



OpenConfigを用いたネットワーク機器操作

日本電信電話株式会社
ソフトウェアイノベーションセンタ
市原 裕史

自己紹介



市原 裕史

• 所属

- NTT SIC 第三推進プロジェクト

• **OpenStack Developer**

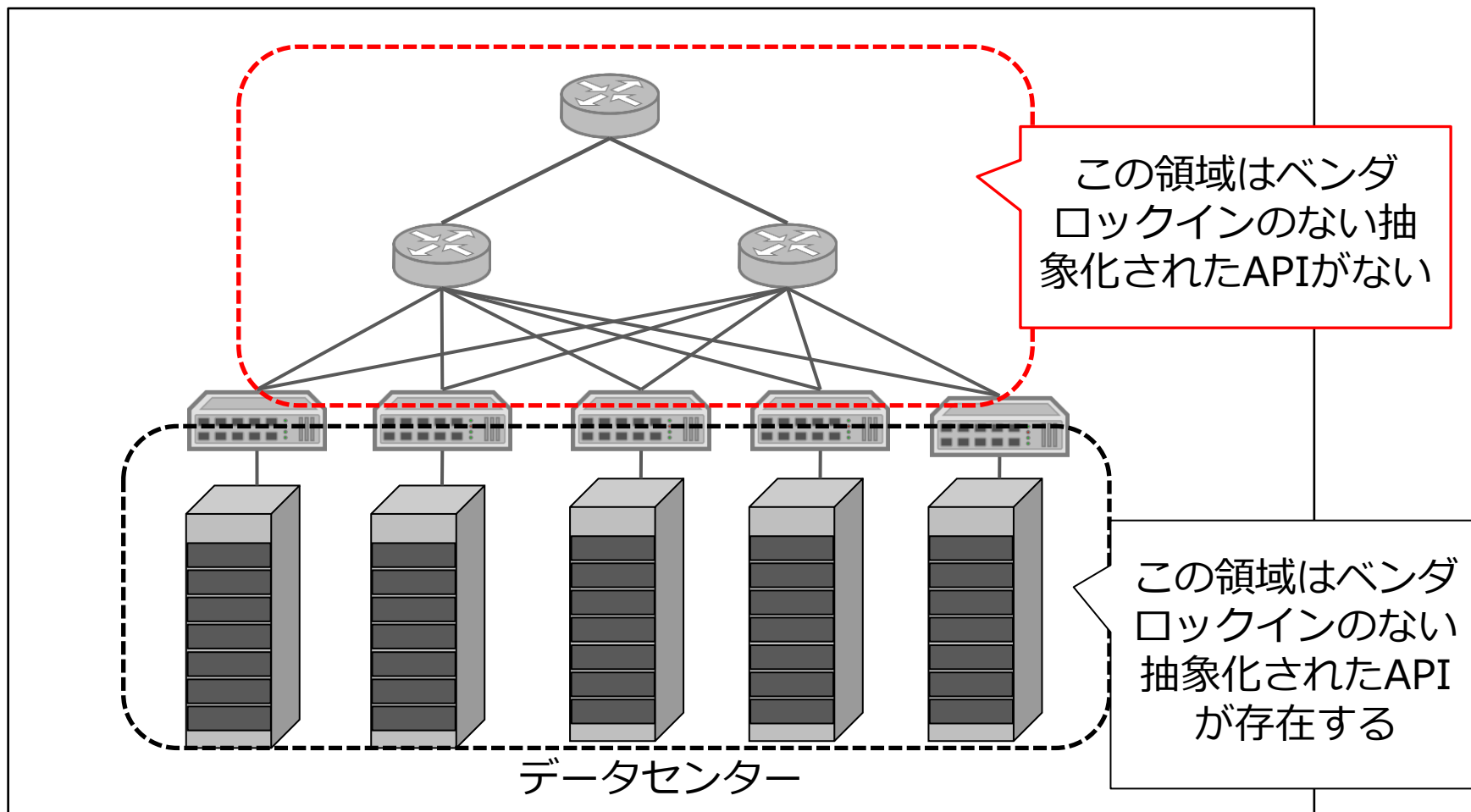
- Neutron Core Reviewer、その他プロジェクトへの貢献

• **技術: SDN/NFV中心**

- パブリッククラウドの開発
- LinuxconでDPDKの性能の発表
- Neutronへの機能提案



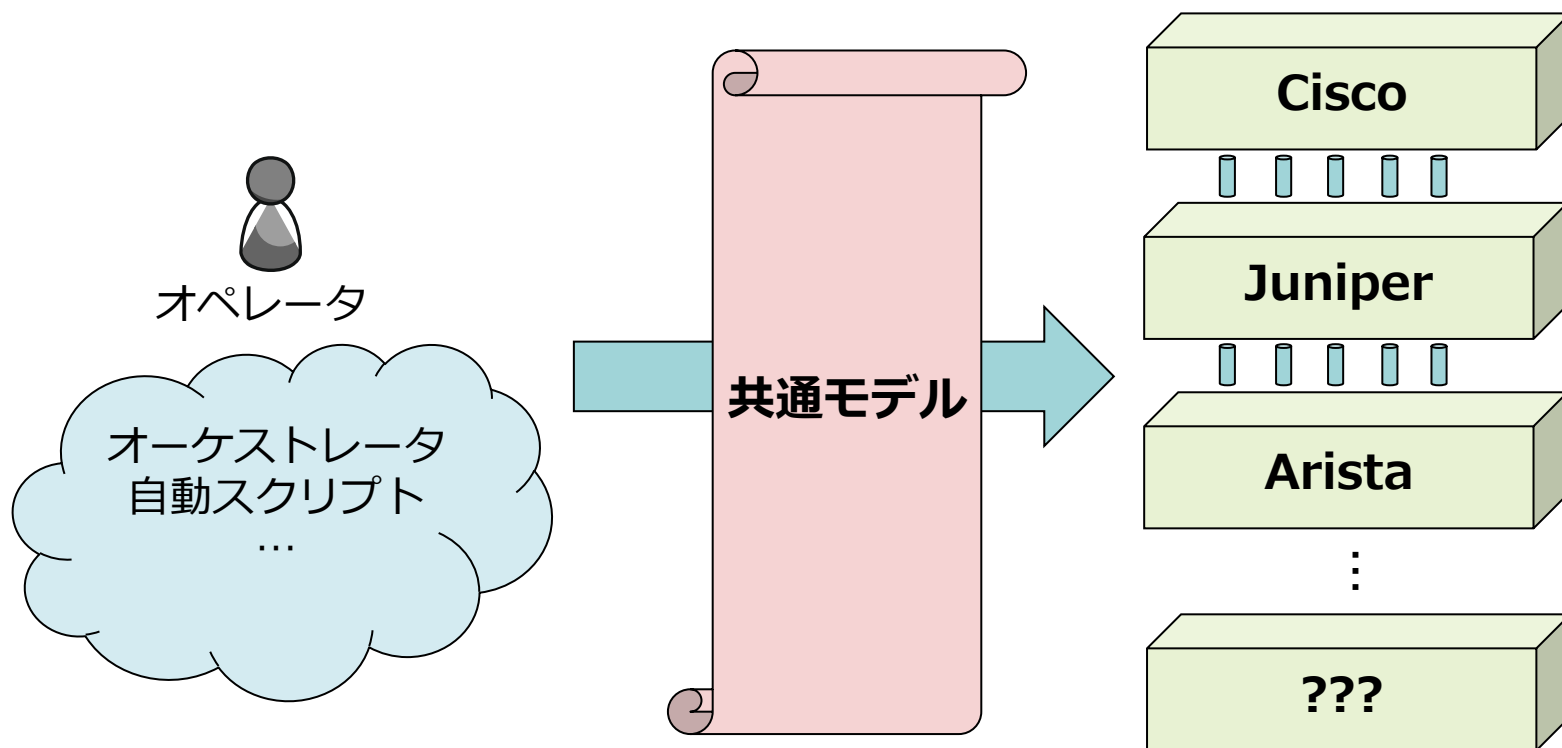
OpenConfigへのモチベーション



OpenConfig



- ネットワーク機器のAPIにベンダニュートラルなモデルを提供することを目的としたワーキンググループ名



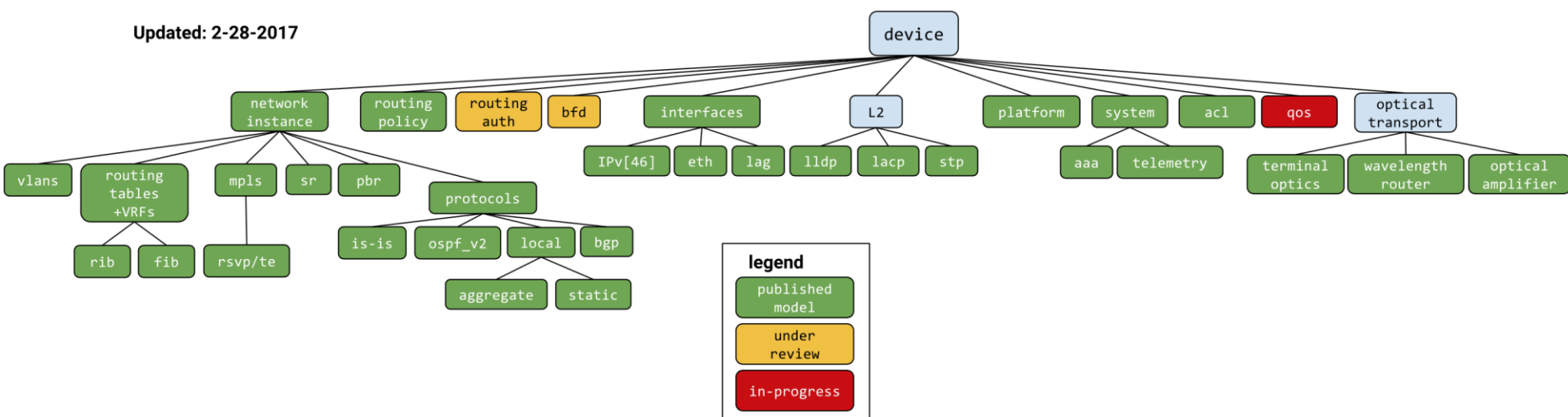
OpenConfig 取り組み

- **共通モデルの作成**
 - ベンダニュートラルなモデル
- **Telemetryの置き換え**
 - SNMPよりもスケールし、拡張性が高い
- **共通のネットワーク管理インターフェースの作成**
 - gRPC Network Management Interface(gNMI)

ベンダ主導ではなくオペレータ主導

OpenConfig 現在定義済みのモデル

Updated: 2-28-2017



出典: <http://www.openconfig.net/projects/models/>

OpenConfig モデル例 interfaces



```
module: openconfig-interfaces
+--rw interfaces
+--rw interface* [name]
+--rw name -> ../config/name
+--rw config
+--rw type identityref
+--rw mtu? uint16
+--rw name? string
+--rw description? string
+--rw enabled? boolean
+--ro state
+--ro type identityref
+--ro mtu? uint16
+--ro name? string
+--ro description? string
+--ro enabled? boolean
+--ro ifindex? uint32
+--ro admin-status enumeration
+--ro oper-status enumeration
+--ro last-change? yang:timeticks
+--ro counters
+--ro in-octets? yang:counter64
+--ro in-unicast-pkts? yang:counter64
+--ro in-broadcast-pkts? yang:counter64
+--ro in-multicast-pkts? yang:counter64
+--ro in-discards? yang:counter64
+--ro in-errors? yang:counter64
+--ro in-unknown-protos? yang:counter32
+--ro out-octets? yang:counter64
+--ro out-unicast-pkts? yang:counter64
+--ro out-broadcast-pkts? yang:counter64
+--ro out-multicast-pkts? yang:counter64
+--ro out-discards? yang:counter64
+--ro out-errors? yang:counter64
+--ro last-clear? yang:date-and-time
+--rw hold-time
+--rw config
+--rw up? uint32
+--rw down? uint32
+--ro state
+--ro up? uint32
+--ro down? uint32
+--rw subinterfaces
+--rw subinterface* [index]
+--rw index -> ../config/index
+--rw config
+--rw index? uint32
+--rw name? string
+--rw description? string
+--rw enabled? boolean
+--ro state
+--ro index? uint32
+--ro name? string
+--ro description? string
+--ro enabled? boolean
+--ro ifindex? uint32
+--ro admin-status enumeration
+--ro oper-status enumeration
+--ro last-change? yang:timeticks
+--ro counters
+--ro in-octets? yang:counter64
+--ro in-unicast-pkts? yang:counter64
+--ro in-broadcast-pkts? yang:counter64
+--ro in-multicast-pkts? yang:counter64
+--ro in-discards? yang:counter64
+--ro in-errors? yang:counter64
+--ro in-unknown-protos? yang:counter32
+--ro out-octets? yang:counter64
+--ro out-unicast-pkts? yang:counter64
+--ro out-broadcast-pkts? yang:counter64
+--ro out-multicast-pkts? yang:counter64
+--ro out-discards? yang:counter64
+--ro out-errors? yang:counter64
+--ro last-clear? yang:date-and-time
```

OpenConfigメンバー



Google



AT&T



Microsoft



British Telecom



Facebook



Comcast



Verizon



Level 3



Cox Communications



Yahoo!



Apple



Jive Communications



Deutsche Telekom / Terastream



Bell Canada



SK Telecom



Bloomberg



Netflix



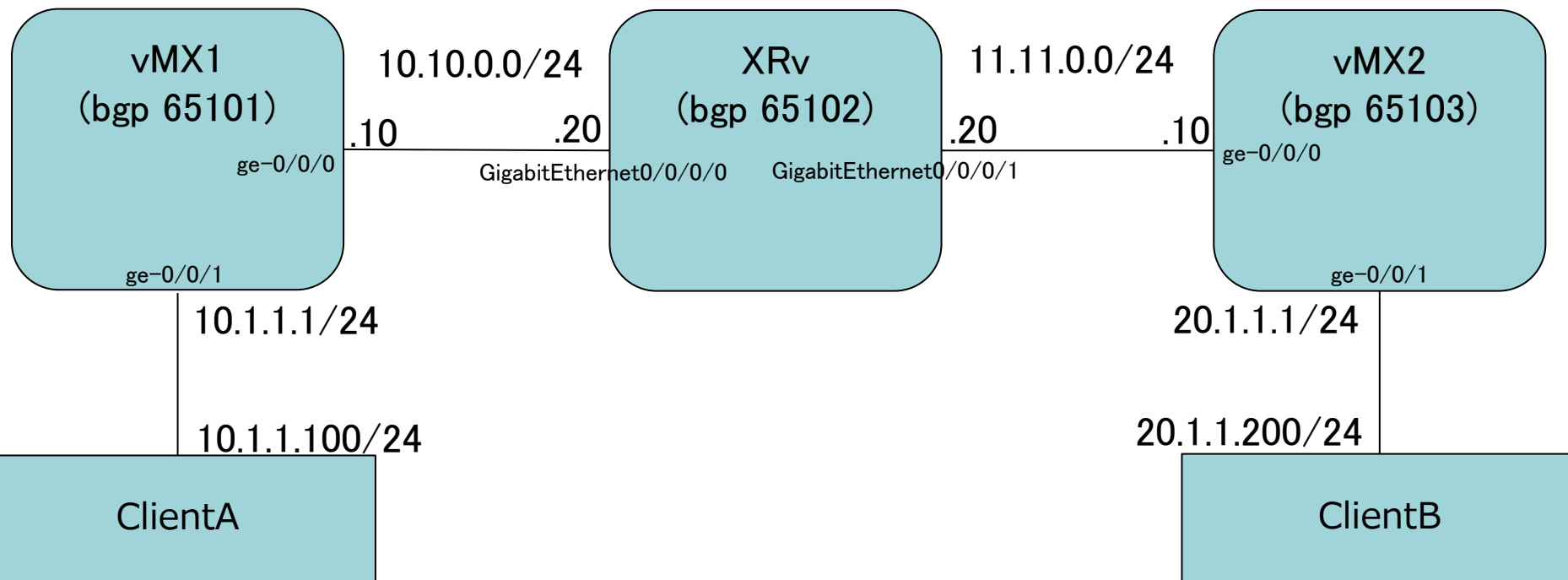
Cloudflare

出典: <http://openconfig.net/about/participants/>

デモ構成



- vMX: vMX 17.2R1.13
- XRv: iosxrv 9k demo 6.1.2
- Host OS: Ubuntu 16.04
- NETCONF Client: ncclient 0.5.3



デモ① VLAN



vMXとXRvへVLAN設定

vMX1

**ge-0/0/0
vlan 100**

XRv

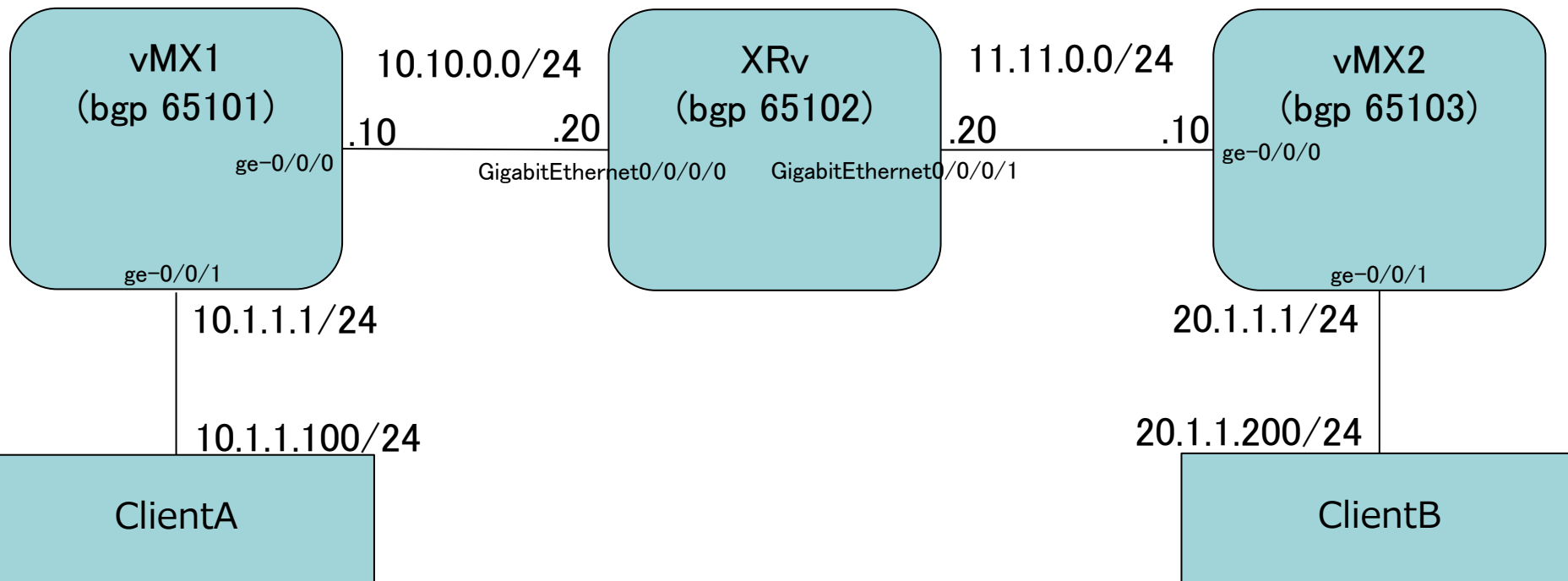
**GigabitEthernet0/0/0/0
vlan 100**

```
<config>
<interfaces xmlns="http://openconfig.net/yang/interfaces">
  <interface>
    <name>インターフェース名</name>
    <subinterfaces>
      <subinterface>
        <index>100</index>
        <vlan xmlns="http://openconfig.net/yang/vlan">
          <config>
            <vlan-id>100</vlan-id>
          </config>
        </vlan>
      </subinterface>
    </subinterfaces>
  </interface>
</interfaces>
</config>
```

デモ② BGP



インターフェース、BGP、Routing-policyを設定し、ClientAとClientB間で疎通確認



デモ② BGP デモコンフィグ



```
<config>
<interfaces xmlns="http://openconfig.net/yang/interfaces">
<interface>
<name>ge-0/0/0</name>
<subinterfaces>
<subinterface>
<index>0</index>
<config>
<index>0</index>
<enabled>true</enabled>
</config>
</subinterface>
</subinterfaces>
</interface>
<interface>
<name>ge-0/0/1</name>
<subinterfaces>
<subinterface>
<index>0</index>
<config>
<index>0</index>
<enabled>true</enabled>
</config>
</subinterface>
</subinterfaces>
</interface>
<interface>
<name>ge-0/0/1.1</name>
<subinterfaces>
<subinterface>
<index>0</index>
<config>
<index>0</index>
<enabled>true</enabled>
</config>
</subinterface>
</subinterfaces>
</interface>
</interfaces>
<routing-policy xmlns="http://openconfig.net/yang/routing-policy">
<policy-definitions>
<policy-definition>
<name>bgp-in</name>
<statements>
<statement>
<name>bgp-in</name>
<actions>
<config>
<accept-route/>
</config>
</actions>
</statement>
</statements>
</policy-definition>
<policy-definition>
<name>bgp-out</name>
<statements>
<statement>
<name>bgp-out</name>
<actions>
<config>
<accept-route/>
</config>
</actions>
</statement>
</statements>
</policy-definition>
</routing-policy>
<bgp xmlns="http://openconfig.net/yang/bgp">
<global>
<config>
<as>65101</as>
<router-id>10.10.10.10</router-id>
</config>
</global>
<afi-safis>
<afi-safi>
<afi-safi-name xmlns:idx="http://openconfig.net/yang/bgp-types">idx:IPV4_UNICAST</afi-safi-name>
<config>
<enabled>true</enabled>
</config>
</afi-safi>
</afi-safis>
<neighbors>
<neighbor>
<neighbor-address>10.10.10.10</neighbor-address>
<config>
<peer-group>group65101</peer-group>
</config>
</neighbor>
<neighbor>
<neighbor-address>10.10.10.10</neighbor-address>
<config>
<peer-group>group65102</peer-group>
</config>
</neighbor>
</neighbors>
<afi-safis>
<afi-safi>
<afi-safi-name xmlns:idx="http://openconfig.net/yang/bgp-types">idx:IPV4_UNICAST</afi-safi-name>
<config>
<enabled>true</enabled>
</config>
</afi-safi>
</afi-safis>
<neighbors>
<neighbor>
<neighbor-address>10.10.10.10</neighbor-address>
<config>
<import-policy>bgp-in</import-policy>
<export-policy>bgp-out</export-policy>
</config>
</neighbor>
</neighbors>
<peer-groups>
<peer-group>
<peer-group-name>group65102</peer-group-name>
<config>
<peer-as>65102</peer-as>
</config>
</peer-group>
</peer-groups>
</bgp>
</config>
```

vmx1.xml

```
<config>
<interfaces xmlns="http://openconfig.net/yang/interfaces">
<interface>
<name>GigabitEthernet0/0/0</name>
<subinterfaces>
<subinterface>
<index>0</index>
<config>
<index>0</index>
<enabled>true</enabled>
</config>
</subinterface>
</subinterfaces>
</interface>
<interface>
<name>GigabitEthernet0/0/1</name>
<subinterfaces>
<subinterface>
<index>0</index>
<config>
<index>0</index>
<enabled>true</enabled>
</config>
</subinterface>
</subinterfaces>
</interface>
</interfaces>
<routing-policy xmlns="http://openconfig.net/yang/routing-policy">
<policy-definitions>
<policy-definition>
<name>bgp-in</name>
<statements>
<statement>
<name>bgp-in</name>
<actions>
<accept-route/>
</actions>
</statement>
</statements>
</policy-definition>
<policy-definition>
<name>bgp-out</name>
<statements>
<statement>
<name>bgp-out</name>
<actions>
<accept-route/>
</actions>
</statement>
</statements>
</policy-definition>
</routing-policy>
<bgp xmlns="http://openconfig.net/yang/bgp">
<global>
<config>
<as>65102</as>
<router-id>10.10.10.10</router-id>
</config>
</global>
<afi-safis>
<afi-safi>
<afi-safi-name xmlns:idx="http://openconfig.net/yang/bgp-types">idx:ipv4-unicast</afi-safi-name>
<config>
<enabled>true</enabled>
</config>
</afi-safi>
</afi-safis>
<neighbors>
<neighbor>
<neighbor-address>10.10.10.10</neighbor-address>
<config>
<peer-group>group65101</peer-group>
</config>
</neighbor>
<neighbor>
<neighbor-address>10.10.10.10</neighbor-address>
<config>
<peer-group>group65102</peer-group>
</config>
</neighbor>
</neighbors>
<afi-safis>
<afi-safi>
<afi-safi-name xmlns:idx="http://openconfig.net/yang/bgp-types">idx:ipv4-unicast</afi-safi-name>
<config>
<enabled>true</enabled>
</config>
</afi-safi>
</afi-safis>
<neighbors>
<neighbor>
<neighbor-address>10.10.10.10</neighbor-address>
<config>
<import-policy>bgp-in</import-policy>
<export-policy>bgp-out</export-policy>
</config>
</neighbor>
</neighbors>
<peer-groups>
<peer-group>
<peer-group-name>group65101</peer-group-name>
<config>
<peer-as>65101</peer-as>
</config>
</peer-group>
<peer-group>
<peer-group-name>group65103</peer-group-name>
<config>
<peer-as>65103</peer-as>
</config>
</peer-group>
</peer-groups>
</bgp>
</config>
```

xrv.xml



Innovative R&D by NTT

デモ②のコンフィグ

<https://gist.github.com/hichihara/b5ec91275513742489119f564cf732ed>

検証していて辛かったこと

• vMXとXRvの実装バージョンの違い

- vMXのopenconfig-if-ipはバージョン1.0.1(図左)で実装されているが、XRvは0.2.0(図右)で実装されている

module: openconfig-if-ip	module: openconfig-if-ip
<pre> augment /oc-if:interfaces/oc-if:interface/oc-if:subinterfaces/oc-if:subinterface: +--rw ipv4 +--rw addresses +--rw address* [ip] +--rw ip -> ../config/ip +--rw config +--rw ip? inet:ipv4-address-no-zone +--rw prefix-length? uint8 +--ro state +--ro ip? inet:ipv4-address-no-zone +--ro prefix-length? uint8 +--ro origin? ip-address-origin </pre>	<pre> augment /ocif:interfaces/ocif:interface/ocif:subinterfaces/ocif:subinterface: +--rw ipv4! +--rw address* [ip] +--rw ip -> ../config/ip +--rw config +--rw ip? inet:ipv4-address-no-zone +--rw prefix-length? uint8 +--ro state +--ro ip? inet:ipv4-address-no-zone +--ro prefix-length? uint8 +--ro origin? ip-address-origin </pre>

• 機器によって使用可能なインターフェースが異なる

- vMX、XRvはnetconfに対応、vEOSはnetconfに未対応
- vEOSはgRPCに対応、vMX、XRvは対応しているらしいがドキュメントが不足していてまともに使えない

• 機器によってOpenConfigでの設定が通常の設定と別に管理される→次ページ参照

JunosのOpenConfigの設定の仕方



```
interfaces {
  ge-0/0/1 {
    vlan-tagging;
    unit 101 {
      vlan-id 101;
      family inet {
        address 192.168.101.1/24;
      }
    }
  }
  fxp0 {
    unit 0 {
      family inet {
        address 172.26.211.26/16;
      }
    }
  }
}
openconfig-interfaces:interfaces {
  interface ge-0/0/1 {
    subinterfaces {
      subinterface 200 {
        config {
          index 200;
          enabled true;
        }
        openconfig-if-ip:ipv4 {
          addresses {
            address 192.168.200.1 {
              config {
                ip 192.168.200.1;
                prefix-length 24;
              }
            }
          }
        }
      }
    }
    openconfig-vlan:vlan {
      config {
        vlan-id 200;
      }
    }
  }
}
}
```

**Junosの一般的な操作
(CLI、Juniper名前空間でのnetconf) で設定したものはここに反映**

OpenConfig名前空間でのnetconf操作で設定したものはここに反映(CLIでの設定は可能)

検証していて辛かったこと②

• 機器によって使用するスキーマが違う

- openconfig-mplsでのMPLSインターフェース設定

```
+---rw mpls
```

```
+---rw global
```

```
| +---rw config
```

```
| +---ro state
```

```
| +---rw interface-attributes
```

```
| | +---rw interface* [interface-id]
```

```
| |   +---rw interface-id -> ../config/interface-id
```

```
| |   +---rw config
```

```
| |     | +---rw interface-id? oc-if:interface-id
```

```
| |     | +---rw mpls-enabled? boolean
```

```
| + +
```

```
|
```

```
+---rw te-interface-attributes
```

```
| +---rw interface* [interface-id]
```

```
|   +---rw interface-id -> ../config/interface-id
```

```
|   +---rw config
```

```
|     | +---rw interface-id? oc-if:interface-id
```

vMXで有効

XRvで有効

デモボツエピソード



MTU設定をデモしようとしていたがvMXとXRvで実装が違っていた

```
<config>
  <interfaces xmlns="http://openconfig.net/yang/interfaces">
    <interface>
      <name>インターフェース名</name>
      <config>
        <type xmlns:idx="urn:ietf:params:xml:ns:yang:iana-if-type">idx:ethernetCsmacd</type>
        <mtu>1400</mtu>
      </config>
    </interface>
  </interfaces>
</config>
```

コンフィグ投入できるが実際はサポートされていない

Command Name	OpenConfig Command Path	Junos Configuration
Interface Type	<code>/interfaces/interface/config/type</code>	Not supported.

出典: https://www.juniper.net/documentation/en_US/junos/information-products/pathway-pages/open-config/open-config-feature-guide.pdf



OpenConfig 注意点・懸念点

- すべてのモデルが存在するわけではない
- モデルがあってもスキーマに不足がある
 - インターフェースのスキーマにDHCP設定がない
- バージョンによってスキーマが異なる場合がある
 - openconfig-if-ip
- OSSだがオープンコミュニティではない
- 機器側で実装がない場合がある
 - VRF設定用のopenconfig-network-instanceは実装がない
 - モデルがありスキーマの実装もあるが未対応パラメータがある

OpenConfigへの期待

あらゆるベンダ機器間のギャップの抽象化

