**A1 Interface**

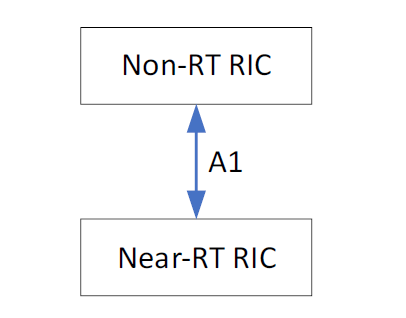
The O-RAN architecture description includes Non-RT RICs and Near-RT RICs connected by the A1 interface as shown in the figure

Fig: A1 is the interface between the Non-RT RIC and the Near-RT RIC.

**Role of A1 in the O-RAN architecture:**

The non-RT-RIC function resides in the SMO layer as described in the O-RAN architecture description. which also handles the distribution, configuration, and collection of data from the RAN nodes.

The SMO layer also includes features that support the AI/ML workflow, e.g. Training and updating of ML models as well as functions to deploy ML models and other applications as described in O-RAN WG2: AI/ML Workflow Description and Requirements. The SMO layer can also access data other than that available in the RAN functions, and this enrichment information can be used to improve the RAN's routing and optimization functions.

One of the goals of the SMO layer is to optimize the performance of the RAN to meet the SLAs mandated by the RAN. The purpose of the A1 interface is to allow the non-RT RIC to provide rule-based guidance, ML model management, and information to enhance the near-RT RIC, allowing the RAN under certain

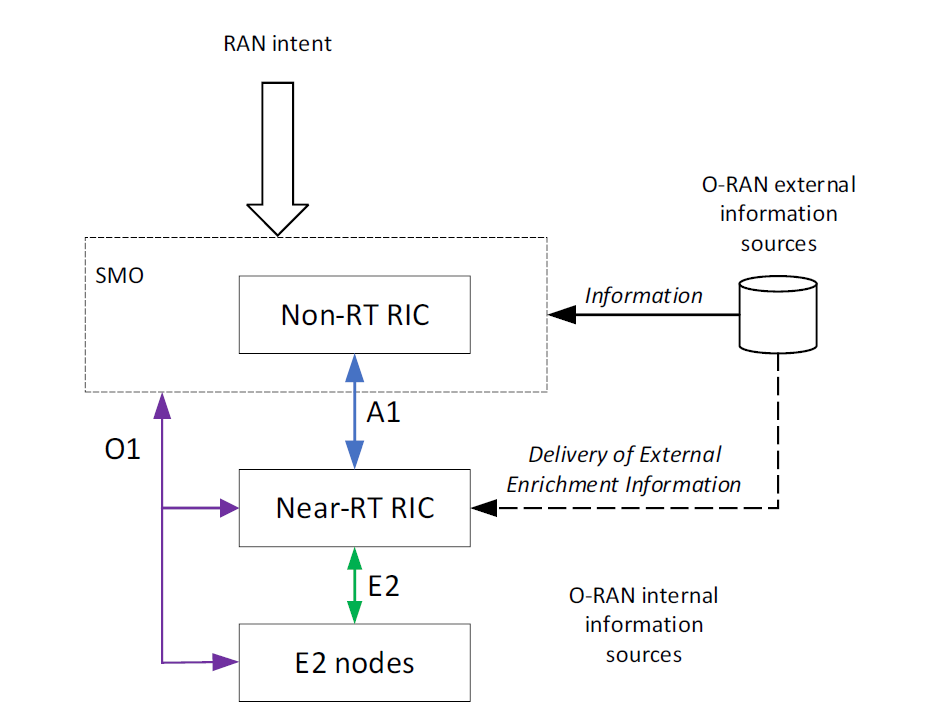


Fig: Illustration of A1 and related interfaces in the O-RAN architecture.

The nodes E2 in the figure refer to RAN nodes such as O-CU-CP, O-CU-UP, O-DU or O-eNB. A1 rules and A1 enrichment information from different sources (internal O-RAN and external O-RAN) are provided from the Non-RT-RIC via the A1-interface to the Near-RT-RIC.

**A1 service architecture**

**A1 policy management service (A1-P)**

Based on the high-level goals for the system represented in the RAN intent and observables (events and counters) provided on O1, the non-RT RIC can determine the policies provided for RIC Near-RT on O1's A1 interface. The purpose of A1 policies is to guide the RAN's performance toward the overall goal expressed in the RAN's purpose. A1 policies are declarative policies that contain statements of policy objectives and policy resources that apply to, for example, the UE and the cell.The

Non-RT RIC can manage A1 policies based on the A1 policy response and the network state provided on O1. A non-RT RIC may use O1 Observables to continuously assess the impact of A1 policies on achieving RAN intents and, depending on internal conditions, it may decide to publish/update update the objectives expressed in policies A1.

RIC Near-RT is working based on its internal functions or applications, configuration received on O1 and transient policies received on A1. A1 policies may not persist upon restarting RIC Near-RT.

To support policy enforcement in the Near-RT RIC, the Non-RT RIC may provide additional information on the A1 interface.

**A1 enrichment information service (A1-EI)**

Enrichment information is generally defined as information provided to an entity, in addition to information normally available to that entity, to improve the performance of that entity's tasks. It may be produced based on information from one or more sources, e.g. forward, combine, refine or analyze information from the input source(s).

In the O-RAN architecture, it is described that external systems can provide data enrichment to the SMO.This is called an "O-RAN external source of information", while the network functions within the RAN are an example of an "O-RAN internal information source". The

SMO can collect information from information sources inside O-RAN and outside of O-RAN. Based on this information, a non-RT RIC can derive information that may benefit internal functions or applications of a non-RT RIC and/or a proximal RIC RT, e.g. an ML model, to improve its performance.

**A1 interface general principles:**

The general principles of the A1 interface specification are as follows:

- The A1 interface is an open logical interface in the O-RAN architecture between the Non-RT-RIC functionality in the SMO and the Near-RT-RIC functionality in the RAN;

- The A1 interface is vendor independent and independent of specific implementations of SMO, Non-RT RIC and Near-RT RIC

- the A1 interface is defined extensible, allowing you to add new services and data types without having to change the protocol stack or procedures;

- The A1 interface enables policy-based targeting (destinations, resources) of internal radio resource management functions or applications that are part of the Near-RT RIC;

-A1 interface can provide basic feedback mechanism with near RT-RIC, allowing non-RT-RIC to monitor policy status.

- The RIC Near RT Policy Enforcement Service requires that policies sent over the A1 interface be expressed using a standardized mechanism (language, syntax, ...);

- the A1 interface enables the transmission of A1 enhancement information from the non-RT RIC to the near-end RT RIC as required by the near-end RT RIC.

- The A1 interface allows Near-RT to detect available enrichment information for which safe delivery can be guaranteed and to request which enrichment information to provide.

- A1 policies are created, modified and canceled by the non-RT RIC;

- Policy A1 is enforced until deleted. It is the non-RT RIC's responsibility to create and delete rules based on context information, such as RAN observability indicating achievement of the RAN intent received from O1. The A1 interface allows Near RT RICs to provide feedback to non-RT RICs on the status of A1 policy enforcement;

- EI tasks are created, modified and deleted by Near-RT RIC as required for its internal functions;

- Non-RT-RIC provides A1 enrichment information to near-RT-RIC based on EI activity as long as EI activity is enabled and until cleared.

**A1 interface specification objectives**

The A1 interface specification supports the following:

- connects non-RT RIC functionality in SMO with near-RT RIC functionality in radio access networks provided by different manufacturers.

- provides policy for each UE or group of UEs.

- provides basic policy status feedback from RT-near RICs allowing non-RT RICs to monitor policy usage.

- provides enrichment information required by Near-RT RIC.

**A1 interface capabilities**

As described in O-RAN WG2: "Non-RT RIC & A1 interfaces: Use cases and requirements" [2], the A1 interface supports:

- Policy management information transfer from Non-RT RIC to Near -RT RIC ;

- Policy response from RT-near RIC to non-RT RIC;

– Detects and requests A1 enrichment information from Near-RT RIC to Non-RT RIC and provides A1 enrichment information from Non-RT RIC to Near-RT RIC.

**Functions of the A1 interface**

**Policy management**

The purpose of the A1 policies is to allow the non-RT RIC function in the SMO to instruct the RIC function of the Near-RT, and thus the RAN, to better fulfill the intent of the RAN.

NOTE: A RAN Intent represents e.g. the BSS (Business Support System) SLA that the RAN must accommodate for all users or for a small subset of users (e.g.:QCI or slice). Based on fixed configuration and available resources, RAN tries to provide expected QoS to users.

Using observability on O1 and feedback from policy A1, the RIC not RT can conclude that the RAN intent was not achieved. The non-RT RIC can then decide to use A1 policies that allow RIC Near-RT, for example:optimize RRM for a single UE or for a group of UEs.

EXAMPLE: When a non-RT RIC understands that the resources available in a given area are not sufficient to satisfy the SLA for all users for a moment, it may decide to temporarily modify the entries. QoS targets for certain users (dynamic user groups) belonging to the same slice (predefined user groups).

There are different types of A1 fonts known as A1 fonts. A non-RT RIC does not need to use all A1 policy types, and a particular function in RIC Near-RT can only support a specific type of A1 policy. Non-RT RICs can discover available A1 policy types through the A1 interface.

NOTE: Since the A1 policy types represent the capabilities of the Near-RT RIC, they can also be expressed for the Non-RT RIC by other means.

**The policy management function**

Policy management functionality is used by non-RT RICs to provision and manage A1 policies in RIC near RT. The Non-RT RIC is responsible for defining and managing A1 policies.

This function is used to create, update, and delete A1 policies in Near-RT RIC.

This function is used to query the presence, content, and enforcement status of policy A1 in Near-RT RIC.

This function is supported by A1 policy fallback from RIC Near-RT to non-RT RIC.

NOTE 1: Since RIC Near-RT cannot modify the content of the policy, it can only notify changes in application state. If the context is changed in such a way that the execution state of the policy is changed, the non-RT RIC will be notified and may decide to remove the policy.

NOTE 2: A1 policy-compliant responses in RIC Near-RT are not transmitted over the A1 interface. Non-RT RICs can estimate the performance of A1 policies based on O1 observations.

NOTE 3: Before and after creating the A1 policy, a non-RT RIC can monitor the network to understand the impact of the A1 policy on performance.Performance tracking or tracking is not handled through the A1 interface.

**Open Network Automation Platform (ONAP)**

The Open Network Automation Platform (ONAP) is an open source project hosted by the Linux Foundation.

ONAP provides a comprehensive platform for orchestrating and automating real-time and policy-based services. ONAP enables service providers and developers to rapidly automate the initialization and configuration of physical and virtual network functions, and supports full lifecycle management operations. By consolidating open source membership resources, ONAP helps accelerate the growth of a vibrant ecosystem around a globally shared architecture and faster deployment of

network automation. time out with any single product.

ONAP is enriched with many features from version to version.Each version bears the name of a city.

| **Release Name** | **Release Version** | **Release Date** |
| --- | --- | --- |
| Kohn | 11.0.0 | 2022, December 1st |
| Jakarta | 10.0.0 | 2022, June 30th |
| Istanbul | 9.0.0 | 2021, November 15th |
| Honolulu | 8.0.0 | 2021, May 11th |
| Guilin | 7.0.0 | 2020, December 3rd |
| Frankfurt | 6.0.0 | 2020, June 11th |
| El Alto | 5.0.0 | 2019, October 24th |
| Dublin | 4.0.0 | 2019, July 9th |
| Casablanca | 3.0.0 | 2019, April 15th |
| Beijing | 2.0.0 | 2018, June 7th |
| Amsterdam | 1.0.0 | 2017, November 16th |

**‘Kohn’ Release Notes:**

This page provides release notes for the ONAP 'Kohn' release. This includes details on software versions, known limitations, and open issue reports.

Release notes are cumulative for a release, which means that this release note will have an entry for each major, minor, and maintenance release, if applicable.

Each component of the ONAP solution maintains its own component-level release notes, and links to these release notes are provided below. Details of the specific items distributed in each release of each component are maintained in the component-specific release notes.

**Features**

ONAP 'Kohn' focuses on:

• Deeper O-RAN integration with A1 and O1 policy control for SON use cases

• Improved flows for orchestrating and upgrading network functions Cloud Native (CNF) Networking with CCVPN Use Case

• Compute powerful KPIs for use in intent-based E2E network slicing

• Improved configuration query and change notification in Configuration Retention Service image (CPS)

• Improved slice parsing in control loop automation

• Continued policy framework modernization, including messaging and Native Kafka Service Grid integration

• Security enhancements including removing known vulnerabilities and adopting critical software supply chain artifacts

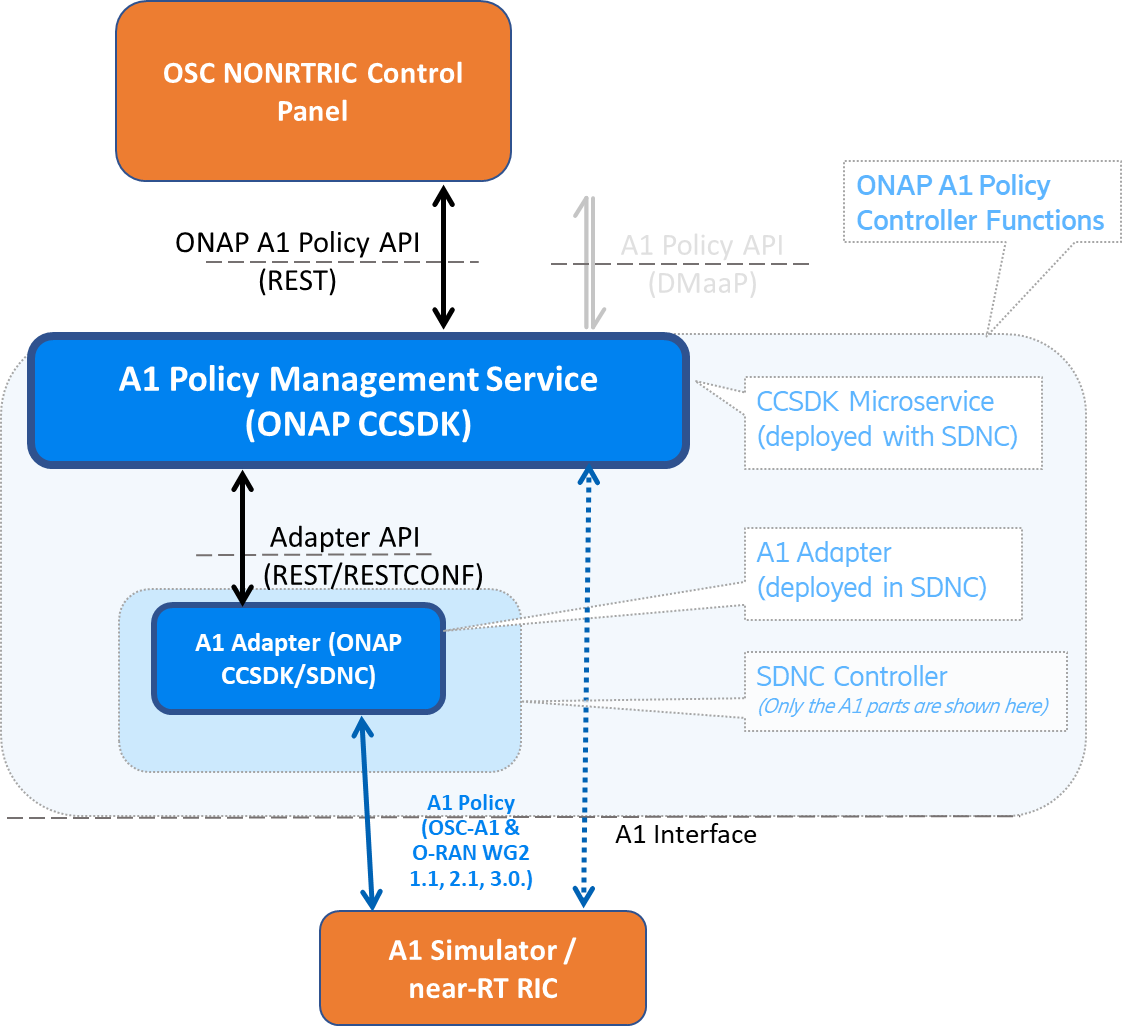
**Functional Requirements**

**Richer set of Cloud Native Functionality**

* CDS support for Application Service Descriptor
  + Onboarding ASD CSARs
  + Transformation to ONAP SDC CSAR
  + Model updates to support ASD TOSCA types
  + Support in SDC TOSCA parser
* SO improved flows around the CNF orchestration, CNF Upgrade, and minor bug fixes around the slicing use case.
  + Create multiple PNF instances in the same request
  + Support for long-running CDS process
  + Recursive orchestration Support
  + CNF Upgrade Workflow with Da-2 supported

**More information:** [‘Kohn’ Release Notes — onap kohn documentation](https://docs.onap.org/en/kohn/release/index.html#e2e-network-slicing)

**Architecture for Kohn Release:**

****

**Common Controller Software Development Kit (CCSDK).**