

Bronze Release (Jun 2020)



Bronze release page for the O-RAN Software community

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

This page contains all the information specific to the Bronze release and frozen code (code freeze = Jun-14, 2020) and documentation. The main trunk will continue on to support future releases.

This is the second release and is in partnership with the O-RAN Alliance. The specification and software are being worked at the same time and the software is considered pre-specification software.

The projects have limited capabilities which will increase over future releases.

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

Second release capabilities include contributions under the following projects:

- [Bronze release page for the O-RAN Software community](#)
- [Near real-time RIC X-APPs \(RICAPP\)](#)
- [Near Realtime RAN Intelligent Controller Platform \(E2 Interface\) \(RIC-PLT\)](#)
- [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 \(DOC\)](#)

Get Bronze

- Source code:
 - Release Date: [o-ran-sc-bronze-20200621.tar.gz](#)
 - Maintenance: [o-ran-sc-bronze-20200810.tar.gz](#)
 - Updates:
 - doc: made fixes for generating bronze branch documentation;
 - it/dep: added scripts used in "Getting Started" demonstrations;
 - nonrtrc: added support for configuration of certificates;
 - pti/rtp: skipping certain types of image generation;
 - ric-plt/xapp-frame-py: documentation touch-up.
- Per-repository details: [Releasing Bronze Tasks](#)

Documentation

- [Developer Documentation](#)
- [Wiki](#)
- [Getting Started](#) with Bronze to set up and run O-RAN
- Bronze [xAPP SDK](#) for developing xApps

Bronze Release Timeline

[Bronze Timeline](#)

Learn Bronze

- Documentation home: <http://docs.o-ran-sc.org>
- [Getting Started](#) to set up and run O-RAN
- [xAPP SDK](#) for developing xApps

Near real-time RIC X-APPs (RICAPP)

Primary Goals: This includes open source sample xApps and platform applications that can be used for integration, testing, and demonstrations.

Bronze Release Highlights (6-3-2020):

- An architecture of co-operating xApps that use data collection, analysis, machine learning, and E2 interface operations to implement a UE-level control loop for traffic steering, based on A1 Policy Intents. The Bronze release includes initial versions of 3 of the 4 xApps in the traffic steering use case.
- A demo xApp (HelloWorld - HW) that demonstrates how to write xApps in C++ including messaging, shared data layer, and handling A1 (health check) and E2 messages, including an example E2SM.

PTL: [Matti Hiltunen](#)

Jira: Count of Epics, User Stories, Tasks, and Issues: [234 issues](#)

Status 05-12-2020:

- HW (HelloWorld) xApp is ready except for documentation
- MC (Measurement Campaign) xApp is ready except for documentation
- TS (Traffic Steering) xApp is ready for integration testing
- QP-driver (QuE Prediction Driver) xApp is ready for integration testing
- KPIMON is waiting for legal review at Samsung
- QP (QoE Prediction) xApp status pending (5-12-2020)
- Full integration of the traffic steering xApps (TS, QP, QP-driver, KPIMON) likely in Maintenance release.

Status 3-31-2020:

- Repo issues for new xApps have been resolved
- Detailed design of message schemas for traffic steering use case completed and code commits started for use case xApps
- Health check use case features being implemented in xApps (inc. A1 health check)
- Demo xApp (HelloWorld) making steady progress - xApp demonstrates how to implement typical xApp functionality using C++
- Measurement Campaign (MC) xApp making steady progress
- Blockers: CU/DU simulator/emulator uncertainty (affects KPIMON xApp)

Near Realtime RAN Intelligent Controller Platform (E2 Interface) (RIC-PLT)

Primary Goals: Develop platform for xApp hosting with stable APIs for xApps using RIC's E2 interface, xApp management interface and other auxiliary xApp interfaces, like a database storage interface.

Bronze release highlights (2020-06-03): The E2AP protocol implementation of the near-RT RIC platform has been adapted from one based on pre-specification assumptions to the E2AP protocol as defined by the O-RAN alliance. The new near-RT RIC O1 mediator implements the first O1 interfaces based on netconf and Yang models, e.g., to check the health status of xApps. The near-RT RIC internal messaging and message routing solution has been rewritten to significantly improve throughput, latency, and to behave more robustly in failure and high-load conditions. The Redis-based shared data layer now supports HA (high-available) deployments. Various Near-RT RIC platform components use it to store their state and recover that state in failure, restart and upgrade situations. xApps can use the same shared data HA later solution for their state. xApps that use E2AP subscriptions can now use the new E2AP subscription API instead of having to directly generate ASN.1 encoded E2AP subscription messages. In addition to a Go version, the Bronze release of the near-RT RIC platform now also provides a C++ and python framework to simplify the implementation of xApps.

Original Bronze Feature Scope at start of release: Adapting RIC from pre-spec to WG3's first formal E2 protocol specification (for communication with RAN nodes, like CU/DU). Implement netconf-based O1. Enhance RIC's messaging solution (RMR) to achieve higher throughput, various health check capabilities including a capability to ping E2 nodes via E2 and to do a health check of the A1 interface, better failure handling in various components, ...

PTL: [Thoralf Czichy](#)

Status 2020-04-29 and 2020-04-30 and 2020-05-06 and 2020-05-12

- Cherry-R5 plans: [link](#) and epics only [link](#)
- Epics that are done: [link](#) (36 as of 2020-05-12), spilling over into Cherry-R5: **13 items**. Using filter "[movedoutofbronzeR4](#)" you can see items fully moved from Bronze to Cherry-R5: 9 items. Items moved from Bronze to more distant future: 8
- Still open items:
 - RIC-56 (framework for handling alarms ready, but actual first alarms under this item not ready (RIC-202, RIC-203, RIC-204) - specific alarm work continues in Cherry-R5
 - DB related work: RIC-246 (UE NIB reader API documentation will continue in Cherry. Additionally we have RIC-371 for R5 implementation and documentation work) // RIC-105, RIC-110 moved out bronze to R5 // RIC-101 moved out of bronze to remote future // RIC-106 (HA integration DBAAS = leftover from R3) is done
 - RIC-291 is done on E2T side. E2M does not use ASN.1 anymore. Subscription manager part of RIC-291 also done. RIC-60 almost complete. Some minor work in Cherry (RIC-306) and in Bronze maintenance (RIC-306), RIC-83 was started in Bronze, but plan is to continue in Cherry-R5 (with US RIC-208). For RIC-124 we had some O1 work, but the E2M side work for RIC-124 is not yet done (so we continue in Cherry)
 - RIC-95: A still working on this cannot be tested due to lack of simulator. Likely to be ready only in beginning of Cherry. I now mark this as spilling into Cherry-R5.
 - O1-related: RIC-59 moved to future (post Cherry-R5 as no netconf models yet, work not started yet), RIC-139 (not yet implemented. Nune checked that E2T and E2M are only components that do not have health check yet. Added US for them under this item. Full item moved to Cherry)
 - RIC-173, RIC-174 (E2 specification preparation work for Future work): specifications still being worked on, but blocked by 3GPP license issues. We add these items back into actual planned work, once the 3GPP issues are resolved
 - RIC-136 (moved to Cherry-R5 as no permission from O-RAN yet)
 - RIC-internal E2E work : RIC-57 (documentation) will continue in Cherry-R5 and RIC-149 (test cases) we will mark as done at the end of Bronze.
 - RIC-217 only documentation left (target: late commit/work in Bronze-R4)
 - RIC-80 actually done, but commit problems. Plan is to merge this still into Bronze-R4, but for now I mark it to Cherry-R5 so we don't forget it. We mark RIC-80 as done once we can commit it. We have created RIC-375 that includes removal of the old ASN.1 based interface support and which might include minor cleanup work in this context.
 - RIC-84 implementation ready, but came to late for Bronze-R4. We merge in the beginning of Cherry-R5. Marked as to be completed in Cherry-R5.
 - Moved out from Bronze-R4: RIC-177, RIC-139, RIC-136, RIC-123, RIC-107, RIC-118 and RIC-119 as well as user story: RIC-104, RIC-203. Use this filter for label "[movedoutofbronzeR4](#)"
 - RIC-120 Antti working on this in his spare time. We continue the work in Cherry-R5
- Healthcheck use case supporting items that are **done**: A1 healthcheck (RIC-46), xApp health state as O1 (RIC-14). **Partially done**, but not yet easy to show-case: RIC-124 (E2M not yet publishing SDL notifications), RIC-56 (alarm path available, but no actual alarms) **Not done**, but either coming late or in early Cherry-R5: RIC-95 (REST interface for checking the E2 connection states (lack of simulator)). RIC-139 (platform health via O1)

Status 2020-03-31

- Jira: 63 epics: [link](#) (requires free LinuxFoundation account). Items that we worked on already before the first formal development are marked as Bronze-R3 in JIRA "fix release" (11 = 2 (in progress) + 9 (done) // 2 moved to R4) and the ones to be worked on during the five three-week sprints in Bronze-R4 Feb-3 to May-17 (52 = 6 (in progress) + 8 (done or done, but pending review by PO) + 38 (not yet started)). For useful near-RT RIC JIRA links check §8 in [Jira usage conventions](#).
- 2 new epics: one related RIC-291 to (replacing RIC-136 due to legal issue around copying ASN.1 also mentioned in last status update) and RIC-246
- Question to TOC: progress on ASN.1 legal issues.
- [Demos of sprint D3](#) planned for Thursday, Apr-2, 9am ET (see groups.io calendar)
- Deployment time decision between DBAAS SA (standalone) and DBAAS HA (high available) is transparent to RIC components as expectation is that DBAAS "client configmap" is included in RIC component help charts.
- Switch to new E2AP still pending integration (estimate Apr-3). Multiple components are ready for it.

Status 2020-03-04

- Jira: 61 epics: [link](#) (requires free LinuxFoundation account). Items that we worked on already before the first formal development are marked as Bronze-R3 in JIRA "fix release" (13 = 1 (in progress) + 8 (done)) and the ones to be worked on during the five three-week sprints in Bronze-R4 Feb-3 to May-17 (48 = 7 (in progress) + 1 (done) + 40 (not yet started)). For useful near-RT RIC JIRA links check §8 in [Jira usage conventions](#).
- Legal aspects related to automatically using WG3 ASN.1 of E2APv1.0 still open. Thoralf to send information to Jinri by Friday this week.
- Samsung's Arun started working RIC-203 and RIC-95
- Demos of sprint D2 (Feb-23 - Mar-12) planned for Thursday, Mar-12, 9am ET (note special summer time adjustments in this week in other locations). Thoralf will add event to groups.io calendar. After this demo every three weeks some date & time.

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.

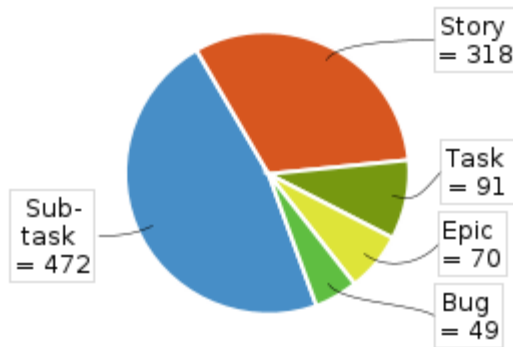
It can also perform intelligent radio resource management function in non-real-time interval (i.e., greater than 1 second).

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.

Bronze Feature Scope:
Initial design and start implementation of NONRTRIC Platform.
Continue evolution of A1 interface.
Continue evolution of A1 Policy Management functions.
Modeling and implementation of the Non-RealTime RIC aspects of the "Health Check" and "Traffic Steering" use cases

PTL: [John Keeney](#)

Jira: Count of Epics ([70 issues](#)), User Stories, Tasks, and Issues: [1000 issues](#)



OAM (O1 Interface)

Primary Goals: Implementation of O-RAN Alliance OAM Architecture and Interface Specifications.

Bronze Feature Scope: Modeling and implementation of the [Health Check](#) use case and Deployment of an [SMO](#).

PTL: [Martin Skorupski](#)

Status: 08 Apr 2020

- Please log in into Jira: <https://jira.o-ran-sc.org/secure/RapidBoard.jspa?rapidView=8&projectKey=OAM&view=reporting&chart=sprintRetrospective&sprint=25>
- ongoing work on SMO Distribution and Documentation
- Heathcheck use case can be demonstarted any time with real equipment in OWL
- First ideas for a Cherry Use Case: SoftwareManagement: Upgrade from xRan-fronthaul to o-ran-fronthaul to o-ran-o1-ru
- Close interworking with 3GPP yang editors via O-RAN Working Group 1 Modeling team - preparation of LS and CR

O-RAN Central Unit (OCU)

Primary Goals:

The O-CU project provides 5G SA reference implementation according to the O-RAN WG8 documents. In addition we provide the implementation of RRC SDAP PDCP and F1AP module and API between them.

Basic CU functions defined in O-RAN WG8

Support E2E use case defined in O-RAN

Bronze Feature Scope:

F1

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

SDAP

- support transfer of user plane data;
- support mapping between a QoS flow and a DRB for both DL and UL;
- support marking QoS flow ID in both DL and UL packets;
- support reflective QoS flow to DRB mapping for the UL SDAP data PDUs.

PDCP

- support transfer of data (user plane or control plane);
- support maintenance of PDCP SNs;
- support header compression and decompression using the ROHC protocol;
- support ciphering and deciphering;
- support integrity protection and integrity verification;
- support timer based SDU discard;
- support reordering and in-order delivery;
- support out-of-order delivery;

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

PTL: [user-a934b](#)

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

[OCU-1](#) - Getting issue details... [STATUS](#) F1

[OCU-2](#) - Getting issue details... [STATUS](#) SDAP

[OCU-3](#) - Getting issue details... [STATUS](#) PDCP

[OCU-4](#) - Getting issue details... [STATUS](#) RRC

[OCU-5](#) - Getting issue details... [STATUS](#) Ng

[OCU-6](#) - Getting issue details... [STATUS](#) E1

O-DU High

Primary Goals: Develop Amber release code to enhance O-DU layers source code

Bronze Feature Scope:

- O-DU High layers (MAC, RLC and app):
 - Re-align seed code to 3GPP Release 15.3.0
 - Align seed code to WG8 AAD specification and interfaces
 - Implementation of cell broadcast procedure and UE attach procedure (SA mode) for FDD mode and FR1 (Numerology = 0, Bandwidth = 20 MHz) and basic scheduler APIs for single UE and single HARQ transmission
- F1-U interface development
- F1-C interface enhancement:
 - Support for following additional F1AP messages:
 - Initial UL RRC Message Transfer
 - UL/DL RRC Message Transfer
 - UE Context Setup Request/Response
 - Enhance F1AP messages:
 - F1AP Setup Request/Response
 - GNB DU Config Update
- Basic FAPI messages Implementation

Out of Scope:

- Use cases – Traffic Steering, Health Check related messages and call flows
- TDD functionality, NSA
- End to end testing

Limitations/Dependencies:

- FAPI Implementation – Dependency on O-DU Low to open source WLS files, interface files
- Testing: Currently only some unit testing can be done due to lack test infrastructure i.e UE or UE simulator, O-RU, O-CU and core components.

PTL: [Sachin Srivastava](#)

Jira: EPICS are mentioned below:

- <https://jira.o-ran-sc.org/browse/ODUHIGH-1> - Done
 - As an O-DU L2 developer, I want to create F1AP module based on F1 interface APIs and develop them as per the architecture defined
- <https://jira.o-ran-sc.org/browse/ODUHIGH-8> - Done
 - As an O-DU L2 developer, I want to implement FAPI interface towards O-DU low
- <https://jira.o-ran-sc.org/browse/ODUHIGH-9> - Done
 - As an O-DU L2 developer, I want to implement cell broadcast procedure at MAC Layer
- <https://jira.o-ran-sc.org/browse/ODUHIGH-10> - WIP
 - As an O-DU L2 developer, I want to implement UE attach procedure with basic scheduling
- <https://jira.o-ran-sc.org/browse/ODUHIGH-11> - Done
 - As an O-DU L2 developer, I want to implement E2 interface
- <https://jira.o-ran-sc.org/browse/ODUHIGH-27> - Done
 - As an O-DU L2 developer, I want to develop RLC layer interfaces

O-DU Low

Bronze Release highlights(Jun.8):

The Bronze Release includes:

- ORAN WG8 Software Specification compliant DU Low implementation including O-RAN FrontHaul compliant Radio<-> L1 interface, FAPI compliant L1<->L2 interfaces, and a shared memory and buffer management library for efficient L1<->L2 communication
- Ability to link in a high-performance L1 stack application with advanced 5G NR features including 3GPP TS 38.211, 212, 213, 214 and 215, running on Intel Xeon processor based O-DU hardware, and packaged with a comprehensive functional and performance evaluation framework.

Primary Goals: The O-RAN-SC-ODU LOW project provides reference implementation of ODU-LOW according to the O-RAN AAD specification (WG8) documents. The reference design will follow the open interface toward ODU HIGH, RRU and Accelerator, provide physical layer signal processing functionality according to ORAN specification and 3GPP .

Bronze Feature Scope: enhance O-RAN FH interface lib, develop FAPI API, deliver O-DU Low full functionality binary

PTL: @Zhimin Yuan

Jira: 3 Epics, 4 Stories, Tasks, and Issues:

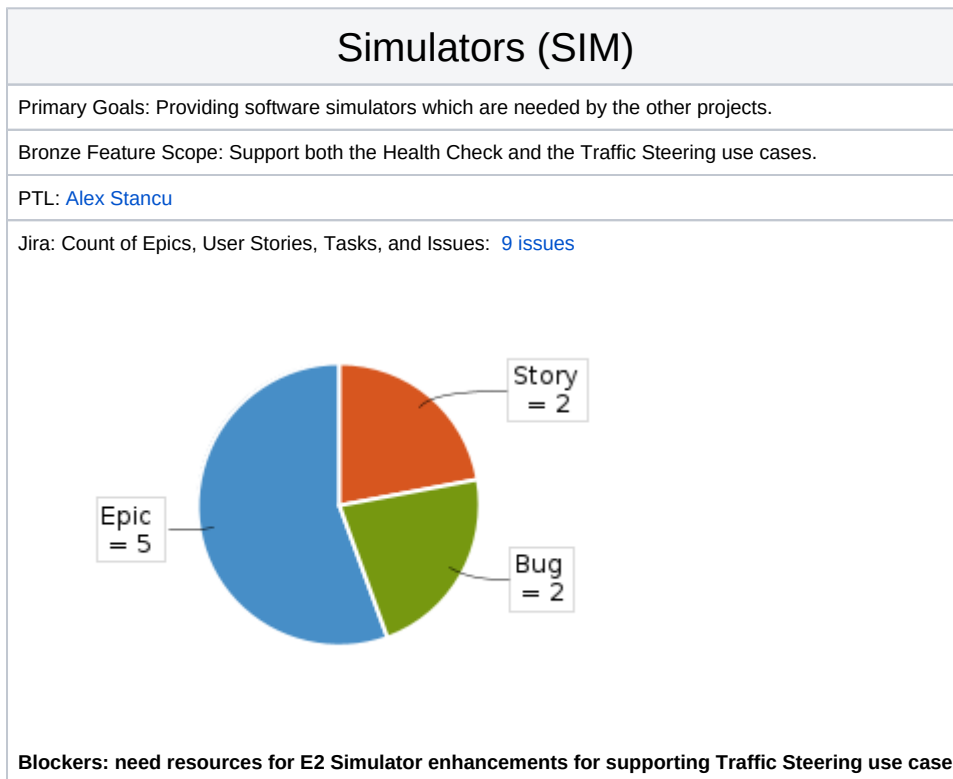
Epics	Planning	Jira Epic Link	status
-------	----------	----------------	--------

Develop L1/L2 FAPI Interface API	1.Develop the L1/L2 FAPI API files 2.Develop the interface parse software for O-DU low software	ODULOW-2 - Getting issue details... STATUS	Do ne
Develop O-DU Low binary	1.Upstream the O-DU low binary file with typical test cases 2.Provide loader for OSC to download and test	ODULOW-6 - Getting issue details... STATUS	Do ne
Enhance O-RAN Front Haul Library	1.Develop new features, include support category B, Support block floating point compression and decompression 2.Develop the CI/CD infrastructure according to O-RAN requirement	ODULOW-7 - Getting issue details... STATUS	Do ne
O-DU interface with accelerator	Depend on the specification definition(TBD)	track in DPDK	

Limitations/Dependencies:

- L1 binary license agreement – affect the L1 binary release
- ICC usage agreement – it'll affect FAPI lib, O-RAN FH lib compilation and execution. it'll affect L1 binary execution

Status update:



Infrastructure (INF)
Primary Goals: Provide an open source refence implementation of O-Cloud complying with WG6 specifications. Use it as the edge cloud to be able to run vO-DU and vO-CU services on it.

Bronze Feature Scope: Based on release Amber, support the ARM based system. Align with requirement of the O-Cloud, add more services such as fault management , configuraiton management , software management , host management and service management. Improve the automation by adopting the ansible, and integrate with INT project.

Bronze Releae Highlights:

- Implement the container based O-Cloud reference design, and with real time features which is critical for the performance sensitive application such as DU.
- Integrate the telco grade features include fault management, software management, service management, host management and configuration management which leverage from StarlingX.
- Support the ARM based real time Kubernetes cluster.

PTL: [Xiaohua Zhang](#)

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

[0 issues](#)

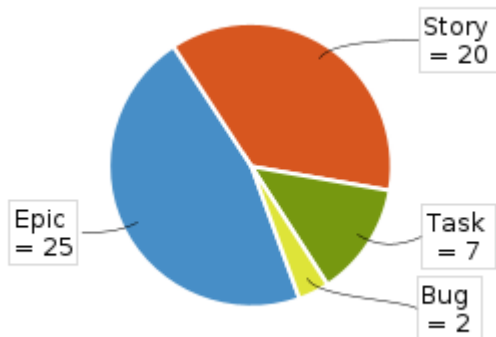
Integration and Test (INT)

Primary Goals: CICD, integration and deployment of O-RAN SC developed software, software testing.

Bronze Feature Scope: 1. CICD to support software static analysis and quality reporting, and integrated testing for Near RT RIC; 2. "One-click" deployment of Non RT RIC, OAM, and Near RT RIC, and simulators for supporting the Bronze use cases; 3. Deployment and integration of other components; 4. Integrated testing and use case flow testing.

PTL: [user-d3360](#)

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



Documentation (DOC)

Primary Goals: Bronze release documentation for OSC and developers

Bronze Feature Scope:
Improve current documentation skeleton and content, make documentation more understandable for readers.

Focusing on integration documentation, the new feature of Bronze release.

PTL: [weichen ni](#)

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