



White Paper

# SELECTING NETCONF/YANG MANAGEMENT INTERFACE SOFTWARE

A 5th Layer white paper produced for Enea

Author: Joe Kidder, Principal, 5th Layer Consulting

**ENEAA**

# CONTENTS

Demand for Automated Network Management .....	3
How to Automate Networks? How to Make Network Devices Programmable? .....	3
What Alternatives? How Did We Get Here? .....	3
NETCONF and YANG .....	3
A Clean Approach to Using NETCONF/YANG for Network Management .....	4
Benefits of Clean Living .....	5
What to Look For in a NETCONF/YANG Offering .....	5
Product Features .....	6
Product Core .....	7
Support .....	8
Vendor Profile .....	8
Summary .....	8

*This paper discusses the increasing demand for automated network management, why an approach based on NETCONF and YANG enables programmable network devices, and how to evaluate and select a NETCONF and YANG solution for your network devices.*

## **Demand for Automated Network Management**

Today's networks provide constant connectivity, across greater distances and into every corner of the world, with whatever quality of service is necessary for the right user experience and for as little cost as possible. Network endpoints are human users, applications on phones, datacenters and small business. And things – sensors, actuators, robots – that create grids of awareness and capability.

The services and applications delivered by networks are created quickly and can spread like wildfire, going from the spark of an idea to millions of users.

This increases the speed at which network devices are added, removed, and moved within networks and how their configurations and relationships are changed. Human operators increasingly prefer to express policies and specify their intent for the desired network behavior rather than configuring each device making up the network. Automation and programmability transform this policy and intent into configuration and monitoring at the device level.

## **How to Automate Networks? How to Make Network Devices Programmable?**

Let's discuss basic needs for automation of network management by looking at how to make the network devices programmable.

Programmability in network device management starts with a model of what can be configured and monitored in the device. Modeling approaches and languages range from general purpose - a UML model, an XML schema (XSD or DTD), or a SQL database schema - to network-oriented modeling languages like YANG or SMI/ASN.1 used in SNMP MIBs.

Programmability also requires an agreed upon protocol or method to exchange modeled information and specify what is done with the data. Usually called a management protocol, this can be network-specific, like SNMP or NETCONF, or general purpose, like XML-RPC, ODBC or REST APIs.

## **What Alternatives? How Did We Get Here?**

How did we get so many choices for models and protocols? Network management approaches have evolved as network devices and network deployments have evolved.

The SNMP protocol and SNMP MIBs were designed in the 1980s to allow configuring and monitoring network devices over a network. While SNMP remains an excellent standards-based approach for monitoring networks, shortcomings in its ability to perform transactions and lack of an integrated database concept render SNMP a poor approach for network device configuration. Many network equipment providers have created proprietary solutions based on SQL databases for configuration and used SNMP for network monitoring.

## **NETCONF and YANG**

The most recent significant step in the evolution of network management is the NETCONF protocol and the associated YANG modeling language.

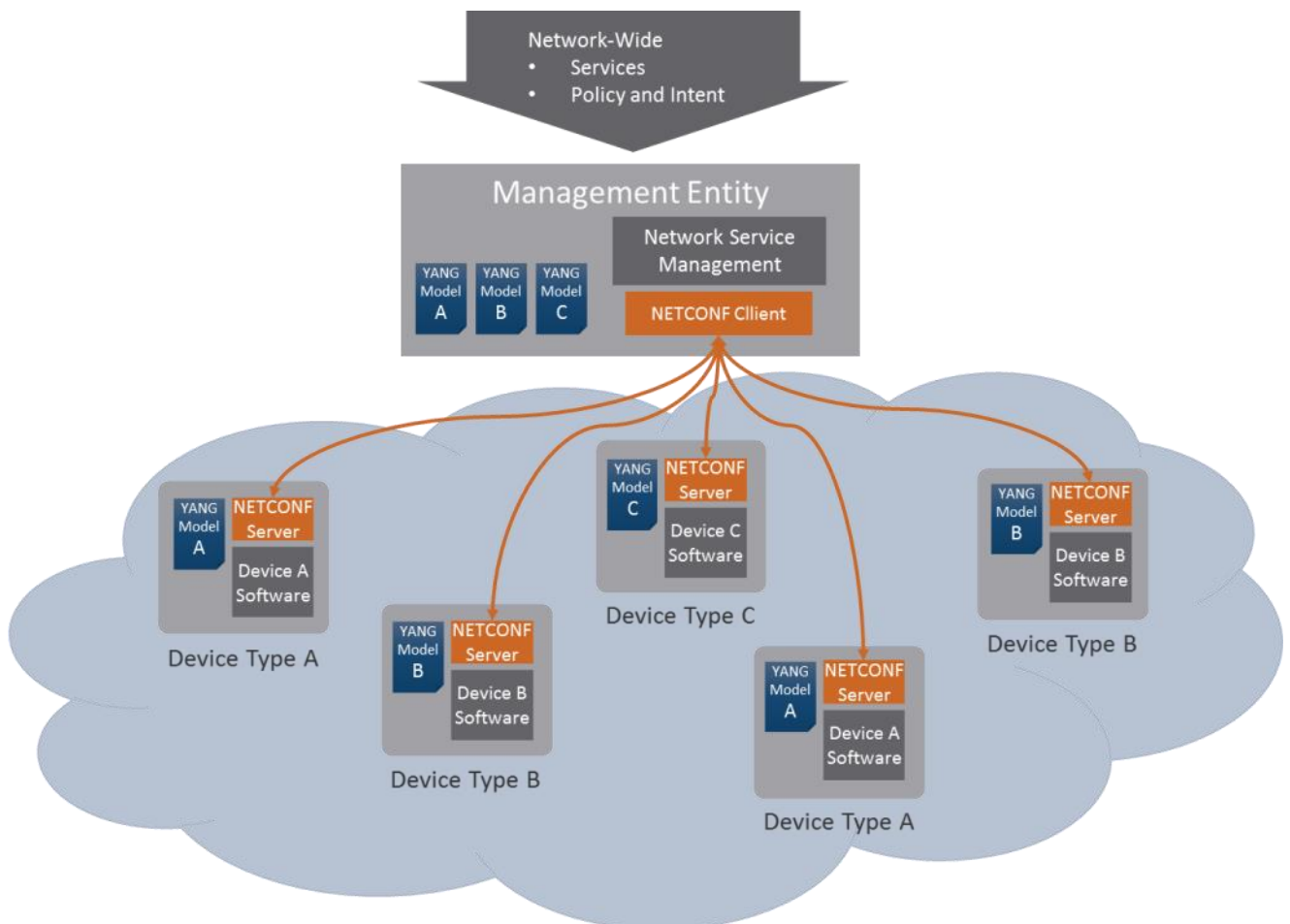
NETCONF was created to provide a robust configuration solution in addition to monitoring capabilities. Transaction support is fundamental in NETCONF, with support for a datastore or database concept to hold configuration information. Security is also handled more directly and simply in NETCONF than in SNMP.

The YANG modeling language was created to provide a standard for defining data models to be used with NETCONF. YANG was designed to be easier for users to write than XML. YANG also includes mechanisms to describe model-defined constraints.

Users write YANG models to define the configuration data, state data, notification events, and actions for devices. NETCONF uses the YANG models to govern the content of the conversation between a NETCONF client (e.g. a managing entity) and a NETCONF server (e.g. a managed device). The NETCONF client and the NETCONF server use the data exchanged in accordance with the YANG model.

### A Clean Approach to Using NETCONF/YANG for Network Management

Once one has chosen to use NETCONF and YANG for the managing network devices, there are some basic guidelines for moving forward.



The first step in using NETCONF/YANG is to model the managed device. This may include one model that describes the entire device or several models that describe different aspects or modules in the device. Whenever possible it is advisable to use standard YANG models.

A managed device should include a NETCONF Server with the associated configuration datastores (running, candidate, and startup) to support configuration transactions and properly form model-compliant responses. The managing entity (NMS, EMS, Controller, Remote Shell, etc.) should include a NETCONF Client to properly form model-compliant requests.

### **Benefits of Clean Living**

What are the benefits of using a complete NETCONF/YANG solution for device management?

The primary benefits of a NETCONF/YANG device management approach are programmability and interoperability. The structured description of all managed objects in the device through the YANG model and the structured set of methods for managed object exchange provided by NETCONF are the building blocks for programmability and automation. Using NETCONF/YANG standards provides interoperability leading to flexibility in the choices of devices and device types and management solutions. The ability to use standard YANG models can save development time as well as ensure that devices that share a common feature set can present the same set of managed objects, simplifying a managing entity's challenge when managing multi-vendor networks.

A NETCONF/YANG solution provides high availability benefits. The YANG model is a contract that defines how the managing entity configures and monitors the device as well as how the device responds, reducing mistakes due to misunderstanding. Validation of configuration changes against the model can prevent malformed/inconsistent changes from being applied. Validation catches/prevents mistakes. Further validation can be applied within the device based on current state of the device. Using robust transactions eliminates the inconsistent outcomes due to some changes working and some not working within a given set of changes. For completed transactions that result in unexpected behavior such as instability or degraded service, a clean transaction rollback can quickly and safely restore stability and availability.

Authentication, Authorization, and Auditing/Accounting (AAA) is a valuable addition to a NETCONF/YANG device management solution. Authentication ensures that the user (machine or human) is who they say they are. This is typically done using a user/password pair or a security key pair (public/private exchange). Authorization ensures that an authenticated user (yes, it is Bill whose role is a provisioner) can only do the things that the user's role is authorized to do (e.g. add new users) and not the things that he is not authorized to do (e.g. reboot the device). Lastly, auditing and accounting maintains an audit trail of activity over the management interface(s) and statistics related to things like user authentication attempts, both successful and unsuccessful.

### **What to Look For in a NETCONF/YANG Offering**

What is the best way forward with NETCONF/YANG? Are there alternative solutions, and how does one evaluate various offerings?

There are a number of ways to add NETCONF/YANG capability to a device, ranging from proprietary in-house solutions to commercial solutions to open source solutions. If there weren't



options, it would be hard to argue that the NETCONF/YANG standard is truly an open standard shared and used by many rather than a reference document for a proprietary offering.

Table 1 summarizes the most important considerations when selecting a NETCONF/YANG management interface software.

Key Areas to Consider When Selecting a NETCONF/YANG solution		Categories
Product Features	Ensure that the offering contains the necessary set of features to provide a standard and secure solution that interfaces both with Management clients and the software within the device.	NETCONF
		YANG
		Management Interfaces
		Device Software APIs
		AAA
Product Core	A complete offering minimizes development time, and buildable source code simplifies development and maintenance. A scalable and partitioned architecture simplifies product evolution.	Complete Solution
		Architecture
		Buildable Source Code
		Scalability
Support	Education accelerates onboarding, and good support drastically reduces risk during development and maintenance after deployment.	Documentation
		Training
		Support
Vendor Profile	An independent software vendor with experience in network software and devices is ideal. Company stability and a commitment to the product and market ensures longevity.	Independence
		Domain Experience
		Complementary Portfolio
		Stability

Table 1.

### Product Features

- **Adherence to Standards** - Adherence to the NETCONF and YANG standards is paramount.
  - **NETCONF** - Interoperability between management NETCONF clients and device NETCONF servers depends on adherence to the NETCONF standard. Interoperability between managing entity and managed device depends on each side fully and correctly interpreting the YANG model describing the managed objects.
  - **YANG** - Tools and runtime support should consistently interpret the YANG model according to the YANG specification. This starts with the ability to consume standard YANG models, which use many features of the YANG specification. Making the YANG models truly the “source code” by reducing occurrences of developers “reading the model” and writing code based on the model reduces the opportunity for mistakes as well as code rewrites required by model updates...or failure to update code required by model updates.
- **Datastore** - The solution must provide a datastore within the device that meets the expectations of the NETCONF protocol, including storage of startup, running, and candidate versions of the configuration as well as support for complete and consistent transactions and rollback of transactions. Further, the Southbound interfaces to software within the managed device should accommodate both performing transactions as well as rolling back transactions.

- **Validation** - Support for validation based on model consistency as well as explicit constraints specified in the model is required. Support for additional validation performed based on run-time state is highly desirable.
- **Programmable Management Interfaces** - The offering should include management interfaces for NETCONF at a minimum. In addition, support for other machine-oriented interfaces such as SNMP (including YANG to ASN.1 MIB generation) for integration with traditional monitoring tools and REST to support alternate management applications that may not want/need NETCONF as the means to access managed objects.
- **Human Interfaces** - The offering should also include model-driven human interfaces such as a CLI with command completion based on the model as well as the contents of the configuration data store or even run-time state. When the model changes the CLI should change behavior accordingly. Ideally, the CLI should interpret the model at run time, allowing model changes to be handled on the fly.
- **Device Software Interfaces** - An on-device management solution will also need to interface with the device software that consumes the configuration managed objects as well provides the state data managed objects. When developing a new device, this is one of the areas of largest impact on the software development team for the device, aside from defining the YANG models.
- **AAA (Authentication, Authorization, Accounting/Auditing)** - AAA is also a firm requirement. The very nature of device reachability and programmability makes it very easy for malicious software to find systems, probe them, and attempt unwanted operations at high speed. Blocking users through authentication and limiting users' actions according to well-defined access roles reduce the damage possible. In the worst case of a breach of the above, audit trails and accounting data for access can help decipher and quantify what happened.

## Product Core

The characteristics of the software implementation of a management offering go beyond the checklist of features described above. These characteristics can impact the speed of integration of the management offering with a networking device as well as how easily the device can evolve over time, both in feature set and device scale.

- **Completeness** - The solution should be a complete solution, including modeling tools, build tools and runtime tools. The solution should include simple integration with needed platform services like SSH for NETCONF and filesystem support for datastore persistence.
- **Architecture** - The software architecture of the solution should decouple "Northbound" management interfaces from the "Southbound" on-device software interface, avoiding the traditional "stovepipe architecture" in which, for example, adding an SNMP management agent requires adding SNMP support into the underlying device software. The architecture should support distributed on-device software ranging from multiple memory protected processes on a single CPU complex to software running across a distributed system on multiple OS instances. Further the architecture should support running on virtual machines and/or containers. Finally, the architecture should support high availability, including the ability for a failed application to restart and reconnect to the management services,

support for redundant datastores with replicated transactions, and support for software upgrade including new YANG models that might include new objects, remove old objects, or modify existing objects.

- **Buildable Source Code** - The availability of buildable source code can greatly improve both the development cycle of a product and the maintenance cycle after product release into the field. Visibility for debugging and troubleshooting is improved with source code access. Source code transparency makes code auditing simpler for copyright compliance and security vulnerabilities. Patching code for debug and maintenance is simpler as is applying security updates.
- **Scalability** - The solution should be scalable to small or large systems. The ability to support small footprint and resource constrained systems is a plus for edge devices and managed “things.” The ability to scale to large configuration datastores and support extremely large sets of state data may be a requirement for many solutions that work at cloud scale.

### Support

The support that either an open source community or a commercial vendor supplies can provide time savings and cost savings and is often critical to development completion, deployment success, and maintenance survival in the field once deployed. This starts with clear, complete, and accurate documentation of the on-device management solution and progresses to training for both developers and support teams associated with the device. Finally, access to worldwide, 24/7 support for the on-device management solution in accordance with SLAs that guarantee prompt and complete resolution of issues is required of a commercial solution.

### Vendor Profile

The business model of a NETCONF/YANG solution is important to consider. Cost structure is important, including the initial cost and recurring costs for maintenance and licensing. The amount of load on engineering resources should also be considered. Developing the solution in-house can be a big drain, and the often-required involvement in an open source solution can add up as well. Risks to be aware of include the vendor’s economic viability and stability. Choosing an independent software vendor should reduce the risk of conflicts of interest and increase predictability of feature and product availability and product support.

### Summary

A NETCONF/YANG solution for network management enables automation by making network devices programmable. Choosing the right offering should involve a balanced big picture evaluation. A turnkey solution that includes all the necessary features for programmability and interoperability helps time to market and initial deployment. The right product foundation helps the device’s evolution and allows scaling the device family up or down. Dependable and comprehensive support provides smooth sailing over the deployment lifetime. Lastly, a viable and stable vendor that maintains an ongoing commitment to your success will ensure the availability of product enhancements and support for years to come.





5th Layer Consulting provides professional services in the areas of software architecture, network architecture, and cloud infrastructure. Joe Kidder, Principal at 5th Layer Consulting, has over 30 years' experience in software and system architecture, design and development. He has developed switches and routers for Wellfleet Communications, Nortel Networks and startup Equipe Communications, and holds 13 patents related to system architecture, high availability software, and quality of service.

For more information: [5thlayer.com](http://5thlayer.com)



Enea develops the software foundation for the connected society. We provide solutions for mobile traffic optimization, subscriber data management, network virtualization, traffic classification, embedded operating systems, and professional services. Solution vendors, systems integrators, and service providers use Enea to create new world-leading networking products and services. More than 3 billion people around the globe already rely on Enea technologies in their daily lives.

For more information: [www.enea.com/NETCONF](http://www.enea.com/NETCONF)