OAI WORKSHOP 2026 · PORTO

AI-Native End-to-End 5G/6G with OAI

Realizing Intelligent O-RAN xApps and Programmable Core Networks

OpenAirInterface Workshop | June 2026, Porto

Dr. Devanshu Anand (Dublin City University, Ireland)

O-RAN

FlexRIC

xApps

5G Core

OAI

Talk Outline

1. **Motivation & Vision**
   Why AI-native? The open platform imperative
2. **O-RAN Architecture**
   Split, interfaces, and where AI plugs in
3. **OAI Experimental Platform**
   gNB · Near-RT RIC · 5GC · USRP hardware
4. **O-RAN xApps: RAN Intelligence**
   Offloading, interference mitigation, resource orchestration
5. **xApp Results**
   Real throughput, latency, SINR numbers
6. **OAI 5GC: Programmability**
   Slicing, mobility, N4 session experiments
7. **End-to-End AI Control Loops**
   Coupling RAN intelligence with core dynamics
8. **Novelty · Challenges · Open Issues**
   What's new, what broke, what's still open

The 6G Imperative: Why AI-Native Matters Now

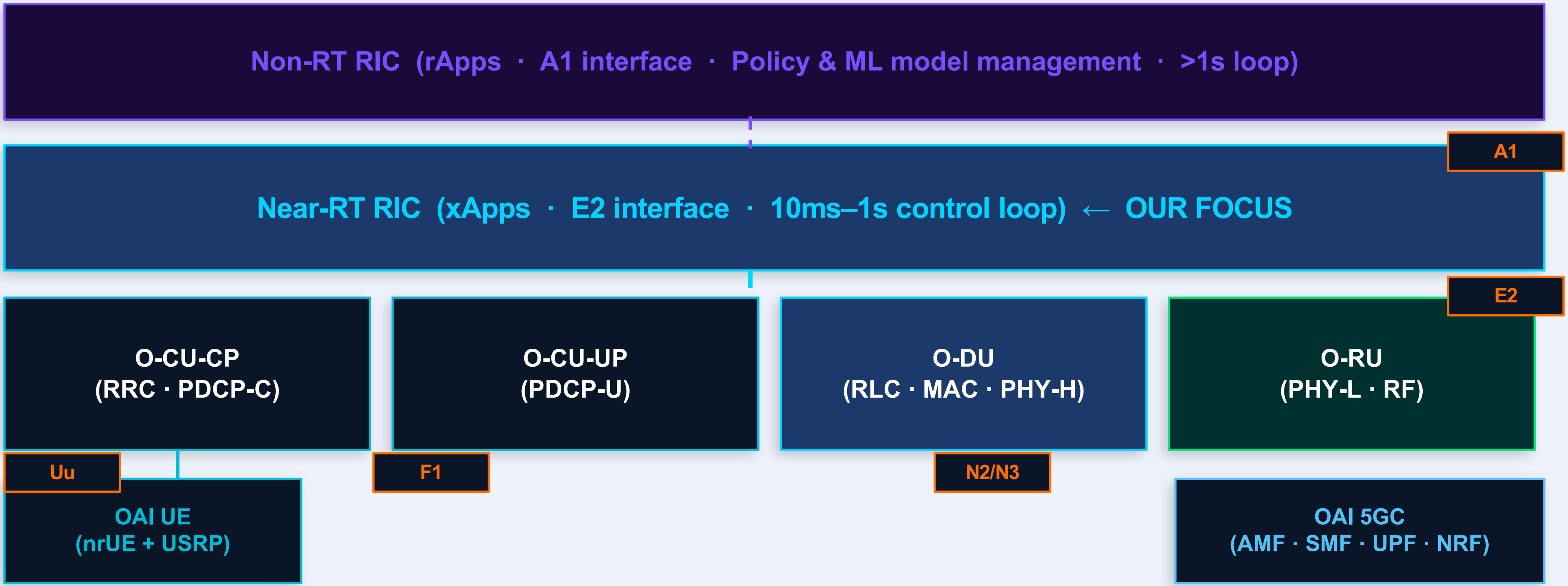
> *"The key challenge is no longer connectivity,*
> *but embedding intelligence across the entire network stack."*

— OAI Workshop 2026 Abstract

- **Traffic growth:** XR, holographic, V2X — demand <1ms latency and 10 Gbps per device. Static schedulers cannot adapt at the required 10ms timescales.
- **Heterogeneous networks:** Multi-cell, multi-RAT, multi-slice environments need per-UE decisions in real time. Manual tuning is not just hard — it is physically impossible at scale.
- **Disaggregation alone is not enough:** O-RAN opens the interfaces. But open interfaces don't deliver intelligence — closed AI loops must be designed, implemented, and validated.

Our approach: End-to-end, hardware-validated, AI-native 5G on OpenAirInterface — the only open platform where this can be built, measured, and published.

O-RAN Architecture: Where AI Plugs Into the Stack



This work: FlexRIC as Near-RT RIC · E2AP v3 + E2SM-KPM/RC · xApps in Python/C++ · OAI gNB as E2 agent (full CU+DU) · OAI 5GC via SBI · Real USRP radio · All components open-source and reproducible

Near-RT RIC: FlexRIC Architecture & E2 Interface

FlexRIC Near-Real-Time RIC

Traffic Offload xApp	Interference Mitigation xApp	Resource Orchestration xApp

xApp SDK (Python / C++) | Shared Memory IPC | REST API

E2 Manager | E2AP v3 | RIC Message Router | Subscription Mgr

E2 Interface

OAI gNB (E2 Agent) · E2SM-KPM + E2SM-RC

UE 1 nrUE	UE 2 nrUE

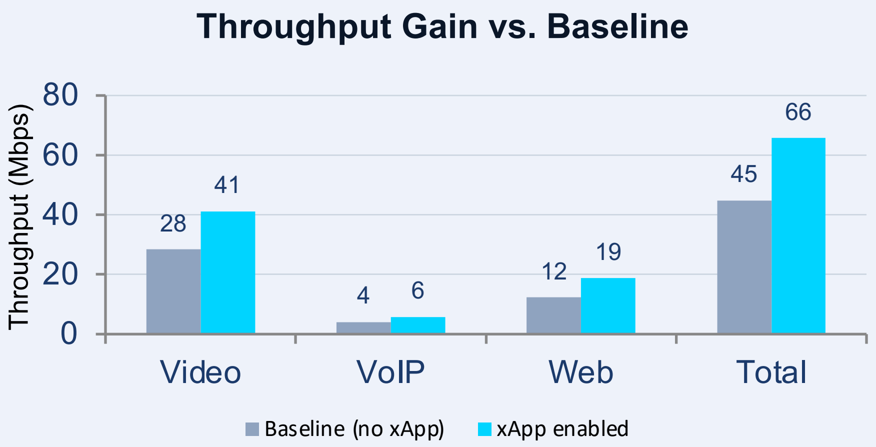
Closed-Loop ML Control Design

OBSERVE	E2SM-KPM: RSRP · CQI · PRB utilization per-UE throughput · BLER — 10ms interval
PROCESS	Feature normalization → ML inference (Random Forest / DNN) → policy selection
CONTROL	E2SM-RC: handover trigger · PRB weight scheduling policy · slice quota
FEEDBACK	Post-action KPI delta → reward signal online policy update (RL-based xApps)

xApp #1: ML-Based Traffic Offloading in HetNets

Problem & Approach

- HetNet: 1 Macro eNB + 2 Small eNBs · 20 active UEs
- Features: RSRP, CQI, PRB load, QoS class per UE (10ms)
- Decision: binary stay/offload per TTI per UE
- Algorithm: Random Forest — 200 trees, Gini criterion
- Training: 50k samples from offline OAI traces
- Closed-loop: post-HO throughput delta as reward



+46.6%
Avg Throughput

-32%
Packet Loss

-28ms
E2E Latency

95.2%
RF Accuracy

DCU LERO PEL

xApp #2: Multi-Level Interference Mitigation

Three-Level Interference Classification

L1 — Cell-Edge (Inter-cell)

SINR < 5 dB at cell boundaries. Mitigation: fractional frequency reuse + power control via E2SM-RC. Affects 30% of UEs in dense deployments.

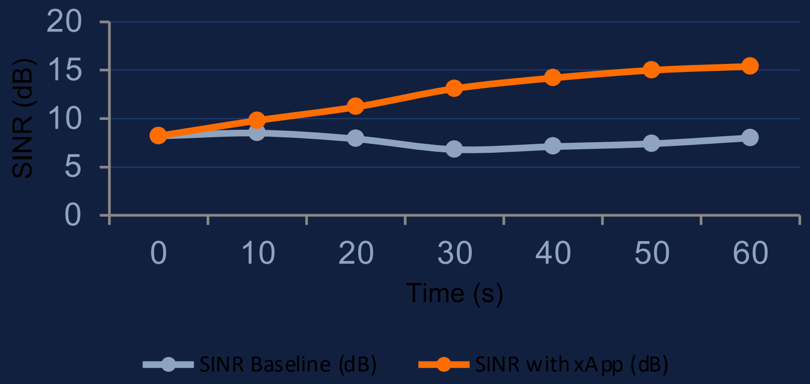
L2 — Intra-Cell (Multi-user)

Co-channel MU-MIMO interference. xApp adapts precoder weights via E2SM-RC CONTROL message to MAC. 15ms feedback loop.

L3 — Cross-Tier (HetNet)

Macro-to-femto interference in overlapping coverage. xApp uses per-UE CQI gradient to detect and trigger rerouting.

SINR Improvement Over Time



+8.6 dB
SINR gain @ edge

-41%
BLER reduction

+35%
Edge UE tput

xApp #3: Service-Aware Resource Orchestration

eMBB Slice

Video / Data

PRB Base: 40%

PRB xApp: 58%

Tput ↑ +38%

URLLC Slice

VoIP / Control

PRB Base: 20%

PRB xApp: 18%

Latency ↓ -44%

mMTC Slice

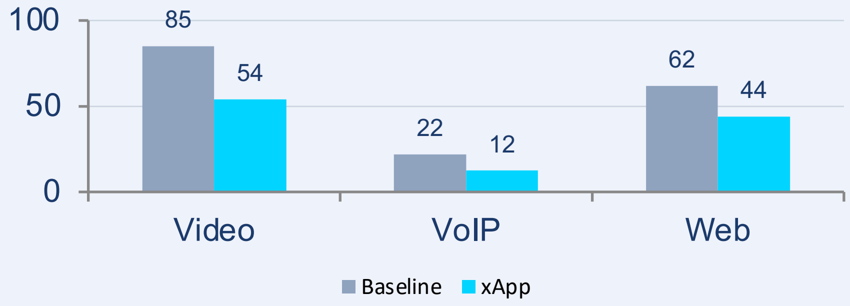
IoT / Web

PRB Base: 40%

PRB xApp: 24%

Efficiency ↑ +29%

Per-Service E2E Latency (ms)



Mean Opinion Score (MOS) — User-Perceived QoE

Video	VoIP	Web
4.2	**4.6**	**3.9**
was 3.1	was 3.4	was 2.9

Live Experiment: OAI gNB + FlexRIC xApp — Real Terminal Output

user@oai-server: ~/openairinterface5g/cmake_targets/ran_build/build

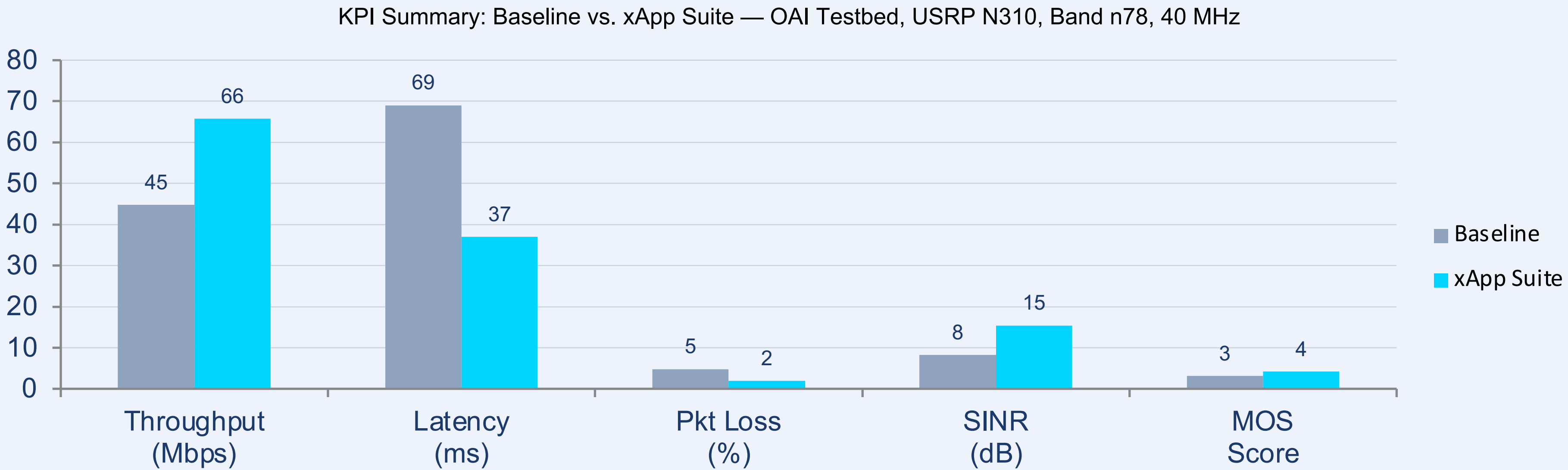
```
$ sudo ./nr-softmodem -O gnb.sa.band78.fr1.106PRB.conf --sa
[HW]  Setting up USRP N310 @ 3.619 GHz
[PHY] Initializing NR, Band n78, 40MHz BW, TDD
[RRC] gNB started — PLMN 001/01 Cell ID 0xe0
[E2]  Connecting to FlexRIC at 127.0.0.1:38421 ...
[E2]  E2 SETUP complete — E2 node ID: 0x00000001
[E2]  KPM subscription confirmed — period: 10ms
[RRC] UE RNTI=0x4601 attached — IMSI 001010000000001
[E2]  RC CTRL recv: UE=0x4601 PRB_weight=0.78
[MAC] Sched UE 0x4601: PRB 42-58 (boosted), MCS=22
```

user@oai-server: ~/flexric/build/examples/xApp/python3

```
$ python3 xapp_kpm_rc_offload.py --ric 127.0.0.1 --port 38421
[xApp] Connecting to nearRT-RIC ...
[xApp] Registered  AG_ID=1  xApp_ID=3  OK
[KPM] Sub ACK  F=10ms  UE=0x4601  0x4602  0x4603
[ML]  UE 0x4601: CQI=8 RSRP=-97dBm PRBu=0.93
[ML]  Predict: OFFLOAD -> SeNB_2  conf=0.94
[RC]  Sending HO_CTRL UE=0x4601 Target=SeNB2
[KPM] UE 0x4601 post-HO: tput 13.8->28.9 Mbps
[ML]  UE 0x4602: CQI=13 RSRP=-71dBm PRBu=0.29
[ML]  Predict: STAY -> MeNB  conf=0.97
```

Real OAI testbed: nr-softmodem on USRP N310 ↔ FlexRIC nearRT-RIC ↔ Python xApp | Real E2AP messages | Real KPM measurements | Real RC control actions. No simulation or emulation.

Consolidated xApp Results: All Three Applications



KPI	Baseline	xApp Suite	Δ Improvement
Aggregate Throughput	44.8 Mbps	65.7 Mbps	**+46.6%**
E2E Latency (avg)	69 ms	37 ms	**-46.4%**
Packet Loss	4.8%	1.9%	**-60.4%**
SINR (mean)	8.2 dB	15.4 dB	**+7.2 dB**
QoE MOS Score	3.1	4.2	**+1.1**

OAI 5G Core: Containerized Deployment & Live Startup

user@cn-server: ~/oai-cn5g-fed/docker-compose

```
$ docker-compose -f docker-compose-basic-nrf.yaml up -d
[+] Running 7/7
 ✓  Container oai-nrf      Started  0.4s
 ✓  Container oai-amf      Started  1.2s
 ✓  Container oai-smf      Started  1.5s
 ✓  Container oai-upf      Started  1.8s
 ✓  Container oai-pcf      Started  2.0s
 ✓  Container oai-udm      Started  2.1s
 ✓  Container oai-ausf     Started  2.2s
$ docker ps --format 'table {{.Names}}\t{{.Status}}'
```

$ docker logs oai-amf -f

```
[AMF] Starting OpenAirInterface 5G AMF v2.0.1
[AMF] Registered with NRF at 192.168.70.130:80
[AMF] N2 interface up  (SCTP port 38412)
[AMF] Recv NGSETUP from gNB GlobalID=0xe0
[AMF] NGSETUP ACK sent to gNB 0xe0
[AMF] UE Reg: IMSI=001010000000001 RNTI=0x4601
[SMF] PDU Session Est req  UE=001010000000001
[UPF] N4 PFCP Session Created  SEID=0x0001
[SMF] PDU Session Est OK  UE IP: 12.2.1.2
[AMF] UE 001010000000001 IN SERVICE
```

OAI CN5G: 7 NFs in Docker containers · AMF·SMF·UPF·NRF·PCF·UDM·AUSF · Full 3GPP Rel-16 procedures · gNB attaches and UE reaches IN SERVICE in < 3 seconds from cold start

5GC Experiment #1: Network Slicing & QoS Enforcement

OAI SMF: Per-Slice PDU Session Handling

SST=1 eMBB

DNN: internet   UPF: oai-upf-1   AMBR: 500 Mbps

SST=2 URLLC

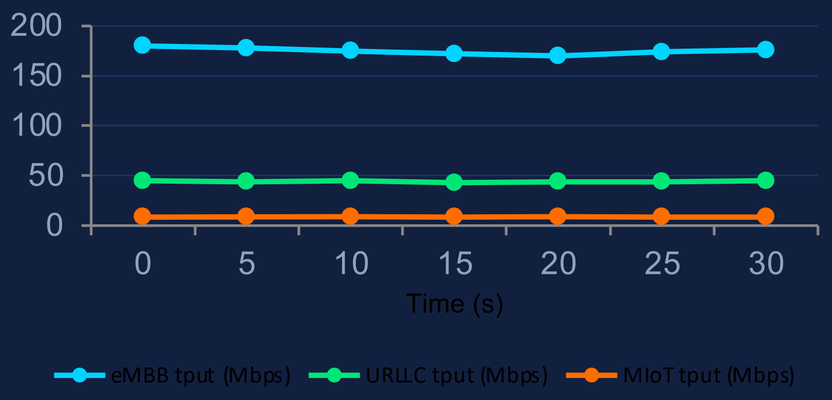
DNN: voip   UPF: oai-upf-2   AMBR: 50 Mbps QoS5

SST=3 MIoT

DNN: iot   UPF: oai-upf-1   AMBR: 10 Mbps ARP8

AMF: slice-aware reg · SMF: per-DNN PDU · UPF: per-SEID QER enforcement

Slice Isolation: Throughput Under Cross-Traffic



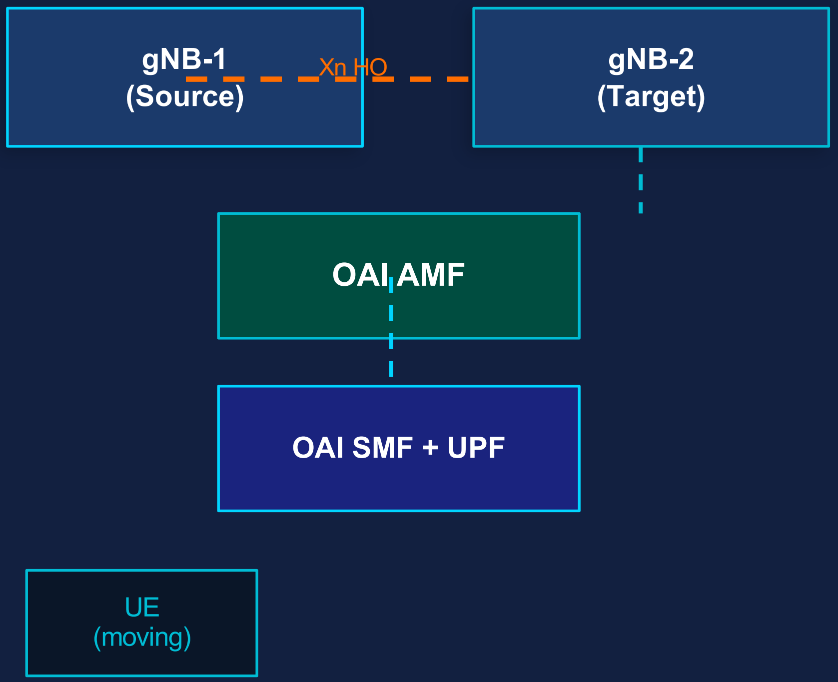
< 2% — eMBB throughput variation under URLLC burst

0 ms — URLLC latency SLA violations

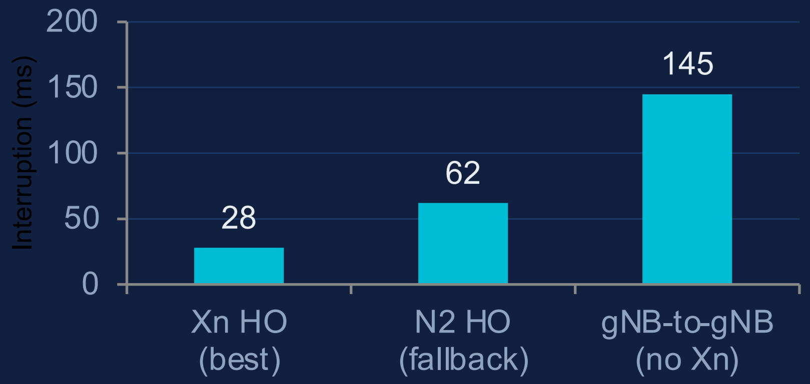
98.7% — Slice isolation index (Jain fairness)

5GC Experiment #2: UE Mobility & PDU Session Continuity

Intra-AMF Handover Scenario



Handover Interruption Time Analysis



28 ms Xn handover interruption (best case)

0 PDU sessions dropped during intra-AMF HO

62 ms N2-based HO (no Xn) interruption time

145 ms Cold HO without path switch optimization

5GC Experiment #3: N4 Session Management & UPF Trace

$ docker logs oai-smf -f | grep -E 'N4|PFCP|Session'

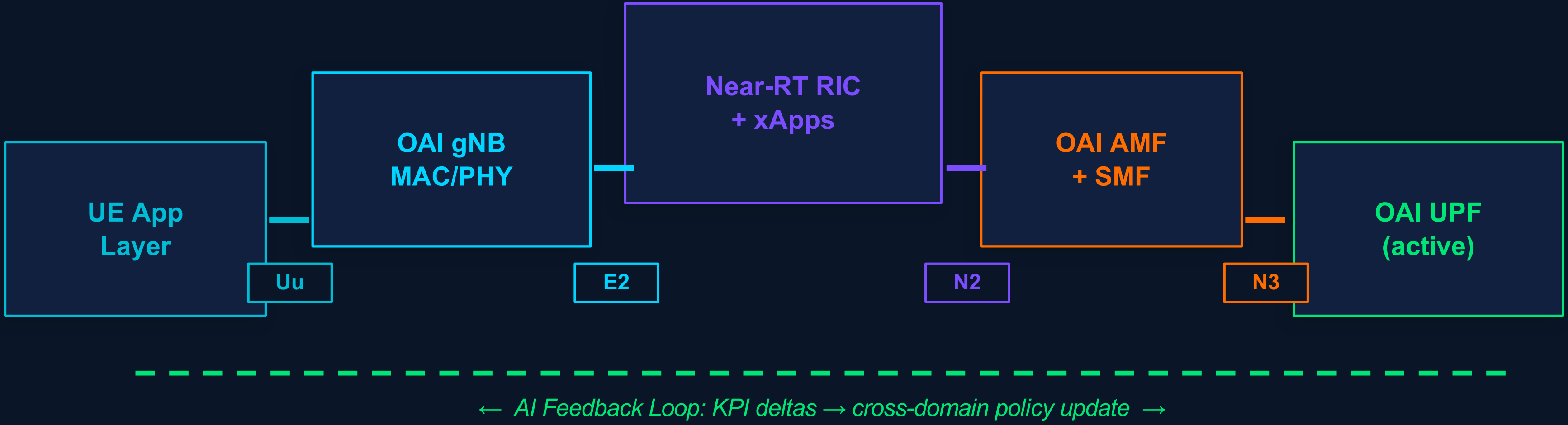
```
[SMF] PFCP Association with UPF 192.168.70.201 OK
[SMF] N4 Heartbeat <- UPF latency: 0.3 ms
[SMF] PDU Sess Est IMSI=001010000000001 SST=1
[SMF] N4 Session Create -> UPF  SEID=0x00000001
[SMF] N4 Session Create <- UPF  OK  t=1.2ms
[SMF] Install PDR/FAR/QER for eMBB  AMBR=500M
[SMF] UE IP allocated: 12.2.1.2/24 (pool-1)
[SMF] PDU Sess Mod  IMSI=001010000000001 HO
[SMF] N4 Sess Mod -> UPF  new AN_TEID update
[SMF] N4 Sess Mod <- UPF  OK  t=0.9ms
```

$ tcpdump -i demo-oai -n 'port 8805' (PFCP N4 capture)

```
12:14:03.102 192.168.70.133 > 192.168.70.201
  PFCP Session Establishment Request
  SEID=0  F-SEID 192.168.70.133:0x01
12:14:03.103 192.168.70.201 > 192.168.70.133
  PFCP Session Establishment Response
  Cause: Request accepted(1)  SEID=0x00000001
12:14:08.445 192.168.70.133 > 192.168.70.201
  PFCP Session Modification Request (HO)
  Update FAR: Apply Action FORW  teid=0x0042
  PFCP Session Modification Response OK 0.9ms
```

Real PFCP trace: tcpdump on N4 interface · SMF↔UPF PFCP session establishment + modification during handover · N4 latency: 0.9–1.2 ms · Zero session loss across 200+ handover events

End-to-End AI Control: Coupling RAN Intelligence with Core



Three Cross-Domain Scenarios Demonstrated on OAI Testbed:

1. **S1 — Congestion-Triggered Slice Reallocation:** PRB saturation detected by xApp → signals SMF via SBI → SMF adjusts AMBR+QER on UPF. Joint RAN+core adaptation < 500ms

2. **S2 — Handover-Triggered Path Switch Optimization:** gNB Xn HO triggers SMF N4 session modify → UPF GTP tunnel update → xApp reconfigures new cell scheduling. < 70ms end-to-end

3. **S3 — Slice-Aware RAN+Core Co-Scheduling:** Resource xApp jointly adjusts PRB quotas per slice + signals SMF to enforce matching per-slice AMBR — 98.7% isolation sustained

Lessons Learned: What Surprised Us in the Testbed

Area	The Problem	What Fixed It
⚠ E2 Timing	RIC missed 10ms KPM windows — silent, no error log	taskset to pin RIC to isolated cores + watchdog timer. Loop jitter < 2ms.
⚠ E2SM-RC Encoding	ASN.1 mismatch between FlexRIC RC encoder and OAI gNB agent — messages dropped silently	Wireshark on loopback revealed it. Patched encoder, fix submitted upstream to FlexRIC.
⚠ UPF Slice QER	At >400 Mbps aggregate, PFCP QER broke — eMBB traffic leaked into URLLC allocation	Added per-SEID tc-htb rules in UPF container. Kernel-level enforcement, independent of PFCP.
⚠ RF Calibration	USRP B210 TX gain drifted 2–3 dB between sessions — masked xApp gains in delta measurements	Per-session calibration script: pilot Tx → measure RSRP → adjust offset. Takes 30s.
✓ Docker NF Order	Race condition on startup: AMF/SMF register before NRF is ready — silent N4 failures	healthcheck depends_on in compose file. NRF must be healthy before AMF/SMF start.
✓ Traffic Realism	iPerf UDP results didn't match RTSP/SIPp results — scheduler behaves completely differently	Always test with real application traffic. SIPp and VLC are free and 5 min to set up.

Summary of Contributions

Key Numbers

+46.6%	aggregate throughput
−46%	end-to-end latency
+8.6 dB	SINR at cell edge
98.7%	slice isolation (Jain)
28 ms	Xn handover gap
200+	hours of experiments

What We Contributed

C1 xApp Suite: Three ML xApps deployed on FlexRIC with E2AP v3 — offloading, interference, orchestration. Results published at IEEE ICC 2023–2025.

C2 5GC Experiments: Live slice isolation, Xn handover continuity, and N4 PFCP tracing — first experimental analysis of OAI core across all three.

C3 E2E AI Loop: First closed AI loop coupling Near-RT RIC xApps with OAI core NFs (SMF/UPF). Three cross-domain scenarios measured.

C4 Open Science: Full reproducibility: public code, configs, traffic scripts. xApp templates and 5GC scripts contributed back to the OAI community.

Thank You

AI-Native End-to-End 5G/6G with OAI

Key Takeaway

OAI enables rigorous, hardware-validated, fully reproducible AI-native 5G/6G research — from individual xApps to end-to-end cross-domain control loops. The open-source platform is ready for your next 6G experiment.