E Release (Dec 2021)

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

The E 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: 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).

E release plan (12-01-21):

- New xApps: RC (RAN Control) by Mavenir - implements subset of E2 SM RC
- Improved xApps:
    - LP (Load Prediction) by ChinaMobile: Include trained ML model, will populate predictions in inFlux DB
    - 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.
- Integration of AD, QP, TS, LP, RC, and KPIMON with Viavi simulator.
- Extensive performance benchmarking of the RIC platform using Bouncer and E2 Simulator (HCL)
- Design for xApps to support network slicing use case.

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

Jira: Count of Epics, User Stories, Tasks, and Issues: 165 issues

E release highlights/accomplishments (12-14-21):

- The release introduces a new xApp RC (RAN Control) from Mavenir. RC implements a subset of the new E2 SM RC to support traffic steering end to end use case.
- QP (QoE Prediction) xApps from HCL include prediction for current serving cell, incorporate predicted load as a feature, provide sequence of predictions.
- The Bouncer xApp from HCL test RIC platform with Multiple E2Sim nodes.
- The LP (Load Prediction) xApp from CMCC now includes a trained ML model (and the issue of how to handle large trained ML models was resolved.)
- The TS (Traffic Steering) xApp maintained by the Federal Technological University of Parana, Brazil includes the ability to call the new RC xApp via gRPC-based interface.
- The demo HW (HelloWorld) xApps in python and go from Samsung include additional functionality.
- The AD, QP, and TS xApps have been integration tested using Viavi E2 tester.

E release source code, container images and deployment instructions

Each repository has a branch named "e-release" that can be accessed using git. For example, the source code for the AD xApp can be retrieved using "git clone --branch e-release "https://gerrit.o-ran-sc.org/r/ric-app/ad". The other xApps in the E release can be found at ric-app/qp, ric-app/ts, ric-app/lp, ric-app/hw, ric-app/hw-go, ric-app/hw-python, ric-app/mc, ric-app/bouncer, ric-app/rc, and scp/ric-app/kpimon. Note that the other ric-app repos are obsolete.

Note that this branch is in maintenance and all new development is done in branch "master".

In order to deploy the E release xApps, you can re-use the pre-created container images as defined here and the instructions on testing the xApps can be found here.

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

Mission: E2 updates with first E2APv1.1 support and improvements in E2 subscription handling.

Original primary goals: Update to E2APv1.1 (~~E2 Node configuration transfer in E2 Setup and E2 Configuration Update (RIC-638, even if likely changing again in E2APv2.0~~) and E2SM OID support in internal E2SM function query interfaces (RIC-640)) // RIC-809 Subscription manager to delete subscriptions in case of E2 disconnect (incl. Xapp changes) // RIC-796 sub mgr and xapp-frame error cases // ~~RIC-793 Prevent A1 Job ID conflicts from multiple RICs using the same A1 producer (SMO)~~ // Partial only: RIC-647 (first step of reimplementation A1 mediator in golang to avoid A1 being the only python container in RIC platform // RIC-709 E2 stats exposing individual counters instead of groups //~~ RIC-714 support for DMS REST interface in addition to DMS CLI~~ // RIC-113 DB: SDL CLI for debugging and testing // RIC-110 FindKeys/GetAllKeys SDL API to support glob-style patterns // RIC-676 Update to Ubuntu 20.04 as base image for all containers // improvements in xApp testing // 29 Epics planned: link and 30 items as stretch goals: link

Achieved E release highlights = high-level release notes (2021-12-03) below (note that the release image list is here: link)

- we updated from E2APv1.0 to E2APv1.1. The platform now stores OIDs (introduced in E2APv1.1) for the E2SM of E2 function definitions in RNIB. Since E2APv1.1 is backwards compatible with 1.0, you can still connect E2 nodes that support E2APv1.0. Note that for the next release we plan to switch to E2APv2.0 only.
- The E2 subscription manager now automatically deletes its stored subscriptions if it gets notified (by the E2 manager) of E2 nodes having disconnected. xApps are expected to do the same and need to re-issue their subscriptions once the E2 node is reconnected. This behavior is different to earlier behavior where the subscription manager kept the subscriptions in such situations. Note that the standard requires the E2 node to delete its subscriptions if there's the E2 interface is disconnected.
- The E2 subscription manager now handles various error scenarios that previously were not handled.
- We will continue the re-implementation of the A1 mediator in golang in release F. The first parts are already implemented, but in the E release we stay with the "old" python-implementation of the A1 mediator.
- On SDL side we now have a SDL CLI that can be used in testing (instead of direct usage of the Redis CLI). The SDL API for findkeys/getkeys now supports glob-style patterns.
- The golang SDL API/library now handles namespaces as part of its function signatures instead of this being a global parameter. This eases usage of multiple namespaces from the same application.

For the E 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 E

PTL: Thoralf Czichy

Status 2021-12-08: From the 29 epics planned (link) we implemented 16 (link). 13 items have been moved out of the E release, e.g, because of implementation delays (link). Incomplete items: 0 (link).

E release source code, container images and deployment instructions

Each repository has a branch named "e-release" that can be accessed using git: "git clone --branch e-release "https://gerrit.o-ran-sc.org/r/ric-plt/e2". Make sure to replace the URL with correct repositories. Note that this branch is in maintenance and all new development is done in branch "master". The complete list of repositories belonging to the RIC platform is defined here: Scope of the near-RT RIC platform and its components (summary).

In order to deploy the E release of the near-RT RIC platform you can re-use the pre-created container images as defined here. The same instructions as always apply, i.e., follow the general latest instructions: https://docs.o-ran-sc.org/projects/o-ran-sc-ric-plt-ric-dep/en/latest/ Installing Near Realtime RIC in RIC Cluster, **but** make sure to use "git clone --branch e-release ..." instead of "git clone ..." when cloning it/dep and ric-plt/ric-dep. If you clone it/dep and use the "ric-dep" as submodule, make sure to run: "cd ric-dep; git checkout e-release" after "git submodule update --init --recursive --force"

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 also host the new R1 interface (between rApps and SMO/NONRTRIC services)

E Feature Scope / Achievements:

- E Release Priorities:
  - ONAP Control Loop -> O-RAN rApp : “The *rApp-ification* of ONAP Control Loops”
    - Adopt ONAP CL work as a starting point, continue to identify gaps, then extend
    - Identify & motivate where an rApp is different from a CL
    - Types of rApps:
      - Microservice-based rApps
      - Non-Microservice-based rApps
  - NONRTRIC Service Exposure/Gateway -> O-RAN R1 : “The *R1-ification* of Service Exposure”
    - Service-independent aspects
    - Types of exposure support in R1:
      - Microservice-based rApps & Service
      - Non-Microservice-based rApps & Service
  - Use cases of rApps & Exposing specific Services via R1
    - Requirements drivers & demonstrators
    - O-RU FH recovery (multiple), O-DU Slice Assurance (multiple), Existing Function Tests, various other use cases in ONAP
  - Continued Evolution & Support for A1 functions
- NONRTRIC Functions:
  - Integrated A1 Adapter from ONAP (A1 Policy (A1-P) controller – mediation)
  - Integrated A1 Policy Management Service from ONAP (A1 Policy (A1-P) controller)
  - rApp/Control Loop Manager (ONAP & OSC)
  - OSC Information Coordinator (controller – Data Management & Exposure & A1 Enrichment Information (A1-EI) Job management)
  - OSC Non-RT-RIC Control Panel (GUI – for A1-P & A1-EI Job management)
  - OSC A1 Simulator (a stateful test stub to simulate near-RT-RIC end of A1 interface – A1-P & A1-EI)
  - Initial OSC APP catalog (for registering/querying APPs)
  - K8S Helm Chart LCM Manager - for APP µServices etc. (ONAP & OSC)
  - Exposure Gateway Functions
  - Coordinated service exposure for R1 interface
  - DMaaP Information Producer Mediator/Adapter (multiple)
- E Release: discussions: (NONRTRIC Wiki Release E)
- E Release: How to build: (NONRTRIC Wiki Release E - Build)
- E Release: How to run: (NONRTRIC Wiki Release E - Run)

PTL:

- John Keeney

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**:
  - nonrtric : *https://gerrit.o-ran-sc.org/r/admin/repos/nonrtric*
  - portal/nonrtric-controlpanel : *https://gerrit.o-ran-sc.org/r/admin/repos/portal/nonrtric-controlpanel*
  - sim/a1-interface : *https://gerrit.o-ran-sc.org/r/admin/repos/sim/a1-interface*
- **Docs**:
  - nonrtric : *https://docs.o-ran-sc.org/projects/o-ran-sc-nonrtric*
  - portal/nonrtric-controlpanel : *https://docs.o-ran-sc.org/projects/o-ran-sc-portal-nonrtric-controlpanel*
  - sim/a1-interface : *https://docs.o-ran-sc.org/projects/o-ran-sc-sim-a1-interface*
- **Testing:**
  - Function Tests
- **Weekly Meetings**:
  - Meetings

Operation and Maintenance (OAM)

Primary Goals:

- Support of RSAC VES architecture updates
  - Deployment of VES-Collector and Message Bus at the edge
- Support of O-DU use case ()
  - Definition of the to be used yang model.
- Support of pre-spec PM streaming - VES like approach (see https://wiki.o-ran-sc.org/download/attachments/35881433/E_rel_v3.pptx?api=v2 page #97)
  - creation of a CR to WG10 OAM Interface Specification - "Model-driven PM streaming"

See more details on OAM wiki: E-Release

E Feature Scope:

- Switch to Opendaylight version Silicon-SR2
- Standard-defined VES for
  - NotifyNewAlarm
  - NotifyAlarmCleared
  - NotifyHeartbeat
  - NotifyMoiChanges
  - ~~PerfDataStreamingMNS~~ (does not include measurements - therefor a O-RAN-SC specific message-body required - ...hello-world.oas3.yaml)

PTL: Martin Skorupski

E 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:

- to be done

Release notes:

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

O-RAN Central Unit (OCU)

Primary Goals:

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

D Release Feature Scope:

- Radisys Commercial CU being used as a test fixture for E2E testing

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.

D release source code, container images and deployment instructions

not applicable

O-DU High

Primary Goals:

New Feature Development

- Support for Multi UE (Connected = 16, Active =4, Per Slot = 1)
- Only slot based round robin scheduling support for multi UE scheduling
- Support for Multi Bearer
- Basic RAN slicing support
- Support for HARQ

Feature Verification

- Closed Loop Automation Feature Verification
- 16 QAM and 64 QAM Support Verification

End to End Integration Support

- TDD/Mu1/100MHz
- FDD/Mu0/20MHz * (Spillover from D release)

E Feature Scope:

- Support for Multi UE (Connected = 16, Active =4, Per Slot = 1)
- Only slot based round robin scheduling support for multi UE scheduling
- Support for Multi Bearer
- Basic RAN slicing support
- Support for HARQ
- End to End Integration Support (TDD and FDD stack validation)
- Closed Loop Automation Feature Verification

PTL: user-30c9d

Status on 12/15/2021

JIRA REFERENCE

1) Support for Multi bearers

https://jira.o-ran-sc.org/browse/ODUHIGH-351

Status: Completed

2) Support for Multi UE

https://jira.o-ran-sc.org/browse/ODUHIGH-352

Status: Completed

3) Network Slicing support

https://jira.o-ran-sc.org/browse/ODUHIGH-363

Status: Completed

4) Resource allocation in time domain changes to meet flexible k0, k1 and k2 values

https://jira.o-ran-sc.org/browse/ODUHIGH-340

Status: Completed

5) Support for cell configuration over O1 interface

https://jira.o-ran-sc.org/browse/ODUHIGH-361

Status: Completed

12/08/2021:

Support for Multi UE feature

Jira:

https://jira.o-ran-sc.org/browse/ODUHIGH-352

Status

Completed

Support for Multi Bearer feature

Jira:

https://jira.o-ran-sc.org/browse/ODUHIGH-357

https://jira.o-ran-sc.org/browse/ODUHIGH-358

https://jira.o-ran-sc.org/browse/ODUHIGH-359

Status

Completed

Support for Basic RAN Slicing feature

Jira:

https://jira.o-ran-sc.org/browse/ODUHIGH-365

https://jira.o-ran-sc.org/browse/ODUHIGH-366

https://jira.o-ran-sc.org/browse/ODUHIGH-368

https://jira.o-ran-sc.org/browse/ODUHIGH-369

https://jira.o-ran-sc.org/browse/ODUHIGH-371

https://jira.o-ran-sc.org/browse/ODUHIGH-375

https://jira.o-ran-sc.org/browse/ODUHIGH-377

https://jira.o-ran-sc.org/browse/ODUHIGH-378

https://jira.o-ran-sc.org/browse/ODUHIGH-379

https://jira.o-ran-sc.org/browse/ODUHIGH-381

https://jira.o-ran-sc.org/browse/ODUHIGH-389

Status

Completed

Support for HARQ

Jira:

https://jira.o-ran-sc.org/browse/ODUHIGH-356

Status

Deferred to F-Release

Support for E2E Integration support and Validation

Jira:

https://jira.o-ran-sc.org/browse/ODUHIGH-269

Status

Deferred to E-Release maintenance

Support for Closed-Loop Automation Feature Validation

Jira:

Status

Deferred to E-Release maintenance

Dependency/Blockers:

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

HCL's update

Support for Basic RAN Slicing feature - O1 interface

Jira:

https://jira.o-ran-sc.org/browse/ODUHIGH-362

https://jira.o-ran-sc.org/browse/ODUHIGH-383

https://jira.o-ran-sc.org/browse/ODUHIGH-384

Status

Completed

e-release maintenance activity

Jira: https://jira.o-ran-sc.org/browse/ODUHIGH-395 (Optimization, scaling and rework)

https://jira.o-ran-sc.org/browse/ODUHIGH-396

https://jira.o-ran-sc.org/browse/ODUHIGH-397

https://jira.o-ran-sc.org/browse/ODUHIGH-398

https://jira.o-ran-sc.org/browse/ODUHIGH-399

https://jira.o-ran-sc.org/browse/ODUHIGH-400

https://jira.o-ran-sc.org/browse/ODUHIGH-401

Status

Completed

Dependency/Blockers:

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

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

e-release maintenance source code: https://gerrit.o-ran-sc.org/r/admin/repos/o-du/l2
Release notes: https://docs.o-ran-sc.org/projects/o-ran-sc-o-du-l2/en/latest/release-notes.html#e-release

O-DU Low

Primary Goals:

—Continue O-DU low and O-DU high pairwise test.

—FAPI P7 message integration -> Ongoing

—Continue O-DU Low and O-RU emulator test.

—Further CU plane testing -> Ongoing

—Continue E2E test with UE simulator.

—Support the UE attachment test

—Development activity for Closed Loop Automation use-case

—Support and test for cell stop and restart within O-DU High layers

E Release Feature Scope:

PTL: Luis Farias , Alternate: @Chenxi Yue

- Status

E release source code, container images and deployment instructions

TODO

Simulators (SIM)

Primary Goals:

- Support rapid prototyping by providing simulated interfaces

E Feature Scope:

- Support of O-RAN-SC E-Release Network Slicing use case by Radisys - support of O-DU projects for end-to-end closed loop use cases for RAN network slicing (implement any message flows in the O-DU Simulator, if needed)
- Align O1 Simulator with the latest specifications released by O-RAN Alliance.
- Support of NETCONF CallHome via TLS, for the O1 simulator
- VES stndDefined implementation

PTL: Alex Stancu

Jira: Count of Epics, User Stories, Tasks, and Issues: 5 issues

Status:

- Implemented Network Slicing support in the O1 Simulator:
  - support for the new O-DU hello world YANG model from OAM project
  - support for creating VES subscriptions
  - support for creating PM Jobs which send VES PM stndDefined messages
- Implemented NETCONF CallHome via TLS

E 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

Container images are described here: Simulation (E-Release)

Instructions: Network Slicing Use Case

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.

E release Feature Scope:

- Enable the 2 AIO severs with additional worker nodes deployment scenario
- Major components upgrade
- Implement the O2 interface as the MVP

PTL: Jackie Huang

E release highlights/accomplishments (17 Dec 2021):

- Implement the O2 interface as the MVP
  Description: Design and implement O2 IMS and O2 DMS for O-RAN-SC INF project
  - Realize O2 Inventory API to expose O-Cloud resource to SMO
  - Realize O2 Subscription API to allow SMO subscribe for notification of resource changes
  - Realize O2 Deployment Service API (as a PoC) to allow SMO deploy CNF in helm charts to O-Cloud.
- Major components upgrade
  Description: Upgrade the following major components and all dependent packages to align with StarlingX 5.0(more than 200 packages are upgraded):
  - Configuration Management
  - Fault Management
  - Host Management
  - Software Management
  - Service Management
  - Major dependent packages: kubernetes, helm, armada, ceph, etc.
- Enable the AIO-DX with additional worker nodes deployment scenario
  Description: The All-in-one Duplex (AIO-DX) deployment provides a pair of high availability (HA) servers with each server providing all three cloud functions (Controller, Worker, and Storage). In addition, extra worker nodes (providing only Worker function for the cloud) can be added to extend the cloud capacity.

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

Update at 17 Dec 2021

- EPICs:
  - INF-193 - Getting issue details... STATUS
    - 5 sub-tasks:
      - 2 done
      - 2 in-progress
      - 1 not started
  - INF-173 - Getting issue details... STATUS
    - 3 sub-tasks:
      - AIO-SX done:
        - INF-224 - Getting issue details... STATUS
      - AIO-DX done:
        - INF-225 - Getting issue details... STATUS
      - AIO-DX + worker done with defects:
        - INF-226 - Getting issue details... STATUS
        - INF-240 - Getting issue details... STATUS
  - INF-215 - Getting issue details... STATUS
    - 8 sub-tasks
      - INF-217 - Getting issue details... STATUS
      - INF-218 - Getting issue details... STATUS
      - INF-219 - Getting issue details... STATUS
      - INF-220 - Getting issue details... STATUS
      - INF-221 - Getting issue details... STATUS
      - INF-222 - Getting issue details... STATUS
      - INF-223 - Getting issue details... STATUS
      - INF-231 - Getting issue details... STATUS
- Issues:
  - INF-240 - Getting issue details... STATUS
  - INF-241 - Getting issue details... STATUS

Test status:

- INF platform
  - AIO-SX tested on VMs and bare metal (HP DL380p Gen8) and pass
  - AIO-DX tested on VMs and bare metal (HP DL380p Gen8) and pass
  - AIO-DX + worker tested on bare metal (HP DL380p Gen8) with defects
  - Getting Started /Sample test process - Infrastructure - Confluence (o-ran-sc.org)
- O2
  - TODO: update the test status

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 "e-release" that can be accessed using git:
  - For pti/rtp repo[gitweb]: git clone --branch e-release "https://gerrit.o-ran-sc.org/r/pti/rtp"
  - For pti/o2 repo[gitweb]: git clone --branch e-release "https://gerrit.o-ran-sc.org/r/pti/o2"
- Images for INF project
  - ISO image for INF: inf-image-aio-installer-intel-corei7-64.iso
  - Container image for o2: TODO
- Deployment instruction:
  - Installation Guide — pti-rtp master documentation (o-ran-sc.org)
  - Getting Started /Sample test process - Infrastructure - Confluence (o-ran-sc.org)

Integration and Test (INT)

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

Feature Scope:

- Automated CLM and SonarQube Scanning CI Jobs
- Improve CI for OSC projects
- Validate and and Test platform and use cases

A full integration of all OSC components are not ready yet. Within E release timeframe only limited integration tests were performed across OSC sub-modules, and around the network slicking use case.

PTL: James Li

Jira: Count of Epics, User Stories, Tasks, and Issues: 54 issues

E 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 integrate different software artifacts of existing open-source projects creating a fully functional open-source Service Management and Orchestration (SMO).

E Feature:

- SMO was extended to support network slicing. In particular, RSAC has come up with closed loop automation use case for network slicing which involves the SMO collecting PM counters related to network slicing, and based on them breaching some thresholds will cause a change in the configuration of the network slice. That means the SMO had to have support for PM counters related to network slicing, and an ability to reconfigure the O-DU for the network slice.
- Separately, SMO now implements a disaggregated VES solution that separates the collection of VES events from how it is presented to any application that wants to view them. In particular, all events now get posted on the Kafka bus, and different "connectors" are provided to make the data available in different formats. For example, the "DMaaP Connector" provides an ability for a application to read the message in DMaaP format. Alternatively, if the data needs to be synced to a dB, the "InfluxdB Connector" sinks the data to an InfluxdB.

PTL: Mahesh Jethanandani

Status:

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

Tasks:

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

18 issues

and Issues: 6 issues

E release source code, container images and deployment instructions

Docker image and instruction on how to install SMO O1 NETCONF interface can be found here.

Docker image for 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/ves"