Virtualization

abstraction of computer resources may, but does not have to change the interface. End-user has limited or no knowledge about the real resources behind the virtualization layer. Original / physical characteristics are hidden.
Virtualization

join, aggregation, concatenation, array, N→1
ühendamine, agregeerimine, konkateneerimine, massiiv
Virtualization

partitioning, fragmenting, $1 \rightarrow N$

partitsioneerimine, fragmenteerimine
Virtualization

emulation, simulation, encapsulating, 1→1
emuleerimine, simuleerimine, kapseldamine
Why Virtualize?

If...

➔ the characteristics of our existing resources are not quite what we need
  ➔ replacing all the hardware is expensive
  ➔ our demands change frequently
➔ we are unsure of our needs
  ➔ sorting out our needs could prove costly when trying out different physical configurations

...then virtualization could be of help
Why Virtualize?

➔ consolidation cuts the costs
➔ virtualization adds extra flexibility
  ➔ answer to frequently changing needs
  ➔ resource management is more granular
➔ possible security gains from isolating the application into virtual environment
➔ affordable testing platforms
➔ simplified end-user interfaces
Virtualization

the term “virtualization” is ~50 years old!

- platform virtualization
- resource virtualization
Platform Virtualization

[Diagram showing a server with CPU, HDD, RAM, and NIC, with smaller replicas of the server showing virtualization at work.]
Platform Virtualization

- server virtualization, virtual machines
- virtualized CPU, memory, I/O, devices
- guest virtual machines run on host hardware
CPU Virtualization

- emulation
  - imitates the behaviour of one type of CPU with another CPU
  - usually implemented in software, performance issues
- QEmu
CPU Virtualization

- full virtualization (*täisvirtualiseerimine*)
  - virtual processor equal to underlying processor
  - can run unmodified guest OS
  - some instructions can be executed directly on physical hardware, some must be trapped/translated

- VMWare Workstation/Server
- VirtualBox
- MS Virtual PC/Server
CPU Virtualization

- full virtualization, hosted on OS

```
x86 apps
x86 OS
x86 full virt VMM
```
```
x86_64 apps
x86_64 OS
x86_64 full virt VMM
```
```
x86_64
```
```
x86_64 apps
```
CPU Virtualization

- full virtualization with hypervisor (*hüperviisör*)
- hypervisor - layer between host hardware and guest OS
- VMM - Virtual Machine Monitor
CPU Virtualization

- paravirtualization (*paravirtualiseerimine*)
  - virtual CPU interface differs (in parts) from physical CPU interface
  - guest OS must be ported
  - can provide better performance in many scenarios
  - Xen, Citrix XenServer, VMWare ESX, MS Hyper-V
CPU Virtualization

→ paravirtualization with hypervisor
CPU Virtualization

→ hardware assist
  → trapping and emulating privileged/sensitive calls
  → AMD: AMD-V (Pacifica)
  → Intel: Intel VT (Vanderpool)
→ VMWare, Xen/XenServer, VirtualBox, Hyper-V
Platform Virtualization

- containers, partitions
  - multiple userspaces on the same OS kernel
    - Solaris Containers
    - Linux Vserver
  - virtual directory trees
    - chroot,
    - BSD Jail
Resource Virtualization

→ virtual memory, virtual machines(?!)
  → natural part of modern operating systems
Resource Virtualization

- storage virtualization
  - RAID
  - Logical Volume Management, LVM
  - Network-Attached Storage
Resource Virtualization

- network virtualization
  - Virtual Lan, VLAN
  - Virtual Private Network, VPN
  - link aggregation
  - Network Address Translation, NAT
Resource Virtualization

- virtualizing the computing power
  - clusters
  - GRID
  - Cloud Computing
  - ....
Virtualization Upsides

- better resource utilization
- cost reduction
  - hardware, power, administration
- recovery from single (virtual) machine disaster is easier
- rolling out new systems is faster
- extensive changes can be made without touching the physical layer
- flexible resources matching the needs
Virtualization Downsides

- additional virtualization layers add more complexity
- recovery from large-scale disaster is complicated
- security: virtualization layer must be carefully protected
- resource overhead: additional layers consume resources