Cherry Release (Dec 2020)

Cherry release page for the O-RAN Software community

Welcome to the Cherry release page for the O-RAN Software community

This page contains all the information specific to the Cherry release

See Requirements and Software Architecture under Committees and Projects for more details on current activities.

Second release capabilities include contributions under the following projects:

- Cherry release page for the O-RAN Software community
- Near-Real-time RIC X-APPs (RICAPP)
- Near-Real-time RAN Intelligent Controller Platform (E2 Interface) (RICPLT)
- Non-Real-time RIC (A1 Interface) (NONRTRIC)
- OAM (O1 Interface)
- 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)

Get Cherry

- **Getting Started** with Cherry to set up and run O-RAN
- **Release Date:**
  - **Source Code:**
  - **Release Tar ball:**
    - o-ran-sc-cherry-20201215-part-1.tar.gz
    - o-ran-sc-cherry-20201215-part-2.tar.gz
    - o-ran-sc-cherry-20201215-part-3.tar.gz
  - **Cherry release docker image list:** Release Image List

Maintenance:

- **Release Changelog:**
  - scp/ric-app/kpimon: Release version 1.0.1. Use SDL library.
  - nontric: Release new docker images. Fix maven project settings.
  - ric-plt/e2mgr: Update RMR version to 4.5.2.
  - ric-plt/e2: Update RMR version to 4.5.2.
  - ric-plt/alarm-go: Update RMR version to 4.5.2.
  - ric-plt/o1: Update RMR version to 4.5.2.
  - ric-plt/rtmgr: Update RMR version to 4.5.2.
  - ric-plt/submgr: Update RMR version to 4.5.2.
  - sim/a1-interface: Update cherry release documents

Documentation

- **Developer Documentation**
- **Highlighted use cases**
  - Service Management and Orchestration (SMO) [ RSA C LCM call flow ]
  - YANG-model-verification
  - CSAR package validation
  - Applications (xApps) [ RSAC Traffic Steering call flow ]
  - Traffic Steering [ MWC LA demo ]
  - Anomaly Detection
- **O-DU**
- **O-DU Low and High startup and integration**
- **Non-RT RIC**
  - A1 Enrichment Interface (A1-EI)
- **Wiki**

Cherry Release Timeline

Cherry Timeline

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, health check, life cycle management) 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).

**Cherry release highlights (12-08-20):**
- Expanded set of xApps from expanded community: Cherry release includes xApps from AT&T (TS, QP-D, QP, HW, MC), Samsung (KPIMON), and HCL (AD).
- Traffic steering use case has been extended to include data collection via E2 (using pre-spec extensions to the E2 KPM SM) by the KPIMON xApp. The KPIs used by the traffic steering use case are generated by Viavi RAN simulator.
- ML-based xApp: The AD (Anomaly Detection) xApp from HCL uses ML-based algorithm for anomaly detection. The AD xApp is loosely integrated with the TS use case in Cherry but a tighter integration is planned for Dawn.
- The demo HW (HelloWorld) xApp now has a full implementation of the HW E2 SM including subscription, indication and control as well as C++ class wrappers for all the E2 messages related to HW E2 SM.

**PTL:** Matti Hiltunen

**Jira: Count of Epics, User Stories, Tasks, and Issues:** 184 issues

**Status (9-8-20):** Integration of KPIMON in progress, new xApps under design, QP-Drive updated to use new SDK for alarms and metrics.

**Status (9-30-20):** Samsung taking over integration and testing of KPIMON; Viavi simulator producing data – an update in the data content expected this week (UE throughput, UE ID); ML-model building started based on Viavi and other simulated data (QP, AD); new xApp repos requested (Signal Storm Protection, Load Prediction); bug fixes to HelloWorld, E2 AP/SM abstraction layer underway for HelloWorld.

**Status(11-17-20):** ML-xApps (QP,AD) code committed - dealing with merge issues. End to end testing and bug fixes underway. E2AP/SM abstraction work for HelloWorld (C++) xApp completed. New repos created for HelloWorld in go and python (for Dawn).

---

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

**Mission:** Better manageability of, for example, E2 connections, the RIC platform as well as xApps via the xApp frameworks.

**Primary Goals:** Support operator-initiated health-check via E2 and be able to close all current or reject all future E2 connections. More manageability of the RIC platform and xApps, including, for example, platform statistics from E2 and A1 and more capabilities in the language-specific xApp frameworks and SDL (shared data layer).

**Cherry release highlights (2020-12-01):** We implemented the operator-initiated E2 connection health check. There is now more alignment in the level of support of various xapp framework and SDL features in python, c++ and go. For example: SDL (shared data layer) now also supports notifications in Python. The xApp framework for C++ now supports alarms, statistics and configuration management. xApps can now use SDL notifications for changes in connection states of E2 nodes. By confirming route updates before continuing we increased robustness in management of internal routes. We now also support using existing non-E2 protocols for communication with RAN. We also adapted the scaling implementation for E2 termination to the reversal of E2AP connection initiation introduced by E2AP 01.00. The E2 manager can now reject new E2 connections requests in addition to closing all E2 connections. E2 and A1 now provide platform statistics for their connections.

Detailed list of JIRA items that we worked on in Cherry (requires LinuxFoundation login): Remaining open items: link (1 items). Items that are marked as done: link (23 items). Moved out from Cherry: link (24 = 13 stretch goal items and 11 “normal items”).

**PTL:** Thoralf Czichy
Non-Real-time RIC (A1 Interface) (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 / SMO applications) to perform Non-RT-RIC tasks.
- Non-RT-RIC will also host the new R1 interface (between rApps and SMO services)
Cherry Feature Scope:

- **A1 Policies:**
  - A1 Policy Management Service (hosted in ONAP CCSDK)
  - A1 Policy Controller Adapter (hosted in ONAP CCSDK)
  - A1 Simulator / test stub (SIM project)
  - A1 Policy Control-Panel (PORTAL project)
  - Support A1-AP v2.0.
- **A1 Enrichment Information**
  - Initial version of A1 EI Job coordination function/service
- **Deployment/Integration**
  - Docker-based & OSC Kubernetes deployment & ONAP OOM Kubernetes deployment (Overlaps with SMO project)
- **rApps**
  - Very simple hello-world rApp
  - TrafficSteering usecase
  - Healthcheck usecase
  - rApp support
    - Initial rApp catalogue/inventory
      - overlaps with "common App" usecase, & SMO project
  - Requirement gathering & initial test/integration with ML supporting functions in NONTRRIC/SMO
  - Requirement gathering & initial test/integration with data collection & coordination functions in NONTRRIC/SMO
  - Requirement gathering & initial test/integration with CMDB (if progressed in ONAP), Topology & Inventory functions in SMO

**Update - 06/October/2020**

- Prototyping of the A1 enrichment Information coordination service is in progress.
- Initial prototype of rApp registration / catalogue function, Very simple hello-world rApp.
- Ongoing work to support O-RAN WG2 A1-P spec.
- A1 Policy Management Service and A1 Policy Controller Adapter hosted in ONAP CCSDK are getting ready for ONAP Guilin Milestone RC0.
- Other ongoing tasks: documentation, automated integration testing, CSIT, Run benchmark test in a cloud instance, Function Test environment refactoring, etc.

**Update - 04/November/2020**

- Initial rApp catalogue is mostly ready for first tests.
- A1 Enrichment Information job coordination function ready.
- Ongoing work on A1 EI job control in NONTRRIC Control Panel.
- A1-AP v2.0 (A1 Policy) is ready for implementation (Internal implementation already completed, now ready for release to the community)
- Integration of ONAP A1 Controller functions
- Extensions & evolution for NONTRRIC Control Panel & A1 SIM (version update, & general improvements)
- A1 Policy Management Benchmark completed ... more info to follow
- Other ongoing tasks: documentation, automated integration testing, CSIT, Function Test environment refactoring, etc.

**Update - 08/December/2020**

- A1 Enrichment Information (EI) Coordination function ready for release
- Updated A1 EI job control in NONTRRIC Control Panel ready for release
- Updated A1 EI simulator integrated in A1 SIM
- Initial rApp catalogue ready for release
- ONAP A1 Controller functions integrated
- General extensions & evolution for NONTRRIC functions (version update, & general improvements)
- HTTP Proxy support added for A1 Interface (Policy)
- Other ongoing project maintenance tasks: documentation, automated integration testing, Function Test environment refactoring, etc.
- Demo available at: 2020-12-04 - OSC NONTRRIC Cherry Demo

PTL: John Keeney
Jira: Count of Epics (41 issues), User Stories, Tasks, and Issues: 732 issues

OAM (O1 Interface)

Primary Goals:
- update OAM projects for latest O-RAN Specifications
  - O-RAN Operations and Maintenance Architecture Version 3.0 - April 2020
  - O-RAN Operations and Maintenance Interface Version 3.0 - April 2020
  - O-RAN Use Cases Detailed Specification 2.0 - April 2020
  - Draft O1 yang models implemented and tested against the SIM implementation - not published in LNF repos
- support of the application LCM use case
- Discussion about the details together with the SMO project
- handover SMO artifacts to new SMO project

Cherry Feature Scope:
- Switch to Java11
- Switch to OpenDaylight version Sodium (O1 termination NetConf)
- https only support for VES-Collector (O1 termination VES)
- full IPv6 support

Please see OAM Cherry page for further details

PTL: Martin Skorupski
Status:

Jira: Count of Epics (21 issues), User Stories, Tasks, and Issues: 257 issues

Demo of OAM use cases

### O-RAN Central Unit (OCU)

**Primary Goals:**
- Source code includes RRC, Ng, E1, F1. Platform and OM are provided in dependent libraries.
- O-CU-CP is integrated with O-CU-UP. O-CU functions should be complete.

**Cherry Feature Scope:**

**RRC:**
- support Broadcast of system information
- support RRC connection control

**NG:**
- support PDU Session Management Procedures
- support UE Context Management Procedures
- support Transport of NAS Messages Procedures
- support Interface Management Procedures

**E1:**
- support Interface Management procedures
- support Bearer Context Management procedures

**F1:**
- support Interface Management procedures
- support UE Context Management procedures
- support RRC Message Transfer procedures
- support System Information Procedures

**PTL:** Yingying Wang
Jira: Count of Epics, User Stories, Tasks, and Issues:

<table>
<thead>
<tr>
<th>Epics</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>OCU-1</td>
<td>Getting issue details...</td>
</tr>
<tr>
<td>OCU-2</td>
<td>Getting issue details...</td>
</tr>
<tr>
<td>OCU-3</td>
<td>Getting issue details...</td>
</tr>
<tr>
<td>OCU-4</td>
<td>Getting issue details...</td>
</tr>
<tr>
<td>OCU-5</td>
<td>Getting issue details...</td>
</tr>
<tr>
<td>OCU-6</td>
<td>Getting issue details...</td>
</tr>
</tbody>
</table>

O-DU High

Primary Goals:
- Develop L2 layers to achieve UE attach with UL and DL data on FDD, mu=0, BW=20 MHz
- Interface with O-CU on F1AP
- Interface with near RT RIC on E2AP
- Interface with O-DU Low using FAPI
- Interface with OAM on O1

Cherry Feature Scope:
- Implement UE attach procedure with basic scheduling on FDD, mu=0, BW=20 MHz
- Implement single UE DL and UL data path and bench-marking
- Add support for 64QAM modulation scheme in DL and 16QAM in UL
- Add support for all short PRACH formats
- Integrate O-DU High with O-DU Low
- Integrate with Viavi sim/O-CU
- Explore O1 interface
- Establish Netconf session for O1 interface for CM
- Support Health Check use-case
Documentation and release related activities for Cherry release have been completed.

Jira: EPICS Status below:

- https://jira.o-ran-sc.org/browse/ODUHIGH-10 - Done
- https://jira.o-ran-sc.org/browse/ODUHIGH-188 - Done
  - As an O-DU L2 developer, I want to implement UE attach procedure with basic scheduling
- https://jira.o-ran-sc.org/browse/ODUHIGH-191 - Done
  - As an O-DU L2 developer, I want to explore O1 interface
    - Made certain exploration and begun work on CM and health check use-case
- https://jira.o-ran-sc.org/browse/ODUHIGH-196 - WIP
  - As an O-DU L2 developer, I want to support Health Check use-case
    - get-alarm list to be supported i.e., Health Status Retrieval
    - Code merged into master branch
- https://jira.o-ran-sc.org/browse/ODUHIGH-214 - Done
  - As an O-DU L2 developer, I want to integrate O-DU High with O-DU Low
    - O-DU High successfully integrated with O-DU Low in timer mode
    - O-DU High completed aligning with latest FAPI files from Intel for Radio mode
    - Radio mode testing to be begin once O-RU integration is complete
- https://jira.o-ran-sc.org/browse/ODUHIGH-184 - WIP
  - As an O-DU L2 developer, I want to implement single UE DL data path and bench-marking
    - Design in progress
- https://jira.o-ran-sc.org/browse/ODUHIGH-185 - WIP
  - As an O-DU L2 developer, I want to implement single UE UL data path and bench-marking
    - Design in progress
    - PUCCH code changes in progress
- https://jira.o-ran-sc.org/browse/ODUHIGH-186 - WIP
  - As an O-DU L2 developer, I want to add support for 64QAM modulation scheme in DL
    - Code under review for signaling
- https://jira.o-ran-sc.org/browse/ODUHIGH-187 - TODO
  - As an O-DU L2 developer, I want to add support for 16QAM modulation scheme in UL
    - Code under review for signaling
- https://jira.o-ran-sc.org/browse/ODUHIGH-190 - WIP
  - As an O-DU L2 developer, I want to integrate O-DU High with Viavi softwares
    - integration plan discussion begun.

Dependency/Blockers:

- Custom Yang files will be used for Dev activity.
- FAPI files being used provided by INTEL, which is not completely in-line with the latest released version from SCF.

---

**O-DU Low**

Primary Goals:

- Integrate with O-DU High with FAPI interface with cherry release aligned IOT profile
- Integrate with O-DU emulator from Viavi with cherry release aligned IOT profile
- Support E2E integration with O-CU, O-DU High, O-RU emulator and UE for UE attachment

Cherry Release Feature Scope:

- O-DU Low and O-DU High integration according to RSAC and INT project alignment features and scope
- O-DU Low and O-RU/RRU emulator integration according to RSAC and INT project alignment features and scope
- E2E integration according to RSAC and INT project alignment features and scope
- O-DU Low integrated with thirty party commercial SW to verify the UE attachment and traffic, update the O-DU Low version accordingly

PTL: @Zhimin Yuan
• Status
  • O-DU Low integrated with thirty party commercial SW to verify the UE attachment and traffic, update the O-DU Low version accordingly – Done

**ODULOW-14** - Getting issue details...  STATUS

- successfully do the UE attachment and traffic

**ODULOW-11** - Getting issue details...  STATUS

• O-DU Low and O-DU High integration – in progress
  • OSC Lab environment is installing OSC INF, O-DU Low can build/run in OSC INF
  • O-DU Low and O-DU High P5 massage integration using O-CU stub and O-DU low time mode – done
  • O-DU Low and O-DU High further P7 integration - not start

**ODULOW-12** - Getting issue details...  STATUS

• O-DU Low and O-RU/RRU emulator integration - in progress
  • according to RSAC and INT project alignment features and scope – Done
  • integration the S-plane, O-DU low and O-RU emulator get synchronized – in progress
  • C-plane test, O-RU emulator can parse the C-plane message correctly – not start
  • U-plane test, pass SSB, PDSCH, PUSCH, PRACH channel data exchange – not start

**ODULOW-13** - Getting issue details...  STATUS

• E2E integration – not start
  • support E2E integration with O-DU High, O-CU, O-RU emulator and UE
  • align with RSAC and INT project alignment features and scope

---

**Simulators (SIM)**

Primary Goals:
- Support rapid prototyping by providing simulated interfaces

Cherry Feature Scope:
- O1 Simulator enhancements
  • Upgrade NETCONF Server framework
- E2 Simulator enhancements
  • Implement more message types
  • Assess if E2 Simulator can be used for benchmarking
  • Maintain alignment with latest YANG models

PTL: Alex Stancu
Jira: Count of Epics, User Stories, Tasks, and Issues: 8 issues

Status (07 Oct. 2020):

- E2 Simulator - support for new E2 Messages - done
- E2 Simulator - benchmarking of RIC - in progress
- O1 Simulator - upgrade of NETCONF Server - in progress
- OTF integration - TBD

Infrastructure (INF)

Primary Goals:

- 2 servers. 2 AIO servers with HA (high availability), the controller functionality and storage functionality will be deployed at the 2 servers with standby-active mode managed by "service management". If one server or one service in one server has error, it will be switched from active to standby one to maintain the service availability.
- 2 AIO servers with additional worker node.

Cherry Feature Scope: TODO

PTL: Xiaohua Zhang
The INF project status update at INF project status for 26-Aug-2020:
- The first major goal of "enable the duplex deployment with HA features" has been done. No block issue in JIRA although there are some issues left.

Status update 24-Sept-2020:
- The AIO deployment scenario has been supported. Although there are some issues/bugs, but it won't affect the functionalities at all.
  - For the AIO with additional worker nodes, the codes had been checked into the repo, the verification and test is still ongoing. (it will be delayed to Dawn release).
  - Continue to support the integration test.
  - Continue to prepare the document of Cherry release.

For the AIO with additional worker nodes, the codes had been checked into the repo, the verification and test is still ongoing. (it will be delayed to Dawn release).

Status update 28-Oct-2020:
- The AIO deployment scenario has been supported. Although there are some issues/bugs, but it won't affect the functionalities at all.
  - For the AIO with additional worker nodes, the codes had been checked into the repo, the verification and test is still ongoing. (it will be delayed to Dawn release).
  - Continue to support the integration test.
  - Continue to prepare the document of Cherry release.

Status update 11-Nov-2020:
- Already add the packages which were missing during the ODU-High/Low integration.
- Some other issues will be addressed before 14-Nov-2020
- In general, ready to cut-off by 14-Nov-2020

Status update 09-Dec-2020:
- Final image has been generated.
- Finish the integrate test with O-DU Low/High
- Document has been updated into repo for release note and others.

Integration and Test (INT)

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

Cherry Feature Scope:
- Automated CLM and SonarQube Scanning CI Jobs
- Improve CI for OSC projects
- Validate and test platform and use cases
Jira: Count of Epics, User Stories, Tasks, and Issues: 54 issues

Status (28 Oct. 2020):
- CLM scanning jobs:
  - done: A1 and python xApp-framework
  - Pending: Rest of the projects
- SonarQube scanning jobs:
  - 21 projects passed
  - 7 project failed
- OSC Lab internal CI pipeline:
  - Initiate works with O-DU-low CI
- OSC Lab Testing and Validation
  - SMO Yang model tests completed.
  - VM created for TS use case validation.

**Documentation (DOC)**

Primary Goals: TODO

Bronze Feature Scope: TODO

PTL: weichen ni

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

**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).

Cherry Feature Scope: SMO entered the Cherry release in the middle of third sprint of code development. As such its scope is fairly modest. They are validation of application packages. assuming that we can agree on the format of the package, on boarding of applications and storing them in a package catalog which also has to be agreed upon, and as a stretch goal, setting up an environment where YANG modules that will be used by O-RAN, whether they are from 3GPP, and O-RAN itself can be used by vendors developing RIC, CU, DU and the RU to test a MVP configuration.

PTL: Mahesh Jethanandani
Status: A proposal was made on application package format, and there was some agreement on it following the ETSI SOL 004 specification. The contributions into the SMO project currently validate that part of the agreement, and allow for Network Function, xApp and rApp vendors to validate their package using the tools developed in SMO.

David Kinsey is driving package catalog requirements, which is LCM Step 3. But it was determined that package catalog can be implementation specific, and therefore SMO cannot validate any particular catalog. As such LCM Step 3 will be skipped.

The second part of SMO was the setup of a framework for testing of YANG models that are going to be used by SMO and the Network Functions that constitute the O-RAN solution. Thanks Martin Skorupski, Zhe Huang, and Alex in setting up that framework in the OSC lab.

For a detailed workflow and end-to-end test manual of the two parts, refer to Cherry Release Test Plans for SMO.

Jira: Count of Epics (3 issues), User Stories, Tasks, and Issues: 54 issues