NOKIA

# A software-defined networking approach to SBC

White paper

## About the NFV Insight Series

The application of virtualization and cloud principles, such as network functions virtualization (NFV) and software defined networks (SDN), represent a major shift for the communications and networking industry. Until recently, this approach appeared to be unworkable because of stringent performance, availability, reliability, and security requirements of communication networks. Leading communications service providers (CSPs) now implement Network optimized Cloud architectures for SBC and SDN to gain competitive advantage through increased automation and responsiveness, as well as by delivering an enhanced customer experience, while reducing operational costs. This series of briefs and white papers addresses some of the key technical and business challenges faced by service providers as they move Session Border Controllers (SBC) functions to the cloud.

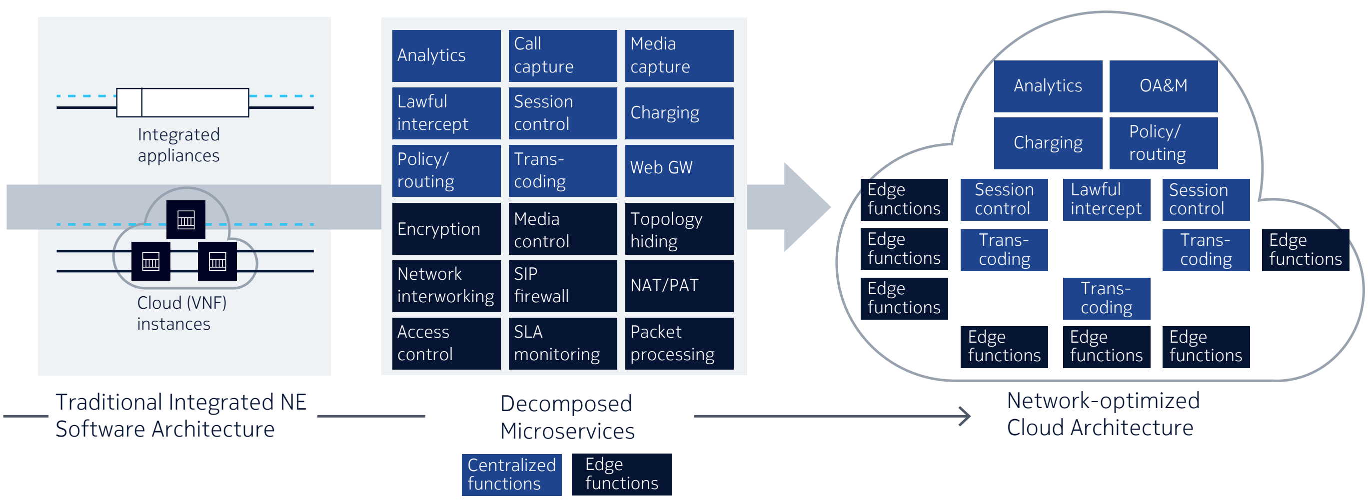
Contents

# Introduction

The growing importance of the session border controller (SBC) as an SIP networks guardian, as well as its unique and strategic position at the network edge, is also being challenged by advances in cloud computing and networking automation.

As Software-defined Networking (SDN) takes hold in CSP networks, the SBC will need to be optimized at the network level. Moreover, it must also become simpler for operators to distribute and manage in conjunction with other network service functions (SFs). What's more, the SBC's decomposed services must be capable of being deployed in the optimal place, while still maintaining a centrally managed view. This will offer the operator both visibility and ease in managing operations.[1]

Figure 1. Evolving SBC towards a Network-Optimized Cloud architecture




SDN has evolved to address the modern data center's requirement for a much faster and agile computing and storage infrastructure due to the emergence of edge computing. To a great extent, the traditional approach to networking has reached its limits. As a result, it needs to catch up with the rapid innovation typical of today's infrastructure ecosystems and data-hungry technologies. These technologies include OpenStack, OpenDaylight, Docker, Kubernetes, Internet of Things (IoT), machine learning and artificial intelligence (AI).

Advances in security have also made it possible to diagnose problems at all levels of the stack. Accessibility and visibility into all layers of SDN has been challenging with the more traditional and proprietary networking systems. However, that has now changed. Operational teams can now quickly correlate abnormalities with monitoring tools and access more networking and infrastructure data in an unprecedented way. Indeed, the next phase will see the data being used with advanced machine learning algorithms in order to detect, identify, and resolve security attacks in real time.

1 See the Nokia white paper: 'Network-optimized cloud Architectures for session border controller (SBC),' for insights into how SBC functions can be evolved towards a fully distributed, microservices oriented, network-optimized cloud architecture.

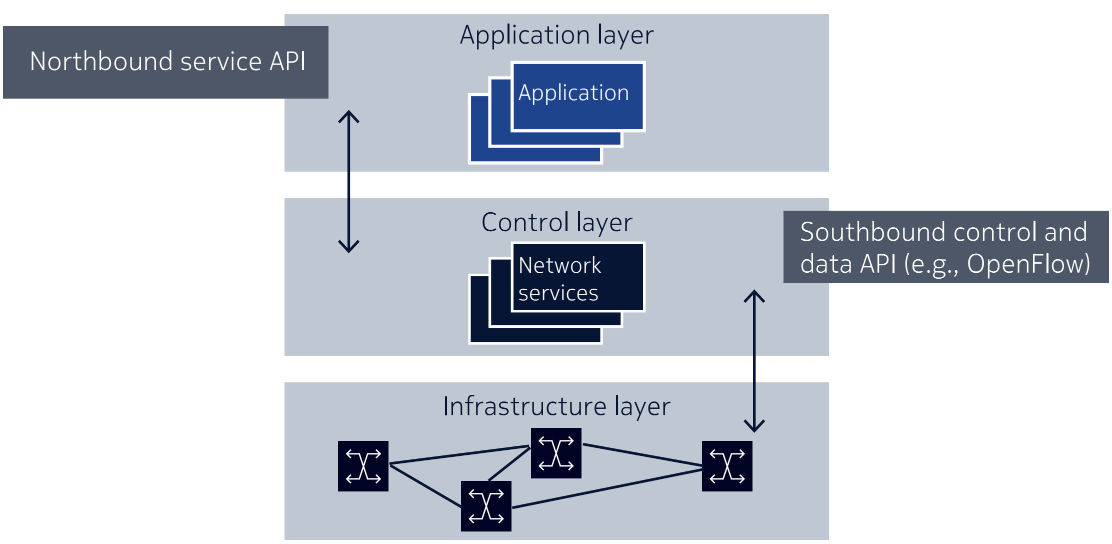
SDN encompasses a variety of networking technologies — both old and new. These are designed to facilitate a software-driven approach to network services programmability. This paper reviews some of these technologies together with prevalent architectural principles. It also examines how these principles can be leveraged in a renewed approach to building an SBC optimized for networking of the future.

# SDN concepts

In today's world of computing and storage, the traditional model of static networks is no longer sufficient. It cannot meet dynamic and growing user needs.  Using open interfaces, SDN offers an architectural style to decompose the network. At the same time, SDN enables a more dynamic and programmatic approach to initializing, controlling, and changing network management. In all cases, the separation of the data plane, (also known as the media plane), from the control plane, (also known as the signaling plane), is key to unleashing the flexibility needed in a software-defined network.

With the separation of the control and data planes, innovative approaches to network services building become more possible. Applications can leverage a programmable infrastructure. Among other things, the separation allows developers to get much closer to the packet routing layer without the requirement for specialized hardware skillsets. As computing and storage infrastructure becomes commoditized, so does the SDN infrastructure layer. Indeed, industry trends indicate that forwarding and routing networking is being implemented on COTS hardware, otherwise known as white boxes. These boxes have a small footprint and low-energy components for edge networking needs. One advantage of taking this approach is that switching can be easily customized to the specific needs of a deployed application.

Figure 2. SDN architecture




- In Figure 2, the SDN architecture control layer uses OpenFlow,[2] or a similar southbound API. This enables the programming of the white boxes to meet the connectivity needs of a given application. The network applications use the northbound API, provided by the SDN controller, to meet their individual business logic needs. The use of this architecture delivers some key benefits, including service agility and service innovation with the separation of concerns.

2 https://www.opennetworking.org/sdn-resources/openflow

- A much simpler networking design and common troubleshooting approach.
- A reduction in hardware costs by decoupling the functionality, which leads to less duplication.

To drive down costs, in addition to the operator's reliance on proprietary closed-door systems, future networks will be heavily influenced by ongoing innovation within open-source software groups. As the industry matures, SDNs will drive the revolution of hardware ecosystems to support next-generation data center networking optimization, including cryptography, transport offloads, and SmartNICs.

# Service chaining concepts

As described in IETF RFC 7665, Service Function Chaining (SFC) Architecture, "the delivery of end-to-end services often requires various service functions.  These include traditional network service functions such as firewalls and traditional IP Network Address Translators (NATs), as well as application-specific functions. The definition and instantiation of an ordered set of service functions and subsequent "steering" of traffic through them is termed Service Function Chaining (SFC)."[3]

To enable an elastic and agile deployment of today's services, a new mechanism for influencing service chaining is needed. This agility is achieved by the Network Service Header (NSH), which is described as: ... [defining] a new protocol and associated encapsulation for the creation of dynamic service chains, operating at the service plane."[4]  The NSH provides the packet encapsulation and mechanism for the exchange of context metadata along the established paths of the service chain.

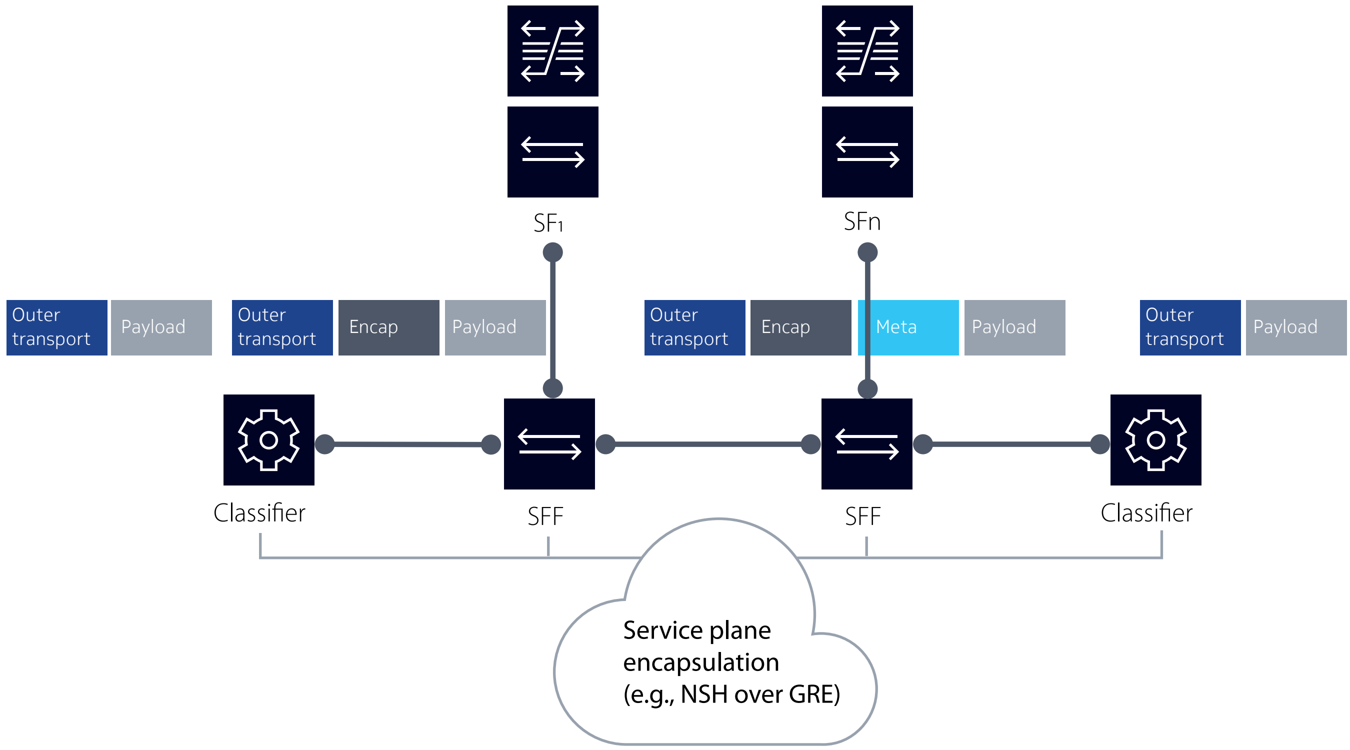
The service chaining approach is illustrated in Figure 3. The role of the classifier is to determine what traffic, based on policy, needs to be chained.  Once a packet has been accepted into a service chain, the SFT details are encapsulated, forwarded, and delivered by the service function forwarder (SFF) to the next SFF in the service function path (SFP). The SFP is the sequence of routing constraints, which govern the path that packets must take, which are assigned to a certain SFP.  As the packet traverses the SFP, additional metadata and context can be added and/or modified by SFs and classifiers in the service function chain. The encapsulated information is transport independent and used by SFC-aware functions. For packets egressing the SFC administrative domain, the encapsulation and metadata information are removed.[5]

3 Service Function Chaining (SFC) Architecture, IETF RFC 7665 - https://datatracker.ietf.org/doc/rfc7665/

4 Service Function Chaining (SFC) Architecture, IETF RFC 7665 - https://datatracker.ietf.org/doc/rfc7665/

5 For more information, consult: Service Function Chaining (SFC) Architecture, IETF RFC 7665 - https://datatracker.ietf.org/doc/rfc7665/ and Network Service Header (NSH), IETF RFC Draft - https://tools.ietf.org/html/draft-ietf-sfc-nsh-21

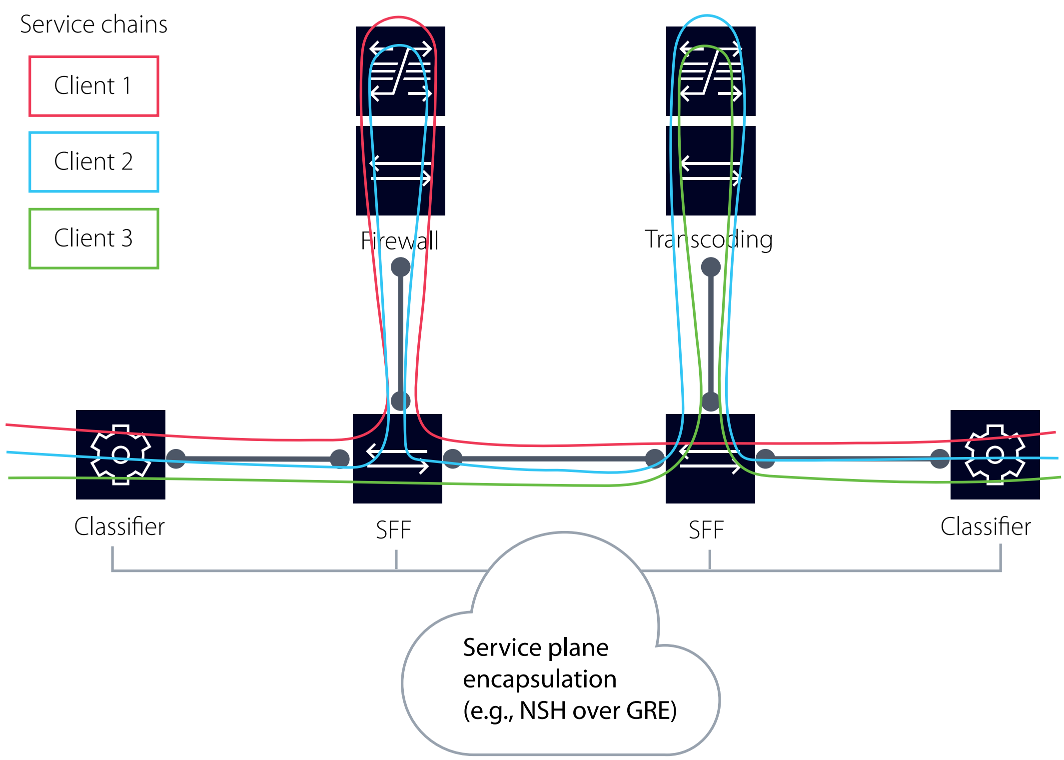
Figure 3. Service function chaining




The concept of service chaining has been leveraged ever since networking infrastructures have allowed the proliferation of services and applications to communicate across distances. For the most part, these services tend to be fixed in how they receive network packets. However, the practice of using rigid service chains lacks the flexibility and the speed required to influence traffic flows at a granular level. With SDN now taking hold, the priority has shifted to ensuring that the network can meet the needs of applications, as well as directly influence changes as these relate to security, policies, and computing conditions.

Figure 4 provides an example in which flexible service chaining enables selective classification of packets through an analysis of traffic flows provided by the Virtual Network Function (VNF). Based on the continuous analysis of these traffic flows, as well as the networking conditions, the VNF, in real time, can influence policies put in place and modify service chains in order to maintain the required quality of service (QoS).

Figure 4. Service chaining




# SDN-Optimized SBC

As a typical network element, the SBC is not immune to offering network features that may already be part of the underlying network but are not in the path of the service chain. Just as SDN was born to create a programmatic and dynamic networking approach, SBCs should be on a similar path. What's needed is a renewed perspective on decomposing SBCs' core functions from those features that could eventually be replaced by a service function in a network capable of service function chaining.

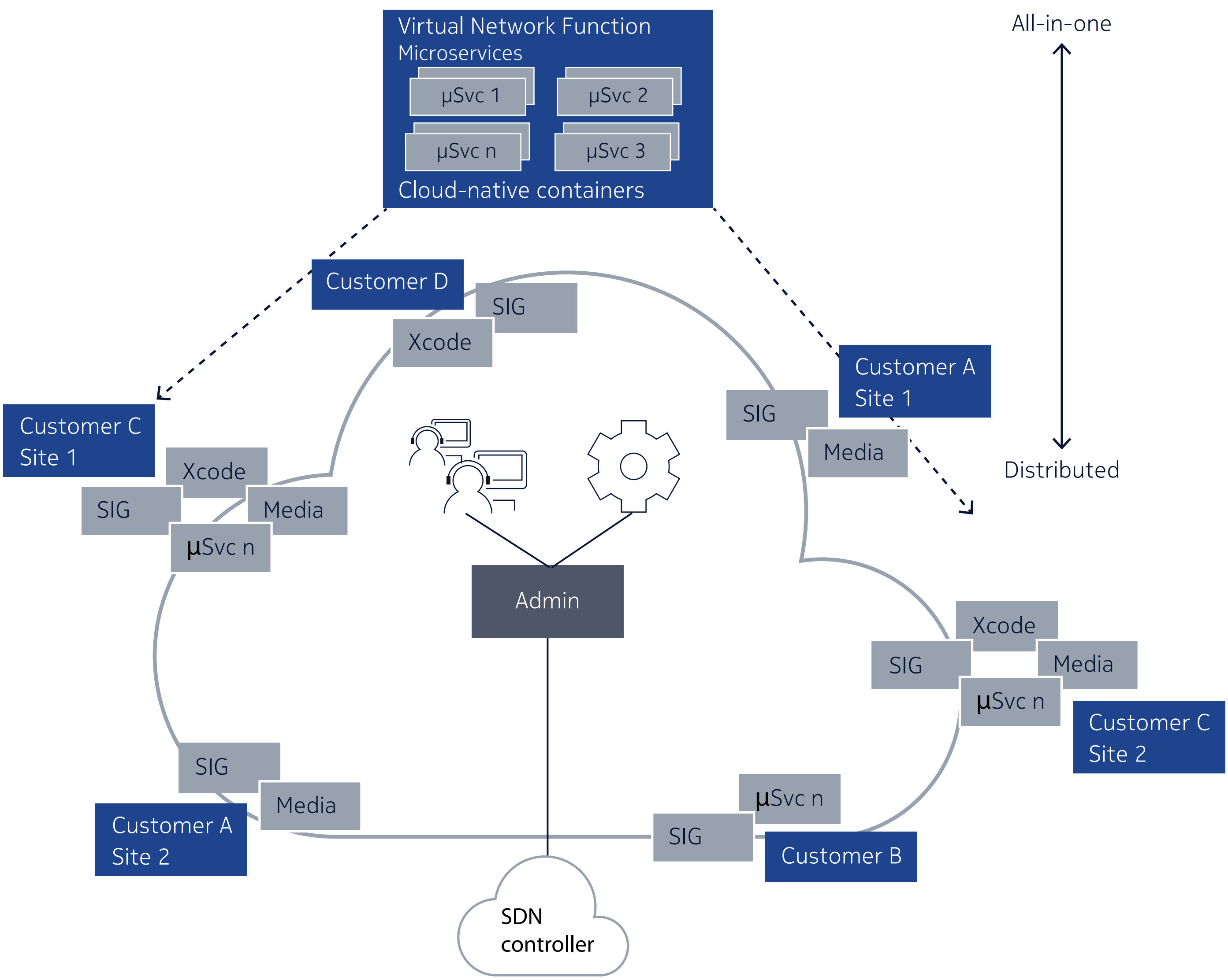
If an SBC VNF can be fully distributed across a network — that is, spanning the cloud data center, the regions, and the edge — SDN can play a pivotal role in providing ubiquitous computing across these segments. In this way, SBC functions can be composed as "service functions as-a-service" (SFaaS) that are independently managed and controlled as a cloud native microservice.

Connecting SFaaS into a service chain enables the composition of multiple value-added services that provide revenue generating services to end users.[6] This model places the choice and flexibility of service offerings, the composition of service chains, as well as service placement across the SDN completely under the control of the CSP or the customers' CSP. As the network changes, these service chains can be established on demand, re-distributed, and automatically scaled to meet the growth and demand of end users.

6 One approach to building SFaaS components is covered in more detail in our Microservice Anatomy Insight article. https://insight.nokia.com/microservice-anatomy

Figure 5 illustrates a cloud-native SBC VNF, which is decomposed into independent services. These services are centrally managed and optimized from a network-wide perspective. With a network-optimized SBC, the distributed services connectivity can be centrally monitored and managed to maintain the required QoS. It can also provide policy enforcement at the network edge for signaling streams, as well as media data flowing across different IP-communications services.

Figure 5. SDN-optimized SBC



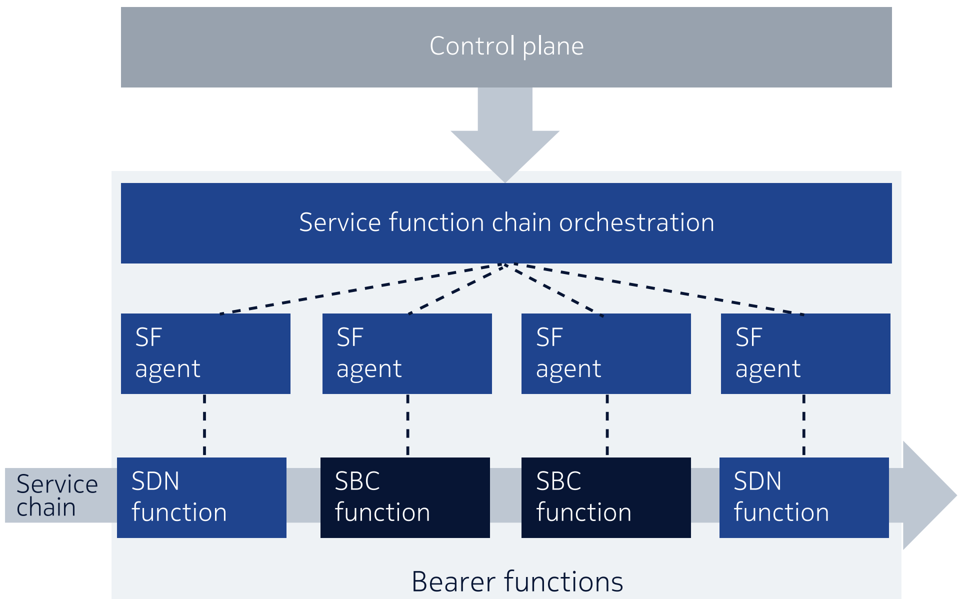

In an SDN-optimized SBC, network edge services are associated with bearer functions. These deal with real-time and high rates of packet processing that demand stringent latency and jitter requirements.

Technologies, such as SFC along with the Network Service Header (NSH), are becoming a popular combination. Figure 6 shows how these technologies can be used for decentralizing the SBC application as decoupled Service Functions (SFs) invoked in the service chain. Once decoupled, the network-optimized

NOKIA

SBC Routing and Policy engine can influence the establishment and maintenance of media streams that cross SBC boundaries from the initial admission and classification of packets at an edge access point to forwarding, routing, and transformation. The control and policy enforcement of the SBC media plane is dynamic and flexibly driven by signaling control protocols and policy rules. For realizing the benefits of this approach, Agile dynamic service insertion is a key requirement, as well.

Figure 6. Service chaining-Optimized SBC




In addition, telemetry data across all services and layers are streamed, centrally processed, and analyzed to provide a rich set of data about the SBC's operation and behavior. Each service function in the service chain contributes to the global view by monitoring and reporting its performance metrics.

SDN service modeling and provision automation underpin the management ease of the operator's intentions. Leading open-source SDN platforms, such as OpenDaylight (ODL), enable the automation of service delivery, using the provisioning capabilities for Group Based Policies (GBP), Model-Driven Service Abstraction Layer (MD-SAL), SFC, and others.

Taking advantage of a cloud-native containerized SFaaS can generate several benefits. Resource consumption can be managed at a granular level leading to savings in power, size, and footprint across the computing infrastructure. This also extends to managing the lifecycle of an individual SFaaS independent of others using a DevOps approach (i.e., the speed of introducing new service revenue generation offerings). Moreover, new service offerings can be composed by creating new SFs and inserting them into the service chain.

Traditional approaches to configuring data center networks are quickly becoming unmanageable. They do not scale with the automation and security practices required in today's cloud infrastructures. With SDN, the application of policy and operator intentions is managed from the control layer with automation tools capable of handling real-time application reconfiguration needs. The specific policies and intentions of an SBC — as an SDN application — can be co-managed alongside the SDN control layer to require specific forms of network support, such as connectivity, transport, resilience, latency, as well as additional QoS parameters. With a policy-based approach, initial deployment parameters can be bundled to meet initial connectivity and transport requirements. This policy-based approach can also make adjustments as the SBC scales and learns from the network (e.g., additional peering partners, attack vectors, and jitter).

# Conclusion

Nokia is committed to driving agility in cloud services and in advancing CSPs and enterprise digitalization. Evolving products towards a cloud-native architecture while taking advantage of emergent SDN technologies along with service chaining can deliver a more scalable, flexible, dynamic access and peering SBC.

Combining Nokia SDN and SBC offers our customers the control and choice for how service delivery is managed in their network. It also enables key benefits, including network-level optimization and flexibility, modularity and scalability, extended visibility and operations simplicity, as well as container support for automation and DevOps.

# References


1. A Cloud Native Vision for SBC insight article
2. Microservice Anatomy insight article
3. Network-Optimized Cloud Architectures SBC
4. Migrating SBCs to the Cloud white paper
5. https://datatracker.ietf.org/doc/rfc7665/
6. Network Service Header (NSH), IETF RFC Draft
7. ETSI GS NFV-EVE 005 V1.1.1 (2015-12)
8. OpenFlow
9. OpenDaylight


# Acronyms

| | |
|---|---|
| GBP | Group Based Policy |
| IoT | Internet of Things |
| MD-SAL | Model-Driven Service Abstraction Layer |
| NFV | Network Functions Virtualization |
| NIC | Network interface card |
| NSH | Network Service Header |
| QoS | Quality of Service |
| SBC | Session Border Controller |
| SC | Service Classifier |

| | |
|---|---|
| SDN | Software-Defined Networking |
| SF | Service Function |
| SFaaS | Service Function as a Service |
| SFC | Service Function Chaining |
| SFF | Service Function Forwarder |
| SLA | Service Level Agreement |
| VM | Virtual Machine |
| VNF | Virtual Network Functions |

Nokia is a registered trademark of Nokia Corporation. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

Nokia Oyj
Karaportti 3
FI-02610 Espoo
Finland
Tel. +358 (0) 10 44 88 000

Product code: SR1710017682EN (November)

© 2017 Nokia