ARISTA

Modern & Scalable Data Center Networks

Benoît “tsuna” Sigoure
Member of the Yak Shaving Staff
tsuna@aristanetworks.com

@tsunanet

What is SDN?

Purist View

a strict separation of control plane and data plane

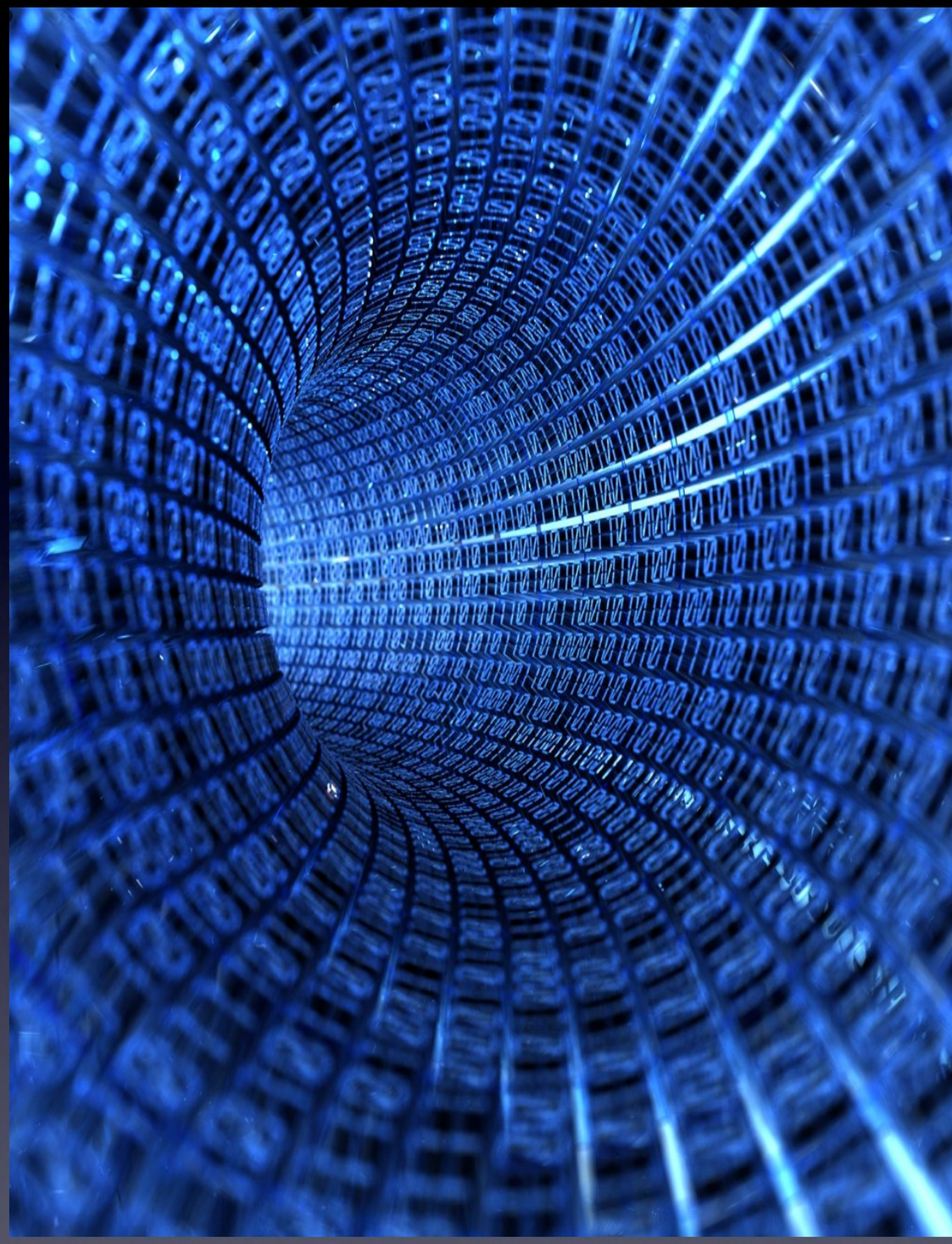
Pragmatic View

a network architecture designed to be programmed by high-level languages and APIs

A Common View

SDN = Network Virtualization

SDN = OpenFlow

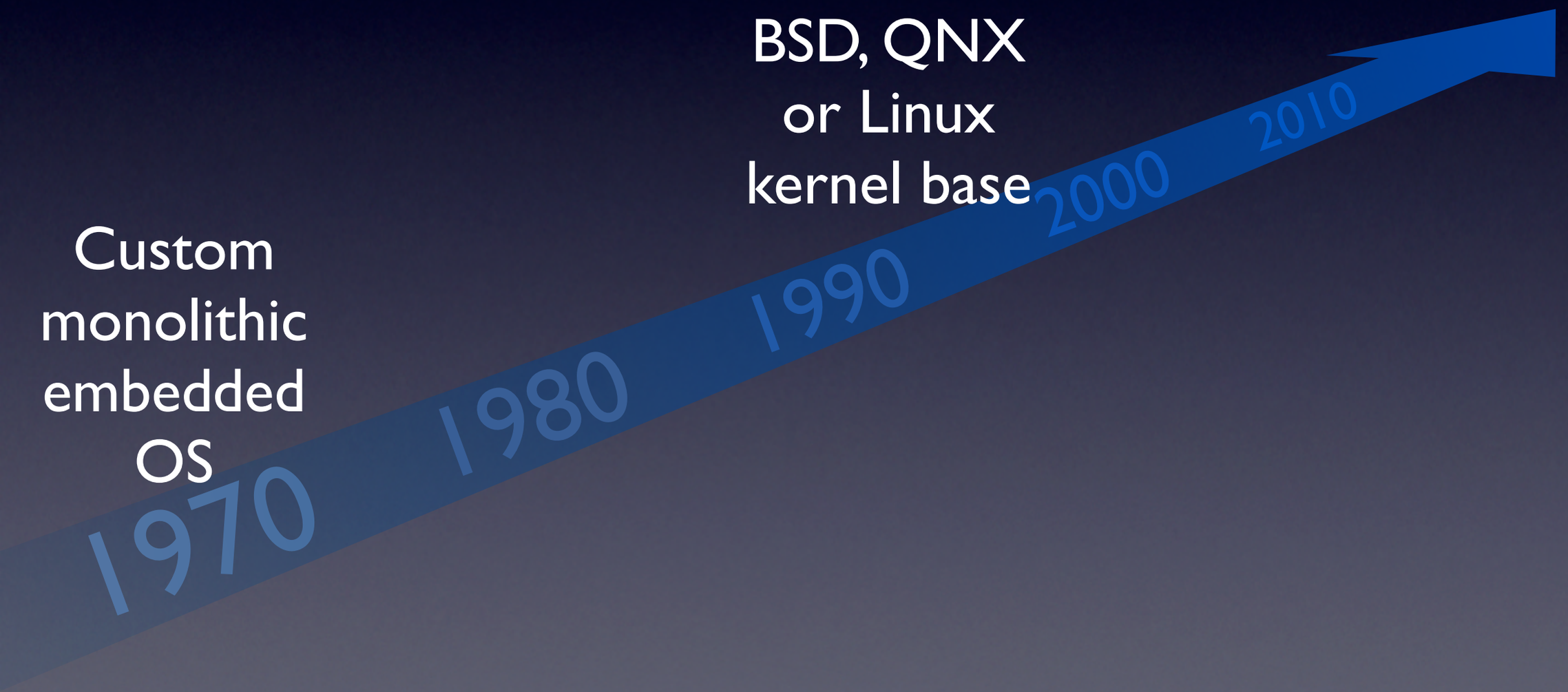


ARISTA

A Brief History of Network Software

Modified
BSD, QNX
or Linux
kernel base

Custom
monolithic
embedded
OS

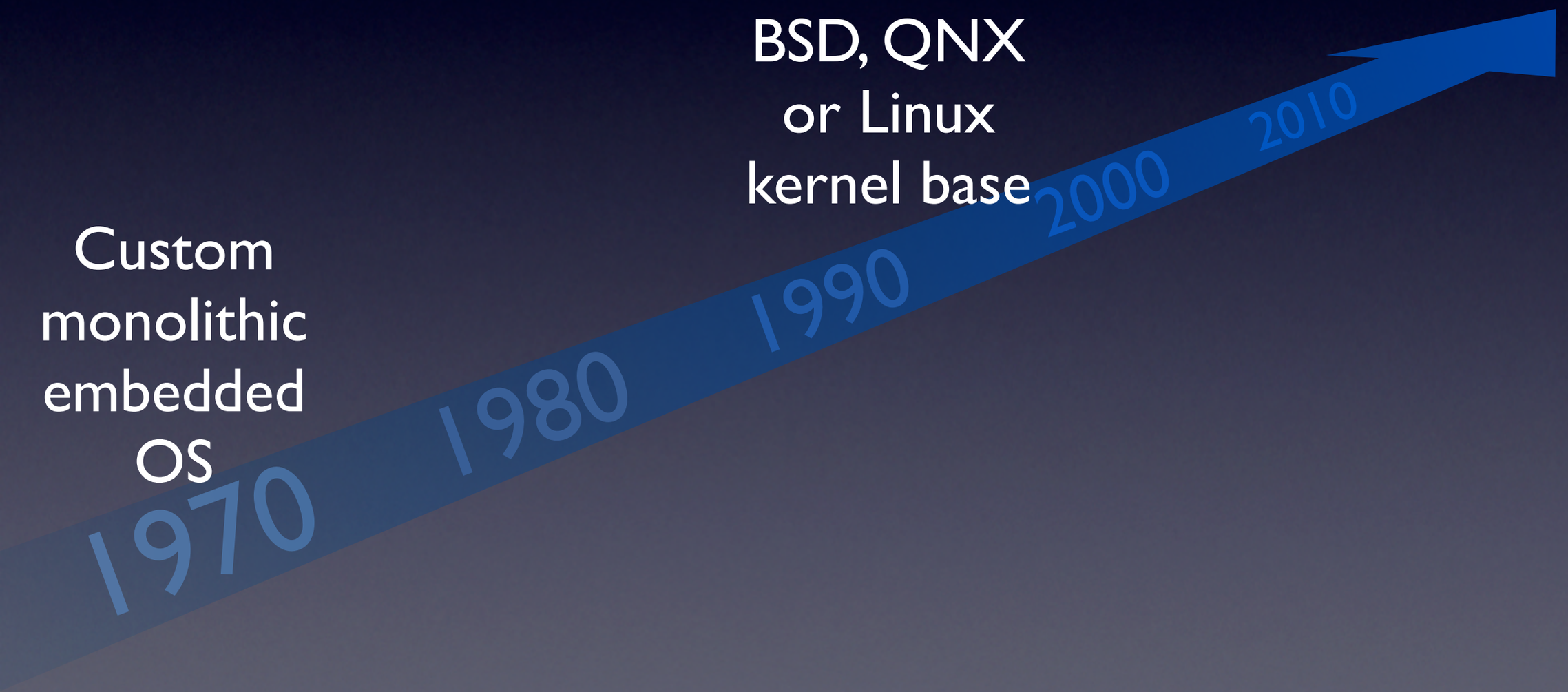


A Brief History of Network Software

Custom
monolithic
embedded
OS

Modified
BSD, QNX
or Linux
kernel base

"The switch
is just a
Linux box"



A Brief History of Network Software

1970

Custom monolithic embedded OS

1980

1990

Modified BSD, QNX or Linux kernel base

2000

2010

"The switch is just a Linux box"

EOS = Linux

A Brief History of Network Software

1970

Custom monolithic embedded OS

1980

Modified BSD, QNX or Linux kernel base

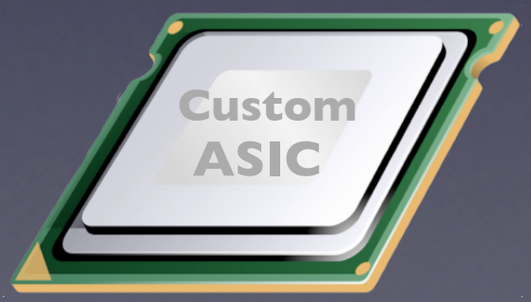
1990

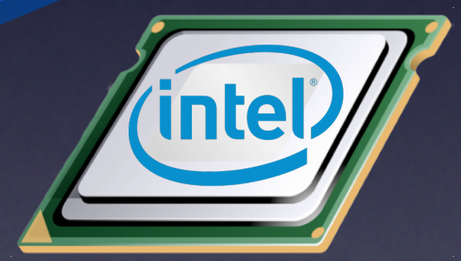
2000

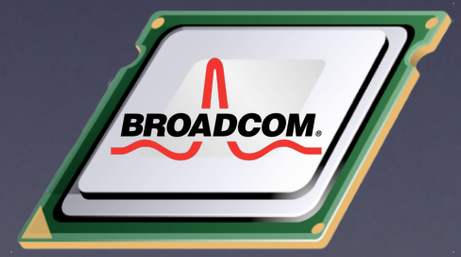
2010

"The switch is just a Linux box"

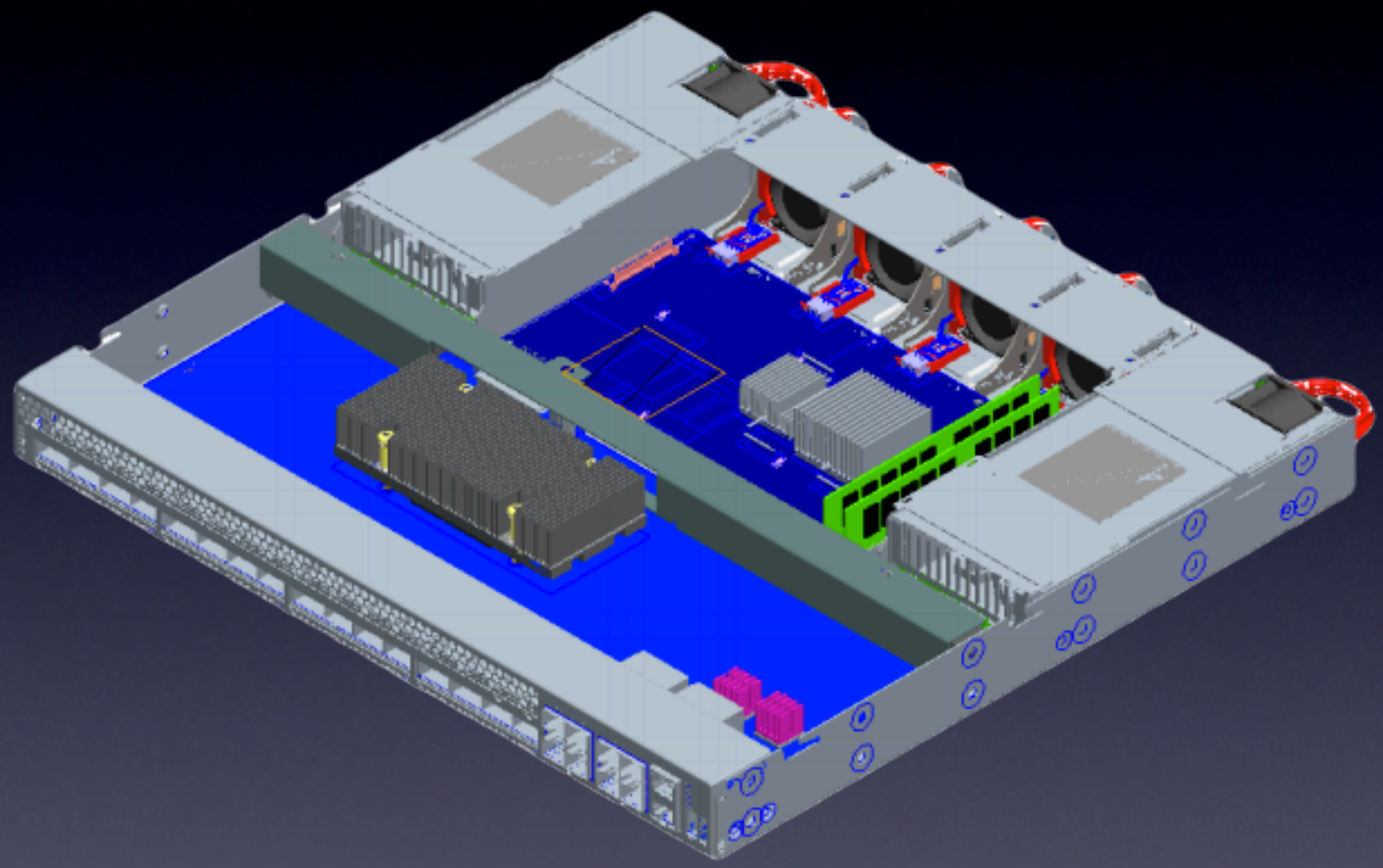
EOS = Linux



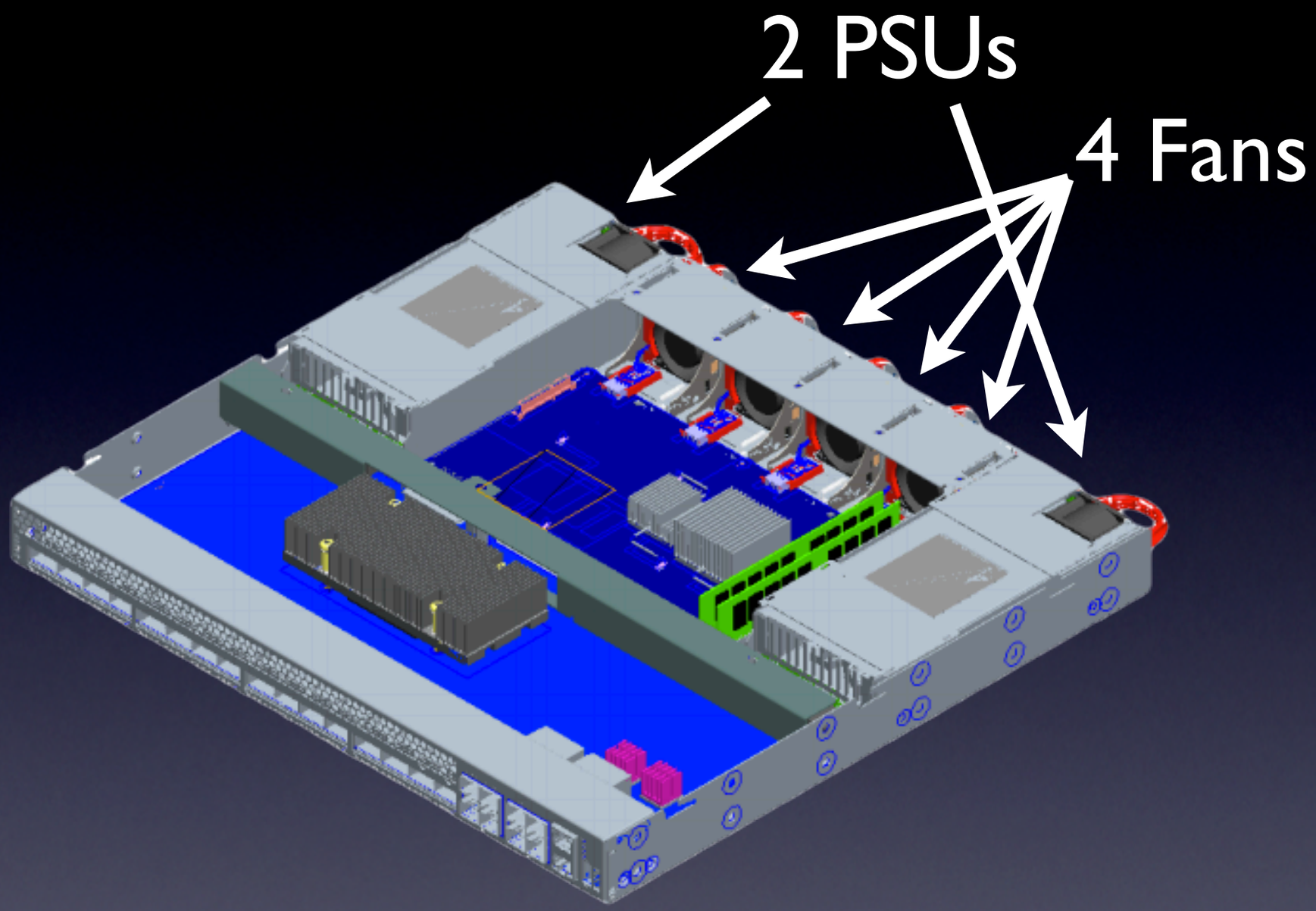




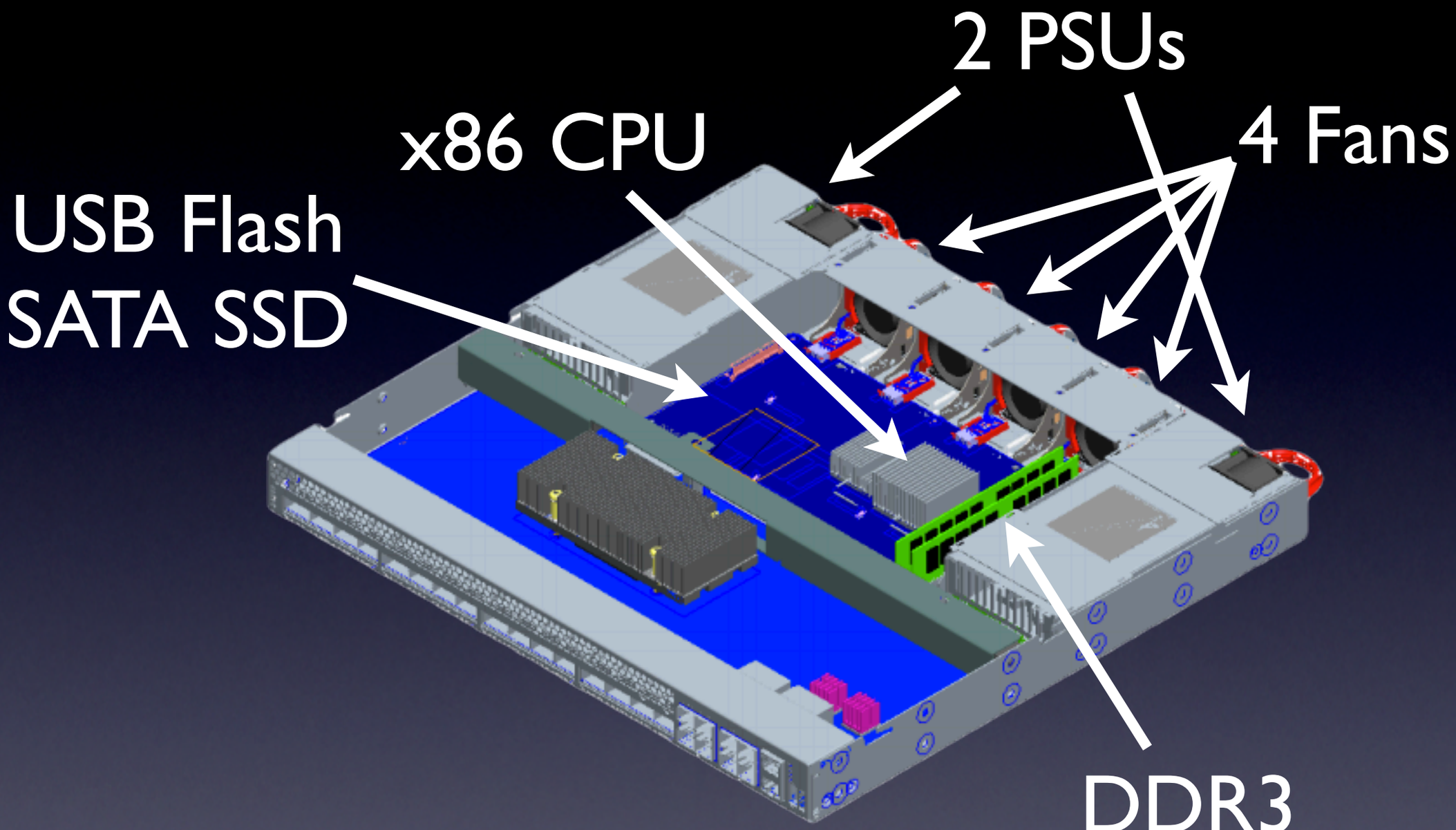
Inside a Modern Switch



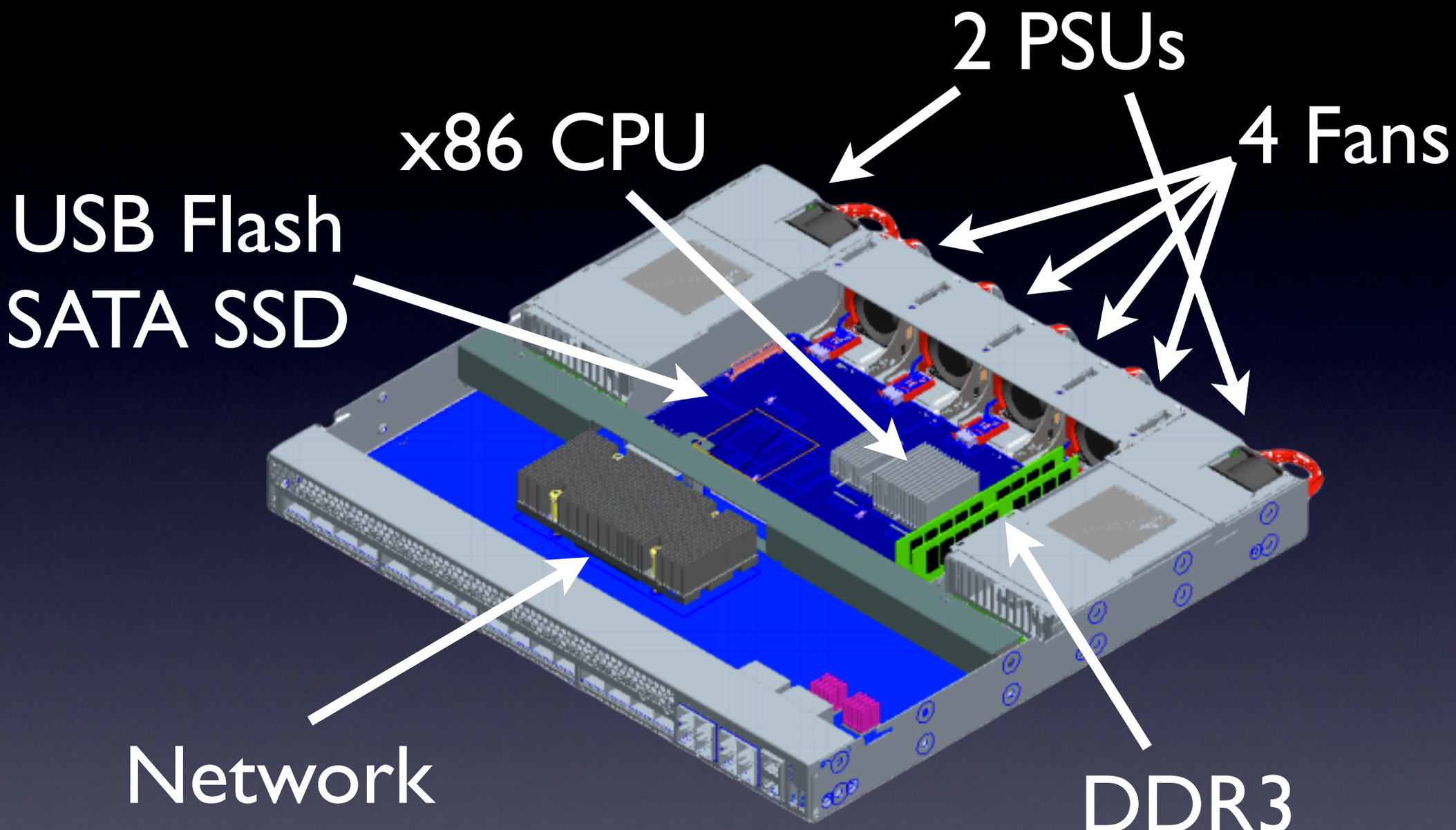
Inside a Modern Switch



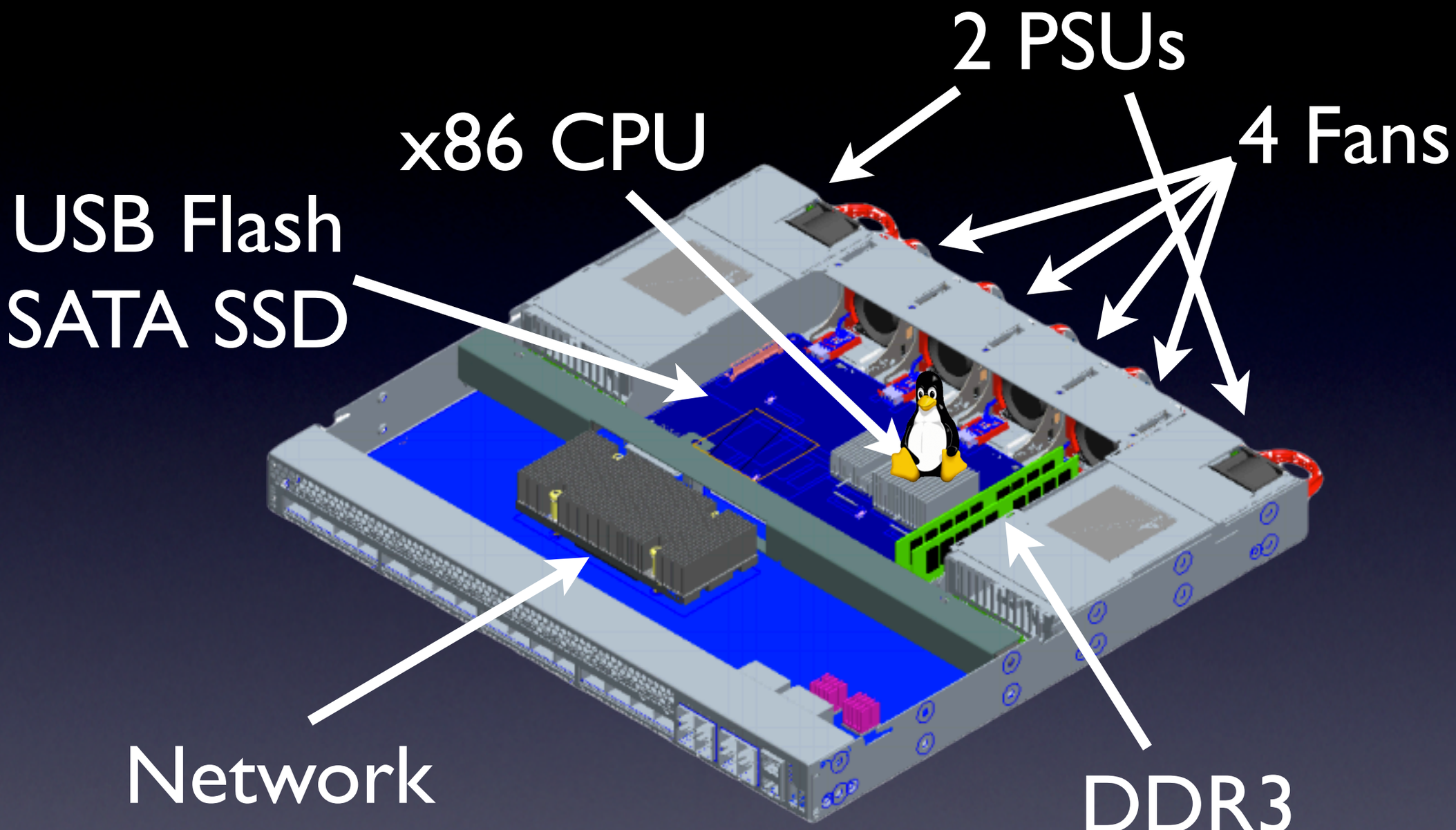
Inside a Modern Switch



Inside a Modern Switch

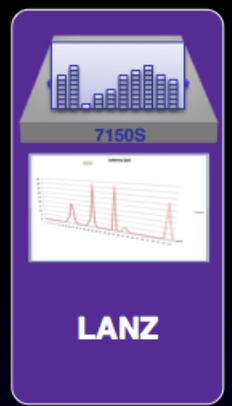


Inside a Modern Switch



It's All About The Software

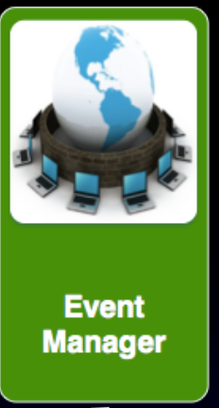
Time Stamping



LANZ

Advanced Mirroring

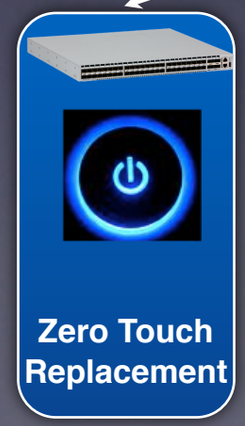
CLI Scheduler



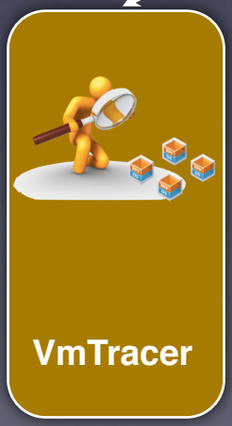
Event Manager

Command API

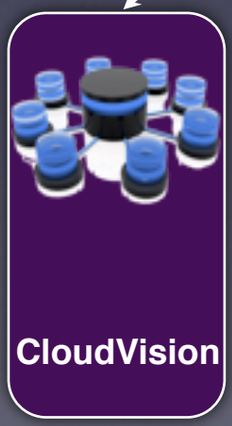
Event Monitor



Zero Touch Replacement



VmTracer

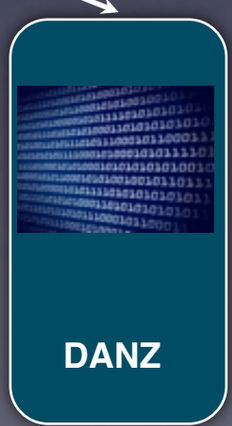


CloudVision

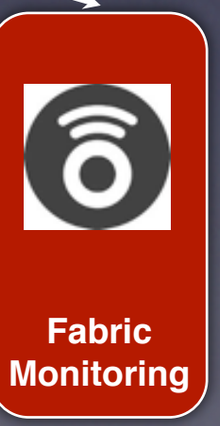
Email

Wireshark

Fast Failover



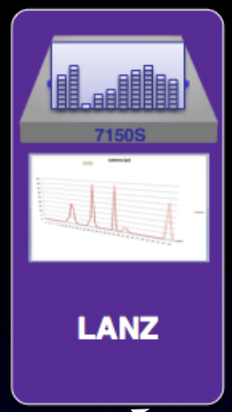
DANZ



Fabric Monitoring

It's All About The Software

Time Stamping



LANZ

Advanced Mirroring

CLI Scheduler

Event Manager

Command API

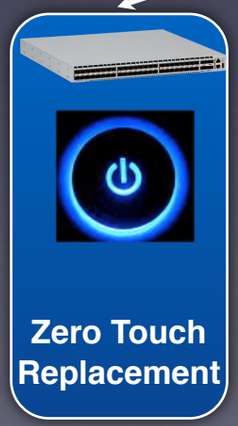
Event Monitor

Chef

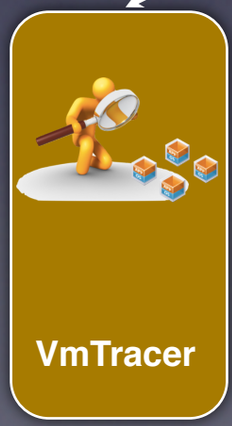
NGINX

Ganglia

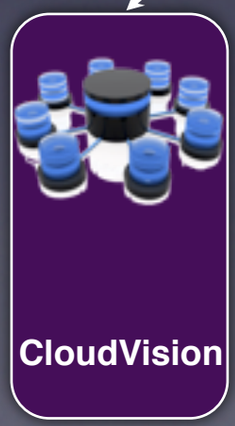
OpenTSDB



Zero Touch Replacement



VmTracer

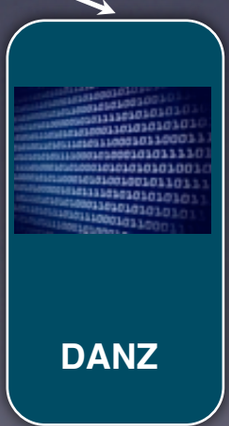


CloudVision

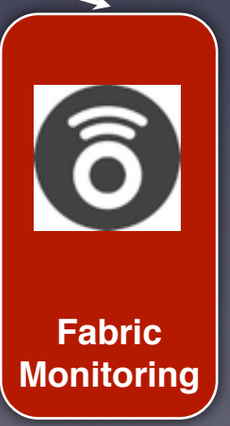
Email

Wireshark

Fast Failover



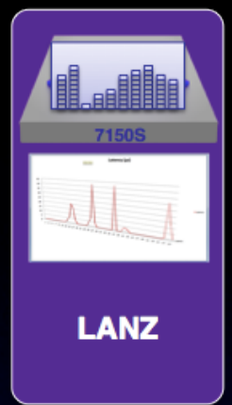
DANZ



Fabric Monitoring

It's All About The Software

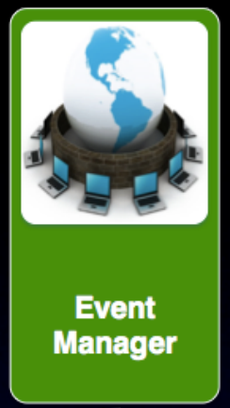
Time Stamping



LANZ

Advanced Mirroring

CLI Scheduler



Event Manager

Command API

Event Monitor

python

#!/bin/bash

puppet labs

Chef

NGINX

Ganglia

OpenTSDB

Zero Touch Replacement

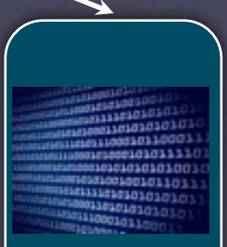
VmTracer

CloudVision

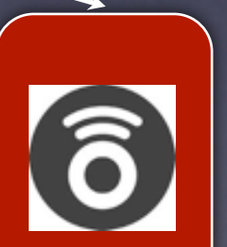
Email

Wireshark

Fast Failover



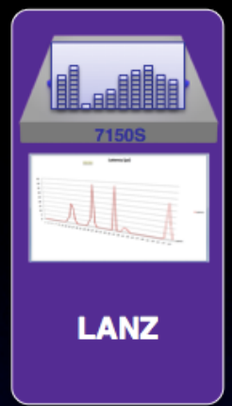
DANZ



Fabric Monitoring

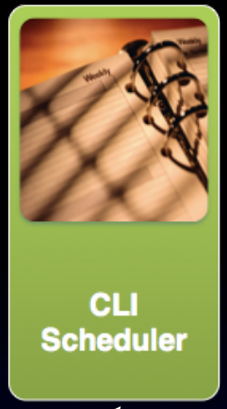
It’s All About The Software

Time Stamping

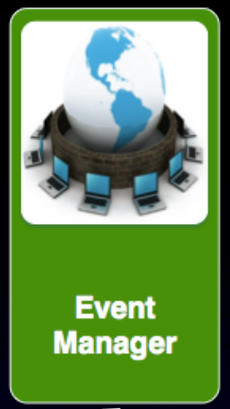


LANZ

Advanced Mirroring



CLI Scheduler



Event Manager

Command API

Event Monitor

python

#!/bin/bash

puppet labs

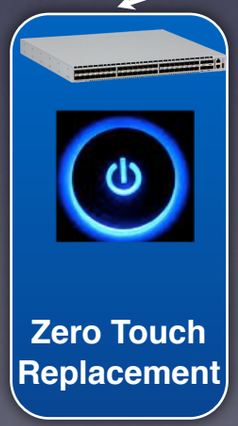
Chef

NGINX

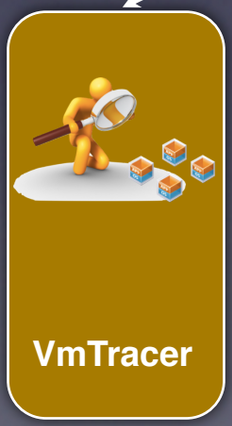
Ganglia

OpenTSDB

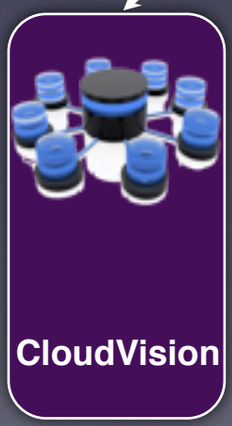
$ sudo yum install <pkg>



Zero Touch Replacement



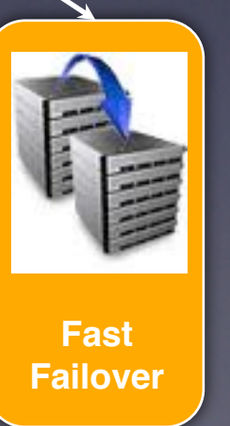
VmTracer



CloudVision

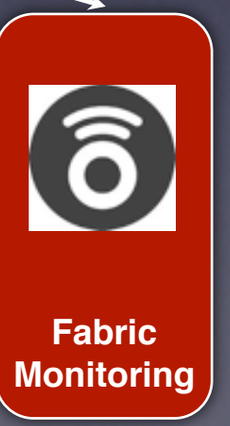
Email

Wireshark



Fast Failover

DANZ



Fabric Monitoring

Command API
Or a CLI over JSON-RPC

Command API
Or a CLI over JSON-RPC

stackoverflow Questions Tags Users Badges Unanswered

python regular expression across multiple lines

1

I'm gathering some info from some cisco devices using python and pexpect, and had a lot of success with REs to extract pesky little items. I'm afraid i've hit the wall on this. Some switches stack together, I have identified this in the script and used a separate routine to parse the data. If the switch is stacked you see the following (extracted from the sho ver output)

```
Top Assembly Part Number            : 800-25858-06
Top Assembly Revision Number        : A0
Version ID                          : V08
CLEI Code Number                    : COMDE10BRA
Hardware Board Revision Number      : 0x01


Switch   Ports  Model              SW Version            SW Image
------   -----  -----              ----------            ----------
*    1   52     WS-C3750-48P       12.2(35)SE5           C3750-IPBASE-M
     2   52     WS-C3750-48P       12.2(35)SE5           C3750-IPBASE-M
     3   52     WS-C3750-48P       12.2(35)SE5           C3750-IPBASE-M
     4   52     WS-C3750-48P       12.2(35)SE5           C3750-IPBASE-M


Switch 02
---------
Switch Uptime                       : 11 weeks, 2 days, 16 hours, 27 minutes
Base ethernet MAC Address           : 00:26:52:96:2A:80
Motherboard assembly number         : 73-9675-15
```

Command API
Or a CLI over JSON-RPC

stackoverflow Questions Tags Users Badges Unanswered

python regular expression across multiple lines

1

I'm gathering some info from some cisco devices using python and pexpect, and had a lot of success with REs to extract pesky little items. I'm afraid i've hit the wall on this. Some switches stack together, I have identified this in the script and used a separate routine to parse the data. If the switch is stacked you see the following (extracted from the sho ver output)

```
Top Assembly Part Number             : 800-25858-06
Top Assembly Revision Number         : A0
Version ID                           : V08
CLEI Code Number                     : COMDE10BRA
Hardware Board Revision Number       : 0x01

Switch    Ports  Model                SW Version            SW Image
------    -----  -----                ----------            ----------
*    1    52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M
     2    52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M
     3    52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M
     4    52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M

Switch 02
---------
Switch Uptime                        : 11 weeks, 2 days, 16 hours, 27 minutes
Base ethernet MAC Address            : 00:26:52:96:2A:80
Motherboard assembly number          : 73-9675-15
```

Command API
Or a CLI over JSON-RPC

stackoverflow Questions Tags Users Badges Unanswered

python regular expression across multiple lines

1

I'm gathering some info from some cisco devices using python and pexpect, and had a lot of success with REs to extract pesky little items. I'm afraid i've hit the wall on this. Some switches stack together, I have identified this in the script and used a separate routine to parse the data. If the switch is stacked you see the following (extracted from the sho ver output)

```
Top Assembly Part Number             : 800-25858-06
Top Assembly Revision Number         : A0
Version ID                           : V08
CLEI Code Number                     : COMDE10BRA
Hardware Board Revision Number       : 0x01

Switch    Ports  Model                SW Version            SW Image
------    -----  -----                ----------            ----------
*    1    52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M
     2    52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M
     3    52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M
     4    52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M

Switch 02
---------
Switch Uptime                        : 11 weeks, 2 days, 16 hours, 27 minutes
Base ethernet MAC Address            : 00:26:52:96:2A:80
Motherboard assembly number          : 73-9675-15
```

i'd like to get

```
*,1,WS-C3750-48P
,2,WS-C3750-48P
,3,WS-C3750-48P
,4,WS-C3750-48P
```

```python
import pexpect, re
connector = pexpect.spawn("ssh admin@1.2.3.4")
connector.expect(".ssword:*")
connector.sendline(password)
index = connector.expect([">","#"])
if index == 0:
  connector.sendline("enable")
  index = connector.expect(["assword", "#"])
  if index == 0:
    connector.sendline(enable)
    connector.expect("#")
connector.sendline("show version")
connector.expect("#")
output = connector.before
switches = re.findall("^(\*?)\s+(\d)\s+\d+\s+([A-Z\d-]+)", output, re.MULTILINE)
for master, num, model in switches:
   ...
```

stackoverfl

python regular exp

1

I'm gathering some inf
with REs to extract pe
have identified this in the script and used a separate routine to parse the data. If the switch is stacked you see the following (extracted from the sho ver output)

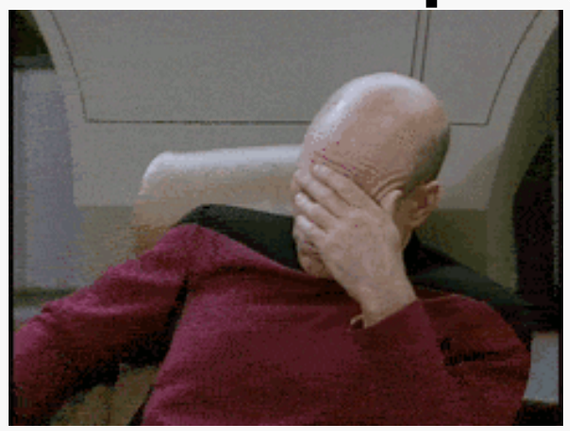
```
Top Assembly Part Number            : 800-25858-06
Top Assembly Revision Number        : A0
Version ID                          : V08
CLEI Code Number                    : COMDE10BRA
Hardware Board Revision Number      : 0x01

Switch   Ports  Model                SW Version            SW Image
------   -----  -----                ----------            ----------
*    1   52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M
     2   52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M
     3   52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M
     4   52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M

Switch 02
---------
Switch Uptime                       : 11 weeks, 2 days, 16 hours, 27 minutes
Base ethernet MAC Address           : 00:26:52:96:2A:80
Motherboard assembly number         : 73-9675-15
```

i'd like to get

```
*,1,WS-C3750-48P
,2,WS-C3750-48P
,3,WS-C3750-48P
,4,WS-C3750-48P
```

Traditional approach: “screen scraping”

```
import pexpect, re
connector = pexpect.spawn("ssh admin@1.2.3.4")
connector.expect(".ssword:*")
connector.sendline(password)
index = connector.expect([">","#"])
if index == 0:
  connector.sendline("enable")
  index = connector.expect(["assword", "#"])
  if index == 0:
    connector.sendline(enable)
    connector.expect("#")
connector.sendline("show version")
connector.expect("#")
output = connector.before
switches = re.findall("^(\*?)\s+(\d)\s+\d+\s+([A-Z\d-]+)", output, re.MULTILINE)
for master, num, model in switches:
   ...
```



stackoverfl

python regular exp

1

I'm gathering some inf
with REs to extract pe
have identified this in the script and used a separate routine to parse the data. If the switch is stacked you see the following (extracted from the sho ver output)

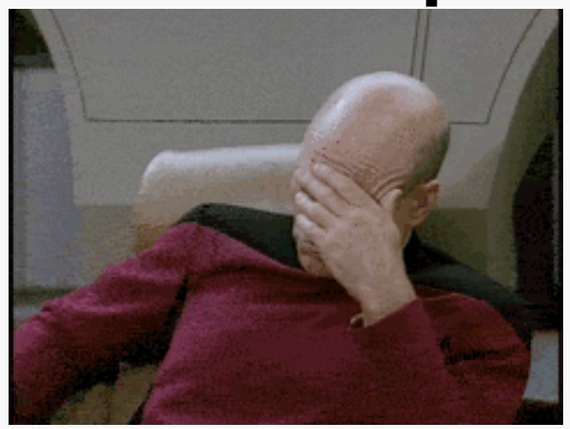
```
Top Assembly Part Number             : 800-25858-06
Top Assembly Revision Number         : A0
Version ID                           : V08
CLEI Code Number                     : COMDE10BRA
Hardware Board Revision Number       : 0x01

Switch   Ports  Model                SW Version            SW Image
------   -----  -----                ----------            ----------
*    1   52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M
     2   52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M
     3   52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M
     4   52     WS-C3750-48P         12.2(35)SE5           C3750-IPBASE-M

Switch 02
---------
Switch Uptime                        : 11 weeks, 2 days, 16 hours, 27 minutes
Base ethernet MAC Address            : 00:26:52:96:2A:80
Motherboard assembly number          : 73-9675-15
```

i'd like to get

```
*,1,WS-C3750-48P
,2,WS-C3750-48P
,3,WS-C3750-48P
,4,WS-C3750-48P
```

Traditional approach: "screen scraping"

```
import pexpect, re
connector = pexpect.spawn("ssh admin@1.2.3.4")
connector.expect(".ssword:*")
connector.sendline(password)
index = connector.expect([">","#"])
if index == 0:
  connector.sendline("enable")
  index = connector.expect(["assword", "#"])
  if index == 0:
    connector.sendline(enable)
    connector.expect("#")
connector.sendline("show version")
connector.expect("#")
output = connector.before
switches = re.findall("^(\*?)\s+(\d)\s+\d+\s+([A-Z\d-]+)", output, re.MULTILINE)
for master, num, model in switches:
   ...
```



stackoverfl

python regular exp

I'm gathering some inf
with REs to extract pe
have identified this in the script and used a separate routine to parse the data. If the switch is stacked you see the following (extracted from the sho ver output)

1

```
Top Assembly Part Number            : 800-25858-06
Top Assembly Revision Number        : A0
Version ID                          : V08
CLEI Code Number                    : COMDE10BRA
Hardware Board Revision Number      : 0x01

Switch   Ports  Model                SW Version            SW Image
------   -----  -----                ----------            --------
*    1   52     WS-C3750-48P         1
     2   52     WS-C3750-48P         1
     3   52     WS-C3750-48P         1
     4   52     WS-C3750-48P         1

Switch 02
---------
Switch Uptime                       : 11
Base ethernet MAC Address           : 00
Motherboard assembly number         : 73-9675-15
```

i'd like to get

```
*,1,WS-C3750-48P
,2,WS-C3750-48P
,3,WS-C3750-48P
,4,WS-C3750-48P
```

Compare to: A Standard HTTP API

```
from jsonrpclib import Server
switch = Server("http://admin:" + passwrd + "@1.2.3.4/command-api")

response = switch.runCmds(1, ["show privilege"])
if response[0]["privilegeLevel"] != 15:
   switch.runCmds(1, [{"cmd": "enable", "input": enable}])

response = switch.runCmds(1, ["show version"])
for num, switch in enumerate(response[0]["stack"]):
   # use switch["model"] and switch["master"]
```

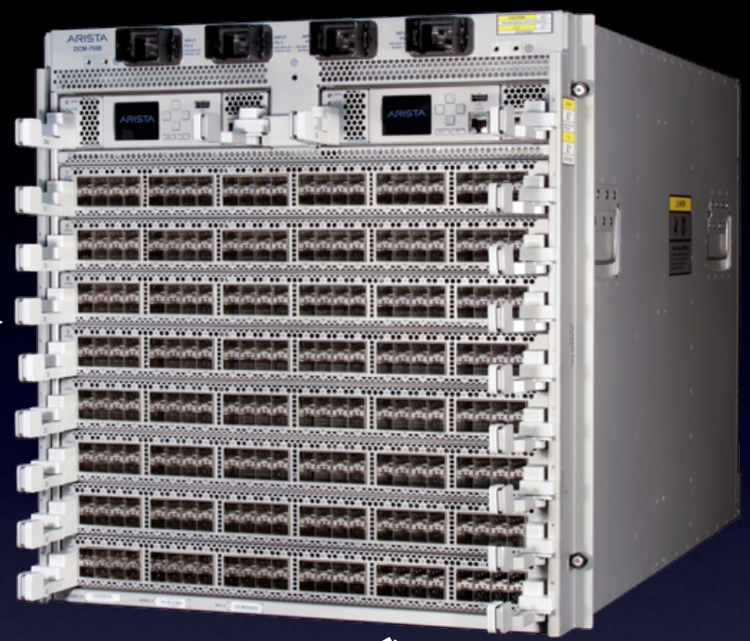
openstack™ Integration

Quantum

OVS Plugin

Arista Driver

EAPI



Spine Layer

Datacenter Leaf Layer

TOR1 TOR2 TOR3

Red VLAN

Blue VLAN

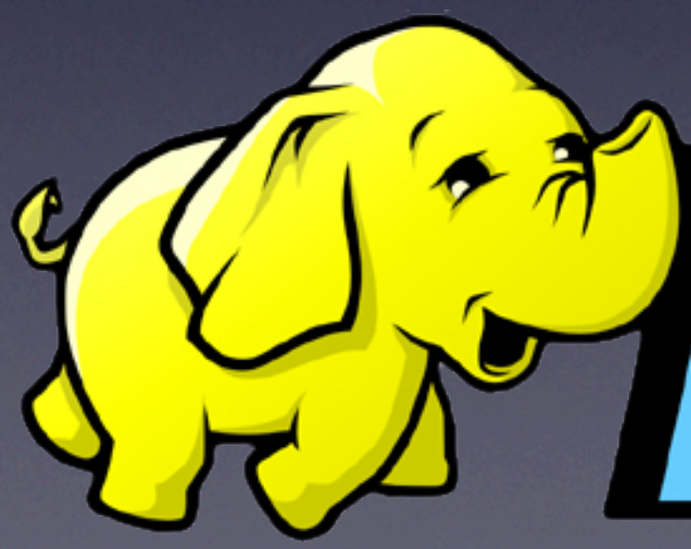
Host 1 Host 2 Host 3 Host 4 Host 5 Host 6

VM1 VM1 VM2 VM2 VM3 VM3 VM4

Fast Server Failover: Making the Network Work Better With

HBASE

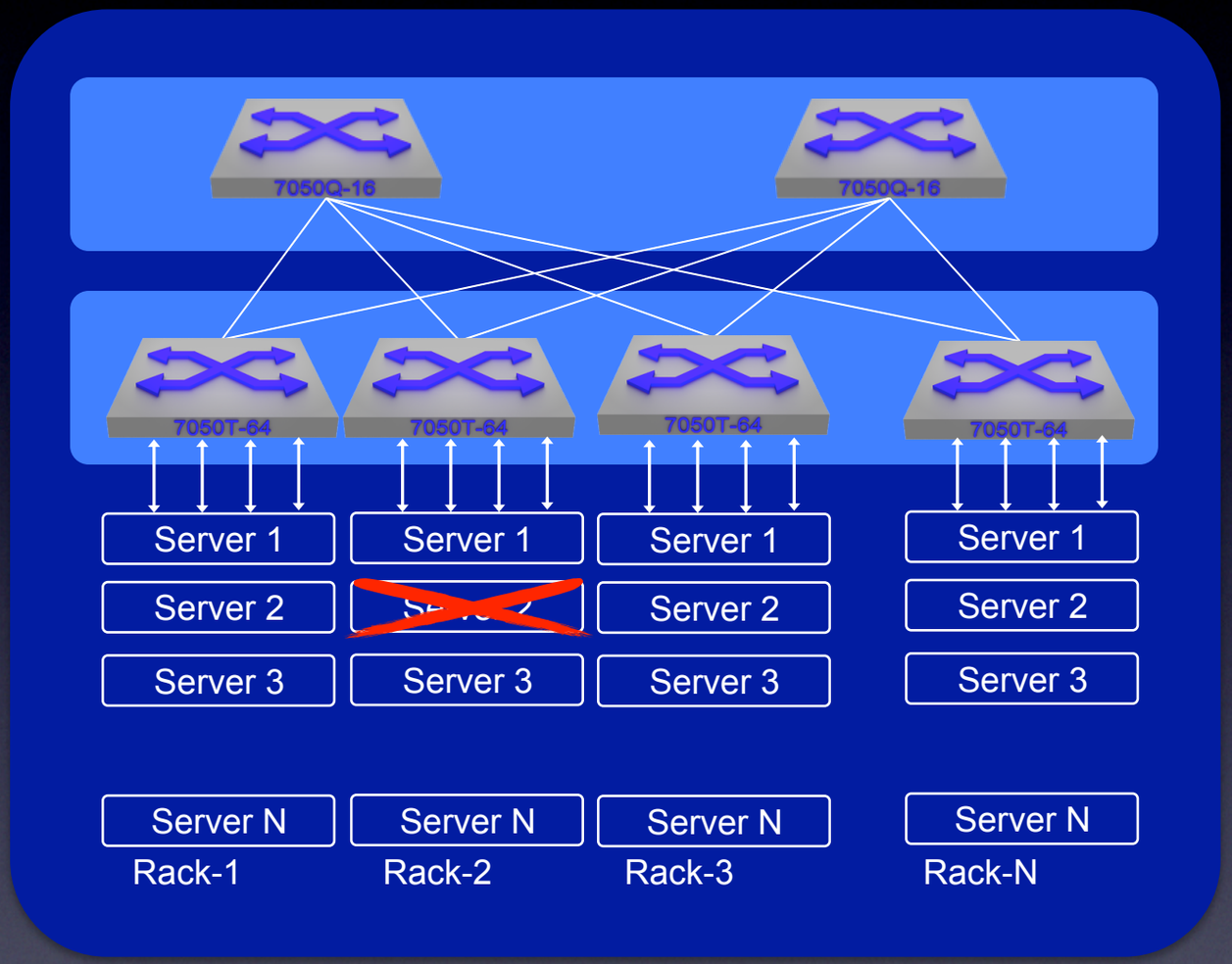
&



hadoop

Host Failures in Hadoop Clusters

- Hardware failures
- Kernel panics
- Operator errors
- NIC driver bugs

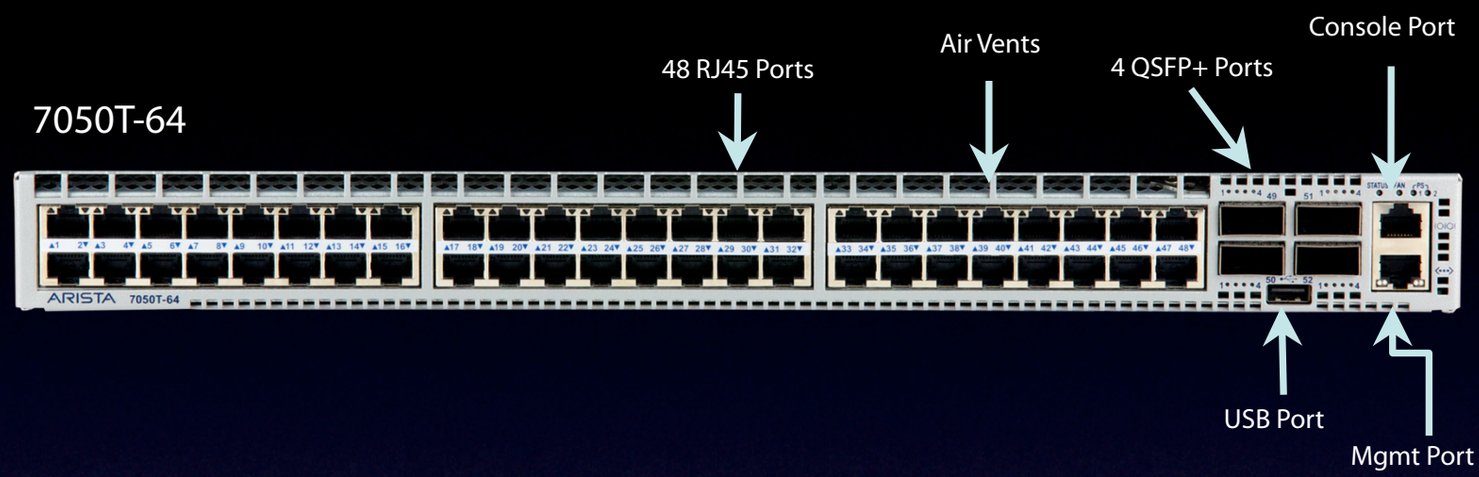


Host Failures for HBase & Hadoop

- RegionServer: wait for ZooKeeper lease timeout
  Typical: ~30s
- DataNode: wait for heartbeats to timeout
  enough for NameNode to declare dead

Typical: ~10min (or ~30s with `dfs.namenode.check.stale.datanode` see HDFS-3703)

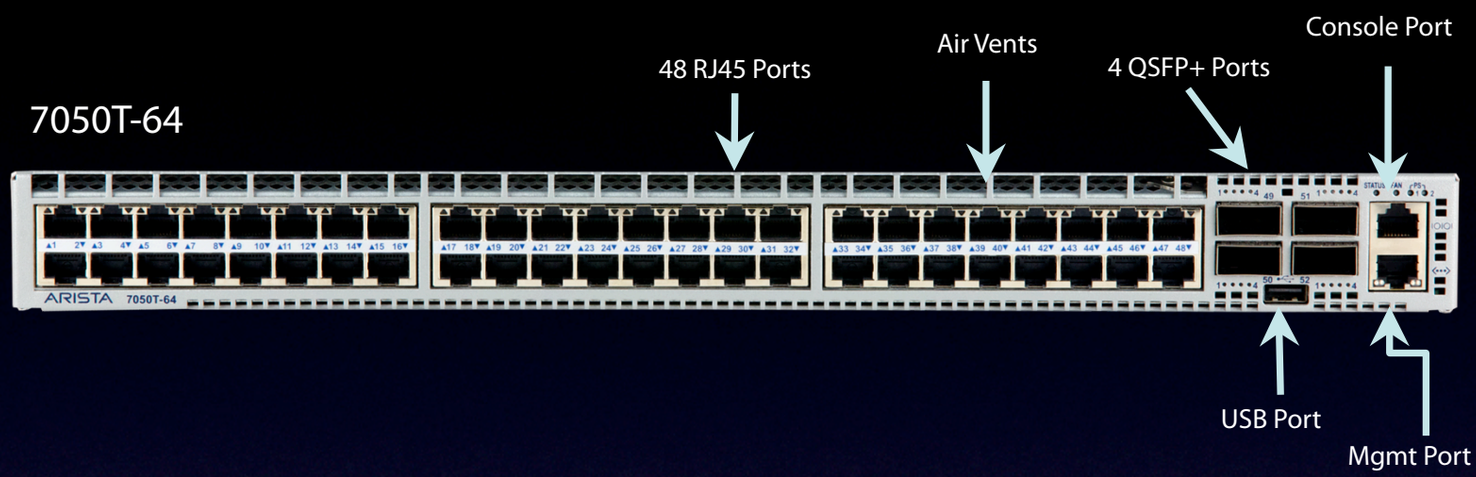
Manually Mitigating Host Failures



Process:

- Log into ToR
- Add IP address of the failed host as a secondary IP on the SVI used as the default gateway
- Remove IP when the host comes back

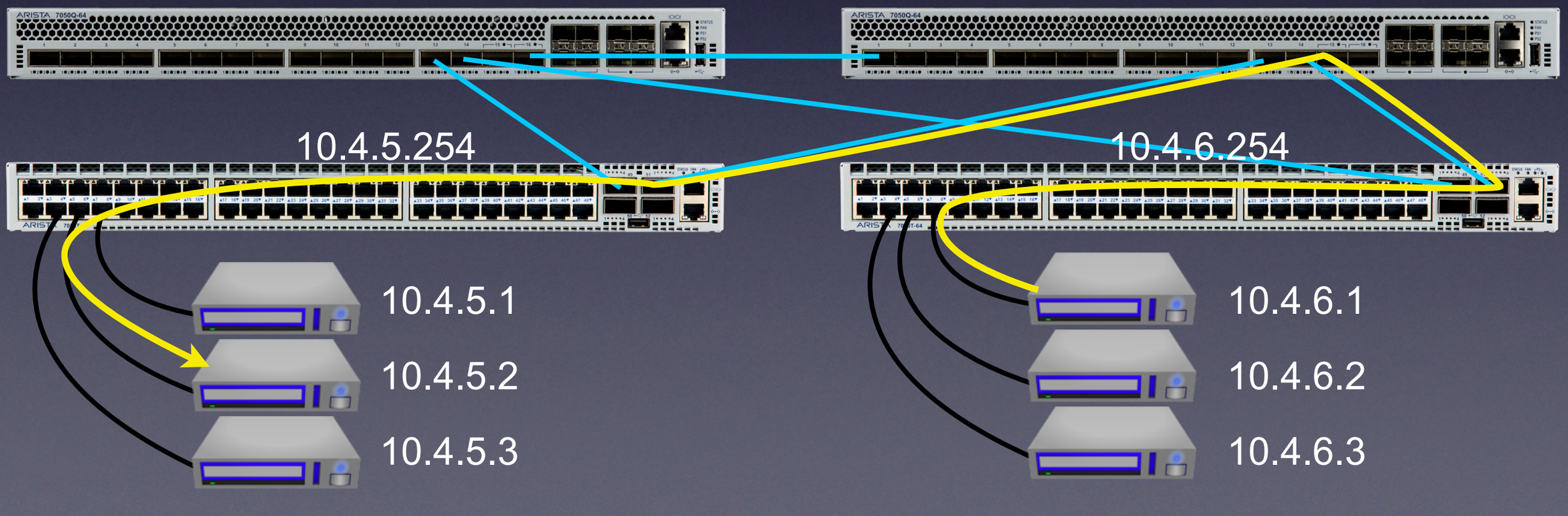
Manually Mitigating Host Failures



Process is brittle and not bullet proof

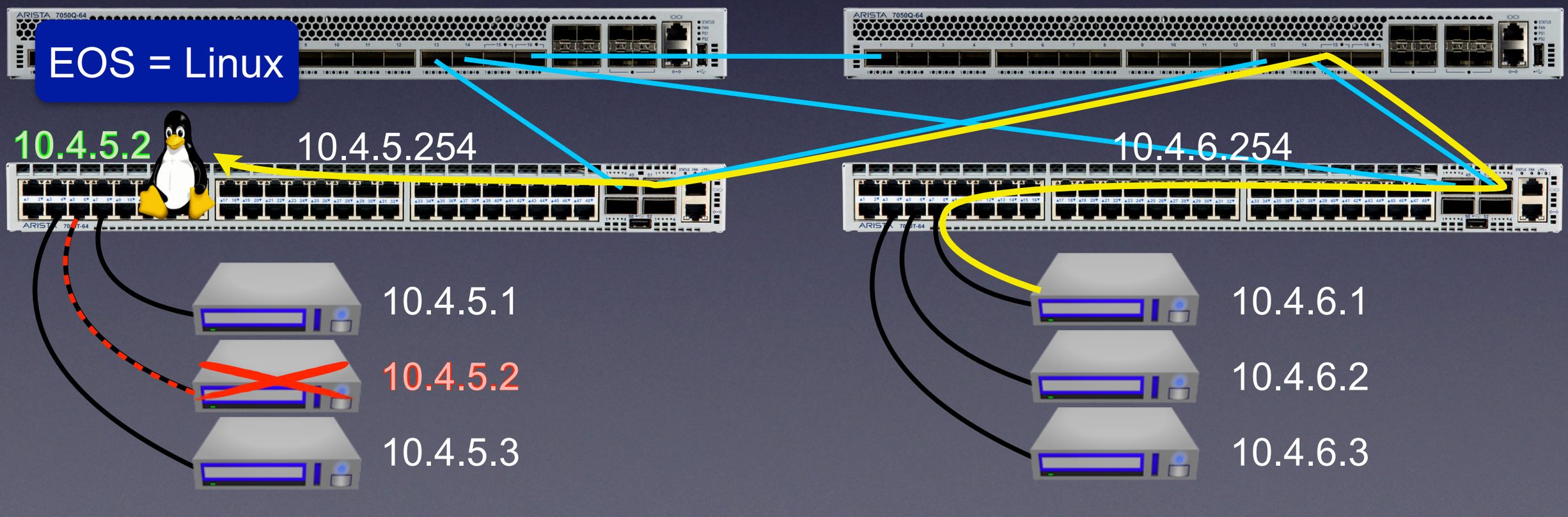
How Does This Work?

- Redirect traffic to the switch
- Have the switch's kernel send TCP resets
- Immediately kills all existing and new flows



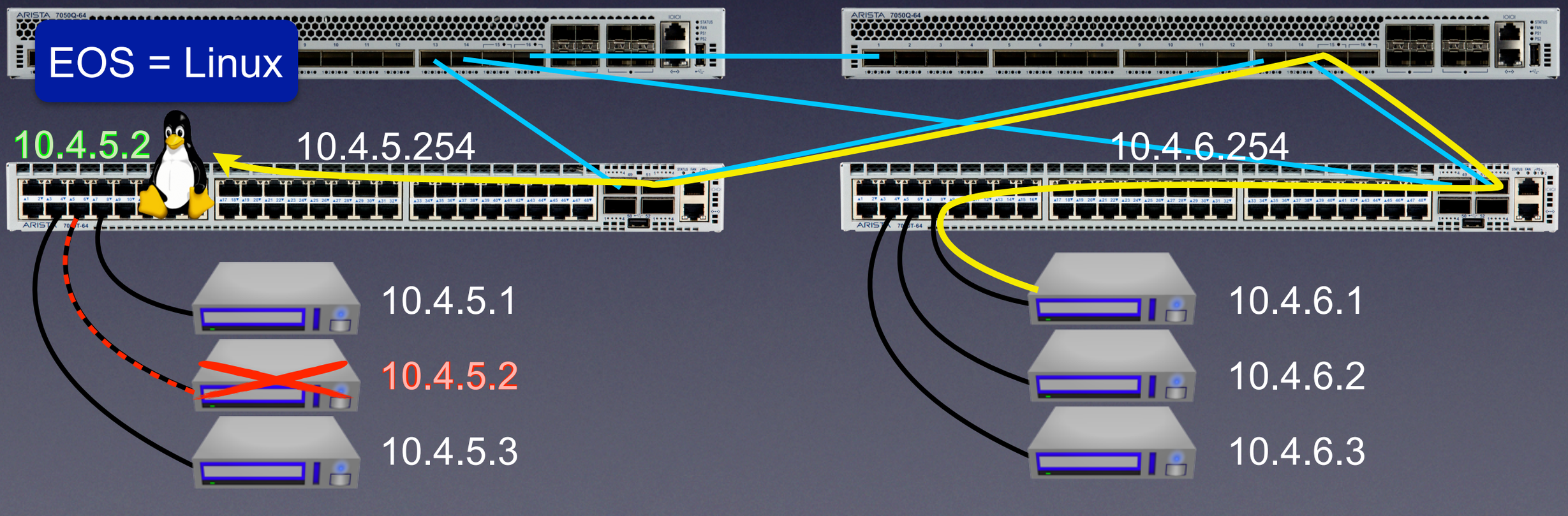
How Does This Work?

- Redirect traffic to the switch
- Have the switch's kernel send TCP resets
- Immediately kills all existing and new flows



Fast Server Failover

- Switch learns & tracks IP ↔ port mapping
- Port down → take over IP *and MAC* addresses
- Kicks in as soon as hardware notifies software of the port going down, within a few *milliseconds*
- Port back up → resume normal operation
- Can also run custom shell script on each event



Thank You

We’re hiring in SF, Santa Clara, Vancouver, and Bangalore

ARISTA

Benoît “tsuna” Sigoure
Member of the Yak Shaving Staff
tsuna@aristanetworks.com

@tsunanet