SDN and IPv6 – Better Together?

Ivan Pepelnjak (ip@ipSpace.net)
Network Architect

ipSpace.net AG

Who is Ivan Pepelnjak (@ioshints)

Past

- Kernel programmer, network OS and web developer
- Sysadmin, database admin, network engineer, CCIE
- Trainer, course developer, curriculum architect
- Team lead, CTO, business owner

Present

- Network architect, consultant, blogger, webinar and book author

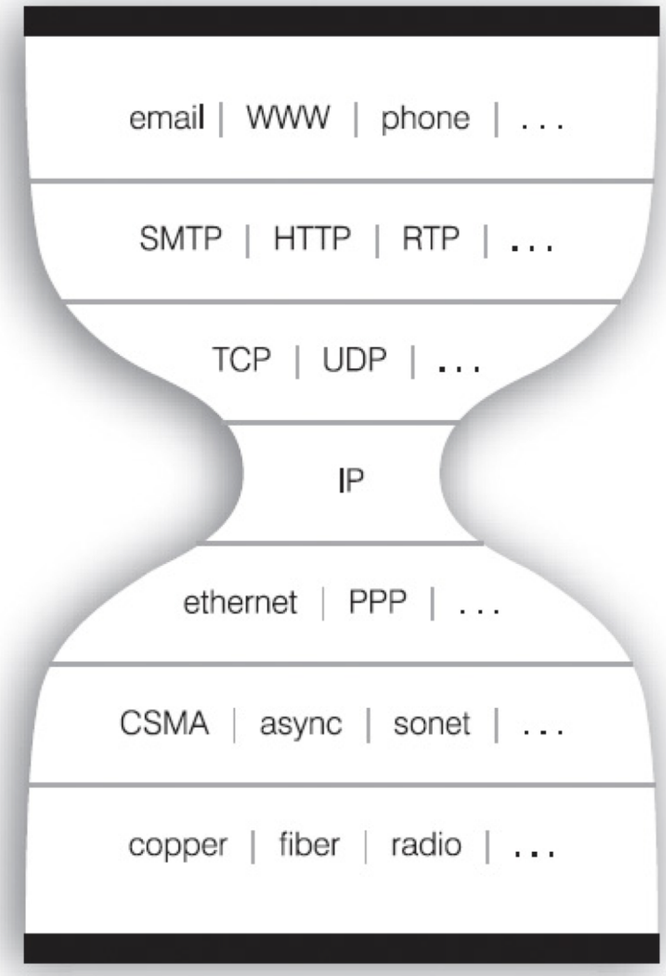
Focus

- SDN and network automation
- Large-scale data centers, clouds and network virtualization
- Scalable application design
- Core IP routing/MPLS, IPv6, VPN

CISCO CCIE EMERITUS

vmware vEXPERT

Before We Start: In Case You Haven't Noticed



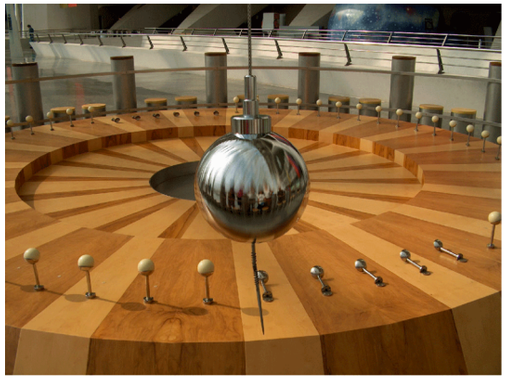
Networking hasn't changed much in the last 40 years

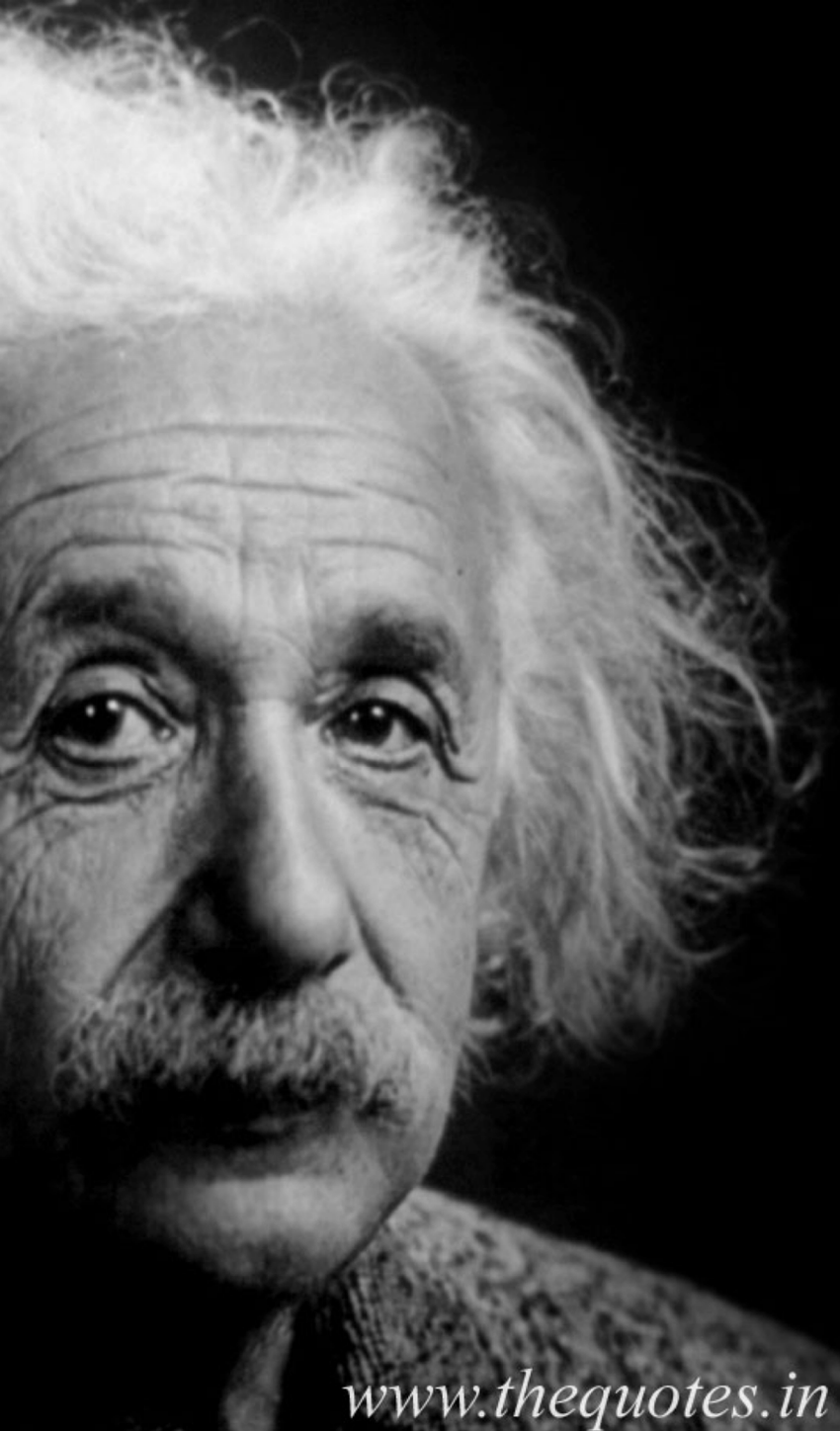
- We lost a few bits and pieces below IP
- We got 128 bit addresses instead of 32 bit addresses
- Everything runs on Ethernet these days

… and every 5 years someone reinvents large-scale bridging

… and causes a few large-scale meltdowns

… and then the pendulum swings back





Insanity: doing the same thing over and over again and expecting different results.

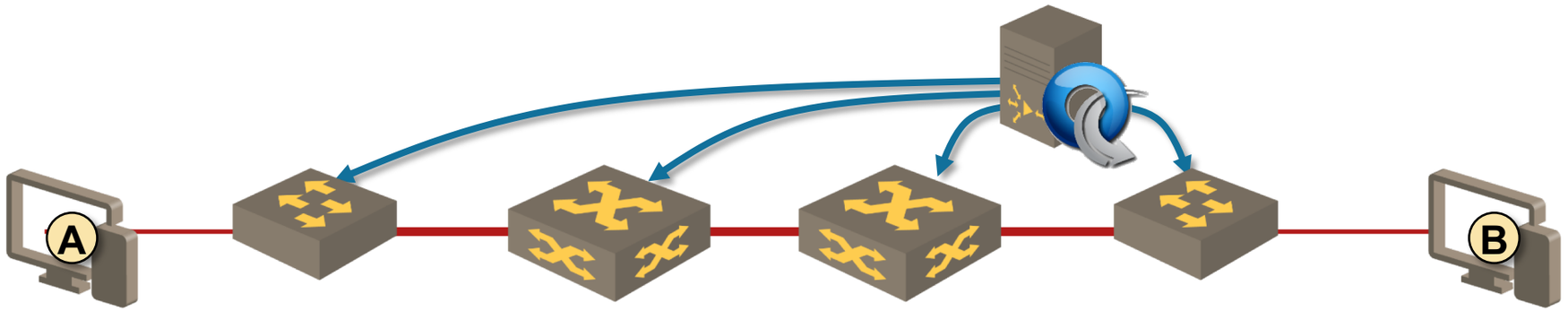
Albert Einstein

Can We Make SDN Better With IPv6 (or Vice Versa)

What Exactly Is SDN?

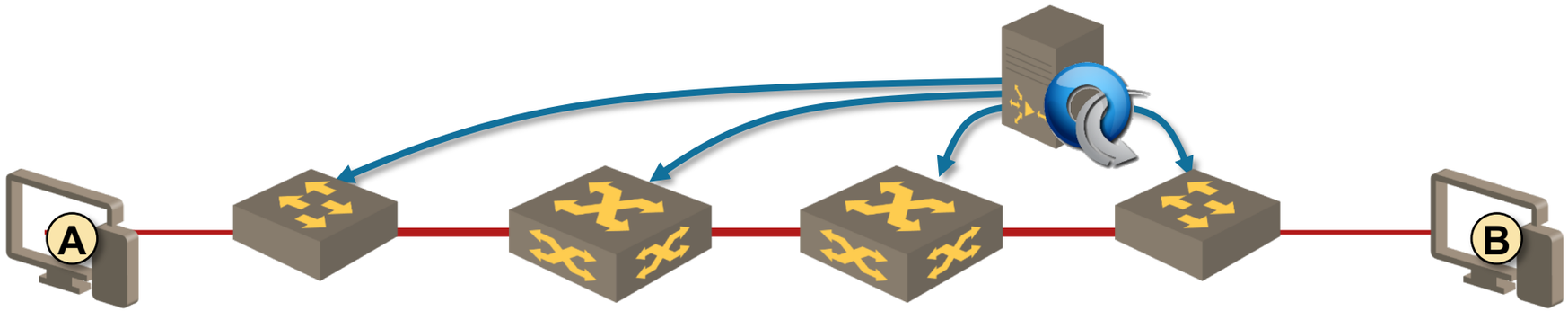
The Madness Started in March 2011

SDN is the physical separation of the network control plane from the forwarding plane, and where a control plane controls several devices



Remember SNA, Frame Relay, ATM? Yeah, they worked really well.

Challenges of Centralized Control Plane



Conceptual challenges

- Out-of-band control plane network
- No distributed intelligence ➔ no resilience to failures
- Controller is the central point of failure
- Total loss of control-plane protocols after a controller failure
- Lack of shared fate (requires end-to-end OAM)

Real-life challenges

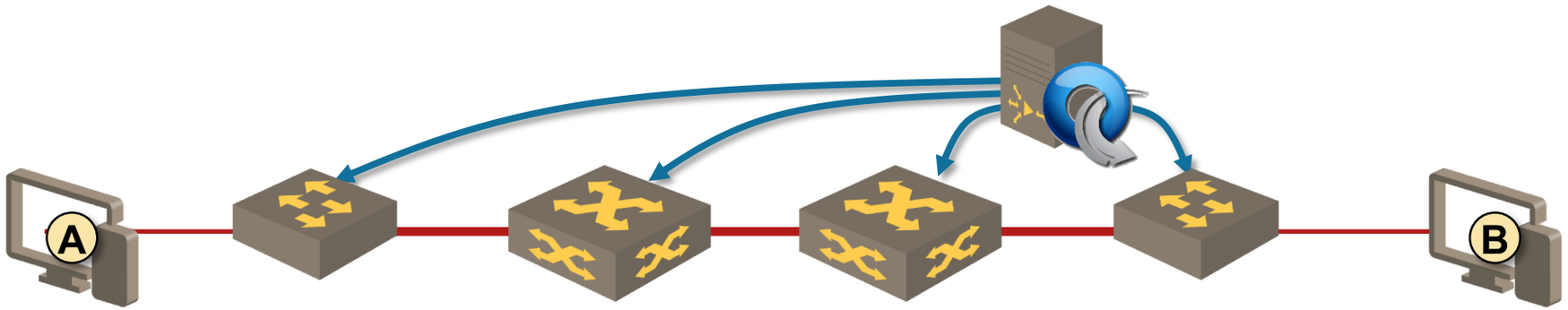
- Poor OpenFlow implementations (very limited multi-table support)
- Limited TCAM sizes (few thousands)
- Low TCAM update speed (less than thousand entries per second)
- Slow switch-to-controller channel due to underpowered switch CPUs

More in *OpenFlow Deep Dive* webinar

Every old idea will be proposed again with a different name and a different presentation, regardless of whether it works.

RFC 1925, Rule 11

How Does This Apply to IPv6?



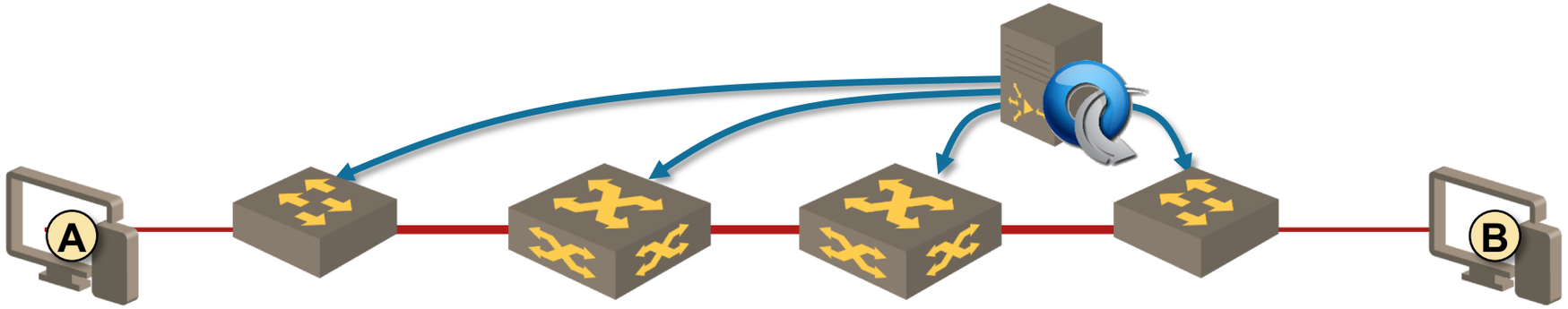
- Doesn’t matter whether you process AppleTalk, DECnet, IPX, IPv4 or IPv6
- IPv6 was always considered a second-class citizen (remember: the craze started in 2011)
- OpenFlow didn’t support IPv6 at all (for a long time)

Takeaway: once we change the forwarding paradigm, we can be creative about what bits mean

More in *OpenFlow Deep Dive* webinar

Getting Creative with Bits

What Have We Got in IPv6?



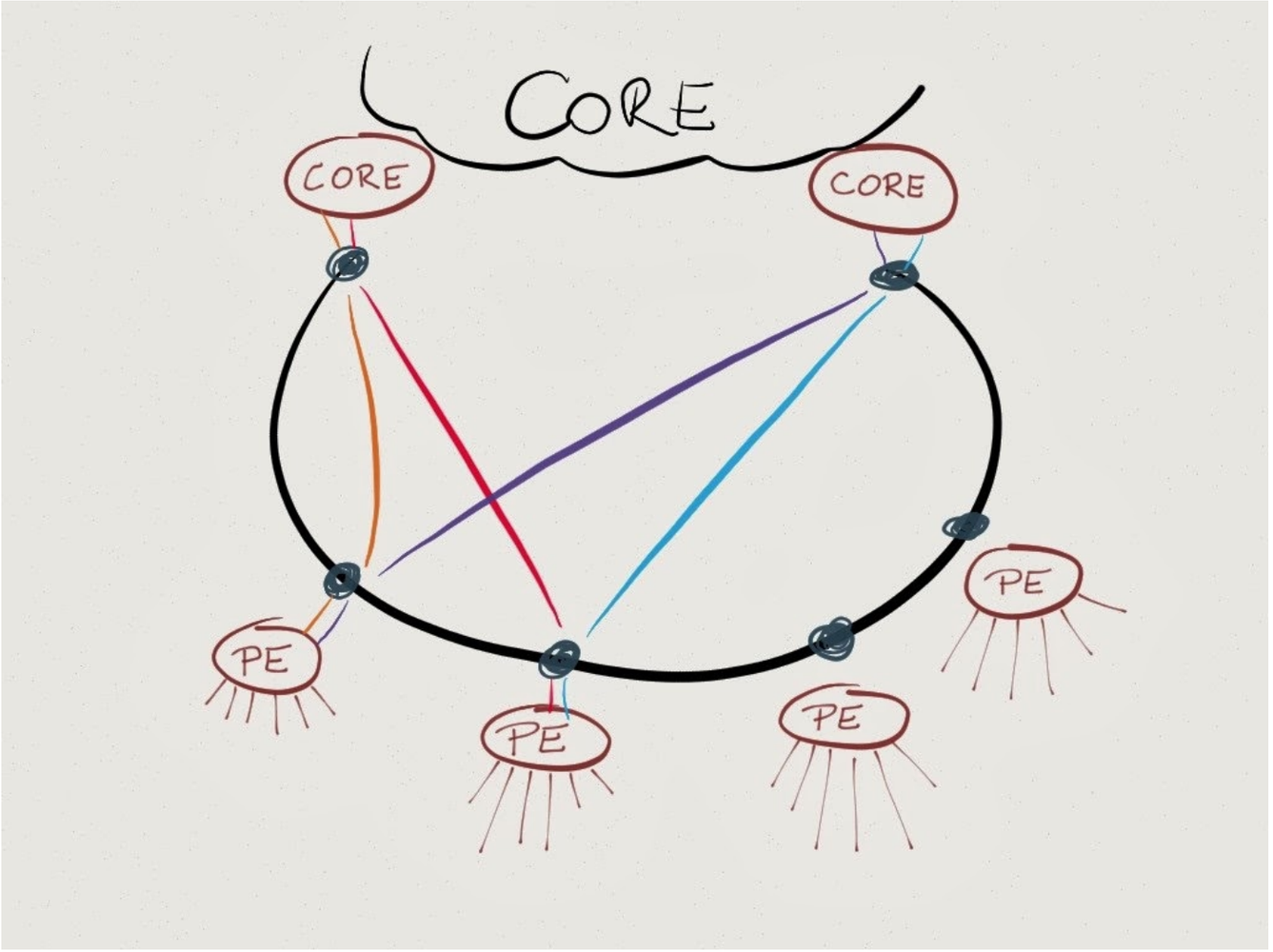
- Large addresses
- Extension headers
- Flow labels

More in *Enterprise IPv6 101* webinar

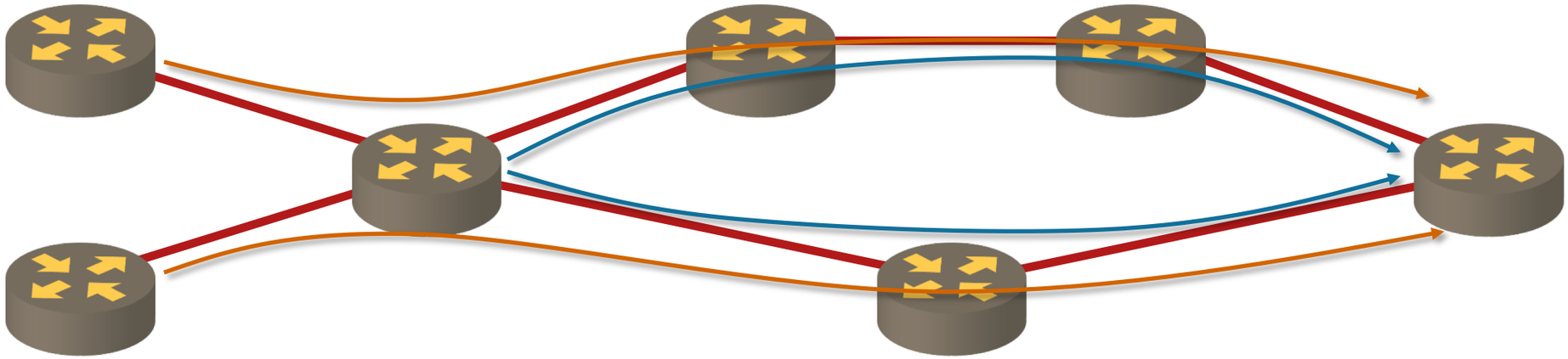
Radical IPv6-Based Redesign: Deutsche Telekom Terastream

- IPv6-only transport network
- IPv4-as-a-Service
- Perfectly symmetrical structure
- No MPLS, no TE

Customer services encoded in IPv6 address bits



Segment Routing: Source Routing Reinvented



Traffic engineering (like MPLS TE) is a hard problem

- Bandwidth estimates are imprecise
- Traffic paths are unpredictable and may change after failure/recovery
- Reservations must be kept in the core routers
- Continuous state refresh (RSVP-TE)

What if we would…

- Use a controller to compute paths (Frame Relay says *hi*)
- Use some mechanism to indicate loose path through the network in the packet (let's call it *Segment Routing*)
- Install paths in head-end routers

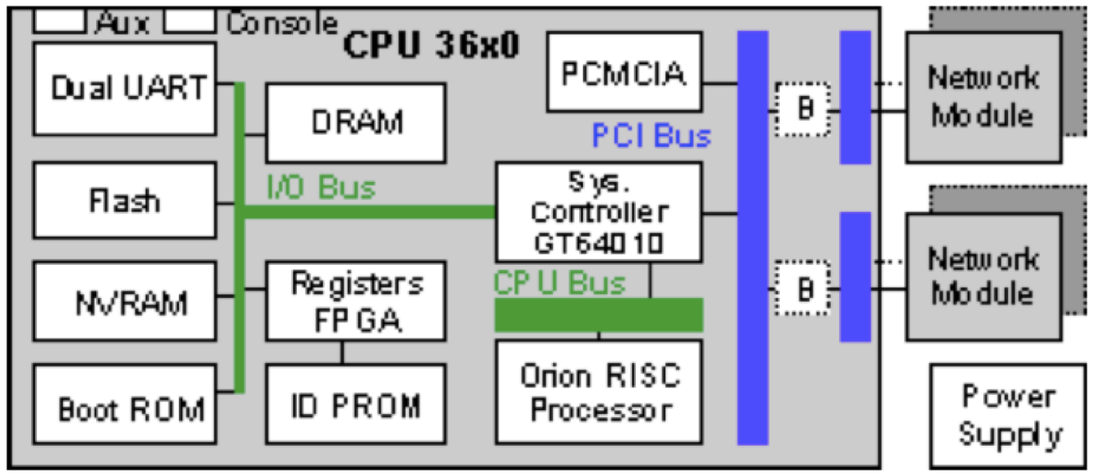
Congratulations, you reinvented Token Ring SRB

Software-Defined Packet Forwarding

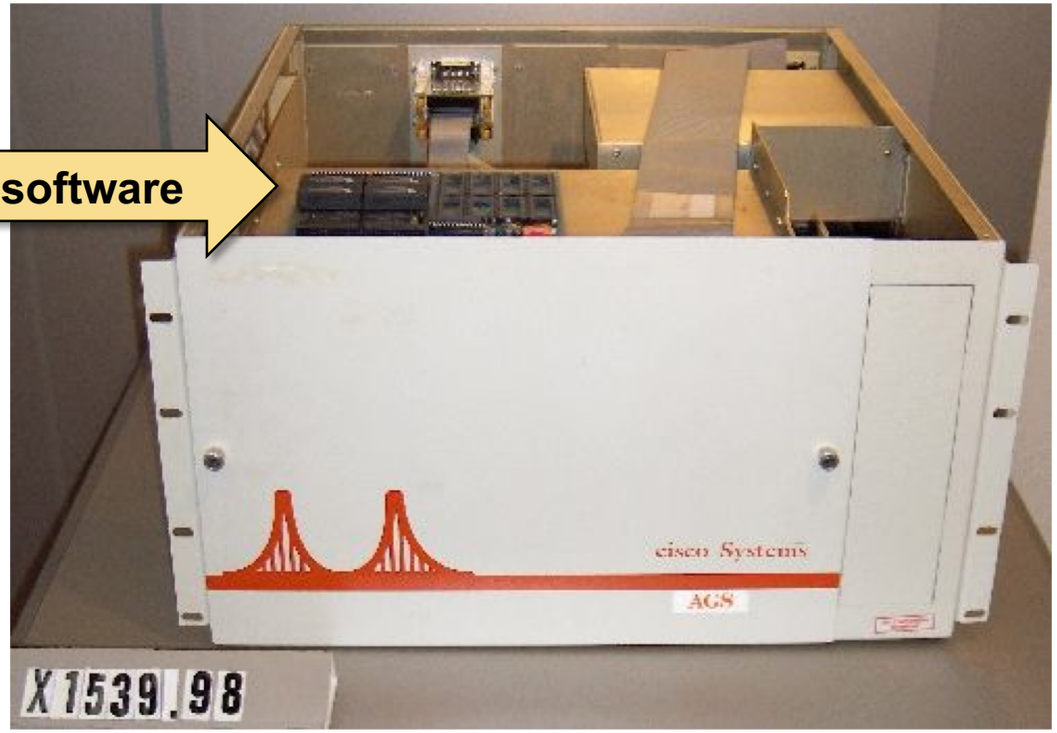
SDN is packet forwarding done in software (on x86 platform)

When I Was Still Young…

… we did all packet forwarding in software … and most low-end network devices still do.



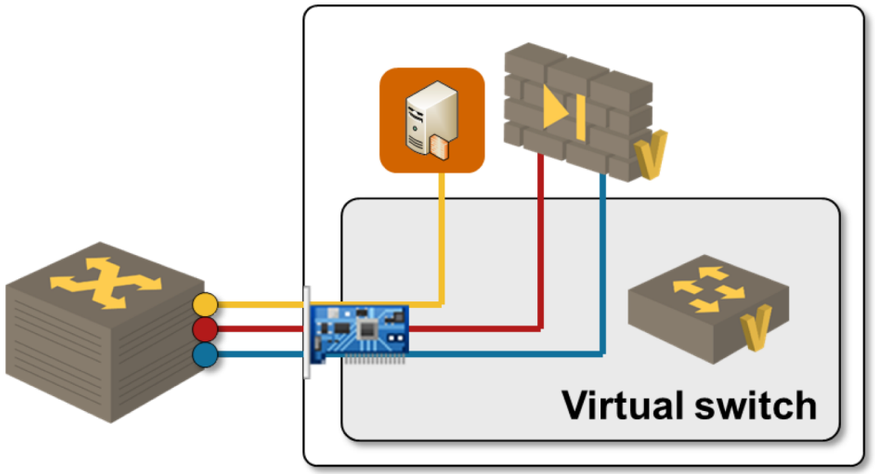
Here be software



Cisco AGS at Computer History Museum
Source: Evilrouters.Net

Huge Success (When Applied Correctly)

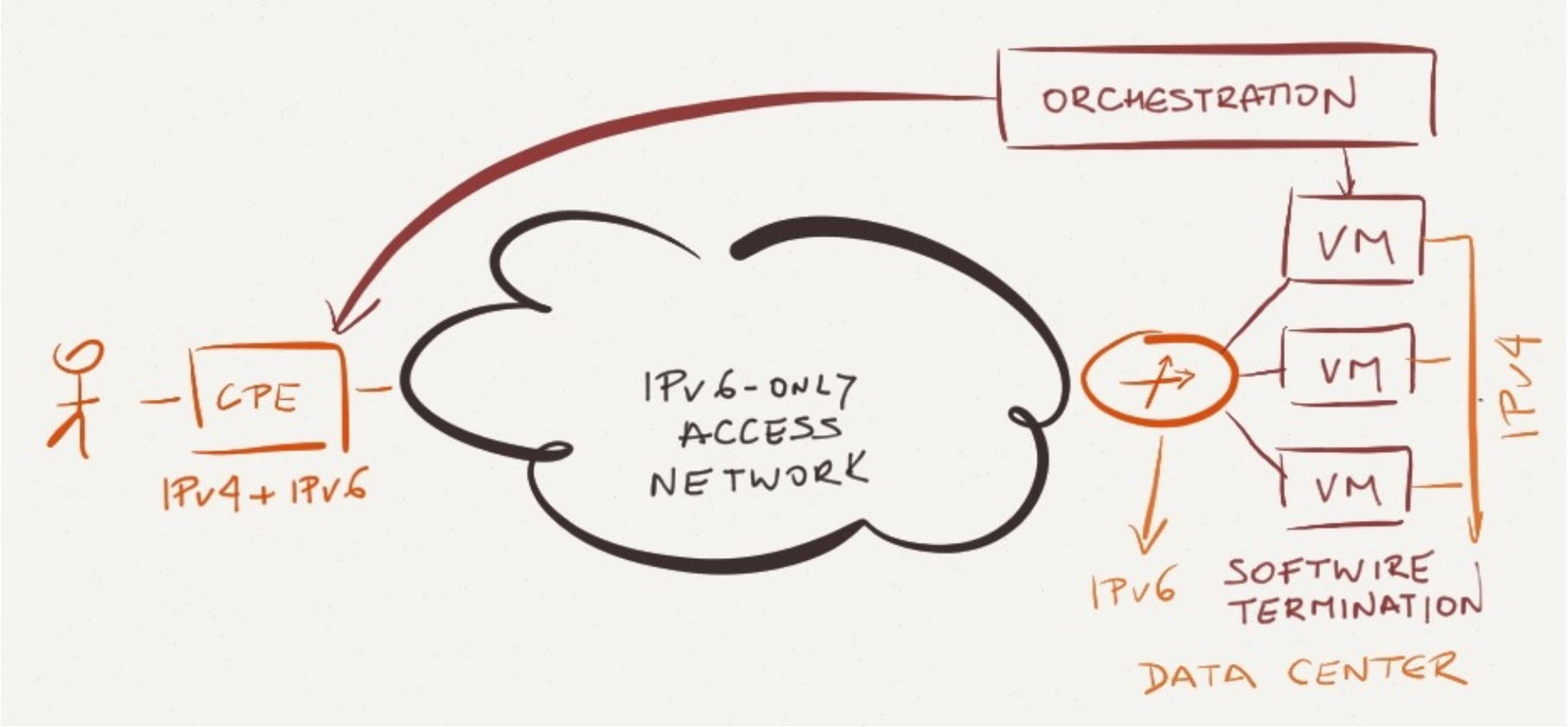
- 20 Gbps per core, 100+ Gbps per x86 server
- Innovative appliances (example: L2VPN over IPv6, 4-over-6 tunneling…)
- Major networking vendors offering virtualized devices with DPDK/6WIND or equivalent



More in *Network Function Virtualization* webinar

Service Provider Use Case: Lightweight 4over6 in Terastream

- Reduce network complexity ➔ IPv6-only access network
- Flexible IPv4 support ➔ VM-based pseudowire termination



More in *SDN Use Cases* webinar and *Software Gone Wild* Episodes 52 (4over6) & 17 (L2VPN over IPv6)

SDN is packet forwarding done in software (on x86 platform)

SDN = Whitebox Switching

All Data Center Switching Vendors Use Merchant Silicon

BROADCOM®

- High-speed packet forwarding is becoming commodity
- Limited differentiation in hardware ➔ developing custom ASICs makes little sense
- Major vendors focus on software, integration or logistics

More in *Market Overview* part of *Data Center Fabrics* webinar

Software / Hardware Disaggregation

- Hardware costs are 30-40% of the product costs (gross margin of networking vendors is above 60%)
- Software and support are the really expensive parts (and yet we're all buying boxes)
- Why can't we buy hardware and software as separate items?

Benefits:

- Increased flexibility (reuse the same hardware)
- Simplified sparing

SDN is whitebox switching (running software on third-party cheap hardware)

Real Benefits

Install your own software on networking devices

- Control-plane daemons
- Customized telemetry
- Push agents
- Pilot data-plane implementations (SR-IPv6)

Linux everywhere

- Unified management of servers and network devices
- Common tooling
- Common control-plane functionality (including shared bugs)

More in *Open Networking* and *Cumulus Linux* webinars, explore also *Software Gone Wild* podcast

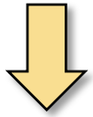
SDN = Network Automation

SDN is an approach to computer networking that allows network administrators to manage network services through abstraction of lower level functionality

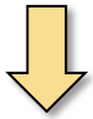
Everything Well-Defined Can Be Automated

How About IPv6 Deployments?

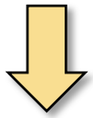
Educate



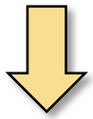
Research



Design



Test



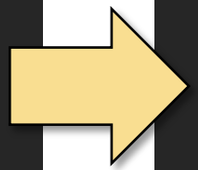
Deploy

IPv6 Deployment Is Utterly Boring

IPv6 configuration is very similar to IPv4 configuration

- Slightly different commands and caveats
- Different addresses
- Deploying IPv6 is boring…
- … and boredom results in mistakes

```
interface Loopback0
 ip address 10.0.1.1 …
 ip ospf 1 area 0
```



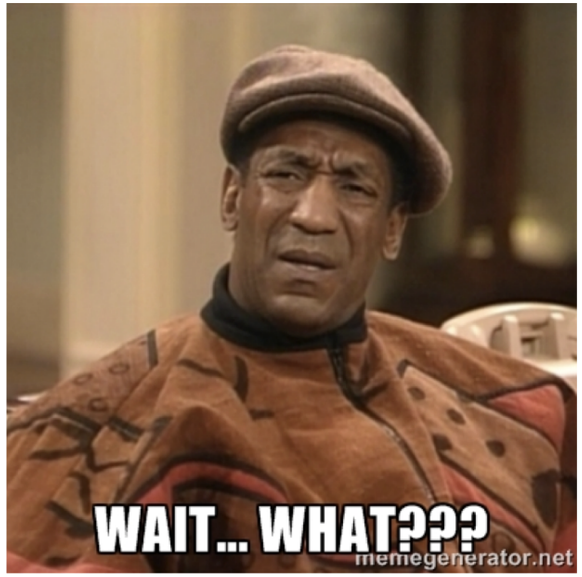
```
interface Loopback0
 ip address 10.0.1.1 …
 ip ospf 1 area 0
 ipv6 address FD00:DB8:1/128
 ipv6 ospf 1 area 0
```

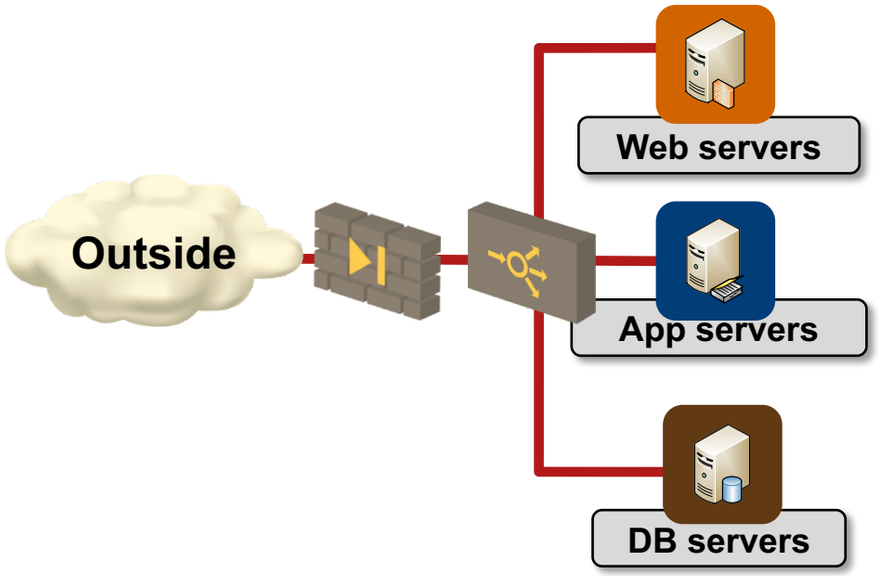
Failures Are Expensive: Real-Life Example

- Enable IPv6 in database segment ➔ OK
- Enable IPv6 in other segments ➔ OK
- Test connectivity ➔ OK

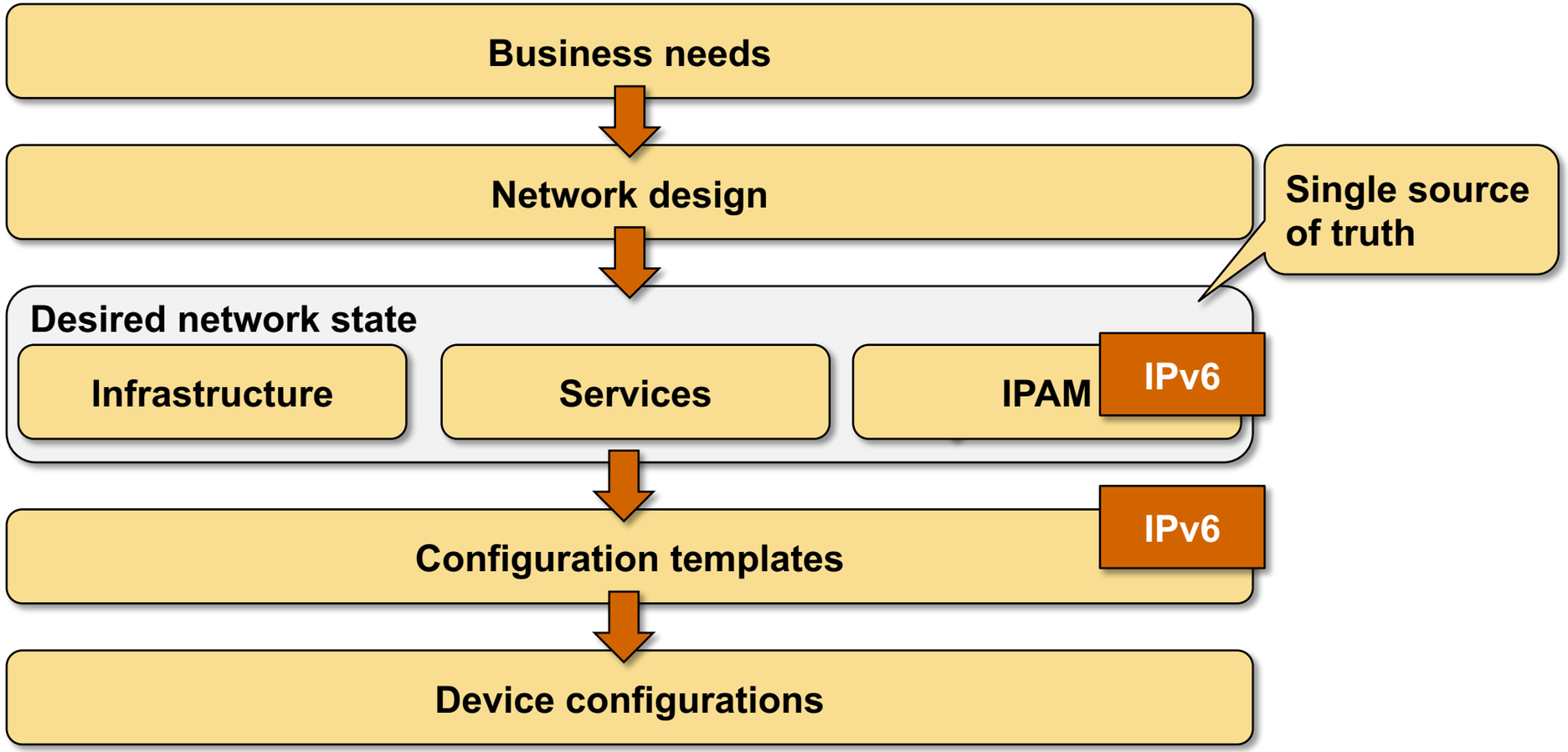
Weeks later…

- Add DNS server AAAA record ➔ **CRASH**





In the Ideal World



Back on Planet Earth

```
upgrade fpd auto
version 15.0
service timestamps debug datetime msec
!
hostname PE-A
!
boot-start-marker
boot-end-marker
!
logging buffered 4096
!
interface GigabitEthernet0/1
  description to PE1
  ip address 10.0.0.5 255.255.255.252
```

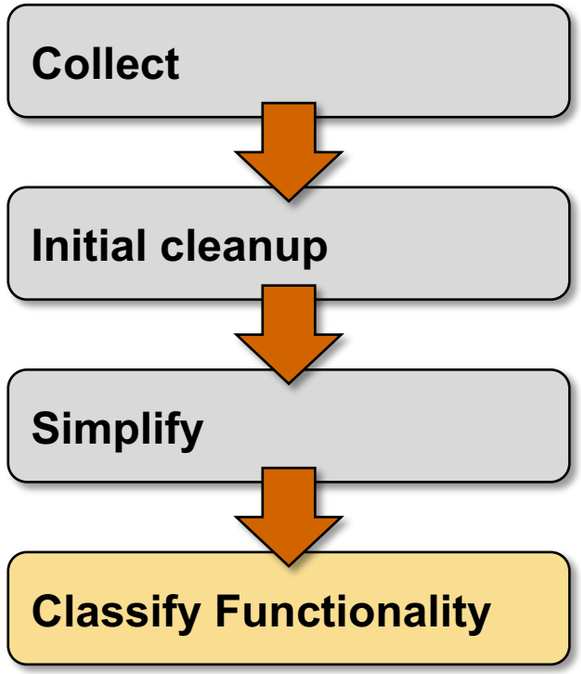
The only source of truth

Prepare for Migration: Functionality Classification

Identify parts of configuration that have to be migrated to IPv6

Potential classification outcomes:

- Functionality is not IP-dependent
- The functionality will remain on IPv4
- We need dual-stack functionality
- Functionality will move to IPv6



More in *Automating IPv6 Deployments* part of *Network Automation Use Cases* webinar

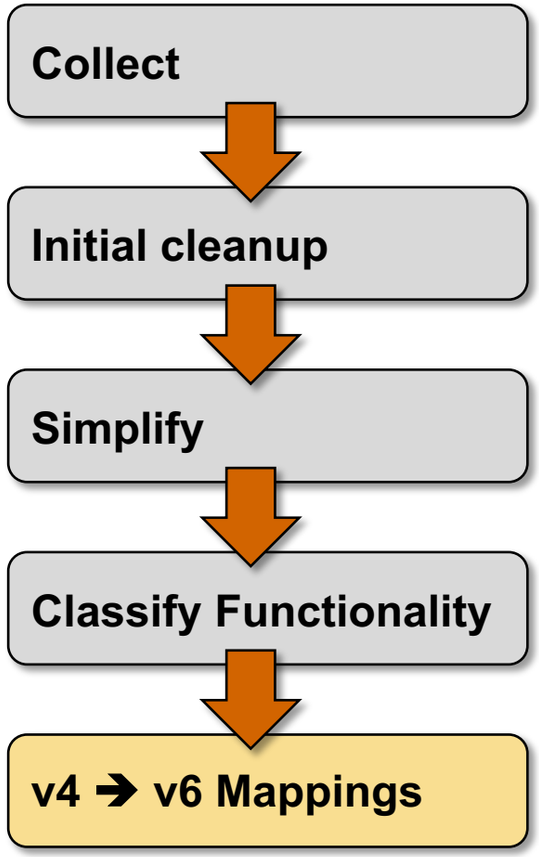
Prepare for Migration: v4 ➔ v6 Mappings

Add IPv6 equivalent of IPv4 configuration for every bit of dual-stack functionality

- Sounds simple
- Need well-defined v4 ➔ v6 mapping
- Where will you get it?

We need single source of (addressing) truth

```
interface Loopback0
 ip address 10.0.1.1 …
 ip ospf 1 area 0
 ipv6 address FEC0::CCCC:1/128
 ipv6 ospf 1 area 0
```



More in *Automating IPv6 Deployments* part of *Network Automation Use Cases* webinar

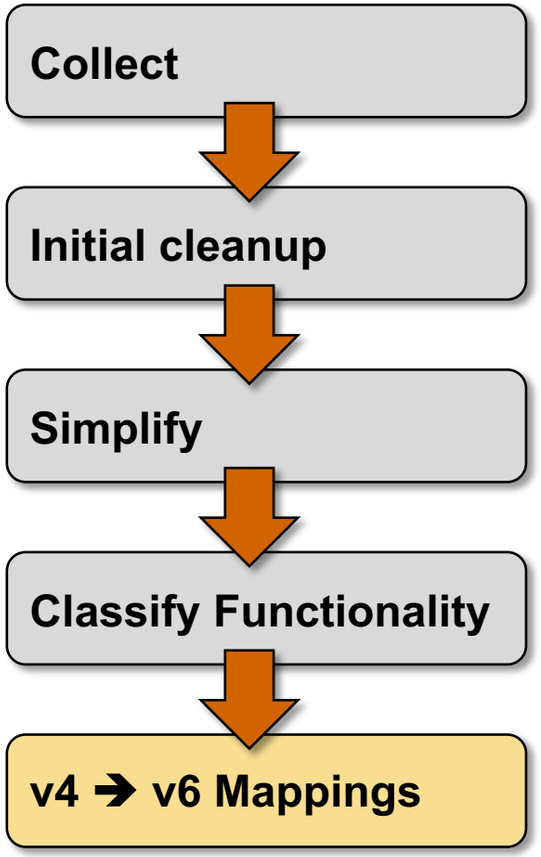
v4 ➔ v6 Mappings: Recovering from Worst Case

Assumptions:

- No IPAM (or reliable Excel)
- Device configurations are the only source of truth

Build v4-to-v6 mappings

- Analyze router configurations
- Scrape subnet information from interfaces
- Use simple algorithmic v4 ➔ v6 mapping to build IPv6 subnets and host addresses



Unfortunately we can't use DNS lookups

Use IPv6 Deployment as an Excuse to Build Source-of-Truth

Even More Playing-with-Bits

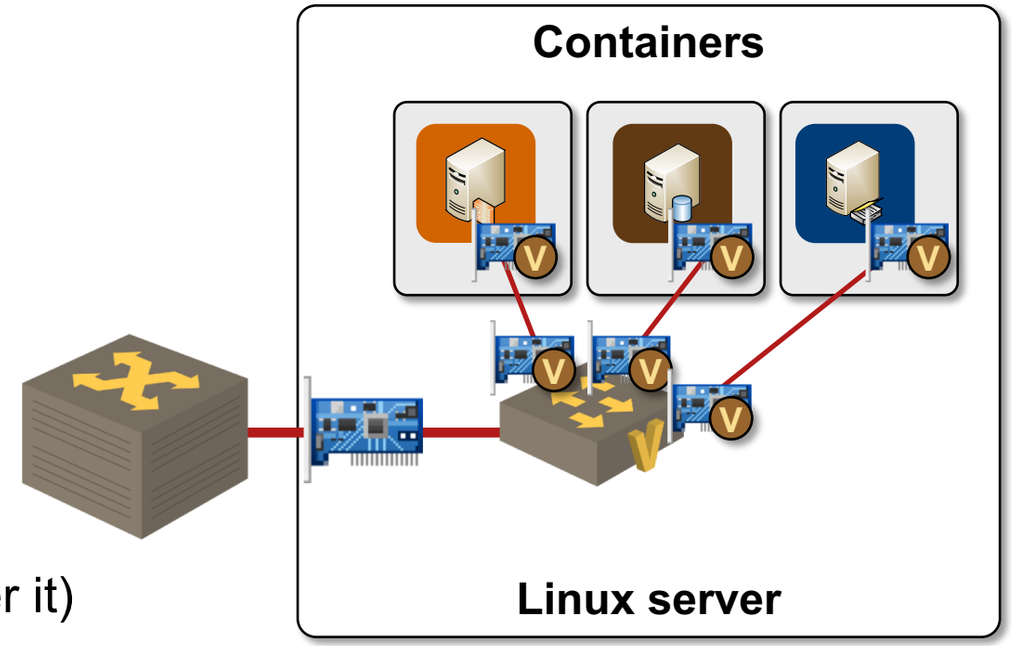
Container and VM Networking

Most Docker networking implementations use vEth pairs

- Virtual cable connecting container to virtual switch
- Docker's implementation uses Linux bridge (not OVS)

Docker IPv6 Networking

- A /64 prefix is assigned to Linux bridge internal network
- ToR switch gets a /64 route toward each host (BGP, DHCPv6 PD…)
- It's possible to use LLA on physical network (and run BGP over it)



Remember CLNP?

- Addresses were assigned to hosts (not interfaces)
- Interfaces were unnumbered
- Hosts were advertising their addresses with ES-IS

Identifier-Locator Addressing for IPv6

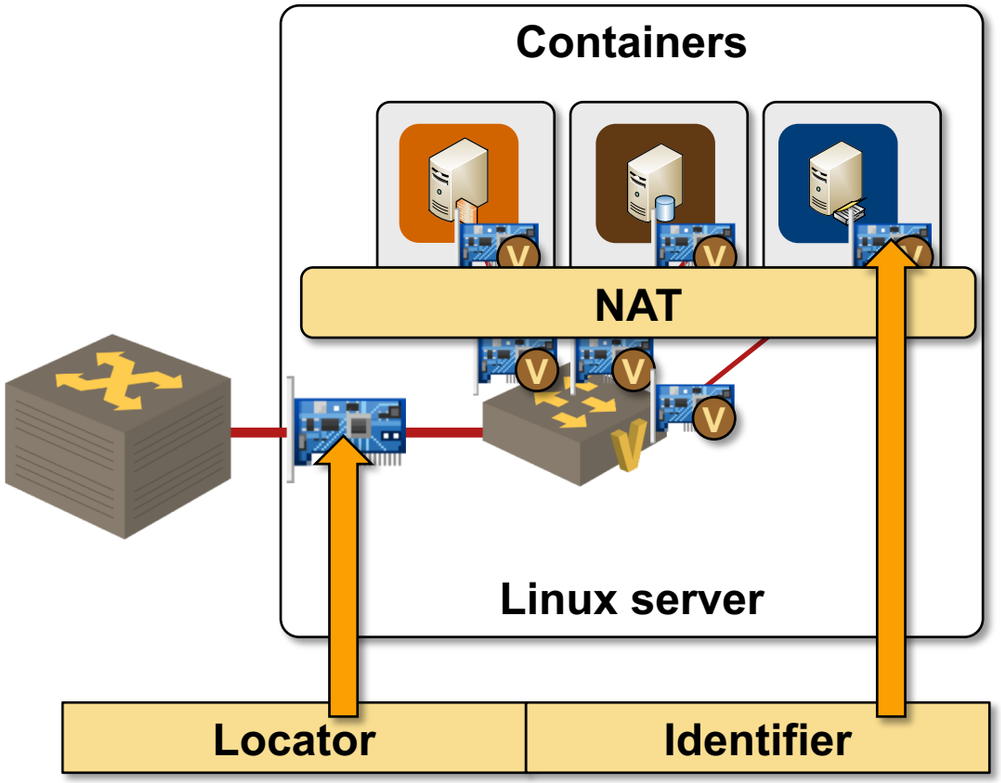
- Decent application architectures should use fixed addresses and service discovery
- But what if we'd fix the IPv6 address in a scalable manner?

Welcome to ILA

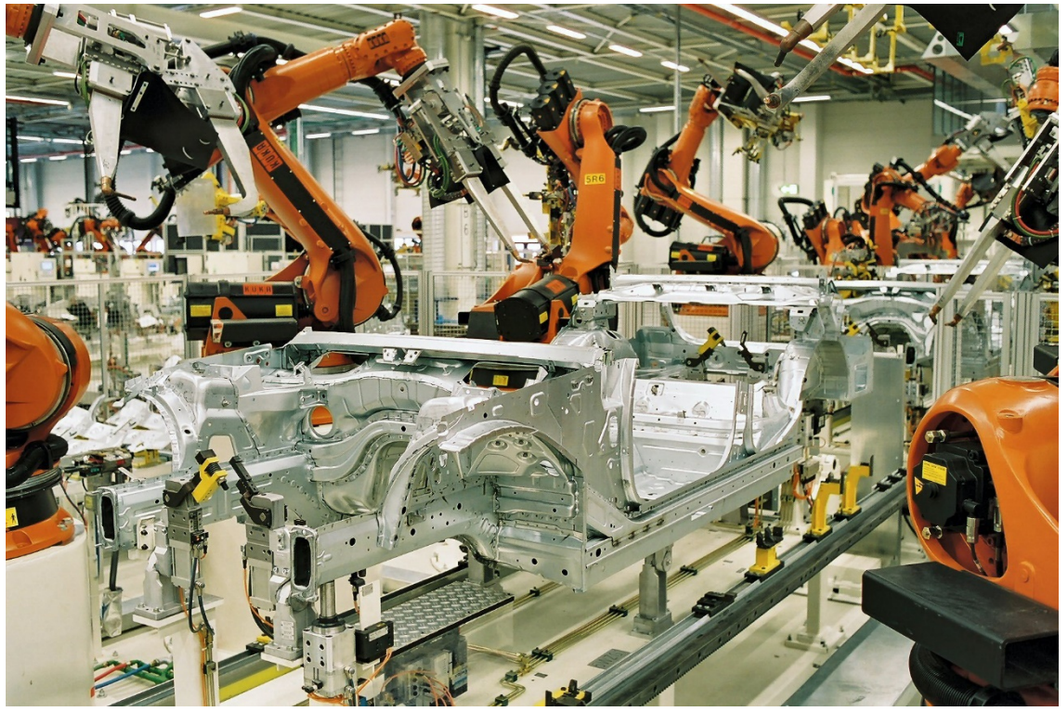
- Low-end 64 bits = endpoint identifier
- High-order 64 bits = location identifier

NAT anyone?

- Endpoints use *Standard Identifiers* (with fixed location part)
- Every host (hypervisor, container) contains mapping rules from standard identifier to endpoint current location
- NAT, right? Of course… at least it's stateless NPT



Back to the Big Picture



SDN Principles Revisited

What we would love to have

- Automated and consistent network services deployment
- Consistent policies
- End-to-end visibility
- Decisions made on centralized view of end-to-end visibility
- Automatic programming or configuration of network devices
- Automated response to events or changes in traffic or topology

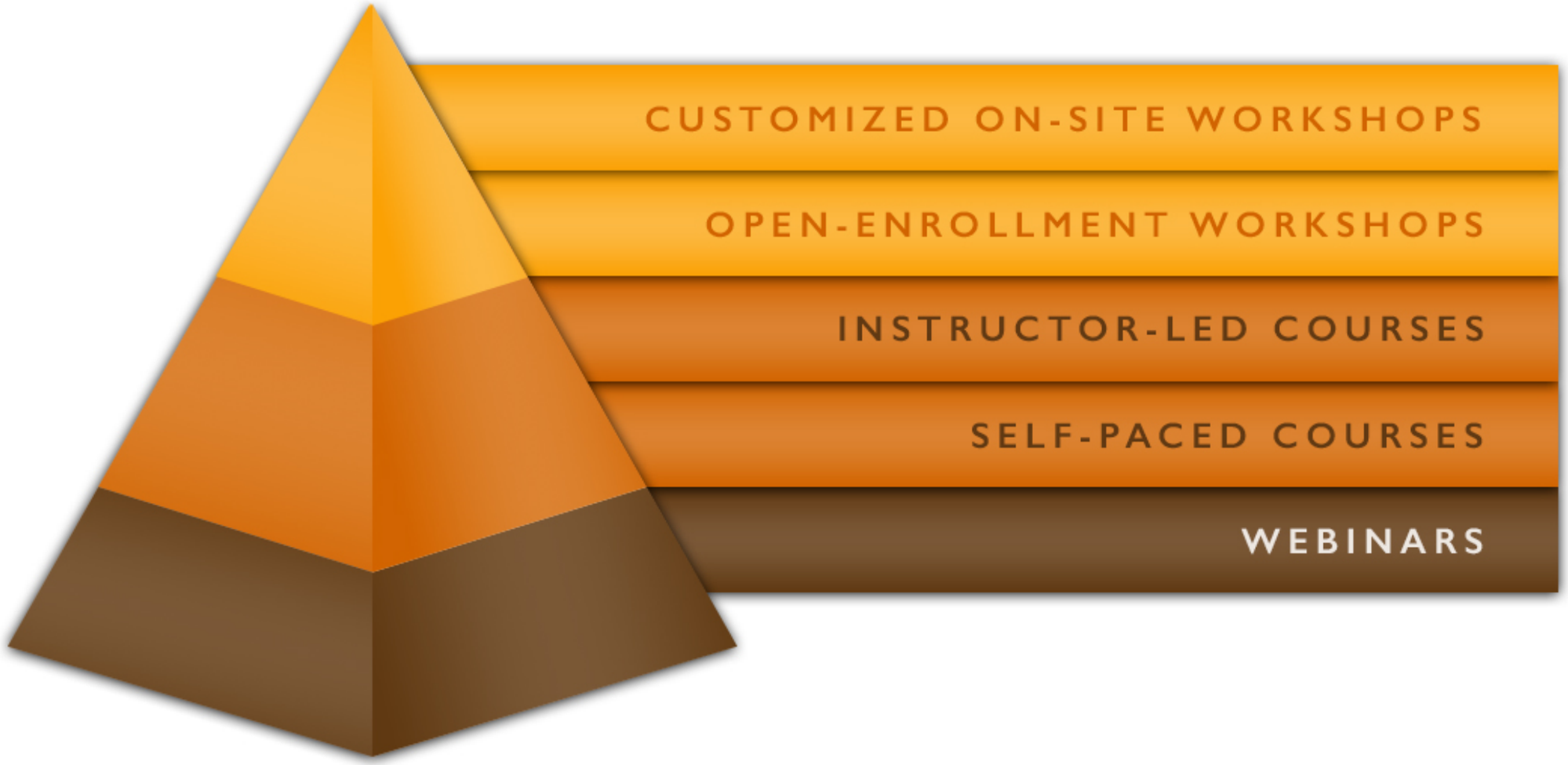
Build or Buy?

Others Made It Work…

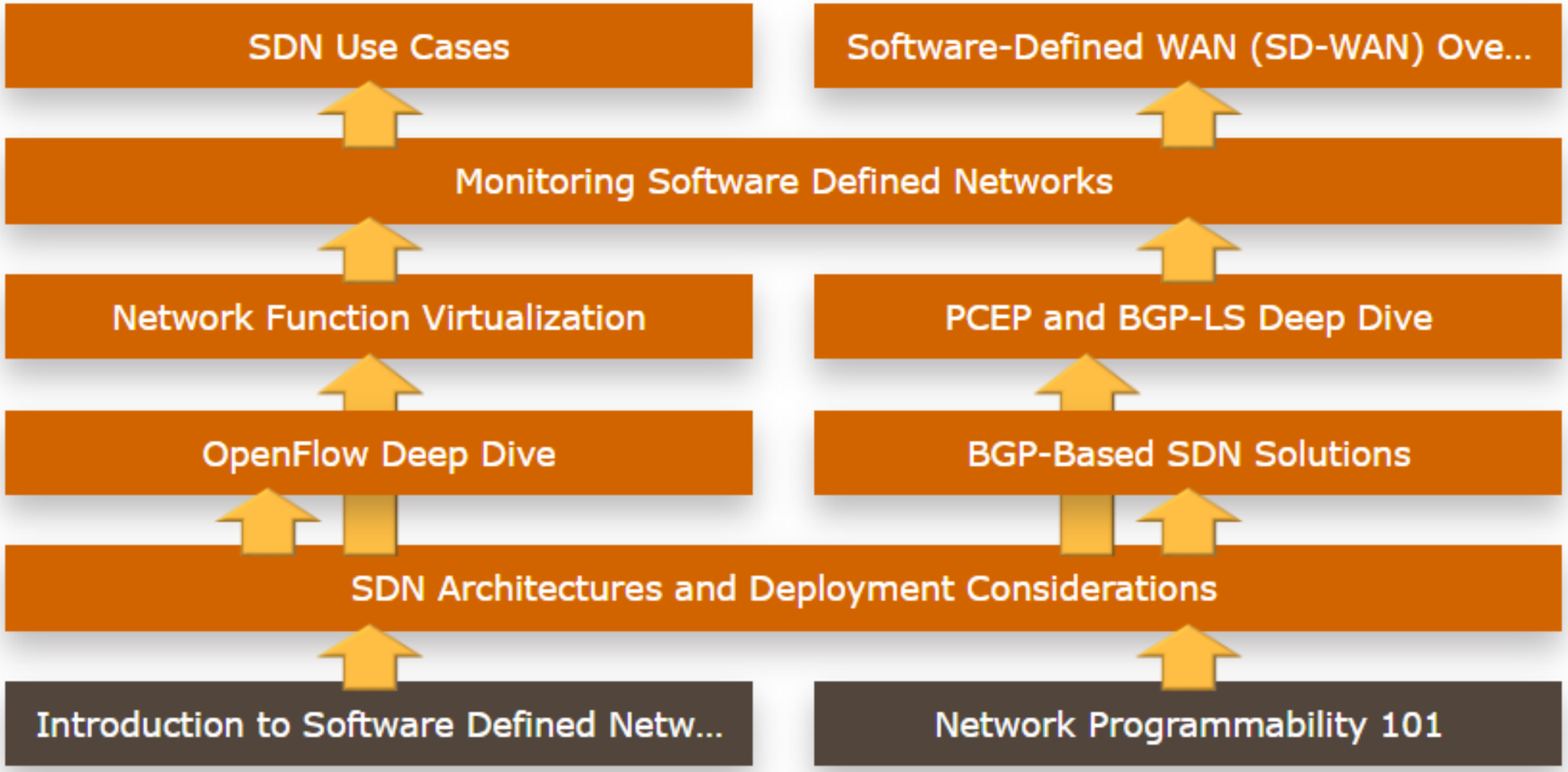
… When Will You?

More Information

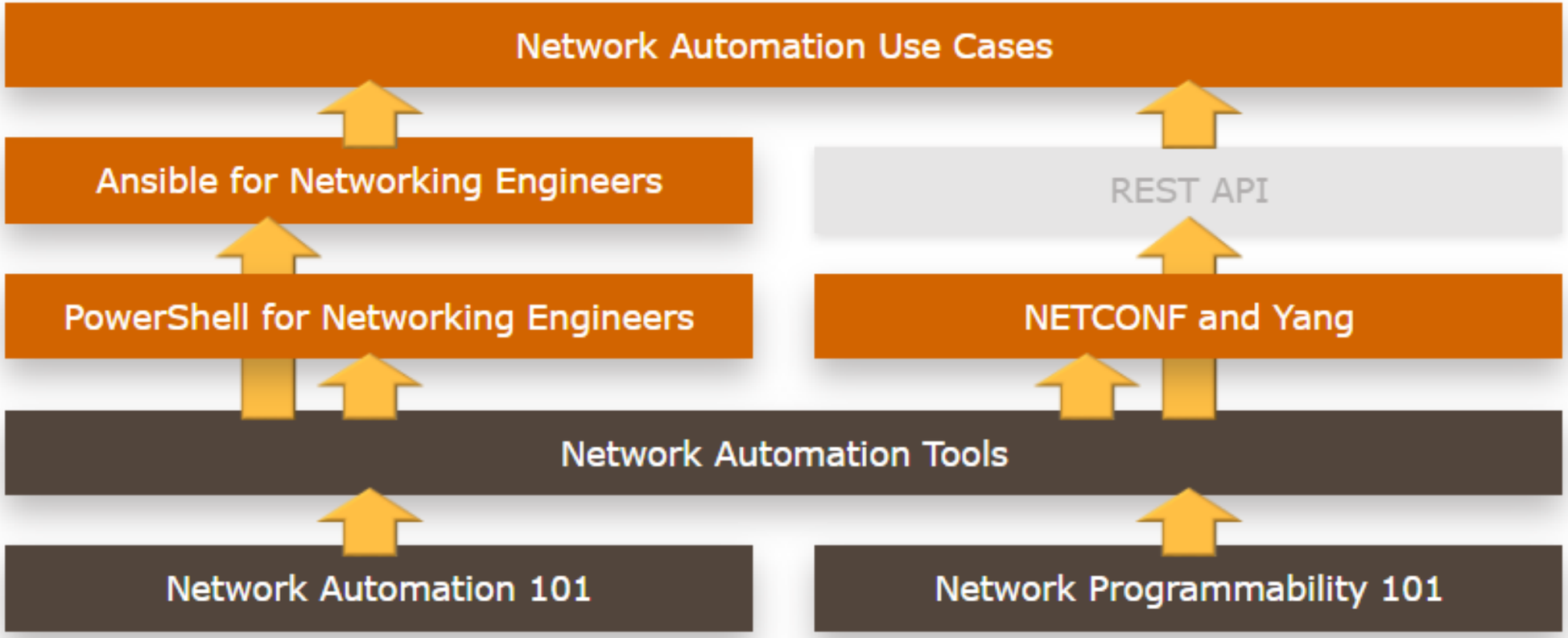
Learn At Your Own Pace



SDN Webinars



Network Automation Webinars



Questions?

Web:	ipSpace.net
Blog:	blog.ipSpace.net
Email:	ip@ipSpace.net
Twitter:	@ioshints

Automation:	ipSpace.net/NetAutSol
Data center:	ipSpace.net/NextGenDC
Webinars:	ipSpace.net/Webinars
Consulting:	ipSpace.net/Consulting