



LF NETWORKING
Developer & Testing Forum

ONAP: Streamlining the process

Magnus Buhrgard, Byung-Woo Jun - Ericsson

June 2023

<https://lfnetworking.org>



Anti-Trust Policy Notice

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.
- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at <http://www.linuxfoundation.org/antitrustpolicy>. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Updegrove LLP, which provides legal counsel to the Linux Foundation.

Agenda

Workshop on streamlining the ONAP process based on the new mission statement

- New mission statement
- Short recap
- Right size the ambition level
- Inspiration from other open-source projects
- Some suggestions

- Discussion

New ONAP Mission Statement

ONAP is an open-source **project containing a set of autonomous components** for orchestration, management, and automation of network and edge computing services for network operators, cloud providers, and enterprises.

The components are designed by **self-organizing projects**, evolving in dialogue with code consumers, as well as **adhering to ONAP's global requirements and best practices**. To increase the value of the components to the consumers, **use cases are identified to guide the bundling of components** into packages and integrating components for proof-of-concept demonstrations.

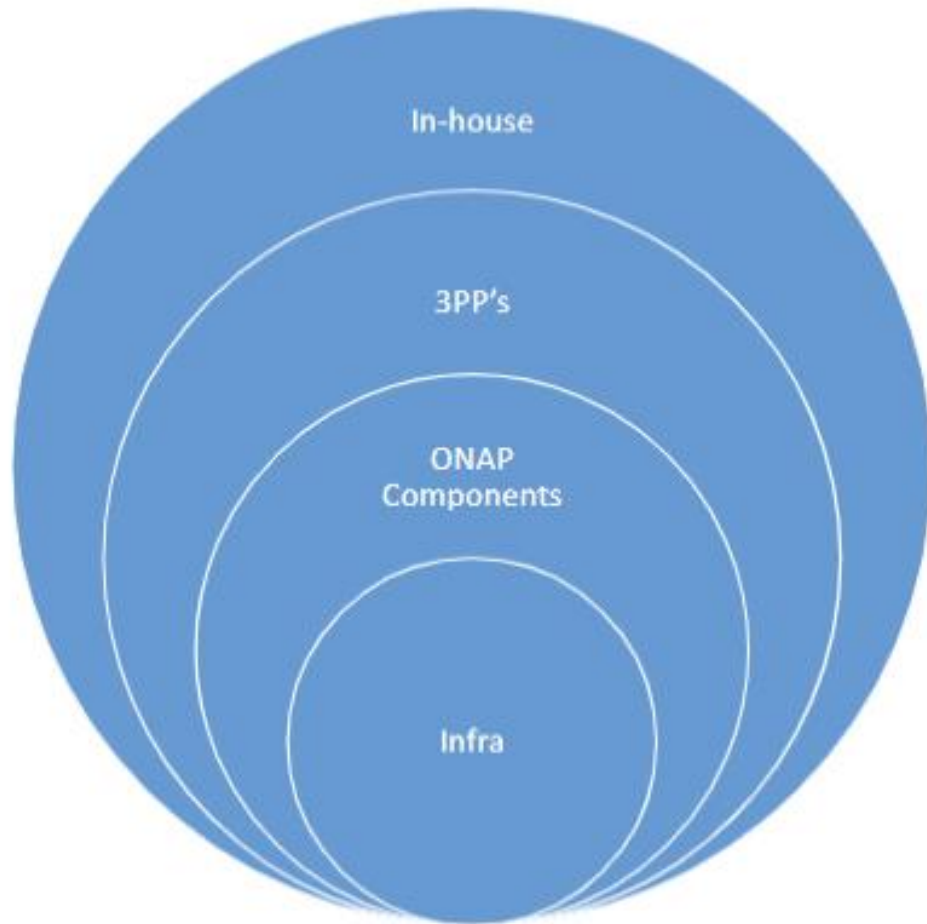
ONAP is **also an industry collaboration** to define, describe and evolve a **reference architecture** for real-time, policy-driven orchestration, management, and automation of physical, virtual, and cloud-native network functions.

ONAP – Key takeaways from Bell Canada

Ram Krishna Verma

Senior Solutions Architect

How are we using ONAP?



Components Used

- ✓ SDC
- ✓ SO
- ✓ AAI
- ✓ Policy Framework
- ✓ CDS
- ✓ SDNC
- ✓ DMaaP

Use Cases

- ✓ Orchestration
 - Network Slicing
 - Zero touch provisioning
- ✓ Automation
 - Closed Control Loop
 - Open Control Loop

London Architecture Overview

Feb 2022



OSS / BSS / Other

NBI

Legend

Active

Proposed (new)

Design-Time

Run-Time

Manage ONAP

ONAP Operations Manager (OOM)

London



VNF Validation

VVP

VNF SDK

Portal-NG

O&M Dashboard (VID)

Use-Case UI

External APIs

CLI

Interfaces

Service Design & Creation (SDC)

Service/xNF Design

xNF Onboarding

Workflow Designer

Catalog

(...)

Policy Framework

Correlation Engine (Holmes)

Service Orchestration (SO)

Active & Available Inventory (AAI)

Microservice Bus (MSB)

Message & Data Routers (DMaaP)

Shared Services

Security Framework

OOF

CPS

Logging Framework(poc)

DCAE

Analytics

Collectors

Infrastructure Adaptation (Multi-VIM/ Cloud)

Controller Design Studio (CDS)

SDN Controller (SDNC)

Virtual Function Controller (VFC)

External Systems

Third Party Controllers

sVNFM

EMS

ONAP Shared Utilities

CCSDK

Model Utilities

Network Function Layer

VNFs

...

CNFs

PNFs

Hypervisor / OS Layer

OpenStack

Commercial VIM

Kubernetes

Public Cloud

Private Edge Cloud

MPLS

Private DC Cloud

IP

Public Cloud

Managed Environment

ONAP benefits to the industry

- ONAP as a platform has shown e2e network automation to the industry.
- Operators, vendors and enterprises have learned how service/network automation (modeling, orchestration, policy-based closed loop, optimization...) works on VM and Cloud-Native environments for VNF, PNF, CNF, NS, Network/RAN Slicing and e2e service thru ONAP.
- Now, the operators, vendors and enterprises want to select and apply ONAP functions to their portfolio. No one needs to take ONAP as a whole.
- In ONAP, there are numerous valuable use cases, that leverage and coordinate clusters of ONAP component functions (e.g., SDC, SO, A&AI, DCAE, Policy, SDNC, SDNR, CPS, CDS...) to achieve objectives, such as:
 - E2E Network Slicing
 - RAN slicing
 - Closed Loop
 - ETSI-based NS & VNF orchestration
 - Helm-based CNF orchestration
 - ASD-based (including Helm) CNF orchestration



- Our goal is to continue to support those use cases efficiently for use in commercial production environments and portfolios.
 - We expect the industry wants to pick and choose desired ONAP component functions, swap some of the ONAP functions, and integrate those functions into their portfolios seamlessly, without bringing in a platform.
 - ONAP streamlining, which drives individual components and clusters of components guided by use cases, will enable the flexible and dynamic function adoption by the industry.
- ✓ ONAP stakeholders are thinking about connecting ONAP, ORAN, Nephio, EMCO, and other communities for larger objectives.
 - ✓ Reuse of selected ONAP functions
 - ✓ Functional delegations
 - ✓ Under these circumstances, ONAP streamlining is more desirable.

Great accomplishments!

What is consumable in ONAP?

- Individual components (run by self organizing teams)
 - The teams dictate their own processes and timelines
 - Centers of excellence
 - Flexible dialogue with users
 - Continuous development and responsive deliverables



What is consumable in ONAP?

- Individual components (run by self organizing teams)
 - The teams dictate their own processes and timelines
 - Centers of excellence
 - Flexible dialogue with users
 - Continuous development and responsive deliverables
- Cluster of components guided by use cases
 - Bringing greater value than individual components
 - Useful in marketing, Proof-of-Concept



What is consumable in ONAP?

- Individual components (run by self organizing teams)
 - The teams dictate their own processes and timelines
 - Centers of excellence
 - Flexible dialogue with users
 - Continuous development and responsive deliverables
- Cluster of components guided by use cases
 - Bringing greater value than individual components
 - Useful in marketing, Proof-of-Concept
- Platform
 - No commercial uptake
 - No smooth upgrade
 - Sets expectations for a scope way beyond what can be expected from a “normal” open-source community
 - Based on a corporate development mindset



ONAP needs to get more agile and better at managing expectations

Right size the ambition level

- The individual component teams set the expectations
- Proposals needs to backed up with secured development resources
- Smooth interoperability is a benefit - not a default
- Documentation is best-effort – complemented by contact information to the developer network
- Governance → Special Interest Groups + Support

Inspiration from other projects

- [Cloud Native Computing Foundation \(cncf.io\)](https://cncf.io)
- [CD Foundation](#)

MAKE CLOUD NATIVE UBIQUITOUS

CNCF is the open source, vendor-neutral hub of **cloud native computing**, hosting projects like Kubernetes and Prometheus to make cloud native universal and sustainable.

[ABOUT CNCF](#)





















158
Projects

203K
Contributors

12.9M
Contributions

188
Countries

CNCF graduated projects

 argo ★ 12,791 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 containerd ★ 13,753 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 CoreDNS ★ 10,545 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 envoy ★ 21,858 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 etcd ★ 43,211 Cloud Native Computing Foundation (CNCF) Funding: \$3M
 fluentd ★ 11,914 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 flux ★ 4,731 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 HARBOR ★ 19,867 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 HELM ★ 24,171 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 JAEGER ★ 17,481 Cloud Native Computing Foundation (CNCF) Funding: \$3M
 kubernetes ★ 97,680 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 LINKERD ★ 9,519 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 Open Policy Agent ★ 7,935 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 Prometheus ★ 47,774 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 ROOK ★ 11,305 Cloud Native Computing Foundation (CNCF) Funding: \$3M
 spiffe ★ 1,213 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 SPIRE ★ 1,392 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 The Update Framework (TUF) ★ 1,500 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 KV ★ 12,969 Cloud Native Computing Foundation (CNCF) Funding: \$3M	 Vitess ★ 16,067 Cloud Native Computing Foundation (CNCF) Funding: \$3M

CNCF outreach

CNCF Ambassadors

BROWSE GEOGRAPHICALLY



Technical Advisory Groups

The TOC has approved the formation of TAGs. Currently, the following Technical advisory Groups are active:

- [TAG-Security](#)
- [TAG-Storage](#)
- [TAG-App-Delivery](#)
- [TAG-Network](#)
- [TAG-Runtime](#)
- [TAG Contributor Strategy](#)
- [TAG Observability](#)
- [TAG Environmental Sustainability](#)

**Improving the World's Ability
to Deliver Software with
Security & Speed**



CD.FOUNDATION

CDF Projects



Special Interest Groups

Best Practices



Interoperability



Events



Software Supply Chain



MLOps



Technical Initiatives

Key Initiatives

Interoperability

Reference Architecture

Software Supply Chain

Best Practices

Ambassador Cohort 2023



Garima Bajpai
Ambassador Chair



Adam Gardner
Dynatrace | Australia



Adetokunbo Ige
AWS Community Builder



Anna Daugherty
Armory



Anurag Sharma
Zensar Technologies UK LTD



Aravind Arumugham
Fidelity Investments



Batuhan Apaydin
Trendyol Group | Turkey



Bervianto Leo Pratama
Mitrans



Brad McCoy
Play Travel | New Zealand



Eduardo Piairo
Devoteam



Enric Forn Jorba
CaixabankTech | Spain



Everest K.C.
Boise State University



Ganesh Sharma
LambdaTest



Giorgi Keratishvili
Georgia



Giovanni Galloro
Google



Gurtej Pal Singh
Inosys



Jesus Rodriguez
ABCDin



Kais Salhi
Hydro Quebec



Matheus Paes Pereira
PicPay | Brazil



Michael Cade
Kasten By Veeam



Moïse Kamen
Hydro-Québec | Canada



Muktesh Mishra
Adobe | USA



Natale Vinto
Red Hat



Oleg Nenashev
WireMock Inc



Pawel Pliwosz
EPAM Systems | Poland



Raghavendra Guttr
Dell EMC



Rajat Gupta
Jenkins X



Raminder Rathore
HCLTech



Ricardo Castro
FanDuel/Blip.pt | Portugal



Saifeer C.M.
Flipkart | India



Santosh Kumar Perumal
Tata Consultancy Services |



Savinder Puri
TBC Bank, Tbilisi | Georgia



Sriram R.J.
Capgemini | USA



Stephane Montri
Palo Alto Networks | France



Suresh Babu
OpsMx



Szymon (Simon) Grzebieta
Innabr | Australia



Thomas Schuetz
WhizUs | Austria



CD.FOUNDATION



This wiki is for **developers** of ONAP, an open source software platform for the orchestration, management, and automation of network and edge computing services for network operators, cloud providers, and enterprises. Its real-time, policy-driven orchestration and automation of physical and virtual network functions enables rapid automation of new services and complete lifecycle management critical for 5G and next-generation networks.

If you are looking for documentation on administering or deploying ONAP, please head to our official ReadTheDocs site: <https://docs.onap.org>



Latest: Release 11 (Kohn)

[Get The Code](#)

Check ReadTheDocs for ...

- [Official Documentation](#)
- [Release Notes](#)
- [Architecture Overview](#)

ONAP Projects List

Getting Involved in the ONAP Community

- [Code of Conduct](#)
- [Slack channel](#) (most contributing company domains should already work)
- [LFN Membership Guide](#)

Technical Steering Committee (TSC)

- [Community Meetings & Calendar](#)
- [How-to articles](#)
- [Instant Messaging](#)
- [Security Space Wiki Access List](#)
- [Community Offices and Governance](#)
- [Intel/Windriver Openstack Lab Access](#)
- [Communications, Contacts & Email](#)
- [Developer Best Practices](#)
- [Community Meetings & Calendar \(deprecated\)](#)
- [2017 Launch Presentations](#)
- [Legacy ONAP event repo](#)
- [Mailing Lists](#)
- [Joining the ONAP Technical Community](#)

Release Planning & Management

- [Release Planning](#)
- [Project Status](#)

Upcoming Events

- [2023-02 LFN Developer & Testing Forum \(Virtual\)](#)

Previous Events

LF Networking - proposal

LF Networking Projects



A&AI



SDN-C



UC UI



CCSDK



DCAE



PortalNG



SDC



CPS



VFC/NFVO

by



OPEN NETWORK AUTOMATION PLATFORM

ONAP component obstacles, observations & challenges

- ONAP components are designed for ONAP-specific consumption.
 - Instead of a component being graduated, an ONAP component becomes obsolete or unmaintained if ONAP does not have use cases for it.
 - Some ONAP component-specific features tend to be ignored if they are not used by other ONAP components.
 - ONAP component functions should be used by not only ONAP but also non-ONAP.
 - Component design should be generic and extensible in a way that would enable it to be used in non-ONAP
 - If components are more generally applicable, there is the potential to gain more traction.
- Component dependencies and couplings to other ONAP components are in an ONAP-specific way.
 - Those dependencies and couplings could be both syntactic and semantic.
 - Numerous intra-ONAP component interfaces and communications are ONAP-specific.
 - Some limited APIs standardization efforts are in place, such ETSI MANO APIs, ASD, 3GPP...
- Making each ONAP component 'stand-alone' will highlight to potential users that they can take a single component, without getting involved in the whole of ONAP.

ONAP component obstacles, observations & challenges

- Deviating from standards makes integration with other systems problematic, especially for non-ONAP.
 - Aligning with standards where possible should be global requirements.
 - If there must be a deviation, that can be done in an extensible way that enables the standard-based approach
- Component Helm charts in OOM may need to be re-written to build/deploy a component individually.
 - CI build/integration of a vendor/operator could be less compatible with ONAP one.
 - OOM is not used by some vendor/operators.
 - In some cases, a vendor maintains a completely different set of Helm charts for ONAP components.
- Vendor/operator-specific security and logging requirements could be different. It causes integration issues. The current security based on Service-Mesh, Ingress and Keycloak should be maintained.
- Timelines and cadence of the ONAP release are inflexible for accommodating different release strategies.
 - Cannot create a 'Release' in JIRA for the component releases
 - Branching strategies are not aligned with ONAP CMO (Current Mod of Operation)
 - Resulting in an artificial split in functionality between releases

ONAP component streamlining target

- Modularity & independent management
 - Stand-alone component
- Interface abstraction & loose coupling
 - Including standardization where possible
- Extensibility & interchangeability
- Scalability (component addition, update and deletion without disruption)
- Autonomous self management
- Design for general use (ONAP & non-ONAP consumers)
- Conformance to industry security & logging
- Clustering components by use cases
 - Selection of the best components for a particular task in systems
 - Responsive integration and delivery
 - ONAP still can provide reference automation for coordination



A&AI



SDN-C

UC UI

CCSDK



DCAE



DESIGN
CREATION



CPS

SDC

PortalNG



VFC/NFVO

by



Special Interest Groups (SIG)

- Technical coordination and governance (former TSC)
- Architecture & Interoperability (could be on LFN level)
- LFN security
- LFN common practices
- Modeling
- LFN documentation consistency
- Technical outreach (SDO & Open-source)

Release Management Tasks

- Assuming that we keep coordinated *by*  **ONAP** releases even when the platform has been discontinued
- Continued review of Release management tasks for further streamlining – proposals in the following slides

Release Management Tasks

	M	Project Tasks	Intent / Background	
1	M1	Review Code Coverage goal vs. actuals	Maintain a minimum level of test coverage of code	Removed
2	M1, M4	Update the FOSS (Free and Open Source Software) wiki page (Project FOSS → Project)	Maintain a record of OSS used in the project.	Removed
3	M1	Request an architectural subcommittee review	Make sure that PTLs request a review well in advance of M2.	Keep – SIG Architecture & Interop (LFN level?)
4	M1	Document API issues in the requirement description	Ensure that API changes are documented for any dependencies.	Removed
5	M1	DOCS: create documentation tracking page and pre-fill information.	DOCS project only. Make sure that the documentation tracking page is in place for each release	Change to SIG LFN documentation consistency
6	M1	Complete release planning template	Provides visibility to the TSC and community into the activity planned by each ONAP project for the release.	Simplified Keep
7	M2	Verify information in documentation tracking page. Update as necessary.	Provides visibility to the TSC and community into changes to ONAP documentation. It also enables the DOCS team to track documentation release steps, particularly for new documents.	Moved to M3 Move to SIG LFN documentation consistency

Release Management Tasks

	M	Project Tasks	Intent / Background	
8	M2	Update documented risks	Highlight technical, resource, and schedule risks to the release for each project.	Removed
9	M3, RC	Review license scan issues (Note: this is typically done 2 - 3 times per release)	Prevent releases with licensing violations.	Keep SIG Modeling
10	M2	Data models shared with Modeling subcommittee	?	Modified Keep
11	M2	Complete Architectural subcommittee review	Completing the arch review is a key goal of M2. This task helps ensure that the review has been completed.	Keep – SIG Architecture & Interop (LFN level?)
12	M2	Color code Impact View Per Component page	Ensures that PTLs complete the handshake with requirement owners by indicating which requirements they support.	Modified Delegate to UC owners
13	M2	Communicate API changes to other projects	Ensure that projects with dependencies are informed of API changes.	Keep
14	M3	Verify that test coverage goals have been met	Maintain a minimum level of test coverage of code	Keep
15	M3	Resolve all Global Requirement impacts	Promote compliance with global requirements.	Removed

Release Management Tasks

	M	Project Tasks	Intent / Background	
16	M2, M3, M4	Verify that there are no merge requests older than 36 hours	Ensure that milestone status is evaluated with relevant MRs completed. Note: repeated 2 - 3 times throughout release.	Removed
17	M3	Resolve high/highest priority JIRA issues	Don't allow high priority issues to pile up at the end of the release. Ensure that milestone status is evaluated with high priority issues resolved. Note: repeated 2 times throughout the release.	Removed
18	M4	Start OOM review with updated container image	Ensure that PTLs have started a review by M4.	Remove ?
19	M4	Assign Jira issues to the release	Ensure that the "fix version" field for issues that are planned to be resolved for the release is set to the current release.	Remove ?
20	M4	Complete preliminary documentation	This is primarily for new projects or new documents. The intent is to avoid confusion at the end of the release over basic organization and workflow of documentation.	SIG LFN documentation consistency support PTL
21	M4	Review and update INFO.yaml	Prevent INFO.yaml files from becoming stale and irrelevant.	Keep

Release Management Tasks

	M	Project Tasks	Intent / Background	
22	M4	Update integration weather board	Track progress in passing health check, completing pairwise testing, and reaching platform maturity target	Removed
23	M4	Update Release Platform Maturity and CII badging	Track actual progress vs plan.	Remove
24	M4	DOCS: confirm that PTL repo changes in M2 (new/removed repos) and M4 (preliminary doc) are represented in master and RTD	DOCS team only Ensure that new or removed documentation, and preliminary documentation changes are accurately and properly reflected in the master branch and in read-the-docs.	SIG LFN documentation consistency
25	RC	Create a release branch	Self explanatory	Keep
26	RC	Complete key updates page	Provide fodder for ONAP promotion.	Keep
27	RC	Verify that pairwise testing has been completed	Ensure that critical test step has been completed.	Delegate to UC owners
28	RC	Deliver updated container to integration team, if necessary	Ensure that the integration team has the latest container	Removed
29	RC	Complete project testing	Complete the testing that the project proposed to do in their project release plan.	Moved to sign-off Keep
30	RC	Finalize documentation	Ensure that projects complete their documentation for the release.	SIG LFN documentation consistency

Release Management Tasks

	M	Project Tasks	Intent / Background	
31	SO	Verify readiness of release artifacts	Final check that project artifacts are ready for release.	Removed
32	SO	JIRA Cleanup	Close issues assigned to the current release, or reassign them to the next release by updating the “fix version” field.	Keep
33	SO	DOCS: verify that repo branch exists, verify that RTD branch exists, verify that project release notes have been finalized	DOCS team only. Final evaluation of documentation for release.	Moved to RC SIG LFN documentation consistency