NEOCLEUS
Virtualizing End Points

Neocleus Client Hypervisor

Ze'ev Maor

May 2009

Agenda

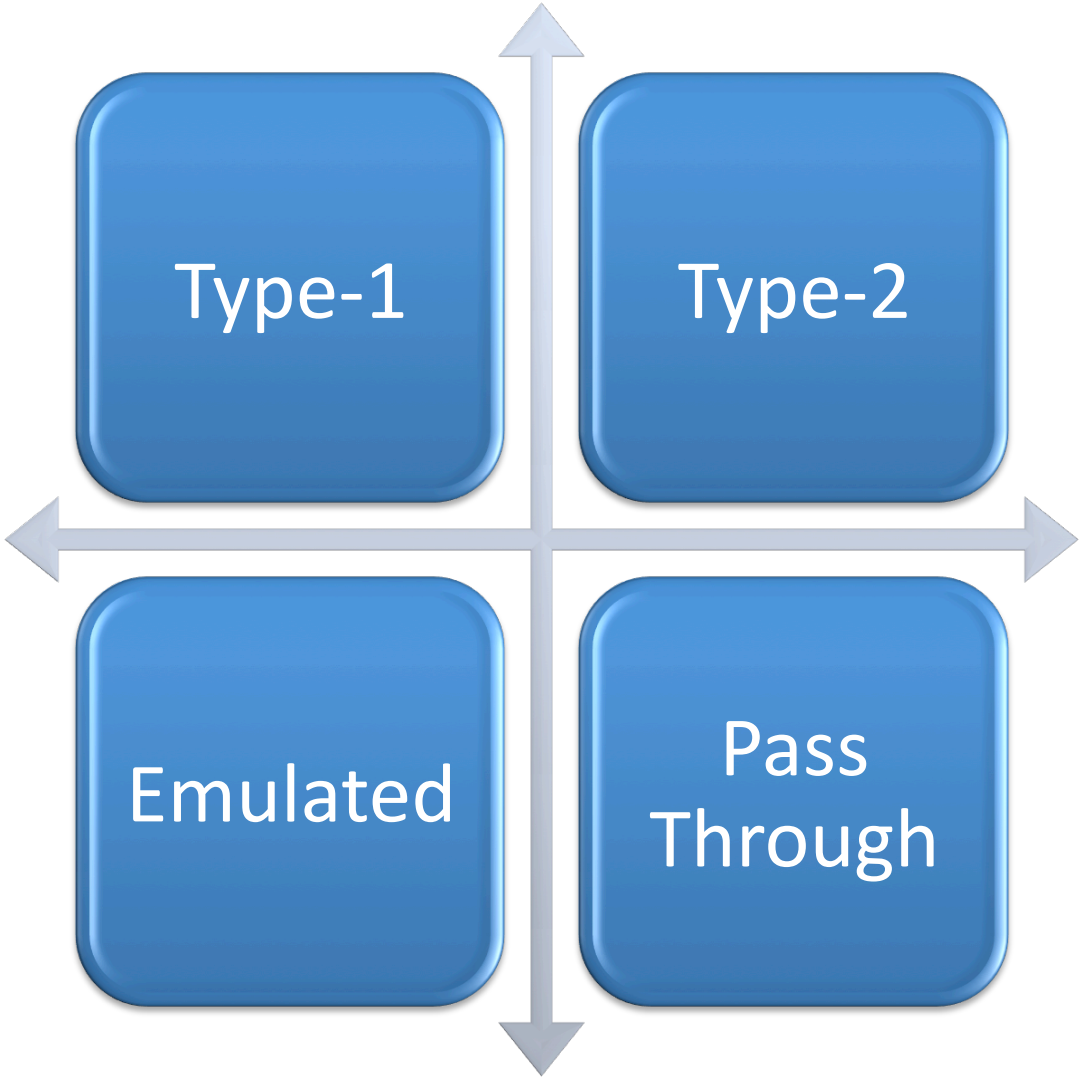
- Client virtualization challenges
- Use cases
- Type 1 vs. Type 2 in clients
- I/O virtualization in clients
- Dynamic device assignment
- Future trends

Client hypervisor challenges

- User centered interactive experience
- Varied range of I/O devices
- Graphics, sound and internet are paramount to the experience
- Full ACPI support is required
- Endpoint is “in the wild”, VM isolation is a concern

Bottom line: endpoints weren’t designed to be virtualized

Virtualization technologies matrix



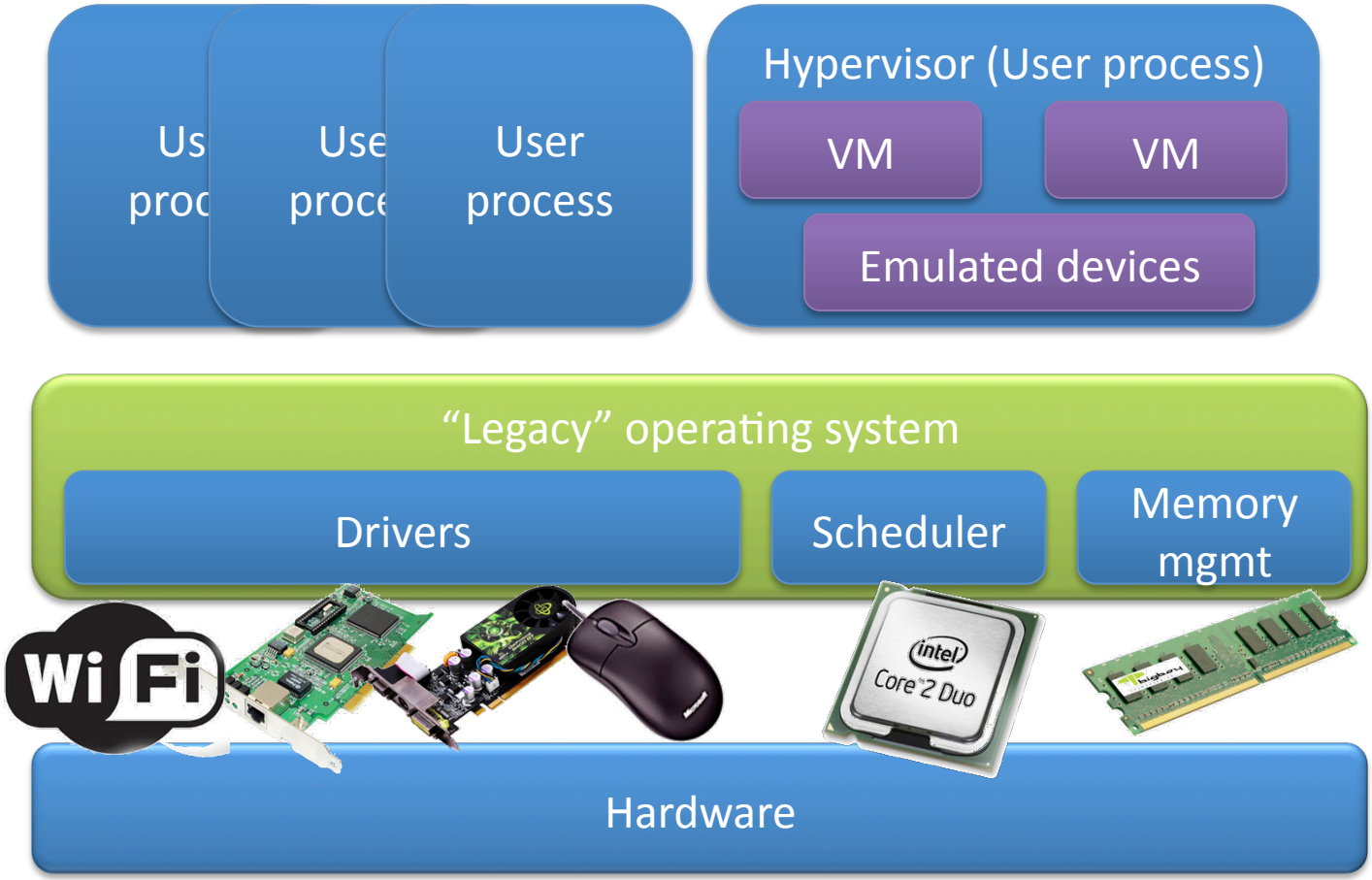
Use cases

- Personal and Corporate environments side-by-side
- Endpoint lifecycle management
- Endpoint consolidation
- “Bring your own PC”
- Trusted computing

Objectives

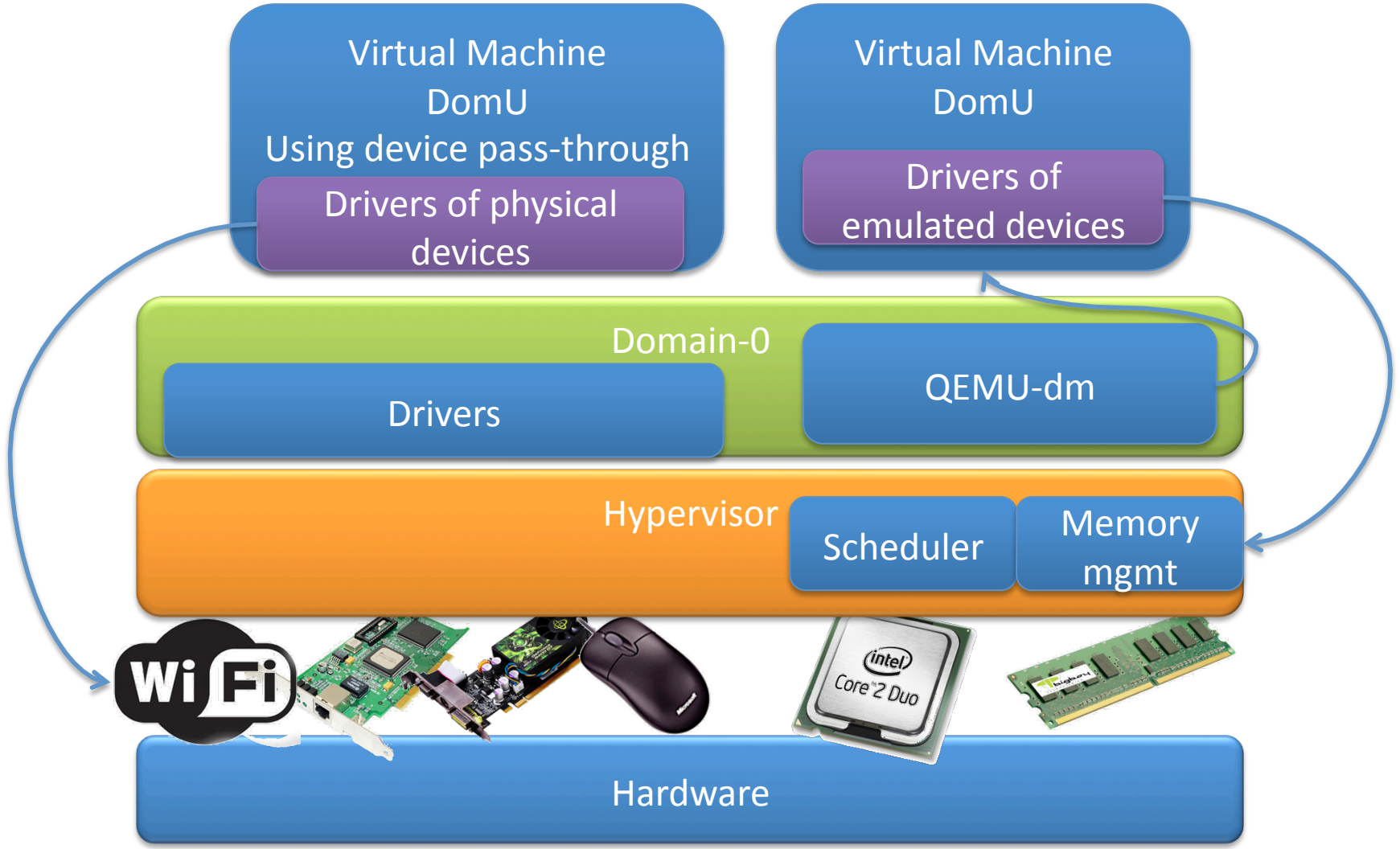
- VM isolation
- Security
- Performance
- HW resource control
- Robustness
- Usability

Type-2 hypervisor



NEOCLEUS
Virtualizing End Points

Type-1 hypervisor (Xen based)



NEOCLEUS
Virtualizing End Points

Type-1 vs. Type-2

Type-1 Hypervisor	Type-2 Hypervisor
Runs directly on the HW – “bare metal”	Runs as a user space process under a standard OS
Intrusive by nature	Non-intrusive
Deployment is a challenge	Simple to deploy
Total VM isolation	VMs (and hypervisor) susceptible to viruses/maleware infecting the OS
Owns the CPU, schedules VMs directly	Depends on the host OS scheduler

Emulated vs. Pass-through I/O

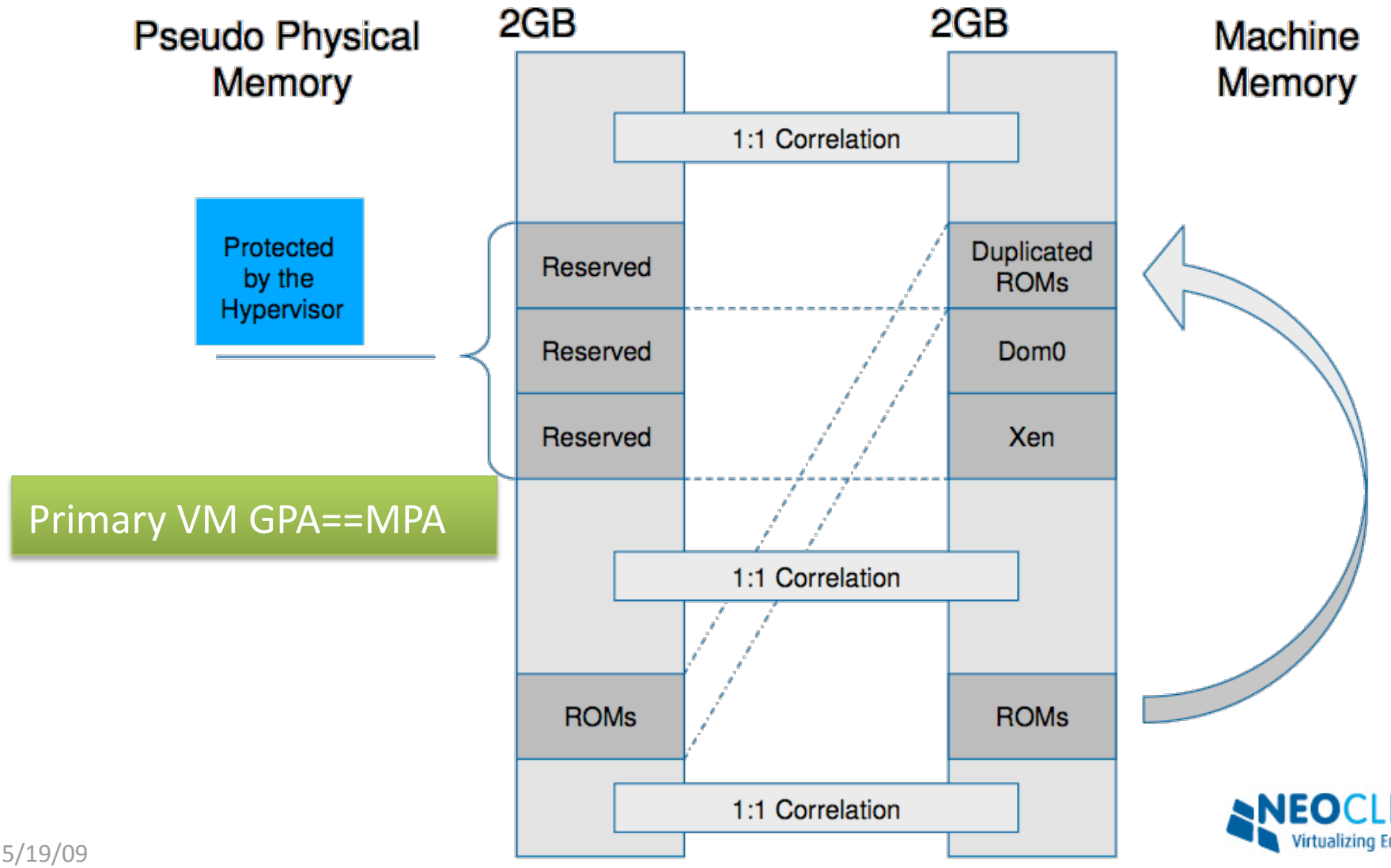
Emulated I/O	Pass-Through I/O
HW independence, easy deployment and migration	Native drivers are required for the specific HW
Low performance regardless of HW	Full native performance
Restricted feature set, lowest common denominator	Full HW feature set, “get what you paid for”
Allows complex HW control scenarios with PV drivers (e.g. dynamic device assignment)	Restricted to OS/native driver behavior

Type 1 I/O Virtualization - passthrough

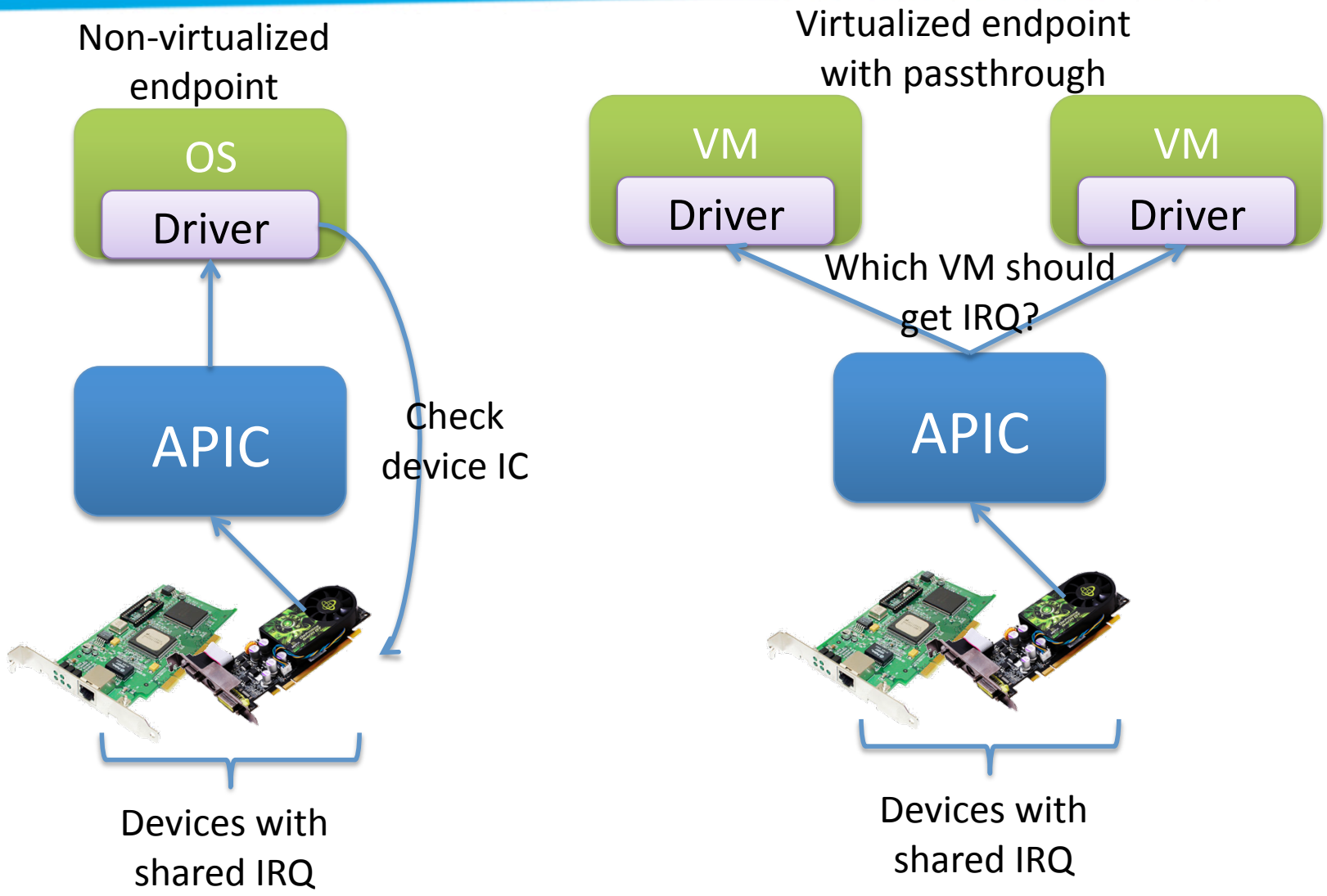
- Main challenges: DMA and interrupts
- Hardware assisted:
  - Intel VT-d (e.g. ICH9/PM45 chipset)
  - AMD IOMMU
- 100s of millions of endpoints without I/O-V assisted HW are deployed.
- Neocleus client hypervisor solution
  - 1:1 memory mapping
  - Interrupt sharing

Neocleus 1:1 MM

Load the “primary” VM into its’ native memory region.



Neocleus interrupt sharing



Dynamic device assignment

- Required in client virtualized environment
- Implementation for specific I/O-V techniques:

I/O virtualization technique	Dynamic assignment
Emulated I/O	PV driver + backend multiplexing
I/O over network	Network switch in Dom0
Passthrough with IOMMU	Switching I/O page tables
Passthrough without IOMMU (1:1 MM)	A real challenge…

ACPI Passthrough

- A must for a proper laptop experience:
  - Battery
  - Lid
  - AC
  - Sleep mode
  - Special function keys
- Solution:
  - decompile vendor DSDT and integrate into the VM DSDT
  - Use ACPID in Dom0 to trap ACPI events and pass them to the VM

CPU virtualization

- Standard OS scheduler
  - has knowledge of process “profile”
  - Can penalize CPU bound processes and credit I/O bound processes
  - The result is a more responsive user interaction
- Hypervisor
  - Sees an entire VM as a single process
  - Can’t discern I/O bound and CPU bound processes
  - Can’t even discern an idle OS process
  - Need to schedule the I/O back end in Dom0 as well

VM isolation

- Why?
  - Robustness requires that no one VM may crash the endpoint
  - Security mandates that compromised VMs can't contaminate other VMs
- How?
  - VMs memory mgmt utilize shadow page tables
  - PT implementations must support dynamic assignment of critical devices
  - I/O passthrough with 1:1 mapping susceptible to DMA attacks - still a challenge…

NEOCLEUS Virtualizing End Points

Future trends

- SR-IOV
- Widespread deployment of IOMMU
- Nested VMs
- Windows switchable graphics
- Commoditized hypervisor

Thank you

Questions ?

Contact us: info@neocleus.com

NEOCLEUS
Virtualizing End Points