# NED – way of working

NSO BE
Cisco Networking

v.1.1 June 2023

# NED Development Best Practices

## Key Highlights

**NSO Versioning**

- Each NED release is compiled and made available for each active NSO branch (minor version). NEDs are generally compatible across their respective NSO branch with exceptions listed in the NED meta-data.

**Trunk Based Development**

- All NEDs use Trunk Based Development, i.e. new NED releases are created from the tip of a single branch. New fixes are delivered to the customer in the latest NED release, not by augmenting an old release.

**Device Expertise**

- The NED team has a lot of YANG modeling expertise, but limited device expertise. When adding to a YANG model, the more details provided about configuration and use case, the better.
- New configurations will be added to the NED regression test suite. The more configuration files delivered to the NED team when requesting development, the better the result.

**Use of NETSIM**

- Using NETSIM will not reveal all problems in the model. Therefore, project teams are recommended to regularly test on real devices to get an early detection of required NED updates.

# NED Development Best Practices

## Key Highlights

**Device Access**

- The NED 3rd party device lab does not host all devices and configurations supported in the NEDs. Customer device access is often required to fulfill customer expectations.
- To leverage on the NED team automated test approach and ensuring good quality – direct device access is needed (either via the NED labs, or when necessary; customer's device). Hence development can't happen over WebEx, remote desktop, or similar.

**NED scope**

- The NED scope does not exhaustively cover the device API/CLI.
- Current NED scope is based on accumulated customer requests.
- NED development is driven reactively, meaning that new scope is added on demand.
- Delivery time of new features is directly dependent on the magnitude of the requirements, so an application-focused scope is recommended.

- What is a NED
- What a NED provides
- What a NED doesn’t contain

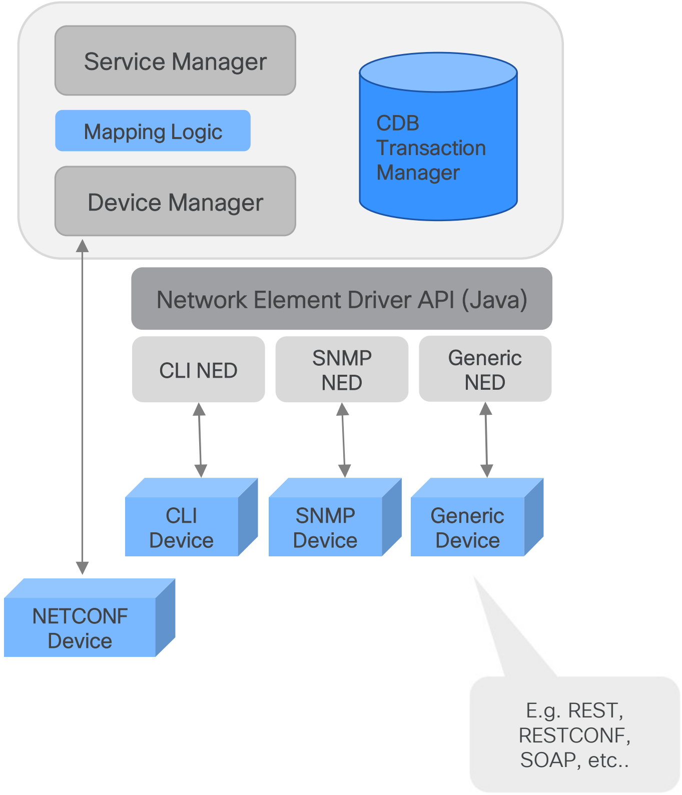
# What is a NED

## What is a NED?

Cisco® Network Services Orchestrator (NSO) provides a single pane of glass for orchestrating a multivendor network. To offer support for an exceptional range of multivendor devices, it uses Network Element Drivers (NEDs).

A NED is a unit of software that provides connectivity and data transfer between NSO and Network Elements. The NED is the adaptation layer between the XML representation of the network configuration contained inside NSO, and the wire protocol between NSO and managed devices.

Network element drivers comprise the network-facing part of NSO. They communicate over the native protocol supported by the device, such as CLI, REST, RESTCONF, NETCONF, SNMP, etc.



# What a NED provides

1 A YANG data model of the device to NSO and services

2 Translation of data changes in the model to device language
- CLI (vendor specific)
- Generic (REST, RESTCONF, SOAP, etc.)
- NETCONF

3 All data modifications in a single transaction

4 A transaction is either completely successful, or failing

# What a NED doesn’t contain

## Device data model

| Topic | A NED does *not* contain | Description |
|---|---|---|
| Device Data Model | • A data model of the entire set in the data | Providing a 100% complete YANG model for a device is not in the scope for NED development. |
| | • An exact copy of the syntax in the device CLI | NED development focuses on representing device data for NSO. As a side effect for CLI NEDs, the NSO CLI will get similar behavior as the device CLI, however this is not the goal. |
| | • Fine-grained validation of data – flexible models are preferred | Adding validations in the YANG model (e.g. mandatory, when, must, ranges, max, min, etc.) leads to inflexible models. When newer versions of the device is released, these validations often needs to be revisited. The policy is to avoid validations in the NEDs, allowing more flexible models. |

# What a NED doesn’t contain

## Auto-Configuration

| Topic | A NED does *not* contain | Description |
|---|---|---|
| Auto-Configuration | • Convenience macros in the device CLI - only discrete data leaves are supported | Some devices have macro-style functionality in the CLI and users may find it annoying that these are not available in NEDs. The convenience macros have proven very dynamic in the parameters they change, causing frequent out-of-sync situations. |
| | • Dynamic configuration in devices - only data in a transaction may change | Dynamic behavior often changes between device versions resulting in inflexible models. If required, the service code can provision the transaction with the extra configuration the device is adding. |
| | • Auto-correction of parameters with multiple syntaxes - use canonical representation | The NED doesn’t allow the same value for a parameter to have a different name, the canonical name displayed in “show-running-config” or similar is used. |
| | • Handling out-of-band changes - model as operational data | Leaves that have out-of-band changes will cause NSO and the device to become out-of-sync, and should be made "config false", or not be part of the model at all. Similarly, actions that cause out-of-band changes are not supported. |

# What a NED doesn’t contain

## Operational & Version Management

| Topic | A NED does *not* contain | Description |
|---|---|---|
| Operational | • Splitting a single transactions into several sub-transactions | For devices that support the transaction paradigm, the NED will never split an NSO transaction in two or more device transactions. The service must handle this by doing multiple NSO transactions. |
| Version Management | • Back-porting of fixes to old NED releases - trunk based development is used | All NEDs use Trunk Based Development, i.e. new NED releases are created from the tip of a single branch, develop. New features and fixes are thus delivered to the stakeholders in the latest NED release, not by backporting an old release. |

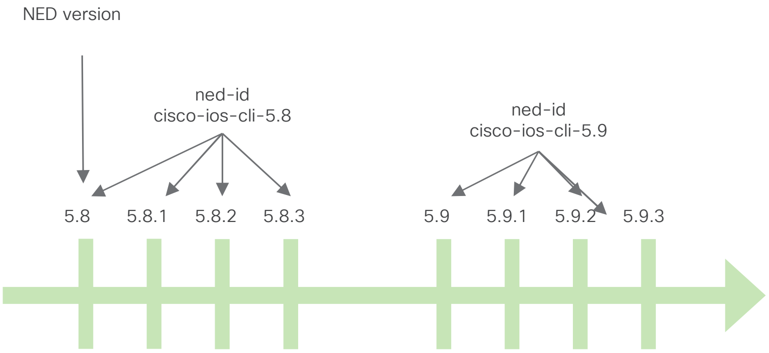
- Version Management
- Upgrade and Migration

# Version Management

## NED releases

### NED releases

- NED engineering uses trunk-based development, i.e. all development is done on a single branch. All NED releases are created from this single branch.
- Most NED releases are compatible with the previous release, in reference to RFC 7950. The new compatible release will have the same  major and minor version as the previous one, e.g. 5.8.1 and 5.8.2.
- Some NED releases are incompatible with the previous release, which is indicated by a new major or minor version, e.g. 5.8.3 and 5.9. Great effort is spent in avoiding this, but it may still be required, e.g. when new members in a device family are introduced.

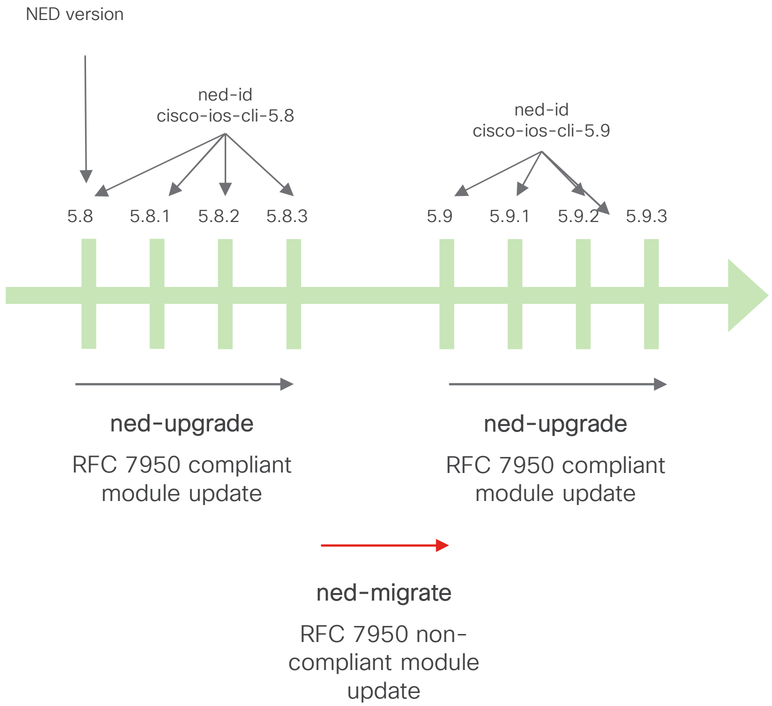


# Version Management

## When a NED upgrade is done and when NED migration is needed

### Upgrade and Migration

- Updating an existing NED version to a compatible version (same major and minor) is called an upgrade. This is done with a reload operation: "packages reload", and will not affect the NSO database.
- Updating an existing NED version to an incompatible version  (different major or minor) is called a migration. This is done with the migrate operation: "devices migrate", and will impact the NSO  database.
- There is tooling to determine how the NSO database will be affected by the incompatible model change. This can be performed in advance, without doing the actual migration.



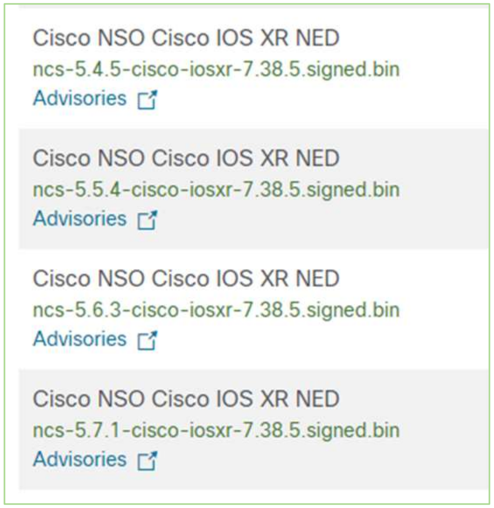
# Version Management

## NSO versions

### NSO versions

- The same NED version is released for all active NSO versions, so a single NED release will comprise several deliverables.
- A NED for a specific major and minor NSO version is compatible with all NSO versions with the same major and minor. Hence, a NED for NSO 5.7.2 can also be used with NSO 5.7.1.
- Some NEDs may require a minimum version of NSO to operate correctly. This is indicated in the NED meta-data, and NSO will give an error if there is a mismatch in versions.

NED version 7.38.5 applicable for all NSO minor versions: 5.4, 5.5, 5.6 and 5.7



Cisco NSO Cisco IOS XR NED
ncs-5.4.5-cisco-iosxr-7.38.5.signed.bin
Advisories

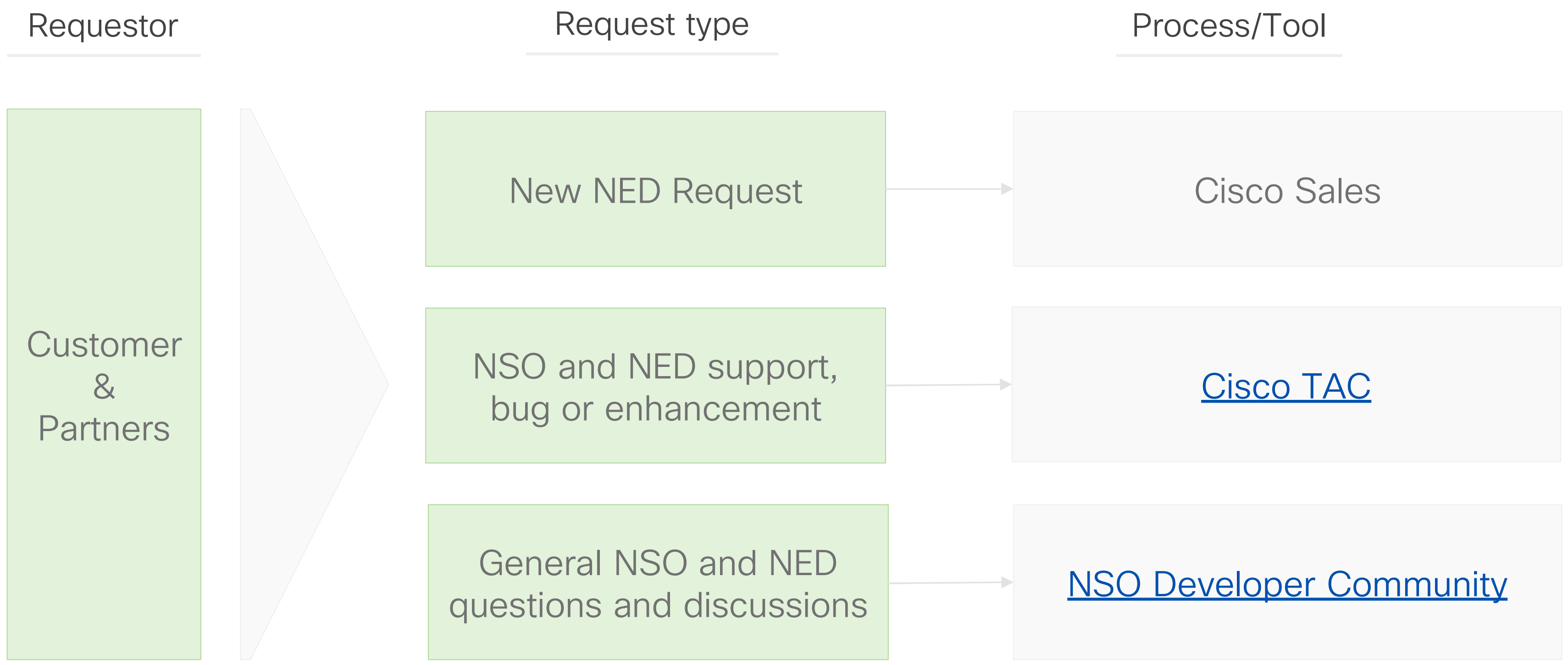
Cisco NSO Cisco IOS XR NED
ncs-5.5.4-cisco-iosxr-7.38.5.signed.bin
Advisories

Cisco NSO Cisco IOS XR NED
ncs-5.6.3-cisco-iosxr-7.38.5.signed.bin
Advisories

Cisco NSO Cisco IOS XR NED
ncs-5.7.1-cisco-iosxr-7.38.5.signed.bin
Advisories

- NED Support Process

# NED Support Process



Requestor

Request type

Process/Tool

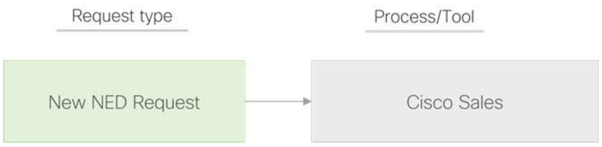
Customer & Partners

New NED Request → Cisco Sales

NSO and NED support, bug or enhancement → Cisco TAC

General NSO and NED questions and discussions → NSO Developer Community

# New NED request


Request type

Process/Tool

New NED Request → Cisco Sales

## Purpose

- For the Cisco NED team to create a new NED, a range of input requirements must be met. The requirements are essential for developing a NED, hence mandatory and must be available during the entire NED creation.
- A new NED request is initiated by contacting your Cisco Sales contact.

## Disclaimer

- The NED development don't start until all input requirements are met. In case of changes to the requirements during the development, the expected NED delivery will risk being delayed and might impact the greater project.

## Expected outcome

- After receiving input requirements, the NED team will estimate the amount of work and include it in the new NED priority list.

## Mandatory input requirements

### NED target delivery date

- For prioritization and aligning with the NED development time plan.

### NSO version

- Target NSO version for the project. This is for the NED team to have an initial release to support. Future releases of NSO (major/minor) will also be supported.

### Device model/OS-version

- Detailed information about the device model and OS-version to be covered by the NED, i.e., show version from CLI or equivalent.

### Sample configuration for project use-cases/services

- The configurations should be equivalent to what is expected to be supported by the NED. These samples will be added to the test system as part of the regression tests.
- Preferably, the NED team should get the exact command sequences to be expected for all different use cases. A single *show running-config* or equivalent from a reference system will require more iterations with the project and hence impact the delivery time.

### Technical contact

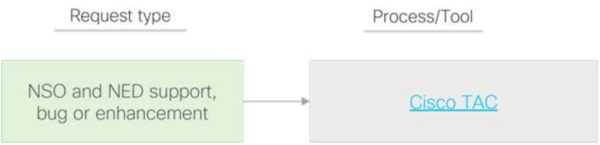
- This person should be able to provide answers to device configuration and management questions.

### API user guide

- Required for the developer to get knowledge about the commands to be implemented.

### Remote access to a target device

- The target device should cover all configuration to be supported by the NED. The NED developer must be able to connect to the target device from his/her local machine. VPN/Tunneling is accepted.
- WebEx session or other remote desktops are not acceptable due to the development environment and test system.


Request type

Process/Tool

NSO and NED support, bug or enhancement

Cisco TAC

# NED Support/Bug/Enhancement

## Purpose

- For the Cisco NSO NED team to support and maintain NEDs, a range of input requirements must be met.
- The requirements are essential for the team's ability to fix or enhance a NED. All requests for support and enhancement are initiated by contacting Cisco TAC.

## Disclaimer

- Due to technical constraints, the expected NSO version can sometimes not be supported by the NED. In such cases, the customer/project must upgrade to a newer version of NSO.
- The NSO NED lab effort is to cover as much as possible of the features to be used by a specific device. In some cases, this is not feasible, and remote access is necessary. In these cases, the NED developer must be able to have access to the target device from his/her local machine. The device should cover all configuration to be supported by the NED. VPN/Tunneling is accepted.
- WebEx session or other remote desktops are not acceptable, due to the development environment and test system.

## Expected outcome

- After the mandatory input requirements are fulfilled, the bug/enhancement is added to the NED backlog. All communication with the customer will be via the ticket.

## Mandatory input requirements

### NED & NSO Info

- NSO and NED versions used.

### Device model/OS-version

- Information about the device model and OS-version.

### Technical contact

- This person should be able to provide answers around the management of the device.

### Debug trace and successful example

- A NED trace that shows the issue in question. Preferably all log files, without changes, from the system (mandatory for all defects and for CLI NED enhancements).

### For Enhancements: Short description of missing config and Sample configuration with the missing config set

- The configurations should be equivalent to what is expected to be supported by the NED. These samples will be added to the NED test system as part of the regression tests. Preferably, the NED team should get the exact command sequence to be expected for all different use cases. A single *show running-config* or equivalent from a reference system will require more iterations with the project and hence impact the delivery time.

### Remote access to a target device

- If device is not available in NED lab. Direct access to target device provided by customer is needed.
- The target device should cover all configuration to be supported by the NED. The NED developer must be able to connect to the target device from his/her local machine. VPN/Tunneling is accepted.
- WebEx session or other remote desktops are not acceptable due to the development environment and test system.

For a more detailed instruction, please visit each NED's README