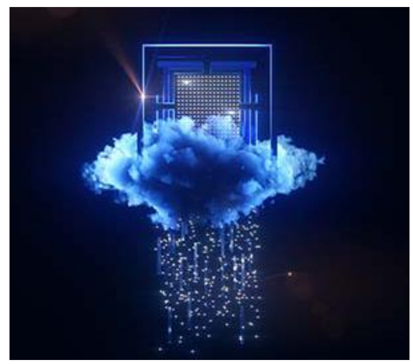
www.pipelinepub.com
Volume 18, Issue 3

The Cloud-Native Telecom Future

By: Reed Peterson



The telecom sector is in the midst of making some serious bets about the future. The advent of the pandemic forced many companies, telecom operators included, to speed up their digital transformation initiatives. According to TM Forum's Digital Transformation Tracker, the number of operators that said they were midway through their implementations jumped from 23 percent in 2020 to 38 percent in 2021, while 42 percent had started their moves. In total, 80 percent of companies are going through major changes today.

This work includes both external customer experience and internal operations—more than 65 percent of communication service providers (CSPs) are making changes here—while around 45 percent are implementing cloud-native technologies as part of these moves. According to Omdia research on BSS adoption trends, 34 percent of CSPs plan to refactor their BSS applications to be cloud-native in 2021.

The change to cloud-native

Moving to the cloud—whether public, private or a hybrid approach—involves a change in approach. Rather than simply moving existing technology to a different platform, migrating to cloud-native involves some changes in architecture. For many organizations, this change represents a potential source of risk. The potential benefits, however, far outweigh the exposure.

The term 'cloud-native' refers to technologies or processes that have been specifically developed with cloud in mind. Rather than taking a traditional approach that would previously focus on how to implement single machines or datacenter instances, cloud-native designs are built to scale up from the start. They are designed to run across multiple locations and services as their default, rather than being extended. Changing to a cloud-native approach can help CSPs to disaggregate their software, essentially moving to a more compartmentalized model that can scale up and down based on real demand levels.

This approach is commonly referred to as a microservices design. Rather than creating large applications that cover all functionality in one monolithic code base, microservices designs split each service into discrete components. These components connect to each other to produce the same results for the customer. Rather than requiring full-scale integration at huge cost, all traffic goes through APIs between components.

Putting applications into smaller units makes it easier to adopt new services or launch new applications for customers. A good example of this approach is Netflix: each service on the Netflix homepage is a different microservice covering functions like search or recommendations. If a new service is needed, it can be built in the background and then added once it is ready using APIs. Changes in existing software components are easier to handle; as long as the APIs provide the expected results back to the other service components, the actual software or application behind the API does not matter.

As demand for a service goes up—for example, more customers start using a service, or network traffic increases—then the application can add more resources that are needed automatically where they are needed. The result from this? More efficient systems and lower costs, while still delivering at the level of reliability that all CSPs need.

All this work points to a change in how CSPs think about their approach to serving customers, providing services and—most importantly—using data in their operations. These changes will affect multiple areas around how CSPs run their services from activation and provisioning through to inventory, authentication, and ongoing network performance. Each of these areas connects with each other to provide what customers expect, so any change can have a knock-on effect as well.

Migrating to cloud-native means looking at data

Deploying cloud-native applications or services does involve looking at the data that those applications create, too. Each service within a telecom company's infrastructure must be reliable and available, and traditionally that would have meant a fully redundant data center deployment. Moving to cloud-native should deliver reduced costs and more flexibility around how and where to deploy, but this cannot be at the expense of availability. Any deployment will have to deliver the same level of redundancy.In practice, this means that those applications—and the data they create—have to run across multiple locations, while still providing availability in the event that something fails. Any deployment will have to span those locations. For some CSPs, this will involve implementing their applications across multiple data centers that they own and run and make them available as a private cloud deployment. For others, this will involve using public cloud services with applications deployed across multiple availability zones. For other CSPs, this will be a mix of both private and public cloud. Whichever approach the CSP chooses to take, the data for this service still has to function as a single logical unit.

Like all applications, microservices designs create data that must be stored. This includes requests from customers and responses from the application, log data on activity, and the application data itself. Each microservice will create data for storing, and this data will have to be managed over time. The ideal approach for this is to use the same architecture as the applications themselves—so, as you move to cloud-native systems for your applications, you will also have to look at cloud-native data, too.

Cloud-native and open-source approaches

As part of the move to cloud-native, CSPs also have to consider their use of open source. Previously, open source would have made its way into telecom infrastructure to support specific functions. Operating systems like Linux would be used to run applications, but each implementation would be standalone and focused on low-value cost reduction. As OpenRAN gains in importance for building and supporting 5G networks, other open-source tools and platforms are gaining ground, too.

Cloud-native deployments rely on open source to function. Cloud computing services rely on open source for their functionality, while the container orchestration platform Kubernetes, originally developed at Google before it was contributed to the community, is also gaining in importance for CSPs. Companies have the option to deploy the same container images across any location, from edge deployments through to cloud infrastructure, and they will run in the same way.

Kubernetes manages the application infrastructure running and can respond based on conditions. For example, if an individual container stops working, Kubernetes can detect the issue and automatically restart an image. For CSPs, Kubernetes can manage this at scale, so availability is not an issue.

This also makes migration easier and supports the potential demand for edge computing applications over time. Services from AWS like Outposts and Google Cloud Anthos make it simple to deploy cloud services into telecom data center environments based on Kubernetes. With Kubernetes in place, the container images can run wherever the CSP needs them to.

While this takes care of the application side, however, it does not handle the data side. This must be handled using the same model, with data managed alongside the applications that create it. Software containers were originally designed as stateless instances that could be created and then destroyed when they were no longer needed. Data, on the other hand, has to be stateful, as it has to exist over time.

Running data in containers involves managing those stateful images over time. This issue has now been solved using Kubernetes Operators to manage those container images and persist them over time. These stateful containers can then be managed in the same way as the application containers.

This will involve looking at data across multiple potential locations, from edge computing deployments to centralized applications and from single data center locations through to hybrid and multi-cloud approaches. Whatever approach the CSP takes, they must be consistent in how they manage their data over time, and they will have to look at typical concerns like availability and reliability at the scale that they run at.

For CSPs, open-source projects will play a fundamental role in how they design and build their strategies for the future. As telecom companies move to cloud and look at scale, they also want to break up their applications and be independent of any one provider. As CSPs look to cloud-native to refactor their applications, and support their aims for a digital future, they will also increase their reliance on joined-up open-source strategies, particularly around data. Open source can help in this process, from the infrastructure that will host these new digital services and transformation projects through to the supporting elements like databases and data-streaming components.