

Seed Code Contribution/ New Project Proposal Process (DRAFT)

2021

Company	Functions /Components implemented by contribution	Running Platform/ Environment	Source code repos	Description	License	committers	Language	Target Date	Status
ORAN Alliance WG6	AAL FEC	x86 Server	aal_fec/profile	<p>Why</p> <p>WG6 is considering splitting its AAL FEC specification into two portions to be published together in the July 2021 train – a) a formal specification document published through the normal O-RAN FRAND process that covers stage 1 and stage 2 specifications, and b) a stage 3 API document that covers data structures and programming language level details specific to open source implementations such as DPDK that would be published in ORAN OSC repo under a BSD license and referenced from the formal AAL FEC specification. The purpose is to enable easier ingestion of the ORAN FEC gaps identified by the ORAN community in upstream communities such as DPDK.</p> <p>What</p> <p>To facilitate this split, WG6 co-chairs would like to request an OSC repo titled aal_fec/profile to be created for the above purpose. The purpose of this repository would be to store stage 3 specification documents and in the future, header files as well as test artifacts for the AAL FEC profile. Releases from this repository will be aligned with ORAN specification release trains as we intend to publish the stage 3 specification updates conjointly with the main spec.</p> <p>License</p> <p>BSD – we would prefer a BSD license to be compatible with the upstream community (DPDK) with which the spec is to be aligned.</p>	Exception request: BSD	niall.powe@intel.com (PTL)	C / C++		To be considered (June /30 /21)

2020-2019

Company	Functions /Components implemented by contribution	Running Platform/ Environment	Source code repos					Target Contribution Date	Status
			repo name	Description	code license	committers	language		
Lenovo	Abstraction Adaptation Layer	OpenStack Hypervisor	aal/lib		Apache2	liufeng24@lenovo.com, lujq1@lenovo.com			done
			aal/logic		Apache2	liufeng24@lenovo.com, lujq1@lenovo.com			done
			aal/mgmt		Apache2	liufeng24@lenovo.com, lujq1@lenovo.com			done
			aal/virt		Apache2	liufeng24@lenovo.com, lujq1@lenovo.com	C/C++		done
China Mobile	Integrated eNB	x86 Server							
Inspur	Infrastructure monitoring	x86 Server	imp/metal	Host Configuration Management	Apache2	gaosong.lc@inspur.com, liutao.lc01@inspur.com, qiaoxj@inspur.com	Python		requested 2019/06 /11, LF Ticket #76 725
			imp/metal-api	Host management interface	Apache2	gaosong.lc@inspur.com, liutao.lc01@inspur.com, qiaoxj@inspur.com	Python		requested 2019/06 /11, LF Ticket #76 725

			imp/fm	Host Fault Monitoring, Detection and Recovery	Apache2	gaosong.lc@inspur.com , liutao.lc01@inspur.com , qiaoxj@inspur.com	Python		requested 2019/06 /11, LF Ticket #76 725
Intel	DU layer 1	x86 Server	o-du/phy	O-RAN DU Layer 1	Apache 2	zhimin.yuan@intel.com , vitaliy.zakharchenko@intel.com , nicolas.chautru@intel.com , luis.farias@intel.com	C/C++		requested 2019/06 /11, LF Ticket #76 725
Radisys	DU layer 2 and layer 3	x86 Server	o-du	O-RAN DU	Apache 2	balaji.shankaran@radisys.com , somsh.ekar.ydlapur@radisys.com	C		requested 2019/06 /11, LF Ticket #76 725
			o-du/l2	O-RAN DU Layer 2	Apache 2	balaji.shankaran@radisys.com , somsh.ekar.ydlapur@radisys.com	C		requested 2019/06 /11, LF Ticket #76 725

Guidelines:

Seed code preparation:

1. Check dependency license
 - a. Make sure they are compatible with the contribution license
2. Remove company proprietary and internal information
 - a. Internal information may include internal email addresses, URLs, etc
3. Include LICENSES.txt file at root of the repo
 - a. For Apache 2 and Creative Commons 4 licenses: [LICENSES.txt](#)
 - b. For O-RAN Software License and Creative Commons 4 licenses: (to be added)
4. Include license and copyright claim at the beginning of every source code file
 - a. For Apache 2 (using language specific code comment format)

```
#####
#   Copyright (c) [years] [company name]                                     #
#                                                                                   #
#   Licensed under the Apache License, Version 2.0 (the "License");             #
#   you may not use this file except in compliance with the License.             #
#   You may obtain a copy of the License at                                       #
#                                                                                   #
#       http://www.apache.org/licenses/LICENSE-2.0                             #
#                                                                                   #
#   Unless required by applicable law or agreed to in writing, software           #
#   distributed under the License is distributed on an "AS IS" BASIS,             #
#   WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.     #
#   See the License for the specific language governing permissions and           #
#   limitations under the License.                                                #
#####
```

- a. For O-RAN Software License: (to be added)

Repo creation:

1. Completing the columns under "Source code repos"
2. Review your proposal with the Requirements and Software Architecture committee
3. ToC approval of the creation and committers
4. Making request to Linux Foundation

What is a project?

IS / HAS	IS NOT / DOES NOT
<ul style="list-style-type: none"> • A long term endeavor to deliver features across multiple releases • Rationalized based upon clear technical and industry benefits • Has a clearly defined scope that can be leveraged across many use cases • Requires a dedicated content repository • A community which acts collectively as a single development entity • Has Committers with expertise in the relevant areas 	<ul style="list-style-type: none"> • Intended for a single ONAP release • Rationalized based upon convenience for an individual entity • Scoped broadly or a collection of unrelated items to satisfy a single use case • Something that can exist without having a dedicated repo • A fully self-contained internal development team • Require Committers with different expertise

The basic steps for submitting a proposal

- Talk to community members first about the problem you think needs to be addressed.
- Draft a project proposal
- Meet with the RSAC committee for an initial review
- Address any concerns raised by the Architecture subcommittee scheduling a follow-up review if necessary
- **A MINIMUM OF 2 WEEKS PRIOR TO MAKING YOUR PRESENTATION TO THE TOC** your submission should be considered to be locked. Nothing else should be added or changed from that point on.
 - Add your proposal presentation to an upcoming TOC at least two weeks in the future (you can do that yourself)
 - Enter your proposal in the wiki (or email)
 - Send email to toc@lists.o-ran-sc.org notifying the community that it has been posted and include a link to the proposal

The TOC considers many things when they look at approving a new project

Technology for the sake of technology won't cut it, neither will solutions in search of problems.

- Is your project addressing a real problem in the industry?
- How does this affect other projects?
- Does this advance our architecture?
- How does this fit with the next release? How about R+1, +2..?
- Does it increase our risk of delivering on time?
- Does it support multiple use case scenarios?
- Do you have enough resources?
- Are those resources new, or are you drawing resources away from other projects?
- Will this project make ORAN-SC more attractive to service providers?
- Would this simplify or complicate end user adoption?



Pro Tip

The TOC will always look more favorably on project proposals that address gaps in existing ORAN-SC functionality vs. replacing existing functionality with a new proposal.

Best Practices for successful proposal

- Always remember this is an open source community, not contract labor. No one is required to do anything.
- Talk to people from as many different companies as possible in advance
- Ensure participation of at least 3-4 organizations, including at least one operator (preferably more)
- Circulate draft proposals among the community to obtain feedback before presenting to the Architecture subcommittee
- Respond promptly to any questions from the community, especially after you notify the TOC list.

Things to avoid

- Changing the architecture in the project proposal- Architecture changes should be discussed in the Architecture subcommittee **before** the new project is ever proposed
- Duplicating, moving, or changing functionality in existing projects, **UNLESS**
 - Unless you have the full support from the existing project team that would be impacted
 - Or the other project's component was designed to be modular and you are adding a driver
- Thinking someone has made a resource commitment just because they agree with something you said



Tip

The TOC will always look more favorably on project proposals that address gaps in existing ORAN-SC functionality vs. replacing existing functionality with a new proposal.

Filling out your proposal

As you go through the proposal here are the things to keep in mind. Many of these are typical of the questions that you get asked over and over. A project proposal should answer six fundamental questions:

1. **WHO** will be doing the work?
2. **WHAT** do you propose doing?
3. **WHEN** do you plan to deliver it?
4. **WHERE** will you put your deliverables?

5. **WHY** should we consider doing this?
6. **HOW** does this fit into our architecture?

The Overview

- What is the problem?
- Why can't it be solved in existing projects?
- How do you propose we solve it?

The Architecture

- How does your proposal fit with the **existing** ORAN-SC architecture?
 - This is the question that will ultimately receive the most scrutiny. If that cannot be clearly explained your project won't be taken seriously.
- Failing to adequately engage the Architecture subcommittee and failing to heed their input is a losing strategy. Although the subcommittee cannot approve or disapprove a project proposal its recommendation to the TSC is highly weighted.
 - If you think that the problem to be solved is fundamentally an architecture problem then bring that issue up for discussion on an Architecture call first **before** you ever trying to proposing a new project.
 - If you think that the ORAN-SC architecture needs to be modified in order to accommodate your project, be prepared for very long and drawn out review process that is unlikely to be approved.

The Resources

- This page is for listing **commitments**. DO NOT list people that you hope will get involved, or folks that you think might benefit, or even people that may have said, "I think this is a good idea." Only list those who have signed up to work on your project. Period.
- The PTL should have had some manner of direct 1:1 conversation with everyone listed to before making a proposal.



Tip

Thinking of or speaking of companies or individuals as "proposed contributors" to your project will almost always get you into trouble. It is probably best to consider it to be a binary; either they directly committed resources to work on the project to you first hand, or they did not. Comments such as, "Yes, that sounds like a good idea", "Yes, my company would like something like that", or even "Yes, I'd be very interested in your project!" are not the same as someone saying, "**Yes, I can work on your project**".

Source: Credit to ONAP community for new [Project Proposal page](#) guidelines, which we have replicated.