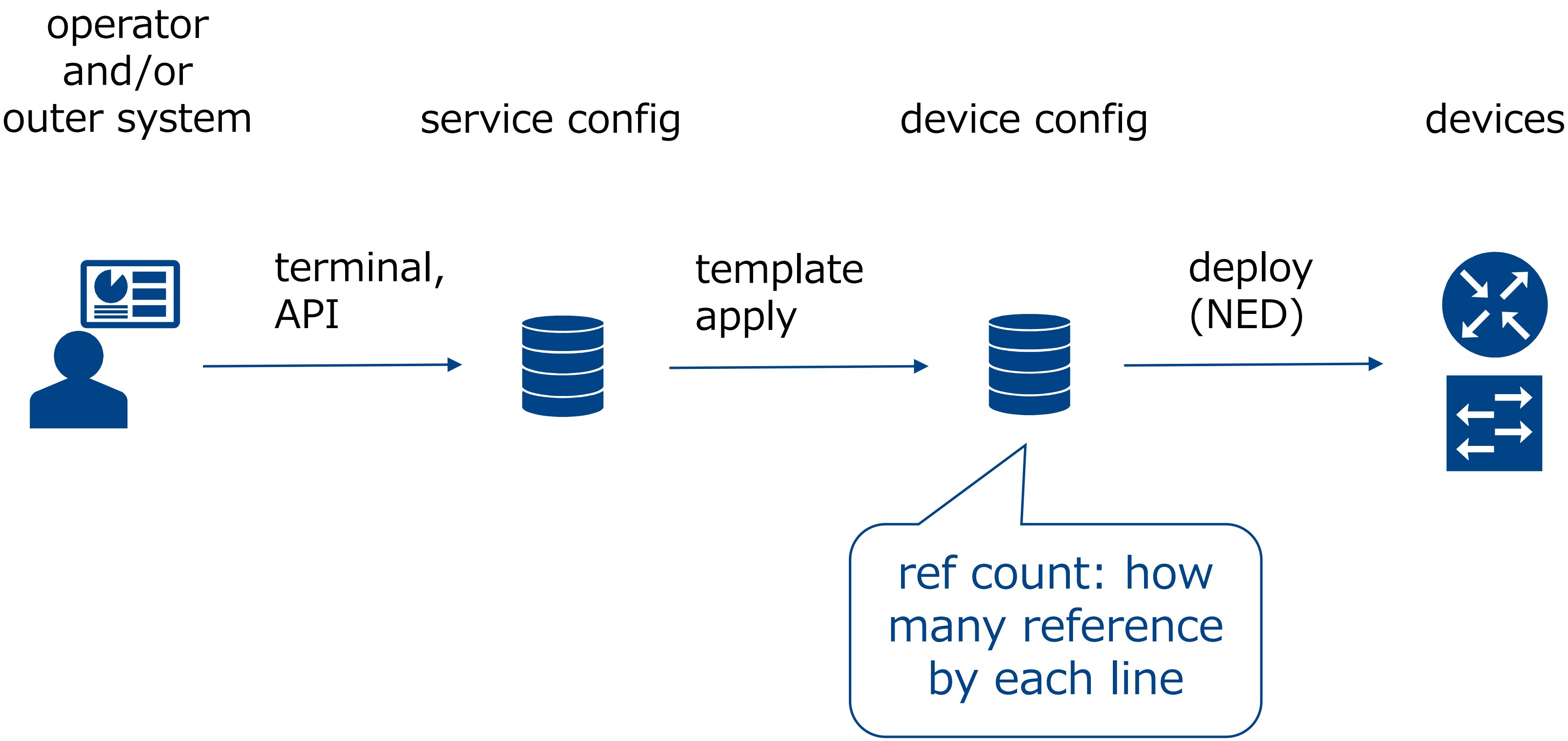
Live together,
NSO and manual operations
in multi-vendor "growing brownfield"

NSO Developer Days 2020
@virtual, 17 Jun 2020

Teruhisa Tajima
H.Okui, F.Morifuji, S.Kimura
NTT Communications

Making Config in NSO



operator
and/or
outer system

service config

device config

devices

terminal,
API

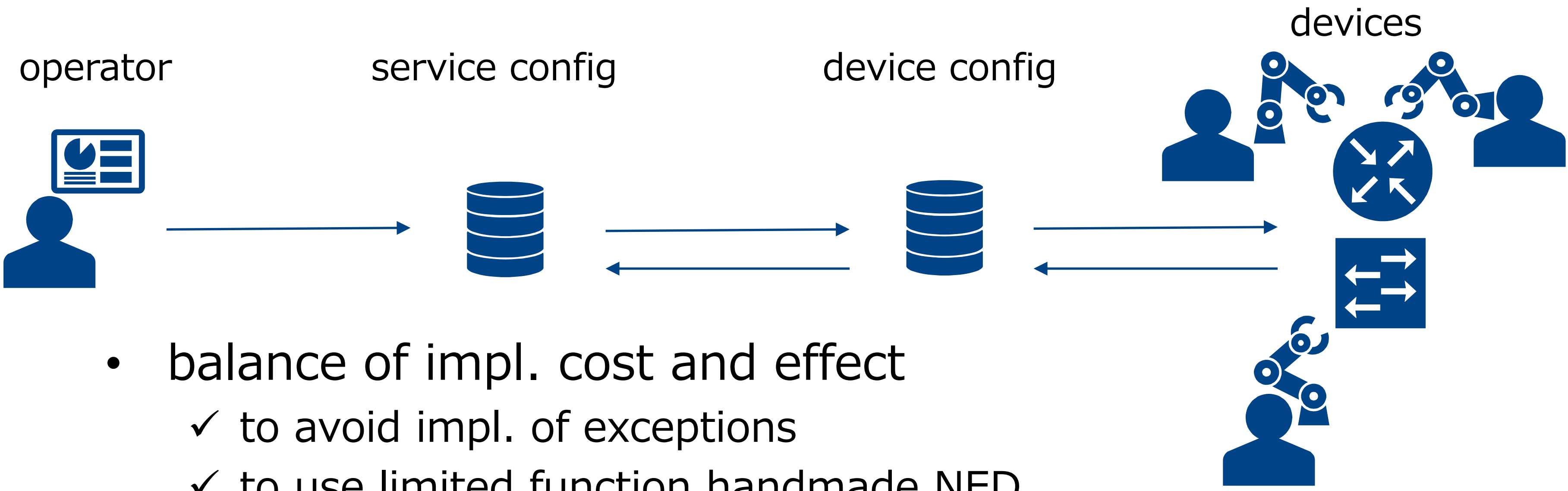
template
apply

deploy
(NED)

ref count: how many reference by each line

Our Usage of NSO

■ changing config outside of NSO



operator

service config

device config

devices

- balance of impl. cost and effect
  - ✓ to avoid impl. of exceptions
  - ✓ to use limited function handmade NED
- training of device operation, NSO development

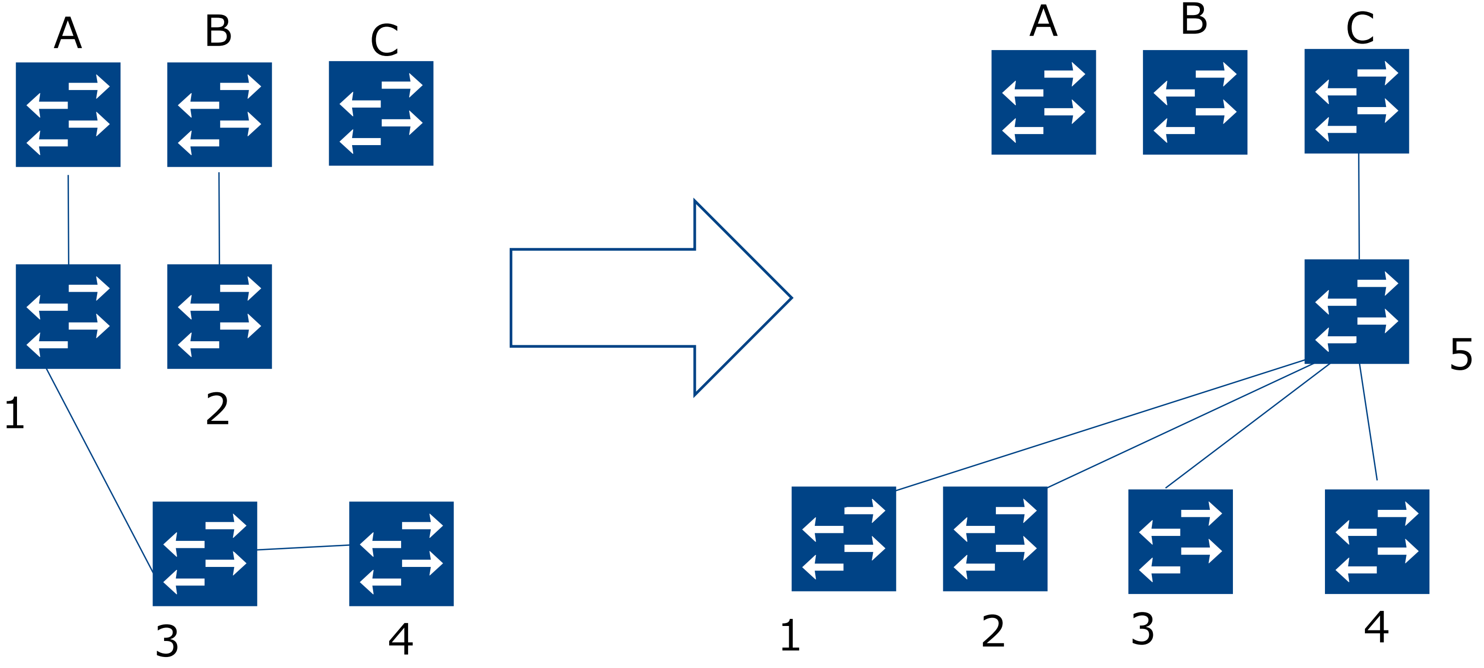
Our Key Benefits

- implement multi-vendor topology model using augment
  → can change topology after deployed NSO

- control ref-count using redeploy and reconcile
  → can do both manual config and NSO operation

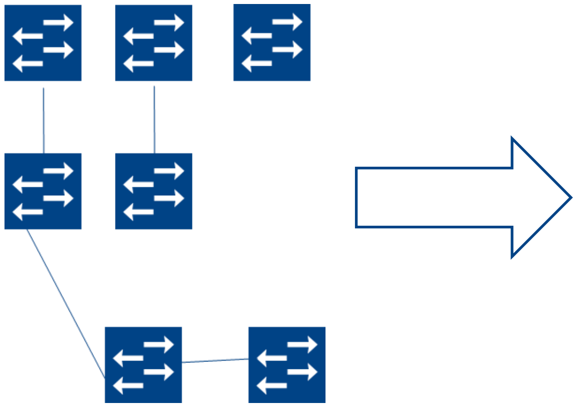
Our Use Case

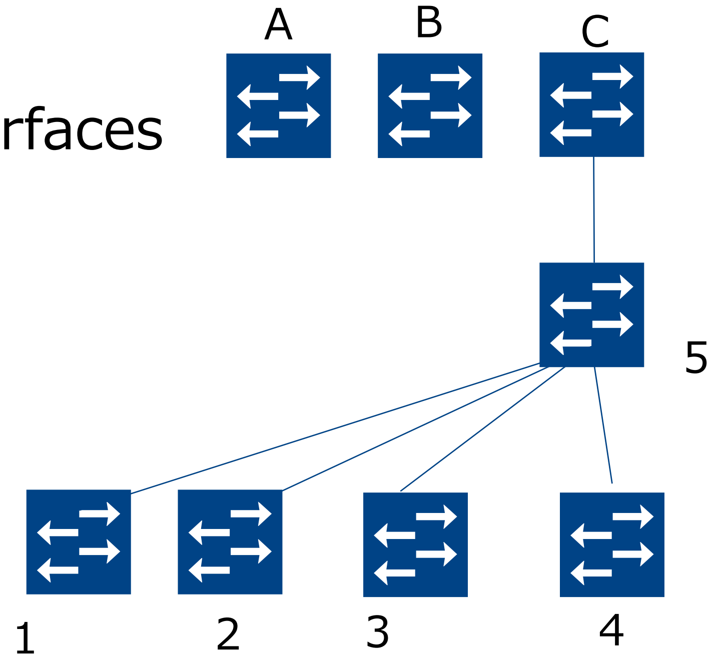
- change topology drastically
  - due to circuit outage, increase bandwidth, etc.



Operation Steps

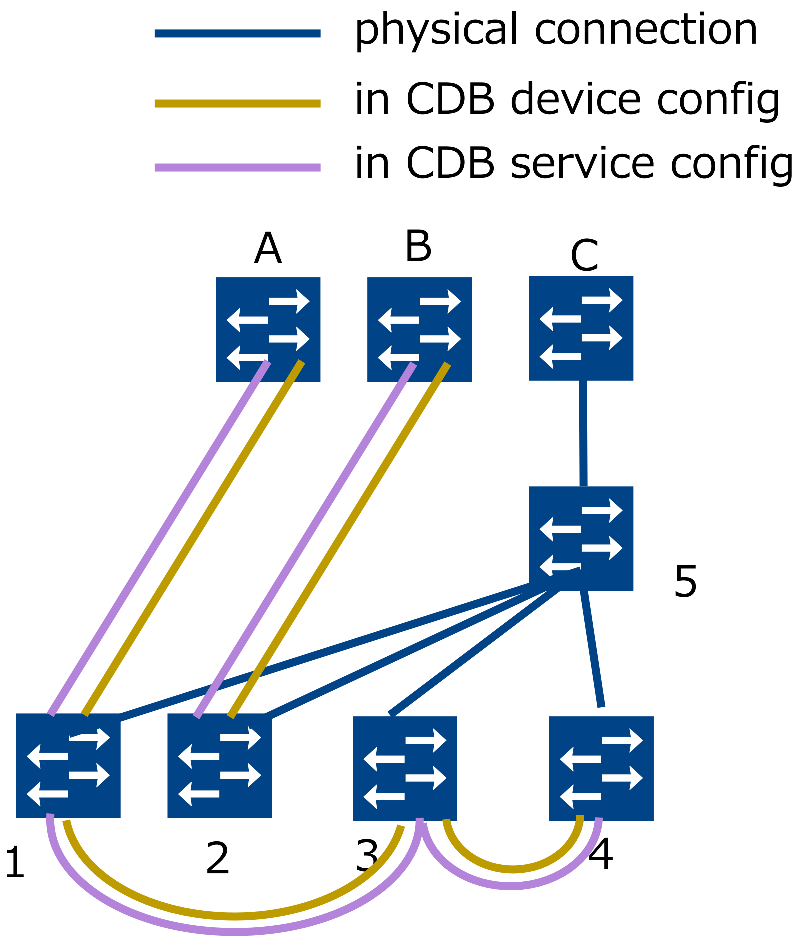
1. pre-configuration of SW C and 5
2. change cable connections
3. post-configuration in NSO
    - clear interconnection interfaces
    - set ref-count





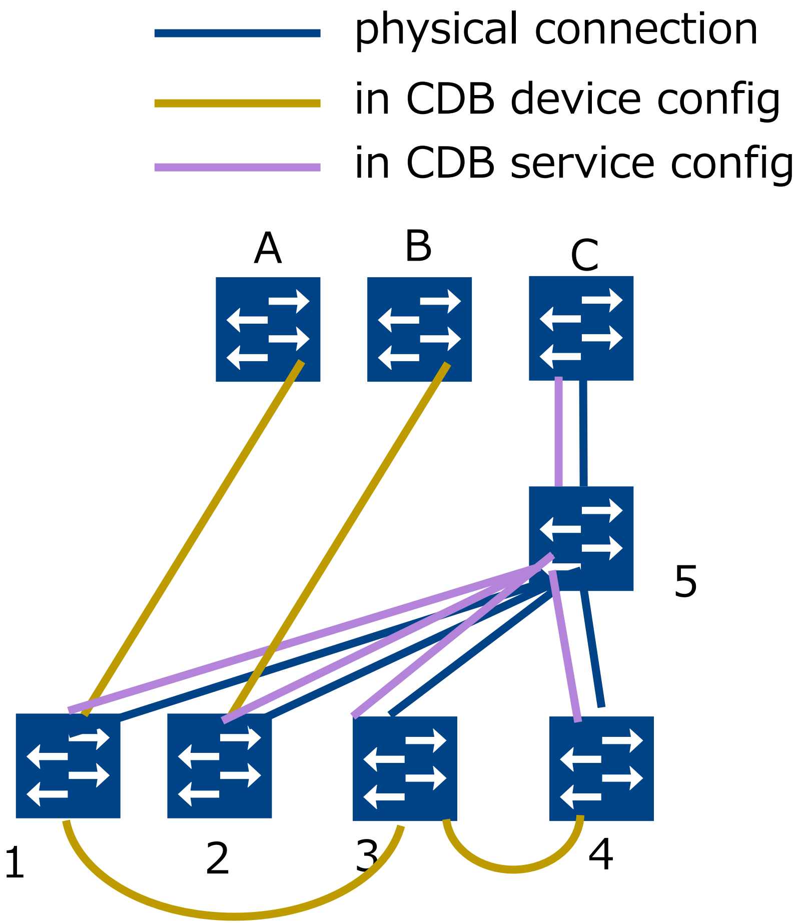
3. Post-Config = NSO Maintenance

- need to sync connection
- sync step
  - modify service config
  - redeploy (no-networking)
  - sync-from
  - service discovery
  - redeploy
  - reconcile



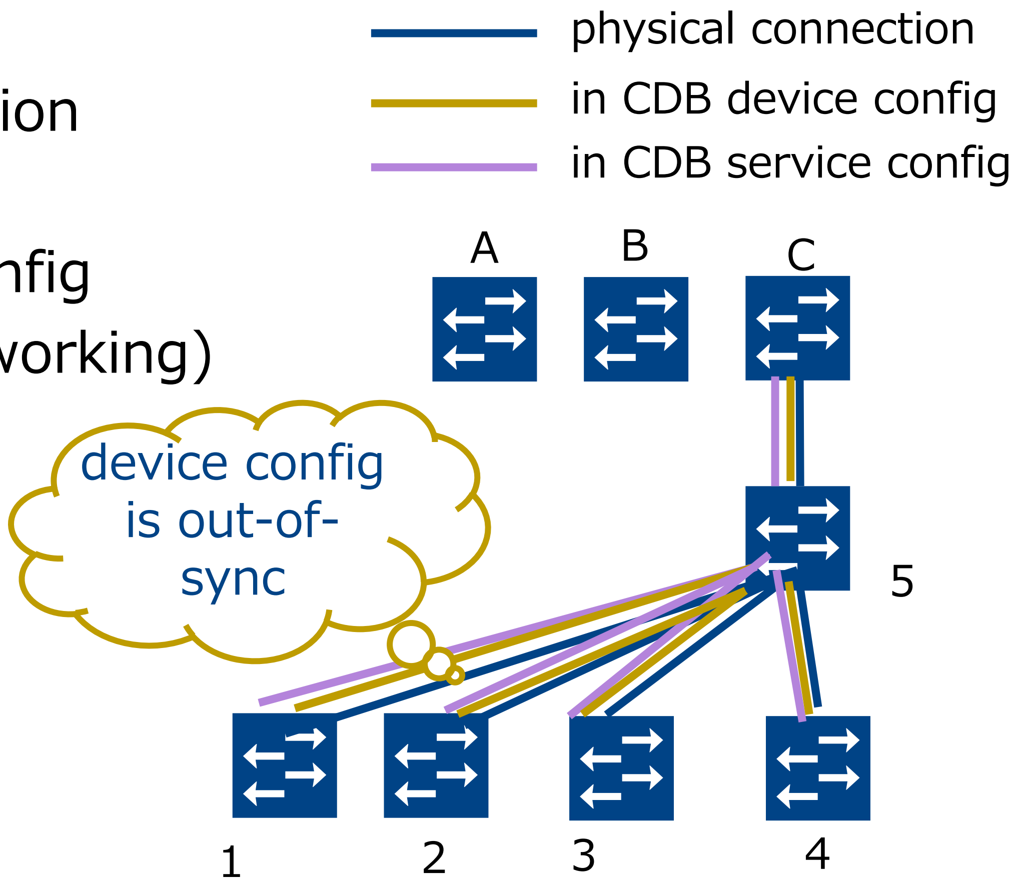
3. Post-Config = NSO Maintenance

- need to sync connection
- sync step
  - modify service config
  - redeploy (no-networking)
  - sync-from
  - service discovery
  - redeploy
  - reconcile



3. Post-Config = NSO Maintenance

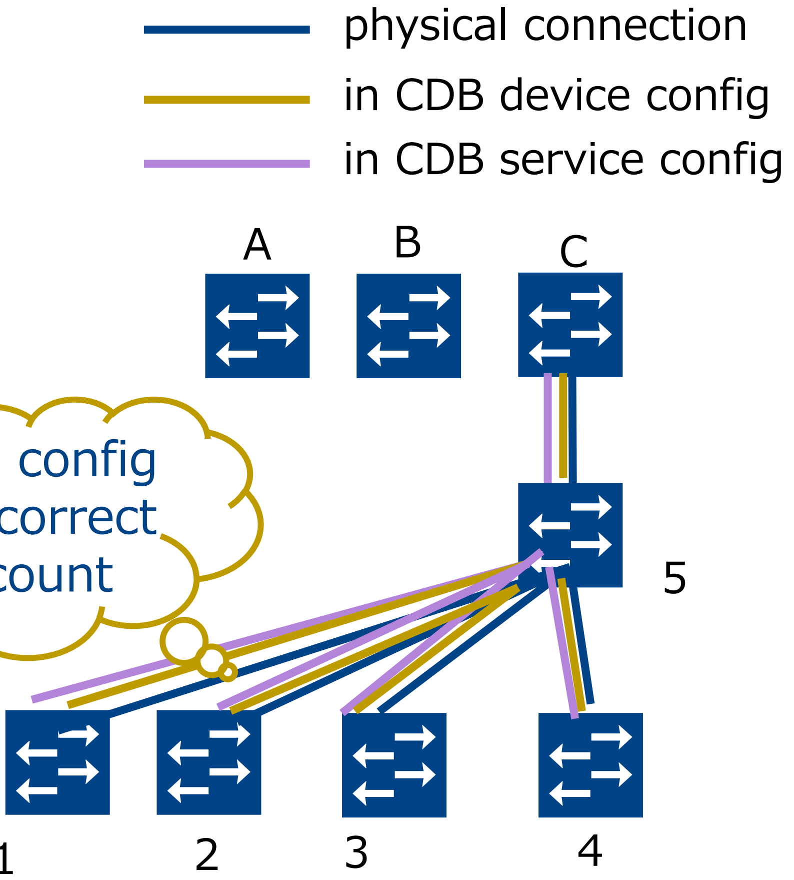
- need to sync connection
- sync step
  - modify service config
  - ⇨ redeploy (no-networking)
  - sync-from
  - service discovery
  - redeploy
  - reconcile



3. Post-Config = NSO Maintenance

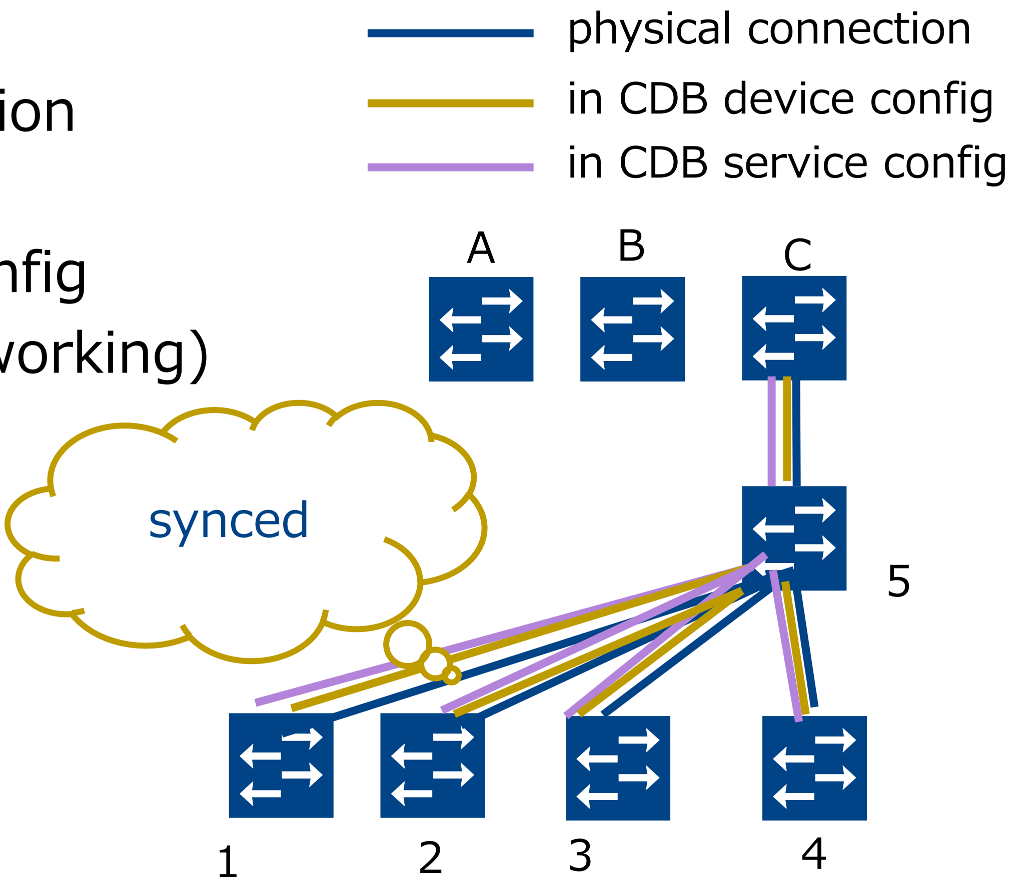
- need to sync connection
- sync step
  - modify service config
  - redeploy (no-networking)
  - ⇨ sync-from
  - service discovery
  - redeploy
  - reconcile

device config has incorrect ref-count



3. Post-Config = NSO Maintenance

- need to sync connection
- sync step
  - modify service config
  - redeploy (no-networking)
  - sync-from
  - ⇨ service discovery
  - ⇨ redeploy
  - ⇨ reconcile



Approach

- service model implementation
  - topology model
  - multi-vendor model using augment
- sync configuration between NSO <> devices
  - = how to resolve inconsistency
  - service discovery
  - redeploy & reconcile

Approach

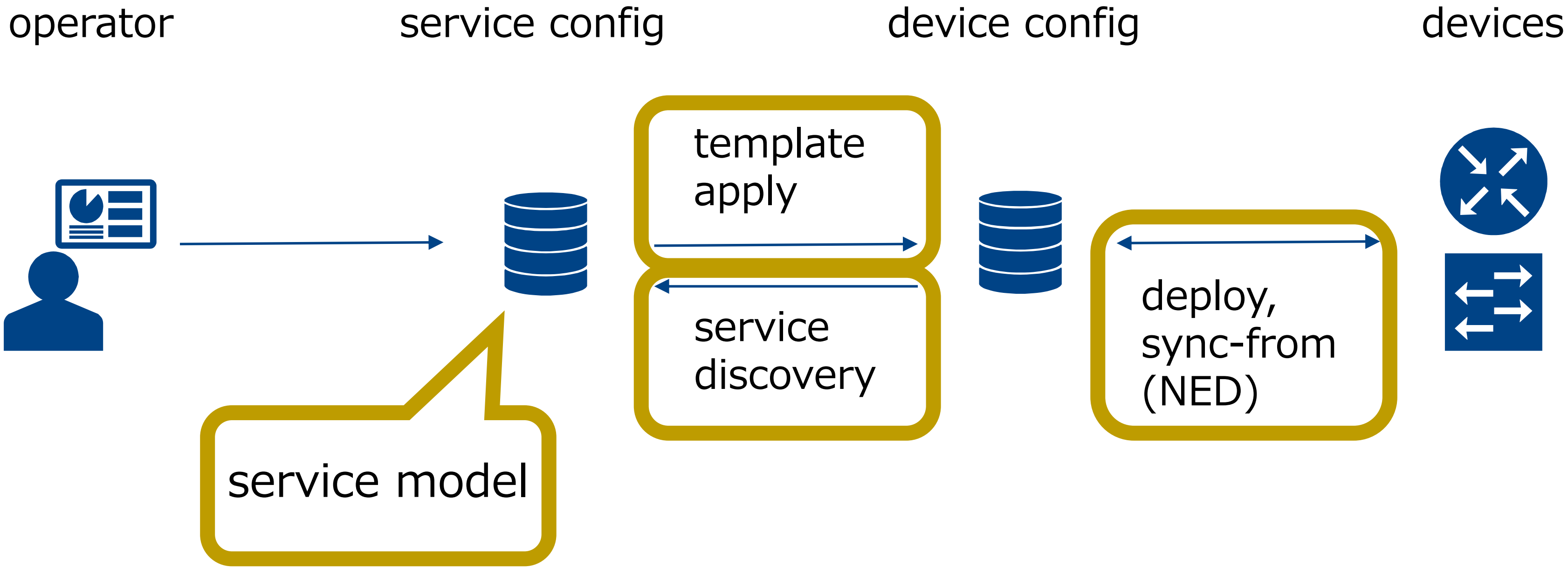
- ■ service model implementation
  - topology model
  - multi-vendor model using augment
- ■ sync configuration between NSO <> devices
  - = how to resolve inconsistency
  - service discovery
  - redeploy & reconcile

Our Lab's Objectives, Service and Specs

- Test new feature, (inter-)operability, performance, etc.
  - "stable lab NW" is NOT our goal
  - our lab NW is DUT NW in sometimes
- NW's main service: L2 VLAN in metro area
  - some packets through tunnel(L2VPN)
- 100+ SWs/RTs
  - AlaxalA
  - Brocade
  - Cisco
  - Juniper

Components to Develop Service

- service config -> device config -> deploy

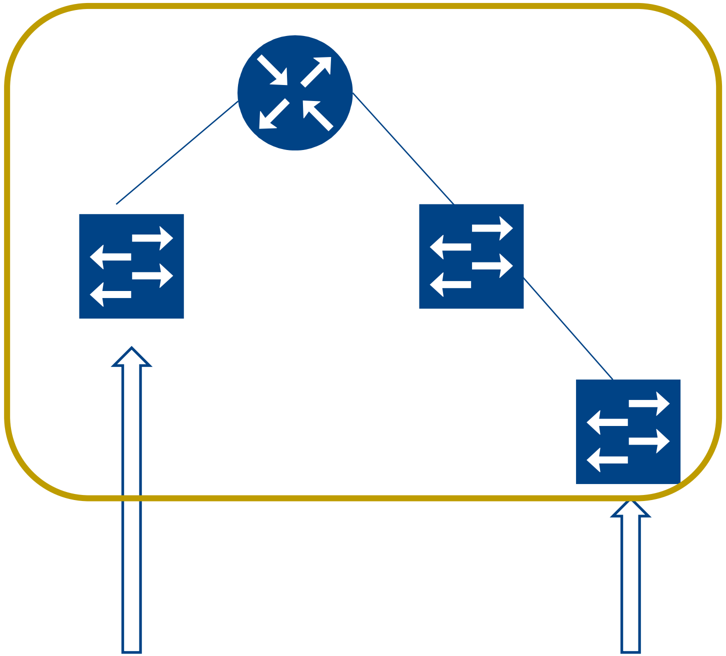


How to Calc Path (1)

- physical connections

```
list testbed-physical-connection {
   key name;
   leaf name {
     tailf:info "Unique service id";
     type string;
   }
   container down-side {
     leaf device-type {
       type common:device-type-down-side;
     }
   }
   container up-side {
     leaf device-type {
       type common:device-typ
     }
   }
   leaf disabled {
     type empty;
   }
 }
```

```
typedef device-type {
   type enumeration {
     enum alaxala;
     enum junos;
     enum ios;
     enum brocade;
```



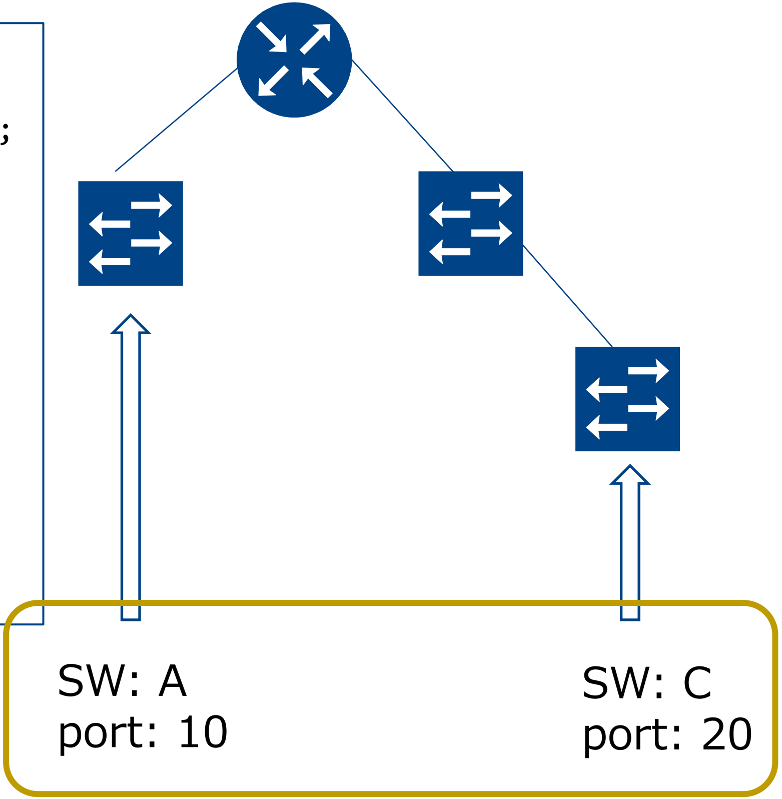
SW: A
port: 10

SW: C
port: 20

How to Calc Path (2)

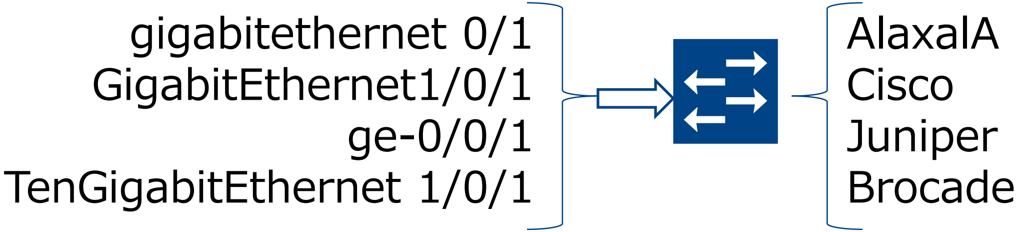
- endpoints := interface ports

```
list testbed-e2e-service {
   uses ncs:service-data;
   ncs:servicepoint testbed-e2e-service-servicepoint;

   key vlan;
   leaf vlan {
     mandatory true;
     type uint16 {
       range "1 .. 4094";
     }
   }

   container endpoints {
     // for augment
   }
```



How to Define Multi-Vendor Model

- equals to how to describe this one port in yang model



gigabitethernet 0/1
GigabitEthernet1/0/1
ge-0/0/1
TenGigabitEthernet 1/0/1

AlaxalA
Cisco
Juniper
Brocade

- patterns
  - augment
  - refer each vendor model directly
  - translate abstract model to vendor model

Augment

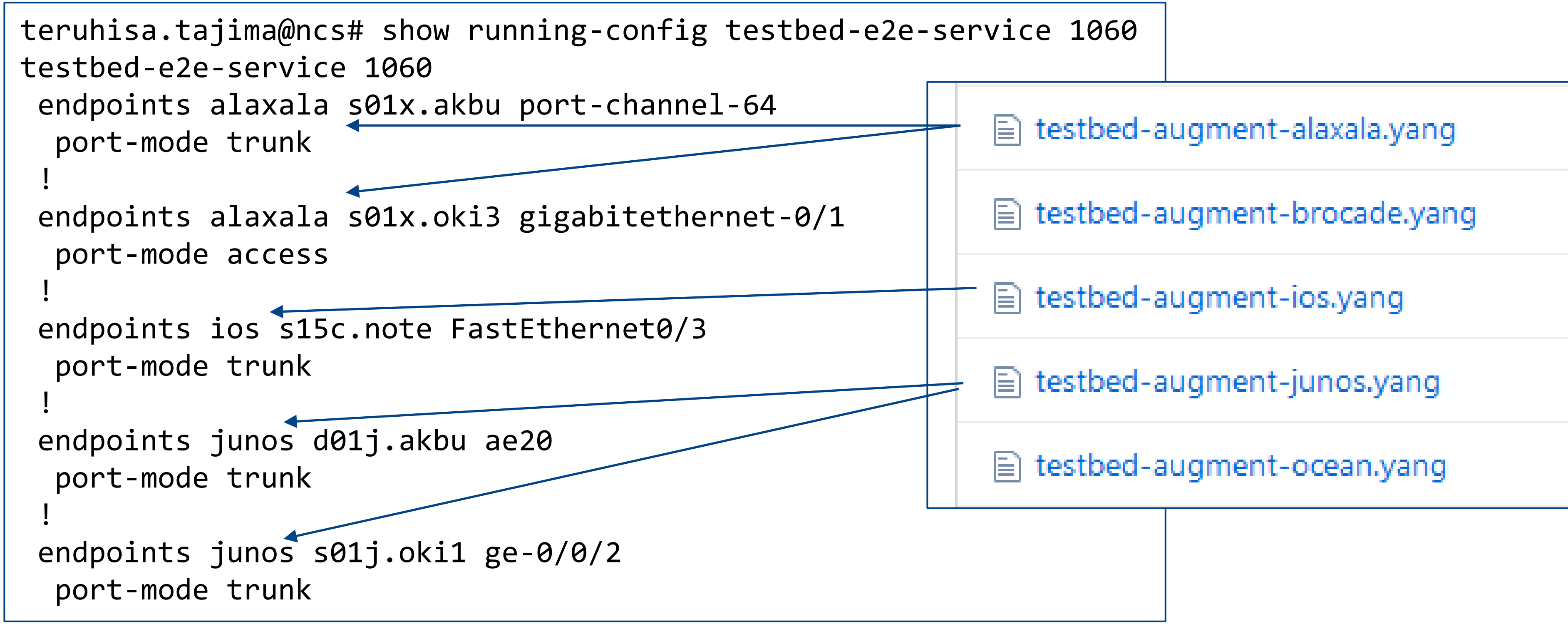
- inject from each vendor model
- describe as each list-name and leaf-name

```
container endpoints {
  // for augment
}
```

```
augment "/e2e:testbed-e2e-service/e2e:endpoints" {
  uses ios-endpoints;
}
grouping ios-endpoints {
  list ios {
    key "device interface";
    leaf device { type leafref {...} }
    leaf interface {
      type leafref {
        path "deref(../device)...
```

- pros: get cli suggest(complement), validation , can add model as another yang file when add new vendor
- cons: need to learn augment (but easy :-)

Example of Augment



```
teruhisa.tajima@ncs# show running-config testbed-e2e-service 1060
testbed-e2e-service 1060
 endpoints alaxala s01x.akbu port-channel-64
  port-mode trunk
 !
 endpoints alaxala s01x.oki3 gigabitethernet-0/1
  port-mode access
 !
 endpoints ios s15c.note FastEthernet0/3
  port-mode trunk
 !
 endpoints junos d01j.akbu ae20
  port-mode trunk
 !
 endpoints junos s01j.oki1 ge-0/0/2
  port-mode trunk
```

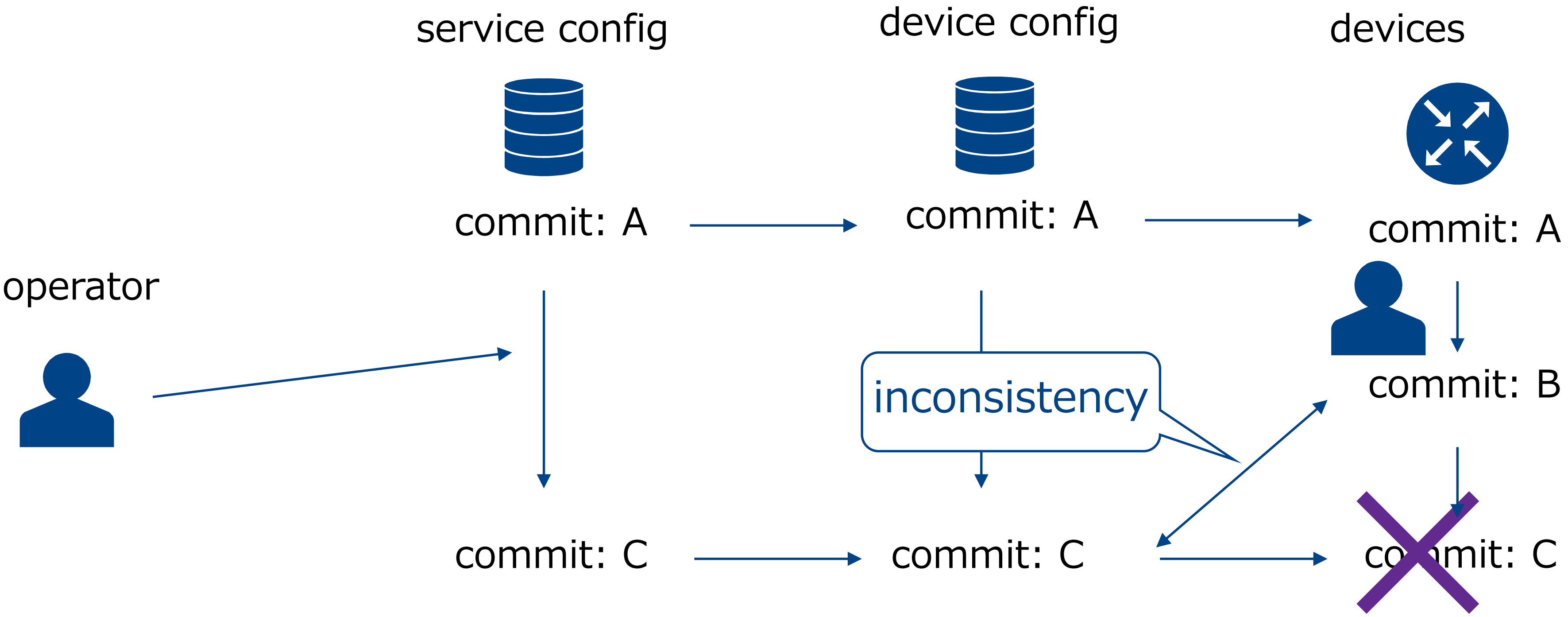
- testbed-augment-alaxala.yang
- testbed-augment-brocade.yang
- testbed-augment-ios.yang
- testbed-augment-junos.yang
- testbed-augment-ocean.yang

Approach

- service model implementation
  - topology model
  - multi-vendor model using augment
- sync configuration between NSO <> devices
  - = how to resolve inconsistency
  - service discovery
  - redeploy & reconcile

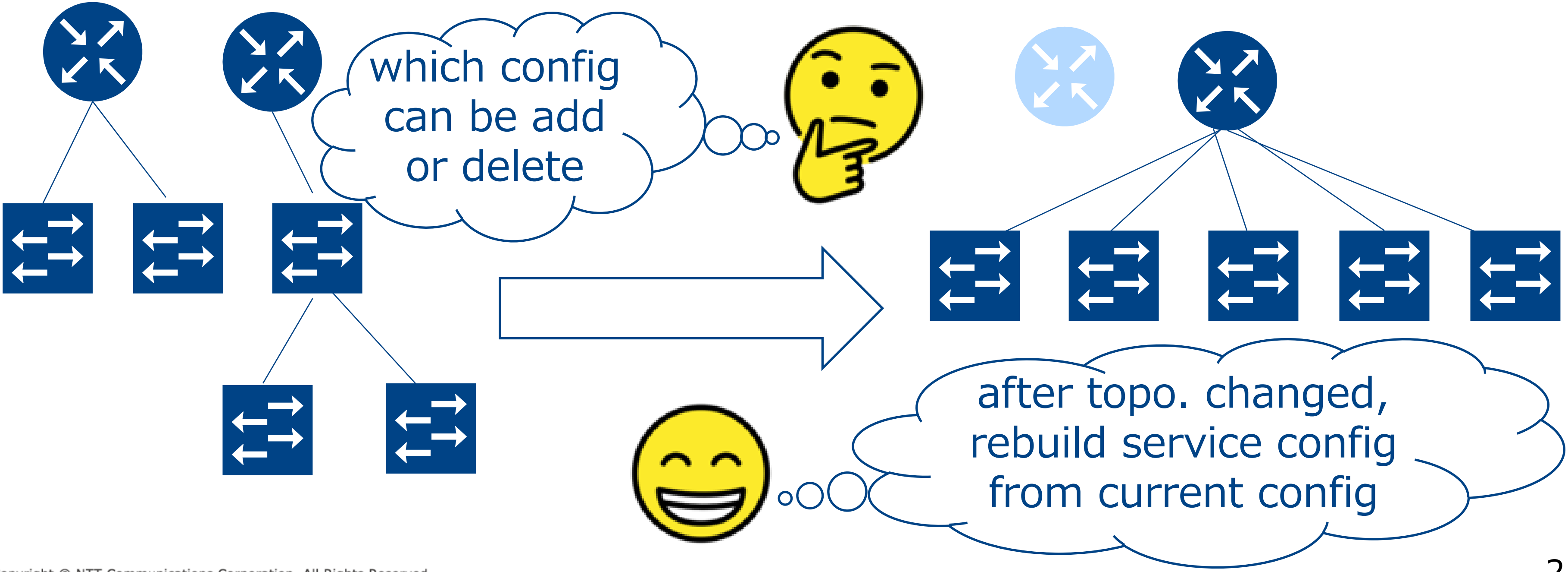
If Change Device Config Manually

- need to solve inconsistency



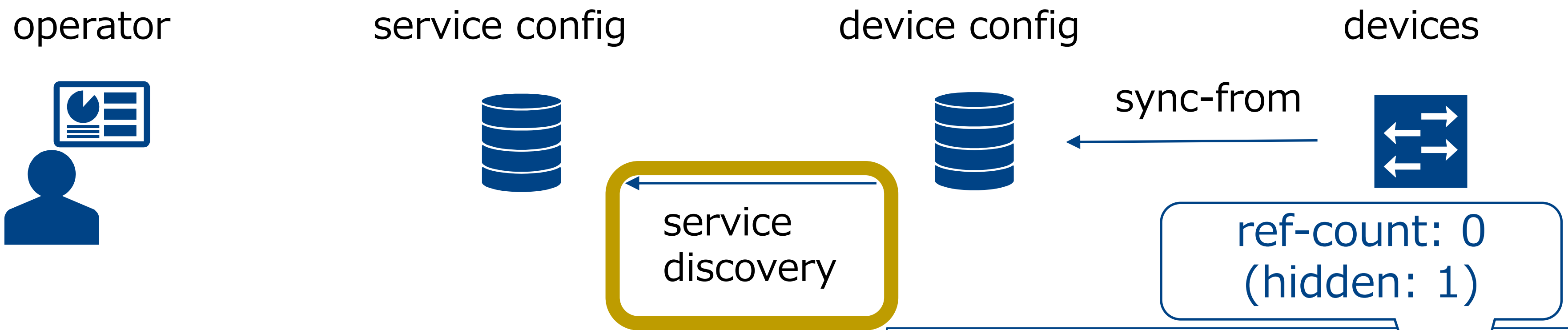
If Change Topology Drastically

- hard to get diff between before/after topology
  - difficult to know what configuration to set


which config can be add or delete
after topo. changed, rebuild service config from current config

Service Discovery

- implement service discovery as actions

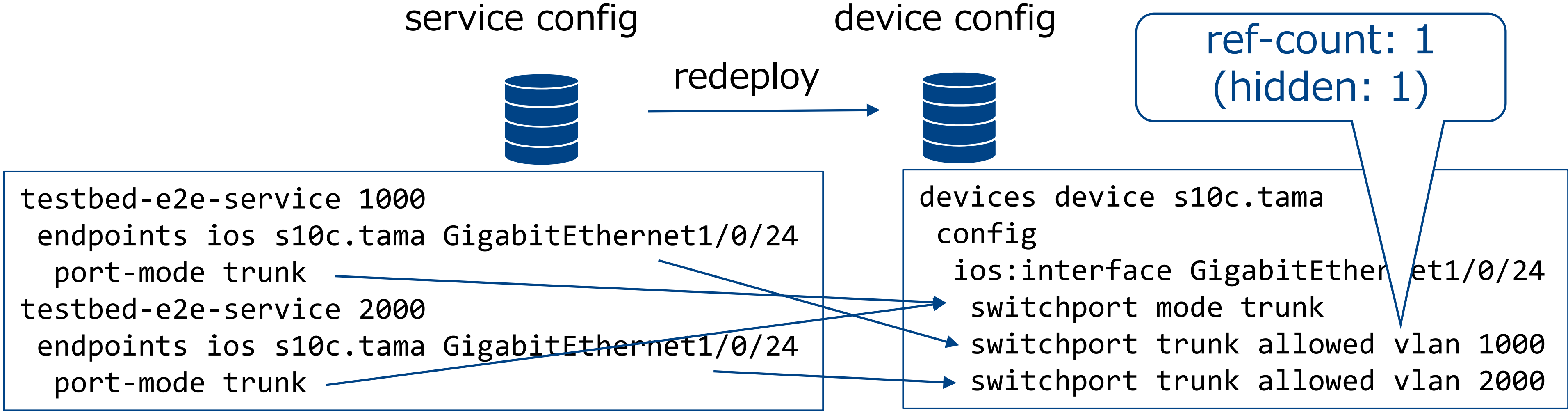


```
testbed-e2e-service 1000
 endpoints ios s10c.tama GigabitEthernet1/0/24
  port-mode trunk
testbed-e2e-service 2000
 endpoints ios s10c.tama GigabitEthernet1/0/24
  port-mode trunk
```

```
devices device s10c.tama
 config
  ios:interface GigabitEthernet1/0/24
   switchport mode trunk
   switchport trunk allowed vlan 1000
   switchport trunk allowed vlan 2000
```

Redeploy & Reconcile (set ref-count)

- redeploy
  - make relationships between service and device config
  - each configs are independent if only service discovery

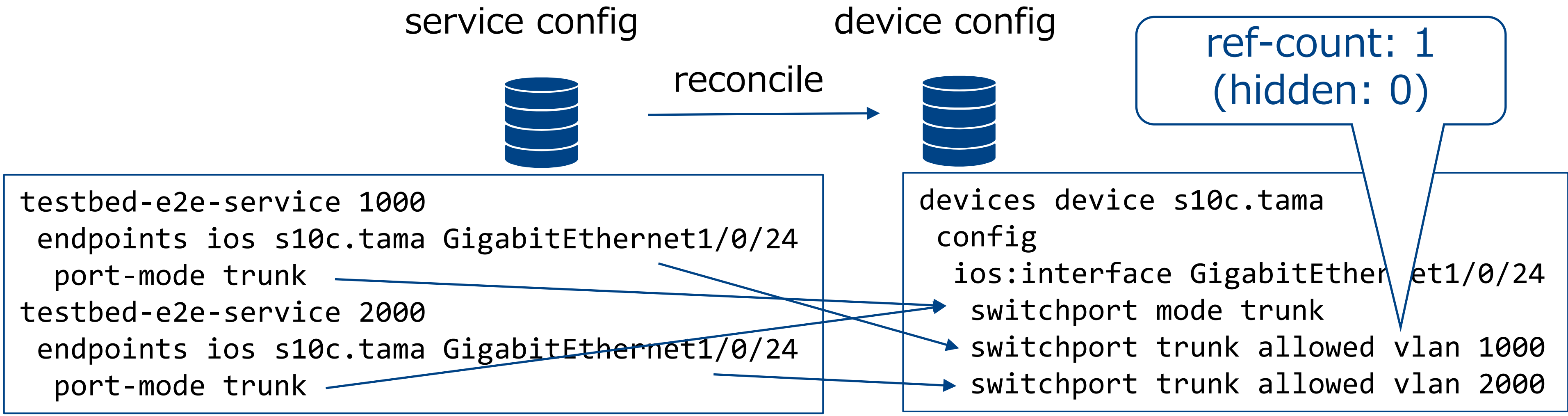


```
testbed-e2e-service 1000
 endpoints ios s10c.tama GigabitEthernet1/0/24
  port-mode trunk
testbed-e2e-service 2000
 endpoints ios s10c.tama GigabitEthernet1/0/24
  port-mode trunk
```

```
devices device s10c.tama
 config
  ios:interface GigabitEthernet1/0/24
   switchport mode trunk
   switchport trunk allowed vlan 1000
   switchport trunk allowed vlan 2000
```

Redeploy & Reconcile (set ref-count)

- reconcile
  - clear relationships which made by sync-from
  - after reconcile, same status as deployed by NSO



```
testbed-e2e-service 1000
 endpoints ios s10c.tama GigabitEthernet1/0/24
  port-mode trunk
testbed-e2e-service 2000
 endpoints ios s10c.tama GigabitEthernet1/0/24
  port-mode trunk
```

```
devices device s10c.tama
 config
  ios:interface GigabitEthernet1/0/24
   switchport mode trunk
   switchport trunk allowed vlan 1000
   switchport trunk allowed vlan 2000
```

Other Methods

- use partial-sync-to/from?
  - → definition of partial region is difficult
    if do partial-sync-from, also need to same discovery, redeploy and reconcile
  - → sync-from whole config
- force to using NSO at all?
  - → a few operations are too difficult to impl. in NSO
  - → the most cases: operated by NSO
    and a few cases: operated by hand (and sync NSO)

Conclusion

- implement multi-vendor topology model using augment
  - can add new device, change topology easily
- control ref-count using redeploy and reconcile
  - operate by hands and by NSO at same time
  - avoid implementing difficult and rare situation