PROJECT CALICO

Simple, Secure, Scalable networking for the virtualized datacentre

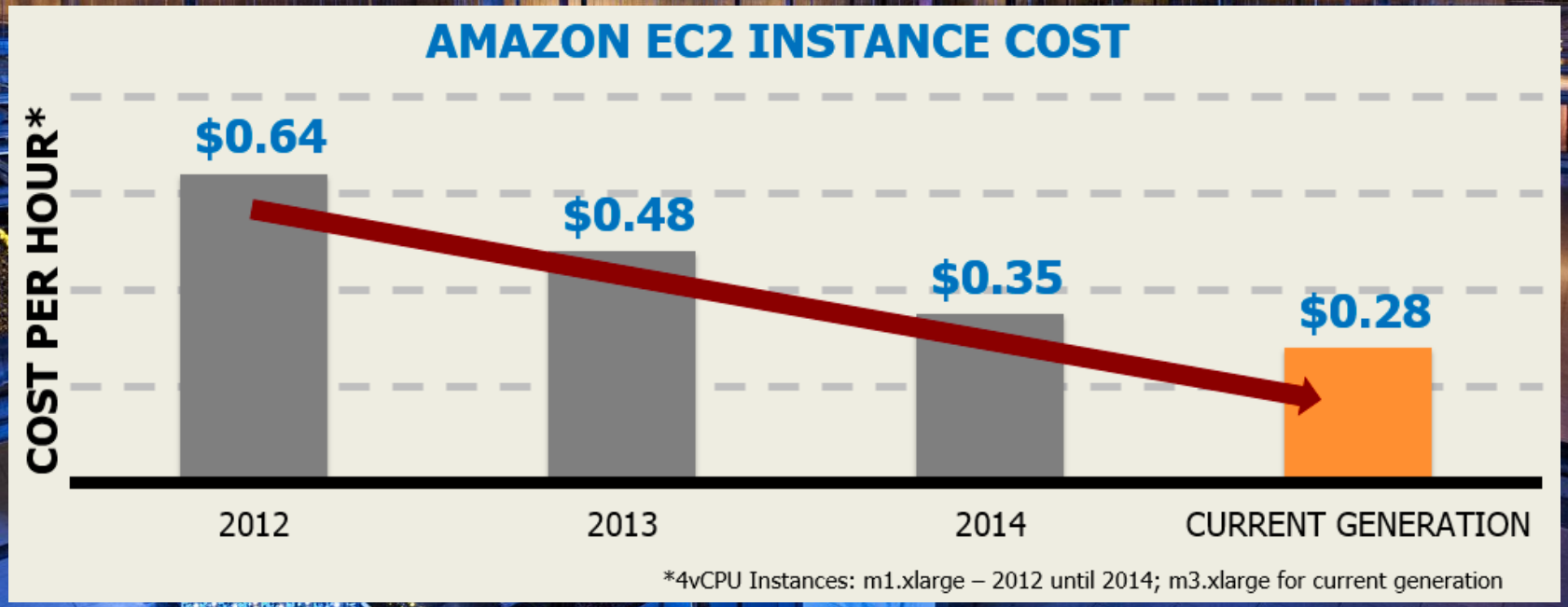
UKNOF 33

Ed Harrison
@eepyaich

19th January 2016

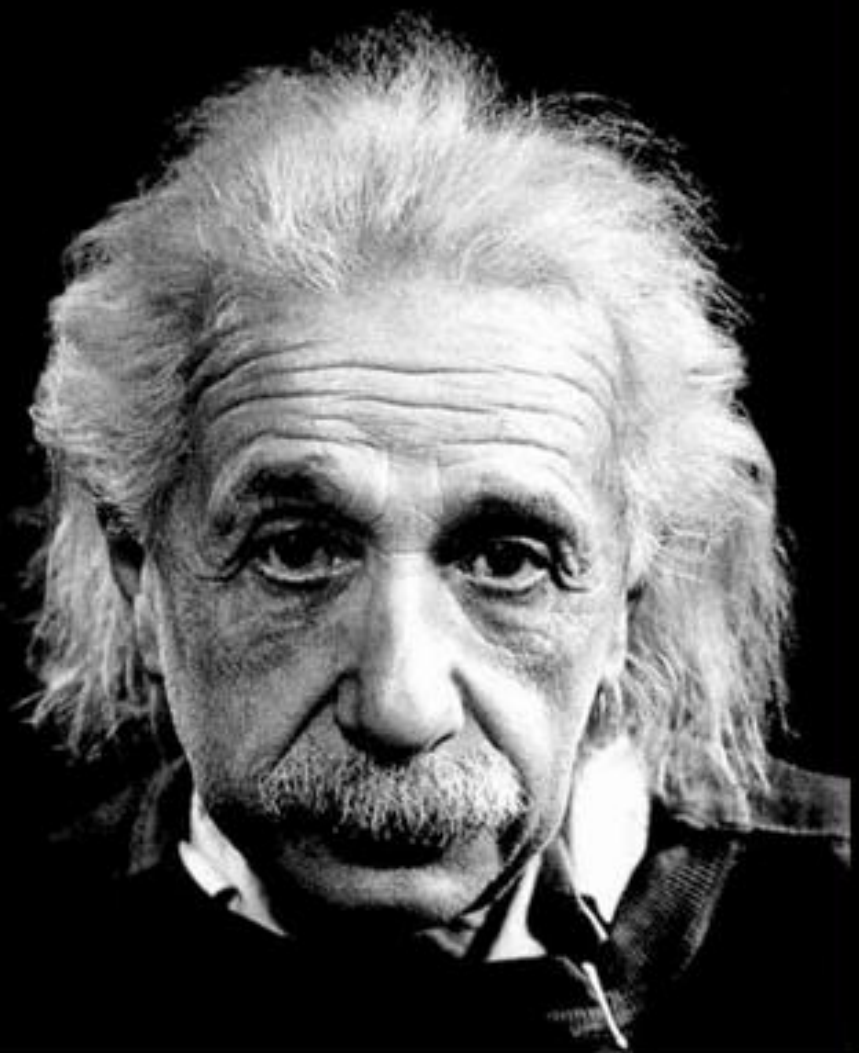
The Goal:

Hyperscale Efficiency

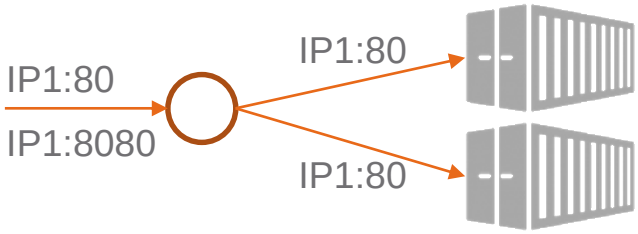


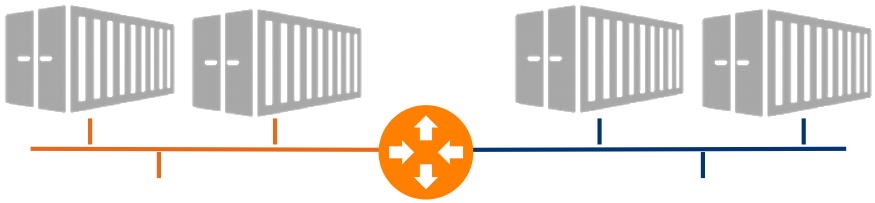
“Everything should be made as simple as possible, but not simpler.”

Albert Einstein



Virtual Networking Today (VM's / Containers)





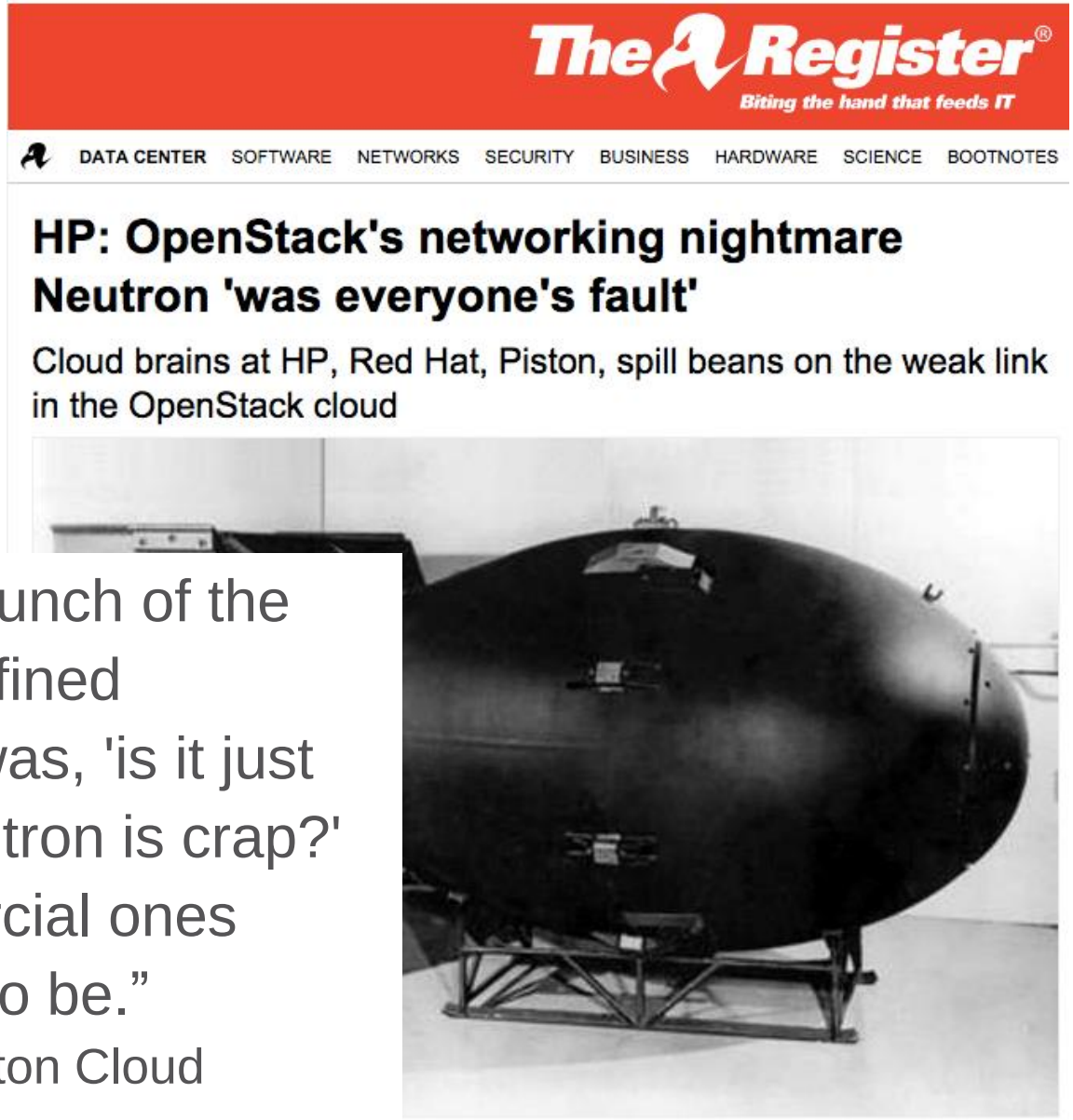
Port forwarding / NAT

- 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

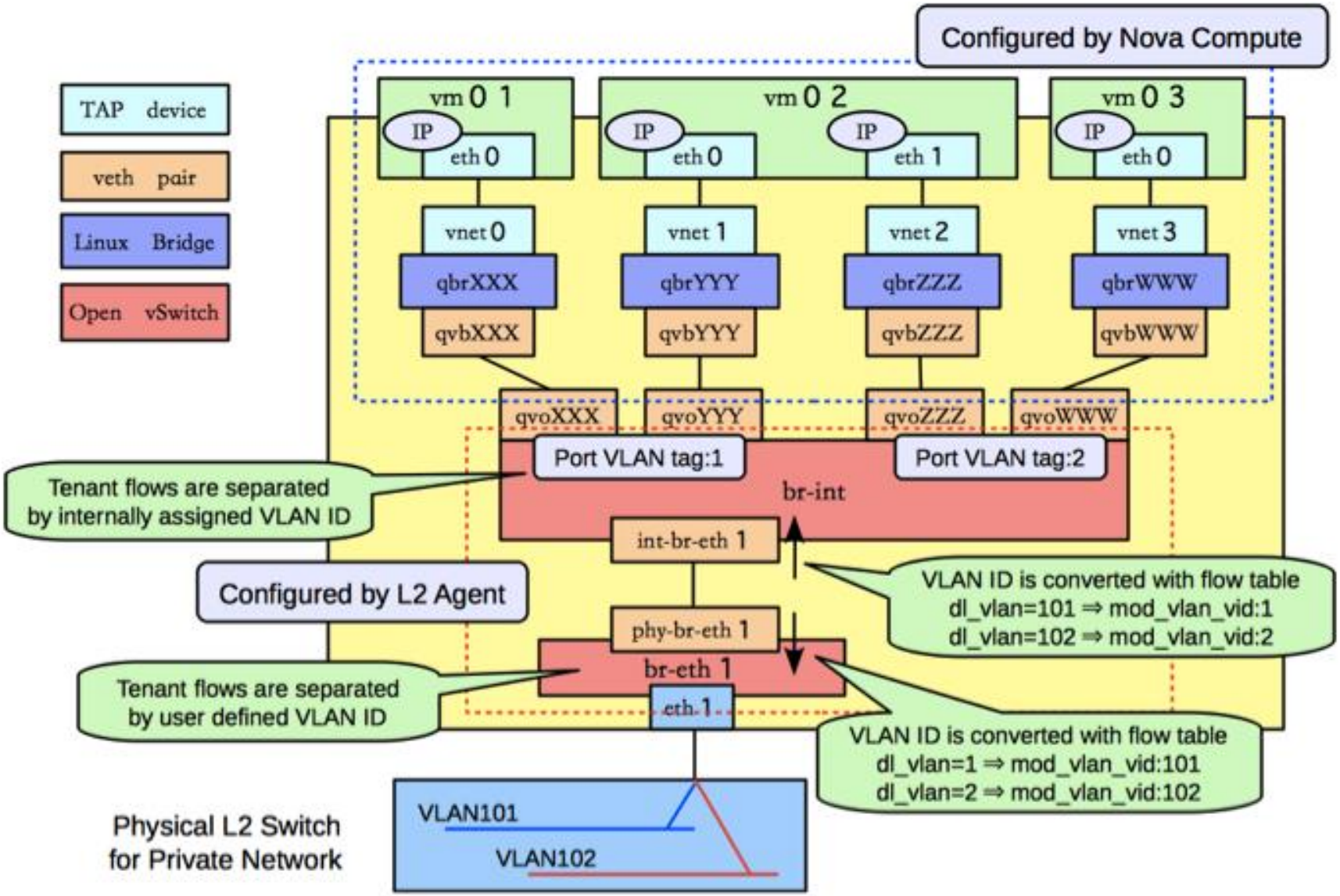
They shoulda listened to Einstein…

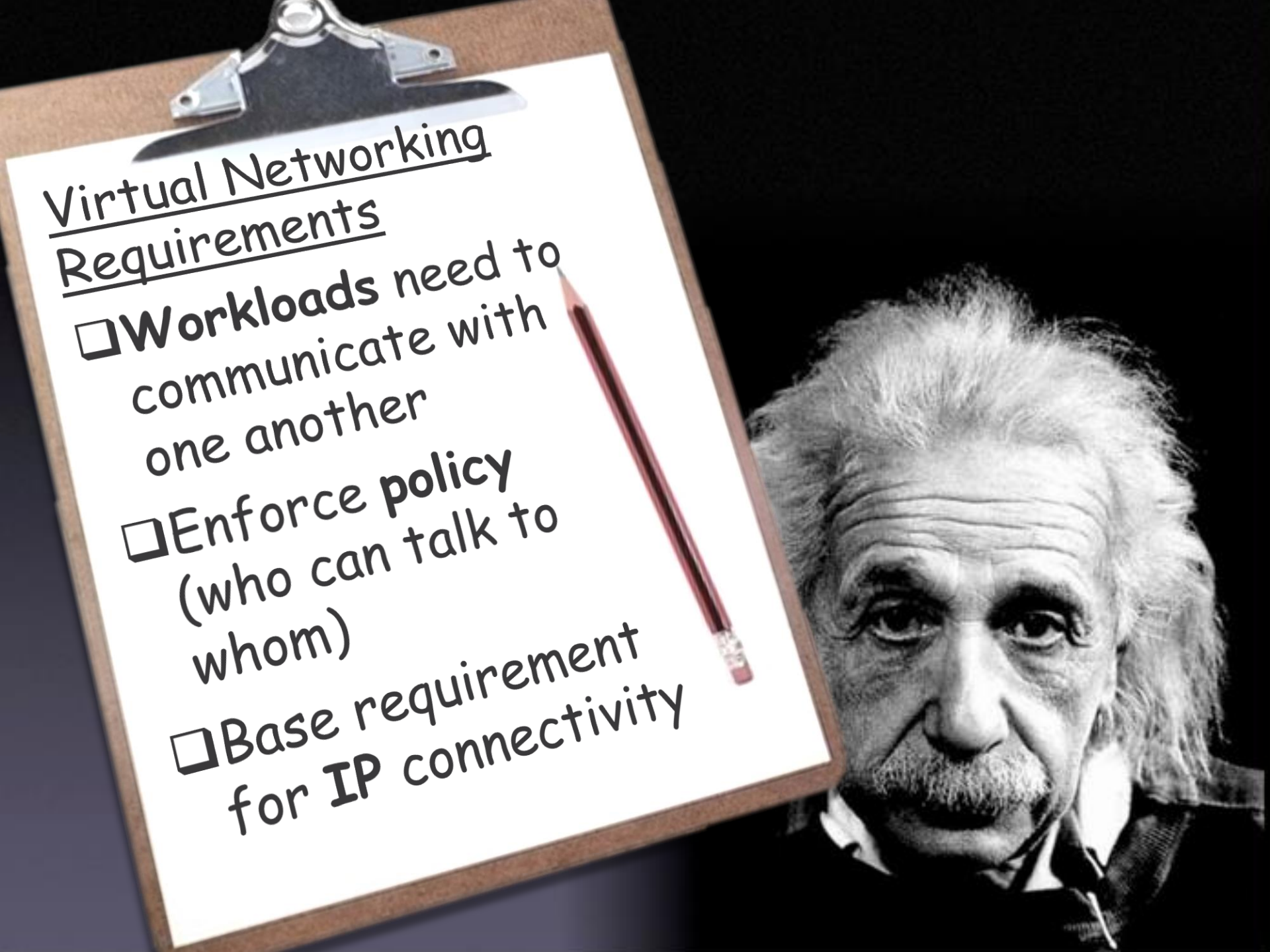


"We tried supporting a bunch of the commercial software-defined networks... The theory was, 'is it just the open vSwitch in Neutron is crap?' – [but] even the commercial ones aren't where they need to be."

– 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

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

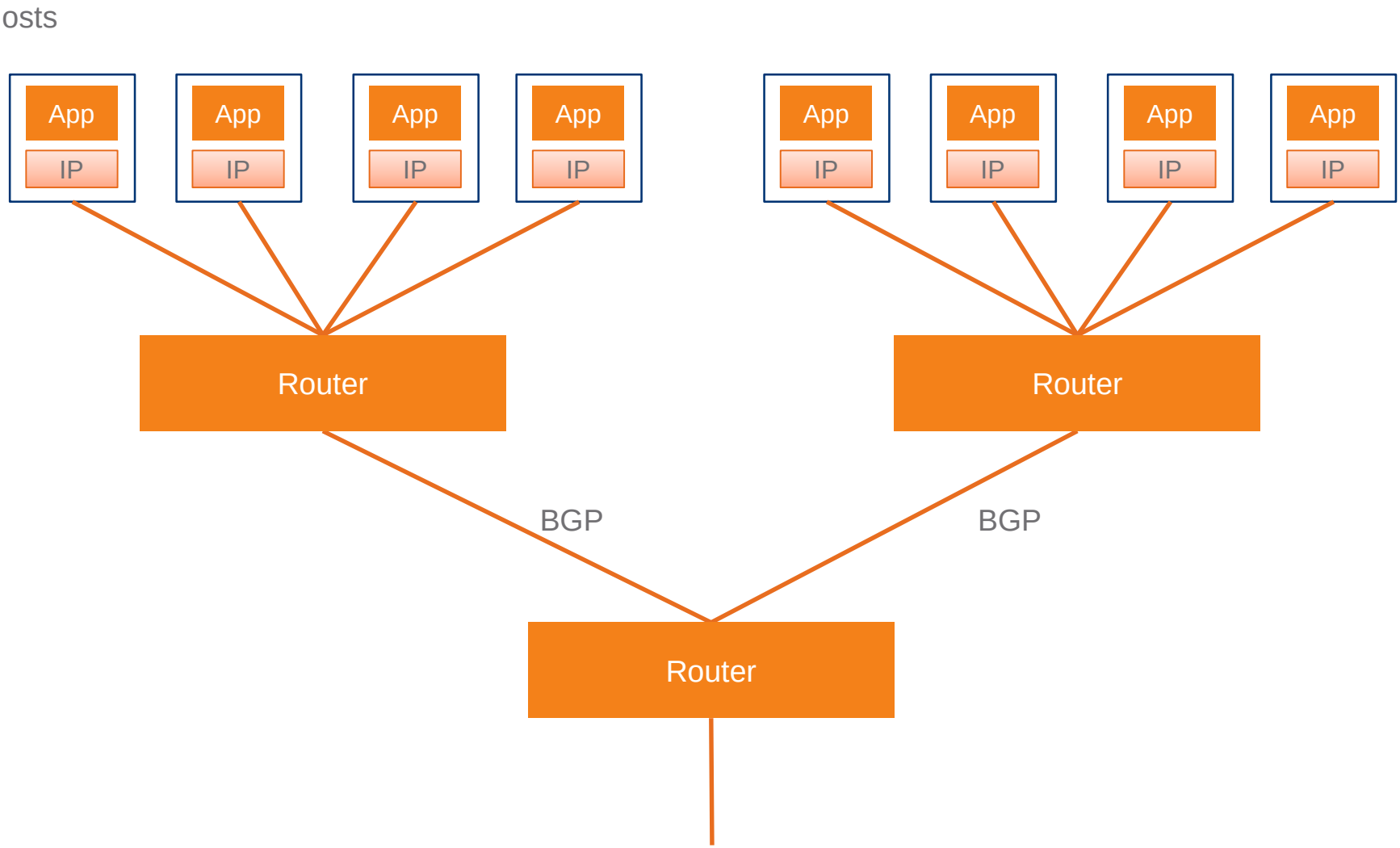
Simple

Don't demand users to be networking experts

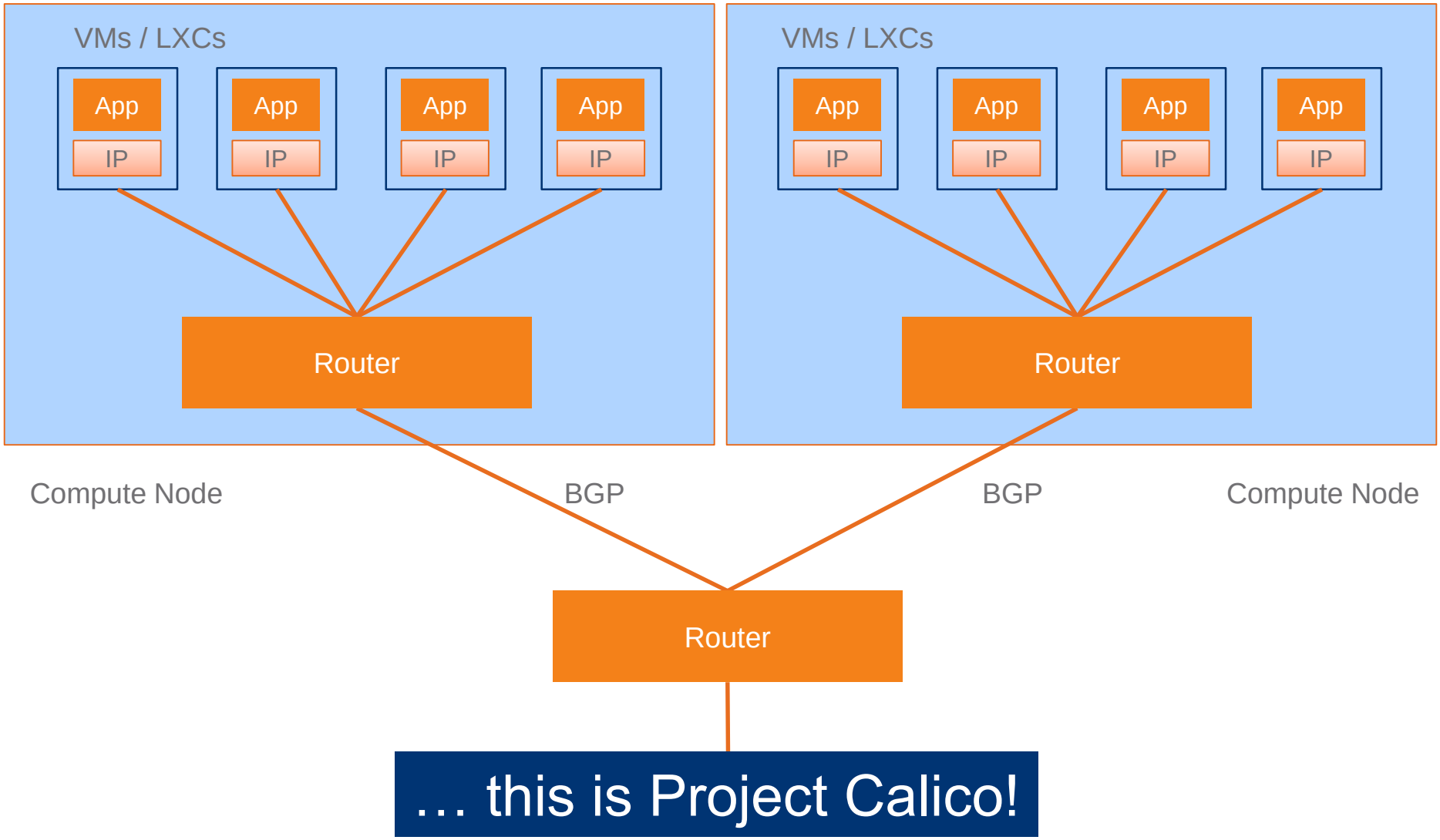
Secure

Rich micro-service policy framework

What if we built a data center like the internet?



What if we built a data center like the internet?



Coarse-grained Policy (network/tenant isolation)

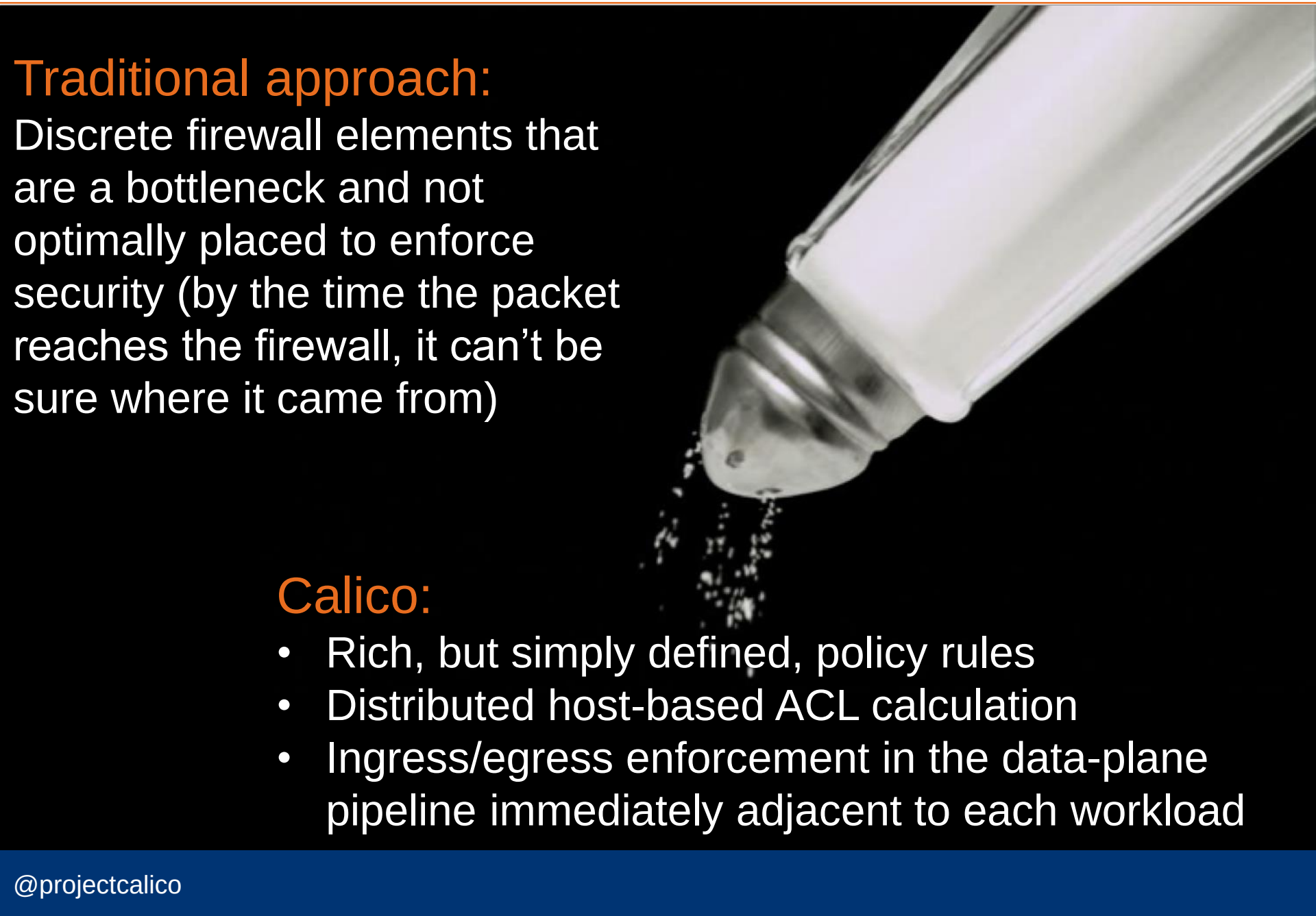
Traditional approach:

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

Fine-grained Policy (micro-service segmentation)



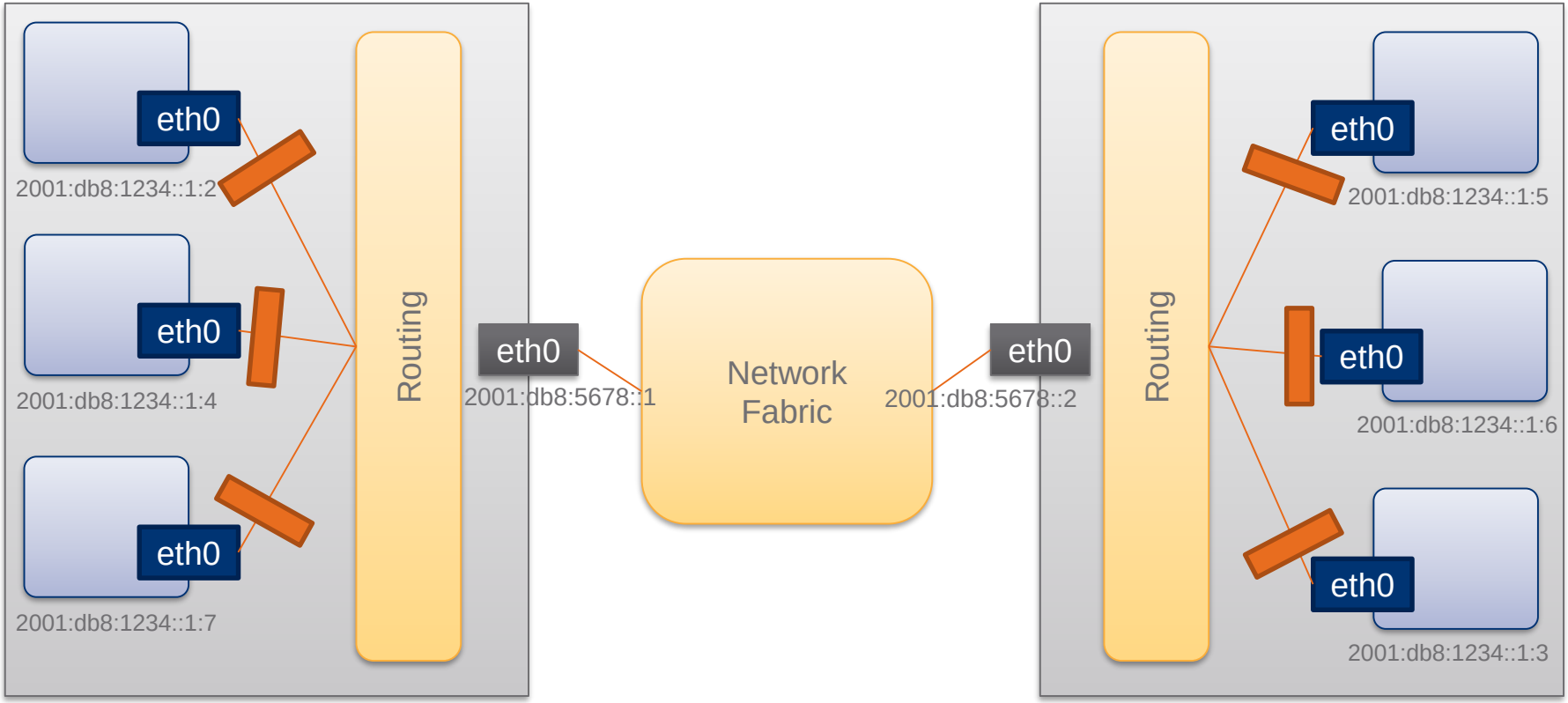
Traditional approach:

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)

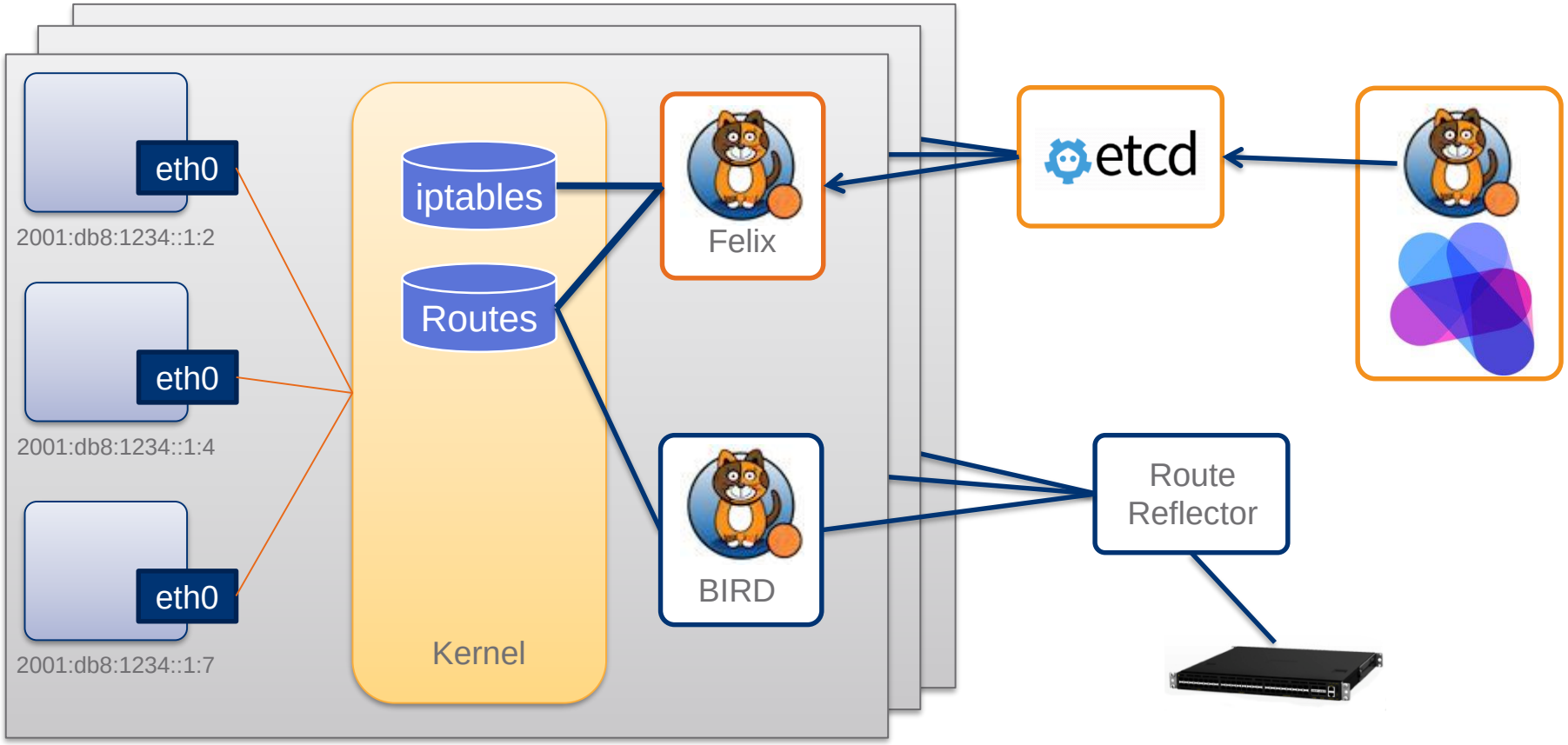
Calico:

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

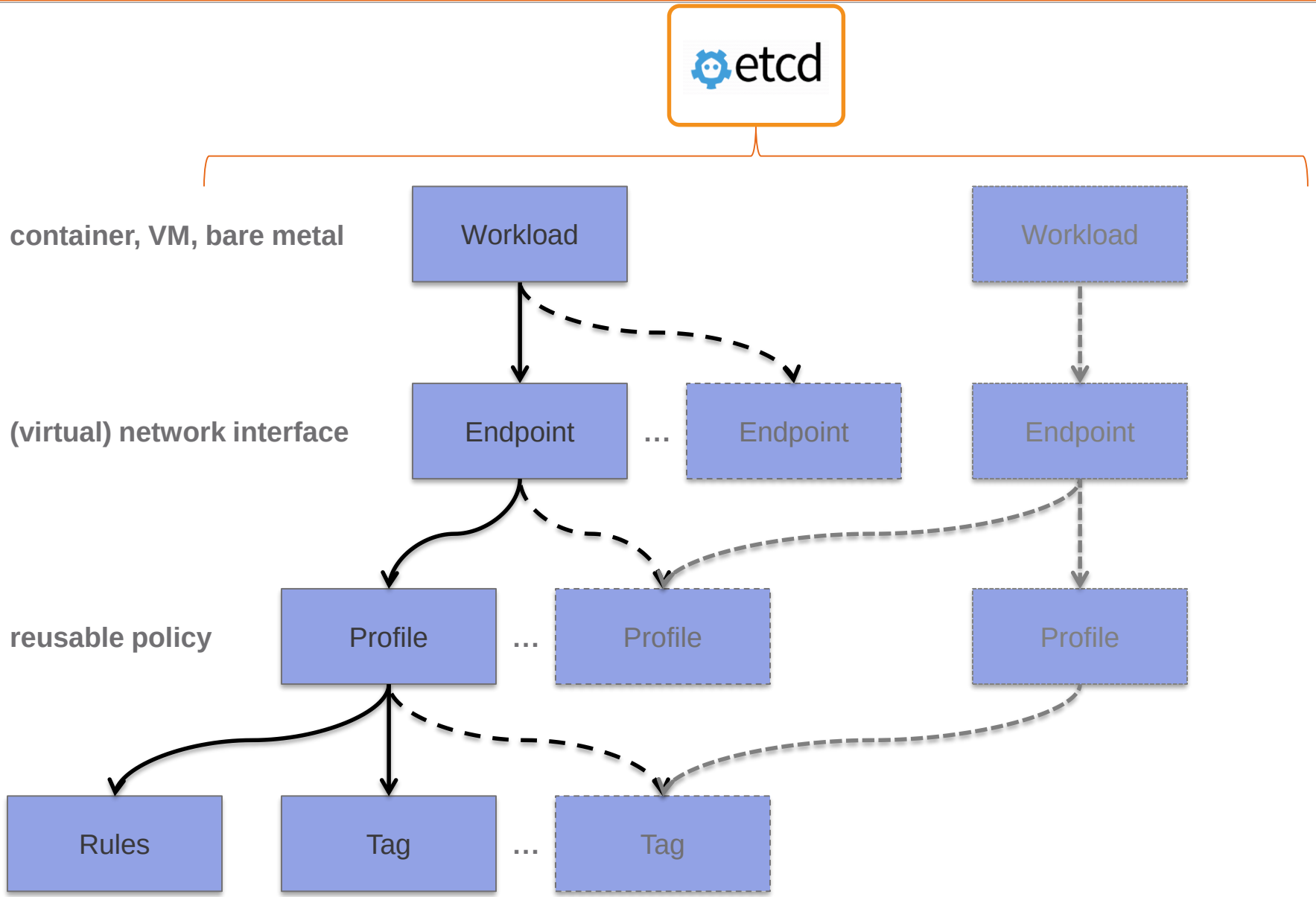
The Distributed Firewall



Project Calico architecture



Calico Network Policy Model



Life Before and after Calico

Before Calico		After Calico
Scale challenges above few hundred servers / thousands of workloads		Scale to millions of workloads with minimal CPU and network overhead
Troubleshooting connectivity issues can take hours		What is happening is “obvious” – traceroute, ping, etc., work as expected
On/off ramps + NAT to break out of overlay		Path from workload to non-virtual device or public internet (or even between data centers) is *just* a route
High availability / load balancing across links requires LB function (virtual or physical) and/or app-specific logic		Equal Cost Multi-Path (ECMP) & Anycast just work, enabling scalable resilience and full utilization of physical links
CCNA or equivalent required to understand end-to-end networking, deploy applications		Basic IP networking knowledge only required

Get Involved

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