UNIVERSITÀ DEGLI STUDI DI FERRARA
- EX LABORE FRUCTUS -

Co-funded by the Erasmus+ Programme of the European Union

TORUS

TowardOpenResourcesUsingServices

6 – 10 June, 2016 @ University of Ferrara

Distributed Computing Architectures and Environmental Science Applications

Overview of Distributed Computing Architectures

Luca Tomassetti - UNIFE

Contents

- Basic Definitions & Motivations for Distributed Computing
- Overview of Grid Architecture
- DIRAC as a tool to harmonize access to distributed resources
- Key Components of a Cloud
  - Characteristics, Delivery and Deployment Models
- Cloud Reference Architecture & Virtualization
- TORUS needs?

Basic Definitions

- A distributed system consists of multiple autonomous computers that communicate through a computer network.
- Distributed computing utilizes a network of many computers, each accomplishing a portion of an overall task, to achieve a computational result much more quickly than with a single computer.
- Distributed computing is any computing that involves multiple computers remote from each other that each have a role in a computation problem or information processing.

Co-funded by the Erasmus+ Programme of the European Union

Basic Definitions

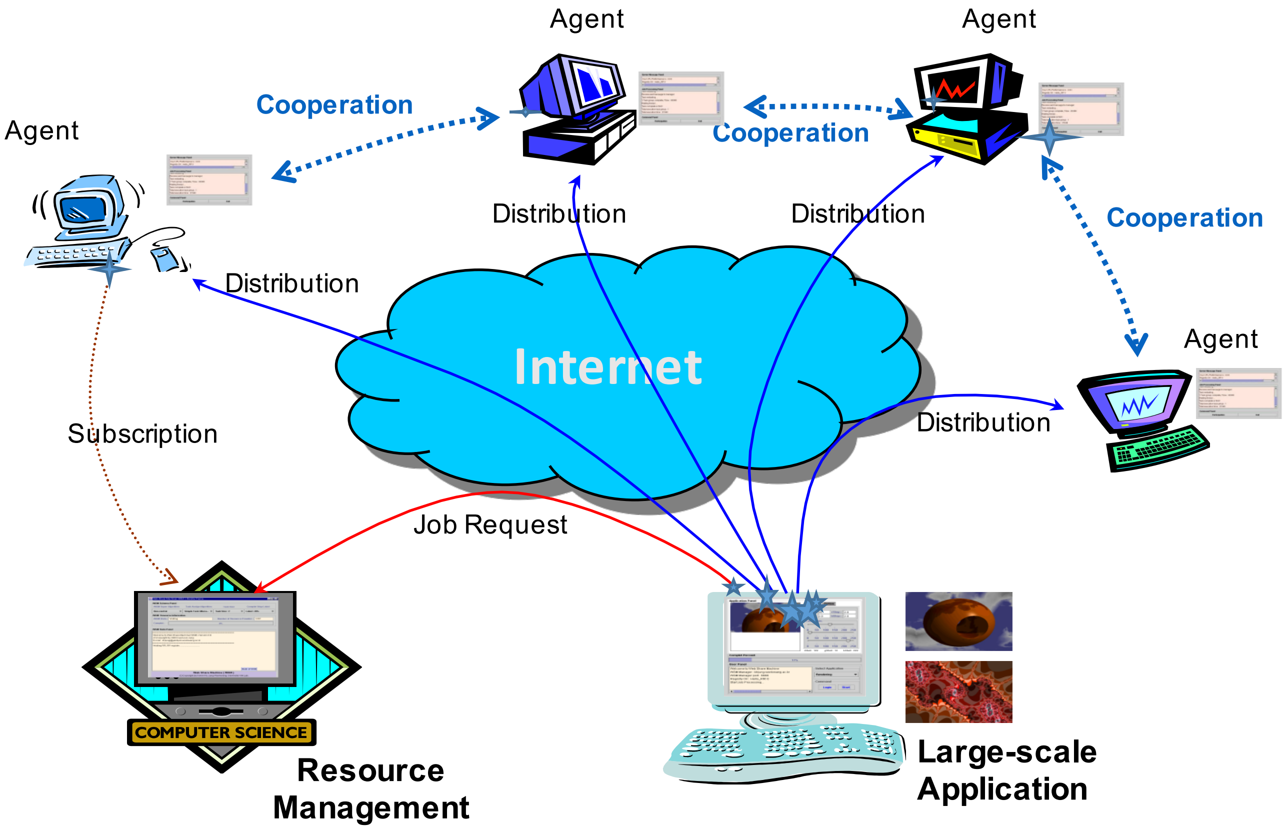
- **A distributed system is one in which hardware or software components located at networked computers communicate and coordinate their actions only by message passing.**
- In the term distributed computing, the word *distributed* means *spread out across space*. Thus, distributed computing is an activity performed on a spatially distributed system.
- These networked computers may be in the same room, same campus, same country, or in different continents (eventually away from us)

Basic Definitions

- **A distributed system is a collection of independent computers that appears to its users as a single coherent system.** *[Andrew S. Tanenbaum]*
- A distributed system consists of several autonomous computers, connected through a network and distributed operating system software, which enables computers to coordinate their activities and to share the resources of the system - hardware, software and data, so that users perceive the system as a single, integrated computing facility.
- **This means that one way or the other the autonomous components need to collaborate. How to establish this collaboration lies at the heart of developing distributed systems.**

Co-funded by the Erasmus+ Programme of the European Union

Overview



Motivation

- Inherently distributed applications.
  Distributed systems have come into existence in some very natural ways, e.g., in our society people are distributed and information could/should also be distributed.
- Performance/cost.
  The parallelism of distributed systems reduces processing bottlenecks and provides improved all-around performance, i.e., distributed systems offer a better price/performance ratio.
- Resource sharing
  Distributed systems can efficiently support information and resource (hardware and software) sharing for users at different locations.
- Flexibility and extensibility
  Distributed systems are capable of incremental growth and have the added advantage of facilitating modification or extension of a system to adapt to a changing environment without disrupting its operations.
- Availability and fault tolerance
  With the multiplicity of storage units and processing elements, distributed systems have the potential ability to continue operation in the presence of failures in the system.
- Scalability
  Distributed systems can be easily scaled to include additional resources (both hardware and software).
- Network connectivity is highly available
- Combination of cheap processors often more cost-effective than one expensive fast system
- Potential increase of reliability

Co-funded by the Erasmus+ Programme of the European Union

Goals

- Making Resources Accessible
  - Data sharing and device sharing
- Distribution Transparency
  - Access, location, migration, relocation, replication, concurrency, failure
- Communication
  - Make human-to-human communications easier.
- Flexibility
  - Spread the work load over the available machines in the most cost effective way
- To coordinate the use of shared resources
- To solve large computational problem

Co-funded by the Erasmus+ Programme of the European Union

Characteristics

- Resource Sharing
- Openness
- Concurrency
- Scalability
- Fault Tolerance
- Transparency

Key point is … sharing resources

- The main goal of a distributed computing system is to connect users and IT resources in a transparent, open, cost-effective, reliable and scalable way.
- The resources that can be shared in grids, clouds and other distributed computing systems include:

- Physical resources
  - Computational power
  - Storage devices
  - Communication capacity
- Virtual resources
  which can be exchanged and are independent from its physical location
  - Operating systems
  - Software and licenses
  - Tasks and applications
  - Services

Taxonomy

- A number of new paradigms and terms related to distributed computing have been introduced, promising to deliver IT as a service.

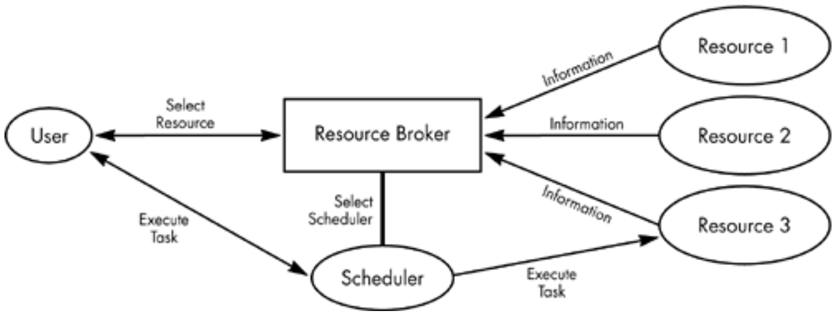
While experts disagree on the precise boundaries between these new computing models, the following table provides a rough taxonomy.

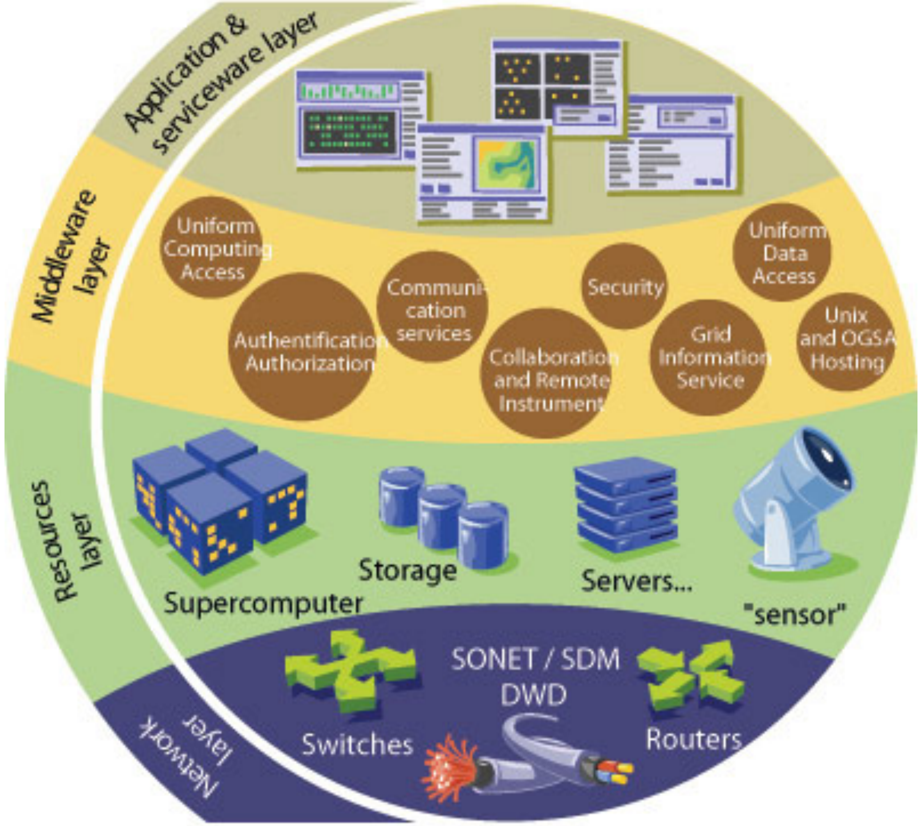
New Computing Paradigms	New Services	New or echanced Features
• **Cloud** computing • Edge computing • **Grid** computing • Utility computing	• IaaS – Infrastructure as a Service • PaaS – Platform as a Service • SaaS – Software as a Service • SOA – Service-Oriented Architecture	• Ubiquitous access • Reliability • Scalability • Virtualization • Exchangeability / Location independence • Cost-effectiveness

Co-funded by the Erasmus+ Programme of the European Union

Grid Architecture

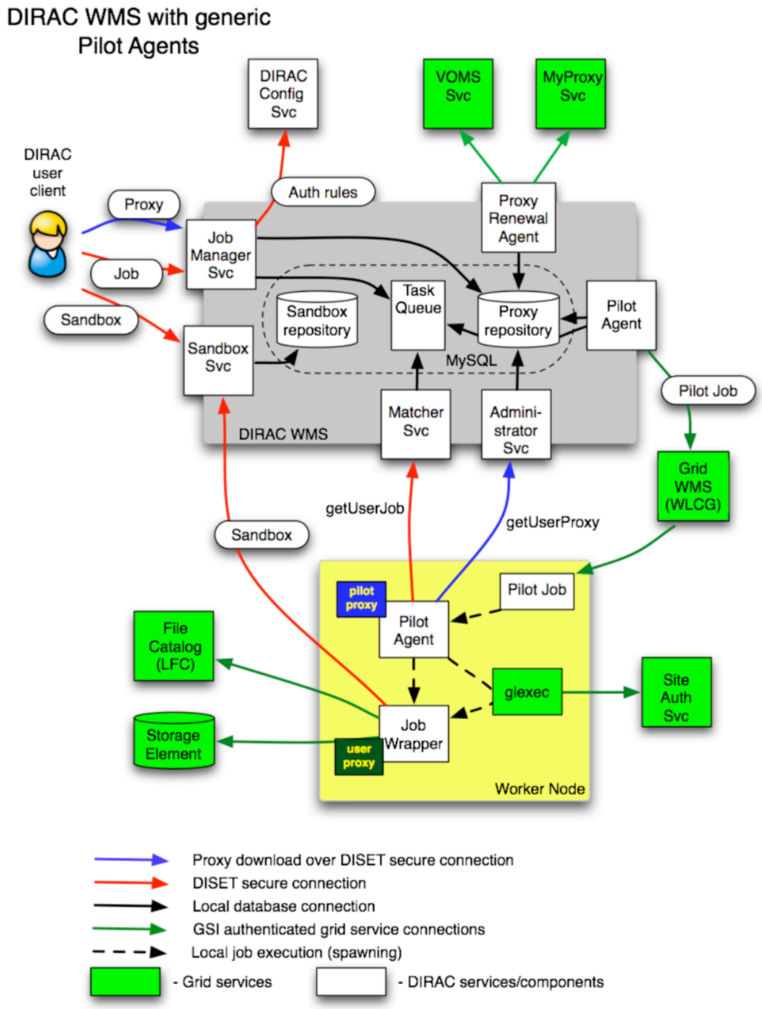
- Middleware layer takes care of
  - Discovery and match available resources
  - Cooperation among resources
  - Admin/utility tasks
- Several 'flavours'
- Where data is? How long the job will last?





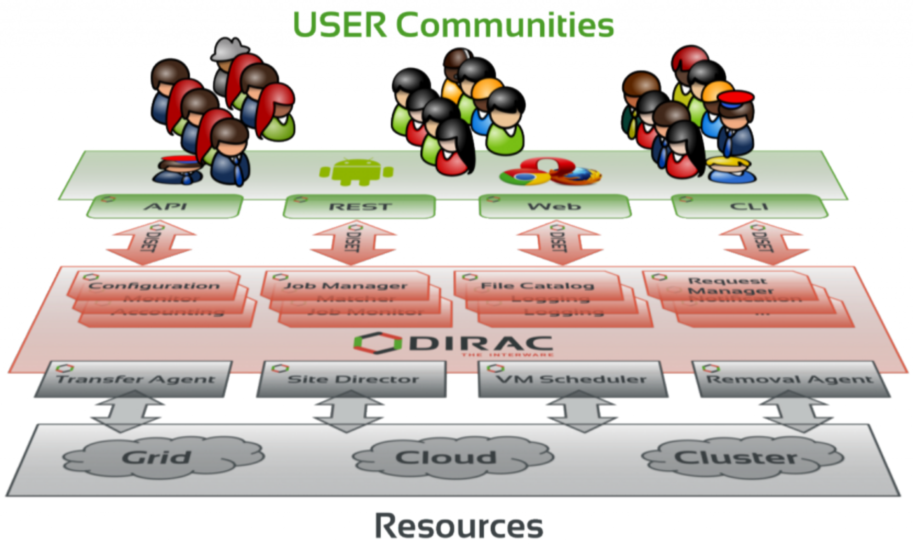
Grid Architecture

- Nowadays original paradigm switched to a 'pilot'–based approach
  - Send a 'fake' job to some suitable WN
  - Let it pull 'real' jobs there



DIRAC
THE INTERWARE

- DIRAC (Distributed Infrastructure with Remote Agent Control) INTERWARE is a software framework for distributed computing providing a complete solution to one (or more) user community requiring access to distributed resources.
- DIRAC builds a layer between the users and the resources offering a common interface to a number of heterogeneous providers, integrating them in a seamless manner, providing interoperability, at the same time as an optimized, transparent and reliable usage of the resources.
- The Workload Management System with Pilot Jobs deployed in DIRAC is now widely used in various distributed computing infrastructures.
- This concept allows to aggregate in a single system computing resources of different source and nature, such as computational grids, clouds or clusters, transparently for the end users.



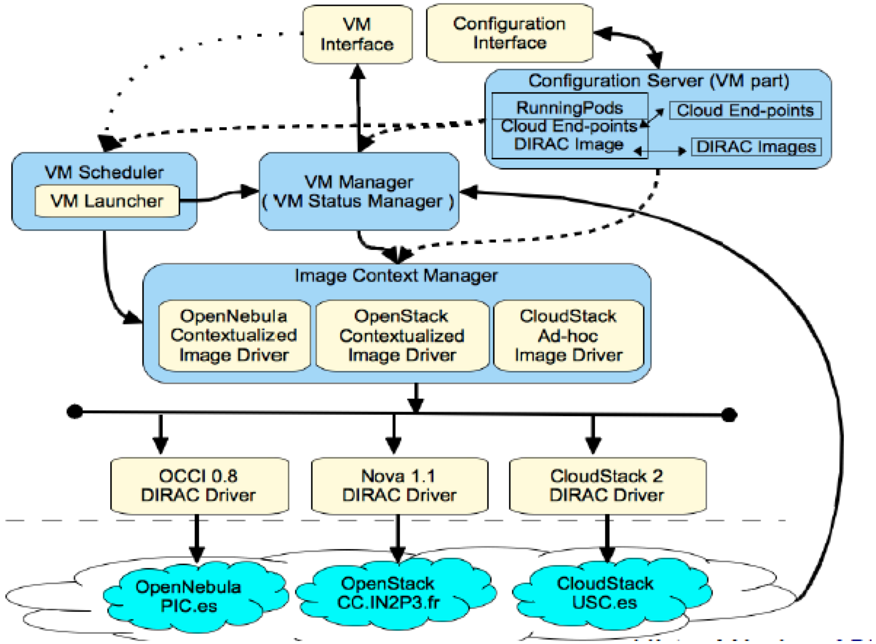
Università degli Studi di Ferrara

Co-funded by the Erasmus+ Programme of the European Union

DIRAC
THE INTERWARE

- VMDIRAC provides an enterprise contextualization method, based in a golden image with cloudinit installed, and some automatic scripts for each cloud endpoint and DIRAC image to run a VM.
- Supported cloudmangers for cloudinit with DIRAC:
  - OpenNebula 4.6.1
  - Openstack Grizzly
  - Amazon EC2

VMDIRAC Architecture



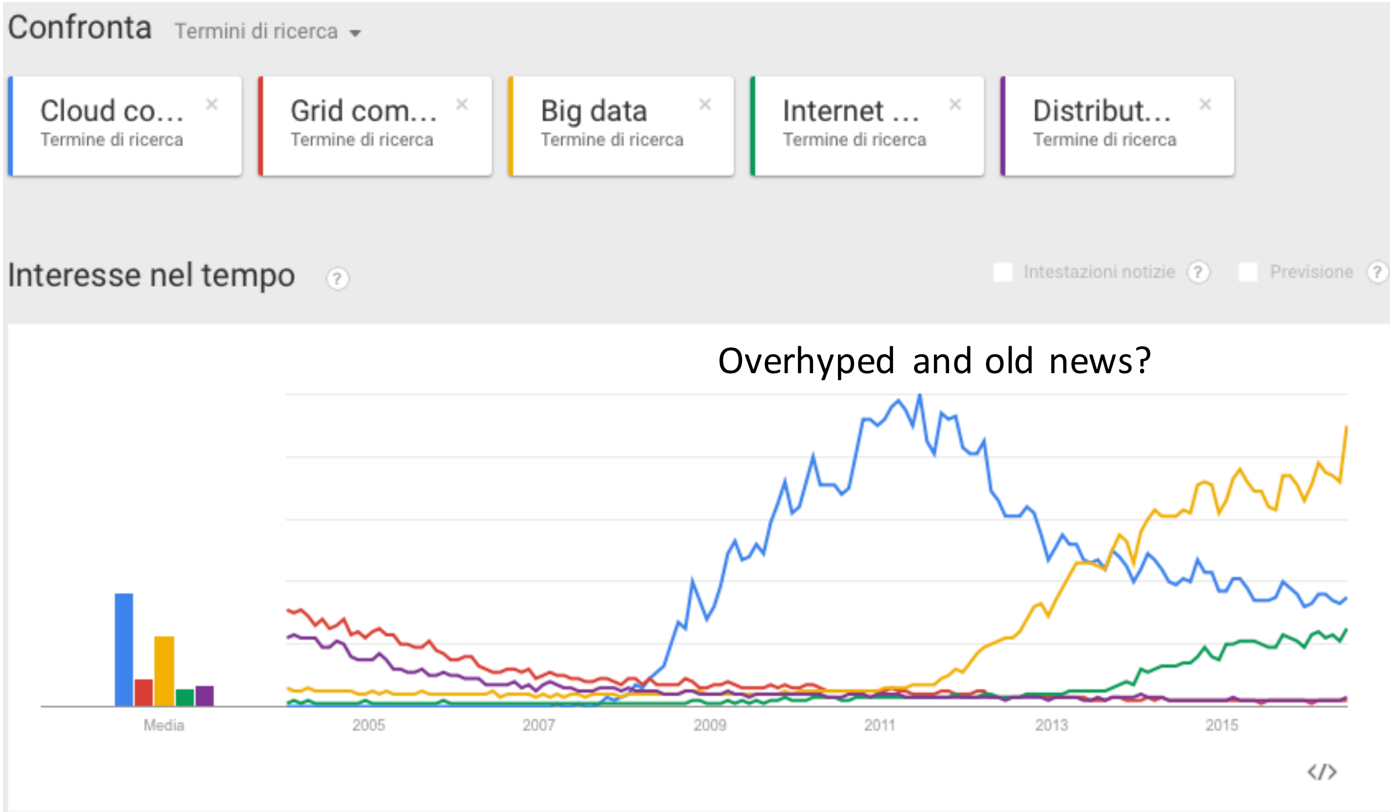
Cloud Definition

- U.S. National Institute for Standards and Technology (NIST):
  "Cloud computing is a model for enabling ubiquitous, convenient, on–demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" *[Mell & Grance, 2011]*

- Outsourced shared-computing where resources
  - are virtualised, distributed and pooled amongst external data centres
  - accessed by users through the internet *[Venters & Whitley 2012]*

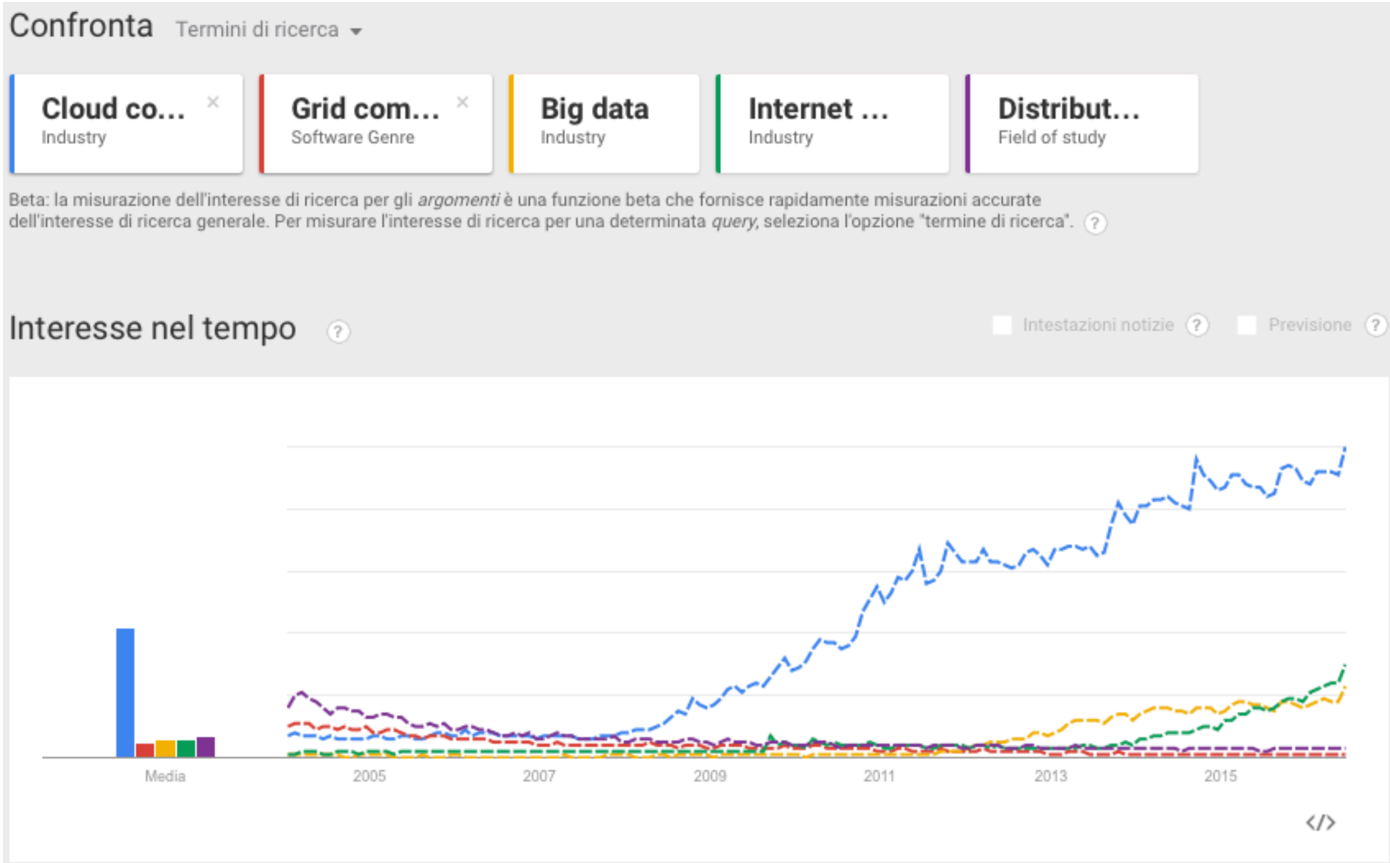
Trends (search engine queries)

- Cloud Computing
- Grid Computing
- Big Data
- Internet of Things
- Distributed Computing



Trends (industry)

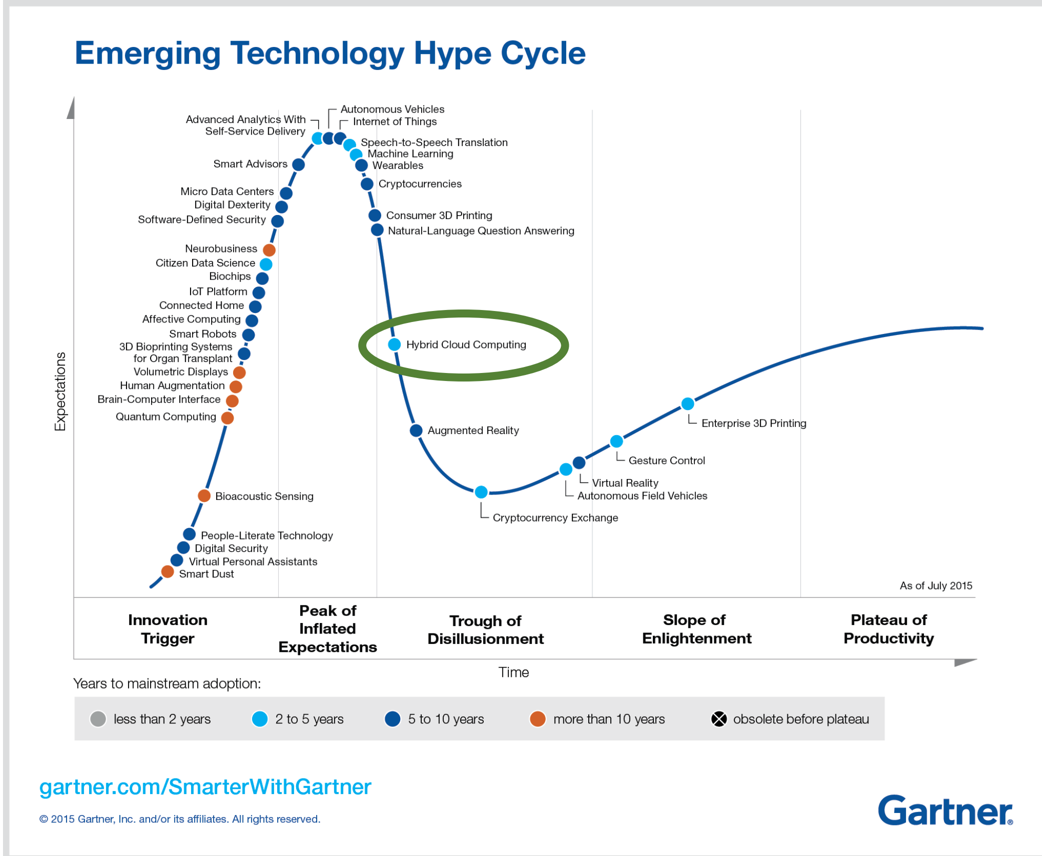
- Cloud Computing
- Grid Computing
- Big Data
- Internet of Things
- Distributed Computing



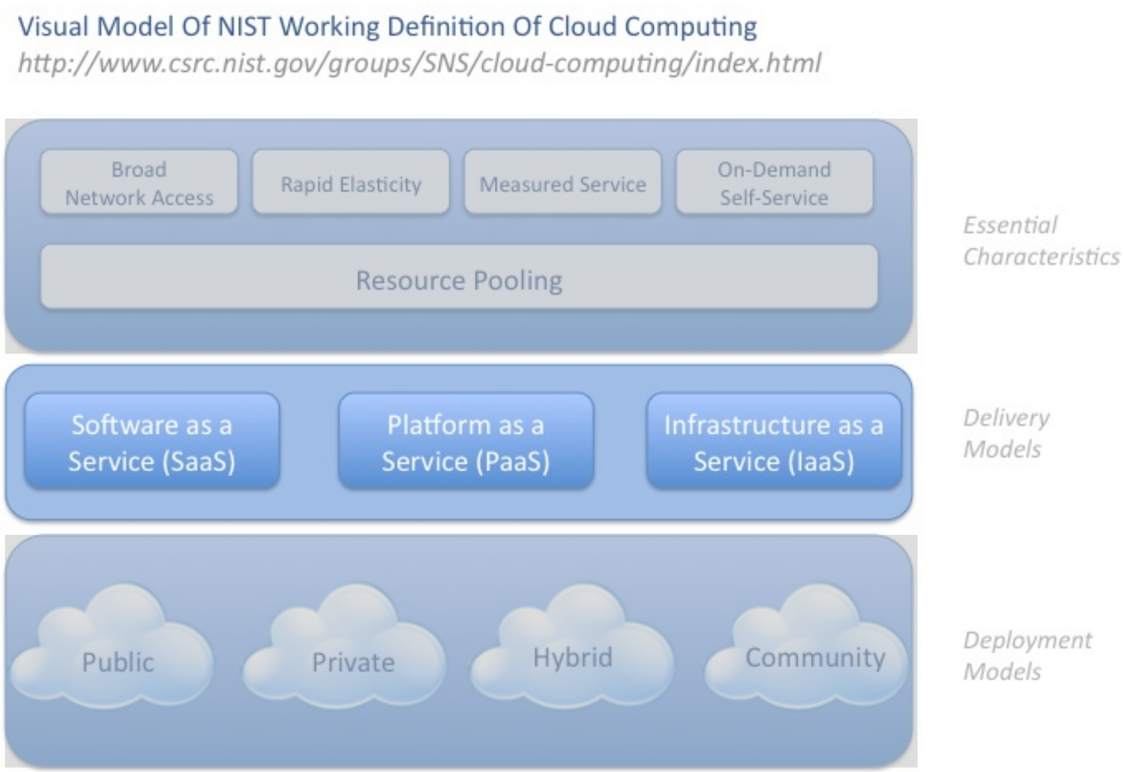
Trends (of interest?)

- Just entered or
  about to enter the mainstream?

- Anyway, nowadays there is a huge utilization of Cloud(-like) resources in several fields
- Even Physicists ☺



Key components of Clouds



Delivery Models:

- Cloud providers are offering services, according to several models.

That means, customers may purchase:

- `IaaS` – Infrastructure as a Service
- `PaaS` – Platform as a Service
- `SaaS` – Software as a Service

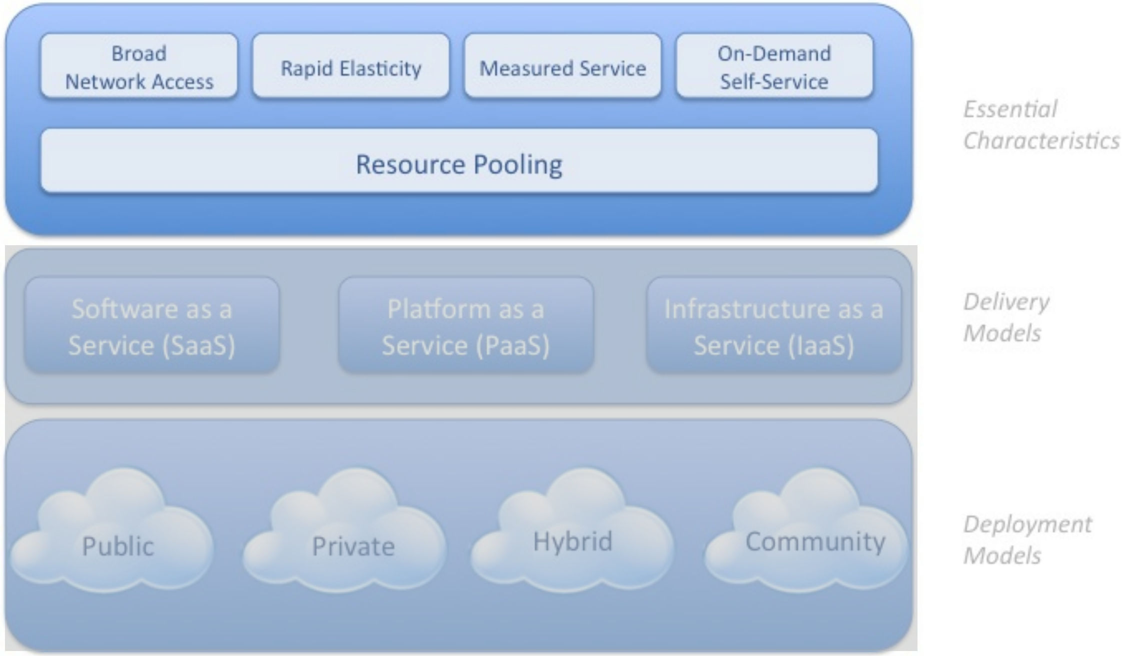
Delivery – Service models (what customers may purchase)

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

• Virtualisation of physical Compute Assets
  • Storage
  • Processing
• No control over underlying cloud infrastructure
• Control over ability to deploy and run software
  • operating systems and applications
• *Amazon Web Services (AWS), OpenStack, CloudStack, …*
	• Virtual development environment
	• Develop & deploy applications for the Cloud
	• No control over underlying Cloud infrastructure
	• Control over deployed application
	  • e.g. provisioning and access
	• *Google App Engine, Microsoft Azure*
		• Access to Service Providers Applications that execute on the Cloud
		• Accessed via thin client interface such as a web browser (or smartphone app)
		• No control over underlying Cloud infrastructure
		• Minimal control over application settings
		• *Gmail, Google Docs, DropBox, Facebook, Evernote, ...*

Key components of Clouds

Visual Model Of NIST Working Definition Of Cloud Computing
http://www.csrc.nist.gov/groups/SNS/cloud-computing/index.html

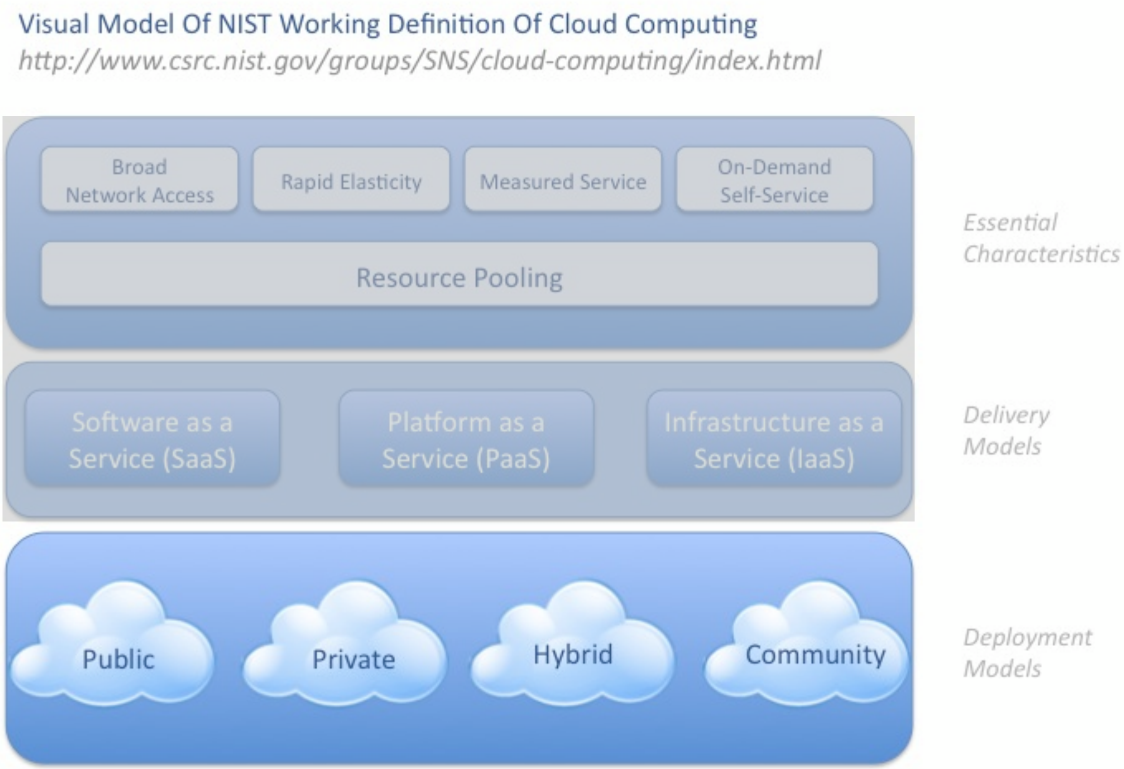


Essential characteristics:

- On Demand Self Service
  - Commoditised
- Measured Service
  - Variable Cost Model, pay for capacity you use
- Resource Pooling
  - High Utilisation & Economies of scale
- Rapid Elasticity
  - Commission / Decommission Capacity
- Broad Network Access
  - Accessibility over internet

Which usually meets Organitation's desires of simplification

Key components of Clouds

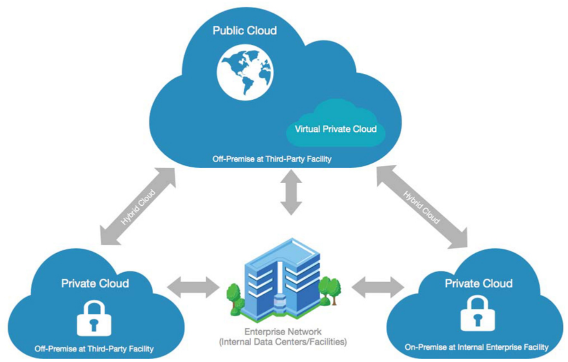


Deployment

- Cloud can be deployed in different ways
  - Public
  - Private
- Which have implications on the organisation ……. control!

Co-funded by the Erasmus+ Programme of the European Union

Deployment models

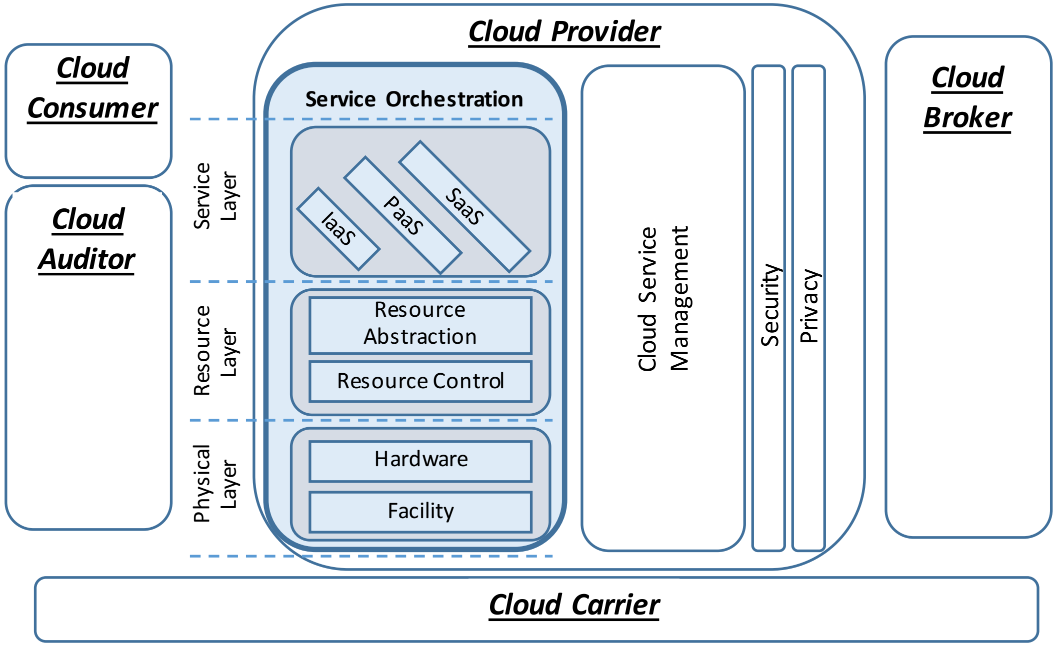


Public	Private

• Amazon AWS, Google App Engine, Microsoft * …
• Benefits of computing with:
  • Significant Cost Savings (Economies of Scale, PAYG, Low Overheads)
  • High Performance (Super computer power, Latency)
  • Very Flexible (switching on & off Virtualised Hardware and Software)
• Comes at cost of loss of control
  • Lack of transparency
  • Sharing of computing assets
• Multi-tenancy architecture shared by all
	• Not shared - operated solely for a single organization.
	• Hosted / Non Hosted Solutions
	• Benefit: Under organization control
	• Whilst VM architecture essential, it will lack benefits of sharing:
	  • Cost; Scalability; Performance

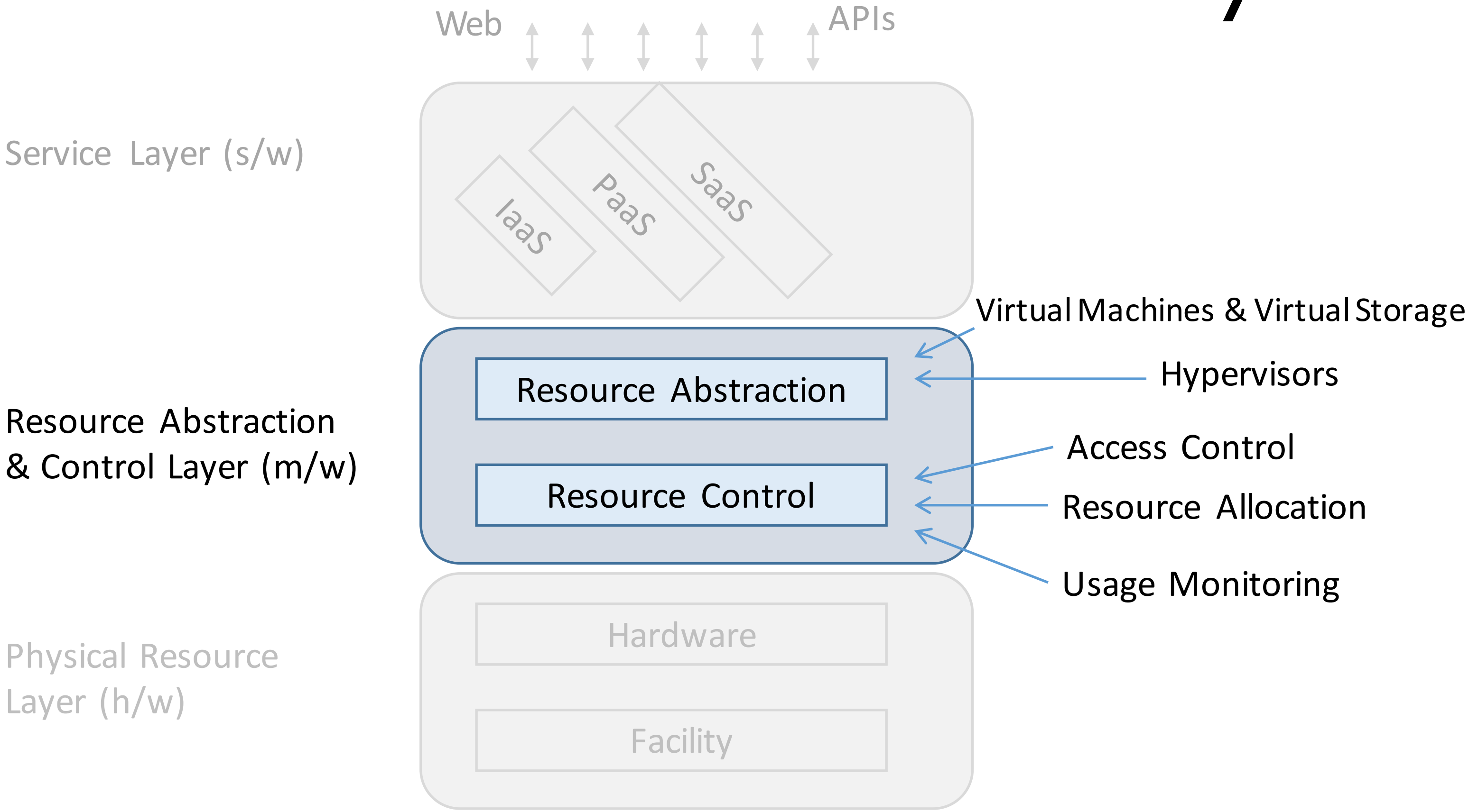
Cloud reference Architecture

- NIST reference
- Public clouds
  - Monolithic
  - Black Boxed
  - Closed
  - Cloud Service Provider has Control
- Private clouds
  - Open
  - Layered Modular
  - Outsourcer has Control



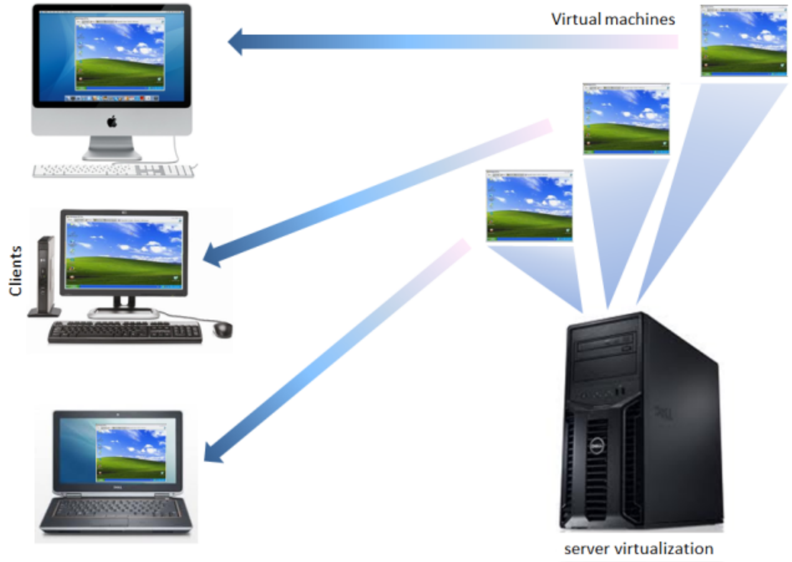
Service Orchestration

Resources Abstraction & Control Layer

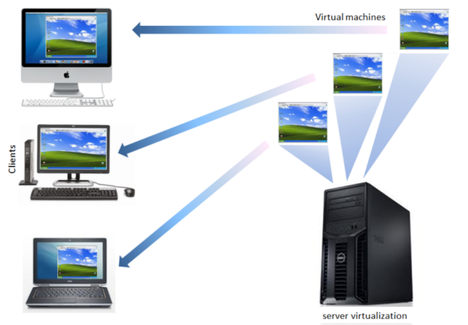


Virtualization

- VM Images *(Can be used by several VM simultaneously and shared)*
  - OS
  - Data
- VM Templates *(Can be instatiated several times and shared)*
  - CPU and Memory capacity
  - Set of Network interfaces to VNs
  - Set of Images
  - Context
- VM Instances



UNIVERSITÀ DEGLI STUDI DI FERRARA
- EX LABORE FRUCTUS -

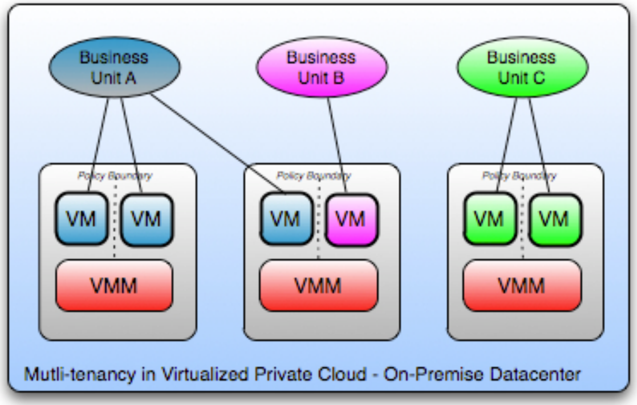


Co-funded by the Erasmus+ Programme of the European Union

Virtualization

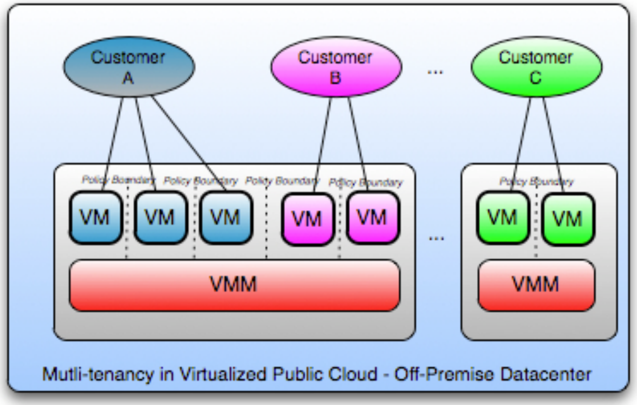
- Single vs. Multi Tenancy

On-Premises Data Centre
(e.g. Private Cloud)



Private Cloud of Company XYZ with 3 business units, each with different security, SLA, governance and chargeback policies on shared infrastructure

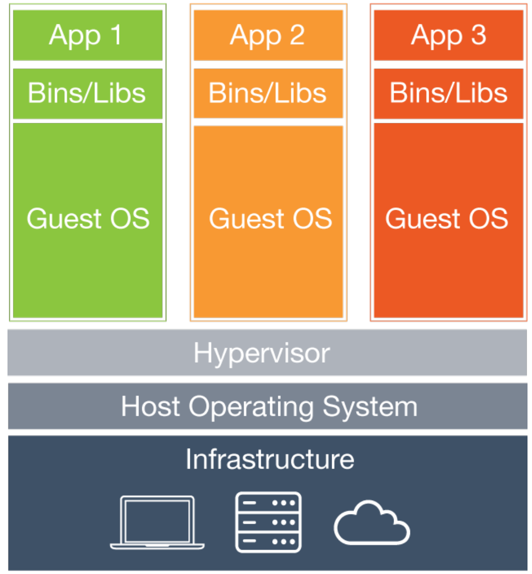
Off-Premises Data Centre
(e.g. Public Cloud)

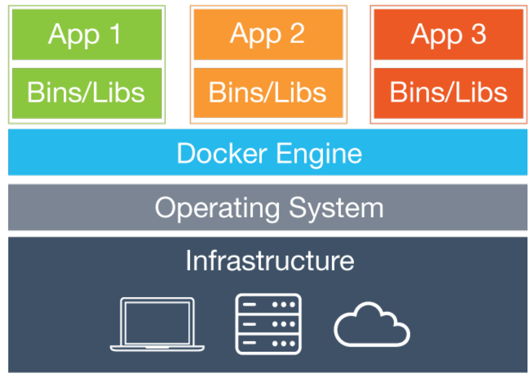


Public Cloud Provider with 3 business customers, each with different security, SLA, governance and billing policies on shared infrastructure

Containers

- See for example: docker.com
- Virtual Machines
  - Each virtual machine includes the application, the necessary binaries and libraries and an entire guest operating system (>1GBs in size).
- Containers
  - Containers include the application and all of its dependencies, but share the kernel with other containers. They run as an isolated process in userspace on the host operating system. They're also not tied to any specific infrastructure – containers run on any computer, on any infrastructure and in any cloud.





TORUS needs?

- An IaaS product?
- A SaaS product?

- We must list all the requirements we have in the domain of:
  - Data size, ownership, location
  - Network capacity
  - Algorithms in use, so memory and CPU needs
  - Possibility to divide the computational (and data I/O) task in several parts

Thanks

- diracgrid.org
- aws.amazon.com
- www.openstack.org
- cloudstack.apache.org
- www.opennebula.org
- www.docker.com
- spark.apache.org