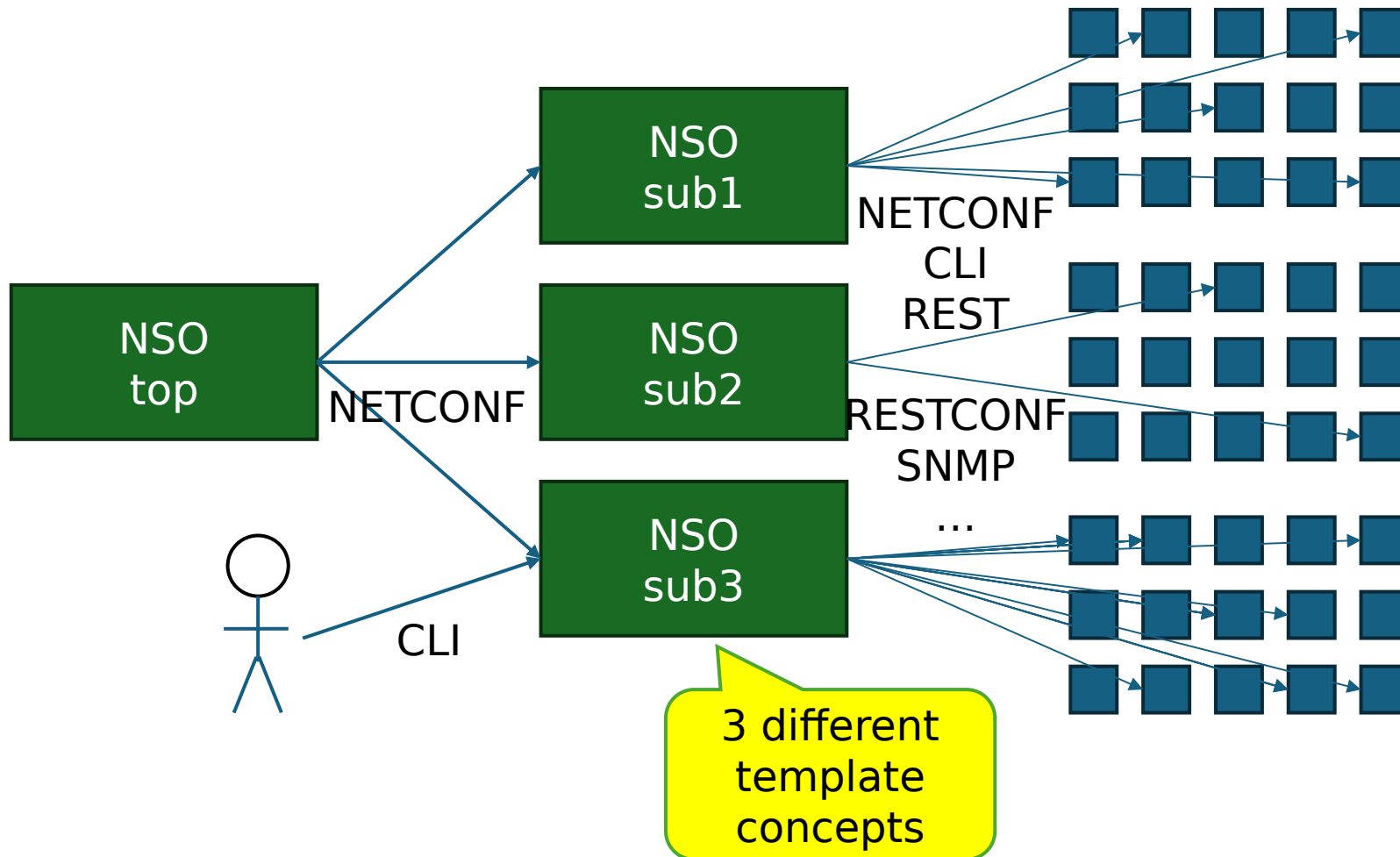


Templates in Cisco's Network Services Orchestrator (NSO)

IETF 121, Nov 2024

Jan Lindblad
jan.lindblad@protonmail.com

NSO is a **Client** (Orchestrator / Controller)



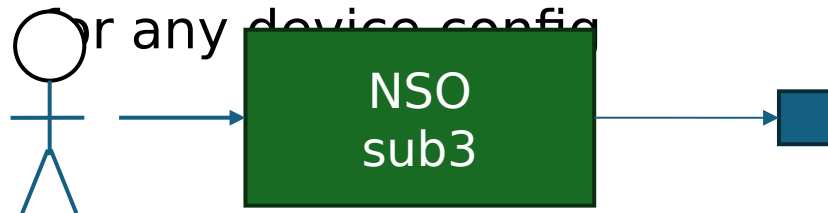
Many devices have some sort of "template" or "high-level" configuration interface.

Many NSO deployments use such functionality today.

Template Concept #1: Device Templates

Operator applies one or more templates to a specific device, then commits

- Each template handles multiple types, brands, versions of devices
- Template typically contains variables & XPath expressions
- Operator can create/update/delete templates



Makes handling of similar devices by different vendors and versions easier

```
# show full-configuration devices template
devices template snmp1
  ned-id cisco-ios-cli-3.8 config
    ios:snmp-server community {$COMMUNITY} RO

  ned-id cisco-iosxr-cli-3.5 config
    cisco-ios-xr:snmp-server community {$COMMUNITY} RO

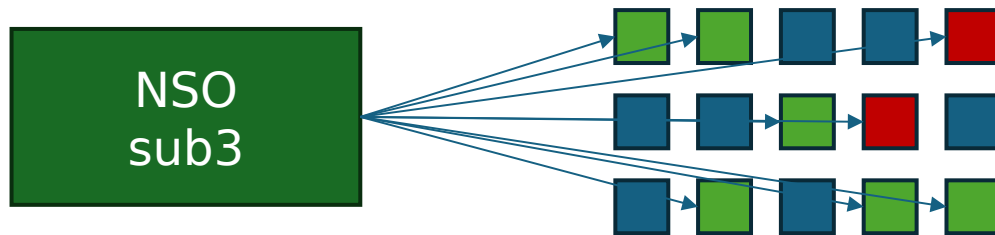
  ned-id juniper-junos-nc-3.0 config
    junos:configuration snmp community {$COMMUNITY}
    authorization read-only

# devices device ce2 apply-template template-name
snmp1 variable { name COMMUNITY value 'FUZBAR' }
```

Template Concept #2: Compliance Templates

Describes acceptable values across devices

- Each template handles multiple types, brands, versions of devices
- Template contains fixed values and regular expressions
- Operator can create/update/delete templates for any device config



Makes handling of similar devices by different vendors and versions easier

```
# compliance template internal-dns
# ned-id router-nc-1.0 config sys
dns server 10\\\\.+.

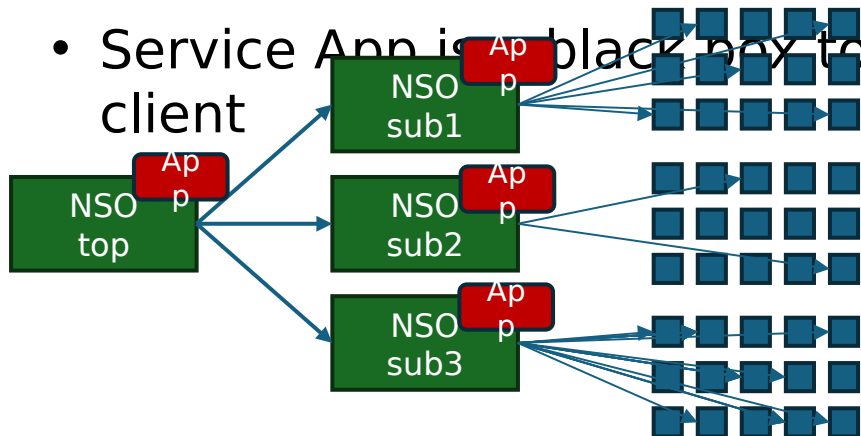
# compliance template internal-dns
# check device ex0
check-result {
  device ex0
  result violations
  diff config {
    sys {
      dns {
+       # after server 10.2.3.4
+       /* No match of 10\\\\.+. */
+       server 192.168.100.10;
      }
    }
  }
}
```

Template Concept #3a: Service Apps

Service App provides YANG-to-YANG mapping. Client configures intent, Service App translates to lower-level services and devices

- Each template works for multiple types, brands, versions of devices and services

- Service App is black box to client



Raise abstraction level, integrate with environment, life-cycle management

l3vpn.yang:

```
container vpn {
  list l3vpn {
    key name;
    leaf name { type string; }
    leaf as-number { type uint32; }

    list endpoint {
      key id;
      leaf id { type string; }
      leaf ce-device {
        type leafref {
          path "/ncs:devices/ncs:device/ncs:name";
        }
      }
      leaf ce-interface { ... }
      leaf ip-network { type inet:ip-prefix; }
      leaf bandwidth { type uint32; }
    }
  }
  container qos {
    leaf qos-policy {
      type leafref {
        path "/l3vpn:qos/qos-policy/name";
      }
    }
  }
}
```

...

NSO is a **Client** ... but also a Server

