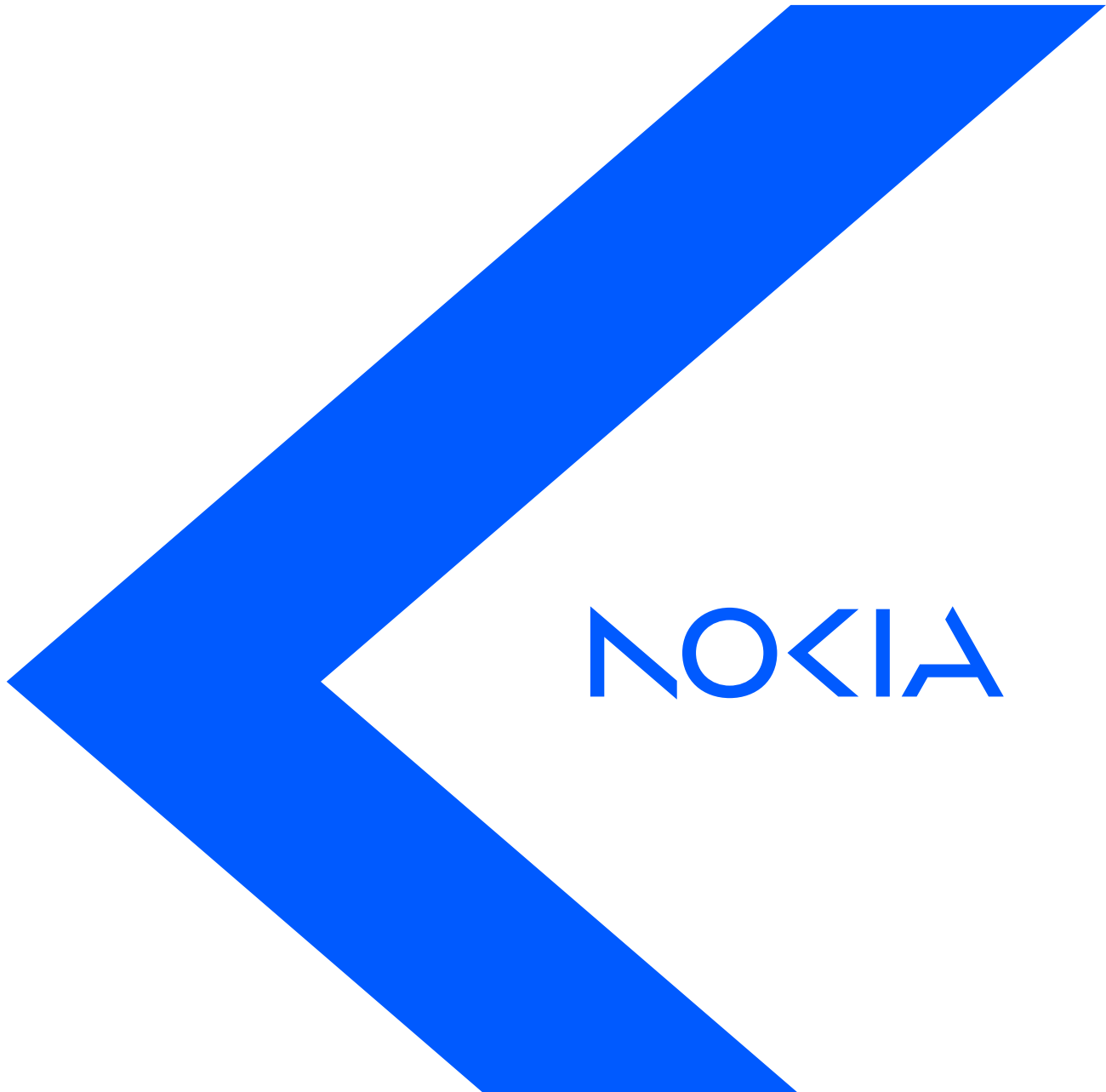


Nokia Service Router Linux

An open, extensible and resilient
network operating system (NOS)

Product description



Abstract

Cloud has become a driver of change. Digital infrastructures are constantly transforming due to content located closer to end users, applications evolving to cloud-native approaches, the adoption of cloud-edge applications and the need to be ready for technology inflections.

IP and data center networks need to keep pace and transform to become more “cloud like” with more agility, customizability, scalability and performance than previous generations of network infrastructures.

The challenge facing network operations teams is to balance the need to constantly scale their networks and operations against the increasing costs of designing, building and operating these critical networks.

Nokia Service Router Linux (SR Linux) is an open, extensible and resilient network operating system (NOS) that delivers scalability, flexibility and efficiency in modern networks.

This document describes Nokia SR Linux and explains how its unique architecture and design helps simplify the design, deployment and operation of modern networks.

Contents

| | |
|--|----|
| Abstract | 2 |
| Introduction | 4 |
| Nokia SR Linux overview | 4 |
| Supporting a diverse set of deployment use cases | 5 |
| Next-gen IP access, aggregation and edge networks | 5 |
| Modern data center switching | 5 |
| Unique architecture and design | 5 |
| Cloud-native design principles | 5 |
| Linux-based NOS | 6 |
| Modular, state-sharing architecture | 6 |
| Ground-up, model-driven management infrastructure | 7 |
| Best-in-class scalable streaming telemetry | 8 |
| Enhanced NetOps Development Kit | 9 |
| Robust foundation | 9 |
| Field-proven protocol stacks | 9 |
| Resiliency and high availability | 9 |
| Automation at scale | 10 |
| IP network automation and operations toolkit | 10 |
| Data center fabric automation and operations toolkit | 10 |
| Unmatched extensibility | 10 |
| Software extensibility | 10 |
| Superior CLI programmability | 11 |
| Build custom apps with SR Linux NDK | 11 |
| Support for open community ecosystems | 12 |
| Hardware extensibility | 12 |
| Summary | 13 |
| Abbreviations | 13 |

Introduction

Cloud has become a driver of change. Digital infrastructures are constantly transforming due to content located closer to end users, applications evolving to cloud-native approaches, the adoption of cloud-edge applications and the need to be ready for technology inflections.

IP and data center networks need to keep pace and transform to become more “cloud like” with more agility, customizability, scalability and performance than previous generations of network infrastructures.

Nokia Service Router Linux (SR Linux) network operating system (NOS) delivers scalability, flexibility and resiliency within modern networks.

SR Linux leverages Nokia expertise in IP routing and network operations, the Nokia Service Router Operating System (SR OS), our IP network automation solutions and our proven track record of building business-critical Ethernet and IP/MPLS networks for enterprises, service providers and webscale globally.

This document describes Nokia SR Linux and explains how its unique architecture and design helps simplify the design, deployment and operation of modern networks.

Nokia SR Linux overview

Nokia SR Linux was developed in close collaboration with some of the world’s largest network operators. It was designed to solve real-world challenges in IP and data center networks where the primary challenges are scalability, ease of operations or both.

In addition to scalability, the overall network must provide high levels of programmability, openness, reliability and extensibility to meet growing DevOps and agility requirements. These diverging requirements cannot be met by current industry solutions, and they were the genesis for the development of SR Linux.

Nokia SR Linux implements a unique architecture that delivers an open, extensible framework designed from the ground up for automation with advanced software features leveraging proven quality and resiliency. It provides openness, flexibility, robustness and automation to make networks easier to scale, adapt and operate.

Nokia SR Linux delivers:

- An open, extensible and resilient NOS that is fully programmable and massively scalable
- A unique architecture that embraces cloud-native principles, featuring a microservices-based, state-efficient design
- A unique, model-driven architecture designed from the ground up for simplified operations and integrations and ultimate visibility
- Field-hardened protocol stacks for enhanced IP routing and security features
- Extensive streaming telemetry, designed from the core out to support unprecedented granularity and volume
- A customizable, open-source command line interface (CLI) and on-demand, customized CLIs leveraging the flexibility of the Python programming language
- Unrivalled support for integrating community and customer-driven applications into the core of the system



- Maximum portability with support for a “free to try” version of a container SR Linux image as well as a tool (Containerlab) to easily deploy networking lab topologies
- Flexible hardware integration and support for a wide range of hardware networking chipsets.

Supporting a diverse set of deployment use cases

SR Linux is extremely portable and uses our eXtensible Data Path (XDP) to provide a hardware abstraction layer that simplifies interfacing with a broad range of hardware platforms. By supporting container and virtualized instantiations, SR Linux enables a wide range of deployment use cases and network applications.

Next-gen IP access, aggregation and edge networks

The [Nokia 7730 Service Interconnect Router \(SXR\)](#) series simplifies service evolution and network operations with programmable FPCx silicon complemented by Nokia SR Linux, which delivers high-performance IP routing and security features, and the [Nokia Network Services Platform \(NSP\)](#), which automates network management and optimizes resources across IP networks.

[Learn why SR Linux is the right choice for your IP networks](#)

Modern data center switching

Our next-generation [Data Center Fabric solution](#) includes Nokia SR Linux, together with high-performance merchant silicon-based hardware platforms for data center leaf/spine/super-spine applications. It also includes the [Nokia Event-Driven Automation \(EDA\)](#), which provides an intent-based toolkit that lets you automate all phases of the data center fabric operations life cycle.

[Learn why SR Linux is the right choice for your data center networks](#)

The following sections will review key SR Linux architecture and feature innovations and their associated benefits for enterprise, service provider and webscale IP and data center networking teams.

Unique architecture and design

Nokia SR Linux provides an open Linux®-based NOS with an innovative architecture foundation, field-hardened protocol stacks and model-driven management. It enables an open, modular and extensible networking infrastructure with support for third-party network applications.

Cloud-native design principles

Cloud-native application innovation and related technologies are making it easier to achieve the goals of business agility, continuity and innovation, while empowering staff and breaking down traditional silos.

Much like cloud-native applications, network and cloud infrastructures need to become much more dynamic and require more scalability and performance than previous generations of network infrastructure.

SR Linux embraces a cloud-native design approach, which helps deliver superior programmability, unrivalled flexibility and resilient IP routing capabilities to make network operations more agile and adaptable. Cloud-native innovations include the support for a Linux-based kernel, ground-up modular model-driven framework and a microservices-based, state-efficient design.

Linux-based NOS

In the past, NOS's were closed, proprietary systems. Openness is best achieved with an underlying open operating system such as Linux. Nokia SR Linux is a Linux-based NOS, which builds on the Linux kernel to provide a set of loosely coupled services that come together to provide the functional blocks of an NOS, along with simple interfaces for their consumption.

Nokia SR Linux provides a set of modular applications that are isolated into their own failure domains. A central application manager is responsible for the life cycle of each application and provides full control of the protocols running on the system. As a result, each operator needs to run only the applications required for a given environment; this simplifies administration and improves security.

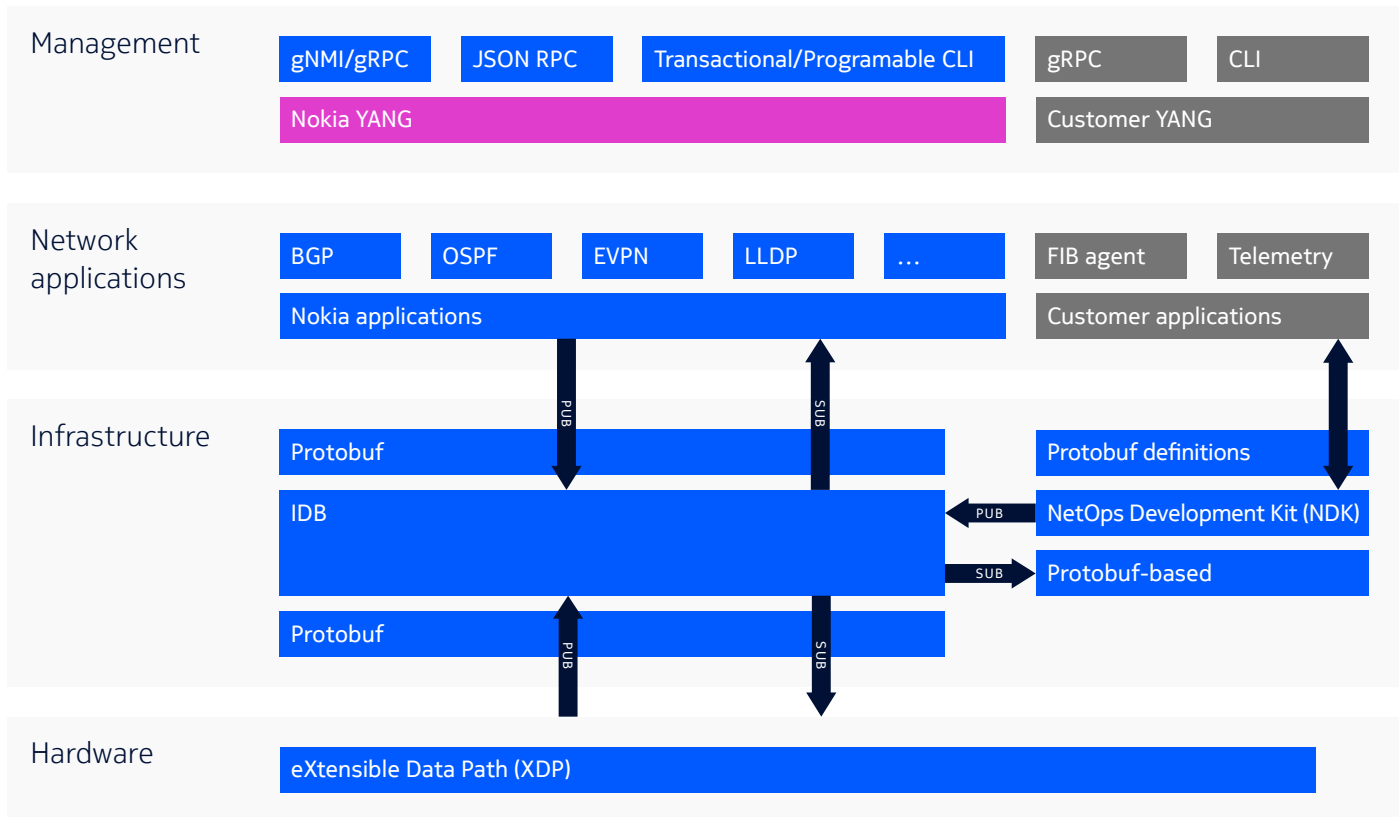
Unlike some other Linux-based NOS's in the industry, SR Linux uses an unmodified Linux kernel as the foundation on which to build a suite of network applications. This provides many benefits, including reliability, portability and ease of application development. Using an unmodified kernel also speeds the availability of non-Nokia applications (for example, OpenSSH) and security patches for operating system components.

Modular, state-sharing architecture

Nokia SR Linux applications share state with each other through a publish/subscribe (pub/sub) architecture (see Figure 1). The Nokia pub/sub architecture uses two recent open-source innovations: protocol buffers (protobufs) and generalized Remote Procedure Call (gRPC).

Previous-generation NOS's were forced to implement proprietary and non-performant inter-process-communication (IPC), defeating any hope of extensibility, and leading to potential resource starvation and inconsistencies under convergence scenarios. Protobufs and gRPC were chosen because they are both extremely fast, efficient and scalable, which are key attributes for any NOS. In addition, external applications can be supported with no extra effort required.

Figure 1. Nokia SR Linux architecture provides an open infrastructure



SR Linux has a state-sharing architecture that relies on the Nokia Impart Database (IDB), a lightweight database developed by Nokia specifically to meet the performance demands of network applications. The Nokia IDB runs in memory and is optimized for handling high volumes of messages while protecting against any one application slowing down the whole system.

The IDB acts as a highly scalable key-value store, where both the keys and values are serialized protobufs, and IDB is completely agnostic to the data being published. This promotes simplicity and scalability in the architecture and provides extensibility at the heart of the system.

In addition, the IDB uses backpressure mechanisms and intelligent message coalescing to ensure that neither a chatty publisher nor a slow subscriber can negatively impact any other applications. Message coalescing allows subscribers that are under load to process updates when they can, ensuring that the intent of the operator will be met, even when network convergence events drive contention for resources.

Unlike other pub/sub architectures that act as simple message queues, the IDB provides a reliable and scalable delivery mechanism, guaranteeing delivery of updates to subscribers. gRPC provides an efficient, secure communication channel for applications and allows native extensions through third-party applications.

Ground-up, model-driven management infrastructure

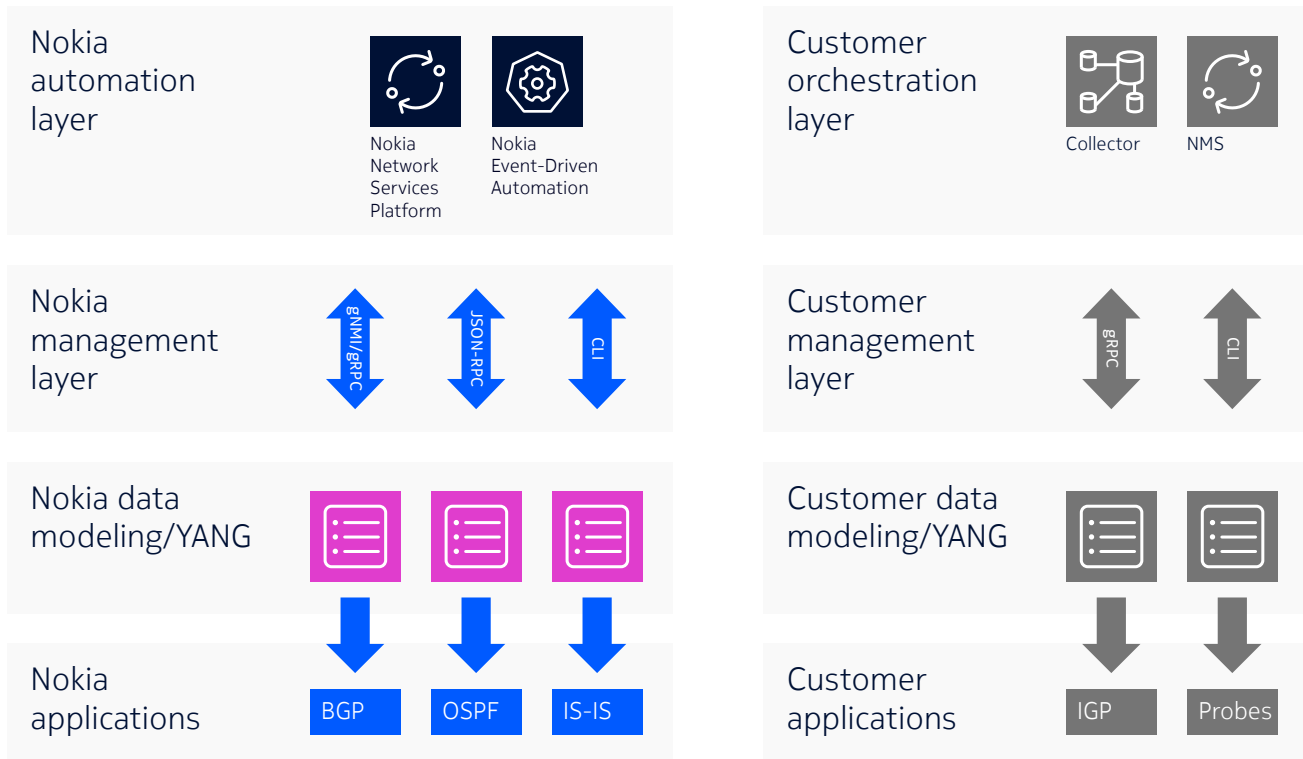
Openness cannot be an afterthought and must be designed from the foundation. Nokia believes a strong data model is the foundation for enabling rich and flexible automation. A data model defines the language between the operating system and the outside world of users and northbound systems. The data model defines the syntax and semantics on which to base applications, scripts and documentation.

In the past, operators relied solely on Simple Network Management Protocol (SNMP) data models, which are often proprietary, are not human-readable and are inflexible to changes. As an alternative, operators often reverted to CLI scripting and “screen scraping” for automation. Although functional, neither of these options is scalable.

SR Linux implements a truly open architecture built around modular model-driven management and modern interfaces for its consumption (see Figure 2). Nokia SR Linux features a complete and ground-up model-driven architecture that delivers flexible and simplified management and operations.

Our core design approach opens up the infrastructure by adopting emerging technologies, to allow services to expose their functionality in the form of gRPC services and protobufs. Our fully modular design, where each network application has its own YANG data structure, delivers complete openness. It gives network applications the ability to simplify their internal schemas based on data modeling languages, and then expose these schemas directly for consumption by northbound interfaces: any object, any interface.

Figure 2. Ground-up, model-driven management infrastructure



Best-in-class scalable streaming telemetry

SR Linux is designed to meet the demands of a model-driven world where visibility—and the scalability and granularity of that visibility—are paramount. SR Linux delivers an open, extensible and performant infrastructure that allows the retrieval of fine-grained system state, setting of configuration and a scalable interface to support more granular data with push-based streaming.



SR Linux was built with an open, scalable telemetry framework at its core, where the system and applications internally speak the same transport and message-declaration mechanisms as used in gRPC, gRPC Network Management Interface (gNMI) and protobufs. Because SR Linux is natively model-driven, it is immediately ready for streaming telemetry without requiring any translation layers.

Enhanced NetOps Development Kit

Nokia SR Linux allows third-party applications to be fully integrated into the system with the same functionality as Nokia applications. This includes consistent configuration using YANG, telemetry support, life-cycle management and visibility of system resources.

The Nokia NetOps Development Kit (NDK) enables networking teams to leverage SR Linux's underlying model-driven architecture, with a simple, clean, decoupled integration. This allows all applications in the system to support data modeling, transactional configuration and—most important—massively scalable streaming telemetry.

The NDK uses gRPC and protobufs to provide maximum flexibility for languages supported and backwards compatibility. This approach differs from others, which are restricted to certain languages, versions and/or libraries.

With the mindset to build for openness, common infrastructure utilized by Nokia applications is exposed, allowing a uniform operating model and visibility deep into the heart of the system.

Robust foundation

A robust NOS is characterized by operating system software stability and its ability to support field-proven, scalable and reliable software feature sets.

Field-proven protocol stacks

Nokia designed SR Linux for the DevOps era using the field-proven protocol stacks from the Nokia Service Router Operating System (SR OS). The Nokia SR OS has a strong pedigree in IP routing developed over the past 20+ years. Nokia customers have deployed over 1.7 million routers in over 2,500 IP networks, including the internet backbone and some of the largest service provider networks in the world.

By using these field-proven protocol stacks, network design and operations teams can immediately benefit from a feature-rich, hardened, interoperable and secure NOS.

Resiliency and high availability

Hardware and software high availability is crucial for ensuring maximum system uptime in IP and data center networks. SR Linux-enabled hardware platforms support redundant fan and power configurations as well as hot-swappable, redundant control and fabric modules in modular chassis-based hardware platforms.

SR Linux microservices-based, state-efficient design paves the path for enabling per-application hitless upgrades and always on networking.

SR Linux supports warm reboot features that can be used to perform a soft reset or trigger an in-service software upgrade (ISSU). Nonstop forwarding (NSF) capabilities minimize data plane outages. NSF capabilities also enable hardware platforms to continue to forward packets. While leveraging control plane graceful restart, peers can continue to pass traffic.

Automation at scale

SR Linux was designed and built to enable confident automation and operations at scale. Its ground-up, model-driven, and modular architecture with independent data modeling protocols like YANG and OpenConfig and open scalable foundation is well suited for telemetry requirements in modern networks. Network management and automation platforms can immediately benefit from the granular and comprehensive multidimensional telemetry data from SR Linux to monitor and gain deep insights into network traffic. This helps understand the current state of the network and compare with desired intent to identify and implement the required changes to the network.

IP network automation and operations toolkit

In combination with the [Nokia Network Services Platform \(NSP\)](#), the 7730 Service Interconnect Router (SXR) platforms can be deployed to introduce scalable and integrated network management and automation in the WAN. The NSP offers a complete suite of ready-to-use applications that help your network operations teams cover all use cases for network management, orchestration and control. It offers an open programmable platform that enables network operations teams to automate network operations and ease integration with orchestrators and operations support systems (OSS).

Data center fabric automation and operations toolkit

The [Nokia Event-Driven Automation \(EDA\)](#) complements and extends the capabilities provided by our SR Linux architecture foundation to enable a full turnkey data center solution for automation, intent-based approaches, telemetry collection and analytics. Nokia Event-Driven Automation (EDA), a modern data center network automation platform that combines speed with reliability and simplicity and provides guardrails that detect errors caused by automation.

Unmatched extensibility

SR Linux enables network operations teams to leverage SR Linux's underlying open architecture to deliver unprecedented customizability.

Software extensibility

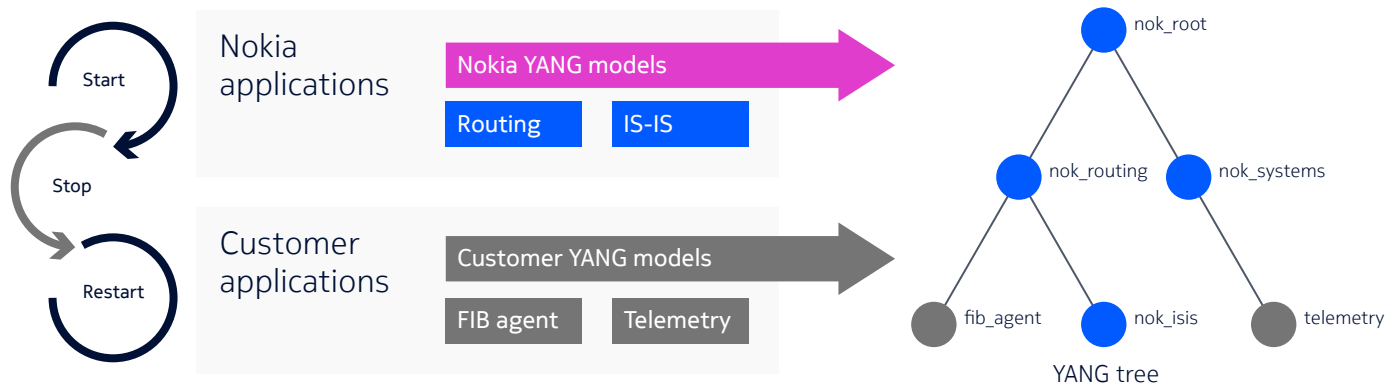
Nokia has built a ground-up management architecture to meet the demands of a model-driven world where visibility—and the scalability and granularity of that visibility—are paramount. YANG has become the networking industry standard for data modeling because it is human readable, extensible and easy to learn.

Every SR Linux application (including third-party applications) supports its own YANG model, which can be loaded into the system. Operators can see and define the syntax and semantics of their application in a simple and standardized form. With this design, the YANG data model is defined first, and from it operators can derive the CLI, the APIs, the show-output formats and even the documentation of every feature of the system.

SR Linux handles management and operations through gNMI. Because SR Linux is natively model-driven, it is immediately ready for streaming telemetry without requiring any translation layers.

Third-party applications also have access to the full streaming telemetry framework. This enables applications developed for the SR Linux architecture to be operationalized, monitored, configured and debugged in the same way as any other application on the system, with standard open interfaces (see Figure 3).

Figure 3. Nokia SR Linux software extensibility



Superior CLI programmability

In addition to the gNMI interface, SR Linux includes an advanced, Python-based CLI and a JSON-RPC API for management. The CLI provides a flexible framework for accessing the system’s underlying data models and is based on quality-of-life (features that improve the operator experience) embraced by DevOps communities.

Operators can also leverage plugins to completely customize the way the CLI operates, plugging in Linux commands, or pulling the state/configuration from various locations, combining them with system state/configuration to allow advanced logic. This capability streamlines the adoption of SR Linux because the interface can be customized.

The JSON-RPC API provides a modern interface that supports queries to the data models as well as allowing programmable interface access to the extensible plugin framework in the CLI. This access allows simplified use of operator customizations, tying them directly into the automation layer.

Build custom apps with SR Linux NDK

Nokia SR Linux enables network teams to create high-performance applications, which run alongside native apps on SR Linux. These “on-box custom applications” can be deeply integrated with the rest of the SR Linux system and thus can perform tasks that are not possible with traditional management interfaces standard for the typical network operating systems.

Nokia SR Linux allows third-party applications to be fully integrated into the system with the same functionality as Nokia applications. This includes consistent configuration using YANG, telemetry support, life-cycle management (LCM) and visibility of system resources.

The NDK allows network operators to teach the network their language and respond directly to business demands—all without worrying about the scalability or functionality of the routing stack or the underlying infrastructure.

In addition, the SR Linux NDK provides full access to the IDB, CLI and APIs. The NDK uses gRPC and protobufs to provide maximum flexibility for languages supported and backwards compatibility. This approach differs from others, which are restricted to certain languages, versions and/or libraries. The other approaches also require third-party applications to be managed independently, which creates additional operational overhead.

The NDK provides the ability to be ready for technology inflections; for example, it's easy for operational teams to learn and experience how AI/ML tools like ChatGPT can easily be integrated with their IP routers. With SR Linux, this is as simple as building an NDK application for that. To learn more about the SR Linux NDK, refer to the [SR Linux portal](#).

Support for open community ecosystems

- **Public SR Linux container image:** It is a well-established fact that learning by doing yields the best results. With that in mind, we offer the SR Linux container image at “no cost” and available without any registration or licensing requirements. The SR Linux image is available to everyone at a [publicly accessible GitHub container registry](#).
- **Containerlab:** [Containerlab](#) is a tool designed to deploy networking lab topologies. The public SR Linux container image, when powered by containerlab, allows network operations teams to deploy and configure their lab with network protocols and services they require and test the control and data plane functions. Containerlab supports native containerized NOS's as well as traditional virtual machine-based routers.
- **Integration with open ecosystems and tools:** SR Linux supports integration with DevOps and automation tools like Ansible, NAPALM, NetBox, gNMIc and more. To learn more about SR Linux programmability and integration with open ecosystems and tools, refer to <https://learn.srlinux.dev/programmability/>.
- **Learn SR Linux portal:** SR Linux packs many unique features that networking teams can leverage, several of which are truly new to the networking domain. Our [SR Linux portal](#) features tutorials, documentation, blogs, programmability and developer tools (YANG, NDK, event handler, plugins and more).

Hardware extensibility

Hardware flexibility is an essential requirement for network operators. Nokia SR Linux supports a variety of networking chipsets through the Nokia eXtensible Data Path (XDP).

The Nokia XDP serves as a hardware abstraction layer that speeds time-to-market for new or different networking chipsets. The XDP provides a common set of software instructions that northbound applications use so they are not directly dependent on ASIC-vendor SDKs. The XDP borrows from the Nokia development experience for high-performance virtual network functions and makes use of user space acceleration for traffic destined to the control plane and any non-ASIC interfaces.

Summary

As the demands on IP and data center networks continue to drive openness, flexibility and efficiency, Nokia is ready to support network operations teams with improving their design, deployment and day-to-day operations practises.

Nokia SR Linux is a truly open, extensible, and resilient NOS. It implements a ground-up, model-driven architecture combined with proven and mature routing protocol stacks to create a unique foundation that delivers superior scalability, flexibility and efficiency in modern networks. This superior design foundation includes extensive telemetry, unrivalled third-party application support and plug-and-play integration—all critical features for modern networking.

Abbreviations

| | | | |
|--------|--|----------|--|
| API | application programming interface | MP-BGP | Multiprotocol Border Gateway Protocol |
| ASIC | application-specific integrated circuit | MPLS | Multiprotocol Label Switching |
| BGP | Border Gateway Protocol | NDK | SR Linux NetOps Development Kit |
| CLI | command line interface | NMS | network management system |
| DevOps | development and operations | NOS | network operating system |
| EVPN | Ethernet virtual private network | OSF | Open Software Foundation |
| FIB | forwarding information base | Pub/sub | publish/subscribe |
| gNMI | gRPC Network Management Interface | protobuf | protocol buffer |
| gRPC | generalized Remote Procedure Call | RPC | remote procedure call |
| IDB | Nokia Impart Database | SDK | software development kit |
| IGP | Interior Gateway Protocol | SR Linux | Nokia Service Router Linux |
| IP | Internet Protocol | SR OS | Nokia Service Router Operating System |
| IS-IS | Intermediate System to Intermediate System | SSH | Secure Shell |
| JSON | JavaScript Object Notation | VXLAN | Virtual eXtensible Local Area Network |
| LCM | life-cycle management | XDP | eXtensible Data Path |
| LLDP | Link Layer Discovery Protocol | YANG | Yet Another Next Generation (data modeling language) |



About Nokia

At Nokia, we create technology that helps the world act together.

As a B2B technology innovation leader, we are pioneering networks that sense, think and act by leveraging our work across mobile, fixed and cloud networks. In addition, we create value with intellectual property and long-term research, led by the award-winning Nokia Bell Labs.

Service providers, enterprises and partners worldwide trust Nokia to deliver secure, reliable and sustainable networks today – and work with us to create the digital services and applications of the future.

Nokia operates a policy of ongoing development and has made all reasonable efforts to ensure that the content of this document is adequate and free of material errors and omissions. Nokia assumes no responsibility for any inaccuracies in this document and reserves the right to change, modify, transfer, or otherwise revise this publication without notice. Nokia is a registered trademark of Nokia Corporation.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

© 2024 Nokia

Nokia OYJ
Karakaari 7
02610 Espoo
Finland
Tel. +358 (0) 10 44 88 000

701650 (November) CID207604