NGMN

Experience on Cloud Native Adoption

—

v1.1

WE MAKE BETTER CONNECTIONS

Experience on Cloud Native Adoption

by NGMN Alliance

Version:	1.1
Date:	28.01.2022
Document Type:	Final Deliverable (approved)
Confidentiality Class:	P - Public

Project:	Future Networks Cloud Native Platform
Editor / Submitter:	**Fabrizio Moggio (TIM)**
Contributors:	**Andreas Volk (HPE), Marie-Paule Odini (HPE), Özge Gure Alkan (Turkcell), Zhiqiang Yu (China Mobile), Hanyu Ding (China Mobile), Sebastian Robitzsch (InterDigital), Mohamad Yassin (Orange), Hervé Oudin (Keysight), Javan Erfanian (Bell), Gary Li (Intel), Tim Costello (BT)**
Approved by / Date:	**NGMN Board, 16th February 2022**

NGMN e. V.
Großer Hasenpfad 30 • 60598 Frankfurt • Germany
Phone +49 69/9 07 49 98-0 • Email office@ngmn.org

Abstract: Short introduction and purpose of document

This document reports experience from the NGMN Partners on the adoption and on the studies currently ongoing on Cloud Native. NGMN, at the beginning of 2021, released a White Paper providing an overall view on the Cloud Native evolution [1]. That document is about a long journey that is still ongoing and foresees different steps to reach full maturity. The intention of this document is to highlight some of the areas that are on the spotlight now, and the steps that the Partners are currently doing in respect to that journey.

Contents

1 INTRODUCTION

The cloudification of the Telco infrastructure is well started and is running on track. Many, if not all the departments of a Telco company are affected by this evolution. Anyway, it is progressing with a different pace not only according to maturity of the involved technology but also to the readiness of the related business models or operational activities that must be updated. For a deepening on the Cloudification of the telco platform, please refer to [1]. For a deepening on the operation of a disaggregated network, please refer to [2].

The goal of this document is to offer an inside look on some of the activities that are undertaken in these very days as they are experienced by NGMN Partners.

Cloud Native adoption is not a straightforward path and the actual experience shows a different level of maturity and some areas that are gathering more interest than others.

This document does not represent a complete view on the technology adoption, also because it may vary from one Telco to another. This document gives indeed a snapshot of NGMN Partners' experience, providing a concrete taste on the currently ongoing activities in some selected areas.

2 SERVICE ASSURANCE

2.1 Telco Cloud Service Assurance

NFV infrastructures include different hardware units, VIM (Virtual Infrastructure Manager), VNF (Virtual Network Function), and management layers. It is important to collect information on the physical resources (Server, Physical Switch Port, NIC port, Numa, etc.) and virtual resources (vCPU, vRAM, Open Virtual Switch-port, etc.) via a VM (Virtual Machine) running on any infrastructure. To guarantee service assurance, it is necessary to check multiple points and to perform manual operations in the existing infrastructure. For a root cause analysis, many technical aspects, on different logical layers, are under analysis, such as virtual CPU consumption or the status of a physical switch operating system. To identify the cause of an anomaly at service level, it is indeed required to examine the effects and the root causes of problems occurring at any point in the network. The resource usage of distributed NFV infrastructures and certain VIM components must be traceable from a single point. It is needed that the monitored NFV infrastructures have automation capabilities in case of any problems.

A solution that will bring the above-mentioned capabilities is required on Telco Cloud Infrastructures. Service Assurance in a Telco Cloud is supported by a Quality & Assurance application positioned to meet NFV-OSS integration development. Such a component is needed to complete the Telco Cloud transformation.

Service Assurance in a Telco Cloud requires to collect alarms, KPI's and logs from all Telco Cloud components; hardware/servers, operating systems, virtual infrastructure managers, hypervisors and importantly applications (core network nodes etc.) that reside and operate on top of the NFV infrastructure to create visualisations using all these data. The Assurance solution also correlates all these with the data obtained from the auxiliary network components such as switches, customer experience managements tools etc. Performance KPI's are gathered from the virtualised nodes using the service assurance. With this solution, these metrics can be visualised for the service owners as well as allow the users to create rule sets for the KPI's. A service assurance is a system that directly impacts the NFVI lifecycle via root cause analysis.

Some Telco Cloud infrastructures contain a hybrid server architecture including different virtualisation technologies and an ecosystem of a growing number of VNF for user and control planes. Most of them have their own element management systems. All these systems can be managed via Generic and Specific VNF managers that are compliant with the accepted standard, specification, and orchestration. Telco Cloud Service Assurance is a tool in monitoring all these separate systems and structures and correlating their data. SA (Service Assurance) can be a single platform which collects data and correlates all data. Operators could benefit from Telco Cloud Service Assurance principles as the following:

- Closed Loop Automation
- Root Cause Analysis of the problems in Telco Cloud system
- Effective and centralized KPI, Alarm and Log monitoring for VNF applications
- Multi-site cloud data correlation for resource planning activities
- Machine Learning based anomaly detection algorithms implementation for key KPIs
- Agility

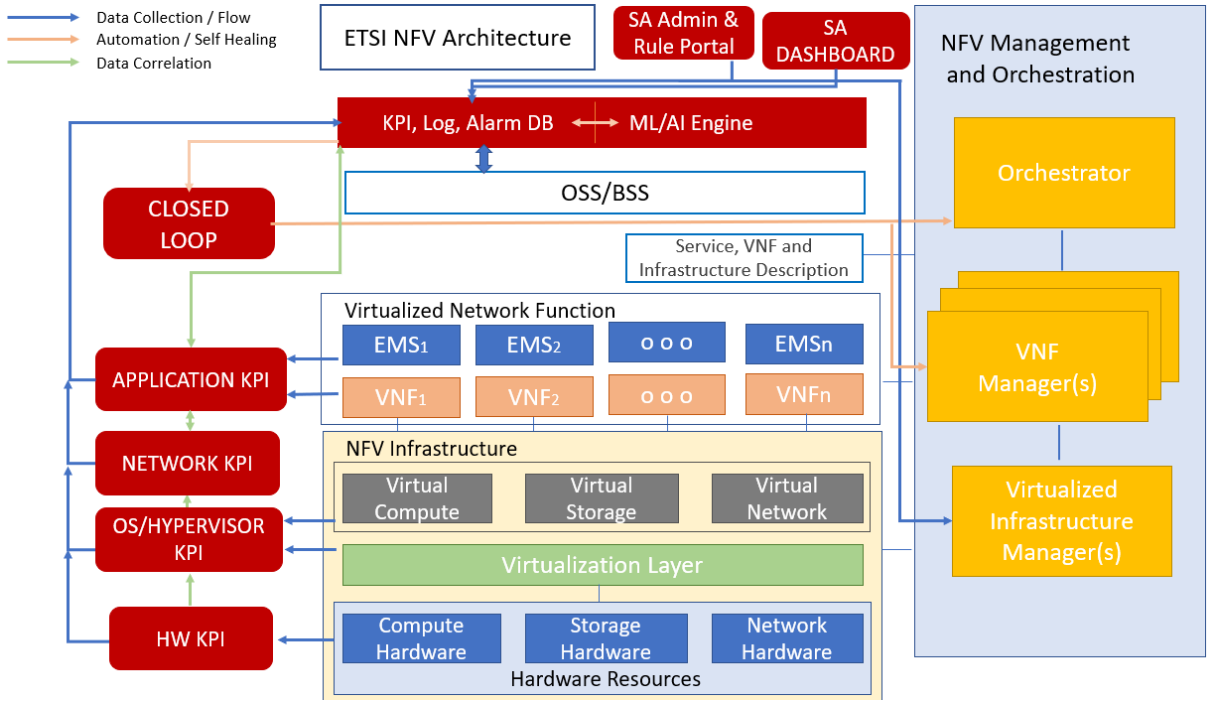


Figure 1: Service Assurance Architecture

2.2 Architectural Overview

A Telco Cloud Service Assurance consists of the following main components.

The collection layer includes open-source components which is compatible with Cloud Native Applications for future compliance, especially by considering a microservices based 5G architecture. EFK stack and Prometheus can be used in this layer as well as home-grown tools developed for resource and inventory management. The open-source components are mainly used for collecting KPIs/Alarms/Logs etc. The other tools are used for collecting inventory/resource information from the SA to monitor/manage resources and keep the relationship among different components.

The analysis layer includes Data processing, Rule Engine, and Machine Learning based Anomaly Detection and Automation Controller (which is integrated with Domain Orchestrator to trigger actions) components. This layer is a combination of open-source components and local software developments.

The visualisation layer is a combination of Grafana, Kibana and related Dashboards for KPI & Log monitoring, analytics and rule creation.

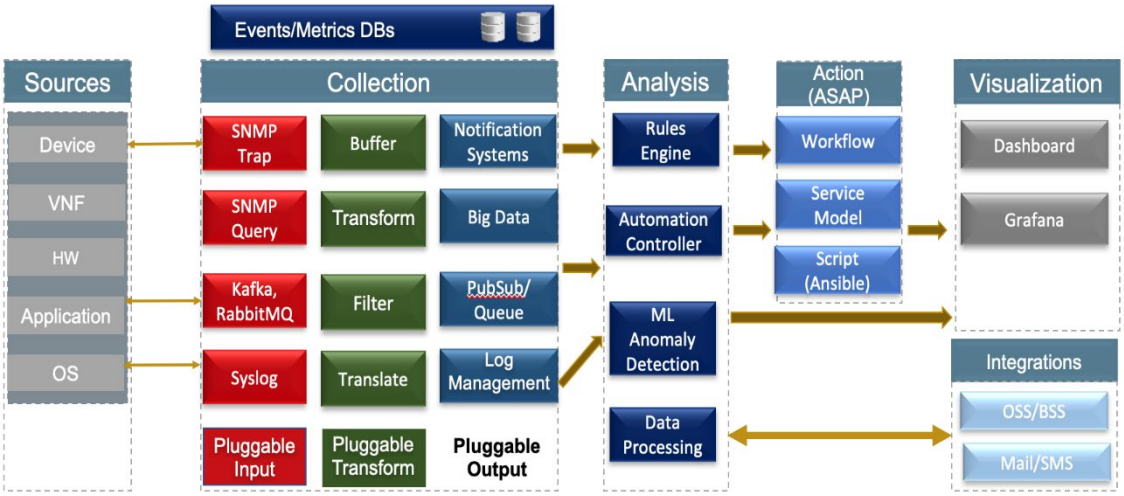


Figure 2: Service Assurance Areas

A Service Assurance solution includes core components that can be seen from the above architecture for data collection, analysis, and visualisation. It employs multiple open source components in order to follow the latest trends in a Telco environment. Prometheus and Grafana can be main open source components.

A Service Assurance solution is a 5G-ready architecture design to achieve the following advantages over the traditional solutions:

Advanced VNF KPI integration & analysis: The solution is able to get the KPIs and logs from VNFs by using different mechanisms. More importantly, the information collected from the VNFs is used to make analysis, prediction for the future usage and root cause analysis together with the data collected from the cloud infrastructure level and from external systems wherever applicable.

Machine Learning based anomaly detection: One of the most challenging tasks in today's complex mobile operator's environment is to find out the anomalies based on the historical data. The solution provides the ability to use advanced ML based anomaly detection techniques even when there is no existing predefined rule for the anomalies.

Closed Loop Automation: The solution is integrated with MANO (Management and Orchestration) products which enables service orchestration features in operator's network to achieve triggering automated actions through monitoring the system. It is possible to make integrations with other service orchestration solutions as well.

Flexible Rule Engine: The solution contains a portal so that flexible rules can be created based on the combination of KPIs, sites, projects or virtual machines. The rules and the actions to be taken can be dynamically created and modified.

Root Cause Analysis of the problems: The most complex task is to make root cause analysis after the incident in an operator's network. The solution is capable of doing RCA with the help of advanced data processing capabilities together with powerful data correlation techniques.

Multi-site data correlation: The data from different sites is correlated and unified so that it can be queried and used for data processing for relevant rules.

2.3 Data Collection Methods and Data Analysis

2.3.1 HW Data Collection

Hardware information (CPU, memory and disk details) and some KPIs (HW temperature, energy consumption etc.) can be collected from hardware managements systems. The selection of important KPI's is the key issue here as well as the definition of the APIs used to collect data from a heterogenous, multi-vendor environment.

2.3.2 OS Data Collection

KPI collection from the host OS will be an important topic via monitoring agents. The following option can be supported in SA Solutions.

Real time performance monitoring agent tool like netdata has the advantage of being hosted by CNCF, providing KPI at 1 second intervals, higher default KPI number, no need for configuration and operable with GUI. The retention time of KPI data in the RAM is important (30 minutes of KPI data is kept in the RAM. The duration can be set and KPIs are transferred to Prometheus DB in the meantime). Disadvantages are, it is arelatively new software and has a bit higher more memory usage around 0.2-0.3G.

2.3.3 VIM (Virtual Infrastructure Manager) Data Collection

The following data can be collected, processed and a necessary mapping can be done for queries from e.g. Openstack.

- Openstack Information
    - VM Information

  - Host Information
  - Flavour Information
  - Image Information
  - Network Information
  - Subnet Information
  - Availability Zone Information

- KPIs from Openstack services
  - Memcached KPIs
  - HA-Proxy KPIs
  - MariaDB KPIs
  - RabbitMQ KPIs

Apart from the information above, resource usage information can be collected, parsed and processed so that resource usage can be queried per host, project, and site accordingly.

2.3.4 Hypervisor Data Collection

The following information can be collected from hypervisor/OVS. That information can be collected over an SSH connection towards hosts.

- KVM information over virsh
  - Disk IOPS information
  - CPU usage
  - Memory Usage

- OVS switch information
  - Switch metadata
  - Interface statistics

2.3.5 VNF Data Collection

Data collected from VIM level are not sufficient to address all monitoring challenges effectively due to the fact that VNF related issues might be invisible to VIM level (SW errors, crashes, specific KPI degradation which has an effect to QoE of the subscriber etc.). As a result, KPIs and logs should be collected directly from VNFs wherever applicable for effective service monitoring. Some VNFs have VMs that receive all system KPIs. KPIs can be collected from here or from the management systems of the relevant VNF.

2.3.6 Supplementary Data Collection Details

Alarm Management Systems Data Integration: One of the data sources to process and analyse is the fault management data related to the hardware, operating systems, virtualisation SW and VNF application itself. A Service Assurance solution needs to work with an external fault management system in order to collect and process FM data for root cause analysis and correlation of the alarms with other anomalies in the system. One of methods that can be used is the Impala interface on SA System. Impala will provide an SQL interface to a Big Data environment where the Alarm data is located. It would be better if only relevant data are fetched from the Impala interfaces by using filtering mechanisms. The target interval can be 1 minute, but it may change as per the interface requirements and system resource effect in the target database, which can be determined during integration tests.

Physical Function (Switch/Router) Data Collection: Physical functions play an important role to verify that the system is functioning properly. It is also important to get the network related KPIs including but not limited to traffic measurements per sites, projects, packet drops, retransmission and protocol errors. Network level information is used to make some important mapping (such as SR-IOV usage mapping from VM to switch, root cause analysis of interface problems etc.) related to the network.

SNMP, Nxapi, Netconf, Restconf, SSH can be used to fetch those critical data from network elements.

Customer Experience Management Tools Data Collection: From an end-to-end perspective, data collection and processing may not be enough to cover and detect some QoE related issues because the KPIs collected from the systems may not produce a clear alarm, log or deviation from the average KPIs. However, it might be possible to correlate a slight deviation of important KPIs with a considerable increase/decrease for external KPIs which is directly related to QoE over data probes. Those data probes contain very useful aggregated data regarding the QoE aspect of the monitoring topic, which might be correlated to VNFs, or in some cases to the end users.

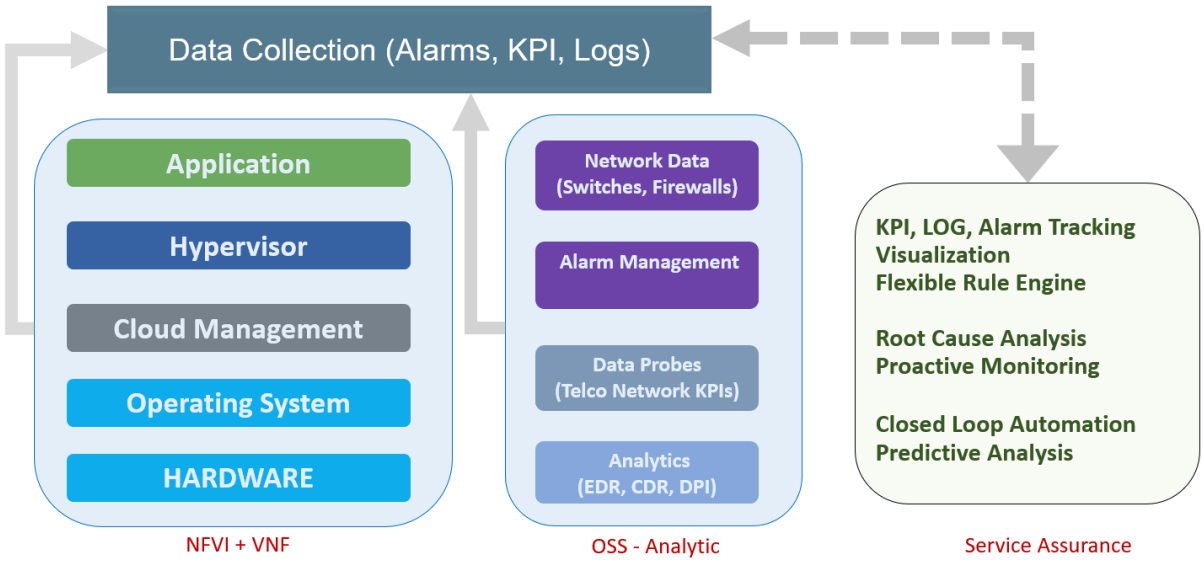


Figure 3: Data Collections

2.3.7 Data Analysis

Data Analysis is at the core of the Service Assurance. All the data coming from the VNFs, OS, physical functions, external systems, HW and VIMs are processed for actionable insights either via service rules created by the operator or dynamic ML based anomaly detection mechanisms.

The following are the details of data processing ML based anomaly detection.

In addition to the service rule creation, which is based on a static binding of the rules in the section above, Machine Learning based anomaly detection methods can be used in the solution.

This approach is particularly suitable for finding out anomalies of the dynamic resource usage or network statistics, which changes frequently without a clear pattern or shows seasonality.

Relevant KPIs and logs can be monitored with this approach to increase the accuracy of the detection of anomalies within an operator network.

The default ML library used in the solution is Keras on top of TensorFlow. Additionally, the LSTM network can be used to detect an anomaly based on time series data with seasonality, wherever applicable.

Data processing is done primarily by using either of the approaches summarised above. Moreover, raw data regarding resource usage and infrastructure data (e.g. VM details, limits, host, network details etc.) are processed and mapped for making queries.

Root Cause Analysis: RCA (root cause analysis) can be done by employing specific algorithms. Dynamic RCA can be summarised as follows:

An RCA process is triggered when a problem/anomaly is detected while processing data, either by static rules or ML based processing.

After triggering an RCA process, relevant information is calculated starting from HW to the level that the anomaly is found. For example, if there is an anomaly in VNF KPIs, VNF -> hypervisor -> VIM -> host -> HW association is dynamically searched to find out a correlation. Determined problems are tried to be correlated with others. As an example, VNF KPI problem is correlated to Host OS interface traffic decrease via the mentioned association.

In some cases, the data in the system is good enough to make the correlation between the problem/anomaly detected in the system and the recent logs/KPIs/alarms/events within the data collected from the whole ecosystem. In this case, the necessary report is generated and sent to the relevant persons accordingly.

If there is no clear linkage between the problem/anomaly detected in the system and the recent data analysis, another report is prepared with the data which might be relevant so that it can be a starting point for the investigation.

3 EDGE-CLOUD COOPERATION

Telco Platform cloudification provides the means for an Operator to plan its deployments according to business choices and market opportunities leveraging on a common platform in the different locations. The Operator is less limited by technological constraints in term of installation and operation. Cloudification indeed provides a common and shared technological infrastructure in the centralised Telco Data Centre, in the Telco Edge and in the Cloud. As a reference and for further details on Hybrid Cloud see Chapter 6 of [1].

3.1 Multi-layered Application

An edge and cloud cooperation analysis should consider the different market opportunities and different end-user requirements. The solution must be indeed flexible and open to support

different use cases and business models. Best practices in this area are not yet defined and innovation, engineering and business are indeed all drivers to identify solutions. It is important to consider short term implementations based on the current available technologies with an outlook to the evolution based on new edge components, whose maturity is not yet here.

New high-performance and tailored network services, close to the end-user, are needed to support specific multi-layered applications. Edge is not the solution for everything. It's not the most adequate place for any kind of application. Complex multi-layered applications must be analysed to identify what can be moved to the edge. Considering, for example, an Industry 4.0 application, a traditional deployment foresees a module nearby the controlled devices, applications that implement algorithms requiring low latency for effective decision-to-action tasks and back-end applications. Let's consider the following figure representing a "traditional" architecture not yet leveraging on Edge solutions.

Figure 4: Traditional Deployment without Edge

To support such an architecture, a relevant deployment on premises is required. This architecture foresees consistent investments and operation on premises. Such a solution must be analysed considering the different layers (Control, Logic and back-end) to identify which components can be moved on a shared platform such as Edge datacentres or Cloud back-end. It is important to identify the requirements and to balance the distribution of the modules in the different locations considering requirements, costs and opportunities. The Control system has generally requirements that place it nearby the devices, On Prem or at the Edge. The back-end is generally designed around data storage or cloud resources and there are no specific network requirements in terms of latency. A possible deployment leveraging on Edge and Cloud resources could be the one represented in the following figure.

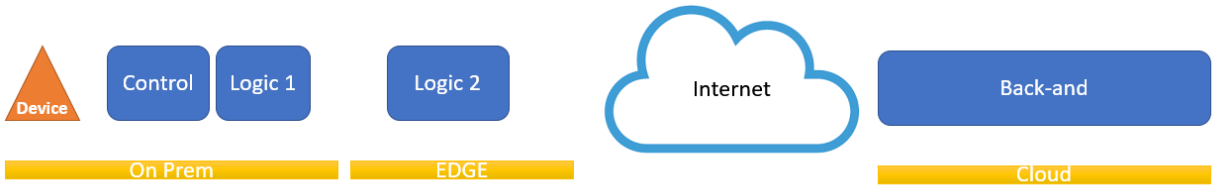


Figure 5: Deployment with Edge

In the picture above the "traditional" solution is analysed:

- The back-end components, being usually integrated with cloud resources (such as data lake), are still well placed in the cloud. It is important indeed to notice that the Edge is usually a more expansive environment with less resources. It is intended to support latency-oriented service while data-oriented services are more suited for public cloud deployment.
- The Logic components must be analysed and maybe redistributed leveraging on the micro-services architecture. Those micro-services that require a tight coupling with the devices and the Control logic should stay On Prem, as before. There are components running a logic whose latency requirements are not compatible with a cloud deployment but that are compatible with an Edge deployment.
- The actuator device (e.g. PLC) control modules with the coupled logic should stay on Prem or at the Edge according to the latency requirements and architecture design.

To support such a scenario, it is important to offer a solution that can manage the deployment and the lifecycle management of complex, multi-layered communication services on top of a distributed infrastructure.

Edge Data Centres, Central/Regional Data Centres and Cloud solutions must be managed coherently as part of the same, widely spread, Telco cloudified infrastructure. Considering the maturity of the existing cloud solutions, the different kinds of applications that are currently based on different cloud platforms and services, a multi-Cloud approach may be adopted to be able to provide the right solution according to the different end-user's requirements.

3.2 Network and Application Services at the Edge

An innovative product is a sort of complex system the end-user mostly perceives as a black box acquired maybe from an order management portal or marketplace. It is indeed composed by many underlying services, devices, configurations and features. For the Telco related aspects, following a Cloud Native approach, it should be layered on top of a distributed infrastructure

supporting both network and application deployments. For Edge services, this distributed infrastructure leverages on network and application capabilities integrated at the Edge. As showed in the following picture, network resources and capabilities and application resources and capabilities cooperate to provide an E2E service at the Edge. back-end applications in the Cloud are accessible through an internet access or via a virtual private connection. This architecture provides an open environment balancing edge and cloud solutions being part of the same service.

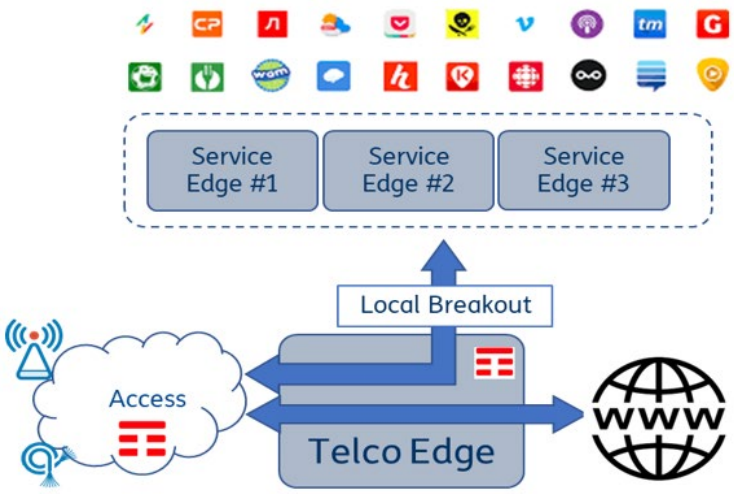


Figure 6: Local Breakout

The infrastructure supporting network capabilities at the edge has the following main characteristics:

- routing of selected access traffic flows (Local Breakout): for example, a small portion of 5G Core Network placed at the edge of the network that opens IP interfaces towards service logics.
- routing of the local traffic towards an edge data network.
- hosting of Telco functionalities which are controlled or in tied relationship with the core of the Telco network.
- Telco network features are exposed via APIs.
- it is completely under the control of the Telco Operator: implemented and managed by the Telco Operator.

The application infrastructure at the Edge has the following main characteristics:

- hosting of specific Edge applications, usually provided by 3rd party developers or partners.
- it offers APIs to enable the environment to external application developers.
- it supports a native hybrid cloud design: public and private cloud cooperation.
- it can be offered by the Operator, by Hyperscale providers or in the context of partnerships with operators / third parties.

Telco and application infrastructures are based on the same cloud native technologies. This synergy provides a big potential advantage in terms of deployment and operation of the platforms. Currently the proposed solution by vendors, even if clearly based on the same technologies, architectures and maybe open-source components, are not ready yet to be managed as a whole system. An integration among the different components, even if supported by similarities and standard architectures and APIs, is needed in any case.

Different approaches can be adopted to integrate network and application infrastructures considering the different orchestrators used by subsystems. Many solutions are proposed by vendors as E2E while often the Operator needs a multi-vendor-based system, where the different orchestrators, in different domains, may need to cooperate.

A communication service request can be managed at the service layer, decomposing it in different network or application services to be set up. The network or the application solutions could be provided by the vendors with a specific orchestration layer. It is a needed choice to decide how to integrate the orchestrators also considering that the underlying platform, even if it can be potentially shared, is usually dedicated to Network Functions (NFs) or Application Functions (AFs). A possible implementation is with the Service layer interacting with the two domains: Telco and Application. Another approach can foresee the Service layer asking to the Telco domain a network slice comprising AFs. In this scenario, it is the Telco orchestrator that, therefore, interacts with the Application domain requesting an AF deployment.

It is important to consider that the Edge and Cloud balancing depends on many factors related to the product itself (network requirements, SLA, isolation etc.) and partly related to the need of balancing the resource usage. The overall management system must be able to take care of all these aspects automating the design and the deployment of all the components in the most appropriate location. For this reason, the Service layer and the underlying domains must cooperate considering service requirements, policies and resource allocation.

3.3 Traffic Steering at the Edge

Edge and Cloud cooperation also means leveraging on network capabilities to steer the traffic towards the right destination minimising latency. Traffic steering should be exposed to a third party following a "beyond connectivity" approach.

Traffic steering APIs are an important asset for Telco Operators to offer a tailored network feature providing on demand and customised network acceleration at the edge.

Traffic steering at the edge is standardised by 3GPP for the 5G network but it is also possible to be achieved in 4G networks, leveraging existing network functions exposing traffic steering APIs.

Telco and application infrastructures can be deployed jointly or separately (e.g. on a different infrastructure) at the Edge Data Centres.

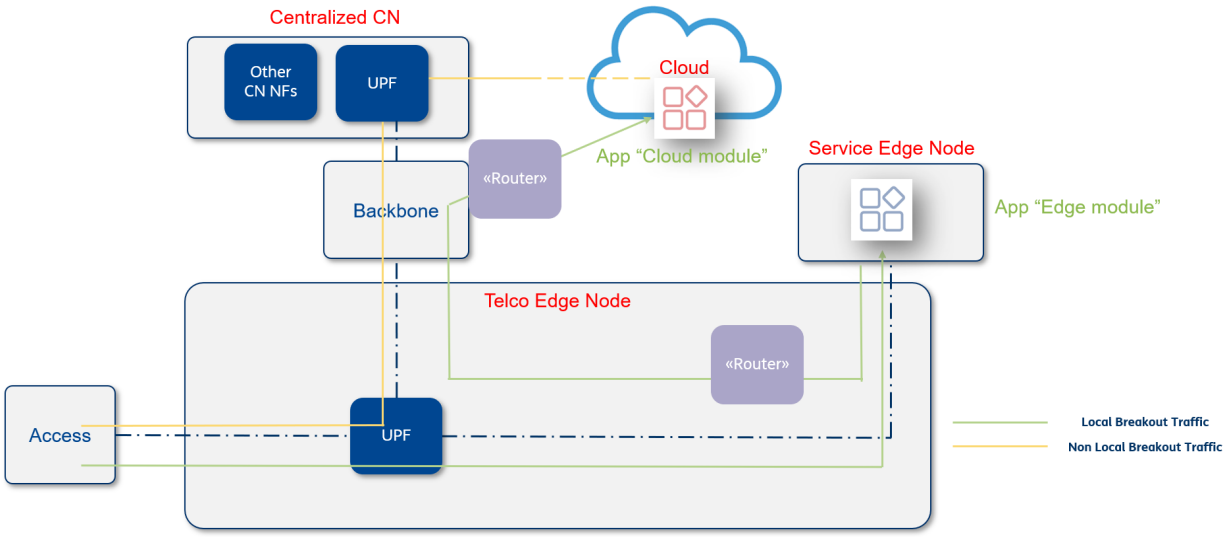


Figure 7: Traffic Steering to Connect and Cloud App Components

Considering the edge nodes, both 4G and 5G Core Network NFs can be part of a geographically spread solution. It is important for an Operator to define an API that hides this complexity to the API users. It is also important for Operators to agree on a set of APIs for edge enablement to provide developers a common set to use.

Application's deployment at the Edge could be enhanced by means of a web portal provided by the Operator. Such a tool should simplify the Edge deployment allowing an easy definition

of the geographical zones supporting the Service in terms of network features (e.g. expected latency in a geographical area) rather than just based on computing capabilities.

This tool can also leverage on the emerging standard solutions for edge discovery recently defined by GSMA and 3GPP [1]. This would enable support of Registration, Discovery and Mobility of Applications among different edge DCs.

This tool should be very easy to use to identify the best Application DC for Edge deployment with the appropriate correlation to the Telco DC according to the network requirements. API for Edge acceleration such as traffic steering out should be exploited.

4 NETWORK FUNCTION REDESIGN TO BE CLOUD NATIVE

The motivation for cloud-native functions, architecture and operation is well articulated and recognised. In short and ultimately, it is about agile value creation and delivery, which is dynamic and responsive, resilient and available, flexible and scalable, modular and interoperable, and efficient.

The design and redesign of physical and virtual network functions to become cloud native have been ongoing while several co-existence scenarios may exist. The implicit abstraction and extensibility should be able to accommodate the co-existence as well as the path forward, while the DevOps paradigm brings in concepts such as continuous integration and continuous delivery pipelines' optimisation, integration and transition.

Cloud native applications are created with a great deal of composability, using containerization and orchestration of microservices. This allows efficiency, interoperability, and agility in availability, time to market, responsiveness, re-use and change.

This chapter discusses the experiences of redesigning networking functions to be cloud native utilising the microservice paradigm. The experiences and advances shared in this chapter were obtained through work in the area of Service-based Architectures and cloud native orchestration of 5GCs.

4.1 Design Patterns towards Microservice Architecture

When moving away from a monolithic software design towards a microservice-based realisation, the 12-factor app methodology [3] provides the key software design patterns on how to achieve that, as described in detail in D1 [1]. However, the key challenge is to move a monolithic NF into a set of functions realized as microservices that form the NF in its entirety. The criteria presented herein on how to achieve may not be applicable to all type of NFs and has been developed focusing on 3GPP's control plane NFs.

The criteria for decomposing a monolithic NF into a set of independent code blocks with Application Programming Interfaces (APIs) in between can be conducted using a set of criteria. The most straight forward one is the decomposition along functionalities of a Network Function. When looking at 3GPP, the 23.501 technical specification provides the boundaries for this for each 5G Core (5GC) NF. This split is based on functional modules and disregards the data modules which define the usage in the entire system characterised by (but not limited to):

- Bottleneck and Parallel Execution: Functionality that poses a bottleneck in terms of time delay processing requests may be an indicator of creating a dedicated microservice for that. The intention then is to allow the utilization of more compute capabilities for this microservice to mitigate the bottleneck. Furthermore, being able to scale up/down/in/out such microservices greatly allows coping with demand and enable a greener power consumption.
- Resilience: Based on Service-level Agreements (SLAs), key components of an NF can be identified that allow increasing resilience against failures and increase availability of a service.
- State Dependency: To process a request, a decomposed NF may require a certain state before providing the response. If the state is unknown, it must be requested first which can add delay and network load. Clearly, the state relates to the context under which requests are dispatched to specific microservices that handle them and to ensure state transfer is kept at a reasonable minimum. But once the context becomes way to complex to be identified by the routing component or message proxy, this may be an indicator of not decomposing this functionality into microservices.

Also, when decomposing a monolithic NF into a set of microservices, it increases the attack surface for unauthorized access. Thus, it is of paramount importance to have authorization and authentication built into the microservices by the likes of JSON Web Tokens (JWTs).

4.2 Addressing of Network Functions

Monolithic NFs are exposed via a single addressable identifier, e.g. an IP address or an FQDN. When decomposing a monolithic NF into a set of microservices, it is still a rather common practice (as part of the 12 factor app methodology) to implement a proxy component which exposes the NF through a single addressable identifier. Figure 8 illustrates such scenario, with P being the proxy and FA through FC the various type of microservices. As each microservice type can be theoretically provisioned as more than one instance, each functionality has a second numeric subscript indicating the instance number. The addressable identifier in this example is nf.foo.com which the proxy receives and dispatches the request to a specific microservice type based on the web resource in the HTTP request, assuming HTTP is used as the application layer protocol. This approach is commonly utilized in the cloud world and current 5GC realizations with technologies such as Kubernetes providing the framework to orchestrate such set of microservices.

However, this addressing only allows the provisioning of all microservice types that form a single NF together so that any request to the addressable identifier (the FQDN nf.foo.com in the example here) can be served. When aiming for a more diverse deployment across multiple locations (data centres/edges), a single addressing identifier for the entire NF means all microservices that form the NF must be provisioned across all locations.

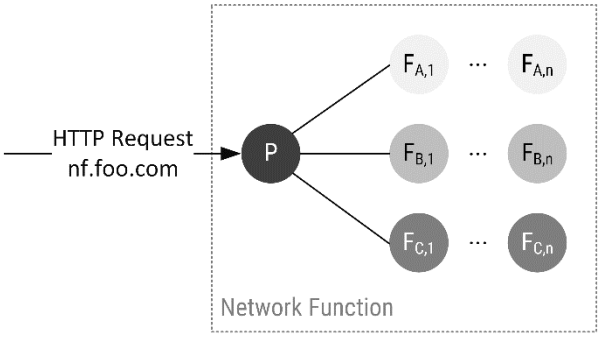


Figure 8: Traditional Addressing of Cloudified Virtual Network Functions

The key reason why 3GPP had adopted cloud principles and defined a Service-based Architecture (SBA) with Release 15, was the argument to enable multi-vendor deployments of 5GCs. While a single addressing identifier for an NF does allow the vendor multiplexing of NFs, a more extreme scenario where functionality of a single NF is based on multiple vendors is not a feasible scenario. As the NF software architecture must be shared among vendors to realize a single NF, any standardization effort in relation to this attempt is outside of 3GPP's scope and

merely a software design discussion with significant burdens around agile code development and integration, which is very often proprietary to vendors. Thus, standardized addressing identifiers for NF functionalities would allow such intermixing of vendor solutions and still protects the key assets of 5GC providers, i.e. their software code. Figure 9 illustrates this approach as an evolution of addressing identifiers in a 3GPP-based SBA system [4].

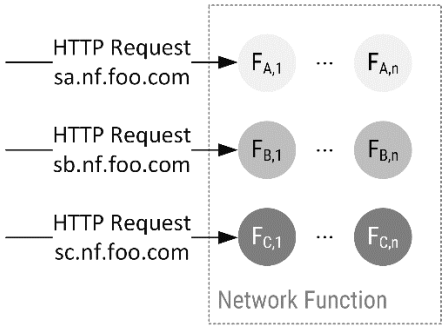


Figure 9: Exposing Internal Components of Network Functions as Addressable Endpoints

Furthermore, allowing each NF functionality to be addressed individually also fosters the design and deployment of slimmer 5GCs for Non-Public Network deployments (Standalone NPNs (S-NPNs) in particular), where not all 5GC NFs are required in their entirety or at all. For instance, in an Industry 4.0 scenario where robotic equipment uses 5G as the communication, infrastructure billing and paging is not required. Instead, special features such as Time-Sensitive Networking (TSN) or 5GLAN are needed.

4.3 Instance Affinities

Independently from which granularity of address identifiers are in place, another question arises on the affinity of instances to each other knowing that in a 5GC many (not to say all) NFs have an UE context as their state in order to process requests. An affinity defines how long, and under which condition(s), a routing or message proxy component keeps request and response instances affiliated with each other beyond a single HTTP transaction (request/response). This is to avoid a constant state transfer to new producer instances of the same type but for the same UE context. Figure 10 illustrates this scenario where 3GPP's optional Service Communication Proxy (SCP) is shown as the routing component and a set of functions, F_A, F_B, F_C, with different number of instances, $F_{A,1}$, $F_{B,1}$, $F_{B,2}$, $F_{B,2}$, $F_{C,1}$, distributed across three Service Hosts (SHs) (aka locations), i.e. SH_1, SH_2 and SH_3.

Software components that implement a service, e.g. 5G Core Network Functions, do not have any additional code that takes care of functionality other than of handling 5G control plane

requests and responses. Thus, for being classified as "cloud native" any affinity question must be answered by an external component such as the SCP. Assuming a 5GC-wide unified telemetry framework which feeds directly into the orchestration (lifecycle management) and routing component, the affinity question as in when to change the instance relationships becomes a rather complex problem to solve, given that the key input is the context (state) which must be shifted to a new microservice in case affinities have been reset.

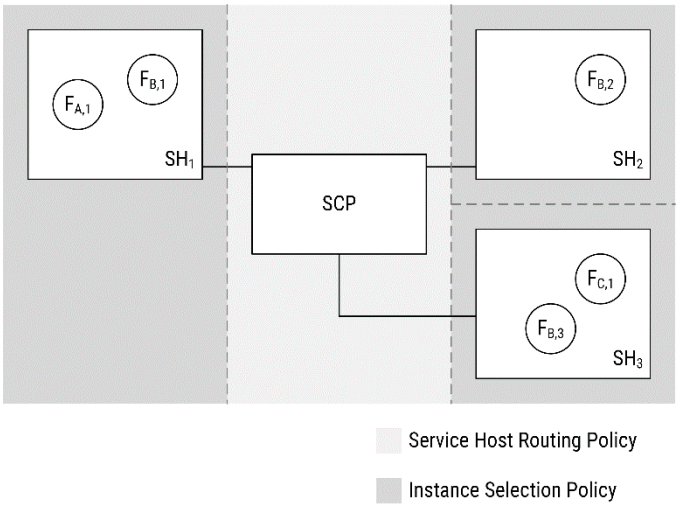


Figure 10: Instance Affinities and Their Policies

4.4 Concluding Statement

It is worth noting that the experiences and challenges presented in this chapter were merely derived from current state of the art methodologies and technologies. The single-focused usage of Kubernetes as the cloud native technologies for microservices has directly affected the adoption of cloud principles in the telco world. As Kubernetes is a workload and container management framework designed and built for a single data centre, it must be asserted that the telco domain has different requirements around multi-locations (edge/far edge/fog), containerized connectivity-oriented (network) functions and different Service-level Agreement in regard to resiliency and traffic patterns. Also, when leering at the cloud technology research community, concepts around Function-as-a-Service (FaaS) (aka serverless) emerges as the next evolution of further decomposing microservices into even smaller code blocks. As a result, the orchestrator (NFVO) receives the duty to find the most suitable host where this function should run, how it is provisioned (bare-metal, container, VM, etc.) and when which function must be operational. Once these concepts further materialize, it becomes apparent that the

experiences described in this chapter strengthens the demand for a more open programmable and vendor-independent and network-oriented cloud native orchestration framework.

5 TELCO EDGE

With the rapid development of 5G industry, the Internet and Internet of Things industries have entered the era of big data. Data computing are nowadays a key to the success of the development of Internet and Internet of Things. With the gradual landing of 5G, various industries have gradually entered a new stage of rapid development. We can indeed consider examples such as V2X, smart medical care, industrial internet. Internet of things and AI have gradually entered a stage of rapid development and this requires the support of a large amount of computing power. Under the premise that traditional cloud computing cannot meet the demand, edge computing has become another solution.

5.1 Scenario for Edge Computing

Generally, the most suitable scenarios for edge computing include: ultra-low latency, real-time processing, real-time computing, rendering and analysis, large-capacity data transmission, deterministic networking, security and data protection, etc. Edge computing shortens the physical distance between communication nodes, significantly reduces the delay, enables real-time rendering and analysis, and greatly increases the bandwidth. Putting core functions on the edge (e.g. UPF) can also transfer massive amounts of data more efficiently, thus reducing network operating expenses.

The combination of MEC and 5G is the core for Operators to upgrade their networks services, expand new customers and new fields. Operators need to enlarge and strengthen the mobile connection business, newly expand the 2B enterprise market, take the connection as the starting point, and develop edge computing.

Communication is the basic function of the network. All kinds of services of edge computing are closer to end users, and the project environment is different. Therefore, it is necessary to support all kinds of network access to meet the differentiated communication needs. Including services with large bandwidth and low delay requirements. When deploying a 5G network, it is also necessary to comprehensively consider 5G coverage, business requirements, costs and compatibility with 4G networks.

The 5G network traffic steering function node is UPF, and the deployment forms can be hardware and virtualisation. The UPF deployment strategy should be depended on a combination of the application scenario requirements, manufacturer's product maturity, computer room conditions, performance requirements and other factors.

5.2 Edge Platforms

Edge computing IaaS platform serves edge applications in cloud form and is a cloud infrastructure for deploying and running edge computing services and related network element functions.

Edge IaaS platform usually includes virtual machines, containers and other virtualisation forms. Virtual machine is an important way of edge computing IaaS platform. The virtual machine is very mature in the field of NFV, which mainly uses OpenStack technology as the main component to manage various types of hypervisors and virtual machines hosted by hypervisors and ensure the smooth operation of business on the platform.

The edge cloud PaaS platform provides the running environment and necessary components for edge applications and can cut and deploy functions according to customer requirements, thus supporting a lightweight deployment.

Generally, the edge computing PaaS platform consists of a capability layer and an API gateway. The capability layer mainly includes network capability, characteristic industry capability and general basic capability. The network capabilities include local distribution, basic networking service capabilities such as NAT, virtual firewall (vFW), DNS, and service load balancing, and also provide services such as radio network information service (RNIS), bandwidth management, user identification, and location information. Industry capabilities such as AI capabilities, video coding and decoding capabilities, IoT device management and data acquisition and analysis capabilities, etc. to enrich and improve the edge computing PaaS capability layer; General capabilities, such as middleware and databases such as Kafka and RabbitMQ, provide a necessary guarantee for the normal operation of applications.

An API gateway mainly aggregates the services exposed by various subsystems as coherent APIs to edge applications. The gateway can uniformly control the API, such as for authentication, authentication, flow control and monitoring.

5.3 Edge Application and Requirements

Electronic manufacturing industry usually needs to organize production in "small batches and multiple batches". Flexible manufacturing technology is very important for intelligent manufacturing. The local diversion, low latency and local computing power of edge computing can solve several major "pain points" of enterprise network infrastructure construction: often enterprises have specific requirements in terms of network performances and data security that prevent the deployment on public cloud systems. In addition, WiFi, LoRa and other solutions, under certain conditions, fail to meet specific requirements of bandwidth, delay, stability and reliability. Third, the demand of industry for efficient manufacturing makes the manufacturing industry introduce a large number of intelligent technologies based on AI, AR, etc., and at the same time has a greater demand for computing power.

The video data volume of the industry is large, which often requires great bandwidth. In addition, the rise of high-definition live broadcast, AR/VR and other real-time services also requires low latency. The new technology of edge computing will greatly improve the video efficiency of the industry.

Based on 5G+MEC, 4K ultra-high-definition video surveillance will be built, and the city surveillance service with wide coverage, quick response and intelligent interconnection can be provided for governments, public security, comprehensive management and transportation supervision departments. Based on a unified cloud platform, end-users are provided with an overall video surveillance services such as video collection, storage, management and analysis. Under the 5G eMBB scene, the monitoring equipment will achieve 4k/8K ultra-high definition resolution, and the video details will be richer, which can effectively improve the value and accuracy of monitoring video analysis. Feature extraction and recognition of view data can leverage on edge AI technology. Subject of the analysis can be people and vehicles in real-time videos, and identification, comparison and alarm with black and white list database. The whole network deployment, dynamic detection and trajectory tracking of targets can be realised.

6 CLOUD NATIVE ORCHESTRATION AND LIFECYCLE MANAGEMENT FOR TELCOS

As Cloud Native infrastructure evolves to container-based infrastructure, the increasing number of containers and their dynamicity make it more and more difficult to operate with

traditional tools and methodology. The industry is evolving towards an automated self-healing environment called zero touch network and service orchestration.

6.1 Recovery Oriented Computing

In the ultimate case, the operator just observes the self-healing network, thanks to the 100% zero-touch and closed loop automation. Their focus moves from MTTF (mean time to failure) to improve the MTTR (mean time to repair) process by continuous automation improvements. The operator becomes an automation designer!

Cloud Native orchestration is adopting what is called Recovery Oriented Computing: *"ROC takes the perspective that hardware faults, software bugs, and operator errors are facts to be coped with, not problems to be solved. By concentrating on MTTR (rather than MTTF), ROC can reduce recovery time and thus offers higher availability." [5]*

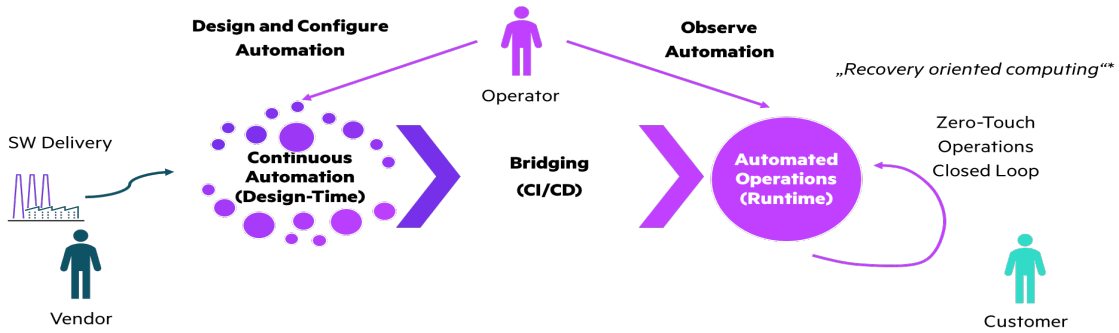


Figure 11: Recovery Oriented Computing

6.2 CI/CD – Continuous Integration/Continuous Delivery

Zero-Touch Management is enabled by Continuous Automation within a CI/CD environment.

The operator observes the automation behavior and initiates changes, e.g.

- Deployment changes (scale, heal etc.)
- Parameter changes
- Automation policies
- Software requests
- Software upgrades
- Reconfiguration at runtime

In addition, a "vendor-self-service" should allow the vendor to upload, deploy and test his new Software (e.g. in a pre-production environment), before the new release is integrated by the operator.

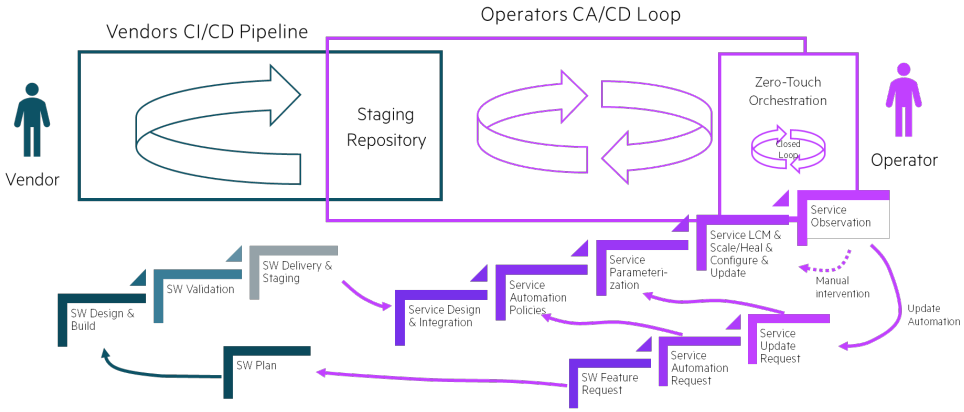


Figure 12: CI/CD Environment

Although CI/CD is commonly known and frequently used for Software development, it was generally adopted by the Telco industry with the advent of 5G networks. This adoption is powered by the virtualisation of mobile Network Functions (NFs) as well as the decomposition of these NFs into finer granular virtualised functions running as micro-services. From a mobile operator perspective, CI/CD enables the automation of Virtual Network Functions (VNFs) lifecycle management including the initial deployment, and reconfiguration at runtime.

An automated CI/CD environment becomes essential with the evolution towards more complex networks that rely on stateless CNFs. In fact, these complex systems require more frequent updates to the different CNFs in order to modify the service, and to answer clients' needs. CNFs orchestration and lifecycle management require continuous supervision and frequent interventions to make sure the different services are running correctly, and to ensure they answer clients' needs. Therefore, an automated and continuous supervision, maintenance, and upgrade process for mobile network CNFs should be used to ensure the following:

1. Reducing deployment and maintenance costs
2. Simplifying components' lifecycle management
3. Automating the update of running network functions

4. Enabling more frequent network changes in a transparent way

Such an automated CI/CD environment as described above can be called a CA/CD (Continuous Automation / Continuous Deployment) Loop. Since it runs at the Operator the overlap to the CI/CD pipeline of the vendor would be the staging repository. Ideally it leverages a Zero Touch Orchestration process, but it can also incorporate multiple separate orchestration solutions.

6.3 Heterogeneous Container Environment

While ETSI defines 5 use cases in ETSI IFA029 for container usage by VNFs:

- Container-based NFV Micro-Services within the VNF
- VNFC in container on bare metal
- VNFC in container in a virtual machine
- VNFC in a group of containers
- NFVI provides containers on bare metal and VM

We actually observe three most common deployment models on VM:

1. OpenStack VMs, serving VNFs
2. Kubernetes running on one or more VMs, serving container-based VNF (“CNF”)
   - CNFs provided as VM packages, with their own Kubernetes
   - CNFs on top of a shared virtual Kubernetes, setup on VMs upfront. CNFs are managed as for item 3.
3. Kubernetes running on bare metal, serving CNF

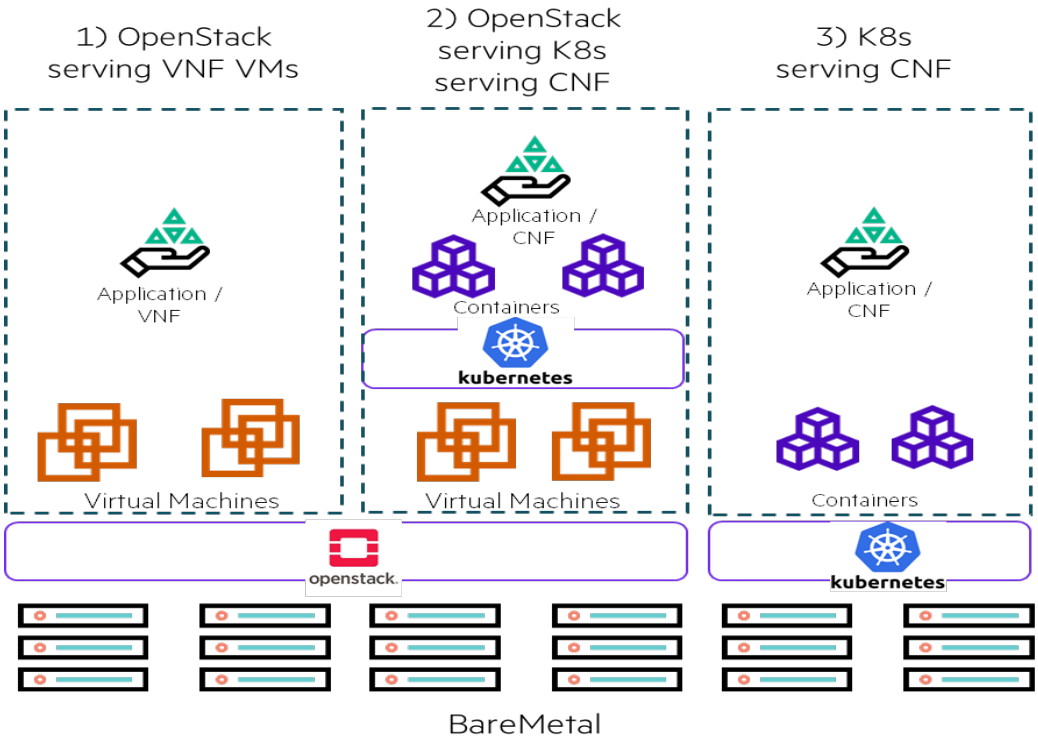


Figure 13: Most Common Cloud Native Deployment Model

6.4 Layered Operational Model and Complexity

Cloud Native infrastructure is typically deployed across multiple sites, heterogeneous cloud environments, multi-vendor equipment, heterogeneous virtualization technologies and providers, multiple applications with different design models and operational tools. Different stakeholders operate various parts of the network: hardware infrastructure, virtualization layer, application layer, and service layer.

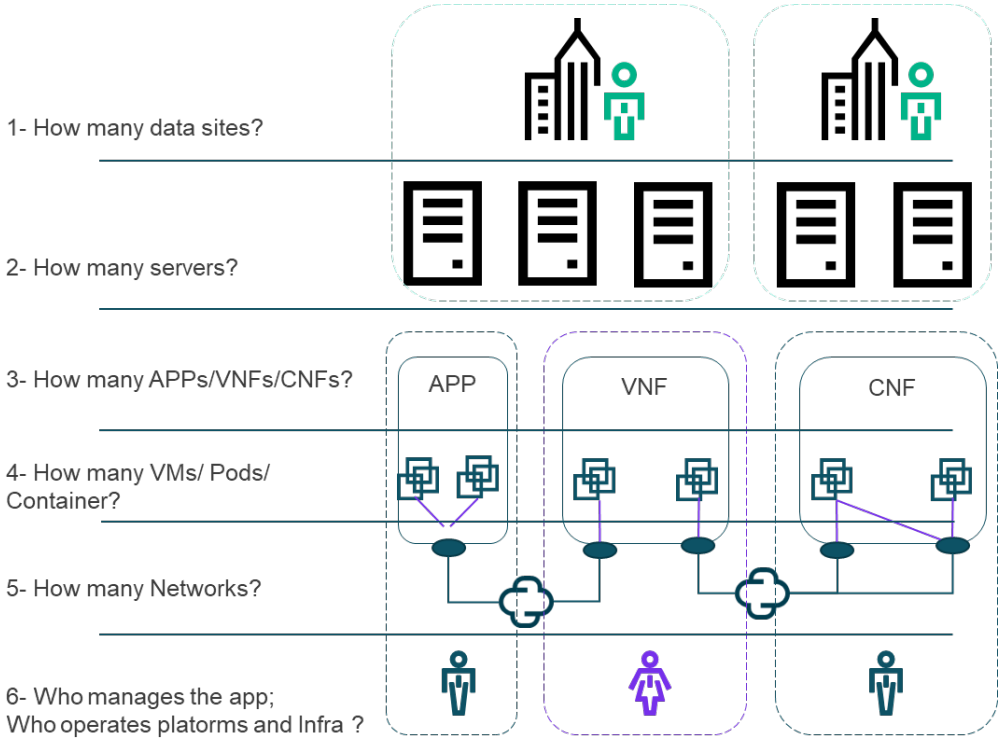


Figure 14: Layered Operational Model

6.5 Federated Information Model

Different technologies co-exist such as SDN, NFV and different vendor products with specific capabilities, so multiple Information Models (IMs) are being used to represent and operate the deployed environment. For example ONF has defined an information model for SDN, while ETSI NFV has another information model for NFV environment [6] and Linux Foundation has a Cloud Information Model [7]. Different models can also be defined for different layers: a resource model versus a service model. These different information models need to be federated if they are to provide an end-to-end view of the cloud native environment, they also need some common definitions, tools (e.g. Papyrus) and languages (such as UML/Json/Yaml), to integrate and update the models. Generally, a set of common models are defined for each technology or layers, with extensions provided either by specific standards, operator needs or vendor implementation.

Several network functions of the RAN or the Core network segments can use a common Information Model; then, extensions to this model are brought for each of these functions depending on their specificities and needs. Indeed, the Fault Configuration Accounting Performance and Security (FCAPS) operations have common aspects for all the network

functions e.g., whether the NF is running correctly or not. However, each NF is characterised by its own configuration parameters, such as the number of radio units managed by a single gNB Distributed Unit, and its own fault alarms and performance counters e.g., number of packets successfully transmitted on the radio interface, or the throughput at PDCP layer. New services are then introduced with model-based descriptions such as TOSCA or Yang descriptors, with behavioral semantics.

From an orchestration and management perspective, specifying an IM and Data Model (DM) for each of the managed functions helps with automating the initial deployment, configuration, and reconfiguration of the network functions at runtime. The common information model provides a formal representation of the NFs, and their common properties, relationships, and operations that can be performed on them. For each NF, the IM is then extended and enriched with properties and attributes that are specific to this NF. For instance, in the RAN segment attributes and parameters related to antenna configuration and radio interface alarms need to be added to the Radio Unit (RU) IM.

Information modelling helps providing visual, traceable, and user-friendly representation of the network functions. It illustrates their main attributes, properties, possible operations, and how they interact between each other. Conversely, data modelling helps to provide a machine-readable representation of the network function. It implements the properties, attributes, and functions defined in the IM to facilitate the management of the network function by the network orchestration and management entity. Standards Developing Organizations (SDOs), such as 3GPP, O-RAN, and ONAP provide common and specific IMs and DMs for the management and orchestration of mobile network functions.

6.6 Lifecycle Orchestration

As new services are being introduced, new equipment and new applications are being deployed by different domain stakeholders, the cloud native environment evolves and orchestration is involved to support these day to day evolutions while delivering service continuity and quality of service across the entire network. From design, to deployment, service assurance and service orchestration across the different layers, cloud native orchestration leverages all the metrics, tools and Artificial Intelligence/Machine Learning capabilities to operate increasingly complex distributed networks efficiently and automatically.

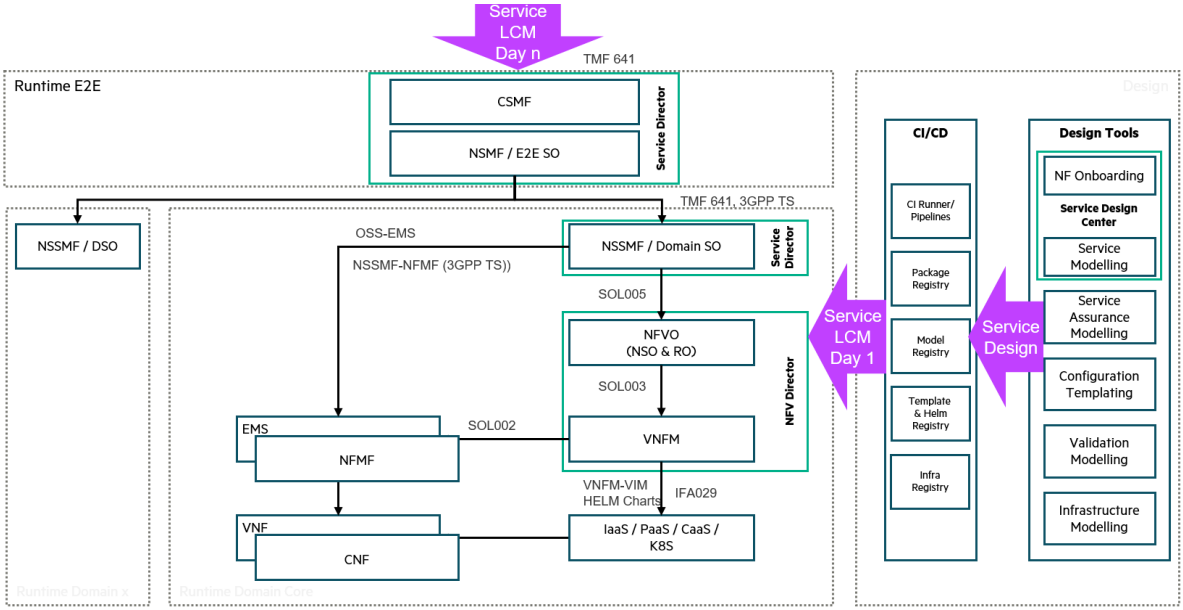


Figure 15: Lifecycle Orchestration

An example of K8s cluster provisioning and creation for carrier-grade Cloud native operator is given below. The process starts when the request for cluster creation is received and ends as soon as the provisioning of the necessary resources, and the initial configuration are done in accordance with the received request.

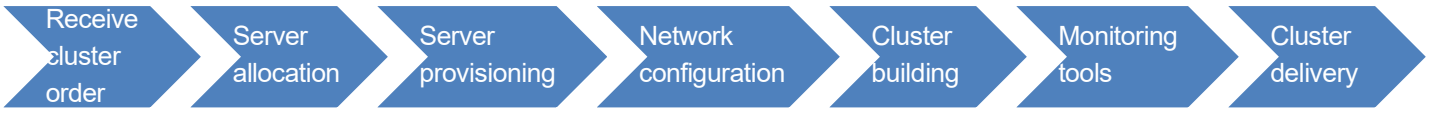


Figure 16: K8s Cluster Provisioning Steps

The required operations include: server allocation and provisioning, network configuration, cluster building, and monitoring tools installation. These operations shall be done while taking security requirements into account. Besides the initial K8s cluster creation and provisioning, service management and orchestration framework are required to monitor, maintain, and manage the running network functions.

6.7 Standards & Open Source Role; Open APIs

Multiple technological standards have been defined which provide standard metrics, KPIs, APIs for service assurance and service orchestration at different levels. The levels are hardware infrastructure management (HIM), Virtual Infrastructure Management (VIM) and

IaaS/PaaS/CaaS but also at the application level with NFV MANO and other OSS capabilities, and the service level with service orchestration, NSMF and CSMF. These topics are covered by 3GPP, O-RAN, DMTF, ETSI, GSMA, TMF, MEF in particular, they have corresponding open source reference implementations within Linux Foundation and other open source projects. Other common standards such as the P4 open source programming language may be necessary to enhance the programmability of the Data Plane with open standard P4 Program APIs.

The support of industry standards and open APIs goes some way to guarantee interoperability and consistency across the cloud native environment and facilitates a homogenous zero touch orchestration.

6.8 Challenges and Opportunities

Cloud native orchestration should address the challenges listed above and provide a unified flexible and efficient way to manage and operate 'zero touch' these hybrid and dynamic environments, while hiding the underlying complexity.

In summary, Cloud Native Orchestration:

- together with AI-Ops, is a means to achieve Zero-Touch operations
- together with the CI/CD pipelines, it provides the Operator with a Continuous Automation capability
- can manage Containers in different Scenarios with a cloud computing platform (such as OpenStack) or on Bare metal
- allows matching virtualised NFs to appropriate hardware, optimizing the use of software enabled infrastructure
- structures management of people, processes and managed technologies
- allows LCM of hybrid Services build on PNF, VNF or CNF
- is part of a customer's Automation Operations Platform

6.9 Protocol Evolution; Networking-as-a-Service

The internet as we know it from a networking and transport perspective is a concatenation of various compute networks that form the end-to-end system to interconnect clients and servers with each other. The aforementioned clients and servers see an all-IP internet that has been fine-tuned towards a "best effort" approach.

However, with the introduction of Virtualization, especially in the telecommunication realm, a trend can be observed in NFV deployments where layers of traffic encapsulation using VLAN identifiers or IP tunnelling are created.

Network Virtualisation can also support inefficiencies of packet overhead due to the adoption of decades-old technology, i.e. IP, VLANs, tunnels at the core of many NFV frameworks and has brought forward multiple and rather different approaches on SDN (Software defined Networking). Since in reality cloud platforms are established "stand alone" it is not uncommon that each cloud platform uses a different SDN overlay technology.

In today's far stretched telco networks that enclose RAN Cells, Mobile Edge Clouds, aggregation and consolidation sites as well as multiple network cores with their OSS and BSS platforms, complex E2E services might get composed from sub-services running in different parts of the network/platforms using various SDNs. In this case network virtualization can become a massive inhibitor.

Especially when looking at certain 5G interfaces (e.g. N3 between the gNB and UPF or N2 between the gNB and AMF) a mix of multiple network, tunnelling and session control protocols are in place demonstrating that the requirements towards 5G (and beyond) systems have pushed traditional protocols and isolation approaches to their limits.

With future networks there are methodologies coming along which introduce fundamental changes to how networking is executed and can be re-programmed without much or no state change in the underlying switching fabric. These approaches should be able to support the establishment of a true Network as a Service (NaaS) layer that will provide virtualized networking services at minimal overhead fulfilling the requirements of network specific workloads.

7 TELCO API ORCHESTRATION

The cloudification of the Telco infrastructure brings many novelties and advantages. It can reduce costs, simplify operation and improve time to market for example. Anyway, there is another aspect that brings further possibilities for a Telco to innovate. Telco API exposure is nowadays a concrete possibility because it leverages on shared approach by the Telco ecosystem. The new 5G mobile infrastructure is indeed greatly based on a service-based architecture that well matches with the cloud paradigms of the underlying infrastructure. It is

easy to understand how both the infrastructure and the network applications running on top of it are, once cloudified, ready to be exposed. As a reference and for further details on the cloudified open infrastructure and architecture see chapter 4 and 5 of [1].

This chapter focuses on the experience for the definition and first trial implementation of an exposure layer. For deepening on Telco APIs and the SDOs working on it, see chapter 3 of [1].

7.1 API Orchestration

At its core, API orchestration is the act of integrating applications into a single and unified offering. Typically, it is used to merge API calls into a single frontend, automate processes, or merge multiple internal APIs from a user experience perspective.

Telco's IT infrastructure is an enabler by exposing data assets as a service to a broader audience. IT can enable lines of business to self-serve.

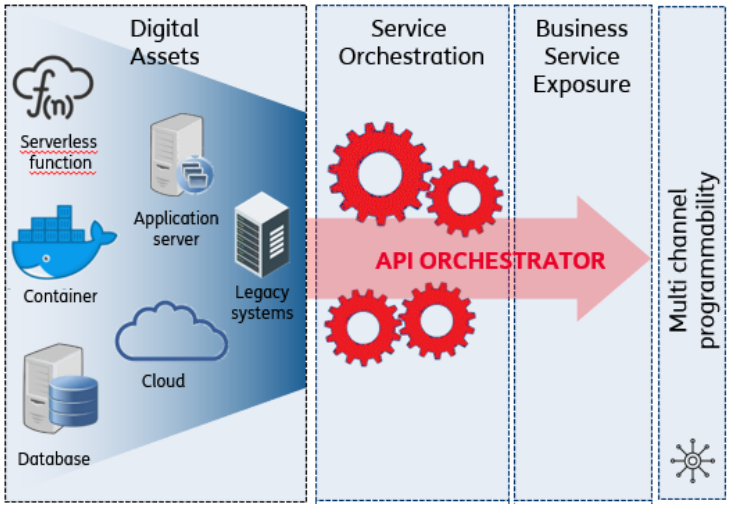


Figure 17: API Orchestrator

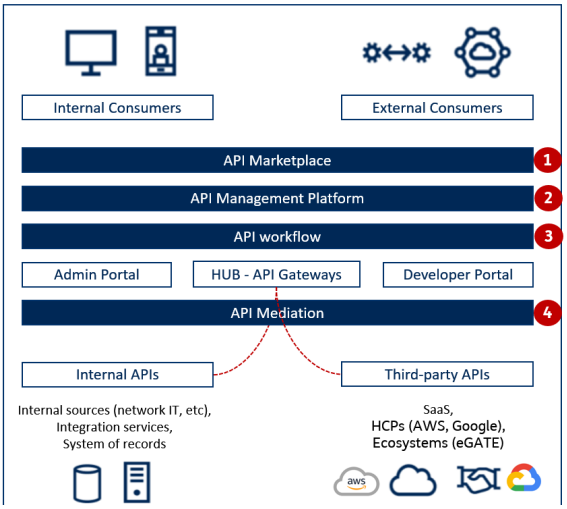
One main concept behind exposure is the possibility to increase developers' productivity through reuse. An API driven approach is consistent with a service-oriented approach whereby logic is distilled to its constituent parts and reused across applications. This prevents duplication of effort and allows developers to build on top of each other's efforts.

An API-led connectivity approach recognises that there is not a one-size-fits-all architecture. This allows connectivity to be addressed in small pieces, and for that capability to be exposed through the API or microservice.

Greater agility through loose coupling of systems can be achieved having separate API tiers. This allows a different level of governance and control to exist at each layer, making possible simultaneous loose-tight coupling.

The API orchestration first goal is to decouple business logic to technical aspects reducing time to market for new business services. It transforms a set of technical APIs into easily reusable business-oriented APIs by implementing the related workflow and hiding technical complexities.

API exposure is conceived with a multi-layer approach. The Telco API producers are inside the single network domain. Each domain could have an API mediation and gateway to expose them towards an upper aggregation layer. Those APIs then need to be managed to be properly exposed to developers via an API marketplace.



Component scope		
1	API Marketplace	Expose a searchable API marketplace for both internal and external consumers
2	API Management Platform	Uses an API management platform to provide administrative and developer portals
3	API Workflow	Uses a workflow system to • Fast create workflow definitions • Run workflow instances • Manage workflow instances and definitions
4	API Mediation	uses API gateways and other mediators to implement the API mediation capabilities

Figure 11: API Layers

API Orchestration capabilities may be partially overlapped with service orchestration capabilities. The boundary between the two systems is defined according to specific needs.
API orchestration and API gateway don't exactly have the same purpose. API gateway focuses on exposure while API Orchestration focuses on delivering value-added business services. Partial overlap can be identified in the exposure of business services.

7.2 Characteristics of an API Orchestrator

Orchestrating APIs is not just exposure, it is integrating and orchestrating them to expose a complex service. The goal is to expose a simple interface for a complex service while it requires a flow composing different APIs.

Table 1: Some Characteristics for an API Orchestrator

Characteristic	Description
Stateful Workflow	Maintain state of application during execution. Built-in or possibility to implement: state machine (stateful apps), decisions, correlation IDs patterns
Persistance	In case of stateful API, how long the state can be maintained
Support workflow patterns	Support for workflow patterns described in the specific section of this document (Composer, branching, error handling, parallel processing ...)
Support async patterns	Support for Asynchronous patterns described in the specific section of this document (Polling, Webhook...)
Scalability	The platform must be able to scale out and in automatically
Availability	The platform must be able to be resilient to HW and SW failures
Third parties integration	Integration of third parties' products / software / databases
Plug-In Architecture	Possibility to extend features by develop your own module (integrations, optimisation, closed source – legacy)
Serverless workflow	No need of extra infrastructure components. Workflow run on the same platform
Workload Administration	Coordination of workloads among distributed components and underlying architecture (K8s, FaaS, PaaS)
Standard	Is possible to develop API based on standard Specification like: OAS, RAML
Speed in workflow implementation	Availability of features to accelerate the workflow development process
Easy Programming	Development Learning Curve
Easy Management	Easiness of use during high-level platform management
Documentation	Possibility to describe the API (input and output parameters). Support of Swagger or equivalent UI tools for testing and automatic client SDK code generation (boilerplate-code).
Vision	Product ability to integrate and discover new / innovative (i.e. support of AI/ML, GraphQL, WebSockets, Edge Computing ...)

7.3 Products Examples

API Orchestration platform capabilities are a mix between the following three categories of software products:

- Integration centric
  - Full Lifecycle API Management such as Apigee, Mulesoft
  - Enterprise integration platform such as Boomi
- Process centric
  - Robotic Process Automation such as Pegasystems or Camunda

Here are some tools as an example:

Mulesoft: MuleSoft, which Salesforce acquired in 2018, offers the Anypoint Platform as its full lifecycle API management offering, which combines API management and integration capabilities in a single platform. A packaged option providing only API management is also available. In 2019, MuleSoft introduced Anypoint API Community Manager to create and grow an ecosystem of API consumers and drive adoption of API products. MuleSoft offers Istio support via Anypoint Service Mesh. MuleSoft sells its platform both directly and through an ecosystem of partners. It has midsize and large customers worldwide.

Apigee: The core Apigee API management platform is available for public cloud, private cloud or data centre and hybrid (customer-managed runtime and Google-managed control plane) deployment. Apigee Sense (for bot protection), Apigee API Monetization and Apigee Advanced API Ops (which is in beta at the time of writing) are also part of the Apigee platform. There are also adapters for Envoy and Istio to enable API management features in both. Google's roadmap for Apigee includes delivering a fully managed API platform for multi-cloud and hybrid deployments, building ecosystems of citizen developers, integrating with marketplaces, and extending Google technologies such as artificial intelligence (AI) and machine learning (ML) to API management. Most Google (Apigee) clients are located in the U.S., Europe, Australia, New Zealand, India and Southeast Asia. The Apigee team markets its offering as a cross-cloud API platform and as a platform for digital business.

Boomi: Boomi is a wholly owned subsidiary of Dell Technologies. Its API management offering is part of its AtomSphere solution, which provides integration platform as a service (iPaaS), master data management (MDM), B2B integration and low-code development capabilities. Boomi's offering is sold globally, and most of its customers are midsize or large organizations.

It supports hybrid/multi-cloud and private cloud deployment of the Atom runtime, but the administration experience is purely cloud-based.

Pega: Pegasystems includes its RPA product within the Pega Infinity platform (version 8.4), which offers RPA along with complementary iBPMS, multi-experience development platform (MXDP), CRM and LCAP capabilities. Pegasystems, based in Cambridge, Massachusetts, U.S., has operations across the world and a focus on large-enterprise customers. Its roadmap includes a complete UI upgrade and a desktop application focused on business users. Additionally, Pegasystems has announced Pega Process Fabric — a completely serverless and distributed process management solution.

Camunda: Camunda Platform is an open-source, workflow and decision automation platform. Camunda Platform ships with tools for creating workflow and decision models, operating deployed models in production, and allowing users to execute workflow tasks assigned to them.
Camunda Platform is a lightweight, Java-based framework. It can be used as a standalone process engine server or embedded inside custom Java applications. It offers non-Java developers a REST API and dedicated client libraries to build applications connecting to a remote workflow engine.

7.4 Orchestration Workflow Patterns

There are many different patterns that can be used to setup a workflow for API orchestration. In the following figure, the main logical constructs that can be adopted to implement business logic are represented.

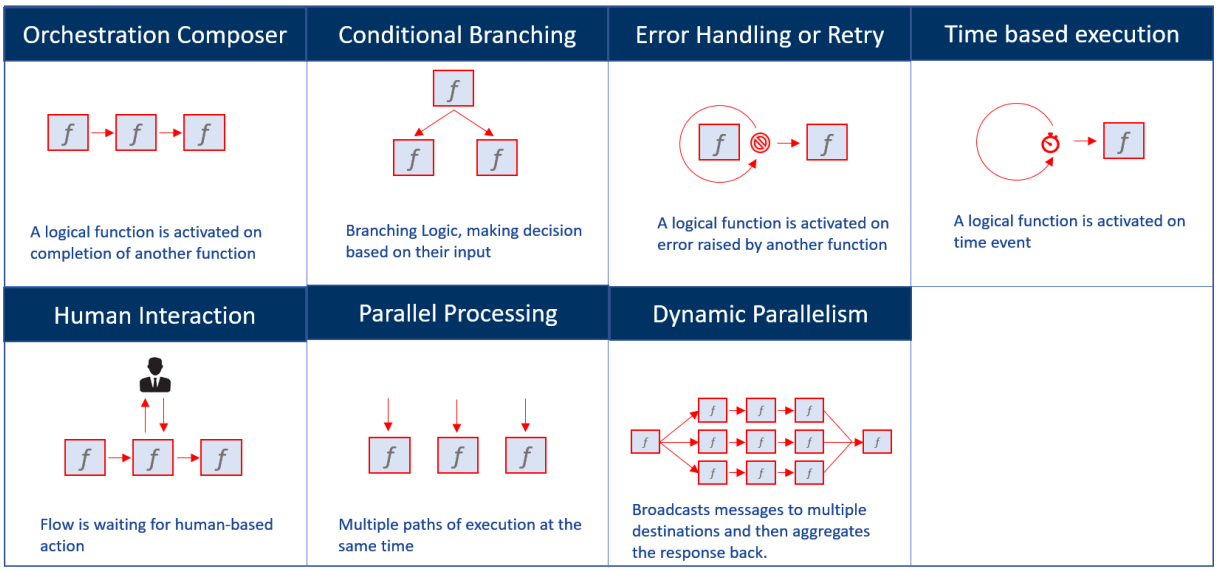


Figure 12: Main Logical Constructs

The following patterns are relevant.

Async Pattern - **Client Polling / Stateful Workaround**: Client asks the API endpoint for a resource, which needs to be created by an asynchronous job on the back-end. The API endpoint hides the job complexity on creation and process management, immediately returning to the Client a feedback. The Client, to get the requested resource status, must check it periodically by invoking an opportune API call. Finally, after a few tries, when the request is fulfilled, the Client can directly get the result of the requested service.

Async Pattern – Webhook with Subscriptions: The client asks a Mediator for a long running job. The job is actually exposed by a third-party service which asynchronously notifies when the job is done via a unique Mediator webhook (the same «static» endpoint for the whole pool of clients' requests). The Mediator creates, sends and traces request ids in order to correlate responses with the initial client request.

Long Running Distributed Transaction Pattern – SAGA: it is a stateful long running process that typically includes a state machine definition. Incoming messages for the SAGA need to identify which state they are intended for and the payload that will be applied as input to the state machine (along with the current state). It is a very common way to implement transactional state changes that spans across multiple microservices.

SAGA with Choreography: Every Service sends and listens for events incoming from other services. Can be implemented either with Message Bus/Broker (Pub/Sub) or with REST API.

SAGA with Orchestration: Orchestrator sends message directly to involved services, tracking the transaction states, managing rollback if error occurs.

7.5 Example Use Case

The considered scenario is for developers requiring the deployment of their application at the Telco Edge. The goal is to provide a simple Telco API to ask the Telco platform for deployment of an application at the Edge and the activation of the Local Brake Out (LBO) to route the traffic from the Edge UPF to the local data network.

The exposed API is intended to be used by an external platform offering a deployment service to the customer. The usage of the external platform should simplify the Edge deployment allowing an easy definition of the geographical zones supporting the Service. Today developers, while deploying an application on a Cloud platform, need to select the appropriate regional data centre from a list. The Cloud platform deployment tool usually does not provide enough information about the radio «coverage» and the provided network performance (e.g. latency).

The idea is to offer a simplified model to request latency requirements via the Telco API. This allows to request specific performance at the Edge while requesting for the application deployment.

Future evolutions can foresee the integration in the API orchestration workflow also of other Edge APIs interacting with the 3GPP Edge Cloud NFs such as EES, ECS defined by SA6 [8]. This would enable support of Registration, Discovery and Mobility of Applications among different Service Edge Nodes.

8 TELCO CLOUD TESTING & SECURITY

5G networks architectures exist in a highly virtual and automated environment and Telcos are using agile development and test pipelines to reliably deploy, operate and maintain mobile network services within cloud software orchestration and automation frameworks. They bring next generation cloud native Service assurance to the hyperscale and distributed Clouds to the recently transformed service providers eco-system.

Cloud-native, operation and real-time performance monitoring within microsecond accuracy become the trust model for next generation service-based architecture.

This is driving classical network assurance and network visibility platform global transformation with the automation of function and service testing and advanced analytics. The transformation is essential to the agile "Telco Cloud" environment and in a "Lab to Live" context environment and in live network operation mode such as:

- CI/CD/CT (Constant Testing)
- Service activation
- Service monitoring
- Triggered diagnostic
- Extended visibility meta data & analytics
- Telco Cloud Security Towards "Zero Trust" System
- Administration & System Health

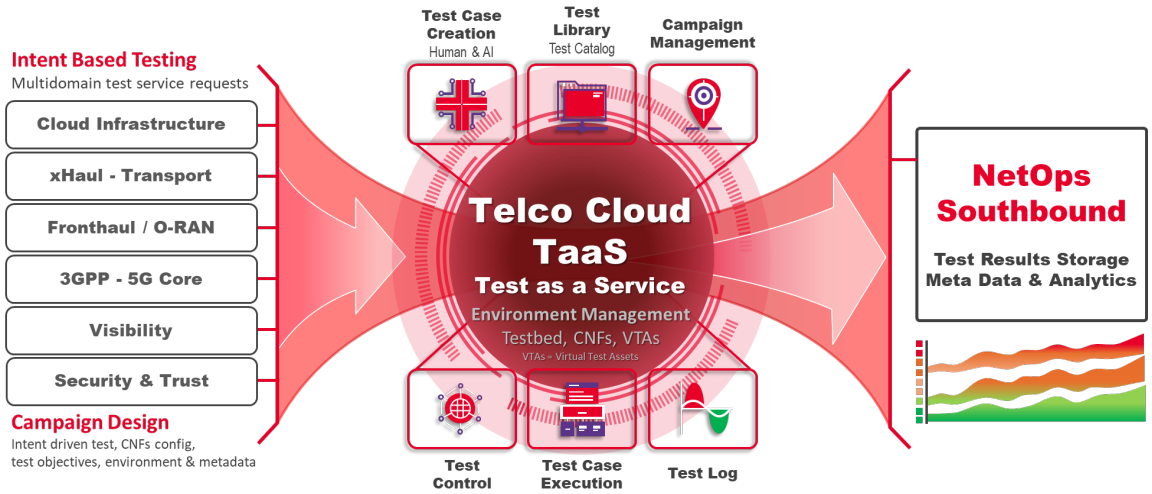


Figure 20: Telco Cloud Test as a Service Platform

Visibility and assurance are no longer silos, bringing modular & open service access from network planning to operation and commissioning, and from user/devices through to RAN/OpenRAN, Transport xHaul, Core/5GCore, data network (DN) and end-user applications. This common Telco Cloud testing and automation framework spans from passive to active testing and from Multiaccess Edge Computing (MEC) to Open Radio Access Networks (OpenRAN) as well as to Private 5G wireless network architectures including hybrid public hyperscale and private cloud, disaggregated and distributed cloud infrastructure.

8.1 Lab to Live Cloud Network CI/CD/CT Process Automation

"Lab to Live" concept is a tight integration and path between lab and live network operation which now are a part of CI/CD integrated processes.

In addition, another new concept in use by Mobile Network Operators is "constant testing" lifecycle complementing the "CI/CD" process automation with "CT" as part of "Lab to Live" cloud network lifecycle framework and from Day 0 through to operation and optimisation processes as described in the picture below.

Supported use models:

- Lab–to–Live deployment
- Test On-demand/Ad-hoc
- Continuous Active Monitoring

Cloud native architecture providing:

- Test Management and Result Analytics
- UE/application throughput/QoS
- Comprehensive 5G core/MEC/gNB emulation

Test Process & Workflow:

- Suite of test plans including 5G and application test
- Test scheduling/execution/results analysis
- RCA – Root Cause Analysis

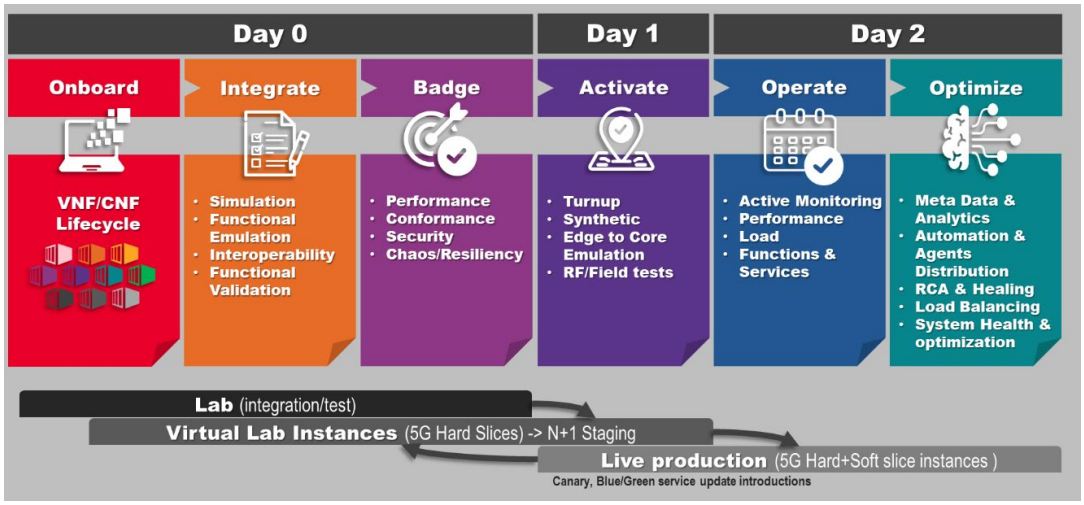


Figure 21: Lab to Live Cloud Network Testing Lifecycle

Distributed & Automated Network Assurance overview:

We explore some of the use cases that have been addressed in the past few months such as:

- Cloud Infrastructure Performance & Capacity Benchmarking
- MEC/Hybrid Cloud Assurance - CI/CD/CT
- O-Cloud OpenRAN
- Service Monitoring
- Triggered Diagnosis & Root Cause Analysis
- Meta Data & Analytics
- Administration & System Health

For the special MEC/Hybrid Cloud Assurance use case - CI/CD/CT, MEC assurance become essential for critical edge compute application and performance and particularly in multi-cloud environment at Carrier/Hyperscale gateway.

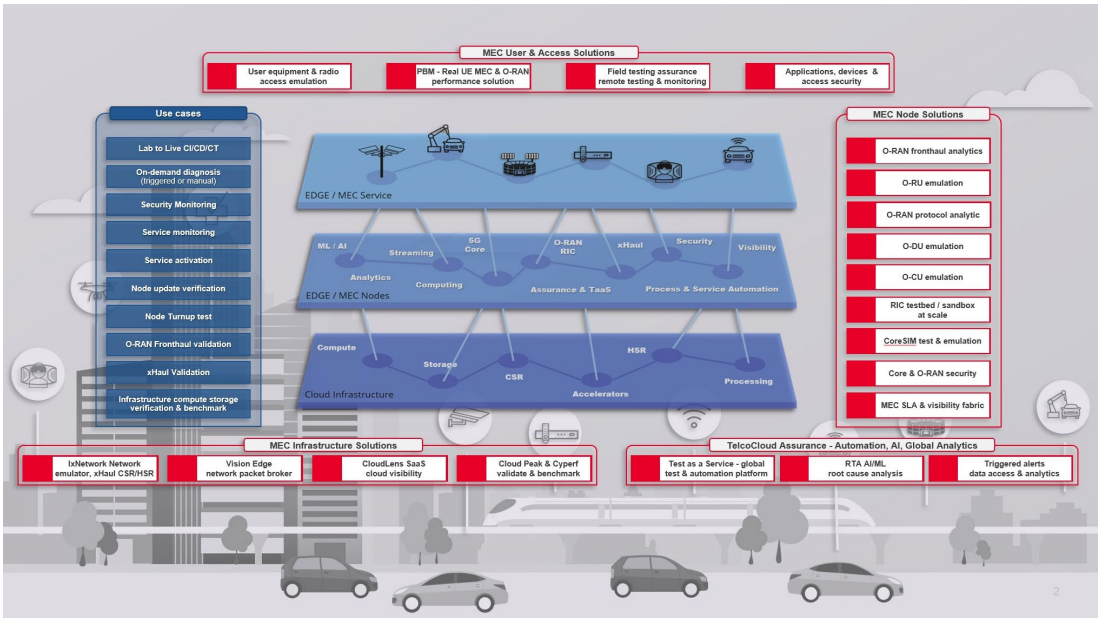


Figure 22: Telco Cloud, EDGE & MEC Next-Gen Service Assurance at Scale

MEC validation platform provides full stack MEC testing & performance coverage including global security assessment, and this is divided in 3 main parts:

1. **Starting from Cloud Infrastructure Validation**
    - Capacity & Performance
        - Latency
        - Bandwidth
        - Resiliency

- Benchmarking
- Scaling
- Secure Access Service Edge (SASE)

2. **MEC Nodes Validation**
   - QoS / QoE Validation
   - Jitter Latency
   - Video & Audio Processing
   - O-RAN RIC
   - 5G Core UPF split / N9 interface
   - xHaul Transport as a Service
   - Extended Visibility
   - Security Assurance Specification (SCAS)

3. **MEC Services:**
   - QoS / QoE Validation
   - Jitter Latency
   - O-Cloud
   - Video & Audio Processing
   - C-V2X
   - Application Security

Global MEC validation at scale overview:

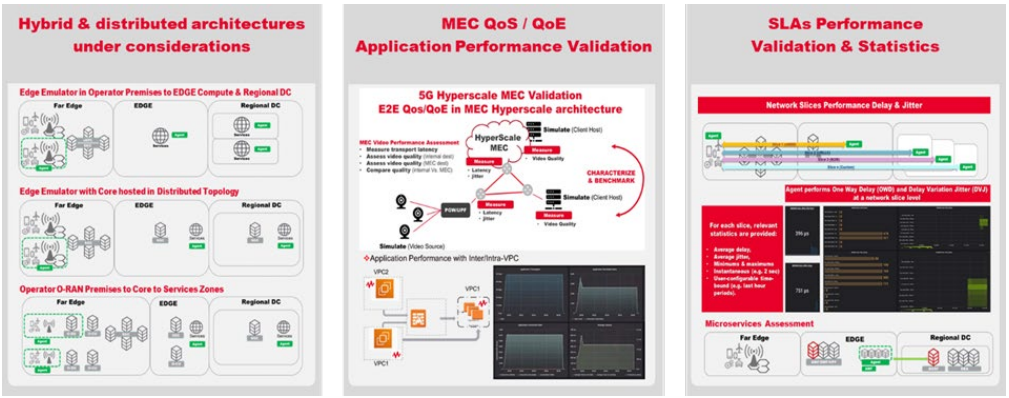


Figure 23: MEC Hybrid Architecture, Distribution and Service Validation

1. **Continuous path latency monitoring** for low latency application assurance/SLAs
2. **Segmented path latency** for rapid problem isolation, root cause determination and resolution
3. **MEC/Cloud Gateway traffic monitoring** volume by route and type (UE-local/region, Cloud-internet) for SLA reporting, interconnect billing and capacity planning
4. **Enriched session level data** subscriber/UE/device

5. **Correlated transport and multi-cloud performance**
6. **In-session test** CP/UP to MEC/Cloud/DN

8.2 Hybrid Cloud Infrastructure & SLA Performance Validation & Benchmarking

It is known that Telco Clouds are going to be diverse and hybrid using multiservice providers as described in [1].

The performance level of such diverse and hybrid architecture must be assessed and characterised to guarantee the service level agreement and end service user expectations.

All these services and hybrid configurations and as much as systems that need to be validated for are:

- O-Cloud Validation including O-DU / O-CU / RIC
- Hybrid Architecture Performance Validation
- Application Performance Validation - MEC QoS / QoE
- SLAs Performance Validation & Statistics

8.3 Telco Cloud Infrastructure Visibility

Network visibility is also fairly transforming. As the world moves to 5G, the number of mobile subscribers is increasing rapidly. Subscribers use mobile devices for more complex and substantial tasks than before including 4K video like Netflix or Amazon and social streaming applications such as YouTube, Vimeo, Facebook and Instagram in addition to services such as AR/VR/XR, i4.0, Cellular V2X, Video Surveillance and the Cellular IoT in general.

Obtaining the visibility into 5G-SA Core networks introduces challenges and barriers. Here is the list of challenges Mobile Network Operators are facing implementing high-performing 5G network visibility today:

1. **Hybrid architecture**
   Service provider operation teams are increasingly challenged with gaining network and application visibility across physical, virtual and cloud infrastructures.

2. **Packet data availability**
   Probes that monitor the 5G core require consistent network packet flow delivery to monitor 5G networks effectively. Yet, obtaining 5G core network packet access is difficult

due to encryption, agility and data volume.

3. **Vendor specific vTAP**
   As a method for 5G access, network function providers (NFPs) are building their own virtual packet access solutions that capture 5G southbound interfaces (SBI) traffic at the packet level before forwarding to probes.

4. **Integration complexity**
   However, each NFP has a different encapsulation method for packet delivery to probes, and every probe has different requirements for data ingestion.
   Thus, as more NFPs and Probe vendors need to communicate together, the integration complexity increases exponentially.

5. **5G SBA data access reliability**
   In addition, the 5G service-based architecture (SBA) data access complexity may add latency and indeterminate packet delivery to the probes. The probes already have a complex job working on real time tasks.

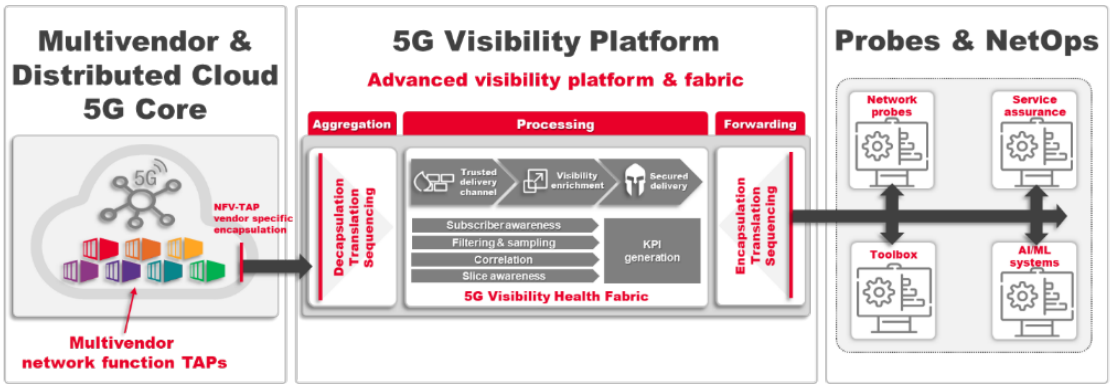


Figure 24: Multivendor and Distributed Cloud 5G Visibility Platform - 5G Visibility for Multi vTAP Network Function Providers

8.4 Telco Cloud Infrastructure Security Validation

This is yet another subject that Telco Cloud is totally transforming which needs full CI/CD/CT integration pipeline and from network Development to Deployment and Operation.

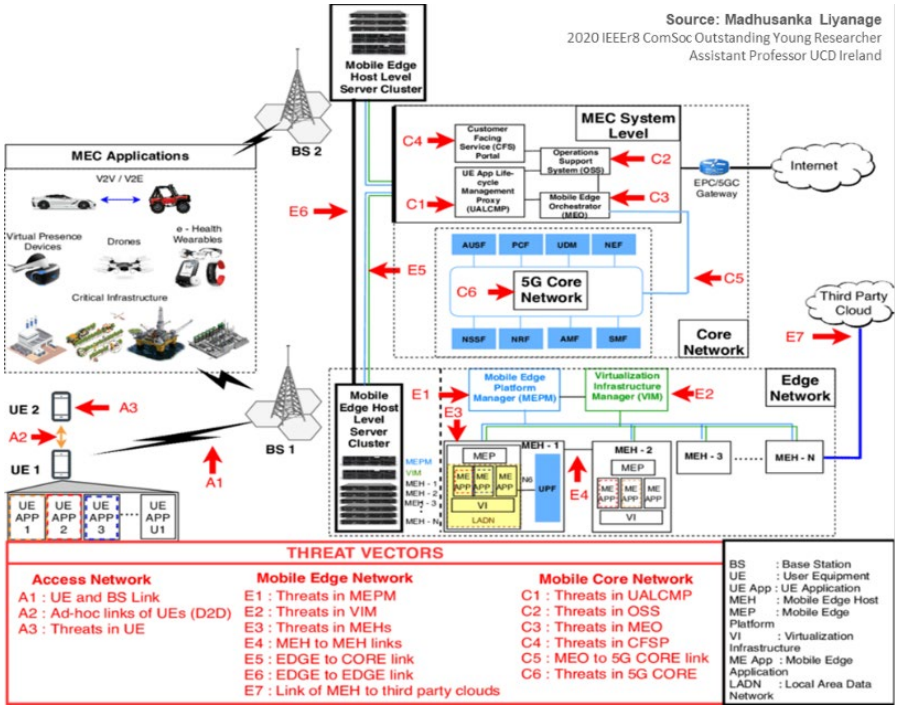


Figure 25: ETSI ISG MEC Access, Edge and Core Threat Vectors Taxonomy

The Telco Cloud and particularly the MEC platform poses several security challenges and the side effect of drastically extending the attack surface and global vulnerability.

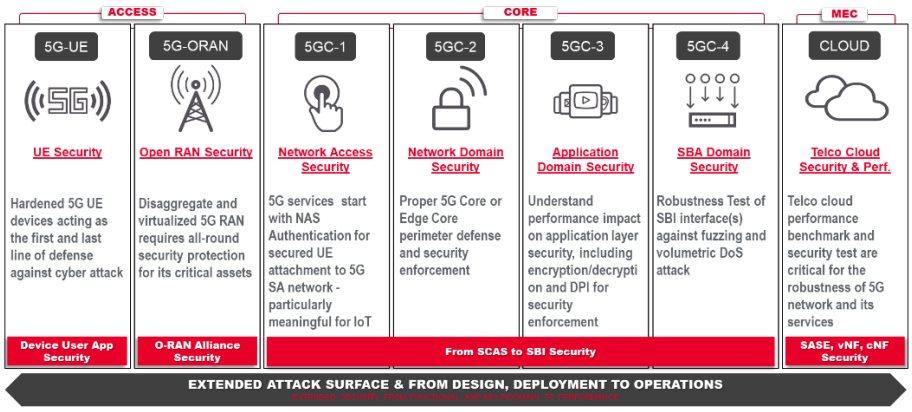


Figure 26: 5G Specific Security Coverage & Zero-Trust Network Strategy

The Telco Cloud and MEC platform needs to simultaneously fulfil 3GPP-related security requirements and extend towards Open Fronthaul, OpenRAN, multidomain security assessment and to the IT virtual security assets (vFGW, vFWA, Secure Access Service Edge (SASE) framework…) while ensuring cloud performance and trust in order to build up meaningful service provider's zero-trust strategy.

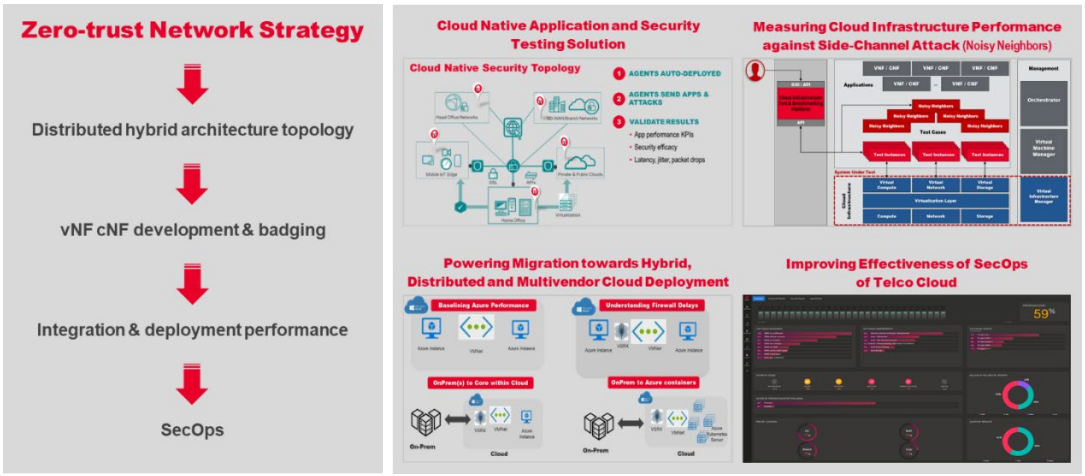


Figure 27: Zero-Trust Network Strategy Overview

9 SUMMARY

This paper summarises some of the experience gained by NGMN Partners on the cloudification process. This represents an interesting map of the areas currently receiving the most attention. It appears clear that the level of adoption and maturity is very different according to the area of interest.

In terms of innovation, many steps must be covered to have a standard adoption of cloud native techniques to build up network functions. This process is still ongoing. We expect a set of fully cloudified and interoperable NFs by different vendors leveraging on the same infrastructure and adopting the same models. Actually, this is still to come although some common path for their development is becoming more and more consolidated. Nevertheless, the telecommunication ecosystem is evolving towards fully virtualised cloud native networks, including RAN and Core networks. The adoption of such solutions for carrier-grade deployment is currently being investigated by several operators and vendors.

Another technological aspect whose maturity is not ready yet is cloud native orchestration. Many tools and techniques are available. The basic components and patterns seem very aligned in the different solutions, but many challenges are still to be solved. Integration of different components, management at different levels of the infrastructure, domains with different level of maturity, together with the challenge of a softwarised ecosystem still represent a challenge to win.

Considering the topics covered by this paper, it is evident that a big attention is given to aspects that are related to new business opportunities such as edge, telco API, service assurance, testing and security. This picture well represents the strategy of the Telco operators to go beyond connectivity. Telco edge deployment is an important reality that is happening to support the deployment of network functions and applications considering how complex applications can be decomposed on-premises, at the edge or in the cloud. Telco API exposure can be a driver to expand the presence of Telcos in the ecosystem gaining a distinctive role. Security and testing are key elements to share the platform and keep pace with the software evolution.

As a conclusion, the cloudification of the network and the evolution of the ecosystem is proceeding with activities that are at different stage of maturity. This is due to many aspects such as the level of readiness of the involved stakeholders and the level of maturity of the adopted technologies. Considering the experiences in this document, it is clear that cloudification is not just meant for cost reduction but it is indeed an opportunity to expand business opportunities leveraging on the flexibility of a cloudified network.

LIST OF ABBREVIATIONS

5GC	5G-Core
AF	Application Function
AMF	Access and Mobility Management Function
API	Application Programmable Interface
BBU	BaseBand Unit
BSS	Business Support Systems
CA/CD	continuous automation continuous deployment
CAPEX	Capital Expenditure
CI/CD	continuous integration continuous deployment
CM	Configuration Management
CNF	Cloud-native Network Function OR Containerised Network Function
COTS	Common of the shelf
CSMF	Communication Service Management Function
CU	Central Unit
CU-C	CU Control Plane (or CU-CP)
CU-U	CU User Plane (or CU-UP)

cVNF	Cloudified Virtualized Network Function
DDoS	Distributed Denial of Service
DM	Data Model
DMZ	Demilitarized Zone
DN	Data Network
DoS	Denial of Service
DPDK	Data Plane Development Kit
DU	Distribution Unit
eMBB	enhanced Mobile BroadBand
E-UTRA	Evolved Universal Terrestrial Radio Access
EVPN	Ethernet Virtual Private Network
FCAPS	Fault Configuration Accounting Performance and Security
FM	Fault Management
gNB	Next Generation NodeB
GNBCUCPF	Next Generation NodeB Central Unit Control Plane Function
GNBCUUPF	Next Generation NodeB Central Unit User Plane Function
GNBDUF	Next Generation NodeB Central Distribution Unit Function
GSMA	GSM Association
HCP	Hyperscale Cloud Providers
ICT	Information and Communications Technology
IM	Information Model
ISP	Internet Service Provider
LBO	Local Breakout
LCM	Lifecycle Management
MAC	Medium Access Control
MANO	Management and Orchestration
MEC	Multi-Access edge Compute
mMCT	massive Machine Type Communication
MNO	Mobile Network Operator
NbR	Name-based Routing
NF	Network Function
NFMF	Network Function Management Function
NFV	Network Function Virtualisation
NRF	Network Repository Function

ng-eNB	Next Generation evolved NodeB
NG-RAN	Next Generation RAN
Non-RT RIC	None Realtime RIC
n-RT RIC	Near-Realtime RIC
NSA	Non-Stand-Alone
NSMF	Network Slice Management Function
NSSMF	Network Slice Subnet Management Function
O-Cloud	Open Cloud SW
O-CU	Open Central Unit
O-DU	Open Distribution Unit
On Prem	on-premises
OPEX	Operational Expenditure
OSS	Operational Support Systems
OTT	Over-the-Top
P4	Programming Protocol-independent Packet Processors
PaaS	Platform-as-a-Service
PDCP	Packet Data Convergence Protocol
PHY-H	Physical Layer - Higher
PHY-L	Physical Layer - Lower
PM	Performance Management
PNF	Physical Network Function
QoS	Quality of Service
RAN	Radio Access Network
RDMA	Remote Direct Memory Access
RF	Radio Frequency
RIC	RAN Intelligent Controller
RLC	Radio Link Control
RRH	Remote Radio Head
RU	Radio Unit
SA	Service Assurance
SBA	Service-based Architecture
SCP	Service Communication Proxy
SDN	Software-defined Networking
SMO	Service Management and Orchestration

SMOF	Service Management and Orchestration Function
SR-IOV	Single-Route Input/Output Virtualization
TCO	Total Cost of Ownership
Telco	Telecommunication Company
UE	User Equipment
UPF	User Plane Function
URLLC	Ultra-Reliable Low Latency Communication
VNF	Virtualized Network Function
VM	Virtual Machine
vRAN	Virtualized RAN

REFERENCES

[1] NGMN Alliance, "NGMN Cloud Native Enabling Future Telco Platforms V5.2", May 2021, https://www.ngmn.org/publications/cloud-native-enabling-future-telco-platforms.html

[2] NGMN Alliance, "ODiN–Operating Disaggregated Networks", October 2021, https://www.ngmn.org/publications/odin-operating-disaggregated-networks.html

[3] A. Wiggins, "The Twelve Factor App", 2020, https://12factor.net

[4] The FUDGE-5G Consortium, "Fully Disintegrated Private Networks for 5G Verticals (FUDGE-5G)", Horizon 2020 5G-PPP Project, https://www.fudge-5g.eu

[5] https://www2.eecs.berkeley.edu/Pubs/TechRpts/2002/5574.html

[6] IFA015, https://docbox.etsi.org/isg/nfv/open/Publications_pdf/Specs-Reports/NFV-IFA015v2.1.2 - GR - Info Model Report.pdf

[7] CIM, https://cloudinformationmodel.org/

[8] 3GPP TS 23.558, Architecture for enabling Edge Applications (EA)