



## - Goals and topics

### - Goals

- introduce cloud computing
- discuss some technical, architectural and economic aspects of cloud computing
- discuss the cloud as a platform for software release

### - Subjects

- introduction to cloud computing
- examples of services in the cloud
- cloud computing
- cloud architecture
- economics of cloud computing
- cloud software systems
- discussion



## \* Introduction to cloud computing

- Cloud computing comes from an old idea: computing (computational resources) as a utility service

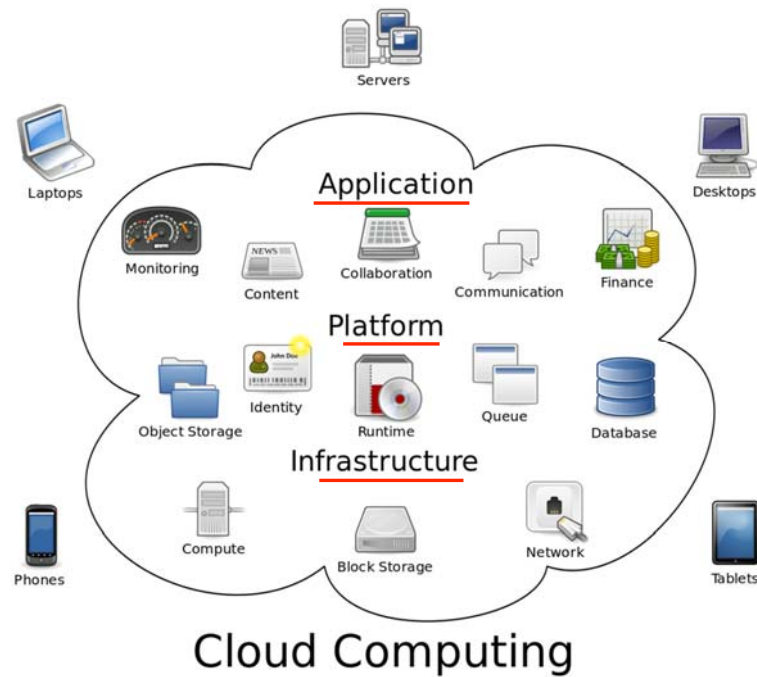
can be seen as an utility (service?)





# Introduction to cloud computing

- Cloud computing comes from an old idea: computing (computational resources) as a utility service



5

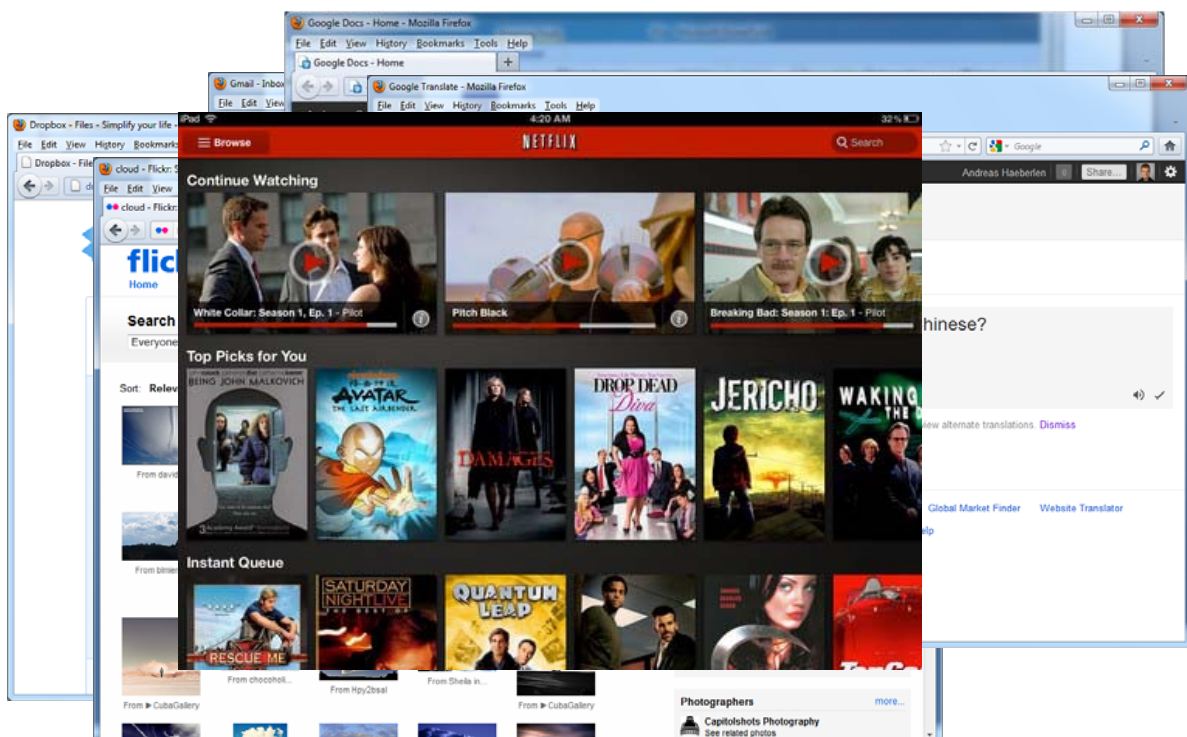
Cloud Computing

Luca Cabibbo ASW



# Cloud computing

- Have you ever used the cloud before?



6

Cloud Computing

Luca Cabibbo ASW




## Cloud software

- Today, it is increasingly common to release your software systems in the cloud - or in data centers managed as private clouds
  - it is therefore important to understand
    - what is cloud computing and the cloud as a platform for software release
    - the architecture of software systems in the cloud
  - in contrast, the release of a software system in its own data center is said on premises



## A definition of cloud computing

- The *cloud computing* is [NIST]
  - a processing model
  - which enables ubiquitous and convenient on-demand network access  *anywhere, anytime*
  - to a pool of computing resources (CPU, storage, networks, operating systems, services and / or applications) that are shared and configurable
  - which can be acquired and released quickly and dynamically
  - with minimal management effort, or in any case with minimal interaction with the service provider



## Service models

- Three *service models* main
  - *Software as a Service - SaaS* the final sw application
    - software applications
  - *Platform as a Service - PaaS* a stack of sw layers that allows you to develop and run your sw
    - platforms for developing and running applications
  - *Infrastructure as a Service - IaaS* servers, storage and connectivity (and VMs) resources
    - infrastructure services - such as servers (CPUs and operating systems), storage and connectivity

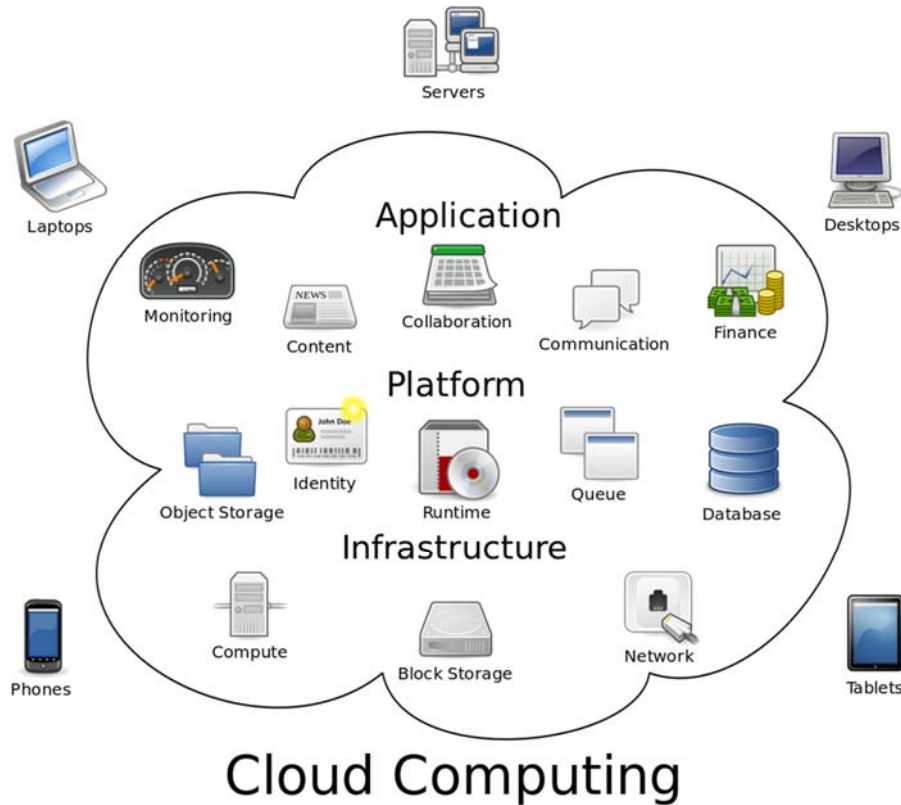


## Service models

- Some examples
  - Software as a Service - SaaS
    - Google Workspace (Gmail, Google Docs, ...)
    - Microsoft Office 365
    - Netflix
    - Salesforce.com applications (e.g., CRM)
  - Platform as a Service - PaaS
    - Google App Engine
    - Amazon Elastic Beanstalk
    - Microsoft Azure App Service
  - Infrastructure as a Service - IaaS
    - many Amazon Web Services (AWS) products / services - including Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3)



## Cloud computing and services



11

Cloud Computing

Luca Cabibbo ASW



## Cloud computing and services

- In the cloud, for *service* means
  - a very specific computational entity
  - managed by an organization - the *supplier* of the service
  - which is encapsulated - has a contractually defined interface, and its implementation is transparent to users
  - which can be accessed over the Internet - by a client that is the *user* (or *consumer*) of the service

12

Cloud Computing

Luca Cabibbo ASW



## \* Examples of cloud services

- We now present some examples of cloud computing services



13

Cloud Computing

Luca Cabibbo ASW




## - Google Workspace

- *Google Workspace* (SaaS)
  - "a single suite of products with all the best tools "
  - "everything you need to work at your best in a single package that works perfectly on your PC, smartphone or tablet "

 Gmail  
Email aziendale personalizzata

 Documenti  
Elaborazione di testi


 Keep  
Note ed elenchi


 Meet  
Conferenze video e voce

 Fogli  
Fogli di lavoro

 Apps Script  
Ottimizza il tuo modo di lavorare

 Chat  
Messaggistica per i team

 Presentazioni  
Strumento per la creazione di presentazioni

 Cloud Search  
Ricerca intelligente in tutti i prodotti Google Workspace

 Calendar  
Calendari condivisi

 Moduli  
Strumento per la creazione di sondaggi

 Drive  
Spazio di archiviazione sul cloud

 Sites  
Strumento per la creazione di siti web

- with versions for companies, schools, public administrations, ...

14

Cloud Computing

Luca Cabibbo ASW



## - Google App Engine



- *Google App Engine* (*GAE*) (PaaS)
  - "creates highly scalable applications on a fully managed serverless platform "
  - for dynamic web applications - built with Java, PHP, Node.js, Python, C #, Ruby, Go, ...
  - just add the code - then it will run on the Google Cloud
- features
  - automatic scalability and load balancing
  - authentication and security (sandboxing)
  - simplified administration
  - initially free - then you only pay for what you use



## - Amazon AWS

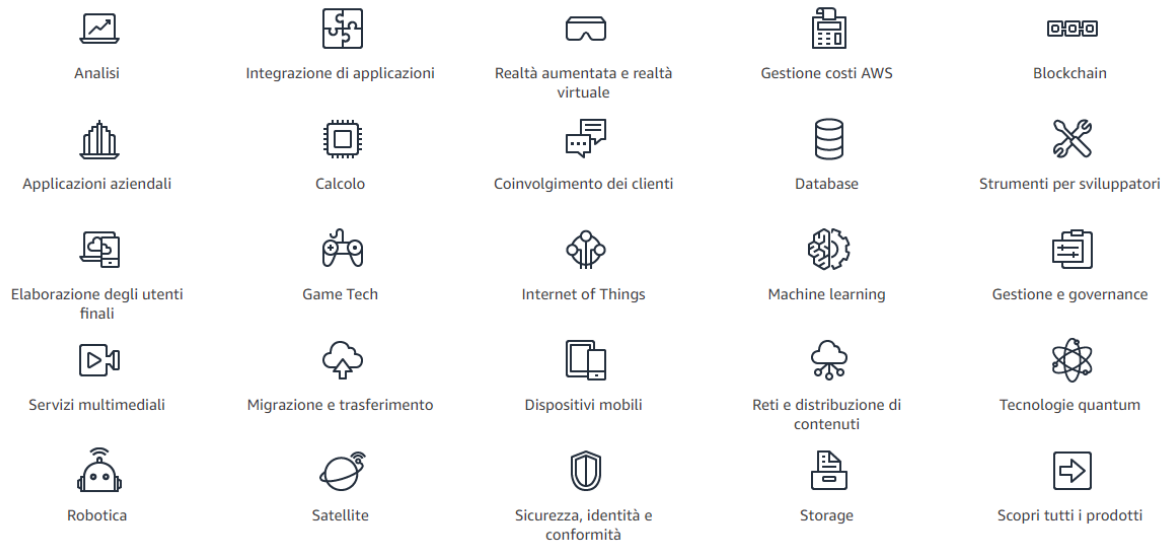


- *Amazon Web Services* (*AWS*) (IaaS and PaaS)
  - "AWS offers compute services, database storage, content distribution and other features, ideal to help you build sophisticated applications in a flexible, scalable and reliable way "
  - "to create any application - regardless of the business domain "
  - "to save time and money - without compromising scalability or security "
  - AWS offers both IaaS and PaaS services





## - Amazon AWS Service Categories



## - Some Amazon AWS services

- infrastructure services (IaaS)
  - compute resources - Amazon EC2 (virtual machines) and EC2 Container Service (container)
  - storage - Amazon S3 and EBS
  - databases - Amazon RDS and DynamoDB
  - networks - Amazon Virtual Private Cloud and Elastic Load Balancing
- platform services (PaaS) and application services
  - Scalable web applications - AWS Elastic Beanstalk and Lambda (serverless)
  - messaging and publish-subscribe - Amazon SQS and SNS
  - for developers (Dev) and operators (Ops) - AWS CodeCommit, CodeDeploy and CodePipeline
- but also analysis services, blockchain, Internet of Things, Machine Learning, Security and identity, ...





## Amazon EC2

- *Amazon Elastic Compute Cloud (EC2)* (IaaS)
  - computing capacity (virtual machines) in the cloud - Amazon Machine Instance (AMI)
    - several predefined AMI types - 1 to 96 64-bit vCPU (Intel Xeon), 0.5GB to 192GB RAM, EBS storage (HD or SSD, up to 8x1.9TB SSD) - and even bare metal instances
    - AMIs can be preconfigured with a variety of operating systems (e.g., Linux or Windows) and software (e.g., IBM DB2 or Oracle Database)
  - in the acquired AMIs it is possible to install the software you want
    - including its own applications
  - you can acquire multiple AMIs - in different Regions and Availability Zones (discussed later) - and network them (regulations)
  - integrates with most AWS services



## Amazon S3

- *Amazon Simple Storage Service (S3)* (IaaS)
  - cloud file storage
  - with a simple web interface (REST and SOAP) to store and retrieve your data
  - the data is persistent "objects" - from 1 byte to 5 GB
    - "opaque" objects - with a unique key
    - CRUD operations - on any number of objects
    - stored in Europe or the United States
  - support for different qualities
    - security - authentication, private and public objects, ACLs
    - different levels of reliability
  - it can be used as a storage mechanism for EC2



## Amazon EBS

- *Amazon Elastic Block Storage (EBS)* (IaaS)
  - block storage
  - an EBS instance is a volume - made up of blocks - that can be mounted on an EC2 instance
    - a volume can be used like any hard disk - with capacities from 0.5GB to several TB
  - support for different qualities
    - different levels of performance and reliability
    - selection of the zone in which the instance is allocated



## Amazon RDS and DynamoDB

- *Amazon Relational Database Service (RDS)* (IaaS)
  - relational databases on the cloud
  - allows access, with full functionality, to databases such as MySQL, Oracle, PostgreSQL or Amazon Aurora
- *Amazon DynamoDB* (IaaS)
  - a NoSQL datastore - that is, non-relational databases
  - a service with high and predictable performance, scalable and reliable



## AWS IaaS Services Discussion

- AWS infrastructure services can be used in an integrated way to create even complex execution environments in which to release your applications
  - provisioning of these compute resources can be done
    - interactively, through a web interface
    - in an automated way, through the use of scripts and additional AWS services to support the release of the software
  - multiple execution environments are possible
  - you pay for the use, with no upfront costs
  - 99.99% availability (monthly uptime) is "promised"
    - there are discounts if the availability in a period falls below 99.99% (10% discount) or below 99.0% (30%)



## Amazon Elastic Beanstalk

- *Amazon Elastic Beanstalk* (PaaS)
  - scalable platform for web applications
  - "AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go and Docker on common servers such as Apache, Nginx, Passenger and IIS "
  - "by simply uploading your own code, Elastic Beanstalk automatically manages implementation, from capacity provisioning and auto scaling to application health monitoring "
  - "there is no additional cost for Elastic Beanstalk: you only pay for the AWS resources you need to store and run your applications "



## Amazon ECS

- *Amazon Elastic Container Service (ECS)* (IaaS / PaaS)
  - a highly scalable service for containers
  - "Amazon Elastic Container Service (Amazon ECS) is a fully managed container orchestration service "
  - It allows you to easily run containerized applications in a cluster - managed by a set of Amazon EC2 nodes
- a service (partly platform, partly infrastructural) that avoids having to manage (install and size) an infrastructure for managing a container cluster



## AWS Lambda

- *AWS Lambda* (PaaS)
  - a fully managed serverless ("serverless") computing service
  - "AWS Lambda is a serverless compute service that allows you to run code without provisioning or managing servers "
- allows you to execute application code (Lambda functions) in response to certain events - e.g., HTTP requests via Amazon API Gateway, changes to S3 objects, or updates to DynamoDB tables
  - takes care of automatically managing the computational resources necessary to perform these functions
- "with Lambda, you can run code for any type of back-end application or service "- you can write Lambda functions in the language of your choice "- just load the code "



## AWS PaaS Services Discussion

- AWS platform services enable the development and release of applications based on a standardized software architecture in scalable standardized (predefined) execution environments
  - when an application is released in a PaaS service, an execution environment for the application is created (automatically and transparently) and then evolved - built as a set of IaaS resources
- IaaS and PaaS services provide abstractions at different levels
  - PaaS services are automatically implemented through the use and composition of IaaS services
- implications in the use of PaaS services in the development of a software application
  - greater focus on application functionality
  - transfer of control over the architecture and the execution environment



## AWS and DevOps

- AWS also provides a set of services to quickly and reliably build and release software in the AWS Cloud via DevOps practices - for example
  - infrastructure provisioning and management
  - application code management
  - automation of software release
  - monitoring



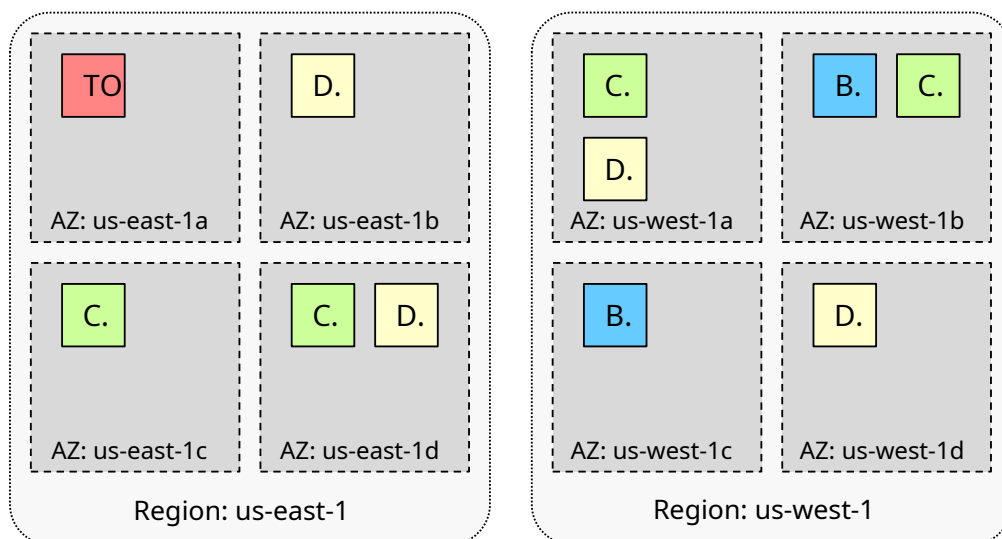
## Regions and Availability Zones

- The AWS Cloud operates in more than 70 Availability Zones spread across over 20 geographic regions around the world
  - a **region** is a geographic place where AWS data centers are grouped
    - e.g., EU (Ireland) (eu-west-1) and US East (Ohio) (us-east-2)
    - each region is partitioned into Availability Zones
  - a **availability zone (AZ, availability zone)** is a collection of one or more data centers with redundant (independent) power, network and connectivity in an AWS Region
- Regions and Availability Zones support high availability, fault tolerance, and scalability

**Oss: choosing the right region and AZ guarantees much fault tolerance**  
eg. if you want to create a replica it makes sense to do it in a different AZ while the first version should be on the AZ nearest you



## Regions and Availability Zones



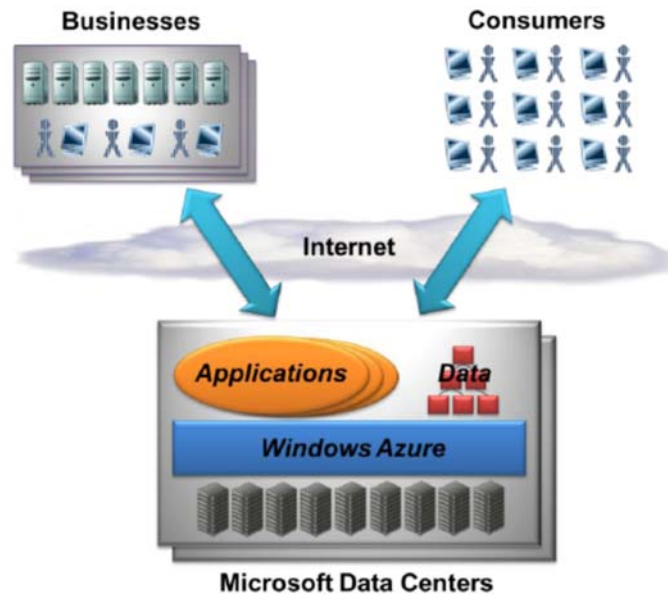


## - Microsoft Azure



### - *Microsoft Azure*

- a platform to enable the use of resources in the Microsoft cloud



## Microsoft Azure



### - *Microsoft Azure*

- Microsoft's cloud services
- "we give you the tools - you can change the world "
- "on-premises, hybrid, for multiple clouds or on edge devices
- build secure, future-ready cloud solutions in Azure "
- another generic and flexible cloud platform
  - provides the cloud version of many proprietary Microsoft solutions  
**Oss: not only windows based application: you can develop/use any**
  - today also provides numerous "open" solutions and not necessarily linked to the Microsoft world - for example, Linux virtual machines and orchestration of Docker containers





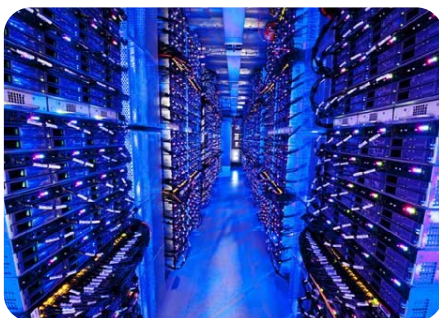
## - Some Microsoft Azure technologies

- Virtual Machines - Windows or Linux virtual machines - for running applications and storing data on Microsoft's data center computers in Azure we call: as IaaS
- Azure App Service - for the development of advanced web applications, running in the Microsoft cloud, on a fully managed platform as PaaS
- Azure SQL Database - a relational database manager, based on SQL Server
- Azure Kubernetes Service (AKS) - infrastructure services for container orchestration (lambda for AWS)
- Azure Functions - event processing with serverless code
- Visual Studio - flexible environment for developing applications for the cloud (is a PaaS you're a user but you develop or run app)



## - Some Microsoft cloud data centers (expanding)

- in Dublin, a traditional data center - 38500 m<sup>2</sup> (2013)
- in Chicago, a container-based data center (2010) - 112 containers, 1800-2500 server per container, 65000m<sup>2</sup> (equal to 9 football fields)





## - Salesforce.com



### - *Salesforce.com*

- "with cloud computing it is no longer necessary to install any software or hardware - you will get immediate operation and the first positive results in a much shorter time "
- "market leader in SaaS applications - everything you need to manage your business on the cloud "
  - applications (customizable), e.g., for sales, customer service and collaboration
  - accessible on the internet - both from PC and from smartphone or tablet
- also platform for the development of customized applications (PaaS)



## - Cloud Foundry



### - *Cloud Foundry* ([www.cloudfoundry.org](http://www.cloudfoundry.org))

- is a platform (PaaS) open source cloud computing
- makes it easier and faster to build, test, release and scale applications - offering the choice between different clouds, development frameworks and application services
- is an open source project, which is available in a variety of private cloud deployments and public cloud instances
- in practice, it has a container-based architecture
  - where you can run applications in any programming language
  - that can be released in any cloud, public or private - you can even move containers between clouds, without modifying the applications





- *Netflix* (SaaS)

- a software application for distributing films, television series and other entertainment content via the Internet
  - a microservices application released in containers in the Amazon cloud
- in practice, the company Netflix Inc.
  - is the provider of the Netflix SaaS service
  - is a consumer of Amazon AWS cloud services



\* Cloud computing

- The *cloud computing* is [NIST]
  - a processing model
  - which enables ubiquitous and convenient on-demand network access
  - to a pool of computing resources (CPU, storage, networks, operating systems, services and / or applications) that are shared and configurable
  - which can be acquired and released quickly and dynamically
  - with minimal management effort, or in any case with minimal interaction with the service provider
- In addition, this compute model has five essential features, three service models, and four deployment models



## - Essential features

- Five *essential features* for cloud computing
  - *services on request*
    - a consumer can acquire computing resources unilaterally and automatically
  - *network access*
    - computing resources are accessible on the Internet
  - *resource pooling*
    - a vendor's computing resources are pooled to serve many consumers in a multi-tenant model
  - *quick elasticity*
    - computing resources can be obtained quickly and flexibly
  - *measure of services*
    - the use of resources is controlled automatically, based on appropriate measures

39

Cloud Computing

Luca Cabibbo ASW



## - Service models

- Cloud computing involves three *service models*
  - *Software as a Service - SaaS*
    - the service is an application running in the cloud - the consumer is an end user of the application - the provider is who built it
  - *Platform as a Service - PaaS* *that allows you to build your app on it*
    - the service is a platform - the consumer is an application developer for the platform - the provider is who manages the platform
  - *Infrastructure as a Service - IaaS*
    - the service relates to virtual infrastructure computational resources - the consumer is an operator (Ops) for a virtual environment - the provider is the one who manages the infrastructure

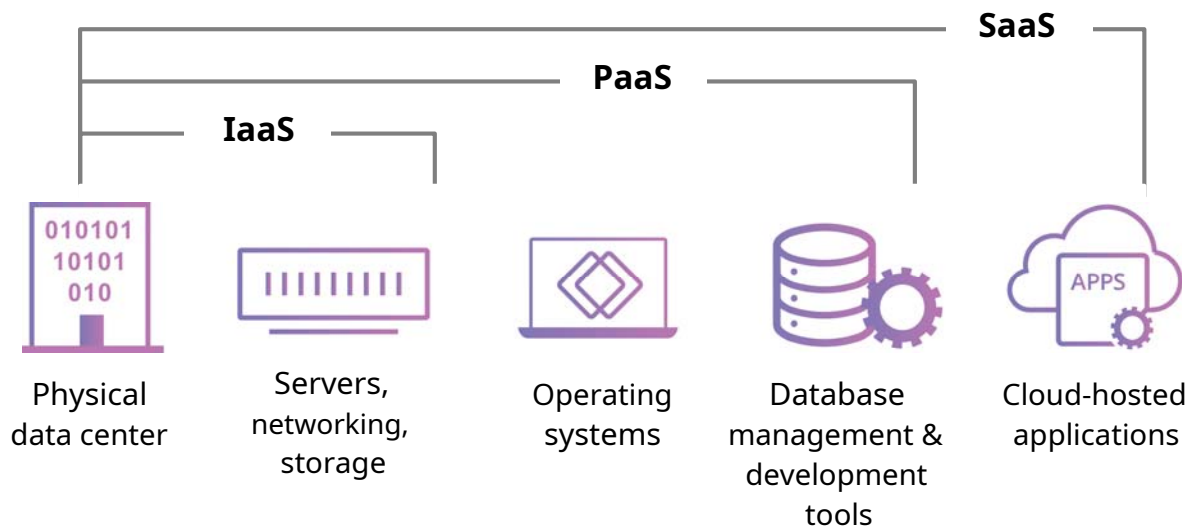
40

Cloud Computing

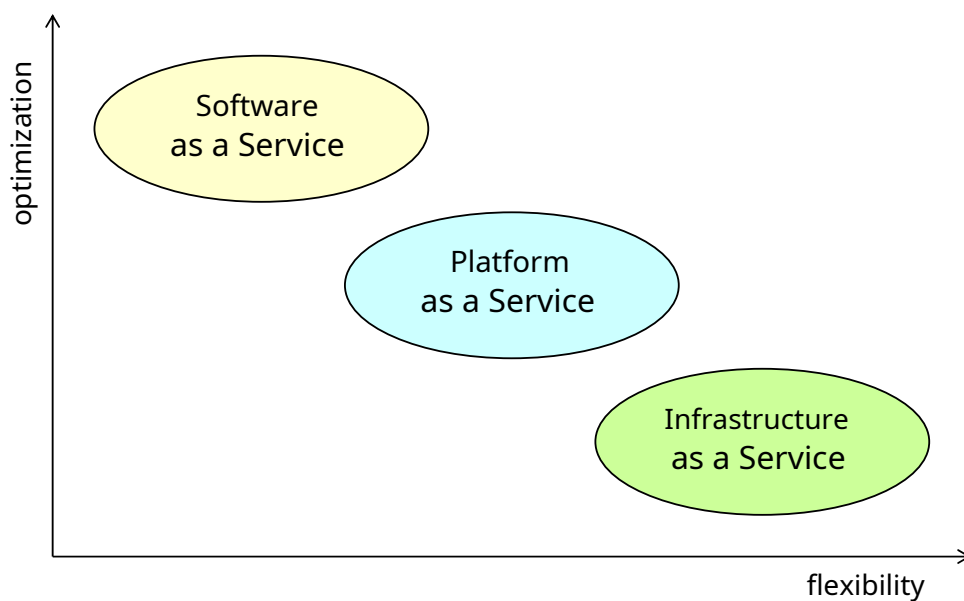
Luca Cabibbo ASW



## The SPI model



## The SPI model





## - Deployment models

- Cloud computing involves four *deployment models*

- **public cloud**

- the cloud infrastructure is made available to the public **users**

- **private cloud**

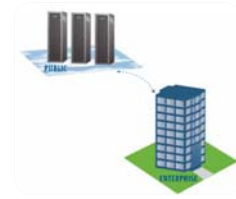
- the cloud infrastructure is managed for a single organization

- **hybrid cloud** **eg AWS**

- cloud infrastructure is the composition of two or more clouds

- **community cloud**

- the cloud infrastructure is shared by multiple organizations



## - Some alternative definitions



- Some alternative definitions of cloud computing

- The *cloud computing* is [Vaquero]

- a large set of resources (such as hardware, development platforms and / or services)
  - which are virtualized ,
  - and are easily accessible and usable
- these resources
  - they can be dynamically reconfigured - to adapt them to a variable load (they are scalable) - favoring an optimal use of resources
  - they are typically used on the basis of a pay-as-you-go model
  - they come with guarantees - based on customized SLAs - offered by their supplier



## Some alternative definitions



- Some alternative definitions of cloud computing
- The *cloud computing* [Armbrust et al.]
  - it refers both to applications delivered as services on the Internet and to the hardware and software in data centers that provide these services
    - the term *cloud* indicates the hardware and software present in these data centers
    - *Software as a Service (SaaS)* indicates applications delivered by a cloud
    - *utility computing* indicates the set of services provided by a cloud and made available to the public with pay-as-you-go methods
  - the *cloud computing* is given by the union of *SaaS* And *utility computing*



## - Further considerations

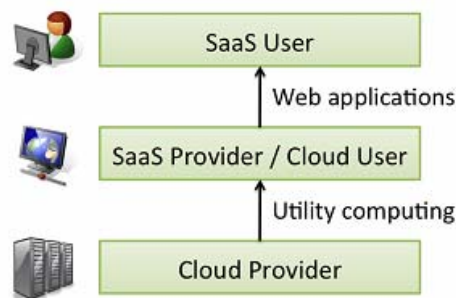
- Cloud means "cloud"
  - just as there are different types of clouds in the sky - so there are different types of services and features offered in the cloud
  - the internal implementation of these services is opaque (**black box**)
  - the boundaries between different services are often blurred
  - clouds can overlap
  - clouds can dynamically change shape





## Actors at play

- An organization or person can perform one (or more) of the following roles
  - *cloud provider* - utility computing provider (IaaS or PaaS)
  - *cloud user* - user (consumer) of utility computing (IaaS or PaaS)
- *SaaS provider* - provider of a SaaS application
  - a SaaS provider could also be a cloud user
- *SaaS user* - user (consumer) of a SaaS application



47

Cloud Computing

Luca Cabibbo ASW



## Actors at play

- As an example, let's consider John Smith, who is a client of the Netflix service - which is made by Netflix Inc. and released in the Amazon cloud.
  - Netflix is a SaaS service
  - Mario Rossi is a user (consumer) of the Netflix service (SaaS user)
  - Netflix Inc. is the company that developed the Netflix service - therefore Netflix Inc. is the provider of the Netflix service (SaaS provider)
  - Amazon is a provider of cloud services, of type IaaS and PaaS (cloud provider)
  - Netflix Inc. is a consumer of Amazon's IaaS cloud services (cloud user)

48

Cloud Computing

Luca Cabibbo ASW

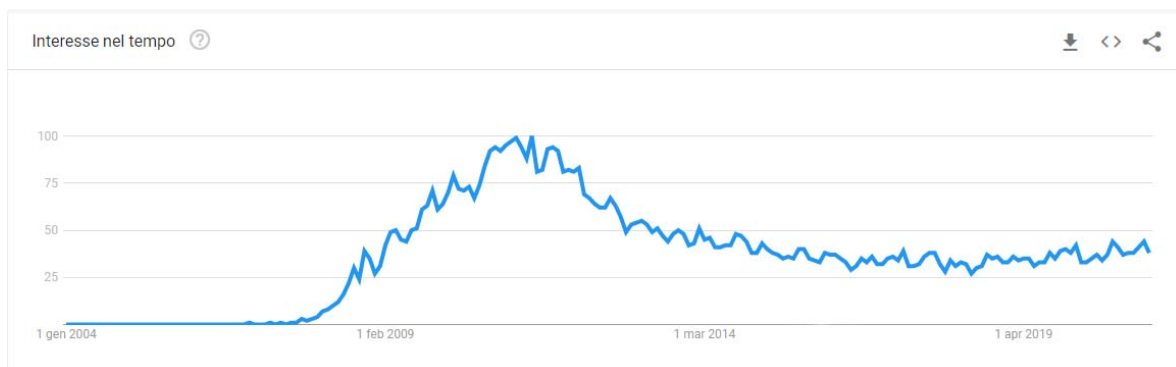


## Fields of application

- Some possible fields of application of cloud computing
  - web applications
  - desktop type software extension - e.g., Matlab and Mathematica
  - applications with temporary needs for large computing resources
  - prototyping
  - startup
  - single tasks
  - carrying out research activities



## When was cloud computing born?

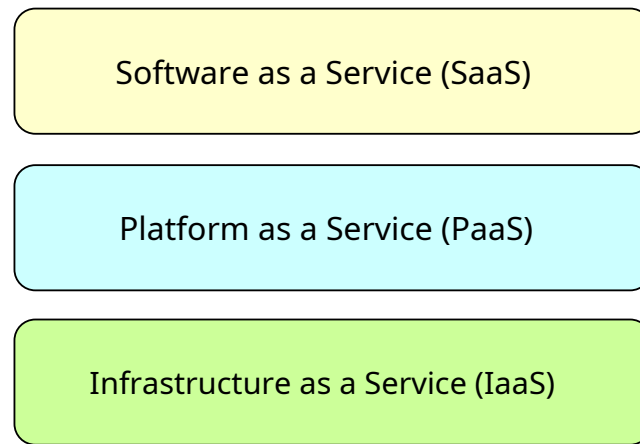


- Amazon was the first company to offer a rich set of services in the cloud
  - very broad Internet presence
  - experience in the automated management of a very large data center - configured for peak load
  - this "need" is converted into an opportunity - launching AWS in 2002 - then the service will take off in 2007

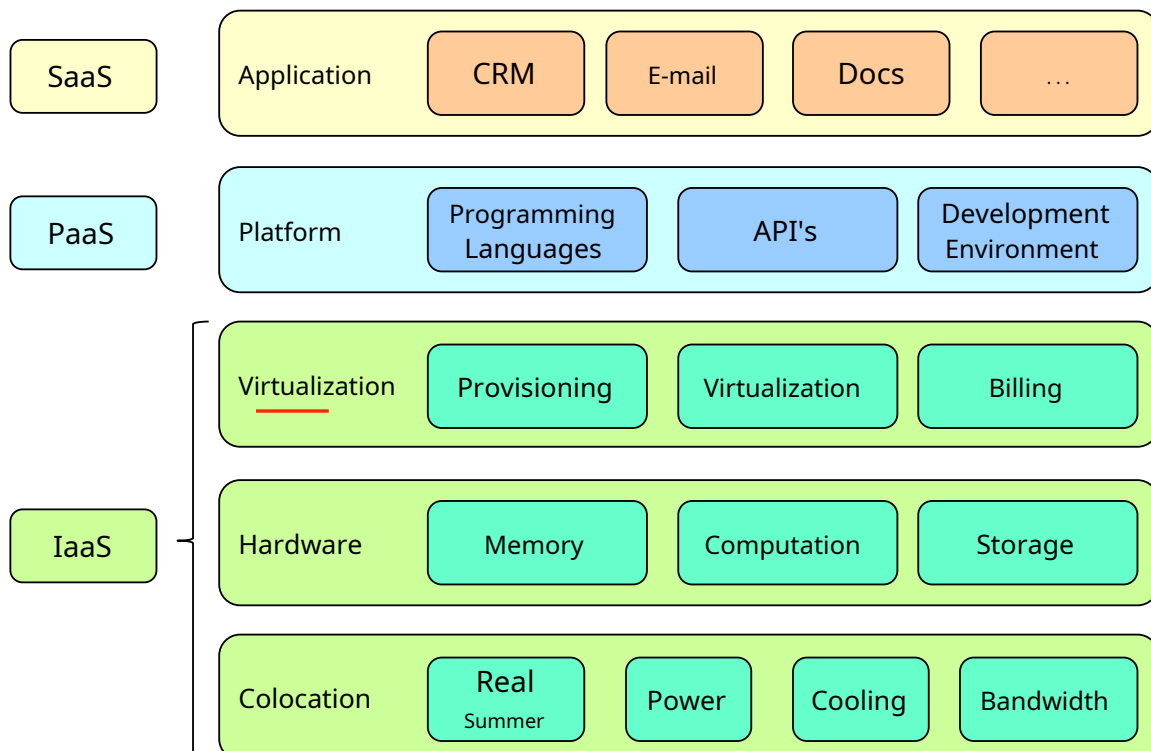


## \* Cloud architecture

- It is possible to think of the cloud in terms of a layered architecture
- by referring to the SPI model



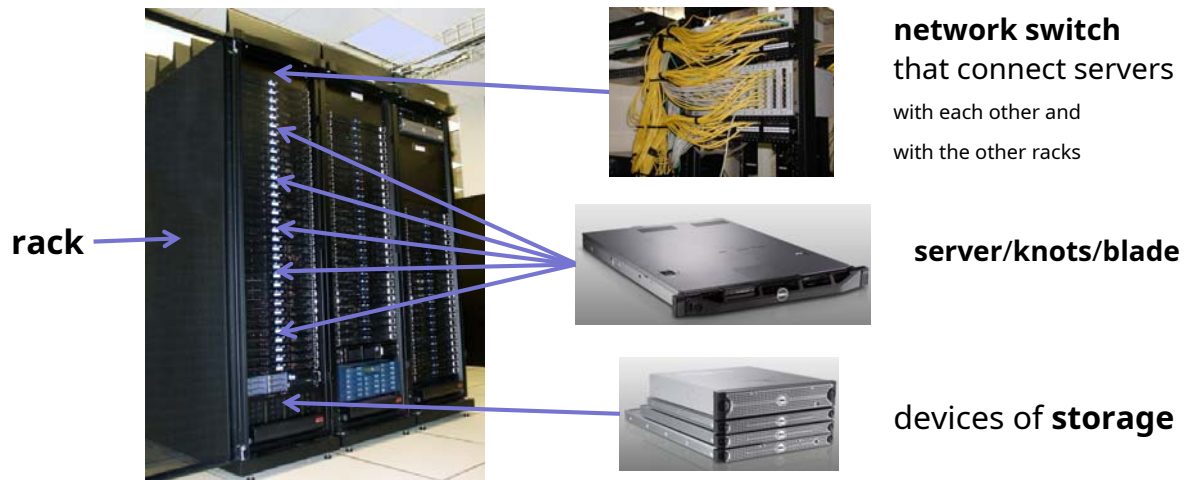
## Cloud architecture





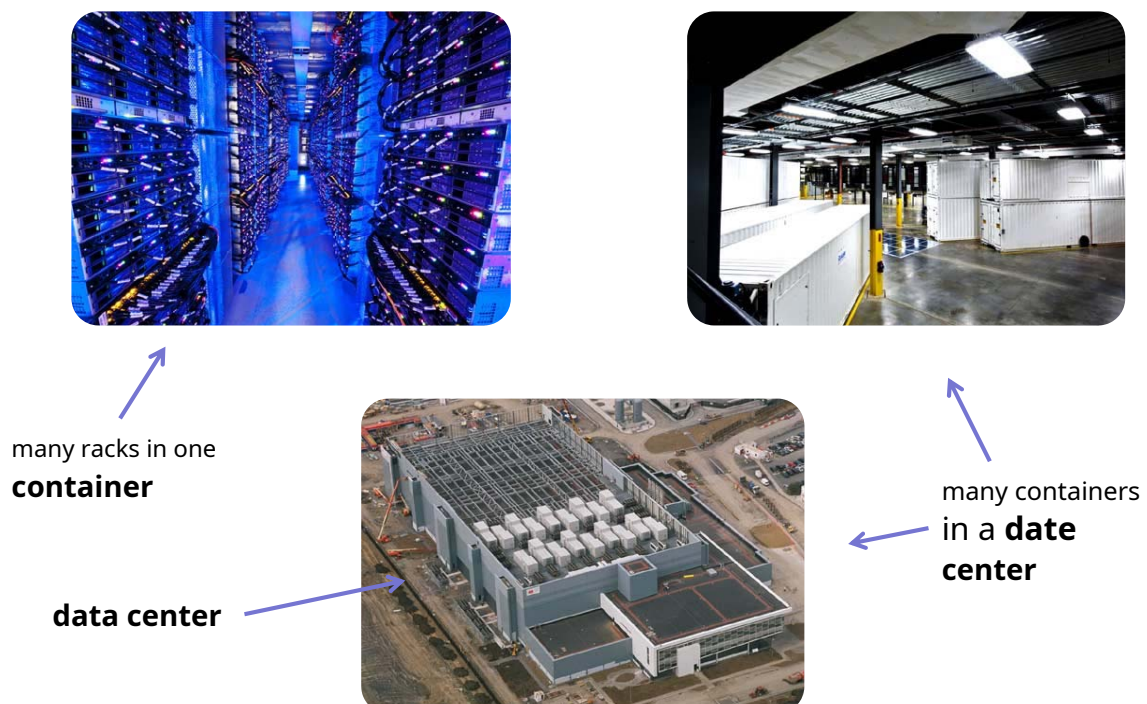
## Cloud-enabling technologies

### - Hardware



## Cloud-enabling technologies

### - Data center (co-location)





## Cloud-enabling technologies

### - Other technologies

networks



electric energy



conditioning



55

Cloud Computing

Luca Cabibbo ASW



## Cloud-enabling technologies

### - Many data centers



many data centers  
in an infrastructure  
from **cloud**



56

Cloud Computing

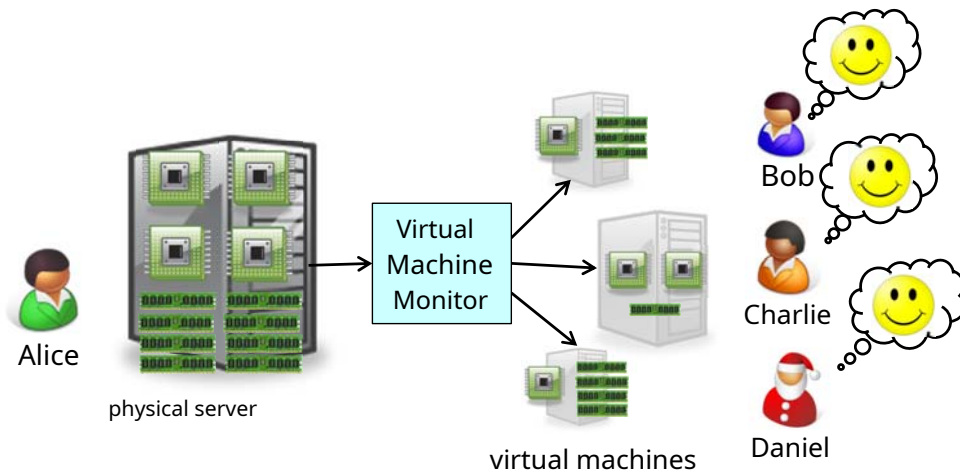
Luca Cabibbo ASW





## Cloud-enabling technologies

- Virtualization - and tools to manage virtual environments



## - Food for thought

- In the light of what has been studied so far, discuss how it is possible to offer simple cloud services (accessible through a simple web portal) with the following characteristics
  - a simple IaaS type service for virtual machines - allowing you to choose
    - number of vCPUs, amount of memory and OS to install
    - any pre-installed applications on the VM - from a predefined list
  - a simple PaaS type service (scalable, but without elasticity) for Tomcat-based web applications - which allows you to choose
    - the release unit - for simplicity, a single war file
    - the number of Tomcat servers on which to run the application



## \* Economics of cloud computing



- Cloud computing is a technological solution for its potential consumers
  - ?
    - but is it cost-effective for consumers of cloud services? and is it for cloud service providers?
    - What are the cases in which, for a consumer, cloud computing is preferable to the private management of one's own data center?

so cloud is convenient but is it always cost effective?



## Economics of cloud computing



- Consumer point of view - aspects to consider in an economic evaluation of cloud computing
  - pay-as-you-go model
  - cons:
    - allows the transition from a capital expenditure system (CAPEX) to a current expenditure system (OPEX)
      - physical resources
      - money
    - elasticity
  - adv:
    - allows you to mitigate the risks associated with incorrect sizing of the necessary infrastructure
    - economies of scale
      - suppliers can offer prices that are advantageous to consumers (and also advantageous for the suppliers themselves) | ?



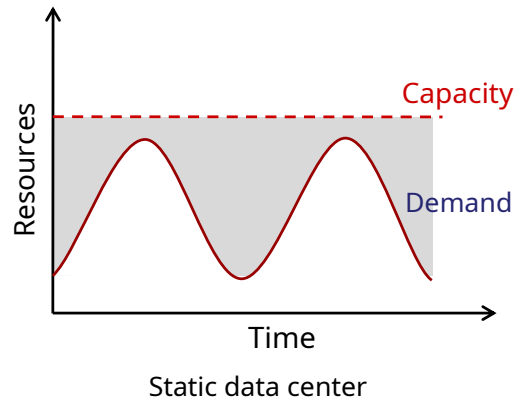


=> is fundamental to choose whether you need a high scalability or not

## Economy of the cc: elasticity



- Infrastructure sizing - when the demand for resources varies over time
  - provisioning for peak load



Unused resources

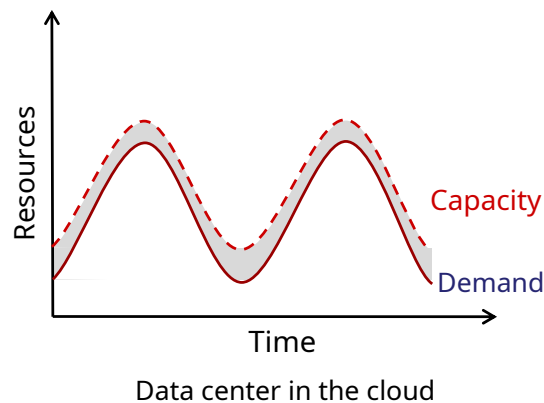
oss: it's a complex choice but the key anyway is the virtualization: if you develop your sws on MV's you will be able to switch from cloud to on premise and viceversa (otherwise NO)



## Economy of the cc: elasticity



- Infrastructure sizing - when the demand for resources varies over time
  - cloud computing and elasticity

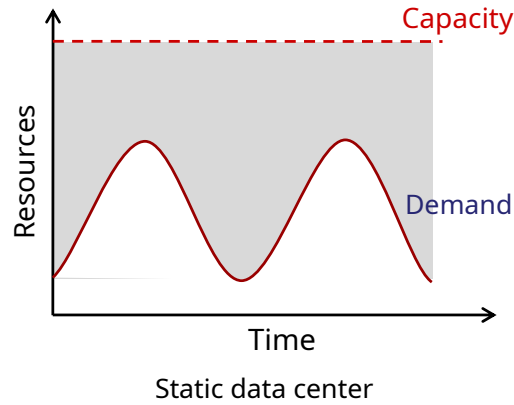




## Economy of the cc: elasticity



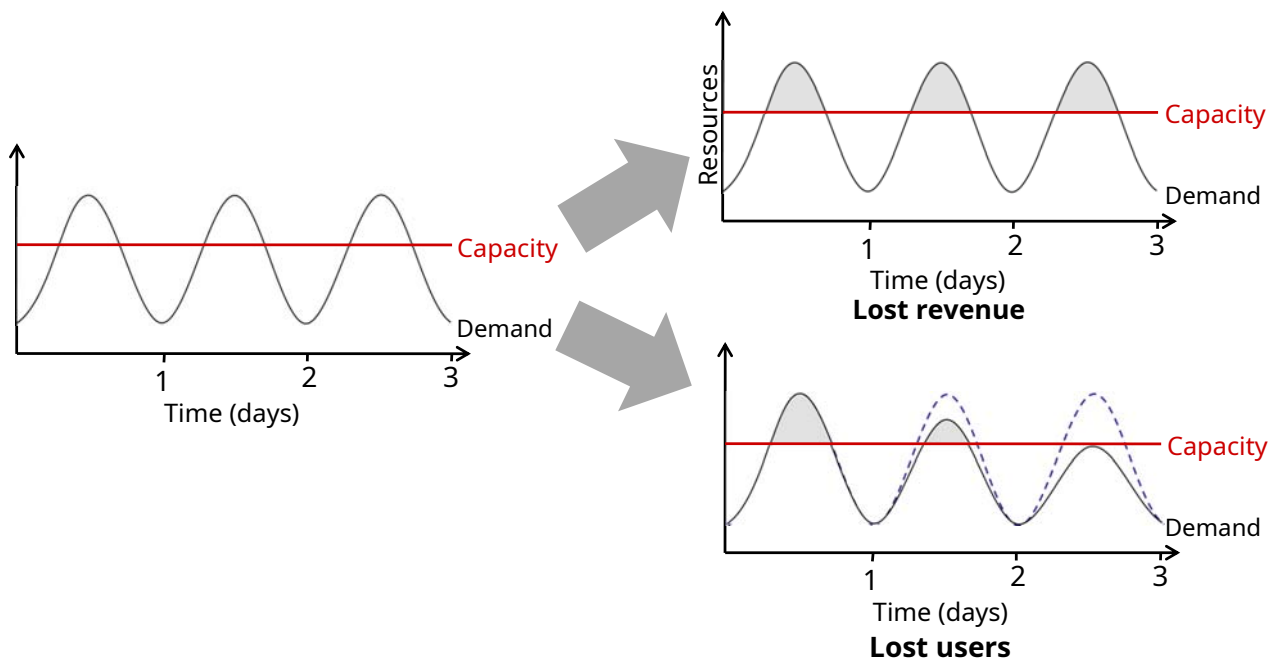
- Infrastructure sizing - when the demand for resources varies over time
  - over-provisioning



## Economy of the cc: elasticity



- Infrastructure sizing - when the demand for resources varies over time
  - under-provisioning





## Economy of the cc: elasticity



- Another case - performance of intensive tasks from a computational point of view
  - eg, batch analysis of large amounts of data
  - if the task is "parallelizable", then it may make sense to use a large number of servers for a short time - instead of just one server for a long time
    - at the same cost, it is possible to obtain the required results in a much shorter time!



## Economics of cloud computing



- Vendor's point of view - possible benefits
  - make profits - leveraging economies of scale
  - capitalize on their investments
    - Amazon - taking advantage of the remaining computing capacity (outside peak periods)
    - Google - leveraging existing infrastructure
  - defend a brand
    - Microsoft - for example, to sell .NET tools
  - strengthen relationships with their customers
    - for example, offer its customers a disaster recovery service in the cloud



### - Economies of scale

- a large data center can benefit from significant economies of scale compared to medium-small data centers - both for technological resources and for other infrastructures (electricity, heating, premises, ...)
- and it can sell these assets at lower costs than is typical for a small to mid-sized data center

Resource	Cost for a DC medium	Cost for a DC very big	Relationship
Net	\$ 95 / Mbps / month	\$ 13 / Mbps / month	7.1x
Storage	\$ 2.20 / GB / month	0.40 \$ / GB / month	5.7x
Administration	≈140 server / admin	> 1000 server / admin	7.1x



## - Discussion



### - Some Economic Consequences of Cloud Computing

- cloud service providers can build very large data centers, and sell computational resources and services profitably
- cloud service users can use or build software (scalable and available) on the basis of a pay-as-you-go model - with no upfront costs and no additional infrastructure management costs
- the cloud also supports innovation and competition
  - small software vendors can start their businesses with less reliance on outside investors
  - you can start a millionaire business by owning just a PC and an internet connection - and a lot of talent -
  - investors can target their investments more effectively



## \* Software systems for the cloud

- Today, it is increasingly common to release your software systems to the cloud - this raises a host of risks, opportunities and challenges
  - examples of risks are the release in a shared execution environment (security) and the use of "non-traditional" platforms and services in the cloud (eg, NoSQL systems)
  - an example of an opportunity is the release on elastic, scalable and available platforms
  - one possible challenge is to make an application or service actually scalable, available and modifiable
  - these risks, opportunities and challenges are often related to achieving quality - and therefore cloud software requires an architectural change



## Software systems for the cloud

- Cloud software requires an architectural change
  - the cloud software architecture must take into account the characteristics of the cloud and the cloud services that are intended to be used in the design, development, release and management of applications
  - it is important to understand
    - the cloud as a platform for software release
    - cloud software architecture - the *cloud-native software* is software designed and developed specifically for the cloud  
**again: it's fundamental to develop your sw as cloud-native**
    - how to release software in the cloud



## Software systems for the cloud

- Some requirements for cloud software systems
  - high availability: no service interruption
  - scalability: accept an increasing number of users or requests
  - modifiability: development cycles and rapid (continuous) feedback
  - support for mobile clients and multiple access devices
  - support for IoT (Internet of Things)
  - Big Data support



## Software systems for the cloud

- Some features of the cloud that you need to consider
  - use of shared resources and limited control over their location
  - constant changes in the execution environment
    - nodes and services may be stopped and restarted
      - the location of these nodes and services on the network can change over time, as can their number
    - it is possible that failures will occur or that latency in communication between distributed nodes may increase
    - changes in the environment must coexist with changes in the software
  - it is generally necessary to use the specific services offered by the cloud provider - which may be different from those traditionally used
    - and also to automate the management of the infrastructure and releases
  - it is often necessary to apply specific tactics and patterns



## \* Discussion

- It is useful to discuss cloud computing in the specific context of software delivery
  - the cloud offers several software release-oriented service models - providing infrastructure resources (IaaS) and platforms (PaaS) - supporting different options for execution environments
  - in addition, the cloud allows for different deployment models
  - the cloud also provides DevOps services - to manage software release in an automated way