SDN for the Cloud

Albert Greenberg
Distinguished Engineer
Director of Networking @ Microsoft Azure
albert@microsoft.com

Microsoft

Road to SDN

- WAN
  - Motivating scenario: network engineering at scale
  - Innovation: infer traffic, control routing, centralize control to meet network-wide goals

TomoGravity
ACM Sigmetrics 2013 Test of Time Award

RCP
Usenix NSDI 2015 Test of Time Award

4D
ACM Sigcomm 2015 Test of Time Award

- Cloud
  - Motivating scenario: software defined data center, requiring per customer virtual networks
  - Innovation: **VL2**
    - scale-out L3 fabric
    - network virtualization at scale, enabling SDN and NFV

VL2
ACM Sigcomm 2009

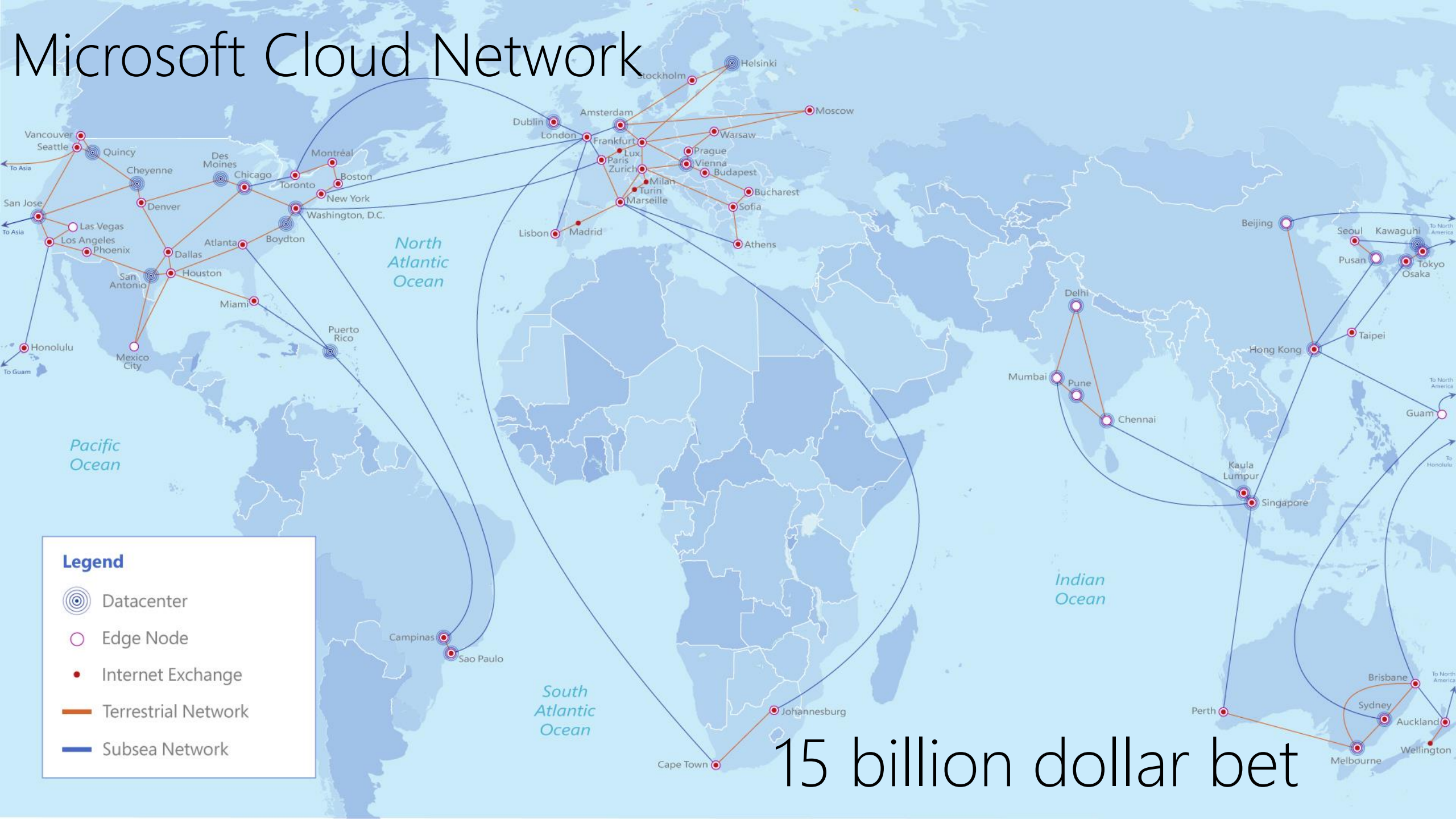
Cloud provided the killer scenario for SDN

- Cloud has the right scenarios
  - Economic and scale pressure → huge leverage
  - Control → huge degree of control to make changes in the right places
  - Virtualized Data Center for each customer → prior art fell short
- Cloud had the right developers and the right systems
  - High scale fault tolerant distributed systems and data management

At Azure we changed everything because we had to, from optics to host to NIC to physical fabric to WAN to Edge/CDN to ExpressRoute (last mile)

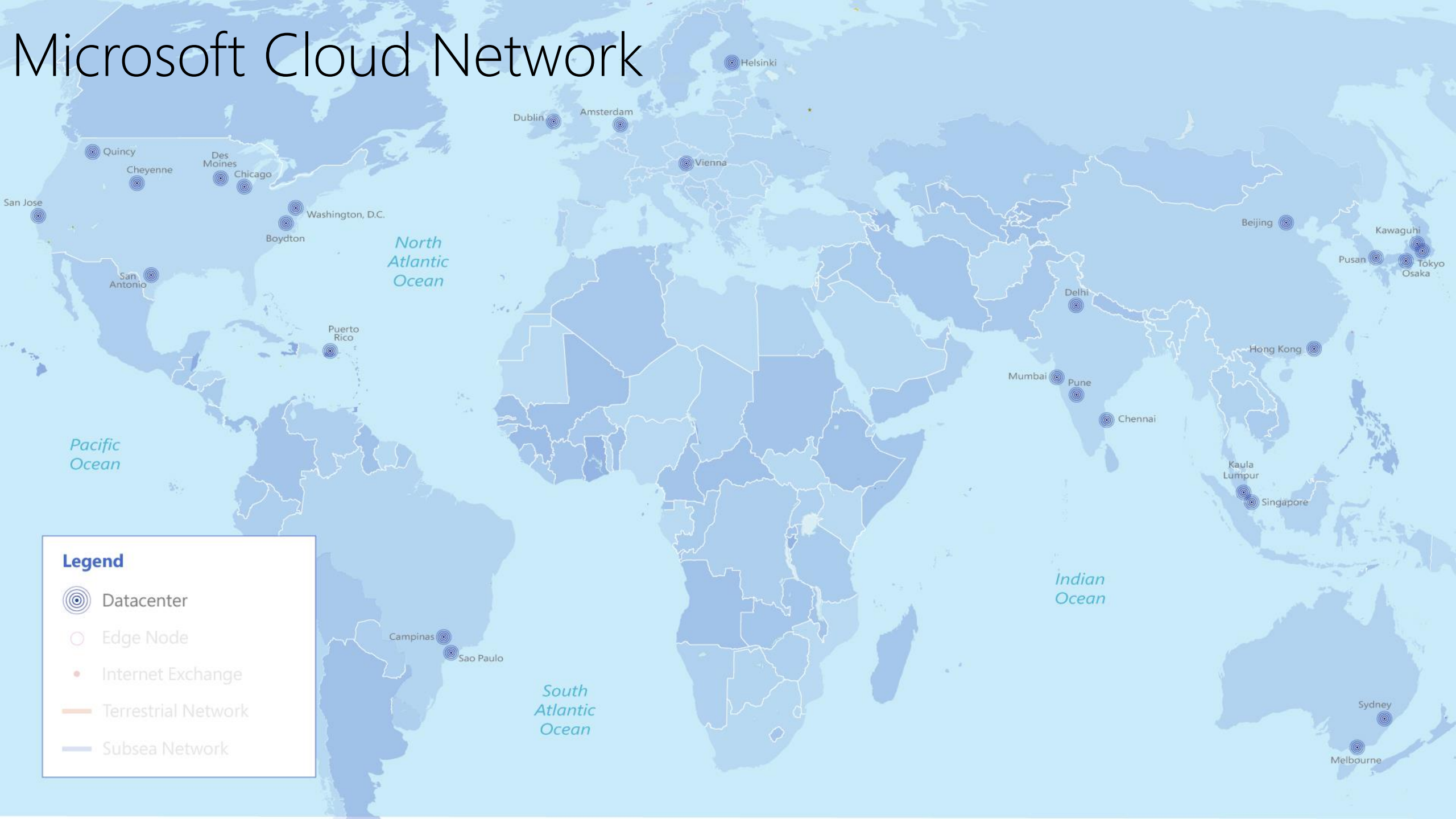
Hyperscale Cloud

Microsoft Cloud Network

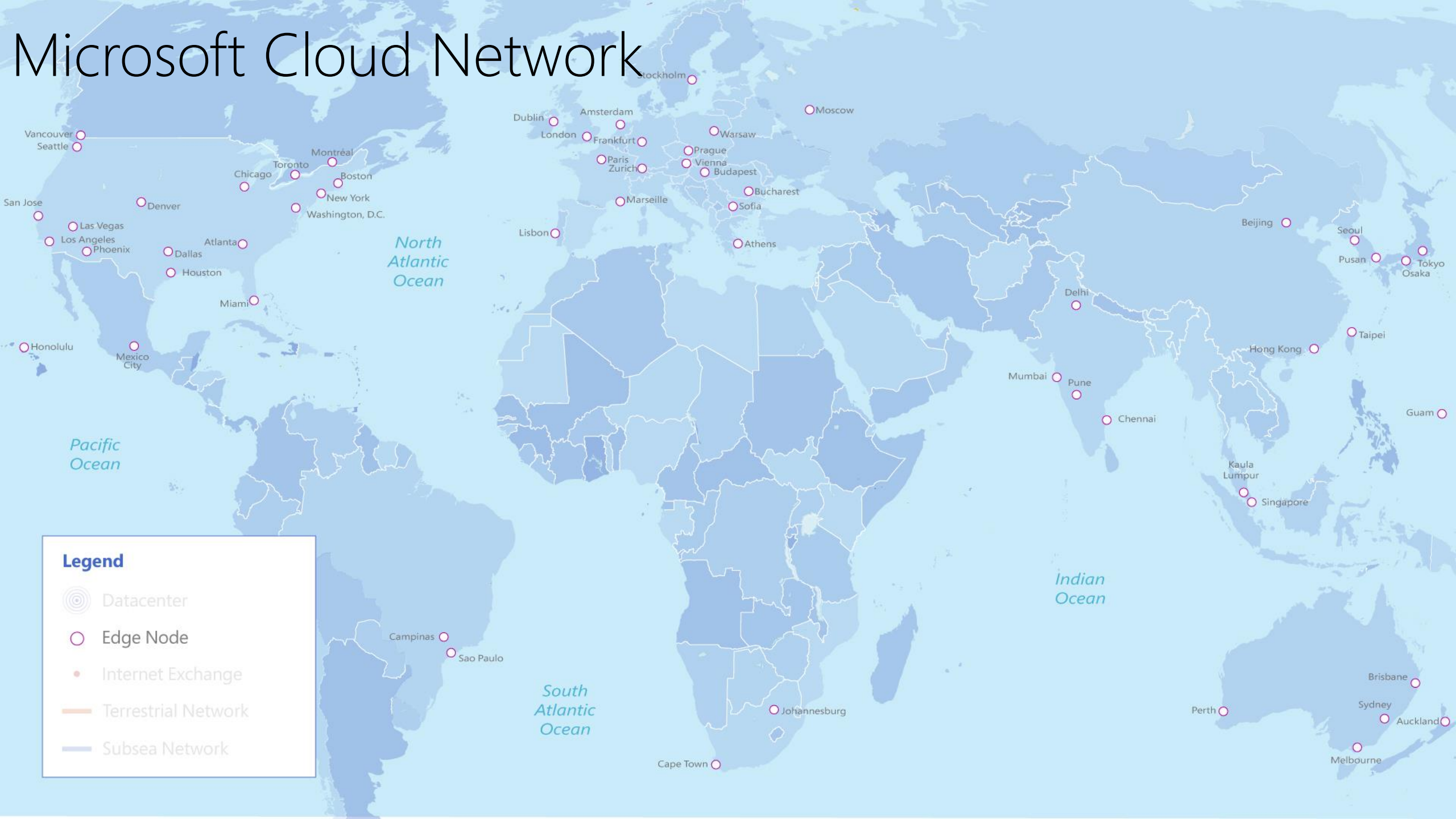


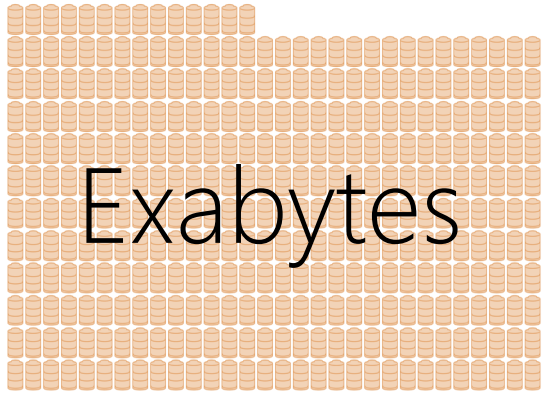
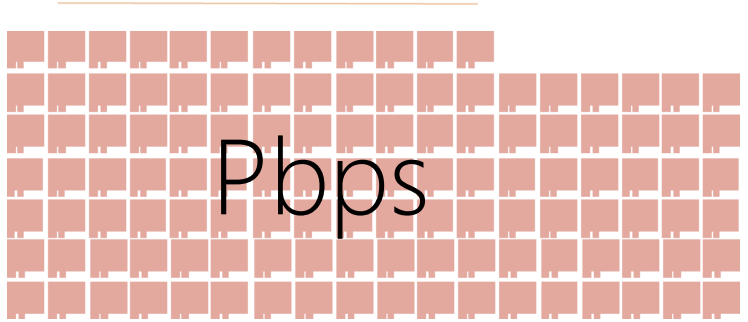
15 billion dollar bet

Microsoft Cloud Network



Microsoft Cloud Network



	2010	2015
Compute Instances	100K 	Millions 
Azure Storage	10's of PB 	Exabytes 
Datacenter Network	10's of Tbps 	Pbps 

Scale

>85%
Fortune 500 using Microsoft Cloud

425 MILLION
Azure Active Directory users

1 TRILLION
Azure Event Hubs events/month

>93,000
New Azure customers a month

>18 BILLION
Azure Active Directory authentications/week

>60 TRILLION
Azure storage objects

1 out of 4 Azure VMs are Linux VMs

1,400,000
SQL databases in Azure

>5 MILLION
requests/sec

Microsoft

Agenda

Consistent cloud design principles for SDN

Physical and Virtual networks, NFV

Integration of enterprise & cloud, physical & virtual

Future: reconfigurable network hardware

Demo

Career Advice

Acknowledgements

Azure SmartNIC

Microsoft

Cloud Design Principles

Scale-out N-Active Data Plane

Embrace and Isolate failures

Centralized control plane: drive network to target state

Resource managers service requests, while meeting system wide objectives

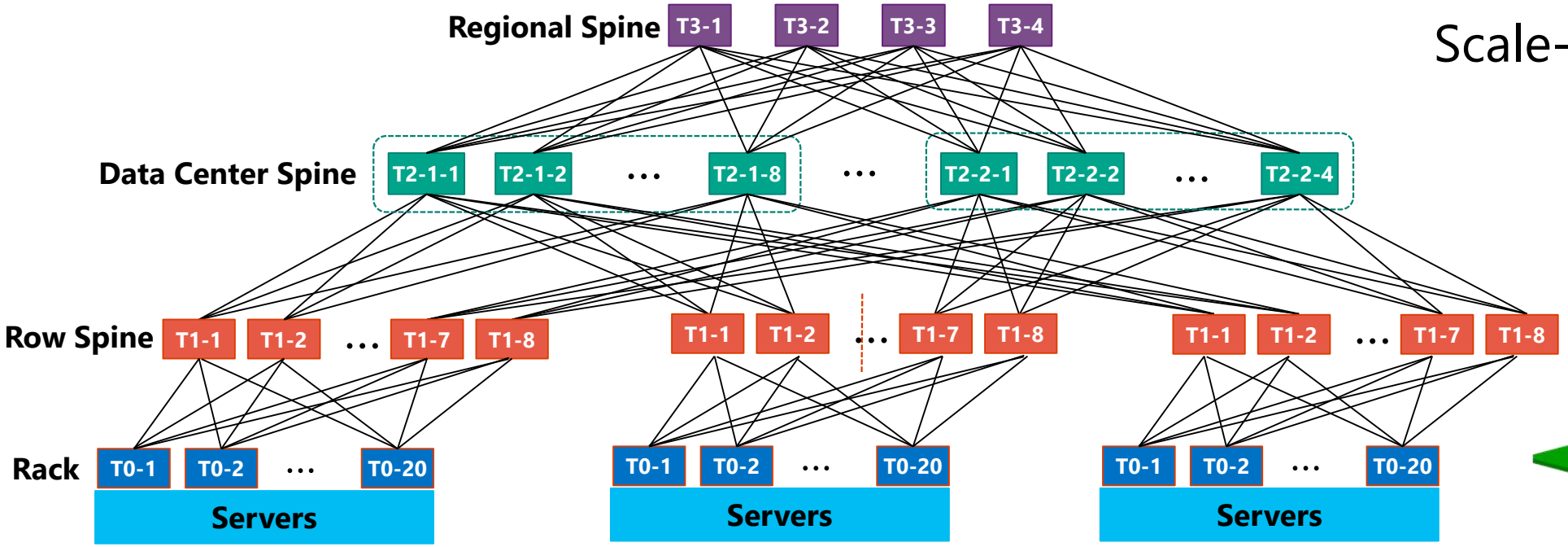
Controllers drive each component relentlessly to the target state

Stateless agents plumb the policies dictated by the controllers

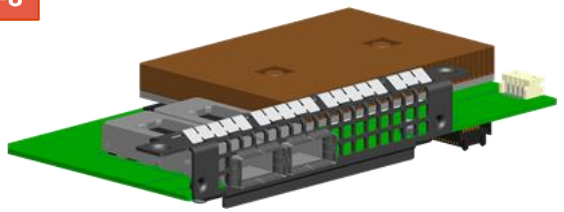
These principles are built into every component

Hyperscale Physical Networks

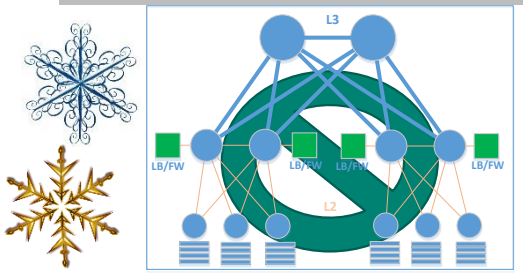
VL2 → Azure Clos Fabrics with 40G NICs



Scale-out, active-active



Scale-up, active-passive



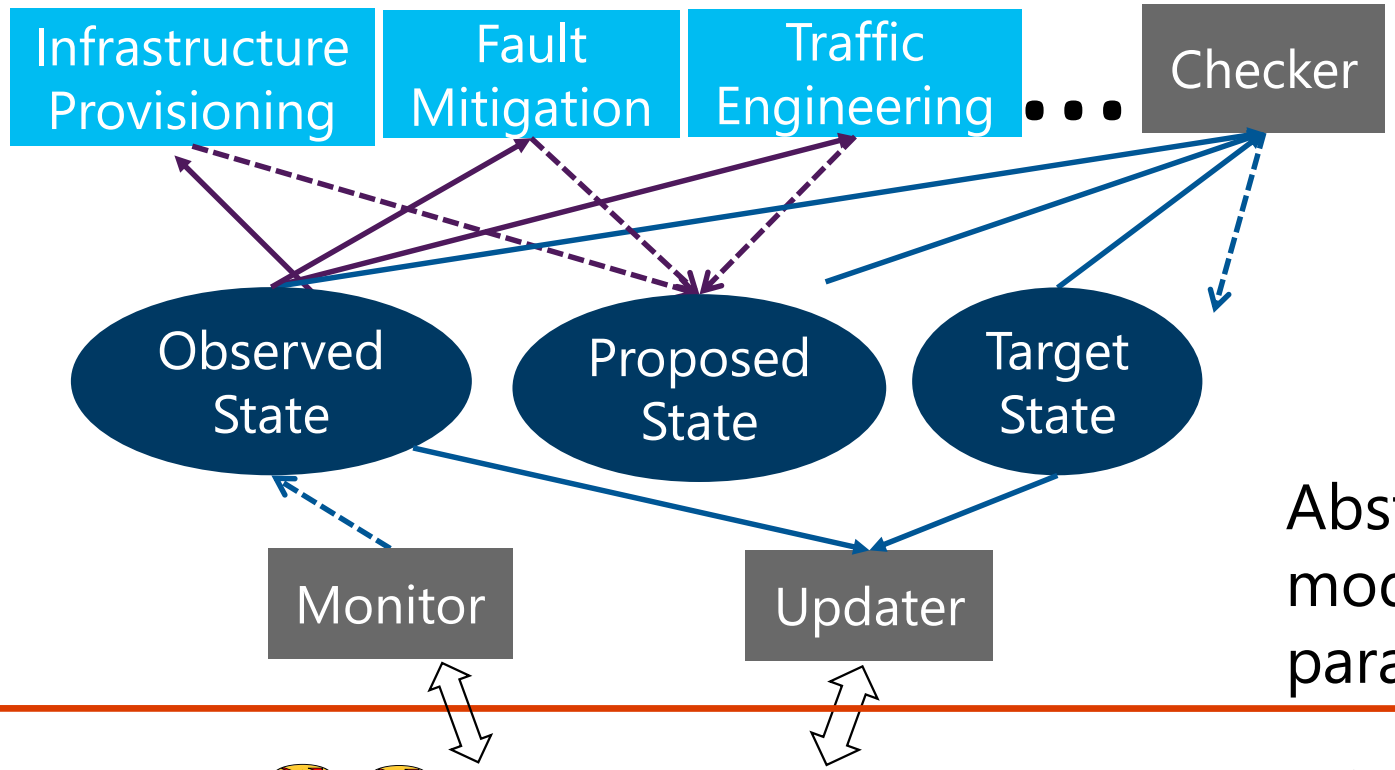
Outcome of >10 years of history, with major revisions every six months

Challenges of Scale

- Clos network management problem
  - Huge number of paths, ASICS, switches to examine, with a dynamic set of gray failure modes, when chasing app latency issues at 99.995% levels
- Solution
  - Infrastructure for graph and state tracking to provide an app platform
  - Monitoring to drive out gray failures
  - Azure Cloud Switch OS to manage the switches as we do servers

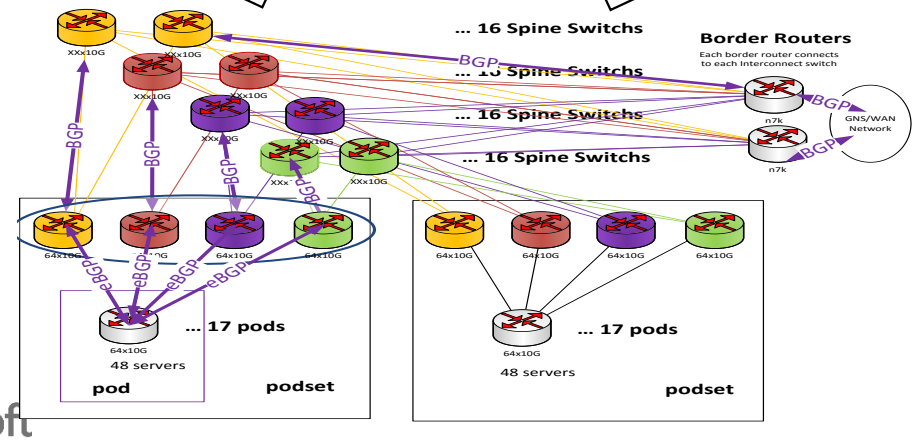
Capex $/Tbps and Opex are 100X smaller than counterparts for prior networks

Azure State Management System Architecture



Centralized control plane: drive network to target state

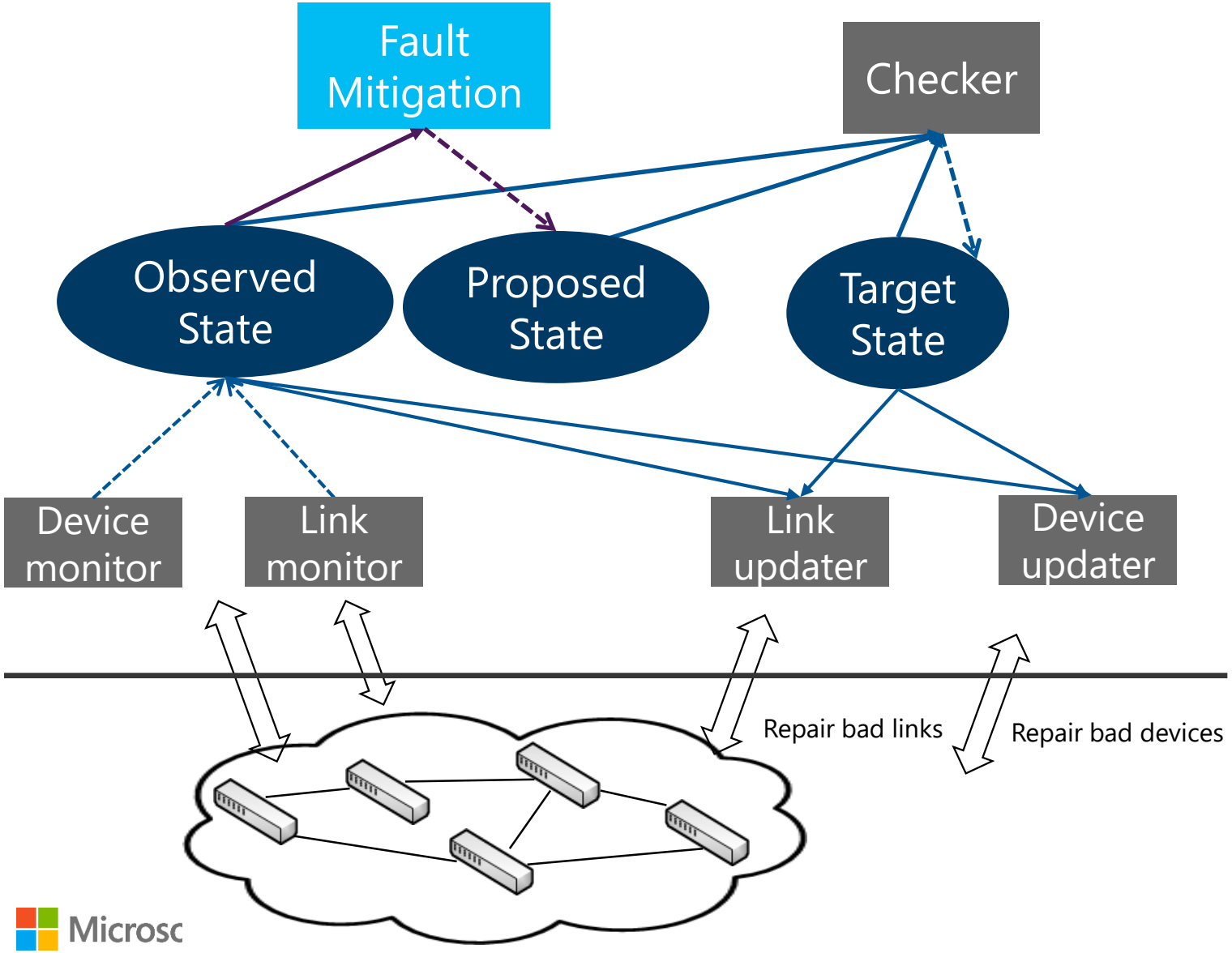
Abstract network graph and state model: the basic programming paradigm

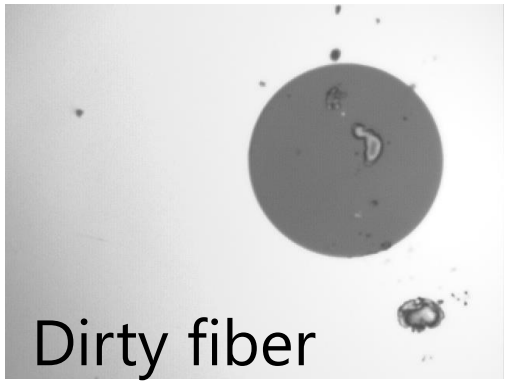


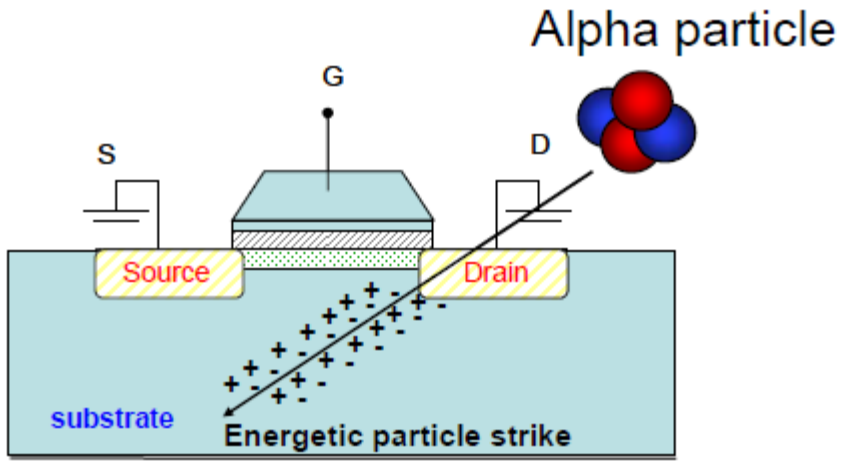
High scale infrastructure is complex: multiple vendors, designs, software versions, failures

Microsoft

App I: Automatic Failure Mitigation

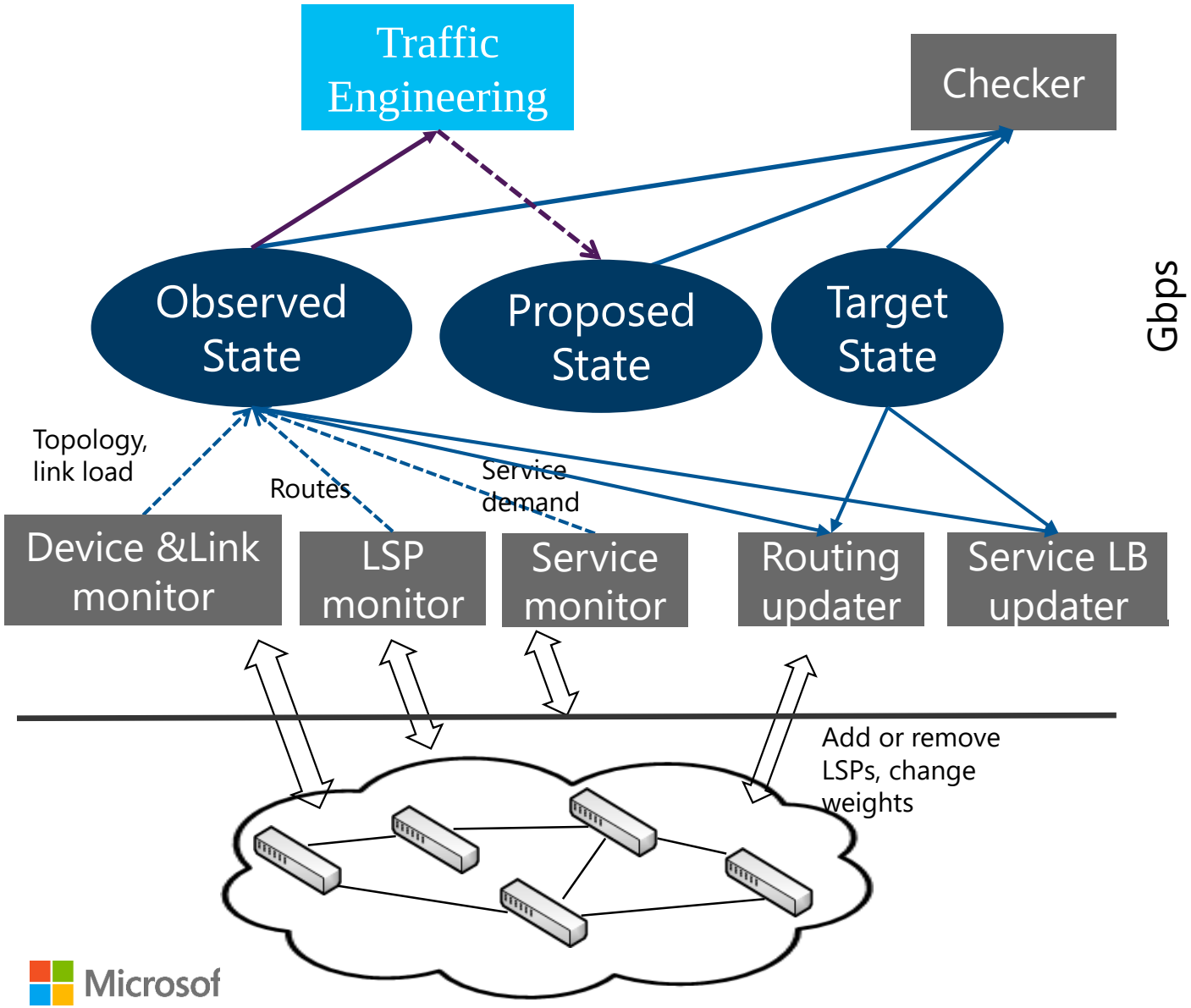


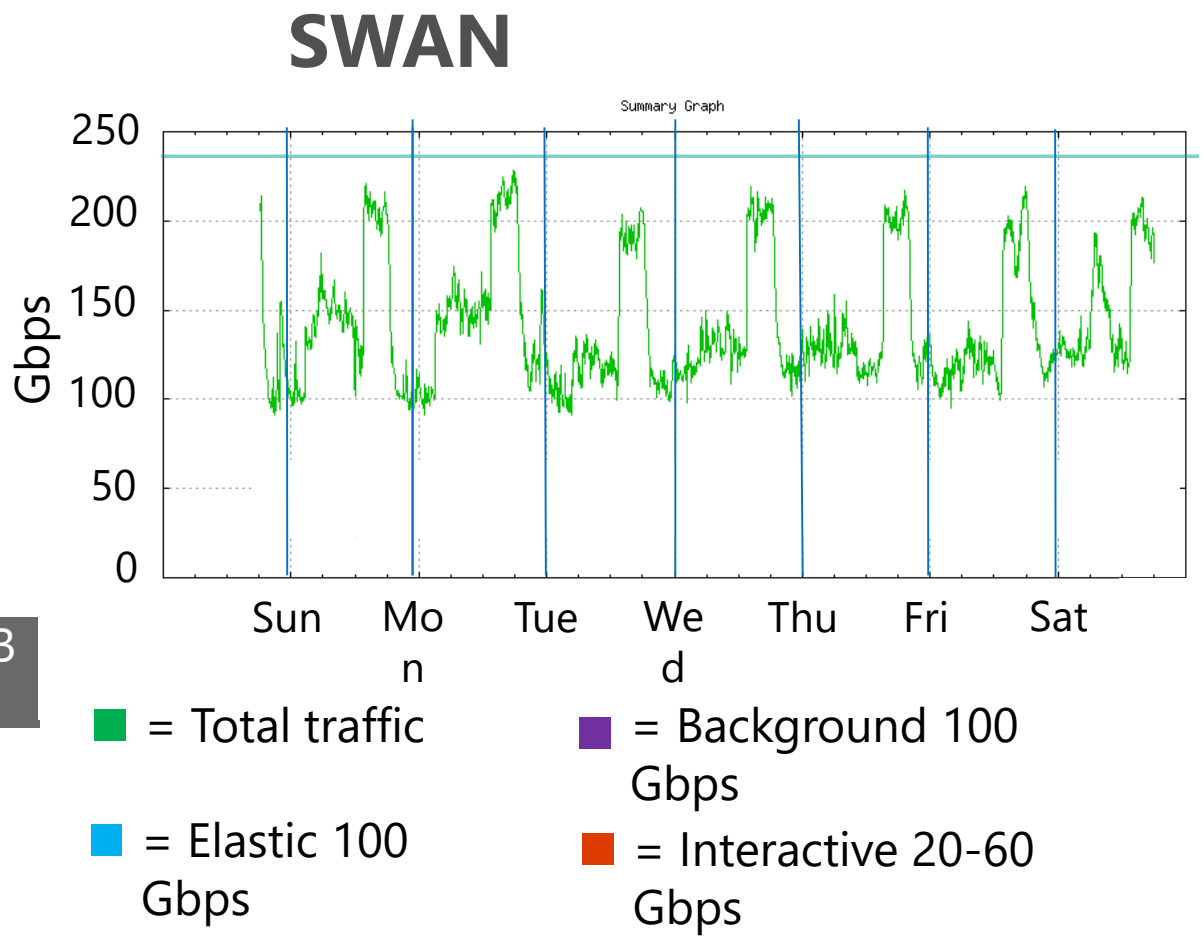




Parity Error in ASIC

App II: Traffic Engineering Towards High Utilization



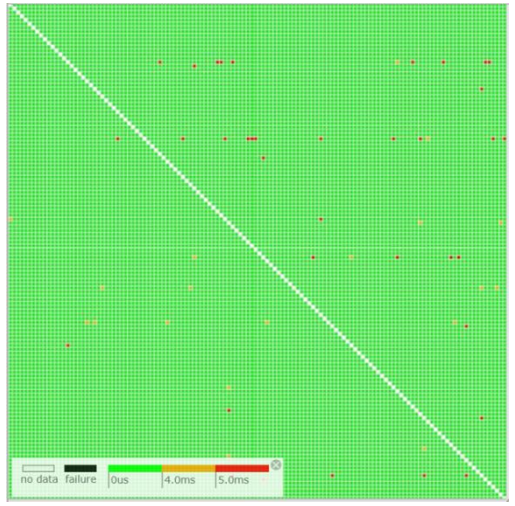


Microsof

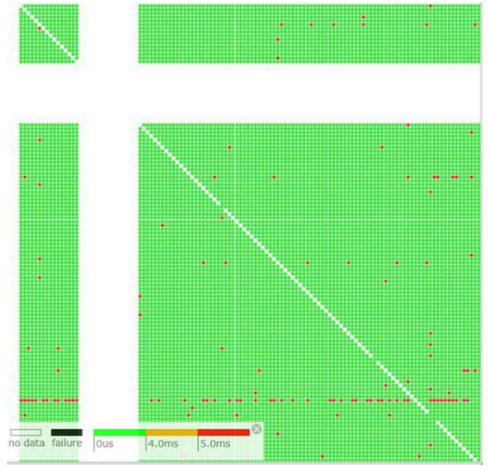
Azure Scale Monitoring – Pingmesh

- Problem: Is it the app or the net explaining app latency issues?
- Solution: Measure the network latency between any two servers
- Full coverage, always-on, brute force
- Running in Microsoft DCs for near 5 years, generating 200+B probes every day

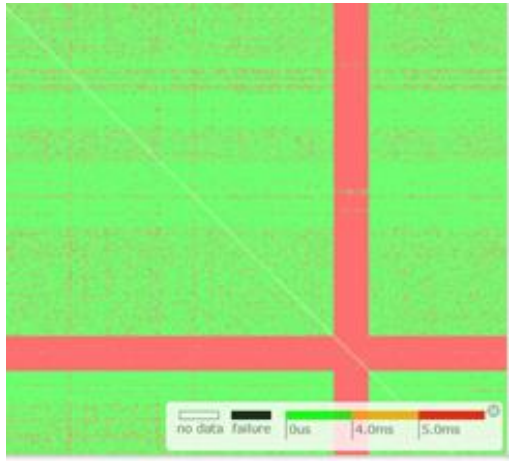
Use high scale cloud computing to monitor cloud network



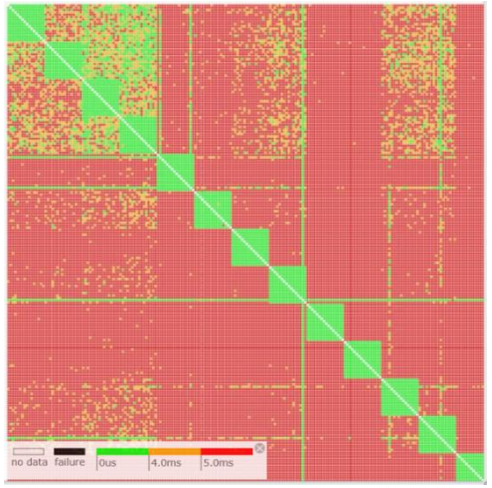
Normal



Podset down



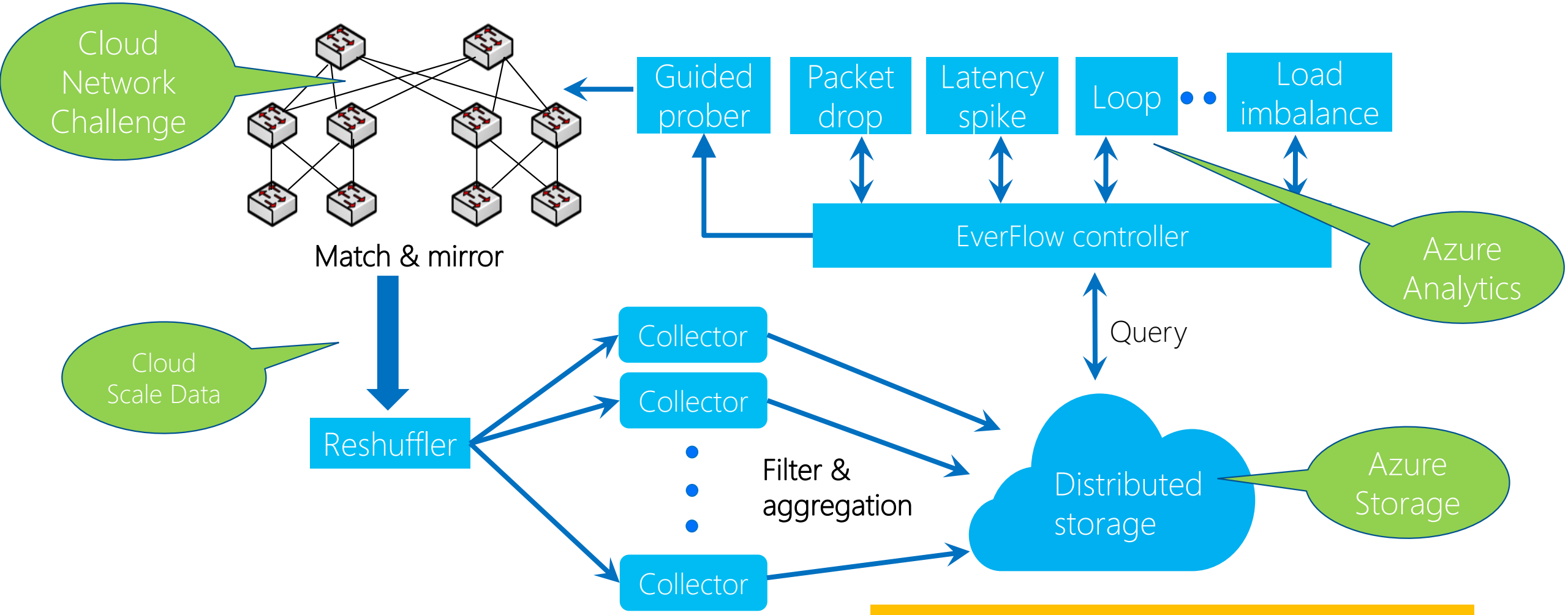
Podset failure



Spine failure

Microsoft

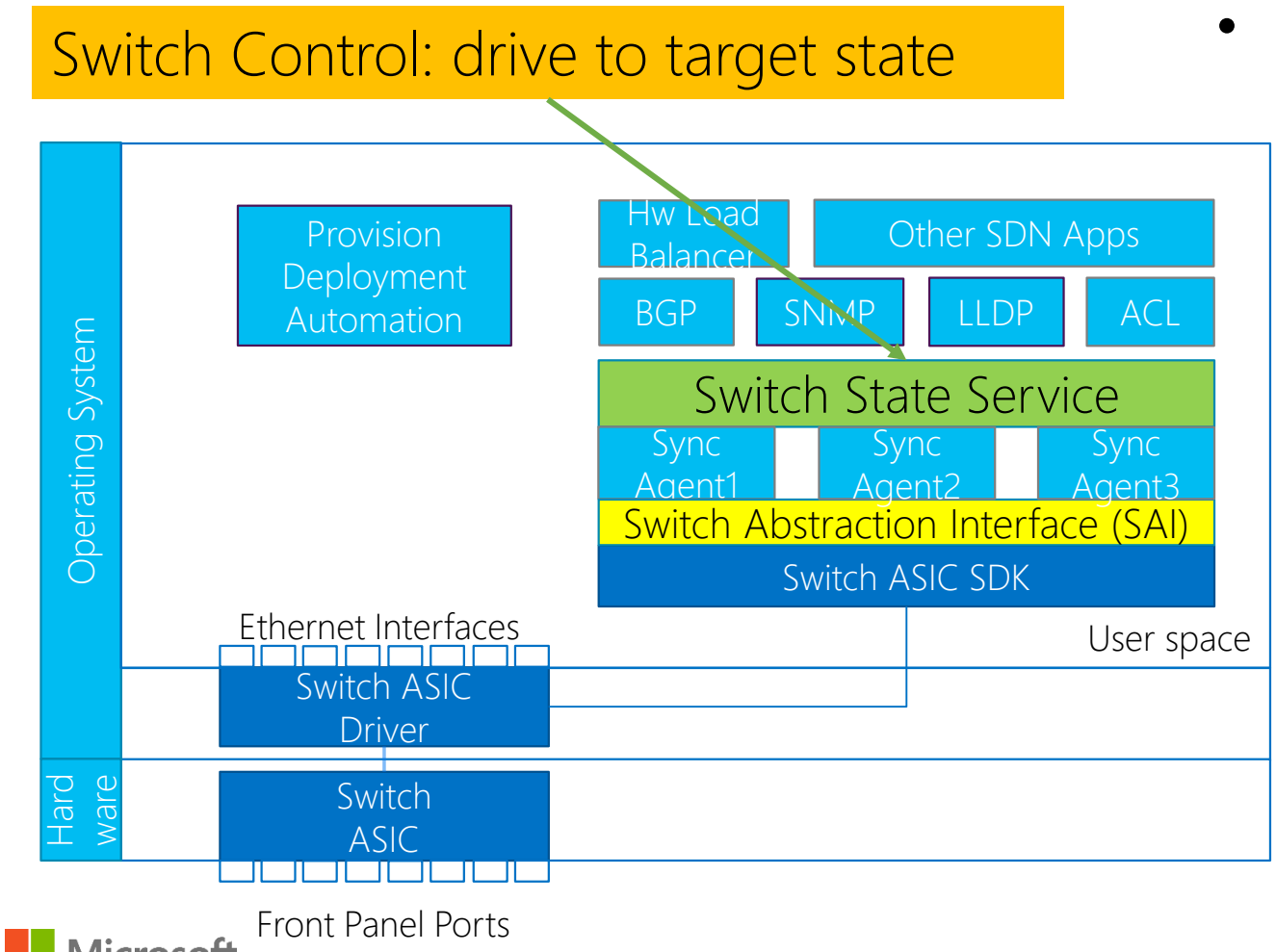
EverFlow: Packet-level Telemetry + Cloud Analytics

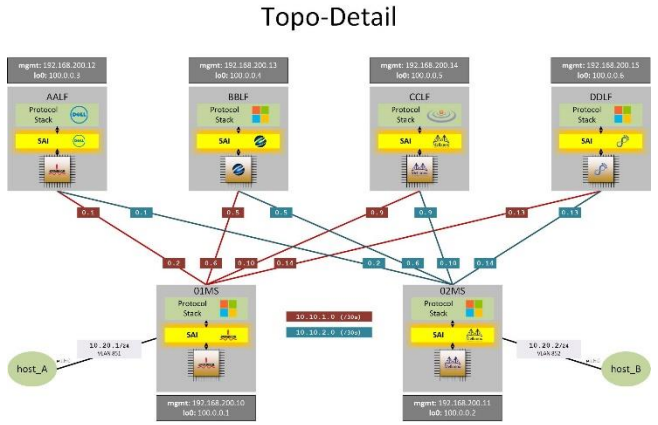


Fast time to pinpoint root cause

Azure Cloud Switch – Open Way to Build Switch OS

- SAI collaboration is industry wide
- SAI simplifies bringing up Azure Cloud Switch (Azure's switch OS) on new ASICS





Thurs PM demo

SAI is on github

Hyperscale Virtual Networks

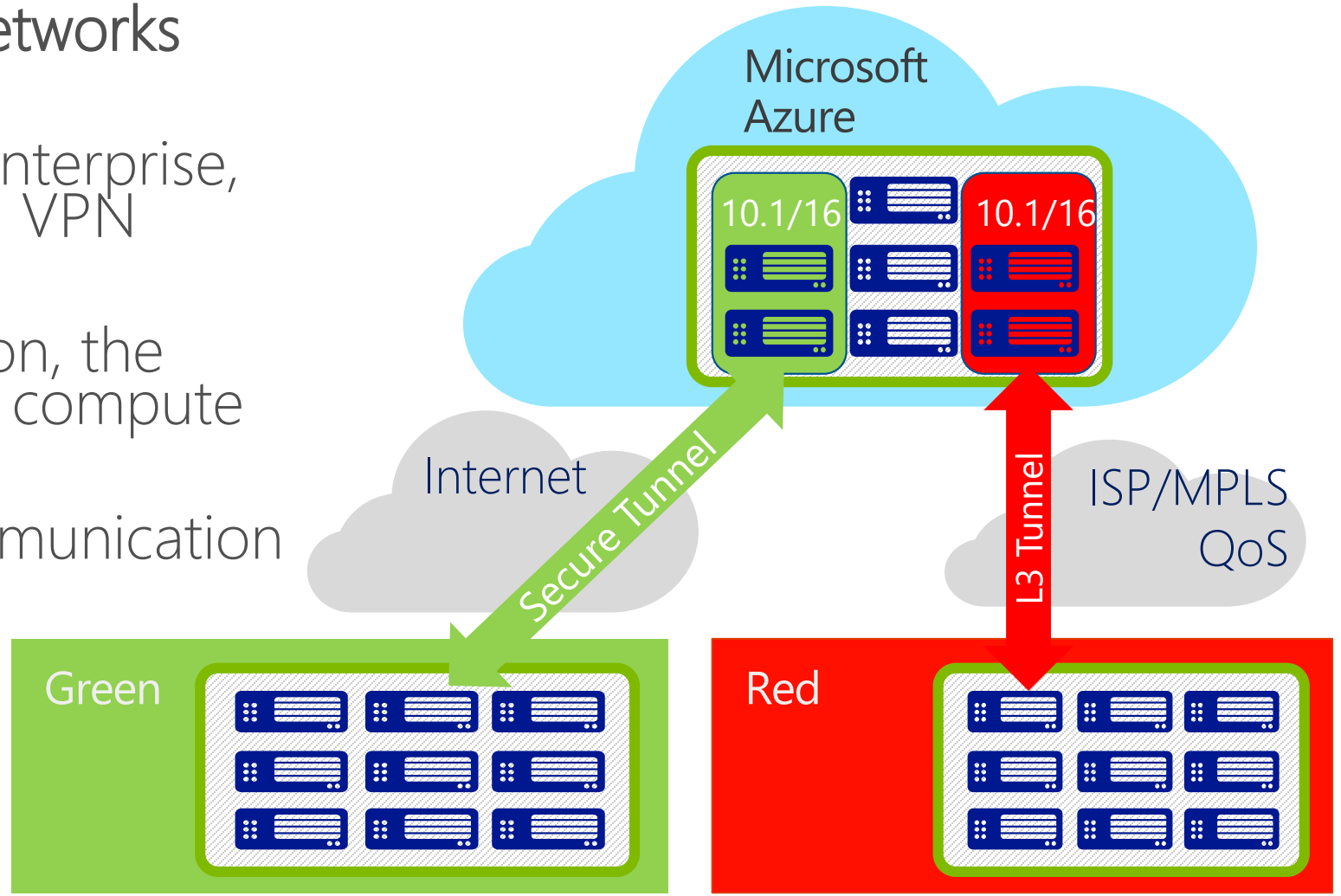
Network Virtualization (VNet)

Microsoft Azure Virtual Networks

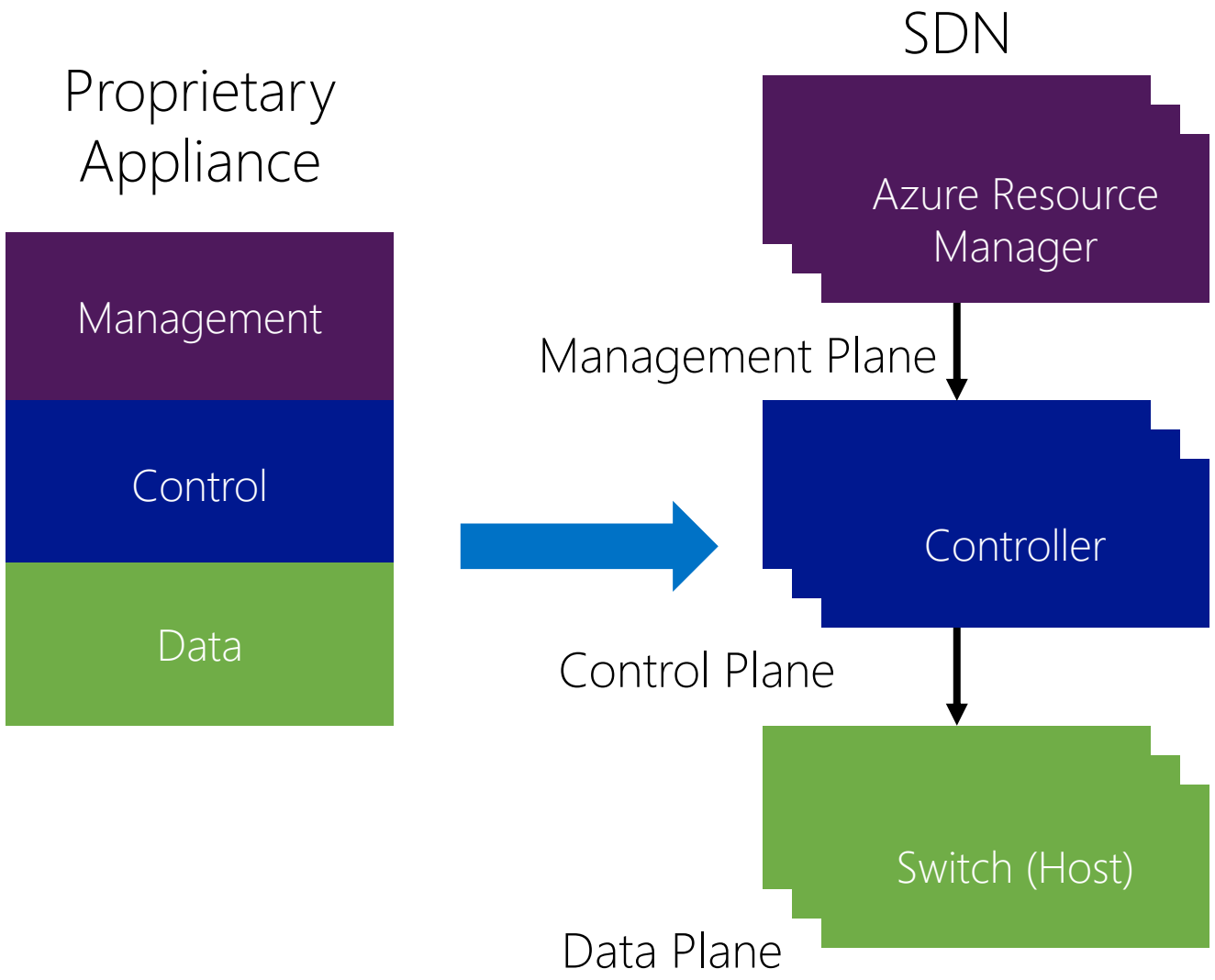
Azure is the hub of your enterprise, reach to branch offices via VPN

VNet is the right abstraction, the counterpart of the VM for compute

Efficient and scalable communication within and across VNets



Hyperscale SDN: All Policy is in the Host



Key Challenges for Hyperscale SDN Controllers

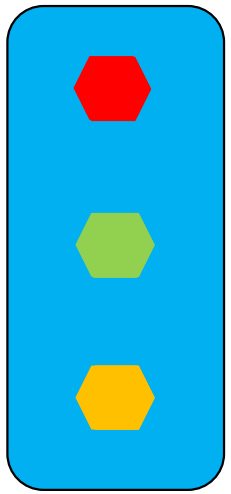
Must scale up to 500k+ Hosts in a region

Needs to scale down to small deployments too

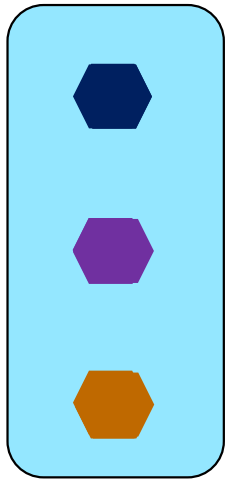
Must handle millions of updates per day

Must support frequent updates without downtime

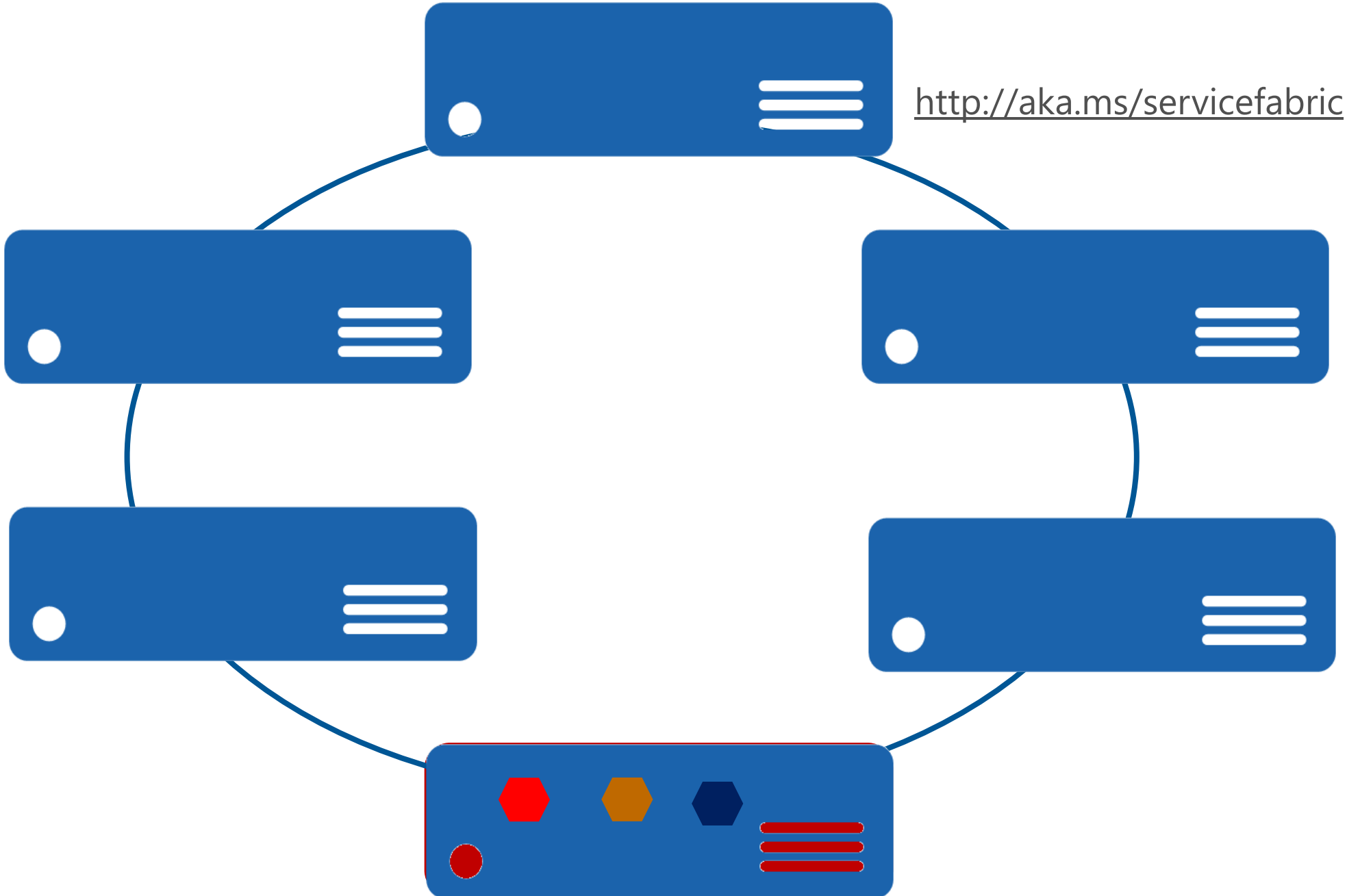
Microsoft Azure Service Fabric: A platform for reliable, hyperscale, microservice-based applications



App1



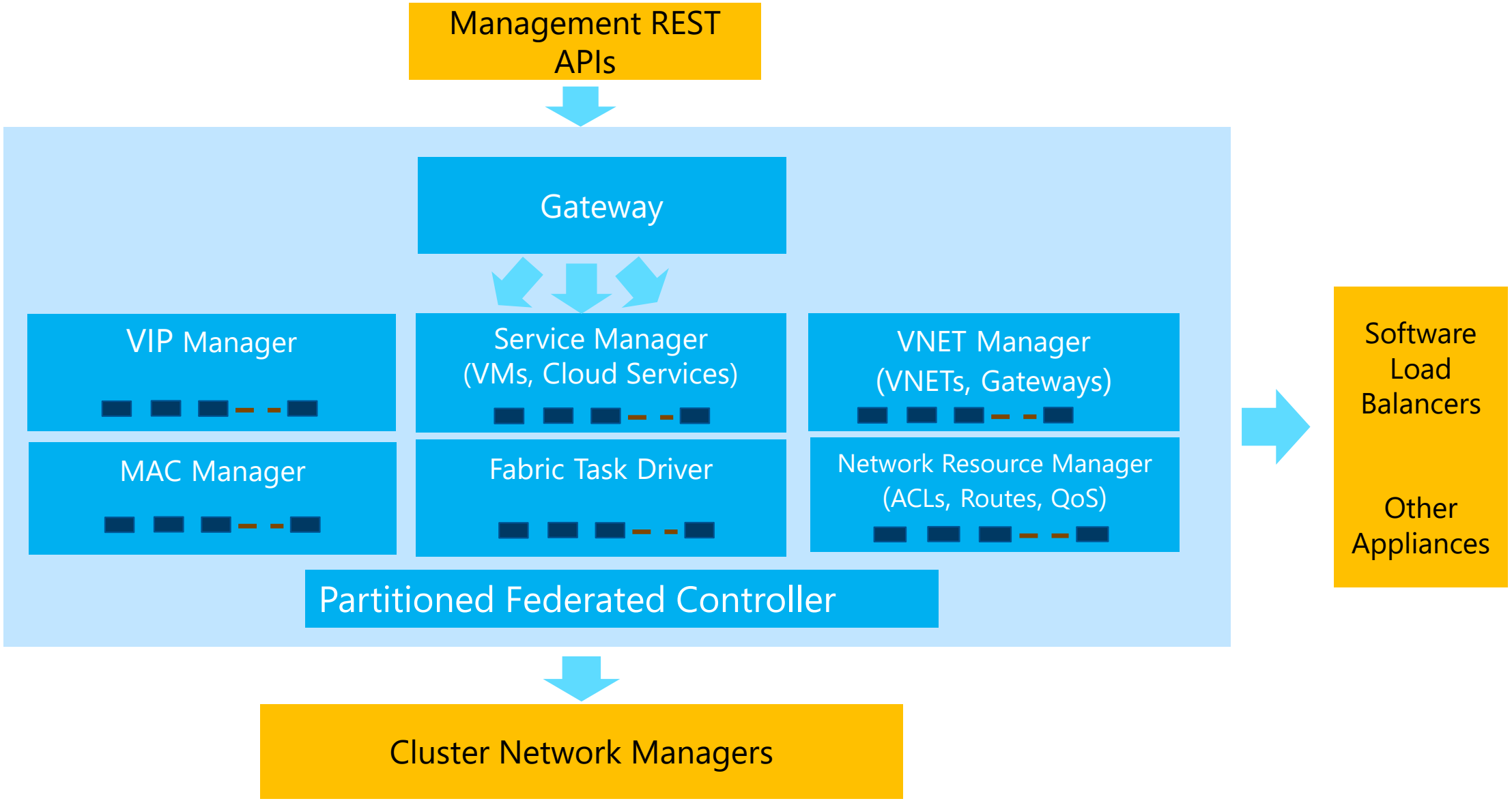
App2



http://aka.ms/servicefabric

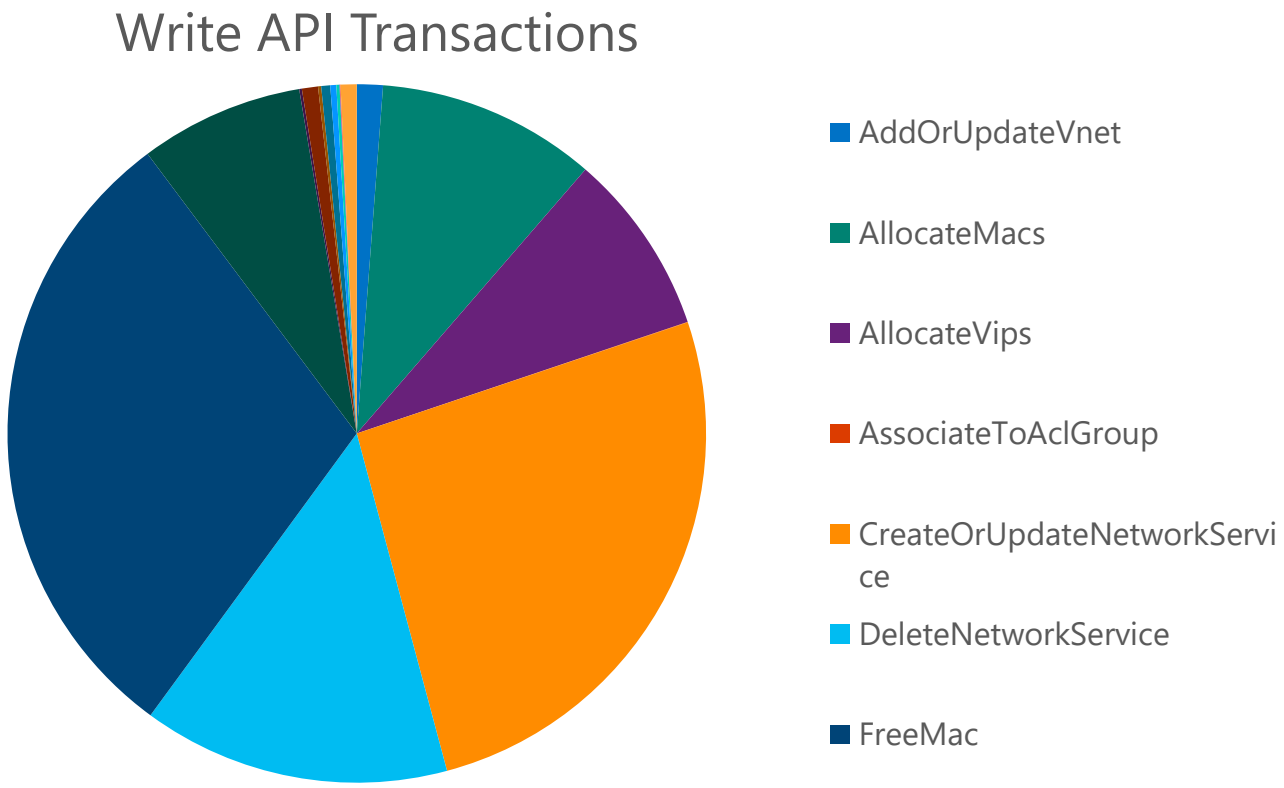
Microsoft

Regional Network Manager Microservices



Regional Network Controller Stats

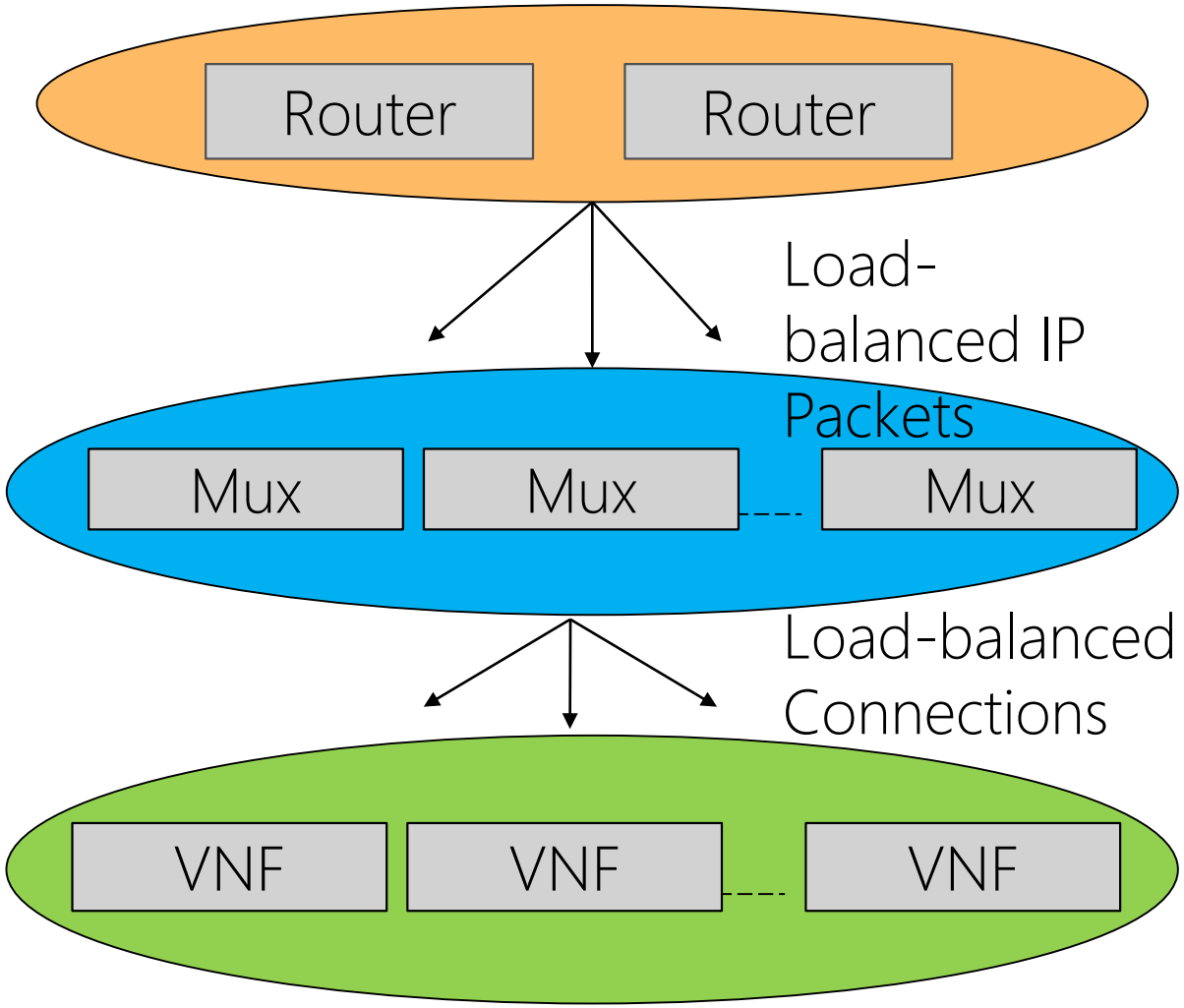
- 10s of millions of API calls per day
- API execution time
  - Read : <50 milliseconds
  - Write : <150 milliseconds
- Varying deployment footprint
  - Smallest : <10 Hosts
  - Largest : >100 Hosts



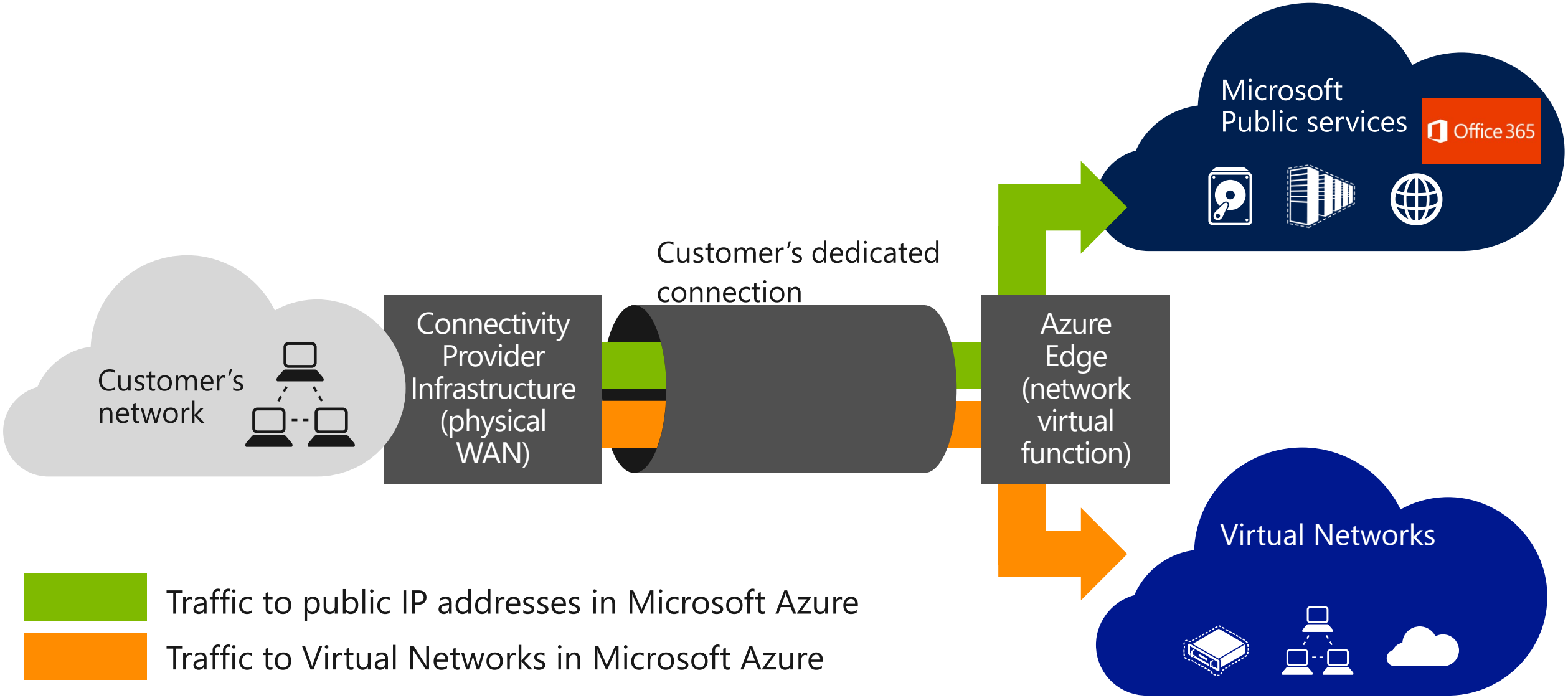
Hyperscale Network Function Virtualization

Azure SLB: Scaling Virtual Network Functions

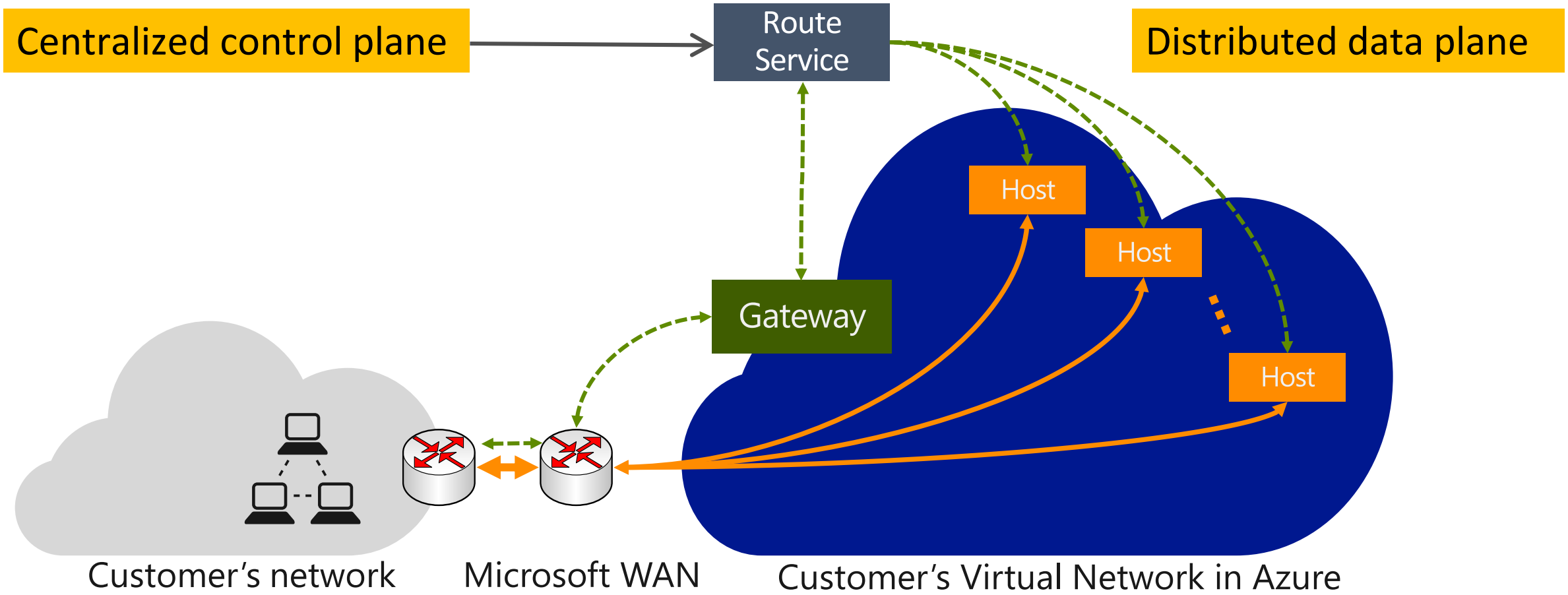
- Key Idea: Decompose Load Balancing into Tiers to achieve scale-out data plane and centralized control plane
- Tier 1: Distribute packets (Layer 3)
  - Routers ECMP
- Tier 2: Distribute connections (Layer 3-4)
  - Multiplexer or Mux
  - Enable high availability and scale-out
- Tier 3: Virtualized Network Functions (Layer 3-7)
  - Example: Azure VPN, Azure Application Gateway, third-party firewall



Express Route: Direct Connectivity to the Cloud



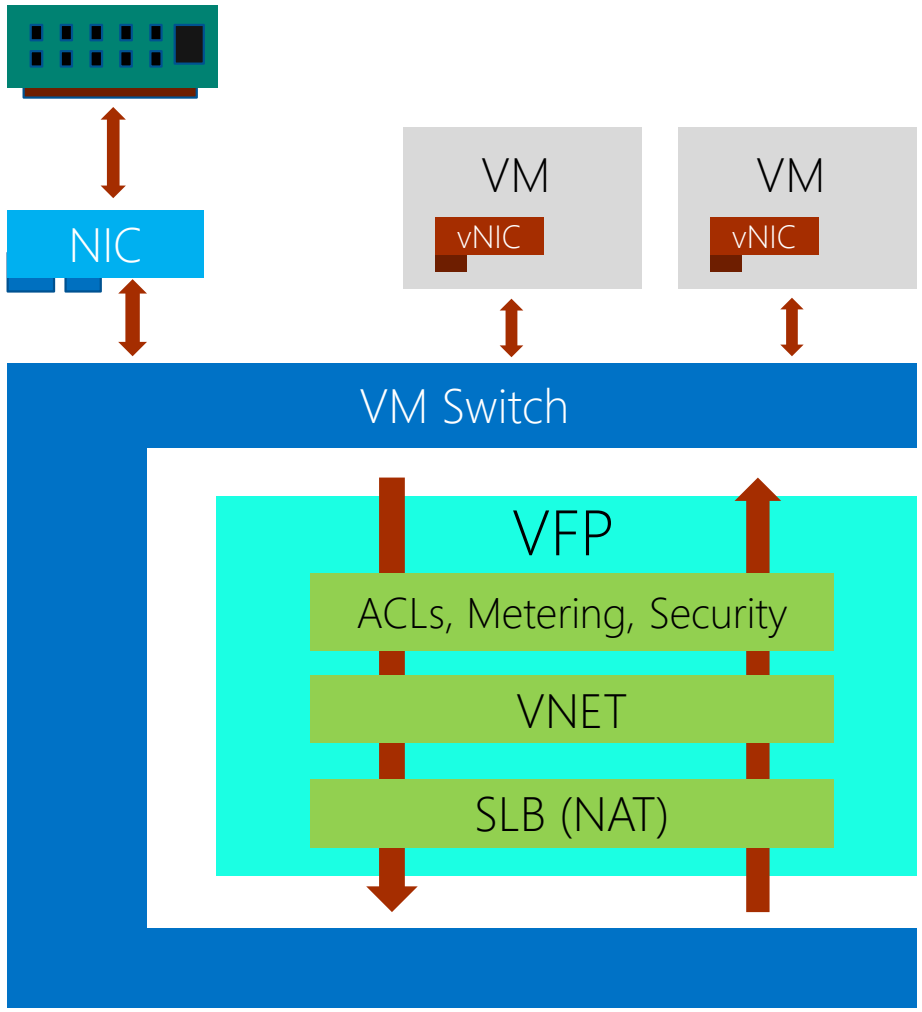
Data Center-Scale Distributed Router



Extreme Scale: 10K customers, 10K routes each → 100M routes

Building a Hyperscale Host SDN

Virtual Filtering Platform (VFP)



Acts as a virtual switch inside Hyper-V VMSwitch

Provides core SDN functionality for Azure networking services, including:

- Address Virtualization for VNET
- VIP -> DIP Translation for SLB
- ACLs, Metering, and Security Guards

Uses programmable rule/flow tables to perform per-packet actions

Supports all Azure data plane policy at 40GbE+ with offloads

Coming to private cloud in Windows Server 2016

Flow Tables: the Right Abstraction for the Host

VMSwitch exposes a typed Match-Action-Table API to the controller

- Controllers define policy
- One table per policy

Key insight: Let controller tell switch exactly what to do with which packets

- e.g. encap/decap, rather than trying to use existing abstractions (tunnels, …)

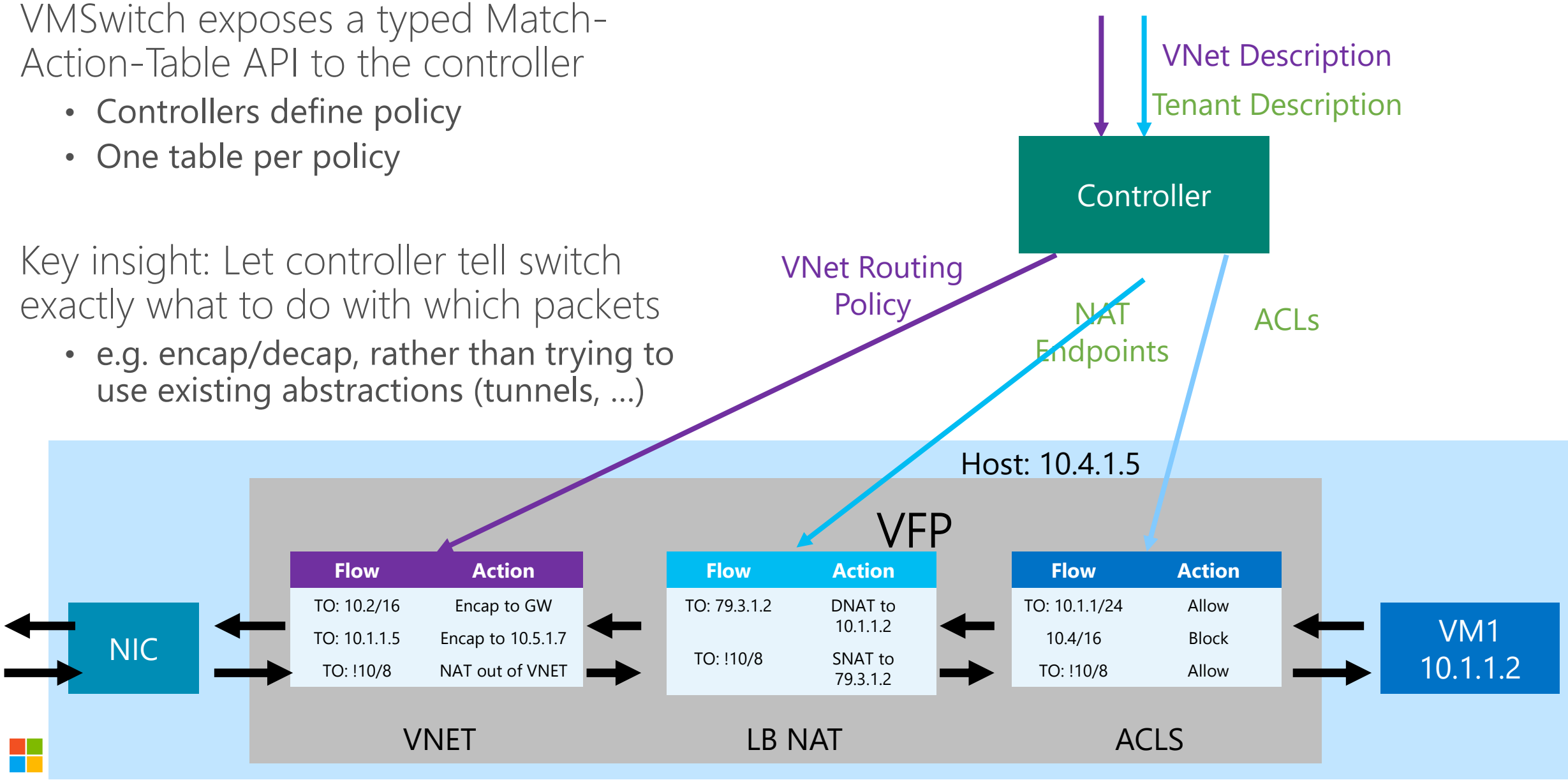
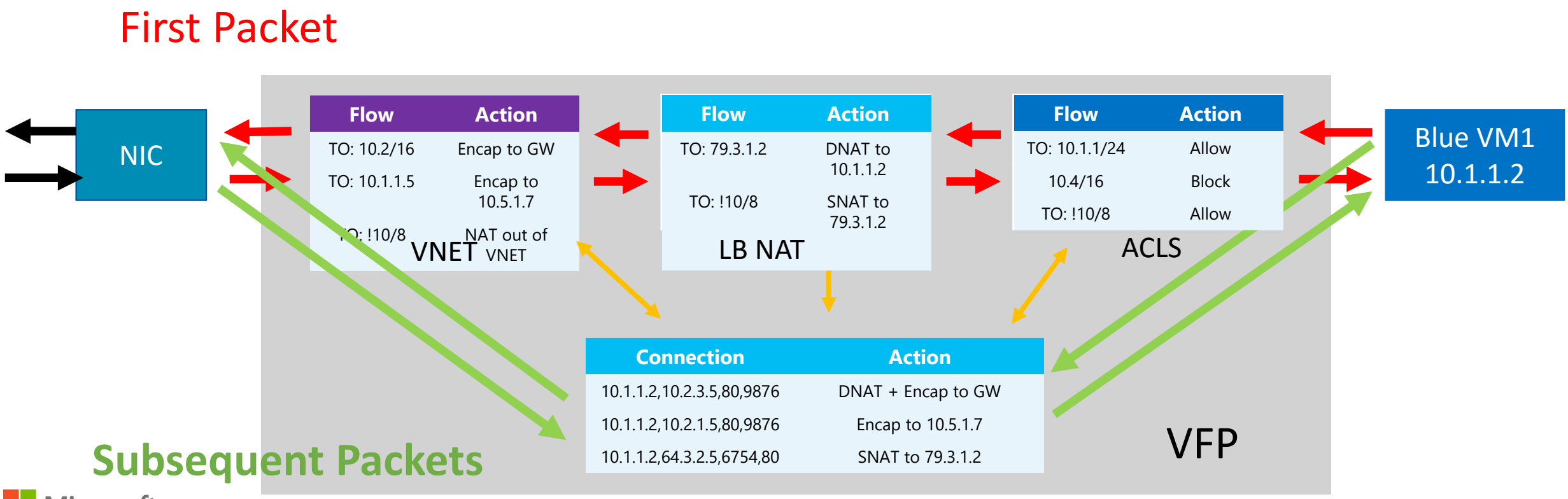
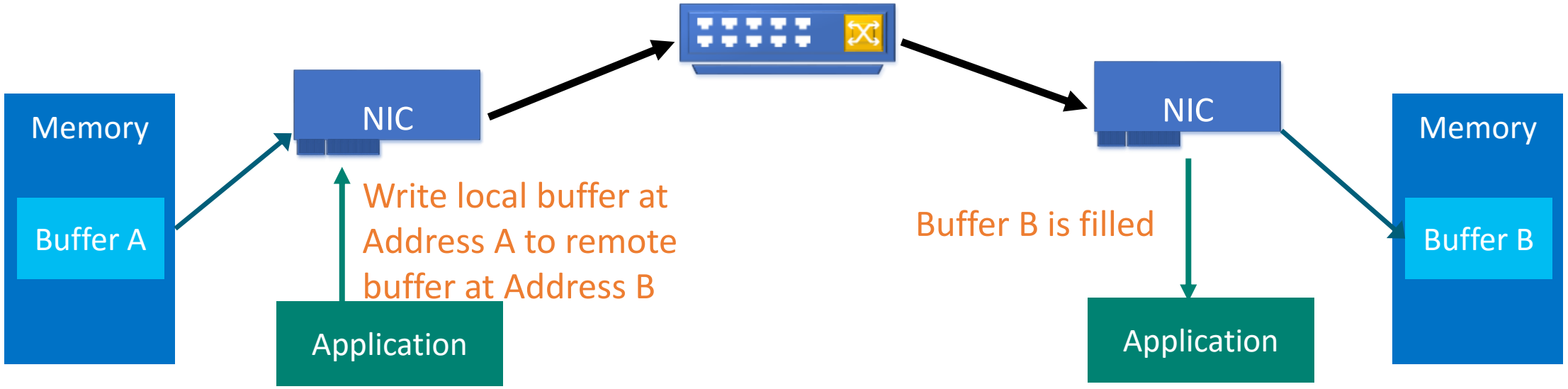


Table Typing/Flow Caching are Critical to Performance

- COGS in the cloud is driven by VM density: 40GbE is here
- First-packet actions can be complex
- Established-flow matches must be typed, predictable, and simple hash lookups



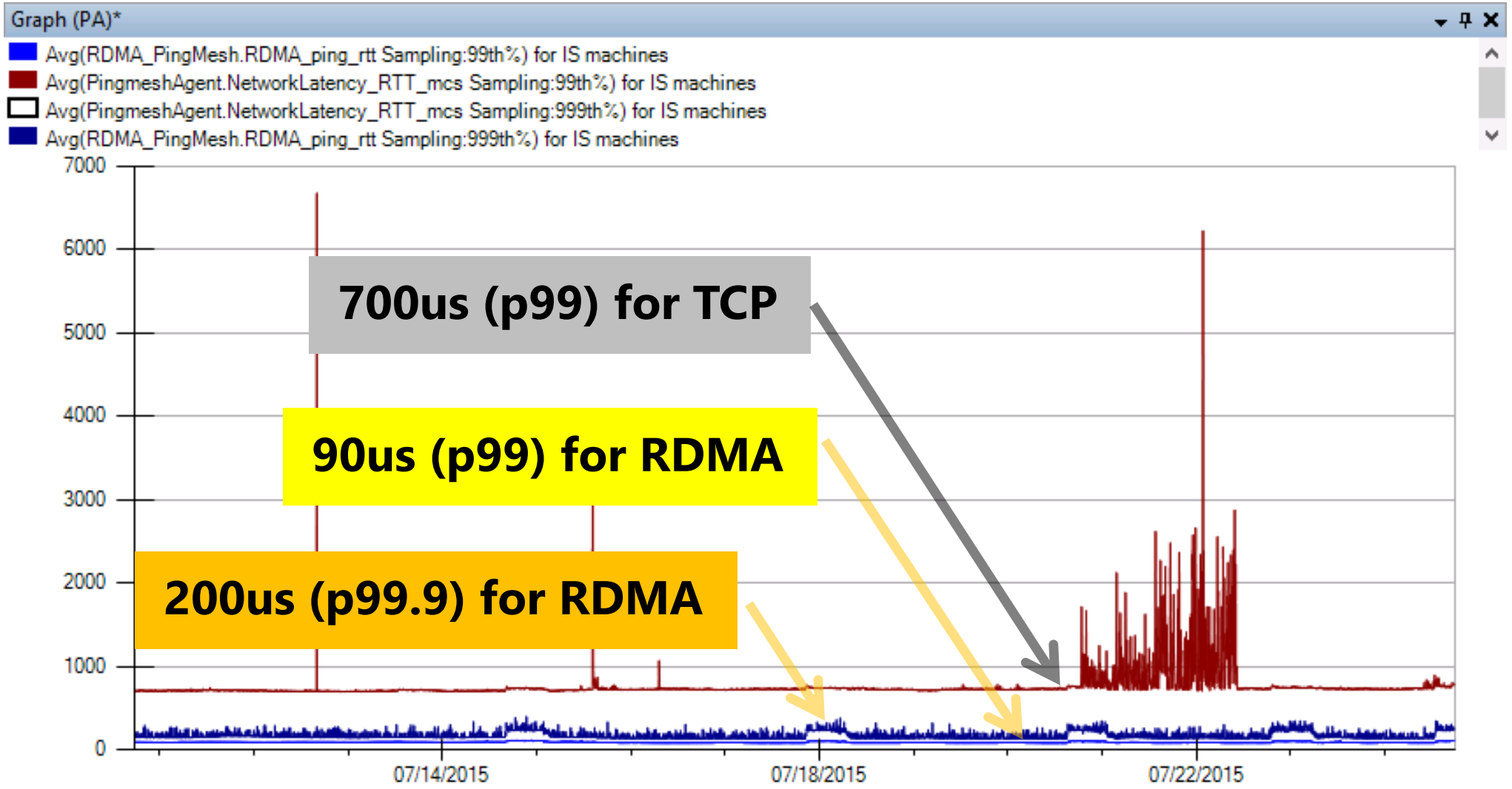
RDMA/RoCEv2 at Scale in Azure



- RDMA addresses high CPU cost and long latency tail of TCP
  - Zero CPU Utilization at 40Gbps
  - µs level E2E latency
- Running RDMA at scale
  - RoCEv2 for RDMA over commodity IP/Ethernet switches
  - Cluster-level RDMA
  - DCQCN* for end-to-end congestion control

*DCQCN is running on Azure NICs

RDMA Latency Reduction at 99.9th %ile in Bing



Microsoft

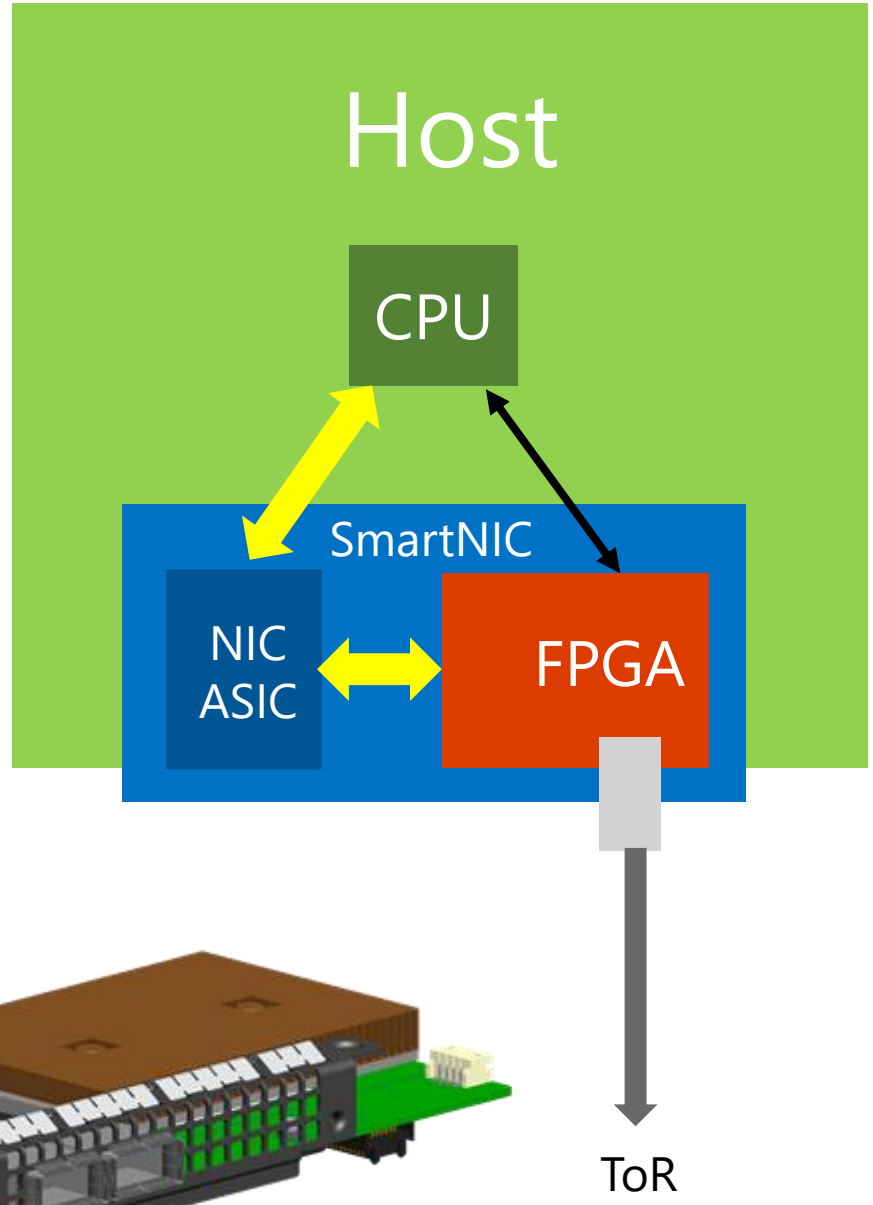
Host SDN Scale Challenges

- Host network is Scaling Up: 1G → 10G → 40G → 50G → 100G
  - The driver is VM density (more VMs per host), reducing COGs
  - Need the performance of hardware to implement policy without CPU
- Need to support new scenarios: BYO IP, BYO Topology, BYO Appliance
  - We are always pushing richer semantics to virtual networks
  - Need the programmability of software to be agile and future-proof

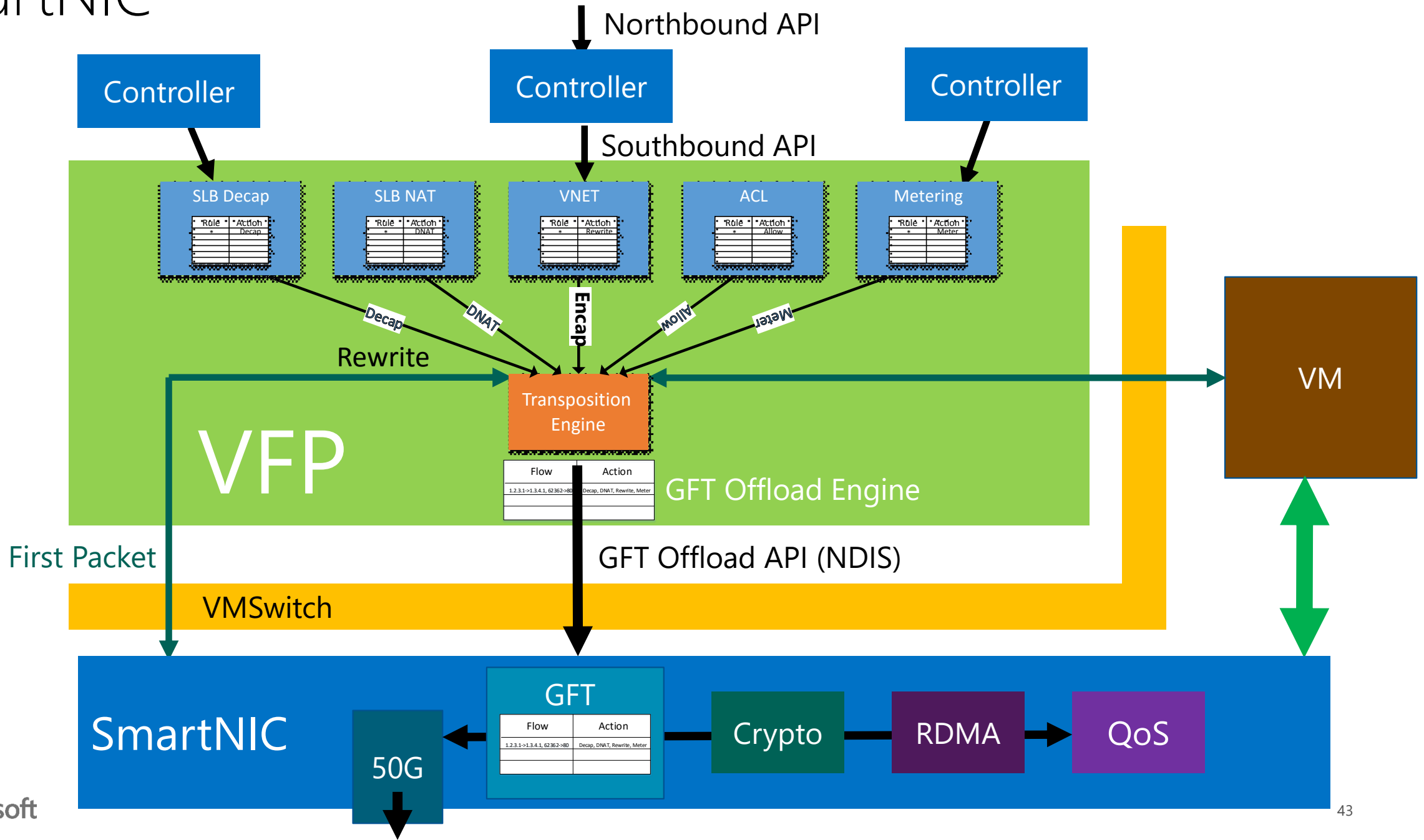
How do we get the performance of hardware with programmability of software?

Azure SmartNIC

- Use an FPGA for reconfigurable functions
  - FPGAs are already used in Bing (Catapult)
  - Roll out Hardware as we do software
- Programmed using Generic Flow Tables (GFT)
  - Language for programming SDN to hardware
  - Uses connections and structured actions as primitives
- SmartNIC can also do Crypto, QoS, storage acceleration, and more…



SmartNIC


Northbound API
Controller
Controller
Controller
Southbound API
SLB Decap
SLB NAT
VNET
ACL
Metering
Decap
DNAT
Encap
Allow
Meter
Rewrite
Transposition Engine
VFP
Flow
Action
1.2.3.1->1.3.4.1, 62362->80
Decap, DNAT, Rewrite, Meter
GFT Offload Engine
VM
First Packet
GFT Offload API (NDIS)
VMSwitch
SmartNIC
GFT
Flow
Action
1.2.3.1->1.3.4.1, 62362->80
Decap, DNAT, Rewrite, Meter
50G
Crypto
RDMA
QoS

Azure SmartNIC

Microsoft

Demo: SmartNIC Encryption

Closing Thoughts

Cloud scale, financial pressure unblocked SDN

Control and systems developed earlier for compute, storage, power helped

Moore's Law helped: order 7B transistors per ASIC

We did not wait for a moment for standards, for vendor persuasion

SDN realized through consistent application of principles of Cloud design

Embrace and isolate failure

Centralize (partition, federate) control and relentlessly drive to target state

Microsoft Azure re-imagined networking, created SDN and it paid off

Career Advice

Cloud

Software → leverage and agility

Even for hardware people

Quantity time

With team, with project

Hard infrastructure problems take >3 years, but it's worth it

Usage and its measurement → oxygen for ideas

Quick wins (3 years is a long time)

Foundation and proof that the innovation matters

Shout-Out to Colleagues & Mentors

AT&T & Bell Labs

- Han Nguyen, Brian Freeman, Jennifer Yates, … and entire AT&T Labs team
- Alumni: Dave Belanger, Rob Calderbank, Debasis Mitra, Andrew Odlyzko, Eric Sumner Jr.

Microsoft Azure

- Reza Baghai, Victor Bahl, Deepak Bansal, Yiqun Cai, Luis Irun-Briz, Yiqun Cai, Alireza Dabagh, Nasser Elaawar, Gopal Kakivaya, Yousef Khalidi, Chuck Lenzmeier, Dave Maltz, Aaron Ogus, Parveen Patel, Mark Russinovich, Murari Sridharan, Marne Staples, Junhua Wang, Jason Zander, and entire Azure team
- Alumni: Arne Josefsberg, James Hamilton, Randy Kern, Joe Chau, Changhoon Kim, Parantap Lahiri, Clyde Rodriguez, Amitabh Srivastava, Sumeet Singh, Haiyong Wang

Academia

- Nick Mckeown, Jennifer Rexford, Hui Zhang

MSFT @ SIGCOMM'15

DCQCN
Congestion control for large RDMA deployments
2000x reduction in Pause Frames, 16x better performance

Everflow
Packet-level telemetry for large DC networks
10^6x reduction in trace overhead, pinpoint accuracy

PingMesh
DC network latency measurement and analysis
200 billion probes per day!

Corral
Joint data & compute placement for big data jobs
56% reduction in completion time over Yarn

Silo
Virtual networks with guaranteed bw and latency
No changes to host stack!

R2C2
Network stack for rack-scale computing
Rack is the new building block!

Hopper
Speculation-aware scheduling of big-data jobs
66% speedup of production queries

Iridium
Low-latency geo-distributed analytics
19x query speedup, 64% reduction in WAN traffic

Eden
Enable network functions at end host
Dynamic, stateful policies, F#-based API

Microsoft See http://research.microsoft.com/en-us/um/cambridge/events/sigcomm2015/papers.html