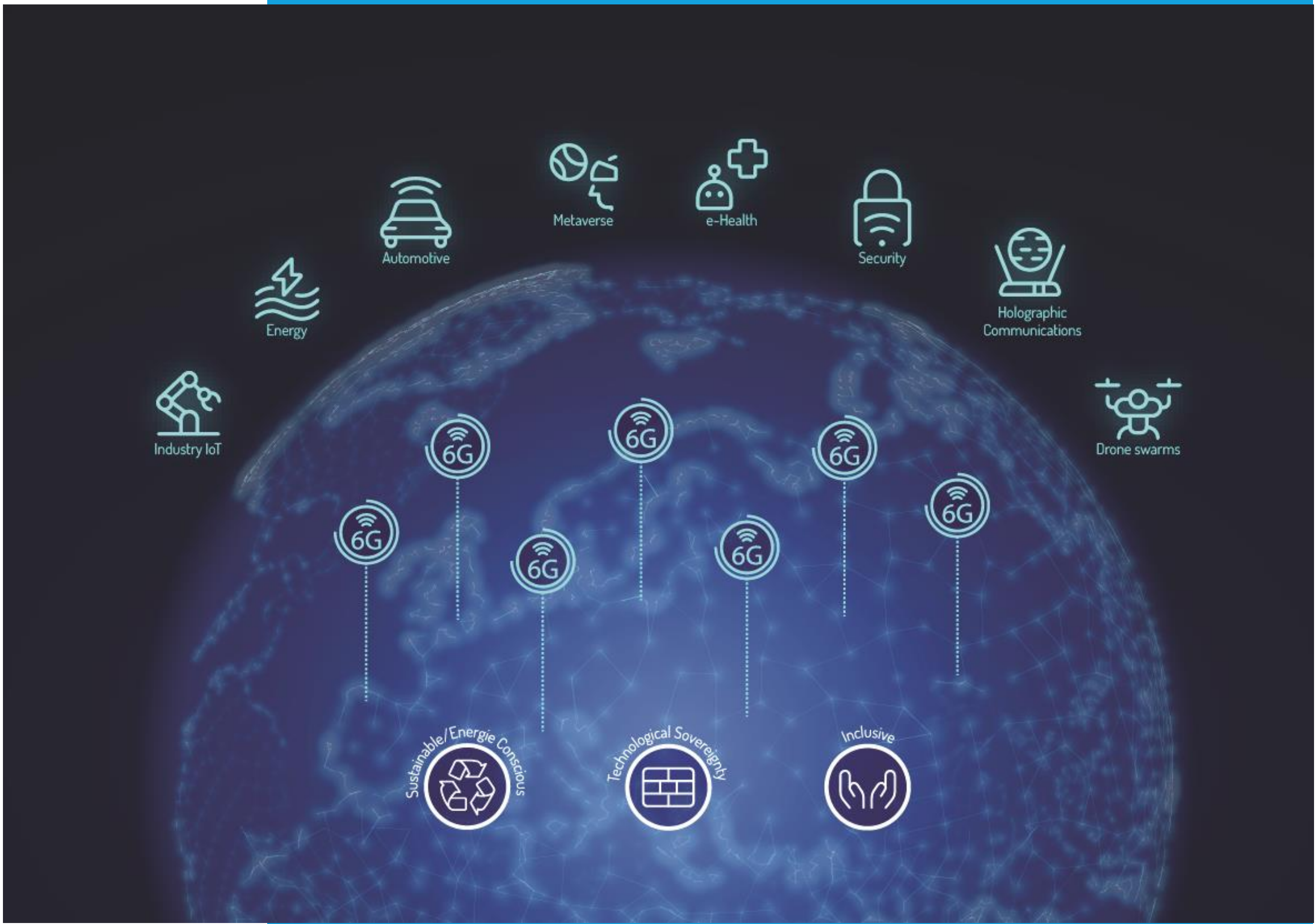
6G Smart Networks and Services Industry Association (6G-IA)



Version 1.0

December 2023

Open Networks and Services Whitepaper

Platforms Subdomain

DOI:
URL:

C2 General

EXECUTIVE SUMMARY

While 5G development is ongoing and maturing, 6G development has started with progress on various aspects of the system and performance. In general, one of 5G and 6G's major developments will be the move towards more open technologies. One main driver for this is the need for tighter integration of the 6G system with other technologies and domains. The interaction will borrow from internet success stories such as cloud computing, web technologies, and scalable design. Open technologies should therefore be focused on cloud systems and platform interconnects as those have the highest impact on end users.

Open platforms run and execute network functions, application functions, and workloads. Making their execution locations/platforms open is a critical aspect of the end-to-end system. The use of open platforms will overcome the barrier for fast and secure deployment of new services by new operators and cloud systems. Even though there are many prominent open platform systems, the area is still an ongoing development and more research and engagement from the different vendors and operators is needed to achieve widespread adoption.

This white paper provides an overview of prominent open platforms that are needed to run and execute network functions and next generation workloads. Further, it provides a taxonomy and classifications of their openness, features, and complexity with regards to their different characteristics and readiness to run 6G workloads and systems.

We provide recommendations to push vendors as well as operators to use open platforms to enable cross compatibility, energy conservation, resource efficiency, as well as application/data readiness of the network. Open Platforms will be important for 6G applications, since it will offer more cloud-like execution of new applications requiring stringent requirements as well as low cost as well as ease the transition from 5G to 6G.

Finally, Open platforms are an important aspect to achieve EU readiness and independence for next generation communication systems.

TABLE OF CONTENTS

1. OPEN PLATFORMS

1.1. OVERALL SCOPE

This whitepaper focuses on open platforms for realizing open systems and networks. The term platform is quite general and the first thing to do is try to define it and to limit its scope.

1.2. DEFINITION OF OPEN PLATFORM

On the one hand, a platform is a generic term that is usually used in the datacenter space to define the underlying hardware and software substrate that is used to run specific services and applications. Of course, different systems and architectures define a platform differently. To try to be as generic as possible yet specific, we define a platform as the interacting constellation of a specific set of resources that enables the execution of a certain service using well defined interfaces, tools, and methods.

On the other hand, openness is a general term that captures a certain property of being transparent and/or accessible from outside. In general openness can be related to any specific aspect of the architectural decision. This can include open access to the source code of the platforms, or it could be the availability of specific Application Programmable Interfaces (APIs) that can be used to access the services, or can simply refer to the possibility of anyone or any institution to use a certain service. Indeed openness can cover different aspects as we will discuss later. That is why it is important to create a certain classification of different dimensions of openness and try to provide the general approach for the different paltforms.

2. BENEFITS AND CHALLENGES OF OPEN PLATFORMS

In this chapter we go over the benefits and challenges of open platforms.

2.1. BENEFITS

Platforms are gaining more importance recently mainly due to the fact that they enable execution of more types of services. This ranges from compute, to network, and storage services and applications. What makes platforms important is that they enable running softwarized services that can be adapted and changed depending on the requirements of the services. By assuming a generic set of off-the-shelf underlying resources, it is possible to enhance compatibility and cross interaction between different software tools. Benefits in this context also include the possibility to use an open market of software and function providers that will add value to the network operators. In addition, any developer for a classical cloud system will be able to use his/her knowledge within the mobile system and deploy new innovative services that include both compute and communication in one service.

2.2. CHALLENGES

In general, a major challenge for open platforms is the dominance of major cloud providers for hosting and executing applications. This puts pressure on mobile networks to act as mere pipes for data traffic. What makes the problem more critical, is that most internet traffic is end-to-end encrypted, therefore mobile operators are not able to monetize the new types of hosting and added value applications. Further, there are no open interfaces to interact with those cloud providers, making any end-to-end interaction optimization of resources very difficult. Additional challenges in this context include the high cost for using open platforms since this includes a learning curve for operators and vendors. The operators need to hire specialists so that they can offer guidance and support. However, as we show below, such an investment is not only necessary but could potentially open up new business potential for operators and put them in the driving seat for enabling new types of services and applications.

2.3. OPPORTUNITIES

Major opportunities in this context relate to the possibility for telco operators to ride the wave of cloudification and embarrass this technology in the design and deployment. By having the possibility to use open platforms and standard APIs for the interaction, it is possible to reduce costs, enhance performance, and enable a more agile network rollout. Indeed, it is important for telco operators to use such techniques to move beyond best effort networks in general and to offer added services and functionalities not supported by classical telco platforms.

3. PLATFORM TYPES AND CLASSIFICATION

In this chapter, we explain the methodology for working with platforms. We will explain their scope, and different types of platforms, as well as a possible taxonomy that defines the important aspects of open platforms. The purpose is to provide vendors, operators, and system developers with the means to assess what different open platforms are and if they fit the needs of the executing party.

3.1. PLATFORM TYPES

There are different types of platforms to consider. In general, and since execution platforms can be provided by different vendors: such as hardware vendors, software vendors, and application vendors, it is important to present the different types of a platform. Finally, we show other ways to use platforms using platforms as a service.

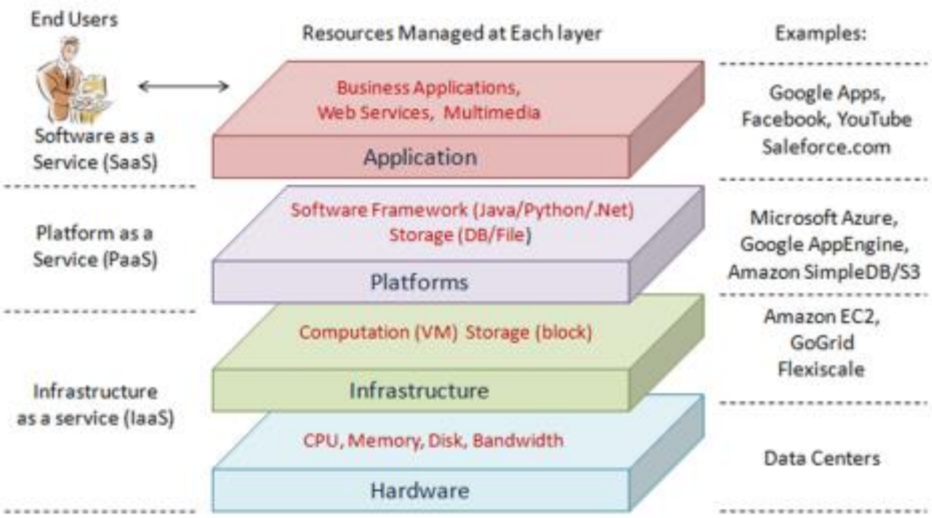


Figure 1 Cloud computing architecture

3.1.1. HARDWARE PLATFORMS

The term "hardware platform" describes all the physical components that support the operation of a function. For example, a computer system is composed of hard disks, peripherals (like mouse, screen), mother boards, GPU, etc. All these components will enable the computer to accommodate services (that will be provided by software). Without these services, this computer is called a bare metal component to indicate that it contains the structure and the components but will need further software to be used.

These components are usually classified in compute, storage or network resources.

3.1.2. SOFTWARE PLATFORMS

Software platforms are a specific set of software tools that enable hosting of applications or mobile network functions on top of hardware resources. Most prominent examples are Kubernetes and OpenStack which will be covered in more detail in this document.

3.1.3. STANDARD PLATFORMS

A standard platform is a platform that follows an industrial standard in order to define specific interfaces and API when interacting with platforms. For example, the ETSI MANO has defined in [MANO] an architecture for management and orchestration. ETSI MEC defines the architecture for running edge computing applications. ETSI NFV defines the architecture for running network functions on top of virtualized infrastructures and platforms.

3.1.4. PLATFORM-AS-A-SERVICE

A platform can be offered as a service to customers/users.This enables many service providers to focus on the service and applications instead of having to deal with the complexity of platforms. In addition, platform providers can make use of economies of scale by offering the same platform for multiple services providers.

As per the NIST definition, we now cover the three building blocks of platforms: namely: software, platform, and infrastructure as a service.

Software as a Service (SaaS). The capability provided to the consumer is to use the provider's applications running on a cloud infrastructure . The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user specific application configuration settings.

Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming

languages, libraries, services, and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly configuration settings for the application-hosting environment.

Infrastructure as a Service (IaaS). The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls.

3.2. PLATFORM DOMAINS FOR EXECUTION

An actual platform could be used to run and execute specific functions. While we aim to encourage platforms that can execute in any domain, this is not always feasible. For instance, to run a RAN function on a certain platform, it is necessary that this platform supports specific RAN functionalities and features. We now go over specific execution domains and possibility some requirements for the platforms on which they will execute.

3.2.1. CORE NETWORK FUNCTIONS

The core network handles control of the mobile network to make sure UEs get allocated with resources so that they can send and receive data over the network. With the modern mobile network, network functions of the core network are nothing more than applications running as a container or virtual machine with specific interfaces exposed to other network functions. This made it possible for running core network functions just as applications, opening the possibility to use classical virtualisation and containerisation technologies, such as docker and kubernetes. In general this will allow operators for better utilization of resources as well as allow easier migrations and cross interaction.

3.2.2. RAN NETWORK FUNCTIONS

Even RAN network functions can be executed on top of standardized platforms. However, they require the availability of specific functionalities related to the radio transmission in order to function. This makes it more challenging to design open platforms for RAN function, since it might be vendor specific. Design open platforms, starters by separating the compute required for RAN functions from the actual data transmission and modulation functions. THis would allow them to harness off the shelf hardware and platform providers for execution while the actual radio transmission is handled by their specialized vendor. In addition, specific APIs and interfaces for controlling RAN interaction have to be developed.

3.2.3. TRANSPORT NETWORK FUNCTIONS

Transport networks are similar to RAN network functions in the sense that they require specific functionalities and underlying resources. This again follows the same recommendation that any compute/storage functionality of the transport network should be separated from the specifics of the underlying transport network in order to make use of generic platform tools and providers.

3.3. CLASSIFICATION/TAXONOMY DIMENSIONS

We now go over a classification and taxonomy of open platforms in order to understand the different defining dimensions that have to be considered.

3.3.1. OPEN PLATFORM CHARACTERISTICS AND FEATURES

Characteristic/Feature	Explanation
Open source	The open platform source code is available and can be audited. It is possible to verify that a certain binary application/platform corresponds to a certain source code version of the platform

Open access	Access to the platform is not restricted to any specific individual/country/etc…
Open API	The specific function calls that can be used to access a platform are well defined and documented enabling any user to develop own applications that uses a specific platform
Open characteristics	Specific characteristics of a platform are well documented, this can include but not limited to: energy requirements, achievable performance, availability, durability, etc…
Open governance	Controllability and manageability of the platform is possible independent of the vendor/operators

3.3.2. WHAT CAN PLATFORMS HOST?

As mentioned above, platforms can be used to host and execute different types of network functions.

Aspect	Description
Containerized applications	Applications that have been developed in Docker can be run and controlled and given access to the outside
Radio	Radio specific logic and code can be executed
Transport/Routing	Transport and routing applications can be executed, this can include specific control of tunneling and transport protocols
Core	Core Network functions can be executed as provided by the user

3.4. LIMITS BETWEEN SW/HW

The European commission is acting to support an "EU Chip act" to promote Europa as a leader for semiconductors. This action is in favor of components that are close to hardware. If the code running on these components can be loaded dynamically, it opens the way to "Open Hardware " and will enable High Performances and Acceleration of processing necessary for future URLLC 5G services.

4. PROMINENT OPEN PLATFORMS

In this chapter, we go over some of the most widely used open platforms for building and managing network infrastructures and services in different domains (e.g., RAN, core, edge, cloud). We classify these platforms into three categories based on their objectives and functionalities: resource orchestration platforms, SDN controller platforms and infrastructure management platforms.

4.1. RESOURCE ORCHESTRATION PLATFORMS

Resource orchestration platforms offer mechanisms to deliver full management of network service lifecycles. By leveraging cloud computing, NFV and SDN technologies, these platforms enable automatic deployment and operation of end-to-end communication services. Among the different functions performed by resource orchestration platforms, some key functions include resource provisioning, scaling and elasticity, deployment and configuration management, lifecycle management, monitoring and optimization. In general, such systems have been historically closed sources and controlled by large companies. With the movement towards more open sources, and the interest of different companies to realize solutions that integrate different techniques, new systems were developed and can be used. We now present the prominent resource orchestration platforms.

4.1.1. ONAP

Open Network Automation Platform (ONAP) is a platform for end-to-end orchestration, management, and automation of network and edge computing services for network operators, cloud providers, and enterprises. 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.

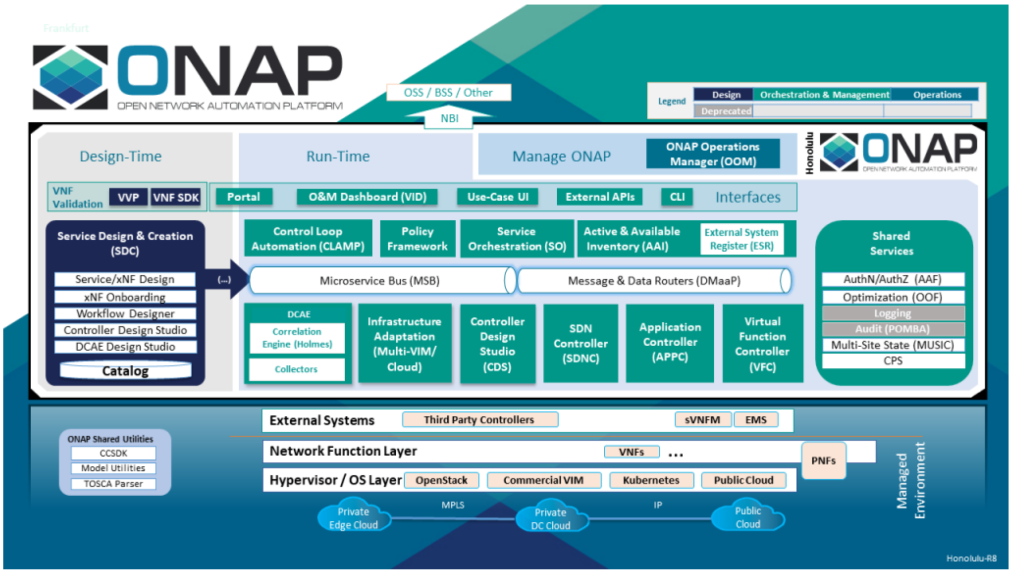


Figure 2 ONAP architecture

4.1.2. OSM

Open Source MANO (OSM) is an open source NFV management and orchestration platform hosted by ETSI. This platform was developed to align with the ETSI NFV information models and to meet production requirements. As a community-led project, OSM delivers a production-quality platform for the management and orchestration of network services in virtualized and cloud environments. The platform is designed to meet the needs of service providers, network operators, and enterprises that are looking to deploy and manage network functions and services in an efficient and scalable manner. OSM interacts with virtual infrastructure managers such as openstack and kubernetes. It can also be linked to SDN and SD WAN controllers such as opendaylight.

OSM offers the possibility to create and on-board Network functions, network services and network slices packages. It is also possible to use life cycle management primitives such as Network function scaling and healing. The development of OSM is an on-going process and there is still a lack of automation procedures and lack of global support to all the NFV/SDN requirements. However, this tool consumes much less resources and is more easy to install in contrast to the ONAP system.

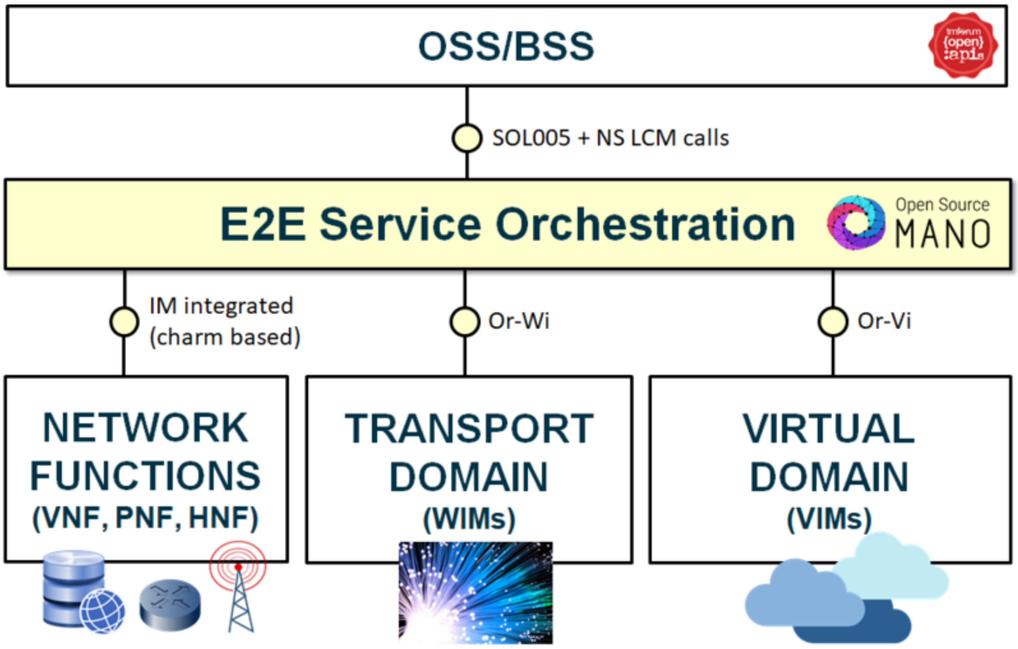


Figure 3 OSM architecture

4.1.3. CORD

CORD (Central Office Re-architected as a Datacenter) platform leverages SDN, NFV and Cloud technologies to build agile datacenters for the network edge. Integrating multiple open source projects, CORD delivers a cloud-native, open, programmable, agile platform for network operators to create innovative services.

CORD provides a complete integrated platform, integrating everything needed to create a complete operational edge datacenter with built-in service capabilities, all built on commodity hardware using the latest in cloud-native design principles.

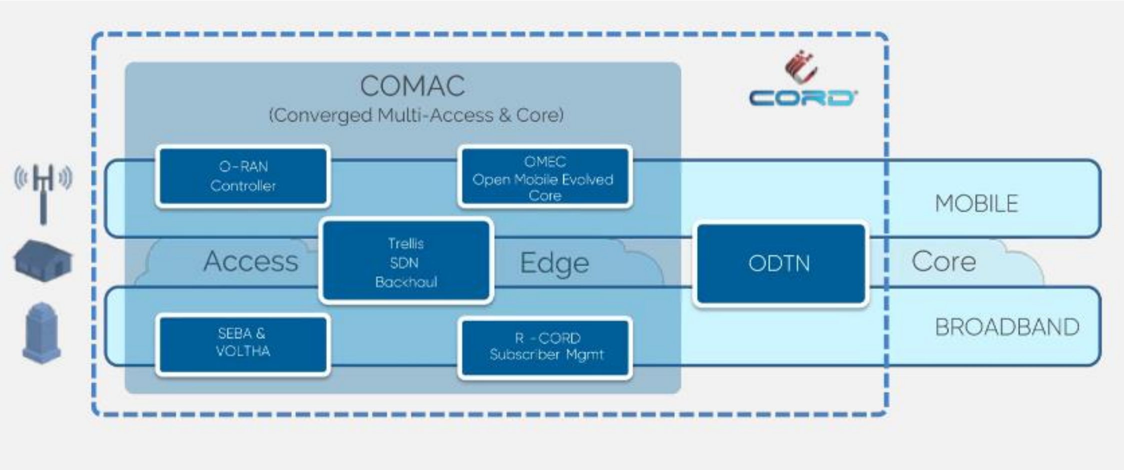


Figure 4 Cord architecture

4.1.4. OPENSHIFT

OpenShift Container Platform (formerly known as OpenShift Enterprise) is Red Hat's on-premises private platform as a service product, built around application containers powered by CRI-O, with orchestration and management provided by Kubernetes, on Red Hat Enterprise Linux and Red Hat Enterprise Linux CoreOS. OpenShift provides a platform for building, deploying, and managing applications across hybrid and multi-cloud environments.

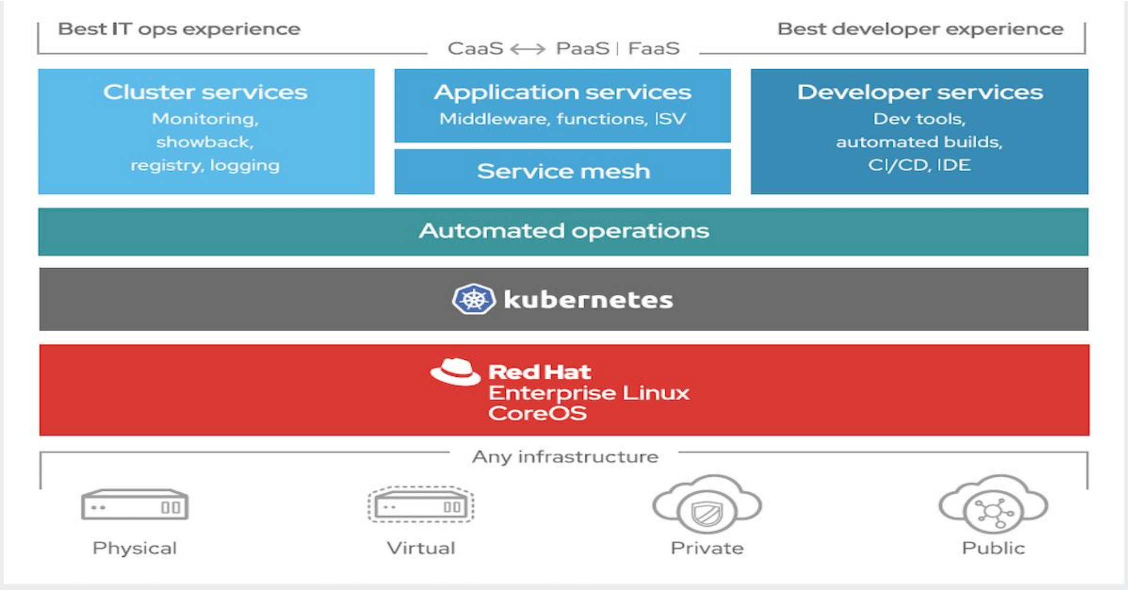


Figure 5 Open shift architecture

4.1.5. OPENBATON

OpenBaton is an open source implementation of the Network Function Virtualization Orchestrator (NFVO), which is based on the ETSI NFV MANO reference standard and the

OASIS TOSCA specification . The Fraunhofer Fokus Institute pioneered OpenBaton. The latest release, Release 4, includes many features that comply with the ETSI NFV MANO specification. These features include an NFVO for end-to-end service orchestration, a generic VNFM for multi-vendor infrastructure management, FCAPs, autoscaling, and event management engines. OpenBaton can orchestrate services across multiple administrative domains and can be run on different NFVI, including AWS, Openstack, Docker containers, and LXC containers. However, the OpenBaton community is small, which poses a threat to the project's maintainability and lifespan.

4.2. SDN CONTROLLER PLATFORMS

Software define network or SDN, is based on the premise that a network should be softwarized for easier and more efficient deployment and control. Many SDN controller technologies have been developed. They aim at making network control and management easier by allowing the routers and switches to be controllable using specific interfaces and APIs. We now cover specific SDN controller platforms that can be used to realize general open platforms.

4.2.1. ONOS

Open Network Operating System (ONOS) is a prominent open source SDN controller for building SDN/NFV solutions. ONOS has the target to build a platform to meet the needs of operators wishing to build carrier-grade solutions that leverage the economics of white box merchant silicon hardware while offering the flexibility to create and deploy new dynamic network services with simplified programmatic interfaces. ONOS supports both configuration and real-time control of the network, eliminating the need to run routing and switching control protocols inside the network fabric. By moving intelligence into the ONOS cloud controller, innovation is enabled and end-users can easily create new network applications without the need to alter the dataplane systems.

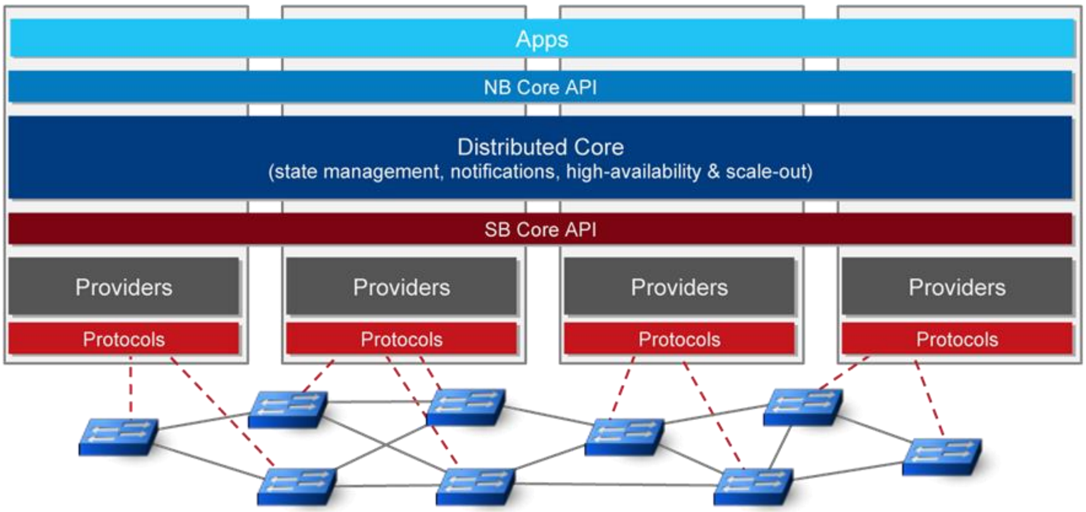


Figure 6 Architecture of ONOS

4.2.2. ODL

OpenDaylight (ODL) is an open-source SDN controller platform that is designed to enable the creation of network services and applications. It is a collaborative project of the Linux Foundation and is based on the Java programming language. It provides a wide range of SDN applications and services, including network virtualization, traffic engineering, security and service chaining. It also provides support for the OpenFlow protocol, which enables the management and configuration of network devices through a centralized controller. OpenDaylight has a large and active community of developers, users, and vendors, which contributes to the ongoing development and improvement of the platform. It is used by a wide range of organizations, including service providers, enterprises, and academic institutions, to build and manage their SDN infrastructures.

4.2.3. RYU

RYU is a platform for building SDN applications that provides useful libraries and well-defined API. It is written in python allowing it to be run on different environments. The developers claim it is vendor-neutral, supports open interface (eg., OpenFlow), and used by some switch vendors. It is targeted to be a framework for SDN application development instead of an all-purpose big monolithic “controller”.

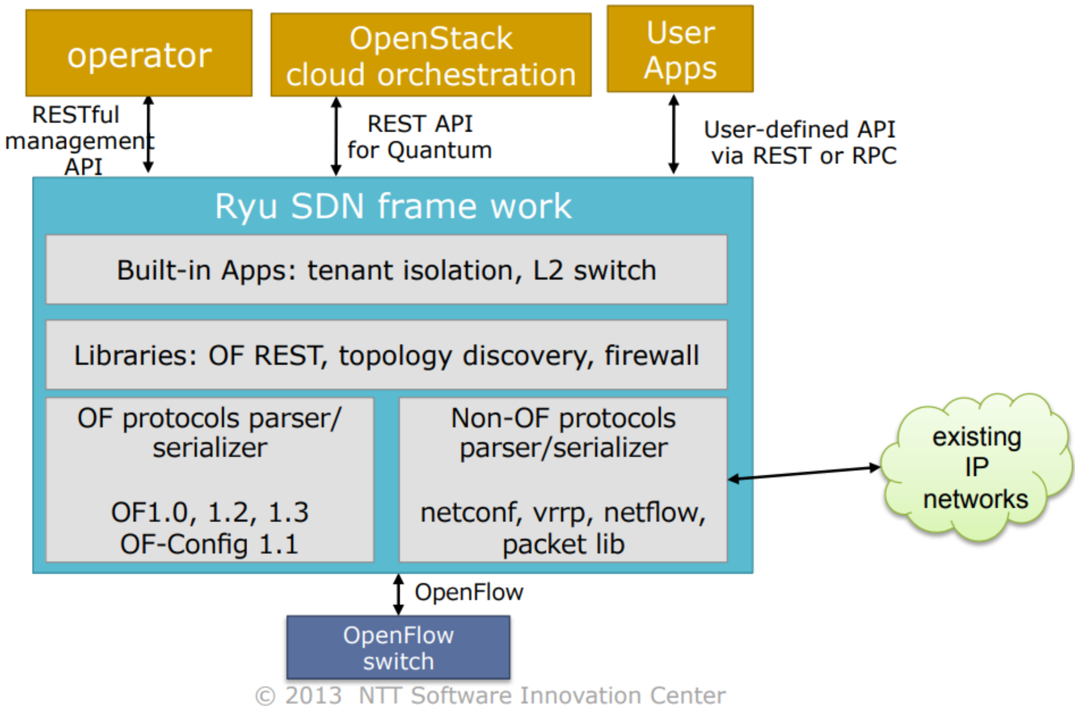


Figure 7 RYU SDN orchestrator architecture

4.3. INFRASTRUCTURE MANAGEMENT PLATFORMS

Infrastructure management platforms are more complex projects that integrate multiple orchestration and control functionalities into one system. They usually touch on different aspects of orchestration and therefore could have partially overlapping functions. Here we present the most prominent infrastructure management platforms.

4.3.1. OPENSTACK

Openstack enables the control of pools of compute, storage, and networking resources that can be managed using specific APIs or a dashboard. Beyond standard infrastructure-as-a-service functionality, additional components can provide orchestration, fault management and service management amongst other services to ensure high availability of user applications.

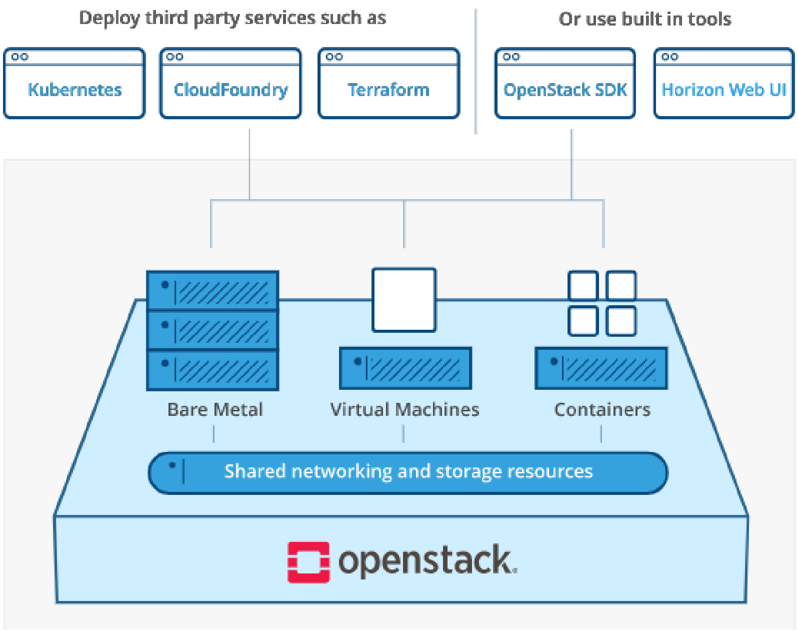


Figure 8 OpenStack architecture

4.3.2. KUBERNETES

Kubernetes is a container orchestration system. It specializes in creating a specific set of functions that control and manage containers in a data center over multiple servers. It is built to handle large workloads while making sure failover and reliability are maintained.

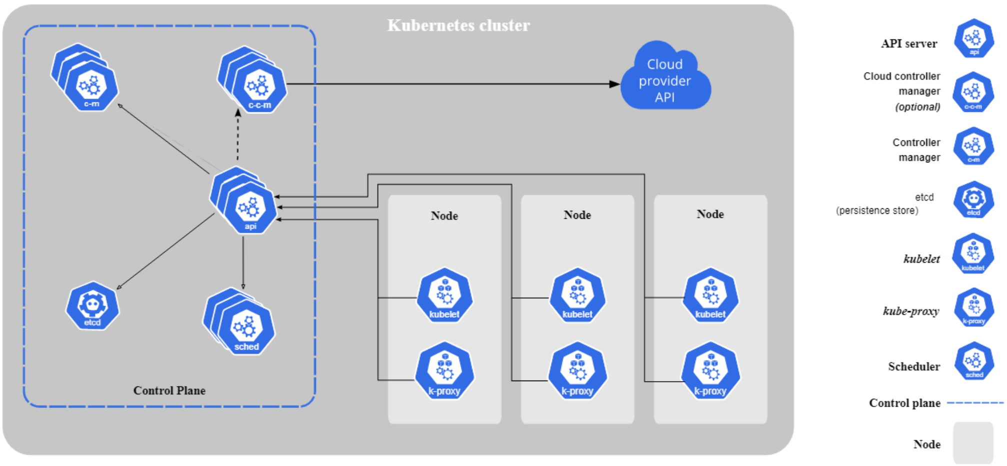


Figure 9 Kubernetes architecture

4.4. SUMMARY AND CLASSIFICATION

In the following we will try to classify and evaluate the different open platforms according to different criteria. We will also present possible weaknesses. This table is not intended to be an exhaustive list, rather it should give the reader some guidelines on the focus and possible strengths and weaknesses of the different projects.

Platform	Objective	Domain	Complexity	Strength	Weakness
ONAP	Resource orchestrator	Network automation and orchestration	High	Supports telecom requirements	Resource greedy and complex deployment
OSM	Resource orchestrator	Network service orchestration	Low	Lightweight, easy deployment	Does not support all telecom orchestration requirements
CORD	Resource orchestrator	Access network virtualization	Medium	Integration of cloud, access and edge components	Limited adoption and ecosystem
OpenShift	Application orchestrator	Application deployment and management	Medium	Robust Kubernetes-based orchestration, developer friendly tools	Focus on application orchestration rather than resource or infrastructure

OpenBaton	Resource orchestrator	Network service orchestration	Medium	Flexible and modular, supports NFV standards	Small community and limited adoption
ONOS	SDN controller	Network controller and management	High	Scalability, high performance, advanced features	Limited ecosystem
ODL	SDN controller	Network control and management	Medium	Large ecosystem, extensible framework	Complex configuration scalability challenges
Ryu	SDN controller	Network control and management	Medium	Lightweight, python-based, flexible	Limited scalability, lacks some advanced features
Openstack	Infrastructure manager	Cloud infrastructure management	High	Feature-rich, widely adopted	Complex deployment
Kubernetes	Infrastructure manager	Container orchestration	Medium	Widely used, ecosystem and community support	Mainly focused on container orchestration rather than telecom needs
Silva	Cloud software framework	Telco and edge orchestration	Medium	Ongoing development	Ongoing development
TeraFlowSDN	SDN controller	SDN cloud	Medium	Ongoing development	Ongoing development
Nephio	Kubernetes controller	Kubernetes controller	Medium	Ongoing development	Ongoing development

5. OPEN PLATFORM EVOLUTION BEYOND 5G AND TOWARDS 6G

Open platforms will become even more essential for network evolution towards 6G. This is due to the fact that 6G will be more application oriented and push the network to become data aware. Therefore, by using open platforms, it is possible to bring in applications and services from major cloud providers as well as developers and service providers. Therefore, we believe that open platforms have to be an essential aspect of the evolution toward 6G.

In general, a major shortcoming of many platforms is the lack of automation. Automation features are essential to support upcoming 6G technologies. This is due to the different underlying technologies that have to be integrated. Those need to be orchestrated in an autonomous way such that the operator does not have to deal with the complexity of such systems.

6. CONCLUSION AND RECOMMENDATIONS

As a conclusion, we now summarize the main messages for the platform working group within the open SNS 6G-AI consortium:

- Open platforms are an important in order to prepare for a **data oriented next generation networks.**
- This white paper provides an **overview of prominent open platforms** that help execute network functions as well as a **taxonomy and classifications of their openness, features, and complexity.**
- Most benefits of open platforms are applications and core network functions.
- We provide **recommendations to push** vendors as well as operators to use **open platforms** to enable cross compatibility, energy conservation, resource efficiency, as well as application/data readiness of the network.
- Open Platforms will be important for 6G applications, since it will offer more **cloud-like execution of new applications requiring stringent requirements** as well as low cost.

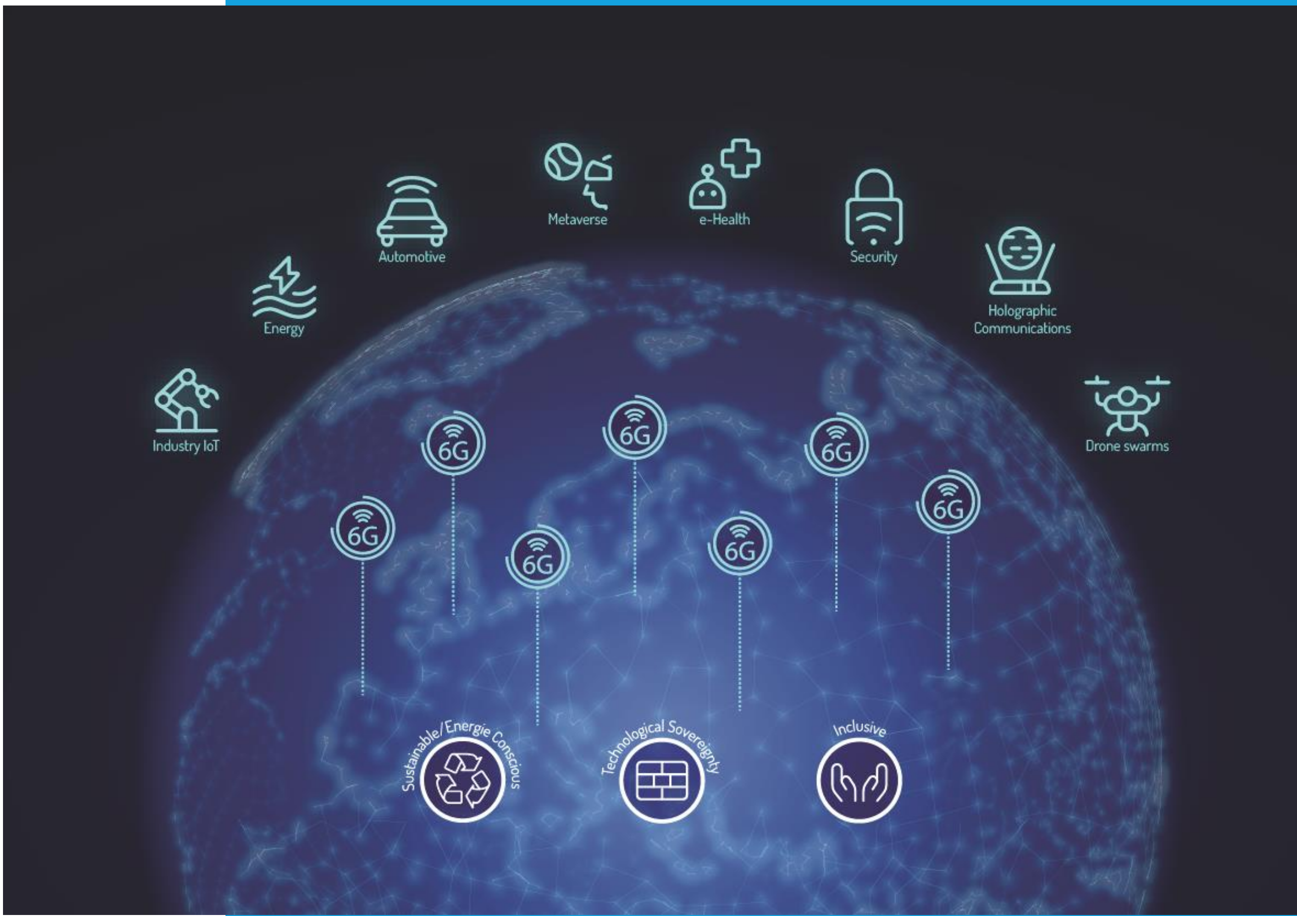
A1 REFERENCES

Ref	Title	Author
MANO	ETSI GS NFV-MAN 001, V.1.1.1, Network Functions Virtualisation (NFV); Management and orchestration.	ETSI
NIST	The NIST Definition of Cloud Computing, recommendations of the National Institute of Standards and Technology	M. Peter and G. Thimothy
CloudComp	Cloud computing: state-of-the-art and research challenges. J Internet Serv Appl 1, 7–18 (2010). https://doi.org/10.1007/s13174-010-0007-6	Zhang, Q., Cheng, L. & Boutaba, R.

A2 LIST OF EDITORS

Surname	***Name***	***Role***	***Company / Institute / University***
Abboud	Osama	Editor	Huawei
Hecker	Artur	Editor	Huawei
EL Amine	Julien	Editor	IRT Saint Exupéry
Djemai	Tanissia	Editor	IRT Saint Exupéry

6G-IA is the voice of European Industry and Reseach for Next Generation Networks and Services



Contact: Office@6g-ia.eu

Website: https://6g-ia.eu/

DOI:
URL: