REC Architecture Document

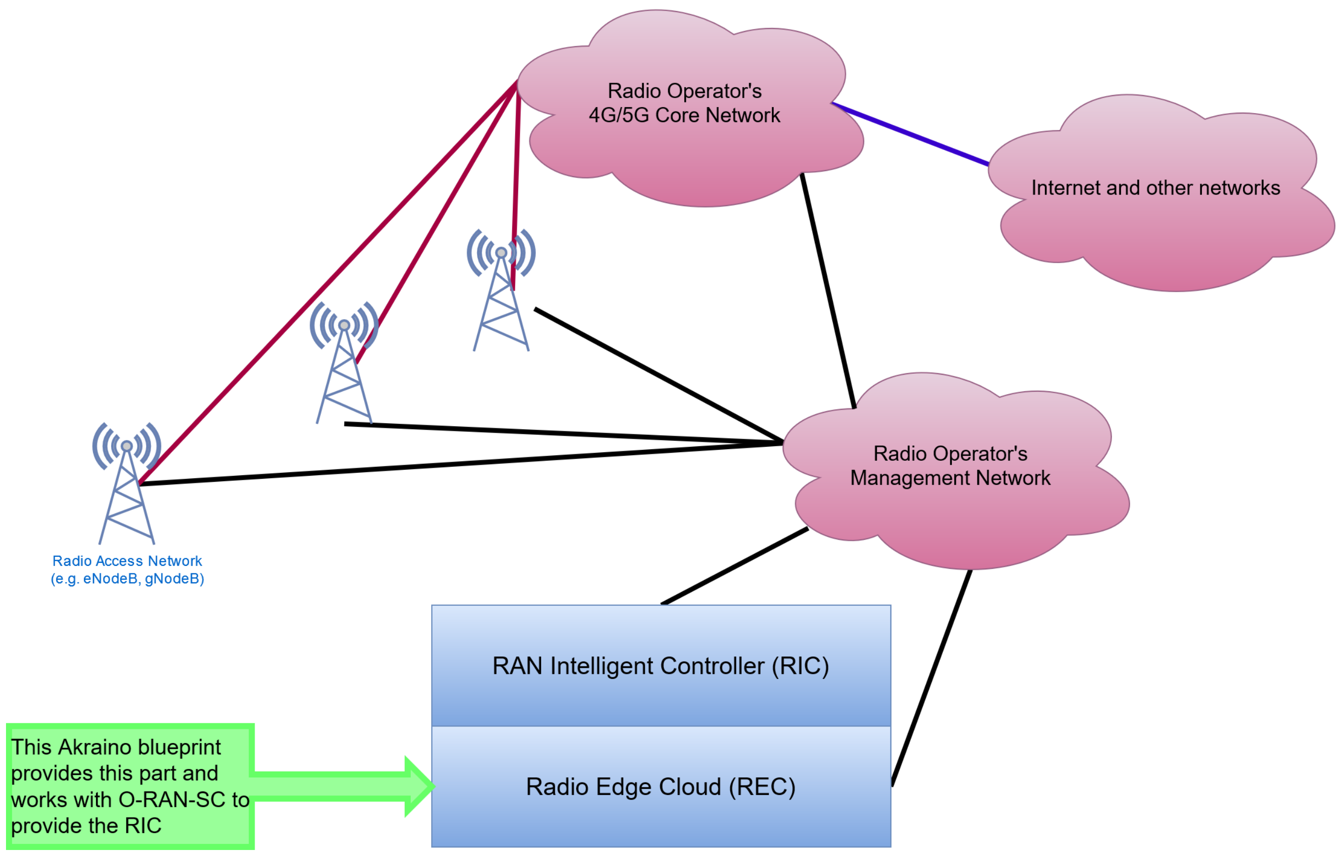
Licensing

Radio Edge Cloud is Apache 2.0 licensed. The goal of the project is the packaging and installation of upstream Open Source projects. Each of those upstream projects is separately licensed. For a full list of packages included in REC you can refer to https://logs.akraino.org/production/vex-yul-akraino-jenkins-prod-1/ta-ci-build-amd64/313/work/results/rpmlists/rpmlist (the 313 in this URL is the Akraino REC/TA build number, see https://logs.akraino.org/production/vex-yul-akraino-jenkins-prod-1/ta-ci-build-amd64/ for the latest build.) All of the upstream projects that are packaged into the REC/TA build image are Open Source.

Introduction and Purpose of the REC Architecture

Akraino Radio Edge Cloud (REC) provides an appliance tuned to support the O-RAN Alliance and O-RAN Software Community's Radio Access Network Intelligent Controller (RIC) and is the first example of the Telco Appliance blueprint family which provides a reusable set of modules that will be used to create sibling blueprints for other purpose tuned appliances.

The goal of the REC blueprint is to eventually perform fully automated bare metal deployment of the RIC. Currently The RIC must be installed on top of the REC after the REC's automated installation completes. In order to be useful, the RIC requires a 4G and/or 5G RAN that supports the O-RAN specified interfaces that are used by the RIC. The REC is intended to be deployed into a radio operator's management network with connectivity to the operator's eNodeB/gNodeB radios. The RIC provides a platform as a service environment for running "xApps" which interact with the radios to control them in useful and intelligent ways. For more details about the RIC and xApps refer to O-RAN and the O-RAN Software Community documentation.

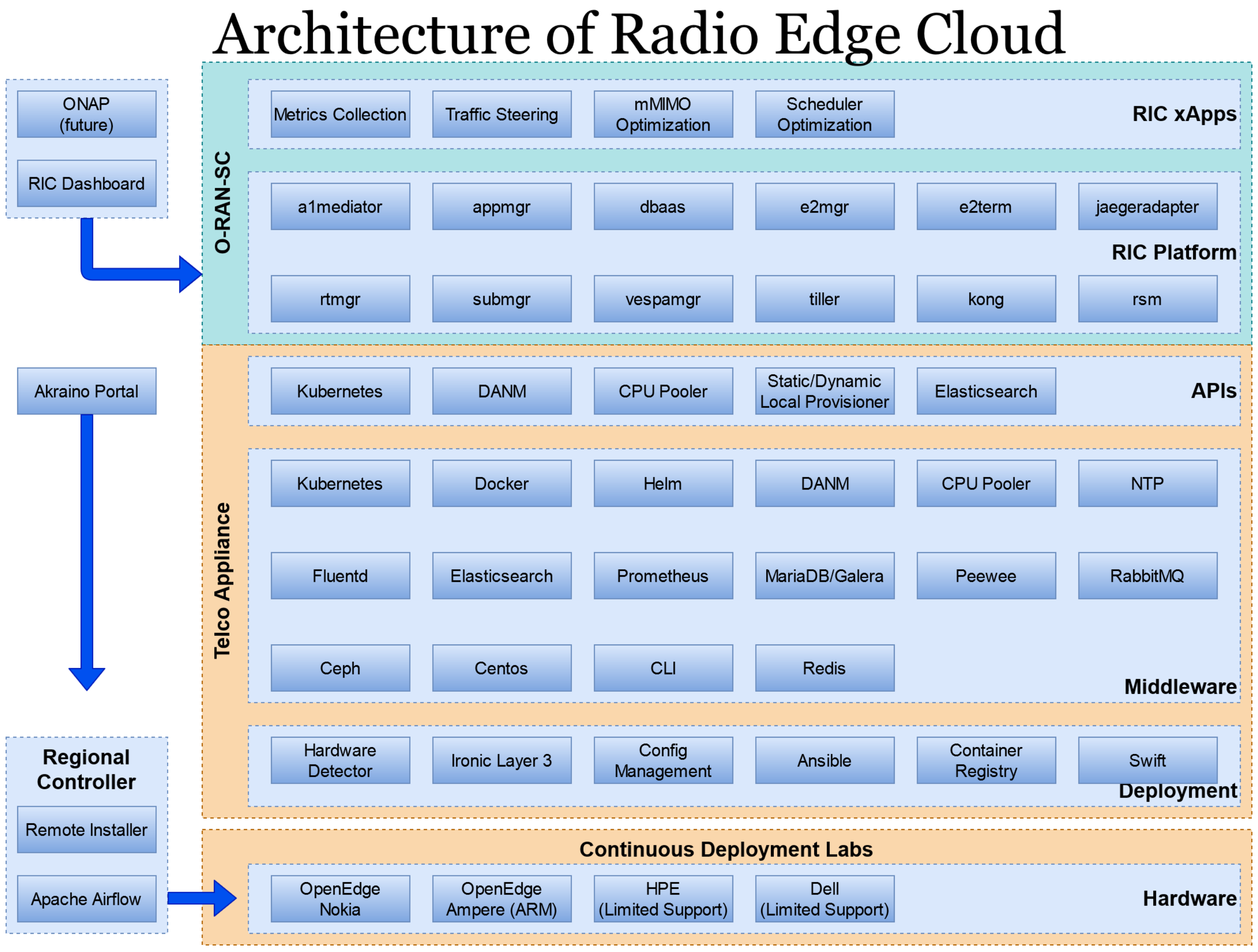


Radio Operator's 4G/5G Core Network
Internet and other networks
Radio Operator's Management Network
Radio Access Network (e.g. eNodeB, gNodeB)
RAN Intelligent Controller (RIC)
Radio Edge Cloud (REC)
This Akraino blueprint provides this part and works with O-RAN-SC to provide the RIC

- RIC on Kubernetes on “bare metal” tuned for low latency round trip messaging between RIC and eNodeB/gNodeB,
- Support for telco networking requirements such as SRIOV, dual POD interfaces, IPVLAN
- Built from reusable components of the “Telco Appliance” blueprint family

- Automated Continuous Deployment pipeline testing the full software stack (bottom to top, from firmware up to and including application) simultaneously on chassis based extended environmental range servers and commodity datacenter servers
- Integrated with Regional Controller (Akraino Feature Project) for "zero touch" deployment of REC to edge sites
- Deployable to multiple hardware models

High level architectural view:



Objectives

- Fully automated simultaneous deployment and testing on multiple hardware platforms
  - Blueprint defines exact hardware configurations
  - Each hardware variant is deployed into a Continuous Deployment system that runs the full test suite
- Appliance model automates the installation, configuration and testing of:
  - Firmware and/or BIOS/UEFI
  - Base Operating System
  - Components for management of containers, performance, fault, logging, networking, CPU
- Application:
  - RIC is the application running on the REC appliance
  - Other appliances will be created by combining other applications with the same underlying components to create additional blueprints
  - Fully automated testing includes running full application test suite

Components of Radio Edge Cloud

A detailed listing of the git code repositories hosted on the Akraino Gerrit server is available at Gerrit Code Repository Overview.

Components Used in Creation of the ISO Image

- Build-tools: Based on OpenStack Disk Image Builder
- Dracut: Tool for building ISO images for CentOS
- RPM Builder: Common code for creating RPM packages
- Specs: the build specification for each RPM package
- Dockerfiles: the build specifications for each Docker container
- Unit files: the systemd configuration for starting/stopping services
- Ansible playbooks: Configuration of all the various components
- Test automation framework

Components Which Provide Additional REC Functionality

- L3 Deployer: an OpenStack Ironic-based hardware manager framework
- Hardware Detector: Used to adapt L3 deployer to specific hardware
- Virtual installer: tooling to deploy REC on a VM (for testing only)
- North-bound REST API framework: For creating/extending REC APIs
- CLI interface
- AAA server to manage cloud infrastructure users and their roles
- Configuration management
- Container image registry
- Security hardening configuration
- Remote Installer: Docker image used by Regional Controller to launch deployer

Most Notable Upstream Components That Are Packaged Into REC with Configuration and Tuning:

- Kubernetes
- Docker
- CPU-Pooler: for enhanced CPU management in K8s
- DANM: for TelCo grade network management in K8s
- CNI: to provision specific network interfaces for containers
- SR-IOV CNI and Device Plugin: to provision SR-IOV Virtual Functions for containers
- Flannel: a CNI backend, implementing an overlay management network for containers
- Helm: K8s package manager
- etcd: a distributed key-value store
- kubedns: K8s in-built service discovery
- Fluentd: Log aggregation and forwarding service
- Elasticsearch: Log collection, store, and analysis service
- Prometheus: Performance measurement service
- OpenStack Swift: Used for container image storage
- Ceph: Distributed block storage
- NTP: Network Time Protocol
- MariaDB, Galera: Database for OpenStack components
- RabbitMQ: Message Queue for Openstack components
- Python Peewee: A Python ORM
- Redis: high-available configuration data store
- The static local provisioner is included as a beta/preview feature https://github.com/kubernetes-sigs/sig-storage-local-static-provisioner
- The dynamic local provisioner is included as a beta/preview feature https://github.com/nokia/dynamic-local-pv-provisioner

Additional Component Details

Some of the components of particular interest are documented in the following child pages:

- CM-Plugins
- CPU Pooler
- DANM - TelCo grade K8s network manager
- Gerrit Code Repository Overview
- How to Build a REC or Telco Appliance ISO
- Workload performance management and elasticity