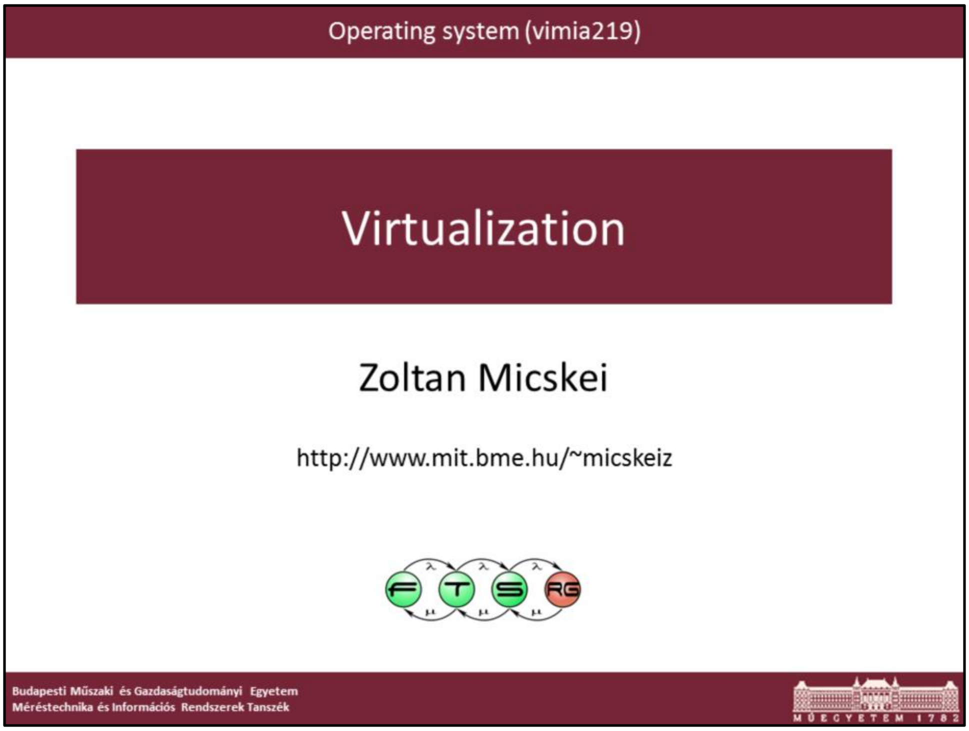
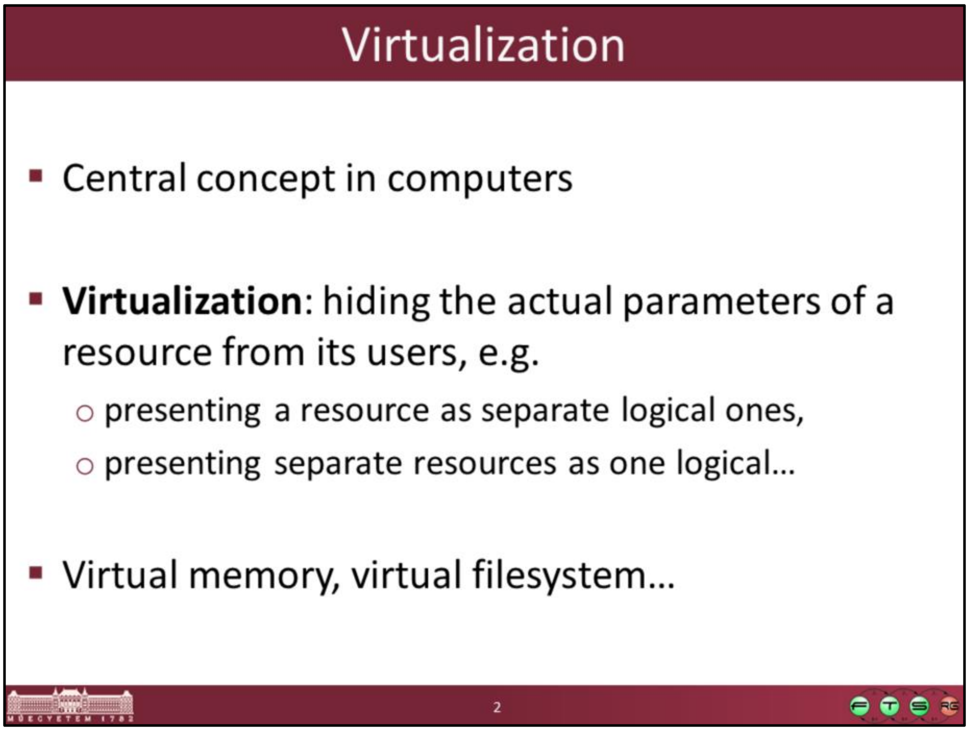


# Virtualization

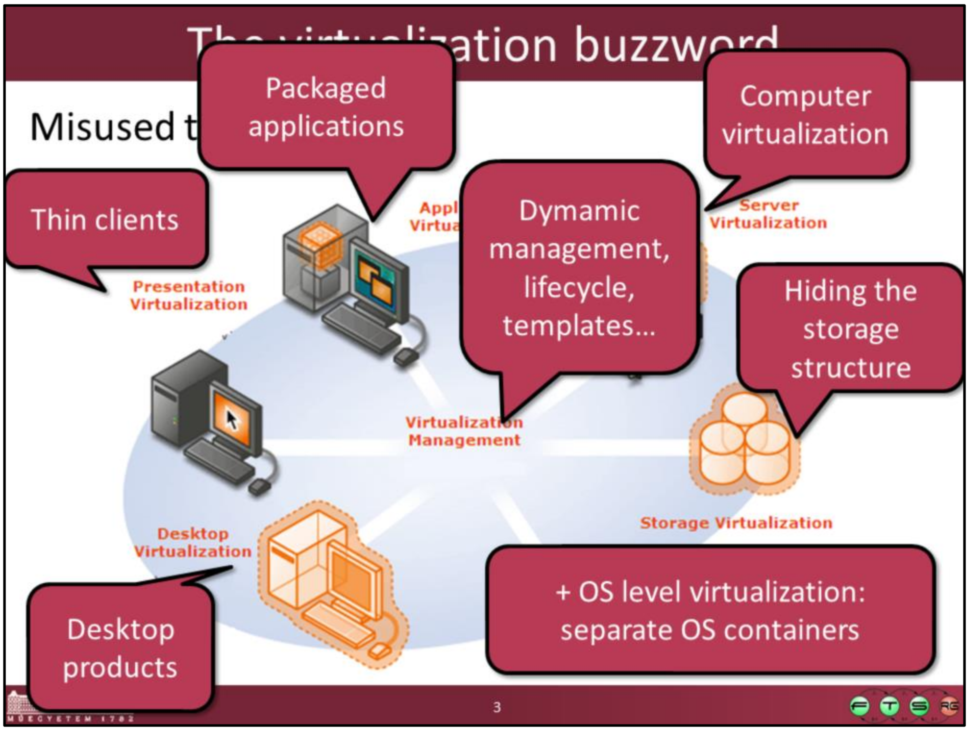
Zoltan Micskei

http://www.mit.bme.hu/~micskeiz



# Virtualization

- Central concept in computers
- **Virtualization**: hiding the actual parameters of a resource from its users, e.g.
  - ○ presenting a resource as separate logical ones,
  - ○ presenting separate resources as one logical…
- Virtual memory, virtual filesystem…



## The virtualization buzzword

Misused t

Packaged applications

Computer virtualization

Thin clients

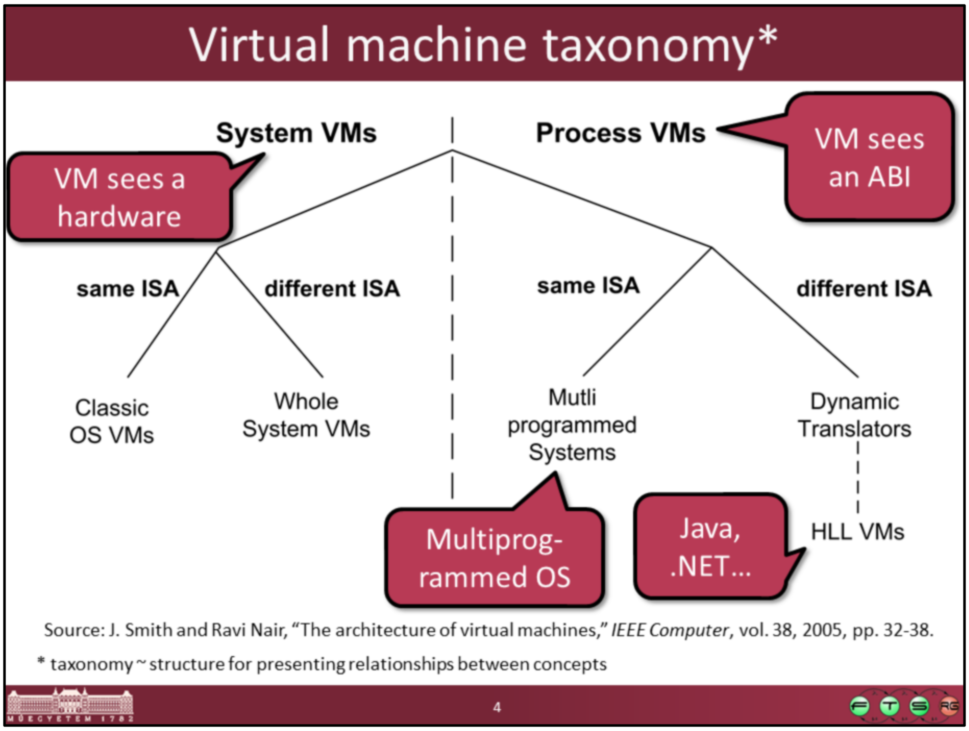
Dymamic management, lifecycle, templates…

Hiding the storage structure

+ OS level virtualization: separate OS containers

Desktop products

Source: http://www.microsoft.com/virtualization/default.mspx



# Virtual machine taxonomy*

System VMs

Process VMs

VM sees a hardware

VM sees an ABI

same ISA

different ISA

same ISA

different ISA

Classic OS VMs

Whole System VMs

Mutli programmed Systems

Dynamic Translators

HLL VMs

Multiprog-rammed OS

Java, .NET…

Source: J. Smith and Ravi Nair, “The architecture of virtual machines,” *IEEE Computer*, vol. 38, 2005, pp. 32-38.

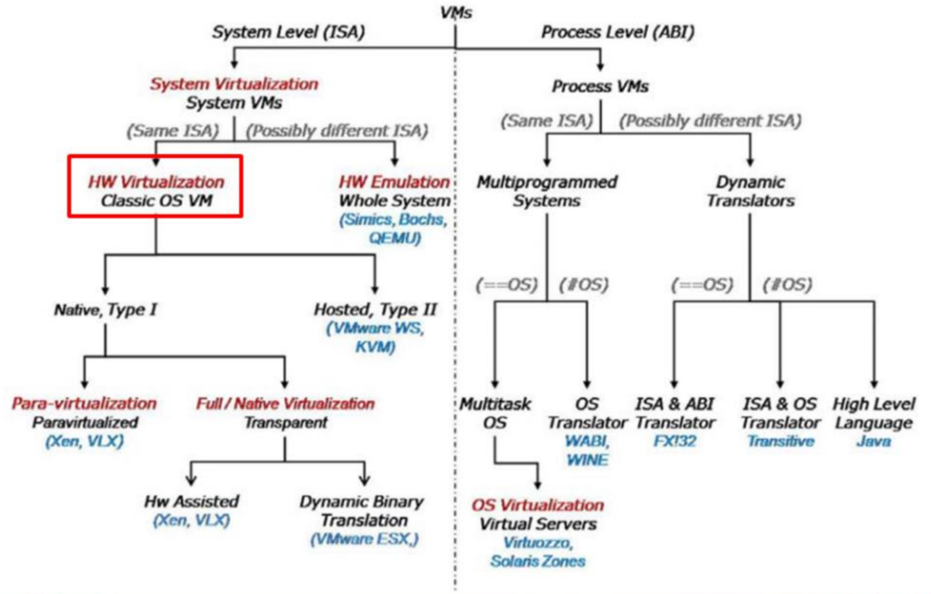
* taxonomy ~ structure for presenting relationships between concepts

Source: J. Smith and Ravi Nair, “The architecture of virtual machines,” *IEEE Computer*, vol. 38, 2005, pp. 32-38.
http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1430629

**Process VM**: „A process VM is a virtual platform that executes an individual process. This type of VM exists solely to support the process; it is created when the process is created and terminates when the process terminates.”
**System VM**: „A system VM provides a complete, persistent system environment that supports an operating system along with its many user processes. It provides the guest operating system with access to virtual hardware resources, including networking, I/O, and perhaps a graphical user interface along with a processor and memory. ”

ISA: Instruction Set Architecture
ABI: Application Binary Interface
API: Application Programming Interface

# Virtual machine taxonomy (detailed)



VMs

System Level (ISA)

System Virtualization
System VMs

(Same ISA)

HW Virtualization
Classic OS VM

Native, Type I

Para-virtualization
Paravirtualized
(Xen, VLX)

Full / Native Virtualization
Transparent

Hw Assisted
(Xen, VLX)

Dynamic Binary
Translation
(VMware ESX,)

Hosted, Type II
(VMware WS,
KVM)

(Possibly different ISA)

HW Emulation
Whole System
(Simics, Bochs,
QEMU)

Process Level (ABI)

Process VMs

(Same ISA)

Multiprogrammed
Systems

(==OS)

Multitask
OS

OS Virtualization
Virtual Servers
Virtuozzo,
Solaris Zones

(≠OS)

OS
Translator
WABI,
WINE

(Possibly different ISA)

Dynamic
Translators

(==OS)

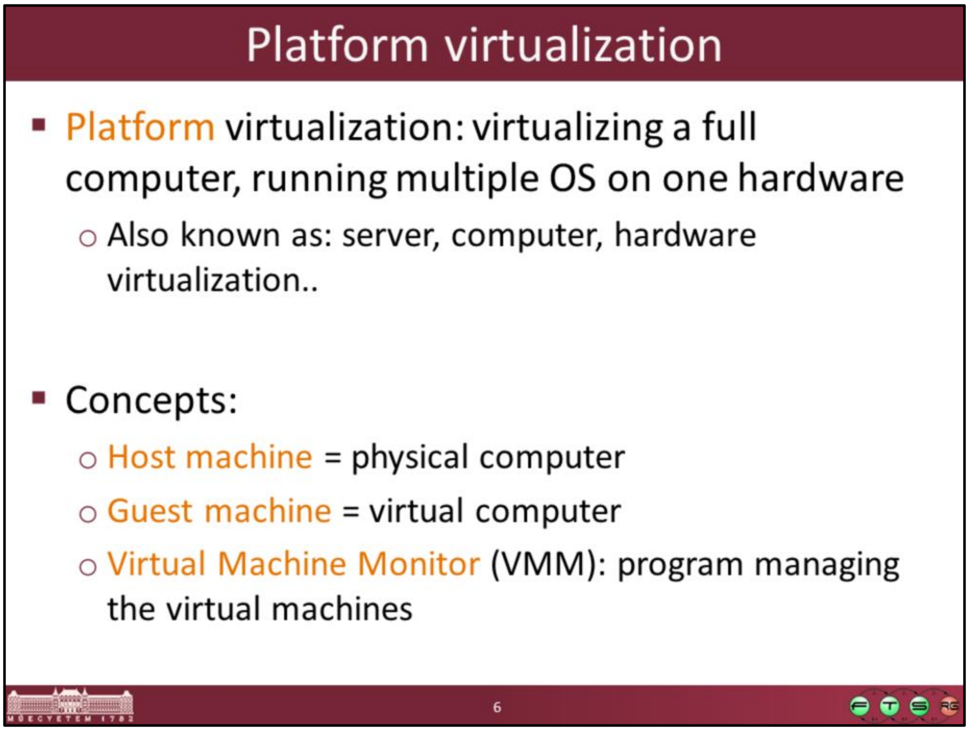
ISA & ABI
Translator
FX!32

(≠OS)

ISA & OS
Translator
Transitive

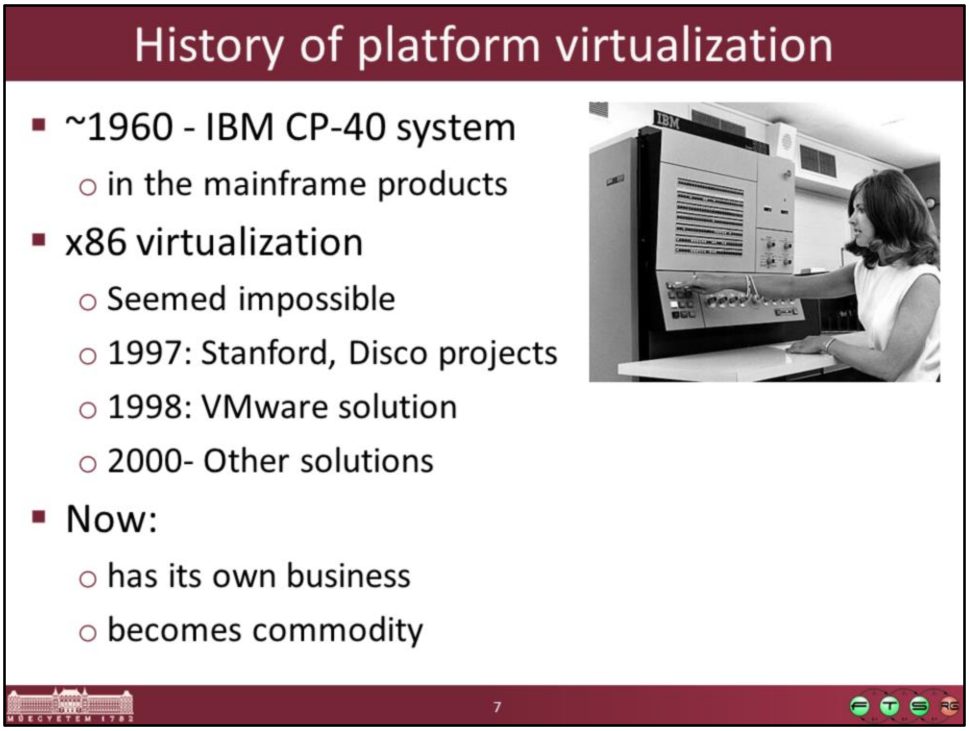
High Level
Language
Java

Source: Scope Alliance, Virtualization: State of the Art, 2008.
http://scope-alliance.org/sites/default/files/documents/SCOPE-Virtualization-StateofTheArt-Version-1.0.pdf



# Platform virtualization

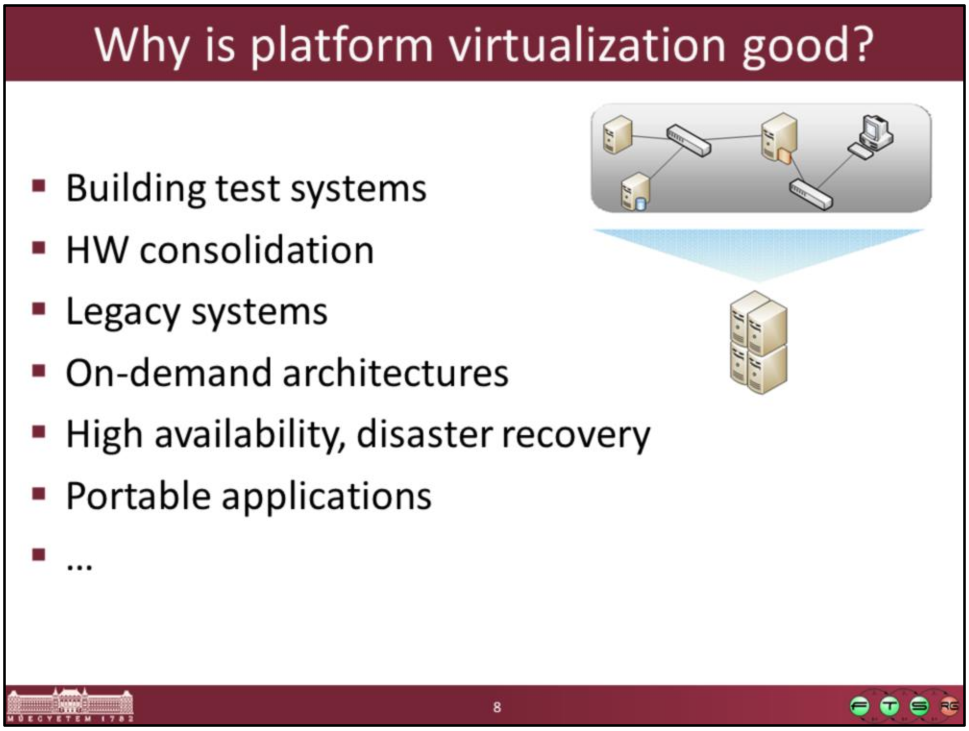
- Platform virtualization: virtualizing a full computer, running multiple OS on one hardware
  - Also known as: server, computer, hardware virtualization..
- Concepts:
  - Host machine = physical computer
  - Guest machine = virtual computer
  - Virtual Machine Monitor (VMM): program managing the virtual machines



## History of platform virtualization

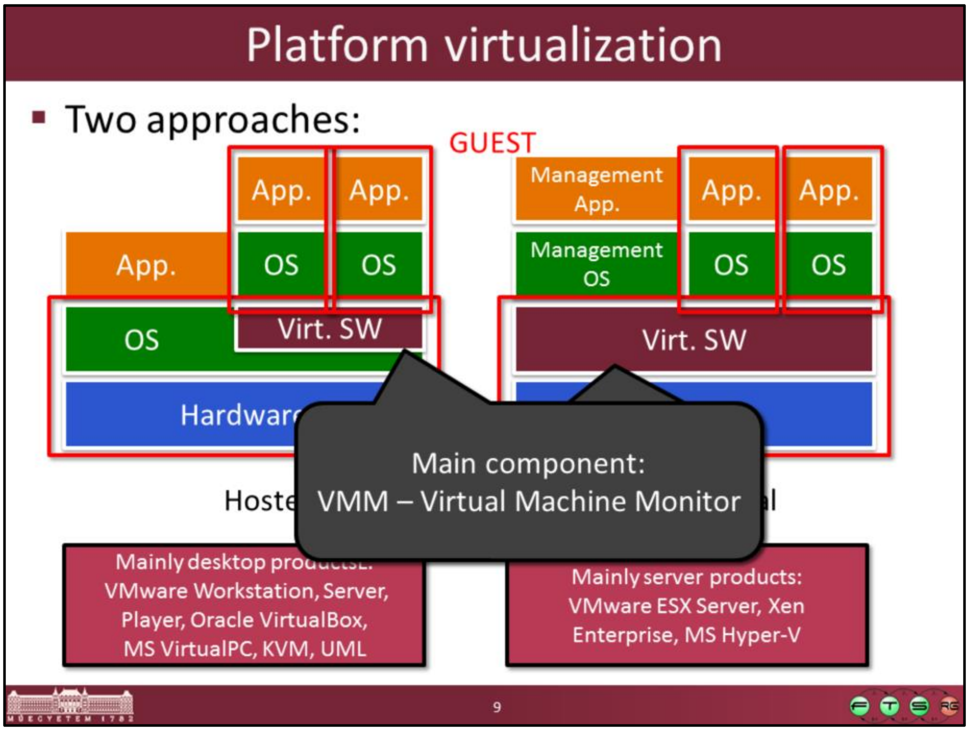
- ~1960 - IBM CP-40 system
  - in the mainframe products
- x86 virtualization
  - Seemed impossible
  - 1997: Stanford, Disco projects
  - 1998: VMware solution
  - 2000- Other solutions
- Now:
  - has its own business
  - becomes commodity

Source: IBM Mainframes reference room
http://www-03.ibm.com/ibm/history/exhibits/mainframe/mainframe_room.html



## Why is platform virtualization good?

- Building test systems
- HW consolidation
- Legacy systems
- On-demand architectures
- High availability, disaster recovery
- Portable applications
- …



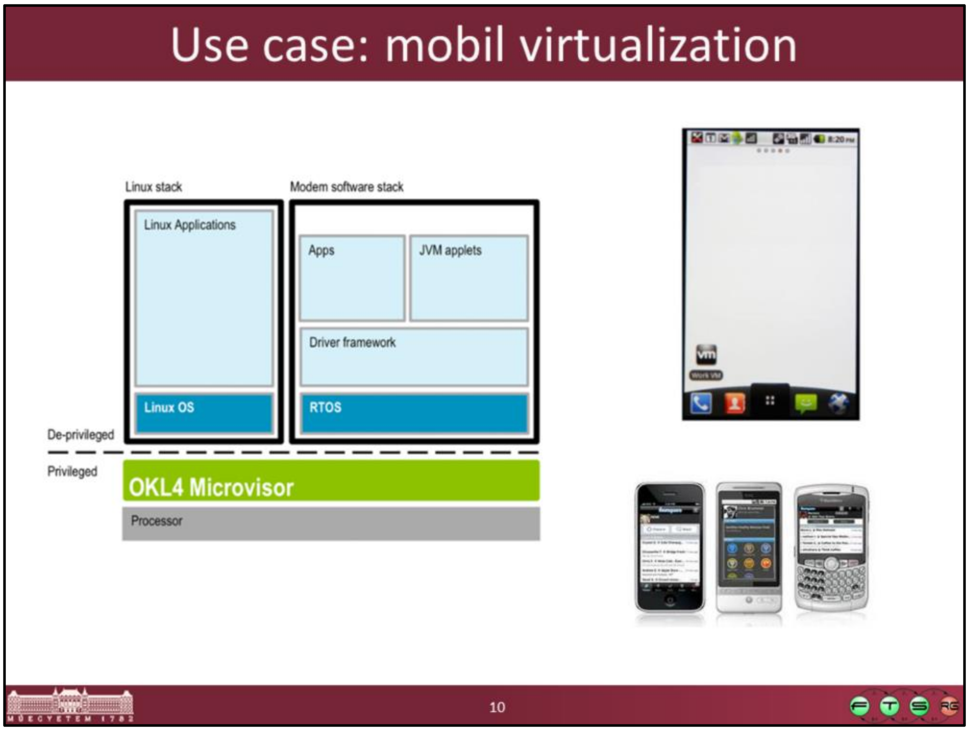
# Platform virtualization

- Two approaches:

Main component:
VMM – Virtual Machine Monitor

Mainly desktop productsL:
VMware Workstation, Server,
Player, Oracle VirtualBox,
MS VirtualPC, KVM, UML

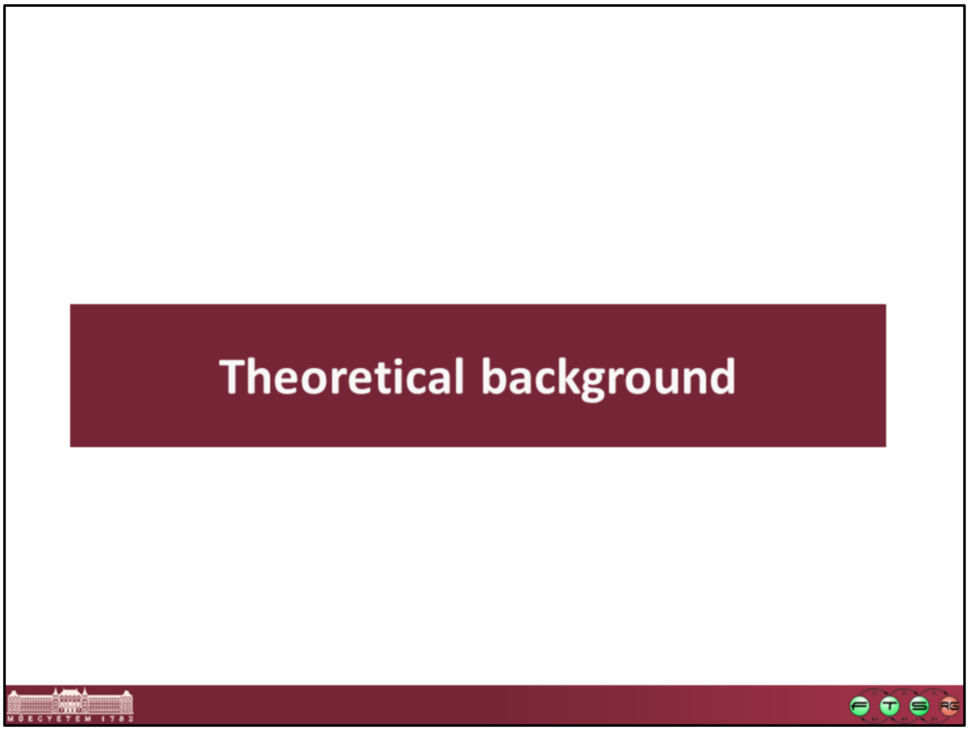
Mainly server products:
VMware ESX Server, Xen
Enterprise, MS Hyper-V



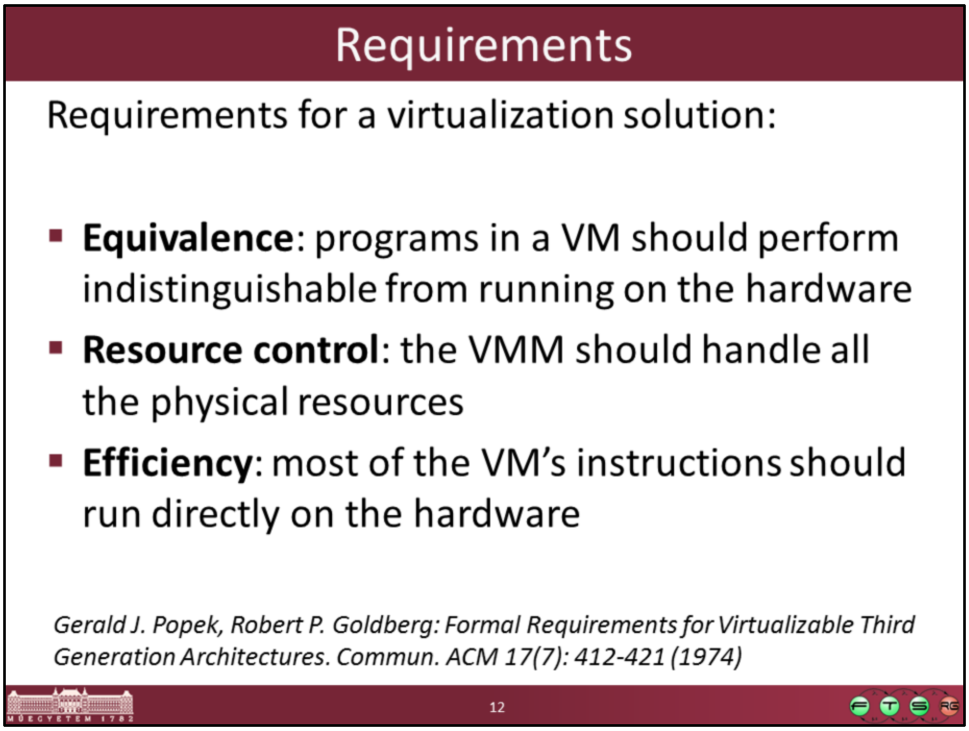
# Use case: mobil virtualization

Sources:

- Left: http://www.ok-labs.com/solutions/what-is-mobile-phone-virtualization
- Right: http://mobiputing.com/2010/12/vmware/



# Theoretical background

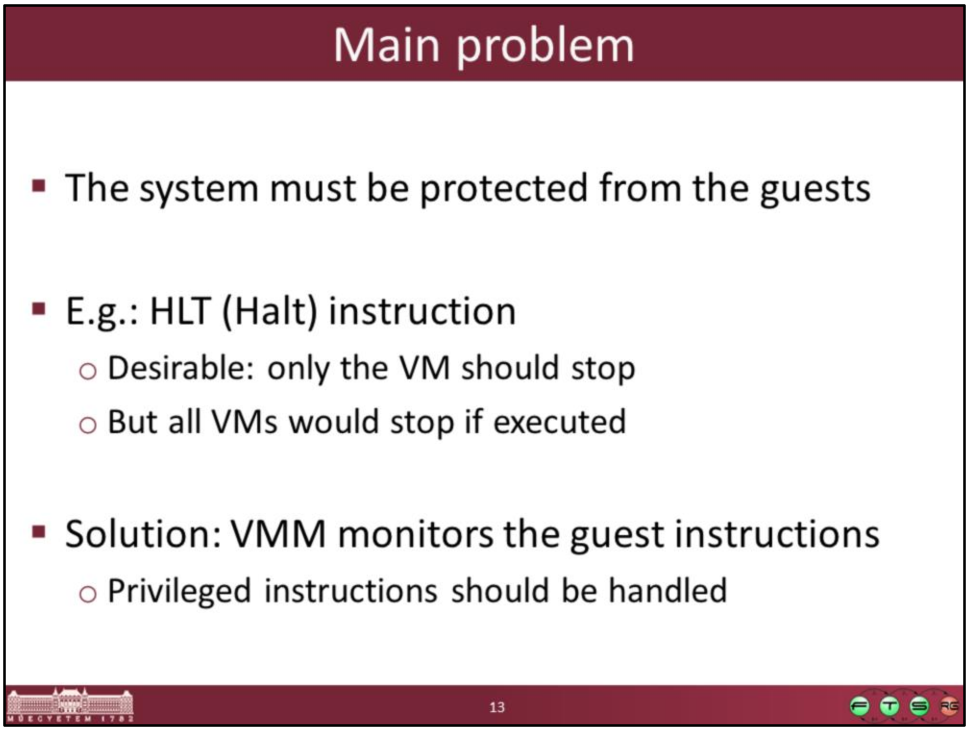


# Requirements

Requirements for a virtualization solution:

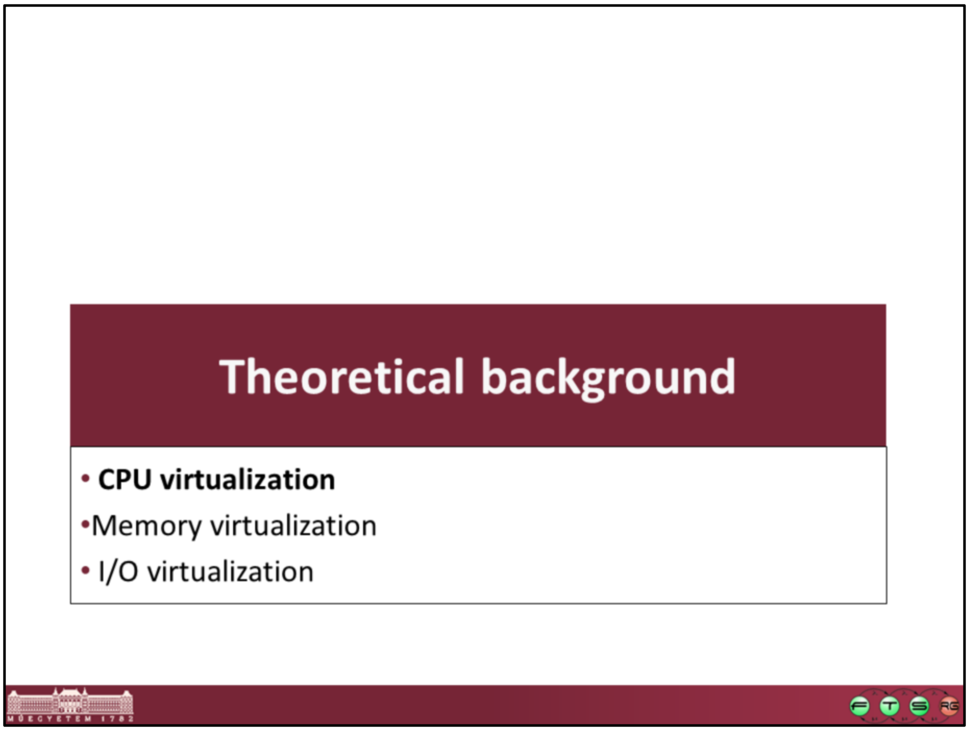
- **Equivalence**: programs in a VM should perform indistinguishable from running on the hardware
- **Resource control**: the VMM should handle all the physical resources
- **Efficiency**: most of the VM's instructions should run directly on the hardware

*Gerald J. Popek, Robert P. Goldberg: Formal Requirements for Virtualizable Third Generation Architectures. Commun. ACM 17(7): 412-421 (1974)*



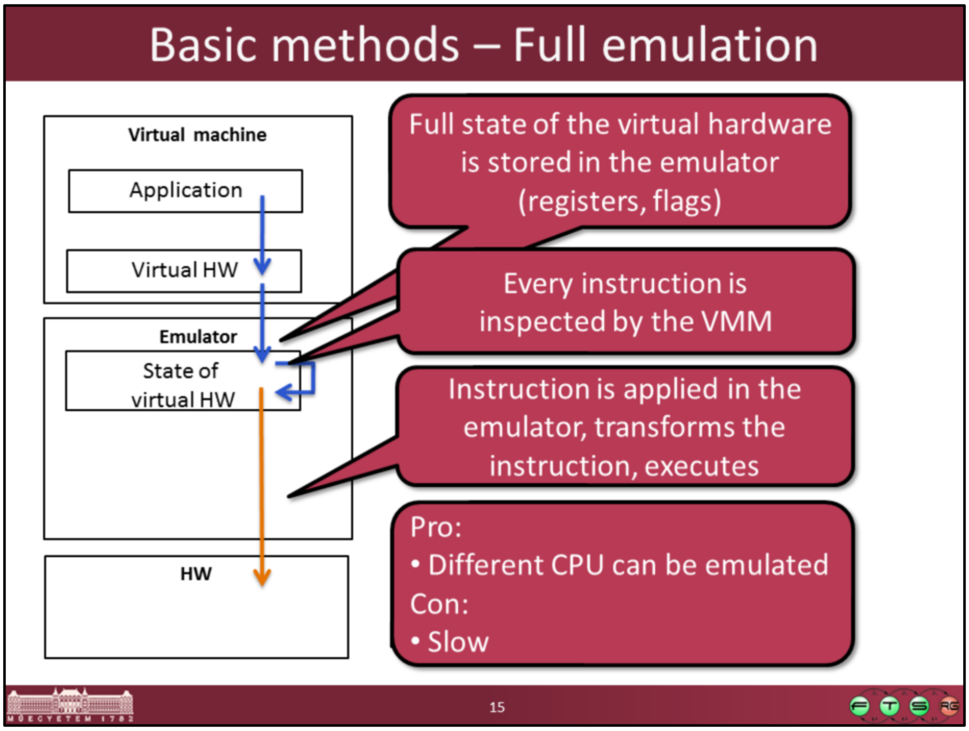
# Main problem

- The system must be protected from the guests
- E.g.: HLT (Halt) instruction
  - Desirable: only the VM should stop
  - But all VMs would stop if executed
- Solution: VMM monitors the guest instructions
  - Privileged instructions should be handled



# Theoretical background

- **CPU virtualization**
- Memory virtualization
- I/O virtualization



# Basic methods – Full emulation

Full state of the virtual hardware is stored in the emulator (registers, flags)

Every instruction is inspected by the VMM

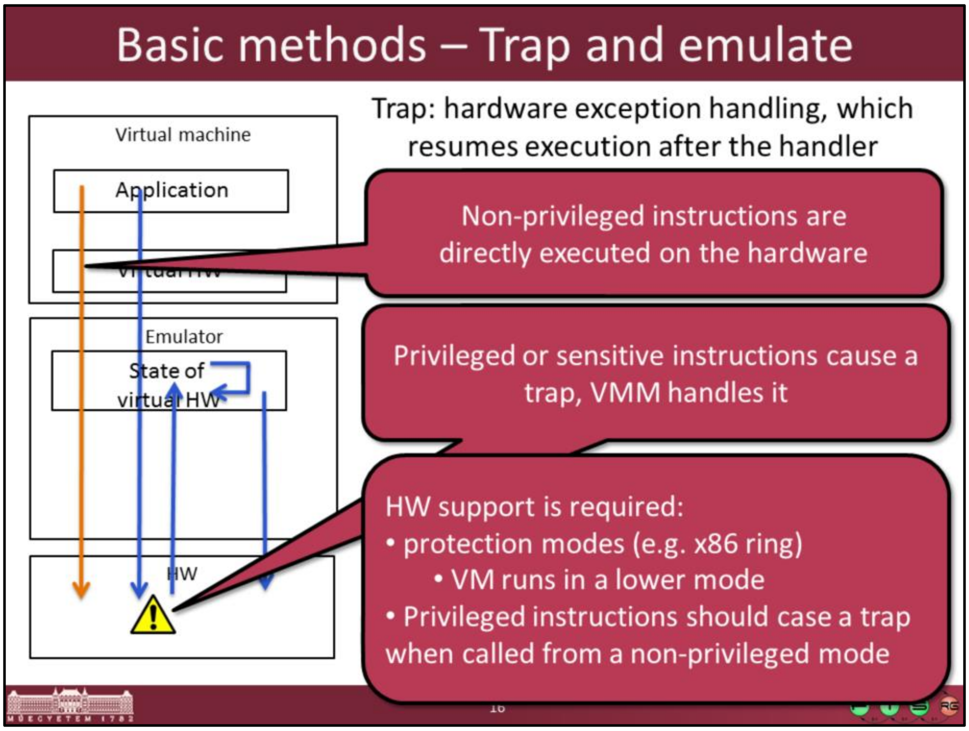
Instruction is applied in the emulator, transforms the instruction, executes

Pro:

- Different CPU can be emulated

Con:

- Slow



# Basic methods – Trap and emulate

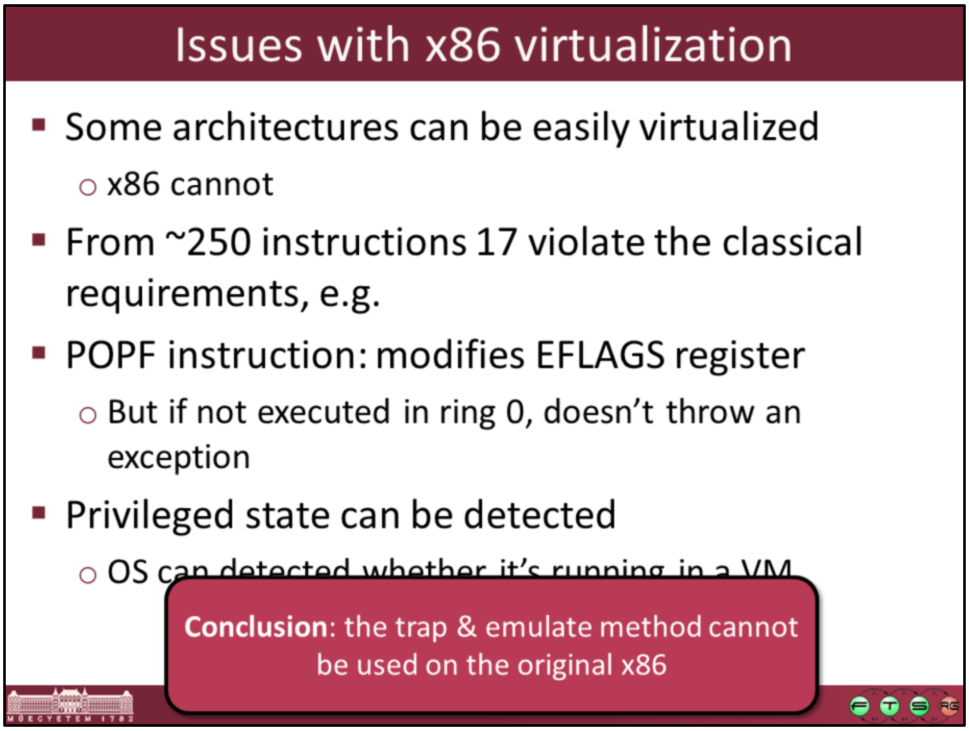
Trap: hardware exception handling, which resumes execution after the handler

Non-privileged instructions are directly executed on the hardware

Privileged or sensitive instructions cause a trap, VMM handles it

HW support is required:
- protection modes (e.g. x86 ring)
  - VM runs in a lower mode
- Privileged instructions should case a trap when called from a non-privileged mode

-Non-sensitive, unprivileged application instructions can be executed directly on the processor with no VMM intervention.
-Sensitive, privileged instructions will be detected when they trap after being executed in user mode. The trap should be delivered to the VMM that will emulate the expected behavior of the instruction in software.
-Sensitive, unprivileged instructions must be detected so that control can be transferred to the VMM.

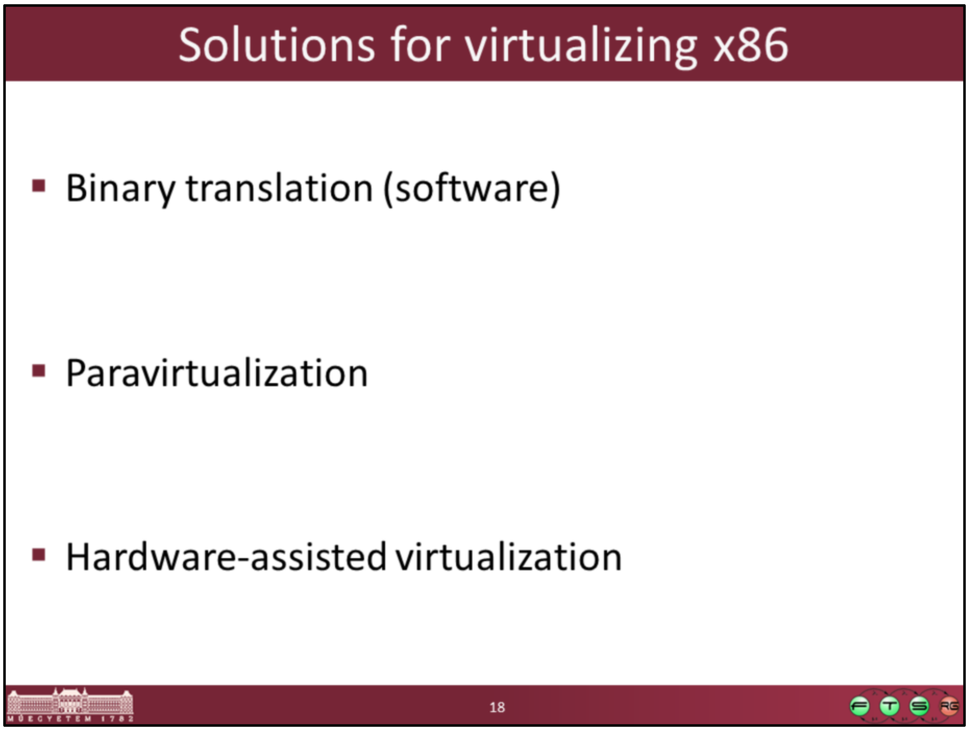


## Issues with x86 virtualization

- Some architectures can be easily virtualized
  - x86 cannot
- From ~250 instructions 17 violate the classical requirements, e.g.
- POPF instruction: modifies EFLAGS register
  - But if not executed in ring 0, doesn't throw an exception
- Privileged state can be detected
  - OS can detected whether it's running in a VM

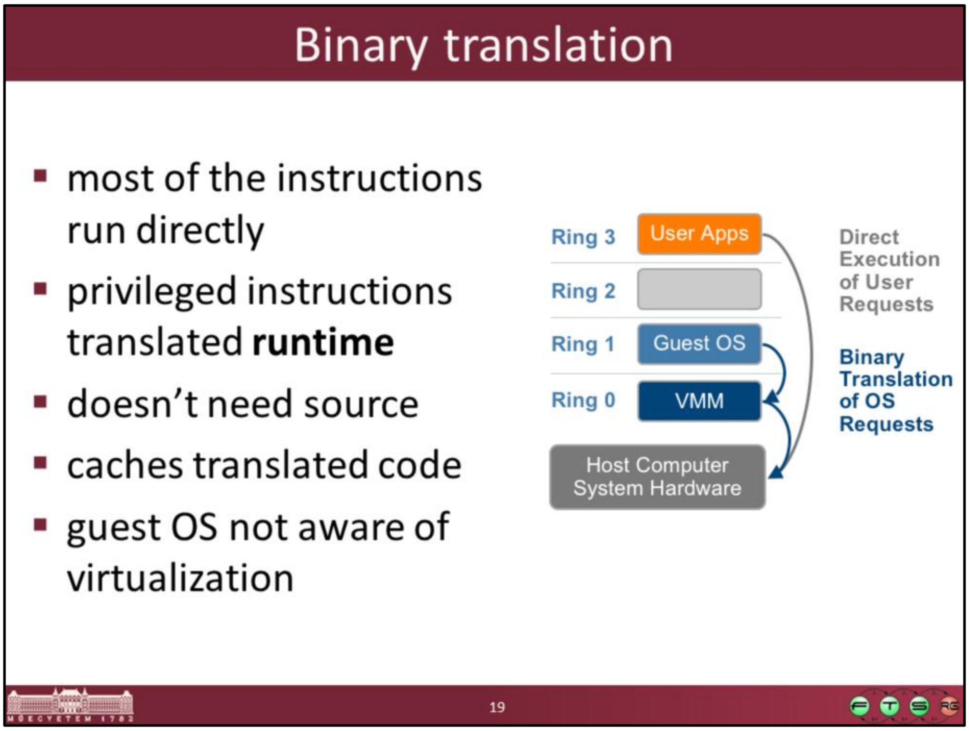
**Conclusion**: the trap & emulate method cannot be used on the original x86

See: J. S. Robin and C. E. Irvine. Analysis of the Intel Pentium's ability to support a secure virtual machine monitor. In *Proceedings of the 9th USENIX Security Symposium, Denver, CO, USA, pages 129.144,* Aug. 2000.



# Solutions for virtualizing x86

- Binary translation (software)
- Paravirtualization
- Hardware-assisted virtualization

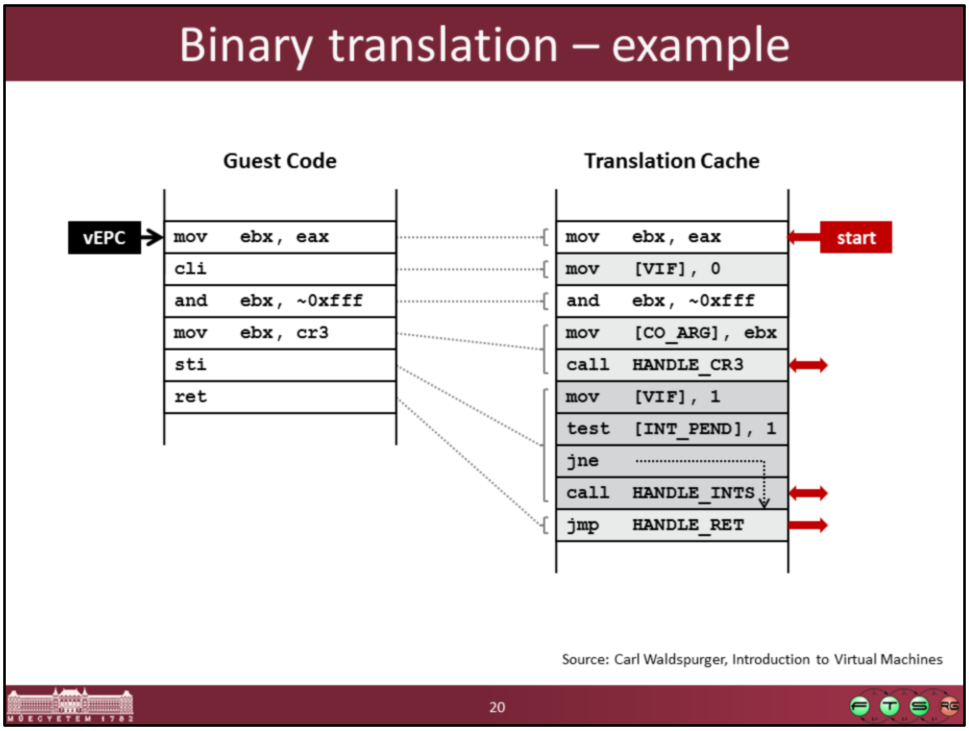


# Binary translation

- most of the instructions run directly
- privileged instructions translated **runtime**
- doesn't need source
- caches translated code
- guest OS not aware of virtualization

Ring 3
User Apps
Ring 2
Ring 1
Guest OS
Ring 0
VMM
Host Computer System Hardware
Direct Execution of User Requests
Binary Translation of OS Requests
19

Source: VMware, Understanding Full Virtualization, Paravirtualization, and Hardware Assisted Virtualization
http://www.vmware.com/files/pdf/VMware_paravirtualization.pdf



# Binary translation – example

Guest Code

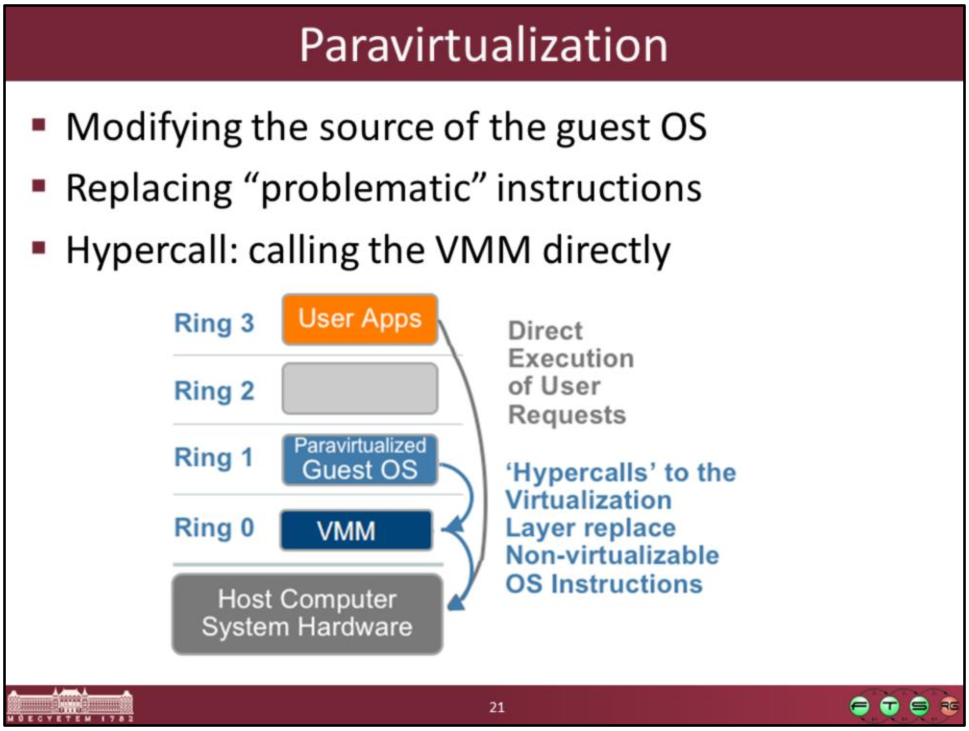
```
mov   ebx, eax
cli
and   ebx, ~0xfff
mov   ebx, cr3
sti
ret
```

Translation Cache

```
mov   ebx, eax
mov   [VIF], 0
and   ebx, ~0xfff
mov   [CO_ARG], ebx
call  HANDLE_CR3
mov   [VIF], 1
test  [INT_PEND], 1
jne
call  HANDLE_INTS
jmp   HANDLE_RET
```

Source: Carl Waldspurger, Introduction to Virtual Machines

Source: http://labs.vmware.com/academic/mit-iap-2010



## Paravirtualization

- Modifying the source of the guest OS
- Replacing “problematic” instructions
- Hypercall: calling the VMM directly

Ring 3 User Apps

Ring 2

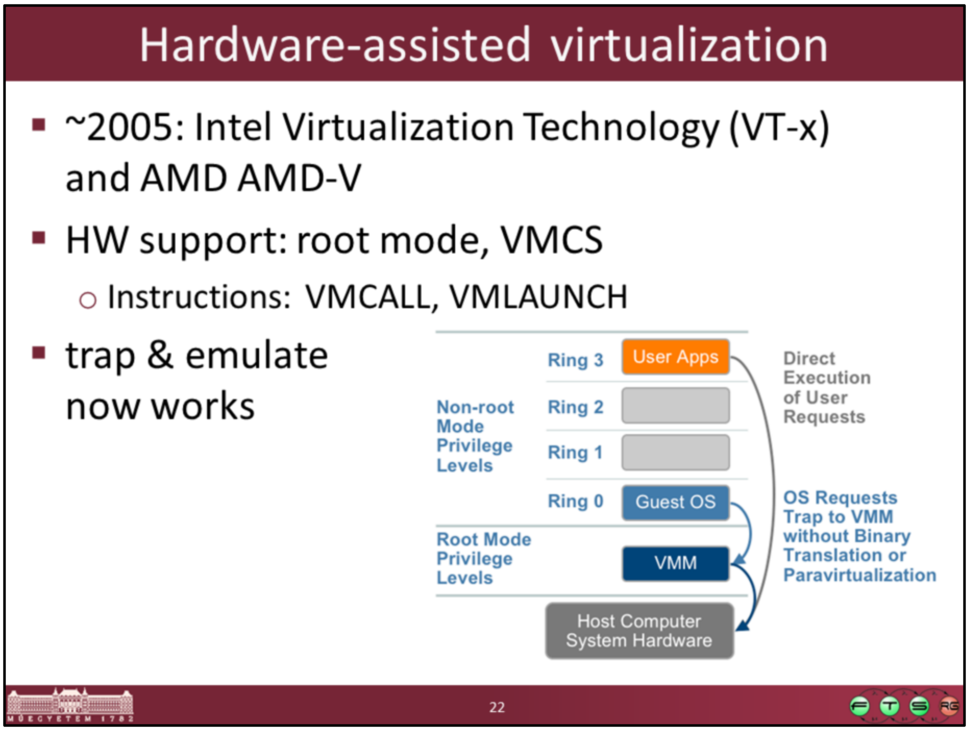
Ring 1 Paravirtualized Guest OS

Ring 0 VMM

Host Computer System Hardware

Direct Execution of User Requests

‘Hypercalls’ to the Virtualization Layer replace Non-virtualizable OS Instructions



## Hardware-assisted virtualization

- ~2005: Intel Virtualization Technology (VT-x) and AMD AMD-V
- HW support: root mode, VMCS
  - Instructions: VMCALL, VMLAUNCH
- trap & emulate now works

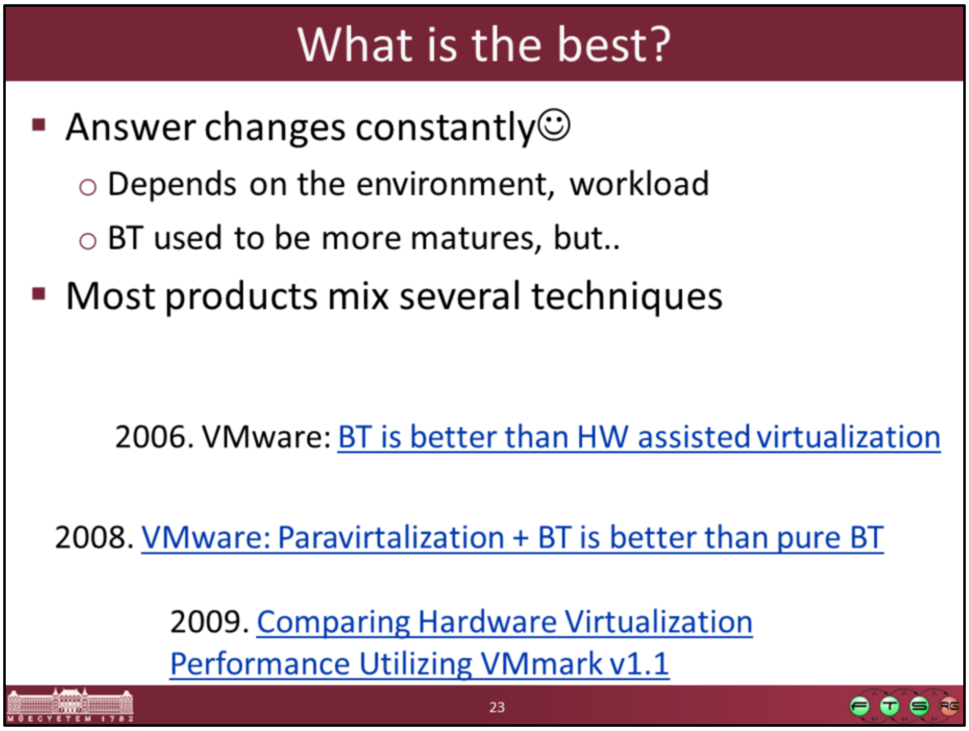
Ring 3
User Apps
Direct Execution of User Requests
Non-root Mode Privilege Levels
Ring 2
Ring 1
Ring 0
Guest OS
OS Requests Trap to VMM without Binary Translation or Paravirtualization
Root Mode Privilege Levels
VMM
Host Computer System Hardware
MŰEGYETEM 1782

Intel VT-x:

- VMCS (Virtual Machine Control Structure)
- VMLAUNCH Launches a virtual machine managed by the VMCS. A VM entry occurs, transferring control to the VM.
- VMCALL Allows a guest in VMX non-root operation to call the VMM for service. A VM exit occurs, transferring control to the VMM.

More info:

- Intel® Virtualization Technology: Hardware Support for Efficient Processor Virtualization, Intel Technology Journal, Volume 10, Issue 03, http://www.intel.com/technology/itj/2006/v10i3/1-hardware/1-abstract.htm



## What is the best?

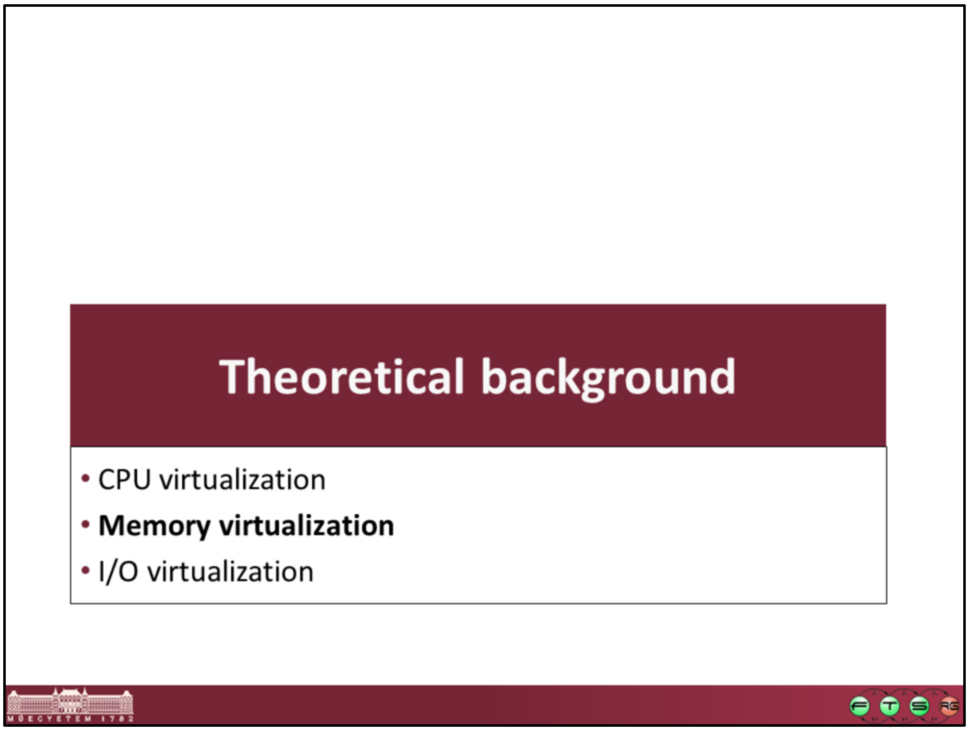
- Answer changes constantly☺
  - Depends on the environment, workload
  - BT used to be more matures, but..
- Most products mix several techniques

2006. VMware: BT is better than HW assisted virtualization

2008. VMware: Paravirtalization + BT is better than pure BT

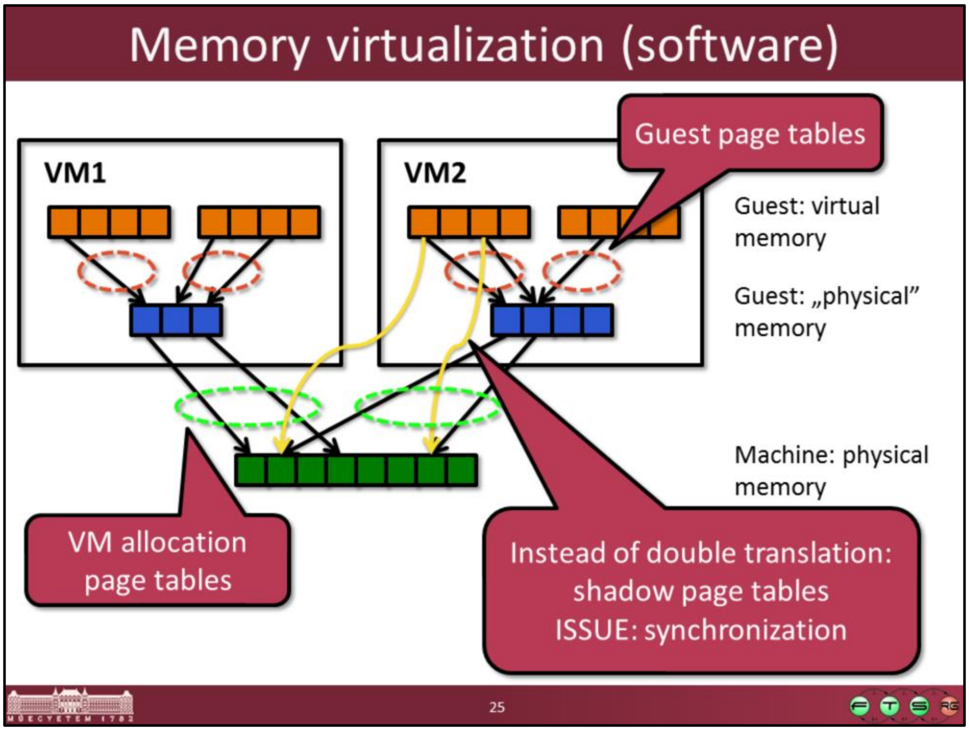
2009. Comparing Hardware Virtualization Performance Utilizing VMmark v1.1

- Binary translation: VMware Player/Workstation, VMware ESX (some 32 bit guest), Virtual PC, MS Virtual Server
- Paravirtualization: Xen (Linux VM), partially MS Hyper-V (for some Windows and Linux)
- HW virtualization: Xen (Windows VM), MS Hyper-V (HW support is a requirement), VMware (64 bit guest)



# Theoretical background

- CPU virtualization
- **Memory virtualization**
- I/O virtualization



# Memory virtualization (software)

VM1

VM2

Guest page tables

Guest: virtual memory

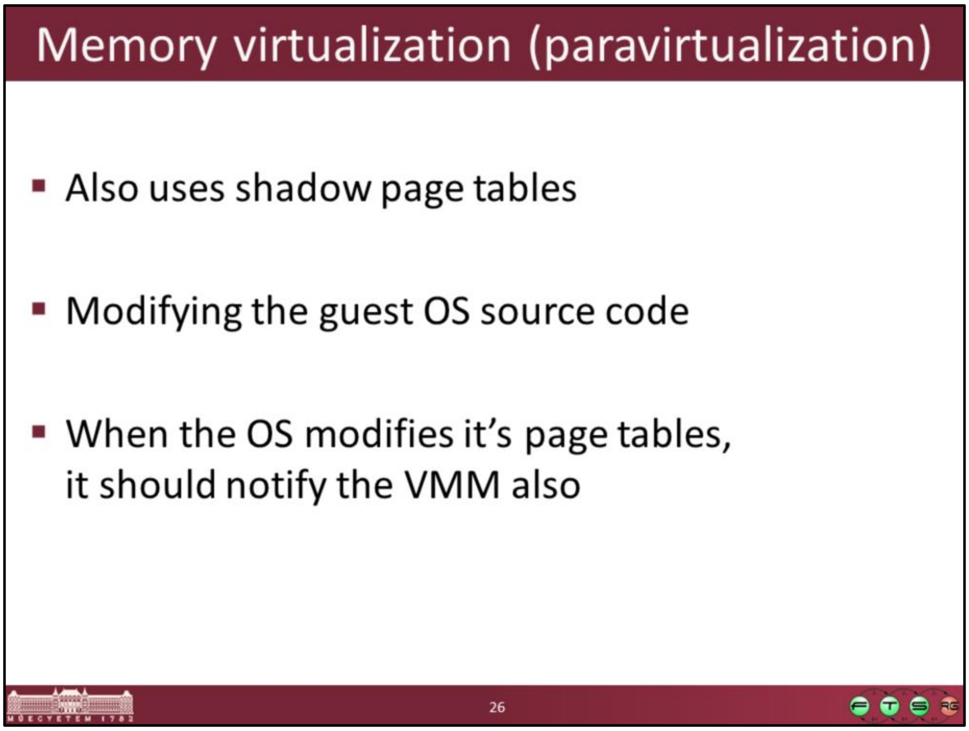
Guest: „physical" memory

Machine: physical memory

VM allocation page tables

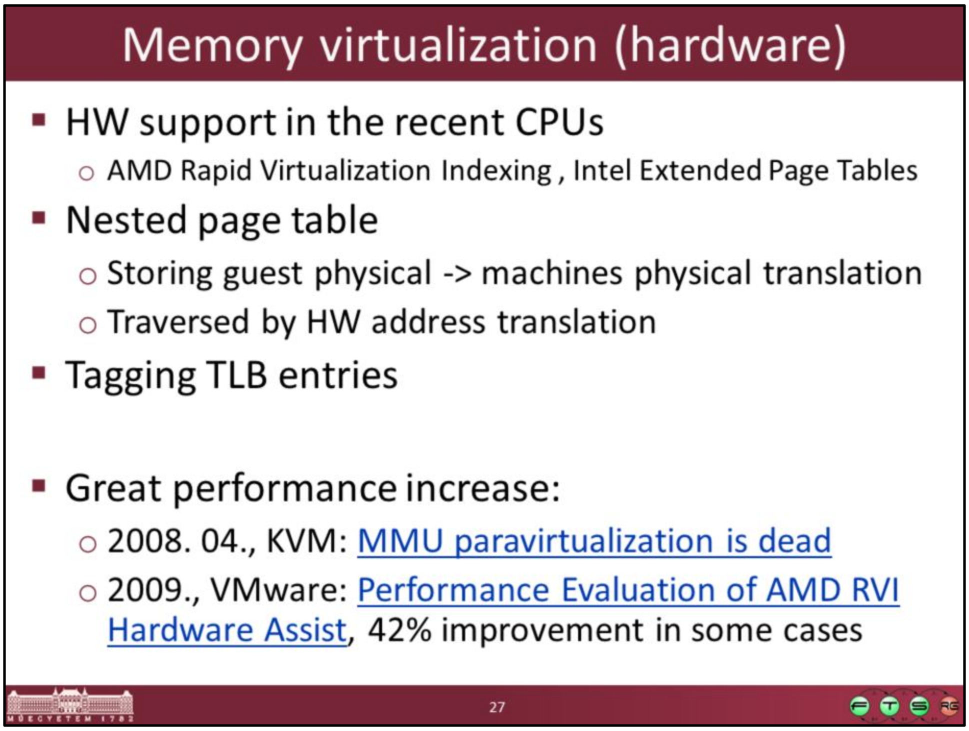
Instead of double translation:
shadow page tables
ISSUE: synchronization

More info on VMware's solution: C.A. Waldspurger, "Memory resource management in VMware ESX server," *SIGOPS Oper. Syst. Rev.*, vol. 36, 2002, pp. 181-194. , http://www.waldspurger.org/carl/papers/esx-mem-osdi02.pdf



# Memory virtualization (paravirtualization)

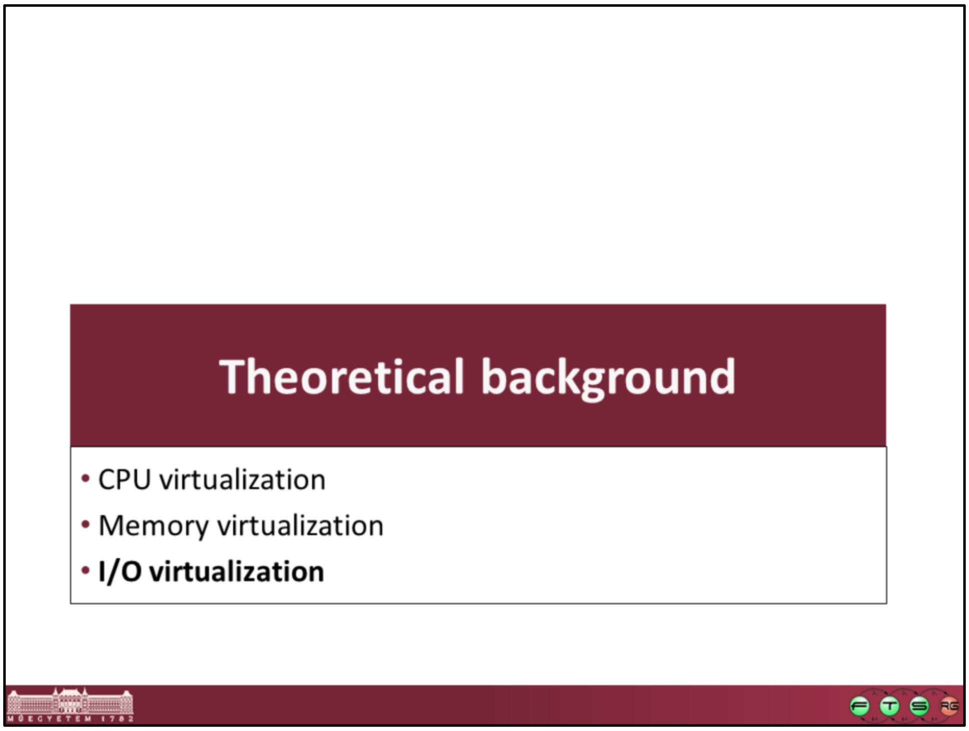
- Also uses shadow page tables
- Modifying the guest OS source code
- When the OS modifies it’s page tables, it should notify the VMM also



## Memory virtualization (hardware)

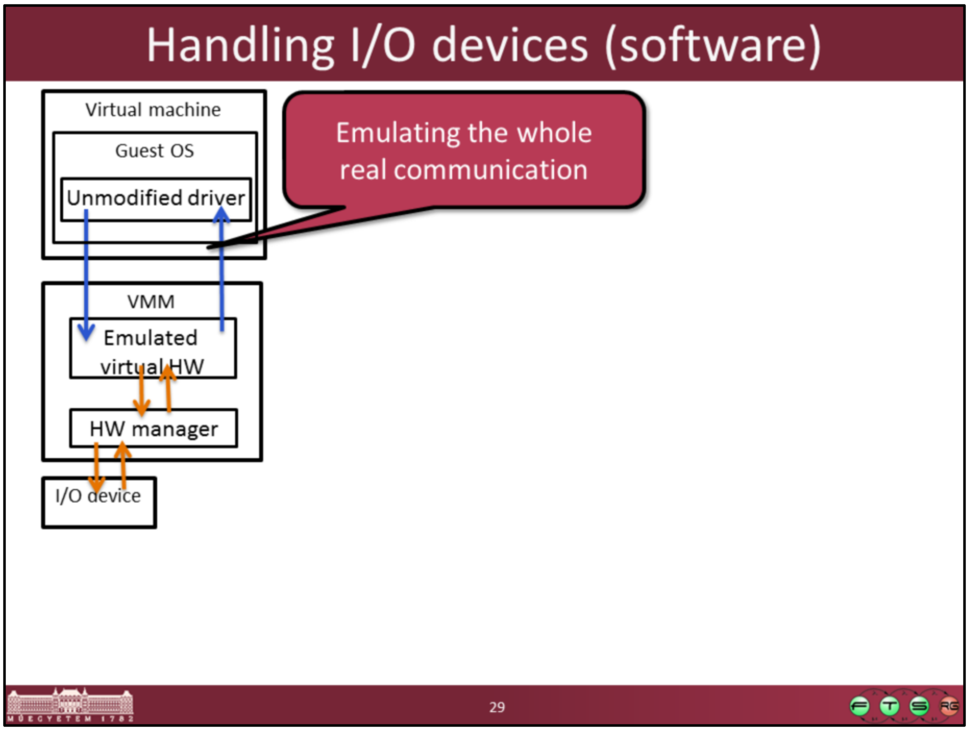
- HW support in the recent CPUs
  - AMD Rapid Virtualization Indexing , Intel Extended Page Tables
- Nested page table
  - Storing guest physical -> machines physical translation
  - Traversed by HW address translation
- Tagging TLB entries
- Great performance increase:
  - 2008. 04., KVM: MMU paravirtualization is dead
  - 2009., VMware: Performance Evaluation of AMD RVI Hardware Assist, 42% improvement in some cases

More info on AMD RVI: http://developer.amd.com/assets/NPT-WP-1%201-final-TM.pdf



## Theoretical background

- CPU virtualization
- Memory virtualization
- **I/O virtualization**



# Handling I/O devices (software)

Virtual machine

Guest OS

Unmodified driver

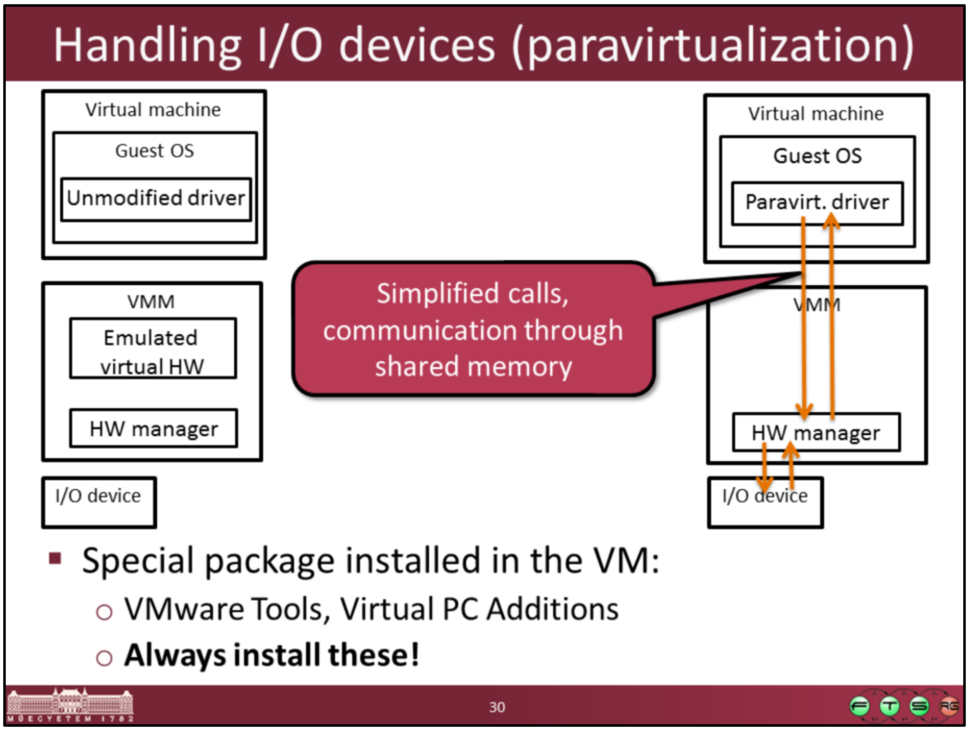
VMM

Emulated virtual HW

HW manager

I/O device

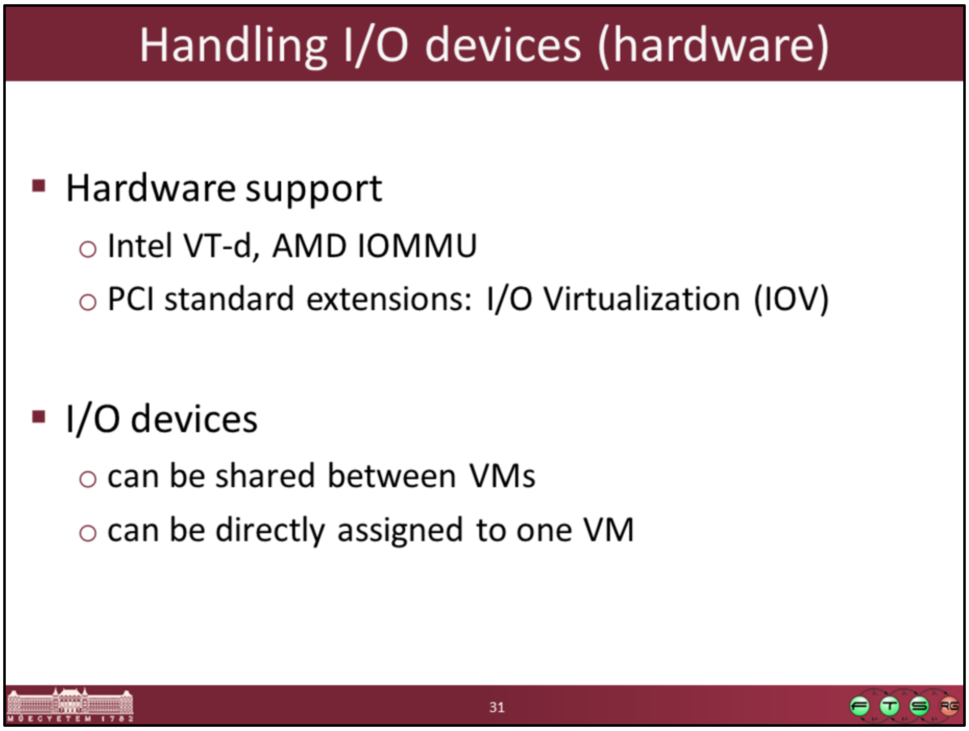
Emulating the whole real communication



# Handling I/O devices (paravirtualization)

Simplified calls, communication through shared memory

- Special package installed in the VM:
  - VMware Tools, Virtual PC Additions
  - **Always install these!**

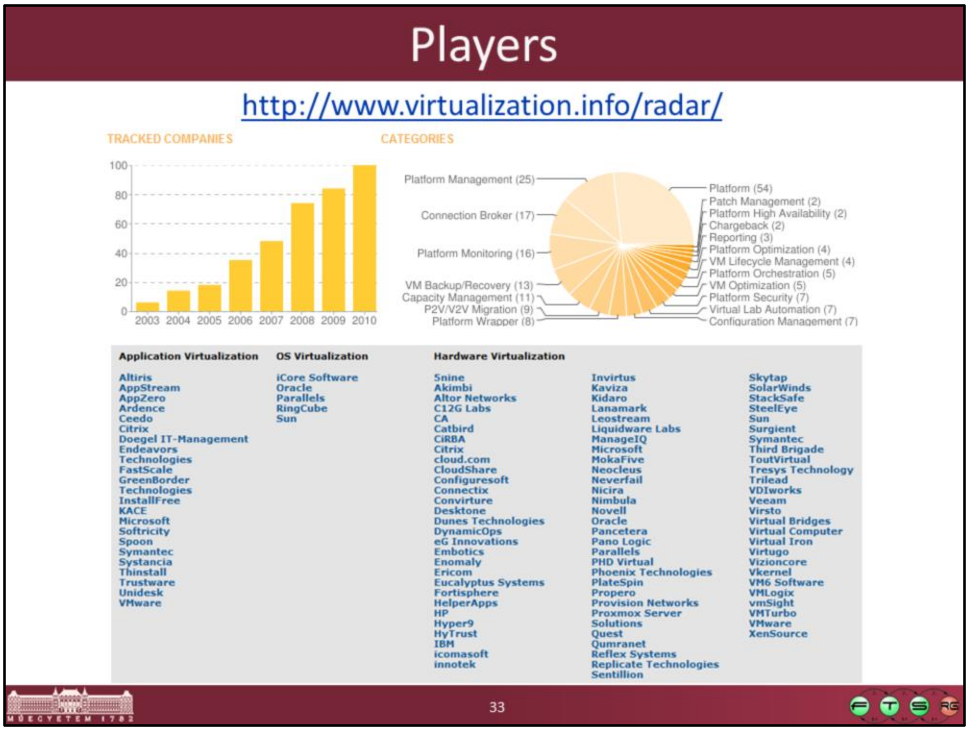


## Handling I/O devices (hardware)

- Hardware support
  - Intel VT-d, AMD IOMMU
  - PCI standard extensions: I/O Virtualization (IOV)
- I/O devices
  - can be shared between VMs
  - can be directly assigned to one VM

More info: Carl Waldspurger and Mendel Rosenblum. 2012. I/O virtualization. *Commun. ACM* 55, 1 (January 2012), 66-73. DOI=10.1145/2063176.2063194 (http://doi.acm.org/10.1145/2063176.2063194)

# Products and companies



# Players

http://www.virtualization.info/radar/

**Application Virtualization**

Altiris
AppStream
AppZero
Ardence
Ceedo
Citrix
Doegel IT-Management
Endeavors
Technologies
FastScale
GreenBorder
Technologies
InstallFree
KACE
Microsoft
Softricity
Spoon
Symantec
Systancia
Thinstall
Trustware
Unidesk
VMware

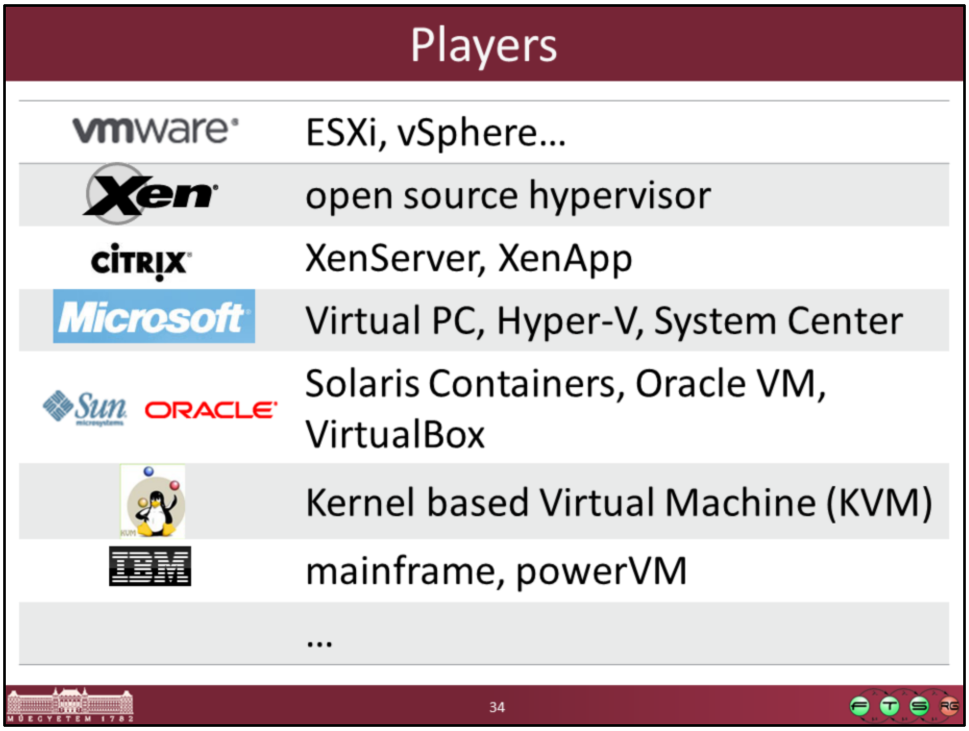
**OS Virtualization**

iCore Software
Oracle
Parallels
RingCube
Sun

**Hardware Virtualization**

5nine
Akimbi
Altor Networks
C12G Labs
CA
Catbird
CiRBA
Citrix
cloud.com
CloudShare
Configuresoft
Connectix
Convirture
Desktone
Dunes Technologies
DynamicOps
eG Innovations
Embotics
Enomaly
Ericom
Eucalyptus Systems
Fortisphere
HelperApps
HP
Hyper9
HyTrust
IBM
icomasoft
innotek
Invirtus
Kaviza
Kidaro
Lanamark
Leostream
Liquidware Labs
ManageIQ
Microsoft
MokaFive
Neocleus
Neverfail
Nicira
Nimbula
Novell
Oracle
Pancetera
Pano Logic
Parallels
PHD Virtual
Phoenix Technologies
PlateSpin
Propero
Provision Networks
Proxmox Server
Solutions
Quest
Qumranet
Reflex Systems
Replicate Technologies
Sentillion
Skytap
SolarWinds
StackSafe
SteelEye
Sun
Surgient
Symantec
Third Brigade
ToutVirtual
Tresys Technology
Trilead
VDIworks
Veeam
Virsto
Virtual Bridges
Virtual Computer
Virtual Iron
Virtugo
Vizioncore
Vkernel
VM6 Software
VMLogix
vmSight
VMTurbo
VMware
XenSource

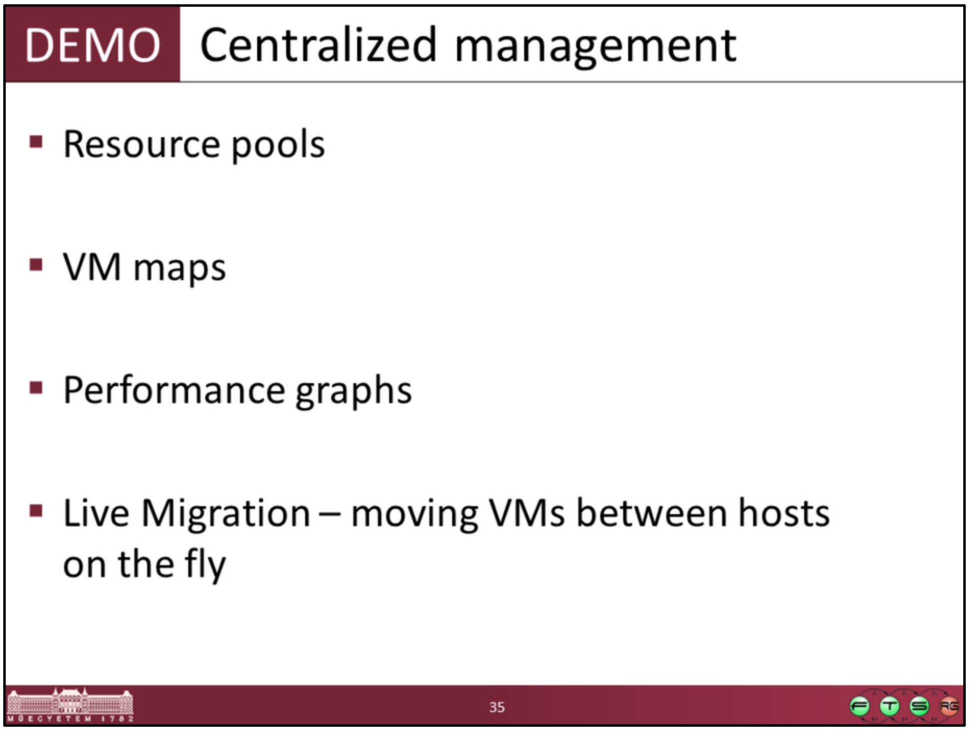
Or: http://en.wikipedia.org/wiki/Comparison_of_virtual_machines



## Players

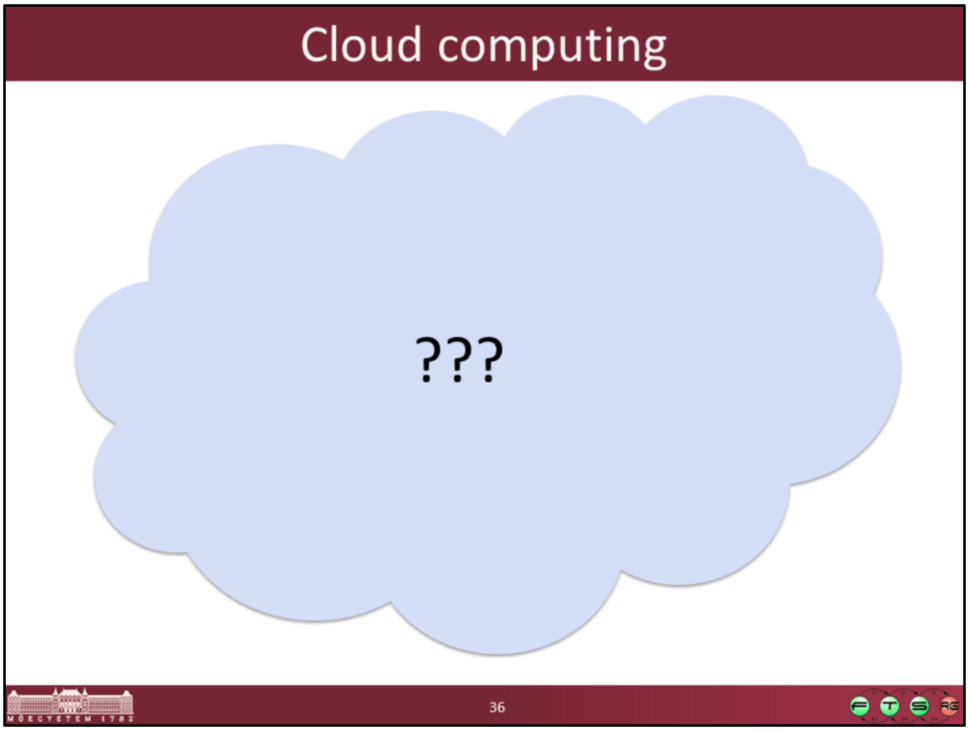
| | |
|---|---|
| vmware | ESXi, vSphere… |
| Xen | open source hypervisor |
| CITRIX | XenServer, XenApp |
| Microsoft | Virtual PC, Hyper-V, System Center |
| Sun ORACLE | Solaris Containers, Oracle VM, VirtualBox |
| KVM | Kernel based Virtual Machine (KVM) |
| IBM | mainframe, powerVM |
| | … |

Only a partial list!



# DEMO Centralized management

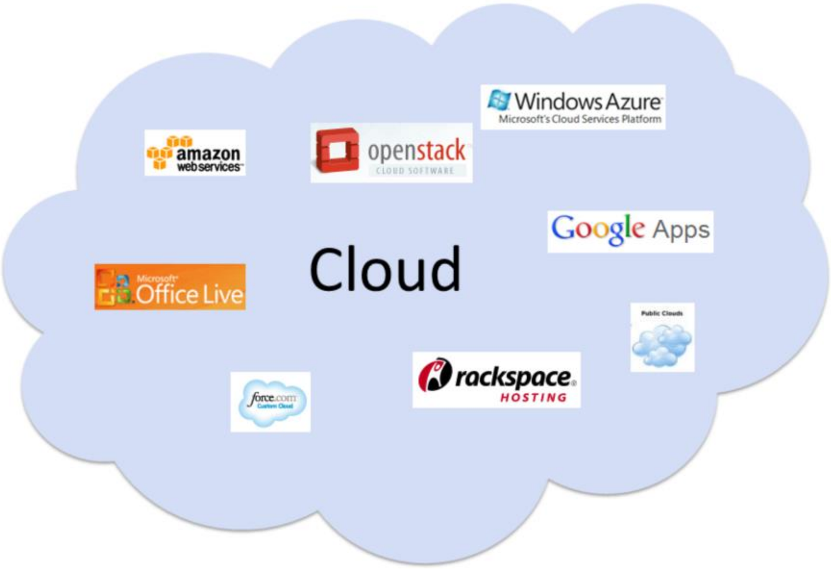
- Resource pools
- VM maps
- Performance graphs
- Live Migration – moving VMs between hosts on the fly

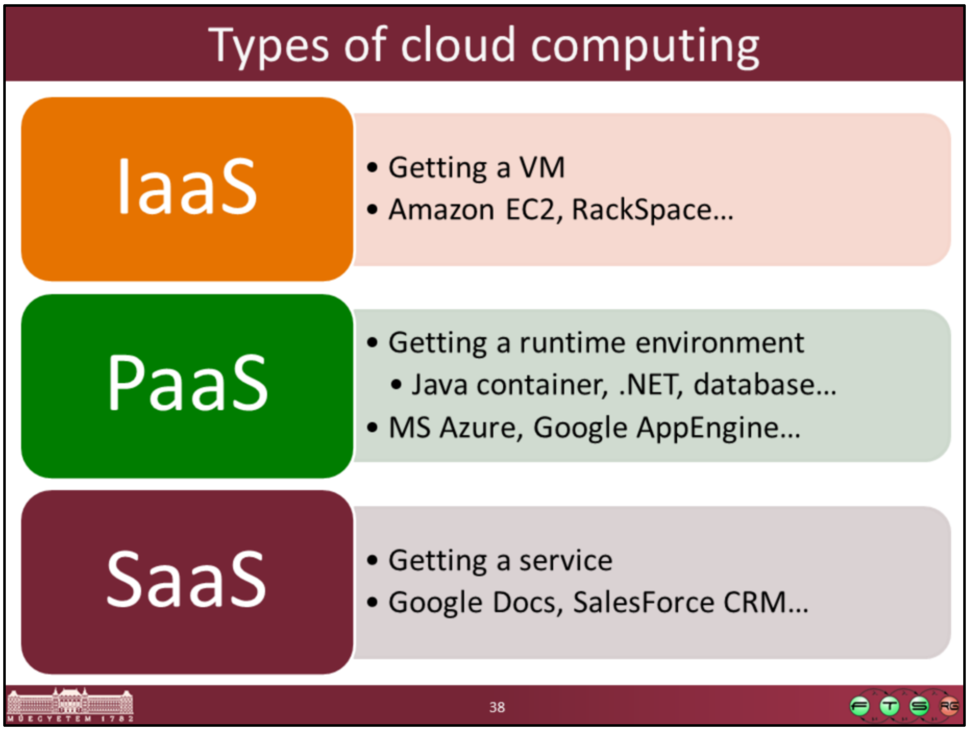


# Cloud computing

???

# Cloud computing





# Types of cloud computing

IaaS

- Getting a VM
- Amazon EC2, RackSpace…

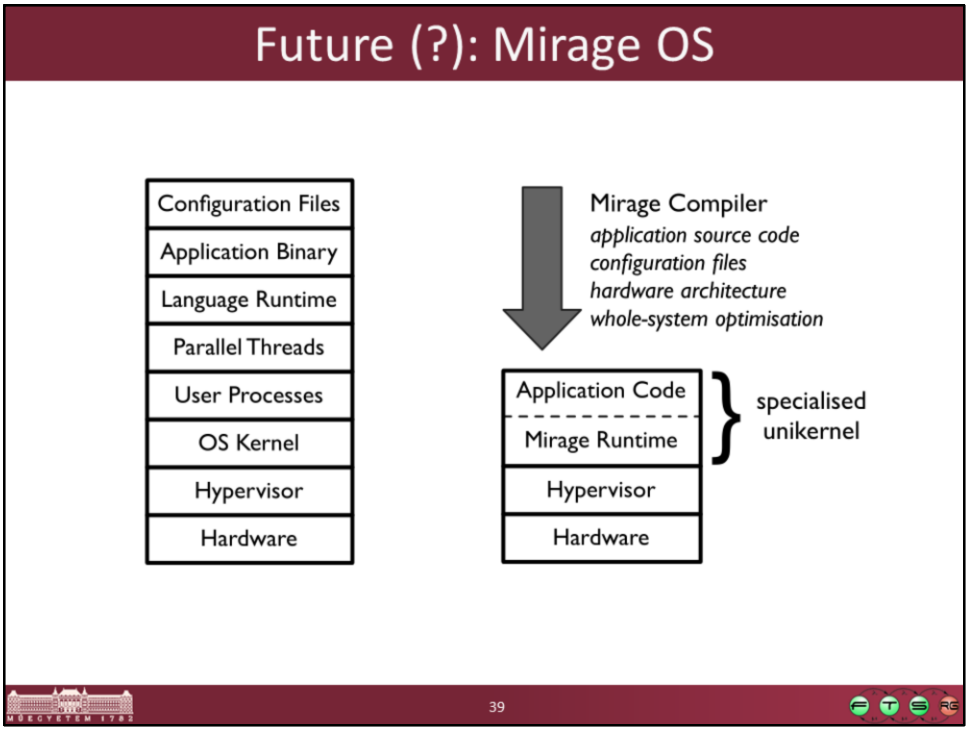
PaaS

- Getting a runtime environment
  - Java container, .NET, database…
- MS Azure, Google AppEngine…

SaaS

- Getting a service
- Google Docs, SalesForce CRM…

IaaS = Infrastructure as a Service
PaaS = Platform as a Service
SaaS = Software as a Service



# Future (?): Mirage OS

Configuration Files

Application Binary

Language Runtime

Parallel Threads

User Processes

OS Kernel

Hypervisor

Hardware

Mirage Compiler

*application source code*

*configuration files*

*hardware architecture*

*whole-system optimisation*

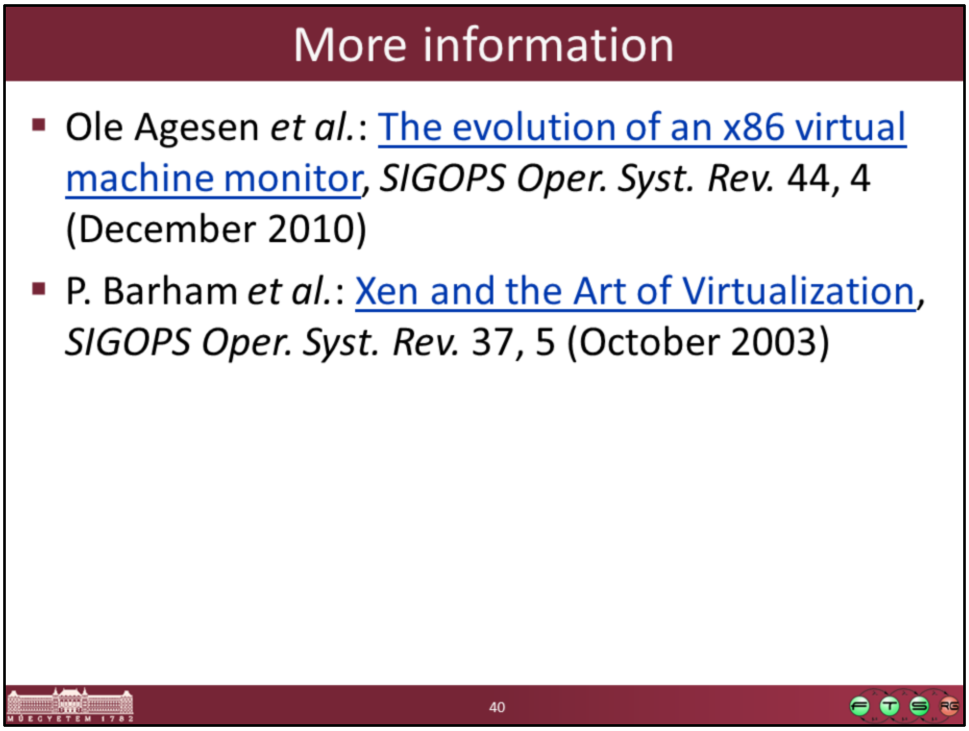
Application Code

Mirage Runtime

Hypervisor

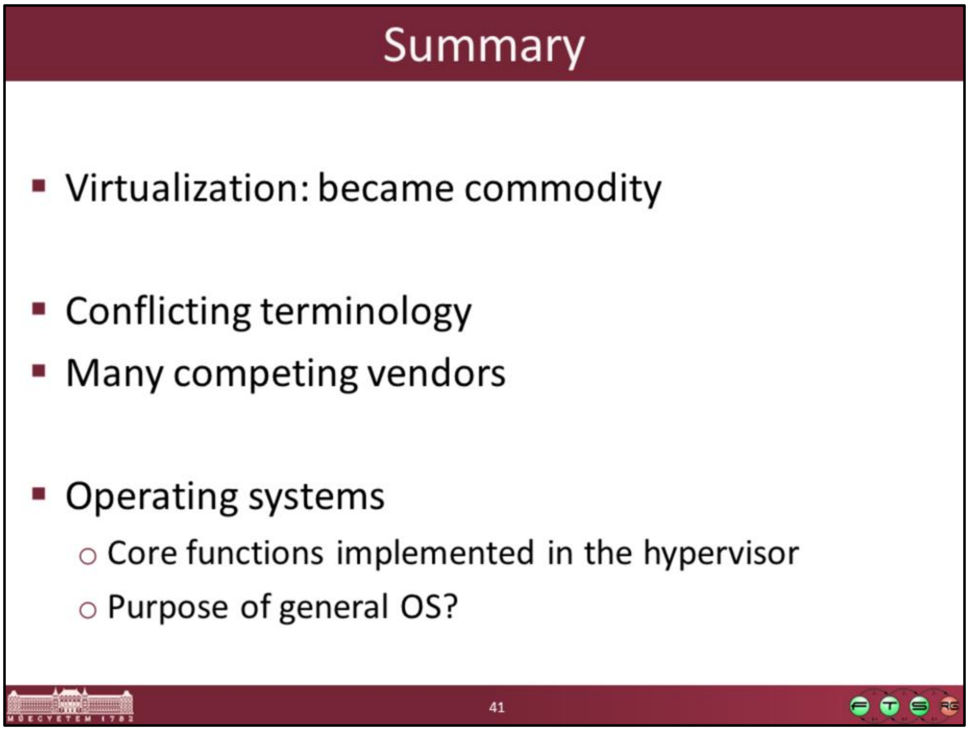
Hardware

specialised unikernel



# More information

- Ole Agesen *et al.*: The evolution of an x86 virtual machine monitor, *SIGOPS Oper. Syst. Rev.* 44, 4 (December 2010)
- P. Barham *et al.*: Xen and the Art of Virtualization, *SIGOPS Oper. Syst. Rev.* 37, 5 (October 2003)



# Summary

- Virtualization: became commodity

- Conflicting terminology
- Many competing vendors

- Operating systems
  - Core functions implemented in the hypervisor
  - Purpose of general OS?