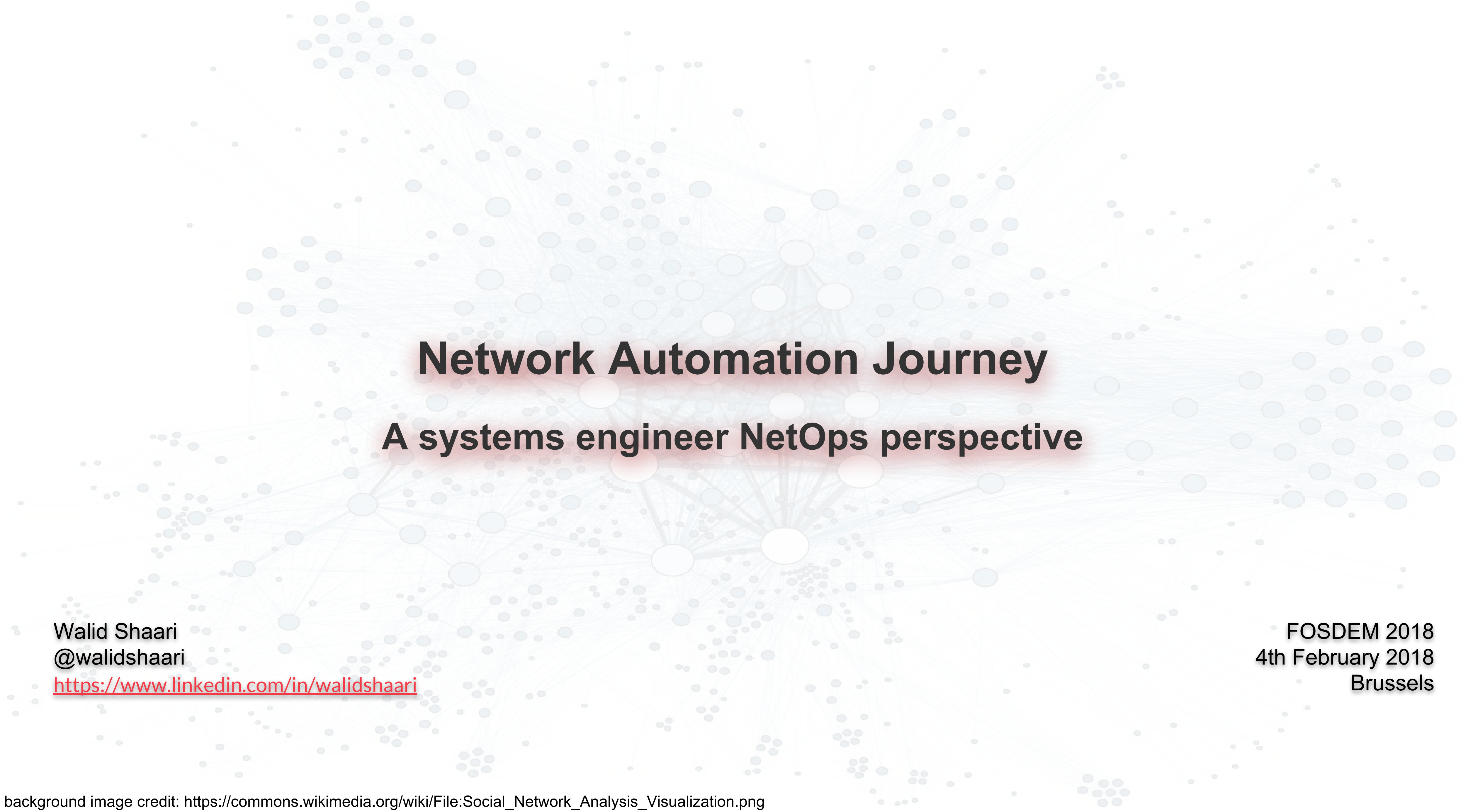


Network Automation Journey

A systems engineer NetOps perspective

Walid Shaari
@walidshaari
https://www.linkedin.com/in/walidshaari

FOSDEM 2018
4th February 2018
Brussels

background image credit: https://commons.wikimedia.org/wiki/File:Social_Network_Analysis_Visualization.png

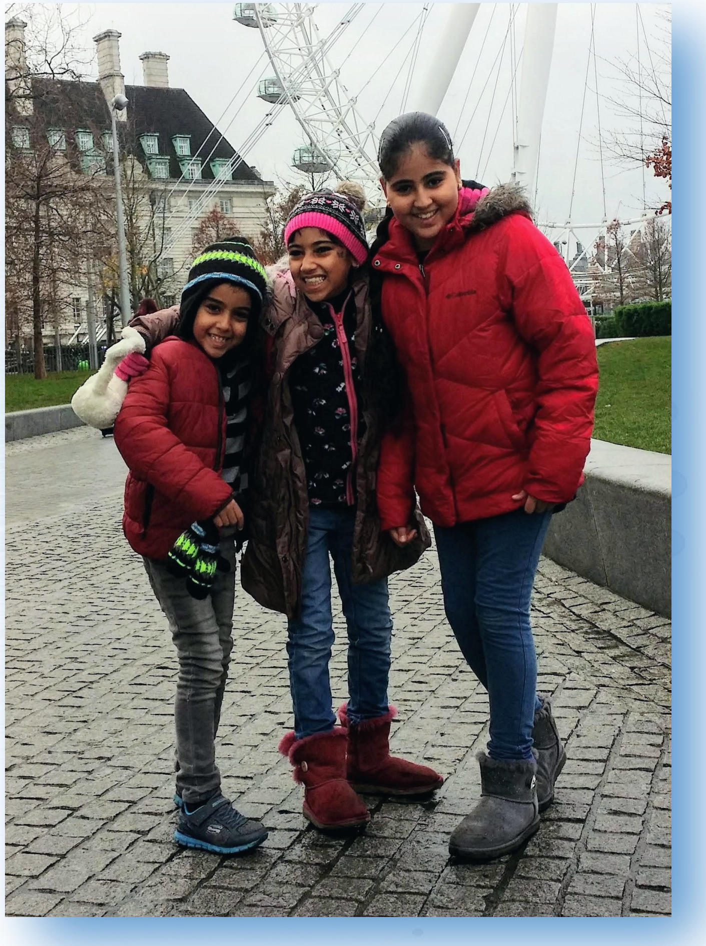
> show user

Walid Shaari

@walidshaari

https://www.linkedin.com/in/walidshaari

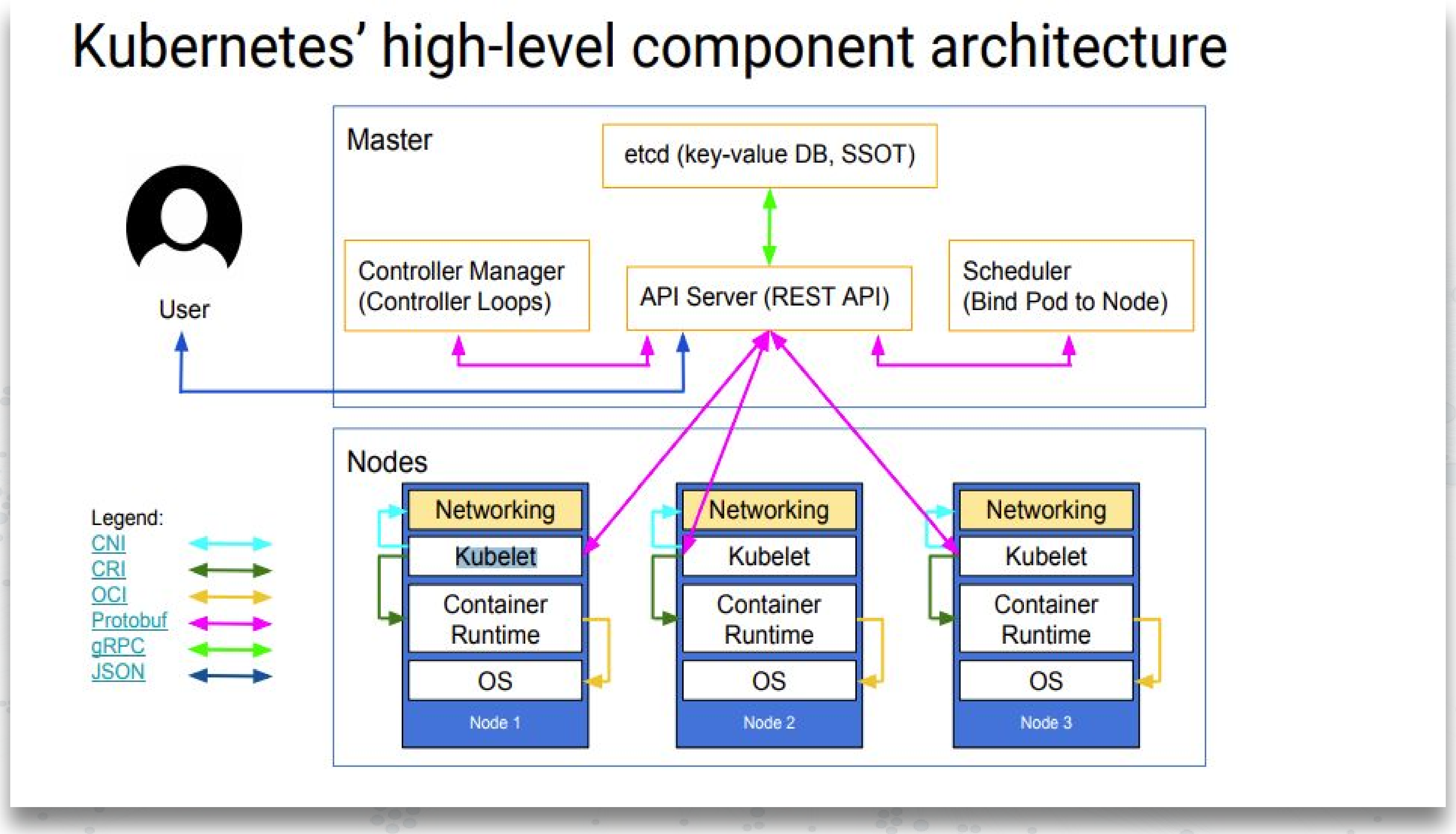
- System engineer supporting HPC Linux clusters
- Configuration management evaluation and deployment project in 2014
- Advocating open source, automation, containers and Kubernetes
- Husband and father of 3 lovely kids
- Last 3 months in short work assignment with Network management team



CERTIFIED kubernetes ADMINISTRATOR

RED HAT CERTIFIED ARCHITECT redhat. 110-346-749

Incentives



Kubernetes’ high-level component architecture

- Open source and standards
- Pure Layer 3 network implementation
- Lightweight IP to IP encapsulation
- Policy based secure networking
- Scalable and simple
- Scale out SDN controller
- Openstack , Docker, Kubernetes, Mesos and CNI

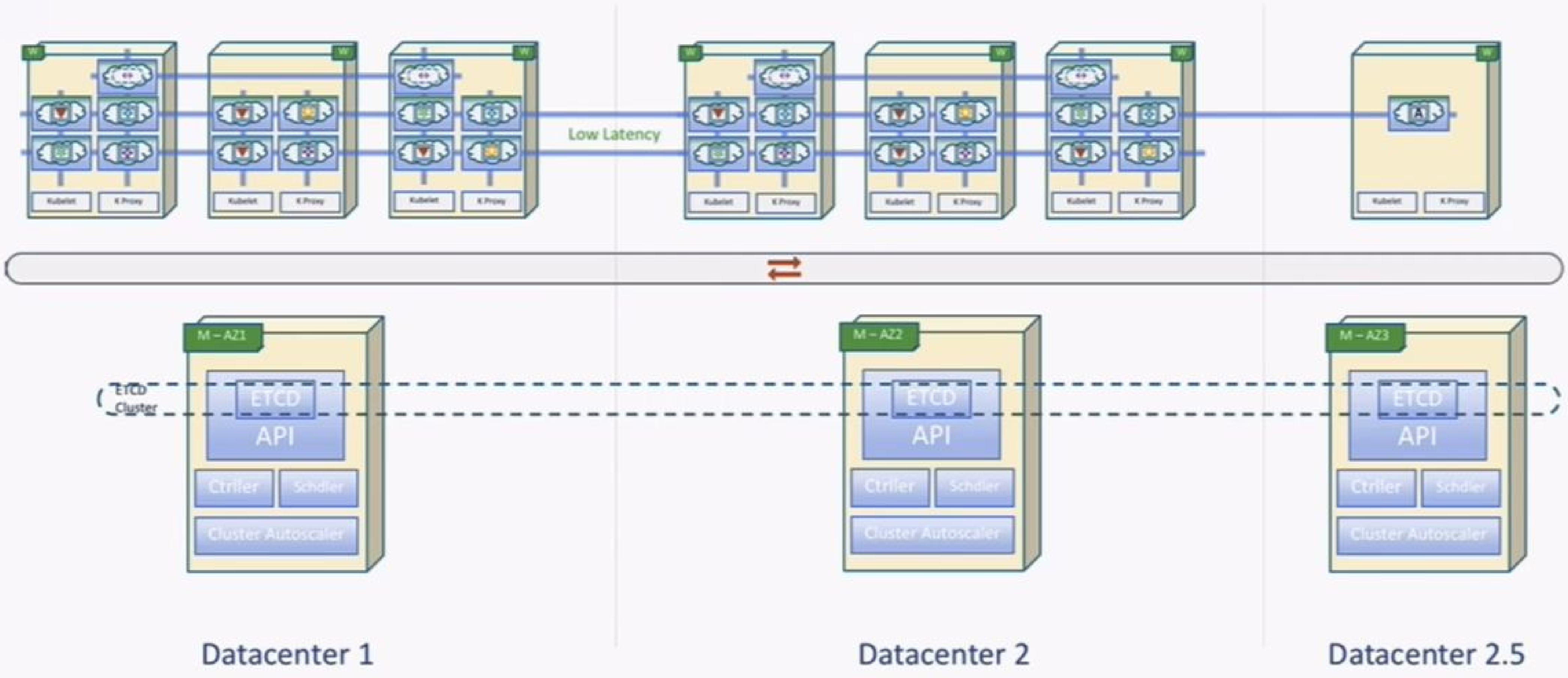
Networking setups in 2018

“In 2018, we will see more demands placed on the **continuous delivery** of changes to networking setups due to pressure from containerisation, distributed systems, and security needs. Thus, networking must become as flexible and automation-friendly as the software that runs over it, and become less of a bottleneck.”

Nigel Kersten, Chief Technical Strategist at Puppet

https://www.itproportal.com/features/what-do-organisations-need-to-prepare-for-in-2018/

Multi-Datacenter Setup



©The Glue

How to build an event driven, dynamically reconfigurable microservices platform by Sven Beauprez:
https://www.youtube.com/watch?time_continue=388&v=1D8hyLWMtfM

People
Process
Technology

Enterprise Network management trends

CLI

Cut & Paste
NPA Notepad Automation

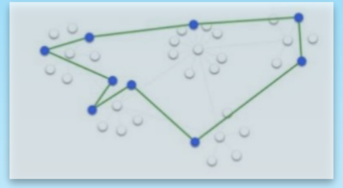
Frameworks & Controllers

- Excel
- Python jinja2
- Templating engines
- Ansible,
- Puppet
- Chef

Event Driven Automation

- Sensor triggered events
- napalm_logs
- Salt
- IFTTT
  - StackStorm
    Ansible AWX

Intent Based Networking

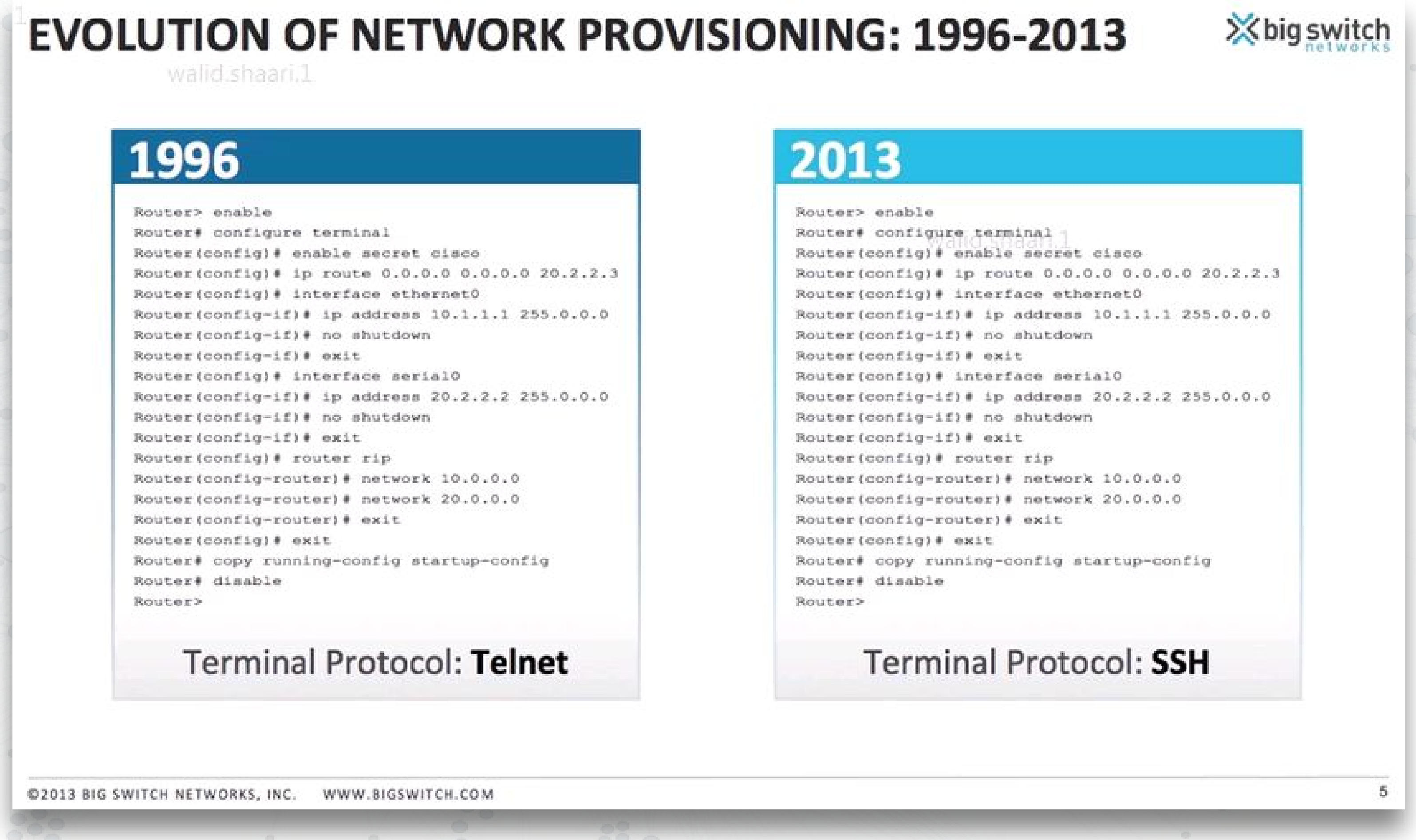


Declarative

Network Intent Composition
Aspen
Boulder

Enterprise Network management

- Manual
- Cut & Paste
- Serial
- Inconsistent
- Error prone



EVOLUTION OF NETWORK PROVISIONING: 1996-2013

1996

```
Router> enable
Router# configure terminal
Router(config)# enable secret cisco
Router(config)# ip route 0.0.0.0 0.0.0.0 20.2.2.3
Router(config)# interface ethernet0
Router(config-if)# ip address 10.1.1.1 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial0
Router(config-if)# ip address 20.2.2.2 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 20.0.0.0
Router(config-router)# exit
Router(config)# exit
Router# copy running-config startup-config
Router# disable
Router>
```

Terminal Protocol: **Telnet**

2013

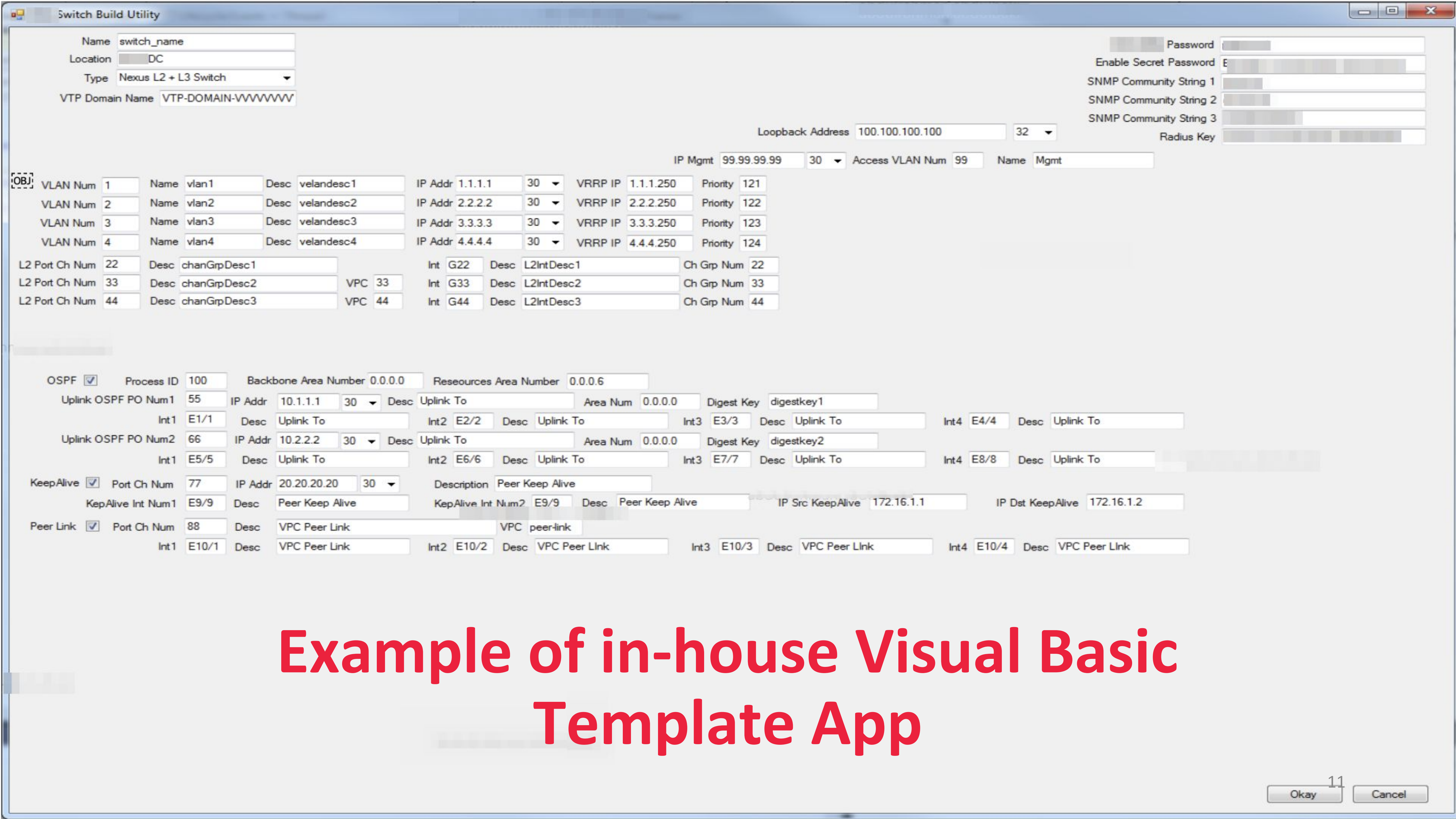
```
Router> enable
Router# configure terminal
Router(config)# enable secret cisco
Router(config)# ip route 0.0.0.0 0.0.0.0 20.2.2.3
Router(config)# interface ethernet0
Router(config-if)# ip address 10.1.1.1 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# interface serial0
Router(config-if)# ip address 20.2.2.2 255.0.0.0
Router(config-if)# no shutdown
Router(config-if)# exit
Router(config)# router rip
Router(config-router)# network 10.0.0.0
Router(config-router)# network 20.0.0.0
Router(config-router)# exit
Router(config)# exit
Router# copy running-config startup-config
Router# disable
Router>
```

Terminal Protocol: **SSH**

https://www.slideshare.net/opennetsummit/ons2013-guido-appenzellerbig-switch-networks

Why we are not automating?

- Just a fad: All these are is vendor driven buzzwords, exaggerated, will fade away
- Not relevant to our setup,
  - Mixed diverse vendor and legacy platform environment, no solution can handle this, it will be cost prohibitive.
  - Do not need a bazooka to handle a mosquito. We are small
  - Busy, shortage of resources
  - None of my acquaintances in the industry is doing it.
- It is hard, steep learning curve. we need to learn a lot of new things
- Vendor advanced training e.g. CCIE teaches us differently
- This will affect our job security, automate us out of the job
- Automation pushes mistakes faster, can bring everything down
- Do not know where to start?



Switch Build Utility [redacted]

Name switch_name

Location [redacted] DC

Type Nexus L2 + L3 Switch

VTP Domain Name VTP-DOMAIN-VVVVVVVV

[redacted] Password [redacted]

Enable Secret Password E[redacted]

SNMP Community String 1 [redacted]

SNMP Community String 2 [redacted]

SNMP Community String 3 [redacted]

Radius Key [redacted]

Loopback Address 100.100.100.100 32

IP Mgmt 99.99.99.99 30 Access VLAN Num 99 Name Mgmt

OBJ

VLAN Num 1 Name vlan1 Desc velandesc1 IP Addr 1.1.1.1 30 VRRP IP 1.1.1.250 Priority 121

VLAN Num 2 Name vlan2 Desc velandesc2 IP Addr 2.2.2.2 30 VRRP IP 2.2.2.250 Priority 122

VLAN Num 3 Name vlan3 Desc velandesc3 IP Addr 3.3.3.3 30 VRRP IP 3.3.3.250 Priority 123

VLAN Num 4 Name vlan4 Desc velandesc4 IP Addr 4.4.4.4 30 VRRP IP 4.4.4.250 Priority 124

L2 Port Ch Num 22 Desc chanGrpDesc1 Int G22 Desc L2IntDesc1 Ch Grp Num 22

L2 Port Ch Num 33 Desc chanGrpDesc2 VPC 33 Int G33 Desc L2IntDesc2 Ch Grp Num 33

L2 Port Ch Num 44 Desc chanGrpDesc3 VPC 44 Int G44 Desc L2IntDesc3 Ch Grp Num 44

OSPF ☑ Process ID 100 Backbone Area Number 0.0.0.0 Reseources Area Number 0.0.0.6

Uplink OSPF PO Num1 55 IP Addr 10.1.1.1 30 Desc Uplink To Area Num 0.0.0.0 Digest Key digestkey1

Int1 E1/1 Desc Uplink To Int2 E2/2 Desc Uplink To Int3 E3/3 Desc Uplink To Int4 E4/4 Desc Uplink To

Uplink OSPF PO Num2 66 IP Addr 10.2.2.2 30 Desc Uplink To Area Num 0.0.0.0 Digest Key digestkey2

Int1 E5/5 Desc Uplink To Int2 E6/6 Desc Uplink To Int3 E7/7 Desc Uplink To Int4 E8/8 Desc Uplink To

KeepAlive ☑ Port Ch Num 77 IP Addr 20.20.20.20 30 Description Peer Keep Alive

KepAlive Int Num1 E9/9 Desc Peer Keep Alive KepAlive Int Num2 E9/9 Desc Peer Keep Alive IP Src KeepAlive 172.16.1.1 IP Dst KeepAlive 172.16.1.2

Peer Link ☑ Port Ch Num 88 Desc VPC Peer Link VPC peer-link

Int1 E10/1 Desc VPC Peer Link Int2 E10/2 Desc VPC Peer LInk Int3 E10/3 Desc VPC Peer LInk Int4 E10/4 Desc VPC Peer LInk

Example of in-house Visual Basic Template App

Kickstart: Brainstorm

- What is NetOps, NetDevOps?
- What problems we are trying to solve?
- How much visibility into operations we have?
- What are your responsibilities?
- Tasks/processes you are responsible for ?
- how much time you can set aside to look into automation?
- What current automations already in place? (processes or tools)
- Why would you like to have automations processes/tools?
- What automations you are aware of, or would like to have? processes/tools
- sharing medium, how should we enhance activity communications?
  (knowledge, process, updates, documentation, in-job training)
- How should we start?

Context

Team:

- handful network engineers supported by handful infrastructure & cabling

Network:

- Campus several building and Data Centers
- Medium scale less than 200 switches
- heterogeneous platforms and models:
- existing tooling in place
    - CMDB
    - IPAM
    - NMS : SNMP based
- **setup**: Traditional Three tier architecture “Core-Distribution-Access” with extended layer 2
- **Frequent operational activities:**
    - Changes
    - Troubleshooting
    - New applications

constraints

Possible Business Drives

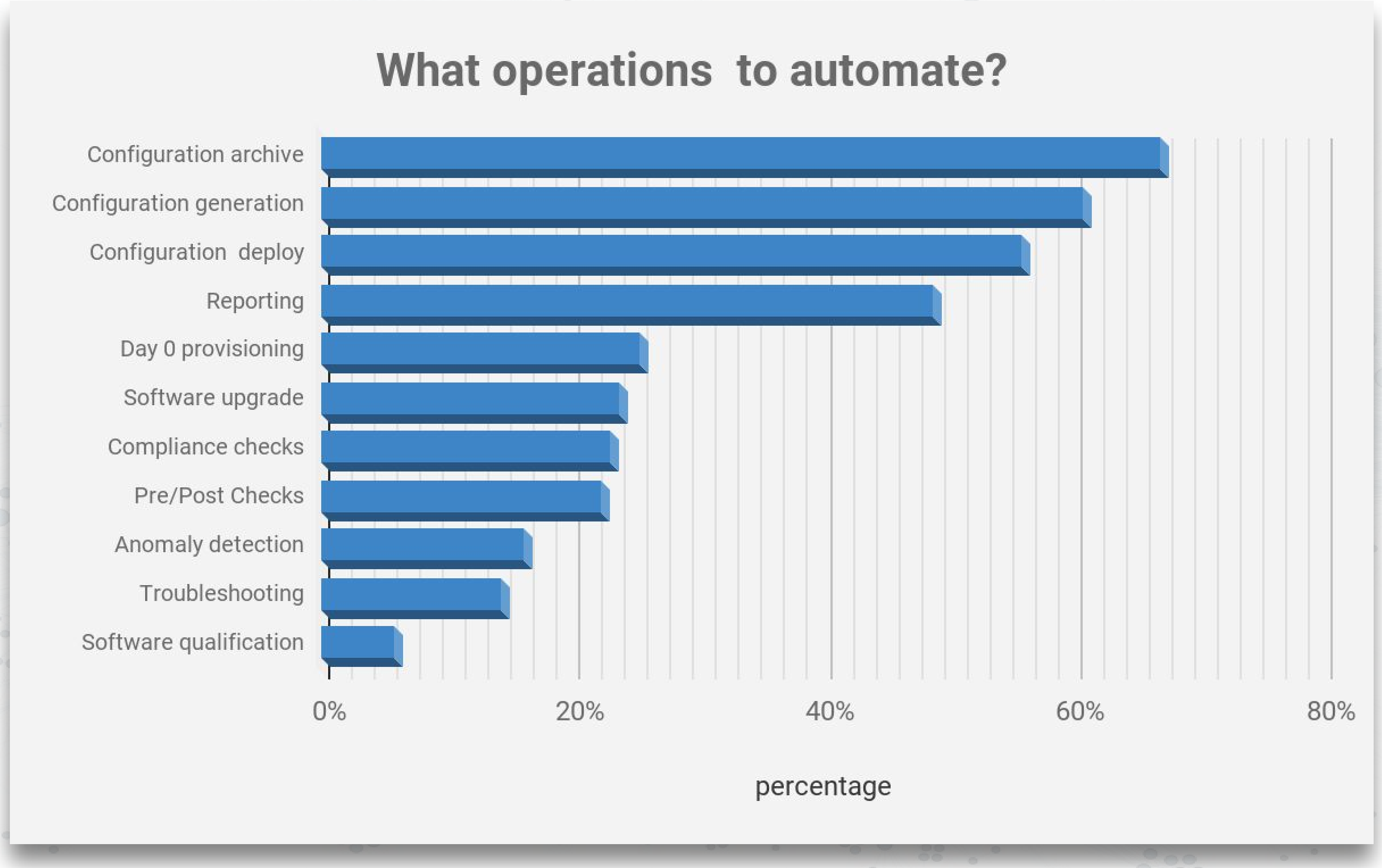
Pressing Needs:

- **Scale**: increasing environment size, scale and diversity
- **Optimization**: more to do with less engineers
- Failure management: solve new problems, guarantee level of performance and **consistency**.

Opportunities:

- 4IR: fourth Industrial revolution
  - IoT
  - Big Data
  - Digital transformations and cloud initiatives
  - Artificial intelligence and machine learning
  - Services “e.g. microservices” demand growing infrastructure and hence network scale

What to Automate?



https://docs.google.com/forms/d/e/1FAIpQLSdiBNMK0ZUmgBSNEaOWa-YHGQ4AlZo7EhB52_dXzvMqic3eHA/viewanalytics

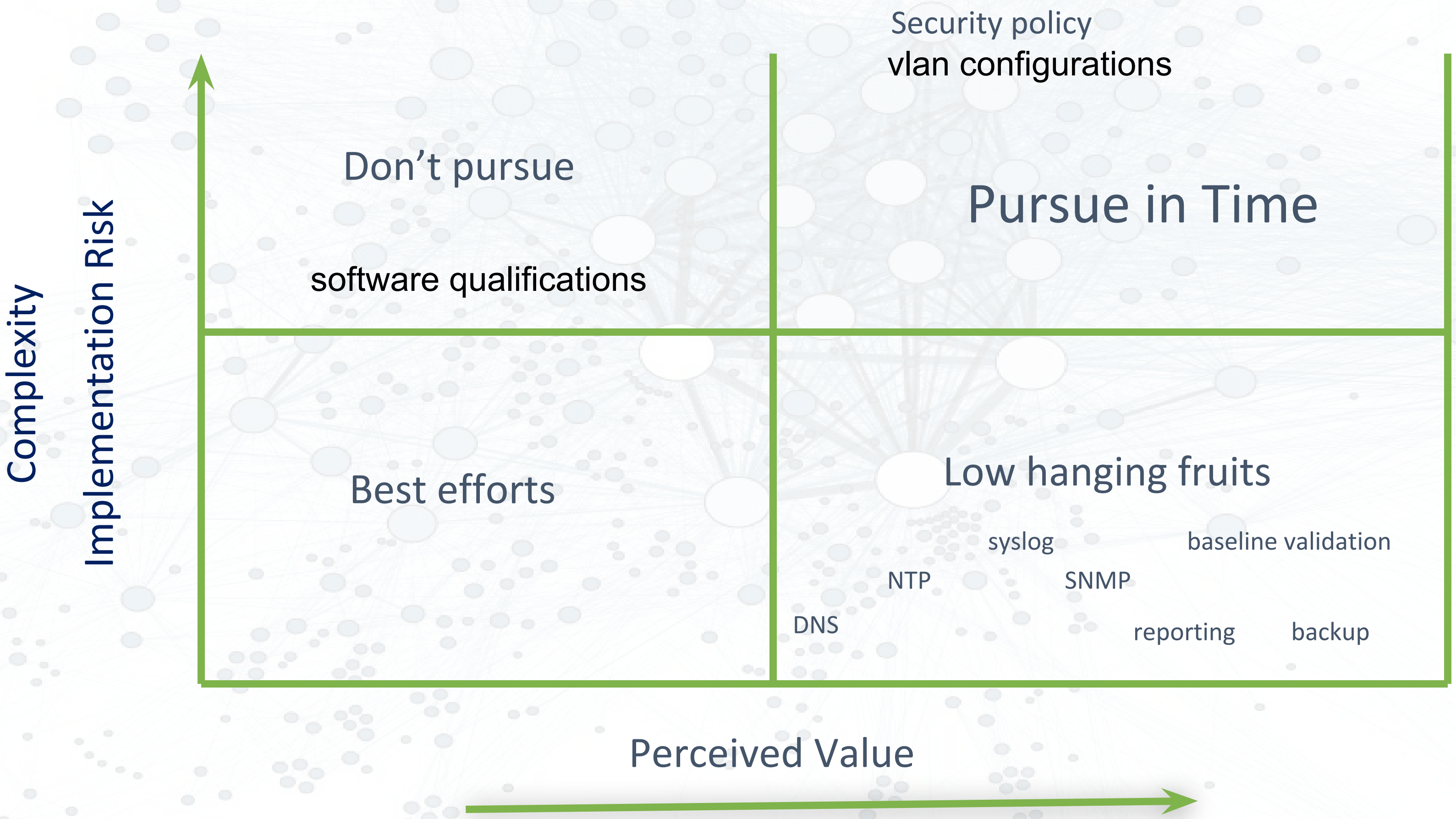
Top traditional network issues

Spanning Tree misconfigurations

VLAN(s) misconfiguration

and more

Priorities



Complexity

Implementation Risk

Don’t pursue

software qualifications

Security policy

vlan configurations

Pursue in Time

Best efforts

Low hanging fruits

syslog

baseline validation

NTP

SNMP

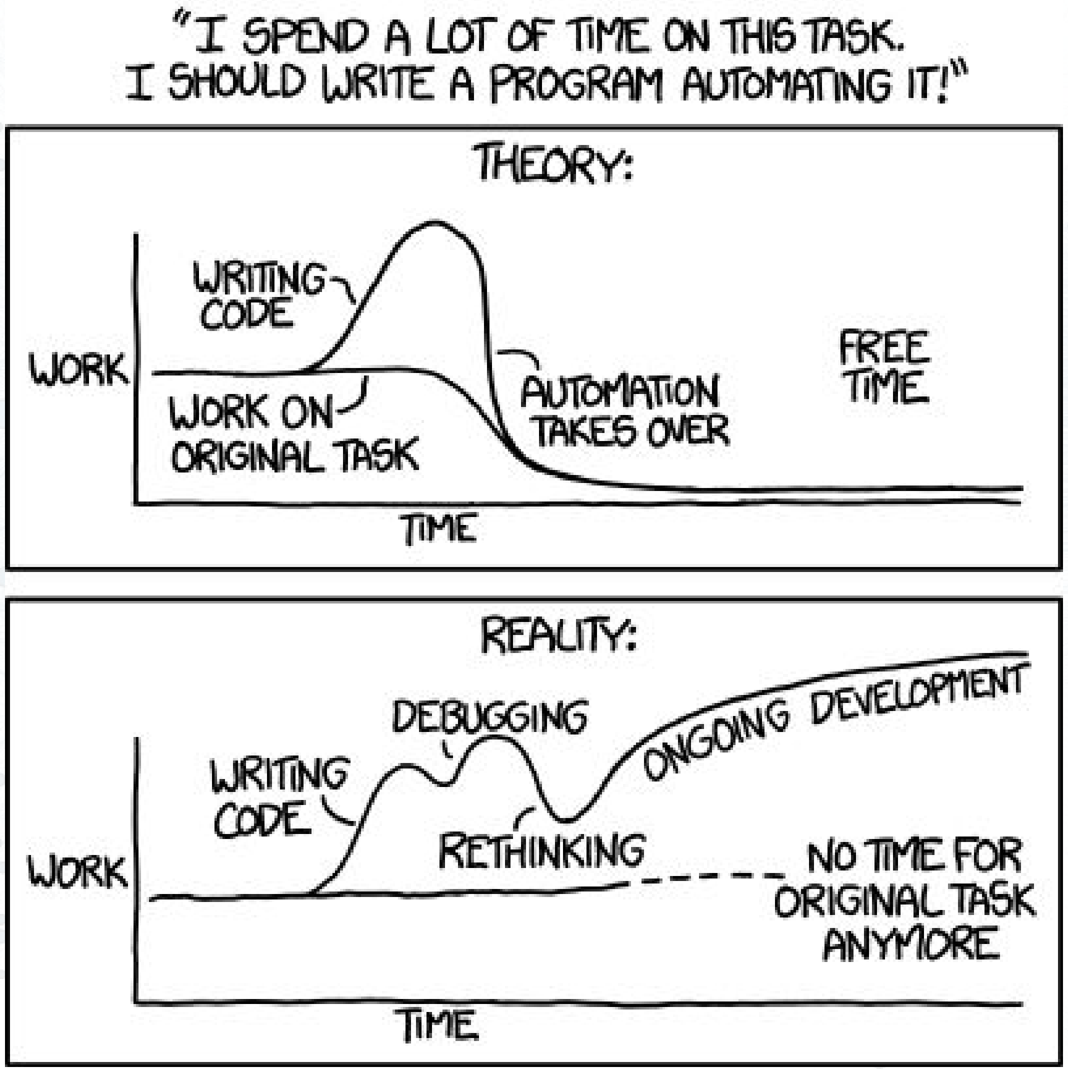
DNS

reporting

backup

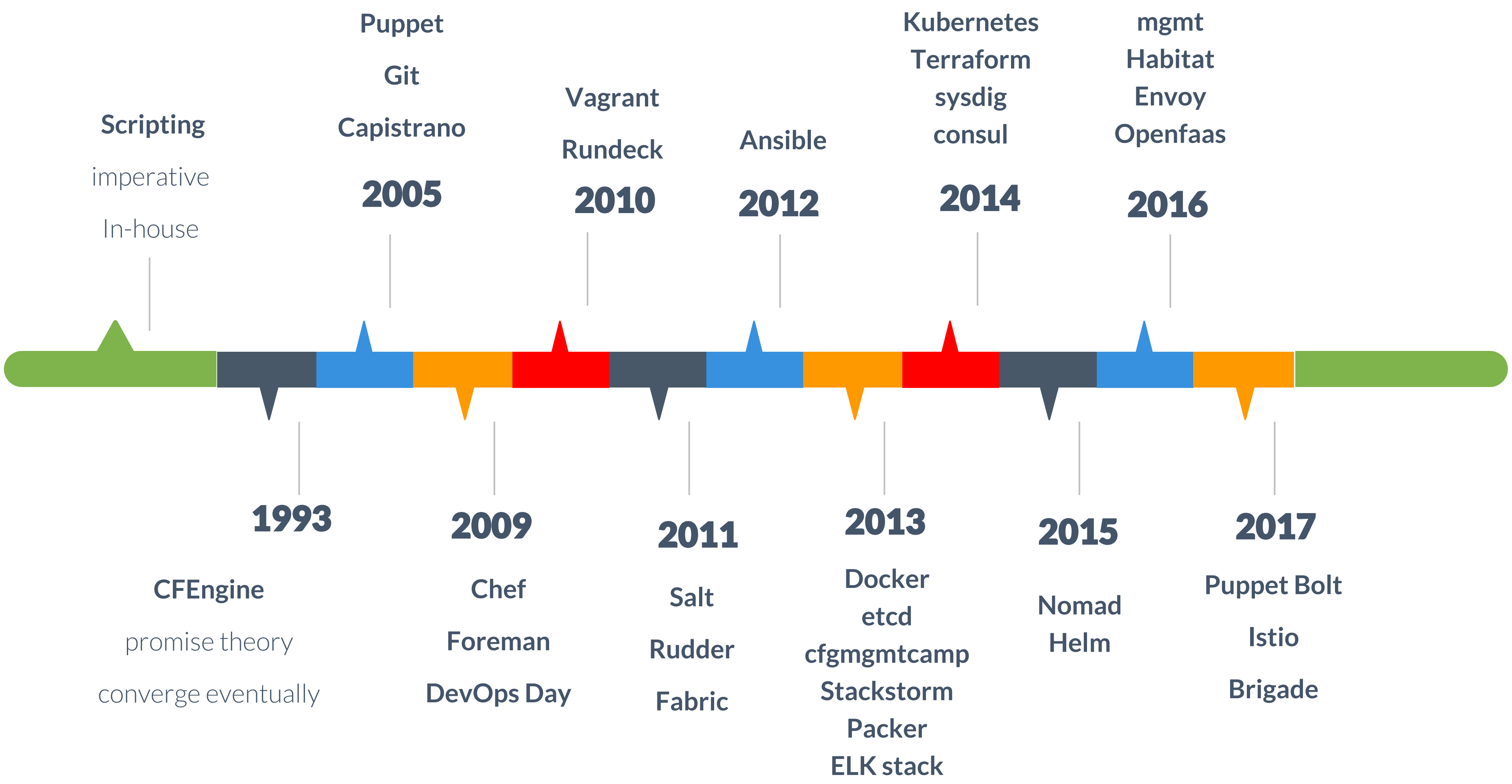
Perceived Value

What Not to Automate?



https://imgs.xkcd.com/comics/automation.png

Application and system automation



Scripting

imperative

In-house

1993

CFEngine

promise theory

converge eventually

2005

Puppet

Git

Capistrano

2009

Chef

Foreman

DevOps Day

2010

Vagrant

Rundeck

2011

Salt

Rudder

Fabric

2012

Ansible

2013

Docker
etcd
cfgmgmtcamp
Stackstorm
Packer
ELK stack

2014

Kubernetes
Terraform
sysdig
consul

2015

Nomad
Helm

2016

mgmt
Habitat
Envoy
Openfaas

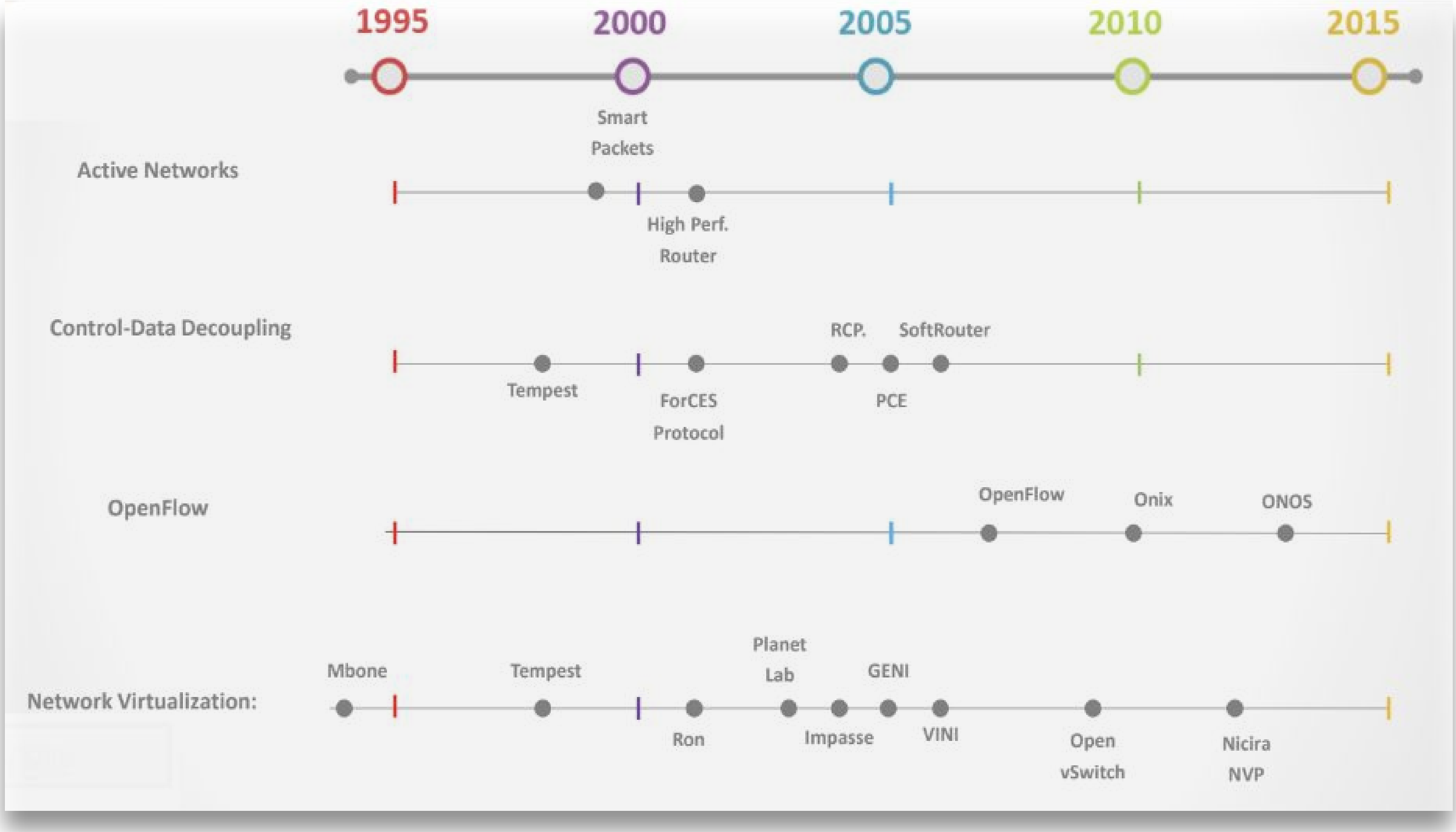
2017

Puppet Bolt

Istio

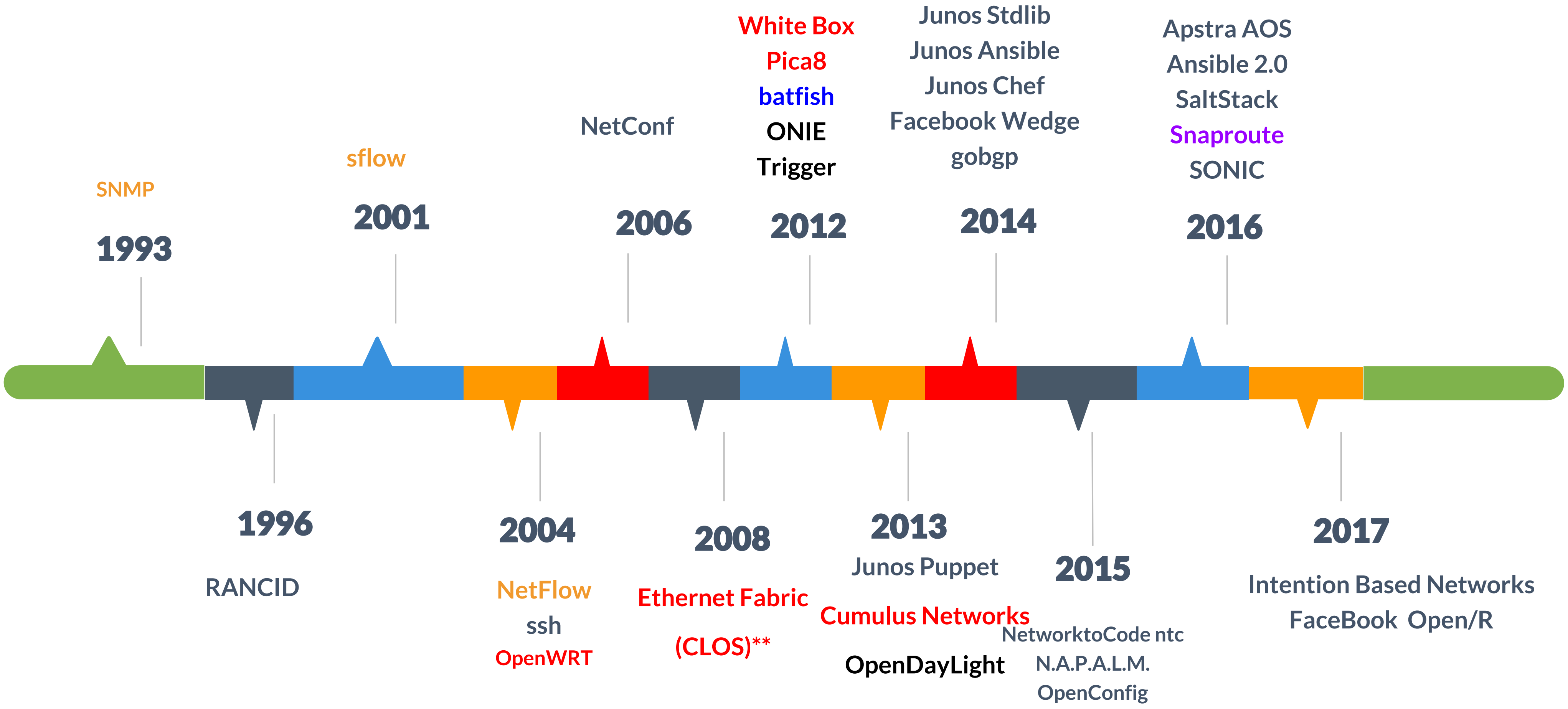
Brigade

History of Programmable Network



The Road to SDN: https://queue.acm.org/detail.cfm?id=2560327

The network world perspective



1993
SNMP

1996
RANCID

2001
sflow

2004
NetFlow
ssh
OpenWRT

2006
NetConf

2008
Ethernet Fabric
(CLOS)**

2012
White Box
Pica8
batfish
ONIE
Trigger

2013
Junos Puppet
Cumulus Networks
OpenDayLight

2014
Junos Stdlib
Junos Ansible
Junos Chef
Facebook Wedge
gobgp

2015
NetworktoCode ntc
N.A.P.A.L.M.
OpenConfig

2016
Apstra AOS
Ansible 2.0
SaltStack
Snaproute
SONIC

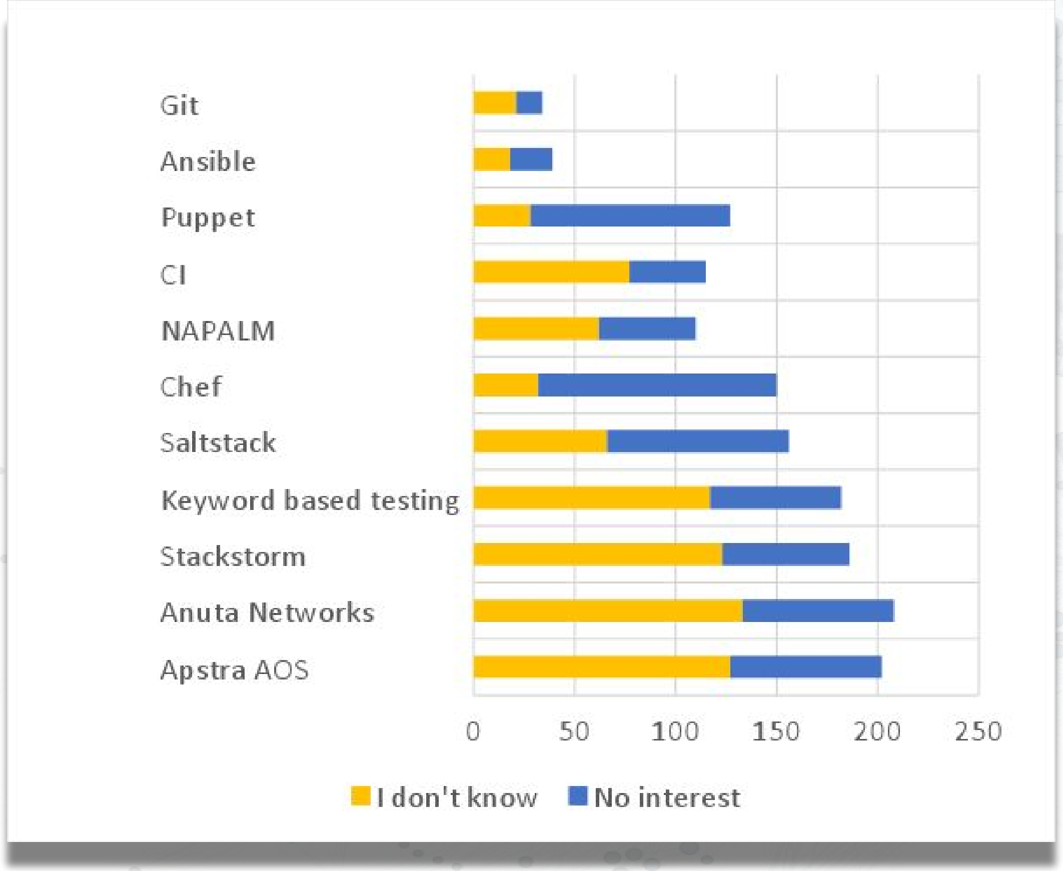
2017
Intention Based Networks
FaceBook Open/R

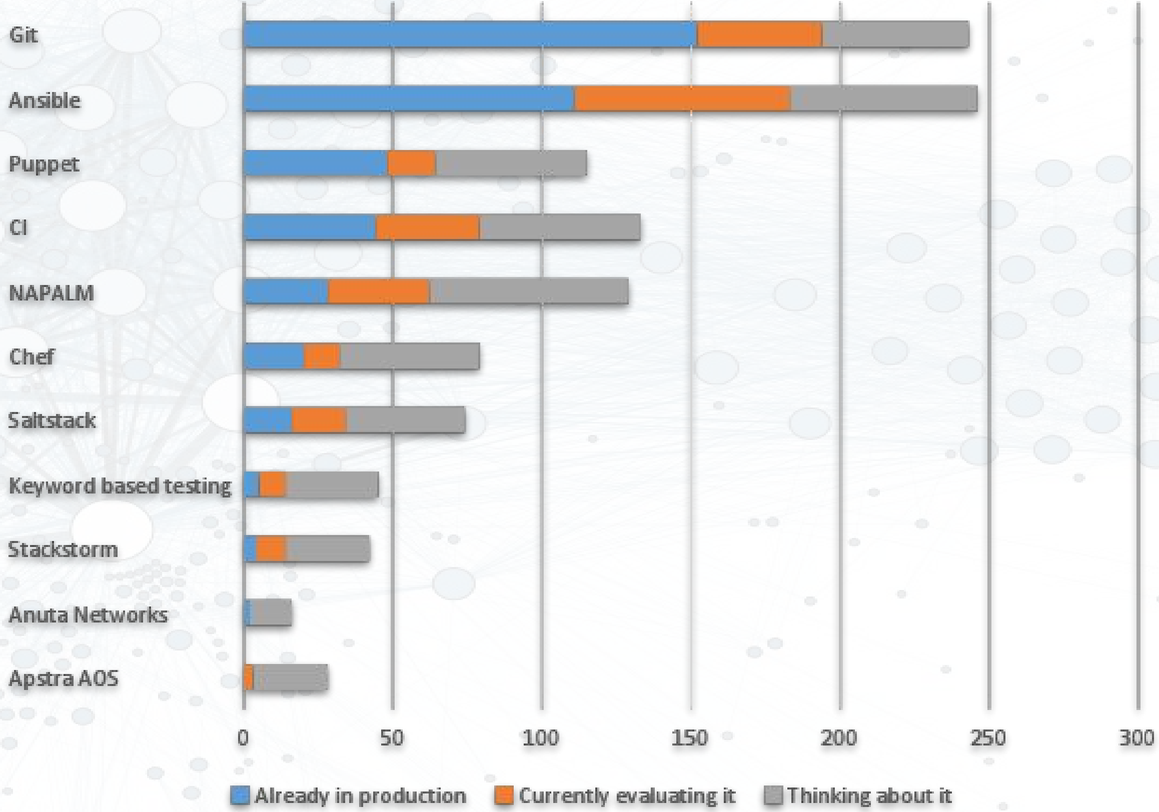
* Dates may be inaccurate. they were collected from initial release of standard, commit, or project info and other talks
** Research has roots back from 1953

Challenge 1: The Sandbox

vrnetlab

Technology: Tooling





Ansible! however,

From the book **“Automating Junos Administration”**
by Jonathan Looney and Stacy Smith:

> “Ansible configurations can grow to become somewhat complex. There are multiple files for inventory, variables, playbooks, and roles. Like with any critical system, it’s a good idea to keep all of these files under a revision control system such as **Git.** You may also want to couple revision control with a review and testing process to ensure any changes to the Ansible configurations are thoroughly verified before applying them to a production network.”

In other Words

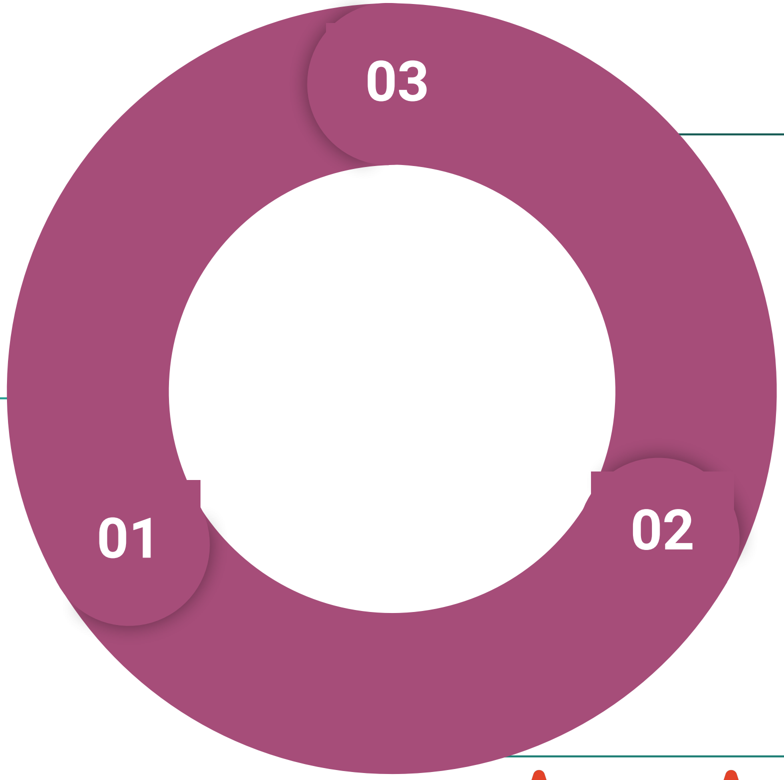
- Collaboration
- Version control
- How to manage scale and growth
- Testing

Small wins

- Do what you do everyday
  However try to improve one thing a time
  Don’t try to learn more than one thing at a time
  Automation will not stand in your way

- Think and do simple:
  - Start simple use cases
  - Stay simple handling generic cases

Process & Technology
KISS Workflow



The Modern CLI

- syntax highlighting
- Validation, linting, indentation
- the Automation UX

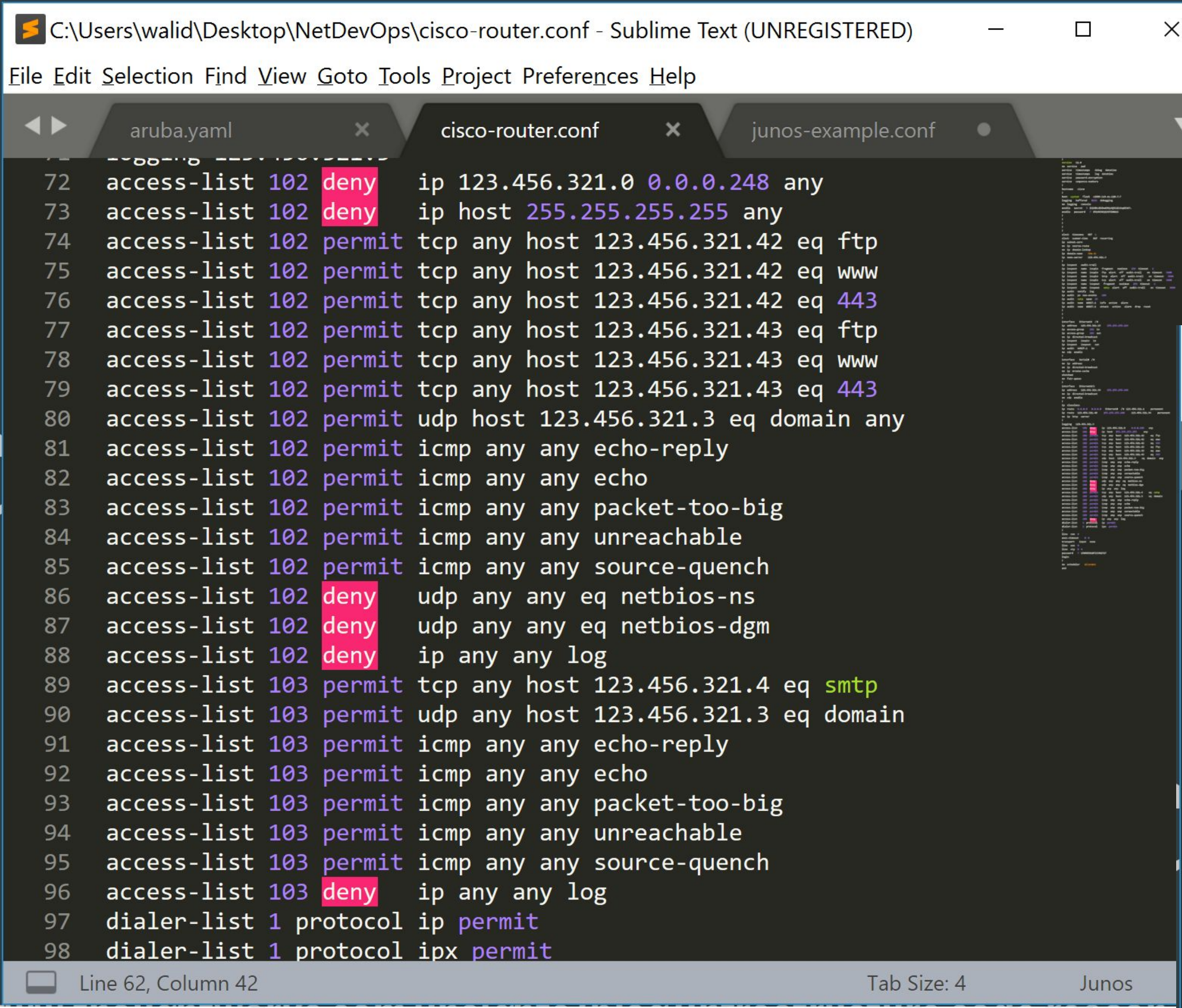
Automation Platform

- Configuration management
- Orchestration
- Role Based Access
- Scheduling
- Remote Execution
- Event based triggers

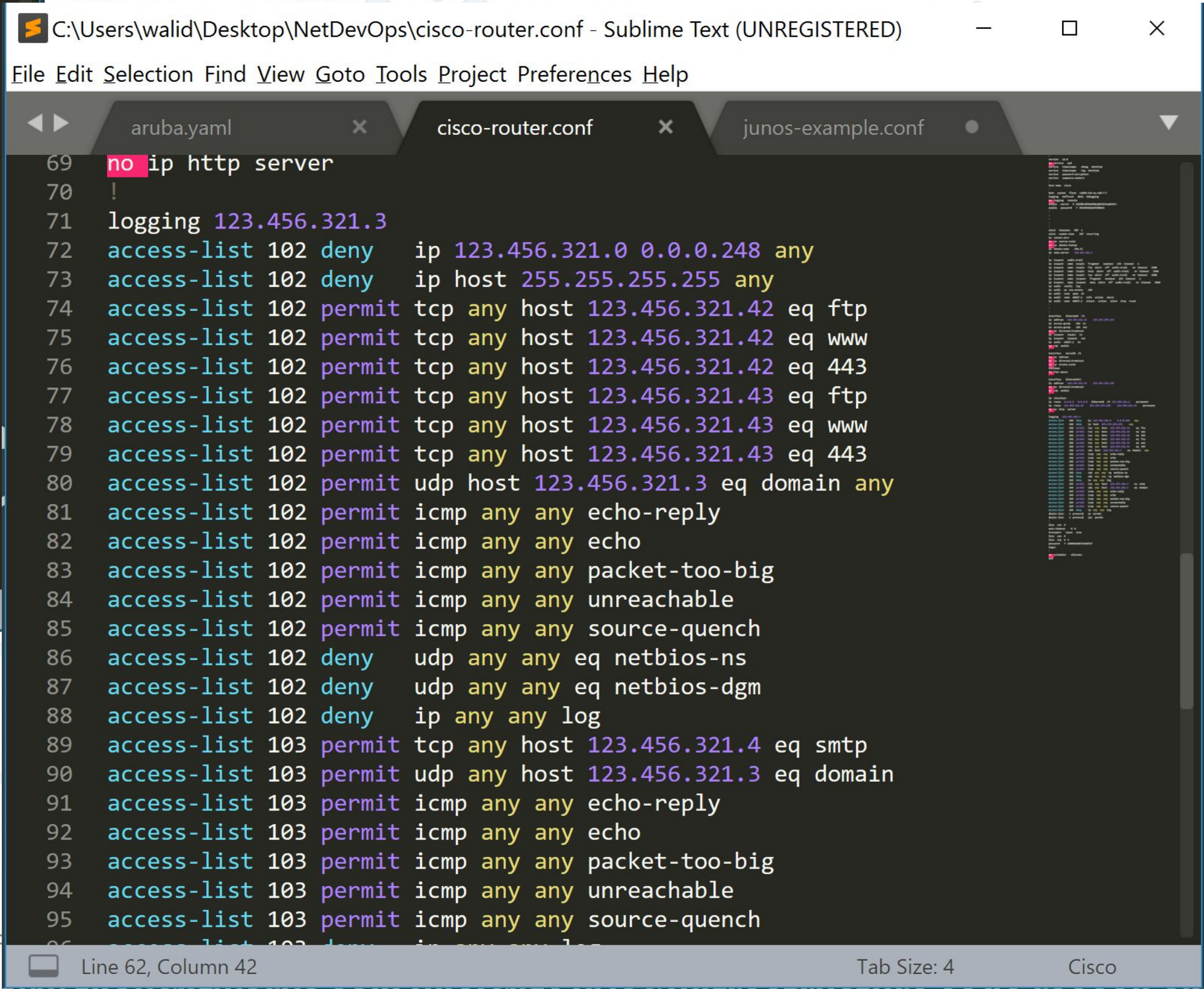
Version Control System

- History tracking
- Peer review
- Collaboration Engine
- Live Documentation
- Integration with issue tracker

Start small and simple


C:\Users\walid\Desktop\NetDevOps\cisco-router.conf - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
aruba.yaml
cisco-router.conf
junos-example.conf
Line 62, Column 42
Tab Size: 4
Junos

```
access-list 102 deny   ip 123.456.321.0 0.0.0.248 any
access-list 102 deny   ip host 255.255.255.255 any
access-list 102 permit tcp any host 123.456.321.42 eq ftp
access-list 102 permit tcp any host 123.456.321.42 eq www
access-list 102 permit tcp any host 123.456.321.42 eq 443
access-list 102 permit tcp any host 123.456.321.43 eq ftp
access-list 102 permit tcp any host 123.456.321.43 eq www
access-list 102 permit tcp any host 123.456.321.43 eq 443
access-list 102 permit udp host 123.456.321.3 eq domain any
access-list 102 permit icmp any any echo-reply
access-list 102 permit icmp any any echo
access-list 102 permit icmp any any packet-too-big
access-list 102 permit icmp any any unreachable
access-list 102 permit icmp any any source-quench
access-list 102 deny   udp any any eq netbios-ns
access-list 102 deny   udp any any eq netbios-dgm
access-list 102 deny   ip any any log
access-list 103 permit tcp any host 123.456.321.4 eq smtp
access-list 103 permit udp any host 123.456.321.3 eq domain
access-list 103 permit icmp any any echo-reply
access-list 103 permit icmp any any echo
access-list 103 permit icmp any any packet-too-big
access-list 103 permit icmp any any unreachable
access-list 103 permit icmp any any source-quench
access-list 103 deny   ip any any log
dialer-list 1 protocol ip permit
dialer-list 1 protocol ipx permit
```


C:\Users\walid\Desktop\NetDevOps\cisco-router.conf - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
aruba.yaml
cisco-router.conf
junos-example.conf
Line 62, Column 42
Tab Size: 4
Cisco

```
no ip http server
!
logging 123.456.321.3
access-list 102 deny   ip 123.456.321.0 0.0.0.248 any
access-list 102 deny   ip host 255.255.255.255 any
access-list 102 permit tcp any host 123.456.321.42 eq ftp
access-list 102 permit tcp any host 123.456.321.42 eq www
access-list 102 permit tcp any host 123.456.321.42 eq 443
access-list 102 permit tcp any host 123.456.321.43 eq ftp
access-list 102 permit tcp any host 123.456.321.43 eq www
access-list 102 permit tcp any host 123.456.321.43 eq 443
access-list 102 permit udp host 123.456.321.3 eq domain any
access-list 102 permit icmp any any echo-reply
access-list 102 permit icmp any any echo
access-list 102 permit icmp any any packet-too-big
access-list 102 permit icmp any any unreachable
access-list 102 permit icmp any any source-quench
access-list 102 deny   udp any any eq netbios-ns
access-list 102 deny   udp any any eq netbios-dgm
access-list 102 deny   ip any any log
access-list 103 permit tcp any host 123.456.321.4 eq smtp
access-list 103 permit udp any host 123.456.321.3 eq domain
access-list 103 permit icmp any any echo-reply
access-list 103 permit icmp any any echo
access-list 103 permit icmp any any packet-too-big
access-list 103 permit icmp any any unreachable
access-list 103 permit icmp any any source-quench
```

Recommendations

- Favor open solutions over proprietary
- Gain yearly saving of over 25% in 2023
- Invest back the savings into the people
- PP-RPC
- Create cross functional assignments

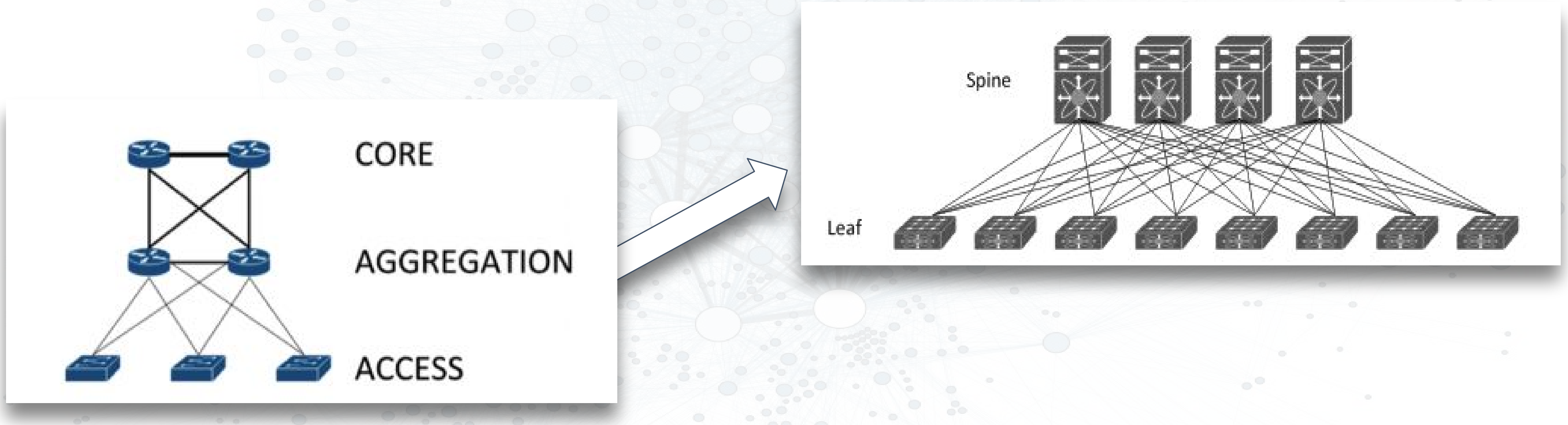
https://www.gartner.com/doc/3446727/time-shift-network-spend-premium

Think ahead
Devices, processes, and automation

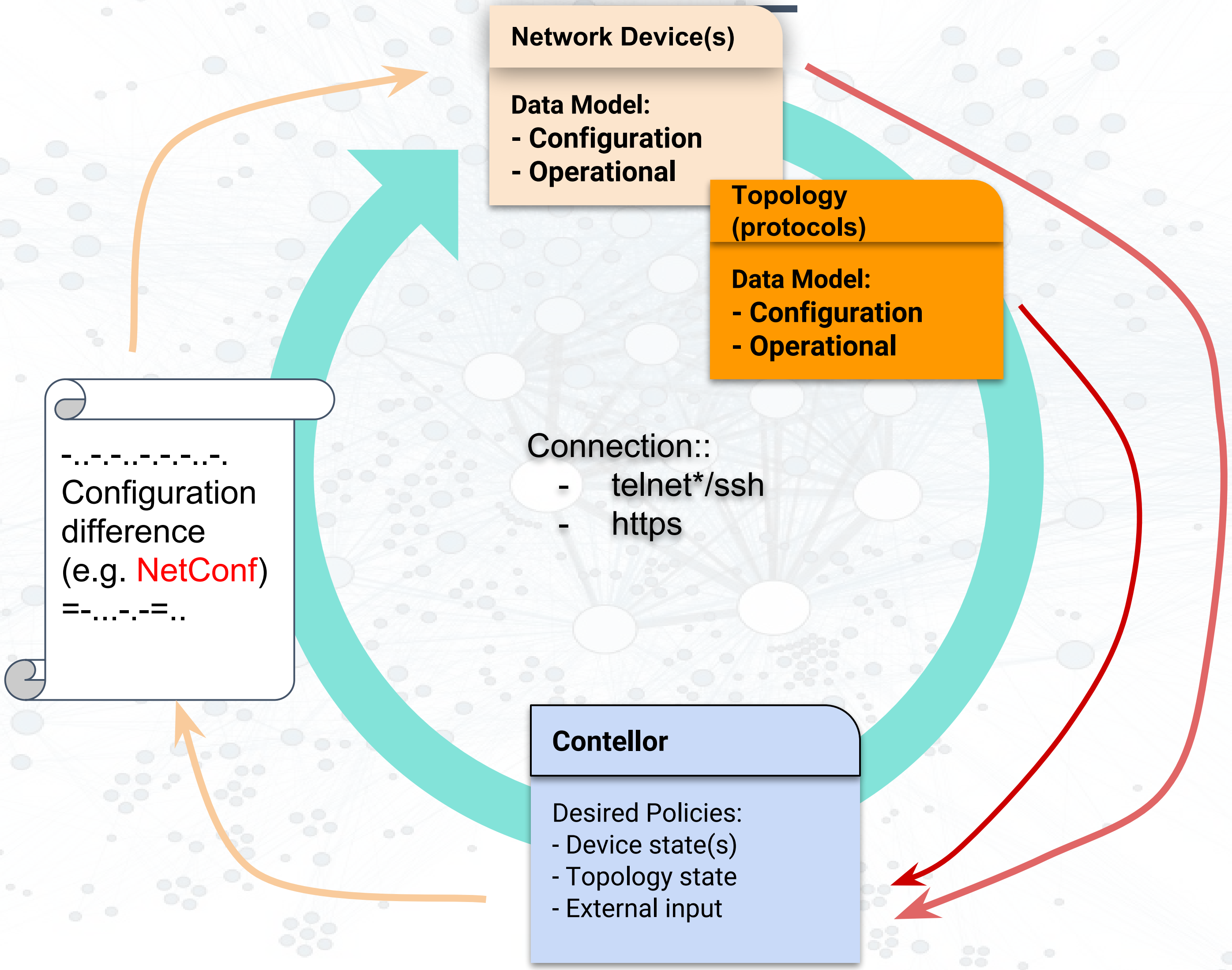
- Standardize network elements as much as possible
  - Have standardized configurations and processes
  - Avoid massive variations in vendors, platforms, and OS versions
  - Avoid massive variations in topologies and feature use (e.g. virtual router vs. zones)

- Insist on hardware that does have usable API (e.g. Netconf) and avoid to relying on screen-scraping for automation
- Hardware that does have good commit, rollback, and diff mechanism.
- Hardware that virtual images to be able to test and validate changes.

Topology, hardware matters



Configuration management 101



Network Device(s)

Data Model:
- Configuration
- Operational

Topology (protocols)

Data Model:
- Configuration
- Operational

-..-.-..-.-.-..-.
Configuration difference (e.g. NetConf)
=-...-.-=..

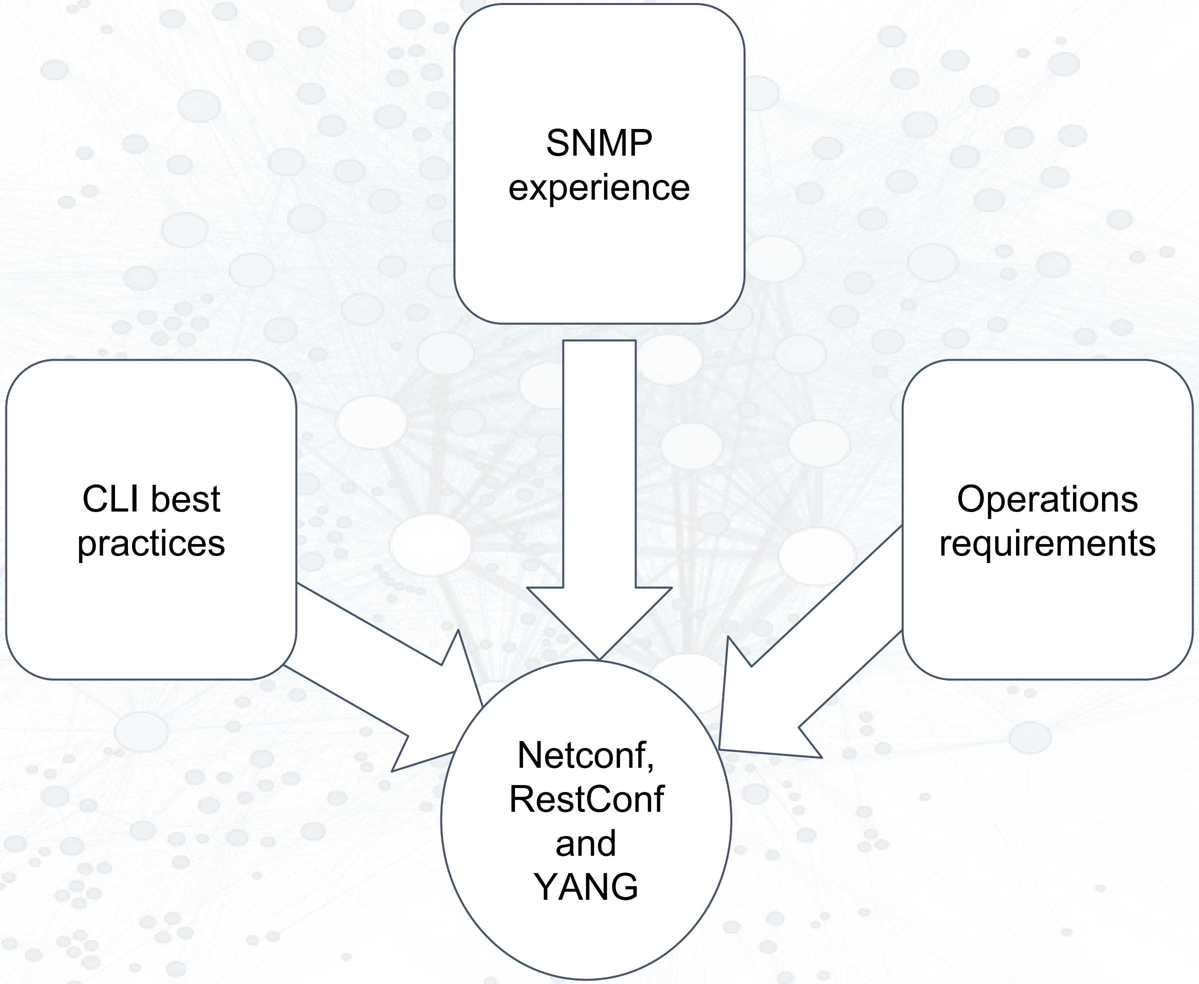
Connection::
- telnet*/ssh
- https

Contellor

Desired Policies:
- Device state(s)
- Topology state
- External input

State:
- Telemetry
- SNMP
- NetConf
- RestConf
- Syslog

IETF Programmability Strategy



SNMP experience

CLI best practices

Operations requirements

Netconf, RestConf and YANG

Charles Eckel, Cisco DevNet: https://fosdem.org/2018/schedule/event/opendaylight/

YANG

IETF Data Modeling language standard

- IETF standard defined in RFC 6020
- Data modeling language
  - Models Configuration and operational state data
  - Data Source of Truth
  - Easy to extend on existing models “**DRY**”
  - think of it as a database or xml schema definition XSD
  - if you are into kubernetes, custom resource types definitions
- Maintains compatibility with SNMP SMIv2
- A unified solution to the multi-vendor device data discrepancy
- *Not All vendors yet serious about it*

NetConf

IETF Network management protocol

- Defined in RFC 4741 (2006), updated by RFC 6241 (2011)
- Provides mechanisms to install, manipulate, and delete the configuration of network devices
- Model driven APIs
- Distinguishes between configuration and operational/state dat
- Multiple configuration datastores (candidate, running, startup)
- Configuration change validation and transactions
- Selective data retrieval via filtering
- Streaming and playback of event notifications

RestConf

- IETF RFC 8040
- Configuration data and state data exposed as resources
- How to access the data using REST verbs (GET / PUT / POST/ ...)
- How to construct URIs to access the data
- HTTP instead of SSH for transport
- JSON in addition to XML for data encoding

YANG: Yet Another Next Generation

Data model language

  used to model Configuration and operational state data

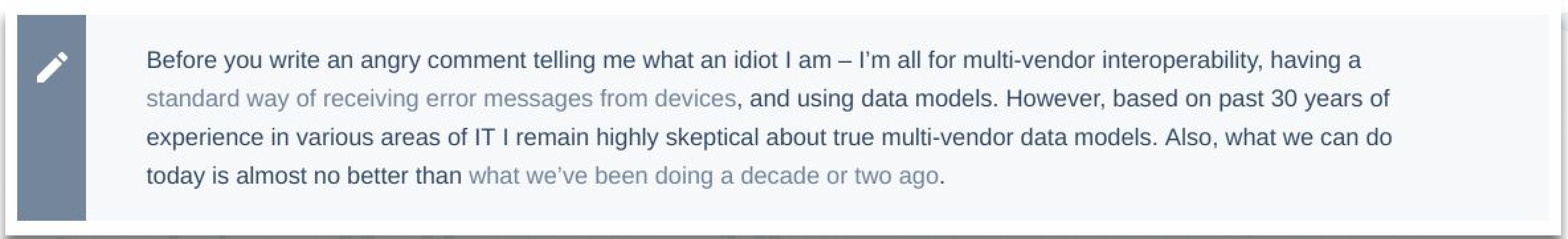
  easy to extend on existing models

IETF standard defined in RFC 6020

Data model language for both Configuration and operational state data

A solution to the multi-vendor device data discrepancy

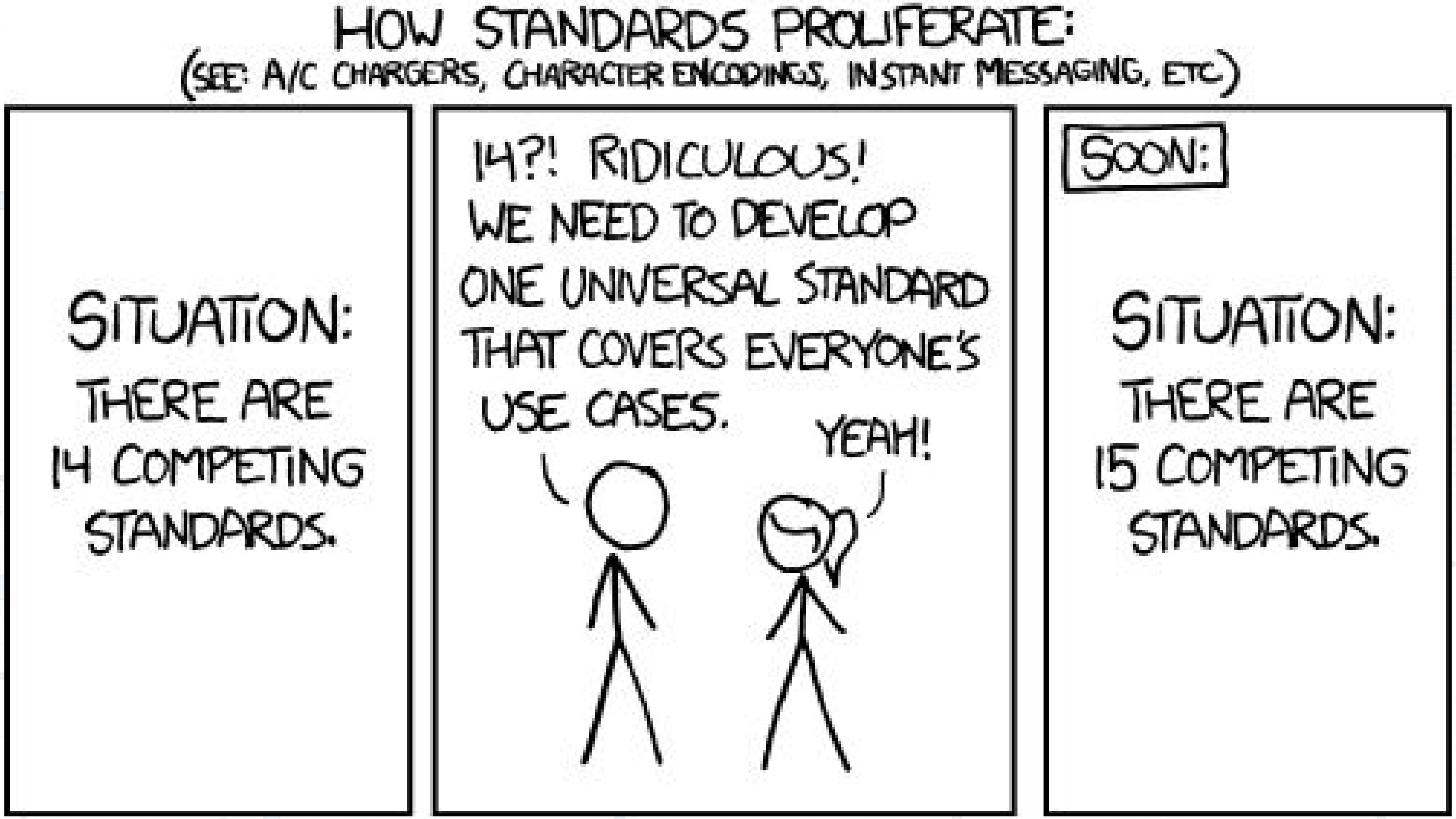
Not All vendors yet serious about it



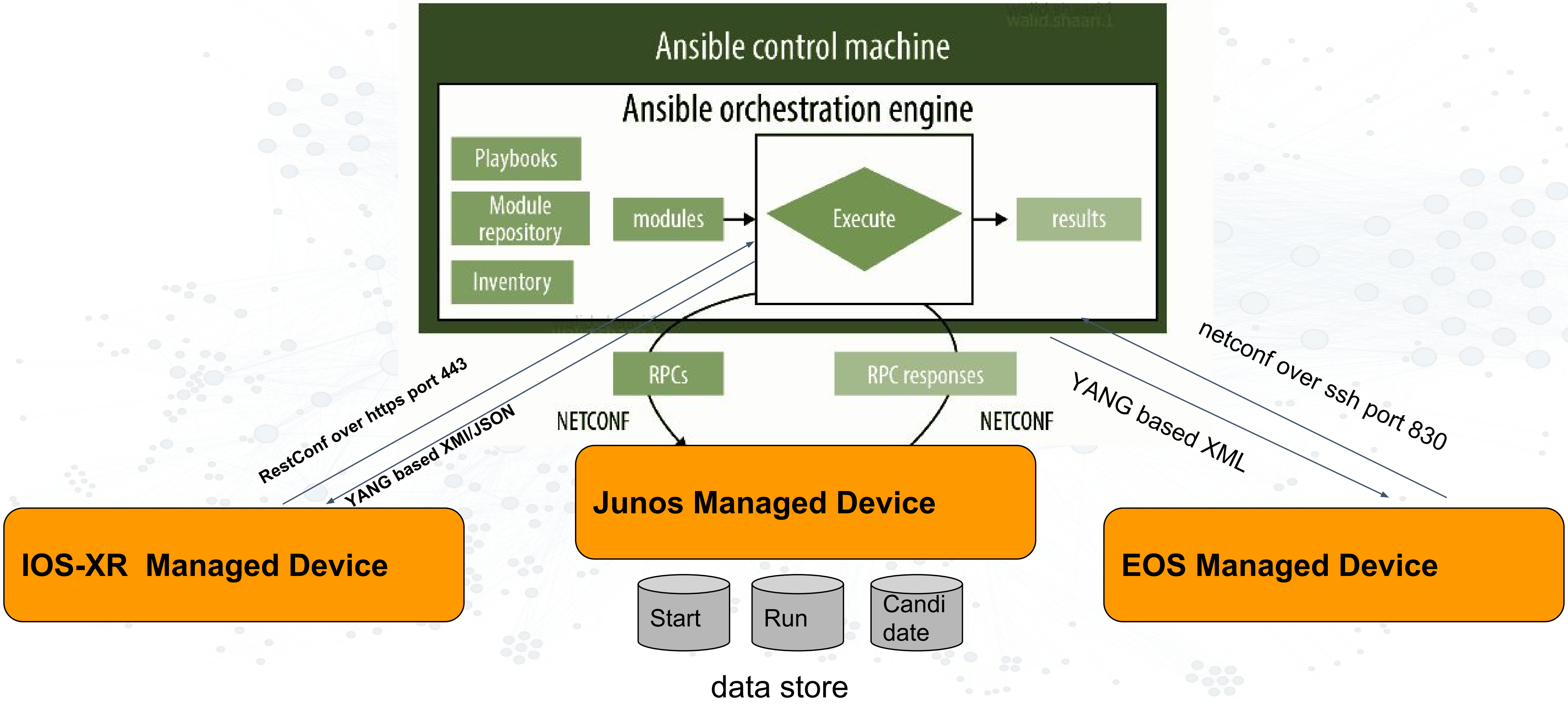
Before you write an angry comment telling me what an idiot I am – I'm all for multi-vendor interoperability, having a standard way of receiving error messages from devices, and using data models. However, based on past 30 years of experience in various areas of IT I remain highly skeptical about true multi-vendor data models. Also, what we can do today is almost no better than what we've been doing a decade or two ago.

Ivan Pepelnjak   on Monday, January 29,2018 blogged::
http://blog.ipspace.net/2018/01/use-yang-data-models-to-configure.html

Takes time and effort to standardise


HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Ansible netconf module



Ansible control machine

Ansible orchestration engine

Playbooks

Module repository

Inventory

modules

Execute

results

RPCs

RPC responses

NETCONF

NETCONF

RestConf over https port 443

YANG based XMl/JSON

netconf over ssh port 830

YANG based XML

IOS-XR Managed Device

Junos Managed Device

EOS Managed Device

Start

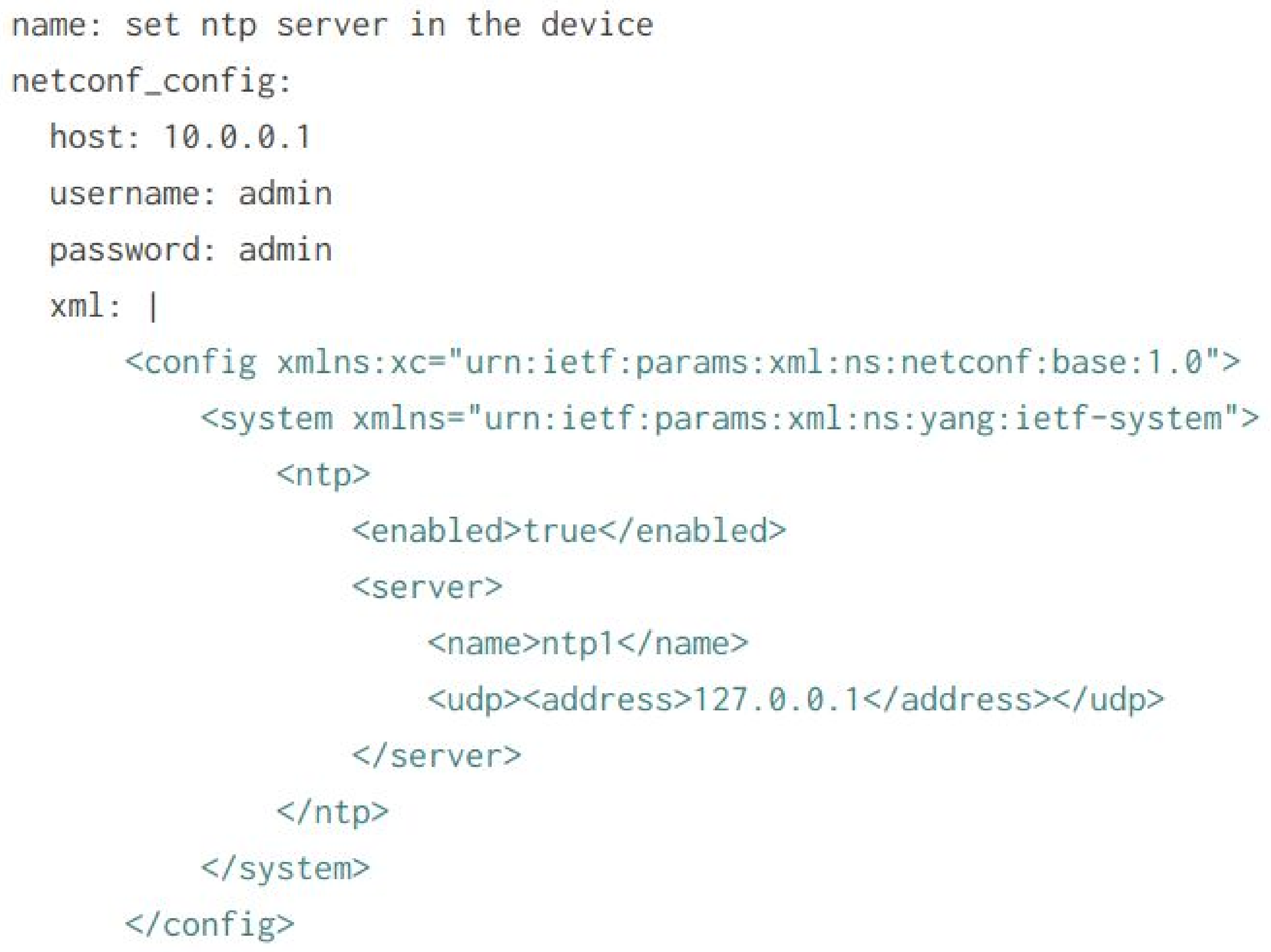
Run

Candi date

data store

Ansible netconf module

ntp server [vrf MGMT] 192.168.1.1



```
name: set ntp server in the device
netconf_config:
  host: 10.0.0.1
  username: admin
  password: admin
  xml: |
      <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.0">
          <system xmlns="urn:ietf:params:xml:ns:yang:ietf-system">
              <ntp>
                  <enabled>true</enabled>
                  <server>
                      <name>ntp1</name>
                      <udp><address>127.0.0.1</address></udp>
                  </server>
              </ntp>
          </system>
      </config>
```

Ansible vendor modules

Arista EOS

```
 -    name: set ntp server in the device
          eos_commands:
                        - “ntp server  {{ ntp_servername }}”
          host: {{ inventory_hostname }}
            username: admin
            password: admin
       register:
          eos_command_output
```

CISCO IOS

```
 -    name: set ntp server in the device
           ios_commands:
              - “ntp server {{ ntp_servername }}”
           host: {{ inventory_hostname }}
             username: admin
             password: admin
        register:
           ios_command_output
```

Juniper Junos

```
 -    name: set ntp server in the device
          junos_commands:
            - “set system ntp server  {{ ntp_servername }}”
          host: {{ inventory_hostname }}
            username: admin
            password: admin
       register:
          junos_command_output
```

To declare or not to declare vendor_config

- Playbooks becomes operating manuals.

  easy to understand and replicate in the CLI
- Gradual step toward declarative, declarative network modules coverage is not complete. e.g. Junos syslog
- favor the human interactive CLI over the cut & paste machine structures.

Other types of network modules

Ansible supported modules:

- netconf:
  - netconf_config
- vendor_config
  - ios_config
- vendor_cmmand
  - ios_command

Minimum Viable Platform Agnostic modules:

e.g. net_interface

Vendor/Community supported modules:

- netconf:
  - junos_netconf
  - ce_netconf
- Network to Code:
  - ntc_install_os
  - ntc_get_facts
- N.A.P.A.L.M:
  - napalm_diff_yang
  - napalm_get_facts

Custom built module: https://www.ansible.com/ansible-module-development-101

Hit Refresh

- *Continuous Improvements*
- *Review and Improve what has been done*
- *Improve one thing at a time*
- *Learn and review past work*
- *Document and prioritize new ~~problems~~ challenges*

Resources

Networktocode slack channel http://networktocode.herokuapp.com/

SDN & NFV: https://fosdem.org/2018/schedule/event/opendaylight/

Blogs:

- Csilla Bessenyei Networker and coder https://networkerandcoder.wordpress.com/
- Kirk Byers "Python for network engineers" https://pynet.twb-tech.com/
- Mircea Ulinic https://mirceaulinic.net
- Jason Edelman http://jedelman.com/
- David Lore http://ipengineer.net/
- netmiko https://github.com/ktbyers/netmiko
- Napalm https://napalm-automation.net/

Training:

- gns3 Academy http://academy.gns3.com/
- Ansible network automation examples: https://github.com/network-automation
- saltstack: https://docs.saltstack.com/en/develop/topics/network_automation/index.html

Net survey:

- https://docs.google.com/forms/d/e/1FAIpQLSdiBNMK0ZUmgBSNEaOWa-YHGQ4AlZo7EhB52_dXzvMqic3eHA/viewanalytics
- https://interestingtraffic.nl/2017/03/27/insights-from-the-netdevops-fall-2016-survey/

ipspace blog and podcast: https://www.ipspace.net
packetpushers podcast: http://packetpushers.net/

Thank you

The modern Network Engineer

Productive :

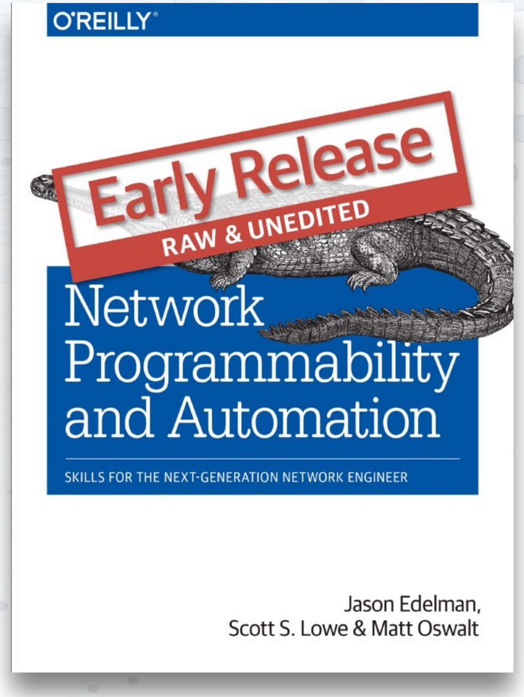
Less # lines of config manually?

Curious:

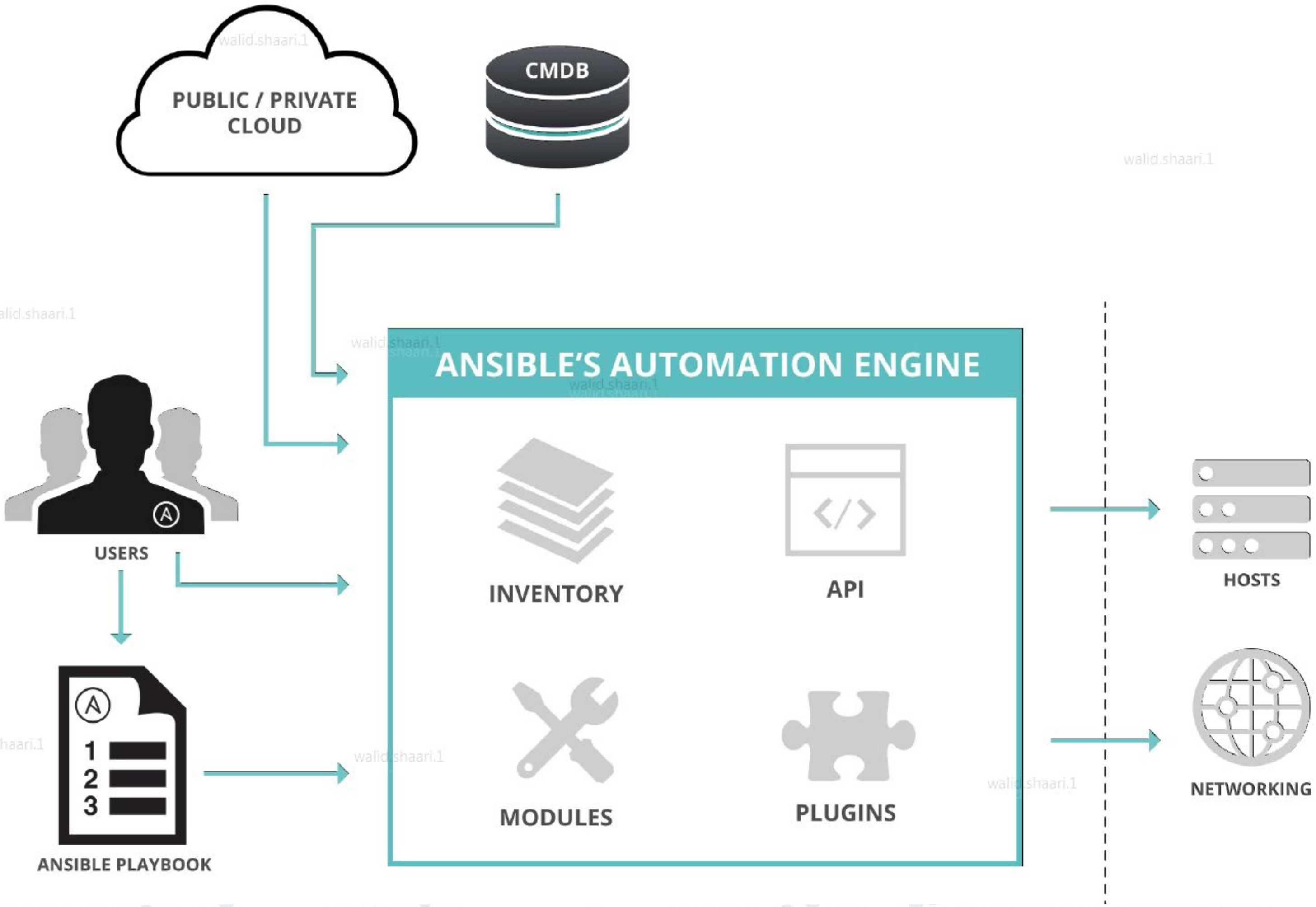
Interest in finding new problems to solve

Collaborates with other teams

developers, server and application support and peers


O'REILLY
Early Release
RAW & UNEDITED
Network Programmability and Automation
SKILLS FOR THE NEXT-GENERATION NETWORK ENGINEER
Jason Edelman,
Scott S. Lowe & Matt Oswalt

Ansible under the hoods



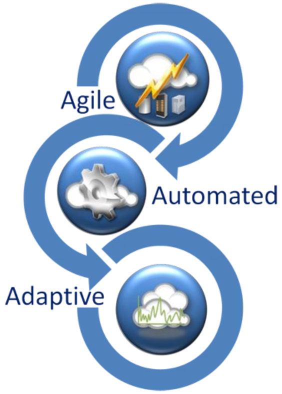
Initial Assignment Objective :

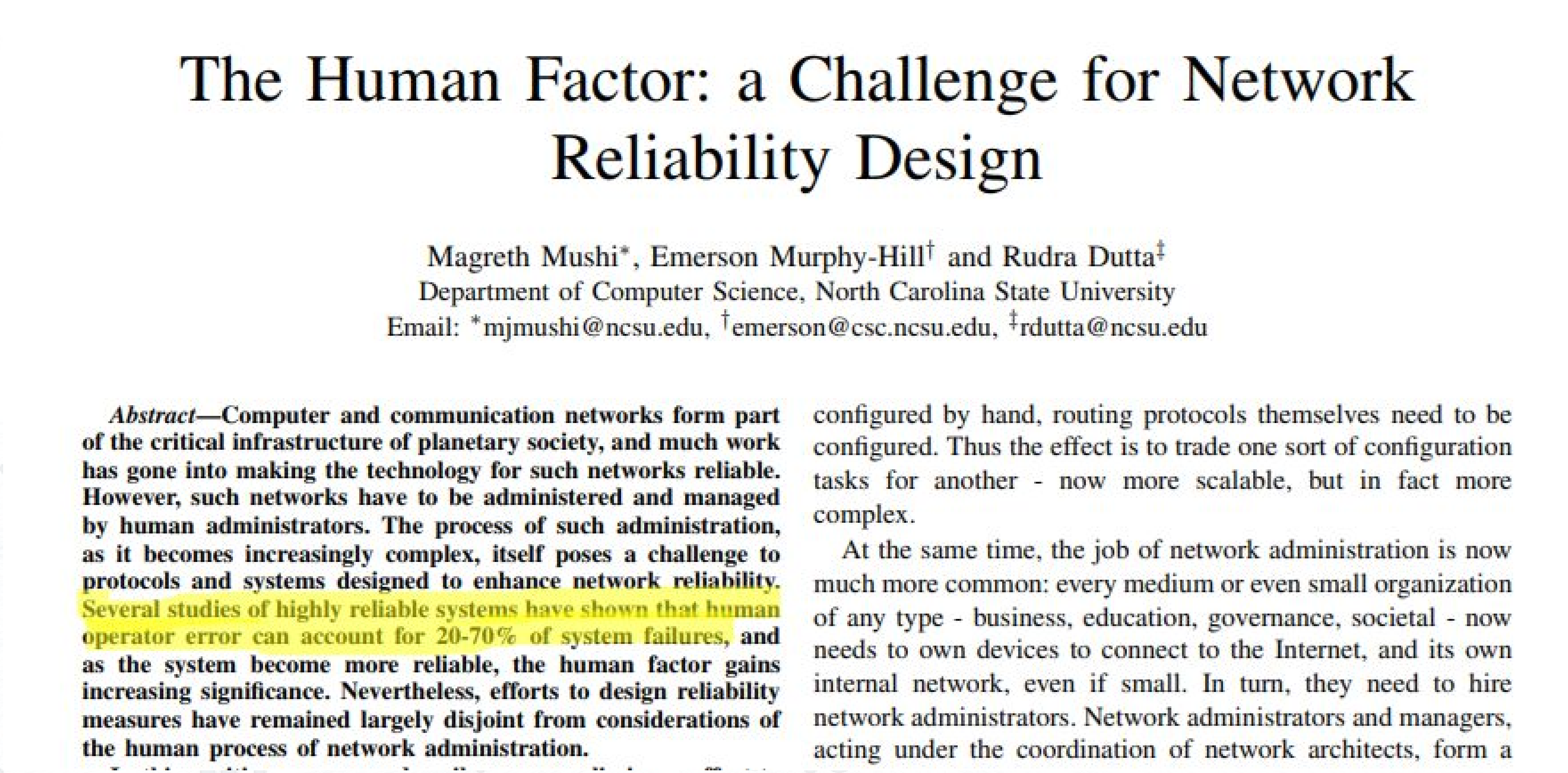
Deploy Ansible for network devices backup, and interface descriptions on Junos devices.

Involve only 3 out team of 9.

Post-assignment recommendation:

Continuous network infrastructure improvements through adopting NetDevOps culture and tooling to address future business requirements.





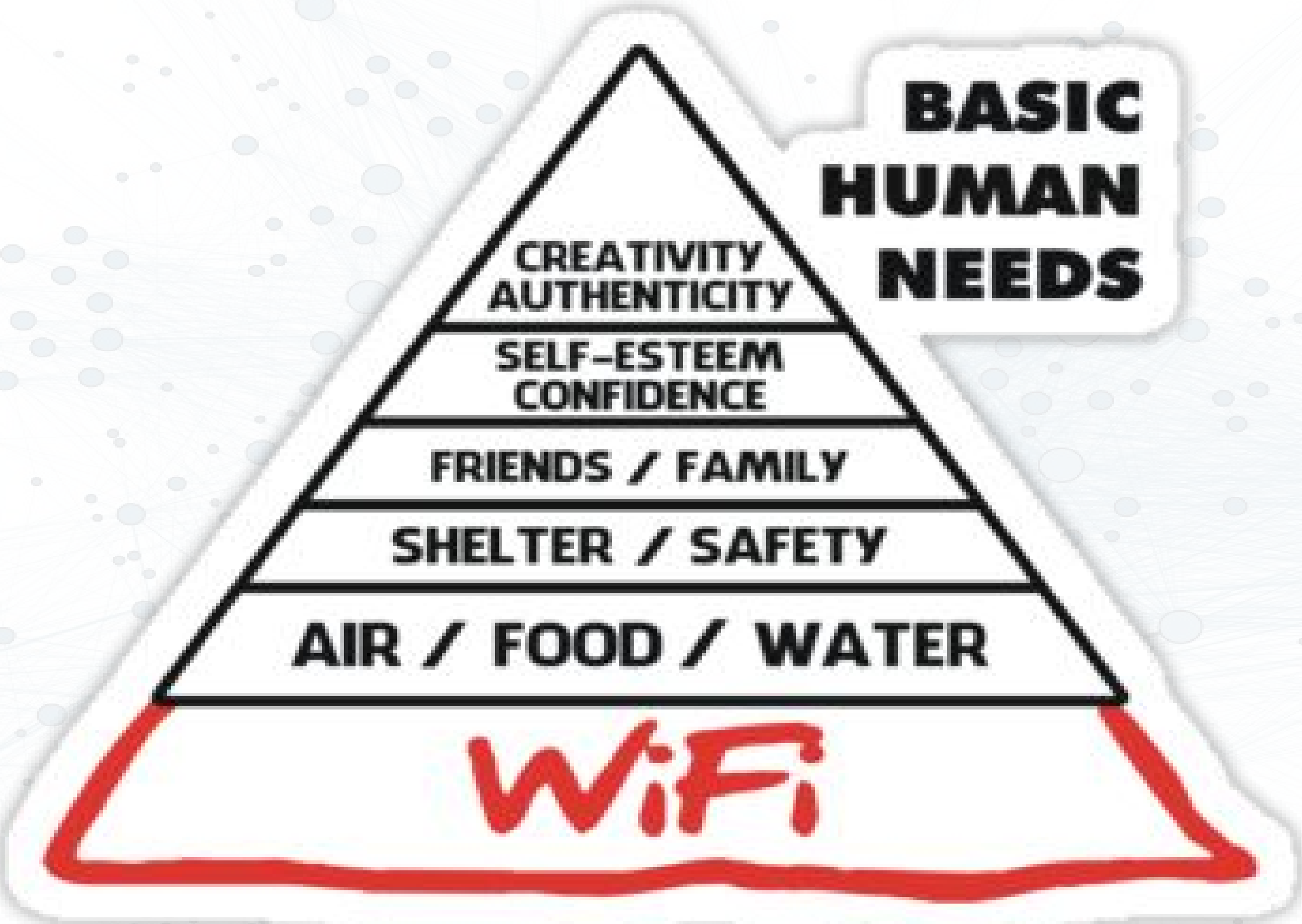
The Human Factor: a Challenge for Network Reliability Design

Magreth Mushi*, Emerson Murphy-Hill† and Rudra Dutta‡
Department of Computer Science, North Carolina State University
Email: *mjmushi@ncsu.edu, †emerson@csc.ncsu.edu, ‡rdutta@ncsu.edu

***Abstract*—Computer and communication networks form part of the critical infrastructure of planetary society, and much work has gone into making the technology for such networks reliable. However, such networks have to be administered and managed by human administrators. The process of such administration, as it becomes increasingly complex, itself poses a challenge to protocols and systems designed to enhance network reliability. Several studies of highly reliable systems have shown that human operator error can account for 20-70% of system failures, and as the system become more reliable, the human factor gains increasing significance. Nevertheless, efforts to design reliability measures have remained largely disjoint from considerations of the human process of network administration.**

configured by hand, routing protocols themselves need to be configured. Thus the effect is to trade one sort of configuration tasks for another - now more scalable, but in fact more complex.

At the same time, the job of network administration is now much more common: every medium or even small organization of any type - business, education, governance, societal - now needs to own devices to connect to the Internet, and its own internal network, even if small. In turn, they need to hire network administrators. Network administrators and managers, acting under the coordination of network architects, form a

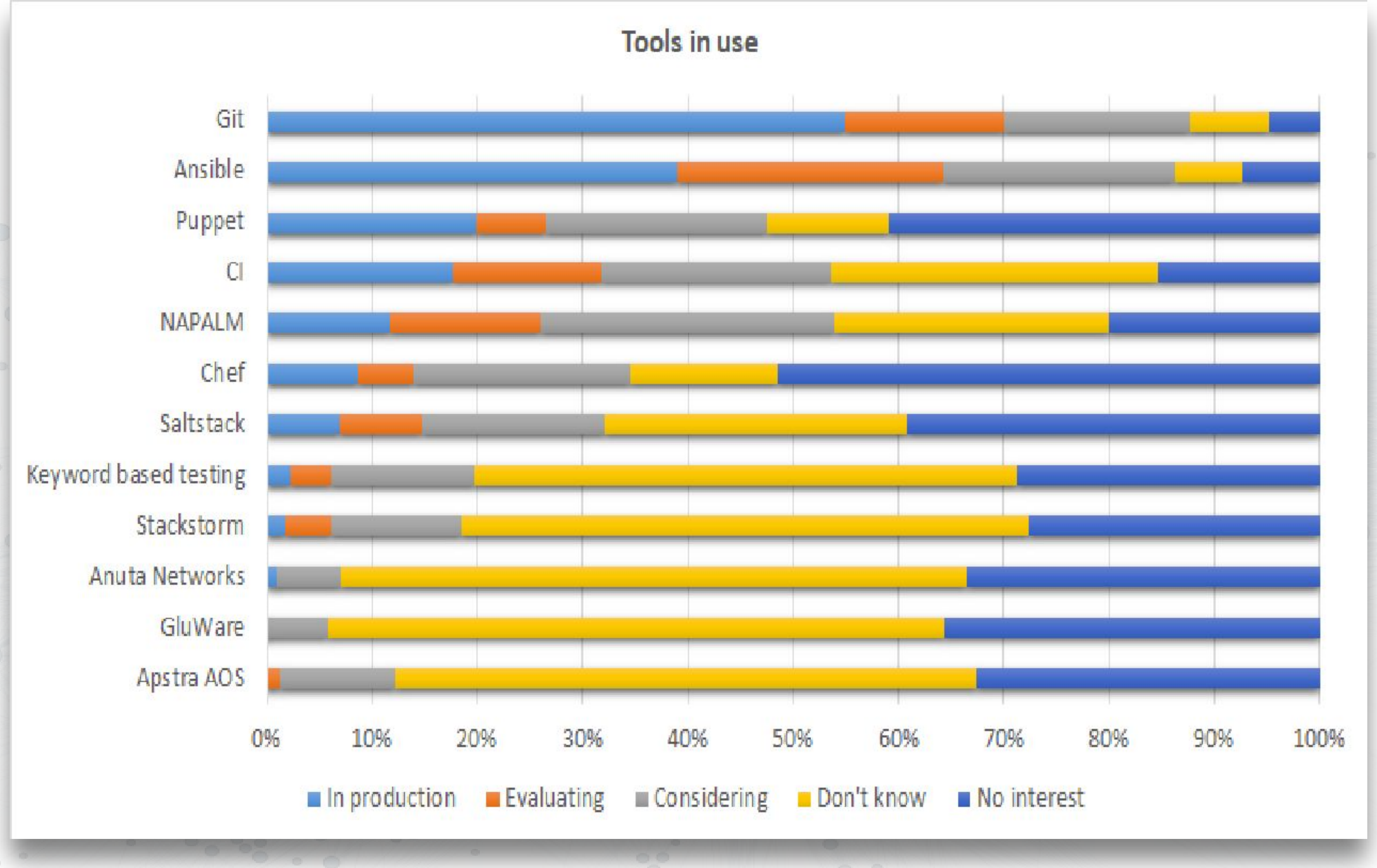


https://www.infoworld.com/article/3114195/system-management/the-8-fallacies-of-distributed-computing-are-becoming-irrelevant.html

Roles

- establish the main core modules and functionality
- roles:
    - automation engineer : writes new playbooks when needed, mostly will be updating data
    - operator: runs playbooks when necessary
    - reviewer: inputs or reviews data
        -

NetDevOps 2016 survey
Tools of Interest



https://interestingtraffic.nl/2017/03/27/insights-from-the-netdevops-fall-2016-survey/

How: Utilize current team knowledge

Improve upon the knowledge team already have on the networking side.

Add Ansible manifests to capture processes, repeatability, documentation, you should start automating one task at at time.

little bit of code and formatting from Ansible they are able to start automating from day one.