IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 10, October 2020

DOI 10.17148/IJARCCE.2020.91007

Open Network Automation Platform: Solution Summary

Jasneet Singh Parmar[1]

Program Manager, De Pedagogics, Gurgaon, Haryana, India[1]

Abstract: Open Network Automation Platform (ONAP) role is to develop global and massive scale (Multi-Site and Multi-VIM) automation capabilities for both physical and virtual network elements. It facilitates service agility by supporting data models for rapid service and resource deployment, and providing a common set of Northbound REST APIs that are open and interoperable, and by supporting model driven interfaces to the networks. This technology was implemented to gain deeper knowledge of its architecture and to test its limitations in communication technology.

Keywords: AWS, Docker, Kubernetes, ONAP, OOM, Rancher.

I. INTRODUCTION

The ONAP project addresses the rising need for a common automation platform for telecommunication, cable, and cloud service providers—and their solution providers—to deliver differentiated network services on demand, profitably and competitively, while leveraging existing investments.

Prior to ONAP, operators of telecommunication networks have been challenged to keep up with the scale and cost of manual changes required to implement new service offerings, from installing new data center equipment to, in some cases, upgrading on-premises customer equipment. Many are seeking to exploit SDN and NFV to improve service velocity, simplify equipment interoperability and integration, and reduce overall CapEx and OpEx costs. In addition, the current, highly fragmented management landscape makes it difficult to monitor and guarantee service-level agreements (SLAs). These challenges are still very real now as ONAP creates its third release.

In this document we will be covering up the different components of ONAP. Also, we will be looking into how to implement ONAP using different technologies available and implementing some use cases.

II. TECHNICAL ARCHITECTURE

Architecture of ONAP:

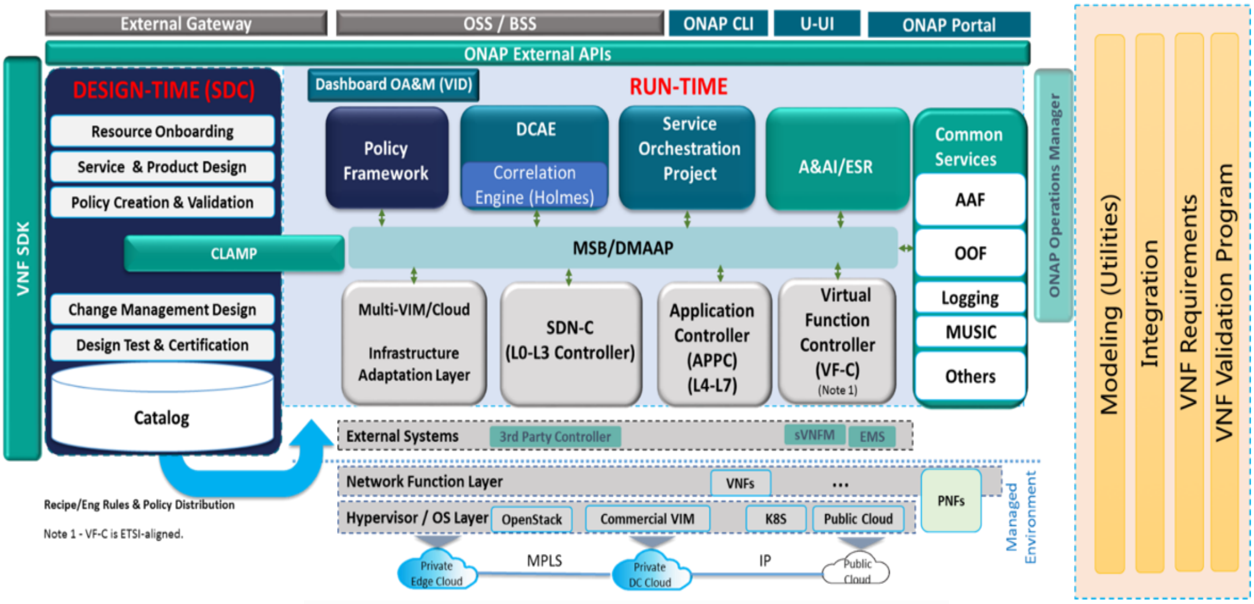

Figure 1. ONAP Platform architecture (Casablanca Release)

IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 10, October 2020

DOI 10.17148/IJARCCE.2020.91007

ONAP consists of a number of software subsystems. These subsystems are part of two major architectural frameworks:
- a design-time environment to design, define and program the platform
- an execution-time environment to execute the logic programmed in the design phase.

The design-time framework is an integrated development environment with tools, techniques, and repositories for defining and describing deployable assets. It supports the development of new capabilities, augmentation of existing capabilities and continuous operational improvement throughout the life cycle of a service. The execution-time framework uses closed-loop, policy-driven automation to drive down operational costs. Built-in dynamic, policy-enforced functions are provided for component and workload shaping, placement, execution, and administration. Access to the design-time and execution-time frameworks are provided by the ONAP Portal, a role-based user interface and ONAP CLI.

Design-time framework

The design-time framework consists of the following subsystems:
- Service Design and Creation (SDC): - SDC is the ONAP visual modeling and design tool. It creates internal metadata that describes assets used by all ONAP components, both at design time and run time.
- Policy: - POLICY is a subsystem of ONAP that maintains, distributes, and operates on the set of rules that underlie ONAP's control, orchestration, and management functions.

The SDC subsystem enables developers to define, simulate, and certify assets and their associated processes and policies. The Policy subsystem enables the creation and deployment of rules that instantiate conditions, requirements, constraints, attributes, or needs regarding the assets that must be provisioned, maintained, or enforced.

The design-time framework provides a set of common services and utilities and is intended for a variety of users with a different role. For example, the design studio enables product and service designers to onboard, extend and retire resources, services and products. Also using the design studio, operations engineers, security experts and customer experience experts can create workflows, policies and methods.

Run-time framework

- The run-time execution framework distributes and executes the rules and policies that are designed within the design time framework, and consists of the following subsystems:
- Active and Available Inventory (AAI): - Active and Available Inventory (AAI) is the ONAP subsystem that provides real-time views of available Resources and Services and their relationships. AAI (sometimes referred to as A&AI) 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.
- Controllers: - A Controller manages the state of a single Resource (Application or Network). It executes the Resource's configuration and instantiation, and is the primary agent in ongoing management, such as control loop actions, migration, and scaling.
- Dashboard: - The ONAP Portal is a platform that provides the ability to integrate different ONAP applications into a centralized Portal Core.
- Data Collection, Analytics and Events (DCAE): - The Data Collection, Analytics, and Events (DCAE) subsystem, in conjunction with other ONAP components, gathers performance, usage, and configuration data from the managed environment.
- Master Service Orchestrator (MSO): - The Master Service Orchestrator (MSO) manages orchestration at the top level and facilitates additional orchestration that takes place within underlying controllers.
- ONAP Optimization Framework (OOF): - The OOF provides a policy-driven and model-driven framework for creating optimization applications for a broad range of use cases.
- Security Framework: - There are two main aspects to security in relation to the OpenECOMP platform: security of the platform itself and the capability to integrate security into the cloud services. These cloud services are created and orchestrated by the OpenECOMP platform. This approach is referred to as security by design.

All these subsystems rely on Common Services to provide access control, logging, data management, and other support.

In Figure 2 above, we provide a functional view of the architecture, which highlights the role of key new components:
1. The Beijing release standardizes and improves northbound interoperability for the ONAP Platform using the External API component.
2. OOM provides the ability to manage cloud-native installation and deployments to Kubernetes managed cloud environments.
3. ONAP Common Services now manage more complex and optimized topologies. MUSIC allows ONAP to scale to multi-site environments to support global scale infrastructure requirements. The ONAP Optimization Framework (OOF) provides a declarative, policy-driven approach for creating and running optimization applications like Homing/Placement, and Change Management Scheduling Optimization.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 10, October 2020

DOI 10.17148/IJARCCE.2020.91007

4. Information Model and framework utilities have evolved to harmonize the topology, workflow, and policy models from a number of SDOs including ETSI NFV MANO, TM Forum SID, ONF Core, OASIS TOSCA, IETF and MEF.

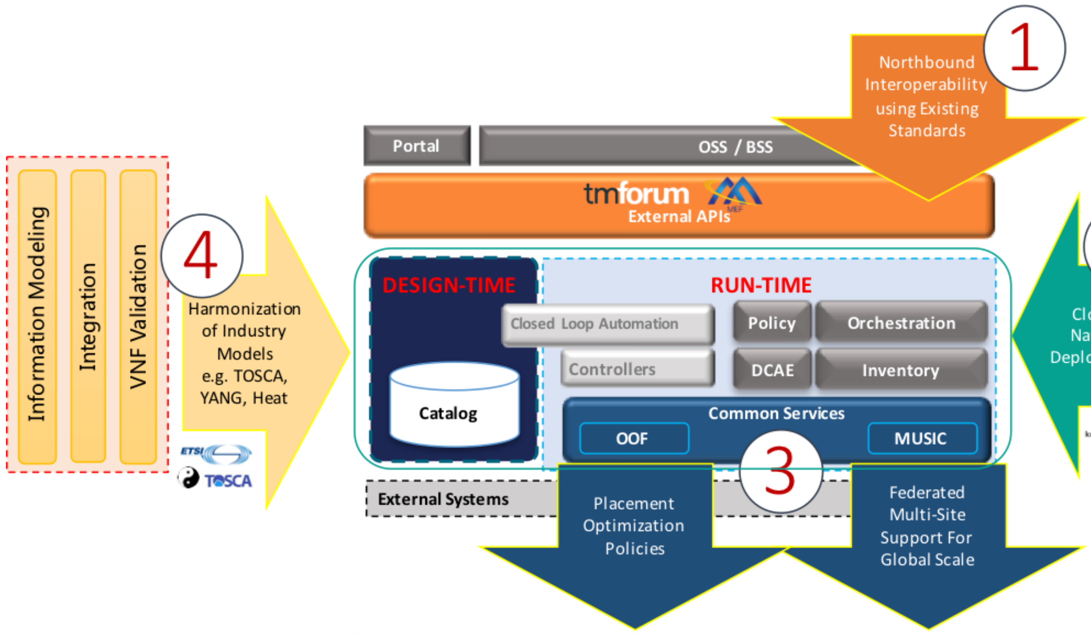

Figure 2. Functional view of the ONAP architecture

Hardware Components Required for ONAP

ONAP is deployed using the ONAP Operations Manager (OOM).

The recommended ONAP deployment is based on Kubernetes, Docker containers and Helm installer.

The following is the recommended component version:

Software	Version
Kubernetes	1.11.2
Helm	2.9.1
kubectl	1.11.2
Docker	17.03.x

Table 1. Recommended software version

The ONAP full installation is validated with the following footprint:

- 14 VM (1 Rancher, 13 K8s nodes) - 8 vCPU - 16 GB RAM
- 160 GB Storage

III. SOLUTION DETAILS

In this section we will be talking about the different components/technologies which we can use to implement ONAP.

1. ONAP- ONAP provides a comprehensive platform for real-time, policy-driven orchestration and automation of physical and virtual network functions that will enable software, network, IT and cloud providers and developers to rapidly automate new services and support complete lifecycle management.
2. OOM- The ONAP Operations Manager (OOM) is responsible for life-cycle management of the ONAP platform itself; components such as MSO, SDNC, etc.
3. Kubernetes- Kubernetes is a portable, extensible open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation.
4. Rancher- Rancher not only deploys Kubernetes clusters anywhere, on any provider, but it also unites them under centralized authentication and access control.
5. Docker- Docker is a computer program that performs operating-system-level virtualization. (same working as Kubernetes)

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 10, October 2020

DOI 10.17148/IJARCCE.2020.91007

Open Network Automation Platform (ONAP)

ONAP is addressing these challenges by developing global and massive scale (multi-site and multi-VIM) automation capabilities for both physical and virtual network elements. It facilitates service agility by supporting data models for rapid service and resource deployment, and providing a common set of Northbound REST APIs that are open and interoperable, and by supporting model driven interfaces to the networks. ONAP's modular and layered nature improves interoperability and simplifies integration, allowing it to support multiple VNF environments by integrating with multiple VIMs, VNFMs, SDN Controllers, and even legacy equipment. ONAP's consolidated VNF requirements publication will enable commercial development of ONAP-compliant VNFs. This approach allows network and cloud operators to optimize their physical and virtual infrastructure for cost and performance; at the same time, ONAP's use of standard models reduces integration and deployment costs of heterogeneous equipment, while minimizing management fragmentation.

The ONAP platform allows end user organizations and their network/cloud providers to collaboratively instantiate network elements and services in a dynamic, closed control loop process, with real-time response to actionable events. In order to design, engineer, plan, bill and assure these dynamic services, there are three major requirements:

- A robust design framework that allows specification of the service in all aspects – modelling the resources and relationships that make up the service, specifying the policy rules that guide the service behaviour, specifying the applications, analytics and closed control loop events needed for the elastic management of the service.
- An orchestration and control framework (Service Orchestrator and Controllers) that is recipe/policy-driven to provide automated instantiation of the service when needed and managing service demands in an elastic manner.
- An analytic framework that closely monitors the service behaviour during the service lifecycle based on the specified design, analytics and policies to enable response as required from the control framework, to deal with situations ranging from those that require healing to those that require scaling of the resources to elastically adjust to demand variations.

To achieve this, ONAP decouples the details of specific services and technologies from the common information models, core orchestration platform, and generic management engines (for discovery, provisioning, assurance etc.). Furthermore, it marries the speed and style of a DevOps/NetOps approach with the formal models and processes operators require to introduce new services and technologies. It leverages cloud-native technologies including Kubernetes to manage and rapidly deploy the ONAP platform and related components. This is in stark contrast to traditional OSS/Management software platform architectures, which hardcoded services and technologies, and required lengthy software development and integration cycles to incorporate changes.

The ONAP Platform enables product/service independent capabilities for design, creation and lifecycle management, in accordance with the following foundational principles:

- Ability to dynamically introduce full-service lifecycle orchestration (design, provisioning and operation) and service API for new services and technologies without the need for new platform software releases or without affecting operations for the existing services
- Carrier-grade scalability including horizontal scaling (linear scale-out) and distribution to support large number of services and large networks
- Metadata-driven and policy-driven architecture to ensure flexible and automated ways in which capabilities are used and delivered
- The architecture shall enable sourcing best-in-class components
- Common capabilities are 'developed' once and 'used' many times
- Core capabilities shall support many diverse services and infrastructures
- The architecture shall support elastic scaling as needs grow or shrink

ONAP Operations Manager (OOM)

The ONAP Operations Manager (OOM) is responsible for life-cycle management of the ONAP platform itself; components such as MSO, SDNC, etc. It is not responsible for the management of services, VNFs or infrastructure instantiated by ONAP or used by ONAP to host such services or VNFs. OOM uses the open-source Kubernetes container management system as a means to manage the Docker containers that compose ONAP where the containers are hosted either directly on bare-metal servers or on VMs hosted by a 3rd party management system. OOM ensures that ONAP is easily deployable and maintainable throughout its life cycle while using hardware resources efficiently.

In summary OOM provides the following capabilities:

1. Deployment - with built-in component dependency management (including multiple clusters, federated deployments across sites, and anti-affinity rules)
2. Configuration - unified configuration across all ONAP components
3. Monitoring - real-time health monitoring feeding to a Consul UI and Kubernetes

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 9, Issue 10, October 2020
DOI 10.17148/IJARCCE.2020.91007

4. Heal- failed ONAP containers are recreated automatically
5. Clustering and Scaling - cluster ONAP services to enable seamless scaling
6. Upgrade - change-out containers or configuration with little or no service impact
7. Deletion - clean-up individual containers or entire deployments
8. OOM supports a wide variety of cloud infrastructures to suit your individual requirements.

Kubernetes

Kubernetes is a portable, extensible open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available.
Google open-sourced the Kubernetes project in 2014. Kubernetes builds upon a decade and a half of experience that Google has with running production workloads at scale, combined with best-of-breed ideas and practices from the community.

Kubernetes is not a traditional, all-inclusive PaaS (Platform as a Service) system. Since Kubernetes operates at the container level rather than at the hardware level, it provides some generally applicable features common to PaaS offerings, such as deployment, scaling, load balancing, logging, and monitoring. However, Kubernetes is not monolithic, and these default solutions are optional and pluggable. Kubernetes provides the building blocks for building developer platforms, but preserves user choice and flexibility where it is important.

Kubernetes:

- Does not limit the types of applications supported. Kubernetes aims to support an extremely diverse variety of workloads, including stateless, stateful, and data-processing workloads. If an application can run in a container, it should run great on Kubernetes.
- Does not deploy source code and does not build your application. Continuous Integration, Delivery, and Deployment (CI/CD) workflows are determined by organization cultures and preferences as well as technical requirements.
- Does not provide application-level services, such as middleware (e.g., message buses), data-processing frameworks (for example, Spark), databases (e.g., mysql), caches, nor cluster storage systems (e.g., Ceph) as built-in services. Such components can run on Kubernetes, and/or can be accessed by applications running on Kubernetes through portable mechanisms, such as the Open Service Broker.
- Does not dictate logging, monitoring, or alerting solutions. It provides some integrations as proof of concept, and mechanisms to collect and export metrics.
- Does not provide nor mandate a configuration language/system (e.g., jsonnet). It provides a declarative API that may be targeted by arbitrary forms of declarative specifications.
- Does not provide nor adopt any comprehensive machine configuration, maintenance, management, or self-healing systems.

Additionally, Kubernetes is not a mere orchestration system. In fact, it eliminates the need for orchestration. The technical definition of orchestration is execution of a defined workflow: first do A, then B, then C. In contrast, Kubernetes is comprised of a set of independent, composable control processes that continuously drive the current state towards the provided desired state. It shouldn't matter how you get from A to C. Centralized control is also not required. This results in a system that is easier to use and more powerful, robust, resilient, and extensible.

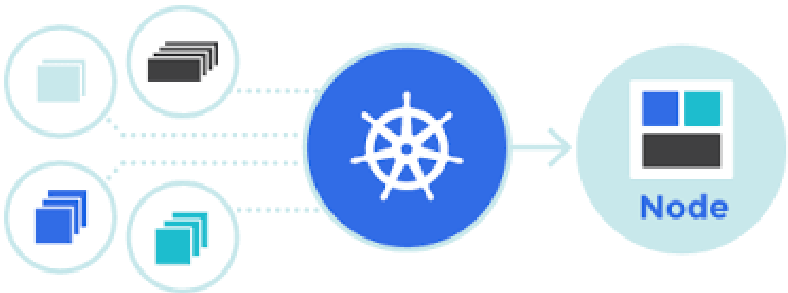

Figure 3. Working of Kubernetes

Rancher

The people who depend on you demand that your services stay up, no matter what. The best way to ensure you meet demand is to deploy services in multiple regions, on multiple providers. You'll need an efficient and reliable platform, built to manage multiple Kubernetes clusters in production.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 9, Issue 10, October 2020
DOI 10.17148/IJARCCE.2020.91007

Rancher not only deploys Kubernetes clusters anywhere, on any provider, but it also unites them under centralized authentication and access control. Because it's agnostic about where the resources run, you can easily bring up clusters in a different provider and deploy applications to all of them. Instead of having several independent Kubernetes deployments, Rancher unifies them as a single, managed Kubernetes Cloud.

Kubernetes is not without its challenges, and keeping a tight reign on one cluster can strain operations teams. Keeping tabs on more than one can overwhelm them. It's easy for Kubernetes to introduce excess complexity to your software. As more vendors deploy solutions around Kubernetes, avoiding lock-in will only become more difficult.

Rancher solves these problems. Out of the box it deploys Prometheus and Grafana for detailed visibility into cluster and workload metrics. It makes it dead simple for new Kubernetes users to deploy workloads while still keeping all of the traditional access channels for power users to leverage. It unifies every cluster under a centralized authentication provider. It makes best practices into easy practices, and this leads to systems which are more secure, teams who are more productive, and the flexibility for a business to pivot to wherever the market may turn.

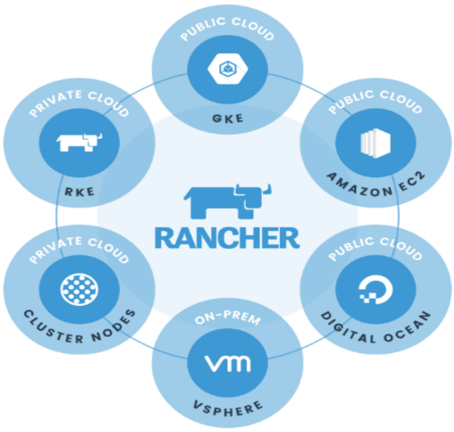

Figure 4. Working of Rancher

IV. USE CASES

In this section we will be talking about use cases such as: -

1. VNF- Network functions virtualization is a network architecture concept that uses the technologies of IT virtualization to virtualize entire classes of network node functions into building blocks that may connect, or chain together, to create communication services.
2. 5G- 5G is the latest generation of cellular mobile communications. It succeeds the 4G, 3G and 2G systems. 5G performance targets high data rate, reduced latency, energy saving, cost reduction, higher system capacity, and massive device connectivity.

Virtual Network Function (VNF)

Network functions virtualization (NFV) sometimes goes by another name in the industry — virtual network function (VNF). Often used interchangeably, both focus primarily on optimization of the network services, contrary to software-defined networking (SDN), which separates the control and forwarding plane for a centralized view of the network. Virtual network function is designed to consolidate and deliver the networking components necessary to support a full virtualized environment.

However, in an NFV environment, a virtual network function (or VNF) takes on the responsibility of handling specific network functions that run on one or more virtual machines (VMs) on top of the hardware networking infrastructure — routers, switches, etc. Individual virtual network functions can be connected or combined together as building blocks to offer a full-scale networking communication service.

Virtual network function came to fruition when service providers attempted to speed up deployment of new network services in order to advance their revenue and growth plans. They soon discovered that hardware-based appliances limited their ability to achieve these goals. They looked to standard IT virtualization technologies and found virtual network function helped accelerate service innovation and provisioning. With this, several providers banded together and created

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 10, October 2020

DOI 10.17148/IJARCCE.2020.91007

the European Telecommunications Standards Institute (ETSI). The creation of ETSI resulted in the foundation of virtual network function's basic requirements and architecture.

ETSI continues to innovate virtual network functions with new projects. Announced in September 2014, the Linux Foundation announced its open source reference platform, the Open Platform for NFV Project (OPNFV). OPNFV will work closely with ETSI to push for consistent open standards for NFV based VNFs.

Benefits of using VNFs

Traditionally, new services and network functions are installed manually, configured along with their dedicated hardware devices or boxes. But with service chaining, for example, if certain functions need to be linked to perform a desired sequence, each dedicated device needs to be manually cabled together accordingly. Since VNFs virtualize those functions, eliminating the need for specific hardware, new functions can be deployed as VMs more quickly.

VNFs can help increase network scalability and agility, while also enabling better use of network resources. Other benefits include reducing power consumption and increasing available physical space, since VNFs replace physical hardware. These benefits also result in reduced operational and capital expenditure.

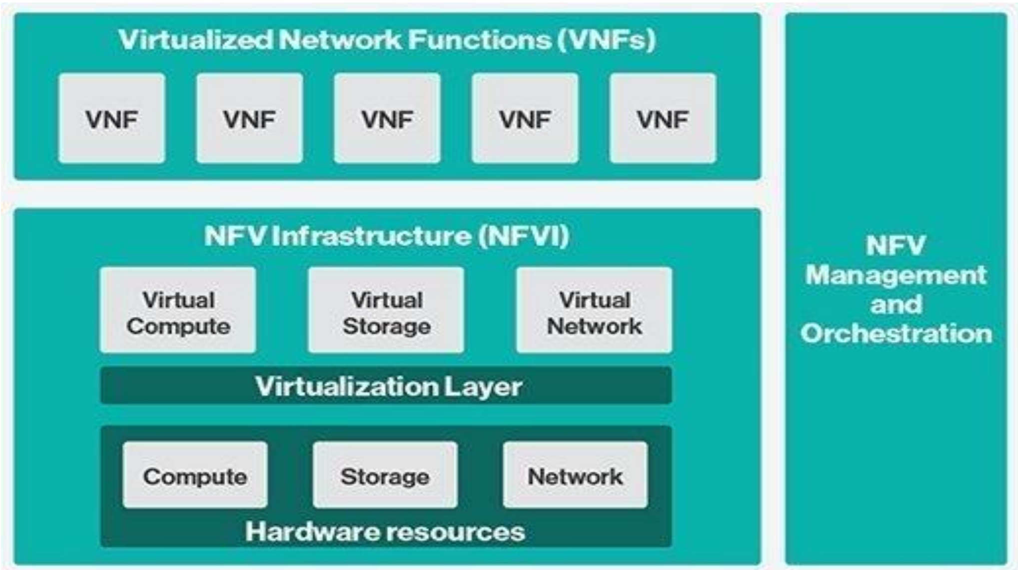

Figure 5. NFV v/s VNF

5G

General Description

The purpose of this document is to present a high-level use case for the deployment and optimized management of a 5G network while using the slicing concept for management of shared resources using ONAP. While the intent of the use case is the management of the overall 5G network, the initial focus will be on the RAN. The use case describes a Service Provider (SP) need to deploy a disaggregated 5G Radio Access Network (3GPP 5G Option 2-2 configuration). Some of the disaggregated network functions are expected to be virtualized (VNF), running on a cloud infrastructure and while others will be PNF (e.g. appliance-based peripherals).

The network elements in scope for such a disaggregated 5G RAN (in this release) are:

- Distributed Radio Element
- Distributed BBU
- Centralized BBU and nrt-L2 function (CU-UP)
- Centralized Radio Control Unit (CU-CP)

This disaggregation can include moving processing closer to the edge and require deployment of Multi-access Edge Computing (MEC) components and applications in order to meet 5G performance goals.

ONAP should support the complete lifecycle management of this 5G Radio Access Network, using Service Design and Creation (SDC) for design and onboarding of the various models and artifacts for physical and virtual network functions, including creation of recipes/descriptors and policies for their initial deployment, and associated transport (front/mid/back-haul) connectivity. Further, to support the SP needs for lifecycle management of the shared resources,

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 9, Issue 10, October 2020
DOI 10.17148/IJARCCE.2020.91007

ONAP should support the data collection and analysis, policies and analytics to identify actionable conditions, and support automatic closed loop actions for the RAN deployment, optimization and slicing management.

The use case is divided into three sub cases that are described in the following sections:
- Disaggregated RAN Deployment
- RAN Slicing
- RAN Optimization

The description of the use cases will address at least 3 different ONAP related roles:
- VNF/PNF Provider: The supplier of the network/service function (and its underlying infrastructure) – e.g. VNF/PNF vendors; the VNF/PNF provider will supply the artifacts necessary for the on-boarding into ONAP (this includes both the VNF and PNF that are supplied)
- ONAP Service Designer: The user of ONAP design studio that is used to on-board the VNF/PNF and who defines, models the workflows, analytics, policies and extensions needed for management & automation (i.e. making the RAN operational)
- Service/Network Operator: The ONAP operations user (operator of the realized service managed by ONAP) that processes or consumes the various dashboards/reports/logs and is responsible for various management functions (change/configuration etc.); the Operations user either directly (via a portal) or indirectly (via system APIs) enables the initial deployment of resources and services that make the RAN operational

Overview

The move from 4G to 5G has already started. The true promise of 5G is not just the 50x more speed, 10x lower latency, and 1000x more capacity, but that it will dynamically support multiple different services concurrently.

This combination of capabilities will transform the global economy by unlocking new use cases such as immersive media, autonomous vehicles, smart factories/cities/buildings, connected health, next generation education, and others. Unlike previous generations, 50% of the data generated in 5G will be through IoT devices. With these new 5G applications, ABI Research predicts that the total 5G economic output by 2035 will be $10T.

Figure 6. Mobile Network Data by 2020 For Sample Applications

Some key technologies in 5G are:
- eMMB (enhanced mobile broadband): Promises to provide broadband capability to users with a peak data rate of 20 Mbps enabling services such as immersive video.
- uRLLC (ultra-reliable low-latency communications for remote device): Guarantees sub millisecond response times enabling services such as industry 4.0, remote surgery, AR/VR, and rescue and smart car applications.
- MMTC (massive machine-type communications): Supports 1 million devices per km2 enabling applications such as autonomous vehicles, M2M applications, and IoT.

These new use cases and technologies bring with them a level of dynamic network behaviour unlike previous generations of wireless technologies.

Problem

The level of dynamic behaviour stems from different applications requiring different levels of latency, reliability, availability, mobility, bandwidth, and cost. For example:

Table 2. Different materials

Autonomous vehicle	Low latency, high mobility
IoT	Low cost, low bandwidth
Factories	High reliability, low mobility
Video	High bandwidth, high latency

The core technology that supports these diverse requirements is called network slicing, that per 5G.co.uk is the ability to "provide dedicated virtual networks with functionality specific to the service or customer over a common network infrastructure". Furthermore, the environment becomes even more dynamic with the need to deliver edge computing applications to subscribers on-demand.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 9, Issue 10, October 2020
DOI 10.17148/IJARCCE.2020.91007

To support 5G networks, network automation becomes a key consideration, and the following new requirements emerge for the management and orchestration (MANO) layer:

1. Hybrid 5G radio network: 5G will require some parts of the radio network to be implemented in the form of Physical Network Functions (PNFs). This means that a 5G network services will need to support both PNFs and virtual network functions (VNFs3).
2. Edge Automation: Traditional networks services and their constituent VNFs have been deployed in big datacentres. With 5G, centralization of radio processing and decentralization of core/edge application processing is expected to create a large number of highly distributed edge locations.
3. Real-time analytics: 5G networks will need to optimize themselves in real-time in response to subscriber requests and network behaviour. This means that edge locations will need to perform real-time analytics to influence the lifecycle management actions such as scaling, fault management, performance optimization, and others.

Solution

The Open Network Automation Platform (ONAP) project automates 5G using software defined networking (SDN) and network functions virtualization (NFV) technologies. ONAP is an open source project that provides a common platform for telecommunications, cable and cloud operators and their solution providers to rapidly design, implement and manage differentiated services. ONAP provides orchestration, automation and end-to-end lifecycle management of network services. It includes all the Management and Orchestration (MANO) layer functionality specified by the ETSI NFV architecture; additionally, it provides a network service design framework and FCAPS (fault, configuration, accounting, performance, security) functionality.

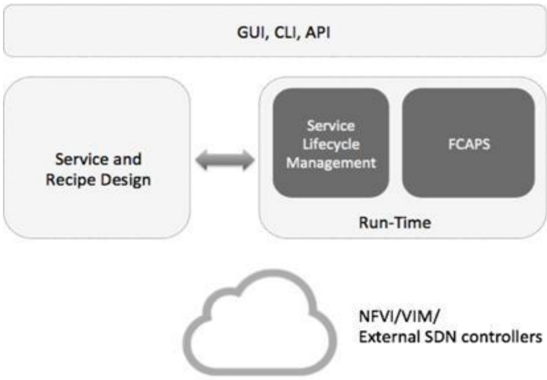

Figure 7. ONAP Functionality

5G automation is a complex topic that will be fully addressed over multiple ONAP releases. The Casablanca release enables an initial set of 5G features and sets the overall direction for subsequent releases. The major 5G related ONAP initiatives include:

- PNF discovery & integration
- Network slicing (Not included in Casablanca, planned as a roadmap item)
- 5G network optimization

PNF integration

With 5G, CSPs need to deploy disaggregated 5G Radio Access Network (RAN) elements (3GPP 5G Option 2-2 configuration). Some of these network functions are virtualized, running on an edge cloud infrastructure while others are appliance-based or PNFs. ONAP lays the groundwork to support the complete lifecycle management of PNFs. The various steps to support PNFs are:

- Design time
  - PNF modeling
- Runtime
  - PNF instance declaration
  - PNF bootstrapping
  - PNF registration
  - PNF activation

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 9, Issue 10, October 2020
DOI 10.17148/IJARCCE.2020.91007

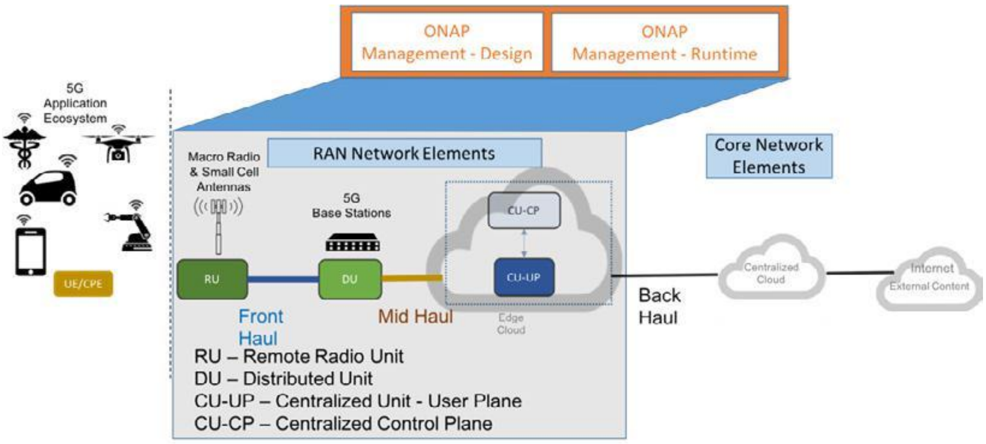

Figure 8. Disaggregated Hybrid RAN

Network Slicing

The Casablanca release does not include any Network Slicing functionality yet, but rather lays the foundation for subsequent releases by kicking off modeling work; nevertheless, this section is included for completeness.

As discussed above, different 5G services require different latency, reliability, availability, mobility, bandwidth, and cost requirements. End-to-end network slicing as defined by 3GPP provides specifications for creating efficient multiple logical networks using common network infrastructure that can meet requirements for each of the services. Over subsequent releases, ONAP will support the complete orchestration and lifecycle management of such 5G E2E network slicing. To accomplish this, ONAP will need to support slice design, orchestration, configuration, automatic re-configuration due to requirement or condition changes, transport network configuration, and vEPC configuration.

5G Network Optimization

A CSP will need to, in real-time, optimize the performance of the 5G service. This optimization will require dynamic configuration of relevant 5G radio and backhaul network parameters. To date, optimization functions have been realized in 3G and 4G networks via vendor-proprietary hardware and software. ONAP will enable the design and implementation of an open ecosystem for 5G optimization across multiple releases.

The list of network optimization ONAP features, current and future, is:

- High Volume Performance Management: 5G requires real-time analytics for time sensitive Performance Management (PM) data delivered at frequent intervals (less than one minute) from a large number of edge locations. This analytics data is used to drive network and customer experience optimization. Until 4G, optimization algorithms have resided in network elements; ONAP will change that by absorbing these algorithms to unburden xNFs. This analytics capability will ultimately lead to AI/ML based algorithms that will fully automate the network.
- Bulk Analytics: 5G also requires batch processing of bulk PM data delivered less frequently (say every 5-15 minutes) for optimization purposes.
- Homing: ONAP will find the best edge location for a given workload. Workloads include VNFs, edge analytics, and possibly edge computing (MEC4) applications. Homing includes considerations such as Physical Cell ID (PCI) and RF optimization. ONAP also allows homing policies based on hardware platform awareness.
- Scaling & Healing: With the sheer increase in edge locations, it becomes important to automatically scale-in/out Centralized Units (CU) and other VNFs to provide a balanced network with the right amount of capacity. Self-healing goes hand-in-hand with scaling.
- Edge Automation: For all of the above reasons, it is critical to onboard and register edge compute locations with ease.

In addition to these three major initiatives, the community has also been active in-service definition and management, dynamic network inventory management, and harmonizing model definitions with 3GPP and ETSI to allow vendors to easily support multiple platforms.

Implementation Details

ONAP Casablanca provides the first installment of 5G functionality with the following key features.

- The ONAP Service Orchestrator (SO), various controllers (VF-C, APP-C), Active & Available Inventory project (A&AI), Data Collection Analytics and Events (DCAE), and VID (Virtual Infrastructure Deployment) projects support PNF registration and PNF discovery, orchestration, lifecycle management, and monitoring.

ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Vol. 9, Issue 10, October 2020

DOI 10.17148/IJARCCE.2020.91007

- A new high volume VES collector that uses Google Protocol Buffers encoding over TCP/TLS allows, ONAP DCAE to now support real-time PM data collection. DCAE also supports File collection to ingest bulk PM data.
- The MultiCloud project has tighter integration with SO simplifying onboarding of edge clouds.
- The OOF project has a number of homing improvements to support 5G requirements such as PCI optimization and Hardware Platform Awareness (HPA).
- The Policy project supports requirements such as PCI control loop.
- The modeling project has made advances in PNF Descriptor Resource Information Model.

By utilizing these features, the ONAP community has shown five demonstrations:

1. PNF support using a 5G Distributed Unit (DU) emulator
2. PNF software upgrade using an external Element Management System (EMS)
3. Real-time high-volume PM data collection
4. Bulk PM data collection
5. Homing optimization for PCI (a SON Self-Optimization Use Case)

V. CONCLUSION

ONAP technology was successfully implemented. Following are the conclusions of the study:

1. VNF was implemented in the network
2. 5G networking was configured according to use and implemented
3. The network has been constantly updated as the technology evolving

ACKNOWLEDGMENT

I would like to thanks Mr. Krishan Dhall and Mr. Shijo Joseph of M/s Wipro Ltd for guiding me to implement ONAP technology.

REFERENCES

1. V. Q. Rodriguez, F. Guillemin and A. Boubendir, "5G E2E Network Slicing Management with ONAP," 2020 23rd Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), Paris, France, 2020, pp. 87-94, doi: 10.1109/ICIN48450.2020.9059507.
2. F. Slim F. Guillemin A. Gravey and Y. Hadjadj-Aoul "To-wards a dynamic adaptive placement of virtual network functions under ONAP" Third International '17- O4SDI-Workshop on Orchestration for Software-Defined Infrastructures) 2017.
3. "Network Functions Virtualisation (NFV). Management and Orchestration" Technical Report ETSI GS NFV-MAN 001 V1.1.1 2014.
4. Girma M. Yilma,Zarrar F. Yousaf,Vincenzo Sciancalepore,Xavier Costa-Perez, Benchmarking open source NFV MANO systems: OSM and ONAP, Computer Communications, Volume 161, 1 September 2020, Pages 86-98.
5. ONAP main site: https://www.onap.org/, Accessed: 15/01/2019
6. HELM installation: https://docs.helm.sh/using_helm/#installing-helm, Accessed: 03/03/2020
7. OOM installation: https://docs.onap.org/en/casablanca/submodules/oom.git/docs/oom_quickstart_guide.html, Accessed: 15/01/2019
8. Kubernetes and Rancher installation: https://docs.onap.org/en/casablanca/submodules/oom.git/docs/oom_setup_kubernetes_rancher.html#onap-on-kubernetes-with-rancher, Accessed: 21/02/2020
9. openstack.-rancher.sh: https://docs.onap.org/en/casablanca/_downloads/0b365a2342af5abd655f1724b962f5b5/openstack-rancher.sh, Accessed: 30/04/2020
10. openstack-k8-node.sh: https://docs.onap.org/en/casablanca/_downloads/b20b581d56982e9f15a72527a358d56b/openstack-k8s-node.sh, Accessed: 26/06/2020
11. master_nfs_node.sh: https://docs.onap.org/en/casablanca/_downloads/305eb4622e94c72dd5c2300d21e49806/master_nfs_node.sh, Accessed: 01/09/2020
12. slave_nfs_node.sh:https://docs.onap.org/en/casablanca/_downloads/91443f8e0ad897ea9d7fb384edd29153/slave_nfs_node.sh,Accessed: 19/09/20

BIOGRAPHY

B. Tech in Computer Science and Engineering (2015-2019) from SRM Institute of Science and Technology, Chennai, India. Currently working as Program Manager at De Pedagogics, Gurgaon, India.