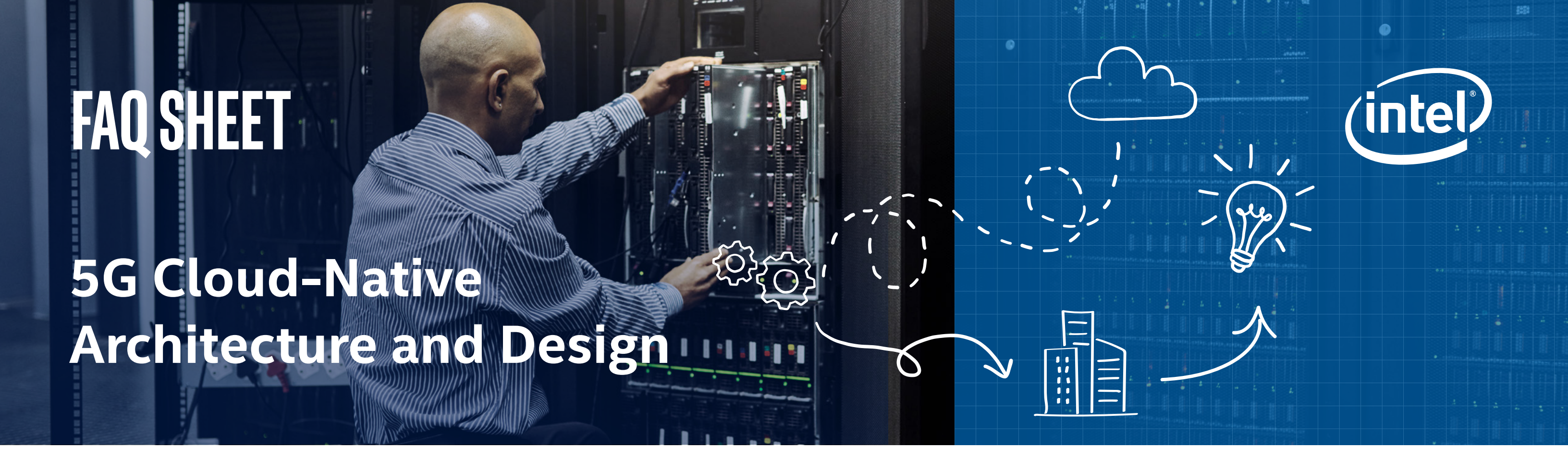


FAQ SHEET

5G Cloud-Native Architecture and Design

Q What is “cloud native” and why is it important to communications service providers (CoSPs)?

A The Cloud Native Computing Foundation (CNCF) defines cloud native as “scalable applications” running in “modern dynamic environments” that use technologies such as containers, microservices, and declarative APIs. More specifically, the cloud-native approach uses lightweight, standalone and executable software packages (**containers**) that can be **integrated and deployed continuously**—resulting in significant operational efficiencies. These containers are easy to define, install and upgrade by specifying infrastructure as code (using, for example, a Helm chart or Ansible), and can be orchestrated and managed throughout their lifecycle in large, dynamic environments. Kubernetes is currently one of the most popular container-orchestration platforms and was the first CNCF project. Cloud native delivers web scale, rapid prototyping and rapid development, high availability and responsiveness, and strong resilience and flexibility through autonomous and self-healing capabilities. These benefits stem from virtualized network functions (VNFs) evolving to Cloud-as-a-Service (CaaS) and containerized network functions (CNFs).

Q What are the advantages of CNFs compared to virtual machine (VM)-based functions?

A The advantages are best described by the companies already using CNFs.

- A leading provider of mobile wireless telecommunication equipment and solutions has deployed a web-scale, cloud-native platform capable of delivering 5G core solutions in the public cloud. It enables integration of 5G standalone, 5G non-standalone, 4G, 3G, 2G and broadband network gateways. The platform supports a variety of vertical markets using network slicing and its resilience minimizes service interruption or downtime.
- A media services provider and production company is using cloud-native computing to add new features and perform software upgrades with no service interruptions, using canary testing. Cloud native enables a seamless transition to service-based architecture (SBA), where microservices can be selectively deployed.
- An executive at a major CoSP has found that evolving the core network beyond virtual machines (VMs) to CNFs and designing its software architecture to run on cloud-native technology has resulted in new levels of operational automation, flexibility and adaptability.
- A global telecommunications equipment manufacturer (TEM) believes that the industry’s evolution to cloud native will support the agility needed to manage workloads dynamically at the edge, as required for many new 5G use cases.

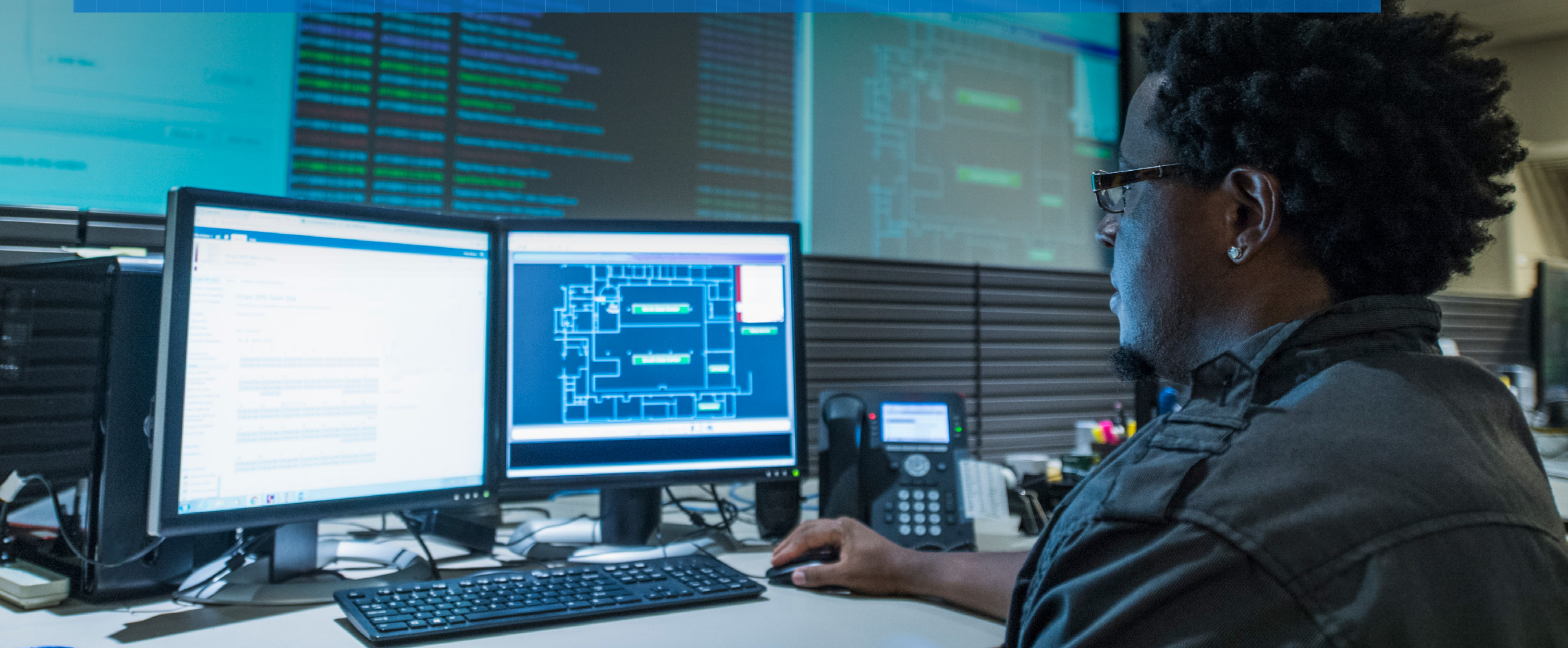
To conclude, the decomposition of software enables simplified and faster lifecycle management of independent microservices, and better resource utilization. Each microservice scales independently and can be upgraded independently, leading to increased speed and agility.

How is cloud-native code different from legacy code?

VM-based legacy applications are "heavy" compared to lightweight cloud-native applications. That is, on a VM, the application is packaged with the entire guest OS, resulting in a bigger footprint. In addition, the infrastructure has hypervisor overhead. In contrast, cloud-native applications share the host kernel/OS and the containers include only the needed user space code; hence the footprint is smaller, and applications are lightweight. Because of this, **cloud-native applications spin up quickly** compared to VM-based legacy applications.[1] Another difference between VM-based legacy applications and cloud-native applications is the **agility of continuous integration and continuous deployment** (CI/CD). Cloud-native code with its microservice architecture and CI/CD CaaS infrastructure enables a faster and more agile update process, using canary testing. Further, compared to VMs, a major benefit of containerized microservices is the ability to orchestrate the containers so that separate lifecycle management processes can be applied to each service. This allows for each service to be versioned and upgraded independently, as opposed to upgrading the entire VNF in a VM. The container scheduler determines which individual services have changed and deploys only those specific services.

Why can't traditional IP-routing networking cope with microservices?

Instances of microservices are created, moved, and stopped far more quickly than traditional routing can handle. Instead, cloud-native code uses a service registry to register services and cloud infrastructure open software tools to find other microservices and to perform health monitoring and auto scaling. Also, with advanced cloud-native tools such as a service mesh (such as Istio), cloud-native code can focus on business logic and leave the transport logic to the CaaS tools.



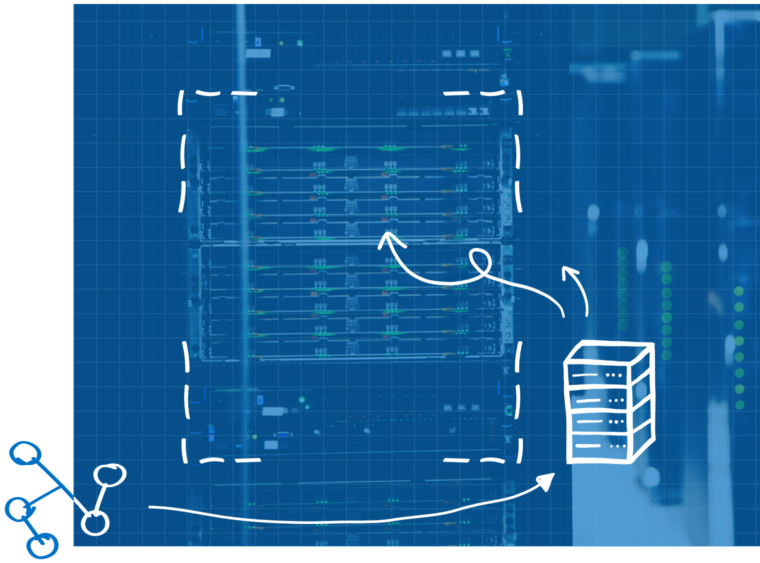
Q Are there tools out there to help telecom operators manage cloud-native deployments?

A Cloud-native tools like Cloudify 3.4—often referred to as Ansible for cloud—comes with a new set of features that enables the pushing of updates to both the application and the infrastructure itself. Cloudify includes a robust set of new features, network virtualized function (NFV) specific plugins and blueprints showcasing modeling of VNFs and service function chaining using the Topology and Orchestration Specification for Cloud Applications (TOSCA)[2]

Q Are there specific VNFs that are better suited for containerization than others, or does it work for all VNFs?

A Both management plane and control plane VNFs can be containerized. However, the control plane VNFs will significantly benefit from Kubernetes plug-ins that enhance performance. Some performance-optimizing plug-ins developed by Intel and then open sourced are listed below:

- Kubernetes SR-IOV Plug-in (I/O acceleration)
- Kubernetes Topology Manager Plug-in (NUMA enhancement)
- Kubernetes User Space Plug-in (east-west traffic)
- Kubernetes Multus Plug-ins (separation of control/data)



Q How does a CoSP evaluate a supplier's claim of a "cloud-native" offering?

A Use this checklist to verify whether an offering is cloud native:

- Is it decoupled from the underlying infrastructure and platform services?
- Is it resilient to full failures and impairments?
- Can it support five-nines service quality, and is CNF availability configurable at deployment time?
- Can decomposed components be upgraded separately?
- Are existing components patched/upgraded or is the supplier always rolling out new ones?
- Does it support stateless processing and desired observability?

Q How does Intel support the transition to cloud-native deployments?

A Intel is working with the telecommunications industry to deliver software-defined networking and automation powered by Intel® hardware and software, through the Intel® Network Builders program. Developers can take advantage of Intel Containers Experience Kits, which help to address barriers to commercial containers-networking adoption and to deliver consumable capabilities concerning configuration, manageability, performance, service assurance, and resilience. The container bare metal experience kits are a set of collateral introducing newly developed open sourced consumable capabilities. The collection of all the container bare metal experience kits creates a library of best practice guidelines to address containers-bare-metal networking development and deployability gaps. It includes a variety of documents and demonstrations targeting developers, product leads, and a sales and marketing audience.

Intel has also prepared a "Cloud-Native Transition Playbook" that can help CoSPs take advantage of the synergy between 5G and cloud-native technologies. (This playbook is available only under a non-disclosure agreement [NDA]. Contact your Intel representative for more details.)

In addition, Intel is enabling standard, reusable, shared platforms for NFV that are easy to upgrade, maintain and scale. High-performance Intel hardware, such as 2nd generation Intel® Xeon® Scalable processors, Intel® Solid State Drives, Intel® Optane™ persistent memory and Intel® field programmable gate arrays (Intel® FPGAs) can contribute power and scalability to a CoSP's 5G efforts. And Intel is continually developing Intel® Select Solutions that can ease the path to network transformation.

This FAQ guide is a companion study guide with the white paper titled "Why use containers and cloud-native functions anyway?"; it is recommended to read both in parallel.

[1] https://www.5gamericas.org/wp-content/uploads/2019/12/5G-Americas_5G-and-the-Cloud.pdf

[2] Topology and Orchestration Specification for Cloud Applications (TOSCA) is an emerging modeling language whose mission is to be able to model every aspect of an application. It is an AT&T ECOMP Project

0520/FP/CAT/PDF 342107-001EN