

# Cloud Computing – an introduction

January 30, 2012

Netzprogrammierung

(Algorithmen und Programmierung V)

# Our topics today in more detail

What is cloud computing - old wine in new bottles?

From the first idea to the BIG business.

Why and when is Cloud Computing useful?

The core concept of Cloud Computing – virtualization.

The cloud computing stack: SaaS, PaaS, IaaS.



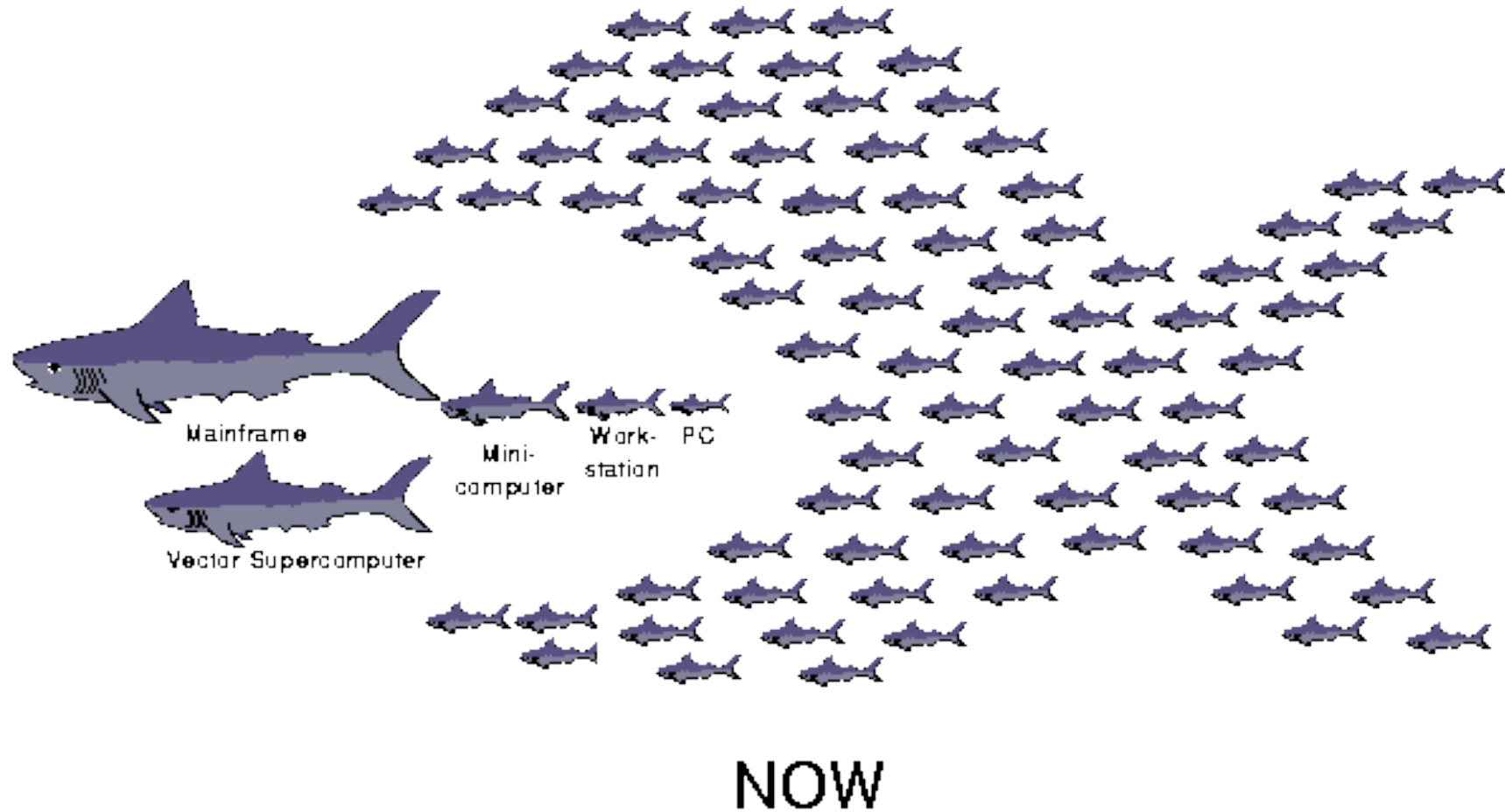
# Cloud Computing **Introduction**

<http://www.flickr.com/photos/pagedooley/2511369048/>

# Motivation

1. **Google**: Enables users to search the world's information, including webpages, images, and videos.
2. **Facebook**: A social utility that connects people, to keep up with friends, upload photos, share links and ...
3. **YouTube**: is a way to get your videos to the people who matter to you.
4. **Yahoo!**: A major internet portal and service provider offering search results, customizable content, ...
5. **Baidu.com**: The leading Chinese language search engine.
6. **Wikipedia**: A free encyclopedia built collaboratively using wiki software.
7. **Blogspot.com**: Free weblog publishing tool from Google, for sharing text, photos and video.
8. **Windows Live**: Search engine from Microsoft.
9. **Twitter**: Social networking and microblogging service.
10. **Amazon.com**: Seeks to be Earth's most customer-centric company.

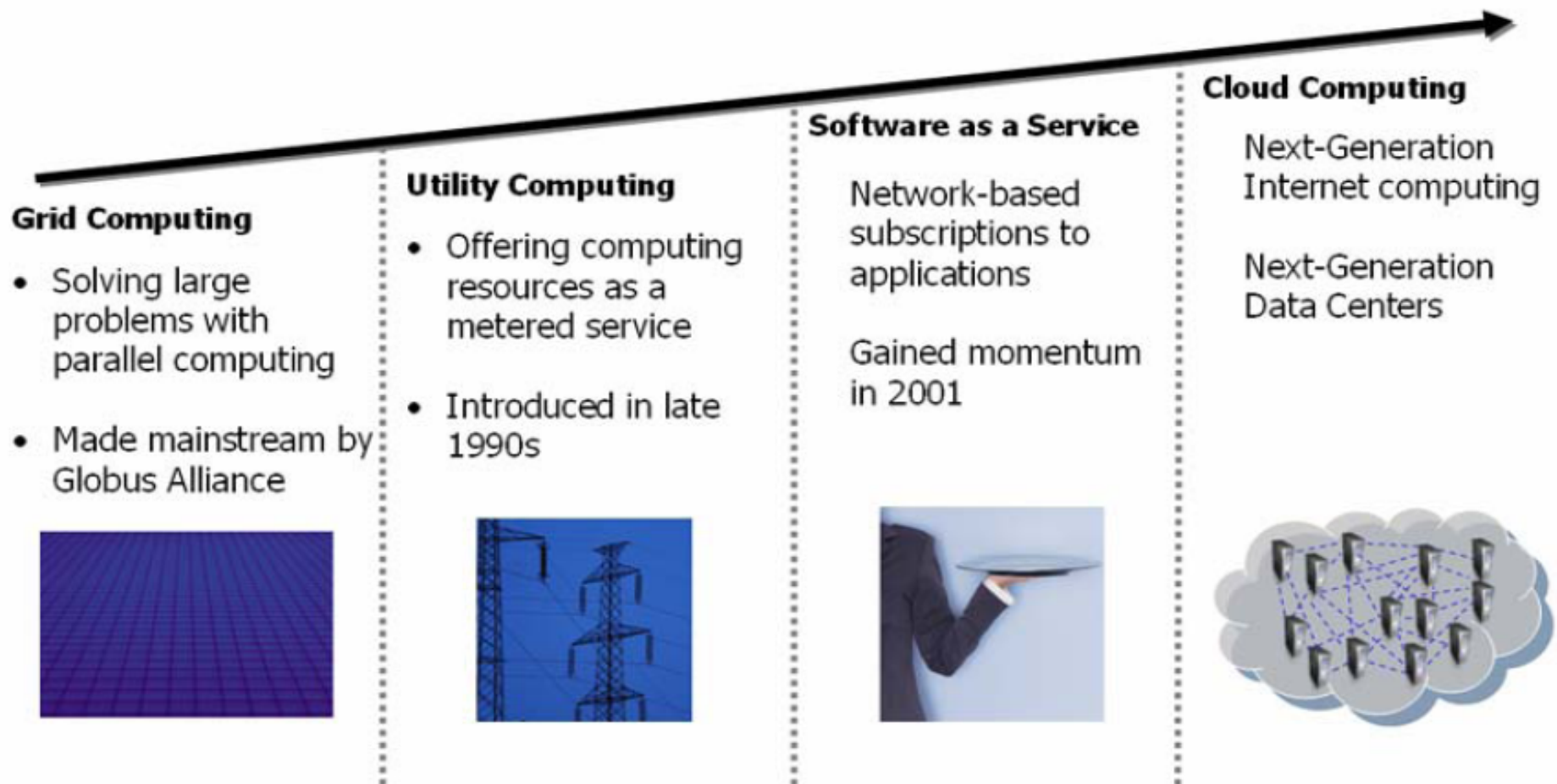
# Berkeley Network of Workstations project





# Computing Models

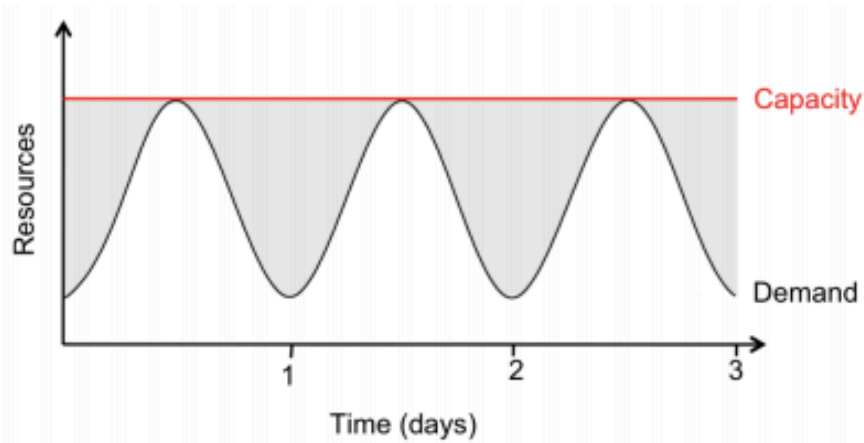
“Why do it yourself if you can pay someone to do it for you?”



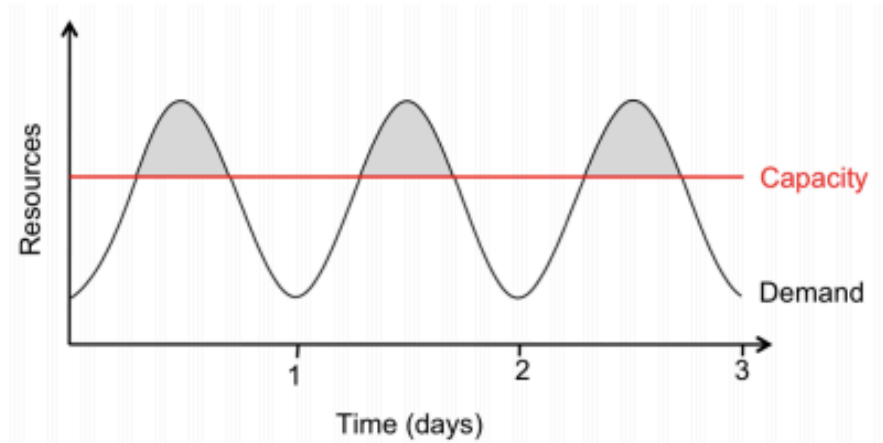
# What is new about cloud computing?

1. The *illusion of infinite computing resources available on demand*, thereby eliminating the need for cloud computing users to plan far ahead for provisioning.
2. The *elimination of an up-front commitment by users*, thereby allowing companies to start small and increase hardware resources only when there is an increase in their needs.
3. The *ability to pay for use of computing resources on a short-term basis as needed* (e.g., processors by the hour and storage by the day) and release them as needed, thereby rewarding conservation by letting machines and storage go when they are no longer useful.

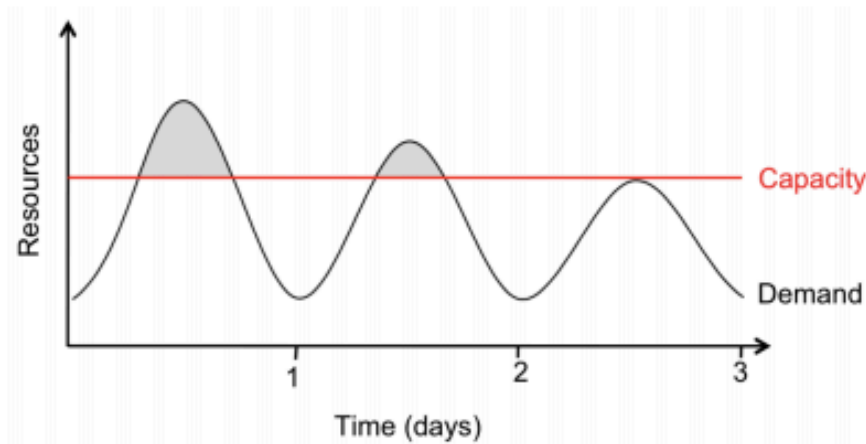
# A bit cloud economics..



(a) Provisioning for peak load



(b) Underprovisioning 1



(c) Underprovisioning 2

(Armbrust et al., 2009)



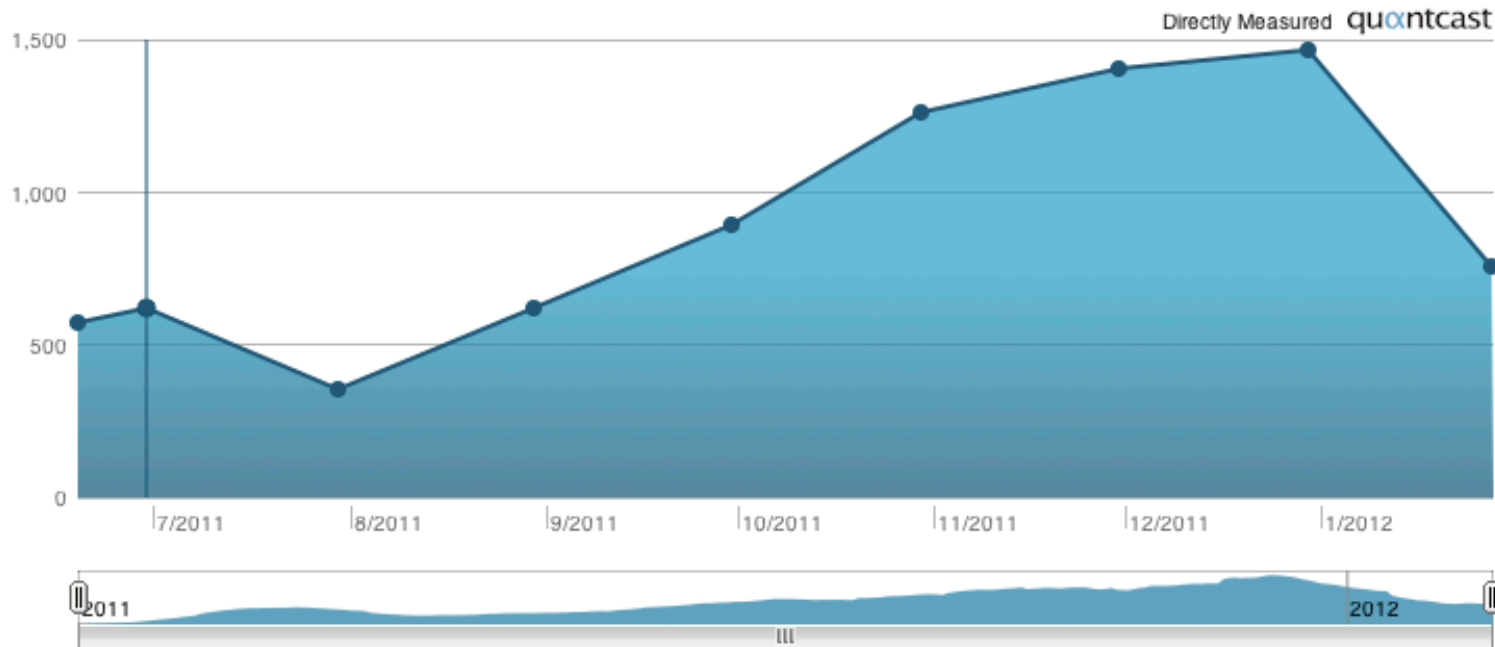
# People using the Amazon website



People (Global) per Day | Week | Month

May 21, 2011 — Jan 28, 2012 ▾

More Options ▾



Updated Jan 29, 2012 • Next: Jan 30, 2012 by 9AM PST

1w | 1m | 3m | 6m | All

Embed [See More](#)

http://www.quantcast.com/p-a7zqvkcXRC66q2

# What is the difference between cloud and grid computing?

Grid computing is where more than one computer coordinates to solve a problem together. Often used for problems involving a lot of number crunching, which can be easily parallelizable.

Cloud computing is where an application doesn't access resources it requires directly, rather it accesses them through something like a service.

The service maps any requests for resources to its physical resources, in order to provide for the application. Usually the service has access to a large amount of physical resources, and can dynamically allocate them as they are needed.

A cloud would usually use a grid. A grid is not necessarily a cloud or part of a cloud.

This excellent answer is taken from:

<http://stackoverflow.com/questions/1067987/what-is-the-difference-between-cloud-computing-and-grid-computing>

# Cloud Computing

## **Cloud computing stack**

# Cloud computing stack

Software as a Service (SaaS)

**consume**

- A way to access applications hosted on the web through your web browser

Platform as a Service (PaaS)

**build**

- A pay-as-you-go model for IT resources accessed over the Internet

Infrastructure as a Service (IaaS)

**host**

- Use of commodity computers, distributed across Internet, to perform parallel processing, distributed storage, indexing and mining of data
- Virtualization



Level of Abstraction

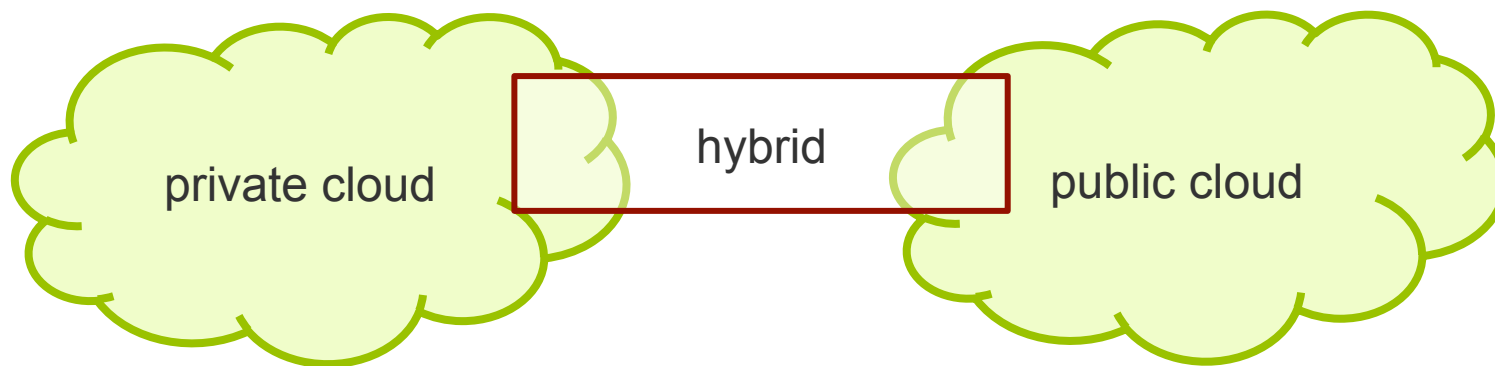


# Deployment models

Private clouds are a variant of generic cloud computing where internal data center resources of an enterprise or organization are not available to the general public.

The public clouds such as Amazon and Google were originally used as private clouds by those companies.

Hybrid clouds combine private and public clouds.



## Cloud computing stack

# Virtualization

Main content of this part is taken from Carlos Garcia Fernandez and Luigi Gallerani, CERN and their lecture on “Virtualization”



# What can be virtualized?

**Network virtualization** splits up the available bandwidth into channels, each of which is independent from the others, and each of which can be assigned to a particular server or device in real time.

**Storage virtualization** is the pooling of physical storage from multiple network storage devices into what appears to be a single storage device. It is commonly used in storage area networks (SANs).

**Server virtualization** is the idea of taking a physical server and partition it, or divide it up, so that it appears as several “virtual servers”, each of which can run their own copy of an OS. The main purpose of this approach is isolation.

# Hypervisor

Also called Virtual Machine Monitor (VMM) or Virtualization Manager.

It is a software program or part of the code in firmware that manages either multiple OS or multiple instances of the same OS on a single computer system.

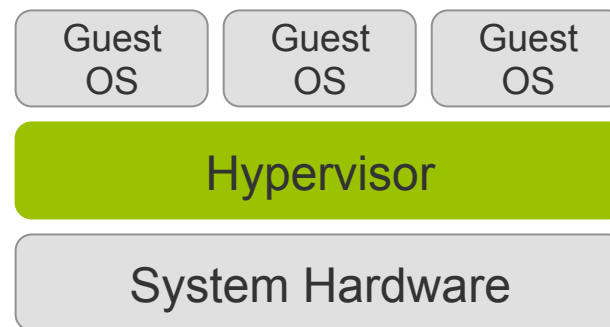
What is controlled by the hypervisor:

- CPU
- Memory
- Other resources required by the operating system

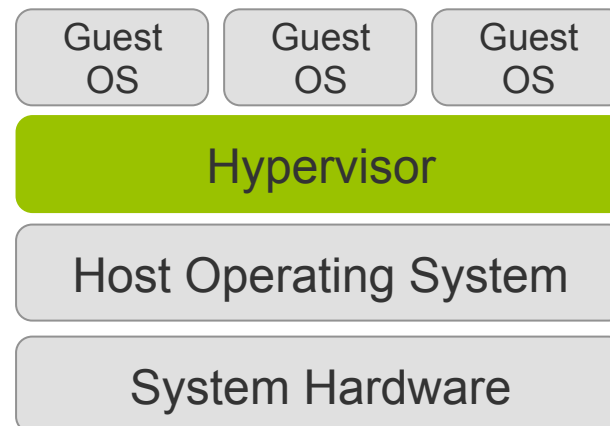
It validates all the guest-issued CPU instructions and manages any executed code that requires additional privileges.

# Types of hypervisor

Type 1 hypervisors are those that run directly on the system hardware and offer a higher level of virtualization efficiency and security.



Type 2 hypervisors are those that run on a host operating system that provides virtualization services, such as I/O device support and memory management.



# Needed concept: protection rings

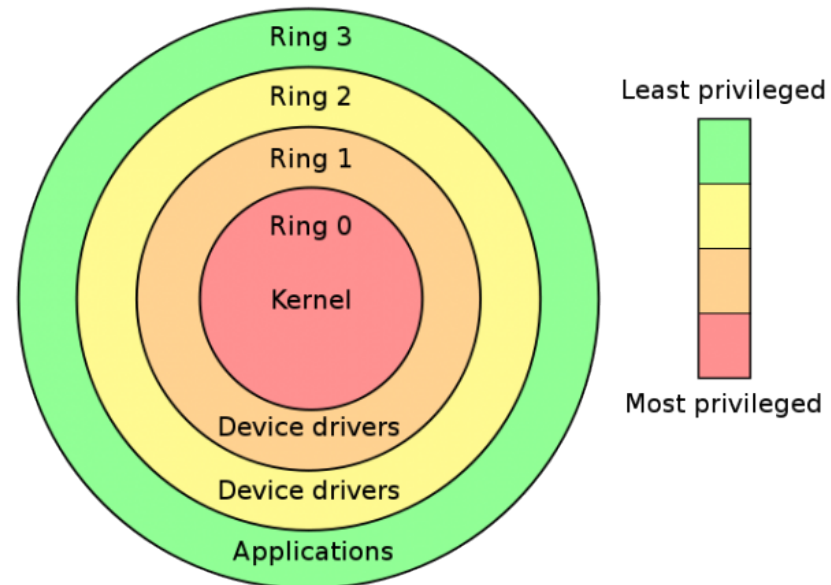
Concept: Mechanism to set the layers of privilege used to protect the data and the functionality

*Ring 0 (kernel):* Can execute any CPU instruction and reference any memory address

*Ring 3 (user):* The access to hardware and memory reference needs to be arbitrated

*Hypervisor type 1* runs directly on the hardware of the host system on ring 0; task of the hypervisor is to handle the resource and memory allocation for the virtual machines

*Guest's kernels* should run in less privileged CPU rings but unfortunately most operating kernels are written to run in ring 0



# Virtualization techniques

- Emulation
- Full virtualization
- Paravirtualization
- Hardware-assisted virtualization
- Application-level virtualization

# Emulation

- VM emulates/simulates complete hardware
- Unmodified guest OS for a different PC can run in this emulation
- It is possible to emulate one architecture in a completely different one

What runs where?

- Ring 0: run the host
- Ring 3: run the guest OS (as applications)

Pros: guest OS doesn't need modifications

Cons: very slow because of the not native execution

Example: Virtual PC for Mac



# Full virtualization

- Host OS emulates a hardware layer for each guest OS

What runs where?

- Ring 0: run the guest OS privileged operations
- Hypervisor provides CPU emulation to handle CPU operations

Pros: Stability, guest OS doesn't need modifications

Cons: System resources, time performance

Examples: VMWare, Parallels

# Paravirtualization

- The hypervisor interfaces the hardware to all OS's (host and guest)
- Guests OS are modified to run on the hypervisor which performs all tasks on behalf of the guest system
- Replacement of privileged operations (ring 0) with calls to the hypervisor

What runs where?

- Ring 0: run an hypervisor
- Ring 1: run the guest OS
- Ring 3: run the applications

Pros: Stability and performance is very good and overhead is very low

Cons: Not easy to implement and to maintain because OS needs to be adapted

Example: XEN, VMWare

# Hardware-assisted virtualization

Hardware provides support to run instructions independently for each OS

What runs where?

- Ring 0: run the guest OS privileged operations
- Latest generation CPUs provide built-in features to run unmodified guest OS without overhead

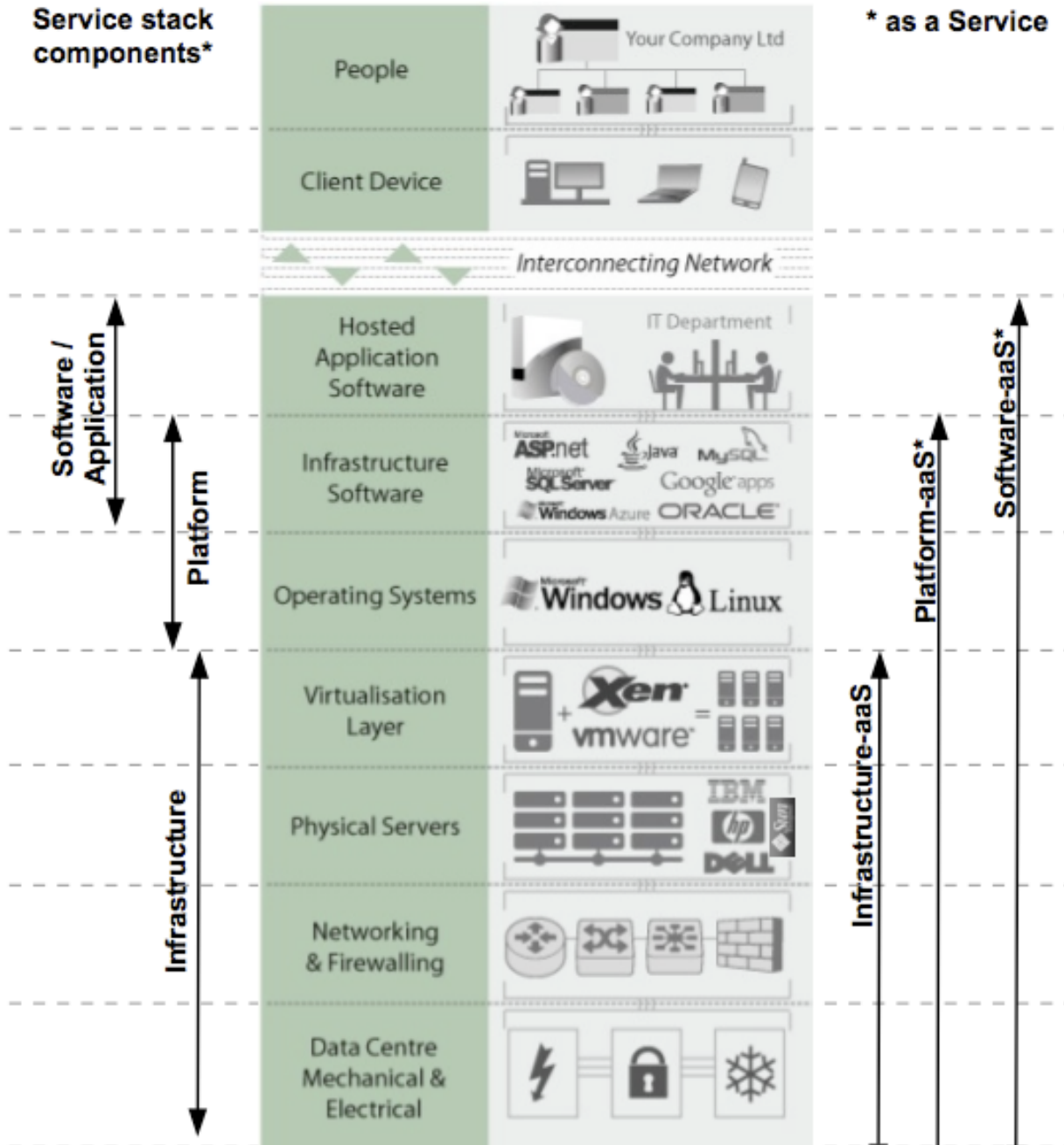
Pros

- No need to patch the guest OS
- Most optimal performance
- Great stability

Cons: Need to be supported by the hardware

# Service Layers Definition

Service stack components\*



\* as a Service

\* Assumed to incorporate subordinate layers.

<http://www.katescomment.com/iaas-paas-saas-definition/>

Cloud computing stack  
**Infrastructure as a service**

# Introducing IaaS

IaaS is a cloud infrastructure service in which hardware is virtualized in the cloud. In this particular model, the service vendor owns the equipment: servers, storage, network infrastructure, and so forth.

The developer creates virtual hardware on which to develop applications and services.

Essentially, an IaaS vendor has created a hardware utility service where the user provisions virtual resources as required.

## Examples

- Amazon S3, Amazon EBS
- Amazon's SimpleDB, Google's BigTable, Apache Cassandra



# IaaS workloads

The fundamental unit of virtualized clients in an IaaS deployment is called a workload.

A workload simulates the ability of a certain type of real or physical server to do an amount of work.

The work done can be measured by the number of Transactions Per Minute (TPM) or a similar metric against a certain type of system.

Clients would reserve a machine equivalent required to run each of their needed workload.

## Example 1: Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud and it is one of the very early pioneers of cloud computing.

*In a nutshell: On Demand “Operating System”*

EC2 provides a complete virtual computer with CPU, memory and disk space that is based on the XEN virtual image platform.

It is available for a variety of operating systems

- Linux (Fedora, Ubuntu, CentOS, etc)
- Open Solaris
- Microsoft Window

URL: <http://aws.amazon.com/ec2/>



## Some EC2 features

Very simple pricing structure

- CPU hours
- Machine size
- Bandwidth in and out of cloud

Extremely fast start up (2-3 minutes from start to finish)

Very large/generous disk space provisioning (160GB minimum for the standard instances)

Flexible API to control everything

Wide range of virtual machine types

## Example 2: Simple Storage Service (S3)

S3 enables to upload, download, and store data across the Internet

Data are stored in buckets

- Buckets are the fundamental container in Amazon S3 for data storage
- No limit on no of objects that can be stored in a bucket
- Can store up to 5 TB of data in one object
- Object stores data and metadata
- Objects stored in a region never leave the region
- You cannot modify or append data to an existing object

URL: <http://aws.amazon.com/s3/>



## S3 (*continued*)

Objects are retrieved via a unique, developer-assigned key

Example key: <http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.wSDL>  
("doc" is the bucket name and "2006-03-01/AmazonS3.wSDL" is the key)

Again, a very simple pricing scheme

Downloading data is possible from any where

You can enable others to download and can charge them (Using Amazon DevPay)

Cloud computing stack  
**Platform as a Service**

# Introducing PaaS

The Platform as a Service model describes a software environment in which a developer can create customized solutions within the context of the development tools that the platform provides.

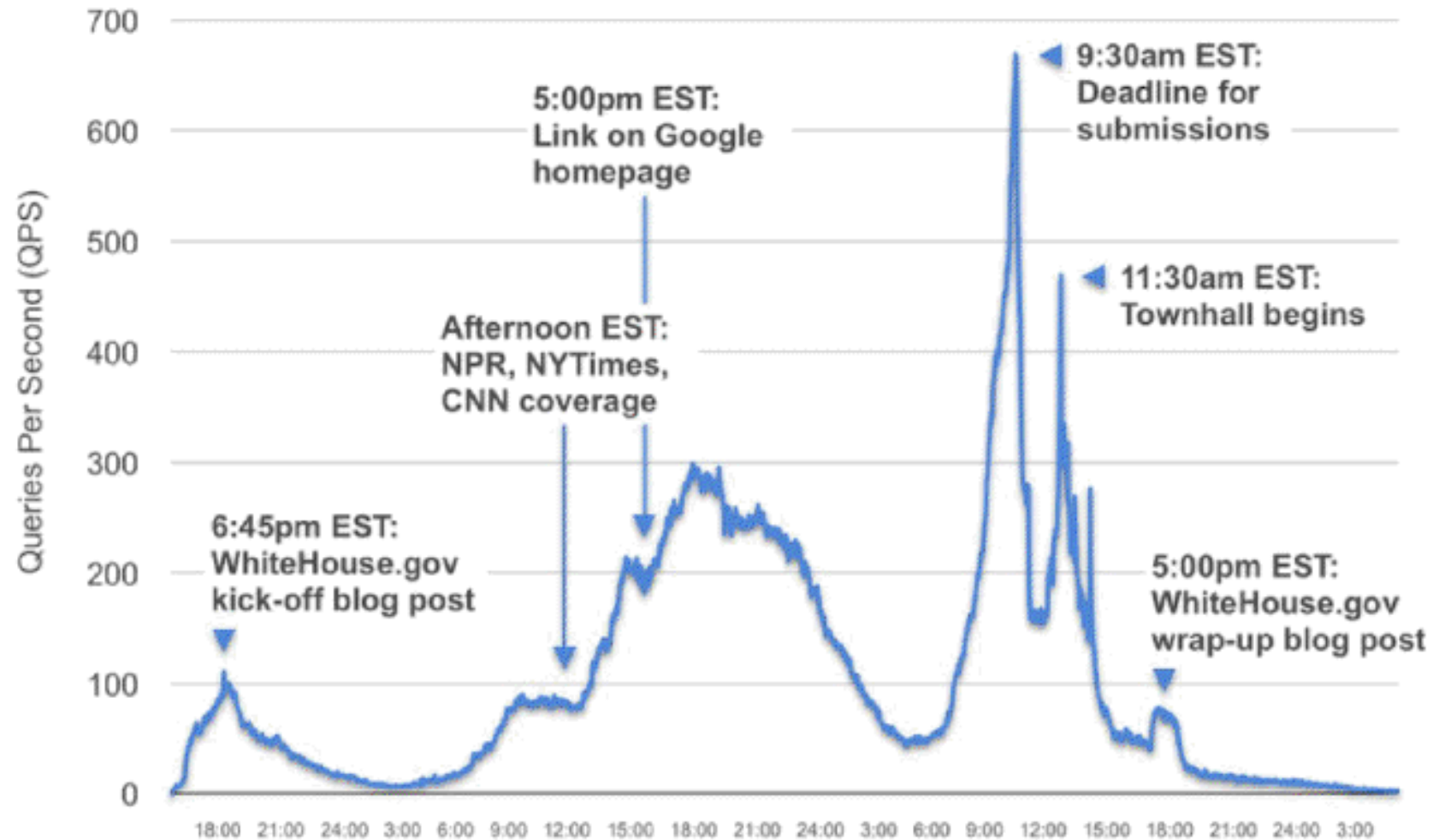
Platforms can be based on specific types of development languages, application frameworks, or other constructs.

A PaaS offering provides the tools and development environment to deploy applications on another vendor's application. Often a PaaS tool is a fully integrated development environment; that is, all the tools and services are part of the PaaS.

Examples:

- Google App Engine - Python (initially, now also Java)
- Force.com (SalesForce)
- Microsoft's Azure - .NET

# whitehouse.gov/openforquestions





# Example PaaS 1: Google App Engine



Google App Engine enables you to build and host web apps on the same systems that power Google applications. App Engine offers fast development and deployment; simple administration, with no need to worry about hardware, patches or backups; and effortless scalability. (<http://code.google.com/appengine/>)

## Some features

- Dynamic web serving, with full support for common web technologies
- Persistent storage with queries, sorting and transactions
- Automatic scaling and load balancing
- APIs for authenticating users and sending email using Google Accounts
- A local development environment that simulates Google App Engine on your computer (Java, Python, also Eclipse plugin)

## Example PaaS 2: Ruby on Rails



Ruby on Rails, often shortened to Rails or RoR, is an open source full-stack web application framework for the Ruby programming language.

Lock-in is not a big concern here because many choices of RoR vendors exist.

More information on RoR <http://rubyonrails.org/>

One arbitrary provider of PaaS is Heroku (<http://www.heroku.com/>)

Cloud computing stack  
**Software as a Service**

# Introducing SaaS

It is the most complete cloud computing service model because the computing hardware and software, as well as the solution itself, are provided by a vendor as a complete service offering.

SaaS provides the complete infrastructure, software, and solution stack as the service offering.

## Examples

- Google Apps - Gmail / Google Docs
- Apple's MobileMe
- Microsoft's Live - Hotmail, Live Spaces
- Salesforce.com



# Cloud Computing **Summary**

## What have we discussed today?

- We talked a lot about clouds, their emergence out of untapped potential. We mentioned that Amazon has provided one of the first services in this area.
- The concept of cloud computing is not new and there is a relation to grid computing that a cloud would usually use a grid but a grid is not necessarily a cloud or part of a cloud.
- We talked about the enabling technology – virtualization and discussed by using the concept of protection rings, various virtualization techniques.
- Finally, we briefly discussed the three levels of abstraction and which services are provided on each of these levels and what are existing applications.

# References

George Coulouris, Jean Dollimore, Tim Kindberg: *Distributed Systems: Concepts and Design*. 5th edition, Addison Wesley, 2011.

Vogels, W. A Head in the Clouds—The Power of Infrastructure as a Service. In First workshop on Cloud Computing and in Applications (CCA '08) (October 2008)

Armbrust, M., et al. Above the clouds: A Berkeley view of cloud computing. Tech. Rep. UCB/EECS-2009-28, EECS Department, U.C. Berkeley, Feb 2009.

<http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.pdf>

Amazon Web (Cloud) Services - documentation <http://aws.amazon.com/documentation/>

Some slides are taken from Satish Sriram: Cloud Provider. University of Tartu. 2011

Mohamed Fawzi: Virtualization and Protection Rings

<http://fawzi.wordpress.com/2009/05/24/virtualization-and-protection-rings-welcome-to-ring-1-part-i/>