CUSTOMIZE • OPERATE • CAPITALIZE

MX-PDK

Spec V1.3

A Flexible Multi-x 5G/6G O-RAN Platform Development Kit

OUR TECHNOLOGY

bubble ran

YOUR NETWORK

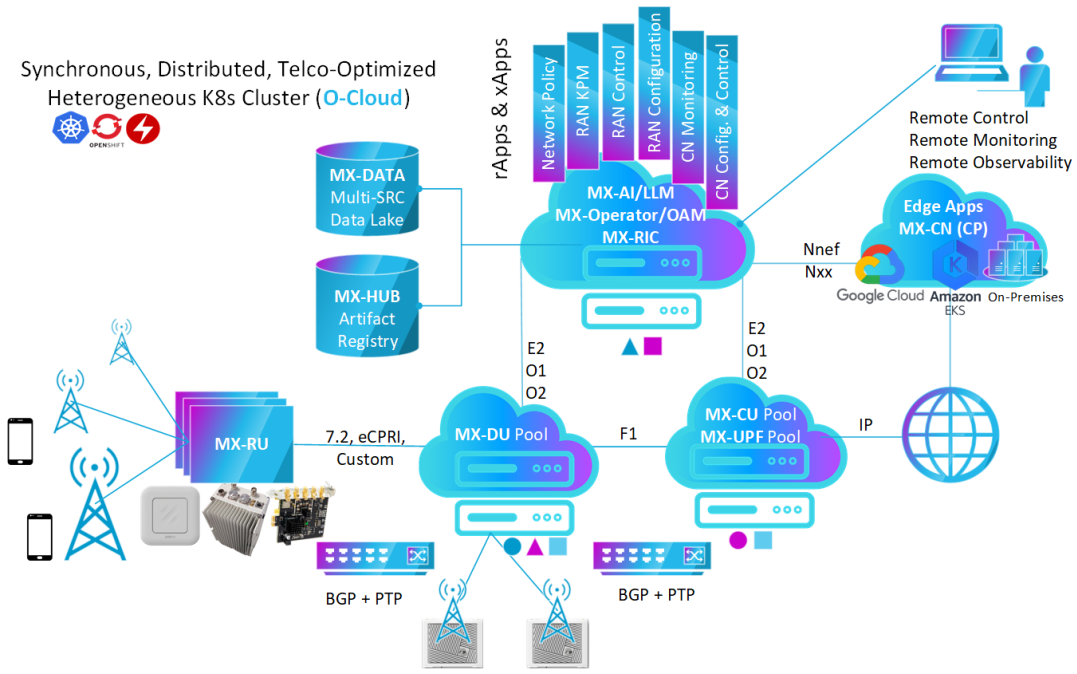
1 General Description and Features

The multi-x[1] platform development kit (MX-PDK) is an integrated turnkey HW-SW platform on x86 processing units, allowing to build a **cloud-native multi-vendor 5G/6G network infrastructure** interoperable with most types of user equipment and devices. It supports both Open Source 5G stacks, namely OpenAirInterface, SRSRAN, and Open5GS, and industrial-grade 5G stacks, namely Amarisoft and Lite-On, and extend them with a cloud-native multi-vendor O-RAN compliant stack designed and developed by BubbleRAN. MX-PDK features **end-to-end network slicing, network Intelligence, Data-lake and observability stack, and open ecosystem of xApps and rApps.**

1.1 MX-PDK Software Components

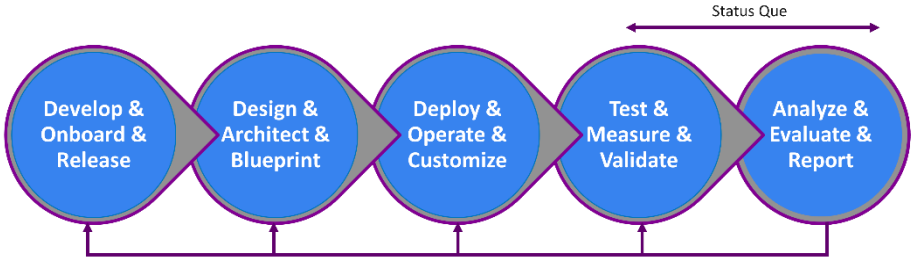
Figure below illustrates software components by means of a sample deployment model allowing you to operate and automate a high-performance, reliable, and secure 4G/5G O-RAN infrastructure at scale that is simple to use, customize and extend.

- **O-Cloud:** Telco-Optimized Kubernetes Cluster featuring synchronization, auto-device discovery, optimized data plane, and eBPF observability;
- **Multi-x 5G/6G Stack:** 5GC, IMS, and eNodeB, gNodeB, CU, DU in Non-standalone mode (NSA) and Standalone mode (SA) deployments in frequency range FR1 and FR2;
- **Multi-x O-RAN Stack:** SMO/OAM, Non-RT-RIC, Near-RT RIC, xApps, and rApps;
- **Multi-x Data:** Multi-source data lake and observability stack with metrics, stats, logs, traces from infrastructure, 5G network, 5G Terminal, and energy consumption;
- **Multi-x Hub:** Artifact registry of reusable images, packages, and blueprints.
- **Multi-x AI stack:** Multi-agent intent-driven intelligent plane with GenAI/LLM/LAM.



[1] *What are the dimensions of Multi-X ?* Multi-vendor, Multi-version, Multi-RAT, Multi-Frequency, Multi-RF, Multi-cloud, Multi-OS, Multi-deployment.

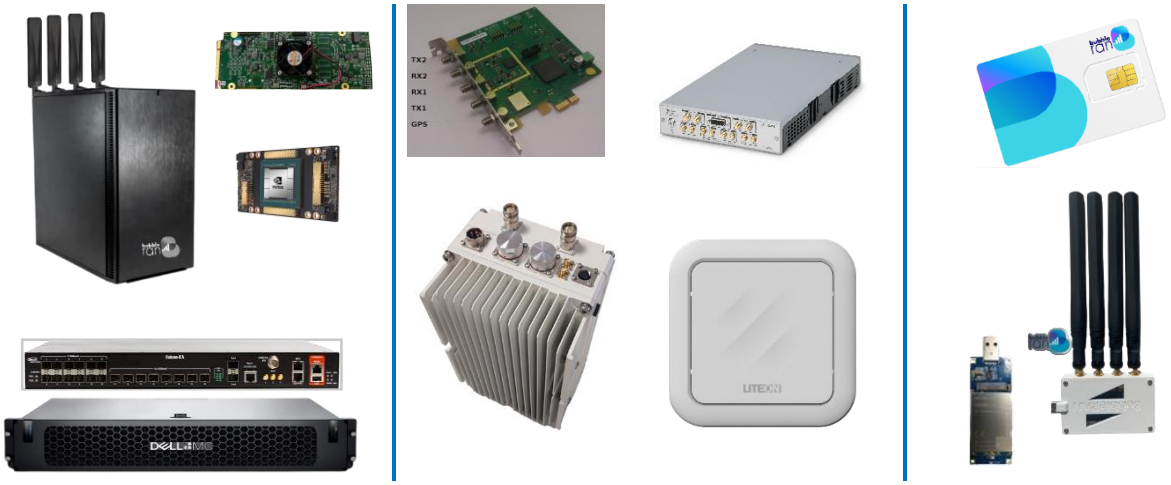
BubbleRAN MX-PDK is designed for **R&D, PoC/MVP, and Test and Validation** showcasing divers use-cases and business applications with guaranteed performance and reliability with the **target TRL from 4 to 7**. It also provides a built-in **Open RAN studio** (ORS)[2], compliant with the O-RAN specification, allowing to learn, develop, integrate, and test an end-to-end 5G/6G O-RAN with xApps/rApps and UEs in the loop. The SMO/OAM comes with a set of user interfaces, including CLI, APIs, and GUI that facilitates the operation of the MX-PDK infrastructure. As shown in the figure, users are able to **(1)** develop network functions and customize and extend the network, **(2)** design, mix-and-match, and deploy different vendors to create multiple concurrent networks, **(3)** perform full lifecycle operations from "Day 0 to Day n" on an end-to-end 3GPP and O-RAN compliant network with user and application in the loop, and **(4)** collect user and network data, analyze and evaluate the results.



1.2 MX-PDK Hardware Components

The MX-PDK hardware platform is **optional** to allow customers to reuse their existing hardware and radio equipment to build and extend their multi-x 5G/6G O-RAN infrastructure.

- **O-Cloud:** high-performance compute nodes optionally with hardware accelerators, under Ubuntu or RHEL Linux distribution. It is recommended to have at least three machines with intel-based or AMD-based;[3]
- **Multi-x Radio unit:** O-RAN 7.2 RU, eCPRI split 8 RRH, and SDRs;
- **4G/5G Device**: Quectel Modem, SIM cards, and accessories;
- **Synchronization**: PTP time-aware swathing fabric and grande master (GM).



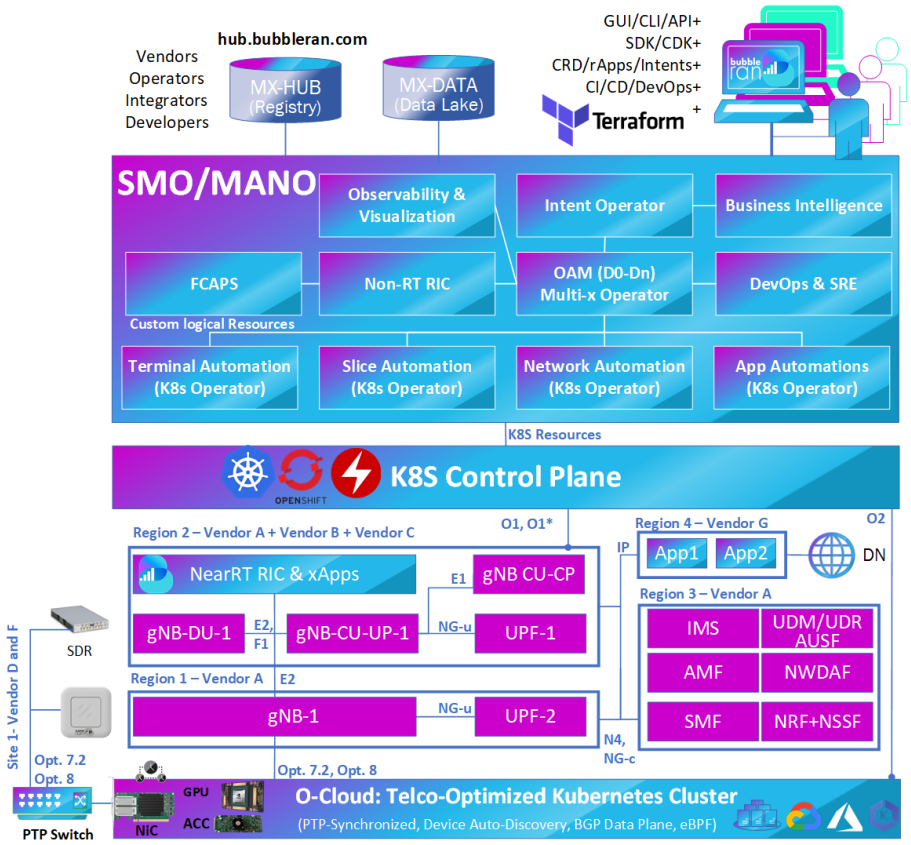
[2] https://bubbleran.com/products/mx-ors/

[3] Typical Spec: i9/Xeon/Epyc3, >4GHz/16C, >32RAM, > 250 GB SSD (write-intensive), NIC: 10/25Gbps (Intel E810, E820, X710), Form Factor: Tower or RACK, Cooling: Liquid + Fan

2 MX-PDK Software Stack

The software stack and components of MX-PDK is shown below. It includes multi-vendor 3GPP and O-RAN compliant software stacks running on the top of the same Multi-x hardware platform.

- **MX-RAN:**
  - **OAI:** 4G eNB, 5G gNB/CU/DU for Frequency Range 1; 4G/5G soft UEs.
  - **srsRAN:** 4G eNB; 5G gNB, and 4G/5G soft UE;
- **MX-CN:**
  - **OAI:** 4GC/5GC;
  - **Open5GS:** 4GC/5GC;
- **MX-HUB:** Artifact Registry serving archives, packages, images, models, Operator bundles, and configurations;
- **O-RAN O-Cloud:** optimized Kubernetes-based 4G/5G cloud infrastructure featuring synchronization, auto-device discovery, optimized data plane, and eBPF.
- **O-RAN SMO:** Cloud-native intelligence, full Life-cycle management and automation, observability stack and Multi-source data lake (MX-DATA), on-premises and public cloud deployment.
- **O-RAN MX-RIC:** multi-vendor near-RT-RIC and Non-RT-RIC supporting OAI, srsRAN, Amarisoft, and Lite-On. It includes the following O-RAN compliant service models with E2AP V2/v3 and A1AP v1: KPM (v2/v3), RAN control (RC v1.x), and Cell configuration and Control (CCC v3.x);
- **O-RAN rApps/xApps:** an ecosystem of xApps and rApps showcasing the O-RAN service models.
- **Tool box:** realtime E2E performance measurement, logging, and tracing tools, CLI, API, and dashboard to instrument the network



3 MX-RAN

Features	Value
3GPP Release	16
Frequency Bands	All TDD bands in FR1
Bandwidth	Up to 100MHz
MIMO	2x2 and 4x4
Subcarrier Spacing	Data and SSB subcarrier spacing: 15,30 KHz
Modulation and Coding Scheme	Up to 256QAM in DL and 64QAM in UL
Number of UEs	16
Handover	Experimental
Supported Modes	SA, NSA (Experimental)
Deployment Scenario	Monolithic (gNB), Disaggregated (CU, DU)
Use case	eMBB, uRLLC, O-RAN
Network Interfaces	NG interface (NGAP and GTP-U), Xn, E2, F1, E1 (OAI)

4 MX-CN

Features	Value
3GPP Release	16
Network elements	AMF, SMF, AUSF, UDM, UPF, NRF, NSSF
AS integrity and encryption	Snow3G and AES
IP version	OAI & Open5GS: IPv4, IPv6
QoS	Configurable QFI
PDU Session	Multi PDU sessions support – Ipv4 , IPv6 (EXP)
Handover	Xn
Slicing	Multiple Slice (shared or dedicated NF)
Network Interface	NG, N2, N3, N4, N8, N10, N11, N12,N13

5 O-RAN RIC (MX-RIC)

Features	Value
Multi-Vendor	Yes
RT and NearRT RIC	Yes
Non-RT RIC	Yes
Service Models	O-RAN Key Performance Measurement (KPM V2.0/V3.0) O-RAN RAN Control (RC v1.x) O-RAN Cell Configuration and Control (CCC v3.x) BubbleRAN Custom Service Models
xApps	Data collections, performance monitoring, RAN stats, RAN slicing, RAN reconfiguration, load balancing.
rApp	FlexMon FlexPolicy
xApp Language	C, C++, Python, Go, Java.
E2	Yes, V2 and V3
A1	Yes (SDK approach at xApp)
O1	Yes (supported for Lite-ON All-in-On gNB)

Detail of service models and xApp features are included in the table below.

xApp Features

Language	O-RAN SMs				Customized SMs					
	KPM		RC	CCC	MAC	RLC	PDCP	NGAP	SLICE CTRL	Traffic CTRL
	v2	v3	v1	v3						
C	v	v	v	v	v	v	v	v	v	v
Python	v	v	v	v	v	v	v	v	v	TBD
Go	v	v	24'Q3	24'Q3	v	v	v	v	v	TBD
Database										
SQLite3	v	v	24'Q3	24'Q3	v	v	v	v	v	N/A
MySQL	v	v	24'Q3	24'Q3	v	v	v	v	v	N/A
Timescale DB	v	v	TBD	TBD	v	v	v	TBD	TBD	N/A

6 O-RAN SMO (MX-Operator)

Features	Value
Level 1 life cycle operation	Resource detection/discovery (day 0), deploy (day 1), test (day 2)
Level 2 life cycle operation	Release and upgrade (day 2)
Level 3 life cycle operation	Full lifecycle control, including provisioning (day 0), configuration (day 1), and reconfiguration (day 2)
Level 4 life cycle operation	Deep insight, observability stack, log processing, metrics, and alarms
Multi-X	Multi-vendor, multi-version, multi-container, multi-OS, multi-node, multi-RAT, multi-RF
Security	Isolation, Signed Artifacts, Unprivileged, Rootless, Access Control, RBAC
User Interface	API and CLI, GUI, and Dashboard.
Image registry	MX-HUB (hub.bubbleran.com)
Networking	Calico (BGP, eBPF, IPVS) over Ethernet/Fiber, Multus
Storage	Host storage, Rook and Ceph
CI/CD	Consistent artifact generation, integration, validation, security, CDK, Jenkins,
Infrastructure	K8s distribution on Kubeadm, MicroK8s, Openshift, cloud-init, hardware resource detection and automation
Container runtime	Docker, Containerd, Podman, CRI-O, Snap, LXD (coming soon)

6.1 Salient Features

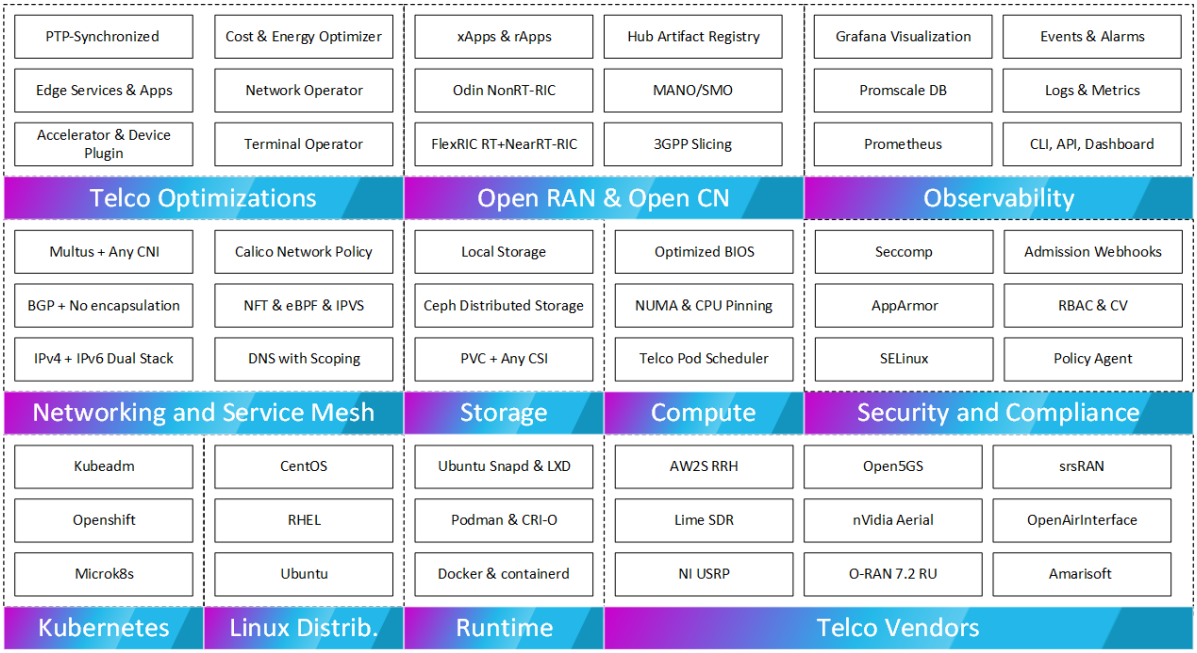
- Full Day-0, Day-1, Day-2 Level-5 Autopilot[4]
- Slice definition, assignment, and scaling
- Multiple interfaces support for the RAN and CN
- Full scheduling control on the container placement
- DNS scoping and Edge services
- High scalability with minimum deployment time (E2E less than a minute)
- Reconfiguration and full-stack observability
- Very low overhead and high-performance containers
- Full customization and extensibility for both the Manager and Operators
- Fully cloud-native with non-privileged containers supporting both private (on-premise) and public cloud providers
- Automatic device discovery and mapping for GPUs, SDRs, RRHs, etc.
- Network terminal management
- Application in the loop [definition of the application for the terminals]
- Idempotent and declarative logic design

[4] https://sdk.operatorframework.io/docs/overview/operator-capabilities/

7 Building Blocks

BubbleRAN features multi-vendor 4G/5G RAN, CN, and IMS stacks as well as SDR, RRH, and O-RAN compliant RU options with both Open-Source and Industrial grade 4G/5G stacks, as shown in the figure below. They can be deployed in a bare-metal, container, a Kubernetes cluster in a private or public cloud fully operated by the MX-Operator, a.k.a. SMO in O-RAN terminology.

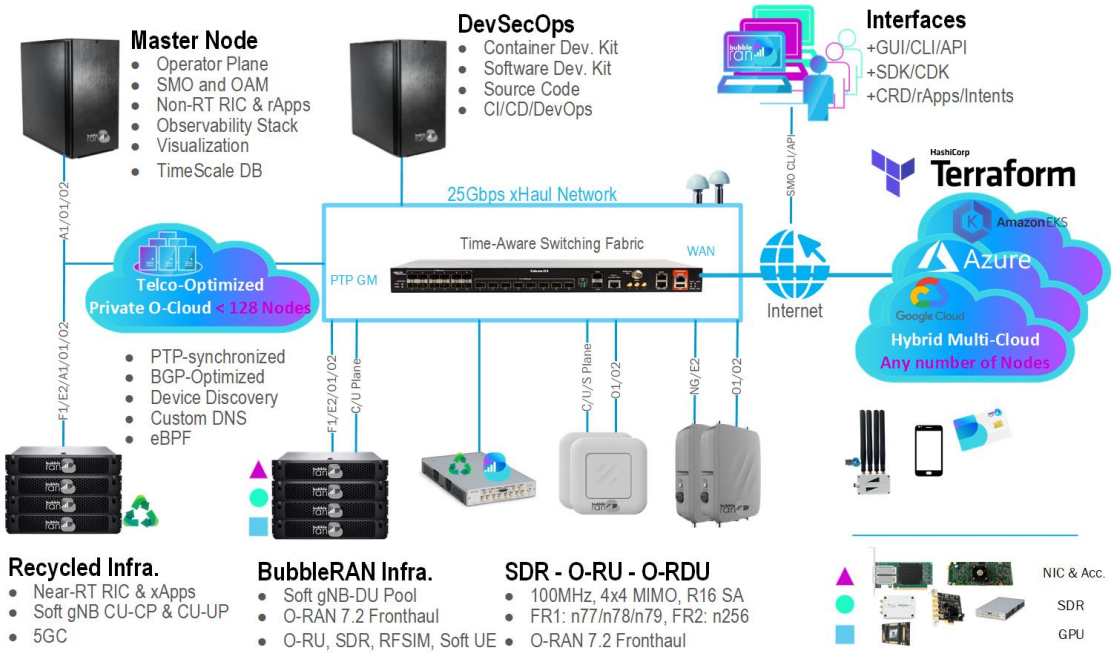
As detailed in the figure below, the MX-PDK product is composed of three main layers including **(1)** host telco-optimized infrastructure and a multi-vendor support, **(2)** cloud-native resources and security, and **(3)** telco software operators, applications and services designed to deliver high performances and reliability, security, observability, and fault management.



8 Sample Cloud-Native 5G/6G O-RAN Infrastructure

Figure below shows a sample cloud-native 5G/6G O-RAN infrastructure highlighting both hardware and software components, as described in Section 1.1 and 1.2. MX-PDK introduce different node roles, namely:

- **Master Node or K8s control role:** where the entire infrastructure and network operations are triggered. Typically, it includes different Kuberntes Operators, namely O-RAN SMO/OAM as well as the user interface to operate and manage the whole infrastructure and network. Typically, one master node is enough but for high reliability and availability, it is recommended to use three nodes.
- **Worker nodes:** where different network functions and applications are deployed. Typically the number of worker nodes depends on the scale of the deployment. Indicatively, for each gNB (100 MHz, 4x4), you would need 68 CPUs at 4GHz.
- **DevOps nodes (Optional):** where custom images, NF, xApps, and rApps are developed and manually deployed and attached to the network. DevOps machine includes xApp/rApp software development kit (SDK), container development kit (CDK) to build a custom image, source code 5G stacks, and snap packages. Typically, one node is enough, but if multiple developers would like to work in parallel, it is recommended to have one DevOps machine per developer.



CUSTOMIZE • OPERATE • CAPITALIZE

bubble ran

We are helping organizations to seamlessly build, customize, and operate their private 4G/5G infrastructure by consolidating open RAN and cloud-native architectures with a green MANO/SMO offering more than 10x efficiency and delivery cycle with lower carbon footprint for a wide range of R&D and enterprise use-cases from the lab to the production environment.

OUR TECHNOLOGY

bubble ran

YOUR NETWORK

SRSRAN FOXCONN amarisoft Open5GS OPEN AIR INTERFACE National Instruments AW2S LimeSDR

Twitter @BubbleRANTech
LinkedIn @BubbleRAN
E-mail contact@bubbleran.com
Website https://bubbleran.com
Address 450 route des Chappes F-06410 BIOT Sophia Antipolis

CUSTOMIZE • OPERATE • CAPITALIZE

MX-ORS/MX-PDK

Frequently Asked Questions

V1.3

Explore multi-x dimensions in 5G/6G O-RAN

OUR TECHNOLOGY

bubble ran

YOUR NETWORK

1 Frequently Asked Questions

Q1. What interfaces from the O-RAN architecture does it support??

We support E2AP v2/3, A1AP v1.x, and O1 interfaces in the O-RAN Architecture.

For R1 we use A1AP to expose policy to the rApps.

Q2. What are the supported parameters in each O-RAN service model implementation?

xApp Features

Language	O-RAN SMs				Customized SMs					
	KPM		RC	CCC	MAC	RLC	PDCP	NGAP	SLICE CTRL	Traffic CTRL
	v2	v3	v1	V3						
C	v	v	v	v	v	v	v	v	v	v
Python	v	v	v	v	v	v	v	v	v	TBD
Go	v	v	24'Q3	24'Q3	v	v	v	v	v	TBD
Database										
SQLite3	v	v	24'Q3	24'Q3	v	v	v	v	v	N/A
MySQL	v	v	24'Q3	24'Q3	v	v	v	v	v	N/A
Timescale DB	v	v	TBD	TBD	v	v	v	TBD	TBD	N/A

E2SM KPM:

"DRB.PdcpSduVolumeDL"
"DRB.PdcpSduVolumeUL"
"DRB.RlcSduDelayDl", only OAI
"DRB.UEThpDl"
"DRB.UEThpUl"
"RRU.PrbTotDl", only OAI
"RRU.PrbTotUl", only OAI

3GPP Reference for the KPM measurement names:
TS 28.552: https://www.3gpp.org/ftp/Specs/archive/28_series/28.552/28552-i50.zip
TS32.425: https://www.3gpp.org/ftp/Specs/archive/32_series/32.425/32425-h10.zip

E2SM RC

Supported O-RAN controls are:
DRB QoS Modification (OAI)
Slice-level PRB Quota (OAI, RAN Slicing)
Handover Control (AMR)
Handover Control (OAI soon)

O-RAN Reference:
ORAN.WG3.E2SM-RC-v01.03

E2SM CCC

UL/DL "O-BWP" (Amarisoft)
Frequency reconfiguration (AMR)
BWP reconfiguration in DL and UL (OAI soon)
Frequency reconfiguration (OAI soon)

O-RAN Reference:

ORAN.WG3.E2SM-CCC-R003-v03.00

Relevant Links:

- Monitoring: See the links below
- RAN slicing: See the link below
- HO (AMR): Live demo during our next on-boarding session.
- FlexSLice paper

Q3. What are the use-cases supported service model?

1. KPM: Performance Monitoring, Data-set collections
2. RC: Mobility management (HO), RAN Slicing, Load Balancing, QoS Control, Energy Management, Cell Management
3. CCC: BWP reconfiguration, Frequency reconfiguration
4. Mac/RLC/PDCP/GTP: Extended monitoring, data-set collection
5. Slice Control: Radio Resource Slicing, Resource Allocation Policy, Slicing Policy, MCS Control
6. Traffic Control: QoS Control, Flow-level Slicing

Q4. What are the metrics that are available from the infrastructure, network and application?

Infrastructure. We have CPU, memory, storage, network, and energy, all consolidated in the timescaleDB/Prometheus.

Application. None.

Network. We support custom service models that are the superset of the O-RAN KPM service model to extract then network metrics. Below is the list of the supported parameters.

MAC	RLC	PDCP & GTP

typedef struct
{
uint64_t dl_aggr_tbs;
uint64_t ul_aggr_tbs;
uint64_t
dl_aggr_bytes_sdus;
uint64_t
ul_aggr_bytes_sdus;
uint64_t dl_curr_tbs;
uint64_t ul_curr_tbs;
uint64_t dl_sched_rb;
uint64_t ul_sched_rb;
 	
float pusch_snr; //: float = -64;
float pucch_snr; //: float = -64;
 	
float dl_bler;
float ul_bler;
 	
uint32_t dl_harq[5];
uint32_t ul_harq[5];
uint32_t dl_num_harq;
uint32_t ul_num_harq;
 	
uint32_t rnti;
uint32_t dl_aggr_prb;
uint32_t ul_aggr_prb;
uint32_t dl_aggr_sdus;
uint32_t ul_aggr_sdus;
uint32_t dl_aggr_retx_prb;
uint32_t ul_aggr_retx_prb;
 	
uint32_t bsr;
uint16_t frame;
uint16_t slot;
 	
uint8_t wb_cqi;
uint8_t dl_mcs1;
uint8_t ul_mcs1;
uint8_t dl_mcs2;
	typedef struct {
	/* TX */
	uint32_t txpdu_pkts;
	uint32_t txpdu_bytes;
	uint32_t txpdu_wt_ms;
	uint32_t txpdu_dd_pkts;
	uint32_t txpdu_dd_bytes;
	uint32_t txpdu_retx_pkts;
	uint32_t txpdu_retx_bytes;
	uint32_t txpdu_segmented;
	uint32_t txpdu_status_pkts;
	uint32_t txpdu_status_bytes;
	uint32_t txbuf_occ_bytes;
	uint32_t txbuf_occ_pkts;
	
	/* RX */
	uint32_t rxpdu_pkts;
	uint32_t rxpdu_bytes;
	uint32_t rxpdu_dup_pkts;
	uint32_t rxpdu_dup_bytes;
	uint32_t rxpdu_dd_pkts;
	uint32_t rxpdu_dd_bytes;
	uint32_t rxpdu_ow_pkts;
	uint32_t rxpdu_ow_bytes;
	uint32_t rxpdu_status_pkts;
	uint32_t rxpdu_status_bytes;
	uint32_t rxbuf_occ_bytes;
	uint32_t rxbuf_occ_pkts;
	
	/* TX SDU stats */
	uint32_t txsdu_pkts;
	uint64_t txsdu_bytes;
	double txsdu_avg_time_to_tx;
	uint32_t txsdu_wt_us;
	
	/* RX SDU stats */
	uint32_t rxsdu_pkts;
	uint64_t rxsdu_bytes;
	uint32_t rxsdu_dd_pkts;
	uint32_t rxsdu_dd_bytes;
	uint32_t rnti;
	uint8_t mode;
	uint8_t rbid;
	} rlc_radio_bearer_stats_t;
		typedef struct{
		uint32_t txpdu_pkts;
		uint32_t txpdu_bytes;
		uint32_t txpdu_sn;
		uint32_t rxpdu_pkts;
		uint32_t rxpdu_bytes;
		uint32_t rxpdu_sn;
		uint32_t rxpdu_oo_pkts;
		uint32_t rxpdu_oo_bytes;
		uint32_t rxpdu_dd_pkts;
		uint32_t rxpdu_dd_bytes;
		uint32_t rxpdu_ro_count;
		uint32_t txsdu_pkts;
		uint32_t txsdu_bytes;
		uint32_t rxsdu_pkts;
		uint32_t rxsdu_bytes;
		uint32_t rnti;
		uint8_t mode;
		uint8_t rbid;
		}
		pdcp_radio_bearer_stats_t;
		
		typedef struct {
		uint32_t rnti;
		uint32_t teidgnb;
		uint32_t teidupf;
		uint8_t qfi;
		} gtp_ngu_t_stats_t;

uint8_t ul_mcs2; int8_t phr; } mac_ue_stats_impl_t;		

Q5. What are the OAM command line?

Commands	Descriptions	
observe	Open up the observation toolbox	
cic	CLI in CLI	
install	Install a component on Trirematics	
	Ex: cli install operator/model/network	
extract	Extract contents from the containers	
	config	Get decoded configurations out of an Element
	logs	Get logs from a specific Element
	graph	Get the network graph
	pcap	Get PCAP from the network interfaces of the Workload
	infra	Get infrastructure data
	port	Get particular port number and IP address from a specific Element
test	Perform an evaluation	
	rtt	Measure the round-trip time
	throughput	Measure the network throughput
remove	Remove a component on Trirematics	
	Ex: cli remove operator/model/network	
completion	Generate the auto-completion script for the specified shell	
	Ex: cli completion bash/powershell/fish/zsh	
login	Authenticate to Harbor for the current namespace	
list	List all the operators, models, or networks	
diag	Perform diagnostics against the cluster	
help	Help about any command	
run	Run an arbitrary command list	

Q6. How closely the RIC is tied to OAI?

There is no coupling between MX-RIC and OAI. In fact, RIC is tied to E2 Agent, and the E2 agent is tied to the RAN functions.

MX-RIC is by design multi-vendor, and currently supports OAI, srsRAN, Amarisoft, and a tier3 vendors such as Lite-On. This means that each vendor implements a subset of parameters in each service model.

Q7. Is the A1 interface fully standard-compliant? Are there A1 clients/agents already available / tested against the near-RT RIC A1 interface?

A1 interface is fully standard compliant and it is extended to support additional functionalities, such as direct RAN slicing in an effort to provide more control capabilities in the Non-RT and rApp levels.

Today the policies that are supported through the rApps are the following:

- **Target PRB Utilization:** A target RAN PRB utilization is enforced and the appropriate xApp does the necessary control actions in the RAN to bring the current closer to the desired state
- **Slice Enforce:** Slicing control policies are sent, such as creating, updating or deleting slices and also associating the UEs to the various slices
- **Monitoring Job:** A flexible monitoring job is enforced in order to monitor a list of target statistics of one or multiple cells under one or multiple networks.

The rApps are implemented as applications running on top of the Non-RT RIC consuming its APIs. The latter is a Kubernetes Operator exposing new Custom Resource Definitions (CRD) to be consumed. Hence, rApps use these CRDs in order to send policies and receive feedback utilizing the cabernets' client. The rApps are written in go, python and JavaScript.

R1 is not currently clearly specified. It is shown to include multiple services, such as A1 services among other SMO services. Hence, for R1, we mirrored and extended the A1 as CRDs between non-RT RIC and rApps.

Relevant Links:

- https://youtu.be/9mnun7RXVg8?si=pDendeG-WAqUhqDb

Q8. Is the O1 interface fully standard-compliant? Are there O1 clients/server already available / tested against the near-RT RIC A1 interface?

Yes.

O1-CM (Configuration and Reconfiguration manager) only available with the Liteon AIO and RU (soon).

We do not use the O-RAN O1-compliant interfaces to configure and manage OpenAirInterface, srsRAN, and Amarisoft since they do not implement O1 servers. Recently, there have been some activities for OAI on the O1 interface but they do not provide all the required features for our SMO.

Nevertheless, this does not limit our SMO in its functions with respect to the NFs or RAN software, but rather improves it for day-2 operations. If you use the `cli extract pcap` command to extract the PCAP in the containers, you would see some HTTP messages

transported for the REST API between the Workload and the Manager containers that might resemble the O1 messages that you are looking for.

In terms of the operations, any change in the YAML file used for the network is propagated automatically to all the affected NFs. In terms of monitoring and fault-tolerance, we use cloud-native software such as Prometheus that are natively integrated with our SMO.

If you would need a specific control mechanism, please request.

Q9. What are different xApp classes?

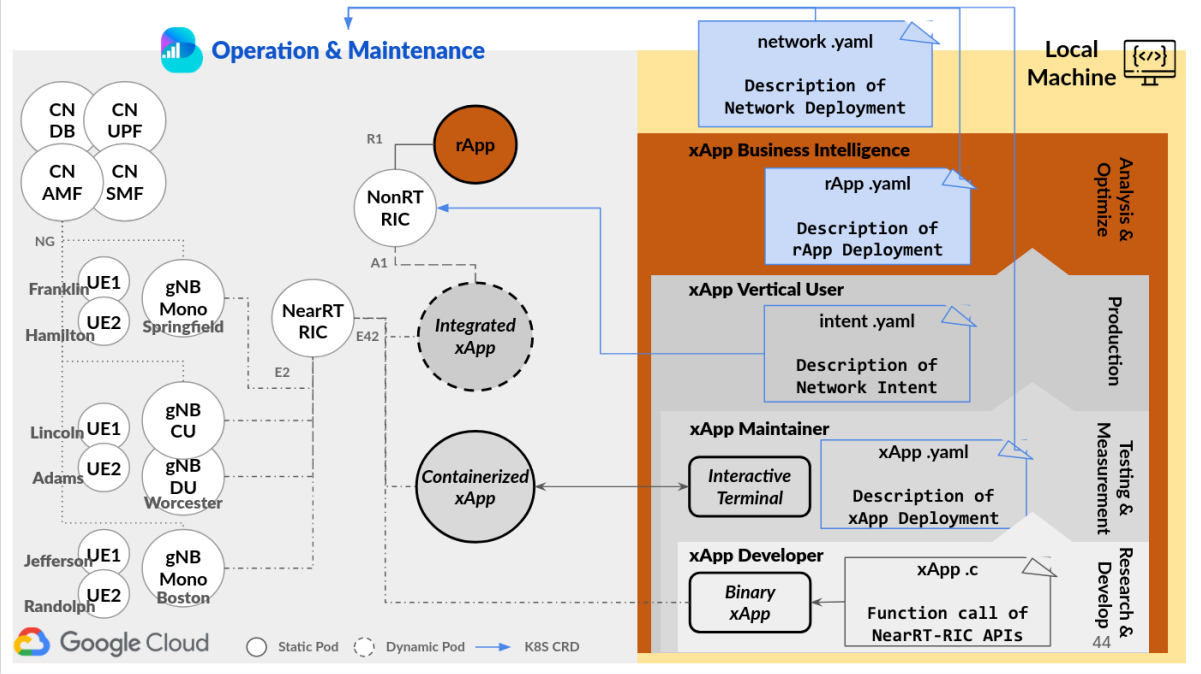
xApp Independent Lifecycle			
xApp Developer	**xApp Maintainer**	**xApp Vertical User**	**xApp Business Intelligent**
◆*RANs & UEs Monitoring xApp* ◆*Resource Allocation xApp* ◆*Network Slice Control xApp* ◆*Handover Control xApp*	◆*Interactive Monitor & Control xApp* ◆*Auto-configured xApp* ◆*Performance Visualization xApp*	◆***FlexMon**: Intent-based Monitoring xApp* ◆***FlexPolicy**: Policy Enforcement xApp*	◆*Cell Management rApp* ◆*E2E Slice Operator*

Q10. How to develop and deploy xApps?

Open documentation includes xApp and SM documentation for users and developers.

- https://bubbleran.com/docs/category/xapp
- https://bubbleran.com/docs/tutorials/flexric/developers-guide/xapp/xapp-devops-evolution/

Below you can see the xApp /rApp lifecycle from development to production.



A software-development kit (SDK) is provided for the development of an xApp. Please have a look at the following link for more information about the SDK.

- https://bubbleran.com/docs/category/ric-development-guide

For onboarding an xApp or a NF as a container, a Container Development Kit (CDK) is provided in the DevOps package that allows you to integrate your custom function into the MX-ORS and MX-PDK.

Q11. *BubbleRAN Related Publication and Research Projects?*

https://bubbleran.com/research/

For reference, here is the list of publications:

- MX-RIC: https://www.eurecom.fr/publication/6737/download/comsys-publi-6737.pdf
- MX-Operator: https://www.eurecom.fr/publication/7428/download/comsys-publi-7428.pdf
- E2SM TC-RAN: https://www.eurecom.fr/publication/7498/download/comsys-publi-7498.pdf
- OpenAirInterface: https://www.eurecom.fr/publication/4434/download/cm-publi-4434.pdf
- FlexSlice: https://www.eurecom.fr/publication/7416/download/comsys-publi-7416.pdf

Q12. Could an xApp interwork with physical interfaces? Currently xApps work with simulated devices.

Yes. See the answer to Q18.

The xApps would work regardless of the mode of connectivity in the RAN. If you use SDR, USRP, O-RAN 7.2, eCPRI, etc. all work the same.

Q13. How are xApps deployed? Is there an SDK? How shall we package an xApp into the container.

The xApps have an SDK provided by BubbleRAN. You could read the documentation at:

https://bubbleran.com/docs/category/ric-development-guide

For onboarding them as containers, you could use our Container Development Kit (CDK) or we could just help you package it with the required APIs. We are working on a documentation for this, but for now your xApps could just run as normal processes outside the cluster and connect to the RIC and gNBs running inside the cluster.

Q14. How can we monitor O1 OAI and Amarisoft Data? Which O1 control mechanisms are available?

We do not use the O-RAN O1-compliant interfaces to configure and manage neither of our vendors (except LITEON) since they do not implement O1 servers. Recently, there have been some activities for OAI on the O1 interface but they do not provide all the required features for our SMO.

Nevertheless, this does not limit our SMO in its functions with respect to the NFs or RAN software, but rather improves it for day-2 operations. If you use the cli extract pcap command to extract the PCAP in the containers, you would see some HTTP messages transported for the REST API between the Workload and the Manager containers that might resemble the O1 messages that you are looking for.

In terms of the operations, any change in the YAML file used for the network is propagated automatically to all the affected NFs. In terms of monitoring and fault-tolerance, we use cloud-native software such as Prometheus that are natively integrated with our SMO.

If you would need a specific control mechanism, please request.

Q15. Are 5QIs supported in OAI / Amarisoft RAN and Core? Is PDU Session modification supported?

The 5QIs are defined in the CN configuration in Amarisoft and SMF configuration in OAI. The difference between the two is that in Amarisoft you are allowed to define multiple E-RABs per DNN, but in OAI it is limited to a single one.

Upon the request from the UE for a particular DNN, to establish the PDU session, for each of the E-RABs a corresponding DRB is created. In Amarisoft this is governed by the DRB mapping configuration and in OAI it is always mapped to a single DRB.

In Amarisoft, the default configuration supports all the 5QI values and their corresponding DRBs, but only the DNN provided by your slice configuration in the YAML file as well as IMS and SOS DNNs are created by default. The DNNs for IMS and SOS use standard values for 5QI (5, 1, and 2) and the other one is always 5QI = 9. If you want to change any of these parameters, we could define some workarounds with annotations in the YAML file.

For the matter of PDU Session modification, if you are looking for APIs to connect to the core network to modify them, they are not provided by neither Amarisoft nor OAI.

Q16. What are the capabilities of the Non-RT RIC / SMO?

1. **Current (Non-RT RIC):**
   - **FlexPolicy:** Policy Job defines different control actions and policies to be applied on a selected set of E2 nodes, an evolution of the A1-P service. Use cases include load-balancing.
   - **FlexMon:** Monitoring Job defines programmable monitoring targeting selected E2 nodes with post-processing and metric selection exported to Timescale Database (TSDB) compatible with Prometheus and Grafana.

2. **Current (OAM):**
   - Full Day-0, Day-1, Day-2 Level-5 Autopilot
     - https://sdk.operatorframework.io/docs/overview/operator-capabilities/
   - Slice definition, assignment, and scaling
   - Multiple interfaces support for the RAN and CN
   - Full scheduling control on the container placement
   - DNS scoping and Edge services
   - High scalability with minimum deployment time (E2E less than a minute)
   - Reconfiguration and full-stack observability
   - Very low overhead and high-performance containers
   - Full customization and extensibility for both the Manager and Operators
   - Fully cloud-native with non-privileged containers supporting both private (on-premise) and public cloud providers
   - Automatic device discovery and mapping for GPUs, SDRs, RRHs, etc.
   - Network terminal management
   - Application in the loop [definition of the application for the terminals]
   - Idempotent and declarative logic design

3. **Planned (OAM)**

- Energy efficiency
- Extending the policy job to support use cases such as QoE, Slice SLA, etc.
- Advanced E2E slicing support (joint RAN and CN)
- Support of Fault tolerance and management
- Support of network security

Q17. How to develop and deploy rApps – is interworking via O1 and A1 supported?

Since the rApps in BubbleRAN are using cloud-native Kubernetes APIs, any SDK supporting Kubernetes clients would do. This means, you could implement your own logic in any language (Python, Go, JavaScript, or any REST-client implementation) by just consuming the Custom Resource Definitions (CRDs) created by our SMO.

Currently, three rApps are available:

- Cell manager
- Spectrum sharing
- Dynamic resource and NF scaling

We plan to develop an rApp SDK to facilitate the development of rApps.

Q18. *Could we connect an external workload (e.g. xApp or NF such as Core Network)?*

BubbleRAN fully supports external interfaces and each of the NFs could be adjusted to connect externally (as clients) or listen for external connection (as servers).

In addition, we support external DNS entries.

[1] YouTube Videos of MWC:
https://youtube.com/playlist?list=PLQlXa77TNMqa5NIHRNJymW4PQlGOk-PF_&feature=shared

Q19. *What wireless channels are supported by MX-ORS/MX-PDK?*

It depends on the RAN vendor. Please have a look at the OAI and srsRAN supported channel models.

Q20. How to integrate an application (video streaming app – client server, video call – peer-to-peer, robotic app – distributed docker containers deployed in the UE and in the

edge server) in MX-ORS/MX-PDK products, both one the 5G UE terminal and edge cloud?

MX-ORS and MX-PDK support any arbitrary containerized application to be attached to the user-terminal and be deployed at the edge cloud.

The actual integration might need our support, and requires evaluation.

Q21. *What are the steps to integrate AI models into xApps?*

This is the same as developing a new xApp from the MX-ORS and MX-PDK. Please refer to Q10.

Q22. *How many gNBs and UEs are supported in both emulation and over-the-air with the standard BubbleRAN hardware platform?*

It depends on different factor including: bandwidth, number of layers (e.g. MIMO), and radio frontend.

Indicatively for 100MHz 2x2 single carrier with an Intel i9, 4GHz/16C, 32RAM, NIC: 10Gbps, you can run:

- 3 gNBs
- 1 gNBs and 2-4 UEs.
- 2 gNBs and 2 UEs

Q23. *Is there any backup procedure?*

There is no specific procedure related to backup.

However, your deployment (subject to no change in the used composition models) is immutable and is captured in your Network YAML file and its composition model that ensure deployment reproducibility at any time in the future. Anything beyond that is not reproducible (like if you for example executed an interactive xApp and did something). You can use the command `cli list model` to factory reset the models, and use `cli install network` to deploy the same network.

To dump the current status of the cluster including Pods, Services, resources, etc. and using it for backup or debug purposes,you can use the `cli diag` command.

Related link:

- https://bubbleran.com/docs/category/cli-reference

Q24. Is there any UE related data available in the exported metrics?

Yes, by means of O-RAN E2SM and BubbleRAN custom service models. See the answers in Q1 to Q4.

Q25. Is it possible to modify the RAN configuration params like MCS, RB?

MCS is not a configurable parameters, it is decided based on BLER and/or CQI. RB is the function of the bandwidth, and you can change the bandwidth to change the total available RB. Using RAN slicing, you can dynamically reconfigure the total RB per slice over time.

In general, you can change the configuration file in two ways:

1. Management system using the SMO: day 0and day 2 operations
2. Control system using near-RT RIC: CCC E2SM allows you to perform a runtime reconfiguration

Q26. *Is it possible to* access a container of a NF and modify source code? What is needed?

You can access the container of any NF in a given deployment.

If the containers come with the source code, you can modify the source code on the fly. Some of our xApps provides such features.

You can change the source codes of any open-source based NFs that is supported by BubbleRAN (OAI, srsRAN), and connect to the MX-ORS and MX-PDK product family as an external off-cluster resources (see Q18). To on-board such a NF you would need the CDK (see Q13) included in the BubbleRAN DevOps package.

Q27. *Can more than one network get configured and used by separate persons?*

YES, if you mean more than one 5G Network. Note that the number of concurrent networks deployed on your infrastructure depends on the available compute/memory/network resources. This is indeed an effective infrastructure sharing across different users and networks. In K8s, there is the concept of namespace which allows to logically separate the infrastructures across different users and/or networks.

Q28. *What the supported commercial 5G smartphones and UEs/Devices?*

Quectel, Google phone, Samsung Galaxy, iPhone, oneplus.

Q29. How many users may operate the same system configurations simultaneously?

We don't have any limitations regarding this whether it is in the same namespace or across different namespaces.

Q30. BubbleRAN technology related question?

Check out our FAQ here: https://bubbleran.com/faq/

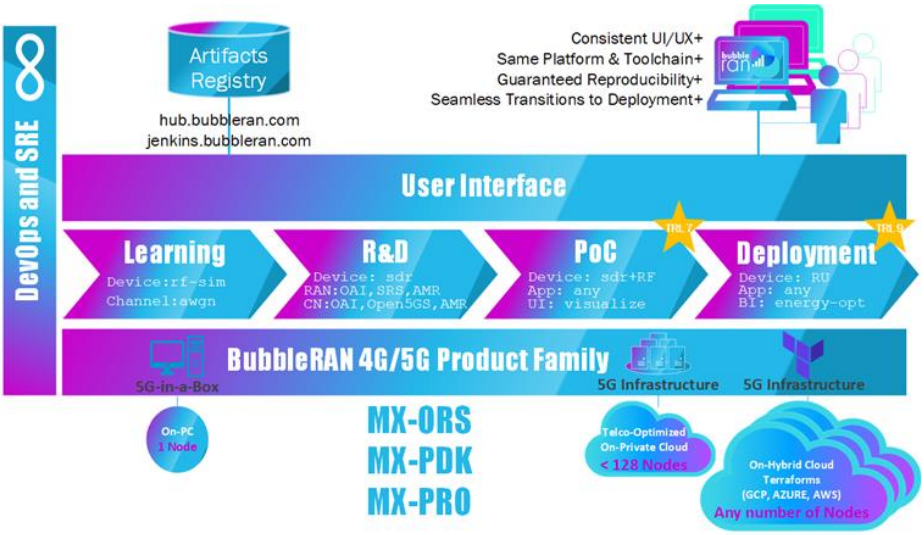
2 References

1. https://www.linkedin.com/company/bubbleran/
2. https://www.youtube.com/@bubbleran
3. https://bubbleran.com/technology/
4. https://bubbleran.com/products/mx-pdk/
5. https://bubbleran.com/products/mx-ors/
6. https://bubbleran.com/media/
7. https://bubbleran.com/faq/

3 Appendix

3.1 MX-PDK Use-cases

MX-PDK is all you need to build a multi-vendor 5G/6G infrastructure at scale for the purpose of R&D, demonstration, and use-case validation. Formally, it is a production-grade cloud-native 5G/6G infrastructure allowing to seamlessly design, operate, automate, and observe multiple concurrent end-to-end multi-vendor 4G/5G networks featuring Open RAN, slicing, and edge services, at scale.[1]



MX-PDK assists you from learning to deployment with a consistent and inclusive platform in any environment offering a consistent user experience and seamless transition from R&D to PoC, and from PoC to Deployment with guaranteed reproducibility. For example, an algorithm that has been tested by team A during the simulation, can be validated and demonstrated by team B during the PoC, effectively fostering cross-organization and cross-user collaboration.

[1] https://bubbleran.com/products/mx-pdk/

C U S T O M I Z E • O P E R A T E • C A P I T A L I Z E

bubble ran

We are helping organizations to seamlessly build, customize, and operate their private 4G/5G infrastructure by consolidating open RAN and cloud-native architectures with a green MANO/SMO offering more than 10x efficiency and delivery cycle with lower carbon footprint for a wide range of R&D and enterprise use-cases from the lab to the production environment.

OUR TECHNOLOGY

bubble ran

YOUR NETWORK

SRSRAN FOXCONN amarisoft Open5GS OPEN AIR INTERFACE National Instruments AW2S LimeSDR

Twitter @BubbleRANTech
LinkedIn @BubbleRAN
E-mail contact@bubbleran.com
Website https://bubbleran.com
Address 450 route des Chappes F-06410 BIOT Sophia Antipolis