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:

- Non-Real-time RIC (A1 Interface) (NONRTRIC)
- Near Realtime RAN Intelligent Controller Platform (E2 Interface) (RIC-PLT)
- Near Real-time RIC X-Apps (RICAPP)
- OAM (O1 Interface)
- O-RAN Central Unit (OCU)
- O-DU High
- O-DU Low
- Dashboard (OAM)
- Simulators (SIM)
- Infrastructure (INF)
- Integration and Test (INT)
- Documentation (DOC)

Get Bronze

- Source code: Release Date: 14 Jun 2020
- Per-repository details: Release Date: 14 Jun 2020

Documentation

- Developer Documentation
- Wiki

Table of Projects

Bronze Release Timeline

Bronze Timeline

Learn Bronze

- Documentation home: http://docs.o-ran-sc.org
- Video tutorials and demos:

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 Feature Scope: Support Bronze use cases by developing and evolving existing xApps. For the traffic steering use case, we are extending the existing KPIMON xApp and developing new TS, QP, and QP Driver xApps using the different programming languages supported by the RIC platform. For the health check use case we are developing a demo xApp that supports the A1 health check and extending existing xApps with more thorough liveness checking from K8s and new statistics/alarming APIs provided by the RIC platform.

PTL: Matti Hiltunen

Jira: Count of Epics, User Stories, Tasks, and Issues: 100 issues

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 Feature Scope: 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

- Epics that are done: link (35 as of 2020-04-29)
- Still open items:
  - Still under clarification: RIC-56 (framework for handling alarms ready, but actual first alarms under this item not ready (RIC-202, RIC-203, RIC-204)
  - Still under clarification: DB related work to be checked: RIC-105, RIC-110, RIC-246, RIC-106, RIC-101
  - Still under clarification: E2 work to be checked: RIC-291 (use precompiled ASN.1) is probably already done. RIC-60 almost complete. Some minor work in Cherry, RIC-83 was started in Bronze, but plan is to continue in Cherry-R5. 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)
  - Still under clarification: RIC-95: A still working on this cannot be tested due to lack of simulator. Likely to be ready only in beginning of Cherry.
  - Still under clarification: O1-related: RIC-59 moved to Cherry-R5 (no netconf models, work not started yet), RIC-139 (not yet implemented. Moved to Cherry)
  - Still under clarification: RIC-173, RIC-174 (E2 specification preparation work for Cherry-R6): specifications still being worked on. Check from A.
  - Clear: RIC-136 (moved to Cherry-R5 as no permission from O-RAN yet)
  - Still under clarification: RIC-internal E2 work: RIC-57 (documentation) and RIC-149 (test cases) checking status from B.
  - Clear: RIC-217 only documentation left (target: late commit work in Bronze-R4)
  - Still under clarification: RIC-80 actually done, but commit problems
  - Clear: RIC-84 implementation ready, but came to late for Bronze-R4. We merge in the beginning of 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”
  - 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-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) Unclear: RIC-124

Status 2020-03-31

- Jira: 53 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) + 6 (done or done, but pending review by PO) + 35 (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 (46 = 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.
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](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 demonstrated any time with real equipment in OWL
- First ideas for a Cherry Use Case: Software Management: Upgrade from xRan-frontend to o-ran-frontend 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: Yingying Wang

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

- [OCU-1] - O-CU F1 interface for Release B
- [OCU-2] - O-CU SDAP
- [OCU-3] - O-CU PDCP
- [OCU-4] - O-CU RRC
- [OCU-5] - O-CU Ng
- [OCU-6] - O-CU 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
- https://jira.o-ran-sc.org/browse/ODUHIGH-8
- https://jira.o-ran-sc.org/browse/ODUHIGH-9
- https://jira.o-ran-sc.org/browse/ODUHIGH-10
- https://jira.o-ran-sc.org/browse/ODUHIGH-11
- https://jira.o-ran-sc.org/browse/ODUHIGH-27

O-DU Low

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:

<table>
<thead>
<tr>
<th>Epics</th>
<th>Planning</th>
<th>Jira Epic Link</th>
<th>status</th>
</tr>
</thead>
</table>
| 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](https://jira.o-ran-sc.org/browse/ODULOW-2) - create L1/L2 Interface APIs based on FAPI specification | on-going |

1 epic  
1 story
**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* - create L1 application binary linker (TO DO)

**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* - enhance FH lib to support new features (TO DO)

**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:**

---

**Simulators (SIM)**

**Primary Goals:** Providing software simulators which are needed by the other projects.

**Bronze Feature Scope:** Support both the Health Check and the Traffic Steering use cases.

**PTL:** Alex Stancu

**Jira: Count of Epics, User Stories, Tasks, and Issues:** 12 issues

---

**Infrastructure (INF)**

**Primary Goals:** Provide an open source reference 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.
<table>
<thead>
<tr>
<th>Documentation (DOC)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary Goals:</strong> Bronze release documentation for OSC and developers</td>
</tr>
<tr>
<td><strong>Bronze Feature Scope:</strong> Improve current documentation skeleton and content, make documentation more understandable for readers. Focusing on integration documentation, the new feature of Bronze release.</td>
</tr>
<tr>
<td><strong>PTL:</strong> weichen ni</td>
</tr>
<tr>
<td><strong>Jira:</strong> Count of Epics, User Stories, Tasks, and Issues:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Integration and Test (INT)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary Goals:</strong> CICD, integration and deployment of O-RAN SC developed software, software testing.</td>
</tr>
<tr>
<td><strong>Bronze Feature Scope:</strong> 1. CICD to support software static analysis and quality reporting, and integrated testing for Near RT RIC; 2. &quot;One-click&quot; 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.</td>
</tr>
<tr>
<td><strong>PTL:</strong> Lusheng Ji</td>
</tr>
<tr>
<td><strong>Jira:</strong> Count of Epics, User Stories, Tasks, and Issues: 52 issues</td>
</tr>
</tbody>
</table>

| 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, configuration management, software management, host management and service management. Improve the automation by adopting the ansible, and integrate with INT project. |
| **PTL:** Xiaohua Zhang |
| **Jira:** Count of Epics, User Stories, Tasks, and Issues: |
| **INF project EPICs in Release B** |
| 15 EPICs in total, 2 EPICs ongoing, 1 EPIC (document) is todo. Currently is during the bug fixing phase (by the end of 27th-April-2020) |

⚠ com.atlassian.sal.api.net.ResponseException: Can not retrieve jira chart image