BITS Pilani

Cloud Computing

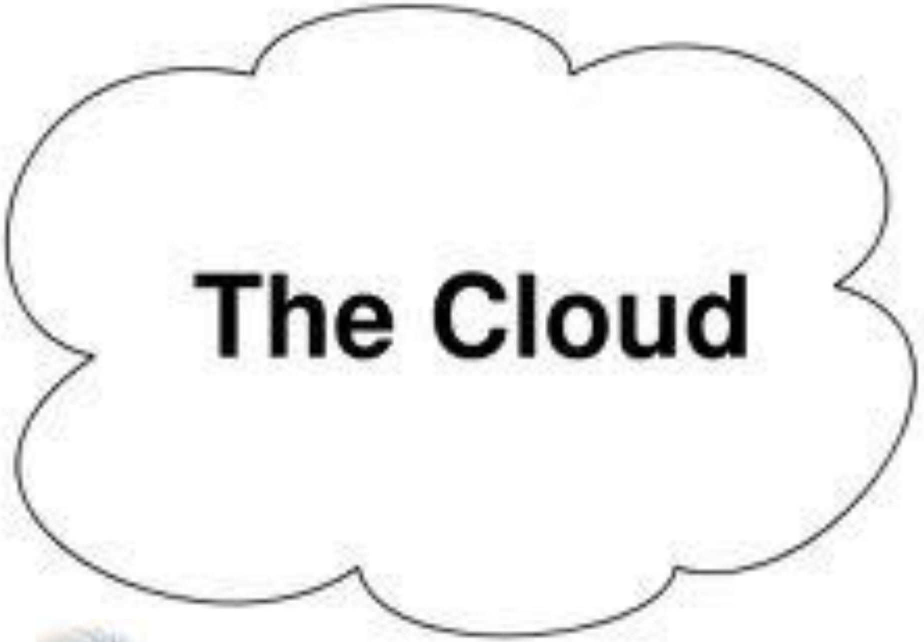
Session 13

Capacity Management & Scheduling in Cloud

Agenda

- Cloud Resource management (CRM)
- Capacity Planning & Resource Scheduling
- Virtual Infrastructure Management (VIM)
- Focus Areas for VIM
    - Distributed management of virtual machines
    - Reservation-based provisioning of virtualized resources
    - Provisioning to meet SLA commitments.
- OpenNebula

Cloud ?  That looks easy!!!

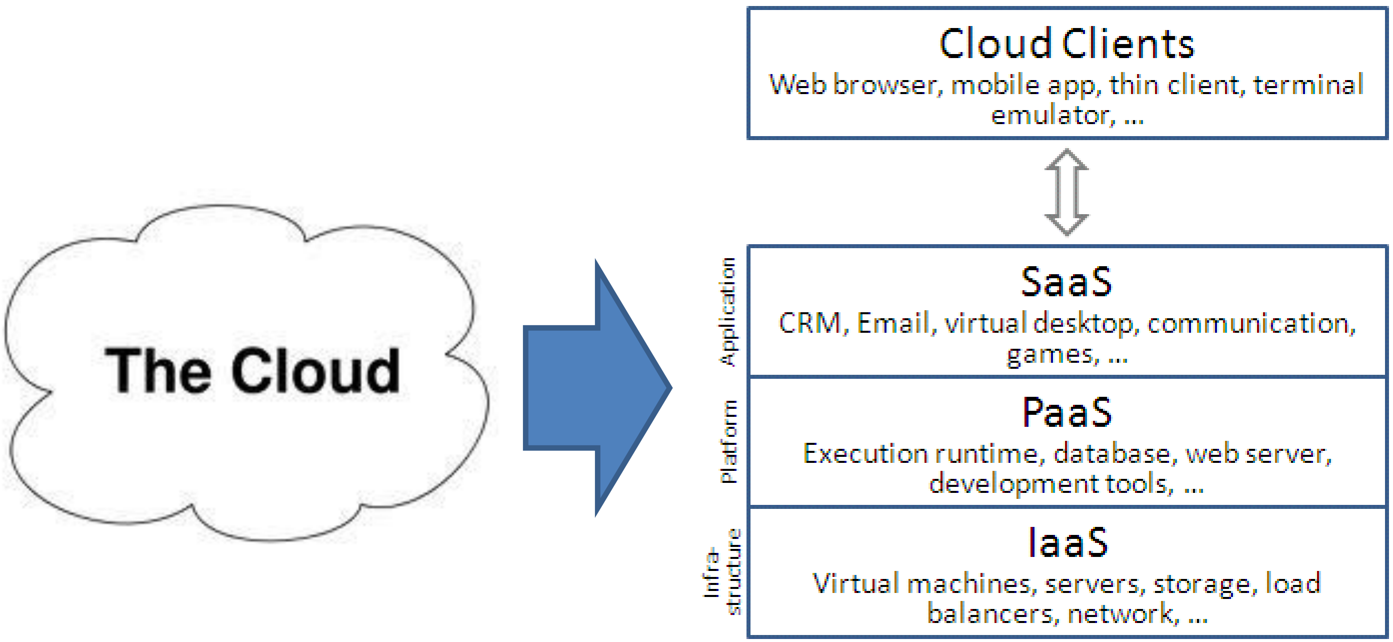


Is it So?????

Introduction

So CLOUD requires managing……

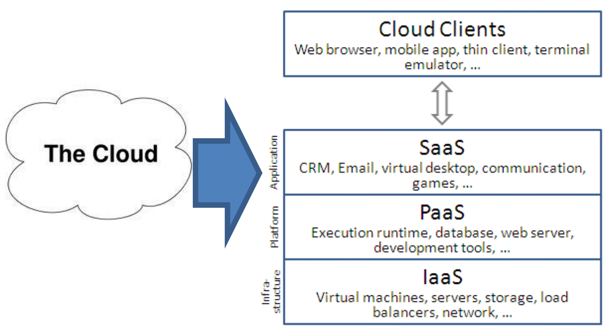
But what will you manage

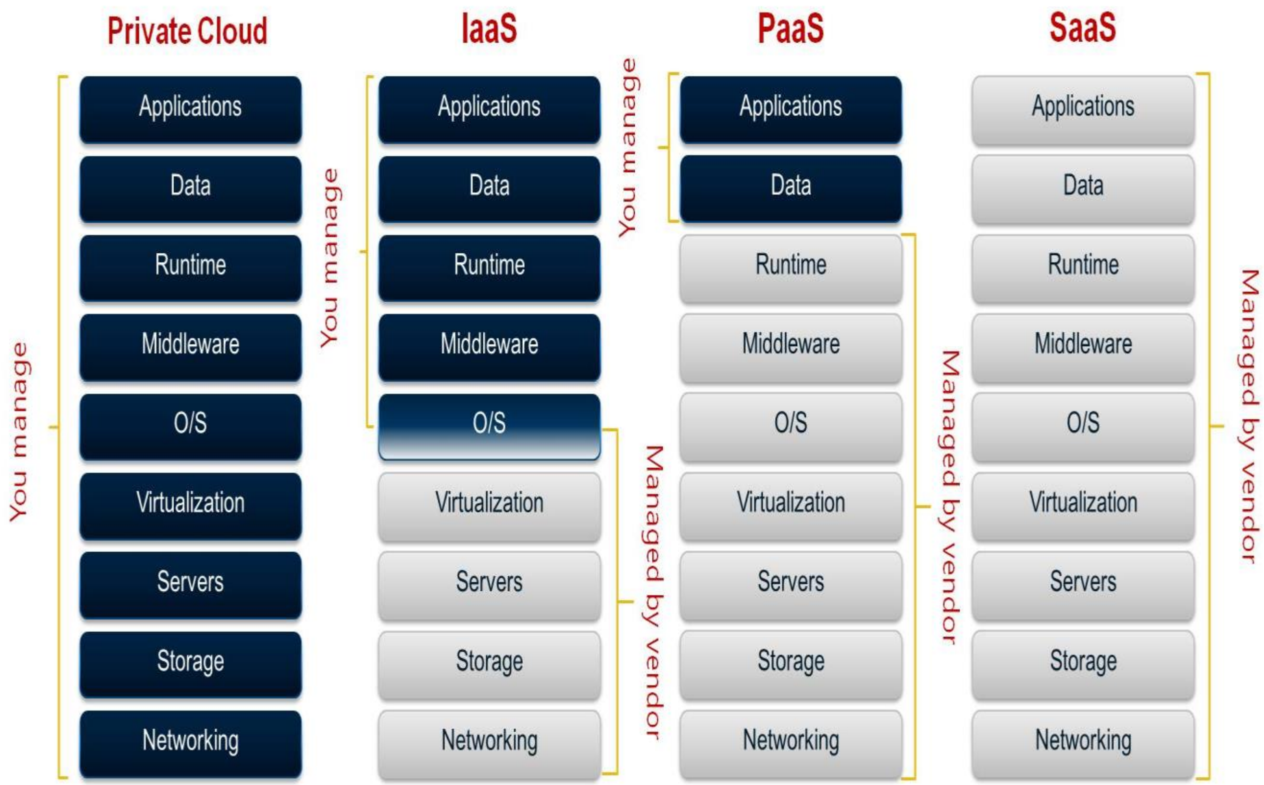


Introduction

Ok, ok got to know what to manage…..

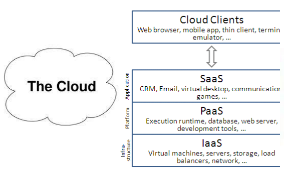
Is it so……….????

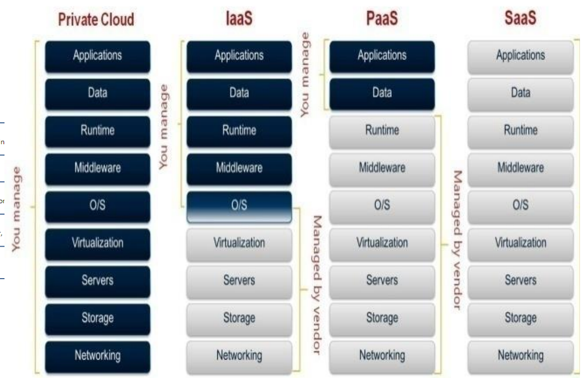




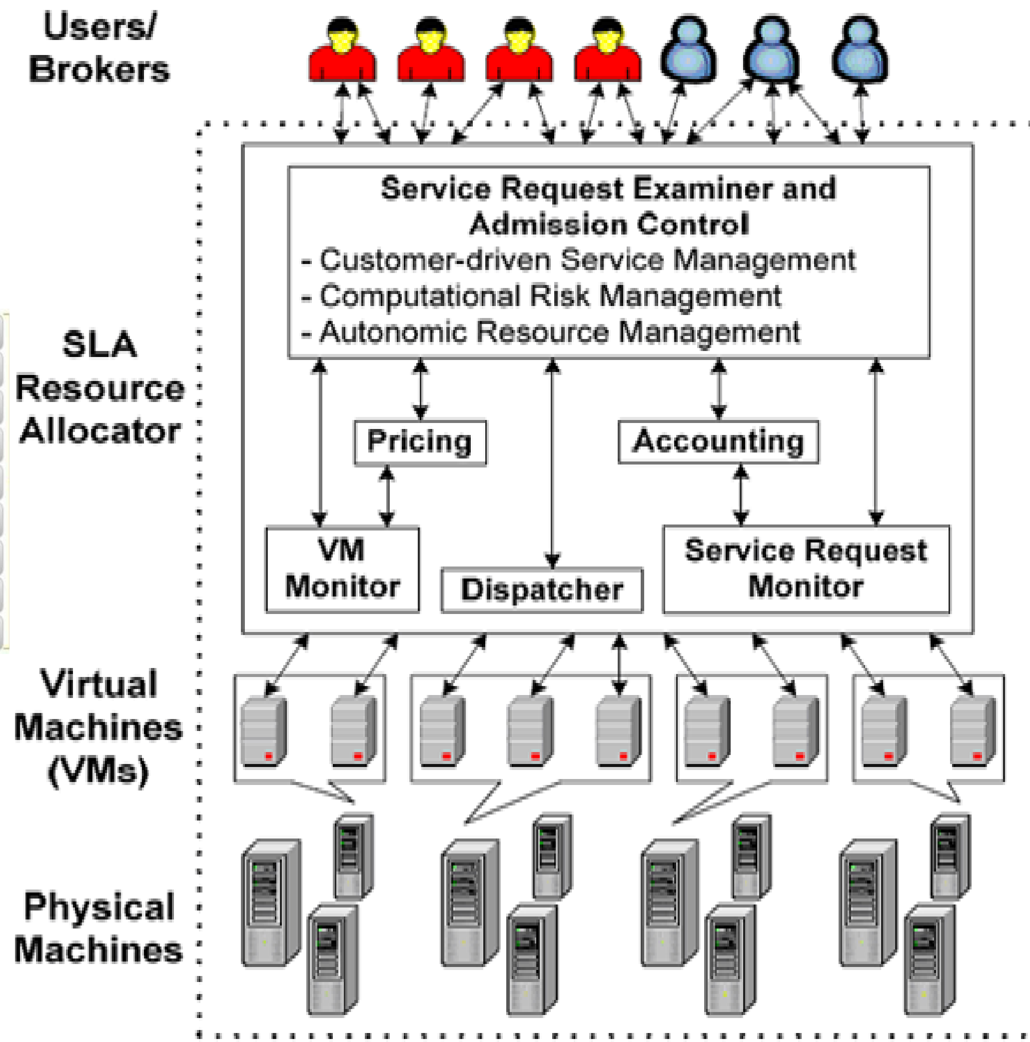
Resource Management

OMG!!!!





Therefore the key requirement for cloud architecture is “**efficient management**” of resources at all the three layers of cloud stack



Need for Cloud Resource Management

https://www.bacancytechnology.com/blog/cloud-resource-management

- Functionality, Performance, Cost, Security
- Unpredictable situations: E.g., system failures, security attacks, fluctuating loads

Capacity Planning

- How much workload the system can handle and maintain ?
- No direct goals on improving the efficiency.
- Determines whether the system is working properly
- Measure their performance
- Determine the usage of patterns
- Predict future demand of cloud-capacity.

Image courtesy: Bytebytego

Steps of Capacity Planning

1. Determine the distinctiveness of the present system.
2. Determine the working load for different resources
3. Load the system until it gets overloaded
4. Predict the future based on older statistical reports
5. Deploy resources to meet the predictions & calculations.

Repeat steps 1 to 5 in a loop – as required.

Capacity Versus Performance

Images - Microsoft Designer

Capacity: How much workload a system can hold ?

Performance: How a system performs a given task or handle the work load. E.g: Throughput, Response time, Availability, Reliability

Scheduling

- Decide how to allocate resources of a system
- E.g: Memory, CPU cycles, I/O and network bandwidth, between users and tasks.
- Deciding factors
  - Policies and mechanisms
  - **Policy**: Principles guiding decisions.
  - **Mechanisms**: The means to implement policies.

CRM Policies

1. **Admission Control:** Aims to prevent the system from accepting work load in violation of high level system policies

2. **Capacity Allocation:** Allocate resources for individual instances /servers.

3. **Load balancing:** Even distribution of the load among available servers or instances

Cloud Resource Management Policies

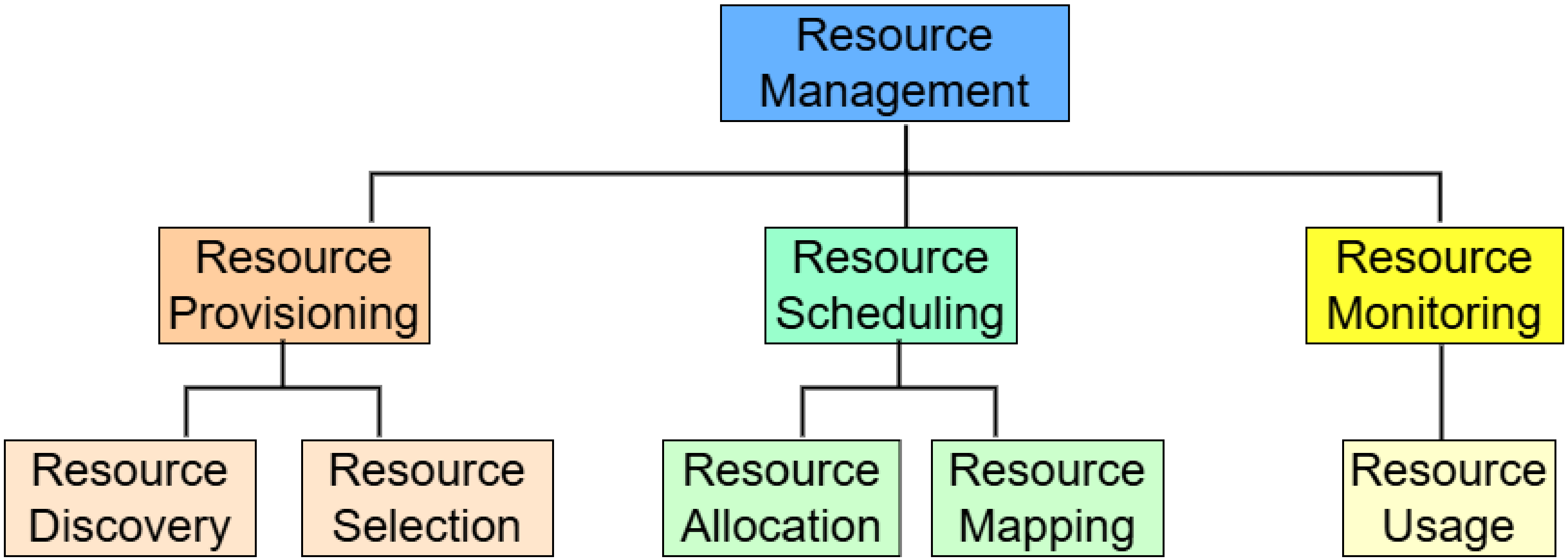
4. Energy optimization: Minimize energy consumption

Minimization of the cost to provide cloud service

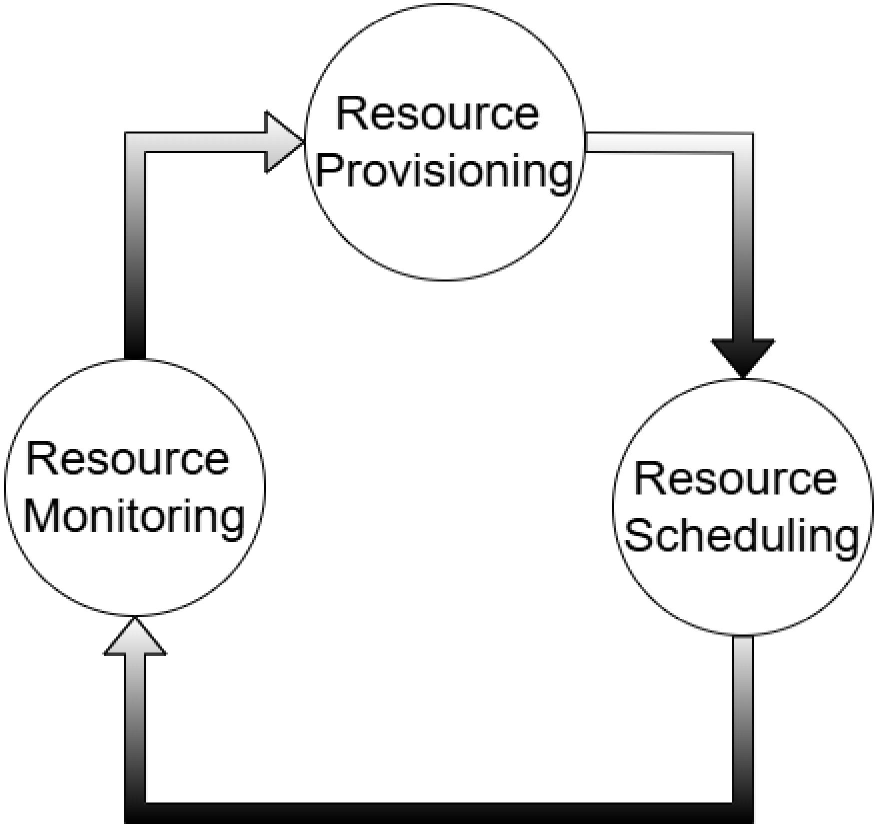
Load balancing and energy optimization are interlinked. i.e. Minimize the number of servers to the smallest possible

5. QoS Guarantees: The ultimate service guarantees, criteria set by Service Level Agreements (SLA).

Effective Resource Management



Effective Capacity Planning



Closed Loop Automation Framework

Integration considerations: Seamless integrations of multiple cloud services or transitions between on-premises and cloud solutions without resource conflicts.

Feedback loops: Continuously revisit and adjust capacity plans

Effective Capacity Planning & CRM

1. **Demand forecasting**: Analyze historical data and current trends to anticipate future cloud resource needs.
2. **Performance analysis**: Monitor resource performance, guiding decisions on scaling needs.
3. **Resilience**: Prepare for unexpected events (from traffic spikes to system failures)
4. **Cost management**: Understand both direct and indirect costs of cloud resources to optimize spending without compromising performance.

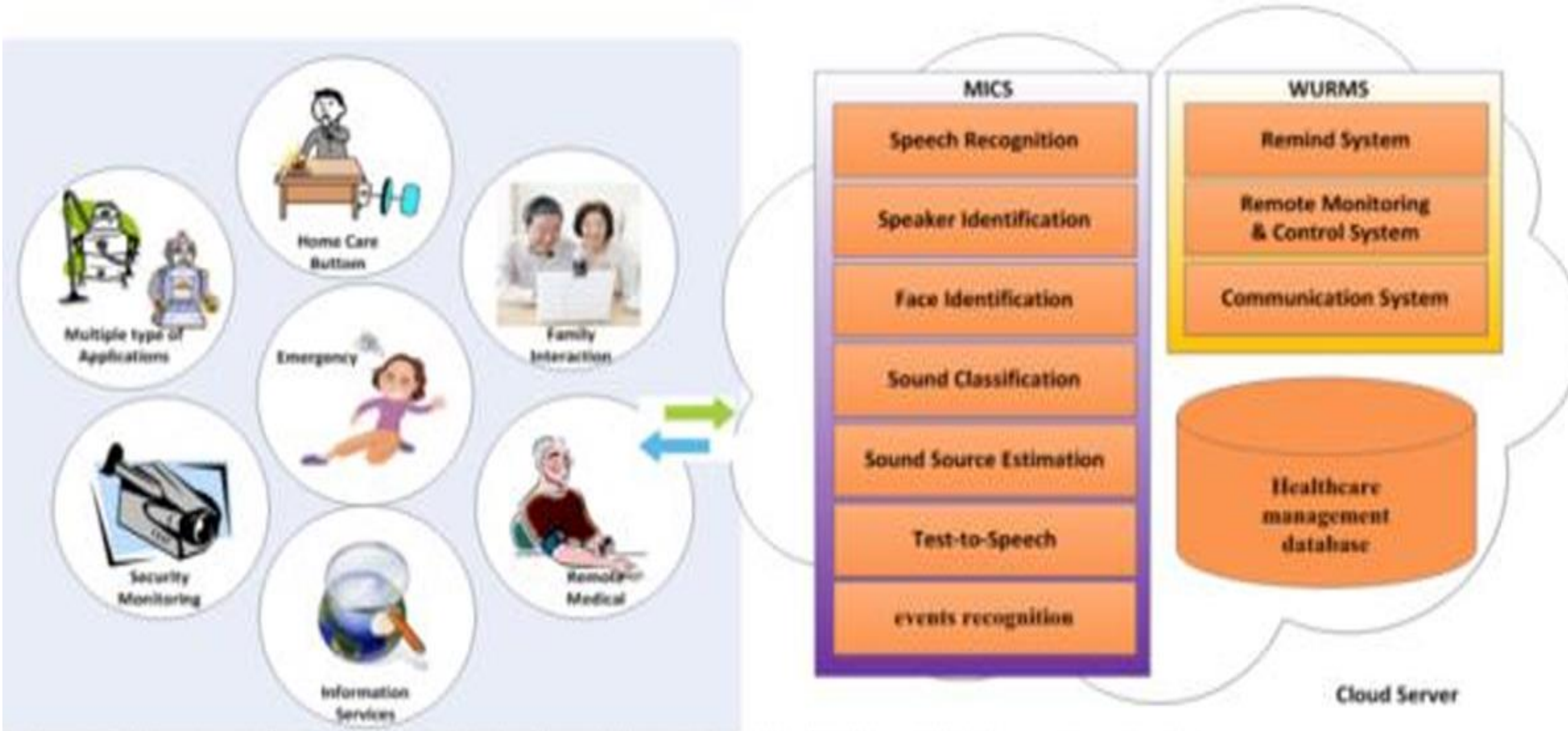
- Task scheduling and VM placement to resource allocation in Cloud computing: challenges and opportunities | Cluster Computing – 2023
- Effective Resource Management through VM Allocation in Cloud Data Center | Proceedings of the 18th Innovations in Software Engineering Conference - 2025

Automating HIPAA-Compliant Infrastructure on AWS – Case Study

Healthcare organizations are increasingly relying on cloud platforms to store, manage, and deliver critical patient services, yet they face a complex challenge: balancing **security**, **scalability**, **cost-efficiency**, and **regulatory compliance**. As patient data must be protected under strict HIPAA regulations, any misconfiguration in cloud resources can lead to serious privacy breaches and legal consequences. At the same time, these systems must remain highly available and performant to support services like Electronic Health Records (EHRs), telemedicine, and real-time diagnostics—especially during high-demand periods.

1. HIPAA: Health Insurance Portability and Accountability Act

Automating HIPAA-Compliant Infrastructure on AWS



Example **Ubiquitous Healthcare System (UHS) framework** from [A Comparative Analysis of Various Healthcare Frameworks in Cloud Environment]

Healthcare Case Study – Challenges & Focus Areas

- Need for scalable resources for peak hours (video consults)
- Avoiding over-spend on idle compute
- HIPAA compliance requirements

- **Performance:** Fast access to patient data, scalable health portals
- **Cost:** Optimizing compute/storage for large imaging data
- **Automation:** Auto-scaling based on patient traffic
- **Security:** HIPAA[1] compliance, encryption, auditing

Healthcare Case Study - Solutions

- Auto scaling EC2 for video consults
- Lambda for serverless appointment scheduling
- Encrypted S3 for medical records, RDS with encryption
- CloudTrail, AWS Config, and Shield for audit and security

Need of the hour???

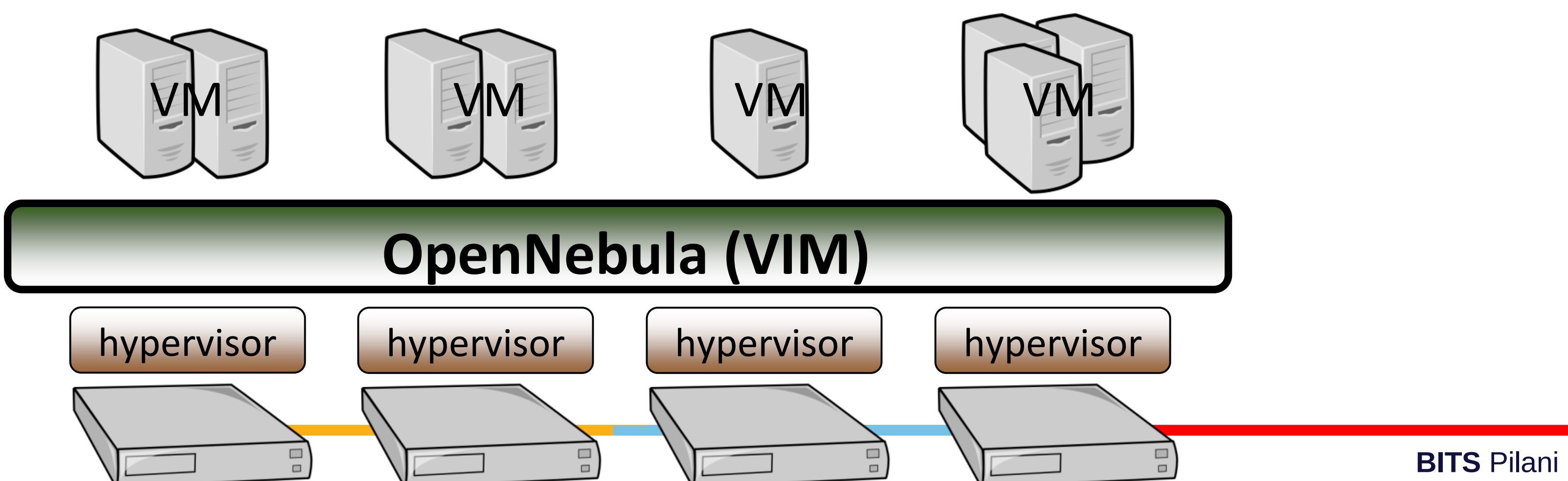
Virtual Infrastructure Managers

Need for Automation

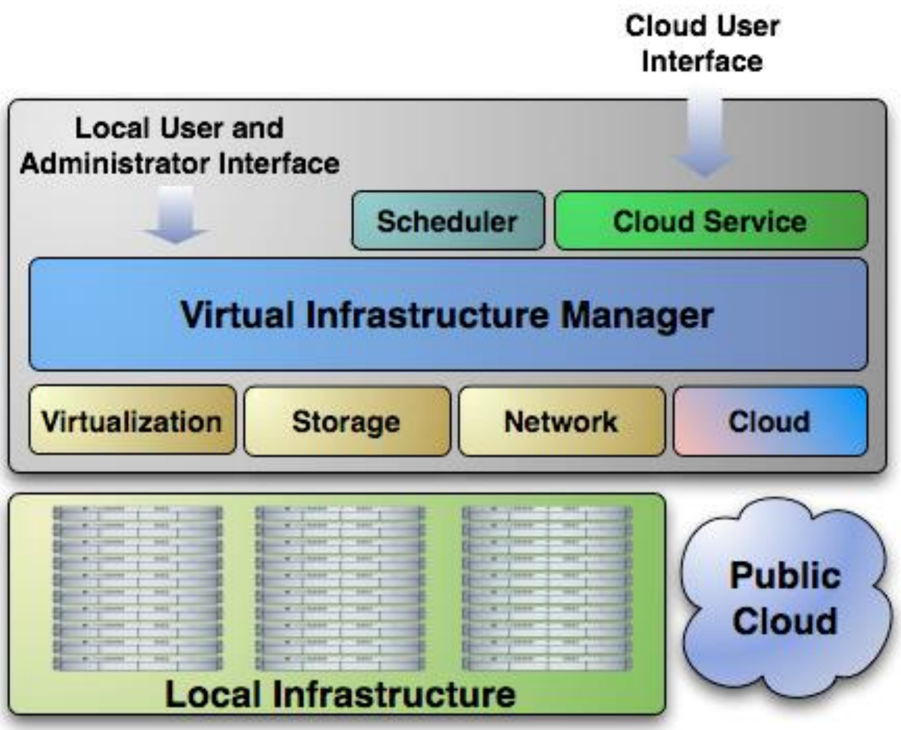
- Cloud **<u>Distributed environment</u>**
  - With large scale of systems to manage
  - Support of multi-tenancy
  - Management to maintain SLAs
- So there is need for **<u>automation</u>** to replace manual operations and to reduce overall cost

Why a Virtual Infrastructure Manager?

- VMs are great!!...but something more is needed
  - Where did/do I put my VM? (***scheduling & monitoring***)
  - How do I provision a new cluster node? (***clone***)
  - What IP addresses are available? (***networking)***
- Provide a ***uniform view*** of the resource pool
- ***Life-cycle management*** and monitoring of VM
- The VIM should ***integrate*** Image, Network, and Virtualization



Extending the Benefits of Virtualization to Clusters



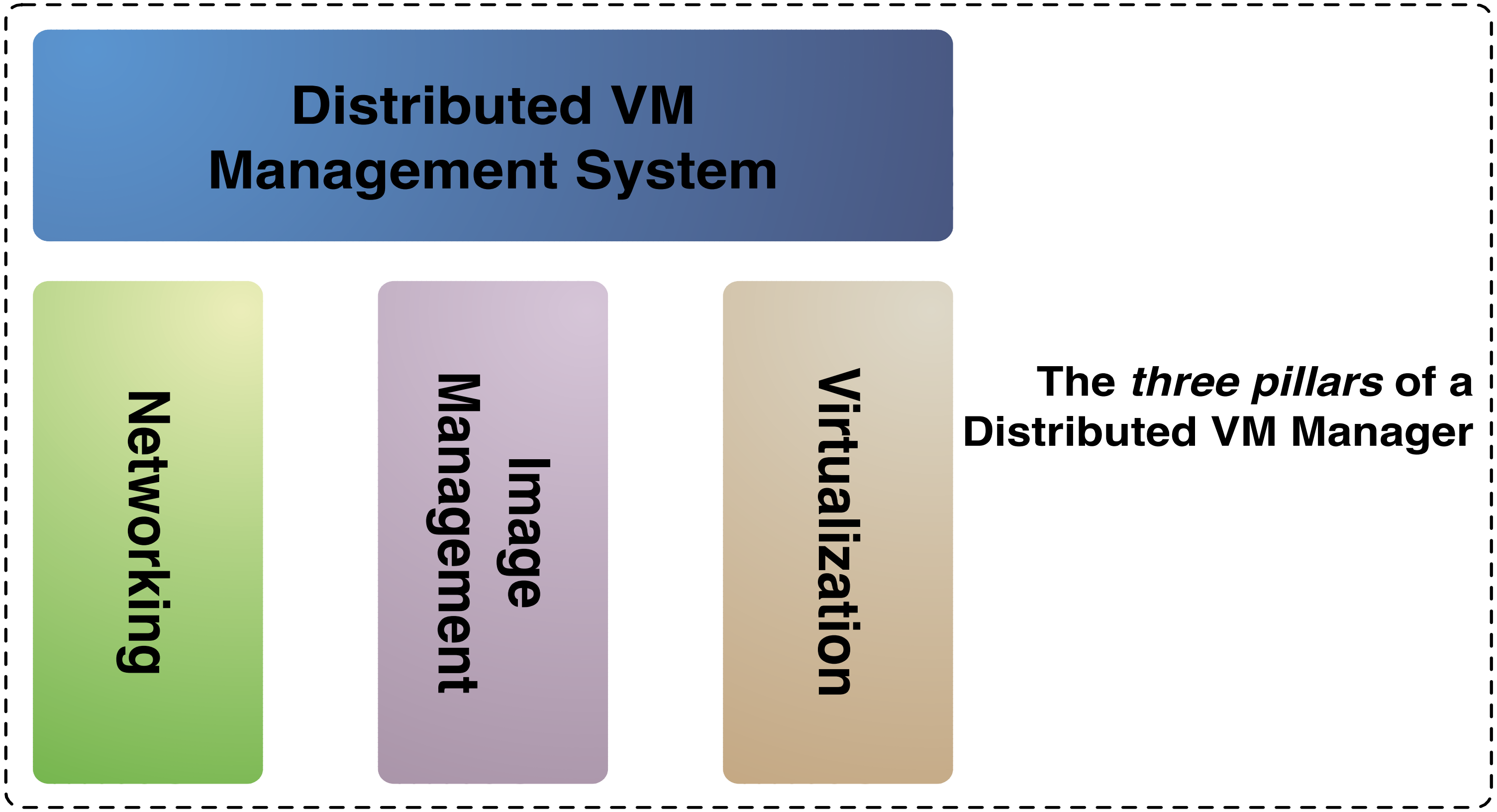
- Dynamic deployment and re-placement of virtual machines on a pool of physical resources
- Transform a rigid distributed physical infrastructure into a flexible and agile virtual infrastructure

Challenges for VIM

- VMs require a fair amount of configuration
  - Preparation of the VM's software environment and network configuration.
- Need to configure groups of VMs that will provide a specific service. E.g: An application requiring a Web server and a database server.
- Configuration must be done on-the-fly,
  - Little time between the time the VMs are requested and the time they are available to the users.
- VIM must also be capable of allocating resources efficiently
  - Energy optimization
  - Save operational costs
  - Reacting to changes in the physical infrastructure.

Virtual Machine Management Model

Distributed VM Management Model



Resource Provisioning in Private Clouds

VIM in **private clouds** has to deal with an additional problem:
Enough Resources ? Provide the illusion of "infinite capacity." ?

Courtesy: Capacity Testing: Explained with Types

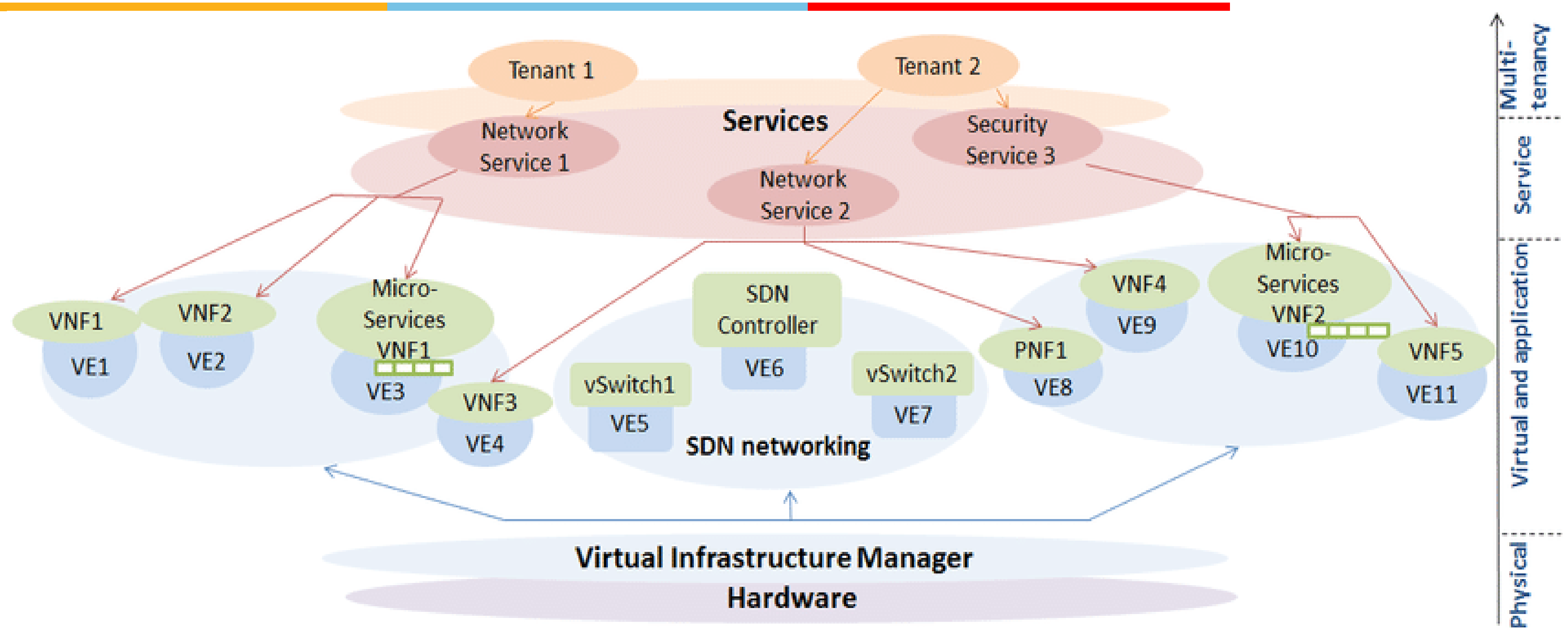
- The immediate provisioning scheme used in public clouds, is **ineffective in private clouds**.
- Support for additional provisioning schemes:
- **Best-effort provisioning** and
- **Advance reservations** to guarantee quality of service (QoS).

Text content courtesy: PowerPoint Presentation

Key Focus Areas

1. Distributed management of virtual machines
2. Reservation-based provisioning of virtualized resources
3. Provisioning to meet SLA commitments.

Distributed Management of Virtual Machines



- Managing VMs in a pool of distributed physical resources is a key concern in IaaS clouds
- Requires the use of a virtual infrastructure manager

S. Cherrared, S. Imadali, E. Fabre, G. Gössler and I. G. B. Yahia, "A Survey of Fault Management in Network Virtualization Environments: Challenges and Solutions," in *IEEE Transactions on Network and Service Management*, vol. 16, no. 4, pp. 1537-1551, Dec. 2019, doi: 10.1109/TNSM.2019.2948420.

Distributed Management of Virtual Machines

- **VI managers must orchestrate the tasks efficiently**
  - The need to set up custom software environments for VMs
  - Setting up and managing networking for interrelated VMs
  - Reducing the various overheads involved in using VMs

- **Traditional:** VM-based resource scheduling follows a **static approach E.g: Greedy technique**
- **VI managers** must be able to support **flexible and complex scheduling policies**
- Must L*everage* (use) the ability of VMs to suspend, resume, and migrate.

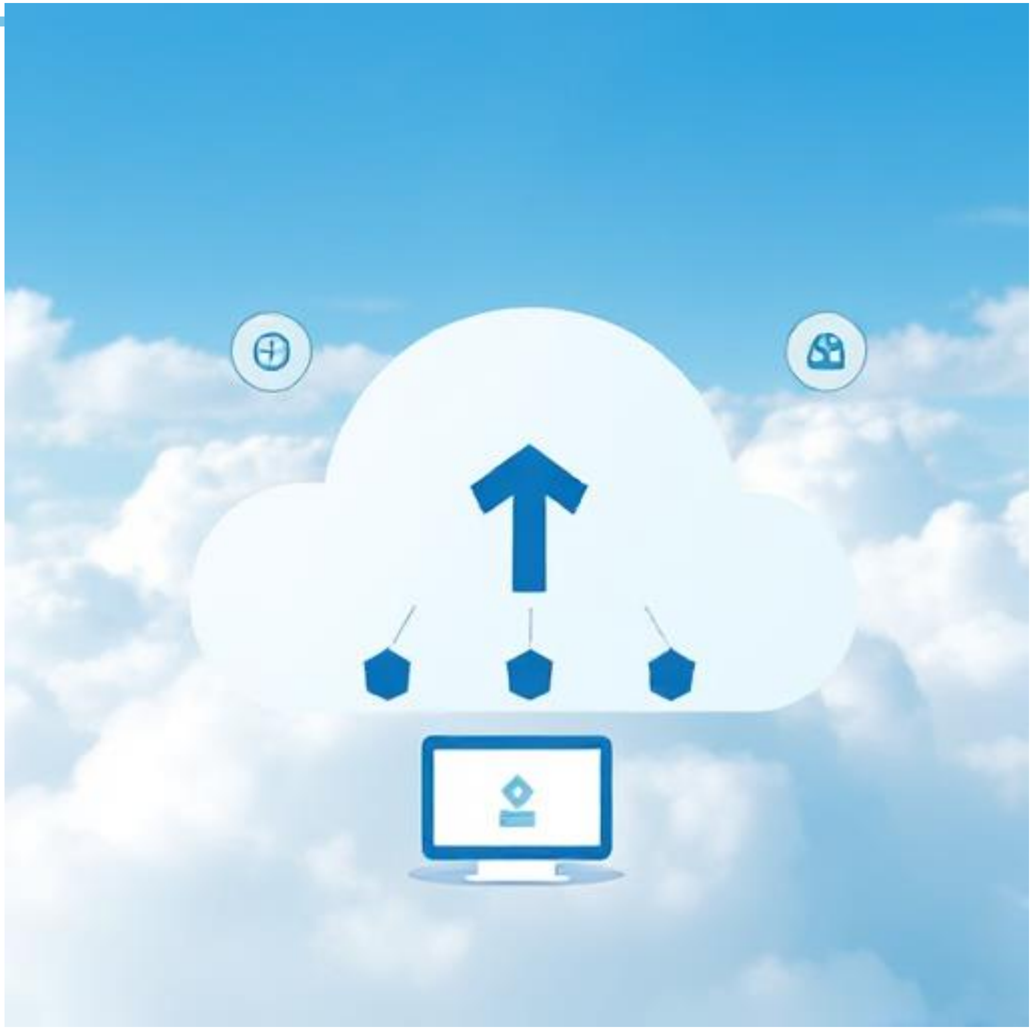
Reservation-Based Provisioning of Virtualized Resources

- How to deal with situations where the demand for resources is known beforehand
- Can reservation be suffice ?
  - Looks simple. But is it really ?
  - It is known to cause resources to be underutilized, due to the difficulty of scheduling other requests around an inflexible reservation.
- When dealing with **finite capacity**, a different approach is needed.

Provisioning to meet SLA commitments

Key Role Players

1. Cloud Provider
2. Service Owner
3. End User



Service Level Agreement (SLA) Commitments

Mutual consent on certain metrics/guarantees with respect to service

E.g: Application specific performance – serving time, throughput

Image courtesy: [Understanding Cloud Service Level Agreements (SLAs): Types, Elements, and Best Practices - Nhu Aqt]

Provisioning to meet SLA commitments

- **Cloud Provider:** Provides cloud services
- **Cloud Consumer** (i.e., the **service owner**; for instance, the company that develops and manages the applications)
- **End users: Service users** E.g: Users that access the applications

- **Service Level agreements (SLAs) between cloud provider and service owner**
  - Requirements are formalized in infrastructure SLAs between the service owner and cloud provider
- **SLAs between service owner and service user**
  - Covering guarantees such as the timeliness with which these services will respond.

Text content courtesy: PowerPoint Presentation

Provisioning to Meet SLA Commitments

- Problems with the service owner
    - Not resourceful enough to perform an exact service sizing OR
    - Service workloads are hard to anticipate in advance.
- So, to protect high-level SLAs, the cloud provider should cater for **elasticity on demand**.
- Resource allocation decisions are **application-specific,** driven by the **application level metrics**
- **Scaling and de-scaling** of an application is best managed by the application itself.

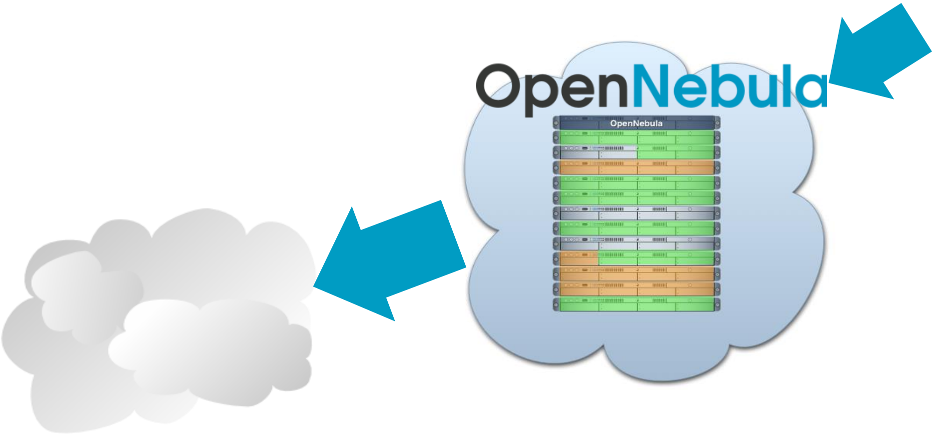
Provisioning to Meet SLA Commitments

- The elasticity of the application should be contracted and formalized as part of the SLA between the **cloud provider** and **service owner**
- **Two Key issues to focus**
  - SLA-oriented capacity planning to guarantee service elasticity.
  - Continuous resource placement and scheduling optimization
    - Lowers operational costs
    - Takes advantage of available capacity transparently to the service

What is OpenNebula?

[What Is a Nebula? | NASA Space Place – NASA Science for Kids]

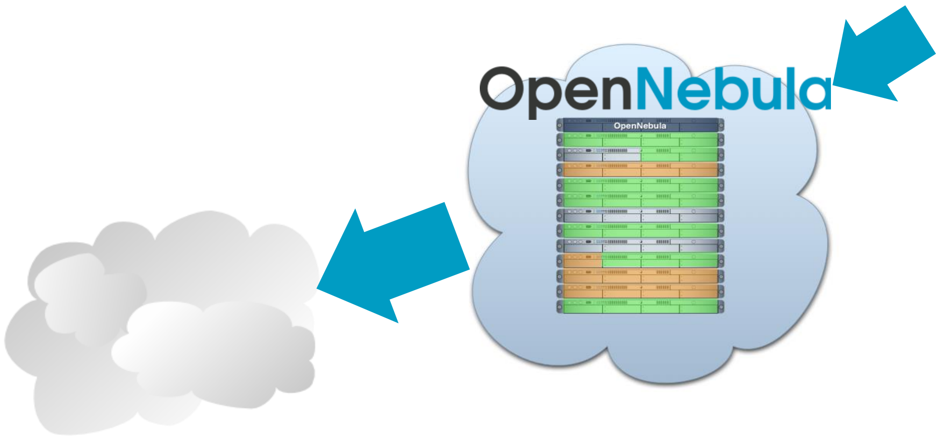
What is OpenNebula?



- Open source cloud computing platform
- Private cloud to simplify and optimize internal operations
- Hybrid cloud to supplement the capacity of the Private Cloud
- Public cloud to expose needed private services to external users

[Courtesy: OpenNebula – Open Source Cloud & Edge Computing Platform]

What is OpenNebula?



- Manages distributed heterogenous data center infrastructures
- Automatic provision and elasticity to offer
- On-demand applications and services on enterprise
- Hybrid and edge environments
- **OpenNebula 6.10 'Bubble'** is the latest version

[Courtesy: OpenNebula – Open Source Cloud & Edge Computing Platform]()

Distributed Management of Virtual Infrastructures

- OpenNebula is capable of managing groups of interconnected VMs
  - Support for the Xen, KVM, and VMWare platforms
  - Data centers and private clouds that involve a large amount of virtual and physical servers.

VM Attributes in OpenNebula

- The primary target of OpenNebula is to manage VMs.
- Within OpenNebula, a VM is modeled with
  - A capacity in terms of memory and CPU.
  - A set of NICs attached to one or more virtual networks.
  - A set of disk images[1].
  - A state file (optional) or recovery file that contains the memory image of a running VM plus some hypervisor-specific information.

[1] In general it might be necessary to transfer some of these image files to/from the physical machine the VM will be running in.

VM Life Cycle in OpenNebula

1. **Resource Selection**
   - Set the resources for VM
   - Done by the Scheduler: **Rank scheduling policy** is the default one –
     - Allows site administrators to configure the scheduler to prioritize the resources that are more suitable for the VM
   - Can use **Haizea lease manager**[1] to support more complex scheduling policies.
2. **Resource Preparation**
   - The disk images of the VM are transferred to the target physical resource.
     - The VM is **contextualized** during the boot process.
     - E.g: Setting up the network and the machine hostname, or registering the new VM with a service (e.g., the head node in a compute cluster)

1. http://haizea.cs.uchicago.edu/

Leasing in Haizea

- **Lease:**
- A negotiated and renegotiable agreement between a resource provider and a resource consumer
- The former agrees to make a set of resources available to the latter, based on a set of lease terms presented by the resource consumer.
- E.g: Hardware resources, software resources, and availability period

Three Types

1. **Advanced reservation leases:** Resources must be available at a specific time.
2. **Immediate leases:** Resources are provisioned when requested or not at all.
3. **Best-effort leases:** Resources are provisioned as soon as possible and requests are placed on a queue if necessary.

1. http://haizea.cs.uchicago.edu/

VM Life Cycle in OpenNebula

3. VM Creation

- The VM is booted by the resource hypervisor.

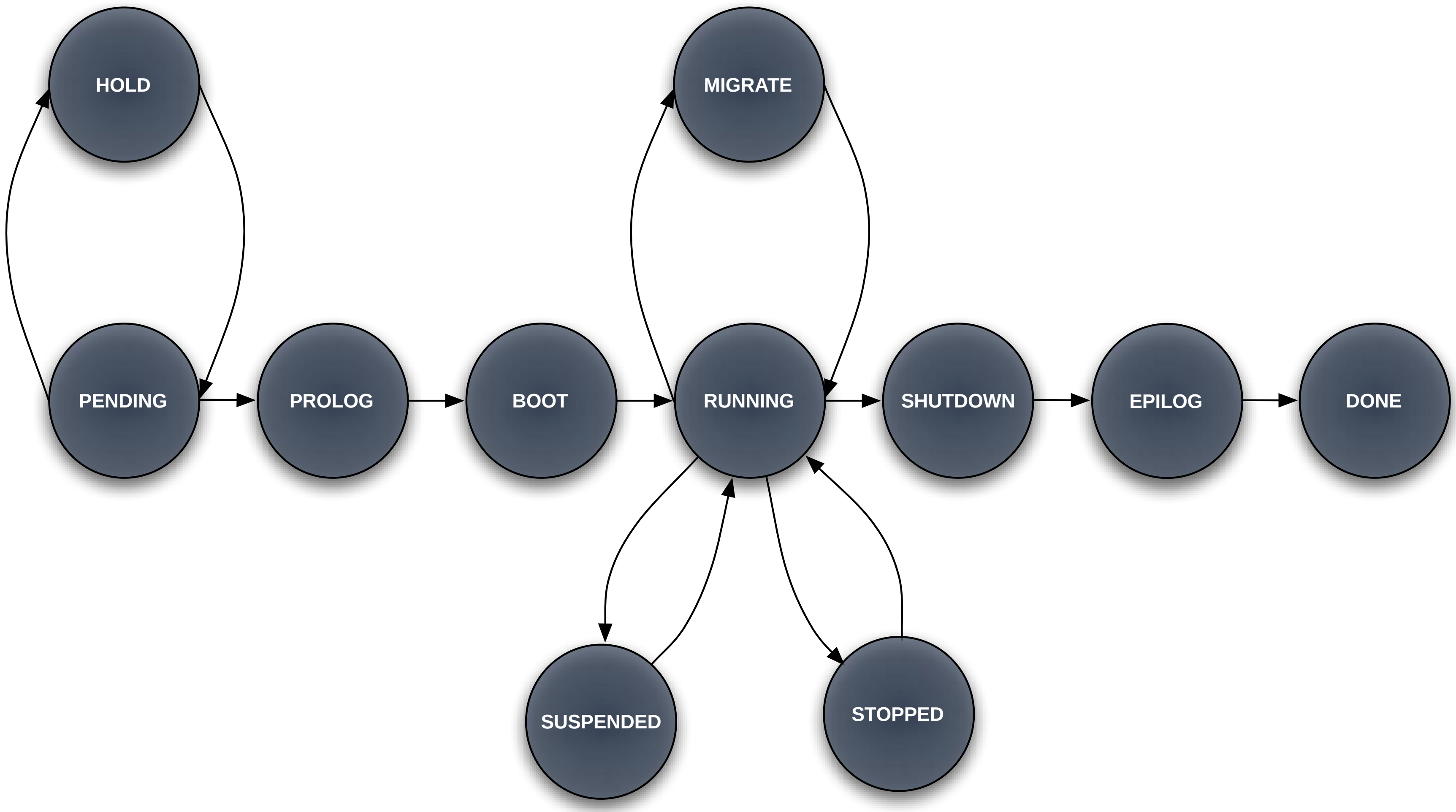
4. VM Migration – On Demand

- The VM potentially is migrated to a more suitable resource (e.g., to optimize the power consumption of the physical resources).

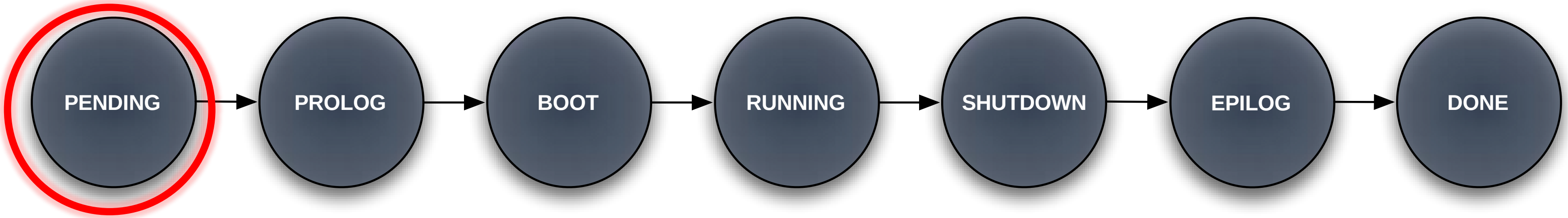
5. VM Termination

- VM gets shut down. OpenNebula can transfer back its disk images to a known location. So, changes in the VM can be kept for a future use.

VM States

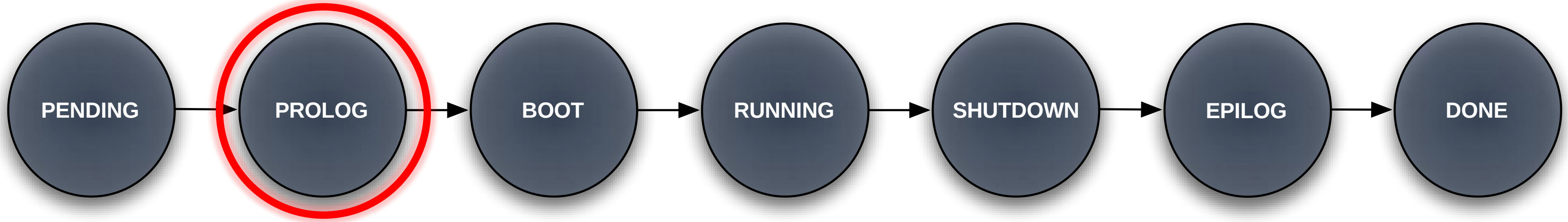

HOLD
MIGRATE
PENDING
PROLOG
BOOT
RUNNING
SHUTDOWN
EPILOG
DONE
SUSPENDED
STOPPED

VM - Pending State



- After submitting a VM description it is added to the database and its state is set to PENDING.
- In this state IP and MAC addresses are also chosen if they are not explicitly defined.
- The scheduler awakes every 30 seconds and looks for VM descriptions in PENDING state and searches for a physical node that meets its requirements.

Prolog State

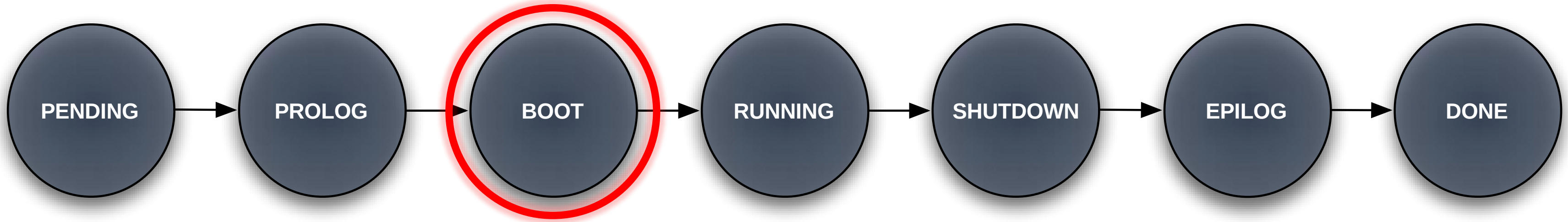


In PROLOG state the images to be used by the VM are prepared

Transfer actions:

- **CLONE**: Makes a copy of a disk image file to be used by the VM. If Clone option for that file is set to false and the Transfer Driver is configured for NFS then a symbolic link is created.
- **MKSWAP**: Creates a swap disk image on the fly to be used by the VM if it is specified in the VM description.

Boot State

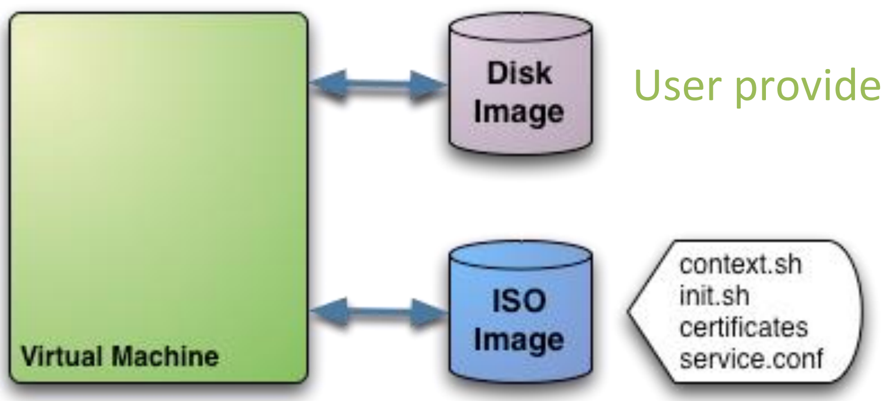


- In this state a **deployment file specific is generated** using the information provided in the VM description file.
- Then VM driver sends deploy command to the virtual host to start the VM.
- The VM will be in this state until deployment finishes or fails.

Contextualization

The ISO image has the contextualization for that VM:

- **context.sh**: contains configuration variables
- **init.sh**: script called by VM at start to configure specific services
- **certificates**: directory that contains certificates for some service
- **service.conf**: service configuration

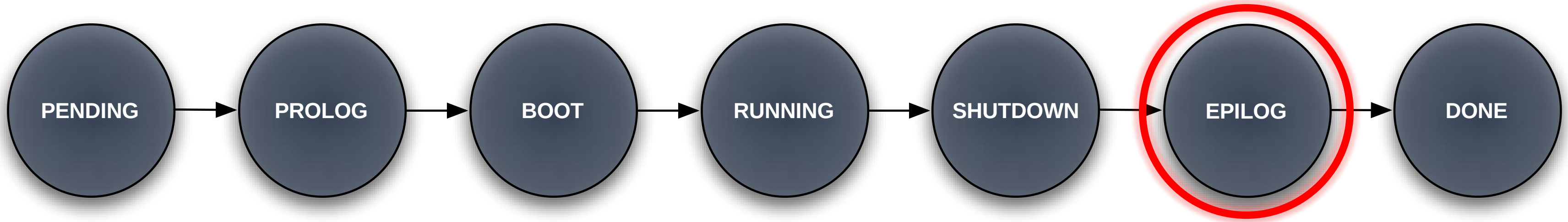


User provided, OpenNebula provided contextualization info

Running and Shutdown states

- While the VM is in RUNNING state it will be periodically polled to get its consumption and state.
- In SHUTDOWN state VM driver will send the shutdown command to the underlying virtual infrastructure.

Epilog state



In EPILOG state :

- Copy back the images that have ***SAVE****=yes* option.
- Delete images that were cloned or generated by **MKSWAP**.

The Benefits of OpenNebula

For the Infrastructure User

- Faster delivery and scalability of services
- Support for heterogeneous execution environments
- Full control of the lifecycle of virtualized services management

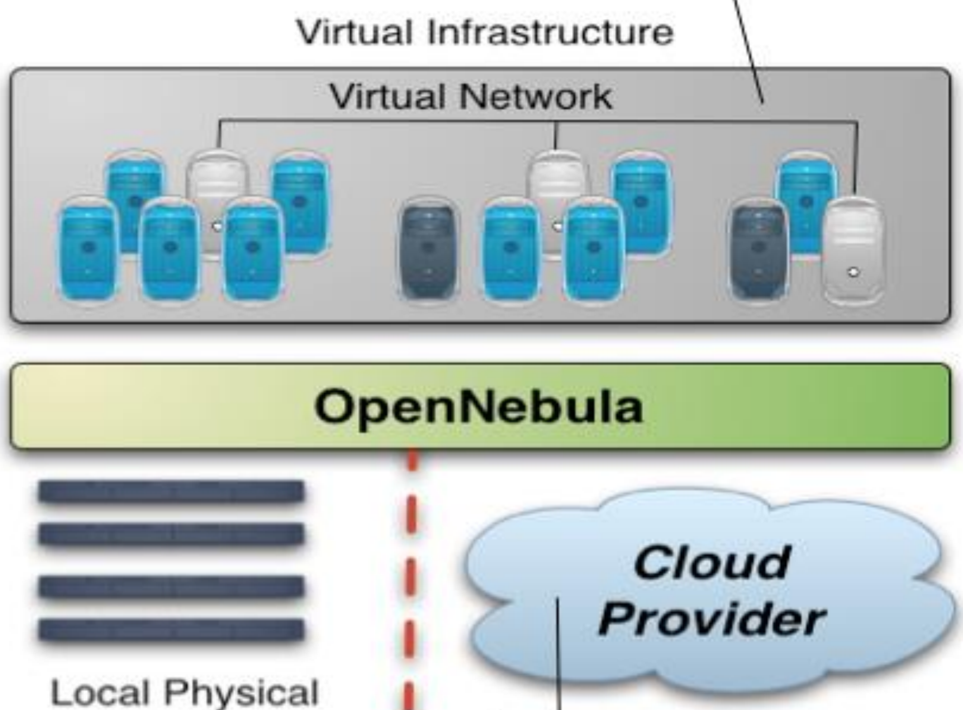
The Benefits of OpenNebula

For the Infrastructure Manager

- Centralized management of VM workload and distributed infrastructures
- Support for VM placement policies: balance of workload, server consolidation…
- Dynamic resizing of the infrastructure
- Dynamic partition and isolation of clusters
- Dynamic scaling of private infrastructure to meet fluctuating demands
- Lower infrastructure expenses combining local and remote cloud resources

Designing an Hybrid Cloud

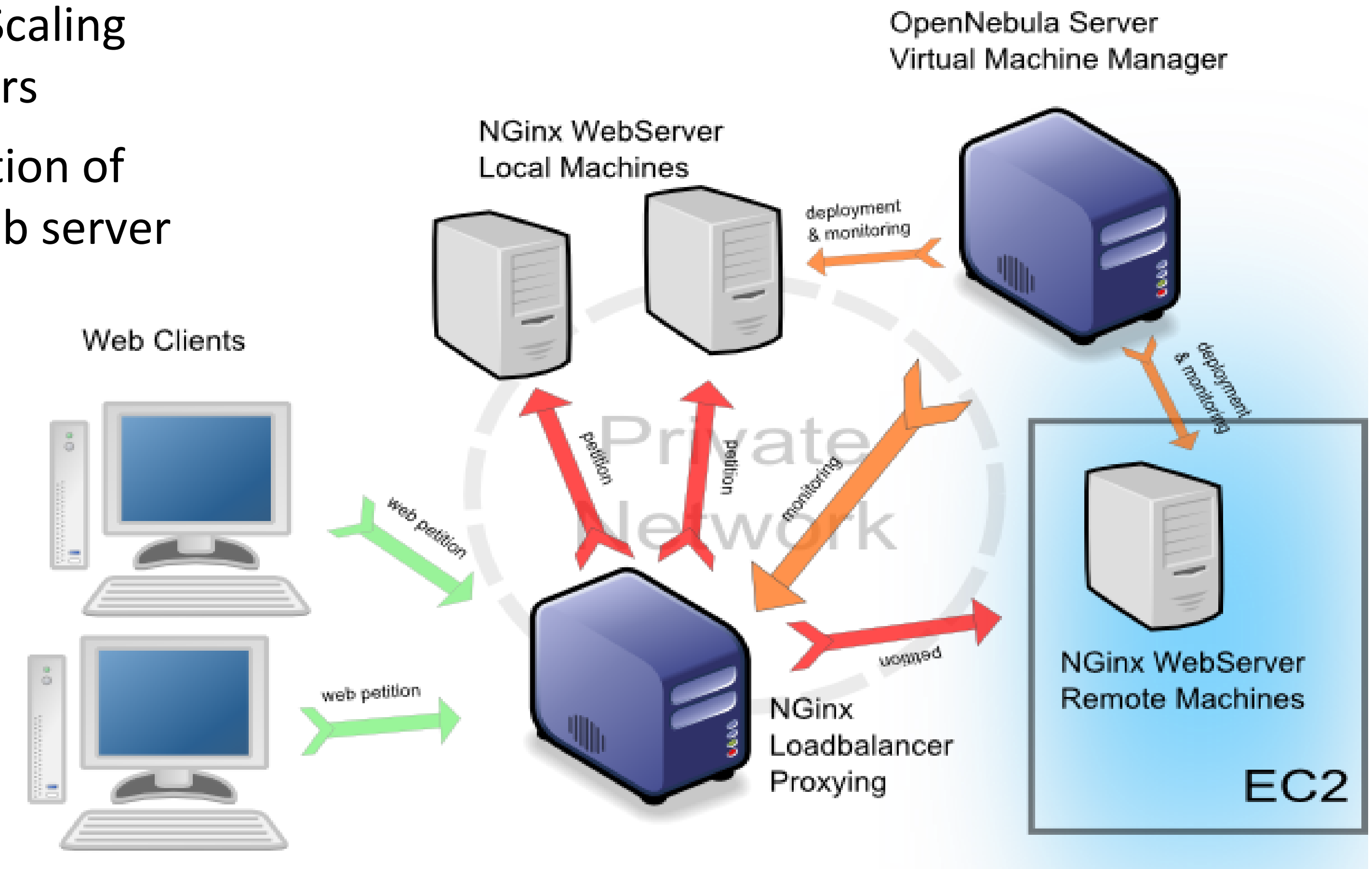
- VMs can be local or remote
- VM connectivity has to be configured, usually VPNs



- External Clouds are like any other host
- Placement constraints

Hybrid Cloud Use Case

- On-demand Scaling of Web Servers
- Elastic execution of the NGinx web server



- The capacity of the elastic web application can be dynamically increased or decreased by adding or removing NGinx instances

Summary

- Need for Cloud Resource management (CRM)
- Capacity Planning & Resource Scheduling
- Virtual Infrastructure Management (VIM)
- Challenges for Private Cloud : Finite Capacity
- Focus Areas for VIM
  - Distributed management of virtual machines
  - Reservation-based provisioning of virtualized resources
  - Provisioning to meet SLA commitments.
- OpenNebula

References

- Llorente, I.M., Montero, R.S., Sotomayor, B., Breitgand, D., Maraschini, A., Levy, E. and Rochwerger, B. (2011). On the Management of Virtual Machines for Cloud Infrastructures. In Cloud Computing (eds R. Buyya, J. Broberg and A. Goscinski). https://doi.org/10.1002/9780470940105.ch6
- On the Management of Virtual Machines for Cloud Infrastructures - PowerPoint Presentation
- https://opennebula.io/, https://opennebula.io/blog/newsletter/2024-pivotal-year-opennebula/,
- https://docs.opennebula.io/6.10/quick_start/deployment_basics/try_opennebula_on_kvm.html
- Task scheduling and VM placement to resource allocation in Cloud computing: challenges and opportunities | Cluster Computing - 2023
- Effective Resource Management through VM Allocation in Cloud Data Center | Proceedings of the 18th Innovations in Software Engineering Conference – 2025
- Resource Management in Cloud IaaS via Machine Learning Algorithms | SpringerLink

Additional Slides

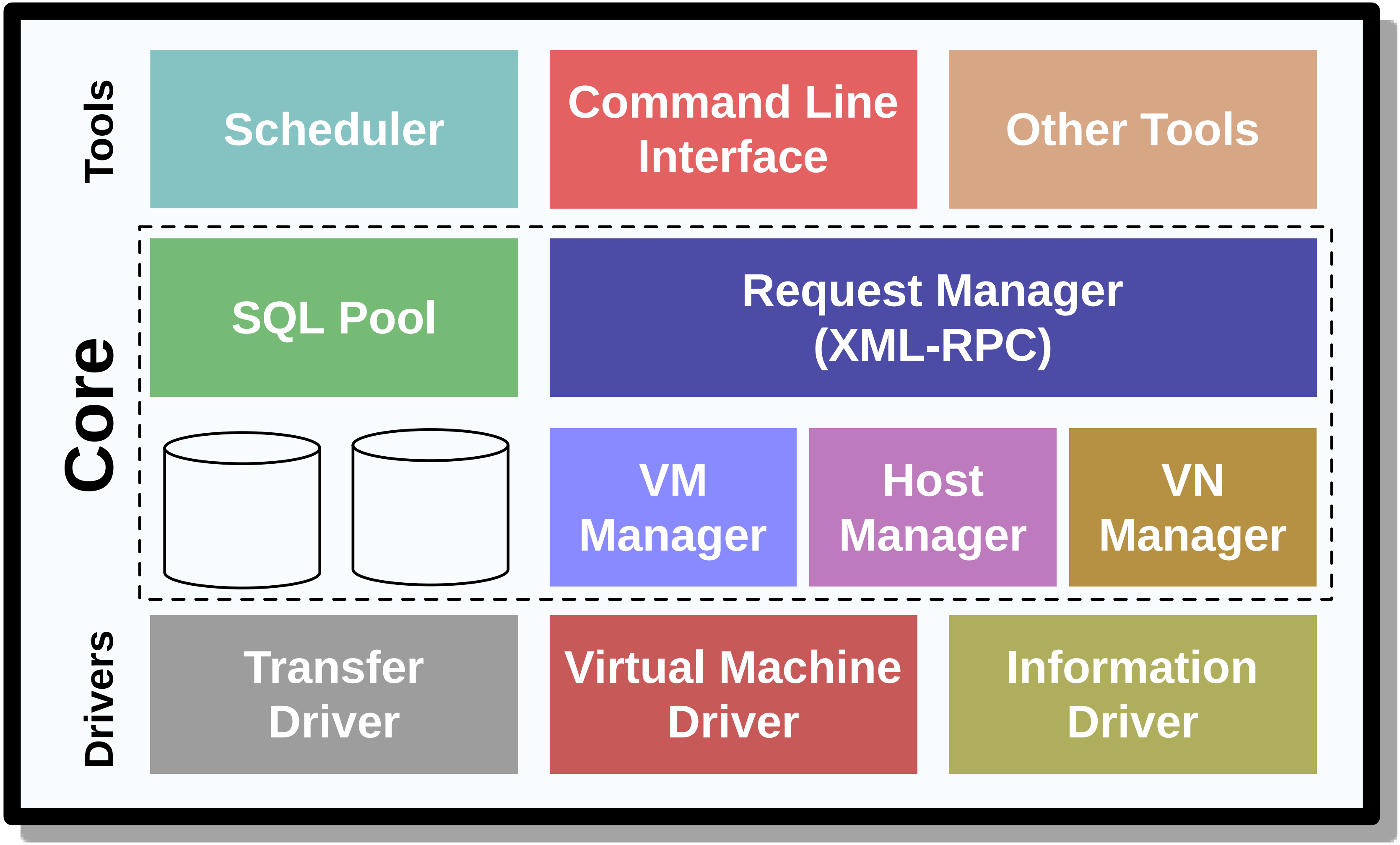
The Main Features of OpenNebula

Feature	Function
Internal Interface	• Unix-like CLI for fully management of VM life-cycle and physical boxes • XML-RPC API and libvirt virtualization API
Scheduler	• Requirement/rank matchmaker allowing the definition of workload and resource-aware allocation policies • Support for advance reservation of capacity through Haizea
Virtualization Management	• Xen, KVM, and VMware • Generic libvirt connector (VirtualBox)
Image Management	• General mechanisms to transfer and clone VM images
Network Management	• Definition of isolated virtual networks to interconnect VMs
Service Management and Contextualization	• Support for multi-tier services consisting of groups of inter-connected VMs, and their auto-configuration at boot time
Security	• Management of users by the infrastructure administrator
Fault Tolerance	• Persistent database backend to store host and VM information
Scalability	• Tested in the management of medium scale infrastructures with hundreds of servers and VMs (no scalability issues has been reported)
Flexibility and Extensibility	• Open, flexible and extensible architecture, interfaces and components, allowing its integration with any product or tool

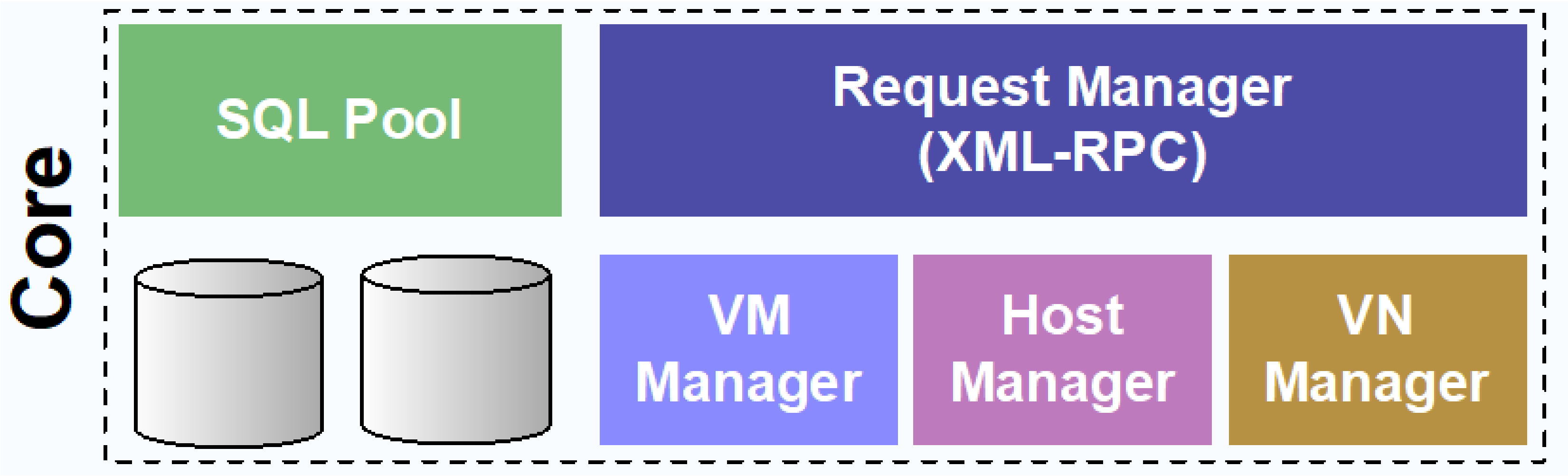
Comparison with Similar Technologies

	Platform ISF	VMware Vsphere	Eucalyptus	Nimbus	OpenNebula
Virtualization Management	**VMware, Xen**	**VMware**	**Xen, KVM**	**Xen**	**Xen, KVM, VMware**
Virtual Network Management	**Yes**	**Yes**	No	**Yes**	**Yes**
Image Management	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**
Service Contextualization	No	No	No	**Yes**	**Yes**
Scheduling	**Yes**	**Yes**	No	No	**Yes**
Administration Interface	**Yes**	**Yes**	No	No	**Yes**
Hybrid Cloud Computing	No	No	No	No	**Yes**
Cloud Interfaces	No	**vCloud**	**EC2**	**WSRF, EC2**	**EC2 Query, OGF OCCI**
Flexibility and Extensibility	**Yes**	No	**Yes**	**Yes**	**Yes**
Open Source	No	No	**GPL**	**Apache**	**Apache**

OpenNebula Architecture

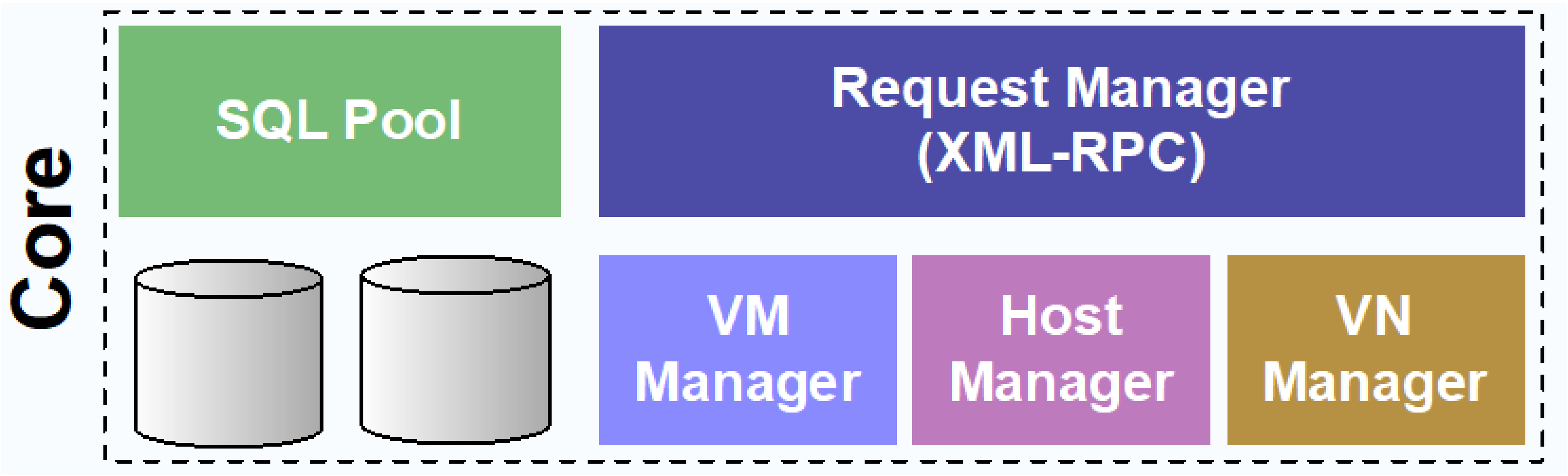


The Core



- **Request manager:** Provides a XML-RPC interface to manage and get information about ONE entities.
- **SQL Pool:** Database that holds the state of ONE entities.
- **VM Manager (virtual machine):** Takes care of the VM life cycle.

The Core



- **Host Manager:** Holds information about hosts.
- **VN (Virtual Network) Manager:** In charge of generating MAC and IP addresses.

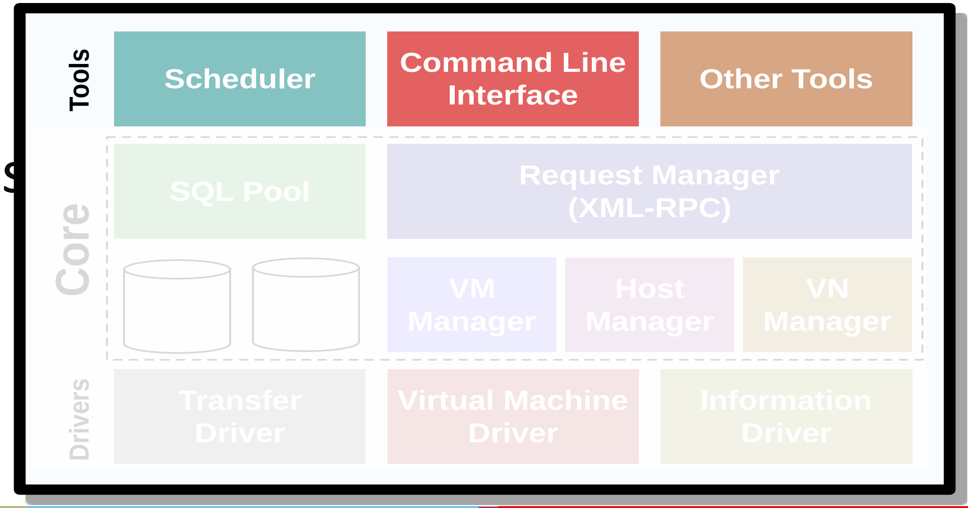
The Tools Layer

Scheduler:

- Searches for physical hosts to deploy newly defined VMs

Command Line Interface:

- Commands to manage OpenNebula.
- onevm: Virtual Machines
  - create, list, migrate…
- onehost: Hosts
  - create, list, disable…
- onevnet: Virtual Networks
  - create, list, delete…



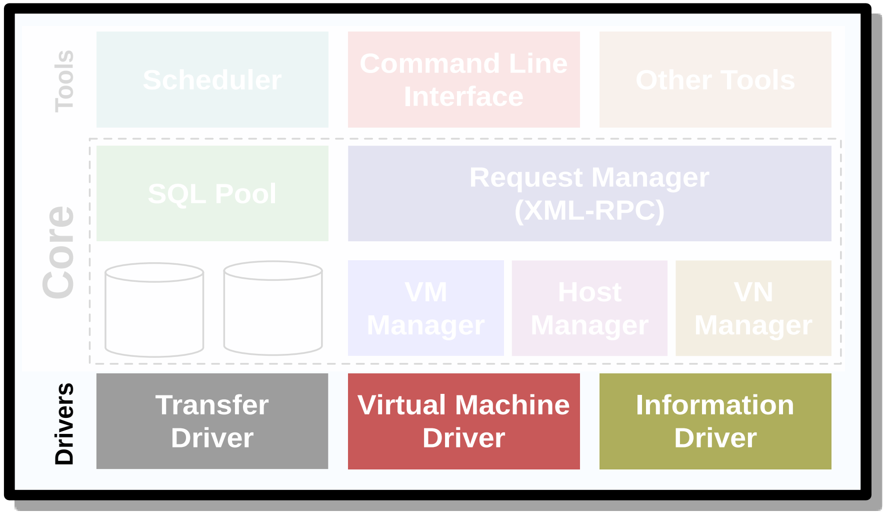
The Drivers Layer

Transfer Manager Driver: Takes care of the images.

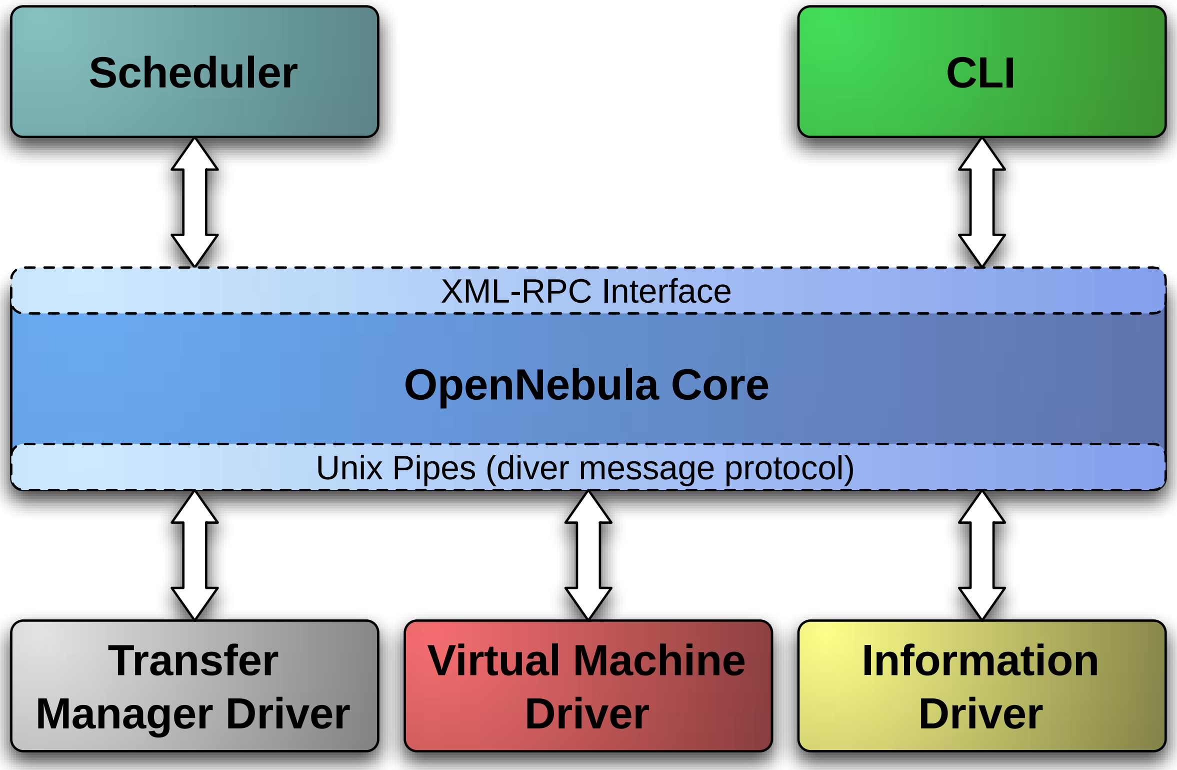
- cloning, deleting, creating swap image…

Virtual Machine Driver: Manager of the lifecycle of a virtual machine - deploy, shutdown, poll, migrate…

Information Driver: Executes scripts in physical hosts to gather information about resources - total memory, free memory, total #cpus, cpu consumed, etc.



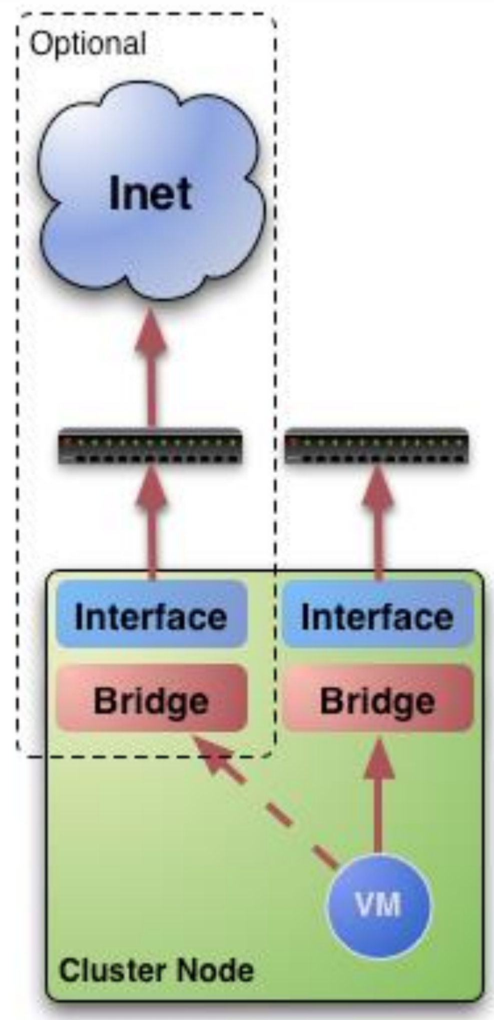
Process Separation



- Scheduler is a separated process, just like command line interface.
- Drivers are also separate processes using a simple text messaging protocol to communicate with OpenNebula Core Daemon (oned)

Networking for Private Clouds

- OpenNebula management operations use ssh connections
- ***Image traffic,*** may require the movement of heavy files (VM images, checkpoints). Dedicated storage links may be a good idea
- ***VM demands,*** consider the typical requirements of your VMs. Several NICs to support the VM traffic may be a good idea
- OpenNebula relies on bridge networking for the VMs



Example network setup in a private cloud