

F Release

Welcome to the F release page for the O-RAN Software community.

The F release is completed and its source code is maintained within the master branch of each repo. The release image lists and source code can be found here for each projects.

- [Near-Real-time RIC X-APPs \(RICAPP\)](#)
- [Near-Real-time RAN Intelligent Controller Platform \(E2 Interface\) \(RICPLT\)](#)
- [Non-Real-time RIC \(A1 & R1 Interfaces\) \(NONRTRIC\)](#)
- [Operation and Maintenance \(OAM\)](#)
- [O-RAN Central Unit \(OCU\)](#)
- [O-DU High](#)
- [O-DU Low](#)
- [Simulators \(SIM\)](#)
- [Infrastructure \(INF\)](#)
- [Integration and Test \(INT\)](#)
- [Documentation \(DOC\)](#)
- [Service Management and Orchestration \(SMO\)](#)

Near-Real-time RIC X-APPs (RICAPP)

Primary Goals: Primary Goals: Expand the community working on open source xApps for O-RAN SC. Enhance the set of open source xApps in support of the R-SAC use cases (traffic steering, network slicing) as well new use cases. Update and enhance existing xApps to take advantage of the new features in xApp SDK (implemented by the xApp frameworks in C++, go, and python).

F release plan (<date>):

- New xApps
 - KPIMON-GO by HCL (pushed to next release)
- Improved xApps:
 - RC (RAN Control) by Mavenir - implements subset of E2 SM RC Ver2.0
 - AD (Anomaly Detection) by HCL: Will identify a new anomaly type (area anomaly), use geo-location as a feature. Dependency on Data Stream from KPIMON to influxDB. (Currently AD is working on Static data).
 - QP (QoE Predictor) by HCL: Include prediction for current serving cell, incorporate predicted load as a feature, provide sequence of predictions.
 - TS (Traffic Steering) by UTFPR (University, Parana, Brazil): Call RC xApp to trigger UE handover, improvements in traffic steering logic.
 - Bouncer by HCL: Increase performance and functional testing capabilities; continue identifying RIC platform bottlenecks.
 - HW (HelloWorld) demo xApps in C++, go and python by AT&T and Samsung: Add usage of more platform features, update usage of platform features that are evolving.
- VIAVI RIC test tool is used to test Traffic Steering use case.

PTL: [SUNIL SINGH](#) (former PTL still helping out: [Matti Hiltunen](#))

Jira: Count of Epics, User Stories, Tasks, and Issues: [165 issues](#)

F release highlights/accomplishments (23-Jun-2022):

- New xApp KPIMON-GO
- TS use case integration with RC xApp
- Bouncer xApp modification for RIC benchmarking
- VIAVI RIC test tool is used to test Traffic Steering use case.

F release source code, container images and deployment instructions

Code Coverage Reports : Latest reports can be found at the following Link: [Projects - O-RAN Software Community \(sonarcloud.io\)](#).

Near-Real-time RAN Intelligent Controller Platform (E2 Interface) (RICPLT)

Mission: E2 updates with first E2APv2.0 support and improvements in registration and subscription interface in xapp frameworks.

Original primary goals: ~~RIC-878 A1 mediator testing in nanobot; RIC-860 xapp testing nanobot enhancements: wider coverage of functionality, e.g., Subscription deletion // RIC-849 Second step of reimplementation of A1 mediator in golang, excl. A1AP-EI // RIC-714 support for DMS REST interface in addition to DMS CLI // RIC-698 Switch from Redis 5 to Redis 6 // partially done only and moved to post F RIC-704 E2 check, validate and define how various overload and disconnect case are handled, incl. checking highload results from dawn; RIC-886 Support two SCTP streams in same association for global (support functions) and functional procedures (ric services); RIC-916 new reconnect timer in E2 to reject new connect for x seconds; RIC-888 interface so that xApps can only query E2SMs with a given OID; RIC-722 + RIC-867 base support for E2APv2.0 (same level of functionality as before) in e2t, e2m, submgr, and nodeb-riub; RIC-638 Adapt configuration transfer capabilities as defined in E2AP v2.0; RIC-383 Epic: implement E2AP RESET as per E2APv2.0; RIC-445 SSH container/pod that includes a RIC CLI ; RIC-870 helm3 in default deployment and move ric platform deployment to ric-dep; RIC-815 ric-dep documentation only in ric-dep and it-dep only has references ; RIC-827 support deployment of ric platform in different k8s namespaces to allow two parallel installations // RIC-808 List subscriptions by xApp debug interface; RIC-375 Remove support for ASN.1 (over RMR)-based subscription in subscription manager (incl. xapp modifications) // RIC-705 update xappframework for c++ to change in xApp registration // RIC-906 time series (influx db) wrapper // RIC-642 Add REST subscription support in Python framework~~

Achieved F release highlights = high-level release notes (2022-05-18) below (note that the release image list is here: [link](#))

- we updated from E2APv1.1 to E2APv2.0 for the existing supported E2 procedures.
- Additionally E2 configuration transfer procedures (in E2 setup and via explicit E2 Node Configuration Update) are now supported.
- Enhancements in the handling of E2 disconnect+connect cycles and in SCTP stream handling
- xApps can now query the list of RAN functions by E2SM OID (and nodeb ID)
- The reimplementation of the A1 mediator in golang has proceeded, but is not yet ready and not yet replacing the existing python implementation
- Switch from Redis 5 to Redis 6 - primarily due to Redis 5 EOL
- The ric deployment is now possible without the it/dep repo
- More information in the debug interface of the subscription manager (which had some small adaptations for E2APv2.0, but these changes are not visible to xApps using the E2 subscription manager REST interface)
- In addition to the go xapp framework also the python xapp framework support xapp registration and xApp E2 subscriptions via REST.
- a new influxdb wrapper as component stslgo (shared time-series layer "golang")
- bug fixes as per [link](#). Bugs relevant to production (i.e., not test) code: RIC-903 DefaultUser setup fails when authentication is enabled for InfluxDB; RIC-895 missing static route entries for service update messages (from E2T to E2M and back); RIC-889 Subscription Manager Swagger YAML file is wrong; RIC-872 RMR routing statistic data printout crash; RIC-885 sdlgo multi-namespace SDL event namespace filter issue; RIC-880 Wrong date in sdl (C++) rpm spec file; RIC-873 SDLCLI get - writes wrongly results to stderr in success case; RIC-862 xapp-frame-py caching of error code data not happening (was: Python RMR Memory Leak); RIC-869 SDLCLI healthcheck - Redis sentinel status is wrong

For the F release of the near-RT RIC we did only limited integration testing: only the use cases: deploy RIC, deploy xApp and make E2 connection were tested.

Filled in end-of-release checklist : Release criteria checklist - [Release F](#)

PTL: [Thoralf Czichy](#)

Status 2022-06-21: From the 28 epics planned ([link](#)) we implemented 14 ([link](#)). 14 items have been moved out of the F release, e.g, because of implementation delays ([link](#)). Incomplete items: 0 ([link](#)). Additionally we fixed 19 bugs and small implementation tasks ([link](#))

F release source code, container images and deployment instructions

The list of container images for the F release ([link](#)). A demo video that shows

- how to deploy the near-RT RIC platform,
- compile connect the E2 (e2 node) simulator from the OSC simulator project and
- compile the hw-go xapp from the xapp project and use the dms_cli to deploy it.

Code coverage: [Code coverage reports](#) (current coverage and list of components that need to set up Jenkins job for auto-generation of the reports as part of CI)

Non-Real-time RIC (A1 & R1 Interfaces) (NONRTRIC)

Primary Goals:

- The primary goal of Non-RT RIC is to support intelligent RAN optimization by providing policy-based guidance, ML model management and enrichment information to the near-RT RIC function so that the RAN can optimize, e.g., RRM under certain conditions.
- It can also perform intelligent radio resource management function in non-real-time interval (i.e., greater than 1 second).
- Non-RT RIC can use data analytics and AI/ML training/inference to determine the RAN optimization actions for which it can leverage SMO services such as data collection and provisioning services of the O-RAN nodes.
- Non-RT-RIC will define and coordinate rApps (Non-RT-RIC applications) to perform Non-RT-RIC tasks.
- Non-RT-RIC will host the A1 interface (between NONRTRIC & near-RT RICs)
- Non-RT-RIC will also host the new R1 interface (between rApps and SMO/NONRTRIC services)

F Feature Scope / Achievements:

- Study & prototype Coordinated Service Exposure (SE)
 - Continue SE contribution building on the manual approaches already studied/completed.
 - Create/apply K8S configurations to isolate platform services and rApp microservices, then configure controlled secure access between service
 - Prototype CAPIF compliant API for Service/rApp registration/discovery, and service provider/consumer registration/configuration
- Data Management & Exposure (DME):
 - Pre-spec O1 PM via pre-spec R1 DME demo
 - Configure & connect to PM data - collected by SMO (ONAP)
 - Collect, Filter & Coordinate Delivery of PM data from DMaaP/Kafka to rApps over R1 (ICS)
- General activities
 - Continue to provide spec-compliant implementation of A1-Policy & A1-EI functions
 - NONRTRIC repo re-org
 - Continue to integrate and deploy SMO/NONRTRIC platform/rApps in OSC integration env.
 - Continue to expand NONRTRIC test platform & testsets
 - Show various versions rApps implemented/deployed as holistic "Automation Compositions" (ref ONAP ACM)
 - Continue to provide & integrate strawman rApps to comply with OSC RSAC integration usecases.

PTL:

- [John Keeney](#)

F release source code, container images and deployment instructions

- **Source Code:**
 - See "Gerrit /Code" below: branch "f-release"
- **Build & Run:**
 - F Release: discussions: ([NONRTRIC Wiki Release F](#))
 - F Release: How to build: ([NONRTRIC Wiki Release F - Build](#))
 - F Release: How to run:
 - Docker ([NONRTRIC Wiki Release F - Run - Docker](#))
 - Kubernetes ([NONRTRIC Wiki Release F - Run - Kubernetes](#))
- **Released Images / Tags / Versions:**
 - [NONRTRIC \(F-Release\)](#)
- **Spikes / Discussions / Studies:**
 - [Release F: Technical spikes](#)

Jira:

- Count of Epics ([20 issues](#)), User Stories, Tasks, and Issues: ([455 issues](#))

- **Wiki:** <https://wiki.o-ran-sc.org/display/RICNR>
- **JIRA:** <https://jira.o-ran-sc.org/projects/NONRTRIC/issues>
- **Gerrit / Code:**
 - nonrtrc : <https://gerrit.o-ran-sc.org/r/admin/repos/nonrtrc>
 - portal/nonrtrc-controlpanel : <https://gerrit.o-ran-sc.org/r/admin/repos/portal/nonrtrc-controlpanel>
 - sim/a1-interface : <https://gerrit.o-ran-sc.org/r/admin/repos/sim/a1-interface>
 - nonrtrc-plt-a1policymanagementservice : <https://gerrit.o-ran-sc.org/r/admin/repos/nonrtrc/plt/a1policymanagementservice>
 - nonrtrc-plt-dmaapadapter : <https://gerrit.o-ran-sc.org/r/admin/repos/nonrtrc/plt/dmaapadapter>
 - nonrtrc-plt-informationcoordinatorservice : <https://gerrit.o-ran-sc.org/r/admin/repos/nonrtrc/plt/informationcoordinatorservice>
 - nonrtrc-plt-rappcatalogue : <https://gerrit.o-ran-sc.org/r/admin/repos/nonrtrc/plt/rappcatalogue>
 - nonrtrc-plt-dmaapmediatorproducer : <https://gerrit.o-ran-sc.org/r/admin/repos/nonrtrc/plt/dmaapmediatorproducer>
- **Sonar / Test Coverage Reports**
 - nonrtrc-plt-dmaapadapter : https://sonarcloud.io/project/overview?id=o-ran-sc_nonrtrc-plt-dmaapadapter
 - nonrtrc-plt-dmaapmediatorproducer : https://sonarcloud.io/project/overview?id=o-ran-sc_nonrtrc-plt-dmaapmediatorproducer
 - nonrtrc-plt-a1policymanagementservice : https://sonarcloud.io/project/overview?id=o-ran-sc_nonrtrc-plt-a1policymanagementservice
 - nonrtrc-plt-rappcatalogue : https://sonarcloud.io/project/overview?id=o-ran-sc_nonrtrc-plt-rappcatalogue
 - nonrtrc use cases : <https://sonarcloud.io/organizations/o-ran-sc/projects?search=nonrtrc&sort=name>
- **Docs:**
 - nonrtrc : <https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtrc>
 - portal/nonrtrc-controlpanel : <https://docs.o-ran-sc.org/projects/o-ran-sc-portal-nonrtrc-controlpanel>
 - sim/a1-interface : <https://docs.o-ran-sc.org/projects/o-ran-sc-sim-a1-interface>
 - nonrtrc-plt-a1policymanagementservice : <https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtrc-plt-a1policymanagementservice>
 - nonrtrc-plt-dmaapadapter : <https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtrc-plt-dmaapadapter>
 - nonrtrc-plt-informationcoordinatorservice : <https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtrc-plt-informationcoordinatorservice>
 - nonrtrc-plt-rappcatalogue : <https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtrc-plt-rappcatalogue>
 - nonrtrc-plt-dmaapmediatorproducer : <https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtrc-plt-dmaapmediatorproducer>
- **Testing:**
 - [Function Tests](#)
- **Weekly Meetings:**
 - [Meetings](#)

Operation and Maintenance (OAM)

Primary Goals:

According to the O-RAN-SC-OAM-Architecture document, all ManagedElements (near-real-time-RIC, O-CU-CP, O-CU-UP, O-DU and O-RU) implement the O1-interface.

F Feature Scope / Achievements::

- Providing an abstract topology for rApp and CNF deployment
 - Topology links are related to the following Interfaces: A1, E1, E2, F1, N1, N2, N3, O1, O2, OFH-CUSM
 - Topology Node are related to the following components:
 - SMO: Non-RT-RIC, O1-Controller, VES-Collector, File-Client, O2-Controller;
 - 5G Core: UPF, AMF
 - RAN: Near-RT-RIC, O-CU-UP, O-CU-CP, O-DU, O-RU
 - Transport: Fronthaul Gateway
 - User Equipment
- Topology Generator, Topology Reader
- Enhancing automated test cases validating the end-to-end message flows related to OAM interfaces (O1, OpenFronthaul M-Plane)
 - Adding Wireshark for traffic flow analysis

PTL: [Martin Skorupski](#)

F release source code, container images and deployment instructions (and status)

Jira: Count of Epics ([15 issues](#)), User Stories, Tasks, and Issues: [166 issues](#)

Source Code:

- [OAM master](#)
- [OAM Controller features master](#)
- [ONAP VES Collector master](#)

Integration:

- [helm charts](#)

Release notes:

- in docs: docs.o-ran-sc.org/oam/release-notes (latest)
- in wiki: [E-Release notes](#)

Code coverage:

- [OAM: o1-controller apps](#), (2022-04-13: 55.5%)
- [OAM: o1-controller frame](#), (2022-04-13: 55.3%)
- [OAM: ves-collector](#), (2022-04-13: 74.4%)
- [OAM: message-router](#), (2022-04-13: 58.4%)
- [RIC Dashboard](#), (2022-04-13: 86.9%) archived
- [O1-TR069-Adapter](#) (2022-04-13: 22.0%) archived

O-RAN Central Unit (OCU)

Primary Goals:

- In the absence of O-CU, Radisys commercial CU image to be used for E2E testing

F Release Feature Scope:

- Radisys Commercial CU is being used as a test fixture for E2E testing
- This is containerized CU image with following
 - Release version 2.5.3
 - NG interface with SOCKET mode and veth type
 - F1 interface with SOCKET mode and veth type
 - E2 interface support
 - Software Crypto

PTL:

Status:

Radisys Commercial CU being used as a test fixture.

H/W and S/W requirements have been shared and awaiting the same to be configured.

F release source code, container images, and deployment instructions

not applicable

O-DU High

Primary Goals:**O-DU new Feature Development**

1. HARQ framework support and scheduler enhancement to prioritize retransmission
2. Enhancement of Scheduler
3. Mobility mode Support (Intra-CU handover)
4. Support for Idle Mode Paging

Feature verification

1. Closed-Loop Automation
2. 16QAM and 64 QAM (Spillover from D release)

End to End Integration support

1. TDD/Mu1/100MHz
2. FDD/Mu0/20MHz * (Spillover from D/E release)

Enhancement

1. E2 AP upgrade 2.0

F Feature Scope:

- HARQ framework support
- Scheduler enhancement
- Mobility (Inter-DU handover) support
- Idle mode paging support
- E2AP upgrade to v2.0
- End to end integration support

F Feature Achievement:

- HARQ framework support - Added
- Scheduler enhancement - enhanced to support HARQ retransmissions
- Mobility (Inter-DU handover) support - Added
- Idle mode paging support - Added
- E2AP upgrade to v2.0 - Aligned
- End to end integration support - Deferred to future release due to lab setup issues

PTL: -30c9d

Status on 30 . 2022 .

JIRA REFERENCE

1) Implementation of HARQ feature

<https://jira.o-ran-sc.org/browse/ODUHIGH-402>

Status:Completed

2) Scheduler enhancement to support multiple UEs per slot

<https://jira.o-ran-sc.org/browse/ODUHIGH-403>

Status: To be done in a future release (E2E testing for one UE per slot to be done in the lab)

3) Support for E2AP version 2.0

<https://jira.o-ran-sc.org/browse/ODUHIGH-404>

Status: Completed

4) Intra CU Handover

<https://jira.o-ran-sc.org/browse/ODUHIGH-405>

Status: Completed

5) Idle mode paging

<https://jira.o-ran-sc.org/browse/ODUHIGH-406>

Status: Completed

Intra CU Handover

<https://jira.o-ran-sc.org/browse/ODUHIGH-407>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-423>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-427>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-420>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-425>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-414>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-434>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-435>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-437>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-443>: Done

Idle mode paging

<https://jira.o-ran-sc.org/browse/ODUHIGH-408>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-419>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-415>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-438>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-442>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-446>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-452>: Done

Implementation of HARQ

<https://jira.o-ran-sc.org/browse/ODUHIGH-410>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-417>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-418>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-424>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-436>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-444>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-445>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-453>: Done

HCL's updates

O1 Enhancements

<https://jira.o-ran-sc.org/browse/ODUHIGH-430>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-431>: Done

<https://jira.o-ran-sc.org/browse/ODUHIGH-454>: Done

Dependency/Blockers:

Blocked for PTP grandmaster at the OSC lab to start the end to end integration activities

F release maintenance source code, container images, and deployment instructions

Source code: <https://gerrit.o-ran-sc.org/r/gitweb?p=o-du%2Ff12.git;a=shortlog;h=refs%2Fheads%2Ff-release>

Release notes: <https://docs.o-ran-sc.org/projects/o-ran-sc-o-du-l2/en/latest/release-notes.html#f-release>

Document: <https://docs.o-ran-sc.org/projects/o-ran-sc-o-du-l2/en/latest/>

Code coverage: To be planned as UT framework is not available to provide code coverage.

O-DU Low

Primary Goals:

Implementation of the O-DU Low Physical Layer functions for a 5G Open Access Radio Network allowing the flexibility of a software implementation coupled with the ability of incorporating hardware accelerators on a selective basis and meets the O-RAN architecture goals of scalability, mix and match multi-vendor modules that are interoperable and that can be upgraded as the standards evolve by software updates.

The O-DU Low physical layer functions follow the 3GPP TS 38 series recommendations for 5G and the 3GPP TS 36 series recommendations for LTE with the 3GPP 7.2 functional split between O-DU Low and O-RU. In 3GPP terms the O-DU Low is referred to HIGH-PHY in the functional split for 5G.

Implementation of the Open Front Haul interface to the O-RU per O-RAN WG4 CUS specifications.

Integration of this component with multi-vendor implementations of O-DU High and O-RU modules for end to end interoperability and compatibility verification.

F Release Feature Scope:

The O-DU Low F release adds support for Massive MIMO, URLLC and it is based on the commercial FlexRan 21.11 release. This release is an incremental improvement over the E-maintenance release code released on March of this year and that still needs to be integrated with the RSYS O-DU High code.

The O-DU Low F Release code is an Intel contribution in collaboration with Tieto Poland for the source code releases in the O-RAN gerrit and for the binary blobs contributed via GitHub.

For the documentation preparation of this release Intel worked with collaboration from Fransiscus Bimo and Professor Ray-Guang Cheng from National Taiwan University of Science and Technology (NTUST).

The F release can be used for end to end testing and it is based on the E maintenance release that was used for the 2021 November US O-RAN Plugfest and tested in conjunction with 2 stack partners and 2 different Test equipment vendors. The Front Haul Interface was also tested for compliance using Keysight's Front Haul Test equipment.

Container images and deployment instructions are to be provided during the F maintenance release cycle

E Release Feature Scope:

O-DU Low E maintenance release was provided in March, 2022 and it is based on the code used in the November 2021 US O-RAN Plugfest. There are additional changes in the vendor specific portion of the API to support Massive MIMO and URLLC.

Since the code was deployed in the US O-RAN Plugfest using the commercial versions of the stack partners O-DU High code, the open source version from the partners need to be integrated as well.

The validation done prior to the release used the timer-mode and the test-mac to check all the new features.

Provided O-DU Low E-Maintenance Release on March 2022 to the O-RAN gerrit repo, documentation was updated. Binary blobs for the FlexRan L1 and testmac codes were also provided in github in March.

PTL: [Luis Farias](#) , Alternate: @Chenxi Yue

- E Maintenance O-DU Low code was provided and expecting integration with RSYS.

E release source code, container images and deployment instructions

Source code and documentation including deployment instructions have been provided to the O-RAN documentation site

Simulators (SIM)

Primary Goals:

- Keep alignment with latest O-RAN specifications (O1, E2)

F Feature Scope / Achievements:

- Provide topology-service image that exposes a topology
- Provide a tool that generates a docker-compose file to start a simulated topology based on a topology file provided by the OAM project
- Keep alignment with latest O-RAN specifications for O1 and E2
- Preparations for code-coverage for the C/C++ code, will spill to G release because important code modifications are needed for Sonar integration

PTL: [Alex Stancu](#)

Jira: Count of Epics, User Stories, Tasks, and Issues: [5 issues](#)

F release source code, container images and deployment instructions

Source code: <https://gerrit.o-ran-sc.org/r/gitweb?p=sim%2Fo1-interface.git;a=shortlog;h=refs%2Fheads%2Fe-release> TODO

Container images are described here: TODO

Instructions: [Network Slicing Use Case](#) TODO update

Code coverage: in progress (sonar for C/C++ code in LF repos)

Infrastructure (INF)

Primary Goals:

- Implement the O-Cloud reference design, provide the real time performance to allow the O-DU and other components running on top of it.
- Provide interaction capabilities with other components.

F release Feature Scope:

- MultiOS support: add CentOS as the base OS
- Enable the multiple deployment scenarios: Add support for Distributed Cloud
- Enhance INF O2 implementation:
 - Enable O2 DMS by proxying k8s API endpoint
 - Enable O2 IMS for distributed cloud

PTL: [Jackie Huang](#)

F release highlights/accomplishments (7 . 2022 .):

- MultiOS support:
 - Add CentOS as the base OS
 - Two images will be provided:
 - Yocto based image: supports AIO-SX, AIO-DX, AIO-DX + workers
 - CentOS based image: supports AIO-SX, AIO-DX, AIO-DX + workers, standard, Distributed Cloud
- Enable the multiple deployment scenarios:
 - Add support for Distributed Cloud on the CentOS based image
- Verified sample application deployment:
 - Deploy FlexRAN 21.11 on INF (CentOS based)
- Enhance INF O2 implementation:
 - Enable O2 DMS by proxying k8s API endpoint
 - Enable O2 IMS for distributed cloud

Jira: Count of Epics, User Stories, Tasks, and Issues:

Update at 9 . 2022 .

- EPICs:

- INF-230 - ... ☐
- INF-274 - ... ☐
- INF-262 - ... ☐
 - INF-263 - ... ☐
 - INF-264 - ... ☐
 - INF-275 - ... ☐
 - INF-276 - ... ☐

- Tasks:

- INF-277 - ... ☐

- Issues:

- INF-241 - ... ☐
- INF-271 - ... ☐
- INF-272 - ... ☐
- INF-273 - ... ☐

Test status:

Code coverage:

- INF platform
 - [Code coverage report: o-ran-pti-rtp - \(sonarcloud.io\)](#)
 - INF is a downstream project of [StarlingX](#) and [Yocto Project](#), the above coverage report may not reflect the real code coverage so we also need to refer to the status from upstream projects.
- O2
 - Total coverage: 54%
 - Detail report: [cov-report_20220609.txt](#)

Release Note:

- [Release-notes — pti-rtp master documentation \(o-ran-sc.org\)](#)

E release source code, images and deployment instructions

- Each repository has a branch named "f-release" that can be accessed using git:
 - For pti/rtp repo[[gitweb](#)]: git clone --branch f-release "<https://gerrit.o-ran-sc.org/r/pti/rtp>"
 - For pti/o2 repo[[gitweb](#)]: git clone --branch f-release "<https://gerrit.o-ran-sc.org/r/pti/o2>"
- Images for INF project
 - ISO image for INF:
 - CentOS based image: [inf-image-centos-all-x86-64.iso](#)
 - Yocto Based image: [inf-image-yocto-aio-x86-64.iso](#)
 - Container image for o2:
- Deployment instruction:
 - [Installation Guide](#)
 - [Getting Started /Sample test process](#)
 - [Deploy FlexRAN 22.03 on INF F-Release](#)

Integration and Test (INT)

Primary Goals: To support OSC project CI pipeline. To test and validate the components and use cases

Feature Scope:

- To set up test automation that can run at release time to verify features and integration (initial efforts to make use of the Open Test Framework were incomplete due to resource issue)
- To add CI/CD Jenkins job against the it/dep repository to start regular validation of OSC components installation/deployment and corresponding health check, which also lay the foundation for future test cases implementation (across OSC components, end-to-end use cases, etc.)
- Work with OSC open labs (US east coast, US west coast, Asia Pacific) to facilitate community testing.
- Explore the POWDER testbed for OSC integration test needs.

PTL: [James Li](#)

Jira: Count of Epics, User Stories, Tasks, and Issues: [54 issues](#)

F release source code, container images and deployment instructions

not applicable

Documentation (DOC)

Primary Goals: TODO

Bronze Feature Scope: TODO

PTL: [weichen ni](#)

Jira: Count of Epics, User Stories, Tasks, and Issues:

D release source code, container images and deployment instructions

not applicable

Service Management and Orchestration (SMO)

Primary Goals: The primary goal of the SMO project is to orchestrate and manage different network functions within O-RAN. It supports several interfaces, O1, O2, O1/VES and others. It does this orchestration and management by integrating different software artifacts of existing open-source projects to create a fully functional open-source Service Management and Orchestration (SMO).

F Feature:

The focus of the F-release is around two major interfaces, O1/VES and O2. O1 continues to be a stable interface with no change requirements.

- O1/VES interface
 - Add support for StdDefined messages.
 - Expand on the list of PM counters supported
- O2 interface
 - Tacker project will commit VNF support

PTL: [Mahesh Jethanandani](#)

Status:

- O1/VES interface
 - Added support for StdDefined messages
 - Support for PM counters, although this is blocked the Copyright issue related to 3GPP/O-RAN.
- O2 interface
 - Tacker project committed. See details below.

Jira: Count of Epics ([0 issues](#)), User Stories,

Tasks:

✓	SMO-143	Handle oran-file-management netconf notifications in SDNR	Unassigned	bhanu chandra	=	TO DO	Unresolved	. 06, 2023	. 06, 2023
✓	SMO-142	Handle oran-software-management netconf notifications in SDNR	bhanu chandra	bhanu chandra	=	TO DO	Unresolved	. 06, 2023	. 06, 2023
✓	SMO-141	Enable/disable netconf subscriptions while adding a device in SMO GUI	bhanu chandra	bhanu chandra	=	TO DO	Unresolved	. 06, 2023	. 06, 2023
✓	SMO-138	Support Accelerator framework to deploy vDU	Toshiaki Takahashi	Toshiaki Takahashi	=	TO DO	Unresolved	. 25, 2023	. 22, 2023
✓	SMO-137	O2dms FM/PM	Toshiaki Takahashi	Toshiaki Takahashi	=	TO DO	Unresolved	. 25, 2023	. 25, 2023
✓	SMO-136	Integration with other SMO components for automated App Healing	Toshiaki Takahashi	Toshiaki Takahashi	=	TO DO	Unresolved	. 25, 2023	. 25, 2023
✓	SMO-128	Authenticate InfluxDB API	Unassigned	santanu de	=	TO DO	Unresolved	. 08, 2023	. 28, 2023
✓	SMO-127	Resolve Sonar violation	Unassigned	santanu de	=	TO DO	Unresolved	. 08, 2023	. 28, 2023
✓	SMO-126	Upgrade Standard defined validator to support new Standard defined domain	Unassigned	santanu de	=	TO DO	Unresolved	. 08, 2023	. 28, 2023
✓	SMO-125	Implementation of Kubernetes	Unassigned	santanu de	=	TO DO	Unresolved	. 08, 2023	. 28, 2023
✓	SMO-119	Automated API Conformance testing	Toshiaki Takahashi	Toshiaki Takahashi	=	IN PROGRESS	Unresolved	. 15, 2022	. 25, 2023
✓	SMO-103	PNF Registration event should result in O1 initiating connection to the NF	santanu de	Mahesh Jethanandani	=	TO DO	Unresolved	. 18, 2022	. 18, 2022
✓	SMO-91	Add fault-id and affected-objects to the faults page	bhanu chandra	bhanu chandra	=	TO DO	Unresolved	. 08, 2022	. 07, 2022
✓	SMO-88	Refresh SDNR image to the latest on SMO O1 project	bhanu chandra	bhanu chandra	=	IN PROGRESS	Unresolved	. 29, 2022	. 14, 2022
✓	SMO-80	Integration with other SMO components for automated App instantiation	Toshiaki Takahashi	Toshiaki Takahashi	=	IN PROGRESS	Unresolved	. 17, 2022	. 25, 2023
✓	SMO-71	Documentation for smo-ves project	santanu de	Mahesh Jethanandani	=	IN PROGRESS	Unresolved	. 30, 2022	. 23, 2022
✓	SMO-10	Convert NETCONF notifications into VES events	bhanu chandra	Mahesh Jethanandani	=	TO DO	Unresolved	. 16, 2021	. 06, 2021
✓	SMO-3	Create sample configuration for FH interface YANG model	santanu de	Mahesh Jethanandani	=	TO DO	Unresolved	. 08, 2020	. 16, 2020

: 18

and Issues: [6 issues](#)

F release source code, container images and deployment instructions

- Docker image and instruction on how to install SMO O1 NETCONF interface can be found [here](#).
- Instructions on how to install SMO O1/VES interface can be found [here](#).
- For the O1 interface the repository can be found at "git clone --branch e-release "https://gerrit.o-ran-sc.org/r/smo/o1mo/o1", whereas the O1/VES repository can be had by "git clone --branch e-release "https://gerrit.o-ran-sc.org/r/smo/vs"
- For the O2 Interface (Tacker) project
 - Source code: <https://opendev.org/openstack/tacker>
 - Code coverage: [Coverage report \(rackcdn.com\)](#) (Latest OpenStack verification results)