Crimson Owl

Technologies

Crimson Owl Technologies

https://www.crimsonowl.nl

Evil mastermind and head of tinkering

Former MSFT (PFE, CSA, CXP -10 years), IT since 2003…

Technology expertise:

- Azure Infrastructure
- Azure Networking
- Identity
- M365
- Security
- Governance
- Containerization
- VDI
- Azure Kubernetes
- Automation
- Design & Architecture
- Intune

Connect on LinkedIn!

LEVEL 300 DEEP DIVE

Battling the Assumptions

How what you think you know about Azure Networking is a lie

Today's Journey

1. The Foundation: How Azure Really Works
2. Myth: VNets Are Like Physical Networks
3. Myth: Subnets Are Broadcast Domains
4. Myth: ARP and DHCP Work Normally
5. Myth: Routing Works Like On-Prem
6. Myth: NSGs Are Simple ACLs
7. The VFP Deep Dive
8. Accelerated Networking & SR-IOV

45 minutes

The Traditional Networking Mindset

THE TRAP

Network engineers come to Azure with decades of physical networking knowledge - and it betrays them.

We think in terms of:

- Switches and routers
- VLANs and broadcast domains
- ARP tables and MAC learning
- Spanning tree and routing protocols
- Physical cables and port channels

"

Azure networking looks like traditional networking, but it's fundamentally different under the hood.

This mismatch causes design failures, troubleshooting nightmares, and security gaps.

SECTION 1

The Foundation: Azure SDN Architecture

Azure's networking is built on **Software Defined Networking (SDN)** running on Hyper-V hosts. The key components:

- **Virtual Filtering Platform (VFP)** - The programmable vSwitch
- **Network Controller** - Central policy management
- **VXLAN Encapsulation** - Overlay networking
- **Host Agent** - Programs VFP with policies

DEPLOYED SCALE

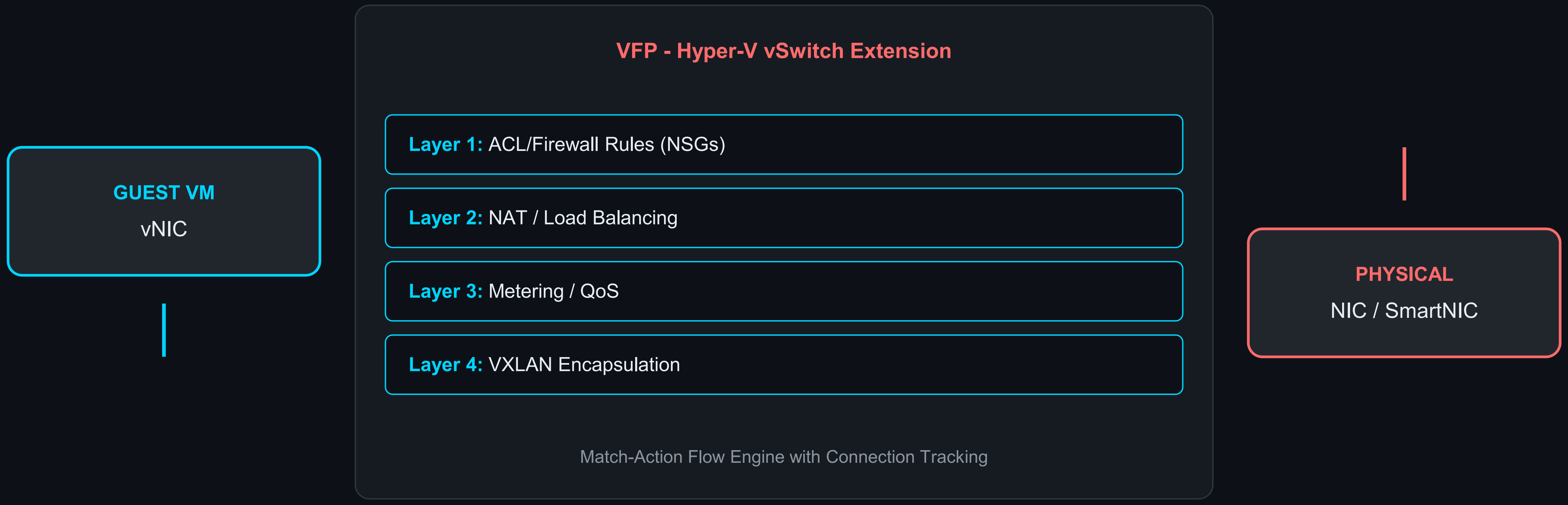
1M+

hosts running VFP

Scaling from 1 Gbps to 200 Gbps per host with SmartNIC offloads

Source: Microsoft Research - VFP NSDI 2017

Virtual Filtering Platform (VFP) Architecture



GUEST VM

vNIC

VFP - Hyper-V vSwitch Extension

Layer 1: ACL/Firewall Rules (NSGs)

Layer 2: NAT / Load Balancing

Layer 3: Metering / QoS

Layer 4: VXLAN Encapsulation

Match-Action Flow Engine with Connection Tracking

PHYSICAL

NIC / SmartNIC

Inbound: Layers processed top → bottom

Outbound: Layers processed bottom → top

MYTH #1

"Virtual Networks are like Physical Networks"

The VNet abstraction hides a completely different reality

The Truth: Azure VNets Are a Venn Diagram

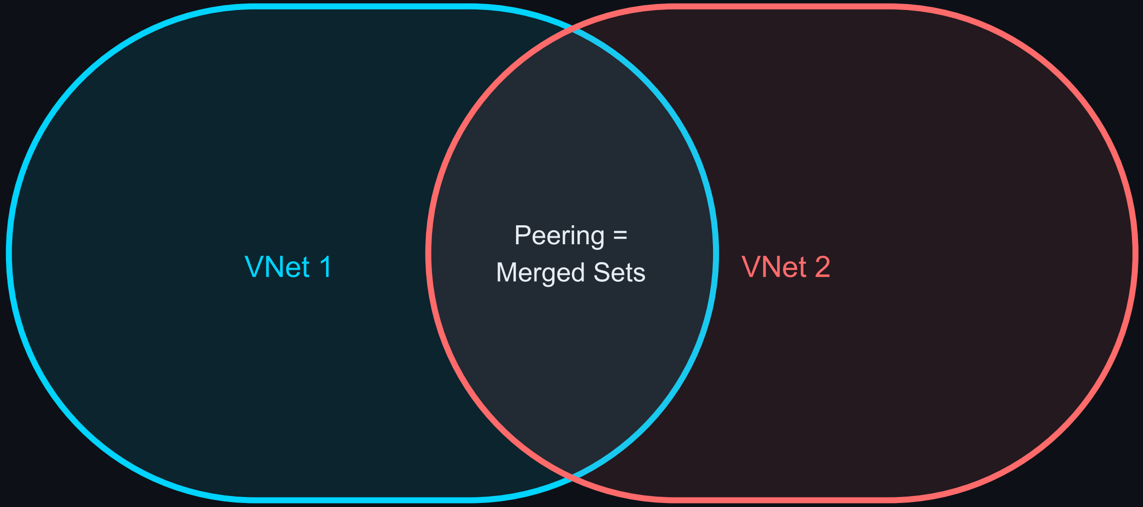
WHAT YOU THINK

A VNet is a virtualized switch/router that traffic flows through

WHAT'S ACTUALLY TRUE

A VNet is **metadata** that instructs the fabric which NICs can route to each other. There is no "network" - just policy.

Creating a VNet tells Azure: "These NICs belong to the same routing domain." The physical network has no knowledge of your VNet structure.

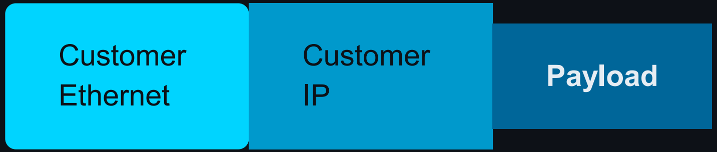


Peering doesn't create a "pipe" - it just expands which NICs can reach each other

How Packets Actually Travel: VXLAN Encapsulation

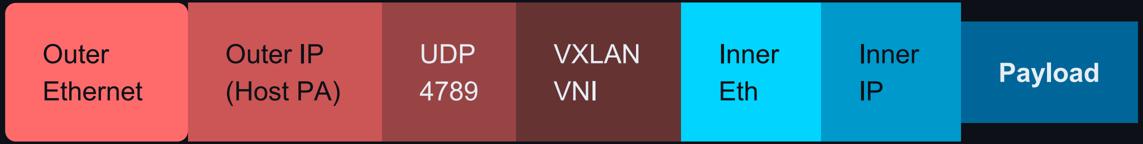
Customer packets are wrapped in VXLAN headers and sent over Microsoft's physical network. Your "Layer 2" traffic is actually Layer 3+.

Original Customer Packet:



Customer Ethernet | Customer IP | **Payload**

After VXLAN Encapsulation:



Outer Ethernet | Outer IP (Host PA) | UDP 4789 | VXLAN VNI | Inner Eth | Inner IP | **Payload**

MTU Impact

Internal Azure MTU is **1400** (not 1500) due to encapsulation overhead

VNI = 24-bit

~16 million virtual networks vs. 4,096 VLANs

MYTH #2

"Subnets are Broadcast Domains"

No broadcast. No multicast. No L2 switching.

What Azure VNets Block by Design

BLOCKED TRAFFIC TYPES

- ✗ **Multicast** - No IGMP, no multicast routing
- ✗ **Broadcast** - L2 and L3 broadcasts blocked
- ✗ **GRE packets** - Generic Routing Encapsulation
- ✗ **IP-in-IP** - Encapsulated packets
- ✗ **DHCP unicast** - Ports UDP/67-68 intercepted

REAL-WORLD IMPACT

- No VRRP/HSRP for HA - use Azure Load Balancer
- No Windows Clustering heartbeats via multicast
- No SQL Server AG multicast listeners
- No traditional NLB in multicast mode

WHAT DOES WORK

Unicast IPv4/IPv6, TCP, UDP, ICMP (for testing)

Subnets = Routing Policy Containers

An Azure subnet is **not** a Layer 2 broadcast domain. It's a logical grouping of NICs that share the same:

1. **Routing policy** (Route Tables / UDRs)
2. **Security policy** (NSGs at subnet level)
3. **Service Endpoints** (PaaS connectivity)
4. **Delegations** (Azure service integrations)

DESIGN IMPLICATION

Put resources that need **different routing or security policies** in different subnets - not based on "network segments".

⚠ COMMON MISTAKE

Creating "DMZ" subnets thinking they provide L2 isolation. There's no L2 in Azure - only policy enforced at the NIC.

MYTH #3

"ARP and DHCP Work Like Normal"

The fabric intercepts and synthesizes responses

ARP in Azure: It's All a Lie

ARP requests in Azure don't reach other VMs. The **VFP intercepts them** and synthesizes responses.

```
$ arp -a (from any Azure VM)

10.0.0.1     12:34:56:78:9a:bc
10.0.0.5     12:34:56:78:9a:bc
10.0.0.10    12:34:56:78:9a:bc
```

Every host returns the same MAC!

The synthetic MAC 12:34:56:78:9a:bc is the Azure Virtual Router. All traffic goes to this "router" which forwards based on IP, not MAC.

HOW IT ACTUALLY WORKS

- VM sends ARP request
- VFP intercepts at vSwitch
- Network Controller lookups destination
- VFP injects synthetic ARP response
- Packet forwarded based on IP routing policy

IMPLICATION: MAC addresses are meaningless in Azure. Don't design around them.

DHCP: Controlled by the Fabric

DHCP in Azure is provided by the host, not a DHCP server you deploy. The IP address is determined by your Azure configuration.

⚠ NEVER DO THIS

- Disable DHCP on VM NICs
- Manually assign static IPs inside the guest OS
- Run your own DHCP server for Azure VMs

CORRECT APPROACH

Set "static" IPs in Azure portal/ARM - DHCP still delivers them, but they don't change.

RESERVED UDP PORTS

67	DHCP Server
68	DHCP Client
4789	VXLAN

Host services available via 168.63.129.16: DHCP, DNS, health probes, IMDS

MYTH #4

"Routing Works Like On-Premises"

No recursive lookups. Routing happens at the NIC.

Routing Happens at the NIC, Not a Router

Azure NICs have three built-in capabilities. There is **no central router** - each NIC makes its own routing decisions.

Routing

Effective routes determine where packets go

Filtering

NSGs enforced at the NIC level

NAT

Public IP translation at the NIC

NO RECURSIVE ROUTING

Unlike physical routers, Azure doesn't do recursive lookups. If the next-hop isn't directly reachable, the route fails.

```
Effective routes example:

10.0.0.0/16  → VirtualNetwork
0.0.0.0/0    → Internet
10.1.0.0/16  → VNetPeering
```

Check effective routes on any NIC in the portal under "Effective routes" to see what Azure will actually do.

Route Selection: The Priority Order

PRIORITY ORDER (highest first)

1. **User-Defined Routes (UDRs)**
   Your custom routes in Route Tables

2. **BGP Routes**
   From VPN Gateway / ExpressRoute

3. **System Routes**
   Azure's automatic routes for VNet, Internet

⚠ EXCEPTION: Service Traffic

System routes for VNet peering, Service Endpoints, and VNet service traffic **always win** over BGP routes - even with longer prefix match.

LONGEST PREFIX MATCH

Within the same priority tier, the most specific route (longest prefix) wins.

`10.0.1.0/24 beats 10.0.0.0/16`

MYTH #5

"NSGs are Simple Stateless ACLs"

Did you know they’re actually statefull?!

NSGs Are Stateful Firewalls

Unlike AWS NACLs, Azure NSGs **track connection state**. Return traffic is automatically allowed.

EXAMPLE: Web Server on Port 443

✓ Allow inbound TCP/443 from Internet

⊘ No outbound rule needed for responses

The flow record tracks the connection and auto-permits return traffic.

GOTCHA: Existing Connections

Adding a deny rule doesn't kill existing flows. New connections blocked, old ones continue until timeout.

NSG PROCESSING ORDER

INBOUND TRAFFIC

Subnet NSG → NIC NSG

Both must allow for traffic to pass

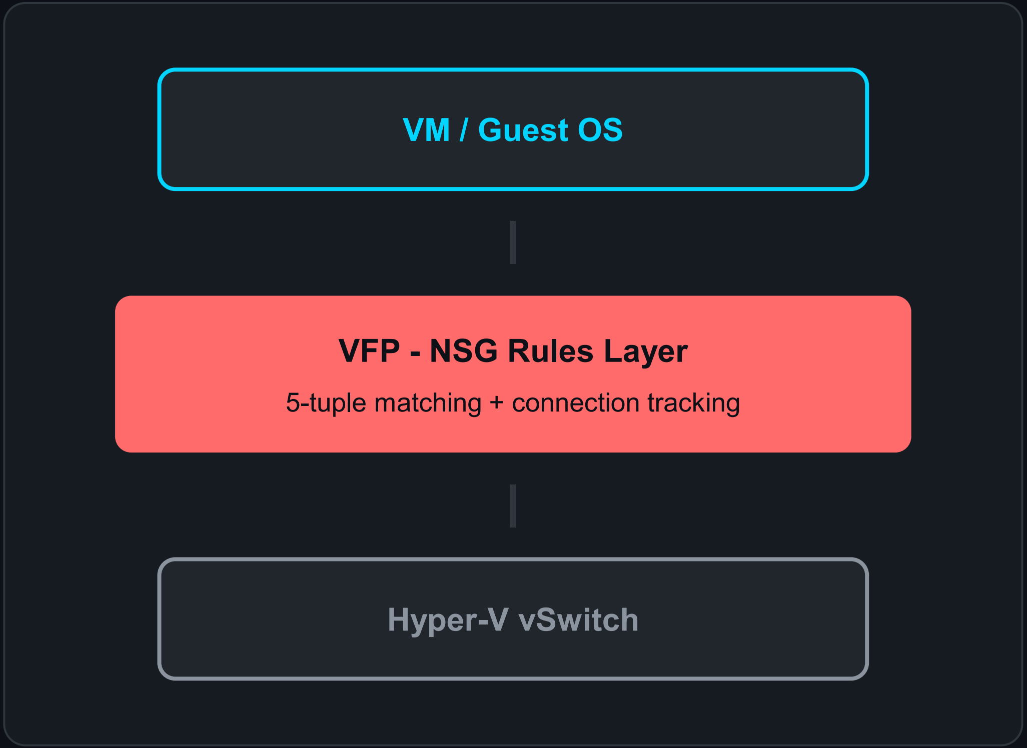
OUTBOUND TRAFFIC

NIC NSG → Subnet NSG

Order is reversed from inbound

Best practice: Use subnet-level NSGs for shared policy, NIC-level for specific overrides.

Where NSGs Actually Execute



NSG rules are programmed into the **VFP layer** of the Hyper-V vSwitch on each host. They're not running in a separate appliance.

5-Tuple Matching

Source IP, Source Port, Dest IP, Dest Port, Protocol

Connection State Tracking

Flow records persist for 4+ minutes by default

No Performance Penalty

Hardware offload via SmartNICs when using Accelerated Networking

DEEP DIVE

The Virtual Filtering Platform

Established: September 1, 2011

VFP: A Match-Action Flow Engine

VFP processes packets through a series of **Layers**, each containing **Groups** of **Rules**.

PROCESSING MODEL

1. **Match** - Classify packet against rules
2. **Action** - Transform, forward, or drop
3. **Cache** - Create Unified Flow for fast path

First packet takes the slow path (full rule evaluation). Subsequent packets in the same flow hit the cached Unified Flow - microsecond latency.

VFP OBJECT HIERARCHY

```
Port (vNIC)
  └─ Layer (ACL, NAT, Encap...)
    └─ Group (Rule container)
      └─ Rule (Match + Action)
```

KEY INSIGHT

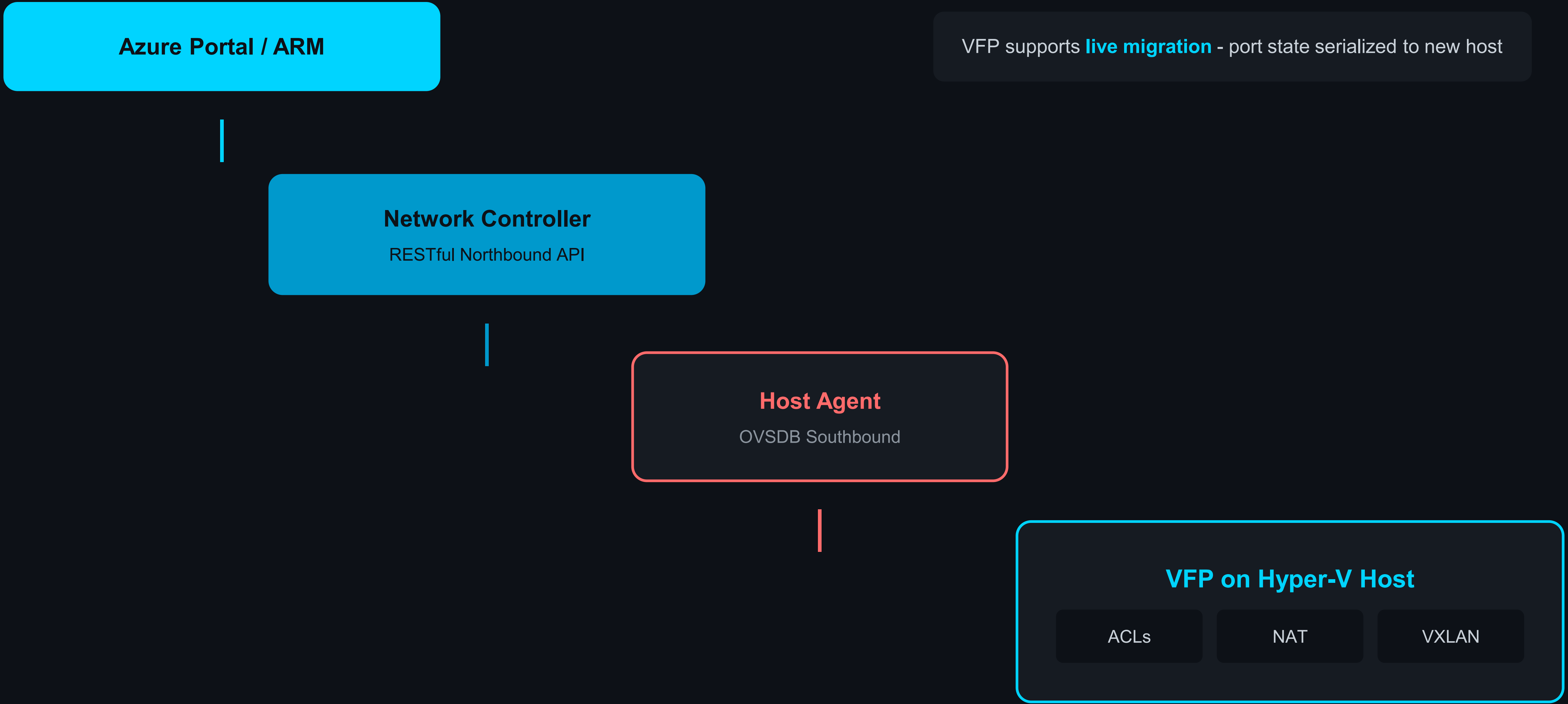
Inbound and outbound packets traverse layers in **opposite order**. This creates symmetric policy enforcement without duplicate rules.

Source: Microsoft Research - VFP NSDI 2017

Control Plane: How Policy Reaches VFP

Policy is **declarative** - you define intent, controller figures out rules

VFP supports **live migration** - port state serialized to new host



Azure Portal / ARM

Network Controller

RESTful Northbound API

Host Agent

OVSDB Southbound

VFP on Hyper-V Host

ACLs

NAT

VXLAN

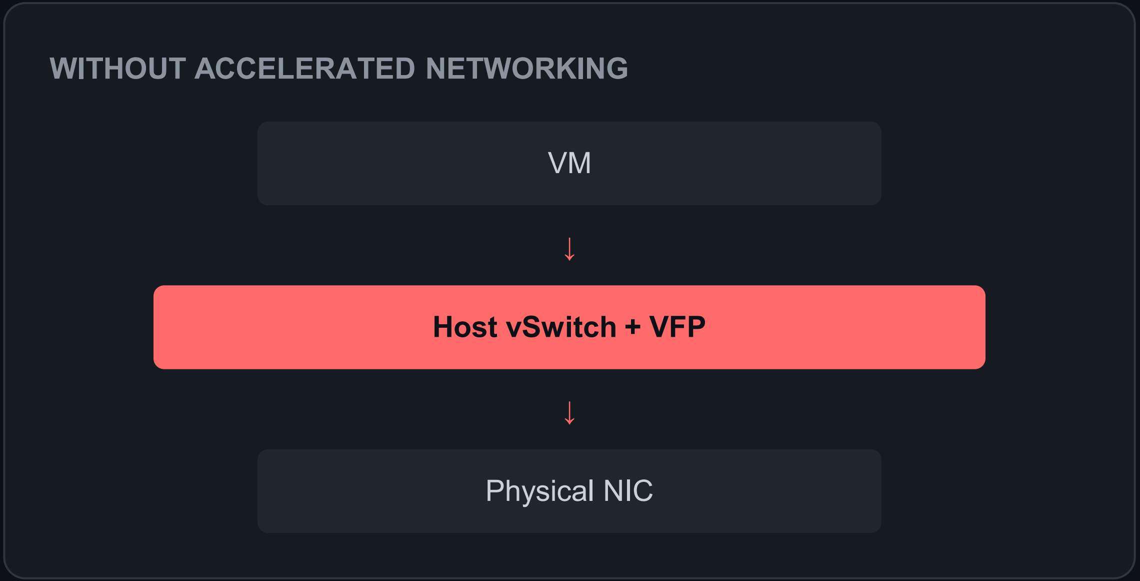
PERFORMANCE

Accelerated Networking & SR-IOV

Team Rocket blast off at the speed of light!

SR-IOV: Bypassing the Host

Single Root I/O Virtualization (SR-IOV) allows VMs to talk directly to the physical NIC, bypassing the host's vSwitch.



WITHOUT ACCELERATED NETWORKING

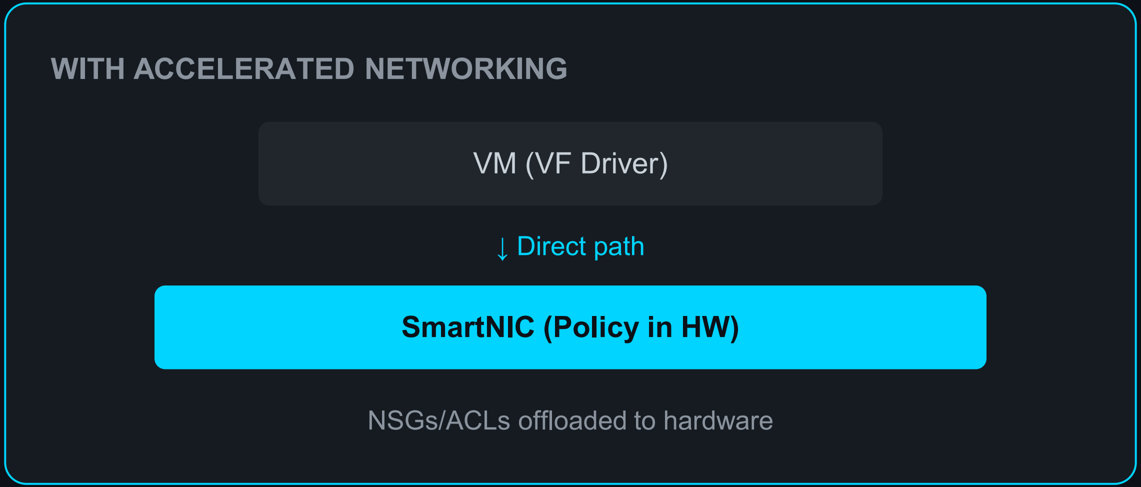
VM

↓

Host vSwitch + VFP

↓

Physical NIC



WITH ACCELERATED NETWORKING

VM (VF Driver)

↓ Direct path

SmartNIC (Policy in HW)

NSGs/ACLs offloaded to hardware

< 25µs
Latency

30M+
PPS

~0%
CPU overhead

Azure SmartNICs: The Secret Weapon

Azure uses FPGA-based SmartNICs to offload VFP policies to hardware.
This isn't just SR-IOV - it's **programmable hardware acceleration**.

Mellanox ConnectX NICs with FPGA offload

VFP Unified Flows compiled to hardware tables

Full policy enforcement in NIC - not just fast path

FAILSAFE DESIGN

If VF fails, traffic automatically falls back to synthetic NIC through host vSwitch. No connectivity loss.

WHAT GETS OFFLOADED

- ✓ NSG / ACL rules
- ✓ VXLAN encap/decap
- ✓ NAT / Load balancer hairpin
- ✓ Connection tracking
- ✓ QoS metering

THE DEEPER TRUTH

Packets Don't Traverse Networks—They Get Dropped Into Memory

How Azure’s architecture makes networking an illusion

"Wormhole" Networking: Source to Destination

WHAT YOU EXPECT

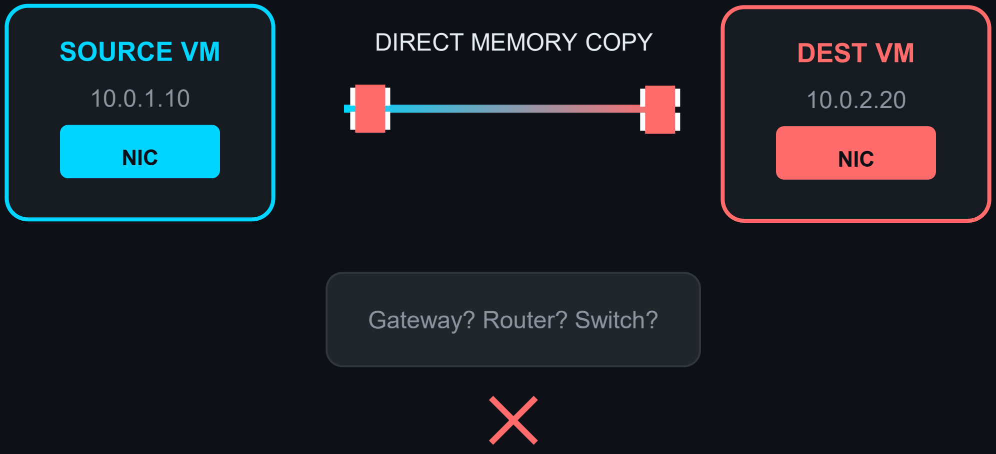
VM → Virtual Switch → Virtual Network → Subnet → Default Gateway → Routing → Destination NIC

WHAT ACTUALLY HAPPENS

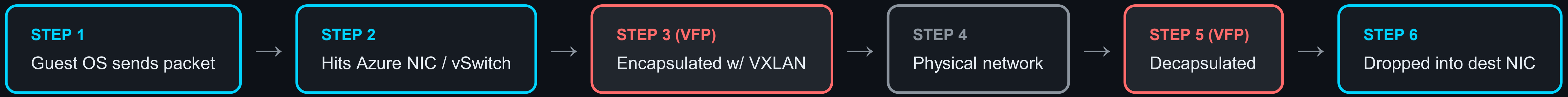
Packet leaves source NIC → **appears directly** at destination NIC. No hops, no routers, no gateways.

> *"Wormhole area networking is the best way of describing it. Packet went in, came out the other end."*

This is why **traceroute is pointless** in Azure and why the default gateway doesn't respond to ping.



The Real Packet Journey: Encapsulation & Memory



STEP 1 Guest OS sends packet →

STEP 2 Hits Azure NIC / vSwitch →

STEP 3 (VFP) Encapsulated w/ VXLAN →

STEP 4 Physical network →

STEP 5 (VFP) Decapsulated →

STEP 6 Dropped into dest NIC

SAME HOST = NO NETWORK

If source and destination VMs are on the same Hyper-V host, the packet **never touches the physical network**. It's a pure memory-to-memory copy via the vSwitch.

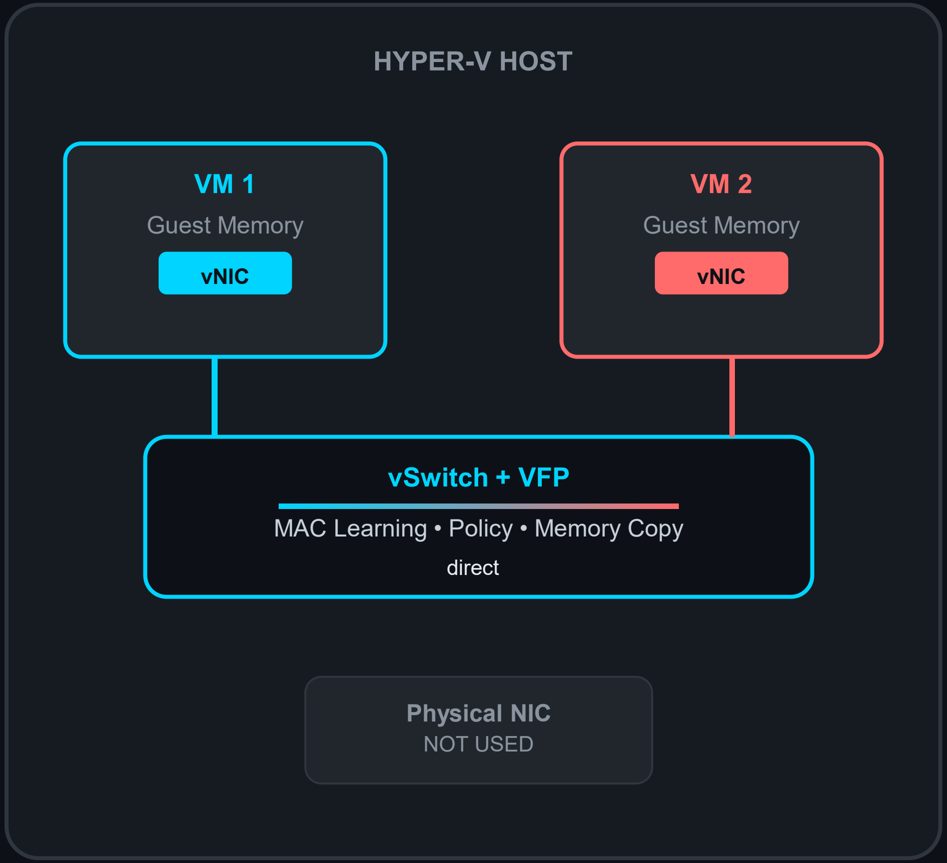
DIFFERENT HOST = TUNNEL

When hosts differ, the packet is encapsulated with **physical addresses** (PA) and tunneled. The customer address (CA) is hidden from the physical network.

> "The packet reaches the source NIC, hits the virtual switch and is encapsulated. The physical network could be in the same rack, a neighboring data center, or somewhere around the world—but from our perspective, that doesn't matter. All we see is packets going directly from source to destination."
>
> — Aidan Finn, D2DO273: Azure VNets Don't Exist

KEY INSIGHT: The "network" you design exists only as metadata. The physical layer is invisible to you.

Same-Host VMs: Pure Memory Copy



WHEN VMs SHARE A HOST

The Hyper-V vSwitch knows both VMs are local. It reads the destination MAC, finds the target vNIC in its table, and **copies the packet directly to the destination VM's memory**.

NO PHYSICAL NETWORK TRAVERSAL

- No encapsulation needed
- No VXLAN overhead
- No physical NIC involvement
- Sub-microsecond latency

> **Cross-host and external flows** leverage encapsulation; **same-host flows are direct via the vSwitch**.
>
> — Azure Local SDN Documentation

VFP still enforces ACLs, QoS, and monitoring… But the packet never leaves host memory. This is why VM placement can affect network performance.

What This Means for Your Architecture

BECAUSE PACKETS GO DIRECT...

Subnets ≠ Isolation

Different subnets in the same VNet route directly. Subnets don't segment—NSGs do.

No Default Gateway Hop

The default gateway doesn't process traffic—it's a fiction for your OS. UDRs override direct routing.

VNet Peering = Merged Sets

When you peer VNets, no cable is laid. The fabric just expands which NICs can reach each other directly.

YOU MUST DESIGN DIFFERENTLY...

Use Smaller VNets

One workload per VNet. Peer to hub. Traffic isolation via the hub firewall—not giant monolithic VNets.

Routing = Your "Cabling"

On-prem cables force traffic paths. In Azure, **UDRs** force traffic through your firewall—that's your security model.

NSG = Your Segmentation

Micro-segment with NSGs + ASGs. Block everything by default (custom deny-all rule), then allow only what's needed.

> "There is no cabling in the fabric. Packets are encapsulated on the source host, transmitted over the physical network, decapsulated on the destination host, and presented to the destination VM's NIC. In short, packets leave the source NIC and **magically appear** on the destination NIC with no hops in between."

REMEMBER: If you're thinking in cables and switches, you're designing for a network that doesn't exist.

Wrapping it up

Is your brain fried? Mine sure is…

Design Implications: What This Means For You

STOP DOING

- Designing based on L2 broadcast domain assumptions
- Expecting multicast/broadcast for clustering/HA
- Manually assigning IPs in guest OS
- Relying on MAC addresses for anything
- Creating complex NSG outbound rules for allowed inbound

START DOING

- Design subnets around policy requirements, not "segments"
- Use Azure Load Balancer for HA instead of VRRP/HSRP
- Enable Accelerated Networking on all supported VMs
- Check effective routes when troubleshooting
- Use NSG flow logs for visibility into traffic patterns

Troubleshooting With This Knowledge

1. Check Effective Routes

Portal → NIC → Effective Routes. This shows what Azure *actually* does, not what you configured.

2. Network Watcher Next Hop

Test where traffic from a specific source would go. Accounts for UDRs, system routes, and BGP.

3. NSG Flow Logs

See exactly which NSG rule allowed/denied traffic. Records 5-tuple + decision.

4. IP Flow Verify

Test if NSGs would allow/deny a specific flow without sending real traffic.

REMEMBER

Can't ping the default gateway (168.63.129.16 doesn't respond to ICMP) - this is by design, not a problem.

```
Useful commands:

arp -a
```

All MACs = 12:34:56:78:9a:bc

```
route print (Windows)
```

Shows routes from guest perspective

Key Takeaways

1 **VNets are policy, not infrastructure**

They define which NICs can route to each other - there's no physical network construct.

2 **There is no Layer 2 in Azure**

No broadcast, no multicast, no real ARP. Everything is routed via VXLAN encapsulation.

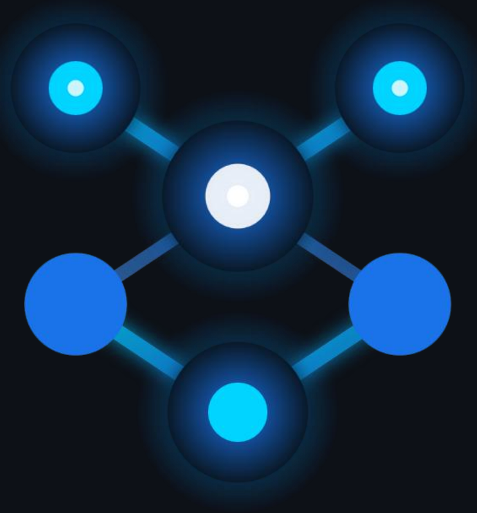
3 **VFP is the brain of Azure networking**

All policies (NSGs, NAT, routing) are programmed into VFP layers on each host's vSwitch.

4 **Accelerated Networking is a game-changer**

SR-IOV + SmartNICs deliver near-bare-metal performance with full policy enforcement in hardware.

Coming soon…

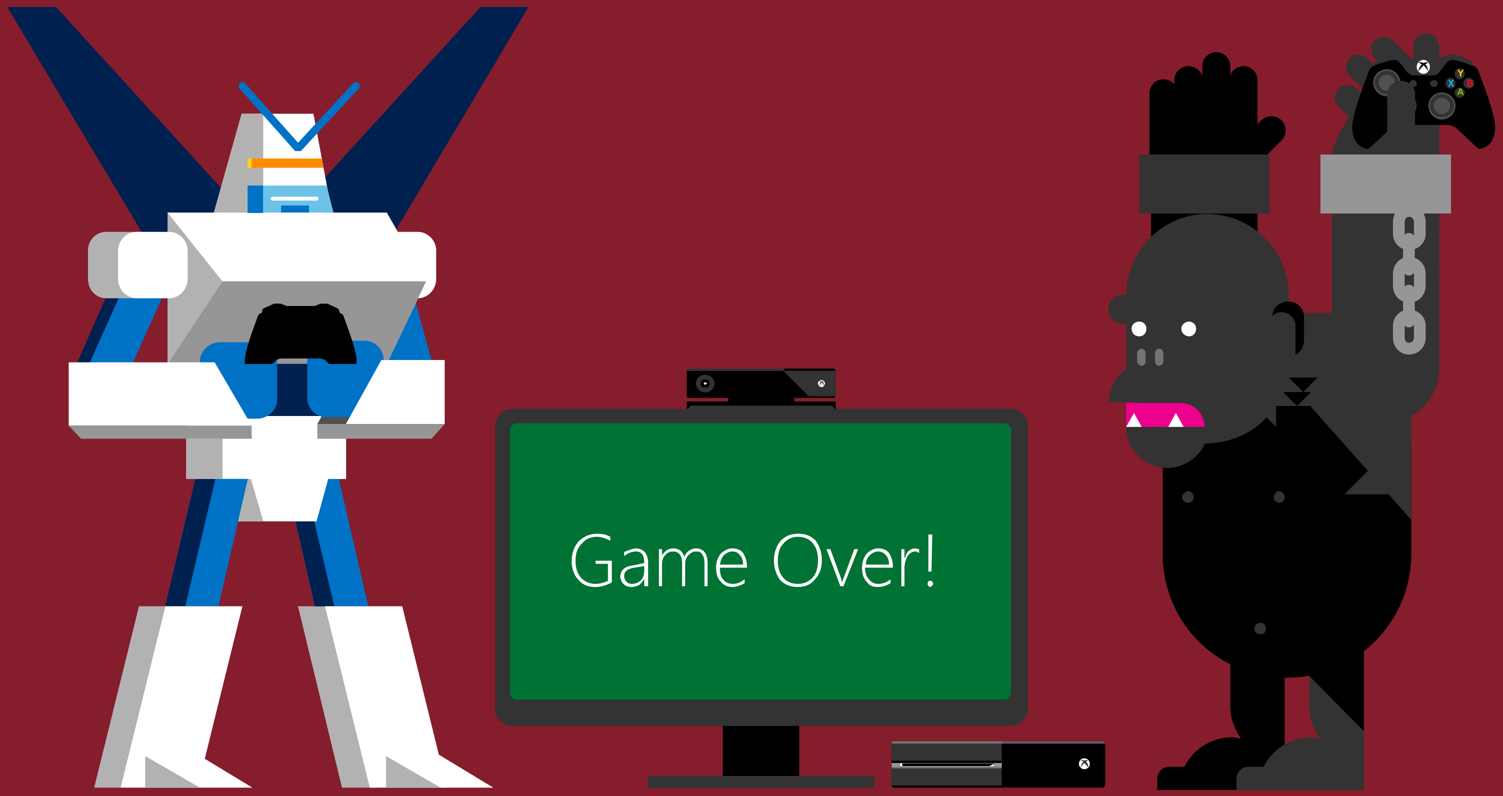


COMPOSABLE

TRUST

Secure and compliant by design, not by manual effort

Security. Compliance. Confidence.


Game Over!