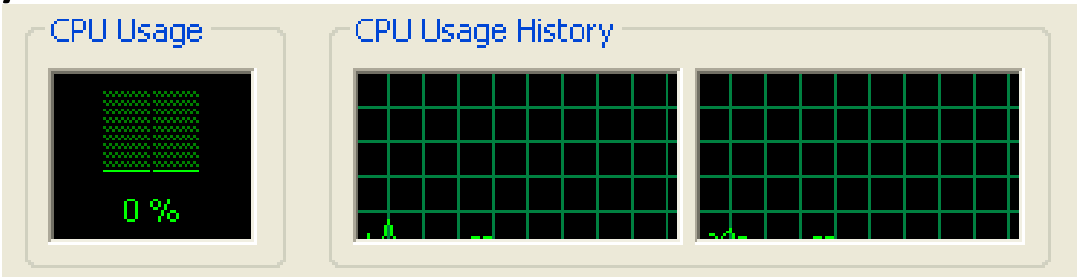
Virtualizáció, Cloud, SDN

Gombos Gergő

Virtualizáció

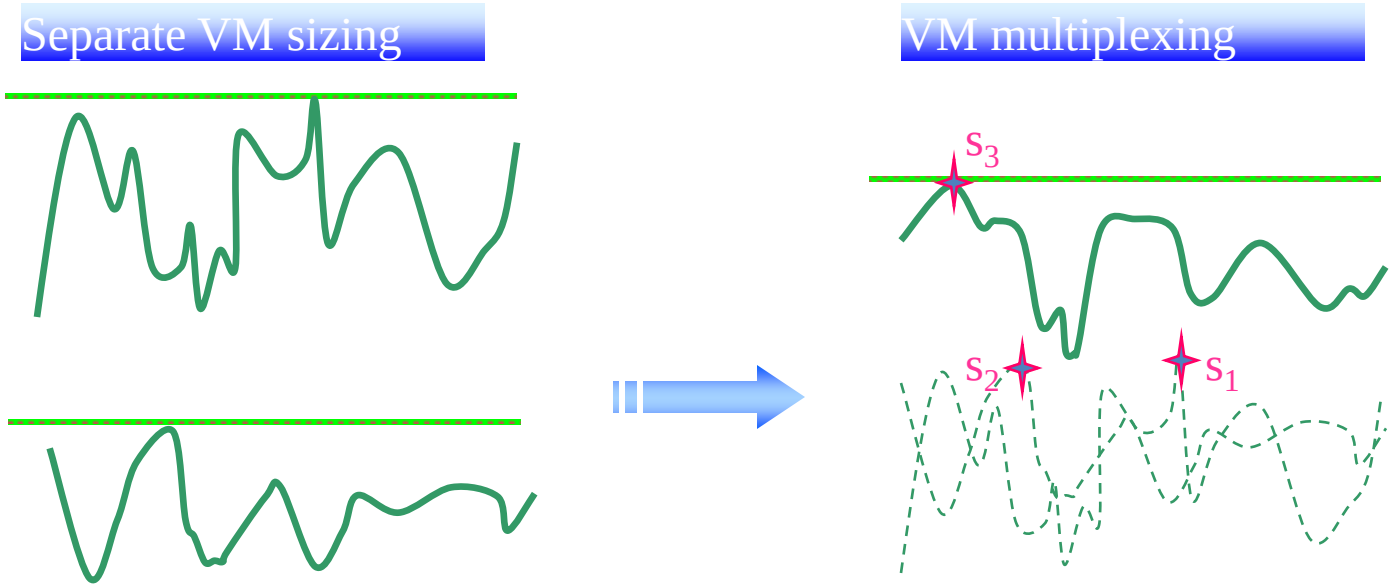
Non-virtualized Servers

- Too many servers for too little work



- High costs and infrastructure needs
  - Maintenance
  - Networking
  - Floor space
  - Cooling
  - Power
  - Disaster Recovery

VM workload multiplexing



- Multiplex VMs' workload on same physical server
  - Aggregate multiple workload. Estimate total capacity need based on aggregated workload
  - Performance level of each VM be preserved

Mi a virtualizáció?

- A virtualizáció egy olyan technikai megoldás, ami lehetővé teszi a rendszer erőforrásainak felosztását több (esetleg eltérő) virtuális számítógépre.
- Egyszerűen fogalmazva költséghatékonyan lehet virtuálisan több környezetet, vagy operációs rendszereket és alkalmazásokat futtatni ugyanazon a számítógépen egyszerre.
- Konkrétabban megfogalmazva: a virtualizáció leválasztja nemcsak a felhasználót, hanem az operációs rendszert is a hardverről.

Kezdetek – Memória virtualizálás

- Az egész rendszer virtualizálásának ötlete előtt megoldották a memória virtualizálásának kezelését.
- Először - Manchester-i egyetem – Atlas
- A virtuális memória lehetőséget ad arra, hogy egy olyan rendszer, aminek amúgy kevés fizikai memóriája van, egyes alkalmazásoknak úgy tűnjön, mintha sokkal több lenne – a merevlemez segítségével

Virtualization

- Virtualization helps us break the “one service per server” model
- Consolidate many services into a fewer number of machines when workload is low, reducing costs
- Conversely, as demand for a particular service increases, we can shift more virtual machines to run that service
- We can build a data center with fewer total resources, since resources are used as needed instead of being dedicated to single services

Virtualizáció felhasználásával

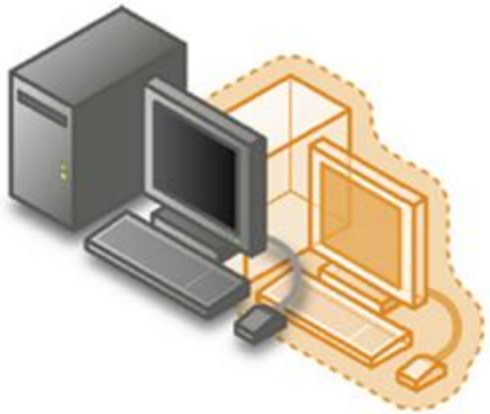
- Egységesített munkafolyamatok
- Csökkentett hardver- , áram- , helyköltségek
- Egy fizikai számítógépen akár több operációs rendszer (biztonsági szempontok – sandbox - izoláció, átláthatóság, hibakeresés)
- Rugalmas infrastruktúra (egyszerűbb létrehozás/törlés/mozgatás/módosítás)

A Virtualizáció hátránya

- Ha a fizikai hardver sérül, a virtuális gépek működésében is problémák merülnek fel, egyszerre leállhat az összes virtuális gép / szolgáltatás, szélsőséges esetben adatvesztés
- Túl sok virtuális gép fut, akkor „harcolhatnak” az erőforrásért

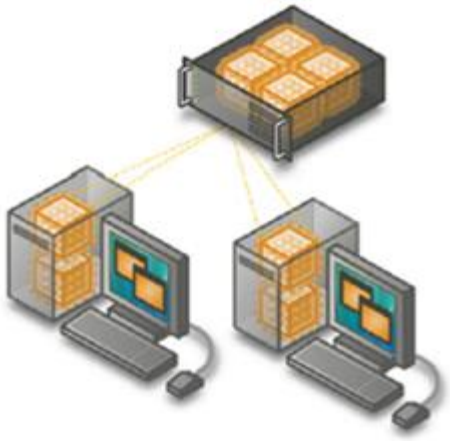
Virtualizáció Típusai I.

- Erőforrás virtualizáció
  - Az operációs rendszerek már jó ideje végeznek valamilyen szintű virtualizációt pl. processzor-ütemezés, virtuális memória, kép file-ok (iso/img)
  - Ezek a módszerek elhitetik a programokkal, hogy másmilyen hardver van a számítógépben, mint ami ténylegesen adott.



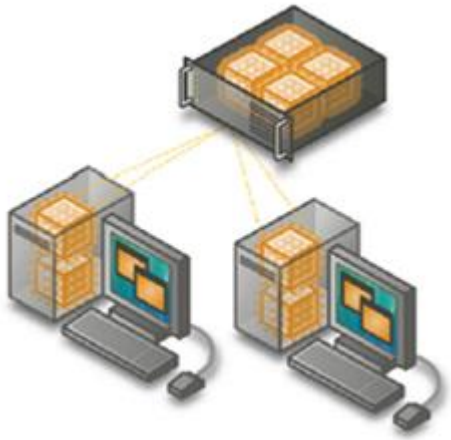
Virtualizáció Típusai II.

- Alkalmazás virtualizáció
  - Gyűjtőfogalom, olyan szoftveres megoldások tartoznak ide,amik segítik a programokat hordozhatóbbá tenni és biztonságos környezetben történő futásukat lehetővé tenni
  - Nem kell minden gépre telepíteni és beállítani az alkalmazást, akár a szerveren is lehet futtatni
  - Kevesebb erőforrást használ, mint egy VM
  - Nem lehet minden szoftvert virtualizálni (pl. amikhez eszközmeghajtó szükséges, vagy az osztott memóriában kell futnia, vagy ha ez egy 16 bites alkalmazás)



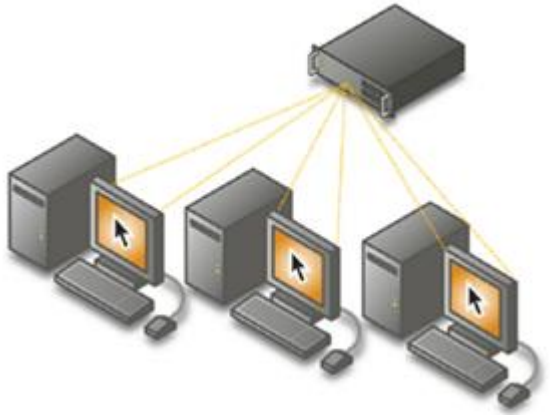
Virtualizáció Típusai III.

- Alkalmazás szintű virtualizáció
  - A platformok közti átjárhatóságon a hangsúly, a cél,hogy különböző operációs rendszereken a program gond nélkül fusson.
  - Pl. Java – saját futtatókörnyezet, wine/cygwin



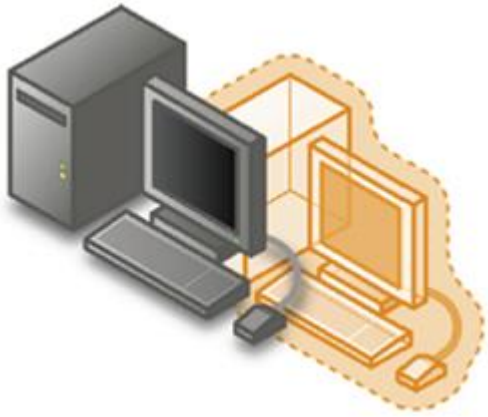
Virtualizáció Típusai IV.

- Desktop virtualizáció
  - Egy kliens segítségével a hálózaton keresztül lehet bejelentkezni a saját számítógépre, így megvan a felület, de a programok nem a helyi gépen futnak, csak a megjelenítés kerül át.
  - Gyakorlatilag ez a távoli asztal (VNC, Microsoft - Remote Desktop Protocol = RDP)



Virtualizáció Típusai V.

- Emuláció
  - Az emuláció során a fizikai hardvertől teljesen különböző virtuális hardverelemeket lehet létrehozni, ez segít abban, hogy egy programot másik környezetben (pl más architektúrájú processzoron) is ki lehessen próbálni. A teljesítmény gyakran a natív futáshoz képest 20%-os veszteséget is eléri.
  - Emulációt szoktak még régi alkalmazások futtatásához is használni, pl. C64, Nintendo, Dos, Amiga platformra írt programok esetén.
  - Az emulációs szoftver egy folyamat az operációs rendszerben, ebből a szempontból olyan, mint bármelyik másik program.



Virtualizáció Típusai VI.

- Platform virtualizáció
  - Elvárás:
    - A virtualizált operációs rendszer ugyanúgy fusson, mintha közvetlenül lenne feltelepítve, és kizárólagos hozzáférése lenne az erőforrásokhoz.
  - Fajtái:
    - Operációs rendszer szintű virtualizáció
    - Szoftveres virtualizáció
    - Type-2 hypervisor vagy hibrid esetleg hosted virtualizáció
    - Bare-metal vagy Type-1 hypervisor

Platform Virtualizáció

- Operációs rendszer szintű virtualizáció
  - Ha egy meglévő rendszert kell több példányban futtatni – közös kernel, de van lehetőség külön eszközök kialakítására (saját hálózati interfész / memóriaterület / tárhely / felhasználói csoport)
  - Teljes értékű operációs rendszer, csak az eredeti rendszert lehet sokszorosítani
  - Korai megoldása a chroot-olt környezet
    - A chroot beszűkíti az elérhető filerendszert, és a bezárt program nem fog kilátni a neki megadott virtuális környezetből.
    - Nem ad teljes biztonságot, már több kitörési lehetőség ismert

Platform Virtualizáció

- Szoftveres virtualizáció
  - A Virtual Machine Manager (felügyeli a virtuális gépeket, biztosítja számukra az erőforrásokat és kezeli a környezetüket) egy program, ami az operációs rendszerben fut (emulációnak is tekinthető).
  - Ilyen például a Java VM. Rendkívül rugalmas , mivel a virtuális gépnek bármilyen utasításkészletet képes biztosítani.

Platform Virtualizáció

- Type-2 hypervisor vagy hibrid esetleg hosted virtualizáció
  - Az operációs rendszer maga a Virtual Machine Manager, ezek kliens alapú megoldások, mint pl. Sun Virtualbox, Microsoft Virtual PC, VMware Worksation/Player/Server.
  - A hardverre közvetlenül telepítve van egy operációs rendszer (host OS), és efelett fut egy virtualizációs szoftver, ami futtatja a virtuális gépeket (guest OS) és a virtuális hardverkörnyezetet biztosítja.
  - Előnye, hogy viszonylag könnyen telepíthető meglevő rendszerekre. A hardver eléréséhez a host driver-eit veszi igénybe .
  - Megoldható,hogy a virtuális gépek közvetlenül elérjék a host OS-t (közös mappa,amit mind a host OS mind a guest OS lát).

Platform Virtualizáció

- Bare-metal vagy Type-1 hypervisor
  - A Virtual Machine Manager az operációs rendszer és a hardver közé kerül.
  - Amikor feltelepül az operációs rendszer, azzal együtt feltelepül a hypervisor (speciális operációs rendszer kernel, ami virtuális hardverkörnyezetet biztosít, minden virtuális gép efelett fut, még az ezt kezelő operációs rendszer is), és ez fogja vezérelni az operációs rendszert is.
  - Pl. Oracle VM, VMware ESX, Citrix XenServer és Microsoft Hyper-V.

A Virtualizációról

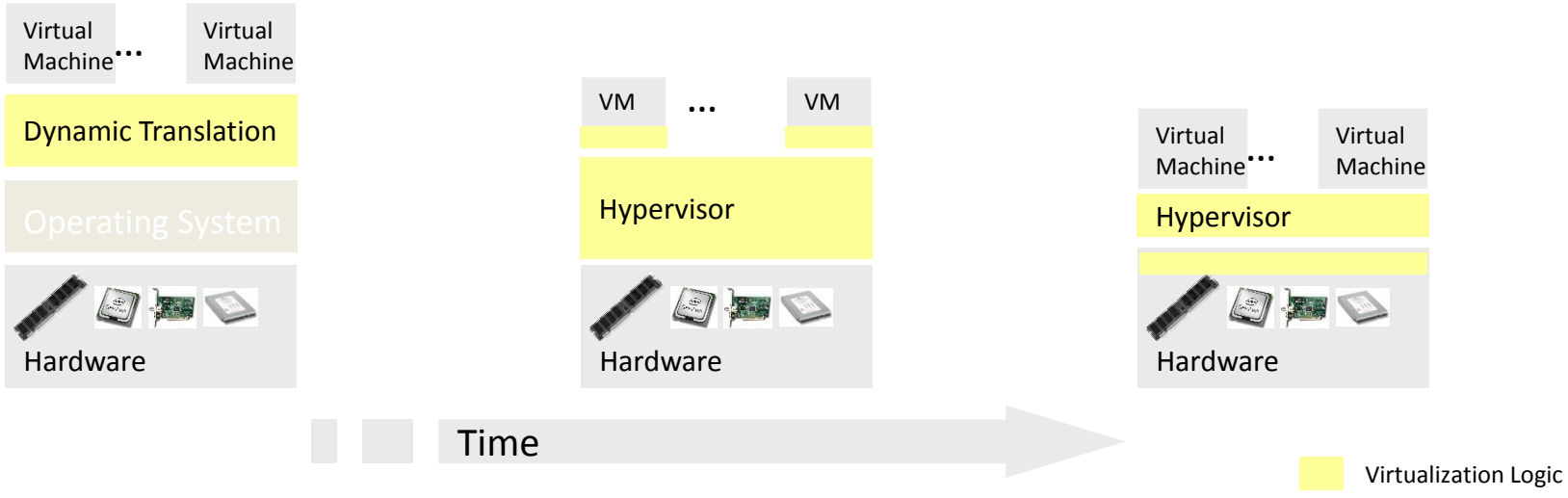
- A különböző virtualizációs programokból vannak szerver és asztali megoldások is
- Az asztali megoldásoknál a könnyű kezelhetőség, átlátható grafikus felület a lényeges, a szerver megoldásoknál pedig a magas rendelkezésre állás.
- Fontos, hogy egy virtuális gép felépítésénél az alap hardvert ismerni kell (nem érdemes több CPU-t / memóriát / tárhelyet kiosztani, mint ami rendelkezésre áll)

VM-ek migrálása

- Cold migration
  - Gépet leállítjuk
  - Átmásoljuk az új hostra
  - Elindítjuk az új hoston
- Live migration
  - A gépet leállítás nélkül másoljuk át az új hostra

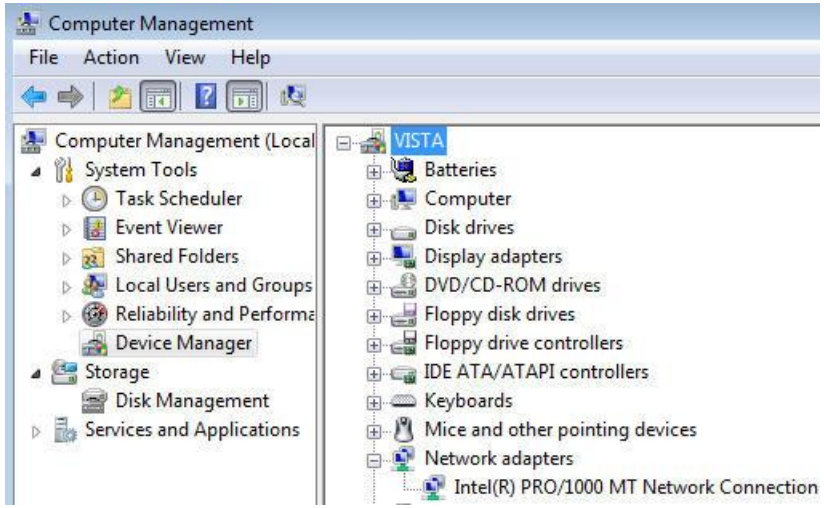
Evolution of Software solutions

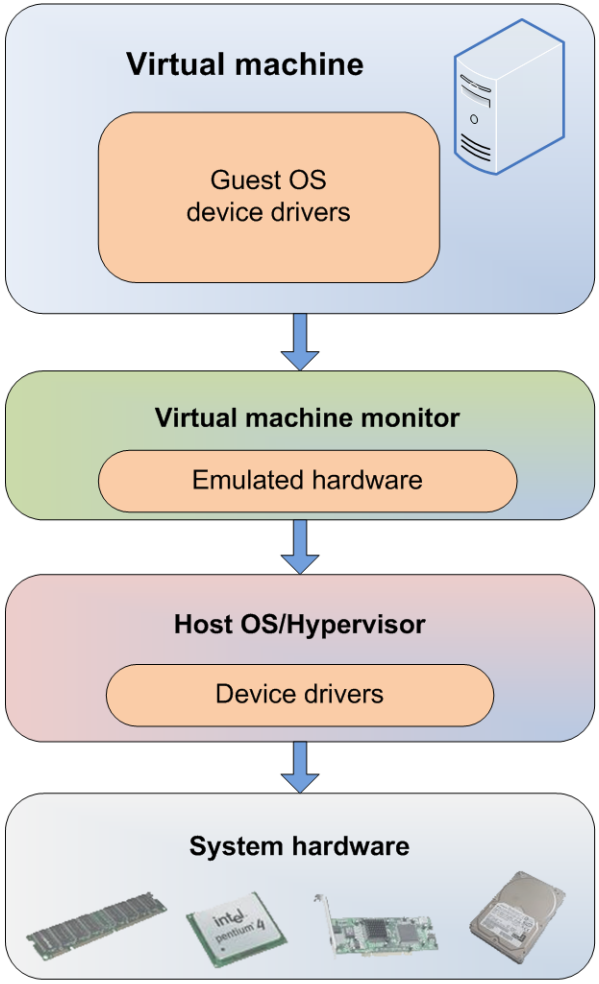
- 1st Generation: Full virtualization (Binary rewriting)
  - Software Based
  - VMware and Microsoft
- 2nd Generation: Paravirtualization
  - Cooperative virtualization
  - Modified guest
  - VMware, Xen
- 3rd Generation: Silicon-based (Hardware-assisted) virtualization
  - Unmodified guest
  - VMware and Xen on virtualization-aware hardware platforms



Full Virtualization

- Everything is virtualized
- Full hardware emulation
- Emulation = latency





Paravirtualization

–OS or system devices are virtualization aware

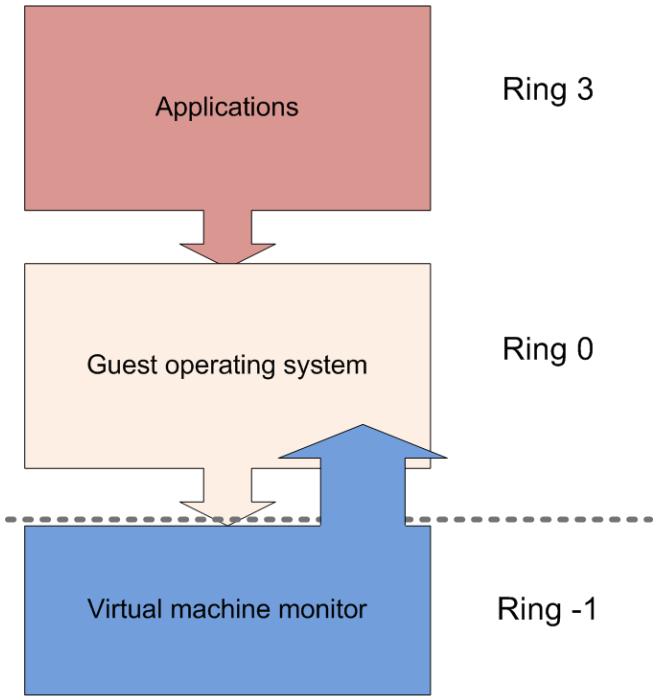
Requirements:

–OS level – recompiled kernel

–Device level – paravirtualized or “enlightened” device drivers

Hardware-assisted Virtualization

- Server hardware is virtualization aware
- Hypervisor and VMM load at privilege Ring -1 (firmware)
- Removes CPU emulation bottleneck
- Memory virtualization coming in quad core AMD and Intel CPUs



Hardware-assisted virtualization

Cloud

Felhők

- NIST (National Institute of Standards)
  - „A számítási felhő (Cloud Computing) modell révén a felhasználók kényelmesen és igény szerint férhetnek hozzá a megosztott, beállítható informatikai erőforrásokhoz (például nyomtatókhoz, kiszolgálókhoz, tárolókapacitáshoz, alkalmazásokhoz és szolgáltatásokhoz), amelyeket gyorsan és minimális adminisztrációs megterhelés vagy szolgáltatói beavatkozás mellett rendelkezésre lehet bocsátani és fel lehet szabadítani.”

What It Provides

- Cloud computing provides shared services as opposed to local servers or storage resources
- Enables access to information from most web-enabled hardware
- Allows for cost savings – reduced facility, hardware/software investments, support

Basic Cloud Characteristics

- The “no-need-to-know” in terms of the underlying details of infrastructure, applications interface with the infrastructure via the APIs.

- The “pay as much as used and needed” type of utility computing and the “always on!, anywhere and any place” type of network-based computing.

Essential Characteristics

- On-demand self-service
  - A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider.
- Broad network access
  - Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

  - Source: NIST Special Publication 800-145

Characteristics

- Resource pooling
  - The provider's computing resources are pooled to serve multiple consumers
  - Resources can be dynamically assigned and reassigned according to customer demand
  - Customer generally may not care where the resources are physically located but should be aware of risks if they are located offshore

  - Source: NIST Special Publication 800-145

Characteristics

- Rapid elasticity
  - Capabilities can be expanded or released automatically (i.e., more cpu power, or ability to handle additional users)
  - To the customer this appears seamless, limitless, and responsive to their changing requirements

- Measured service
  - Customers are charged for the services they use and the amounts
  - There is a metering concept where customer resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service
  - Source: NIST Special Publication 800-145

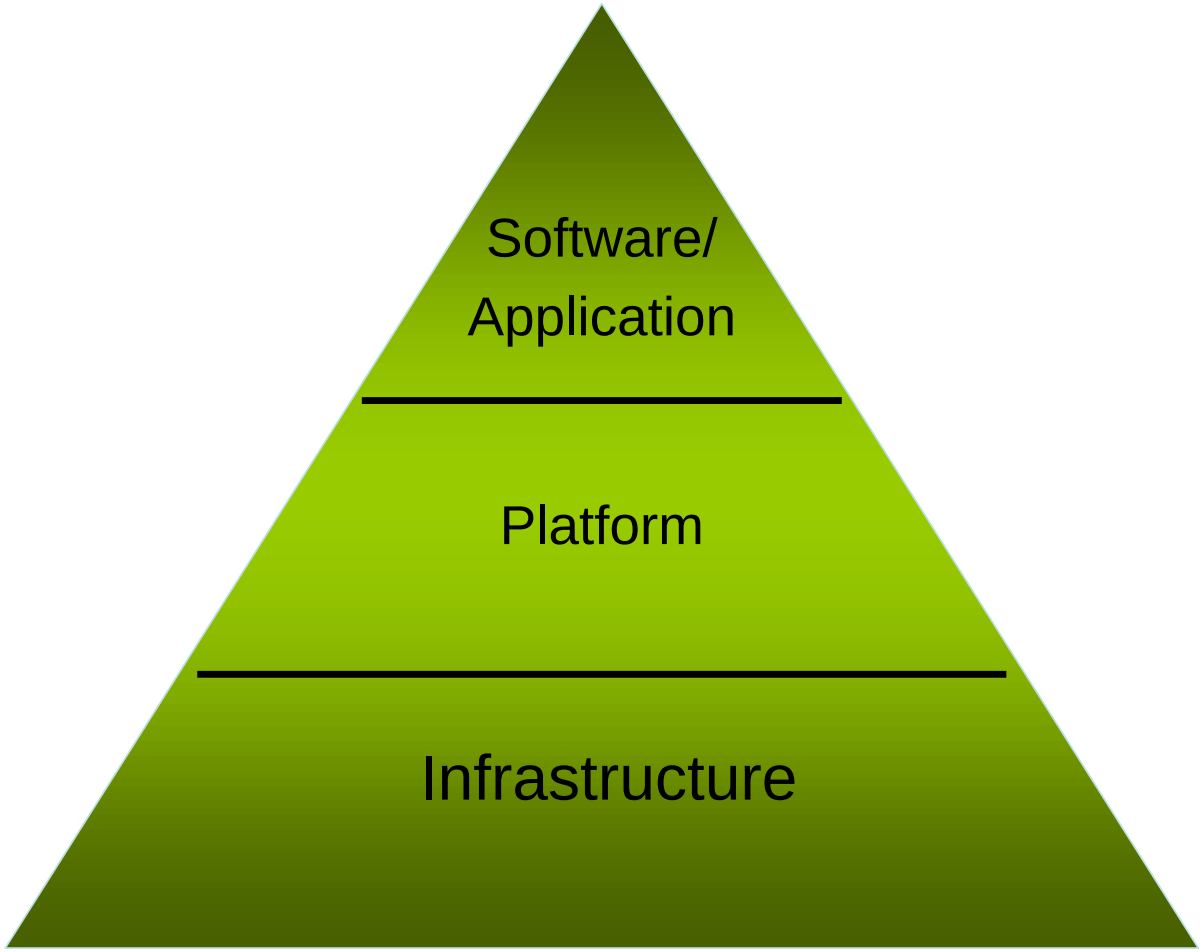
Felhő (szolgáltató)

- A számítási felhő összehozza a szolgáltatás-orientáltságot a virtualizáció skálázhatóságával.
- Egy felhő üzemeltetésénél fontos a virtualizáció, mivel az erőforrásokat fontos elválasztani a fizikai hardvertől, mert különben nagyon nehéz lenne a felhő kezelése.
- A felhő maga hardver, hálózat, háttértár, szolgáltatások, interfészek összessége, ami lehetővé teszi a számítási kapacitás szolgáltatását. A felhő szolgáltatás magában foglalja a szoftver, infrastruktúra és háttértár jelenlétét is az interneten keresztül a felhasználó kérésére.

Felhő (felhasználó)

- Az adott (nagy számítási kapacitást vagy tárhelyet igénylő) művelet nem a saját számítógépen, hanem egy térben elosztott rendszerrel hajthatja végre a felhasználó egy kliensprogramon keresztül.
- A szolgáltatások bármikor igénybe vehetőek, a skálázhatóságnak köszönhetően bővíthető a terhelésnek megfelelően, és a rendszer karbantartását nem a felhasználónak kell elvégeznie.

Service Models



A Felhő rétegei

- Infrastructure as a Service – IaaS
  - Háttértárat és számítási kapacitást nyújt
  - Gyakorlatilag a hardver szolgáltatásként való bérlése.
  - Tartozhat hozzá dinamikus skálázhatóság, hogyha több/kevesebb erőforrásra lenne szükség
  - Egyik legnagyobb szolgáltató: Amazon Elastic Compute Cloud (E2C)

A Felhő rétegei

- Platform as a Service – PaaS
  - Fejlesztői környezetet nyújt, aminek segítségével felhő-kész alkalmazásokat lehet létrehozni általában webes felületen.
  - Bezár a felhőbe – azokat az eszközöket kell használni, amit a felhő nyújt, az exportálás kizárásának lehetősége is előfordul -> Open PaaS
  - Például a Google App Engine, Microsoft Azure

A Felhő rétegei

- Software as a Service – SaaS
  - Adott feladatra szánt alkalmazásokat nyújt
  - Leggyakoribb alkalmazása a CRM (Customer Relationship Management) rendszer
  - Pl. salesforce, facebook, ebay, skype, paypal, youtube, wikipedia, twitter, Google, …


BITPORT
HORN GROUP
horngroup.com
Kommunikáció és közösségi megoldások
twitter
Yammer
SOCIALCAST
lithium
jive
chatter
tibbr
Software as a Service
Marketo
SERVICESOURCE
INTERMEDIA
pivotlink
coupa
workday
OpSource
RightNow
SuccessFactors
ORACLE
Salesforce
Microsoft
SCRIBE
NETSUITE
EnterpriseDB
Taleo
box
SafeNet
zuora
CAST IRON
PERVASIVE
DELL
BOOMI
Intacct
aria
Lombardi
Google
hostanalytics
CORDYS
Appian
INFORMATICA
Platform as a Service
Microsoft
heroku
daptiv
Engine Yard
Google
iTKO LISA
Joyent
intuit
salesforce
MAGIC SOFTWARE
Infrastructure as a Service
SAVVIS
egnyte
Skytap
at&t
hp invent
CITRIX
COLLABNET
IBM
SUNGARD
i365
EMC
nirvanix
SugarSync
GOGRID
amazon.com
rackspace
Microsoft
verizon wireless
Dropbox
T··Systems·

Felhő típusok

- Eddig a Publikus Felhőről volt szó.
- Privát Felhő
  - magasan virtualizált felhő-adatközpont, a (céges) tűzfalon belül jól menedzselhető környezettel
  - a helyi informatikusok rendelkeznek a szolgáltatások és erőforrások kiosztása felett
- Hibrid Felhő
  - a publikus és privát felhők vegyítése. Például hogyha egy vállalatnak tetszik az egyik szolgáltató SaaS alkalmazása, és azt szeretnék használni egy privát felhőben (biztonsági okokból), akkor kapnak hozzá egy VPN megoldást, így már megfelelő számukra a környezet.

Cloud vs Own

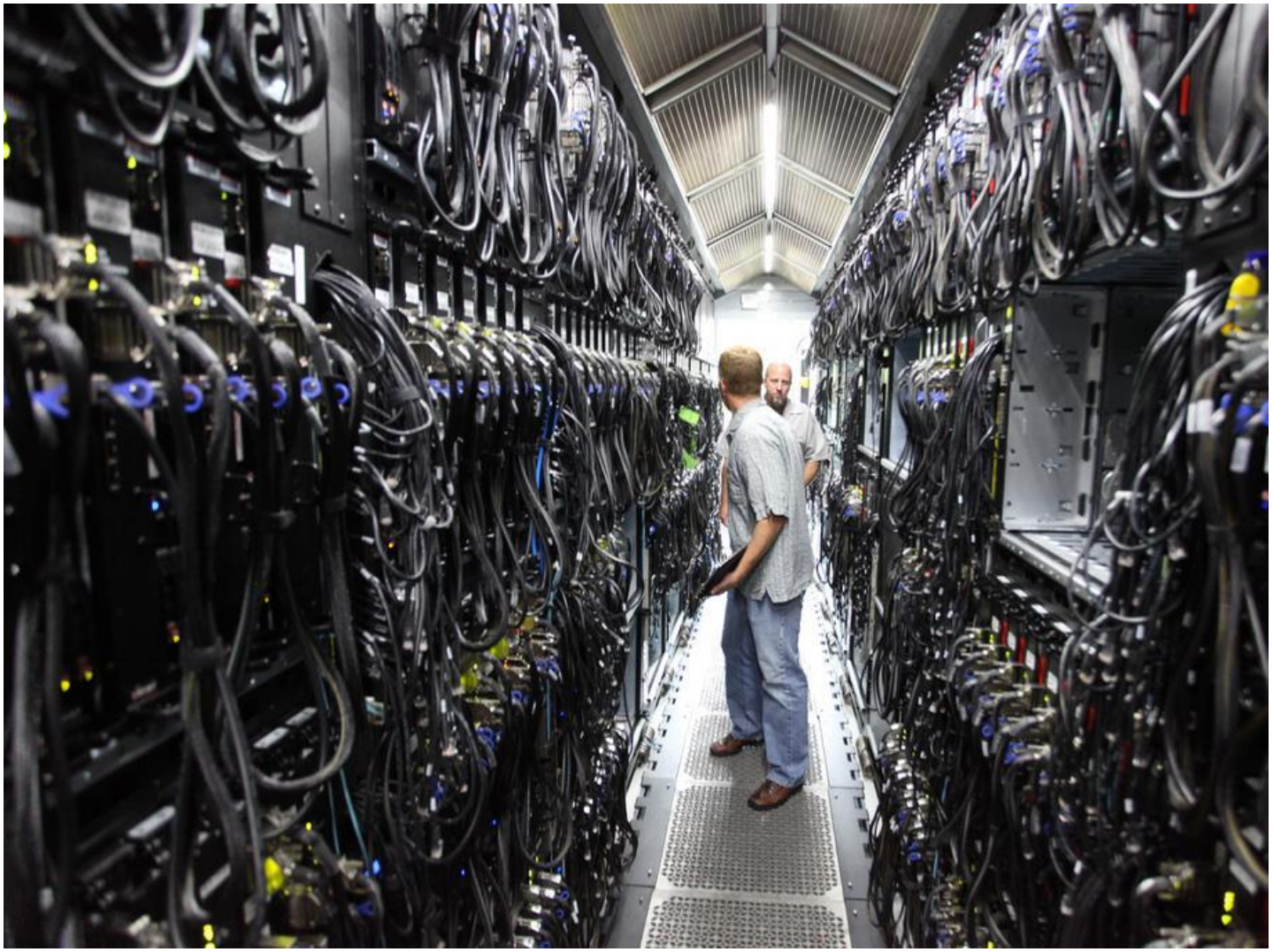
- Service required 128 cores and 524 TB storage for M months
- Cloud (AWS)
  - S3 costs: 0.12$ per GB
  - EC2 cots: 0.10$ per CPU hour
  - Total = Storage + Compute =0.12*524*1000 + 0.10 *1024 * 24 * 30 ~ 136K $
- Own
  - Storage ~ 349K $ /M
  - Total ~ 1555 K $ / M + 7.5K $ (1 admin/ 100 nodes)

Cloud Computing Risks

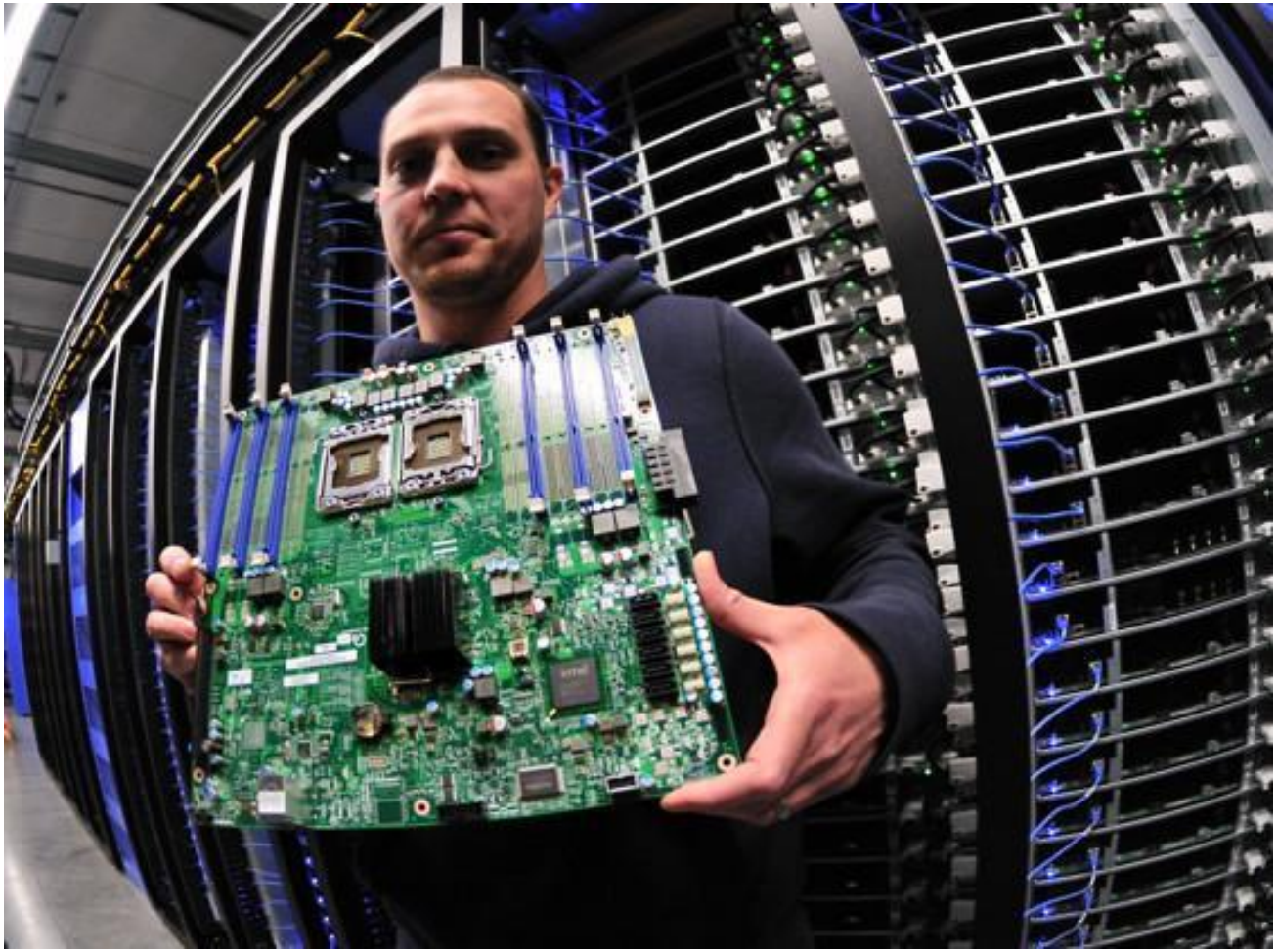
- Risk Areas
  - Service Provider Risks
  - Technical Risks
  - External (Overseas) Risks
  - Management/Oversight Risks
  - Security / Connectivity / Privacy Risks

Cloud Datacenter

Cloud Datacenter

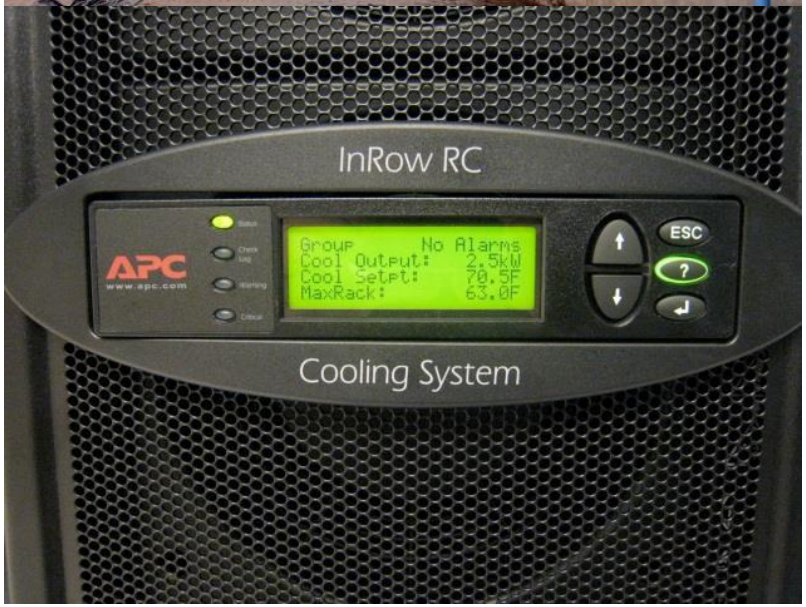


Cloud Datacenter



Cloud Datacenter

Cooling


InRow RC
APC
www.apc.com
Group No Alarms
Cool Output: 2.5kW
Cool Setpt: 70.5F
MaxRack: 63.0F
ESC
Cooling System

Massive scale

- Facebook [GigaOm, 2012]
  - 30K (2009), 60K (2010), 180K (2012)
- Microsoft [NYTimes, 2008]
  - 150K (+10K / months)
- Yahoo! [2009]
  - 100K
- AWS EC2 [Randy Bias, 2009]
  - 40K
- eBay [2012]
  - 50K
- HP [2012]
  - 380K

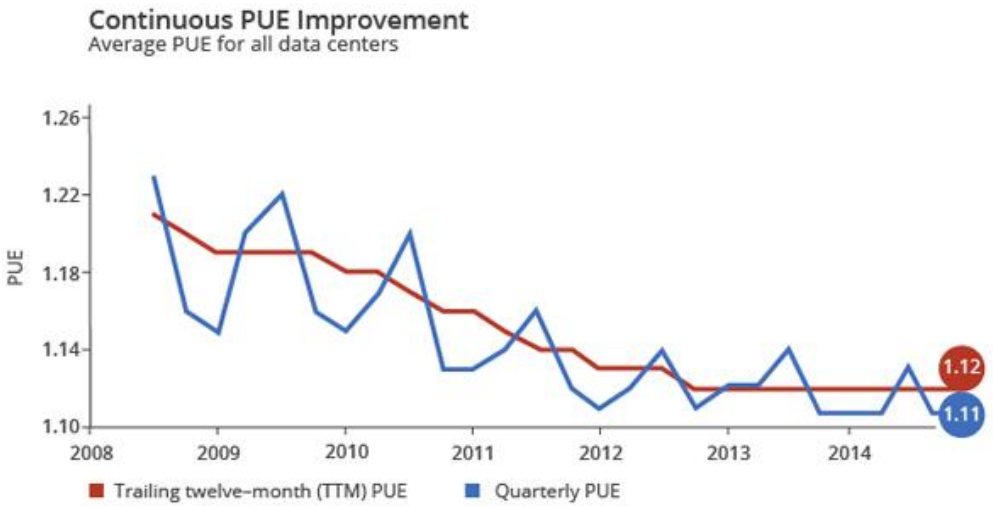
Cloud Datacenter mérőszámok

- PUE (Power usage Effectivness)
  - = Total Facility Power / IT Equipment Energy
- WUE (Water Usage Effectivness)
  - = Water Usage / IT Equipment Energy (L/kWh)
- GPUE (Green Power Usage Effectivness)
  - = G × PUE
  - G = Weighed sum of energy sources (coal, water, solar) and their lifecycle

Cloud Computing and Green IT

PUE	Level of Efficiency
3.0	Extremely Inefficient
2.5	Inefficient
2.0	Industry Average
1.5	Very Efficient
1.0	Ideal

Google datacenters



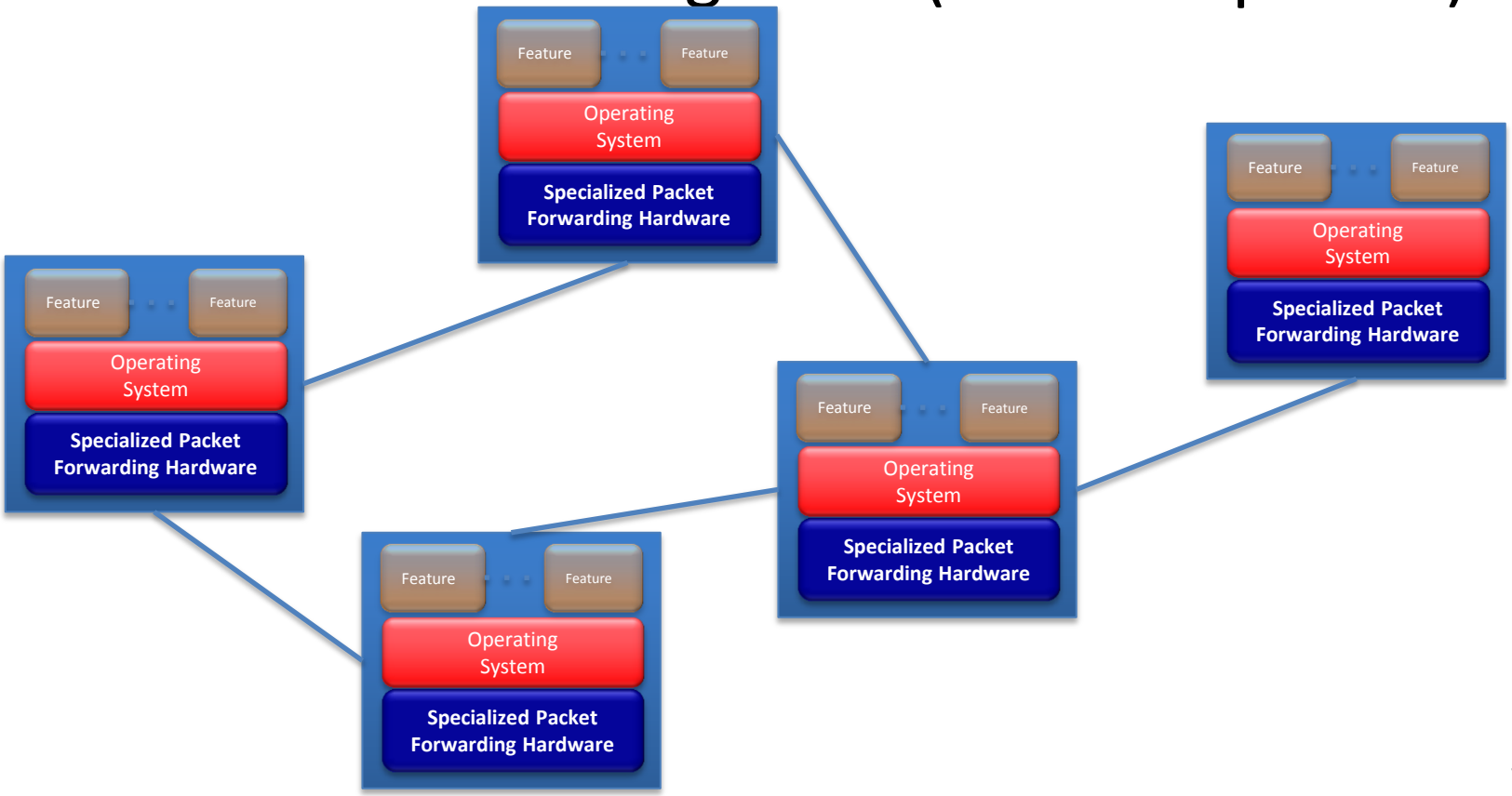
New Cloud Programming Paradigms

- Google: MapReduce
  - Indexing: chain of 24 jobs
  - ~200K jobs processing 50PB/month (2006)
- Yahoo! (Hadoop + Pig)
  - WebMap: chain of 100 MapReduce jobs
- Facebook (Hadoop + Hive)
  - 3K jobs processing 55TB/day

Softwear defined network (SDN)

Classical network architecture

- Hardwear: ports (Forward port 1)
- OS & feature: routing table (Forward packet)



The Networking Industry (2007)

Routing, management, mobility management, access control, VPNs, …

Feature ..... Feature

Operating System

Million of lines of source code — 5400 RFCs — Barrier to entry

Specialized Packet Forwarding Hardware

Billions of gates — Complex — Power Hungry

Closed, vertically integrated, boated, complex, proprietary

Many complex functions baked into the infrastructure

OSPF, BGP, multicast, differentiated services,
Traffic Engineering, NAT, firewalls, MPLS, redundant layers, …

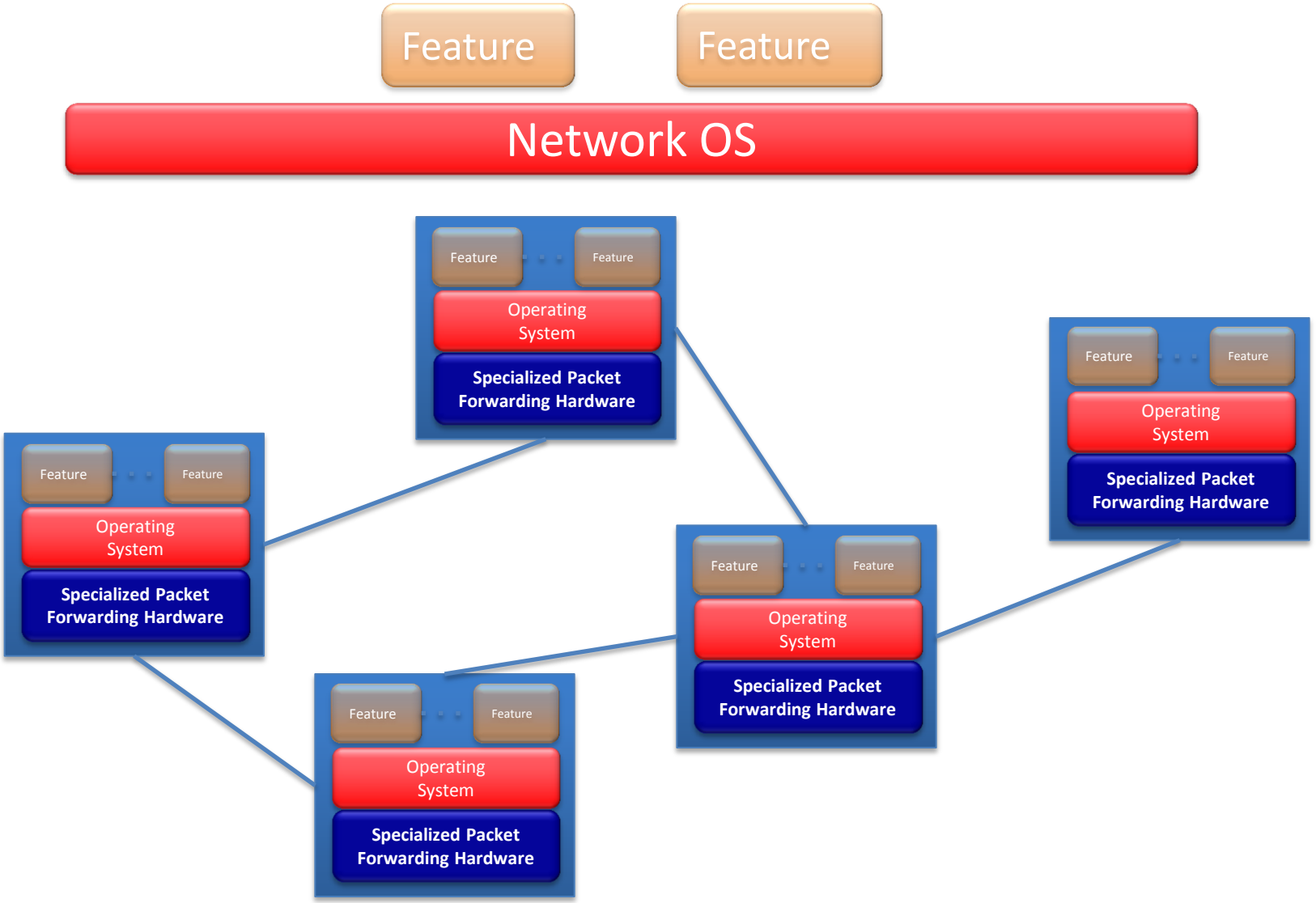
Little ability for non-telco network operators to get what they want

Functionality defined by standards, put in hardware, deployed on nodes

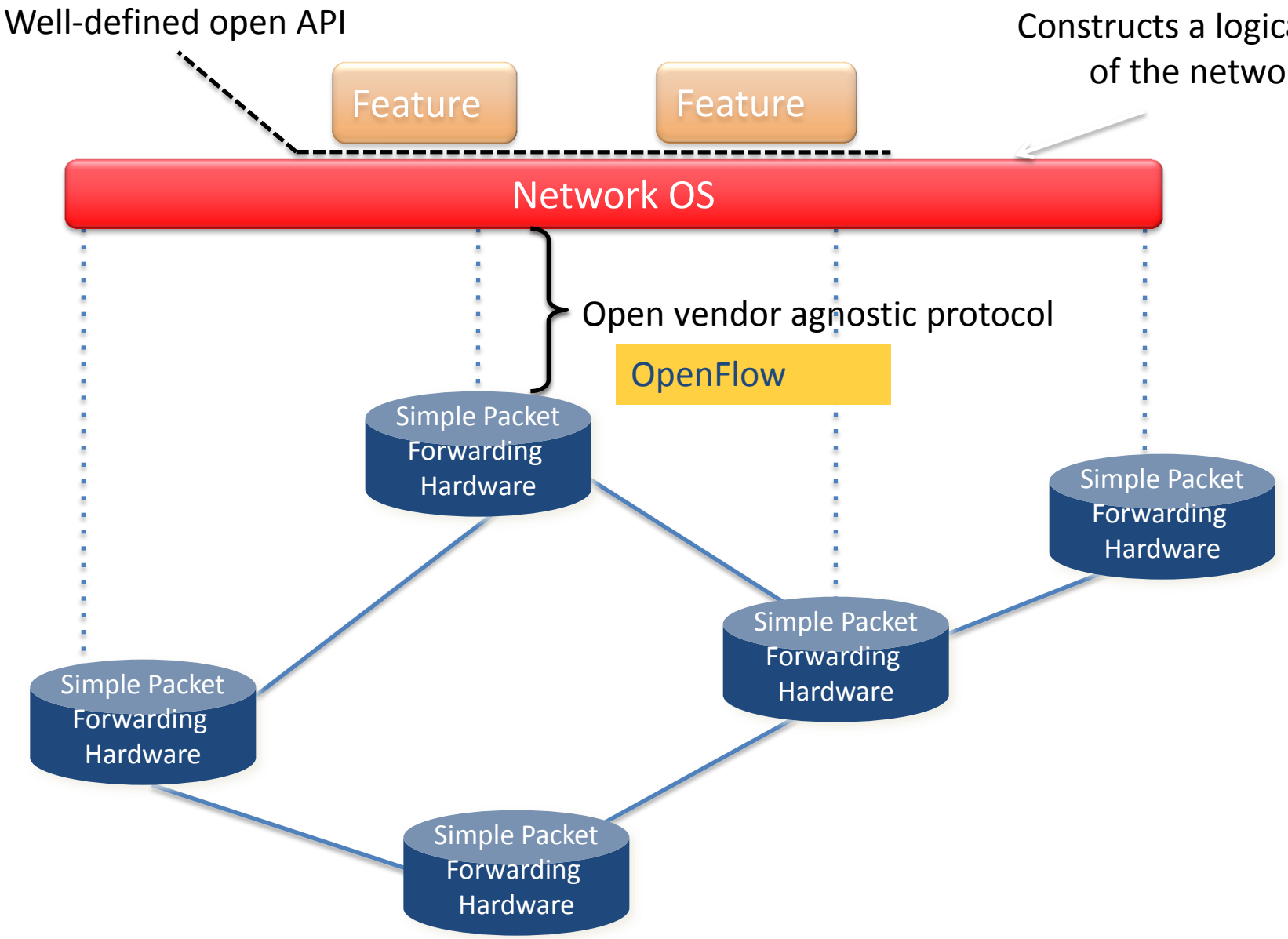
SDN

- Possible definitions:
  - SDN is a new network architecture:
    - that's makes it easier to program networks.
    - with the core idea that software remotely controls network hardware.
    - …

From Vertically Integrated to …



Software Defined Network



Network OS

- Network OS: distributed system that creates a consistent, up-to-date network view
  - Runs on servers (controllers) in the network

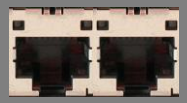
- Uses an open protocol to:
  - Get state information from forwarding elements
  - Give control directives to forwarding elements

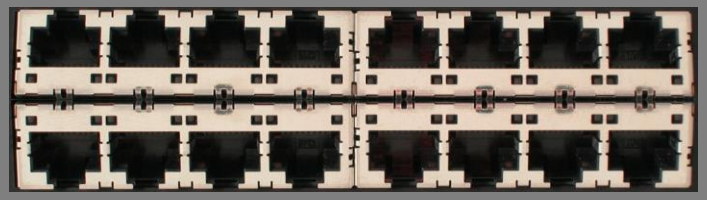
OpenFlow

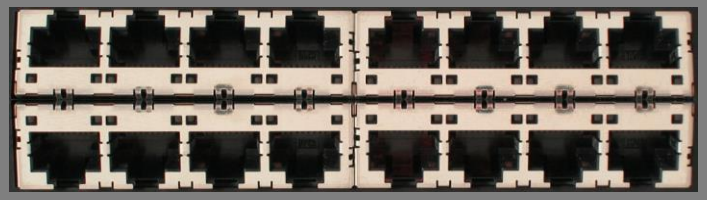
- OpenFlow
  - is a protocol for remotely controlling the forwarding table of a switch or router
  - is one element of SDN

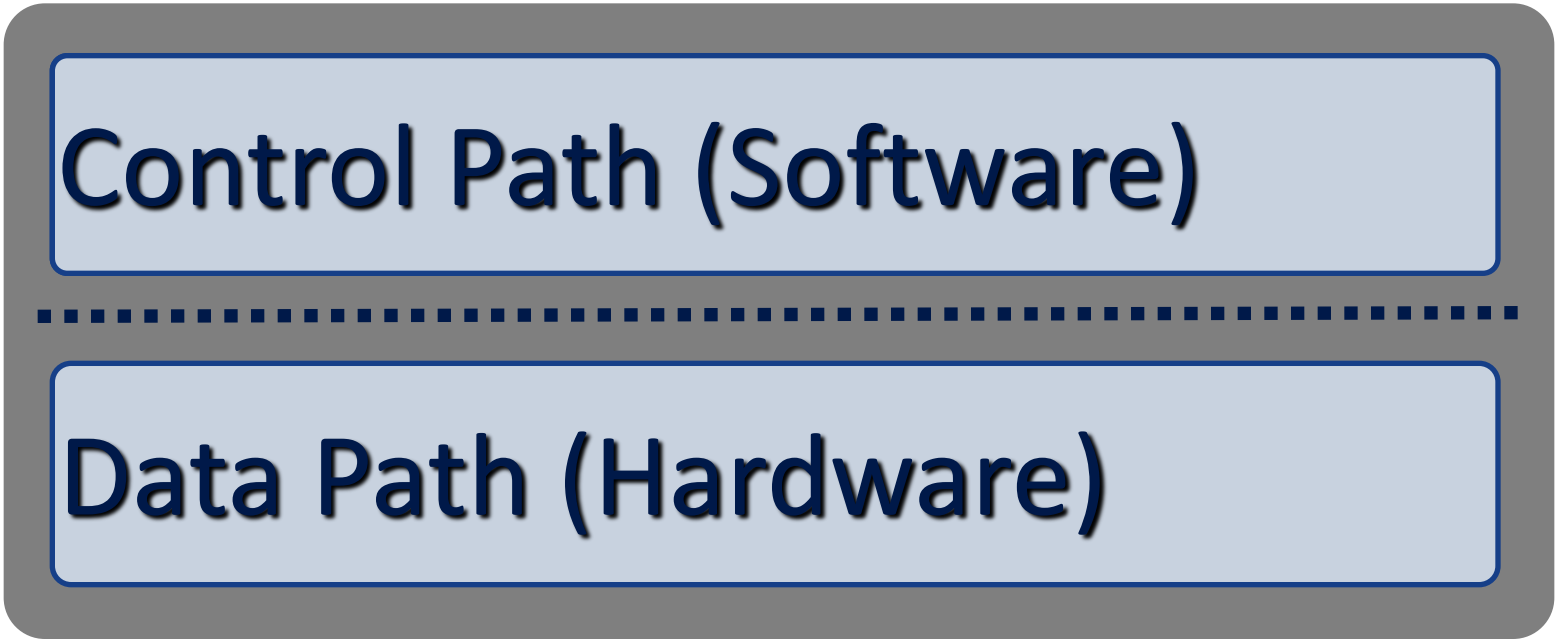
How does OpenFlow work?

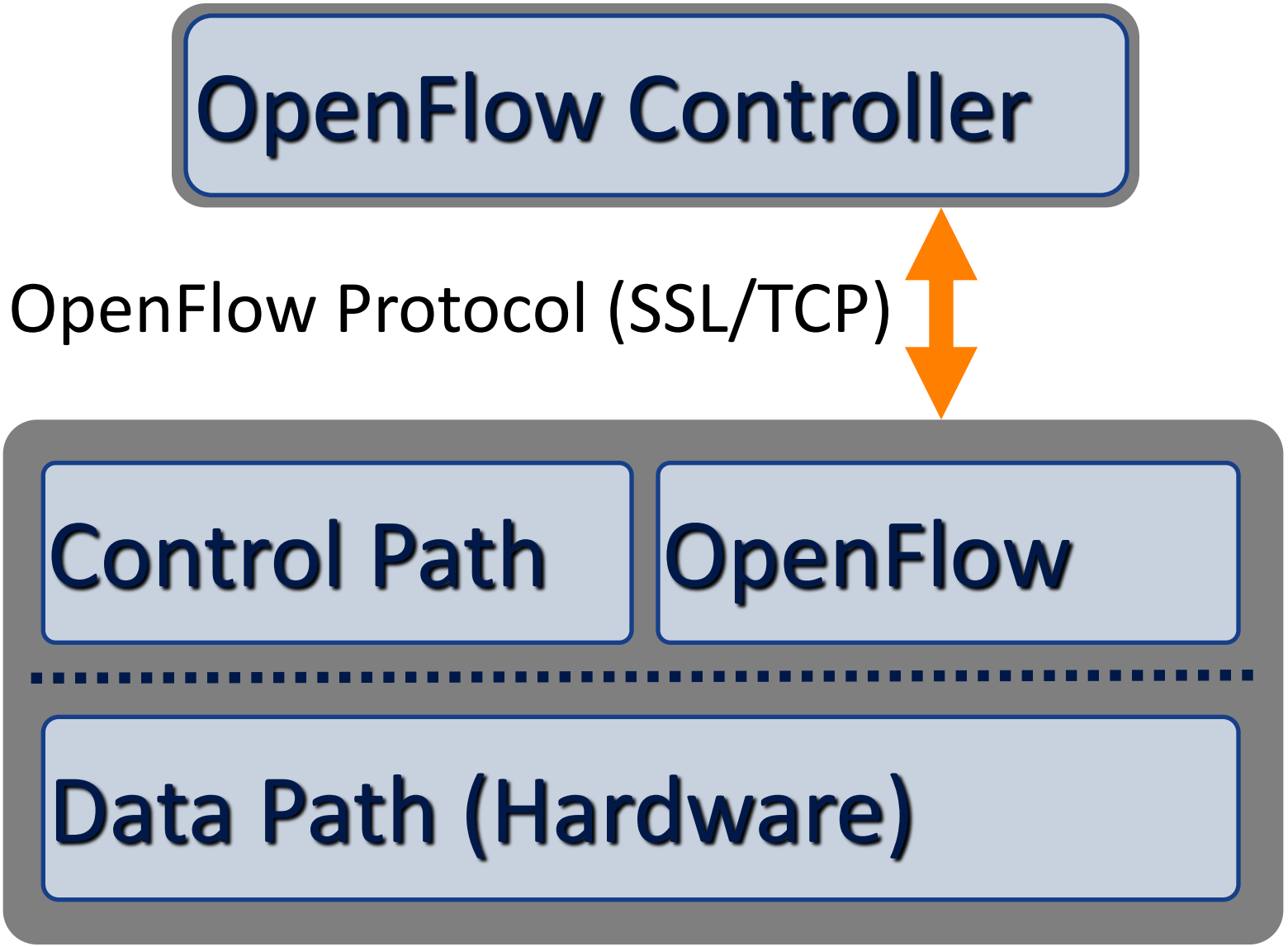
Ethernet Switch



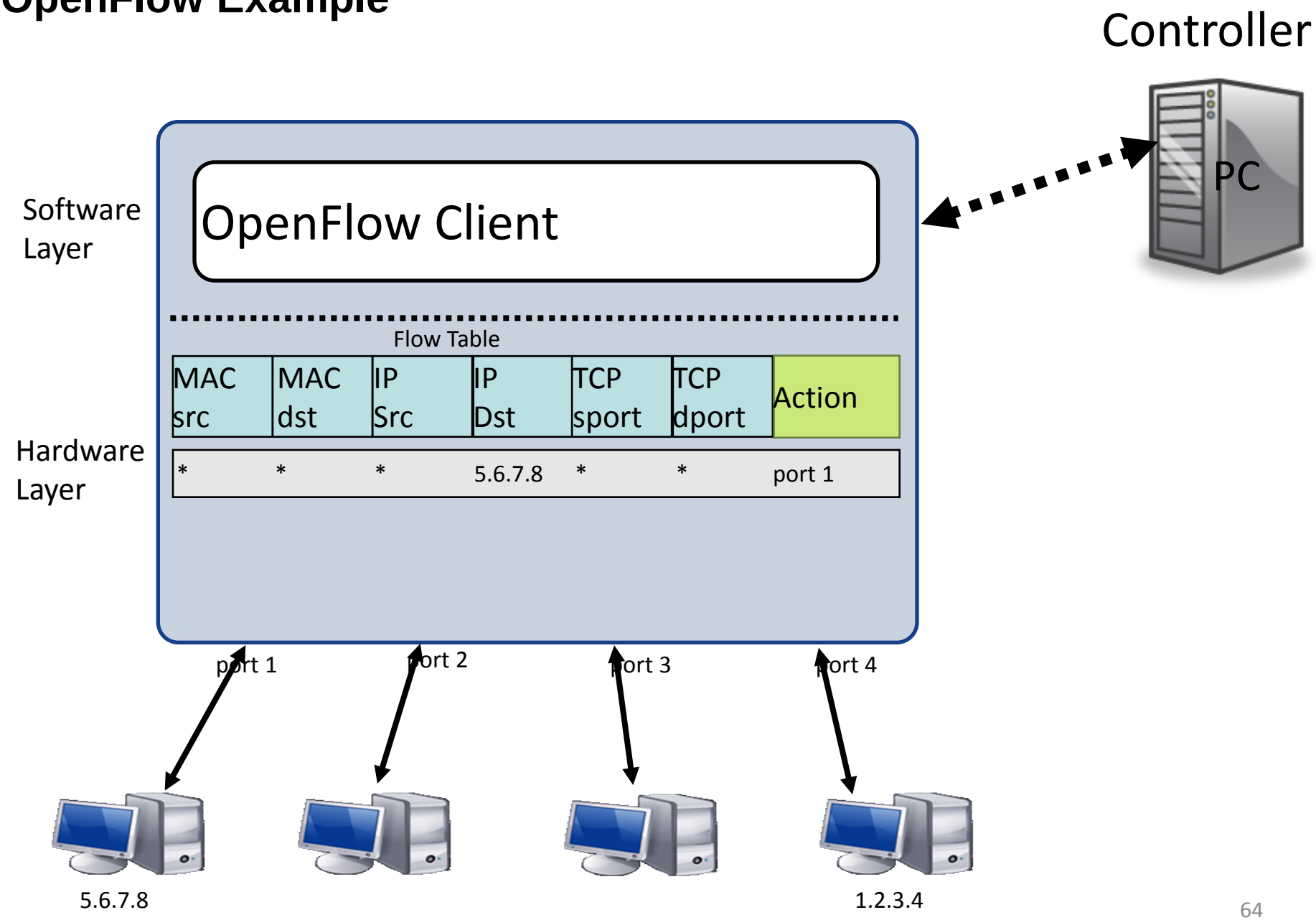




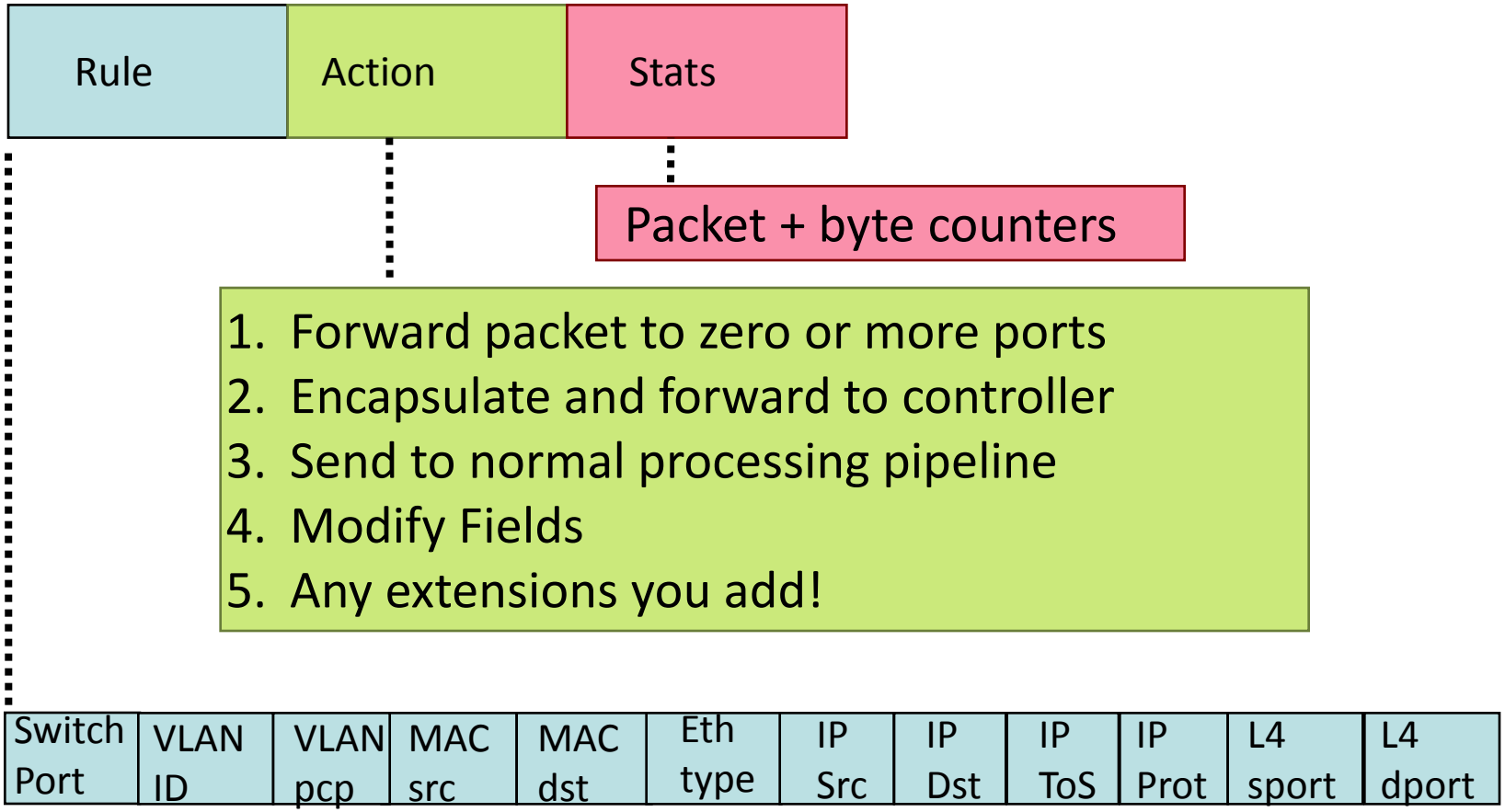

Control Path (Software)
Data Path (Hardware)


OpenFlow Controller
OpenFlow Protocol (SSL/TCP)
Control Path
OpenFlow
Data Path (Hardware)

OpenFlow Example



OpenFlow Basics
Flow Table Entries



+ mask what fields to match

Examples

Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f:..	*	*	*	*	*	*	*	port6

Flow Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
port3	00:20..	00:1f..	0800	vlan1	1.2.3.4	5.6.7.8	4	17264	80	port6

Firewall

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	*	*	*	22	drop

Examples

Routing

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	*	*	*	*	5.6.7.8	*	*	*	port6

VLAN Switching

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	IP Src	IP Dst	IP Prot	TCP sport	TCP dport	Action
*	*	00:1f..	*	vlan1	*	*	*	*	*	port6, port7, port9

Secure Channel

- SSL Connection, site-specific key
- Controller discovery protocol
- Encapsulate packets for controller
- Send link/port state to controller

Main Concepts of Architecture

- Separate data from control
  - A standard protocol between data and control
- Define a generalized flow table
  - Very flexible and generalized flow abstraction
  - Open up layers1-7
- Open control API
  - For control and management applications
- Virtualization of the data and control plane
- Backward compatible
  - Though allows completely new header

Other SDN Use Cases

- Energy conservation, routing, and management in data centers
- Seamless use of diverse wireless networks
- Network based load balancing
- Traffic engineering
- Slicing and scalable remote control/management of home networks
- Experimentation with new approaches and protocols using selected production traffic
- Run virtual shadow network for traffic analysis and re-configuration
- And many more …

See http://www.openflow.org/videos/

Current status of SDN

- Hardware support

Juniper MX-series

NEC IP8800

WiMax (NEC)

HP Procurve 5400

Netgear 7324

PC Engines

Pronto 3240/3290

Ciena Coredirector

More coming soon…

Köszönöm a Figyelmet!

Források:

- http://cse.unl.edu/~ylu/csce351/notes/VirtualMachines.ppt
- https://www.event-plans.com/Uploads/files/GSA/Fred%20Wuensch%20-%20Total%20Virtualization%20and%20Cloud%20Computing.ppt
- http://www.vkontrole.lt/wgita/materials/wgita/Cloud%20Computing_USA.ppt
- http://www.csd.uoc.gr/~hy490-31