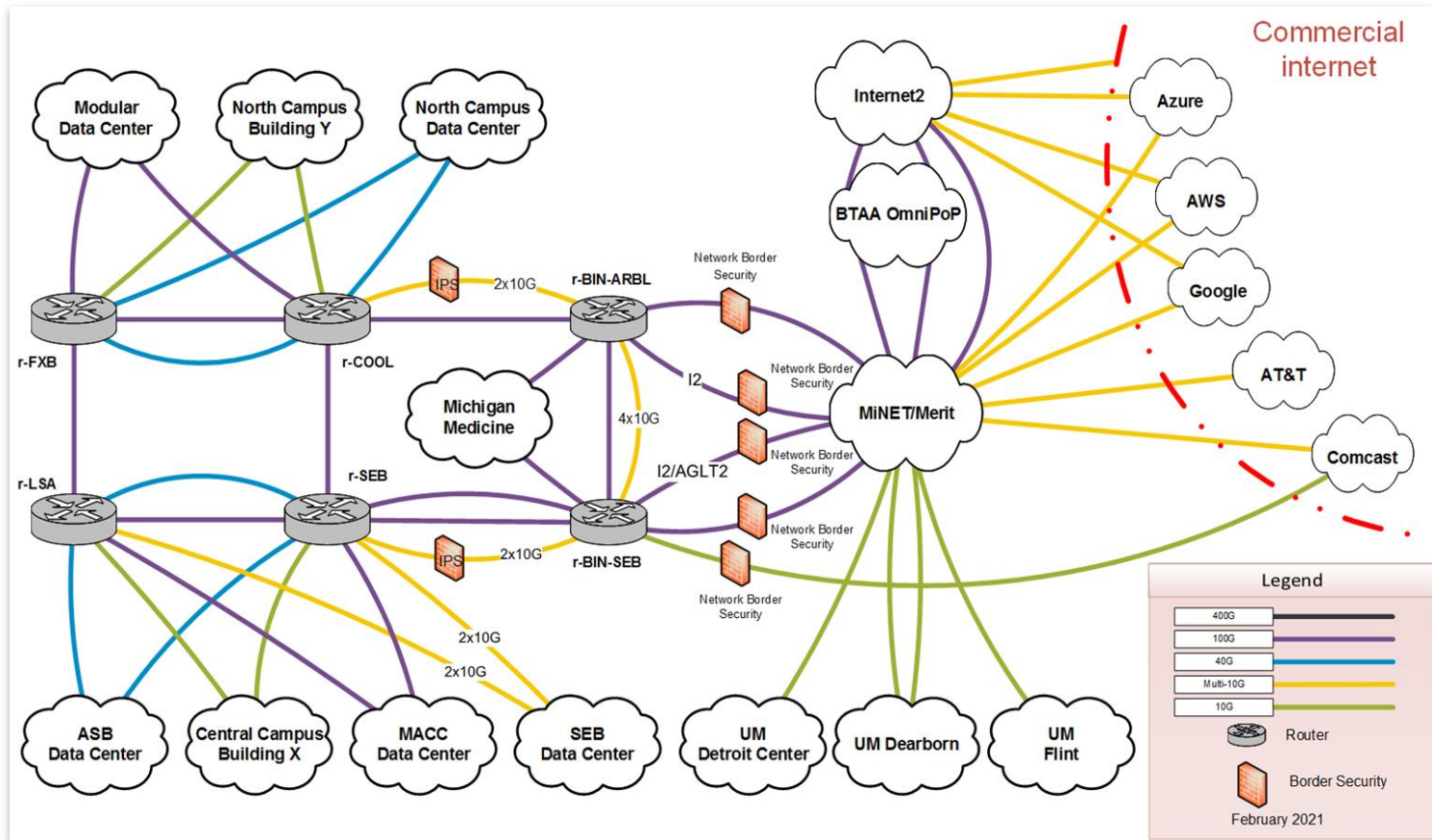


Automating the Campus Network with Cisco NSO

Amy Liebowitz, Nick Grundler, James Ostrander





Architecting for Automation

- Defining the problem is the most difficult part.
 - “19 schools and colleges that share a football team”
 - config changes coming from many different directions
 - DCT, some units have CLI access, home-grown scripts, etc.
- Creating software models to organize our data is a powerful tool for addressing complexity.
- We are deep in the process of building these models from the bottom up based on existing/deployed configuration
- Collaborative effort spanning engineering, operations, and software teams
- Also working on greenfield EVPN deployment for our next-gen core network



INFORMATION AND
TECHNOLOGY SERVICES
UNIVERSITY OF MICHIGAN

Issues with lack of data models:

The “Grand Oral Tradition” of ITS, and other scary stories...

1. Current state: services are defined by an undocumented, shared understanding in of what the important values and variables are for configuring networks.
 - a. How do you reliably automate services without a concrete definition?
2. Very often we have no idea why a particular configuration in the network exists and who needed it there -- if we don't understand it we are reluctant to change it.
3. The consequences of this are numerous:
 - a. automation is fragile and difficult
 - b. design intent often must be reverse-engineered by attempting to interpret configuration that might exist across several devices
 - c. heavily and increasingly reliant on institutional/esoteric knowledge of a small number of experienced staff
 - d. ...all of the above means it can take a long time to safely and reliably implement changes in the network, and we introduce opportunities for human error by forcing the network operators to deal with this complexity manually.
 - e. stressed out staff are more likely to make errors in the first place



Why YANG?

RFC 7950:

<https://tools.ietf.org/html/rfc7950>

“The model defines a contract between a YANG-based client and server; this contract allows both parties to have faith that the other knows the syntax and semantics behind the modeled data. The strength of YANG lies in the strength of this contract.”



INFORMATION AND
TECHNOLOGY SERVICES
UNIVERSITY OF MICHIGAN

```
list user {  
    key "login-name";  
    leaf login-name {  
        type string;  
    }  
    leaf full-name {  
        type string;  
    }  
}
```

Example XML instances:

```
<user>  
  <login-name>hakanm</login-name>  
  <full-name>Hakan Millroth</fullname>  
</user>  
<user>  
  <login-name>mbj</login-name>  
  <full-name>Martin Bjorklund</fullname>  
</user>
```



Multi-vendor Automation with NSO

- Purpose-built for managing multi-vendor networks
 - extensible support for Cisco IOS/XR/XE/NX/ASA/Aireos, Juniper, Arista, F5, and more
- Abstracts all network device configuration into structured, hierarchical in-memory XML database
- Configuration database enables transactional configuration changes for ALL supported devices
- Very fast feature development cycle for individual network element drivers(days)



Mapping NSO Services to Devices

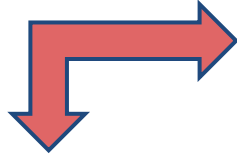


```
services {  
  distribution ARBL  
    switch s-ARBL3-1496-1 {  
      switchport GigabitEthernet1/0/1 {  
        description "V-CBLDGA-AP";  
        mode {  
          access {  
            vlan V-BLDGA-AP;  
          }  
        }  
      }  
      switchport GigabitEthernet1/0/2 {  
        ...  
      }  
    }  
}
```

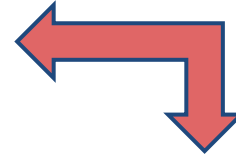
```
<interface xmlns="urn:ios">  
  <GigabitEthernet>  
    <name>1/0/1</name>  
    <description>V-CBLDGA-AP</description>  
    <switchport>  
      <mode>  
        <access/>  
      </mode>  
      <access>  
        <vlan>9</vlan>  
      </access>  
    </switchport>  
  </GigabitEthernet>  
</interface>
```



NSO device configuration model



NSO network element
driver(NED)
YANG model



```
<interface xmlns="urn:ios">
  <GigabitEthernet>
    <name>1/0/1</name>
    <description>V-CBLDGA-AP</description>
    <switchport>
      <mode>
        <access/>
      </mode>
      <access>
        <vlan>9</vlan>
      </access>
    </switchport>
  </GigabitEthernet>
</interface>
```

```
interface GigabitEthernet1/0/1
  description V-CBLDGA-AP
  switchport access vlan 9
  switchport mode access
```



UMnet Distribution Service

module: distribution

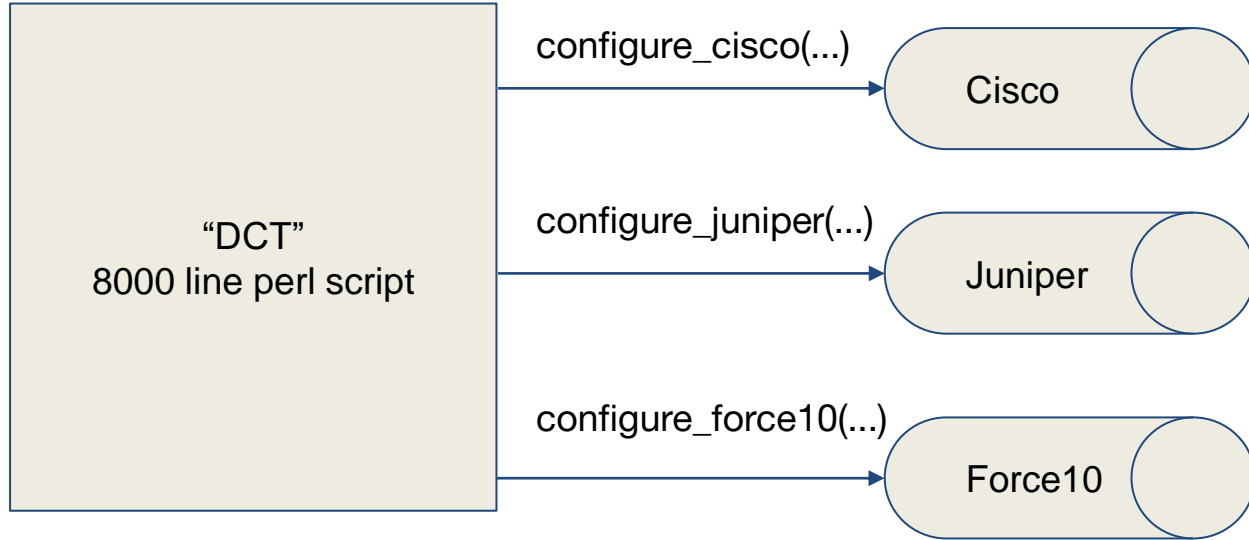
augment /ncs:services:

```
    +--rw distribution*
    +--rw name
    |
    |   +--rw routing!
    |   +--rw network* [name]
    |   | +--rw name
    |   +--rw primary-router
    |   | +--rw name
    |   | +--rw vtep-loopback
    |   | +--rw underlay-
loopback
    |   | +--rw core-uplinks
    |   | | +--rw p2p*
    |   | | +--rw name
    |   | | +--rw upstream
    |   ...
    +--rw switch*
    | +--rw switchport*
    | ...
```

- This is our primary service that represents building networks on our campus.
 - approx 180+ instances
- Each model instance represents a building network, and defines high-level configuration for the router(s) and switches in that building, such as:
 - links between devices, i.e. the physical topology within the building LAN as well as the connection to the core network
 - the networks (VLANs and IPv4/IPv6 prefixes) that are routed out of that building.



Building Software on NSO - Context



Building Software on NSO - Context (2)

“DCT”

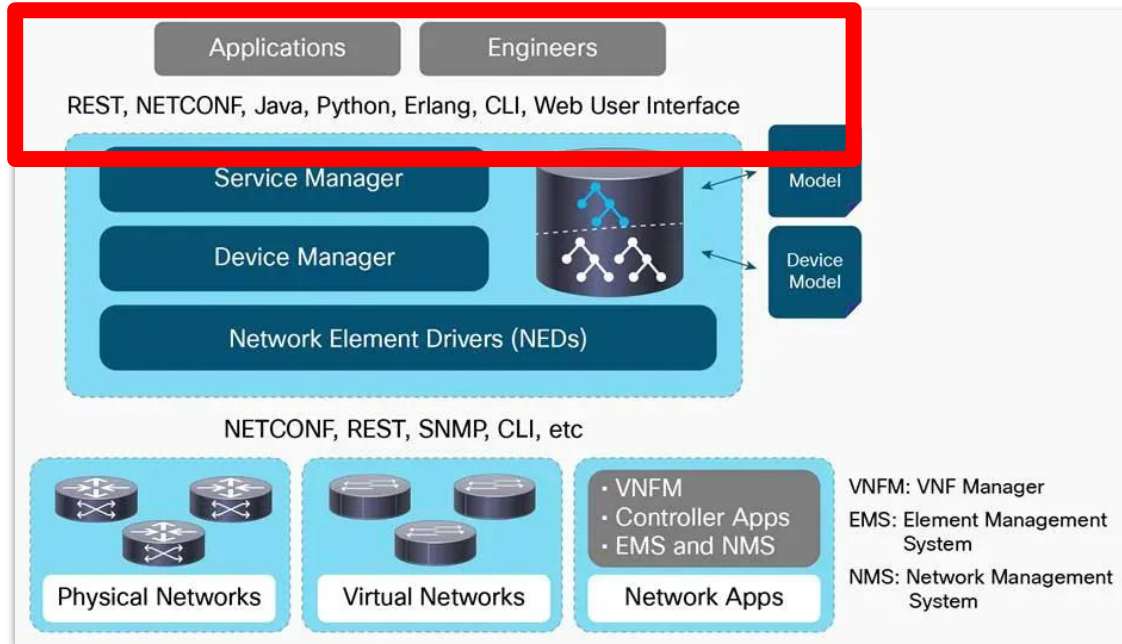
- Sends commands via SSH
- Retrieves data via SNMP
- Many branches & conditionals



```
my $cli = 'set cli screen-width 120'; # ARRRGGGGHHH!!!!
```

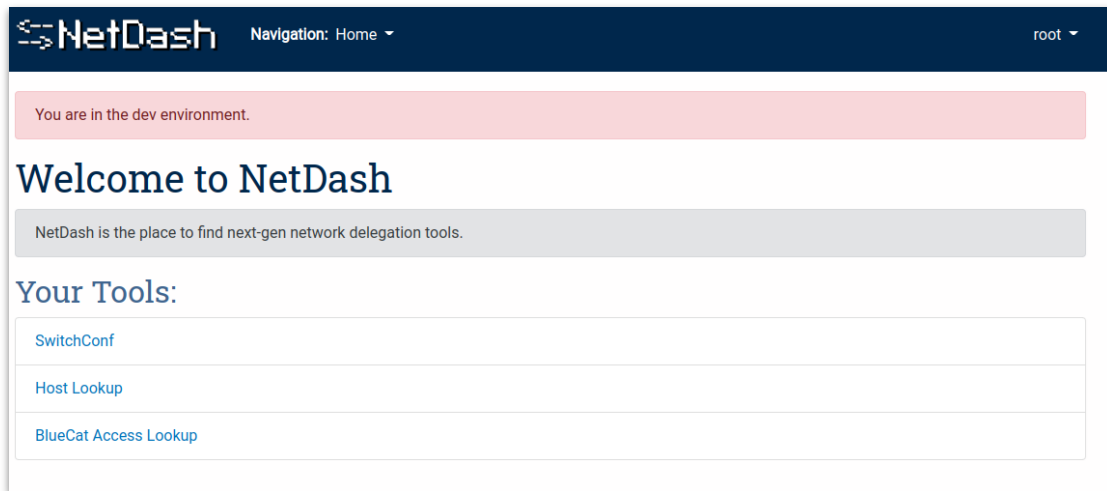
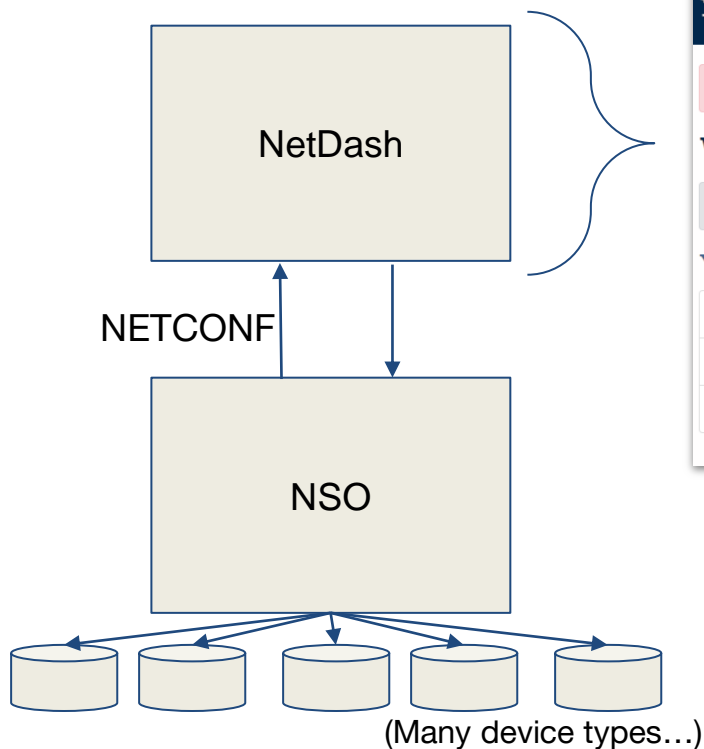


Building Software on NSO - Interfaces




- REST
- **NETCONF**
- Java
- Python
- Erlang
- CLI
- Web UI

Building Software - “NetDash”



- Uses Django (Python web framework)
- Much simpler codebase, targeting only “NSO” instead of dozens of device types. (Fewer code paths)

Building Software - NetDash SwitchConf

 Navigation: SwitchConf ▾ root ▾

You are in the dev environment.

SwitchConf

Lookup by end-user device...

<input type="text" value="MAC or IP Address"/>	<input type="text" value="Select Building..."/> ▾	<input type="submit" value="Submit"/>
--	---	---------------------------------------

-OR-

Lookup access layer switch...

<input type="text" value="Name or IP Address"/>	<input type="submit" value="Submit"/>
---	---------------------------------------



Building Software - NetDash SwitchConf (2)

NetDash

Navigation

root

You are in the dev environment.

Back

Edit

Edit Data

Description: my fun plantops thing

VLAN: 15: NGFW-LSA-GADGETS

Speed/Duplex: Unchanged

Admin Status: False

VoIP: Unchanged

Close Save changes

Port	Description	VLAN	Status	Oper Status	MAC Address		
Gi1/0/1	asdf	10: V-BLDGA-USER	✗	auto	auto	✓	✗
Gi1/0/2	my dumb plantops thing	15: NGFW-LSA-GADGETS	✗	auto	auto	✗	✗
Gi1/0/3	my dumb plantops thing	20: V-PO-ILAB	✓	auto	auto	✗	✗
Gi1/0/4	1422-03D	10: V-BLDGA-USER	✗	1000	full	✓	✗
Gi1/0/5	ccsame desc	10: V-BLDGA-USER	✗	auto	auto	✓	✗
Gi1/0/6	1430-01D	10: V-BLDGA-USER	✗	auto	auto	✓	✗



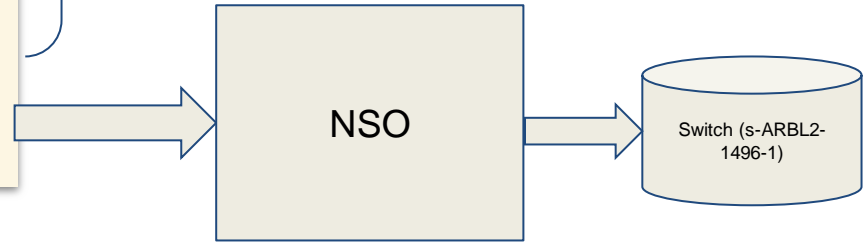
Building Software - NetDash SwitchConf (3)

```
<action xmlns="urn:ietf:params:xml:ns:yang:1">
  <services xmlns="http://tail-f.com/ns/ncs">
    <netsplash xmlns="http://umnet.umich.edu/netsplash">
      <update-interfaces xmlns="http://umnet.umich.edu/netsplash">

        <config xmlns="http://tail-f.com/ns/config/1.0">
          <services xmlns="http://tail-f.com/ns/ncs">
            <distribution xmlns="http://umnet.umich.edu/distribution">
              <name>ARBL</name>
              <switch>
                <name>s-ARBL2-1496-1</name>
                <switchport>
                  <name>ge-0/0/0</name>
                  <description>1410-03C</description>
                  <mode>
                    <access>
                      <vlan>NGFW-ITS-COMM-AL</vlan>
                    </access>
                  </mode>
                </switchport>
              </switch>
            </distribution>
          </services>
        </config>
      </update-interfaces>
    </netsplash>
  </services>
</action>
```

Envelope - NSO Action

Configuration Payload



Questions?

GÉANT learning resources:

- Network Automation eAcademy Index with map:
<https://wiki.geant.org/display/NETDEV/OAV+Training+Portal> (or
<https://e-academy.geant.org/moodle/course/index.php?categoryid=20>)
- YANG: <https://e-academy.geant.org/moodle/course/view.php?id=63>
- XML: <https://e-academy.geant.org/moodle/course/view.php?id=132>
- NSO: <https://e-academy.geant.org/moodle/course/view.php?id=114>
- NETCONF: <https://e-academy.geant.org/moodle/course/view.php?id=126>
- Introduction to APIs: <https://e-academy.geant.org/moodle/course/view.php?id=67>

