

Creating Agile, Intent-Based Service Delivery Fabrics in CSPs' Distributed Clouds

Paul Parker-Johnson

Sponsored by

NOKIA

No Doubt about It, the Benefits of Cloud-Native Designs Are Compelling

The prominence of cloud-based services in our fully connected, software-driven economy is indisputable and growing. Expectations of easy to access, easy to use applications are increasingly built into our concepts of how we run our businesses and our daily lives. That said, the expansion of cloud-native use cases always brings with it a thoughtful consideration of how the new frameworks can be applied in environments they have not previously supported. How do we bring cloud-based operations into healthcare, energy production, supply chains, manufacturing, public safety, and telecommunications? Acknowledging the value of this analysis in new areas of adoption, we are still drawn by the simplicity, flexibility, and efficiency at scale that the cloud-native model provides.

Cloud-Native Is Now the Model of Choice for CSPs' Service Delivery Infrastructures

These dynamics are occurring in their own way in the market for technologies powering communications service providers' (CSPs) and related service providers' infrastructures¹. The cloud-native model has created value for CSPs supporting new revenue-generating services (such as per-subscriber entertainment bundles and enterprise offerings providing access to multiple cloud operators' infrastructures). It is guiding the development of new operations management software that scales and adapts to technological developments with greater agility than has been possible in the past. And it is being used as the architectural foundation of many new infrastructure designs that are powering the creation of new service offerings in multiple domains (including 5G, edge, industry 4.0 and residential broadband)².

¹ Service providers including inter-exchange providers (IXPs), application hosting and data center infrastructure providers, and cloud service providers of diverse types are **all** engaged in transformations of their own related to the adoption of cloud-native designs.

² See *System Architecture Milestone of 5G Phase 1 is Achieved*, https://www.3gpp.org/news-events/1930-sys_architecture, O-RAN Use Cases and Deployment Scenarios, <https://www.o-ran.org/resources>, and <https://www.cncf.io/blog/2021/07/05/kubeedge-cloud-native-edge-computing/> for examples of cloud-native design being adopted for use in primary deployment environments of direct interest to CSPs.

The model's signature attributes of openness, modularity, extensibility, and dramatically simplified operations (made possible by automation) are providing the framework for every layer of CSPs' service and operations designs moving forward.

These deployments will include elements working at several layers (compute, storage, networking, software infrastructure, and others). Each has a bearing on the outcomes. In this paper we focus on the most important attributes of the networking implementations that provide the scalable, performant, and software-driven communications that are the foundation on which the services will be delivered in the clouds.

What Do We Look for in the DNA of Intent-Based Data Center Fabrics?

The center line of logic to use in cloud-native data center networking is to view the network as an intrinsic part of the service delivery platform being used to provide the applications that users are consuming. This may sound contradictory to the concepts of layering and modularity that are also essential to delivering robust infrastructures in the cloud. It may also sound contradictory to the often-emphasized separation of concerns between networks and applications that has been a part of infrastructure design thinking for decades.

However, the idea is not to erase modularity and layering. It is to emphasize the need for a deep symbiosis between the multiple elements used in the implementation of a data center so they can work together efficiently and responsively at scale. This symbiosis is grounded in using a richly featured, self-optimizing, intent-driven set of capabilities that has not been possible to implement in prior generations of solutions that were based on less feature-rich, less flexible designs. By embracing these more agile designs, providers will be able to accelerate dramatically the delivery of compelling new applications.

Strands in the DNA of the Data Center Fabrics

To build service delivery clouds poised for success in the future, CSPs need to focus on building network infrastructures that provide the capabilities required in those cloud-native operations. We describe our perspective on the most important of those here.

The first point of focus needs to be on **seamless integration of the network with the cloud software platforms it is supporting**. These two entities need to work architecturally as one to ensure the accomplishment of service delivery goals. This is typically accomplished by management plane integration between the cloud software infrastructure and the network using APIs and data models aligned with the operation of the cloud software platforms. Without this automated, simplified level of integration, it is impossible for the network to be as responsive to the needs of the applications it is supporting as the cloud-savvy stakeholder community expects.

Figure 1 illustrates this symbiosis between types of cloud software infrastructure and the network fabric. Cloud management platforms (labeled CMPs in the diagram) include virtual machine-based solutions such as OpenStack-based distributions and VMware's ESXi based cloud software offerings, and container-based solutions based on Kubernetes orchestration from various suppliers. Continuous integration of the network fabric is shown being implemented via plug-in or listener software modules integrating the fabric's management software with each cloud software infrastructure.

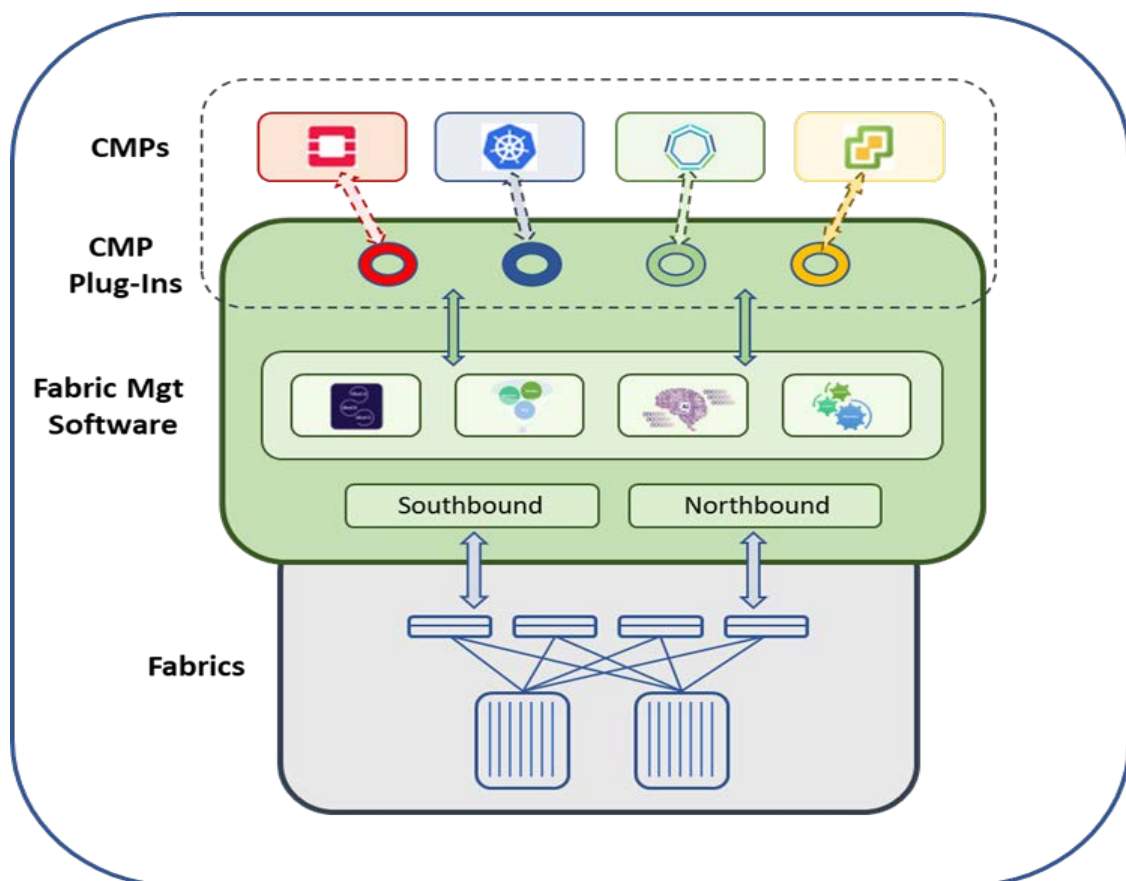


Figure 1. Continuous Integration of Cloud Management Software Infrastructures with the Software-Driven Networking Fabrics Supporting Them

Modules in the fabric management software (supporting a life cycle of operation from design and configuration to monitoring, analysis, optimization, and update) process the data exchanged with the cloud management platforms and apply it to the network it is supporting.

A second goal is to enable the network to function in alignment with **service delivery intent** at exceptionally large scale. This is partially made possible by the integrations previously described. Its biggest enabler is the software used to streamline provisioning and align the infrastructure with service delivery intent from the start. Intent-based provisioning allows managers to express, record, and translate the intended attributes of the desired service delivery into the operation of the network. While accomplishing this in a single site is meaningful, the reality is that CSPs' clouds will comprise a handful of large data center sites and 100s, if not 1,000s, of smaller sites as operations extend to the edge. The task is to align the intent of the services with the design of the networks and apply that to the operation of a collection of network domains and many, many distributed data center sites.

Figure 2 shows a conceptual framework for applying intent-based provisioning to a collection of data center fabrics and interconnecting network domains typical of this type of distributed cloud. At the upper end of the diagram and at the origin of expressing the intent that will guide the operation of the network is a software function conveying that intent to an infrastructure management module. For purposes of this diagram, we leave it undefined as to whether that is a single domain or a cross-domain function, as its purpose will ultimately be applied to multiple domains in one way or another. That intent is conveyed to the fabric management software running in each site using APIs and data models pertinent to those sites (and indicated by the translation, input, integration, and provisioning steps in the diagram). The process could be partitioned differently than is shown here to support accomplishing these goals to the range of site types we show, but the outcome would be the same. Communications in the model are shown flowing in both directions (to and from the sites and the related modules). This is to indicate that the flows have relevance throughout the life cycle of the operation, as feedback from operations will be used to tune the intent and its subsequent updates.

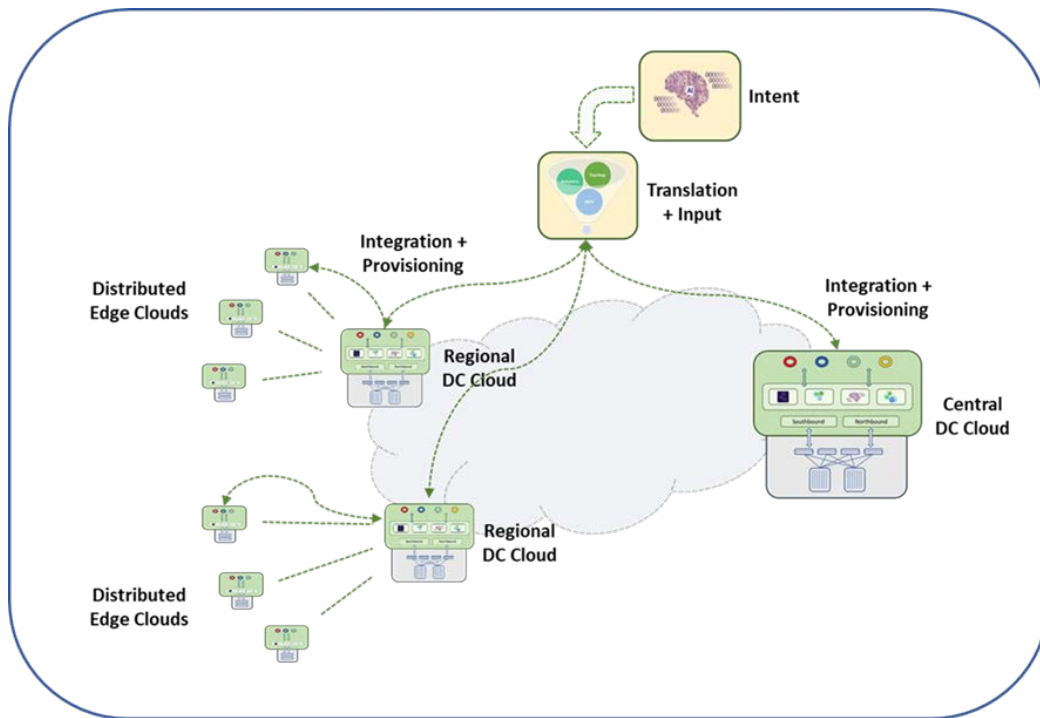


Figure 2 . Integration of Service Delivery Intent into the Fabrics of the Operator's Distributed Cloud
Are Unique to the Edge

A corollary to this level of distribution in the clouds is **support for a wide range of form factors** in deployments. Smaller sites at the edge require more compact, often differently configured elements than those designed for large numbers of nodes running in the core data center sites. To achieve this and remain efficient at large scale, solutions can use prevalidated templates embedded in their software to accelerate the deployment of sites using well-understood configurations ready to apply as appropriate.

To accomplish the foregoing requires a **commitment to solutions that are comprehensively software-driven**, designed for extensibility, automation, and adaptation to diverse requirements. In practical terms this demands leveraging cloud-native, micro-services design consistently, simplifying steps at as many points in the life cycle as possible.

Architecturally, making this work in distributed clouds means making good on a **commitment to being open at every layer**. While easy to say, this is more painstaking and challenging to accomplish than it appears. We identify four layers of functionality that need to collaborate in the realization of the goals: hardware abstraction, the network operating system (NOS) used in the network's nodes, network orchestration and automation, and robust analytics and artificial intelligence.

Hardware abstraction allows a common software implementation for the network's nodes to function consistently while running on a range of hardware designs. For example, it may be important for the software of a network to run consistently on platforms based on hardware from either Broadcom or Intel or another networking chip-set supplier. Having an abstraction layer built into the NOS makes this achievable much more efficiently than would be the case if a solution does not have such a layer. In the context of a network containing 1,000s of nodes and delivering a common suite of services across that infrastructure, this is a strategically meaningful characteristic.

Secondly the **NOS** itself is critical to implement in an open, modular way. As a first principle, being able to evolve components independently (such as protocol stacks or message passing and data storage functions) using a microservices-based design for each function is crucial. Beyond this, using such a framework allows a solution to incorporate functionality that a stakeholder such as a partner or a customer may want to add and to do so using the programmability of the solution. All of this is of great value operationally and in expanding solution functionality over time.

Figure 3 shows the conceptual framework of an open, modular NOS for use in cloud-native data center fabrics. Functions are implemented as individual, often container-based microservices throughout the NOS. Communications between modules in the NOS are based on open, model-driven message-passing protocols and APIs. Networking protocols are standards-based. And communications with external management and orchestration systems rely on model-driven APIs as defined by each application framework. Instances of the NOS run in each data center fabric node, whether leaf or spine.

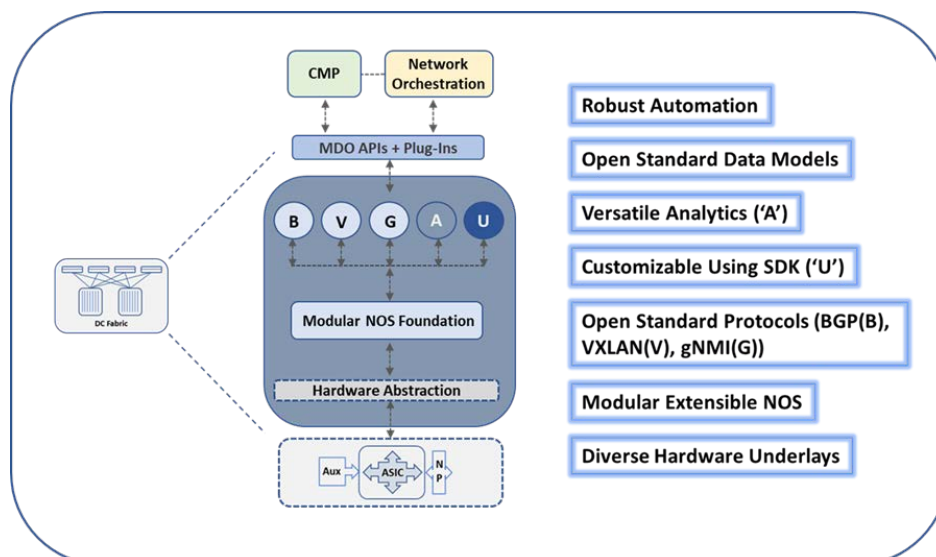


Figure 3. Conceptual Model of an Open, Modular Network Operating System for Use in Software-Driven Data Center Networking for Intent-Based, Cloud-Native Operations

A third layer of implementation that benefits from cloud-native design is **network orchestration and automation**. Benefits accrue at two levels in this layer: first, by reducing the number of tasks that need to be performed in the life-cycle management of a sophisticated deployment (such as the multiservice, multitenant, multilayered fabrics we are referencing); and second, by streamlining the accomplishment of those tasks **at very large scale** in both single large data center sites and in multiple highly distributed sites where staff are not always available for accomplishing all of the required tasks.

We call out both orchestration and automation in this layer because while they are closely related, they are also subtly different. Orchestration is largely focused on the arrangement and flow of tasks in life-cycle management: network design, testing, configuration, provisioning, upgrading, measuring, monitoring, expanding—rinse and repeat. By contrast automation is applied at literally every level of functionality in the solution in accomplishing these goals: simplifying design, accelerating testing, streamlining provisioning, and dramatically scaling the footprint of the infrastructure supportable by a given team of engineers.

All of these tasks benefit fundamentally from a deep and complete commitment to the use of cloud-native, microservices design at every step of the workflow. This modularity allows for functions to be enhanced at their own pace without hindering others. This helps in adding functionality to a module, as well as in helping it work at scale. Solutions built with this commitment to cloud-native designs in the full life cycle of operations stand a stronger chance of supporting the agility and versatility at scale that CSPs require.

The final layer benefiting from microservices-based designs is the set of modules providing **analytics and artificial intelligence** in a deployment. In some cases, architects place this into the same set of modules doing orchestration. We separate them to give them a distinct place in the flow of tasks involved in managing the life cycle of the network. Analytics and AI can be used throughout the life cycle of a deployment, from Day 0 (planning and design) to Day 1 (provisioning and activation) to Day 2+ (ongoing operation). They can be used in line, in real time to evaluate the health of the network and whether it is achieving the operator's intent. They can also be used as an embedded integrity check to evaluate the impact of a recommended change (whether the recommendation comes from a human or an AI based source).

They can be used as a component in planning and design to evaluate the viability of various alternatives. In all cases, having the modularity to run as a part of separate workflows with the agility to activate them whenever they are needed, to whatever extent they are needed (leveraging the elasticity of the cloud-native infrastructure) is a fundamental benefit to the operator, especially in a large, distributed cloud.

Figure 4 shows a conceptual framework for incorporating machine learning, analytics and AI into a cloud-native data center fabric. A suite of analytics and AI modules is shown at center right in the diagram. These are microservice-based applications composable in a variety of ways for analyzing and providing insights for the operation of the fabric:

- They are applied to performance analysis, fault diagnosis, capacity analysis, threat detection and mitigation, and many other tasks.
- They are fed by streaming telemetry data from the software-driven fabric of the network.
- They are linked to other applications in the management implementation supporting visibility to fabric operations, time-series reporting and analyses, alignment of the fabric with operator intent, and supporting a range of automated loop-closing designs from partial to fully closed loops.

This framework incorporates telemetry, machine learning, analytics and AI as integral, versatile and scalable components of an operator's intent-driven data center fabric.

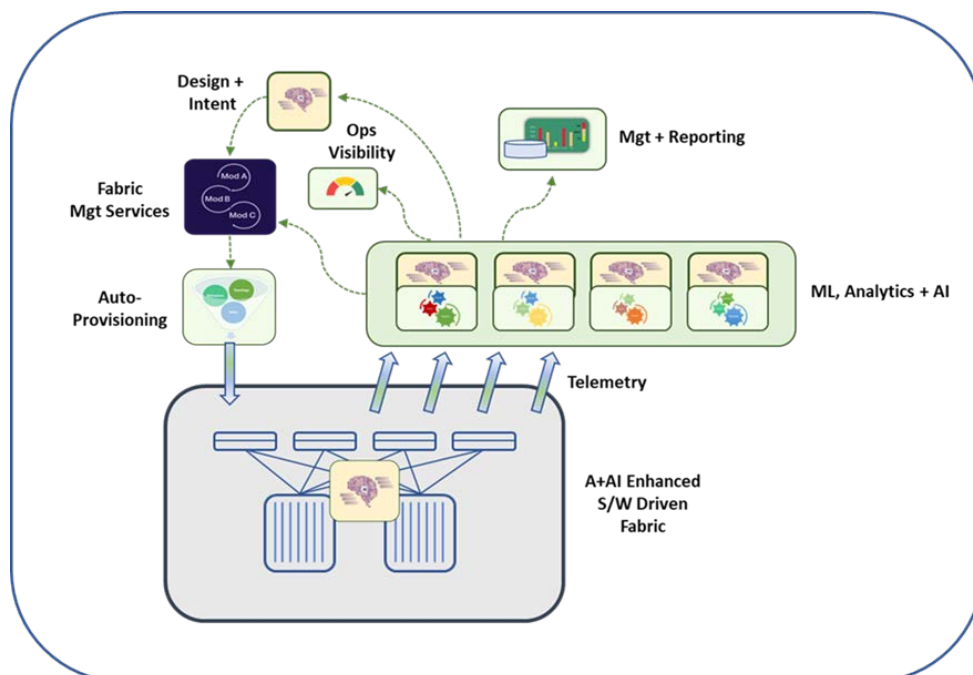


Figure 4. Machine Learning, Analytics and AI as Microservices Flexibly Designed into the Architecture of a Cloud-Native Data Center Fabric

Enabling such functions at the scale of a distributed cloud also depends on having an unwavering commitment to using open data models and protocols as building blocks in the design. These practices include architectural grounding in standardized and well-understood protocols such as OSPF, BGP, LLDP, EVPN, and VXLAN in the forwarding and control planes of the network; gNMI, gRPC, and REST in its management plane; and open data modeling frameworks such as YANG, JSON, and protobufs in the data exchanged among modules as a key to unlocking the agility.

With solutions incorporating these elements, CSPs will be able to deploy services that maximize their value and their flexibility in the footprints they need to support moving forward.

Nokia's Adaptive Cloud Networking: an Engine of Agility, Intent-Based Automation in CSPs' Distributed Clouds

Few companies have the depth of expertise and portfolio to create a truly agile, intent-based, highly automated, and scalable family of networking capabilities to support the service delivery objectives we have described. Nokia is among that elite set. Having applied itself in depth to deploying large-scale networks in many mission-critical service delivery environments over time, and with the guidance of its cloud and network operator customers, Nokia set out to create a data center networking solution designed from the start to meet customers' needs for agile, intent-based, scalable network infrastructures to meet their service delivery goals. Guided by this vision it created its Adaptive Cloud Networking solution.

Adaptive Cloud Networking creates an agile, intent-based, scalable infrastructure for delivering cloud-native network services from large central data center sites to smaller, widely distributed edge data center sites, as well as the WAN transport infrastructure in between in an end-to-end, comprehensively automated, intent-based network. Figure 5 illustrates this scope.

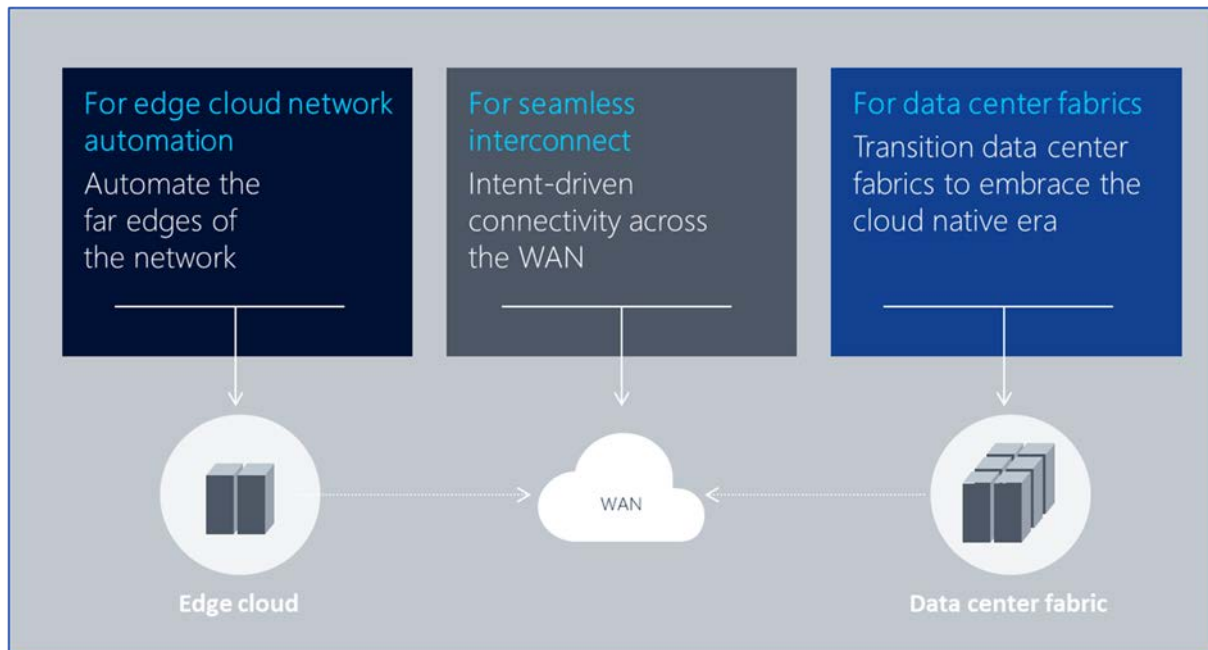


Figure 5. The Core to Edge Scope of Nokia's Adaptive Cloud Networking

In the sections that follow we highlight the most important capabilities of Adaptive Cloud Networking that illustrate its readiness to provide the cloud-native and intent-based operations we have described. We concentrate on the design and value of the Adaptive Cloud Networking's Data Center Fabric offering (on the right in Figure 5) that is focused on providing intent-based, agile operations in operators' larger scaled data center sites. In a second paper complementary to this paper, we focus on the innovations and value in the Edge Cloud Networking portion of Adaptive Cloud Networking, which delivers intent-based support for cloud-native operations in smaller scale edge data center sites (on the left in Figure 5)³.

Figure 6 shows the high-level composition of the Data Center Fabric used by Adaptive Cloud Networking. The whole solution includes the Fabric Services System, which is the engine of its automation, insight, and intent; the SR-Linux NOS; and Nokia's high-performance routing and switching hardware nodes. The principal capabilities that each of these layers possesses are shown within them in the diagram. We discuss the innovations and the value delivered by each in further sections.

³ *The Edge is a Cloud-Native Domain: Delivering High Value Services at the Edge with Network Infrastructures Built for the Cloud*, ACG Research, February 2022.

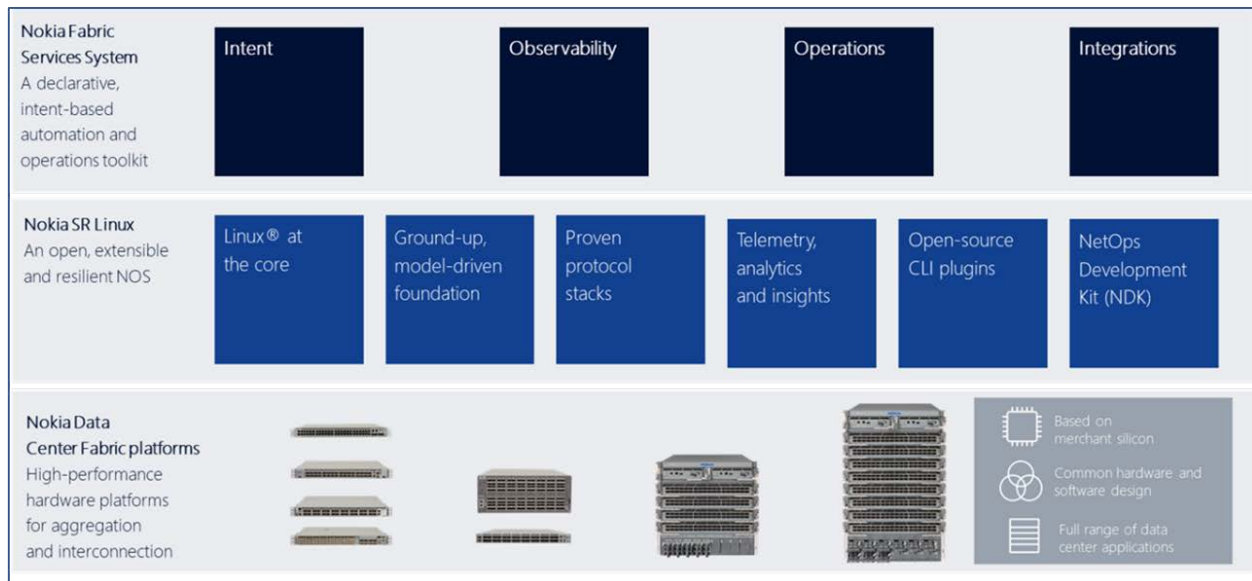


Figure 6. Nokia's Data Center Fabric Solution for Adaptive Cloud Networking

The elements of the data center fabric solution that are central in delivering its value are:

- Comprehensive use of **cloud-native designs**
- Robust functionality for **integrating intent** into deployments
- Software to integrate the network with **higher level cloud software platforms**
- Innovative use of **digital twins** in network design, analysis, and operation
- Robust functionality for **visibility and analytics** to support network operations
- Switching and routing nodes' software based on an **innovative, modular NOS**
- Use of **field hardened protocols** for high confidence in deployments at scale
- Use of a **hardware abstraction layer** in the node software's design
- Use of **open, model-driven protocols** in network management and control
- Integration with **higher level orchestration systems** for ease in life cycle management
- Deployable in a wide **range of configurations, site sizes, and form factors**

Nokia's Data Center Fabric solution is built using microservices-based, cloud-native software at every layer. One benefit of this is simplicity of integration into cloud-native environments using tools and models already in use (such as Kubernetes). Another is broad-based agility established on the ease of introducing enhancements into software, using CI/CD pipelines and DevOps, adapted to CSPs' infrastructures using a NetOps version of those techniques. Figure 7 illustrates the solution's approach to using collections of container-based microservices in one of its major components, its Fabric Services System.

It is easy to see the flexibility and precision that can be achieved by updating software for one operation's function (such as observability) at one point in time, while updating another function (say, integration with a higher-level software platform) at a different time. This benefit accrues to functions at many layers of implementation in the solution, as each collection of microservices-based modules is itself built on a microservices foundation, allowing for layers of agility to be at work in the solution simultaneously and over time.

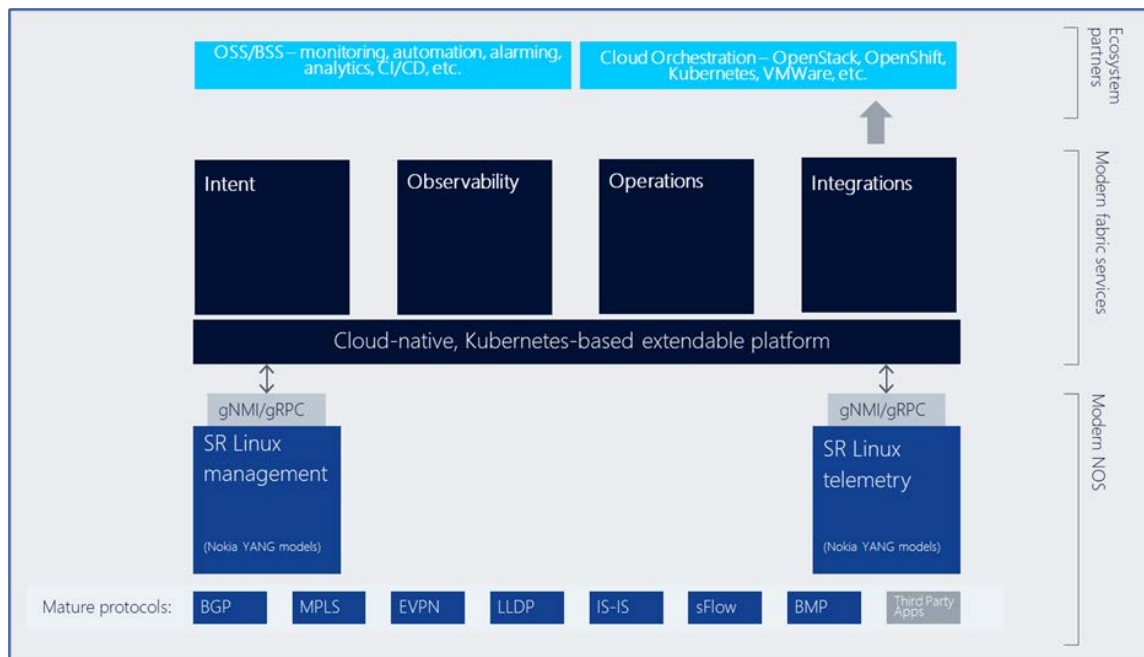


Figure 7. Use of Cloud-Native Design at the Heart of the Adaptive Cloud Networking Implementation

A second area of advanced functionality in the solution is the use of **intuitive applications and a set of related microservices to incorporate the intent of operators' service delivery plans into the network.** This capability supports distinct layers of deployment in the manner that customer teams typically encounter them in managing their deployments.

To illustrate, consider the intent of a network deployment at two different but inherently related levels: first at the level of an application and second at the level of the underlying network. With an application or a workload (for example, a 5G core network function), the Fabric Services System presents a dialogue to the user that captures key dimensions of how this workload will be deployed in the network. (The system can also obtain inputs for this purpose via a dialogue it maintains with a northbound cloud management platform such as OpenStack or Kubernetes, but for simplicity we focus just on the dialogue with a human user.)

Examples of dimensions the Fabric Services System collects are, what is the footprint the workload will employ (for example, will it run in one or in multiple data center racks or will it run in multiple data center sites)? Are there access controls for the workloads that need to be applied (for example, block and permit filters to prevent unauthorized access to the workload)? After these attributes are provided to the Fabric Services System, it interprets them and provides the necessary parameters to the networking elements that will be supporting the workloads, matching the service delivery intent to the deployed configuration. This is illustrated in Figure 8.

A similar function is provided to accelerate the deployment of the physical network fabric (or network underlay) in one or more data center sites. Capturing **fabric** intent allows designers to express, for example, how much resiliency or over-subscription they want at different points in their network or how they would like to accomplish data center interconnections (using a border leaf switch or a data center edge router, for example). In each case Nokia has incorporated the option of using validated design templates (prevalidated configuration profiles) in the deployment, saving time in deploying, configuring, and activating the nodes.

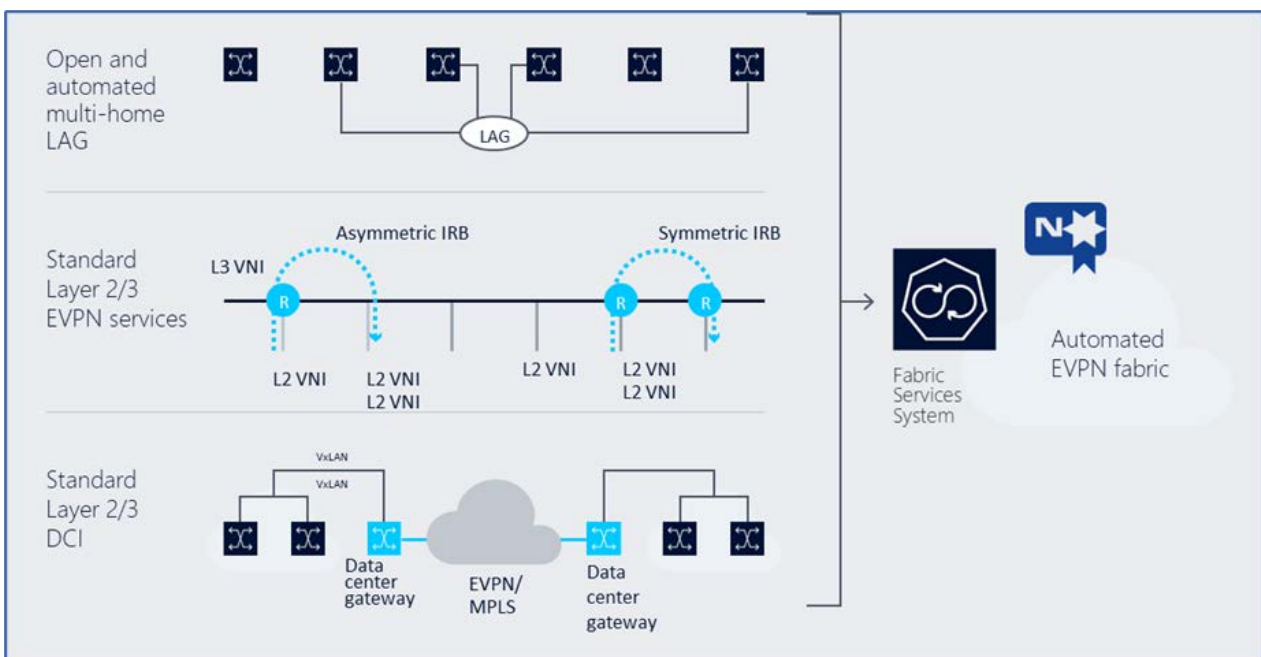


Figure 8. Variety of Deployment Designs (Leveraging Prevalidated Blueprints, Represented by the Blue + White Starred Ribbon) Based on an Operator's Intent for the Fabric Being Configured

Intents that have been deployed into the configuration are usable **throughout the life cycle** of the deployment to optimize the operation of the network. The intelligence that is embedded and the functionality that is supplied are robust. For example, if a deviation from intent is detected in the network, an alert is provided to the operator, and one of two actions can be taken. First, **the network can be adjusted** to ensure it is functioning in line with the intent; or **the intent can be adjusted** to align with the network's behavior if the operator decides that is a better path. When working at the scale of a distributed 5G and edge applications cloud, performing these steps in such an intuitive, streamlined manner is of great benefit to the team doing the work.

A third area in which the Data Center Fabric delivers a compelling level of agility is through a **layer of software integrations** that continuously align the network with the cloud management software that provides virtualized compute and storage that the operator's workloads require. These include virtual machine software from VMware and a range of OpenStack suppliers, as well as container-based solutions from suppliers using Kubernetes as their container orchestration engine. In these integrations, the Data Center Fabric solution proactively incorporates the topology and key attributes of the workloads into which it is integrating and listens to the management plane of the cloud management system to identify changes that affect network operations. It identifies changes that require an update to the configuration of the network and implements them automatically. This tight coordination between the operations of the cloud workloads and the network that is supporting them saves operations teams significant amounts of time assuring clients that this alignment is present in their deployments.

Expanding further on the versatility of Fabric Services System, Nokia has added a Digital Sandbox, **a robust network digital twin**, to use as a tool for evaluating design and configuration changes in a software-based test kit before the changes are made.

The sandbox is composed of containerized, software-based instances of fabric nodes configured as they are in a given deployment. The sandbox configuration is kept in alignment with the fabrics with which it is twinned by ingesting information on an ongoing basis from the fabrics. This information includes the state of the active

network (BGP agencies, FIB, RIB, port operational status, etc.), which further enriches the value the Sandbox can offer. Using sandbox functionality, operators can evaluate the merits of new designs to decide if they like the changes that would result, or not, saving time and money in both testing and deployment.

The role of the Digital Sandbox at multiple stages in the life cycle of a deployment is shown in Figure 9.

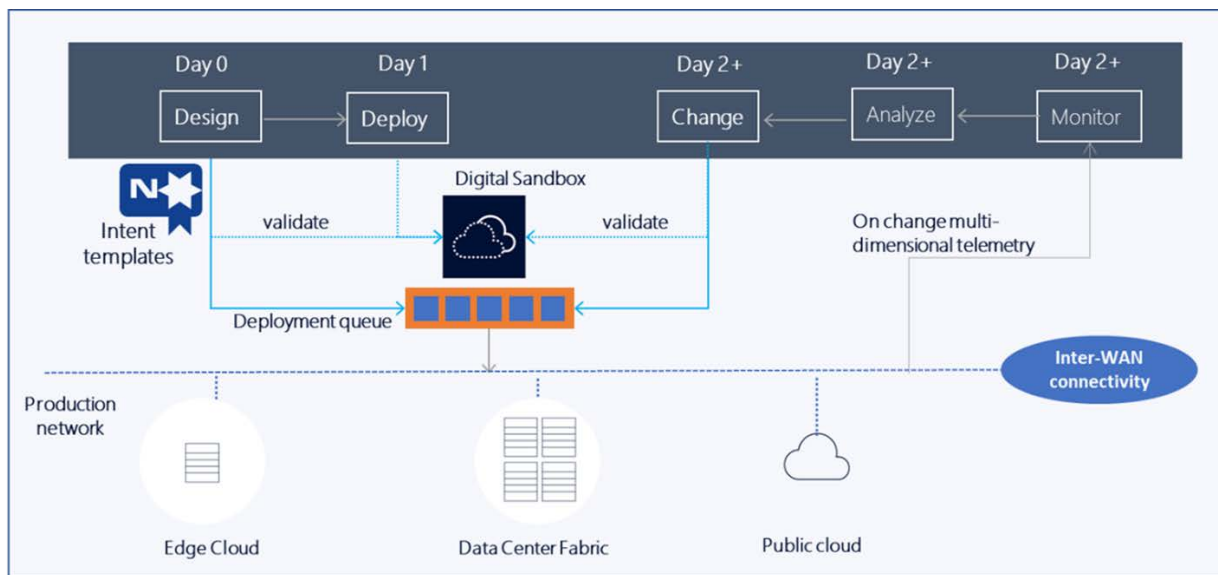


Figure 9. Digital Twin as a Value Add at Multiple Stages in the Life Cycle of a Data Center Fabric Deployment

A fourth area in which the Fabric Services System provides efficiencies is in its **acquisition layer**, which is purpose-built to ingest high volumes of streaming telemetry at scale. The telemetry is delivered to it by SR Linux through a robust gNMI collector service. Information on data plane and control plane operation and degree of alignment with the intent of the organization is provided in this manner. Cloud-native monitoring and alert-generation software such as Prometheus⁴, which is well-suited to deployment in Kubernetes environments like Fabric Services System, can be used. Visualization software such as Grafana⁵ can be used to display trends and observations that help operations and analysis teams increase their effectiveness. Results of analyses can be fed into the change management functionality of the Fabric Services System, including validations using the digital sandbox, to make improvements to the designs and configurations.

⁴ <https://prometheus.io/docs/introduction/overview/>

⁵ <https://grafana.com/>

We turn our attention now from the Fabric Services System to the underlying physical networking nodes in the Data Center Fabric that also distinguish the solution and make it ready to meet the challenges of cloud-native service delivery in distributed operator clouds.

If the Fabric Services System provides the cognitive capabilities required to align the networking infrastructure with operators' intents, then the nodes constituting the fabrics provide the mechanisms required to carry out the intents in a wide range of topologies and footprints that comprise the distributed cloud.

The heart of the operation of each node in the fabric is Nokia's **modular, Linux-based network operating system, Service Router Linux (SR Linux)**. SR Linux is an imaginative, forward-looking adaptation of the field-proven Service Router Operating System (SR OS) used in other Nokia routing platforms to meet the requirements of adaptive cloud networking at scale. The modular structure of SR-Linux is shown in Figure 10.

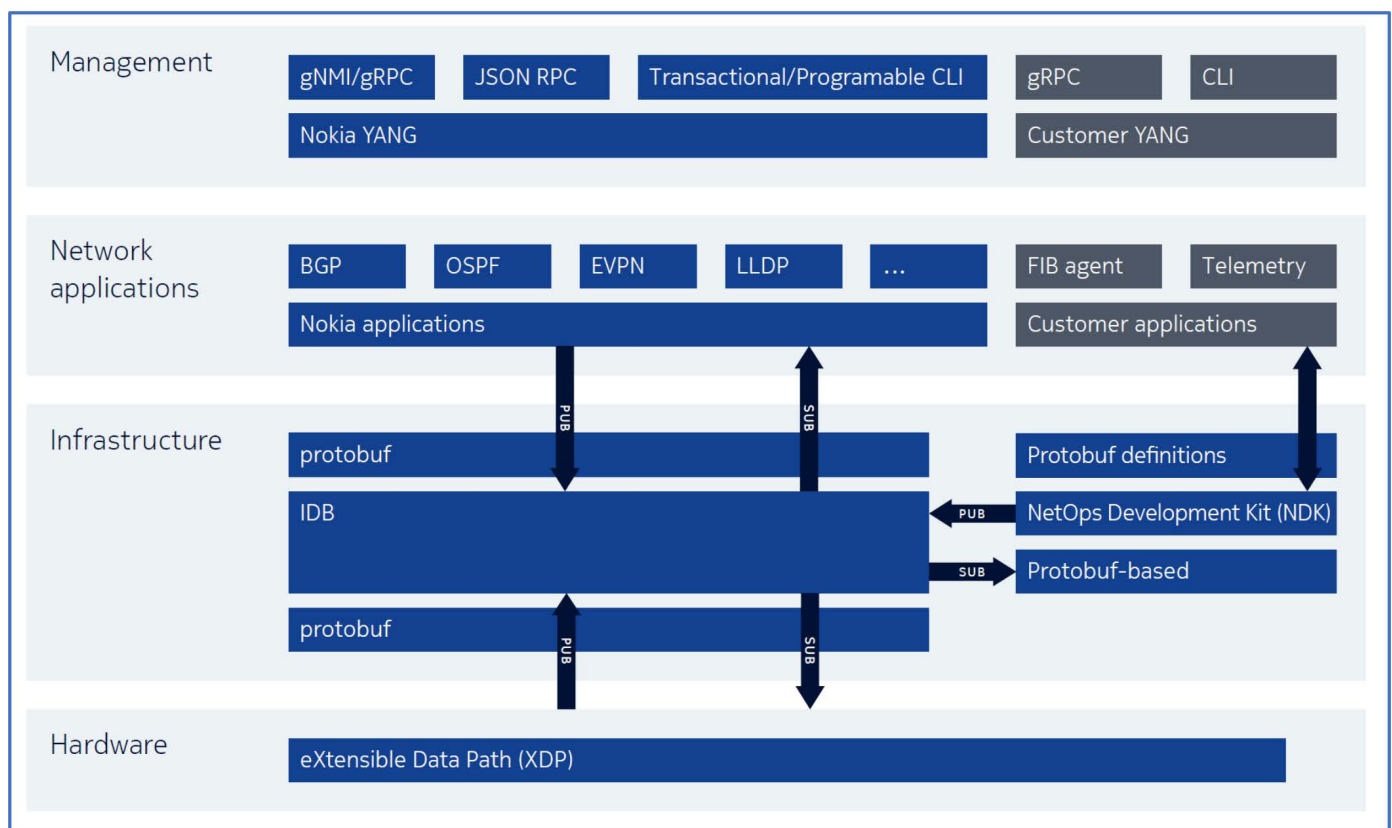


Figure 10. Logical Structure of Nokia's SR-Linux Network Operating System

Its modularity is apparent in the diagram. Its effectiveness in cloud-native operations is achieved by each module's implementation as a set of **container-based microservices** using the same framework as is employed in other parts of the solution. Another cloud-native attribute in SR Linux is its use of publish-subscribe (pub-sub) mechanisms between modules in each node, underscoring its flexibility for adapting to the needs of a range of scenarios, necessary in distributed clouds. Additionally, SR Linux's use of protobuf-based data transfer ⁶ between modules enables its nodes' efficient participation in continuously connected, data-driven, self-optimizing network operations.

Additional efficiencies are created by use of **open, model-based communications** between network nodes and higher-level applications based on gRPC, gNMI, and YANG data modeling. Communicating southbound to SR Linux nodes, these protocols and models make interaction with diverse higher level management systems possible. Heading northbound from SRL nodes to higher-level applications (such as Fabric Services System data collection modules), their use of gRPC makes them active partners in realizing the intent-based, agile service delivery model to which operators aspire.

An additional dimension of SR Linux that delivers high value in operators' highly distributed clouds is its use of field-hardened protocol modules originally developed for Nokia's SR OS routing platforms and tuned for use in Adaptive Cloud Networking. Nokia's data center fabric nodes run proven implementations of BGP, EVPN, OSPF, VXLAN, LLDP, and other protocols in large, distributed clouds.

A final element of SR Linux's design to highlight is its support of a hardware abstraction layer in the NOS. Having this abstraction layer allows SR Linux to support a range of hardware chip-sets as required over time. It provides a consistent set of instructions to network operating system software that needs to interface consistently with underlying hardware and allows each layer to evolve independently. When considering the range of form factors and levels of intelligence needed in hardware to solve different processing problems at the scale of an operator's distributed cloud, it is easy to see the value of having the NOS support consistent operations across a diverse combination of footprints.

Three other aspects of versatility made possible for operators by the SR Linux implementation are important to highlight.

First, expanding on the use of gNMI, gRPC, and YANG for communicating between the Data Center Fabric's brain (Fabric Services System) and a wide range of networking elements in a deployment, the

⁶ For a description of protocol buffers, or protobufs, see https://en.wikipedia.org/wiki/Protocol_Buffers.

use of those protocols positions the Data Center Fabric to evolve with customers' needs by supporting not only Nokia elements, but also elements supplied by other vendors that support those protocols as deployments evolve. Whether this is done to support gradual evolution of infrastructures between different types of platforms over time or to include diverse capabilities in a customer's deployment in a consistently managed mode, the value of providing such a capability is very high. Supporting this flexibility to include multiple vendors' nodes in a Data Center Fabric deployment and in the reverse direction to include Nokia's nodes in a deployment managed by another management implementation (such as a cloud provider's in-house management system) allows networks to function as a more consumable resource for operators. The easier it is to incorporate critical functionality in the service delivery platform, the better it is for operators to accommodate multiple requirements and to meet their customers' needs.

Additional flexibility is available using SR Linux via its Network Development Kit (NDK). Use of the NDK for integrating customers' special functions is shown on the right-hand side of Figure 6. The NDK adds value for operators that want to distinguish their services by providing refinements to the operation of the network specific to their scenarios. Flexibility in maintaining the applications over time is enabled by the NDK leveraging other standardized elements of the SR Linux environment, including protobufs, gNMI, gRPC, YANG based modeling, and Linux-based application development procedures.

A final dimension of the agility at scale of Adaptive Cloud Networking comes from its integration with higher level network and service orchestration systems to simplify integration into multidomain network operations. This is enabled by open APIs and data modeling as we have described. In Nokia's own portfolio, this makes for ready incorporation of Data Center Fabrics into end-to-end wide area network transport leveraging IP/MPLS and Segment Routing, using Nokia's Network Services Platform. It also allows the data center and edge solutions of Adaptive Cloud Networking to be integrated into other end-to-end network orchestration systems, whether developed in-house by operators or supplied by other orchestration vendors. Having this flexibility also supports integration into network slicing as solutions supporting slicing in different use cases emerge. And it provides simplified integration with domains running downstream from the network edge in either CSPs' or end customers' networks. Examples are specialized broadband access networks in service providers' mobile (5G) and residential (DSL, PON and cable) services, as well as enterprise campus and private wireless systems growing in use for industry 4.0, IoT, and digital transformation.

On each of the dimensions we have analyzed (and on each strand of the DNA of cloud-native infrastructure we have described) we see Adaptive Cloud Networking as flexible, scalable, and robust, well-built for the agility and scale required in CSPs' distributed clouds.

Looking at the path forward, if the future calls for cloud-native applications, using intent-based, software-driven fabrics in CSPs' distributed sites will be a central part of the architecture for delivering them successfully.

Experience has shown us that the open, software-driven, extensively automated deployments of the cloud create compelling levels of efficiency and deliver services that are easy to consume and expand. Using cloud-native designs in a similar manner will be critical to the success of CSPs as they build out more highly distributed infrastructures and weave themselves more tightly into their customers' applications and experiences.

Creating cloud-native service delivery infrastructures will require CSPs to use solutions built with greater openness, automation, intelligence, and versatility of configuration at scale than has been possible using platforms delivered in the past. From understanding the intent of operators' service delivery plans, to simplifying the use of functionality that realizes that intent in hundreds or thousands of sites, the cloud-native fabrics that CSPs' data centers run will need to be designed from the start to succeed in the flexible, pervasively automated, continuously self-optimizing manner that cloud-native applications require.

In this paper we articulated the capabilities networking fabrics supporting cloud-native applications need to possess to help CSPs accomplish their goals. We have shared our perspective on Nokia's Adaptive Cloud Networking solution, designed from its inception to meet those goals. With its innovations for incorporating service delivery intent into deployments, for leveraging network digital twins in evaluating alternative designs, and continuously, dynamically understanding the needs of the cloud-native workloads it is supporting, Adaptive Cloud Networking is a forward-looking solution fine-tuned to meeting the requirements of operators' distributed services. At every level of its architecture, it is well-suited to addressing the diverse requirements of the cloud-native applications that operators are supporting, and to evolve flexibly in meeting future requirements. In building out their versatile, software-driven, intent-based infrastructures for the cloud, CSPs will be rewarded by looking into how Adaptive Cloud Networking could address their application delivery goals, using multiple cloud deployment frameworks, running from top to bottom in each operational stack, and working from core to edge in their service delivery footprints.

About the Author



Paul Parker-Johnson

Paul Parker-Johnson is ACG's research lead in cloud and virtual system infrastructures and their use in private, hybrid, edge and multi-cloud designs. His work encompasses physical, virtual and cloud-native infrastructures; use of intent-based modeling and automation; fusion of OT and IT in distributed cloud designs; and support of use cases in 5G, industry 4.0, and digital transformation. pj@acgcc.com

ACG Research delivers information and communication technology market share/forecast reports, consulting services, and business case analysis services. Copyright © 2022 ACG Research. The copyright in this publication or the material on this website (including without limitation the text, computer code, artwork, photographs, images, music, audio material, video material and audio-visual material on this website) is owned by ACG Research. All Rights Reserved.