Virtualization & Cloud Computing

Corso di: Big Data Architectures

Prof. Paolo Nesi

Department of Systems and Informatics, University of Florence
Via S. Marta 3, 50139, Firenze, Italy
tel: +39-055-2758515, fax: +39-055-2758570

DISIT Lab, Sistemi Distribuiti e Tecnologie Internet

http://www.disit.dinfo.unifi.it/

paolo.nesi@unifi.it
http://www.disit.dinfo.unifi.it/nesi

Structure

- Elements of cloud Computing
  - Motivation and definitions
- Virtualization
- Cloud Computing
- High Availability
- vSphere Infrastructure
- Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions
- PaaS Solutions
- SaaS Solutions
- ICARO project
- Container

Datacenter, definition

- **Datacenter**
  - A computer factory/farm in which servers/computers are called HOSTS and are hosted and organized:
    - power, net maintenance, etc.
    - As: industrial computers, blades
  - They can be exploited for private purposes, grid, cloud computing, renting/hosting, etc.

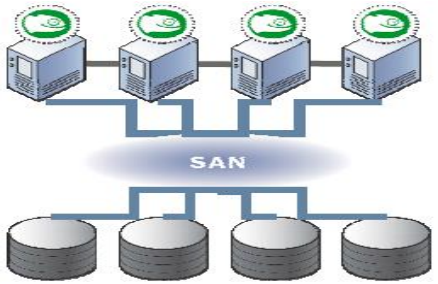
Server

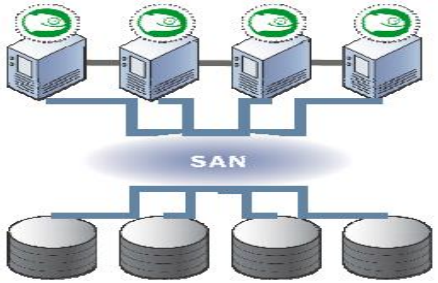
- blade

- server

Infrastructure, definition

- A set of datacenter and clusters
  - Cluster: group of servers plus storage (if any)
- A storage as a set of
  - NAS: Network Area Storage
  - SAN: Storage Area Network
- Etc..





SAN vs NAS

- In most cases they are wrongly considered the same staff
  - It is true that: NAS/SAN typically are HD in Raid connected/shared to/by Servers
- **NAS: Network Area Storage**
  - Network HD sharing content at level of File via HTTP, NFS, CIFS, etc., protocols
  - Multiple servers may access to them, the HD can be easily mounted, and see on the OS
  - Reduced performances with respect to SAN
- **SAN: Storage Area Network**
  - HD system segmented in LUN, mounted by the Server/VM OS at level of disk block
  - The OS accessing to them has to format with its own file system
  - Multiple server may access to them
  - Possible protocols: low level IP, iSCSI, Fiber, etc.

Consideration

- The fact that we have servers and storage is not a key for cloud
- Cloud is not a synonimous of data center
- Since 60' **John McCarthy supposed to have a number of servers providing services for all**
- Servers have been typically used
- Close to the end of the 90', the first Software as a Service, SaaS, started.
  - Since…..

Housing

- Moving HW, Moving servers from local premise to an infrastructure:

- Buying HW and finding a safe location to placing,
  - Keeping HW on maintenance, H24
  - Keeping HW on Power and UPS, Continuity, H24
  - Keeping HW on constant temperature, H24
  - Keeping HW on safe, H24

Motivations for Cloud computing and Virtualization

- **Computer Based Physical Systems (servers)** are
  - not flexible:
    - The HW/SW has to be dimensioned for the workload worst case and not for the real/typical one
    - HW/SW can be hardly reused for different purposes since specific allocation is performed
  - subjected to HW failure, and thus:
    - Software has to be installed again when changing hardware
    - OS has to be reinstalled in most cases

Motivations for Cloud computing and Virtualization

- **Computer Based Physical Systems (servers)** are
  - Subjected to high costs of setup, installation and use:
    - Power, conditioning, network connection, etc.
  - not scalable, to scale up frequently implies hardware change/re-engineering, (it was the gold period of the Moore's Law, of performance duplication every two years)
    - Data duplication, migration,
    - sharing databases, etc.
    - Distribution of activities among HW
    - Balancing or workload,
    - Add more HW,

Final Motivations

- Reduction of costs for HW and operating system SW maintenance.
  - High costs to guarantee high availability, HA: 99,999% of up time
  - High costs to guarantee high reliability
  - High costs to follow the HW/SW technological trends for performances, computational needs
  - Critical mass is needed to justify HW costs
- Sharing resources
  - Among for multiple applications and solutions
- Needs of High flexibility in terms of features
  - Most of the SW is becoming a services (licensing per year/month or users) and not anymore a product.
  - Many vendors provide complex SW systems in terms of services, on the basis of their consumption via network connection

Hosting, definition

- **HPC: high performance computing**
  - Solution based on cloud/grid for parallel execution of algorithms and tools.
- **Hosting web portal into a datacenter/cluster**
  - Renting a web space via some SLA (service level agreement), contract, monthly rate to publishing web pages: service httpd
  - Additional services: mysql, php, asp, ftp, ssh, https, etc..
  - Features:
    - Space on disk, networking
    - Domain space, etc.

- **Hosting a machine (computer/VM) into a datacenter**
  - According to some SLA, contract…
  - Renting a Computer/VM/Cluster into a data center

Service Level Agreement

Availability %	Downtime per year[2]	Downtime per month	Downtime per week	Downtime per day
55.5555555% ("nine fives")	**162.33 days**	**13.53 days**	**74.92 hours**	**10.67 hours**
90% ("one nine")	36.53 days	73.05 hours	16.80 hours	2.40 hours
95% ("one and a half nines")	18.26 days	36.53 hours	8.40 hours	1.20 hours
97%	10.96 days	21.92 hours	5.04 hours	43.20 minutes
98%	7.31 days	14.61 hours	3.36 hours	28.80 minutes
99% ("two nines")	**3.65 days**	**7.31 hours**	**1.68 hours**	**14.40 minutes**
99.5% ("two and a half nines")	1.83 days	3.65 hours	50.40 minutes	7.20 minutes
99.8%	17.53 hours	87.66 minutes	20.16 minutes	2.88 minutes
99.9% ("three nines")	**8.77 hours**	**43.83 minutes**	**10.08 minutes**	**1.44 minutes**
99.95% ("three and a half nines")	4.38 hours	21.92 minutes	5.04 minutes	43.20 seconds
99.99% ("four nines")	**52.60 minutes**	**4.38 minutes**	**1.01 minutes**	**8.64 seconds**
99.995% ("four and a half nines")	26.30 minutes	2.19 minutes	30.24 seconds	4.32 seconds
99.999% ("five nines")	**5.26 minutes**	**26.30 seconds**	**6.05 seconds**	**864.00 milliseconds**
99.9999% ("six nines")	**31.56 seconds**	**2.63 seconds**	**604.80 milliseconds**	**86.40 milliseconds**
99.99999% ("seven nines")	**3.16 seconds**	**262.98 milliseconds**	**60.48 milliseconds**	**8.64 milliseconds**
99.999999% ("eight nines")	**315.58 milliseconds**	**26.30 milliseconds**	**6.05 milliseconds**	**864.00 microseconds**
99.9999999% ("nine nines")	**31.56 milliseconds**	**2.63 milliseconds**	**604.80 microseconds**	**86.40 microseconds**

			99,8	99,98	99,998	99,999	
			0,2	0,02	0,002	0,001	
	60						
ore	24		17,52	1,752	0,1752	0,0876	ore anno
	365						
minuti	525600		1051,2	105,12	10,512	5,256	minuti
	100	0,2					
	525600	X					
			35,04	3,504			

Infrastructure as a Service, IaaS

- the provision of infrastructure with a relevant processing power, storage, network and other basic resources independently of application services of any type.
  - Example: a server in rent, a VM for rent

- Thus, it is possible to use an infrastructure of a provider to run your application, against a payment based on the renting (consumption) of the infrastructure, leaving the responsibility of the HW, OS, and SW (libraries, middleware, backup, storage, network, etc.) as well as of the application itself to the provider.
  - Amazon EC2 is an example of IaaS service.

Motivations for Applications on the Cloud

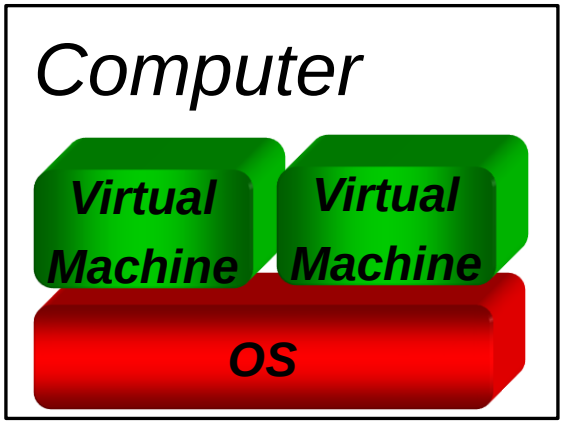
- Single-tier applications, such as small web portals, do not demand a complex system, and
  - do not justify the acquisition of a host server !!!???
- Multi-tier applications have multiple servers, such as:
  - High performance web server, high number of users
  - SN: Social networks
  - CMS: content management systems
  - CRM: customer relationship management
  - CDN: Content delivering network
  - ERP: enterprise resource planning
  - P2P torrent tracker: see Piratebay
- *All typically parallel applications that may need grid, computation, storage, etc. may run on cloud, etc.*

Structure

- Elements of cloud Computing
- Virtualization
  - emulation, para-virtualization
  - virtual resources
  - snapshots
- Cloud Computing
- High Availability
- vSphere Infrastructure
- Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions, SaaS Solutions, PaaS Solutions
- ICARO Project
- Container

Concept of Virtual Server

- Hyp:
  - you can make the image (in a file) of the HD of a Physical Server (with its Operating System and files) and
  - put in execution the OS from the HD (file) by a program of "emulation“



- Thus we have:
  - a computer with an operating system OS
  - that hosts a program of "emulation“,
  - which in turn, it is capable to run in separate processes the images of the corresponding OS VM

Typical Features for renting a Computer or a Virtual Machine into a Datacenter (1/2)

- **Requesting** to have one or more Computers/Hosts and/or VMs
- **Hardware**:
  - CPU: 32/64 bit, number of cores/CPU, frequency of work, intel/amd, etc.
  - RAM memory: size and frequency of work
  - Power supply: fault-tolerant or not; with UPS or not, etc..
  - HD: space, speed (7.2, 15Kgiri), security level/RAID, type SAS/SATA, SCSI
  - NAS/SAN, Network area storage/SAN, fiber/internet: size, RAID, etc.
  - Network features:
    - Number of connections/cards, NICs
    - number of IP addresses, static/DHCP
    - Transfer rate: minimum guaranteed, maximum possible, down/upload
    - Maximum transferred bytes: per day, per month, etc.

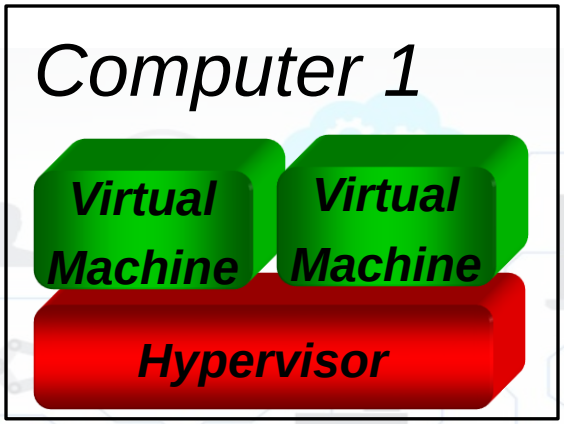
Typical Features for renting a Computer or a Virtual Machine into a Datacenter (2/2)

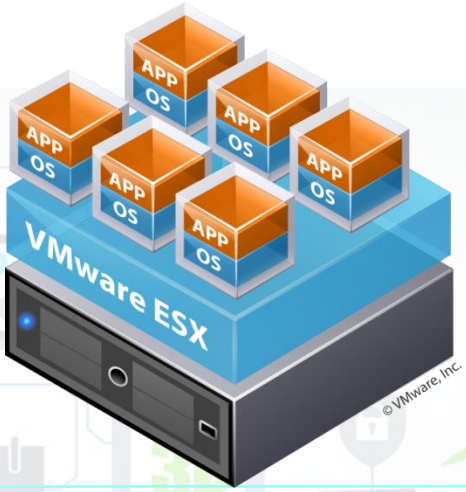
- **Software**:
  - Operating systems: versions, with maintenance, etc.
  - Software preinstalled into the Computer/VM, see in the following

- **Services**:
  - Periodic back up: details on HD space
  - Access to VM/Computer: remote desktop or KVM tool
  - Reboot or not of the Server, for example via Plesk.

Virtual Machine, Virtualization

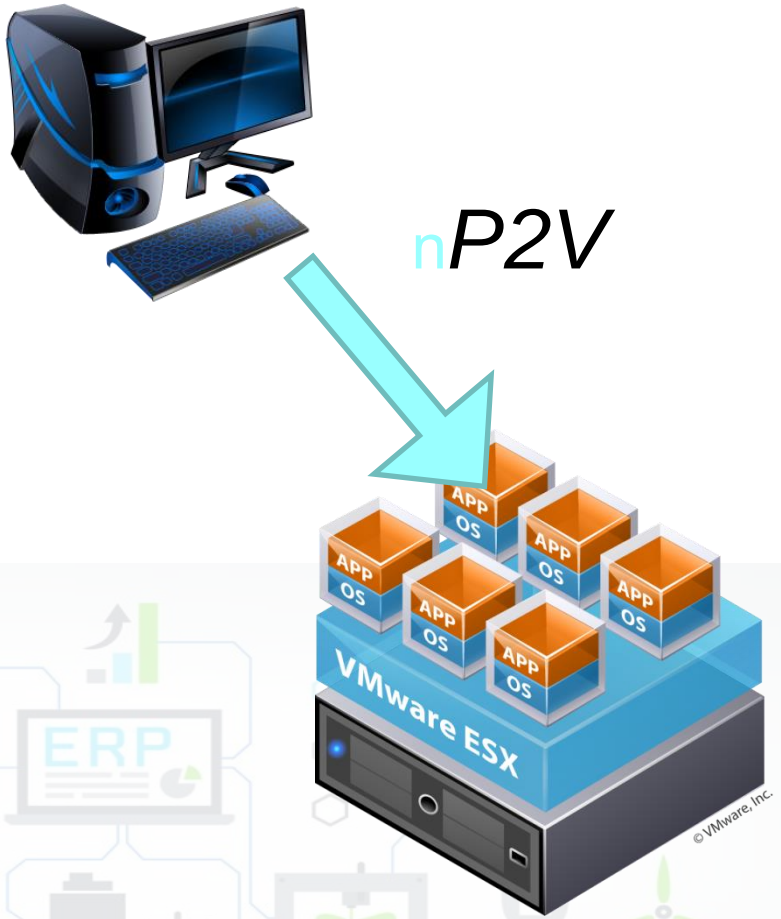
- **Virtual Machine:**
  - An image of an operating system that can be put in execution into a real host/computer creating a virtual computer that exploits a part of all of the host resources
  - E.g.: Host may be Linux-like while the VM may be Window, Mac, Linux, …
- **Virtualization**
  - Transforming a physical computer into a VM, virtual machine, hosted on some Host computer
- **Hypervisor** (VM Monitor) to manage the several VMs on the host





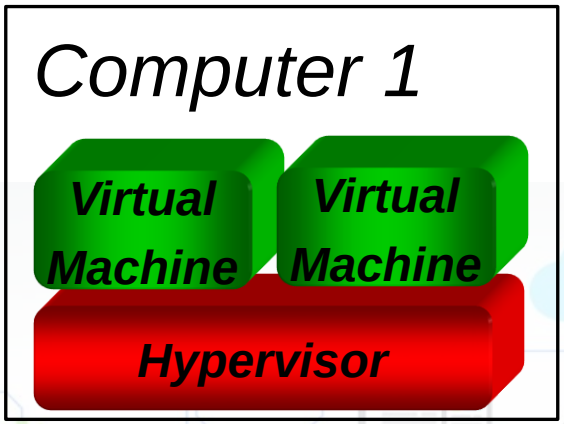
The Virtualization

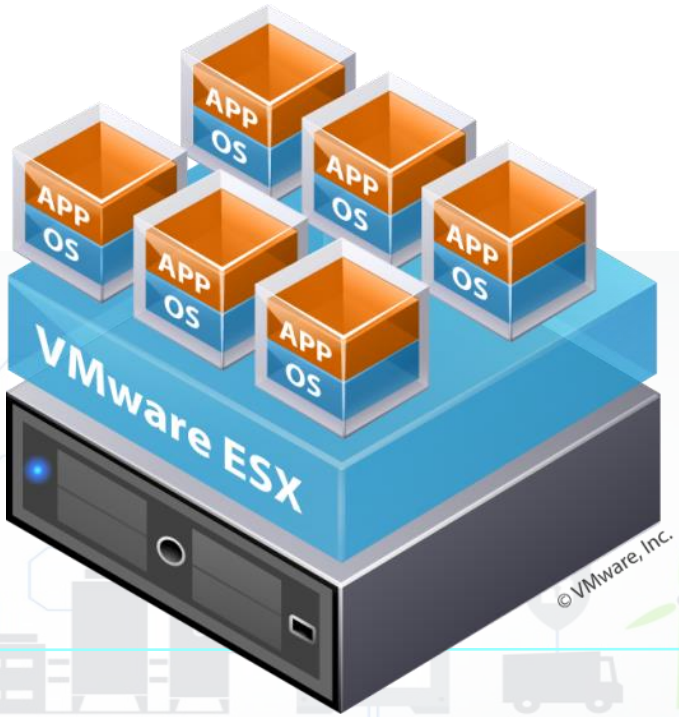
- Process by which it is possible to create a virtual machine, VM,
- For example by:
  - An installation DVD
  - Cloning a physical machine (P2V)



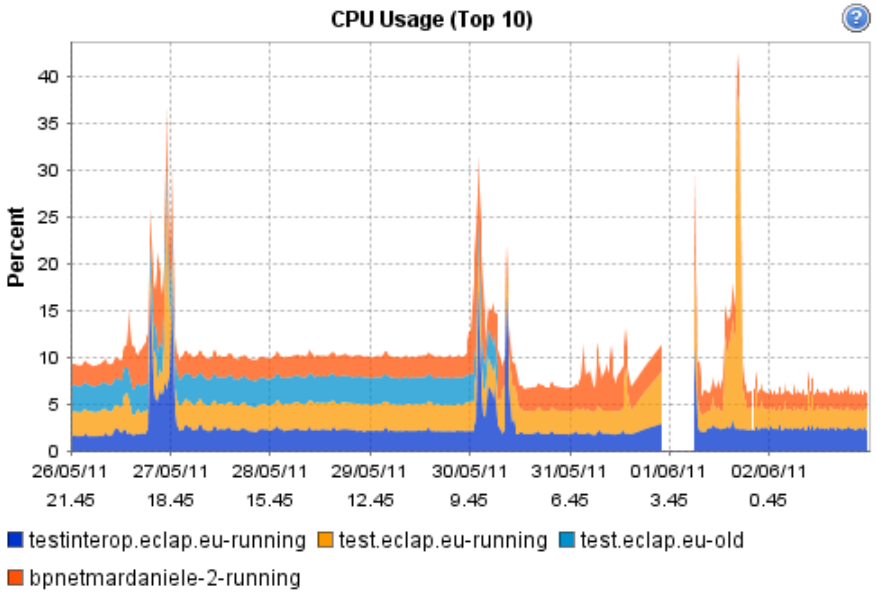
The Hypervisor

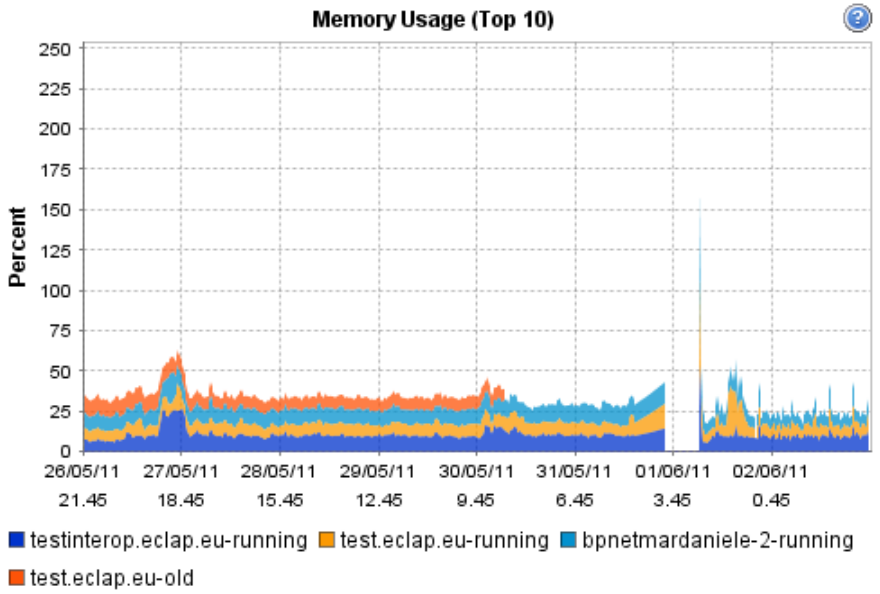
- It is the essential OS, that can be installed in a Host and can put in execution and manage one or more VM
- The Host is a server HW that host the Hypervisor
- ESX of VMware is an Hypervisor.

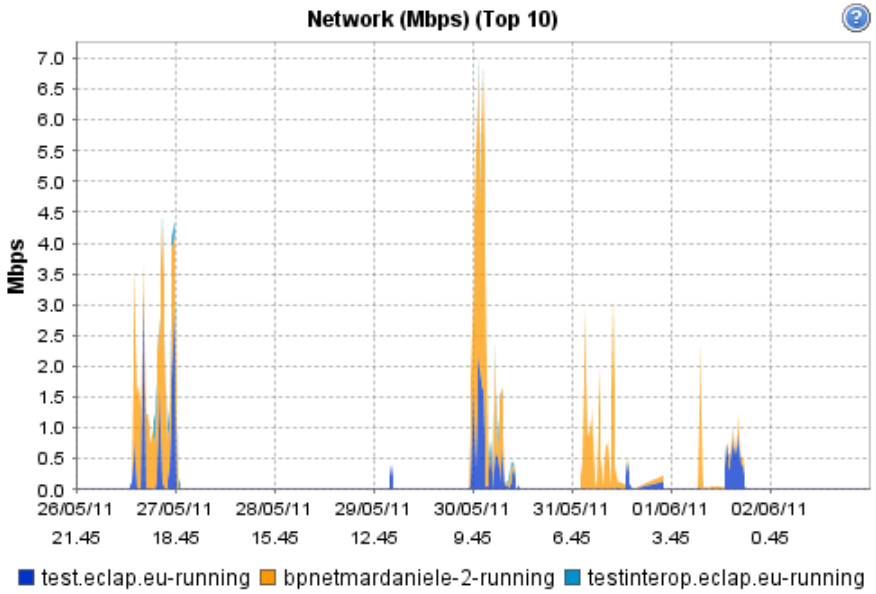


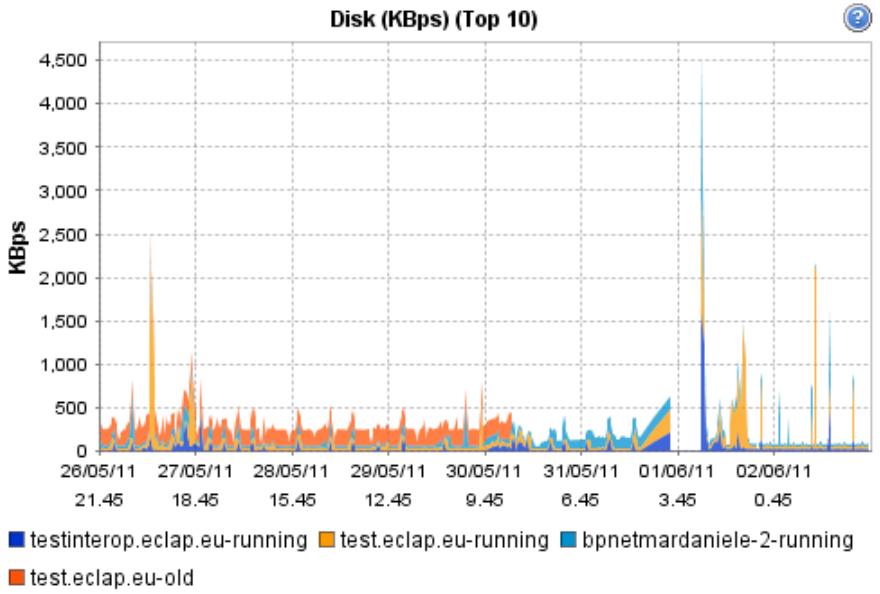


Performance Analysis of VM on the Host









Motivations for Cloud computing and Virtualization

- **Most of the Host/datacenter capabilities** are not exploited at 100% in every time instant
  - They are typically present in large industries/institutions or in large services: google, amazon, tiscali, Dada, ibm, cnr, etc
  - Their size is typically defined on the basis of the workload worst case
- ***If they are big***: the exploitation of the remaining resources for cloud/hosting is a solution to recover money, since a non working machine (like a caw) is a costs without any return.
- ***If they are small:*** it could be a solution to host machines on a professional infrastructure to reduce annual costs for HW/SW
  - Delegating to cluster owners the costs for *maintenance, renovating hardware, renovating software, network costs, back up costs, power supply, etc.*

Benefits of Virtual Machine

- **Main benefits:**
  - Separation of OS+SW with respect to the needed HW
  - Exploiting legacy solutions which can be wrapped into a VM and protected with a physical firewall without reinstalling and recompiling old applications.
  - The simple upgrade can be obtained by giving more CPU instead of changing HW and reinstalling all SW elements.
- **For example:**
  - An old Linux Server hosting several web portals with a old versions of: MySQL, PHP, etc.. and many configuration aspects: users, mailing lists, etc. very time consuming to port on a new server
  - An old Cobol application running only on an old Windows 2000 Server, which cannot be recompiled into a new Windows Server 2008 at 64 bits without spending months of work.
  - A Cobol application running on machine based on an IBM system 36….
  - Etc.

VM on PC workstations

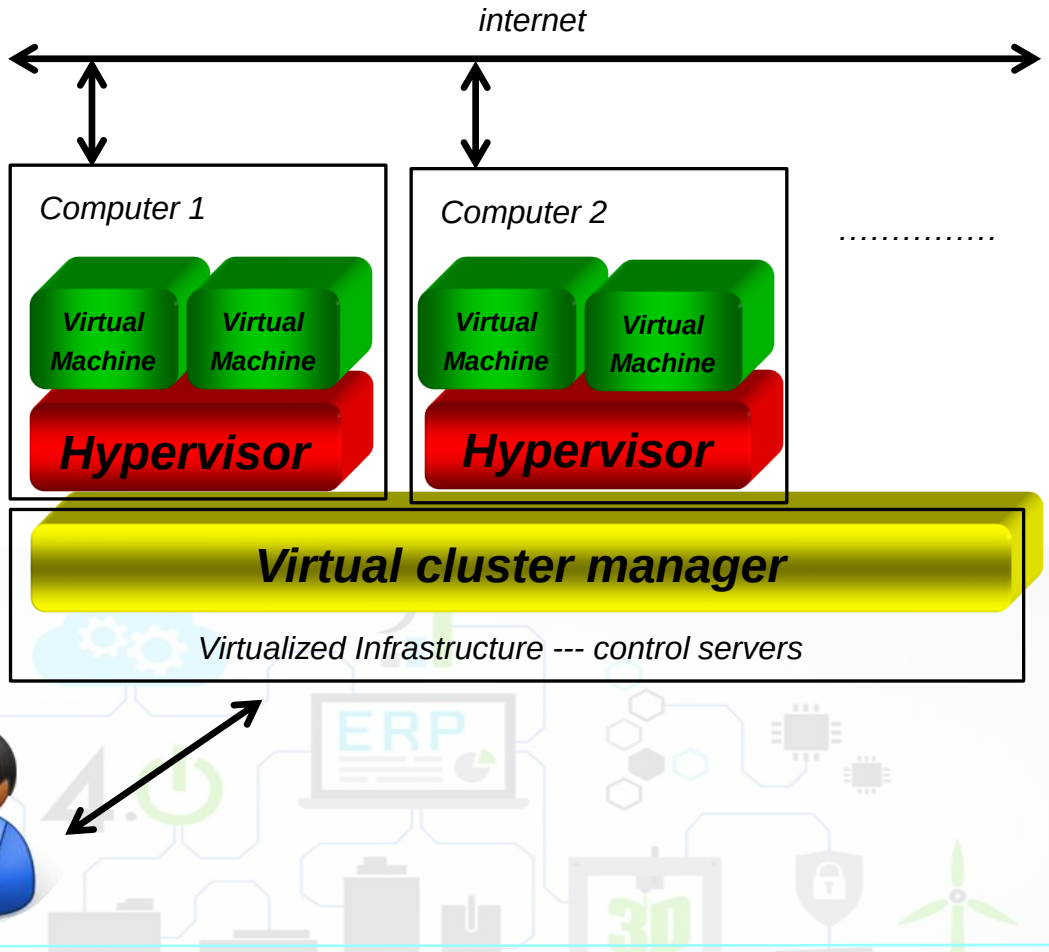
- On a single Computer it is possible to put in execution a VM by using a standalone Hypervisor
- **VMware Workstation** can play on Windows:
  - VMware Virtual Machines with: MS Win, Linux, Mac, ChromeOS, etc.
- **VMware Player** can play on Windows or Linux
  - Free of charge
  - VMware Virtual Machines with: MS Win, Linux, Mac, ChromeOS, etc
- **VMware Fusion** can play on MAC:
  - VMware Virtual Machines with: MS Win, Win 7, ChromeOS, etc.
- **Microsoft Virtual PC** on Windows 7:
  - Free of charge
  - Create VM with Win XP, Linux Ubuntu,

Software/Components preinstalled into rented Computers/VMs

- **Several kinds of software components/tools** that can be accessible on rented VM and/or Computers.
- **Availability of HW/SW**: for example at 98% or 99.999%
- **Typical components** that could be requested:
  - DB: MySQL
  - FTP: server and client
  - Web Server: Apache, IIS
    - Add-on: PHP, Perl, Python, cache tools on several levels
  - SMTP address, antispam
  - Web Application Server: TomCat, ….
  - And: Antivirus, backup, email, Drupal, Joomla, etc…
- **A full server can be customized, so that any other tool can be installed as well from the user**

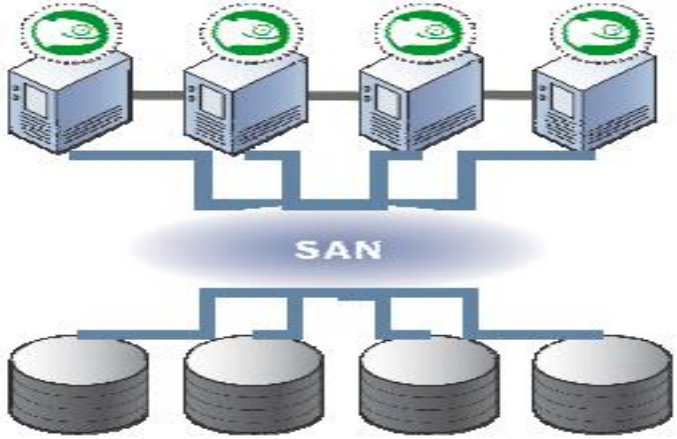
Cloud Computing with VMs

Several Hypervisors on a Clusters



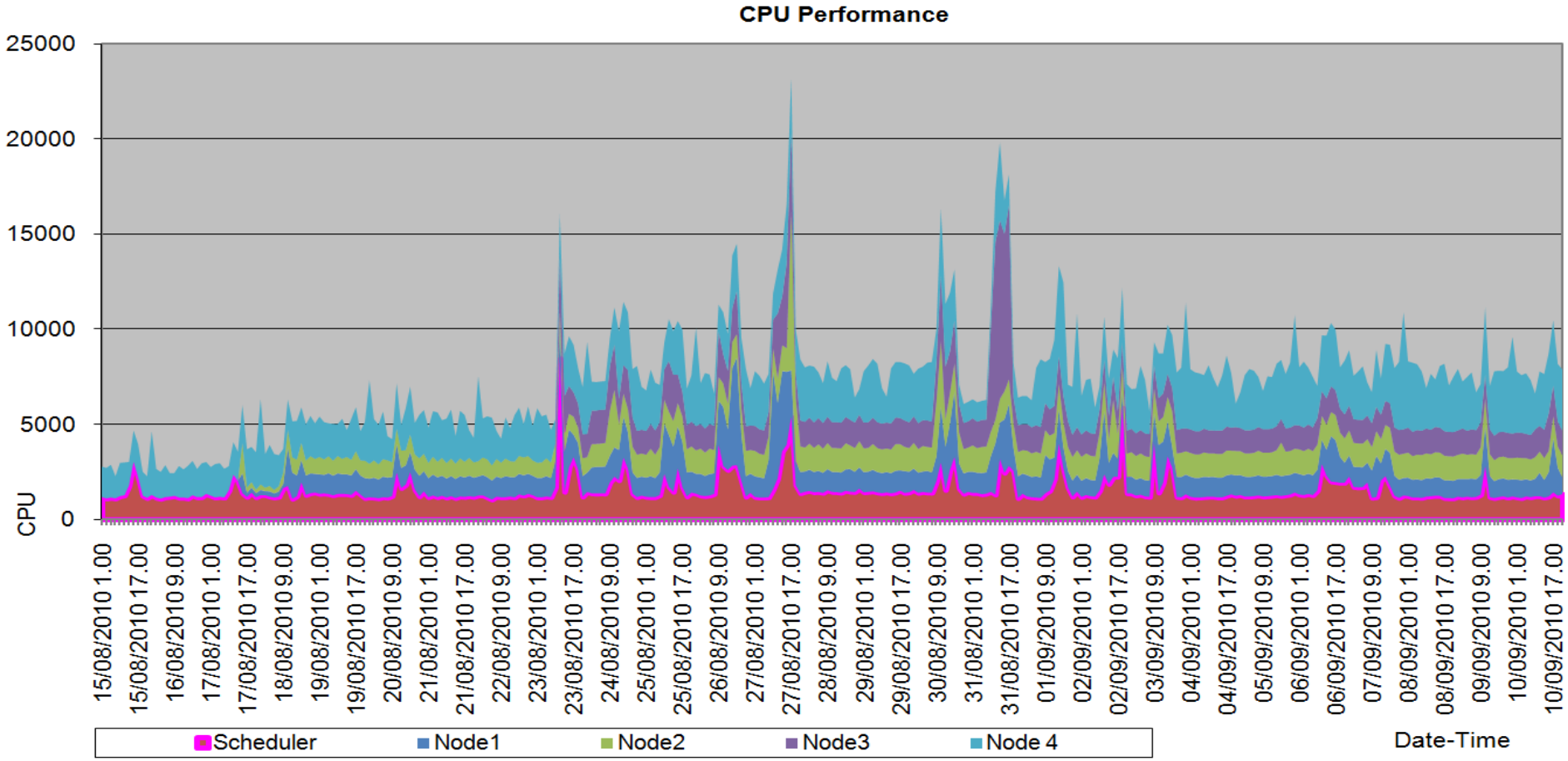
Clusters, generic definition

- **For Cluster, it is typically intended:**
  - Several kinds of computers and/or VMs used to compose a group of resources, that can be managed together, and may be sharing the same domain or not.
  - a set of computers/VMs into a datacenter which are dedicated to a unique problem, for example:
    - A microgrid, an Hadoop cluster, ……
    - A social network…….
    - A web portal with its multi-tier servers: front-end portals, balancer, database computers, backoffice microgrid nodes, etc.



From Clustering VM to Cloud

- The first step has been the exploitation of unused resources to provide them as a service to third party.

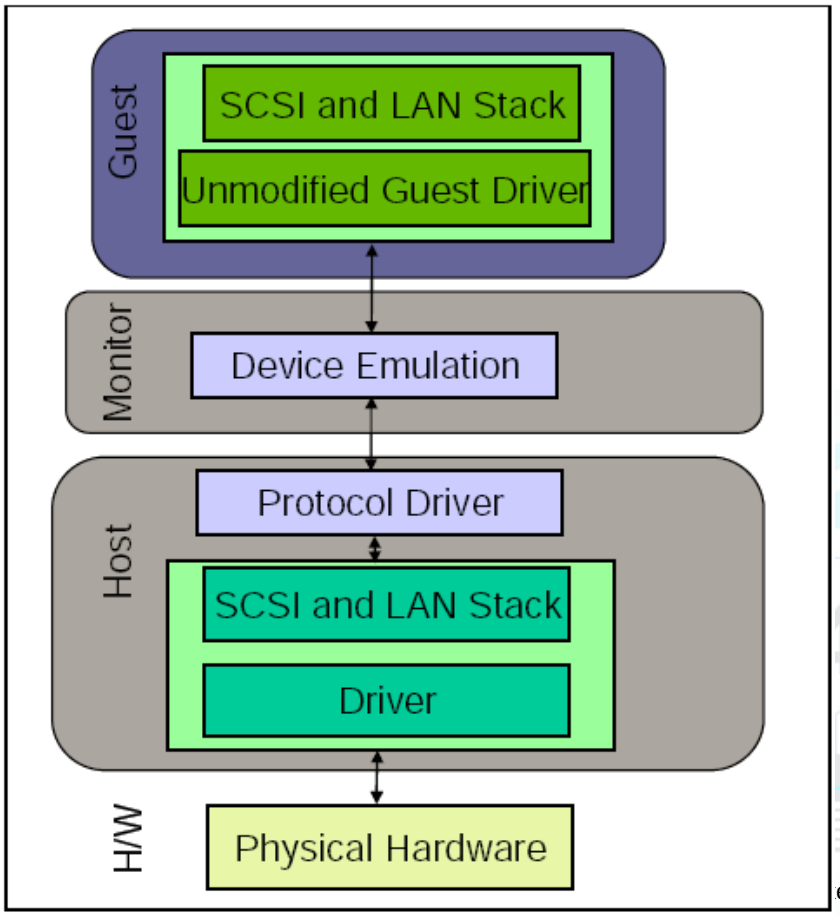


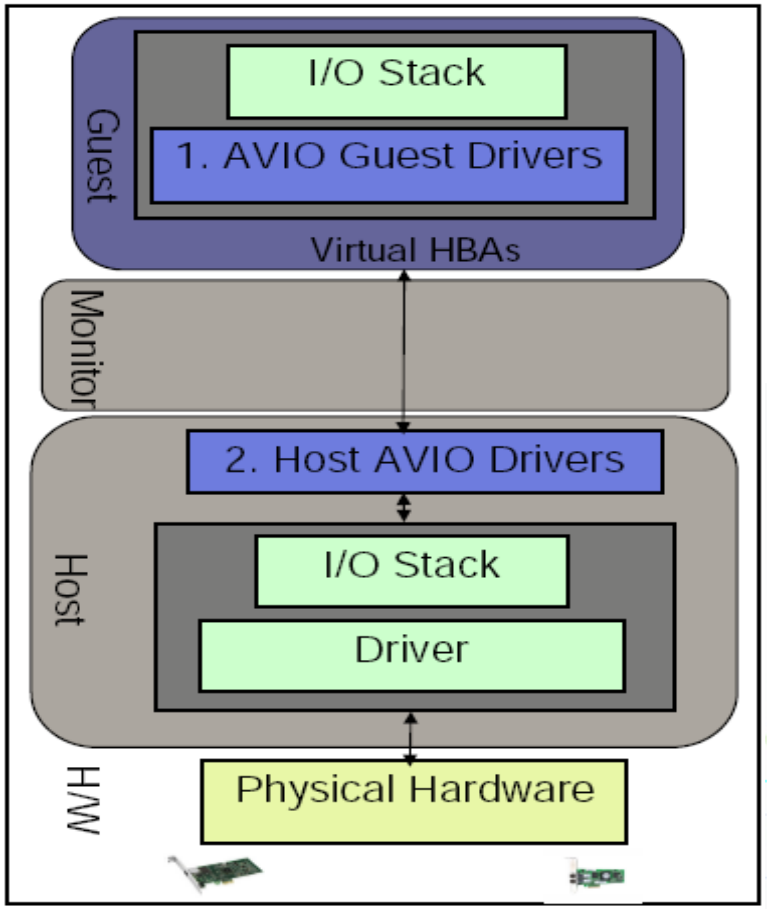
Approaches for Virtualization

- **Hardware Emulation**
  - Full emulation of Hardware devices and features
  - It is possible to use an original OS without changes, may be with some drivers installed, but **not kernel** changes
  - Higher isolation among VMs, strong robustness, limited efficiency.
  - **Used by** Vmware core
  - Typically 10% of overhead
- **Paravirtualization (total or only on some devices)**
  - The execution is performed via specific API and the hosted OS has to be modified to use them instead of the original HW
  - Lower isolation, higher efficiency
  - Lower robustness: VM crashes may crash the whole system
  - **Used by**: HP-VM, Xen (both of them which can also go in emulation mode)
  - **Used by VMware**: VMXNET (100 Gbps net), PVSCSI in vSphere4
  - Typically 2% of overhead

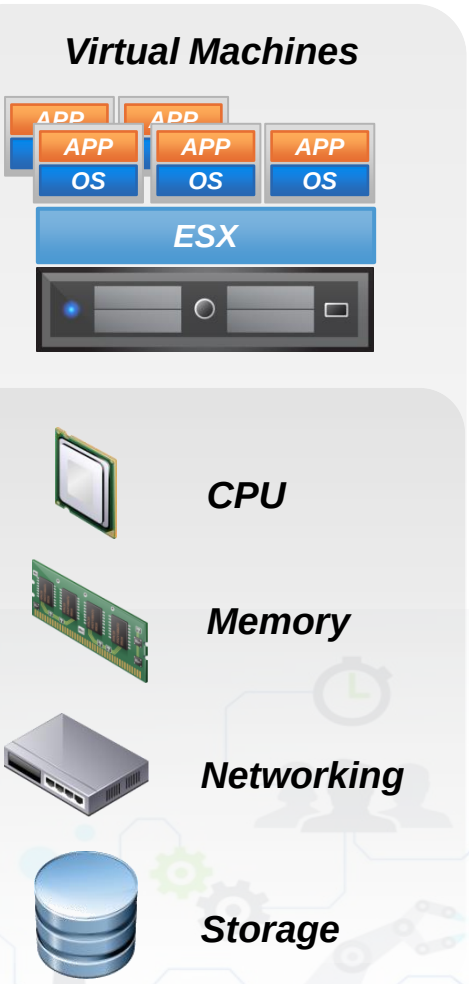
Para-virtualization

- It is intended as a solution to cut out a part of the I/O stack, for example for the HD access or network.
- The following example is related to HP VM v. 3.5





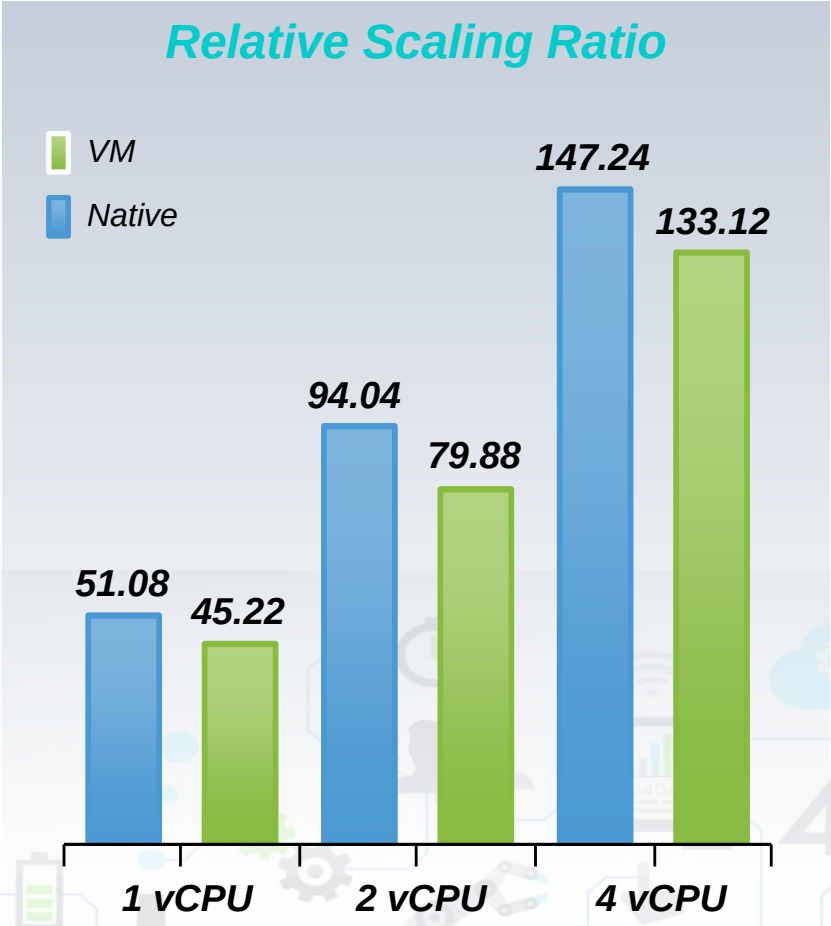
"Speeds and Feeds"
Optimization for the Highest Consolidation Ratios



- **VM Scale Up**
  - *8-way vSMP and 255 GB of RAM per VM*
- **Hardware Scale Up**
  - *64 cores and 512GB of physical RAM*
- **Hardware Assist**
- **Purpose Built Scheduler**
  - *Lowest CPU overhead*
- **Hardware Assist**
- **Page Sharing**
- **Ballooning**
  - *Maximum memory efficiency*
- **VMXNET3**
- **VMDirectPath I/O**
  - *Wirespeed network access*
- **Storage stack optimization**
- **VMDirectPath I/O**
  - *Greater than 360k iops per second*
  - *Lower than 20 microsecond latency*

Current **NEW**

ESX 4.0 Performance with SQL Server 2008

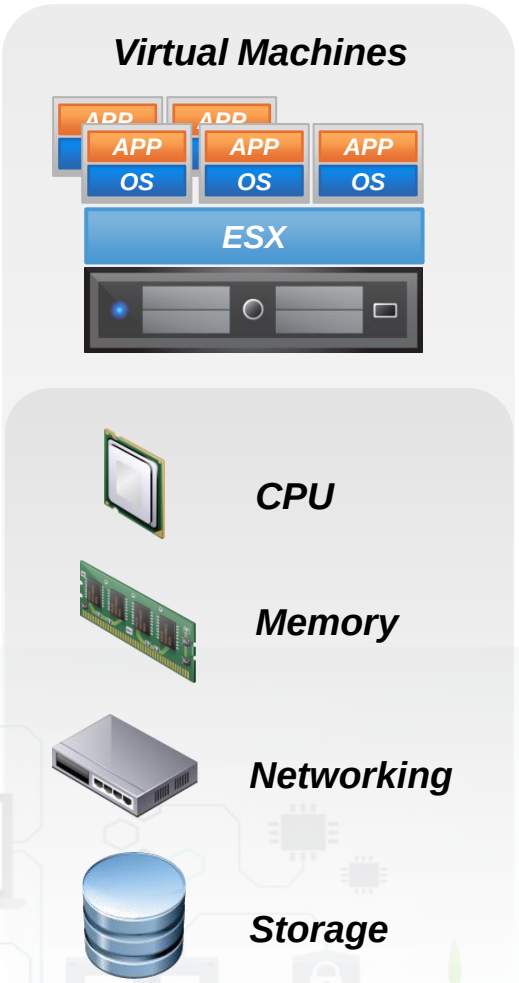


- ESX achieves 90% of native performance on 4.0 vCPU VM
- Workload transaction latency unchanged between ESX 4.0 and Native

Host: risorse primarie

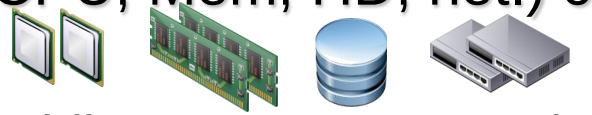
Host Profile

- Memory Reservation
- Storage
- Networking
- Date and Time
- Firewall
- Security
- Services
- Users and User Groups
- Security



Limiting VM Resources

- **VM Resources** (CPU, Mem, HD, net.) consists also in providing support for:

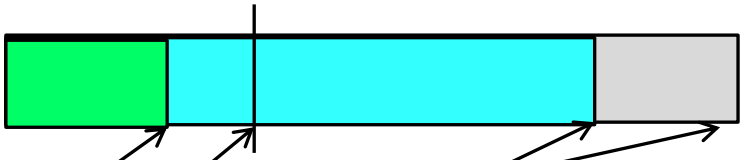


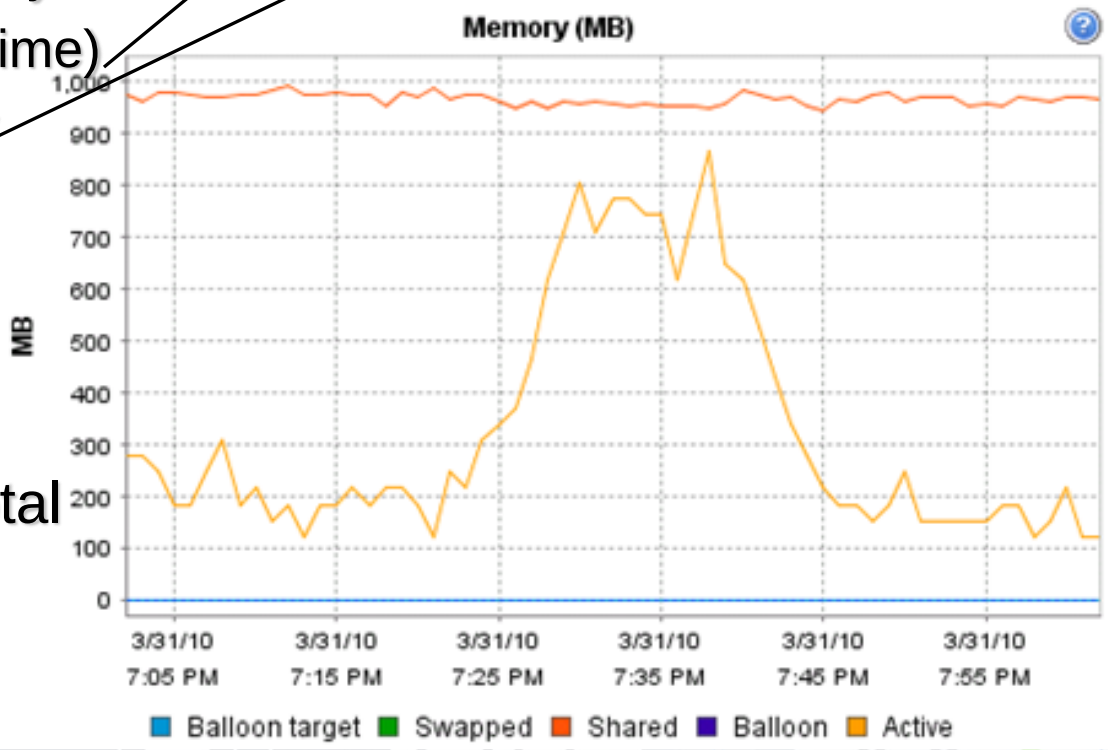
  - Dynamically providing resources over the reserved values that can be negotiation into the SLA/ contract.

- **Controlling and limiting** access and the exploitation of HW resources:
  - A limit on the number of CPUs
    - A limit on the number of Clocks, over of a reserved number of clocks
  - A limit on the maximum size of the RAM, over of the reserved number of Mbytes
  - A limit on the size of the HD, SAN/NAS access
  - A limit on the number of network cards, number of Mbps, etc.

Model for Memory

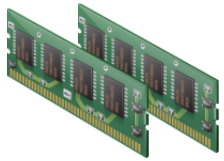
- Memory
  - Total host memory
  - For each VM
    - Max memory for the OS
    - Reserved Memory
    - Used Memory (time)
    - Limited memory





- Be carefull to overcommitted resources on the Host respect to the total CPU and Mem of the hosted VMs

VM: Memory Definitions

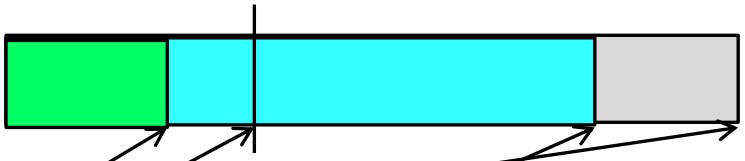


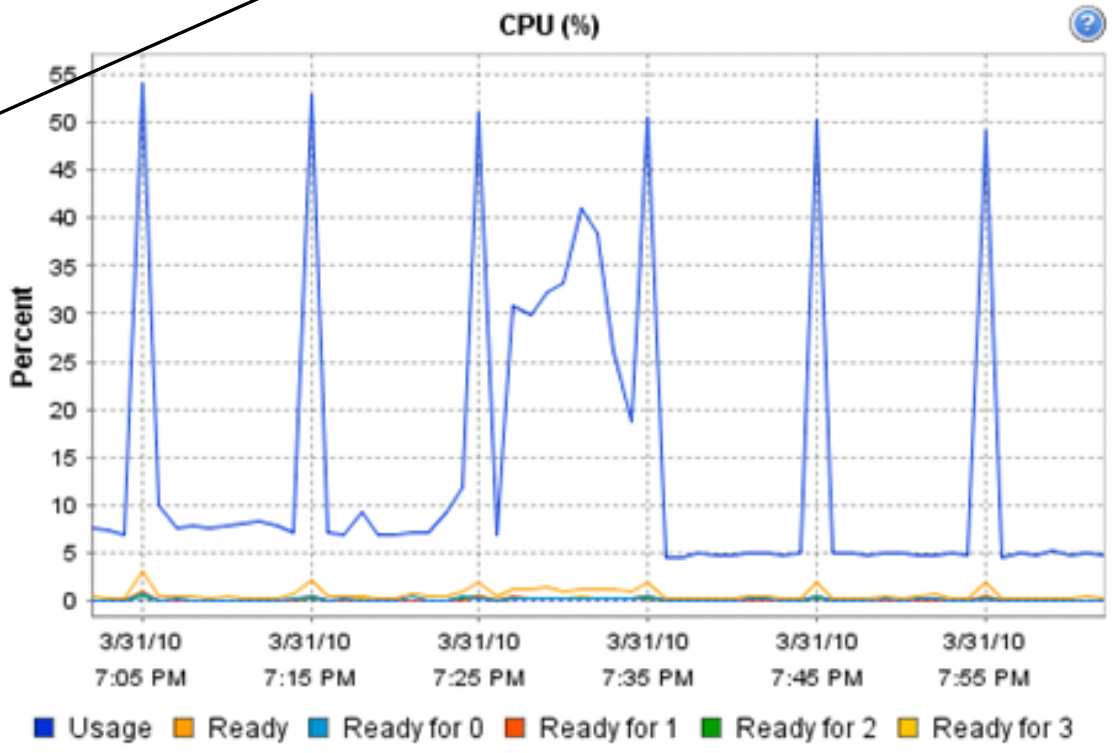
- **Reserved: not available for the other VM on the same host!!**

- **Active**: memory actually used
- **Ballon**: Memory requested by the VM to the host
  - This memory is shared among several hosts
- **Shared**: memory available for sharing
- **Swapped**: amount of memory that has been swapped into the HD by the VMKernel of the host
  - This parameter has to be low as much as possible
  - Increse the reserved to make it low or reduce the number of VM on the host

Model for CPU

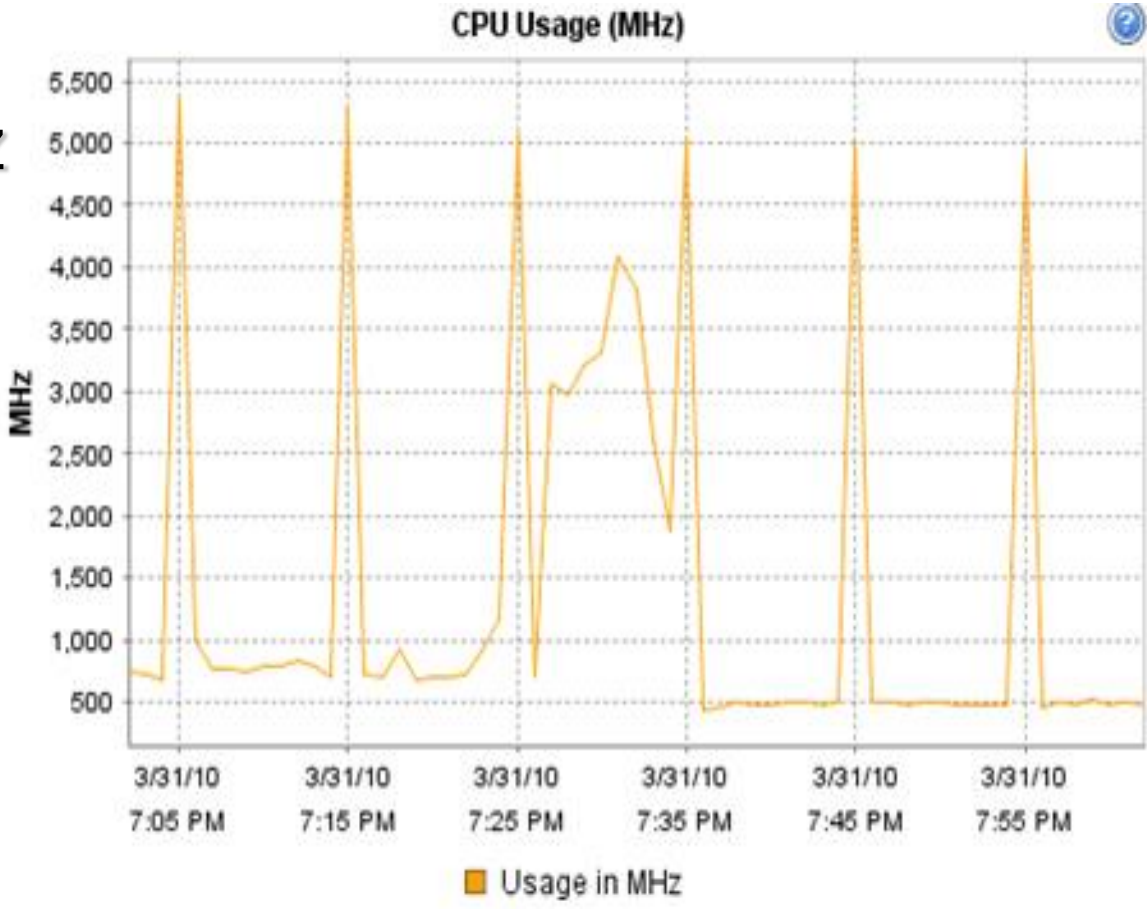
- Memory
  - Total host Clock
  - For each VM
    - Max #cks/s host
    - Reserved #cks/s
    - Used #cks/s
    - Limited #cks/s
- Ck/s → Hz
- 1 GHz → 1000 MHz





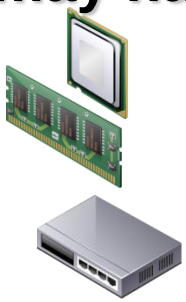
Model for CPU

- An host may have multiple CPU with multiple Core
- An host with 4 CPU at 3Ghz and 6 core each has
  - 24 cores at 3Ghz
  - 72.000 MHz
- VMs share this amount



Virtual Resources, 1/2

- The idea of Virtual Resources (CPU, Mem, HD, net.) consists in providing a number of resources larger than those that physically available and manage them virtually and/or dynamically
- **For example, one Host may have 2 VM and HW resources:**
  - **2 cores at 1300 Mhz**
  - **1.8 Gbyte RAM**
  - **2 network cards**
  - Best case:
    - CPU=400+800 Mhz
    - RAM=400+400 Mbyte
    - 2 Network cards shared
  - Worst case → no resources enough:
    - CPU=3 cores at 1300 Mhz
    - RAM=2 Gbyte
    - 2 Network cards shared



VM1:

- *1 CPU 1300 Mhz, 400 Mhz reserved*
- *1 Gbyte RAM max, 400 Mbyte reserved*
- *2 network cards, 2 IPs*

VM2:

- *2 CPU 1300 Mhz, 800 Mhz reserved*
- *1 Gbyte RAM max, 400 Mbyte reserved*
- *2 network cards, 2 IPs*

Virtual Resources, 2/2

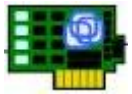
- The server hosting the VMs
  - Min Mhz:
    - 400Mhz + 800Mhz
  - Max Mhz:
    - 1300Mhz + 2*1300Mhz
  - Min RAM:
    - 400Mbyte + 400Mbye
  - Max RAM:
    - 1000Mbyte + 1000Mbye
  - Network:
    - No limits on the number of virtual IP addresses/cards

VM1:

- *1 CPU 1300 Mhz, 400 Mhz reserved*
- *1 Gbyte RAM max, 400 Mbyte reserved*
- *2 network cards, 2 IPs*

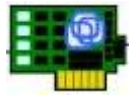
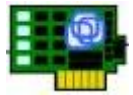
VM2:

- *2 CPU 1300 Mhz, 800 Mhz reserved*
- *1 Gbyte RAM max, 400 Mbyte reserved*
- *2 network cards, 2 IPs*

Model for network

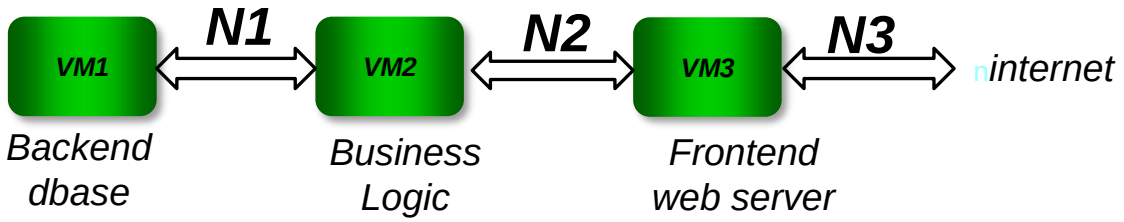
- An host may have multiple network cards with one physical IP each or some of them bridged
- Each single VM may have:
  - access to a number of networks (physical or virtual)
  - Installed multiple network cards
    - ➔Each card can have one IP and may be connected to a network

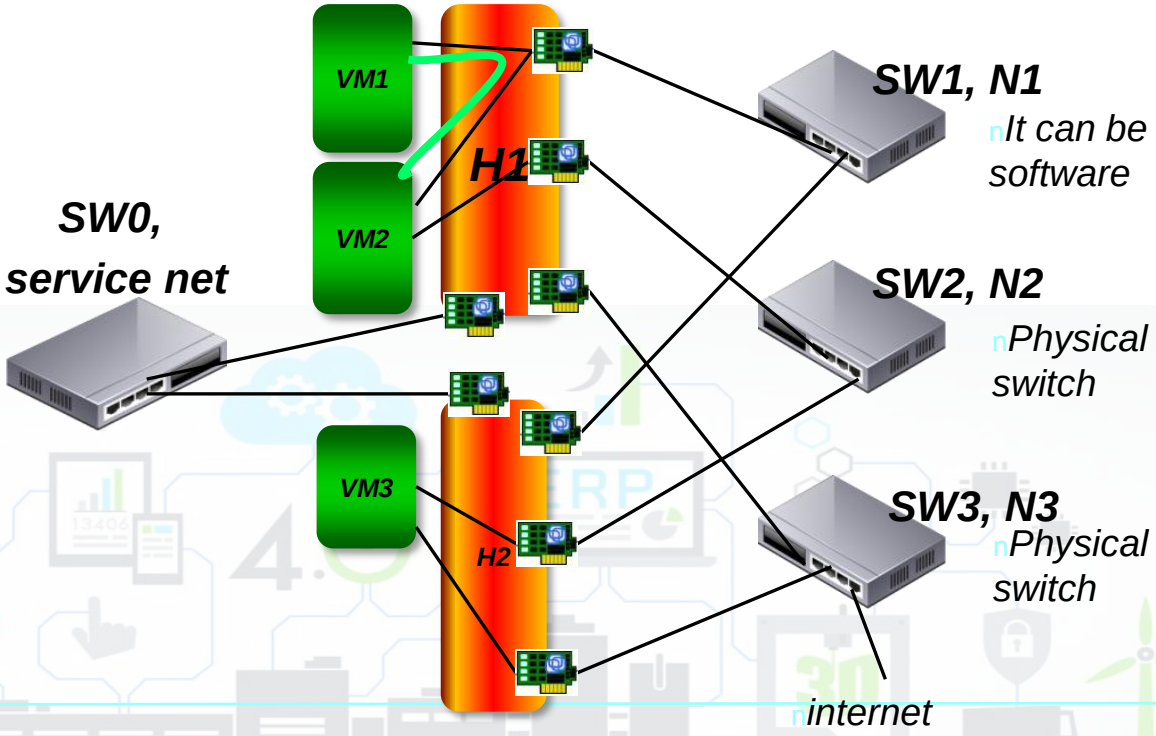
- This means that an host may be connected to multiple networks with multiple IP
- Network cards may be connected to:
  - Physical switch
  - Virtual switch

Example of Cloud Networking

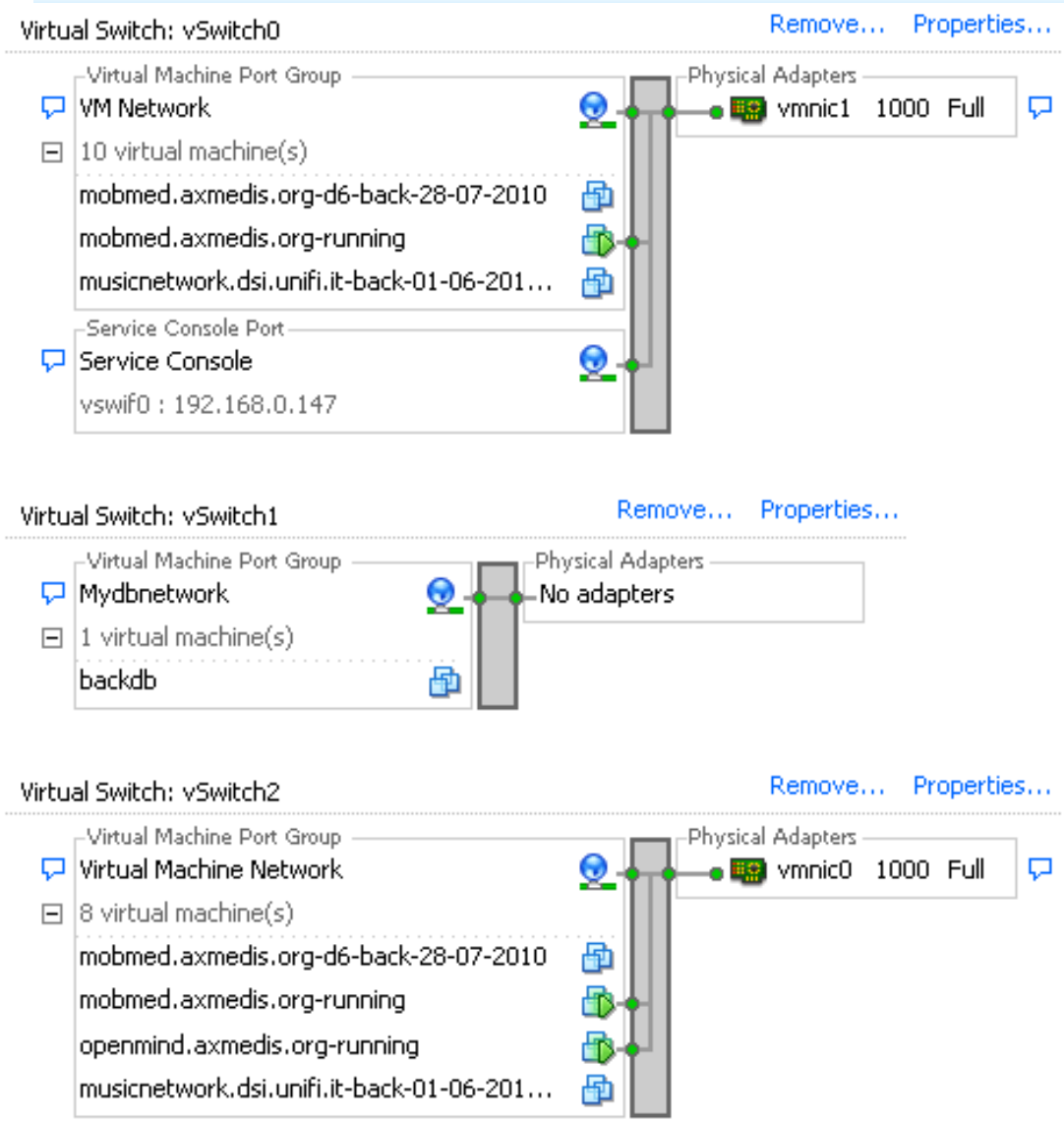
- Example of allocation of a three tiers solution
- 3 different networks (N1, N2, N3)
- 2 Host: H1, H2
- 3 VM: VM1, VM2, VM3



- → SW1 is convenient to be software (full virtual). If it is HW, the VM1 can be moved from H1 to H2, dynamically
- → SW2 could be fully virtual if VM3 is moved on H1



Network and Virtual Networks



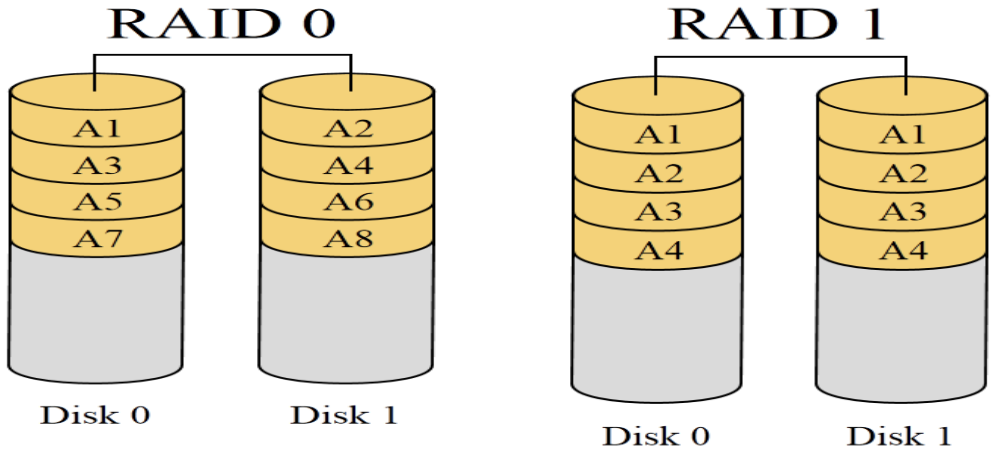
- The same VM with access to 2 different network via real network adapters
- A virtual network

Model for HD, Storage

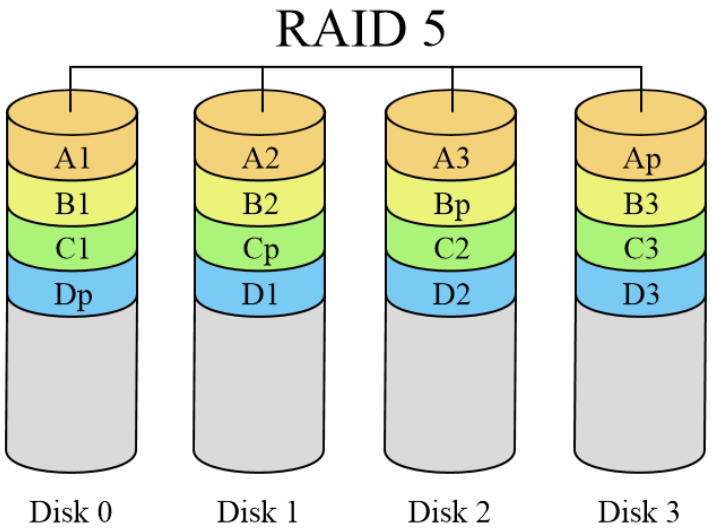
- Storage:
  - HD contained into VM
  - SAN: External to VM and mounted for their usage
  - In any case connected with some Operating Systems (may be dedicated)

- HD with fault tolerant solution:
  - Redundancy: replications into the Storage
  - Distributed/federate/replicated: replications into multiple storage

Internal replications: HD RAID

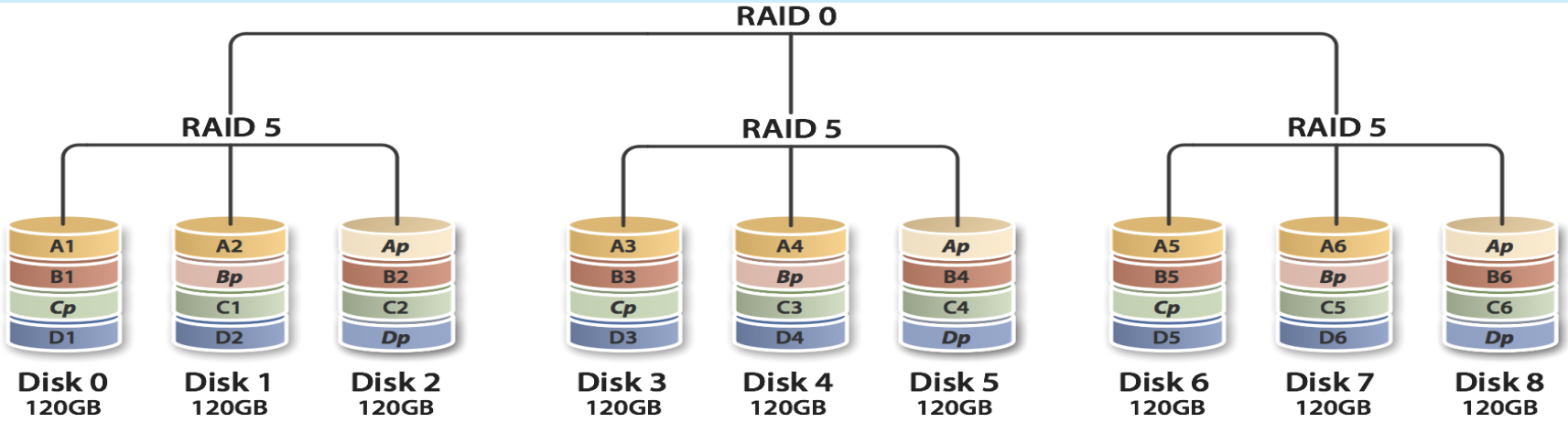


- *RAID 0: HD0+HD1*
- *RAID 1: HD0 cloned on HD1*



- *RAID 5: gestione del guasto di un disco*
  - *valido fino a 14*
  - *per la doppia parità*
    → RAID 6

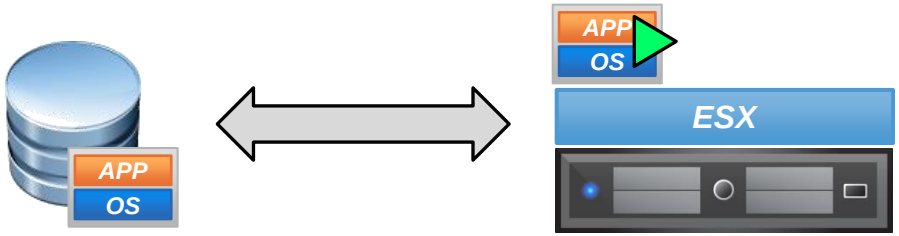
Internal replications: RAID 50



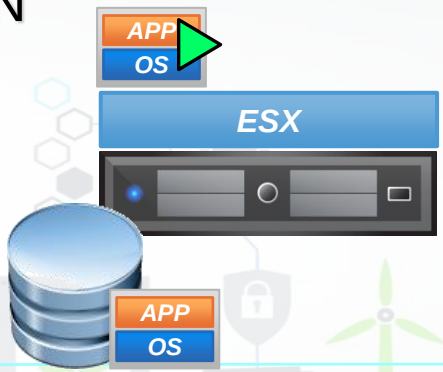
- RAID 50: A1,A2,A3,A4,A5,A6 and three Parities
  - Min 9HD, can support failures of three HDs
  - improves upon the performance of RAID 5
- In this case:
  - Size: 6 over 9 HD, you lose a HD per R5
  - Read/Write speed up: n(m − 1)XHD
    - → n=3, m=3 → 6x in this case

Simple configuration

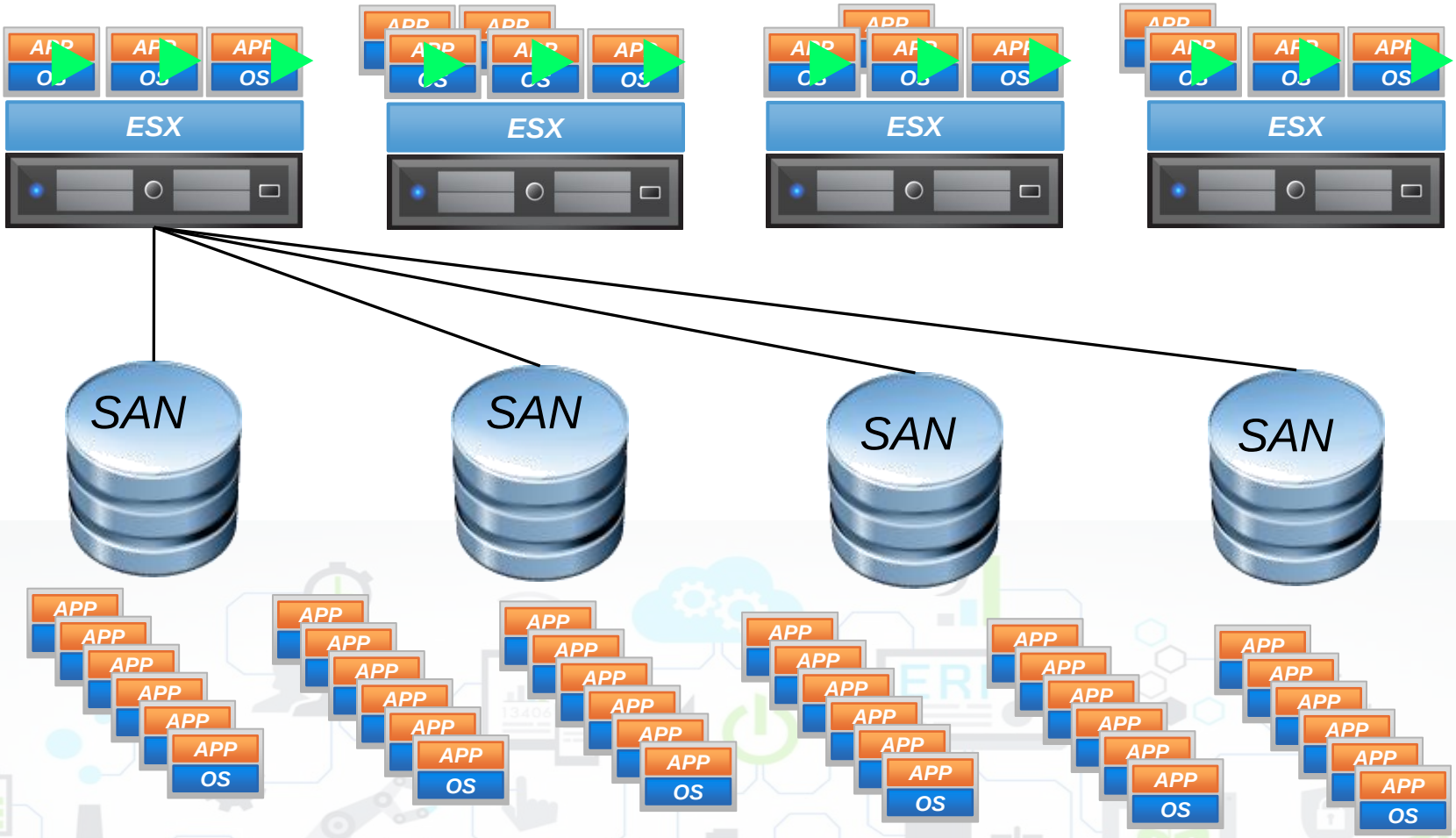
- VM stored on a storage and executed on a remote hypervisor host
  - At the start is uploaded on ESX memory and RUN
  - Each write is passed back to the storage, SAN
  - All via network



- VM stored and executed on the same hypervisor host
  - At the start is uploaded on ESX memory and RUN
  - Each write is passed back to the storage
  - All via internal communication
  - → higher performances
  - → lower flexibility

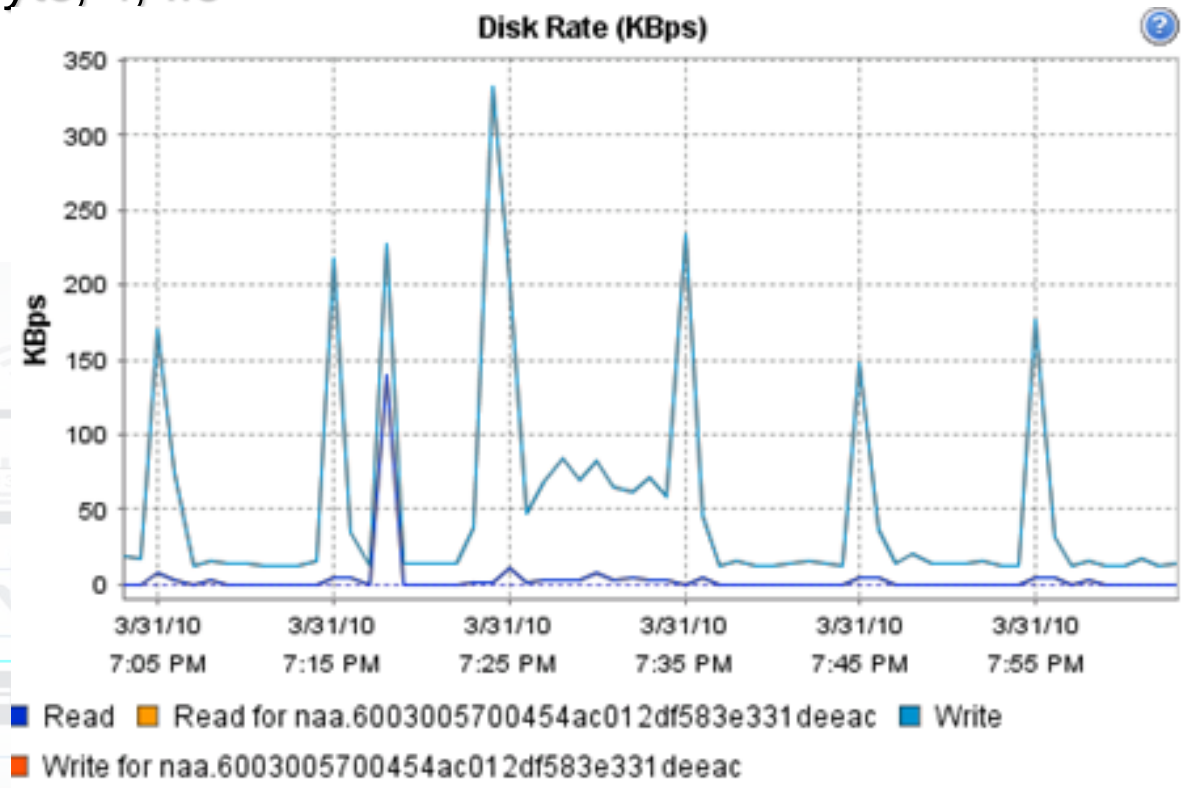


Cluster of Hosts in DRS



Model for storage consumption

- HDD Typical:
  - 100 Mbyte per sec. (high/max value)
  - Seek time 5-7 Msec.
  - 5000-7500-10000-15000 rpm (revolution per minute), spindle
  - Thus latency: (5.56, 4.17, 3.00, 2.00) respectively
  - Format Block sie: 2Mbyte, 4, ..8
- Cache, no cache; different kind: …..
- SSD:
  - 30% faster



Storage Consumption

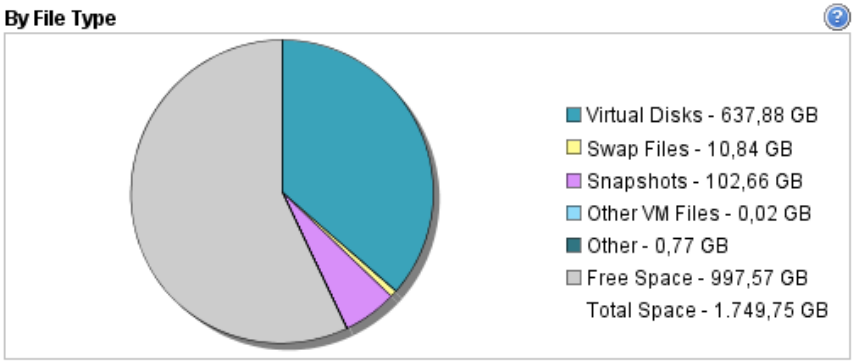
- Each host may have
  - internal Storage one or more HD
  - External storage may be mounted (SAN or NAS)
- External storage may be shared or reserved for a specific Host

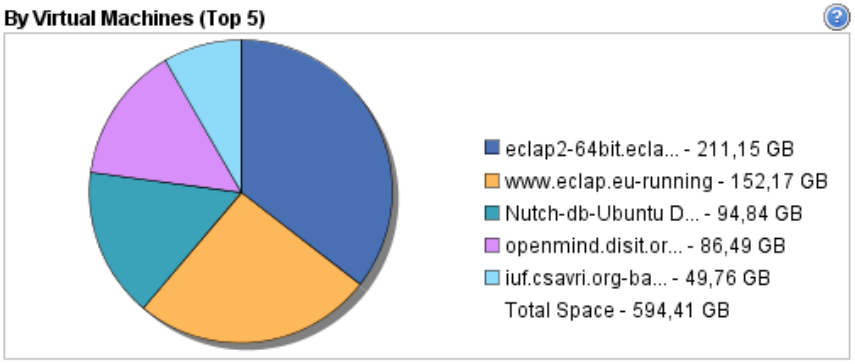
- Each VM has its permanent memory consumption:
  - Image for the VM
  - + space for snapshots (HD incremental changes + memory when the snapshot is performed on running)

Examples

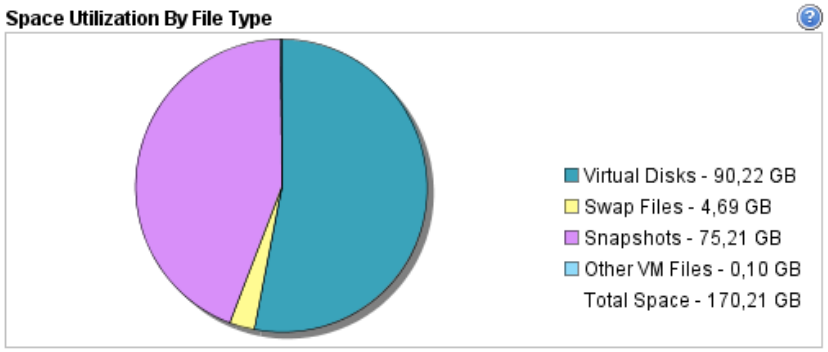
Host

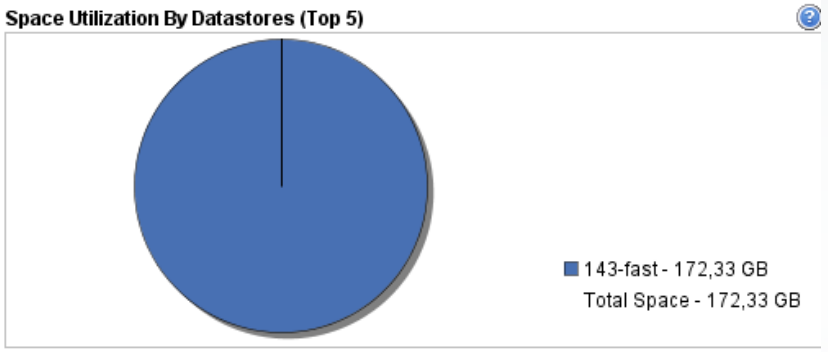
Space Utilization for 145-Storage-A-4





Virtual Machine





Virtual Machine: Snapshots

- **Working on VM Snapshots**
  - Creating a Snapshot:
    - A point from which it is possible to reboot, restart
    - Consuming HD space
    - Making back up since the core image of the VM is not changed, changes are confined in the files representing the last status “you are here” and not the previous conditions
  - Restarting from a past snapshot
    - Losing current point: “you are here”, to avoid this do another snapshot!!
  - Deleting a past snapshot
    - Recovering HD space, removing a past restarting point
  - Removing all snapshots
- **Defragmenting images of the HDs** into the VM

VM Snapshots

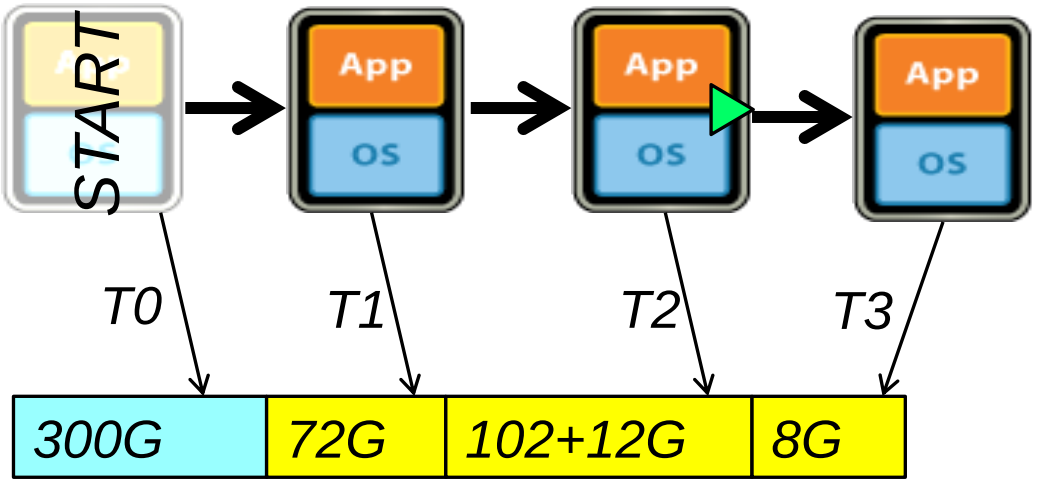
- **VM Snapshots can be at VM Off or ON**
  - Snapshots of running VM have implications…
- **Removing Snapshots**
  - **Defragmenting images of the HDs** into the VM
  - Consolidating the changes in the previous version

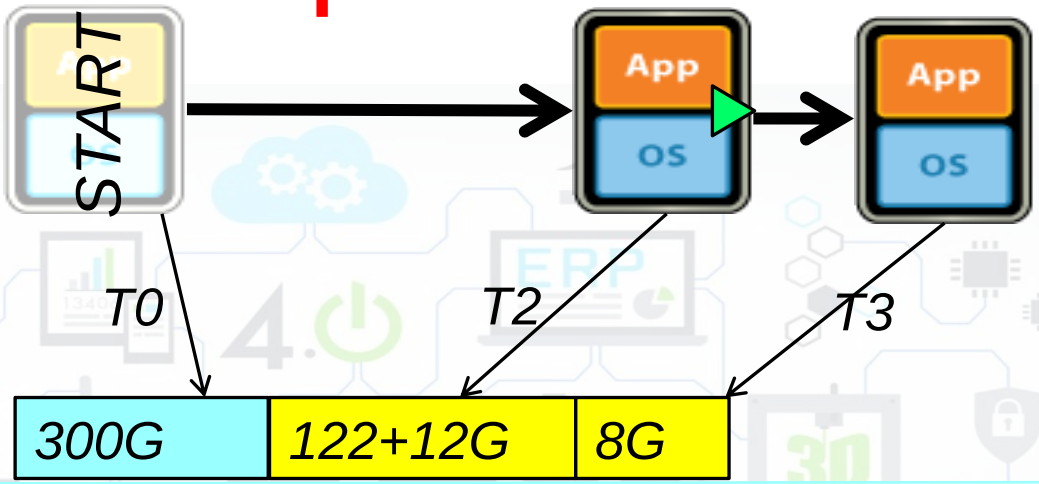
- VMware WS has an automated Snapshot model to plan the periodic snapshotting of the VM, for example:
  - every hour, day, week, …

- A way to make back up
  - A different way can be to clone the VM on different host or NAS. In most cases, the cloning implies the lost of performed snapshots

Snapshot process

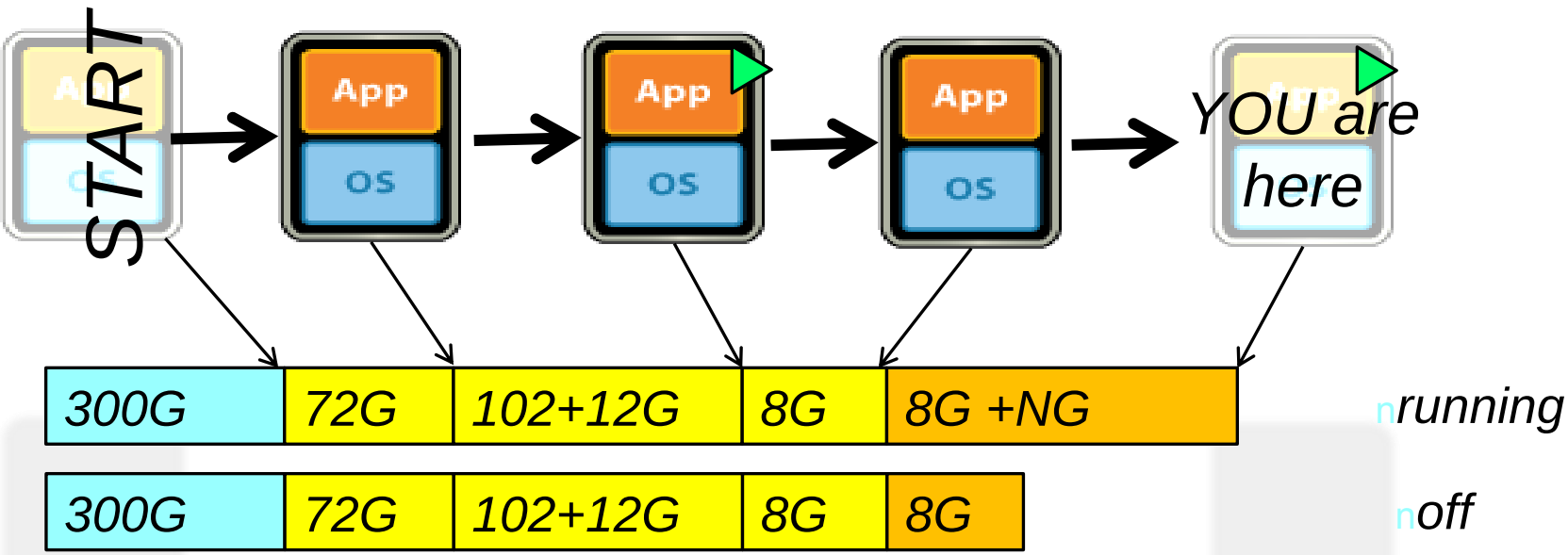
- Original version
  - 1 VM
  - 12 Gb ram
  - 300Gbyte hd
  - 72+102+12+8 Gbyte snapshots





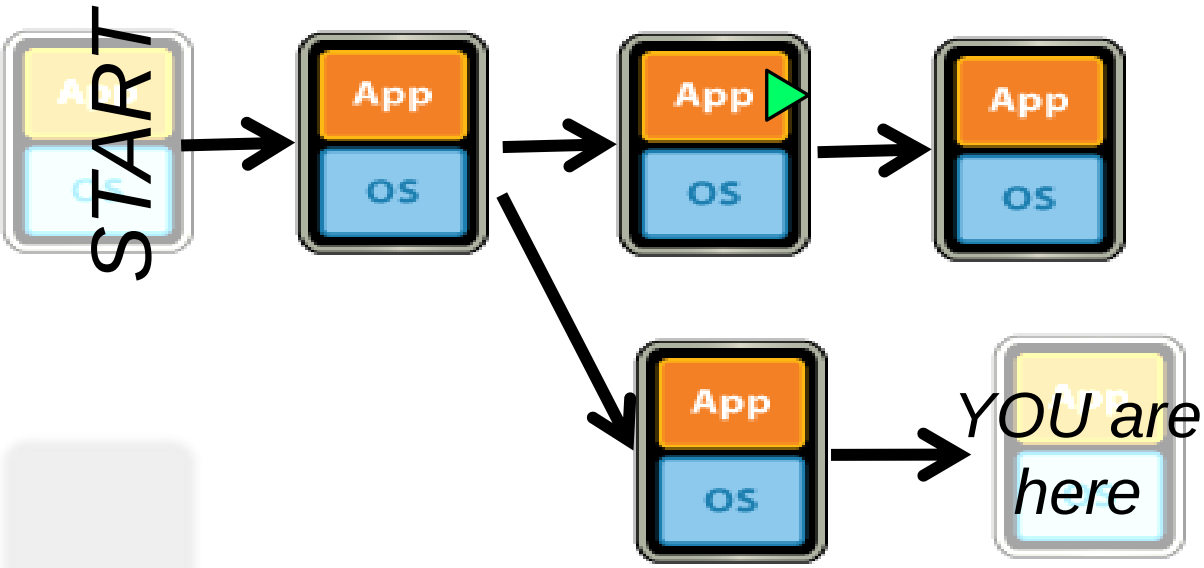
- After T1 Rimoval
  - 1 VM
  - 12 Gb ram
  - 300Gbyte hd
  - 72+122+12+8 Gbyte snapshots

Virtual Machine: Snapshots, 1/3

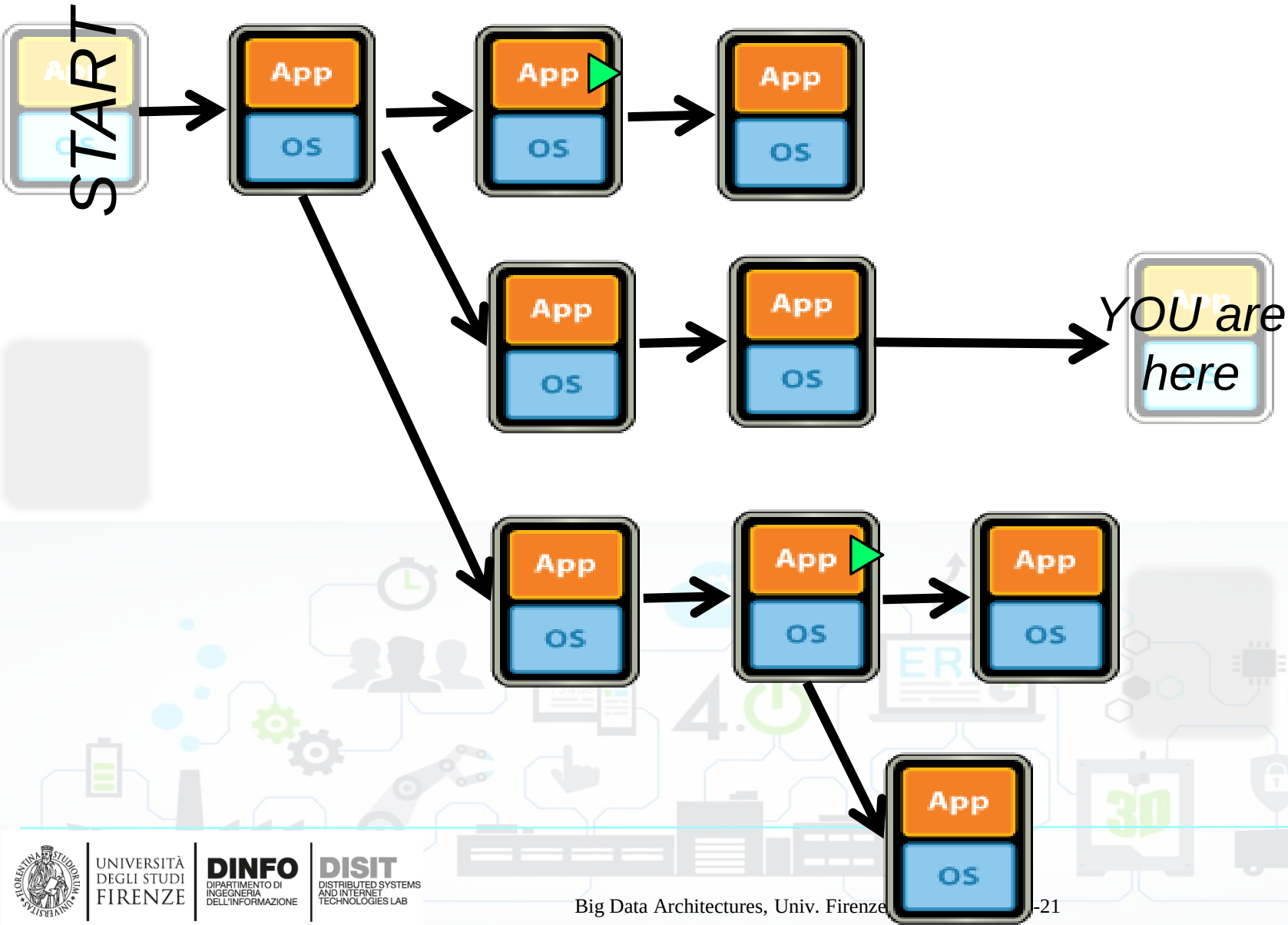


- When a number of snapshots are in place:
  - the access to HD is deteriorating since the data are not contiguous, the data access implies multiple seek for multiple accesses at different segments of the same data in different file representing the different progressive snapshots!

Virtual Machine: Snapshots, 2/3



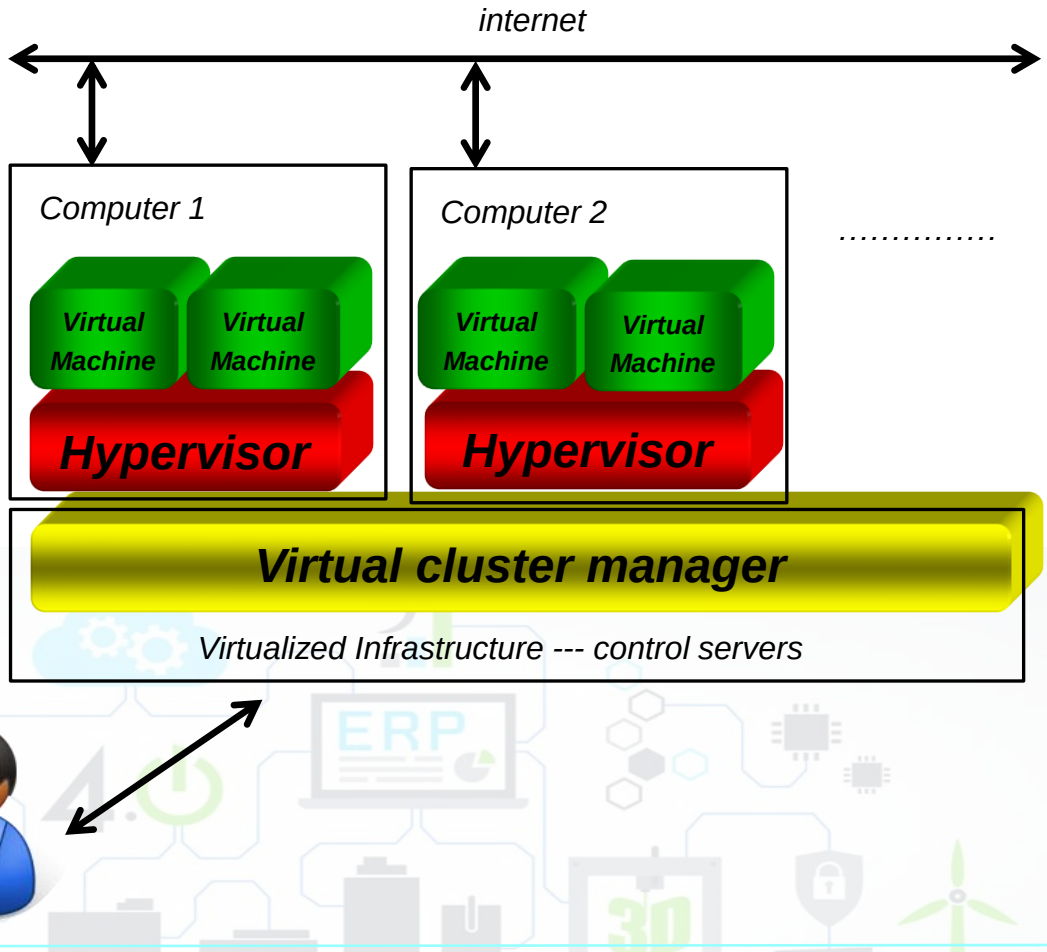
Virtual Machine: Snapshots, 3/3



Structure

- Elements of cloud Computing
- Virtualization
- Cloud Computing
  - cloud vs grid
  - goals of cloud computing
  - Solutions as a Service
- High Availability
- vSphere Infrastructure
- Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions, SaaS Solutions, PaaS Solutions
- Progetto ICARO

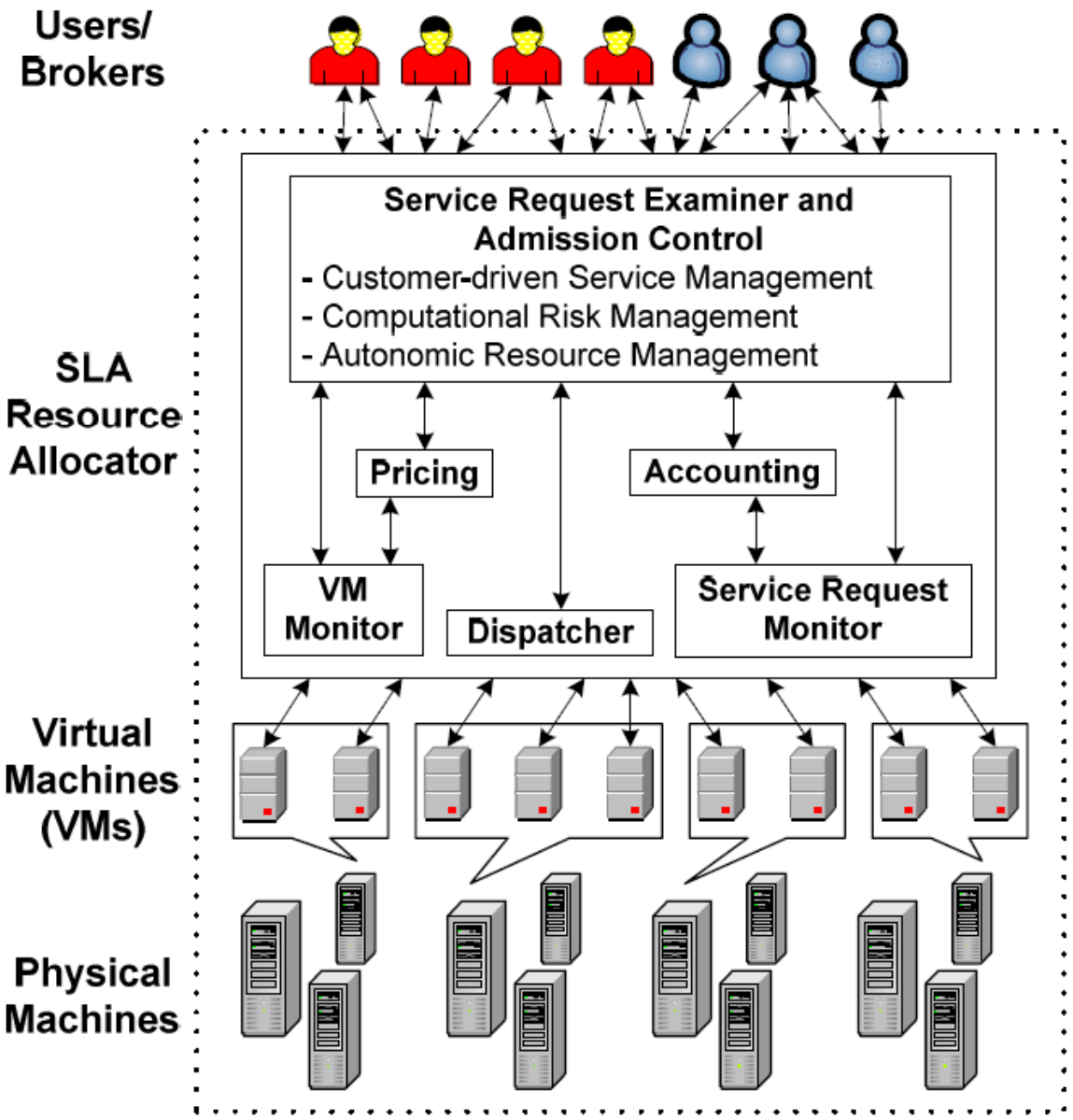
Cloud Computing with VMs
Several Hypervisors on a Clusters



Cloud Computing

- **A Cloud:** a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers, VMs
  - They are dynamically provisioned and presented as one or more unified computing resources
  - based on service-level agreements, SLA, established through negotiation between the service provider and consumers
- Subset of grid computing where the allocated process are virtual computers
- **Implies**
  - An alternative way to have local servers or GRIDs
  - Outsourcing: HW and SW tools, they may be grid elements
  - Outsourcing: network, CPU, memory, HD, etc.
  - Definition of some service agreement, monthly rate, minimum network capability, kind of HW, mem space, minimum level of CPU/mem, etc.

Abstraction of Cloud Computing, (Buyya & Yeo)



High-level market-oriented cloud architecture.

Cloud vs GRID computing

- **Cloud computing** is an evolution of GRID computing
- **Renting a GRID** service means:
  - Parallelize the algorithm, demand the execution, wait for the results, etc.
  - To do no know where the processes are executed. In most cases batch processing such as on Globus
- **Renting a VM/Computer** into a Cloud means:
  - Simple contracts, remote access to servers
    - ➔Get access to virtual or physical resources at 100%
  - Privacy of the allocated processes in your VM/Host
  - Processes running on your preferred OS and do not have to be recompiled as in most Grid solutions.
  - Scalability in terms of number of CPU, Mem, network performance, computers, etc. OR you have to change your OWN architecture
    - ➔Simple creation and reconfiguration of multitier solutions
      - Creating fault tolerant solutions
      - Balancing load of CPU, Memory, network, etc.

Cloud vs SuperComputers

- **Cloud Computing** is based on a set of Hosts for hosting VM or providing access to single/multiple Servers fully at disposal of the customers accessing to the cloud via remote access (see later).
  - They are MIMD computers
  - Hosts may contains hypervisors to host VM of diff. OSs.
  - Hosts may be single computers with Linux, Windows,…
- **Super Computers**, such as Blue Gene/P:
  - Processor PowerPC 450 with 4 cores, 850 Mhz
  - Each board: 32 CPU processors
  - Each Rack: 32 boards
  - Jülich Research Centre in Germania has a Blue Gene with 65536 processors → 167 teraFlops, the strongest computer in the world!

Comparing Computing Services

System / Property	Amazon Elastic Compute Cloud (EC2)	Google App Engine	Microsoft Live Mesh	Sun Network.com (Sun Grid)	GRIDS Lab Aneka
Focus	Infrastructure	Platform	Infrastructure	Infrastructure	Software Platform for enterprise Clouds
Service Type	Compute, Storage (Amazon S3)	Web application	Storage	Compute	Compute
Virtualisation	OS Level running on a Xen hypervisor	Application container	OS level	Job management system (Sun Grid Engine)	Resource Manager and Scheduler
Dynamic Negotiation of QoS Parameters	None	None	None	None	SLA-based Resource Reservation on Aneka side.
User Access Interface	Amazon EC2 Command-line Tools	Web-based Administration Console	Web-based Live Desktop and any devices with Live Mesh installed	Job submission scripts, Sun Grid Web portal	Workbench, Web-based portal
Web APIs	Yes	Yes	Unknown	Yes	Yes
Value-added Service Providers	Yes	No	No	Yes	No
Programming Framework	Customizable Linux-based Amazon Machine Image (AMI)	Python	Not applicable	Solaris OS, Java, C, C++, FORTRAN	APIs supporting different programming models in C# and other .Net supported languages

Comparing Computing Services

- In the previous table, different solution for computing services in the network are compared.
- They are mainly: cloud computing, grid services, application services,
- The solutions taken have been selected as representative of their category
  - Please see the slides on GRID for more details and a wider comparison on the grid solutions
  - Please see the slides regarding the general distributed systems for multitier applications

Advantages of Cloud Coumputing

- **Grids are** difficult to use and to maintain:
  - GRID customers have too many different needs that make the creation of fully open grid very difficult.
- **Cloud computing**
  - HW/SW is hosting virtual computers that can be moved to other solutions with low costs
  - Lower costs since the HW/SW is seen as a service
    - No maintenance, centralized services such as back up, scaling, etc.
  - Lower costs for Small Business:
    - Reduction of costs since the admortment can be performed by who is exploiting the computer
  - Scalability similar to grid:
    - Horizontal scaling
    - Parallelization as MIMD

Goals of Cloud Computing

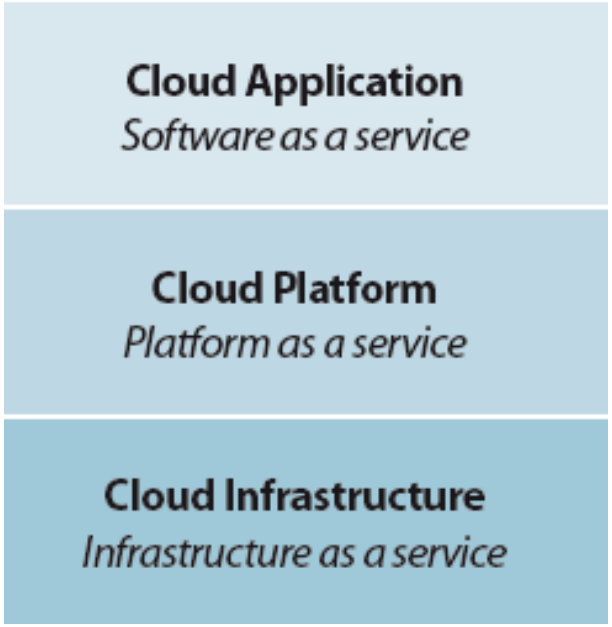
- ***Scalability***.
  - scaling with workload demands so that performance and compliance with service levels remain on target
- ***Availability***.
  - users of Internet applications expect them to be up and running every minute of every day, i.e.: h24, 24/7
- ***Reliability***
  - physical system components rarely fail, but it happen. So that, they can be replaced without disruption.
  - Today, reliability means that applications do not fail and most importantly they do not lose data, and the service is not stopped.
- ***Security***.
  - Applications need to provide access only to authorized, authenticated users, that need to be able to trust that their data is secure.

Goals of Cloud Computing

- ***Flexibility and agility:***
  - Adapt rapidly to changes of business conditions by increasing the velocity at which applications are delivered into customer hands. E.g.: more CPU, more clock, more memory, more network cards, etc.
- ***Serviceability:***
  - In the past this meant using servers that could be repaired without, or with minimal, downtime.
  - Today it means that an application's underlying infrastructure components can be updated or even replaced without disrupting its characteristics including availability and security.
- ***Efficiency:***
  - differentiates the cloud computing. The process allocation and costs have to be very effective with respect to the investment.

Definizioni

- **Classificazione NIST**
  - **Software as a Service (SaaS)**
  - **Platform as a Service (PaaS)**
  - **Infrastructure as a Service (IaaS)**



- **Business Process as a Service (BPaaS)**
  - **Aggiunto in seguito**

- **Everything as a Service (XaaS)**
  - Middleware as a Service !!!!

Infrastructure as a Service (IaaS)

- erogazione di servizi infrastrutturali relativi a capacità elaborativa, storage, rete e altri elementi di base assolutamente indipendenti da servizi applicativi di qualunque tipo.
- Si utilizza quindi l'infrastruttura messa a disposizione dal provider per eseguire la propria applicazione,
  - pagamento in base al consumo dell'infrastruttura
  - lasciando sotto la responsabilità dell'utente la gestione del sistema operativo, dell'eventuale middleware e della parte di runtime, oltre che dell'applicazione stessa.
- Amazon EC2 è un esempio di servizio IaaS.

Platform as a Service (PaaS)

- erogazione di servizi applicativi di base come sistemi operativi, middleware, linguaggi, tecnologie di base dati e l'ambiente runtime necessari per eseguire l'applicazione,
- L'applicazione rimane l'unica cosa sotto la responsabilità dell'utente, oltre alla definizione del modello (e.g., numero e dimensione dei server, datacenter, caratteristiche del networking) da utilizzare per l'esecuzione dell'applicazione.
- Google AppEngine è un esempio di Platform as a Service.
- A livello PaaS viene anche collocato l'insieme dei servizi MaaS, Middleware as a Service.

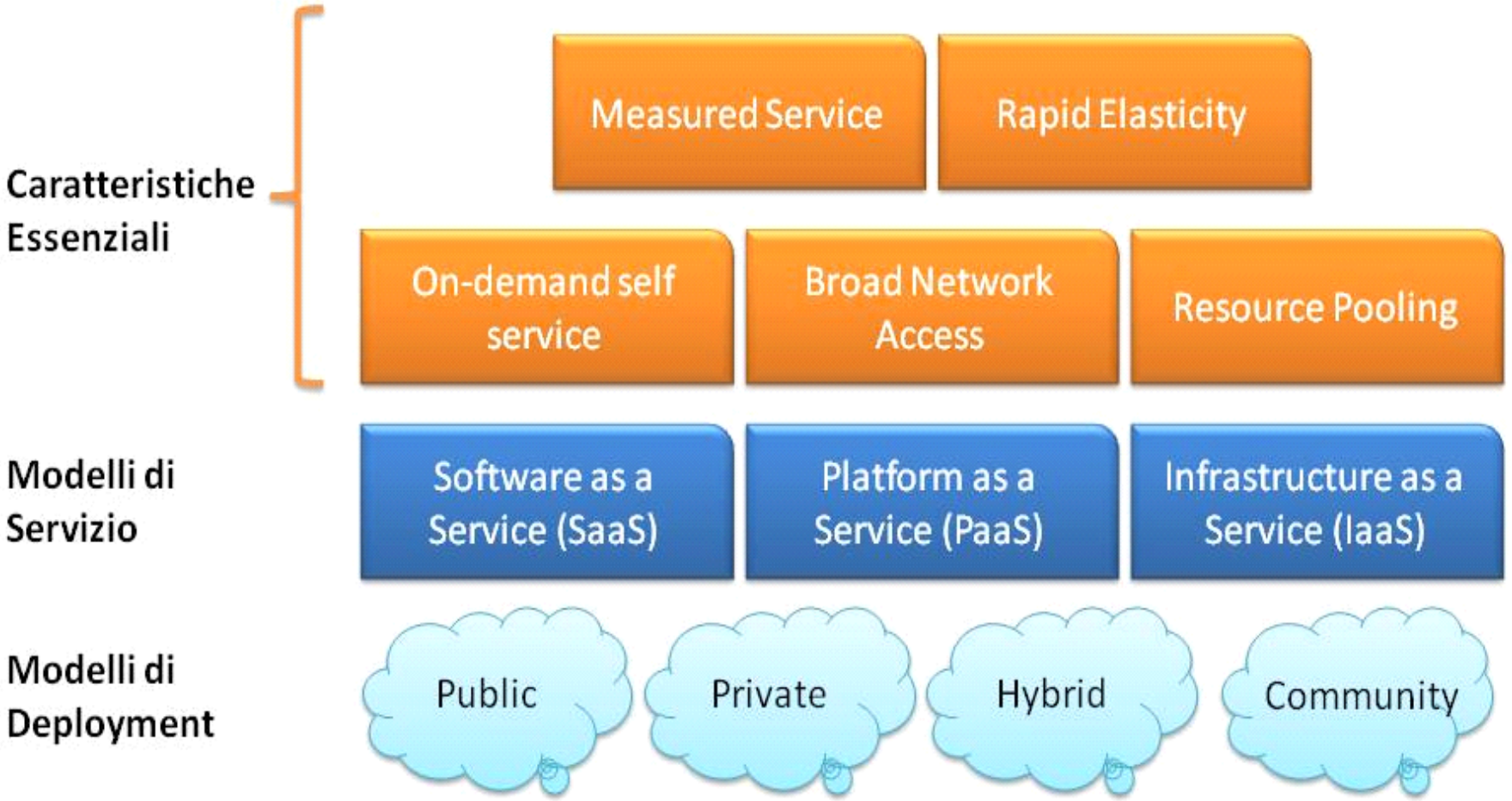
Software as a Service (SaaS)

- erogazione di servizi applicativi di qualunque tipo, accessibili indipendentemente dalla collocazione e dal tipo di device utilizzato.
- Non è eseguita un'applicazione proprietaria del cliente, ma il cliente stesso paga il diritto (mediante licenza o canone di affitto) di utilizzo di un'applicazione messa a disposizione dal provider, senza preoccuparsi di come essa venga realizzata e gestita nel cloud.
- L'unica preoccupazione del cliente in questo caso, oltre ovviamente alla scelta della corretta applicazione che soddisfi le sue necessità, è gestire il numero di licenze richieste in funzione del numero di utenti.
- SalesForce.com Customer Relationship Management (CRM) è un esempio di soluzione in cui il software è venduto in modalità as a service.

Business Process as a Service (BPaaS):

- erogazione di servizi non esclusivamente riferiti ad ambiti applicativi ma direttamente alle funzionalità di business o di processo, potenzialmente trasversali rispetto alle piattaforme applicative.
- un processo di business mappato interamente nel cloud (composto da servizi, applicazioni web, applicazioni legacy, servizi di integrazione, etc.).
- Il processo di business è un pattern di servizi ed include problemi di sicurezza, costi, scalabilità connessione fra local e cloud bidirezionale, il cloud può essere un burst per l'azienda, e può sgravare i costi nel momento del bisogno.

Modello Generale



Definizioni

- ***Private Cloud.*** abilitata per operare soltanto per un'organizzazione. può essere gestita dalla stessa organizzazione o da parte di terzi.
- ***Community Cloud.*** **co**ndivisa da più organizzazioni a supporto di una singola community che ha interessi e obbiettivi comuni. Questa può essere gestita dalle stesse organizzazioni o da terzi in modalità on-premise e off-premise.
- ***Public Cloud****.* resa disponibile in maniera pubblica ed è di proprietà di un organizzazione che vi gestisce la vendita di servizi cloud.
- ***Hybrid Cloud.*** Infrastruttura composizione di due o più cloud (siano essi private, community o pubblici), rimangono entità separate, ma comunque accomunate da standard o tecnologie proprietarie che abilitano un certo livello di portabilità di dati e/o applicazioni di migrazione e/o bursting.

Structure

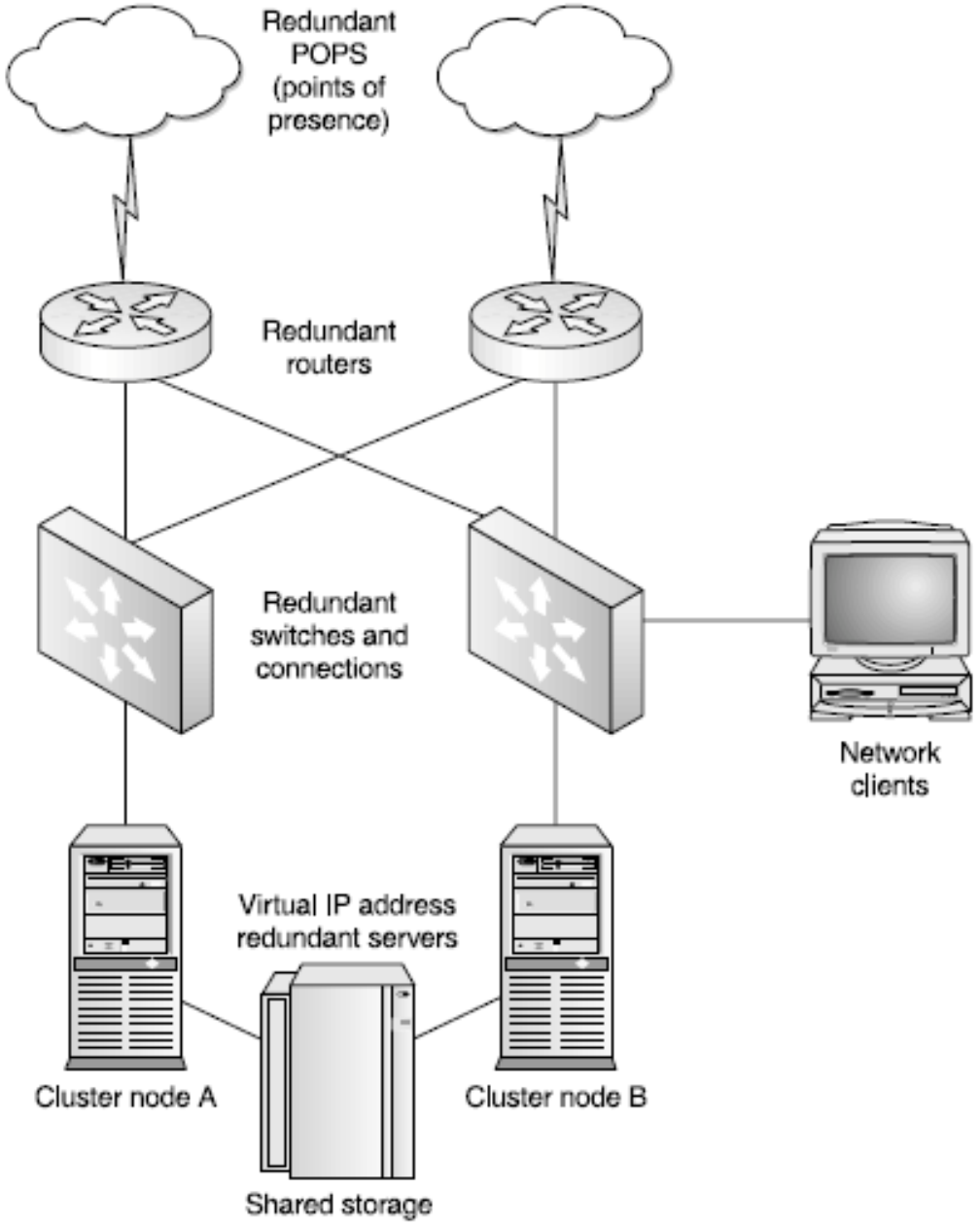
- Elements of cloud Computing
- Virtualization
- Cloud Computing
- High Availability
  - Workload Balancing
  - RAID on HD
  - SAN/NAS
- vSphere Infrastructure
- Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions, SaaS Solutions, PaaS Solutions
- ICARO project

High Availability

- The high availability has to be guaranteed only by the integration of features of:
  - ***High Reliability***
  - ***High Serviceability***
  - ***Fault tolerance***
  - ***Migration of VM to different HW***
  - ***Disaster recovering***

High Availability

- ***High Availability***, available 99.999 % (called “Five Nines”) percent of the time.
- ***Five Nines*** *is the term for saying a service or system will be up almost 100 percent of the* time.
- In case of failure:
  - the path changes to guarantee the service



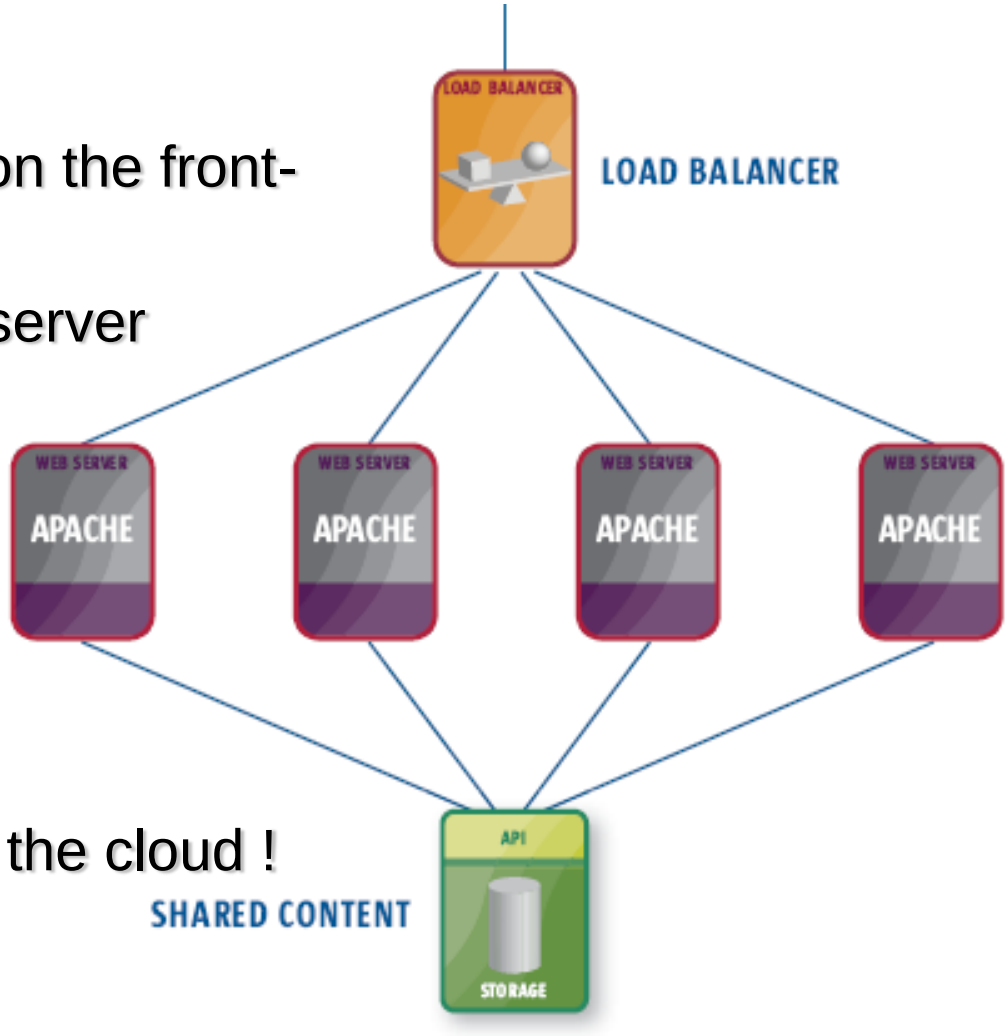
High Availability

- May be achieved by redundancy of :
  - Servers and Services in hot spare or balancing
    - VM based architectures
  - HW/SW: power supply, network connections, etc.

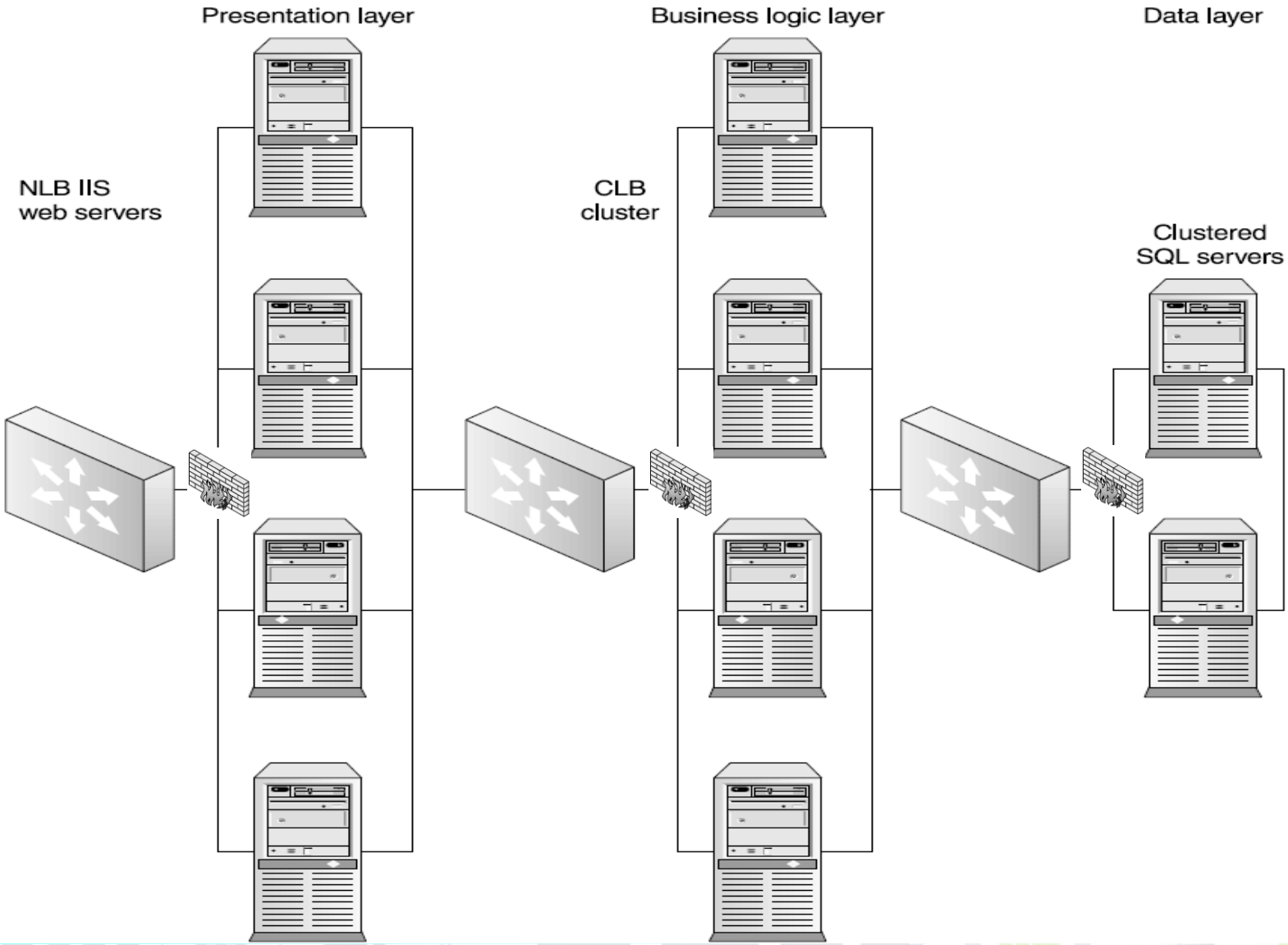
- **Load Balancing/Balancer** according to server traffic/requests
  - Server Clustering, multitier solution
- **Hot Spare**:
  - Server: cloned server to be used when the main is not functioning: heartbeat to detect the server availability and thus failover. (heartbeat signal to communicate the correct running of a process/CPU)
  - HD: Raid based on SAN/NAS, LUN, in host
- **Mixed: balancing and hot spare**

High Availability: Load Balancing

- Service distributed/cloned on more servers
  - According to different policies
    - Round robin
    - Network traffic
- Single transfer rate capability on the front-end of the load balancer
- Sensing the availability of the server on balancer +
  - heartbeat solutions to understand if the servers are alive or not
- Common NAS/SAN
- That is a Cluster of servers on the cloud !

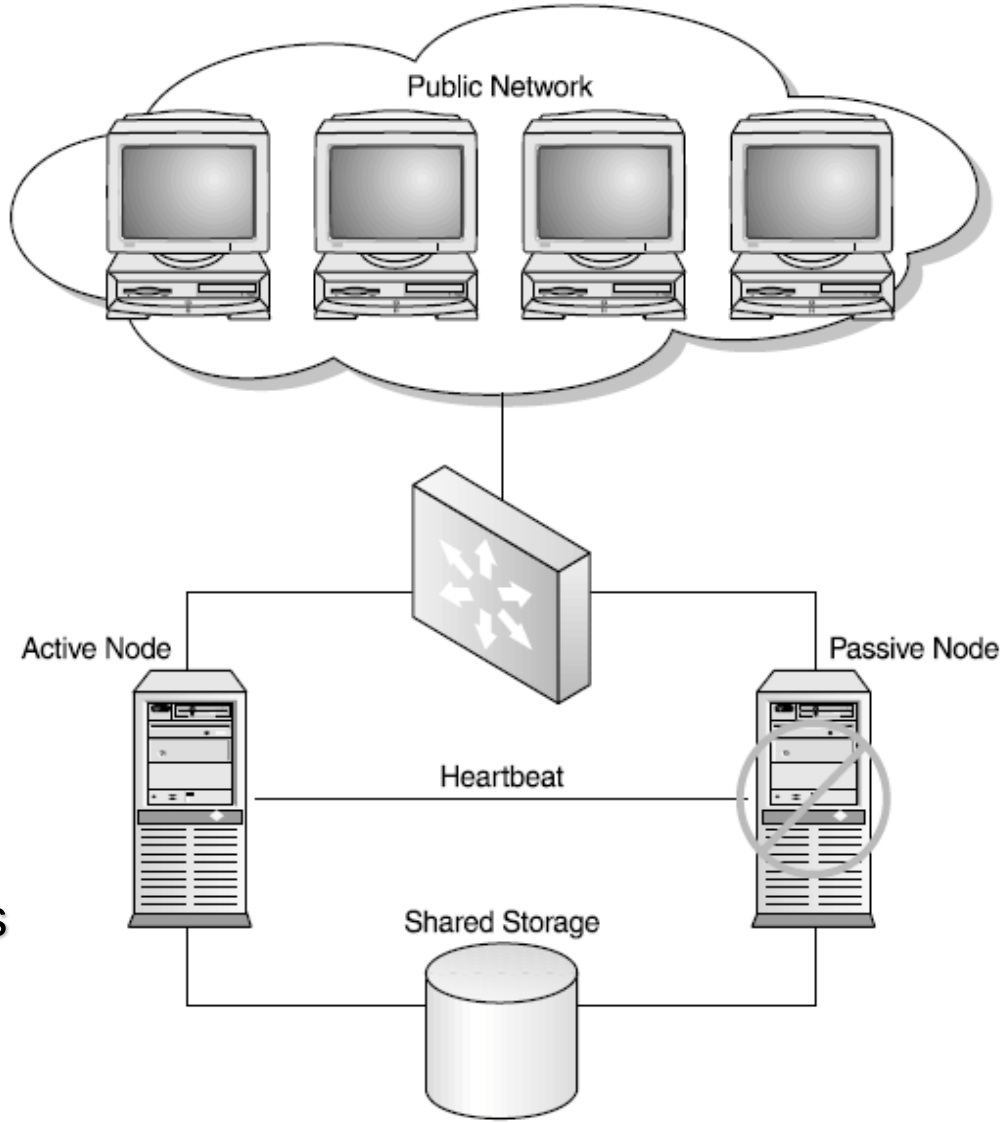


Balancing multi-tier, 3-tier



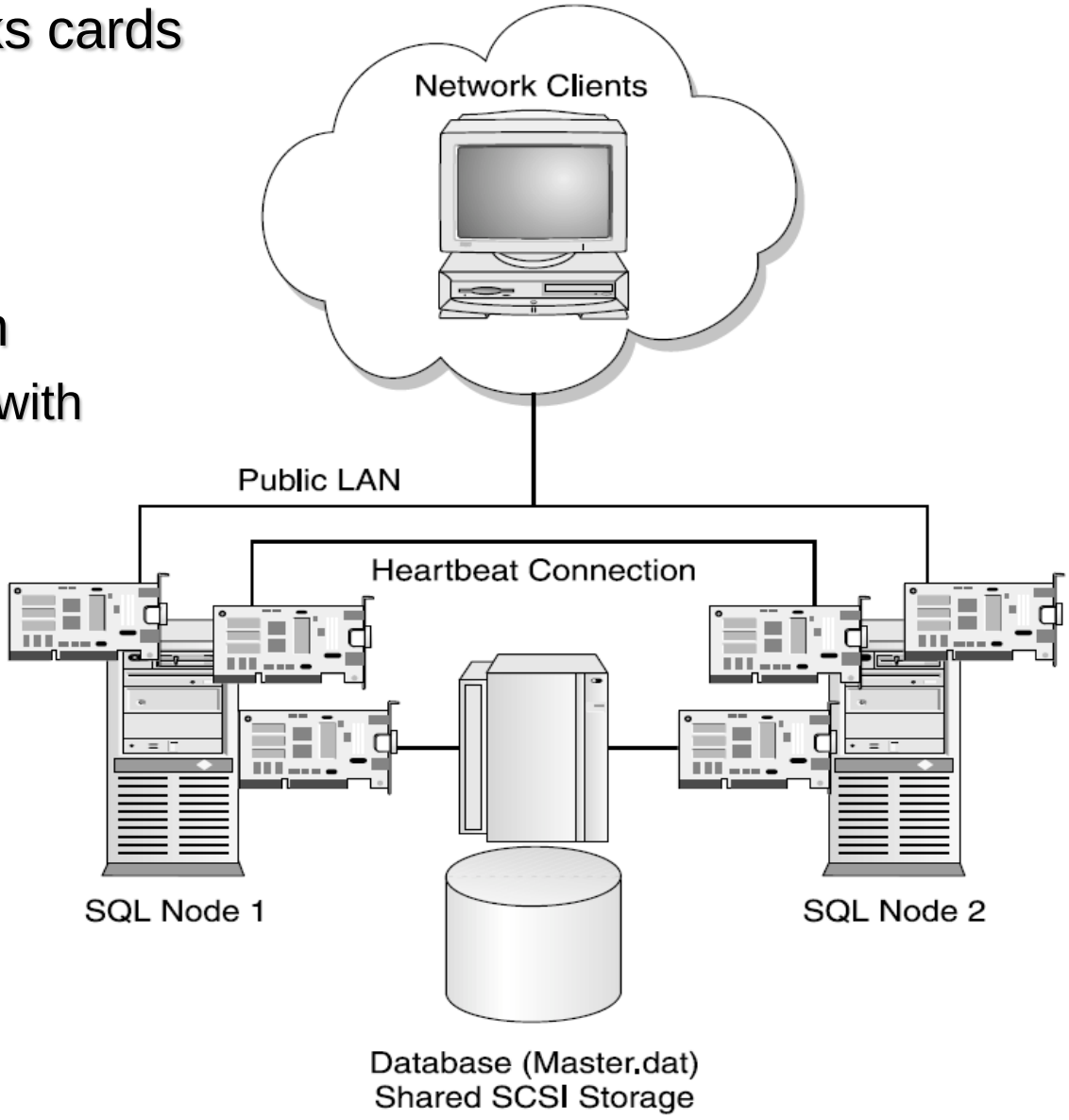
High Availability: Hot spare

- **Fully cloned** servers to be alternatively used when the running one fails
- **Internal Network**
  - *Heartbeat* to detect the server availability and thus failover.
  - *[To keep servers aligned on context and data]*
- **Shared data** storage is a simplification and optional.
  - A different solution may be to have a cloned storage to keep aligned among the two servers

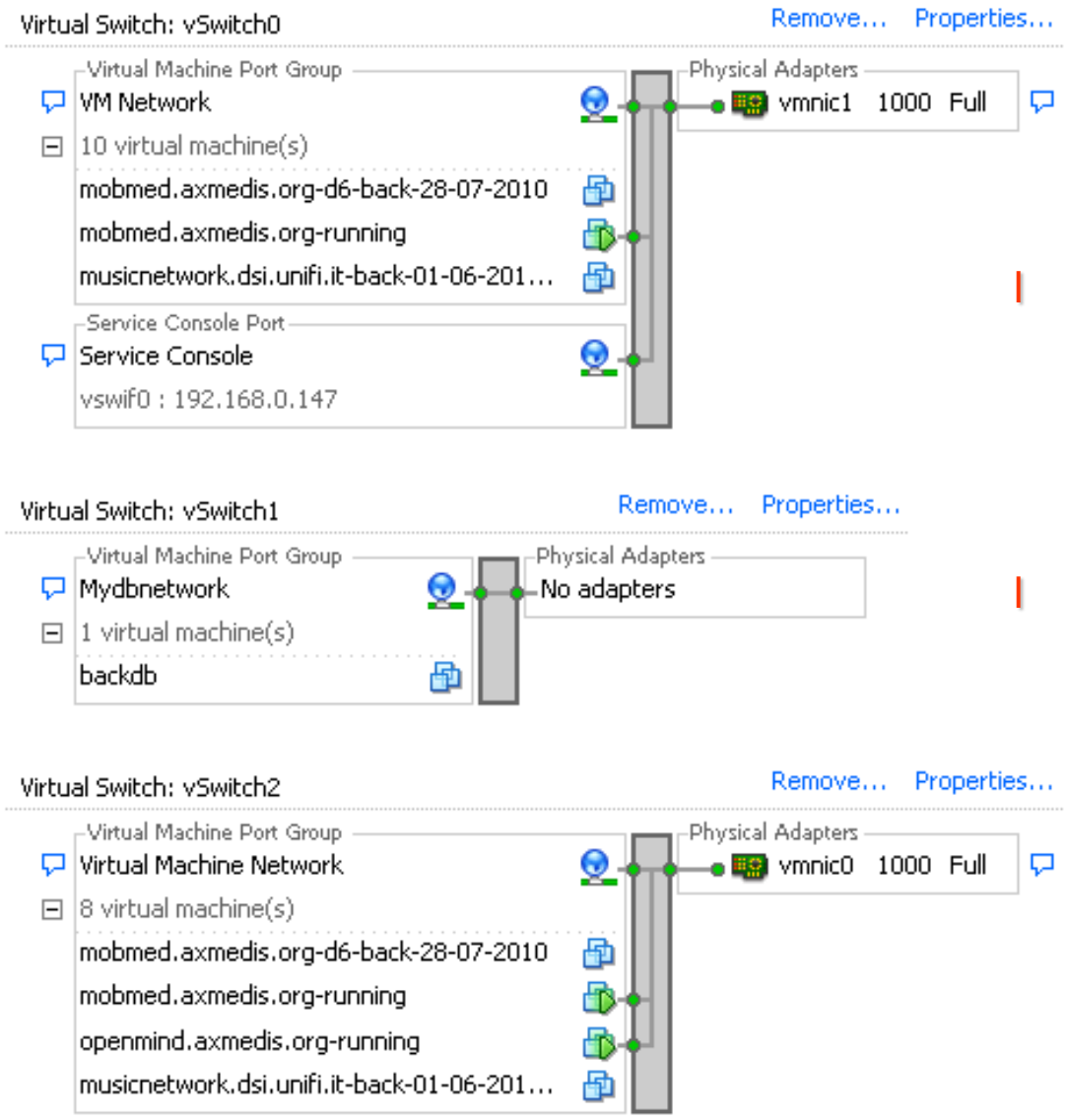


High Availability: Hot spare, hw

- Three separate networks cards
  - Front end
  - Heartbeat
  - Database NAS/SAN
- UPS/APC solutions with
  - 2 UPS, each of which with network card
- NAS/SAN
  - Raid 5 or 6, 60
  - Fiber connection



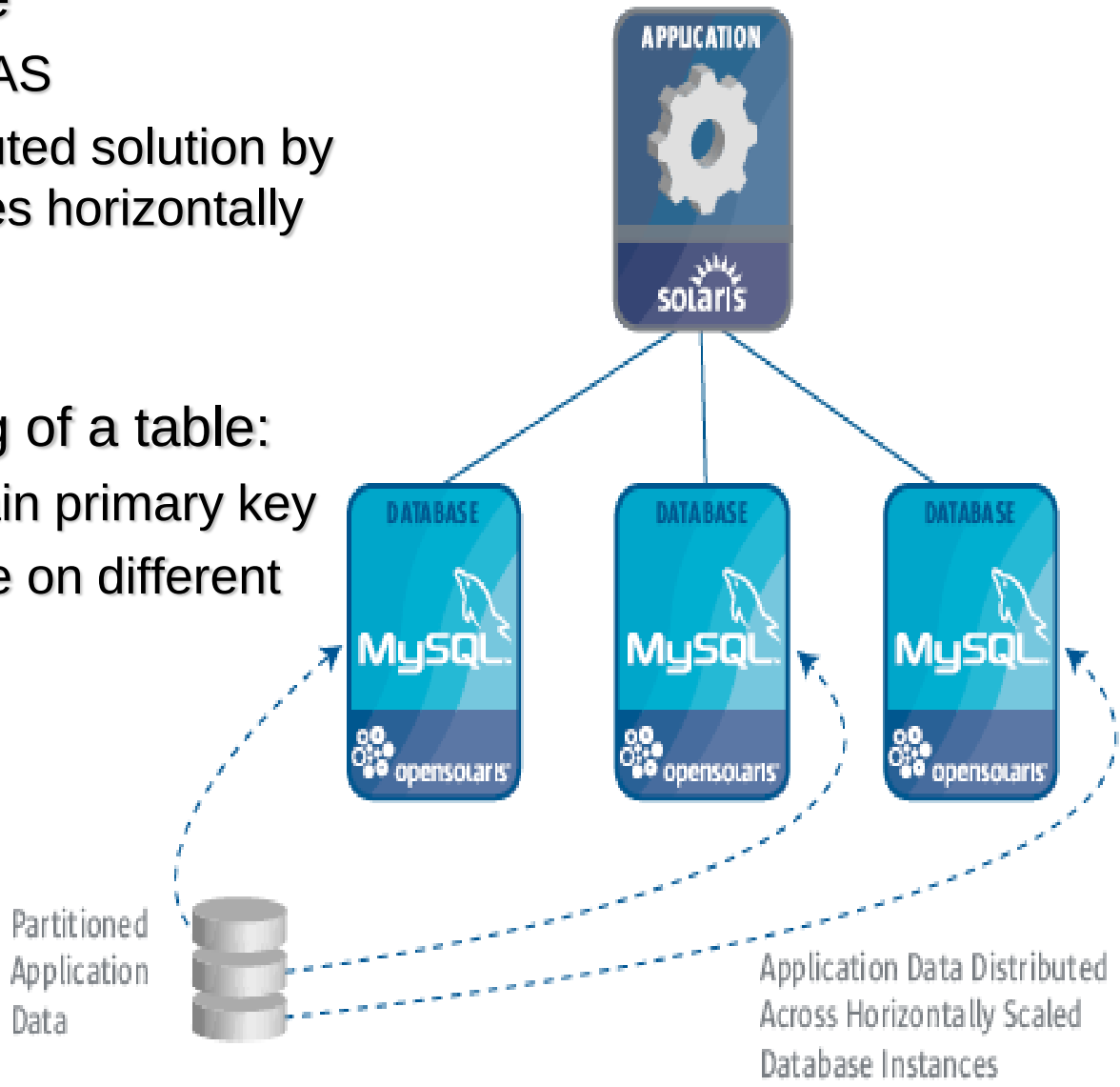
Network and Virtual Networks



- The same VM with access to 2 different network via real network adapters
- A virtual network

Partitioning the database

- The database may be
  - stored in a single NAS
  - stored into a distributed solution by partitioning the tables horizontally or vertically
- Horizontal partitioning of a table:
  - According to a certain primary key
  - Distributing the table on different servers



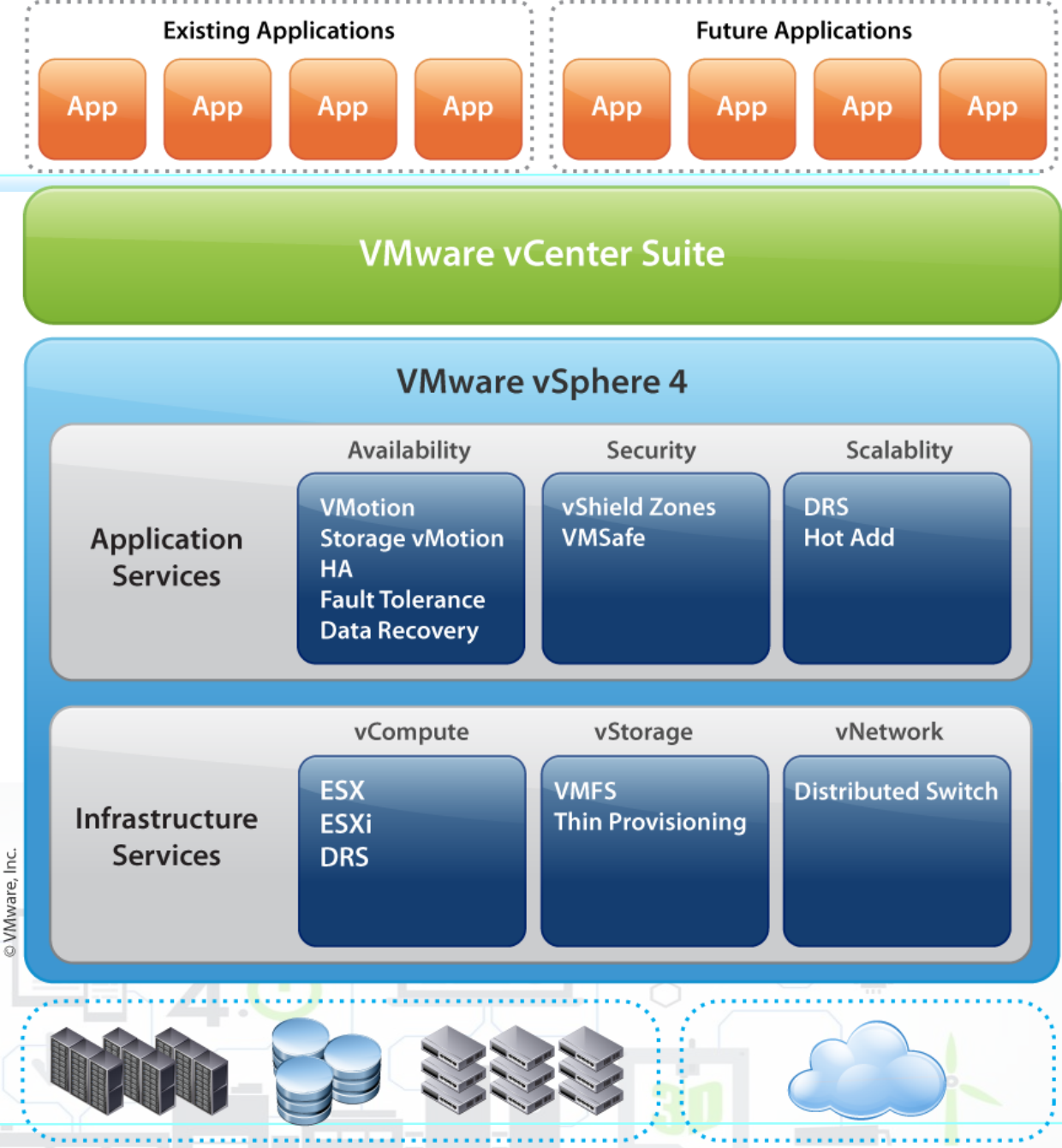
Structure

- Elements of cloud Computing
- Virtualization
- Cloud Computing
- High Availability
- vSphere Infrastructure
  - Vmotion
  - Power Management
  - Resource Scheduling
  - Fault Tolerance
- Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions, SaaS Solutions, PaaS Solutions
- ICARO project

vSphere 4 infrastructure of VMware

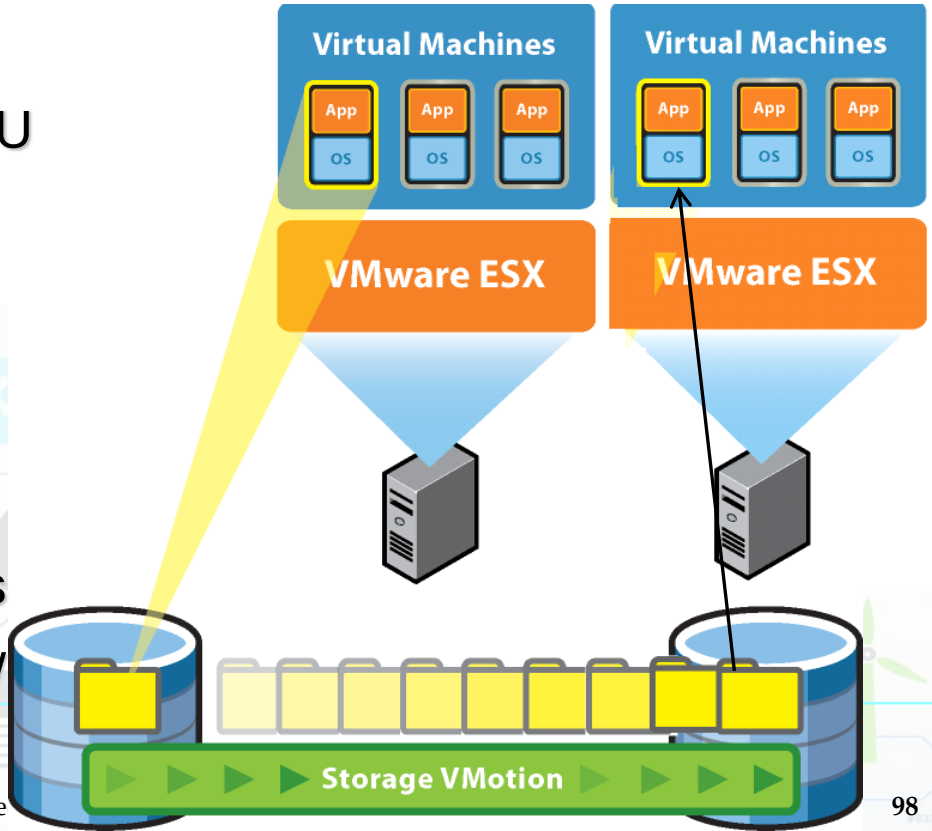
- **High level features:**
  - HA: high availability
  - DRS: Distributed Resource Scheduling
  - Creating Fault Tolerance architectures
  - DPM: datacenter power management, based on VMotion
  - Converting VM into VM for infrastructure, from physical to VM
  - vApp: are Virtual Application Services
  - Cloning and Moving VM
  - Making Templates for VM
  - Backup VM and thus virtual servers
- **Low level features:**
  - VMotion: VM moving among Hosts
  - Dynamic increment of: CPU ck, MEM, net…

Summary of VMware vSphere 4.0



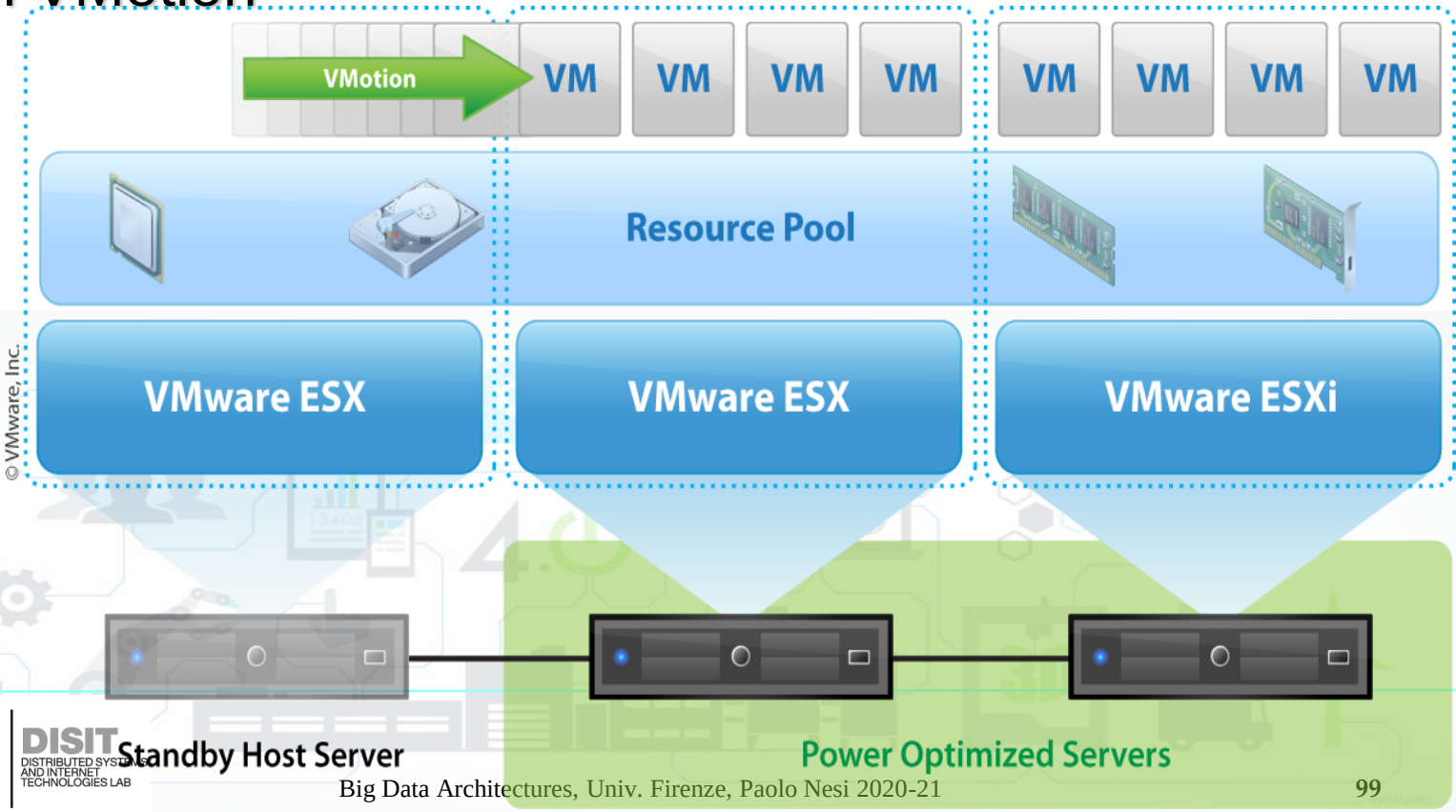
VMotion of VMware vSphere

- **you need**:
  - VM without snapshots
  - VM must be powered off to simultaneously migrate both host and datastore
  - Compatibility among host CPU and VMs
  - Dedicated virtual network
  - **The VMs** can be ON
- Steps:
  1. moving HD images
  2. aligning OS and CPU status
  3. off-the old and then on-new



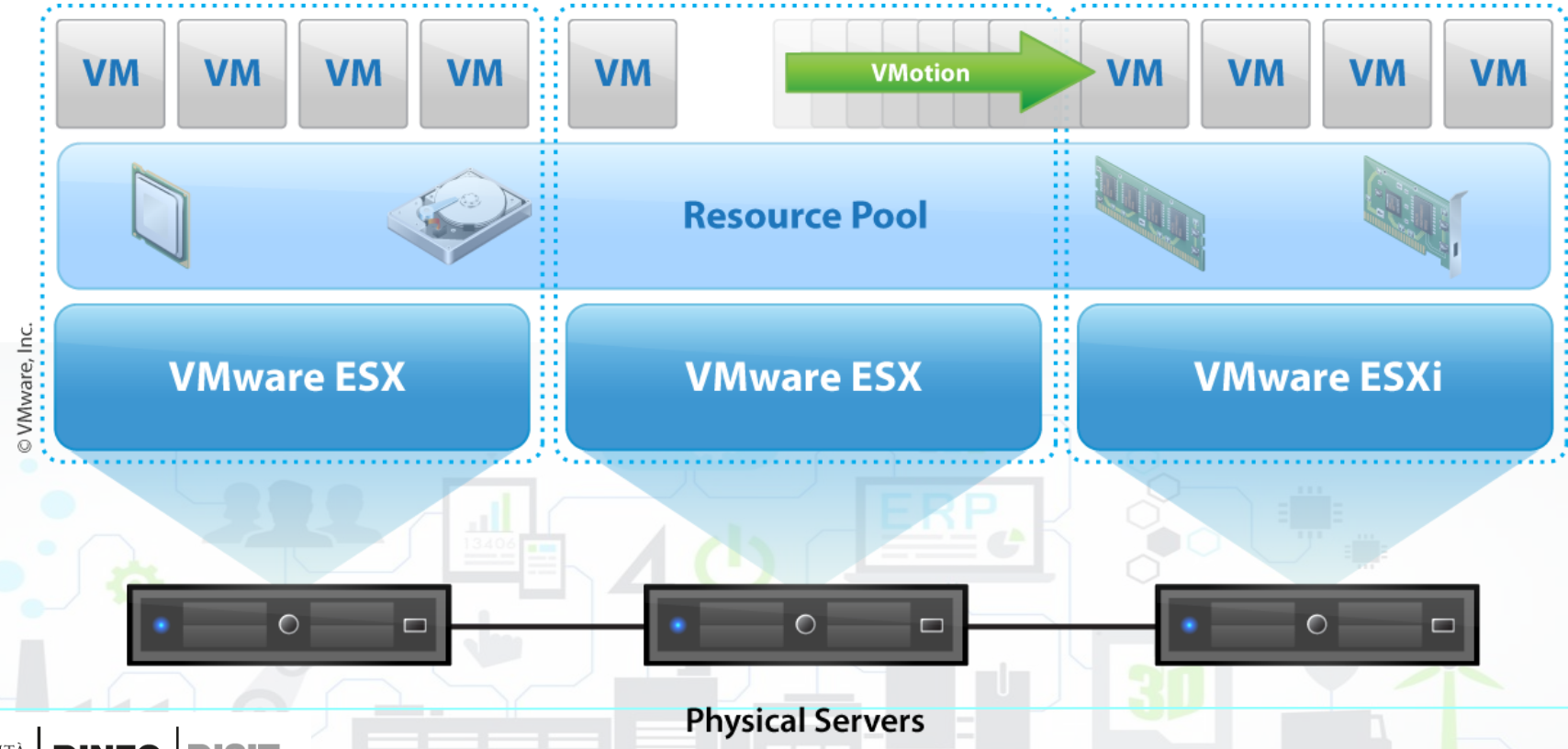
VMware DPM, Power Management

- DPM consolidates workloads to reduce power consumption
  - Cuts power and cooling costs
  - Automates management of energy efficiency
  - optimizing host resources
- Based on VMotion



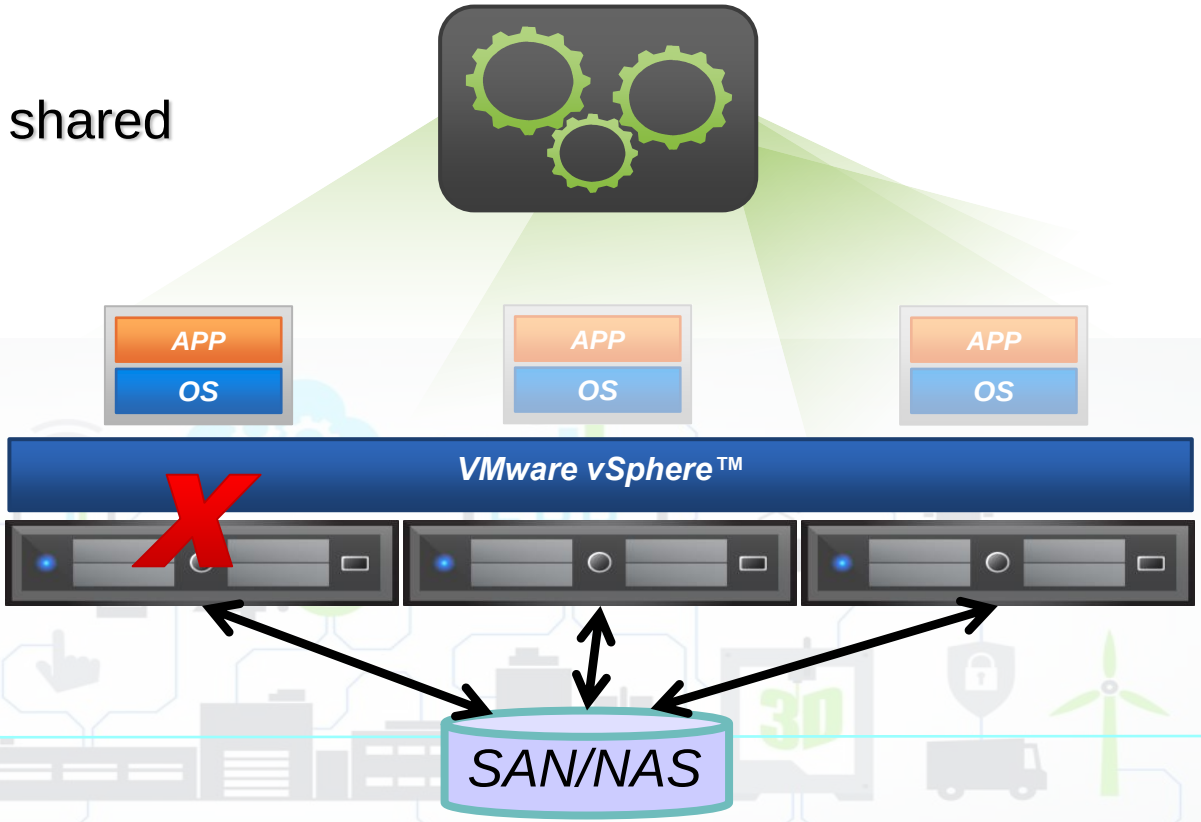
VMware DRS, Distrib. Res. Scheduling

- DRS is used to balance the workload among Hosts
- Moving VMs is a tools for balancing the workload on Hosts
- ..

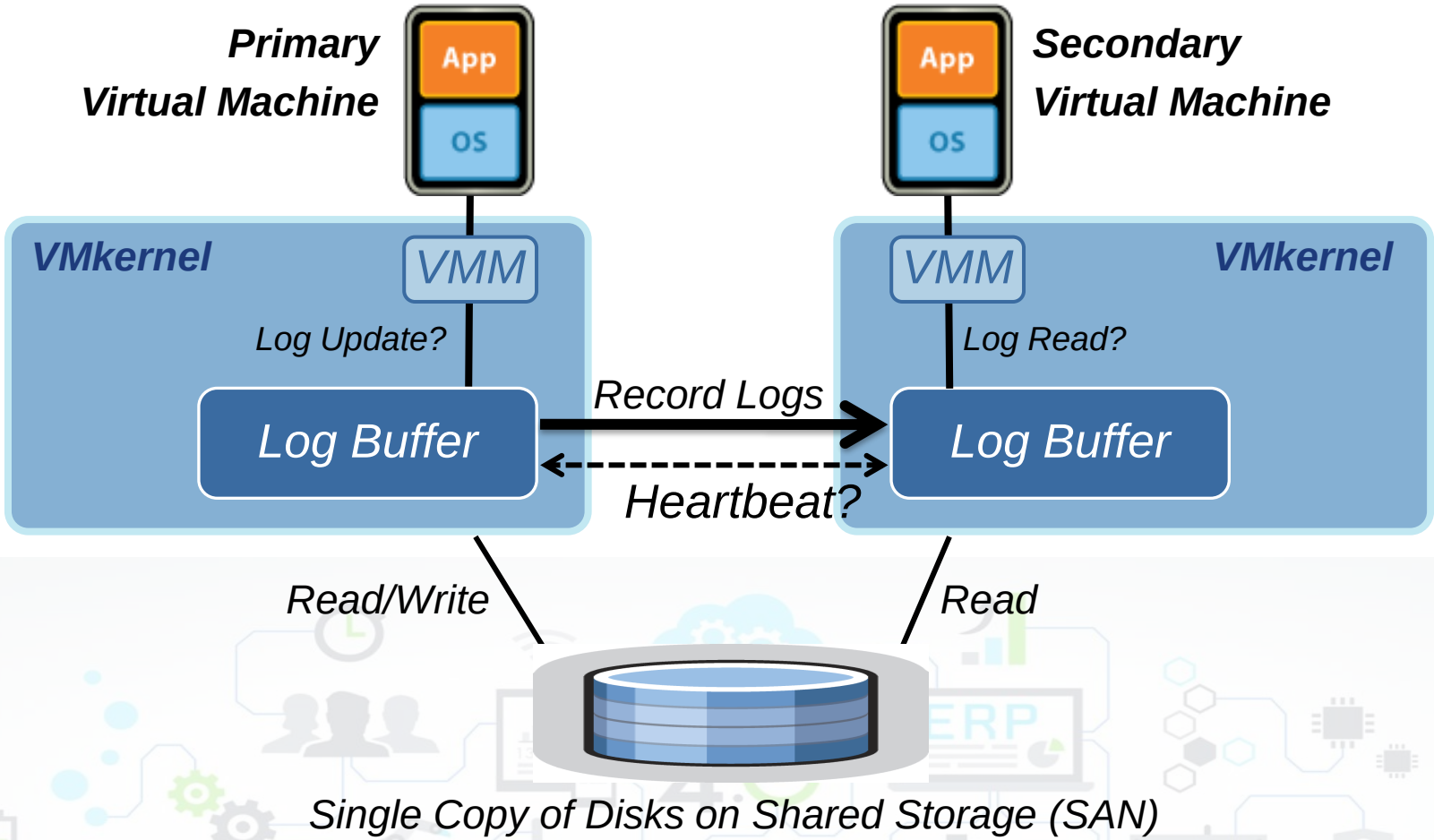


VMware Fault Tolerance, FT

- Single identical VMs running in lockstep on separate hosts
- Zero downtime, zero data loss failover for all virtual machines in case of hardware failures
- Single common mechanism for all applications and Operating systems
- Need to have a storage shared by the same VM
- The VM itself can be stored in the SAN

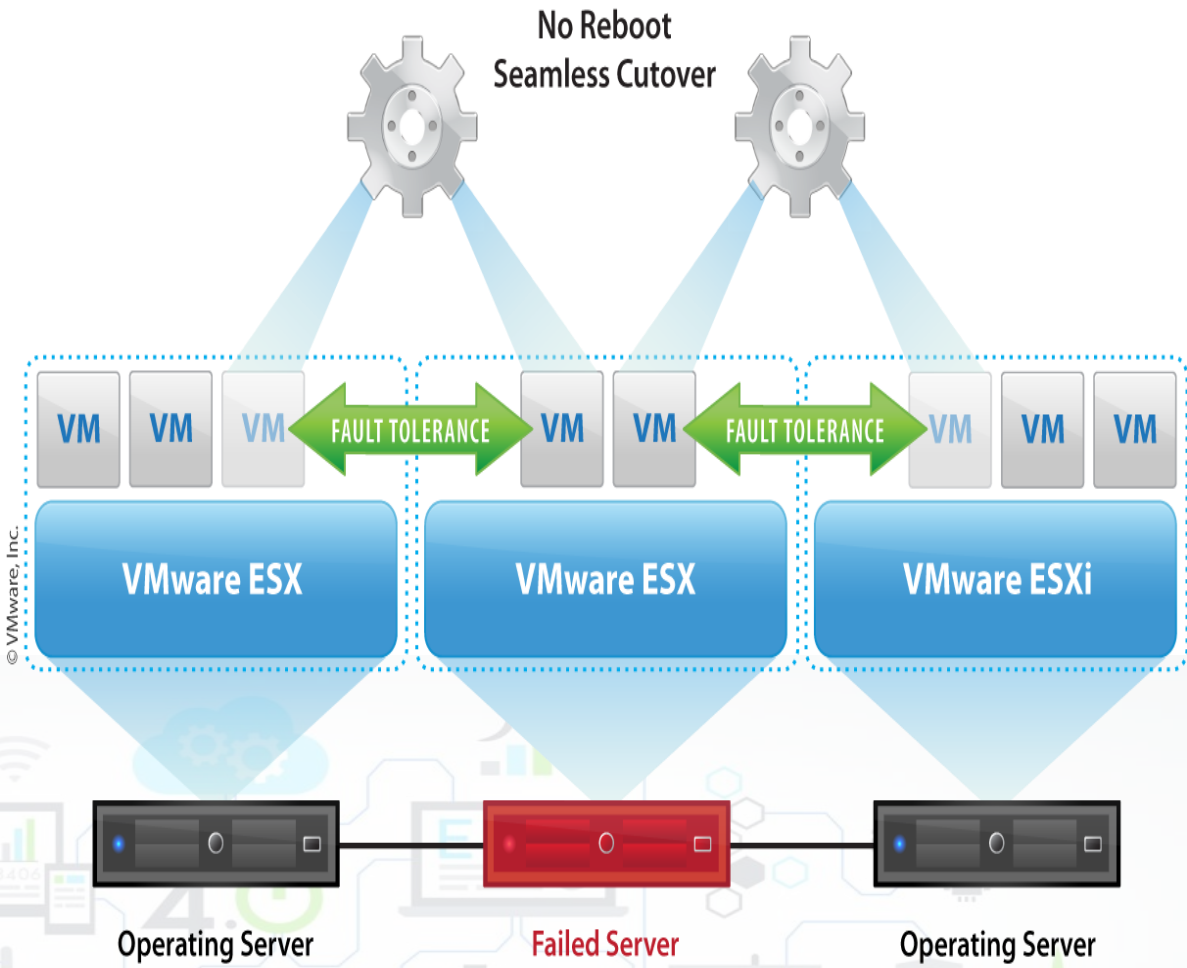


How VMware FT Works



HA: High Availability of vSphere

- **When a host fails**, the running VM on the host may be turned ON on another hosts
  - Just the time to turn on again the host
- **HOT Spare** solution:
  - It is also possible to keep aligned 2 distinct hosts to make a faster switch OFF→ON of the VM on the faulty host
  - implies to have duplicated resources: Host, CPU etc.



Structure

- Elements of cloud Computing
- Virtualization
- Cloud Computing
- High Availability
- vSphere Infrastructure
- Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions, SaaS Solutions, PaaS Solutions
- ICARO project

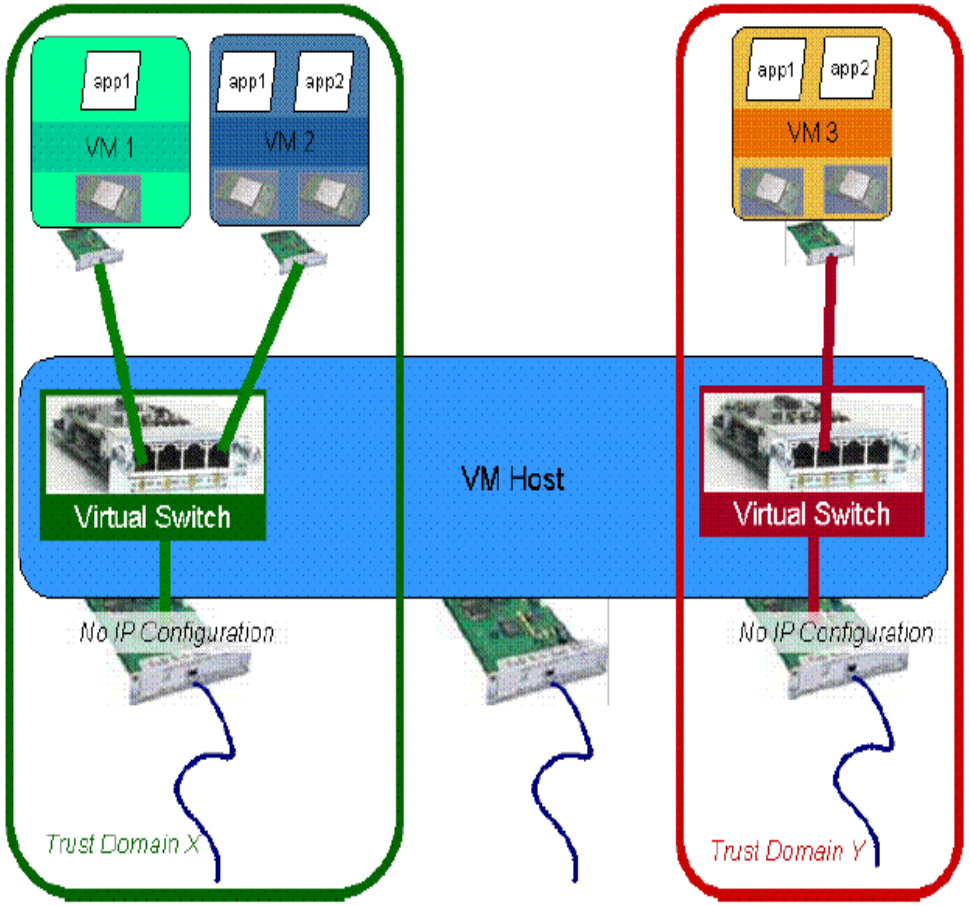
Security on the cloud

- Protecting VMs from external access
- Protecting VMs each other in the cloud
- Technologies:
  - Accessing to other VM via dedicated virtual networks, using Virtual Networking, Virtual Switch
  - Avoiding shared disk, at least using authenticated connections
  - Using Firewall
  - Communicating with other VMs via protected connections: protected WS, HTTPs, SSL, SFTP, etc.

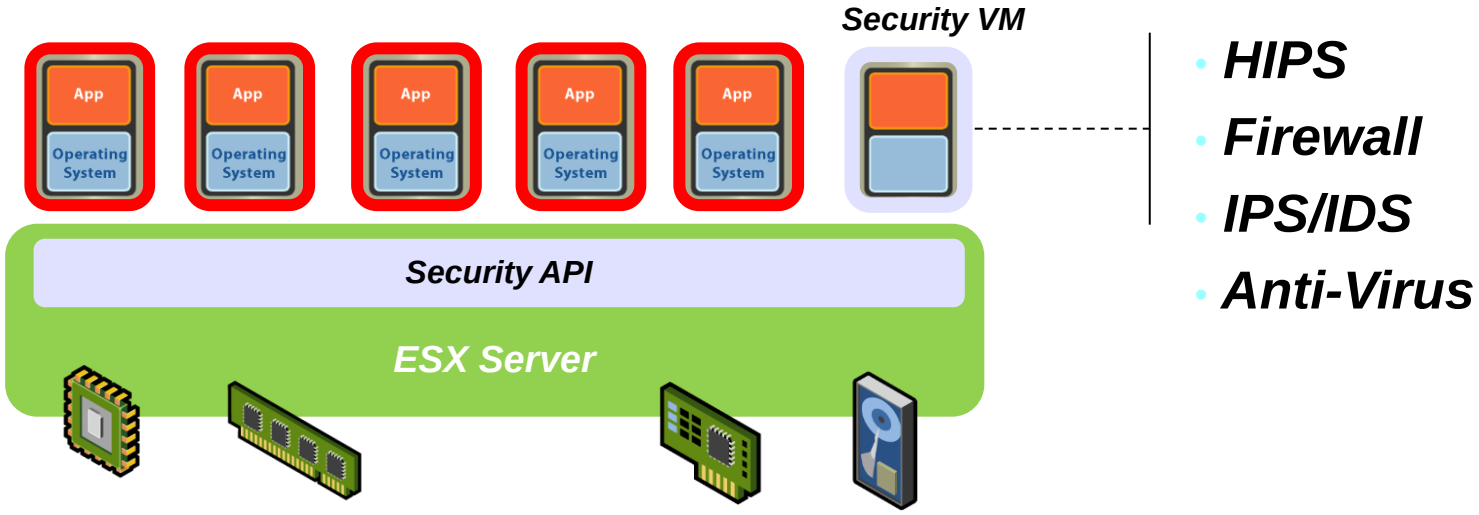
Virtual Networking

- Per isolare meglio dei guest OS da altri che non sono nello stesso trust domain
- isolare i virtual switch dei singoli trust domain.
- Solo i guest che condividono lo stesso domain hanno schede di rete virtuali sullo stesso virtual switch.
- virtual switch su porte logiche del sistema host che non hanno indirizzo ip configurato.

Figure 1 - Isolating virtual machines to separate network trust domains



Vmware Vsphere VMsafe

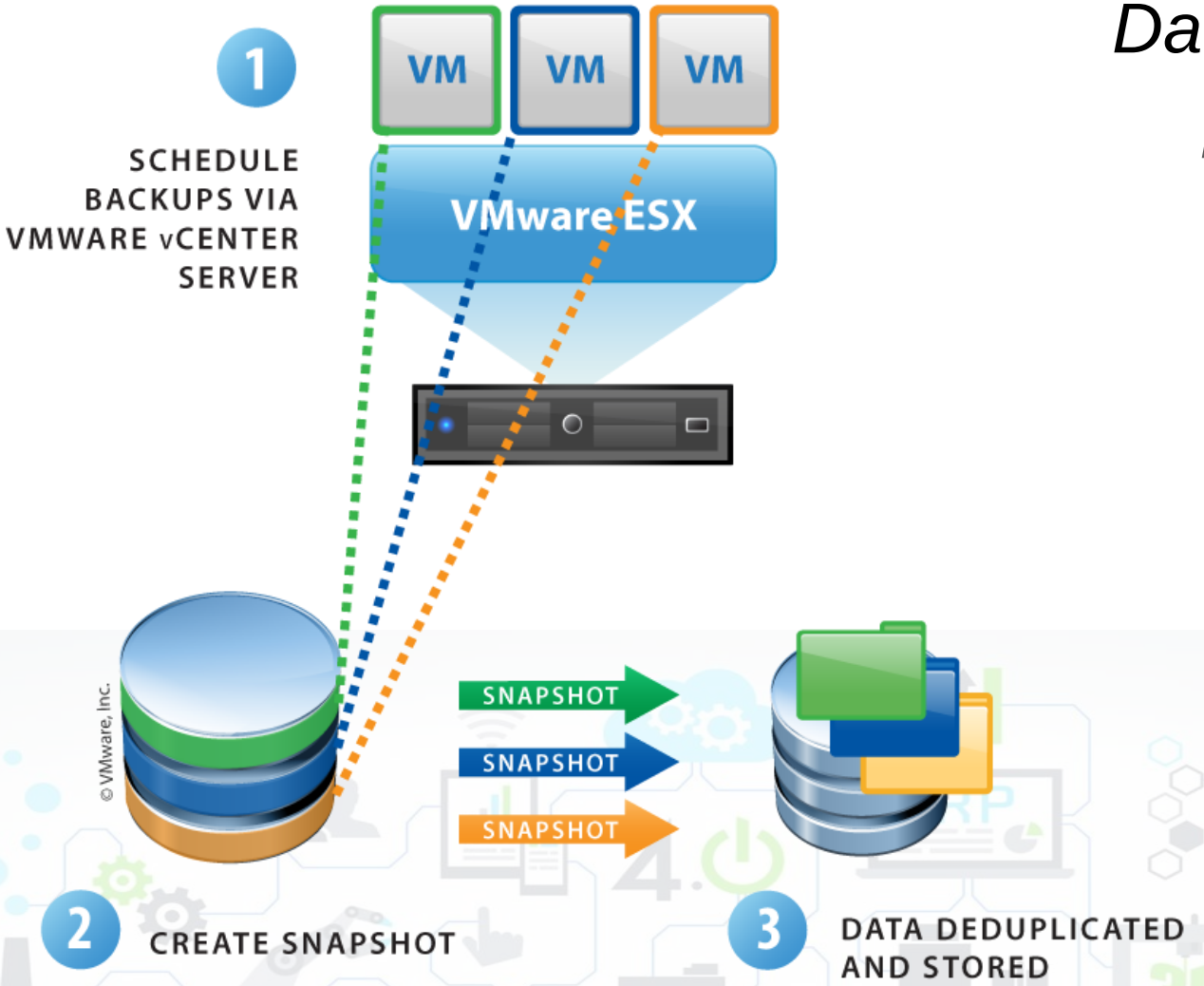


- Creates a new, stronger layer of defense – fundamentally changes protection profile for VMs running on VMware Infrastructure
- Protect the VM by inspection of virtual components (CPU, Memory, Network and Storage)
- Complete integration and awareness of VMotion, Storage VMotion, HA, etc.
- Provides an unprecedented level of security for the application and the data inside the VM

VMware vSphere

Data Recovery

Backup Only

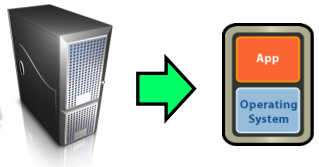


Structure

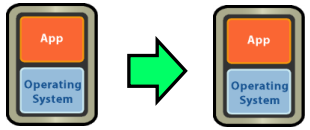
- Elements of cloud Computing
- Virtualization
- Cloud Computing
- High Availability
- vSphere Infrastructure
- Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions, SaaS Solutions, PaaS Solutions
- ICARO project

VM Converter, the migration

- **Conversion possibilities, migration possibilities**
- **P2V, from Physical → Virtual machine**

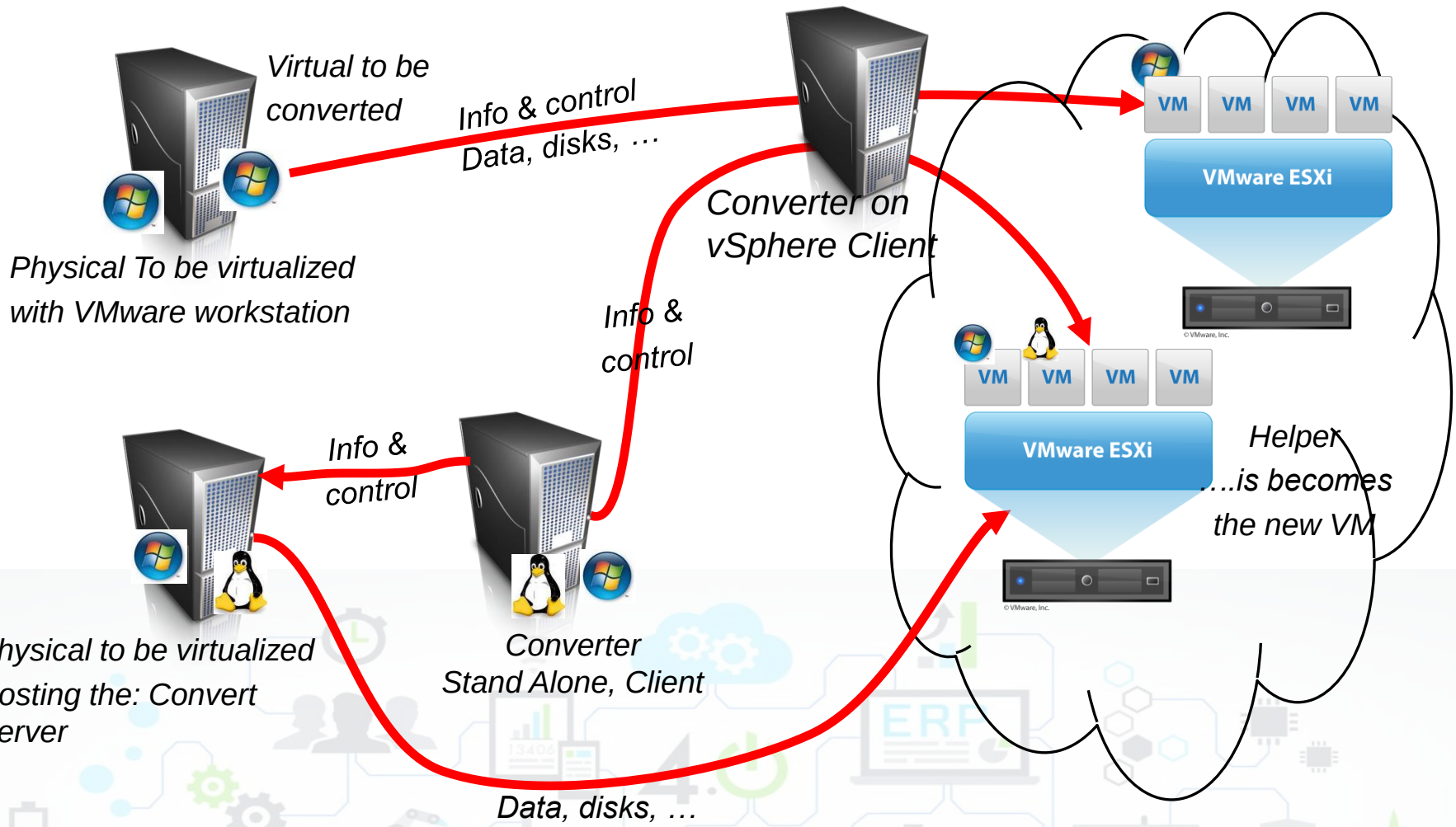


  - Reusing legacy servers into stronger and new HW machines
  - From ISO CD of an OS → VM
- **V2V, from Virtual → to Virtual**

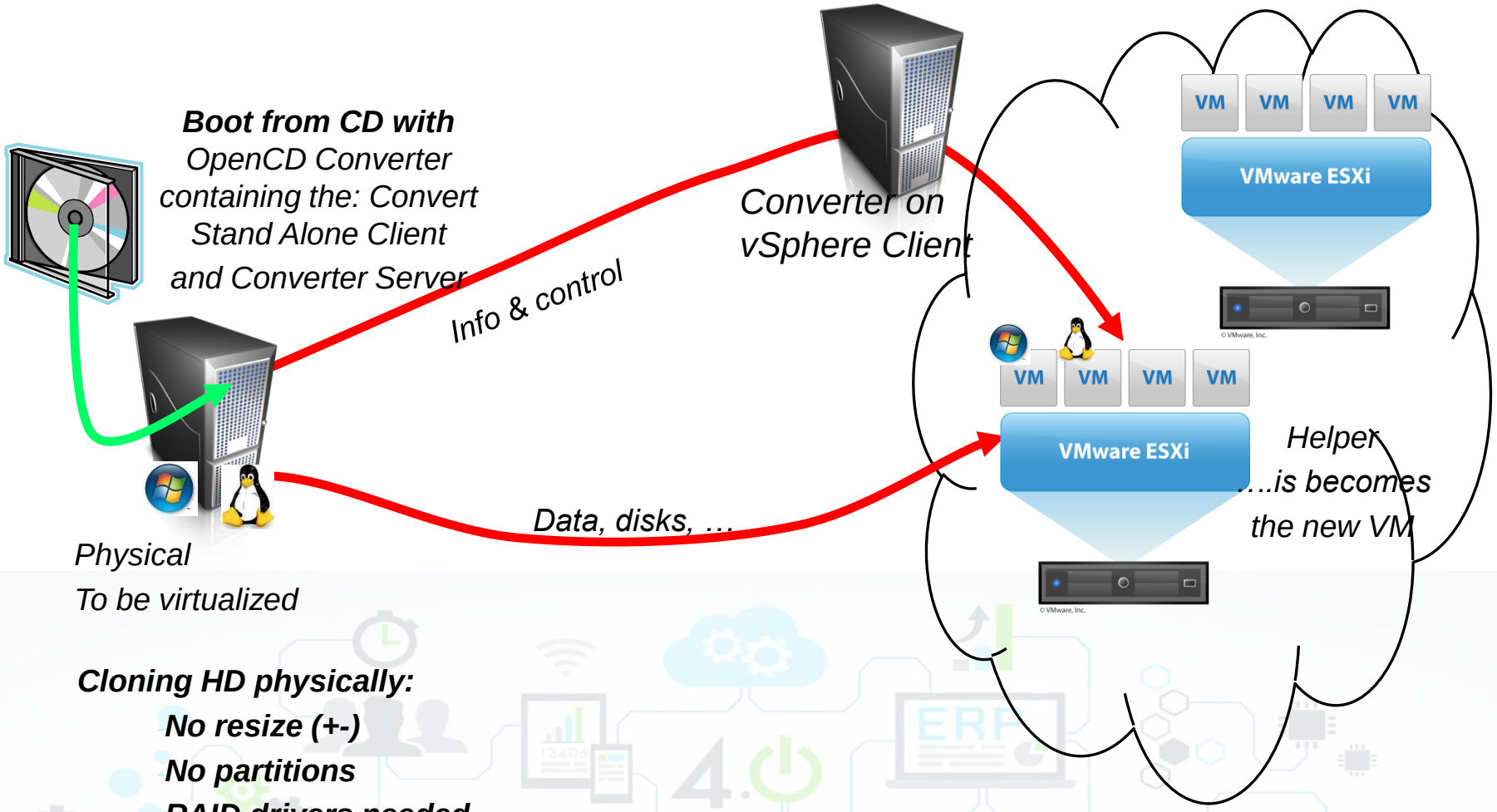


  - Import/export a VM from/to different standards
  - From VM Workstation → Infrastructure VM
  - From Infrastructure VM → template for VM with some parameters
  - From Infrastructure VM →VM Workstation
  - From Infrastructure VM → Infrastructure VM changing parameters
  - …..

P2V, vConverter vSphere



P2V, OpenCD vConverter vSphere

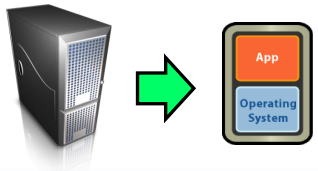


Cloning HD physically:

- ***No resize (+-)***
- ***No partitions***
- ***RAID drivers needed***

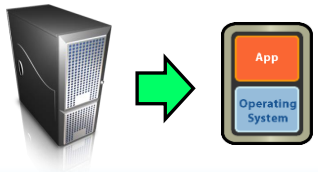
P2V conversion

- **How to convert**, typically:
  - From **VM Converter Stand Alone** on Windows or Linux
    - it includes a Server side and a client consol,
      - server side has to be installed on the P machine to be converted.
      - Client side is a consol
    - From Linux to VM on ESX Host
    - From Windows to CM on ESX Host
  - From **VM converter as plugin of vCenter**, via vCenter Client
    - From Windows to VM on ESX Host
  - From **OpenCD**, with included VM Converter and alone
    - From Windows to CM on ESX Host
  - From **VMware Workstation**
    - From Windows to CM on ESX Host

P2V conversion

- **Hypothesis:**
  - Converter server installed on Physical machine
  - Converter in third machine or on the physical (windows)
  - All snapshots are removed and the disks re-compacted.
- **First step:**
  - The Converter Client is launched
  - The Converter Client creates an Helper VM on the vCenter Host
- **Second step:**
  - The Helper VM works independently to transfer the data from VM Converter Server to the new VM via the Helper VM, under the control of Client.
  - The Converter Client provides commands to close and destroy the Helper VM and put in execution the new VM with the final shape

P2V conversion

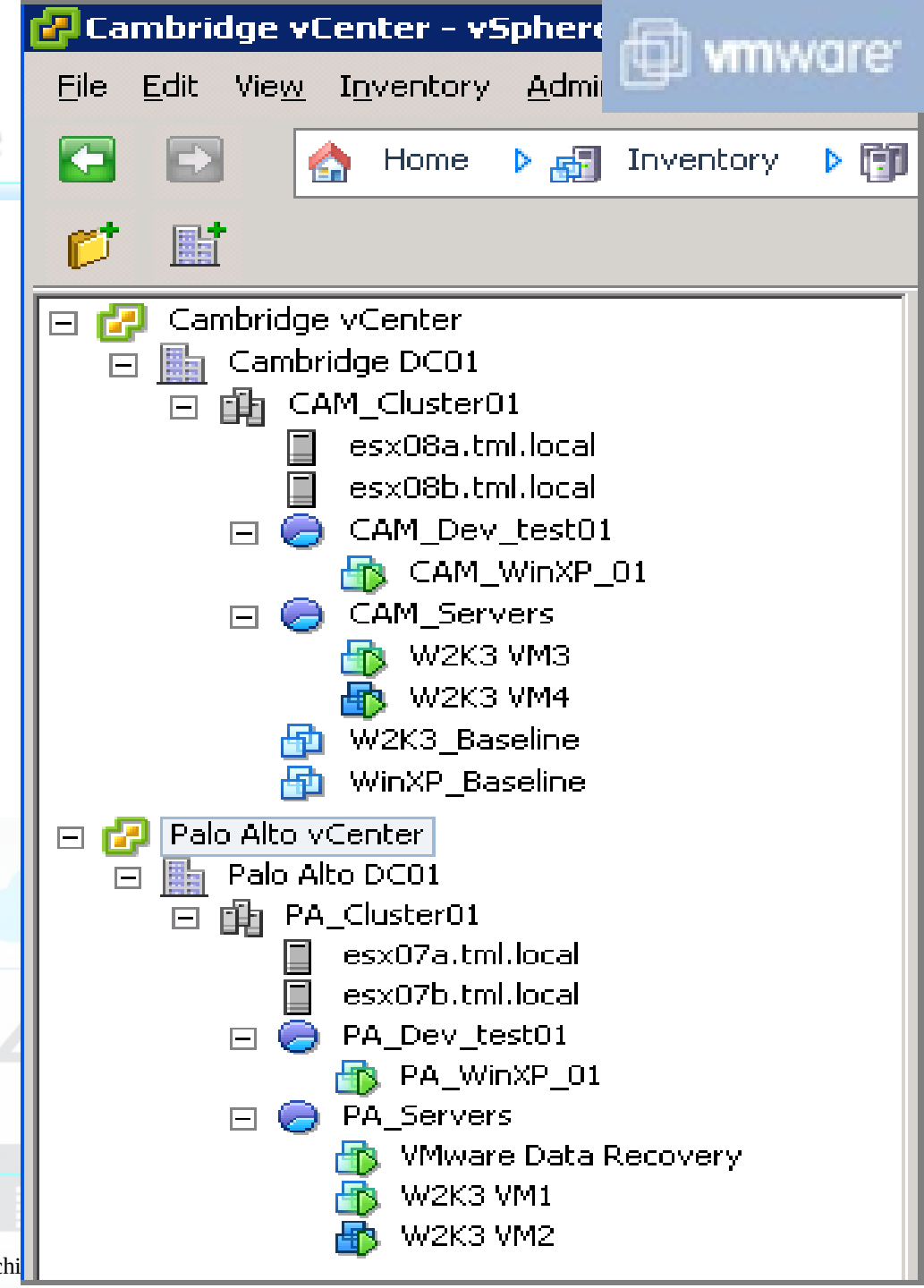
- **Third step:**
  - Reboot the new machine as Virtual machine
    - Then install the VMware tools on the VM from the vCenter or from the VMware Workstation hypervisor.
    - Revise the general config, network, etc.
    - See if it possible to use paravirtualized drivers instead of physical drivers.
  - If the converted VM is not booting
    - Verify the VM setting: disk, operating system, disk drivers, etc.
    - Regenerate the VM tools on the VM and try again

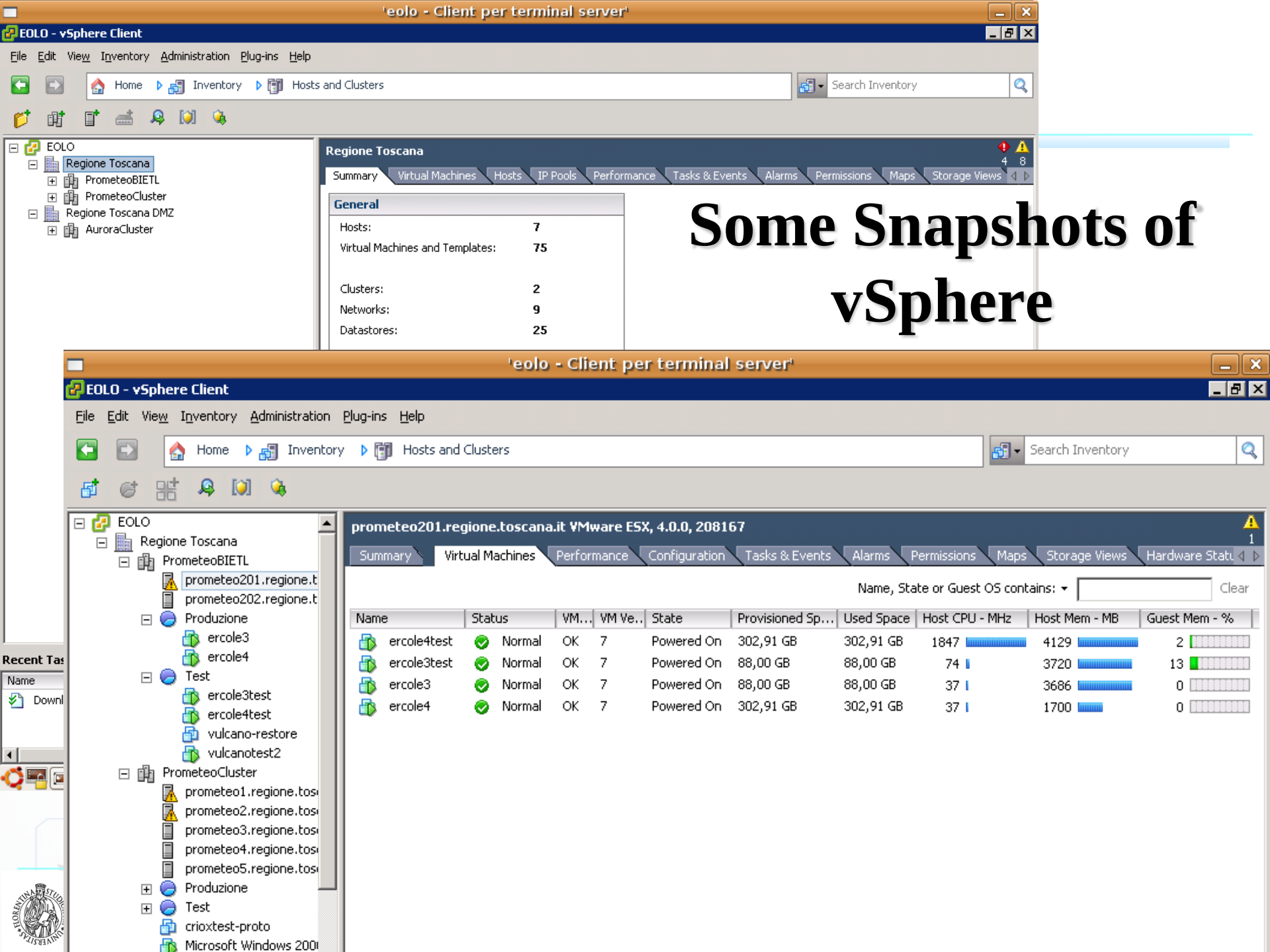
Structure

- Elements of cloud Computing
- Virtualization
- Cloud Computing
- High Availability
- vSphere Infrastructure
- Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions, SaaS Solutions, PaaS Solutions
- ICARO project

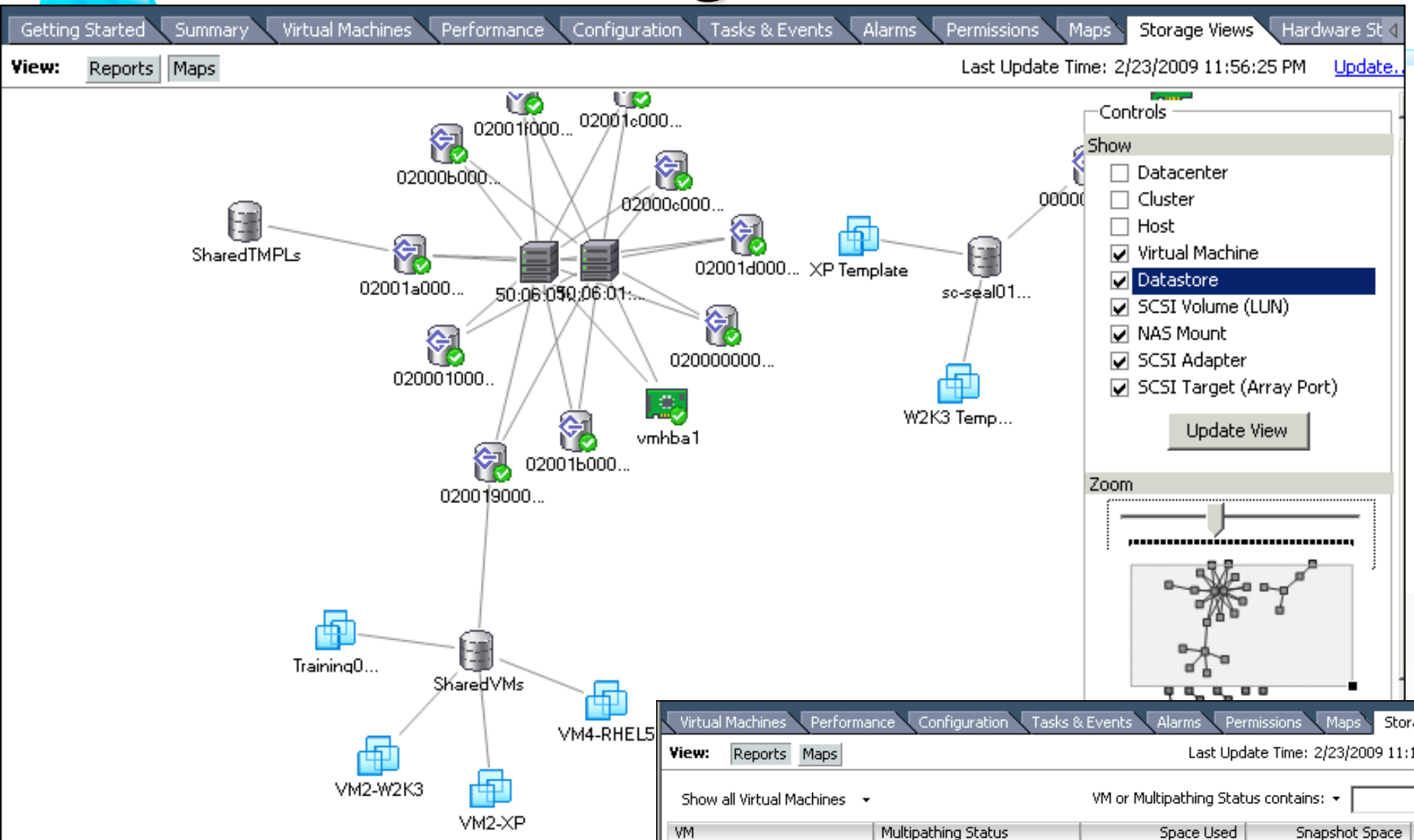
vCenter vSphere

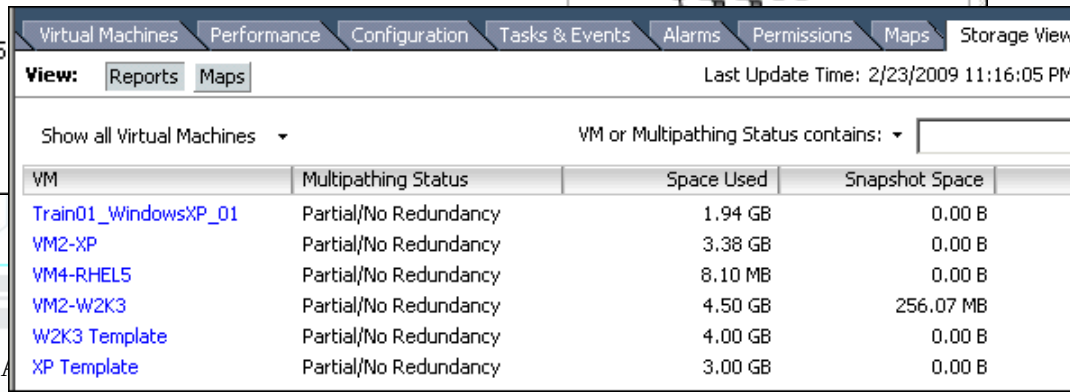
- The so called Inventory
- Datacenter
  - Cluster00
    - Host….
    - Host….
    - Application:
      - VM1
      - VM2
    - VM3
    - VM4
  - Cluster01
    - H2..
    - …




'eolo - Client per terminal server'
EOLO - vSphere Client
File Edit View Inventory Administration Plug-ins Help
Home Inventory Hosts and Clusters
Search Inventory
EOLO
Regione Toscana
PrometeoBIETL
PrometeoCluster
Regione Toscana DMZ
AuroraCluster
Regione Toscana
Summary Virtual Machines Hosts IP Pools Performance Tasks & Events Alarms Permissions Maps Storage Views
General
Hosts: 7
Virtual Machines and Templates: 75
Clusters: 2
Networks: 9
Datastores: 25
Some Snapshots of vSphere
'eolo - Client per terminal server'
EOLO - vSphere Client
File Edit View Inventory Administration Plug-ins Help
Home Inventory Hosts and Clusters
Search Inventory
EOLO
Regione Toscana
PrometeoBIETL
prometeo201.regione.t
prometeo202.regione.t
Produzione
ercole3
ercole4
Test
ercole3test
ercole4test
vulcano-restore
vulcanotest2
PrometeoCluster
prometeo1.regione.tos
prometeo2.regione.tos
prometeo3.regione.tos
prometeo4.regione.tos
prometeo5.regione.tos
Produzione
Test
crioxtest-proto
Microsoft Windows 200
prometeo201.regione.toscana.it VMware ESX, 4.0.0, 208167
Summary Virtual Machines Performance Configuration Tasks & Events Alarms Permissions Maps Storage Views Hardware Statu
Name, State or Guest OS contains:
Clear
Name Status VM... VM Ve.. State Provisioned Sp... Used Space Host CPU - MHz Host Mem - MB Guest Mem - %
ercole4test Normal OK 7 Powered On 302,91 GB 302,91 GB 1847 4129 2
ercole3test Normal OK 7 Powered On 88,00 GB 88,00 GB 74 3720 13
ercole3 Normal OK 7 Powered On 88,00 GB 88,00 GB 37 3686 0
ercole4 Normal OK 7 Powered On 302,91 GB 302,91 GB 37 1700 0

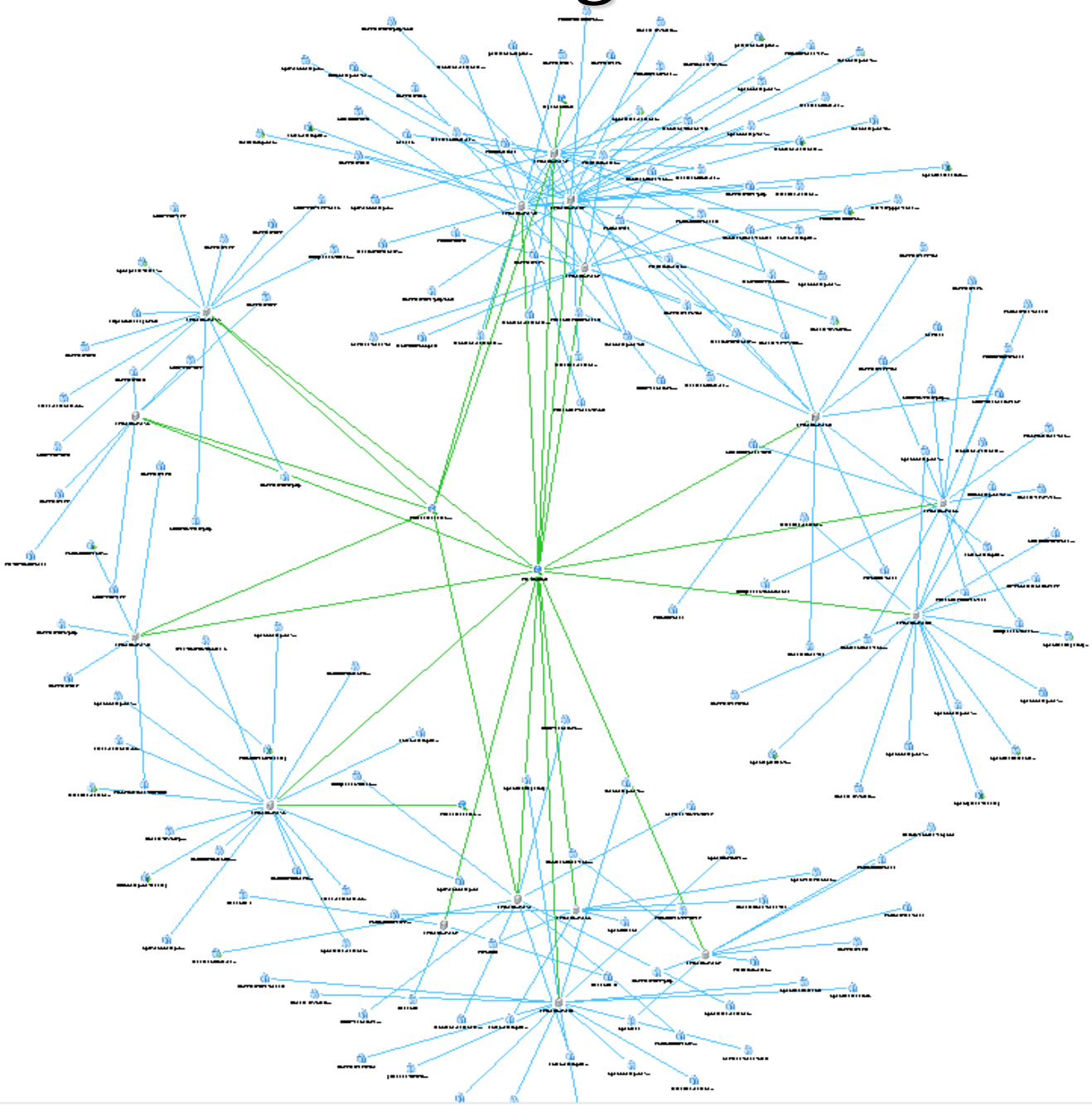
Storage View





VM	Multipathing Status	Space Used	Snapshot Space
Train01_WindowsXP_01	Partial/No Redundancy	1.94 GB	0.00 B
VM2-XP	Partial/No Redundancy	3.38 GB	0.00 B
VM4-RHEL5	Partial/No Redundancy	8.10 MB	0.00 B
VM2-W2K3	Partial/No Redundancy	4.50 GB	256.07 MB
W2K3 Template	Partial/No Redundancy	4.00 GB	0.00 B
XP Template	Partial/No Redundancy	3.00 GB	0.00 B

A Larger Datacenter



Time since last data update: 01:02 Refresh

Overview

Map Relationships:

Custom Map

Host Options

- [x] Host to VM
- [x] Host to Network
- [ ] Host to Datastore

VM Options

- [ ] Fault Tolerance relationships
- [ ] VM to Network
- [ ] VM to Datastore
- [ ] Show only powered on VMs

Apply Relationships

vmware

Partially Collapsed with Separate Physical Trust Zones

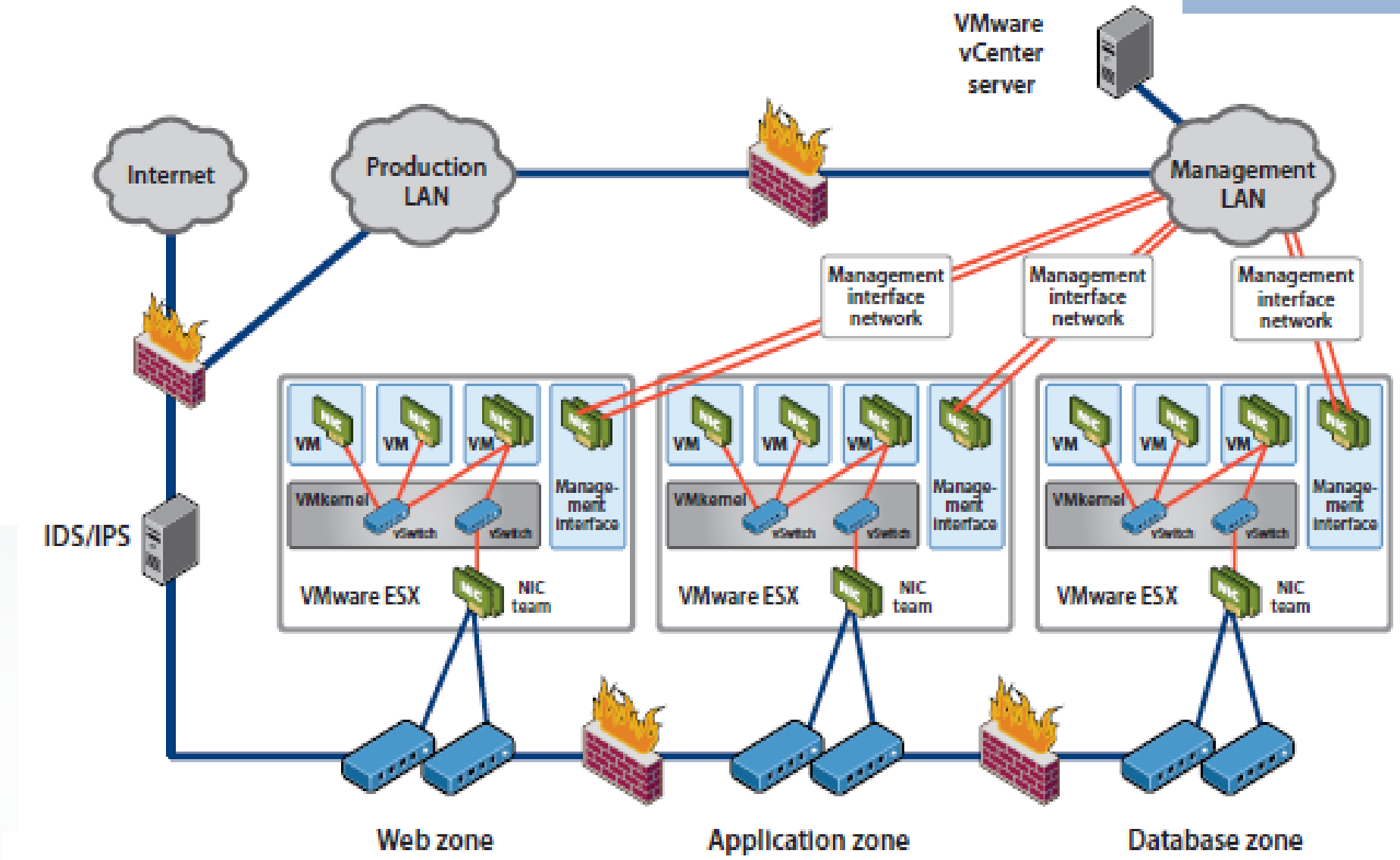


Figure 2 — Partially collapsed with separate physical trust zones

Monitoring the Solution

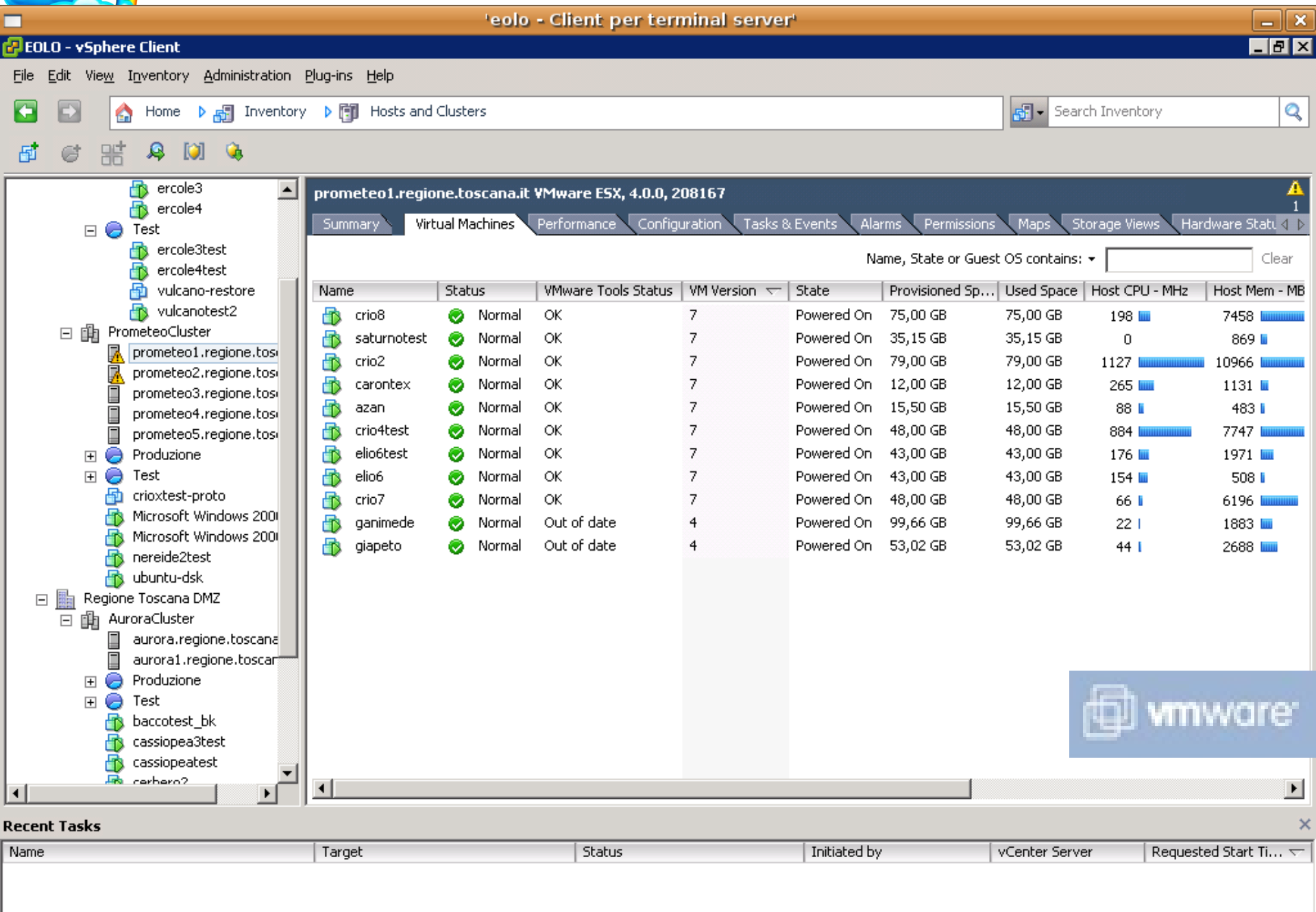
- Monitoring and assessing performance at level of:
  - Datacenter, Cluster, Host
  - Virtual Machine from outside

  - Virtual Machine inside:
    - this has to be performed by using tools inside the VM operating system
    - Windows:
      - System monitoring hosts, detailed performances
    - Linux:
      - Top or other tools

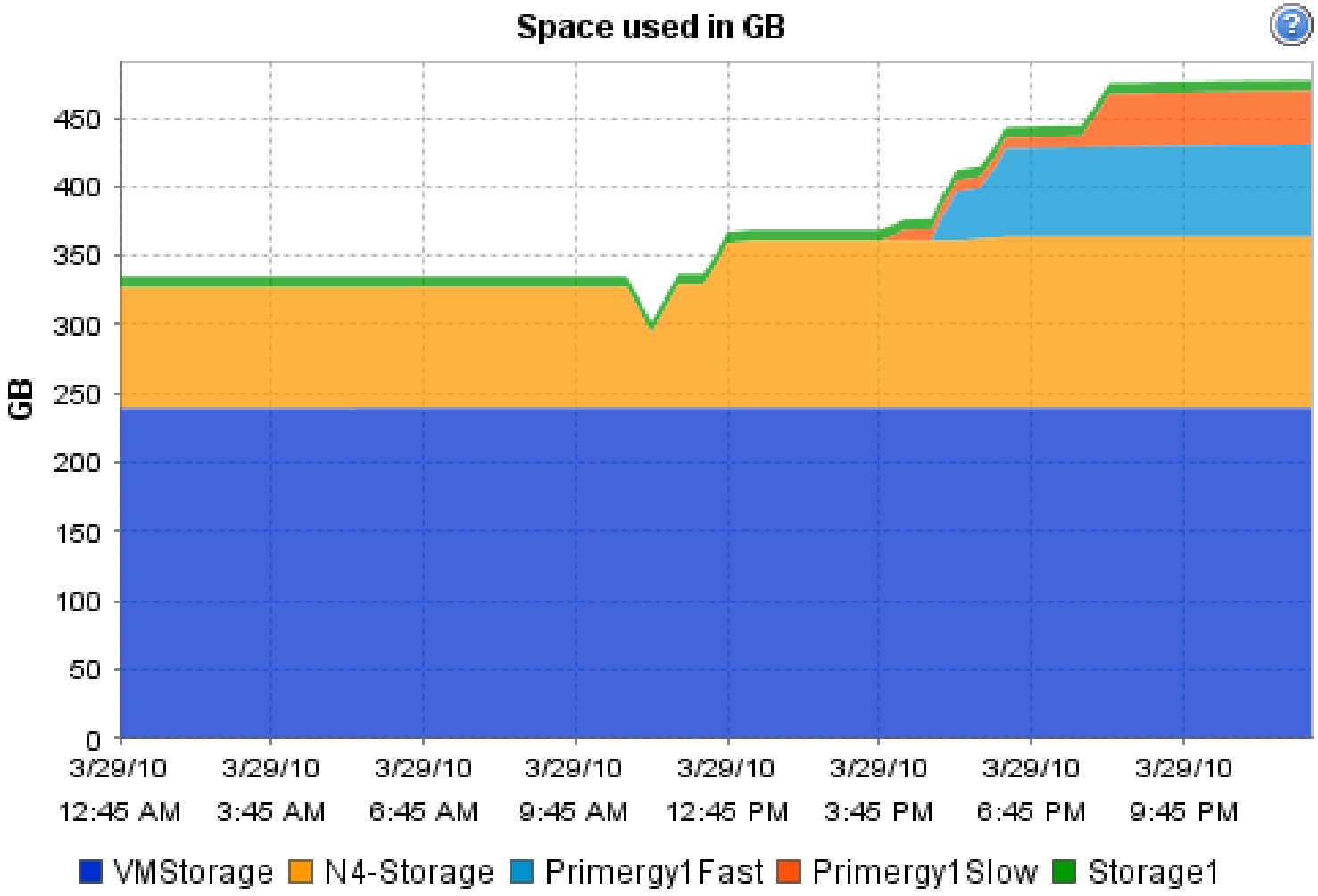
Main Monitoring methods

- Agent and Agentless
  - Problems to maintain the ceritication and assistance level
- Protocols:
  - SSH
  - API of hosts hypervisors
  - Storage protocols
  - Network monitoring protocols
- Call from applications and services to the Collector:
  - Call REST
  - Call to WebService
  - Write on database

Performance Analysis, Cluster/DC



Datacenter space consumed again time



Performance Analysis, Single VM

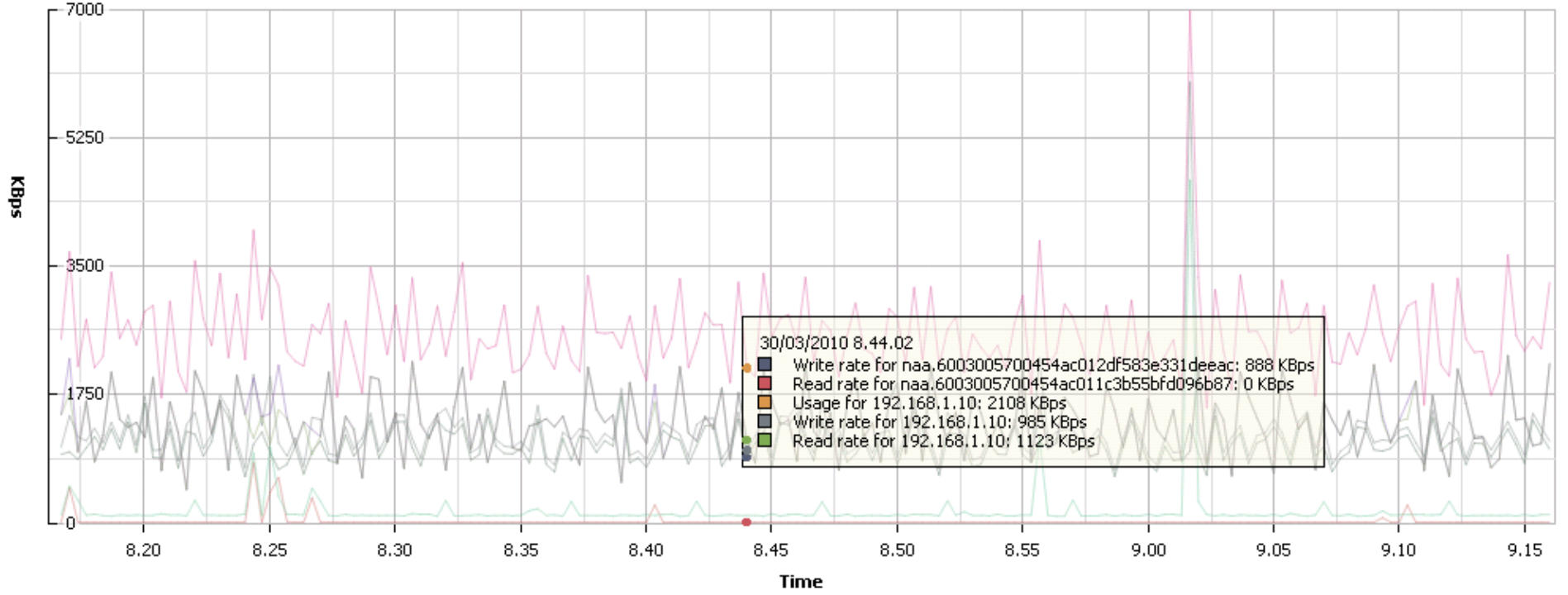
Getting Started | Summary | Virtual Machines | Resource Allocation | Performance | Configuration | Tasks & Events | Alarms | Permissions | Maps | Storage Views | Hardware Status

Overview | Advanced

Disk/Real-time, 30/03/2010 8.16.12 - 30/03/2010 9.16.12 Chart Options...

Graph refreshes every 20 seconds

Switch to: Disk



Performance Chart Legend

Key	Object	Measurement	Rollup	Units	Latest	Maximum	Minimum	Average
	naa.600300570...	Write rate	Average	KBps	1001	1727	626	1058,693
	naa.600300570...	Read rate	Average	KBps	0	820	0	18,045
	192.168.1.10	Usage	Average	KBps	3281	6992	1564	2594,006
	192.168.1.10	Write rate	Average	KBps	1110	6019	716	1218,615
	192.168.1.10	Read rate	Average	KBps	2170	2279	446	1374,872
	naa.600300570...	Read rate	Average	KBps	2170	2279	446	1356,693

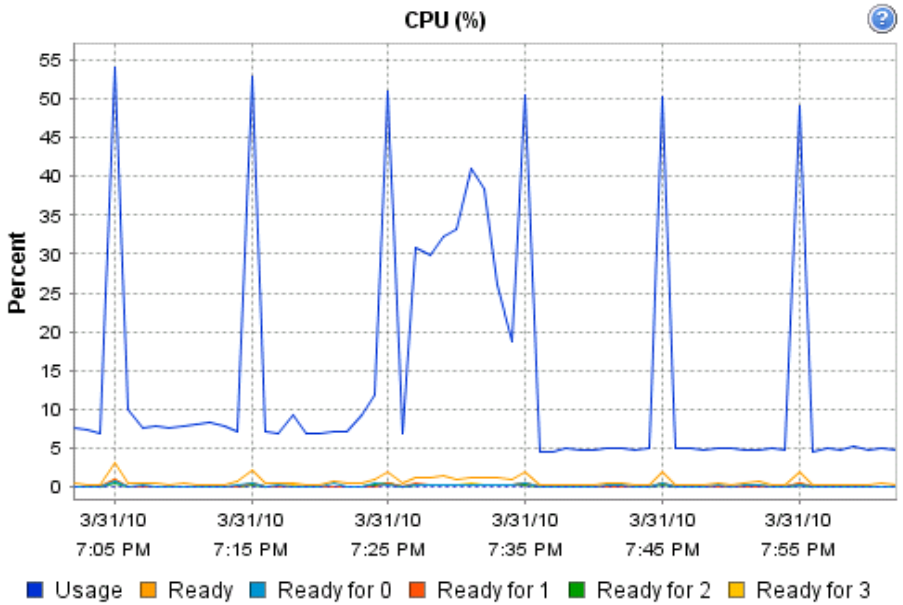
vmware

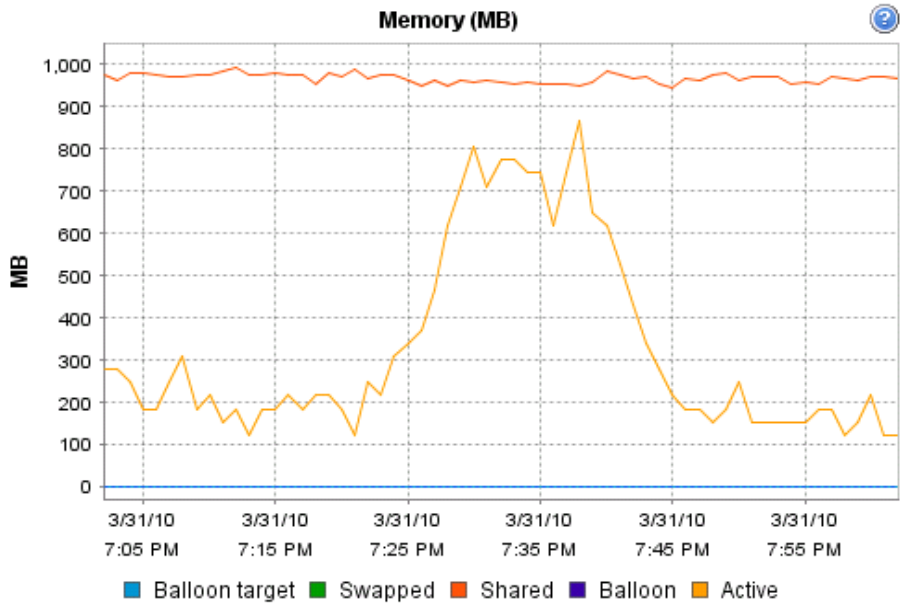
Performance Analysis, Single VM

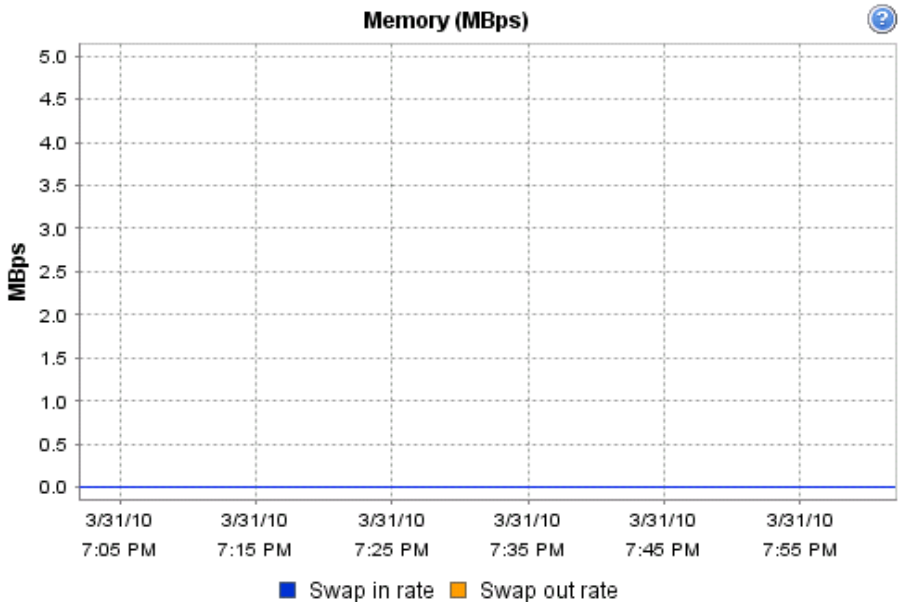
Getting Started | Summary | Resource Allocation | Performance | Tasks & Events | Alarms | Console | Permissions | Maps | Storage Views

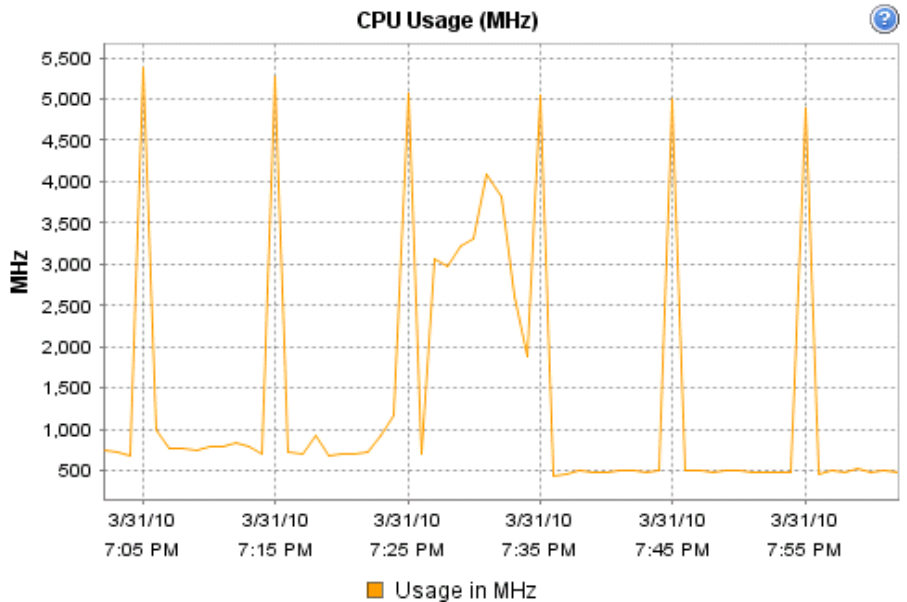
Overview | Advanced

Realtime summary for bpnet.axmedis.org

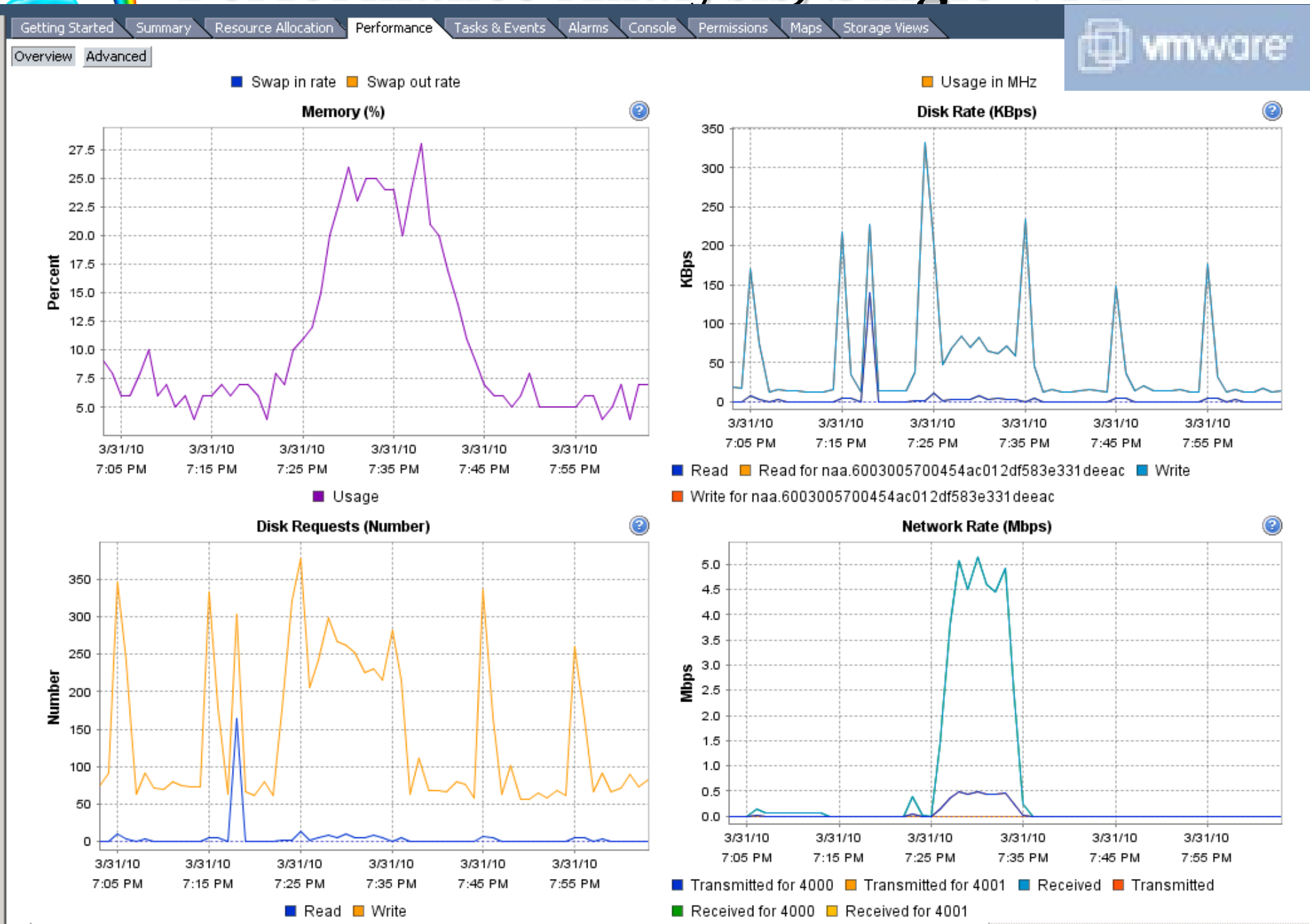




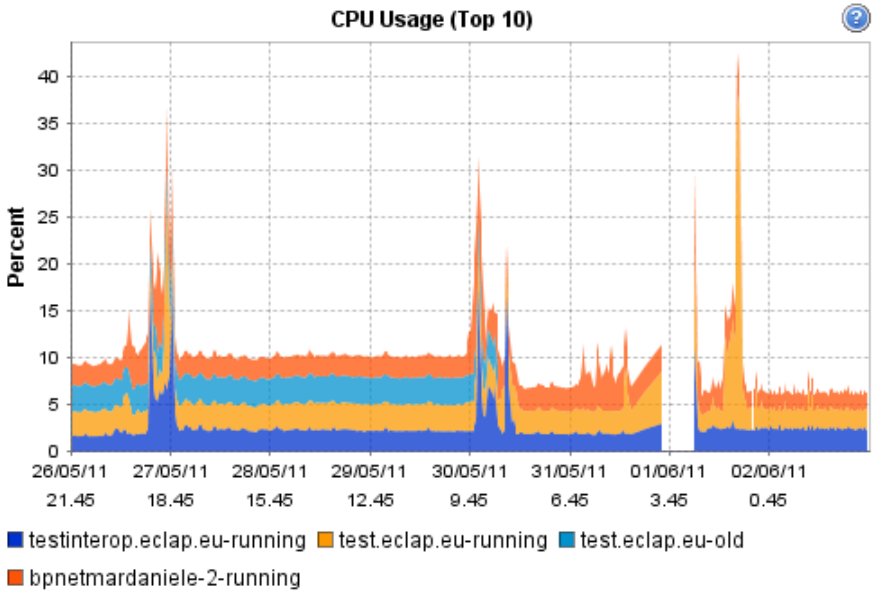


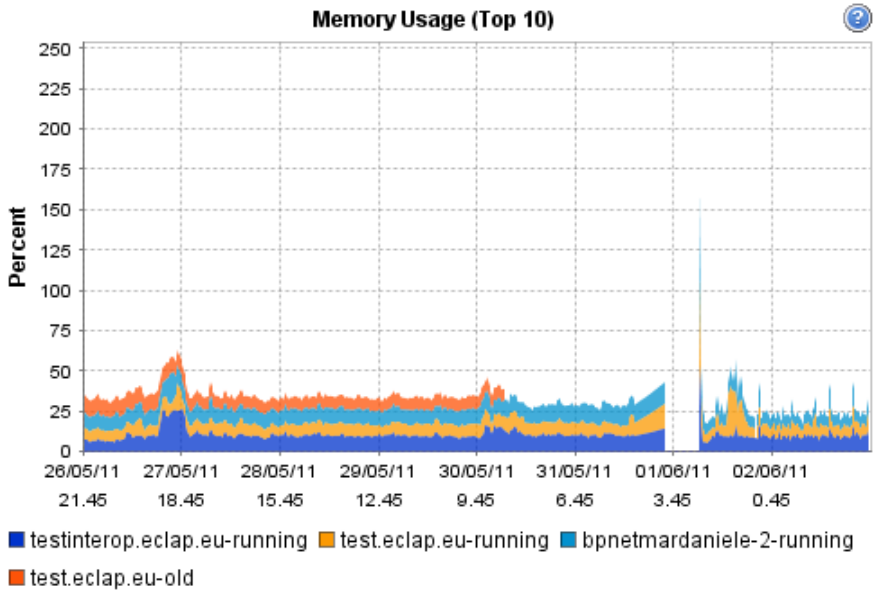


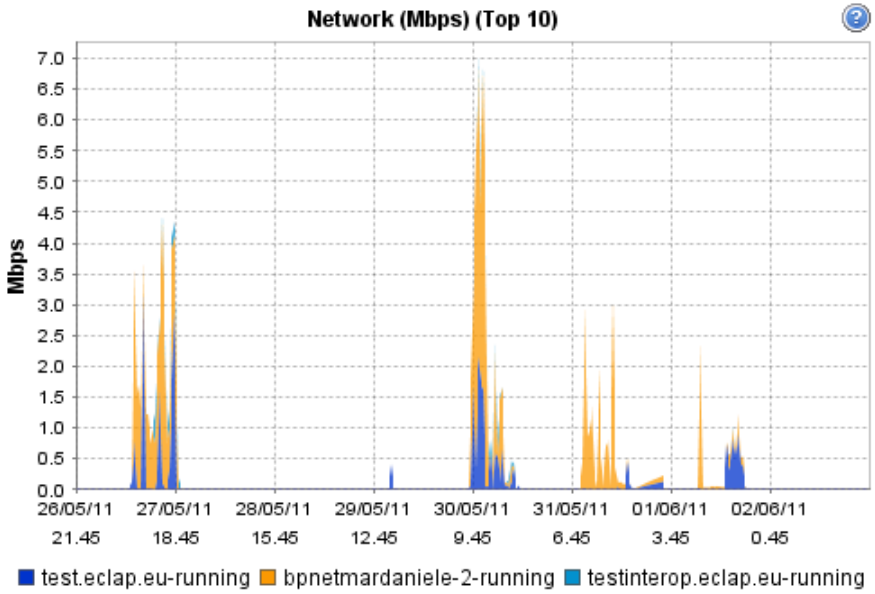
Performance Analysis, Single VM

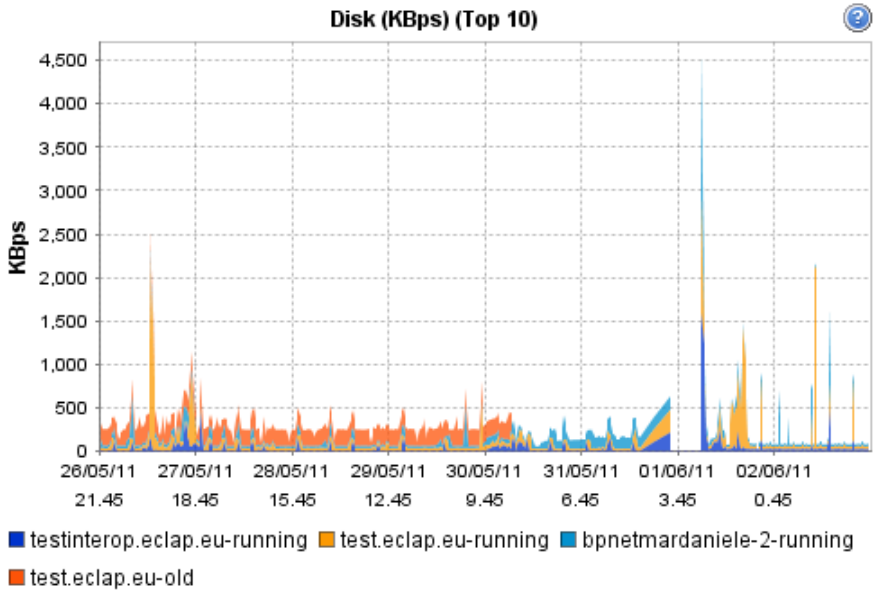


Performance Analysis of VM on the Host









Problematiche del Monitoraggio

- Parametri da monitorare: IaaS, PaaS, e SaaS:
  - Condizioni e parametri di stato degli HOST
  - Parametri di consumo degli HOST
    - Spazio storage, snapshots, etc.
  - Condizioni e parametri di stato delle VM e del sistema operativo
  - Parametri di consumo delle VM
    - Spazio disco, memoria, connessione
  - Condizioni e parametri di stato delle applicazioni e dei servizi dentro le VM
  - Parametri di consumo delle applicazioni e dei servizi dentro le VM:
    - E.g.: Numero di fatture, numero di utenti

Monitorare perche'

- Actions:
  - Allarmi gialli/rossi (75%/90%) se si superano soglie definite
  - Invio di: email, sms, etc.
  - Attivazione di riconfigurazioni, cambi di configurazione, estensioni di risorse
  - Attivazione di moving
  - Shutdown di emergenza
  - Network off/on
  - Cambio di billing in base alla SLA
  - Etc. etc.

Confronto fra sistemi di monitoring

Requisiti PaaS/SaaS	Nagios (core)	Ganglia (core)	Zenoss (core)	Zabbix 2.2.0	Hyperic HQ 4.6.6
Monitoraggio Virtual Machine (Low level Metrics)	X	X	X	X	X
Monitoraggio Applicazioni/Servizi Agent-Based	X	X	X(^^)	X	X
Monitoraggio Applicazioni/Servizi Agent-Less	X	-	X	X	X
Agent WIN OS	X	X	X(^^)	X	X
Agent/daemon Linux OS	X	X	X(^^)	X	X
IaaS					
Monitoraggio Host (Low level Metrics)	X	X	X	X	X
Monitoraggio Hypervisor (Low level Metrics)	X	X (+)	X(^)	X(§?)	X(°°)
Agent per Hypervisor	X	X (+)	X(^^)	X(§§)	X(°)
Notifiche					
Eventi e notifica allarmi	X	X	X	X	X
High Level Metrics					
Definizione HLM	X (**)	X (+++)	X(^^)	X(§)	X(°)
Calcolo HLM	X (**)	X (+++)	X(^^)	X(§)	X(°)
Storage HLM	X (**)	X (+++)	X(^^)	X	X(°)
Interoperabilità					
SDK /API accesso dati	X	X	X	X	X
Plugin	X	X	X	X(§)	X
Integrazione					
Linguaggio	C/CGI	C, Perl, PHP, Python	Python under Zope	PHP,	Java/C
DB Data Storage	MySQL (via event broker con NDOutils o Merlin)	MySQL (++)	MySQL	MySQL, Oracle, PostgreSQL, SQLite, IBM DB2	MySQL, Oracle, PostgreSQL
FS Data Storage	RRD tool	RRD tool	-	-	-
Linux OS	X	X	X	X	X
Win OS	X (***)	X	-	X	X
Licenza	GPL	BSD	GPL	GPL	GPL
Web GUI	Nagios XI (*)	Ganglia Web 2.2.0+	X	X	X

Structure

- Elements of cloud Computing
- Virtualization
- Cloud Computing
- High Availability
- vSphere Infrastructure
- Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions, PaaS Solutions, SaaS Solutions
- ICARO project

Virtualization Solutions

- **VMware vSphere**
  - Datacenter and VM management, a large range of OS
- **Microsoft Hyper-V**
  - Based on Microsoft Windows Server 2008
- **HP Integrity**
  - Datacenter and VM management, x86, linux, etc.
  - No GUI for monitoring
- **XEN**
  - Virtual machine monitor, hypervisor: Suse, RedHat, Sun Solaris, Debian
  - X86, 64, PowerPC 970
  - GPL licensing
- **others**
  - SWsoft with its Virtuozzo
  - IBM Power VM
  - VirtualBox of Sun, open source

Comparison (www.ctistrategy.com)

Aspect	VMware vSphere 4	Microsoft Hyper-V R2
Hosts		
CPUs supported	Recent AMD, Intel	Recent AMD, Intel
# CPU cores supported	64	64
Memory supported	Unlimited	2TB
I/O devices supported	IDE, SCSI, SAS, SATA, FC, 1Gb and 10Gb Ethernet, iSCSI, NFS, FCOE, Infiniband	IDE, SCSI, SAS, SATA, FC, 1Gb and 10Gb Ethernet, iSCSI, CIFS, FCOE, Infiniband
Memory optimization	Over-commit, transparent page sharing, ballooning, large memory pages	Standard Windows Server 2008 features
Platform support	Fewer vendors	More vendors
Supported storage of guest VMs	Direct, SAN, NAS, iSCSI	Direct, SAN, iSCSI
Number of nodes in a cluster	32 nodes if < 40 VMs per node	16

Comparison, 2

Aspect	VMware vSphere 4	Microsoft Hyper-V R2
Guest		
Operating systems supported	Asainux, CentOS, Debian, FreeBSD, OS/2, Solaris 10, SCO OpenServer, SCO Unixware, Windows Server, RHEL, SUSE, MS-DOS, Netware	Windows Server, Vista, XP, SUSE Linux
Operating systems tools provided (per OS)	Yes, for most guests	Yes, for most guests
# virtual CPUs supported	8	4
# guests per host	256 running	512 (192 running)
# Amount virtual memory	255GB	64GB
Virtual NICs	10	Yes, limit unknown
# of snapshots	32 per VM (?)	50 per VM
Types of guests supported	32-bit, 64-bit, simultaneously	32-bit, 64-bit, simultaneously
Ability to hot-add disk images and external storage	Yes	Virtual SCSI devices only, not IDE

Aspect	VMware vSphere 4	Microsoft Hyper-V R2
Features		
VM move	Live	Live
Direct I/O	VMDirectPath I/O	-
VM synchronization	With limits (1 vCPU, many features disabled)	No
Directly boot from VM image	Only if ESXi installed	Yes
P to V	Included	Included
V to P	Included	Included
H/A via clustering and failover	Yes	Yes
Replication	Integration with 3rd party storage products	Yes (DFS-R)
Performance monitoring	Yes, vCenter Server	Yes, SC Operations Manager
Network features	Virtual switch, VLAN tagging, Network vMotion, Network traffic shaper, IPv6, CDP, NIC teaming	Standard Windows Server 2008 features
Storage features	Thin provisioning, consumption-based monitoring, reports and topology maps, LUN discovery, adaptive block sizing, storage vMotion	Standard Windows Server 2008 features
Patching of guests	vCenter Update Manager (both running and halted guests, Windows and some Unix)	Standard Windows Server 2008 features for booted Windows guests, Offline Machine Servicing Tool for halted Windows guests
Security	Layer 2 security policies, vShield, VMsafe 3rd party security products	Native firewall, 3rd party security products
Backups	Native via VMware Data Recovery, Support from major vendors	Native, Support from major vendors
Resource management	Yes, many options	Yes, some options
Physical server power on / off as needed	Via VMware DRS, DPM	No

Standards in Virtual Machine formats

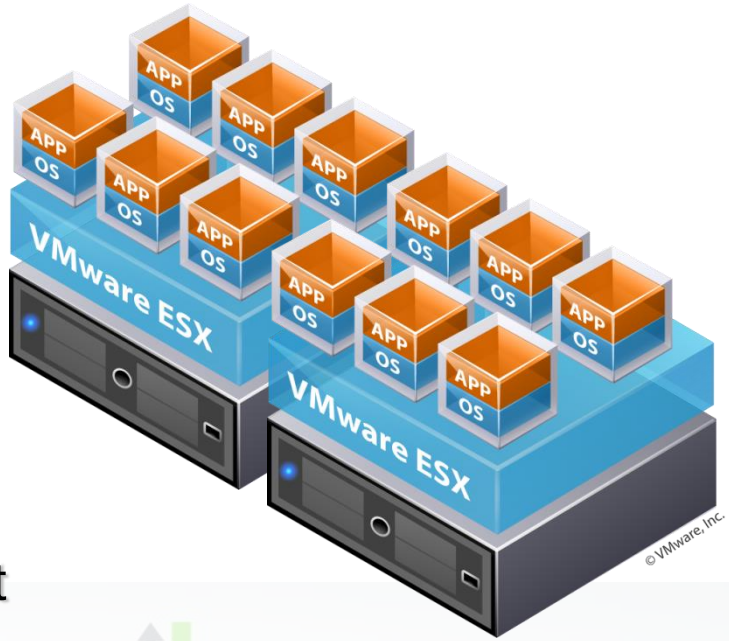
- Virtual Machine formats on the HD
- **OVF:** Open Virtualization Format, on the push of VMware
  - Format for VM
- VM disk format of VMware is a standard which is supported by:
  - VMware workstation
  - vSphere VMware
  - VirtualBox of SUN

Structure

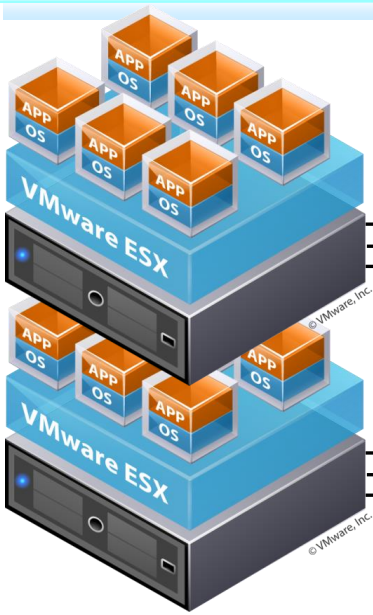
- Elements of cloud Computing
- Virtualization
- Cloud Computing
- High Availability
- vSphere Infrastructure
- Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions, PaaS Solutions, SaaS Solutions
- ICARO project

How to WORK with VMs in the Cloud

- **KVM solutions**
  - Local access via local KVM
  - Local server access via HTTP
- **Windowing Terminal**
  - MS Windows Remote Desktop
  - X Terminal to linux
- **Remote Solutions**:
  - VNC, Radmin, etc.
  - VNC: Also possible via HTTP port
- **Telnet, char based consol, SSH**
  - VT100 terminal for example



KVM



Keyboard, Video, Mouse

Keyboard, Video, Mouse

kvm

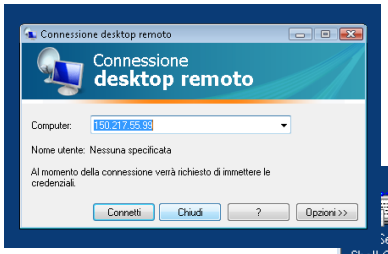
Network:

-- control

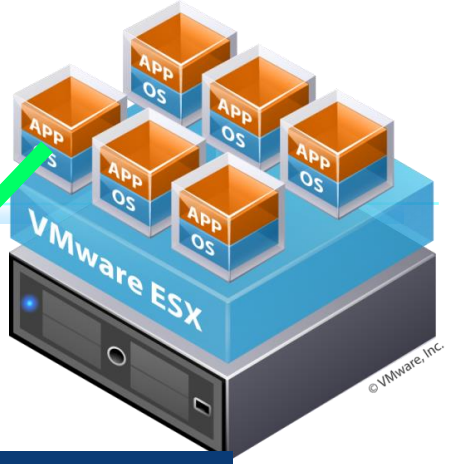
-- remote access/desktop

RJ45 KVM

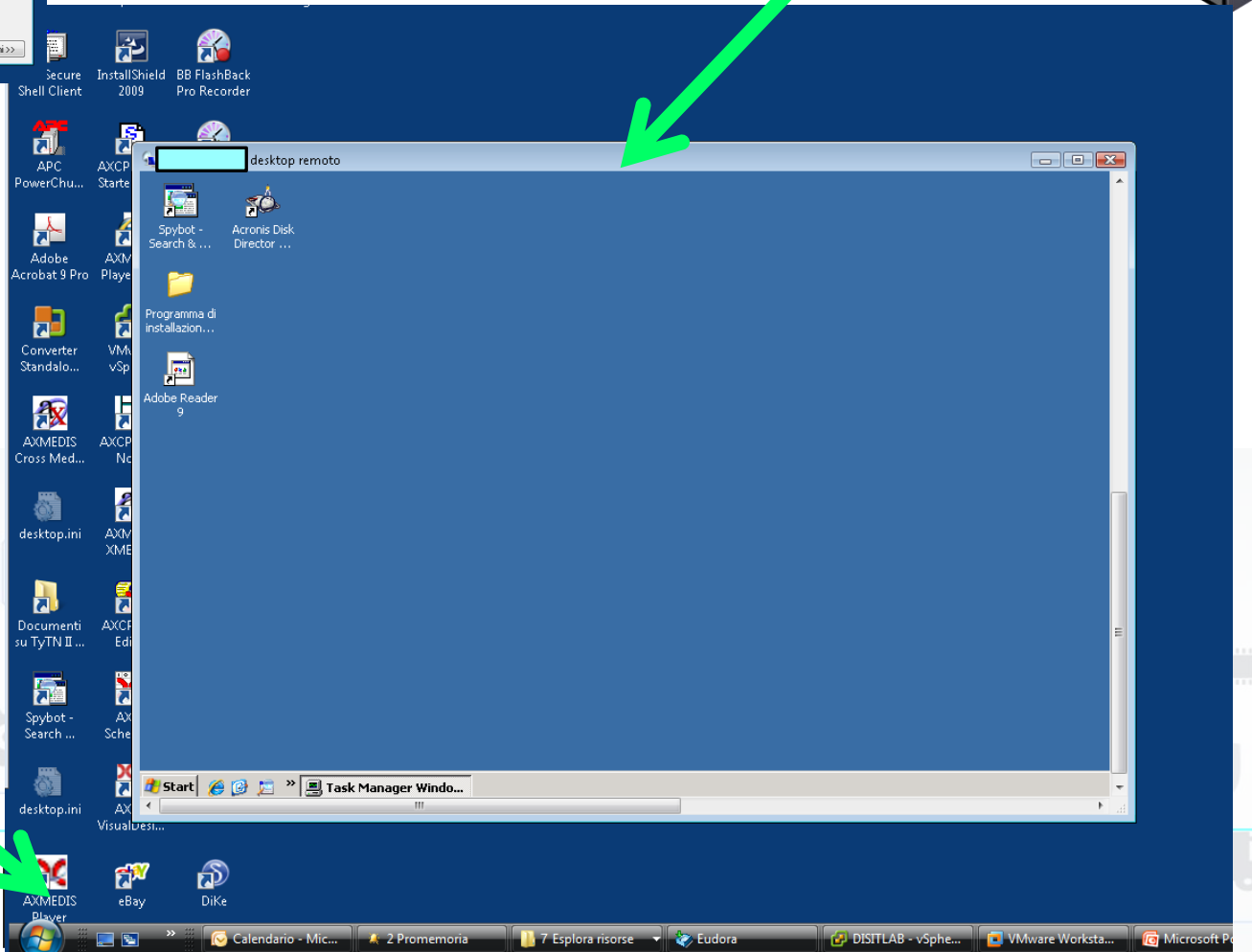
Windows Remote Desktop



- *Remote machine*
- *Windows 2003*

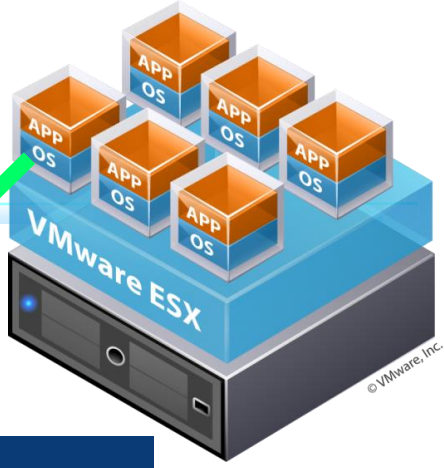


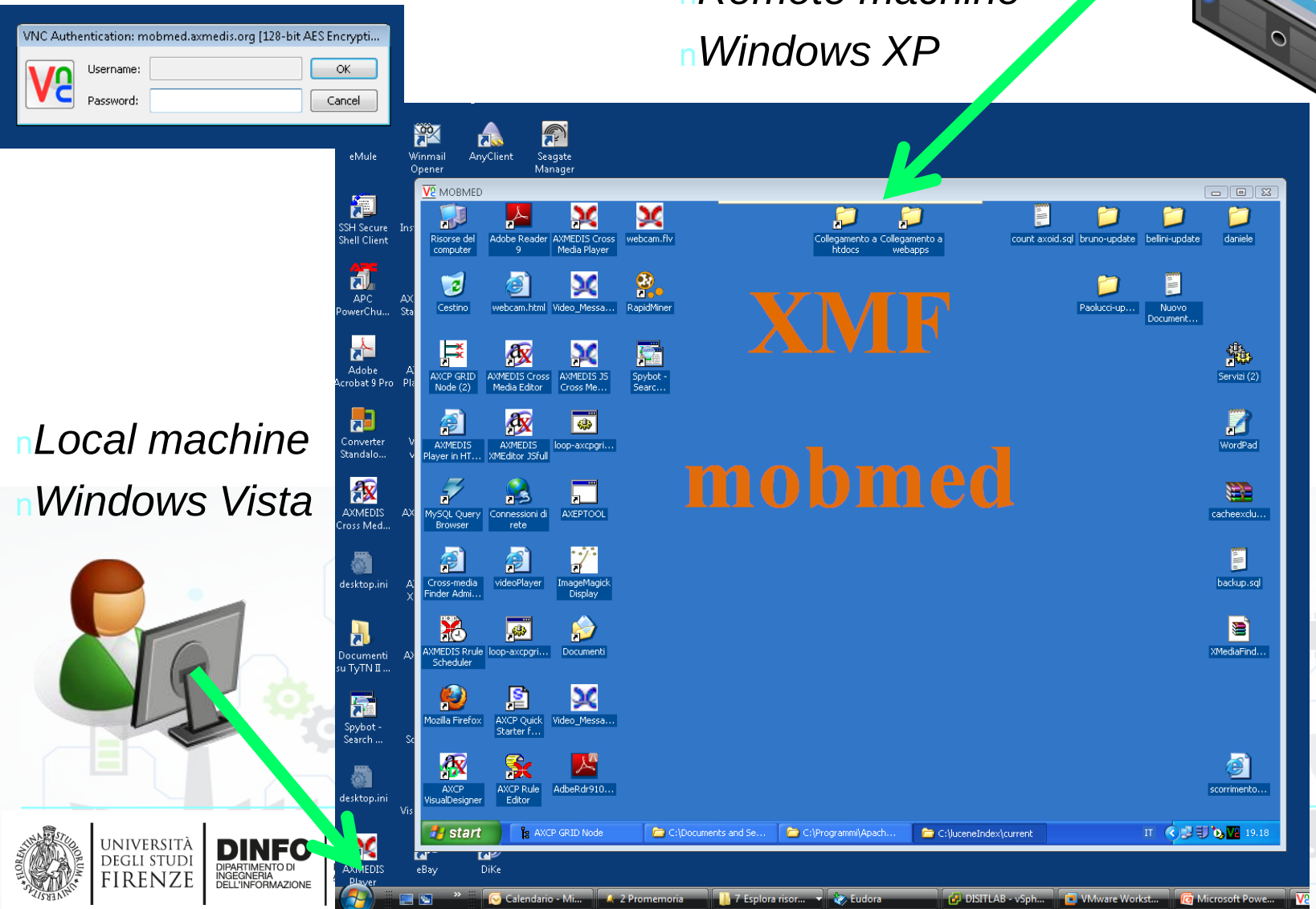
- *Local machine*
- *Windows Vista*



VNC remote connection

- *Remote machine*
- *Windows XP*



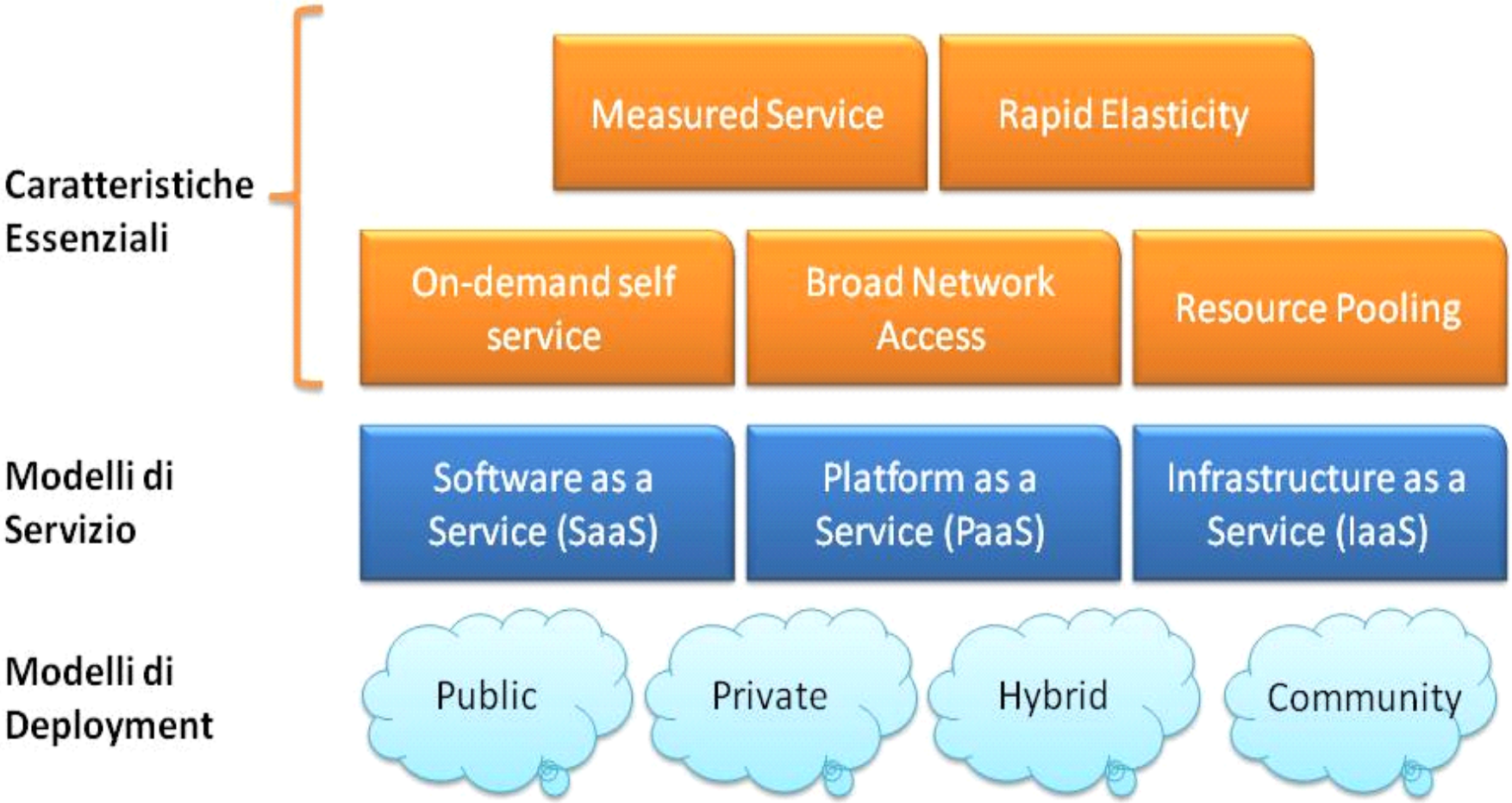


- *Local machine*
- *Windows Vista*

Structure

- Elements of cloud Computing
- Virtualization
- Cloud Computing
- High Availability
- vSphere Infrastructure, Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions
- PaaS Solutions
- SaaS Solutions
- ICARO project

Modello Generale



Infrastructure as a Service (IaaS)

- erogazione di servizi infrastrutturali relativi a:
  - CPU come clock, storage as HD, rete, mem
  - altri elementi di base assolutamente indipendenti da servizi applicativi di qualunque tipo.
- infrastruttura messa a disposizione dal provider per eseguire la propria applicazione:
  - Affitto di Server e/o macchine virtuali
  - pagamento in base al consumo dell'infrastruttura
  - lascia sotto la responsabilità dell'utente la gestione del sistema operativo, dell'eventuale middleware e della parte di runtime, oltre che dell'applicazione stessa.
- Amazon EC2 è un esempio di servizio IaaS.

IaaS

- Una serie di Host, VM e NAS/SAN
- Un gestore di macchine virtuali
- Un sistema per il deploy basato su templates
- Un sistema di monitoring and alarm su risorse fisiche e parametri di hosts
- Un sistema per l'accounting e il deploy automatico
- Alcuni servizi per l'accesso diretto a Host e VM

IaaS Solutions

- Based on Virtualization
- A set of Datacenters, Hosts, NAS/SAN, networks, etc.
- IaaS commerciali come
  - Amazon's Elastic Compute Cloud (EC2) and Simple Storage Service (S3). Amazon EC2 è il leader del mercato dei provider IasS.
- Come Technology provider di soluzioni IaaS si hanno:
  - Oracle,
  - VMware: Vcenter, Vsphere, etc.;
  - **Nimbus**;
  - OpenNebula;
  - HP;
  - **Eucalyptus; Ubuntu** Enterprise Cloud (basato su Eucalyptus);
  - GoGrid;
  - Flexiscale;
  - UNISYS,
  - Enterprise Cloud Manager, ECM, on Hybrid.

Service Model	IaaS														
Infrastructure / Feature	Amazon EC2	BitRefinery	GoDaddy	GoGrid	Hosting.com	NephoScale	OpSource	Rackspace	ReliaCloud	Softlayer	Terremark	Eucalyptus	OpenNebula	Nimbus	
Pay As You Go	x		x	x		x	x	x		x	x				
Dynamic Service Level Agreement															
Certificazioni (e.g., PCI o SAS 70)	x	x			x		x	x	x	x	x				
Scale Up		x	x	x	x	x	x	x			x				
Scale Out	x	x	x	x	x	x	x	x	x	x	x	x	x	x	
Live Support		x	x	x	x	x	x	x	x						
Monitoring Tools	x		x		x		x	x		x			x		
APIs	x			x		x	x	x	x	x	x	x	x	x	
Free Tier	x					x						x	x		
Highly customizable istances		x			x		x				x				
Cloud Burst													x		

IaaS confronto (legenda)

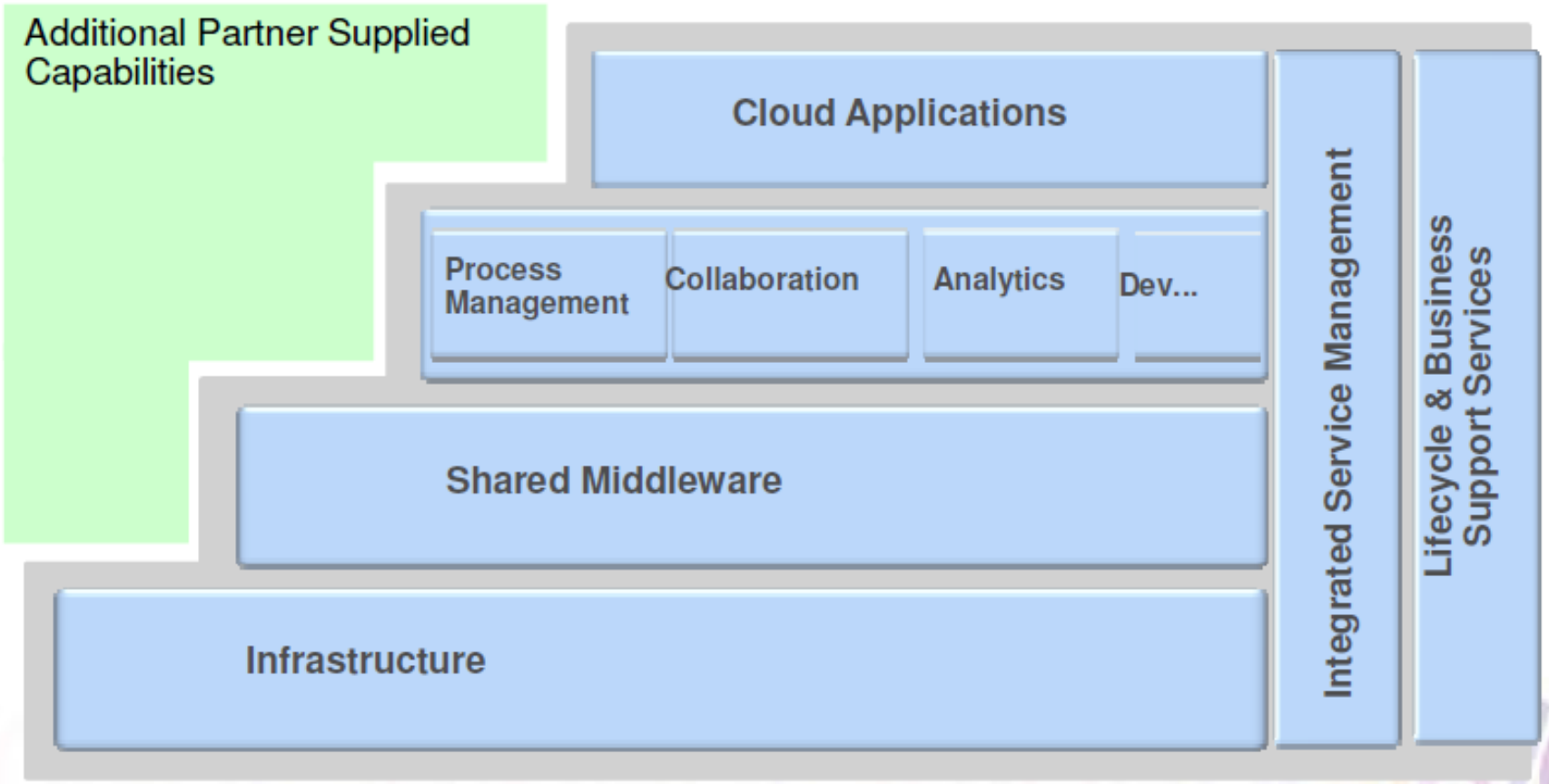
- **Pay per use** – se si paga a consumo, in realtà molti provider hanno una filigrana di offerte più ampia, includendo anche piani mensili, sconti e promozioni ecc.
- **Dynamic SLA** – se viene offerta la possibilità di ridefinire gli SLA.
- **Certifications** – se il provider offer certificazioni sulla compliance/sicurezza come PCI o SAS 70.
- **Scale Up** – se è possibile lo scale up di single istanze di server, tramite l'aggiunta di memoria, extra CPU o storage.
- **Scale Out** – se è possibile fare il deploy veloce di nuove istanze dei server.
- **Live Support** – può essere diviso in:
  - *Poor (n)* – solo forum di supporto for free; in alternativa a pagamento.
  - *Average (y)* – supporto 24×7 gratis (telefono, chat, forum).
  - *Extensive (y)* – offerte di supporto multiplo per ogni soluzione proposta.
- **Monitoring** – può essere diviso in:
  - *Poor (n)* – nessuna soluzione di monitoring/alert integrata, sono necessari strumenti di terze parti da acquistare separatamente.
  - *Average (y)* – strumenti di monitoring minimali e senza servizi di alert.
  - *Extensive (y)* – soluzioni complete e integrate di strumenti di monitoring compresi nel prezzo.
- **APIs** – se vi sono API per interagire con i server.
- **Free Tier** – se vengono offerte soluzioni di prova per il test dei servizi

Structure

- Elements of cloud Computing
- Virtualization
- Cloud Computing
- High Availability
- vSphere Infrastructure, Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions
- PaaS Solutions
- SaaS Solutions
- ICARO project

Platform as a Service (PaaS)

- erogazione di servizi applicativi di base come:
  - sistemi operativi, middleware, linguaggi,
  - tecnologie di base dati
  - ambiente runtime necessari per eseguire l'applicazione,
- rimane sotto la responsabilità dell'utente,
  - applicazione
  - la definizione del modello (e.g., numero e dimensione dei server, datacenter, caratteristiche del networking) da utilizzare per l'esecuzione dell'applicazione.
- Google AppEngine è un esempio di Platform as a Service.
- A livello PaaS viene anche collocato l'insieme dei servizi MaaS, Middleware as a Service.

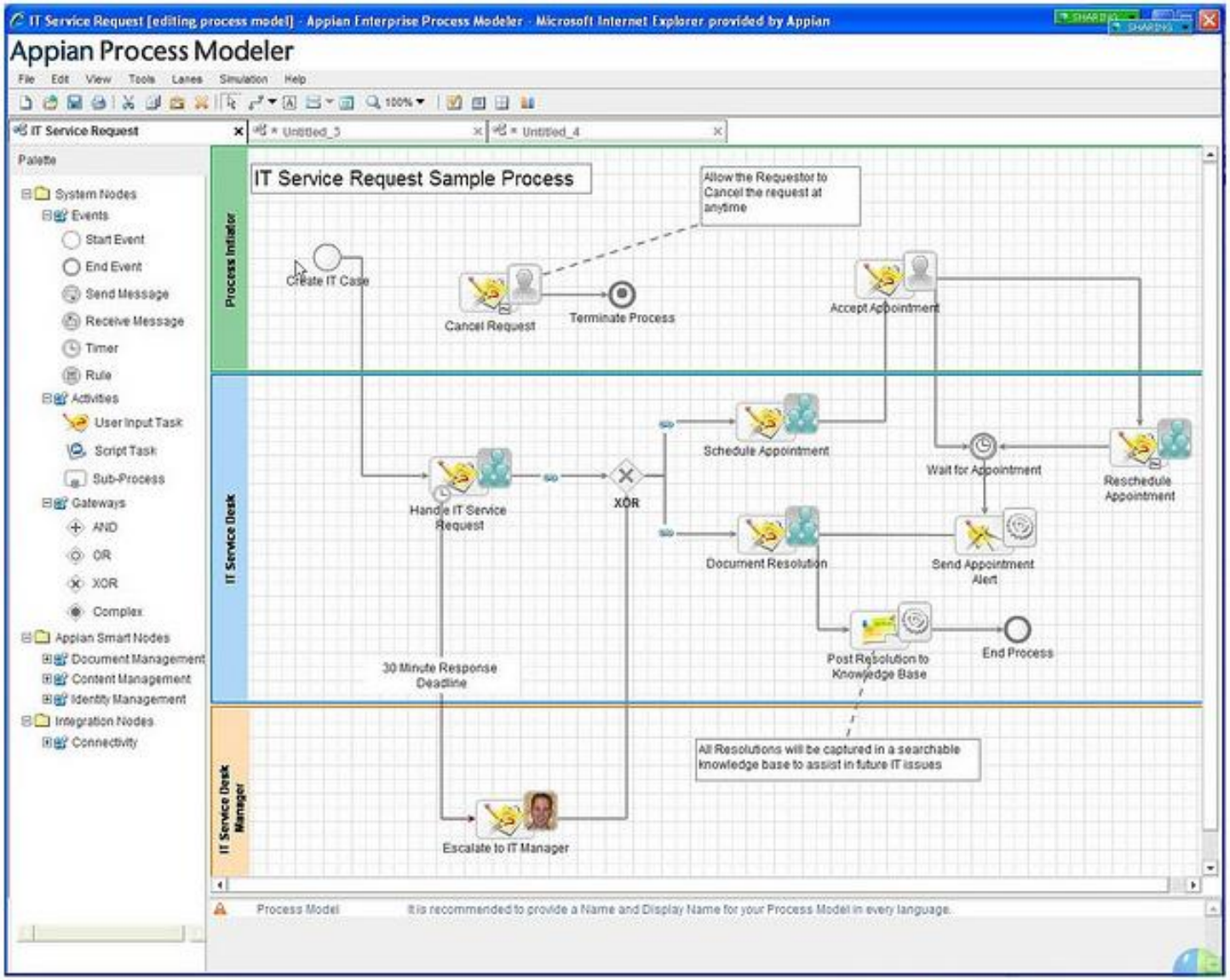

Additional Partner Supplied Capabilities
Cloud Applications
Process Management
Collaboration
Analytics
Dev...
Shared Middleware
Infrastructure
Integrated Service Management
Lifecycle & Business Support Services

PaaS

- **Middleware per accesso ai servizi di base**
  - **Middleware as a Service, MaaS**
  - **API per accesso a questi servizi**
- **Sistema di sviluppo** per applicazioni che possono usare le API
- **Servizi e risorse base**

- **Business Process Modeling:**
  - **permette definire come allocare e configurare le macchine virtuali ed i servizi sul cloud**.
  - A livello SaaS: Crea procedure, script, etc. per il deploy e la configurazione di VM, Applicazioni, servizi, etc. anche riconfigurazioni e trasformazioni
- **Processo di lavoro**

Appliant Process Modeler



Alcuni esempi di PaaS

- Google App Engine (https://appengine.google.com/) detto anche GAE
- Windows Azure Platform (http://www.microsoft.com/windowsazure/)
- Force (http://www.salesforce.com/platform/)
- Oracle Fusion Middleware (http://www.oracle.com/it/products/middleware/index.html)
- Eccentex AppBase (http://www.eccentex.com/platform/features.html)
- 3Tera AppLogic (http://www.ca.com/us/cloud-platform.aspx)

Service Model	Platform Feature	Tecnologie	Cloud hosting	Content Delivery Network (CDN)	Repositories	Databases	APIs	Dashboard/Console centralizzata
PaaS	SalesForce Force.com	Java, .NET, Ruby, PHP, Python, Perl, Adobe Flex e AIR				SOQL, SOSL	REST	
	Red Hat OpenShift	Java, Java EE, Python, Perl, PHP, Ruby	Multi IaaS Provider (AWS, etc.)		Git, SSH, rsync	MySQL, MongoDB, MemBase, Memcache	REST	x
	Rackspace Cloud Sites	PHP, ASP, .NET	Multi-tenant PaaS	Rackspace Cloud Files/Akamai	FTP/SFTP	MySQL 5 & MS SQL Server 2008	object storage/ CDN	x
	Microsoft Windows Azure	C#, Java, PHP, Ruby					REST	
	Jelastic	Java			Maven, Git, SVN	Maria DB, MySQL, PostgreSQL, MongoDB, CouchDB		x
	Heroku	Ruby				Amazon RDS (MySQL), MongoDB, Redis, CouchDB		
	Google App Engine	Java, Python				Google Cloud SQL	JDBC, DB-API	
	CumuLogic	Java, Spring	Amazon EC2, OpenStack, CloudStack, Eucalyptus, VMware vSphere		Git, SVN	MySQL, MySQL cluster, MongoDB	REST	x
	Cloudify	Java, .NET, Groovy, Ruby, C++, Node.JS, Spring (more soon)	deploy su qualsiasi IaaS e BYON (bring-your-own-nodes)		Git	Cassandra, MongoDB, MySQL, HSQL	CLI, REST, Web	x
	CloudFoundry	Java Spring, Groovy Grails, Ruby Rails Sinatra, Node.js			Git	MongoDB, MySQL, Redis	CLI	
	CloudBees	Java			Maven, Git, SVN	MySQL	REST, CLI	x
	AWS Elastic Beanstalk	Java	AWS Cloud					

Structure

- Elements of cloud Computing
- Virtualization
- Cloud Computing
- High Availability
- vSphere Infrastructure, Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions
- PaaS Solutions
- SaaS Solutions
- ICARO project

Software as a Service (SaaS)

- **erogazione di servizi** applicativi accessibili indipendentemente dalla collocazione e dal tipo di device utilizzato.
- **Non sono eseguite** applicazioni del cliente
  - il cliente paga il diritto (mediante licenza o canone di affitto) di utilizzo di un'applicazione messa a disposizione dal provider, senza preoccuparsi di come essa venga realizzata e gestita nel cloud.
- Il cliente deve scegliere la corretta applicazione che soddisfi le sue necessità
- SalesForce.com Customer Relationship Management (CRM) è un esempio di soluzione in cui il software è venduto in modalità as a service.

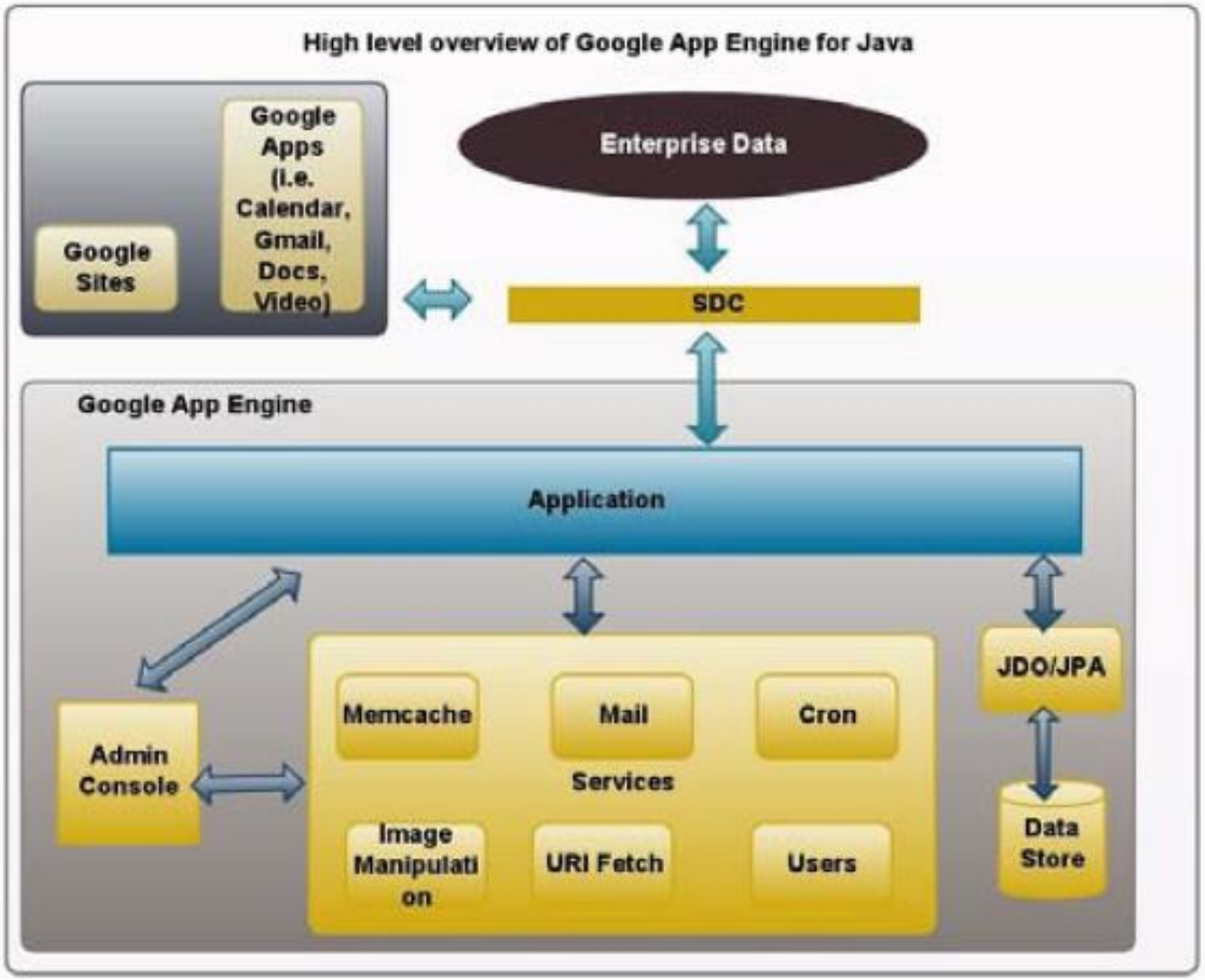
Architetture SaaS

- **Interfacce esterne**
- **L'insieme di API,** rappresentano i metodi con cui le applicazioni possono venir relazionate con altri servizi o software,
  - Cloud support for mobile, come Apple
  - Push service come Apple
- **Middleware** è lo strumento che permette all'applicazione di sfruttare tutti i servizi e le risorse di livello più basso
- **Ci puo' essere un Business Producer** per effettuare il deploy automatizzato di Software a servizio.
  - Tali soluzioni devono tenere conto dello sviluppo dei programmatori e delle SLA (con i corrispondenti parametri di consumo da monitorare).

Soluzioni SaaS

- **Google Apps** (Google) **-** Google offre gratuitamente ai privati i propri servizi applicativi come:
  - Google Documents, GMail, Calendars, Sites.
- **Oracle CRM** On Demand in grado di supportare tipologie di offerta sia in multi tenancy sia in single tenancy
- **SalesForce.com** (Salesforce) **-** Customer Relationship Manager.
- **Zoho Office** (Zoho) **-** Propone un insieme di applicazioni mirate alle aziende. Il portfolio di servizi prevede più di 20 applicazioni tra cui ad esempio posta elettronica, office app CRM
- **IBM LotusLive** (IBM) **-** Consiste di una raccolta di soluzioni di collaborazione aziendale online e servizi di social networking.

GAP: Google App Engine



GAP: Google App Engine

- **Datastore**: un database non relazionale distribuito e scalabile, accessibile attraverso un set di API e gestibile attraverso il pannello di controllo dell'applicazione. Il servizio di basa su BigTable, un sistema proprietario per l'archiviazione di dati strutturati.
- **Google accounts:** un set di API che permette l'autenticazione di utenti attraverso account Google.
- **URL Fetch:** API che permette di ricavare risorse sul web.
- **Mail**: API che permette l'invio di e-mail.
- **Memcache**: sistema di caching di tipo chiave-valore. I contenuti di tale cache sono condivisi tra le varie istanze in esecuzione dell'applicazione.
- **Image manipulation:** API utilizzabile per la manipolazione di immagini.
- **Tasks**: Lo sviluppatore ha inoltre la possibilità di schedulare task secondo determinati orari o utilizzare una coda di task per permettere alla propria applicazione di eseguire delle operazioni di background mentre delle richieste web vengono servite.

Service Model	SaaS					
Feature / Software	Aplicor	NetSuite	Microsoft Dynamics	RightNow	Salesforce.com	
Campaign Management	x	x	x	x	x	
Lead management	x	x	x	x	x	
Event Management	x			x		
Pay Per Click (PPC) integration	x	x			x	
E-mail con trackable link e click-through	x	x	x	x	x	
Account Management	x	x	x	x	x	
Contact Management	x	x	x	x	x	
Opportunity Management	x	x	x	x	x	
Competitive Intelligence	x					
Sales Analytics	x	x	x	x	x	
Data Deduplication		x				
Ticket/case/incident management	x	x	x	x	x	
Email to case creation	x			x	x	
Routing	x	x	x	x	x	
Escalation	x	x	x	x	x	
Customer Surveys	x	x		x	x	
Knowledge base	x	x		x		
Self Service Portal	x	x	x	x	x	
Dashboard	x	x	x	x	x	
Ad Hoc Report Writer	x	x	x	x	x	
Data Warehouses	x				x	
OLAP	x		x		x	
Groupware Integration	x	x	x	x	x	
Partner relationship management	x	x		x		
E-commerce Suite		x				
Sales Order Processing	x	x				
Workflow Designer	x			x		
Offline Edition	x	x			x	
Enterprise Resource Planning	x	x	x			
Certifications	ISO, NIST	SAS70			SAS70	
SLA with Guarantee	x					

Structure

- Elements of cloud Computing
- Virtualization
- Cloud Computing
- High Availability
- vSphere Infrastructure, Security on the Cloud
- Conversions among VM and physical machines
- vCenter, datacenters and cluster management
- Comparison among virtual computing solutions
- How to work with Virtual Machines
- IaaS solutions
- PaaS Solutions
- SaaS Solutions
- ICARO project

iCaro

x Cloud Service Provider

Paolo Nesi (UNIFI, DISIT Lab)

Feb 2015

Contesto IaaS

- **IaaS, Infrastructure as a Service:**
  - **Business:** vendita di host a consumo
  - **Gestione:** limitata al parco degli Host → vari *Gestori*
  - **Monitoraggio** delle risorse → varie soluzioni di monitoraggio per gli utenti finali e per l'amministratore,
    - spesso integrate con il Gestore
    - configurazione di monitoraggio automatizzabile
    - Dati di monitoraggio accessibili via API del Gestore
  - **Costi di gestione:**
    - accettabili con una buona automazione se mono datacenter con Gestore monomarca, etc.. Altrimenti I costi salgono…
    - Difficili da gestire se vi sono aspetti di contrattuali stringenti da tenere sotto controllo

Contesto PaaS

- **PaaS, Platform as a Service:**
  - **Business:** vendita di VM a consumo ..
  - **Gestione** di template con sistemi operativi → vari Gestori
  - **Monitoraggio** delle risorse → varie soluzioni di monitoraggio,
    - spesso non molto connesse con il vostro cliente finale
    - Difficile automazione della soluzione di monitoraggio, ma possibile
    - Dati di monitoraggio accessibili via API delle piattaforme o tramite il sistema di monitoraggio
  - **Costi di gestione:**
    - **accettabili** con una discreta automazione se mono datacenter e gestore monomarca, etc..
    - Non trascurabili quando vi sono aspetti di contrattuali stringenti da tenere sotto controllo

Contesto SaaS, 1/2

- **SaaS: Software as a Service**
  - **Business:**
    - vendita a consumo di soluzioni software di terzi:
      - tipicamente software con licenze standard, mensili o annuali
      - a consumo significa anche con licenze specifiche: numero di utenti, numero di fatture, etc.
    - COME: Applicazioni singola VM FINO A configurazioni Multi-tier complesse
      - Le semplici applicazioni a singola VM sono vendibili con svariate configurazioni, l'uso di template è molto limitato
      - Le configurazioni complesse possono avere decine di servizi, connessioni virtuali, aspetti contrattuali complessi, spesso customizzazioni rilevanti, etc....
      - Le configurazioni two tier, sono in numero elevato e costo molto in configurazione
  - **Gestori:** in questo caso sono soluzioni che
    - **Aiutano** a predisporre le applicazioni in modo che possano essere gestite in modo semplice: questi vanno da quelli che fanno tutto a quelli che definiscono linee guida complesse che solo tramite una programmazione spinta permettono di automatizzare il processo, anche tramite vari step a mano
    - **Allocano** (deploy) e configurano le applicazioni software in modo automatico nel cloud in base a varie pattern. Questi gestori permettono di definire delle procedure di deploy (workflow) anche con parametri, in alcuni casi hanno un sistema di monitoraggio integrato anche se non sofisticato.

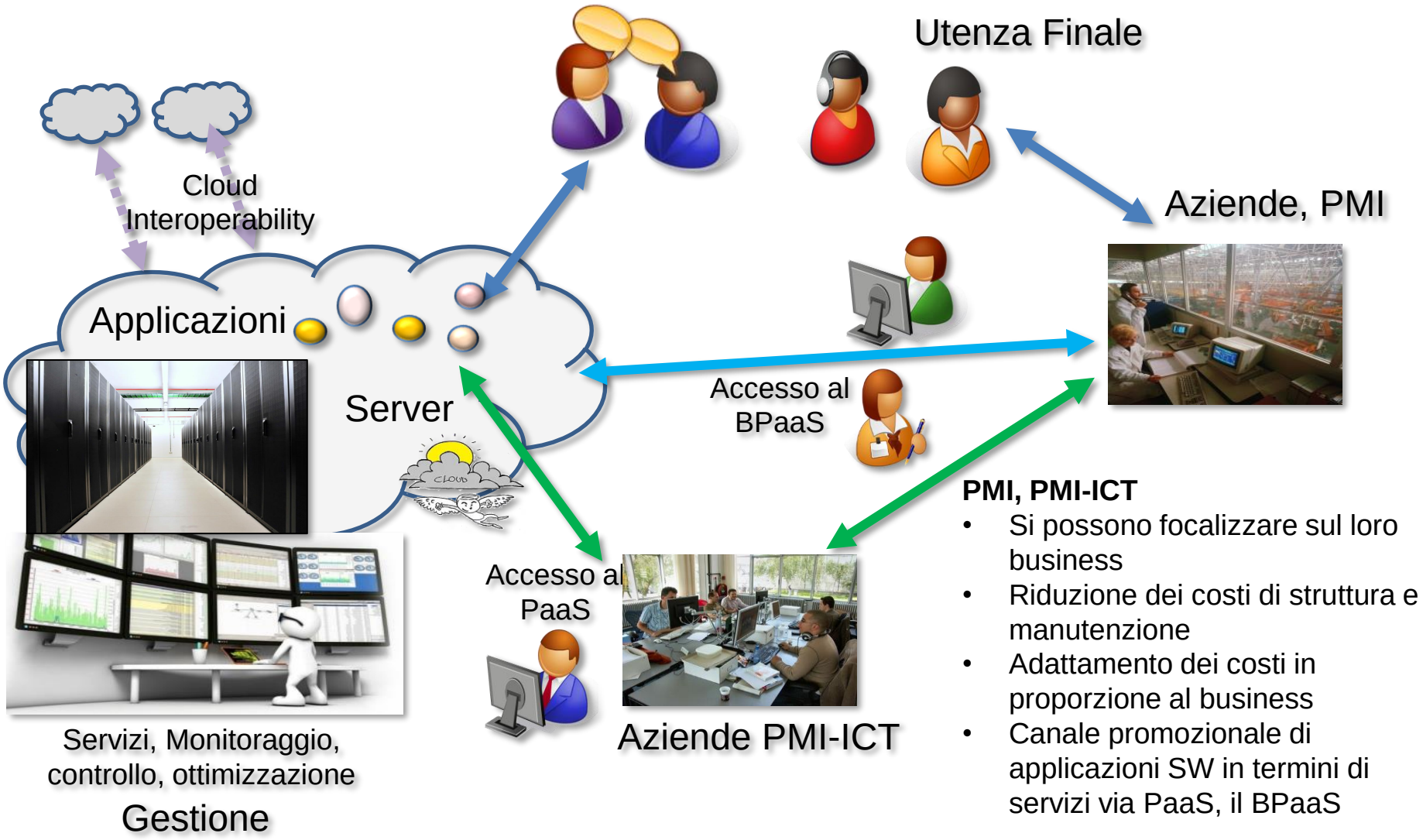
Contesto SaaS, 2/2

- **SaaS: Software as a Service (continua)**
  - **Monitoraggio**: soluzioni che dovrebbero permettere di
    - **controllare** non solo il consumo delle risorse a livello IaaS, PaaS ma anche gli aspetti SaaS dell'Applicazioni a consumo, per esempio le relazioni, i sevizi interni, etc..
    - **Automatizzare** la configurazione del sistema di monitoraggio: sia per singole che VM che per business complessi con varie VM e Servizi collegati
      - **Definire** delle metriche di alto livello che possano mostrare il vero comportamento del sistema
      - **Valutarle** in modo automatico valori collegati al contratto
    - **Esportare** i dati di monitoraggio verso gli amministratori e anche verso i clienti finali
  - **Costi di gestione: *molto elevati*** *se vengono svolte a mano* le:
    - operazioni di configurazione, allocazione e deploy su cloud
    - operazioni di configurazione del monitoraggio: basso e alto livello, connessione fra ogni singola metrica e il contratto
    - valutazioni sui cambi di configurazione, connessione fra ogni singola metrica e il contratto
    - riconfigurazioni sulla base di valutazioni complesse e strategie non formalizzate
    - …

Obiettivi di ICARO

- **Ridurre i costi di gestione** in caso di configurazioni complesse: IaaS, PaaS, SaaS, PBaaS, automatizzando i processi di
  - **Vendita delle soluzioni su cloud**
  - **Configurazioni integrate: IaaS, PaaS, SaaS, PBaaS**
    - **Config e Deploy di applicazioni complesse, multitier …**
    - **Config e gestione di soluzioni di monitoraggio a livello di metriche integrate per aspetti: IaaS, PaaS, SaaS, PBaaS, business, SLA**
  - **Controllo e monitoraggio dei sistemi, Business Level, SLA level**
  - **Ri-configurazione sulla base della valutazione di condizioni complesse**
- *Le soluzioni **Cloud attuali** sono spesso:*
  - *rigide, provocano inerzie notevoli all'adattamento rispetto a nuove esigenze, ad incrementi di carico/mercato, etc.*
- ***ICARO ha inteso sviluppare** soluzioni per avere:*
  - *Ridurre i costi e fornire maggiore flessibilità sul Cloud*
  - *Adattare il loro parco software alle nuove esigenze*
  - *Fornire servizi a consumo: Business Process as a Service, BPaaS*

ICARO Cloud



PMI, PMI-ICT

- Si possono focalizzare sul loro business
- Riduzione dei costi di struttura e manutenzione
- Adattamento dei costi in proporzione al business
- Canale promozionale di applicazioni SW in termini di servizi via PaaS, il BPaaS

Obiettivi Tecnici, overview

- *modello descrittivo per servizi e applicazioni*
- *sistema automatico di configurazione*
- *reasoner che prendere decisioni su configurazioni: consistenza e completezza*
- *soluzione di produzione del business, config automatica*
- *motore di intelligence per il cloud*
- *algoritmi per il monitoraggio del comportamento di servizi e applicazioni: IaaS, PaaS, SaaS,…*
- *soluzione PaaS di tipo evoluto*
- *algoritmi per la valutazione di modelli di costo e di business*
- *adeguamento dell'architettura su alcune applicazioni*
- *algoritmi di ottimizzazione della gestione del cloud*

Cloud Orchestration

- *the ability to instantly manage, automate and streamline all elements of a cloud platform including the man agement of physical and virtual resources*

- Una piattaforma di orchestrazione consente ai cloud service provider di separare le risorse virtuali da quelle fisiche
- Attraverso il software, il cliente è in grado di riconfigurare un server in pochi minuti
- Se un progetto è finito o in pausa, il server può essere usato per altri compiti
- Attraverso un pannello di controllo o un API il cliente accende/spenge quello che desidera, e il software fa il resto

- *Orchestration*: è un concetto fondamentale del Cloud Computing, spesso confuso con quello di *Automation*
- L'automazione non basta
- Diventa necessario orchestrare i processi
  - si mettono assieme i vari aspetti della gestione del cloud
  - semplificare la gestione dell'automazione e trovare un bilanciamento per gestire hypervisor, l'utilizzazione delle risorse, disponibilità, scalabilità, prestazioni, sulla base dei requisiti business, con l'intento di fornire servizi velocemente e in modo efficace
  - Gestire workload tramite un'interfaccia singola, in modo da garantire efficienza, controllo e scalabilità
  - Identifica i task automatici che compongono un processo e ne assicura la riusabilità all'interno delle operazioni
  - Semplifica la produzione, il provisioning, il de-provisioning
  - Permette di non fare azioni ripetitive manualmente

Orchestration vs Automation

	Automation	Orchestration
Numero di task coinvolti	Riguarda un singolo task	Combina i singoli task in processi
Pre-requisiti per le prestazioni	I task automatici devono essere eseguiti in sequenza, in accordo alle linee guida sulla sicurezza, con gli opportuni permessi	L'orchestration assicura le prestazioni di ciascuno dei task automatici in un ordine definito, all'interno del workflow
Error-proneness (propensione all'errore)	Richiede molto lavoro di coding, con il rischio di errori nell'esecuzione manuale dei comandi	Elimina gli errori ottimizzando la quantità di codice richiesto. Riutilizza i blocchi di codice in modo da eliminare le ridondanze
Utilizzazione delle risorse	I task automatici esistono come entità autonome, e quindi sfruttano le risorse indipendentemente: spreco delle risorse	Enumera le risorse, IAM (Identity and Access Management) roles, i tipi di istanze, ecc. in modo da ridurre i tempi e fornire risultati più precisi
Esempio	Avviare o configurare un web server	Combina i task automatici in un singolo workflow in modo da soddisfare le richieste

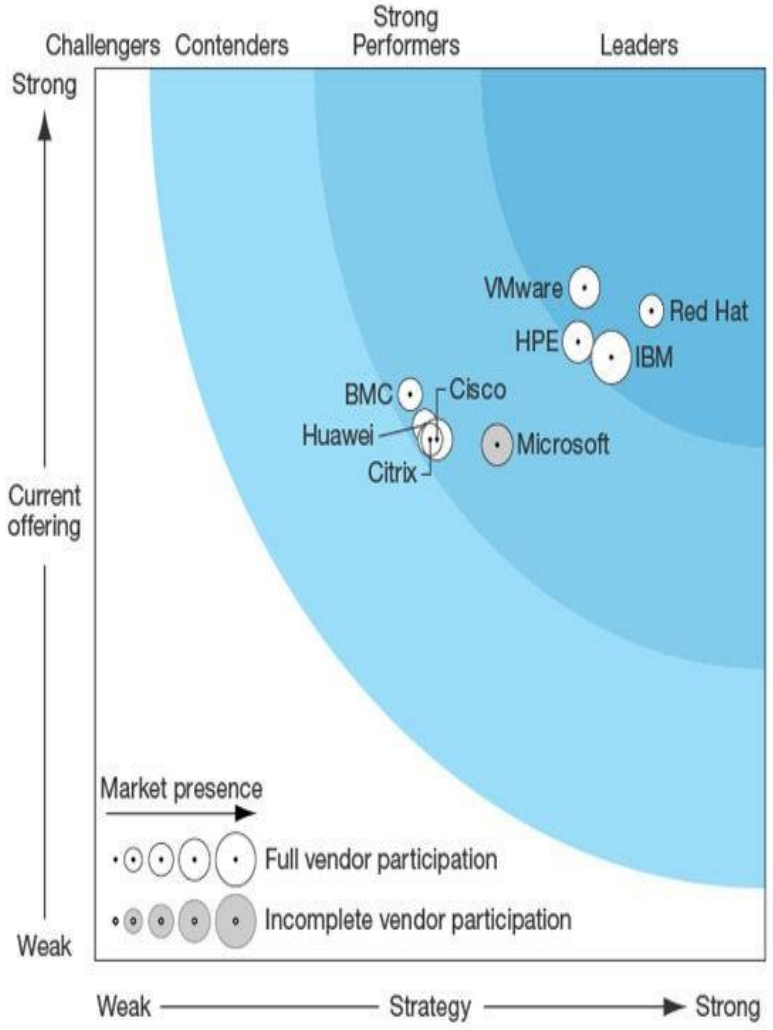
Orchestration

Cloud Orchestration = Automation + Integration + Best Practices

Automation = automazione di specifici task

Orchestration = automazione di processi e workflow

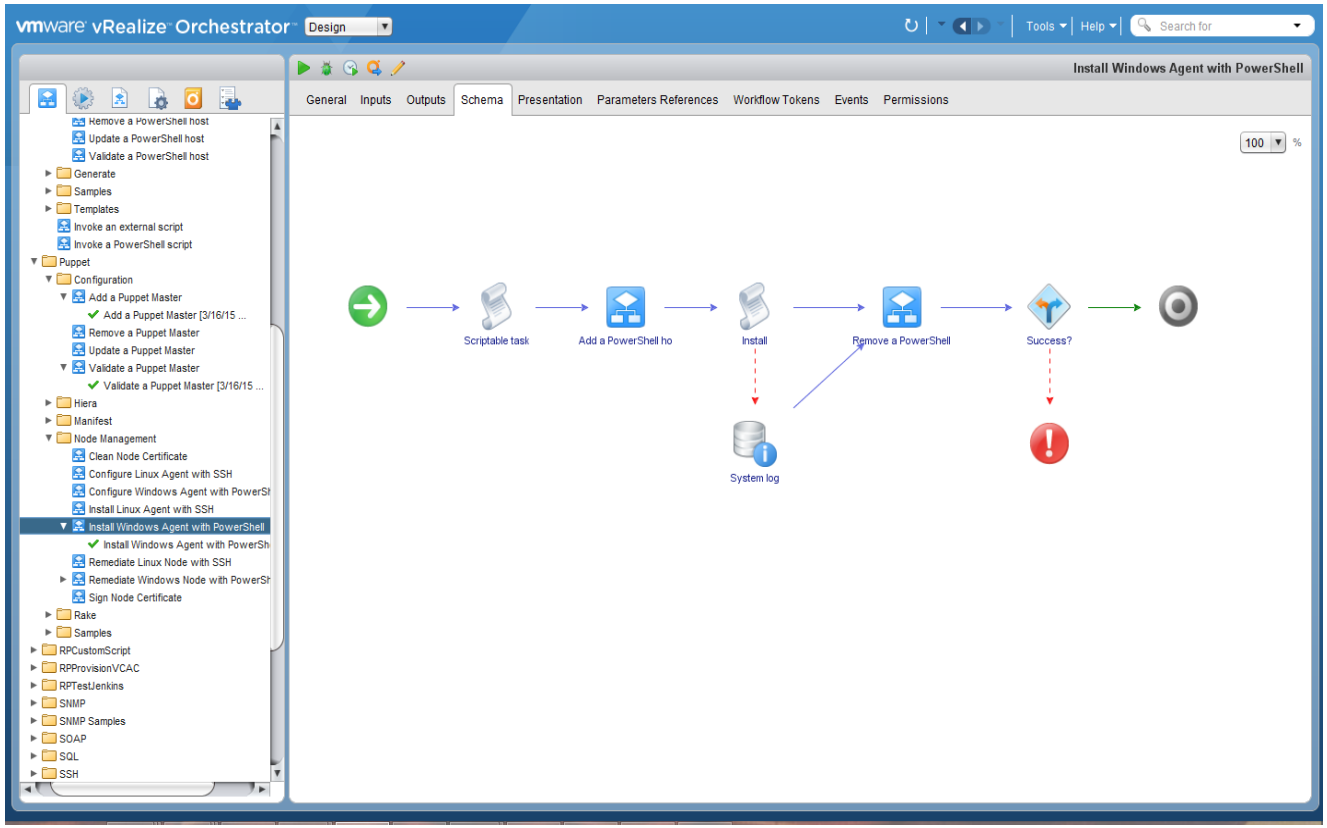
Mercato



- *In pochi anni sono state sviluppate centinaia di soluzioni per il cloud management (e molte specificamente per l'orchestrazione dei processi cloud)*
- *Però i cloud leader (quelli che si dividono la fetta maggiore del mercato) sono pochi*

vRealize: workflow schema

Workflow Schema, è una rapperesentazione grafica che mostra il workflow come un diagramma che interconnette gli elementi del workflow. È l'elemento più importante di un workflow perché ne determina la logica



vRealize: workflow schema elements

Start Workflow, il punto di inizio del workflow. Tutti i workflow contengono questo elemento (e soltanto uno). Ha un output e nessun input e non può essere ri vmware lo schema.

Scriptable task, task di tipo generale per la scrittura di funzioni Javascript.

Decision, una funzione booleana. Riceve un input e restituisce true/false. Il tipo di decisione che l'elemento prende dipende dal tipo di input parameter, e condiziona la direzione del flusso che il workflow prende.

Custom decision, una funziona booleana. Riceve diversi input e li processa in accordo allo script definito restituendo true/false.

Decision activity, una funzione booleana. Esegue un workflow e collega i suoi output parameter a un percorso true/false.

User Interaction, consente agli utenti di passare nuovi parametri di input al workflow. Permette di definire il modo in cui i presentare la richiesta bloccante di parametri di input e impostare vincoli (e.g., range, timeout). Consente di impostare permessi in modo da definire quali utenti possono impostare i parametri.

vRealize: workflow schema elements

Waiting timer, utilizzato per workflow che richiedono tempi di esecuzione lunghi. Quando il workflow arriva al waiting timer entra in stato di attesa. Si può impostare una data alla quale il workflow riprende l'esecuzione. Mentre è in attesa della data, il workflow è nello stato *waiting-signal*.

Waiting event, utilizzato per workflow che richiedono tempi di esecuzione lunghi. Quando il workflow arriva al waiting event entra in stato di attesa. Si può definire un trigger event che il workflow si aspetta per poter riprendere l'esecuzione. Mentre è in attesa dell'evento il workflow è nello stato *waiting-signal*.

End workflow, il punto finale di un workflow. Si possono avere end workflow multipli in uno schema, che rappresentano le possibili uscite prodotte da esso. Hanno un input e nessun output.

Thrown exception, crea un'eccezione e ferma il workflow. Può comparire molte volte in un workflow. Ha un parametro di input (solo di tipo String) e nessun output. Quando il workflow raggiunge un elemento Exception entra nello stato *failed*.

vRealize: workflow schema elements

vmware

Workflow note, consente di annotare le varie sezioni di un workflow.

Action element, invoca un'azione dalla libreria delle azioni dell'Orchestrator. Quando il workflow raggiunge questo element esegue l'azione.

Workflow element, avvia un workflow in modo sincrono. Il workflow originario continua la sua esecuzione solo dopo che è completata quella del workflow invocato.

Foreach element, esegue un workflow da ogni elemento di un array. Per esempio, si può eseguire un'operazione di rename di tutte le Virtual Machine di una cartella.

Asynchronous workflow, avvia un workflow in modo asincrono. Il workflow originario non attende il completamento dell'esecuzione di quello invocato, ma continua la sua esecuzione.

vRealize: workflow schema elements

Schedule workflow, crea un task per eseguire il workflow a un tempo prestabilito e poi il workflow continua la sua esecuzione.

Nested workflows, esegue workflow multipli simultaneamente. Si può scegliere di annidare workflow locali o remoti, cioè che risiedono su differenti Orchestrator server. Si possono eseguire workflow con differenti credenziali. Il workflow attende il completamento di tutti i workflow prima di riprendere la sua esecuzione.

Handle error, gestisce gli errori per uno specifico elemento del workflow. Il workflow può gestire l'errore creando un exception, chiamando un altro workflow, o eseguendo uno script.

Default error handler, gestisce gli errori del workflow che non sono gestiti dallo standard error handler.

Switch, commuta verso workflow alternativi, sulla base di un parametro o di un attributo.

Chef

Chef, è un potente strumento di *configuration management* per la gestione di infrastrutture attraverso il codice. Automatizza la configurazione, il deploy, e la gestione delle applicazioni

- Utilizza *cookbook* per determinare come ogni nodo deve essere configurato
- Un cookbook consiste di molte “ricette” (recipe); una ricetta è uno script di automazione per un particolare servizio scritto in Ruby
- *Chef Client*, è un agent che gira su un nodo ed esegue i task che lo configurano
- Chef può gestire qualunque cosa può eseguire un client Chef (macchine fisiche, container, istanze cloud-based)
- *Chef server*, è il repository per tutti i dati di configurazione. Sia il client che il server Chef comunicano in modo sicuro, utilizzando una combinazione di chiavi pubbliche e private, che assicurano che il server risponda soltanto alle richieste del client

Chef

- Facile da installare (utilizza un third-party installer)
- È utilizzato particolarmente per l'automazione a livello OS, come il deploy dei server, l'applicazione di patch
- Richiede esperienza di programmazione
- Ha una curva di apprendimento molto ripida, una volta superata la quale offre potenza e flessibilità

Puppet

Puppet, è simile a Chef. Richiede l'installazione di un master server e un client agent nei nodi

- Si possono scaricare e installare i moduli necessary tramite comandi Puppet. Come Chef, offre una versione enterprise che consente di avere funzionalità aggiuntive
- Chef e Puppet svolgono le stesse funzioni di base, ma utlizzano un approccio differente. Chef è un sistema più integrato e monolitico, mentre Puppet consiste di diversi servizi. Questo rende Chef in un certo senso più facile da configurare e gestire
- È pensato per gli amministratori di sistema che necessitano di specificare configurazioni come dipendenze, mentre Chef è fatto per gli sviluppatori che scrivono il codice per il deploy
- Dipende fortemente dal proprio Domain Specific Language (DSL) per la definizione di regole di configurazione, mentre il DSL di Chef è solo un "supplemento" a Ruby, linguaggio utilizzato per le ricette Chef
- È utilizzato per la mid-level automation (e.g., installare database, avviare Apache)

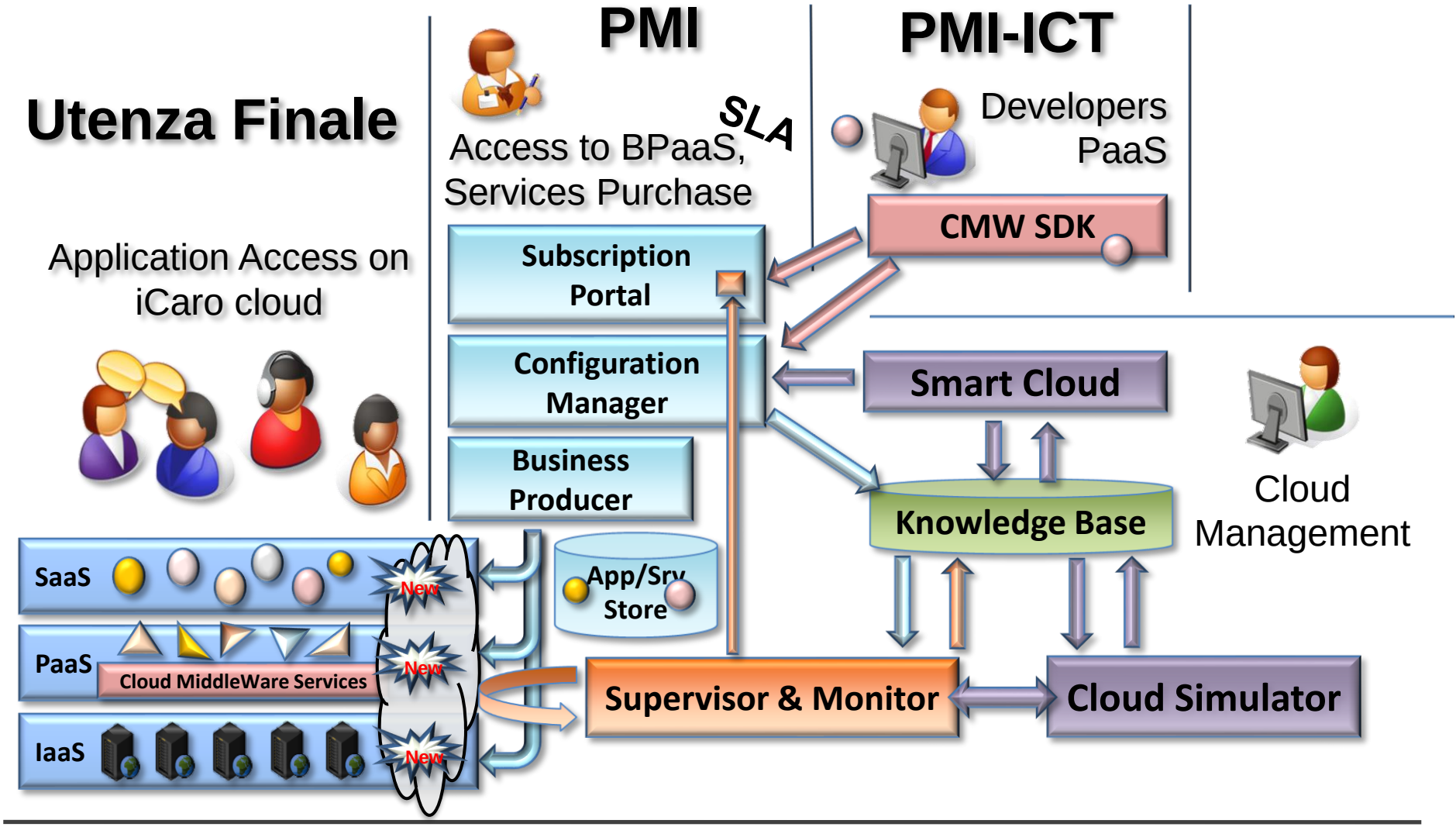
OpenStack

- *OpenStack*, è una piattaforma di cloud computing open source (licenza Apache), utilizzata principalmente come soluzione Infrastructure as a service (IaaS)
- Consiste in una serie di progetti interconnessi che controllano insiemi di risorse di processing, storage, rete, attraverso un data center
- Gli utenti gestiscono le risorse attraverso una dashboard web-based, strumenti a riga di comando, oppure API RESTful
- È un progetto congiunto iniziato nel 2010 fra Rackspace Hosting e NASA. È gestito dalla OpenStack Foundation (organizzazione no-profit). Ne fanno parte più di 200 aziende (e.g., IBM, Canonical, Cisco, Dell, EMC, Ericsson, Hewlett-Packard, Huawei, Intel, Mellanox, Mirantis, Oracle, Red Hat, SUSE Linux, VMware, Yahoo)
- I componenti principali sono: Nova (compute), Cinder (block storage), Glance (image library), Swift (object storage), Neutron (network), Keystone (identity), Heat (orchestration tool)

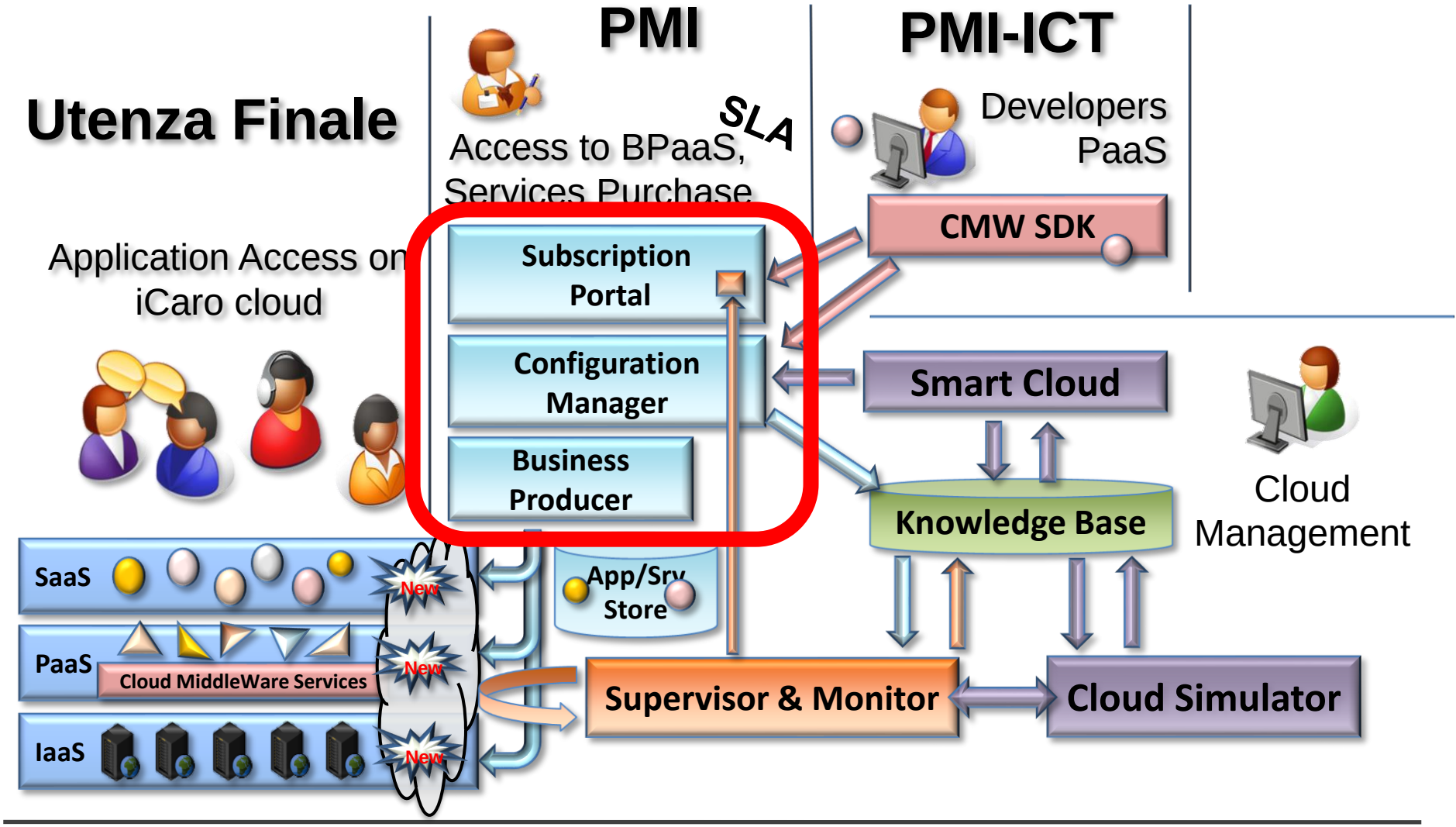
OpenStack (Heat)

- *Heat*, è un meccanismo di orchestrazione conosciuto come progetto *Orchestration for OpenStack on OpenStack* (OOO).
- Fornisce un'orchestrazione basata su template per descrivere un'applicazione cloud tramite l'esecuzione di appropriate chiamate alle API OpenStack, che generano applicazioni nel cloud
- Integra altri componenti core di OpenStack in un template
- I template consentono la creazione della maggior parte dei tipi di risorse OpenStack (*instances*, *floating IPs*, *volume*, *security group*, *user*) così come funzionalità più avanzate come instance *high availability*, *instance autoscaling*, *nested stack*
- Alternativo alla scrittura di uno script per la gestione delle risorse in OpenStack (setting up dei server, aggiunta volumi, gestione reti, ecc.)
  - Si crea un Heat template che specifica l'infrastruttura richiesta
  - In caso di cambiamenti futuri al servizio si può modificare il template Heat e l'engine di Heat farà i cambiamenti necessari quando il template verrà rieseguito

Architettura ICARO



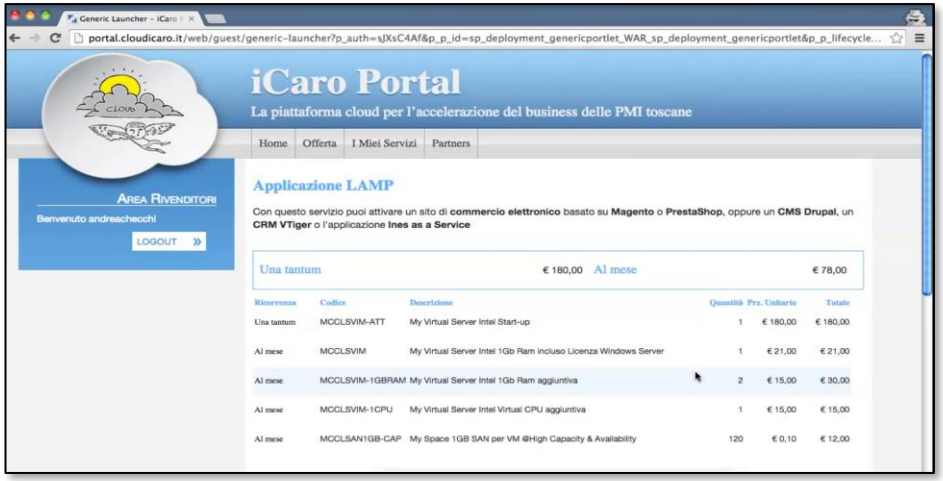
Architettura ICARO

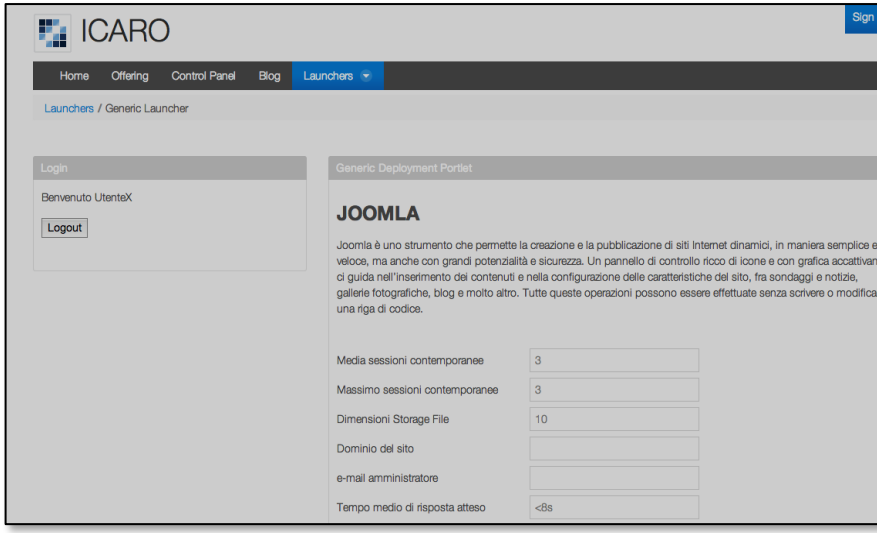


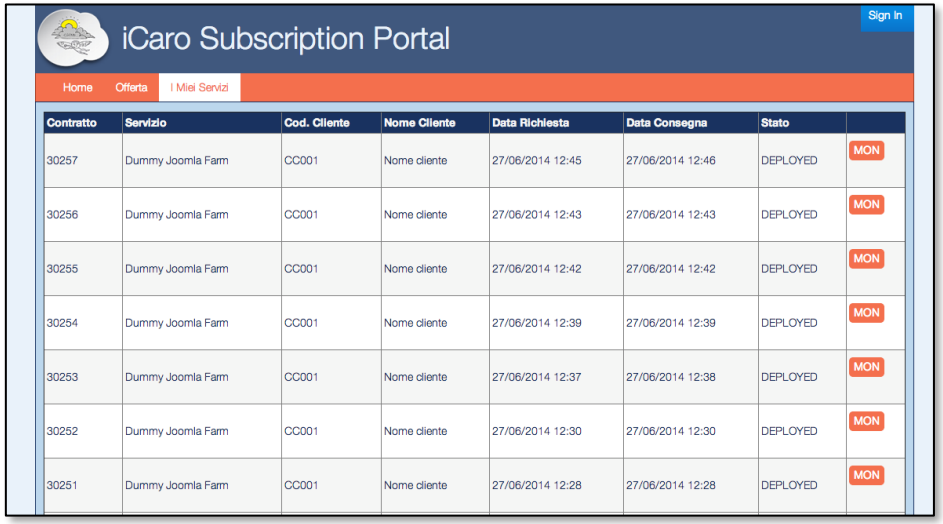
Subscription Portal

- Interfaccia di **vendita di soluzioni a consumo**, che possono andare da Host, Macchine Virtuali, applicazioni e combinazioni complesse.
  - Svariate possibili configurazioni di applicazioni: ERP, CRM, etc.,
- **Soluzioni offerte con varie tipologie di contratti di servizio** (Service Level Agreement, SLA) e relativi parametri per il calcolo dei costi a consumo: rete, disco, fatture, email, etc.
- **produzione automatica dei pattern/configurazioni** complesse composte da: VM, applicazioni Web e non solo, servizi sulla base di buone pratiche, profiling, applicazioni legacy, e servizi per connessione con applicazioni on site remote, etc.
- **Il Business Producer**:
  - evoluzione dei tradizionali orchestrator, che tramite workflow permettono di eseguire la procedura di deploy delle applicazioni sul Cloud.
  - lavora con VCO VMware,
  - può essere esteso ad altre soluzioni di mercato.

Subscription Portal







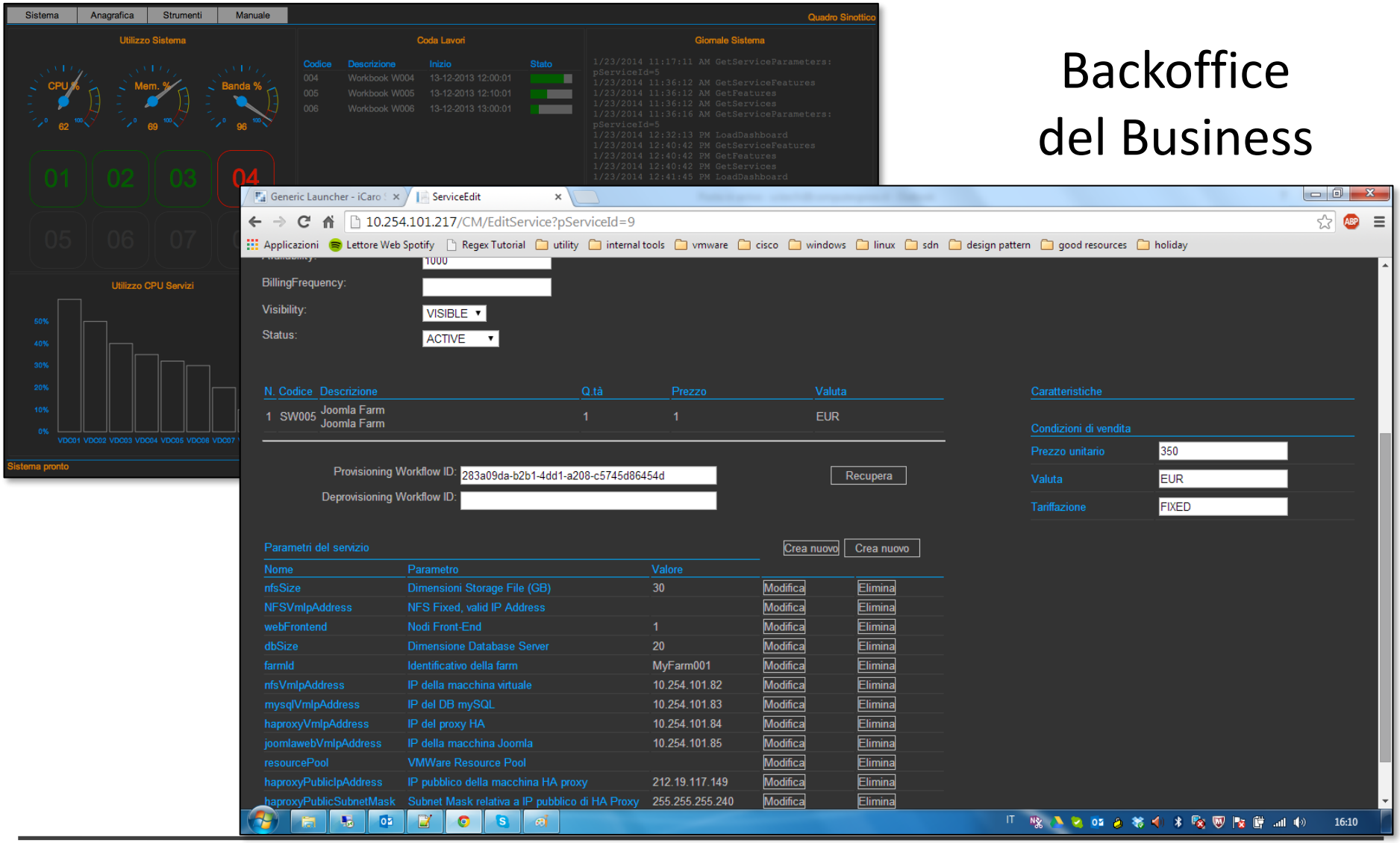
Accesso al BPaaS

Configuration Manager

- Strumento tecnico per la g**enerazione e gestione automatizzata delle configurazioni**
- **Configurazioni e SLA complesse** con regole di adattamento che includono la gestione automatizzata delle operazioni di deploy e update sul cloud tramite Business Producer.
  - Business Producer può essere un cloud manager di mercato.
- Il **Configuration Manager**, CM
  - comunica le configurazioni e le SLA allo Smart Cloud che le verifica, e che eventualmente provvede a suggerire al CM eventuali modifiche ed adattamenti.
  - richiede grafici e dati al sistema di monitoraggio, S&M.

Configuration Manager

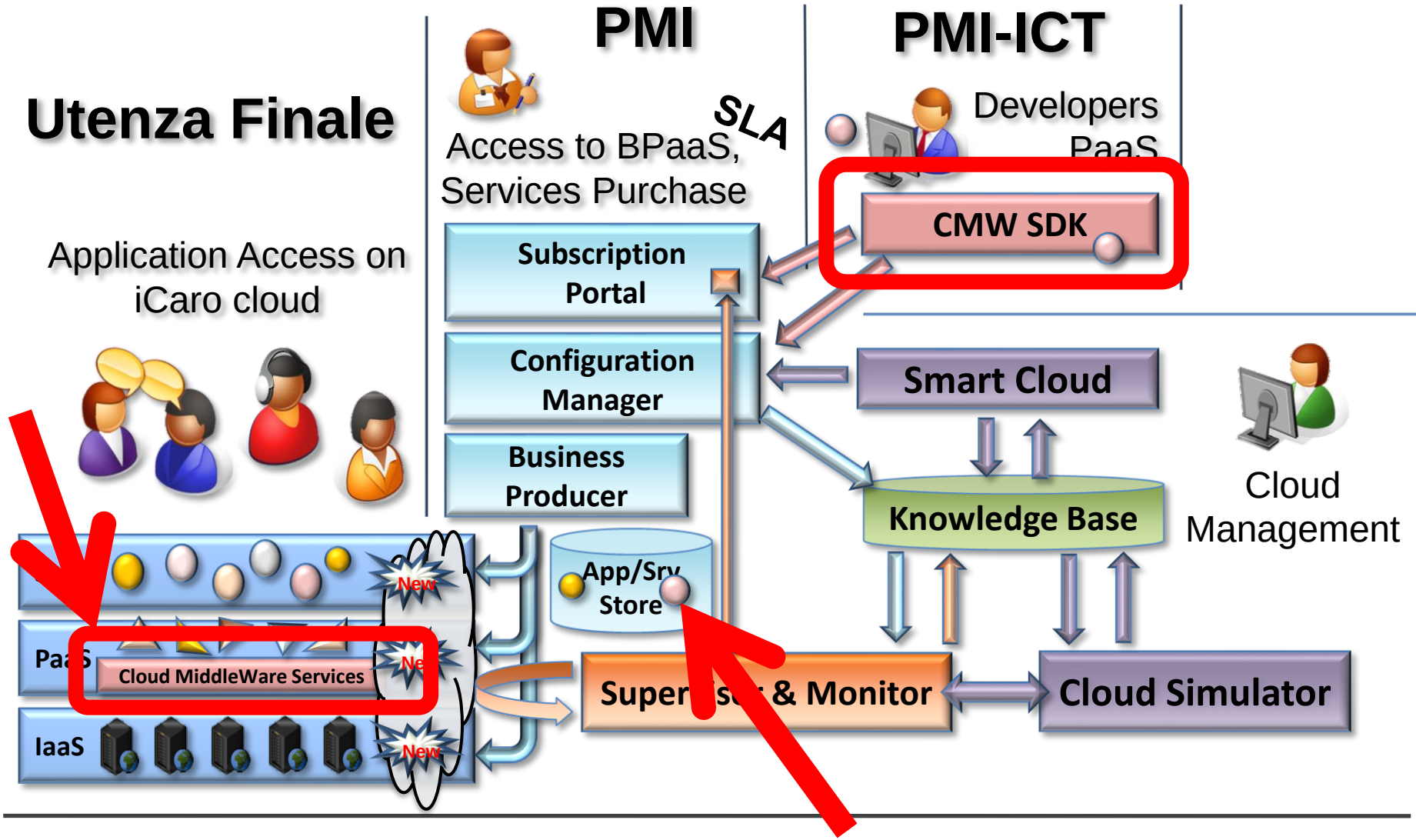
Backoffice del Business



Business Producer

- **produzione automatica dei pattern/configurazioni** complesse composte da: VM, applicazioni Web e non solo, servizi sulla base di buone pratiche, profiling, applicazioni legacy, e servizi per connessione con applicazioni on site remote, etc.
  - Il Business producer è un'evoluzione dei tradizionali orchestrator, che tramite workflow permettono di eseguire la procedura di deploy delle applicazioni sul Cloud.
- **Il Business Producer:**
  - nella sperimentazione ICARO lavora con VCO VMware,
  - può essere esteso ad altre soluzioni di mercato.

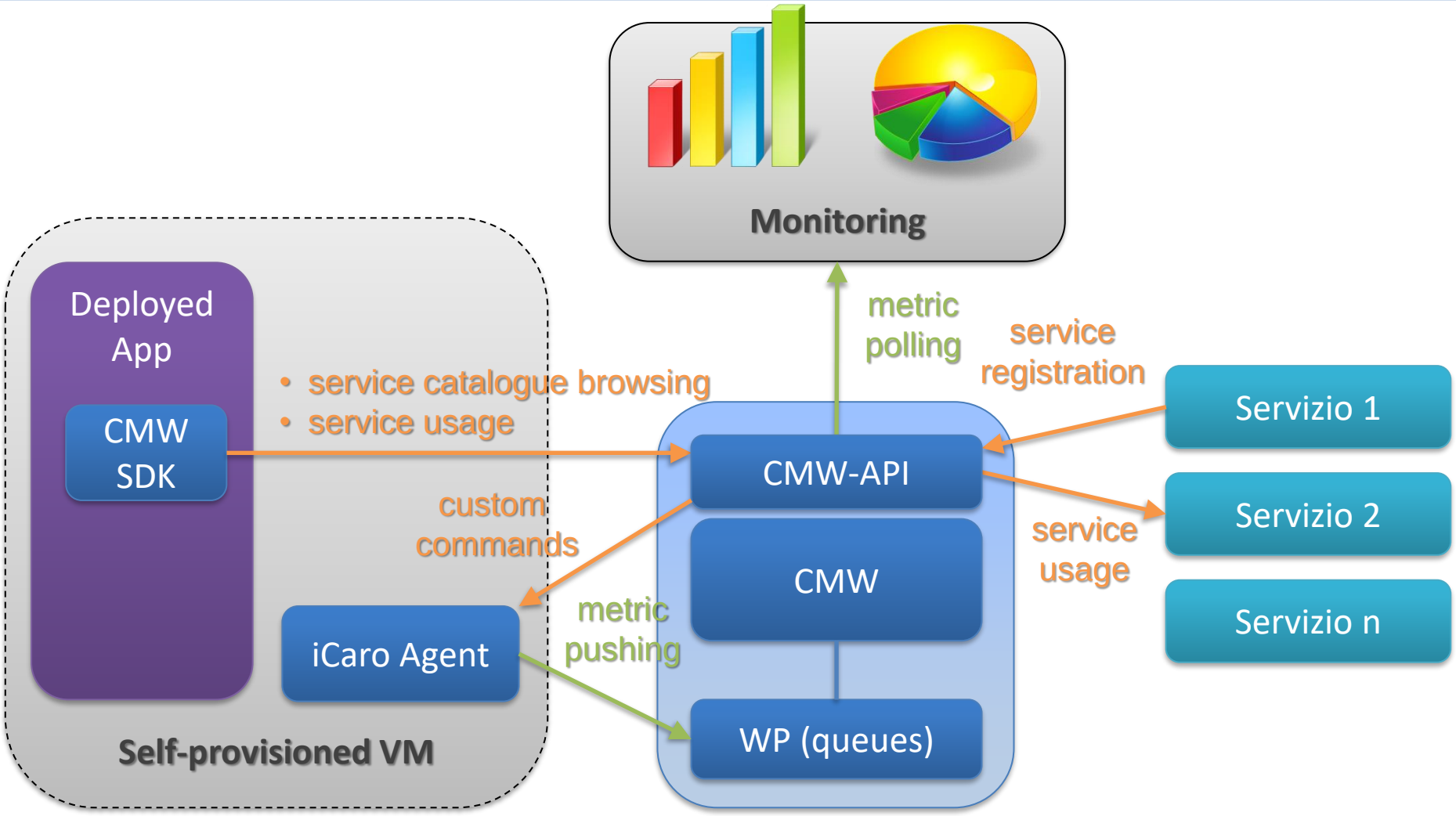
Architettura ICARO



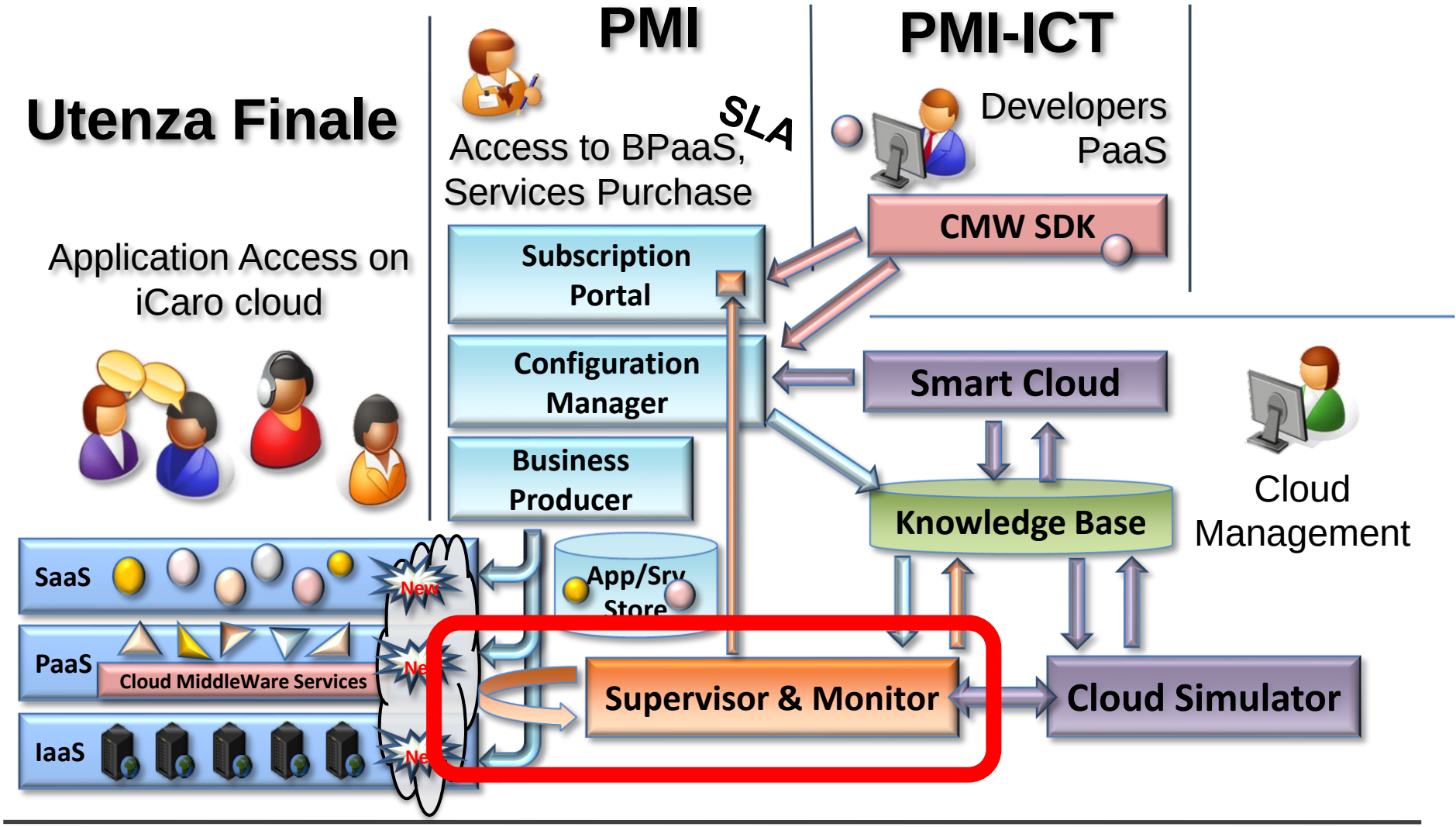
Cloud Middleware & Cloud Middleware SDK

- **Supporto per lo sviluppo** (in Java e PHP) che consente a Sviluppatori di Applicazioni ICT di:
  - **Adattare le applicazioni** per essere gestite, vendute in ottica di consumo su ICARO.
  - utilizzare, mediante un sistema di messaggistica asincrona, uno o più servizi presenti nel catalogo gestito dal modulo CMW (es. postgreSQL, SQL Server, SMTP server, Object Storage).
  - recuperare i valori attuali delle "metriche applicative" inerenti i servizi offerti da CMW (es. dimensione di un database, numero mail inviate, ecc.).

CMW, CMW-SDK, WP: Architettura



Architettura ICARO

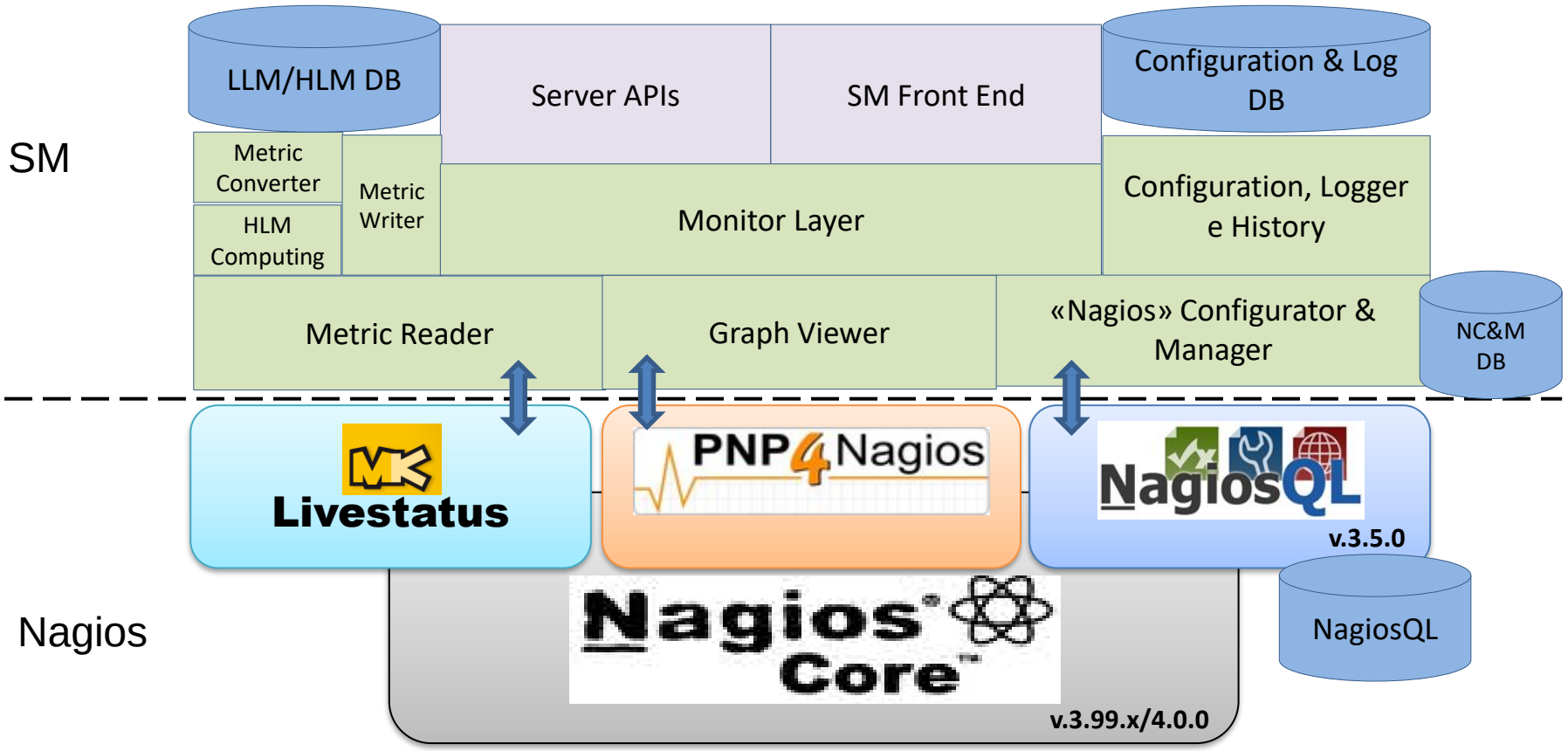


Supervisor & Monitor

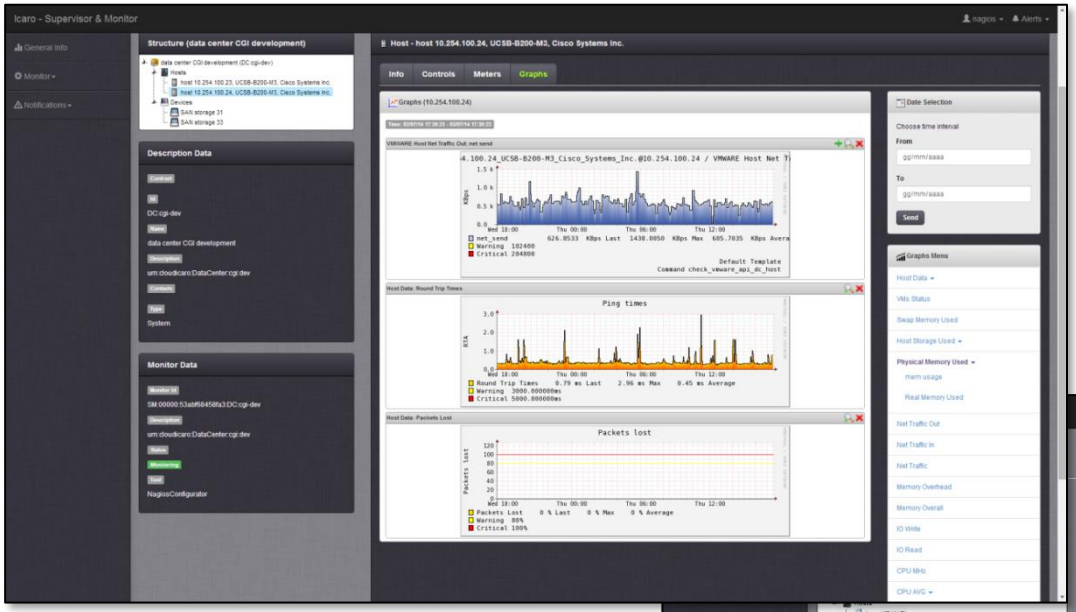
- **Supervisione e monitoraggio delle risorse e dei consumi** in modo integrato analizzando e tenendo sotto controllo:
  - risorse cloud ai livelli: IaaS, SaaS, PaaS, BPaaS;
  - metriche applicative di Applicazioni e Servizi single/multi-tier: standard e caricati tramite il PaaS;
  - metriche definite in relazione alle SLA;
  - servizi interni ed esterni anche locati in altri cloud e sistemi, come supervisione dello stato dei processi: http, ftp, reti, server esterni, Web App Server, etc.

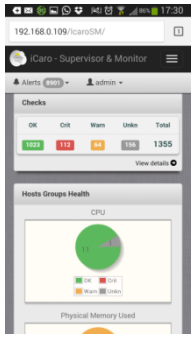
- **Il Supervisor & Monitor:**
  - è configurato in modo automatico dalla Knowledge Base
  - in ICARO utilizza il tool Nagios ed è in grado di controllare e configurare Nagios in modo automatizzato e di accedere in remoto alle funzionalità dei suoi componenti
  - Livello di astrazione del monitoraggio: possibilità di utilizzare differenti sistemi di monitoraggio di basso livello

SM – Architettura & Nagios

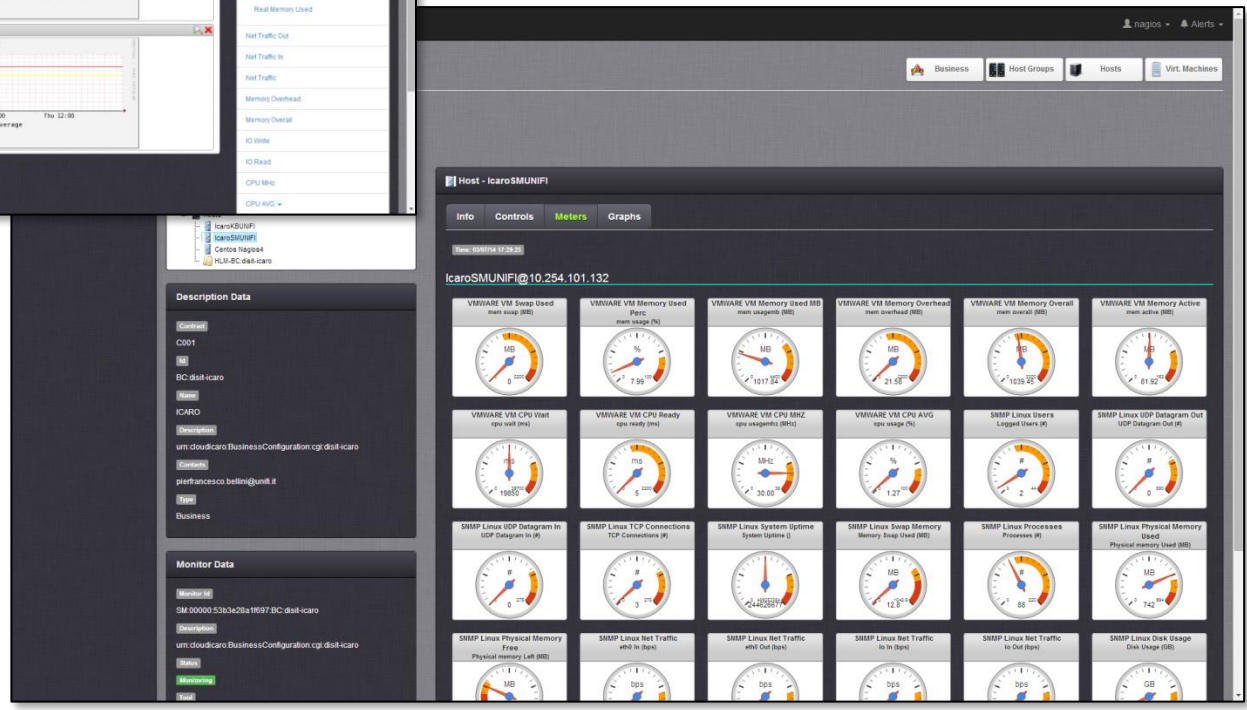


Supervisor & Monitor Front-End



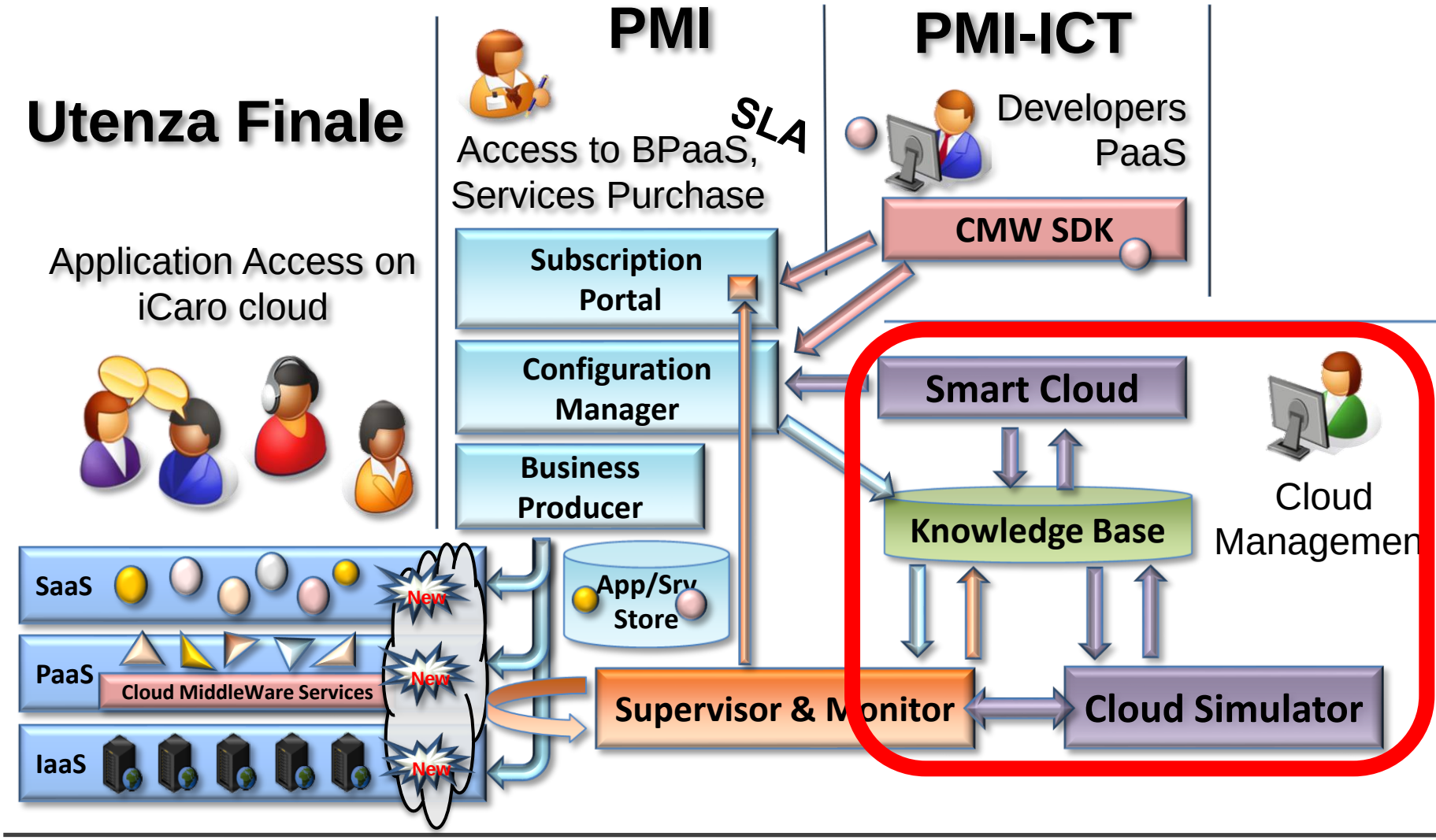


...on Mobile Devices



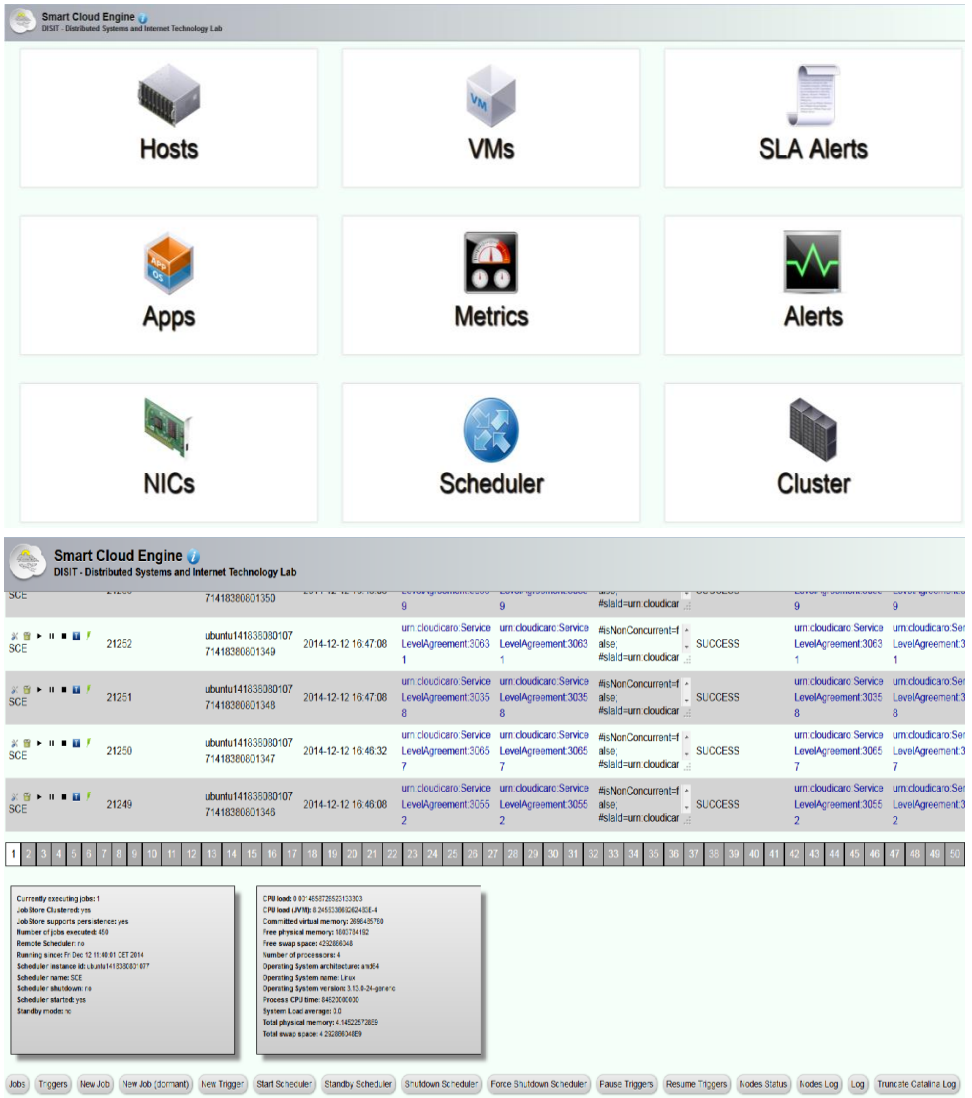
- Monitoraggio del Business
- Su PC e Mobile
- Completamente automatizzato

Architettura ICARO



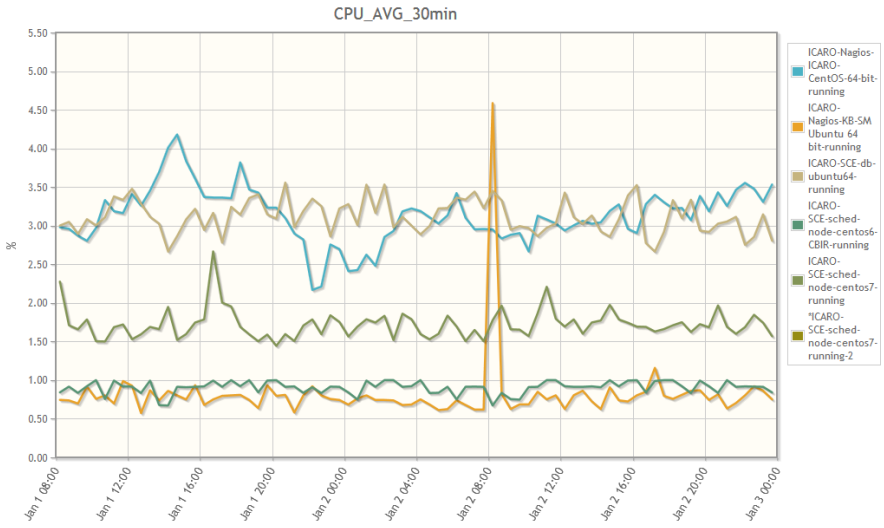
Smart Cloud Engine

- **Processi AUTOMATICI per**
  - Verifica e validazione di consistenza e completezza delle configurazioni
  - Controllo della salute e del comportamento IaaS, PaaS, SaaS, Business, SLA con metriche complesse
  - Supporto alle decisioni per Scaling, cloning, migrazione e riconfigurazione
  - Processi di ottimizzazione
- Usa come modello la KB



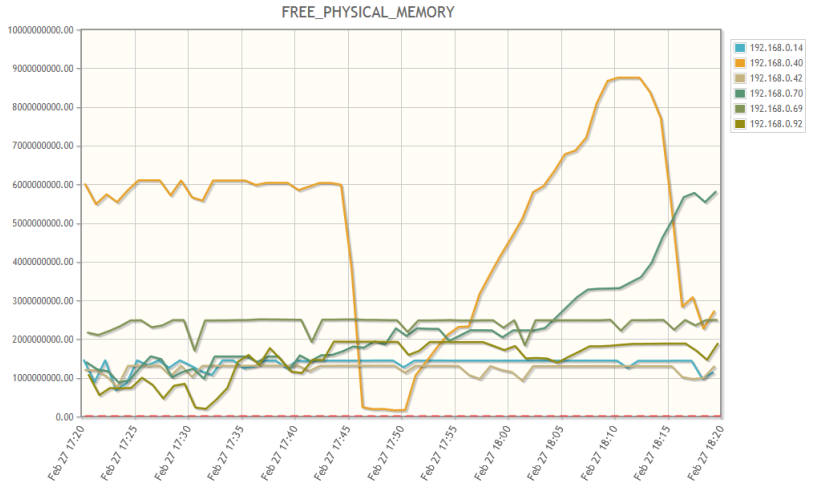
Smart Cloud Engine

- Report degli allarmi relativi alle singole SLA;
- Grafici delle metriche combinati per SLA o singoli;



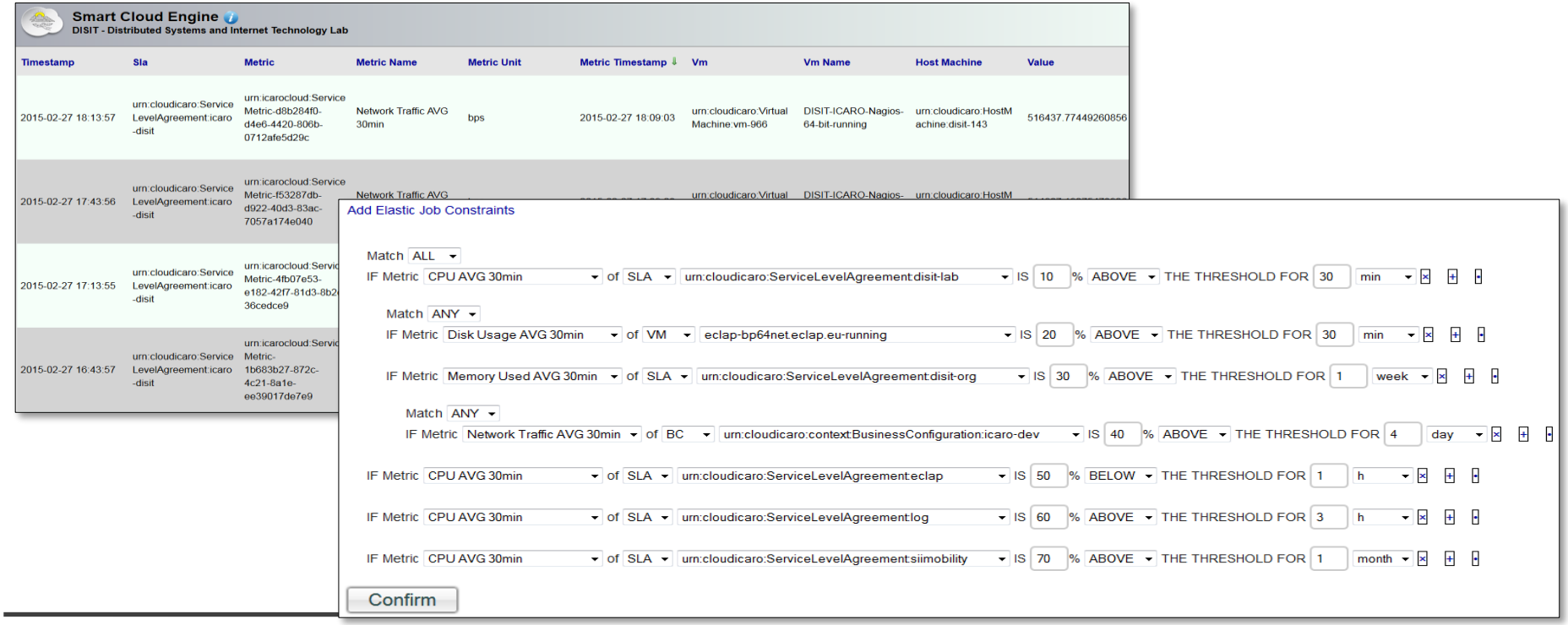
- Visualizzazione dello stato del cluster sia in forma aggregata che per singolo parametro (memoria, cpu, swap ecc.);

- Notifica delle VM che non producono dati in legenda;
- Definizione intervalli temporali;



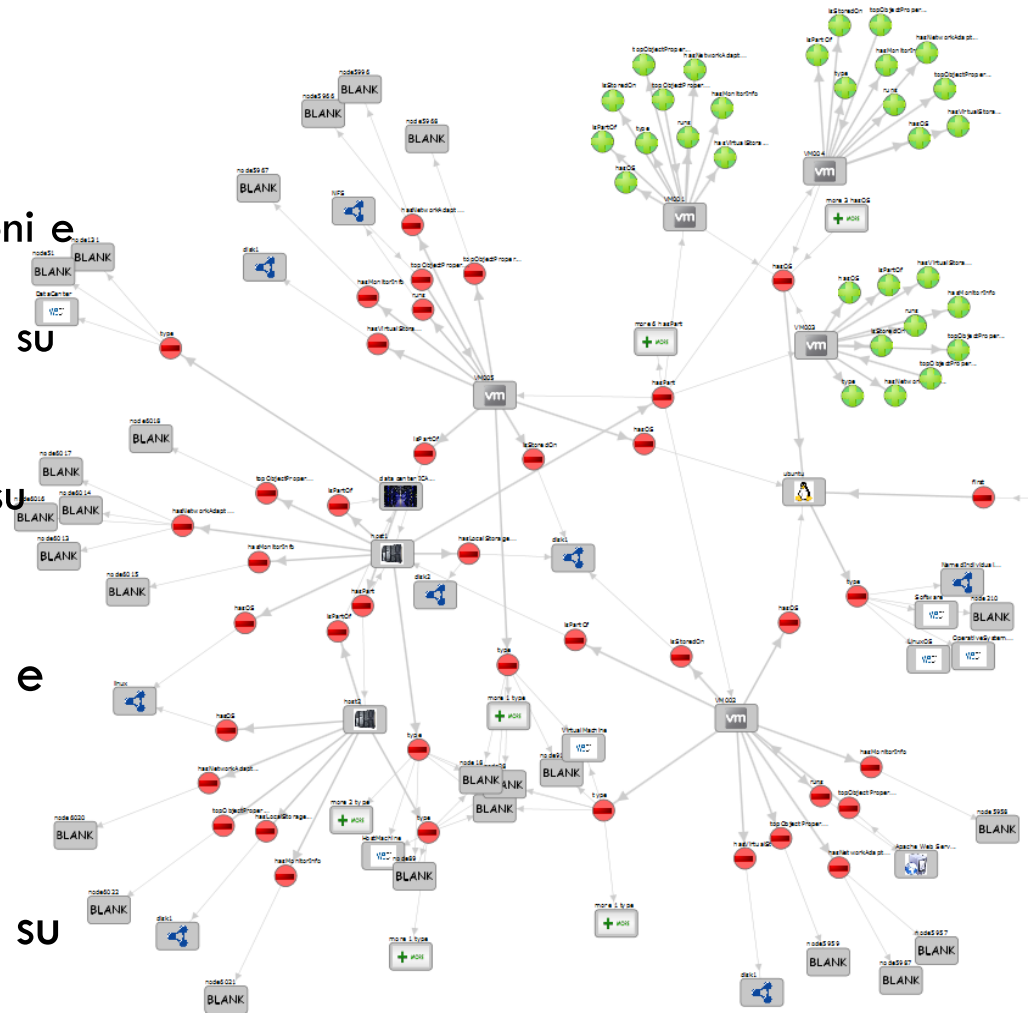
Smart Cloud Engine

- Visualizzazione eventi di allarme sia in forma aggregata che per singola metrica (data, soglia, valore misurato, configurazione ecc.)
- Strategy Condition Editor per l'Elastic Cloud
  - definizione di regole booleane di complessità arbitraria per l'attivazione di procedure di scaling, controllo, autoregolazione ecc.
  - vincoli su metriche e SLA, VM, Business Configuration con impostazione di soglie (%)



Knowledge Base & Tools

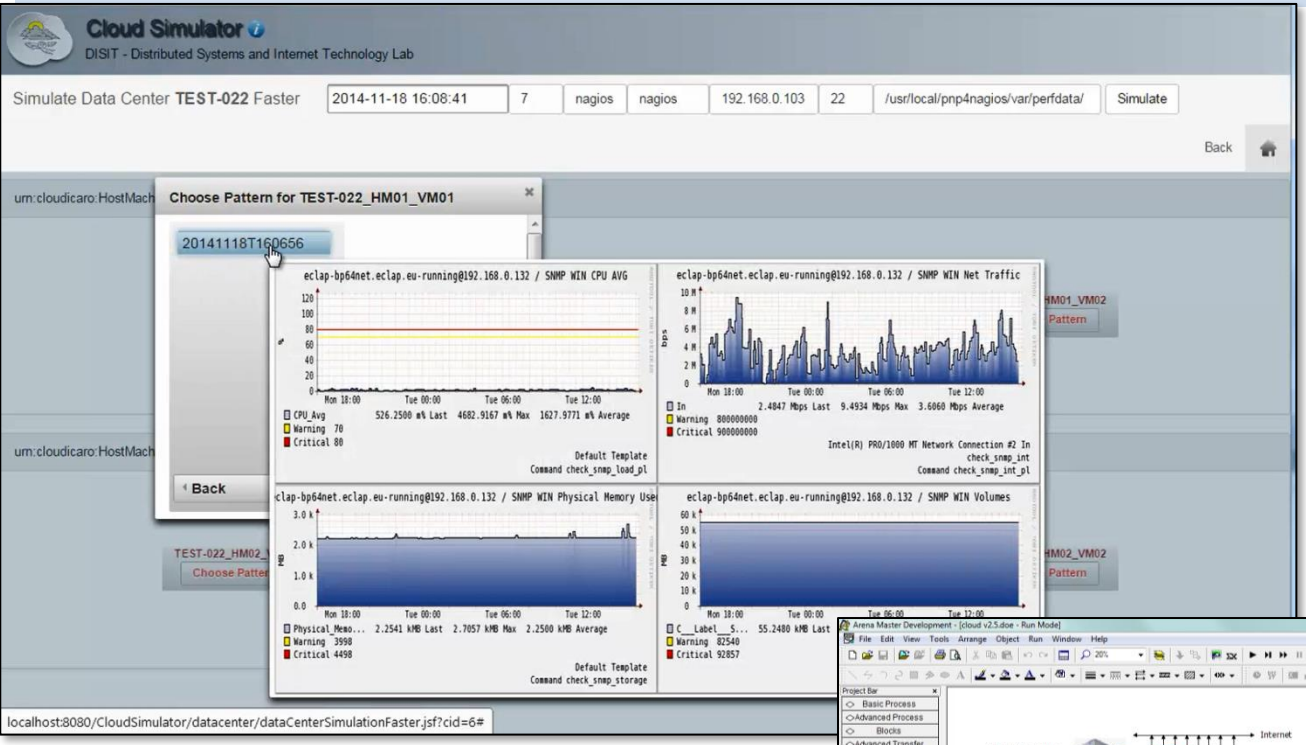
- **Via API Rest riceve le configurazioni e le variazioni**
- **Modello di Cloud intelligence**
  - Formalizzazione di configurazioni e SLA (Service Level Agreement)
  - reasoner supporto alle decisioni su configurazioni: consistenza e completezza
  - adeguamento dell'architettura su alcune applicazioni
- **Tecnologia**
  - Knowledge base: RDF store e inference engine
  - Smart Cloud Ontology: http://www.disit.org/5604
  - Esempio di dato accessibile su http://log.disit.org



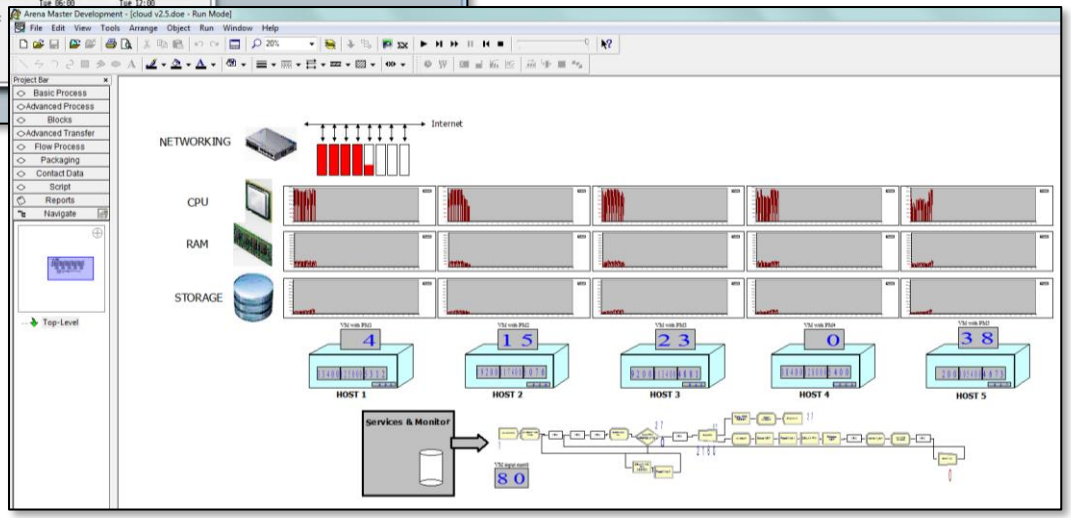
Cloud Simulator

- **Permette di**
  - **Simulare** il comportamento di carico di datacenter complessi
  - **creare situazioni di carico** partendo da andamenti di carico reali dallo storico del sistema di monitoraggio
  - **studiare** gli effetti del carico sulle risorse di base a livello IaaS
- **Produce andamenti Simulati** accessibili e analizzabili da Supervisor & Monitor come dallo Smart Cloud Engine
- **Si integra** con
  - Lo Smart Cloud Engine per l'esecuzione di processi di controllo e valutazione e
  - la Knowledge Base per gestione delle configurazioni e dei dati, navigazione nella rappresentazione complessa del cloud
  - Il Supervisor & Monitor per l'accesso ai dati di monitoraggio, e la produzione di grafici

Cloud Simulator



Simulare il comportamento di carico di datacenter complessi



Identificare allocazioni ottime delle risorse

Vantaggi di ICARO

- **Automazione** di svariati aspetti di gestione cloud
  - gestire con: SM, KB, SCE, gestori e data center diversi
  - allocare applicazioni in modo automatizzato
  - Vendere configurazioni complesse con costi di produzione e gestione ridotti.
- **Ridurre i costi** automatizzando i processi di
  - **Vendita delle soluzioni su cloud**
  - **Configurazione, integrando: IaaS, PaaS, SaaS, PBaaS**
    - **Config e Deploy di applicazioni SaaS anche complesse…**
    - **Config e gestione di soluzioni di monitoraggio a livello di metriche integrate: IaaS, PaaS, SaaS, PBaaS, business, SLA**
  - **Controllo e monitoraggio dei sistemi, Business Level, SLA level**
  - **Ri-configurazione sulla base della valutazione di condizioni complesse**

Portabilità e Interoperabilità

- La soluzione CM e SP vanno calate sul particolare Orchestrator anche se sono in larga parte indipendenti da questo ma sono dipendenti dai modelli di business del Cloud Service provider.
- La parte CMW è agnostica rispetto ai processi di deploy
- La soluzione di monitoraggio, SM, è trasportabile su altri monitor di basso livello, oggi usa Nagios
- La soluzione Smart Cloud, SCE-KB, è agnostica rispetto al gestore e completamente integrata con SM

Riferimenti per approfondimenti

- Versione completa della slide: http://www.disit.org/6587
- ICARO project official web page dove in documenti vi sono documenti accessibili e video: http://www.cloudicaro.it/
- Altra documentazione accessibile su
  - Video overview: http://www.disit.org/6558
  - Page with Smart Cloud Videos: http://www.disit.org/6544
  - ICARO page at DISIT http://www.disit.org/5482

References

- **VMware**: http://www.vmware.com
- **HP**: www.hp.com/go/integrityvm
- **Microsoft Hyper-V**: http://www.microsoft.com/hyper-v-server/en/us/default.aspx
- **"Windows Server 2008 Hyper-V Technical Overview"-** http://download.microsoft.com/download/4/2/b/42bea8d6-9c77-4db8-b405-6bffce59b157/Hyper-V%20Technical%20Overview.docx
- **Comparison of Hypervisors by VMware -** http://www.vmware.com/technology/whyvmware/architectures.html#c132894
- **"VMware vSphere 4" Datasheet -** http://www.vmware.com/products/vsphere/
- ICARO project: http://www.disit.org/5482, http://www.cloudicaro.it/en/

- *All the pictures representing VMware staff have been taken from Vmware library and are under rights of VMware. These slides represents only the view of the author and not that of VMware or of other companies mentioned in the slides such as HP, Microsoft or others. The slides are only provided for didactical purpose and no for profit.*

Containers

Corso di: Big Data Architectures

Prof. Paolo Nesi

Department of Systems and Informatics, University of Florence
Via S. Marta 3, 50139, Firenze, Italy
tel: +39-055-2758515, fax: +39-055-2758570

DISIT Lab, Sistemi Distribuiti e Tecnologie Internet

http://www.disit.dinfo.unifi.it/

paolo.nesi@unifi.it
http://www.disit.dinfo.unifi.it/nesi

The path to Containers

- Most of the processes have been moved to Cloud
- Cloud costs are high since the hypervisors are expensive and the orchestrator are very expensive
- The costs for cloud management are mainly:
  - Per CPU
  - Per Cores
  - Per hosts
  - Per VM
- This implies a costs per process.

- *The next slides have been taken from Docker*

Applications are transforming

From Docker

Application Modernization

Developer Issues:

- Minor code changes require full re-compile and re-test
- Application becomes single point of failure
- Application is difficult to scale

Microservices: Break application into separate operations

12-Factor Apps: Make the app independently scalable, stateless, highly available by design

	MICROSERVICES	TRADITIONAL APPS
Cloud or New Infrastructure	You are either here..	
Old Infrastructure		…or here

History of Docker

2004
Solaris Containers / Zones technology introduced

2008
Linux containers (LXC 1.0) introduced

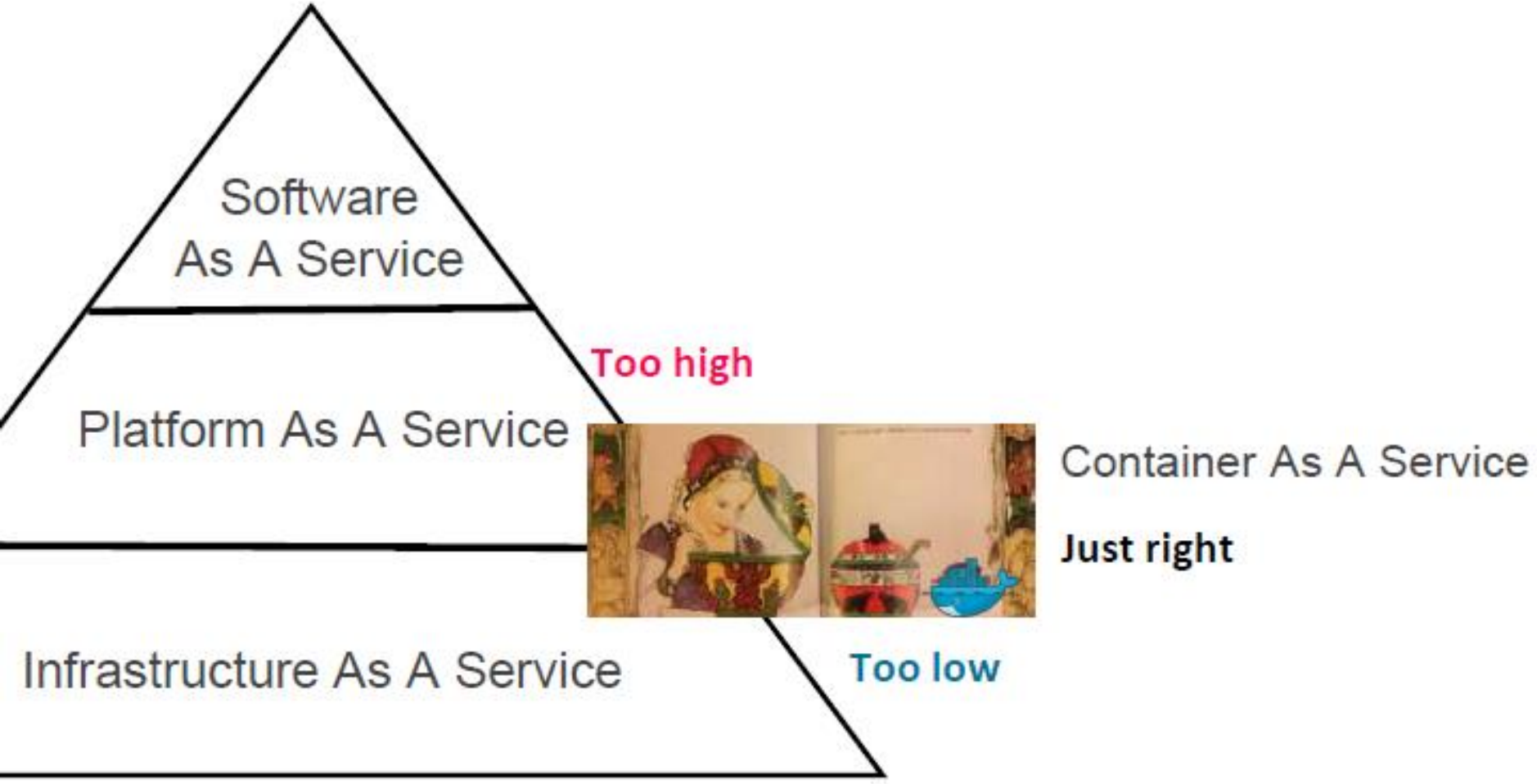
Mar 2013
Docker released to open source

2013
Solomon Hykes starts Docker as an internal project within dotCloud

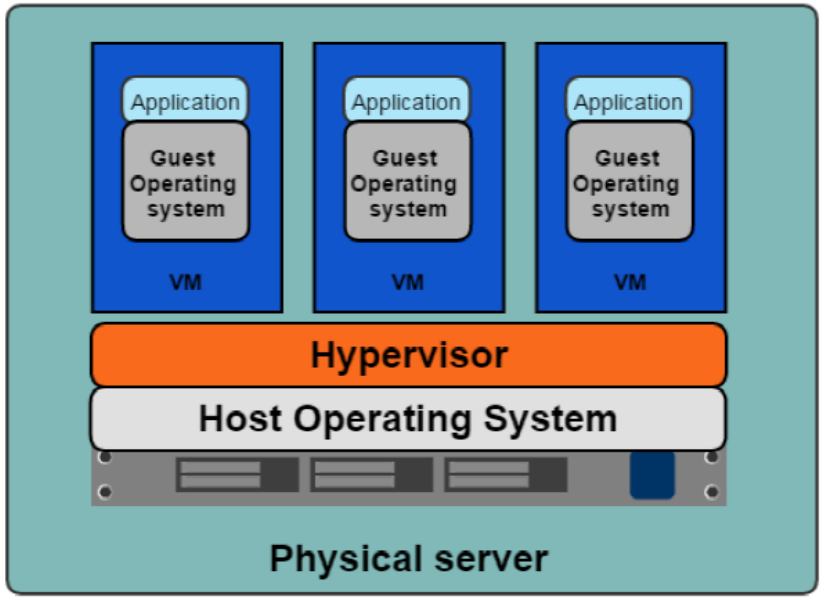
Feb 2016
Docker introduces first commercial product – now called Docker Enterprise Edition

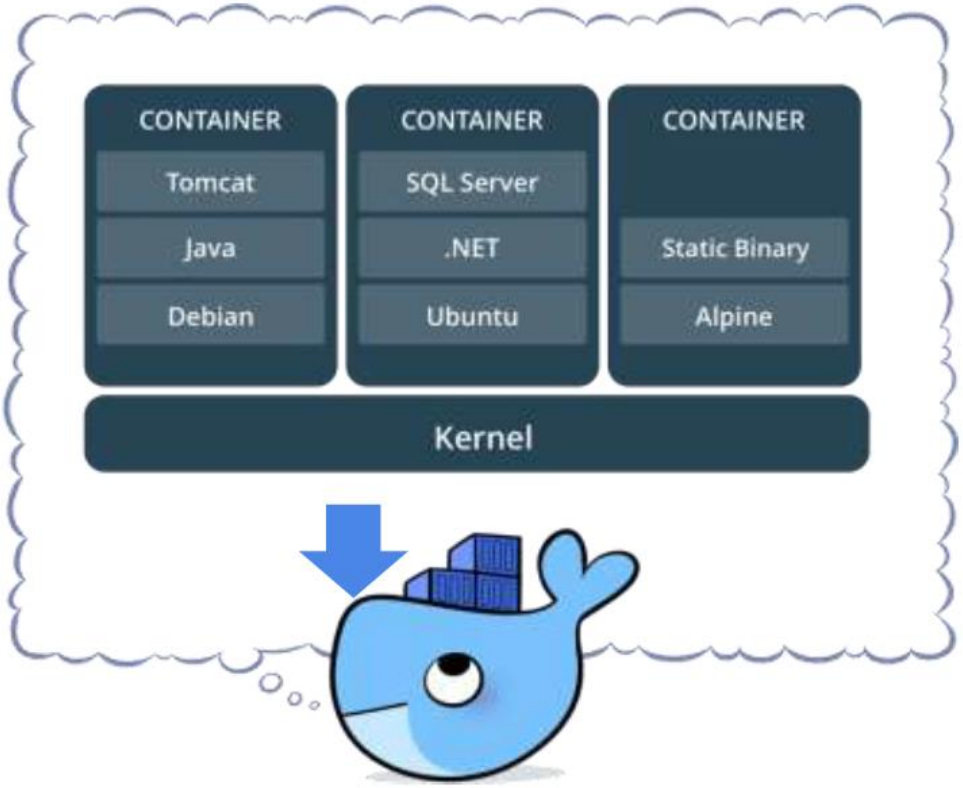
Today
Open source community includes:
- 3,300+ contributors
- 43,000+ stars
- 12,000+ forks

the 4 XaaS


Software
As A Service
Too high
Platform As A Service
Container As A Service
Just right
Infrastructure As A Service
Too low

Cloud vs Docker





Cloud

Pros and Cons of Cloud and VM

PROS

- Better resource pooling
  - One physical machine divided into multiple virtual machines
- Easier to scale
  - Elastically V/H
- VMs in the cloud
  - Rapid elasticity
  - as a service

CONS

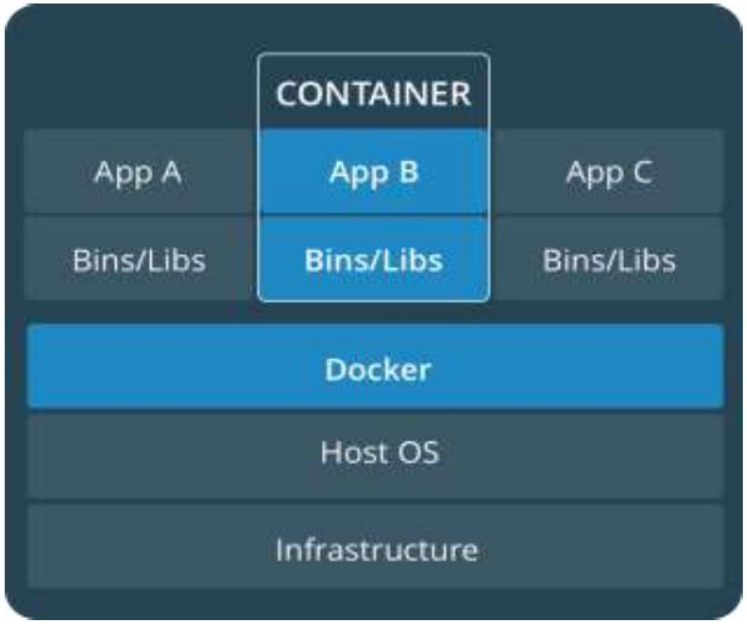
- Each VM stills requires
  - CPU allocation
  - Storage
  - Memory
  - An entire guest operating system
- more VMs, more resources you need
- Guest OS means wasted resources
- Application portability not guaranteed

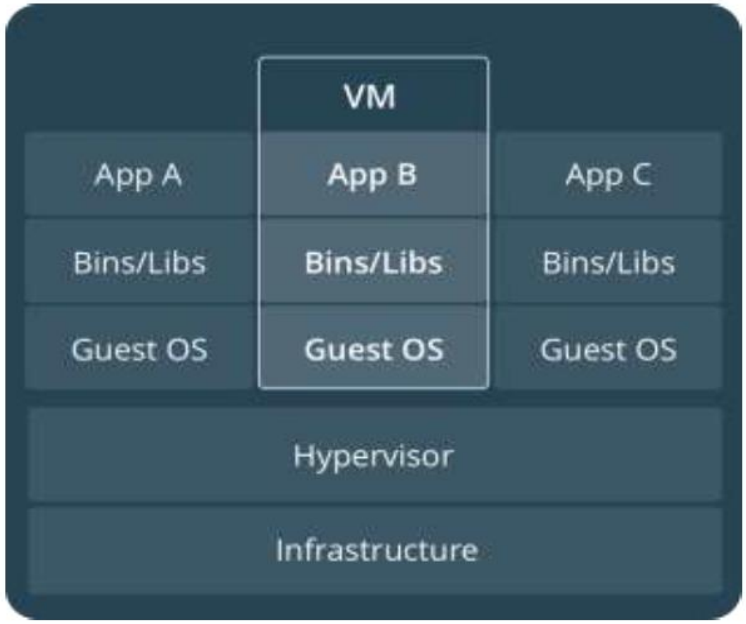
Pros of Containers

- Standardized packaging for software and dependencies
- Isolate apps from each other
- Share the same OS kernel
- Works with all major Linux and Windows Server

- Speed: No OS to boot = applications online in seconds
- Portability: Less dependencies between process layers =
  - ability to move between infrastructure
- Efficiency: Less OS overhead; Improved VM density

Comparing Container and VMs





Containers are an app level construct

VMs are an infrastructure level construct to turn one machine into many servers

Prompt dei comandi

```
Microsoft Windows [Versione 10.0.17134.137]
(c) 2018 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\paolo>docker --version
Docker version 18.06.1-ce, build e68fc7a

C:\Users\paolo>docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
d1725b59e92d: Pull complete
Digest: sha256:0add3ace90ecb4adbf7777e9aacf18357296e799f81cabc9fde470971e499788
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/


C:\Users\paolo>
```

```
C:\Users\paolo>docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
473ede7ed136: Pull complete
c46b5fa4d940: Pull complete
93ae3df89c92: Pull complete
6b1eed27cade: Pull complete
Digest: sha256:29934af957c53004d7fb6340139880d23fb1952505a15d69a03af0d1418878cb
Status: Downloaded newer image for ubuntu:latest
root@a5e45c9653f4:/#
```

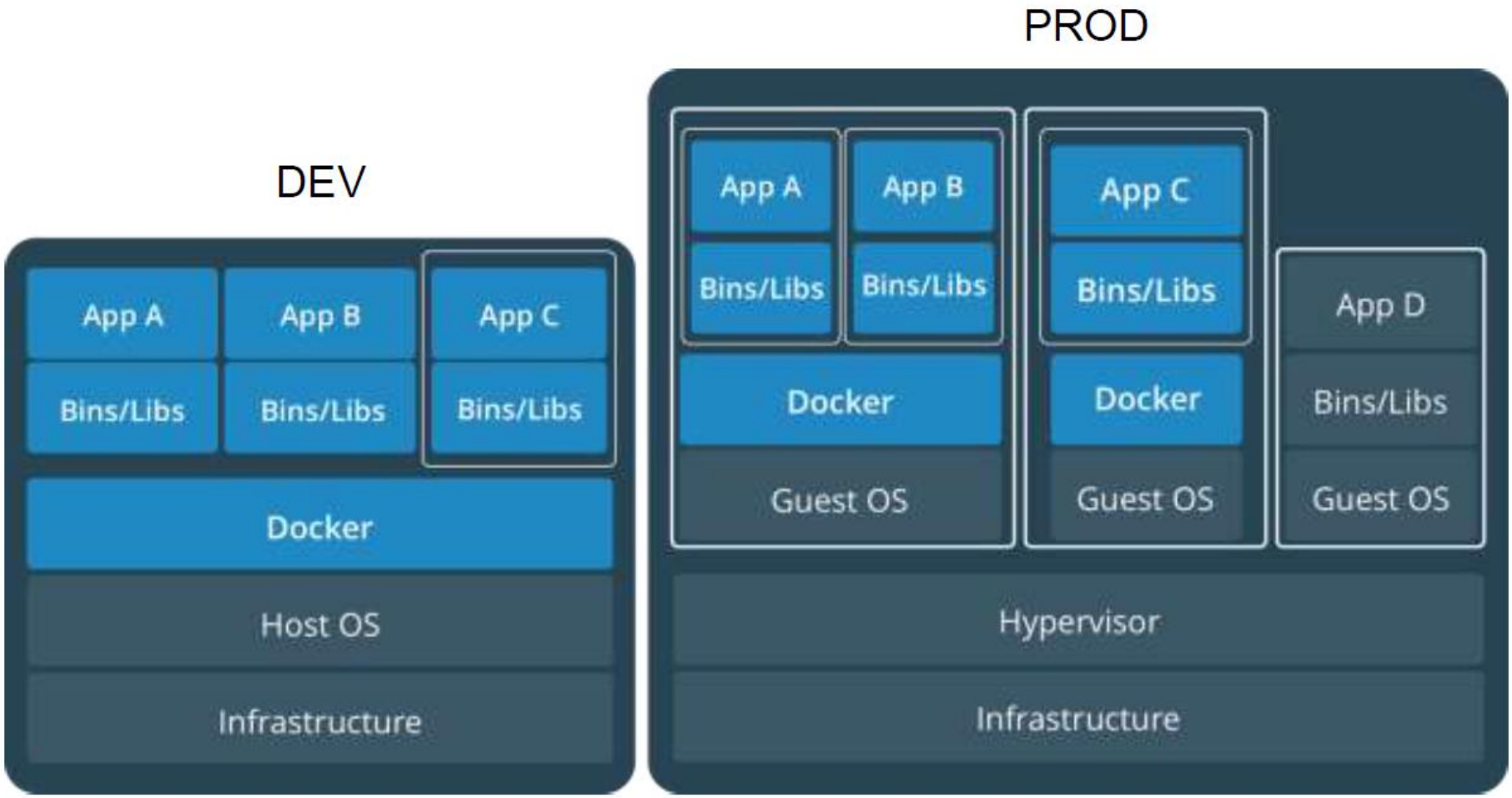
```
root@a5e45c9653f4:/# ls -la
total 72
drwxr-xr-x   1 root root 4096 Oct 29 18:19 .
drwxr-xr-x   1 root root 4096 Oct 29 18:19 ..
-rwxr-xr-x   1 root root    0 Oct 29 18:19 .dockerenv
drwxr-xr-x   2 root root 4096 Oct 18 21:03 bin
drwxr-xr-x   2 root root 4096 Apr 24  2018 boot
drwxr-xr-x   5 root root  360 Oct 29 18:19 dev
drwxr-xr-x   1 root root 4096 Oct 29 18:19 etc
drwxr-xr-x   2 root root 4096 Apr 24  2018 home
drwxr-xr-x   8 root root 4096 Oct 18 21:02 lib
drwxr-xr-x   2 root root 4096 Oct 18 21:02 lib64
drwxr-xr-x   2 root root 4096 Oct 18 21:02 media
drwxr-xr-x   2 root root 4096 Oct 18 21:02 mnt
drwxr-xr-x   2 root root 4096 Oct 18 21:02 opt
dr-xr-xr-x 138 root root    0 Oct 29 18:19 proc
drwx------   2 root root 4096 Oct 18 21:03 root
drwxr-xr-x   1 root root 4096 Oct 19 00:47 run
drwxr-xr-x   1 root root 4096 Oct 19 00:47 sbin
drwxr-xr-x   2 root root 4096 Oct 18 21:02 srv
dr-xr-xr-x  13 root root    0 Oct 29 18:19 sys
drwxrwxrwt   2 root root 4096 Oct 18 21:03 tmp
drwxr-xr-x   1 root root 4096 Oct 18 21:02 usr
drwxr-xr-x   1 root root 4096 Oct 18 21:03 var
root@a5e45c9653f4:/#
```

```
root@a5e45c9653f4:/dev# ls -la
total 4
drwxr-xr-x 5 root root    360 Oct 29 18:19 .
drwxr-xr-x 1 root root   4096 Oct 29 18:19 ..
crw--w---- 1 root tty  136, 0 Oct 29 18:21 console
lrwxrwxrwx 1 root root     11 Oct 29 18:19 core -> /proc/kcore
lrwxrwxrwx 1 root root     13 Oct 29 18:19 fd -> /proc/self/fd
crw-rw-rw- 1 root root   1, 7 Oct 29 18:19 full
drwxrwxrwt 2 root root     40 Oct 29 18:19 mqueue
crw-rw-rw- 1 root root   1, 3 Oct 29 18:19 null
lrwxrwxrwx 1 root root      8 Oct 29 18:19 ptmx -> pts/ptmx
drwxr-xr-x 2 root root      0 Oct 29 18:19 pts
crw-rw-rw- 1 root root   1, 8 Oct 29 18:19 random
drwxrwxrwt 2 root root     40 Oct 29 18:19 shm
lrwxrwxrwx 1 root root     15 Oct 29 18:19 stderr -> /proc/self/fd/2
lrwxrwxrwx 1 root root     15 Oct 29 18:19 stdin -> /proc/self/fd/0
lrwxrwxrwx 1 root root     15 Oct 29 18:19 stdout -> /proc/self/fd/1
crw-rw-rw- 1 root root   5, 0 Oct 29 18:19 tty
crw-rw-rw- 1 root root   1, 9 Oct 29 18:19 urandom
crw-rw-rw- 1 root root   1, 5 Oct 29 18:19 zero
root@a5e45c9653f4:/dev#
```

```
C:\Users\paolo>docker container --all
unknown flag: --all
See 'docker container --help'.

Usage:  docker container COMMAND

Manage containers

Commands:
  attach      Attach local standard input, output, and error streams to a running container
  commit      Create a new image from a container's changes
  cp          Copy files/folders between a container and the local filesystem
  create      Create a new container
  diff        Inspect changes to files or directories on a container's filesystem
  exec        Run a command in a running container
  export      Export a container's filesystem as a tar archive
  inspect     Display detailed information on one or more containers
  kill        Kill one or more running containers
  logs        Fetch the logs of a container
  ls          List containers
  pause       Pause all processes within one or more containers
  port        List port mappings or a specific mapping for the container
  prune       Remove all stopped containers
  rename      Rename a container
  restart     Restart one or more containers
  rm          Remove one or more containers
  run         Run a command in a new container
  start       Start one or more stopped containers
  stats       Display a live stream of container(s) resource usage statistics
  stop        Stop one or more running containers
  top         Display the running processes of a container
  unpause     Unpause all processes within one or more containers
  update      Update configuration of one or more containers
  wait        Block until one or more containers stop, then print their exit codes

Run 'docker container COMMAND --help' for more information on a command.
```

Lo stato da un'altra consolle

```
C:\Users\paolo>docker container ls --all
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
e7aeeed23313        ubuntu              "bash"              47 seconds ago      Up 45 seconds
305a2fd8806b        hello-world         "/hello"            About a minute ago  Exited (0) About a minute ago
a5e45c9653f4        ubuntu              "bash"              6 minutes ago       Exited (127) About a minute ago
db904c68f17c        hello-world         "/hello"            9 minutes ago       Exited (0) 9 minutes ago
```



Containers and VMs together provide a tremendous amount of flexibility for IT to optimally deploy and manage apps.

Docker Basics

Image

The basis of a Docker container. The content at rest.

Container

The image when it is 'running.' The standard unit for app service

Engine

The software that executes commands for containers. Networking and volumes are part of Engine. Can be clustered together.

Registry

Stores, distributes and manages Docker images

Control Plane

Management plane for container and cluster orchestration

Building a Software Supply Chain