



PROJECT
CALICO

Simple, Secure, Scalable networking for
the virtualized datacentre

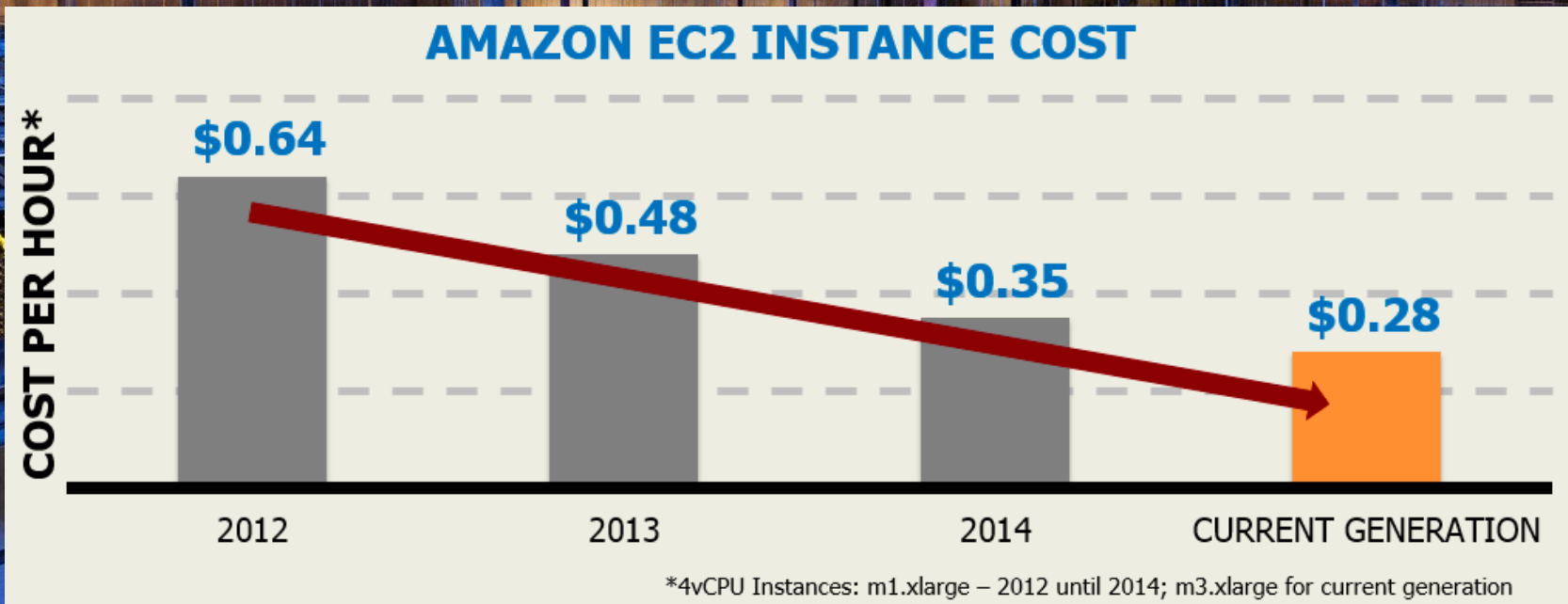
UKNOF 33

Ed Harrison
@eepyaich

19th January
2016

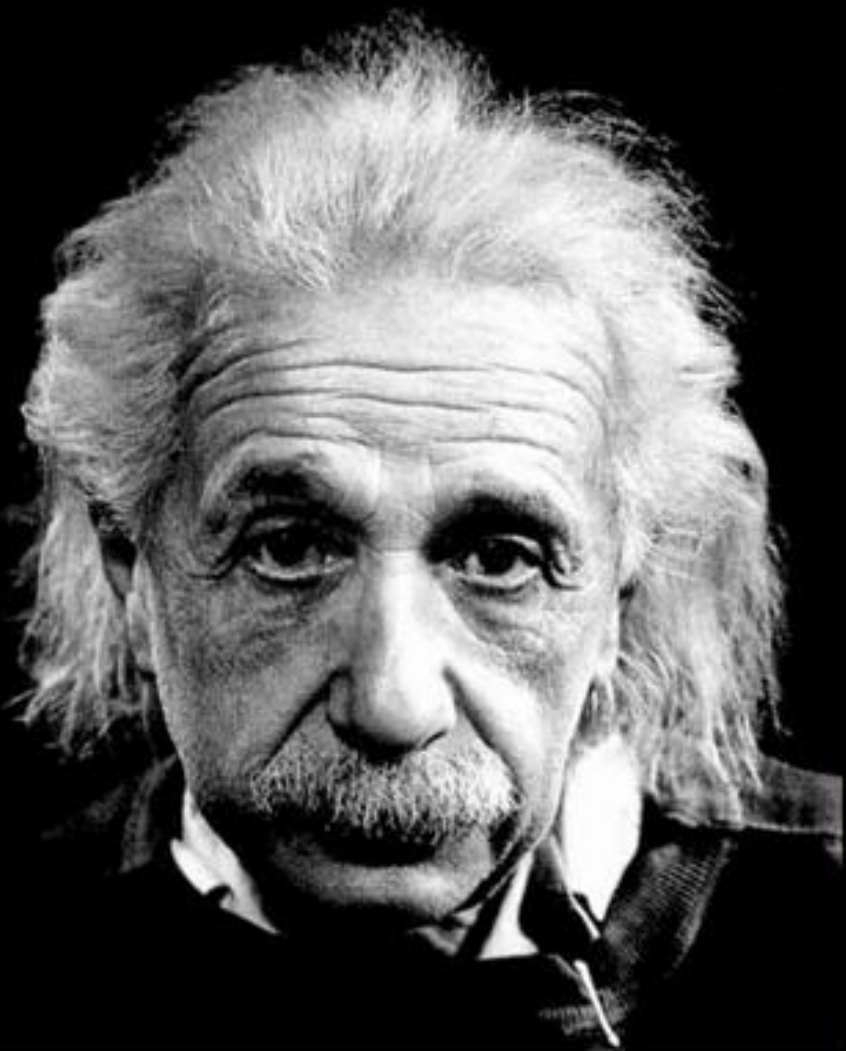
The Goal:

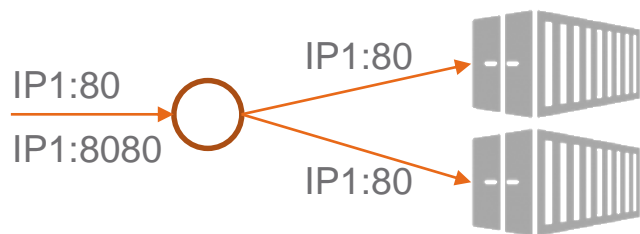
Hyperscale Efficiency



“Everything should be made
as simple as possible,
but not simpler.”

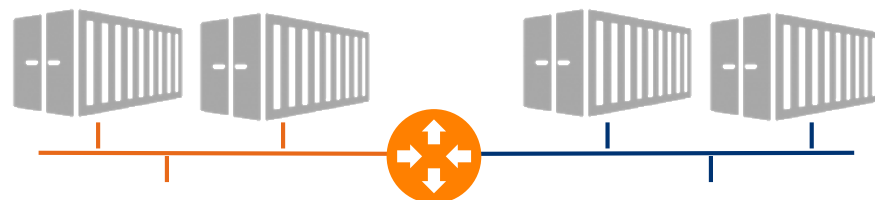
Albert Einstein





Port forwarding / NAT

- Simple
- Works “out of the box”
- Easily understood
- ... but not “real IP networking”
 - Won't work with all applications (e.g. IPsec)
 - Onerous port assignment constraints on applications
 - Requires **app developers** to be aware of constraints
- Widely accepted as unsuitable for at-scale deployments



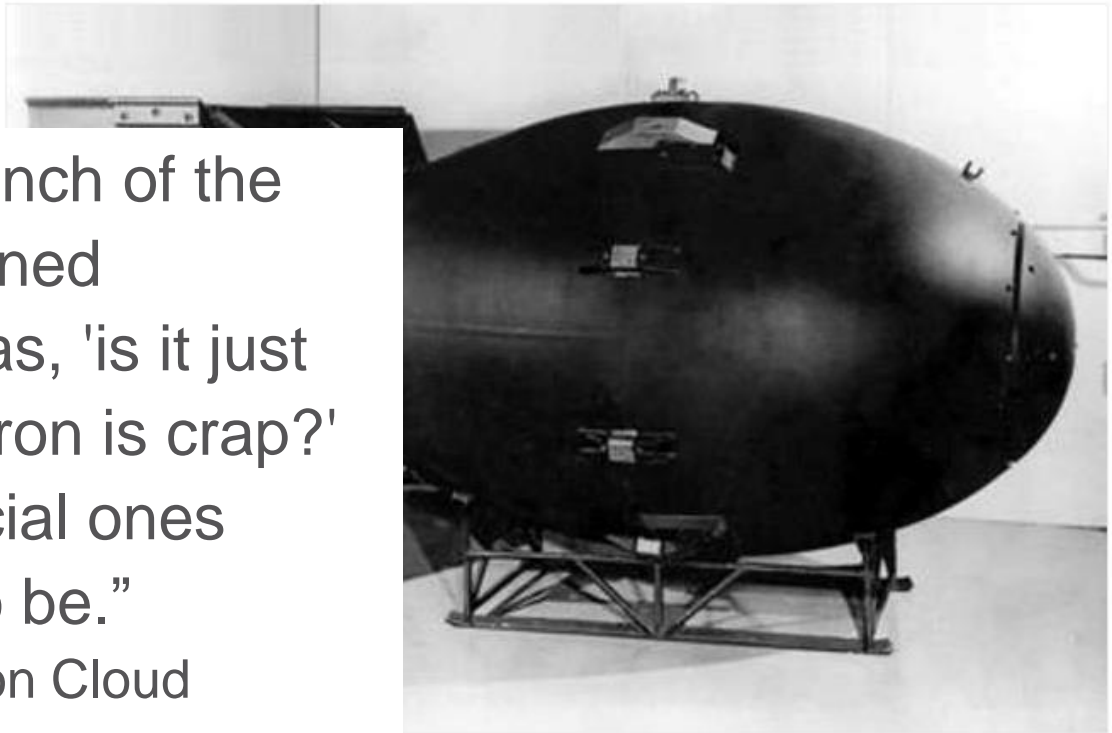
Overlay networks

- Connect each container to a virtual Layer 2 segment
- Separate “overlay” domain over “underlay” network with GRE, MPLS, VXLAN, or proprietary tunneling protocols
- But...
 - Lots of state – 1,000 machines => full mesh of 499,500 tunnels!
 - Breaking out of virtual network sandboxes requires NAT / router
 - L2 typically not required today
 - Requires **app developers** to be networking experts



HP: OpenStack's networking nightmare Neutron 'was everyone's fault'

Cloud brains at HP, Red Hat, Piston, spill beans on the weak link in the OpenStack cloud

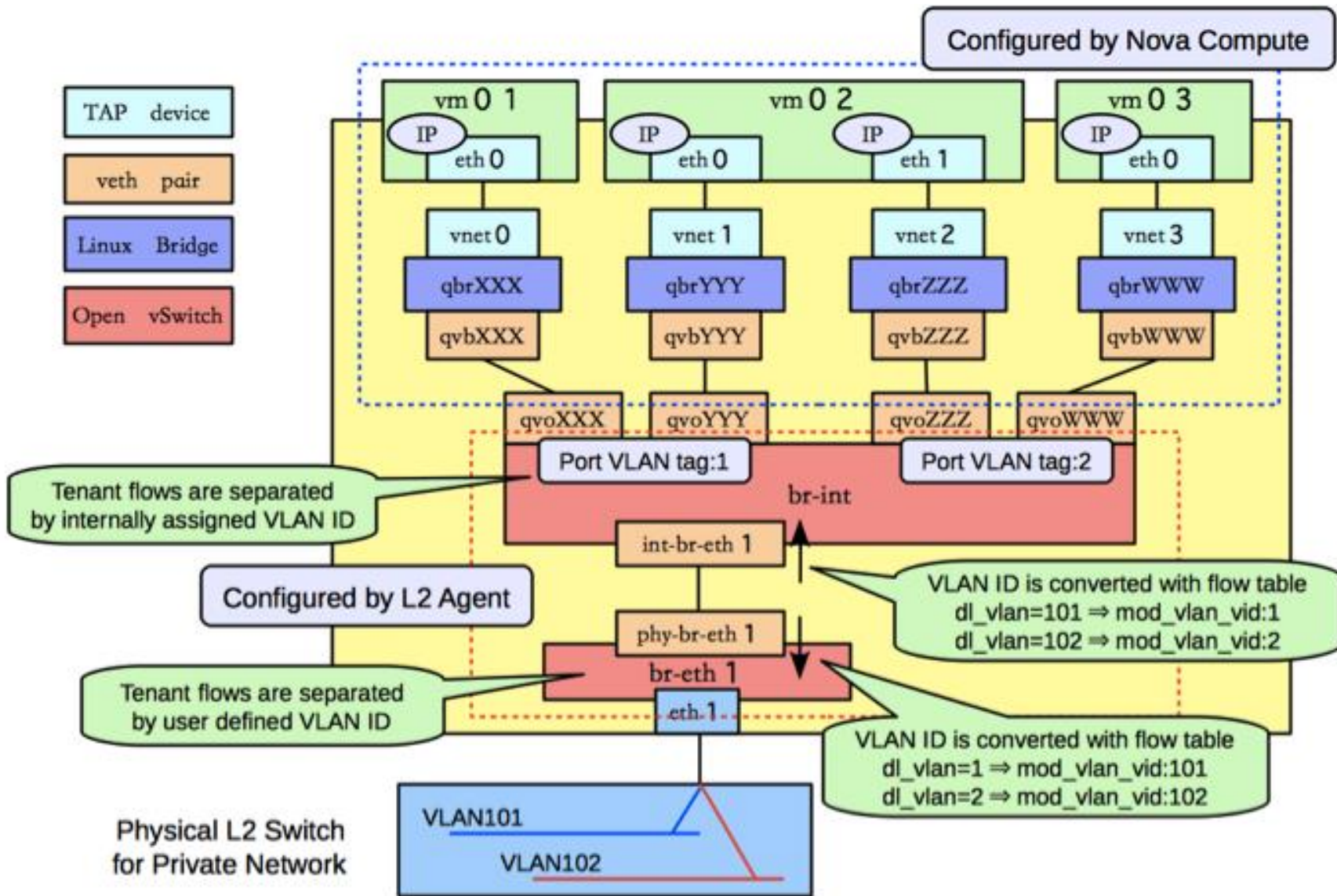


“We tried supporting a bunch of the commercial software-defined networks... The theory was, 'is it just the open vSwitch in Neutron is crap?' – [but] even the commercial ones aren't where they need to be.”

– Joshua McKenty, CTO, Piston Cloud

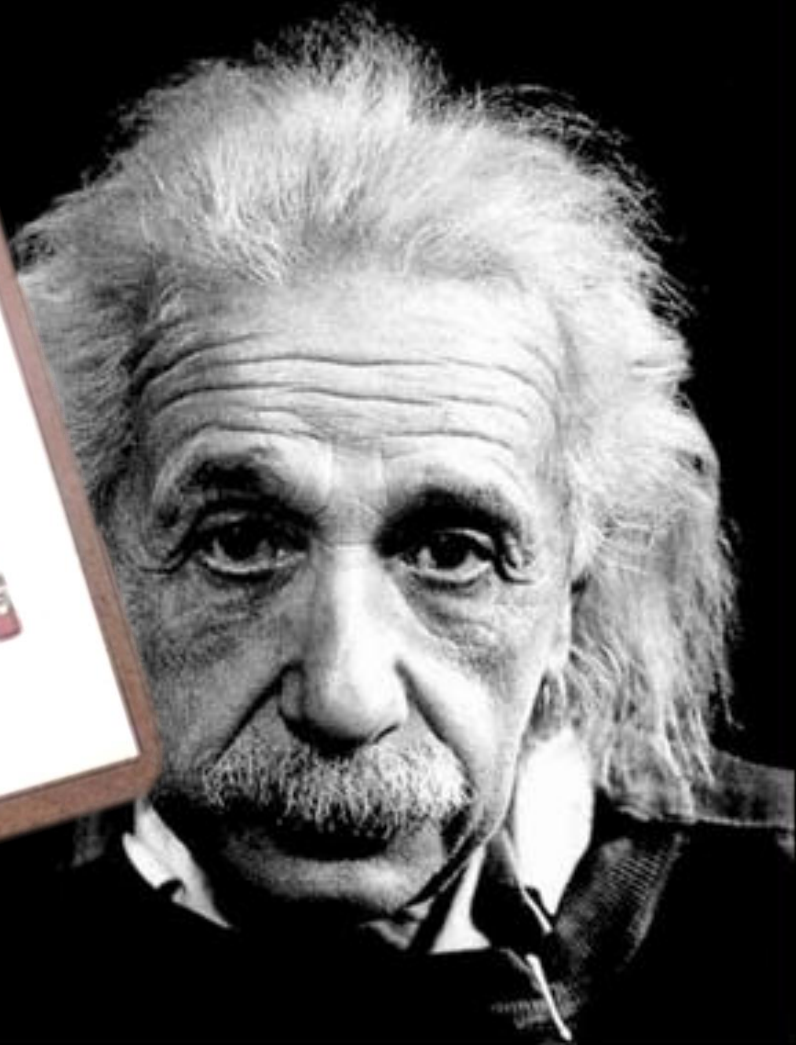


A Glimpse at Overlay Networking Complexity



Virtual Networking Requirements

- Workloads need to communicate with one another
- Enforce policy (who can talk to whom)
- Base requirement for IP connectivity

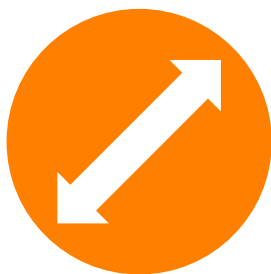




PROJECT
CALICO

What is Calico?

An open source project to enable scalable, simple and secure IP networking in a data center / cloud environment



Scalable

Thousands of servers,
100k's of workloads



Simple

Don't demand users to
be networking experts

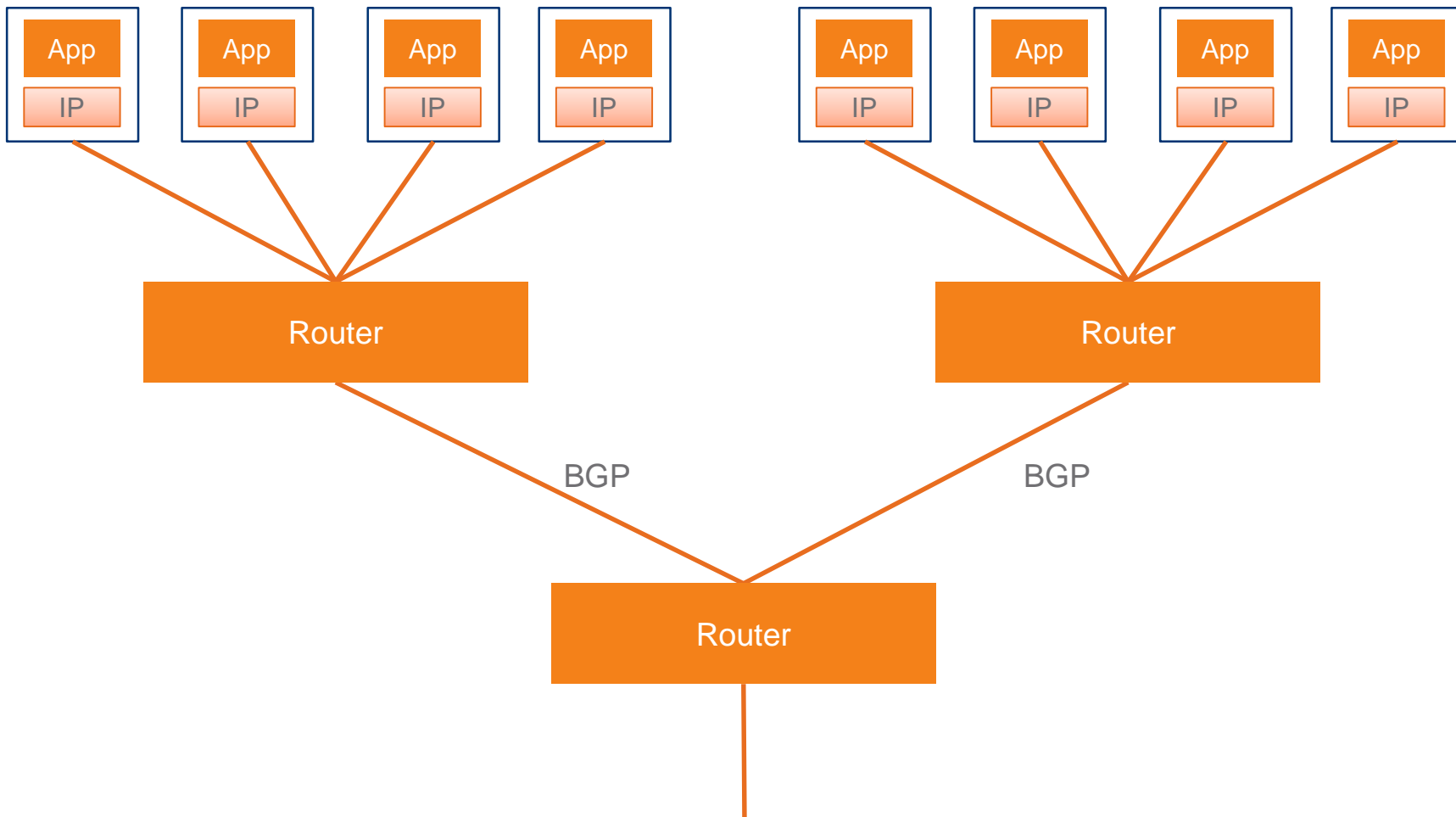


Secure

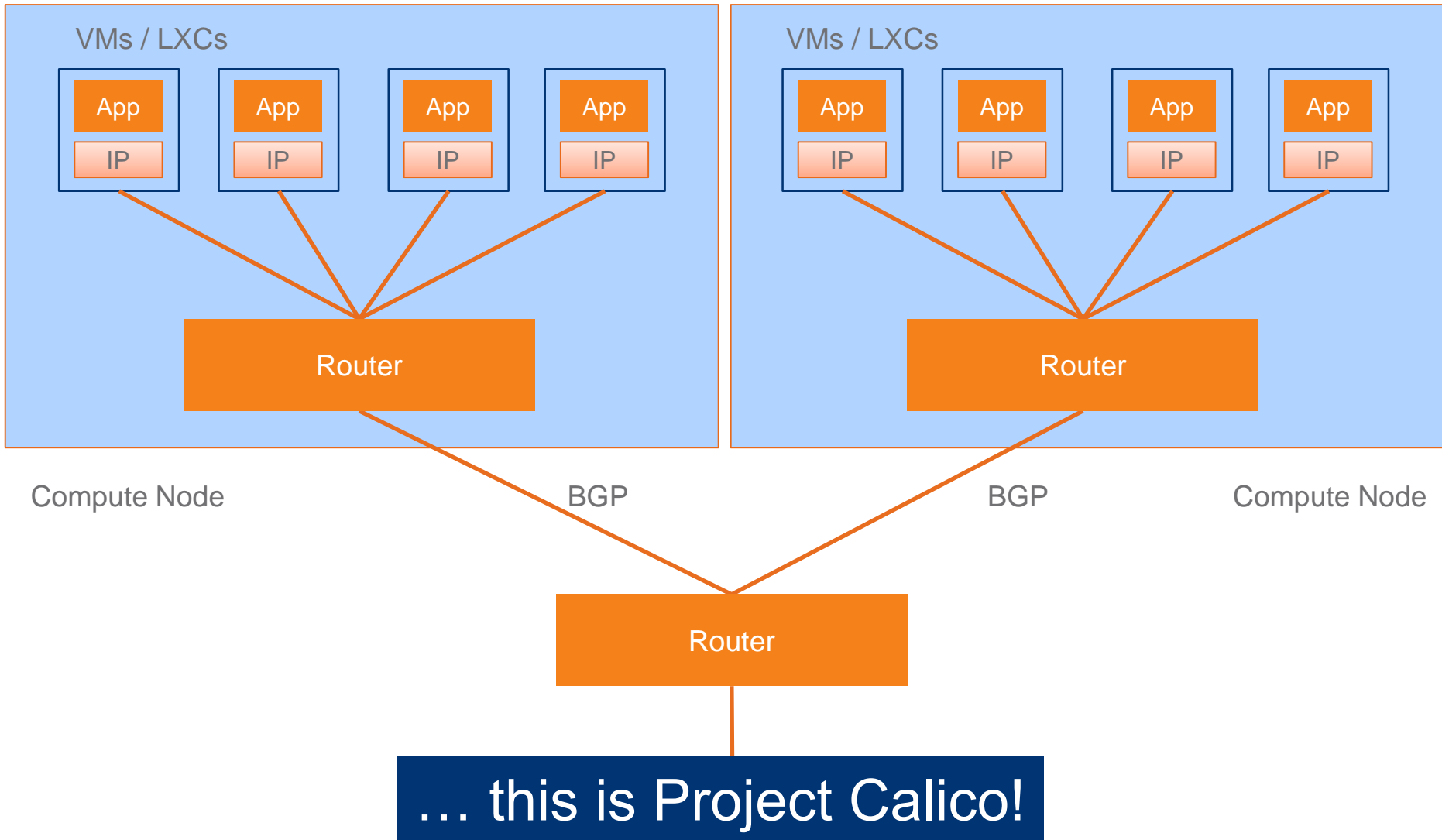
Rich micro-service
policy framework

What if we built a data center like the internet?

Hosts



What if we built a data center like the internet?





Traditional approach:
VLANs or mesh of tunnels
(separation as by-product)

Calico:

Just a set of simple policy rules – all workloads in a given “network” just share a tag; all with the same tag are whitelisted, others blacklisted



Traditional approach:

Discrete firewall elements that are a bottleneck and not optimally placed to enforce security (by the time the packet reaches the firewall, it can't be sure where it came from)

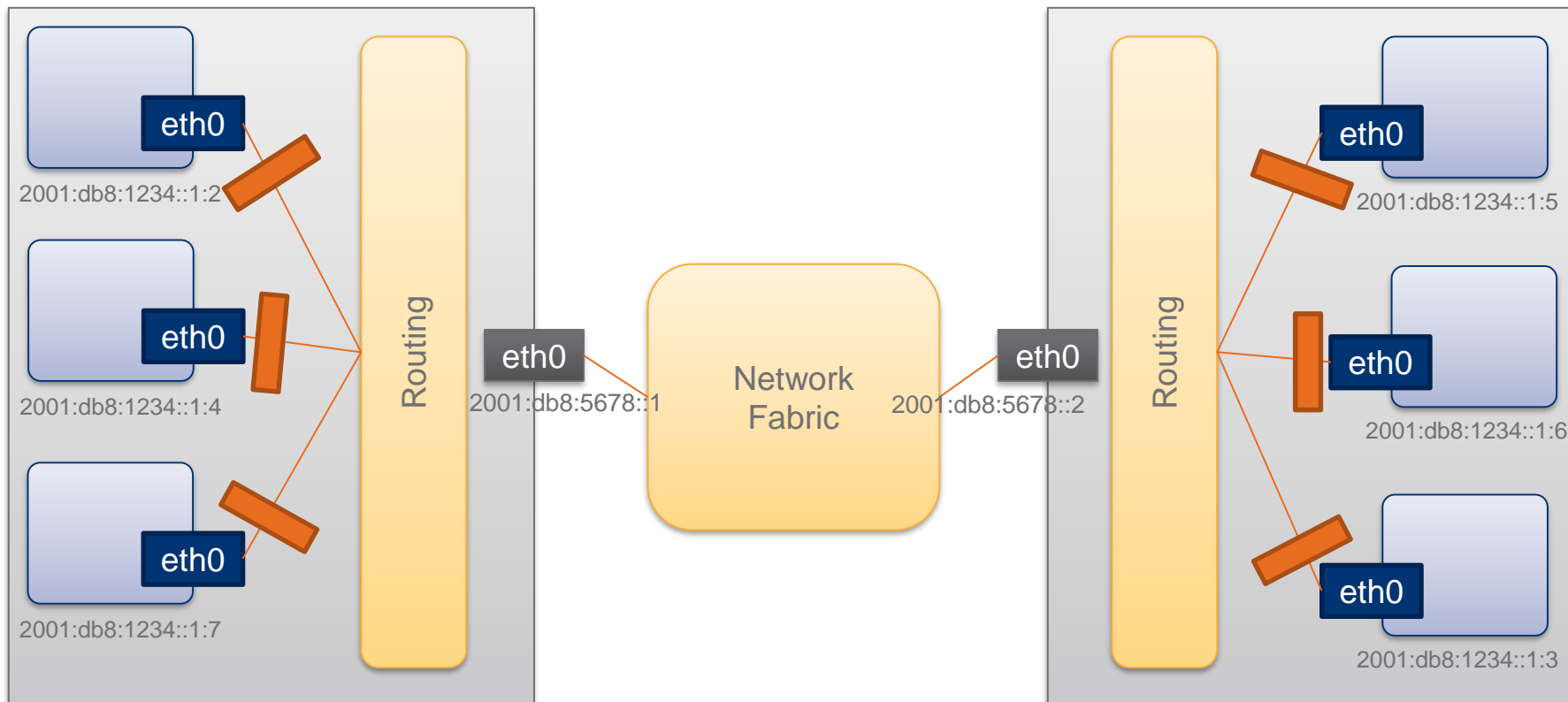


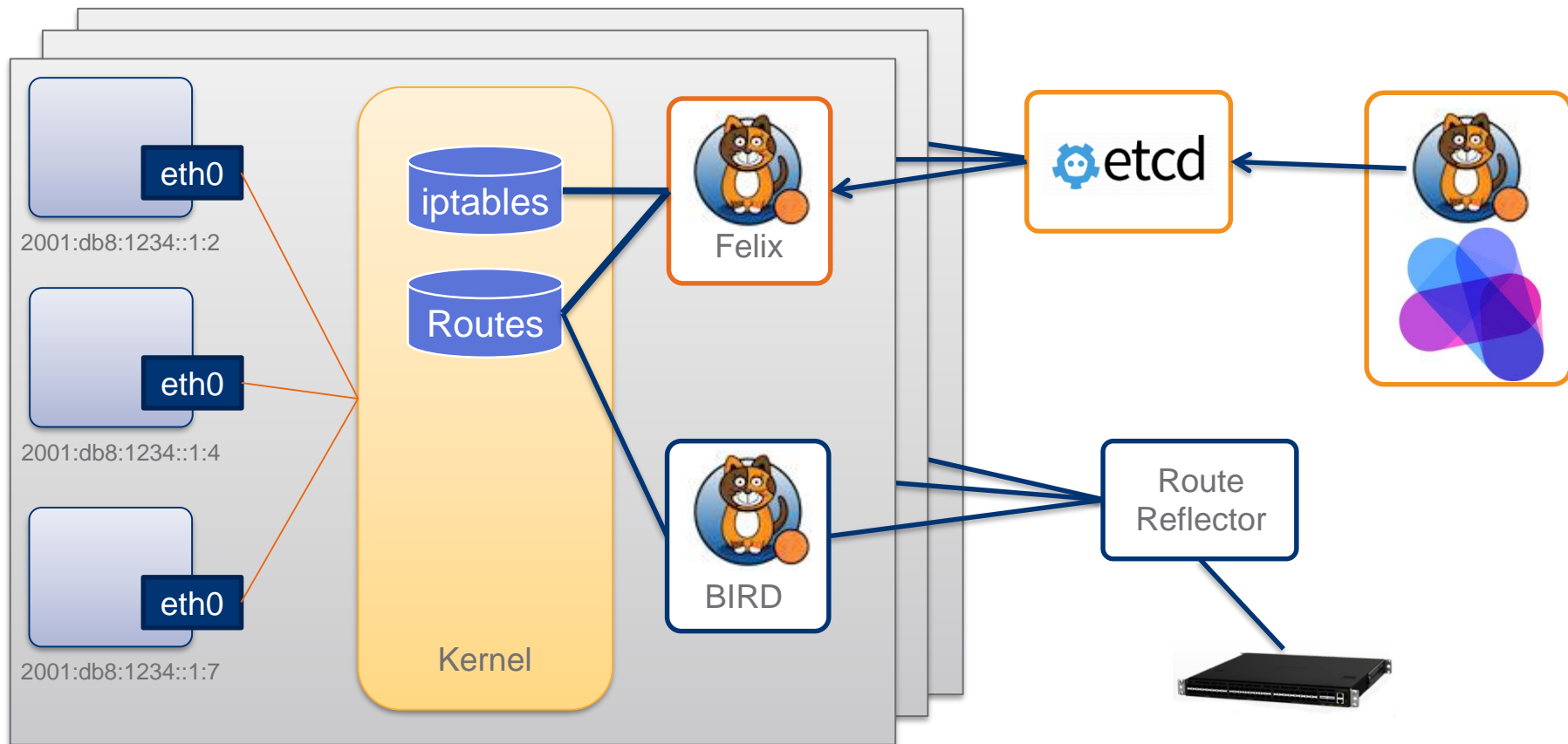
Calico:

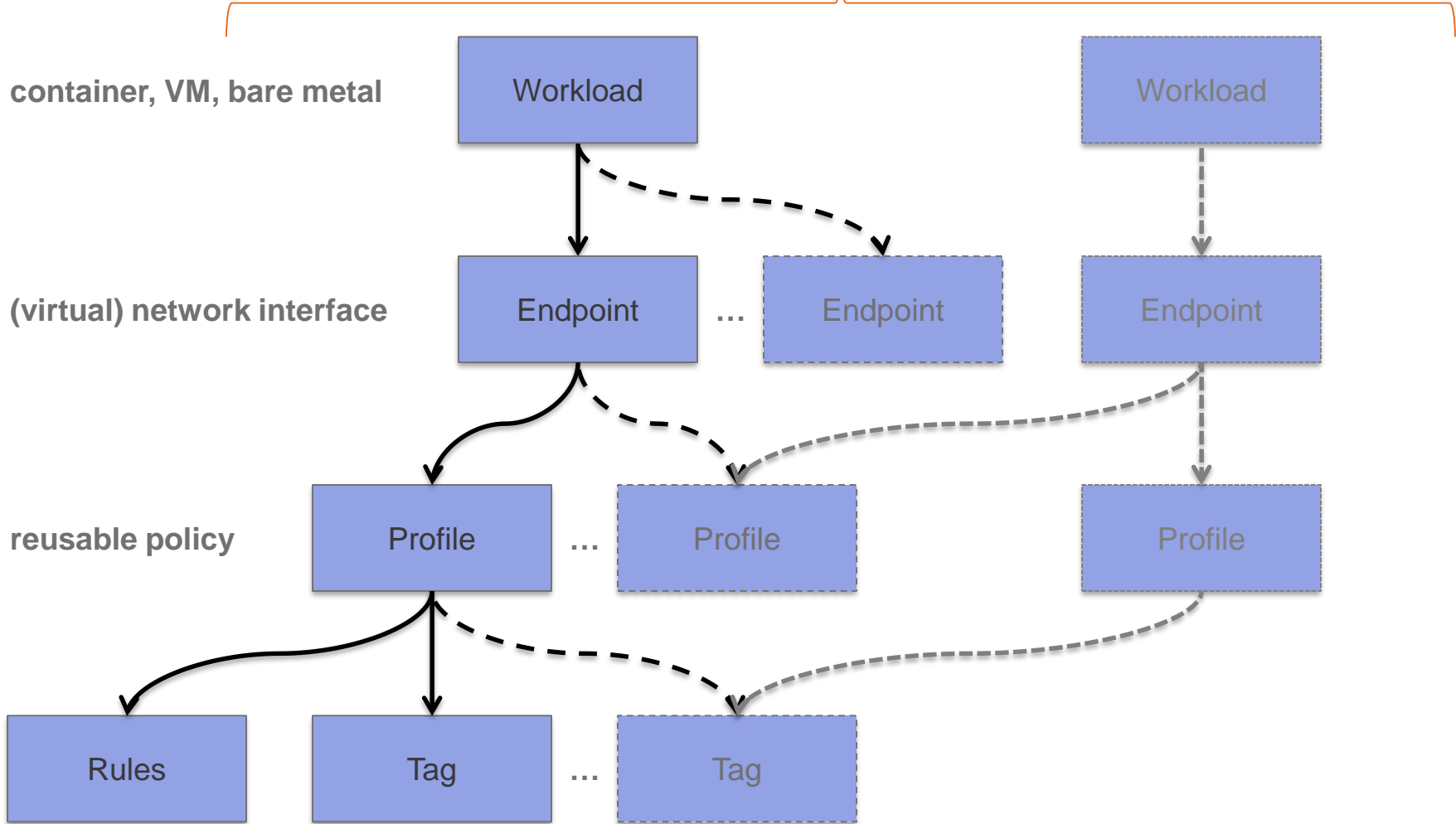
- Rich, but simply defined, policy rules
- Distributed host-based ACL calculation
- Ingress/egress enforcement in the data-plane pipeline immediately adjacent to each workload



The Distributed Firewall









Before Calico

Scale challenges above few hundred servers / thousands of workloads

Troubleshooting connectivity issues can take hours

On/off ramps + NAT to break out of overlay

High availability / load balancing across links requires LB function (virtual or physical) and/or app-specific logic

CCNA or equivalent required to understand end-to-end networking, deploy applications



After Calico

Scale to millions of workloads with minimal CPU and network overhead

What is happening is “obvious” – traceroute, ping, etc., work as expected

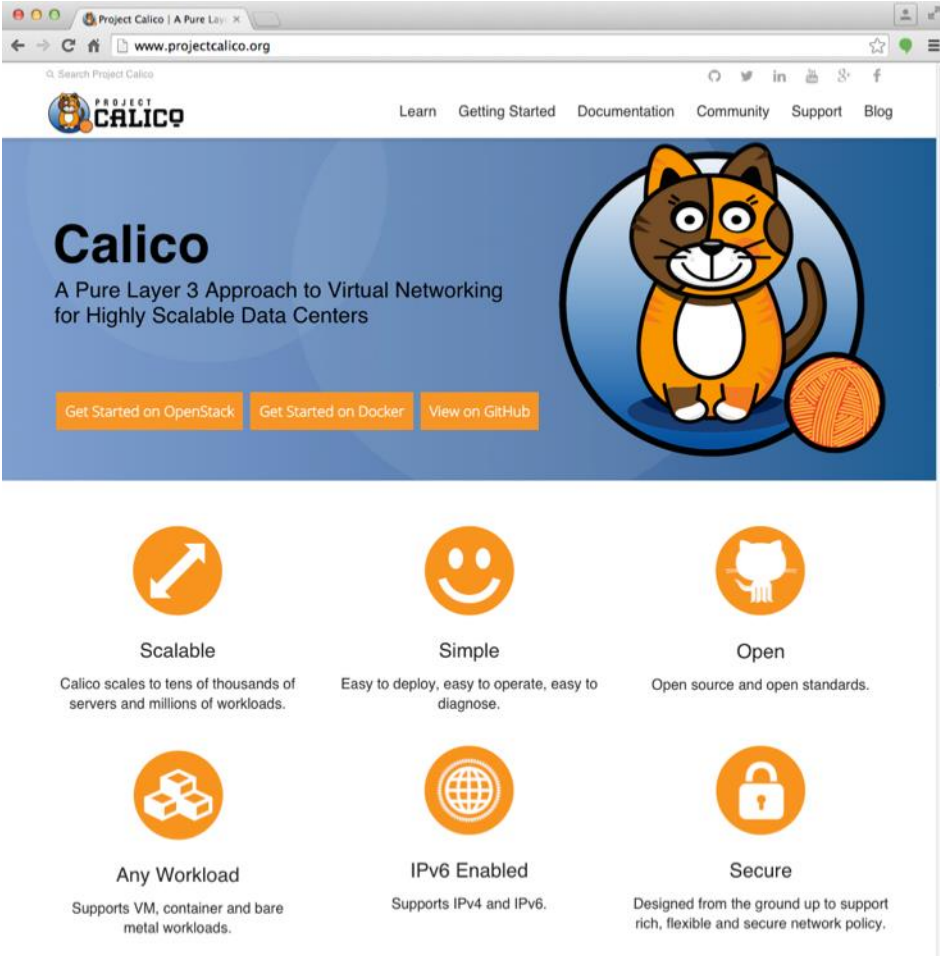
Path from workload to non-virtual device or public internet (or even between data centers) is *just* a route

Equal Cost Multi-Path (ECMP) & Anycast just work, enabling scalable resilience and full utilization of physical links

Basic IP networking knowledge only required



- Main project website: www.projectcalico.org
- Github
 - github.com/projectcalico
- Mailing list, Slack info:
 - projectcalico.org/contact/
- freenode IRC: [#calico](https://freenode.net/#calico)
- Download & try it out
- We welcome your feedback and contributions
- Follow us  [@projectcalico](https://twitter.com/projectcalico)



The screenshot shows the Project Calico website homepage. The browser address bar displays "www.projectcalico.org". The page features a navigation menu with links for "Learn", "Getting Started", "Documentation", "Community", "Support", and "Blog". The main content area has a blue background with the Calico logo (a cartoon cat) and the text "Calico: A Pure Layer 3 Approach to Virtual Networking for Highly Scalable Data Centers". Below this are three buttons: "Get Started on OpenStack", "Get Started on Docker", and "View on GitHub". The page is organized into six feature cards, each with an icon and a brief description:

- Scalable**: Calico scales to tens of thousands of servers and millions of workloads. (Icon: arrow pointing up and right)
- Simple**: Easy to deploy, easy to operate, easy to diagnose. (Icon: smiley face)
- Open**: Open source and open standards. (Icon: GitHub logo)
- Any Workload**: Supports VM, container and bare metal workloads. (Icon: three stacked cubes)
- IPv6 Enabled**: Supports IPv4 and IPv6. (Icon: globe)
- Secure**: Designed from the ground up to support rich, flexible and secure network policy. (Icon: padlock)