White Paper

HardenStance

A Blueprint for a Cloud Native Telco

By Patrick Donegan, Principal Analyst, HardenStance

Sponsored by

BlueRocket FORTINET Netcracker An NEC Company NetNumber

February 2020

Executive Summary

- Telcos must commit to a Cloud Native operating model that adjusts dynamically and allows change to be implemented daily. The alternative is declining competitiveness.
- Telcos must not deviate from using the same Cloud Native principles and tools used by cloud providers to drive networking to the level of cloud compute and storage.
- Intensive training is needed in agile collaboration and Cloud Native tooling because Cloud Native can be operationally daunting as well as architecturally inspiring.
- Operational security must be enhanced by augmenting core tools. Access rights and key management must be added to harden Kubernetes' default security posture.

Beyond NFV to a Cloud Native Future

Emulating the global cloud providers to transform their companies from slow-moving, hardware and manual labour driven organizations into businesses that are software-driven, fast moving and highly automated has been a consistent goal of the telecom industry's leaders for at least the last ten years.

The energy and innovation in telco transformation is increasingly focused on the Cloud Native ecosystem.

What has changed in the last year or two is the sector's assumptions about what specific application and network architectures, organizational structures and processes, tools and partnerships are needed to gain the full benefits of digital transformation. Increasingly, telcos recognize that while the end goals of digital transformation are as valid as ever, they need a new means of achieving those ends. What that new means consists of is a commitment by telcos to converge on the exact same 'Cloud Native' principles and tools that the public cloud providers have built out and that continues to drive their phenomenal success.

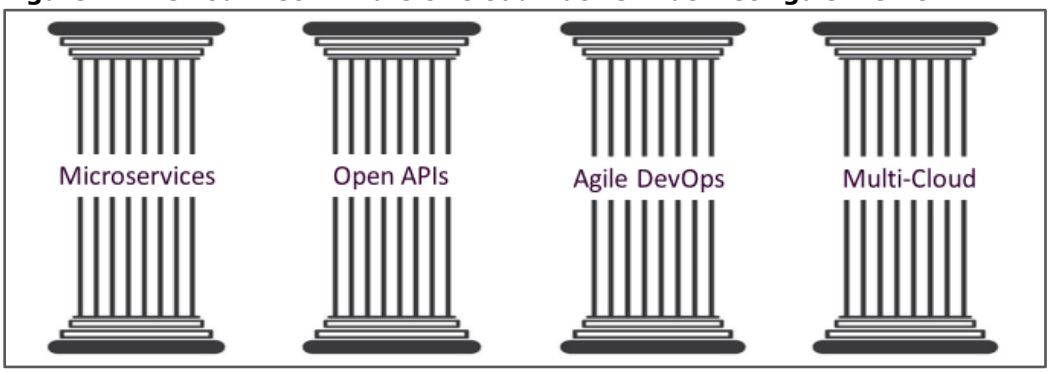
Some investments made in the telecom sector's own Network Functions Virtualization (NFV) ecosystem have generated marginal cost savings but none of the hoped-for gains in agility, automation or service velocity have materialized. Business customers have been able to buy cloud-based compute and storage in hours or minutes for many years – communications services still take days or weeks.

NFV is Far from Dead (as some have claimed)

Against the benchmark of strong ongoing orders for components of the NFV ecosystem, NFV is far from dead. Most telcos will continue sweating their existing NFV assets, potentially for many years. Most of them will continue investing in OpenStack for their own telco cloud. Many will also continue investing in other parts of the NFV ecosystem such as monolithic or vertically integrated Virtual Network Functions (VNFs).

However, as a cloud migration architecture for inspiring telcos to get to the next level in digital transformation, NFV has to be considered legacy now. The energy and innovation in telco transformation is increasingly focused on going Cloud Native.

Figure 1: The Four Tech Pillars of Cloud Native That Distinguish it from NFV



Source: HardenStance

As shown by the Deutsche Telekom case study on this page, telcos should start by introducing new pockets of Cloud Native deployments to augment the current NFV architecture. Over time, Cloud Native will become the de facto architecture.

At one level, Cloud Native principles and tools can be thought of as whatever the current best software development and operations practices of the public cloud providers are. At a more granular level, and as shown in **Figure 1** on the previous page, this White Paper defines four pillars of Cloud Native applications that differentiates them clearly from telco applications designed for the current NFV ecosystem. These four pillars are:

- They are broken down into a number of microservices, running in containers.
- They use open APIs.
- They embrace agile software development and a DevOps framework.
- They can run on any cloud, whether it be a private or public cloud.

Telcos must let go of traditional assumptions that they must always build all their own infrastructure themselves.

Most VNFs that are Shipping Today are Not Cloud Native

These differentiated characteristics of Cloud Native development are discussed in more detail further on. They are central to enabling telcos to dynamically manage changing priorities on a par with the cloud providers, which NFV is clearly not capable of delivering. For example, most VNFs shipping today don't comply with two or three of the above four pillars. Many don't comply with any of them.

Leading telcos are already investing in the Cloud Native model. As shown in the Deutsche Telekom case study below, first deployments are concentrated in the BSS/OSS domain. As it's part of a telco's IT estate, it's inherently more cloud-ready than a telco's network infrastructure. As discussed further on, this is reflected in the Cloud Native world where compute and storage tools are more advanced than those for networking. In adopting Cloud Native models, telcos must let go of traditional assumptions that they must always build all their own infrastructure themselves. There is accumulating evidence of leading

Deutsche Telekom Exploits Agile DevOps for Transport Automation

During 2019 Deutsche Telekom began undertaking an Agile DevOps-driven transformation of its operations environment in order to bring unification and automation to its transport networks.

The deployment is interesting because it leverages Agile DevOps methods for its Domain Orchestration platform layer at the OSS layer, even though the composition of the underlying transport layer itself comprises physical devices and SDN controllers. The project introduced collaborative Agile DevOps working methods with its supplier, a substantial restructuring of the way development and operations teams work together around building a Minimum Viable Product (MVP), and containerized software drops scheduled at four-week intervals.

As transport networks haven't featured prominently in standards development relating to telco transformation up until now, some data models and workflows had to be jointly built from scratch.

Deutsche Telekom has stated that the project has already delivered some substantial improvements in speed. Automated IP Trunk provisioning using the new Domain Orchestration solution is already live. Additional capabilities including unified network discovery, visualization and trunk provisioning across the IP and optical domains are also in the process of being rolled out.

Netcracker provides Deutsche Telekom with its Network Domain Orchestrator for this project. According to Netcracker, as well as using Agile DevOps development and deployment practices, the orchestration applications in use are bona fide Cloud Native. They are deployed as a set of re-usable, containerized microservices that run on any container platform, hence on any cloud.

telcos being willing to deploy services in public clouds. For example, AT&T is deploying its BSS/OSS in Azure. Verizon, Vodafone KDDI and SKT are all are partnering AWS to develop edge computing services.

The 5G SA Core uses a Microservices Architecture

It's the rollout of the first 5G networks that is drawing a line under NFV based on VNFs as an operating model and ushering in the Cloud Native era. Beginning later this year, leading telcos will roll out the first deployments of the 5G Stand Alone (5G SA) core. For many telcos, these deployments - especially at the edge - are the right insertion point. 3GPP's Service Based Architecture (SBA) for the 5G SA core is itself based on a microservices architecture and requires a Cloud Native platform.

While they must take account of unique aspects of their businesses, telcos shouldn't indulge those differences to the point of creating their own distinctive silo.

Core Principles of Cloud Native Strategy

For a Cloud Native strategy to succeed, a telco needs to be decisive about whether it really wants to break with its legacy business model or not. Reaching that conclusion requires being wholly convinced of the following facts of business life in the 2020s:

- failure to achieve more rapid process automation in development and operations will inevitably result in slow but steady decline in competitiveness.
- a Cloud Native approach is the only one that can enable that goal now.

Take Some Account of Telco Uniqueness – but Not Too Much

There are aspects of a telco's business that mean that the trajectory of its Cloud Native journey is necessarily different from most enterprises. For example:

- Telcos must automate the full stack including the network and operational/business layer. But as the first BSS/OSS deployments show, Cloud Native deployments need not cover the full stack all at once; the network can follow later.
- In most businesses, development and operations both tend to be run internally. Historically, vendors have undertaken most of a telco's development effort.

While they must take account of unique aspects of their business, telcos shouldn't indulge them to the point of creating a distinct silo. Telcos must not deviate from the huge scale and innovation of the broader Cloud Native tooling environment. If capabilities are lacking, they must participate in fixing and enhancing those same tools.

Strong Leadership with Targeted Messaging

Executing well requires buy-in at the board level and strong enforcement from the CEO down. The most effective way to execute is on a project-by-project basis, beginning with a first successful one that drives engagement in follow-up projects elsewhere in the organization. Organization-wide 'Big Bangs' typically have the effect of trying to 'boil the ocean' – they usually don't work.

Success requires a strong CEO to ensure the organization's resources are put at the disposal of an agile project as required. Otherwise a nascent island of change can easily fail at the hands of institutional inertia – or 'organizational anti-bodies'. Success also requires equally strong leadership on the part of the specific project's business leader.

Perhaps the trickiest part lies in the organization-wide communication strategy. Too little emphasis on what's required of the organization as a whole runs the risk of employees failing to understand what's required of them. On the other hand, too much of an organization-wide rallying cry risks people seeing career advantage in being seen to support a cutting edge initiative. That often achieves little more than unhelpful side-tracking of projects or people otherwise just getting in the way and introducing delay.

Extensive Investment in People and Training

Cloud Native projects require substantial investment in employee training. Some development teams might already be somewhat familiar with agile development and DevOps principles. Operations personnel are typically much less familiar with both.

Re-training in collaborative, multi-disciplinary, working models - including so-called 'Scrum Teams' – can be challenging for employees whose operating model hasn't changed much in decades. Open APIs and continuous improvement is the language of agile software. But agile also requires employees re-training in the human equivalents – openness to collaboration with other teams and continuous self-improvement.

New practices have to be learnt. Agile groups require formal collaborative processes that govern exactly how decisions should be made in these 'flatter' environments. Issues sometimes have to be navigated in teams combining telco and vendor employees. For example, some labour laws prevent a telco employee managing a vendor employee.

Most telcos start with very little agile expertise in-house. That gap needs to be filled with an optimal combination of working with vendors that have experience in the relevant domain, hiring experienced leaders and hiring independent consultants. Over time, customer references in the consulting aspects of delivering agile programs should become an increasingly important criteria in vendor selection.

Cost reduction shouldn't feature prominently in the goals of any one project. Cost savings are more likely to be generated longer term.

Goals and Expectations

The goals of Cloud Native development are revolutionary, but they can only be achieved with evolutionary means. Initial projects need to be carefully selected, focused on building a Minimum Viable Product (MVP) that delivers demonstrable results than can then be built upon elsewhere in the organization. There are many domains for which a Cloud Native transformation is hard to justify today. Some organizations are more ready for agile development than they are for DevOps. In other cases its vice-versa.

As elaborated on further below, the core goal of agile projects should be to enable services to be brought to market faster via iterative enhancements rather than releases delivered at intervals of several months. Cost savings shouldn't feature prominently in the goals of any one project. They are more likely to be generated longer term, once deployments are operating at scale, with speed and velocity, and a lot of re-use.

Agile projects also require a more agile approach to budgeting than most telcos are used to. Since agile is expressly designed to allow an organization to adapt to dynamically changing priorities, the traditional approach of fixing a budget for a specific project timeframe doesn't work well. All this has to be managed in the context of the traditionally conservative expectations of a telco's shareholders.

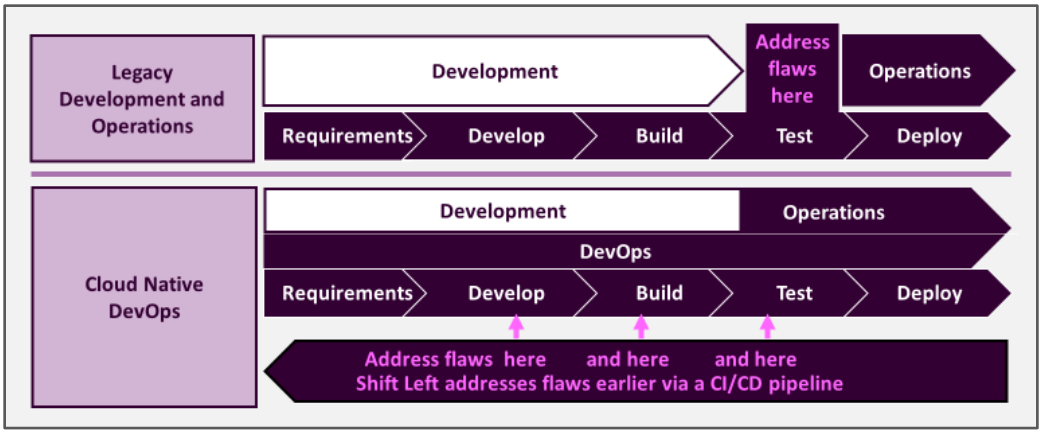
DevOps Working Practices and the Shift Left

As alluded to above, while development and operations teams each have challenges and opportunities that are unique to their own domains, one of the biggest challenges relates to the need for new collaborative working practices across the two domains. From a developer perspective, for example, they have to be able to see beyond the abstraction of containers and Kubernetes to the actual properties of the operations framework of the underlying cloud infrastructure that their applications will run on.

Central to that is the role for a unified DevOps philosophy, practices and common tools across the two environments to exploit the benefits of greater collaboration across the Software Development Life Cycle (SDLC).

Improving the quality of software earlier in the lifecycle in terms of its readiness for live deployment through Continuous Integration and Continuous Development (CICD) allows flaws to be identified earlier rather than towards the end, when they are so much costlier to fix in terms of time and money. Earlier quality controls are what should give both development and operations teams the confidence to allow higher levels of automation.

Figure 2: DevOps and A Target Architecture for Shifting Left



Source: HardenStance

A core principle that underpins this in DevOps is the so-called 'Shift Left' movement. Very few organizations actually address flaws in this exact way today, but **Figure 2** is indicative of the 'Shift Left' principle and target end state. There should only be one goal: working software that customers will buy as measured by limited defects, engineering value and on-time delivery.

Multiple integration, production-ready tests need to be written so that assumptions can be verified early on.

A key example of the type of cross functional collaboration that's needed within a DevOps framework is the requirement for intensive, automated testing and monitoring throughout the DevOps lifecycle. This becomes enormously important in a Cloud Native world. Here, we're not talking about simple unit tests that developers undertake to mimic an API's behaviour to see if its syntax is correct. Rather multiple integration, production-ready, tests need to be written so that assumptions can be verified early on.

Clearly, it's the development teams that have the skills to write these tests. However, since they are the ones that have to be able to trust that the test outcomes accurately predict outcomes in their real-world production environment, operations teams have to engage in helping development teams ensure that the tests are scoped correctly.

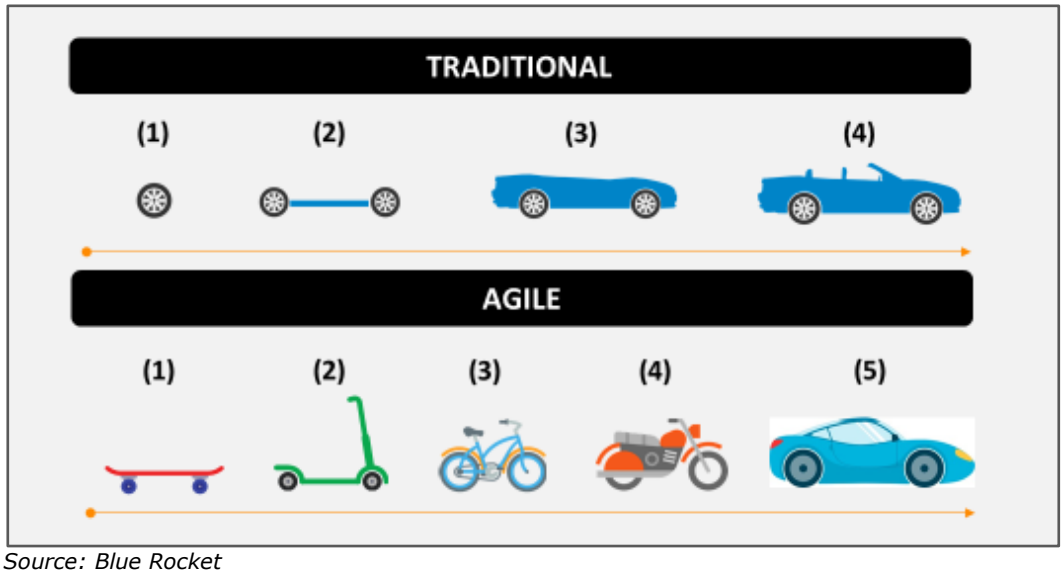
Telcos must not under-estimate what's required here. An operator's unique network environment, performance targets and risk appetite typically don't lend themselves to off-the-shelf test suites from vendors. Operators typically need to write these tests themselves. Because quality becomes everyone's problem, traditional demarcations between job functions do become blurred. An embedded, institutionalized, DevOps framework helps both the organization and individual employees adapt and succeed.

Agile Development the Cloud Native Way

As shown in **Figure 3** on the next page, agile development allows applications to be built in a completely different way – faster and iteratively, in a way that can adjust dynamically to changing customer requirements. It's essential to have the involvement of business leaders in these groups. They are first to know when business priorities change. Their engagement is critical to ensuring the dynamic alignment of projects with the needs of the business. Without strong engagement and guidance of business leaders, the focus of development effort can become decoupled from the actual needs of the business for weeks or months.

Agile project leaders must therefore assume that a high percentage of initial requirements – as much as half – will change in order to map to those new business needs. The quid pro quo is that through automation a project can move very much faster through the middle and latter stages towards live deployment than in traditional models and is also available for re-use.

Figure 3: The Difference Between Traditional and Agile Development



Source: Blue Rocket

Many telco's IT departments already have some agile operations underway but, as stated, progress in the telco domain itself has been painfully slow. In a traditional telco environment supporting a mix of Physical Network Functions (PNFs) and VNFs, legacy monolithic applications have high levels of dependency between all components across the full stack. This is reinforced by proprietary interfaces between components.

Cloud Native applications are built on a set of independent, self-contained microservices which are only loosely coupled via APIs.

Monolithic Apps are like a House of Cards

For that reason monolithic apps have the same characteristics as a house of cards. To maintain availability, you can only undertake comprehensive changes at fixed intervals separated by one or more quarters. In the meantime, because of those dependencies, an issue with any one component can bring the whole thing crashing down.

One of the key aspects of a Cloud Native agile development project is that Cloud Native applications are built on a set of independent, self-contained microservices which are only loosely coupled via APIs that are open rather than proprietary. For telcos, open APIs mean those designed by industry associations and standards bodies such as the MEF Forum, 3GPP, ETSI, TM Forum and the GSM Association (GSMA).

This model allows individual components to be patched or updated independently of one another without impacting other components. Frequently replacing rather than changing individual components so that an application is effectively re-deployed each time drives greater agility and robustness or what is often called an immutable infrastructure.

Unlike most VNFs that can only support a 1+1 resiliency model, Cloud Native applications support N+K, so they consume less compute resources. Two other aspects of Cloud Native microservices consist of how they handle state and how the control flow of a computation is expressed:

- A lot of microservices are stateless. Rather than being supported in the code of every microservice, state is stored and shared in common registries like MongoDB.
- Programming is also declarative rather than imperative. This means it allows automated self - programming to get to a specified end state using closed loop methods rather than scripting each of the steps required.

In a Cloud Native environment, containers are also the default way of packaging microservices. They offer much better efficiency and portability than VMs. For example, by means of a shared kernel, each container doesn't require its own OS image. Kubernetes has also become the de facto choice for orchestrating those containers on

Cloud Native infrastructure, not just among public cloud providers and other large enterprises but among telcos as well.

Best in Class Software Sourcing

As telcos are much more dependent on vendors for their development activity than most organizations, the core of a telco's agile development environment model requires the building of the right collaborative working practices and commercial partnerships with its vendor partners, while also exploiting the growing potential of open source software.

Telcos typically prefer vendor-supported versions of open source software. This assures an SLA that covers security risk, albeit at the expense of delayed access to some of the latest versions. For ultimate competitive advantage, and loosening of dependence on vendors, some may want to integrate the latest open source components, along with all the patching of different versions that requires.

Most vendors are having to undergo internal upheaval in agile transformation at the same time as their telco customers. The development model in which vendors develop software, deliver it and then make changes at intervals of several months is no longer fit for purpose. The model now has to be a much closer co-development model in which changes can be worked on jointly and at much more frequent intervals.

The development model now has to be a much closer co-development model in which changes can be worked on jointly and at much more frequent intervals.

Far more Frequent Software Drops

The goal is to get software into the network in much more frequent drops than today's environment of yearly or twice yearly releases. Leading vendors have already released what they have said is their final major release in favour of incremental releases Ultimately, functionality will be developed and delivered in real-time or near real-time. Vendor licensing models also need to adjust. For example, vendors need to loosen their dependence on perpetual licensing. Larger telcos should also consider whether writing some of their own software can confer competitive advantage.

Over the next year or two, distinguishing between leading vendors and followers or laggards that are merely "Cloud Native washing" their portfolio will be key. This requires a deep understanding of a vendor's microservices at the code level. A monolithic VNF that's packaged up in a container and deployed in a private telco cloud isn't Cloud Native. There is no easy, universal, metric for determining how many individual microservices a given application should be broken down into, but the following should be considered:

- The appropriate number of microservices an application should be broken down into is heavily dependent on the nature of each application but it's safe to say that in many cases, two or three is not adequate to derive the full benefits.
- Some network functions are harder to break down as microservices than others. For example, the design of some legacy applications assumes communications between static entities rather than a more dynamic, agile, environment.
- In some cases, there may be a risk of too many microservices generating application performance issues including increased latency.

Cloud Native Telco Operations

Cloud Native development takes it for granted that hardware is inherently available, and with the right performance characteristics. The ecosystem delivers that for compute and storage today but not yet for the network. It falls on telcos and their operations teams to put this right and manage colossal volumes of containers while also maintaining their core remit of 'keeping the lights on' at all times.

Implementing change here is the biggest challenge in becoming a Cloud Native telco. Re-training people from proprietary scripting to open modelling using languages like TOSCA and Yang isn't even the hardest part.

The deeper cultural challenges can be broken down into three components:

- Institutional memory teaches telco operations that change often makes bad things happen. Operations needs to be persuaded to view this reality differently: bad things happen because operations is tinkering with an outdated model that's no longer fit for purpose. Making lots of small changes that are all easy to verify is also less risky than accumulating large volumes of functional change and testing them all.
- Deferred gratification doesn't work in telco operations. It's not acceptable for performance metrics to improve long term if it's at the expense of them declining in the short term. Hence why many telcos initially prefer greenfield agile projects.
- Automation threatens job losses for those who are unable or unwilling to adapt.

Independent of psychological or sociological barriers, moving to an 'Infrastructure as Code' model which is API and model-driven rather than script and ticketing-driven is a major discontinuity. Embracing what Netflix's cloud pioneers celebrate as this 'chaos engineering' is a major challenge for telco operations . This section looks at key aspects of telco operations that change with the move to Cloud Native. It then addresses bringing Cloud Native networking tools up to the same level as storage and compute.

Embracing what Netflix's cloud pioneers celebrate as 'chaos engineering' is a major challenge for telco operations.

Intensive Monitoring as well as Testing

The case for intensive testing across the SDLC and how that has to span development and operations has already been made. New requirements for monitoring network traffic as a part of 'chaos engineering' are just as important. This is because there is very little point in development driving investment in containerized Cloud Native applications if operations doesn't have the Cloud Native tools it needs to monitor their behaviour.

Operations teams therefore need training in popular opensource management tools like Prometheus and Grafana, aligned with development timescales. Where telcos want to run containers in Virtual Machines (VMs), most available monitoring tools provide good visibility into the VM but little or none into the container. It's also worth considering that in the case of database management – which matters a lot to telcos – to date many organizations have chosen to manage their databases outside the Kubernetes environment rather than within it (in some cases because the Kubernetes database management features are considered too complex).

Many Vendors Prefer Cloud Native Tools to NFV Tools

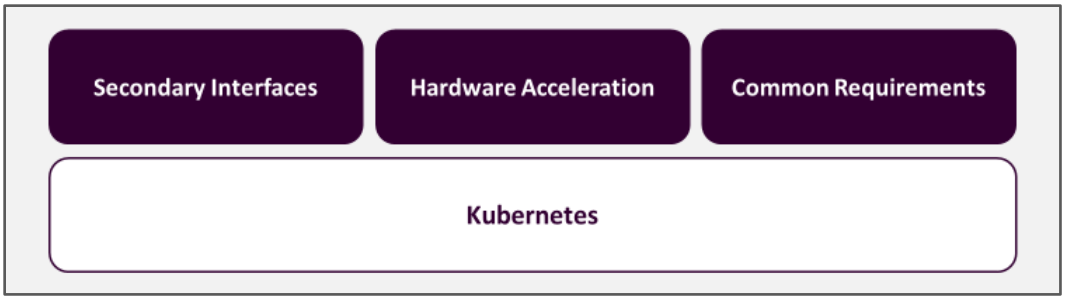
A lot of networking vendors are enthusiastic about the opportunity that migrating from NFV to Cloud Native represents from a tooling perspective. Specifically, vendors report that re-using the same automation tooling from one telco customer environment to the next is much easier in the Cloud Native environment.

For vendors, the story of recent years has often been one of investing too much in customizing automation tools for each of their telco customers. Sure, many telcos have prescribed standardized open source tools for their NFV environments, like Ansible for example, but often that's as far as standardization has gone. On top of that telcos have tended to layer on a substantial set of their own unique requirements to integrate Ansible into their OpenStack environment.

This has cost vendors time and money. The solutions they've developed are not very portable into other customer environments. Cloud Native tools look a lot more standardized. Vendors can be confident that an Ansible playbook developed for Kubernetes will be a lot more standardized, hence a lot more portable.

This paper assumes that Kubernetes is the de facto choice for orchestrating microservices running in containers. At this point there is no viable alternative. Telcos and their partners have to invest in fixing flaws and developing new capabilities within Kubernetes rather than recoiling at its present-day shortcomings or expecting cloud providers to develop the fixes for them.

Figure 4: Further Development is Needed in Cloud Native Networking Tools



Source: HardenStance

Some of the required fixes in networking capabilities are itemized below:

- Good progress has been made developing the secondary interfaces that some of a telco's Cloud Native network functions have to expose to a Kubernetes Pod. There's still work to be done here, though. The tools aren't there yet to support the visibility that telco operations needs into those secondary interfaces. Seeing into exactly what's going on from within Kubernetes is still more challenging than it needs to be - with regard to optimizing default routes, for example. Moreover, secondary interfaces don't seem to be very well supported by public cloud providers yet.
- For some networking applications, hardware acceleration capabilities like Field Programmable Arrays (FPGAs) smart Network Interface Cards (SmartNICs) and Graphics Processing Units (GPUs) may need to be exposed via the Kubernetes Custom Resource Definition (CRD) for more efficient use of hardware and better application scheduling. These are in the pipeline but they're not there yet.
- As vendors currently have their own description files for their network functions, once they are on-boarded they each have different expectations of the underlying infrastructure. The telco sector must get better at providing common Cloud Native requirements for vendors. The Common NFVI Telco Taskforce (CNTT), formed by the GSMA and the Linux Foundation in October 2019, is a positive step here.

New security patches can be deployed in production within hours of being released rather than wait days or weeks for the next patch window.

Cloud Native Security

Security in the cloud – whether it's your own private cloud or someone else's – has to be robust, flexible and automated enough to accelerate service velocity and agility without increasing risk. Sustained commitment to these three core principles can actually improve security relative to legacy models :

- Security policies and practices also have to 'shift left'
- Cloud Native security tools and practices must be embraced.
- A telco's security organization has to be fundamentally overhauled.

Security Policies and Practices Must 'Shift Left' Too

Today's development model carries with it the burden of exhaustive end-of-process security reviews that delay time to market or result in security policies being violated. We are a very long way from security requirements like threat modelling being accorded the same status as functional design and performance at the very outset of the design process. However, that's where the industry is ultimately headed for some applications.

In the meantime, doing security correctly, as an integral part of intermittent testing early in the CI/CD pipeline, reduces cost and time to market as well as organizational risk. This is one of the key things that can give telcos the confidence to automate manual processes. New security patches can be deployed in production within hours of being released rather than wait days or weeks for the next patch window. Today, many telcos still take up to a month to implement even critical patches.

The way CI/CD principles prohibit manual configurations or customizations also creates an intrusion detection framework that flags unauthorized changes as malicious.

Embracing Cloud Native Security Tools and Operating Practices

Telcos need to be able to master Cloud Native tooling and operating practices in order to exploit the potential of Cloud Native security. In development, granular security controls need to be built in at the level of individual microservices. All these controls need embedding into an application template so that DevOps can identify what needs to be built and what the workflow will look like.

The exposure of network ports follows a whitelist model – containers only expose the ports that developers explicitly ask them to. The minimal, declarative and predictable nature of containers helps threat detection platforms learn their behaviours and identify anomalies. The fact that they are intentionally immutable – they don't change once they're deployed - also means that a container that somehow gets modified serves as an in-built security alert.

Mitigating the highly insecure default security posture of Kubernetes is a mandatory requirement.

Out of the Box, Kubernetes is Highly Insecure

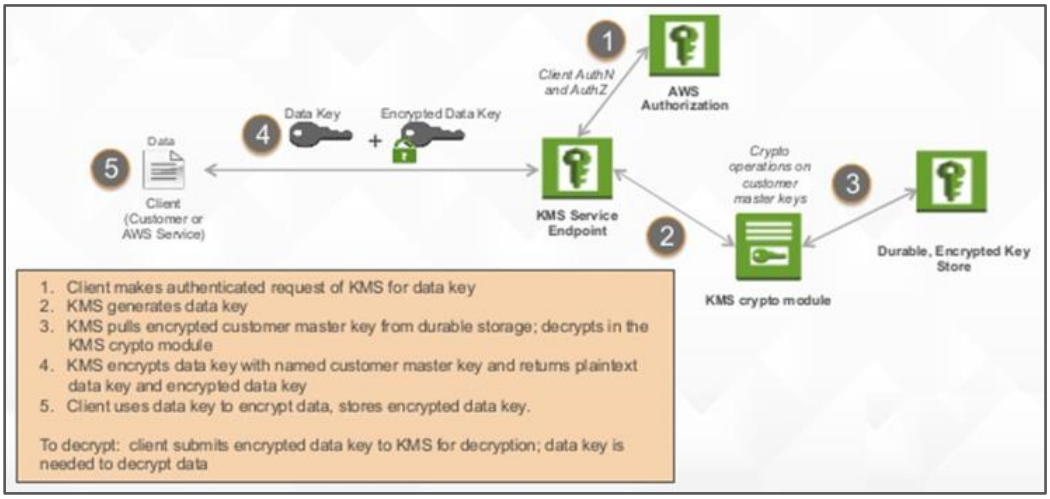
On the operations side, mitigating the highly insecure default security posture of Kubernetes is a mandatory requirement. In a Cloud Native model it's pieces of code themselves that need to be granted specific access rights to resources. It's no longer humans – Systems Administrators or Sysadmins - with keys loaded onto their laptops.

Out of the box Kubernetes allows containers to run as root. This means they are authorized to execute any command and access any resource they like. Unfortunately, this fact is obscured by some Kubernetes feature labelling. Specifically 'Secret', which is the name of Kubernetes' built-in object for secret management, doesn't even provide encryption. Kubernetes secrets are only base64 encoded.

Without any hardening, even entry-level level hackers can obtain API keys, gain full access to a Kubernetes environment in minutes, and potentially shut down the whole network. Hardening Kubernetes with the following two critical security functions is therefore a bare minimum requirement:

- a robust key management solution. This could be a standalone product or something like AWS Key Management Service (KMS) for AWS as shown in **Figure 5**.
- Role Based Access Control (RBAC) enabled with least access privilege to ensure that any given piece of code gets only the bare minimum access rights it needs.

Figure 5: How AWS Key Management Service (KMS) Works



Source: AWS

The above represents only the bare essentials of Kubernetes security for telcos. The complete stack must be secured. Every layer the operations team manages has to be secured. The security features used by each of the upper layers must be understood, verified, and managed by the security teams.

A Telco's Security Organization Must be Fundamentally Overhauled

It's relatively easy to make the theoretical case for a shift left in security. Implementing it in the context of organizational cultures and individual human beings is much harder.

Driving collaboration between development and operations teams is challenging enough. Adding members of the security team into a DevOps mix - according to a still more integrated DevSecOps model - is more challenging still. Developers tend to be motivated and rewarded according to how quickly they can innovate. Security people are motivated and rewarded according to how well they minimize exposure to risk. These are very different working cultures, that have traditionally been set in opposition to one another.

Security teams need to offer themselves up as peer-to-peer consultants and advisors, not gate-keepers as they are today.

No matter how challenging it is, greater integration of security into DevOps needs to be pushed through. Individual teams each need to be persuaded that shifting security to the left is in their own interests as well as their organization's. To look at each in turn:

- Security teams tend to be overwhelmed, understaffed and preoccupied with the latest threats. They aren't likely to see their headcount grow much so they need as much support on low-level security tasks as they can get from other colleagues. Greater automation should also be highly motivating for security professionals.
- Developers are used to time to market being held up by end-of-process security reviews. Certainly, most are not going to be motivated to spend a lot of time on security themselves. But if security teams can demonstrate how engagement with them intermittently, as well as at the outset, can help keep their projects on track, good development teams with good leaders are persuadable.

There are three aspects of evolving a security team's role that can provide some of the organizational and human 'glue' between security, development and operations teams.

- Security teams must be more willing and better able to teach security to other colleagues. They need to offer themselves up as peer-to-peer consultants and advisors or even services organizations, not gate-keepers as they are today.
- There's a case for limited delegation. Rather than expecting an entire development team to be up to speed on all security requirements, one or more designated security 'champions' within a development team can serve as key points of liaison.
- As they are in competition with adversaries, security professionals already change configurations and undertake emergency roll-out of new threat signatures on a daily basis. They're more used to agile-like models than most of their peers. This is a good starting point for engaging development and operations colleagues. ■

About the Sponsors

About Blue Rocket

Blue Rocket helps our clients unlock greater performance through strategic and operational transformation. We are experienced operators and strategists who "have been there and done that" as Fortune 500 executives, start-up founders and innovative technologists. Clients include Cisco, Google, iDirect, NetNumber, Salesforce, Zendesk, and Gitlab. We see and do consulting differently by focusing on decision-making and implementation, partnering with our clients through the entire process of transformation, guiding decision-making and activating strategies. Visit www.bluerocket.io

About Fortinet

Fortinet (NASDAQ: FTNT) secures the largest enterprise, service provider, and government organizations around the world. Fortinet empowers its customers with intelligent, seamless protection across the expanding attack surface and the power to take on ever-increasing performance requirements of the borderless network - today and into the future. Only the Fortinet Security Fabric architecture can deliver security features without compromise to address the most critical security challenges, whether in networked, application, cloud or mobile environments. Fortinet ranks #1 in the most security appliances shipped world-wide and more than 375,000 customers trust Fortinet to protect their businesses. Learn more at www.fortinet.com.

About Netcracker Technology

Netcracker Technology, a wholly owned subsidiary of NEC Corporation, offers mission-critical digital transformation solutions to service providers around the globe. Our comprehensive portfolio of Cloud Native BSS, OSS and Orchestration solutions and professional services, including Agile/DevOps practices, enables large-scale digital transformations, unlocking the opportunities of the cloud and the evolving mobile ecosystem. With an unbroken service delivery track record of more than 25 years, our unique combination of technology, people and expertise helps companies transform their networks and enable better digital experiences for their customers. For more information, visit www.netcracker.com.

About NetNumber

NetNumber, Inc. brings 20 years of experience delivering platforms that power global telecom and enterprise networks. Our software-based signaling-control solutions accelerate delivery of new services like Private LTE and IoT/M2M solutions across multi-gen networks, dramatically simplifying the core and reducing opex.

These solutions span a range of network types from 2G-3G-4G-5G to future G delivered on the industry's most robust signaling platform called TITAN. NetNumber Data Services are essential for global inter-carrier routing, roaming, voice and messaging. Data powers fraud detection and prevention solutions and enables enterprise B2B and B2C communications platforms. NetNumber multi-protocol signaling firewall, fraud-detection, and robocalling solutions help secure networks against current/emerging threats. For more information visit www.netnumber.com

About HardenStance

HardenStance provides trusted research, analysis and insight in IT and telecom security. HardenStance is a well-known voice in telecom and enterprise security, a leader in custom cyber security research, and a leading publisher of cyber security reports and White Papers. HardenStance is also a strong advocate of industry collaboration in cyber security. HardenStance openly supports the work of key industry associations, organizations and SDOs including NetSecOPEN, AMTSO, The Cyber Threat Alliance, The GSM Association, ETSI and TM Forum. To learn more visit www.hardenstance.com