Università degli Studi di Bologna
Scuola di Ingegneria

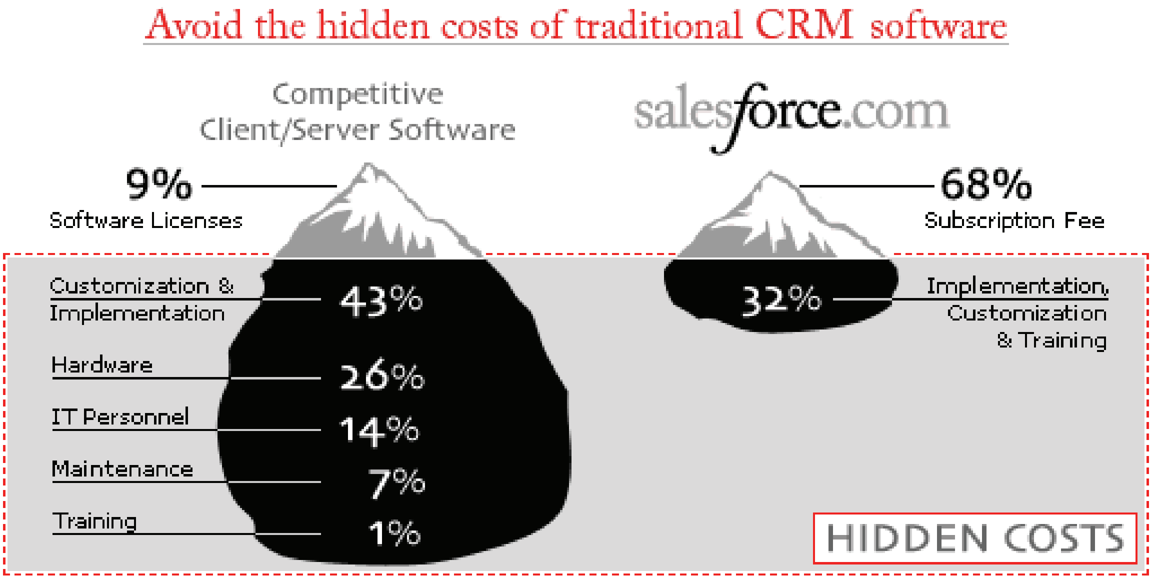
Corso di
Reti di Calcolatori M

Cloud

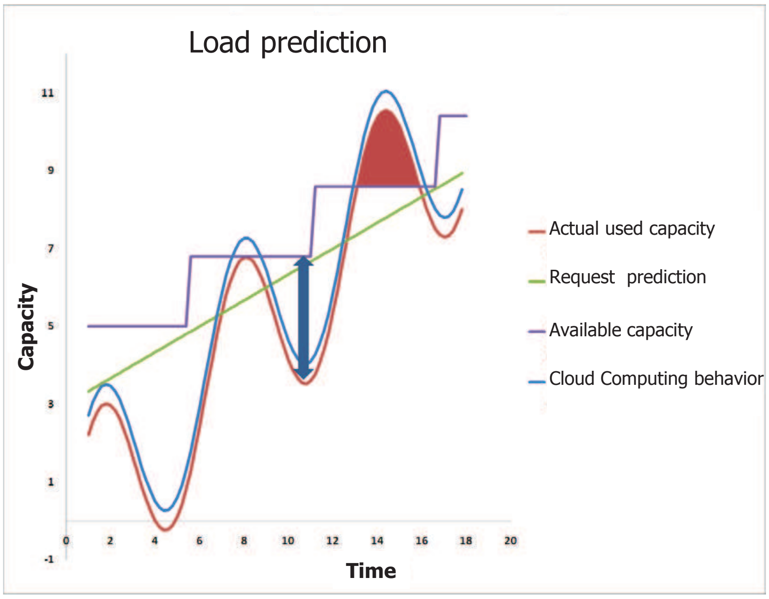
Antonio Corradi
Luca Foschini

Anno accademico 2014/2015

Motivations: Hidden Costs of IT

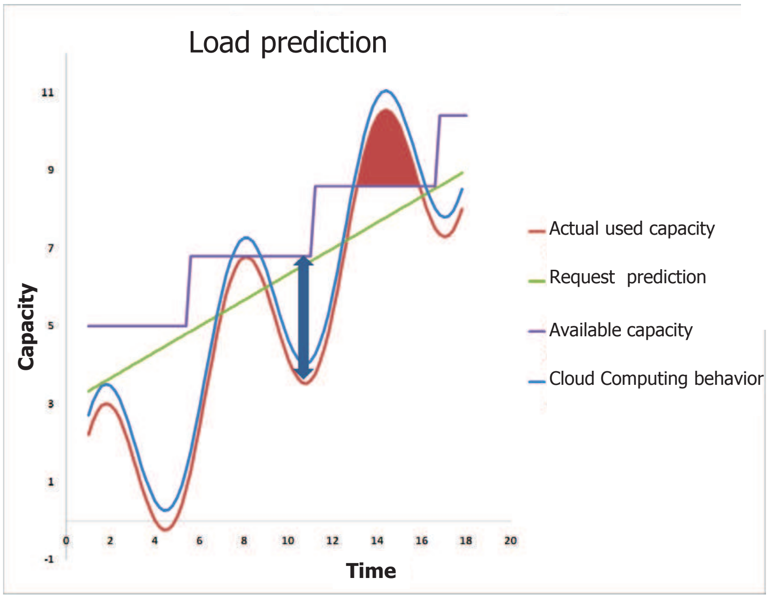


Flexibility and scalability in management



The main interest of a company in the field of Information Technology is to **provide a continuous service, rapidly adapting to economic needs changes**

Flexibility and scalability in management

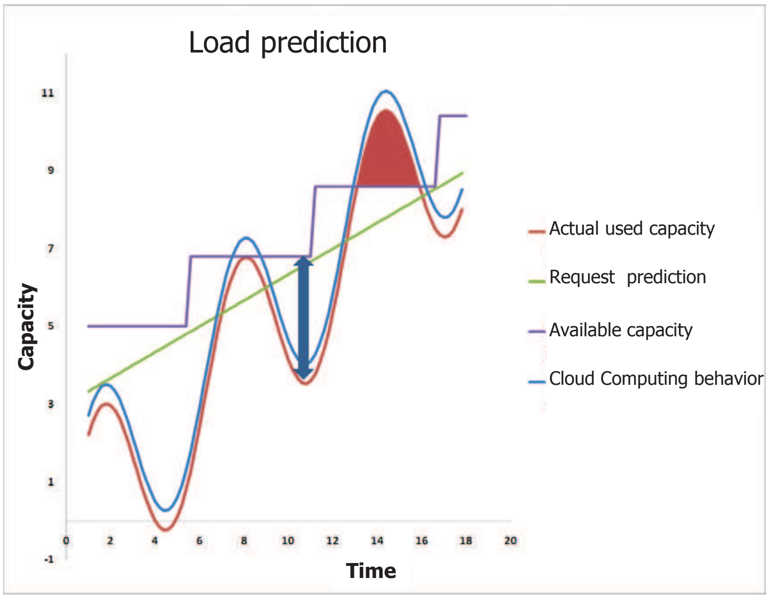


The area between red and purple lines represents the part of infrastructure that the company paid, but actually does not use

↓

Waste of **money**, **time** and **investments**

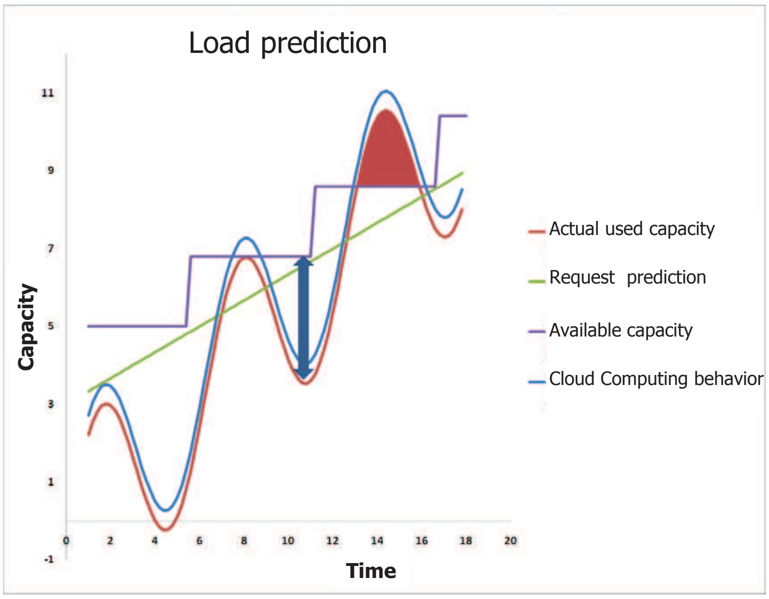
Flexibility and scalability in management



On the other side, if the company has an unexpected success, with a consequent growth of the customer base, computational capacity cannot scale

Slowdowns, service **interruption**, and **loss** of profit

Flexibility and scalability in management

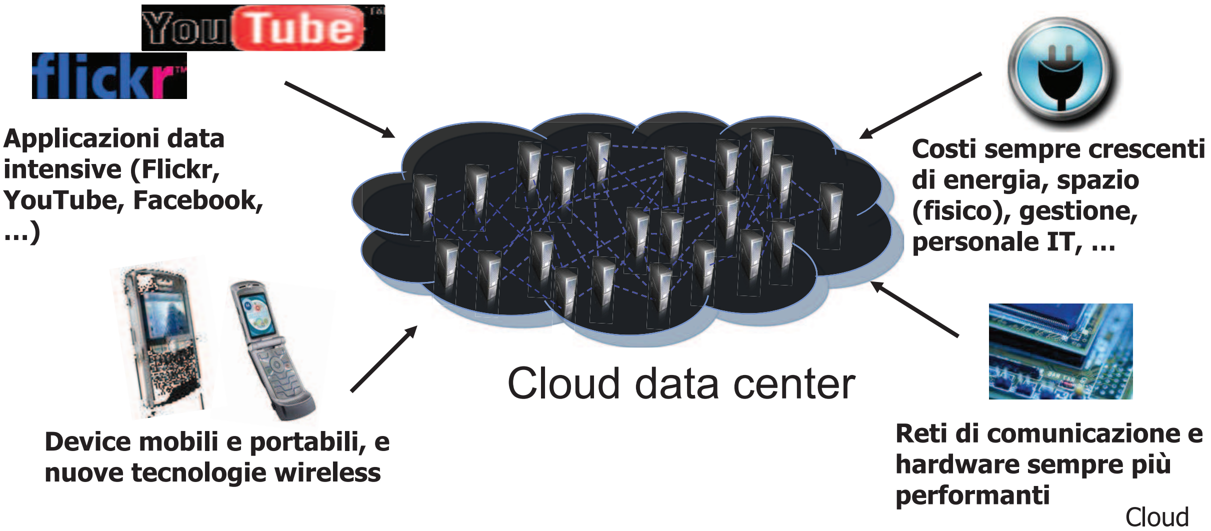


Prediction problems could be resolved by using an **highly flexible infrastructure**, with the ability of **constantly adapting** to computational load variations (**elasticity** ☺)

Cloud computing: problem space

"It starts with the premise that the **data services and architecture should be on servers**. We call it **Cloud computing** – **they should be in a 'Cloud' somewhere.** And that if you have the right kind of browser or the right kind of access, it doesn't matter whether **you have a PC or a Mac or a mobile phone or a BlackBerry or what have you – or new devices still to be developed – you can get access to the Cloud**…"

– Dr. Eric Schmidt, Google CEO, August 2006



Applicazioni data intensive (Flickr, YouTube, Facebook, …)

Costi sempre crescenti di energia, spazio (fisico), gestione, personale IT, …

Cloud data center

Device mobili e portabili, e nuove tecnologie wireless

Reti di comunicazione e hardware sempre più performanti

Cloud: fundamental concepts

- IT on demand pricing
- Best benefits in a reliable context
- Pool of virtualized computer resources
- Rapid live providing while demanding
- Systems on scaling architecture

Cloud keywords

on demand, reliability, virtualization, provisioning, scalability

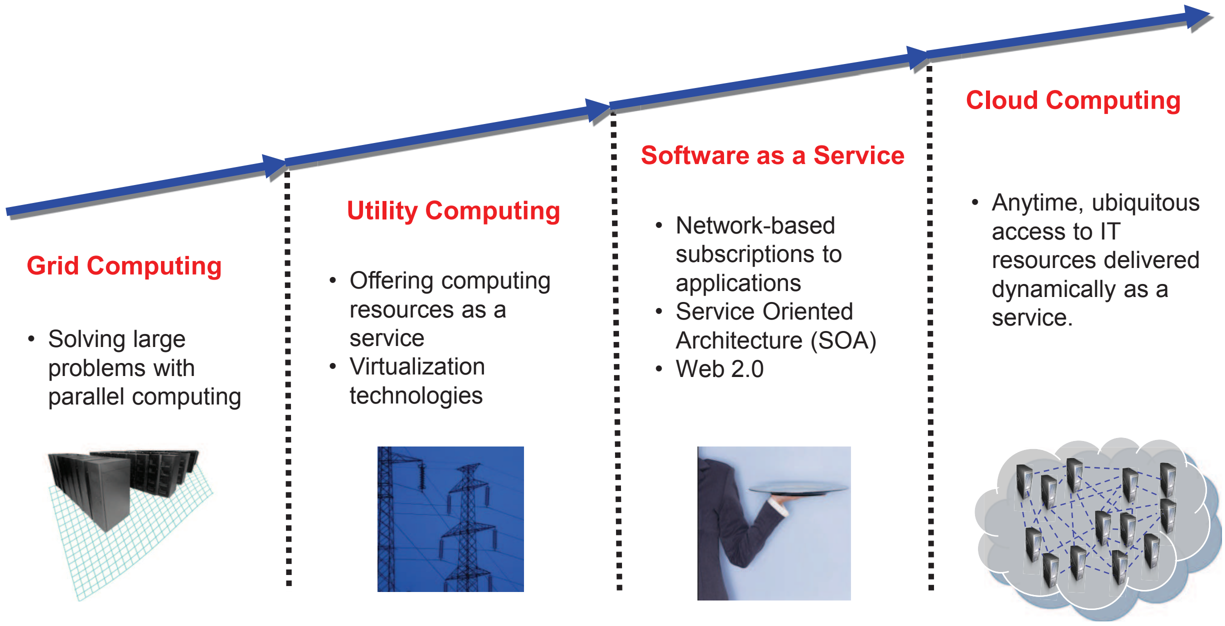
Cloud computing: some definitions

One Cloud is capable of **providing IT resources 'as a service'**

One Cloud is an **IT service** delivered to users that have:

- a **user interface** that makes the infrastructure underlying the service transparent to the user
- reduced **incremental management costs** when additional IT resources are added
- **services oriented management** architecture
- **massive scalability**

Cloud computing: a bit of history



Grid Computing

- Solving large problems with parallel computing

Utility Computing

- Offering computing resources as a service
- Virtualization technologies

Software as a Service

- Network-based subscriptions to applications
- Service Oriented Architecture (SOA)
- Web 2.0

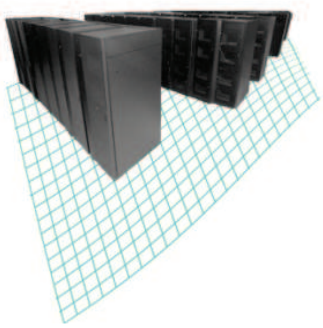
Cloud Computing

- Anytime, ubiquitous access to IT resources delivered dynamically as a service.

Before Cloud computing: GRID

Grid computing

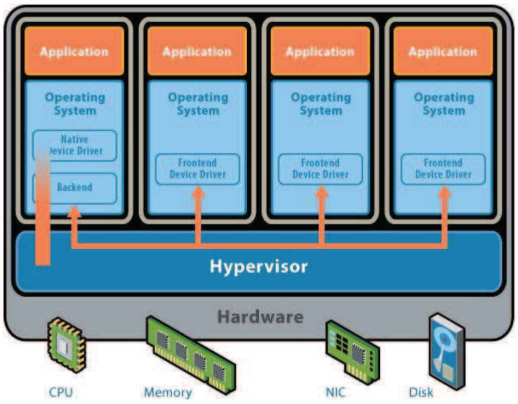
- Sharing of **heterogeneous resources** (computer, software, data, memory, computational power,, …) in **highly distributed environments** with the goal of **creating a virtual organization scalable** (**by need!**)
- Interfaces (for management), often **too fine grained, with low level of abstraction**, and **non self-contained** ☹
- Application areas very **limited and specific** (parallel computation for scientific, engineering scenarios, …)



Before Cloud: virtualization

Virtualization

- Technologies for **virtualization** (either system-based or hosted), as in a server farm: Vmware, Xen, …
- **Isolation** & **personalized infrastructure and/or SW platform** (O.S. and some additional applications)
- Tool for the **efficient management** of computing infrastructures (IBM Tivoli suite, Xen monitoring tools, …)



Before Cloud: Utility computing

Utility computing

- Huge computational and storage capabilities available from **utilities**, the same as for energy and electricity, pay per use
- "Computing may someday be organized as a public utility" - John McCarthy, MIT Centennial in 1961
- **Metered billing** (pay per use)
- **Simple to use interface** to access the capability (e.g., plugging into an outlet)

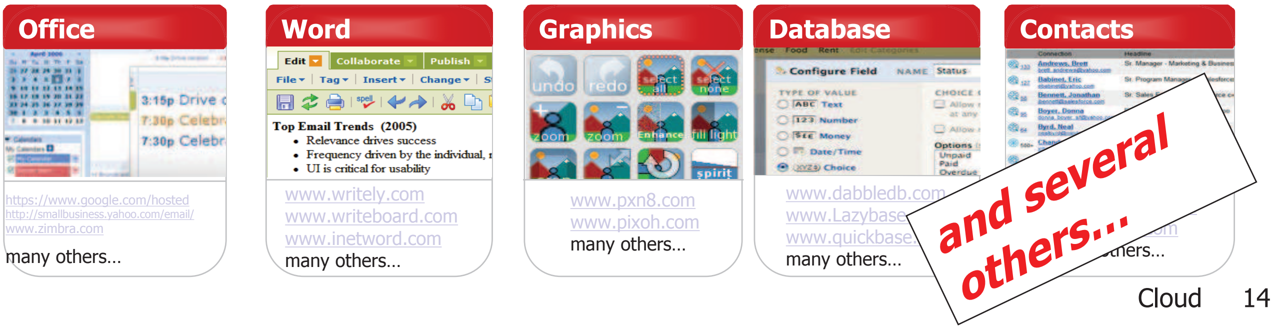
Traditional IT resources: in-house with internal management

On-demand utility: plug-in, subscription, pay-per-use

Before Cloud: Web 2.0

Web 2.0

- **Usage of asynchronous protocols not visible** to users to ask only **really required info** and not the **whole web pages**: **Asynchronous Javascript And XML (AJAX)**
- New way of **using Web services coupled with new applications easier to use**, **collaboration based** and **openly available**, **without requiring any installation** by interested users: new business model**, very, very cooperative** (Software as a Service ☺)



Office

https://www.google.com/hosted
http://smallbusiness.yahoo.com/email/
www.zimbra.com
many others…

Word

Top Email Trends (2005)
- Relevance drives success
- Frequency driven by the individual,
- UI is critical for usability

www.writely.com
www.writeboard.com
www.inetword.com
many others…

Graphics

www.pxn8.com
www.pixoh.com
many others…

Database

www.dabbledb.com
www.Lazybase
www.quickbase.
many others…

Contacts

om
thers…

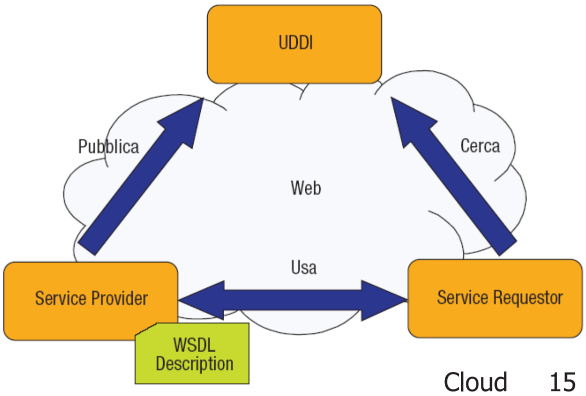
and several others…

Before Cloud computing: SOA

Service oriented Architectures (SOA)

- **Interaction** in terms of **abstract contact** of **offered and requested services**
- **Service** like **process, resource or application abstraction** that can be **standardized** like an **interface,** and can be **published** (**reusable**, **loose coupled** and **coarse-grained**, a black box, stateless, modular, …)

- **Web Services** as a Web architecture that fulfils various basic concepts of (abstract) SOA architectures (WSDL, SOAP, UDDI, …)



Before Cloud computing: SaaS

Software as a Service (SaaS)

Services are seen as **freely available appliances on the Web**, **without requiring any installation**

- **Service providers** own, manage, and provide applications
- Anytime, clients can **request and consume services**, in a **private or shared** manner, while providers are responsible for code and data definition
- Services are usually provided on a **pay-per-use** fashion

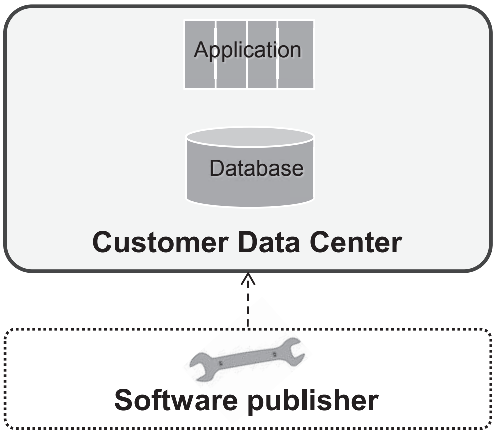
Software as a Service

- **Software ownership costs pushed to vendor** - hardware, software, system security, disaster recovery, maintenance, monitoring
- **Return to core competency** - organizations shift resources to core competencies, vendors focus on managing their SaaS
- **More efficient deployment** – instant evaluation, more collaboration between vendor and IT organization, much faster deployments
- **Modern, Web 2.0 interface** - drive technician usage and better customer interaction with IT
- **Eliminate shelfware & maintenance** - pay for what you use
- **Always on current version** – version-free software means the latest for the customer
- **SaaS homogeneity costs less** – one version for the vendor to support means lower costs for everybody

Application areas suitable for SaaS

- **ERP vertical business applications**, both specialized and very specific
- **General-purpose** applications without any adaptation (potentially sharable)
  - self-service provisioning and ad hoc personalization
  - applications available to several different users
- **Business B2B** applications **domain specific**
  - no need of third party hosting and involvement
- **Customer/Supplier** applications
  - applications where most of users and access is externally to the organization and where ubiquitous access via Web is critical and intrinsic
- **Business** applications **even critical**, but not the **core business ones**

Models: traditional on-premise deployment at the client side



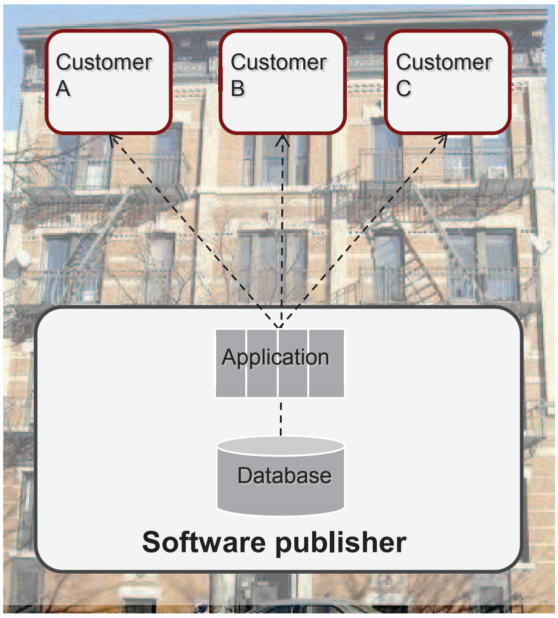
Details

- Full ownership
- Significant implementation costs
- Customizable
- Difficult to upgrade / mantain

Examples

- HP Service Manager
- BMC Remedy
- CA Service Desk
- EMC Infra

Shared SaaS application (multi-tenant)



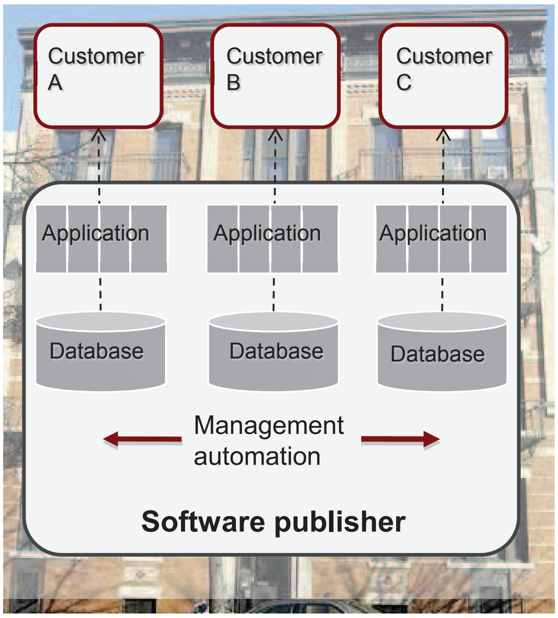
Details

- Software maintained and developed by provider
- Many clients to one application set
- Low customization

Examples

- Salesforce.com
- Workday

Private SaaS application (single-tenant)



Details

- Software maintained and developed by provider
- Customers receive their own application and database
- Extensive customization

Examples

- Service-now.com
- InteQ

Enhancing the SaaS models

Some increasing resources models for providing some resources as a service, …aaS

SaaS Software as a Service

- Resources are simple **applications available** via remote Web access

PaaS Platform as a Service

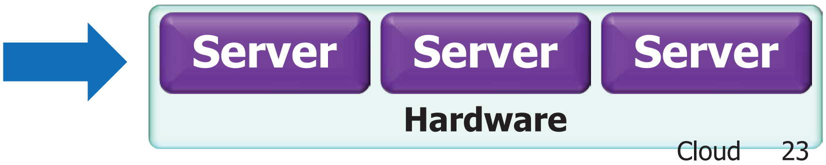
- Resources are **whole software platforms available** for remote execution, i.e., several programs capable of interacting with each other

Iaas Infrastructure as a Service

- Resources are enlarged **in a wider and complete way,** from **hardware platforms**, to **operating systems,** to **support** to final **applications**: usually via virtualization up to **Cloud Computing**

Layered Architecture: IaaS, PaaS & SaaS

- Below the real architecture: **hardware** components & software products



Server

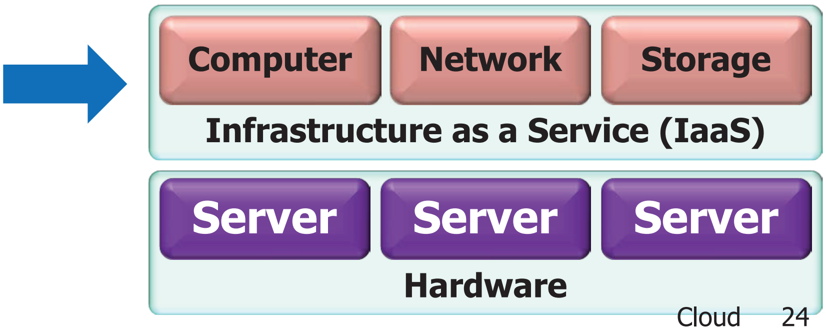
Server

Server

Hardware

Layered Architecture: IaaS, PaaS & SaaS

- **Infrastructure**: layer to enable the distribution of Cloud services, typically realized by a virtualization platform



Computer

Network

Storage

Infrastructure as a Service (IaaS)

Server

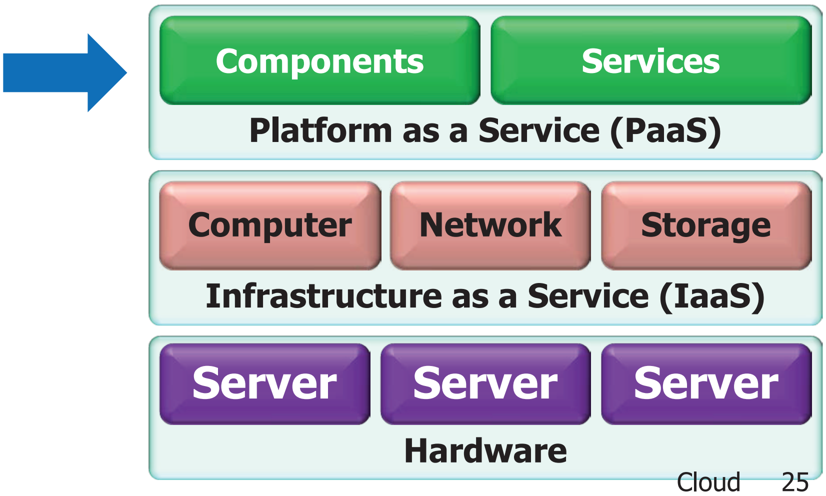
Server

Server

Hardware

Layered Architecture: IaaS, PaaS & SaaS

- **Platform:** layer to provide to upper layers a set of services and components remotely available



Components

Services

Platform as a Service (PaaS)

Computer

Network

Storage

Infrastructure as a Service (IaaS)

Server

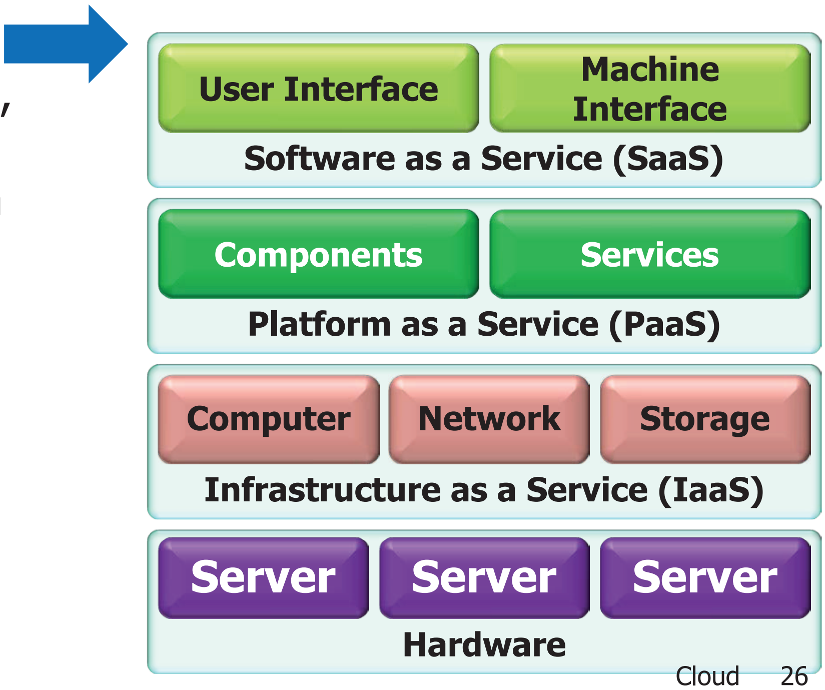
Server

Server

Hardware

Layered Architecture: IaaS, PaaS & SaaS

- **Application**: layer to install applications, to be available via Web and Internet via Cloud API



User Interface

Machine Interface

Software as a Service (SaaS)

Components

Services

Platform as a Service (PaaS)

Computer

Network

Storage

Infrastructure as a Service (IaaS)

Server

Server

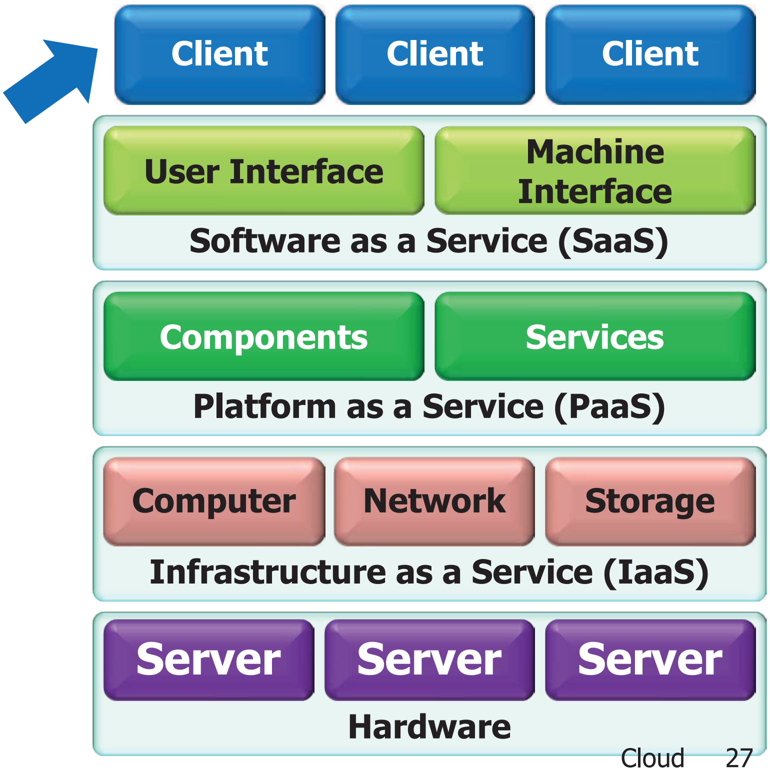
Server

Hardware

Layered Architecture: IaaS, PaaS & SaaS

- **Client** software to get access to the system. Those applications are executed on the **client physical platforms** (remote computers), owned by the final remote user.

  They can communicate with the Cloud via the **available interfaces**



Client | Client | Client

User Interface | Machine Interface

Software as a Service (SaaS)

Components | Services

Platform as a Service (PaaS)

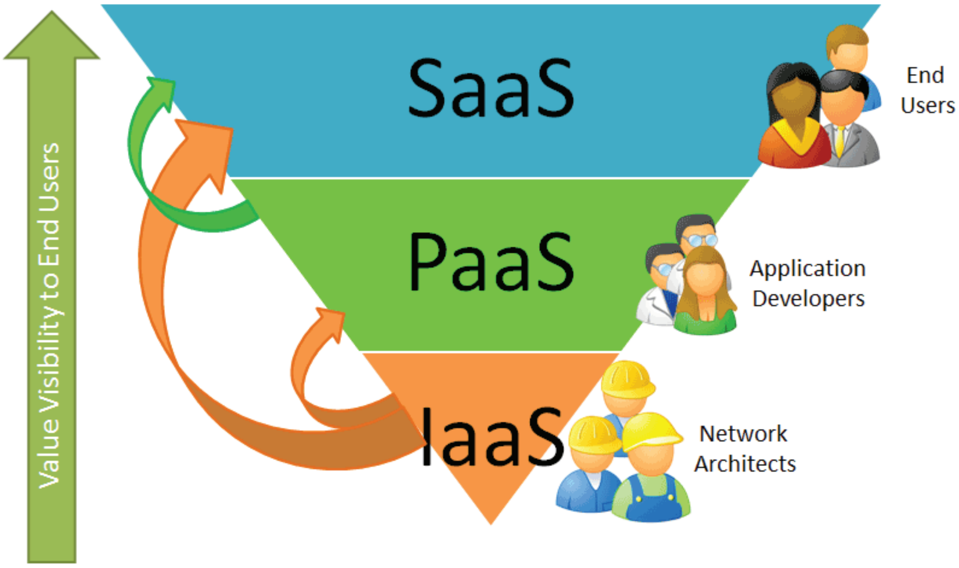
Computer | Network | Storage

Infrastructure as a Service (IaaS)

Server | Server | Server

Hardware

Layered Architecture: main actors



Some SaaS & *aaS examples

Some very well known examples

SaaS - single services

From desktop applications: **Google Apps** (Gmail, Google calendar & docs), **Microsoft Windows live** (Hotmail, Messenger, …) to search engines, Google, Yahoo,

All **social networks**, such as Facebook, LinkedIn, Twitter, …

PaaS - Web services and development platforms

Services available internally to and interacting with other applications, as **Google App Engine (GAE)**, **Google Maps**, **Microsoft Azure**, …

Iaas - some emerging experimental infrastructures

Several examples, with virtualization services , **Amazon Web Services** (S3), **Elastic Computing Cloud** (EC2), **OpenStack**, to several **management and monitoring desktop** to control execution, Sun global desktop, Zimdesk, …

Technology wrap up

SaaS

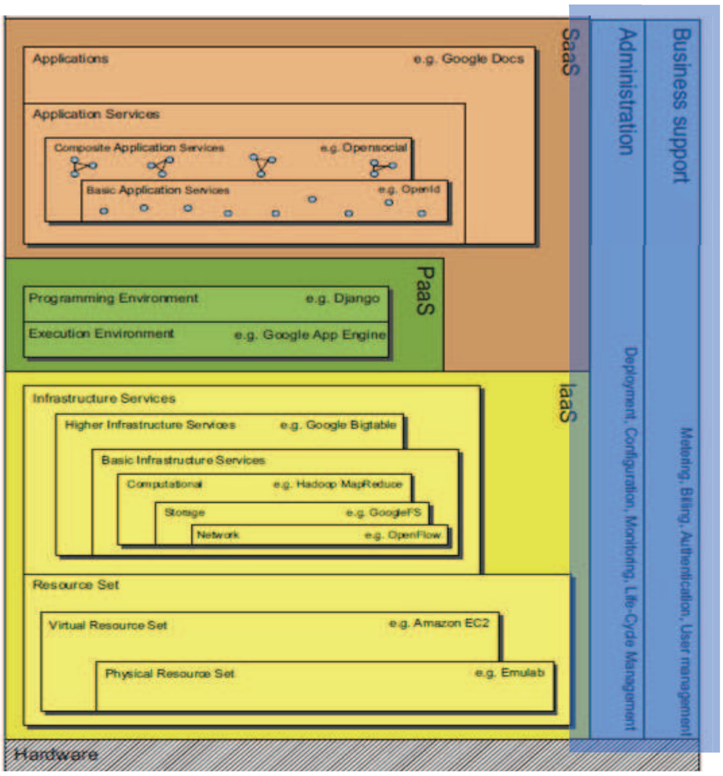
- Google docs

PaaS

- Google App Engine

IaaS

- Google Bigtable
- Hadoop MapReduce (Yahoo)
- GoogleFS
- Amazon EC2
- OpenStack



Vertical support functionalities

Cloud differences from…

Grid Computing

- A cloud is more than a **collection of computer resources** because a cloud provides a mechanism to manage those resources
  - Provisioning, change requests, workload balancing, monitoring
- Cloud computing is an infrastructure that sits on top of a datacenter for efficiency

Utility Computing

- Service that allows users to **deploy, manage, and scale** online services using the provider's resources and pay for resources they consume
- Users want to be in control of what runs on each server
- **Cloud users** want to avoid infrastructure. The provider is in complete control.

SaaS

- Cloud defines two models: who **manages** SaaS **vs.** who **uses** SaaS
- Cloud **shortens time-to-market** and **eases the composition of service ecosystems**
- Cloud can have strong **organizational** consequences → more **cooperative** and **continuously growing** model

Cloud… introduces new problems

Reliability and availability

- We rely on external resources (public Clouds) owned by third-party companies
- **Amazon US outage (21→24 April 2011)**, some papers…
  - "Amazon EC2 Outage Hobbles Websites: Worst Cloud Computing Disaster"
  - "Amazon's lengthy cloud outage shows the danger of complexity"

Security and legislative issues

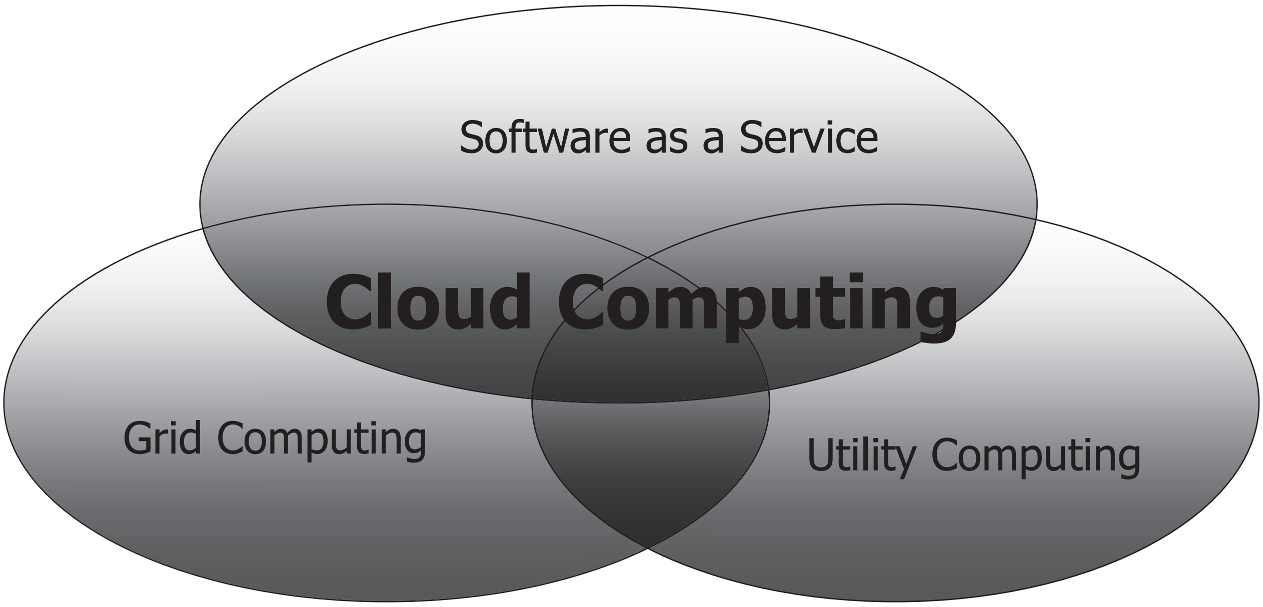
- Data outsourcing in Cloud
- Dematerialization → **where are my data???**

  (with emerging partial solutions…)

Service management and integration

- Possibility to recreate **SaaS silos**
- Integration of different Cloud systems → **standard APIs**, VMs migration protocols, etc.

Evolution of Cloud Computing



Software as a Service

Cloud Computing

Grid Computing

Utility Computing

Business & Technology

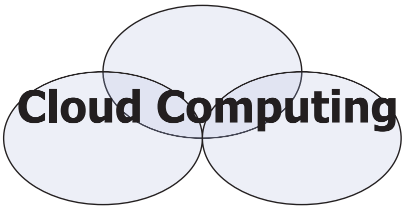
Virtualization
Scalability
Grid Computing
…

SaaS **UaaS**

Technology

Business

End users



Cloud Computing

Cloud Deployment Models

Typically three models

- **Private cloud**
  - enterprise owned or leased
- **Community cloud**
  - shared infrastructure for specific community
- **Public cloud**
  - sold to the public, mega-scale infrastructure
- Hybrid cloud
  - composition of two or more clouds

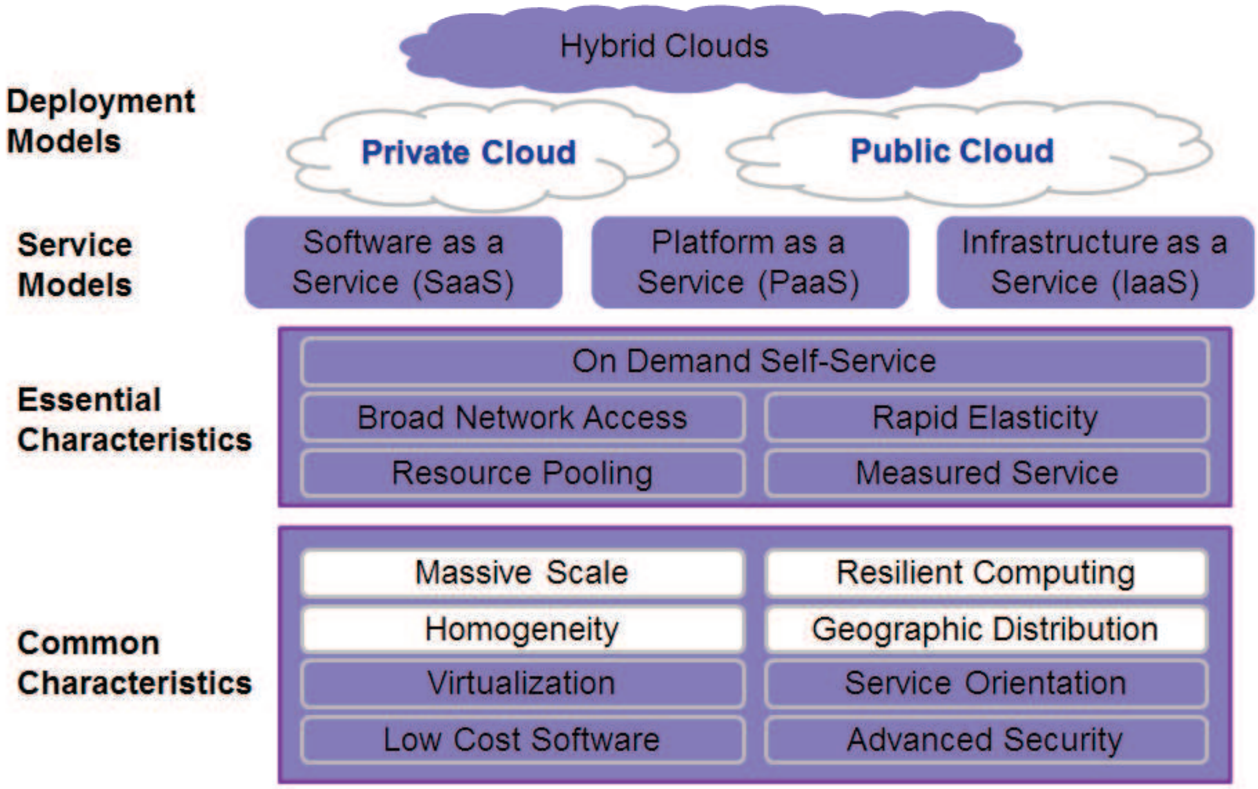
Cloud computing: reality check

- **Google docs, salesForce, …** widespread and very used Web applications (Web 2.0)
- **Google App Engine** Google App Engine, sandbox for management e security, several languages(Python e Java)
- **HP/Yahoo/Intel Open Cirrus Test Bed**: virtualized images, Xen, simple SLA console
- **Amazon Elastic Computing – EC2**: Amazon WS (SLA), Amazon Machine Image (DB+Software and middleware+OS) e Xen, Dynamo
- **Openstack**: virtualized images (DB+Software e middleware+OS), Xen, Swift (object storage), Cinder (block storage), Neutron(virtualized network)
- **IBM Cloud**: virtualized images (DB+Software and middleware+OS), Xen, Tivoli (monitoring and management), simple SLA console
- **Microsoft Azure**: Microsoft solution

Cloud Key Goals: Infrastructure Perspective

- How can we provide flexible compute resources quickly to promote **rapid prototyping**?
- How do we deploy applications that **scale up** to meet increasing demands over time?
- How do we manage 100,000’s of machines with **minimal human intervention**?
- How can we make the most **efficient** use of all the compute resources in a data center?

NIST STANDARD CLOUD



Deployment Models

Hybrid Clouds

Private Cloud

Public Cloud

Service Models

Software as a Service (SaaS)

Platform as a Service (PaaS)

Infrastructure as a Service (IaaS)

Essential Characteristics

On Demand Self-Service

Broad Network Access

Rapid Elasticity

Resource Pooling

Measured Service

Common Characteristics

Massive Scale

Resilient Computing

Homogeneity

Geographic Distribution

Virtualization

Service Orientation

Low Cost Software

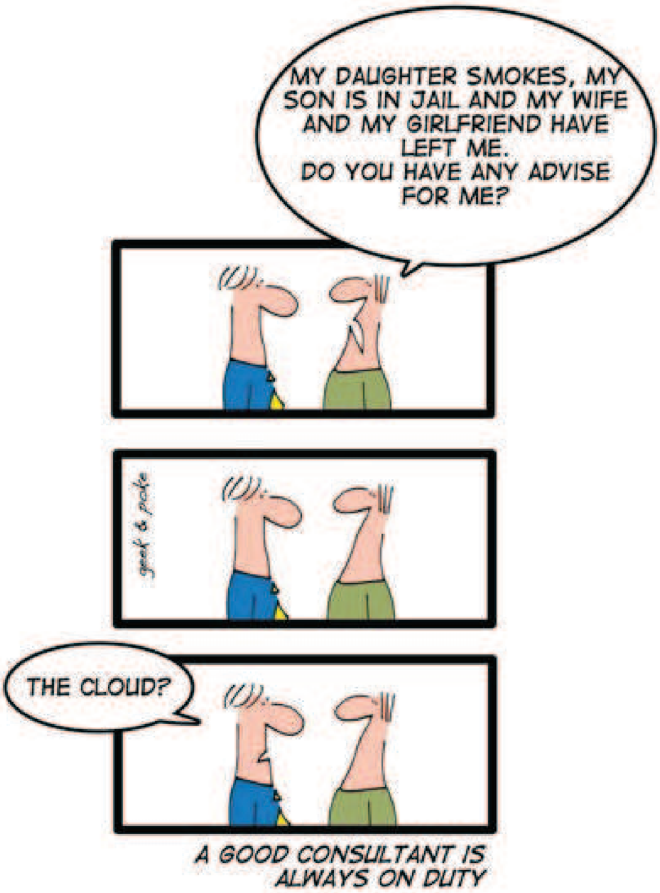
Advanced Security

National Institute of Standards and Technology www.nist.gov/

The fog has (partially) gone…

… and Clouds are letting out the light!

Cloud for everything


MY DAUGHTER SMOKES, MY SON IS IN JAIL AND MY WIFE AND MY GIRLFRIEND HAVE LEFT ME. DO YOU HAVE ANY ADVISE FOR ME?
geek & poke
THE CLOUD?
A GOOD CONSULTANT IS ALWAYS ON DUTY

Non exactly for everything ☺