The bridge to possible

Stop the Chaos, Organize Your Network with NSO and Netbox

Anna Wojcik, Software Engineer Infrastructure, Cisco Systems - NSO BU

cisco Live!

DEVNET-2459

Cisco Webex App

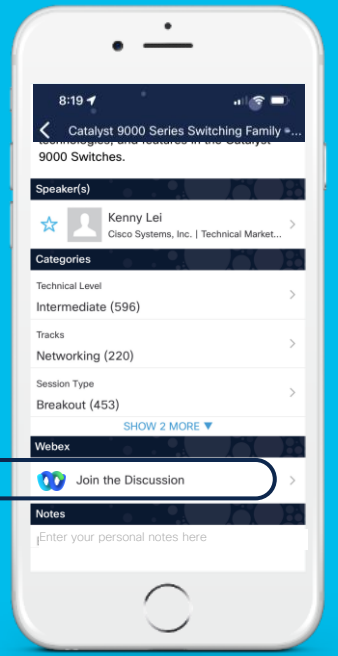
Questions?

Use Cisco Webex App to chat
with the speaker after the session

How

1. Find this session in the Cisco Live Mobile App
2. Click “Join the Discussion”
3. Install the Webex App or go directly to the Webex space
4. Enter messages/questions in the Webex space

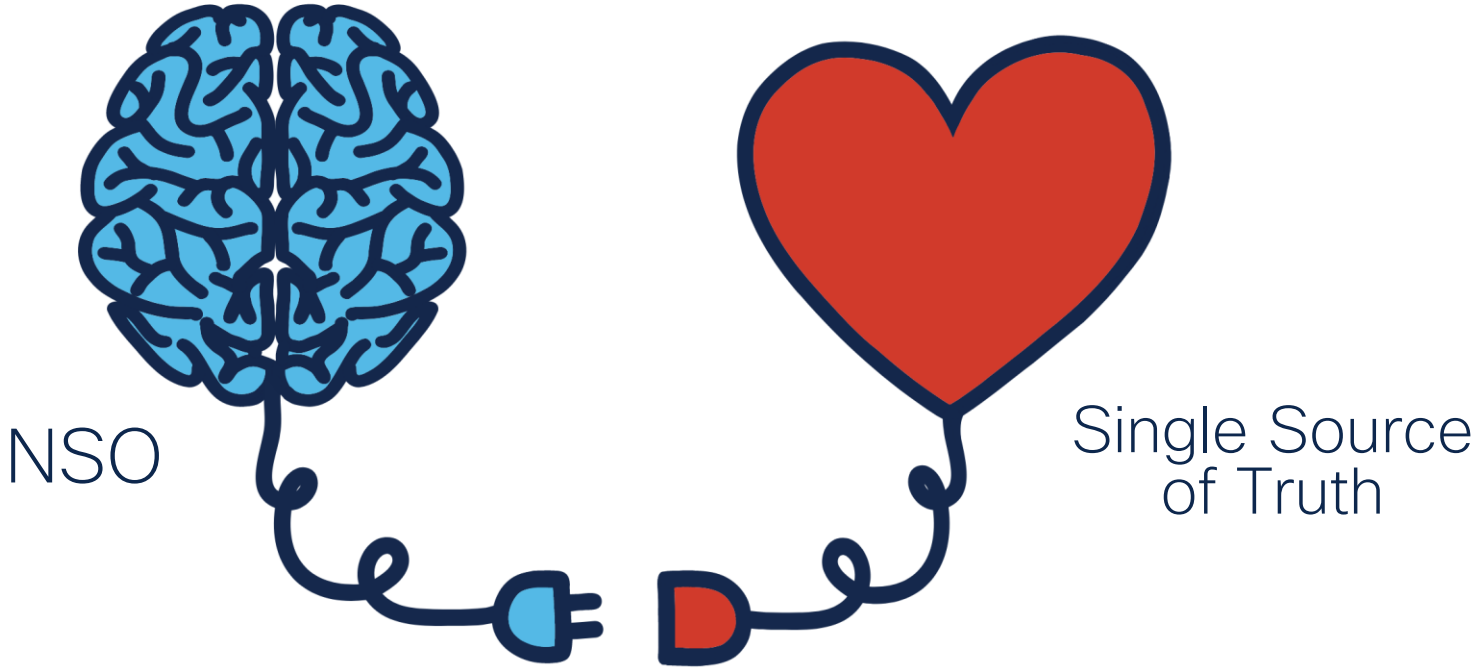
Webex spaces will be moderated until February 24, 2023.



“In the Beginning, There was Chaos”

Agenda

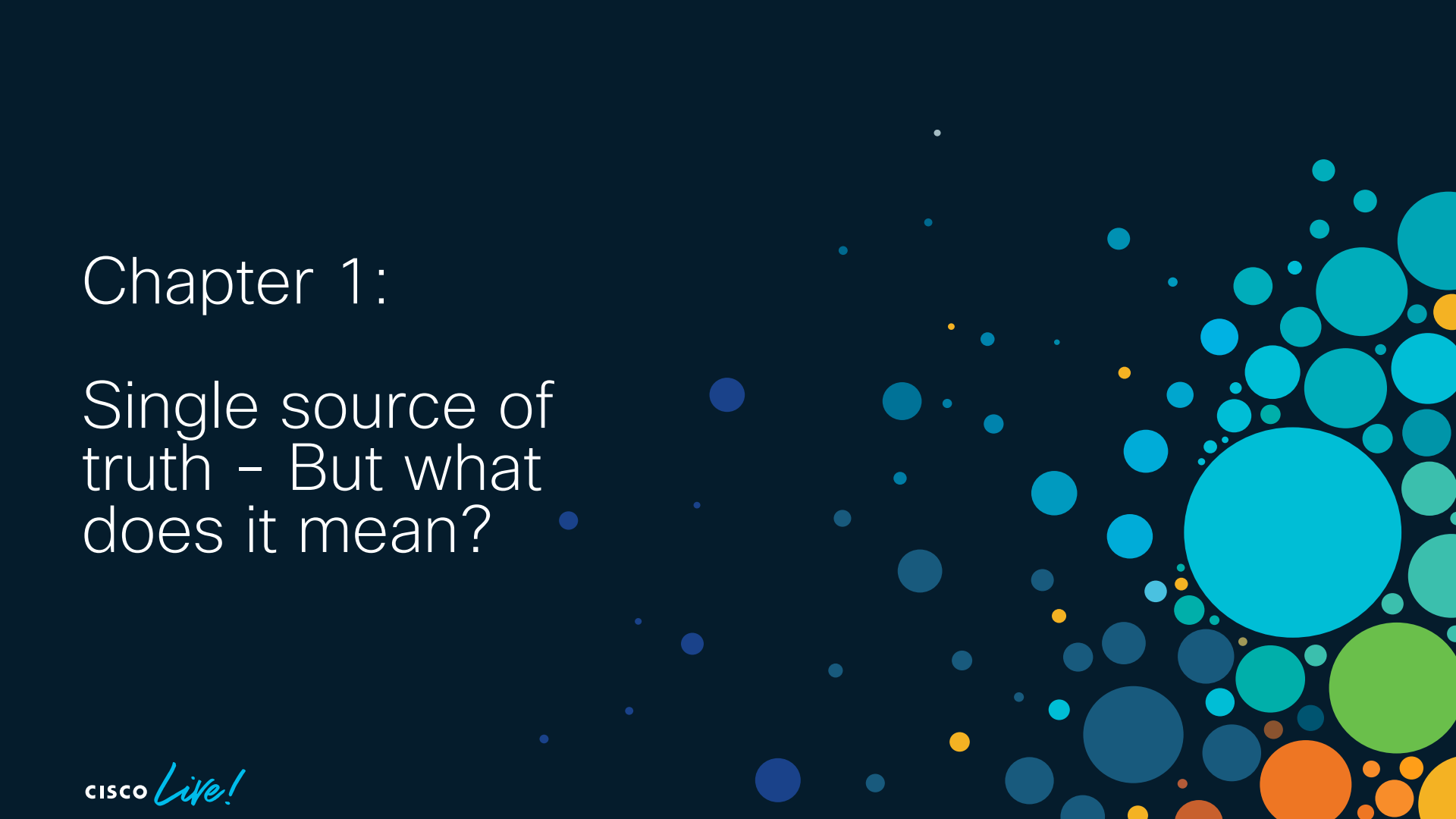
1. Single Source of Truth
2. Single Network Interface
3. NSO and Netbox meet
4. What's next?



Network Automation

Chapter 1:

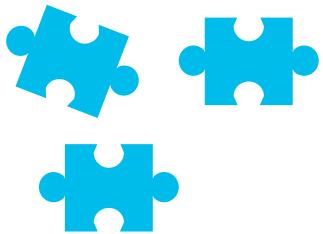
Single source of truth - But what does it mean?



Where do we keep information about the network?

- HLD, LLD
- Spreadsheets
- IPAM
- Git repository
- ...ohhhh, there is this one guy, that should know that...

- Personal notes
- Wiki
- Databases
- NSO
- ...



What information do we want to keep?

- Inventory (racks, devices, modules)
- IP address management
- Cabling and connections
- Locations and sites
- Virtual Machines
- Vlans, VRFs, AS
- Anything else? - Sure

How about Netbox?

Take advantage of Netbox

Base

- Data Model
- Customization
- Open Source
- Dockerized

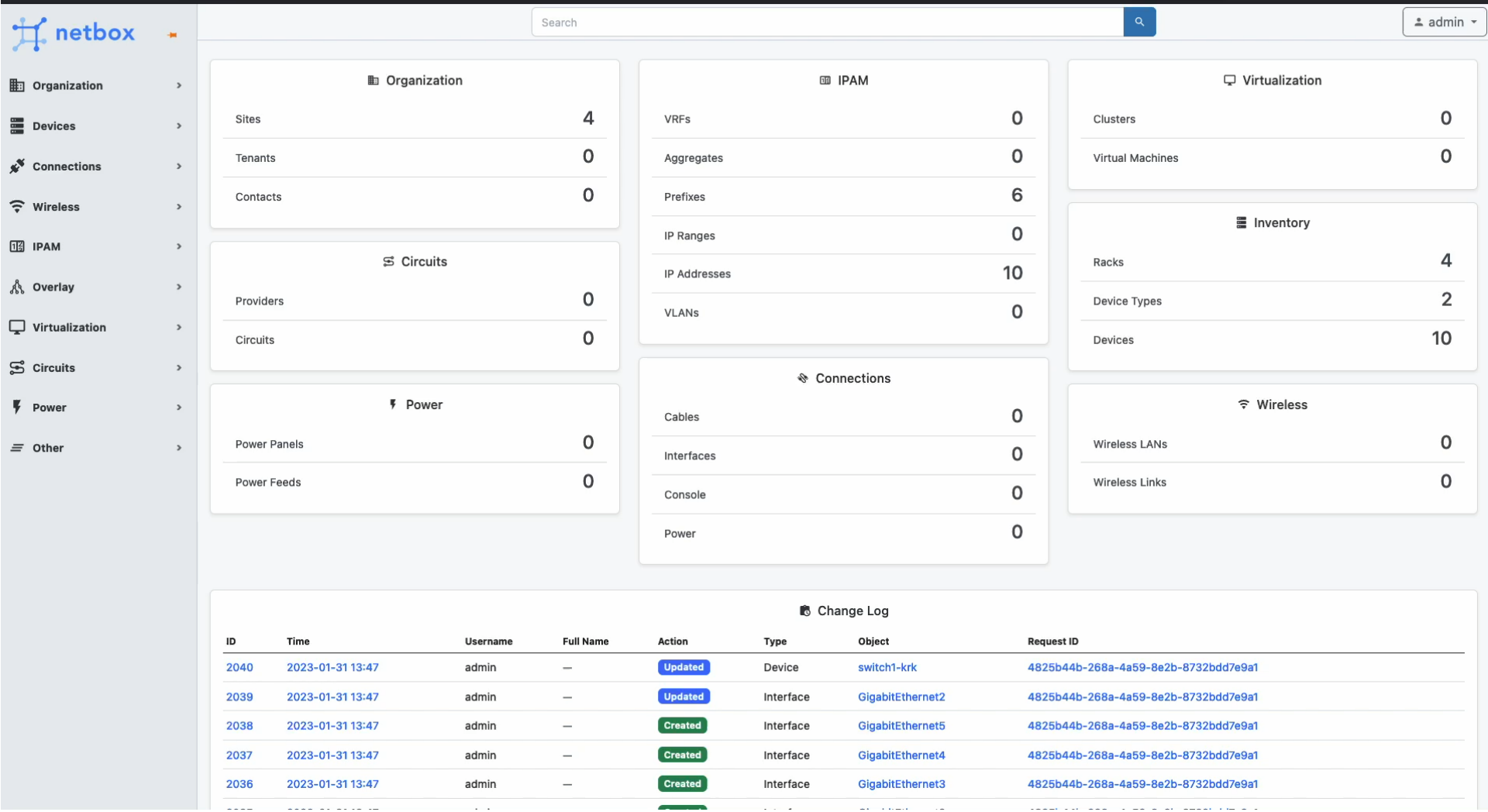
Integrations

- Scripts
- Webhooks
- Plugins

API

- REST
- Swagger
- GraphQL

DEMO

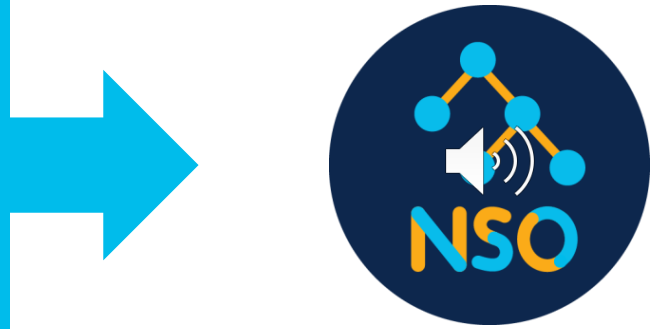

netbox
Search
admin
Organization
Devices
Connections
Wireless
IPAM
Overlay
Virtualization
Circuits
Power
Other
Organization
Sites 4
Tenants 0
Contacts 0
Circuits
Providers 0
Circuits 0
Power
Power Panels 0
Power Feeds 0
IPAM
VRFs 0
Aggregates 0
Prefixes 6
IP Ranges 0
IP Addresses 10
VLANs 0
Connections
Cables 0
Interfaces 0
Console 0
Power 0
Virtualization
Clusters 0
Virtual Machines 0
Inventory
Racks 4
Device Types 2
Devices 10
Wireless
Wireless LANs 0
Wireless Links 0
Change Log
ID Time Username Full Name Action Type Object Request ID
2040 2023-01-31 13:47 admin — Updated Device switch1-krk 4825b44b-268a-4a59-8e2b-8732bdd7e9a1
2039 2023-01-31 13:47 admin — Updated Interface GigabitEthernet2 4825b44b-268a-4a59-8e2b-8732bdd7e9a1
2038 2023-01-31 13:47 admin — Created Interface GigabitEthernet5 4825b44b-268a-4a59-8e2b-8732bdd7e9a1
2037 2023-01-31 13:47 admin — Created Interface GigabitEthernet4 4825b44b-268a-4a59-8e2b-8732bdd7e9a1
2036 2023-01-31 13:47 admin — Created Interface GigabitEthernet3 4825b44b-268a-4a59-8e2b-8732bdd7e9a1

Chapter 2:

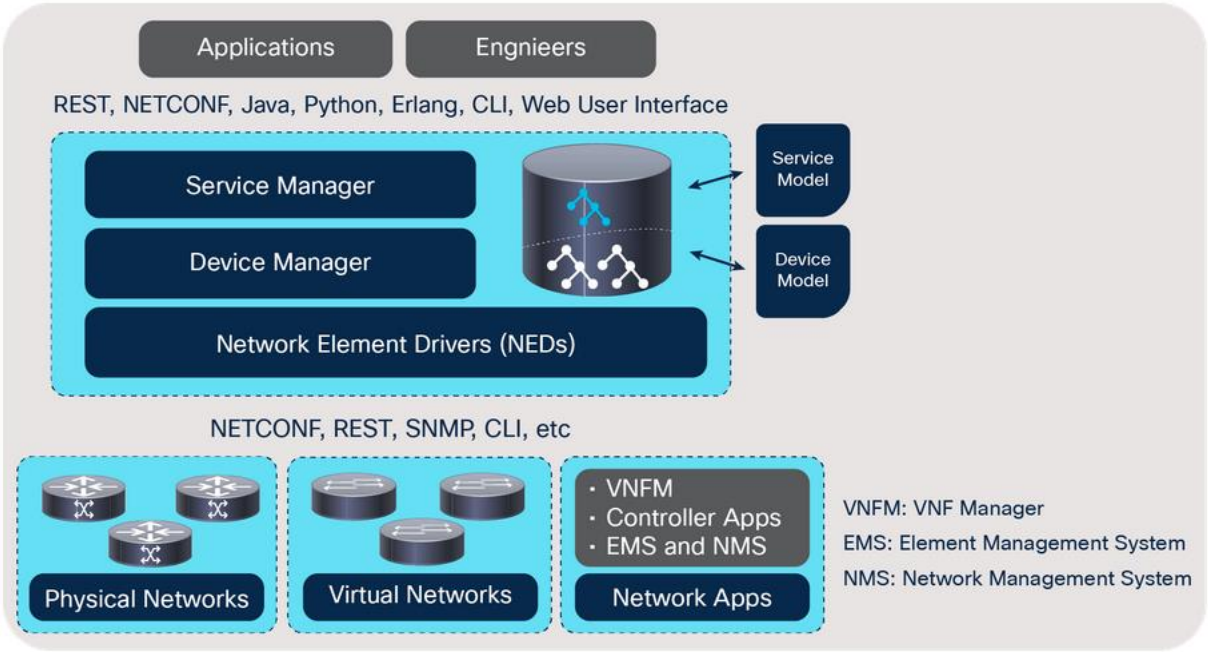
Single Network Interface

How do we talk to our devices?

- CLI
- CLI != CLI
- SNMP
- XML
- NETCONF
- RESTCONF
- Web Interface
- API



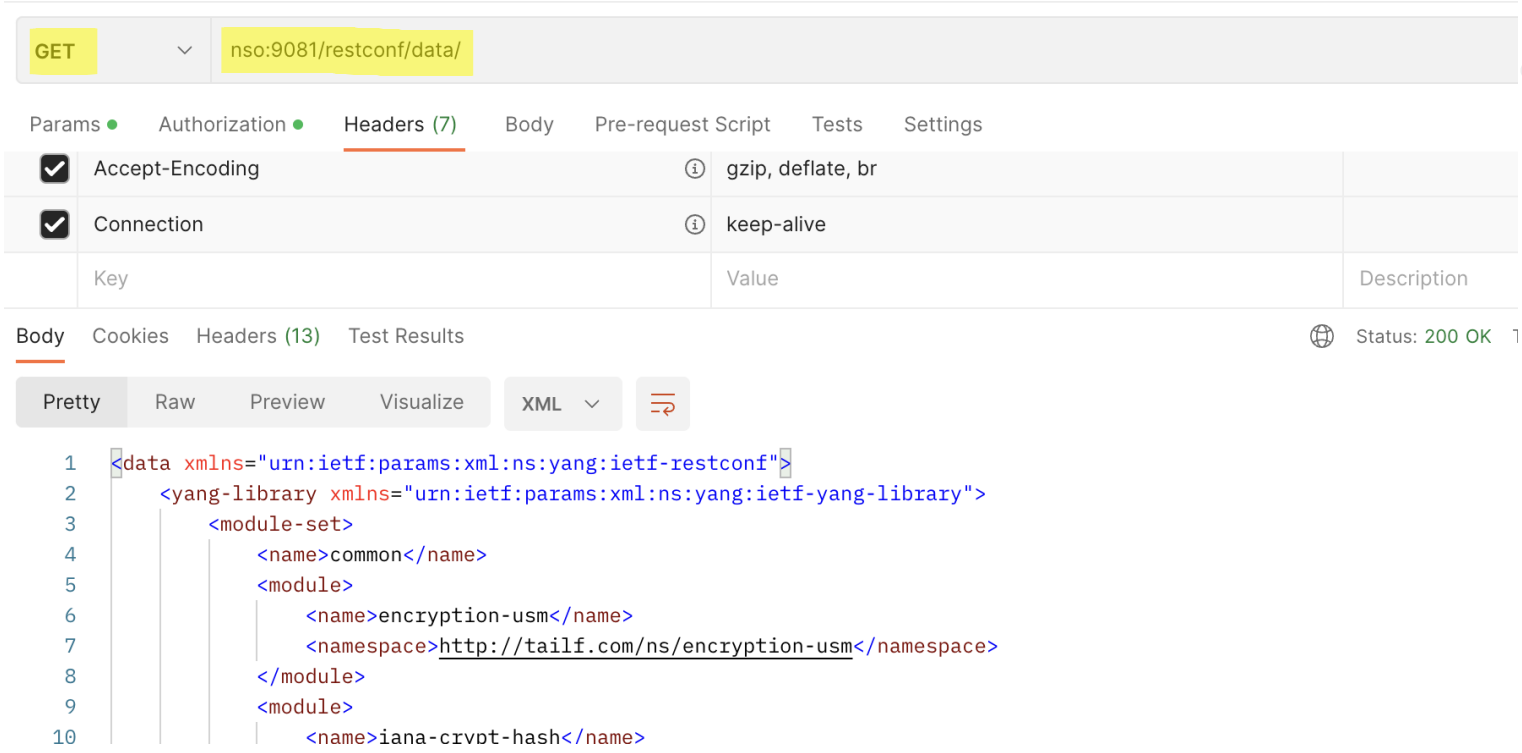
Network Element Driver



NSO

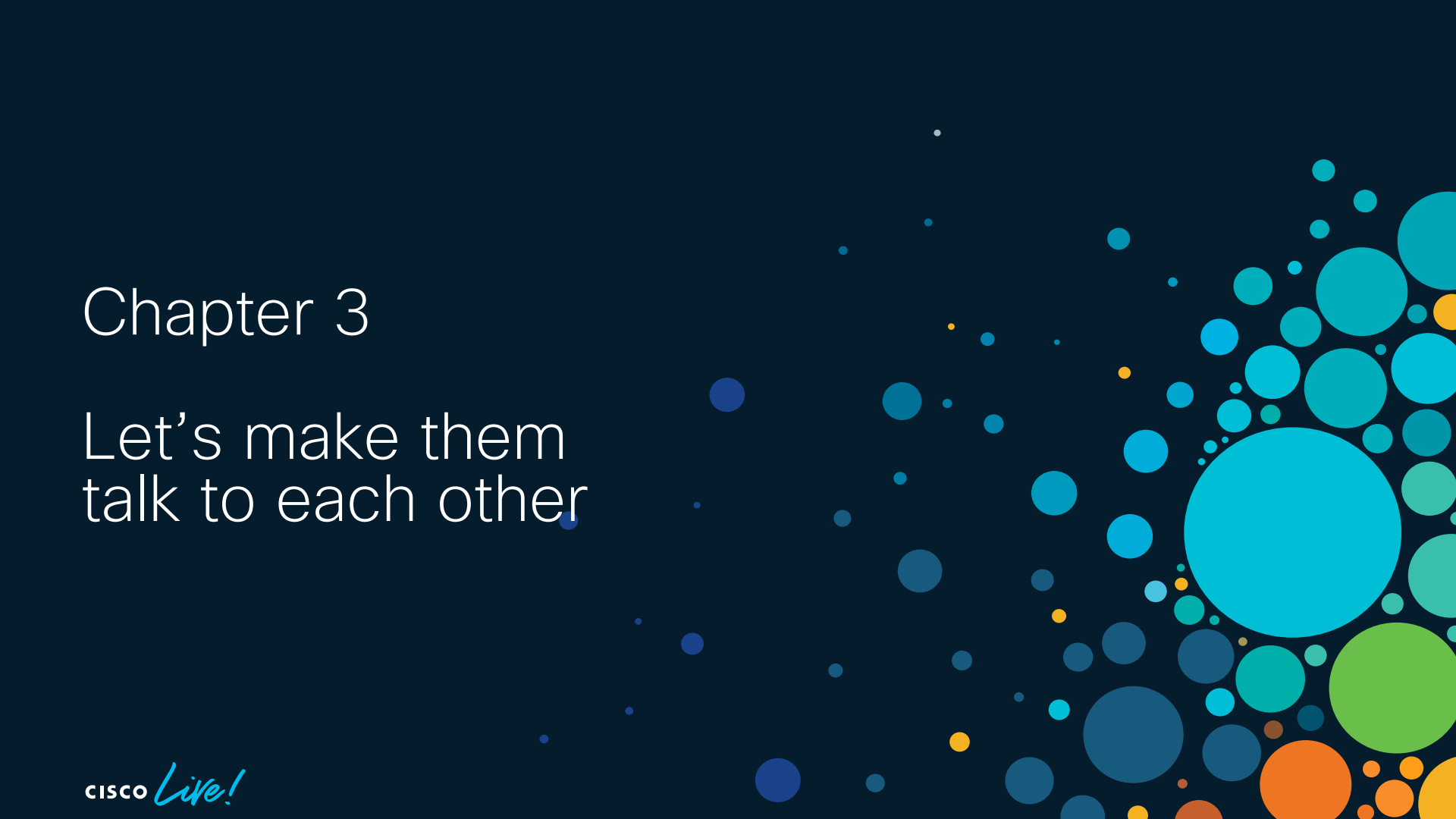
- Multivendor orchestration platform (NED)
- Configuration synchronization
- Configuration templates
- Services (YANG)
- Single API to entire network

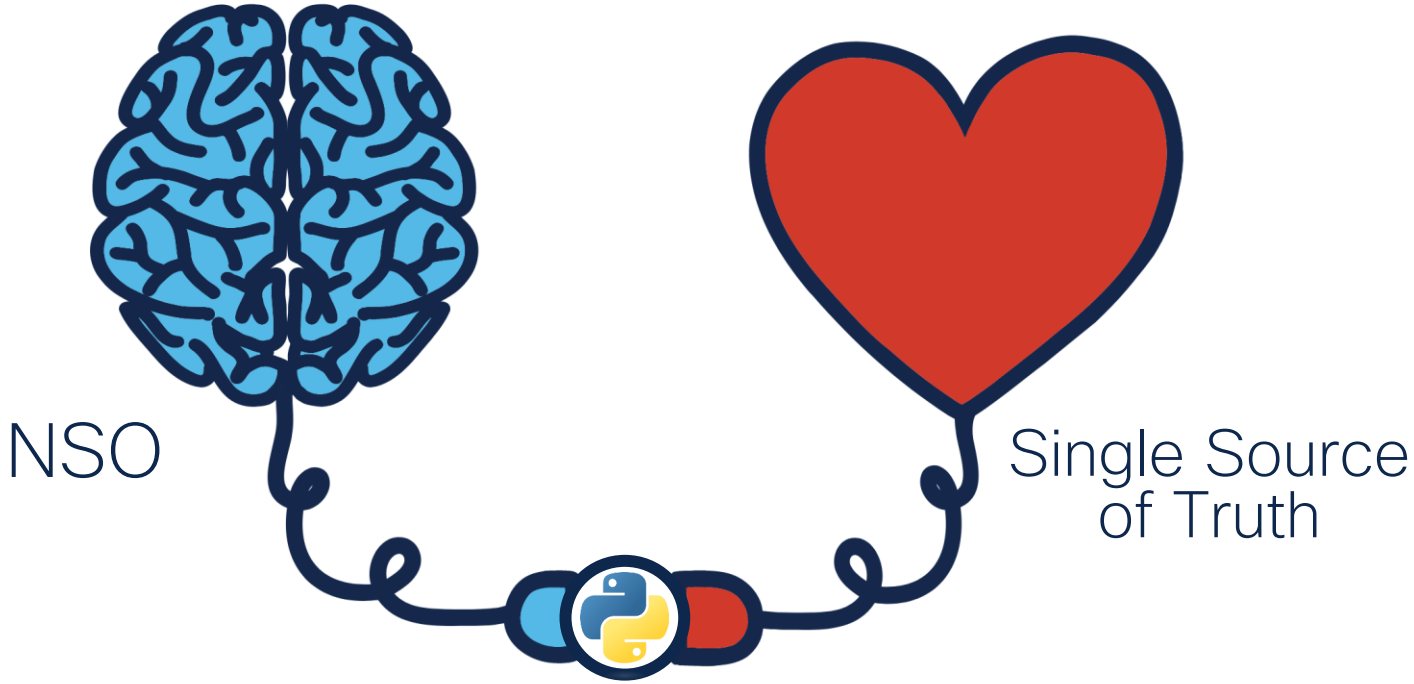
RESTCONF API



Chapter 3

Let’s make them talk to each other





Network Automation

Part 1 - New branch office

1. Create:
   Site, Rack, Devices, IP subnet
2. Create associations
3. Add devices to NSO

How to create Netbox script?

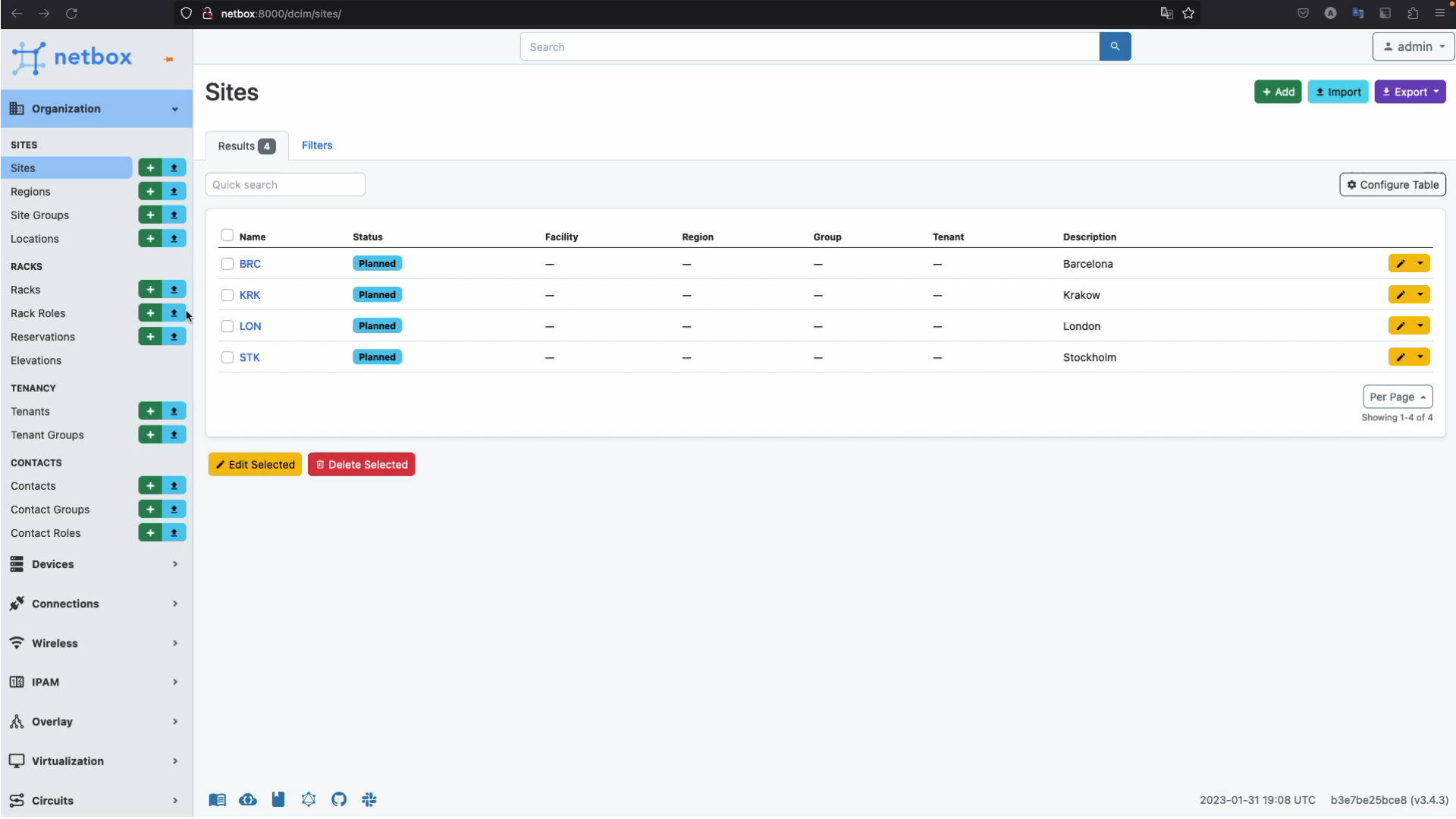
Define input data

```
class NewSiteScript(Script):
    class Meta:
        name = "New Site"
        description = "Provision a new site"
        field_order = ['site_codename', 'site_name', 'switch_count'

    site_codename = StringVar(
        description="Short name of the new site",
    )
    site_name = StringVar(
        description="Full name of the new site",
    )
    switch_count = IntegerVar(
        description="Number of access switches in the site",
        default=1
    )
    switch_model = ObjectVar(
        description="Switch model",
        model=DeviceType
    )
    router_model = ObjectVar(
        description="Router model",
        model=DeviceType,

    )
    add_to_nso = BooleanVar(
        description="Add devices to NSO inventory"
    )
```

Define logic

```
def run(self, data, commit):
    # data from form
    site_codename = data['site_codename']
    site_name = data['site_name']
    switch_count = data['switch_count']
    switch_model = data['switch_model']
    router_model = data['router_model']
    add_to_nso = data['add_to_nso']
    rack_size = 16

    site = self.create_new_site(site_codename, site_name)
    mgmt_id = self.find_next_free_mgmt_id()
    prefix = self.create_mgmt_prefix(site, mgmt_id)
    dns_results = self.dns_allocations(site, mgmt_id, switch_count)
    rack = self.create_rack(site, rack_size)
    router = self.create_router(site, router_model, mgmt_id, rack)
    list_of_switches = self.create_switch(site, switch_model, switch_count, mgmt_id, rack)
    if add_to_nso:
        self.add_devices_to_nso(site, dns_results)
```

DEMO


netbox:8000/dcim/sites/
netbox
Search
admin
Organization
SITES
Sites
Regions
Site Groups
Locations
RACKS
Racks
Rack Roles
Reservations
Elevations
TENANCY
Tenants
Tenant Groups
CONTACTS
Contacts
Contact Groups
Contact Roles
Devices
Connections
Wireless
IPAM
Overlay
Virtualization
Circuits
Sites
+ Add
Import
Export
Results 4
Filters
Quick search
Configure Table
Name
Status
Facility
Region
Group
Tenant
Description
BRC
Planned
—
—
—
—
Barcelona
KRK
Planned
—
—
—
—
Krakow
LON
Planned
—
—
—
—
London
STK
Planned
—
—
—
—
Stockholm
Per Page
Showing 1-4 of 4
Edit Selected
Delete Selected
2023-01-31 19:08 UTC
b3e7be25bce8 (v3.4.3)

Means to interact

1) Netbox data with native python packages

```
def create_new_site(self, site_codename, site_name):
    site = Site(
        name=site_codename,
        slug=slugify(site_codename),
        description=site_name,
        status=SiteStatusChoices.STATUS_PLANNED
    )
    site.save()
    self.log_success("New site %s (%s) created" % (site_codename, site_name))
    return site
```

Means to interact

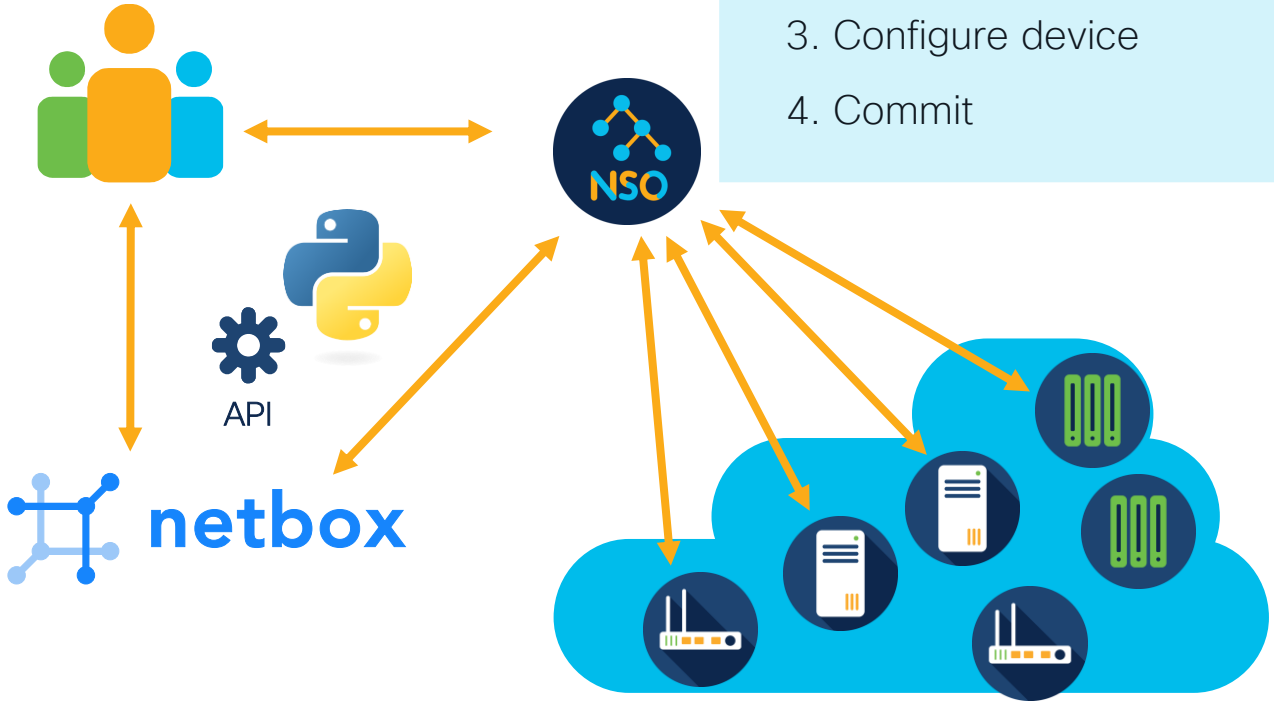
1) Netbox data with native python packages

2) NSO data with requests

```
headers = {
    'Accept': 'application/yang-data+json',
    'Content-Type': 'application/yang-data+json',
    'Authorization': 'Basic YWRtaW46YWRtaW4='
}

payload = json.dumps({
    "tailf-ncs:device": [
        {
            "name": name,
            "address": ip,
            "port": 22,
            "authgroup": "mygroup",
            "device-type": {
                "cli": {
                    "ned-id": "cisco-ios-cli-3.8:cisco-ios-cli-3.8"
                }
            },
            "state": {
                "admin-state": "unlocked"
            }
        }
    ]
})

response = requests.request("POST", url, headers=headers, data=payload)
resp = response.status_code
```

Part 2 – DHCP server

1. GET subnet information from Netbox
2. Reserve IP range for DHCP pool
3. Configure device
4. Commit



How to create NSO python script?

Talk to Netbox

```python
def get_subnet():
    url = netbox + "ipam/prefixes/?site=" + site
    response = requests.request("GET", url, headers=headers, data={}).json()
    subnet = response["results"][0]["prefix"]
    return subnet


def reserve_dhcp_pool(subnet):
    start_address = subnet.replace(".0/24", ".129/24")
    end_address = subnet.replace(".0/24", ".254/24")

    url = netbox + "ipam/ip-ranges/"
    payload = json.dumps({
        "start_address": start_address,
        "end_address": end_address,
        "description": pool_name
    })
    response = requests.request("POST", url, headers=headers, data=payload)
    return response
```

Configure device

```python
def configure_device(subnet, network_number, default_router):
    with ncs.maapi.single_write_trans('admin', 'python', groups=['ncsadmin']) as t:
        root = ncs.maagic.get_root(t)
        device_cdb = root.devices.device[device_name]
        device_cdb.config.ios__ip.dhcp.pool.create(pool_name)
        device_cdb.config.ios__ip.dhcp.pool[pool_name].default_router.create(default_router)
        device_cdb.config.ios__ip.dhcp.pool[pool_name].network.network_number = network_number
        device_cdb.config.ios__ip.dhcp.pool[pool_name].network.mask = "255.255.255.128"
        t.apply()
```

DEMO

root@a3d329aded0b: ~/nso-lab-rundir (ssh)
root@a3d329aded0b:~/nso-lab-rundir#

Means to interact

1) Netbox data with native python packages

2) NSO data with requests

3) NSO data with built in API

```
with ncs.maapi.single_write_trans('admin', 'python', groups=['ncsadmin']) as t:
    root = ncs.maagic.get_root(t)
    device_cdb = root.devices.device[device_name]
    device_cdb.config.ios__ip.dhcp.pool.create(pool_name)
    device_cdb.config.ios__ip.dhcp.pool[pool_name].default_router.create(default_router)
    device_cdb.config.ios__ip.dhcp.pool[pool_name].network.network_number = network_number
    device_cdb.config.ios__ip.dhcp.pool[pool_name].network.mask = network_mask
    params = t.get_params()
    params.dry_run_native()
    result = t.apply_params(True, params)
    t.apply_params(True, t.get_params())
```

Means to interact

1) Netbox data with native python packages

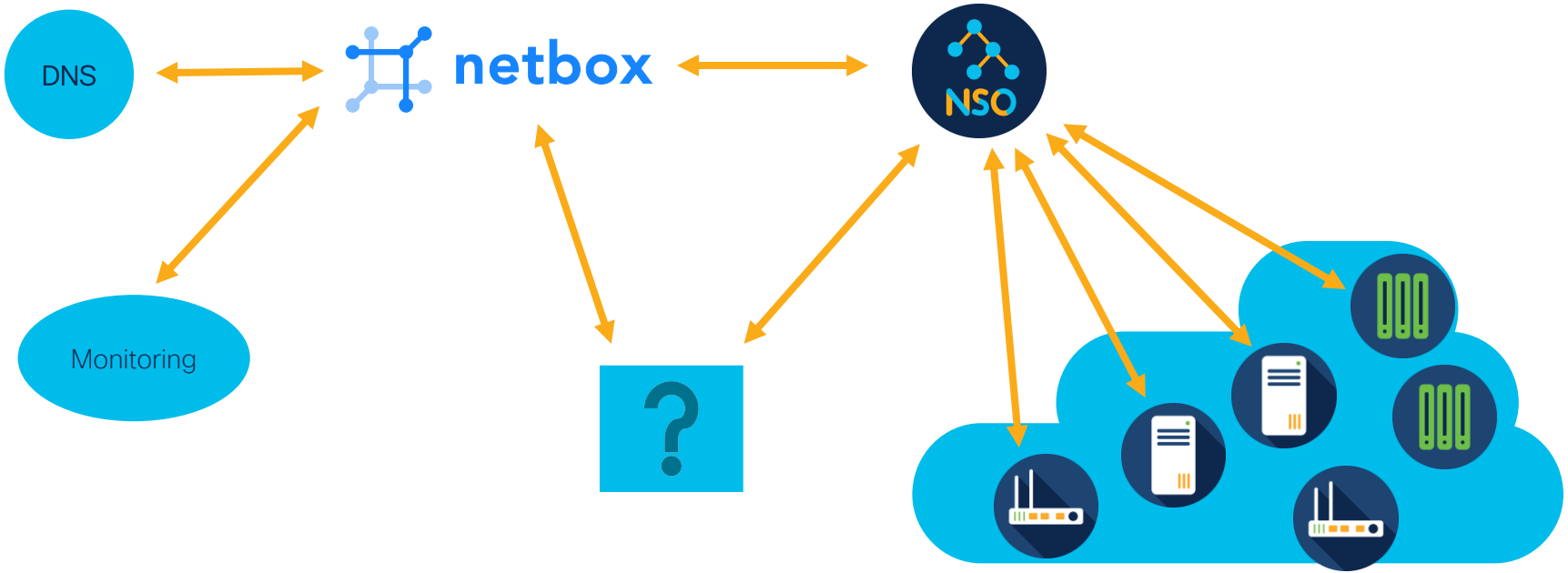
2) NSO data with requests

3) NSO data with built in API

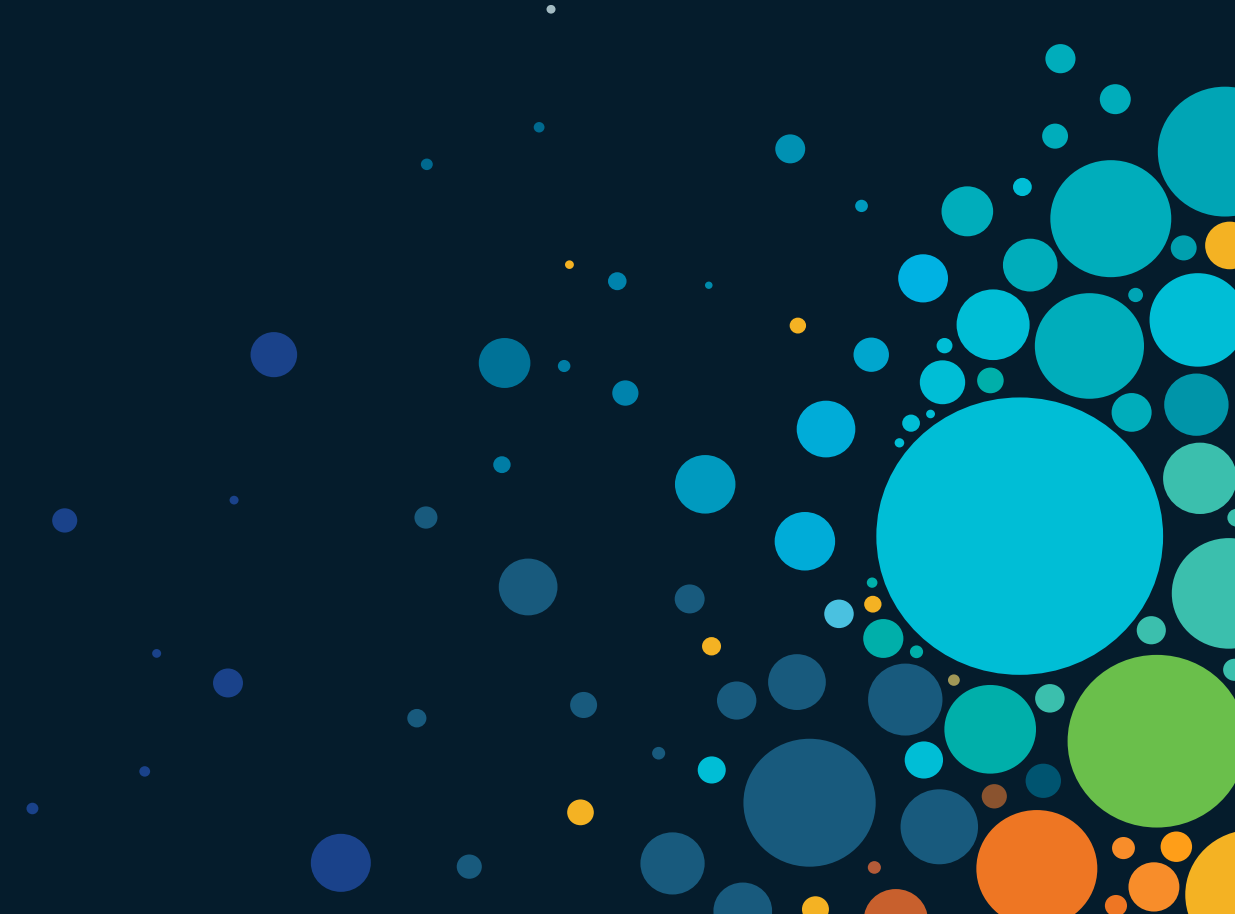
4) Netbox data with requests

```
url = netbox + "ipam/ip-ranges/"
payload = json.dumps({
    "start_address": start_address,
    "end_address": end_address,
    "description": pool_name
})
response = requests.request("POST", url, headers=headers, data=payload)
```

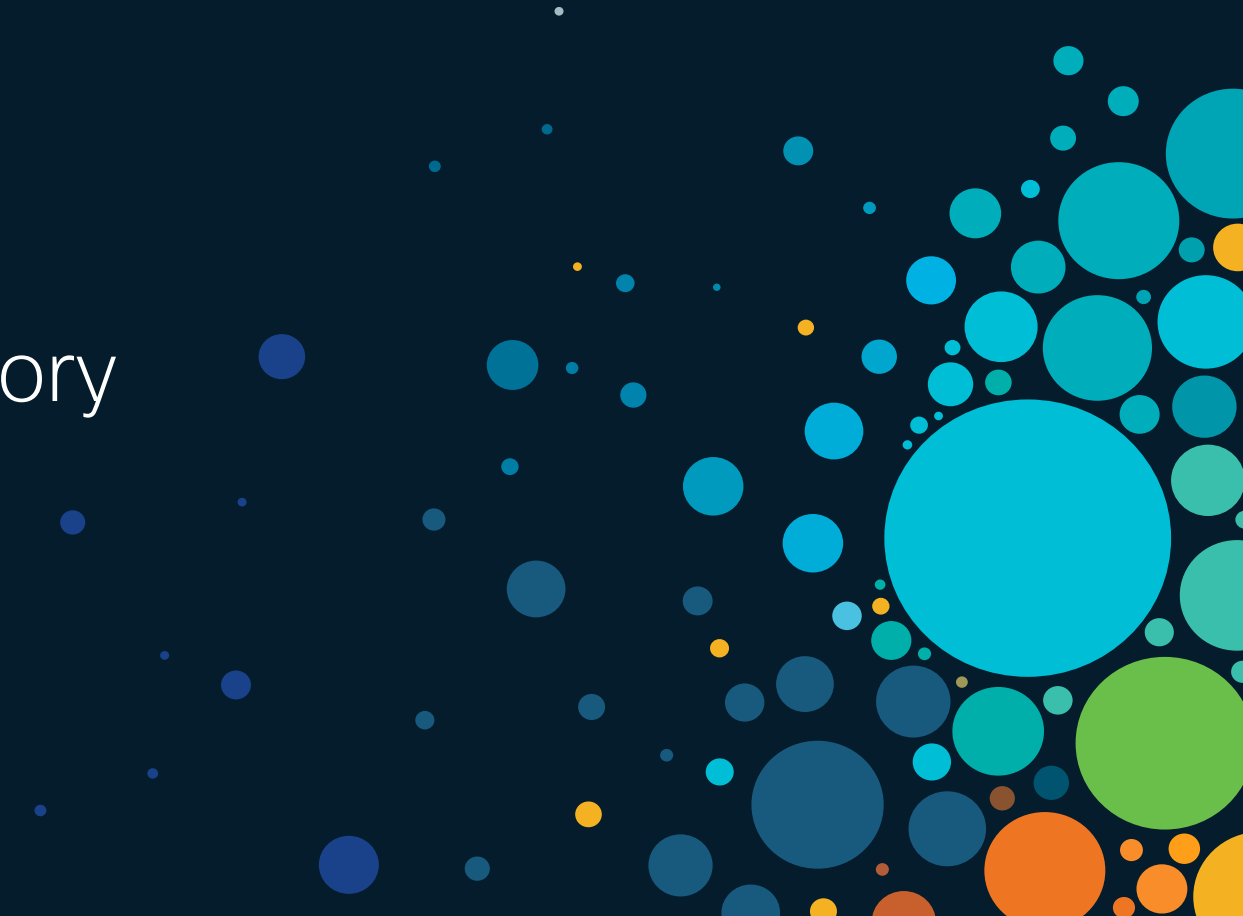
What can happen next?

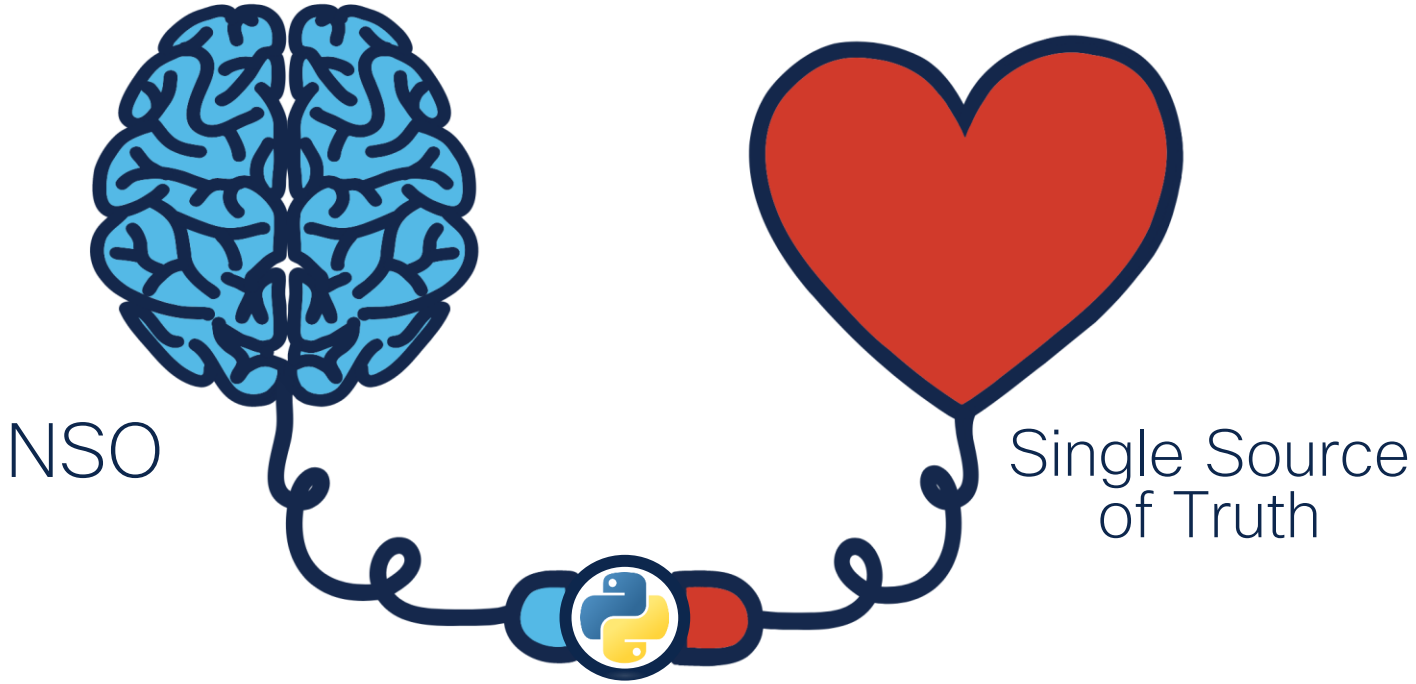


Q&A



Moral of the Story





Network Automation

Add NSO and Netbox to your Automation Toolbox

… and experiment with them

cisco.com/go/nsohub

docs.netbox.dev

github.com/annately/nso-netbox-cl

Developer Days
Automation

cisco
The bridge to possible

Can Automation enable the future Internet and help you transition to a greener network?

Service Developers, Operations and DevOps Engineers

May 9-11, Stockholm

Find information on how to submit a talk or register on The Developer Hub!
www.cisco.com/go/nsohub

Complete your Session Survey

- Please complete your session survey after each session. Your feedback is important.
- All surveys can be taken in the Cisco Events Mobile App or by logging in to the Session Catalog and clicking the "Attendee Dashboard" at https://www.ciscolive.com/emea/learn/sessions/session-catalog.html

Continue Your Education

Visit the Cisco Showcase for related demos.

Book your one-on-one Meet the Engineer meeting.

Attend any of the related sessions at the DevNet, Capture the Flag, and Walk-in Labs zones.

Visit the On-Demand Library for more sessions at ciscolive.com/on-demand.


cisco
The bridge to possible
Thank you
cisco Live!


CISCO Live!
ALL IN