Operációs Rendszerek MSc

Virtualizáció

2019/2020/I.

Dr. Vincze Dávid
Miskolci Egyetem, IIT
vincze.david@iit.uni-miskolc.hu
http://users.iit.uni-miskolc.hu/~vinczed/

Operációs Rendszerek MSc

- Virtualizáció
  - Koncepcionális szinten régóta létezik
  - Implementáció is természetesen
  - Beágyazott rendszereken is jelen van
    - konszolidáció
    - RTOS vs. GPOS
    - fokozott információ biztonság
    - stb.

Operációs Rendszerek MSc

- Virtualizáció
  - Virtuális ...
    - Memória
    - Háttértár (RAID, particionálás, LVM)
    - Hálózat (VLAN, channel)
    - Processz
    - Szuperszámítógép, klaszter
    - Osszunk szét egy gépet virtuális gépekre, azaz virtuális operációs rendszerekre
      (Ellentéte a klaszтereknek, szuperszámítógépnek...)
  - A virtualizáció igazából egy paradigma, sok technológiát, módszert foglal magában

Operációs Rendszerek MSc

- Virtualizáció
  - A kezdetek: **IBM System/370**
  - 70-es évek elején már képes volt virtualizációra
  - VM/370 OS
  - Control Program (hypervisor)
  - Teljes virtualizációt biztosított (CPU, I/O, mem)
  - Később részben paravirtualizációs része is volt

  - Tehát:
    - Régóta létezik virtualizáció ('70)
    - De akkor csak nagygépeken (mainframe)
    - 2005 környékén új fellángolás (x86 HW támogatás)
    - Manapság már mindenhol jelen van:
      - Szerver, desktop, beágyazott rendszerek... → felhő...

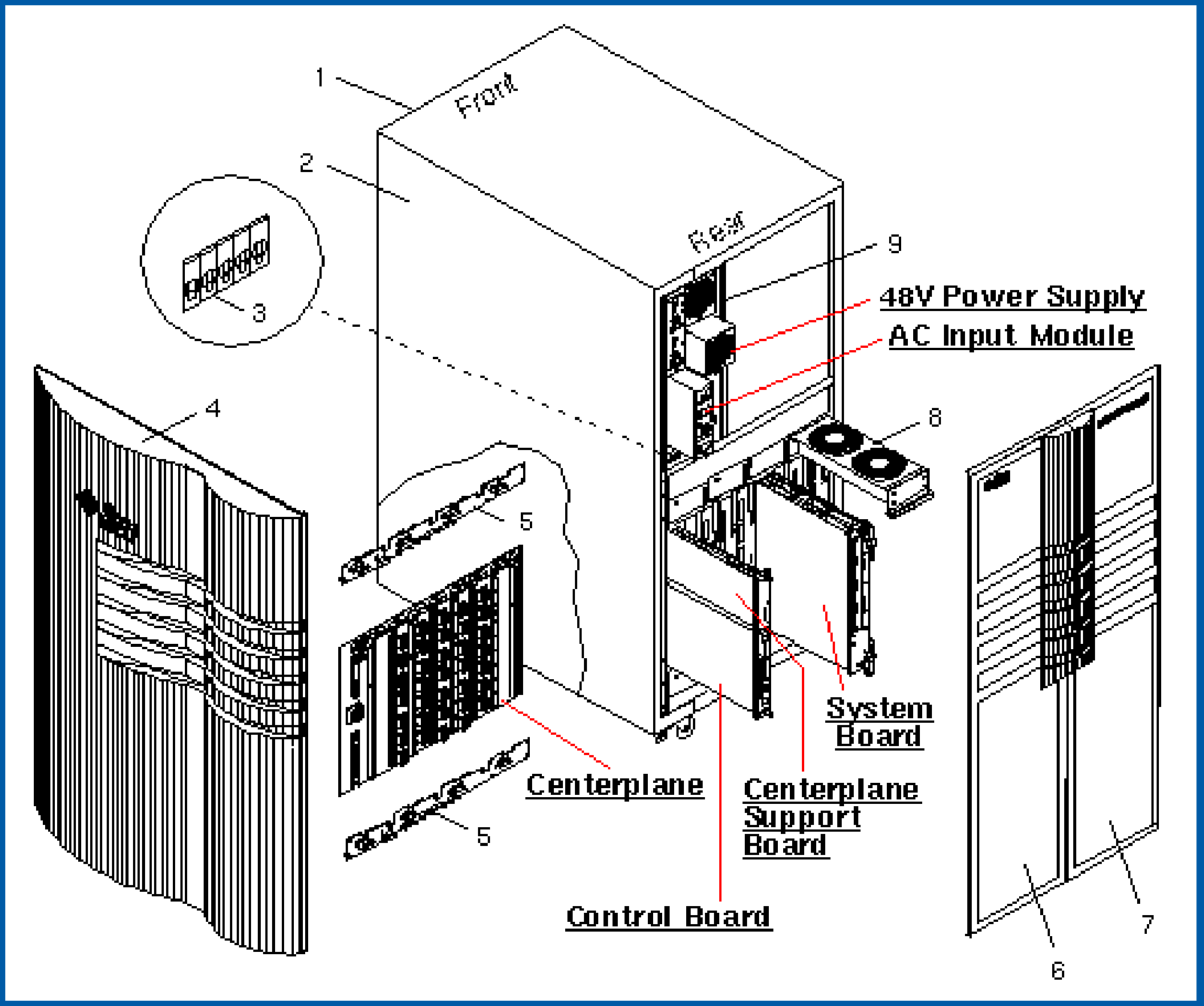
System/370 konzol

Operációs Rendszerek MSc

- Virtualizáció
  - A folytatás: **Sun Enterprise 10000** (Starfire)
  - Ez kicsit más megközelítésű virtualizáció
    - Hard partitions
  - Több CPU kártyából áll a rendszer (sys. board)
  - A CPU kártya az alap egység
    - Ezen van több CPU
    - RAM
    - SBus
      - Ezen többnyire: storage controller (SCSI), network (Eth)
  - Ezekből lehet több virtuális gépet definiálni
  - Összerendeli a CPU kártyákat
  - Ezeken külön-külön OS-t lehet használni (Solaris)
  - Külön virtuális konzollal mindegyik

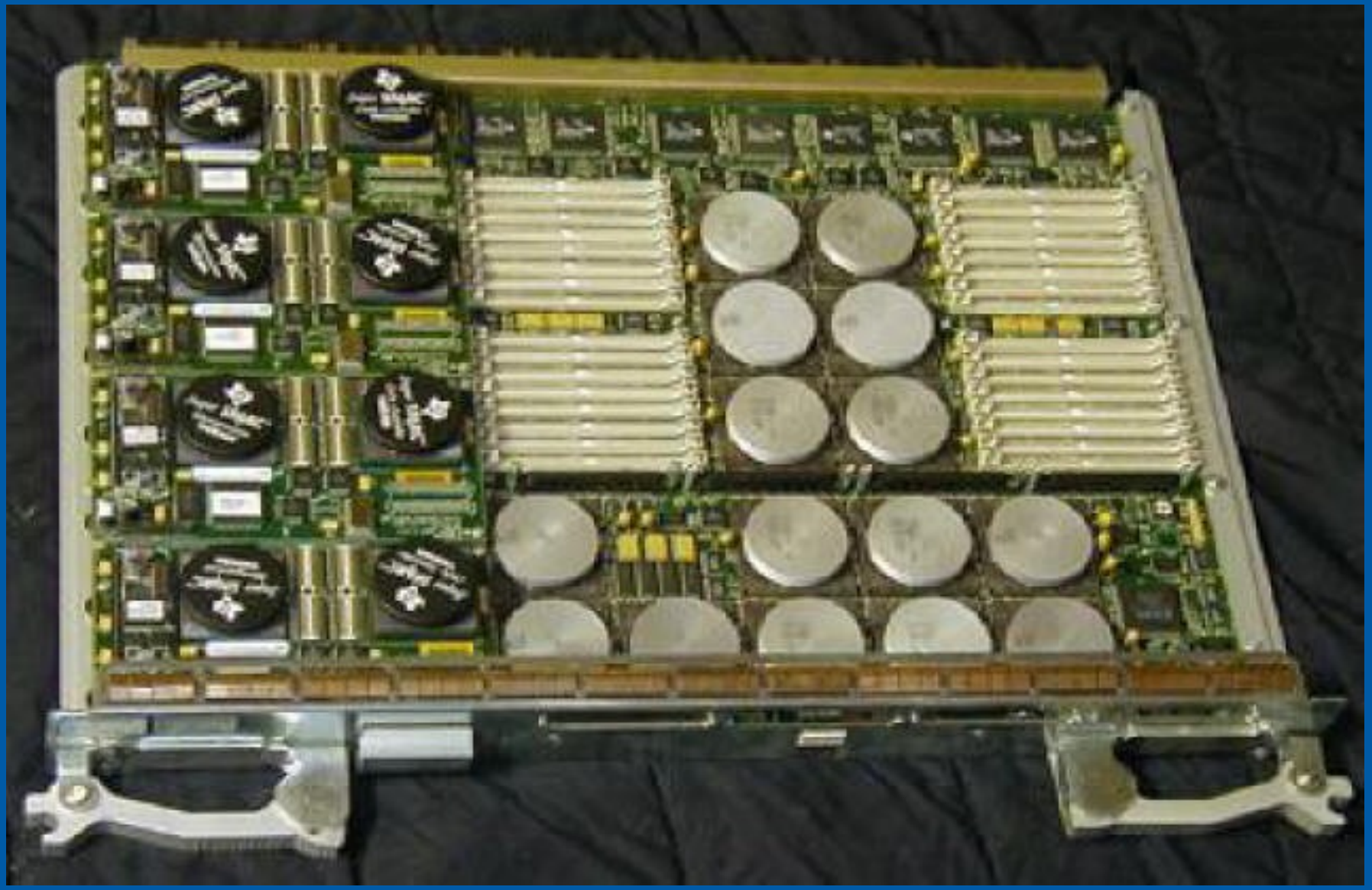
Sun Enterprise 10000

E10000 komponensek

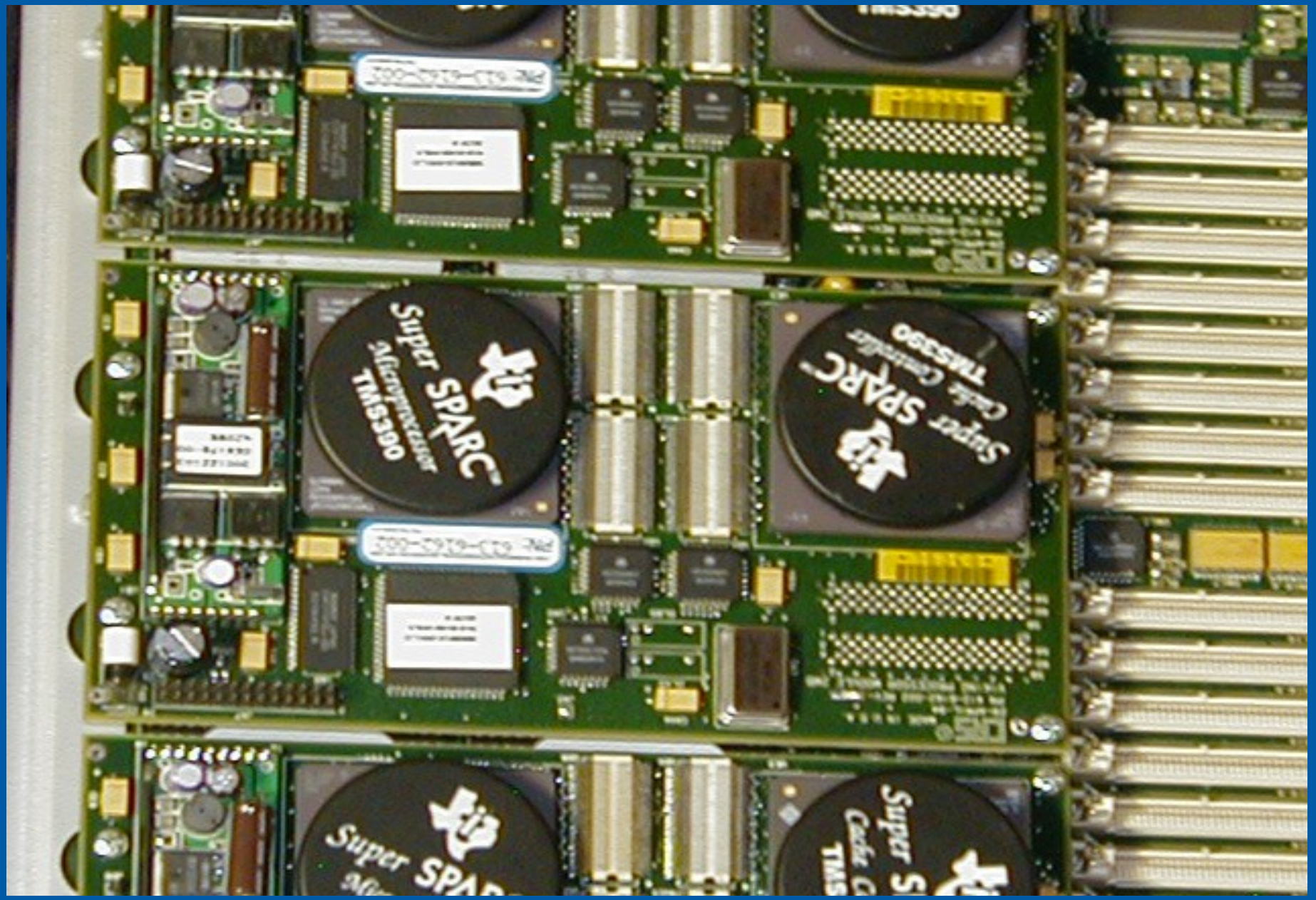


E10000 System Board

E10000 System Board



E10000 CPU



Virtualizáció – IBM LPAR

- Virtualizáció - **IBM LPAR**
  - Logical PARtitioning
  - Nagygépeken elérhető szolgáltatás (POWER arch)
  - Régebben hasonlóan, mint a Sun E10000-nél
  - Később PowerVM-el jobb erőforrás szétosztás
  - CPU: dedikált, vagy osztott lehet a partíciók között
    - Osztott: capped, uncapped
  - RAM: méretet lehet megadni, nem lehet osztozni
  - Periféria: lehet osztozni fizikai diszkeken, hálózati csatolókon, stb. Ehhez egy külön kiszolgáló rész fut, ami biztosítja a virtuális eszközök backend-jét.

Virtualizáció – IBM LPAR

- Virtualizáció - **IBM LPAR**
  - Támogatott vendég OS-ek:
    - AIX
    - z/OS, z/stb..
    - i5/OS
    - Linux
  - Micro-partitioning:
    - 1/10 CPU mag osztható ki legkisebb egységként
    - Gépenként max 254 partíció (ennyi külön OS egyszerre)
    - (254 mert egy „service LPAR” is van)
  - Dynamic LPAR:
    - dinamikusan változtatható erőforrás kiosztás leállás nélkül
  - Workload Partition:
    - AIX-on belüli virtualizáció, izoláció (konténer, lásd később)
    - Hasonló mint Solaris Containers, OpenVZ, stb.

Emuláció - Virtualizáció

- **Emuláció**
  - Mikrokontroller emulátor, C64 emulátor, QEMU, Bochs, DosBox, VMWare (kezdet)
  - CPU-t és perifériákat emulál (komplett gépet, környezetet)
  - Fut rajta az eredeti kód
  - Csak nem közvetlen a CPU-n, hanem egy értelmező program futtatja a kódot (szoftver futtat szoftvert)
  - Sokkal sokkal lassabb, mintha igazi CPU-n futna
  - Jó – ha nincs már ilyen hardware-ünk
  - Jó – mert fejlesztéshez, hibakereséshez kiváló
  - Rossz - mert lassú
  - Rossz - mert lehet valamelyik program speciális (esetleg nem dokumentált) tulajdonságokat használ

OS MSc - Virtualizáció

- OS Kernel alapú virtualizáció - **Chroot**
  - Ez nem „igazi” virtualizáció
  - Alapból támogatják a UNIX-ok, Linux is
  - Megváltoztatja a root-ot (fs), és abban indít egy processzt
    - Aminek a gyerekei szintén ezt a root-ot látják majd
    - Ebben a rootban meg kell lennie minden használni kívánt programnak, library-knek, konfigoknak, stb.
  - Ezen kívülre nem lát
    - így pl. /proc-t sem! (többszöri mount)
  - A processz tér, hálózat, IPC továbbra is közös marad
  - Vannak módszerek, hogyan lehet „kitörni” a chroot-ból
  - Elsősorban biztonsági kernel patch, de chroot-hoz is jól jöhet: GrSecurity
    - Chroot „erősítések”, pl. külső processzek elrejtése, mount tiltás, stb.
    - Így már egész jól elkülöníthetőek bizonyos processz csoportok

OS MSc - Virtualizáció

- OS Kernel alapú virtualizáció - **Konténer**
  - Pl. Linux namespaces, Linux control groups, LXC, Virtuozzo, OpenVZ
  - Egy közös kernel fut
  - Ezért közel azonos a teljesítmény, mintha csak az OS futna a HW-en
  - Ezen belül maga a kernel biztosít kvázi egyenrangú környezetet (namespace-k)
    - Szeparált processz tér, IPC tér, network, felhasználói és csoport azonosítók, stb.
  - Van egy „igazi” globális PID-je is a processzeknek, de a saját környezetükben más PID-el látszódnak, így megoldható, hogy több 1-es PID-ű processz legyen (init processz)

OS MSc - Virtualizáció

- OS Kernel alapú virtualizáció - **Konténer**
  - A processz tér korlátozható
    - Memória limit
    - Processz szám limit
  - Fájlrendszer is különálló lehet
    - Kvótázható
  - Hálózat
    - Saját IP cím is lehet
    - Az ide tartozó processzek ezt fogják forráscímnek használni
  - CPU megosztás
    - Scheduler módosítások
    - Súlyokkal befolyásolható
  - „Kívülről” az összes processz és erőforrás
    - Látszódik
    - Menedzselhető
  - pl. *Docker* is ezekre épül

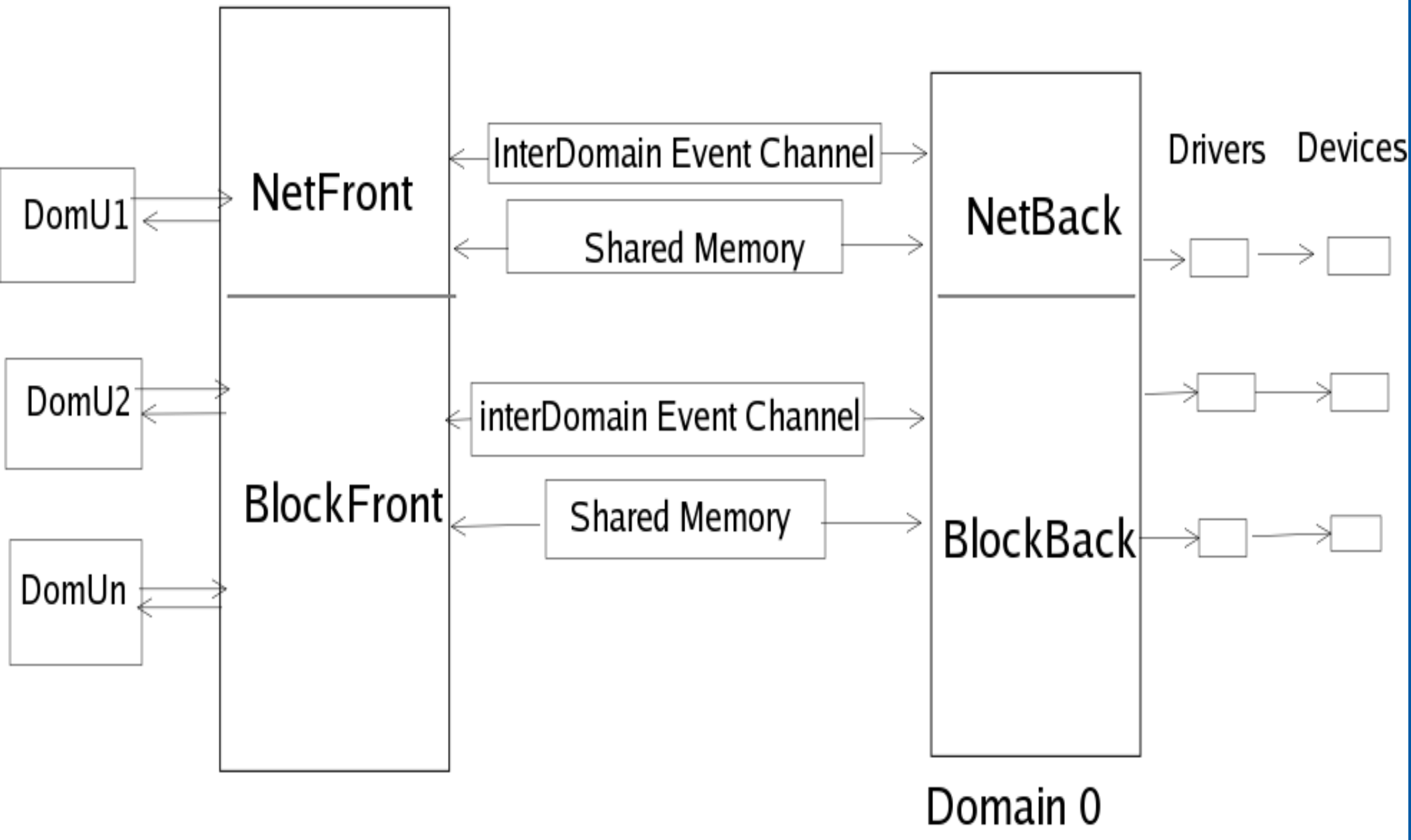
OS MSc – Virtualizáció - XEN

- Virtualizáció - **Paravirtualizáció (XEN)**
  - Külön, teljesen szeparált virtuális gépek futnak
  - Felügyelő program: hypervisor alatt
  - Magát az OS-t is módosítani kell
    - Nyilván az eredeti HW-k nem lesznek közvetlenül elérhetőek
  - Fut egy privilegizált virtuális gép (Domain 0)
    - Ez kezeli az összes hardware-t, rendelkezik az erőforrásokkal
    - Innen fut a management is (létrehozni, erőforrás kiosztás, megállítani, állapot, stb.)
  - A többi virtuális gépnek (Domain U) szolgáltat
    - Virtuális diszke(ke)t
    - Virtuális hálózatot
  - Ezeknek az interfészét implementálni kell az OS-ben
  - Split drivers: Backend (Dom0) – Frontend (DomU)

OS MSc – Virtualizáció - XEN

- Virtualizáció - **Paravirtualizáció (XEN)**
  - Hypervisor megszólítása: hypercall (pl. int 0x82)
  - CPU megosztás: scheduler (több fajta algoritmus is)
  - DomU – Dom0 kommunikáció
  - Frontend block driver
    - A hypervisor segítségével egy osztott memória részt ér el
      - Ezen a Dom0 és az adott DomU osztozik
    - Egy „event channel” van a Dom0 és a DomU-k között
    - Ezen aszinkron módon tudnak kommunikálni
    - Ezen küld egy inter-domain interrupt-ot ha szeretne valamit
    - A Dom0, és az ottani backend driver írja/olvassa az osztott memóriát
    - Végrehajtódik a tényleges diszk művelet
  - Hálózat hasonló elven működik DomU-Dom0 között

Split Drivers Diagram



OS MSc – Virtualizáció - XEN

- Virtualizáció - Paravirtualizáció (XEN)
  - x86 CPU 4 privilégium szintet támogat
    - Ring 0 → Ring 3
      (0: teljes hozzáférés → 3: legkisebb hozzáférés)
    - Igazából csak a ring 0 és ring 3 van használatban a manapság elterjedt OS-ekben.
      - Ring0: OS kernel
      - Ring3: user-space (alkalmazások)
    - Ötlet:
      - Ring0: hypervisor
      - Ring1: OS kernel
      - Ring3: user-space (alkalmazások)
    - Ehhez módosítani kell az OS-t, hogy képes legyen ring1-es privilégium szinten futni. (ring compression)

OS MSc – Virtualizáció - XEN

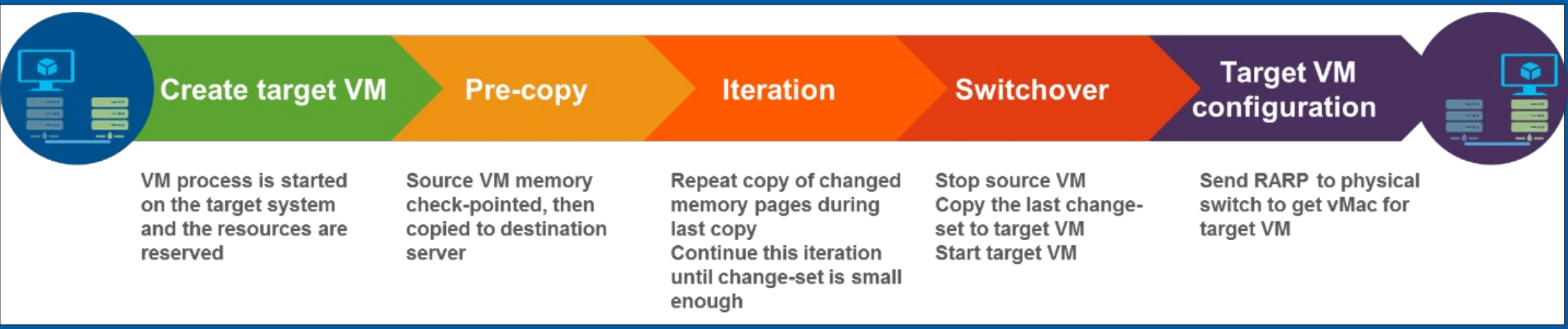
- Virtualizáció
  - Binary translation (VMWare, QEMU)
    - Ha nem megfelelő a privilégium szint és egy másik priv. szintet igénylő utasítás kerül végrehajtásra, akkor „trap” keletkezik
    - Ez detektálható és kezelhető (a hypervisor majd emulálja)
    - De vannak utasítások, amik nem „trap”elhetők
    - Ez akkor gond, ha a program (itt: az OS kernel) úgy gondolja, hogy ring0-ban fut, holott ring1-ben pl.
    - Egy megoldás, arra, hogy az OS módosítás nélkül futhasson:
      - A hypervisor vizsgálja a virtuális memóriát, és ha olyan instrukciókat talál, azokat kicseréli a memóriában, mielőtt még azok végre lennének hajtva
      - Nagyon összetett, de jelentősen jobb teljesítményű, mint a teljes emuláció

Virtualizáció - Live-Migration

- Virtualizáció - Live-migration
  - Menet közben egyik fizikai gépről áthelyezni másik fizikai gépre a virtuális gépet
  - XEN úgy csinálja, hogy:
    - 1. folyamatosan küldi át az A gépről a B gépre a memória lapokat
    - 2. amire közben írás érkezett, akkor azokat még egyszer
    - 3. 2. lépést ismételgeti, amíg el nem ér egy olyan szintet, amikor statisztikai alapon kevés „szennyes” lap (dirty page) marad
    - 4. felfüggeszti a vgépet az A gépen
    - 5. szinkronizálja a maradék memóriát a B gépre
    - 6. processzor állapotot is szinkronizálja
    - 7. startolja a vgépet a B gépen
    - 8. megszünteti az A gépen a vgépet
  - Problémás a storage! (közös storage kell, pl. SAN)

Virtualizáció - Live-Migration

- Virtualizáció – VMware Vmotion



- http://www.mellanox.com/blog/2016/02/set-vmware-vmotion-into-fast-motion-over-high-speed-interconnect/

- *„For VMs that are very active and performing frequent read and write operations, vMotion converges very slowly over 10Gb/s network, but it can succeed over 40Gb/s network with minimal impact on VM read/write operations.”*

OS MSc – Virtualizáció - XEN

- Virtualizáció – IBM LPAR
  - AIX virtualizáció:
    - Live Partition Mobility

      „*Live Partition Mobility is a chargeable feature of IBM POWER6, POWER7 and POWER8 servers, available since 2007, that allows a running LPAR to be relocated from one system to another.*
      In concept, it is similar to VMware VMotion.”

OS MSc – Virtualizáció

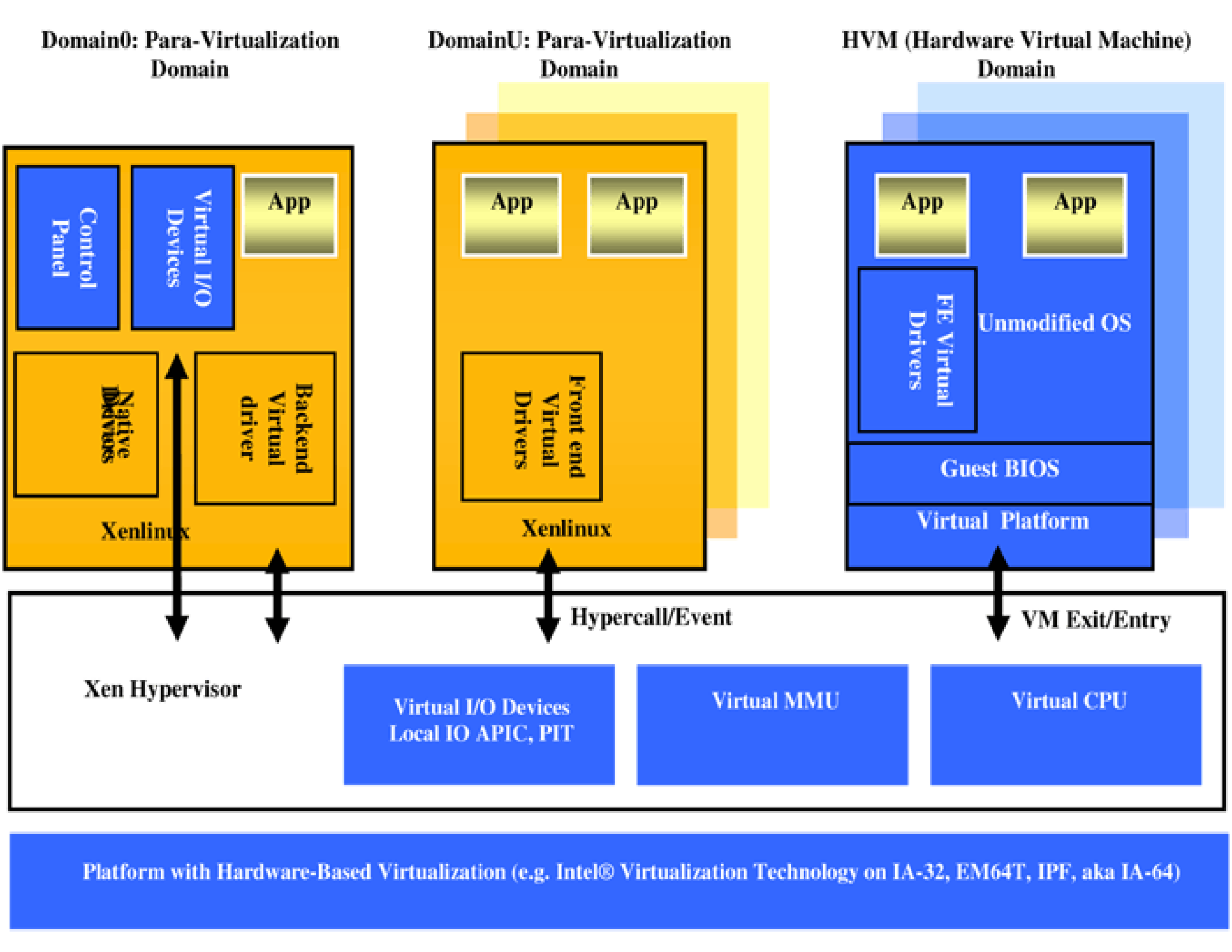
- Virtualizáció – Hypervisor kategorizálás...
  - Type1 hypervisor
    - A hypervisor közvetlen a hardveren fut
  - Type2 hypervisor
    - OS-en belül van a hypervisor (vagy annak megfelelő komponens)

Virtualizáció - Teljes

- **Teljes virtualizáció**
  - Ahány termék, annyi elnevezés, pl.:
    - Hardver asszisztált virtualizáció
    - Gyorsított virtualizáció (accelerated)
    - Natív virtualizáció
    - Teljes értékű környezetet biztosít
    - Magát az OS-t nem kell módosítani
    - Lassabb lehet, mint pl. a paravirtualizáció
    - Hardware támogatás szükséges
      - Intel-VT, AMD-SVM (Pacifica), stb.

Virtualizáció - Teljes

- **Teljes virtualizáció**
  - XEN:
    - HVM – Hardware Virtual Machine
    - BIOS is kell (+ACPI, +MPS tábla, stb. támogatás)
    - Grafikus kártyát (framebuffer) lát maga a DomU, ez kifelé egy VNC-vel hálózaton keresztül elérhető (localhost is lehet persze) virtuális grafikus kártya lesz


Domain0: Para-Virtualization Domain
DomainU: Para-Virtualization Domain
HVM (Hardware Virtual Machine) Domain
Control Panel
Virtual I/O Devices
App
Native Device Drivers
Backend Virtual driver
Xenlinux
App
App
Front end Virtual Drivers
Xenlinux
App
App
FE Virtual Drivers
Unmodified OS
Guest BIOS
Virtual Platform
Hypercall/Event
VM Exit/Entry
Xen Hypervisor
Virtual I/O Devices
Local IO APIC, PIT
Virtual MMU
Virtual CPU
Platform with Hardware-Based Virtualization (e.g. Intel® Virtualization Technology on IA-32, EM64T, IPF, aka IA-64)

Virtualizáció - Hardware

- Virtualizáció
  - Intel VT-x
    - **Cél**: privilege compression-t, binary translation-t, egyebet mellőzni
    - VMX mód (Virtual Machine eXtensions)
    - Új utasítások
    - Új privilégium szint
  - Intel VT-i
    - Itanium (IA64) architektúra kiegészítései
  - Intel VT-d
    - Direct I/O virtualizáció
  - Intel VT-c
    - Connectivity virtualizáció

Virtualizáció - Hardware

- Virtualizáció – Hardware támogatott
  - Intel VT-x
    - Láttuk, hogy problémás ring0-ből ring1-be tenni az OS-t
    - Új privilégium szintet hoztak létre
      - Ring -1
      - Ebben fut a hypervisor
      - A ring0-ring3 megmarad az eredeti felállásra
    - VMX CPU mód
      - VMXON/VMXOFF instrukciók
      - VMX root mód – Hypervisor, teljes jogokkal
      - VMX non-root mód – A vendég OS látszólag teljes jogokkal
      - VM-Entry / VM-Exit – átmenetek a root és non-root mód között
      - VM-Entry: VMLAUNCH és VMRESUME instrukciók
      - VM-Exit: VMEXIT instrukció (és más egyéb implicit módon)

Virtualizáció - Hardware

- Virtualizáció – Hardware támogatott
  - Intel VT-x – VMX mód
    - VMCS – Virtual Machine Control Structure
    - Virtuális gép vezérlő struktúra
    - Csak egy VMCS aktív egy adott időben egy virtuális processzoron
    - Tartalma:
      - VM ki-be lépés vezérlése
      - Vendég állapota (guest-state-area)
        - Exit-kor ebbe mentődik
        - Entry-kor ebből állítja vissza
      - Gazda állapota (host-state-area)
        - Exit-kor ebből állítja vissza
        - Entry-kor ebbe mentődik
      - VM Exit extra információk

Virtualizáció - Hardware

- Virtualizáció – Hardware támogatott
  - Intel VT-x - Processzor virtualizáció
    - A CPU VMX módban kell, hogy fusson
    - **Érzékeny instrukciók speciális kezelése**: CPUID, INVD, MOV CR3-ba,-ból, MOV CR0/CR4-be, RDMSR, WRMSR, HLT, INVLPG, MOV CR8-ból, MOV DR, és MWAIT instrukciók mind VM exit-ként „kapódnak el”
    - Néhány ezek közül konfigurálható, hogy VM exit legyen-e
    - A CR0, CR4, és TSC regiszterek kiolvasása „árnyék” értékeket ad vissza. VM exit-et nem okoz.
    - **Kivételek**, pl. page fault szintén VM exit-ként hajtódnak végre, és VM entry-nél virtuális kivételekként injektálódnak vissza
    - **Külső megszakítások**, amik nem érintik a vgépeket VM exit-et okoznak, és szükség esetén a virtualizált megszakítások VM entry-knél injektálódnak a vgépekbe

Virtualizáció - Hardware

- Virtualizáció – Hardware támogatott
  - Intel VT-x - Memória virtualizáció
    - Még egy lapozási réteg került be
    - GPA – Guest Physical Address
    - MPA/HPA – Machine/Host Physical Address
    - A HVM-ek GPA-kat látnak (de nem tudnak róla)
      - Ezt használják ugyanúgy mintha egy rendes gépen futnának
    - Ezeket a virtuális MMU átfordítja MPA-kra, így jön létre az igazi fizikai cím a memóriában

Virtualizáció - Hardware

- Virtualizáció – Hardware támogatott
  - Intel VT-x - Periféria virtualizáció
    - A XEN HVM DomU-k egy absztrakcióját látják a megszokott PC platformnak:
      - keyboard, mouse, real-time clock, 8259 programmable interrupt controller, 8254 programmable interval timer, CMOS, IDE disk, floppy, CDROM, VGA/graphics
    - HVM-enként egy-egy példány Dom0-ban
    - A teljesítmény szempontjából kritikus „modellek”, mint pl. a Programmable Interrupt Timer (PIT) és a Programmable Interrupt Controller (PIC) a hypervisorban kapnak helyet.
    - Osztott memórián keresztül történik a kommunikáció (vgép és hypervisor között)

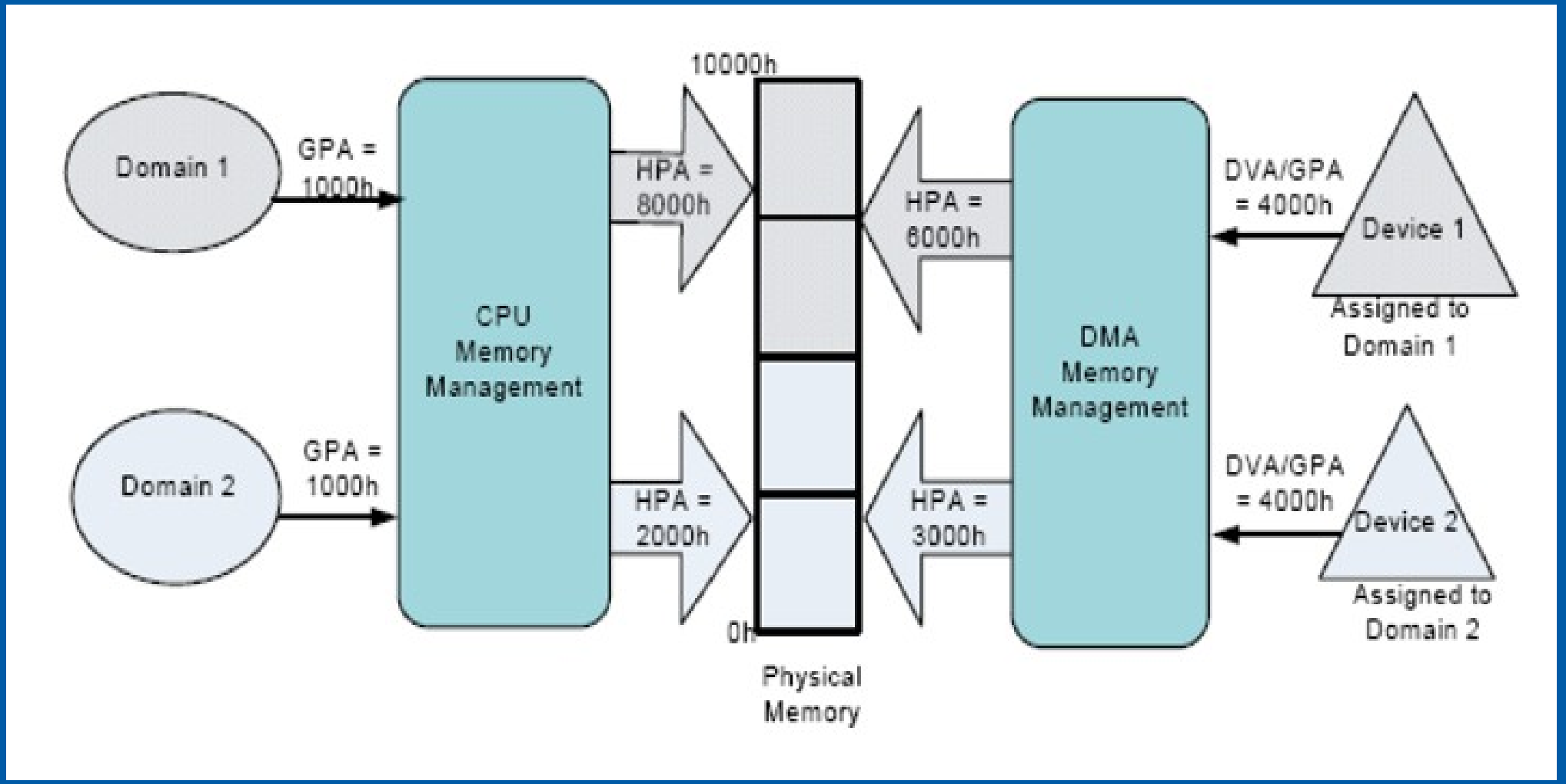
Virtualizáció - Hardware

- Virtualizáció – Hardware támogatott
  - Intel VT-x - Megszakítás virtualizáció
    - Az igazi lokális interrupt vezérlőket (APIC) a hypervisor kezeli. Minden külső megszakítás VM-exit-et okoz.
    - Ami csak a hypervisorra tartozik (pl. timer), azt lerendezi a hypervisor.
    - Ami nem (pl. perifériák), azt a Dom0 handlere (benne futó OS – Linux kernel) fogja kezelni.
    - A HVM DomU-k igazi külső megszakításokat nem kapnak.
    - A HVM DomU-k csak virtuális interruptokat látnak. Ezek a következő VM entry-nél injektálódnak.

Virtualizáció - Hardware

- Virtualizáció – Hardware támogatott
  - Intel VT-d – Direct I/O
    - DMA és IRQ virtualizáció
    - IOMMU már létezik (I/O címeket fordít), de minden sw számára egyformán
    - Chipset-ben van implementálva
    - Protection domain-ek kialakítása
      - Definiálható melyikből mi érhető el
      - I/O eszközök felől sem érhető el a védett rész
      - (DMA izoláció)
      - DMA virtual address (DVA)
      - Leképzés a címek között

Címleképzés GPA → HPA



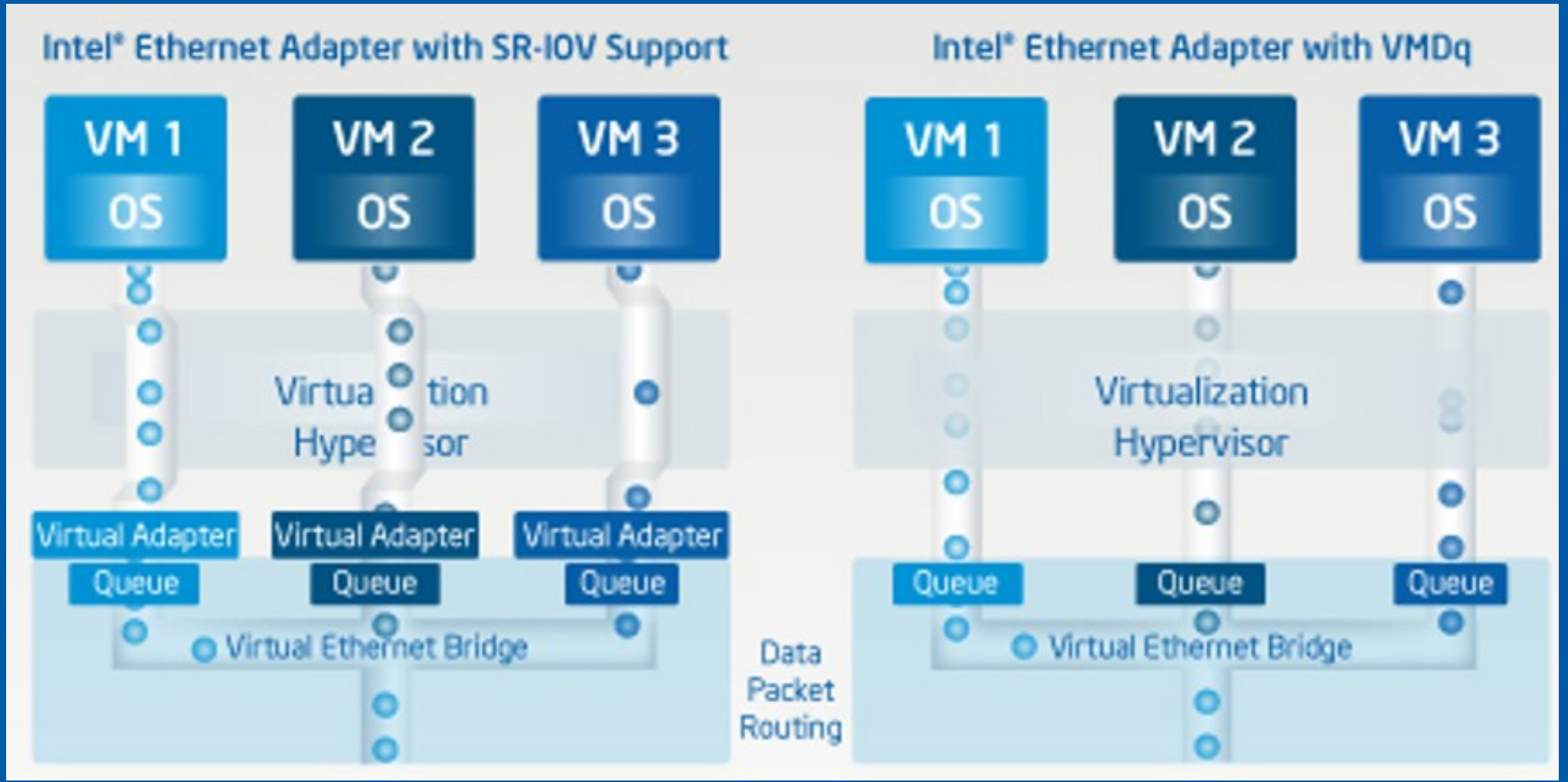
Virtualizáció - Hardware

- Virtualizáció – Hardware támogatott
  - Intel VT-d – Direct I/O
    - Eszköz mappelés protection domain-be
      - Azonosítás PCI ID alapján (Bus/Device/Function)
        - Root-Entry tábla:
          - Bus ID alapján tartalmazza a Context-Entry táblákra mutató pointereket
        - Context-Entry tábla: (több is lehet egy ID-hez)
          - Ezek tartalmazzák a Dev/Func eszközhöz tartozó DMA címfordításhoz szükséges struktúrákra mutató pointereket
        - Ennek a bejárását a HW elvégzi
        - Ez alapján dönti el, hogy milyen R/W jogok lépjenek életbe

Virtualizáció - Hardware

- Virtualizáció – Hardware támogatott
  - Intel VT-c – Connectivity
    - Periférián implementált feature
    - Leveszi a terhet a Hypervisorrol a periféria megosztásban
    - Pl. hálózati adapternél, több várakozó sor, több virtuális I/O port, mind magában a hálózati kártya vezérlőben
    - MAC/VLAN Tag alapján szétszedés
    - Természetesen a meghajtóprogramoknak is támogatni kell
    - Near-native teljesítmény
    - Nagy I/O használatnál térülhet meg
    - Reklám: 4Gbit/sec → 9.2 Gbit/sec (9.5 Jumbo)

Virtualizáció - Hardware

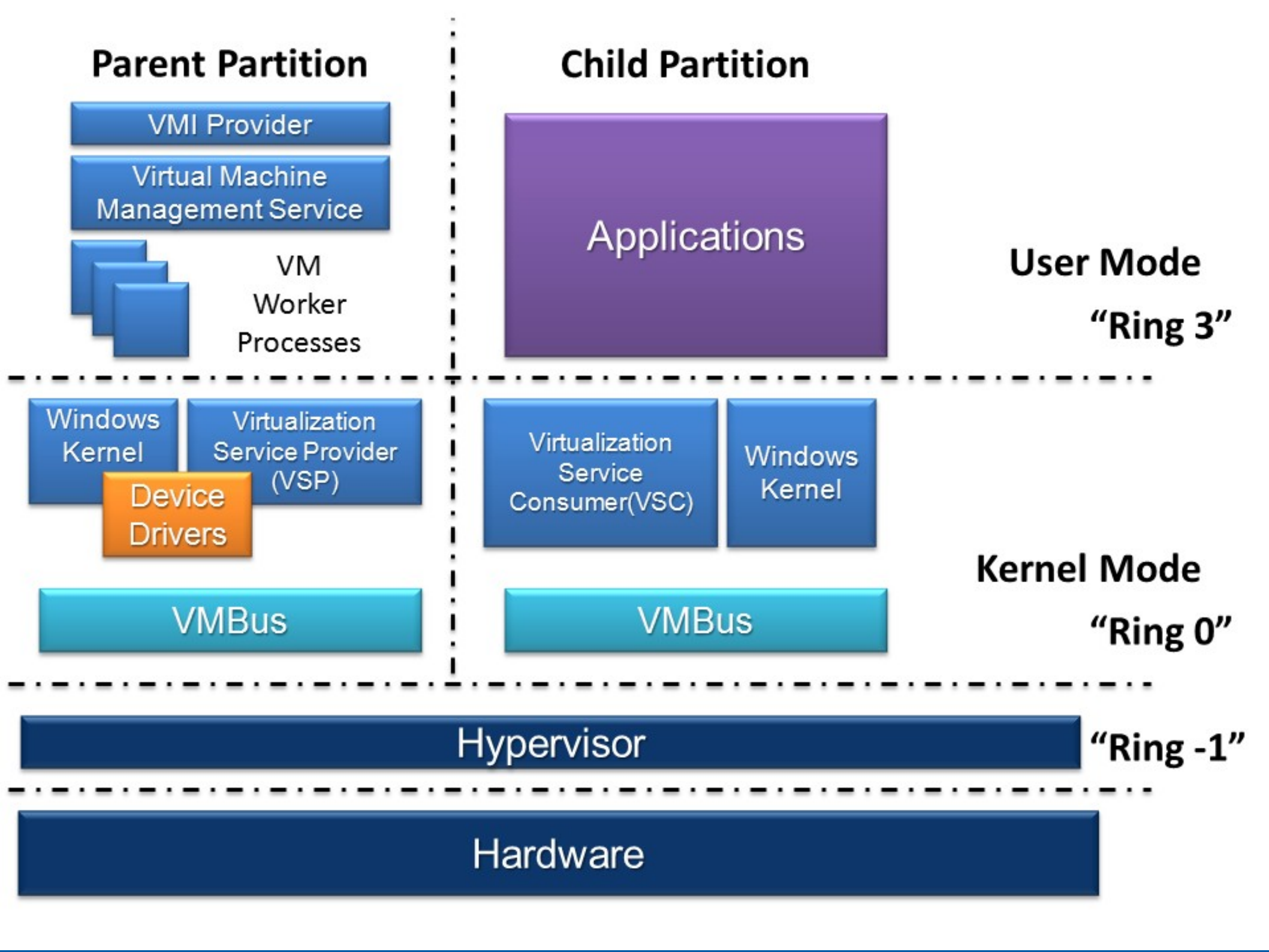


Virtualizáció - Hardware

- Virtualizáció – Hardware támogatott
  - AMD SVM - Pacifica
    - Az AMD hardware-es virtualizációs támogatás
    - Nagyon hasonló a VT-x -hez
    - Az elv ugyanaz
    - A VM leíró méret is megegyezik
    - Mások kicsit az elnevezések
    - „World switch”, Virtual Machine Control Block, stb.
    - Illetve az utasítások
      - VMRUN, VMSAVE, VMLOAD, stb.

Operációs Rendszerek MSc

- Virtualizáció
  - Hyper-V
    - Microsoft virtualizációs megoldása
      - Tud paravirtualizálni
      - Tud hardware támogatott virtualizálni
    - Nagyon hasonló az architektúrája, mint a Xen architektúrája
    - Lássuk egy ábrán


Parent Partition
Child Partition
VMI Provider
Virtual Machine Management Service
VM Worker Processes
Applications
User Mode
"Ring 3"
Windows Kernel
Virtualization Service Provider (VSP)
Device Drivers
Virtualization Service Consumer(VSC)
Windows Kernel
Kernel Mode
"Ring 0"
VMBus
VMBus
Hypervisor
"Ring -1"
Hardware

Operációs Rendszerek MSc

- Virtualizáció
  - KVM – Kernel-based Virtual Machine
    - Kész OS-t változtat hypervisor-rá (Linux)
    - Mindenképpen HW virtualizációt használ
    - Egy processzként fut a gazda OS-en a vendég OS
    - Az ütemezését így a Linux kernel ütemezője végzi
    - A memóriakezelést is a Linux kernel végzi
      - Az egész vgép memóriája a processzhez tartozik
      - Akár swappelhető is...
      - NUMA támogatás
      - Kernel Same Page Merging (KSPM) – CoW

Operációs Rendszerek MSc

- Virtualizáció
  - KVM – Kernel-based Virtual Machine
    - Maga a Linux kernel válik hypervisor-rá
      - Live-Migration képes
      - Vendég OS-ek:
        - Linux, OpenBSD,FreeBSD, OpenSolaris, Solaris, MS-DOS!
      - Hybrid virtualizáció támogatás

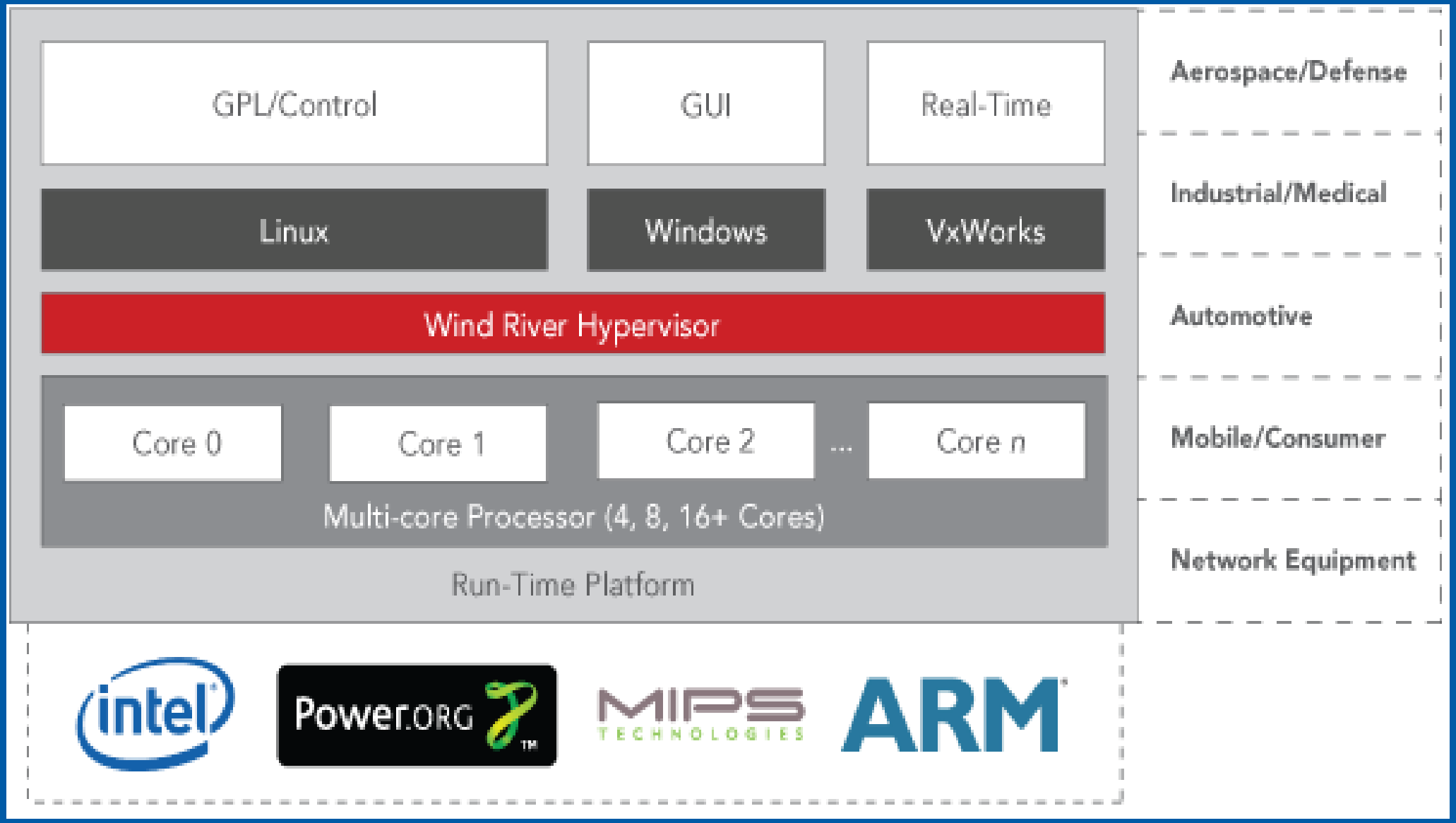
Operációs Rendszerek MSc

- Hibrid virtualizáció
  - **Hybrid virtualization**
    - Igaz lehet módosítatlan OS-t futtatni, de érdemes lehet kikerülni a költséges emulációkat
    - Teljes virtualizációnál is lehet használni a paravirtualizálásnál megismert backend-frontend virtuális diszk/network megoldást
    - pl. XEN + Windows + paravirt drivers
    - Lásd még: XEN PVHVM, XEN PVH
    - Teljesítménynövekedés!

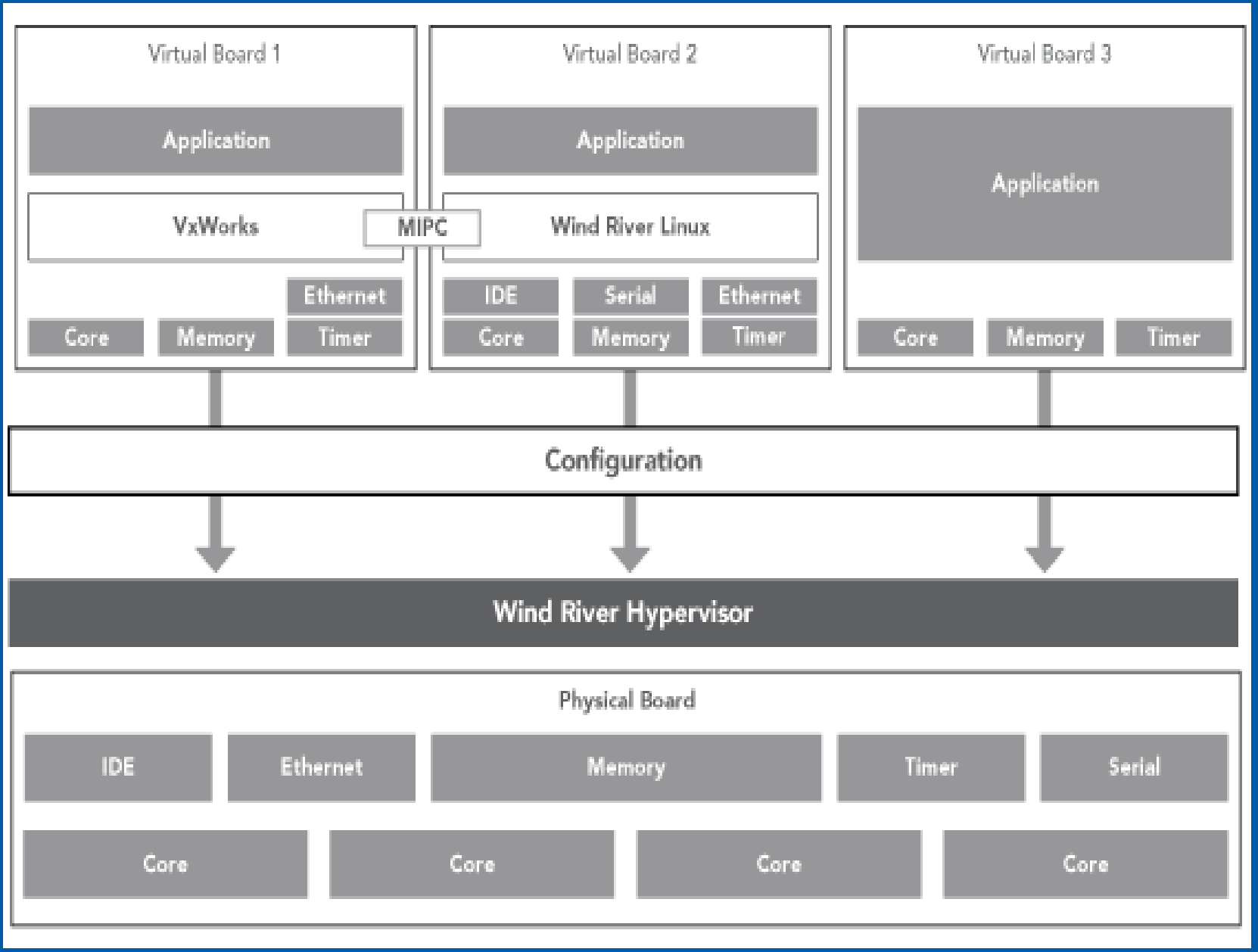
Operációs Rendszerek MSc

- Beágyazott virtualizáció
  - Wind-River Hypervisor
    - Beágyazott rendszerekhez
    - Kicsi
    - Real-time + determinisztikus
    - Alacsony késleltetés
    - Real-time, illetve „biztonsági” rendszerekhez
    - Paravirtualizáció és teljes virtualizáció
    - Eszközhozzáférés lehet direkt (pass-thru), vagy virtualizált
    - Számos architektúrát támogat

Wind-River Hypervisor



Wind-River Hypervisor



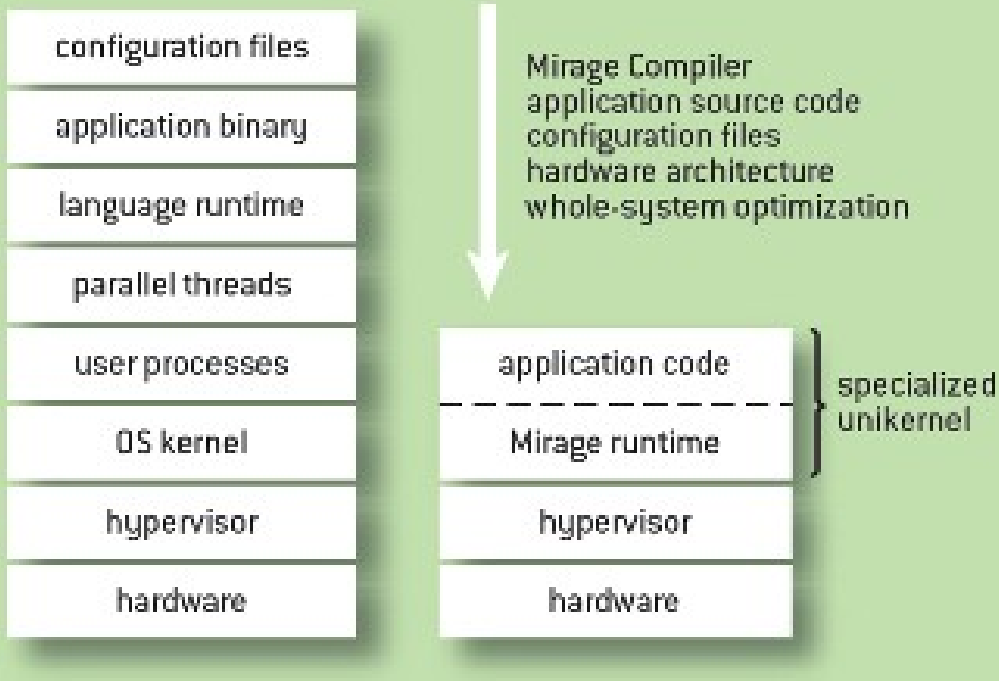
Operációs Rendszerek MSc

- **Unikernel**
  - MirageOS – XEN alá
  - Virtual Library Operating System
  - A legtöbb VM egy-egy dedikált feladatot lát el
  - Spóroljuk le a „felesleges” rétegeket
  - Integráljuk össze az application kódját a kernellel!
  - Ez fusson csak a VM-ben
  - Gyors boot, jóval kisebb erőforrás használat (mem,disk,cpu)
  - Biztonság? Frissítés?
  - OCaml nyelv

Operációs Rendszerek MSc

FIGURE 1

Enterprise Component for A Highly Re-configurable Architectural Style



Operációs Rendszerek MSc

- Phone
  - Dual-SIM
  - Stabilitás
  - Private-Corporate
  - („VMWare dropped the project”...)
- Autóipar
  - ECU (Electronic Control Unit) konszolidáció
  - Izoláció
  - Komponens újrahasznosítás

Operációs Rendszerek MSc

- Virtualizáció
  - Popek és Goldberg virtualizációs követelmények
  - A virtualizáció követelményei, 1974
  - **1. tétel:**
    - Bármely 3. generációs számítógéphez hatékony VMM (hypervisor) készíthető, ha az érzékeny instrukciók halmaza a privilegizált instrukciók halmazának részhalmaza.
  - **2. tétel:**
    - Bármely 3. generációs számítógép rekurzívan virtualizálható, ha:
      - 1. virtualizálható és
      - 2. időzítés függőség nélküli VMM írható rá.

Operációs Rendszerek MSc

- Felhő (cloud computing)
  - Főként marketing...
  - Alapja a virtualizáció
  - Mindegy, hogy hol fut (fizikailag), nem számít, csak szolgáltatást kap a felhasználó
  - Könnyen skálázható
    - Erőforráshalmaz könnyed növelése
    - Vagy csökkentése (spórolás)
  - Szolgáltatások:
    - Infrastruktúra (Iaas)
    - Platform (Paas)
    - Szoftver (Saas)
    - Egyéb nézőpont... (XaaS)

Operációs Rendszerek MSc

- Felhő (cloud computing)
  - Szoftver (Software as a Service)
    - Kapunk egy kész szoftvert
      - webes interfész kézenfekvő
    - Nem kell lefejleszteni
    - Nem kell megvenni, licencelni (benne van)
    - Mindegy, hogy hol fut, nem számít
      - (eddig számított hol fut a facebook.com? :))
    - Pl. webáruház, ügyviteli sw, ticket rendszer, email rendszer, stb. (Google Apps, Office365, stb.)
    - Skálázható
      - Több user kell
      - Nagyobb adatbázis
      - Stb.

Operációs Rendszerek MSc

- Felhő (cloud computing)
  - Platform (Platform as a Service)
    - Egyel „lentebbi” szint mint a SaaS
    - Egy platformot kapunk a saját fejlesztésű alkalmazásunknak
    - Szoftver és hardver management-et a szolgáltató végzi
    - Nem teljes szabadságot
    - pl. Docker, installált konfigurált OS, web szerver, DB kiszolgáló, fejlesztői környezet stb.
      (De nem feltétlen van teljes hozzáférésünk!)

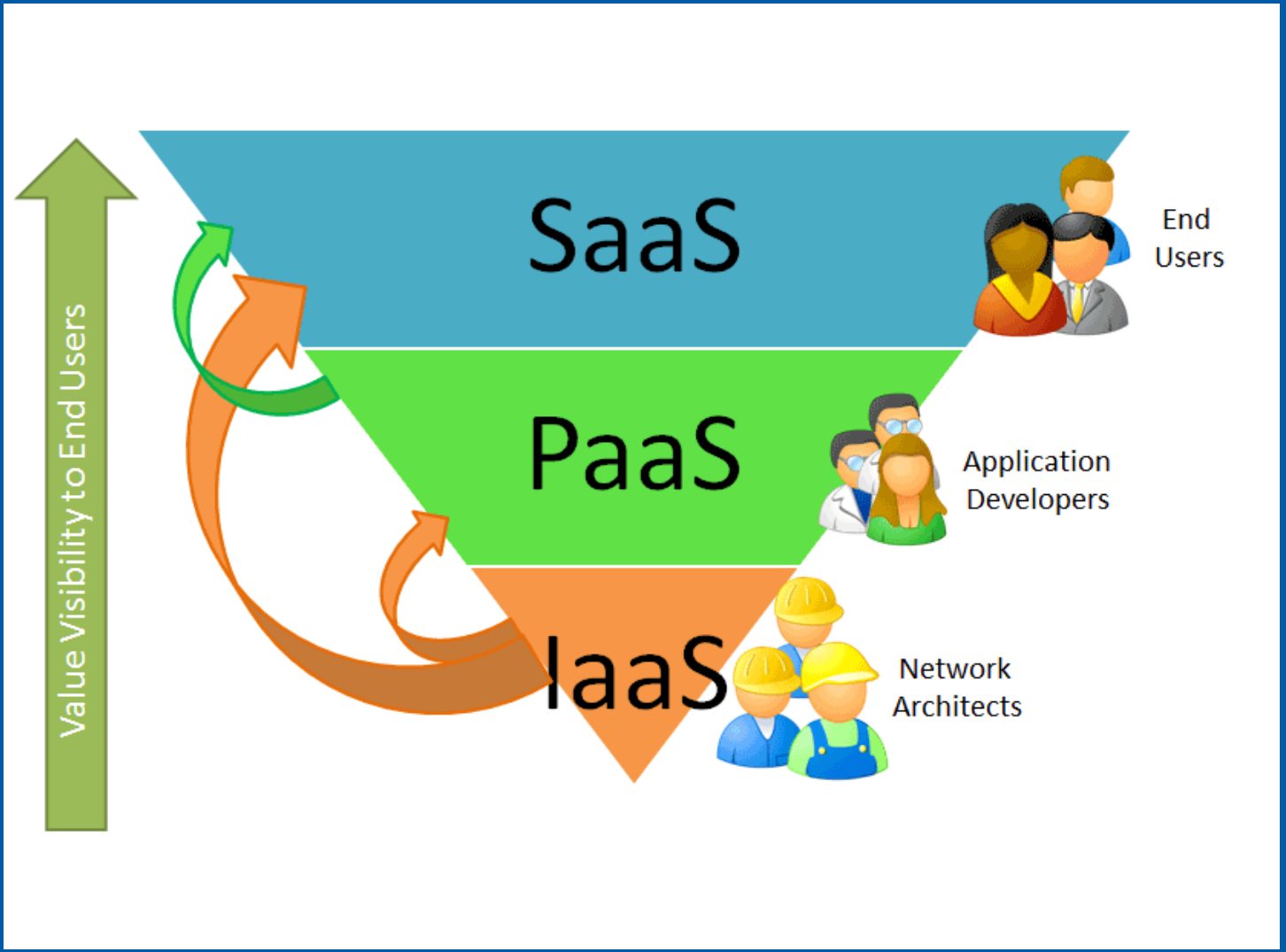
    - Gondoljunk csak bele eddig mi is volt a webhosting?
      - Most ez miben lesz más?

Operációs Rendszerek MSc

- Felhő (cloud computing)
  - Infrastruktúra (Infrastructure as a Service)
    - Egy vagy több komplett virtuális gépet kapunk
      - Azt teszünk rá, amit szeretnénk
      - Mi manageljük az egészet
      - A gép erőforrásait mi osztjuk be
    - Több gép esetén izgalmasabb
      - Lehet saját kis hálózatunk
      - Terheléselosztónk
      - Replikáció, backup
      - Stb.
    - Tehát szervert, hálózatot ad
    - Eddig ez hogy volt?

Operációs Rendszerek MSc

- Felhő (cloud computing)



Operációs Rendszerek MSc

- Porfelhő (dust cloud)
  - Unikerneles VM-ek összessége
  - pl. MirageOS
  - on demand (mivel gyors a bootidő: <1sec)

- Köd… :
  - Van helyi infrastruktúra is, pl. olyan alkalmazások, amik nagy adatforgalmat igényelnek (pl. helyi videostreamek feldolgozása)
  - A többi, ami nem ilyen, pedig „a felhőben”