

ジュニパーネットワークス主催
JUNOSハンズオントレーニング

SRXシリーズ サービス ゲートウェイ コース

ジュニパーネットワークス株式会社

2016年9月 rev.1.2.a

はじめに

- 本資料にあるロードマップの内容は、資料作成時点におけるジュニパーネットワークスの予定を示したものであり、事前の通告無しに内容が変更されることがあります。
- またロードマップに描かれている機能や構成は、購入時の条件になりませんので、ご注意ください。

Legal Disclaimer:

This statement of product direction (formerly called “roadmap”) sets forth Juniper Networks' current intention, and is subject to change at any time without notice. No purchases are contingent upon Juniper Networks delivering any feature or functionality depicted on this statement.

JUNOS Basic

JUNOS Hands-on Training

Juniper Network, K.K.

Training Outline “JUNOS Basic”

トレーニング内容（前半）	記載ページ
ジュニパーネットワークス会社紹介	P.6
JUNOSとは	P.13
運用面からみたJUNOSのアドバンテージ	P.25
トレーニング・デバイスへのアクセス方法	P.35
CLIモードと各モード間の移動	P.39
JUNOS CLI操作 ～Operationalモード～	P.46
JUNOS CLI操作 ～Configurationモード～	P.65
JUNOSシステム設定	P.82
JUNOSインタフェース設定	P.90
JUNOS経路設定	P.98
Firewall Filterの設定	P.102

Training Outline Service Gateway "SRX" course

トレーニング内容（後半）	記載ページ
Juniper SRXシリーズ製品紹介	P.110
LAB.1 JUNOSの基本的な操作・設定	P.120
LAB.2 Firewallの設定	P.132
LAB.3 NATの設定	P.152
LAB.4 Chassis Clusterの設定	P.172
TIPs to be JUNOS Experts	P.207
まとめ	P.232
Appendix A: Chassis Cluster Deep Dive	P.236
Appendix B: IPsec VPNの設定	P.250
Appendix C: NAT pool options	P.266
Appendix D: Security Logging	P.270



ジュニパーネットワークス 会社紹介

ジュニパーネットワークス 会社概要

設立： 1996年

本社所在地： カリフォルニア州サニーベール

Juniper Networks (NYSE: JNPR)

CEO： Rami Rahim

事業概要： IP通信機器（ルータ・スイッチ）及び
セキュリティ製品（ファイアウォール・IPS）の製造販売

従業員： 約9,000名

拠点： 46カ国 100拠点以上

年間売上規模： 約5600億円



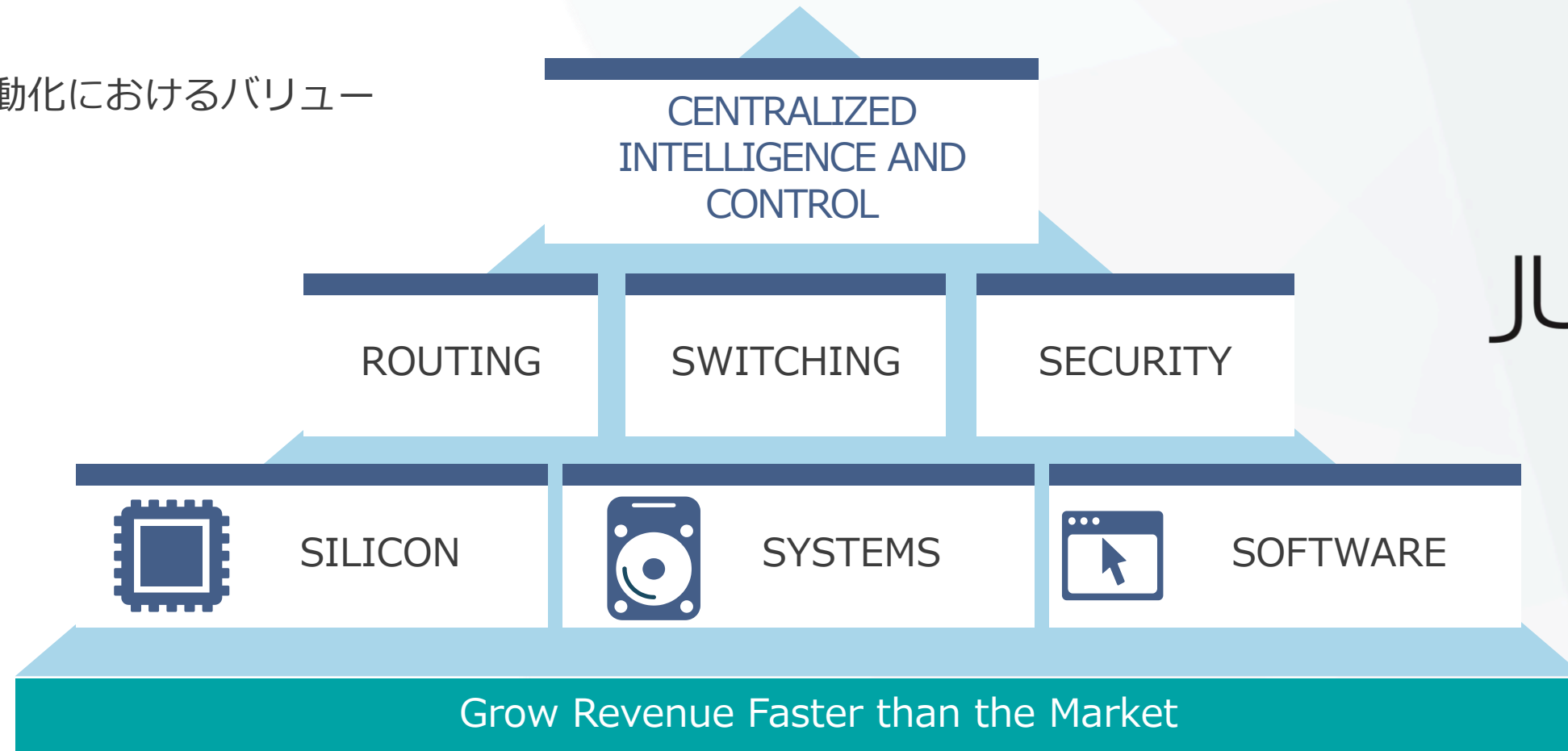
ジュニパーネットワークスの戦略

Vision: ネットワークイノベーションにおけるリーダー

Go-To-Market: ハイパフォーマンスネットワーキングをビジネスの基盤と位置付けるお客様とパートナー様に価値を提供

パフォーマンスと自動化におけるバリュー

- ✓ スケーラブル
- ✓ 信頼性
- ✓ セキュリティ
- ✓ 高コスト効率
- ✓ 俊敏性
- ✓ 高効率




JUNOS[®]

ジュニパーネットワークスの戦略



プロダクト・ポートフォリオ (カテゴリ別)

ROUTING



MX Series

MX2020
MX2010
MX960
MX480
MX240
MX104
MX80
MX40
MX10
MX5
vMX



PTX Series

PTX5000
PTX3000
PTX1000



ACX Series

ACX5000
ACX4000
ACX2100
ACX2000
ACX1100
ACX1000
ACX500

SWITCHING



EX Series

EX9200
EX8200
EX6200
EX4600
EX4550
EX4500
EX4300
EX4200
EX3300
EX2200
EX2200-C



QFX Series

QFX10000
QFX5200
QFX5100
QFX3600
QFX3500
QFabric
(QFX3000-G/M)

SECURITY



SRX Series

SRX5800
SRX5600
SRX5400
SRX3600
SRX3400
SRX1400
SRX1500
SRX550
SRX345
SRX340
SRX320
SRX300
vSRX



NetScreen Series

NetScreen-5200
NetScreen-5400



SSG Series

SSG550M
SSG520M
SSG350M
SSG320M
SSG140



ISG Series

ISG2000
ISG1000



Network Director
Security Director

JUNOS®

JUNOS: THE POWER OF ONE

Integrated Architecture

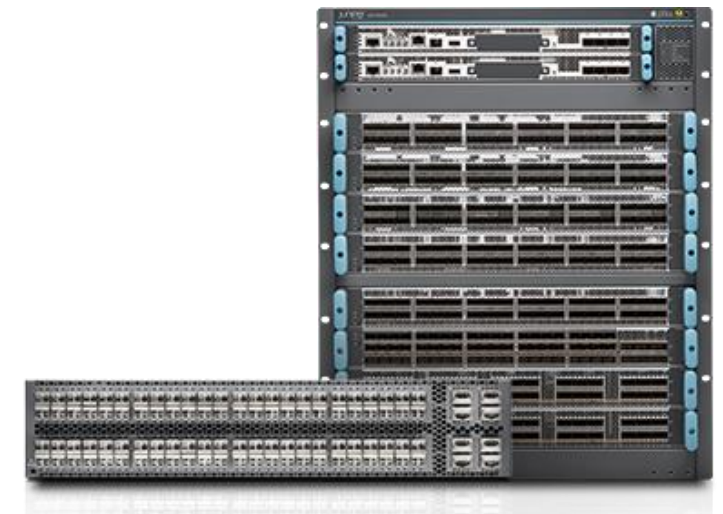
Datacenter Service Gateway
SRX series



Universal Edge Router
MX series



Datacenter Fabric Switch
QFX series



JUNOS: THE POWER OF ONE

Integrated Architecture

Branch Service Gateway
SRX series



Campus Ethernet Switch
EX series





JUNOSとは

multiple operating systems vs. ONE approach



ASA
IOS-XE
IPS OS
IOS
NX-OS
IOS-XR

プラットフォーム毎に異なるOSと機能セット



セキュリティもネットワークもカバーする
業界唯一のシングル・ネットワークOS



THE POWER OF ONE

LEARN ONCE, INTEGRATE ONCE, QUALIFY ONCE

プラットフォーム共通機能

- Routing
- Layer 2 Switching
- Class of Service
- IPv4 and IPv6
- Etc...

Cross-Portfolio Commonality

BGP/MPLS Control Plane

End-to-end Security

In-network Automation

SDK and Licensing of Junos

etc,etc...



ベース・コンポーネント

- Kernel and μ Kernel
- Chassis Management (chassisd)
- IP Services (Telnet, SSH, NTP)
- Network Management
 - (AAA, CLI/mgd, XML/DMI, syslogd)

プラットフォーム専用機能

- Advanced Security (SRX)
- Virtual Chassis Fabric (QFX)
- MPLS/EVPN (MX)
- ISSU (MX&EX9k)
- Etc...

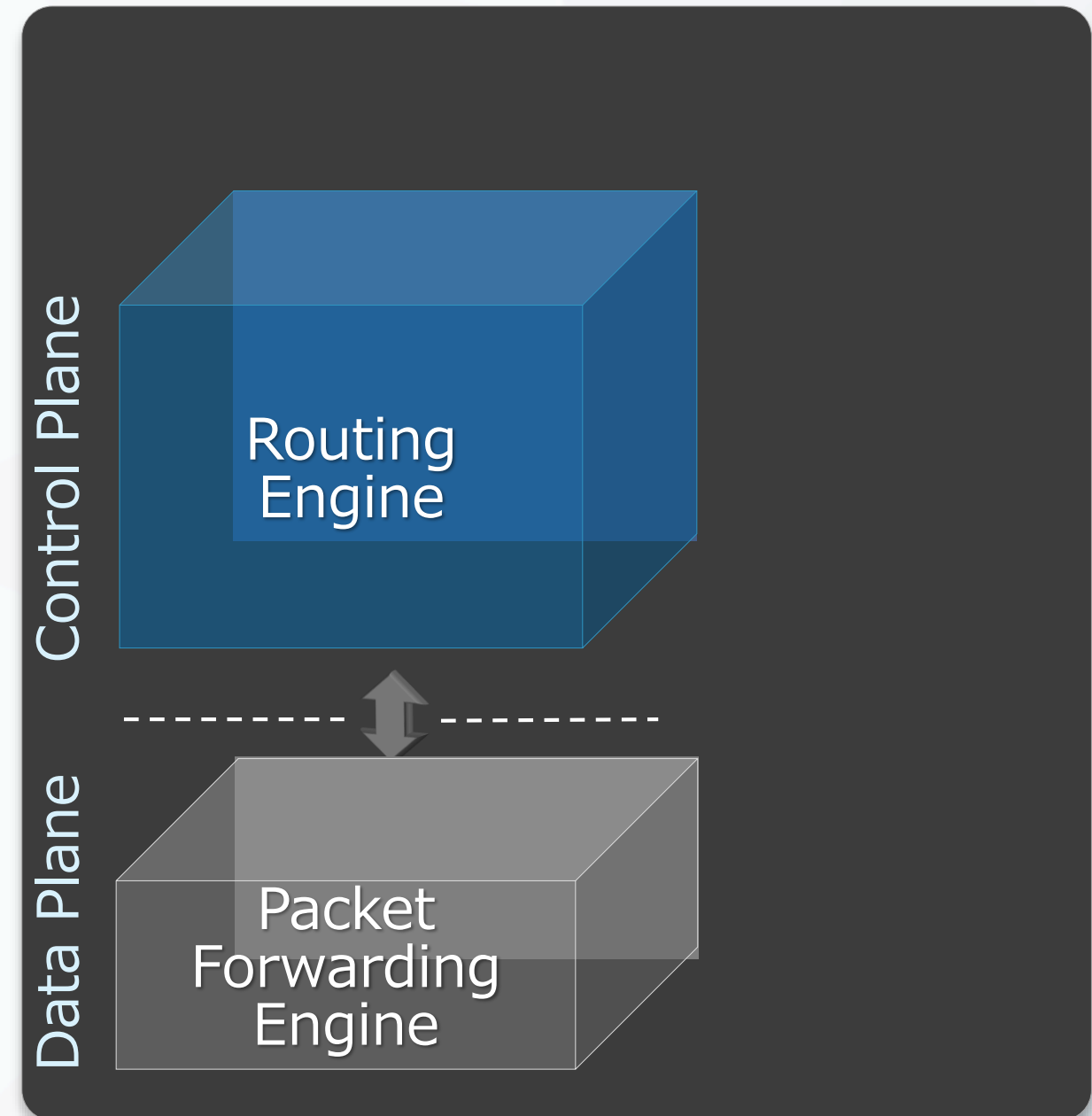
コントロールプレーンとフォワーディングプレーンの分離

Scale and Performance

- 各Planeにおけるパフォーマンスを担保
- より高いパフォーマンスをそれぞれの領域で独立して開発することが可能に

Resilient

- 独立したオペレーション
 - Routing Engine (RE)
 - Packet Forwarding Engine (PFE)
- 冗長化に対するさまざまなオプションをそれぞれに提供



EVOLUTION OF ONE ARCHITECTURE

モジュラー型

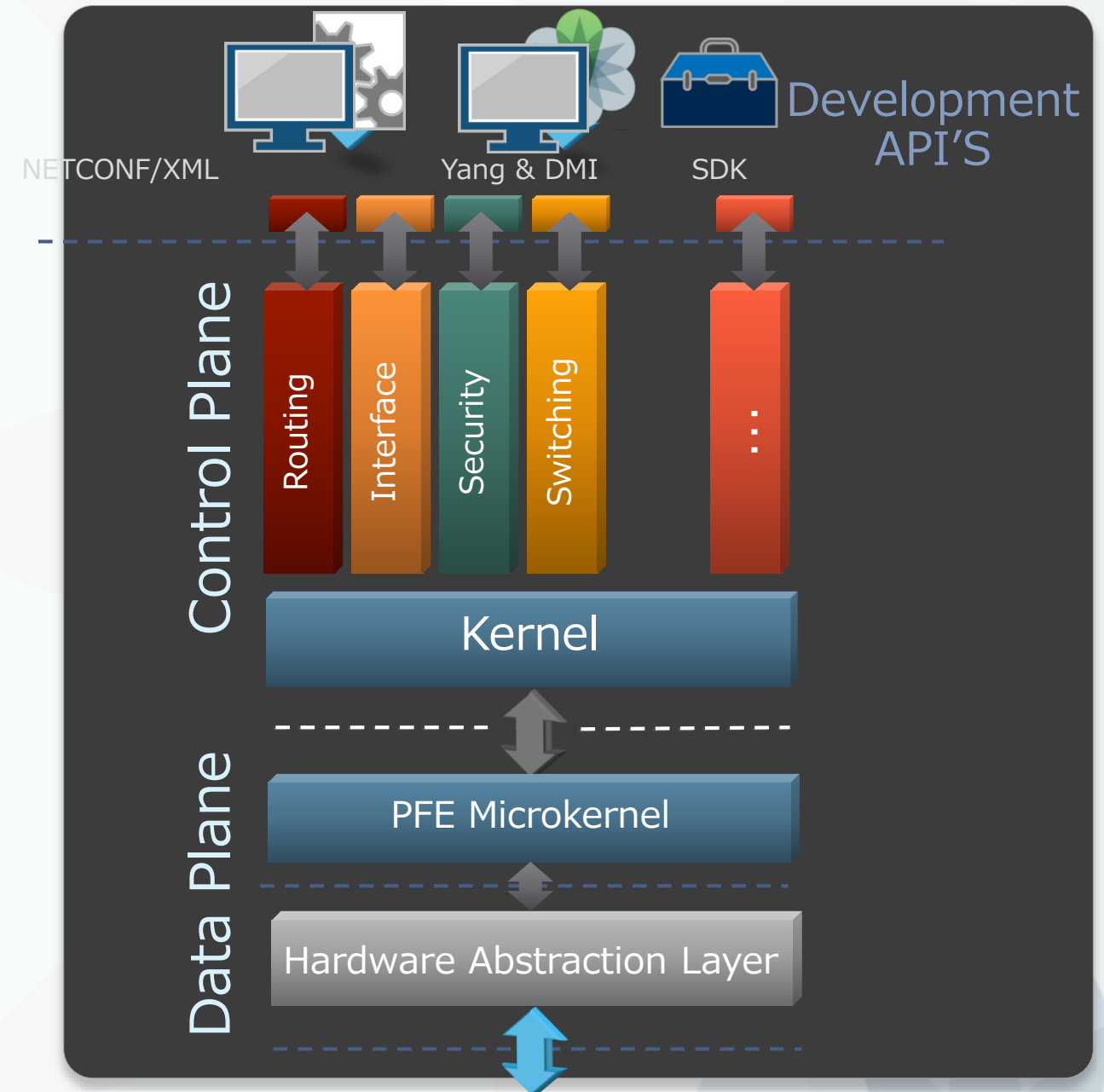
- 拡張性とパフォーマンスを担保するコンポーネント
- 冗長性、安定性、サービス拡張を効率的に提供するための独立したオペレーション

Scalable

- Up: multi-core & 64-bit
- Down: モジュールごとのパッケージング

Open

- Hardware Abstraction Layer
- Automation APIs & Junos SDK



JUNOSのアプローチ・運用者/設計者にとって

ネットワーク・アウトージの原因に対する調査

Planned Maintenance

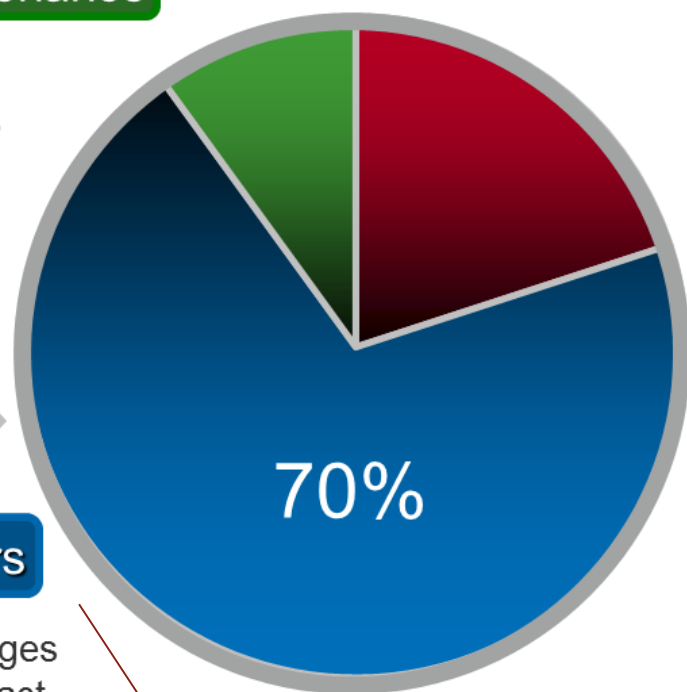
Hardware and software upgrades

Unplanned Events

Network failures, hardware events and software defects

Human Factors

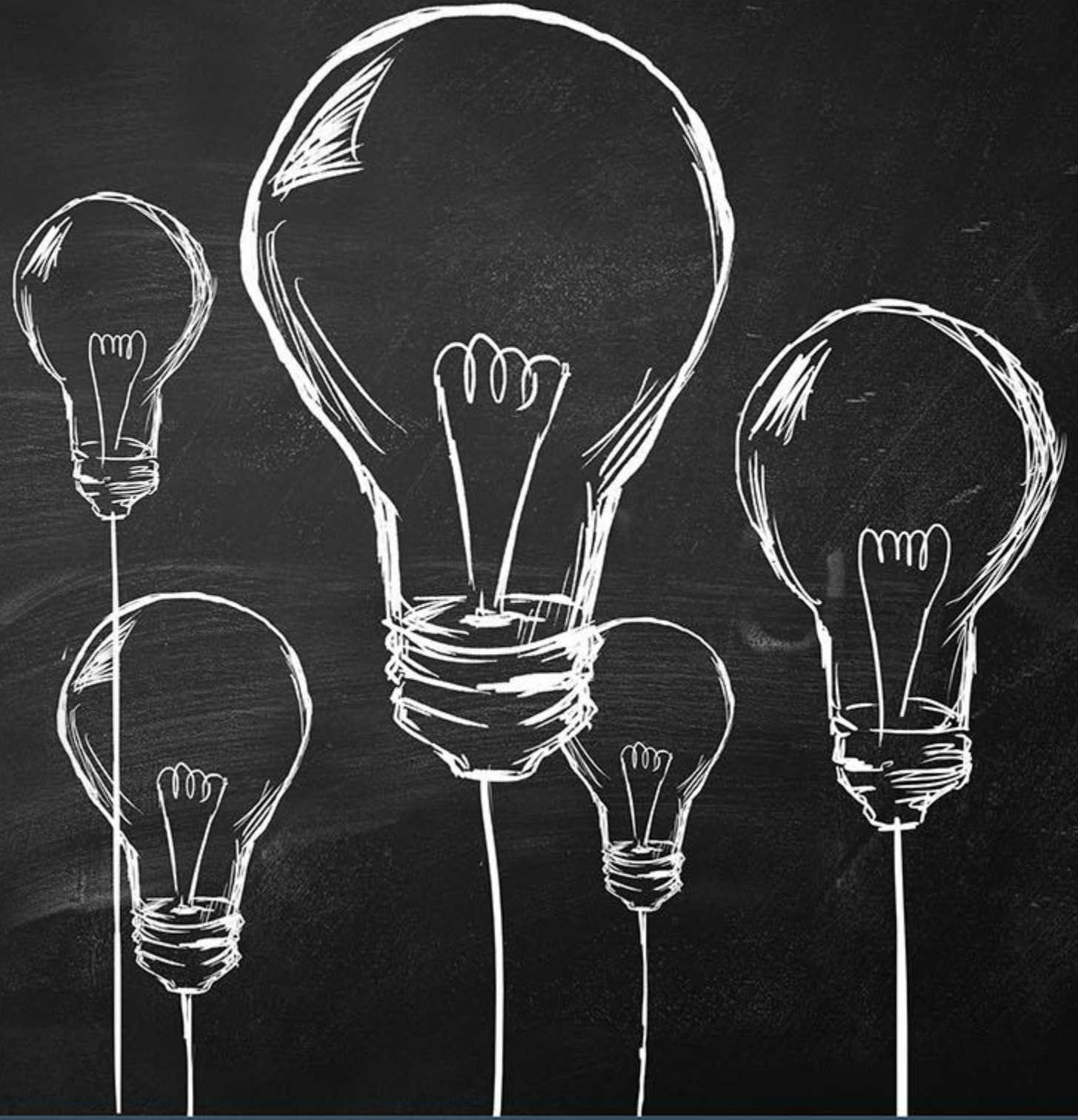
Configuration changes that negatively impact network performance



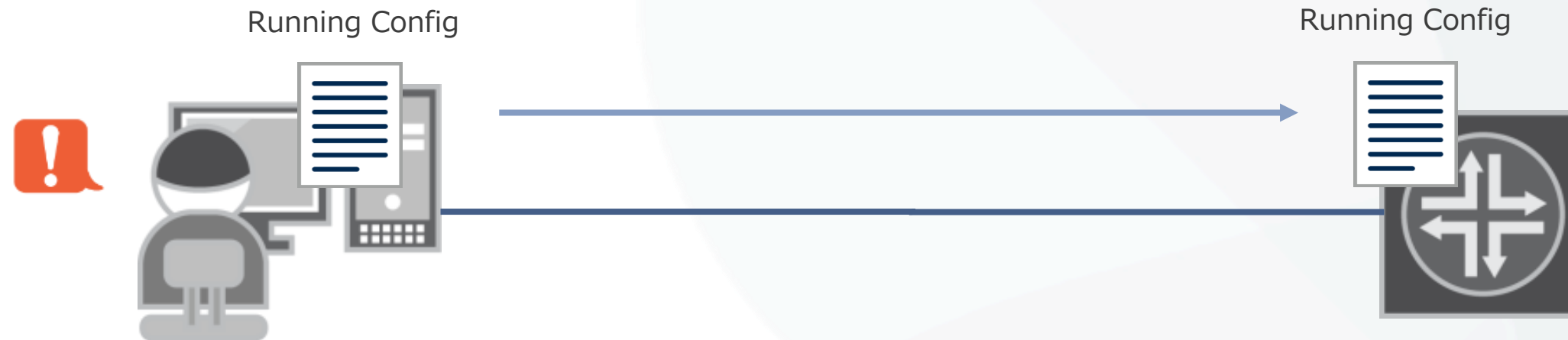
全体の70%が何らかのコンフィグの変更によってネットワークにネガティブなインパクトが発生している

- すべてのネットワーク運用者は、コンフィグ投入や切り戻しといった**作業自体に時間や気力を奪われるべきではない**
- 例えば正確を期すためにネットワーク設定で時間が食われ、新サービスの提供開始が遅れたとしたらサービス提供する側だけでなく、受ける側としては大きなビジネスインパクトを与えてしまう
- 精神論でカバーするだけでなく、いかに**効率的に人的リソースを使用するかはビジネスに多くの影響をもたらすことを認識する必要がある**
- よってJUNOSはIOSライクなCLIとは根本的に異なるアプローチを採用
 - <業界標準型CLI>
コマンド1行毎に命令が実行され、NWへの変更が都度反映される運用
 - <JUNOS>
コマンドで設定ファイルを変更し、意図したタイミングでNWに変更を反映させる運用

CLASSIC



これまでの一般的なNW-OSの不便さ



- 一般的なネットワークOSの場合、管理者がコンソールなどで設定変更を行う際、投入した設定が**即座に実稼働のネットワーク設定へと反映**されてしまう
- このことにより、
 - **ヒューマンエラーが発生する余地がある**
 - **設定の復旧が困難**
 - **意図しない設定を行ってしまうと、機器への通信自体が不可能になってしまうケースがある**などの課題が存在する

MODERN



JUNOSの場合



- JUNOSの場合、管理者が設定変更を行うのは、あくまで**設定ファイル**
これを実ネットワークの設定へと投入するためにはJUNOSによるシステムチェックを行った後に、“commit”というコマンドを投入することにより反映させる
- この仕組みにより、
 - JUNOSのシステムチェックによる**ヒューマンエラーの予防**
 - 設定ファイルは過去50世代まで自動保存されるため、**一瞬で過去の状態へと戻すことができる**
 - 作成した設定ファイルを、“**ためしに**”投入してみることも可能
 - などのメリットを享受することができる。

JUNOSのアプローチ : Human factors への対応

有効なJUNOSツール

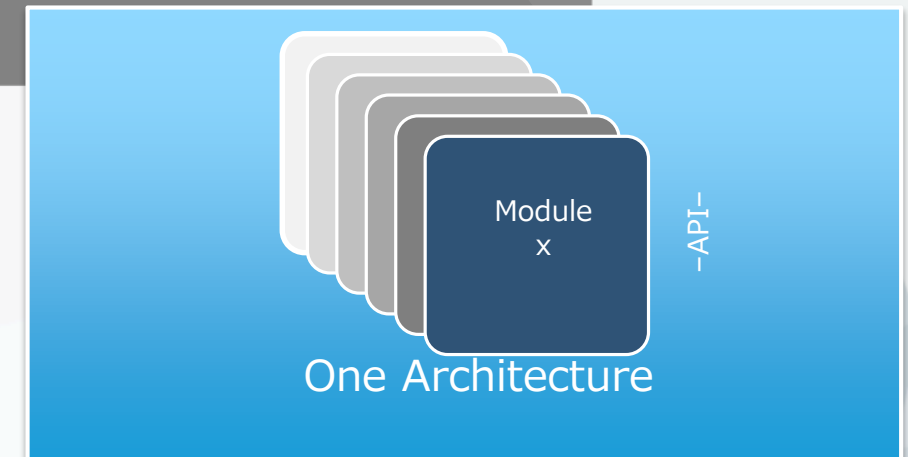
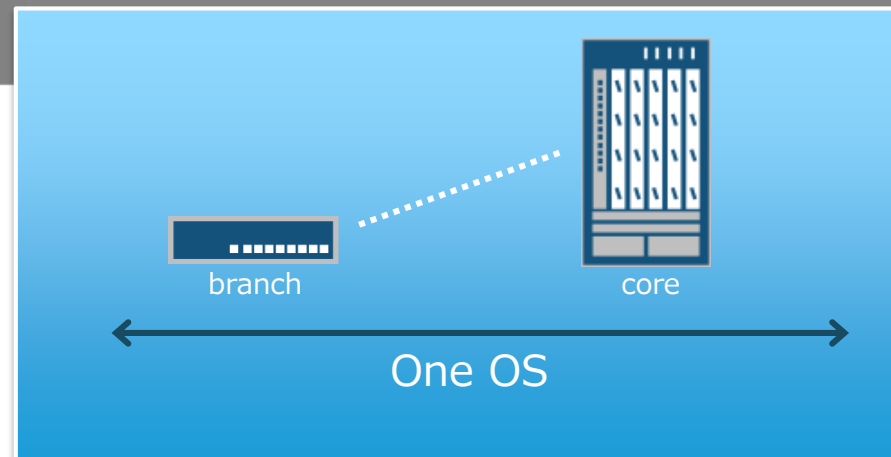
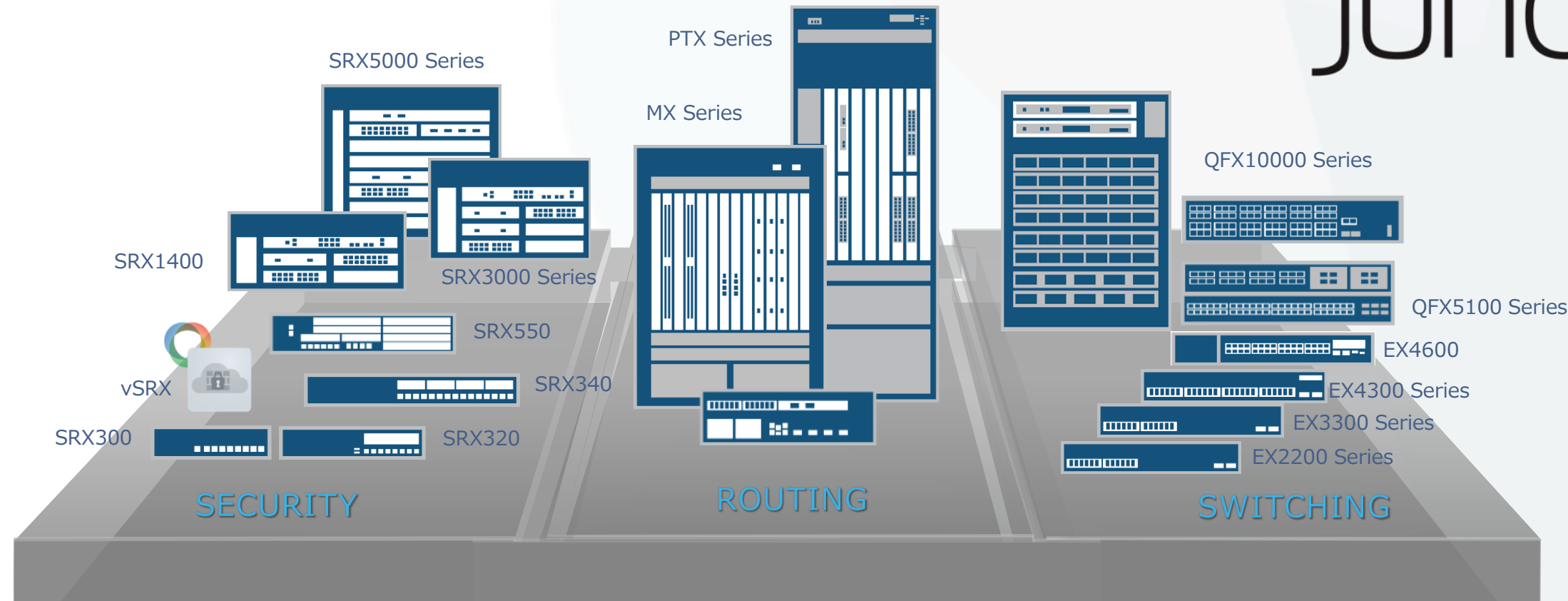
- “commit”
 - 設定変更を有効にするコマンド
 - 有効時にconfigチェックをおこない、誤り（矛盾）がなければ投入した設定が有効となる
- “rollback”
 - 設定の履歴管理、設定・OSの切り戻しを容易に
 - 既存configを含み最大50世代までの管理が可能
 - “Load”コマンドにより外部から設定ファイルを更新することも可能
- “JUNOScript” & “Event Policy”
 - スクリプティングによる自動化ツール
 - イベントをトリガーとした自動化機能

Benefits

- Configミスによるダウンタイムの回避
- Config変更/ 切り戻し作業の時間短縮



JUNOS: THE POWER OF ONE





運用面からみたJUNOSのアドバンテージ

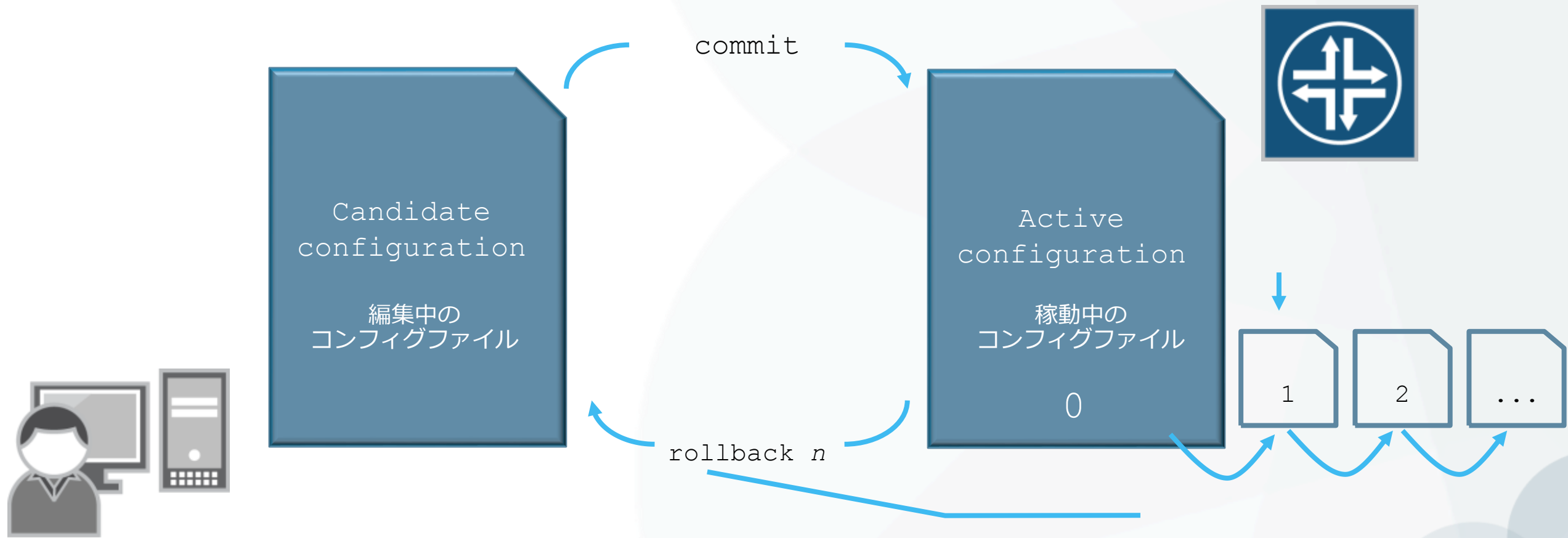
導入, 運用, トラブルシュートに有効なJUNOS UTILITY群

JUNOSは導入、運用、トラブルシュートに有効な様々なツールを提供

- Commit
 - 設定変更を有効にするコマンド
 - check, confirmed, compare、など様々なOptionが使用可能
- Rollback
 - 設定の履歴管理, 設定・OSのきり戻しを容易にする
- 自動化Tool : JUNOScript / Event Policy
 - 運用を自動化するユーティリティ
- Etc...

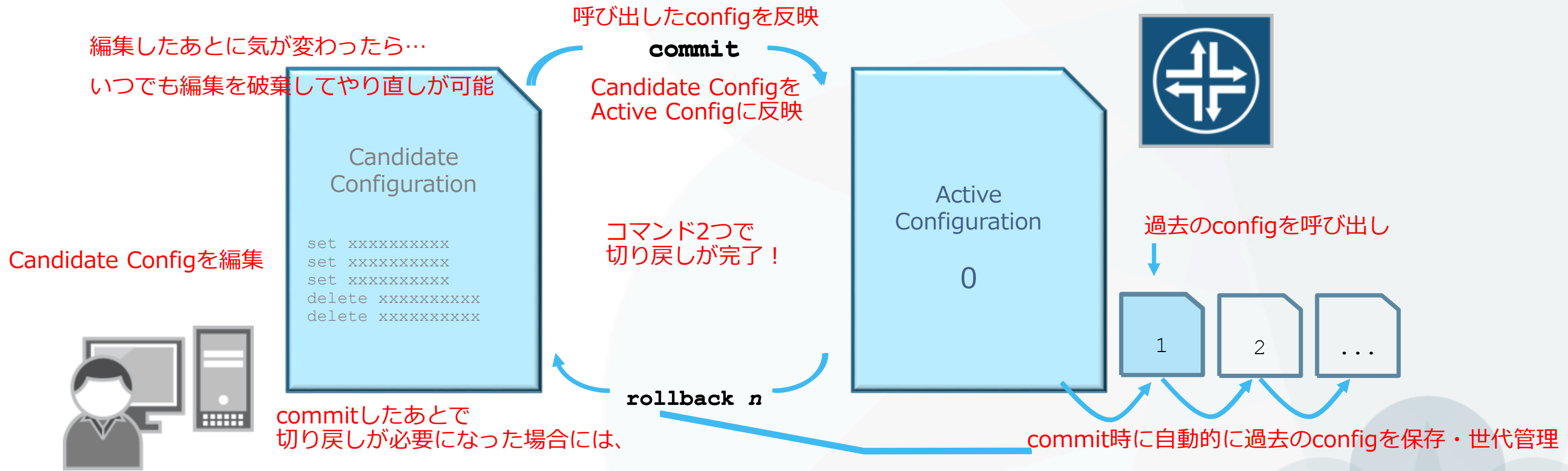
“Commit & Rollback”

Configurationモードで行った設定変更は、Candidate Configurationとして保持され、“Commit”するまで設定はActive Configurationとして反映されません。
万一間違えた場合でも、“rollback”コマンドにてすぐに前の状態に戻ることが可能です。



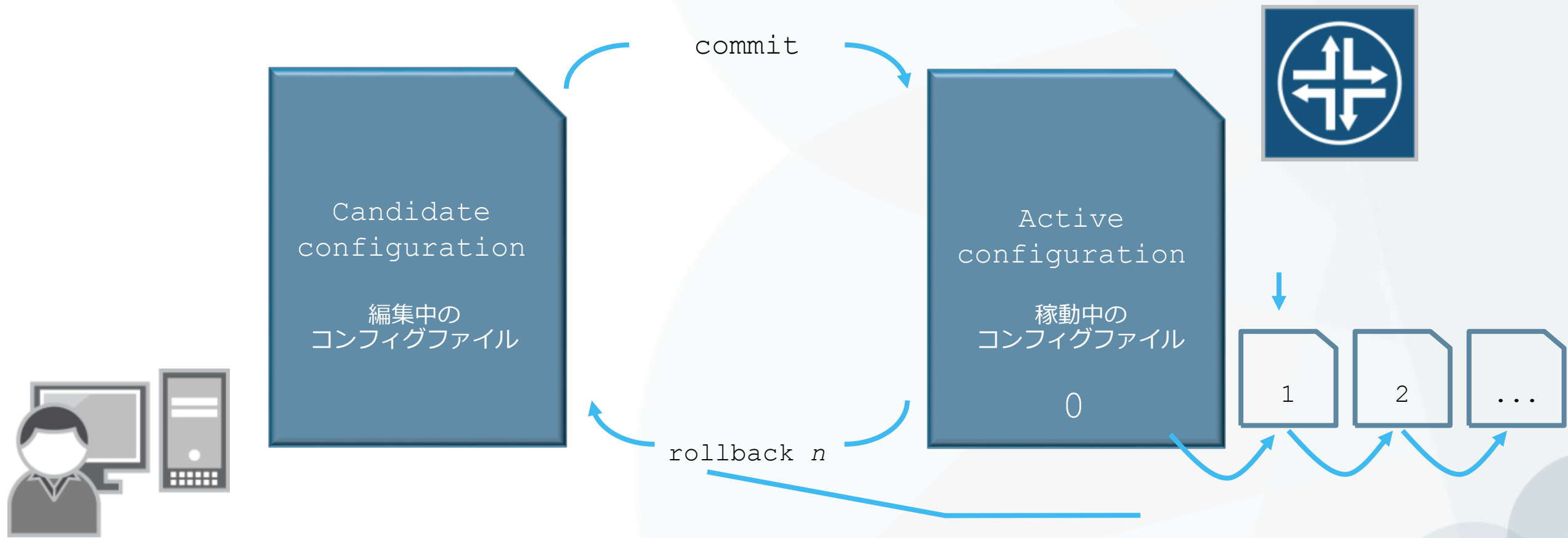
“Commit & Rollback” (アニメ)

Configurationモードで行った設定変更は、Candidate Configurationとして保持され、“Commit”するまで設定はActive Configurationとして反映されません。
万一間違えた場合でも、“rollback”コマンドにてすぐに前の状態に戻ることが可能です。



“Commit & Rollback”

Configurationモードで行った設定変更は、Candidate Configurationとして保持され、“Commit”するまで設定はActive Configurationとして反映されません。
万一間違えた場合でも、“rollback”コマンドにてすぐに前の状態に戻ることが可能です。



JUNOS : commit at time オプション

- 設定反映の時間指定（メンテナンスタイムにおける設定反映）
- commit at xx:xx:xx (time) コマンドでcommitすると、指定した時間に設定ファイルをActivateすることが可能となります

```
[edit]
mike@jnpr1# commit at 02:00:00
commit check succeeds
commit will be executed at 2016-02-02 02:00:00 UTC
Exiting configuration mode
mike@jnpr1>
```

メンテナンスタイムにCommitが自動的に実施されるため、
管理者が該当の時間に操作する必要はなし



JUNOS : commit confirmed オプション

- 設定の自動復旧機能（ヒューマンエラーによるトラブル防止のため）
- commit confirmed コマンドでcommitすると、再度commitしない限り default10分で元のconfigにrollbackします
 - 指定した時間あるいはdefaultの10分以内に2度目のcommitを入れることで、configは完全に格納されます

```
[edit]
root@lab# commit confirmed 5
commit confirmed will be automatically rolled back in 5 minutes unless
confirmed
commit complete
```

設定間違いのままcommitしてしまいSSHなどが繋がらなくなってしまう後も、一定時間のあと1つ前のconfigに自動復旧するため、リモートデバイスのポリシー変更時などに便利



JUNOScriptの概要

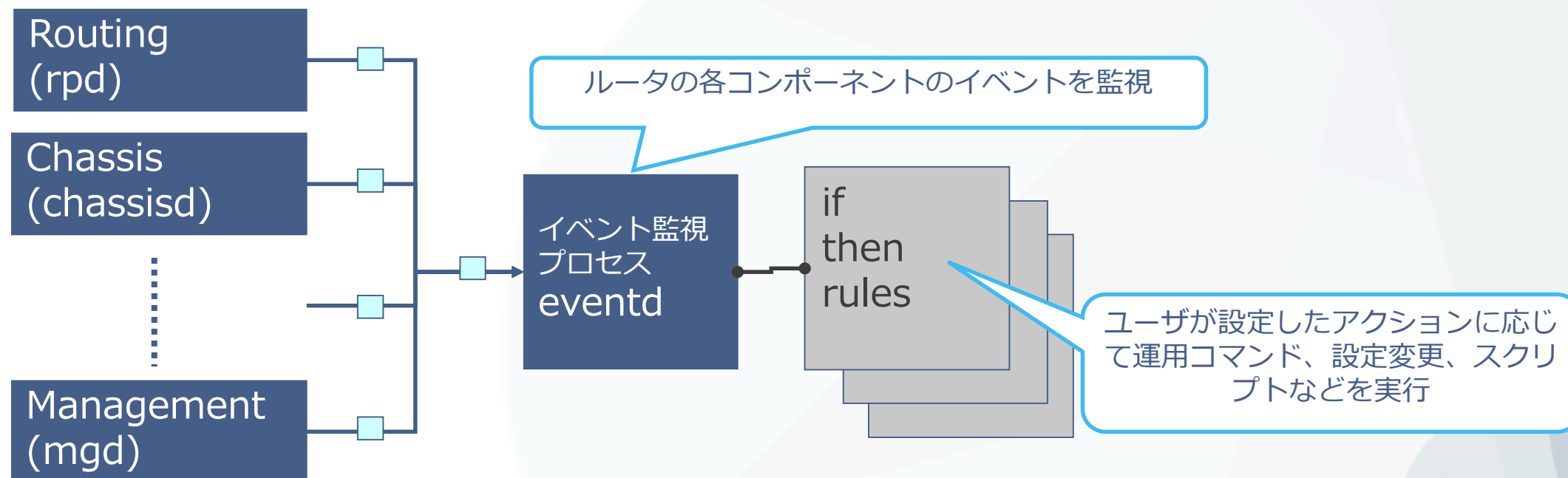
- JUNOScript とはJuniperのネットワーク装置上で動作させることができるスクリプティング機能です。JUNOS自体に手を加える必要がないため、JUNOSの安定性を損なうことなく、ユーザ個別の自動化に対する要望に対し柔軟かつ速やかに対応することができます。
- 大別すると、運用者が起動するスクリプトである“Commit Script”、“Op Script”とシステムが起動するスクリプトである“Event Policy”、“Event Script”が存在します。

XSLT / SLAXベースのスクリプト



JUNOS : Event Policy/Script

- ネットワーク機器上のイベントやタイマーをトリガーとして、コマンドやスクリプトを実行することで、運用の自動化が可能となります。
 - イベントをトリガーとしたアクションを実行 (Self-monitor)
 - ルータ上の特定のイベントをトリガーとして、コマンドやスクリプトを実行
 - タイマーをトリガーとしたアクションの実行
 - インターバル設定や日時指定に応じて、コマンドやスクリプトを実行

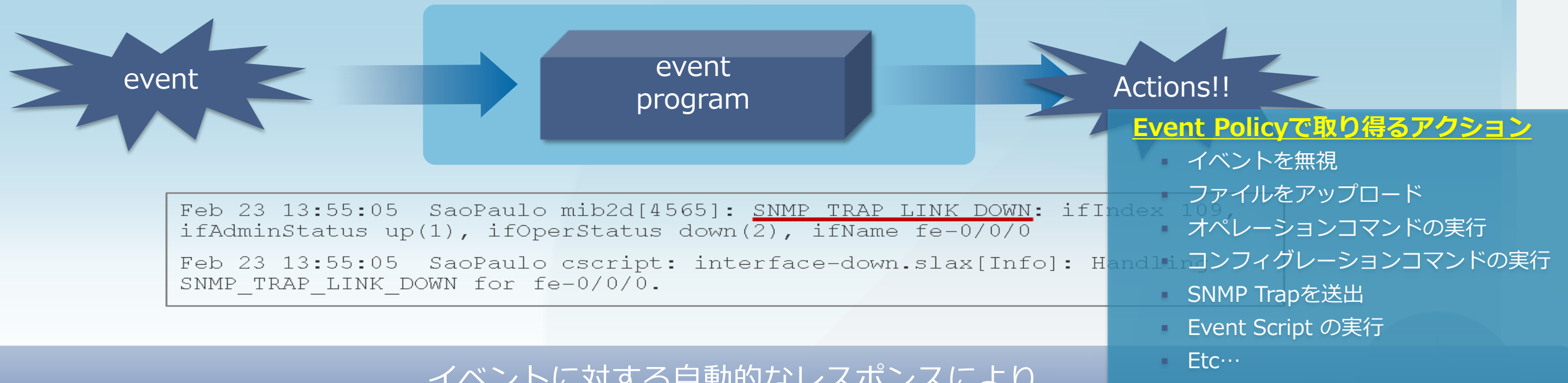


JUNOS: JUNOS Automation EXAMPLE

- EVENT base AUTOMATION

デバイスに発生したイベントに応じて、自動的に発動されるプログラムを事前に設定しておくことが可能

- イベントに応じて自動的にトラブルシューティング用のLogを取得
- イベントに応じて動的なアクションをデバイスに取らせる

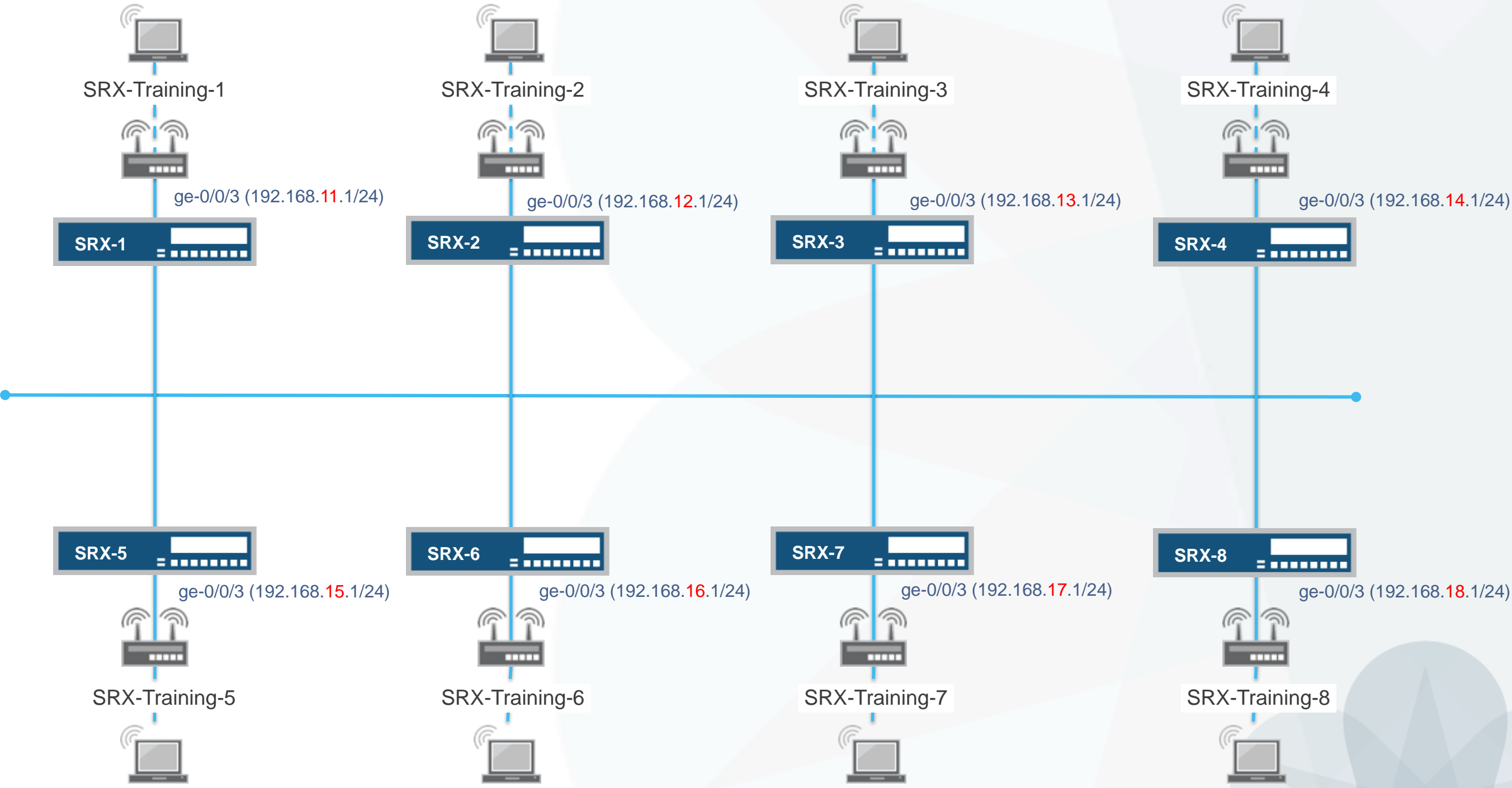


イベントに対する自動的なレスポンスにより
ダウンタイムを削減したり必要なLog取得を自動的に実行することが可能！



トレーニング・デバイスへのアクセス方法

Security "SRX" course Topology (Lab.1 : 基本操作)

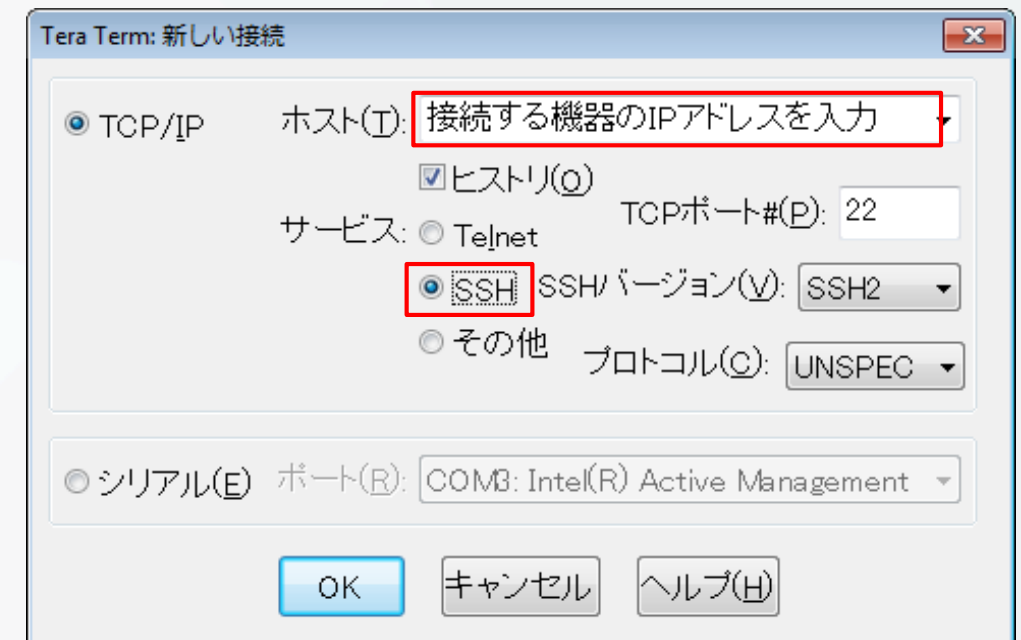


SRXへログイン

初期設定状態のSRXにアカウント'root'でログインします。
cliコマンドでJUNOSのOperationalモードを起動します。

- rootアカウントはserial console、またはssh接続時のみ使用可能です。
- 今回は事前にIPアドレス, rootパスワード, sshを設定済みです。
 - Root Password : [Juniper](#)
- Tera TermからSSHv2接続で接続してください。

```
--- JUNOS 15.1X49-D40.6 built 2016-03-22 05:18:15 UTC
root@% cli
root>
```

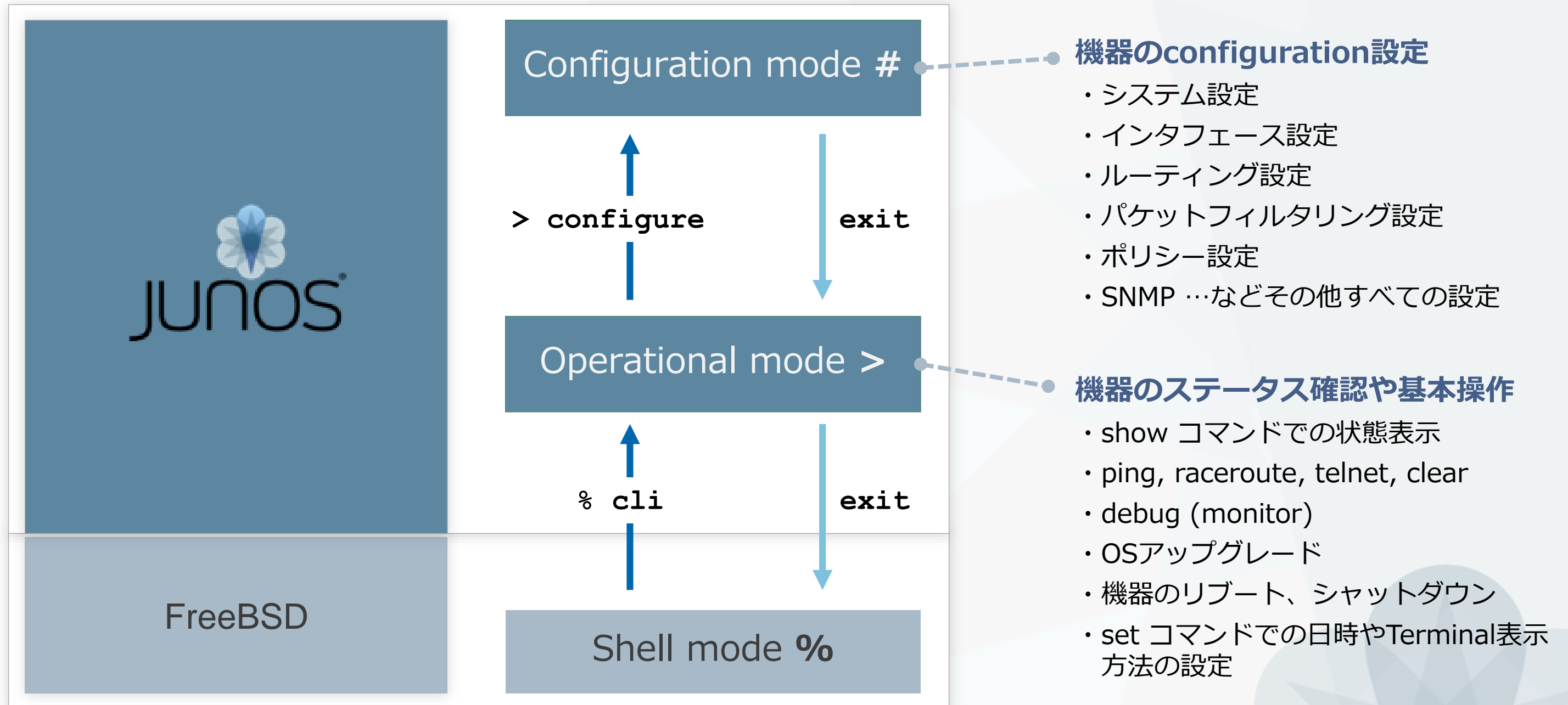




CLIモードと各モード間の移動

CLI概要

- Junos CLIの3つのモード遷移について



Operationalモード

- RootユーザでLoginするとShellモード（プロンプトが“%”）に入ります
 - “cli”と投入することでShellモードからOperationalモードへと移行します

```
login: root
Password:

--- JUNOS 15.1X49-D35 built 2016-02-02 07:16:16 UTC
root@%
root@% cli
root>
```

- Rootユーザ以外でLoginすると、Operationalモード(プロンプトが>)に入ります
 - “start shell”と投入することでOperationalモードからShellモードへと移行します

```
login: AAA
Password:

--- JUNOS 15.1X49-D35 built 2016-02-02 07:16:16 UTC
AAA>
AAA> start shell
%
```

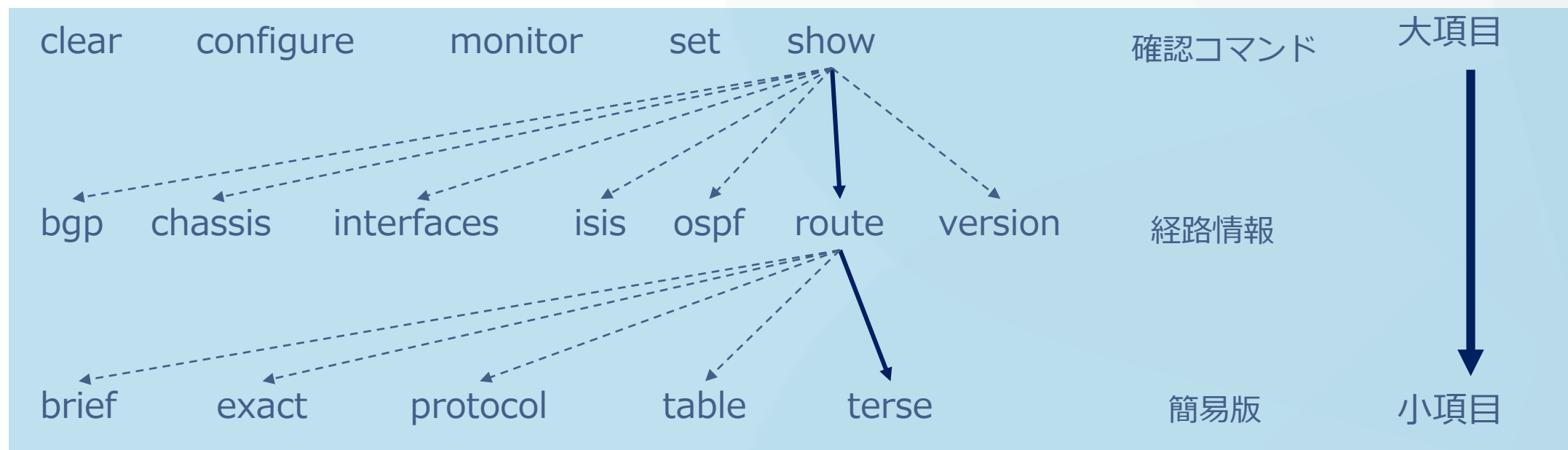

Operationalモード

- Operationalモードではステータスの確認やシステム操作などに用いるコマンドを提供しています。

```
clear          Clear information in the system
configure    Manipulate software configuration information
file          Perform file operations
help         Provide help information
monitor     Show real-time debugging information
mtrace       Trace multicast path from source to receiver
op           Invoke an operation script
ping       Ping remote target
quit        Exit the management session
request    Make system-level requests
restart     Restart software process
set         Set CLI properties, date/time, craft interface message
show      Show system information
ssh        Start secure shell on another host
start      Start shell
telnet     Telnet to another host
test       Perform diagnostic debugging
traceroute Trace route to remote host
```

Operationalモード

- コマンドは階層構造になっています
 - 例: 経路情報(簡易版)を確認



```
root> show route terse
inet.0: 3 destinations, 3 routes (3 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

A V Destination          P Prf  Metric 1   Metric 2   Next hop          AS path
* 0.0.0.0/0              S   5           >172.27.112.1
* 172.27.112.0/22       D   0           >ge-0/0/0.0
* 172.27.113.19/32     L   0           Local
```

Configurationモード

- Operationalモードにてconfigureと投入することでConfigurationモードへ移行します

```
root@lab> configure
  Entering configuration mode
  [edit]
  root@lab#
```

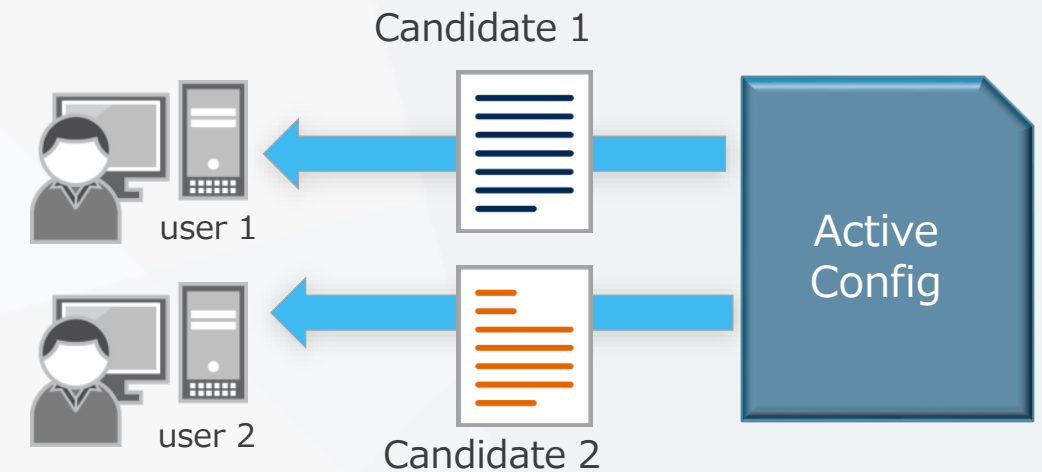
- 他のユーザがconfigurationモードに入っていれば、以下の様に表示されます

```
root@lab> configure
  Entering configuration mode
  Current configuration users:
    fbrooks terminal d0 on since 1999-10-14 07:11:29 UTC,
    idle 00:00:49 [edit protocols ospf]
  The configuration has been changed but not committed
  [edit]
  root@lab#
```

Configurationモード：オプション

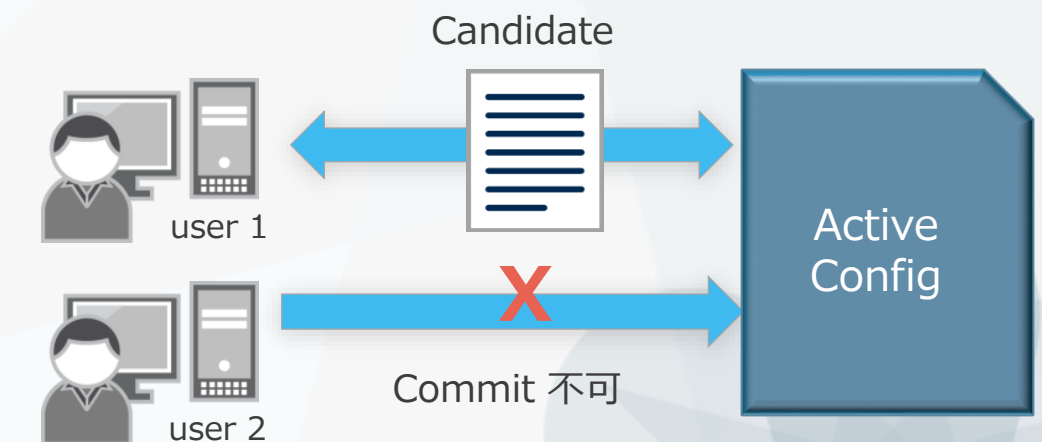
- **configure private** コマンドを使用すると、ログインユーザー専用のcandidate configurationが用意される

```
mike@jnpr1> configure private  
warning: uncommitted changes will be discarded on exit  
Entering configuration mode
```



- **configure exclusive** コマンドを使用すると、ログインユーザーが設定変更を行っている最中に他のログインユーザーが設定変更を行うことを禁止することが可能

```
mike@jnpr1> configure exclusive  
warning: uncommitted changes will be discarded on exit  
Entering configuration mode
```





JUNOS CLI操作 ～Operationalモード～

show コマンド

- **show**コマンド: システム、ステータスに関する情報を表示します
 - > show arp : ARPテーブルを確認する
 - > show chassis environment : 温度、ファンなどの環境状態を確認する
 - > show chassis hardware : ハードウェア情報 (シリアルナンバー等) を確認する
 - > **show chassis routing-engine** : ルーティングエンジン (CPUやMemory) の状態を確認する
 - > show configuration : 稼働中の設定を確認する
 - > **show interfaces** : Interfaceの状態を確認する
 - > **show route** : 経路情報を確認する
 - > show system uptime : 稼働時間を確認する
 - > show system users : ユーザのログイン状況を確認する
 - > show system alarms : システムアラームの有無を確認する
 - > show version : JUNOSソフトウェアバージョンを確認する

show コマンド: オプション

- showコマンドでは`terse`, `brief`, `detail`, もしくは`extensive`オプションを使用することで確認できる情報量を選択することができます。
- `terse`, `brief`のオプションはオプションなしの出力結果と比べ、より簡易的な情報を表示させます。
- `detail`, `extensive`のオプションはオプションなしの際と比べ、より詳細な情報を表示させます。

show コマンド: オプション

> show interfaces ge-0/0/0 **terse**

```
> show interfaces ge-0/0/0 terse
```

Interface	Admin	Link	Proto	Local	Remote
ge-0/0/0	up	up			

> show interfaces ge-0/0/0 **brief**

```
> show interfaces ge-0/0/0 brief
```

```
Physical interface: ge-0/0/0, Enabled, Physical link is Up  
  Link-level type: Ethernet, MTU: 1514, Speed: Auto, Duplex: Auto, Loopback: Disabled, Source filtering:  
Disabled, Flow control: Enabled, Auto-negotiation: Enabled, Remote fault: Online, Media type: Copper,  
  IEEE 802.3az Energy Efficient Ethernet: Disabled  
Device flags      : Present Running  
Interface flags: SNMP-Traps Internal: 0x4000  
Link flags       : None
```


show コマンド: オプション

> show interfaces ge-0/0/2 (オプションなし)

```
> show interfaces ge-0/0/0
```

```
Physical interface: ge-0/0/0, Enabled, Physical link is Up  
Interface index: 139, SNMP ifIndex: 504  
Link-level type: Ethernet, MTU: 1514, Speed: Auto, Duplex: Auto, BPDU Error: None, MAC-REWRITE Error:  
None, Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,  
Remote fault: Online, Media type: Copper, IEEE 802.3az Energy Efficient Ethernet: Disabled  
Device flags      : Present Running  
Interface flags: SNMP-Traps Internal: 0x4000  
Link flags       : None  
CoS queues       : 8 supported, 8 maximum usable queues  
Current address: 5c:5e:ab:7e:75:c3, Hardware address: 5c:5e:ab:7e:75:c3  
Last flapped    : 2016-02-16 18:44:29 JST (8w1d 17:33 ago)  
Input rate      : 0 bps (0 pps)  
Output rate     : 0 bps (0 pps)  
Active alarms   : None  
Active defects  : None  
Interface transmit statistics: Disabled
```

show コマンド: オプション

> show interfaces ge-0/0/0 detail

```
> show interfaces ge-0/0/0 detail
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 139, SNMP ifIndex: 504, Generation: 142
  Link-level type: Ethernet, MTU: 1514, Speed: Auto, Duplex: Auto, BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled, Auto-negotiation: Enabled,
  Remote fault: Online, Media type: Copper, IEEE 802.3az Energy Efficient Ethernet: Disabled
  Device flags      : Present Running
  Interface flags: SNMP-Traps Internal: 0x4000
  Link flags       : None
  CoS queues       : 8 supported, 8 maximum usable queues
  Hold-times       : Up 0 ms, Down 0 ms
  Current address: 5c:5e:ab:7e:75:c3, Hardware address: 5c:5e:ab:7e:75:c3
  Last flapped    : 2016-02-16 18:44:29 JST (8w1d 17:37 ago)
  Statistics last cleared: Never
  Traffic statistics:
    Input bytes   :          995586          0 bps
    Output bytes  :         1473366          0 bps
    Input packets :          10870          0 pps
    Output packets:         15732          0 pps
  IPv6 transit statistics:
    Input bytes   :              0
    Output bytes  :              0
    Input packets :              0
    Output packets:              0
  Egress queues: 8 supported, 4 in use
  Queue counters:
    Queued packets  Transmitted packets  Dropped packets
    0 best-effort   0                9262             0
    1 assured-forw  0                0                0
    5 expedited-fo  0                0                0
    7 network-cont  0                6470             0
  Queue number:
    Mapped forwarding classes
    0 best-effort
    1 assured-forwarding
    5 expedited-forwarding
    7 network-control
  Active alarms   : None
  Active defects  : None
  Interface transmit statistics: Disabled
```

show コマンド: オプション

> show interfaces ge-0/0/0 extensive

```
> show interfaces ge-0/0/0 extensive
Physical interface: ge-0/0/0, Enabled, Physical link is Up
Interface index: 139, SNMP ifIndex: 504, Generation: 142
Link-level type: Ethernet, MTU: 1514, Speed: Auto, Duplex: Auto, BPDU Error: None, MAC-
REWRITE Error: None, Loopback: Disabled, Source filtering: Disabled, Flow control: Enabled,
Auto-negotiation: Enabled,
Remote fault: Online, Media type: Copper, IEEE 802.3az Energy Efficient Ethernet:
Disabled
Device flags      : Present Running
Interface flags: SNMP-Traps Internal: 0x4000
Link flags       : None
CoS queues       : 8 supported, 8 maximum usable queues
Hold-times       : Up 0 ms, Down 0 ms
Current address: 5c:5e:ab:7e:75:c3, Hardware address: 5c:5e:ab:7e:75:c3
Last flapped    : 2016-02-16 18:44:29 JST (8w1d 17:40 ago)
Statistics last cleared: Never
Traffic statistics:
Input bytes      :          995586          0 bps
Output bytes     :         1473366          0 bps
Input packets    :           10870          0 pps
Output packets   :           15732          0 pps
IPv6 transit statistics:
Input bytes      :          0
Output bytes     :          0
Input packets    :          0
Output packets   :          0
Input errors:
Errors: 0, Drops: 0, Framing errors: 0, Runts: 0, Policed discards: 0, L3 incompletes:
0, L2 channel errors: 0, L2 mismatch timeouts: 0, FIFO errors: 0, Resource errors: 0
Output errors:
Carrier transitions: 3, Errors: 0, Drops: 0, Collisions: 0, Aged packets: 0, FIFO
errors: 0, HS link CRC errors: 0, MTU errors: 0, Resource errors: 0
Egress queues: 8 supported, 4 in use
Queue counters:
Queue counters:   Queued packets  Transmitted packets  Dropped packets
0 best-effort    0          9262          0
1 assured-forw   0          0          0
5 expedited-fo   0          0          0
7 network-cont   0          6470         0
Queue number:    Mapped forwarding classes
0                best-effort
1                assured-forwarding
5                expedited-forwarding
7                network-control
Active alarms   : None
Active defects  : None
MAC statistics:
Total octets    Receive          Transmit
Total packets   995586          1473366
Unicast packets 10870           15732
Broadcast packets 8989            9262
Multicast packets 1876            1872
CRC/Align errors 5                4598
FIFO errors      0                0
MAC control frames 0                0
MAC pause frames 0                0
Oversized frames 0
Jabber frames    0
Fragment frames 0
Code violations  0
Autonegotiation information:
Negotiation status: Complete
Link partner:
Link mode: Full-duplex, Flow control: Symmetric, Remote fault: OK,
Link partner Speed: 1000 Mbps
Local resolution:
Flow control: Symmetric, Remote fault: Link OK
Packet Forwarding Engine configuration:
Destination slot: 0 (0x00)
CoS information:
Direction : Output
CoS transmit queue      Bandwidth      Buffer Priority
Limit                   %              bps            %              usec
0 best-effort           95             950000000      95             NA            low
none
7 network-control       5              50000000       5              NA            low
none
Interface transmit statistics: Disabled
```

コンソール画面出力に関する操作

- 画面に **---(more)---** promptが表示されているときは以下のキーで操作します

Space:	次画面に進む
b:	前画面に戻る
d:	1/2画面進む
Enter:	1行進む
/string:	検索
n:	再検索
q:	プロンプトに戻る (出力のAbort)
h:	これらキーヘルプの表示

```
> show configuration
## Last commit: 2016-04-12 17:41:17 JST by lab
version 12.3X48-D20.4;
groups {
    Japan_ENT_POC {
        system {
            host-name mino_srx240;
            backup-router 172.27.112.1;
            time-zone Asia/Tokyo;
            dump-on-panic;
            root-authentication {
                encrypted-password
"$1$YrXW4H1t$0Ry0FagjB/wMKbVM1izGf/"; ## SECRET-DATA
            }
            name-server {
                208.67.222.222;
                208.67.220.220;
            }
            login {
                user lab {
                    uid 2000;
                    class super-user;
                    authentication {
--- (more) ---
```

コンソール画面出力に関する操作 | no-more

- 通常、出力はCLIのスクリーンサイズを考慮して行われます。出力内容が多い場合、CLI画面に---(more)---を表示し、出力を一時停止します。ログ取得時などは“ | no-more” オプションを使用し、全て一度に表示することが可能です

```
[edit]
root@lab> show configuration | no-more
## Last commit: 2016-01-25 11:25:54 JST by root
version 10.4R7.5;
groups {
    Japan_team {
        system {
            backup-router 172.16.1.1;
            time-zone Asia/Tokyo;
            dump-on-panic;
            root-authentication {
                encrypted-password "$1$YrXW4H1t$0Ry0FagjB/wMKbVM1izGf/";
            }
            ## SECRET-DATA
        }
        login {
            user lab {
                uid 2000;
                "$1$4TDfGIs7$.wTBpcNviWRCebbvcSLzv."; ## SECRET-DATA
            }
        }
    }
}
```

パイプ “|” オプションの利用

- Unix同様のパイプ “|” をサポート。configやshowコマンド等で有効利用
 - root@lab> show configuration | **display set**
 - root@lab> show log messages | **no-more**
 - root@lab> show route | **find** 192.168.1.0
 - root@lab# show interface | **save** interface_config.txt

```
root@lab> show configuration | ?
Possible completions:
  compare      Compare configuration changes with prior version
  count        Count occurrences
  display      Show additional kinds of information
  except       Show only text that does not match a pattern
  find         Search for first occurrence of pattern
  hold         Hold text without exiting the --More-- prompt
  last         Display end of output only
  match        Show only text that matches a pattern
  no-more      Don't paginate output
  request      Make system-level requests
  resolve      Resolve IP addresses
  save         Save output text to file
  trim         Trim specified number of columns from start of line
```

パイプ “|” 使用例

- Configurationの表示方法を変更する
 - 階層表記に加え、行単位での表示も可能

```
root@EX2200C> show configuration protocols dot1x
traceoptions {
  file 1x;
  flag all;
}
authenticator {
  authentication-profile-name dot-1x;
  interface {
    ge-0/0/0.0 {
      supplicant single-secure;
      reauthentication 3600;
    }
    ge-0/0/3.0 {
      supplicant single-secure;
    }
  }
}
```

```
root@EX2200C> show configuration protocols dot1x |display set
set protocols dot1x traceoptions file 1x
set protocols dot1x traceoptions flag all
set protocols dot1x authenticator authentication-profile-name
  dot-1x
set protocols dot1x authenticator interface ge-0/0/0.0
  supplicant single-secure
set protocols dot1x authenticator interface ge-0/0/0.0
  reauthentication 3600
set protocols dot1x authenticator interface ge-0/0/3.0
  supplicant single-secure
set protocols dot1x authenticator interface
```

↑ ↑
状況に応じ、お好みの表記方法を選択可能

パイプ “|” 使用例

- Configurationの一部を保存する
- 稼働中のconfigurationの方法

Operationalモードにて **show configuration | save** <出力先+ファイル名>

```
> show configuration | save ftp://abc@172.xx.xxx.xx/Ex_config
Password for abc@172.xx.xxx.xx:
ftp://abc@172.xx.xxx.xx/mx_config      100% of 7928 B   30 MBps
Wrote 352 lines of output to 'ftp://abc@172.xx.xxx.xx/Ex_config'
```

← FTPサーバへ出力

- 編集中のconfigurationの出力方法

Configurationモードにて **save** <出力先+ファイル名>

```
# save /config/EDITING-CONFIG
Wrote 232 lines of configuration to '/config/EDITING-CONFIG'
```

← /config/へ出力

※保存先を指定しない場合、userのhome directoryに出力されます

JUNOSファイルシステムの構成について

- JUNOSでは各種構成ファイルやLogファイルなどをファイルシステム上のディレクトリに管理しています

/config

使用中のコンフィグレーションと過去3世代までのコンフィグレーションを格納。

/var/db/config

4世代以降のコンフィグレーションを格納。gz形式に圧縮されて保存されているがfile showコマンドで表示可能。FreeBSDではzcatコマンドで表示可能。

/var/tmp

JUNOSソフトウェアアップグレード時など、image格納するディレクトリ。また、各デーモンのコアダンプファイルを格納。

/var/log

各種LogやTrace option機能にて取得したデバッグ情報ファイルを格納。

/var/home

各ユーザのホームディレクトリが作成される。

各ユーザがローカルに保存した情報は全て各ユーザのホームディレクトリに格納する。

例えば、現在使用中のコンフィグをsaveコマンドにて保存した場合など

JUNOSファイルシステムの構成について

- 各ディレクトリに格納しているファイルの確認方法

> **file list** / <directory>/

```
root> file list /var/home/  
/var/home/:  
SAMPLE/
```

← /var/home配下の情報を表示
ユーザ(SAMPLE)のホームディレクトリが作成されている

```
root> file list /var/home/SAMPLE/  
/var/home/SAMPLE/:  
TEST_CONFIG
```

← /var/home/SAMPLE配下の情報を表示
ユーザ(SAMPLE)が作成したTEST_CONFIGが保存されている

- ディレクトリ配下のファイル内容の確認方法

> **file show** /<directory>/<file_name>

```
root> file show /var/home/SAMPLE/TEST_CONFIG  
## Last changed: 2016-03-30 17:34:10 UTC  
version 12.3X48-D25.3;  
system {  
  root-authentication {  
    encrypted-password "$1$73UgjTsC$EznYXp/4DfJIRTI6KnFYE1"; ## SECRET-DATA
```

← ユーザ(SAMPLE)が作成した
TEST_CONFIGを確認

~~~~~以下省略~~~~~

# JUNOS運用管理コマンド

- JUNOSでは運用管理に必要な機能をサポートしています。
  - Ping
  - Traceroute
  - telnet / ssh
  - Monitor

Ping : ネットワークの疎通確認をする

> **ping** アドレス + オプション

例: 172.27.112.1へ512 byteのpingを3回実施

```
root> ping 172.27.112.1 count 3 size 512
PING 172.27.112.1 (172.27.112.1): 512 data bytes
520 bytes from 172.27.112.1: icmp_seq=0 ttl=64 time=1.037 ms
520 bytes from 172.27.112.1: icmp_seq=1 ttl=64 time=0.704 ms
520 bytes from 172.27.112.1: icmp_seq=2 ttl=64 time=0.741 ms

--- 172.27.112.1 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.704/0.827/1.037/0.149 ms
```

# JUNOS運用管理コマンド

Traceroute : ネットワークの経路確認をする

> **traceroute** アドレス + オプション

例: 8.8.8.8へge-0/0/0からtrace routeを実施

```
> traceroute 8.8.8.8 interface ge-0/0/0
traceroute to 8.8.8.8 (8.8.8.8), 30 hops max, 40 byte packets
 1  172.27.112.2 (172.27.112.2)  4.394 ms  1.814 ms  2.209 ms
(snip)
13  google-public-dns-a.google.com (8.8.8.8)  4.276 ms  4.286 ms  4.063 ms
```

Telnet / SSH : ネットワークに接続された機器を操作する

> **telnet** アドレス + オプション

例: 172.27.112.161: port 23へtelnetを実施

```
> telnet 172.27.112.161 port 23
Trying 172.27.112.161...
Connected to 172.27.112.161.
Escape character is '^]'.

srx300beta (ttyp0)

login:
```

# monitor コマンド

- **monitor**コマンド: 現在のI/F別トラフィック状況を表示します
  - > **monitor interface traffic**

各Interfaceのトラフィックをリアルタイム表示する

```
Bytes=b, Clear=c, Delta=d, Packets=p, Quit=q or ESC, Rate=r, Up=^U, Down=^D
Interface      Link  Input packets      (pps)      Output packets      (pps)
ge-0/0/0       Down  0                   (0)         0                   (0)
gr-0/0/0       Up    0                   (0)         0                   (0)
ip-0/0/0       Up    0                   (0)         0                   (0)
lsq-0/0/0      Up    0                   (0)         0                   (0)
lt-0/0/0       Up    0                   (0)         0                   (0)
mt-0/0/0       Up    0                   (0)         0                   (0)
sp-0/0/0       Up    0                   (0)         0                   (0)
ge-0/0/1       Down  0                   (0)         0                   (0)
(snip)
```

# requestコマンド

- **request**コマンド： システムの挙動に関するコマンドを実行します

- システムを再起動する

```
> request system reboot
```

- システムをシャットダウンする

```
> request system power-off
```

- 初期化する

```
> request system zeroize
```

- サポートに必要な情報を取得する

```
> request support information
```

- 基本となるConfigurationファイルを保存する (rescue configの保存)

```
> request system configuration rescue save
```

- OSをアップグレードする

```
> request system software add <ファイル名>
```

# JUNOSのソフトウェアアップグレード

- ソフトウェアアップグレード手順

1. 対象のJUNOS OSをダウンロードする。

<https://www.juniper.net/support/downloads/group/?f=junos>

2. CLICOMMANDでJUNOSソフトウェアをFTP/TFTPサーバからデバイス(/var/tmp)に保存

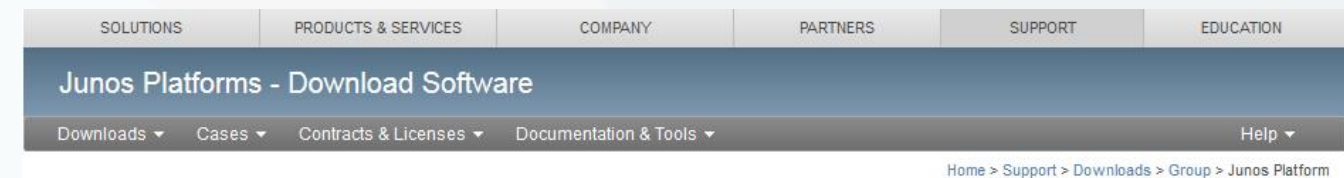
```
file copy ftp://ログインID@アドレス/JUNOSパッケージ名/var/tmp
```

3. デバイスに保存したパッケージをロード

```
request system software add /var/tmp/JUNOSパッケージ名
```

4. 再起動する

```
request system reboot
```



```
root> file copy ftp://abc@172.xx.xxx.xx/jinstall-ex-4200-11.4R1.6-domestic-signed.tgz /var/tmp
Password for abc@172.xx.xxx.xx:
/var/tmp//...transferring.file.....TfBx6L/100% of
381 MB 8099 kBps 00m00s

root> request system software add /var/tmp/ jinstall-
ex-4200-11.4R1.6-domestic-signed.tgz
NOTICE: Validating configuration against jinstall-ex-
4200-11.4R1.6-domestic-signed.tgz.
(snip)

root> request system reboot
```

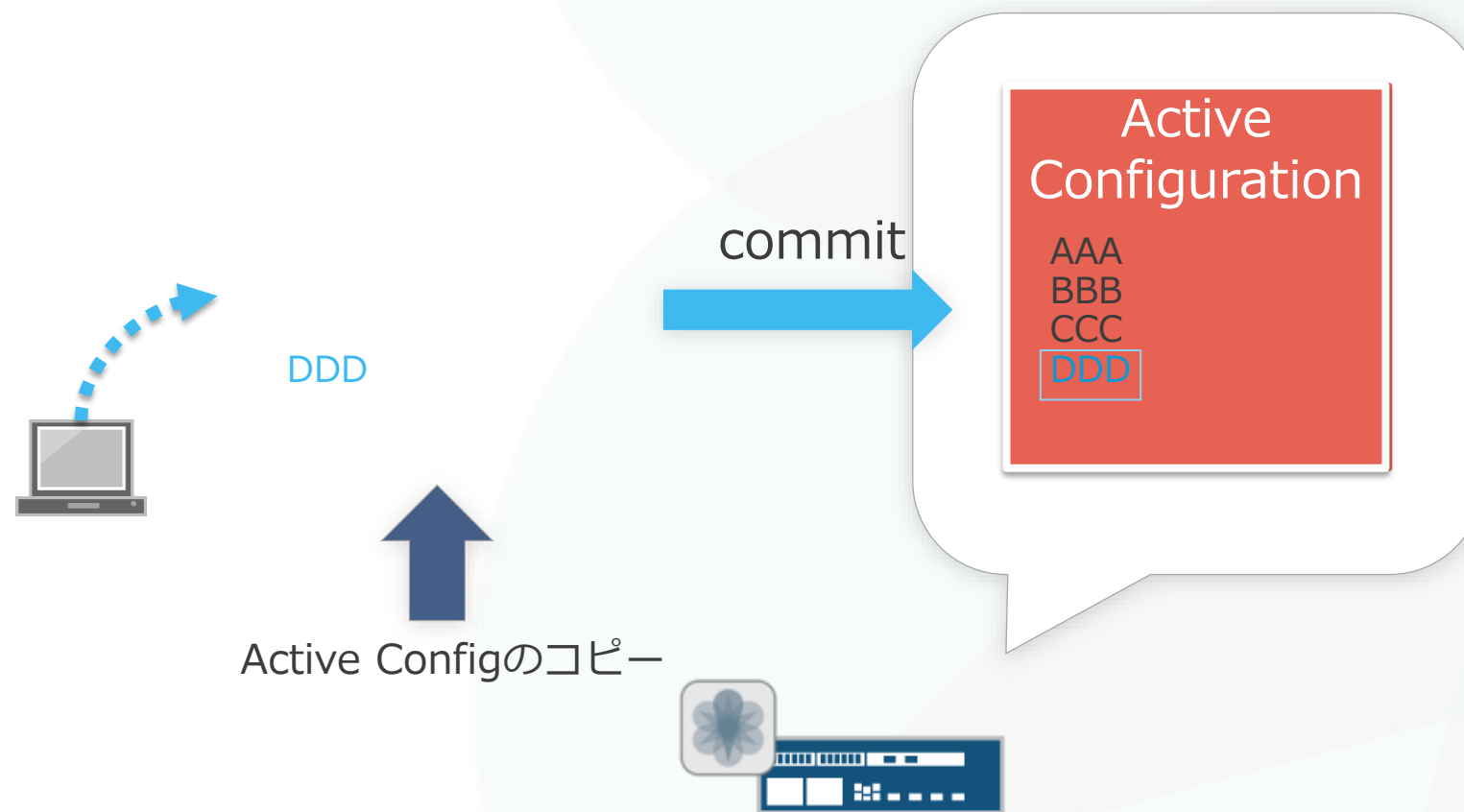


# JUNOS CLI操作 ～Configurationモード～



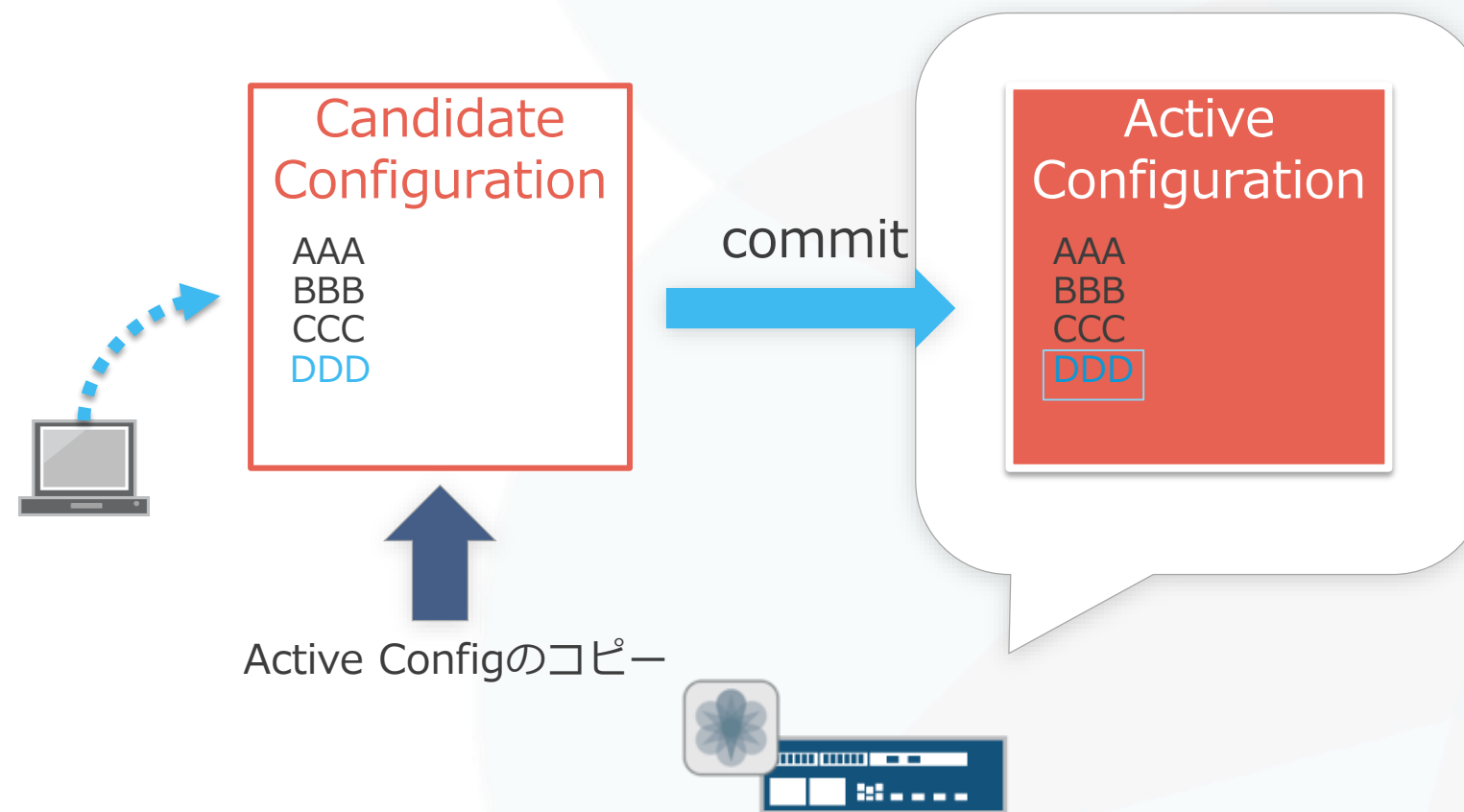
# commit コンセプト (アニメ)

- Configurationモードに入ると編集用configが用意されます
- 設定変更はすべて編集用のconfig上へのみ投入
- commit コマンドでactive configに反映されます



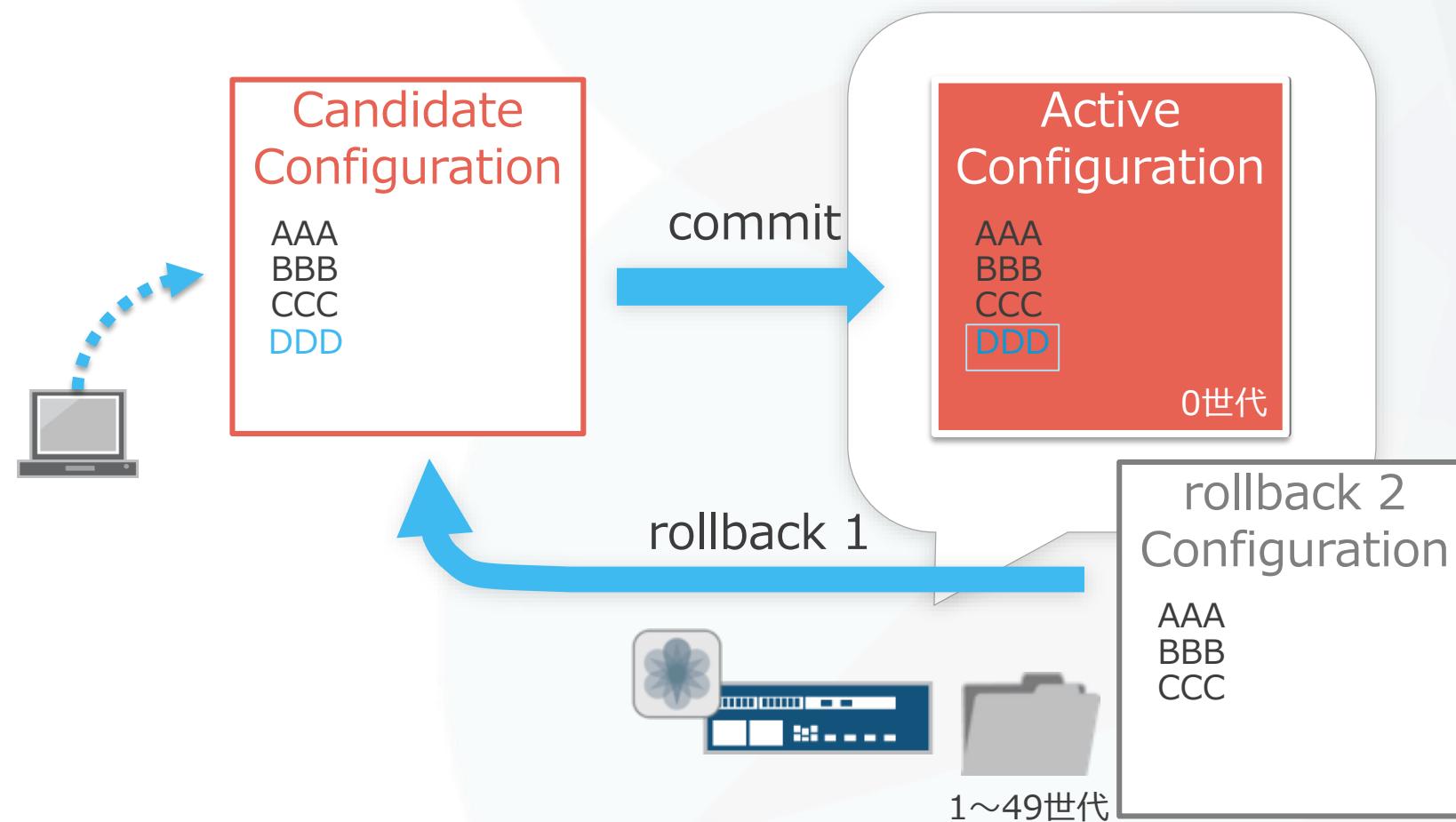
# commit コンセプト

- Configurationモードに入ると編集用configが用意されます
- 設定変更はすべて編集用のconfig上へのみ投入
- commit コマンドでactive configに反映されます



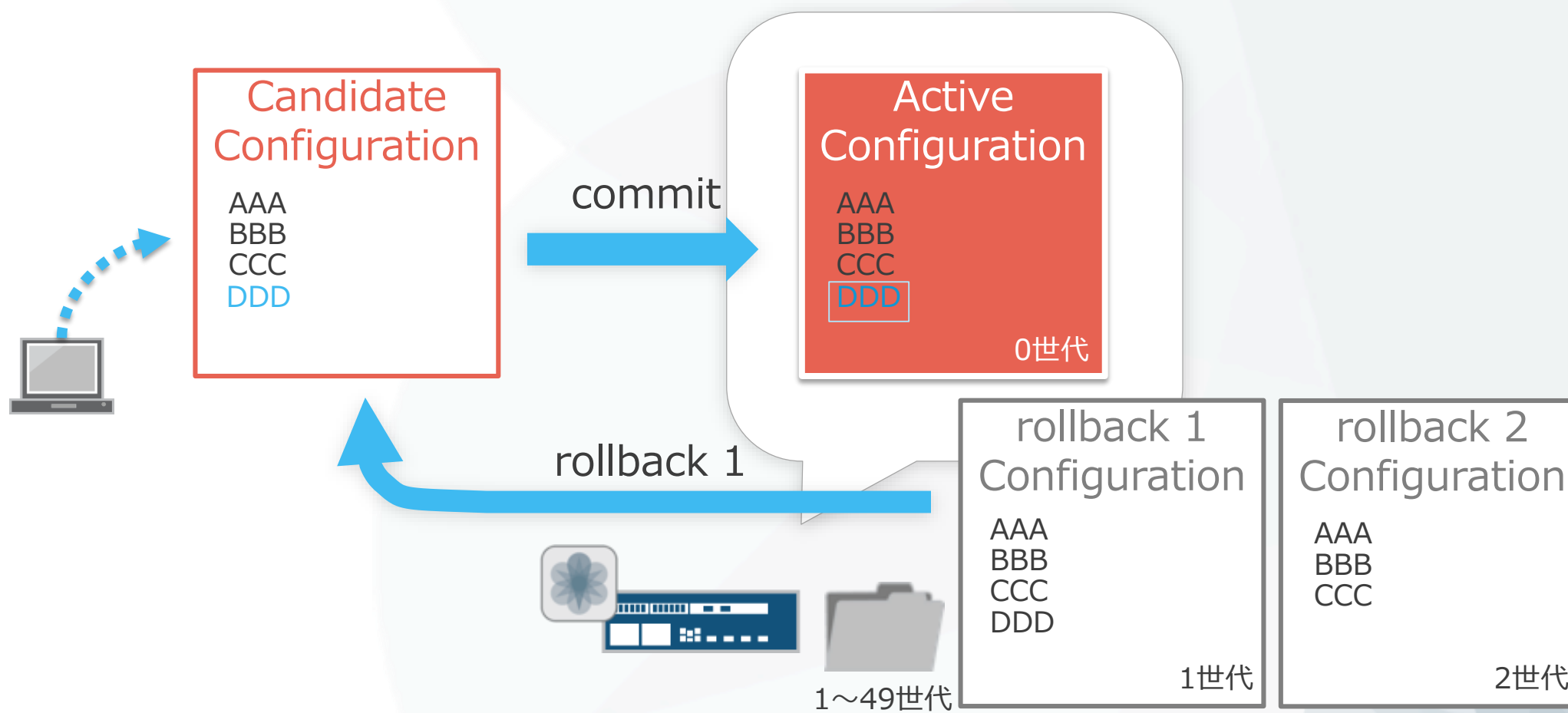
# rollback コンセプト (アニメ)

- commit実行時に、現在稼働中のconfigが履歴として自動的に保存されます
- commit前の状態に戻すときは、rollback 1 コマンドで1世代前をロード
- 再度commit コマンドを実行して、active configに反映



# rollback コンセプト

- commit実行時に、現在稼働中のconfigが履歴として自動的に保存されます
- commit前の状態に戻すときは、rollback 1 コマンドで1世代前をロード
- 再度commit コマンドを実行して、active configに反映



# 設定の追加 (set)

- **set** コマンド：設定の追加変更を行います
  - Commitするまでは設定は反映されません。

```
user@lab# set interface ge-0/0/1 disable
```

- Commitすることで初めて動作しているデバイスに設定追加の変更が反映されます。

```
user@lab# commit
configuration check succeeds

commit complete
```

# 設定の変更 (delete)

- **delete** コマンド：設定の削除変更を行います
  - Commitするまでは設定は反映されません。

```
user@lab# delete interface ge-0/0/1 disable
```

- やはりCommitすることで動作しているデバイスに設定削除の変更が反映されます。

```
user@lab# commit
configuration check succeeds

commit complete
```

# 編集集中の設定確認 (show | compare)

- `show | compare` コマンド : 編集集中の設定と稼動中の設定を比較します

```
user@lab# set interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/24
user@lab# show | compare
[edit interfaces]
+   ge-0/0/0 {
+     unit 0 {
+       family inet {
+         address 192.168.1.1/24;
+       }
+     }
+   }
```

Active configと比較して、ge-0/0/0にIPアドレスが追加されている  
+ : 追加  
- : 削除

- 過去のconfigと編集集中の設定を比較することも可能

```
user@lab# show | compare rollback [1-49]
```



# 設定ファイルの復旧 (rollback)

- **rollback** コマンド：設定ファイルの復旧を行います
  - 変更した設定ファイルを破棄したい場合は、Rollbackコマンドを投入します。(rollbackはrollback 0の略)

```
user@lab# rollback
```

- rollback n (0-49) でファイル番号を指定すると、過去の設定をCandidate Configにコピーすることが可能で、容易に過去の状態に戻すことが可能です。(過去50世代分の設定ファイルを自動保存)

```
user@lab# rollback ?
Possible completions:  <[Enter]>          Execute this command
0                    2016-07-14 08:41:21 JST by root via cli
1 2016-07-13 16:01:54 JST by root via cli
2 2016-07-13 15:59:51 JST by root via cli
3 2016-07-13 15:57:33 JST by root via cli
4 2016-07-13 15:57:20 JST by root via cli commit confirmed, rollback in 2mins
5 2016-07-12 15:21:37 JST by lab via netconf
6 2016-07-08 16:35:39 JST by lab via cli
7 2016-06-22 19:30:53 JST by lab via cli
8 2016-06-22 19:28:39 JST by lab via cli
9 2016-06-22 19:28:18 JST by lab via cli
...(snip)
```



# commit オプション (commit confirmed / at)

- **commit confirmed** コマンド：一時的に設定を反映します
  - 変更を確定する場合は、時間内にcommitを実行します
    - デフォルトでは10分（指定可能）
    - 時間までにcommitされなければ、自動的にcommit前の状態に戻る

```
user@lab# commit confirmed 5
configuration check succeeds
commit confirmed will be automatically rolled back in 5 minutes unless confirmed
commit complete

# commit confirmed will be rolled back in 5 minutes
```

- **commit at** コマンド：日時を指定してcommitの実行を予約します
  - hh:mm:[ss] または "yyyy-mm-dd hh:mm:[ss]"

```
user@lab# commit at "2016-04-20 00:00"
configuration check succeeds
commit at will be executed at 2016-04-20 00:00:00 JST
Exiting configuration mode
```

※予約をキャンセルしたいときは、Operationalモードからclear system commit コマンドを実行

# Configurationのロード (load)

- **load** コマンド : configurationファイルをロードします
  - loadコマンドはいくつかのオプションがあります
    - **load factory-default** 工場出荷時のconfigをロード
    - **load override <filename>** ロードしたconfigによる置き換え
    - **load merge <filename>** ロードしたconfigを追加

```
user@lab# load ?
Possible completions:
  factory-default  Override existing configuration with factory default
  merge            Merge contents with existing configuration
  override         Override existing configuration
  patch           Load patch file into configuration
  replace         Replace configuration data
  set             Execute set of commands on existing configuration
  update          Update existing configuration
```

- configファイルは外部のFTPサーバや機器内ディレクトリからロードすることも可能

```
user@lab# load merge /var/tmp/saved_config.txt
user@lab# load merge ftp://user:passwd@192.168.1.1/saved_config.txt
```

# Configurationのロード (load set terminal)

- **load set terminal** コマンド : CLIで追加のsetコンフィグを貼り付けるときに使用
  - setコマンドの大量コピー&ペースト時にconfigのとりこぼしが防げます

```
user@lab# load set terminal
[Type ^D at a new line to end input]
set services security-intelligence profile feeds-cc-p1 category CC
set services security-intelligence profile feeds-cc-p1 default-rule then action permit
set services security-intelligence profile feeds-cc-p1 default-rule then log
set services security-intelligence profile Inf-hosts category Infected-Hosts
set services security-intelligence profile Inf-hosts default-rule then action permit
set services security-intelligence profile Inf-hosts default-rule then log
set services security-intelligence policy pol-cc CC feeds-cc-p1
set services security-intelligence policy pol-cc Infected-Hosts Inf-hosts
set services advanced-anti-malware policy skyatp_test match application HTTP
set services advanced-anti-malware policy skyatp_test match verdict-threshold 3
set services advanced-anti-malware policy skyatp_test then action permit
set services advanced-anti-malware policy skyatp_test then notification log
set services advanced-anti-malware policy skyatp_test inspection-profile test
set services advanced-anti-malware policy skyatp_test fallback-options action permit
set services advanced-anti-malware policy skyatp_test fallback-options notification log
set services advanced-anti-malware policy skyatp_test whitelist-notification log
set services advanced-anti-malware policy skyatp_test blacklist-notification log
```

貼り付けたいconfigをterminal上でペーストし、最後に改行してからCTRL+Dを押して読み込む

キャンセルしたい場合はCTRL+Cで抜ける

CTRL+D → load complete

# Configurationのロード (load merge terminal)

- **load merge terminal** コマンド : CLIで追加のconfigを貼り付けるときに使用
  - 大量のコピー&ペースト時にもconfigのとりこぼしが防げます、最上位の階層から追加のconfigを投入する階層までのパスが全部必要です
  - relative オプションを付けると今いる階層に応じてconfigの階層もショートカットされます

```
[edit]
user@lab# load merge terminal
[Type ^D at a new line to end input]
protocols {
  ospf {
    export static-route;
    area 0.0.0.0 {
      interface ge-0/0/0.0;
      interface ge-0/0/1.0;
      interface ge-0/0/2.0;
      interface lo0.0 {
        passive;
      }
    }
  }
}
policy-options {
  policy-statement static-route {
    from {
      protocol static;
      route-filter 10.1.1.0/24 longer;
    }
    then accept;
  }
}
```

interfaces, protocolsやpolicy-optionsなど最上位の構文から記述していく

CTRL+D → load complete

```
[edit protocols ospf]
lab# load merge terminal relative
[Type ^D at a new line to end input]
area 0.0.0.0 {
  interface xe-1/0/0.0;
}
area 0.0.0.1 {
  stub default-metric 10 no-summaries;
  area-range 192.168.16.0/20;
  interface ge-0/0/3.0;
}
area 0.0.0.2 {
  nssa {
    default-lsa {
      default-metric 20;
      metric-type 1;
      type-7;
    }
    no-summaries;
    area-range 172.16.12.0/22;
  }
  area-range 192.168.48.0/20;
}
load complete
```

protocols ospfの階層に移動しareaのconfigだけ追加

protocols { ospf { の記述は不要

CTRL+D →

# Configurationモード：コマンドサマリー

- 設定&確認コマンド
  - `set` : パラメータを設定する際に使用します
  - `delete` : パラメータを削除する際に使用します
  - `show` : 設定した内容を確認します
  - `show | compare` : 編集中のconfigと稼働中のconfigを比較します
- 設定反映コマンド
  - `commit` : 編集した設定をactive configに反映させます
  - `rollback` : 過去のconfigをロードして編集内容を元に戻します
  - `load` : 設定したファイルをロードする際に使用します

# 便利なショートカットキー

- カーソルの移動

Ctrl-B      1文字戻る

Ctrl-F      1文字進む

Ctrl-A      行頭に移動

Ctrl-E      行末に移動

- 文字の削除

Delete/Backspace      カーソル前の1文字を削除

Ctrl-D      カーソル後の1文字を削除

Ctrl-K      カーソルから行末までを削除

Ctrl-U      行をすべて削除

Ctrl-W      現在入力途中の単語または、カーソルより左側の1単語を削除

- その他

Ctrl-P or ↑      コマンド履歴の前を表示

Ctrl-N or ↓      コマンド履歴の次を表示

?      次に入力すべきコマンドやパラメータのヒント

# コマンド補完と構文エラー

- コマンド補完機能

- Spaceキー/ Tabキー：固定値を補完
  - Tabキーはユーザが定義したpolicy名やFilter名の補完も可能

```
user@lab# set interfaces ge-0/0/0 unit 0 family inet filter input ?
Possible completions:
  TEST [firewall family inet filter]

user@lab# set interfaces ge-0/0/0 unit 0 family inet filter input T[tab]
```

- 構文エラーの通知

- 構文に誤りがあるとsyntax errorと表示される
- ^ マークはエラーとなる項目を示す

```
user@lab# load replase
                ^
syntax error, expecting <command>.
```

# Configurationモード: Operationalモードのコマンドを実行

- **run**コマンドにより、Configurationモードにおいてshowコマンド等を実行し、status等確認することができます
  - Operationalモードで確認可能な全てのコマンドの実行が可能
  - Operationalモードに戻る必要なし

runコマンドを使用し、interfaceの状態を確認

```
root@lab# run show interfaces
Physical interface: ge-0/0/0, Enabled, Physical link is Up
  Interface index: 134, SNMP ifIndex: 508
  Link-level type: Ethernet, MTU: 1514, Link-mode: Full-duplex, Speed:
100mbps,
  BPDU Error: None, MAC-REWRITE Error: None, Loopback: Disabled,
  Source filtering: Disabled, Flow control: Disabled, Auto-
negotiation: Enabled,
  Remote fault: Online
  Device flags      : Present Runnin

(snip)
```

interfaceの設定を確認

```
root@lab# show interfaces
ge-0/0/0 {
    unit 0;
}
ge-0/0/1 {
    unit 0 {
        family ethernet-
switching {
            vlan {
                members
            }
        }
    }
}
vlan-trust;

(snip)
```





# JUNOSシステム設定

# システム設定

- JUNOSデバイスのシステムに関する主な設定
  - ユーザ設定
  - ホスト名の設定
  - 時刻設定
  - DNS設定
  - デバイスのサービス設定
  - 管理インタフェース設定
  - ログの設定
  - SNMP設定

# システム設定

- ユーザ設定

- rootユーザのパスワードを設定

```
root# set system root-authentication plain-text-password
New password:
Retype new password:
```

- rootユーザ以外のユーザアカウントを作成

- デフォルトでは3つのユーザクラスを選択可能

- read-only : view (show コマンドなど)
- operator : clear, network, reset, trace, view (デーモンの停止, ping/telnet, etc)
- super-user : all (すべて)

```
root# set system login user TEST class super-user authentication plain-text-password
New password:
Retype new password:
```

# システム設定

- ホスト名の設定

```
root# set system host-name LAB
```

- 時刻設定

- Time zoneを指定する

```
root# set system time-zone Asia/Tokyo
```

- NTPサーバを指定する

```
root# set system ntp server 10.10.10.100
```

- DNS設定

```
root# set system name-server 192.168.1.100
```

# システム設定

- デバイスのサービス設定
  - telnet, sshによるアクセスを有効にする

```
root# set system services telnet
root# set system services ssh
root# set system services ssh root-login allow
```

RootユーザとしてSSHでログインしたい場合に設定

- FTP, netconfのサービスを有効にする

```
root# set system services ftp
root# set system services netconf ssh
```

# システム設定

- 管理インタフェース設定

- 例 1 : EXの管理インタフェース(me0)を設定

```
root# set interfaces me0 unit 0 family inet address 192.168.1.1/24
```

- 例 2 : MX, SRXの管理インタフェース(fxp0)を設定

```
root# set interfaces fxp0 unit 0 family inet address 192.168.1.1/24
```

EX3300 rear view



SRX340 front view



※管理ポートは、  
MX/SRXは"FXP0"、EX/QFXは"ME0"、EX/QFXのVCでは"VME (Virtual ME) "と命名されています。  
Branch SRXのLow End (SRX300/320) など、Out of Bandの管理ポートが存在しないモデルもあります。

# システム設定

- ログの設定

- syslogサーバ、ファシリティ、ログレベルを指定
  - 例：すべてのレベルのログを10.10.10.1へ送信する

```
root# set system syslog host 10.10.10.1 any any
```

## ■ Syslogレベルについて

|   |            |                                    |
|---|------------|------------------------------------|
| 高 | emergency: | ソフトウェアコンポーネントの機能停止を招く状況のメッセージ      |
|   | alert:     | データベースなどのデータ破損など、直ちに修復が必要な状況のメッセージ |
|   | critical:  | 物理的なエラーなど重大な問題がある状況のメッセージ          |
|   | error:     | 上記よりも深刻度の低いエラー状況のメッセージ             |
|   | warning:   | モニタリングの必要性がある状況のメッセージ              |
|   | notice:    | エラーではないが、特別な処理が必要となる可能性がある状況のメッセージ |
|   | info:      | 対象のイベントまたは非エラー状況のメッセージ             |
| 低 | any:       | すべてのレベルのメッセージ                      |

# システム設定

- SNMP設定

- SNMPコミュニティを作成する

- 例：コミュニティ名をpublicに設定、読み込みのみ許可

```
root# set snmp community public authorization read-only
```

- SNMPトラップを設定する

- 例：トラップの送信元をLoopback 0に、宛先を10.10.10.1に設定

```
root# set snmp trap-options source-address lo0  
root# set snmp trap-group <group-name> targets 10.10.10.1
```





# JUNOSインタフェース設定

# インタフェースタイプの表記

- インタフェースタイプにより以下のように表記されます



**ge-0/0/0**

|       |           |                               |
|-------|-----------|-------------------------------|
| Type: | fe-x/x/x: | Fast Ethernet ports           |
|       | ge-x/x/x: | Gigabit Ethernet ports        |
|       | xe-x/x/x: | 10 Gigabit Ethernet ports     |
|       | et-x/x/x: | 40/100 Gigabit Ethernet ports |

Port number

PIC slot: Physical Interface Card →アップリンクモジュール

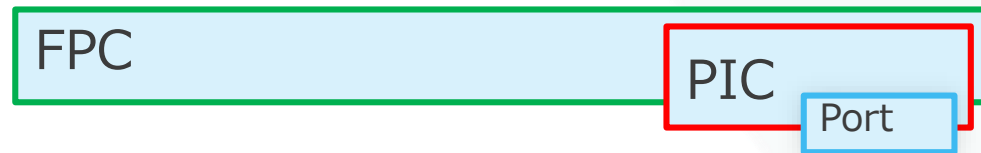
FPC slot: Flexible PIC Concentrator (line card) →筐体ナンバー

- その他のインタフェース
  - ae: LAGインタフェース
  - lo0: Loopbackインタフェース
  - me0: EX, QFXシリーズの管理インタフェース
  - fxp0: SRX, MXシリーズの管理インタフェース

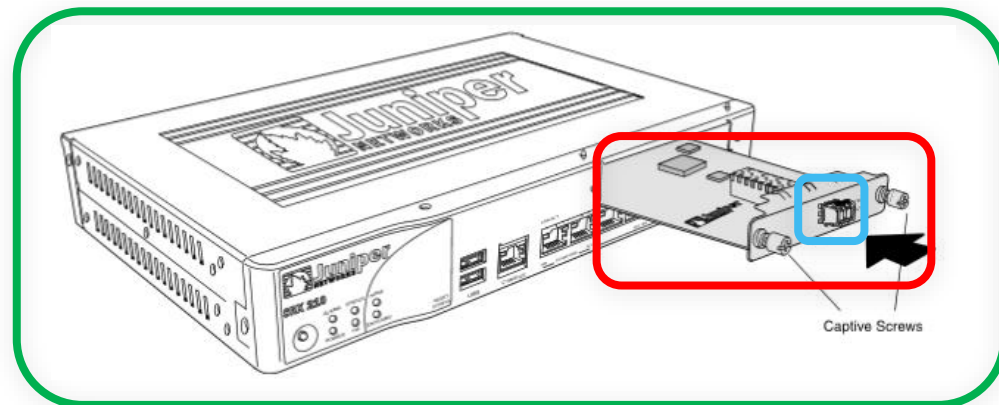
# PICと FPC

FPCはBOX型の筐体番号、Chassis型のラインカード番号に相当します。  
PICはFPCに接続されるアップリンクモジュールを指します。

xx-X/X/X

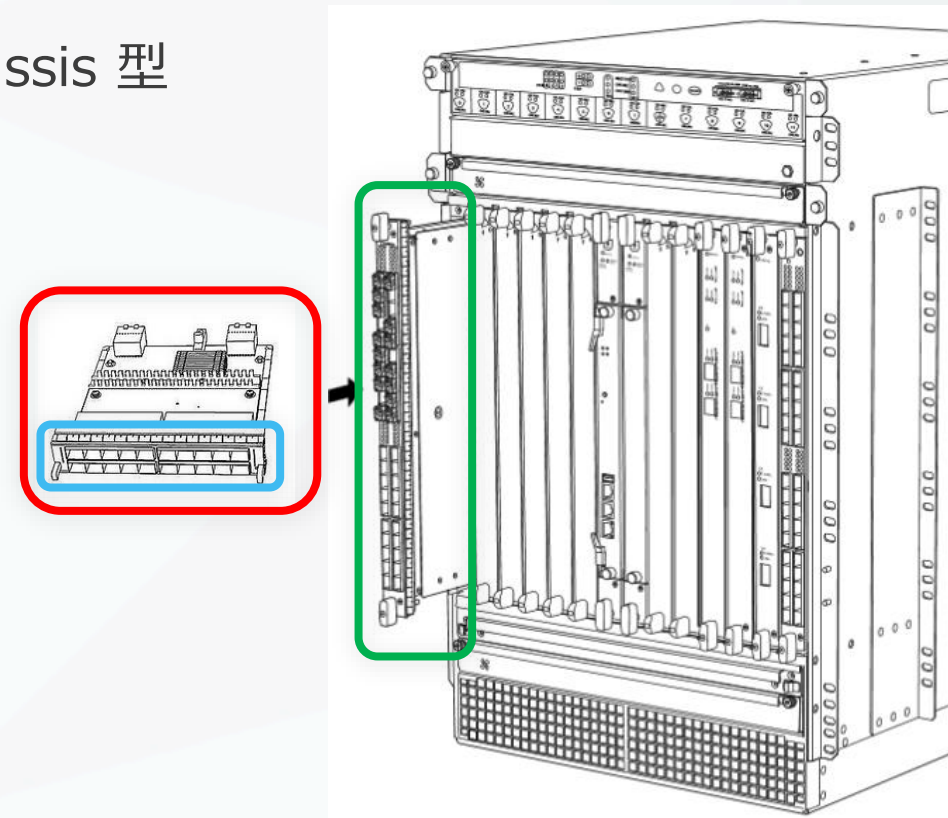


BOX型



※BOX型におけるOn-Board Portは、xx-0/0/Xと表現されます

Chassis 型



# インタフェース設定

- インタフェースの設定は物理プロパティの設定と論理プロパティの設定に分かれます
  - 物理プロパティの設定
    - データリンクプロトコル
    - リンクスピード、半/全2重通信
    - MTU
  - 論理プロパティの設定
    - プロトコルファミリー
      - inet (IPv4の設定)
      - inet6 (IPv6の設定)
      - mpls
      - ethernet-switching

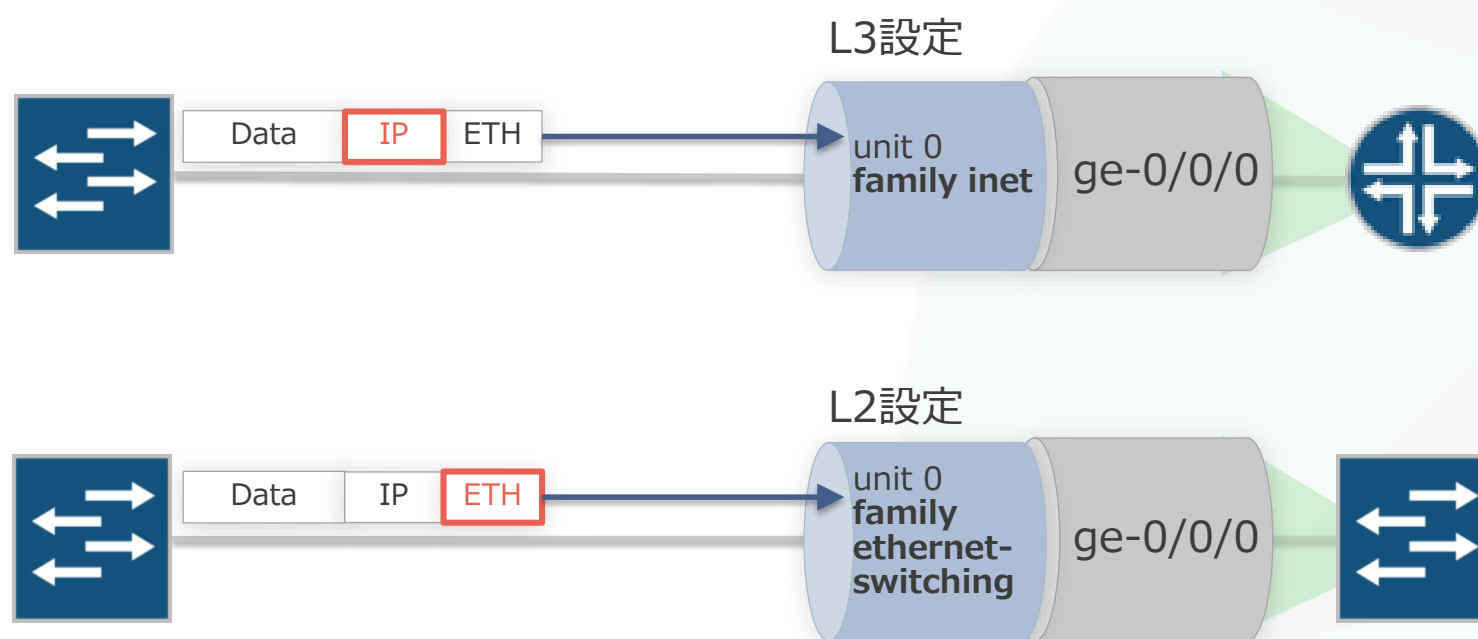
```
interfaces {  
  interface-name {  
    physical-properties;  
    [...]  
    unit unit-number {  
      logical-properties;  
      [...]  
    }  
  }  
}
```

インタフェース名配下に  
物理プロパティを設定

Unit#配下に  
論理プロパティを設定

# Unit ナンバーとは

- ロジカルプロパティを設定するには、“unit”とよばれる単位で設定
  - 一般的なネットワークOSのサブインタフェースに相当
  - unit 0はメインインタフェースに相当
  - インタフェースを動作させるためには最低1つのunitが必須
    - 1つの物理インタフェース上に複数のunitを作成することも可能
  - 物理インタフェースge-0/0/0 の unit 0 は、“ge-0/0/0.0”と表記
    - showコマンドや設定時にunitを指定しなかった場合、自動的にunit 0として補完

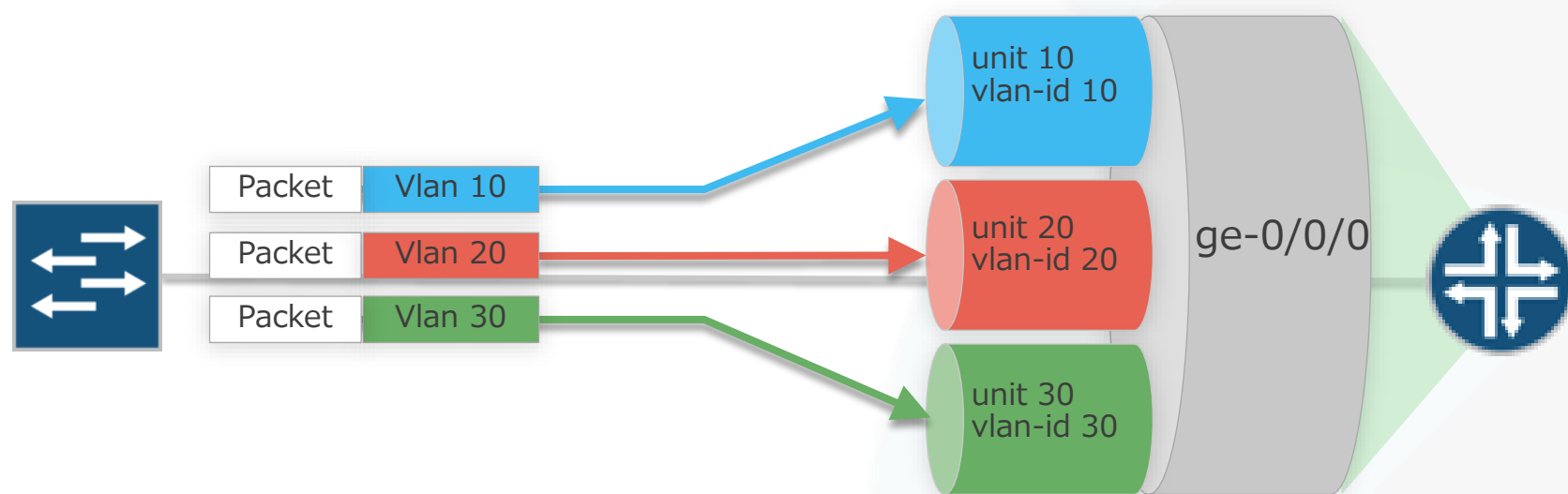


```
ge-0/0/0 {  
  unit 0 {  
    family inet {  
      address 192.168.1.1/24;  
    }  
  }  
}
```

```
ge-0/0/0 {  
  unit 0 {  
    family ethernet-switching {  
      interface-mode access;  
    }  
  }  
}
```

# 複数unitの設定例

- 1つの物理インタフェースに複数のunitを使用するケース
  - unitごとにvlan-idを設定して振り分け
  - IPアドレスやFirewall Filterもunitごとに個別に設定可能



```
ge-0/0/0 {  
  vlan-tagging;  
  unit 10 {  
    vlan-id 10;  
    family inet {  
      address 192.168.1.1/24;  
    }  
  }  
  unit 20 {  
    vlan-id 20;  
    family inet {  
      address 172.16.1.1/24;  
    }  
  }  
  unit 30 {  
    vlan-id 30;  
    family inet {  
      address 10.1.1.1/24;  
    }  
  }  
}
```

# 物理/論理インタフェース設定例

```
ge-0/0/0 {  
  description TEST;  
  speed 1g;  
  mtu 1400;  
  ether-options {  
    no-auto-negotiation;  
    link-mode full-duplex;  
  }  
  unit 0 {  
    description TEST2;  
    family inet {  
      address 10.10.10.1/24;  
    }  
  }  
  unit 100 {  
    description TEST3;  
    family inet6 {  
      address 1::1/64;  
    }  
  }  
}
```

物理プロパティ

論理プロパティ

# 管理者側から強制的にインタフェースを落とす方法

- **Disable**コマンドを使用してインタフェースを落とす

```
root# set interfaces ge-0/0/2 disable
[edit]
root# commit
commit complete
```

```
root# show interfaces
ge-0/0/2 {
  disable; ← admin (オペレータ) の強制的な
  unit 0 {
    family inet {
      address 140.0.0.12/24;
```

```
root# run show interfaces terse
Interface          Admin Link Proto Local
Remote
ge-0/0/0           up    up
ge-0/0/1           up    down
ge-0/0/2           down down
```

- **Disable**コマンドを消去してインタフェースをあげる

```
root# delete interfaces ge-0/0/2 disable
[edit]
root# commit
commit complete
```

```
root# run show interfaces terse
Interface          Admin Link Proto Local
Remote
ge-0/0/0           up    up
ge-0/0/1           up    down
ge-0/0/2           up   up
```





# JUNOS 経路設定

# Static Routeの設定

- Static route設定

```
# set routing-options static route <あて先アドレス>next-hop <ネクストホップアドレス>  
# set routing-options static route <あて先アドレス>オプション設定
```

## 設定例

```
[edit routing-options]  
root# show  
static {  
  route 0.0.0.0/0 next-hop 172.30.25.1;  
  route 172.28.102.0/24 {  
    next-hop 10.210.11.190;  
    no-readvertise;  
  }  
}
```

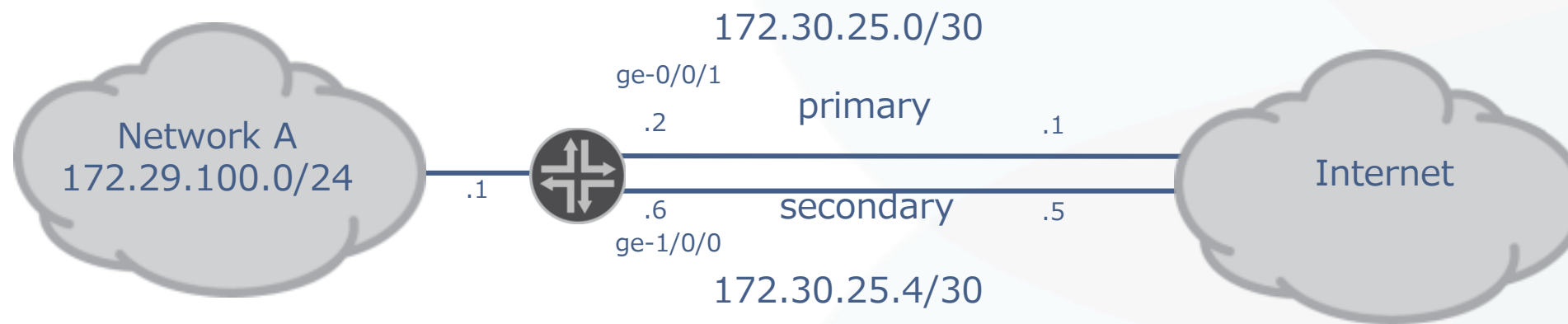
IPv4デフォルトルートの設定

経路を広報させないための設定  
マネージメント用の経路などに利用

# 制限付きネクストホップの設定

- 同じあて先にstatic routeを設定する場合はqualified-next-hopのオプションを利用し、preference(優先)の設定を施します

例: インターネット接続のためのデフォルトルートの設定



```
[edit routing-options]
root# show
static {
  route 0.0.0.0/0 {
    next-hop 172.30.25.1;
    qualified-next-hop 172.30.25.5 {
      preference 7;
    }
  }
}
```

Primary route

※Juniperのstatic routeのpreferenceは5

Secondary route

※preferenceを7に設定することで優先度を下げる

# Static Routeの確認

- showコマンドでstatic routeを確認する

```
root> show route protocol static

inet.0: 4 destinations, 4 routes (4 active, 0 holddown, 0 hidden)
+ = Active Route, - = Last Active, * = Both

0.0.0.0/0          *[Static/5] 00:41:59
                   > to 172.30.25.1 via ge-0/0/1.0
...

```

デフォルトルート

プロトコルとpreference

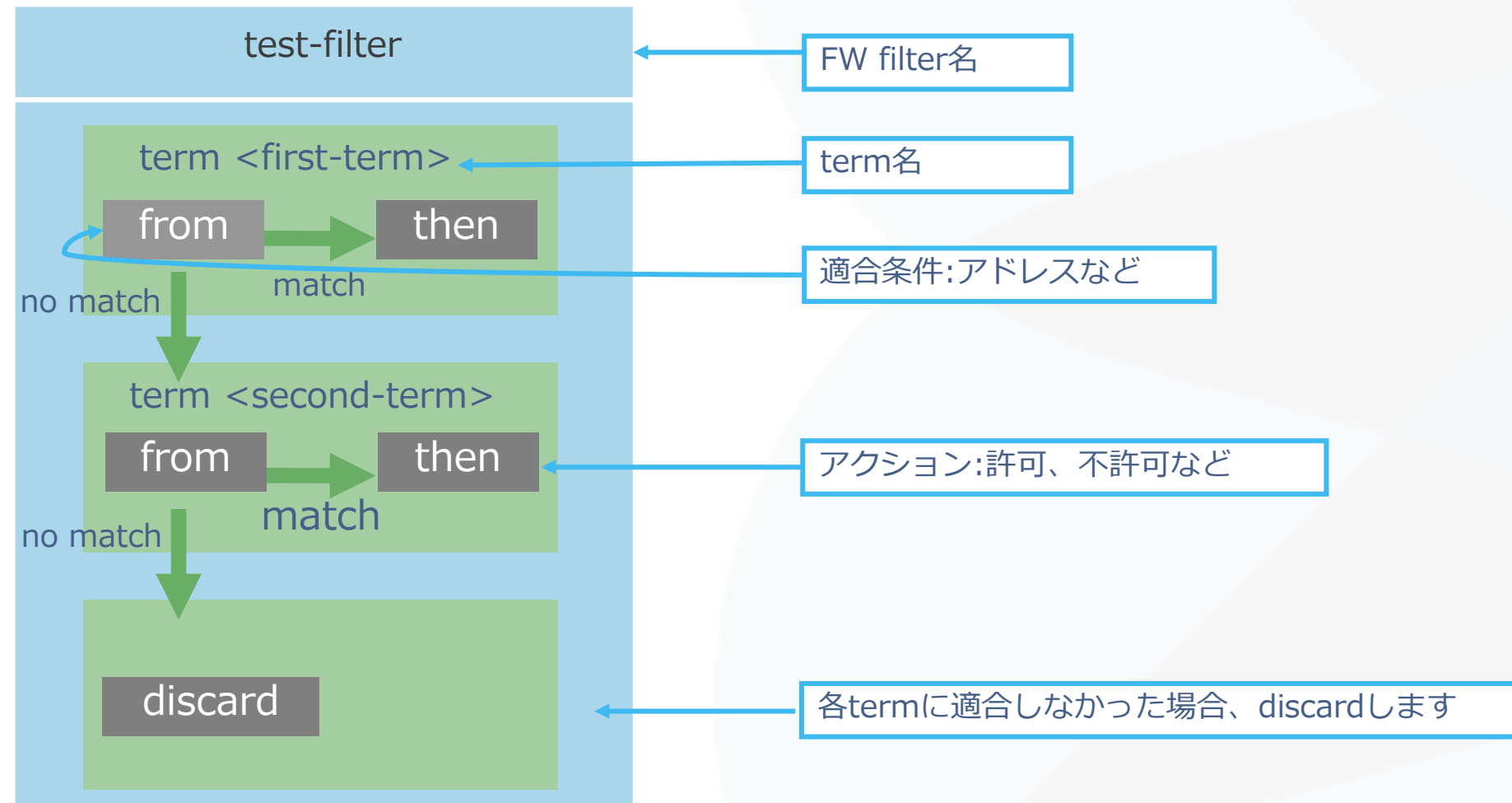
ネクストホップのアドレスとインタフェース



# Firewall Filter (ACL) の設定

# Firewall Filterの設定

- FWフィルタとは個々のパケットのフローを制御するためのステートレスなフィルタリングポリシーです (=ACL)
- FWフィルタではtermと呼ばれる条件付けのブロックを定義します
- フィルタ内のtermはtop→downの順番で精査されます



※新しくtermを作成した際など、評価の順番を変更する際はinsertコマンドを利用して意図した順番にTermを入れ替えて下さい

# Firewall Filterの設定

例 1 : 10.10.10.0/24からの通信を許可しないFWフィルタを作成

```
root# set firewall family inet filter FW-FILTER term BLOCK from source-address 10.10.10.0/24
root# set firewall family inet filter FW-FILTER term BLOCK then discard
root# set firewall family inet filter FW-FILTER term PERMIT then accept
```

```
root# show firewall family inet filter FW-FILTER
term BLOCK {
  from {
    source-address {
      10.10.10.0/24;
    }
  }
  then {
    discard;
  }
}
term PERMIT {
  then accept;
}
```

FW filter名

term名

適合条件:10.10.10.0/24からの通信

アクション:不許可

他のIPからの通信を許可

# Firewall Filterの設定

例 1 : 作成したFWフィルタをインタフェースへ適用

```
root# set interfaces ge-0/0/0 unit 0 family inet filter input FW-FILTER
```

```
root# show interfaces ge-0/0/0
unit 0 {
  family inet {
    filter {
      input FW-FILTER;
    }
  }
}
```

ge-0/0/0に入ってくる通信に対してFW-FILTERを適用

※FWフィルタの設定を有効にする際(commitする際)にcommit confirmを利用すると万が一設定を誤ってしまった場合にも切り戻しが可能になります



# Firewall Filterの設定

## 例2 : Termの順序入れ替え

```
root# set firewall family inet filter FW-FILTER term BLOCK from source-address 10.10.10.0/24
root# set firewall family inet filter FW-FILTER term BLOCK then discard
root# set firewall family inet filter FW-FILTER term PERMIT then accept
root# set firewall family inet filter FW-FILTER term BLOCK2 from protocol udp
root# set firewall family inet filter FW-FILTER term BLOCK2 then discard
```

All permitのあとにtermがあるのでこの順序だとこのtermはLookupされない

Termは設定した順番で設定ファイルに書き込みが行われます。

一方で、意図したフィルターを掛けるためには適切な順序でTermを記載する必要があります (上記例では、all PERMIT termの後にBLOCK2が書かれているので、Lookupがされないことに注意)

- **insert** コマンド : Firewall FilterやFirewall Policyのterm順序を変更する

```
root# insert firewall family inet filter FW-FILTER term BLOCK2 before term PERMIT
```

OR

```
root# insert firewall family inet filter FW-FILTER term PERMIT after term BLOCK2
```

# Firewall Filterの設定

## 例2： Termの順序入れ替え

意図した順番でtermが記載されていることを確認した上で、commitを行います

```
root# show firewall family inet
filter FW-FILTER {
  term BLOCK {
    from {
      source-address {
        10.10.10.0/24;
      }
    }
    then {
      discard;
    }
  }
  term BLOCK2 {
    from {
      protocol udp;
    }
    then {
      discard;
    }
  }
  term PERMIT {
    then accept;
  }
}
```

← insertコマンドによりterm BLOCK2がPERMITの前に移動している

# Firewall Filterの設定

## 例3 : JUNOS製品へのマネージメント通信を制限する

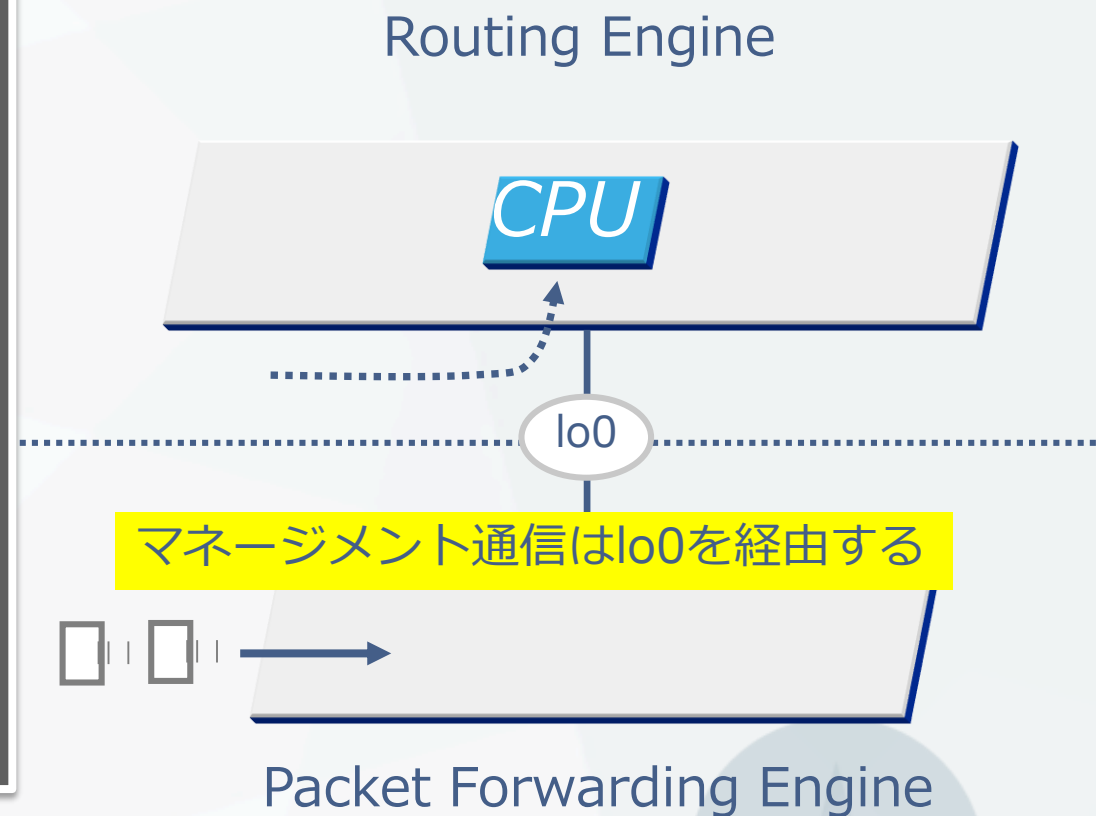
### 1. FWフィルタを作成する

- 192.168.1.0/24のセグメントからSSHでの通信のみ許可

### 2. 作成したFWフィルタをlo0 (ループバックインタフェース)に適用する

```
root# show firewall family inet
filter MANAGEMENT {
  term PERMIT {
    from {
      source-address {
        192.168.1.0/24;
      }
      protocol tcp;
      destination-port ssh;
    }
    then accept;
  }
}
```

```
root# show interfaces
lo0 {
  unit 0 {
    family inet {
      filter {
        input MANAGEMENT;
      }
      address
        10.10.10.1/24;
    }
  }
}
```



※EX, QFXシリーズ自身への通信を制御する場合、lo0および、me0へFirewall Filterを適用する必要があります。

※SRX, MXシリーズ自身への通信を制御する場合、lo0のみにFirewall Filterを適用することで制御可能となります。(管理インタフェースfxp0への適用は不要)

# Service Gateway "SRX" course

---

JUNOS Hands-on Training

Juniper Network, K.K.

# Training Outline Service Gateway "SRX" course

| トレーニング内容（後半）                          | 記載ページ                 |
|---------------------------------------|-----------------------|
| Juniper SRXシリーズ製品紹介                   | <a href="#">P.110</a> |
| LAB.1 JUNOSの基本的な操作・設定                 | <a href="#">P.120</a> |
| LAB.2 Firewallの設定                     | <a href="#">P.132</a> |
| LAB.3 NATの設定                          | <a href="#">P.152</a> |
| LAB.4 Chassis Clusterの設定              | <a href="#">P.172</a> |
| TIPs to be JUNOS Experts              | <a href="#">P.207</a> |
| まとめ                                   | <a href="#">P.232</a> |
| Appendix A: Chassis Cluster Deep Dive | <a href="#">P.236</a> |
| Appendix B: IPsec VPNの設定              | <a href="#">P.250</a> |
| Appendix C: NAT pool options          | <a href="#">P.266</a> |
| Appendix D: Security Logging          | <a href="#">P.270</a> |



# Juniper SRXシリーズ製品紹介

# Juniper Security 広範囲なSecurityサービスをご提供

## 次世代 ファイアウォールサービス

アプリケーションの  
可視化と制御

不正侵入防御 (IPS)

ユーザベース  
ファイアウォール

## Unified Threat Management

アンチウイルス

ウェブ/コンテンツ  
フィルタリング

アンチスパム

## 脅威インテリジェンス プラットフォーム

ボットネット/C&C

GEO-IP

独自のリスト, APT

## 高度な脅威防御 (ゼロデイ)

サンドボックス

Evasive Malware

豊富なレポートと  
分析機能

## SRX 基本サービス

ファイアウォール

マネージメント

アドレス変換 (NAT)

レポート

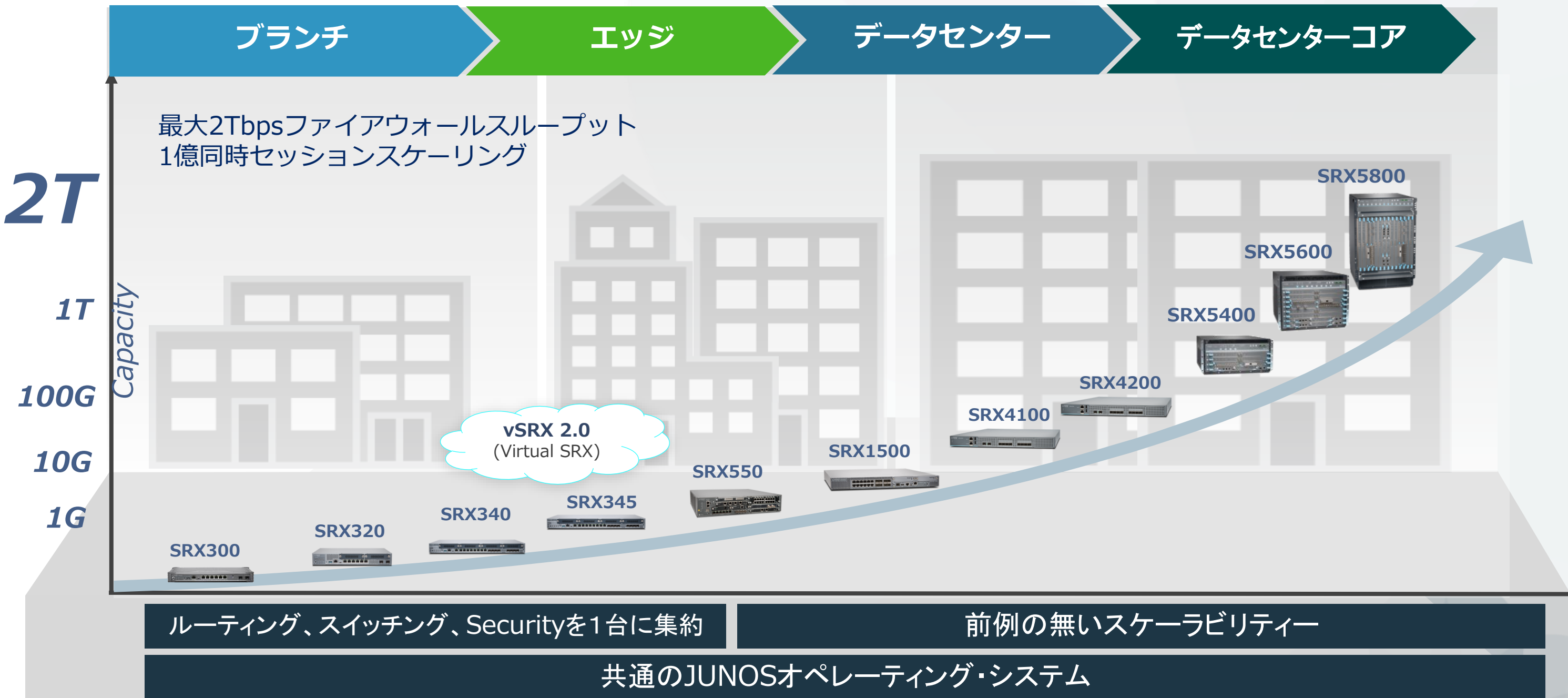
VPN

分析

ルーティング

自動化 (オートメーション)

# SRX 製品ラインナップ





# 現行セキュアルーターの抱える様々な課題を解決する Branch SRX 新しいラインナップ



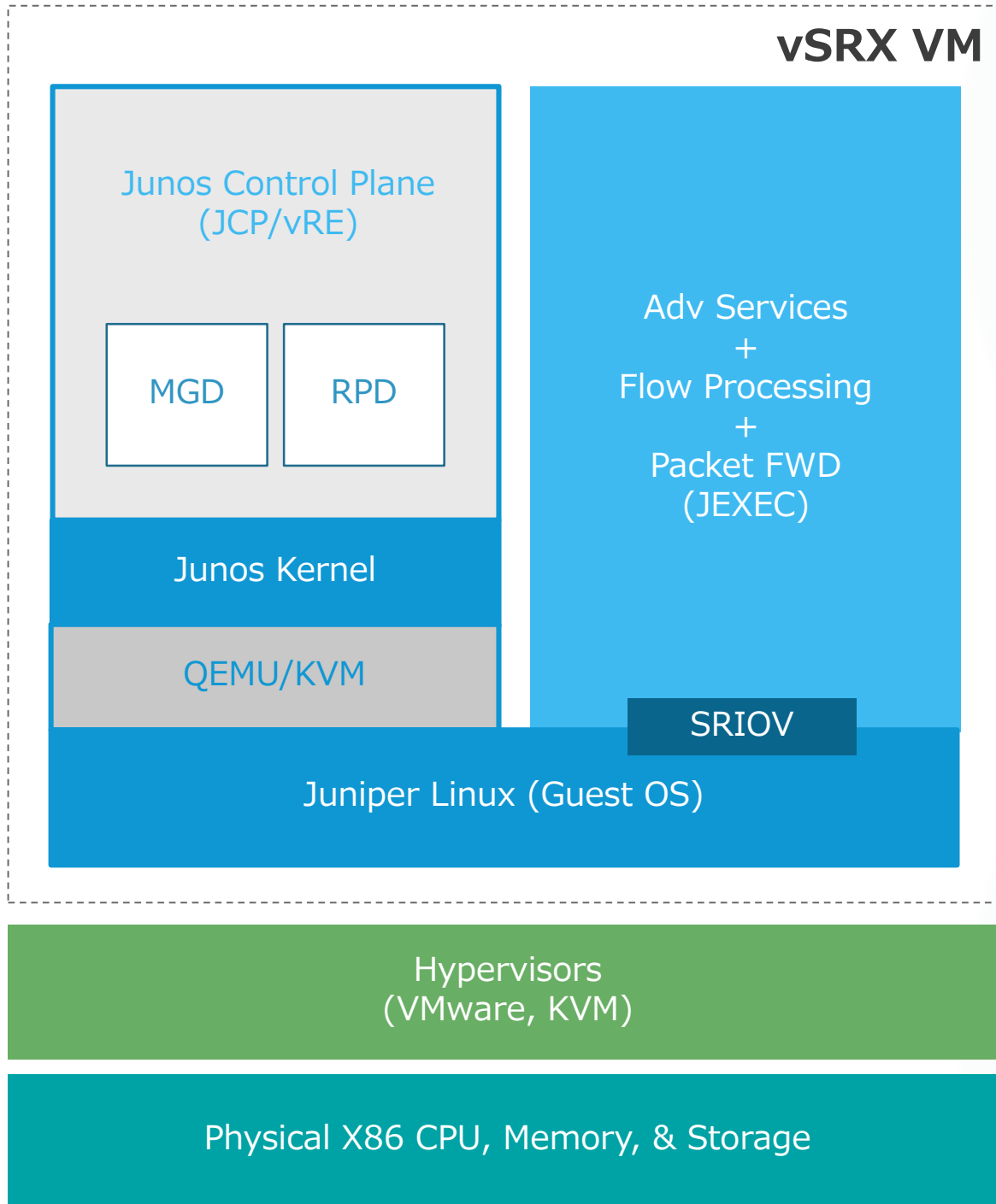
ルーティング、スイッチング、Securityを包括したオールインワン・デバイス



アプリケーション・Security、IPSec、MACsecなど様々なレイヤーに対応したSecurityを提供



エンドユーザ・アプリケーション・エクスペリエンスとオペレーションの効率化を実現



# vSRX(Virtual SRX)

- ✓ **HWアプライアンスSRXと同等の機能実装**  
 (Including Firewall, AppSecure, UTM/IDP, Integrated User Firewall, SSL Proxy, VPN, NAT, Routing, HA Cluster, etc.)
- ✓ **サポートプラットフォーム**
  - VMWare 5.1, 5.5, 6.0
  - CentOS 7.0 (KVM)
  - Ubuntu 14.04 (KVM)
  - Contrail 2.2
- ✓ **vSRXキー・ハイライト**
  - 物理SRXと同一の使用感で操作できる仮想ファイアウォール
  - 業界屈指のパフォーマンス
  - VMwareやKVMなどのハイパーバイザをサポート
  - vCPUを最大12個使用することにより、最大100Gbpsを超えるスループットを実現
  - 2vCPUで、最大約17Gbpsのファイアウォールスループットを実現
  - AWSなどのクラウドサービスにも対応

# ハイエンドSRXシリーズサービスゲートウェイ

重要度の高い資産(リソース)に更なる厳格なSecurityを



業界最高クラスのSECURITY



業界最高クラスのパフォーマンス性能



キャリア・クラウド事業者にて実績多数



# ハイエンドSRX Dynamic Service Architectureによるパフォーマンス拡張

## ■ 従来（他社）のファイアウォール製品

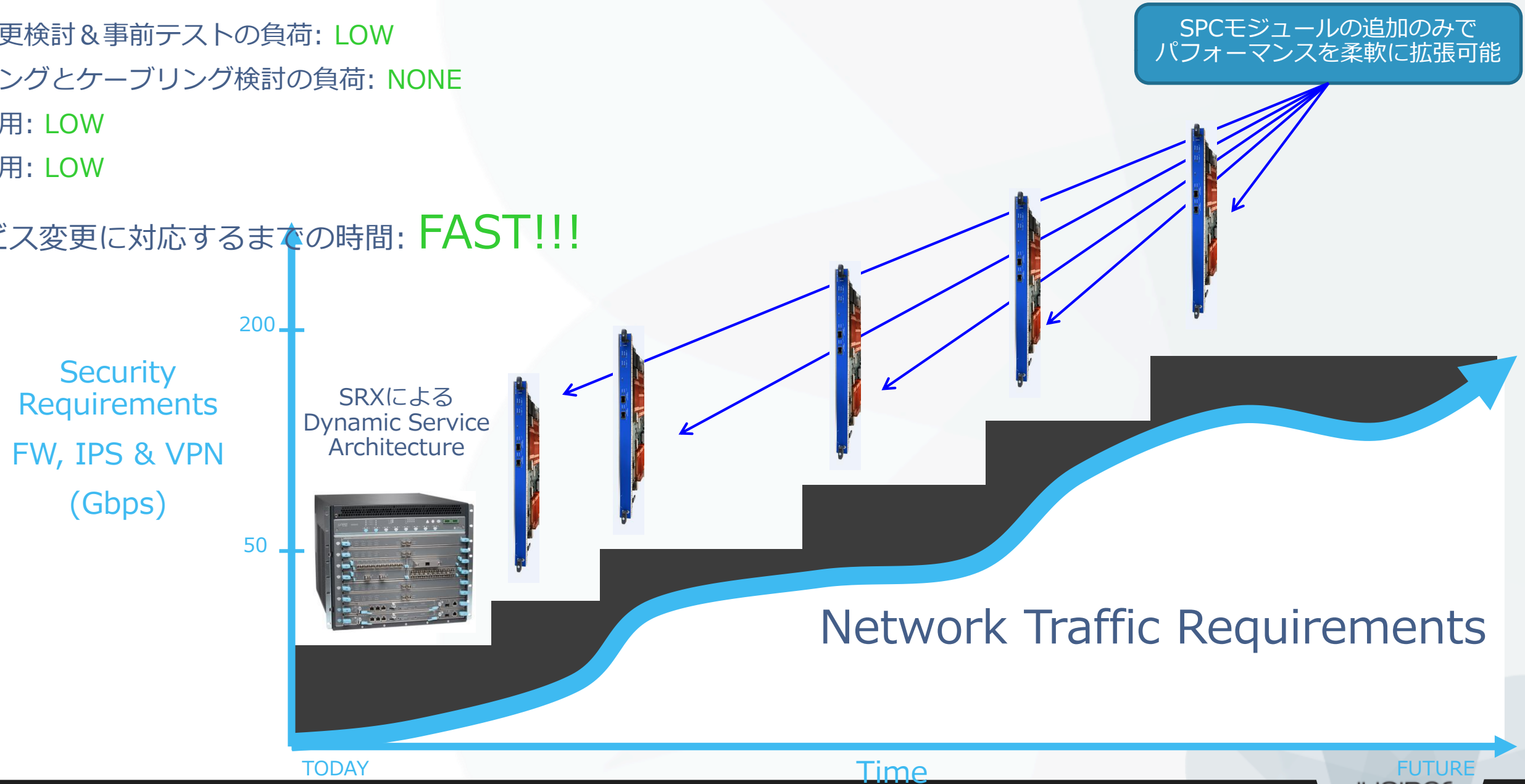
- 構成変更検討 & 事前テストの負荷: **HIGH**
- ラッキングとケーブリング検討の負荷: **HIGH**
- 投資費用: **HIGH**
- 運用費用: **HIGH**
- サービス変更に対応するまでの時間: **SLOW!!!**



# ハイエンドSRX Dynamic Service Architectureによるパフォーマンス拡張

## ■ SRX (Dynamic Service Architecture) の場合

- 構成変更検討 & 事前テストの負荷: **LOW**
- ラッキングとケーブリング検討の負荷: **NONE**
- 投資費用: **LOW**
- 運用費用: **LOW**
- サービス変更に対応するまでの時間: **FAST!!!**



# ブランチSRXシリーズ概要

小売店舗等向け  
50ユーザ未満

小規模支店向け  
最大50ユーザ

中規模支店向け  
最大100ユーザ

中・大規模支店向け  
最大200ユーザ

大規模支店向け  
最大500ユーザ

キャンパス向け  
最大1000ユーザ

## SRX300

- 8xGE (w/ 2xSFP)
- Desktop Form Factor
- ファンレスデザイン
- MAC-Sec (2xSFP)

### PERFORMANCE

- Firewall: 1 Gbps
- IMIX: 500 Mbps
- IPSec: 300 Mbps
- IPS: 200 Mbps

## SRX320

- 8xGE (w/ 2xSFP)
- 2x MPIM Slots
- MAC-Sec (2xSFP)
- POE付きモデル有り

### PERFORMANCE

- Firewall: 1 Gbps
- IMIX: 500 Mbps
- IPSec: 300 Mbps
- IPS: 200 Mbps

## SRX340

- 16xGE (w/ 8xSFP)
- 1RUサイズ
- 4x MPIM Slots
- MAC-Sec (16xGE)
- 管理専用ポート (1xGE)

### PERFORMANCE

- Firewall: 3 Gbps
- IMIX: 1 Gbps
- IPSec: 600 Mbps
- IPS: 400 Mbps

## SRX345

- 16xGE (w/ 8xSFP)
- 1RUサイズ
- 4x MPIM Slots
- MAC-Sec (16xGE)
- 管理専用ポート (1xGE)

### PERFORMANCE

- Firewall: 5 Gbps
- IMIX: 1.7 Gbps
- IPSec: 800 Mbps
- IPS: 600 Mbps

## SRX550-M

- 10xGE (w/ 4xSFP)
- 2U Rack Mount
- 2x MPIM + 6x GPIM
- 1 + 1 AC / DC PSU

### PERFORMANCE

- Firewall: 7 Gbps
- IMIX: 2 Gbps
- IPSec: 1 Gbps
- IPS: 800 Mbps

## SRX1500

- 12x1GE (Cu) + Gx1GE (SFP)
- 4x 10GE (SFP+)
- 2x PIM
- HAコントロール専用ポート

### PERFORMANCE

- Firewall: 9 Gbps
- IMIX: 5 Gbps
- IPSec: 4 Gbps
- IPS: 3 Gbps

# Mid range ハイエンドSRXシリーズ概要

小中規模  
データセンタ向け

SRX1500

- 12x1GE (Cu) + Gx1GE (SFP)
- 4x 10GE (SFP+)
- 2x PIM
- HAコントロール専用ポート

**PERFORMANCE**

- Firewall: 9 Gbps
- IMIX: 5 Gbps
- IPSec: 4 Gbps
- IPS: 3 Gbps

中規模  
データセンタ向け

SRX4100

- 8x 10GE (SFP+)
- 管理専用ポート (1xGE)
- HAコントロール専用ポート
- HAファブリック専用ポート

**PERFORMANCE**

- Firewall: 40 Gbps
- IMIX: 20 Gbps
- IPSec: 5 Gbps
- IPS: 10 Gbps

中規模  
データセンタ向け

SRX4200

- 8x 10GE (SFP+)
- 管理専用ポート (1xGE)
- HAコントロール専用ポート
- HAファブリック専用ポート

**PERFORMANCE**

- Firewall: 80 Gbps
- IMIX: 40 Gbps
- IPSec: 10 Gbps
- IPS: 20 Gbps

大規模  
データセンタ向け

SRX5400

- 100GE-CFP/CFP2
- 40GE-QSFPP
- 10GE-SFP, XFP
- 1GE - SFP

**PERFORMANCE**

- Firewall\*: 480 Gbps
- IMIX: 30 Gbps
- IPSec: 35 Gbps
- IPS: 22 Gbps

大規模  
データセンタ向け

SRX5600

- 100GE-CFP/CFP2
- 40GE-QSFPP
- 10GE-SFP, XFP
- 1GE - SFP

**PERFORMANCE**

- Firewall\*: 960 Gbps
- IMIX: 65 Gbps
- IPSec: 100 Gbps
- IPS: 50 Gbps

大規模  
データセンタ向け

SRX5800

- 100GE-CFP/CFP2
- 40GE-QSFPP
- 10GE-SFP, XFP
- 1GE - SFP

**PERFORMANCE**

- Firewall\*: 2 Tbps
- IMIX: 130 Gbps
- IPSec: 200 Gbps
- IPS: 100 Gbps

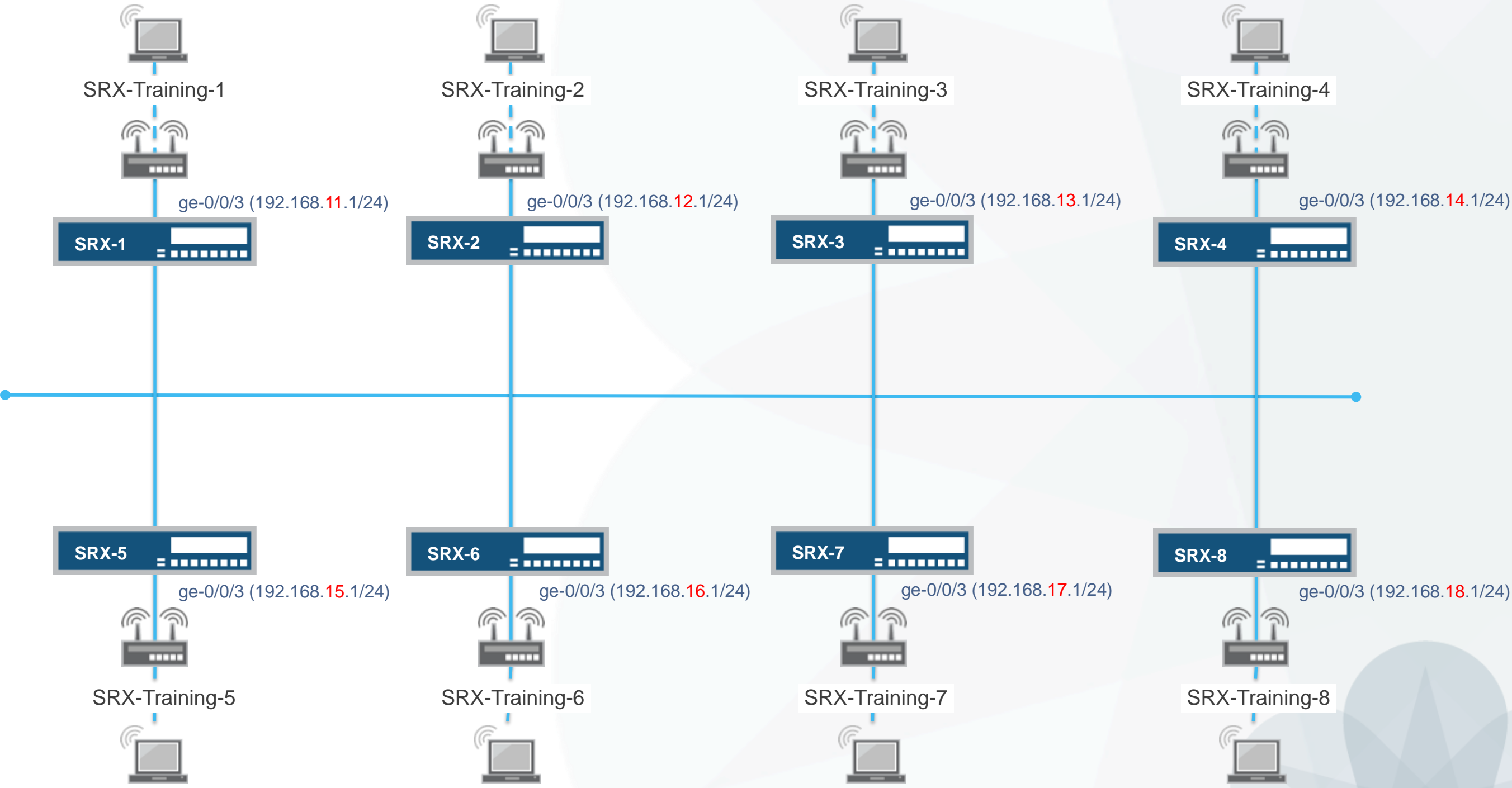
\* Express pass使用時



# LAB.1 JUNOSの基本的な操作・設定



# Security "SRX" course Topology (Lab.1 : 基本操作)



# SRXへログイン

初期設定状態のSRXにアカウント'root'でログインします。  
cliコマンドでJUNOSのOperationalモードを起動します。

- rootアカウントはserial console、またはssh接続時のみ使用可能です。
- 今回は事前にIPアドレス, rootパスワード, sshを設定済みです。
  - Root Password : [Juniper](#)

- Tera TermからSSHv2接続で接続してください。

```
--- JUNOS 15.1X49-D40.6 built 2016-03-22 05:18:15 UTC
root@% cli
root>
```



# Operationalモードのshowコマンド実行

構成やバージョンなど基本情報の確認を実施します。

- Active configurationを表示

```
root> show configuration
```

- ハードウェア情報を表示

```
root> show chassis hardware
```

- ソフトウェアバージョンの確認

```
root> show version
```

- インタフェースのステータス一覧の表示

```
root> show interface terse
```

- ルーティングテーブル表示

```
root> show route
```

- MACアドレステーブル表示

```
root> show ethernet-switching table
```

- サポートを受ける際に必要な機器情報 (RSI) を一括取得

```
root> request support information
```

※出力が一画面に入らない場合、 | no-more オプションを追加すると最後まで表示されます

# rootアカウントのパスワード設定 (設定済)

Configuration Modeに入り、設定変更の準備を行います。  
下記の手順でrootアカウントにパスワードを設定します。

– root password: **Juniper**

```
root> configure
root# set system root-authentication plain-text-password
      (改行後パスワード入力)
root# commit
```

※rootパスワード設定は必須です。設定が存在しないとcommitに失敗します。

# 新規アカウント作成

管理用アカウント"lab"を以下の設定で作成します。

| Username | Password | Class      |
|----------|----------|------------|
| lab      | lab123   | super-user |

commit完了後、一度rootユーザのセッションをログアウトします。

```
root# set system login user lab class super-user
root# set system login user lab authentication plain-text-password
(改行後パスワード入力)
root# commit and-quit
root> exit
root@% exit
```

sshで、作成したアカウントを使って正常にログインできることを確認します。

```
--- JUNOS 15.1X49-D40.6 built 2016-03-22 05:18:15 UTC
lab>
```

# サービスの起動とホスト名の設定

## サービスの起動

- デフォルトでは各種サービスが起動していないため、追加で設定します。  
(ssh のみ事前に設定済み)
- telnet ftp, httpで機器にアクセスできるようにします。

```
lab# set system services telnet
lab# set system services ftp
lab# set system services web-management http
```

## ホスト名の作成

- Topologyを参照して、各自がログインしている機器のホスト名を設定します。

```
lab# set system host-name SRX-x
```

## 変更したconfigの差分を確認

- Active configと比較して、設定が正しく追加されたことを確認しcommitします。

```
lab# show | compare
lab# commit
```

# サービス起動の確認

## Telnetによるアクセス

- Tera Termからtelnetでアクセスできることを確認します。
  - telnet 192.168.1X.1
  - 作成したユーザ(lab)を使用してログイン

## FTPによるアクセス

- Windowsからコマンドプロンプトを立ち上げ、FTPでアクセスできることを確認します。
  - ftp 192.168.1X.1
  - 作成したユーザ(lab)を使用してログイン
  - lsコマンドでユーザディレクトリを表示できることを確認
- 表示されない場合、Windows FirewallでFTP許可が必要

## ブラウザからWeb GUIへのアクセス

- ブラウザからアクセスし、J-Webの画面が表示されることを確認します。
  - http://192.168.1X.1/
  - root、または作成したユーザ(lab)を使用してログイン

# Configurationの確認

ここまでで設定したconfiguration全体を確認します。

## ① Operationalモードから確認

稼働中のActive configを表示します。

```
lab@SRX-1> show configuration  
lab@SRX-1> show configuration | display set
```

} 同じConfigを異なる形式で表示

## ② Configurationモードから確認

編集中のcandidate configを表示します。

commit後に設定変更をしていなければ、Active configと同じ内容が表示されます。

```
lab@SRX-1> configure  
Entering configuration mode  
  
[edit]  
lab@SRX-1# show  
lab@SRX-1# show | display set
```

} 同じConfigを異なる形式で表示



# Operationalモードのコマンドを表示

Configurationモードから、Operationalモードのコマンドを実行します。

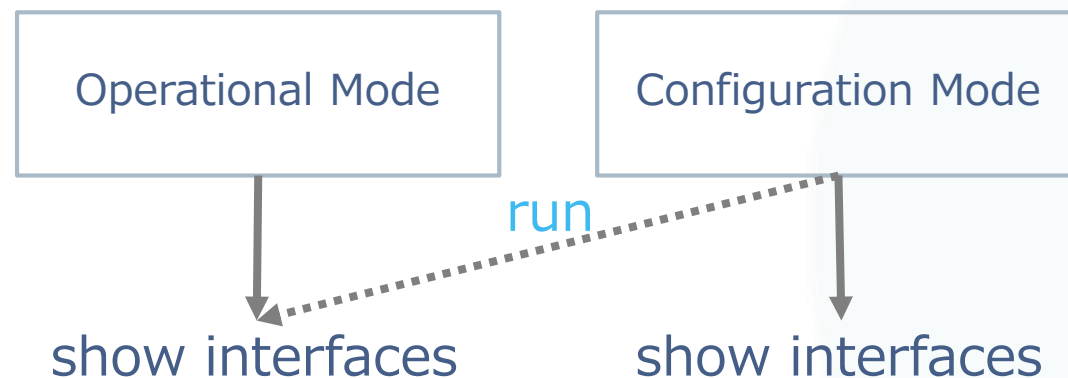
## ① Configurationモードに入ります

```
lab@SRX-1> configure
```

## ② show interfacesコマンドを実行

以下の2つのコマンドを実行し、表示される内容を確認します。

```
lab@SRX-1# show interfaces  
lab@SRX-1# run show interfaces
```



# commit confirmed

誤った設定をしてしまった場合でも設定が自動で元に戻ることを確認します。

①コマンドプロンプトからping 192.168.1X.1 -tを実行しておきます

②管理インタフェースの設定を削除

インターフェースとセキュリティの設定を削除します。 ※commitはまだしないこと

```
lab@SRX-1# delete interfaces ge-0/0/3
lab@SRX-1# delete security zones security-zone trust interfaces ge-0/0/3
lab@SRX-1# show | compare
```

③commit confirmed

commit confirmedオプションを使って、1分後に設定が戻るようにcommitします。  
commit完了メッセージが表示された後、アクセス不能になりTera Termが切断されます。

```
lab@SRX-1# commit confirmed 1
```

④pingが応答が返ってきたら再度ログインし、設定が戻っていることを確認  
削除したインターフェースの設定がもとに戻っていることを確認します。

```
lab@SRX-1> show configuration interfaces ge-0/0/3
```

# Configurationをファイルに保存

次のLabを始める前に、saveコマンドでconfiguration fileをsaveします。  
file listコマンドで正常にsaveできたことを確認します。

```
lab@SRX-1# save lab1-end_YYMMDD
Wrote 213 lines of configuration to 'lab1-end_YYMMDD'

[edit]
lab@SRX-1# exit
Exiting configuration mode

lab@SRX-1> file list

/var/home/lab/:
.ssh/
lab1-end_YYMMDD
```



# Firewallの設定

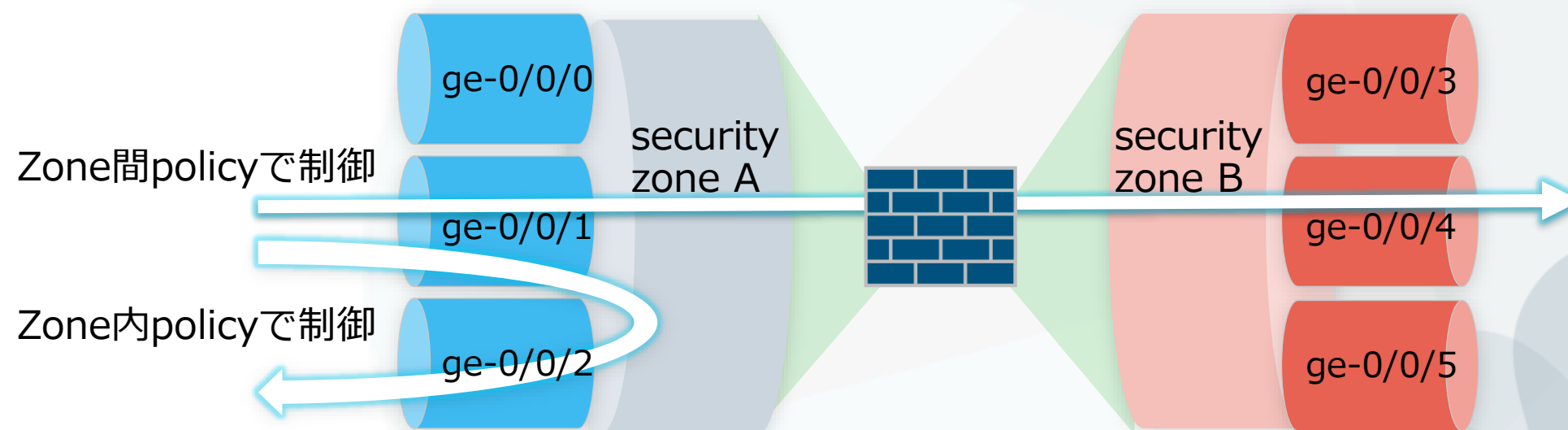
# Security zoneとPolicyによるトラフィック制御

- Security zone

- Security zoneとは、インターフェースに割り当てる仮想的なグループです。
- SRXではSecurity zoneを使ってトラフィックを制御します。

- Security policy

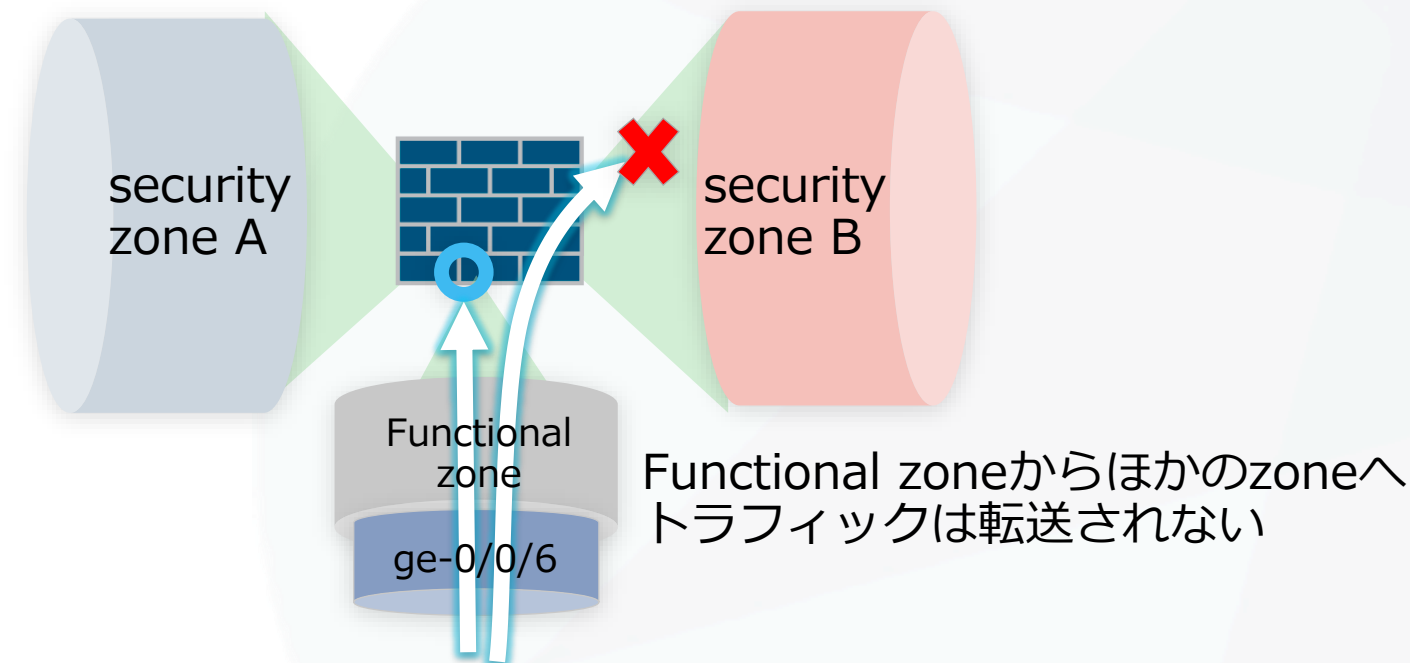
- Security policyとは、SRXを通過するトラフィックを制御するためのルールです。
- zone間トラフィックと、zone内トラフィックにそれぞれ適用されます。
  - Zone間：入力zoneと違うzoneへのpolicy
  - Zone内：入力zoneと同じzoneへのpolicy



# Security zoneとFunctional zone

Zoneには、大きく分けて下記の2タイプがあります

- **Security zone**
  - SRXを通過するトラフィックを制御するためのzoneです。
  - Security policyは、このSecurity zone間で設定されることとなります。
- **Functional zone**
  - SRXを管理するインターフェースを配置するためのzoneです。
  - このzoneで受信したトラフィックが、ほかのzoneに転送されることはありません。
    - fxp0(専用管理インタフェース)を使用する場合、functional zoneを作成する必要はありません



# SRXで終端するトラフィックの制御

## SRXで終端を許可するトラフィックを指定

- **host-inbound-traffic system-services**
  - 送受信を許可するサービスを指定します
    - ftp, http, telnet, ping, etc
    - 各サービスの起動はset system services 配下で設定
- **host-inbound-traffic protocols**
  - 送受信を許可するプロトコルを指定します
    - bgp, ospf, vrrp, etc
- zone単位、またはzone配下のinterface単位で設定
  - 例1: trust zone全体で、system-servicesとprotocolをすべて許可

```
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
```

- 例2: untrust zoneのinterface ge-0/0/0で、pingとospfのみを許可

```
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic system-services ping
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic protocols ospf
```

# Security zoneの設定例

- デフォルト設定 (15.1x49-D50.3)
  - Trust zone
    - Zone単位で host-inbound-traffic system-services, protocolsをすべて許可
    - インターフェース irb.0がバインド

```
set security zones security-zone trust host-inbound-traffic system-services all
set security zones security-zone trust host-inbound-traffic protocols all
set security zones security-zone trust interfaces irb.0
```

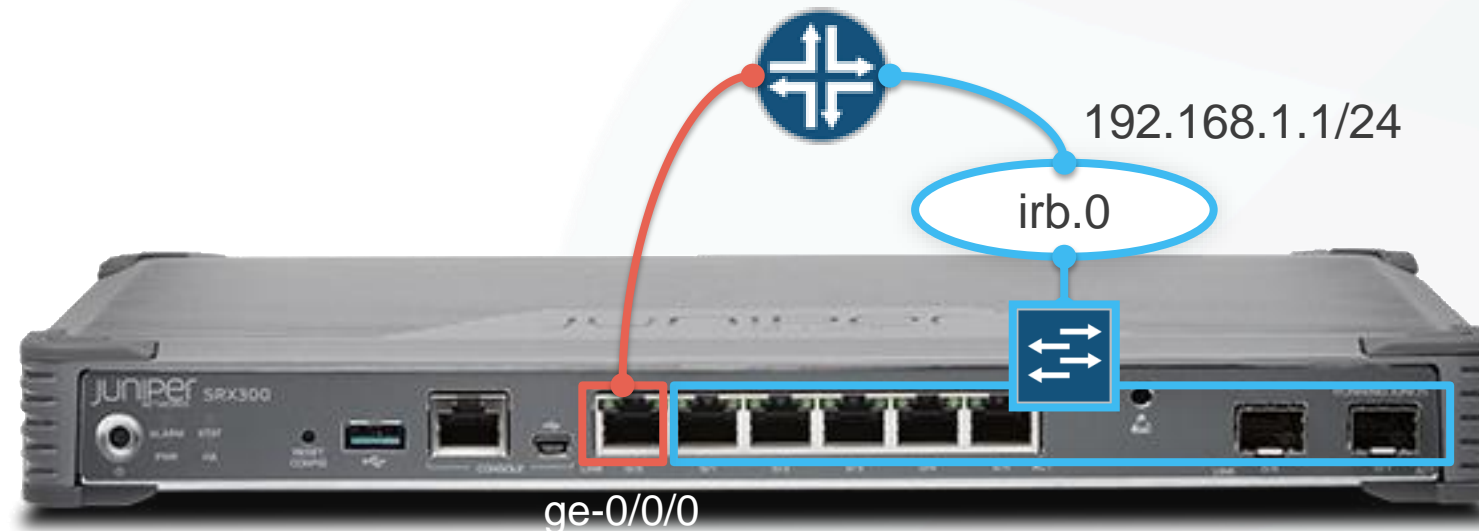
- Untrust zone
  - Zone下のinterface ge-0/0/0でdhcp, tftpのみ許可

```
set security zones security-zone untrust screen untrust-screen
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic system-services dhcp
set security zones security-zone untrust interfaces ge-0/0/0.0 host-inbound-traffic system-services tftp
```



# 補足：IRBインターフェースについて

- IRB(Integrated Routing and Bridging)とは
  - VLANルーティングで使用するインターフェースの名称
  - 通常のインターフェースと同様、IPアドレスをアサインして使用
- SRX300 / 15.1X53-D50の場合
  - ge-0/0/0はL3 ルーティング動作
  - ge-0/0/1~7はL2 スイッチング動作



# Security zoneの設定確認

- Zoneの設定確認コマンド

```
root@> show security zones
```

```
Security zone: trust
```

```
Send reset for non-SYN session TCP packets: Off
```

```
Policy configurable: Yes
```

```
Interfaces bound: 1
```

← バインドされたインターフェース数

```
Interfaces:
```

```
  irb.0
```

← バインドされたインターフェース名

```
Security zone: untrust
```

```
Send reset for non-SYN session TCP packets: Off
```

```
Policy configurable: Yes
```

```
Screen: SCREEN
```

← 適用されているscreen名

```
Interfaces bound: 1
```

```
Interfaces:
```

```
  ge-0/0/0.0
```

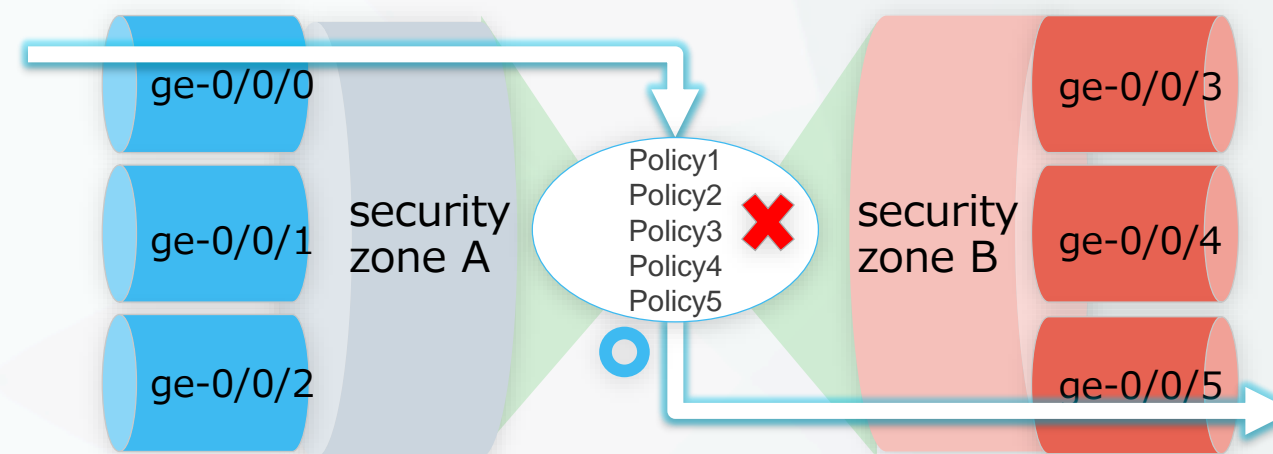
# Security policyの設定 (from-zone/to-zone)

- policyを作成するため、送信元zone(from-zone)と宛先zone(to-zone)を定義

- Zone間通信のpolicy

例1: Zone Aからzone Bへの通信policyを作成

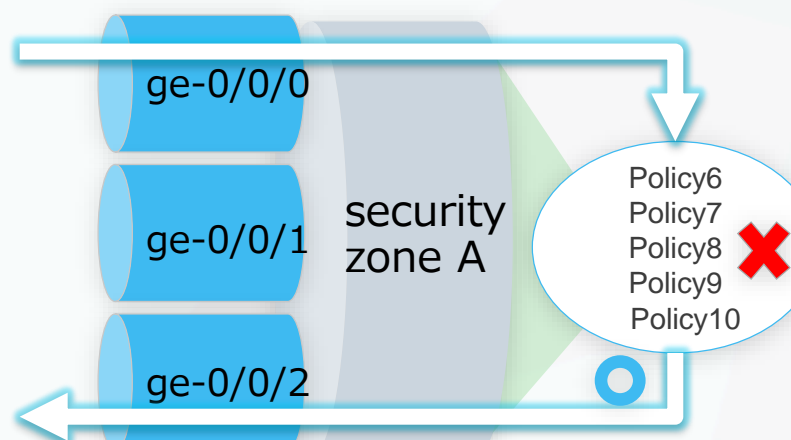
```
root# show security policies
from-zone A to-zone B {
  policy 1 {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
```



- Zone内通信のpolicy

例2: Zone Aからzone Aへの通信policyを作成

```
root# show security policies
from-zone A to-zone A {
  policy 6 {
    match {
      source-address any;
      destination-address any;
      application any;
    }
    then {
      permit;
    }
  }
}
```



※複数のpolicyがある場合、  
設定の上から順に評価されます

# Security policyの設定 (match/then)

- 各policyではmatchとthenでトラフィックを評価してアクションを決定
  - match - policyに合致させる条件を設定する
  - then - 条件に合致した通信に対するアクションを設定する

## ▼ matchで指定する条件

- source-address : 送信元
- destination-address : 宛先
- application : アプリケーション

※各policyではすべて設定必須

## ▼ thenで指定するアクション

- permit : 許可
- deny : 破棄 (無応答、エラーコードを返さない)
- reject : 拒否 (エラーコードを返す)
- log : ログの取得
- count : 該当Policyの packets 数、バイト数を取得

例 : zone Aからzone Bに対してすべての通信を許可

```
set security policies from-zone A to-zone B policy policy1 match source-address any
set security policies from-zone A to-zone B policy policy1 match destination-address any
set security policies from-zone A to-zone B policy policy1 match application any
set security policies from-zone A to-zone B policy policy1 then permit
```

# Security policyの設定 (address-book)

- address-book

- match条件に特定のアドレスなどを指定したい場合
  - source-address/destination-address で使用したいアドレスを設定できます
  - すべてのトラフィックを指定したい場合は、予め定義されている any を使用します
- address-bookの設定

```
set security address-book global address AAA 1.1.1.1/32
set security address-book global address BBB 172.16.0.0/16
set security address-book global address CCC 192.168.1.0/24
```

- address-setの設定

- address-bookを複数組み合わせ使用可能
- 例：BBBとCCCを組み合わせ、BCSETを作成

```
set security address-book global address-set BCSET address BBB
set security address-book global address-set BCSET address CCC
```

- Address-bookの適用

```
set security policies from-zone A to-zone B policy policy1 match source-address AAA
set security policies from-zone A to-zone B policy policy1 match destination-address BCSET
```

# Security policyの設定 (default-policy)

- policyの評価順序
  - policyは設定の上から順番に評価されます
  - matchしたpolicyのアクションが一度だけ実行され、以後のpolicyは評価されません
- 明示的にpolicyを指定しない場合
  - default-policyに指定されているアクションが有効になります
    - default-policyは、どのpolicyにもmatchしなかった場合に、最後に評価されるpolicy
    - デフォルトアクションはすべてのパケットをdropするdeny-all (暗黙のdeny)
  - 以下の設定によりpermit-allに変更することも可能

```
set security policies default-policy permit-all
```

# SCREEN機能

- Screen機能

- L3/L4の基本的な攻撃防御機能を提供します
- IDPモジュールを使用しない独立した機能となり、非常に高速に動作します
- 攻撃防御の組(ids-option)を作成し、それをzoneに適用します

## 1: Screenの設定をuntrust-screenとして作成

```
set security screen ids-option untrust-screen icmp ping-death
set security screen ids-option untrust-screen ip source-route-option
set security screen ids-option untrust-screen ip tear-drop
set security screen ids-option untrust-screen tcp syn-flood alarm-threshold 1024
set security screen ids-option untrust-screen tcp syn-flood attack-threshold 200
set security screen ids-option untrust-screen tcp syn-flood source-threshold 1024
set security screen ids-option untrust-screen tcp syn-flood destination-threshold 2048
set security screen ids-option untrust-screen tcp syn-flood timeout 20
set security screen ids-option untrust-screen tcp land
```

## 2: 作成したuntrust-screenをuntrust zoneに適用

```
set security zones security-zone untrust screen untrust-screen
```

# SCREEN機能の確認

- show security screen statistics

```
lab> show security screen statistics zone untrust
Screen statistics:
```

| IDS attack type        | Statistics |
|------------------------|------------|
| ICMP flood             | 5          |
| UDP flood              | 21         |
| TCP winnuke            | 0          |
| TCP port scan          | 0          |
| ICMP address sweep     | 0          |
| TCP sweep              | 12         |
| UDP sweep              | 15         |
| IP tear drop           | 0          |
| TCP SYN flood          | 0          |
| IP spoofing            | 0          |
| ICMP ping of death     | 0          |
| IP source route option | 0          |
| TCP land attack        | 0          |
| TCP SYN fragment       | 0          |
| TCP no flag            | 0          |
| IP unknown protocol    | 154        |
| IP bad options         | 0          |
| IP record route option | 0          |
| IP timestamp option    | 0          |

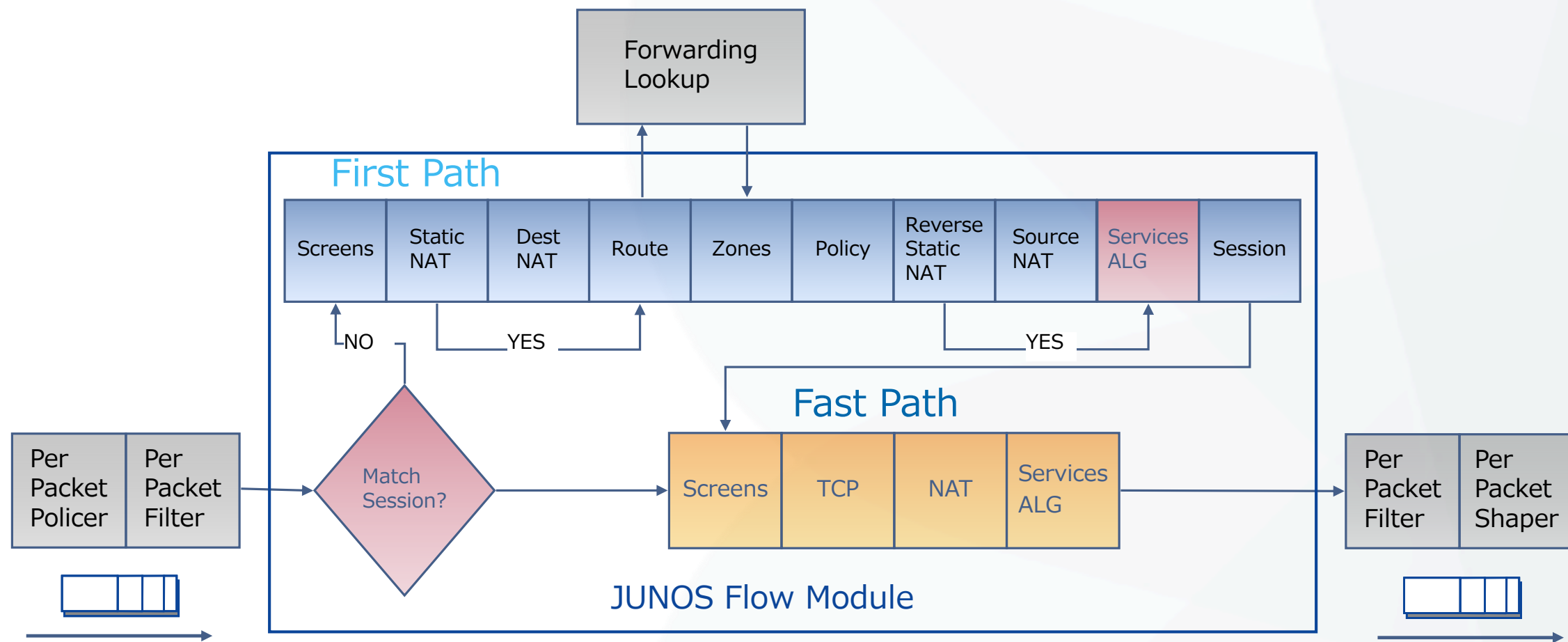
```
~~~~~
```



該当Screenにヒットしたカウント値



# SRXパケット処理の流れ (参考)



- 1) キューからパケットをピックアップ
- 2) パケットPolicy
- 3) パケットをフィルター
- 4) セッションのルックアップ

- 5a) 新規セッションの場合
  - FW Screenをチェック
  - Static、Destination NAT
  - ルートのルックアップ
  - 宛先のゾーンPolicyをチェック
  - Policyのルックアップ
  - リバースStatic、Source NAT
  - ALGをセットアップ
  - セッションのインストール

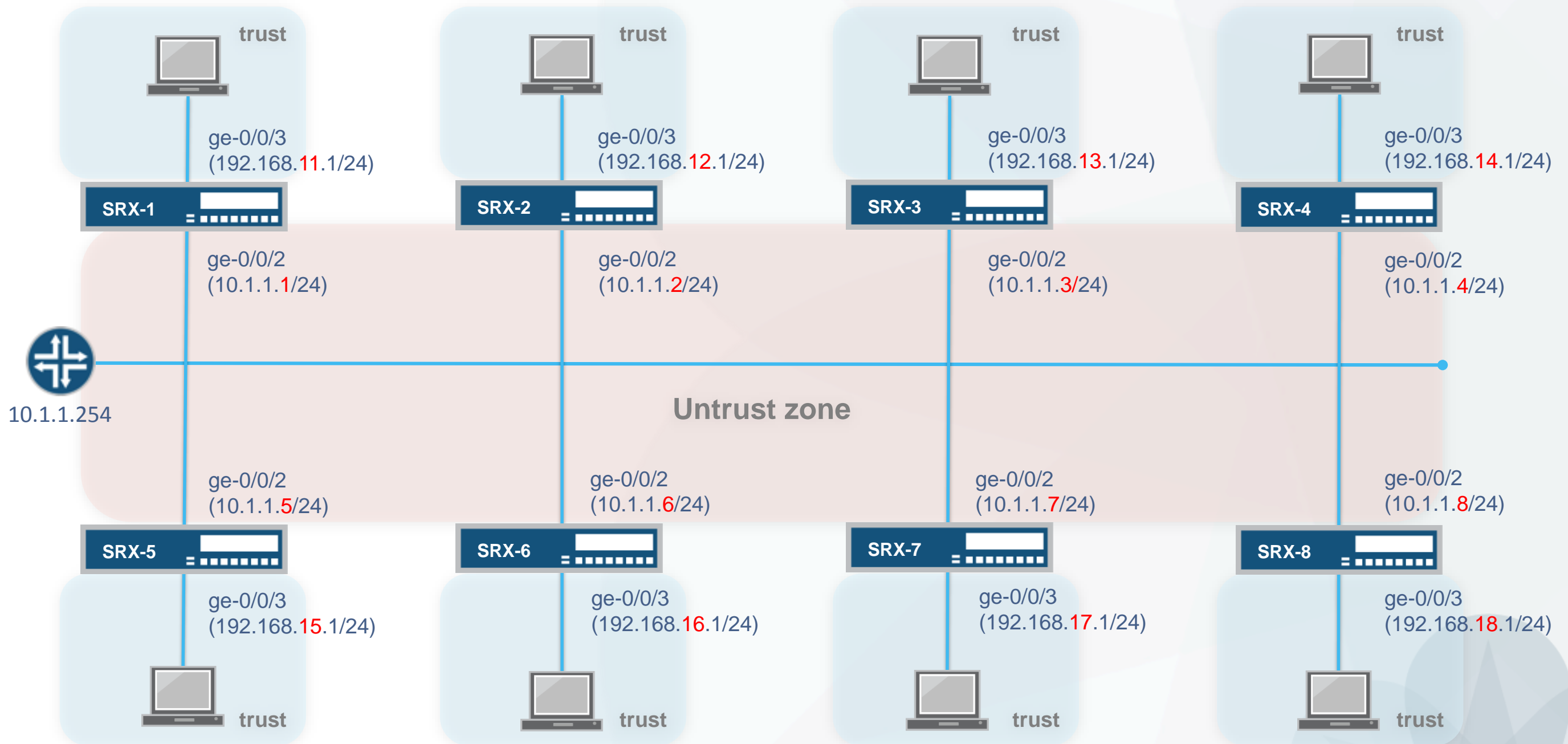
- 5b) セッションが確立している場合
  - FW Screenをチェック
  - TCPをチェック
  - NATトランスレーション
  - ALGプロセッシング

- 6) パケットをフィルター
- 7) パケットをシェーピング
- 8) パケットを転送



# LAB.2 Firewallの設定

# Security "SRX" course Topology (Lab.2 : Firewallの設定)



# Interface, Zoneの設定

- ge-0/0/2(untrust側)にIPアドレスを設定し、デフォルトルートを追加します

```
set interfaces ge-0/0/2 unit 0 family inet address 10.1.1.X/24
set routing-options static route 0/0 next-hop 10.1.1.254
```

- untrust zoneを作成し、ge-0/0/2をバインドします
  - host-inbound-trafficではping, telnet, ssh, http, httpsのサービスを許可

```
set security zones security-zone untrust interfaces ge-0/0/2
set security zones security-zone untrust host-inbound-traffic system-services ping
set security zones security-zone untrust host-inbound-traffic system-services ssh
set security zones security-zone untrust host-inbound-traffic system-services http
set security zones security-zone untrust host-inbound-traffic system-services https
set security zones security-zone untrust host-inbound-traffic system-services telnet
```

# Address bookの設定

- Policyでsource/destination-addressに使用するaddress-bookを作成します
  - Trust zone用

```
set security address-book global address trust-segment 192.168.1X.0/24
```

- Untrust zone用
  - address-setを使用します

```
set security address-book global address untrust-srx 10.1.1.0/24
set security address-book global address untrust-web 192.168.1.0/24
set security address-book global address-set untrust-segment address untrust-srx
set security address-book global address-set untrust-segment address untrust-web
```

# Security policyの設定

## ① trust zoneからuntrust zoneへのポリシーを設定

- ・ trust-segmentからuntrust-segmentの通信はすべて許可



```
set security policies from-zone trust to-zone untrust policy TEST match source-address trust-segment
set security policies from-zone trust to-zone untrust policy TEST match destination-address untrust-segment
set security policies from-zone trust to-zone untrust policy TEST match application any
set security policies from-zone trust to-zone untrust policy TEST then permit
```

## ② untrust zoneからtrust zoneへのポリシーを設定

- ・ すべて不許可



```
set security policies from-zone untrust to-zone trust policy TEST2 match source-address any
set security policies from-zone untrust to-zone trust policy TEST2 match destination-address any
set security policies from-zone untrust to-zone trust policy TEST2 match application any
set security policies from-zone untrust to-zone trust policy TEST2 then deny
```

# Security policyの確認

- trustからuntrustへのポリシー確認

- ① コマンドプロンプトから10.1.1.254に対してpingを実行
  - 応答があれば正しくポリシーが動作していることが確認できます

- ② Tera Termの新規セッションで、隣のSRXのIPアドレス(untrust)にtelnetを実行

- 宛先は右側の表を参照
- ログインプロンプトが開いたら、lab / lab123でログインしてください
- 自分のSRX上で、以下のコマンドを確認します
  - show security flow session
  - show security policies detail
  - show log messages

- untrustからtrustへのポリシー確認

- ③ コマンドプロンプトから隣のSRX(trust)にpingを実行
  - 宛先は右側の表を参照
  - pingに応答がないことを確認します

| 座席番号 | ②の宛先<br>(untrust) | ③の宛先<br>(trust) |
|------|-------------------|-----------------|
| 1    | 10.1.1.2          | 192.168.12.1    |
| 2    | 10.1.1.1          | 192.168.11.1    |
| 3    | 10.1.1.4          | 192.168.14.1    |
| 4    | 10.1.1.3          | 192.168.13.1    |
| 5    | 10.1.1.6          | 192.168.16.1    |
| 6    | 10.1.1.5          | 192.168.15.1    |
| 7    | 10.1.1.8          | 192.168.18.1    |
| 8    | 10.1.1.7          | 192.168.17.1    |

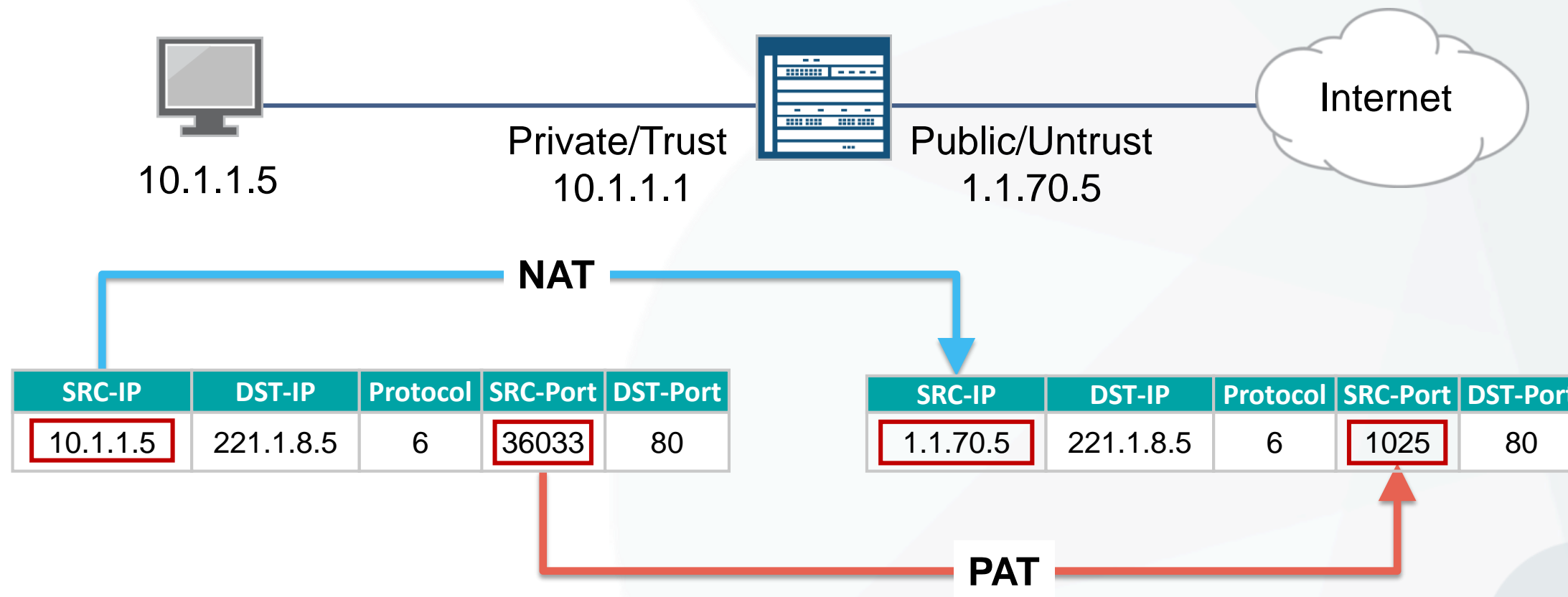


# NATの設定



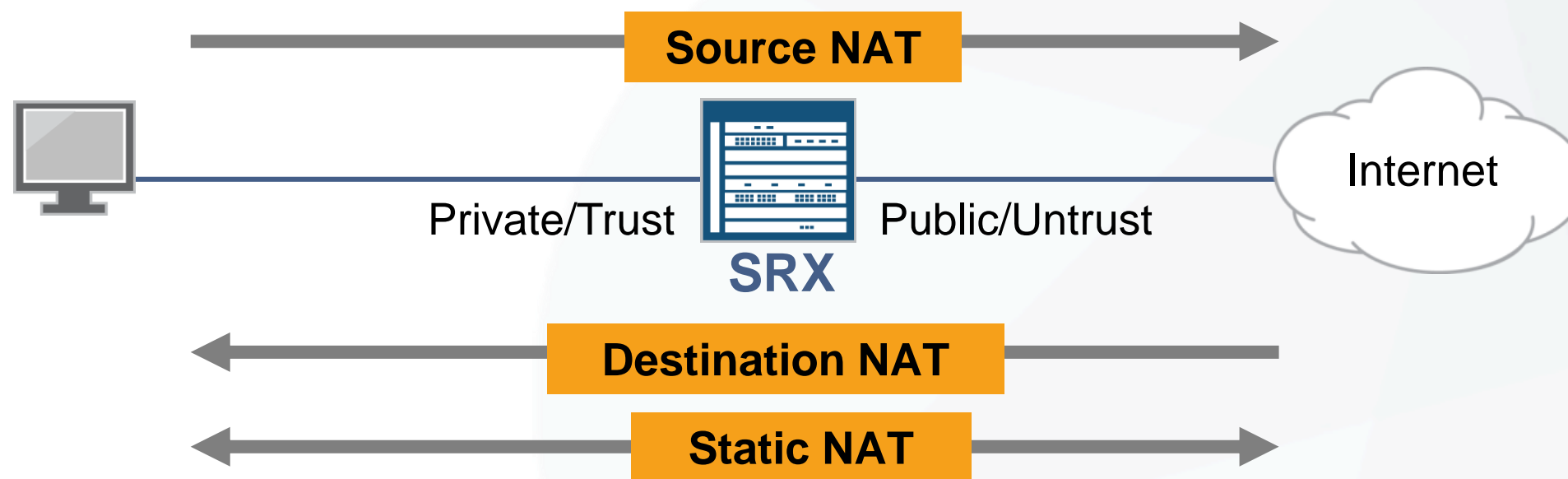
# NAT 概要

- publicとprivateのIPアドレスを変換
  - SRXではセキュリティポリシーとは別に管理・設定
  - ポート変換(Port Address Translation: PAT)もサポート



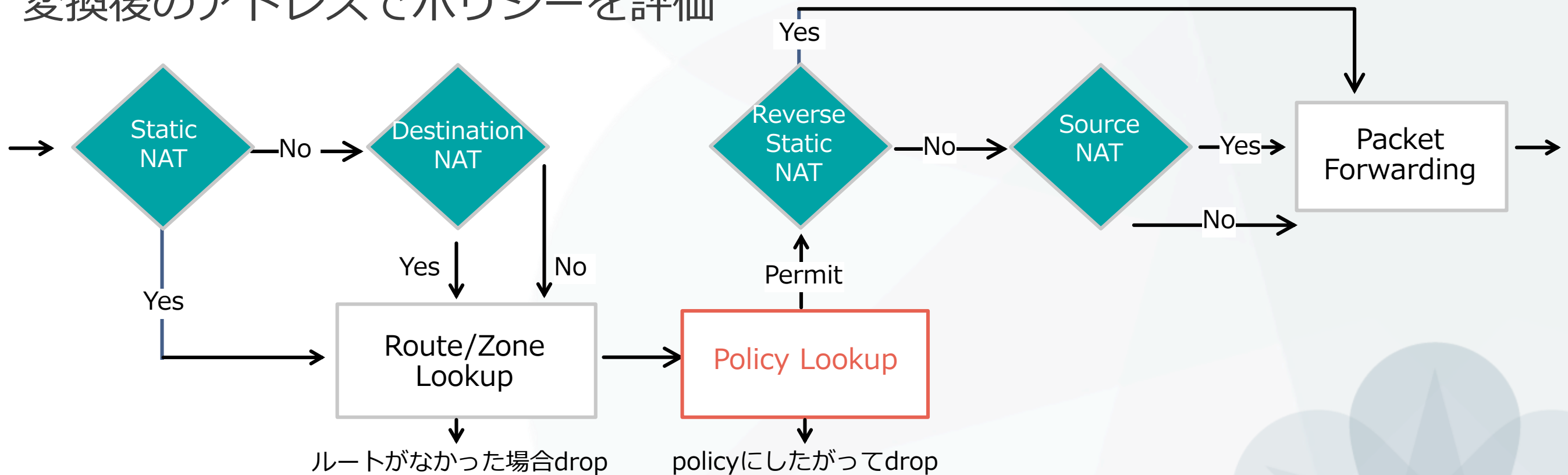
# SRXのNATタイプ

- 大きく分けて以下の3タイプのNAT/PAT方法
  - Source NAT 送信元IPアドレスを変換
  - Destination NAT 宛先IPアドレスを変換
  - Static NAT 1つのPrivate IPに1つのPublic IPをマッピングして変換
- Source/Destinationの組み合わせも可能



# NAT処理の順序

- Source NAT
  - セキュリティポリシー適用後に処理
  - 変換前のアドレスでポリシー評価
- Static & Destination NAT
  - セキュリティポリシー適用前に処理
  - 変換後のアドレスでポリシーを評価



# NATルール適用条件

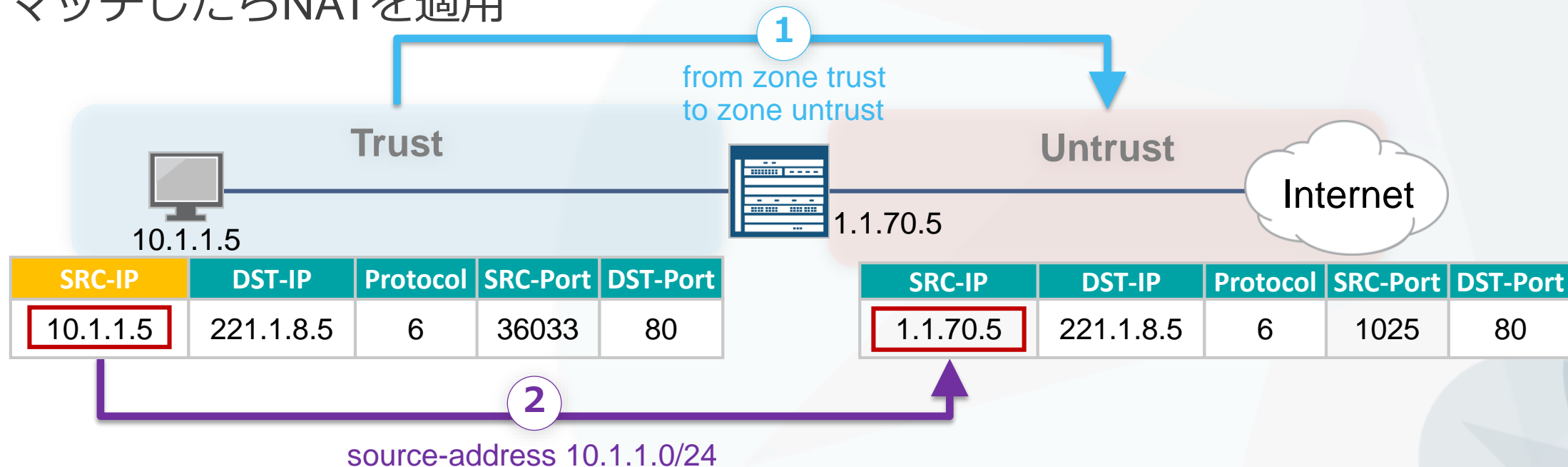
- NATを適用するかどうか決める「2段階」

- ① 通信の方向 (rule-set)

- from - toで、「どこから」「どこへ」の通信かを指定
  - from: zone, interface, routing-instance(VR)
  - to: zone, interface, routing-instance(VR)
- 条件にマッチしたら②の評価に

- ② パケットの情報 (NAT rule)

- 送信元アドレス, 宛先アドレス, ポート番号を条件として「どんな」通信かを指定
- マッチしたらNATを適用

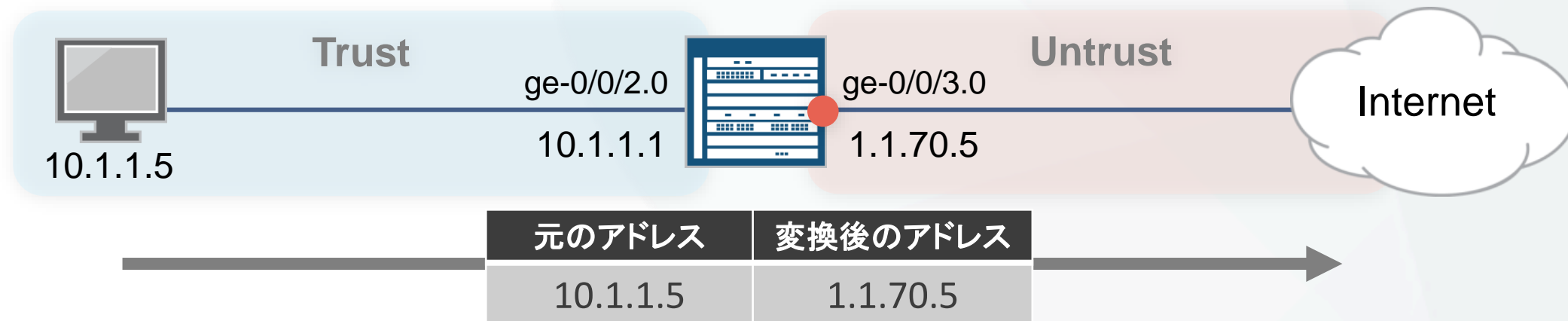


# Source NAT 概要

- 送信元IPアドレスを変換する
  - オプションで送信元ポート番号の変換(PAT)
- Source NATの種類
  - **Interface-based source NAT**
    - SRXのインターフェースアドレスに変換
    - PATは常時動作
  - **Pool-based source NAT**
    - PoolからIPアドレスを動的にアサイン
    - PATはあり、なしどちらも対応

# Interface-based source NAT 設定

- Trustから入ってきたトラフィックの送信元IPアドレスを Untrustの出口側インターフェースのIPアドレス“1.1.70.5”に変換



- NATルールセットで通信の方向を決定

```
set security nat source rule-set 1 from zone trust
set security nat source rule-set 1 to zone untrust
```

- NATルールを設定

- 送信元アドレス (0.0.0.0/0 = any) にマッチしたら、interfaceアドレスに変換

```
set security nat source rule-set 1 rule 1A match source-address 0.0.0.0/0
set security nat source rule-set 1 rule 1A then source-nat interface
```

# Interface-based source NAT 確認

- 変換結果の確認コマンド
  - show security flow session
  - show security nat source summary

```
user@srx> show security flow session
Session ID: 42325, Policy name: default-permit/4, Timeout: 1790
In: 10.1.1.5/1739 --> 1.1.70.6/23;tcp, If: ge-0/0/2.0
Out: 1.1.70.6/23 --> 1.1.70.5/1083;tcp, If: ge-0/0/3.0
```

```
user@srx> show security nat source summary
```

```
Total pools: 0
```

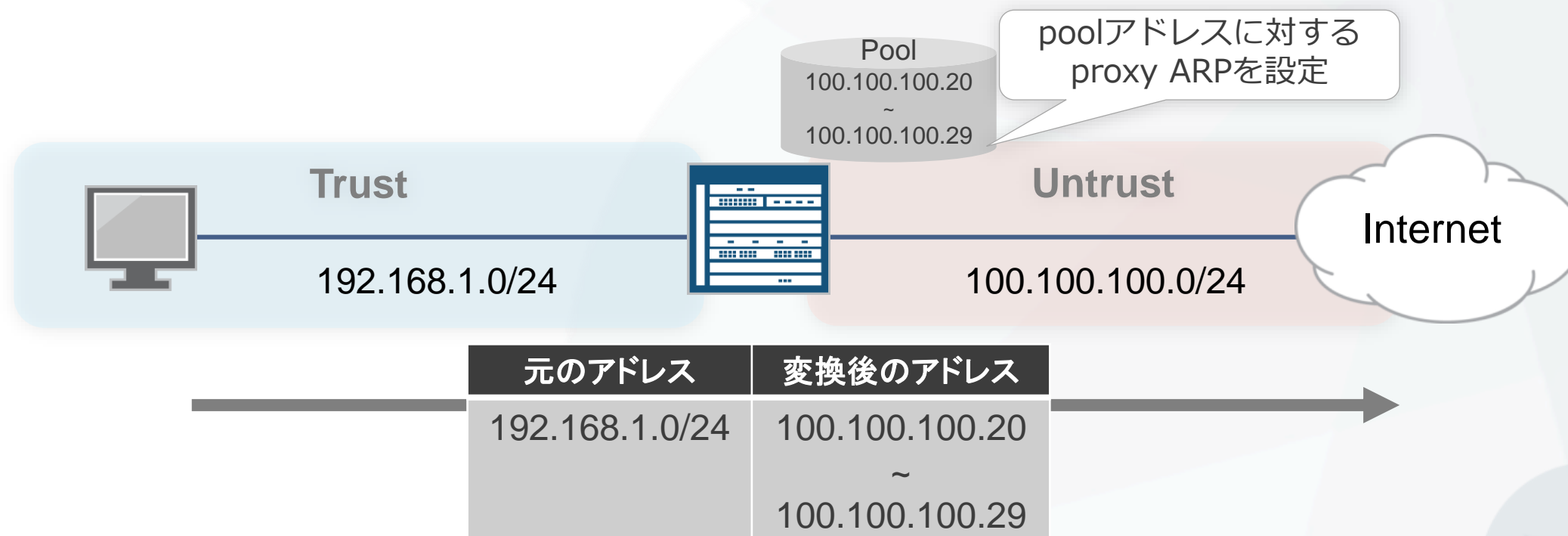
```
Total rules: 1
```

| Rule name | Rule set | From       | To      | Action    |
|-----------|----------|------------|---------|-----------|
| 1A        | 1        | ge-0/0/2.0 | untrust | interface |

PATも同時に動作

# Pool-based source NAT

- TrustからUntrustに抜けるトラフィックの送信元IPアドレスをpoolアドレスに変換
- Proxy ARPを設定
  - Poolアドレスに対してSRXからARP応答するように設定
  - Poolアドレスとインターフェースが同じサブネット上の場合に必要





# Pool-based source NAT 設定

- アドレスプールの設定

```
set security nat source pool src_nat_pool_napt address 100.100.100.20/32 to 100.100.100.29/32
```

- NATルールセットの設定

- TrustゾーンからUntrustゾーンへの通信

```
set security nat source rule-set src_nat_napt from zone trust
set security nat source rule-set src_nat_napt to zone untrust
```

- NATルールの設定

- 送信元アドレスが192.168.1.0/24の場合、Poolアドレスに変換

```
set security nat source rule-set src_nat_napt rule napt_1 match source-address 192.168.1.0/24
set security nat source rule-set src_nat_napt rule napt_1 then source-nat pool src_nat_pool_napt
```

- Proxy ARPの設定

```
set security nat proxy-arp interface ge-0/0/0.0 address 10.10.10.20/32 to 10.10.10.29/32
```

# Pool-based source NAT 確認

- 変換結果の確認コマンド
  - show security flow session
  - show security nat source summary

```
user@srx# run show security flow session
```

```
Session ID: 11010, Policy name: trust-to-untrust/4, Timeout: 1792, Valid
```

```
In: 192.168.1.22/57842 --> 100.100.100.254/23;tcp, If: vlan.0, Pkts: 5, Bytes: 227
```

```
Out: 100.100.100.254/23 --> 100.100.100.26/21626;tcp, If: ge-0/0/0.0, Pkts: 5, Bytes: 259
```

```
user@srx# run show security nat source summary
```

```
Total pools: 1
```

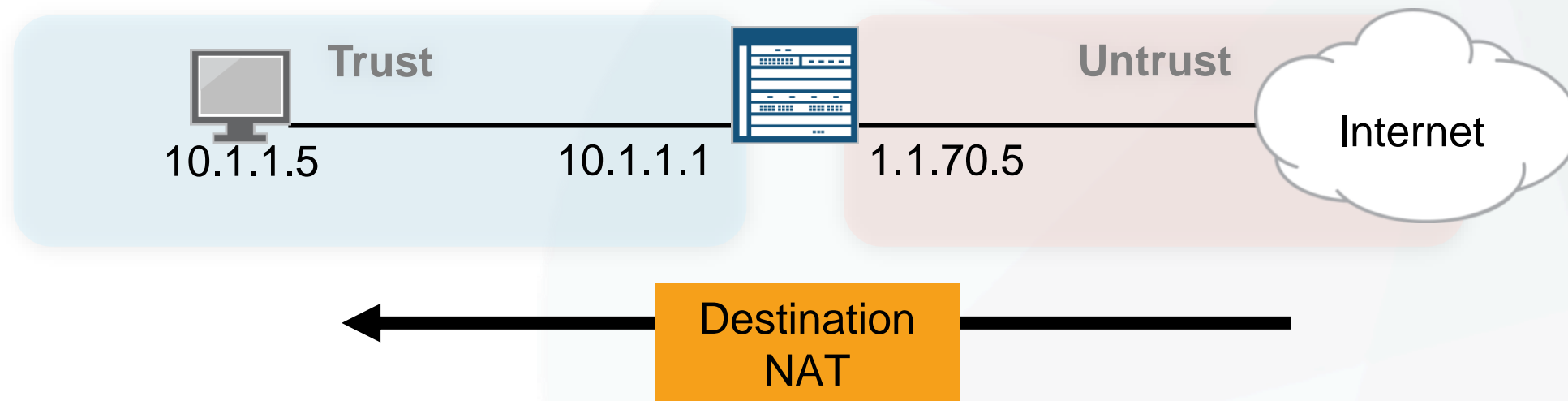
| Pool Name         | Address Range                 | Routing Instance | PAT | Total Address |
|-------------------|-------------------------------|------------------|-----|---------------|
| src_nat_pool_napt | 100.100.100.20-100.100.100.29 | default          | yes | 10            |

```
Total rules: 1
```

| Rule name | Rule set     | From  | To         | Action            |
|-----------|--------------|-------|------------|-------------------|
| napt_1    | src_nat_napt | trust | ge-0/0/0.0 | src_nat_pool_napt |

# Destination NAT 概要

- 宛先IPアドレスを変換する
  - オプションで宛先ポート番号の変換(PAT)
- Destination NAT
  - Pool-based NATのみ対応
    - 1:1マッピング
    - 1:Nマッピング (ポート変換による振り分け)



# Destination NAT (1:1) 設定

- アドレスプールの設定

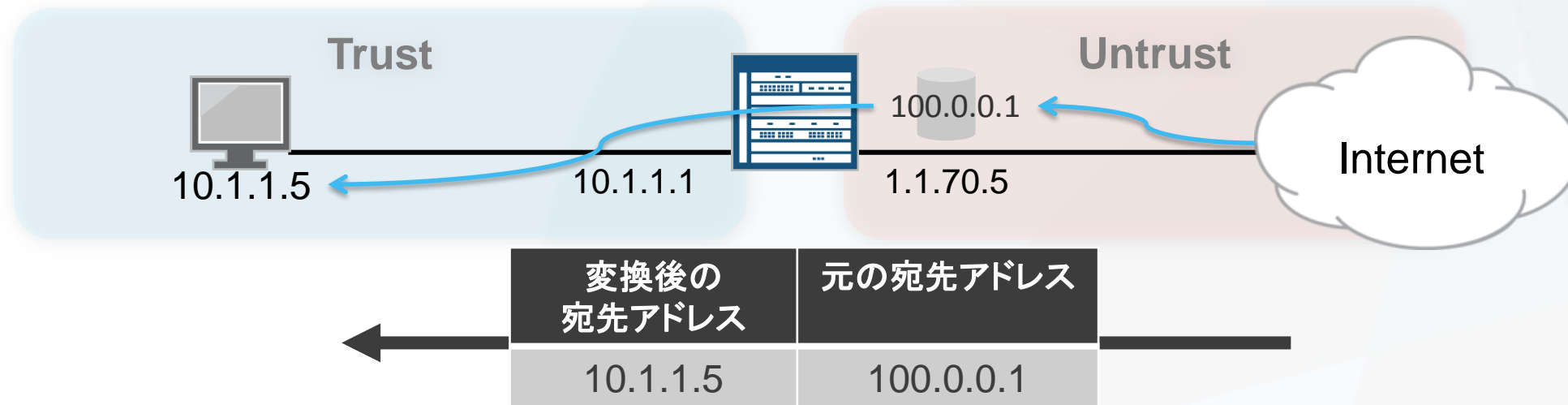
```
set security nat destination pool SVR_A address 10.1.1.5/32
```

- NATルールセットの設定

```
set security nat destination rule-set 1 from zone untrust
```

- NATルールの設定

```
set security nat destination rule-set 1 rule 1A match destination-address 100.0.0.1/32
set security nat destination rule-set 1 rule 1A then destination-nat pool SVR_A
```



# Destination NAT (1:N) 設定

- アドレスプールの設定

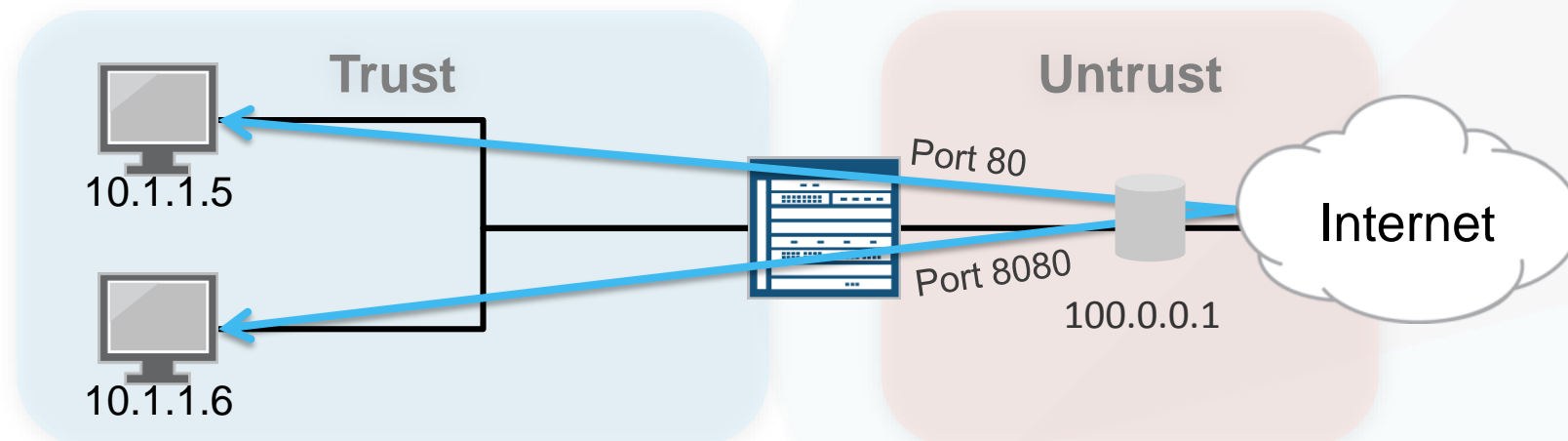
```
set security nat destination pool SVR_A address 10.1.1.5/32 port 80
set security nat destination pool SVR_B address 10.1.1.6/32 port 80
```

- NATルールセットの設定

```
set security nat destination rule-set 1 from zone untrust
```

- NATルールの設定

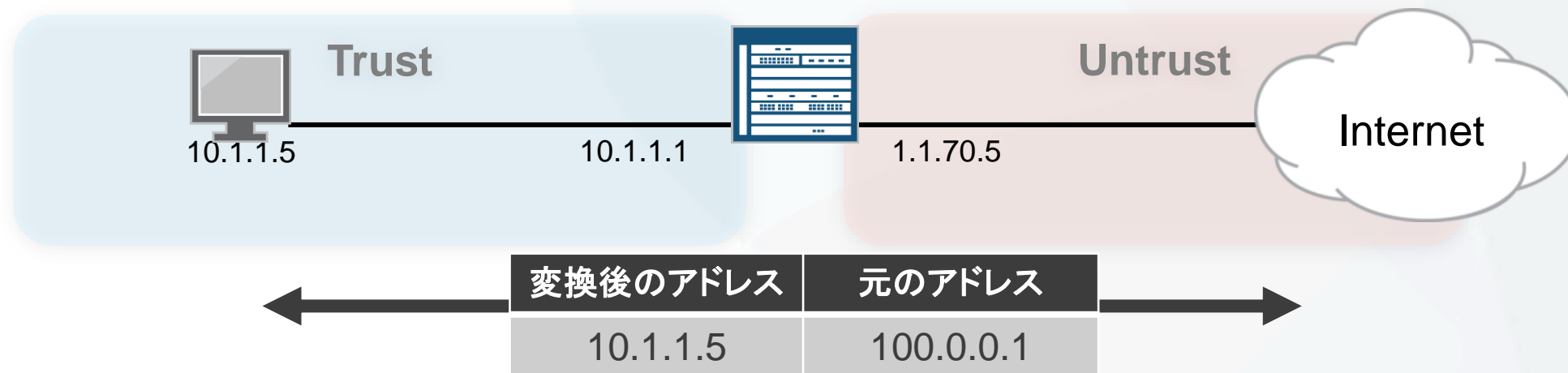
```
set security nat destination rule-set 1 rule 1A match destination-address 100.0.0.1/32
set security nat destination rule-set 1 rule 1A match destination-port 80
set security nat destination rule-set 1 rule 1A then destination-nat pool SVR_A
set security nat destination rule-set 1 rule 1B match destination-address 100.0.0.1/32
set security nat destination rule-set 1 rule 1B match destination-port 8080
set security nat destination rule-set 1 rule 1B then destination-nat pool SVR_B
```



| 変換後の宛先アドレス          | 元の宛先アドレス               |
|---------------------|------------------------|
| 10.1.1.5<br>Port 80 | 100.0.0.1<br>Port 80   |
| 10.1.1.6<br>Port 80 | 100.0.0.1<br>Port 8080 |

# Static NAT 概要

- 1:1でアドレスをマッピングして変換する
  - ポート変換動作はなし
  - 双方向に通信を開始可能



- Static NATの設定

```
set security nat static rule-set R1 from zone untrust
set security nat static rule-set R1 rule 1A match destination-address 100.0.0.1/32
set security nat static rule-set R1 rule 1A then static-nat prefix 10.1.1.5/32
```

# Static NAT 確認

- Untrustから100.0.0.1に対してpingを実行

```
user@srx> show security flow session
Session ID: 7724, Policy name: default-permit/4, Timeout: 2
 In: 1.1.70.6/17 --> 100.0.0.1/2326;icmp, If: ge-0/0/3.10
 Out: 10.1.1.5/2326 --> 1.1.70.6/17;icmp, If: ge-0/0/2.0
```

- 10.1.1.5からuntrustに対してpingを実行
  - 逆方向のStatic source NATは自動的に有効化されている

```
user@srx> show security flow session
Session ID: 18408, Policy name: default-permit/4, Timeout: 2
 In: 10.1.1.5/64513 --> 1.1.70.6/512;icmp, If: ge-0/0/2.0
 Out: 1.1.70.6/512 --> 100.0.0.1/64513;icmp, If: ge-0/0/3.10
```

# NAT設定 & 動作確認コマンド

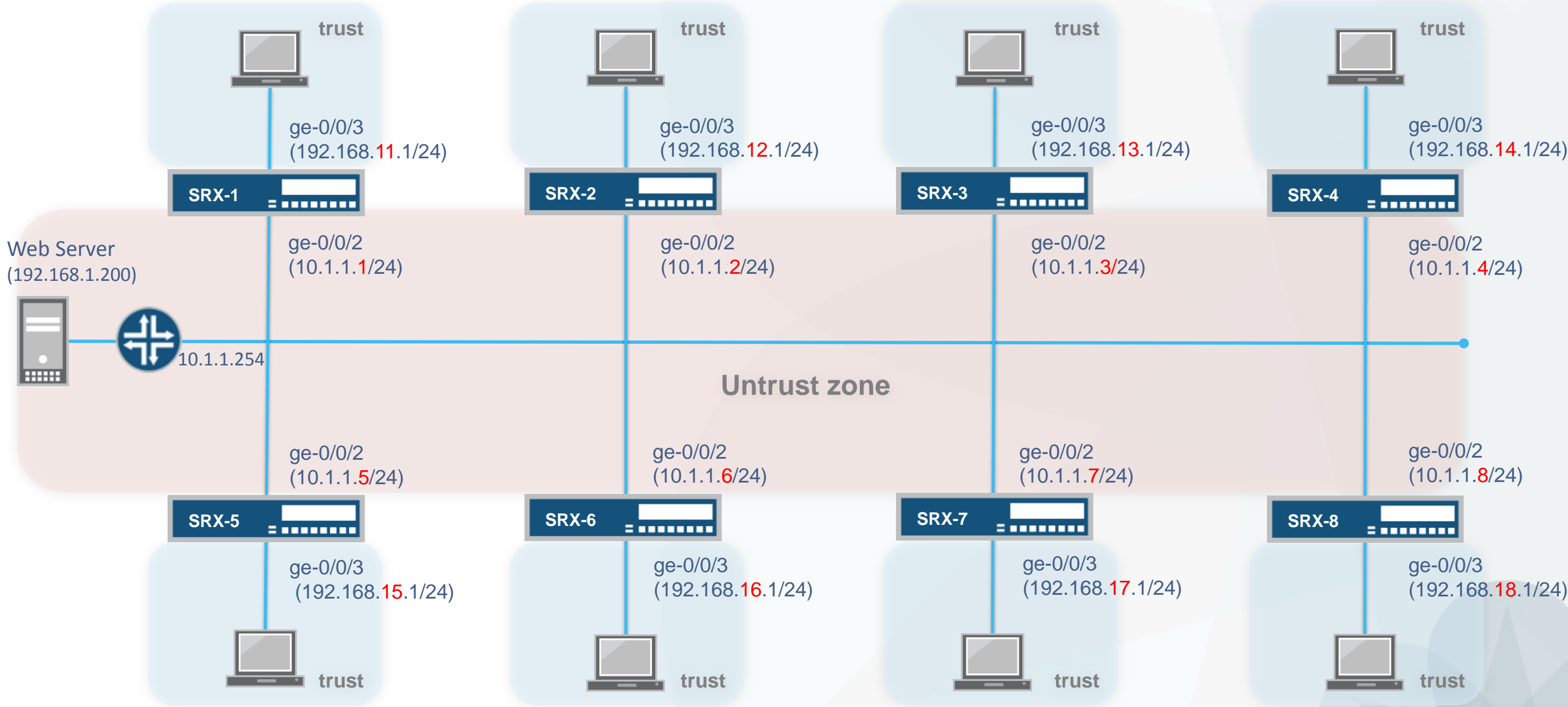
- セッション情報の確認
  - `show security flow session`
- Source NATの確認
  - `show security nat source pool <pool-name | all>`
  - `show security nat source rule <rule-name | all>`
  - `show security nat source summary`
- Destination NATの確認
  - `show security nat destination pool <pool-name | all>`
  - `show security nat destination rule <rule-name | all>`
  - `show security nat destination summary`
- Static NATの確認
  - `show security nat static rule <rule-name | all>`





# LAB.3 NAT

# Security "SRX" course Topology (Lab.3 : NAT)



# Interface-based source NAT

- Trust側の送信元IPアドレスを、SRXのuntrust IPに変換してください

| 元の送信元IP (trust) | 変換後の送信元IP (untrust) |
|-----------------|---------------------|
| 192.168.1X.0/24 | 10.1.1.X            |

- IPアドレスを確認
  - ブラウザから192.168.1.200にアクセスし、表示されるIPアドレスを確認

- ルールセットの設定

```
set security nat source rule-set interface-nat from zone trust
set security nat source rule-set interface-nat to zone untrust
```

- NATルールの設定

```
set security nat source rule-set interface-nat rule rule1 match source-address 0.0.0.0/0
set security nat source rule-set interface-nat rule rule1 then source-nat interface
```

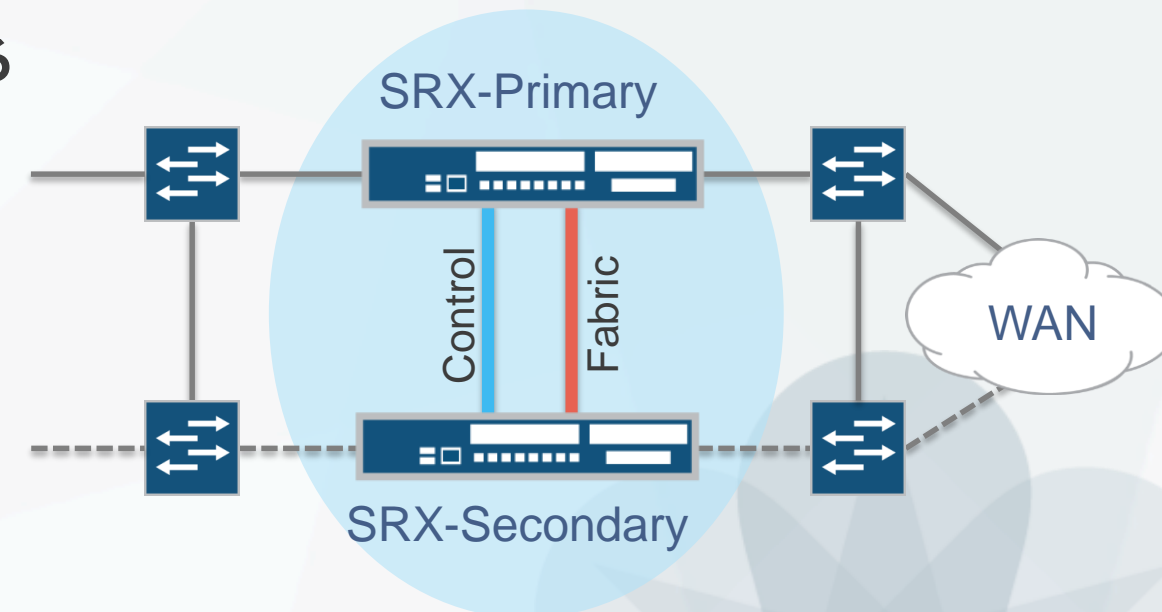
- 設定後、再度IPアドレスを確認
  - ブラウザから192.168.1.200に再度アクセスし、表示されるIPアドレスを確認



# Chassis Clusterの設定

# シャーシクラスタとは

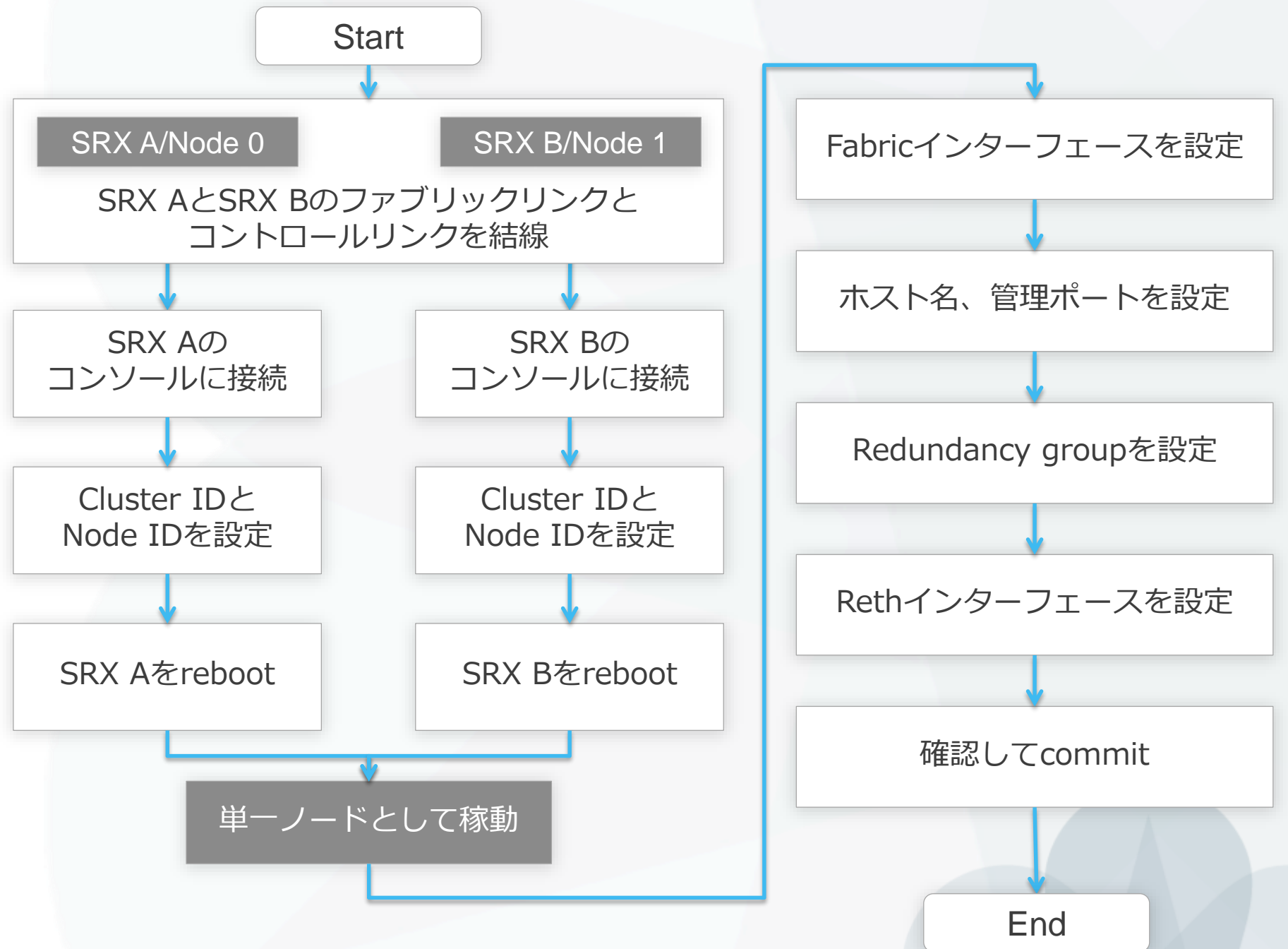
- 2台のSRXシリーズを“単一ノード”として動作させるための機能
  - ステートフルフェールオーバーを実現
    - セッション情報やconfigなど、2台で常に状態を同期
    - プライマリ機に障害が起きても、セカンダリ機が通信を継続
  - シャーシクラスタの構成
    - 相互に状態同期と死活監視をするため、2本の特別なリンクを設定
      - **コントロールリンク**
        - » コンフィグレーションとカーネルの状態を同期
        - » 機種ごとにどのポートを使うか事前に決まっている
      - **ファブリックリンク**
        - » セッション情報の同期とノード間のフロー処理
        - » 任意のポートに設定可能



# シャーシクラスタの設定フロー

## 事前に確認

- ✓ 2台のSRXが同じハードウェアであること
  - ✓ 同じバージョンのOSであること
  - ✓ 各種拡張セキュリティを使用時にはライセンスが同じ状態であること
- \*シャーシクラスタ用には不要



# SRX300 シャーシクラスタポート構成

- **Control link(fxp1)** : node0のge-0/0/1 - node1のge-1/0/1
- **Fabric link(fab0/fab1)** : 任意のポート



- インターフェーススロットのナンバリング



# SRX320 シャーシクラスタポート構成

- **Control link(fxp1)** : node0のge-0/0/1 - node1のge-3/0/1
- **Fabric link(fab0/fab1)** : 任意のポート



- インターフェーススロットのナンバリング





# SRX340/345 シャーシクラスタポート構成

- **Control link(fxp1)** : node0のge-0/0/1 - node1のge-5/0/1
- **Fabric link(fab0/fab1)** : 任意のポート

Node0



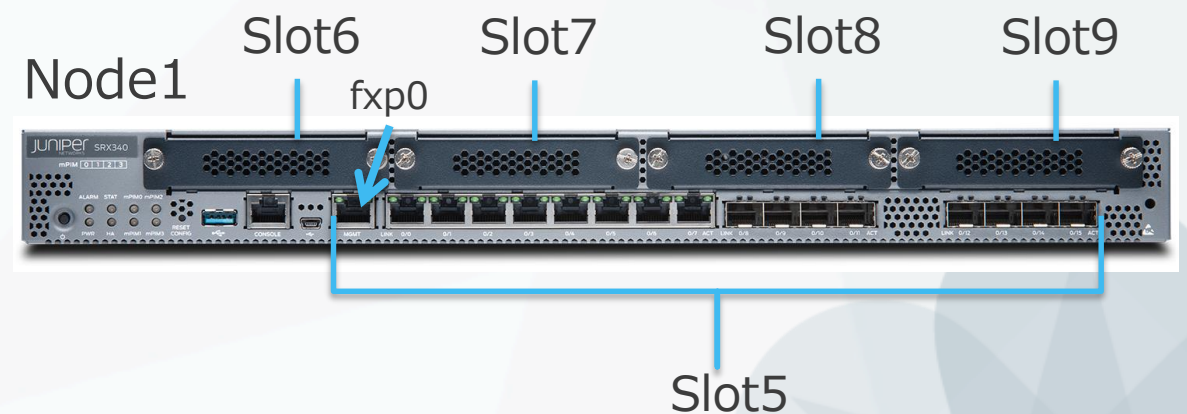
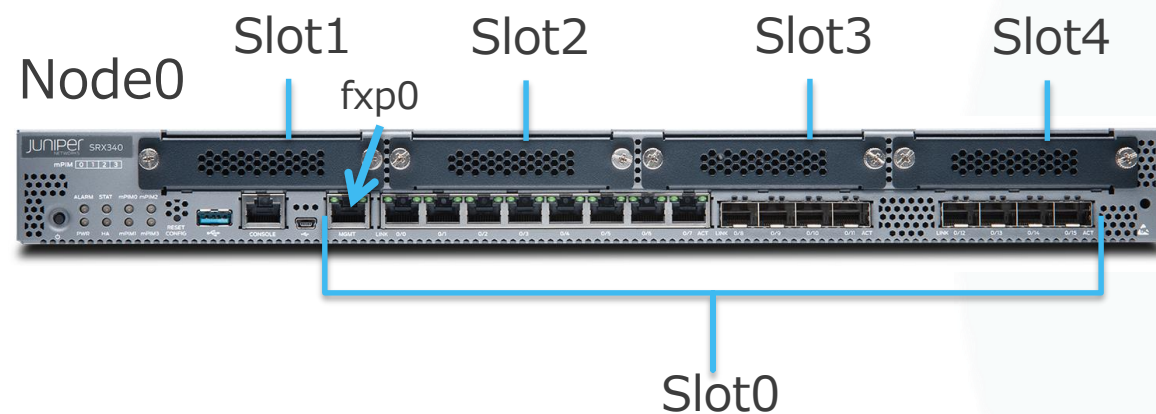
Node1



コントロールリンク

ファブリックリンク

- インターフェイススロットのナンバリング



# SRX550 シャーシクラスポート構成

- Control link(fxp1) : node0のge-0/0/1 - node1のge-9/0/1
- Fabric link(fab0/fab1) : 任意のポート

Node0



コントロールリンク

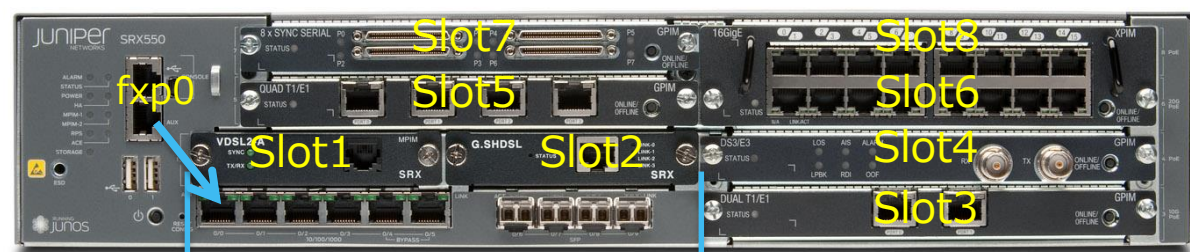
ファブリックリンク

Node1



- インターフェーススロットのナンバリング

Node0



Slot0

Node1



Slot9

# SRX1500 シャーシクラスポート構成

- Control link(em0) : 専用コントロールポート
- Fabric link(fab0/fab1) : 任意のポート

Node0



Node1



コントロールリンク

ファブリックリンク

- インターフェーススロットのナンバリング

Node0



Slot0

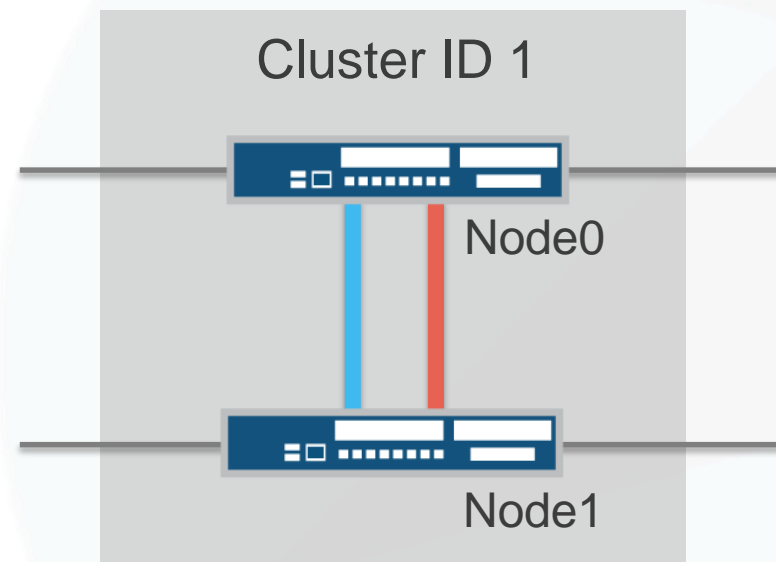
Node1



Slot7

# Cluster IDとNode ID

- Cluster ID
  - クラスタごとに固有に設定するID
  - 同じL2ドメインで1~255まで設定可能
- Node ID
  - クラスタ内でのデバイス固有のID
  - node0かnode1を指定
    - Node IDによりインターフェース番号はリナンバリングされます
    - どちらがプライマリ、セカンダリになるかはIDとは別にプライオリティで設定



# 各ノードにCluster IDとNode IDを設定

- Cluster IDとNode IDを設定
  - Operationalモードで、以下コマンドを実行

- Node0

```
user@srx> set chassis cluster cluster-id 1 node 0 reboot
```

- node1

```
user@srx> set chassis cluster cluster-id 1 node 1 reboot
```

- これらの情報はEPROMに保存される（configには保存されない）
- 設定を反映させるためにはrebootが必要
  - rebootオプションを使うと、コマンド実行直後にreboot

```
user@srx> set chassis cluster cluster-id 1 node 0 reboot
Successfully enabled chassis cluster. Going to reboot now.
```

```
user@srx>
*** FINAL System shutdown message from root@vSRX1 ***
```

```
System going down IMMEDIATELY
```

# シャーシクラスタの状態確認

- 各ノードの再起動後、クラスタが形成される
  - `show chassis cluster status`コマンドで状態を確認

```
lab@srx> show chassis cluster status
Monitor Failure codes:
 CS Cold Sync monitoring FL Fabric Connection monitoring
 GR GRES monitoring HW Hardware monitoring
 IF Interface monitoring IP IP monitoring
 LB Loopback monitoring MB Mbuf monitoring
 NH Nexthop monitoring NP NPC monitoring
 SP SPU monitoring SM Schedule monitoring
 CF Config Sync monitoring

Cluster ID: 1
Node Priority Status Preempt Manual Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 1 primary no no None
node1 1 secondary no no None
```

- 各ノードのプロンプト上のステータスを確認

```
{primary:node0}
lab@srx>
```

```
{secondary:node1}
lab@srx>
```

# ファブリックリンクを設定

- ファブリックリンクを設定
  - Configurationモードで以下コマンドを設定
    - node0側のge-0/0/5をfab0, node1側のge-1/0/5をfab1として設定

```
set interfaces fab0 fabric-options member-interfaces ge-0/0/5
set interfaces fab1 fabric-options member-interfaces ge-1/0/5
```



# 各ノードにホスト名と管理ポートを設定

- 各ノード固有のホスト名と管理ポートを設定
  - シャーシクラスタでは両ノードが同じconfigを共有する
  - ノード固有のconfigを設定したい場合に、groupsオプションを利用
    - Node0用のホスト名、管理ポートを設定

```
set groups node0 system host-name SRX_node0
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.0.101/24
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.0.100/24 master-only
```

各ノードのfxp0に固有のIPを設定

- Node1用のホスト名、管理ポートを設定

```
set groups node1 system host-name SRX_node1
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.0.102/24
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.0.100/24 master-only
```

Primaryにログインできる共通のIPを設定

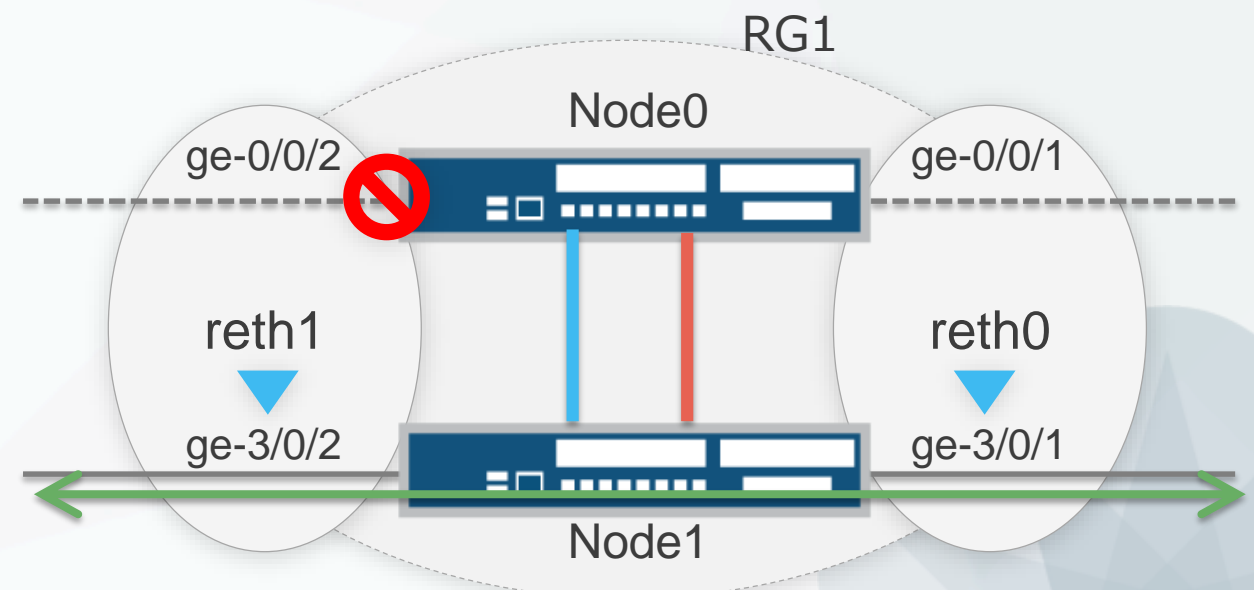
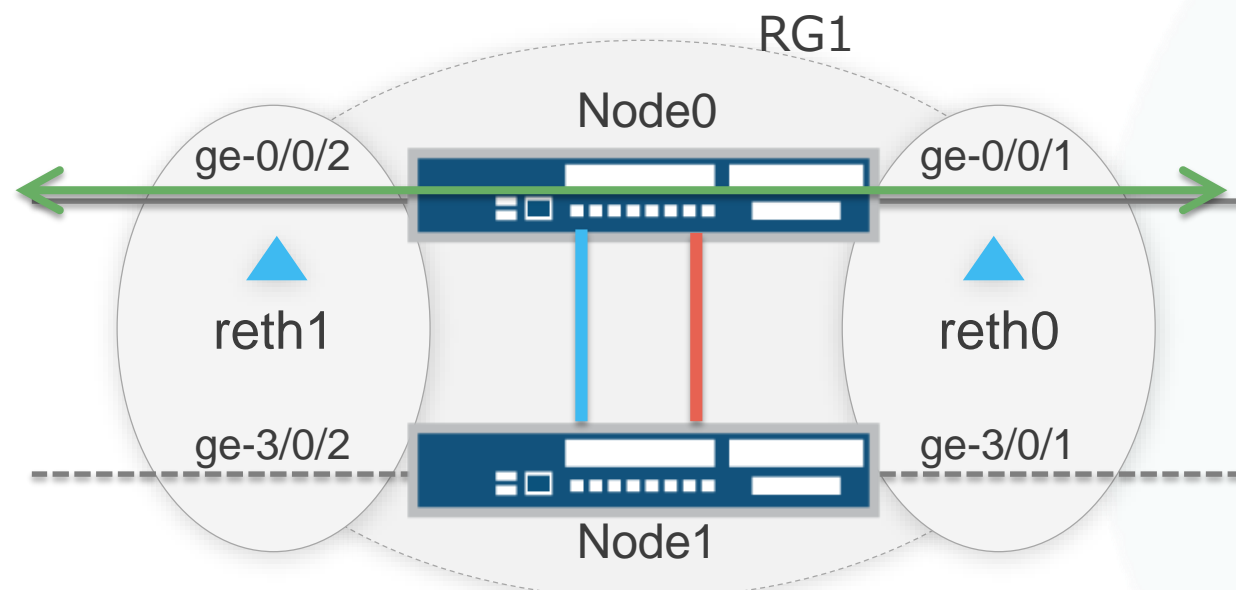
- 2つのグループ設定を適用

```
set apply-groups "${node}"
```



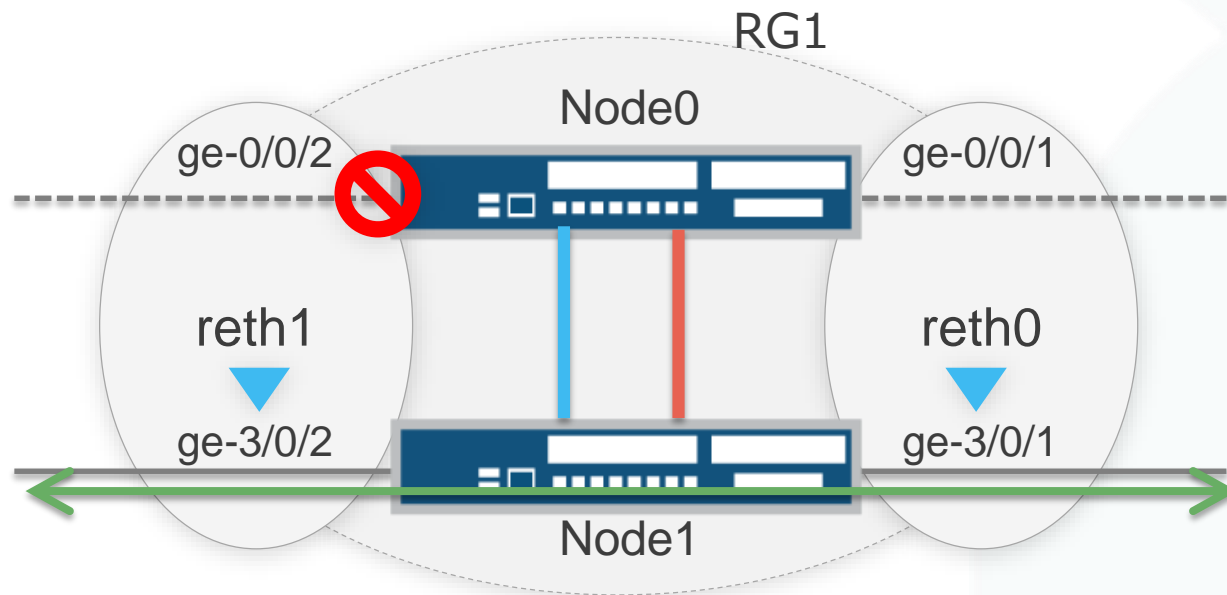
# RETHインターフェースとRedundancy Group①

- Redundant Ethernet Interface(RETH)
  - 2台のノード間で共有される仮想のインターフェース
    - 各ノードの物理リンクを1つのrethにマッピング
  - どちらのノードがrethの転送を担当するか？
    - “Redundancy Group”でノードごとにプライオリティ付け
    - より高いプライオリティを持つノードがプライマリとして転送を担当
  - プライマリに障害が起きた場合
    - 同じRedundancy Groupに所属するすべてのrethがセカンダリにフェールオーバー

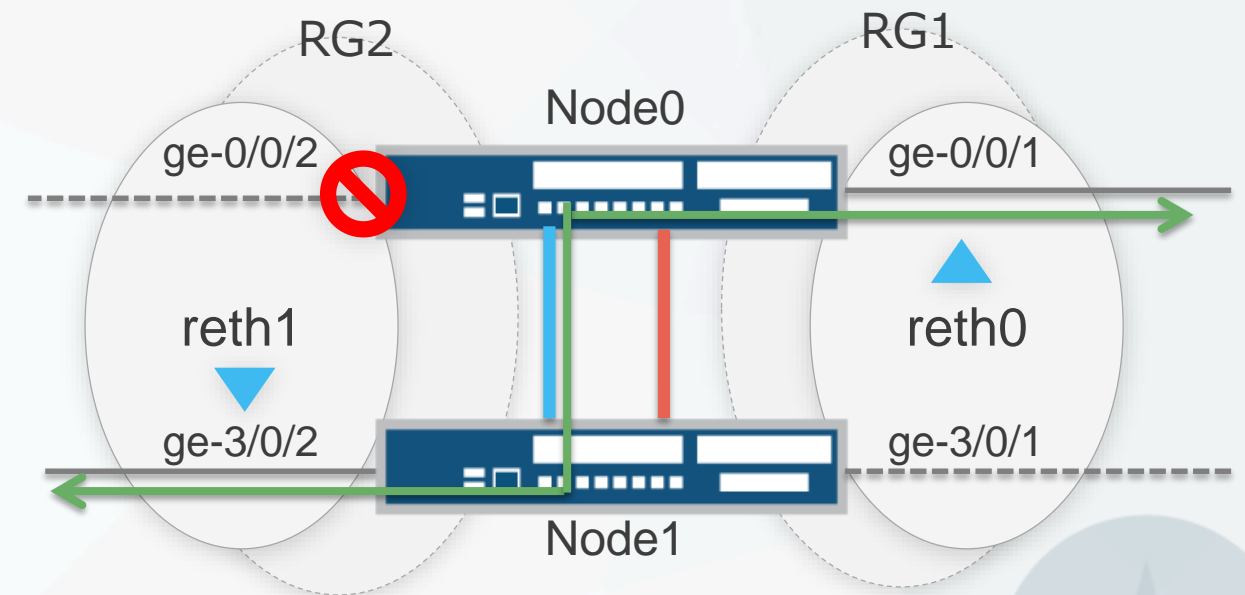


# RETHインターフェースとRedundancy Group②

- Redundancy Group(RG)
  - 障害発生時にフェールオーバーの影響を共有する範囲を指定するためのグループ
    - 1つのグループ内では、どちらかのノードがプライマリとして処理を担当
    - グループごとにノードにプライオリティを設定し、プライマリノードを決定



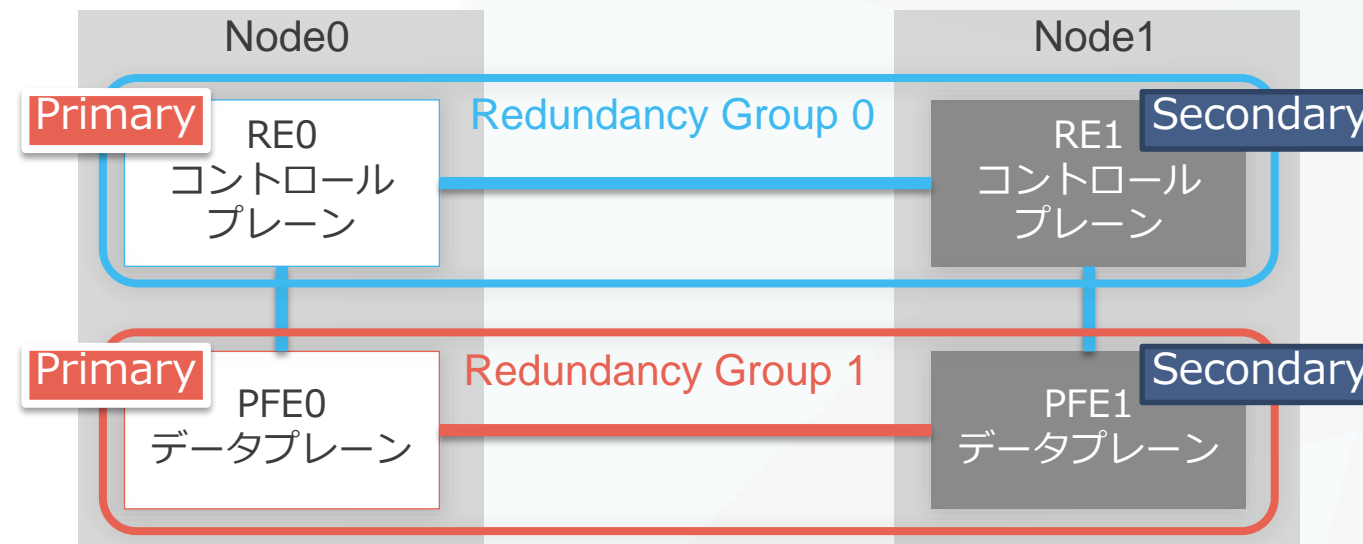
RG1にReth0とReth1をバインドした場合  
(障害時にはReth0とReth1がFailover)



RG1にReth0とRG2にReth1をバインドした場合  
(障害時にはReth1だけがFailover)

# RETHインターフェースとRedundancy Group③

- Redundancy Group(RG) 0と1+
  - RG0はChassis Clusterを制御するコントロールプレーン用のRGとしてシステムに予約されている
    - Redundancy Group 0
      - ルーティングエンジン（コントロールプレーン）のRedundancy Group
    - Redundancy Group 1～ 以上
      - rethインターフェース（データプレーン）のRedundancy Group

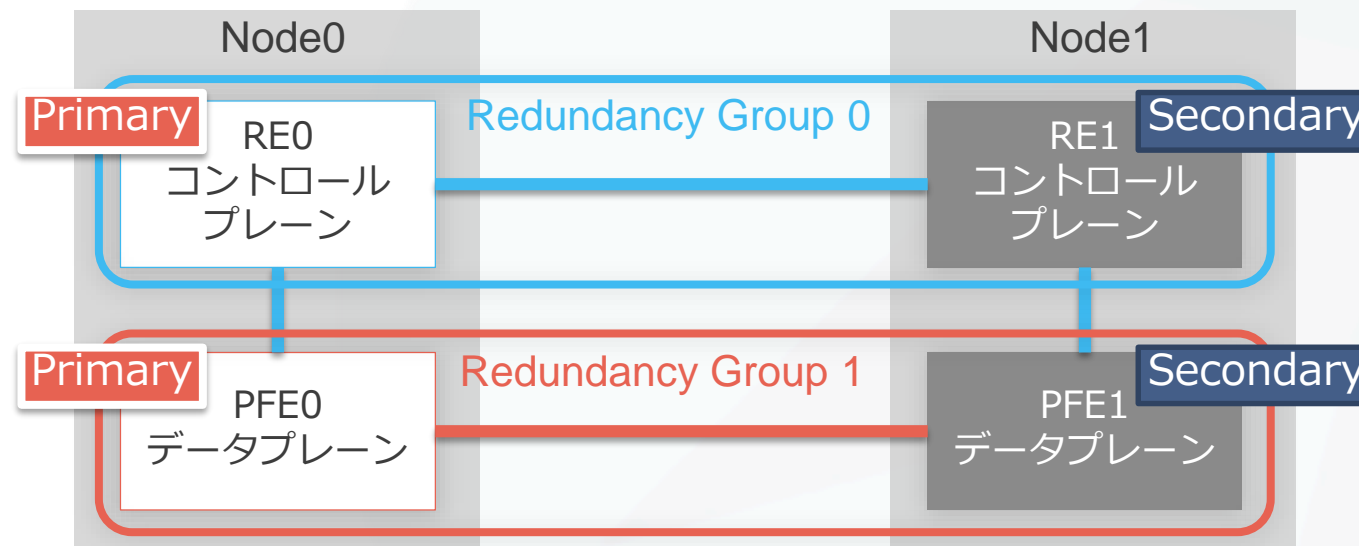


# REDUNDANCY GROUPの設定

- 各ノードをRedundancy Groupに所属させ、プライオリティを設定
  - コンフィギュレーションモードで以下コマンドを設定

```
set chassis cluster redundancy-group 0 node 0 priority 200
set chassis cluster redundancy-group 0 node 1 priority 100
set chassis cluster redundancy-group 1 node 0 priority 200
set chassis cluster redundancy-group 1 node 1 priority 100
```

- Redundancy Group 0
  - ルーティングエンジン(RE)共有のグループ
- Redundancy Group 1～ 以上
  - インターフェース(PFE)共有のグループ



# REDUNDANT ETHERNET INTERFACE(RETH)の設定

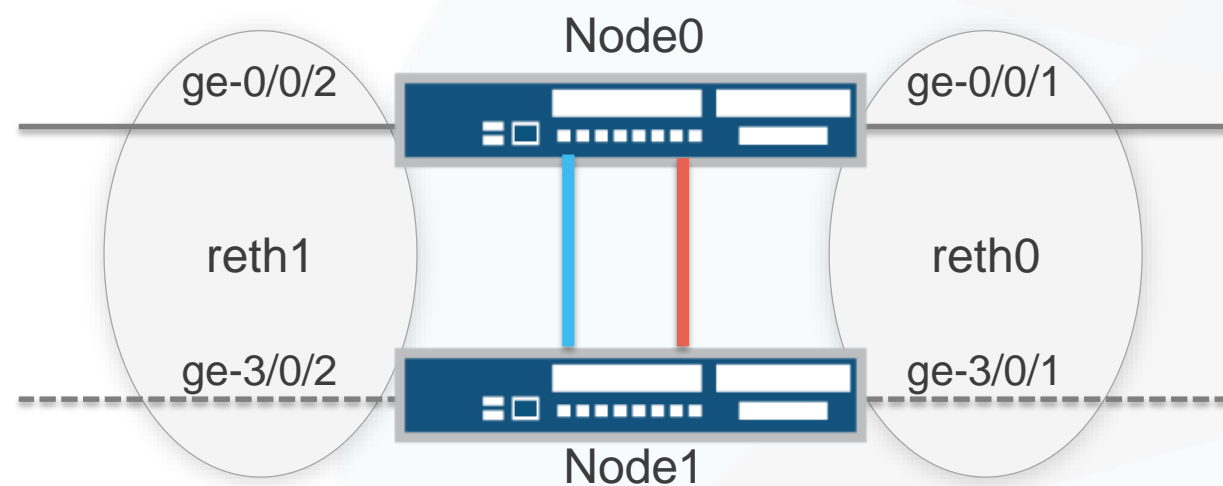
- コンフィギュレーションモードで以下コマンドを設定

- クラスタ内のrethインターフェースの総数を定義

```
set chassis cluster reth-count 2
```

- rethにバインドする物理(または論理)インターフェースを設定

```
set interfaces ge-0/0/1 gigether-options redundant-parent reth0
Set interfaces ge-3/0/1 gigether-options redundant-parent reth0
Set interfaces ge-0/0/2 gigether-options redundant-parent reth1
Set interfaces ge-3/0/2 gigether-options redundant-parent reth1
```



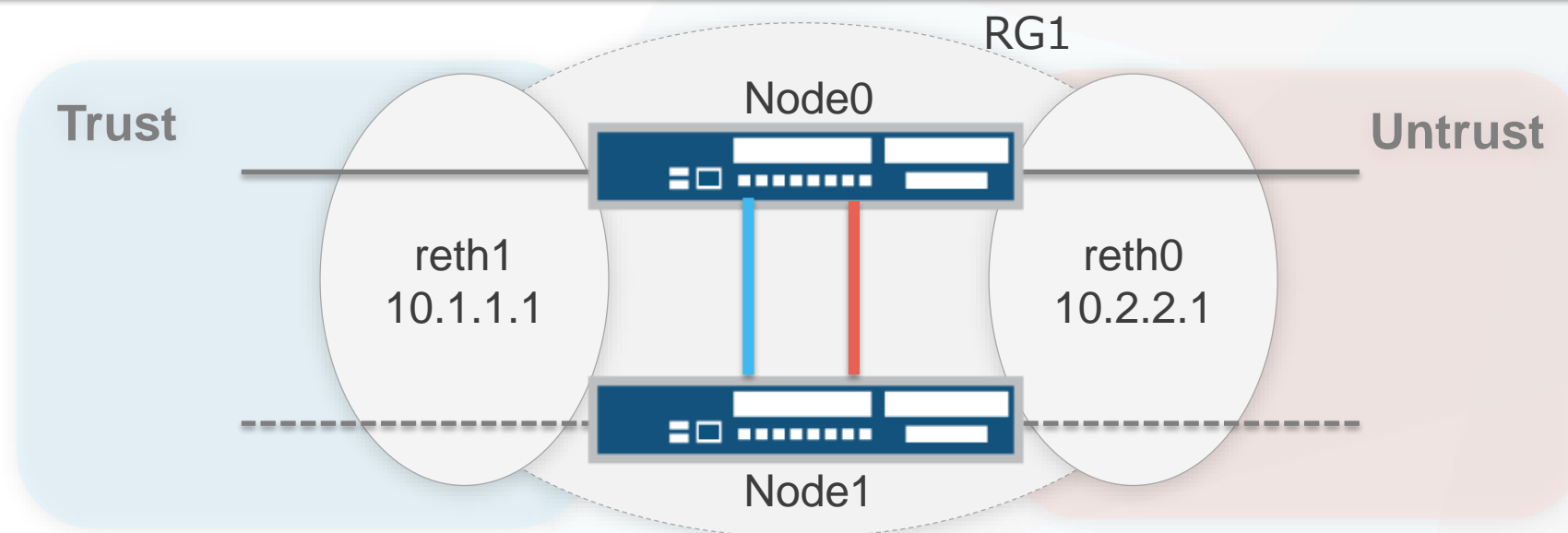
# REDUNDANT ETHERNET INTERFACE(RETH)の設定

- rethをRedundancy Groupに所属させ、IPアドレスを設定

```
set interfaces reth1 redundant-ether-options redundancy-group 1
Set interfaces reth1 unit 0 family inet address 10.1.1.1/24
set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.2.2.1/24
```

- rethをSecurity zoneにバインドする
  - reth1をtrustに、reth0をuntrustにバインド

```
set security zones security-zone trust interfaces reth1.0
set security zones security-zone unrust interfaces reth0.0
```



# プリアンプトとインターフェースモニタリングの設定

- プリアンプトとインターフェースモニタリングの設定
  - Redundancy Groupにプリアンプトを設定
    - 障害復旧時にプライオリティの高いノード側をプライマリに戻す動作

```
set chassis cluster redundancy-group 1 preempt
```

- インターフェースモニタリングを設定
  - RGごとに切り替わりのトリガーとなるインターフェースとweightを指定

```
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-3/0/1 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-3/0/2 weight 255
```

- インターフェースがダウンした時に設定したweight分のプライオリティを下げる
- プライオリティ値がセカンダリよりも下がったときにRGのfailoverが発生

# シャーシクラスタ確認コマンド①

- **show interfaces terse** : インターフェースの確認
  - ファブリックリンクの確認

```
lab@SRX-1> show interfaces terse | match fab
ge-0/0/0.0 up up aenet --> fab0.0
ge-7/0/0.0 up up aenet --> fab1.0
fab0 up up
fab0.0 up up inet 30.17.0.200/24
fab1 up up
fab1.0 up up inet 30.18.0.200/24
```

- rethインターフェースの確認

```
lab@SRX-1> show interfaces terse | match reth
ge-0/0/1.0 up up aenet --> reth0.0
ge-0/0/2.0 up up aenet --> reth1.0
ge-7/0/1.0 up up aenet --> reth0.0
ge-7/0/2.0 up up aenet --> reth1.0
reth0 up up
reth0.0 up up inet 10.1.1.1/24
reth1 up up
reth1.0 up up inet 10.2.2.1/24
```



# シャーシクラスタ確認コマンド②

- **show chassis cluster interface**
  - クラスタに所属するインターフェースの確認

```
lab@SRX-1> show chassis cluster interfaces
Control link status: Up

Control interfaces:
 Index Interface Monitored-Status Internal-SA
 0 em0 Up Disabled

Fabric link status: Up

Fabric interfaces:
 Name Child-interface Status
 (Physical/Monitored)
 fab0 ge-0/0/0 Up / Up
 fab0
 fab1 ge-7/0/0 Up / Up
 fab1

Redundant-ethernet Information:
 Name Status Redundancy-group
 reth0 Up 1
 reth1 Up 1

~~~~~
```

# シャーシクラスタ確認コマンド③

- `show chassis cluster status`
  - シャーシクラスタのステータスを表示

```
lab@SRX-1> show chassis cluster status
Monitor Failure codes:
  CS Cold Sync monitoring          FL Fabric Connection monitoring
  GR GRES monitoring              HW Hardware monitoring
  IF Interface monitoring         IP IP monitoring
  LB Loopback monitoring          MB Mbuf monitoring
  NH Nexthop monitoring           NP NPC monitoring
  SP SPU monitoring               SM Schedule monitoring
  CF Config Sync monitoring

Cluster ID: 1
Node  Priority Status          Preempt Manual  Monitor-failures

Redundancy group: 0 , Failover count: 1
node0 200      primary      no    no    None
node1 100      secondary    no    no    None

Redundancy group: 1 , Failover count: 1
node0 200      primary      no    no    None
node1 100      secondary    no    no    None
```

# シャーシクラスタ確認コマンド④

- **show chassis cluster statistics**
  - シャーシクラスタの統計情報

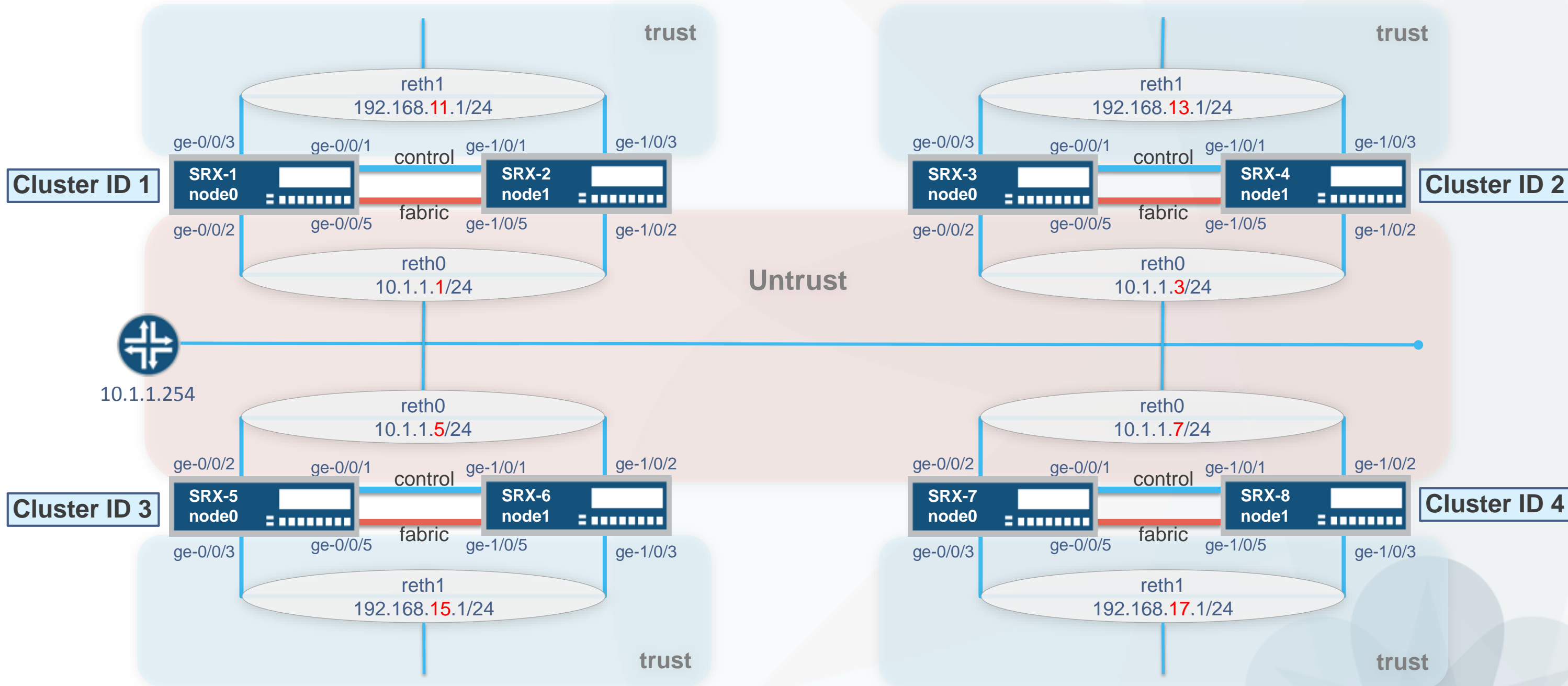
```
lab@SRX-1> show chassis cluster statistics
Control link statistics:
  Control link 0:
    Heartbeat packets sent: 73964
    Heartbeat packets received: 73342
    Heartbeat packet errors: 0
Fabric link statistics:
  Child link 0
    Probes sent: 145811
    Probes received: 145810
  Child link 1
    Probes sent: 0
    Probes received: 0
Services Synchronized:
  Service name          RTOs sent  RTOs received
  Translation context   0          0
  Incoming NAT          0          0
~~~~~
```



# LAB.4

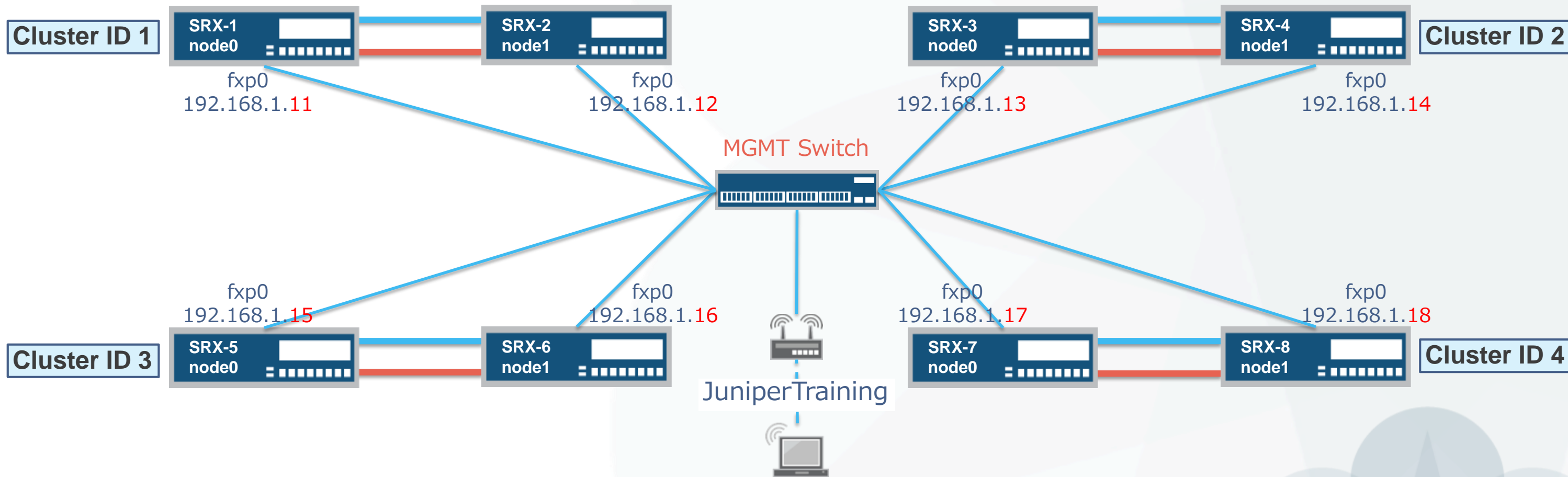
## Chassis Cluster

# Security "SRX" course Topology (Lab.5 : Chassis Clustering)

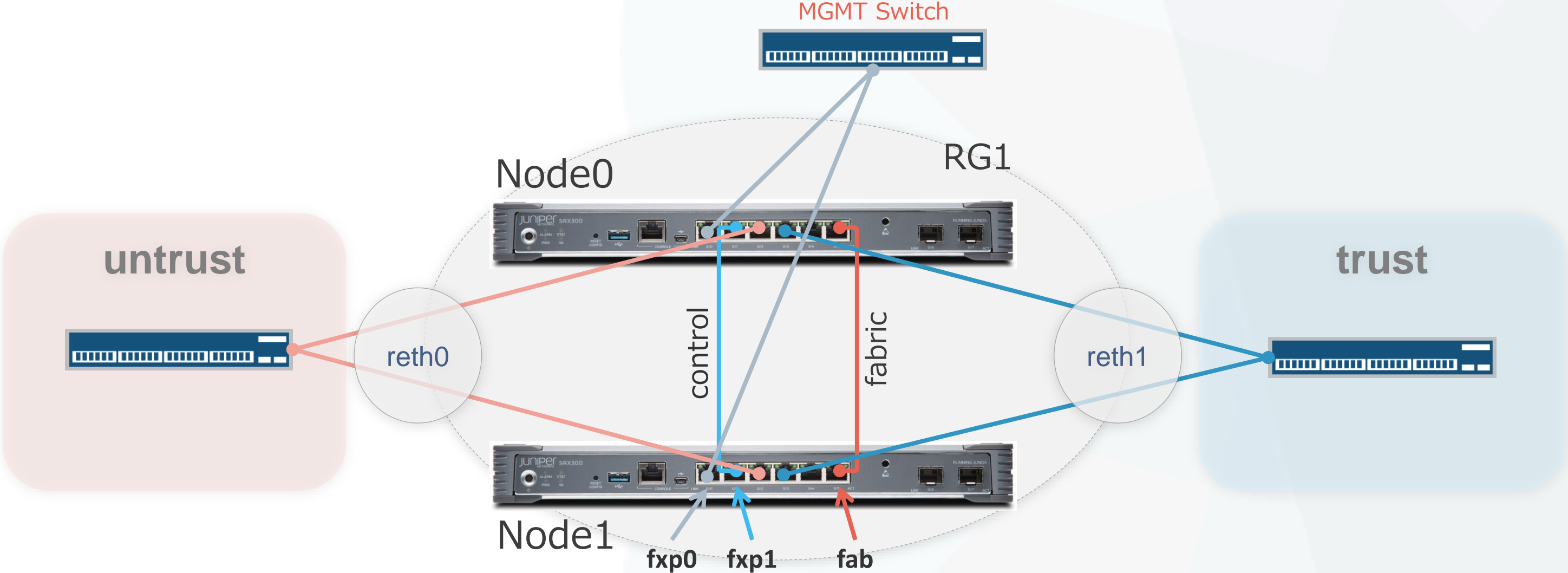


# Security "SRX" course Topology (Lab.5 : Chassis Cluser Management)

- 管理用IPアドレス一覧
  - Fxp0に設定するIPアドレス



# SRX300 シャーシクラスポート構成



# Chassis Clusterの設定

- トポロジー図に従ってChassis Clusterを組んでください
  - Active / Passive構成
  - コントロールリンクとファブリックリンクは1本ずつ用意
  - RETH0をuntrust側インターフェースとして構成
  - RETH1をtrust側インターフェースとして構成
- 以下のコマンドで、ステータスを確認してください
  - show chassis cluster status
  - show chassis cluster interface
  - show chassis cluster statistics



# Cluster IDとNode IDの設定

- Node固有の設定を追加（座席番号が奇数の方のみ）
  - IPアドレスは管理用IPアドレス図を参照

```
set groups node0 system host-name SRX-x_node0
set groups node0 interfaces fxp0 unit 0 family inet address 192.168.1.1x/24
set groups node1 system host-name SRX-x_node1
set groups node1 interfaces fxp0 unit 0 family inet address 192.168.1.1x/24
set apply-groups ${node}
```

- コントロールリンクとファブリックリンクを結線後、以下を実行
  - Node0（座席番号が奇数）

```
user@srx> set chassis cluster cluster-id x node 0 reboot
```

- Node1（座席番号が偶数）

```
user@srx> set chassis cluster cluster-id x node 1 reboot
```

- コマンド実行後、即時rebootに入ります

# 再起動後の状態確認

- WirelessのSSIDを”**JuniperTraining**”に変更してください
- PCのIPアドレスが192.168.1.xxに変わったことを確認します
- 起動後、単一ノードとして稼動状態を確認
  - 設定した管理用IPアドレスにtelnetでログインします
  - CCが組めていれば以下のようにステータスが表示されます

```
{primary:node0}
lab@SRX-1>
```

```
{secondary:node1}
lab@SRX-2>
```

- show chassis cluster statusコマンド
  - Redundancy Group 0のステータスが以下のようにになっていること
    - node0 Primary
    - node1 Secondary

# ファブリックリンクの設定

- 以下の設定変更はすべてPrimary(node0)から実施してください

## ①これまでのLabで設定した不要なconfigを削除

```
delete system host-name
delete system services dhcp
delete interfaces
delete security nat
delete security zones security-zone trust interfaces
delete security zones security-zone untrust interfaces
```

## ②Fabricリンクの設定を追加

```
set interfaces fab0 fabric-options member-interfaces ge-0/0/5
set interfaces fab1 fabric-options member-interfaces ge-1/0/5
```

# Redundant Group(RG)とRETHインターフェースの設定

- 以下の設定変更はすべてPrimary(node0)から実施してください
  - RG0とRG1を設定

```
set chassis cluster redundancy-group 0 node 0 priority 200
set chassis cluster redundancy-group 0 node 1 priority 100
set chassis cluster redundancy-group 1 node 0 priority 200
set chassis cluster redundancy-group 1 node 1 priority 100
```

- RETH0とRETH1を設定

```
set chassis cluster reth-count 2
set interfaces ge-0/0/2 ether-options redundant-parent reth0
set interfaces ge-1/0/2 ether-options redundant-parent reth0
set interfaces ge-0/0/3 ether-options redundant-parent reth1
set interfaces ge-1/0/3 ether-options redundant-parent reth1

set interfaces reth0 redundant-ether-options redundancy-group 1
set interfaces reth0 unit 0 family inet address 10.1.1.x/24
set interfaces reth1 redundant-ether-options redundancy-group 1
set interfaces reth1 unit 0 family inet address 192.168.1x.1/24
```

# インターフェースモニタリングとプリエンプトの設定

- 以下の設定変更はすべてPrimary(node0)から実施してください
  - rethをSecurity zoneにバインドする

```
set security zones security-zone trust interfaces reth1.0
set security zones security-zone untrust interfaces reth0.0
```

- インターフェースモニタリングとプリエンプトを設定

```
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-1/0/2 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-0/0/3 weight 255
set chassis cluster redundancy-group 1 interface-monitor ge-1/0/3 weight 255
set chassis cluster redundancy-group 1 preempt
```

# Chassis Clusterの動作確認

- 以下のコマンドで、ステータスを確認してください
  - show chassis cluster status
  - show chassis cluster interface
  - show chassis cluster statistics
- 障害動作確認
  - SRXから10.1.1.254に対してpingを実行します
  - RETH0のプライマリリンク（node0のge-0/0/2）のケーブルを抜きます
  - 通信が継続していることを確認します



# TIPs to be JUNOS Experts

# 俳句の表示

検証作業やトラブルシュー트에疲れたときには、JUNOSに前向きな気持ちの言葉を表示させ、管理者の気持ちを和らげることが可能です

```
root> show version and haiku
```

```
root> show version and haiku
Model: ex2200-c-12p-2g
Junos: 14.1X53-D25.2
JUNOS EX Software Suite [14.1X53-D25.2]
JUNOS FIPS mode utilities [14.1X53-D25.2]
JUNOS Online Documentation [14.1X53-D25.2]
JUNOS EX 2200 Software Suite [14.1X53-D25.2]
JUNOS Web Management Platform Package
[14.1X53-D25.2]
```

```
Look, mama, no hands!
Only one finger typing.
Easy: commit scripts.
```

```
root> show version and haiku
Model: ex2200-c-12p-2g
Junos: 14.1X53-D25.2
JUNOS EX Software Suite [14.1X53-D25.2]
JUNOS FIPS mode utilities [14.1X53-D25.2]
JUNOS Online Documentation [14.1X53-D25.2]
JUNOS EX 2200 Software Suite [14.1X53-D25.2]
JUNOS Web Management Platform Package
[14.1X53-D25.2]
```

```
Juniper babies
The next generation starts
Gotta get more sleep
```

```
root> show version and haiku
Model: ex2200-c-12p-2g
Junos: 14.1X53-D25.2
JUNOS EX Software Suite [14.1X53-D25.2]
JUNOS FIPS mode utilities [14.1X53-D25.2]
JUNOS Online Documentation [14.1X53-D25.2]
JUNOS EX 2200 Software Suite [14.1X53-D25.2]
JUNOS Web Management Platform Package
[14.1X53-D25.2]
```

```
Weeks of studying,
Days of lab exercises:
JNCIE.
```

※コマンドを打つ度、異なった前向きなポエムが表示される



# 設定のコピー

- copy コマンドにより特定の設定をコピーすることが可能

ge-0/0/1の設定をge-0/0/0へコピー

```
root# copy interfaces ge-0/0/1 to ge-0/0/0
```

```
root# show interfaces
ge-0/0/1 {
 unit 0 {
 family inet {
 address
 }
 }
 192.168.1.1/26;
}
```



```
root# show interfaces
ge-0/0/0 {
 unit 0 {
 family inet {
 address
 }
 }
 192.168.1.1/26;
}
ge-0/0/1 {
 unit 0 {
 family inet {
 address
 }
 }
 192.168.1.1/26;
}
```

# 設定の書き換え

- rename コマンドにより設定したvariable やエレメントを書き換えることも可能

ge-0/0/0のaddressを192.168.2.1/26へ変更

```
root# rename interfaces ge-0/0/0 unit 0 family inet address 192.168.1.1/26 to address 192.168.2.1/26
```

```
root# show interfaces
ge-0/0/0 {
 unit 0 {
 family inet {
 address
192.168.1.1/26;
ge-0/0/1 {
 unit 0 {
 family inet {
 address
192.168.1.1/26;
```



```
root# show interfaces
ge-0/0/0 {
 unit 0 {
 family inet {
 address
192.168.2.1/26;
ge-0/0/1 {
 unit 0 {
 family inet {
 address
192.168.1.1/26;
```

# 設定の項目の置換

- replace コマンドにより設定内の文字列を置換することも可能

ge-0/0/0のaddressを192.168.2.1/26へ変更

```
root# replace pattern /26 with /24
```

```
root# show interfaces
ge-0/0/0 {
 unit 0 {
 family inet {
 address
192.168.2.1/26;
ge-0/0/1 {
 unit 0 {
 family inet {
 address
192.168.1.1/26;
```



```
root# show interfaces
ge-0/0/0 {
 unit 0 {
 family inet {
 address
192.168.2.1/24;
ge-0/0/1 {
 unit 0 {
 family inet {
 address
192.168.1.1/24;
```

# activate/deactivate

- deactivateコマンドを使うことで、設定の一部を削除することなく無効にすることが可能なので、障害時の切り分けなどに便利

192.168.1.2/24を無効化

```
root# deactivate interfaces ge-0/0/1 unit 0 family inet address 192.168.1.2/24
```

```
root# show interfaces
ge-0/0/1 {
 unit 0 {
 family inet {
 address 192.168.1.1/24;
 address 192.168.1.2/24;
```

```
root# show interfaces
ge-0/0/1 {
 unit 0 {
 family inet {
 address 192.168.1.1/24;
 inactive: address 192.168.1.2/24;
```

192.168.1.2/24の無効化を解除（有効化）

```
root# activate interfaces ge-0/0/1 unit 0 family inet address 192.168.1.2/24
```


# wildcard range set/delete

- wildcard rangeコマンドを使用することで、インターフェイスなど複数の対象に対して同じ設定内容を適用することが簡単に可能

```
root# show interfaces
root#
```

```
root# wildcard range set interfaces ge-0/0/[0-3,5,!2] mtu 9000
```

[0-3, 5, !2] ⇒ 0~3と5、ただし2は除く



```
root# show interfaces
ge-0/0/0 { mtu 9000; }
ge-0/0/1 { mtu 9000; }
ge-0/0/3 { mtu 9000; }
ge-0/0/5 { mtu 9000; }
```


ge-0/0/0-1,3,5のMTU設定が一括で投入されている

# wildcard range set/delete

- 同様にDeleteも可能

```
root# show interfaces
ge-0/0/0 { mtu 9000; }
ge-0/0/1 { mtu 9000; }
ge-0/0/3 { mtu 9000; }
ge-0/0/5 { mtu 9000; }
```

```
root# wildcard range delete interfaces ge-0/0/[0-1] mtu
```



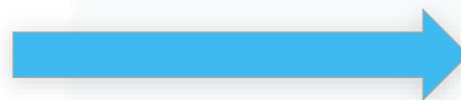
```
root# show interfaces
ge-0/0/3 { mtu 9000; }
ge-0/0/5 { mtu 9000; }
```

# interface-range

- interface-rangeを使用することで、複数のインターフェイスをグループ化して共通の設定を行う事が可能。この設定はwildcardと異なりコンフィグ内に保持される為、一度作成してしまえば様々な設定に対する繰り返しの利用が可能

```
root# show interfaces
root#
```

```
root# set interfaces interface-range CLIENTS member-range ge-0/0/0 to ge-0/0/1
root# set interfaces interface-range CLIENTS member ge-0/0/3
root# set interfaces interface-range CLIENTS mtu 9000
```



CLIENTSというメンバーに入っている、  
ge-0/0/0-1,3のMTUを一括設定

```
root# show interfaces
interface-range CLIENTS {
 member ge-0/0/3;
 member-range ge-0/0/0 to ge-0/0/1;
 mtu 9000;
}
```

# interface-range

- Range内の個別インターフェイス毎に特有の設定を追加することも可能

```
root# show interfaces
interface-range CLIENTS {
 member ge-0/0/3;
 member-range ge-0/0/0 to ge-0/0/1;
 mtu 9000;
}
```

```
root# set interfaces ge-0/0/0 unit 0 family inet address 10.0.0.1/24
```



CLIENTSというメンバー共通でない  
設定をIF単体に設定設定

```
root# show interfaces
interface-range clients {
 member ge-0/0/3;
 member-range ge-0/0/0 to ge-0/0/1;
 mtu 9000;
}
ge-0/0/0 {
 unit 0 {
 family inet {
 address 10.0.0.1/24;
 }
 }
}
```



# 階層間の移動 -1

同じ階層の設定を複数作成する際は階層を移動することで作成する構文を省略することが可能です

例1: FWフィルタの設定(topの階層から設定)

```
show firewall
family inet{
 filter FW-FILTER{
 term BLOCK{
 from{
 source-address{
 10.10.10.0/24;
 }
 destination-address{
 192.168.1.0/24;
 }
 dscp cs5;
 port[https http];
 }
 }
 }
}
```



```
[edit]
set firewall family inet filter FW-FILTER term BLOCK from source-address 10.10.10.0/24
set firewall family inet filter FW-FILTER term BLOCK from destination-address 192.168.1.0/24
set firewall family inet filter FW-FILTER term BLOCK from dscp cs5
set firewall family inet filter FW-FILTER term BLOCK from port https
set firewall family inet filter FW-FILTER term BLOCK from port http
```

※設定を投入する際は繰り返しset firewall family…fromと入力する必要がある

## 階層間の移動 -2

例2: FWフィルタの設定(firewall filter FW-FILTER term BLOCK fromの階層から設定)

```
show firewall
family inet {
 filter FW-FILTER {
 term BLOCK {
 from {
 source-address {
 10.10.10.0/24;
 }
 destination-address {
 192.168.1.0/24;
 }
 dscp cs5;
 port [https http];
 }
 }
 }
}
```

```
[edit firewall family inet filter FW-FILTER term BLOCK from]
set source-address 10.10.10.0/24
set destination-address 192.168.1.0/24
set dscp cs5
set port https
set from port http
```

※設定を投入する際はfirewall family…fromまでを省略して入力することができる

# 階層間の移動 -3

- 階層間は、editコマンドで移動することができます
- exit : 直前にいたレベルに戻ります
  - TOPでEXITを実行すると、Operationalモードに戻ります
  - OperationalモードでEXITを実行すると、システムからLogoutします
    - Shellモードから`cli`でOperationalモードに移動した場合は、Shellモードに戻ります
- up:一つ上のレベルに移動します
- top : 最上位のレベルに移動します

Editで階層を指定



Topで最上位へ

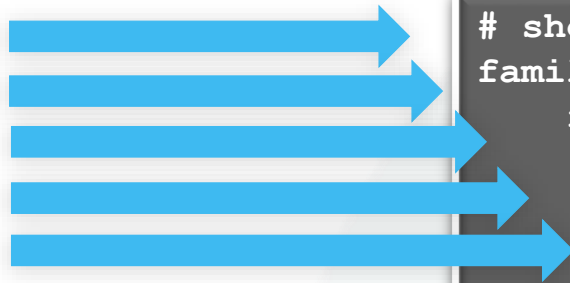


upで一つ上へ



Top

Down



```
show firewall
family inet{
 filter FW-FILTER{
 term BLOCK{
 from{
 source-address{
 10.10.10.0/24;
 }
 destination-address{
 192.168.1.0/24;
 }
 dscp cs5;
 port[https http];
 }
 }
 }
}
```

# Automatic Configuration Archival

- Automatic Configuration Archival機能を使用することで、自動的に最新のコンフィグをリモートのFTP/SCPサーバにバックアップすることが可能
- アップロードのタイミングは、コミットの度もしくは一定時間毎のいずれか、あるいは両方を選択可能

## 1. コミットの度にリモートのサーバにコンフィグをバックアップする設定:

```
user@Junos# set system archival configuration transfer-on-commit
user@Junos# set system archival configuration archive-sites ftp:// loginname:loginpassword@FTP-
server-ip/directory
```

## 2. 一定時間おきにリモートのサーバにコンフィグをバックアップする設定:

(例: 1440分 = 24時間おき)

```
user@Junos# set system archival configuration transfer-interval 1440
user@Junos# set system archival configuration archive-sites ftp:// loginname:loginpassword@FTP-
server-ip/directory
```

# 機器の初期化

Junos機器を初期化する手法は主に以下の3つ

- Configuration modeで load factory-default
  - 実行すると、Candidate Configurationにデフォルトの設定がロードされる
  - 実際に初期設定に戻すには、rootパスワードの設定とCommitが必要となる
  - 設定のみを戻したいときに有効で、ログや過去のConfig(rollback)などは削除されない
- Operation modeで request system zeroize
  - 実行すると、全ての設定やログ、ユーザの作成したファイルが削除され、再起動する
  - システムファイルは削除されない
- USBメモリやCFからのFormat install
  - USBメモリやCFにJunosイメージを書き込み、ブートローダーからJunosを再インストールする
  - システムファイルを含むディスク上の全てのデータが削除され、新たにJunosがインストールされる
  - 実行方法は機種によって異なり、JTACから指示された場合を除き、一般的に使用する必要はない

# コントロールパケットのキャプチャ

以下のコマンドを使用することにより、コントロールパケット(REが受信するパケット)をキャプチャする事が可能

```
root> monitor traffic interface xe-1/2/0.0
verbose output suppressed, use <detail> or <extensive> for full protocol decode
Address resolution is ON. Use <no-resolve> to avoid any reverse lookup delay.
Address resolution timeout is 4s.
Listening on xe-1/2/0.0, capture size 96 bytes

11:39:06.772930 Out IP truncated-ip - 11 bytes missing! 192.168.1.1.bgp > 192.168.1.2.32794: P
635171747:635171766(19) ack 995070346 win 16384 <nop,nop,timestamp 3971359530 2610569>: BGP,
length: 19
11:39:06.803191 In IP 192.168.1.2.32794 > 192.168.1.1.bgp: . ack 19 win 5360 <nop,nop,timestamp
2637232 3971359530>
...
...
```

- このコマンドでキャプチャできるパケットは、PFEで処理されずREで処理されるパケットに限られる
- ICMP Echo(ping)等、PFEによってオフロード処理されるパケットは表示されないので注意
- パケット内容の詳細まで確認したい場合は extensive オプションなどを使用する

# groups/apply-groups

設定の一部をgroupという形で切り出し、apply-groupsで任意の階層に適用する事が可能

- 例: 全てのOSPFインターフェイスのHello-IntervalとDead-Intervalを変更

```
root# show groups
OSPF_COMMON {
 protocols {
 ospf {
 area <*> {
 interface <st*> {
 hello-interval 5;
 dead-interval 20;
 }
 }
 }
 }
}

root# show protocols ospf
apply-groups OSPF_COMMON;
area 0.0.0.0 {
 interface st0.1;
 interface st0.2;
 interface lo0.0 {
 passive;
 }
}
```

インターフェイス名やエリア名、IPアドレス等のユーザが自由入力する値は<\*>とすると全てに適用される

特定のインターフェイスのみに適用したい場合などは、<st\*> といったように一部の文字列を指定することも可能

自動的に共通設定が適用される



```
show protocols ospf | display inheritance
area 0.0.0.0 {
 interface st0.1 {
 ##
 ## '5' was inherited from group 'OSPF_COMMON'
 ##
 hello-interval 5;
 ##
 ## '20' was inherited from group 'OSPF_COMMON'
 ##
 dead-interval 20;
 }
 interface st0.2 {
 ##
 ## '5' was inherited from group 'OSPF_COMMON'
 ##
 hello-interval 5;
 ##
 ## '20' was inherited from group 'OSPF_COMMON'
 ##
 dead-interval 20;
 }
 interface lo0.0 {
 passive;
 }
}
```

※CommitしてもConfigはきちんとグループ化されたままとなる  
実際に適用される設定を確認したい場合は、 show configuration | display inheritance コマンドを使用する

# Prefix-list / apply-path

設定に含まれるIPアドレスから自動的にリストを生成し、Firewall Filterに適用することが可能

```
root# show protocols bgp
group GROUP-A {
 neighbor 1.1.1.1;
 neighbor 2.2.2.2;
}

root# show interfaces
ge-0/0/0 { unit 0 { family inet {
 address 1.1.1.0/30;
} } }
ge-0/0/1 { unit 0 { family inet {
 address 2.2.2.0/30;
} } }
fxp0 { unit 0 { family inet {
 address 192.168.1.10/24;
} } }

root# show policy-options
prefix-list BGP-PEERS {
 apply-path "protocols bgp group <*> neighbor <*>";
}
prefix-list LOCALNETS {
 apply-path "interfaces <ge-*> unit <*> family inet
address <*>";
}
```

IPアドレスが  
自動的コピーされる



```
root# show policy-options | display inheritance
prefix-list BGP-PEERS {
 ##
 ## apply-path was expanded to:
 ## 1.1.1.1/32;
 ## 2.2.2.2/32;
 ##
 apply-path "protocols bgp group <*> neighbor
<*>";
}
prefix-list LOCALNETS {
 ##
 ## apply-path was expanded to:
 ## 1.1.1.0/30;
 ## 2.2.2.0/30;
 ##
 apply-path "interfaces <ge-*> unit <*>
family inet address <*>";
}
```

※実際に適用される設定を確認したい場合は、 show configuration | display inheritance コマンドを使用する



# オンライン・マニュアル

- 豊富な機能の help コマンド
  - help topic : プロトコルや機能の一般的な説明を表示
  - help reference : プロトコルや機能の設定方法を表示 (コマンド・レファレンス)
  - help syslog : syslog メッセージの説明

```
mike@juniper1> help topic interfaces address
```

```
Configuring the Interface Address
```

```
You assign an address to an interface by specifying the address when configuring the protocol family. For the inet family, you configure the interface's IP address. For the iso family, you configure one or more addresses for the loopback interface. For the ccc, tcc, mpls, tnp, and vpls families, you never configure an address.b
```

# JUNOS : help topic

コマンドの概要を確認することが可能

```
user@host> help topic ospf dead-interval
```

## Modifying the Router Dead Interval

If a router does not receive a hello packet from a neighbor within a fixed amount of time, the router modifies its topological database to indicate that the neighbor is nonoperational. The time that the router waits is called the router dead interval. By default, this interval is 40 seconds (four times the default hello interval).

To modify the router dead interval, include the dead-interval statement.

This interval must be the same for all routers on a shared network.

```
dead-interval seconds;
```

For a list of hierarchy levels at which you can include this statement, see the statement summary section for this statement.

# JUNOS : help reference

- コマンドのオンラインマニュアルを参照することが可能

```
user@host> help reference oam action
 action (OAM)

Syntax
action {
 syslog (OAM Action);
 link-down;
 send-critical-event;
}

Hierarchy Level
[edit protocols oam ethernet link-fault-management action-profile]

Release Information
Statement introduced in JUNOS Release 8.5.

...

Description
Define the action or actions to be taken when the OAM fault event occurs.

Usage Guidelines
See Specifying the Actions to Be Taken for Link-Fault Management Events.
```

# JUNOS : help apropos

- 確実に覚えていないコマンド（うろ覚えの場合など）を文字列で検索することが可能

```
user@host# help apropos vstp
set logical-systems <name> protocols vstp
 VLAN Spanning Tree Protocol options
set logical-systems <name> protocols vstp disable
 Disable VSTP
set protocols vstp
 VLAN Spanning Tree Protocol options
set protocols vstp disable
 Disable VSTP
```

Configuration mode

```
user@host# > help apropos vstp
help topic stp vstp
 VLAN Spanning Tree Protocol instance configuration
help topic stp vstp-requirements
 Requirements, limitations for VLAN Spanning Tree Protocol
help reference stp vstp
 VLAN Spanning Tree Protocol configuration
help reference stp vlan-vstp
 VLAN configuration for VLAN Spanning Tree Protocol
```

Operation mode

# CLI : Trace / 充実したdebug機能

例: OSPF Trace-option

注目したいパケットタイプを細かく指定することが可能

- JUNOSでは、プロトコル別にTrace-optionsを非常に細かく設定可能です。
- このTraceの出力先はファイル出力、あるいはmonitorコマンドでReal-timeに画面にてモニタ表示
- トラブルシューティングに役立つ情報を的確に抜き出すことができます

```
lab@Router# set protocols ospf traceoptions flag ?
Possible completions:
 all Trace everything
 database-description Trace database description packets
 error Trace errored packets
 event Trace OSPF state machine events
 flooding Trace LSA flooding
 general Trace general events
 hello Trace hello packets
 lsa-ack Trace LSA acknowledgement packets
 lsa-request Trace LSA request packets
 lsa-update Trace LSA update packets
 normal Trace normal events
 packet-dump Dump the contents of selected packet types
 packets Trace all OSPF packets
 policy Trace policy processing
 route Trace routing information
 spf Trace SPF calculations
 state Trace state transitions
 task Trace routing protocol task processing
 timer Trace routing protocol timer processing
```

# CLI : monitor / リアルタイムにトラフィックを監視

- monitorコマンドで現在のI/F別トラフィック状況を見ることが出来ます
- 表示はAUTOリフレッシュされるため、継続的なモニタリングが可能
- トラフィックの傾向や障害箇所の特定に役立ちます

```
10.0b2 Seconds: 13 Time: 14:50:48
Interface Link Input packets (pps) Output packets (pps)
ge-0/0/0 Up 54175 (4) 4126 (0)
ge-0/0/1 Down 399 (0) 37 (0)
ge-0/0/2 Up 5110 (1) 4224 (0)
ge-0/0/3 Down 0 (0) 0 (0)
ge-0/0/4 Down 0 (0) 0 (0)
ge-0/0/5 Down 0 (0) 0 (0)
ge-0/0/6 Down 0 (0) 0 (0)

Bytes=b, Clear=c, Delta=d, Packets=p, Quit=q or ESC, Rate=r, Up=^U, Down=^D
```

# rescue configuration

- 基本となるconfigurationを予め定義(保存)することが可能

保存方法: > `request system configuration rescue save`

削除方法: > `request system configuration rescue delete`

- Rescue configurationの反映方法
  - Rollbackコマンドからのロード

# rollback rescue

```
root# rollback rescue
load complete
root# commit
```

- ハードウェアからのロード

- SRXシリーズはRESET CONFIGボタンを押すことでハードウェアからロードすることができます。

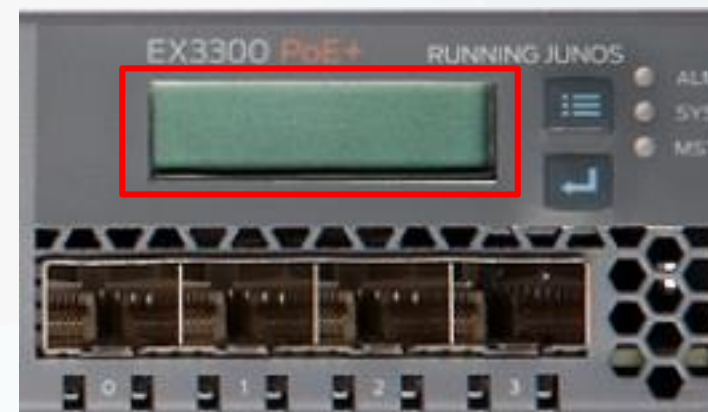
※15秒以上押し続けるとfactory defaultがロードされます

例:  
SRX300



- EXシリーズはLCDパネルでメンテナンスモードを操作することでハードウェアからロードすることができます。

例:  
EX3300





まとめ



# CW4S SSG/NetScreen移行支援ツール

- SRXをSSG/NetScreenのようにGUI設定操作

ツールダウンロードサイトURL  
<http://cw4s.org>

1.ソフトウェアをダウンロードして  
PCにインストール

Classic GUI for SRX

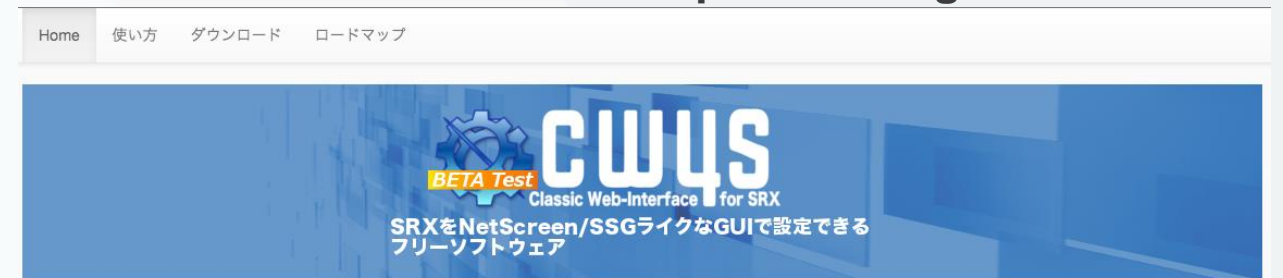


2.ScreenOSと同等のGUIから  
ファイアウォールの設定を行って、  
最後に“Commit”

SRX series

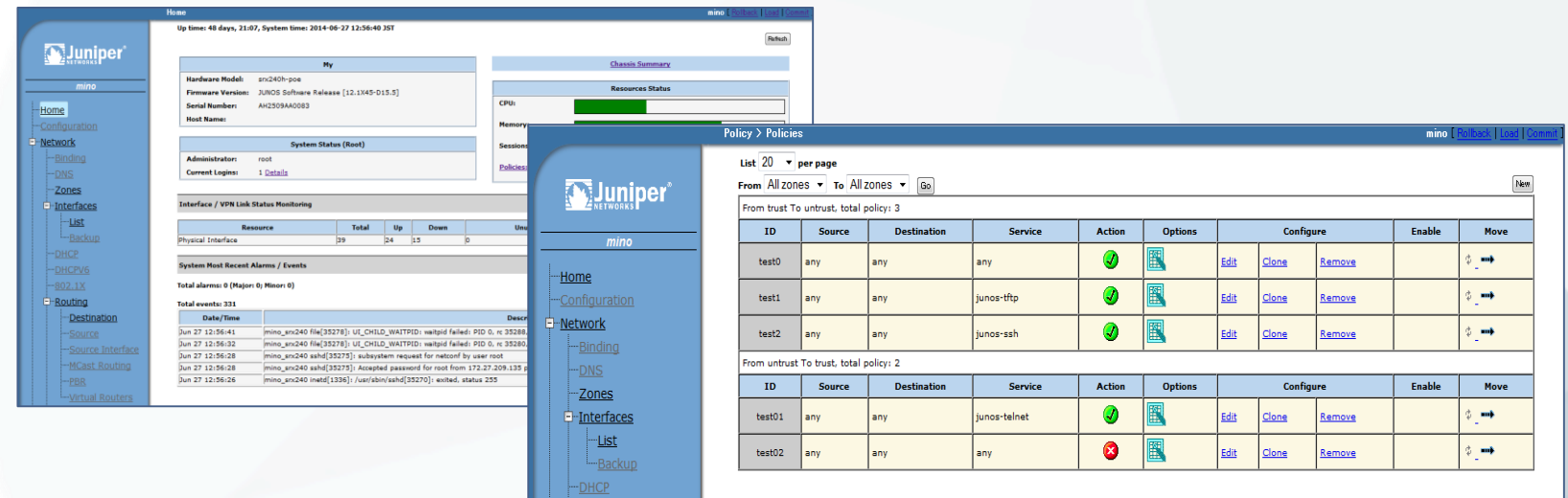


3.NetConfで  
SRXに設定が投入される



## ようこそ！

CW4Sは、ジュニパーネットワークス社のSRXシリーズを、  
**Web ブラウザ** から、**SSG/NSシリーズ** 感覚で設定できるフリーソフトウェアです。  
お手元のPCとSRXシリーズをsshで接続し、Webインターフェースを提供します。  
現在、正式版リリースとソースコード公開に向けて準備中です。  
利用許諾をご確認の上、ご利用ください。



# vSRX on your laptop ~PCで始めるvSRX~

- 仮想Router/FirewallであるvSRXをLaptop PC上で動作させるための指南書

<http://www.slideshare.net/JuniperJapan/vsrx-laptop-201505>



実際のデバイスと同様の設定作成や仕様確認をPC上で実施することが可能！



# Appendix



# Appendix A

## Chassis Cluster Deep Dive

# CLUSTERとNODE ID

- Cluster ID
  - シャーシ間でクラスタリングの設定をする際に、Cluster IDが必要になります
  - Cluster IDは、1から255まで、割り振ることができます。注意点としては、同じレイヤ2ブロードキャストセグメントで他のCluster IDと重複しないようにしなければなりません
- Node ID
  - Cluster内で各々のメンバーは、Node ID(0または1)により識別されます
  - 現在サポートされているノード数は、最大2台です
  - Node IDとCluster IDは、EPROMに、保存されます
  - コンフィギュレーションを初期設定に戻しても、ClusterのDisableを実施しないとClusterは解除されません

# ノード独自(固有)のコンフィグ

- ノード固有のコンフィグ
  - JUNOSでは、両機器に、同じコンフィグレーションを保持しつづけます  
従ってコンフィグは原、Primary側で実施します
  - コンフィグの独自区分は、ノード番号(EEPROMに保存)により示されます
  - どのノードがどのグループ所属するなどを定義するためには、  
JUNOSグループ機能を利用します
  - ノード固有のコンフィグには以下が含まれます
    - fxp0のコンフィグ: マネージメントポート
    - システム名(ホストネーム)
    - バックアップルータIPアドレス

# コントロールポート (コントロールリンク)

- コントロールポート (コントロールリンク)
  - コントロールポートは、RE間のコミュニケーションを許可します
  - Clusterメンバー間で、JSRP、Chassisd、カーネルの情報を共有します
  - 現在、各々の機器に割り当てることができるコントロールポートは、ひとつだけです。(fxp1)が割り当てられます
  - SRXブランチシリーズは、コントロールポートが自動的に割り振られるため、コンフィグをする必要がありません

# ファブリックポート (ファブリックリンク)

- ファブリックポート (ファブリックリンク)
  - データプレーンを直接つなぐファブリックポートです
  - Clusterメンバー間で、同一のデータプレーンを接続します。
  - Cluster全体でサポートされているファブリックリンクは、最大2リンク
  - SRX HAにて、RTOメッセージは、ファブリックリンク (セッション、ルートなど) を介して同期します。Active/Active構成では、データは、メンバー間のファブリックポートを介して(Z型)通信できます
  - 非対称のデータ(ユーザー)トラフィックもサポートします
- ファブリックポート(ファブリックリンク) コンポーネント
  - fab0とfab1の仮想インタフェースは、node0とnode1をつなぐために、作成する必要があります
    - node0側にfab0インタフェースを作成し、node1側にfab1インタフェースを作成し、直接結線することを推奨しています

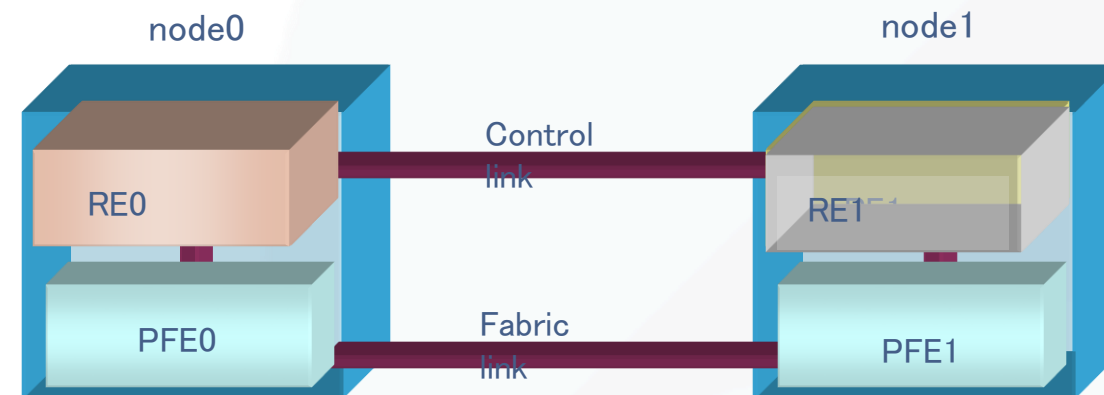


# コントロールポートとファブリックポートの注意事項

- コントロールポートとファブリックポートにスイッチを挟む場合
  - コントロールリンクとファブリックリンクのVLANは分けてください
  - 遅延は、100msec以下にしてください
  - IGMP Snooping機能は、無効にしてください
  - コントロールリンクとファブリックリンクのVLANに他のトラフィックを流さないでください
  - トラフィックを、カプセリングする際は、MTUのサイズに注意してください
    - パケットのフラグメントをサポートしていません

# Redundancy Group

- Redundancy Group
  - コンポーネントをグループ化し、シャーシ間をフェイルオーバーします
  - Redundancy group 0は、ルーティングエンジンとして使われます
  - Redundancy group 1は、Active/PassiveのRedundant interfaceとして使われます。Redundancy Group 1以上は、Active/Activeの時に使われます
  - オペレーションは、ScreenOSのVSDに非常によく似ています。JUNOSでは、コントロールプレーンとデータプレーンを分けるために、少なくともふたつのRedundancy groupが必要となります。Redundancy Group 0は、コントロールプレーン冗長の為に、Redundancy Group 0にマッピングされ、Redundancy Group 1以上は、データプレーンにマッピングされます。

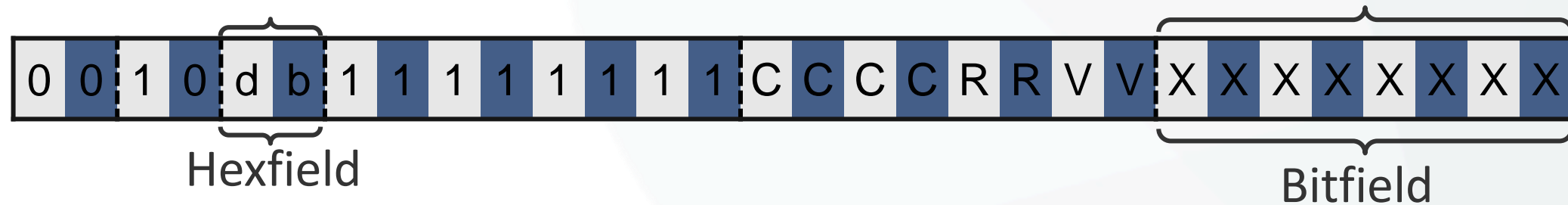


# Redundant Ethernet Interface

- Redundant Interface
  - Redundant Interfaceは、Active/Passiveとしての役割を持つメンバーインタフェースを構成する仮想インタフェースです。
    - SRXのActive/Activeとは、各々のRedundant EthernetメンバーがActive/Activeになるわけではなく、異なるRedundancy Groupを利用して、同時にトラフィックを転送できる構成または、状態を示します。(それぞれのRedundancy GroupのMasterをイレコにする)
  - シャーシ跨ぎのトラフィックの概念を除いてScreenOSとRedundant Interfaceの考え方は同じです
  - コンフィグでは、`reth<番号X>`とします。すべてのロジカルコンフィグは、このインタフェースにする必要があります。物理インタフェースとは、異なります。例えば、IPアドレス、QoS、Zone、VPNなどの設定がそれにあたります。物理プロパティだけは、メンバーインタフェースに適応されます。
- Redundant Interfaceの作成
  - リンクアグリゲーションインタフェースを作成するように、作成することができます。SRXが仮想インタフェースを作成するために、シャーシ内で`re t h`番号を割り振らなければいけません
  - `reth interface`を作成したら、`reth interface`をRedundancy Groupにバインドする必要があります

# Redundant Interface MACアドレス

- Cluster IDを利用して、RETH MACアドレスは提供されます
  - reth MACアドレスの構成



- 構成要素:
  - CCCC - cluster id、ユーザにより割り振られたID番号
  - RR - reserved. 00.
  - VV - version、ファーストリリースは、00
  - XXXXXXXX - Interface id、reth indexから決定される



- Cluster id 1、reth interface 0のMACアドレスのフォーマット例：

# インタフェース モニタリング

- インタフェース モニタリング
  - Cluster内のリンクダウンやインタフェースのリアクションのモニター機能
  - ScreenOSのように、閾値(255)からウェイトの値にて減算利用し、シャーシ内でのフェイルオーバーを実現
  - リモートの障害とフェイルオーバーを関連付けるためには、JUNOS11.2以降でサポートされているIP Monitoringの機能が必要

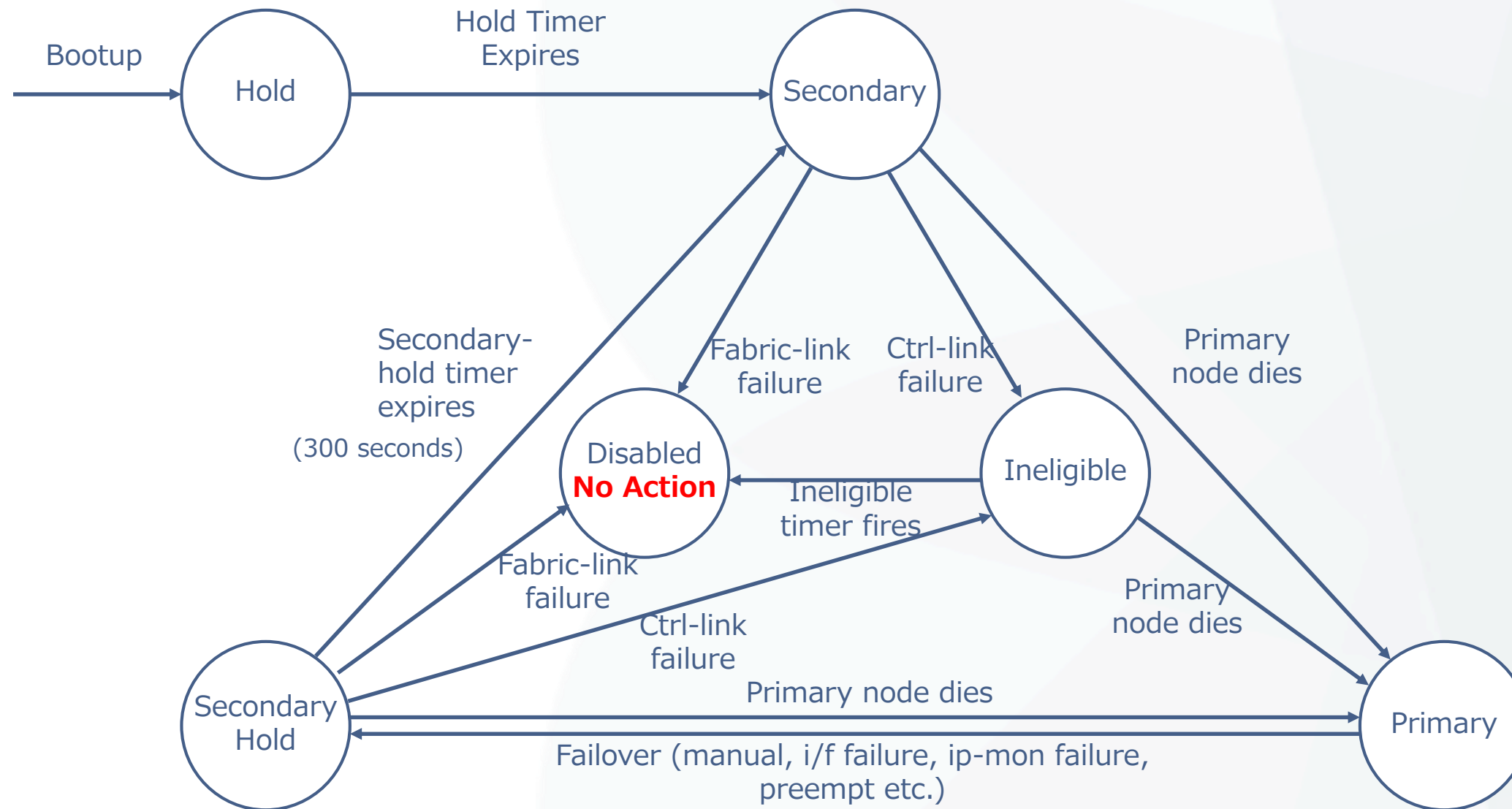
# コントロールリンクモニタリング

- コントロールリンクモニタリング
  - コントロールリンクは、特に設定を加えることなく常にモニターされています。然しながら、コントロールリンクリカバリー機能は、デフォルトでは設定されていません。この設定は、セカンダリーノードが復旧した際に、自動でコントロールリンクを復旧させる機能。30回のハートビート(デフォルトでは、60秒)により正常性が確認できた後、セカンダリーノードをリブートさせる。コントロールリンクがダウンした時、セカンダリーノードは、Disableのステータスになり、両方のノードが分離し別々に機能するのを防ぎます
    - コマンド : `set chassis cluster control-link-recovery`
  - コントロールリンクがダウンした時、コントロールリンクを復旧させるには、コントロールリンクリカバリーの機能を利用するか、手動でセカンダリーノードをリブートするかのいずれかの方法を選択できます

# ファブリックリンクモニタリング

- ファブリックリンクモニタリング
  - ファブリックリンクは、特に設定を加えることなく常にモニターされています。。 JUNOS10.4r4以降では、ファブリックリンクダウン発生から復旧時、リブートすることなく、モニタリングは再開されます。
  - ファブリックリンクは、最大2本まで冗長化することができます。2本有効時、1本は、RTOで利用し、残りの1本は、実データを流すリンクとして利用します

# SRX HA ステータス遷移



- Disableステータスになるのは、セカンダリーノードのみです。
- Disableステータスを復旧させるには、セカンダリーノードのリブートが必要です。
- 赤文字の「No Action」は、JUNOS 10.4r4以降の動作になります。



# シャーシクラスタの無効化

- シャーシクラスタを無効化する場合
  - EPROMに書き込まれている内容をリセットする必要がある
  - 以下どちらかの手順で無効化（どちらも同じ効果）
    - Chassis clusterをdisableにしてreboot

```
user@srx> set chassis cluster disable reboot
```

- または、Cluster IDを0に設定してreboot

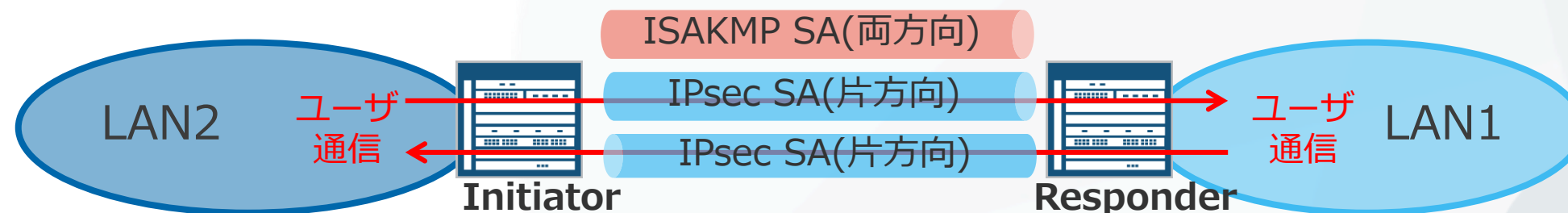
```
user@srx> set chassis cluster-id 0 node 0 reboot
```



# Appendix B IPSec VPNの設定

# IPsec VPNとは

- IPsec
  - 暗号技術を用いてIPパケット単位で改竄防止や秘匿機能を提供するプロトコル
  - セキュリティゲートウェイ間でSA(Security Association)を作成
    - ユーザトラフィックはSA内を通過
- IKE
  - 暗号/認証アルゴリズムの決定、暗号鍵交換のために利用されるプロトコル
  - 2つのフェーズでSAを確立
    - IKEフェーズ1：ISAKMP SA(双方向)を生成
    - IKEフェーズ2：ユーザ通信が通過するためのIPSec SA(片方向 x2)を生成
  - IKE折衝の開始側をInitiator、応答側をResponderと呼びます

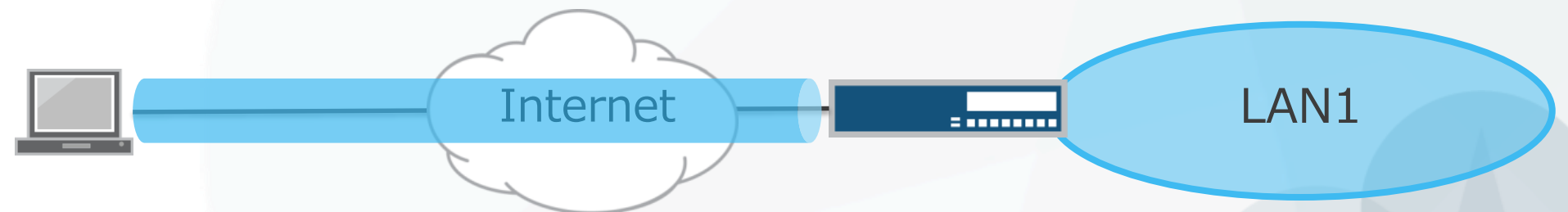


# VPN接続形態

- VPN接続には大きく分けて下記の2通りになります
  - LAN間接続
    - 離れた拠点間のLANセグメント同士をVPN接続



- リモートユーザ接続
  - セキュリティゲートウェイとユーザ端末間でVPN接続
  - 端末側にVPNクライアントとなるソフトウェアが必要



# LAN間接続 IPsec VPNの設定方法

- SRXのLAN間接続 VPNは、以下の2つの設定方法があります
  - ルートベースVPN
    - ルーティングにマッチする全トラフィックをトンネリング
  - ポリシーベースVPN
    - ポリシーにマッチするトラフィックのみをトンネリング

# LAN間接続 IPsec VPN 設定の手順

- LAN間接続 IPsec VPNの設定は以下のステップで行います
  1. フェーズ1 パラメータの設定
    - a. プロポーザルの設定
    - b. ポリシーの設定
    - c. ゲートウェイの設定
  2. フェーズ2 パラメータの設定
    - a. プロポーザルの設定
    - b. ポリシーの設定
    - c. VPNの設定
- ルートベースVPNの場合
  - トンネルインタフェースの作成とゾーンの割り当て
  - ルーティングの設定
  - VPNへのバインディング
- ポリシーベースVPNの場合
  - トンネリングポリシーの作成

# 1-a. フェーズ1プロポーザルの設定

- ISAKMP SAのセキュリティ属性（プロポーザル）を定義
  - 認証方式、鍵交換方式(DH group)、暗号化アルゴリズム、認証アルゴリズム等を指定

```
set security ike proposal IKE_PROPOSAL1 authentication-method pre-shared-keys
set security ike proposal IKE_PROPOSAL1 dh-group group2
set security ike proposal IKE_PROPOSAL1 authentication-algorithm sha1
set security ike proposal IKE_PROPOSAL1 encryption-algorithm aes-128-cbc
```

- パラメータの組合せが予め定義されており、こちらを利用することも可能

| セット名       | 定義内容                                                                                                                                                                                     | 表記                                                                     |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------|
| Basic      | Proposal 1: Preshared key, DH g1, DES, SHA1<br>Proposal 2: Preshared key, DH g1, DES, MD5                                                                                                | pre-g1-des-sha<br>pre-g1-des-md5                                       |
| Compatible | Proposal 1: Preshared key, DH g2, 3DES, SHA1<br>Proposal 2: Preshared key, DH g2, 3DES, MD5<br>Proposal 3: Preshared key, DH g2, DES, SHA1<br>Proposal 4: Preshared key, DH g2, DES, MD5 | pre-g2-3des-sha<br>pre-g2-3des-md5<br>pre-g2-des-sha<br>pre-g2-des-md5 |
| Standard   | Proposal 1: Preshared key, DH g2, 3DES, SHA1<br>Proposal 2: Preshared key, DH g2, AES128, SHA1                                                                                           | pre-g2-3des-sha<br>pre-g2-aes128-sha                                   |

# 1-b, 1-c. フェーズ1ポリシー、ゲートウェイの設定



- IKEポリシーの設定
  - 設定したプロポーザルを適用

```
set security ike policy IKE_POLICY1 proposals IKE_PROPOSAL1
set security ike policy IKE_POLICY1 pre-shared-key ascii-text juniper
```

- IKEゲートウェイの設定
  - IKEポリシー、対向のアドレスとインターフェースを指定

```
set security ike gateway GW1 ike-policy IKE_POLICY1
set security ike gateway GW1 address 10.0.1.1
set security ike gateway GW1 external-interface ge-0/0/0
```



## 2-a. フェーズ2プロポーザルの設定

- IPsec SAのセキュリティ属性（プロポーザル）を定義
  - プロトコル、暗号化アルゴリズム、認証アルゴリズム等を設定

```
set security ipsec proposal IPSEC_PROPOSAL1 protocol esp
set security ipsec proposal IPSEC_PROPOSAL1 authentication-algorithm hmac-sha1-96
set security ipsec proposal IPSEC_PROPOSAL1 encryption-algorithm aes-128-cbc
```

- パラメータの組合せが予め定義されており、こちらを利用することも可能

| セット名       | 定義内容                                                                                                                                                 | 表記                                                                                 |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| Basic      | Proposal 1: no PFS, ESP, DES, SHA1<br>Proposal 2: no PFS, ESP, DES, MD5                                                                              | nopfs-esp-des-sha<br>nopfs-esp-des-md5                                             |
| Compatible | Proposal 1: no PFS, ESP, 3DES, SHA1<br>Proposal 2: no PFS, ESP, 3DES, MD5<br>Proposal 3: no PFS, ESP, DES, SHA1<br>Proposal 4: no PFS, ESP, DES, MD5 | nopfs-esp-3des-sha<br>nopfs-esp-3des-md5<br>nopfs-esp-des-sha<br>nopfs-esp-des-md5 |
| Standard   | Proposal 1: DH g2, ESP, 3DES, SHA1<br>Proposal 2: DH g2, ESP, AES128, SHA1                                                                           | g2-esp-3des-sha<br>g2-esp-aes128-sha                                               |

## 2-b, 2-c. フェーズ2ポリシーの設定、VPNの設定



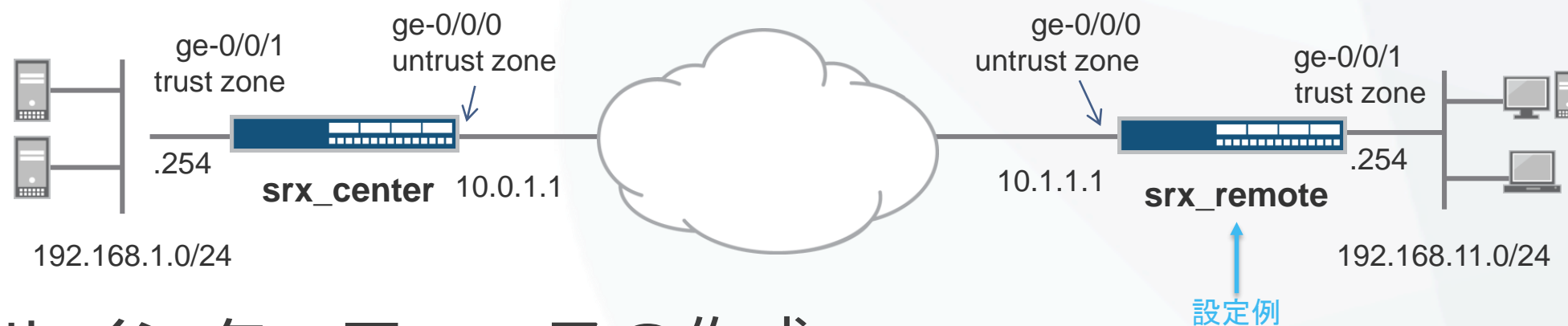
- IPsecポリシーの設定
  - 設定したプロポーザルを適用

```
set security ipsec policy IPSEC_POLICY1 proposals IPSEC_PROPOSAL1
```

- IPsec VPNの設定
  - 設定済みのゲートウェイ、IPsecポリシーを適用

```
set security ipsec vpn VPN1 ike gateway GW1
set security ipsec vpn VPN1 ike ipsec-policy IPSEC_POLICY1
set security ipsec vpn VPN1 establish-tunnels immediately
```

### 3. ルートベースVPNの設定



- トンネルインターフェースの作成

```
set interfaces st0 unit 0 family inet
```

- ルーティングの設定

```
set routing-options static route 192.168.1.0/24 next-hop st0.0
```

- IPsec VPNとのひもづけ

```
set security ipsec vpn VPN1 bind-interface st0.0
```

- Security Zoneにアサイン

```
set security zones security-zone untrust interfaces st0.0
```

## 4. ポリシーベースVPNの設定

- アドレスブックの作成

```
set security zones security-zone trust address-book address Local-LAN 192.168.11.0/24
set security zones security-zone untrust address-book address Remote-LAN 192.168.1.0/24
```

- アクションが “Tunnel” のセキュリティポリシーを作成

- trust -> untrust

```
set security policies from-zone trust to-zone untrust policy 100 match source-address Local-LAN
set security policies from-zone trust to-zone untrust policy 100 match destination-address Remote-LAN
set security policies from-zone trust to-zone untrust policy 100 match application any
set security policies from-zone trust to-zone untrust policy 100 then permit tunnel ipsec-vpn VPN1
```

- untrust -> trust

```
set security policies from-zone untrust to-zone trust policy 200 match source-address Remote-LAN
set security policies from-zone untrust to-zone trust policy 200 match destination-address Local-LAN
set security policies from-zone untrust to-zone trust policy 200 match application any
set security policies from-zone untrust to-zone trust policy 200 then permit tunnel ipsec-vpn VPN1
```

※注意：ポリシーベースVPNとルートベースVPNの混在構成(設定)は不可

# 接続確認 – ISAKMP SA(フェーズ1)の確認

```
user@SRX> show security ike security-associations
```

| Index   | State | Initiator cookie | Responder cookie | Mode | Remote Address |
|---------|-------|------------------|------------------|------|----------------|
| 6706971 | UP    | 845863c590392820 | 8ebfcc763b60a0de | Main | 10.0.1.1       |

```
user@SRX> show security ike security-associations detail
```

```
IKE peer 10.0.1.1, Index 6706971, Gateway Name: GW1
```

```
Role: Responder, State: UP
```

```
Initiator cookie: 845863c590392820, Responder cookie: 8ebfcc763b60a0de
```

```
Exchange type: Main, Authentication method: Pre-shared-keys
```

```
Local: 10.1.1.1:500, Remote: 10.0.1.1:500
```

```
Lifetime: Expires in 25619 seconds
```

```
Peer ike-id: 10.0.1.1
```

```
Xauth user-name: not available
```

```
Xauth assigned IP: 0.0.0.0
```

```
Algorithms:
```

```
Authentication : hmac-sha1-96
```

```
Encryption : aes128-cbc
```

```
Pseudo random function: hmac-sha1
```

```
Diffie-Hellman group : DH-group-2
```

```
Traffic statistics:
```

```
Input bytes : 1148
```

```
Output bytes : 808
```

```
Input packets : 8
```

```
Output packets : 5
```

```
IPSec security associations: 2 created, 1 deleted
```

```
Phase 2 negotiations in progress: 1
```

```
~~~~~
```

State:UPにならないと接続できていない  
設定が対向側と同じになっているかを再チェック

# 接続確認 – IPsec SA(フェーズ2)の確認

```
user@SRX> show security ipsec security-associations
```

```
Total active tunnels: 1
```

| ID      | Algorithm            | SPI      | Life:sec/kb | Mon   | lsys | Port     | Gateway  |
|---------|----------------------|----------|-------------|-------|------|----------|----------|
| <131073 | ESP:aes-cbc-128/sha1 | 7e4cac0d | 2091/       | unlim | -    | root 500 | 10.0.1.1 |
| >131073 | ESP:aes-cbc-128/sha1 | edfd7a93 | 2091/       | unlim | -    | root 500 | 10.0.1.1 |

IPSec SAは片方向なので  
Inbound/outboundの両方が  
作成される

```
user@SRX> show security ipsec security-associations detail
```

```
ID: 131073 Virtual-system: root, VPN Name: VPN1
```

```
Local Gateway: 10.1.1.1, Remote Gateway: 10.0.1.1
```

```
Local Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
```

```
Remote Identity: ipv4_subnet(any:0,[0..7]=0.0.0.0/0)
```

```
Version: IKEv1
```

```
DF-bit: clear, Copy-Outer-DSCP Disabled, Bind-interface: st0.0
```

```
Port: 500, Nego#: 2, Fail#: 0, Def-Del#: 0 Flag: 0x600a29
```

```
Tunnel events:
```

```
Sat Jul 16 2016 07:22:19: IPsec SA negotiation successfully completed (2 times)
```

```
Sat Jul 16 2016 06:32:41: IKE SA negotiation successfully completed (1 times)
```

```
Sat Jul 16 2016
```

```
: Negotiation failed with error code NO_PROPOSAL_CHOSEN received from peer (2 times)
```

```
Sat Jul 16 2016
```

```
: Tunnel is ready. Waiting for trigger event or peer to trigger negotiation (1 times)
```

```
Sat Jul 16 2016 06:31:46: External interface's address received. Information updated (1 times)
```

```
Sat Jul 16 2016 06:31:46: External interface's zone received. Information updated (1 times)
```

```
~~~~~
```

# 接続確認 - 暗号/復号トラフィックの統計確認

- IPsec SA上での暗号化/復号化したバイト数、パケット数を表示

```
root@vSRX1> show security ipsec statistics
```

```
ESP Statistics:
```

|                    |       |
|--------------------|-------|
| Encrypted bytes:   | 75696 |
| Decrypted bytes:   | 5208  |
| Encrypted packets: | 498   |
| Decrypted packets: | 62    |

```
AH Statistics:
```

|                 |   |
|-----------------|---|
| Input bytes:    | 0 |
| Output bytes:   | 0 |
| Input packets:  | 0 |
| Output packets: | 0 |

```
Errors:
```

```
AH authentication failures: 0, Replay errors: 0
ESP authentication failures: 0, ESP decryption failures: 0
Bad headers: 0, Bad trailers: 0
```

# IPSec VPN トラブルシューティング

- IKE のdebugログは、 /var/log/kmd 内に保存
  - debug用設定

```
set security ike traceoptions flag all
set security ike traceoptions flag ike
```

- debugログ(kmdファイル)の参照方法

```
user@SRX> show log kmd
```

- IKE debugログをリアルタイムにモニターする場合

```
user@SRX> monitor start kmd
user@SRX> monitor stop
```

- <http://kb.juniper.net/KB10100>
  - How to troubleshoot a VPN tunnel that is down or not active



# IPSec使用時の考慮点

- トンネルインタフェース (st0) のMTU値はデフォルトで9192 です。ScreenOSとRoute-based VPNを使用して接続する場合に問題となる場合があるので注意が必要です。
- 以下の機能はサポートされておられません。
  - Tunnel Interface(st0.x) でのQoS機能



# Appendix C

## NAT pool options

# アドレスプール設定補足

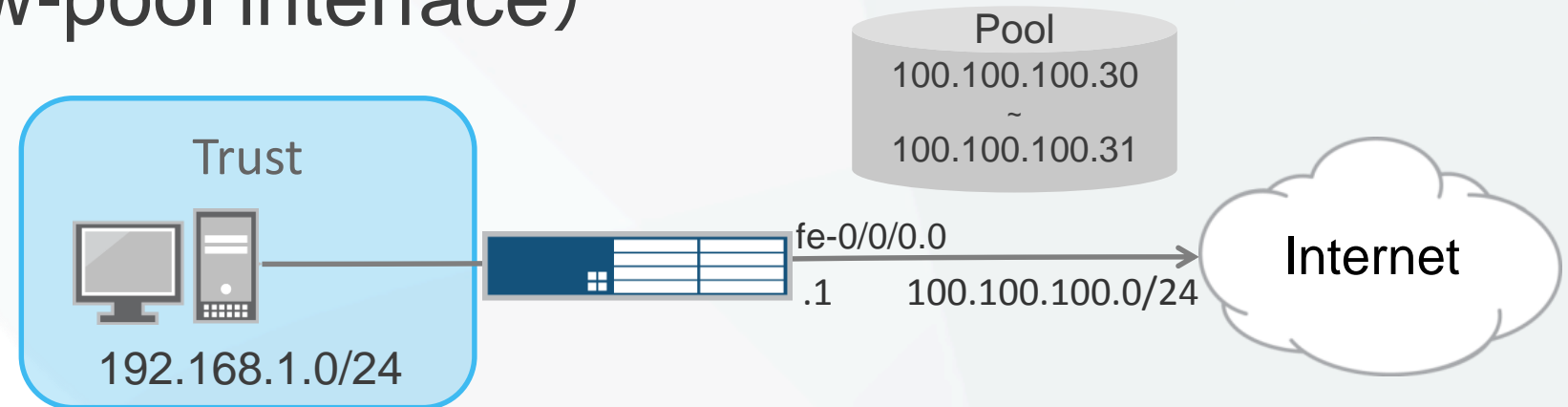
- アドレスプールの形態
  - 単一のIPアドレス
  - IPアドレスレンジ
  - インターフェース (source NATのみ)
- オプション
  - ポート変換オフ (port no-translation)
  - Overflow pools
    - プールのアドレスを使い切った場合のフォールバック用に設定
      - インターフェースアドレスを使用
      - 別のプールを参照
    - ポート変換なしのpoolが必要
  - アドレスシフト

## Source NAT poolの設定例

```
lab@srx# show security nat
source {
 pool src_nat_pool_napt {
 address {
 100.100.100.20/32 to 100.100.100.29/32;
 }
 port {
 no-translation;
 }
 overflow-pool interface;
 }
}
```

# アドレスプール動作の確認

- PAT動作の無効化 (port no-translation)
- プール超過時NAPT (overflow-pool interface)



```
lab@srx> show security flow session
```

```
Session ID: 11120, Policy name: trust-to-untrust/4, Timeout: 1580, Valid
```

```
In: 192.168.1.22/21003 --> 100.100.100.254/23;tcp, If: vlan.0, Pkts: 36, Bytes: 1481
```

```
Out: 100.100.100.254/23 --> 100.100.100.30/21003;tcp, If: fe-0/0/0.0, Pkts: 36, Bytes: 1523
```

```
Session ID: 11127, Policy name: trust-to-untrust/4, Timeout: 1790, Valid
```

```
In: 192.168.1.23/1267 --> 100.100.100.254/22;tcp, If: vlan.0, Pkts: 18, Bytes: 1767
```

```
Out: 100.100.100.254/22 --> 100.100.100.31/1267;tcp, If: fe-0/0/0.0, Pkts: 18, Bytes: 1767
```

```
Session ID: 11159, Policy name: trust-to-untrust/4, Timeout: 1794, Valid
```

```
In: 192.168.1.24/1044 --> 100.100.100.254/80;tcp, If: vlan.0, Pkts: 22, Bytes: 1680
```

```
Out: 100.100.100.254/80 --> 100.100.100.1/64506;tcp, If: fe-0/0/0.0, Pkts: 43, Bytes: 40039
```

NATされているが  
ポート変換されていない

Poolを超過したため  
IFアドレスでNAPTされている

# Source NAT with address-shifting

- NAT動作時にprivate:publicが1:1でマッピングされる
  - Host-address-baseで基点になるprivateアドレスを設定

```
set security nat source pool A address 192.168.1.1/32 to 192.168.1.20/32
set security nat source pool A host-address-base 10.1.1.5/24
```

- show security nat source pool allコマンドで確認
  - 10.1.1.5~25が192.168.1.1~20と1:1で対応

```
root> show security nat source pool all
node0:

Total pools: 1
Pool name : A
Pool id : 4
Routing instance : default
Host address base : 10.1.1.5
Port : no translation
Port overloading : 0
Address assignment : static-paired
Total addresses : 20
Translation hits : 0
Address range Single Ports Twin Ports
192.168.1.1 - 192.168.1.20 0 0
```



# Appendix D

## Security Logging

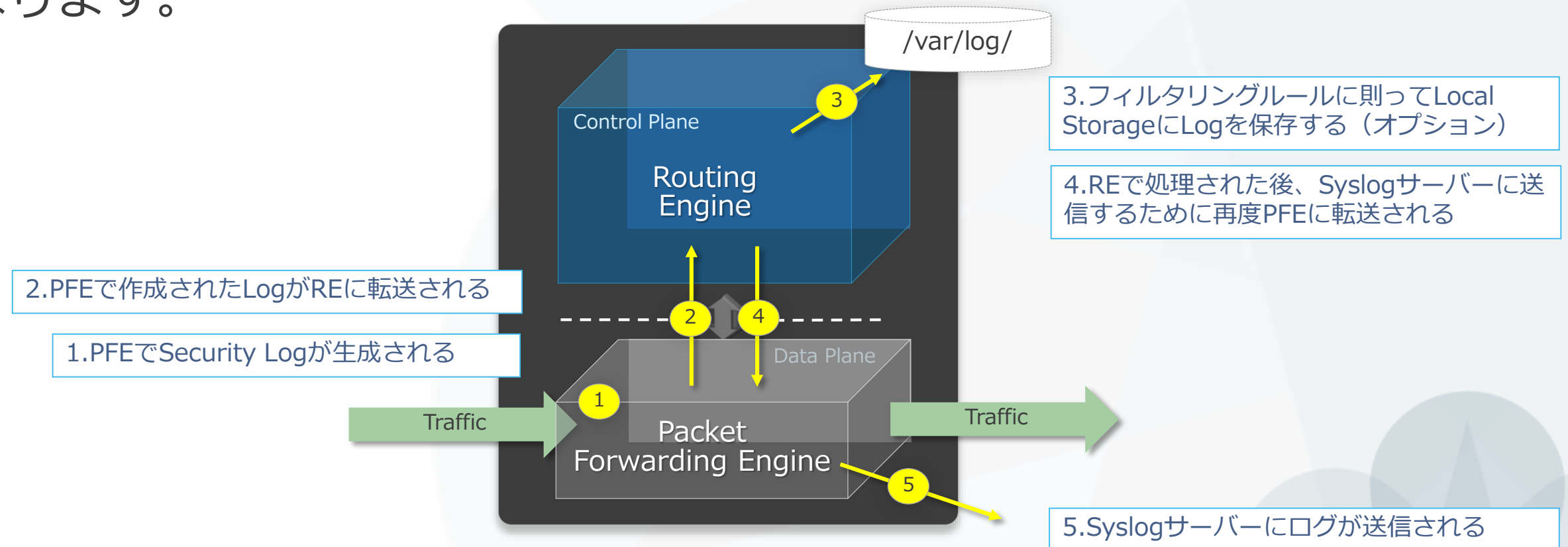
# Security Logging

- JUNOSのシステムにて利用される通常のシステムログとは別に、トラフィックログ (Security Logging) を取得することが可能です。
- Security Loggingは2つのフォーマットから選択が可能
  - 通常のSyslog (RFC3164)
  - Structured Syslog
    - より詳細なセキュリティ情報を取得したい場合に使用
- Security Loggingは2つの収集方法から選択が可能
  - Event Mode
    - Default設定 (最大1500 event/sec ※) ※ただしログングパフォーマンスはプラットフォームに依存
  - Stream Mode
    - 高負荷なトラフィック環境でSecurity Logの取得が必要な場合には推奨されるモード

# Security Logging

- Event Mode

- Security Logは一度Routing Engineで処理した後にSyslogサーバへ送信されるため、高トラフィック時にはRouting Engineの処理負荷が増大するのでデザインに検討が必要です。
- 一方で、Security Logのフィルタリングや内部Storageへの保存が可能な方式となります。





# Security Logging

- Event Mode

```
set security log mode event
set security log event-rate 100
set security log format sd-syslog
```

Event Modeを宣言して、イベントレート、フォーマットなどを指定

```
set system syslog host 192.168.0.99 any any
set system syslog host 192.168.0.99 match RT_FLOW
```

Traffic Logのメッセージは“RT\_FLOW”にマッチする

Syslogサーバーに送信する場合はHostを指定

```
set system syslog file TRAFFIC-LOG any any
set system syslog file TRAFFIC-LOG match RT_FLOW
```

LogをLocal Storageに保存する場合はFile名を指定

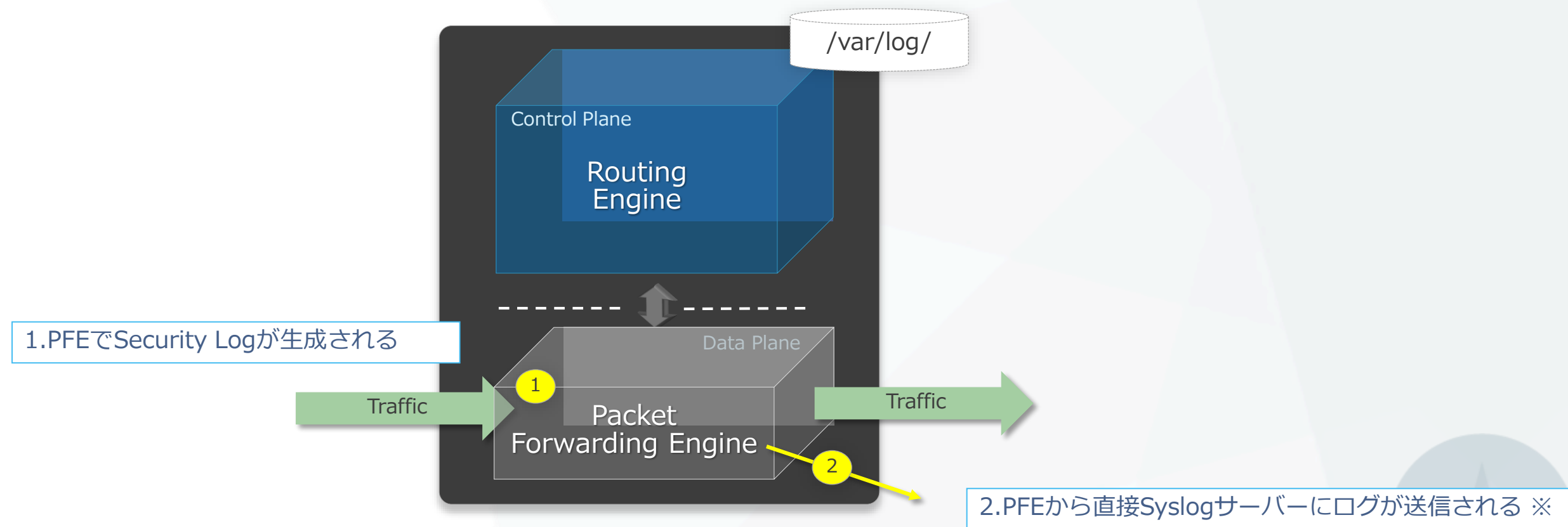
```
set security policies from-zone trust to-zone untrust policy trust-to-untrust match source-address any
set security policies from-zone trust to-zone untrust policy trust-to-untrust match destination-address any
set security policies from-zone trust to-zone untrust policy trust-to-untrust match application any
set security policies from-zone trust to-zone untrust policy trust-to-untrust then permit
set security policies from-zone trust to-zone untrust policy P1 match source-address any
set security policies from-zone trust to-zone untrust policy P1 match destination-address any
set security policies from-zone trust to-zone untrust policy P1 match application any
set security policies from-zone trust to-zone untrust policy P1 then permit
set security policies from-zone trust to-zone untrust policy P1 then log session-init
```

Security Logを取得したいFWポリシーでアクションを指定

# Security Logging

- Stream Mode

- Security Logは、Packet Forwarding Engine内で処理され、Syslogサーバーへ転送されます。（これにより高いLogging Rateを期待することができますが、Local StorageへのLog保存などは行えません。）



※Stream Mode使用時には、Syslog ServerへのLog送信はRevenue Portから送信される必要があります。（FXP0からの送信は未サポートとなります）

# Security Logging

- Stream Mode

```
set security log mode stream
set security log source-address 192.168.0.254
set security log stream TRAFFIC-LOG format sd-syslog
set security log stream TRAFFIC-LOG host 192.168.0.99
```

Stream Modeを宣言して、Source Address、フォーマット、Syslogサーバーのターゲットなどを指定

```
set security policies from-zone trust to-zone untrust policy trust-to-untrust match source-address any
set security policies from-zone trust to-zone untrust policy trust-to-untrust match destination-address any
set security policies from-zone trust to-zone untrust policy trust-to-untrust match application any
set security policies from-zone trust to-zone untrust policy trust-to-untrust then permit
set security policies from-zone trust to-zone untrust policy P1 match source-address any
set security policies from-zone trust to-zone untrust policy P1 match destination-address any
set security policies from-zone trust to-zone untrust policy P1 match application any
set security policies from-zone trust to-zone untrust policy P1 then permit
set security policies from-zone trust to-zone untrust policy P1 then log session-init
```

Security Logを取得したいFWポリシーでアクションを指定

---

# Thank you

---