Hashem Eiza, M, Akwirry, B, Raschella, A, Mackay, M and Maheshwari, M

A Hybrid Zero Trust Deployment Model for Securing O-RAN Architecture in 6G Networks

future internet

MDPI

Article

A Hybrid Zero Trust Deployment Model for Securing O-RAN Architecture in 6G Networks

Max Hashem Eiza [1,*], Brian Akwirry [2], Alessandro Raschella [1], Michael Mackay [1] and Mukesh Kumar Maheshwari [1,3]

[1] School of Computer Science and Mathematics, Liverpool John Moores University, Liverpool L3 3AF, UK; a.raschella@ljmu.ac.uk (A.R.); m.i.mackay@ljmu.ac.uk (M.M.)
[2] School of Engineering and Computing, University of Lancashire, Preston PR1 2HE, UK; boakwirry@uclan.ac.uk
[3] Department of Electrical Engineering, Bahria University, Karachi Campus, Karachi 75260, Pakistan
* Correspondence: m.hashemeiza@ljmu.ac.uk

Abstract

The evolution toward sixth generation (6G) wireless networks promises higher performance, greater flexibility, and enhanced intelligence. However, it also introduces a substantially enlarged attack surface driven by open, disaggregated, and multi-vendor Open RAN (O-RAN) architectures that will be utilised in 6G networks. This paper addresses the urgent need for a practical Zero Trust (ZT) deployment model tailored to O-RAN specification. To do so, we introduce a novel hybrid ZT deployment model that establishes the trusted foundation for AI/ML-driven security in O-RAN, integrating macro-level enclave segmentation with micro-level application sandboxing for xApps/rApps. In our model, the Policy Decision Point (PDP) centrally manages dynamic policies, while distributed Policy Enforcement Points (PEPs) reside in logical enclaves, agents, and gateways to enable per-session, least-privilege access control across all O-RAN interfaces. We demonstrate feasibility via a Proof of Concept (PoC) implemented with Kubernetes and Istio and based on the NIST Policy Machine (PM). The PoC illustrates how pods can represent enclaves and sidecar proxies can embody combined agent/gateway functions. Performance discussion indicates that enclave-based deployment adds 1–10 ms of additional per-connection latency while CPU/memory overhead from running a sidecar proxy per enclave is approximately 5–10% extra utilisation, with each proxy consuming roughly 100–200 MB of RAM.

Keywords: 6G; Security; Next-G Networks; O-RAN; Open RAN; Zero Trust; NIST PM

check for updates

Academic Editor: Jose I. Moreno Novella

Received: 27 June 2025
Revised: 5 August 2025
Accepted: 6 August 2025
Published: 18 August 2025

Citation: Hashem Eiza, M.; Akwirry, B.; Raschella, A.; Mackay, M.; Maheshwari, M.K. A Hybrid Zero Trust Deployment Model for Securing O-RAN Architecture in 6G Networks. *Future Internet* **2025**, *17*, 372. https://doi.org/10.3390/fi17080372

1. Introduction

Research and development activities towards the sixth generation (6G) of wireless communications are gaining pace. The first commercial rollout of 6G networks is expected around 2030 while the standardisation process will start this year (2025) according to the 3GPP development roadmap [1]. 6G networks are envisaged to revolutionise the wireless networking ecosystem providing faster, more responsive, and increasingly ubiquitous connectivity than ever before. Moreover, the number of potential 6G applications (e.g., Internet of Things (IoT)/Industry 4.0, Augmented Reality (AR)/Virtual Reality (VR), vehicle-to-everything) across diverse industries and services introduces unprecedented challenges in meeting network requirements. These requirements will include wider coverage, higher data rates, ultra-low latency, ultra-high location accuracy, integration of communications and sensing, more intelligence, stronger security, and better sustainability in comparison with fifth generation (5G) networks.

To accommodate the diverse needs of 6G networks and its applications, new approaches to network architecture are required. This includes openness, disaggregation, and multi-vendor interoperability of the 6G Radio Access Network (RAN). In this context, Open RAN, aka O-RAN as referred to by the O-RAN Alliance [2], are expected to offer several benefits to 6G networks [3]. O-RAN allows for a greater flexibility in selecting and optimising network functions, components, and services. It can also contribute to a smoother transition to 6G by allowing multi-vendor network components to be reused to support 6G scenarios. However, this openness and disaggregation dramatically increases the attack surface and insider threat potential. There is a wide consensus that 6G networks will include O-RAN components and 3GPP as well as legacy deployments [4]. Therefore, there are a wide range of ongoing research activities within organisations such as O-RAN Alliance, 3GPP, and 6G Smart Networks and Services Industry Association (6G-IA) to identify key architecture principles relevant to the standardisation of a future 6G RAN. These principles are guiding the development of different aspects of future O-RAN standards [5].

Resilience, security, and privacy of 6G networks are among the top priorities identified by those architecture principles. The adoption of open architectures and interfaces, cloud deployments, and multi-vendors' networks will expand the attack surface especially to internal attacks due to Advanced Persistent Threats (APTs) and lateral movement. Hence, in 6G networks, it is critical to pursue a Zero Trust Architecture (ZTA) during the standardisation and design phases to achieve a strong security posture to protect against evolving threats [3]. ZTA is based on the Zero Trust (ZT) security paradigm where threats can originate from both external and internal sources [5]. Hence, ZT emphasises verification and validation of every access request to minimise the risk of security breach [6]. All assets and resources are secured as micro-perimeters and no entity, whether internal or external, is assumed to be trusted for access to applications and data regardless of its location and ownership. Therefore, the pursuit of ZT implementation, especially in critical infrastructure such as 5G/6G networks, is now a strategic goal set by governments, standardisation organisations, and industry (e.g., [7,8]). Examples of the most recent activities and reports for implementing ZT in O-RAN and 5G/6G networks are O-RAN Alliance Working Group (WG11) ZT for secure O-RAN white paper [9] and the National Institute of Standards and Technology (NIST) collaboration with O-RAN Alliance and Alliance for Telecommunications Industry Solutions (ATIS) to fully incorporate ZT into emerging standards for 5G and 6G wireless standards [10].

On the other hand, there are multiple guidance documents that define ZTA, its tenets, different deployment models, and maturity models for full implementation of ZT. Examples of these documents that are relevant to the mobile industry are NIST SP 800-207 ZTA [11], National Security Agency (NSA) and Cybersecurity and Infrastructure Security Agency (CISA) Enduring Security Framework (ESF) Security Guidance for 5G Cloud Infrastructures [12], and CISA ZT Maturity Model (ZTMM) [13]. All these documents explain how to incrementally strive towards a full implementation of ZT through different phases and across different pillars in critical infrastructure such as mobile networks. However, there is a lack of details regarding the deployment and implementation of different ZTA components such as Policy Decision Point (PDP) and Policy Enforcement Point (PEP).

Motivation & Contributions

Despite the growing consensus around ZT as a foundational principle for securing critical infrastructure, current efforts within the O-RAN domain (e.g., O-RAN Alliance WG11 and their next-Generation Research Group (nGRG) Security) remain conceptual, lacking concrete deployment models tailored to its layered and interface-heavy design. Most existing works in the literature either focus on securing specific O-RAN interfaces or propose high-level ZT frameworks without addressing practical enforcement mechanisms, dynamic policy management, or the complexities of microservice-based components such as xApps/rApps. This gap leaves operators and vendors with limited technical guidance on how to implement ZT within O-RAN in a scalable, enforceable, and standards-aligned manner. There is little to no discussion about how various security controls operate in practice and how they should be implemented to achieve a mature ZTA in O-RAN.

This paper addresses this gap by proposing and validating a novel hybrid ZT deployment model for securing O-RAN architecture based on O-RAN Alliance technical specification in [14], as to be explained later in Section 2.1. To the best of our knowledge, this is the first work to propose a ZT deployment model purpose-built for O-RAN with full coverage of all O-RAN interfaces and their security controls. The proposed model uniquely integrates:

1. Enclave-based segmentation to enforce macro-level isolation between logically grouped O-RAN functions,
2. Device application sandboxing to isolate xApps/rApps, thereby limiting intra-enclave lateral movement at the micro-level,
3. Distributed PEPs and a centralised PDP, aligned with NIST ZTA tenets and O-RAN Alliance specifications,
4. A Kubernetes and Istio-based Proof of Concept (PoC) implementation, leveraging the NIST Policy Machine (PM) to demonstrate technical feasibility,
5. A comprehensive mapping of ZT deployment approaches (i.e., agent/gateway-based, enclave, resource portal, and sandboxing) to their applicability in O-RAN contexts, and
6. A performance analysis based on the PoC implementation that considers latency, CPU/memory impact, scalability, and trade-offs of the proposed hybrid ZT model.

Together, these contributions lay a foundation for both research and industry adoption of ZT in O-RAN for the future 6G networks. Unlike AI-centric works, our hybrid ZT model provides the security fabric (i.e., enclaves, sandboxing, dynamic PDP/PEP) necessary to deploy and protect AI-driven xApps/rApps. Without ZTA's per-session verification, AI models themselves could become attack surfaces [15]. Note that prior AI/ML works focus on specific interface and/or component in O-RAN while our model is the first to secure all O-RAN interfaces with ZT principles, creating a trusted foundation for network-wide AI/ML deployment. Finally, when we make references to O-RAN security controls and requirements, the reader can refer to [14,16] for full details. Furthermore, for more details on ZT logical components including PDP, which is composed of the Policy Engine (PE) and the Policy Administrator (PA), and PEP, we refer the reader to NIST SP 800-207 publication [11].

The rest of the paper is organised as follows. In Section 2, the latest works related to securing O-RAN interfaces and ZTA are presented. Section 3 presents the hybrid ZT deployment model for O-RAN and explains the rationale behind it in detail. The strategy for implementing the proposed model, a PoC implementation, and its potential impact on O-RAN performance are explained in Section 4. Further analysis of the proposed deployment and implementation strategies, and discussion of trade-offs of the hybrid ZT deployment model are provided in Section 5. Finally, the paper's conclusions and future work are presented in Section 6.

2. Related Works

2.1. O-RAN Architecture

As mentioned before, the focus of this work is the logical O-RAN architecture illustrated in Figure 1 below including its components and interfaces as defined by O-RAN Alliance [14]. The Service Management and Orchestration (SMO) deals with network orchestration, automation, and optimisation, supporting the Non-Real-Time RIC (non-RT RIC). The non-RT RIC operates with latency above 1s, optimising the network through rApps, and interacts with the Near-Real-Time RIC (near-RT RIC) via the A1 interface for policy control and model updates. The near-RT RIC operates with latency between 10ms and 1s, and implements real-time optimisation through xApps, and controls the O-RAN Centralised Unit (O-CU) and O-RAN Distributed Unit (O-DU) via the E2 interface. The O-CU is composed of O-CU-CP (Control Plane) for signalling and O-CU-UP (User Plane) for user traffic. The O-DU is linked to the O-CU via the F1 interface, and to the Radio Unit (O-RU) via the Open Fronthaul (OpenFH) interface to support different functional split options. The O1 interface connects the SMO to all O-RAN elements for Fault, Configuration, Accounting, Performance, and Security (FCAPS) management. Finally, the O2 interface allows orchestration, management, and automation of cloud-based RAN (O-Cloud) components.

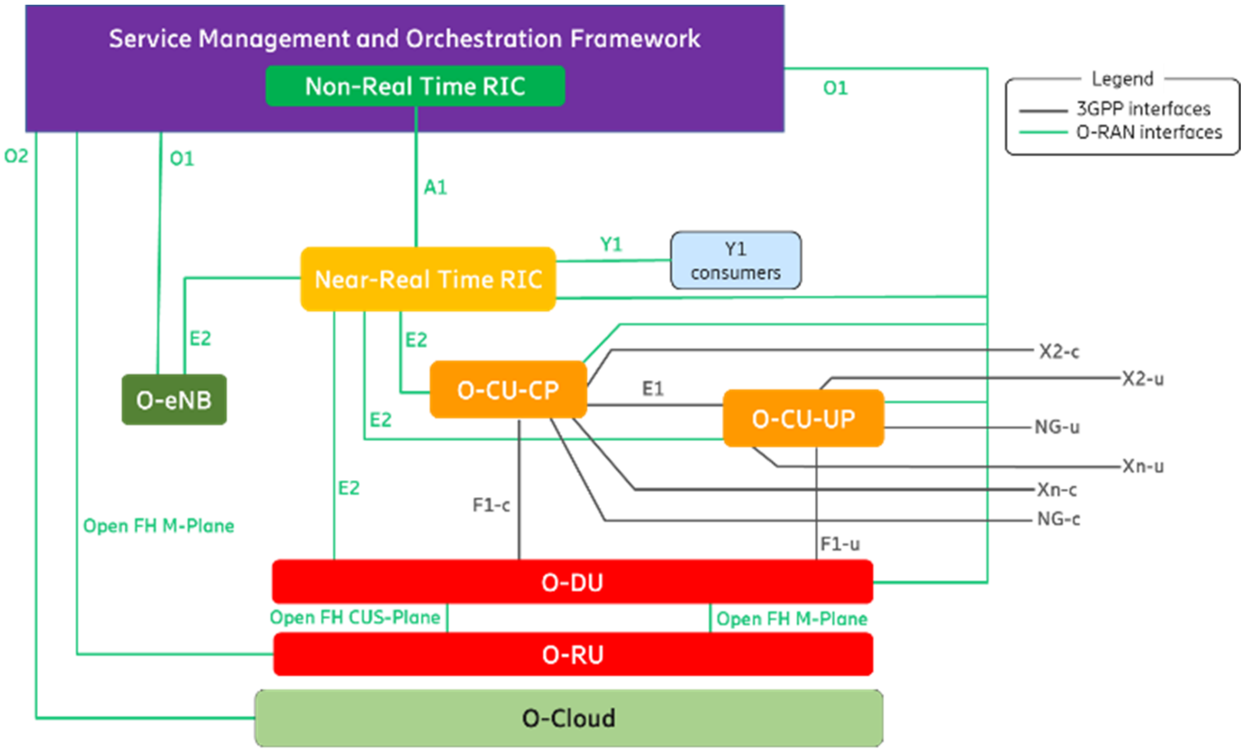

Figure 1. Logical Architecture of O-RAN [14].

2.2. O-RAN Security Challenges & Standards

In this section, we focus on the latest advancements in O-RAN security, particularly those aligned with ZTA principles. In Ref. [17], the authors analysed the security threats and challenges associated with O-RAN architecture and presented a comprehensive threat taxonomy of O-RAN components. They categorised risks into three types: process, technology, and global. The process risks are related to challenges to standardising critical processes in O-RAN where security controls can be applied to ensure privacy and integrity of assets. Technology risks mainly come from the use of open-source software and potential vulnerabilities in different components. Finally, global risks, in the broader sense, are related to securing telecom infrastructure against attacks and espionage from international/global attackers. The paper concludes by stressing the need for a comprehensive and robust security standards and governance to ensure the benefits of O-RAN are realised. In relation to ZTA, the authors touched briefly on how ZTA represents a simpler security model that can enhance the security of O-RAN. However, they did not explore the practical aspects of implementing ZTA in O-RAN nor provided practical security measures to advance O-RAN security towards ZTA.

In Ref. [15], Polese et al. have shown that AI-driven security is pivotal in O-RAN to address dynamic threats. AI could enable real-time anomaly detection, adaptive policy enforcement, and predictive threat mitigation, which are critical for ZT's continuous verification requirements in 6G RAN. This can be achieved via AI-driven anomaly detection engines that reside in O-RAN control loop to continuously analyse interactions across the A1 and E2 interfaces to identify deviations from baseline behaviour, enabling rapid isolation of suspect network elements, and thwarting lateral movement. AI can also aid in further adapting security policies in real time (e.g., tuning mutual TLS (mTLS) cipher suites, IPSec parameters or sandbox permissions) based on evolving threat patterns and contextual telemetry. This AI-based proactive, data-driven policy enforcement will accelerate threat detection and aligns naturally with ZT tenets of continuous verification.

Challenges and solutions to securing O-RAN interfaces were presented in Ref. [18]. However, the authors focused their work on two open interfaces only, the E2 and the Open FH. Both interfaces carry sensitive data that should be protected against threats such as data tampering and unauthorised access. Therefore, the paper investigated the impact of encryption in terms of latency and overhead on the performance of these two interfaces. More specifically, the authors presented an experimental analysis using IPSec for the E2 interface and Media Access Control Security (MACsec) for the Open FH interface. The results indicated that securing the E2 interface via IPSec encryption introduces minimal latency and performance overhead, making it a feasible security solution for this interface. However, when using MACsec on the Open FH interface, there was a significant impact that could affect the whole system performance especially in high traffic scenarios. In terms of ZTA, the authors mentioned it very briefly without any details of how it can be used to secure O-RAN.

In the same context, through its WG11, the O-RAN Alliance continues to advance O-RAN security via their work in three security specifications and a technical report that form the pillars of O-RAN security. These are (1) O-RAN Security Threat Modelling and Risk Assessment [19]; (2) O-RAN Security Requirements and Controls Specifications [20] (3) O-RAN Security Protocols Specifications [21]; and (4) O-RAN Security Tests Specifications [22]. The specified security requirements and security controls aim to mitigate risk from external and internal threats in pursuit of a ZTA, which is one of thirteen active security work items in WG11. For instance, WG11 has suggested that Service Management and Orchestration (SMO) functions must support authentication of both internal and external systems, as well as authorisation of service requests [23]. Mutual authentication is also required for SMO internal communications. On the other hand, the non-RT RIC should support authorisation, and for the transport layer security, mTLS should be used to provide API authentication and authorisation. JSON Web Token (JWT) is recommended for application layer authentication, and strict authorisation protocols for API access and discovery.

Finally, the latest white paper published by O-RAN Alliance WG11 confirmed that all NIST's seven tenets of ZTA are applicable to O-RAN [9]. These tenets are [11]:

1. All data sources and computing services are considered resources.
2. All communication is secured regardless of network location.

3. Access to individual enterprise/operator resources is granted on a per-session basis.
4. Access to resources is determined by dynamic policy.
5. The enterprise/operator monitors and measures the integrity and security posture of all owned and associated assets.
6. All resource authentication and authorisation are dynamic and strictly enforced before access is allowed.
7. The enterprise/operator collects information about the current state of assets, network infrastructure and communications and uses it to improve its security posture.

The white paper, however, did not provide technical details of security controls and/or deployments that are required to realise these tenets. It mentioned that an upcoming report, which has not been published yet, will address key techniques expected to play an important role toward achieving security goals in 6G use cases without any further details. While these O-RAN security specifications lay a strong foundation, they stop short of prescribing a concrete ZT deployment model. The next subsection reviews related works that apply ZT specifically to 6G networks or O-RAN.

2.3. Zero Trust in 6G & O-RAN

Abdalla et al. introduced zero trust RAN (ZTRAN) that includes service authentication, intrusion detection, and secure slicing subsystems that are encapsulated as xApps [24]. The authors highlighted the need for a ZT approach in O-RAN, emphasising continuous verification and authentication of users, devices, and applications. The authentication xApp implements multi-factor authentication (MFA) for securely identifying and verifying user equipment (UE) requesting access to O-RAN services. The intrusion detection xApp monitors the network activities continuously to detect suspicious behaviour. This is done by analysing Key Performance Metrics (KPM) and creating user behaviour profiles. Finally, the slicing xApp implements dynamic network slicing to isolate malicious users by assigning them to restricted instances to ensure legitimate users are not affected by any malicious activities. These three xApps were implemented and tested in the Open Artificial Intelligence Cellular (OAIC) platform to demonstrate its feasibility and effectiveness in terms of improving throughput for legitimate users, latency, and intrusion detection performance (i.e., isolating malicious users). However, some gaps still exist in the proposed ZTRAN framework such as scalability concerns (e.g., with many users and devices generating large-scale traffic) and resolution of xApps conflicts (i.e., where multiple microservices may interfere with each other especially when multiple xApps manage resources simultaneously).

In Ref. [25], the authors proposed the OZTrust framework, a ZT security system specifically designed for the O-RAN Near-RT RIC. OZTrust has two components: (1) an access control module that performs per-packet tagging and verification for each packet sent between xApps; and (2) a policy management module that automatically generates access control policies by tracing communication patterns between xApps through distributed tracing. Note that, in the O-RAN context, a unique tag represents the identity of an xApp derived from its authentic contexts. Therefore, OZTrust offers mitigations against compromised xApps and provides fine-grained xApp's access control policies without relying on static credentials or permissions. The OZTrust prototype is implemented based on eZTrust approach [26] and compared with Kubernetes Role-Based Access Control (RBAC) [27], and two Container Network Interfaces (CNIs) Cilium [28] and Calico [29]. OZTrust demonstrated that it can successfully block unauthorised API access and lateral movement attacks, where other systems failed. Besides its narrow focus on xApps security, OZTrust requires complex instrumentation processes to enable distributed tracing in an xApp. This involved inserting tracing APIs and their dependencies into the xApp's source code manually. This becomes challenging when the number of xApps increases and they come from different open-source platforms and different vendors, which might not necessarily collaborate among each other and with carriers.

Finally, Chen et al. [30] proposed a software-defined ZTA for 6G networks reinforcing the need to establish elastic and scalable security regime, and scalable access control. The authors proposed a distributed, community-based security framework in which each SDN-controlled community acts as its own ZT domain, enforcing access control at border switches and treating roaming devices like home/visited networks. Identity management is decentralised via per-community certificate authorities and a hierarchical identifier scheme stored on a private blockchain, preserving privacy while enabling cross-domain verification. Trust is evaluated at both the device and community levels combining self-reported device posture, third-party feedback, patch history, and anomaly scores. Nonetheless, their work does not address O-RAN's unique multi-vendor, interface-centric architecture. It also lacks both enclave-based segmentation and application sandboxing to contain lateral movement at the macro and micro levels. The validation is done via abstract trust-metric simulations rather than a concrete Kubernetes/Istio/NIST Policy Machine proof-of-concept.

2.4. Comparative Analysis of O-RAN Security Methodologies

Modern O-RAN security solutions span a range of techniques from static rule-based enforcement (e.g., traditional firewalls and Access Control Lists (ACLs)) through sophisticated AI/ML schemes. Table 1 provides different categories of mainstream security approaches for O-RAN and 6G including their advantages and disadvantages. While cryptographic methods provide strong guarantees, they struggle with dynamic threats. AI/ML solutions offer adaptability but lack explainability and require trusted data pipelines. Hardware-assisted techniques (e.g., Trusted Execution Environments (TEEs)) excel in low-latency scenarios but do not address network-level threats. Our hybrid ZTA integrates the strongest aspects: cryptographic controls for data-in-transit (IPSec/mTLS), hardware acceleration where feasible, fully adaptive for dynamic policies while avoiding blockchain's latency penalties.

Table 1. O-RAN Security Methodologies–Comparison.

Methodolgoy	Key Techniques	Advantages	Disadvantages
Cryptographic	IPSec, MACsec, mTLS	Provable security, Standardised	High compute overhead, Key management complexity
Signature Based	IDS/IPS that match network traffic against a database of known threat signatures	High accuracy for known attacks, Low false-positive rate on familiar patterns	Ineffective against zero-day exploits, Signature database must be constantly updated
AI/ML Based	Supervised or unsupervised models trained on normal operations to flag anomalies	Adapts to new threats, Real-time response	Training data vulnerability, May suffer false positives under traffic shifts
Hardware Assisted	TEEs, Trust zones	Tamper-proof execution, Low latency	Limited to supported platforms, Additional cost, Does not address network-level threats

Table 1. *Cont.*

Methodolgoy	Key Techniques	Advantages	Disadvantages
Blockchain Based	Distributed Ledger Technology (DLT), Decentralised identity	Immutable audit trails, No single Point of Failure, Transparent compliance	High latency, Energy intensive, Scalability concerns
Zero Trust Based (Proposed)	Dynamic PEP/PDP, Micro-segmentation	Least-privilege enforcement, full interface coverage, Fully adaptive	Policy management complexity, ~1–10ms latency overhead

Table 2 summarises related works' approaches, security methodology, limitations, and how our work differs.

Table 2. Comparison with Related Works.

Related Work	Approach	Security Methodology	Key Limitations	Our Work
Liyanage et al. [17]	• Threat taxonomy and high-level security challenges in O-RAN. • Brief mention of ZT as a conceptual model.	• Conceptual	• No concrete ZT deployment model tailored to O-RAN. • Lacks practical enforcement or interface level controls.	• Proposes a detailed hybrid ZT model covering all O-RAN interfaces. • Specifies PEP and PDP placement. • Develops a PoC with Kubernetes/Istio/NIST PM.
Groen et al. [18]	• Experimental analysis of IPSec (E2) and MACsec (OpenFH) performance overhead.	• Cryptographic	• Focuses only on two interfaces (E2, OpenFH). • Mentions ZT only briefly without design details or policy mapping.	• Covers every major O-RAN interface. • Integrates ZT tenets into one unified deployment strategy.
O-RAN WG11 (Threats, Security Requirements, and White Paper) [9,19–23]	• O-RAN security threat modelling, requirements, controls, protocols, and tests. • White paper confirms ZT tenets apply.	• AI/ML Based	• Standards/specs describe controls but do not prescribe deployment architecture. • No implementation guidance or PoC.	• Provides a reference architecture with detailed mapping to NIST ZTA tenets and real-world PoC using cloud-native tools.
Abdalla et al. (ZTRAN) [24]	• Zero Trust RAN (xApps for service authentication, intrusion detection, slicing).	• Zero Trust Based	• Narrow focus on xApps within Near-RT RIC. • Does not address macro-level enclaves or multi-interface enforcement.	• Introduces both macro (enclave) and micro (sandbox) isolation. • Covers all O-RAN logical groups (SMO, RICs, DU/CU, O-Cloud) and every interface.

Table 2. *Cont.*

Related Work	Approach	Security Methodology	Key Limitations	Our Work
Jiang et al. (OZTrust) [25]	• Fine-grained access control for xApps via per-packet tagging and distributed tracing.	• Signature Based	• Limited to xApps, requires manual instrumentation. • No enclave-level segmentation or policy engine discussion.	• Demonstrates a central policy engine (i.e., NIST PM) enforcing attribute-based rules across enclaves and sandboxing without manual code changes in xApps.
Chen et al. [26]	• Software-defined ZTA for 6G. • Distributed "communities" with border enforcement, Blockchain IDs, and trust metrics.	• Blockchain Based	• Abstract, non-O-RAN-specific. • No mapping to O-RAN interfaces or components.	• Tailor ZT deployment specifically to O-RAN's multi-vendor, interface-centric architecture. • Validates via Kubernetes/Istio PoC rather than abstract models.

3. The Hybrid Zero Trust Deployment Model for O-RAN Architecture

3.1. ZT Approaches & Potential Implementation in O-RAN

Within the ZT framework, there are several approaches that can be considered for implementation in O-RAN. Each approach implements all tenets of ZT, and a full ZT solution includes elements from all these approaches. In the following, we briefly explain each approach and how it can be applied to an O-RAN architecture.

3.1.1. Enhanced Identity Governance-Driven Approach

This approach focuses on managing and securing identities within an organisation. It ensures that only the right individuals have access to the right resources at the right times, and for the right reasons [31]. Authentication and authorisation are effective measures in disaggregated networks to ensure only authorised identities have access to resources. It mitigates the risk of unauthorised access and potential security breaches. Implementing identity-based controls capabilities could integrate additional capabilities such as compliance to regulatory requirements [32].

Implementation in O-RAN—the enhanced identity governance–driven approach in O-RAN can be utilised to enforce strict identity and access management across the disaggregated and virtualised components of the O-RAN architecture. This approach involves the use of advanced identity management systems to ensure that only authenticated and authorised entities (e.g., RIC administrators, vetted third-party xApp vendors, devices, and services) can access and interact with the O-RAN components. This can be achieved through mechanisms such as MFA, RBAC, access tokens issued by the centralised PDP, and continuous monitoring of identity usage patterns. This could mitigate risks of compromised Commercial-off-the-shelf (COTS) servers or rogue devices in virtualised RAN [9]. Furthermore, the automated identity management reduces administrative overhead and minimises the risk of human error associated with manual identity and access management processes.

3.1.2. Logical Micro-Segmentation

This approach involves dividing the network into smaller, isolated segments at a granular level where access and movement between segments are controlled by a policy [33]. By isolating different parts of the network, micro-segmentation limits the lateral movement of attackers, thus containing potential breaches and reducing the attack surface. This containment is particularly advantageous in a disaggregated and highly dynamic environment like O-RAN. The micro-segmentation also enhances the ability to detect and respond to threats more effectively by providing enhanced visibility in the network [34]. Finally, micro-segmentation enables the application of specific security policies tailored to the needs of each segment.

Implementation in O-RAN—logical micro-segmentation can be applied to O-RAN by dividing the network into smaller, isolated segments based on logical parameters such as function, user group, or data sensitivity. This involves creating virtual network segments within the O-RAN infrastructure, where each segment has its own set of security policies and controls. Each main O-RAN function (e.g., SMO, RIC) resides in its own segment, with strictly scoped east-west policies enforced by PEPs. For example, only the SMO orchestration service which possess an access token can send policy updates on A1 interface to the RIC while other SMO components are limited to read-only on O1 interface. Micro-segmentation can be implemented using software-defined networking (SDN) and network function virtualisation (NFV) technologies to dynamically enforce segmentation policies. This approach provides detailed visibility into network traffic and enables precise control over data flows. Therefore, enhancing the ability to detect and respond to threats more effectively.

3.1.3. Network-Based Segmentation

It involves dividing the network into distinct segments based on physical or virtual boundaries. It typically relies on traditional network infrastructure, such as firewalls and Virtual LANs (VLANs). This approach provides strong isolation between different network functions and segments, preventing unauthorised access and data breaches [35]. It also allows for centralised management and enforcement of security policies across different network segments, simplifying the overall security posture. This offers further flexibility and control in managing security policies across the entire network infrastructure.

Implementation in O-RAN—network-based segmentation can be used to divide the O-RAN infrastructure into distinct zones or segments, each with its own security controls and policies. This can be done using firewalls and VLANs to create secure boundaries between different network segments. Network-based segmentation is particularly useful for separating critical network functions, such as control plane and user plane functions, to enhance security and manageability.

3.2. ZT Deployment Variations for O-RAN

According to NIST SP 800-207 [11], there are four deployment variations/models for ZTA. In the following, we discuss each deployment model including its advantages, disadvantages, and applicability in an O-RAN architecture.

3.2.1. Device Agent/Gateway-Based Deployment

This model splits the PEP into two components: a device agent and a gateway. The device agent, installed on each enterprise-issued asset, handles connections and forwards access requests to the Policy Administrator (PA). The gateway component, located directly in front of a resource, acts as a proxy. This model is suitable for enterprises with a strong device management programme. Hence, it prevents Bring Your Own Device (BYOD) policies, allowing access only via enterprise-owned assets.

In O-RAN, this could translate to deploying device agents on network elements like RUs, DUs, and CUs, with gateways managing access and communication. This model ensures that only authorised devices can communicate. The agent installed on each network element manages the local network traffic, performs initial authentication, and forwards access requests to the PA. It also ensures that each device adheres to the security policies before initiating any communication. The gateway acts as a mediator between the network elements and enterprise resources and enforces security policies based on the decisions made by the PA. Moreover, it is responsible for maintaining secure communication channels and ensures that only authorised requests are processed. The advantages of this model are (a) ensuring that unauthorised devices cannot access critical network resources; (b) each communication request is thoroughly authenticated and authorised, reducing the risk of malicious activities; and (c) this model allows for granular control over network access, ensuring that only necessary and authorised communications are permitted.

However, the disadvantages are (a) this model requires a robust device management programme to handle the deployment and maintenance of device agents across all network elements. The need for continuous monitoring and updates to ensure that all agents and gateways are functioning correctly can add to the management overhead; (b) implementing this model may incur higher costs due to the need for specialised hardware (gateways) and the development and deployment of device agents; (c) the model does not support BYOD policies limiting its flexibility; (d) enterprises must ensure that all network elements and devices are enterprise-issued, which might not be feasible in all scenarios; and (e) it requires significant planning and testing to ensure seamless integration without disrupting existing services.

3.2.2. Enclave-Based Deployment

Unlike the device agent/gateway-based deployment model, this one places the gateway at the boundary of a resource enclave rather than individual resources. Resources within an enclave, such as data centre or cloud-based microservices, are protected by a single gateway. This makes it ideal for environments where resources serve a single business function or for legacy systems that cannot directly communicate with a gateway. This model still needs an agent installed on all devices that need access to resources.

In O-RAN, the network can be segmented into enclaves where each represents a different network segment (e.g., cloud, edge). Gateways at these enclaves' boundaries can manage/control access and enforce security policies for devices/resources within the enclave ensuring that only authorised requests are processed. Resources within an enclave are grouped based on their function or service type. Each enclave operates independently, with the gateway providing a secure access point.

The advantages of this model are (a) centralised security enforcement at the enclave boundary, simplifying the management of access policies; (b) reduces the complexity of deploying and maintaining individual gateways for each resource; (c) provides a protective barrier for legacy systems that cannot support direct communication with a gateway; (d) easily scalable by adding new enclaves and gateways as the network grows; and (e) cost effective as it reduces the need for deploying multiple gateways, lowering the overall cost of implementation. On the other hand, the disadvantages are (a) the gateway at the enclave boundary can become a single point of failure; (b) the gateway requires robust redundancy and failover mechanisms to ensure continuous availability; and (c) offers less granular control compared to the device agent/gateway model, as the gateway enforces policies at the enclave level rather than on individual resources.

3.2.3. Resource Portal-Based Deployment

A resource portal (i.e., gateway) acts as the PEP, in front of the resource(s), integrating with the enterprise's authentication and authorisation systems. Hence, the PEP is a single component in this model. Users can access resources through the portal, which enforces access policies. In O-RAN, the SMO can act as a portal, providing an interface for managing and accessing various O-RAN resources, ensuring compliance with security policies. The SMO serves as a central access point for users, consolidating access to multiple resources. It can integrate with enterprise authentication systems (e.g., Single Sign-On, Identity Providers) to authenticate users. Thus, the SMO provides a unified access interface for users to access diverse resources, simplifying the user experience.

The advantages of this model are (a) there is no need to install a software component on users' devices; (b) centralised authentication and authorisation processes, simplifying management and ensuring uniform policy enforcement; and (c) easily scalable by adding new resources behind the portal. However, the disadvantages are (a) the portal represents a single point of failure which can be susceptible to DoS attacks; (b) the portal requires robust provisioning to ensure high availability; (c) as all access requests pass through the portal, it can become a performance bottleneck, especially under high load conditions; and (d) devices can only be scanned for compliance when they request access making it difficult to continuously monitor them for malware or unpatched vulnerabilities.

3.2.4. Device Application Sandboxing

This deployment involves running applications within a sandboxed environment on devices. It ensures that applications adhere to enterprise security policies by restricting their access to system resources and data via the PEP. Thus, it is effective for managing and securing applications on BYOD devices or other environments where device control is limited. In O-RAN, sandboxing can be used to isolate xApps/rApps to ensure they do not interfere with each other and adhere to security policies. Each sandbox operates independently, ensuring that applications do not access unauthorised resources. Enterprise security policies are enforced within the sandbox, controlling what system resources and data the application can access.

The advantages of this model are (a) ensuring that applications are contained within a secure environment segmented from the rest of the asset; (b) isolating applications from each other, preventing conflicts, and ensuring that one application's malfunction does not affect others. This is particularly important for O-RAN as it has multiple critical applications running simultaneously; and (c) allowing the enterprise to enforce security policies on BYOD devices without needing full control over the device. However, the disadvantages are (a) running applications in sandboxes can introduce performance overhead due to the additional layer of isolation and control; and (b) managing and maintaining sandbox environments for multiple applications can be complex and resource intensive especially when full visibility into client assets is not available for the enterprise.

3.3. The Hybrid ZT Deployment for O-RAN

Considering the O-RAN architecture and the advantages/disadvantages of different ZT deployment models described above, we propose a hybrid ZT deployment model that integrates an enclave-based approach with device application sandboxing. This hybrid model gives us the ability to segment major functional groups via enclave borders, and isolate multi-vendor xApps/rApps via sandboxing. Thus, our hybrid model provides both macro-level (inter-enclave) and micro-level (intra-application) isolation. Figure 2 below shows how management, control, radio, and O-Cloud enclaves interconnect via agents and gateways, with sandboxed xApps/rApps operating inside each RIC as illustrated in Figures 3 and 4 below. Note that, in comparison to Figure 1, we omitted some 3GPP interfaces for simplicity and focused only on O-RAN interfaces.

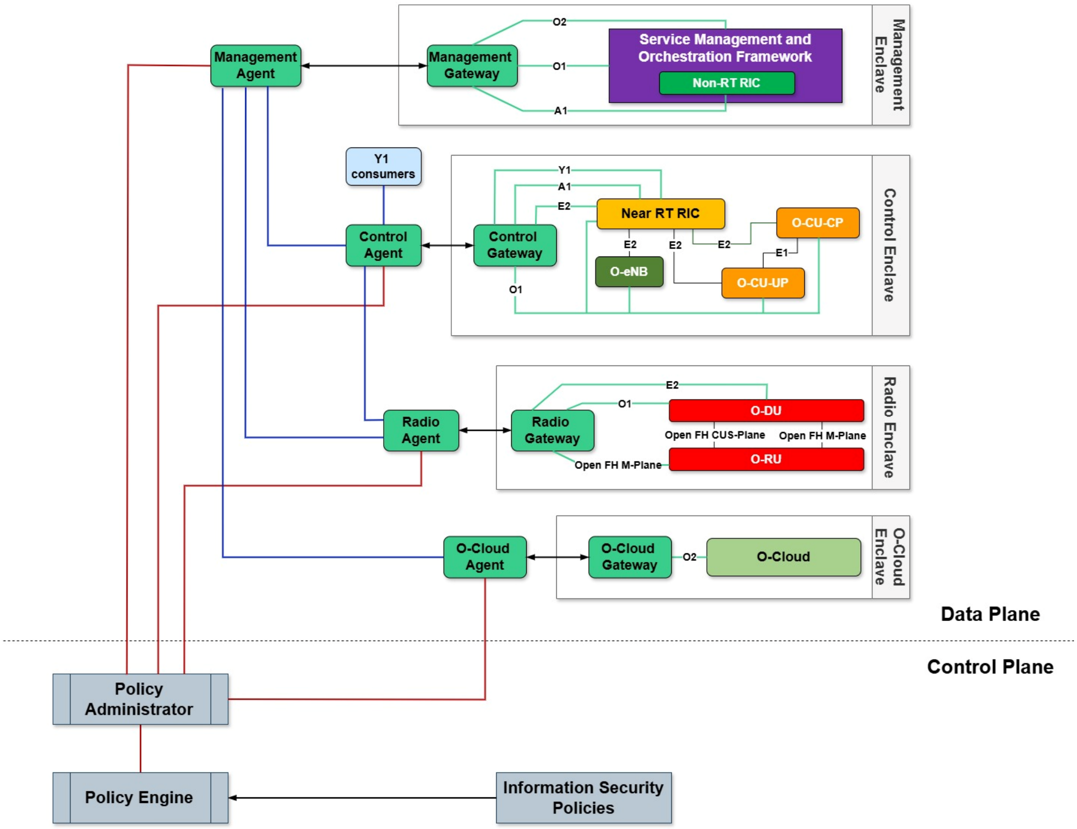

Figure 2. The Hybrid ZT Deployment Model for O-RAN, illustrating enclave boundaries (Management, Control, Radio, O-Cloud) and agent/gateway deployment.

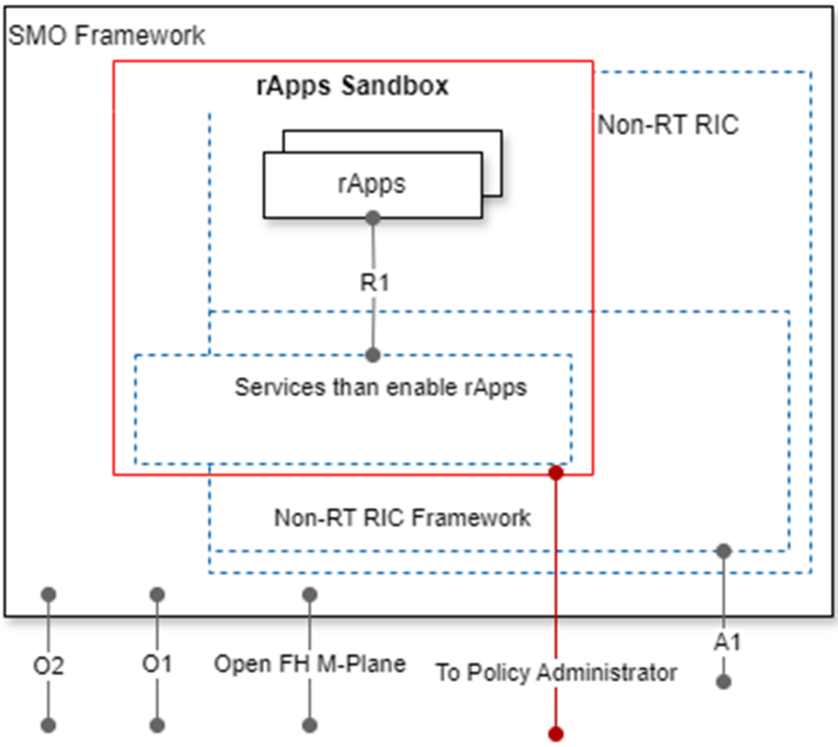

Figure 3. Sandboxed rApps.

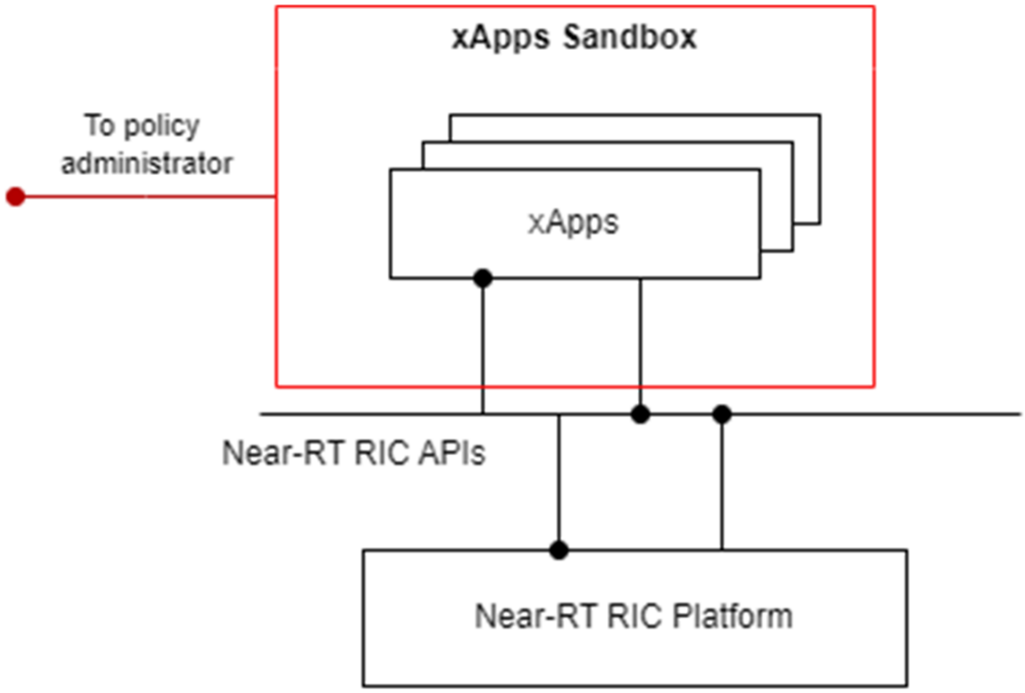

Figure 4. Sandboxed xApps.

In this hybrid model, we have (1) PDP: a central controller that manages dynamic policies for all O-RAN components; (2) Agents: lightweight PEPs on each network element that forward access requests to the PDP; (3) Gateways: boundary PEPs at enclave edges, enforcing approved traffic paths; and (4) Enclaves: logical groupings of related O-RAN functions (e.g., Management, Control, Radio, O-Cloud). The control plane, which includes the PDP, is composed of the policy engine (PE) and the policy administrator (PA), and the information security policies that should be enforced. This ensures a centralised management of these policies that are pushed to the agents associated with the various enclaves, ensuring consistent policy enforcement across the data plane. On the other hand, the PEP is distributed across three key components:

- Enclaves: These are collections of resources, comprising elements of the O-RAN architecture that share similar functionalities and support comparable control loops. Enclaves provide isolation and resource management for these services.
- Agents: These software agents facilitate connections and communication between different enclaves. An agent is responsible for directing a portion (or all) of the traffic to the policy administrator for evaluation.
- Gateways: These act as intermediaries for resources in their respecting enclaves, ensuring that ingress and egress traffic (i.e., communications) between resources across different enclaves occur exclusively through the gateways. The gateway communicates with the PA, via its dedicated agent, enforcing approved communication paths as configured by the PA.

Our hybrid model leverages a combination of traditional network-virtualisation (e.g., VLAN and NFV) for macro-segmentation and modern service-mesh controls for micro-segmentation. For instance, NFV orchestration can be used to instantiate O-RAN functions such as SMO, RIC, O-CU, and O-DU as a set of VNFs or cloud-native network functions (CNFs). Each enclave sits in its own VLAN where access controls ensure that for instance, the O-CU-VNF network segment cannot directly reach the O-DU-UP segment. On the other hand, in terms of micro-segmentation, within each enclave, every microservice (e.g., xApp, telemetry collector) runs alongside a sidecar proxy. mTLS certificates, which are issued by the PDP's PKI authenticate both client and server before any call is allowed. The PDP guarantees a unified identity governance where all enclaves' PEPs fetch policies and updated access controls from the PE.

In Figure 2 above, the operational flow commences as follows. When a resource intends to communicate with another resource, the relevant agent intercepts the access request and forwards it to the PA. The PA then submits this request to the PE for evaluation. Upon authorisation, the PA configures a secure communication channel between the agent and the resource gateway via the control plane, including the necessary security artifacts, port information, and session keys. Encrypted data transfer begins once the agent and the destination gateway establish the connection, which is terminated either upon workflow completion or as directed by the PA.

In terms of device application sandboxing, we propose that ML and AI applications as well as RAN-specific applications (rApps/xApps), operate within secure sandbox environments. These applications can directly communicate with the policy administrator to request access to required resources, allowing them to remain segmented from other assets within the system. The decision whether to run each xApp, rApp in its own sandbox, or bundle all xApps, and all rApps in their respective single sandbox, or bundle similar xApps/rApps functionality-wise in their respective sandboxes, is left for the deployment stage. The decision should consider balancing the security requirements versus the complexity and overhead from managing too many sandboxed environments. More details will be provided regarding xApps/rApps sandboxing in the following sections where we explain each enclave in Figure 2.

In terms of synergy with AI/ML security controls, our hybrid ZT model enables AI-enhanced security in three keyways. First, the PDP can ingest AI-driven threat scores from sandboxed xApps to dynamically adjust access policies. Secondly, AI workloads are sandboxed in their respective xApps/rApps and isolated via micro-segmentation, preventing compromise of adjacent functions. Thirdly, Kiali metrics, as explained in Section 4.4, provide trusted data streams to train ML models without exfiltration risks.

3.3.1. Management Enclave

It combines SMO and the Non-RT RIC since they are interdependent. The management gateway/agent enforces policies over A1 (to Near-RT RIC), O1 (to O-DU and O-eNB), and O2 (to O-Cloud). All rApps run in a single sandboxed environment. This ensures their segmentation and isolation from other system components while reducing management complexity. As identified above, rApps can communicate directly with the policy administrator as shown in Figure 3 above.

3.3.2. Control Enclave

It contains the Near-RT RIC, O-CU-CP, O-CU-UP, and O-eNB. Within this enclave, all xApps run in a single sandboxed environment providing an additional protective layer and ensuring their segmentation from other system components as shown in Figure 4 above. The control gateway handles the E2, Y1, A1, and O1 interfaces. Inside the enclave, the Near-RT RIC establishes connectivity with the O-eNB, O-CU-CP, and O-CU-UP via the E2 interface directly (i.e., without having to go through the control gateway.) Nonetheless, the security policies are still enforced inside the enclave via the control gateway. One of the main differences in comparison to the O-RAN architecture in Figure 1 is that Y1 consumers do not have direct access to the Near-RT RIC, which improves access control. Note that the control gateway also handles communications via the 3GPP interfaces including F1-c, F1-u, NG-c, Xn-c, Xn-u, NG-u, X2-u and X2-c, which were omitted from Figure 2 for simplicity.

Moreover, for implementation purposes, the control gateway can be split in terms of the functionalities and interfaces it supports. We will discuss this in more details in Section 4.

3.3.3. Radio Enclave

It includes the O-DU and O-RU. The radio gateway supports E2, O1, and Open FH M-Plane for communications with other enclaves.

3.3.4. O-Cloud Enclave

It contains the O-Cloud environment where the O-Cloud gateway supports O2 interface for communication with other resources in different enclaves.

4. Implementation Strategy of the Hybrid ZT Deployment Model for O-RAN

In this section, we focus on the security controls and specifications for communications among different components in the proposed hybrid ZT deployment model. We also provide high-level systematic plans to implement a ZT deployment model followed by details of our PoC implementation using NIST PM [36].

4.1. Multi-Layered Security Across Control & Data Planes

To provide defence in depth, a multi-layered security approach should be implemented. This involves using both IPSec and mTLS to create a robust security framework that ensures secure communication both at the network and application layers. This combination leverages the strengths of both protocols, providing comprehensive protection against a wide range of threats. This multi-layered approach is consistent with ZT principles. IPSec operates at the network layer, securing all IP packets between the communicating parties. It provides encryption, integrity, and authentication for all data transmitted over the network. IPSec will handle the secure transportation of IP packets between agents and gateways at the network layer. IPSec encrypts the entire IP packet, ensuring that the data remains confidential while in transit. By using hashing algorithms, IPSec ensures that the data has not been altered during transmission, providing protection against tampering. Moreover, IPSec can authenticate the source of the IP packets, ensuring that they come from a trusted source.

On the other hand, mTLS operates at the application layer, providing end-to-end security between applications. It ensures mutual authentication and encrypts the data between the source and the destination. mTLS will secure the application data by providing encryption, mutual authentication, and integrity verification at the application layer. Both the communicating agents authenticate each other using certificates, ensuring that both parties are trusted. mTLS also establishes a secure session that protects the data throughout the entire communication process.

4.2. Agent/Gateway Security Controls

To comply with the specified O-RAN security controls in [20], the agents and gateways will implement these controls as follows. For any communication over the A1 interface, the management and control agent/gateway will employ (1) TLS for confidentiality, integrity, and replay prevention; (2) mTLS for authenticity and data origin authentication; and (3) OAuth for authorisation. Accordingly, management and control gateways/agents enforce these controls, with the PDP/PE pushing the necessary certificates and tokens at session setup. The O2 interface uses identical constraints.

On the other hand, the E2 interface must utilise IPsec for all security controls, including confidentiality, integrity, authenticity, authorisation, data origin authentication, and replay prevention. Note that the E2 interface between the Near-RT RIC and O-DU operates within a control loop between 10ms and 1s. Therefore, the radio and control gateways and agents must ensure they facilitate this connection within these time constraints. This requirement should be considered while implementing the radio and control gateways and agents to meet the E2 interface performance requirements. It is worth noting that enclave-based deployments have been demonstrated to introduce minimal communication overhead [4], performance overhead [5], and memory usage during operation. These enclaves will therefore introduce minimal effects on the communication speed. The secure communication channel will make use of IPsec, which will be provided by PA/PE.

The O1 interface utilises TLS for confidentiality, integrity, and replay prevention; mTLS for authenticity and data origin authentication; and the Network Configuration Access Control Model (NACM) for authorisation. Connecting the SMO with the O-DU, the O1 interface operates within a control loop of 10ms to 1s. Hence, all the relevant gateways and agents that support the O1 interface should be designed to function within these time constraints. As pointed out earlier, given that enclave-based models have been shown to have minimal impact, communication speed is expected to remain unaffected. Finally, although not included in the proposed hybrid model in Figure 2 for simplicity, 3GPP interfaces are expected to follow the security capabilities recommended by 3GPP and O-RAN specifications including TLS and mTLS.

4.3. Systematic High-Level Implementation Plan for ZT Deployment in O-RAN

In this systematic plan, we explain the steps required before implementing a ZT deployment model for O-RAN architecture. This plan is utilised while implementing our proposed hybrid model as to be explained later in Section 4.4. First, we need to identify the various network elements, interfaces, software components, and data flows involved, along with the roles of human subjects and non-person entities in managing and interacting with these assets.

- **Assets.** These include O-RUs, O-DUs, O-CUs, Near-RT RIC, Non-RT RIC, SMO framework, and O-Cloud infrastructure.
- **Interfaces.** These include Open Fronthaul between O-RUs and O-DUs, E2 interface between Near-RT RIC and DUs/CUs, A1 interface between Non-RT RIC and Near-RT RIC, O1 interface between SMO and O-RAN nodes, and F1 interface between DUs and CUs.
- **Software Components.** These include xApps/rApps running on RICs for advanced control and optimisation, VNFs, and AI/ML algorithms for network management.
- **Data Flows.** This includes:
  a. Control data flows between non-RT RIC and near-RT RIC via A1 interface for policy and control information, and between near-RT RIC and DUs/CUs via E2 interface for real-time control and optimisation.
  b. User data flows from end devices to O-RUs, then to O-DUs, and finally to O-CUs before reaching the core network.
  c. Telemetry data flows from O-RUs/O-DUs/O-CUs to SMO for monitoring and orchestration via O1 interface.
  d. AI/ML data flows where data collected from network operations are fed into AI/ML models for training and inference.
- **Human Subjects.** This includes:
  a. Network Operators. Manage, monitor and maintain the O-RAN infrastructure by ensuring efficient operation, performing troubleshooting, applying configurations, and monitoring performance metrics.

b. Service Providers. Offer telecommunication services and applications that utilise the O-RAN network. Their responsibilities include ensuring service quality, managing service delivery and addressing customer requirements.
c. Vendors. Supply hardware and software components for the O-RAN ecosystem. They develop, deliver, and support network equipment and software solutions.
d. Developers. Develop, maintain, and update xApps/rApps, and other O-RAN software components. This includes application development, testing, and ensuring compatibility with O-RAN specifications.
e. Regulatory Bodies. Ensure compliance with telecommunication standards and regulations. Regulatory bodies also have responsibility to oversee network operations, ensure adherence to regulatory requirements, and protect consumer interests.
f. End Users. Consumers and businesses use services provided over O-RAN networks. Use network services, report issues, and provide feedback on service quality.

- **Non-person Entities.** This includes:

  a. Service Accounts. These are automated accounts for managing and interacting with O-RAN resources. They perform routine tasks such as data collection, software updates and resource provisioning.
  b. AI/ML models deployed within xApps/rApps for network optimisation. They analyse data, predict network conditions and suggest or apply configurations for optimisation.
  c. VNFs and PNFs. These perform specific tasks in the network. Tasks related to data processing signal transmission and network management.
  d. Management and orchestration systems. Systems within the SMO that manage network resources and orchestrate services. They coordinate the deployment and operation of network components, ensure resource allocation, and maintain service quality.
  e. Monitoring and Analytics tools. These collect, process, and analyse network data. Their responsibilities include providing insights into network performance, detect anomalies and support decision making process.
  f. Security systems that enforce security policies and detect threats. They monitor network activities, identify security breaches and apply countermeasures when needed to protect the O-RAN network.

After identifying all the elements in O-RAN architecture, we can proceed with the design, deployment, and testing phases, culminating in continuous monitoring, maintenance, and periodic review to ensure the sustained effectiveness of the deployed hybrid ZT model. These steps are carried out during the preparation phase which we outlined earlier and can be summarised as follows:

- **Preparation.** This phase has been outlined earlier where assets are identified. It also includes defining the scope and objectives of ZT deployment in O-RAN, assessing the current security posture, and identifying gaps.
- **Design.** Develop a detailed ZT implementation plan, including policies, procedures, and tools required. It also includes designing the architecture for integrating IAM, PEPs, segmentation, monitoring, and encryption within O-RAN architecture.
- **Deployment.** This phase includes deploying IAM solutions, PEPs, monitoring tools, and encryption mechanisms across O-RAN components and interfaces. Besides that, the configuration of access policies, segmentation rules, and encryption protocols.

- **Testing.** Conduct thorough testing to ensure all ZT principles are effectively implemented and integrated. In addition, penetration testing and vulnerability assessments to identify and address any security weaknesses.
- **Monitoring & Maintenance.** Continuously monitor the network for security threats and policy violations. Moreover, regularly update security policies, tools, and configurations based on evolving threats and network changes.
- **Review & Improvement.** Periodically review the effectiveness of the ZT hybrid model implementation and make necessary improvements.

4.4. Policy Machine-Based Implementation—Proof of Concept

In this section, we present a PoC implementation of the proposed ZT hybrid model for O-RAN using the following technologies (1) NIST PM; (2) Kubernetes [37]; (3) Istio service mesh [38]; and (4) Kiali for observability [39]. To simplify the implementation and mapping to the design in Figure 2, we define the following two elements.

- **Pod.** Following the terms used in Kubernetes, a pod acts as a logical boundary (i.e., enclave) encapsulating one or more related services. Pods provide isolation and resource management for these services. Each pod can contain multiple containers (i.e., services) that work together. These services are protected from external interference through policies and proxies in front or within the pod.
- **Proxy.** In this case, a proxy could serve as an agent, gateway and policy enforcer. The benefits of combining these components into a proxy include easier management, dynamic policy implementation, and reducing the attack surface. Each proxy serves as an intermediary between services within the enclaves and the outside world, or between internal services, enforcing policies. Proxies receive policy details from the control plane (i.e., the PA) and are responsible for applying them at the pod level, ensuring security and access control. The proxy functionality can be further split into ingress and egress proxies. An ingress proxy handles all incoming traffic to the pod, applying policies to ensure only authorised requests are allowed into the enclave. On the other hand, egress proxy handles traffic exiting the pod. It ensures that any data leaving the pod (i.e., enclave) complies with outbound security policies, preventing unauthorised data transmission or data leaks.

In terms of placing the proxies, two different options can be considered during the deployment phase. First, a central proxy gateway at each physical site, which is easier to harden and monitor, but concentrates all traffic through a single node, introducing a potential single point of failure. The second option is where proxies are embedded with every service. This delivers fine grained enforcement at the expense of higher operational overhead consuming higher resources in terms of memory and computational power. Our PoC implementation adopts a balanced approach whereby we deploy one proxy per enclave. This strikes a pragmatic compromise between visibility and manageability; while still allowing horizontal scaling should future traffic volumes or resilience requirements require finer granularity. Furthermore, during the implementation phase, different decisions can be made to split the functionalities of proxies, arrange services in their enclaves as pods or clusters, and split the gateway functionality to support different interfaces rather than one gateway for multiple interfaces as depicted in Figure 2.

To setup our PoC implementation, virtual machines (VMs) will be utilised. A controller VM will host the Kubernetes control plane, responsible for managing the cluster's state. It will act as the PA and the PE storing and deploying policies for access control. A worker node VM will host a service (e.g., non-RT RIC). It serves as the execution environment for the applications and services within the cluster (i.e., enclave). Test nodes VMs will be used to test access to the service in the worker node VM to ensure authorisation policies and access controls are enforced. We use Kubernetes to create and manage the services, and handle container orchestration, and resource allocation across the VMs. Istio, which is deployed as the service mesh within Kubernetes, is used to secure, connect, and control the services, enabling traffic routing, observability, and applying security policies. Kiali is used to provide GUI to visualise traffic flow, monitor the service mesh, and view connection graphs. It provides real-time insight into the services, traffic distribution, and health of the applications running.

The controller VM runs with 8 vCPUs, 32GB RAM, 60GB SSD, hosting the `kube-api-server`, `istiod` and both PA and PE. Two worker VMs of identical specification host two workloads. The first VM emulates the radio and RAN services as separate containers (i.e., RU, vDU, vCU-UP, vCU-CP), and the other VM emulates the router and core services (i.e., ethernet-router, core-network, near-RT RIC). mTLS is enabled within the network for all communications among services. Traffic generation follows the chain RU → vDU → {vCU-UP, vCU-CP} → ethernet-router → {core-network, near-RT RIC} with services issuing requests periodically to simulate real-life traffic.

To demonstrate the feasibility of our hybrid ZT model, in this PoC, we show two examples of authorisation security policy enforcement. Let this security policy be `AuthorisationPolicy` where we can layer and enforce different controls within different levels of the configuration. To clarify this concept, in the following, we explain two examples Policy 1. `AuthorisationPolicy: allow-ru-to-vdu` and Policy 2. `AuthorisationPolicy: deny-bad-cidr`, which are written in YAML. Policy 1 is scoped to the ran namespace (i.e., enclave) and vDU workload. This policy permits traffic only from the RU's SPIFFE (SPIFFE—Secure Production Identity Framework for Everyone. Available at: https://spiffe.io/) identity `spiffe://cluster.local/ns/radio/sa/ru-service-account` to talk to vDU service. By default, everything else is blocked.

On the other hand, Policy 2 below shows how network isolation can be achieved at the Container Network Interface (CNI) layer, then refined and applied at the proxy service layer via using the `Gateway` and `VirtualService` resources. First, the `Gateway` resource acts as an ingress proxy that allows HTTP traffic on port 80 only. Then, `VirtualService` steers incoming requests to the core-network service. Finally, the `deny-bad-cidr` policy blocks any IP in the specified CIDR.

Policy 1. AuthorisationPolicy: allow-ru-to-vdu

```
apiVersion:  security.istio.io/v1beta1
kind:  AuthorisationPolicy
metadata:
  name:  allow-ru-to-vdu
  namespace:  ran
spec:
  selector:
    matchLabels:
      app:  vdu  # apply to the vDU workload only
  action:  ALLOW
  rules:
  - from:
    - source:
        principals:
        - spiffe://cluster.local/ns/radio/sa/ru-service-account
```

Policy 2. AuthorisationPolicy: deny-bad-cidr

```
# 1) Expose the ingress gateway
apiVersion:  networking.istio.io/v1beta1
kind:  Gateway
metadata:
  name:  public-gw
  namespace:  istio-system
spec:
  selector:
    istio:  ingressgateway
  servers:
  - port:
      number:  80
      name:  http
      protocol:  HTTP
    hosts:
    - ‘‘*’’
---
# 2) Route all ‘‘/’’ traffic to core-nework
apiVersion:  networking.istio.io/v1beta1
kind:  VirtualService
metadata:
  name:  route-to-core
  namespace:  istio-system
spec:
  hosts:
  - ‘‘*’’
  gateways:
  - public-gw
  http:
  - match:
    - uri:
         prefix:  /
    route:
    - destination:
         host:  core-nework.core.svc.cluster.local
         port:
           number:  80
---
# 3) Block traffic from a bad CIDR
apiVersion:  security.istio.io/v1beta1
kind:  AuthorisationPolicy
metadata:
  name:  deny-bad-cidr
  namespace:  istio-system
```

```
spec:
  selector:
    matchLabels:
      istio:  ingressgateway
  action:  DENY
  rules:
  - from:
    - source:
         ipBlocks:
         - 192.0.2.0/24 # replace with the CIDR to block
```

Table 3 shows the attribute mapping that is used in our PoC. Since it is based on NIST's PM, it defines security rules as logical expressions over user attributes (UAs), object attributes (OAs), and the requested operation. The PE evaluates these attributes at run-time and issues a single allow or deny verdict. Hence, in our PoC, every service account is treated as a UA, and each workload label such as `app=vdu` or `namespace=ran` is treated as an OA. At runtime, the proxy calls the PM via the standard `ext_authz` hook, passing the caller's SPIFFE identity and the callee's fully qualified domain name. The PM looks up the caller's UA set, matches the callee to its OA set, evaluates the rule logic, and returns allow or deny. Proxies enforce decisions immediately, so no packet enters the target enclave unless the UA/OA rule permits the requested action. This delivers dynamic attribute-based access control which is fully aligned with the NIST PM framework.

Table 3. Attribute Mapping Table.

Caller UA (SPIFEE)	Target (Namespace/App)	Operation	Decision
`spiffe://.../ns/radio/sa/ru-service-account`	ran, app: vdu	any	Allow
any other principal (*implicit*)	ran, app: vdu	any	Deny
any source IP $\in$ 192.0.2.0/24	istio-system ingressgateway	any	Deny
any source IP $\notin$ 192.0.2.0/24 (*implicit*)	istio-system ingressgateway	any	Allow

Note that our PoC considered a single, centralised PA. While this consideration is advantageous for easier management, simplicity and auditability, there is a risk of a throughput bottleneck once requests or policies increase to hundreds or thousands. To mitigate this issue, we considered two options. The first one involves proxy level policy caching where the PA periodically pushes rule bundles to each proxy. This enables the proxy to decide locally on requests it has already seen, hence reducing the stress on the PA. However, this would need a robust policy update and revocation strategy. The other mitigation strategy is to have a central PA, and multiple PA instances, each responsible for a subset of enclaves. This would remove the single bottleneck point at the expense of extra infrastructure and more complex management.

5. Performance & Trade-Off Analysis of the Hybrid ZT Deployment Model for O-RAN

When applying ZT, it is essential to consider the potential trade-offs in performance and user experience alongside security [40]. It is worth noting that, since no substantial data is yet available from our work and PoC implementation, we discuss similar studies for baseline comparison in terms of latency and CPU/Memory usage. In future work, we will measure end-to-end loop latencies on an O-RAN Software Community (OSC) testbed to validate that our hybrid ZT model added per-hop latency (≈1–10 ms) does not exceed O-RAN control-loop constraints.

5.1. Baseline Performance from Published ZT Approaches

5.1.1. Latency

In Ref. [41], the authors applied an enclave model to a network and tested its performance. Their findings show that an enclave-based ZT model has trivial communication overhead during connection setup time between the client and agent, approximately 1% overhead. The enclave/IPsec adds ~1–10 ms overhead. This minimal overhead is attributed to the required policy rules in the firewall and the decryption of packets before a connection is accepted.

The performance overhead was found to be small and tolerable in the enclave-based deployment applied in Ref. [42]. When compared to traditional security frameworks, the enclave-based model in ZT architecture offers enhanced security with minimal performance trade-offs. Traditional models often involve complex key management processes and longer recovery times, which can significantly impact system performance. In contrast, the enclave-based model maintains a balance by providing robust security measures with only a slight increase in overhead, which is deemed tolerable given the substantial security benefits achieved.

Finally, existing studies show that sidecar proxies can increase request latency by up to 166% in default service mesh configurations in default mode [43]. However, Istio's new 'ambient' mode nearly eliminated that overhead. This suggests a potential performance issue hence, in our proposed model, agents/gateways should avoid unnecessary hops or use kernel-bypass techniques to minimise delay.

5.1.2. CPU/Memory Usage

The researchers in Ref. [44] tested an enclave model in a multi-cloud environment. Their results showed a minimal impact on cluster CPU and memory usage. Segmenting the network using enclaves as microparameters within a ZTA allows for granular security controls and robust data protection while maintaining business productivity [45]. This approach ensures that security measures are precise, flexible, and adaptable to changing business needs, providing a secure foundation for digital transformation and operational efficiency. In terms of device sandboxing, the authors in Ref. [46] showed that applications running in isolated compartments can be deployed in such a way that it will have low impact on overall system performance [46]. TEEs were used to achieve ZT with minimal performance overhead in 5G core networks in Ref. [47]. A comparison between having no security, other security models and ZT application sandboxing showed that application sandboxing had less impact on network and disk access performance in comparison to other security models [48].

While introducing a ZT deployment model may incur additional overheads and costs, it is possible to optimise the deployment to minimise effects. In Ref. [49], the authors enhanced the verification flexibility in the PEP, saving verification cost and optimising the process. ZT deployment models enhance network security but can introduce performance overheads, such as increased CPU and memory usage, and potential latency issues. However, strategic configurations/techniques like policy caching, hardware acceleration (e.g., TLS offload), and selective enforcement on critical flows only can mitigate these effects, balancing security with performance improvements.

5.2. Trade-Offs of Our Hybrid ZT Model

Although our proposed hybrid ZT deployment model provides comprehensive security coverage for all O-RAN interfaces, it also introduces specific performance and operational trade-offs that operators should consider.

5.2.1. Latency Overhead vs. Granular Isolation

By combining enclave-based segmentation (i.e., macro isolation) with application sandboxing (i.e., micro isolation), our hybrid ZT model can tightly contain lateral movement at both inter-enclave and intra-enclave levels. This multi-layered isolation makes it much harder for an attacker to move laterally from one O-RAN function to another (e.g., from xApp to near-RT RIC, or from O-DU to O-CU). However, each additional boundary crossing (e.g., enclave gateway hops or sandbox entry/exit) adds small latency. In practice, enclave/IPsec handshakes incur ≈1–10 ms extra per new connection [41], and sidecar proxies (e.g., Istio's default model) can increase request latency up to 166% before tuning [43]. In a 6G network where sub-10 ms control loops (e.g., E2, O1) are critical, careful placement and optimisation of proxies (e.g., using ambient mode or kernel bypass) are needed to keep total round-trip time within the 10 ms limit.

5.2.2. CPU/Memory Overhead vs. Scalable Policy Enforcement

Istio sidecar proxies combined with the NIST PM enforce attribute-based policies at runtime. This allows a single PDP to push fine-grained, per-session rules into each enclave or sandbox without modifying xApp/rApp code. On the other hand, running a proxy in front of each pod (i.e., enclave or sandbox) consumes additional CPU cycles (e.g., for mTLS/TLS handshakes) and memory. Although the authors in Ref. [44] showed enclave models add only a few percentage points to CPU/memory usage under moderate load, at large scale (e.g., hundreds of pods), the cumulative overhead becomes significant. Mitigation strategies such as policy caching in proxies or splitting the PDP into regional instances can reduce control-plane bottlenecks but increase management complexity.

5.2.3. Operational Complexity vs. End-to-End Visibility

Via Kiali/Prometheus, our hybrid ZT model provides end-to-end telemetry of all traffic flows across enclaves and sandboxes, enabling real-time assessment of every O-RAN component. Operators can trace precisely which UA/OA attributes triggered a deny or allow decision, which is vital for dynamic policy adjustments. Nonetheless, designing, deploying, and maintaining multiple logical enclaves plus sandboxed xApps/rApps requires upfront configuration effort such as defining namespaces, service accounts, SPIFFE identities, policy rules, and inter-pod network policies. This demands a steeper operational learning curve and higher DevOps/CNI skillset.

5.2.4. Single PDP vs. Policy Scalability

A central PDP simplifies auditability for all ZT policies across O-RAN domains. However, as the number of enclaves and sandboxes grows, the central PDP can become a throughput bottleneck. Caching strategies or sharding the PDP into multiple instances (e.g., each responsible for a subset of enclaves) can mitigate this but at the cost of added network traffic for policy propagation and more complex policy synchronisation. Considering these trade-offs, it can be noted that while our proposed hybrid ZT model maximises security coverage and policy granularity, it requires careful tuning and capacity planning. This is especially important for ultra-low-latency O-RAN loops and large-scale, multi-tenant deployments. Table 4 highlights how each benefit (e.g., fine-grained isolation) pairs with its corresponding cost (e.g., CPU overhead), guiding network architects in making informed implementation and deployment decisions.

Table 4. Hybrid ZT Deployment Model—Advantages vs. Limitations.

Feature	Advantages	Limitations
Isolation Granularity	• Macro-level enclave segmentation for inter-function isolation (SMO, RIC, DU/CU, O-Cloud). • Micro-level sandboxing for xApp/rApp isolation.	• Each new boundary hop (gateway or sandbox) adds latency (≈1–10 ms per connection). • Potential for small but cumulative delay in sub-10 ms control loops (e.g., E2).
Policy Enforcement	• Attribute-based decisions via NIST Policy Machine allow dynamic, per-session rules. • Uniform enforcement across all O-RAN interfaces.	• Sidecar proxies (e.g., Istio) introduce CPU cycles for mTLS/TLS handshakes (benchmarks show up to +166% latency in default mode). • PDP can become bottleneck at scale. • Extreme network conditions robustness causing potential policy decision latency
Security Coverage	• Covers all O-RAN interfaces (A1, E2, O1, O2, Y1, OpenFH) • Multi-layered defence (IPSec at network layer, mTLS at application layer).	• Requires careful key/certificate management for IPSec + mTLS across 100s of pods. • Increased attack surface on the PDP and control channel if not properly hardened.
Operational Visibility	• Fine-grained telemetry via Kiali/Prometheus for every UA→OA decision. • Real-time posture assessment of enclaves and sandboxes.	• Higher operational overhead to configure and interpret monitoring data. • DevOps skillset required for Kubernetes/Istio/CNI management.
Scalability/Manageability	• Enclaves can be scaled independently (horizontal pod autoscaling). • Policy caching or PDP sharding can improve performance.	• Increased complexity: defining namespaces, service accounts, and SPIFFE identities for each enclave. • PDP sharding adds synchronization and orchestration overhead.
Performance Overhead	• Minimal additional latency in well-tuned ambient-mode service mesh or kernel-bypass IPSec. • CPU/memory overhead limited (~5–10%).	• Out-of-the-box service mesh modes may incur sizable latency spikes unless optimised. • Memory footprint of Istio sidecars (~100–200 MB per pod).

5.2.5. Power & Bandwidth Consideration

In practical deployments of O-RAN, especially in edge and rural environments, power efficiency and operational bandwidth constraints remain critical factors. As highlighted by Nadeem et al. [50], narrowband communication protocols such as NB-IoT require lightweight and energy-efficient security mechanisms to support long device lifespans and preserve channel availability. Similarly, in O-RAN, where DUs, RUs, and edge components may operate under tight bandwidth budgets, different options are available. For instance, introduce lightweight PEP agents that implements minimal-overhead policy enforcement (e.g., DTLS instead of mTLS). Moreover, the PDP can issue long-lived access tokens for low-power devices to reduce frequent re-authentication, thereby saving power. However, this must be balanced against security requirements, as longer token lifetimes may increase vulnerability if tokens are compromised. In future work, we will explore PDP-driven energy-security trade-off algorithms for massive IoT deployments, extending the pairing concepts in Ref. [50] to ZT policy orchestration.

5.3. Adapting Security Policies Under Non-Ideal Channel Conditions

In key 6G use cases with mission-critical applications such as IIoT, the accuracy of Channel State Information (CSI) directly impacts the efficiency of dynamic security policy enforcement. Non-ideal channel conditions (e.g., due to interference, noise, or sparse multipath propagation common in industrial settings) can degrade the performance of security protocols that rely on timely communication, such as policy updates or authentication exchanges. To address this challenge, we propose integrating the efficient channel estimation techniques proposed by Wang et al. [51] into our model. The authors developed a low-complexity sparse Multiple-input-multiple-output filter bank multicarrier (MIMO-FBMC) channel estimation scheme tailored for sparse industrial channels, achieving higher estimation accuracy and reduced computational complexity compared to traditional methods. Reliable CSI enables our hybrid ZT model to dynamically adjust security policies in real time as follows. The PDP can consider real-time channel quality metrics (e.g., signal-to-noise ratio, delay spread) as an additional attribute in policy decisions. For example, if the channel quality drops below a threshold, the PDP may temporarily grant extended session durations to avoid frequent re-authentication that would otherwise exacerbate delays. On the other hand, the PEPs can switch between security protocols based on channel conditions. For instance, during periods of high interference, they might use lighter but still secure cryptographic algorithms (e.g., AES-128 instead of AES-256) to reduce processing and transmission time, ensuring that the control loop deadlines are met. These adaptations ensure that our hybrid ZT deployment model maintains robust security without compromising the operational requirements of IIoT applications, even under non-ideal channel conditions.

5.4. Robustness in Extreme Network Conditions

While our PoC demonstrates the feasibility and effectiveness of the proposed ZT deployment model in standard operational settings, extreme radio conditions such as high user mobility (e.g., vehicular scenarios), and strong interference (e.g., industrial settings) may introduce channel estimation errors that may impact the enforcement of security policies. As part of future work, we plan to extend our validation with performance simulations of the following scenarios to validate the robustness of our ZT mode model.

- High Mobility: simulate UE moving at speeds up to 120 km/h in an urban environment, causing rapid channel variations and beam misalignment. We will measure policy decision latency and impact on PEP session establishment success rate.
- Strong Interference: introduce controlled interference sources (e.g., at 2.4/5 GHz bands) to SINR levels from 0 to 10 dB. We will measure channel estimation error impact on policy update delays and security policy violation rates (e.g., expired tokens due to delayed PDP responses). Channel estimation errors will be mitigated by integrating robust algorithms such as MIMO-FBMC sparse channel estimation [51] and the training channel-based method for mmWave systems [52].

These simulations will be conducted using the OSC testbed with hardware-in-the-loop emulation of adverse RF conditions.

6. Conclusions & Future Work

Securing O-RAN in 6G networks requires moving beyond perimeter-centric defences toward a Zero Trust paradigm. This paper presented the first dedicated ZT deployment model for O-RAN that integrates enclave-based segmentation with application sandboxing, mapped comprehensively to all O-RAN interfaces and aligned with NIST ZTA tenets. By centralising policy decision via a PDP and distributing enforcement across logical enclaves, agents, and gateways, our hybrid ZT deployment model secures all O-RAN interfaces using industry-standard controls such as mTLS, IPsec, and NACM. The Kubernetes/Istio PoC with the NIST PM validates that pods can act as logical enclaves, sidecar proxies enforce per-session, attribute-based policies, and centralised PDP/PE architecture remains practical. Through trade-off analysis, which is summarised in Table 4, operators can understand latency overheads, resource costs, and operational complexity versus security benefits. We also provided a high-level implementation plan covering preparation, design, deployment, testing, and continuous monitoring stages.

For future work, we will extend our ZT deployment model to address the challenges of millimetre-wave (mmWave) based O-RAN deployments. The adoption of mmWave technology in 6G introduces unique channel characteristics (e.g., high path loss, narrow beams, and rapid spatial variations) that impact security enforcement. The rapid channel variations and potential beam misalignment could affect security deployment, particularly in terms of maintaining continuous authentication and policy enforcement. A promising research direction will be incorporating real-time channel estimation to optimise security policies and maintain reliability (e.g., the method proposed in [52]). Furthermore, future work will include quantitative evaluation on an O-RAN testbed (e.g., OSC) to measure the robustness of our ZT model under extreme network conditions (e.g., high mobility, interference), end-to-end latency and throughput under ZT controls, experimentation with distributed PDP shards, as in [47], to alleviate control-plane bottlenecks, and automated conflict resolution for concurrent xApp/rApp policies. Moreover, we will explore AI-trustworthiness techniques (e.g., explainable AI for policy decisions) and federated learning to preserve privacy across multi-vendor xApps. Ultimately, by integrating our comparison tables and architecture figures, this paper delivers a holistic blueprint for deploying Zero Trust in O-RAN toward full Zero Trust maturity.

Author Contributions: Conceptualization, M.H.E. and B.A.; methodology, M.H.E. and B.A.; software, B.A.; validation, M.H.E. and B.A.; formal analysis, M.H.E., B.A., A.R., M.M. and M.K.M.; writing—original draft preparation, M.H.E. and B.A.; writing—review and editing, A.R., M.M. and M.K.M.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Department for Science, Innovation and Technology (DSIT), UK Government, Liverpool City Region HDD (High Demand Density) project.

Data Availability Statement: Data is unavailable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Larsson, D.C.; Grövlen, A.; Parkvall, S.; Liberg, O. 6G Standardization—An Overview of Timeline and High-Level Technology Principles. 22 March 2024. Available online: https://www.ericsson.com/en/blog/2024/3/6g-standardization-timeline-and-technology-principles (accessed on 1 September 2024).
2. O-RAN Alliance. O-RAN Specifications. Available online: https://www.o-ran.org/specifications (accessed on 10 September 2024).
3. 6G Smart Networks and Services Industry Association (6G-IA). Open RAN and 6G Future Networks Development. 6G SNS IA. 2024. Available online: https://6g-ia.eu/wp-content/uploads/2024/05/6g-ia-open-sns_open-networks-status-and-future-development_ran-final.pdf (accessed on 5 August 2024).
4. O-RAN Next Generation Research Group (nGRG). O-RAN Towards 6G Report ID: RR-2023-01. O-RAN Alliance. 2023. Available online: https://mediastorage.o-ran.org/ngrg-rr/nGRG-RR-2023-01-O-RAN-Towards-6G-v1_3.pdf (accessed on 22 September 2024).
5. O-RAN Next Generation Research Group (nGRG). Architecture Principles for a Cloud-Friendly Future 6G RAN Architecture Report ID: RR-2024-01. ORAN Alliance, 2024. Available online: https://mediastorage.o-ran.org/ngrg-rr/nGRG-RR-2024-01-O-RAN%20Cloud%20Friendly%20Future%206G%20RAN-v1.2.1.pdf (accessed on 11 August 2024).
6. Alevizos, L.; Ta, V.; Eiza, M.H. Augmenting Zero Trust Architecture to Endpoints Using Blockchain: A State-Of-The-Art Review. *Secur. Priv.* **2022**, *5*, e191. [CrossRef]
7. US Government. *National Cybersecurity Strategy*; The White House: Washington, DC, USA, 2023.
8. National Cyber Security Centre (NCSC). *Zero Trust Architecture Design Principles*; NCSC: London, UK, 2021.

9. RAN Alliance. Zero Trust Architecture for Secure O-RAN v1.0. O-RAN Alliance, May 2024. Available online: https://mediastorage.o-ran.org/white-papers/O-RAN.WG11.ZTA%20for%20Secure%20O-RAN%20White%20Paper-2024-05.pdf (accessed on 21 July 2024).
10. National Institute of Standards and Technology (NIST). Advanced Security Architectures for Next Generation Wireless. 15 April 2024. Available online: https://www.nist.gov/programs-projects/advanced-security-architectures-next-generation-wireless (accessed on 13 September 2024).
11. Rose, S.; Borchert, O.; Mitchell, S.; Connelly, S. *Zero Trust Architecture*; NIST Special Publication 800-207; NIST: Gaithersburg, MD, USA, 2020. Available online: https://csrc.nist.gov/pubs/sp/800/207/final (accessed on 15 September 2024).
12. Cybersecurity & Infrastructure Security Agency (CISA), Enduring Security Framework (ESF), Security Guidance for 5G Cloud Infrastructures, Volumes 1–4. US DHS CISA, October-November 2021. Available online: https://www.cisa.gov/resources-tools/groups/enduring-security-framework-esf (accessed on 11 August 2025).
13. US DHS CISA. Zero Trust Maturity Model (ZTMM), Version 2.0. CISA, April 2023. Available online: https://www.cisa.gov/sites/default/files/2023-04/zero_trust_maturity_model_v2_508.pdf (accessed on 26 May 2024).
14. O-RAN Alliance. O-RAN Architecture Description v14.0. O-RAN Alliance. 2025. Available online: https://specifications.o-ran.org/download?id=862 (accessed on 11 August 2025).
15. Polese, M.; Bonati, L.; D'Oro, S.; Basagni, S.; Melodia, T. Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges. *IEEE Commun. Surv. Tutor.* **2023**, *25*, 1376–1411. [CrossRef]
16. O-RAN Alliance. The O-RAN ALLIANCE Security Working Group Continues to Advance O-RAN Security. 9 February 2024. Available online: https://www.o-ran.org/blog/the-o-ran-alliance-security-working-group-continues-to-advance-o-ran-security (accessed on 27 September 2024).
17. Liyanage, M.; Braeken, A.; Shahabuddin, S.; Ranaweera, P. Open RAN security: Challenges and opportunities. *J. Netw. Comput. Appl.* **2023**, *214*. [CrossRef]
18. Groen, J.; D'Oro, S.; Demir, U.; Bonati, L.; Villa, D.; Polese, M. Securing O-RAN Open Interfaces. *IEEE Trans. Mob. Comput.* **2024**, *23*, 11265–11277. [CrossRef]
19. O-RAN Alliance. O-RAN Security Threat Modelling and Risk Assessment 6.0. O-RAN Alliance. 2025. Available online: https://specifications.o-ran.org/download?id=918 (accessed on 11 August 2025).
20. O-RAN Alliance. O-RAN Security Requirements and Controls Specification 12.0. O-RAN Alliance. 2025. Available online: https://specifications.o-ran.org/download?id=914 (accessed on 11 August 2025).
21. O-RAN Alliance. O-RAN Security Protocols Specifications 12.0. O-RAN Alliance. 2025. Available online: https://specifications.o-ran.org/download?id=917 (accessed on 11 August 2025).
22. O-RAN Alliance. O-RAN Security Test Specifications 10.0. O-RAN Alliance. 2025. Available online: https://specifications.o-ran.org/download?id=920 (accessed on 11 August 2025).
23. O-RAN Alliance. O-RAN Study on Security for Service Management and Orchestration (SMO) 6.0. O-RAN Alliance. 2025. Available online: https://specifications.o-ran.org/download?id=852 (accessed on 11 August 2025).
24. Abdalla, A.; Moore, J.; Adhikari, N.; Marojevic, V. ZTRAN: Prototyping Zero Trust Security xApps for Open Radio Access Network Deployments. *IEEE Wirel. Commun.* **2024**, *31*, 66–73. [CrossRef]
25. Jiang, H.; Chang, H.; Mukherjee, S.; Van der Merwe, J. OZTrust: An O-RAN Zero-Trust Security System. In Proceedings of the IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Dresden, Germany, 7 November 2023.
26. Zaheer, Z.; Chang, H.; Mukherjee, S.; Van der Merwe, J. eZTrust: Network-Independent Zero-Trust Perimeterization for Microservices. In Proceedings of the 2019 ACM Symposium on SDN Research (SOSR '19), San Jose, CA USA, 3 April 2019.
27. The Kubernetes Authors. Using RBAC Authorization. 28 June 2024. Available online: https://kubernetes.io/docs/reference/access-authn-authz/rbac/ (accessed on 11 October 2024).
28. The Cilium Authors. Cilium—Cloud Native eBPF-Based Networking, Observability, Security. Available online: https://cilium.io/ (accessed on 11 October 2024).
29. Tigera, Inc. Calico. Available online: https://docs.tigera.io/calico/latest/about/ (accessed on 11 October 2024).
30. Chen, X.; Feng, W.; Ge, N.; Zhang, Y. Zero Trust Architecture for 6G Security. *IEEE Netw.* **2024**, *38*, 224–232. [CrossRef]
31. Dumitru, I.-A. Zero Trust Security. In Proceedings of the International Conference on Cybersecurity and Cybercrime (IC3), Bucharest, Romania, 30 April 2022. [CrossRef]
32. He, Y.; Huang, D.; Chen, L.; Ni, Y.; Ma, X. A Survey on Zero Trust Architecture: Challenges and Future Trends. *Wirel. Commun. Mob. Comput.* **2022**, 6476274. [CrossRef]
33. Basta, N.; Ikram, M.; Kaafar, M.; Walker, A. Towards a Zero-Trust Micro-segmentation Network Security Strategy: An Evaluation Framework. In Proceedings of the NOMS 2022—2022 IEEE/IFIP Network Operations and Management Symposium, Budapest, Hungary, 25 April 2022.

34. Sheikh, N.; Pawar, M.; Lawrence, V. Zero trust using Network Micro Segmentation. In Proceedings of the IEEE INFOCOM 2021—IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Vancouver, BC, Canada, 10 May 2021.
35. Mhaskar, N.; Alabbad, M.; Khedri, R. A Formal Approach to Network Segmentation. *Comput. Secur.* **2021**, *103*. [CrossRef]
36. NIST; Computer Security Resource Centre (CSRC). Policy Machine | CSRC . 6 June 2016. Available online: https://csrc.nist.gov/Projects/Policy-Machine (accessed on 10 February 2025).
37. The Kubernetes Authors. Kubernetes. 2025. Available online: https://kubernetes.io/ (accessed on 25 April 2024).
38. Istio Authors. The Istio Service Mesh. 2025. Available online: https://istio.io/latest/ (accessed on 25 April 2024).
39. Kiali. Kiali—The Console for Istio Service Mesh. 2025. Available online: https://kiali.io/ (accessed on 25 April 2024).
40. Paul, B.; Rao, M. Zero-Trust Model for Smart Manufacturing Industry. *Appl. Sci.* **2023**, *13*, 221. [CrossRef]
41. Ruambo, F.A.; Zou, D.; Yuan, B. Securing SDN/NFV-Enabled Campus Networks with Software-Defined Perimeter-Based Zero-Trust Architecture. *SSRN* **2023**. [CrossRef]
42. Bello, Y.; Hussein, A.; Ulema, M.; Koilpillai, J. On Sustained Zero Trust Conceptualization Security for Mobile Core Networks in 5G and Beyond. *IEEE Trans. Netw. Serv. Manag.* **2022**, *19*, 1876–1889. [CrossRef]
43. Barr, A.; Lavi, O.; Naor, Y.; Rampal, S.; Tavori, J. Technical Report: Performance Comparison of Service Mesh Frameworks: The MTLS Test Case. *arXiv* **2024**, arXiv:2411.02267.
44. Rodigari, S.; O'Shea, D.; McCarthy, P.; McCarry, M.; McSweeney, S. Performance Analysis of Zero-Trust multi-cloud. In Proceedings of the IEEE 14th International Conference on Cloud Computing (CLOUD), Chicago, IL, USA, 5 September 2021.
45. Cunningham, C.; Holmes, D.; Pollard, J. The Eight Business and Security Benefits of Zero Trust Business Case: The Zero Trust Security Playbook. Forrester Research, 2019. Available online: https://www.kennisportal.com/wp-content/uploads/2022/06/Akamai-the-eight-business-and-security-benefits-of-zero-trust-report.pdf (accessed on 27 August 2024).
46. Brasser, F.; Gens, D.; Jauernig, P.; Sadeghi, A.; Stapf, E. SANCTUARY: ARMing TrustZone with User-space Enclaves. In Proceedings of the Network and Distributed Systems Security (NDSS) Symposium, San Diego, CA, USA, 24 February 2019.
47. Vomvas, M.; Ludant, N.; Noubir, G. Establishing Trust in the Beyond-5G Core Network using Trusted Execution Environments. *arXiv* **2024**, arXiv:2405.12177.
48. Zhang, J.; Zheng, J.; Zhang, Z.; Chen, T.; Qiu, K.; Zhang, Q.; Li, Y. Hybrid isolation model for device application sandboxing deployment in Zero Trust architecture. *Int. J. Intell. Syst.* **2022**, *37*, 11167–11187. [CrossRef]
49. Zheng, D.; Xing, H.; Cao, X.; Xu, J. Efficient Zero-Trust-enabled Service Function Chain Deployment in Multi-Vendor Networks. *TechRxiv* **2024**. [CrossRef]
50. Nadeem, A.; Hussain, M.; Iftikhar, A.; Aslam, S. Narrowband IoT Device to Device Pairing Scheme to Save Power. In Proceedings of the IEEE 23rd International Multitopic Conference (INMIC), Bahawalpur, Pakistan, 5 November 2020.
51. Wang, H.; Xu, L.; Yan, Z.; Gulliver, T.A. Low-Complexity MIMO-FBMC Sparse Channel Parameter Estimation for Industrial Big Data Communications. *IEEE Trans. Ind. Inform.* **2020**, *17*, 3422–3430. [CrossRef]
52. Wang, H.; Xiao, P.; Li, X. Channel Parameter Estimation of mmWave MIMO System in Urban Traffic Scene: A Training Channel-Based Method. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 754–762. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.