# Approved Projects

This page lists all projects that have been approved by TSC to be a part of the ONAP Platform.

<table>
<thead>
<tr>
<th>Project Name</th>
<th>Approved Date</th>
<th>Project Tech. Lead</th>
<th>Description</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>A &amp; AI</td>
<td>June 9, 2017</td>
<td>William Rozeh (replacing Saad Forough)</td>
<td>Active and Available Inventory (AAI) is the ONAP subsystem that provides real-time views of Resources and Services and their relationships. AAI not only forms a registry of active, available, and assigned assets, it also maintains up-to-date views of the multidimensional relationships among these assets, including their relevance to different components of ONAP.</td>
<td>Approved v.36</td>
</tr>
<tr>
<td>Application Authorizations Framework</td>
<td>June 29, 2017</td>
<td>TBD (replacing Jonathan Gutman)</td>
<td>The goal of the project is to provide consistent authentication, authorization and security to various ONAP components. AAF organizes software authorizations so that applications, tools and services can match the access needed to perform job functions. AAF is designed to cover Fine-Grained Authorization, meaning that the authorizations provided are able to use an Application’s detailed authorizations, such as whether a user may be on a particular page, or has access to a particular Web-Sub topic controlled within the App. This is a critical function for Cloud environments, as Services need to be able to be installed and running in a very short time, and should not be encumbered with local configurations of Users, Permissions and Passwords. The sister framework CADI (Code Access Identity) allows Java Applications to utilize Identity Authentication methods as plugins. Certificate Manager delivers X509 certificates in support of 2 way X509 TLS.</td>
<td>Approved V.36</td>
</tr>
<tr>
<td>APPC</td>
<td>June 9, 2017</td>
<td>TBD (replacing Tu kumere Cho)</td>
<td>The Application Controller (APPC) performs functions to manage the lifecycle of VNFs and their components providing model driven configuration, abstract cloud/VNF interfaces for repeatable actions, uses vendor agnostic mechanisms (NETCONF, Chef via Chef Server and Ansible) and enables automation.</td>
<td>Approved V.46</td>
</tr>
<tr>
<td>CLAMP</td>
<td>June 8, 2017</td>
<td>Gervais Martial Ngoket</td>
<td>CLAMP is a platform for designing and managing control loops. It is used to design a closed loop, configure it with specific parameters for a particular network service, then deploying and undeploying it. Once deployed, the user can also update the loop with new parameters during runtime, as well as suspending and restarting it.</td>
<td>Approved v.37</td>
</tr>
<tr>
<td>Common Controller SDK</td>
<td>June 9, 2017</td>
<td>Dan Timoney</td>
<td>This project provides a common set of reusable code that can be used across multiple controllers.</td>
<td>Approved v.39</td>
</tr>
<tr>
<td>DCAE</td>
<td>June 9, 2017</td>
<td>Vijay Venkatachalam Kumar (replacing Lu shong Ji)</td>
<td>DCAE is the umbrella name for a number of components collectively fulfilling the role of Data Collection, Analytics, and Events generation for ONAP. The architecture of DCAE targets flexible, plugable, micro-service oriented, model based component deployment and service composition. DCAE also support multi-site collection and analytics which are essential for large ONAP deployments.</td>
<td>Approved v.60</td>
</tr>
<tr>
<td>DMaAP</td>
<td>June 23, 2017</td>
<td>Mandar Sawant (replacing Ram Koya)</td>
<td>DMaAP is a premier platform for high performing and cost effective data movement services that transports and processes data from any source to any target with the format, quality, security, and concurrency required to serve the business and customer needs.</td>
<td>Approved v.50</td>
</tr>
<tr>
<td>Documentatin on</td>
<td>June 9, 2017</td>
<td>Sofia Wallin (replacing Greg Gotier)</td>
<td>Documentation creates and maintains documentation targeted to ONAP user audiences and the tasks they perform.</td>
<td>Approved v.41</td>
</tr>
<tr>
<td>External API Framework</td>
<td>June 9, 2017</td>
<td>Adrian O’Sullivan (replacing Matthew Geenbaart) (replacing Alan Mayer)</td>
<td>The External API Framework project describes and defines the APIs between ONAP and External Systems, including ONAP interfaces targeted on BSS/OSS, peering, B2B, etc.</td>
<td>Approved v.48</td>
</tr>
<tr>
<td>Holmes</td>
<td>June 9, 2017</td>
<td>Guangrong Fu</td>
<td>Holmes project provides alarm correlation and analysis for Telecom cloud infrastructure and services, including hosts, vims, VNFs and NSs. Holmes aims to find the real reason which causes the failure or degradation of services by digging into the ocean of events collected from different levels of the Telecom cloud.</td>
<td>Approved v.29</td>
</tr>
<tr>
<td>Integration</td>
<td>June 8, 2017</td>
<td>Morgan Richomme (replacing Alan Chen)</td>
<td>Integration is responsible for ONAP cross-project system integration, CI/CD, and all related end-to-end release use cases testing with VNFs necessary for the successful delivery and industry adoption of the ONAP project as a whole.</td>
<td>Approved V.71</td>
</tr>
<tr>
<td>Logging Enhancements Project</td>
<td>June 29, 2017</td>
<td>TBD (replacing Francis Au) (replaced Michael O’Brien) (replaced Alan Chen)</td>
<td>ONAP consists of many components and containers, and consequently writes to many logfiles. The volume of logger output may be enormous, especially when debugging. Large, disparate logfiles are difficult to monitor and analyze, and tracing requests across many files, file systems and containers is unfeasible without tooling. The problem of decentralized logger output is addressed by analytics pipelines such as Elastic Stack. Elastic Stack consumes logs, indexes their contents in Elasticsearch, and makes them accessible, queryable and navigable via a sophisticated UI, Kibana Discover. This elevates the importance of standardization and machine-readability. Logfiles can remain browseable, but output can be simplified.</td>
<td>Approved V.9</td>
</tr>
<tr>
<td>Mroservice Bus</td>
<td>June 8, 2017</td>
<td>Huabei Zhao</td>
<td>Microservices Bus provide key infrastructure functionalities to support Microservice Architecture including service registration/discovery, service gateway, service load balancer.</td>
<td>Approved v.73</td>
</tr>
<tr>
<td>Modeling</td>
<td>June 9, 2017</td>
<td>Hu Deng</td>
<td>The unified model-driven approach uses models as sources of data for generating processes/codes and following workflows (not code development as source) – this way, the system can be more flexible and future proof, easy to update and use for cross-platform solutions since the “only” thing needed is Model update and manipulation through Engine.</td>
<td>Approved v.66</td>
</tr>
<tr>
<td>Multi VIM /Cloud</td>
<td>June 8, 2017</td>
<td>Bin Yang (replacing Ix iwu)</td>
<td>Multi VIM/Cloud project aims to enable ONAP to deploy and run on multiple infrastructure environments, for example, OpenStack and its different distributions (e.g. vanilla OpenStack, Wind River, etc...), public and private clouds (e.g. VMware, Azure), and micro services containers, etc.</td>
<td>Approved v.87</td>
</tr>
</tbody>
</table>
Music Jan 16, 2018 TBD (replacing Ilhan Balasubramani) Approved v.61

June 8, 2017 Approved

Jim Hahn

June 9, 2017 Approved

xu ran

S

management

Yuanhong

Of

Kanagaraj

June 9, 2017 Approved

Nermin Itti

June 8, 2017 Approved

St

Approved

Dan

Sylvain

Approved

Kanagaraj

TBD

Mi

Sunder

Ikram

B

state

June 8, 2017 Approved

Seshu

VNF

VNF SDK

VID project

June 8, 2017

Iram Ikramullah

(replacing Illy Stern)

(replacing Ofir Sonsino)

Approved v.61

VNF SDK

June 8, 2017

Kanagaraj Manicham

(replacing Wael Ooa)

(replacing Chris Donley)

Approved v.61

VNF Requirements

June 9, 2017

Travis Lavett

(replacing St even Wright)

Approved v.20

In this project, we identify common state management concerns across ONAP components and provide a multi-site state coordination/management service (MUSIC) with a rich suite of recipes that each ONAP component can simply configure and use for their state-management needs.

ONAP CLI June 8, 2017 Kanagaraj Manicham This project is providing required Command-Line Interface (CLI) as commands to operate ONAP functionalities from Unix or windows shell. Approved v.20

ONAP Operations Manager June 8, 2017 Sylvain Dobrunaux (temporary) (replacing Mike Elliott) (replacing David Sauvageau) This proposal introduces the ONAP Platform OOM (ONAP Operations Manager) to efficiently Deploy, Manage, Operate the ONAP platform and its components (e.g. MSO, DCAE, SDC, etc.) and infrastructure (VMs, Containers). Approved v.70

ONAP Optimization Framework June 23, 2017 krishna morothy (replacing S Harinarayanan Puzhavakath Narayanan) This project currently provides the following two core platform optimization services, which are built to be service independent, policy driven, and extensible along with an optimization framework to enhance these or creating new services. Approved v.42

ONAP University June 22, 2017 Naomin Mohamed Provide overview of the ONAP University training courses for users, developers and any other interested parties of member and non-member companies. Approved v.9

ONAP Usecase UI Project Proposal June 8, 2017 xu ran (replacing Tad Shen) (replacing Christopher Closset (temporary) (replacing Pannir Dragoosh) (replacing S Harinarayanan Puzhavakath Narayanan) The ONAP Usecase UI project is the ONAP subsystem that provides Graphical User Interface (GUI) for operators and end-users from the point of view of use cases. Approved v.30

Policy Framework Project June 8, 2017 Jim Hahn (replacing Pannir Dragoosh) The Policy subsystem of ONAP maintains, distributes, and operates on the set of rules that underlie ONAP’s control, orchestration, and management functions. Policy provides a centralized environment for the creation and management of easily-updatable conditional rules. Approved v.46

Portal Platform Project June 8, 2017 Sundar Tattavarada (replacing Manop Talasila) The ONAP Portal is a platform that provides the ability to integrate different ONAP applications into a centralized Portal Core. Approved v.22

SDN-C June 9, 2017 Dan Timoney The SDN-C project provides a global network controller, built on the Common Controller Framework, which manages, assigns and provisions network resources. Approved v.61

Service Design & Creation June 8, 2017 Christoph Closset (replacing Ofir Sonsino) (replacing Michael Lando) Service Design Creation provides a well-structured organization of visual design & testing tools, templates and catalogs to model and create resources, and services. The output of the SDC is a set of models which drives the orchestration. Approved v.48

Service Orchestrator June 9, 2017 Sathish Kumar M The SD provides the highest level of service orchestration in the ONAP architecture. SD is implemented via BPMN flows that operate on Models distributed from SDC that describe the Services and associated VNFs and other Resource components. Approved v.59

VF.C: Virtual Function Controller June 8, 2017 Yuanhong Deng (replacing Yuan Fang) VF.C leverages ETSI NFV MANO architecture and information model as a reference, and implements full life cycle management and FCAPIs of VNF and NS. Approved v.69

VID project June 8, 2017 Iram Ikramullah (replacing Illy Stern) (replacing Ofir Sonsino) Provides a well-structured organization of infrastructure deployment, instantiation and change management operations used by Operations to derive orchestrations and change-management. Approved v.12

This is an important and challenging problem because of three fundamental reasons:

- Current solutions for state-management of ONAP components like MariaDB clustering, that work very effectively within a site, may not scale across geo-distributed sites (e.g., Beijing, Amsterdam and Irvine) or allow partitioned operation (thereby compromising availability). This is mainly because WAN latencies are much higher across sites and frequent network partitions can occur.

- ONAP components often have a diverse range of requirements in terms of state replication. While some components need to synchronously manage state across replicas, others may tolerate asynchronous replication. This diversity needs to be leveraged to provide better performance and higher availability across sites.

- ONAP components often need to partition state across different replicas, perform consistent operations on them and ensure that on failover, the new owner has access to the latest state. The distributed protocols to achieve such consistent ownership is complex and replete with corner cases, especially in the face of network partitions. Currently, each component is building its own handcrafted solution which is wasteful and prone to erroneous operations.

The VNF Requirements creates and maintains VNF Requirements targeted to VNF Provider and Services Provider and the tasks they perform.
<table>
<thead>
<tr>
<th>VNF Validation (VVP)</th>
<th>June 22, 2017</th>
<th>Steven Stark (replacing Steven Wright)</th>
<th>The project is to develop a validation program to provide assurance of VNF interoperability with ONAP. Obtaining a validation shall be a self-service activity and should be against a reference release of ONAP for use by the VNF provider &amp; any other validation authority.</th>
<th>Approved v.25</th>
</tr>
</thead>
</table>
| ONAP Benchmark      | Dec 13, 2017 | Contact: Chen Yan                      | • The ONAP Benchmark project aims to provide Function/Performance/Scalability/Security test for ONAP component/module/system level.  
• Finding bottleneck for E2E system and making advice for corresponding module for code improvement.  
• Making test report for ONAP developer/user/cooperator.  
• ONAP Benchmark develops auto test environment and auto test scripts/use cases. | Approved v21 |

30 Projects approved