



# NSO as a Tool for NETCONF and YANG Testing

*Wai Tai*

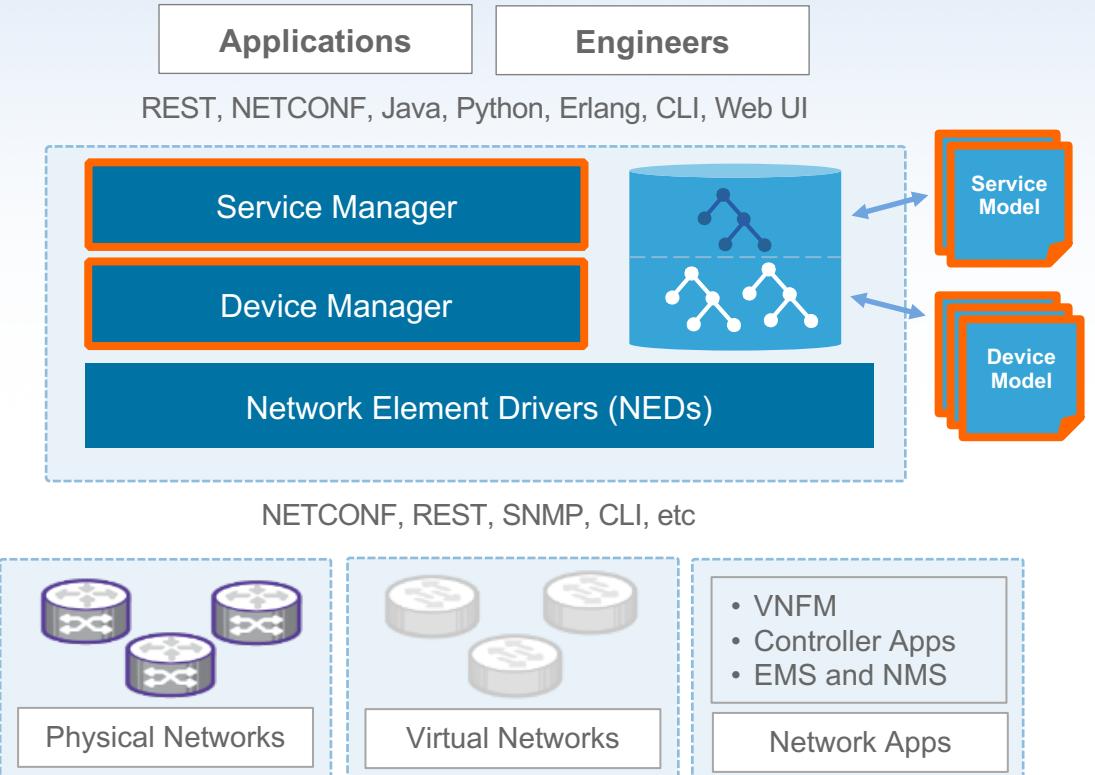
***Solutions Architect***

October 23, 2019

# Topics

- NSO Overview
- What does NETCONF/YANG Interop mean?
- NSO Tutorial
- NETCONF NED Builder
- DrNED Examiner Test Tool
- NSO Interop Lab
- Demo

# NSO Overview



- Real-time, multi-vendor Network Service Orchestrator (NSO)
- NSO is a Platform Framework, customized through:
  - Use case – the ‘service’
  - Specific devices deployed
  - Northbound agents
  - VNFs, VNF-M, VIM
- “Model-driven” approach

# Adapter Tax; The #1 Killer of Cool Telco Business Ideas

tail-f

## NETCONF

- NETCONF/YANG compliant devices
- YANG downloaded from device
- No code required

IOS-XR: 0 loc/leaf  
(0 loc / 200,000 leafs)

Junos: 0 loc/leaf  
(0 loc / 200,000 leafs)

Huawei 9k: 0 loc/leaf  
(0 loc / 50,000 leafs)

## CLI

- Hand written YANG with annotations
- Code to log in, recognize prompts, error messages, parse state data

IOS-XR: 0.10 loc/leaf  
(6 kloc / 60,000 leafs)

IOS: 0.15 loc/leaf  
(12 kloc / 80,000 leafs)

F5BIP: 2.5 loc/leaf  
(10 kloc / 4,000 leafs)

## Other

- REST, SOAP, Corba, TL1 ...
- Hand written YANG
- Code to log in, map requests to RPC calls, sequence calls, parse responses

NS: 4.3 loc/leaf  
(3900 loc / 900 leafs)

VXOA: 7.6 loc/leaf  
(1900 loc / 250 leafs)



# What does "...when it works" mean? How do you know if it "works"?

## "...when it works" means

- RFCs
- Service Automation Criteria

## How to know?

- Test your use cases in the NSO Interop Lab
  - Add ConfD, enable the NETCONF interface and be done?
  - Effort, Quality, Coverage varies greatly between products
  - Non-conformance often local to one YANG module or subsystem
  - Once a use case works, it typically works reliably

# Service Automation Criteria

## Mandatory NETCONF Basics

- Hello
- Locking
- Sub-tree filters

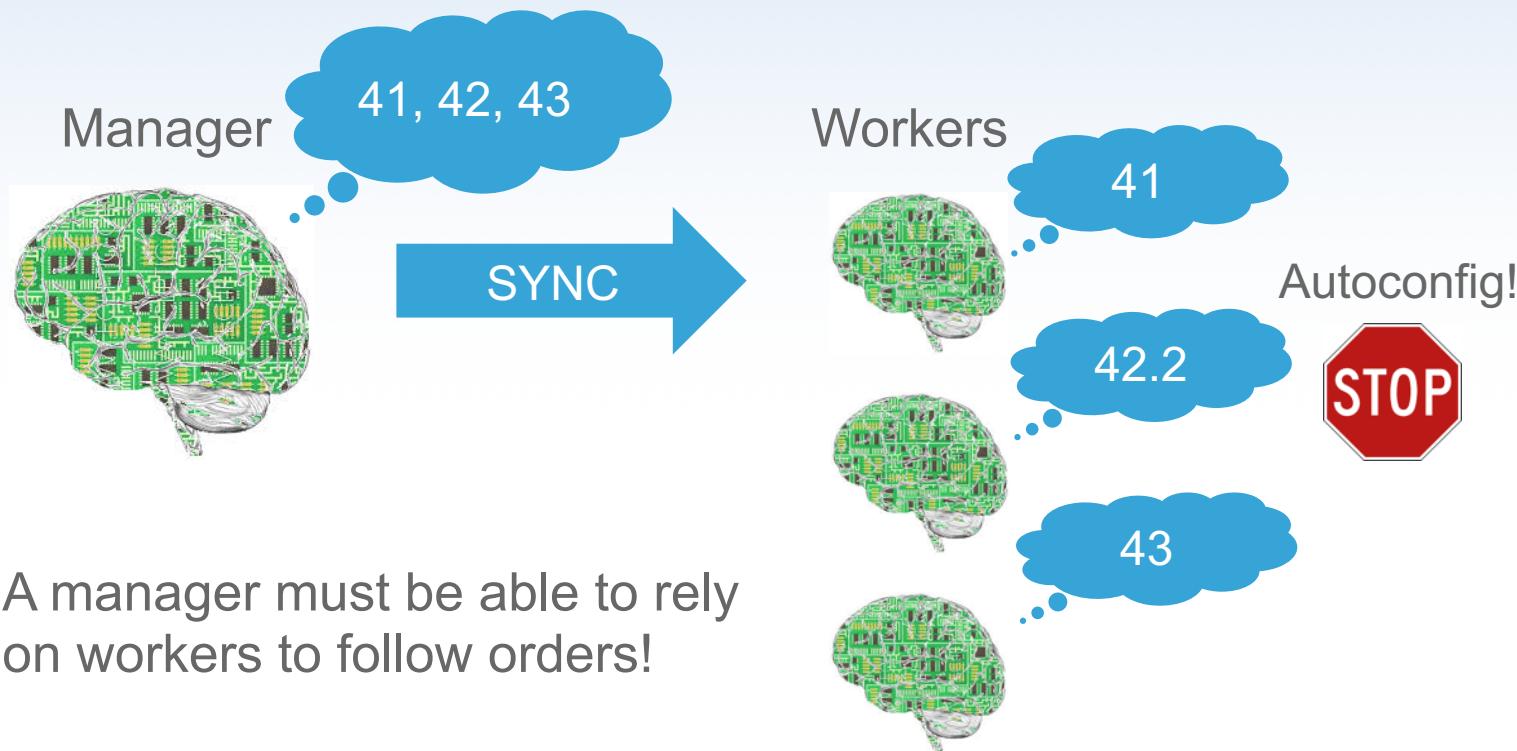
## YANG Model Correctness

- YANG model syntax
- YANG model constraints

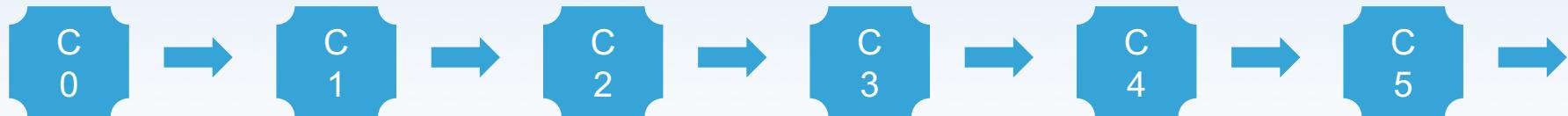
## YANG Model Compliance

- Types, ranges & constraints
- ### Transactionality
- All-or-nothing semantics
  - All-at-once semantics
  - Config validity depends on config alone
  - No Auto-config

# No Autoconfig in edit-config



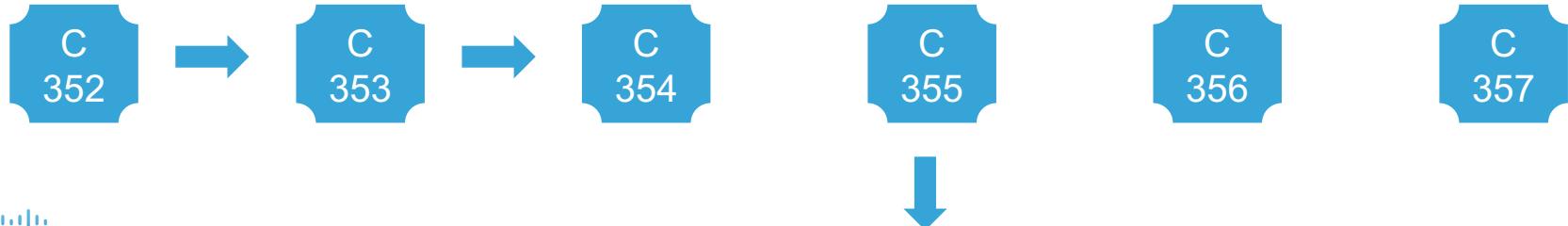
# Validation Dependencies



Since a configuration has been valid in the past, it must be valid now.



Managers rely on being able to go directly to any valid configuration



# Sequencing and Admin status

**“A minimal delta MUST only refer to any particular leaf once, i.e. cannot \*first\* set a leaf to one value, \*then\* set it to another value.”**

A transaction is a set of changes, not a sequence. Concepts such as first, then, last, before, and after are meaningless inside a transaction. A transaction can only give each leaf a single value.

Devices will need to properly sequence the work to reach the desired configuration, and cannot rely on operators to specify this order.

Modes which traditionally have prohibited changes (e.g. admin status) must not block transactional configuration changes.

# NSO Download Location and Resources

- NSO software for non-production use can be downloaded via Cisco DevNet at: <https://developer.cisco.com/site/nso>.
- NSO Interop Testing User Guide (<https://info.tail-f.com/nso-driven-netconf-and-yang-testing>)
  - In the process of being updated for NSO 5.2 and NETCONF NED Builder
- Test Your ConfD device with NSO (<https://info.tail-f.com/test-confd-device-with-ns0>)
- NSO Developer Hub (<https://community.cisco.com/t5/ns0-developer-hub/ct-p/5672j-dev-ns0>)



# NSO Tutorial

# Installing and running NSO

```
$ sh nso-<VERSION>.<OS>.<ARCH>.installer.bin  
/path/to/install_dir  
  
NCS_DIR$ source ncsrc  
  
$ ncs-setup --no-netsim --dest $HOME/ncs-run  
  
$ ls $HOME/ncs-run  
  
README.ncs      logs          ncs-cdb          ncs.conf  
packages       scripts        state  
  
$ cd $HOME/ncs-run  
  
ncs-run$ ncs
```

# Configuring the authentication group for your device in NSO

```
admin@ncs# config
```

```
admin@ncs(config)# devices authgroups group <group-name>
default-map remote-name <username> remote-password
<password>
```

```
admin@ncs(config-group-<group-name>)# commit
```

```
admin@ncs(config-group-<group-name>)# exit
```

# Setting up your test device in NSO

```
admin@ncs# config
```

```
admin@ncs(config)# devices device <device-name> device-type netconf ned-id netconf
```

```
admin@ncs(config-device-<device-name>)# address 10.80.26.213 port 830 authgroup <group-name>
```

```
admin@ncs(config-device-<device-name>)# state admin-state unlocked
```

```
admin@ncs(config-device-<device-name>)# commit
```

```
admin@ncs(config-device-<device-name>)# ssh fetch-host-keys
```



# NSO NETCONF NED Builder

# NETCONF NED Builder: Making the device YANG modules available to NSO

```
ncs-run$ ncs_cli -C -u admin
```

```
admin@ncs# devtools true
```

```
admin@ncs(config)# netconf-ned-builder project tail-dhcpd 1.0 device test1a local-user admin  
vendor Tail-f
```

```
admin@ncs(config-project-tail-dhcpd/1.0)# commit
```

```
admin@ncs(config-project-tail-dhcpd/1.0)# exit
```

```
admin@ncs(config)# exit
```

```
admin@ncs# show netconf-ned-builder project tail-dhcpd
```

```
netconf-ned-builder project tail-dhcpd 1.0
```

```
download-cache-path ./state/netconf-ned-builder/cache/tail-dhcpd-nc-1.0
```

```
ned-directory-path ./state/netconf-ned-builder/tail-dhcpd-nc-1.0
```

```
admin@ncs#
```

# NETCONF NED Builder: Uploading the device YANG modules

```
ncs-run$ ncs_cli -C -u admin
```

```
admin@ncs# devtools true
```

```
admin@ncs# netconf-ned-builder project tailf-dhcp 1.0  
fetch-module-list
```

```
admin@ncs# show netconf-ned-builder project tailf-  
dhcp 1.0 module
```

```
module dhcpcd ""
```

```
namespace http://tail-f.com/ns/example/dhcpcd
```

```
location [ NETCONF ]
```

```
module ietf-inet-types 2013-07-15
```

```
namespace urn:ietf:params:xml:ns:yang:ietf-inet-types
```

```
location [ NETCONF ]
```

```
...
```

```
admin@ncs# netconf-ned-builder project tailf-dhcp 1.0  
module dhcpcd "" select
```

```
admin@ncs# show netconf-ned-builder project tailf-  
dhcp 1.0 module status
```

NAME	REVISION	STATUS
<hr/>		
dhcpcd		selected,downloaded
ietf-inet-types	2013-07-15	selected,downloaded
tailf-common	2019-05-16	selected,downloaded
tailf-xsd-types	2017-11-20	selected,downloaded

# NETCONF NED Builder: Building and Exporting the NED

```
admin@ncs(config)# netconf-ned-builder project tail-dhcpd 1.0 build-ned
```

```
admin@ncs(config)# exit
```

```
admin@ncs# show netconf-ned-builder project tail-dhcpd 1.0 build-status
```

*build-status success*

```
admin@ncs# netconf-ned-builder project tailf-dhcp 1.0 export-ned to-  
directory /tmp
```

*tar-file /tmp/ncs-5.2.0.3-tailf-dhcp-nc-1.0.tar.gz*

```
admin@ncs# exit
```

```
ncs-run$ mv /tmp/ncs-5.2.0.3-tailf-dhcp-nc-1.0.tar.gz packages
```

# NETCONF NED Builder: Installing the NED and updating the ned-id of the test device

```
admin@ncs# packages reload
```

```
>>> System upgrade is starting.
```

```
>>> Sessions in configure mode must exit  
to operational mode.
```

```
>>> No configuration changes can be  
performed until upgrade has completed.
```

```
>>> System upgrade has completed  
successfully.
```

```
reload-result {
```

```
    package tail-dhcpd-nc-1.0
```

```
    result true
```

```
}
```



```
admin@ncs# config
```

```
admin@ncs(config)# devices device  
test1a device-type netconf ned-id
```

Possible completions:

```
    Isa-netconf netconf snmp tail-  
    dhcpd-nc-1.0
```

```
admin@ncs(config)# devices device  
test1a device-type netconf ned-id  
tail-dhcpd-nc-1.0
```

```
admin@ncs(config-device-test1a)#  
commit
```

Commit complete.



# DrNED Examiner Test Tool

# DrNED Examiner Package

- DrNED Examiner tool is a NSO package written in Python
  - Available at <https://github.com/NSO-developer/drned-xmnr>
- Provides automated NETCONF/YANG testing of a device's transactional behavior
  - Device to accept any valid configuration regardless of its current state
  - No auto-config - transaction modifies only what is specified with <edit-config>
  - Requires user to specify a good selection of input configurations

# How to set up DrNED Examiner for use with your device?

- Either use ncs\_cli to configure the device using its NETCONF NED
- Or configure device using any supported mechanisms by the device including CLI followed by sync-from in ncs\_cli
- Perform “devices device device-name drned-xmnr state record-state state-name *test-state-name*” in ncs\_cli’s config mode

# How to use DrNED Examiner's automated testing feature?

- Perform all possible state transitions
  - “devices device device-name drned-xmnr transitions explore-transitions” in ncs\_cli’s config mode
- Perform transition to a particular state
  - “devices device device-name drned-xmnr transitions transition-to-state state-name *test-state-name*” in ncs\_cli’s config mode



# NSO Interop Lab

# NSO Interop Lab

NSO Interop Lab is a free service open to equipment vendors and equipment users

## You get:

- NSO (freely downloadable for non-production use)
- NETCONF NED Builder
- DrNED Xmnr
- NSO Interop email support

## You contribute:

- Device/system to test
- Lab with NSO host and device
- One or more use cases
- Handful of typical configs for each use case

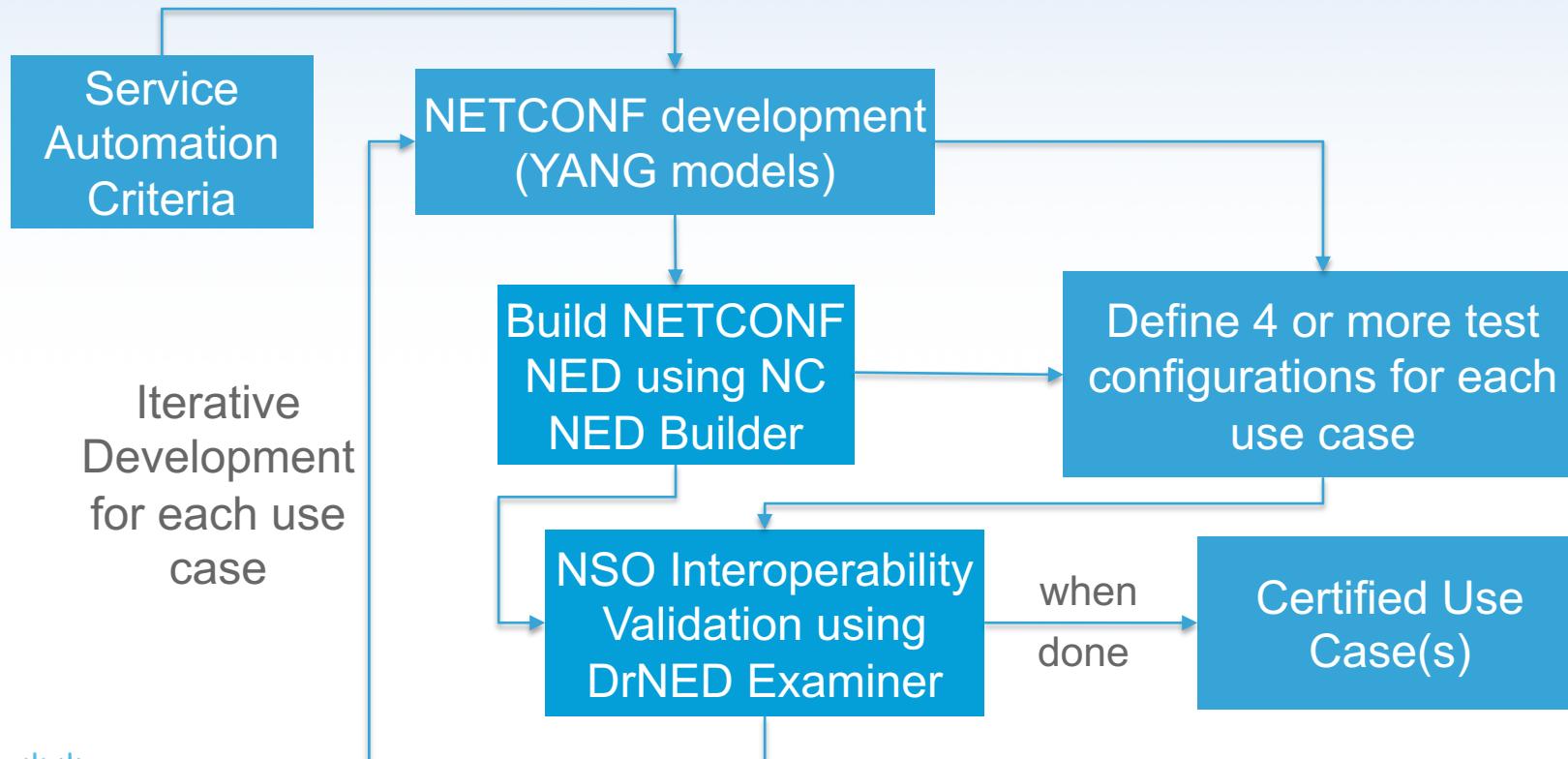
## You earn:

- Official use case listing

# Minimum Success Criteria for Certification

- Don't aim for the sky
- Instead select one or more common use cases for your product
- Define a day-0 base configuration and at least 3 day-1 running configurations as test cases for each use case
- Use DrNED Examiner to demonstrate that all possible combinations of state transitions for the defined test cases can be achieved successfully

# NSO Interoperability Validation Process Flow



# NSO Interop Lab Participation

## Equipment Vendors

- If you are ahead of competition in programmability, make it show! Get an official listing.

## Equipment Users

- Vendor says NETCONF is supported for your use case? Try it out with NSO Interop Lab tools.
- Don't want to do it yourself? Require the vendor to certify your use cases!

Sign up at  
[https://info.tail-f.com/nso\\_interop\\_lab](https://info.tail-f.com/nso_interop_lab)





tail-f

# Demo