps)

As the field of AI/ML grows more and more complex, so does the need for tools to simplify working with these technologies. MLOps emerges as a key solution to the complex challenges of training, deploying, monitoring, and maintaining ML models in production environments. MLOps aims to streamline the entire lifecycle of an ML project, ensuring that models are not only developed efficiently but also deployed and managed effectively [26].

MLOps frameworks are characterized by their emphasis on automation, scalability, and reproducibility, which is why most of them are integrated with orchestration platforms such as Kubernetes. A description of two relevant MLOps frameworks from the SotA which are being considered as baseline for the AI Engine implementation is presented:

- **Kubeflow** [33] is an open-source MLOps platform, built with the intention of simplifying the

[33] [1] https://www.kubeflow.org/

deployment, orchestration, monitoring, and execution of ML workloads on Kubernetes. The aim of Kubeflow is to make scaling machine learning models and deploying them to production as simple as possible, harnessing the power of Kubernetes' scalable and flexible infrastructure.

At its core, Kubeflow provides a suite of tools and services that seamlessly integrate with each other. It enables the creation and handling of interactive Jupyter notebooks, a staple in data science exploration and model prototyping. For the construction and operation of ML workflows, Kubeflow offers a robust pipeline system that allows for the deployment and management of complex ML workflows, ensuring that every step from data preprocessing to model training and evaluation is systematically coordinated.

A key feature of Kubeflow is its built-in service for hyperparameter tuning named Katib, which refines machine learning models for peak performance. It also incorporates KFServing, a component designed to serve ML models efficiently by employing a serverless approach, which facilitates autoscaling of resources and enables advanced deployment strategies. Kubeflow's architecture also includes a Metadata Store, a repository that meticulously tracks and governs the metadata associated with ML workflows. This ensures transparency and reproducibility in ML experiments. Moreover, it supports distributed ML training jobs through specialized training operators that support to a variety of ML frameworks such as TensorFlow, PyTorch, and MXNet, as illustrated in Figure 2-28, thereby promoting flexibility and scalability in training complex models.

- **MLRun**[34] is another open-source MLOps platform, similar to Kubeflow, which is also designed to streamline the machine learning lifecycle on Kubernetes. ML covers the whole ML pipeline, from data ingestion and processing to deployment and monitoring of models in production environments. It provides an intuitive interface for managing the execution of various ML tasks and allows users to track experiments, manage data artifacts, and deploy models with high efficiency and minimal overhead. Figure 2-29 depicts the main architecture, workflows, and tools of MLRun.

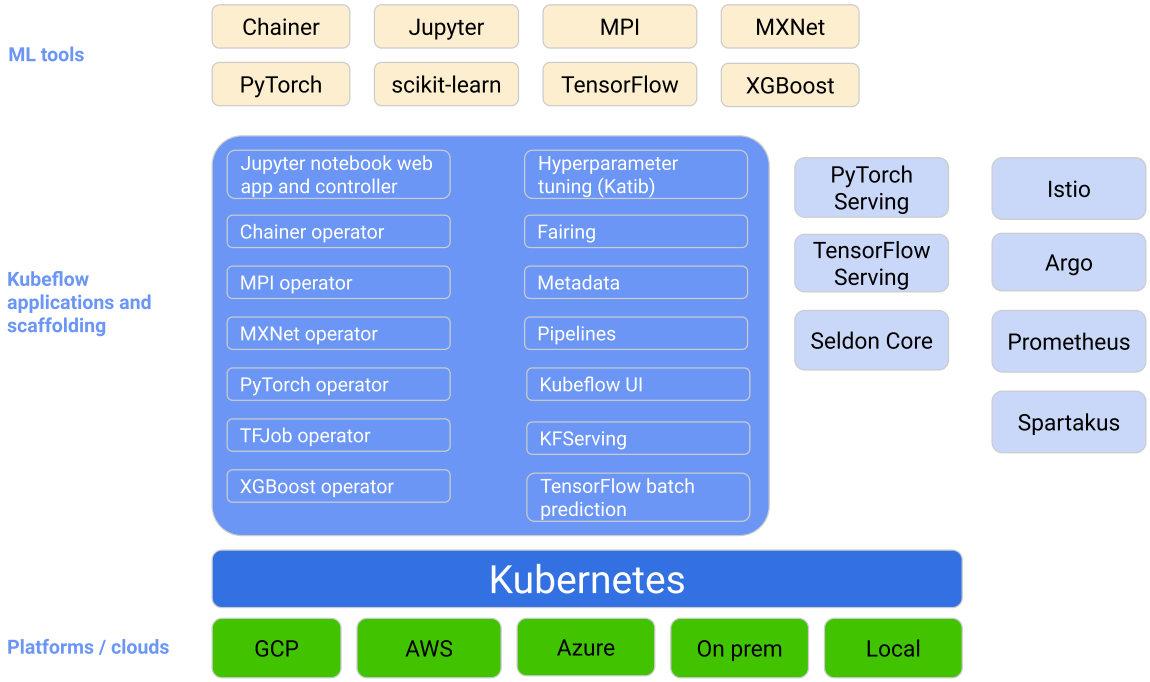


Figure 2-28 Kubeflow conceptual architecture

[34] https://www.mlrun.org/

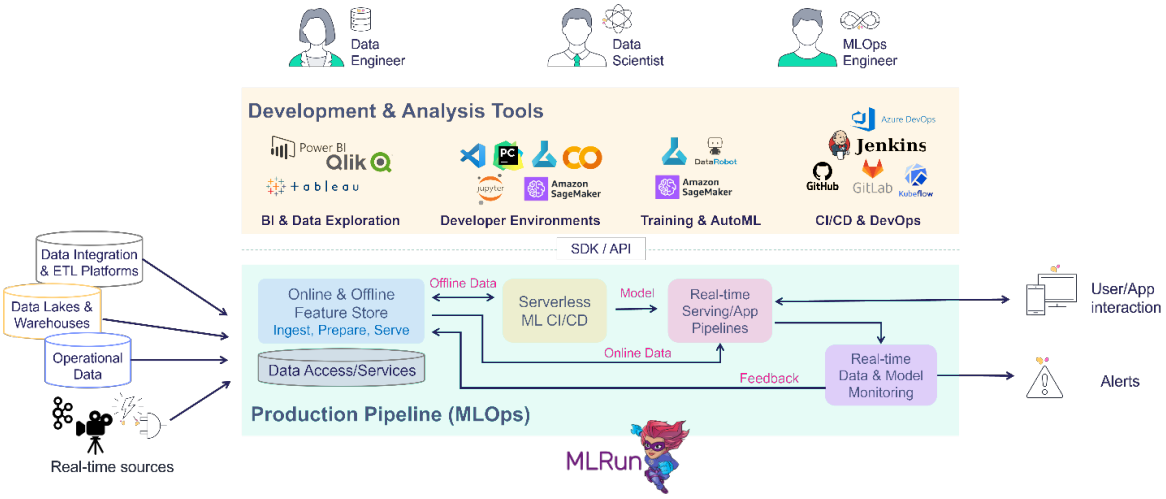


Figure 2-29 MLRun main architecture

One of the distinguishing features of MLRun is its ability to automate and monitor ML pipelines, ensuring that each component of the pipeline is executed in the correct order and environment. This is particularly beneficial when dealing with complex data transformations and model training processes that require careful coordination and resource management. Additionally, MLRun offers advanced model monitoring capabilities, which are essential for maintaining the performance and accuracy of machine learning models once they are deployed in production. It incorporates features that enable continuous monitoring of model behaviour, detection of drift, and triggering of retraining workflows, when necessary, thereby ensuring models remain relevant and effective over time. Additionally, MLRun allows serverless automation through Nuclio.

B. Serverless frameworks

Serverless function services bring together two closely related concepts, namely (i) serverless, a cloud computing execution model where the cloud provider provides the whole server architecture and the customer provides front-end application code to be executed -sometimes also referred to as Backend-as-a-Service (BaaS)-; and (ii) Function-as-a-Service (FaaS), where the focus is on the development of functions (atomic services that are smaller than microservices) while the server side is provided by the FaaS provider. These concepts allow scaling to reduce capacity to zero in times of no demand which is beneficial where pricing is based on the actual amount of resources consumed by an application, rather than on pre-purchased units of capacity.

- **OpenFaas** [35] is an open-source implementation of a FaaS architecture which offers a flexible architecture which can support a wide variety of different functions and includes built-in language support for C Sharp, Go, Java, Node, PHP, Ruby, and Python.

  In contrast to commercial products including Microsoft Azure, Google Cloud and Amazon Web Services (AWS) Lambda, OpenFaaS offers FaaS functionality as open source. OpenFaaS can be set up in the cloud or locally and can run on top of Kubernetes or Docker Swarm. When set up locally, with the availability of suitable hardware OpenFaaS can be used with no costs and provides full control of the deployment.

  While the concept of FaaS, and OpenFaaS in particular, was developed for generic function serving rather than specifically for ML model serving, it is well suited for ML model deployment as well. Here, OpenFaaS allows to easily deploy ML model code from any language, even pre-existing legacy code

[35] https://www.openfaas.com/

can be brought into OpenFaaS function format. Some languages are supported directly by OpenFaaS,

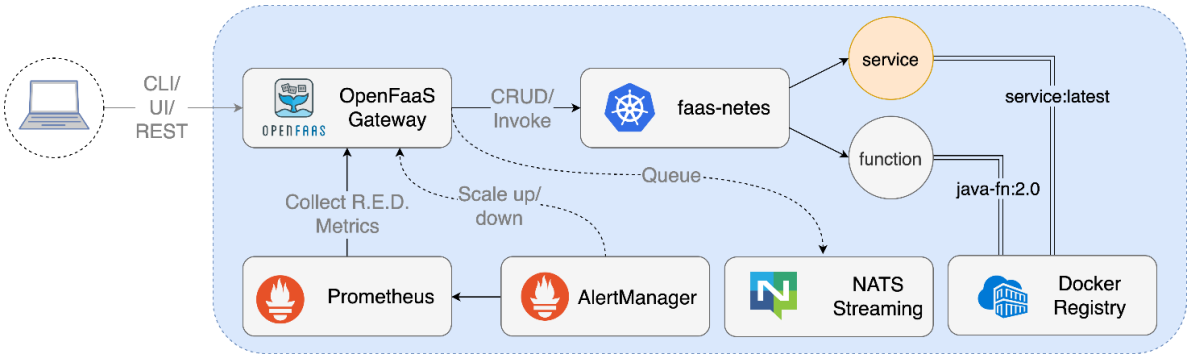


Figure 2-30 OpenFaas architecture

but theoretically ML models of any language can be deployed because the ML models are wrapped into Docker containers. The use of Docker containers not only allows for language independent ML model deployment but also for flexible deployment and scaling.

The functions deployed in OpenFaaS are accessible to other components over REST endpoints that can be called by the other elements in the BeGREEN architecture subject to connectivity and deployment considerations.

The OpenFaaS architecture allows flexibility for different rApps or other components to specify and create their own docker containers to implement various functions and offload these functions to the AI Engine rather than executing within the rApp itself. These functions can be used by other rApps too where the functionality is common to different use cases (e.g., calculation of energy score in terms of bits per joule might be an example of a function that could be implemented in OpenFaaS and used by multiple components).

- **Nuclio**[36] is an open-source and managed serverless platform used to minimize development and maintenance overhead and automate the deployment of data-science based applications. An open-source version of Nuclio is available but it excludes some features like auto-scaling and integration with ML tools. Nuclio's core component is the function processor (written in Go). This processor works through abstract interfaces and works as the function's operating system providing access to events, data, logs, etc. The same function code can be fed from a variety of pluggable event sources (currently supporting Hypertext Transfer Protocol (HTTP), Kinesis, Kafka, RabbitMQ, MQTT, NATS, iguazio's V3IO, and emulators). Notable differences can instead be observed when looking at how considered FaaS platforms document their development and architecture. Open-source solutions typically document the architecture of the platform and its development. OpenFaaS fully documents both its architecture and development while Nuclio documents its architecture, but it only provides guidelines to contribute to its development [27].

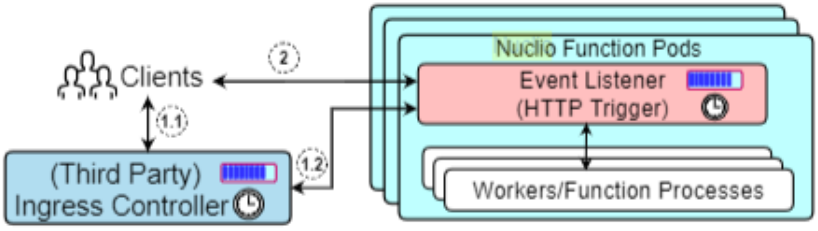


Figure 2-31 Nuclio architecture

[36] https://nuclio.io/

The main components of Nuclio are shown in Figure 2-31. In each function pod, there is one event listener and multiple worker processes. The event listener receives new events and redirect them to worker processes. Multiple worker processes could work in parallel and improve the performance significantly on a multi-core worker node. The worker process number is set to be static and specified by the configuration file. The open-source version does not have a built-in workload-based auto-scaling feature, but the resource-based auto-scaling is supported by Kubernetes Horizontal Pod Autoscaling (HPA). Nuclio supports two ways to trigger functions: (1) invoking the function by name through ingress controller, which can distribute the traffic to different back-end pods according to the pre-set load balancing rule (e.g., round-robin, random and least connection first) and (2) sending requests directly to function pods by Node Port, which is a unique allocated cluster-wide port for the function. In the Node Port method, incoming requests are load balanced at random by the Net filter.

C. Datalake

BeGREEN will implement a datalake to store the data from the network connected to the architecture. Some of the functions in the architecture will consume real-time telemetry. The datalake will provide data storage and make it available to the functions that require to consume non-RT telemetry.

The datalake architectural pattern consists of a large data repository that allows storing of structured and unstructured data, and which is generally scalable based on user needs. A datalake combines this large-scale storage repository with a variety of high-performance processing engines that can be virtualized. Independent scalability of processing and storage is a useful consideration for efficient deployment. A datelake can store the required data as-is, without transformation. Transformation and pre-processing can be carried out as necessary when different types of analytics from Structured Query Language (SQL) queries, real-time analytics, and machine learning to make data driven decisions are required.

Datalakes are useful for providing access to the same underlying data to different categories of users each with their own preferred tools and frameworks. Indeed, it can provide data access to an analytics system which allows analytics to be produced without the need to move data.

Datalakes can allow large data storages to be tiered based on access frequency. It is possible to implement data access strategies based on tiering for efficiently using storage technologies according to the business value or importance of the data. A tiered storage system provides several types of storage comprising different physical types of storage media. In some cases, the tiering can be automated based on last data access information and tiering strategies may be based on the availability, and performance of the object storage.

- **MinIO**[37] is a high-performance, S3 compatible object store. It is built for large scale AI/ML, datelake and database workloads. It is software-defined and runs on any cloud or on-premises infrastructure. MinIO is dual-licensed under open-source GNU AGPL v3 and a commercial enterprise license. Its storage stack has three major components: MinIO Server, MinIO Client (a.k.a. mc, which is a command-line client for the object and file management with any Amazon S3 compatible servers), and MinIO Client SDK, which can be used by application developers to interact with any Amazon S3 compatible server.

  MinIO also provides a variety of deployment options. It can run as a native application on most popular architectures and can also be deployed as a containerized application using Docker or Kubernetes. MinIO is open-source software.

- **InfluxDB** [38] is an open-source Time Series Database (TSDB). It is specialized in operations like monitoring, application metrics, Internet of Things (IoT) sensors data and real-time analytics. It is

[37] https://min.io

[38] www.influxdata.com

mainly written in Go language and is designed for high-performance and high-efficiency storage. It can store thousands of data points every second making it perfect for industrial grade applications. Once stored, data can be queried and analysed using the "Flux" language. This language is an integral part of InfluxDB and permits advanced data manipulations to analyse your data in depth.

- **Prometheus**[39] is an open-source system monitoring and alerting toolkit. Prometheus collects and stores its metrics as time series data, i.e. metrics information is stored with the timestamp at which it was recorded, alongside optional key-value pairs called labels. Prometheus scrapes metrics from instrumented jobs, called Prometheus exporters, either directly or via an intermediary push gateway for short-lived jobs. It stores all scraped samples locally and runs rules over this data to either aggregate and record new time series from existing data or generate alerts. Grafana or other API consumers can be used to visualize the collected data.

D. Energy score calculation

Different ways of defining and measuring the energy efficiency can be found in the literature. On the one hand, 3GPP defines a KPI that shows mobile network data energy efficiency in operational NG-RAN as the Data Volume (DV) divided by the Energy Consumption (EC) of the considered network elements [28]. This metric is represented in bits/J. On the other hand, [28] the International Telecommunication Union (ITU) defines energy efficiency as 'the relationship between the specific functional unit for a piece of equipment (i.e., the useful work of telecommunications) and the energy consumption of that equipment. For example, when transmission time and frequency bandwidth are fixed, a telecommunication system that can transport more data (in bits) with less energy (in Joules) is considered to be more energy efficient [29]. Similarly, ETSI cites a Next Generation Mobile Networks (NGMN) 2015 white paper definition of energy efficiency [30] "*Energy efficiency is defined as the number of bits that can be transmitted per Joule of energy".* A large number of references from the literature refer to bits/Joule as standard measure of energy efficiency used in telecommunications [31][32][33][34][35][36].

According to the previous definitions, the BeGREEN project will calculate the energy efficiency as the measured data volume measurements divided by energy measurements. The measurements of data volumes in the network will be obtained from the performance statistics for the relevant network entities. The measurement of energy consumption will also be obtained from the performance statistics for the relevant network entities, and it will only be possible to calculate energy efficiency for network entities for which both energy consumption and data volume measures are available.

Network entities typically collect measurements of data volume or throughput as part of their performance statistics. For each relevant network entity, a measurement will be chosen as the metric to calculate the energy efficiency for a given level of power consumption. Other metrics were considered such as Physical Resource Block (PRB) utilisation, but data volume or throughput is the most universally applicable metric to be comparable across energy-saving methods used by the different types of elements in the network and are used in the telecommunications standards as detailed above.

We must account for length of period for the throughput measurement because Mbps is megabits per second (a time-relative measurement) whereas kWh is directly convertible to joules (1kWh = 3.6MJ). The Energy Score *Es* can be calculated as Data Volume *Dv* divided by Energy Consumption *Ec* as shown below:

$$Es = \frac{Dv}{3.6 \cdot Ec} \quad (2\text{-}1)$$

The Data Volume can also be calculated as:

$$Dv = Dv_{DL} + Dv_{UL} \quad (2\text{-}2)$$

[39] prometheus.io

Where Dv_{DL} and DV_{UL} represent the DL and UL Data Volume.

Alternatively, the Data Volume can be calculated as:

$$\boldsymbol{Dv = (Th_{DL} + Th_{UL}) \cdot \Delta T} \quad (2\text{-}3)$$

where Th_{DL} and Th_{UL} are the average downlink and uplink throughput measured in the gNB, respectively, and ΔT is the reporting period.

An energy rating metric will also be calculated which will be a relative percentile score for the energy efficiency of a specific class of equipment. For example, the energy score of a gNB will be its energy efficiency in bits per joule and the energy rating of the gNB will be derived from its energy score relative to the energy scores for all other gNBs connected to the BeGREEN architecture.

It will be possible to provide an overall metric for components where energy ratings are calculated for sub-components. E.g. if energy rating can be calculated per cell, then there will also be an energy rating for the node containing those cells and for a network containing those nodes. These aggregate energy ratings can be weighted based on the energy consumption levels of the underlying sub-components.

Energy saving techniques applied during the project must ensure that they maintain or improve user experience and network service levels as well as improving energy efficiency.

2.2.1.2 SMO and non-RT RIC

The main objective of the SMO and non-RT RIC components in BeGREEN will be to expose and abstract to the rApps the required functionalities to implement intelligent control loops targeting energy efficiency. Therefore, as illustrated in Figure 2-32, within the non-RT RIC architecture, we considered the following five main subsystems:

- rApp subsystem: Hosts the rApps and exposes to them the functions or services provided by the other subsystems through the R1 interface. This way, the rApps just implement the logic of the intelligent control loops and apply the optimisations or policies according to the exposed services, without requiring a specific knowledge about the non-RT RIC implementation. This is aligned with the general principles and objectives of the R1 interface as specified by O-RAN [4] . In addition, as introduced in the description of the AI Engine, this component will be used to offload some common rApp functions and to implement AI/ML services accessible by the rApps.
- Data exposure subsystem: Hosts rApps performing as data producers and consumers. Data exchange is managed by the ICS from ORAN's OSC [37], which architecture is depicted in Figure 2-33. The ICS decouples data consumers and data producers by managing data subscriptions or jobs. The ICS defines various types of data or information, which can be accessed through one-to-many producer rApps that are also registered within the ICS. When a consumer rApp expresses interest in one or more specific data types, it establishes a data subscription or job through the ICS. The ICS then manages the available producers to initiate the delivery of the requested data. As shown in Figure 2-33, exposed data can be also consumed by the Near-RT RIC and its xApps, through the A1-EI interface. The data made accessible can encompass a range of sources, including RAN telemetry from O-nodes or E2-nodes, as well as external data (e.g., from the 5GC or the Edge) or newly generated data by other rApps, including AI/ML-based predictions.

  The proposed architecture also includes a REDIS database in order to allow an efficient storing and sharing of specific processed data such as datasets to train AI/ML models, inference outputs which might be useful for several rApps, or AI/ML model KPIs (e.g., precision or recall). The data in this database could be exposed through producers or being accessed directly by rApps with the required authorisation.

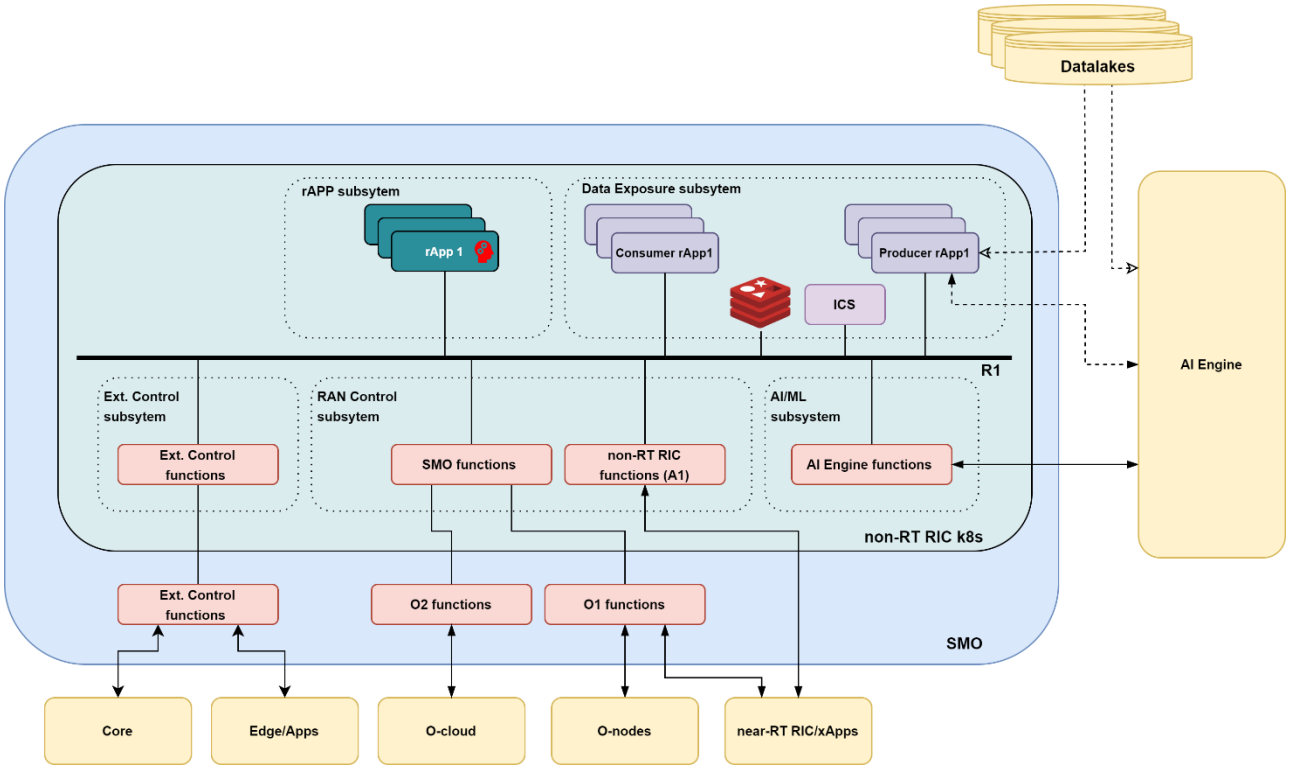


Figure 2-32 SMO, non-RT RIC, architecture

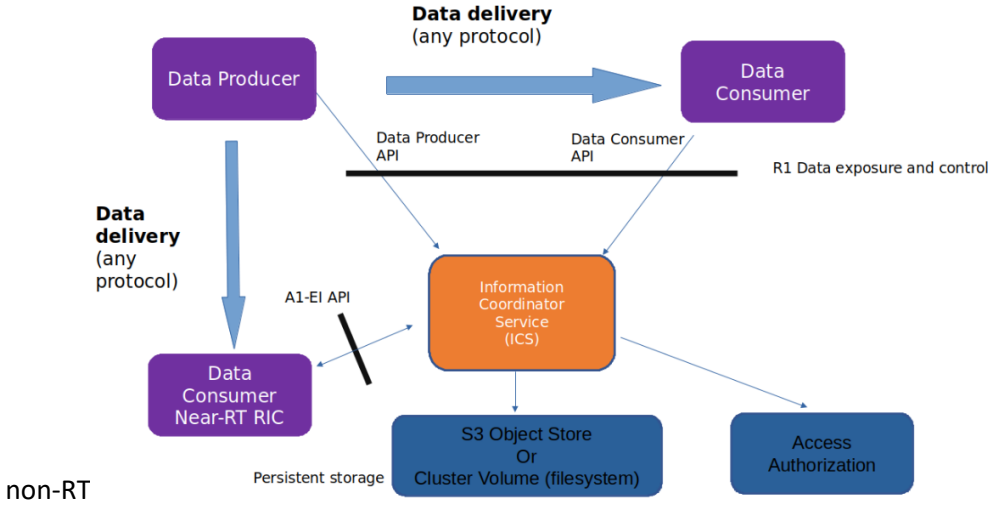


Figure 2-33 ICS architecture [37]

Energy efficiency related data, as is the case of the BeGREEN Energy Score or the Energy Rating, might also be served through specific producers interfacing the AI Engine and/or the BeGREEN datelake. In next deliverables, we will address its definition as a particular information type being exposed in the ICS and made accessible to rApps and xApps.

- RAN Control subsystem: Exposes functions to manage and monitor the O-nodes and the O-Cloud through O1 and O2 interfaces, respectively. In the case of O1, it might expose procedures related to Operation and Maintenance (OAM) or RAN Network Slice Subnet Management Function (NSSMF), which allow rApps to dynamically manage the configuration of the RAN nodes or the deployed slices. Also, in recent O-RAN specifications, energy consumption monitoring and energy saving configuration through O1 is being considered [24].

  Regarding O2, according to O-RAN architecture [38], it might expose to rApps Infrastructure Management Services (IMS) through the Federated O-Cloud Orchestration and Management

(FOCOM) component and Deployment Management Services (DMS) through the Network Function Orchestrator (NFO). While IMS manages and monitors the O-cloud resources, DMS controls and monitors the NF deployments into the O-Cloud. As described in [24], both interfaces are relevant for optimizing the energy efficiency of the O-Cloud; e.g., O-cloud resources and NFs can be adapted to RAN dynamics such as traffic load.

The management of xApps through the Near-RT RIC, mainly their associated policies through the A1-P interface, is another significant service be that might be exposed to rApps. This interface allows rApps to modify the behaviour of xApps doing near-RT optimisations (e.g., Traffic Steering or Load Balancing), according to non-RT objectives. Although a specific xApp or policy for managing E2-nodes Energy Savings has not yet been defined, [24] and [39] already define Energy Saving Use Cases involving the A1 interface. The A1 interface can also be used by the Near-RT RIC to empower the optimisations of the xApps with data external to the Near-RT RIC domain. This is done through the A1-EI interface [5], which, as aforementioned, allows xApps to create data subscriptions or jobs managed by the non-RT RIC (the ICS in our architecture). In BeGREEN, this could be used to expose energy efficiency related like the Energy Score. Similarly, the A1-ML interface, which is still under definition, is to support AI/ML workflows between the non-RT RIC and the Near-RT RIC, for instance in scenarios involving Federated Learning [5]. In BeGREEN, we will make use of this interface to implement policy management from rApps and xApps focused on energy efficiency. Additionally, we will explore its utilisation to communicate "AI Engine Assist" rApps and xApps.

- External Control subsystem: In addition to the RAN, BeGREEN also targets the enhancement of energy efficiency of NFV user-plane functions and of mobile edge services. Therefore, as shown in Figure 2-33, the non-RT RIC/SMO architecture considers interfaces to the 5GC and the Edge components to provide to the rApps relevant data and some control level. In its specification of the non-RT RIC [3], O-RAN considers external terminations to enable communication with external entities outside the scope of the RAN. In addition, in the recent version of the report on Use Cases [40], it is studied how to share non-RT RIC data, in particular analytics generated by the rApps, with the 5GC NFs. In conclusion, the definition and implementation of this type of interfaces is aligned with O-RAN interests.
- AI/ML subsystem: Although the concrete architecture to support AI/ML workflows by O-RAN is still ongoing work, [3] specifies AI/ML workflow services that may be supported by the non-RT RIC to allow model training, inference, performance monitoring and management. In BeGREEN, some of these services will be provided or supported by the AI Engine, whose functionalities will be exposed to the rApps through the AI/ML subsystem of the non-RT RIC. This could be done through specific rApps doing AI/ML assistance, as introduced in the AI Engine description in section 2.2.1.1.

2.2.1.3 Near-RT RIC

The near RT RIC, and xApp, plays a pivotal role within the O-RAN architecture by enabling applications to operate with minimal delay. These xApps are designed to enhance the capabilities of the RAN through swift decision-making processes in the sub-second scale. For example, Dynamic Spectrum Sharing (DSS) is one such near real-time application that swiftly allocates radio spectrum resources, be it for 5G-NR, based on immediate demand. Additionally, interference management applications rapidly identify and mitigate sources of interference, ensuring optimal network performance and improving energy efficiency through an intelligent decision making. In the realm of RU management, near real-time applications offer significant benefits. They facilitate dynamic adjustments of RU configurations in response to rapidly changing network conditions or user demands. This allows for quick allocation of different frequency bands to RUs, promoting load balancing, capacity optimisation, coverage enhancement, and power control. By providing this rapid decision-making capability, the near real-time RIC plays a critical role in elevating the efficiency and

responsiveness of the RAN within the O-RAN framework. A detailed definition of the Near-RT RIC can be found on the section 3.3 of the BeGREEN D3.1 [41].

One of the integrations foreseen for the Near-RT RIC within BeGREEN, is the creation of a direct interface with the AI Engine. Similar to the non-RT RIC, the Near-RT RIC will host an AI Engine functions (i.e., the "AI Engine Assist xApps" introduced in Section 2.2.1.1) to support AI SDK functionalities in the Near-RT RIC connected through the desired interface. This interface, as not fully standardized, poses several challenges associated with the type of functions supported by the AI Engine and the type of data exchanged between.

Additionally, a Telemetry Gateway (TGW) is implemented as part of the Telemetry Framework described in the BeGREEN D2.1 [1]. This xApp is intended to translate Traffic related metrics from DU/RU devices into Energy related metrics that are non-existent or with un-standardised formats. The TGW must collect input values from the Kafka bus and publish the calculated output metrics with the same timestamp, so other xApps use them independently of its source (real radios or emulated/translated information).

The Near-RT RIC components are based on the dRAX presented in Figure 2-14 and closely described in section 2.1.2.4.2 where O1 and E2 interfaces are deployed to control DU/RU nodes based on energy savings decisions made on the energy management xApps. Also, A1 interface is going to be updated to connect with the non-RT RIC described on the previous subsection.

The Near-RT RIC – dRAX implementation will host the xApps implemented in this project, as shown in Figure 2-34, to provide energy savings and integration with the AI engine. In the figure, the general Near-RT- RIC dRAX implementation presents the internal architecture to support the roles of the Near-RT RIC. Initially, the RIC main services are included as follow:

- API Gateway: Provides interconnection with northbound components like non-RTnon-RT RIC, SMO and other systems.
- Service Orchestrator: Enables deployment of xApps and smooth integration into the dRAX. Supported by the O1 interface, provides cooperation with the SMO.
- Service Monitor: Enables monitoring of Kubernetes and RIC Services and provides connection for service discovery by the Service Orchestrator.

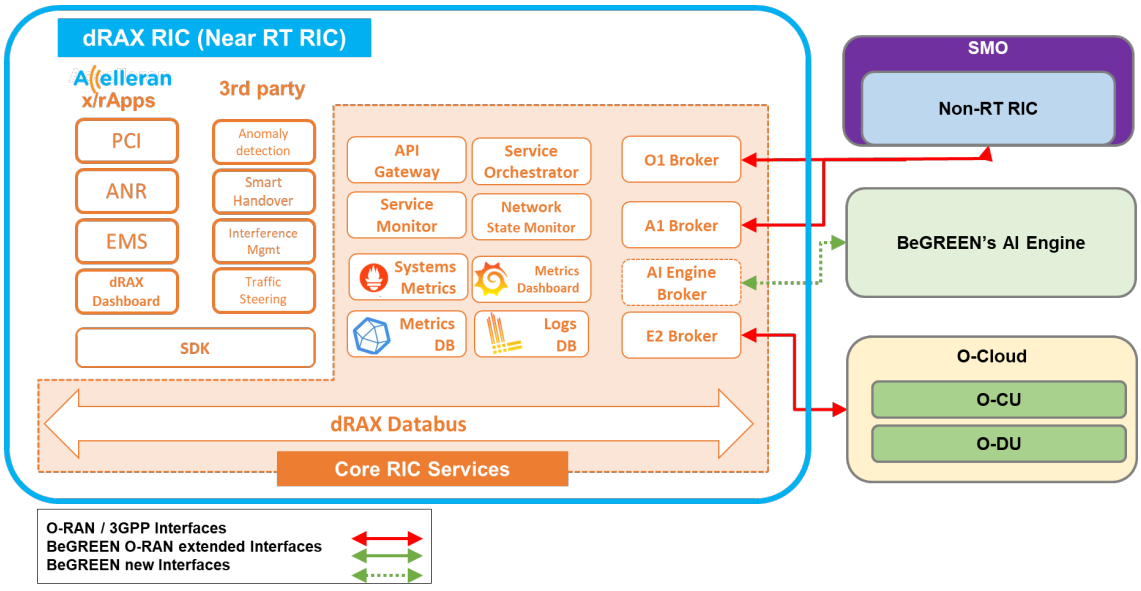


Figure 2-34 dRAX RIC internal architecture

- Metrics: The metrics consist of the System metrics -monitoring the health of the system-, the Dashboard -presenting the information of the system performance-, the Database -collecting the system metrics- and the Logs -logging all the system information-.
- E2 Broker: Provided E2 interface support to provide interconnection with southbound RAN components.
- A1 Broker: Provides abstraction of the A1 protocol while being the Endpoint for A1 protocol northbound to the non-RT RIC.
- O1 Broker: Provides abstraction of the O1 protocol while being the Endpoint for O1 protocol northbound to the SMO.
- dRAX Data bus: Provides connectivity among all the components inside the RIC either by Kafka or NATS depending on the type of messages transported.

Additional to the main RIC services, the Near-RT RIC provides the following components:

- xApp SDK: Implements the xApp API framework, containing predefined endpoints to interface with the dRAX RIC. It is also extendable and fully customizable enabling deployment-time configuration of xApps and exposes the xApp API for Discovery.
- xApp Catalog: Provides the catalog of the deployed xApps either proprietary or from third parties. In collaboration with the xApp SDK provides easy xApp implementation with in the dRAX.

Finally, to support the integration with the AI Engine, an AI Broker will be proposed for the BeGREEN project to support the interconnection with the AI Engine. Functions similar to the ones hosted by the xApp SDK will be designed to provide seamless integration to the AI Engine inside the xApps.

2.2.2 RAN control and monitoring functions

This section provides a comprehensive description of the architecture of the O-Cloud CU/DU to achieve energy efficiency and a description of the RU, RIS and Relay control functions.

2.2.2.1 O-Cloud

The O-RAN Alliance defines O-Cloud as a cloud computing platform, encompassing a set of physical infrastructure nodes meeting O-RAN requirements. This platform is designed to host relevant O-RAN functions, accompanying software components, and the necessary management and orchestration functions. The term "O-Cloud" extends beyond a singular entity and refers to a collection of resource pools situated at one or more locations, along with the associated software managing nodes and their deployments. Within the O-Cloud framework, there exists functionality supporting both deployment-plane and management services. Regarding the separation of hardware and software, or the processes of disaggregation and cloudification, the underlying concept is that the software finds residence on COTS servers. These servers can be located on the cell site, within an Edge Cloud, or in any data centre, introducing a flexible deployment model that transcends geographical constraints [42].

Within this O-Cloud framework, it plays a crucial role in overseeing physical assets, such as servers and networks, hosting essential O-RAN functions, and facilitating communication and management through interfaces like O2. It presents this one to the SMO, facilitating secure communication and enabling the SMO to oversee both the infrastructure and the life cycle of O-RAN network functions. The O2 interface is designed to be infrastructure-agnostic, ensuring compatibility with various cloud platforms, and promoting a multi-vendor environment for wireless network operators.

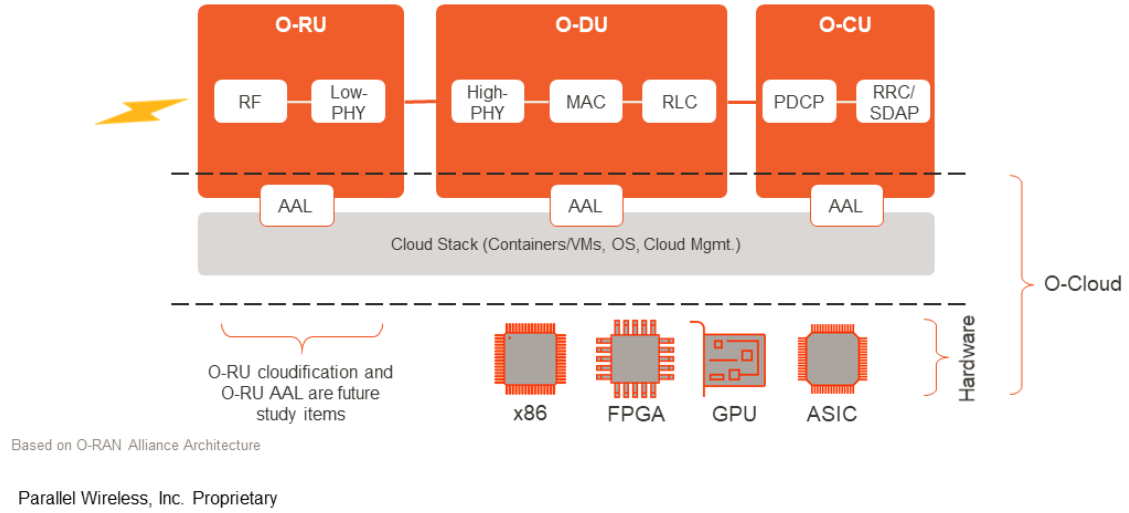


Figure 2-35 Typical components of an O-Cloud architecture in O-RAN [44]

Some of the O-Cloud's services and functions include resource discovery and administration, provisioning of network functions, handling FCAPS aspects of network functions, and managing software life cycles. These services can be categorized into two main classes: Infrastructure management and Network function deployment. The SMO is responsible for managing infrastructure and the lifecycle of Network Functions on timescales that extend to minutes or longer, and it does not support rapid radio policing. To enable real-time communication between NFs and the O-Cloud, O-RAN utilizes an Acceleration Abstraction Layer (AAL). This AAL abstracts the computing resources within the O-Cloud as Logical Processing Units (LPUs). For each NF, an LPU corresponds to a processing unit, subsystem, or hard partition in a High Availability (HA) environment. NFs make use of LPU Queues to access shared O-Cloud resources, but they only have visibility into the status of their local queues. An LPU may then be associated with one or multiple AAL Profiles, which define the functions that can be offloaded to an HA.

Additionally, the O-Cloud compromises the RAN components that can be controlled and are used for monitoring the status of the network. As shown in Figure 2-35 , the O-Cloud is comprised of the AAL and HW abstractions of the O-CU, O-DU, and O-RU. Hence, each O-Cloud Node is a collection of CPUs, memory, storage, NICs (Network Interface Cards), etc., and may include hardware accelerators to offload computational-intense functions with the aim of optimizing the performance of the O-RAN Cloudified Network Functions. This is where the AAL Interface plays a role as it allows different software vendors' network functions to work with different hardware and accelerators. The actual status, at the time of writing this text, the extend of the O-Cloud goes down to the O-DU. It is expected that in the future, even the O-RU will be cloudified and running on COTS servers.

The O-Cloud implementation can have several flavours, depending on the location of the Cloud machines. In Figure 2-36, there is an example comparing two scenarios depending on the location of the regional cloud and the edge cloud. In scenario B, the processing of the BS, involving O-CU and O-DU, is implemented at the network edge to minimize latency. Meanwhile, the Near-RT RIC is centrally located in the regional cloud data center, providing a comprehensive network perspective. This configuration aligns with the initial use case endorsed by O-RAN. In scenario C, the O-DU remains at the edge, ensuring low latency, while the O-CU is transitioned to the regional data center to optimize costs and reduce energy consumption at the edge location. O-CU and Near-RT RIC operate on a similar time scale, facilitating their co-location. Notably, stringent latency requirements are attributed to O-RAN Front-Haul (O-FH), whereas E2/F1 interfaces allow for a more relaxed approach.

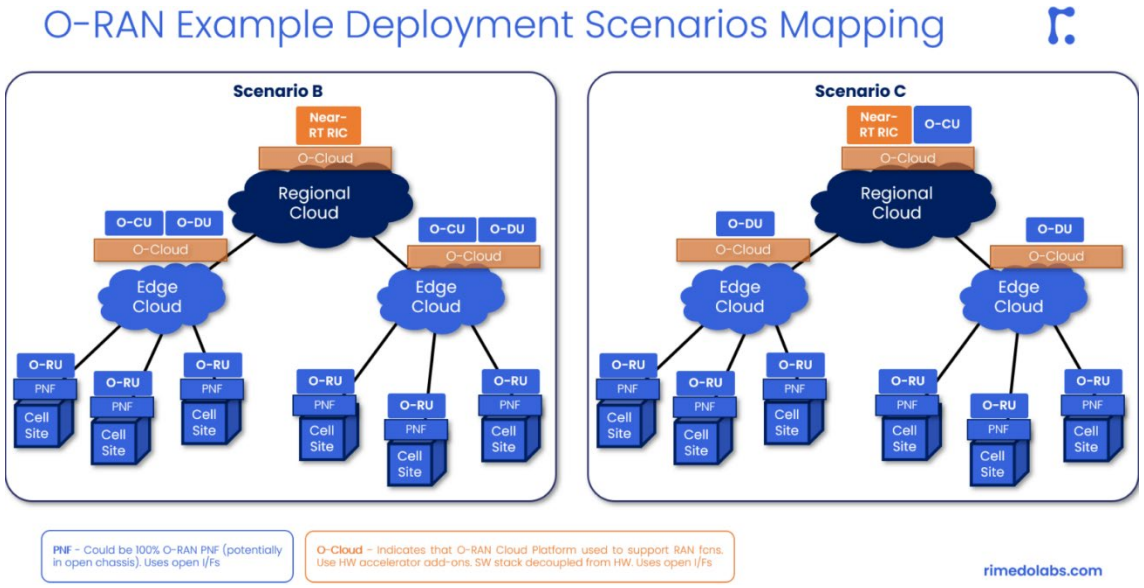


Figure 2-36 O-RAN scenario implementation including O-Cloud [45]

In O-RAN, the O-Cloud that conform the PHY layer is split between the O-CU and the O-DU. The O-DU contains the higher physical layer High-PHY functions while the O-RU contains the lower physical layer Low-PHY functions. The following subsections describe the O-CU and O-DU components.

2.2.2.1.1 O-Cloud central unit (O-CU)

The O-CU platform must be a fully standards-compliant implementation of the 5G gNB CU-CP and CU-UP components supporting interoperable interfaces to 5GC Network and other RAN components. It is designed and architected as a fully cloud-native, carrier-grade quality implementation, highly portable between software environments. The distributed software architecture delivers a scalable and resilient solution with in-built support for network security. The CU architecture is composed of two major components, the O-CU-CP control plane, and the O-CU-UP user plane. These conform to 3GPP functional requirements in both cases. The software implementation is deployed as a set of container-based microservices.

The static O-CU architecture, shown in Figure 2-37, consists of two subcomponents O-CU-CP and O-CU-UP. Each subcomponent represents the functionality implemented as specified by the set of functions.

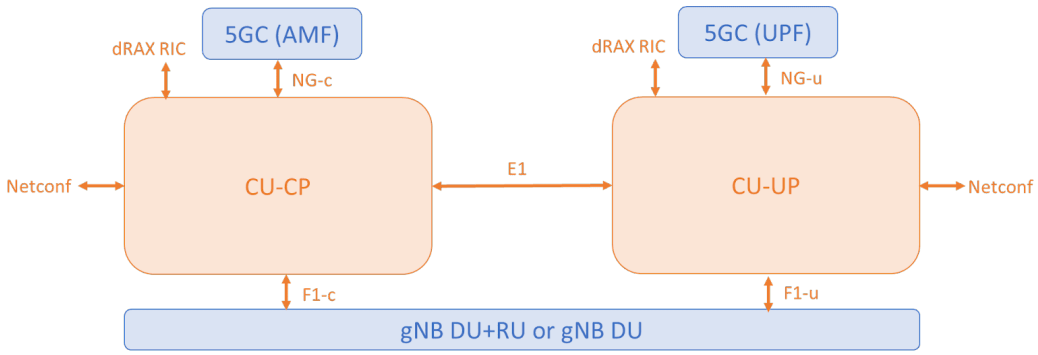


Figure 2-37 Accelleran CU static architecture

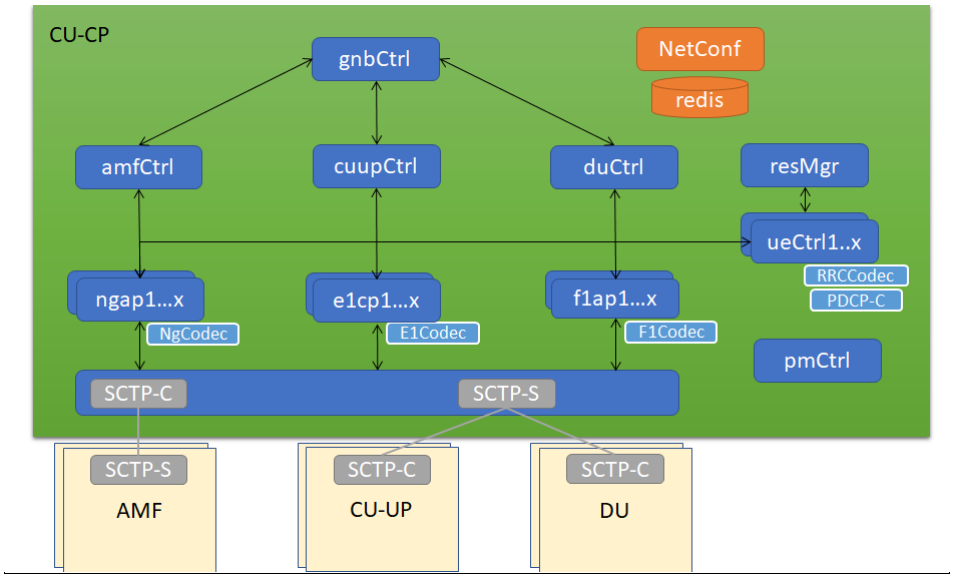


Figure 2-38 CU-UP architecture

A. O-CU – CP architecture

Figure 2-38 shows the outline of the CU-CP architecture. The O-CU-CP is connecting to the Access and Mobility Management Function (AMF), the O-CU-UP and the DU through SCTP. The following subcomponents are part of the O-CU controlling the O-DU:

- *gnbCtrl*: Used for distributing the Admin State (locked and unlocked) over the existing subcomponent. It controls Redis data base for the DU, AMF, UP and Session Information.
- *AmfCtrl*: Responsible for handling the procedures which require interactions with one or more AMFs from the Core side.
- *CuUpCtrl*: Responsible for controlling all User plane entities which are added during run time.
- *DuCtrl*: Controller for all O-DU entities which O-CU-CP is interacting with.
- *resMgr*: Managing all the resources for the O-CU-CP.
- *ueCtrl1..x*: Managing all RRC (Radio Resource Control) UE procedures (e.g., RRC setup for each UE).

B. O-CU – UP Architecture

Figure 2-39 outlines the basic CU-UP architecture. The O-CU-UP is connecting to O-CU-CP using SCTP Protocol, and supports the *E1up* subcomponent which is the Terminator service and uses E1 codec for any messages between O-CU-UP and O-CU-CP. In addition, the following subcomponents are part of the O-CU-UP:

- *resMgr*: Responsible of managing all the resources for the O-CU-UP, the admission control and of verifying the supported Public Land Mobile Networks (PLMNs) in the Packet Data Unit (PDU) sessions.
- *CtrlUp*: Responsible for handling the Bearer Messages and setting up all layer in thePacket Data Convergence Protocol (PDCP), Service Data Application Protocol (SDAP), and *nrUpp*.

Finally, a O-CU implementation can support a vast list of control services and features such as Architecture and Scaling, Quality of Service Management, Radio Resource Management, Security, Mobility, Handover, Voice Service, Emergency Services, Public Warning System, NG-Flex, RAN Sharing, Protocol procedures, Operations and Maintenance, and Engineering Logging and Tracing among others.

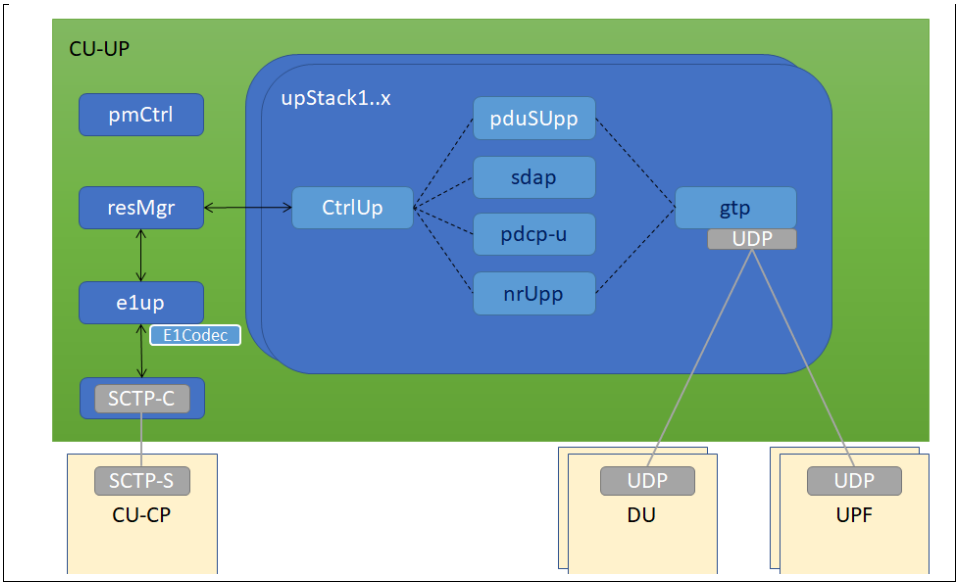


Figure 2-39 CU-UP architecture

C. O-CU energy efficiency

Since the O-CU and its subcomponents are virtualized based on the micro-services architecture using Kubernetes, the control of resources needs to be supported by the Kubernetes resource manager. In particular, each Kubernetes pod can be created to support one or more of the subcomponents, either of the O-CU-CP or the O-CU-UP. The number of pods can be controlled and scalable dynamically based on the O-CU needs, the infrastructure of the network (i.e., the number of DUs and RUs) and the traffic present on the network. Kubernetes offers powerful scaling capabilities that enable dynamic adjustment of resources based on application demand. The O-CU implemented in Kubernetes, has the platform's auto-scaling features providing the flexibility to automatically add or remove pods, optimizing resource utilisation and ensuring efficient energy consumption. This not only enhances performance during peak times but also contributes to energy savings by scaling down during periods of low demand. By setting resource quotas and limits, Kubernetes provides fine-grained control over individual pod resource consumption, creating a flexible and eco-friendly environment for applications in cloud or data centre environments. In case of the O-CU-UP, where the data traffic is handled, the *CtrlUP* can be controlled to include energy efficient algorithms such as the eXpress Data Path (XDP) implementation proposed in BeGREEN. In theory, each HW acceleration implementation can be controlled via E2 interface by the Intelligent Plane and the xApps supporting them.

2.2.2.1.2 O-Cloud Distributed Unit (O-DU)

As described in Figure 2-40, O-DU implements the functional blocks of L1 and L2 layer of a 5G NR protocol stack in Stand-alone mode which interface through the Function Application Platform Interface (FAPI). These layers primarily include NR MAC and NR RLC layers for the L2 Functional Block and the High -PHY for the L1 Functional Block. The Radio Link Control (RLC) Layer Functions at L2 involve several key processes, including the transfer of upper layer PDU, error correction, segmentation and reassembly, and re-segmentation and re-ordering. Moving on to MAC Layer Functions at L2, this encompasses multiplexing/de-multiplexing, scheduling information reporting, handling UEs priority, error correction through Hybrid Automatic Repeat reQuest (HARQ), and logical channel prioritisation. Additionally, the High-PHY Functions at L1 involve essential tasks such as channel coding, modulation, and Forward Error Correction (FEC). These layers collectively contribute to the efficient functioning and communication within a wireless network [46].

O-DU L2 functional blocks consist of the O-DU-OAM-Agent, E2 handler, F1AP Handler, F1 User Plane Interface Handling Modules, RLC Protocol modules, MAC Protocol Modules, and L2 MAC Scheduler. The O-DU-OAM-

Agent module manages startup, registration, and configuration with the SMO, handling software management and performance data for O-DU and O-RU. The E2 handler module terminates the E2 interface, managing E2AP protocol messages, KPI measurements, and responses. The F1AP Handler oversees tasks related to cell and UE management. Additionally, the F1 User Plane Interface Handling Modules and RLC Protocol modules handle tunnel management, data processing, and logical channel data transfer, while MAC Protocol Modules include functions like RACH and HARQ management. The L2 MAC Scheduler orchestrates MAC scheduling components within the O-DU architecture. Figure 2-41 shows the interaction of these functional blocks [47].

Figure 2-42 depicts the L1 functional blocks for a typical 7.2 split supported by O-RAN. It encompasses the interfaces handlers (FAPI and Front-haul), the DL and UL reference signals as well as the physical channels (PUSCH -Physical Uplink Shared Channel-, PDSCH -Physical Downlink Shared Channel-, PUCCH -Physical Uplink Control Channel-, PDCCH -Physical Downlink Control Channel-). More details of the implementation of the O-DU are presented in the BeGREEN D3.1 document [41].

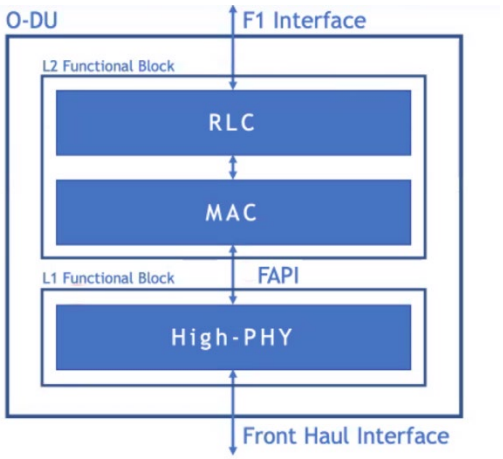


Figure 2-40 O-DU internal architecture

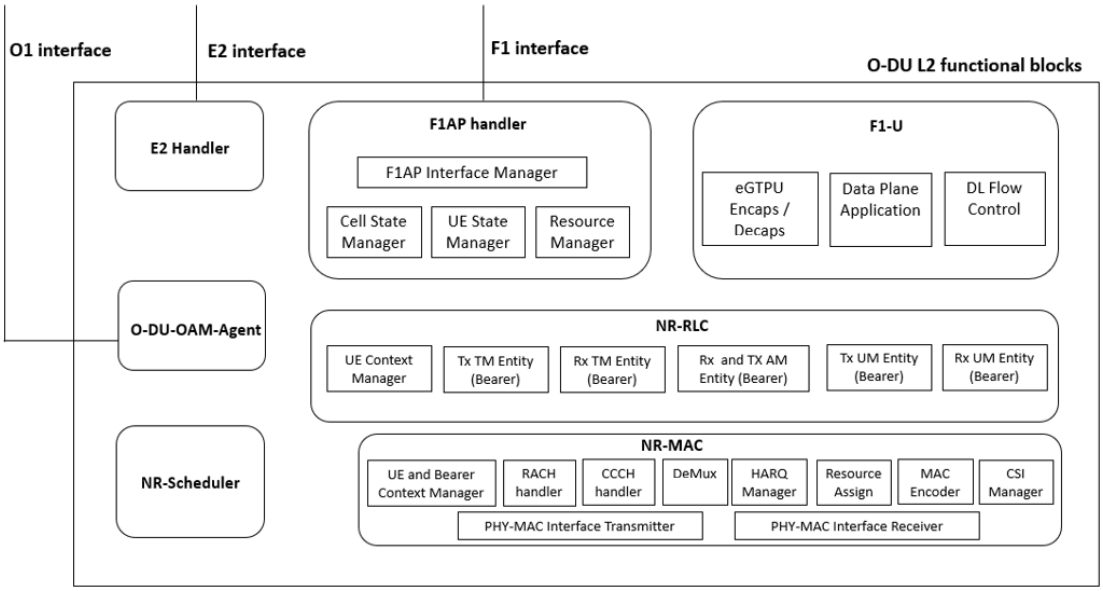


Figure 2-41 O-DU L2 functional blocks

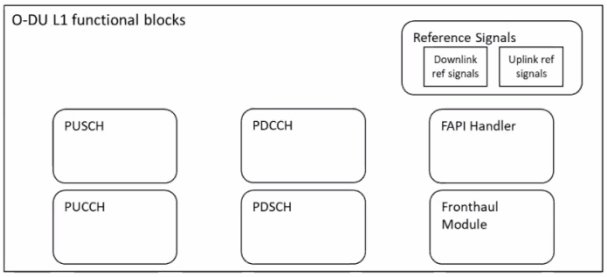


Figure 2-42 O-DU L1 functional blocks

2.2.2.2 Radio unit (RU)

RU control through standard interfaces is being considered by O-RAN [42]. M-Plane architecture and models are depicted in Figure 2-43. A hierarchical model will be used for the RU to enable management via the front-haul link connecting the DU with the RU. NETCONF application and YANG models will be applied for the RU element management. NETCONF server will reside in the RU. The utilisation of this interfaces towards accomplishing RU energy savings is being addressed by different O-RAN Working Groups. More details can be found in the BeGREEN deliverable D3.1 [78]. Also, in BeGREEN D3.1 several RU power consumption optimisation techniques have been described, including AI based Digital Pre-Distortion (DPD) and Envelope Tracking (ET). Those techniques are independent to the RU and operate autonomously and do not require any interfaces to external modules.

Within the scope of BeGREEN WP4, the integration of the RU with the Intelligent Plane will consider the following three approaches:

- **Using RunEL RU:** Currently**,** RunEL proprietary toolset is used for RU control and set up**,** while the monitoring module and task is currently under development. Integration with the Intelligence Plane will be needed for monitoring of the RU real time power consumption. The monitoring system of the RU has two counters which are restarted every reading cycle:
  - Accumulated actual power consumption which is calculated every frame (10 ms) and added to the counter.
  - Actual power consumption peak which is the maximum power measured at one of the frames above.

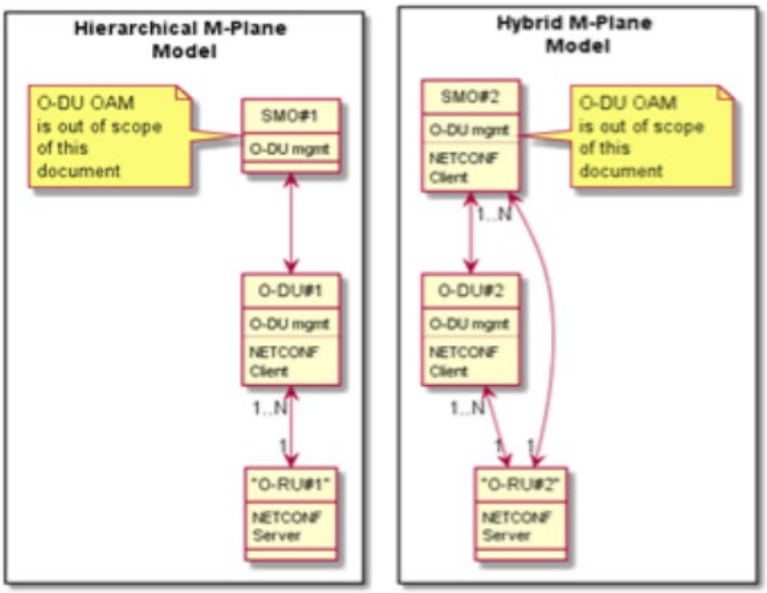


Figure 2-43 RU management plane architecture

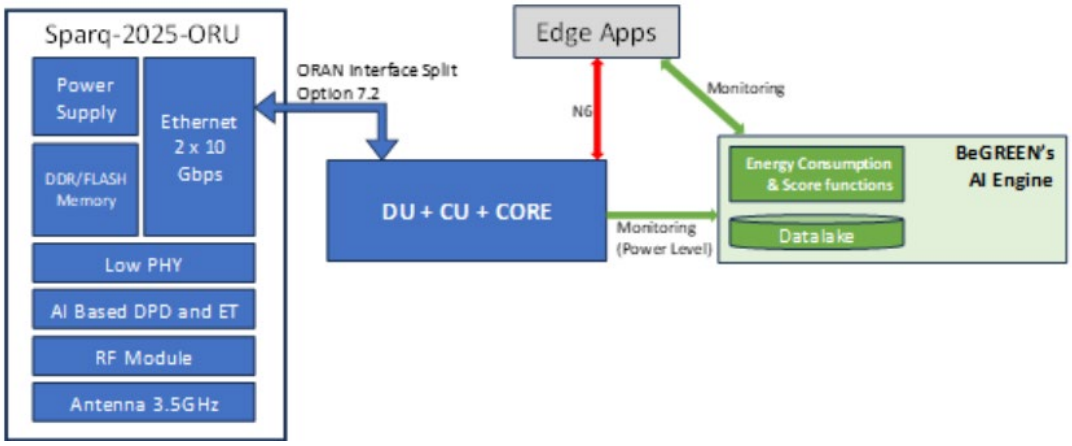


Figure 2-44 RU power consumption monitoring architecture

The access to the counters is done by the management system using Web Socket protocol. This could be exposed through the datalake component in the AI Engine and the Energy Consumption and Score functions. Common interval between two consecutive accessing is 15 minutes, but any number of seconds up to several hours is applicable. The foreseen high-level architecture of the RU power consumption monitoring architecture is outlined in Figure 2-44. Its integration will be considered within the scope of BeGREEN WP5.

- **Benetel RU:** The end-to-end disaggregated 5G O-RAN platform delivered by Accelleran is based on the use of 3rd party RU (Benetel). The RU is currently being configured via SSH, since neither the O-RAN O1 interface to the SMO (hybrid architecture) nor the O-RAN fronthaul management interface to the DU (hierarchical architecture) are supported by Benetel. Nevertheless, we will consider these interfaces in case they are developed by Benetel during BeGREEN. The suitable Benetel RUs that may be used for RU management demo within the scope of BeGREEN WP3 [78] and WP5 [48], focused on RU power management through xApps, are introduced in Table 2-2.
- **TeraVM:** TeraVM RIC Tester, as was introduced in Section 2.1.2.5.3, allows to emulate O-RAN compliant RAN, including the CU, DU and RU elements, and their associated interfaces. In BeGREEN Deliverable D5.1 we described the role of the TeraVM tool in the Intelligent Plane demo to be realized under the umbrella of the BeGREEN WP5 [48]. In particular, E2 and O1 interfaces will be exploited to monitor and control RUs targeting energy savings in the RAN through some of the AI/ML-empowered rApps/xApps proposed in Chapter 3 of this deliverable and in deliverable D3.1 [78].

Table 2-2 Benetel RUs Integrated with Accelleran dRAX

Scenario	Benetel RAN 550 (Indoor)	Benetel RAN 650 (Outdoor)
Frequency band	N78 / N79	N78 / N77U
Power (at antenna port)	250mW/24dBm	5W/37dBm
HW MIMO (SW Support)	4T4R (2T1R)	4T4R (2T1R)
Architecture	Option 7.2	Option 7.2
Bandwidth	40, 100MHz	40, 100MHz

2.2.2.3 RIS control functions

One of the still undefined tasks within the RIS ecosystem is to identify the most suitable network architectures and deployment strategies for exploiting RIS technology. The European project RISE-6G is the spearhead of the standardisation efforts of integrating RIS within the O-RAN architecture.

Tailored to various scenarios and specific application requirements, flexible RIS devices must be seamlessly

integrated into a network in which they can be adaptively (re-)configured and orchestrated based on real-time and may be able to predict network dynamics. In [49], the project RISE-6G preliminary identifies optimized network structures and deployment strategies that can significantly improve the targeted KPIs defined in [50].

To define a RIS-enabled O-RAN architecture, the RISE-6G project redefines the following functional elements which BeGREEN will take as a starting point:

- **Reconfigurable Intelligent Surface (RIS)** employs either reflect-array or meta-material technology and is directly managed by an associated RIS actuator. It operates with time granularity ranging from 100 microseconds to 10 milliseconds. In some cases, the RIS actuator can be integrated into the RIS device, resulting in a novel RIS device controlled directly by the RIS Controller (RISC) function. RIS devices have two operational modes. On one hand, Controlled RIS are controlled by an external entity through a control channel which may be Implicit (no dedicated control channel) or Explicit (where the control channel can be In-band or Out-of-Band). On the other hand, Autonomous RIS (also called Self-Configuring RIS) are operated by a RIS Controller without involving any external entity explicitly (although they may have any kind of implicit control channel).
- **RIS actuator (RISA)** is responsible for executing logical commands received from the RIS Controller, translating them into physical configurations applicable to the RIS device. These configurations may involve phase shifts or customized meta-material state changes. Additionally, the RIS actuator can offer limited feedback or sensing input for various RIS devices. The RISA operates under the guidance of the RIS Controller, with an action time granularity ranging from 1 to 20 milliseconds.
- **RIS controller (RISC)** serves as the controller for an RIS actuator or an RIS function. It generates logical commands for switching operations among RIS elements, such as predefined phase shifts. RISCs exhibit varying levels of complexity and capabilities, and they can incorporate third-party applications to implement intelligent algorithms. An RISC can either receive directives from other network elements, functioning as an interface configuring RIS elements based on external instructions (Controlled RIS), or operate autonomously (Autonomous RIS). The expected action time granularity for RISC operations ranges from 20 milliseconds to 100 milliseconds.
- **RIS orchestrator (RISO),** positioned at a higher hierarchical layer, manages multiple RISCs. Its action time granularity typically falls above 100 milliseconds.

Following these elements, RISE-6G proposes the architecture depicted in Figure 2-45, which includes the RISE-6G specific architecture (left) and its interactions with the O-RAN architecture (right).

Within the RISE-6G architecture, all RISA, RISC, and RISO functionalities can be virtualized, abstracted, and deployed into edge or central cloud infrastructures. Physical RF devices, however, will still require on-site placement. The RISAs can directly control physical devices like Metasurfaces/RIS at various operating frequencies through a universal interface known as the Open Environment. This interface can adapt to different RIS types, including nearly passive RIS, hybrid RIS, holographic RIS, and more. The RISC serves as the trigger for configuring the RISAs and receives specific feedback based on the type of RIS in use, facilitated through the Ra interface. RISO, on the other hand, can manage multiple RISC instances via the Rx interfaces. The RISE-6G architecture is designed to seamlessly integrate with the existing O-RAN network architecture using novel interfaces such as F1-x, R2, and Ro. In this architecture, two primary scenarios to exert control to the RIS can be envisioned:

i) Scenarios where the RIS device is directly connected to eNB/gNB, optimizing transmitter beamforming parameters and RIS configurations becomes possible. The CU C-plane would trigger specific RIS configurations to RISA within milliseconds through the F1-x interface. This is particularly valuable when the RIS deployment is under the network operator's control. In this scenario,

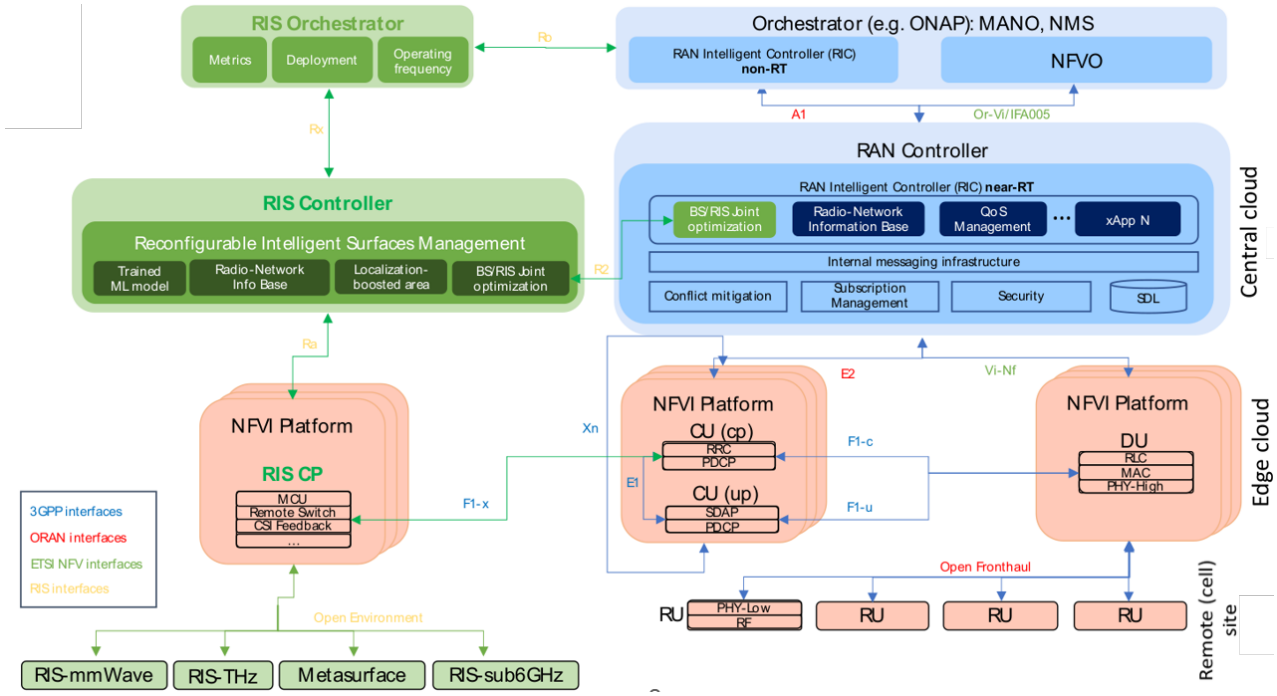


Figure 2-45 Relevant interfaces between RISE-6G architecture and O-RAN/3GPP/ETSI network architectures

xApps/rApps will use control channels already defined in O-RAN.

ii) Scenarios where the RIS device is connected to the management of eNB/gNB in a master/slave or peer-to-peer fashion, and RISA and the RIS device's configuration is exclusively managed by RISC (e.g., the F1-x interface is not available). In this case, RISC may establish direct communication with a dedicated xApp in the near-real-time RIC through the R2 interface, or with an rApp in the non-RT RIC through the Ro interface, via de RIS Orchestrator. This scenario accommodates self-contained and independent RIS deployments.

BeGREEN will consider these two scenarios to accommodate the EE-specific use cases and will study which of the control functions and interfaces proposed by RISE-6G suit better to implement them. For Self-Configuring RIS, BeGREEN will favour the use of a RIS Controller with an implicit control channel. When using externally controlled RIS, all Ro, R2 and F1-x interfaces will be considered.

2.2.2.4 Relay control functions

This subsection presents a general description of the different control functionalities for improving the network performance in terms of system capacity and coverage and reducing energy consumption by means of relays. Figure 2-46 shows the general architecture and highlights the main functions involved in this context. The relay control located inside the SMO is in charge of the coordination and management of all the processes related to the control of relays. On the one hand, the relay control manages the collection of network configuration parameters, network measurements and performance indicators that are reported by the CU/DU, RU and/or Relays. Collected measurements are sent to the datalake that stores a large amount of historical network measurements and performance indicators that are exploited by AI/ML-based functionalities to take adequate relay control decisions, such as the deployment of a new relay or the activation/deactivation of a relay. On the other hand, the relay control is sustained by different rApps located in the non-RT RIC. As shown in the Figure 2-46, these rApps oversee functionalities such as the detection of coverage holes and peaks of traffic, deployment of new relays, activation/deactivation of the different relays and network parameter reconfiguration. These rApps make use of the collected data available in different databases and different AI/ML models available in the BeGREEN AI Engine. The obtained result is sent to the relay control which sends the corresponding reconfiguration commands to the CU/DU, RU and/or Relays.

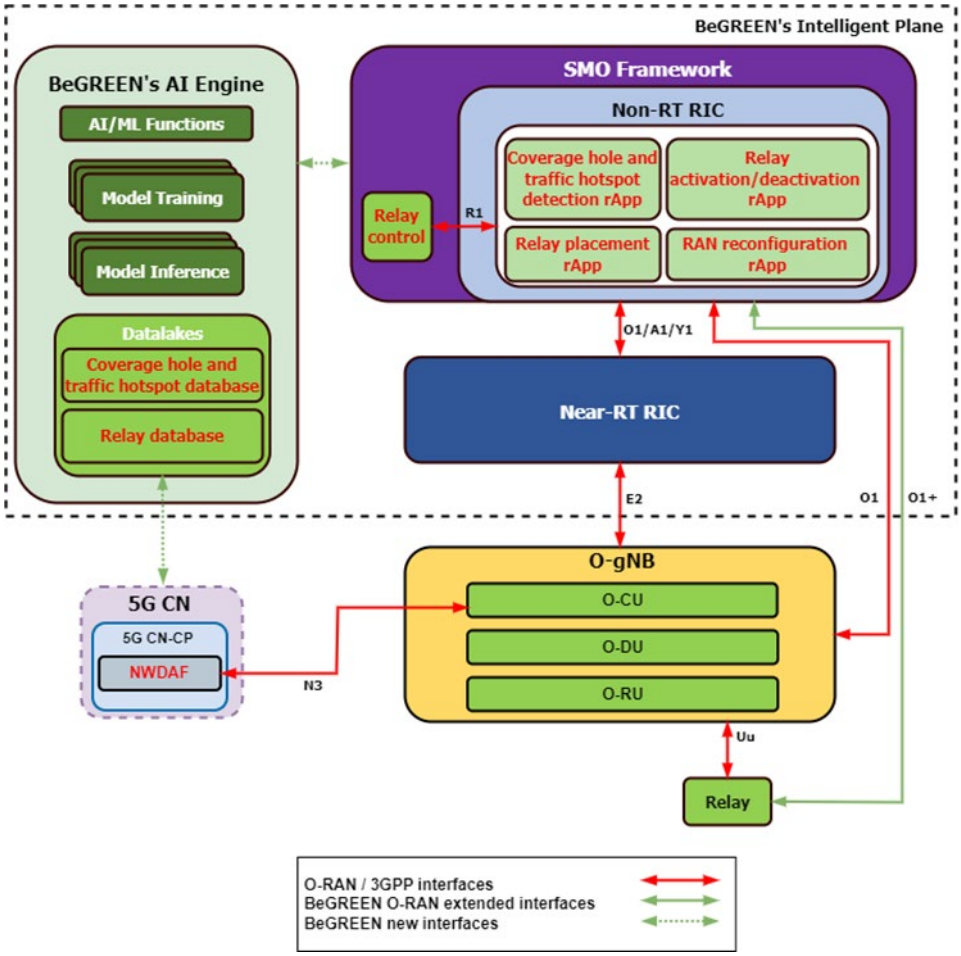


Figure 2-46 Main functions involved in the relay control

A description of the processes related to the relay control are explained below:

A. Collection of measurements

This process is related to the collection of network measurements and performance indicators that can be useful to identify regions with coverage problems, geographical space/time traffic distributions in the network, etc. An accurate coverage and space/time traffic characterisation is essential in order to take adequate decisions of relay placement, decisions of switching some fixed relays to sleep mode or activation/deactivation of RUE relaying functionalities. For this reason, the collected measurements that provide this space/time characterisation need to be associated with the timestamp and UE geo-location information where the measurement was taken.

3GPP standards provide different ways to determine the geographic position and/or velocity of the UE based on measuring radio signals [51]. These positioning methods include network-assisted Global Navigation Satellite System (GNSS), Time Difference of Arrival (TDoA) based on 4G Long Term Evolution (LTE) or 5G NR signals, Wireless Local Area Network (WLAN) positioning, Bluetooth positioning, etc. Location measurements collected from UEs in *RRC_connected* state can be transmitted periodically to the network as part of the radio measurement reporting processes [52]. In turn, UEs in *RRC_idle* or *RRC_inactive* mode can log measurements and transmit them later when the UE enters in *RRC_connected* state, e.g. using the Minimisation of Drive Tests (MDT) feature [53]. According to this, the collected measurements that can be useful for relay control are:

- RSRP (Reference Signal Received Power): It is a measurement of the UE received power of the reference signal coming from a Base Station [54]. This parameter is related to the signal strength received by the UE and is useful for characterizing coverage issues.

- RSRQ (Reference Signal Received Quality): It is a measurement of the UE received signal quality and it is calculated as the RSRP divided by the total UE received power including interferences [54].
- SINR (Signal-to-noise and interference ratio): It is a measurement of the UE received signal quality and it is calculated as the RSRP divided by the total interference and noise power received at the UE [54]. According to the Shannon equation, the SINR may provide an estimation of the spectral efficiency observed by the UE.
- RLF (Radio Link Failure) and HOF (Handover Failure): The Radio Link Failure report contains information related to the latest connection failure experienced by the UE with the geographical location of this failure [53]. It is sent from the UE to the eNB at the subsequent RRC connection restablishment.
- Statistics of UE throughput and data volume at cell level: Average values and distribution of uplink and downlink UE throughput at each gNB is available according to [55].
- CQI (Channel Quality Indicator) distribution: Measurements of the CQI reported by UEs in the cell is a useful metric reflecting signal quality and service quality [55]. Each CQI value can be associated to a spectral efficiency according to [56].
- Number of UEs in *RRC_connected* state to a relay node or a specific cell: Layer 2 measurements defined in 3GPP provide information about the average and maximum number of UEs in *RRC_connected* state over a specific period [57].

The process of measurement collection is activated by the relay control at the SMO with a given periodicity or based on specific events and aims to collect measurement reports and/or performance indicators at the CU/DU, RU, or Relays. These collected measurements (e.g. RSRP, UE geographical location, etc.) are reported to the SMO and stored in the datelake. The process of measurement collection is based in the following steps and is represented in Figure 2-47:

1. When this process is activated, the relay control function placed in the SMO sends a data collection request to E2 Nodes (CU/DU), RU or Relay via the O1 interface to collect network measurements and/or performance indicators.
2. The collected measurements reported by the CU/DU, RU or relays are sent to the relay control at the SMO.
3. The relay control stores these measurements in the datelake.

B. Detection of coverage holes and traffic hotspots

This process oversees the detection of geographical regions with poor coverage, i.e., coverage holes, and the detection of geographical regions with a high traffic demand (traffic hotspots). A coverage hole refers to a geographical area where the signal level of the serving cell is insufficient to maintain a basic service, such as the Signalling Radio Bearer (SRB) and Downlink Shared Channel (DSCH).

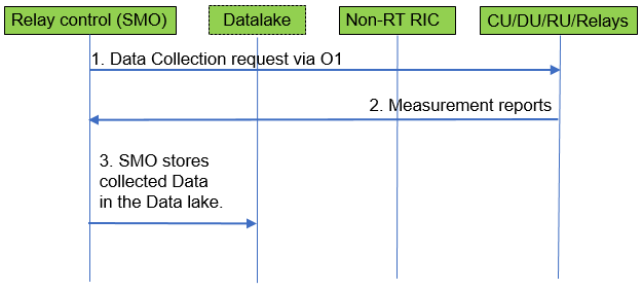


Figure 2-47 Measurement collection flowchart

Coverage holes can be caused by various factors, including physical obstructions like buildings, hills or tunnels, unsuitable antenna parameters, or inadequate RF planning. When a UE is located in a coverage hole, it may experience call drops and radio link failures, which can negatively impact the user experience. A robust methodology to detect the presence of coverage holes and take reconfiguration actions to avoid such negative situations is prime important in the management and optimisation of a cellular network and reduce energy consumption.

A traffic hotspot refers to a geographical region where the user traffic demands are relatively large in comparison to the available resources to satisfy the user requirements. A traffic hotspot is usually caused by a high density of users in a specific area willing to transmit a large amount of data. In these situations, it may happen that the network cannot guarantee the required QoS for the connected users. A robust methodology to detect the presence of traffic hotspots and take reconfiguration actions to solve these situations is relevant for the management and optimisation of a cellular network and reduce energy consumption.

According to the historical network measurements available in the datalake, this process is in charge of identifying geographical regions/clusters with poor coverage and geographical regions with high traffic demands. This is executed by the AI/ML coverage hole and traffic hotspots detection rApp and makes use of a clustering algorithm of geographical locations as it will be detailed in section 3.5.1. Different clustering methodologies can be used to identify these geographical regions (e.g. K-means, DBSCAN, etc.). The output of this process is stored in the coverage hole and traffic hotspot database and can be represented with a list of clusters, where each cluster represents a different geographical region with coverage problems or high traffic demands. Each cluster can be characterised with the list of geographical locations. This process is run according to the following steps and is represented in Figure 2-48:

1. With certain periodicity (or based on specific events) the relay control (located at SMO) activates the coverage hole/traffic hotspot detection process. Then, the relay control sends a command to the non-RT RIC through the R1 interface to activate the coverage hole/traffic hotspot detection rApp.
2. Historical measurements available in the datalake are sent through the R1 interface to the rApp at the non-RT RIC. Other relevant information collected by other entities may also be sent to the non-RT RIC to enrich the process of detection of coverage holes and peaks of traffic.
3. The coverage hole/traffic hotspot detection rApp makes use of the available measurements and runs a clustering process to identify and characterise the coverage hole and the traffic hotspot regions.
4. The list and characterisation of the identified coverage holes and traffic hotspots are stored in the datelake.

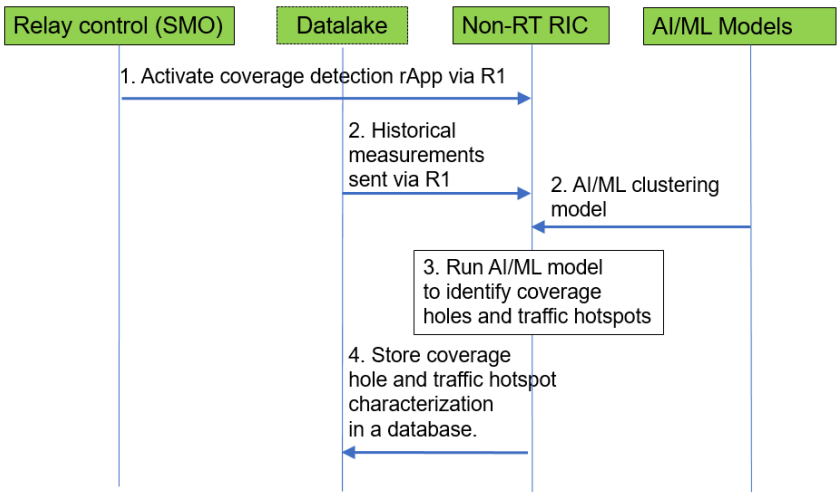


Figure 2-48 Detection of coverage holes and traffic hotpots flowchart

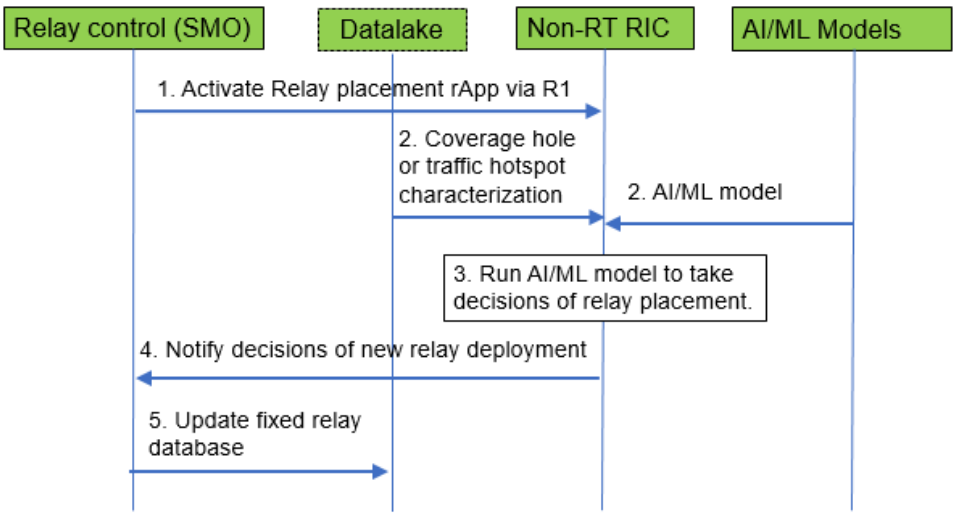


Figure 2-49 Relay placement process flowchart

C. Relay placement

This process oversees determining the necessity of deploying new fixed relays and setting their configuration parameters (geographical location, transmitted power, etc.) to guarantee the coverage requirements, improve the spectral efficiency, reduce total power consumption, etc. These relay placement decisions are assisted by an AI/ML relay placement rApp (located in the non-RT RIC with assistance from the AI Engine) that makes use of inputs such as network configuration parameters, performance indicators measurements, collected historical data (that may be stored in the datalake or in the coverage hole and traffic hotspots database). These inputs are processed according to AI/ML models available in the BeGREEN AI Engine (described in section 3.5.2) to obtain recommendations of the deployment of new relays. The output of this process is a list of new relay/s to be deployed together with their location/s and initial configuration parameters. This process is run according to the following steps and is represented in Figure 2-49:

1. With certain periodicity (or based on a specific event), the relay control (located at SMO) checks the necessity of the deployment of new fixed relays. When this process is activated, the relay control sends a command to the non-RT RIC via the R1 interface to execute the relay placement rApp.
2. The relay placement rApp (in the non-RT RIC) obtains the required information available at the coverage hole and traffic hotspots database.
3. The relay placement rApp makes use of AI/ML models available at the BeGREEN AI Engine and the available collected measurements to determine the necessity of the deployment of new relays.
4. The obtained results at the non-RT RIC are sent to the relay control function to notify the network operator about the necessity of the deployment of new relay(s).
5. In case that new relays are deployed, the configuration and status information of the new deployed relays is updated in the Relay database.

D. Relay activation/deactivation

This process is in charge of taking adequate decisions of relay activation/deactivation with the objective of improving the network performance and reduce energy consumption. For the case of fixed relays, energy savings can be obtained by switching some of them to sleep mode in certain time periods with low load levels at the coverage region of the fixed relay. For the case of relay UEs (RUEs), i.e. UEs with relaying capabilities, some of them can activate its relaying functionality in order to be able to act as a relay in specific time periods to extend coverage, improve spectral efficiency, etc. In other time periods, the relaying capability of the RUEs can also be deactivated in order to reduce energy consumption. These decisions can be assisted by a Relay activation/deactivation rApp (located at the non-RT RIC).

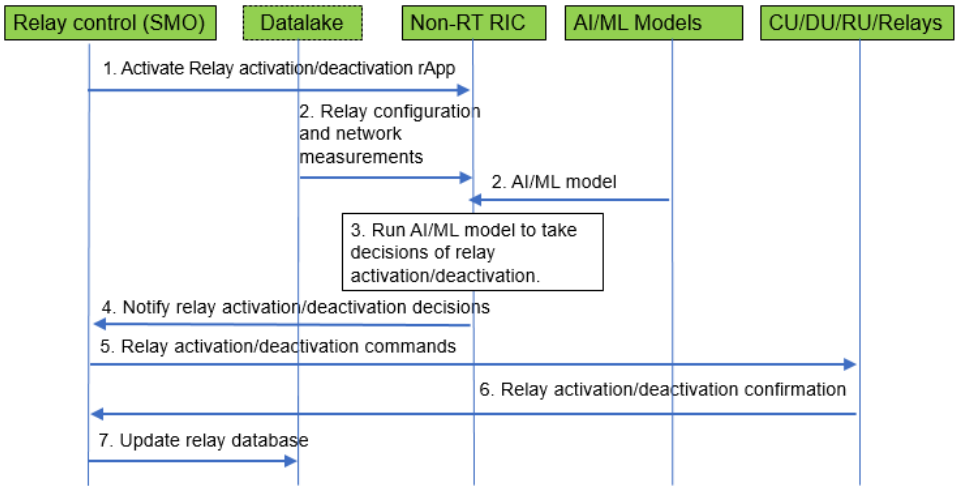


Figure 2-50 Relay activation/deactivation flowchart

The proposed relay activation/deactivation algorithm is described in section 3.5.3. The process of relay activation/deactivation is run according to the following and is represented in Figure 2-50:

1. With certain periodicity (or based on specific events) the relay control (located at SMO) takes the decision to execute this process. Then, the relay control function sends a command to the non-RT RIC through the R1 interface to run the relay activation/deactivation rApp.
2. The non-RT RIC retrieves from the Relay database information related to the status of each relay. The non-RT RIC also retrieves measurements or performance indicators stored in the datelake and/or the coverage hole and traffic hotspot database. It may also collect measurements/indicators from other sources.
3. The relay activation/deactivation rApp make use of ML models and AI/ML functions available at the BeGREEN AI Engine, the available collected measurements and the information related to the relays status to determine the necessity of activating/deactivating the different relays.
4. The non-RT RIC sends the activation/deactivation decisions to the relay control (SMO).
5. The Relay control sends the relay activation/deactivation commands to the CU/DU, RU and/or the Relays.
6. The CU/DU, RU and/or relays send the confirmation of activation/deactivation to the relay control.
7. The relay control sends a message to update the relay database.

E. RAN reconfiguration

This function oversees doing adequate RAN network parameters reconfigurations. After the activation/deactivation of a fixed relay or a RUE, some reconfigurations may be needed in the network (e.g. reduce transmitted power of a base station when a relay is activated for saving energy or increase it when the relay is deactivated for keeping coverage). These reconfigurations can be assisted by an AI/ML RAN reconfiguration rApp (located in the non-RT RIC). This is done according to the following steps and is represented in Figure 2-51:

1. After the activation/deactivation of a specific relay, the relay control entity sends a command to activate the RAN reconfiguration rApp at the non-RT RIC.
2. The non-RT RIC retrieves measurements or performance indicators stored in the datalake and information of the status of the different relays, available at the Relay database.
3. The AI/ML RAN reconfiguration rApp makes use of ML models and AI/ML functions available at the BeGREEN AI Engine, the available collected measurements and the information related to the relay

status to determine the most adequate RAN reconfiguration actions.

4. The non-RT RIC sends the RAN reconfiguration decisions to the relay control (at the SMO).
5. The Relay control sends the RAN reconfiguration commands to the CU/DU, RU and/or the Relays.
6. The CU/DU, RU and/or relays send the confirmation of RAN reconfiguration to the relay control.

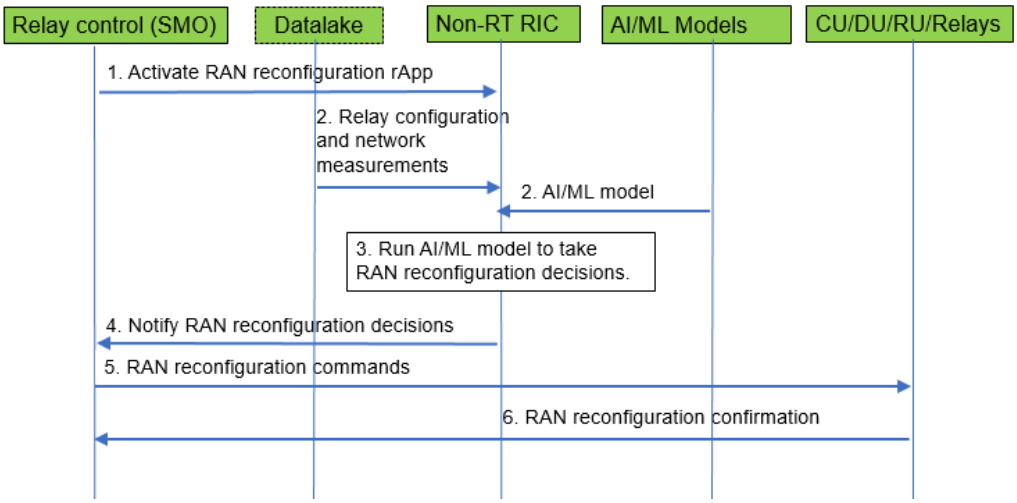


Figure 2-51 RAN Reconfiguration flowchart

2.2.3 Core Network functions

Regarding the integration of O-RAN architecture and the 5GC, O-RAN has started the analysis of different scenarios to provide RAN analytics to the 5GC NFs in the last version of the Uses Cases Analysis Report [40]. Basically, the benefits and disadvantages of different integration options is discussed in this section, mainly:

1. The SMO/non-RT RIC taking the role of a Network Data Analytics Function (NWDAF) which can be directly interfaced by 5GC NFs or by a Data Collection Coordination Function (DCCF) or Messaging Framework Adapter Function (MFAF) which forward analytics requests/replies from/to the 5G NFs.
2. The SMO/non-RT RIC taking the role of an OAM System or RAN NF interfacing the NWDAF of the 5GC.

Figure 2-52 and Figure 2-53 illustrate an example of both options and their impact on the SMO/non-RT RIC interfaces.

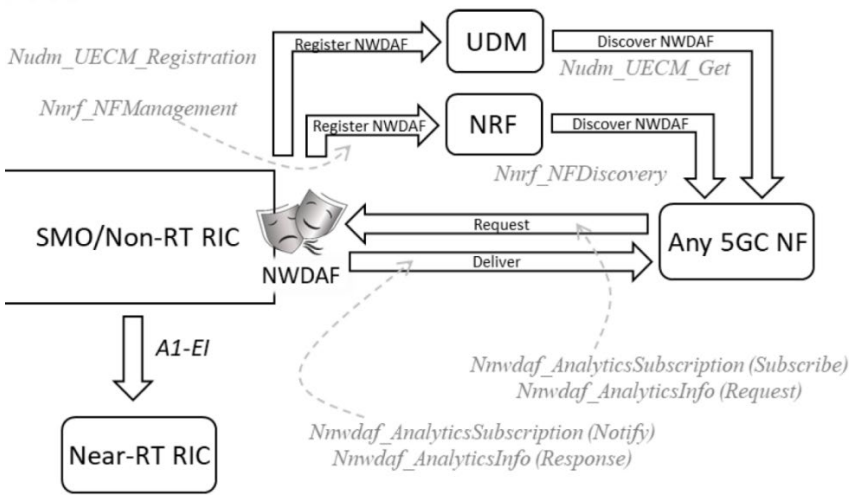


Figure 2-52 Exposure of RAN analytics to the 5GC [40], NWDAF façade

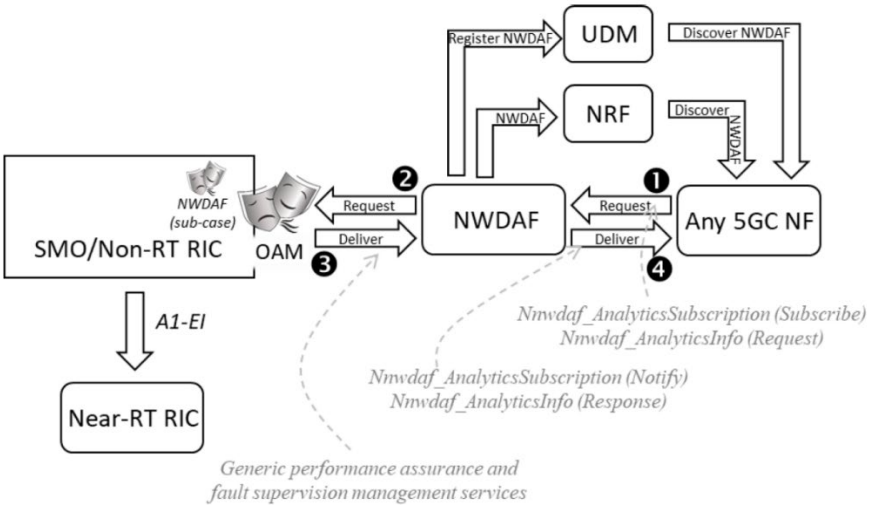


Figure 2-53 Exposure of RAN analytics to the 5GC [40], RAN NF façade

In BeGREEN we will also consider the integration of the O-RAN architecture and the 5GC through the development of management and monitoring interfaces and functions. In particular, we will develop a component in the SMO/non-RT RIC providing management of 5GC NFs and their associated resources. As will be detailed in Section 3.6, this will be exploited by an rApp targeting the optimisation of energy efficiency in the Edge server hosting the UPF NFs. Additionally, BeGREEN proposed optimisations that may require access to monitoring or analytics data from the 5GC, for instance by interfacing the NWDAF as is introduced in Section 3.5.3. These two envisioned interactions were illustrated in Figure 2-26, which depicted the BeGREEN baseline architecture and its relationship with the 5GC through additional/extended O-RAN interfaces such as O2+ and through the AI Engine datelake.

2.2.4 Joint orchestration of RAN and Edge functions

As introduced in Section 2.2, the BeGREEN O-RAN based intelligent plane architecture integrates the functionalities to orchestrate the RAN and edge services jointly. While edge services like Mobile Video Analytics (MVA) are becoming essential utilities for users, their widespread adoption necessitates a significant transformation in how we manage mobile networks. The network's role in these services extends beyond merely transmitting and processing data in transit. Instead, the network must directly enhance service performance by optimizing for accuracy (reliable inferences), end-to-end latency (swift inferences), and task throughput (inferences per second) in a resource-efficient manner. This last requirement is critical because these services generate substantial data flows, involve intensive computations, and consume significant energy.

Functions for optimising edge applications and RAN configuration policies are deployed as rApps in the O-RAN's non-RT RIC [3] to enforce radio control policies in O-RAN-compliant eNBs or gNB. The edge control rApps interact with O-RAN's A1 interface (specifically, the A1's Policy Management Service [5]) to enforce the corresponding radio policies. An xApp handles the A1 service from O-RAN's Near-RT RIC side and uses an E2 interface to forward radio policies to the Base Station. The E2 interface [8] is also used to gather BS KPIs, which are forwarded to the non-RT RIC through the O1 interface. Then, a second xApp manages data KPIs received from the vBS and sends them to the BeGREEN datalakedatelake. Figure 2-54 summarizes the architecture of this use case and the involved interfaces within the BeGREEN architecture.

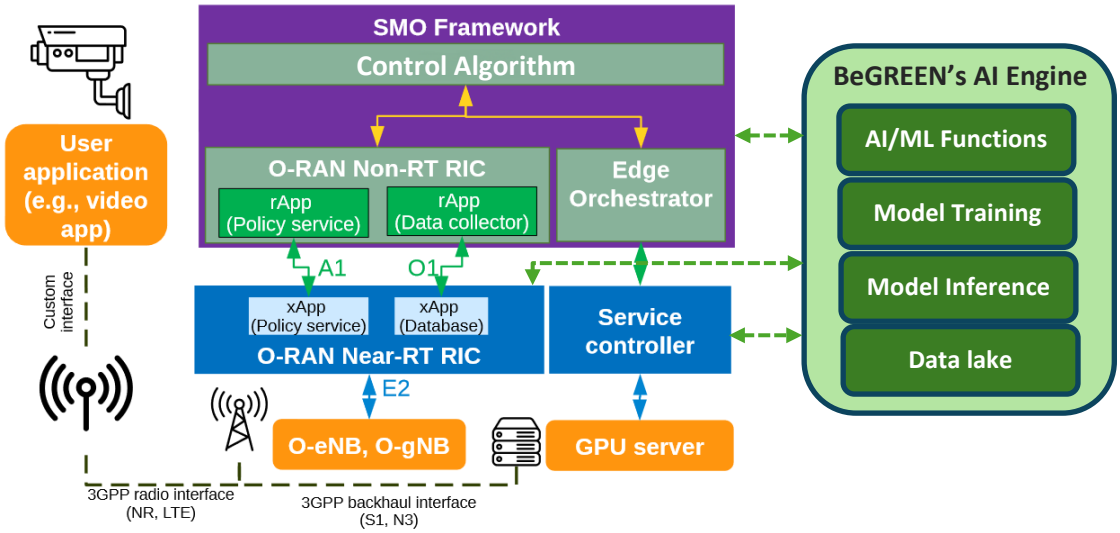


Figure 2-54 O-RAN compliant system architecture for joint orchestration of RAN and edge services

3 AI/ML-Assisted Procedures to Enhance Energy Efficiency

This chapter describes the proposed AI/ML-based optimisation strategies which address the BeGREEN objectives for enhancing energy efficiency of the BeGREEN RAN infrastructure. These strategies are based on AI/ML algorithms which optimize energy efficiency of software-based user-plane processing functions and develop service-aware AI/ML algorithms which optimize overall system energy efficiency of RAN and edge infrastructure. A first stage of this work has already been delivered in the BeGREEN D2.1, especially chapter 6.

Section 3.1 proposes the use of eXplainable AI (XAI) to identify entities and areas of the network where energy savings are achievable and, consequently, to improve energy efficiency. Section 3.2 proposes the use of AI/ML algorithms to dynamically dimension and allocate the computing resources needed for each vBS in an O-RAN O-Cloud Computing Platform to improve the network performance and reduce energy consumption. In section 3.3, AI/ML algorithmic solutions are described to intelligently switch on/off cells according to RAN status and traffic predictions. Sections 3.4 and 3.5 propose AI/ML solutions for RIS and relay control, respectively, with the aim of reducing the energy consumption and improving the network performance. Section 3.6 proposes a solution to enhance the energy efficiency of edge servers hosting UPF NFs by properly managing the CPU frequency of the edge server according to traffic status and prediction. Finally, Section 3.7 proposes a joint orchestration of vRANs and Edge AI services to minimize the power consumption of the whole system (vBS and edge server) subject to the performance constraints of the service.

3.1 Dimensionality reduction and explainable AI

BeGREEN will apply advanced AI/ML approaches to the problem of increasing energy efficiency. With the advancement of XAI, it is possible to identify influencers beyond traffic. Using telemetry, data influencers can be identified (e.g., interference and mobility in case of RAN). The challenge is to utilise the available data in an efficient way so that key inferences can be drawn from the data as using a minimum amount of processing and data storage. Reducing the dimensions of the data used to make decisions about possible energy overuse will also make it possible for these decisions to be made more quickly at levels of the management infrastructure that require faster response times. BeGREEN will explore efficient methods for identifying key influencers of energy efficiency and use identified influencers to determine areas of the network where it is likely that more energy efficiency gains can be made.

3.1.1 State-of-the-art

The explainability of models is an important consideration as it provides insights to users of the model and information on how to identify sub-optimally performing areas of a network.

In [58], an efficient algorithm for computing Shapley Additive Explanation (SHAP) values for piecewise linear decision trees and additive ensembles based on them was proposed. This work first examines model explainability and interpretability with reference to SHAP values which represent the average marginal contribution of a feature value across all possible coalitions.

In [59], which builds on the findings of the EU Celtic-Plus SooGREEN Project, the authors use SHAP values to calculate consumption shares for each service in the base station that are proportional to the marginal contribution.

The application of XAI to O-RAN is addressed in the survey paper [ORAN_XAI]. No work was identified that combined both paradigms but this paper studies the promising deployment of XAI on top of the AI-enabled O-RAN. Areas identified where XAI could contribute to AI-enabled O-RAN include Quality of Experience (QoE) optimisation, traffic steering and resource allocation optimisation.

BeGREEN will leverage the data available at both the non-RT and Near-RT RIC levels to discern the maximum amount of information from the network to help increase energy efficiency. The O-RAN Near-RT RIC Architecture includes a component for data sharing between xApps [95]. BeGREEN will share the required model information and outputs in a way that best supports the constraints of the components involved and with maximum efficiency. It may be useful to train models at the non-RT level and make these models available to xApps at the Near-RT level. Also, calculations by xApps and model outputs can be shared back to the non-RT level and these insights can be used for model retraining or other purposes.

3.1.2 Design principles

There are few challenges foreseen for running the algorithms as xApps at RT/Near-RT RIC level.

1. Certain types of AI/ML algorithms and techniques such as deep learning, require huge amounts of network events for the algorithm to outperform classical AI/ML algorithms such as linear or logistic regression, decision trees and k-means clustering, at training stage. And they demand higher compute and memory or need to parallelize model training for quicker training. Performing such expensive operation at xApps level for several VNFs will be energy expensive.
2. Network events are different according to the use case network conditions. Hence network data at the edge will be restricted only to the network conditions in that small region.
3. To meet the latency needs of xApps (<1 sec) any additional operations such as monitoring and model training/re-training, other than model inference will be expensive at the edge.

To meet the above challenges, BeGREEN will explore a learning approach which leverages information from both non-RT and Near-RT level to maximize efficiency of model training and deployment. Typically, a model is trained, monitored, and re-trained at SMO level and the model can be deployed at xApps level for quick inference. Additionally, the real time data during inference is generally not sent to SMO level. Only model performance is exchanged to SMO as part of the intelligent plane.

It is planned to use Explainable AI methods to achieve dimensionality reduction on the features that need to be analysed to make predictions of energy usage or other metrics such as traffic load or mobility related metrics. Minimizing the amount of data that needs to be processed to identify entities in the network that are operating with low energy efficiency can provide increases in the speed of detection and reductions in the processing power and data storage required for this task [61].

Explainability techniques will also be used to identify areas of the network where energy savings are achievable. Discovering the factors in the available data for the network elements which predict energy will allow possible areas of energy wastage to be quickly and efficiently identified.

3.1.2.1 Initial design

The approach to dimensionality reduction described in [61] will be applied in the BeGREEN project. This comprises techniques for calculating the marginal contribution of individual features in a dataset to energy efficiency. These techniques can be effectively used to significantly reduce the amount of data that need to be processed to detect sub-optimally configured elements in the network that may need reconfiguration to maximize their efficiency. The method involves the use of regression algorithms to determine feature importance and the use of SHAP values to determine the marginal contribution of each feature.

The dataset in [87] comprised performance and configuration metrics from a radio telecommunications network over a three-week period. Several regression algorithms were tested and Extreme Gradient Boost (XGBoost) and Random Forest Regression gave the highest accuracy prediction R^2 scores. Results from [61] demonstrate that high predictive accuracy can be obtained from greatly reduced feature sets.

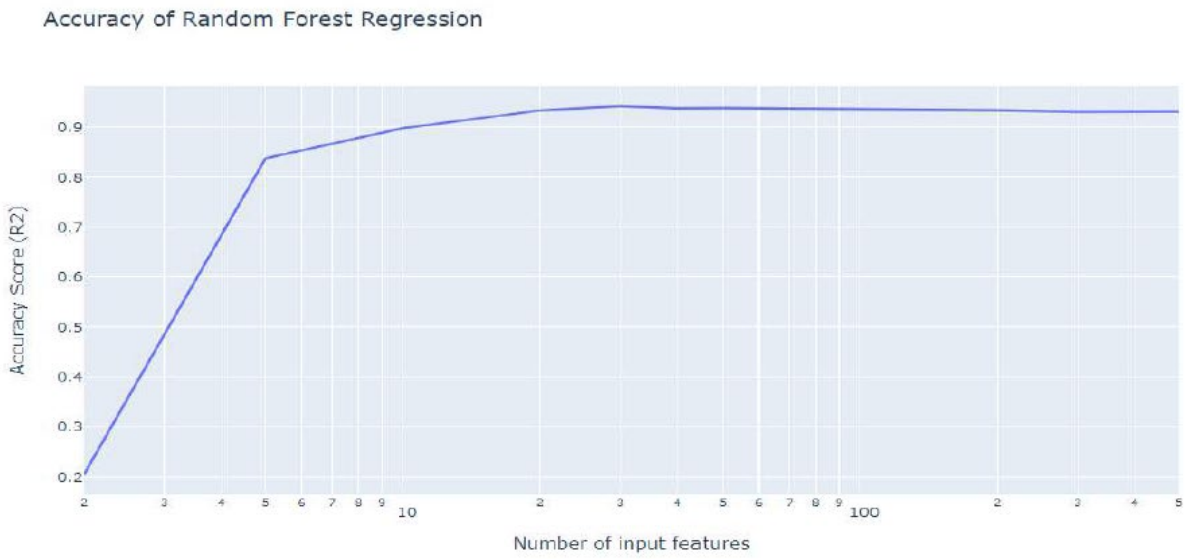


Figure 3-1 Dimensionality reduction

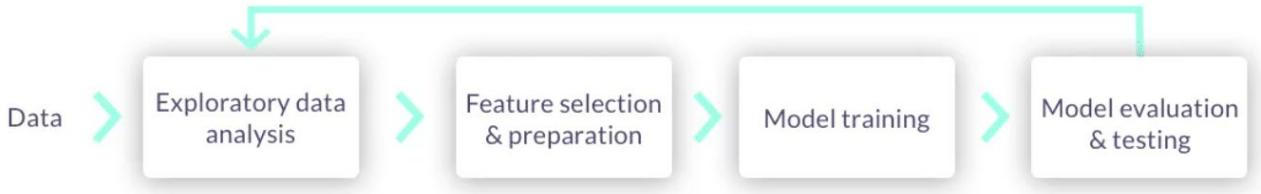


Figure 3-2 Model Development Flow

To find the variables that contribute most to the energy efficiency of an entity in the radio network, it is initially necessary to use a regression algorithm on the entire dataset and assess its capability to accurately predict the energy efficiency of a given entity. The regression algorithms' predictors provide an ordered list of feature importance which is used to sequentially reduce the list of input features while the regression and prediction are repeated. Subsequently a series of tests are executed which finds the R2 score of progressively smaller numbers of input features in the range between 2 and 500. The results are plotted on a logarithmic scale below for better visual representation.

Reducing the size of data sets used for prediction and inference wil allow more flexibility regarding which level model training and other actions take place at. It will also save energy by using less data storage and processing in the machine learning processes used by the project. Figure 3-2 shows an example of a general model development flow used in a MLOps framework. The feature selection step in the general model development flow can be enhanced using these explainable AI and dimensionality reduction techniques to limit the selected features in an explainable way.

3.2 Virtualized resource allocation in vRAN

RAN virtualisation is a crucial technology for reducing the Total Cost of Ownership (TCO) of 5G RAN infrastructure [62][63]. vRANs are expected to import the advantages of NFV, such as exploiting general-purpose computing platforms and shortening deployment cycles. Nonetheless, while shared computing platforms offer enhanced flexibility and cost-effectiveness, they also introduce challenges for 5G base stations, as they compromise the predictability offered by dedicated platforms [64][65]. The term *noisy neighbour's problem* has been coined to refer to the issue when shared resources are consumed in extremis, meaning that another function restricts one virtualized function's resources.

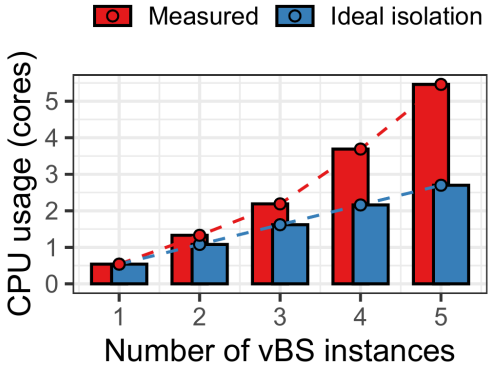


Figure 3-3 vRAN per-core CPU usage with # of vBS

The virtualisation of base stations is not alien to this issue. Virtualized resources for vRANs must be properly isolated and allocated to prevent performance degradation. This is confirmed with based on findings from experiments in a proof-of-concept vRAN system comprised of vBS instances implemented with srsRAN[40]. Using Docker containers, we deployed a set of 10MHz vBS instances in a pool of CPU cores from an Intel core i7-7700K CPU @ 4.20GHz in a shared off-the-shelf server. We then initiated bidirectional data flows, both uplink (UL) and downlink (DL), with maximum load and good wireless channel conditions between each vBS instance and a corresponding UE.

Figure 3-3 depicts the relative CPU usage of the system as a function of the number of vBS instances deployed. The bars in blue show the expected usage assuming perfect resource isolation in place. We compute this by linearly scaling up the CPU usage of a single vBS instance. The red bars show the actual CPU consumption, which unveil an exponentially growing overhead induced by the aforementioned resource contention in imperfectly isolated computing platforms.

In the context of vRAN, exploring the gains and impact of radio network function virtualisation may prove challenging to consider RAN specific characteristics. On the one hand, the vBS workload has strict time deadlines, which makes them much more sensitive to the noisy neighbours' problem than classical VNFs such as switches or firewalls. We confirm this in Figure 3-4, which shows the normalized throughput performance of one vBS for different CPU allocations (x-axis). Note that its throughput rapidly collapses upon deficit of computing resources. This occurs because PHY layer deadlines are missed, which causes users to lose synchronisation with the vBS, resulting in connectivity loss. This differs significantly from the cases of regular VNFs, which suffer from a smoother performance degradation upon computing resource shortages. Figure 3-5 shows the relationship between the normalized energy consumption on top of the system's baseline (i.e., idle) consumption of our vRAN platform as a function of the total computing load. The computing load and the energy are linearly related [66][67]. Hence, it is an essential problem to compute required shared computing resources for vRAN deployments accounting for such impact of the noisy neighbour problem.

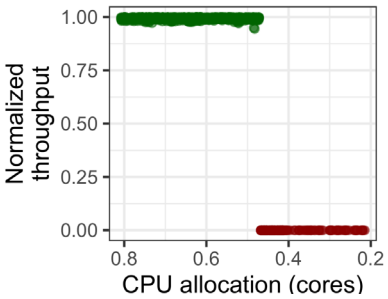


Figure 3-4 Throughput vs. CPU allocation

[40] https://www.srsran.com

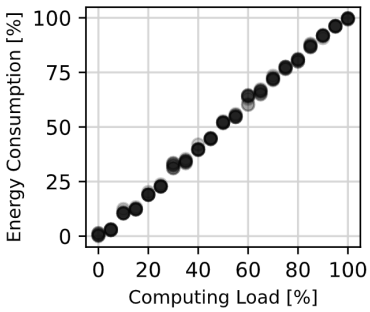


Figure 3-5 Energy consumption as a function of the computing load

3.2.1 State-of-the-art

The noisy neighbour problem has been studied for different types of virtualized resources. The works in [68][69][70] propose different solutions to effectively isolate the networking stack. The authors in [68] show that containers' computing time processing packets via interrupts is not correctly accounted. Therefore, the authors develop a solution called Iron which effectively charges the computing time used for processing packets to each container. Authors in [69] develop PicNIC, a predictable virtualized NIC abstraction that effectively provides predictable performance to cloud providers. Finally, [70] tackles a similar problem than [69] as it focuses on developing methods for efficiently sharing the virtual NICs. However, it goes along a different path. Rather than developing techniques to share the NIC resources to meet user provided NIC, the authors develop different isolation methods so that different tenants can run different applications on top of multitenant data centers' SmartNICs.

Several works explore the noisy neighbours problem in the context of shared memory resources. We review different solutions that propose efficient sharing mechanisms to share the computing and memory resources in virtualized environments. To begin with, we highlight the work in [71] which designs a solution that provides fairness in the entire shared memory of a computing system. The solution tackles both shared caches and memory controllers. To achieve this goal, the authors propose a mechanism that gathers dynamic feedback information about the unfairness in the system and uses this information to dynamically adapt the access memory rate at different computing cores. On [72] authors focus on estimating slowdown of individual applications due to interference in the main memory. The authors present their model called Memory-Interference Induced Slowdown Estimation (MISE) that estimates slowdowns caused by memory interference. The authors demonstrate the effectiveness of their model by developing two new memory scheduling schemes: 1) one that provides soft quality-of-service guarantees, and 2) another that explicitly attempts to minimize the maximum slowdown (i.e., unfairness) in the system. Following MISE, the authors improve their solution in [73] and develop a solution to estimate the slowdown of an application due to the interference from other applications and propose different strategies to better share the memory resources.

Finally, we present different works, [65][74][75] which tackle the noisy neighbour problem in the context of vRANs. The different works on the vRAN orchestration build upon the Open RAN paradigm to provide intelligent solutions on resource allocation for the deployment of vBSs over a shared cloud computing platform. The authors in [65] enhance the physical layer pipeline of operations of a vBS so that it is suitable to run in a non-deterministic computing platform. The authors in [75] develop a solution to increase the CPU utilisation in vRAN platform opportunistically co-locating non-5G workloads while ensuring correct operation. Finally, in [74] the authors develop Agora, a system that can handle the high computational demand of real-time massive MIMO baseband processing on a single many cores server. To achieve this goal, the authors identify the different opportunities of parallelism in massive MIMO baseband processing and exploit them across multiple CPU cores.

3.2.2 Design principles

Cache memory is a very relevant resource that is often overlooked. Although Docker provides efficient mechanisms to partition and isolate different types of resources, it does not provide features to partition cache memory resources effectively. However, cache-intensive applications sharing memory resources tend to evict each other's cache values, which increase the number of cache misses [76].

A core executing a thread loads the most used memory blocks into a cache for faster access. Then, every time a thread references a memory block that is not in a cache, the core triggers an interrupt called a "cache miss" and looks for the data in a higher-layer memory cache. A cache miss results in an increased number of computing cycles. While the extent of the impact of cache misses on the increase in CPU cycles may vary with technology, higher cache misses consistently result in increased computing cycles.

To study the impact of cache contention in vRANs, we used the tool perf to measure the ratio of cache misses, CPU cycles and instructions required by one vBS in a system with 1-to-5 vBS instances. These measurements are summarized in Figure 3-6 and Figure 3-7 which show, respectively, the instructions executed per cycle (IPC), and the number of cache misses per 1000 instructions (MPKI). Both metrics show high correlation. Figure 3-6 evinces that an increasing number of vBS instances has a huge impact on computing efficiency. The red line indicates a boundary point of operation where the system processes 1 instruction per cycle [77]. On the one hand, when IPC > 1, the application is instruction-bounded, i.e., only improving the efficiency of the software code can improve the IPC performance further. On the other hand, when IPC < 1, the application is likely bound by a bottleneck when accessing resources other than CPU, such as memory. In the case of Figure 3-6 the latter occurs for a number of vBS instances larger than 2. Such a bottleneck is remarkable, allowing only 0.6 instructions per cycle when 5 vBSs are instantiated.

Conversely, Figure 3-7 shows a dramatic growth of cache misses per instruction, a 500% increase with 5 vBSs with respect to 1. This, and the strong correlation between cache misses and IPC dynamics, lead us to infer that cache memory is the bottleneck in our vRAN system and, ultimately, the root cause of the anomalous CPU behaviour shown in Figure 3-3.

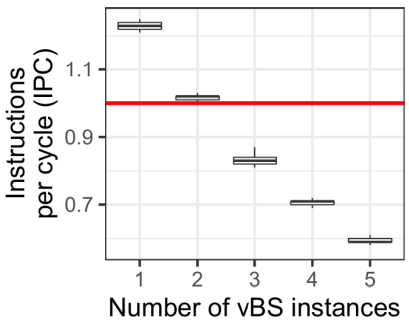


Figure 3-6 Instructions per cycle (IPC) as a function of a vBSs

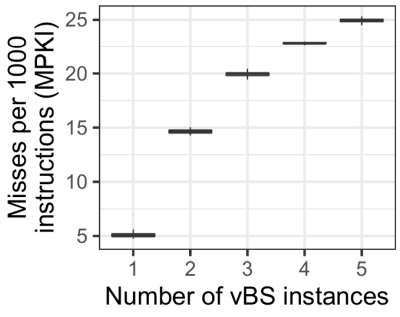


Figure 3-7 Cache Misses per 1000 instructions (MPKI) of a vBS

3.2.3 Initial design

In this section, we first formalize our problem and then we describe our proposed solution, named AIRIC. AIRIC aims to minimize the operating cost of the vRAN infrastructure (based on CPU usage).

3.2.3.1 Problem definition

The computing requirements of a vRAN system are hard to quantify dynamically. To begin with, the amount of computing resources required by a single vBS instance depends on the network traffic demand on both DL and UL directions, the signal-to-noise ratio (SNR) of each wireless link and the associated Modulation and Coding Scheme (MCS) used for communication, in a complex manner [78]. Moreover, estimating the actual requirements for a set of vBS instances sharing a platform is even more challenging because the overhead introduced by computing resource contention (noisy neighbours problem) depends on the computing cores used to process each vBS workload, the amount of isolation across vBS instances, and the maximum computing capacity available.

On the one hand, over-dimensioning the allocation of computing resources incurs high infrastructure costs as many computing cores might not be needed when running a small number of vBS instances or when the aggregated load is low, and the electricity bill associated with unneeded active cores can be substantial. On the other hand, pooling a reduced number of cores across many instances (i.e., forcing vBSs to share) may lead to throughput loss because heavy resource contention leads to severe computing overheads. A shortage of computing resources (due to the influence of the noisy neighbours problem) may cause that the users associated with vBSs in the system lose synchronisation, induce a high number of radio link errors, and cause very high end-to-end latency and jitter.

Moreover, though pinning vBS workloads to specific CPU cores provides better isolation and performance determinism, as shown before, it requires activating a larger pool of CPU cores, which incurs higher energy costs. Hence, our approach is to let all the vBS instances fairly share a pool of CPU cores, using a standard scheduler, and determine dynamically the smallest set of active CPU cores in the pool at every time step to minimize energy costs. The key novelty in our approach is that we do so in a reliable manner, accounting for the costs of sharing.

3.2.3.2 System model

We consider an O-RAN cloud computing platform (O-Cloud) supplying shared computing resources to multiple vBSs. Within this framework, we consider an agent hosted by the SMO system operating according to O-RAN specifications. This agent is responsible for making decisions in discrete time intervals namely “decision intervals”, following O-RAN's non-RT RIC guidelines. Its main objective is to minimize the operating costs of the vRAN infrastructure. To achieve this, we design the agent to dynamically dimension and allocate the computing resources needed for each vBS, taking into account the relationship between the multiple vBS, the resource contention in the computing platform (noisy neighbours problem), and the network performance.

Our agent employs an O-RAN-compliant monitoring system that gathers metrics from the various O-RAN components such as O-DU and O-CU (e.g., number of users, channel quality of the users, traffic demands) and measurements from the O-Cloud platform (i.e., usage of the computational resources). The Near-RT RIC uses the E2 interfaces to periodically receive different radio metrics from the components deployed in the O-Cloud platform [38]. Afterward, the Near-RT RIC passes the data using the O1 interface to the non-RT RIC. On the other hand, to gather metrics from the O-Cloud platform, the agent sets up performance management (PM) jobs that collect different infrastructure metrics (i.e., computing usage, energy consumption) using the O2 interface [38]. All these metrics are stored in the datalake of the BeGREEN AI Engine to train the algorithms.

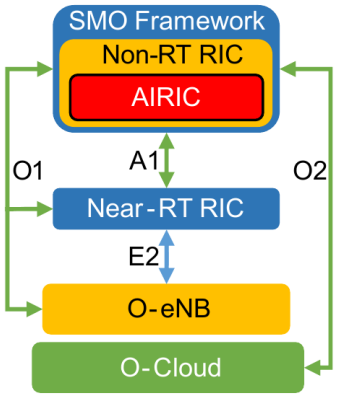


Figure 3-8 AIRIC within O-RAN

Finally, to enforce the different computing policies that our agent computes, it uses the O2 interface to pass those policies to the O-Cloud platform. The training and inference of the learning agent are performed through the BeGREEN AI Engine. Figure 3-8 AIRIC within O-RAN depicts how our agent integrates into the ORAN architecture.

Given the hard-to-model nature of the noisy neighbour problem, we advocate for RL to design our agent. In this way, the agent observes the context and takes an action at the beginning of each decision interval, and then receives a reward at the end of the decision interval. The learning agent stores 3-tuple samples comprised of the context, actions, and the associated rewards at every interval, and uses these experiences to learn and improve the obtained rewards over time. Note that, we do support a number of active vBS instances that may vary over time. To the best of our knowledge, this is the first solution that optimally allocates computing resources in a vRAN system accounting for the overhead of the noisy neighbours problem and a dynamically changing number of vBS instances in the system.

3.2.3.3 Solution design

A variable number of vBS instances imply that the dimensionality of the context information also varies over time. This is particularly challenging to support with standard RL solutions. To address this, we augment a classical Deep Q-Network (DQN) approach [79] with a Relation Network (RN) mechanism [80].

The basic idea of an RL agent is to learn an optimal policy π by interacting with an environment E in discrete time intervals $t \in \{1, 2, \dots, T\}$. Every interval, an agent observes a state (or context) $\vec{s}^{(t)}$, selects an action $a^{(t)}$ and receives a reward $r^{(t)}$ at the end of the time step. A policy π is a distribution of actions over the different states, which captures the goodness of the state-action pair $(\vec{s}^{(t)}, a^{(t)})$. Once the reward $r^{(t)}$ is measured, the system transitions to state $\vec{s}^{(t+1)}$. After T intervals, E reaches its terminal state and the agent refines its policy π using past observations $\{\vec{s}^{(1)}, a^{(1)}, r^{(1)}, \dots \vec{s}^{(T-1)}, a^{(T-1)}, r^{(T-1)}\}$. The goal is to maximize the total discounted reward

$$\boldsymbol{R^{(t)} = r^{(t)} \sum_{t'=t+1}^{T} \gamma^{t'} r^{(t')}}. \quad (3\text{-}1)$$

Most RLs approximate value functions that estimate the importance of actions given a state $\vec{s}$. One of those value functions is $Q^*(\vec{s}, a) := max_\pi E[R^{(t)} | \vec{s}^{(t)} = \vec{s}, a^{(t)} = a]$, which represents the maximum expected return given an action-state pair under the policy π. The optimal Q^*-value function follows the Bellman Optimality Equation, which provides $Q^*(\vec{s}^{(t)}, a^{(t)})$ in terms of $Q^*(\vec{s}^{(t+1)}, a^{(t+1)})$:

$$\boldsymbol{Q^*(\vec{s}, a) = E[r^{(t)} + \gamma Q^*(\vec{s}^{(t+1)}, a^{(t+1)}) | \vec{s}^{(t)} = \vec{s}, a^{(t)} = a]} \quad (3\text{-}2)$$

Using the Bellman Optimality Equation, we can find $Q^*(\vec{s}, a)$ iteratively [81]. Our approach uses neural networks to approximate the optimal $Q^*(\vec{s}, a)$, which is called DQN [79].

In particular, given the large timescale of the non-RTnon-RT RIC, the action taken at one interval $a^{(t)}$ has

little impact on the next state $\vec{s}^{(t+1)}$ and therefore it is enough to maximize instantaneous reward. Hence, to expedite convergence, we simplify our RL setting into a contextual bandit problem by setting $\gamma = 0$ and $T = 1$.

3.3 AI/ML-based algorithmic solutions for non-RT RU control

Energy efficient RU control through switching on/off cells is one of the main use cases of O-RAN's energy saving optimisations due to the significant impact of these components on the overall energy consumption of the network [82]. AI/ML has a relevant role in such scenarios since load and energy consumption predictions are key to enable proactive automated control-loops with enhanced energy efficiency without impairing traffic status [24]. To this end, it is needed to have access to relevant data from real deployments for sufficiently large periods of time.

The approach presented in this section relies on the access to a dataset from a Spanish MNO with real data from the 4G and 5G sites and cells located in a specific area of Spain. This dataset will be used to implement and evaluate RU/cell load and energy consumption predictors, which will feed a RU/Cell Control rApp taking decisions on on/off switching and traffic steering.

3.3.1 State-of-the-art

The O-RAN Alliance introduces in [24] the foreseen workflows and involved components of the O-RAN architecture to support the application of carrier/cell on/off switching to obtain energy savings in the RAN. Although no specific method is described, AI/ML is expected to be used in order to enrich the decision-making process of control rApps and/or xApps; for instance, by exposing the prediction of cell load or utilisation, user mobility or energy consumption. The methods can be applied by rApps or xApps, depending on the targeted control-loop period. Similarly, in the first white paper of SMART-5G project from the ONF [83], authors discuss requirements, gaps, and trade-offs to be considered when defining carrier/cell on/off switching methods enhanced with AI/ML.

Authors in [84] investigate the utilisation of ML algorithms for traffic forecasting and how they impact the trade-off between consumer energy and QoS when applied to enrich strategies for the management of base station sleep modes. The work evaluates different ML models, with a different number of trained models per cell. The input data is based on a dataset from a large Italian MNO. Results show that predictions have a strong dependency on the traffic patterns of the area, and that, although having specific models per cell enhances prediction accuracy, simple ML algorithms using a single model for all the cells can also produce a very effective and flexible approach to energy efficiency and QoS. Also using a dataset from an Italian Telco, [85] proposes a method based on Long Short-Term Memory (LSTM) and Gaussian Process Regression (GPR), which offers an accurate single-cell level prediction. It aims at improving long time and burst data prediction by differentiating the periodic and random/burst components of the dataset and applying to them different ML methods. Authors in [86] claim that data imbalance has a significant impact on training, leading to biased and incorrect load estimates, which may impact energy saving optimisations. Imbalance scenarios include datasets where most load samples do not belong to high load regimes or with a different load distribution among cells. Therefore, they propose a solution that incorporates a Balancing Load Function to address these issues.

3.3.2 Design principles

The main objective of this optimisation is to intelligently switch on/off cells according to RAN status and traffic predictions. To this end, two main developments are needed. First, the cell load predictor, which will be used to obtain the expected traffic per cell in a specific area. Secondly, the control rApp which will make use of this prediction to determine which cells can be stopped to increase energy efficiency in the area

without impairing data traffic. Additionally, the implementation of the cell load predictor will be completely integrated with the AI Engine. This way, as a secondary objective of this solution, we will demonstrate the capabilities of the AI Engine to host ML models and to train, monitor, infer and expose them rApps.

As aforementioned, i2CAT has access to a dataset with real data from the 4G and 5G sites and cells located in a specific area of Spain. This dataset, which has a time granularity of 15 minutes, includes hundreds of KPIs, such as the uplink and downlink throughput, the distribution of users among the cells, the utilisation of PRBs. It also contains measurements of the total energy consumption of the sites. Thus, the first step is to characterize the relationship between the load and the power consumption of the sites. Figure 3-9 depicts an example for one of the weeks considered in the dataset, which shows a clear correlation between 5G throughput (uplink plus downlink) and energy consumption, and between both KPIs and the time of the day.

Figure 3-10 shows the Energy Efficiency in (b/J) computed using the values shown in Figure 3-9 (data not normalized). Results clearly indicate that the energy efficiency of the site increases with a higher throughput, due to a better utilisation of resources. This implies that optimisations focused on intelligently switching off cells and steering traffic to properly use available resources, will benefit the overall energy efficiency of the network.

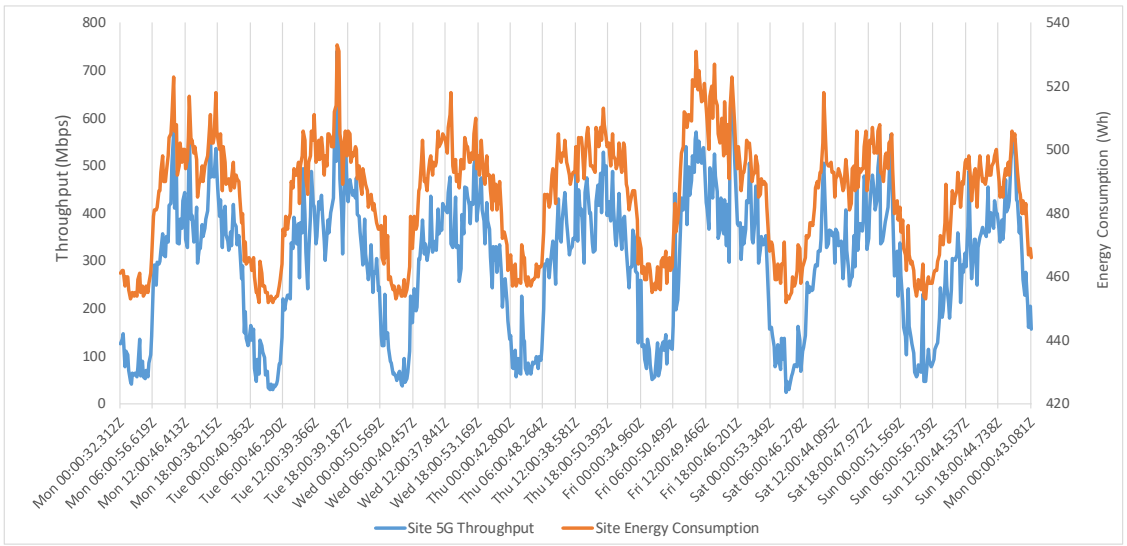


Figure 3-9 Example of site energy consumption and 5G throughput for one week (i2CAT dataset)

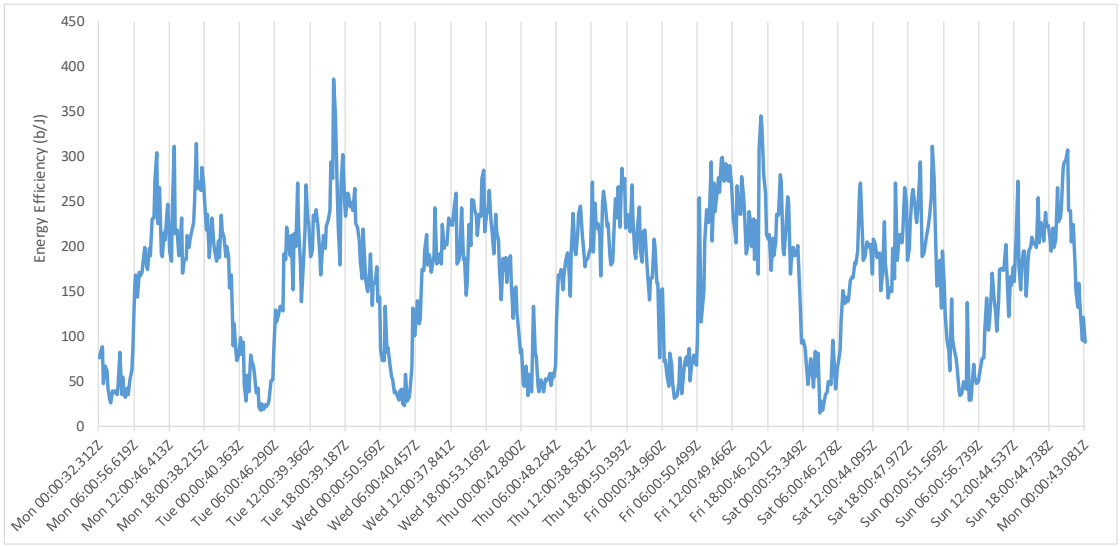


Figure 3-10 Example of the energy efficiency of one site for one week (i2CAT dataset)

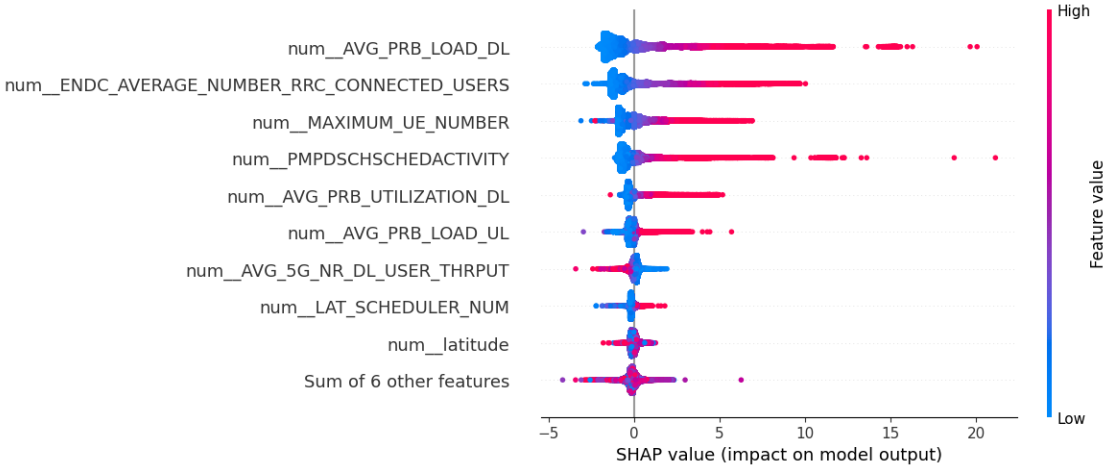


Figure 3-11 Cell load prediction: SHAP values to identify main influencers

Using this dataset, we will create load and energy consumption predictors of the cells and/or sites. Due to the high number of KPIs available in the dataset, we did a first preselection of the main features that may influence the load and energy of the cells; for instance, KPIs related to the PRB utilisation in downlink and uplink directions, the average number of UEs, the position of the cells, etc. Then, we have tested regression algorithms such as Gradient Boosting through the XGBoost Python library[41], which is a well-known and widely used regression algorithm of the state of the art which offers high accuracy and scalability in large datasets [87]. Additionally, it can be used for time series forecasting, which seems a useful approach due to the clear correlation of load and energy consumption with the time [88]. Finally, aligned with the proposal detailed in Section 3.1, we are analysing the utilisation of XAI to help identifying the main influencers in the predictors. Figure 3-11 depicts an initial example of the SHAP values of the features being used to predict the load of the cells, showing that high values of PRB consumption in the downlink and of number of users during the last period have a significant impact on the predicted load for the next period. Nevertheless, specific details about the final implementation and evaluation of the predictor will be reported in next WP4 deliverables.

Once the load prediction is available, we will use it to decide which cells or base stations should be switched off or on and how to steer the traffic. In particular, we are considering a use case based on switching on/off booster 5G cells in 5G NSA urban scenarios. The objective is to steer the traffic to the primary 4G cells in cases where 4G and 5G cells are underutilized, for instance during off-peak hours, thus increasing the energy efficiency in the area. The next section details how this control loop will be implemented through the BeGREEN Intelligent Plane.

3.3.3 Initial design

Figure 3-12 depicts the architecture of the cell on/off control solution and how it is integrated within the BeGREEN Intelligent Plane including the AI Engine and the RICs.

The implementation of the solution will consist of two main components, the Cell Load Predictor (CLP) plus its associated Assist rApp, and the RU Control rApp, which are described as follows:

[41] https://xgboost.readthedocs.io/en/stable/index.html

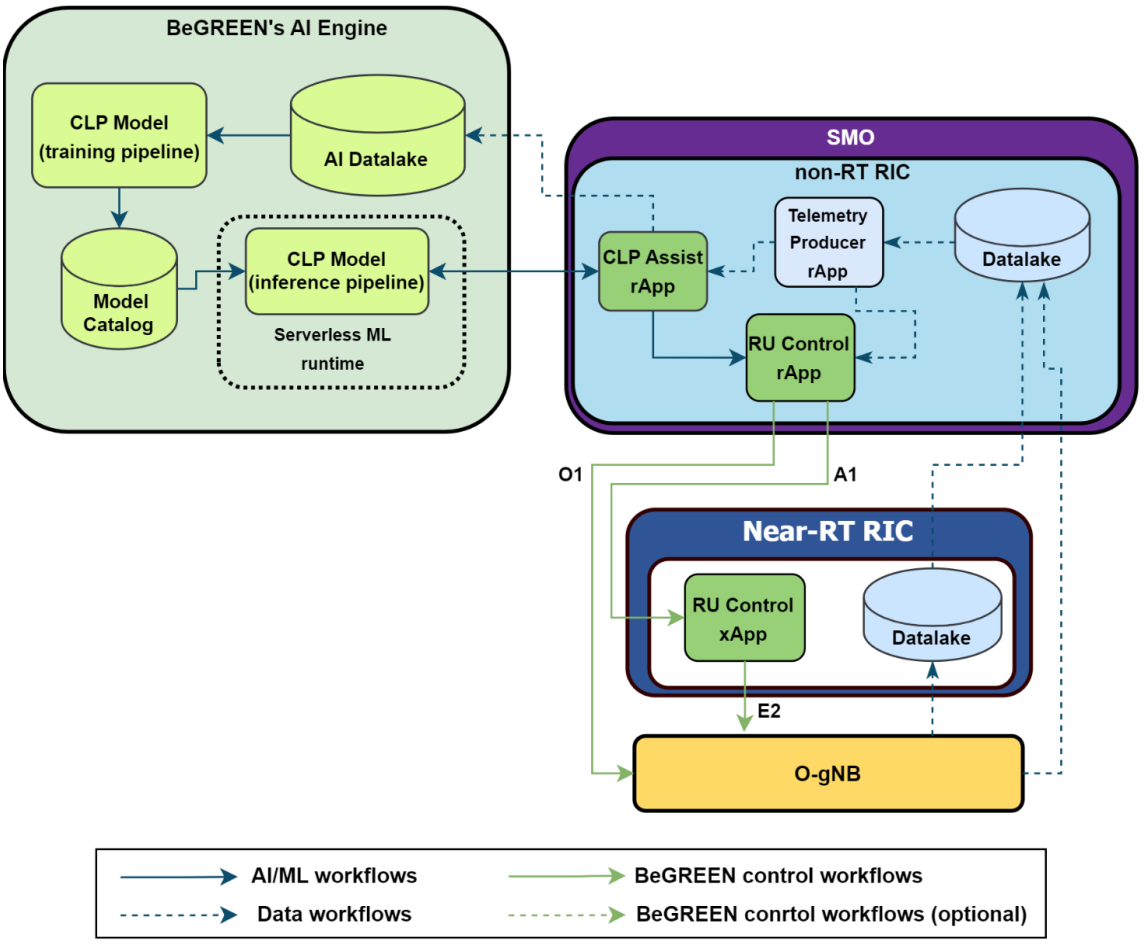


Figure 3-12 Cell on/off control; BeGREEN primary architecture

- CLP and CLP Assist rApp: The CLP will offer cell load predictions to the RU Control rApp as introduced in the previous section. To this end, the Assist rApp will manage (i) the exposure of the required telemetry data from the RAN to produce the prediction, (ii) the trigger of the predictor according to RU Control rApp subscription, and (iii) the exposure of the predicted data to the RU Control rApp. Additionally, the Assist rApp may forward performance KPIs regarding the inference, which may be used by the RU Control rApp to support its decision-making process. For instance, during peak-hour periods cell off decisions could be conservative depending on the prediction accuracy.
- RU Control rApp: This rApp will use load predictions and RAN telemetry to determine the status of the RU cells. As introduced in the previous section, it will try to distribute the predicted load in the targeted area with the objective of increasing the energy efficiency. This will entail steering traffic among cells and switching on/off cells. Once the optimal allocation has been computed, it will make use of O1 (direct RU control) and/or A1 (indirect RU control through xApps plus traffic steering control) interfaces to realize it.

Note that, as presented in the previous section, we will make use of a real dataset to create and train the ML model focused on cell load prediction. Also, we will use sections of this data to demonstrate and evaluate the predictor, and the optimisations done by the control rApp, i.e., we will calculate the impact of the decisions taken by the RU Control rApp using the data stored in the dataset. Finally, within the scope of WP5 emulated RAN data and control from TeraVM might be used to demonstrate the proposed solution.

According to the previous definitions, Figure 3-13 and Figure 3-14 illustrate, respectively, the possible workflows for the training and inference phases of the RU Load Prediction model.

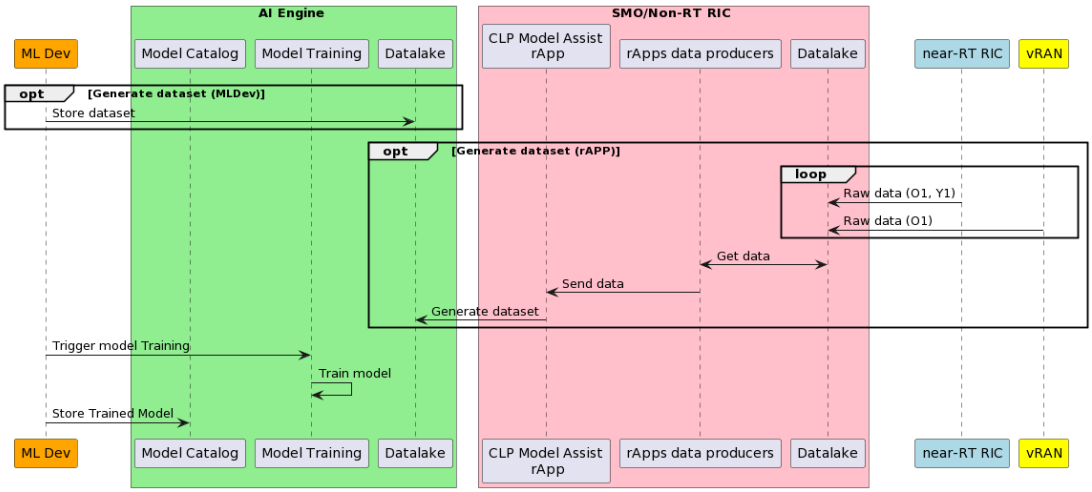


Figure 3-13 Cell load prediction model; training workflow

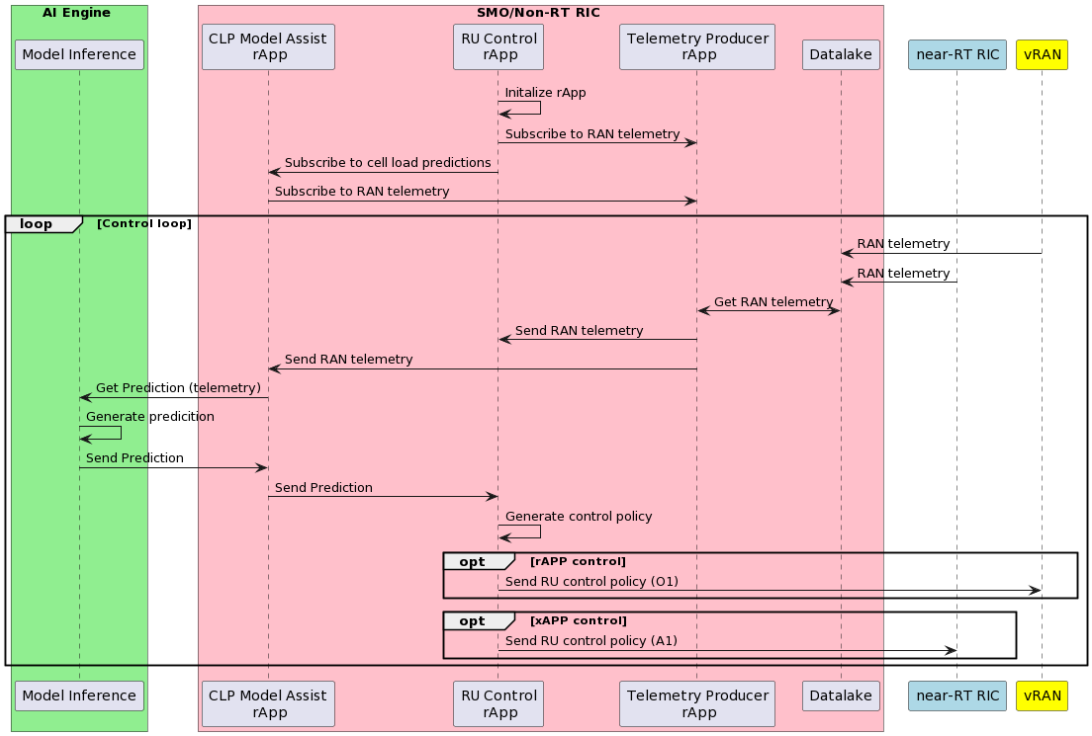


Figure 3-14 Cell load prediction model; inference workflow

3.4 AI/ML-based RIS control

In BeGREEN, we will aim for increasing energy efficiency of edge services such as those that optimize the way gNBs manage the UEs. To do so, we will leverage both Integrating Sensing and Communication (ISAC) solutions and Reconfigurable Intelligent Surfaces. Firstly, we will use ISAC to obtain location information of one or more UEs. Secondly, based on this information, we may use the RIS infrastructure to provide coverage to those UEs if they are in range, allowing to power off the redundant RUs. We will study different AI/ML approaches for assessing when users are covered by sufficiently signal strength coming from the RIS, and when it is safe to power off RUs without significantly impacting the performance of the users.

3.4.1 State-of-the-art

Wireless networks are currently witnessing remarkable advancements thanks to the ISAC paradigm. While it

has been used since the 1960s after the invention of the radar, the maturity level achieved nowadays offers new opportunities to implement sensing by utilizing already existing wireless infrastructures. This would mean that future networks will go beyond classical communication and could provide ubiquitous sensing services to measure surrounding environments. In Liu et al. [89] a comprehensive overview of ISAC is provided, shedding light on the dual-functional wireless networks that are poised to play a central role in 6G and beyond. The most interesting and widely studied use cases for ISAC are in the topic of detecting, tracking and localisation of users. In recent research, Liu et al. [90] explored the integration of ISAC in practical 5G networks, showcasing the potential of ISAC in outdoor multi-target localisation, and focusing on increasing network efficiency and lower operation costs. In the same line, Gómez-Vega et al. [91] conducted experiments, showcasing the feasibility of mm-wave ISAC technology for similar purposes. A number of works also exist targeting high-precision indoor position in 5G NR such as Gao et al. [92], and preliminary studies that open new industry and transport applications, such as Ko et al. [93], where a high-speed train positioning system using 5G NR signals is developed.

RIS technology is promising to revolutionize wireless communications as well. RIS represent a resource-efficient approach for introducing dynamic alterations into the wireless communication landscape within 5G and Beyond 5G systems. Such smart surfaces, comprise numerous nearly passive elements, numbering in the hundreds or thousands, all capable of assuming various roles. These roles encompass functions such as signal relay and blocking, precise position estimation, obstacle identification, narrow beamforming, and the sculpting of multipath characteristics. Remarkably, RIS can subtly control the propagation of radio waves without relying on active power amplifiers, allowing for the focused enhancement of signal performance, the alleviation of obstructions, and the extension of radio coverage into previously inaccessible dead zones. For this reason, RIS are suitable to provide energy-efficiency improvements in the RAN.

There has been a significant progress in the literature to achieve energy-efficient wireless communication through the deployment of RIS technology. Huang et al. [94] was one of the first works that explored the practical implementation of RIS to optimize energy efficiency in wireless communication. Their research emphasizes the real-world applicability of RIS technology in achieving greener and more sustainable wireless networks. Similarly, Yang et al. [95] studied how to dynamically control RIS on-off status and optimize reflection coefficients in a distributed way. Also, Jia et al. [96] provided insights into the application of RIS in enhancing energy efficiency in device-to-device (D2D) communication networks, and Liu et al. [97] introduced preliminary ideas on how to apply AI/ML in RIS technology. This integration may be used in BeGREEN to optimize beams with AI/ML to better cover user movements. In this sense, Wang et al. [98] demonstrated how RIS can be used to enhance location awareness in beyond 5G networks. Their work highlights the role of intelligent surfaces in improving signal coverage and reducing interference, ultimately leading to more efficient network operation.

3.4.2 Design principles

In order to increase energy efficiency of edge services, we will study how to leverage ISAC and RIS to intelligently activate/deactivate RUs or gNBs in order to save energy. An example of such a use case is shown in Figure 3-15. An estimation of where UEs are located can be obtained by using ISAC or fronthaul information (e.g., sector id). Alternatively, AI/ML models can be used to predict user trends and estimate the location of the users based on, e.g., the hour or the day of the week. As we detect that users tend to be located in an area with RIS coverage, we can opportunistically turn off the RU 2. Therefore, the users will be provided with service by the RU 1 (on the right) via the RIS, which extends its coverage to areas where the RU 1 does not reach on its own. This would result in covering the active users of RU 2 with a lower energy footprint.

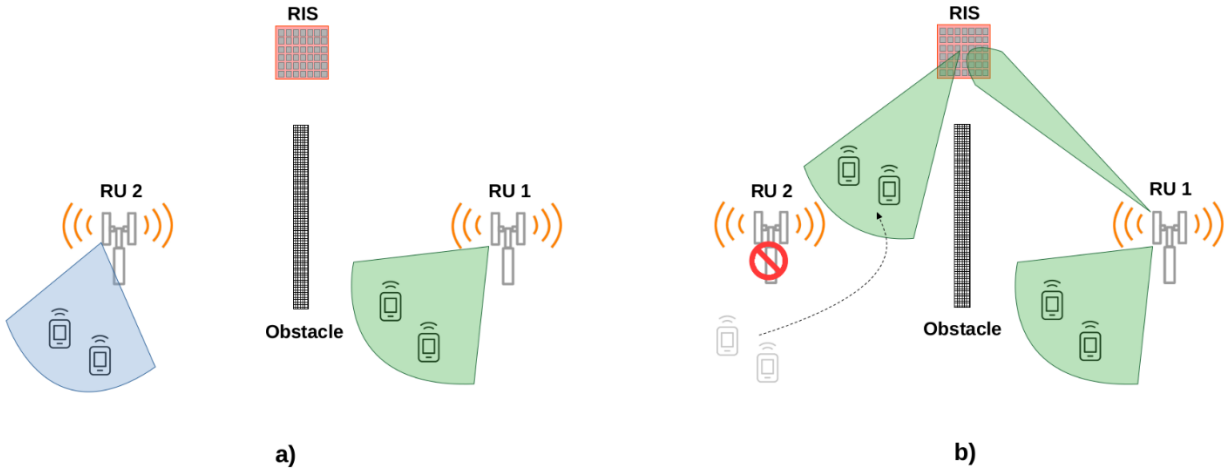


Figure 3-15 Use case example of using RIS and ISAC for improving energy efficiency

To this end, we need intelligent algorithms that allow the Near-RT RIC to decide when to switch off the RU 2 and how to configure the RIS beam. Such algorithms may benefit of the available monitoring data available in the RAN, such as number of users, their location, and their traffic demand pattern (which may change between the day and night).

3.4.3 Initial design

Leveraging the initial RISE-6G architecture described in Section 2.2.2.3, here we describe an initial design of how RIS and edge services can be intelligently controlled within the O-RAN architecture in BeGREEN. Figure 3-16 shows how the main 3GPP and O-RAN interfaces are jointly integrated with the RIS-specific interfaces as reported in RISE-6G. Besides self-configuring RIS, there are two main approaches to actively control a RIS:

- Direct control from the gNB. An operator that owns both the gNB and RIS infrastructure may opt to use the F1-x interface to control the RIS. This interface would connect the O-CU-CP to the RIS. Therefore, a RIS controller deployed in an xApp at the Near-RT RIC would communicate with the RIS passing through the O-CU-CP first.

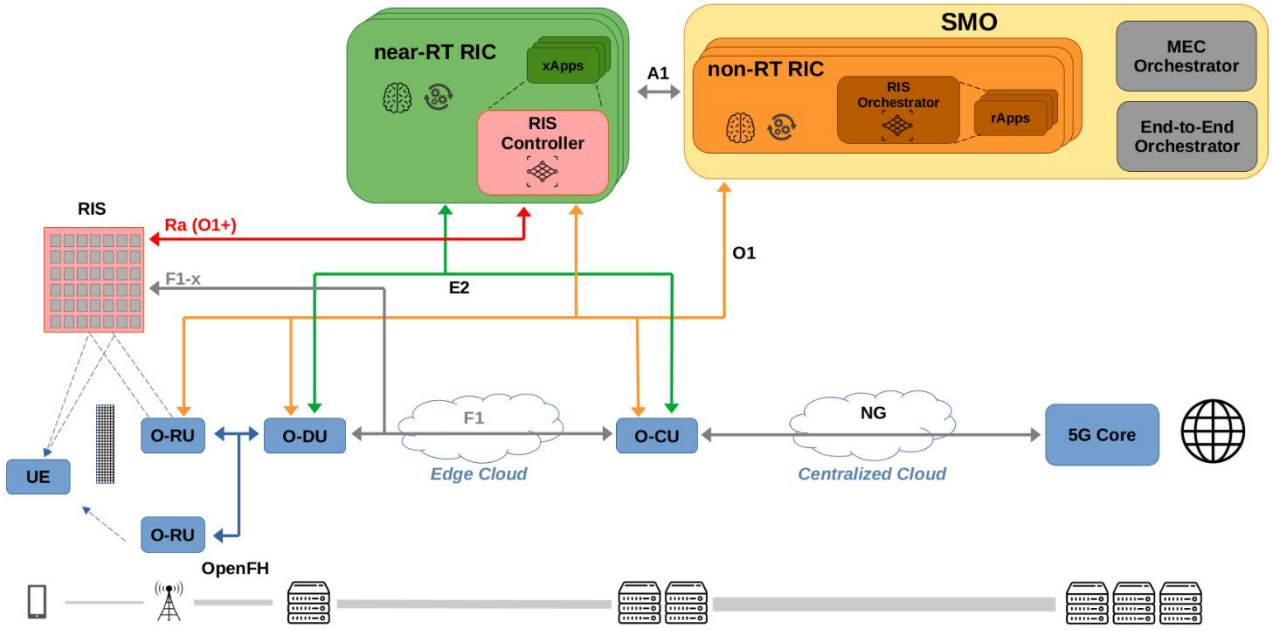


Figure 3-16 Initial network architecture accommodating RIS into O-RAN

- Direct control from the Near-RT RIC. An operator that owns the gNB but externalizes the RIS infrastructure may use the Ra interface, which would connect the RIS Controller xApp located in the Near-RT RIC directly with the RIS, bypassing the O-CU-CP. The Ra interface could be a virtual interface with similar functions as the ones provided by an extended O1 interface (O1+), or an additional ad-hoc interface connecting the RIS with the Near-RT RIC.

In both cases, there is a need for a physical control link to the RIS hardware. While wired out-of-band control links are the most straightforward approach, they involve extra deployment costs. On the other hand, wireless control links, either in-band or out-of-band, require additional implementation effort, but allow the operators to deploy them in a more flexible and inexpensive manner.

To coordinate the RIS operation and the energy-efficient gNB/RU management, the RIS controller and orchestrator requires access to the monitoring data available in the RAN. User and gNB related metrics such as RSRP from the different cells/sectors, PRB usage per user, gNB/RU throughput, number of users per gNB/RU, overall cell throughput, etc. may be used by the RIS controller to infer when a gNB/RU may be switched off and replaced by an already deployed RIS. Also, additional explicit information related to the user location may be obtained by ISAC solutions deployed in the RAN.

Such information can be used to train AI/ML models that forecast user trends or user location tracking which can be used to jointly manage the RUs and the RIS. Such AI/ML models can be trained at wider time scales in the BeGREEN Intelligent Plane, which contains the BeGREEN AI Engine with different models that may fit different goals, e.g., Decaying Deep Q-Network (D-DQN) models with loose timing requirements as the one presented in [97] for high mobility scenarios, or low-complexity greedy searching approaches as the one presented in [95] for static, real-time scenarios. Also, the Intelligent Plane would implement the datalake containing internal and external metrics, and a number of rApps with similar goals that the RIS Controller can interact with.

At shorter time scales, the RIS Controller, which may be located at the near-RT RIC, would need not only to decide when a gNB/RU can be switched off, but would also need to configure the RIS adequately. Because of this, given the large number of parameters to be optimized based on the contextual information and the large amount of information that is available for the RIS Controller, AI/ML techniques are considered to be a pervasive and effective solution for addressing such complex and resource consuming tasks. However, tasks such as AI/ML training (e.g., gradient propagation), which usually require a considerable amount of time and data, should be deployed in the non-RT RIC due to the timing requirements being less strict. On the other hand, the Near-RT RIC may perform the inference of AI/ML methods such as Feedforward Neural Networks (FNN) or Recurrent Neural Networks (RNN). On one hand, FNNs are a common type of artificial Neural Networks where input data only propagates forward, without feedback loops. Convolutional Neural Networks (CNNs) can also be considered a sub-category of FNNs. For the case of the RIS configuration problem, CNNs may be useful to capture the spatial structure and correlations of the matrix-shaped RIS antenna-array deployment [100]. On the other hand, RRNs introduce feedback loops, which make them suitable to work on temporal sequences of data with correlated samples. Such techniques may be useful, for example, for RIS-assisted time-varying channel estimation problems [101].

In the following Deliverable D4.2 we will ground the design of our RIS-enabled framework, specifying which AI/ML techniques are the most suitable to address the energy efficiency problem of BeGREEN.

3.5 AI/ML-based algorithmic solutions for relay-enhanced RAN control

This section provides a description of the proposed algorithms for the different relay control functionalities described in section 2.2.2.4. A general description of the state of the art, the considered design principles and an initial design description is presented for each of the relay control functionalities. Figure 3-17 shows the considered architecture for the relay control.

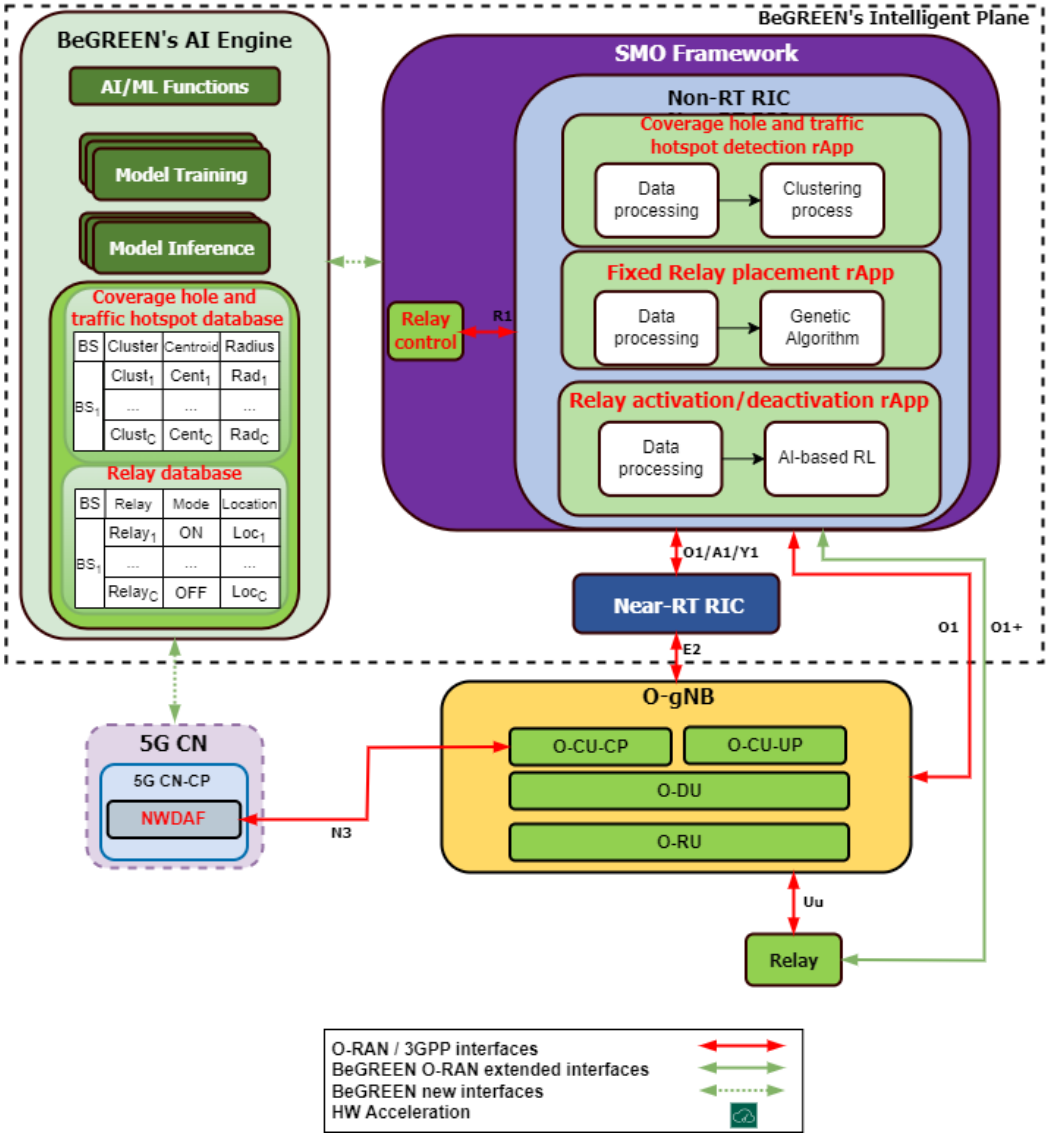


Figure 3-17 AI/ML algorithmic solutions

The proposed AI/ML processes related to coverage hole and traffic hotspot detection, fixed relay placement and relay activation/deactivation are placed in the non-RT RIC. These processes could also be implemented in the AI Engine. More specific details will be provided in BeGREEN D4.2.

3.5.1 Detection of coverage holes and traffic hotspots

This process is in charge of the detection of geographical regions with poor coverage (i.e. coverage holes) and the detection of geographical regions with a high traffic demand (traffic hotspots).

3.5.1.1 State-of-the-art

The detection of coverage holes in a cellular network has been traditionally done based on drive tests which are costly, time-consuming and only provide a partial view of the network since the testing is limited to locations with vehicle access. With the aim to overcome these challenges, the Minimisation of Drive Tests (MDT) mechanism was standardised by 3GPP [53]. The MDT allows to obtain UE measurement reports related to signal strength and quality associated with the UE location. These reports allow to generate e.g. coverage maps that allow the identification of coverage holes.

The availability of such mechanisms to determine the UE geographical location and to report these location

measurements to the network, enables the exploitation of this information by using (Big) Data analytics technologies [102][103]. In this respect, the use of AI/ML methodologies, e.g., clustering of geographical UE locations with low coverage levels, may lead to a more accurate network performance characterisation that allows more adequate actions in the network (e.g. deploying a new relay, relay activation/deactivation, network reconfiguration actions, etc.). As an example, [104] proposed the use of unsupervised learning to identify coverage holes based on a Uniform Manifold Approximation and Projection (UMAP) algorithm. Other unsupervised learning approaches have been proposed in the literature, as in [105] that proposed the use of a clustering based on K-means to identify regions of weak coverage. Other works, such as [106][107], analyse the network performance data to identify regions with weak coverage and high traffic volume by using a clustering technique known as Density-Based Spatial Clustering of Applications and Noise (DBSCAN).

The spatial traffic characterisation in a cellular network has been traditionally conducted by collecting traffic load statistics at a cell level. However, a more granular understanding of the spatial traffic distribution is necessary for an adequate deployment of fixed relays inside the coverage region of some specific cells. Fixed relay deployment can deal with the presence of traffic hotspots inside a specific cell guaranteeing the user performance and reducing the overall power consumption. Several approaches have been proposed in order to obtain a higher resolution view of the spatial traffic distribution. Some works, such as [108][109], aim to extract information of user mobility to provide a space-time traffic characterisation and identify geographical regions with peaks of traffic. Other works, such as [110][111], propose the use of context aware methods based on data coming from social networks. As an example, [110] forecasts the changes in cellular hotspots based on Twitter data. Then, it makes use of a clustering process based on DBSCAN to identify geographical regions with peaks of traffic.

3.5.1.2 Design principles

The proposed architecture for the detection of coverage holes and traffic hotspots is presented in Figure 3-17. As shown, network measurements are periodically collected and sent to the datalake. The datalake stores a large number of historical measurements collected during a large period of time at different geographical locations. Then, when the coverage hole and traffic hotspots detection rApp is executed, it makes use of this collection of historical measurements for the identification of geographical regions with weak coverage (i.e. coverage holes) and geographical regions with high traffic demands (i.e. traffic hotspots). This process is run in a cell-by-cell basis.

For the identification of coverage holes in a specific cell, described in Figure 3-17, the data processing step collects the historical measurements at this cell and selects the set of geographical locations inside the cell with weak coverage, e.g., locations with an RSRP below a specific threshold. The selected location samples with weak coverage are used as input for the clustering methodology. Then, the clustering process identifies groups of samples that are geographically close together and far from other samples that will belong to other clusters. The result of the geographical clustering process is a list of clusters that represent geographical regions with poor signal quality, i.e., each cluster corresponds to a coverage hole and is represented by a list of the geographical locations that belong to this cluster. Then, each coverage hole can be characterized as a circle centred at the cluster centroid with a cluster radius calculated as the distance between the cluster centroid and the furthest sample of the cluster. Similarly, this process can also determine geographical regions with high traffic demands. The output of the clustering processes is stored in a coverage hole and traffic hotspot database that contains a list of geographical regions with coverage problems or high traffic demands in each base station.

3.5.1.3 Initial design

The proposed clustering process is done by means of the DBSCAN algorithm. DBSCAN is a very popular clustering algorithm that divides the data samples into a number of specific clusters or groups, such that the

data samples in the same cluster have similar properties, different from data samples of other clusters. One of the advantages of DBSCAN is that this clustering methodology does not define a priori the number of clusters to be considered. In the proposed approach, the DBSCAN algorithm works by grouping together samples of the dataset that are geographically close based on the density of samples in each geographical region.

The key idea of the clustering process is to identify a group of samples inside a neighbourhood of radius *epsilon* with a minimum number of samples *min_samples*. The algorithm starts by selecting an arbitrary sample and finding all its neighbouring samples within an *epsilon* distance and saving them in a *Neighbour_list*. If there are less than *min_samples* samples within this radius, the sample is marked as noise. Otherwise, a new cluster is formed. Then, the algorithm checks the samples of the *Neighbour_list* and their neighbourhood to see whether new samples must be added to the *Neighbour_list*. The algorithm also checks whether each sample of the *Neighbour_list* can be added to the cluster. When all the samples of the *Neighbour_list* are visited, the current cluster is finished. Then, the algorithm selects another unvisited sample and repeats the process until all the samples have been assigned to a cluster or marked as noise. Details of the DBSCAN algorithm can be found in [112] and described in Figure 3-18.

3.5.2 Fixed relay placement

This process is in charge of determining the necessity of deploying new fixed relays and setting their configuration parameters (geographical location, transmitted power, etc.) to guarantee the coverage requirements, improve the spectral efficiency and reduce total power consumption.

3.5.2.1 State of the art

Concerning the problem of fixed relay placement (i.e., identifying the necessity of deploying new fixed relays and determining their location and their initial configuration parameters), several solutions have been proposed in the last few years. The problem of relay placement can be considered as an optimisation problem and, consequently, most of the proposed solutions are based on meta-heuristic algorithms.

```
For all unvisited samples S in the dataset.
    Mark S as visited
    Select an unvisited sample S
    Determine all the N neighbour samples of S within epsilon
    distance and save them in Neighbour_list.
    If N < min_samples:
        Mark S as noise.
    Else:
        Create a new cluster C
        Add S to cluster C
        For each S' in Neighbor_list:
            If S' is not visited
                Mark S' as visited
                Determine all the N' neighbour samples of S' within
                epsilon distance and save them in Neighbour_list'.
                If N' ≥ min_samples
                    Join Neighbor_list with Neighbor_list'
            If S' is not member of any cluster
                Add S' to cluster C
```

Figure 3-18 DBSCAN algorithm

Specifically, [113] deals with coverage extension by means of D2D relaying and proposes a greedy solution to determine the optimal relay locations. Similarly, [114] makes use of different approaches based on Particle Swarm Optimisation for determining the adequate placement of the relays in order to extend coverage at the cell edge in 4G/5G cellular networks. Other works proposed methodologies to obtain the appropriate placement of Integrated Access and Backhaul (IAB) nodes. In particular, [115] makes use of Genetic Algorithm based optimisation.

3.5.2.2 Design principles

The decision of deploying a new relay is usually based on the identification of a geographical region where the user coverage/quality requirements are not satisfied. For this reason, the fixed relay placement process is closely related to the coverage hole and traffic hotspot detection process. The proposed approach assumes that the decision to deploy new fixed relays and the initial selection of their network configuration parameters is done at the Relay placement rApp in the non-RT RIC, see Figure 3-17. This Relay placement rApp makes use of information of the configuration parameters of the relays (available at the Relay database in the datalake) related to e.g. the location of the relay, whether the relay is on/off, etc. The rApp also makes use of the information related to the coverage hole and traffic hotspot characterisation available in the datalake. The rApp may consist of a metaheuristic optimisation algorithm e.g. with a genetic algorithm as described in section 3.5.2.3. The output of the Relay placement rApp is the location of the new fixed relays, and their transmitted power.

3.5.2.3 Initial design

The proposed fixed relay placement methodology initially assumes the necessity to deploy R relays to provide service to the identified regions with coverage holes and/or traffic hotspots characterized in the datalake. For this purpose, a genetic algorithm is proposed. A genetic algorithm is an optimisation technique inspired by the principles of natural selection and genetics [116] that can be used to find solutions to complex problems by mimicking the process of evolution. The genetic algorithm consists of iteratively proposing candidate solutions (individuals), each one consisting of a combination of relay locations and their transmitted power, and evaluate each individual according to a cost function. The best individuals in an iteration (generation) are combined to obtain new individuals that are again evaluated in the subsequent generation. This process is repeated during multiple generations until a termination condition is fulfilled.

The following steps summarise the operation of the proposed genetic algorithm:

1. *Generation of a population of N individuals*. Each n-th individual (n=1,…,N) is represented by a vector $\boldsymbol{v_n}$ with 3R elements that correspond to the (x,y) geographical location and the P_T transmitted power of each of the R relays to be deployed, i.e. $\boldsymbol{v_n}=\{x_1, y_1, P_{T\_1},\ldots, x_r, y_r, P_{T\_r},\ldots x_R, y_R, P_{T\_R}\}$. Each element is called a gene, and it is initialised randomly among a set of feasible relay locations (i.e. relays may be placed in some specific geographical locations with good propagation conditions to a specific BS) and a set of possible values of the transmitted power.
2. *Execute the following sub-steps for each n-th individual*:
   2.1 Feasibility check: This step checks whether the individual (that represents the location and the transmitted powers of the different relays) guarantees the overall network coverage or not. An individual is not feasible if there is any geographical location with an RSRP lower than a threshold. In order to check this, it is necessary to use a propagation model between the BSs and the relays and between the relays and each of the geographical regions of the scenario. If an individual is not feasible, a new individual is randomly generated.
   2.2 Calculate the cost of each individual: It is determined according to a cost function that measures the power consumption of the relays when the r-th relay transmitted power is P_{T_r}:

$$c_n = \sum_{r=1}^{R}\left(P_{0,r} + \Delta_{p,r} \cdot P_{T\_r}\right) \quad (3\text{-}3)$$

where $P_{0,r}$ is the minimum power consumption at the r-th relay and $\Delta_{p,r}$ is the slope of the load dependent power consumption.

3. *Selection:* This step determines which feasible individuals are selected for the generation of new individuals to be evaluated in the subsequent generation. The cost of each individual c_n is used for the selection process by considering a roulette-wheel [116] so that the probability P_n of selecting the n-th individual for recombination is:

$$P_n = \frac{\frac{1}{c_n}}{\sum_{i=1}^{N}\frac{1}{c_i}} \quad (3\text{-}4)$$

According to this, individuals with lower cost have higher probability to be selected for recombination.

4. *Recombination:* This process combines the different genes (i.e. elements of the vectors $\mathbf{v_n}$) of the two individuals selected in the previous step (called parents) to generate two new individuals (called children). The rationality of this process is to search for new solutions similar to the best individuals of the previous generation by combining their genes. The recombination process considered here is the so-called "1-point crossover" [116]. Considering the genes of the two parents, a crossover point is defined randomly and all the genes beyond this crossover point are swapped between both parents to obtain the children.

5. *Mutation:* This operator makes small random changes in the genes of the two individuals obtained after recombination. The probability of selecting a gene for mutation is *1/3R*. As a result, very few genes of an individual are usually modified by this process. When a gene is selected for mutation, the new value is either an increase or decrease (with equal probability) in one resolution unit, allowing only changes to feasible values.

As a result of the selection, recombination and mutation process, a total of two new individuals will be obtained. This process is repeated until getting the *N* individuals for the new generation. With the newly generated *N* individuals, the algorithm executes the evaluation procedure again and computes the associated costs. The process is repeated iteratively until reaching a maximum number of generations. Then, the final solution of the algorithm is the individual with minimum cost that has been found throughout all the generations.

3.5.3 Relay/RUE activation/deactivation

This process is in charge of taking adequate decisions of relay activation/deactivation with the objective of improving the network performance and reduce energy consumption.

3.5.3.1 State-of-the-art

Adequate decisions of relay switching on/off by means of AI/ML techniques can contribute to obtaining large energy savings in the network while maintaining the service requirements. In this context, a critical functionality is the relay activation/deactivation function. This function is in charge of deciding under what type of conditions a fixed or moving relay or a Relay UE (RUE) can be switched on/off to optimise the network performance obtain energy savings.

In the context of UE-to-Network relaying, different RUE activation strategies have been studied in [117] based on different criteria and context information. Results of [117] revealed that the most efficient strategies from the perspective of outage probability reduction are those that account for the number of UEs that would be served by a RUE based on the experienced spectral efficiency. Leveraging the outcomes

of this previous work, a functional framework for supporting the RUE activation was presented in [118] based on the characterisation of each potential RUE through a utility metric that measures the coverage enhancements brought to the network when the RUE is activated.

3.5.3.2 Design principles

The non-RT RIC includes a relay activation/deactivation rApp that determines when, where and under which conditions fixed relays and RUEs need to be activated or deactivated. The decision on whether to activate or deactivate a relay depends on the benefit in terms of the performance experienced by the UEs connected to it and the cost of having the relay active, e.g., in terms of energy consumption. In particular, for the activation/deactivation of RUEs, acquiring knowledge about the UEs in the area of a BS and their behavioral patterns will be relevant. For example, let us suppose a situation in which a UE is located inside an office building during working hours. Then it is highly likely that it remains stationary and is connected to its serving BS for long periods of time. If the signal quality of this UE is sufficiently good and its battery level is above a certain threshold, this UE can be considered as a potential candidate RUE to be activated.

The relay activation decision-making (for both fixed relays and RUEs) is executed at the non-RT RIC by the Relay activation/deactivation rApp, as shown in Figure 3-17. It is supported by a relay database that contains the list of fixed relays and UEs capable of acting as RUEs in the BS. The relay database includes different fields such as the identifier of the relay, whether the relay is a fixed relay or a RUE, the relay position, whether the relay is on/off at a certain period, etc. These data, together with specific performance measurements collected from the network and the information available in the coverage hole and traffic hotspot database, constitute the inputs to the relay activation/deactivation rApp so that it can decide to activate/deactivate the relays at each period.

Let us focus on the relay activation problem of the R relays that are associated to a given b-th BS. The status of the r-th relay ($r=1,\dots,R$) is represented by $a_{b,r}=\{0,1\}$, where 0 means that the relay is deactivated and 1 means it is activated. Therefore, the global status mode configuration associated with BS b can be defined as the R-length vector $\boldsymbol{C_b}=\{a_{b,r}\}$. The objective of the considered problem is to find a policy that optimally activates/deactivates the relays. This means finding the optimum configuration $\boldsymbol{C_b}(t)=\{a_{b,r}(t)\}$ to be applied at every time t.

Considering that an active relay consumes a certain power level, one possibility is to minimise the time that relays are activated serving a number of users lower than a threshold Th but also minimize the time that the relays are deactivated when a large number of users need to be served. This can be represented by minimizing:

$$\boldsymbol{min}\ \frac{1}{N}\sum_{n=1}^{N}\frac{1}{R}\sum_{r=1}^{R} c_{b,r}(t_n) \tag{3-5}$$

where

$$c_{b,r}(t_n)=\begin{cases}\alpha & if \quad a_{b,r}(t_n)=1 \ and \ N_{b,r}(t_n)<Th\\ \beta & if \quad a_{b,r}(t_n)=0 \ and \ N'_{b,r}(t_n)\geq Th\\ 0 & otherwise\end{cases} \tag{3-6}$$

Decisions are made in times $t_n, t_{n+1},\dots$ with separation ΔT. N is the number of decisions made during a certain observation time. $N_{b,r}(t_n)$ is the average number of UEs served by the r-th relay in period (t_n, $t_n+\Delta T$) while $N'_{b,r}(t_n)$ is the average number of UEs that would have been served by the r-th relay in period (t_n, $t_n+\Delta T$) if the relay had been active. According to equation (37), $c_{b,r}(t_n)=\alpha$ if the r-th relay was active in the previous time period (i.e. $a_{b,r}(t_n)=1$) and the number of served users was below a threshold (i.e. $N_{b,r}(t_n)<Th$). In turn, $c_{b,r}(t_n)=\beta$ if the r-th relay was deactivated (i.e. $a_{b,r}(t_n)=0$) but the number of users that would have been by

this relay in the previous time period was higher than a threshold (i.e. $N_{b,r}(t_n) \geq Th$). Initially, α=1 and β=1 can be considered but also other possible values may be studied.

The number of UEs that would have been served by the *r*-th relay in case that the relay had been active, $N'_{b,r}(t_n)$, can be estimated (for the case of fixed relays) according to historical values of this metric. As an example, measurements of the number of served UEs being the relay active at the same time period of the day in previous days can be used to make this estimation. For the case of RUEs, this term $N'_{b,r}(t_n)$ can be estimated by means of the Relative Proximity Analytics functionality provided by the NWDAF [119]. This type of analytics may help to identify the total number of UEs in the vicinity of a specific target UE.

Some other metrics can also be taken into account for a more adequate RUE activation/deactivation decision. As an example, the UE location, and in particular, the propagation conditions between the BS and the UE, UE mobility patterns, UE remaining battery level, etc., are important aspects to consider for the decision activating the relay capability of these UEs (i.e. to decide that a UE can act as a RUE). Clearly, a UE that it is expected to be static in the near future with good propagation conditions to the BS and a high battery level is a good candidate to become a RUE.

3.5.3.3 Initial design

The development of an efficient solution to the relay activation problem involves a multiplicity of variables, such as the propagation conditions of the relays and the nearby UEs, the traffic dynamics, etc. To address these multiple dimensions, the use of DRL is proposed. DRL techniques combine the use of Deep Neural Networks (DNNs) with Reinforcement Learning to assist a software-based agent that makes decisions in relation to a specific problem. This combination is especially interesting because of its capability to handle large state and action spaces.

Among the different DRL techniques, this work specifically proposes a solution based on the DQN algorithm [120]. In this approach, the learning process is carried out dynamically by a DQN agent that interacts with an environment, and after observing the consequences of its actions measured in terms of a certain reward signal, it learns to modify its own decision-making policy. The DQN algorithm has been selected to address the relay activation problem mainly for two reasons. The first is that the DQN algorithm has been designed to support high-dimensional states and action spaces. This is convenient for the relay activation problem since the network dynamics implies a large amount of data that needs to be considered by the agent. The second reason is that with DQN, the policy is progressively updated by considering individual samples of experience. This feature is suitable for the case of the relay activation problem since continuous learning of the policy is desired. Moreover, DQN is a useful technique for learning how to select actions from discrete action spaces, as in the problem considered here where the actions involve activations or deactivations of relays.

In the proposed solution, the DQN agent is located at the non-RT RIC and makes activation/deactivation decisions of the relays associated with the different BSs. At time *t,* the DQN agent selects an action $\boldsymbol{a(t)}$ that contains the relay activation configuration $\boldsymbol{C_b(t)}$ to be applied to the set of relays of a specific BS in the next time window of duration ΔT. The selection of a given action is dependent on the state observed at time *t* denoted as $\boldsymbol{s(t)}$ together with the available policy π at that time. The state is obtained by processing the data from the network monitoring module, the information available at the relay database and the information available at the coverage hole database.

The outcome of applying a certain action related to the relay activation/deactivation is assessed by means of a reward signal $r(t+1)$. This reward is delivered to the DQN agent at the end of the time window ΔT. It essentially measures how effective or ineffective the selected action was. The reward signal obtained over time after selecting the different actions is utilized to progressively enhance the DQN-agent decision-making policy. The main components of the proposed DQN-based solution along with the policy learning process are described below.

The state $\boldsymbol{s}(t)$ is represented as a vector associated with a particular BS *b,* and it has different components listed in the following:

- $C_b(t)=\{a_{b,1}(t),\ a_{b,2}(t),\ldots,a_{b,R}(t)\}$ denotes the configuration (ON/OFF) of all relays in the previous time period *t.*
- $N_b(t)=\{N_{b,1}(t),\ N_{b,2}(t),\ldots,\ N_{b,R}(t)\}$ corresponds to the average number of UEs that have been served by each *r*elay in the previous time period *t.*
- $N'_b(t)=\{N'_{b,1}(t),\ N'_{b,2}(t),\ldots,\ N'_{b,R}(t)\}$ is the average number of UEs that would have been served by the *r*-th relay in the previous time period if the relay had been active.

The total number of components in the state is $3\cdot R$.

A given action $\boldsymbol{a}(t)$ can be seen as a vector $C_b(t)=\{a_{b,r}(t)\}$ that contains the relay activation configuration applied every time window accounting for all the considered relays. The so-called action space contains all relay activation configurations. Since a relay has only 2 possible modes (activated and deactivated), the total number of possible actions in the action space is 2^R.
The reward signal $r(t+1)$ that assesses the efficiency of the action $\boldsymbol{a}(t)$ that is selected at a state $\boldsymbol{s}(t)$ in relation to the optimisation criterion. The reward can be expressed as:

$$\mathrm{r(t+1)}=1-\frac{\mathbf{1}}{\boldsymbol{R}}\sum_{\boldsymbol{r=1}}^{\boldsymbol{R}}\boldsymbol{c_{b,r}}(\boldsymbol{t_n}) \tag{3-8}$$

A stage of major relevance when applying the DQN technique is the training of the DQN agent. By means of the training, the agent actively learns a decision-making policy π that is used for selecting which action to apply under a particular state.

The fundamental objective of RL-based algorithms is to determine the optimal policy π^* that maximizes the so-called discounted cumulative future reward defined as:

$$\sum_{j=0}^{\infty}\boldsymbol{\gamma}^j r(t+j+1) \tag{3-9}$$

where γ represents the discount factor that takes values between 0 and 1 that is defined to give more relevance to short-term rewards. In the case of the DQN algorithm, the optimal policy results from determining the optimal action-value function denoted as $Q^*(\boldsymbol{s},\boldsymbol{a})$. This function represents the maximum expected discounted cumulative reward that can be obtained by applying an action $\boldsymbol{a}$ for a particular state $\boldsymbol{s}$ starting at a certain time *t* following the policy π. This can be expressed recursively by means of the Bellman equation as:

$$\boldsymbol{Q}^*(\boldsymbol{s},\boldsymbol{a})=\boldsymbol{E}\left\{\boldsymbol{r}(\boldsymbol{t}+\mathbf{1})+\boldsymbol{\gamma}\cdot\max_{a'}\boldsymbol{Q}^*(\boldsymbol{s}(\boldsymbol{t}+\mathbf{1}),\boldsymbol{a}')\ \middle|\ \boldsymbol{s}(\boldsymbol{t})=\boldsymbol{s},\ \boldsymbol{a}(\boldsymbol{t})=\boldsymbol{a},\ \boldsymbol{\pi}\right\} \tag{3-10}$$

Based on this definition, the optimum policy π^* is the one that selects the action that maximizes the action-value function, that is:

$$\boldsymbol{\pi}^*=\mathbf{arg}\ \max_{\boldsymbol{a}}\boldsymbol{Q}^*(\boldsymbol{s},\boldsymbol{a}) \tag{3-11}$$

The DQN algorithm makes use of a DNN to approximate the optimum action-value function $Q^*(\boldsymbol{s},\boldsymbol{a})$. In particular, the DNN takes as input each one of the components of state $\boldsymbol{s}$ and provides an output $Q(\boldsymbol{s},\boldsymbol{a},\theta)$ that represents the approximation of the optimum action-value function for each $\boldsymbol{a}$ action. The term θ denotes the weights of the different interconnections between neurons in the DNN. In this respect, the structure of the DNN includes an input layer with a number of neurons equal to the number of components in state, an output layer with a number of neurons equal to the number of possible actions and one or more hidden layers. The number of hidden layers and the number of neurons in each layer are the hyper-parameters of the DQN that are specified as part of the configuration.

The optimal action-value function can then be learned by iteratively updating the function $Q(\boldsymbol{s},\boldsymbol{a},\theta)$ during the training stage by varying the values of the weights θ in accordance with the experienced rewards. To update the weights, the DQN agent includes the following components:

- **Evaluation DNN $Q(s,a,\theta)$**: It is the DNN that approximates the optimum value function $Q^*(s,a)$. This DNN is used to extract the policy π for deciding the actions to be perform in the environment.
- **Target DNN $Q(s,a,\theta^-)$:** This is another Neural Network with the same structure as the evaluation DNN but with weights θ^-. It is used to calculate the time difference (TD) target expressed as γ. Instead of updating the weights θ^- every time step, they are updated every M time steps with the weights of the evaluation DNN $\theta^-=\theta$. As a consequence, the computation of the TD error, which depend on the target DNN, is no longer dependent on rapidly fluctuating estimates of the Q-values.
- **Experience dataset D**: This dataset gathers the experiences obtained by the DQN agent during the training process. A given experience is expressed by means of a tuple $<\boldsymbol{s}(t), \boldsymbol{a}(t), r(t+1), \boldsymbol{s}(t+1)>$ composed of the state and action performed at time t along with the obtained reward and the new state at time $t+1$. The total length of the dataset is denoted as l. The use of the experience dataset allows the random selection of mini-batches of experiences to update the weights θ of the evaluation DNN. The use of these mini-batches is called Experience Replay and allows a more efficient learning process since it breaks the temporal correlations in the training data.

When the training process starts, the weights of the target and the evaluation DNN are initialized randomly. Then, during the training process, these weights are progressively updated. Overall, the training stage involves two main parts, namely, data collection and the process of updating the weights.

The data collection is the process of filling experience dataset D with the gathered experiences. For this purpose, at each training step, the agent observes the state and chooses an action $\boldsymbol{a}(t)$ following an ε-greedy policy that selects the action based on the current policy, see equation (3-12), with probability 1-ε and a random action with probability ε. This random action selection is needed in the training process for incorporating the capability to explore new actions that are different from the ones that the current policy would select. After applying the selected action, the obtained reward is measured and placed in the experience tuple that is saved in the dataset.

Every time that the experience dataset reaches its storage capacity l, older experiences are removed and substituted by recent ones. Moreover, at the beginning of the training, the agent selects actions randomly (i.e., ε is set to 1) to gather a wide variety of experiences. This is maintained during a number of *InitialCollectSteps* training steps.

The update of the weights of the evaluation DNN is done at every training step by considering the experiences accumulated in the experience dataset. An updating process consists of making a random selection of a mini-batch $U(D)$ of past experiences J belonging to the dataset. These experiences are expressed as e_j, j=1,...,J, where e_j is an experience tuple denoted as $<\boldsymbol{s}_j,\boldsymbol{a}_j,r_j,\boldsymbol{s}_j^*>$. Then, the update is performed by means of a mini-batch gradient descent procedure. To this end, the average Mean Squared Error (MSE) for all the experiences in $U(D)$ is computed first as:

$$\boldsymbol{L(\theta)} = \boldsymbol{E}_{\boldsymbol{e_j \in U(D)}}\left\{\left[\boldsymbol{r_j} + \boldsymbol{\gamma}\ \underset{a'}{\boldsymbol{max}}\ \boldsymbol{Q}\left(\boldsymbol{s_j^*}, \boldsymbol{a'}, \boldsymbol{\theta^-}\right) - \boldsymbol{Q}(\boldsymbol{s_j}, \boldsymbol{a_j}, \boldsymbol{\theta})\right]^2\right\} \quad (3\text{-}13)$$

Then, the mini-batch gradient descent is computed by the derivative of $L(\theta)$ with respect to θ as follows:

$$\boldsymbol{\nabla L(\theta)} = \boldsymbol{E}_{\boldsymbol{e_j \in U(D)}}\left\{\boldsymbol{r_j} + \boldsymbol{\gamma}\ \underset{a'}{\boldsymbol{max}}\ \boldsymbol{Q}\left(\boldsymbol{s_j^*}, \boldsymbol{a'}, \boldsymbol{\theta^-}\right) - \boldsymbol{Q}\left(\boldsymbol{s_j}, \boldsymbol{a_j}, \boldsymbol{\theta}\right) \cdot \boldsymbol{\nabla_\theta Q}\left(\boldsymbol{s_j}, \boldsymbol{a_j}, \boldsymbol{\theta}\right)\right\} \quad (3\text{-}14)$$

The final step consists of updating the weights of the evaluation DNN $Q(\boldsymbol{s},\boldsymbol{a},\theta)$ as follows:

$$\vartheta \leftarrow \vartheta + \alpha \cdot \nabla L(\vartheta) \quad (3\text{-}15)$$

where α represents the learning rate.

Following each update of θ, the obtained $Q(\boldsymbol{s},\boldsymbol{a},\theta)$ will be used to select new actions. In relation to the weights θ^- of the target DNN, they are updated as $\theta^-=\theta$ after every P update of the evaluation DNN.

3.6 Traffic-aware CPU state management

The virtualisation and softwarisation of 5G network functions have fuelled the emergence and availability of 5G RAN and Core solutions, which can run on top of general-purpose hardware and COTS equipment. However, the utilisation of this type of hardware may penalize energy efficiency due to an inefficient utilisation of the CPU resources which are not adapted to the real traffic demands [121]. An example of this scenario is the deployment of user-plane functions such as the CU-UP or the UPF in edge servers. To this end, in this section we will propose a solution to enhance the energy efficiency of edge servers hosting UPF NFs by properly managing the CPU frequency of the edge server according to traffic status and predictions.

Note that although the proposed intelligent and energy efficient management of the 5GC resources could be seen as out of the scope of O-RAN, its definition and implementation will be tightly aligned with the O-RAN Energy Saving Use Case related to O2-based scale-in/scale-out [24]. For instance, we can expect similar solutions for managing the CPU resources of the CU-UP, whose energy consumption is also very dependent on traffic status. In addition, as introduced in Section 2.1.3, other relevant projects such as ONF's Smart-5G are following a similar approach. Therefore, we expected BeGREEN findings to be of high relevance to the O-RAN ecosystem.

3.6.1 State-of-the-art

In the context of O-RAN, the intelligent management of O-Cloud resources, namely the O-DU and the O-CU, is being considered as one of the main Energy Saving uses cases [24]. The adoption of virtualized and containerized architectures in O-RAN enables the scaling of O-cloud resources with the objective of decreasing energy consumption without impairing the network performance. To this objective, two main use cases are considered: (i) horizontal scaling by shutting down O-Cloud nodes in idle times, and (ii) vertical scaling by dynamically setting CPU energy saving modes (standardized or vendor-specific) which may tune the CPU frequency/voltage through C/P-states. In both cases, decision-making is expected to be performed by the non-RT RIC and the rApps by making use of O2 interface to monitor and manage the O-Cloud resources through the FOCOM and NFO components of the SMO. AI/ML support is also considered to empower decision-makers.

As introduced in section 2.1.3, the Smart-5G project from ONF is following a similar approach, also considering energy savings in the 5GC by scaling in/out cloud resources and managing the CPU level of the resources. In particular, and similarly to the BeGREEN approach, the project targets UPF's resource optimisations due to its strong correlation with network traffic. In [83] authors provide a brief discussion on the possible application of AI/ML techniques such as Bayesian Optimisation, Reinforcement Learning, or Deep learning, to deal with the large space of possible performance optimisation targets (e.g., efficiency, throughput, latency) and resource allocation options (e.g., cache line, memory bandwidth, and CPU states). Nevertheless, the selected approach is not defined.

Bayesian Search methods are widely considered to find the optimal configuration or allocation of resources according to probabilistic models representing the relationship between resources/configurations and the obtained performance. For instance, authors in [122] use Bayesian methods to dynamically tune UPF's CPU aiming to minimizing power consumption and packet drops. The solution includes an offline phase, where the dataset is created, and the model is trained and evaluated according to different experiments with different load and resource allocation configurations. Then, in the online phase, the classifier selects the best available CPU configuration according to real traffic and applies it by tuning the CPU frequency governor (Linux *cpufreq* subsystem toolset). Although focused on the UPF, the work doesn't clarify if results are based on a real UPF implementation and real General Packet Radio Service Tunnelling Protocol (GTP) traffic.

Regarding the application of DRL, a method for scaling UPF instances in Kubernetes cluster is proposed in [123] to save resource consumption. In the proposed approach, each UPF instance is mapped to a Pod and

it is assumed to serve a specific PDU session type with defined QoS requirements. Then, for each type of PDU session, the number of active pods defines the total supported number of sessions and the allocated resources. Therefore, the objective of the proposed DRL algorithm, based on Proximal Policy Optimisation, is to set the Pod count dynamically depending on the traffic to minimize the number of pods without impairing the network performance. The solution is compared with Kubernetes's built-in Horizontal Pod Autoscaler, which considers CPU utilisation to scale the number of pods, showing a clear benefit. Nevertheless, note that the evaluation doesn't consider the performance and characterisation of a real UPF implementation serving GTP traffic.

In addition to AI/ML-based automated decision-making using DRL or Bayesian Search approaches, ML models can also be applied to generate predictions which may enrich the decision of rule-based control algorithm. Authors in [124] compare the performance of different tree-based (XGBoost and Random Forest) and Neural Network-based (Multi-layer Perceptron and LSTM) approaches when applied to the prediction of user plane traffic in the eNB and the Serving Gateway/PDN Gateway User-plane (SPGW-U). The work includes details about the feature and hyperparameter selection of the different approaches. Finally, it concludes that tree-based methods lead to higher accuracy. Note that although it is based on a real 5G network based on OAI components, the scale of the testbed and the dataset is very limited. Also, no optimisation method using the predictions is proposed.

3.6.2 Design principles

The main objective of the designed optimisation will be to adapt the energy consumption of the edge server hosting the UPF NF(s) to its traffic load in order to maximize the energy efficiency. In addition to monitoring the actual traffic load, we will make use of ML-based load predictions to enable proactive decision-making and avoid disrupting traffic load or wasting energy. I2CAT has access to a dataset with real measurements from a Spanish MNO which will be used to create and train the predictors. For instance, Figure 3-19, depicts the variation of load in a Packet Data Network Gateway (P-GW) (5G NSA network) during a week, which shows a clear influence of the daytime on the resulting load. Thus, we assume that techniques based on time series forecasting will be a proper option to implement the required predictors.

Energy consumption adaptation will be accomplished through intelligent horizontal and vertical scaling, i.e., by managing the number of UPF instances and the CPU frequency of this instances (e.g., P-states), respectively. Since we target non-real time optimisations, we have not considered the impact of C-states, which are related to the energy consumption during CPU idle times and require of a real-time control to exploit their benefits [125].

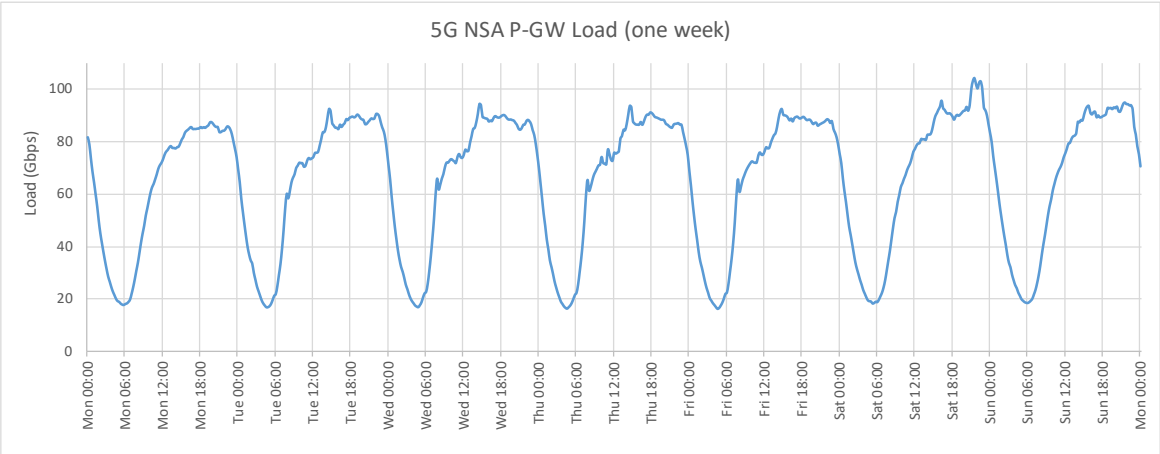


Figure 3-19 5G NSA P-GW Load for one week (i2CAT's dataset)

Initially, we have focused on the x86 CPU architectures using Linux operating systems (OS), although future work during the project may consider ARM platforms. In particular, we have selected an Intel NUC (Next Unit of Computing) platform which offers enough resources to cope with a moderate traffic volume (ethernet interfaces up to 10 Gbps). Then, we have characterized the power and CPU consumption of this platform according to the GTP traffic, the number of UPFs and the frequency of the CPUs hosting the UPFs.

Regarding the UPF implementation, we chose Open5Gs[42], an open-source implementation of the 5GC compliant with 3GPP Rel-17. The UPF implementation of Open5Gs uses a single thread or process per UPF but allows to deploy several UPFs, what enables horizontal scaling. Regarding the RAN domain, we have emulated it through UERANSIM[43] software to simplify the needed procedures during this initial phase. UERANSIM is a well-known and research-oriented open-source 5G UE and gNB simulator, which allows to generate Non-Access Stratum (NAS), Next Generation Application Protocol (NGAP) and GTP traffic. Note that the server under evaluation only hosts the UPF instances, each of them pinned to an isolated CPU. This way, we can characterize the concrete impact of the UPF managing traffic on the CPU and power consumption.

The Linux kernel supports CPU performance scaling by means of the *CPUFreq* toolset. It consists of three layers of code:

- *CPUFreq core*: Common code infrastructure and user space interfaces for all platforms that support CPU performance scaling.
- Scaling Governors: Implement algorithms to estimate the required CPU capacity.
- Scaling Drivers: They provide scaling governors with information on the available P-states (or P-state ranges in some cases) and access platform-specific hardware interfaces to change CPU P-states as requested by scaling governors.

In particular, the *intel_pstate* scaling driver allows two P-state selection algorithms which perform dynamic scaling, namely *powersave* and *performance*. These algorithms operate between a minimum and maximum frequency values which can be set using the *cpupower*[44] tool. On the one hand, the *powersave* policy tries to balance performance and energy savings by selecting the appropriate p-state according to CPU load and limits. On the other hand, the *performance* policy tries to maximize performance by selecting the highest available p-states. Figure 3-20 shows the comparison *of performance, powersave and manual governors* in terms of power consumption, which is monitored through the *powerstat*[45] tool, and throughput according to the offered load.

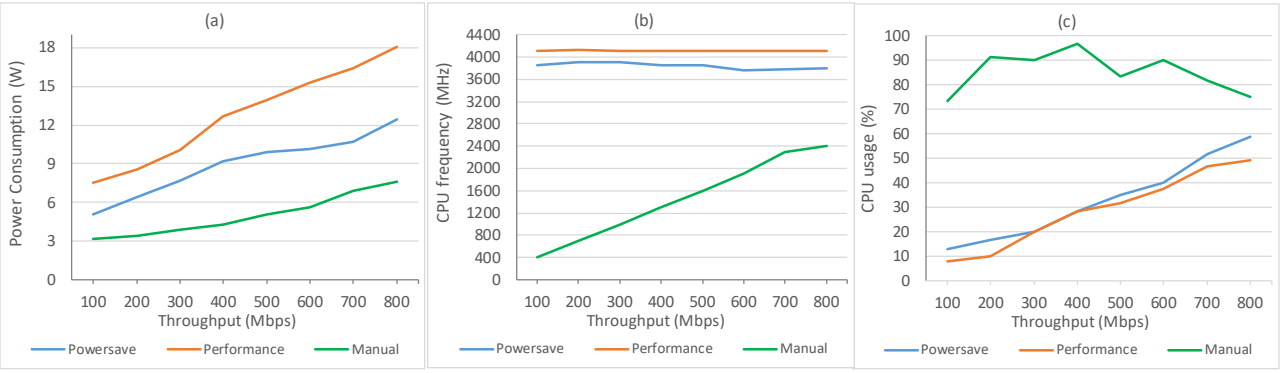


Figure 3-20 Impact of the UPF governor on (a) power consumption, (b) CPU frequency and (c) CPU usage

[42] https://open5gs.org/
[43] https://github.com/aligungr/UERANSIM
[44] https://wiki.archlinux.org/title/CPU_frequency_scaling
[45] https://manpages.ubuntu.com/manpages/focal/man8/powerstat.8.html

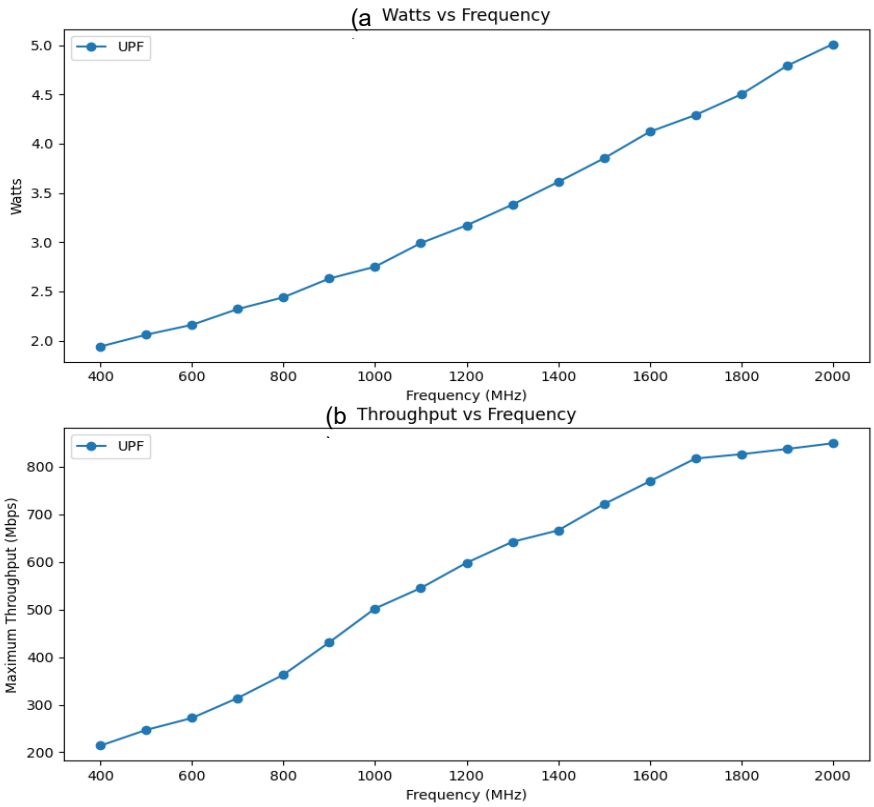


Figure 3-21 UPF characterisation: (a) Power consumption vs CPU Frequency, (b) Maximum achievable throughput vs CPU Frequency

The *manual* governor implies setting the minimum and maximum frequency limits of the governors to the same specific frequency; this way, we could achieve the minimum power consumption without impacting the throughput. Results show that both power save and performance governors tend to configure high CPU frequencies, leading to higher power consumptions. On the contrary, by setting manually lower frequencies, we were able to significantly reduce the power consumption needed to serve the throughput. Therefore, we can conclude that the default governors of the *intel_pstate* driver are not able to provide proper energy efficiency when hosting UPF NFs.

Next, using the *manual* governor, which selected the minimum frequency being able to serve the offered traffic, we characterized the relationship between fixed CPU frequency, power consumption and maximum achievable throughput. As illustrated in Figure 3-21, this relationship is linear until reaching the maximum achievable throughput (1 Gbps Ethernet link between UPF and gNB). As expected, higher CPU frequencies lead to higher throughputs but also to higher power consumption.

Figure 3-22 illustrates how, under fixed load conditions, different CPU frequencies behave in terms of CPU and power consumption. According to the results, we can conclude that to optimize energy efficiency the best strategy is to select the lowest possible frequency being able to serve the traffic load even if it leads to the highest CPU consumption.

Once verified the need of vertical scaling, next step was to evaluate horizontal scaling. Therefore, we evaluated how the number of UPF in the server affects throughput and energy consumption. The UPFs were pinned to an isolated CPU. During the evaluation we found that the modifying the frequency of a specific CPU through the *cpupower* tool (i.e., the manual using the manual *governor*), leaded to the modification of all the CPUs of the server. This drawback will impact the initial design of the proposed optimisation, as will be discussed in Section 3.6.3.

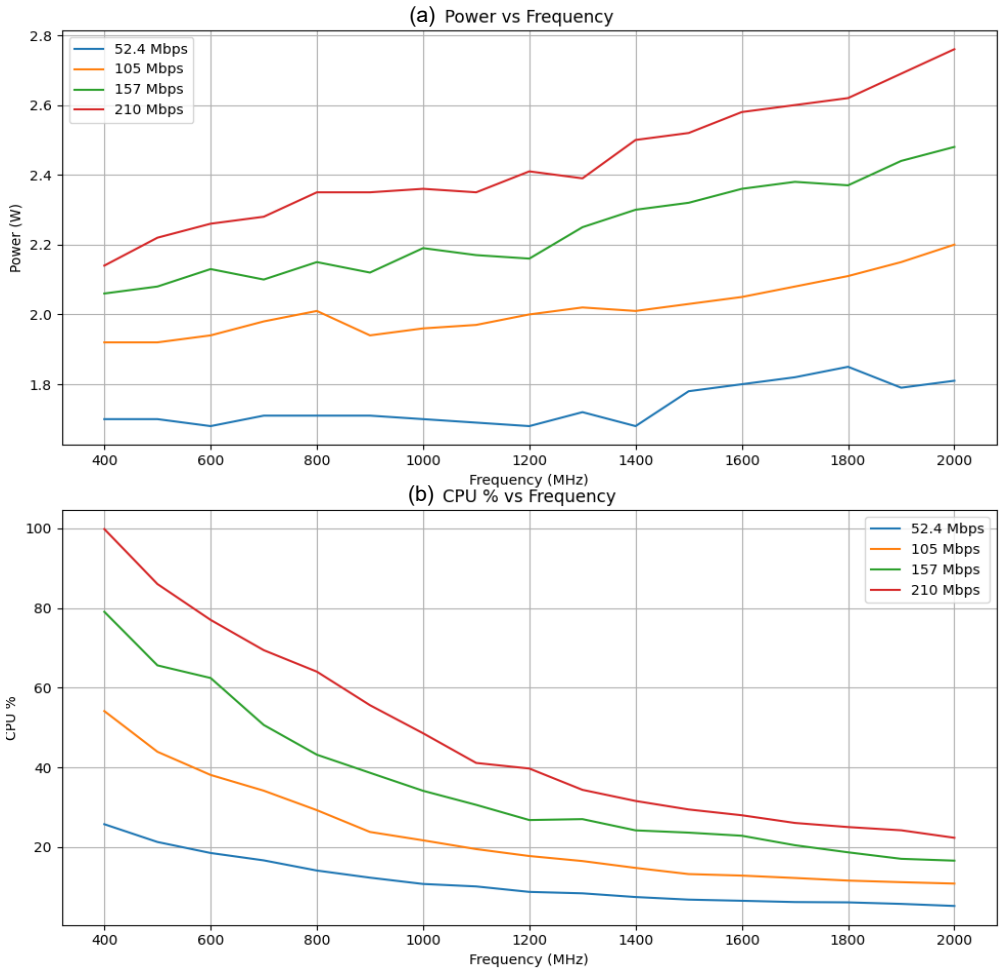


Figure 3-22 UPF characterisation; relationship between CPU frequency and (a) Power consumption and (b) CPU consumption, under fixed throughput conditions

Figure 3-23 shows that, at a given frequency, increasing the number of UPFs leads to a higher throughput until we reach the saturation point. Note that, as aforementioned, all UPF CPUs were set to the same frequency and the throughput was distributed equally among all UPFs.

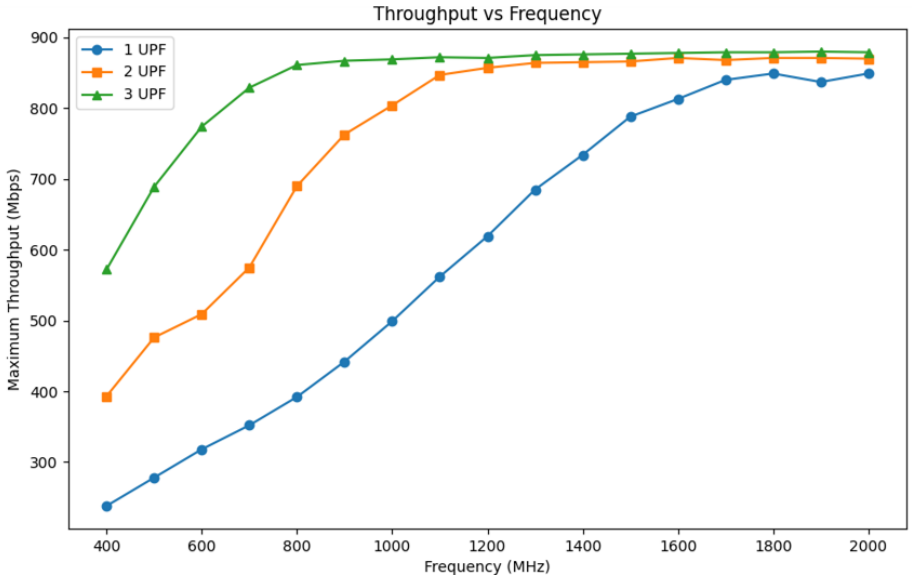


Figure 3-23 Characterisation of multiple UPFs: maximum throughput vs UPF CPU frequency

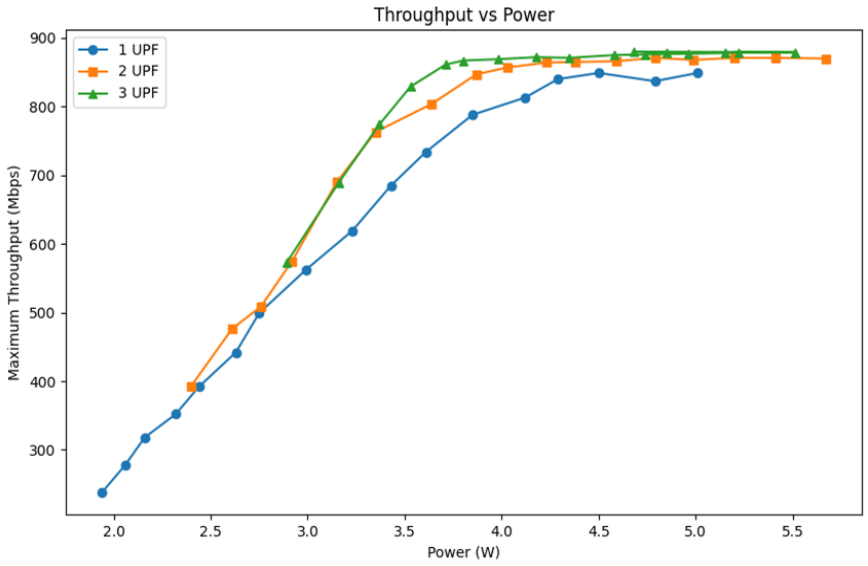


Figure 3-24 Characterisation of multiple UPFs: maximum throughput vs power consumption

Figure 3-24 shows the power consumption obtained during the evaluation. Note that, with an increasing throughput, it starts to be more efficient to have several UPFs using lower CPU frequencies than less UPF instances with higher CPU frequencies. For instance, in order to reach 800 Mbps, (i) using 1 UPF we need to set the CPU at 1.7GHz consuming 4.29W, (ii) using 2 UPFs we need 1GHz per CPU consuming 3.64W and, (iii) using 3 UPFs we can lower the frequency until 700MHz per CPU leading to 3.53W of power consumption.

According to these results, we can conclude that horizontal scaling based on increasing the number of UPFs will also increase the energy efficiency in the system. However, since, as aforementioned it seems that the *manual* governor forces that all the CPUs use the same fixed frequency, to efficiently exploit horizontal scaling we would need that all UPFs are equally loaded. On the contrary, the CPU with the highest load will determine the CPU frequency, thus impacting the energy efficiency of the other CPUs. To avoid this, we will require of Session Management Function (SMF) policies applying load-balancing among UPFs. Additionally, we could consider dynamically pinning UPFs to the same CPU in cases where the sum of CPU loads doesn't achieve the 100%, stopping unused CPUs. We will analyse these scenarios and options in future deliverables.

The next section presents the initial design of the proposed optimisation according to the previous evaluations.

3.6.3 Initial design

According to the findings and conclusions described in the previous section, Figure 3-25 depicts the envisioned architecture of the BeGREEN UPF CPU control optimisation and its relationship with the AI Engine and the Edge Server hosting the UPF(s).

The implementation of the solution is supported by three main components: the UPF Load Predictor (ULP) Assist rApp, the UPF CPU Control rApp, and the UPF CPU Control Agent. These components are described as follows:

- UPF CPU Control Agent: This Agent will work similarly to an xApp in the Edge server, exposing control policies to the UPF CPU Control. These policies will be based on available energy saving modes linked to the amount of supported throughput. According to the selected policy (i.e. the selected energy saving mode), and the number of UPFs and CPUs, the Agent will intelligently manage the Edge Server resources to maximize energy efficiency. For instance, it will set the required CPU frequency, pin UPFs to specific CPUs or stop idle CPUs. It also will provide feedback about the policies to the rApp and monitoring data to the datalake.

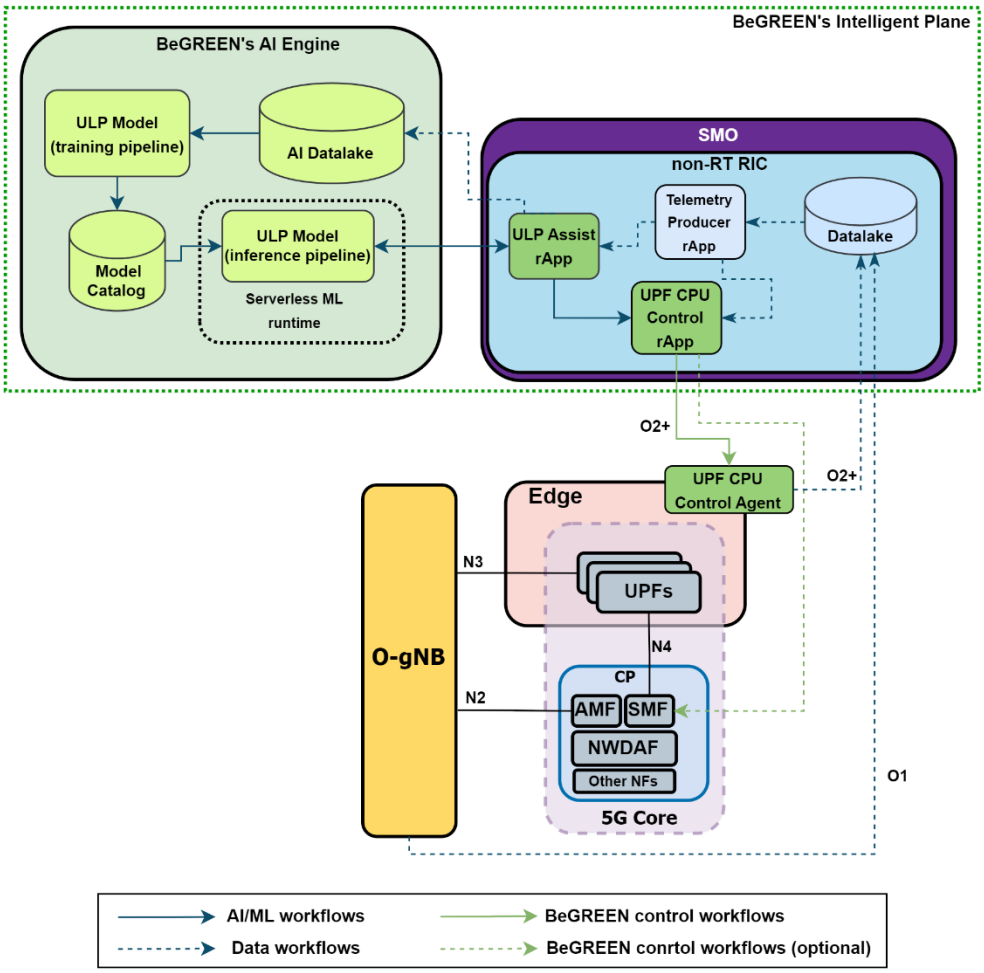


Figure 3-25 UPF CPU control; primary architecture

- UPF CPU Control rApp: This rApp will create policies to manage the CPU of the UPFs hosted in the Edge Server according to a defined set of inputs such as the UPF load predictions, actual telemetry from the RAN/UPF/Edge and other initialisation parameters (e.g., algorithm thresholds or aggressiveness level, SLA-related policies, etc.). In addition, it will discover Edge Server configuration parameters and available policies through the UPF CPU Control Agent (e.g., server characteristics, energy saving modes, number of UPFs…). The final definition of inputs will be presented in future deliverables. Finally, according to its internal logic, it will send the control policies to the UPF CPU Control Agent. Optionally, it may also send load-balancing policies to the SMF to control the offered load to the UPFs.
- ULP Assist rApp: This rApp will work producing and exposing UPF load predictions to the rApps subscribed to these data. According to the demands of the UPF CPU Control rApp, it will trigger the inference of the associated ULP ML Model through the AI Engine, sending the needed data for the inference (e.g., RAN, UPF and Edge Server telemetry) and forwarding the result of the inference to the control rApp. The ULP ML Model, which will be trained through the AI Engine, will be based on time series forecasting and/or regression algorithms as introduced in Section 3.3 regarding the RU Load Predictors.

According to the previous definition of the rApps and Agents, Figure 3-26 shows the contemplated workflow to implement the UPF CPU control loop. The interaction between the rApps will be managed by OSC's ICS component as they will work as data producer and consumers, following R1 interface principles.

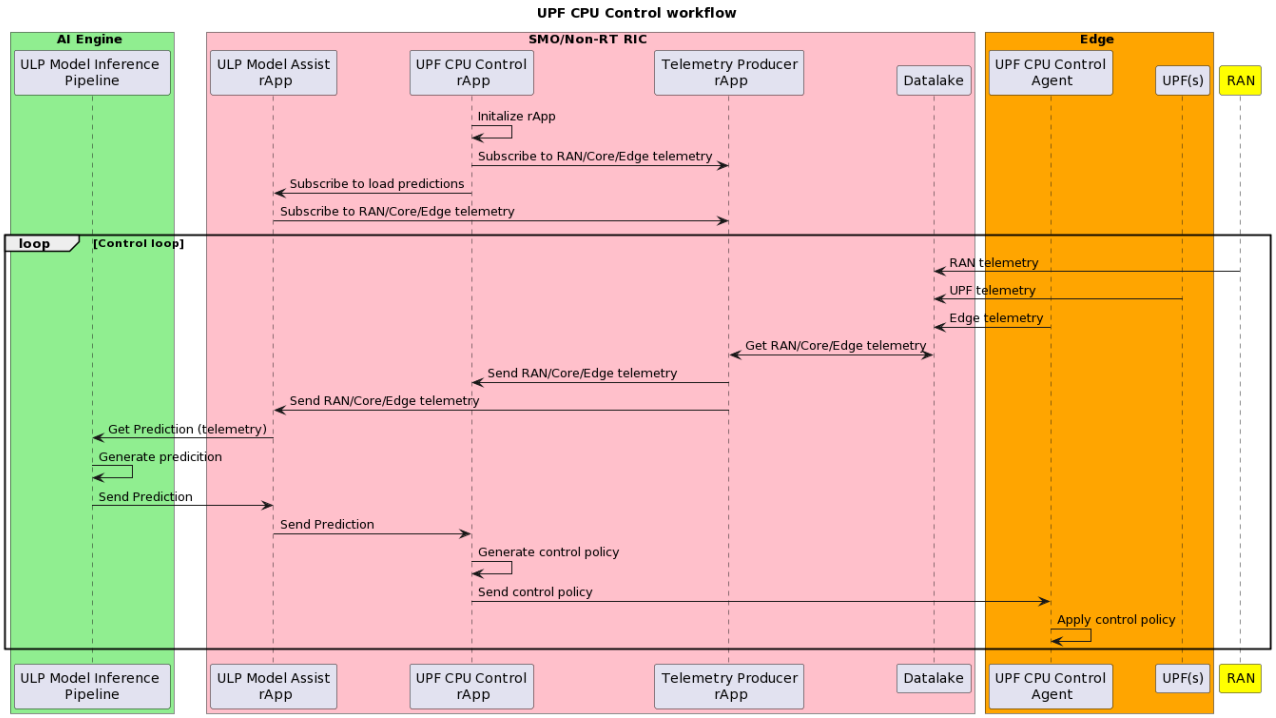


Figure 3-26 UPF CPU control; primary design of the required workflows

Additionally, it is considered to extend O2 interface functionalities to include the interactions with the UPF CPU Control Agent in the Edge server (O2+ interface in Figure 3-25).

3.7 Joint orchestration of vRANs and Edge AI services

Forthcoming generations of mobile networks need to accommodate intelligent services at the edge. These services, exemplified by MVA, necessitate real-time data gathering, transmission, and processing to facilitate various applications such as augmented reality, cognitive assistance, and surveillance. MVA, in particular, entails the transmission of video frames to the network, where they are processed, and the results, including accurately identified objects, are returned.

The widespread adoption of such services mandates a fundamental reconfiguration of mobile network management. In this context, the network's role extends beyond data transfer and processing; it must now directly optimize service performance. This optimisation revolves around key criteria, notably accuracy, low end-to-end latency, and high task throughput, all achieved in a resource-efficient manner. This shift is of paramount importance due to the substantial data volumes, computational intensity, and energy consumption associated with these services.

3.7.1 State-of-the-art

Recent research has explored deploying AI services, particularly Multi-Access Edge Computing (MEC) and Mobile Edge Computing services. Some of these studies aim to enhance performance by employing various techniques such as variable encoding, caching, visual tracking, and adaptive compression [126][127][128][129]. They also investigate the trade-off between accuracy and latency in these AI services [130][131][132]. Other works focus on resource management in MEC systems. For instance, [133] and [134] employ real-time greedy algorithms to find resource-efficient configurations. Meanwhile, [135] and [136] address resource allocation, image compression, video quality, and neural network model selection. It's worth noting that prior research, such as [137][138][139], primarily considers energy consumption at the user device.

Several studies have investigated network orchestration in mobile networks. Some, like [140], utilize predetermined models to select MCS and airtime to maximize throughput. Other approaches, such as model-free methods in [141][142], and [143], address slicing, throughput forecasting, and energy cost reduction but face challenges related to the availability of accurate training data. RL has been employed in tasks like interference coordination, network diagnostics, and SDN control optimisation [144][145][146]. However, RL lacks convergence guarantees. Notably, none of these works consider the interplay between the mobile network and edge servers. Our experimental analysis in the next section indicates that these elements' joint orchestration is vital.

3.7.2 Design principles

This section presents a set of network policies and performance indicators involved in this problem. Moreover, we discuss a comprehensive set of experiments conducted using an experimental platform, which includes a fully-fledged BS, a UE that generates service requests through the BS to a widely recognized object recognition service, and a standard server equipped with an NVIDIA GPU (Graphics Processing Unit) to run the service.

In these experiments, each service request comprises an image containing a varying number of objects sourced from the COCO dataset [147]. The images are transmitted to the service using the uplink channel of the LTE interface. Subsequently, the service processes the images and provides the user with bounding boxes and classification labels for each object detected in the image. This information is transmitted to the user through the downlink channel of the LTE interface.

3.7.2.1 Performance indicators and policies

The *performance indicators* refer to a specific metric or parameter used to measure and assess the quality and efficiency of various aspects of the system. In our case, we define the following ones:

- *Service delay:* End-to-end delay that includes the image pre-processing at the user side, its transmission, the processing at the server (GPU delay), and the return of the bounding boxes and labels.
- *Mean Average Precision (mAP):* The service accuracy is quantified using the Mean Average Precision (mAP) [148]. On the one hand, precision is defined as the ratio of true positives over all positive classifications. On the other hand, the recall measures how well these positives are identified by calculating the ratio between true positives over the sum of true positives and false negatives. The Intersection over Union (IoU) measures the overlap between the calculated bounding box and the ground truth. IoU values above a threshold (set to 0.5 here) trigger a true positive. Then, for a given set of images, the Average Precision (AP) corresponds to the area below the precision-recall curve. Finally, mAP is the mean AP over all object categories, ranging from 0 (worst performance) to 1 (best performance).
- *Server power consumption:* Power cost associated with the computational load of the service's requests, which is dominated by the GPU power consumption.
- *Base Station power consumption:* Power consumption associated with processing the baseband unit in a virtualized RAN environment.

The network policies refer to a set of rules, or configurations that govern the behaviour and operation of the network elements within the system. We define the following policies:

- *Image resolution:* This policy sets the average encoding of every image (number of pixels) which the service can enforce. In our experiments, the maximum (100%) resolution is 640 x 480 pixels. Note that, at any given time, the resolution of an image may be larger or smaller than the policy. However,

it is crucial that the average resolution over the entire decision period for all users satisfies to the threshold.

- *Radio Airtime:* This radio policy imposes a constraint on the radio resources (duty cycle) the vBS allocates to the service traffic. The MAC layer radio scheduler, which operates at msec granularity must allocate radio resources (which may vary across users based on their channels) such that the threshold set by the policy is respected. Due to the nature of this service, we focus on uplink communication.
- *GPU speed:* The server's policy is a GPU power limit that adapts the processing speed of a GPU (or a pool of GPUs) in a slice to meet the adopted power constraint. The GPU controller (e.g., NVIDIA driver) may change the GPU speed at any given time (e.g., for different video frames) as long as the GPU power set by this policy is respected.
- *Radio MCS:* This policy imposes a constraint on the maximum MCS eligible by the vBS to transport the service's data over the air. We note that the MCS selected by the MAC layer may be lower than this bound for some users depending on their channel state.

3.7.2.2 Experimental analysis

The impact of the network policies on the different performance indicators is of paramount importance when designing a control mechanism. In the following, we provide some experimental results and extract insights that we use later in our initial design.

Figure 3-27 depicts the service delay vs the server power consumption, for different airtime radio policies and image resolutions. Note that higher resolution images increase service delay due to the longer transmission time of requests. We also observe that this occurs irrespective of the radio policy configuration. However, the radio policy has an important impact on delay as well. This is expected since lower airtime implies lower usage of radio resources, which further increases the transmission time of the requests at the radio interface. Specifically, our experiments show that an 80% increase in the airtime improves the delay between 65% and 80%. Concerning the server's power consumption, lower-res images and lower radio resource allocations increase this cost. Specifically, there is a 56% increase in power consumption for an 80% increase in radio time resource; a similar increase is attained when there is a 75% increase in image resolution. This is due to the fact that increasing the radio resources allows the user to send a higher rate of requests in a similar way than low-res images do, which ultimately increases the workload assigned to the service's resources (the GPU, in this case).

Figure 3-28 shows the power consumption measured at the baseband unit of the vBS for various airtime and MCS policies and image resolutions. We first observe that lower-resolution images consume less radio resources and hence less vBS power. Second, using larger radio resources (airtime) induces higher power costs because it allows the user to transmit images at a higher rate. Finally, and perhaps surprisingly, higher MCS policies cause lower BS power consumption. The reason is the data load at the BS is relatively low compared to the bandwidth available at the vBS, e.g., higher-res images with 100% airtime generate up to 2.8~Mb/s, compared to a capacity of around 50Mb/s (SISO LTE @ 20MHz bandwidth). In this scenario, even though LTE subframes modulated with higher MCS incur higher instantaneous power consumption, they process the load faster, which pays off in terms of long-term power consumption.

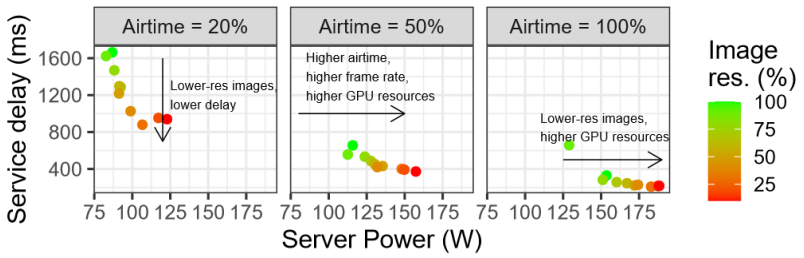


Figure 3-27 Service delay vs. server's power consumption for images with different resolutions and radio policies

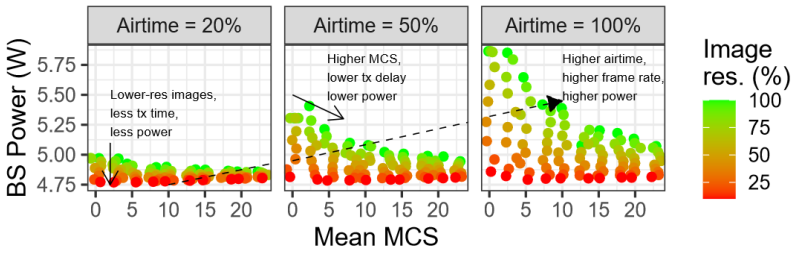


Figure 3-28 BS power consumption vs. radio policies for images with different resolutions

3.7.3 Problem formulation

Let us formally define the elements needed to formulate the online learning problem in the next section. These elements are the contexts, actions, and performance indicators.

Contexts. We define the context at each time period t as $c_t := [n_t, \bar{c}_t, \tilde{c}_t] \in C$, where n_t is the number of users in the slice, and $\bar{c}_t$ and $\tilde{c}_t$ are the mean and the variance of the UL CQI across all users in the slice during the previous period, and C is the context space.

Control policies. Let H denote the set of possible image resolutions; A the set of possible airtime configurations (uplink radio resources) that can be assigned; Γ the possible GPU speed configurations; and M the set of all possible MCS policies (characterizing the data rates) as defined above. Hence, we let:

$$\boldsymbol{x_t} \coloneqq [\boldsymbol{\eta_t}, \boldsymbol{a_t}, \boldsymbol{\gamma_t}, \boldsymbol{m_t}] \in \boldsymbol{X} \coloneqq \boldsymbol{H} \times \boldsymbol{A} \times \boldsymbol{\Gamma} \times \boldsymbol{M} \tag{3-16}$$

denote the control policy selected at time period t. The GPU speed is configured in the same machine where the learning agent runs, the airtime and the MCS policies can be sent to the vBS through the A1-P interface of O-RAN architecture [9], and the image resolution is indicated to the user using the application of the service.

We focus on uplink radio policies because, as our experiments confirm, such AI services have little impact on the downlink as the data surge usually goes upstream with only simple information (bounding boxes, labels) flowing downstream. Note that the policies in X jointly control parameters from the user device, the vBS, and the edge server. These three elements are highly coupled (as shown in the previous section) and for that reason should be configured jointly.

Performance indicators. Similarly, our performance indicators were introduced in the previous section. The service delay experienced by user i is denoted by $D_i(c, x)$, and the mAP is denoted by $Q_i(c, x)$. We then define:

$$\boldsymbol{d}(\boldsymbol{c}, \boldsymbol{x}) \coloneqq \boldsymbol{max_i}\, \boldsymbol{D_i}(\boldsymbol{c}, \boldsymbol{x}), \quad \boldsymbol{and} \qquad \boldsymbol{\rho}(\boldsymbol{c}, \boldsymbol{x}) \coloneqq \boldsymbol{min_i} \boldsymbol{Q_i}(\boldsymbol{c}, \boldsymbol{x}), \tag{3-17}$$

to be the highest delay and lowest mAP, respectively, across all users. The consumed power at the edge server is denoted by $p^s(c, x)$, and the consumed power at the vBS is denoted by $p^b(c, x)$. Henceforth, we denote by $d_t(c_t, x_t)$, $\rho_t(c_t, x_t)$, $p_t^s(c_t, x_t)$, and $p_t^b(c_t, x_t)$ the noisy observations of our performance

indicators at time period t. Feedback from the data plane components including all these performance metrics is received at the end of each time period t.

Our goal is to minimize the power consumption of the whole system (vBS and edge server) subject to the performance constraints of the service. Depending on the form factor of the vBS and the server's configuration (i.e., GPU model, motherboard, etc.) the consumed energy of each entity may have a different order of magnitude. Moreover, the cost associated with energy consumption may vary depending on the scenario. In regular small cell based scenarios, such cost may be related to the price of electricity, which may vary between day and night depending on the rates set by the power suppliers in each country. In other scenarios, such as those based on Power over Ethernet (PoE) or a solar-powered vBS, this cost may reflect the scarcity of the energy resource for the RAN.

In order to capture these different scenarios, we define the following *cost function*:

$$\boldsymbol{u(c,x) = \delta_1 p^{s(c,x)} + \delta_2 p^{b(c,x)}} \tag{3-18}$$

where δ_1 and δ_2 are the costs of the power at the edge server and the vBS, respectively, in monetary units per watt (mu/W).

On the other hand, we consider performance constraints at the service level, going a step beyond other works considering lower-level performance requirements (e.g., [78]) such as data rate or delay.

The mapping between context-action pairs and the service-level performance indicators is very complex and there are no available models, as we detailed in the experimental results of the previous section. For that reason, we learn them from observations. For our object recognition service, we consider two constraints: *(i)* a maximum service delay denoted by d^{max}, which is directly related to the frame rate (number of images per second) that the user is going to process, and *(ii)* a minimum mAP denoted by ρ^{min} which indicates a lower bound on how accurate is the service in detecting the objects. We formulate the problem as follows:

$$\begin{aligned} \min_{\{x_t\}_{t=1}^{T} \in \mathcal{X}} \quad & \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} u_t(c_t, x_t) \\ \text{s.t.} \quad & d_t(c_t, x_t) \le d^{max}, \qquad \forall t \le T \\ & \rho_t(x_t) \ge \rho^{min}, \qquad \forall t \le T. \end{aligned} \tag{3-19}$$

Note that the service constraints are satisfied for the user experiencing the worst service as $d(c,x) := max_i D_i(c,x)$ and $\rho(c,x) := min_i Q_i(c,x)$. It is worth mentioning that these functions are unknown beforehand and platform dependent. That is, the values of performance indicators may change with the software implementing the vBS, the hardware of the server and vBS, and the service running at the edge server. In the experimental analysis of the previous section, we show that the trade-offs between the control policies and the performance indicators for one user are non-trivial and non-linear. With a larger number of users, these relations can be even more complex and exhibit different behaviour. For that reason, it becomes essential the use of AI/ML to learn how to configure the system based on the observations (data-driven). Moreover, there is no unique optimal configuration, as it depends on the context. This renders a very challenging problem as the set of contexts is not finite and changes over time.

3.7.4 Initial design

In this section, we provide an initial design of the learning algorithm to solve the problem defined in the previous section. Note that this problem is formulated as a contextual bandit problem, which is a particularisation of the well-known RL formulation with several differences:

First, RL formulation considers that the probability distribution of the state/context at $t+1$ depends on the state/context and the selected action at t. Conversely, the contextual bandit formulation considers that the

context does not depend on the selected actions. Note that the contexts in our problem (number of users and channel quality) cannot be affected by the configuration of the system.

Second, RL formulation considers that reward can be sparse or delayed, i.e., the selected actions will affect the reward an arbitrary number of time steps later. In some cases, the reward is only revealed at the end of the episode and the algorithm should distribute the credit of this outcome across all the actions in the episode. In contrast, the contextual bandit formulation considers that the reward is instantaneous, i.e., for a given state-action pair at time t, we can observe the associated reward instantaneously. Furthermore, the decisions made at time t will not have an impact on future time steps. This property holds in our problem due to the timescale of the decision-making process. Recall that our solution operates in the Non-Real Time RIC of the O-RAN architecture (timescale of seconds). Based on this time scale, the performance observation at each time t is independent of previous time steps as the system becomes stationary.

Finally, it is worth mentioning that the contextual bandit approach does not need to estimate the accumulated rewards until the end of the episode as it is needed in RL (using Temporal-Difference Learning, Monte Carlo Methods, etc.). These simplifications in the formulation make our solution simpler and more effective.

The contextual bandit $\pi_t(c_t; B_{t-1})$: $C \rightarrow X$ is an algorithm that maps contexts to actions, where $B_{t-1} = [c_1, x_1, u_1, d_1, \rho_1, \dots, c_{t-1}, x_{t-1}, u_{t-1}, d_{t-1}, \rho_{t-1}]$ is the set of historical observations at $t-1$ (i.e., triples of contexts, actions, and performance metrics).

At each time period t, a context c_t is observed and an action $x_t = \pi_t(c_t; B_{t-1})$ is computed and applied to the system. At the end of the time period t the performance indicators associated with the pair (c_t, x_t) are measured to update B_t and consequently π_t. The contextual bandit π_t is sequentially updated with each new measurement from the system towards solving the problem in detailed in the previous section.

4 Summary and Conclusions

This deliverable has presented a SotA review and an initial design of the BeGREEN intelligent plane and the proposed AI/ML-assisted procedures to enhance energy efficiency in the RAN infrastructure.

Chapter 2 presented a comprehensive description of the BeGREEN Intelligent Plane, building upon the work established in Deliverable D2.1. First, we realized an in-depth analysis of the SotA in RICs and their integration with AI/ML approaches. Starting with the specification from the O-RAN Alliance and its reference implementation developed by the O-RAN Software Community, we also reviewed the developments of other relevant implementations of RICs such as FlexRIC from OpenAirInterface and SD-RAN from the ONF. Then, we introduced the commercial RIC solutions from the main vendors of the O-RAN ecosystem, highlighting the characteristics of the solution being considered in BeGREEN, the Accelleran dRAX. We also examined the features of the main simulators and emulators being used for research and validation purposes, focusing again on the solutions to be exploited in BeGREEN, the AIMM simulator and TeraVM. Finally, we explored relevant projects from the SotA, mainly European Projects with participation of BeGREEN partners, focusing on their achievements related to AI and Intelligent Controllers.

The second part of Chapter 2 was devoted to the description of the Intelligent Plane architecture and its main components, namely the AI Engine, the SMO and non-RT RIC, and the near-RT RIC. Regarding the AI Engine, we highlighted its main components and functionalities, and how we plan to integrate it with the rest of BeGREEN architecture. We also introduced the MLOps, Serverless and datalake frameworks being considered to realize the implementation of BeGREEN AI Engine, which will be detailed in future WP4 deliverables. Regarding the RICs, on the one hand, the non-RT RIC will be leverage some of the functions of OSC's implementation, such as de ICS or the A1 interface controller, while additional functions and interfaces will be developed to realize the integration with the AI Engine and the different BeGREEN optimisation targets (e.g., the 5GC). On the other hand, the near-RT RIC will be based on Accelleran's dRAX solution, developing or enhancing built-in brokers to integrate communication with the AI Engine, the non-RT RIC and the O-Cloud according to BeGREEN requirements. Additionally, the Telemetry Gateway will be exploited to facilitate the exposure of RAN metric to external components. Finally, additionally to the components strictly related to the Intelligent Plane implementation, we described other BeGREEN components related to the targeted energy-efficiency optimisations, focusing on its integration with the Intelligent Plane through O-RAN compliant and/or new or extended functions and interfaces. To this end, we presented a comprehensive exploration of O-Cloud, RU, RIS and Relay functions, related to RAN control and monitoring, and of 5GC and Edge functions, related to the joint orchestration and optimisation of RAN and non-RAN resources.

Chapter 3 has presented the SotA analysis, and an initial design of the AI/ML-based solutions proposed in BeGREEN for improving the energy efficiency of the RAN infrastructure. A first stage of this work was already presented in chapter 6 in BeGREEN D2.1. On the one hand, the use of eXplainable AI (XAI) techniques has been proposed to identify entities and areas of the network where energy savings are achievable and, consequently, improve energy efficiency. The proposed methodology is based on a regression algorithm to identify the variables that contribute most to the energy efficiency and predict the energy efficiency of a given entity. On the other hand, an AI/ML solution to dynamically dimension and allocate the computing resources needed for each vBS in an O-RAN O-Cloud Computing Platform is presented. In particular, RL techniques based on a classical DQN approach augmented with a Relation Network mechanism are used to smartly allocate the computing resources, leading to an improved network performance and a reduction in energy consumption. Moreover, several AI/ML solutions have been proposed for an intelligent control of the RU, RIS and Relays. Concerning the RU control, an AI/ML solution is presented to steer traffic among cells and switch on/off RU/cells according to RU load predictions based on time series predictions and regression algorithms. Concerning the RIS control, the proposed solution is based on ISAC technology to track UE location and RL techniques to forecast UE mobility to adequately manage the RUs and the RIS configuration

parameters. Regarding the relay control, a DQN methodology is proposed to activate/deactivate fixed relays or Relay UE (RUEs) to improve the network performance and reduce energy consumption. Additionally, an AI/ML solution to enhance the energy efficiency of edge servers hosting UPF NFs has been proposed. This solution manages properly the CPU frequency of the edge server according to traffic load status and predictions based on time series forecasting and regression algorithms. Finally, a solution for a joint orchestration of vRANs and Edge AI services is presented with the aim of minimising the power consumption of the vBSs and the edge server subject to several performance service constraints.

An initial implementation and evaluation of the BeGREEN Intelligent Plane and the proposed AI/ML solutions to enhance the energy efficiency in the RAN will be presented in BeGREEN D4.2.

5 Bibliography

[1] BeGREEN, D2.1, "BeGREEN Reference Architecture", July 2023. Available: https://www.sns-begreen.com/deliverables?id=971369

[2] M. Polese, L. Bonati, S. D'Oro, S. Basagni and T. Melodia, "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," in IEEE Communications Surveys & Tutorials, Vol. 25, No. 2, pp. 1376-1411, 2023. DOI: 10.1109/COMST.2023.3239220.

[3] O-RAN Alliance WG2, "O-RAN non-RT RIC Architecture 4.0", O-RAN.WG2.non-RT-RIC-ARCH-R003-v04.00, October 2023.

[4] O-RAN Alliance WG2, "O-RAN R1 interface: General Aspects and Principles 6.0", O-RAN.WG2.R1GAP-v06.00, October 2023.

[5] O-RAN Alliance WG2, "O-RAN A1 interface: General Aspects and Principles 3.01", O-RAN.WG2.A1GAP-R003-v03.01, March 2023.

[6] O-RAN Alliance WG3, "O-RAN Near-RT RIC Architecture 5.0", O-RAN.WG3.RICARCH-R003-v05.00, October 2023.

[7] O-RAN Alliance Architecture Overview. [Online] Available: https://docs.o-ran-sc.org/en/latest/architecture/architecture.html

[8] O-RAN Alliance WG3, Near-Real-time RAN Intelligent Controller and E2 Interface Workgroup - Near-RT RIC Architecture, O-RAN.WG3.RICARCH-R003-v04.00. 2023-09-29.

[9] O-RAN Alliance WG1, Use Cases and Overall Architecture, O-RAN Architecture Description. O-RAN.WG1.OAD-R003-v09.00. 2023-09-29.

[10] Tech play on, 'What Is E2 Interface in Open RAN?'. Accessed 2023-10-12. [Online] Available: https://www.techplayon.com/what-is-e2-interface-in-open-ran/

[11] O-RAN Alliance WG2, "O-RAN AI/ML workflow description and requirements 1.03", O-RAN.WG2.AIML-v01.03, October 2021.

[12] OSC Community Lab. https://wiki.o-ran-sc.org/display/IAT/OSC+Community+Labs

[13] A. Lacava, et al., "Programmable and Customized Intelligence for Traffic Steering in 5G Networks Using Open RAN Architectures" in arXiv:2209.14171, October 2022.

[14] R. Schmidt, M. Irazabal, Navid Nikaein, "FlexRIC: an SDK for next-generation SD-RANs," Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies (CoNEXT '21), 2021. DOI: https://doi.org/10.1145/3485983.3494870

[15] Accelleran, dRAX: Cloud-Native Open RAN software, September 2023. [Online] Available: https://accelleran.com/drax/

[16] Accelleran, Acceleran dRAX-2022.3.1 release, September 2023. [Online] Available: https://accelleran.github.io/drax-docs/.

[17] AIMM Simulator. [Online] Available: https://aimm-simulator.readthedocs.io/en/latest/

[18] VIAVI Solutions inc., What is TeraVM?, 2021. [Online] Available: https://www.viavisolutions.com/en-us/literature/teravm-overview-what-teravm-data-sheets-en.pdf

[19] VIAVI Solution inc., TeraVM RIC Test, 2023. [Online] Available: https://www.viavisolutions.com/en-uk/products/teravm-ric-test

[20] J. A. Ayala-Romero, A. Garcia-Saavedra, X. Costa-Perez, and G. Iosifidis, "Bayesian online learning for

energy-aware resource orchestration in virtualized rans," in IEEE INFOCOM 2021-IEEE Conference on Computer Communications, 2021.

[21] A. Cañete, K. Djemame, M. Amor, L. Fuentes, and A. Aljulayfi, "A proactive energy-aware auto-scaling solution for edge-based infrastructures," in 2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC), 2022.

[22] M. Kalntis and G. Iosifidis, "Energy-Aware Scheduling of Virtualized Base Stations in O-RAN with Online Learning," in GLOBECOM 2022-2022 IEEE Global Communications Conference, 2022.

[23] AI@EDGE D2.3, "Consolidated system architecture, interfaces specifications, and techno-economic analysis," March 2023.

[24] O-RAN Alliance, "O-RAN Network Energy Saving Use Cases Technical Report 2.0", June 2023

[25] "European Vision for the 6G Ecosystem," White Paper, July 2021. [Online]. Available: https://5g-ppp.eu/

[26] D. Kreuzberger, N. Kühl, S. Hirschl, "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," IEEE Access, Vol. 11, pp. 31866-31879, 2023. DOI: 10.1109/ACCESS.2023.3262138.

[27] V. Yussupova, et al., "FaaSten Your Decisions: Classification Framework and Technology Review of Function-as-a-Service Platforms," April 2020. [Online] Available: https://arxiv.org/pdf/2004.00969.pdf

[28] 3GPP Rel-15, "Management and orchestration; 5G end to end Key Performance Indicators (KPI)," Technical Specification TS_28.554, 2018.

[29] ITU, "Energy efficiency metrics and measurement methods for telecommunication equipment," Document L.1310, 2020. [Online] Available: https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-L.1310-202009-I!!PDF-E&type=items

[30] ETSI, "Environmental Engineering (EE); Study on methods and metrics to evaluate energy efficiency for future 5G systems", ETSI TR 103 542 V1.1.1, 2018-06. [Online] Available: https://www.etsi.org/deliver/etsi_tr/103500_103599/103542/01.01.01_60/tr_103542v010101p.pdf

[31] K. M. S. Huq, S. Mumtaz, J. Rodriguez, R. L. Aguiar, "Comparison of energy-efficiency in bits per joule on different downlink CoMP techniques," IEEE ICC, 2012. doi: 10.1109/ICC.2012.6364829

[32] E. Björnson and E. G. Larsson, "How Energy-Efficient Can a Wireless Communication System Become?," 2018 52nd Asilomar Conference on Signals, Systems, and Computers, 2018. doi: 10.1109/ACSSC.2018.8645227. [Online] Available: https://arxiv.org/pdf/1812.01688.pdf

[33] M. Usama, M. Erol-Kantarci, "A Survey on Recent Trends and Open Issues in Energy Efficiency of 5G," Sensors (Basel). 2019. doi: 10.3390/s19143126

[34] M. Alsharif, A. Kelechi, J. Kim, J. Kim, "Energy Efficiency and Coverage Trade-Off in 5G for Eco-Friendly and Sustainable Cellular Networks," *Symmetry*, Vol. 11, No. 3, 2019. doi: 10.3390/sym11030408. [Online]. Available: http://dx.doi.org/10.3390/sym11030408

[35] B. Pankajakshan, "A Pathway To Net Zero For Telecom Operators," [Online] Available: https://www.forbes.com/sites/forbestechcouncil/2023/01/04/a-pathway-to-net-zero-for-telecom-operators/

[36] M. Radovanović, Sustainable Energy Management, (Second Edition), 2023. [Online] Available: https://www.sciencedirect.com/book/9780128210864/sustainable-energy-management

[37] O-RAN Alliance, "Information Coordination Service". [Online] Available: https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric-plt-informationcoordinatorservice/en/latest/overview.html

[38] O-RAN Alliance, "O-RAN O2 Interface General Aspects and Principles 4.0", O-RAN.WG6.O2-GA&P-R003-v04.00, June 2023.

[39] O-RAN Alliance, "O-RAN non-RT RIC & A1 interface: Use Cases and Requirements 7.0", O-RAN.WG2.Use-Case-Requirements-v07.00, June 2023.

[40] O-RAN Alliance, "O-RAN Use Cases Analysis Report 11.0", O-RAN.WG1.Use-Cases-Analysis-Report-R003-v11.00, June 2023.

[41] BeGREEN, D3.1, "State-of-the-art on PHY mechanisms energy consumption and specification of efficiency enhancement solutions", December 2023. [Online] Available: https://www.sns-begreen.com/deliverables

[42] O-RAN Alliance, "O-RAN Management Plane Specification 13.0", O-RAN.WG4.MP.0-R003-v13.00, October 2023.

[43] O-RAN Alliance, "O-RAN Acceleration Abstraction Layer – General Aspects and Principles," O-RAN.WG6.AAL-GAnP-v03.00, 2022.

[44] Z. Ghadialy, "An Overview of O-RAN Architecture," Parallel Wireless. Accessed Nov. 17, 2023. [Online]. Available: https://www.parallelwireless.com/blog/an-overview-of-o-ran-architecture/

[45] https://rimedolabs.com/blog/o-ran-deployment-scenarios/

[46] O-RAN Alliance, "O-DU High Overview — o-du-l2 master documentation". Accessed: Nov. 17, 2023. [Online]. Available: https://docs.o-ran-sc.org/projects/o-ran-sc-o-du-l2/en/latest/overview.html#id1

[47] Minxiang (HackMD.io), 'O-DU and O-CU'. Accessed: Nov. 17, 2023. [Online]. Available: https://hackmd.io/@Min-xiang/Hyly1ER0_#7-O-DU-L1-Functional-Blocks

[48] BeGREEN, D5.1, "Use Case Identification and Demonstration Plan", December 2023. Available: https://www.sns-begreen.com/deliverables?id=971371

[49] RISE-6G D2.5, "RISE network architectures and deployment strategies analysis: first results," July 2022. [Online] Available: https://rise-6g.eu/Documents/LIVRABLES/RISE-6G_WP2_D2.5_FINAL.pdf

[50] RISE-6G D2.4, "Metrics and KPIs for RISE wireless systems analysis: final results," Feb 2022. [Online] Available: https://rise-6g.eu/Documents/LIVRABLES/RISE-6G_WP2_D2.4_FINAL.pdf

[51] 3GPP Rel-16, "Stage 2 functional specification of User Equipment (UE) positioning in NG-RAN", Technical Specification 38.305, v16.7.0, 2021-12.

[52] 3GPP Rel-16, "Radio Resource Control (RRC); Protocol Specification", Technical Specification 36331, v16.7, 2021-12.

[53] 3GPP Rel-16, "Radio measurement collection for Minimization of Drive Tests (MDT)", TS 37.320, v16.7.0, 2021-12.

[54] 3GPP Rel-17, "NR Physical Layer Measurements", TS 38.215, v.17.3.0, 2023-03.

[55] 3GPP Rel-18, "Management and orchestration; 5G Performance Measurements", TS 28.552, v.18.3, 2023-06.

[56] 3GPP Rel-17, "NR Physical Layer procedures for data", TS 38.214, v.17.6.0, 2023-06.

[57] 3GPP Rel-17, "NR Layer 2 Measurements", TS 38.314, v.17.3.0, 2023-06.

[58] A. Guryanov, "Efficient Computation of SHAP Values for Piecewise-Linear Decision Trees," 2021 International Conference on Information Technology and Nanotechnology (ITNT), 2021. doi: 10.1109/ITNT52450.2021.9649051.

[59] M. Masoudi, et al., "Green mobile networks for 5G and beyond," in IEEE Access, Vol. 7, 2019. doi: 10.1109/ACCESS.2019.2932777.

[60] https://wiki.o-ran-sc.org/pages/viewpage.action?pageId=20874400

[61] J. Armstrong, E. Fallon, "Dimensionality Reduction for Optimization of Radio Base Station Transmission Based on Energy Efficiency," IEEE ITMS, 2023. doi: 10.1109/ITMS59786.2023.10317744.

[62] M. Masoudi, et al., "Cost-effective migration toward virtualized C-RAN with scalable fronthaul design." IEEE Systems Journal 14.4, 2020.

[63] F. Murti, et al. "An optimal deployment framework for multi-cloud virtualized radio access networks." IEEE Transactions on Wireless Communications, 20.4, 2020.

[64] A. Tootoonchian, et al. "ResQ: Enabling SLOs in Network Function Virtualization." 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18). 2018.

[65] G. Garcia-Aviles, "Nuberu: Reliable RAN virtualization in shared platforms," Proceedings of the 27th Annual International Conference on Mobile Computing and Networking, 2021.

[66] X. Fan, et al., "Power provisioning for a warehouse-sized computer." ACM SIGARCH computer architecture news 35.2, 2007.

[67] C. Lefurgy, et al., "Server-level power control." IEEE Fourth International Conference on Autonomic Computing (ICAC'07), 2007.

[68] J. Khalid, et al., "Iron: Isolating network-based CPU in container environments." 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18), 2018.

[69] P. Kumar, et al. "PicNIC: predictable virtualized NIC." Proceedings of the ACM Special Interest Group on Data Communication, 2019.

[70] S. Grant, et al. "Smartnic performance isolation with fairnic: Programmable networking for the cloud." *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, 2020.

[71] E. Ebrahimi, et al. "Fairness via source throttling: A configurable and high-performance fairness substrate for multicore memory systems." *ACM Transactions on Computer Systems (TOCS),* 30.2, (2012).

[72] L. Subramanian, et al. "MISE: Providing performance predictability and improving fairness in shared main memory systems." *IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*, 2013.

[73] L. Subramanian, et al. "The application slowdown model: Quantifying and controlling the impact of inter-application interference at shared caches and main memory." *Proceedings of the 48th International Symposium on Microarchitecture*, 2015.

[74] J. Ding, et al. "Agora: Real-time massive MIMO baseband processing in software." *Proceedings of the 16th international conference on emerging networking experiments and technologies*, 2020.

[75] X. Foukas, R. Bozidar, "Concordia: Teaching the 5G vRAN to share compute." *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, 2021.

[76] Intel Corporation, "Improving real-time performance by utilizing cache allocation technology." April 2015.

[77] B. Gregg, Systems performance: enterprise and the cloud, Pearson Education, 2014.

[78] J. Ayala-Romero, et al. "vrAIn: A deep learning approach tailoring computing and radio resources in virtualized RANs." *The 25th Annual International Conference on Mobile Computing and Networking*, 2019.

[79] V. Mnih, et al. "Playing atari with deep reinforcement learning." arXiv preprint arXiv:1312.5602, 2013.

[80] D. Raposo, et al. "Discovering objects and their relations from entangled scene representations." arXiv preprint arXiv:1702.05068, 2017.

[81] R. Sutton, A. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[82] L. Larsen, H. Christiansen, S. Ruepp, M. Berger, "Toward Greener 5G and Beyond Radio Access Networks—A Survey," in IEEE Open Journal of the Communications Society, Vol. 4, 2023. doi: 10.1109/OJCOMS.2023.3257889.

[83] S. Puthenpura, et al, "SMaRT-5G Project", White Paper, ONF, June 2023.

[84] G. Vallero, D. Renga, M. Meo, M. A. Marsan, "Greener RAN Operation Through Machine Learning," in IEEE Transactions on Network and Service Management, Vol. 16, No. 3, 2019. doi: 10.1109/TNSM.2019.2923881.

[85] W. Wang, et al., "Cellular Traffic Load Prediction with LSTM and Gaussian Process Regression," ICC, 2020. doi: 10.1109/ICC40277.2020.9148738.

[86] V. Singh, M. Gupta, C. Maciocco, "Intelligent RAN Power Saving using Balanced Model Training in Cellular Networks," International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks (WiOpt), 2022. doi: 10.23919/WiOpt56218.2022.9930603.

[87] T. Chen, C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). https://doi.org/10.1145/2939672.2939785

[88] https://medium.com/mlearning-ai/time-series-forecasting-with-xgboost-and-lightgbm-predicting-energy-consumption-460b675a9cee

[89] F. Liu, et al., "Integrated sensing and communications: Toward dual-functional wireless networks for 6G and beyond," *IEEE journal on selected areas in communications*, *40*(6), 1728-1767, 2022.

[90] R. Liu, et al., "Integrated Sensing and Communication based Outdoor Multi-Target Detection, Tracking and Localization in Practical 5G Networks," 2023. *arXiv:2305.13924*.

[91] C. Gómez-Vega, et al., "Device-Free Localization: Outdoor 5G Experimentation at mm-Waves,"*IEEE Communications Letters*, 2023.

[92] K. Gao, et al., "Toward 5G NR high-precision indoor positioning via channel frequency response: A new paradigm and dataset generation method," *IEEE Journal on Selected Areas in Communications*, *40*(7), 2233-2247, 2023.

[93] K. Ko, et al., "High-speed train positioning using deep kalman filter with 5G NR signals," *IEEE Transactions on Intelligent Transportation Systems*, *23*(9), 2022.

[94] C. Huang, et al. "Reconfigurable intelligent surfaces for energy efficiency in wireless communication." IEEE Transactions on Wireless Communications, 18.8, 2019.

[95] Z. Yang, et al. "Energy-efficient wireless communications with distributed reconfigurable intelligent surfaces." IEEE Transactions on Wireless Communications, 21.1, 2021.

[96] S. Jia, et al., "Reconfigurable intelligent surfaces for energy efficiency in D2D communication network," IEEE Wireless Communications Letters, 10.3, 2020.

[97] X. Liu, et al., “Machine learning empowered trajectory and passive beamforming design in UAV-RIS wireless networks," IEEE Journal on Selected Areas in Communications, 39.7, 2020.

[98] Z. Wang, et al., “Location awareness in beyond 5G networks via reconfigurable intelligent surfaces,” *IEEE Journal on Selected Areas in Communications*, *40*(7), 2022.

[99] A. Zappone, et al., “Wireless networks design in the era of deep learning: Model-based, AI-based, or both?,” IEEE Transactions on Communications, 67(10), 2019.

[100] E. Shtaiwi, et al., “RIS-assisted mmWave channel estimation using convolutional neural networks,” in IEEE WCNC, 2021.

[101] M. Xu, et al., “Deep learning-based time-varying channel estimation for RIS assisted communication,” IEEE Communications Letters, 26(1), 2021.

[102] I. C-L, Y. Liu, S. Han, S. Wang, G. Liu, “On Big Data Analytics for Greener and Softer RAN”, *IEEE Access*, August 2015.

[103] R. Nisbet, G. Miner, K. Yale, *Handbook of Statistical Analysis and Data Mining Applications,* 2nd edition, Academic Press, November 2017.

[104] C. K. Anjinappa, İ. Güvenç, "Coverage Hole Detection for mmWave Networks: An Unsupervised Learning Approach," in *IEEE Communications Letters*, Vol. 25, No. 11, Nov. 2021. doi: 10.1109/LCOMM.2021.3106251.

[105] Z. Wang, "Base station planning problem based on genetic algorithm and K-Means clustering algorithm," *IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, 2023. doi: 10.1109/EEBDA56825.2023.10090812.

[106] L. Jiang, B. Huang, L. Chen, Z. Li, "Research on location planning of 5G base station based on DBSCAN clustering algorithm," *2023 IEEE 2nd International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA)*, 2023. doi: 10.1109/EEBDA56825.2023.10090761.

[107] R. Guo, J. Zhang, “Research on 5G communication station location planning and regional clustering based on K-medoids and DBSCAN algorithm”, IEEE Conference on Telecommunications, Optics and Computer Science (TOCS), 2022.

[108] N. Kuruvatti, J. Molano, H. Schotten, ‘‘Mobility context awareness to improve quality of experience in traffic dense cellular networks,’’ in Proc. 24th Int. Conf. Telecommun. (ICT), May 2017.

[109] N. P. Kuruvatti, A. Klein and H. D. Schotten, "Prediction of Dynamic Crowd Formation in Cellular Networks for Activating Small Cells," *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, 2015. doi: 10.1109/VTCSpring.2015.7146028.

[110] B. Ma, B. Yang, Y. Zhu and J. Zhang, "Context-Aware Proactive 5G Load Balancing and Optimization for Urban Areas," in *IEEE Access*, Vol. 8, 2020. doi: 10.1109/ACCESS.2020.2964562.

[111] B. Yang, W. Guo, B. Chen, G. Yang and J. Zhang, "Estimating Mobile Traffic Demand Using Twitter," in *IEEE Wireless Communications Letters*, Vol. 5, No. 4, Aug 2016. doi: 10.1109/LWC.2016.2561924.

[112] M. Ester, H. Kriegel, J. Sander, X. Xu, “A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise”, in Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, 1996.

[113] W. Qu, G. Li and Y. Zhao, "On the Coverage Problem in Device-to-Device Relay Networks," in *IEEE Communications Letters*, Vol. 23, No. 11, Nov. 2019. doi: 10.1109/LCOMM.2019.2931543.

[114] R. Mokhtar, H. Abdel-Atty, K. R. Mahmoud, "Optimization of the Deployment of Relay Nodes in Cellular

Networks," in *IEEE Access*, Vol. 8, 2020. doi: 10.1109/ACCESS.2020.3011472.

[115] C. Madapatha, B. Makki, A. Muhammad, E. Dahlman, M. -S. Alouini, T. Svensson, "On Topology Optimization and Routing in Integrated Access and Backhaul Networks: A Genetic Algorithm-Based Approach," in *IEEE Open Journal of the Communications Society*, Vol. 2, 2021. doi: 10.1109/OJCOMS.2021.3114669.

[116] L. Davis (ed.), *Handbook of genetic algorithms*, Internat. Thomsom Computer Press, 1996.

[117] *J. Pérez-Romero, O. Sallent, "*On the Value of Context Awareness for Relay Activation in Beyond 5G Radio Access Networks," IEEE VTC2022-Spring, June, 2022.

[118] *J. Pérez-Romero, O. Sallent, O. Ruiz, "*On Relay User Equipment Activation in Beyond 5G Radio Access Networks," IEEE VTC2022 Fall, September, 2022.

[119] 3GPP Rel-18, "Architecture enhancements for 5G System (5GS) to support network data analytics services," TS 23288, v.18.3, 2023-09.

[120] V. Mnih, et al. "Human-level control through deep reinforcement learning," *Nature*, Vol. 518, No. 7540, 2015.

[121] E. Ekudden, "Energy-efficient packet processing in 5G mobile systems", Ericsson blog post. Accessed March 2023. [Online] Available: https://www.ericsson.com/en/reports-and-papers/ericsson-technology-review/articles/energy-efficient-packet-processing-in-5g-mobile-systems.

[122] J. Sydir, et al., "DPM-NFV: Dynamic Power Management Framework for 5G User Plane Function using Bayesian Optimization," GLOBECOM 2022 – 2022. doi: 10.1109/GLOBECOM48099.2022.10001394.

[123] H. T. Nguyen, T. Van Do, C. Rotter, "Scaling UPF Instances in 5G/6G Core with Deep Reinforcement Learning," in IEEE Access, Vol. 9, 2021. doi: 10.1109/ACCESS.2021.3135315.

[124] A. Mudvari, N. Makris and L. Tassiulas, "Exploring ML methods for Dynamic Scaling of beyond 5G Cloud-Native RANs," ICC 2022. doi: 10.1109/ICC45855.2022.9838562.

[125] A. Griffiths, A. Morsman, P. Veitch, "Understanding the Performance and Power Saving Tradeoffs of Server Sleep States", IEEE International Conference on Cloud Networking (IEEE CloudNet2023), November 2023.

[126] W. Zhang, et al., "Jaguar: Low latency mobile augmented reality with flexible tracking," Proceedings of the 26th ACM international conference on Multimedia, 2018.

[127] P. Jain, et al., "Overlay: Practical mobile augmented reality," Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services, 2015.

[128] Z. He, et al. "Adaptive compression for online computer vision: An edge reinforcement learning approach." ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 17.4, 2021.

[129] G. Lu, et al. "Deep learning for visual data compression." Proceedings of the 29th ACM International Conference on Multimedia, 2021.

[130] A. Galanopoulos, et al. "Measurement-driven analysis of an edge-assisted object recognition system," IEEE ICC 2020.

[131] H. Li, et al. "Jalad: Joint accuracy-and latency-aware deep structure decoupling for edge-cloud execution." IEEE 24th international conference on parallel and distributed systems (ICPADS), 2018.

[132] X. Ran, et al. "Deepdecision: A mobile deep learning framework for edge video analytics." IEEE INFOCOM 2018.

[133] J. Jiang, et al. "Chameleon: scalable adaptation of video analytics." Proceedings of the conference of the ACM special interest group on data communication, 2018.

[134] C. Hung, et al. "Videoedge: Processing camera streams using hierarchical clusters." IEEE/ACM Symposium on Edge Computing (SEC), 2018.

[135] Q. Liu, T. Han. "Dare: Dynamic adaptive mobile augmented reality with edge computing." IEEE 26th International Conference on Network Protocols (ICNP), 2018.

[136] P. Yang, et al. "Edge coordinated query configuration for low-latency and accurate video analytics." IEEE Transactions on Industrial Informatics, 16.7, 2019.

[137] Y. Liet al. "Mobiqor: Pushing the envelope of mobile edge computing via quality-of-result optimization." IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017.

[138] S. Alyamkin, et al. "Low-power computer vision: Status, challenges, and opportunities." IEEE Journal on Emerging and Selected Topics in Circuits and Systems, 9.2, 2019.

[139] A. Goel, et al. "A survey of methods for low-power deep learning and computer vision,"in IEEE 6th World Forum on Internet of Things (WF-IoT), 2020.

[140] D. Bega, et al. "CARES: Computation-aware scheduling in virtualized radio access networks." IEEE Transactions on Wireless Communications 17.12, 2018.

[141] D. Bega, et al. "DeepCog: Optimizing resource provisioning in network slicing with AI-based capacity forecasting." IEEE Journal on Selected Areas in Communications 38.2, 2019.

[142] D. Raca, et al. "On leveraging machine and deep learning for throughput prediction in cellular networks: Design, performance, and challenges," IEEE Communications Magazine 58.3, 2020.

[143] J. Ayala-Romero, et al. "Online learning for energy saving and interference coordination in HetNets," IEEE Journal on Selected Areas in Communications, 37.6, 2019).

[144] J. Alcaraz, et al. "Online reinforcement learning for adaptive interference coordination." Transactions on Emerging Telecommunications Technologies, 31.10, 2020.

[145] F. Mismar, et al., "A framework for automated cellular network tuning with reinforcement learning." IEEE Transactions on Communications, 67.10, 2019.

[146] Z. Zhang, et al. "DQ scheduler: Deep reinforcement learning based controller synchronization in distributed SDN." IEEE ICC 2019.

[147] T. Lin, et al. "Microsoft coco: Common objects in context." Computer Vision–ECCV 2014.

[148] M. Everingham, "The pascal visual object classes challenge: A retrospective." International journal of computer vision 111, 2015.