Network Stack as a Service in the Cloud

Zhixiong Niu[1], Hong Xu[1], Dongsu Han[2], Peng Cheng[3], Yongqiang Xiong[3], Guo Chen[3], Keith Winstein[4]

[1]City University of Hong Kong
[2]KAIST
[3]Microsoft Research Asia
[4]Stanford University

Imagine you're a **tenant.** You want to deploy a new stack.

Motivation: Tenants

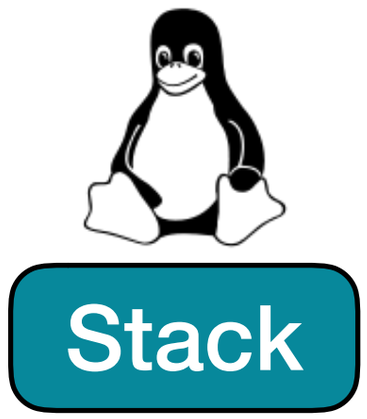
I heard that BBR is great. Let's deploy it to my VMs!

Motivation: Tenants

I heard that BBR is great. Let's deploy it to my VMs!

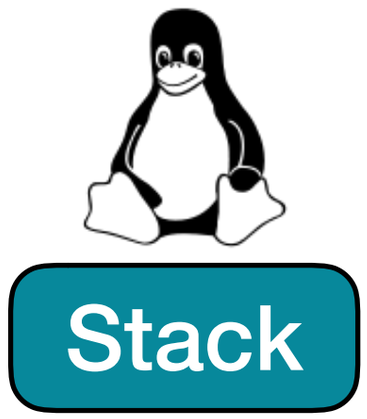
Motivation: Tenants

I heard that BBR is great. Let's deploy it to my VMs!



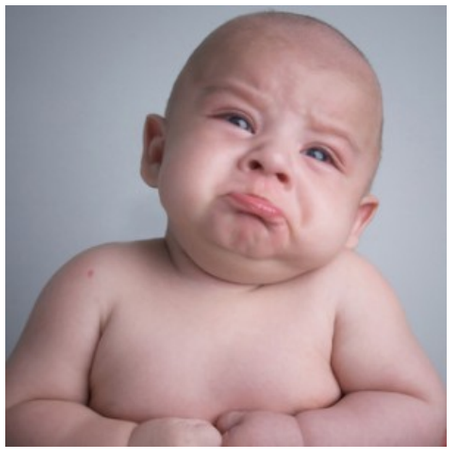
Motivation: Tenants

I heard that BBR is great. Let's deploy it to my VMs!

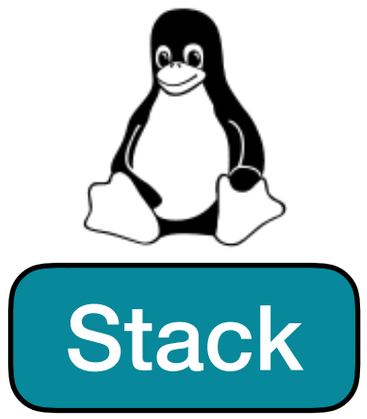


Problem: cannot deploy a stack across OSes

Motivation: Tenants

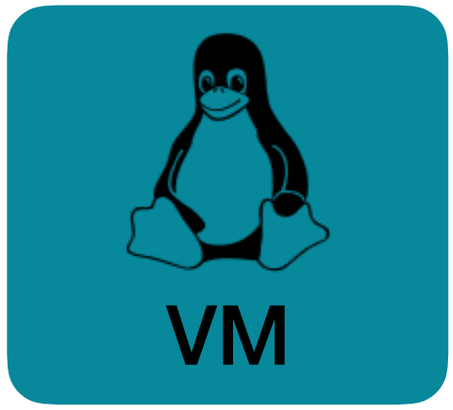


I heard that BBR is great. Let's deploy it to my VMs!



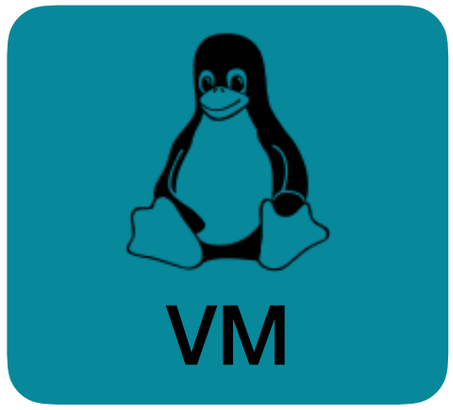
Problem: cannot deploy a stack across OSes

Motivation: Tenants

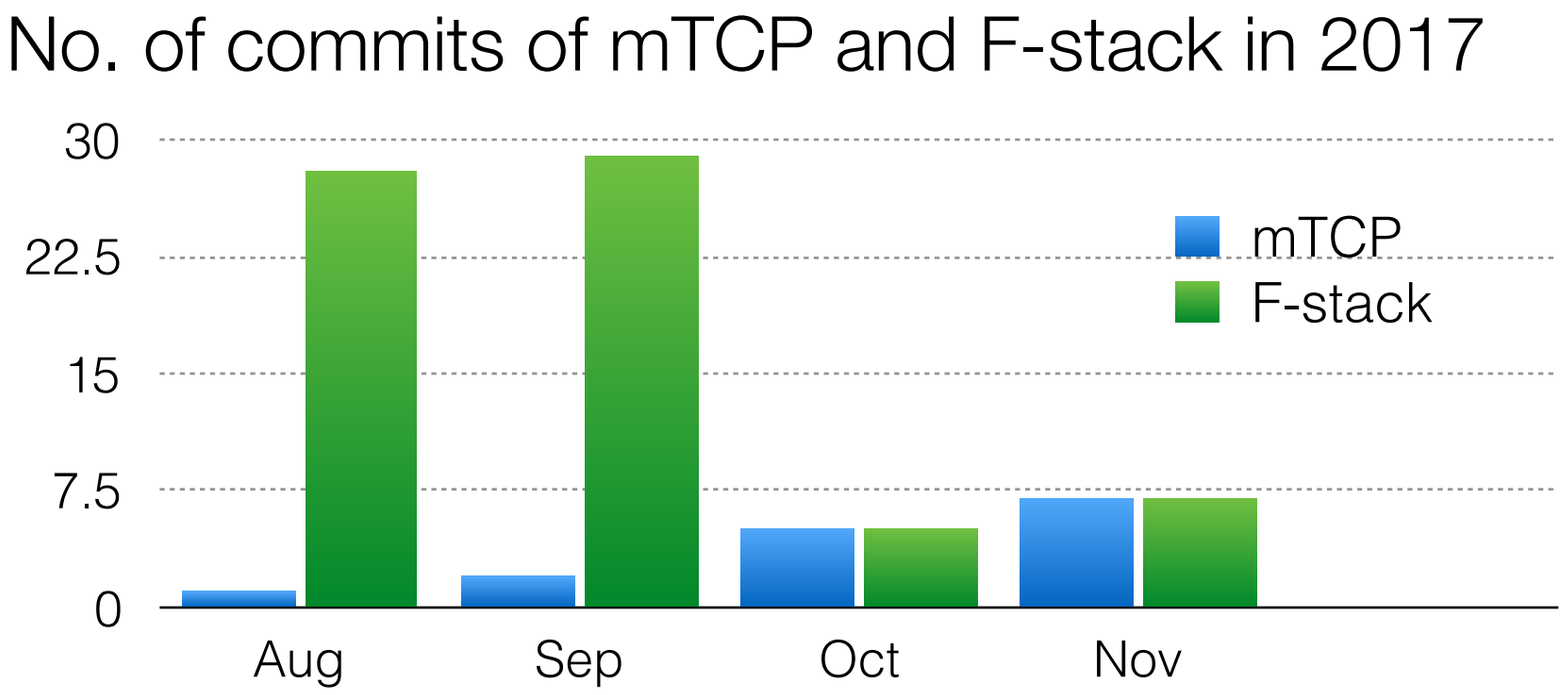


Stack

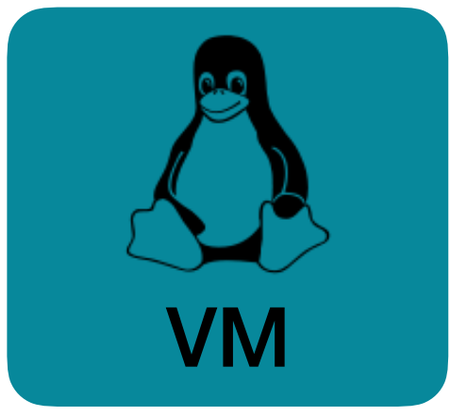
Motivation: Tenants



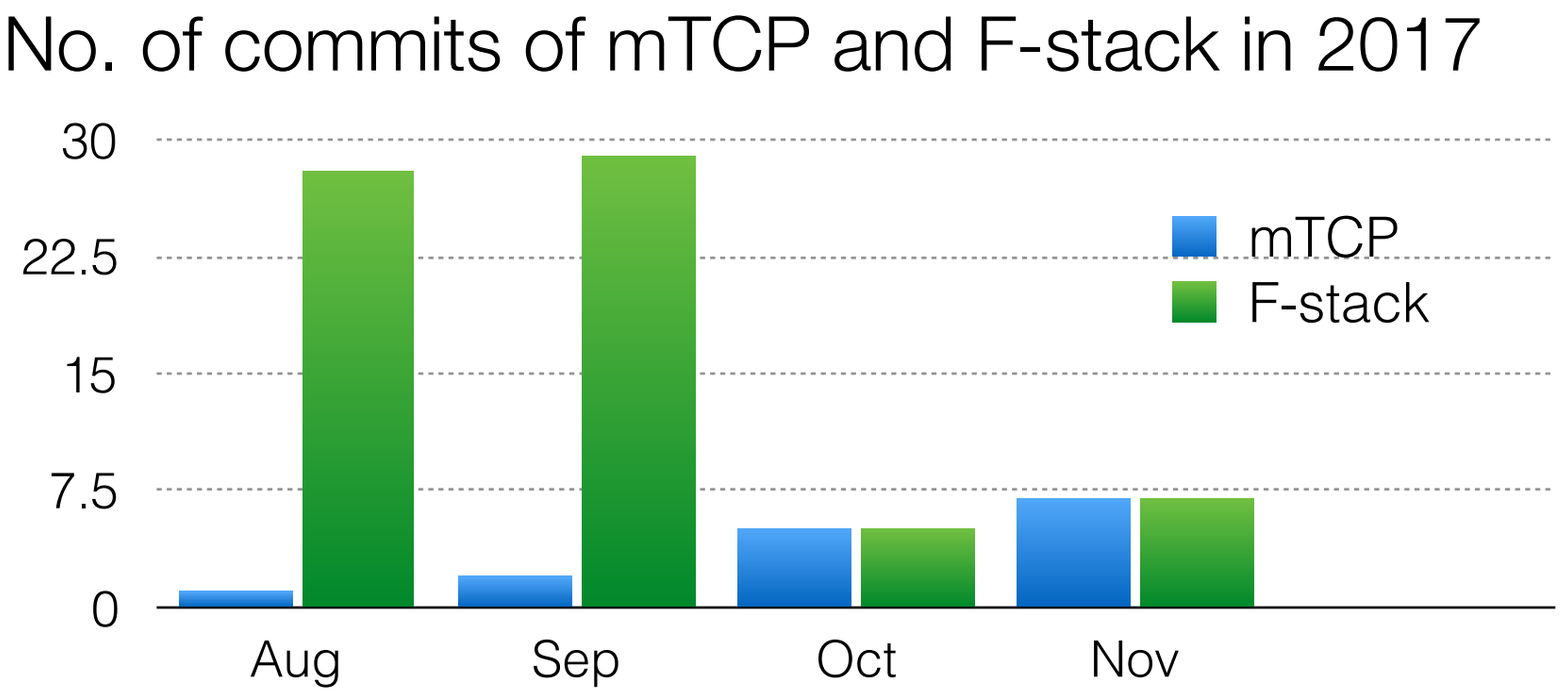
Stack



Motivation: Tenants

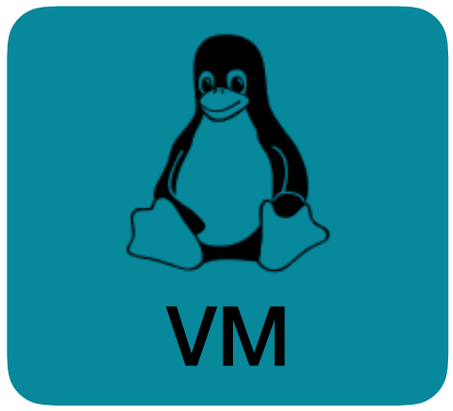


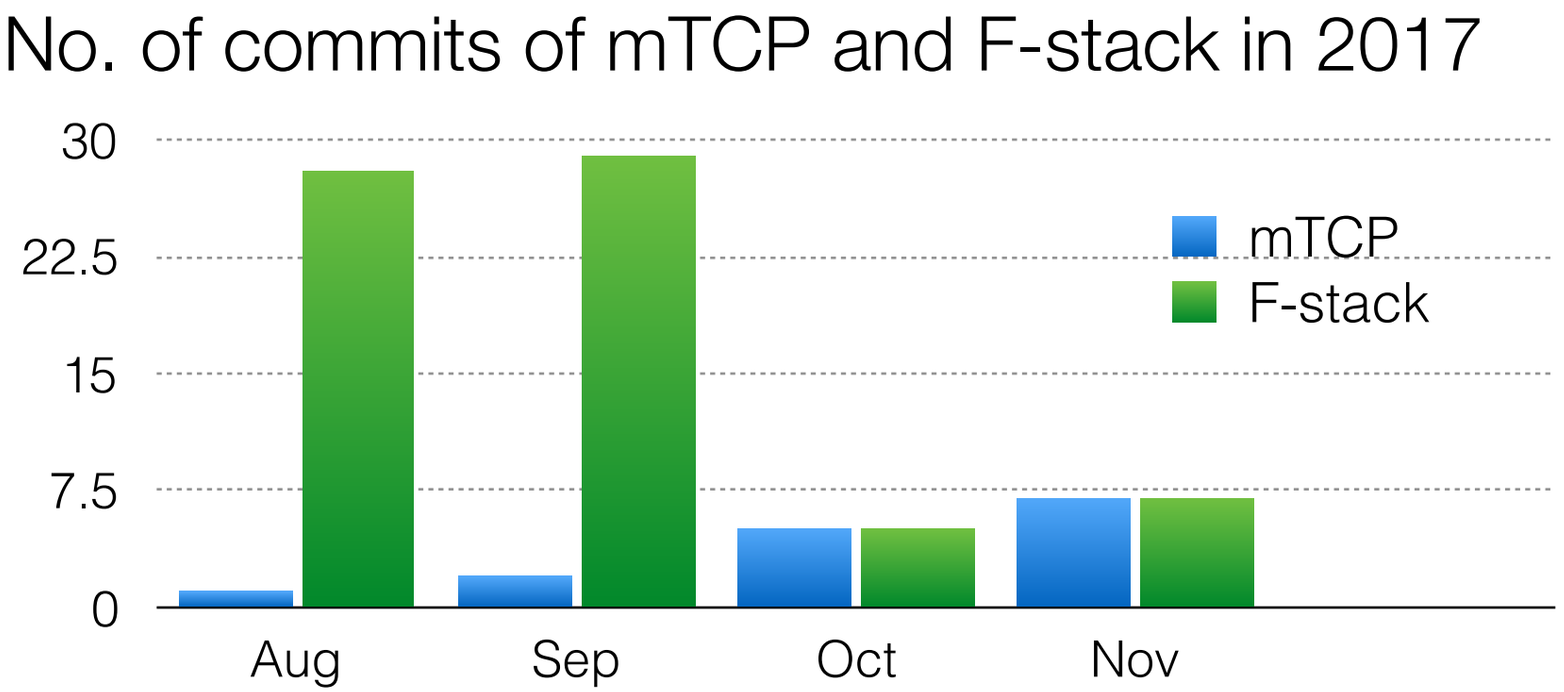
Stack



Problem: high deployment and maintenance cost

Motivation: Tenants



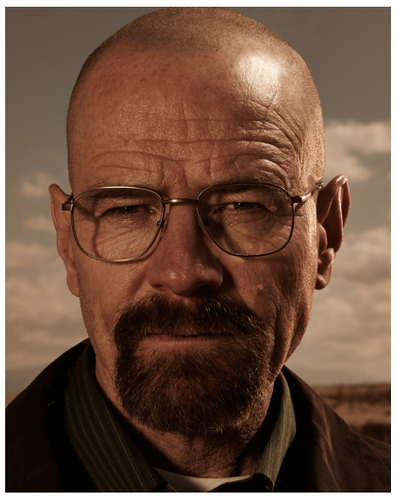


Problem: high deployment and maintenance cost

So your life as a tenant sucks.

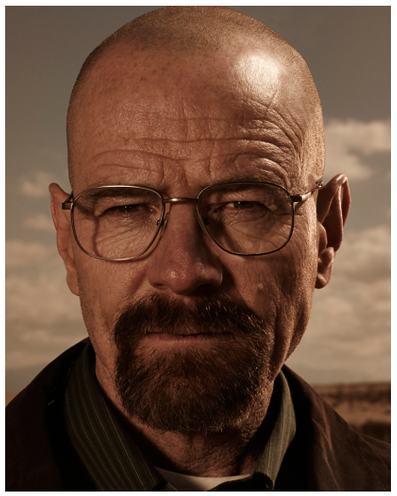
What about the **cloud** provider?

Motivation: Provider



I know that BBR is great.
Let me deploy it for my tenants!

Motivation: Provider

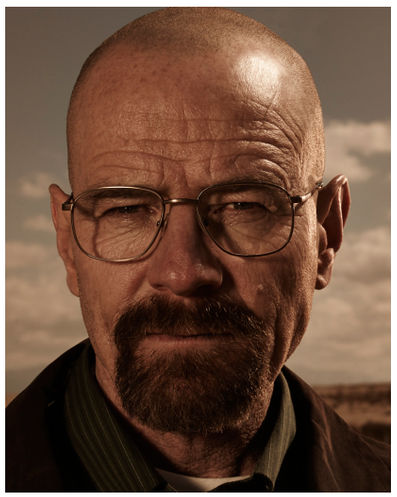


I know that BBR is great.
Let me deploy it for my tenants!

Tenant

Provider

Motivation: Provider



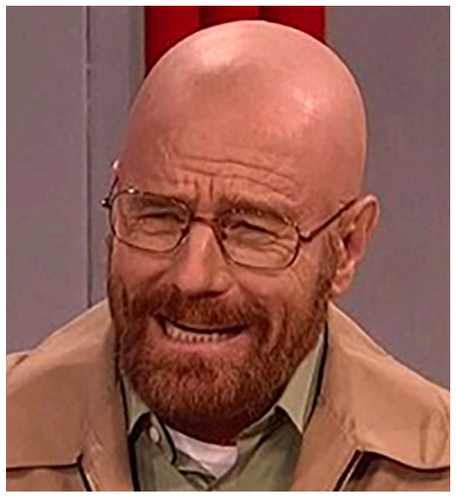
I know that BBR is great.
Let me deploy it for my tenants!

Tenant

Provider

Problem: can't touch the tenant stack

Motivation: Provider



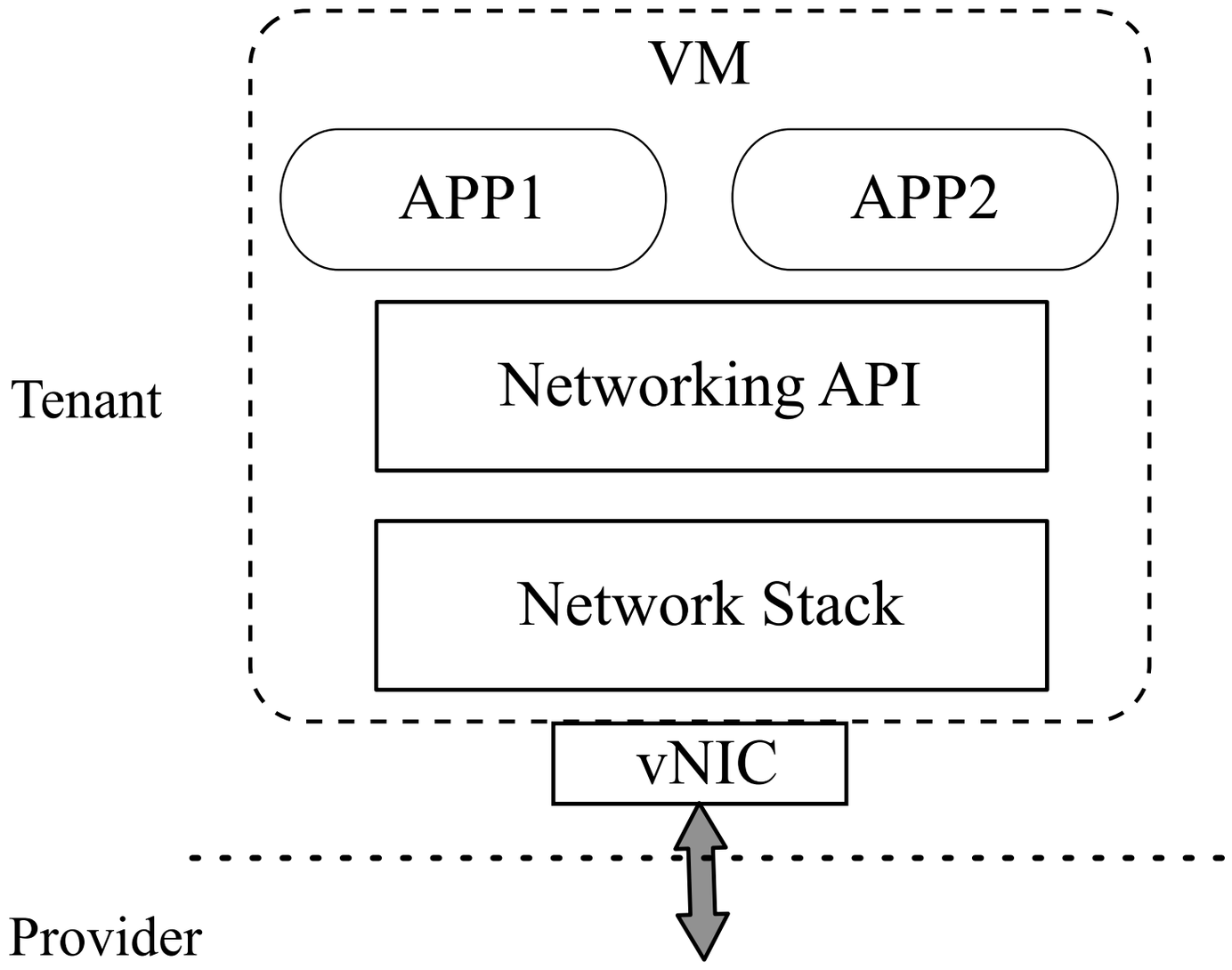
I know that BBR is great.
Let me deploy it for my tenants!

Tenant

Provider

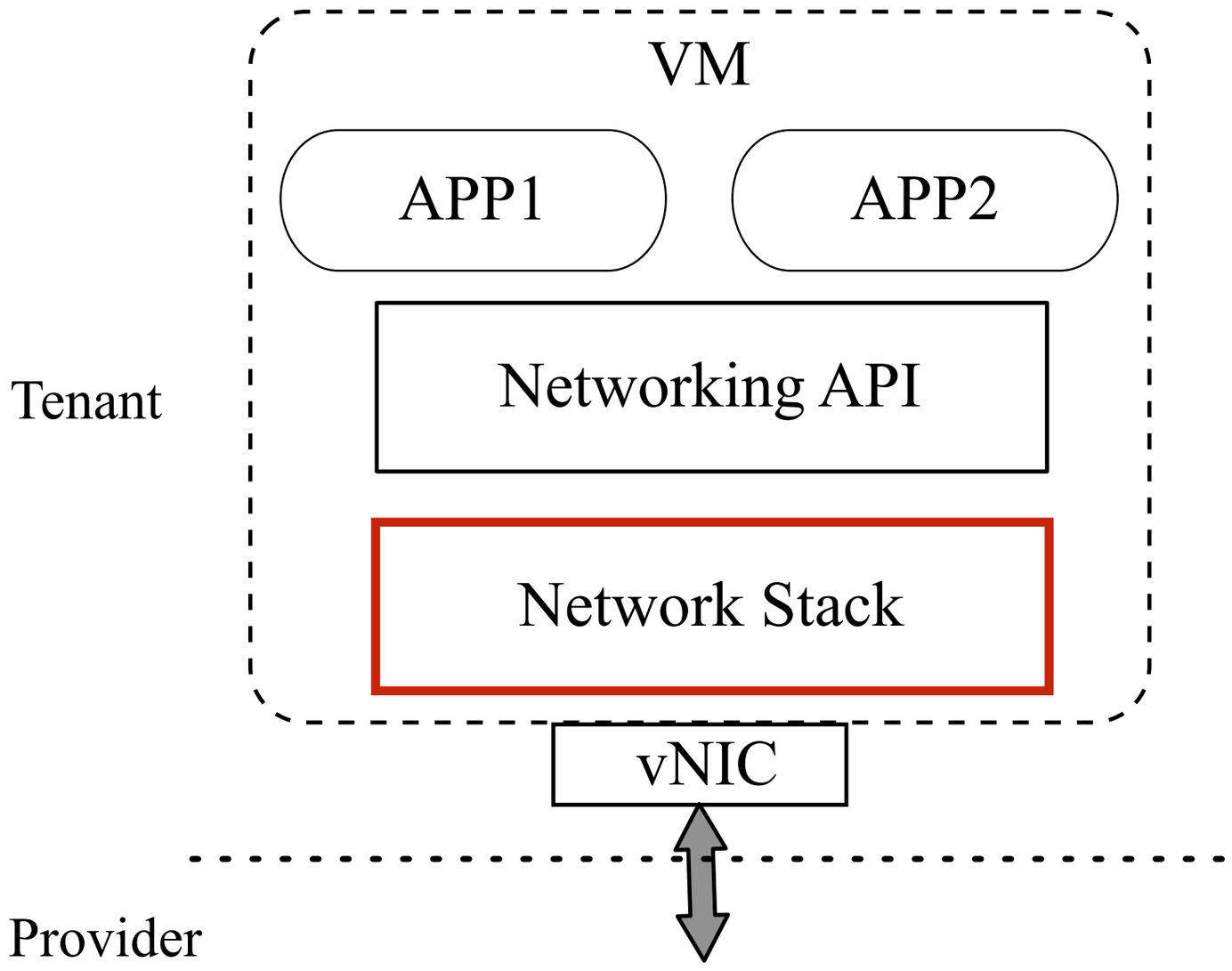
Problem: can't touch the tenant stack

So what’s wrong here?



Current architecture

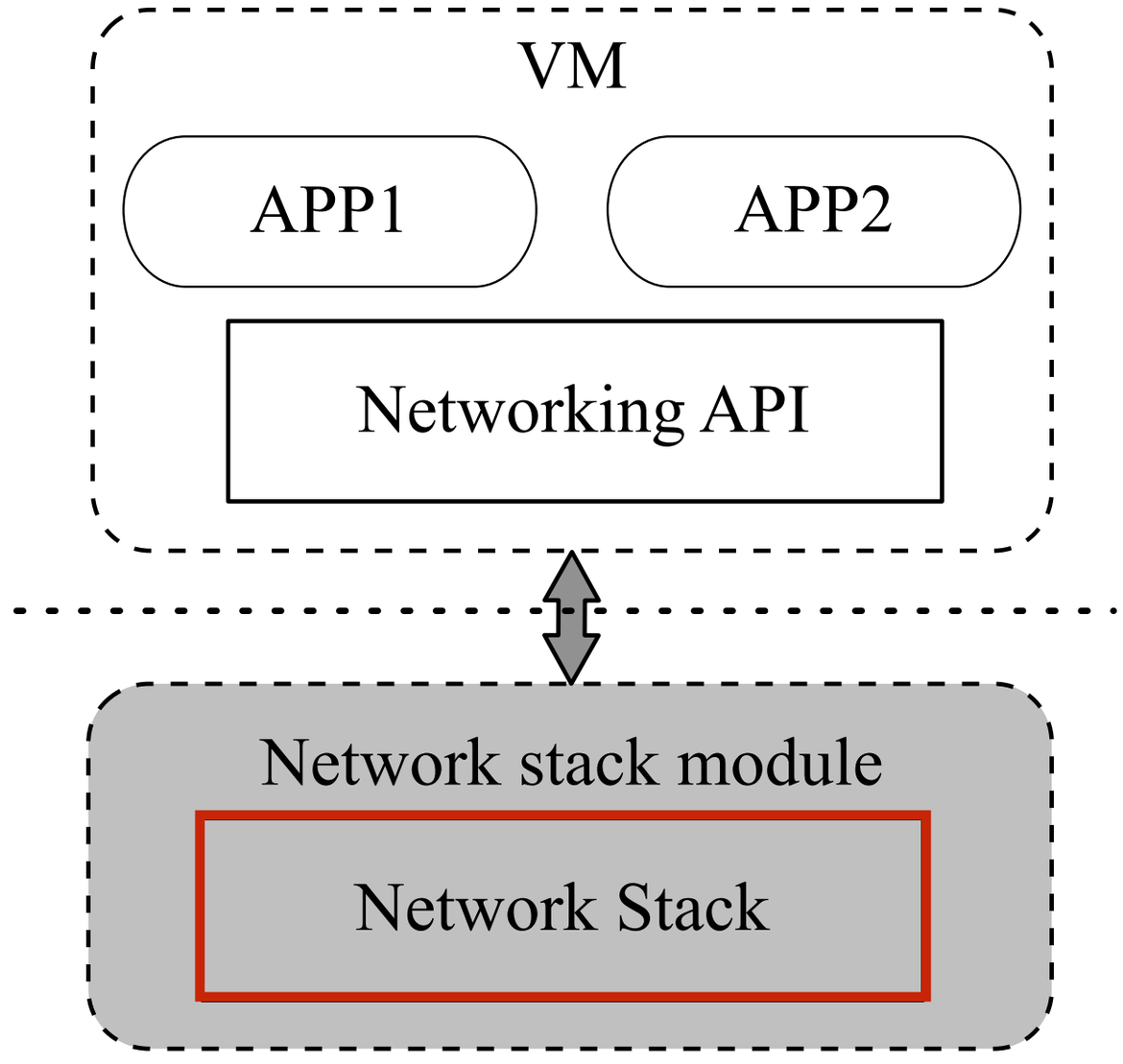
Network stack is coupled to the guest OS

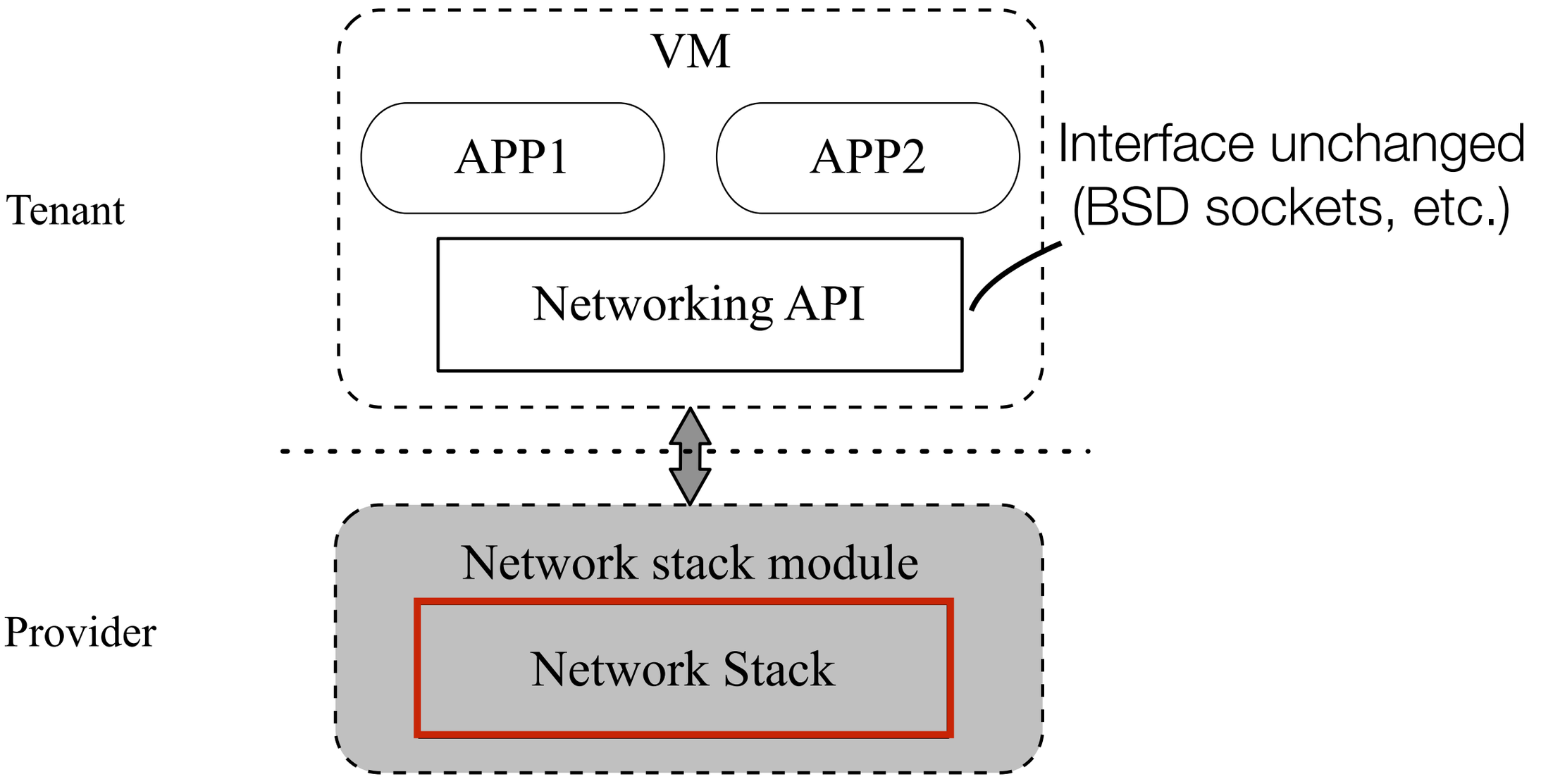


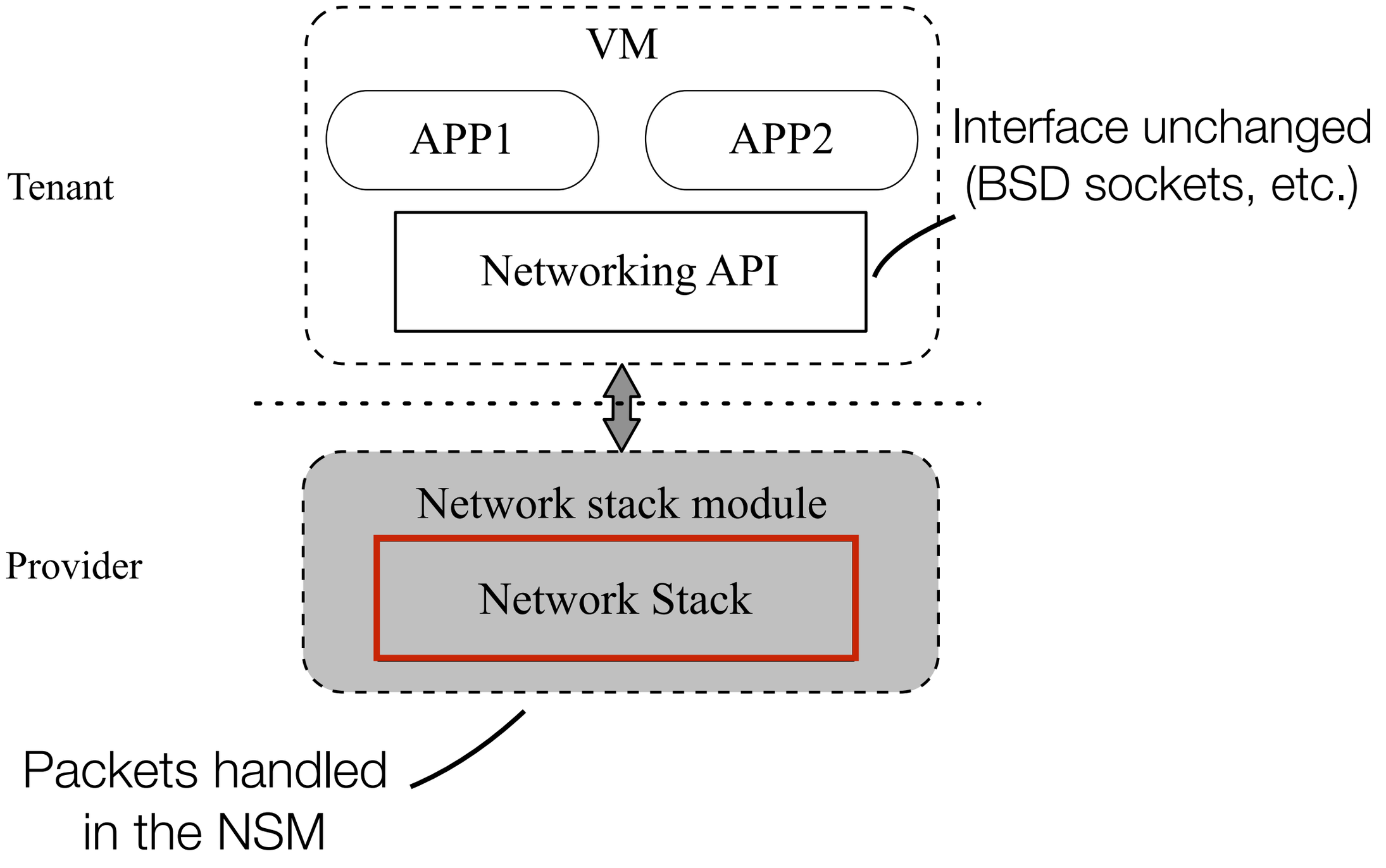
Current architecture

Tenant

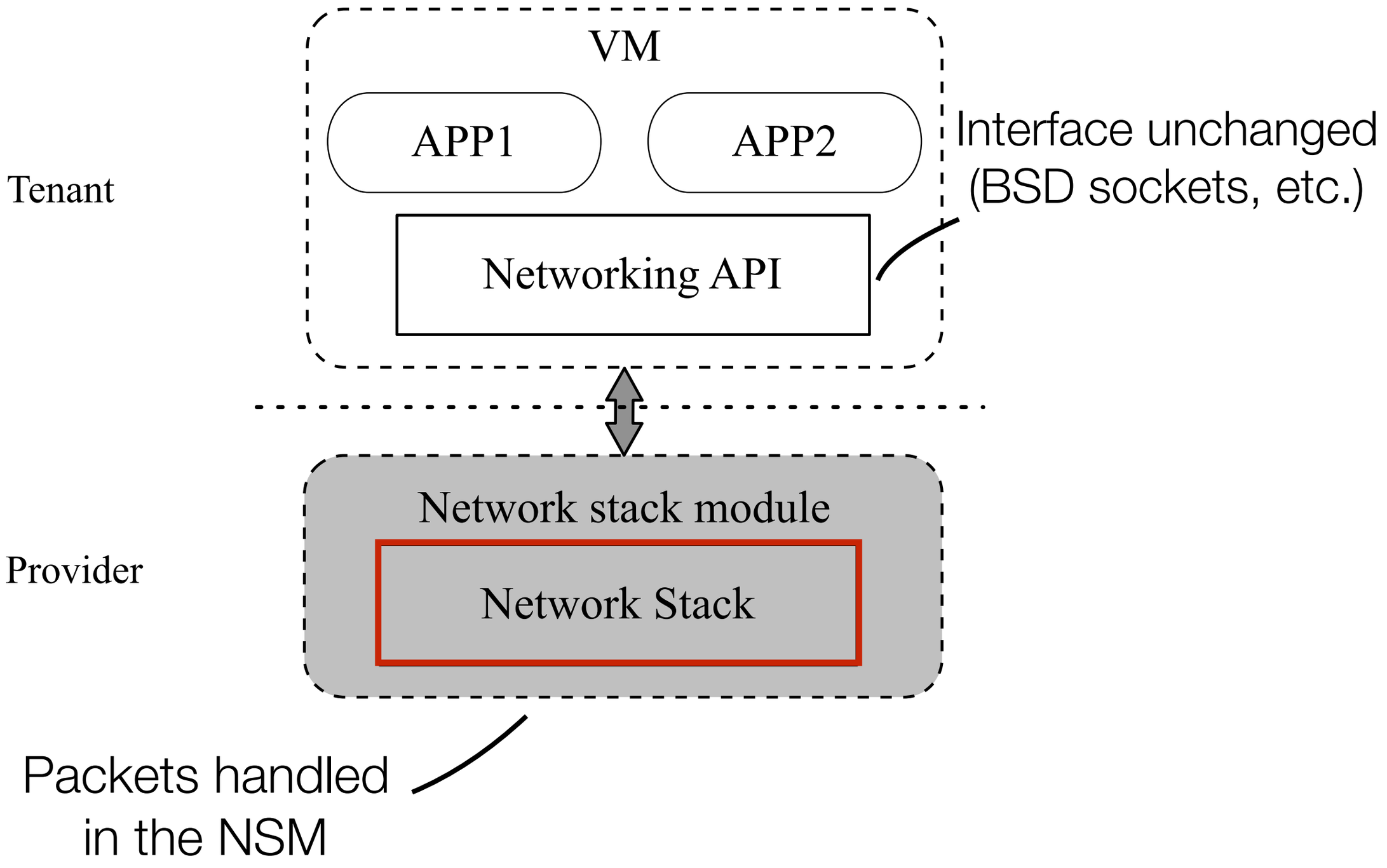
Provider




Tenant
Provider
VM
APP1
APP2
Networking API
Interface unchanged
(BSD sockets, etc.)
Network stack module
Network Stack

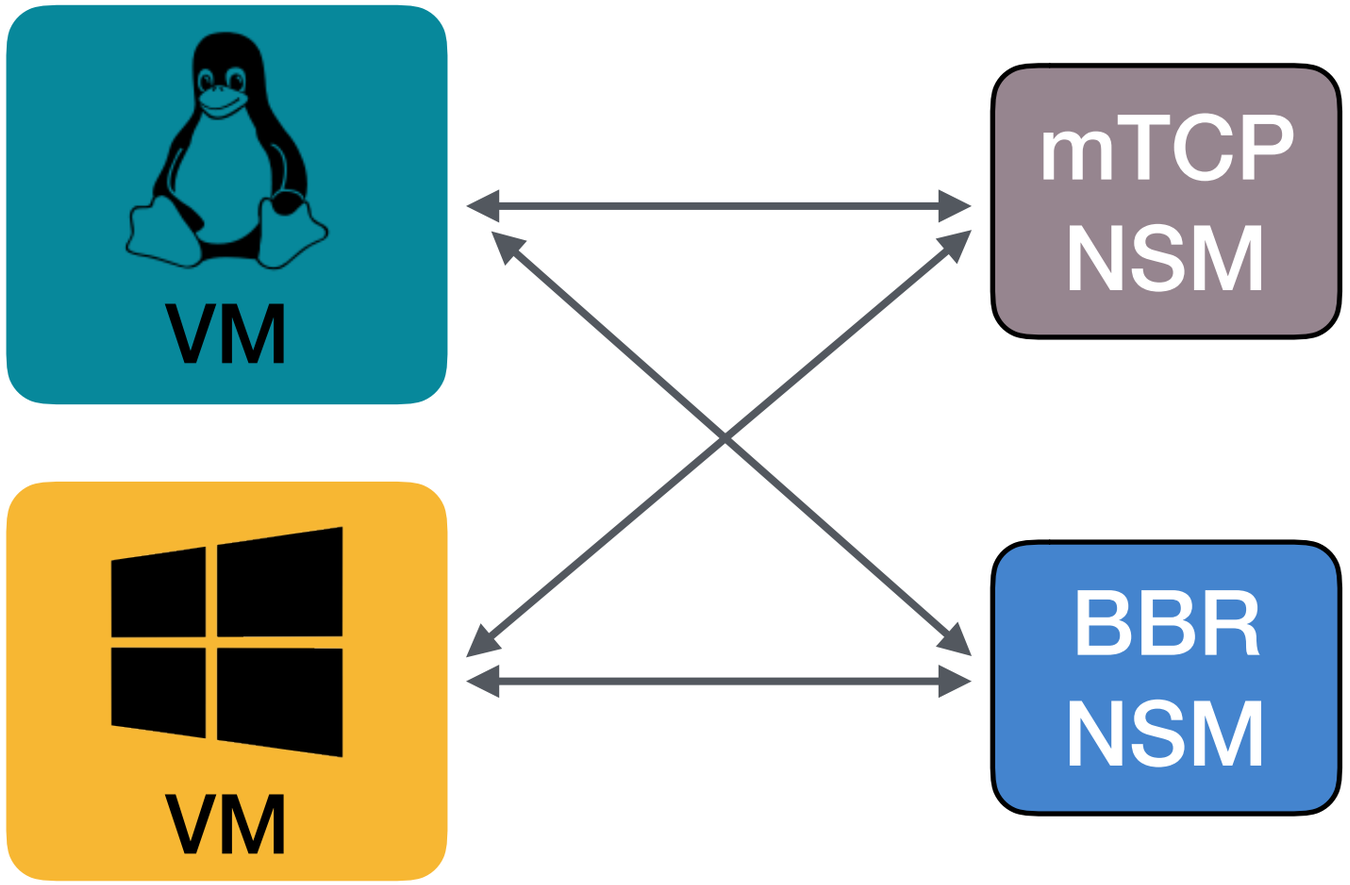

Tenant
VM
APP1
APP2
Networking API
Interface unchanged
(BSD sockets, etc.)
Provider
Network stack module
Network Stack
Packets handled
in the NSM

Vision: Network Stack as a Service

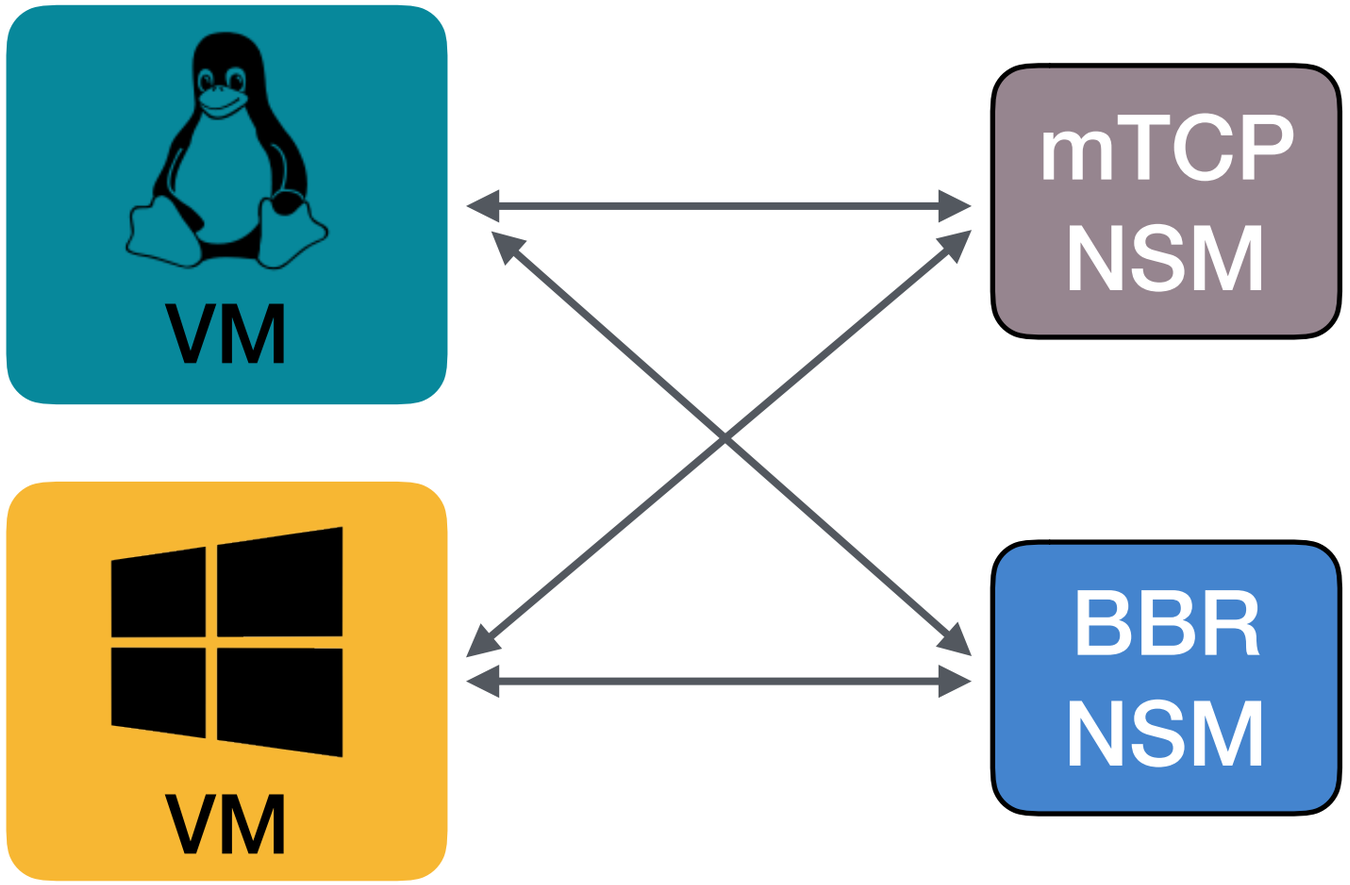


What’re the benefits?

Flexibility for Tenants

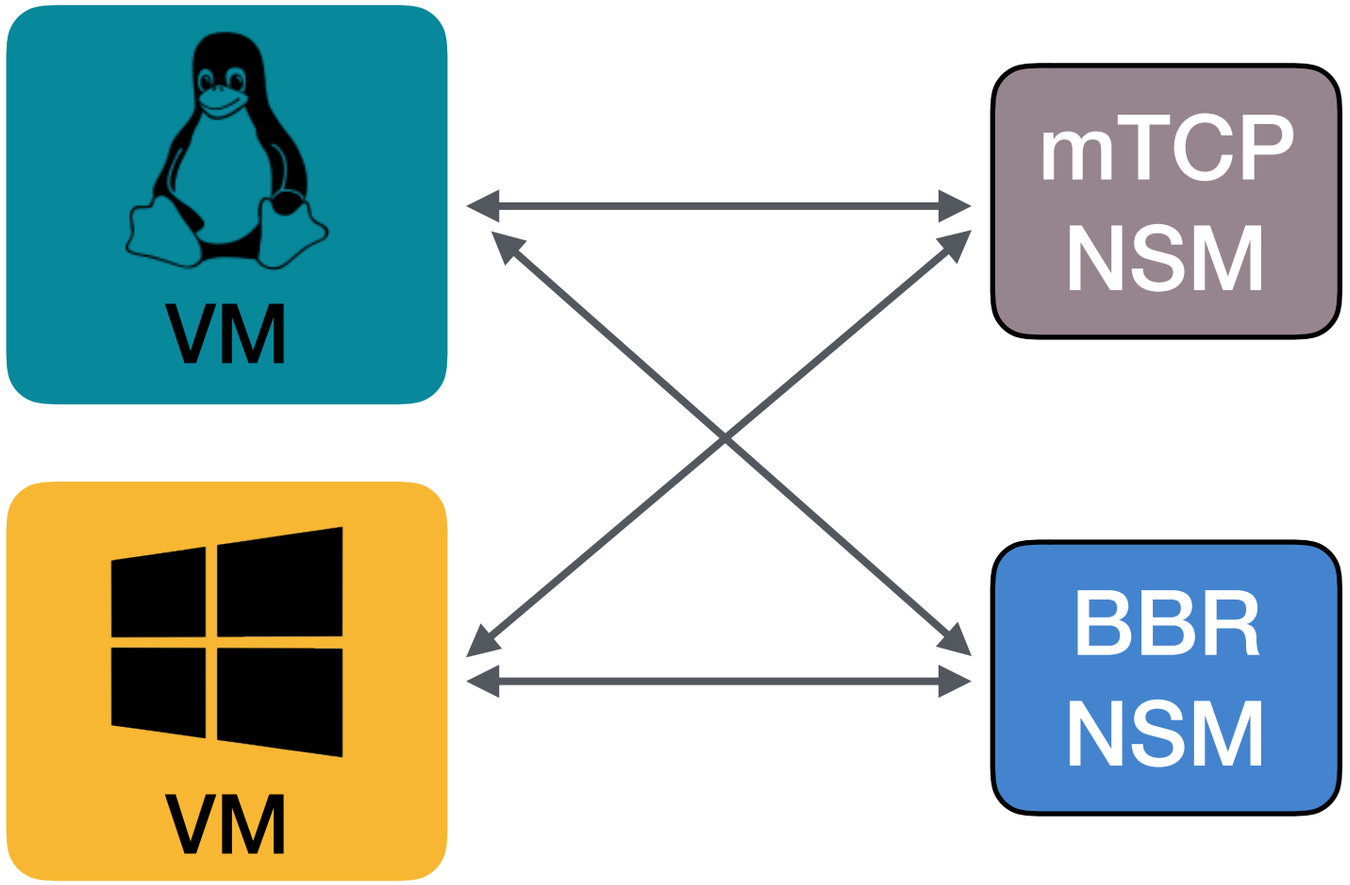


Flexibility for Tenants



- Stack independent of the guest OS

Flexibility for Tenants



- Stack independent of the guest OS
- No deployment or maintenance cost

Efficiency for Provider

Efficiency for Provider

- Offer meaningful SLAs

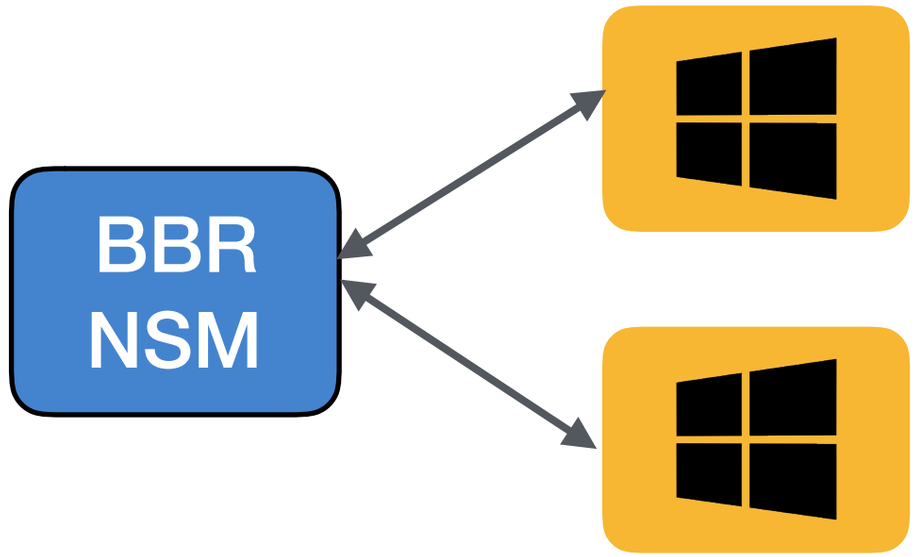
NSM	Capacity	Price
mTCP	25Mpps	$2/hr
mTCP	50Mpps	$4/hr
F-Stack	20Mpps	$2/hr

Efficiency for Provider

- Offer meaningful SLAs

NSM	Capacity	Price
mTCP	25Mpps	$2/hr
mTCP	50Mpps	$4/hr
F-Stack	20Mpps	$2/hr

- Optimize resource utilization

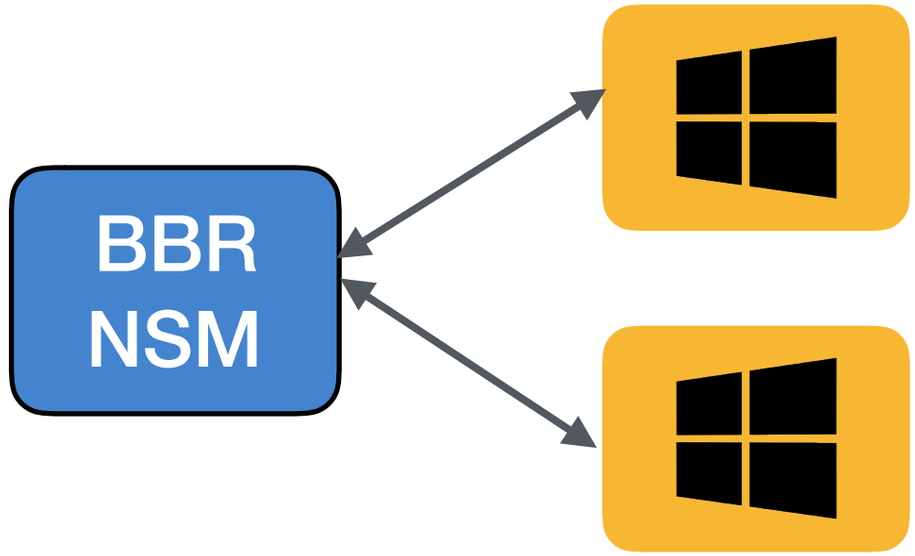


Efficiency for Provider

- Offer meaningful SLAs

NSM	Capacity	Price
mTCP	25Mpps	$2/hr
mTCP	50Mpps	$4/hr
F-Stack	20Mpps	$2/hr

- Optimize resource utilization



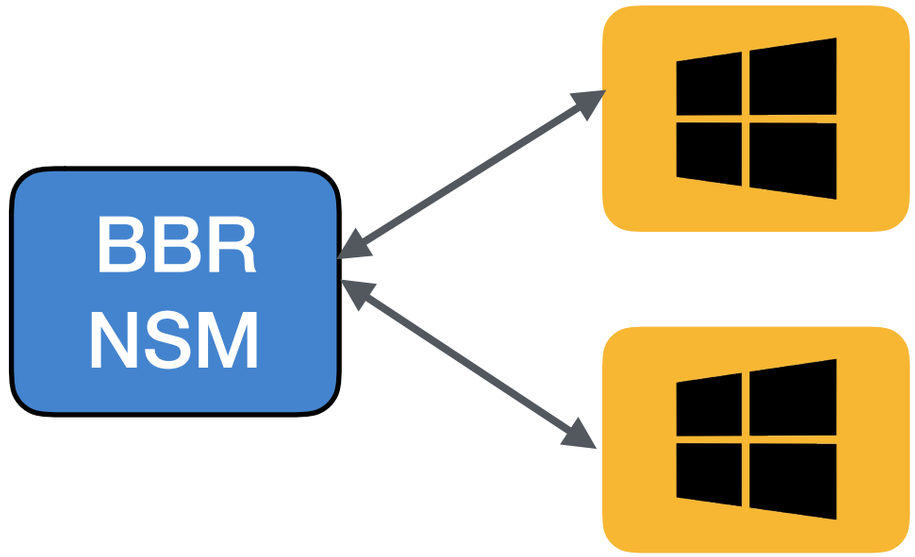
- Easier to assert coordination and control

Efficiency for Provider

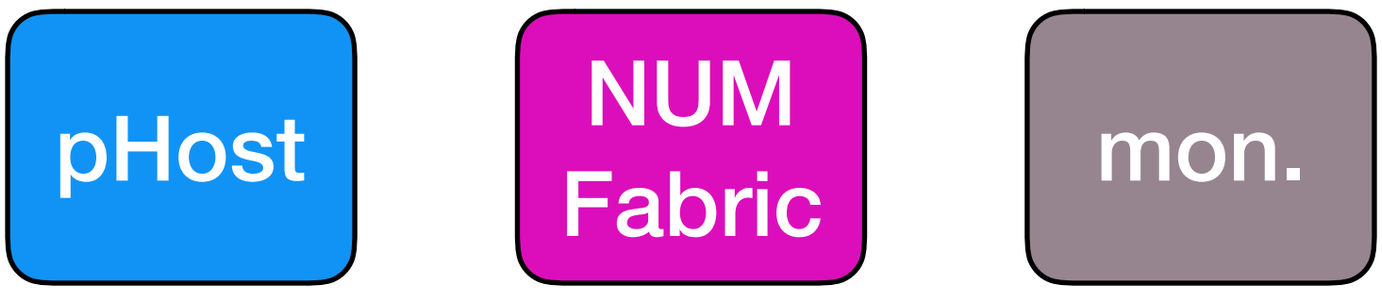
- Offer meaningful SLAs

NSM	Capacity	Price
mTCP	25Mpps	$2/hr
mTCP	50Mpps	$4/hr
F-Stack	20Mpps	$2/hr

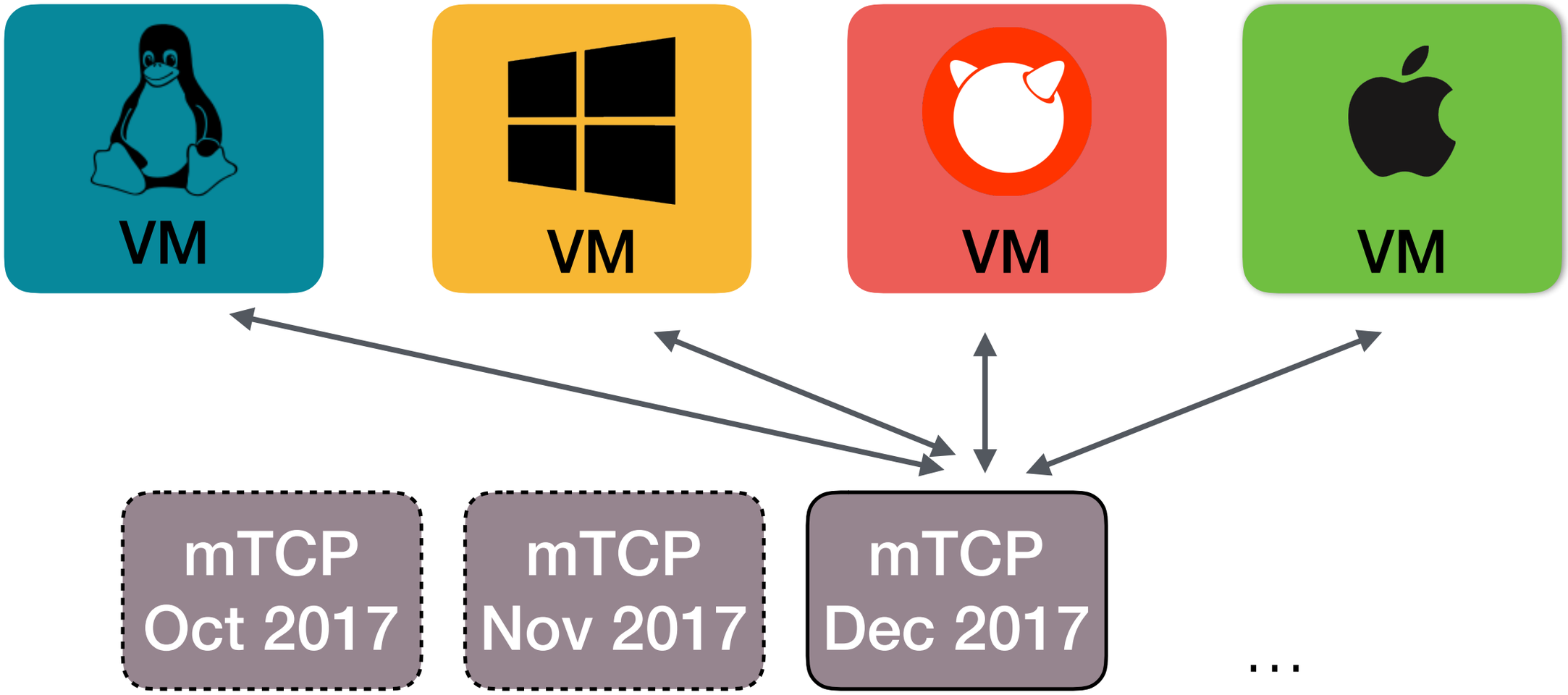
- Optimize resource utilization



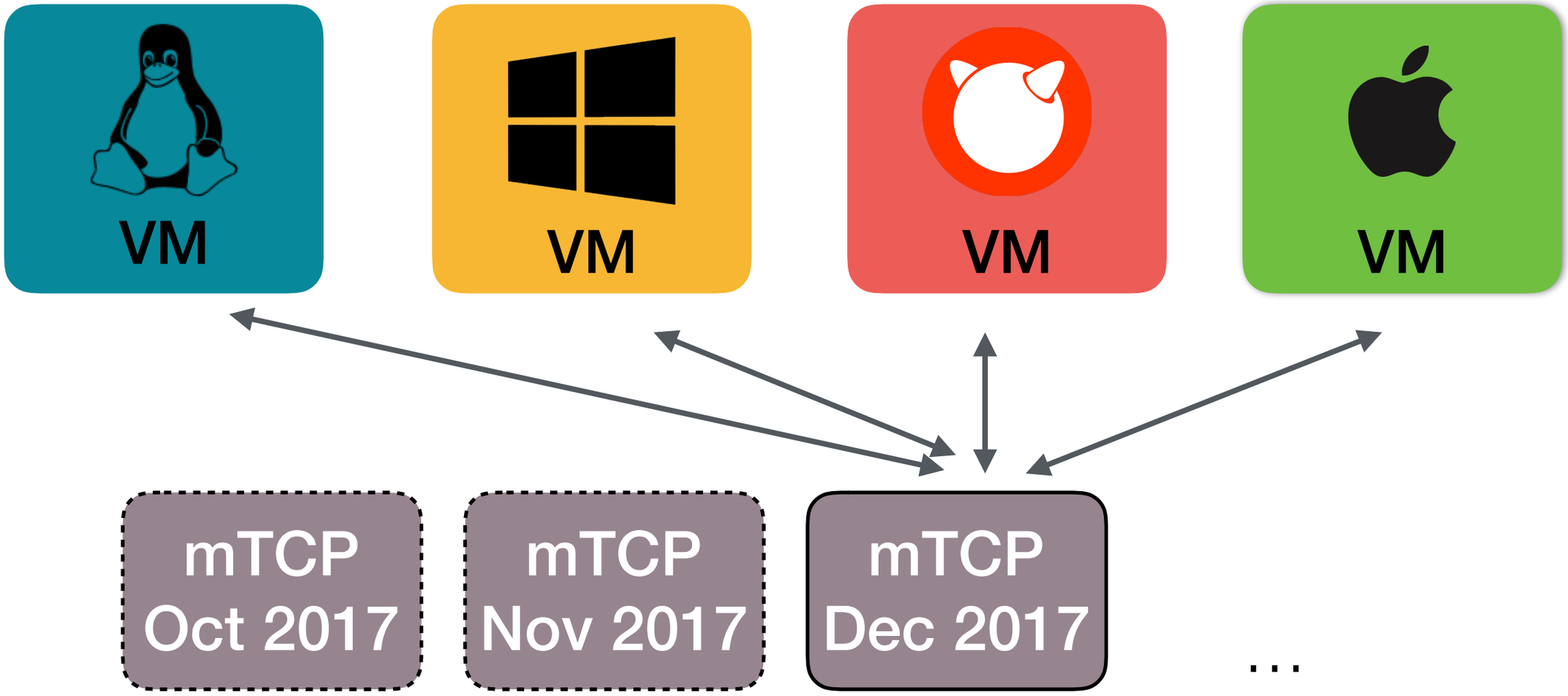
- Easier to assert coordination and control



Accelerate Innovation

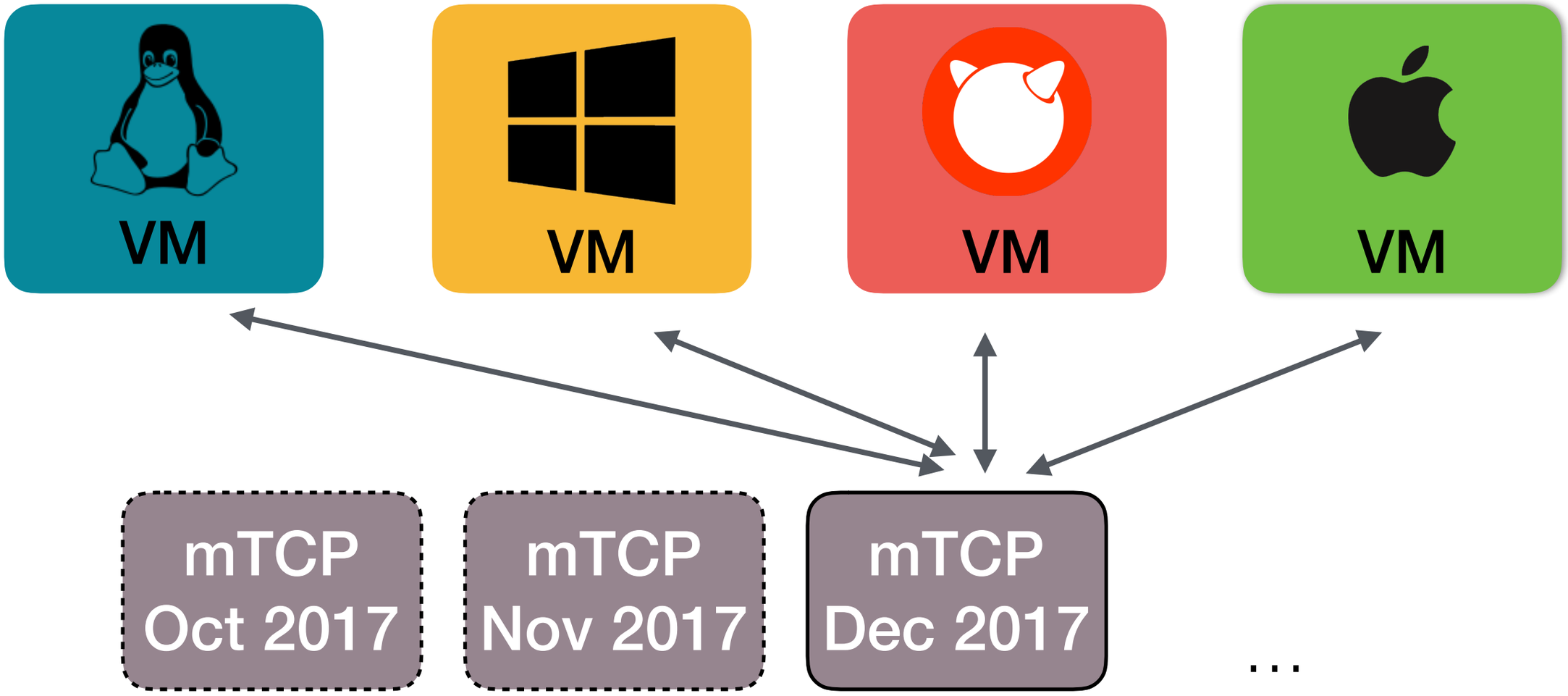


Accelerate Innovation



- Allow stack to evolve independently with the guest OS
- Write once, run everywhere

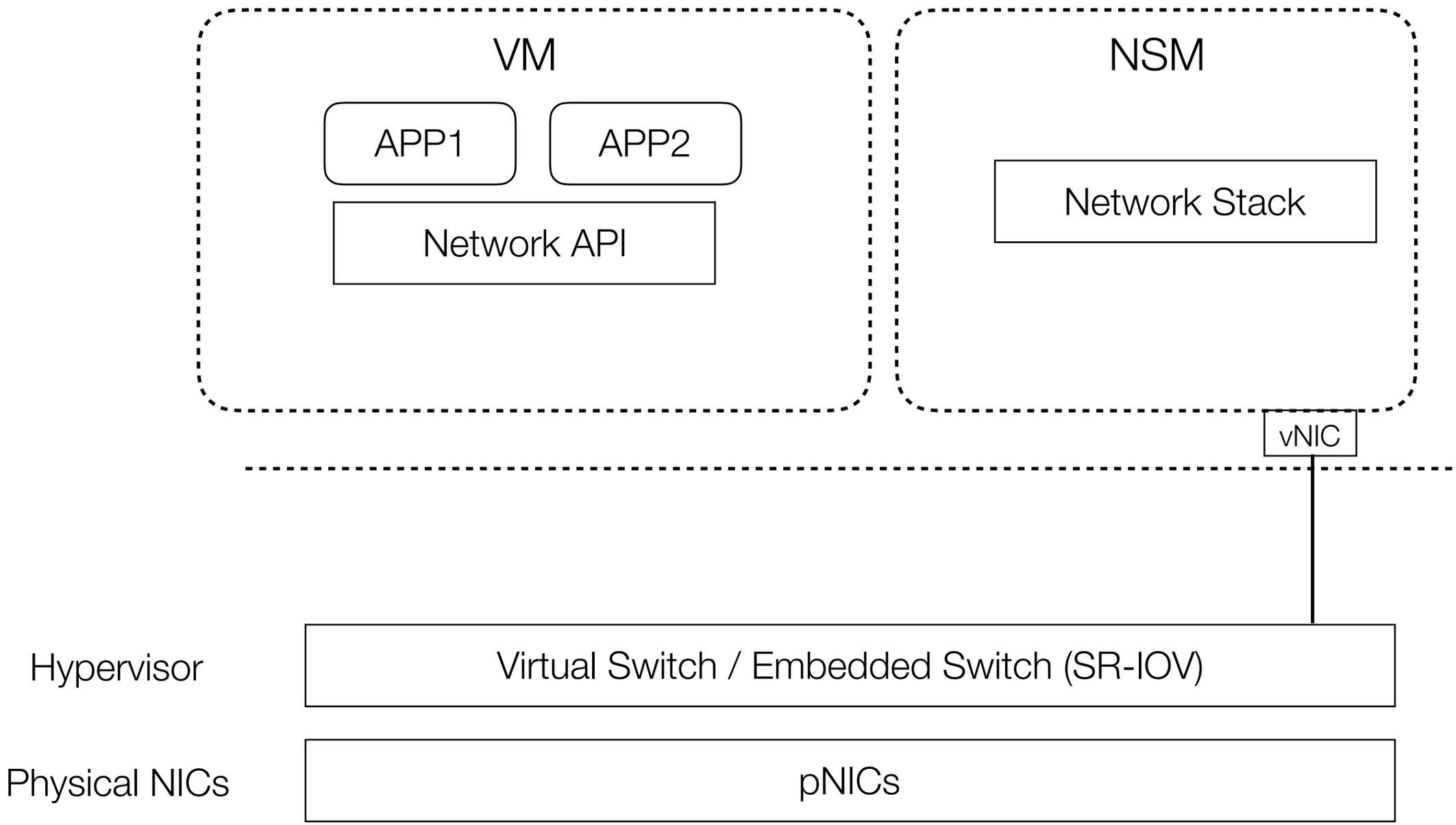
Accelerate Innovation



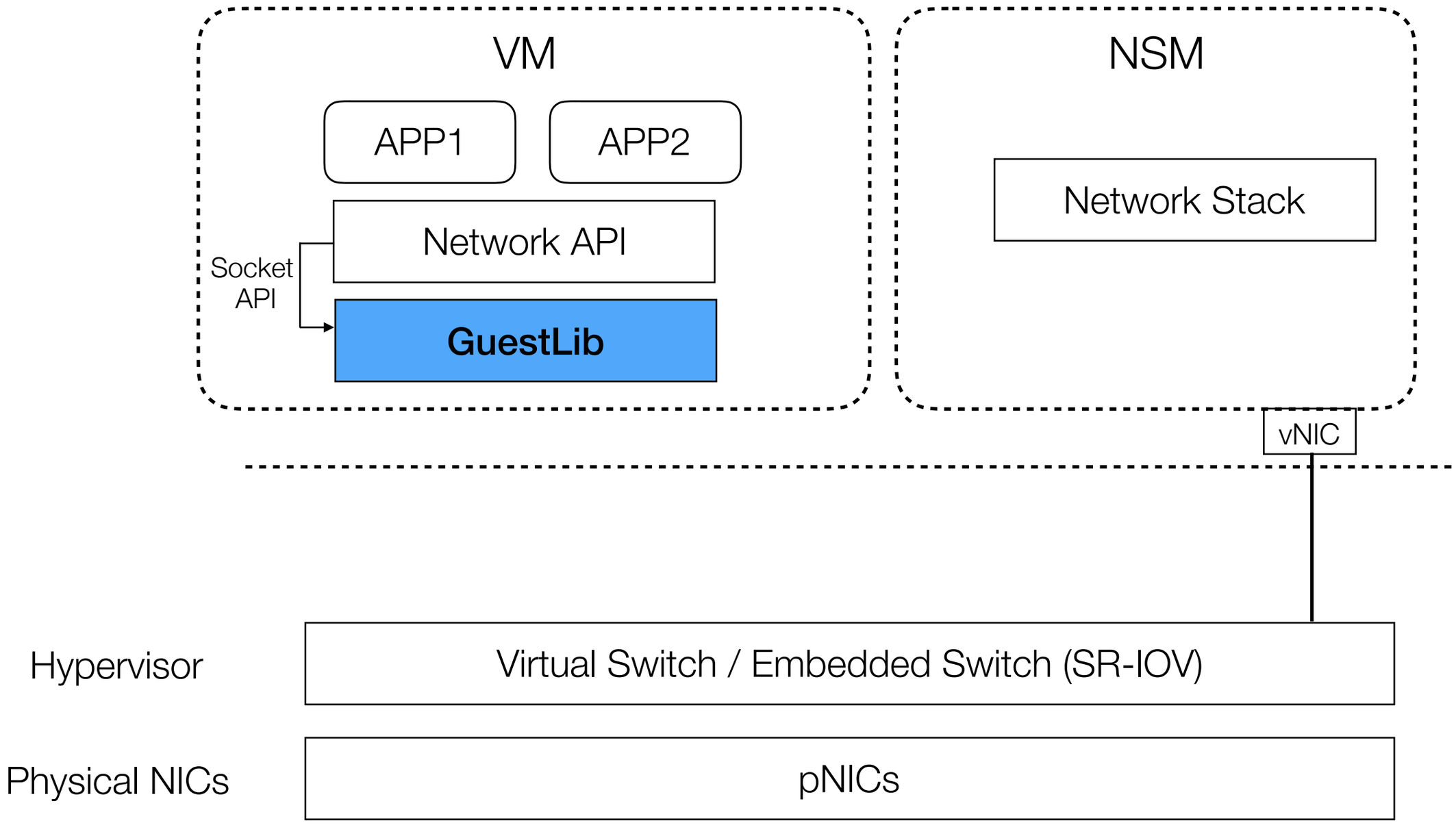
- Allow stack to evolve independently with the guest OS
- Write once, run everywhere

Not possible in current architecture

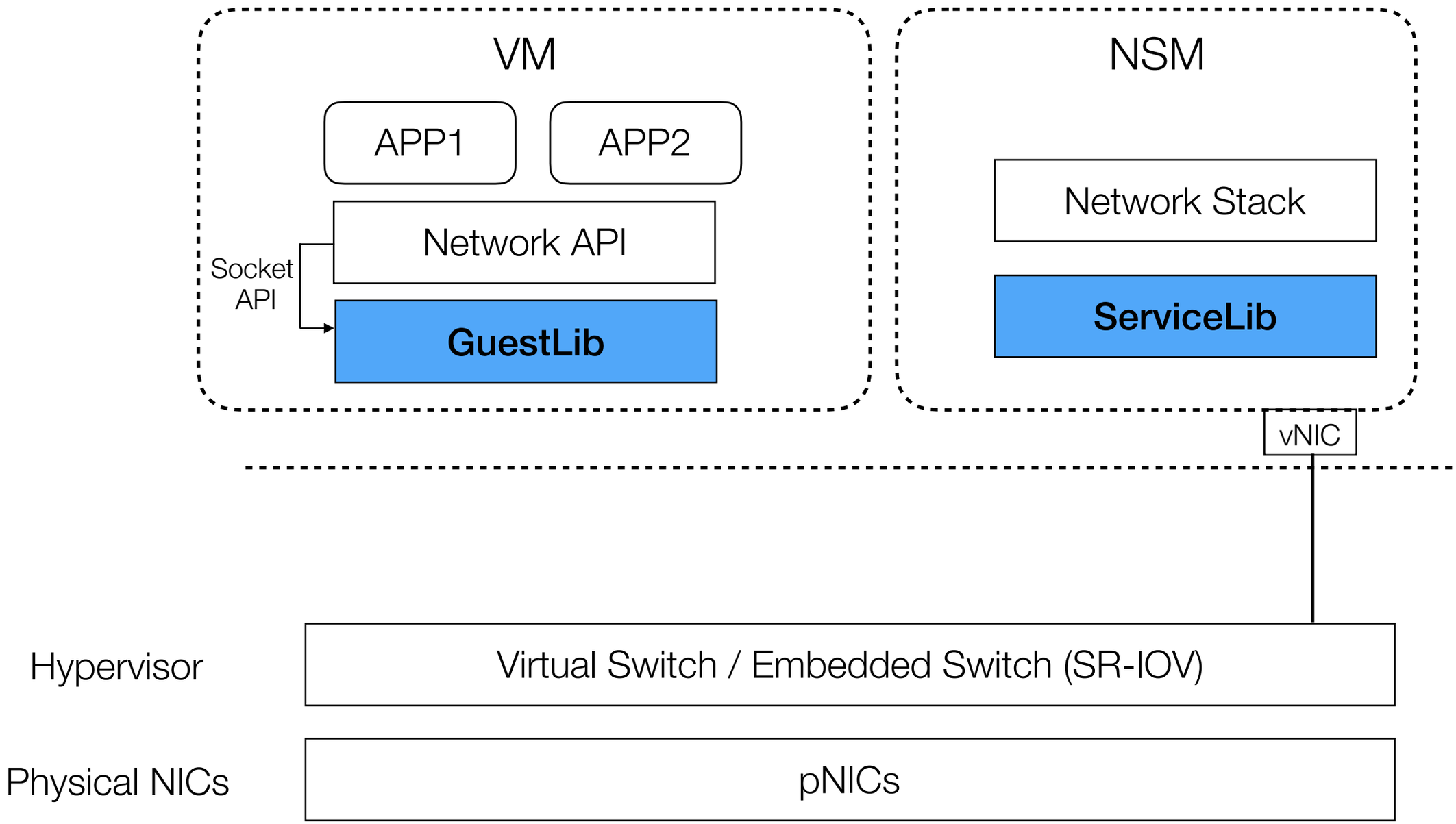
NetKernel


VM
APP1
APP2
Network API
NSM
Network Stack
vNIC
Hypervisor
Virtual Switch / Embedded Switch (SR-IOV)
Physical NICs
pNICs

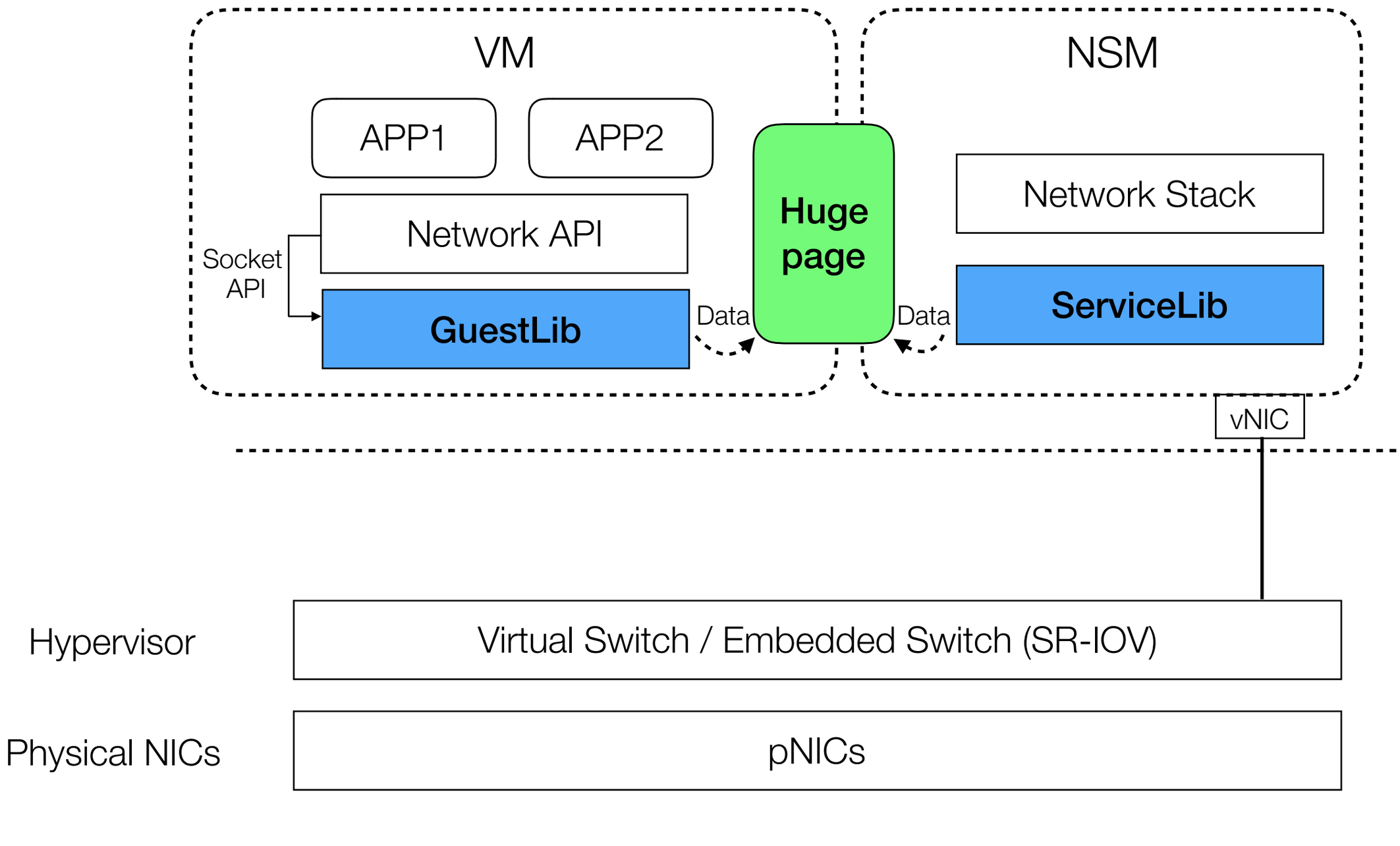
NetKernel


VM
APP1
APP2
Network API
Socket API
GuestLib
NSM
Network Stack
vNIC
Hypervisor
Virtual Switch / Embedded Switch (SR-IOV)
Physical NICs
pNICs

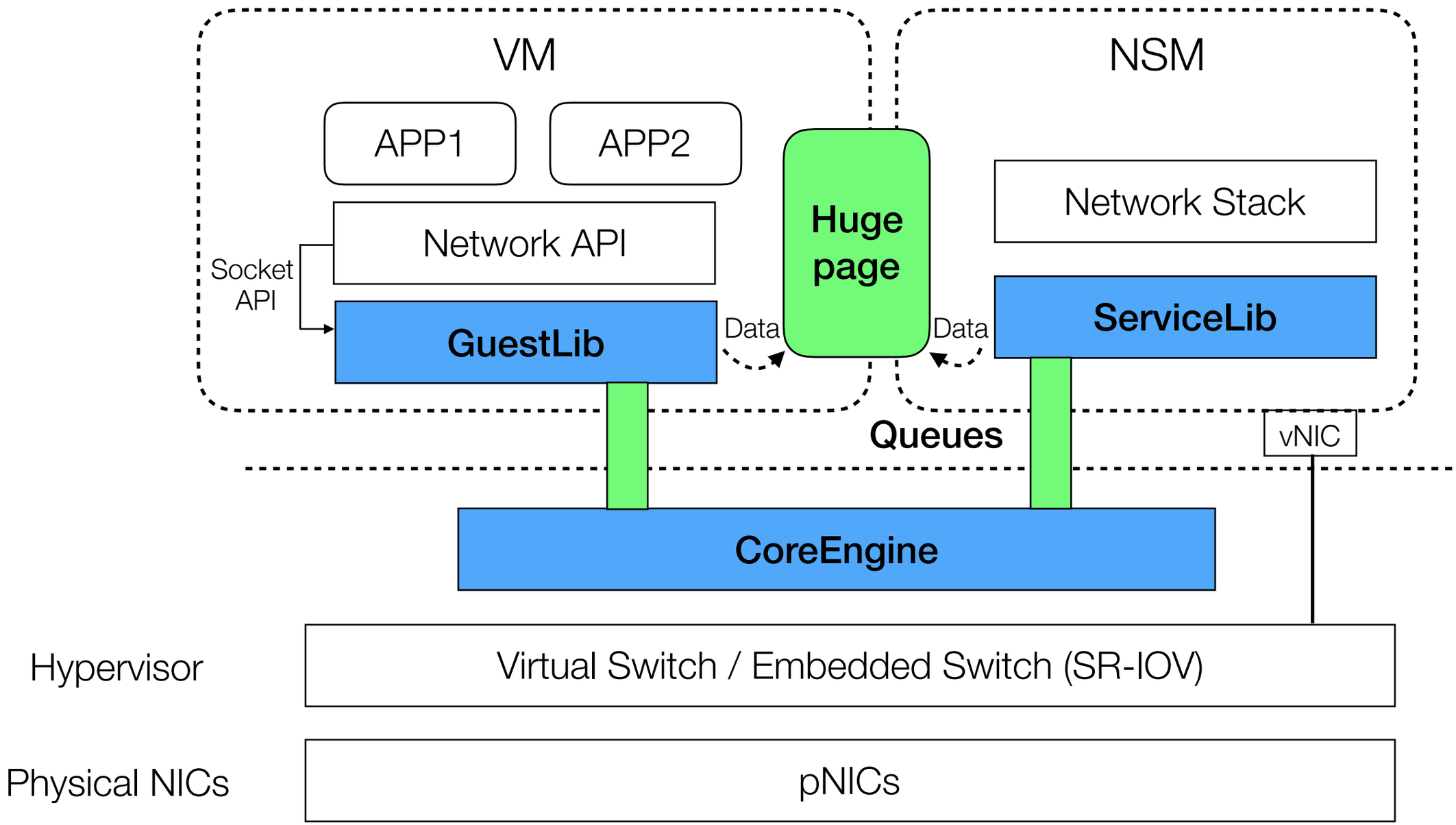
NetKernel


VM
APP1
APP2
Network API
Socket API
GuestLib
NSM
Network Stack
ServiceLib
vNIC
Hypervisor
Virtual Switch / Embedded Switch (SR-IOV)
Physical NICs
pNICs

NetKernel


VM
APP1
APP2
Network API
Socket API
GuestLib
Data
Huge page
Data
NSM
Network Stack
ServiceLib
vNIC
Hypervisor
Virtual Switch / Embedded Switch (SR-IOV)
Physical NICs
pNICs

NetKernel


VM
NSM
APP1
APP2
Network API
Socket API
GuestLib
Huge page
Data
Data
Network Stack
ServiceLib
Queues
vNIC
CoreEngine
Hypervisor
Virtual Switch / Embedded Switch (SR-IOV)
Physical NICs
pNICs

Microbenchmark

- 3000 lines of C code, in user space
- QEMU KVM 2.5.0, Linux Kernel 4.9
- Intel Xeon CPU E5-2618L v3 @ 2.30GHz x 2

Communication between ServiceLib and GuestLib
(Random read and copy)

Chunk size	64B	512B	1KB	2KB	4KB	8KB
Latency	8ns	64ns	117ns	214ns	425ns	809ns

Microbenchmark

- 3000 lines of C code, in user space
- QEMU KVM 2.5.0, Linux Kernel 4.9
- Intel Xeon CPU E5-2618L v3 @ 2.30GHz x 2

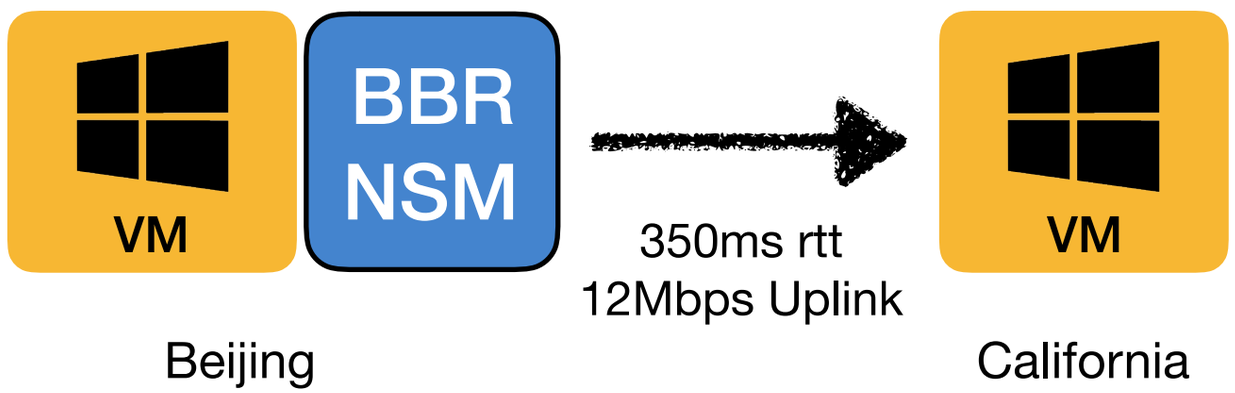
Communication between ServiceLib and GuestLib
(Random read and copy)

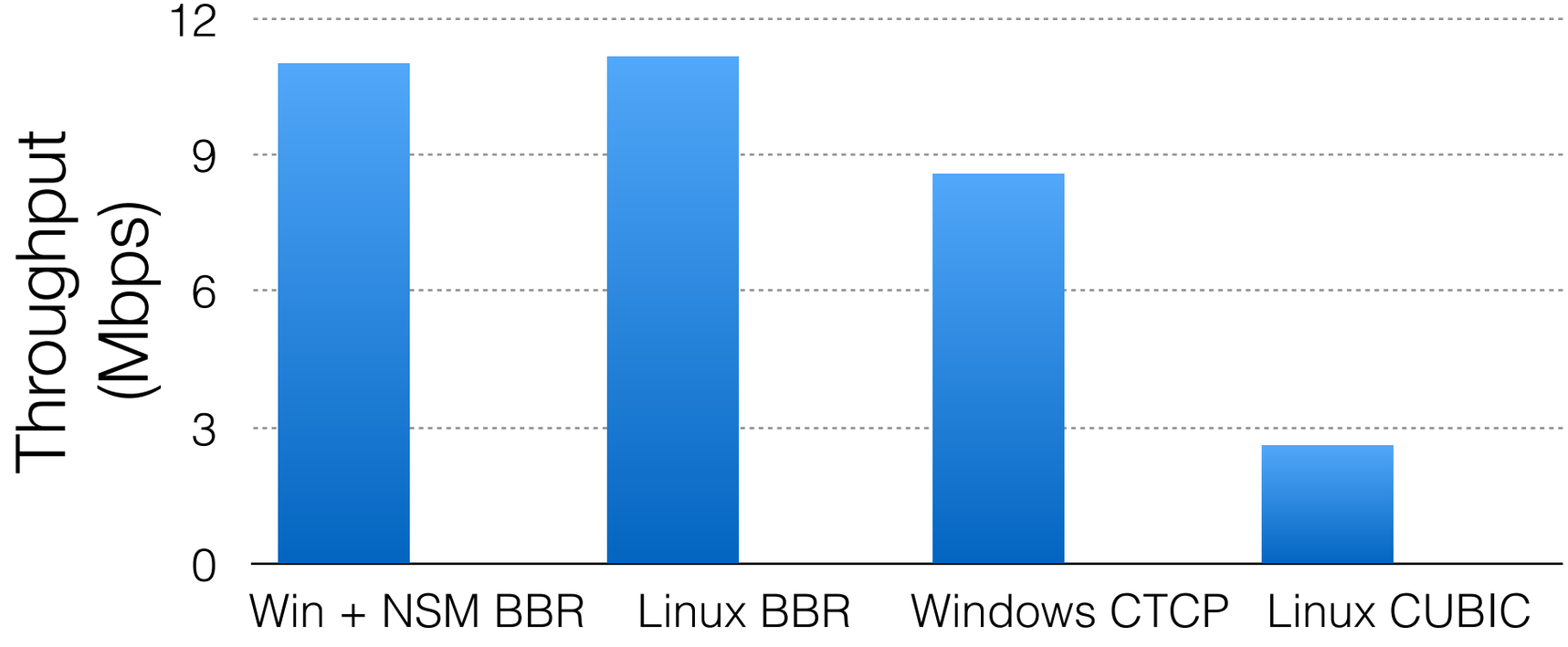
Chunk size	64B	512B	1KB	2KB	4KB	8KB
Latency	8ns	64ns	117ns	214ns	425ns	809ns

64Gbps

81Gbps

Windows VM + BBR NSM





Takeaway

- Vision: Network Stack as a Service
  - Decouple the network stack from the guest OS
  - Better flexibility and efficiency, and faster innovation
- NetKernel as a solution
  - GuestLib, ServiceLib, CoreEngine

Research Agenda

Research Agenda

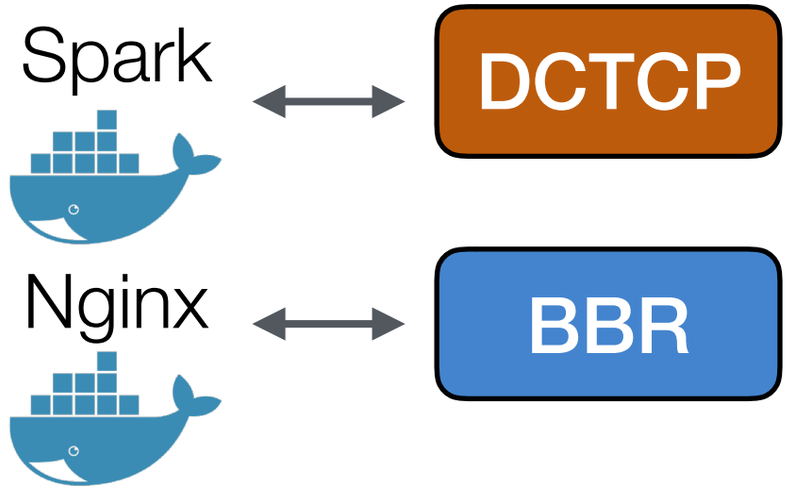
- NSM form
  - VM? unikernel-based VMs? containers? hypervisor modules?

Research Agenda

- NSM form
  - VM? unikernel-based VMs? containers? hypervisor modules?
- Support for containers
  - Currently a container has to use the host stack
  - Different containers on the same host use different stacks

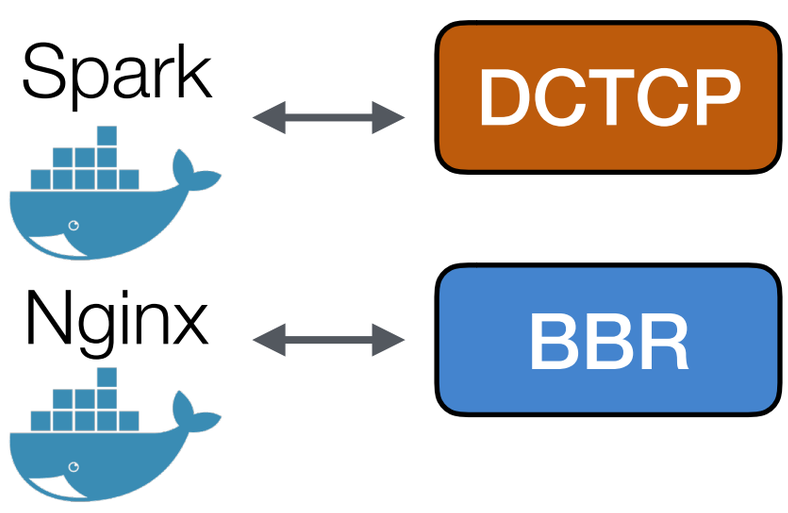
Research Agenda

- NSM form
  - VM? unikernel-based VMs? containers? hypervisor modules?
- Support for containers
  - Currently a container has to use the host stack
  - Different containers on the same host use different stacks



Research Agenda

- NSM form
  - VM? unikernel-based VMs? containers? hypervisor modules?
- Support for containers
  - Currently a container has to use the host stack
  - Different containers on the same host use different stacks



- Network stacks to NSMs
- …

Open Questions

- Any downsides?
- Other use cases in a production cloud?
- How about a private data center?
- What's the right abstraction boundary of the network stack?