# Welcome to Open RAN Studio Mini-Series


**Open RAN Studio Mini-Series - Europe-Asia Edition**

Event by BubbleRAN

Thu, May 25, 2023, 9:00 AM - 10:30 AM (your local time) Add to calendar

Online

# Agenda

| | |
|---|---|
| 8:55 – 9:05 | Welcome speech (Ilias) |
| 9:05 – 9:15 | Why Open RAN Studio? (Navid) |
| 9:15 – 9:30 | Live deployment of a cloud-native 5G SA blueprint (Navid) |
| 9:30 – 9:35 | Break |
| 9:35 – 9:50 | Live deployment of a cloud-native 5G Open RAN blueprint (Alireza) |
| 9:50 – 10:05 | Interactive KPM and data collection xApp (ChiehChun) |
| 10:05 – 10:15 | How to develop an xApp? (Ilias) |
| 10:15 – 10:20 | Towards ML DevKit for xApps (Khoa) |
| 10:20 – 10:30 | Q&A and Take a way message (All) |

# Disclaimer

- This episode will be delivered in a form of an **overview** of Open RAN/O-RAN coupled with two live end-to-end demos and few details about xApp design and development
  - Not to be considered as a tutorial or training.

- The materials and demo presented here are based on the **latest version (Bronze)** of Open RAN studio and not replicatable with the current version (Almond)
  - Existing examples require updates to work with the new version
  - Online Open Documentation is only valid for the current version of ORS (Almond)

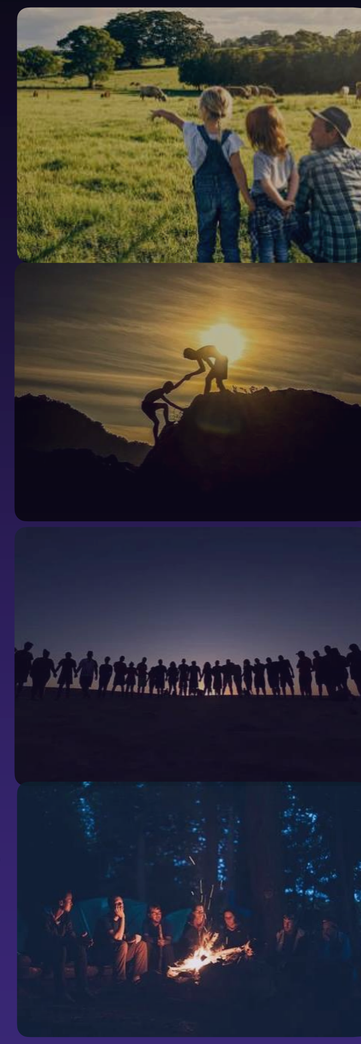
- The presented material should be viewed holistically. Overlooking to the concepts with unexpected accuracy is unadvised. Always do your own research or ask experts for a detailed review.

# Zoom Interactions

- **Always mute the microphone to minimize background noise**
- Please add your affiliation to your zoom name, e.g.  Navid Nikaein (BubbleRAN)
- Use Zoom Chat for any questions or comments outside of the Q&A session. This is highly recommended to be able to answer all the questions. The team will reply you.
- The recorded video will be published on the BubbleRAN Youtube channel. We will send a separate email with all the resources.

# Thanks to The Team

# To make this event possible



# Open RAN in Nutshell

- **What is Open RAN ?**
  - A **collective action** in building an open telecom network that complies to 3GPP
- **Why Open RAN?**
  - Because the established vendor lock-in is restricting innovation and does not allow more favorable horizontal market conditions
- **What is the Objective?**

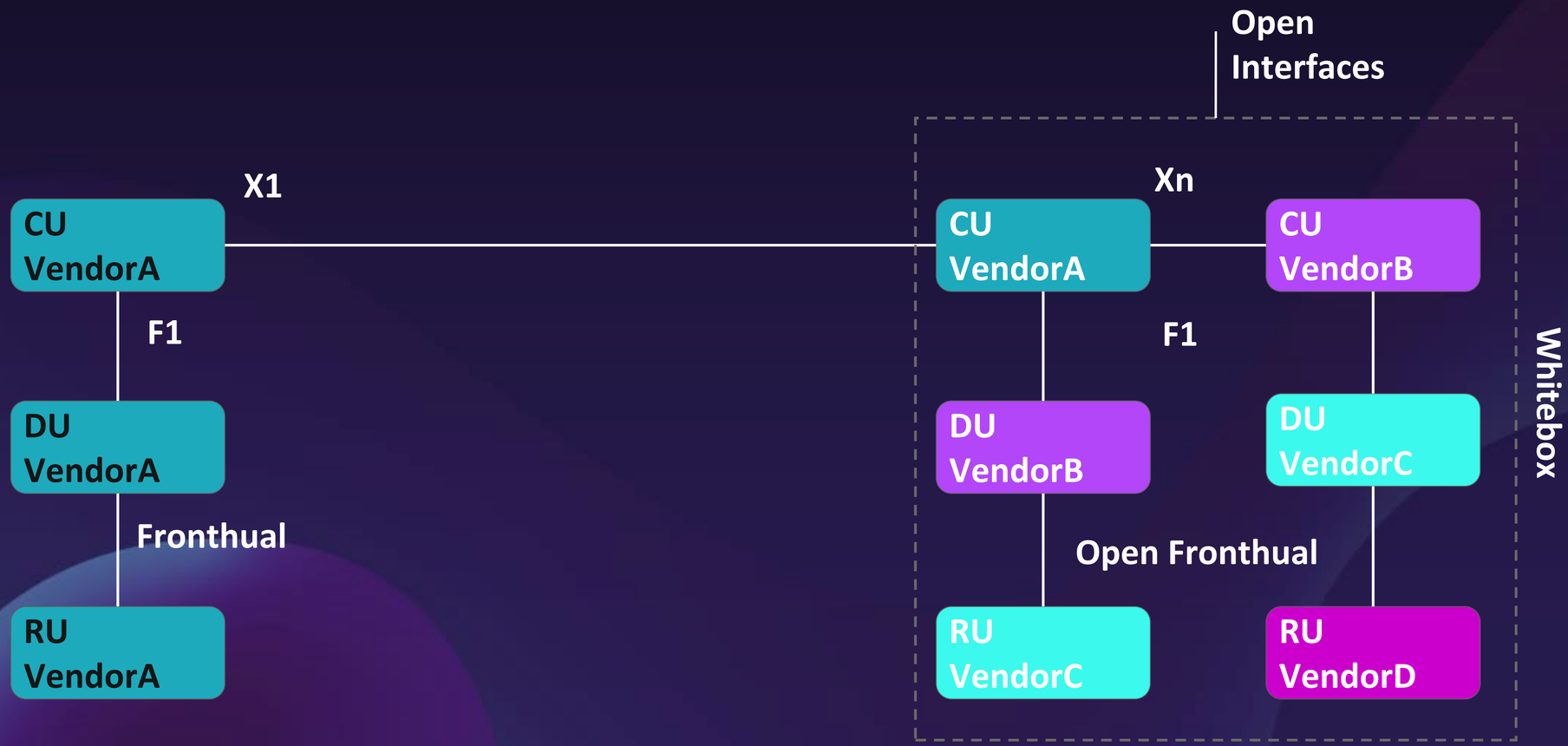
Static/vertical Market
Few Vendors
Proprietary/HW solution

Dynamic/Horizontal Market
Multitude of vendors
WhiteBox Open IF solution

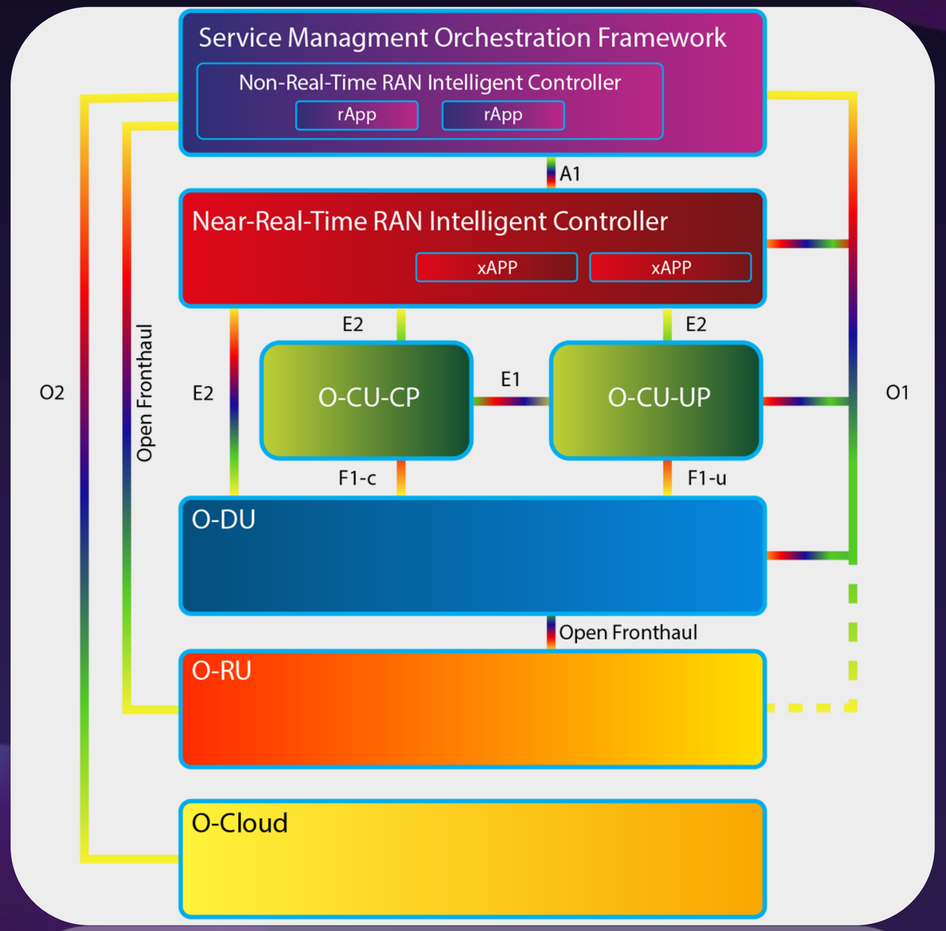
# Open RAN

**Traditional RAN**
Interfaces are vendor specific and non-interoperable with other vendors

**Open RAN**
**Open interfaces are standardized and interoperable among different vendors**



# O-RAN Architecture & Interfaces




O1: SMO to all

O2: SMO to O-Cloud

A1: Non-RT RIC to Neal-RT RIC

E1: O-CU UP and CP

E2: Neal-RT RIC to E2 Nodes

F1-C: O-CU CP to O-DU CP

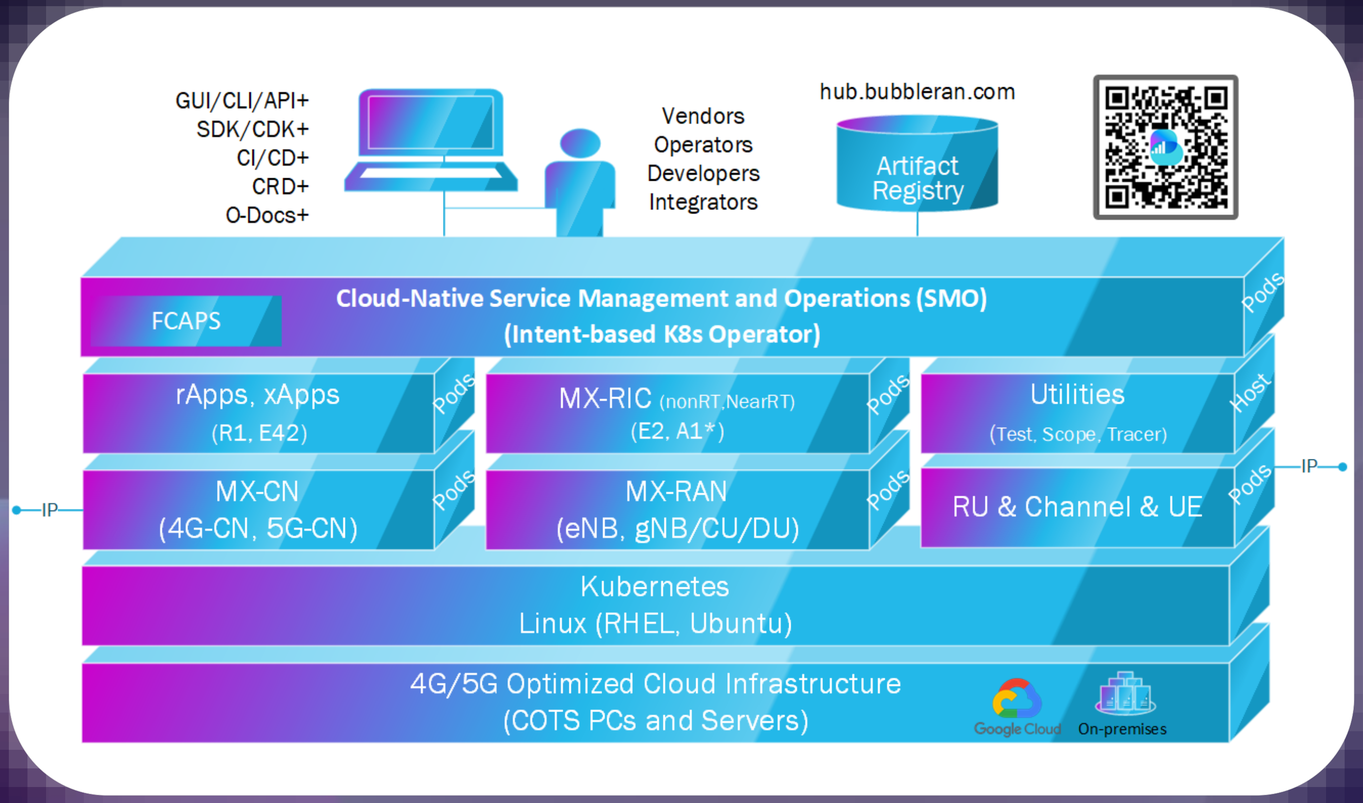
F1-U: O-CU UP to O-DU UP

Open Fronthaul(7.2): O-DU to O-RU

# Need for a realistic end-to-end 4G/5G O-RAN platform and ecosystem of xApps based on Open Source Components

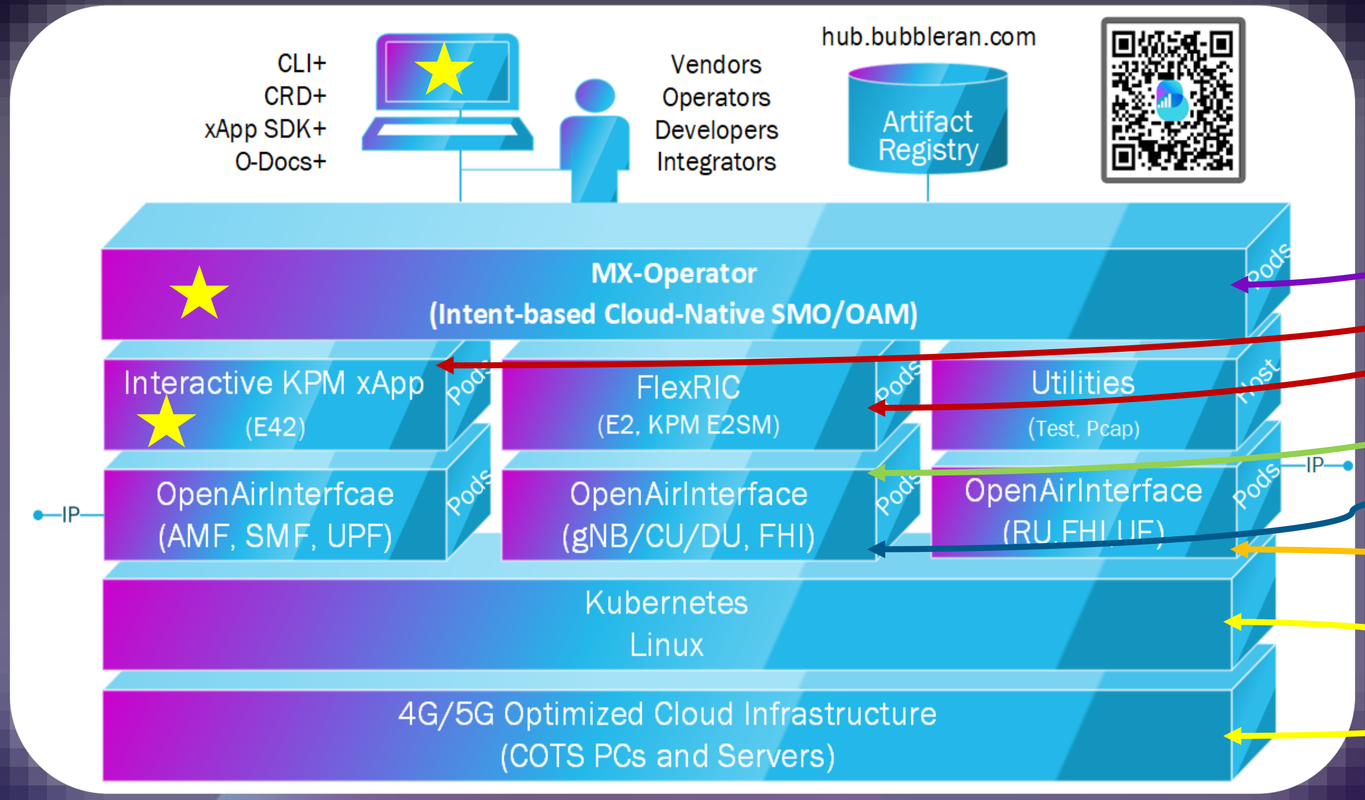
# What is Open RAN Studio (ORS)?

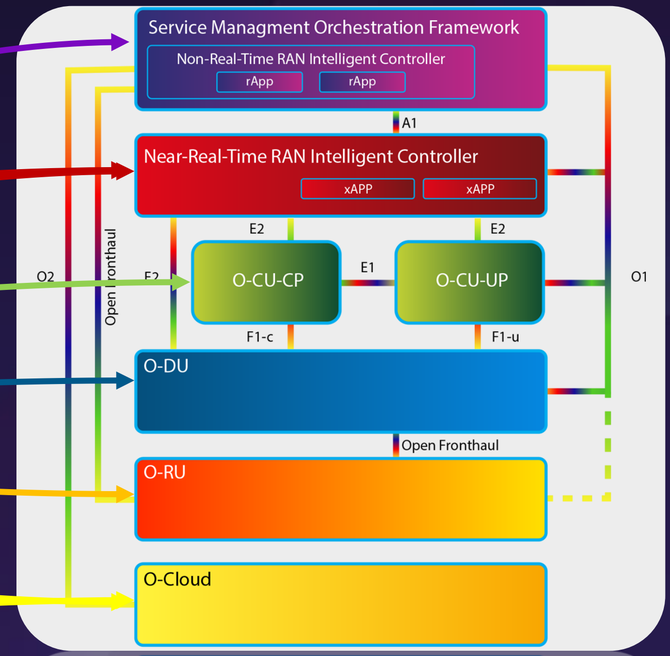
Open RAN Studio is **world-first** production-grade **cloud-native** platform to seamlessly **design, operate, experiment** an emulated **end-to-end** 3GPP & O-RAN standard-compliant network with edge services, **at scale**.



# Today's Focus ★

**End-to-end 5G SA and Open RAN deployment in the emulation mode with RU-CH-UE in the loop based on OpenAirInterface 5G stack and M5G FlexRIC**



# Open RAN Studio Capabilities


**Develop & Built**

+ xApp SDK
+ NF/xApps CDK

**Design Blueprint**

+ Network Topology
+ Deployment Model
+ Terminal Model
+ Slice Model

**Deploy & Customize**

+ Operations
+ Configurations
+ Models
+ NF/xApps

**Measure & Test E2E**

+ Resources
+ Network Readiness
+ Performance
+ Agility

**Analyze & Evaluate**

+ Control Plane
+ User-plane
+ Topology
+ Performance

**Developers, vendors, and Operators are able to design, deploy, test, collect, analyze an end-to-end 3GPP and O-RAN compliant network tailored to their use-case.**

# Agenda Continued

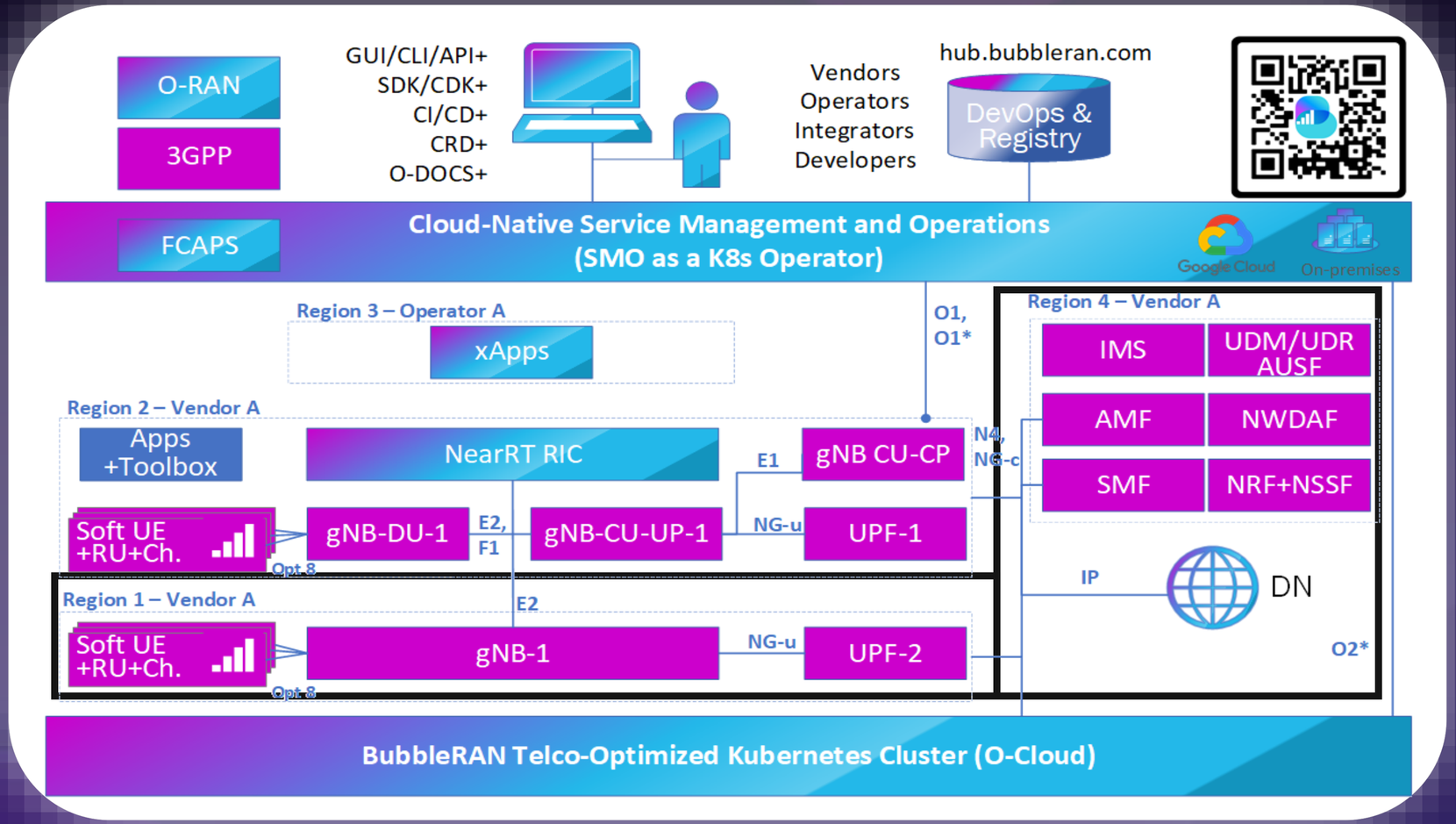
| | |
|---|---|
| 9:00 – 9:05 | Welcome speech (Ilias) |
| 9:05 – 9:15 | Why Open RAN Studio? (Navid) |
| 9:15 – 9:30 | Live deployment of a cloud-native 5G SA blueprint (Navid) |
| 9:30 – 9:35 | Q&A |
| 9:35 – 9:55 | Live deployment of a cloud-native 5G Open RAN blueprint (Alireza) |
| 9:55 – 10:10 | Interactive KPM and data collection xApp (ChiehChun) |
| 10:10 – 10:20 | How to develop an xApp? (Ilias) |
| 10:15 – 10:20 | Towards ML DevKit for xApps (Khoa) |
| 10:20 – 10:30 | Q&A and Take a way message (All) |

# 5G SA Deployment Steps

| Step | Operation | Description | Day |
|---|---|---|---|
| 0 | infra | Extract provisioned resources in the cluster including RUs | 0 |
| 1 | dlueprint | Design/describe an E2E network: slice, access, edge, core | 0 |
| 2 | install | Deploy the network blueprint | 1 |
| 3 | observe | Check the status and readiness of the deployed network | 1 |
| 4 | test | Measure/monitor the E2E performance | 2 |
| 5 | extract | Get information about network: log, config, relation | 2 |
| 6 | Remove | Delete the deployed network | 2 |
| 7 | cic | Embedded workload commands | 2 |
| 8 | upgrade | Roll out any operation: (re-)config, upgrade | 2 |
| 9 | Scale | Replication a service or a workload | 2 |
| 10 | slice | Create a 3GPP slice | 2 |

# Demo: 5G SA Deployment Blueprint

## Fully Emulated Environment






# 5min Break

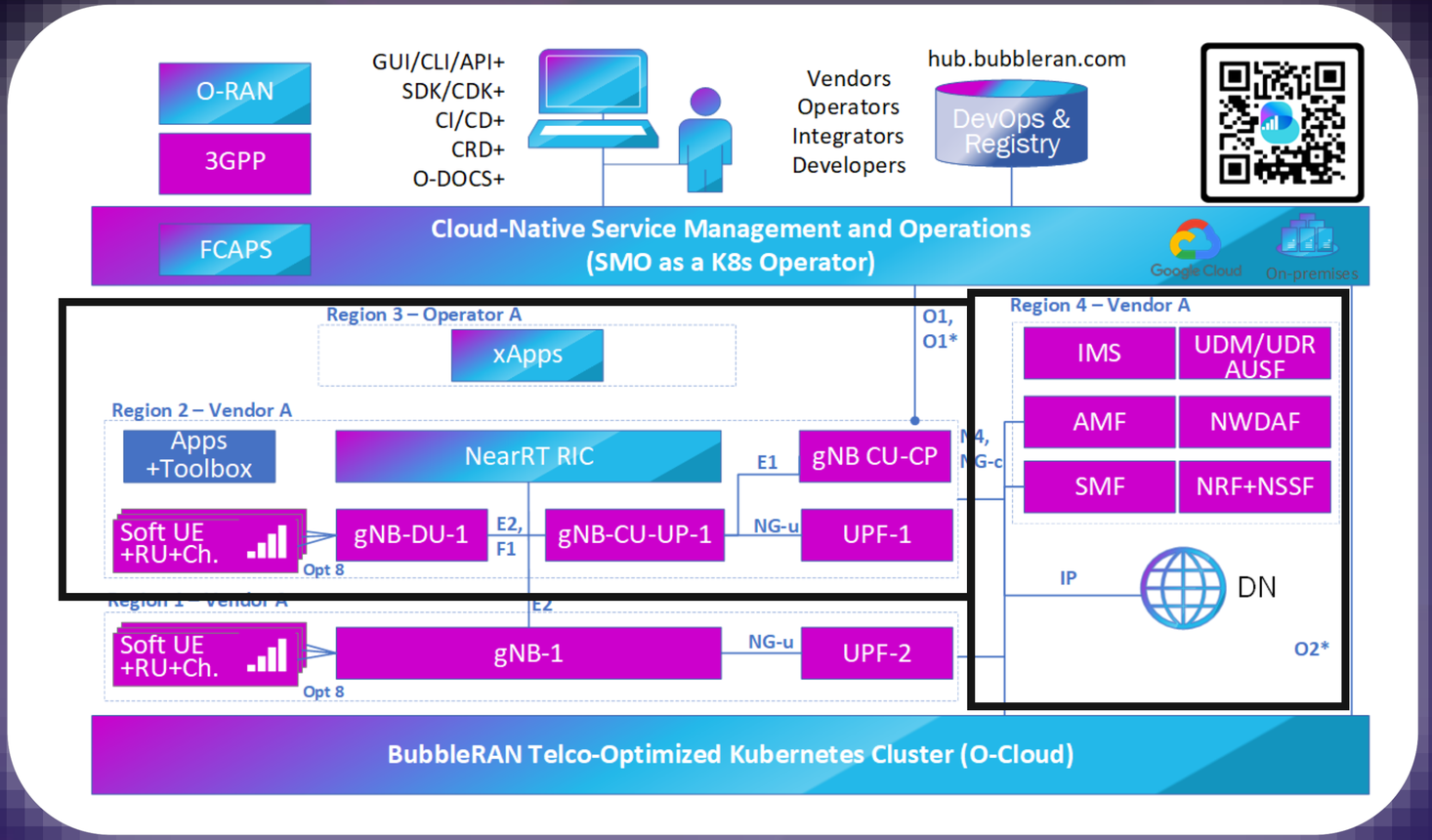



# Demo: 5G Open RAN/O-RAN  Deployment Blueprint

Fully Emulated Environment

## Benefits

- ✓ High level of realism
- ✓ Easy to setup, learn, and run
- ✓ Experiment 4G/5G, Open RAN, and cloud-native all together
- ✓ Data Collection & analysis
- ✓ Open Documentation and training



# Observations

- **Auto-resource discovery:** Resources of the infrastructure is automatically discovered and exposed to the user by a simple command.
- **Readiness:** Even with the agile deployment process, Network readiness indicates 99% service assurance, verifiable via observability.
- **Simplicity:** No expertise of cloud is required to Operate the network!
- **Automation:** Automated and agile lifecycle operations and controls
- **Scalability:** Scalable deployments in a distributed and heterogeneous environments
- **Zero down-time:** Reconfiguration on-the-fly across the whole deployment while retaining service continuity
- **UX:** Adherence to the current user experience by exposing native workload commands

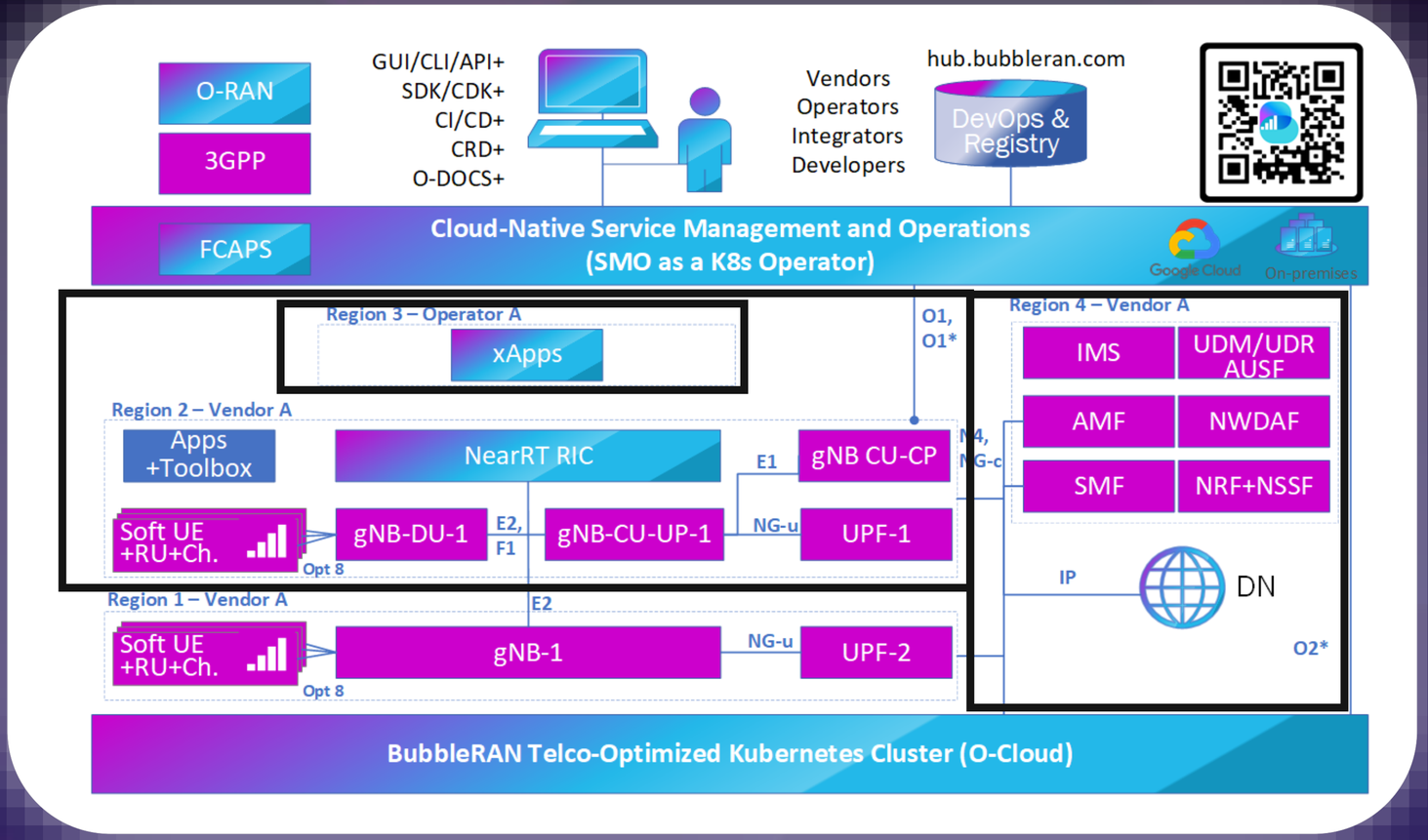
# Agenda Continued

| | |
|---|---|
| 9:00 – 9:05 | Welcome speech (Ilias) |
| 9:05 – 9:15 | Why Open RAN Studio? (Navid) |
| 9:15 – 9:30 | Live deployment of a cloud-native 5G SA blueprint (Navid) |
| 9:30 – 9:35 | Q&A |
| 9:35 – 9:55 | Live deployment of a cloud-native 5G Open RAN blueprint (Alireza) |
| 9:55 – 10:10 | Interactive KPM and data collection xApp (ChiehChun) |
| 10:10 – 10:20 | How to develop an xApp? (Ilias) |
| 10:15 – 10:20 | Towards ML DevKit for xApps (Khoa) |
| 10:20 – 10:30 | Q&A and Take a way message (All) |

# Demo: 5G Open RAN/O-RAN Deployment Blueprint

Fully Emulated Environment

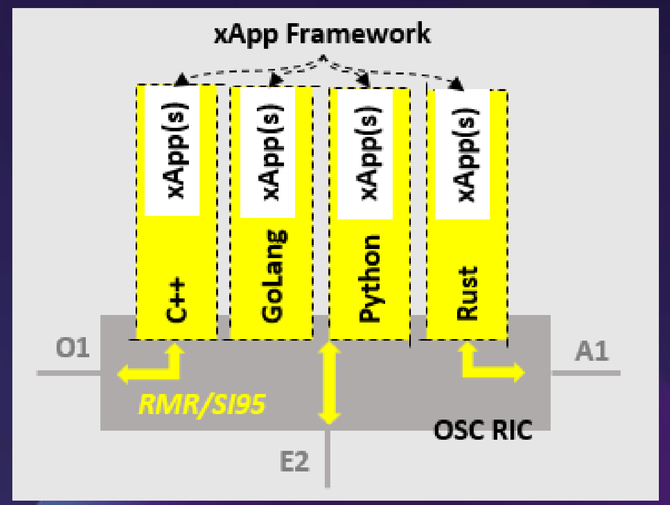
## Interactive KPM xApp

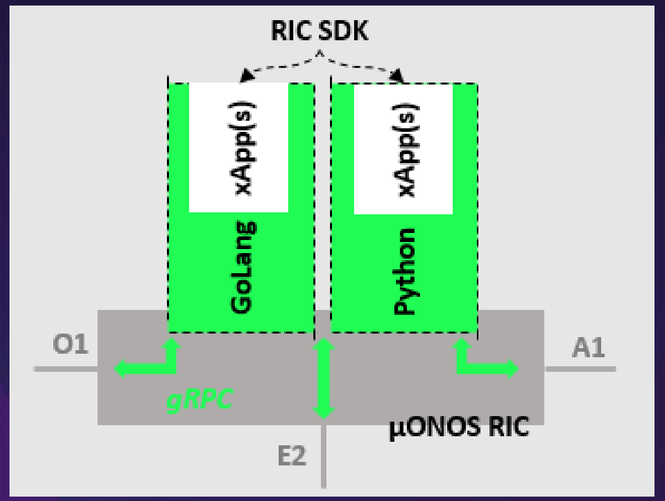


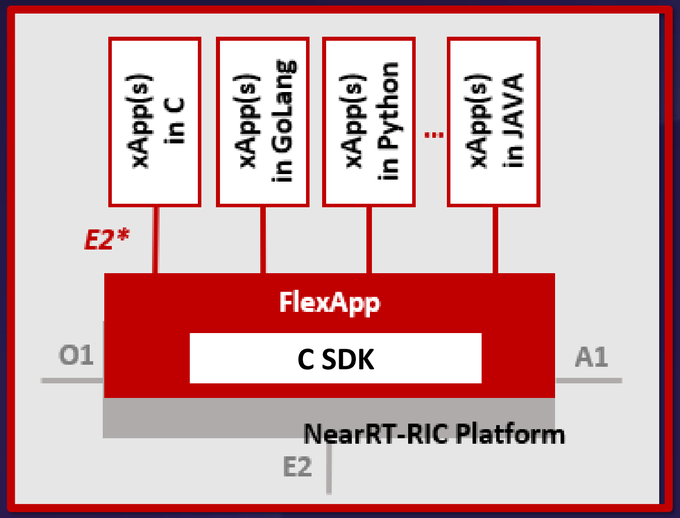
# xApp Environment in Open RAN Studio

| Criteria | OSC xApp framework | ONF RIC SDK | FlexApp |
|---|---|---|---|
| **xApp coupling with NearRT-RIC platform** | Embedded | Embedded | Isolated |
| **Service Model (SM) coupling with NearRT-RIC platform** | Tightly | Tightly | Loosely |
| **xApp baseline SDK** | Language-specific | Language-specific | Language-agnostic |
| **xApp programming languages** | 4 | 2 | 10+ |
| **Latency (1 xApp, 250us message rate)** | >200us | >200us | >50us |

Source: FlexApp paper

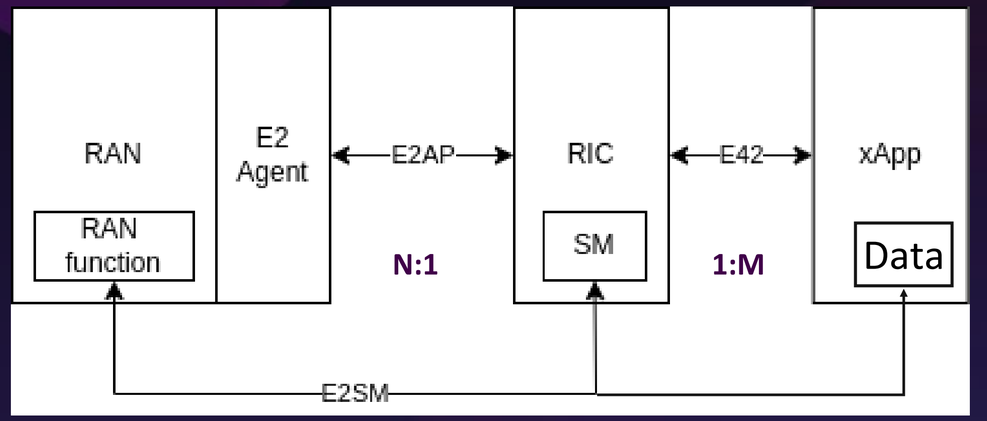


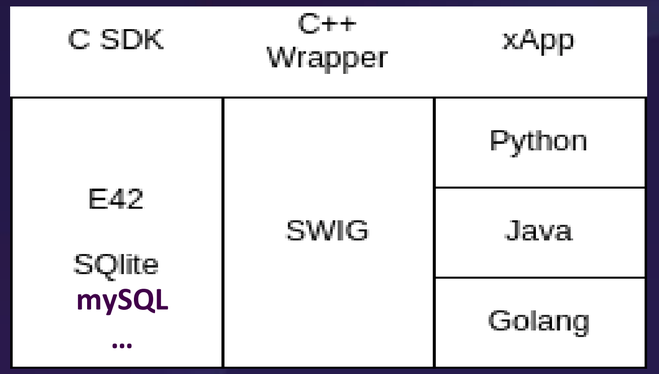



# xApp Internals

- **Interfaces**
  - **E2:** E2 Node to RIC, **N:1 relation**
  - **E2SM:** xApp to RAN function, **N:M Relation**
    - Custom: MAC, PDCP, RLC, NG;
    - ORAN: KPM, RC(Q42023), CCC(Q1/2 2024)
  - **E42:** RIC to xApp, **1:M relation**
- **Multi-Language**
  - xApp samples in C, C++, Python and Golang
  - More is coming…
    - SWIG could support Javascript, Perl, PHP, Python, Tcl, Ruby, C#, D, Go, Java, Android, Lua, OCaml, Octave, Scilab, R and more.

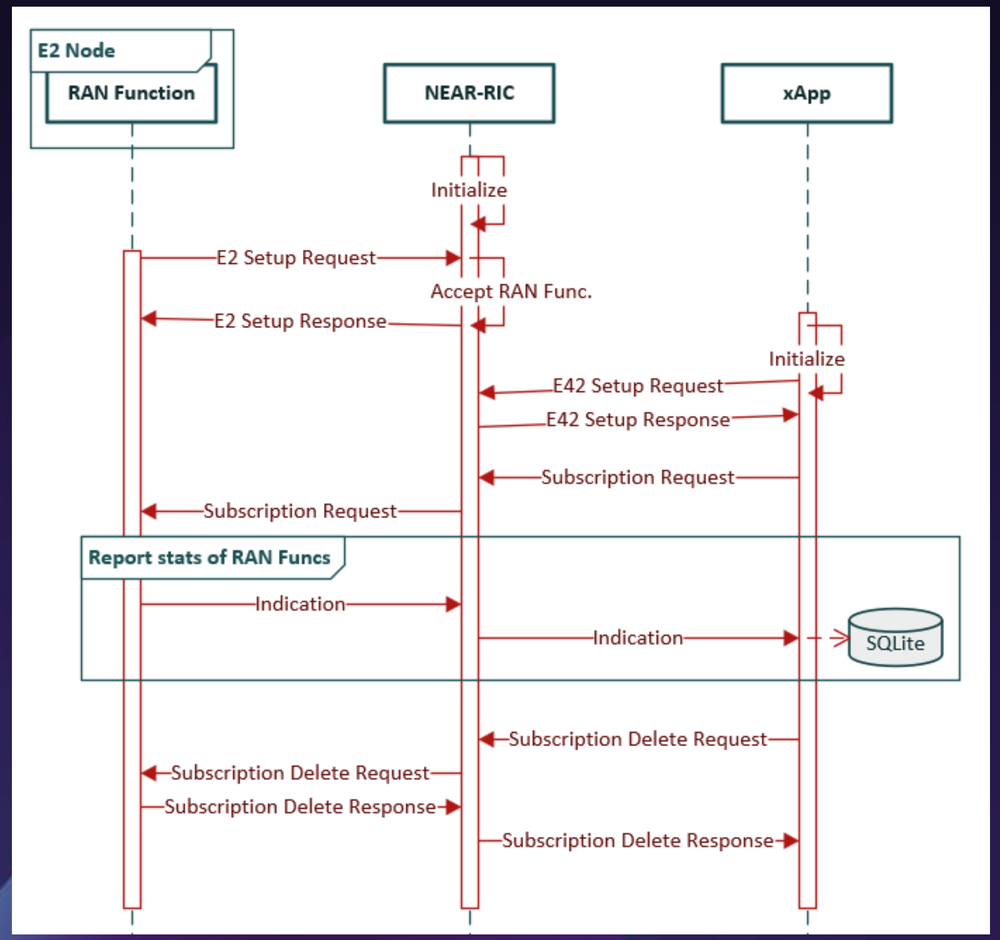


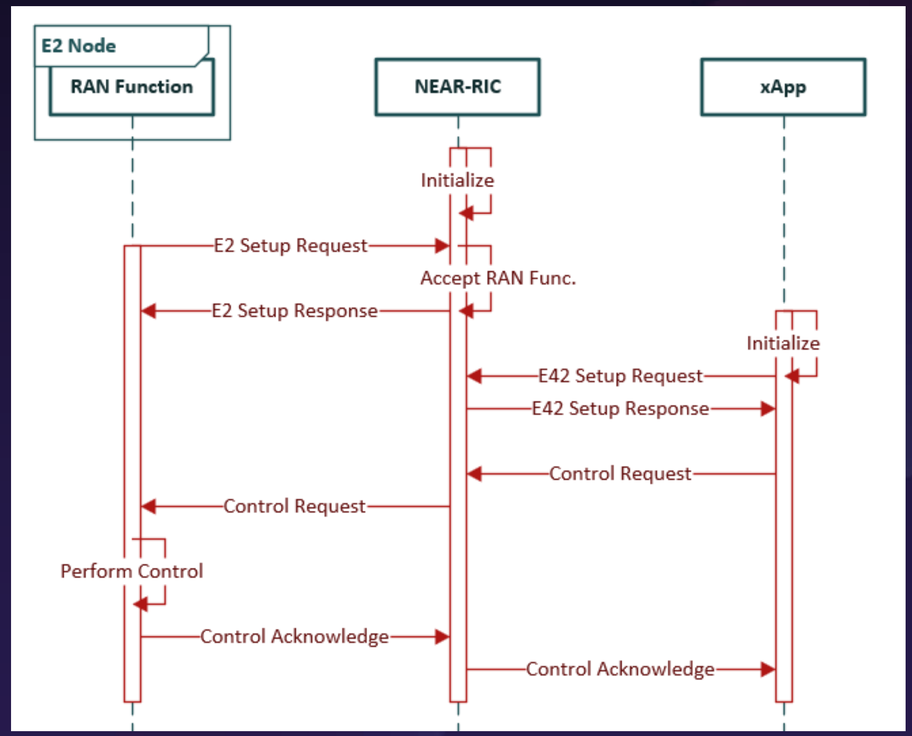
# xApp Internals

## Call Flows

### Report Service




### Control Service



# xApp Internals

## Sample code in **C**

1. **Init**
   - Get config and load the SM libraries
   - Init xApp
2. **Status**
   - Get State of connection of E2 Nodes
3. **Subscribe**
   - Subscribe to KPM Service model
   - Set the interval time of indication message
   - Set the `Callback`
4. **Receive**
   - the indication message in the `Callback`
5. **Process**
   - the data in the indication message





```c
int main(int argc, char *argv[])
{
  fr_args_t args = init_fr_args(argc, argv);

  //Init the xApp
  init_xapp_api(&args);
  sleep(1);

  e2_node_arr_t nodes = e2_nodes_xapp_api();
  defer({ free_e2_node_arr(&nodes); });

  assert(nodes.len > 0);

  printf("Connected E2 nodes = %d\n", nodes.len);

  // KPM indication
  inter_xapp_e i_0 = ms_5;
  sm_ans_xapp_t* kpm_handle = NULL;

  if(nodes.len > 0){
    kpm_handle = calloc( nodes.len, sizeof(sm_ans_xapp_t) );
    assert(kpm_handle  != NULL);
  }

  for (int i = 0; i < nodes.len; i++) {
    e2_node_connected_t* n = &nodes.n[i];
    for (size_t j = 0; j < n->len_rf; j++)
      printf("Registered node %d ran func id = %d \n ", i, n->ack_rf[j].id);

    kpm_handle[i] = report_sm_xapp_api(&nodes.n[i].id, SM_KPM_ID, i_0, sm_cb_kpm);
    assert(kpm_handle[i].success == true);
  }
static
void sm_cb_kpm(sm_ag_if_rd_t const* rd)
{
  assert(rd != NULL);
  assert(rd->type == KPM_STATS_V0);

  int64_t now = time_now_us();

  // KPM has 1 second resolution in its indication header, while 'now' is in microseconds
  int64_t diff = now/1000000 - (int64_t)rd->kpm_stats.hdr.collectStartTime;
  if (diff > 1)
    printf("KPM ind_msg latency = %lu seconds\n", diff);
  else
    printf("KPM ind_msg latency < 1 seconds\n");
}
```

Init xApp Config

Init xApp

Get the state of connection E2 nodes

Subscription request & Give Callback

# xApp Internals

## Sample code in **Python**

1. **Init**
   - Get config and load the SM libraries
   - Init xApp
2. **Status**
   - Get State of connection of E2 Nodes
3. **Subscribe**
   - Subscribe to NG/GTP Service model
   - Set the interval time of indication message
   - Set the `Callback`
4. **Receive**
   - the indication message in the `Callback`
5. **Process**
   - the data in the indication message

```
####################
####  GENERAL
####################

ric.init()

conn = ric.conn_e2_nodes()
assert(len(conn) > 0)
for i in range(0, len(conn)):
    print("Global E2 Node [" + str(i) + "]: PLMN MCC = " + str(conn[i].id.plmn.mcc))
    print("Global E2 Node [" + str(i) + "]: PLMN MNC = " + str(conn[i].id.plmn.mnc))

####################
#### GTP INDICATION
####################

gtp_hndlr = []

for i in range(0, len(conn)):
    gtp_cb = GTPCallback()
    hndlr = ric.report_gtp_sm(conn[i].id, ric.Interval_ms_1, gtp_cb)
    gtp_hndlr.append(hndlr)
    time.sleep(1)


time.sleep(10)

### End

for i in range(0, len(gtp_hndlr)):
    ric.rm_report_gtp_sm(gtp_hndlr[i])

# Avoid deadlock. ToDo revise architecture
while ric.try_stop == 0:
    time.sleep(1)
# Create a callback for GTP which derived it from C++ class gtp_cb
class GTPCallback(ric.gtp_cb):
    def __init__(self):
        # Inherit C++ gtp_cb class
        ric.gtp_cb.__init__(self)
    # Create an override C++ method
    def handle(self, ind):
        if len(ind.gtp_stats) > 0:
            t_now = time.time_ns() / 1000.0
            t_gtp = ind.tstamp / 1.0
            t_diff = t_now - t_gtp
            print('GTP Indication tstamp = ' + str(ind.tstamp) + ' diff = ' + str(t_diff))
```

Init xApp

Get the state of connection E2 nodes

Subscription request & Give Callback

# Towards ML DevKit for xApps

Coming soon!

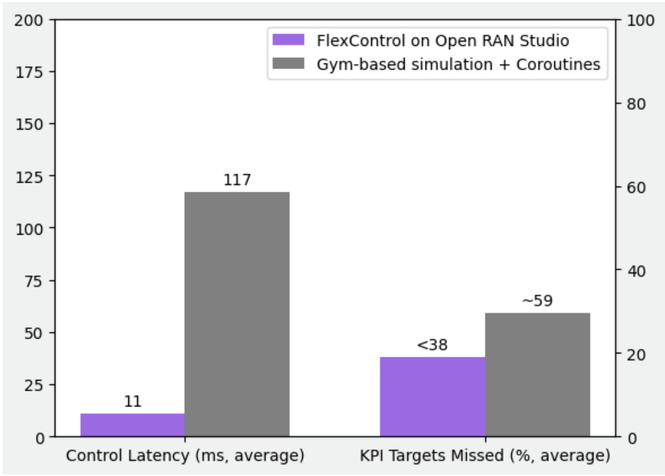
## FlexControl MLOps Framework

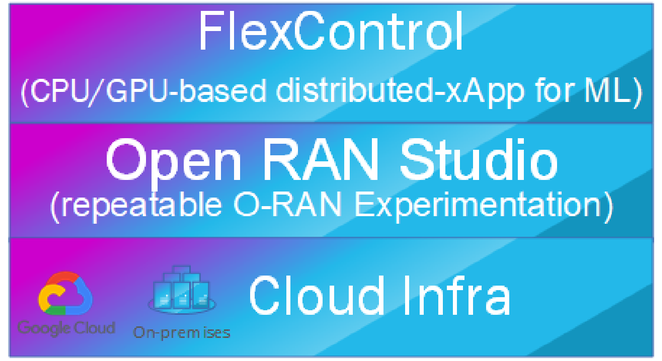
"A developer-friendly way to bring O-RAN-compliant ML into Production."

- ✓ Bridging the Sim2Real gap.
- ✓ Near-RT execution of Online ML.
- ✓ Zero-overhead instead of KubeFlow.

**News:** FlexControl "1-2-3" at EUCNC 2023

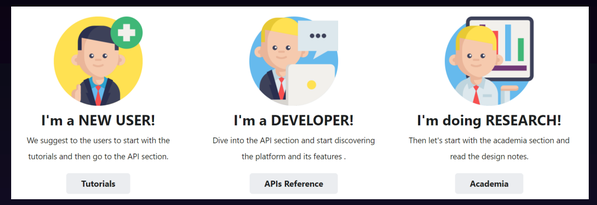
- ✓ Simulation artifacts for Factory Private-Net
- ✓ Plug-and-play fast "Python" components
- ✓ Running MLOps with a single LOC



# Agenda Continued

| | |
|---|---|
| 9:00 – 9:05 | Welcome speech (Ilias) |
| 9:05 – 9:15 | Why Open RAN Studio? (Navid) |
| 9:15 – 9:30 | Live deployment of a cloud-native 5G SA blueprint (Navid) |
| 9:30 – 9:35 | Q&A |
| 9:35 – 9:55 | Live deployment of a cloud-native 5G Open RAN blueprint (Alireza) |
| 9:55 – 10:10 | Interactive KPM and data collection xApp (ChiehChun) |
| 10:10 – 10:20 | How to develop an xApp? (Ilias) |
| 10:15 – 10:20 | Towards ML DevKit for xApps (Khoa) |
| **10:20 – 10:30** | **Q&A and Take a way message (All)** |




# Benefits of Open RAN Studio


https://bubbleran.com/docs/


1. Develop NFs/xApps  or extend the existing one and on-board them using the provided software development kit (SDK) and container-development kit (CDK)
2. Design an end-to-end 4G/5G Open RAN network blueprints, including UE, RAN, CORE, and EDGE elemets, tailored to your use-case
3. Deploy and operate their blueprints at scale
4. Control and reconfigure RAN via xApps/rApps
5. Test and measure network performance
6. Collect datasets and analyze network control and user planes.
7. Online Open Documentation

# Open RAN Studio

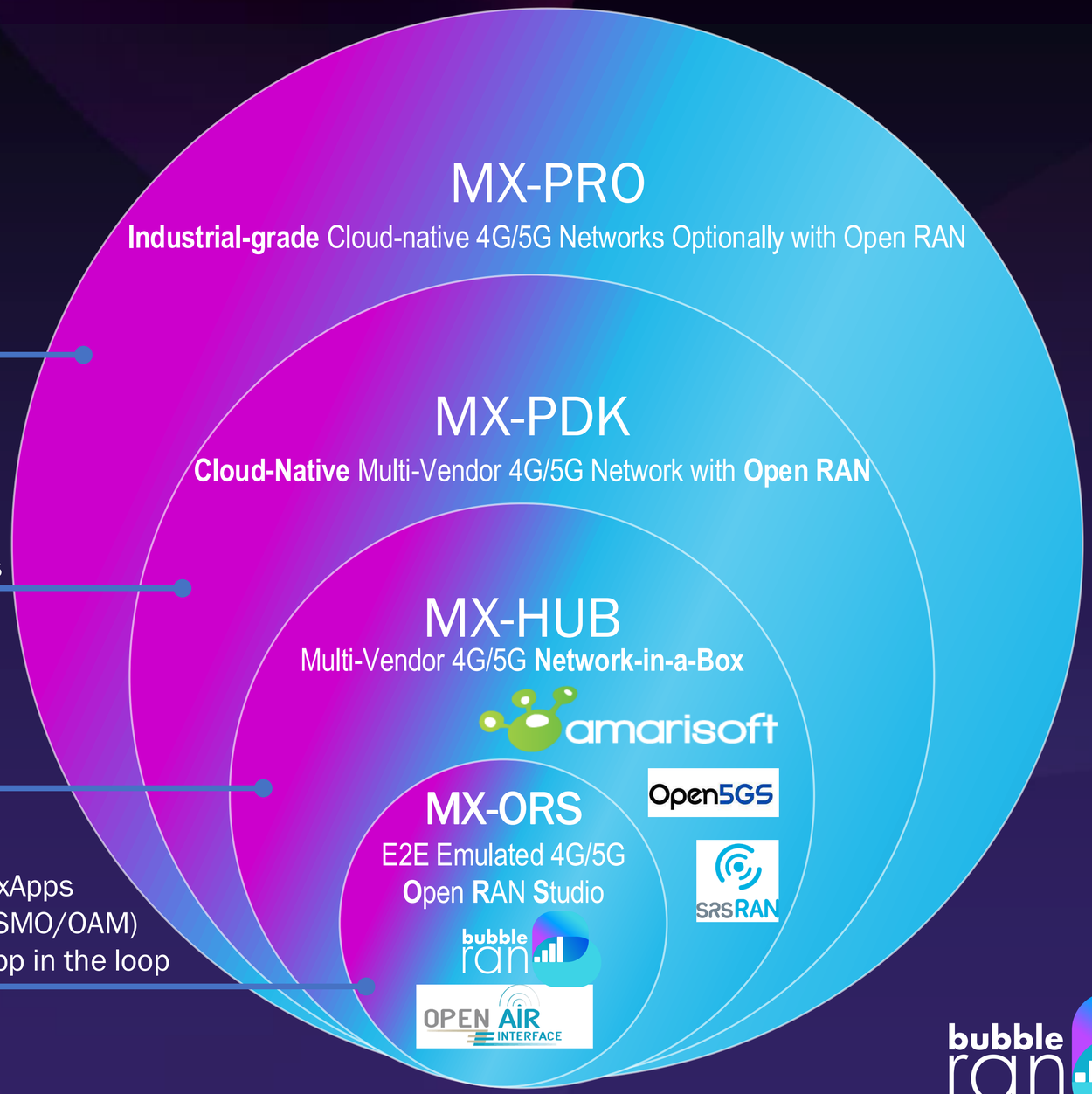
## Relation to other Products

+Performance Guarantee
+High Reliability & Availability
+Network Synchronization

+Full Automation
+Observability Stack
+Open RAN/O-RAN Stack
+Large-scale deployments

+Multi-Vendor
+Edge Services
+Multi-Bakchauling

Near-RT RIC/xApps
Automation (SMO/OAM)
Terminal & App in the loop



# Open RAN 2.0

| | |
|---|---|
| Open RAN Vendor lock-in/Immaturity | Cloud-Native & Open Ecosystem |
| Manual/Slow On-Boarding/Delivery | Automation/CI/CD/DevOps |
| Manual/Slow Operations | Intelligent Automated Operations |
| Manual Processes | Self-Services Processes |

- Definition evolved

  Transforming RAN towards Open, Intelligent, Cloud-Native, and fully interoperable multi-vendor RAN for public-private networks/UCs via a Open Ecosystem and Market

+336 49 92 63 45
@BubbleRANTech
contact@bubbleran.com
@Bubbleran

# Thanks for joining us Today


**Next Episode**

**Tentative Topic: xApps From Scratch**

**Tentative Date: 17/07**

# BubbleRAN O-Docs

Operate. Develop. Learn.

Operator, integrator, Service Provider

## I'm a NEW USER!

We suggest to the users to start with the tutorials and then go to the API section.

**Tutorials**

Vendor, Developer, integrator

## I'm a DEVELOPER!

Dive into the API section and start discovering the platform and its features .

**APIs Reference**

All

## I'm doing RESEARCH!

Then let's start with the academia section and read the design notes.

**Academia**

https://bubbleran.com/docs/

# Follow us on

Please follow us on Linkedin and activate notifications as the subsequent events will be announce there

https://www.linkedin.com/company/bubbleran