

UNIVERSITÀ POLITECNICA DELLE MARCHE
Repository ISTITUZIONALE

The Convergence of Digital Twins and rApps for Proactive RAN Management and What-If Analysis

This is the peer reviewd version of the followng article:

Original
The Convergence of Digital Twins and rApps for Proactive RAN Management and What-If Analysis / Riggio, R.. - In: IEEE COMMUNICATIONS MAGAZINE. - ISSN 0163-6804. - (2026). [Epub ahead of print] [10.1109/mcom.001.2600251]

Availability:
This version is available at: 11566/363952 since: 2026-10-06T06:14:36Z

Publisher:

Published
DOI:10.1109/mcom.001.2600251

Terms of use:

The terms and conditions for the reuse of this version of the manuscript are specified in the publishing policy. The use of copyrighted works requires the consent of the rights' holder (author or publisher). Works made available under a Creative Commons license or a Publisher's custom-made license can be used according to the terms and conditions contained therein. See editor's website for further information and terms and conditions.
This item was downloaded from IRIS Università Politecnica delle Marche (https://iris.univpm.it). When citing, please refer to the published version.

Publisher copyright:

IEEE - Postprint/Author's Accepted Manuscript

©2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. To access the final edited and published work see 10.1109/mcom.001.2600251

(Article begins on next page)

01 November 2026

The Convergence of Digital Twins and rApps for Proactive RAN Management and What-If Analysis

Roberto Riggio
Università Politecnica delle Marche, Ancona, Italy
Email: r.riggio@univpm.it

***Abstract*—The growing complexity of 5G and beyond radio access networks (RANs) demands a shift from reactive management toward proactive, intelligence-driven optimization. The O-RAN Alliance's Non-Real-Time RAN Intelligent Controller (Non-RT RIC) resides in the Service and Management (SMO) framework and introduces rApps as modular applications for long-timescale RAN optimization. Concurrently, digital twin technology for RANs (DT-RAN) has emerged as a means of constructing high-fidelity virtual replicas of physical infrastructure for safe offline simulation and predictive analytics. Despite their complementary strengths, these technologies remain largely disconnected: rApps lack a simulation backend for pre-deployment policy validation, while DT-RAN platforms operate outside the O-RAN programmable control framework. This article proposes a unified architecture that embeds DT-RAN capabilities into the SMO framework, exposing the digital twin as a native simulation and enrichment service through standardized interfaces. A what-if analysis engine enables rApps to define hypothetical scenarios, execute them in a digital twin sandbox, and evaluate multi-objective outcomes before committing policies to the live network. The framework is validated through two use cases demonstrating meaningful improvements over reactive baselines.**

***Index Terms*—O-RAN, digital twin, Non-RT RIC, rApps, what-if analysis, 5G/6G.**

I. Introduction

The disaggregation of RANs under the O-RAN Alliance architecture has unlocked new possibilities for intelligent, vendor-neutral network management. Central to this vision is the RIC, split into a Near-Real-Time RIC (Near-RT RIC) operating at sub-second timescales and a Non-RT RIC operating at timescales above one second [1], [2]. The Non-RT RIC hosts rApps [3], modular third-party applications that leverage the R1 interface to consume platform services and issue policy guidance, enrichment information, and AI/ML model management directives to the Near-RT RIC.

While rApps represent a powerful abstraction for programmable RAN control, their current operational model is fundamentally reactive. Policies are derived from historical performance data and offline-trained models, then deployed into the live network without a mechanism to anticipate their impact under varying conditions. In an era of dense heterogeneous deployments, dynamic network slicing, and stringent energy targets, this trial-and-error approach carries non-trivial risks: degraded quality of experience, SLA violations, and suboptimal resource utilization.

Digital twin technology offers a compelling remedy. A DT-RAN constructs a high-fidelity virtual replica of the physical network that can be queried, stressed, and reconfigured without consequence to live operations [4]. Recent momentum from the O-RAN Alliance's next-Generation Research Group (nGRG), which published its DT-RAN key enablers report in 2024 [5], alongside 3GPP's ongoing study of management aspects of network digital twins in Release 19, confirms that DT-RAN is transitioning from a research concept to an architectural building block.

Yet today, these two technologies evolve in parallel rather than in concert. DT-RAN platforms are typically standalone simulation environments, disconnected from the Non-RT RIC's programmable service framework. Conversely, rApps have no standardized mechanism to invoke a simulation backend, parameterize what-if scenarios, or consume twin-generated predictions as enrichment information. This gap prevents operators from realizing a closed-loop paradigm in which candidate policies are validated in a virtual sandbox before being committed to the network.

This article bridges that gap by proposing a unified architecture that positions the DT-RAN Engine within the SMO framework and exposes it to rApps as a native service through the Non-RT RIC's R1 interface. A what-if analysis engine allows rApps to define hypothetical scenarios and evaluate multi-objective KPI outcomes in a digital twin sandbox before committing A1 policies. The framework is validated across two use cases, namely proactive energy saving and failure impact assessment.

II. Background and Related Work

A. O-RAN Architecture and the Non-RT RIC

rApps are modular, third-party applications that connect to the Non-RT RIC framework via the R1 interface, which exposes platform services such as data management and exposure (DME), A1 policy management, and AI/ML workflow orchestration [3]. Functionally, rApps fall into three categories: *policy-based rApps* that formulate A1 policies (e.g., traffic steering) for enforcement by Near-RT RIC xApps; *enrichment rApps* that produce predictive information such as traffic forecasts; and *AI/ML workflow rApps* that manage the training, deployment, and monitoring lifecycle of ML models. Recent WG2 specifications have introduced R1 conformance testing and A1 policy conflict detection [6].

Despite this progress, the current rApp operational model remains data-retrospective: insights are derived from historical traces and offline-trained models, and policy impact can only be assessed after deployment. No standardized mechanism

exists for an rApp to simulate a candidate policy's outcome before committing it to the live network.

B. Digital Twins for Telecommunications

A digital twin is a continuously synchronized virtual replica of a physical system, enabling simulation, prediction, and optimization without affecting live operations [7]. A DT-RAN replicates physical infrastructure within a software-defined model that can be queried and reconfigured on demand [4].

Three properties distinguish a DT-RAN from conventional simulators. First, *state synchronization*: the twin ingests live telemetry rather than relying on static snapshots. Second, *multi-fidelity modeling*: physics-based models (e.g., 3GPP channel models) are combined with data-driven models (e.g., traffic forecasting) to balance cost and accuracy. Third, *bidirectional feedback*: twin-derived insights feed back into the management plane, closing the virtual-physical loop.

Standardization has accelerated considerably. 3GPP initiated a study on network digital twin in Release 18, with normative work progressing through Releases 19 and 20 [8]. The O-RAN nGRG published a DT-RAN key enablers report in October 2024 [5], and the Alliance plans to incorporate DT-RAN into its Release 005 roadmap. ITU-T has addressed related requirements under its Y.3090 recommendations [9].

On the research front, Nguyen et al. [4] proposed a hierarchical DT-RAN with progressive fidelity levels for O-RAN. Masaracchia et al. [10] explored DT-enabled intelligent closed-loop control. The AIDITA framework [11] introduced generative-AI-augmented incremental learning for traffic analytics within an O-RAN digital twin. The Colosseum platform [12] provides an open-source O-RAN digital twin testbed for large-scale experimentation.

While these contributions demonstrate growing DT-RAN maturity, most treat the digital twin as a standalone platform architecturally separate from the Non-RT RIC's programmable service framework. Notable exceptions exist: Kim et al. [13] proposed a development and validation framework for AI/ML-driven rApps that incorporates digital twin elements for network energy saving, and Ndikumana et al. [14] demonstrated a digital twin-backed closed-loop architecture for energy-aware Open RAN in fixed wireless access deployments. These works confirm that the convergence of DT-RAN and Non-RT RIC intelligence is gaining traction. However, neither exposes the digital twin as a native R1 service nor defines a structured, rApp-invocable what-if analysis workflow, i.e., the two gaps addressed by the present work.

C. Gap Analysis

Table I summarizes representative prior works along four dimensions: digital twin scope, O-RAN integration, what-if support, and limitations. Two structural gaps emerge.

First, *no existing work exposes DT-RAN as a native R1 service*. Current proposals interface the twin at the SMO level or through ad-hoc APIs, but none leverage R1 service discovery and invocation mechanisms, forcing bespoke integration logic that undermines O-RAN's modularity and vendor neutrality.

Second, *systematic what-if analysis is absent from the rApp workflow*. While several proposals mention scenario testing, none define a structured, programmatically invocable workflow, leaving the digital twin as a passive analytics tool rather than an active participant in the policy generation loop.

These gaps motivate the architecture presented in the next section, which embeds the DT-RAN Engine within the Non-RT RIC service framework and introduces a what-if analysis engine accessible to rApps through standardized interfaces.

III. Proposed Architecture: DT-Enhanced rApp Framework

A. High-Level Architecture

Fig. 1 illustrates the end-to-end architecture, guided by three principles: (i) modularity, allowing the DT engine to be deployed and scaled independently; (ii) standards alignment, mapping new interfaces to O-RAN WG2 functional blocks; and (iii) vendor neutrality, avoiding assumptions about specific simulation platforms or data formats.

The DT-RAN Engine is introduced as a new logical function within the SMO, accessible to rApps through R1. A closed-loop data path ingests Performance Management (PM) and Fault Management (FM) data via O1 and optionally near-real-time KPIs via E2 to synchronize the twin with the live network. rApps request simulations, consume twin-generated enrichment information, and translate validated outcomes into A1 policies for Near-RT RIC enforcement.

Crucially, DT-RAN functionality is exposed through the existing R1 service framework rather than a dedicated interface, allowing rApps to discover and invoke DT services via the same service management and exposure (SME) mechanisms used for other platform services, minimizing integration burden and preserving backward compatibility.

B. Digital Twin Engine Design

1) Data Synchronization Layer: This subsystem maintains twin accuracy through continuous data ingestion. Primary feeds include O1-sourced PM counters and FM alarms at 15-minute to 1-hour intervals. Supplementary feeds may include E2-sourced near-real-time KPIs and external contextual data (weather, events, population density). A stream-processing pipeline performs ingestion, anomaly filtering, and schema normalization before writing to the twin's state store. The E2* path shown in Fig. 1 denotes an optional supplementary feed: where the SMO has visibility into E2-sourced near-real-time KPIs (e.g., via the Near-RT RIC's data exposure capabilities), these can be forwarded to the DT-RAN Engine's data synchronization layer to improve twin fidelity beyond what O1 PM intervals provide.

2) Model Repository: Three model categories are supported. *Physics-based models* capture deterministic phenomena such as propagation, antenna patterns, and interference geometry. *Data-driven models* learn statistical relationships from historical data via neural networks or ensemble methods. *Hybrid models* combine physics-based priors with data-driven corrections, e.g., ray-tracing models calibrated by field measurements. All models are versioned with metadata and exposed to rApps through a catalog API.

TABLE I
COMPARISON OF REPRESENTATIVE PRIOR WORKS ON DT-RAN AND O-RAN INTEGRATION.

Reference	DT Scope	O-RAN Integration	What-If Support	Key Limitation
Nguyen et al. [4]	Cell/network-level	SMO-level (conceptual)	Mentioned, not formalized	No R1 interface mapping
AIDITA [11]	Analytics models	RIC-level (xApp/rApp)	Not supported	Focused on model adaptation, not policy
Colosseum [12]	Full-stack emulation	E2/A1 (testbed)	Supported (manual)	Research testbed, not production architecture
Masaracchia et al. [10]	Network-level	Conceptual	Mentioned	No interface specification
nGRG Report [5]	RAN-wide (enablers)	Architecture-level	Identified as use case	No implementation or workflow defined
This work	**Configurable**	**R1 native service**	**Formal 4-stage engine**	**—**

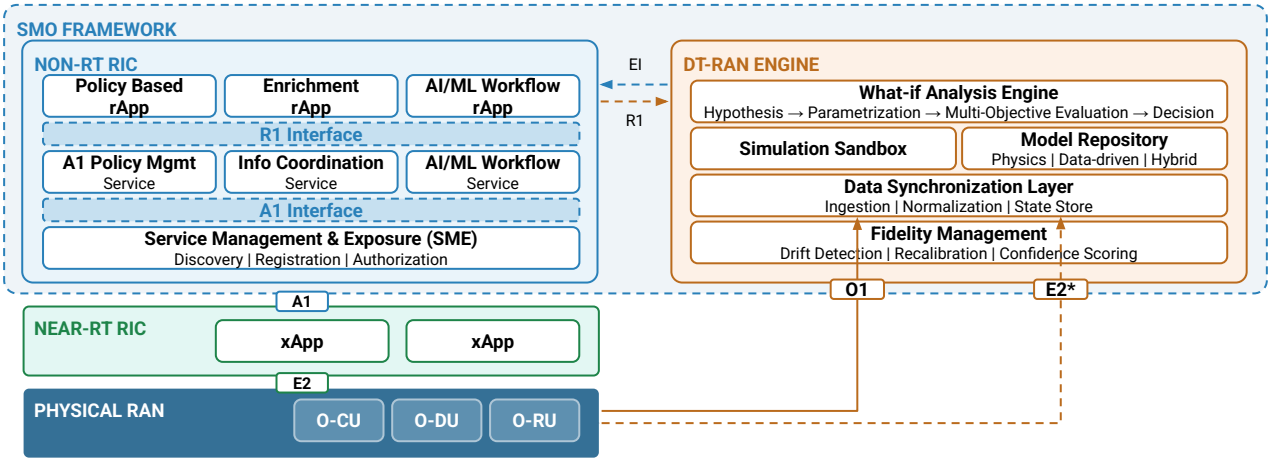


Fig. 1. End-to-end DT-enhanced rApp architecture.

3) Simulation Sandbox: The sandbox provides isolated environments where what-if scenarios are evaluated without risk to the live network. Each instance is initialized from the current twin state and subjected to rApp-defined perturbations. Concurrent execution is supported via containerized workloads. Granularity is configurable: statistical models yield cell-level KPIs in seconds, while full protocol-stack emulations provide per-UE estimates in minutes.

4) Fidelity Management: The trustworthiness of what-if analysis depends on how accurately the digital twin reflects the live network. The fidelity management module quantifies and maintains this accuracy through a three-step closed loop.

Step 1: Continuous monitoring. For each cell c and KPI type k (e.g., downlink throughput, PRB utilization), the module periodically compares the twin's predicted value $\hat{y}_{c,k}$ against the corresponding O1-observed value $y_{c,k}$ at every PM reporting interval. Divergence is captured by per-KPI drift metrics: normalized root-mean-square error (NRMSE) for continuous KPIs and Kullback–Leibler divergence for distributional quantities such as traffic load profiles.

Step 2: Threshold-triggered recalibration. Each drift metric is compared against a configurable threshold τ_k. When $\text{NRMSE}_k > \tau_k$ for a sustained window (e.g., three consecutive reporting intervals, to avoid reacting to transient spikes), the module identifies the degraded model component and triggers targeted retraining or fine-tuning using the most recent data window. Only the affected component is recalibrated, avoiding unnecessary disruption to well-performing models.

Step 3: Confidence scoring. After each monitoring cycle, the module computes a per-cell, per-KPI fidelity score $f_{c,k} \in [0, 1]$ reflecting the twin's current predictive reliability for that network segment. These scores are attached to every what-if result returned to rApps, enabling risk-aware decisions: an rApp may accept a policy recommendation backed by high-fidelity predictions ($f > 0.9$) but request a higher-fidelity simulation or defer the decision when scores fall below an application-specific threshold.

C. rApp Integration Interfaces

The integration between rApps and the DT-RAN Engine is realized through three interfaces.

1) rApp → DT-RAN (Simulation Request Interface): rApps submit structured scenario descriptors specifying simulation type, target scope, parameter perturbations, and KPIs to evaluate. The DT-RAN Engine instantiates a sandbox, executes the simulation, and returns predicted KPI values with confidence intervals and a fidelity score. This interface is exposed as an R1 service following the RESTful pattern established for other Non-RT RIC services.

2) DT-RAN → rApp (Enrichment Interface): The DT-RAN Engine can also operate as an enrichment information (EI) producer, periodically publishing predictive analytics as EI types that rApps subscribe to. This push-based model delivers twin-derived intelligence without explicit simulation requests, supporting continuous optimization workflows.

3) rApp → Near-RT RIC (A1 Policy Loop): Once a candidate policy is validated through what-if analysis, the rApp issues it to the Near-RT RIC via the standard A1 interface, no protocol modifications are required. An optional feedback path routes post-deployment KPI observations back to the DT-RAN Engine for calibration, closing the outer optimization loop.

D. What-If Analysis Engine

The what-if analysis engine enables rApps to leverage the DT for proactive decision-making through four stages.

Stage 1: Hypothesis Definition. The rApp formulates a scenario descriptor specifying the target scope, parameter space, constraints (e.g., minimum coverage), and objective function (e.g., minimize energy subject to a QoE floor).

Stage 2: Scenario Parameterization and Execution. The DT-RAN Engine initializes a sandbox from the current twin state, applies the specified perturbations, and runs the simulation. Parameter sweeps are supported through parallel sandbox instances. Outputs include per-cell and, where supported, per-UE KPI predictions.

Stage 3: Multi-Objective Evaluation. An evaluation module scores each configuration against the rApp's objectives using configurable strategies (weighted-sum scalarization, Pareto front extraction, or constraint-satisfaction filtering), returning ranked candidates with predicted KPI trade-offs.

Stage 4: Decision and Policy Commitment. The rApp decides whether to deploy, refine, or discard the policy. Upon deployment, the corresponding A1 policy is issued to the Near-RT RIC. The engine logs predictions to enable post-deployment validation, feeding the fidelity management loop described previously.

This workflow is deliberately generic, imposing no assumptions on the optimization type, fidelity level, or policy format, and thus serves as a reusable substrate for a wide range of use cases, two of which are detailed next.

IV. Use Cases and Validation Scenarios

This section demonstrates the proposed framework through two use cases. Table II provides a comparative summary.

A. Use Case 1: Proactive Energy Saving

Operators face mounting pressure to reduce energy consumption without compromising service quality. Cell sleep/wake strategies are well studied, yet reactive deployment risks coverage gaps during unexpected traffic surges. An energy-saving rApp leverages the DT-RAN Engine to evaluate candidate sleep schedules before deployment. The scenario descriptor specifies candidate cells, deactivation windows, and a minimum RSRP coverage constraint. The what-if engine initializes a sandbox from current traffic and RF conditions, applies the perturbations, and scores each schedule against three KPIs: energy savings (%), coverage degradation (%), and 5th-percentile user throughput. Configurations violating the constraint are discarded, and the top-ranked schedule is issued as an A1 policy to the Near-RT RIC for enforcement via power management xApps. The key advantage is that dozens of schedule variants can be tested in the digital twin in minutes, rather than being deployed sequentially into the live network over days or weeks.

B. Use Case 2: Failure Impact Assessment and Self-Healing

Cell outages and equipment degradation are inevitable in large-scale deployments. Traditional SON mechanisms apply predefined compensation actions after failure detection, without validating their effectiveness for the specific context. A failure assessment rApp pre-computes healing strategies using the DT-RAN Engine. The rApp defines a library of failure scenarios parameterized by location and severity. For each, the what-if engine injects the failure, simulates coverage holes and traffic redistribution, and evaluates candidate actions (power boost, tilt adjustment, rerouting) against coverage hole area (%), handover success rate, and recovery time. Actions that would overload neighbors are discarded, and validated strategies are stored in a lookup table indexed by failure type and location. When a real failure is detected via O1 alarms, the rApp retrieves the pre-computed strategy and immediately issues an A1 policy, dramatically reducing mitigation time. The twin can be re-invoked in real time if actual conditions differ from the pre-computed scenario.

V. Preliminary Evaluation and Discussion

This section evaluates the proposed framework across the two use cases, compares DT-driven proactive management against reactive baselines, and discusses limitations.

A. Evaluation Setup

We use ns-3 with the 5G-LENA module to model a 3GPP-compliant urban macro deployment: 19 tri-sector sites (57 cells), 500 m inter-site distance, with 4 small cells per macro sector in high-traffic zones. Propagation follows 3GPP TR 38.901 UMa with 3D antenna patterns (65° horizontal HPBW, 15 dBi gain, 10° electrical downtilt). The carrier frequency is 3.5 GHz with 100 MHz bandwidth (273 PRBs, 30 kHz SCS). Aggregate demand follows a diurnal pattern modeled as a mixture of two Gaussians centered at 13:00 ($\sigma = 4.5$ h) and 20:00 ($\sigma = 2.8$ h) with a floor at 15% of peak load, consistent with public mobile traffic traces.

The DT-RAN Engine is instantiated as a calibrated replica ingesting PM counters every 15 minutes. The hybrid model combines 3GPP TR 38.901 ray-tracing propagation (calibrated path-loss exponents per site from the first 7 days of simulation) with a gradient-boosted traffic predictor (XGBoost, 200 estimators, max depth 6, learning rate 0.05) trained on 4 weeks of historical per-cell throughput and PRB utilization traces. Feature inputs include hour-of-day, day-of-week, cell ID, and a 4-interval rolling average of neighbor-cell load. Initial calibration achieves an NRMSE of 6.2% on per-cell downlink throughput, computed over a validation week.

Two baselines are compared: (i) a *reactive* baseline applying rule-based policies without pre-validation, cell sleep is triggered when average PRB utilization falls below 20% for three consecutive 15-minute intervals and revoked when utilization exceeds 30%, and (ii) a *static-optimal* baseline using fixed policies derived from historical averages, the sleep schedule is computed offline as the utilization-weighted minimum-energy configuration over the 4-week training period and remains unchanged during evaluation. All use cases are evaluated over a 7-day window, with the first 24 hours excluded as warm-up.

TABLE II
COMPARATIVE SUMMARY OF VALIDATION USE CASES.

Dimension	UC1: Energy Saving	UC2: Failure Assessment
rApp Type	Policy-based	Policy-based + pre-computation
DT Models Used	Propagation, traffic prediction	Propagation, mobility, capacity
What-If Parameters	Cell subset, schedule, power	Failure type, location, healing actions
Evaluation Strategy	Constraint filtering + ranking	Constraint-satisfaction filtering
Primary KPIs	Energy savings, coverage, throughput	Coverage hole, handover success, recovery time
Timescale	Minutes (offline sweep)	Pre-computed + real-time refinement

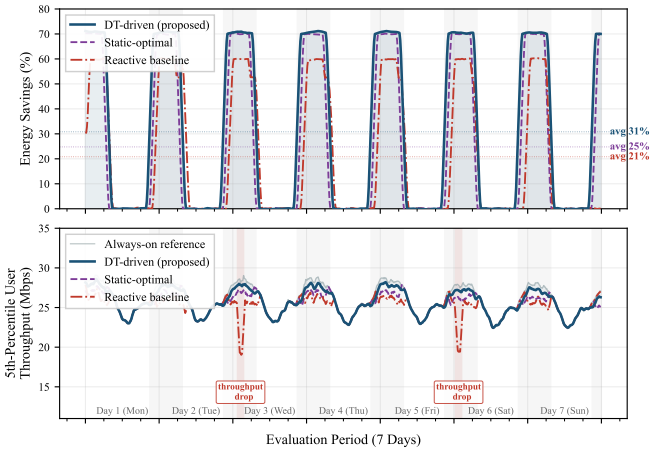


Fig. 2. Energy savings and 5th-percentile throughput over a 7-day evaluation window.

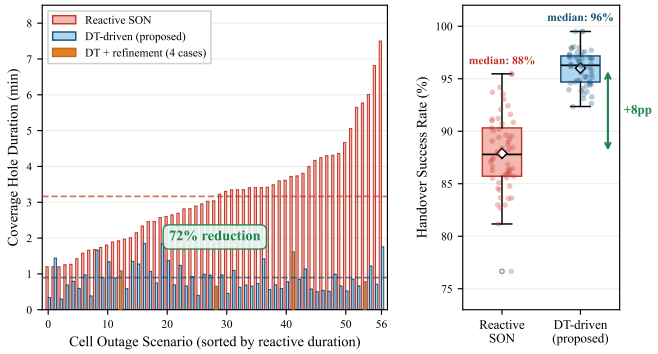


Fig. 3. Coverage hole duration and handover success rate under cell outage.

B. Results

1) UC1: Proactive Energy Saving: The DT-driven rApp evaluates 48 candidate sleep schedules, varying cell subset (3–8 small cells), deactivation window (midnight–5AM, 30-minute granularity), and power reduction level, completing the full sweep in under 4 minutes. Fig. 2 compares the three approaches over 7 days. The DT-driven approach achieves 31% energy savings versus 26% for static-optimal and 22% for reactive, while maintaining throughput within 3% of the always-on reference. The reactive baseline incurs a 9% throughput drop during two overnight periods when unexpected venue traffic exceeded its trigger threshold. The static-optimal policy avoids such incidents but cannot adapt to day-to-day variability.

2) UC2: Failure Impact Assessment: The failure assessment rApp pre-computes healing strategies for 57 single-cell outage and 12 partial degradation scenarios in approximately 35 minutes. During live simulation, the DT-driven rApp retrieves and applies strategies within 8 seconds of detection, versus 45 seconds for the reactive SON baseline. Fig. 3 shows the DT-driven approach reduces average coverage hole duration by 72% (from 3.2 to 0.9 minutes) and improves handover success from 88% to 96%. In 4 of 57 scenarios, traffic conditions deviated from pre-computed assumptions, triggering real-time refinement that added 12 seconds but still outperformed the reactive baseline.

C. Discussion

1) Twin Fidelity vs. Computational Cost: The hybrid modeling approach (ray-tracing plus data-driven traffic prediction) achieves an NRMSE of 6.2% at a computational cost of approximately 5 seconds per single-scenario simulation on a 16-core server. Replacing ray-tracing with a fully data-driven propagation model reduces this to 1.8 seconds but increases NRMSE to 11.4%, which degrades the energy-saving KPI predictions by approximately 4 percentage points.

2) Scalability Analysis: The current evaluation covers 57 cells. We analyze how the three computational bottlenecks scale with network size N.**Sandbox initialization** loads topology, configuration, and traffic state into an isolated environment, scaling as $\mathcal{O}(N)$ in memory and time. For localized what-if scenarios targeting a cell cluster, partial initialization reduces this to $\mathcal{O}(N_{\text{sub}})$ where $N_{\text{sub}} \ll N$. **Simulation execution** depends on fidelity. Statistical models scale as $\mathcal{O}(N \log N)$ due to interference graph computation, while full protocol-stack emulation scales as $\mathcal{O}(N^2)$ from pairwise interference, becoming impractical beyond a few hundred cells. Parallel sweeps over K configurations across P cores scale as $\mathcal{O}(K/P)$ since sandbox instances are independent. **Data synchronization** ingests $\mathcal{O}(N \cdot M)$ PM counters per reporting interval, where M is the number of counter types per cell. This remains lightweight at moderate scale but requires stream-processing infrastructure at $\mathcal{O}(10^4)$ cells. Two strategies mitigate these costs. *Hierarchical twinning* partitions the network into clusters of 50–100 cells managed by independent sub-twins, bounding per-simulation complexity to the cluster size. *Adaptive resolution* applies statistical models for coarse screening of large parameter sweeps, reserving full-fidelity simulation for the top-k Pareto candidates, reducing total compute by an estimated order of magnitude.

3) Data Freshness: The 15-minute PM ingestion interval introduces a staleness window during which the twin state may diverge from reality. For the energy-saving use case, where decisions operate on hourly timescales, this staleness is acceptable. For the failure assessment use case, the 15-minute window can lead to suboptimal pre-computed strategies if traffic patterns shift rapidly. Supplementing O1 PM data with E2-sourced near-real-time KPIs narrows this gap but increases the data synchronization overhead.

4) Control-Loop Latency Sensitivity: The evaluation assumes instantaneous A1 policy enforcement. In practice, the end-to-end loop comprises simulation time t_{sim}, R1 invocation overhead t_{R1}, A1 delivery latency t_{A1}, and xApp enforcement delay t_{enf}. The total $t_{\text{loop}} = t_{\text{sim}} + t_{\text{R1}} + t_{\text{A1}} + t_{\text{enf}}$ introduces a staleness window during which network conditions may drift from the twin snapshot used for what-if analysis. The impact is use-case-dependent. For energy saving (UC1), loop latencies of seconds to minutes are negligible relative to multi-hour decision timescales. Failure healing (UC2) is most sensitive: the 8-second response time would increase by $t_{\text{A1}} + t_{\text{enf}}$ in practice, but retains a significant margin over the 45-second reactive baseline. Two mechanisms mitigate staleness. First, the fidelity management module can attach a *validity horizon* to each result, estimating the window over which predictions remain reliable; the rApp discards stale results and re-invokes the engine if enforcement is delayed beyond this horizon. Second, pre-computation (as in UC2) shifts t_{sim} offline, reducing t_{loop} to $t_{\text{R1}} + t_{\text{A1}} + t_{\text{enf}}$ at enforcement time.

VI. Open Challenges and Future Directions

A. Toward Real-World Deployment

The evaluation presented in this work is entirely simulation-based, which validates the architectural concepts and quantifies expected gains but does not capture the full complexity of production O-RAN environments. Transitioning to real-world validation would involve several concrete steps: (i) deploying the DT-RAN Engine as a containerized microservice within the OSC Non-RT RIC platform; (ii) implementing the data synchronization layer against a live O1 PM/FM feed from commercial O-CU/O-DU/O-RU equipment; (iii) calibrating the digital twin against real RF measurements; and (iv) validating the what-if engine's predictions against actual post-deployment KPIs in a controlled testbed before scaling to operational networks. The OpenRAN@Brasil testbed [15], which spans commercial 5G Open RAN equipment across sites separated by over 400 km, exemplifies the type of large-scale distributed environment where DT-RAN fidelity and data synchronization assumptions would be most heavily stressed and where validation would therefore be most valuable. Integrating the proposed framework into such testbeds, or similar facilities like Colosseum [12] and AERPAW, constitutes a priority for future work.

B. Implementation Challenges for SMO/Non-RT RIC Integration

While the architecture targets standards alignment, integrating the DT-RAN Engine into production platforms raises practical challenges.

R1 Service Registration and SME Extensions. The current SME specification does not anticipate computationally intensive, long-running services such as digital twin simulations. Registering the DT-RAN Engine as an R1 service requires extending the service catalog with simulation-specific metadata (execution time, fidelity level, resource requirements) and augmenting the synchronous request-response pattern with asynchronous callbacks for multi-minute what-if sweeps.

Vendor Interoperability and Data Model Harmonization. In multi-vendor deployments, O1 PM/FM data arrives from heterogeneous sources with vendor-specific counter naming, inconsistent reporting granularity, and missing optional fields. While schema normalization aligned with 3GPP TS 28.552 handles common cases, per-vendor adaptation logic remains unavoidable. More broadly, no unified data model spans O1 topology, E2 real-time KPIs, and external contextual feeds at the fidelity required for simulation. Adopting the 3GPP Network Resource Model (NRM) as the canonical representation, extended with twin-specific metadata and aligned with ITU-T Y.3090, offers a path toward standardization.

Computational Overhead. Parallel sandbox execution during what-if sweeps imposes significant CPU and memory load on shared SMO infrastructure. Production deployments require resource isolation (i.e., dedicated container quotas, scheduling priorities, or offloading to edge/cloud GPU infrastructure) to prevent DT-RAN operations from degrading latency-sensitive platform functions such as A1 policy management.

C. AI/ML Trustworthiness and Governance

The DT-RAN Engine relies on data-driven models yet treats them largely as black boxes. Three aspects require attention. First, the fidelity management module detects drift via aggregate metrics but does not isolate degradation in individual model components (propagation, traffic, mobility); exposing per-model confidence scores alongside what-if results would let rApps assess which predictions carry elevated uncertainty. Second, the multi-objective evaluator indicates *which* KPIs improve but not *why*, integrating post-hoc explainability would improve operator trust and regulatory compliance. Third, emerging governance requirements (O-RAN nGRG responsible AI principles, EU AI Act) call for audit logging of model versions, training data, and what-if decisions, as well as a human-in-the-loop approval path for high-risk scenarios such as URLLC slice reconfiguration.

D. DT-RAN as an rApp Validation Environment

The proposed framework naturally extends to pre-deployment validation of new rApps. Onboarding third-party rApps into a production Non-RT RIC carries inherent risks, misconfigured policies, undertrained models, or unforeseen interactions with concurrent rApps can degrade performance or trigger cascading failures. Static conformance testing against WG2 specifications cannot capture the full complexity of a live multi-cell, multi-slice environment.

The DT-RAN Engine can serve as a staging environment where candidate rApps are exercised against a twin of the production network. The rApp is deployed within the

Non-RT RIC with its A1 output redirected to the sandbox rather than the live Near-RT RIC. The what-if engine then evaluates the rApp's policies across representative scenarios (nominal load, peak traffic, failure events, edge cases) and checks them against operator-defined safety invariants, e.g., no action should reduce throughput beyond a threshold or violate URLLC latency bounds. Violations are flagged with the triggering scenario and affected KPIs.

This addresses two concerns beyond static testing. First, *behavioral safety*: the twin reveals harmful outcomes under conditions the rApp may not have encountered during development. Second, *multi-rApp conflict detection*: operators can test new rApps alongside the existing portfolio, identifying policy conflicts, such as an energy-saving rApp deactivating cells that a load-balancing rApp relies on, before they affect the live network. This capability aligns with the O-RAN Alliance's ongoing rApp certification efforts and complements conflict detection mechanisms under development within WG2.

VII. Conclusion

This article proposed a unified architecture that embeds digital twin capabilities as a native service within the O-RAN Non-RT RIC rApp ecosystem. A four-stage what-if analysis engine was introduced, supporting hypothesis definition, parameterized execution with parallel sandboxes, multi-objective evaluation, and policy commitment with post-deployment fidelity tracking. Two use cases demonstrated the framework's generality: proactive energy saving achieved 31% energy reduction with minimal throughput impact and pre-computed failure healing strategies reduced coverage hole duration by 72% while improving handover success from 88% to 96%.

Key open challenges remain in standardizing DT-RAN interfaces within O-RAN WG2, scaling DT simulations to large-scale deployments, and ensuring AI/ML model trustworthiness within the twin-rApp loop. As DT-RAN matures through 3GPP Release 19/20 and O-RAN Release 005, the convergence of DT and rApps has the potential to transform RAN management from a reactive discipline into a proactive, simulation-driven practice, a capability that will be essential for the complexity of 6G networks.

References

[1] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[2] O-RAN Alliance, "Non-RT RIC: Architecture," O-RAN Alliance, Tech. Rep. O-RAN.WG2.TS.Non-RT-RIC-ARCH-R004-v07.00, 2025.

[3] ——, "Non-RT RIC & A1/R1 interface: Use Cases and Requirements," O-RAN Alliance, Tech. Rep. O-RAN.WG2.TS.Use-Case-Requirements-R005-v11.00, 2026.

[4] H. X. Nguyen, K. Sun, D. To, Q.-T. Vien, and T. A. Le, "Digital twin for o-ran toward 6g," *IEEE Communications Magazine*, vol. 63, no. 3, pp. 174–181, 2025.

[5] O-RAN Alliance nGRG, "Digital twin RAN: Key enablers," O-RAN Alliance, next Generation Research Group, Tech. Rep. nGRG-RS01-RS03, 2024.

[6] O-RAN Alliance, "R1 interface: Test Specification," O-RAN Alliance, Tech. Rep. O-RAN.WG2.TS.R1TS-R005-v05.00, 2026.

[7] A. Masaracchia, V. Sharma, B. Canberk, O. A. Dobre, and T. Q. Duong, "Digital twin for 6g: Taxonomy, research challenges, and the road ahead," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 2137–2150, 2022.

[8] 3GPP, "Study on management aspects of network digital twin phase 2," 3rd Generation Partnership Project, Tech. Rep. TR 28.883, 2025, release 20.

[9] ITU-T, "Digital twin network — requirements and architecture," International Telecommunication Union, Tech. Rep. Recommendation Y.3090, 2022.

[10] A. Masaracchia, V. Sharma, M. Fahim, O. A. Dobre, and T. Q. Duong, "Digital twin for open ran: Toward intelligent and resilient 6g radio access networks," *IEEE Communications Magazine*, vol. 61, no. 11, pp. 112–118, 2023.

[11] H. Navidan, C. Martín, V. Maglogiannis, D. Naudts, M. Díaz, I. Moerman, and A. Shahid, "An end-to-end digital twin framework for dynamic traffic analytics in o-ran," *IEEE Transactions on Network and Service Management*, vol. 23, pp. 76–92, 2026.

[12] M. Polese, L. Bonati, S. D'Oro, P. Johari, D. Villa, S. Velumani, R. Gangula, M. Tsampazi, C. Paul Robinson, G. Gemmi, A. Lacava, S. Maxenti, H. Cheng, and T. Melodia, "Colosseum: The open ran digital twin," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 5452–5466, 2024.

[13] M. Kim, K. S. Lee, S. Jung, J. M. Moon, J.-H. Na, S. D'Oro, L. Bonati, and T. Melodia, "A development and validation framework for AI/ML-driven rApps in open RAN: A case study on network energy saving," *EURASIP Journal on Wireless Communications and Networking*, vol. 2026, no. 56, 2026.

[14] A. Ndikumana, K. K. Nguyen, and M. Cheriet, "Digital twin backed closed-loops for energy-aware and open ran-based fixed wireless access serving rural areas," *IEEE Transactions on Mobile Computing*, vol. 24, no. 3, pp. 1669–1683, 2025.

[15] W. Mauricio, F. H. Costa Neto, M. Silva, R. K. Y. Aoki, E. Melão, S. Barros, F. Farias, L. Bondan, and F. M. Abinader, "Assessing nationwide distributed open ran deployments with the openran@brasil testbed," *IEEE Communications Magazine*, vol. 63, no. 2, pp. 46–52, 2025.

Roberto Riggio is an Associate Professor at 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.