

# Programmability for network automation

Supporting modern workflows while evolving to an intent-based networking system and DevOps

Application note

## Abstract

With a goal of ensuring closer alignment between networking infrastructure workflows and business initiatives, [Nokia has evolved our Network Services Platform \(NSP\) into an intent-based networking system \(IBNS\)](#).

Programmable workflows continue to be important for achieving the most flexible and fine-grained control when developing network automation for full life-cycle management of network infrastructure, including design, implementation, operation and assurance. NSP's programmable network automation leverages its IBN framework to provide a new way to build and operate networks. NSP enables workflows to meet the goals of IBN such as validation of a business intent and the corresponding network configurations. And intent-based policies are enforced to dynamically take real-time action if the intent and the network are out of sync.

This paper focuses on how NSP workflow management being built within an IBN framework is making it easier to develop, maintain and execute on end-to-end operational workflows. It outlines key workflow management use cases that apply to the entire operational life cycle and enable workflows to be developed. Those workflows improve network capabilities, and also enable better IT/network practices to have more flexibility, agility and efficiency for innovation, delivery and operations.

## Contents

Abstract	2
The evolution of workflow management in network operations	4
Network operations challenges in evolving IT/network practices – and a solution	4
Bringing back control of network workflows to network architects	7
The NSP Workflow Manager (WFM)	7
Key NSP WFM use cases	8
Activation testing	8
Software upgrades	9
Other use cases	10
Abbreviations	10
Related resources	11
Appendix: NSP Workflow Manager – product details	11



## The evolution of workflow management in network operations

Traditionally end-to-end operational workflows have been mostly in the scope of network service orchestration and operations support systems (OSS) rather than within network domain controllers and network management systems. This was until recently with the move toward DevOps, the emergence of intent-based networking (IBN) modeling and their abstracted APIs, and, in addition, open-source software becoming more mainstream among network architects.

The evolution of workflow management with IBN has enabled network architects and engineers to leverage their network technology expertise to define domain-specific workflows. The result is a more collaborative approach that puts the ownership of development and maintenance for specific workflows within the most appropriate groups. In other words, it gives contributors the autonomy and focus to be able to independently work on the more challenging implementation tasks (i.e., the tasks that most require their expertise), while making the interworking between the network groups and their stakeholders simpler.

## Network operations challenges in evolving IT/network practices – and a solution

If we consider today's network operations, many follow pre-set and well-practiced workflows that are specific to the network domain technologies in use; these will typically be defined in a method of procedure (MOP). These MOPs can cover all sorts of aspects such as service fulfilment/migrations, initial device commissioning and subsequent configuration, alarm/fault handling, and assurance aspects as well as nodal backup/restore and upgrades. MOPs, although perfect when they are first written, are useful if nothing changes. However, these MOPs do become a burden to maintain. They are only updated when there is a compelling event to justify the hours of investment and distraction from critical daily priorities.

In addition, using the typical approach to execute a deployment can also present many logistical challenges. For example, the delivery of a new service rollout on the network can be a very high-touch and time-consuming operation. It typically involves multiple users coordinating and planning across multiple internal teams and organizations, each team with their own set of tools to perform their task(s). The execution process typically takes hours, sometimes days. Also, various IT tools are required for task tracking and status. These include operational investments in existing non-integrated workflows, scripts, CLI and tools. For successful adoption of an [intent-based networking system \(IBNS\)](#), these existing tools and workflows must be able to be built in and supported by the IBNS.

As a programmable, automation platform, the Nokia Network Services Platform (NSP) allows users to integrate “standard” procedures into workflows that can be automated for increased efficiency, such as by providing low-touch operations to reduce human errors, as well as faster service rollout to customers with consistent task tracking and audit capabilities.

For example, with a centralized and integrated workflow automation engine, NSP allows each user group/organization to program actions required for each task. Each team owns their task scripts, which are hot-swappable, and can be updated when new devices, configurations or processes are introduced (transparent to the service fulfilment and operations teams). Service operators can execute workflows that include a sequence of tasks that have been programmed to satisfy each organization's demands. If a task fails (due to deployment error or activation test failure), the workflow can automatically create

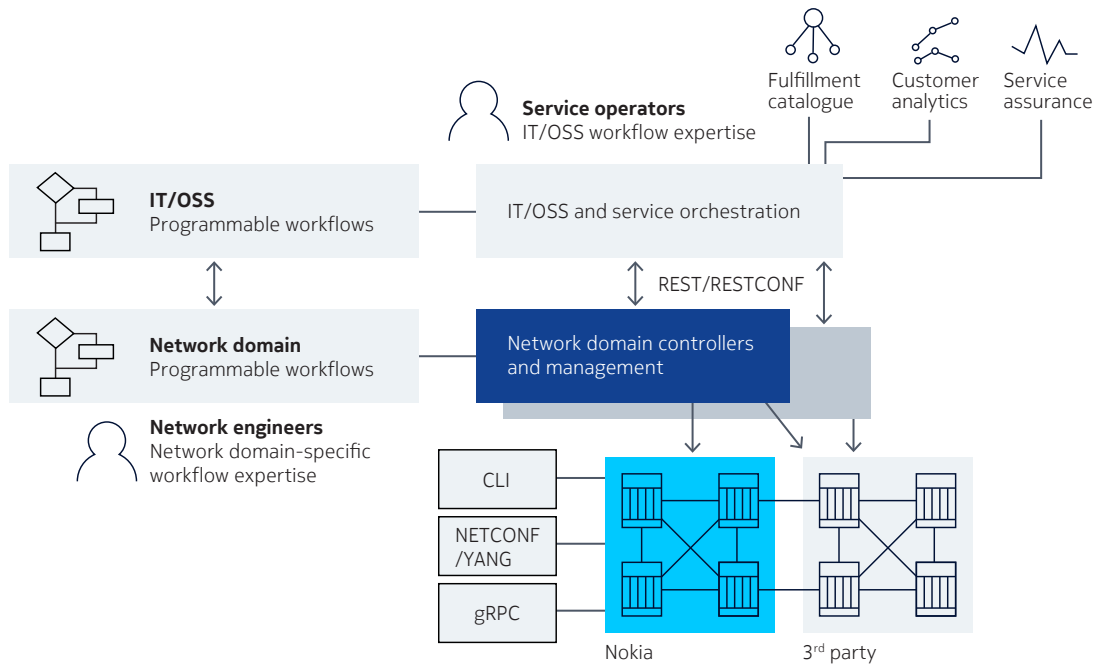
a trouble ticket with the appropriate info (network elements, configurations, OAM test results, etc.). The service operator can then review the ticket and is able to assign it to the appropriate user group to debug/troubleshoot. This is a well-defined, low-touch approach that simplifies operations.

If we consider the present mode of workflow operations that are defined within OSS systems and orchestrators, it requires that network engineers/designers/planners and IT spend time together to define many workflow steps, such as described in multiple MOPs (many times inconsistent across different vendor equipment). These interactions between the OSS and the network all need to be developed and coordinated to achieve the end result. This can mean lots of time and effort spent collaborating, which is easier if the work is done in the same location or with the same project stakeholders; but in many cases end-to-end workflow development is much more complex, involving various stakeholders from different groups—sometimes with very orthogonal viewpoints and goals.

And even when everything is aligned well between stakeholders, the translation of IT/OSS requirements to engineering still needs to be done. This requires a fusion of expert network knowledge with IT/OSS software know-how. This translation between the two worlds can be difficult, especially when interactions between the OSS and the network are not abstracted and are difficult to explain. It is for these reasons that handling network evolution cases is often held back by IT/OSS development times. It involves overcoming the cultural, technological and operational barriers that historically divide the networking and IT domains. And for implementation, it involves harmonizing disjointed service policies, resolving a mismatch of interworking protocols, and integrating various islands of automation.

Leveraging domain-level subject matter experts to create and streamline operational workflows enables end-to-end and broader OSS-level workflows to be developed faster and more efficiently and allows for them to be continually maintained by the right stakeholders. This separation of IT/OSS workflows from domain-level workflows ultimately allows developers to focus their efforts on the areas where they have the most expertise (see Figure 1). With network architects and engineers focusing on domain-level workflows, more time is opened up for IT/OSS workflow developers to focus on the integration of operational and business-level requirements. The workflow integration into the domain layers has been abstracted and simplified through enabling calls to domain-level workflows that have already been developed and are maintained by the appropriate experts. These experts are best suited to develop workflows because of their inherent knowledge of how to apply network vendor implementations of NETCONF/YANG and gRPC. In many cases these interfaces are consistent with CLI implementations. For example, Nokia Service Router Operating System (SR OS) model-driven CLI command structures are actually auto-generated from its YANG specifications to ensure consistency for every release of the SR OS. (For absolute uniformity, all Nokia SR OS interfaces are auto-generated from its YANG models, including for gRPC.)

Figure 1. Workflow development—collaboration between IT/OSS and network domain experts



By allowing network expertise to be built into abstracted network models, the network service orchestrator can then be unhindered while continuing to focus on IT/OSS-level functions such as order capture from customer relationship management, sales automation and customer order management software, as well as order decomposition that inherits the predefined process steps from its technically abstract service catalogue. This model-driven approach also minimizes the need for professional services customization at this upper service fulfilment layer through automation and simplification of the domain-level interactions. The integration with the domain-level API drives the necessary domain commands to the appropriate domain controllers and subsequently each domain's own expert-developed network workflows.

The resourcing benefit is that the IT/OSS workflows would be defined by IT/OSS people who will no longer need to have domain-specific knowledge of the underlay IP network and technologies, since these will be developed by the network engineering team. This enables IT/OSS developers to have minimum service parameters to call for API interactions with the domain controller. It also enables operational staff to self-adapt to a mode of working where network and services can organically evolve without being limited by cumbersome dependencies and interaction between siloed groups. This improved way of working further helps to prevent inefficiencies and human errors that root from individual workflow developers having to work outside of their areas of subject matter expertise.

Even for small operations that are reliant on a small number of personnel, adaptations to end-to-end and cross-software-stack workflows will become easier and more scalable to develop with an approach that ensures consistent delivery and error avoidance across every individual in the team.

## Bringing back control of network workflows to network architects

Typically, a workflow programmed for a network domain's resource controller has two key roles: to capture the abstracted requests from the end-to-end network service orchestrator or OSS layer, and to transpose this into the actual service delivery to the network. The workflow should also try to enforce that the services being activated adhere to the required SLAs, while taking into account what is possible based on available network technologies and topology.

Network engineering would define these workflows so that they, at minimum, need to know the objective of the service catalogue. However, much expertise is also required that includes at least a high level of understanding of what is possible in the network elements (NEs) and deployed transport technologies, as well as what capabilities are available from the network domain's resource controller. This is exactly why we need to bring back control of domain workflows to network architects and engineers.

There are many examples where network domain expertise is fundamental for accurate and efficient implementation. For example, for IP/MPLS networks, one use case would be for enabling (with no change to the orchestrators abstracted requests) premium services that would use latency-based Segment Routing Traffic Engineering (SR-TE) tunnels, including path diversity. This requires significantly more IP domain expertise to add all the extra optimization value of using SR-TE compared with mainstream approaches where Resource Reservation Protocol-Traffic Engineering (RSVP-TE) tunnels are typically over-engineered with excess commissioned bandwidth to meet peak demand.

Another example where IP domain expertise in developing workflows is important is when implementing network service creation using Ethernet VPN (EVPN) instead of E-LAN, for instance to deliver a specific premium service offering with a higher quality of service (QoS), say a "Silver" service. When creating workflows that span from the orchestration/OSS layer to the IP domain controller, modern approaches enable very dynamic automation. For example, creation of new entries in the orchestrator's service catalogue can be designed to automatically result in triggering the IP domain controller to perform closed loop automation, such as for an IP-data center workflow. When this domain-specific workflow is invoked, it could create the required QoS profiles across different vendors' equipment to make the network ready for the first service delivery. This domain-specific workflow may also enable enhanced service level reporting that invokes OAM performance and SLA test suites, not only to benchmark the service at creation time, but also to allow continuous monitoring in the event of changes in the service path or configuration.

At Nokia, we believe that the key to addressing these challenges is through enabling closed loop automation, which can be maintained by network architects and engineers at the domain controller level in order to provide abstraction to the IT/OSS orchestration workflow layer.

## The NSP Workflow Manager (WFM)

The NSP Workflow Manager (WFM) is used for programmable network and service automation. The WFM enables orchestration across all NSP-managed NEs, NSP's operating system infrastructure and applications, and external systems. Its programmable automation enables fine-grained control for a wide scope of operational responsibilities, including network migration, configuration management, performance testing, security management and software management. Workflows can be initiated manually, scheduled, or triggered by network events or as side effects of other NSP operations. Existing workflows can be easily adjusted to adapt to customer-specific procedures and to support new NE releases, including third-party vendor equipment.

## Customer ROI and key benefits

(as reported by NSP Tier 1 customer in Asia-Pacific region)

- **Save time:** Run more than twice as many upgrades in the same (limited) maintenance window
- **Improve OPEX usage:** Release qualified engineers for other activities
- **Increase reliability:** The upgrade is more predictable. The service reliability increases as your network evolves in a predictable and consistent manner.

## Qualitative values identified:

- **Customizable:** Workflows (scripts) can be added/modified to match specific needs. No hard coding (unlike classical product features). Avoid the need to upgrade/restart NSP.
- **Integrated:** Solution is part of NSP and shares its carrier-grade principles: redundant, scalable, authentication/logging (future planning: scale-on-demand, health monitoring, authorization). Workflows can be actively executed by NSP apps and can be triggered by NSP Kafka messages. WFM can consume all NSP APIs without the need to care about authentication tokens (get/refresh/revoke). WFM can talk to NEs (CLI or NETCONF) leveraging known authentication credentials (mediation policies) and CLI profiles.
- **Supported:** Nokia professional services experts are fully behind the WFM and can help support when needed. If the workflow developer in your team is suddenly unavailable and there is an important script to be developed or maintained, then you know you can still rely on the Nokia professional services team to support this activity, and to be able to customize and integrate the solution in your environment (no matter how custom or complex).

## Key NSP WFM use cases

### Activation testing

Activation testing is a use case within end-to-end service life-cycle automation. Creation and execution of OAM diagnostics testing is used to validate SLAs as the last step in the service activation process.

Example use case:

**Preparation:** Service operator executes a service order request using the OSS portal. Automated workflow engine starts task execution.

**Task 1:** Commission device/ports – configuration applied to NEs to make ports “service-ready”

**Task 2:** Service fulfillment request is triggered. Service configurations are applied to each NE based on NE/service-specific templates.

**Task3:** Automatically execute OAM tests to validate SLA requirements. If test(s) pass, proceed with service activation. If tests fail, notify operator and rollback service configurations on each NE.

**Task 4:** If activation tests pass, set service to Admin UP, and enable statistics for performance monitoring on each service endpoint for a predefined period (to be specified as completing after a fixed duration or a specific calendared date/time).

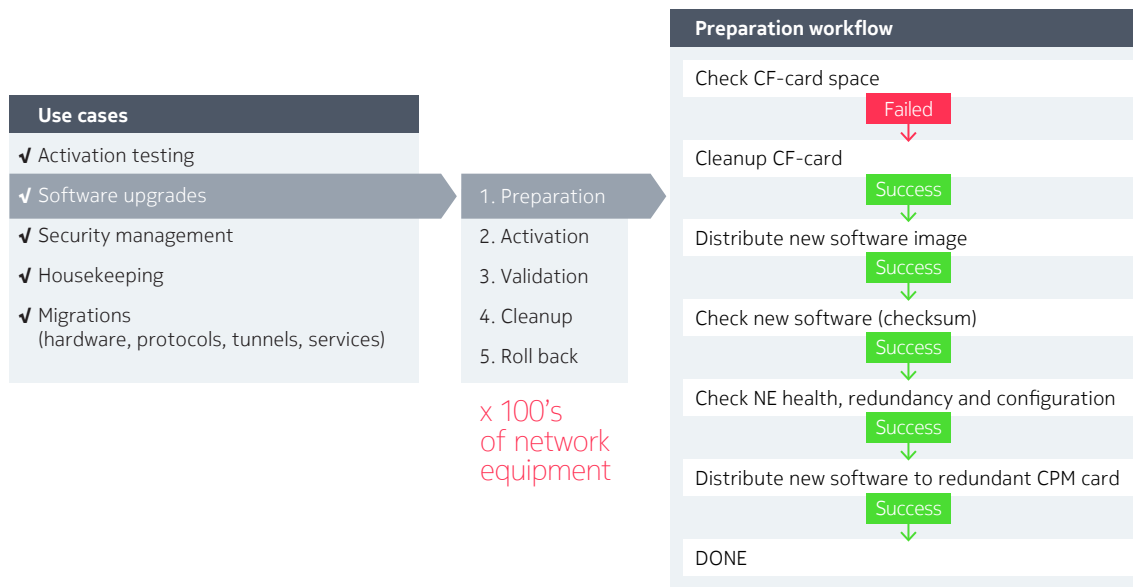


## Software upgrades

To upgrade the software of network equipment, every operator goes through a number of steps, and they repeat these steps for every piece of equipment in the network—and there can be hundreds. In scenarios like for a mobile backhaul network, this can easily scale to tens of thousands of NEs—all while operators typically have constraints on how fast new releases need to be rolled out, and while the number of maintenance windows and operators is limited.

Often the actual recipe (MOP) is specific to specific vendor NEs and not their broader intent, while having a workflow allows the flexibility to adjust to the operator’s broader need. For example, there are specific steps operators typically go through when they want to check that an NE is ready for a software upgrade. In Figure 2 we label this as the “Preparation” phase. For example, in many IP/MPLS networks today, MOPs many times only perform parts of the preparation and validation steps, and are typically designed to support only a specific vendor’s equipment (i.e., requiring separate MOPs for each vendor). The flexibility that NSP workflows bring is especially valuable in comparison to traditional approaches since it makes it possible to automate more steps and seamlessly support third-party vendors.

Figure 2. Example of a software upgrade use case



As in Figure 2, when the validation phase is a success, the next step is cleanup (remove previous version of the software). However, if the cleanup for step 4 fails, then the next step is roll back to the initial state.

Automation of software upgrades is not just important to reduce time-to-market (for new NE functionality) and to reduce operational cost. It also helps customers to install security fixes faster preventing hackers from using known vulnerabilities for their attacks. It helps to introduce newer hardware items faster – which typically helps to reduce the cost (e.g. USD/100 Gb/s) and keeps the installed base more up-to-date (e.g., enabling upgrade to hardware that offers savings for power/cooling consumption).

## Other use cases

### Security management

- Distribute and update passwords, pre-shared keys (PSKs), security profiles, access control lists (ACLs) and certificates network-wide
- Some coordination might be required (e.g., updating PSK on both ends of an IP connection or using key chains if applicable)

### Housekeeping/maintenance

- Tasks to clean up things that are no longer needed
- Removing old files from NE compact flash is an example.

### Migrations

- Hardware: Many kinds of port moves, card migrations, chassis migrations
- Protocol: For example, to move from OSPF to IS-IS
- Tunnels: For example, to change service tunnel technology from LDP to Segment Routing
- Services: For example, to move from T-LDP-based E-LINE to EVPN E-LINE, or moving service endpoints to different ports/chassis

### Elastic link aggregation

- Dynamically resize link(s) for bandwidth-on-demand use cases.

### Flexible execution

- Workflows can be executed by the user or north-bound interface, scheduled (once/periodically), actively called by other NSP apps or automatically triggered by NSP Kafka events. This enables much more upcoming use cases from those listed above, such as automatic troubleshooting to zoom in on failures.

## Abbreviations

ACL	access control list
API	application programming interface
CLI	command line interface
EVPN	Ethernet VPN
IBN	intent-based networking
IBNS	intent-based networking system
IS-IS	Intermediate System-to-Intermediate System
MOP	method of procedure
NE	network element
NSP	Network Services Platform
OAM	operations, administration and maintenance
OPEX	operating expenses

OSPF	Open Shortest Path First
OSS	operations support system
PSK	pre-shared key
QoS	quality of service
ROI	return on investment
RSVP-TE	Resource Reservation Protocol-Traffic Engineering
SLA	service level agreement
SR OS	Service Router Operating System
SR-TE	Segment Routing-Traffic Engineering
T-LDP	Targeted Label Distribution Protocol
WFM	Workflow Manager

## Related resources

- [Demo: “Network and service automation with a workflow manager”](#)
- [Application note: “Network automation and programmability: Using the Nokia NSP for intent-based networking \(IBN\)”](#)
- [Data sheet: Network Services Platform](#)

## Appendix: NSP Workflow Manager – product details

Embedded in NSP is a Workflow Manager (WFM) that uses a standards-based programming language with the ability to listen to the status of the network and invoke engineering-defined workflows as required.

WFM technologies for programmable automation include:

- Mistral Domain Specific Language (DSL)
- YAML
- YAQL
- Jinja2
- JavaScript
- REST API

For example, the NSP WFM enables network engineers to define the workflows in YAML using the OpenStack Mistral DSL v2 specification. This has a rich set of functions that are learned very quickly as the NSP will include a wide range of different code snippets and examples, freely available within NSP’s DevOps portal, which engineers can use out-of-the-box or modify for their own purposes. The [Network Developer Portal](#) provides many NSP reference use cases, implementation examples, documentation, SDK tools, and remote cloud labs to sand-box development.

The NSP WFM will allow the ability to import/export workflow definitions (YAML) from/to the operator’s file system to create new workflows; they also can be shared through the NSP DevOps portal, therefore creating an open community that aids workflow development.



To provide flexibility, workflows can be invoked on-demand or scheduled. This can be controlled through the user interface or through the REST API. Visibility is greatly enhanced through the user interface to provide a representation of the workflow defined in YAML in a graphical flow chart (OpenStack Mistral workflow visualization). This workflow visualization can be used by engineers to articulate design intent and execution paths easily to others.

The WFM dashboard provides vital statistics on each workflow to allow engineers to isolate how many times a workflow has successfully been executed or has failed. It also provides more detailed information and reasons for why a workflow failed. For developing workflows, this adds value by enabling quick adjustment during the implementation process.

To aid closed loop assurance, the WFM is able to subscribe to events that are being published on the NSP north-bound API's Kafka bus. The events on the Kafka bus could include network and service correlated alarms, threshold crossing alarms (TCAs) on performance statistics or even triggers notifying that new objects (e.g., new NE, new service) have been created in the network. For example, if a TCA has been raised for packet loss on an interface, a workflow could be triggered to automatically execute. This workflow could then invoke OAM tests over that interface, and on the OAM results, potentially decide to automatically perform a specific series of actions such as:

- Moving specific services from this interface, and/or
- Placing the interface into maintenance mode, and/or
- Allowing the NSP's stateful Path Computation Engine (PCE) to invoke optimization.

There are many examples of how a flow could be automatically invoked and what that workflow will do. In addition, the WFM can also interact with aspects outside the NSP as a domain controller, such as with IT/OSS orchestration, which helps provide a greater scope of value for the network engineering efforts.

## About Nokia

We create the technology to connect the world. Powered by the research and innovation of Nokia Bell Labs, we serve communications service providers, governments, large enterprises and consumers, with the industry's most complete, end-to-end portfolio of products, services and licensing.

From the enabling infrastructure for 5G and the Internet of Things, to emerging applications in digital health, we are shaping the future of technology to transform the human experience. [networks.nokia.com](https://networks.nokia.com)

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.

© 2019 Nokia

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

Document code: SR1909038524EN (October) CID206795