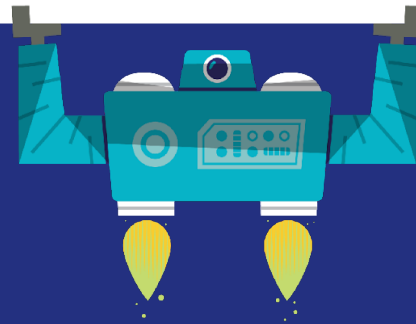


# DevOps for NetOps

Rick “Shermdog” Sherman  
Puppet Labs



# Introduction

~9.5 years - Juniper Networks

- Professional Services
  - Identity and Policy Management
  - Workflow systems
- Security Business Unit
  - Cloud Architect
- Junos Manageability
  - PyEZ (Python micro-framework)
  - Ansible Modules
  - Onbox scripting
  - NetDev Evangelism

~3.5 months - Puppet Labs

- Release Engineering
  - Network Platform Expansion



# Life of a Network Engineer

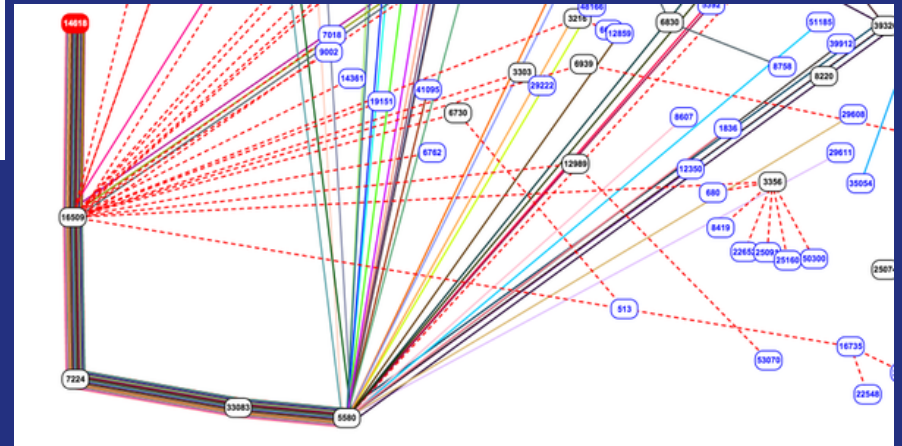
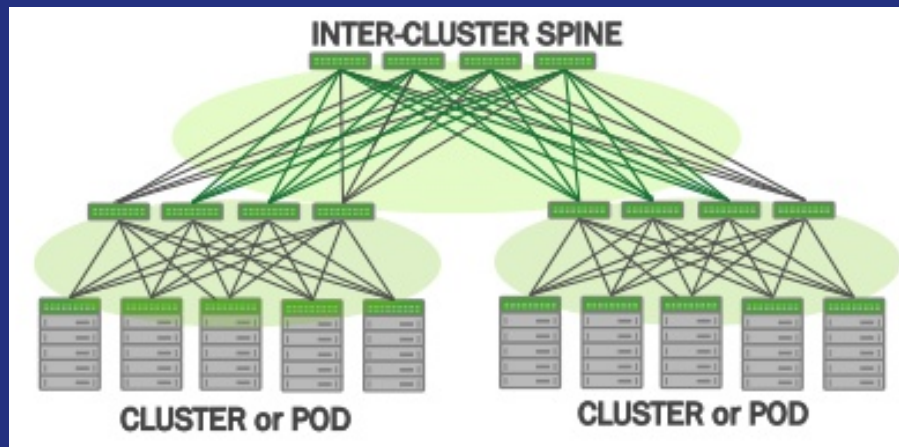
Let's make some generalizations (what could go wrong?)

- Networks are a complex ecosystem inter-connected devices
  - Services are spread over multiple systems
  - Equipment is often heterogeneous
- Require a lot of planning, testing, and validation
- A lot of time is spent fire fighting
- Also a lot of mundane tasks

# How does that differ from Sys Admins?

- Network devices have historically been closed systems with vendor specific CLIs
  - They often differ between the same vendor device types and versions
- Configurations are hundreds if not thousands of lines (per system)
- Configuration != Desired state
  - Often peering with other systems not under our control
- Vendors slow to introduce features, sometimes 18-24 months - upgrade cycle is just as long.
- Network Engineers typically do not have a Sys Admin or programming background

# Inter-tubes? More like spaghetti o.O



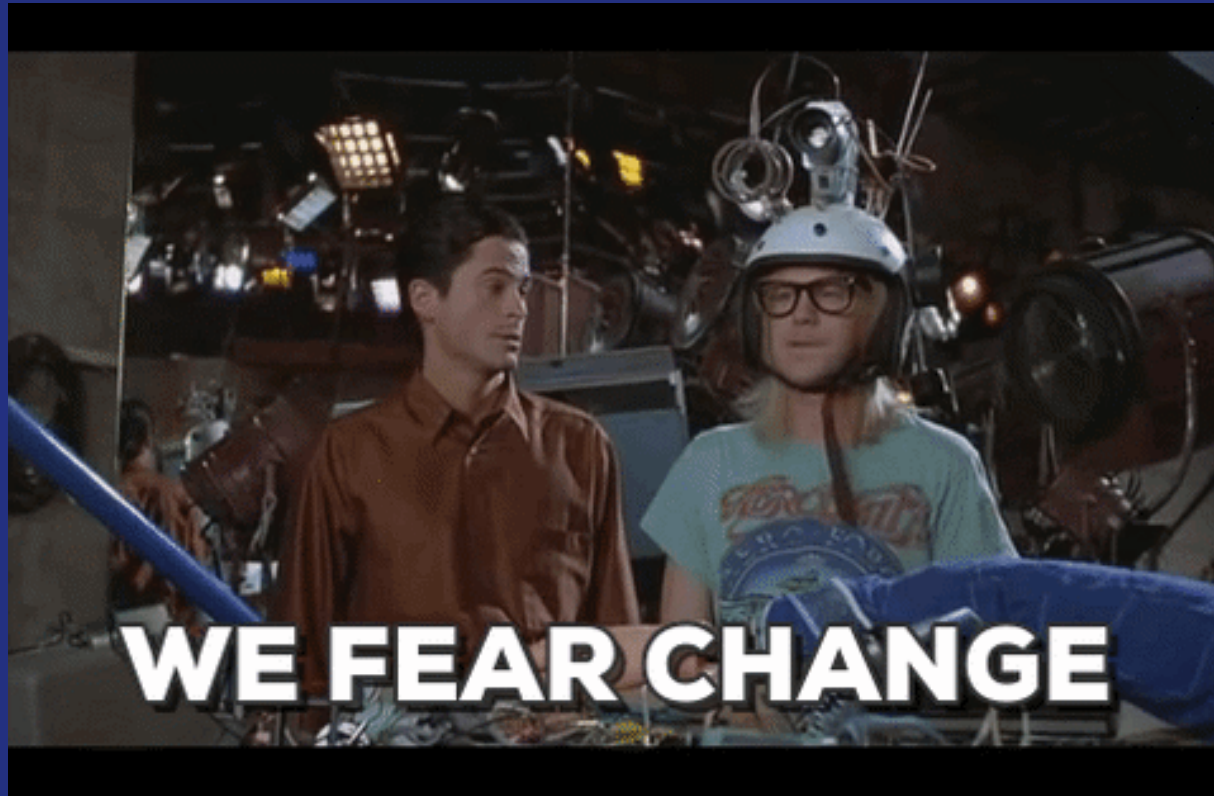
# Ad-hoc management is difficult



# What is DevOps

- Collaborative
  - Tear down silos
    - We should all be working towards the same goal and have each other's back
- Systematic
  - Emphasis on the big picture. All the bandwidth and uptime in the world means nothing if the services fail
- Iterative
  - Work towards a series of goals
  - Don't have to boil the ocean - start small and get feedback often
- Automated
  - Build, Test, and Deliver at scale. Eliminate time sucks.
  - Infrastructure as Code

Change?





# State of DevOps

<https://puppetlabs.com/2015-devops-report>

- High-performing IT organizations experience **60 times fewer failures and recover from failure 168 times faster** than their lower-performing peers. They also **deploy 30 times more frequently with 200 times shorter lead times**. Failures are unavoidable, but how quickly you detect and recover from failure can mean the difference between leading the market and struggling to catch up with the competition.
- Burnout can be prevented, and DevOps can help. **Burnout is associated with pathological cultures and unproductive, wasteful work**. The consequences of burnout are huge, both for individuals and for organizations. Organizations can fix the conditions that lead to burnout by fostering a supportive work environment and ensuring work is meaningful, and that **employees understand how their own work ties to strategic objectives**.

# NetEng's "must become programmers"



# You are not the CLI

Industry has rewarded memorizing CLI commands.

Network engineers are well versed in understanding complex problems and distributed systems.

Realize the value you can provide to your organizations - move beyond the CLI



**Matt Oswalt**  
@Mierdin

10 Nov

It is clear to me that networking has a lot of work to do in meeting infrastructure devs where they live



**Matt Oswalt**  
@Mierdin

 Follow

The reality is that networking is not that hard, or at least it doesn't have to be. The fact that this is not obvious means we have failed.

6:15 PM - 10 Nov 2015

  1  5

# Think like a programmer

- In basic terms, programming is the manipulation of data.
- You already know the core concepts of data types and how to manipulate them, the missing link is language and tools.

IT'S SHOWTIME

BECAUSE I'M GOING TO SAY PLEASE a

TALK TO THE HAND "a is true"

BULLSHIT

TALK TO THE HAND "a is not true"

YOU HAVE NO RESPECT FOR LOGIC

YOU HAVE BEEN TERMINATED



ArnoldC

# Separate the HOW from the WHAT

Regardless of the language you speak, you know what this is.

You know that you can open and close this object and you may also be able to lock and unlock it.



## Hand crafted - artisanal configs



# A tale of two configs

## Cisco

```
hostname nanog
ip domain-name shermdog.com
ip name-server 10.0.0.1
ntp server 10.14.99.10
```

## Juniper

```
system {
  host-name nanog;
  domain-name shermdog.com;
  name-server {
    10.0.0.1;
  }
  ntp {
    server 10.14.99.10;
  }
}
```

# The How from the What

## Cisco

```
hostname nanog
ip domain-name shermdog.com
ip name-server 10.0.0.1
ntp server 10.14.99.10
```

## Juniper

```
system {
  host-name nanog;
  domain-name shermdog.com;
  name-server {
    10.0.0.1;
  }
  ntp {
    server 10.14.99.10;
  }
}
```



# Where's the beef?

Data can come from a variety of sources - YAML, JSON, SQL, etc. Source control it!

---

```
host_name: nanog
domain: shermdog.com
dns: 10.0.0.1
ntp_server: 10.14.99.10
```

# Templates

## Cisco

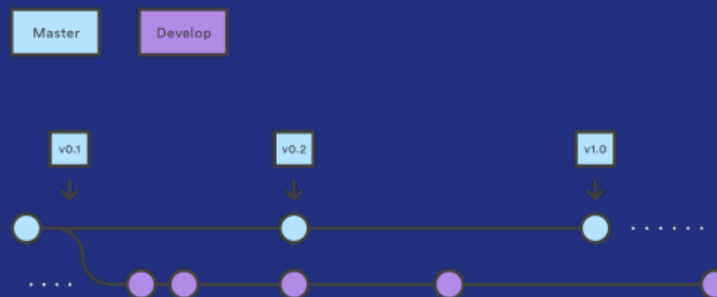
```
hostname {{ host_name }}
ip domain-name {{ domain }}
ip name-server {{ dns }}
ntp server {{ ntp_server }}
```

## Juniper

```
system {
    host-name {{ host_name }};
    domain-name {{ domain }};
    name-server {
        {{ dns }};
    }
    ntp {
        server {{ ntp_server }};
    }
}
```

# Git with the program


- Source control is \*AMAZING\*
- Git is a version control tool. It create a facility to store version history of files and folders (organized as projects). It has mechanism for teamwork and sharing with a foundation around file and history integrity.
- Unlike traditional source control where versions are stored as a set of diffs, Git stores a snapshot of the entire project – much like a file system. This gives users great flexibility to retrieve code throughout the history.



<https://www.atlassian.com/git/tutorials/>

# Stop, Collaborate and Listen.

```
104 - # can't autogen getters and setters because the default_<prop>
105 - # functions are class functions
106 - def deadline
107 -   return :default if @resource[:deadline] == :default &&
108 -     @property_hash[:deadline] ==
109 -     Cisco::AaaServerGroup.default_deadtime
110 -   @property_hash[:deadline]
111 - end
112 -
113 - def deadline=(set_value)
114 -   set_value = Cisco::AaaServerGroup.default_deadtime if
115 -   set_value == :default
116 -   @property_flush[:deadline] = set_value
117 - end
118 -
119 + # can't autogen server_hosts, special array handling
120 + def server_hosts
121 +   return [:default] if @resource[:server_hosts] &&
122 +     @resource[:server_hosts][0] == :default
123 +     @property_hash[:server_hosts] ==
124 +     Cisco::AaaServerGroup.default_servers
125 +     @aaa_group.default_servers
```

 chrisvanheuvell added a note 4 days ago

nit: this return (and others in this file) would be cleaner as:

```
return [:default] if
  @resource[:server_hosts] &&
  @resource[:server_hosts][0] == :default &&
  @property_hash[:server_hosts] == @aaa_group.default_servers
```

Commit Message	Author	Commit Hash	Line	Code Snippet
(PUP-3695) yum provider handling ...	ihoblitt	fd58fa3	69	def self.check_updates(enablerepo, disablerepo, c...
(PUP-4055) Make necessary chang...	whopper	657ba78	70	args = [command(:cmd), 'check-update']
(PUP-1362) (PUP-1775) map yumhe...	ihoblitt	f8853df	71	args.concat(enablerepo.map {  repo  ["--enabler...
(PUP-3695) yum provider handling ...	ihoblitt	fd58fa3	72	args.concat(disablerepo.map {  repo  ["--disabl...
(PUP-1060) Respect yum enable an...	adrienthebo	6dc8a0e	73	args.concat(disableexcludes.map {  repo  ["--di...
(PUP-1362) (PUP-1775) Use yum c...	adrienthebo	181d1de	74	
(PUP-1362) (PUP-1775) Use yum c...	adrienthebo	181d1de	75	output = Puppet::Util::Execution.execute(args, :failonfail => false, :combine => f...
(PUP-4055) Make necessary chang...	whopper	657ba78	76	
(PUP-1362) (PUP-1775) Use yum c...	adrienthebo	181d1de	77	updates = {}
(PUP-4055) Make necessary chang...	whopper	657ba78	78	if output.exitstatus == 100
(PUP-1362) (PUP-1775) Use yum c...	adrienthebo	181d1de	79	updates = parse_updates(output)
(PUP-4055) Make necessary chang...	whopper	657ba78	80	elsif output.exitstatus == 0
(PUP-1362) (PUP-1775) Use yum c...	adrienthebo	181d1de	81	self.debug "#{command(:cmd)} check-update exited with 0; no package updates avail...
(PUP-4055) Make necessary chang...	whopper	657ba78	82	else
(PUP-1362) (PUP-1775) Use yum c...	adrienthebo	181d1de	83	self.warn "Could not check for updates, '#{command(:cmd)} check-update' exited w...
(PUP-4055) Make necessary chang...	whopper	657ba78	84	end
(PUP-1362) (PUP-1775) Use yum c...	adrienthebo	181d1de	85	updates
(PUP-4055) Make necessary chang...	whopper	657ba78	86	end

Code Pull requests 3

Branch: master

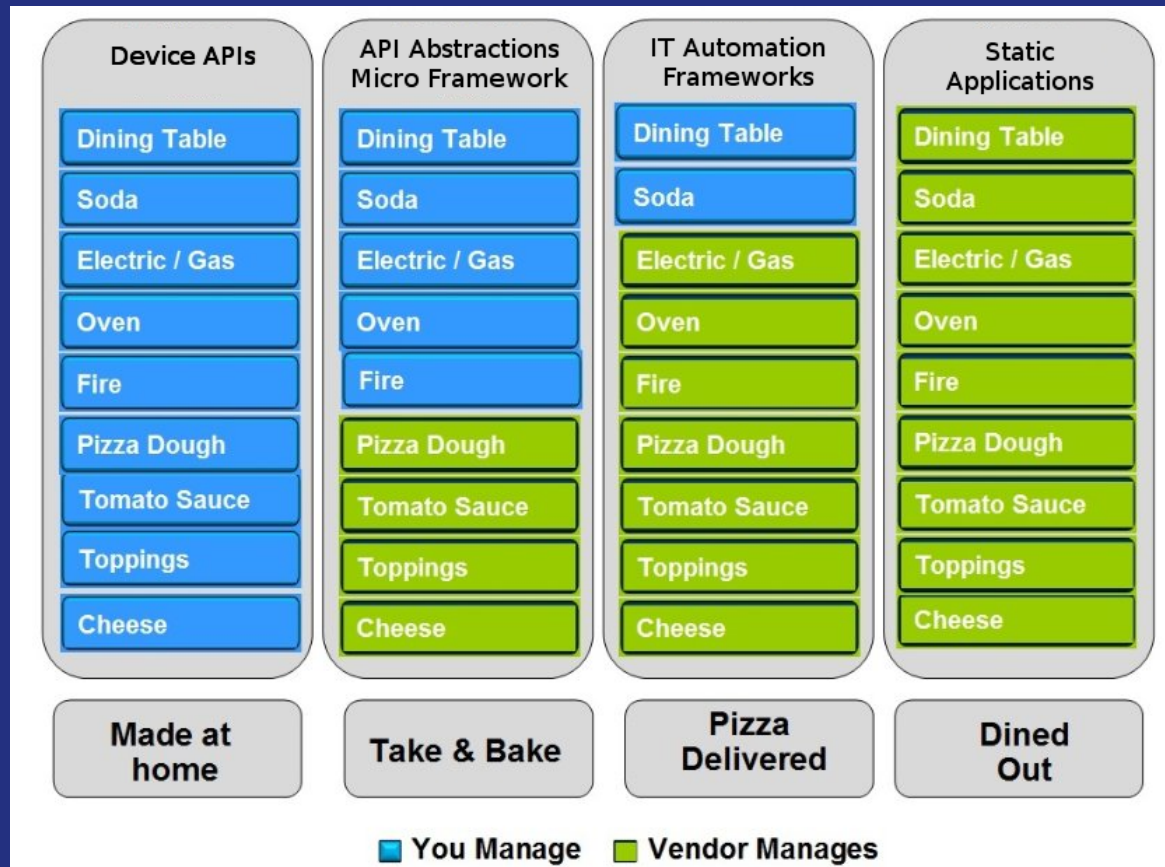
Switch branches/tags

Find a tag...

Branches Tags

- 0.4.1
- 0.4.0
- 0.3.19
- 0.3.18

# Network Automation as Pizza







# Rise of the API

Vendors are opening up their platforms with a variety of API's and abstraction layers (highlights in no particular order)

- Cisco
  - NX-API, onePK
  - Python API
- Juniper
  - Python PyEZ
  - JET
- Arista
  - eAPI Python Library

# IT Automation Frameworks

Ruby	Python
<ul style="list-style-type: none"><li>• Agent Based (some agentless support)</li><li>• Puppet DSL</li><li>• Network Devices - Officially Supported</li><li>• Large community</li><li>• Mature commercial offering</li></ul> 	<ul style="list-style-type: none"><li>• Agentless</li><li>• YAML + Jinja2 Filters</li><li>• Network Devices - Vendor/Community Supported</li><li>• Growing community</li><li>• Basic commercial offering</li></ul> 
<ul style="list-style-type: none"><li>• Agent Based</li><li>• Ruby DSL</li><li>• Network Devices - Officially Supported</li><li>• Large community</li><li>• Mature commercial offering</li></ul> 	<ul style="list-style-type: none"><li>• Agent Based (some agentless support)</li><li>• YAML / Jinja</li><li>• Minimal Network</li><li>• Small community</li><li>• Basic commercial offering</li></ul> 

# GNS3

evpn-cumulus-juniper-cisco-arista.gns3

Topology Summary

- SW1
  - swp2 <-> ge-0/0/0 PE-R1
- PE-R1
  - ge-0/0/0 <-> swp2 SW1
  - ge-0/0/1 <-> g0/0/0/1 P-R2
- P-R2
  - g0/0/0/0 <-> g0/0/0/0 P-R3
  - g0/0/0/1 <-> ge-0/0/1 PE-R1
- P-R3
  - g0/0/0/0 <-> g0/0/0/0 P-R2
  - g0/0/0/1 <-> ge-0/0/1 PE-R4
- SW2
  - e1 <-> ge-0/0/0 PE-R4
- PE-R4
  - ge-0/0/0 <-> e1 SW2
  - ge-0/0/1 <-> g0/0/0/1 P-R3

Console

```
}  
},  
"type": "topology",  
"version": "1.4.0beta2"  
}  
=> version  
GNS3 version is 1.4.0beta2 (compiled)  
GNS3 Converter version is 1.2.3  
Python version is 3.5.0 (64-bit) with utf-8 encoding  
Qt version is 5.5.0  
PyQt version is 5.5  
SIP version is 4.16.9  
=>
```

Jungle Newsfeed

**GNS3**  
Jungle

**THE ONLY RESOURCE YOU NEED**  
The Jungle has everything you will ever need for GNS3. Come check it out now.

[Go to the Jungle](#)



# Cisco - VIRL

The screenshot displays the VIRL software interface, which is used for creating and managing virtual network environments. The main workspace shows a network topology diagram with several nodes: two core routers (csr1000v-1 and csr1000v-2), two leaf switches (iosxrv-1 and iosxrv-2), and two virtual switches (iosv-1 and iosv-2). The connections are represented by blue lines.

On the left side, there is a 'Tools' panel with icons for 'Select' and 'Connect'. Below that, a 'Nodes' panel lists various network devices like IOSr, IOS X8v, CSR1000v, Server, NX-OSv, Vanta, VSRX, and FTDv-2. A 'General' section includes 'Multipoint Connection', 'Site', 'L2 External (DMZ)', 'L2 External (DMT)', and 'External Router'.

At the bottom left, a 'Projects' list shows a series of configuration files, including 'a1\_ahdhpas\_connection\_topo.virt'. The main bottom panel displays the configuration for the selected node, which is 'iosv-1'. The configuration text is as follows:

```
node
  This will be auto-generated by AutoNetS.
  Automation
    T 03: 88 Conf generated on 2014-11-18 19:31
    by 986. Configuration Engine 0.10.7
  Configuration
    hostname iosv-1
    service timestamps log datetime msec
    service timestamps debug datetime msec
    service timestamps local datetime msec
    service timestamps log datetime msec
    no ip domain-lookup
    line console 0
    exec-timeout 720 0
    no logging
    line default
    exec-timeout 720 0
    vty 0 4
    vty 5 15
    vty 16 25
    vty 26 35
    vty 36 45
    vty 46 55
    vty 56 65
    vty 66 75
    vty 76 85
    vty 86 95
    vty 96 105
    vty 106 115
    vty 116 125
    vty 126 135
    vty 136 145
    vty 146 155
    vty 156 165
    vty 166 175
    vty 176 185
    vty 186 195
    vty 196 205
    vty 206 215
    vty 216 225
    vty 226 235
    vty 236 245
    vty 246 255
    vty 256 265
    vty 266 275
    vty 276 285
    vty 286 295
    vty 296 305
    vty 306 315
    vty 316 325
    vty 326 335
    vty 336 345
    vty 346 355
    vty 356 365
    vty 366 375
    vty 376 385
    vty 386 395
    vty 396 405
    vty 406 415
    vty 416 425
    vty 426 435
    vty 436 445
    vty 446 455
    vty 456 465
    vty 466 475
    vty 476 485
    vty 486 495
    vty 496 505
    vty 506 515
    vty 516 525
    vty 526 535
    vty 536 545
    vty 546 555
    vty 556 565
    vty 566 575
    vty 576 585
    vty 586 595
    vty 596 605
    vty 606 615
    vty 616 625
    vty 626 635
    vty 636 645
    vty 646 655
    vty 656 665
    vty 666 675
    vty 676 685
    vty 686 695
    vty 696 705
    vty 706 715
    vty 716 725
    vty 726 735
    vty 736 745
    vty 746 755
    vty 756 765
    vty 766 775
    vty 776 785
    vty 786 795
    vty 796 805
    vty 806 815
    vty 816 825
    vty 826 835
    vty 836 845
    vty 846 855
    vty 856 865
    vty 866 875
    vty 876 885
    vty 886 895
    vty 896 905
    vty 906 915
    vty 916 925
    vty 926 935
    vty 936 945
    vty 946 955
    vty 956 965
    vty 966 975
    vty 976 985
    vty 986 995
    vty 996 1005
    interface Loopback0
      description Loopback
      ip address 10.0.0.1 255.255.255.252
      no shutdown
    interface Ethernet0/0/0
      description to-core-1
      ip address 10.0.0.17 255.255.255.252
      no shutdown
    interface Ethernet0/0/1
      description to-core-2
      ip address 10.0.0.18 255.255.255.252
      no shutdown
    interface Ethernet0/0/2
      description to-core-3
```

On the right side, there is a 'Node Editor' panel with a 'Device' dropdown set to 'iosv-1'. Below it, a 'Graph Overview' section shows a simplified version of the network topology with nodes represented by icons and connected by lines. The 'Cisco' logo is visible in the bottom right corner of the interface.

# The Unicorns

Google

 Microsoft



# **Cross-Vendor Standards and the Future of Network Automation**

# NETCONF

## NETCONF - IETF network management standard

- XML based encoding
  - Vendor specific data models and implementation
- Configuration RPCs
  - get-config, edit-config, copy-config, delete-config, lock, unlock
- Operational state RPCs
  - Generally map to CLI “show” commands
- Transport: SSH, HTTPS, TLS, BEEP

# YANG

## YANG - IETF Data Modeling Language for Netconf

- Human-readable representation of data
- Hierarchical data node representation
- Built-in data types
- Constraints can be placed on the data
- Extensible

Data is still vendor (or group) specific

**WHERE TO BEGIN?**

**HOW CAN I HELP?**

**THANK YOU!**

**shermdog@puppetlabs.com**  
**@shermdog01**

