Skalierbare Automatisierung und Orchestrierung im Netzwerk – Welche Vorteile sind entscheidend?

Christian Klewar, Juniper Networks

06 September 2017 | HSLU Informatik

Juniper Networks at a glance

Who are we?

Founded: February 1996
Headquarters: Sunnyvale, California
Employees: 9,300+
Offices: 88 locations in 43 countries
Revenue: USD $4.99 billion (2016)

JUNIPER NETWORKS SUPPORTS

- The Top 10 telecom companies in the world
- 10 of the top 12 global technology companies
- More than 1,400 national government organizations around the world.
- We operate 16 around-the-clock technical support centers around the world.

NETWORKING SPECIALISTS

Routing Switching Security Software

What do we do?

Our customers don't set out to build networks. They build on ideas that reinvent, reimagine, and improve the world around them.

If your business depends on the network to deliver mission-critical transactions, applications, and services, you do business with Juniper Networks.

3 PRIMARY SOLUTION AREAS

OPEN AND AUTOMATED CLOUD
Spine & Leaf Switching
SDN Automation
Network Analytics

HIGH PERFORMANCE AGILE WAN
Core & Edge Routing
Branch SD-WAN
Data Centre Interconnect
Optical Transport

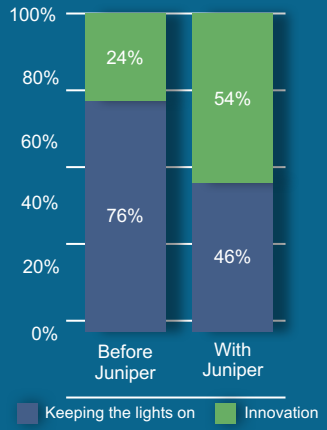
SECURE NETWORKS
Virtual & Physical Security
Dynamic Anti-Malware Protection
Software Defined Security

WE **AUTOMATE** IT
WE **CONNECT** IT
WE **SECURE** IT

Why work with us?

ebay classifieds group · TURKISH AIRLINES · PUMA · SONY · REVLON · orange

IT NETWORKING STAFF EFFICIENCY IMPROVEMENT*



PROVING THE BUSINESS VALUE OF NETWORK TRANSFORMATION

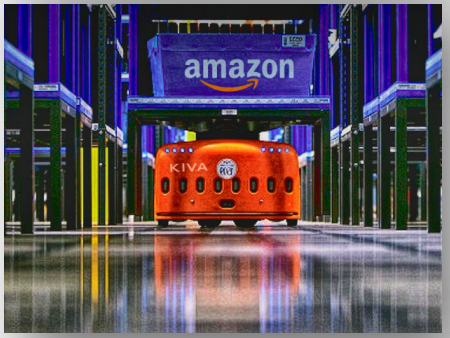
** IDC White Paper, sponsored by Juniper, Juniper Networks: Proving the Business Value of Network Transformation, September 2016* Juniper calculations based on IDC estimates for savings per 100 users

Automation: It's changing life around us

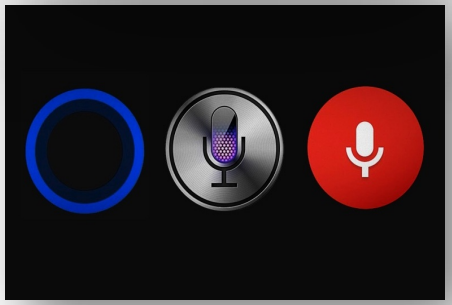
HOMES

CARS

SHOPPING



ASSISTANTS



LANGUAGE

GAMES

Automation: Setting the context

What?

"Using machines to run machines"

-- Peter F Drucker'1955

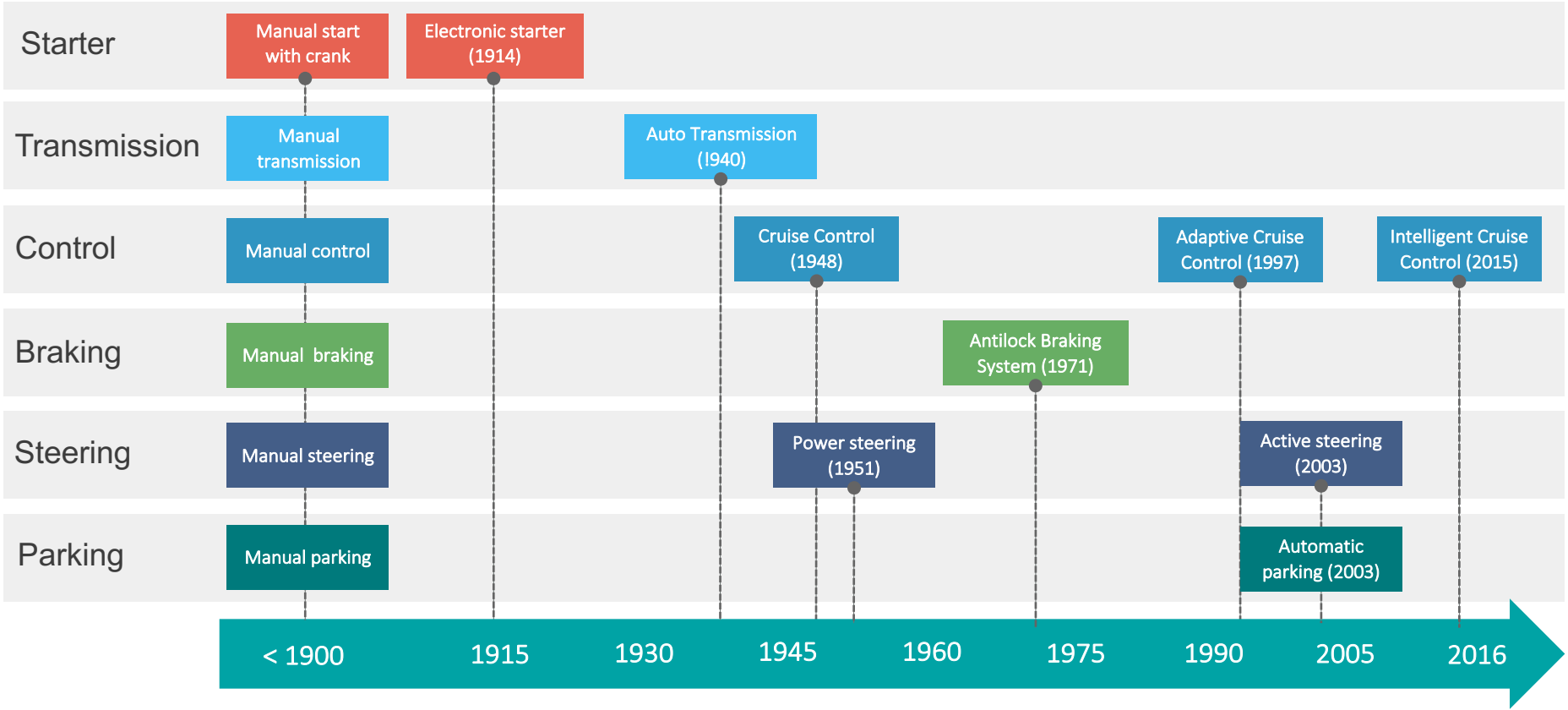
Why?

Agility! Delivering outcomes @ speed

How?

Technology, Culture and Process

Evolution of the automobile



Disruption of the automobile

- Not just an incremental improvement, a disruptive change

The Self-Driving Car

Is it a Car….Is it a Computer?

The Promise

Ownership: Delineate ownership & usage. Uber++, ZipCar++

Safety: Human errors cause 94% of car crashes

Planning: No more traffic lights? Triple highway capacity?

Logistics: Self-driving trucks to revolutionize package delivery

The Impact

Don't need drivers: Need programmers, operational folks

Don't need cops: Cars can (will) self-police

Don't need witnesses: Cars will be the most objective witnesses

How does insurance work: Who pays for the glitches?

Disruption of the Network

- Self-Driving Cars '*need*' Self-Driving Networks

A self-driving network would

- Accept *"guidance"* from a network operator
- Self-discover its constituent parts
- Self-configure
- Self-monitor using probes and other techniques
- Auto-detect when a new service is needed and auto-enable it
- Automatically monitor and update services to optimize service delivery
- Use machine learning for introspection (self-analysis)
- Self-report periodically or when an unexpected situation arises

Self-Driving Networks: A vision worth pursuing


Automation @ scale

WEB 2.0 COMPANIES

Reduce DC cooling bill by 40%

2014: Machine-learning algorithms used to predict Power Usage Effectiveness (PUE) of the datacenters with up to 99.6% accuracy

2016: Google DeepMind AI uses *historical data, telemetry & deep neural networks* to reduce Datacenter cooling bills by 40%

1 Engineer = 25,000 servers

2011: Facebook Auto Remediation (FBAR) to proactively detect and address production problems on *individual servers*.

2016: Automated Maintenance Handlers to safely automate maintenance on *multiple servers*. Dapper to co-ordinate both *automated and manual processes*

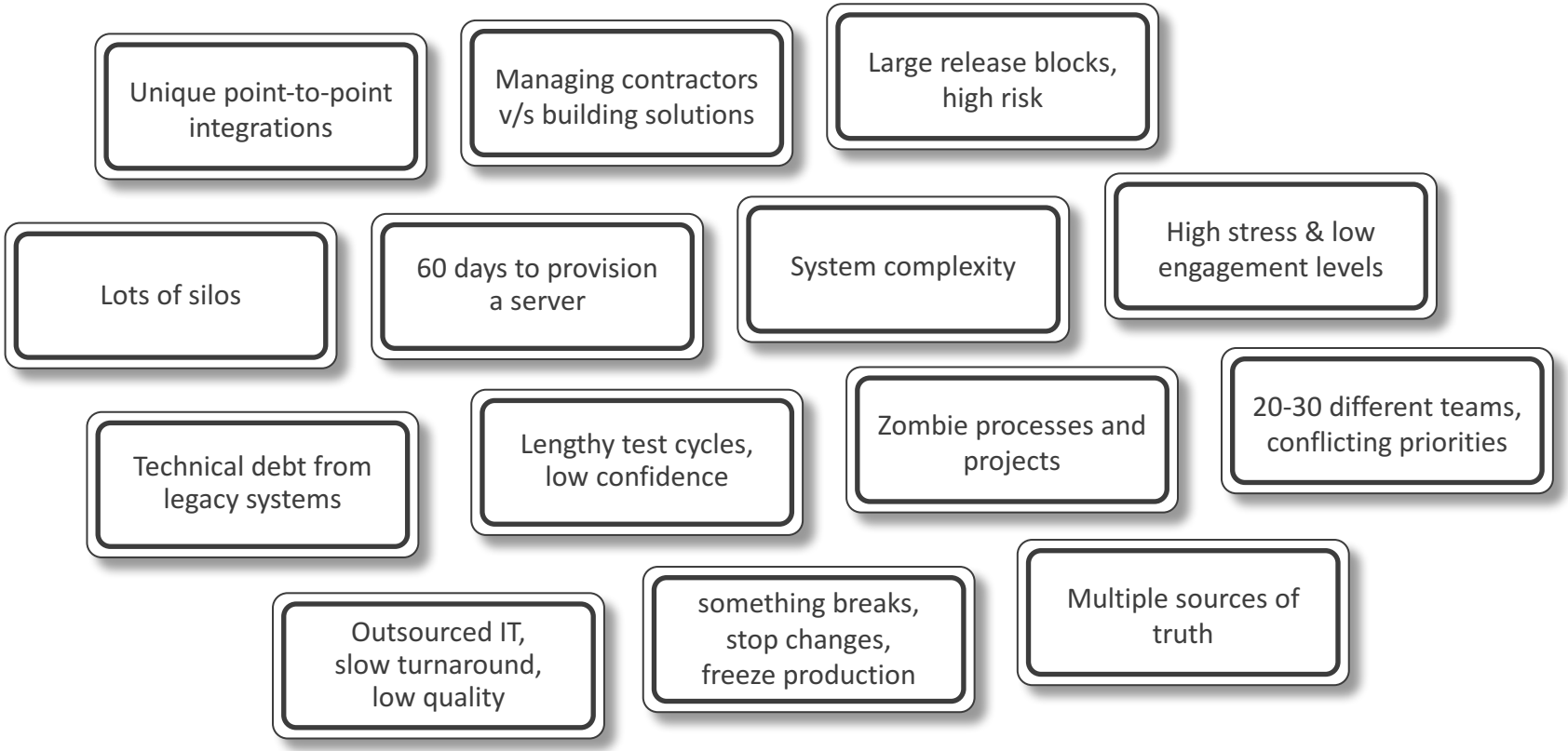
Helping developers sleep better

2013: Atlas, a sophisticated home-grown telemetry tool that collects up to *1.2 Billion data points per minute*

2016: Winston, outsources *repeatable diagnostics and remediation* tasks. Run automatically in response to events from Atlas

Charting the course for Automation

- *Identify, acknowledge and target your roadblocks*

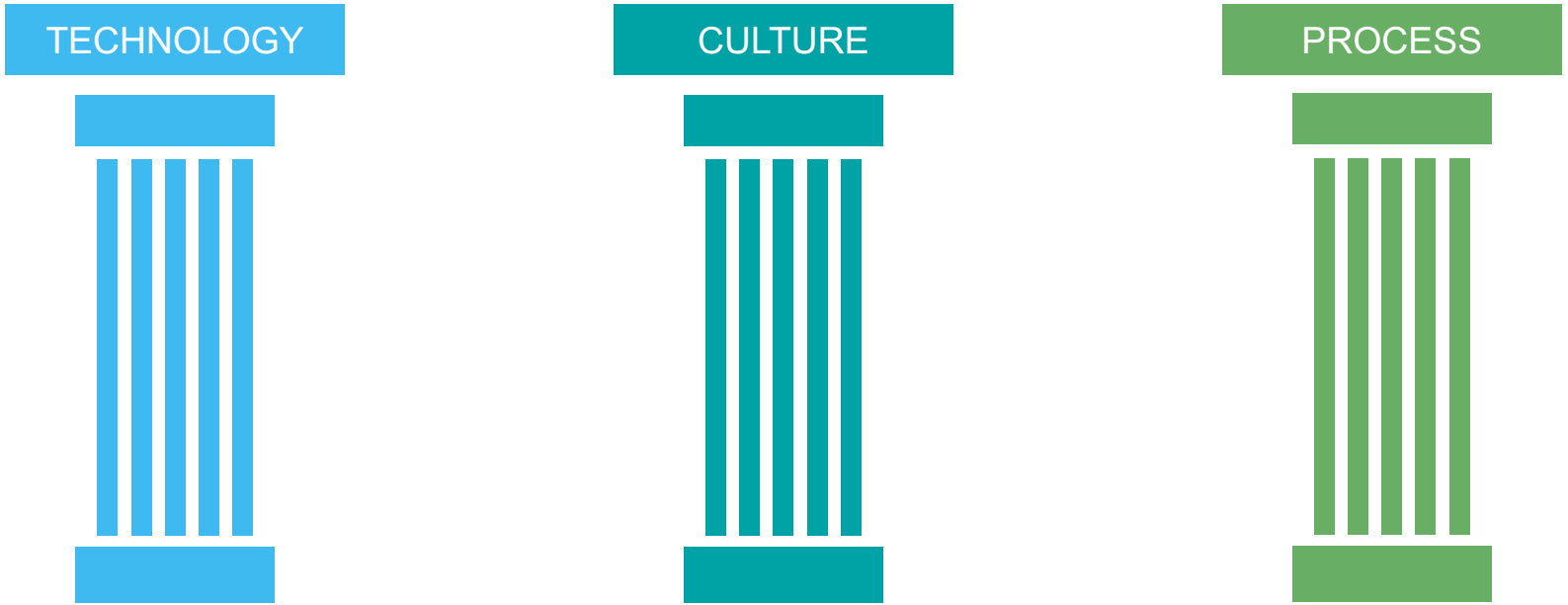


Charting the course for Automation

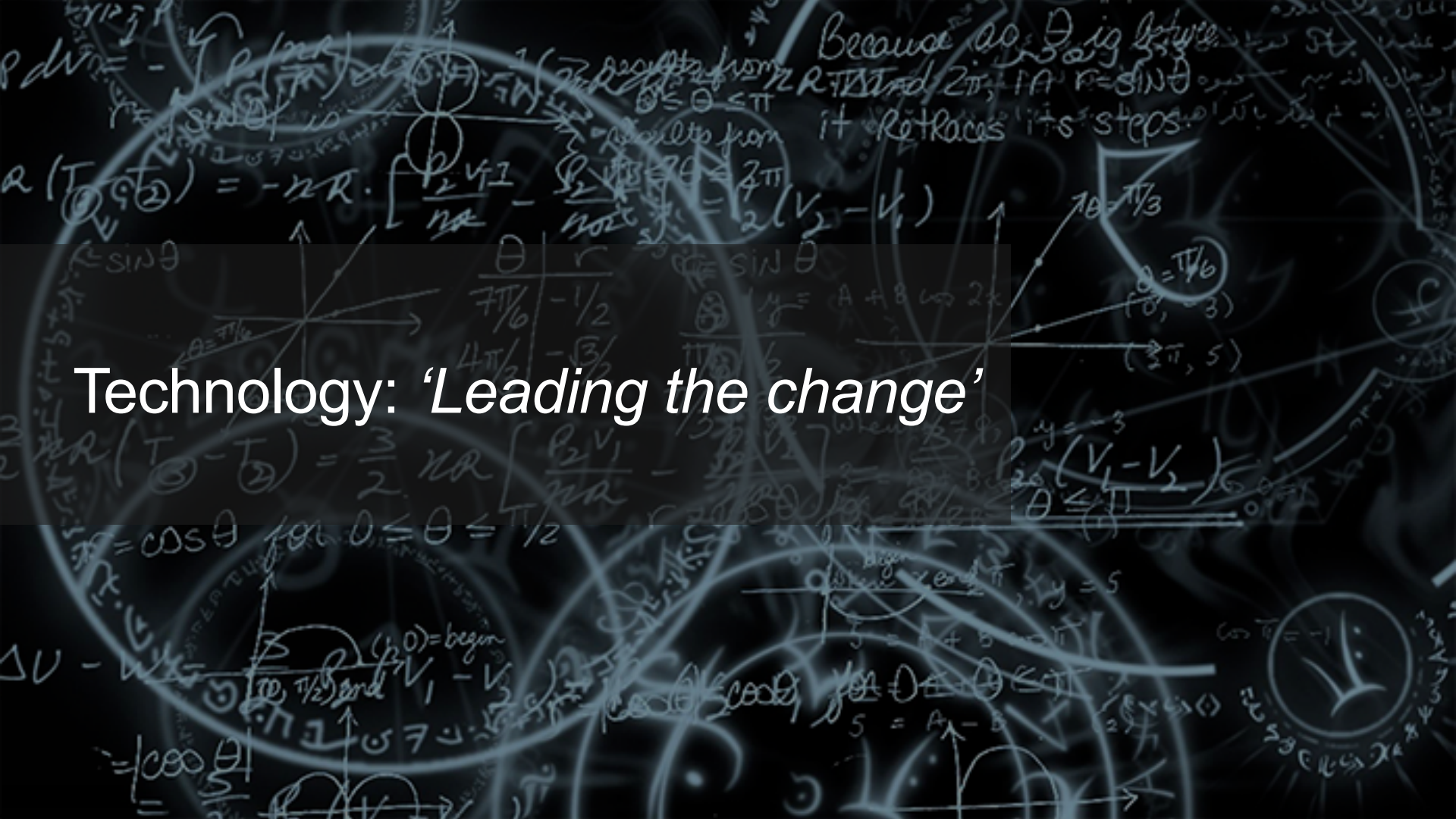
- *Identify, acknowledge and target your roadblocks*

Charting the course for Automation

Three building blocks of Automation



Find the right balance


Technology: *‘Leading the change’*

Network Automation: How do we get there?

Build an evolution path to fuel disruption

Day 0

Human-driven automation

Establish standard-based network interfaces and data models

Automate network provisioning & management

Simplify network operations

Day 1

Event-driven automation

Gather network information (Telemetry)

Use network information to make decisions

Rule-based action on network events (Closed loop automation)

Day N

Machine-driven automation

Use machine-learning tools to train the system

Machines makes decisions and drives network change

Humans make decisions where machines cannot

Automation Technologies - Landscape

Automation Technologies - Landscape

Day 0 - Human-driven Automation

Day 0

Human-driven automation

Establish standard-based network interfaces and data models

Automate network provisioning & management

Simplify network operations

Standards-based network interfaces and data-models

netconf

YANG

OPENCONFIG

Automate network provisioning & management

CHEF

puppet

ANSIBLE

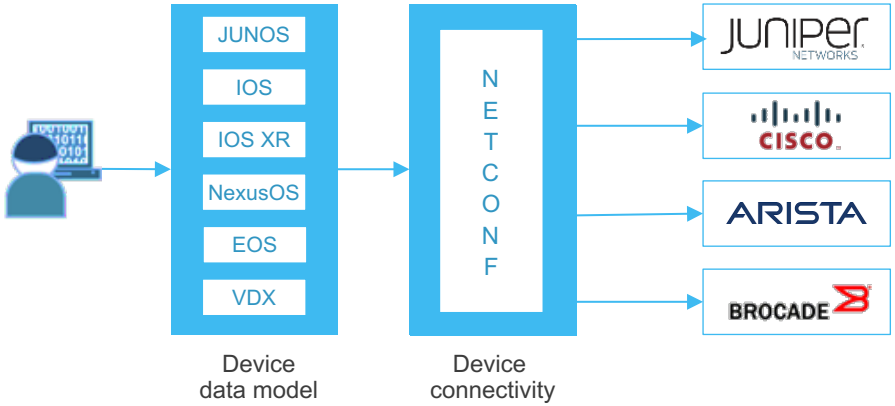
PyEZ

Simplify network operations

JSNAPy

Standards-based Network interfaces

- Uniform, vendor-neutral approach to access and configure devices



NETCONF

Protocol to “install, manipulate and delete configuration”

Uses XML-based data encoding for configuration data and protocol messages

NETCONF protocol operations over a simple RPC layer

Programmable: Python libraries (ncclient), Juniper PyEZ

Based on RFC Standards (RFC 4741, 4742, 6241, 6242)

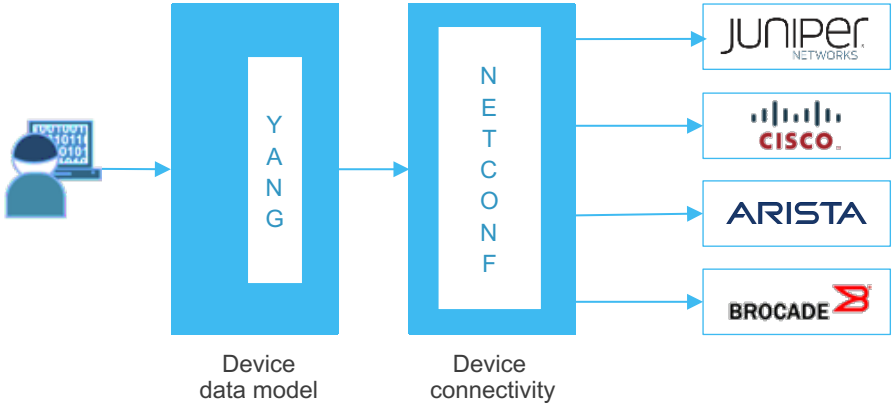
Juniper support

NETCONF is the IETF standard for managing devices and is derived from JUNOScript API (2001)

Supported on all Juniper platforms, pre-JUNOS 4.0

Standards-based Network interfaces

- Uniform, vendor-neutral approach to access and configure devices



YANG

Data model language for the Network Configuration Protocol (NETCONF)

Human readable, Supports multiple encoding formats, including XML and JSON

Transport over NETCONF over SSH and recently also over gRPC

Based on RFC standards (RFC 6020)

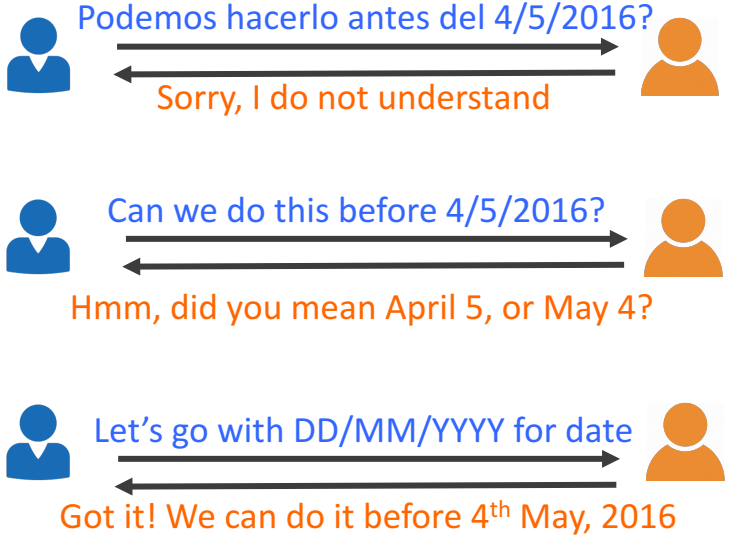
Juniper support

YANG is the industry-standard data modeling language and is based of Juniper's Data Definition Language (DDL, 2001)

Active IETF participation to define standard YANG modules, Support for custom YANG modules

Supported on MX/EX/M/PTX/T-series platforms, JUNOS 14.2 onwards

An Analogy to Netconf and YANG



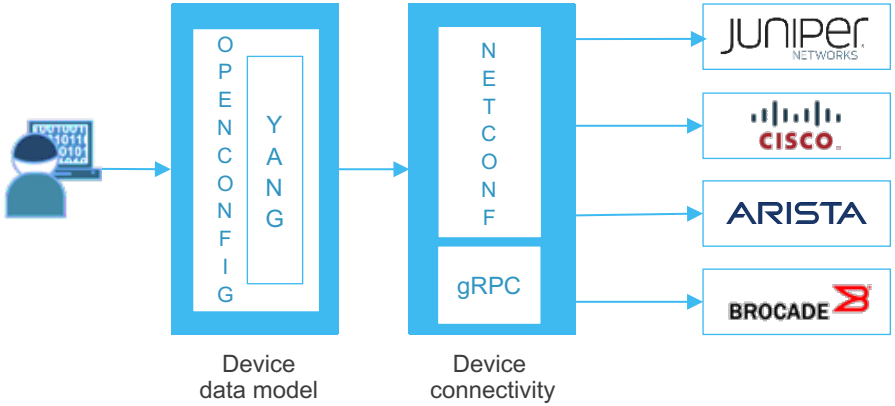
- Different communication protocols
- Systems do not understand each other without translation

- **Common protocol between systems (i.e Netconf)**
- However, model for representing date is unclear

- **Common data model for date: DD/MM/YY (i.e YANG)**
- Unambiguous and easy communication

Standards-based Network interfaces

- Uniform, vendor-neutral approach to access and configure devices



OPENCONFIG

Vendor-neutral, model-driven network management

Common Data Models written in YANG

Streaming Telemetry

Industry-driven: *Google, AT&T, British Telecom, Microsoft, Facebook, Comcast, Verizon, Level3, Apple, Deutsche Telekom, Bell Canada*

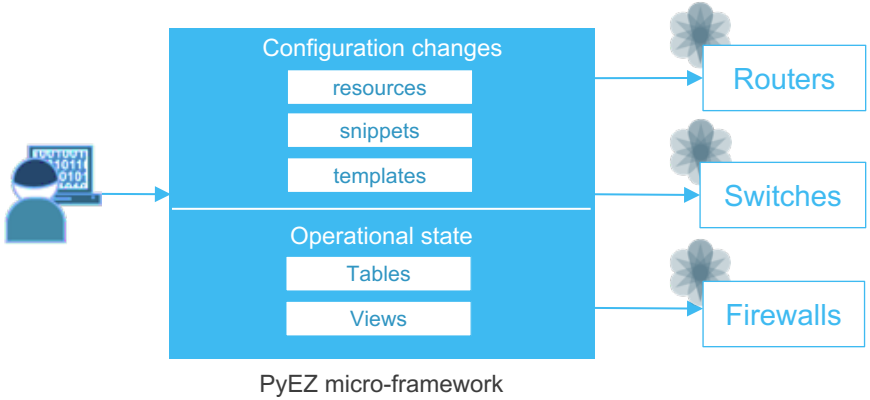
Juniper support

Actively engaged in the OpenConfig initiative since inception

Supported on MX/M/PTX/T-series platforms, JUNOS 16.1 onwards. EX/QFX platforms on roadmap

Automated network provisioning and management

- Consistent and compliant network operation



PyEZ micro-framework

PyEZ

Automation micro-framework for JUNOS devices

Remote connectivity and management

Retrieve configuration, operational or run-state information

Make configuration changes, secure copy of files and software updates

Juniper support

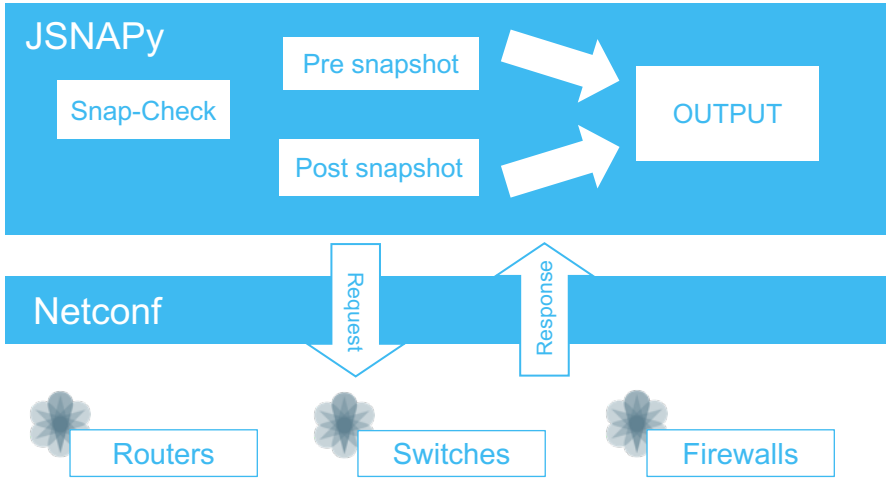
Supported on all Juniper platforms, JUNOS 11.4 onwards

Increasing popularity in the Juniper DevOps community (>300 stars on Github)

https://github.com/Juniper/py-junos-eznc

Simplified network operations

- Compare before and after operational state



JSNAPy

JUNOS Snapshot Administrator in Python

Capture and audit runtime environment snapshots of Juniper devices

Compare pre & post operation outputs

Simplified YAML-based test cases

Juniper support

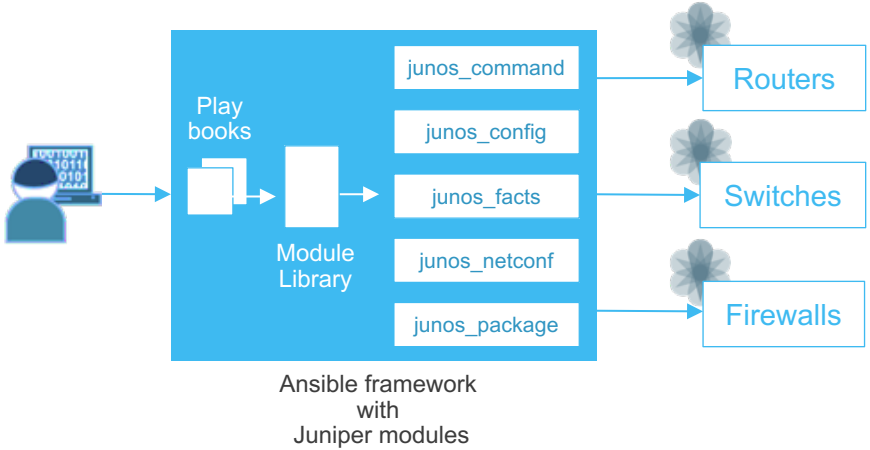
Juniper proprietary

Supported on all Juniper platforms, JUNOS 16.1 onwards

https://github.com/Juniper/jsnapy

Automated network provisioning and management

- Consistent and compliant network operation



ANSIBLE

Agent-less approach, Easy to deploy

Uses YAML ‘playbooks’ to define automation tasks

Works by pushing ‘Ansible modules’ to devices

Supports workflow engine

Juniper support

Supported on all Juniper platforms, JUNOS 12.3 onwards

Enhanced capabilities using Ansible modules for Juniper.

https://www.ansible.com/ansible-juniper

Ansible to set SNMP Community on many devices

```
$ cat resources/set-snmp.set
set snmp community public authorization read-only
```

Junos CLI Command to apply

```
$ cat set-snmp.yml
---- name: Enable snmp on my Junos devices
  hosts: mylab
  roles:  - Juniper.junos
  connection: local
  gather_facts: no
  tasks:
  - name: Checking NETCONF connectivity
    wait_for: host={{ inventory_hostname }} port=830  timeout=5

  - name: Enable snmp and set community public
    junos_install_config:
      host={{ inventory_hostname }}
      user=mwiget
      file=resources/set-snmp.set
      overwrite=false
      logfile=logs/set-snmp.log
```

Ansible Playbook

Running the Ansible Playbook against devices

```
$ export ANSIBLE_HOSTS=$PWD/hosts
$ ansible-playbook –i hosts enable-snmp.yml

PLAY [Enable snmp on my Junos devices]
****************************************

TASK: [Checking NETCONF connectivity]
****************************************
ok: [vmx-yang]
ok: [vmx1]
ok: [vmx-automation]

TASK: [Enable snmp and set community public]
*********************************
ok: [vmx1]
ok: [vmx-yang]
changed: [vmx-automation]

PLAY RECAP
************************************************************************
vmx-automation             : ok=2    changed=1    unreachable=0    failed=0
vmx-yang                   : ok=2    changed=0    unreachable=0    failed=0
vmx1                       : ok=2    changed=0    unreachable=0    failed=0
```

cat ~/hosts

```
[mylab]
vmx1
vmx-yang
vmx-automation
```

Day 1 – Event-driven Automation

Day 1

Event-driven automation

Gather network information (Telemetry)

Use network information to make decisions

Rule-based action on network events (Closed loop automation)

Gathering Network Telemetry

OPENCONFIG

GRPC

JVision

Rule-based actions on network events

SALTSTACK

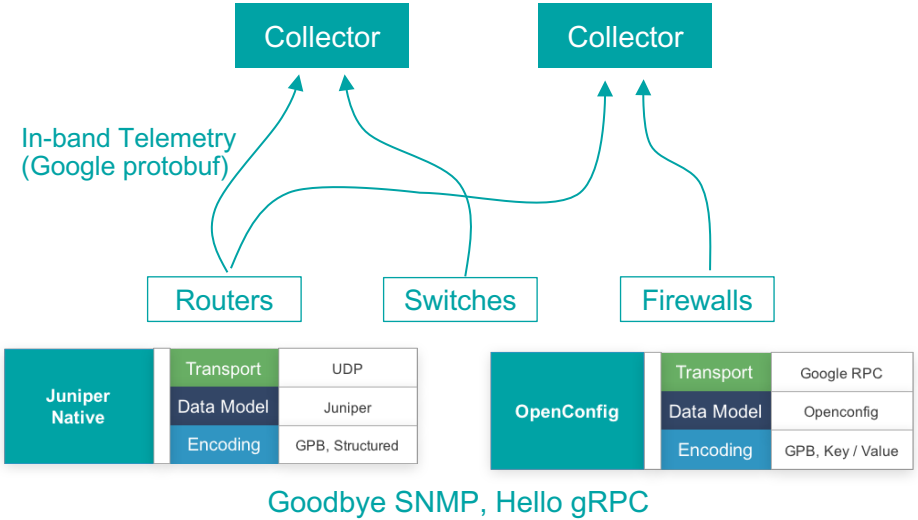
python

SLAX

JET

Gathering Network Telemetry

- What gets measured, gets managed



Goodbye SNMP, Hello gRPC

gRPC & JVision

Push-based telemetry model (v/s pull-based SNMP)

Continuous streaming of Network telemetry data based on subscriptions

Observe network state through time-series data stream and take action.

Uses Google protocol buffer encoding format

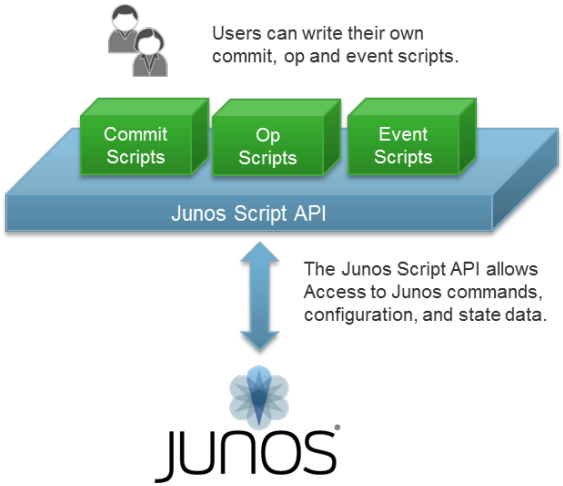
Juniper support

Supported on Juniper MX and PTX platforms JUNOS 15.1F3 onwards

OSS Collector OpenNTI https://github.com/Juniper/open-nti

Rule-based actions on network events “on box”

- If-This-Then-That (IFTTT) model



SLAX & Python (on-box)

Support for SLAX and now Python on-box on JUNOS devices

Write scripts to react to on-box network events

Commit Script: Configuration consistency checks

Operational Scripts: Monitoring and troubleshooting

Event scripts: Event-based triggers

Juniper support

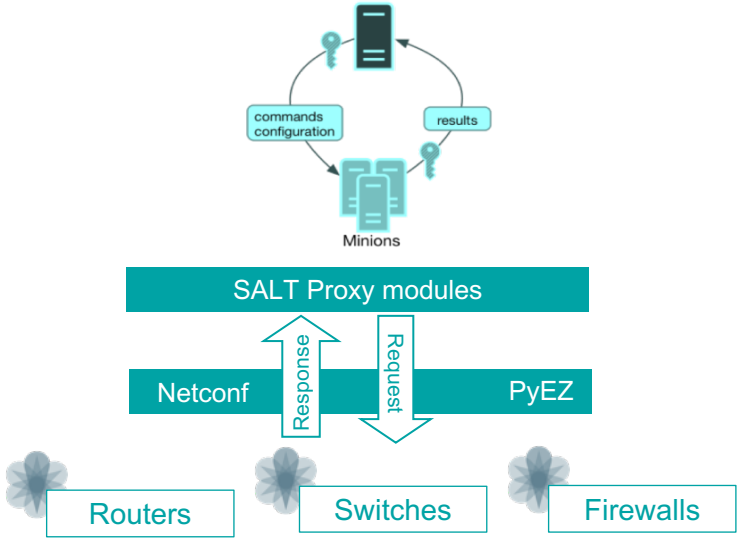
Supported on all Juniper MX, PTX, QFX platforms,

SLAX: pre-JUNOS 7.0 onwards; Python on-box: JUNOS 16.1 onwards

https://github.com/Juniper/junoscriptorium

Rule-based actions on network events “off box”

- If-This-Then-That (IFTTT) model



Salt Stack

Configuration management system, capable of maintaining remote nodes in defined states

Uses a pub/sub model to publish events from master or agent

Rule-based actions can be built to react to these events

Junos support for SaltStack is through a off-box proxy minion

Juniper support

Supported on all Juniper MX, PTX, QFX platforms, JUNOS 11.4 onwards

Example: Automate provisioning devices

Use cases

Provisioning

- Provision / stage devices
  - SRX / vSRX / EX / NFX
  - Third party using Napalm (Configuration only)
- Verification

- Provisioning can be done using
  - staging area / network
  - drop shipping device

- Real ZTP with
  - staging Area approach (DCHP / TFTP)
  - drop shipping approach (PHS)

Use cases

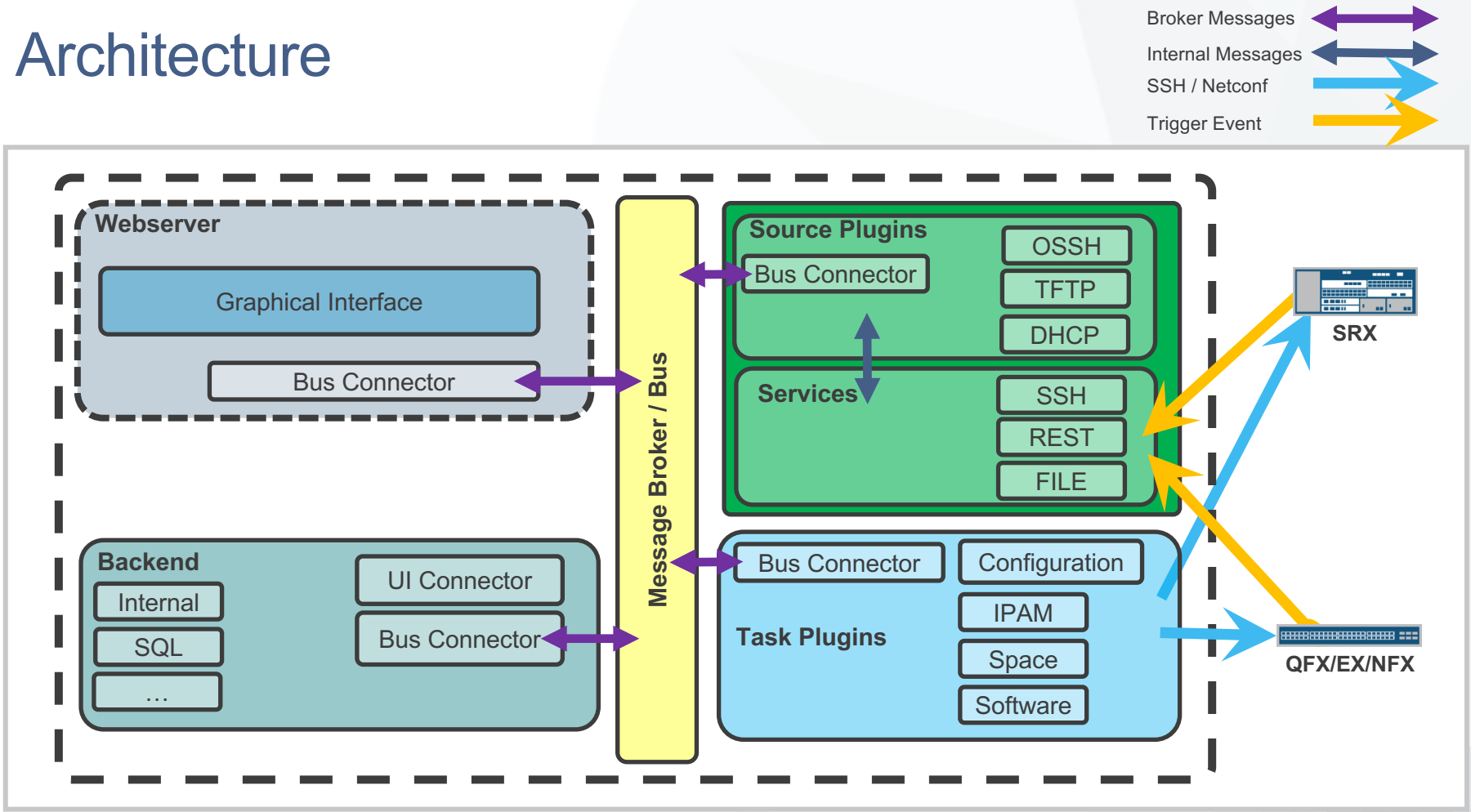
RMA

- RMA dead device

- Use drop shipping approach (ZTP)
- Use of additional apps for activation

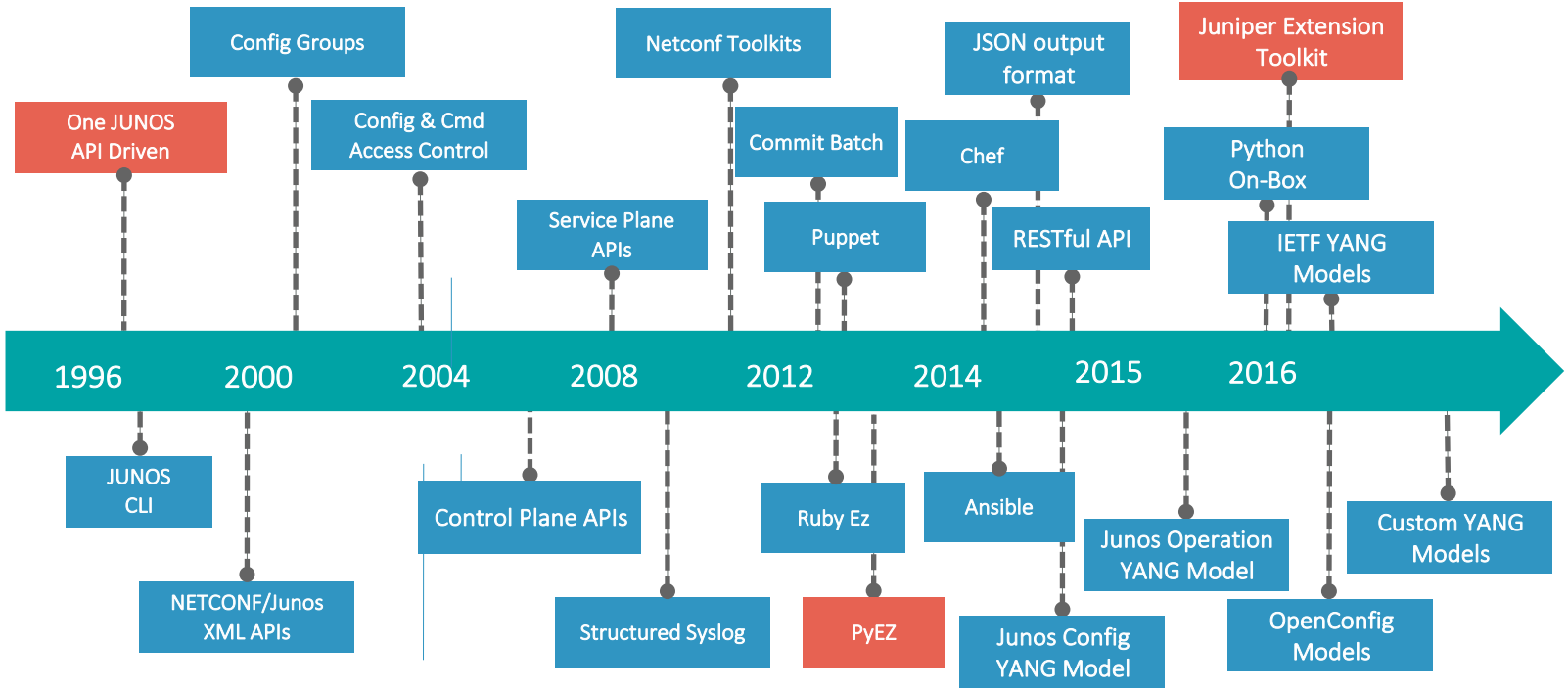
- Keep RMA process as simple as possible
- IT personnel also need some sleep

Architecture



Automation @ Juniper: It's in our DNA

- Built with an *'Automation-first'* mindset



Network Automation: Leading the pack

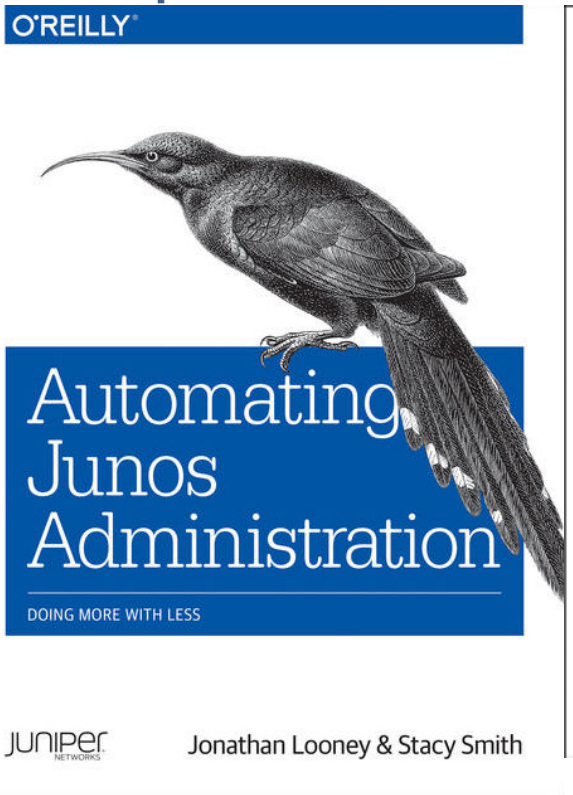
Third-party perspective on platform-vendor capabilities for Network Automation

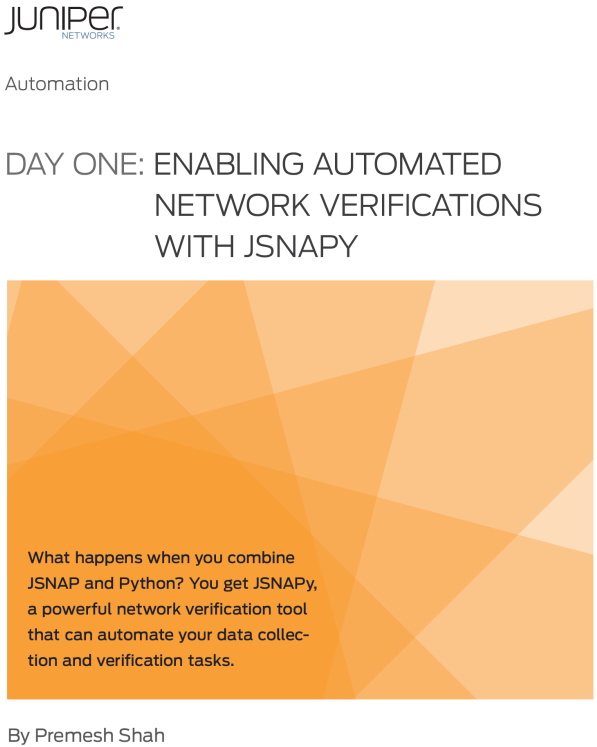
1 On-device APIs to read / write configuration and operational data ✓ Juniper Networks: PASS	2 Structured operational data for easy programmatic analysis ✓ Juniper Networks: PASS	3 Structured device configuration data for easy programmatic analysis ✓ Juniper Networks: PASS	4 Atomic configuration changes to avoid partial updates ✓ Juniper Networks: PASS
5 Configuration rollback to minimize risk ✓ Juniper Networks: PASS	6 Full configuration replace that makes templates easy to use ✓ Juniper Networks: PASS	7 Configuration difference analysis to simplify manual approvals ✓ Juniper Networks: PASS	8 Industry standard data models for configurations ✓ Juniper Networks: PASS

READ THE BLOG

http://blog.ipspace.net/2016/10/network-automation-rfp-requirements.html

Juniper Books On Automation





Summary

Automation is here to simplify network operation, reduce errors and allow faster time to market