

Model-driven management with the Nokia SR OS

Simplify and automate IP network operations

Application note

The Nokia logo is displayed in a blue, sans-serif font. It is positioned in the lower right quadrant of the page. A large, solid blue diagonal shape, resembling a stylized 'N' or a thick line, runs from the top left towards the bottom right, partially overlapping the logo area.

NOKIA



Abstract

The evolution of network management is driving the need for model-driven programmability and automation. The [Nokia Service Router Operating System \(SR OS\)](#) provides the foundation for the comprehensive [Nokia IP router portfolio](#). This application note explains how model-driven management enabled by the Nokia SR OS can help network operators succeed with their IP network evolution initiatives by simplifying and automating IP network operations.

Contents

Abstract	2
Network automation challenges	4
Nokia SR OS: Model-driven management	4
Modeling	5
Transport and encoding	5
Interfaces	5
APIs	6
YANG	6
NETCONF	7
gRPC	8
MD-CLI	10
Smooth migration to the MD-CLI	10
MD-CLI based on YANG models	10
Transactional configuration	10
Configuration access control and safety	10
Templating with configuration groups	10
Getting started with SR OS programmability	11
Using SR OS model-driven management applications	11
Python integration and customization API	11
Learning about SR OS network programmability	11
Summary	11
Learn more	11
Abbreviations	12

Network automation challenges

Network operators have been looking for ways to simplify repetitive tasks in multivendor networks for many years. The massive increase in demand for 5G and cloud services is pushing the limits of network scaling. Operators now need to manage tens of thousands of network devices while ensuring that they deliver innovative services in a timely manner and provide an excellent quality of experience for customers.

Network automation and the move to a NetOps-centric approach are critical elements that can help deliver operational efficiency and cost savings. A key part of this paradigm shift is the adoption of model-driven approaches, which are fundamentally different from the present mode of operation and deliver some dramatic improvements compared to the current process.

In the past, operators needed to create complex adapters that could translate between their management systems and vendor-specific command line interfaces (CLIs). This approach is cumbersome, error prone and time consuming. Operators had to invest significant time and effort to adapt to new equipment software releases because changes to CLIs can easily break previous integrations and because CLIs often have inconsistent behavior.

By contrast, model-driven management approaches create consistent, rule-based behavior that is much easier to adapt and integrate. A key aspect of reducing integration effort across vendors, products and releases is the use of a common industry-wide data modeling language. Data modeling based on the Yet Another Next Generation (YANG) language can help reduce the cost of automating the management of network equipment. Devices can then be configured using programmatic interfaces with protocols such as the Network Configuration Protocol (NETCONF), gRPC Remote Procedure Call (gRPC) or the model-driven CLI (MD-CLI).

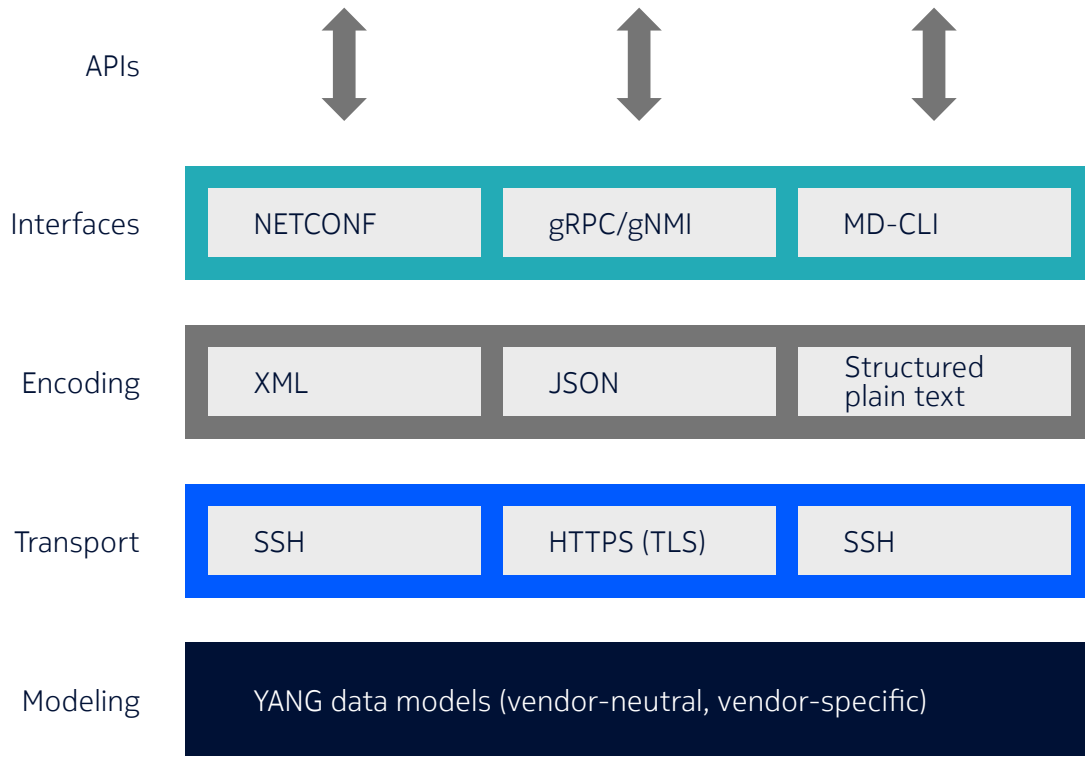
Nokia SR OS: Model-driven management

The Nokia Service Router Operating System (SR OS) is built to power the most demanding, dynamic and reliable Ethernet and IP/MPLS networks. Nokia uses this robust and scalable OS to provide the foundation for a comprehensive portfolio of physical and virtualized routers. The SR OS has been proven in more than 2,300 service provider, enterprise and webscale networks worldwide.

The primary driver for model-driven management is to move from manual, CLI-based router configuration and control to centralized, automated orchestration based on industry-defined methods. This approach enables simpler, more reliable and more efficient network management.

The Nokia SR OS model-driven management framework focuses on three key areas: programmability, visibility and operations. Programmability allows humans and machines to use programmatic interfaces to access structured data based on YANG models. Visibility provides statistics and state information and includes the use of telemetry to push operational data to northbound management systems. Operations provides the ability to consistently manage network devices in automated network environments using the CLI and programmatic interfaces such as NETCONF and gRPC. Figure 1 shows the elements and protocols that enable model-driven programmability with the Nokia SR OS.

Figure 1. Enabling model-driven programmability with the Nokia SR OS



Modeling

YANG-based data modeling delivers the foundation for model-driven programmability within the SR OS. These data models provide a standards-based modeling for configuring and operating IP routers. YANG data models are human-readable, structured data models that define each network construct with its constraints and dependences. The SR OS supports a comprehensive set of rich data models for configuration, state and operations.

Transport and encoding

Transport and encoding are based on protocols standardized by the industry. Secure Shell (SSH) for the MD-CLI is used with structured plain text encoding. NETCONF runs over SSH with eXtensible Markup Language (XML) encoding. gRPC/gNMI runs over HTTPS (TLS) with various encodings, including JavaScript Object Notation (JSON). The broad choice of protocols and the separation of encoding from the data model provide significant flexibility for implementations.

Interfaces

The SR OS supports model-driven interfaces such as NETCONF, gRPC and the MD-CLI. These interfaces are all based on the same YANG models, which ensures the same look and feel across all interfaces. For example, if an operator sends a configuration with NETCONF and then views this configuration in the MD-CLI, the operator sees the exact same configuration. This makes it easy to integrate with network automation systems because machines and humans have access to and interpret the same thing, ensuring consistency between what the automation systems send and what the human operators type.

APIs

The SR OS allows applications to manage IP networks using the NETCONF, gRPC and MD-CLI interfaces. It also provides a Python 3 programming interface using the pySROS library to enable automation and integration either on the IP router or from a remote server.

The following sections discuss several key elements of the SR OS model-driven framework in more detail.

YANG

At the root of the desire to automate IP networks is a need to represent constructs such as network elements, policies or services in a consistent language. The representation model must be a language that humans and machines can understand without significant processing overhead.

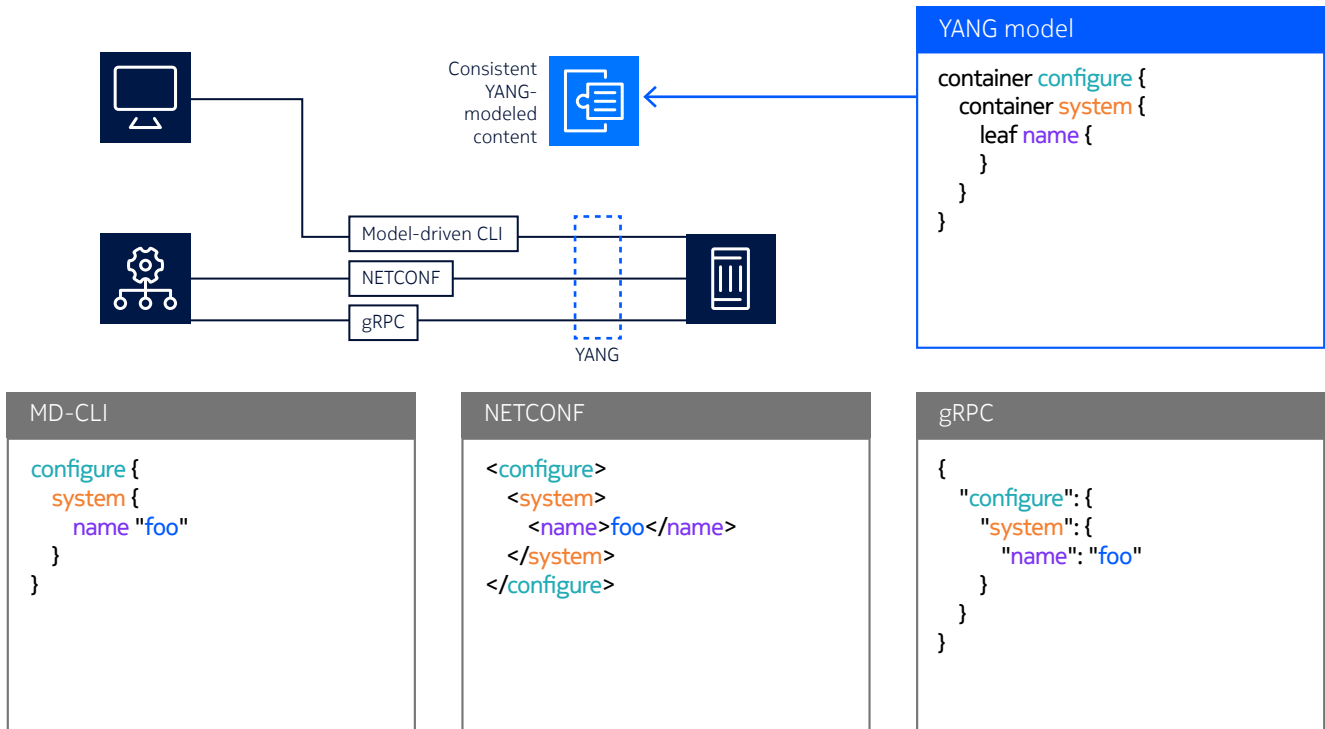
Simple Network Management Protocol (SNMP) has been the dominant protocol for managing and monitoring networks over the past few decades. However, SNMP and the Structure of Management Information Version 2 (SMIv2) modeling language lack the ability to provide human-readable and machine-readable models, which are essential for the massive-scale network automation required for the networks of the future.

The YANG data modeling language for configuration, state and operations is defined in several IETF RFCs that specify how data is organized hierarchically. YANG is designed to be readable by humans and machines. It is the standard way to model network devices and network device information. YANG models can also communicate constraints and dependencies in addition to the types and ranges that SMIv2 provides.

YANG models are at the heart of the Nokia SR OS and provide a fundamentally different approach to managing network devices. Model-driven management ensures that configuration, state and operations are defined by the YANG models provided with the SR OS software release.

The SR OS adds standardized communication interfaces to this modeled information so that IP router management has a common look and feel and is consistent across the MD-CLI, NETCONF and gRPC interfaces (see Figure 2). Integration with many open source tools is straightforward and does not require proprietary vendor integration.

Figure 2. YANG providing consistency across model-driven interfaces



The SR OS approach to model-driven management makes programming over NETCONF or gRPC intuitive to MD-CLI operators. This simpler programming complements the network engineering teams' efforts to develop workflows and pipelines as part of their broader model-driven management and automation initiatives.

The Nokia SR OS is a unified routing OS that is used for the entire Nokia IP routing portfolio. YANG modeling is available for all functionality supported by the SR OS. This ensures consistency and simplifies operations for IP routers that are deployed across different domains in the IP network.

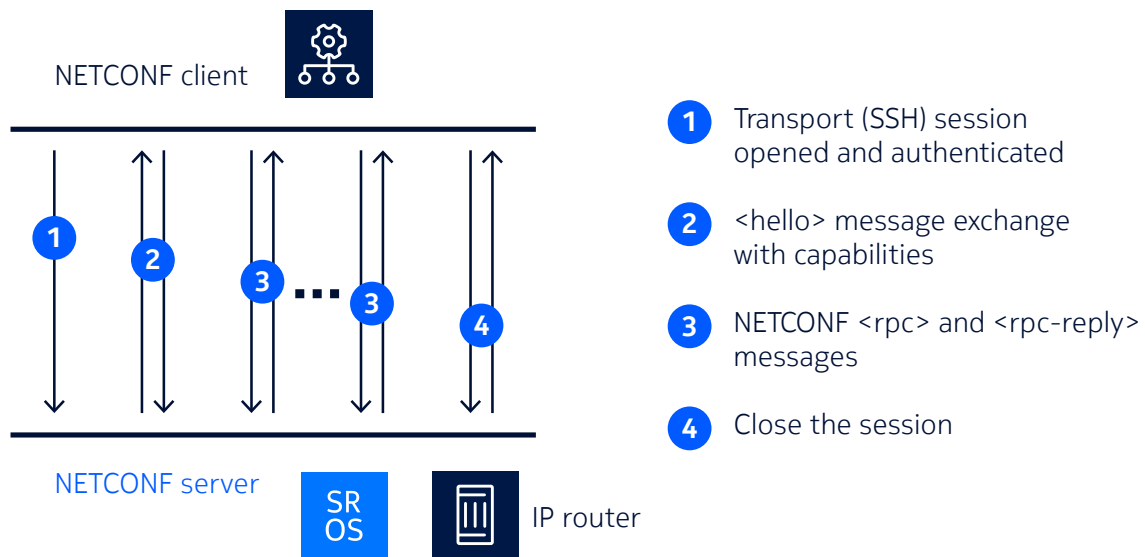
For additional information on SR OS YANG models, visit the [Nokia SR OS Network Developer Portal](#) or the [Nokia SR OS YANG models](#) repository on GitHub.

NETCONF

Before the development of NETCONF, the only way to make automated configuration changes was to use CLI scripting or SNMP. CLI scripting lacked error management and often had changing syntax and a lack of structure, which made it costly and complex to manage. SNMP has historically been used for fault and performance monitoring. While SNMP has the capability to write changes, the data is not modeled in a flexible way, and there is no correlation between SNMP object identifiers and the CLI.

NETCONF, as defined in [IETF RFC 6241](#), provides mechanisms to manage network devices. NETCONF uses XML data encoding for modeled configuration data and protocol messages. The NETCONF operations are performed on top of a simple RPC layer. Figure 3 shows a NETCONF communications workflow.

Figure 3. NETCONF communications workflow



NETCONF accesses one or more datastores. A datastore is the complete set of data that is required to provision a device from its initial default state into the desired operational state.

The SR OS supports a comprehensive set of NETCONF features, including:

- Extensive RPC mechanisms
- Datastores:
 - To store and access start up configuration at device boot
 - Running (currently active) and candidate (working) configurations
 - To view operational configuration and state
- IETF NETCONF monitoring capability, which allows the device to be queried for supported YANG models
- Transactional configuration
- Execution of operations modeled in YANG

gRPC

gRPC is an open source, high-performance RPC framework that can run in many environments. gRPC enables client and server applications to communicate transparently and makes it easier to build connected systems.

The gRPC Network Management Interface (gNMI) protocol is a unified management service that is built on top of gRPC. It defines how configuration operational data interacts with a network element.

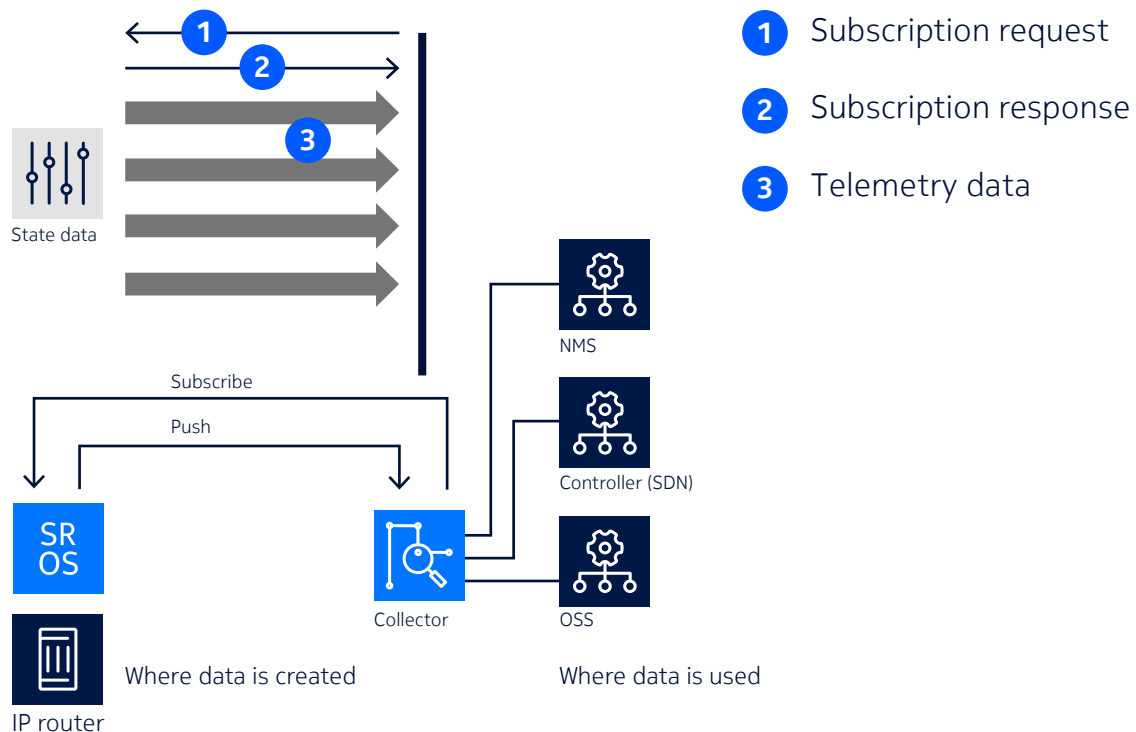
Like NETCONF, gRPC has a client-server architecture. While gRPC supports configuration management, its main use case is for monitoring using streaming telemetry. Streaming telemetry does not rely on collectors continuously pulling data from the network elements. Instead, network elements push statistics and operational state data to the subscribing collector(s) based on defined paths or frequencies or on changes of values.

The Nokia SR OS gRPC implementation supports several use cases, including:

- Configuration management: Configuring and retrieving network element configuration information
- Telemetry monitoring: Pushing operational information from network elements to management systems
- Dial-in and dial-out connections: Reaching remote management systems
- Automation of operations: Automating tasks such as certificate management, software upgrades and file management

gRPC uses protocol buffers (protobufs), which provide a flexible, efficient, automated mechanism for serializing structured data. The SR OS supports multiple protobuf-defined services, including gNMI and elements of the gRPC Network Operations Interface (gNOI). A protobuf-defined service is a set of predefined protobufs that perform networking operations. For example, the gNMI protobuf-defined service provides the Capabilities, Get, Set and Subscribe RPCs. The Nokia SR OS gRPC implementation supports dial-in and dial-out connections, as shown in Figure 4.

Figure 4. Model-driven telemetry with gRPC (dial-in connection)



Telemetry subscriptions can request various streaming modes for the returned data:

- A SAMPLE stream subscription tells the server to provide updates at regular intervals, for example every 10 seconds.
- An ON_CHANGE stream subscription tells the server to provide updates only when there has been a change.
- A TARGET_DEFINED stream subscription allows the client to delegate the selection of SAMPLE or ON_CHANGE streaming to the server.

These subscription modes provide flexibility regarding how much statistics data can be streamed and how often depending on the architecture implemented for telemetry. Telemetry data can be returned in a variety of encodings, including JSON, JSON_IETF, BYTES or PROTO.

MD-CLI

Nokia has evolved the SR OS to support the MD-CLI. The SR OS MD-CLI is designed around model-driven concepts. It includes several key design attributes and features that are critical for model-driven programmability.

Smooth migration to the MD-CLI

For operators who have existing SR OS deployments, it is imperative to ensure a smooth transition to the enhanced SR OS MD-CLI. The SR OS MD-CLI was developed with this in mind, and the user interface provides a familiar experience to operators, along with the features of a model-driven interface. The SR OS MD-CLI provides enhanced functionality and familiarity for operators and developers. To ease the transition for existing SR OS deployments that use the classic CLI, the system automatically migrates the IP router configuration when the operator decides to make the move — and not before. The SR OS does not force operators to migrate to the MD-CLI.

MD-CLI based on YANG models

The MD-CLI is designed to provide consistency across all model-driven interfaces based on the implementation of YANG models. This ensures that all model-driven interface management over NETCONF, gRPC or the MD-CLI provides the same results and is consistent, making it easy to operate, automate and troubleshoot.

Transactional configuration

Usability and ease of operation were key design considerations for the SR OS MD-CLI. It is essential for the operator to be able to control when changes are applied. The MD-CLI supports transactional configuration with configuration rollback as a standard capability in the implementation. When an operator makes changes in the MD-CLI, the changes go into a candidate configuration. This candidate configuration takes effect when the operator is ready to apply (commit) the changes.

Configuration access control and safety

The MD-CLI supports configuration access control and safety for operating in a multiuser environment. Operators can gain exclusive access to the configuration so that no one else can make changes at the same time. This ensures configuration safety. Private candidates provide the operator with a private copy of the configuration that can be edited without another operator interfering with the configuration changes. Read-only candidates allow operators to monitor candidate changes made by other humans or machines while the changes are in progress.

Templating with configuration groups

Templating helps simplify configuration. For example, if the operator wants to apply a setting to 1,000 ports without a template, the operator would need to make that configuration change 1,000 times. With templating, the operator can define a template with required commands and then apply that template to each port instead of applying each command individually. Any subsequent changes are also applied to all the ports at the same time, which makes configuration simpler and more flexible.

For additional details about the Nokia SR OS MD-CLI, see the [Model-driven CLI \(MD-CLI\) configuration videos playlist](#).

Getting started with SR OS programmability

Using SR OS model-driven management applications

The SR OS provides a choice of interfaces for managing IP routers. Operators can choose an interface based on what they are trying to do and which technology or protocol best fits into their operational framework and plans.

- NETCONF has been developed to manage configurations on network devices. It is a full-featured configuration management protocol that offers benefits such as locking, multiple datastore access, and transactional configuration and operations.
- gRPC is a newer technology than NETCONF. Telemetry is a key use case for gRPC, but gRPC is also used for configuration and operational tasks.

Python integration and customization API

Model-driven management provides structured data for input and output. This allows for SR OS to be easily integrated into any tool or application. SR OS provides a built-in Python interpreter and pre-installed model-driven Python library (pySROS) on all systems. The library can be downloaded for use on servers and workstations.

With model-driven concepts coupled with extensions in SR OS, the integration journey has never been easier. Whether the goal is to create new MD-CLI commands on the SR OS device or plug SR OS devices into existing business systems, pySROS provides a model-driven, user-friendly programming API that helps operators complete the job in minutes.

Learning about SR OS network programmability

The Nokia SR OS Developer Portal is designed to help developers integrate SR OS network devices into their own infrastructure. The portal provides tips, examples and tutorials to help developers use Nokia-developed tools and third-party open-source tools to integrate with the SR OS-based product families that run many of today's global IP networks.

To learn more, visit the [Nokia SR OS Developer Portal](#).

Summary

The continuing evolution of network management is driving the need for model-based programmability and automation. The Nokia SR OS supports a comprehensive set of features that can help network operators with their IP network evolution initiatives. These features include YANG models that provide configuration, state and operations models designed for automation. They also include NETCONF, gRPC and MD-CLI interfaces based on the YANG models, which provide a set of model-driven management interfaces with a common human- and machine-readable configuration syntax. Operators can take advantage of these features to enable simplified, accelerated and cost-optimized IP network operations.

Learn more

To learn more about how to simplify and automate network operations with the Nokia SR OS:

Visit the [Nokia SR OS Developer Portal](#)

Watch the [Model-driven CLI \(MD-CLI\) configuration video playlist](#)

Visit the [Nokia SR OS web page](#)

Abbreviations

API	application programming interface
CLI	command line interface
gNMI	gRPC Network Management Interface
gNOI	gRPC Network Operations Interface
gRPC	gRPC Remote Procedure Call
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
IETF	Internet Engineering Task Force
IP	Internet Protocol
MD-CLI	model-driven command line interface
MPLS	Multiprotocol Label Switching
NETCONF	Network Configuration Protocol
NMS	network management system
OSS	operations support system
RFC	Request for Comments
RPC	Remote Procedure Call
SMIv2	Structure of Management Information Version 2
SNMP	Simple Network Management Protocol
SR OS	Service Router Operating System
SSH	Secure Shell
TLS	Transport Layer Security
XML	eXtensible Markup Language
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. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

© 2023 Nokia

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

Document code: (June) CID207070