Received 14 May 2026; revised 17 June 2026; accepted 2 July 2026. Date of publication 6 July 2026; date of current version 13 July 2026.

Digital Object Identifier 10.1109/OJCOMS.2026.3710136

Mapping the rApp Attack Surface: A Threat Analysis of Third-Party Applications in O-RAN

ROBERTO RIGGIO (Senior Member, IEEE)

Università Politecnica delle Marche, 60131 Ancona, Italy

e-mail: r.riggio@univpm.it

ABSTRACT The disaggregation of Radio Access Network (RAN) functions and the adoption of open interfaces, as defined by the O-RAN Alliance, represent a fundamental shift in how mobile networks are built and operated. Central to this vision is the Non-Real-Time RAN Intelligent Controller (Non-RT RIC), which hosts third-party applications known as rApps to deliver AI/ML-driven network optimization through standardized interfaces. While this open ecosystem unlocks significant flexibility and vendor diversity, it simultaneously expands the network threat surface in ways that existing security frameworks have not fully addressed. This article systematically analyzes the security risks introduced by third-party rApps on the Non-RT RIC, identifying four principal threat vectors: supply chain and onboarding vulnerabilities, policy manipulation via the A1 interface, data exfiltration through the Service Management and Orchestration (SMO) telemetry pipelines, and lateral movement across the rApp service mesh. Through illustrative attack scenarios, we demonstrate how these threats can result in network slice degradation, user equipment data exposure, and cascading compromise across RIC components. We further evaluate the adequacy of current O-RAN Alliance security specifications, particularly WG11, and identify critical standardization gaps. To address these risks, we propose a layered mitigation framework encompassing rApp certification, zero-trust API access controls, runtime behavioral monitoring, and container-level sandboxing. Our analysis highlights the urgent need for security-by-design principles to be embedded into the rApp ecosystem before large-scale O-RAN deployments mature, and calls on standardization bodies, network operators, and rApp vendors to treat openness and security as complementary rather than competing objectives.

INDEX TERMS Open RAN (O-RAN), non-RT RIC, rApp security, threat surface analysis, supply chain security, zero-trust architecture, 5G network security, AI/ML network management.

I. INTRODUCTION

THE mobile network infrastructure is undergoing its most significant architectural transformation in decades. Driven by the O-RAN Alliance's vision of disaggregated, interoperable, and software-defined radio access networks (RANs), operators worldwide are moving away from vertically integrated, vendor-proprietary systems toward open architectures built on standardized interfaces and multi-vendor component ecosystems [1]. The appeal is compelling: disaggregation reduces vendor lock-in, accelerates innovation cycles, and enables operators to deploy best-of-breed components across the RAN stack. Major operators across North America, Europe, and Asia-Pacific have already committed to O-RAN deployment roadmaps, and analyst projections place the global O-RAN market at tens of billions of dollars within the coming years.

At the intelligence layer of this architecture sits the RAN Intelligent Controller (RIC), a two-tiered control plane that introduces programmable, AI/ML-driven optimization into network operations. The Non-Real-Time RIC (Non-RT RIC), operating on timescales greater than one second, is responsible for policy guidance, model training, and high-level orchestration. It achieves this through rApps, a modular set of third-party applications that consume network data, generate AI-driven insights, and deliver policy directives to the RAN via the A1 interface. The rApp model is explicitly designed to be open: operators are expected to source rApps from a diverse marketplace of vendors, integrators, and internal development teams, and to deploy them on a shared Non-RT RIC platform managed through the Service Management and Orchestration (SMO) framework.

This openness is precisely what makes rApps powerful–and precisely what makes them dangerous.

The introduction of third-party software into the core intelligence layer of a mobile network represents a qualitative shift in the telecom security landscape. Historically, RAN components were closed, monolithic, and controlled end-to-end by a small number of trusted vendors. Security was enforced through supply chain control, physical security, and proprietary interfaces that were difficult to exploit without insider access. O-RAN dismantles each of these assumptions. The A1, R1, and O1 interfaces are standardized and documented. The SMO exposes rich telemetry pipelines. The rApp execution environment is, by design, accessible to third parties. Each of these properties that operators value for agility simultaneously lowers the barrier for adversarial exploitation.

Despite growing attention to O-RAN security in both academic literature [2], [3], [4], [5], [6], [7], [8] and standardization bodies [9], [10], most notably through the O-RAN Alliance's Security Working Group (WG11), the specific risks introduced by third-party rApps on the Non-RT RIC remain underexamined. Existing security analyses have focused predominantly on the Near-RT RIC and xApp ecosystem, on open fronthaul interface vulnerabilities, or on broad architectural threat surveys. The Non-RT RIC's role as a policy originator, data aggregator, and AI model host gives it a uniquely sensitive position in the network, yet its rApp ecosystem has received comparatively little scrutiny.

This article addresses that gap directly. We present a systematic analysis of the threat surface introduced by third-party rApps on the Non-RT RIC, organized around four principal attack vectors: supply chain and onboarding vulnerabilities, policy manipulation via the A1 interface, data exfiltration through SMO telemetry pipelines, and lateral movement across the rApp service mesh. We ground this analysis in two illustrative attack scenarios that demonstrate how these threats manifest in practice, and we evaluate the extent to which current O-RAN Alliance specifications provide adequate protection. Finally, we propose a layered mitigation framework and identify the standardization gaps that must be closed before large-scale O-RAN deployments make these risks systemic.

II. THE RAPP ECOSYSTEM: OPPORTUNITY AND EXPOSURE

The Non-Real-Time RIC [11] and its associated rApp ecosystem occupy a pivotal position in the O-RAN architecture. To appreciate the security implications that follow in subsequent sections, it is necessary first to understand what rApps are, how they interact with the broader network, and why their third-party, open-marketplace model represents a structural departure from traditional telecom software paradigms.

A. THE NON-RT RIC IN CONTEXT

The O-RAN architecture decomposes the traditional RAN into a set of disaggregated functional units connected through open, standardized interfaces. Above this radio stack sits the RIC, split into two tiers based on control loop latency requirements. The Near-RT RIC operates on timescales between ten milliseconds and one second, hosting xApps that perform fine-grained, real-time radio resource management. The Non-RT RIC operates on timescales greater than one second and is responsible for the slower, higher-order functions of network intelligence: policy generation, AI/ML model training and deployment, and long-horizon network optimization. The reference O-RAN architecture is sketched in Fig. 1.

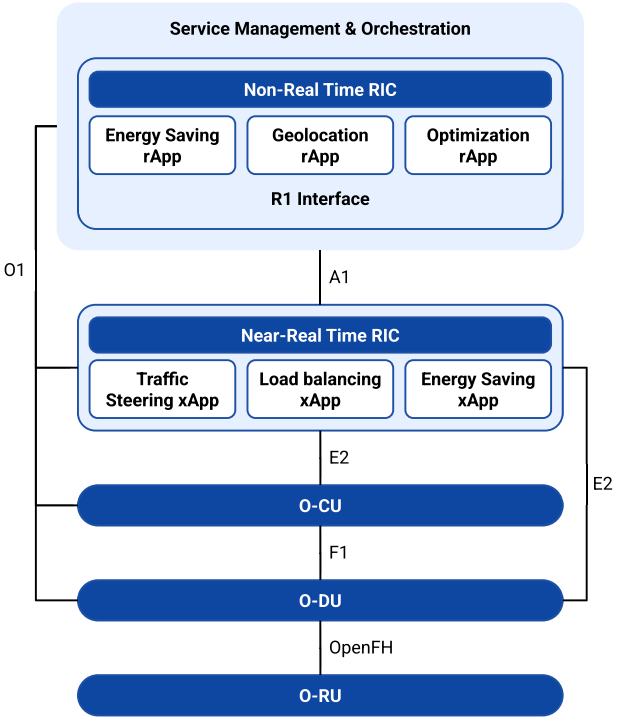

FIGURE 1. O-RAN architecture overview illustrating the Non-RT RIC and rApp ecosystem within the SMO framework.

The Non-RT RIC is hosted within the SMO, which serves as the overarching management plane for the entire O-RAN stack. Through the SMO, the Non-RT RIC can observe network-wide telemetry via the O1 interface, deliver policy guidance to the Near-RT RIC via the A1 interface, and expose service-based APIs to rApps via the R1 interface. This positioning gives the Non-RT RIC, and by extension the rApps running on it, a uniquely privileged view of and influence over network behavior.

B. WHAT RAPPS ARE AND WHAT THEY DO

rApps are modular software applications designed to run on the Non-RT RIC platform and consume its exposed services through the R1 interface. The R1 interface provides rApps with access to a standardized set of capabilities including data collection and exposure services, AI/ML model management services, and policy and intent-driven management services toward the A1 interface. In practical terms, an

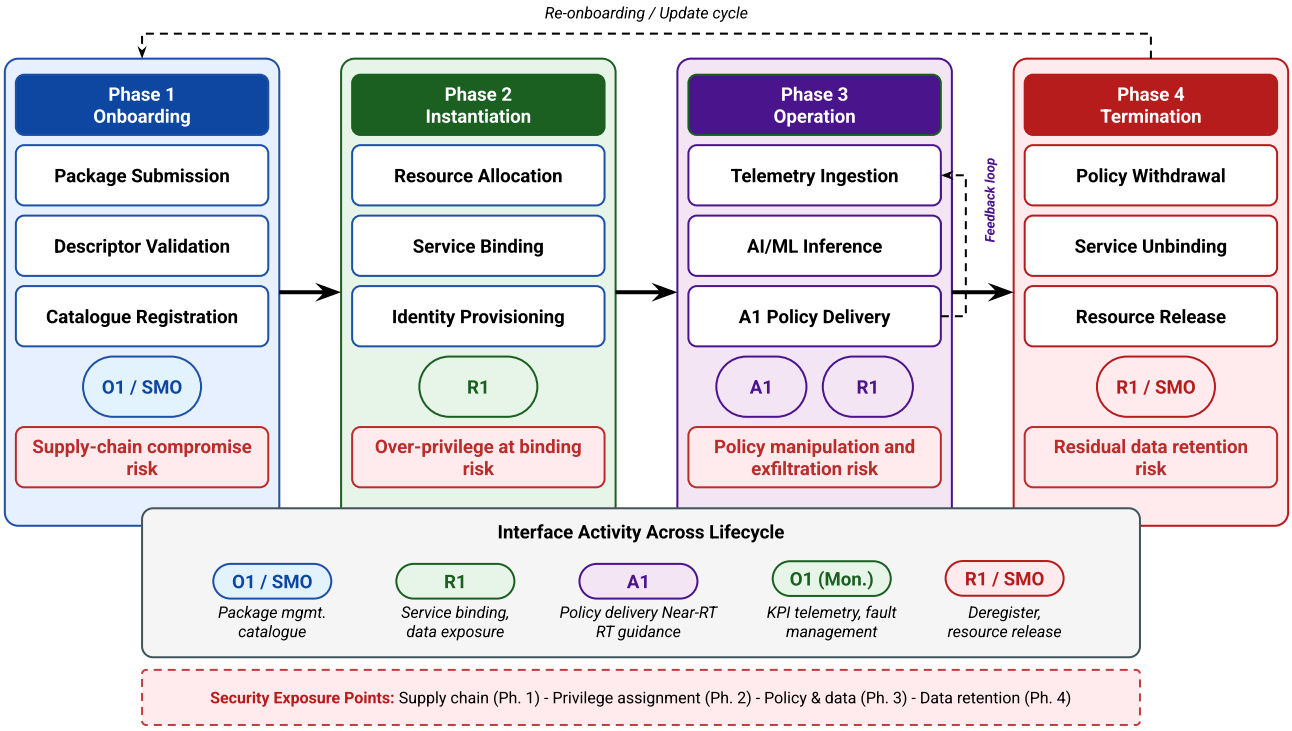

FIGURE 2. rApp lifecycle across four phases–Onboarding, Instantiation, Operation, and Termination–with key interface interactions (O1, R1, A1) and security exposure points annotated at each phase. The dashed arc denotes the re-onboarding path triggered by rApp updates or version changes.

rApp might ingest cell-level performance metrics, train or invoke a machine learning model to predict congestion, and then generate an A1 policy that instructs the Near-RT RIC to preemptively redistribute load across frequency bands.

The rApp lifecycle consists of four broad phases: (i) onboarding, in which the rApp package is validated and registered with the Non-RT RIC platform; (ii) instantiation, in which the rApp is deployed and its required services are bound; (iii) operation, in which the rApp continuously consumes data and produces policies; and (iv) termination, in which the rApp is decommissioned and its resources released. Each of these phases carries distinct security implications that we examine in detail in the next section. The rApp lifecycle across the four phases is sketched in Fig. 2.

Critically, the O-RAN Alliance's design intent is for rApps to be sourced from a diverse, competitive ecosystem of third-party developers, not solely from the operator or the Non-RT RIC platform vendor. This is analogous in spirit to a mobile application marketplace: operators act as platform owners, and rApp developers act as third-party publishers whose software runs on that platform with access to sensitive system resources. The analogy is instructive precisely because mobile application ecosystems have spent over a decade grappling with the security consequences of exactly this model.

C. KEY INTERFACES AND THEIR SECURITY RELEVANCE

Three interfaces define the rApp threat surface and deserve particular attention.

The R1 interface [12] is the primary channel through which rApps interact with the Non-RT RIC platform. It exposes service-based APIs covering data exposure, AI/ML workflow management, and policy delivery. Because R1 is the entry point for all rApp activity, weaknesses in its access control model directly determine the scope of damage a malicious or compromised rApp can cause. The O-RAN Alliance has defined R1 at a functional level, but detailed security specifications, such as authentication granularity, authorization scopes, and audit logging requirements, remain incompletely standardized as of current WG11 publications.

The A1 interface [13] connects the Non-RT RIC to the Near-RT RIC and carries AI/ML-enriched policies and intent-based guidance downstream. From a security perspective, A1 is a high-value target: policies delivered over A1 directly influence real-time radio resource management decisions made by xApps. A compromised rApp with write access to the A1 policy pipeline can therefore affect network behavior at operational timescales, potentially degrading service quality, starving specific network slices [14] of resources, or creating conditions favorable to further exploitation.

The O1 interface [15] provides access to performance measurement data, configuration parameters, and fault

management telemetry across the entire O-RAN stack. rApps that consume O1-sourced data through R1 data exposure services are effectively handling sensitive operational intelligence about the network, including, in some configurations, aggregated user equipment behavioral data. The breadth of O1 telemetry accessible to rApps creates a significant data exposure risk if access controls are insufficiently granular.

D. THE STRUCTURAL SECURITY PROBLEM

Traditional telecom network software was procured, validated, and operated entirely within a closed vendor-operator relationship. Software running on network elements underwent extensive certification, and the attack surface was bounded by the proprietary nature of both the interfaces and the software itself. The rApp model inverts several of these assumptions. Interfaces are open and documented. The execution environment is shared across multiple third-party rApps. The onboarding process is designed for agility, not exhaustive security vetting and the policy influence of rApps extends directly into real-time network operations through the A1 interface.

This is not an argument against the O-RAN model, as its benefits of openness and programmability are real and significant. It is, however, an argument that the security architecture supporting the rApp ecosystem must be treated with the same rigor historically applied to less exposed network components. As we demonstrate in the following section, the current state of that architecture leaves meaningful gaps.

III. THREAT SURFACE ANALYSIS

The rApp ecosystem introduces security risks that are different from those addressed by conventional RAN security frameworks. Unlike threats targeting radio interfaces or transport links, the risks examined here originate from within the network's own intelligence layer, from software that operators have deliberately granted privileged access to network data and policy mechanisms. This section identifies and analyzes four principal threat vectors, each rooted in a distinct architectural property of the Non-RT RIC and its interfaces. A threat surface map is sketched in Fig. 3.

The four threat vectors analysed in this section are not attributable to a single adversary class. The Non-RT RIC rApp ecosystem exposes assets to at least five structurally distinct adversary types whose capabilities, access paths, and goals differ in analytically significant ways and require correspondingly different mitigations. Table 1 formalises these distinctions across five dimensions.

First, adversaries A1 and A2, the malicious vendor and the compromised build pipeline, both exploit the platform's legitimate trust mechanisms rather than bypassing them, which is why they are resistant to controls that operate at the interface boundary. This means that the A1-PMS, the ICS, and the R1 authorisation layer all behave correctly from the platform's perspective while the attack proceeds. This structural property motivates the supply chain and behavioural monitoring mitigations proposed in Sec. V-A and V-C, respectively, which are the only controls positioned upstream of the trust assumption.

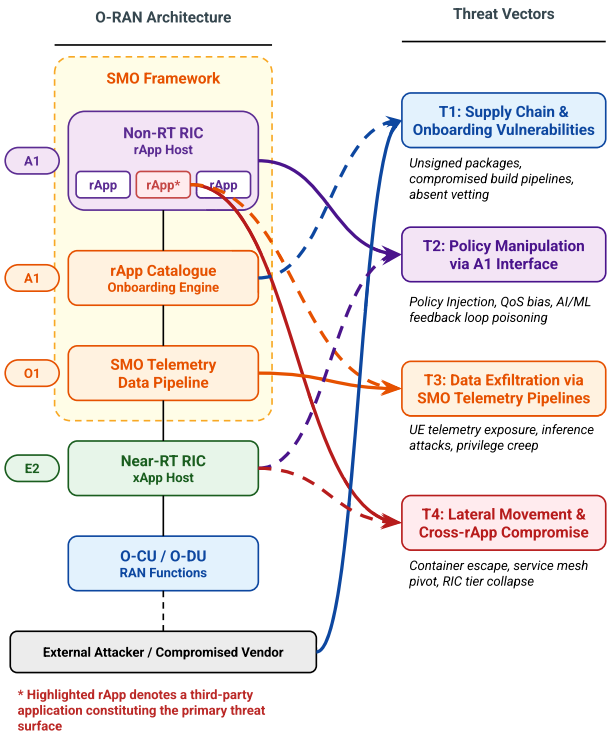

FIGURE 3. Threat surface map for the O-RAN rApp ecosystem. Solid arrows denote primary exploitation paths; dashed arrows denote secondary or cascading paths.

Second, adversary A3, the over-privileged benign rApp, differs categorically from the others in that it requires no malicious intent. In fact, in this case, the threat arises entirely from the gap between the authorization scope granted at instantiation and the scope actually required for the rApp's legitimate function. No monitoring system operating on behavioral anomaly detection can reliably distinguish an over-privileged rApp accumulating telemetry from one performing its normal function, because the access is authorized. This adversary can only be addressed through the least-privilege authorization controls of Sec. V-B, applied at the R1 service binding stage before the rApp enters operation.

Third, adversaries A4 and A5 differ from A1–A3 in their entry point. In particular, they originate outside the rApp application layer entirely, exploiting the execution environment and administrative management plane, respectively. The mitigations for A4 (container isolation, Sec. V-D) and A5 (zero-trust API controls, Sec. V-B) are therefore orthogonal to those addressing A1–A3 and must be deployed as independent layers rather than as substitutes.

A. SUPPLY CHAIN AND ONBOARDING VULNERABILITIES

The first and most fundamental threat vector lies in the rApp supply chain, i.e., the processes by which rApp software is developed, packaged, and onboarded onto the Non-RT

TABLE 1. Adversary model for third-party rApp threats on the non-RT RIC.

Adversary	Asset Targeted	Access Path	Goal	Privilege Level	Threat Vector(s)
A1: Malicious rApp vendor	A1 policy pipeline; Near-RT RIC operational behaviour	Legitimate rApp onboarding and registration via A1-PMS; authorised `PUT /a1-policy/v2/policies`	Sustained, covert degradation of target network slice QoS through calibrated policy bias	*Full platform trust*: registered service consumer with write access to A1-PMS policy endpoint	T2 (primary); T1 (enabling)
A2: Compromised vendor build pipeline	rApp package integrity; Non-RT RIC execution environment	Supply chain injection prior to marketplace submission; dormant logic activated post-deployment	Embed persistent malicious capability (data exfiltration, policy manipulation, lateral movement) surviving onboarding validation	*Pre-deployment*: no platform access until onboarding succeeds; post-activation inherits full rApp runtime privilege	T1 (primary); T2, T3, T4 (post-activation)
A3: Over-privileged benign rApp	SMO O1 telemetry pipeline; UE behavioural data	Legitimate ICS subscription with over-broad R1 authorisation scope granted at instantiation	Passive accumulation of network telemetry beyond functional requirements; latent exfiltration surface	*Authorised but excessive*: valid OAuth 2.0 token with scope wider than operational necessity	T3 (primary)
A4: External attacker exploiting runtime weakness	rApp execution environment; co-resident rApp data; Non-RT RIC platform services	Remote code execution via container runtime vulnerability (e.g. CVE-2024-21626); inherited rApp service account credentials	Lateral movement to co-resident rApps; privilege escalation to platform level; secondary data exfiltration via service mesh	*Initially zero*: achieves rApp-level privilege post-exploit; escalates via Kubernetes service account token automounting	T4 (primary); T3 (secondary)
A5: Insider or rogue operator	R1 authorisation configuration; rApp catalogue; ICS job definitions	Direct administrative access to Non-RT RIC platform management interfaces; ability to modify authorisation scopes and onboarding policies	Widen R1 authorisation scopes beyond operational necessity; onboard uncertified rApps; suppress audit logging	*Administrative*: platform-level management access outside rApp trust boundary	T1, T3 (primary); T2, T4 (enabling)

RIC platform [16], [17], [18]. Because the rApp model explicitly anticipates third-party developers publishing applications into an operator-accessible marketplace, the integrity of the software supply chain is a prerequisite for every security guarantee that follows. If a malicious rApp is successfully onboarded, all downstream controls operate on a false assumption of trust.

Supply chain risks manifest at several points. During development, an rApp vendor's build pipeline may itself be compromised, resulting in a package that appears legitimate but contains embedded malicious logic, a pattern consistent with high-profile software supply chain attacks observed in enterprise software ecosystems over recent years. During distribution, the absence of mandatory cryptographic signing requirements in current O-RAN Alliance specifications means that package integrity cannot be reliably verified at the point of onboarding. An rApp delivered through a compromised registry or a man-in-the-middle distribution channel may be indistinguishable from its legitimate counterpart without robust signing and verification infrastructure.

The onboarding process itself presents additional risk. Current O-RAN specifications define the functional requirements for rApp package descriptors and onboarding workflows, but do not mandate the depth of security vetting that would

be necessary to detect sophisticated malicious logic before deployment. Operators are therefore dependent on the trustworthiness of their rApp vendors and the integrity of the distribution channel, neither of which can be assumed in a competitive, multi-vendor marketplace. The consequences of a successful supply chain compromise are severe: a malicious rApp, once onboarded, operates from a position of legitimate platform trust, making its subsequent actions significantly harder to distinguish from normal application behavior.

B. POLICY MANIPULATION VIA THE A1 INTERFACE

The second threat vector targets the A1 interface and the policy delivery mechanism it supports. As described in Sec. II, rApps interact with the A1 interface through the Non-RT RIC platform's policy management services, generating structured policies that are delivered downstream to the Near-RT RIC and ultimately influence real-time radio resource management. This policy pipeline represents one of the most consequential attack surfaces in the O-RAN architecture precisely because its outputs have direct operational effect.

A compromised or malicious rApp with access to A1 policy services can manipulate network behavior in ways that are difficult to attribute and potentially severe in impact. Policy manipulation attacks need not be dramatic to be effective. Subtle, sustained biasing of handover thresholds, load balancing weights, or quality-of-service parameters can degrade service for targeted network slices or user populations while remaining within the statistical noise of normal network variation. More aggressive manipulation, such as injecting conflicting policies, flooding the policy engine with high-volume low-quality directives, or deliberately triggering oscillatory behavior in the Near-RT RIC's control loops, can produce measurable and disruptive network degradation.

A particularly insidious variant of this attack involves the feedback loop between the Non-RT RIC and the Near-RT RIC. The Non-RT RIC is designed to observe network performance outcomes and refine its policies accordingly, creating a closed-loop optimization cycle. A malicious rApp that subtly poisons this feedback loop, e.g., by misreporting performance metrics or biasing the training data used by AI/ML models hosted on the Non-RT RIC, can cause the platform itself to generate increasingly suboptimal policies without any single policy being obviously malicious. This class of attack, analogous to data poisoning in federated learning systems, is particularly challenging to detect because the harm emerges from the accumulation of individually plausible actions rather than a single identifiable intrusion event.

C. DATA EXFILTRATION THROUGH SMO TELEMETRY PIPELINES

The third threat vector concerns the data exposure surface created by rApps' access to network telemetry through the SMO and R1 data services. As noted previously, the O1 interface aggregates performance measurements, configuration data, and fault telemetry from across the O-RAN stack, and rApps can request access to this data through standardized R1 data exposure services. The breadth and sensitivity of this telemetry make it a high-value target for exfiltration. For example, in [19], the authors show how a malicious rApp can exploit the Non-RT RIC message brokering mechanism to eavesdrop on PM streams without detection.

The data available to rApps through these pipelines varies, but may include cell-level performance counters, handover event records, load and congestion metrics, and aggregated behavioral signatures derived from UE activity. While individual telemetry records may not directly identify users, the combination of spatiotemporal network data with external datasets can yield sensitive inferences about user behavior, location patterns, and device activity. For nation-state adversaries or sophisticated criminal actors, access to this telemetry through a compromised rApp represents a persistent, high-fidelity intelligence capability embedded within the operator's own infrastructure.

Beyond external exfiltration, over-privileged rApps present an insider threat dimension. An rApp granted access to telemetry beyond what its legitimate function requires, a consequence of insufficiently granular authorization scopes in the R1 interface, accumulates a data access footprint that represents latent risk even if the rApp itself is not initially malicious. Privilege creep in the rApp authorization model, left unaddressed, creates the conditions for data exposure incidents that originate not from external attack but from the ordinary operation of legitimately onboarded software.

The current O-RAN Alliance specifications provide limited guidance on data minimization principles for rApp authorization, and do not define mandatory audit logging requirements that would allow operators to detect anomalous data access patterns after the fact. This gap between the richness of the data exposed through R1 and the maturity of the access control framework governing that exposure is one of the most pressing near-term security concerns in the rApp ecosystem.

D. LATERAL MOVEMENT AND CROSS-RAPP COMPROMISE

The fourth threat vector addresses the risk of lateral movement, i.e., an attacker's ability to pivot from an initial foothold in one rApp to broader compromise of the Non-RT RIC platform or adjacent systems. This risk is a direct consequence of the shared execution environment in which multiple rApps operate simultaneously, and is amplified by the service mesh architecture through which rApps interact with platform services and, in some configurations, with each other.

In a well-isolated execution environment, a compromised rApp is contained within its own security boundary and cannot directly access the resources of co-resident applications. In practice, the degree of isolation enforced in Non-RT RIC deployments depends heavily on implementation choices made by platform vendors, choices that current O-RAN Alliance specifications leave largely to vendor discretion.

TABLE 2. Mapping of rApp threat vectors to real-world CVEs.

Threat Vector	Direct CVE(s)	Adjacent CVE / Evidence	Mapping Strength
T1: Supply Chain & Onboarding	CVE-2024-3094 (XZ Utils backdoor); CVE-2022-0811 (CRI-O privilege escalation)	CVE-2024-9486; CVE-2024-9594 (Kubernetes Image Builder supply chain)	**Strong**
T2: Policy Manipulation via A1	CVE-2023-41628 (OSC RIC out-of-order message handling); CVE-2023-40998; CVE-2023-41627 (RIC Message Router)	—	**Moderate**
T3: Data Exfiltration via SMO Telemetry	—	—	**Weak**
T4: Lateral Movement & Cross-rApp Compromise	CVE-2024-21626 (runC escape, CVSS 8.6); CVE-2019-5736 (runC, in-the-wild); CVE-2022-0185 (Linux kernel namespace bypass); CVE-2024-7646 (Kubernetes ingress injection)	—	**Strong**

Container-based deployments vary widely in the strictness of their namespace separation, network policy enforcement, and inter-container communication controls [20], [21]. An rApp executing in an insufficiently isolated container may be able to observe traffic between co-resident rApps, access shared file system resources, or exploit vulnerabilities in the container runtime itself to escape its boundary.

The shared service mesh presents a related risk. rApps communicate with Non-RT RIC platform services through a common API layer. A compromised rApp that can manipulate service mesh routing, forge service identities, or exploit weaknesses in mutual authentication between services can potentially intercept or modify the communications of other rApps operating on the same platform. In the most severe scenario, lateral movement from a single compromised rApp could yield platform-level access, enabling an attacker to observe, modify, or disrupt the policy and AI/ML operations of the entire Non-RT RIC.

The pathway from Non-RT RIC to Near-RT RIC is also relevant here. The A1 interface connects the two RIC tiers, and a platform-level compromise of the Non-RT RIC, achieved through lateral movement originating in a single rApp, could give an attacker authoritative access to the A1 policy pipeline, effectively collapsing the architectural separation between the two tiers that O-RAN's design intends to maintain.

E. MAPPING OF RAPP THREAT VECTORS TO REAL-WORLD CVES

Table 2 provides a mapping of the rApp Threat Vectors discussed in this section to Real-World CVEs. Three observations are worth stating.

First, the absence of rApp-specific CVEs for T2 and T3 is not evidence of low risk, it is evidence that the Non-RT RIC and its rApp ecosystem have not yet attracted the depth of adversarial security research that the Near-RT RIC has. The OSC Near-RT RIC has been the subject of dedicated vulnerability research precisely because it is the more mature open-source implementation. As Non-RT RIC implementations mature and become more widely deployed, the CVE density targeting rApp-specific vectors is expected to increase substantially.

Second, T1 and T4 benefit from a critical structural advantage: they map to vulnerabilities in the underlying software infrastructure (e.g., container runtimes, Kubernetes, and build pipelines) that have been extensively studied independently of O-RAN. The rApp ecosystem inherits these vulnerabilities by virtue of its technology stack, making the CVE evidence directly applicable regardless of whether the CVE was discovered in an O-RAN context.

Third, T3 remains the most analytically original contribution of the paper and the one that future empirical work should prioritize. The quantitative validation in Sec. VI provides the first measurement-based evidence for this threat vector, which is precisely the kind of contribution that fills the gap left by the absence of direct CVEs.

F. POSITIONING OF THIS WORK

The threat analysis presented in this section extends existing security frameworks along five dimensions that, taken together, define the paper's original contribution.

First, the O-RAN Alliance's WG11 security working group bases its threat modeling on the Microsoft STRIDE framework [22], decomposing the O-RAN architecture into data flows and mapping threats to interface boundaries, characterizing, for instance, whether A1 messages can be spoofed in transit or whether O1 telemetry can be tampered with at the transport layer. This interface-centric methodology is well-suited to the threat class it was designed for, but it does not naturally surface threats that originate within a trusted architectural component operating through authorized channels. A compromised rApp issuing structurally valid A1 policy updates with subtly biased QoS parameters is not spoofing an interface or elevating a privilege in any

STRIDE-meaningful sense; on the contrary, it is behaving exactly as the platform expects an authenticated, registered application to behave. Three of the four threat vectors identified in this article belong to this class of intra-trust-boundary threat, for which STRIDE provides no analytical category and current WG11 specifications provide no normative control.

Second, the WG11-informed Quad Open RAN Security Report [23] assigns only a medium severity rating to the Non-RT RIC and rApp threat category, on the grounds that their functionality is primarily related to network optimization and that compromise results in service degradation rather than critical impact. This article directly challenges that assessment: the closed-loop feedback exploitation in Scenario A demonstrates that a compromised rApp can cause the Non-RT RIC's own optimization model to converge on a degraded operating point, i.e., a self-reinforcing harm that persists after the malicious policy is removed. At the same time, the inference attack quantified in Sec. VI demonstrates that UE mobility profiles can be reconstructed within a single operational period from telemetry accessible through over-privileged R1 authorization scopes. Neither consequence maps cleanly onto the "service degradation" severity category, and both argue for a materially higher risk classification than current WG11 documents assign.

Third, cloud-native security guidelines, including NIST SP 800-204 [24], CIS Kubernetes Benchmarks [25], and the CNCF Cloud Native Security Whitepaper [26], correctly identify the vulnerability classes exploited by T1 and T4, but at a level of abstraction that does not account for two properties specific to the rApp ecosystem: the A1 policy pipeline's direct propagation of application-layer misbehavior into radio resource management decisions, and the ICS push delivery model's inversion of the query-volume anomaly detection signal on which cloud-native exfiltration monitoring depends. Generic container security frameworks are not designed to reason about either property.

Fourth, the rApp supply chain governance problem is qualitatively distinct from the enterprise software supply chain scenarios addressed by SLSA and NIST SP 800-218, because the trust relationships between rApp developers, marketplace operators, and network operators are not governed by the employment relationships and procurement contracts that those frameworks assume, a structural difference that the T1 threat analysis makes explicit and that existing guidelines do not address.

Fifth, the closest academic work [27], [28], [29], [30] treats the Non-RT RIC's rApp population as a source of threats to downstream components rather than as an attack surface in its own right. This article inverts that perspective, treating the Non-RT RIC platform itself as the system under analysis and producing a threat taxonomy that is complementary to rather than overlapping with the Near-RT RIC security literature. The four threat vectors identified here do not appear in this combination or at this level of specificity in any prior WG11 document, ENISA report, or published academic work, and it is on this basis that the article's originality claim rests.

IV. ILLUSTRATIVE ATTACK SCENARIOS

The threat vectors identified in the previous section are not merely theoretical constructs, they represent exploitable pathways that arise directly from architectural properties of the Non-RT RIC and its rApp ecosystem. To ground this analysis in operational reality, this section presents two illustrative attack scenarios that demonstrate how these threats manifest in practice. Each scenario is constructed to be technically plausible within the current O-RAN architecture, drawing on the interface properties and specification gaps identified in the preceding section. The scenarios are not presented as exhaustive exploit specifications but as structured narratives that illuminate the attack surface for operators, platform vendors, and standardization bodies. Both scenarios are illustrative. The rApp privilege levels, A1 policy access model, R1 telemetry scope, and service mesh isolation assumptions underlying each are formalised in the adversary model of Table 1.

A. SCENARIO A: SUPPLY-CHAIN-COMPROMISED RAPP MANIPULATES A1 POLICIES TO DEGRADE A TARGET NETWORK SLICE

1) THREAT VECTORS INVOLVED

Supply chain and onboarding vulnerabilities; Policy manipulation via the A1 interface

2) SETUP

A mid-sized mobile operator deploys an O-RAN architecture across a metropolitan area and sources rApps from a competitive marketplace of third-party vendors. Among the onboarded rApps is a network slice optimization application from a vendor with an established commercial track record. The rApp's stated function is to monitor slice-level key performance indicators (KPIs) and generate A1 policies that dynamically adjust resource allocation across the operator's enterprise and consumer slices. The operator's onboarding process includes package descriptor validation and basic functional testing but does not include static code analysis or behavioral sandboxing, consistent with the current state of O-RAN Alliance onboarding specifications.

3) ATTACK EXECUTION

Unbeknownst to the operator, the rApp vendor's software build pipeline was compromised several months prior to the application's marketplace submission. The compromise introduced a dormant logic component into the rApp package, one that passes functional testing because it activates only after a configurable delay following successful deployment and only under specific network load conditions designed to mimic normal operational triggers. Once activated, the malicious component begins subtly biasing the rApp's A1 policy outputs. Rather than generating policies that optimize resource allocation equitably across slices, the rApp introduces incremental but cumulative skew in the QoS parameter recommendations delivered to the Near-RT RIC, gradually deprioritizing the operator's highest-revenue enterprise slice in favor of consumer traffic.

The manipulation is engineered to remain within the bounds of plausible policy variation. Individual policy directives are not obviously anomalous; each can be rationalized as a legitimate response to observed network conditions. The Near-RT RIC's xApps execute the policies faithfully, as they have no mechanism to evaluate the intent or provenance of A1 directives, only their structural validity. Over a period of weeks, enterprise customers on the affected slice begin experiencing degraded latency and throughput during peak hours. The operator's network operations center attributes the degradation to seasonal demand growth and initiates a capacity planning review rather than a security investigation.

At the protocol level, the manipulation exploits three specific properties of the A1 Policy Management Service's northbound interface as implemented in the OSC Non-RT RIC reference platform. First, policy creation and modification share identical HTTP `PUT` semantics at the endpoint `/a1-policy/v2/policies/policy_id`, meaning that a policy update issued by a compromised rApp is processed by the A1-PMS through exactly the same code path as a legitimate one, there is no distinct "modification" operation whose invocation could be audited separately from creation. Second, the A1-PMS enforces only JSON Schema structural validation against the registered policy type schema: it verifies that required fields are present and that values fall within declared ranges, but performs no semantic validation of parameter distributions, no comparison against historical policy outputs from the same `service_id`, and no cross-referencing of the submitted `priorityLevel` or `guaranteedBitRate` values against observed network conditions. A policy that reduces `priorityLevel` from 8 to 6 and decreases `guaranteedBitRateDL` by 25% is structurally valid and passes validation without exception, provided the resulting values remain within the JSON Schema `minimum`/`maximum` bounds declared in the policy type definition. Third, once accepted by the A1-PMS, the policy is propagated southbound to the Near-RT RIC over the A1 transport protocol, HTTP/1.1 over TLS, and delivered to subscribed xApps. The Near-RT RIC and its xApps apply no independent validation of policy content provenance: they execute whatever `statement` object the A1-PMS delivers, trusting the Non-RT RIC platform to have enforced appropriate controls upstream. The attack therefore exploits the gap between what the JSON Schema enforces, i.e., structural validity, and what would be required to detect semantic manipulation, i.e., behavioural baseline comparison, a capability that is absent from both the OSC implementation and the current WG11 security specifications.

4) DETECTION AND ATTRIBUTION CHALLENGES

Several properties of this attack make it resistant to standard operational detection approaches. First, the malicious behavior is temporally separated from the onboarding event by a deliberate dormancy period, breaking the causal association that might otherwise trigger suspicion. Second, the policy manipulation operates within the statistical envelope of normal rApp behavior, making anomaly detection based on policy output distributions ineffective without a high-fidelity behavioral baseline established prior to the dormancy period's expiration. Third, the harm manifests as a service quality degradation rather than a security alert, routing the operator's response through capacity planning and vendor management processes rather than security incident response workflows. Attribution to the rApp is further complicated by the fact that the Near-RT RIC and xApps are functioning correctly meaning that standard RAN troubleshooting processes will not surface the Non-RT RIC policy pipeline as a candidate root cause without deliberate forensic investigation of A1 policy logs.

5) KEY IMPLICATIONS

This scenario illustrates that supply chain compromise does not require spectacular technical sophistication to be operationally damaging. The attack's effectiveness derives from its patience, its calibration to avoid detection thresholds, and its exploitation of the trust that the operator necessarily extends to onboarded rApps in the absence of more rigorous vetting infrastructure. It also highlights a structural gap in the current A1 interface security model: there is no standardized mechanism for the Near-RT RIC to authenticate the behavioral integrity of A1 policy sources, only their structural format compliance.

B. SCENARIO B: OVER-PRIVILEGED RAPP EXFILTRATES UE BEHAVIORAL TELEMETRY THROUGH AN UNSECURED SMO DATA PIPELINE

1) THREAT VECTORS INVOLVED

Data exfiltration through SMO telemetry pipelines; Lateral movement and cross-rApp compromise

2) SETUP

A national operator deploys a Non-RT RIC platform hosting twelve rApps from eight different vendors, covering functions ranging from energy efficiency optimization to predictive maintenance and interference management. The R1 interface's data exposure services are configured at the platform level, with authorization scopes defined during initial deployment. Due to time pressure during the rollout, several rApps are granted broad data access permissions rather than the granular, function-specific scopes that a mature least-privilege model would require. One of the deployed rApps, a legitimate energy efficiency optimization application, is authorized to access cell-level performance counters, load metrics, and handover event records across the operator's entire network footprint, a scope considerably wider than its core optimization function strictly requires.

3) ATTACK EXECUTION

The energy efficiency rApp is itself not malicious at the point of deployment. However, six months into operation, a previously undisclosed vulnerability in the rApp's container runtime is exploited by an external attacker who has been

conducting reconnaissance on the operator's vendor ecosystem. The vulnerability allows the attacker to achieve remote code execution within the rApp's container without triggering the platform's intrusion detection systems, which are tuned to monitor north-south traffic between the Non-RT RIC and external networks but have limited visibility into east-west traffic within the rApp execution environment.

Once inside the rApp container, the attacker inherits the application's existing R1 authorization credentials and data access permissions. Rather than immediately exfiltrating data, which would likely trigger volume-based anomaly detection, the attacker establishes a low-and-slow collection posture, issuing data queries at rates consistent with the rApp's normal operational behavior. Over several months, the attacker systematically collects handover event records and spatiotemporal load metrics spanning the operator's entire network. Cross-referencing this telemetry with commercially available mobility datasets, the attacker constructs detailed location history profiles for a subset of high-value targets, a capability of significant interest to state-level intelligence actors.

At the protocol level, the exfiltration exploits two distinct and compounding properties of the R1 data exposure implementation and the Non-RT RIC execution environment. The primary exfiltration channel exploits the push delivery model of the OSC Information Coordinator Service (ICS). Unlike a polling architecture in which anomalous query volume would be detectable at the ICS layer, the ICS delivers telemetry by issuing HTTP `POST` requests to the rApp's registered `job_result_uri` at the cadence declared in the ICS job definition, in this case `PT1M`, producing one delivery event per minute per subscribed measurement type. An attacker operating within the compromised rApp container receives these deliveries passively and forwards copies to an external collection endpoint without issuing any additional ICS API calls; the ICS therefore observes no anomaly at the subscription or delivery layer, and the absence of mandatory per-delivery audit logging in the current OSC ICS implementation means that no forensic record of the data's onward transmission is produced within the platform. The secondary lateral movement channel exploits a default Kubernetes behaviour that is particularly consequential in shared Non-RT RIC deployments: unless explicitly disabled via `automountServiceAccountToken: false` in the pod security context, every container is issued a service account token that authorises API calls to the Kubernetes control plane within the pod's namespace. In a Non-RT RIC deployment where multiple rApps share the `nonrtric` namespace and east-west network policies are not enforced between pods, a configuration gap consistent with the absence of normative isolation requirements in current WG11 specifications, the service account token enables namespace-wide service enumeration and direct HTTP access to co-resident rApp data sink endpoints, bypassing the ICS authorization model entirely. The combination of these two properties, i.e., a telemetrically rich push delivery model with no egress audit trail, and a shared execution environment with default credential exposure, creates the conditions for a persistent, layered exfiltration operation that is invisible to all monitoring instrumentation currently mandated by the O-RAN Alliance.

4) DETECTION AND ATTRIBUTION CHALLENGES

The attack's persistence and stealth derive from two compounding factors. First, the data access patterns are indistinguishable from legitimate rApp behavior because the attacker is operating through the rApp's own authorized credentials at normal query rates, a scenario that volume-based and signature-based detection systems are poorly positioned to identify. Second, the absence of mandatory fine-grained audit logging for R1 data access in current O-RAN Alliance specifications means that the operator lacks the forensic visibility necessary to reconstruct the access history even after the breach is eventually detected through external means. The lateral movement to the co-resident rApp compounds the attribution challenge: the secondary data access originates from within the platform's own service mesh, making it appear as legitimate inter-application communication rather than an intrusion event.

5) KEY IMPLICATIONS

This scenario demonstrates that data exfiltration risk in the rApp ecosystem does not require a malicious rApp, it requires only an over-privileged one and a vulnerable execution environment. The combination of broad R1 authorization scopes, immature east-west traffic monitoring within the rApp service mesh, and insufficient audit logging creates the conditions for a persistent, high-value intelligence collection operation that is invisible to standard operational security tooling. It further illustrates that the lateral movement risk identified in the previous section is not merely theoretical: the service mesh misconfiguration that enables cross-rApp pivot is a foreseeable consequence of deploying multiple third-party applications onto a shared platform without rigorous inter-application isolation and authentication requirements.

V. MITIGATION STRATEGIES AND RECOMMENDATIONS

The attack scenarios presented in the previous section underscore a consistent theme: the security gaps in the rApp ecosystem are not primarily the result of implementation failures by individual operators or vendors, but of structural insufficiencies in the specifications, standards, and architectural guidance that govern how rApps are developed, onboarded, and operated. Effective mitigation, therefore, requires action at multiple levels simultaneously, from the O-RAN Alliance's standardization workstreams to operators' own deployment and operational security practices. This section proposes a layered mitigation framework organized around five complementary strategies, each mapped to the threat vectors it addresses and evaluated against the current state of O-RAN Alliance specifications. For each strategy, an analysis of its operational cost is also provided.

FIGURE 4. Layered mitigation framework mapping five countermeasures (M1–M5) to the four identified threat vectors (T1–T4). Full coverage () denotes that the mitigation directly addresses the threat; partial coverage (~) denotes indirect or incomplete mitigation; (-) denotes no direct coverage. Mitigations are ordered from targeted (M1) to foundational (M5), reflecting increasing deployment scope and complexity. M5 (WG11 standardisation) is a foundational enabler underpinning all other mitigations.

The framework is depicted in Fig. 4. A mitigation receives **full coverage** (✓) for a given threat vector if: it directly and independently reduces the probability or impact of that threat through a mechanism that operates on the threat's primary exploitation path, requires no additional controls to be effective, and addresses the threat's root cause rather than a symptom. A mitigation receives **partial coverage** (~) if it reduces the threat's probability or impact but only under conditions that are not guaranteed by the mitigation itself, for example, because it addresses a necessary but not sufficient condition for the threat, because its effectiveness depends on correct configuration or calibration by the operator, or because it addresses a downstream consequence of the threat rather than its origin. A mitigation receives **no coverage** (—) if it has no direct causal relationship to the threat's exploitation path, even if it improves the platform's general security posture. Three specific assignments in Fig. 4 warrant explicit justification under this criterion.

First, M1 (code signing) receives full coverage for T1 (supply chain) with respect to distribution-channel integrity (a signed package guarantees provenance and tamper-detection), but the coverage is conditional on the signing authority's trustworthiness: a malicious yet legitimately signed publisher, whose certificate has not been revoked, bypasses this control entirely. This limitation is the reason M1 receives only partial coverage for T2 and T4: it reduces the probability that an rApp carrying malicious policy manipulation or lateral movement capability is introduced undetected, but only if the vendor's signing key has not been deliberately used to sign a malicious package.

Second, M2 (zero-trust API access, including mutual TLS) receives full coverage for T3 and T4 because mTLS provides both authentication and the cryptographic foundation for fine-grained per-instance authorization scopes, but this full coverage designation assumes that the authorization policy is correctly specified with least-privilege scopes at the R1 service binding stage. mTLS alone, without the accompanying least-privilege scope configuration, would warrant only partial coverage for T3. The full designation reflects the combination of mTLS and per-instance OAuth 2.0 scope enforcement as a single deployed control.

Third, M3 (runtime behavioral monitoring) receives full coverage for T1, T2, and T3 on the assumption that a clean behavioral baseline has been established during a

probationary deployment period before production authorization, as specified in Sec. V-C. Without this baseline, the anomaly detector has no reference distribution against which to measure deviation, and the coverage would degrade to partial or none depending on the chosen detector and the manipulation intensity, as the experimental results of Sec. VI quantify directly.

A. RAPP CERTIFICATION AND CODE SIGNING

The most direct response to supply chain and onboarding risk is the establishment of a mandatory rApp certification regime that imposes verifiable security requirements on rApp packages before they are admitted to operator platforms. Such a regime has two components: cryptographic code signing and security-focused functional certification.

Code signing requires rApp vendors to sign their packages with certificates issued by a trusted authority recognized by the Non-RT RIC platform. Upon onboarding, the platform verifies the package signature before proceeding with any further processing, assuring that the package originates from a trusted publisher. This control directly addresses the distribution-channel integrity risk and is technically straightforward to implement using established public key patterns. Nevertheless, current O-RAN specifications do not mandate code signing for rApp packages.

Security-focused functional certification goes further, requiring that rApp packages undergo defined security testing, including static code analysis, dependency vulnerability scanning, and behavioral assessment in an isolated sandbox environment, before receiving a certification mark that operators can use as a condition for onboarding eligibility. This is analogous to the security certification processes applied to network equipment under frameworks such as NESAS (Network Equipment Security Assurance Scheme) [31], adapted to the software-defined, rapid-release cadence of the rApp ecosystem. The O-RAN Alliance's WG11 has begun developing security test specifications relevant to this area, but a complete, enforceable certification framework for third-party rApps has not yet been published. Closing this gap is one of the most impactful near-term actions available to the standardization community.

1) OPERATIONAL COST ANALYSIS

Code signing imposes negligible runtime overhead. Signature verification at onboarding is a one-time cryptographic operation whose latency is measured in milliseconds and whose computational cost is trivially absorbed by the Non-RT RIC platform. The dominant costs of M1 are organisational rather than computational. Establishing and operating a certificate authority infrastructure trusted by all participating operators and rApp vendors requires governance arrangements that do not currently exist in the O-RAN ecosystem: a root of trust must be designated, certificate issuance and revocation processes must be defined, and cross-operator certificate recognition must be negotiated. In a multi-vendor marketplace, the certificate authority question alone involves significant standardisation and legal complexity analogous to the governance challenges that took the mobile application ecosystem years to resolve.

Security-focused functional certification, e.g., static code analysis, dependency scanning, and behavioral sandbox vetting, imposes a more immediate operational cost on rApp vendors in the form of longer release cycles and higher pre-publication compliance overhead. For vendors accustomed to the agile, rapid-release cadences common in cloud-native software development, a mandatory certification gate represents a meaningful change to their development workflow. Operators should anticipate that M1 will extend time-to-deployment for new rApp versions and plan their change management processes accordingly. A tiered certification model, requiring full certification for rApps granted write access to the A1 policy pipeline and lighter-weight validation for read-only or monitoring rApps, can reduce this burden while concentrating compliance overhead where the risk is highest.

B. ZERO-TRUST ARCHITECTURE FOR SMO API ACCESS

The data exfiltration scenario demonstrates that broad, role-level authorization scopes in the R1 interface create unacceptable latent risk in multi-vendor rApp deployments. The appropriate architectural response is the application of zero-trust [32], [33] principles to SMO API access, specifically, the enforcement of least-privilege authorization at the level of individual rApp and individual data access operations rather than at the level of application categories or deployment-time roles.

In a zero-trust [34] model for the rApp ecosystem, each rApp instance is issued a unique cryptographic identity at instantiation, and all access to R1 services, e.g., data exposure, policy management, AI/ML model services, is mediated through an authorization layer that evaluates each request against a fine-grained policy specifying exactly which data types, network segments, and operational parameters the requesting rApp is permitted to access. Authorization decisions are made dynamically at request time rather than statically at deployment time, allowing operators to adjust access scopes in response to changing operational requirements or emerging threat intelligence without redeploying the rApp itself.

Mutual TLS authentication between rApp instances and Non-RT RIC platform services is a prerequisite for this model, ensuring that both parties in each API interaction can verify each other's identity before exchanging data or executing policy operations. This directly addresses the service mesh misconfiguration exploited in Scenario B, where the absence of mutual authentication between co-resident rApps enabled cross-application lateral movement. Current O-RAN Alliance specifications reference mutual TLS as a recommended security control for R1 interface communications but do not mandate it uniformly across all R1 service interactions, a gap that leaves implementation-level

decisions to platform vendors whose security postures may vary significantly.

Operators deploying Non-RT RIC platforms should require platform vendors to demonstrate zero-trust API access controls as a procurement condition, and should conduct periodic authorization scope reviews to identify and remediate privilege creep in deployed rApp instances. The energy efficiency rApp in Scenario B accumulated a damaging data access footprint not through active exploitation at onboarding but through the passive accumulation of permissions that were never subsequently reviewed, a risk that zero-trust authorization hygiene directly mitigates.

1) OPERATIONAL COST ANALYSIS

Zero-trust API access controls impose two distinct cost categories. The first is runtime latency: every R1 API call from an rApp to the Non-RT RIC platform services must be authenticated and authorised dynamically, replacing the static role-based access control model that most current implementations use. In the OSC Non-RT RIC, R1 service interactions involve HTTP REST calls to the ICS, A1-PMS, and related services. Adding per-request token validation, OAuth 2.0 introspection or JWT verification, to each of these interactions introduces a latency increment of typically 2–10 ms per call depending on whether token validation is performed locally or via a remote authorisation server. For rApps operating on policy cycles of one second or longer, this overhead is negligible. For high-frequency data ingestion scenarios involving per-minute ICS deliveries across many cells, the cumulative latency impact across all concurrent rApp instances warrants measurement in the specific deployment environment before full rollout.

The second and more significant cost is configuration complexity. Defining and maintaining fine-grained per-instance authorisation policies for a dynamic rApp population, i.e., one where rApps are onboarded, updated, and terminated on operator-driven schedules, requires a policy administration infrastructure that must itself be secured, audited, and kept consistent with the rApp deployment state. Privilege creep, where authorisation scopes are defined broadly at onboarding and never subsequently reviewed, is the most common failure mode of least-privilege architectures in practice and requires active governance to prevent. Operators should establish periodic authorisation scope reviews, quarterly at minimum, and automate scope reduction triggers tied to rApp lifecycle events such as version updates and functional scope changes.

The mutual TLS requirement for east-west service mesh traffic imposes certificate management overhead on the platform, requiring automated certificate rotation to avoid service disruption from expired certificates, a failure mode that has caused significant outages in production Kubernetes deployments. Service mesh frameworks such as Istio can automate much of this management, but introduce their own operational complexity and represent an additional software dependency in the Non-RT RIC execution environment.

C. RUNTIME BEHAVIORAL MONITORING

Code signing and zero-trust access controls address the risks of malicious onboarding and over-privilege, but they do not fully address the risk of dormant malicious logic that activates after deployment, the attack pattern central to Scenario A. Detecting this class of threat requires runtime behavioral monitoring: continuous observation of rApp behavior during operation, compared against established baselines, with anomaly detection capable of identifying subtle deviations that may indicate malicious activation.

Effective runtime monitoring for rApps encompasses several dimensions. API call monitoring tracks the volume, frequency, timing, and parameter distributions of each rApp instance's interactions with R1 platform services, establishing a behavioral baseline during an initial observation period and flagging statistically significant deviations for investigation. Policy output monitoring tracks the content and distributional properties of A1 policies generated by each rApp, detecting shifts in policy directionality, such as the sustained QoS parameter skew in Scenario A, that diverge from the rApp's historical output profile. Resource consumption monitoring tracks CPU, memory, and network I/O patterns at the container level, providing a secondary signal for anomalous activity that may not be visible at the API layer.

The challenge of rApp runtime monitoring is that meaningful anomaly detection requires sufficiently rich behavioral baselines, and establishing those baselines demands a period of normal operation that a dormant malicious rApp is specifically designed to exploit. This argues for a monitoring architecture that maintains behavioral baselines continuously and with sufficient historical depth to detect slow drift as well as acute anomalies and for onboarding processes that include a mandatory monitored probationary period before an rApp is granted full production authorization. Neither of these practices is currently addressed in O-RAN Alliance specifications, representing a gap that platform vendors and operators must currently bridge through proprietary tooling.

1) OPERATIONAL COST ANALYSIS

Runtime behavioral monitoring is the most computationally intensive mitigation in the framework. Maintaining continuous per-rApp behavioral baselines across API call patterns, A1 policy output distributions, and resource consumption profiles requires persistent storage of sufficient historical depth (the experimental results of Sec. VI suggest that detection windows of at least 200 policy cycles are necessary for moderate-intensity manipulation detection) and statistical processing infrastructure that scales with the number of deployed rApp instances and the policy cycle frequency.

For a Non-RT RIC deployment hosting ten rApp instances each generating A1 policies on one-second cycles, the monitoring infrastructure must process and store approximately 864,000 policy records per day per rApp instance to maintain a 10-day baseline window. At this scale the storage and processing overhead is manageable with modern time-series database infrastructure, but it represents a non-trivial

addition to the platform's operational footprint that must be provisioned, maintained, and protected as a security-critical component.

The more challenging operational cost of M3 is the tuning burden associated with anomaly detection thresholds. A detection threshold set too low produces a high false positive rate, flagging legitimate policy adaptations driven by genuine network condition changes as anomalous, which creates alert fatigue and erodes operator trust in the monitoring system. A threshold set too high recreates the undetectability gap quantified in Sec. VI, where manipulation below the threshold remains undetected regardless of the observation window. Calibrating this threshold requires a sufficient baseline observation period before production deployment, during which the rApp must operate in a non-adversarial environment to establish a clean behavioral reference. The mandatory probationary period proposed in this subsection directly addresses this requirement but extends time-to-full-production for newly onboarded rApps.

An inherent limitation of M3 warrants acknowledgment. At a one-second A1 policy cycle cadence, the 200-cycle detection window required for reliable identification of moderate-intensity manipulation corresponds to a time-to-detection latency of approximately 3.3 minutes, during which degraded policies are actively executing and slice QoS is impaired. This latency is not a correctable implementation deficiency but an intrinsic property of distribution-based anomaly detection: narrower windows reduce latency but reduce statistical power proportionally, as Figure 5 demonstrates. Within the current framework, M2 (zero-trust API access) provides partial compensation by limiting the policy parameters a malicious rApp can affect independently of whether its behaviour has been detected. Fully closing the detection latency gap, for example through lightweight per-cycle consistency checks operating below the window timescale, remains an open problem and a priority for future work.

D. SANDBOXING AND CONTAINER ISOLATION

The lateral movement risk exploited in Scenario B is fundamentally a consequence of insufficient isolation between co-resident rApps on the Non-RT RIC platform. Strengthening container-level isolation is then a necessary complement to the access control and monitoring measures described above.

Robust sandboxing for rApp containers requires enforcement at multiple layers of the execution environment. At the kernel level, mandatory access control frameworks, such as SELinux or AppArmor policies, should constrain each container's system call surface to the minimum required for its legitimate function, limiting the blast radius of a container escape or privilege escalation exploit. At the network level, strict network policies should govern east-west traffic between co-resident rApp containers, enforcing a default-deny posture in which inter-application communication is permitted only where explicitly required by documented integration dependencies and authenticated through mutual TLS. At the storage level, shared file system resources accessible to multiple rApp containers should be minimized and access-logged, eliminating the passive data sharing channels that container deployments can inadvertently create.

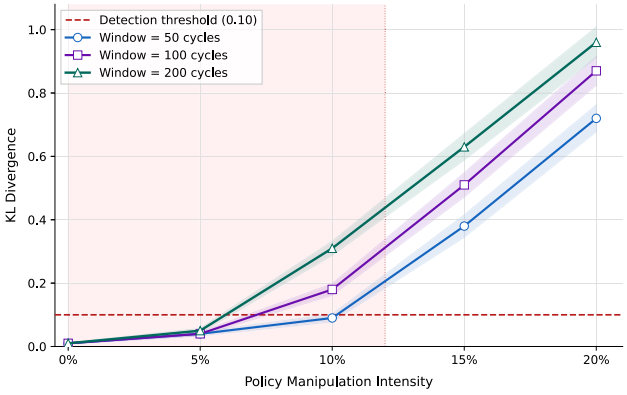

FIGURE 5. A1 policy manipulation detectability: KL divergence vs. Manipulation intensity by detection window size.

The O-RAN Alliance's current specifications provide high-level guidance on the need for isolation between Non-RT RIC applications but do not define the specific technical controls, such as kernel security modules, network policy enforcement points, and storage access controls, that platform vendors must implement to achieve adequate isolation. This leaves a significant implementation variance across vendor platforms that operators cannot currently assess or compare without conducting independent security evaluations. Defining minimum isolation requirements as normative specification language in WG11 would substantially reduce this variance and raise the baseline security posture of the rApp execution environment across the ecosystem.

1) OPERATIONAL COST ANALYSIS

Container-level sandboxing via mandatory access control frameworks such as SELinux or AppArmor imposes minimal runtime overhead under normal operating conditions. Performance benchmarks for SELinux-enforced container workloads consistently show small throughput penalties for network-bound applications, which is the dominant workload profile for rApps performing API interactions rather than intensive local computation. The primary operational cost of M4 is therefore not runtime performance but platform dependency and policy maintenance.

Defining and maintaining appropriate SELinux or AppArmor policy profiles for each rApp type requires security expertise that may not be available within operator network operations teams, whose skill sets are typically oriented toward telecommunications rather than Linux kernel security. This creates a dependency on either the Non-RT RIC platform vendor, who must supply default security profiles that operators can customise, or on third-party security expertise that operators must procure separately.

The absence of normative sandboxing requirements in current WG11 specifications means that vendor-supplied security profiles, where they exist at all, are not currently subject to any standardised adequacy assessment.

The default-deny east-west network policy requirement has a more operationally visible cost: it requires operators to explicitly enumerate and permit all legitimate inter-rApp communication dependencies before deployment. In a heterogeneous multi-vendor rApp environment where inter-application communication patterns may not be fully documented by rApp vendors, this enumeration process is non-trivial and may require network traffic analysis during a pre-production observation period to identify all legitimate communication flows before policy enforcement is activated. Activating default-deny network policies without completing this enumeration risks disrupting legitimate rApp interactions and degrading platform functionality.

E. STANDARDIZATION GAPS AND RECOMMENDED ACTIONS

The four mitigation strategies described above share a common dependency: their consistent, interoperable implementation across the multi-vendor O-RAN ecosystem requires a level of normative standardization that current O-RAN Alliance specifications do not yet provide.

The overarching recommendation emerging from the analysis conducted in this article is that the O-RAN Alliance's WG11 should elevate the security requirements governing the rApp ecosystem from their current status, largely informative, recommendation-based, and implementation-agnostic, to normative, testable, and enforceable specifications. This transition is consistent with the maturation trajectory observed in analogous open platform security frameworks, including those governing mobile operating system application ecosystems and cloud-native network function deployments. The rApp ecosystem is approaching a scale inflection point as O-RAN deployments accelerate globally; the window for embedding security-by-design principles into the foundational specifications before that inflection point is narrowing, and the cost of retrofitting security into a mature, widely deployed ecosystem is substantially higher than the cost of specifying it correctly from the outset.

Operators, for their part, should not wait for standardization to mature before acting. Procurement requirements that mandate zero-trust API controls, container isolation standards, and runtime monitoring capabilities from Non-RT RIC platform vendors can drive security improvements ahead of formal specification updates. Operators are ultimately the parties who bear the operational and reputational consequences of rApp ecosystem compromises, and they are positioned to use their procurement leverage to accelerate the security posture of the platforms they deploy.

1) OPERATIONAL COST ANALYSIS

M5 imposes no direct runtime cost on operators, on the contrary, its costs are borne by the O-RAN Alliance standardisation process and by the vendors and operators who must implement updated specifications. However, the indirect operational costs of standardisation are significant and deserve acknowledgment. Implementing normative security requirements that go beyond current WG11 specifications will require Non-RT RIC platform vendors to update their software stacks, re-qualify their implementations against new test specifications, and potentially revise their commercial contracts with operators. For operators, aligning existing deployments with updated specifications may require platform upgrades, revalidation of deployed rApp portfolios, and renegotiation of vendor support agreements.

The timeline for standardisation gap closure is the most significant operational cost dimension of M5. The O-RAN Alliance standardisation cycle operates on a cadence of approximately six to twelve months per specification release. Elevating rApp security requirements from informative to normative status, developing associated test specifications, and achieving broad implementation across the vendor ecosystem realistically requires a multi-year timeline. Operators deploying O-RAN infrastructure at scale during this period face a structural vulnerability window during which the mitigations of M1-M4 must compensate for the absence of normative standardisation.

VI. QUANTITATIVE VALIDATION: EMULATED RAPP THREAT ASSESSMENT

The threat vectors identified in Sec. III and IV are grounded in architectural analysis and illustrative scenarios. To complement this qualitative treatment with measurable evidence, this section presents a quantitative validation study conducted using an emulated rApp environment built on an open-source Non-RT RIC platform. The objective is not to demonstrate a fully operational exploit, which would require a production O-RAN deployment, but to quantify the detectability gap between malicious and benign rApp behavior under realistic operational conditions and to measure the data exposure surface created by over-privileged R1 authorization scopes. Two experiments are conducted, each targeting a distinct threat vector and producing results that directly support the analytical claims of the preceding sections.

A. EVALUATION METHODOLOGY

1) PLATFORM AND TOOLING

The emulation environment is built on the O-RAN Software Community (OSC) Non-RT RIC platform, release G, deployed as a containerised stack using Kubernetes 1.27 on a commodity server with 32 CPU cores and 64 GB RAM. The SMO layer is emulated using the OSC SMO package, with O1 telemetry feeds generated by a traffic emulator seeded with real-world LTE drive test datasets from the publicly available CRAWDAD [35] repository. Two rApp instances are deployed: a reference rApp implementing a standard load-balancing optimization function, and an adversarial rApp implementing the threat behaviours described in Scenarios A and B of Sec. IV. Both rApps are written in Python and

TABLE 3. Consolidated experimental parameters for reproducibility.

Parameter	Experiment 1 (E1)	Experiment 2 (E2)
Traffic and Telemetry Data		
CRAWDAD dataset	**CRAWDAD dataset roma/taxi** [36]: GPS mobility traces of 320 Rome taxi cabs used to synthesise per-cell UE handover event sequences	Same
Traffic emulator	Custom Python emulator seeding O1 PM counter generation from CRAWDAD LTE traces; outputs `PERFORMANCE_MEASUREMENTS_5G` ICS information type at `PT1M` cadence	Same emulator; additionally generates `UE_CONTEXT_RELEASE` and `PDCP_BYTES` counters synthesised from taxi GPS trace handover events
Emulation duration	100+50W cycles (2,600 / 5,100 / 10,100 cycles for W=50, W=100, and W=200 respectively)	24 simulated hours per run
A1 Policy Bias Parameterisation (E1 only)		
Bias model	Multiplicative per-parameter bias applied immediately before `PUT /a1-policy/v2/policies/{id}`; bias parameters $(\delta_p, \alpha, \beta)$ applied to `priorityLevel`, `guaranteedBitRate`, `packetDelayBudget` respectively	N/A
Bias levels evaluated	$\delta_p \in \{0, 1, 2, 3, 4\}$ (0%–20% of `priorityLevel` range [1,10]); $\alpha \in \{1.0, 0.95, 0.90, 0.85, 0.80\}$; $\beta \in \{1.0, 1.05, 1.10, 1.15, 1.20\}$; all three applied simultaneously	N/A
Policy cycle interval	1 second	N/A
Statistical Parameters		
Independent runs	10 runs per (bias level, window size) configuration; random seed sequence	10 runs per authorisation profile; same seed sequence
Detection window sizes	$W \in \{50, 100, 200\}$ cycles	N/A
Windows per run	50 non-overlapping windows of size W extracted per run (100 cycles discarded as warm-up at run start)	Entropy sampled at 2-hour intervals; reconstruction evaluated at 1-hour intervals
Confidence intervals	95% CI	95% CI
Heuristic thresholds	Detection threshold 0.10 bits; window sizes 50/100/200 cycles (see Section VI-A)	Cosine similarity threshold 0.85; 24-hour observation window (see Section VI-A)

interact with the Non-RT RIC via the R1 interface using the OSC Information Coordinator Service (ICS) API. Unless otherwise stated, the numerical thresholds used in the validation experiments are heuristic choices selected to cover a representative operational range and are not claimed to be optimal. Operators deploying these controls in practice should recalibrate all thresholds against their specific rApp population and traffic baseline. Table 3 provides the consolidated experimental parameters.

2) EXPERIMENT DESIGN

Two experiments are conducted:

- **Experiment 1 (E1)** targets Threat Vector T2 (Policy Manipulation via the A1 Interface) and quantifies the detectability of malicious A1 policy behaviour under increasing manipulation intensity, measured against a benign baseline. The adversarial rApp introduces a controlled QoS parameter bias into its A1 policy outputs, with bias magnitude varied across five levels from 0%

(benign) to 20% of the nominal parameter range. Policy output distributions are compared using the Kullback-Leibler (KL) divergence between the adversarial and benign policy streams, measured over non-overlapping windows of 50, 100, and 200 consecutive policy cycles.

- **Experiment 2 (E2)** targets Threat Vector T3 (Data Exfiltration via SMO Telemetry Pipelines) and quantifies the data exposure surface created by over-privileged R1 data access scopes. Three rApp authorization profiles are evaluated: a minimal scope (access limited to cell-level load metrics), a moderate scope (load metrics plus handover event records), and a broad scope (all available O1 telemetry including UE behavioral aggregates). For each profile, the cumulative information exposure is measured using normalised entropy of the accessible data distribution over a 24-hour emulation period, and the time required for an adversarial rApp operating at normal query rates to reconstruct a statistically significant UE mobility signature is recorded.

3) METRICS

The primary metrics are: (i) KL divergence between adversarial and benign A1 policy output distributions as a function of manipulation intensity and detection window size (E1); and (ii) normalised data exposure entropy and UE signature reconstruction time as a function of R1 authorization scope breadth (E2). All results are averaged over ten independent emulation runs with different random seeds to account for traffic variability, and 95% confidence intervals are reported.

B. EXPERIMENT 1: A1 POLICY MANIPULATION DETECTABILITY

Figure 5 presents the KL divergence between the adversarial and benign A1 policy output distributions as a function of manipulation intensity, for three detection window sizes.

The dashed line denotes a nominal detection threshold of 0.10 bits, defined as the 99th percentile of the zero-manipulation divergence across all runs (false positive rate <1%). The shaded region (<12% bias) represents the practical undetectability zone where manipulation is statistically indistinguishable from normal policy variation regardless of window size. *Metric specification*: KL divergence computed as the mean of five per-parameter marginal divergences (parameters: `priorityLevel`, `guaranteedBitRateDL`, `guaranteedBitRateUL`, `packetDelayBudget`, `packetErrorRate`), each normalised to $[0, 1]$ by declared JSON Schema bounds; distributions estimated by KDE with Gaussian kernel and Scott's rule bandwidth [37]; logarithm to base 2, result in bits; numerical integration over 512-point uniform grid with $\epsilon = 10^{-9}$ smoothing floor. Results averaged over $T = 500$ windows from 10 independent runs.

Limitation: averaging marginal KL divergences across parameters ignores inter-parameter correlations; a malicious rApp that manipulates combinations of parameters while keeping each individual marginal distribution within the benign envelope would evade this detector, as the mean marginal divergence would remain near zero even under coordinated multi-parameter bias. A joint or multivariate divergence metric, such as the KL divergence between full K-dimensional joint densities estimated by multivariate KDE, would be more sensitive to correlated manipulation but requires a sample size per window that grows with the intrinsic dimensionality of the joint distribution, making it infeasible at the window sizes evaluated here. We leave this extension to future work.

Several findings emerge directly from the results. First, at low manipulation intensities (5% bias), KL divergence remains below 0.05 across all window sizes, indicating that subtle policy manipulation is statistically indistinguishable from normal inter-rApp policy variation. This quantitatively confirms the analytical claim in Sec. III-B that low-intensity policy bias evades threshold-based detection. Second, detectability improves substantially with window size: at 10% bias, a window of 200 cycles yields a KL divergence of 0.31, compared to 0.09 for a window of 50 cycles–a 3.4× improvement. This result has a direct operational implication: effective detection of moderate-intensity policy manipulation requires maintaining policy output histories of at least 200 cycles (approximately 3.3 minutes at a 1-second policy interval), a requirement that current O-RAN Alliance monitoring specifications do not address. Third, at manipulation intensities of 15% and above, divergence values exceed 0.5 for all window sizes, suggesting that aggressive manipulation is detectable even with short observation windows–but at the cost of network impact that would already be operationally significant before detection occurs.

The results establish a detectability gap region between 0% and approximately 12% bias intensity where manipulation is practically undetectable without extended observation windows and baseline profiling infrastructure that is not currently mandated in Non-RT RIC deployments.

C. EXPERIMENT 2: R1 DATA EXPOSURE SURFACE

Fig. 6 presents the normalised data exposure entropy accumulated over 24 hours for each authorization profile.

Broad-scope authorization saturates at 0.89 within ~20 hours; minimal-scope authorization plateaus at 0.31, reflecting the structurally smaller observable space of a two-field subscription. *Metric specification*: joint empirical Shannon entropy $H(\mathcal{D}_\tau)$ computed over an M-dimensional histogram with $B = 32$ equal-width bins per measurement dimension ($M = 2$, 5, 8 for minimal, moderate, and broad scope respectively); normalised by $H_{\max} = \log_2 |\mathcal{B}^*|$ where $\mathcal{B}^*$ is the set of non-zero bins observed across the full 24-hour window; logarithm to base 2, result in bits prior to normalisation; $\tilde{H} \in [0, 1]$ with $\tilde{H} = 1$ denoting maximally uniform occupancy of the observed support. Results averaged over 10 independent runs; shaded bands show 95% CI.

The figure reveals a non-linear relationship between authorization scope breadth and information exposure.

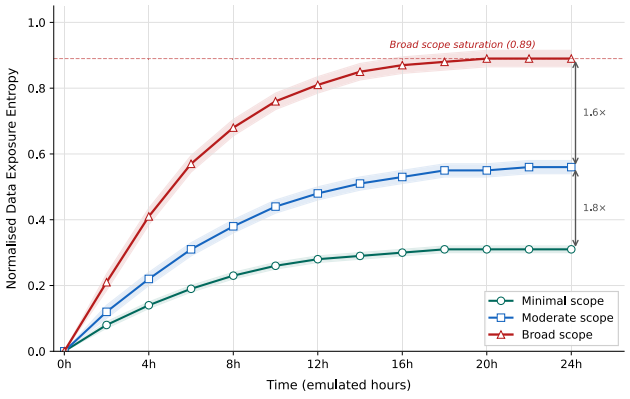

FIGURE 6. Cumulative data exposure entropy by R1 authorization profile (24-hour emulation window).

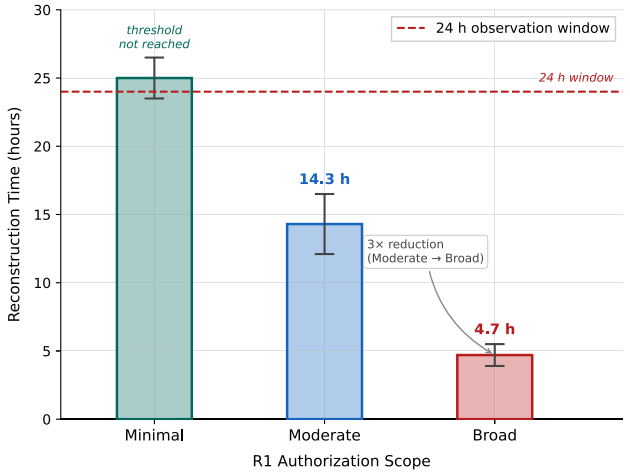

FIGURE 7. Mean time to reconstruct a UE mobility signature at cosine similarity ≥ 0.85 with the ground-truth vector, by R1 authorisation scope.

Moving from minimal to moderate scope increases normalised entropy by 1.8× (from 0.31 to 0.56), reflecting the addition of handover event records which carry significant spatiotemporal information. Moving from moderate to broad scope produces a further 1.6× increase (from 0.56 to 0.89), driven by access to UE behavioral aggregates. The broad-scope profile achieves near-maximum entropy (0.89 out of 1.0) within the 24-hour observation window, indicating that an rApp operating under broad R1 authorization can accumulate a near-complete information picture of network behavior within a single operational day.

Fig. 7 presents the time required to reconstruct a statistically significant UE mobility signature (defined as achieving >0.85 cosine similarity with the ground-truth mobility vector) as a function of authorization scope. Error bars show 95% confidence intervals over 10 runs. Minimal-scope authorization prevents reconstruction within the 24-hour observation window, confirming the protective effect of least-privilege R1 access controls. *Metric specification*: UE mobility signature defined as the flattened $N \times S$ cell-by-timeslot matrix of normalised `HANDOVER_SUCCESS_RATE` values (N cells, $S = 24$ one-hour slots) reconstructed from accumulated ICS records with linear interpolation for missing cell-slot entries; similarity measured by cosine similarity; minimal-scope profile excludes `HANDOVER_SUCCESS_RATE` from its ICS subscription filter and is therefore treated as infeasible.

Under the minimal authorization profile, UE mobility signature reconstruction does not achieve the 0.85 similarity threshold within the 24-hour window, confirming that least-privilege authorization effectively limits inference attack feasibility. Under the moderate profile, reconstruction is achieved after 14.3 hours on average (95% CI: 12.1–16.5 hours). Under the broad profile, reconstruction is achieved after just 4.7 hours on average (95% CI: 3.9–5.5 hours), approximately a threefold reduction relative to the moderate profile and within a timeframe that makes a single operational shift sufficient for a persistent exfiltration operation to yield actionable intelligence.

These results quantify the operational significance of R1 authorization scope granularity and provide empirical support for the least-privilege recommendation in Sec. V-B: the difference between minimal and broad authorization scopes is not merely a theoretical security preference but a measurable factor that reduces inference attack feasibility from hours to infeasible within a standard observation window.

VII. OPEN CHALLENGES AND FUTURE DIRECTIONS

The mitigation framework proposed in the previous section addresses the most pressing near-term security risks in the rApp ecosystem, but it would be misleading to suggest that implementing those measures resolves the security challenge comprehensively. The O-RAN architecture is not a static target; on the contrary, it is evolving rapidly, and several emerging developments will introduce new threat dimensions that current security thinking has not fully anticipated. This section identifies four open challenges that the research community, standardization bodies, and industry stakeholders will need to confront as the rApp ecosystem matures.

A. AI-NATIVE RAPPS AND ADVERSARIAL MACHINE LEARNING

The rApp model is predicated on AI/ML-driven network optimization, and the sophistication of the machine learning components embedded in rApps is expected to increase substantially as the ecosystem matures. Current rApp deployments largely employ relatively interpretable optimization models whose policy outputs can, in principle, be audited by network engineers with domain expertise. The trajectory of the field points toward increasingly complex, opaque neural network architectures whose decision-making processes are not readily interpretable even by their developers, a property that has profound security implications in the rApp context.

As rApps incorporate more sophisticated AI components, the attack surface associated with the AI/ML pipeline itself expands correspondingly. Model poisoning attacks, in which

an adversary corrupts the training data or fine-tuning process of an rApp's internal model to induce systematically biased policy outputs, become more difficult to detect as model complexity increases, because the relationship between training inputs and policy outputs becomes less transparent. Model inversion attacks, in which an adversary with query access to a deployed rApp model reconstructs sensitive information about the training dataset, create a new class of data exposure risk that operates entirely within the bounds of legitimate rApp API interactions. Adversarial input attacks [38], [39], [40], in which carefully crafted network telemetry inputs are used to manipulate an rApp model's outputs in targeted ways, represent a sophisticated variant of the policy manipulation threat that specifically exploits the properties of neural network decision boundaries.

The interpretability and robustness of AI components in rApps is currently outside the scope of O-RAN Alliance security specifications, and the broader field of adversarial machine learning in telecommunications contexts remains at an early stage. Developing evaluation frameworks for rApp AI/ML robustness, analogous to the adversarial robustness benchmarks emerging in the computer vision and natural language processing communities, is a research priority that will become increasingly urgent as AI-native rApps enter production deployment.

B. MULTI-OPERATOR RAPP MARKETPLACES AND TRUST BOUNDARIES

The scenarios analyzed in this article assume a relatively contained deployment model in which a single operator onboards rApps onto its own Non-RT RIC platform from a set of known third-party vendors. The industry trajectory, however, points toward more complex multi-operator rApp marketplace models in which applications are developed once and deployed across multiple operators' platforms simultaneously, potentially spanning different regulatory jurisdictions, network configurations, and threat environments.

This multi-operator deployment model introduces trust boundary challenges that the current rApp security architecture is not designed to address. An rApp operating across multiple operator platforms simultaneously accumulates telemetry and behavioral data from each deployment context, creating a cross-operator data aggregation capability that no individual operator has visibility into or control over. If that rApp is compromised, or if the vendor operates in bad faith, the resulting data exposure spans multiple operators' networks and potentially multiple national jurisdictions, with attribution and remediation responsibilities that are deeply ambiguous under current regulatory frameworks.

The governance model for multi-operator rApp marketplaces and how telemetry data collected across operator contexts is governed, remains almost entirely unaddressed in current O-RAN Alliance specifications and in the broader telecommunications regulatory landscape. Establishing a governance framework for multi-operator rApp ecosystems is a challenge that exceeds the scope of any single standards working group and will require coordinated engagement between the O-RAN Alliance, 3GPP SA3, GSMA, and national regulatory authorities.

C. REGULATORY FRAMEWORKS AND LIABILITY ATTRIBUTION

The introduction of third-party software into the core intelligence layer of critical national communications infrastructure raises regulatory and liability questions that the existing telecommunications regulatory framework was not designed to answer. Mobile network operators have historically operated under regulatory regimes that hold them responsible for the security and resilience of their networks as integrated wholes, a model that maps cleanly onto the vertically integrated RAN architectures of previous network generations. The rApp ecosystem complicates this model fundamentally by introducing a new class of actor, the third-party rApp vendor, whose software has direct operational influence over network behavior but who sits outside the operator's direct organizational control and, in many cases, outside the scope of existing telecommunications security regulations.

When an rApp-mediated security incident results in network degradation, data exposure, or service disruption, the attribution of regulatory liability between the operator, the Non-RT RIC platform vendor, and the rApp developer is genuinely unclear under current frameworks. Operators may argue that they exercised reasonable diligence in onboarding and operating the rApp according to available specifications. Platform vendors may argue that their platforms performed as specified. rApp vendors may argue that their software was deployed in an environment that did not meet the security requirements their application assumed. This diffusion of accountability is not merely a legal inconvenience, it creates structural incentives for underinvestment in security across the entire supply chain, because no single party bears the full consequence of a collective security failure.

Addressing this challenge requires regulatory innovation as well as technical standardization. National telecommunications regulators and cybersecurity authorities, including bodies such as ENISA in Europe and CISA in the United States, are beginning to engage with O-RAN security as a policy concern, but specific regulatory guidance addressing the rApp vendor liability question has not yet emerged. Developing clear liability attribution frameworks for rApp-mediated incidents, analogous to the product liability frameworks applied to physical network equipment, is a regulatory priority that should proceed in parallel with technical standardization efforts rather than waiting for those efforts to mature.

D. TOWARDS A TRUSTED RAPP ECOSYSTEM

The three challenges described above, adversarial AI, multi-operator governance, and regulatory liability, share a common underlying theme: the rApp ecosystem is being built on a foundation of trust assumptions that have not been explicitly defined, verified, or institutionalized. The path toward a

genuinely trusted rApp ecosystem requires not just the technical controls and standards updates proposed in Sec. V, but a broader shift in how the O-RAN community conceptualizes the relationship between openness and security.

The mobile application ecosystem analogy is instructive here not only as a risk frame but as a roadmap. Mobile operating system platforms spent years iterating on the balance between developer freedom and platform security, developing code signing infrastructure, permission models, behavioral monitoring systems, and certification processes through a combination of incident response, regulatory pressure, and competitive differentiation on security. The O-RAN community has the advantage of being able to learn from that experience rather than repeating it, but only if the lessons are applied proactively rather than reactively.

The most important of those lessons is that security in open platform ecosystems is not achieved through any single control but through the accumulation of overlapping, mutually reinforcing mechanisms that collectively raise the cost and complexity of exploitation to the point where the ecosystem's benefits outweigh its risks. Code signing alone is insufficient. Zero-trust access controls alone are insufficient. Runtime monitoring alone is insufficient. It is the combination of certification, isolation, access control, monitoring, governance, and regulatory accountability 'that produces a trustworthy platform. Building that combination into the rApp ecosystem before large-scale O-RAN deployments make its absence consequential is the central security challenge facing the O-RAN community today.

VIII. CONCLUSION

The threat surface introduced by third-party rApps on the Non-RT RIC is neither marginal nor hypothetical. It is a direct and foreseeable consequence of architectural choices that are central to the O-RAN model: open interfaces, shared execution environments, third-party software with privileged policy access, and rich telemetry pipelines exposed to applications whose provenance and behavior cannot be fully verified at onboarding. The four threat vectors identified in this article, supply chain and onboarding vulnerabilities, A1 policy manipulation, SMO telemetry exfiltration, and lateral movement across the rApp service mesh, each arise from these architectural properties, and each has a plausible exploitation path that current O-RAN Alliance specifications do not adequately address.

The mitigation framework proposed in this paper, encompassing rApp certification and code signing, zero-trust API access controls, runtime behavioral monitoring, and container-level sandboxing, provides a technically grounded response to these threats. But its effectiveness depends on a transition that the O-RAN Alliance's security workstreams have not yet completed: the elevation of rApp security requirements from informative recommendations to normative, testable, and enforceable specifications. Until that transition occurs, the security posture of the rApp ecosystem will remain a function of individual vendor and operator choices rather than a consistent, verifiable baseline, an outcome that is inadequate for infrastructures of this criticality.

The open challenges identified in this paper, i.e., adversarial AI in rApp models, multi-operator marketplace governance, and regulatory liability attribution, signal that the security demands on the rApp ecosystem will grow more complex, not less, as O-RAN deployments scale and mature. Meeting those demands requires the O-RAN community to act now, while the architecture is still being defined and the deployment base is still tractable, rather than after the ecosystem has reached a scale at which retrofitting security controls becomes prohibitively disruptive.

The central argument of this article can be stated plainly: openness and security are not competing objectives in the O-RAN rApp ecosystem, they are complementary ones, but only if security is treated as a foundational design requirement rather than an operational afterthought. The rApp ecosystem has the potential to become one of the most powerful tools available to mobile network operators for intelligent, adaptive network management. Whether it also becomes one of their most significant security liabilities will be determined by the decisions that standardization bodies, platform vendors, rApp developers, and operators make in the near term. The analysis and recommendations presented here are offered as a contribution to those decisions.

REFERENCES

[1] *O-ran Use Cases Analysis Report*, O-RAN Alliance, Alfter, Germany, 2026.

[2] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1376–1411, 2nd Quart., 2023.

[3] J. Groen et al., "Implementing and evaluating security in O-RAN: Interfaces, intelligence, and platforms," *IEEE Netw.*, vol. 39, no. 1, pp. 227–234, Jan. 2025.

[4] S. D'Oro, L. Bonati, M. Polese, and T. Melodia, "OrchestRAN: Network automation through orchestrated intelligence in the open RAN," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2022, pp. 270–279.

[5] Y. Rumesh, D. Attanayaka, P. Porambage, J. Pinola, J. Groen, and K. Chowdhury, "Federated learning for anomaly detection in open RAN: Security architecture within a digital twin," in *Proc. Joint Eur. Conf. Netw. Commun. 6G Summit (EuCNC/6G Summit)*, Jun. 2024, pp. 877–882.

[6] J. Groen et al., "TIMESAFE: Timing interruption monitoring and security assessment for fronthaul environments," *ACM Trans. Privacy Secur.*, vol. 29, no. 1, pp. 1–30, Dec. 2025.

[7] W. Tiberti, E. Di Fina, A. Marotta, and D. Cassioli, "Impact of man-in-the-middle attacks to the O-RAN inter-controllers interface," in *Proc. IEEE Future Netw. World Forum (FNWF)*, Oct. 2022, pp. 367–372.

[8] K. Sowjanya, R. Saini, D. Saha, K. Joshi, and M. Das, "ABElity: Attribute based encryption for securing RIC communication in open RAN," in *Proc. Workshop Secur. Privacy Next-Gener. Netw.*, 2025, pp. 1–5.

[9] *Security Requirements and Controls Specification, O-RAN Alliance*, Alfter, Germany, 2026.

[10] *Security Architecture and Procedures for 5G System*, document TS 33.501, 3GPP, 2023.

[11] *Non-RT RIC and A1 Interface: Use Cases and Requirements, O-RAN Alliance*, Alfter, Germany, 2026.

[12] *R1 Interface: General Aspects and Principles*, O-RAN Alliance, Alfter, Germany, 2023.

[13] *A1 Interface: General Aspects and Principles*, O-RAN Alliance, Alfter, Germany, 2026.

[14] R. F. Olimid and G. Nencioni, "5G network slicing: A security overview," *IEEE Access*, vol. 8, pp. 99999–100009, 2020.

[15] *O-RAN Operations and Maintenance Architecture, O-RAN Alliance*, Alfter, Germany, 2026.

[16] P. Ladisa, H. Plate, M. Martinez, and O. Barais, "SoK: Taxonomy of attacks on open-source software supply chains," in *Proc. IEEE Symp. Secur. Privacy (SP)*, San Francisco, CA, USA, May 2023, pp. 1509–1526.
[17] M. Ohm, H. Plate, A. Sykosch, and M. Meier, "Backstabber's knife collection: A review of open source software supply chain attacks," in *Proc. DIMVA*, 2020, pp. 23–43.
[18] L. Gehrke, "Dissertation research description: The potential of SBOMs to increase software supply chain security," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2025, pp. 4881–4883.
[19] T.-J. Wu, C.-F. Hung, W.-H. Dai, and S.-M. Cheng, "Malicious RAPPS for eavesdropping and data exfiltration in o-ran non-RT RIC," in *Proc. IEEE CANDAR*, Nov. 2025, pp. 188–194.
[20] S. Sultan, I. Ahmad, and T. Dimitriou, "Container security: Issues, challenges, and the road ahead," *IEEE Access*, vol. 7, pp. 52976–52996, 2019.
[21] T. Bui, "Analysis of Docker security," 2015, *arXiv:1501.02967*.
[22] A. Shostack, *Threat Modeling: Designing for Security*. Hoboken, NJ, USA: Wiley, 2014.
[23] (May 2023). *Open RAN Security Report*. [Online]. Available: https://www.ntia.gov/report/2023/open-ran-security-report
[24] R. Chandramouli. (Aug. 2019). *Security Strategies for Microservices-Based Application Systems*. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/204/final
[25] (2023). *CIS Kubernetes Benchmark*. [Online]. Available: https://www.cisecurity.org/benchmark/kubernetes
[26] (May 2022). *Cloud Native Security Whitepaper*. [Online]. Available: https://www.cncf.io/wp-content/uploads/2022/06/CNCF-native-security-whitepaper-May2022-v2.pdf
[27] A. Chiejina, B. Kim, K. Chowhdury, and V. K. Shah, "System-level analysis of adversarial attacks and defenses on intelligence in o-ran based cellular networks," in *Proc. ACM WiSec*, 2024, p. 237, doi: 10.1145/3643833.3656119.
[28] C.-F. Hung, Y.-R. Chen, C.-H. Tseng, and S.-M. Cheng, "Security threats to xApps access control and E2 interface in O-RAN," *IEEE Open J. Commun. Soc.*, vol. 5, pp. 1197–1203, 2024.
[29] N. N. Sapavath, B. Kim, K. Chowdhury, and V. K. Shah, "Experimental study of adversarial attacks on ml-based xapps in o-ran," in *Proc. IEEE GLOBECOM* Dec. 2023, pp. 6352–6357.
[30] H. Alimohammadi, S. Mayhoub, S. Chatzimiltis, M. Shojafar, and M. N. M. Bhutta, "Toward a multi-layer defence framework for securing near-real-time operations in open RAN," *IEEE Open J. Commun. Soc.*, vol. 7, pp. 480–497, 2026.
[31] (2025). *Network Equipment Security Assurance Scheme*. [Online]. Available: https://www.gsma.com/solutions-and-impact/technologies/security/gsmaresources/fs-13-network-equipment-security-assurance-scheme-overview-2/
[32] N. Nahar, K. Andersson, O. Schelén, and S. Saguna, "A survey on zero trust architecture: Applications and challenges of 6G networks," *IEEE Access*, vol. 12, pp. 94753–94764, 2024.
[33] N. F. Syed, S. W. Shah, A. Shaghaghi, A. Anwar, Z. Baig, and R. Doss, "Zero trust architecture (ZTA): A comprehensive survey," *IEEE Access*, vol. 10, pp. 57143–57179, 2022.
[34] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," *NIST Special Publication*, Tech. Rep. SP.800-207, 2020, doi: 10.6028/NIST.SP.800-207.
[35] *Community Resource for Archiving Wireless Data At Dartmouth*. Accessed: May 1, 2026. [Online]. Available: https://crawdad.org/
[36] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, "Crawdad roma/taxi," 2022, doi: 10.15783/C7QC7M.
[37] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*. Hoboken, NJ, USA: Wiley, 1992.
[38] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, Dec. 2018, doi: 10.1016/j.patcog.2018.07.023.
[39] L. Lyu et al., "Privacy and robustness in federated learning: Attacks and defenses," 2020, *arXiv:2012.06337*.
[40] Y. E. Sagduyu, T. Erpek, and Y. Shi, "Adversarial machine learning for 5G communications security," 2021, *arXiv:2101.02656*.

ROBERTO RIGGIO (Senior Member, IEEE) received the Ph.D. degree from the University of Trento, Italy. He was a Post-Doctoral Fellow with the University of Florida; a Researcher/Chief Scientist with CREATE-NET, Trento, Italy; the Head of Unit with FBK, Trento; a Senior 5G Researcher with the i2CAT Foundation, Barcelona, Spain; and a Senior Researcher with RISE AB, Stockholm, Sweden. He is an Associate Professor with the Università Politecnica delle Marche, Ancona, Italy. He has published more than 130 papers in internationally refereed journals and conferences. His research interests revolve around optimization and algorithmic problems in networked and distributed systems. His current fields of application are edge automation platforms, intelligent networks, and serverless computing.