

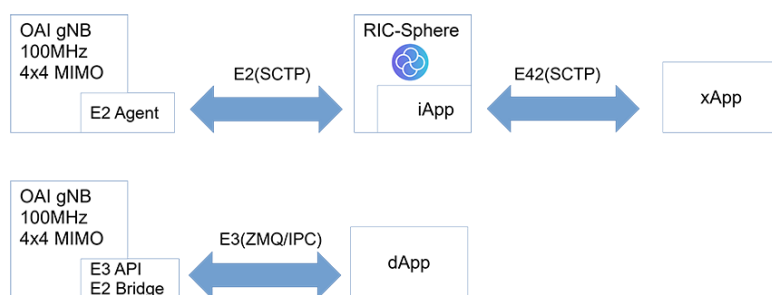
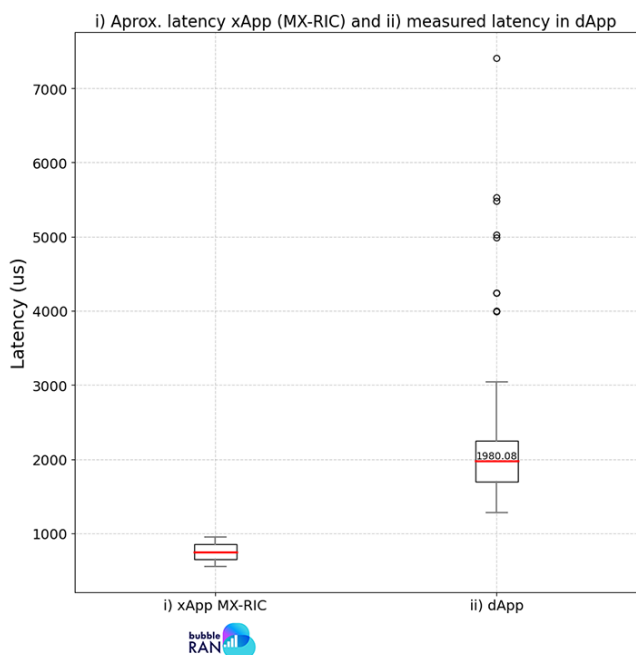


rApps, xApps and dApps in the O-RAN and AI-RAN landscape

Abstract

Open RAN intelligence is no longer a single-controller story. It is becoming a layered software system in which rApps operate in the Non-RT domain, xApps operate in the Near-RT domain, and dApps push logic into the CU/DU boundary when access to low-layer data and sub-10 ms reactions are required. Commercial momentum today is strongest around rApps, R1, SMO, and Non-RT RIC modernization, while the most aggressive exploration of ultra-fast control loops is still happening in advanced xApp and dApp research.

This white paper compares rApps, xApps, and dApps in a nutshell, discusses their main benefits and indicative timescales, and then examines an OpenAirInterface-based benchmark using the O-RAN Low Layer Control (LLC) service model for I/Q sample reporting. In the benchmark supplied for this paper, the BubbleRAN xApp path delivers deterministic end-to-end real-time reporting and control at about 650 us average and 950 us maximum, while the compared dApp path is reported in the attached benchmark note at roughly 2 ms with visible variance. The result does not mean that dApps are inherently slower than xApps; it does show that well engineered ultra-lean Near-RT RIC implementations can already deliver sub-millisecond, production-oriented closed loops for demanding O-RAN and AI-RAN workloads.



rApps, xApps and dApps in the O-RAN and AI-RAN landscape

1. rApps, xApps, and dApps in a nutshell

The simplest way to understand the three application classes is by their control horizon and where they execute.

rApps belong to the Non-RT RIC / SMO domain. They are best suited to long-horizon automation tasks such as policy generation, SLA assurance, data exposure, model lifecycle management, and cross-network optimization. Their natural timescale is above one second, even though the underlying platform may move policies and model updates much faster.

xApps belong to the Near-RT RIC. They implement standardized O-RAN closed loops for monitoring, optimization, and control, traditionally in the 10 ms to 1 s range, but in practical systems increasingly extending toward the sub-millisecond edge for specific low-layer workflows when the RIC stack is ultra-lean and paired with suitable service models.

dApps are an emerging extension aimed at the below-10 ms domain. They are deployed closer to, or directly inside, the CU/DU boundary to access user-plane or low-layer data that is difficult to export efficiently over classical O-RAN interfaces. Their appeal is locality, timing, and direct access to data such as I/Q samples. Their challenge is that they are more intrusive, less operationally mature, and harder to isolate than xApps.

Taken together, the hierarchy is clear: rApps optimize the network as a business and policy system, xApps optimize the RAN as a standardized control system, and dApps optimize the radio stack as a real-time embedded system. The simplest, and inaccurate, way to think about the three models is this:

App type	Runs where	Main interfaces	Indicative time-scale	Best fit	Main benefit	Main limitation
rApp	Non-RT RIC / SMO domain	R1, A1, O1	> 1 s	policy, analytics, AI/ML training, intent orchestration	network-wide optimization with low intrusiveness	too slow for fast PHY/MAC-adjacent loops
xApp	Near-RT RIC	E2, A1-related APIs	10 ms to 1 s in the standard hierarchy	closed-loop monitoring/control through standardized service models	multivendor, externally hosted, operationally safer	extra hop from the RAN and limited locality to user-plane data

2. Commercial adoption versus research frontier

The current market trajectory is uneven. Commercial adoption is clearly accelerating first around rApps, R1, SMO, and Non-RT RIC platforms. This is where operators and vendors can introduce automation with relatively low operational risk, clear governance, and strong multi-vendor alignment. The rise of the Ericsson EIAP ecosystem and agentic rApp offerings illustrates this direction.

By contrast, the research frontier is more active around advanced xApps and dApps, where the goal is to close faster loops, expose richer data, and support AI-for-RAN tasks that cannot wait for higher-layer analytics windows. The attached dApp and Wasm papers make this point directly: several important decisions in scheduling, sensing, beam management, and real-time inference occur below the timescales traditionally associated with O-RAN control.

rApps, xApps and dApps in the O-RAN and AI-RAN landscape

3. RIC-Sphere: BubbleRAN's approach to rApps, xApps and dApps

BubbleRAN does not force a trade-off between commercial readiness and research agility through its RIC-Sphere product. It combines Non-RT RIC, Near-RT RIC, and a Proxy-E2 Agent in one O-RAN-compliant platform, with support for R1, A1, and E2, reusable xApp/rApp SDKs, cloud-native deployment options, and an engineering workflow that spans development, integration, onboarding, and operation from lab to production.

For industry, that translates into an interoperable automation platform with a practical path to deployment. The RIC-Sphere supports for R1AP, A1, E2AP v3.0, and key service models including KPM, RC, CCC, and LLC, together with use cases such as SLA assurance, mobility load balancing, interference management, dataset collection, localization, and sensing. RIC-Sphere also provides compatibility with the Ericsson EIAP ecosystem on the rApp side, allowing operators to modernize around R1 and Non-RT automation without fragmenting their broader O-RAN roadmap while rApp developer benefit from this compatibility to extend their outreach.

For R&D, RIC-Sphere provides something equally valuable: a programmable and reproducible environment with SDKs, samples, documentation, open ecosystem assets, and an open ecosystem via the TelcoFabric portal. This is exactly the kind of infrastructure that allows researchers, vendors, and system integrators to move beyond isolated demos and toward repeatable, verifiable, and production-relevant experimentation.

In that sense, RIC-Sphere is not just another RIC. It is a bridge between research-grade programmability and industry-grade operability.

4. Benchmark: BubbleRAN xApp path versus the compared dApp path

The attached benchmark note compares two OpenAirInterface-based paths using the O-RAN LLC service model to report I/Q samples. In the first path, I/Q samples arriving at the PHY are sent through E2AP v3 and LLC v1 to the Near-RT RIC-Sphere, decoded, and forwarded to the subscribed xApp. In the second path, the architecture uses an E3 Agent over ZMQ/IPC with the subscribed Python dApp, running on the same Zen 5 machine.

In the benchmark figures supplied for this paper, the BubbleRAN xApp path averages about 650 us and peaks at 950 us. The attached benchmark note is directionally consistent with those figures: it reports the xApp path at less than 1 ms, while the compared dApp path is in the order of 2 ms with non-negligible variance.

The key point is not that dApps are categorically slower. In fact, the broader dApp literature shows that highly optimized dApp implementations can achieve sub-millisecond control loops. The conclusion is that both xApps and dApps (and even certain rApps) can achieve realtime and deterministic control, the questions is what are the decision drivers and use-case requirements.

5. Why this result matters

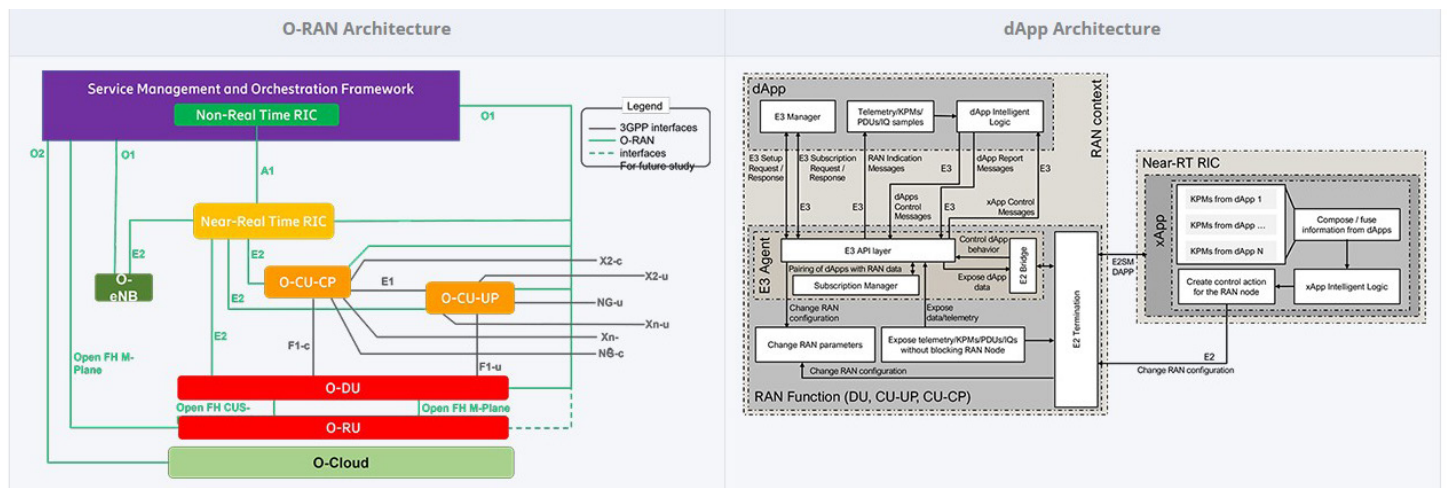
This benchmark matters because it changes a common assumption in the market. dApps are often introduced as the obvious answer for any low-layer, low-latency, I/Q-driven use case. This result shows that this assumption is too simplistic. A well-engineered Near-RT RIC can already support real-time I/Q reporting and control with deterministic sub-millisecond behavior.

That is strategically important. rApps and xApps remain non-intrusive, standardized, extensible, and multi-vendor friendly. They are easier to govern, easier to isolate operationally, and more natural to integrate into an industrial O-RAN lifecycle. If a Near-RT RIC can already meet the latency envelope for important sensing and control loops, then the burden of proof shifts to dApps: they must show a clear advantage in locality, data access, or timing that cannot be matched by a modern RIC implementation.

This is exactly where BubbleRAN's expertise becomes visible. Rather than treating O-RAN compliance and real-time performance as opposing goals, RIC-Sphere demonstrates that they can be combined in one architecture.

rApps, xApps and dApps in the O-RAN and AI-RAN landscape

6. Architecture placeholders



References:

O-RAN Arch: <https://docs.o-ran-sc.org/en/k-release/architecture/architecture.html>

dApp Arch (NEU): <https://arxiv.org/html/2501.16502v1>

7. Open questions

- Which concrete O-RAN and AI-for-RAN use cases still remain out of reach for xApps even when a Near-RT RIC already supports LLC and deterministic sub-millisecond loops?
- If xApps are standardized and non-intrusive, what are the real and defensible advantages of dApps beyond locality and privileged data access?
- Can sandboxed dApps close enough of the security and operability gap that operators will accept them in production at scale?
- What is the right split of responsibility between rApp intelligence, xApp execution, and dApp reflexes in future AI-agent-driven RAN automation?
- How should conflicts be resolved when rApps, xApps, and dApps all influence the same radio resources at different timescales?

Conclusion

The next phase of Open RAN is not about choosing one application class over another. It is about building a coherent hierarchy of automation and intelligence. rApps will continue to drive policy, orchestration, and AI lifecycle functions. xApps will remain the standardized execution layer for near-real-time optimization. Real-time Apps will continue to push the frontier for the most timing-critical and data-local functions.

Read a related Blog:

From Network Automation to Autonomous Networks:

<https://bubbleran.com/news/network-automation-towards-autonomous-networks-v1.2/>