Institute of Computer Science
Department of Mathematics and Computer Science

Freie Universität Berlin

Algorithms and Programming IV

Cloud Computing

Summer Term 2023 | 03.07.2023

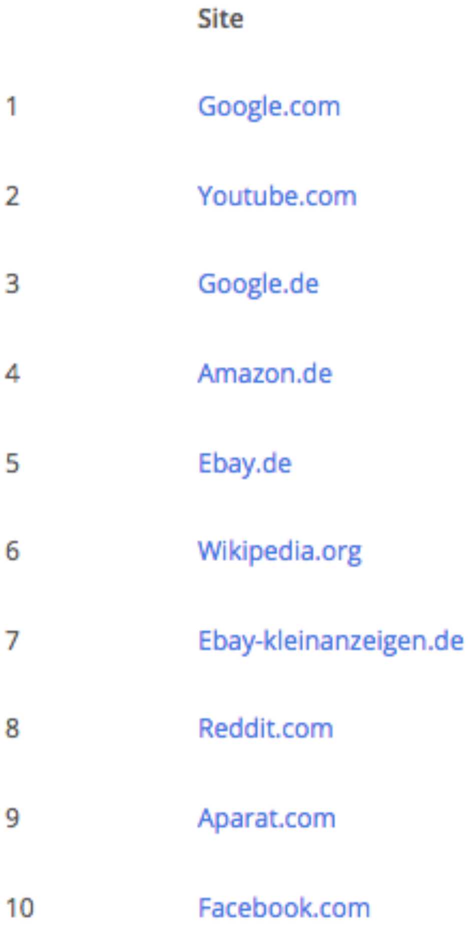
Claudia Müller-Birn, Barry Linnert

Our topics today

- Motivation
- Background Technologies
  - Virtualization (Hypervisor and Protection Rings)
  - Abstraction (SOA and Web Services)
- Deployment and Service Models
  - Public cloud, private cloud, and hybrid cloud
  - Everything-as-a-Service (XaaS) Concept

Motivation

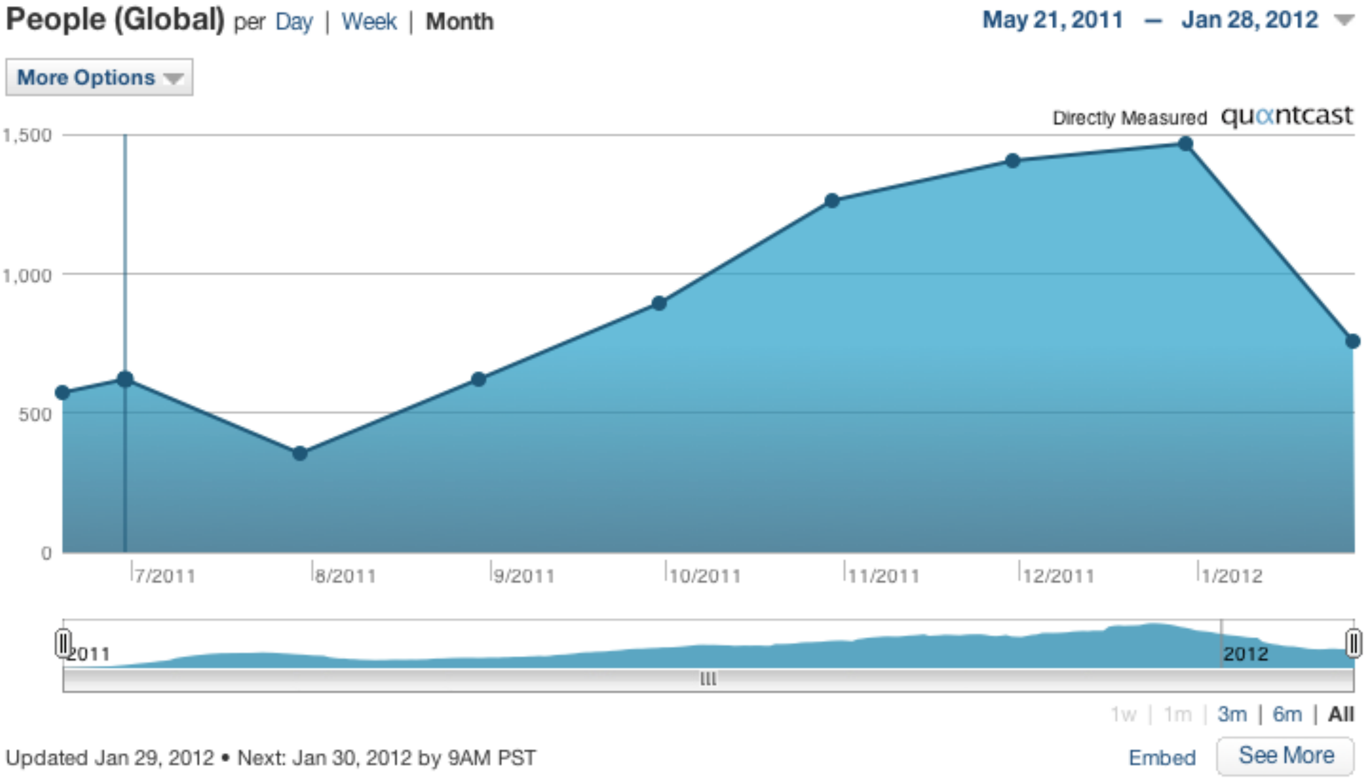
Top Sites in Germany



	Site
1	Google.com
2	Youtube.com
3	Google.de
4	Amazon.de
5	Ebay.de
6	Wikipedia.org
7	Ebay-kleinanzeigen.de
8	Reddit.com
9	Aparat.com
10	Facebook.com

sites/countries/DE (Accessed 6/27/2020)

People using the Amazon website



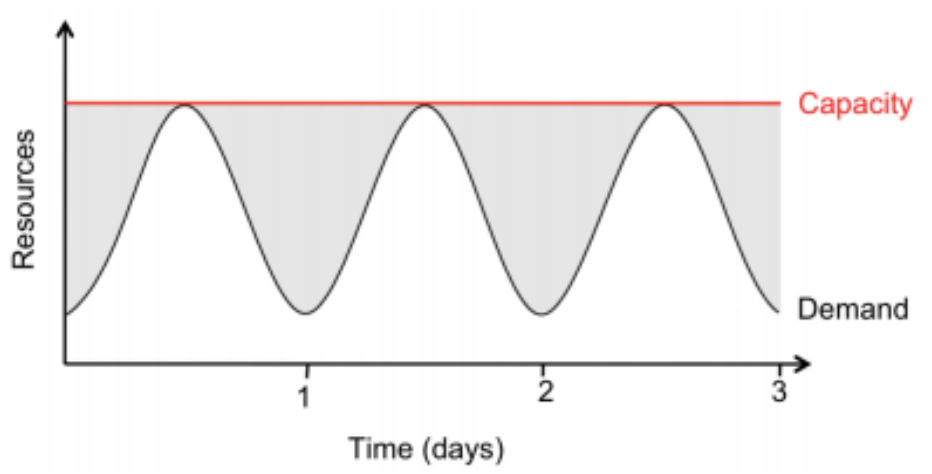
People (Global) per Day | Week | Month

May 21, 2011 — Jan 28, 2012

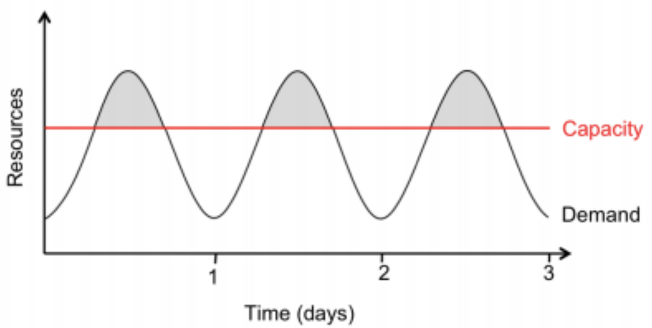
Updated Jan 29, 2012 • Next: Jan 30, 2012 by 9AM PST

http://www.quantcast.com/p-a7zqvkCxRC6q2

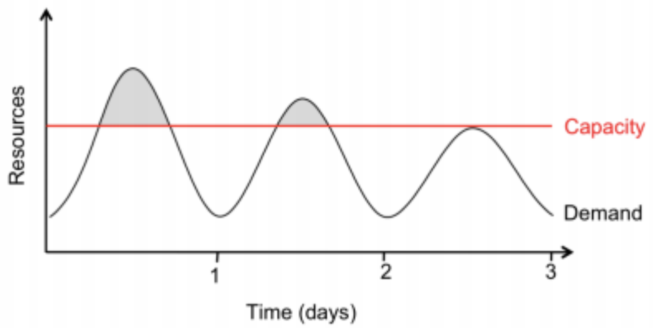
What is the Problem?



(a) Provisioning for peak load



(b) Underprovisioning 1



(c) Underprovisioning 2

(Armbrust et al., 2009)

Defining the Term „Cloud Computing”

- The term captures the vision of computing as a utility.

- It is defined as a set of Internet-based application, storage and computing services sufficient to support most users’ needs, thus enabling them to largely or totally dispense with local data storage and application software.

- The term also promotes a view of everything as a service, from physical or virtual infrastructure through to software.

What is new about Cloud Computing?

1. The *illusion of infinite computing resources available on demand* eliminates the need for cloud computing users to plan far ahead for provisioning.

2. The *elimination of an up-front commitment by users* allows companies to start small and increase hardware resources only when there is an increase in their needs.

3. The *ability to pay for the 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.

(Vogels, 2008)

Defining Cloud Computing

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. This cloud model is composed of five essential **characteristics**, three **service models**, and four **deployment models**.

NIST, US Department of Commerce
http://csrc.nist.gov/publications/nistpubs/800- 145/SP800- 145.pdf

Everything-as-a-Service-Paradigma (XaaS)

Software as a Service (SaaS)

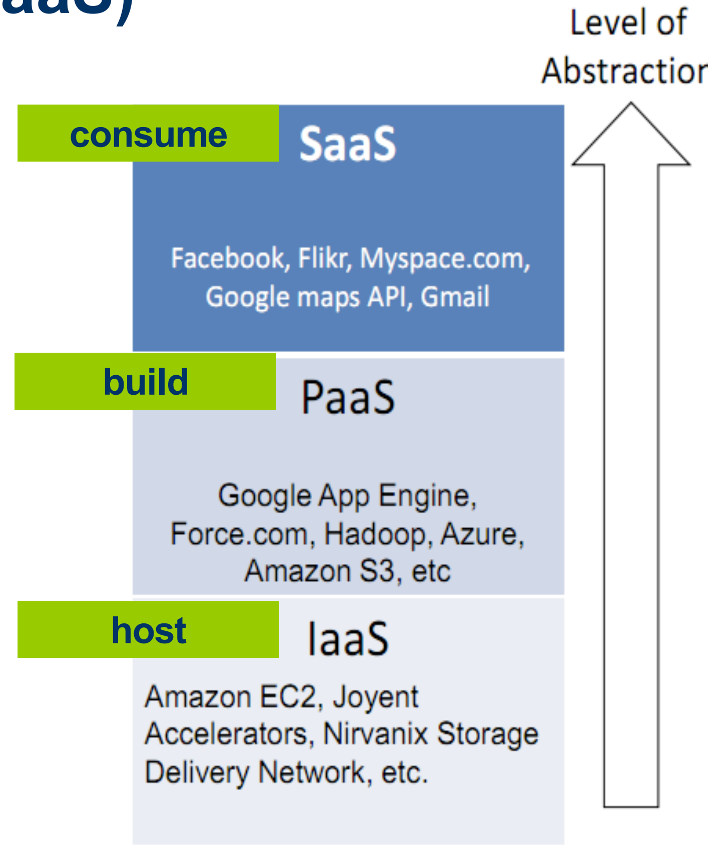
- A way to access applications hosted on the web through your web browser

Platform as a Service (PaaS)

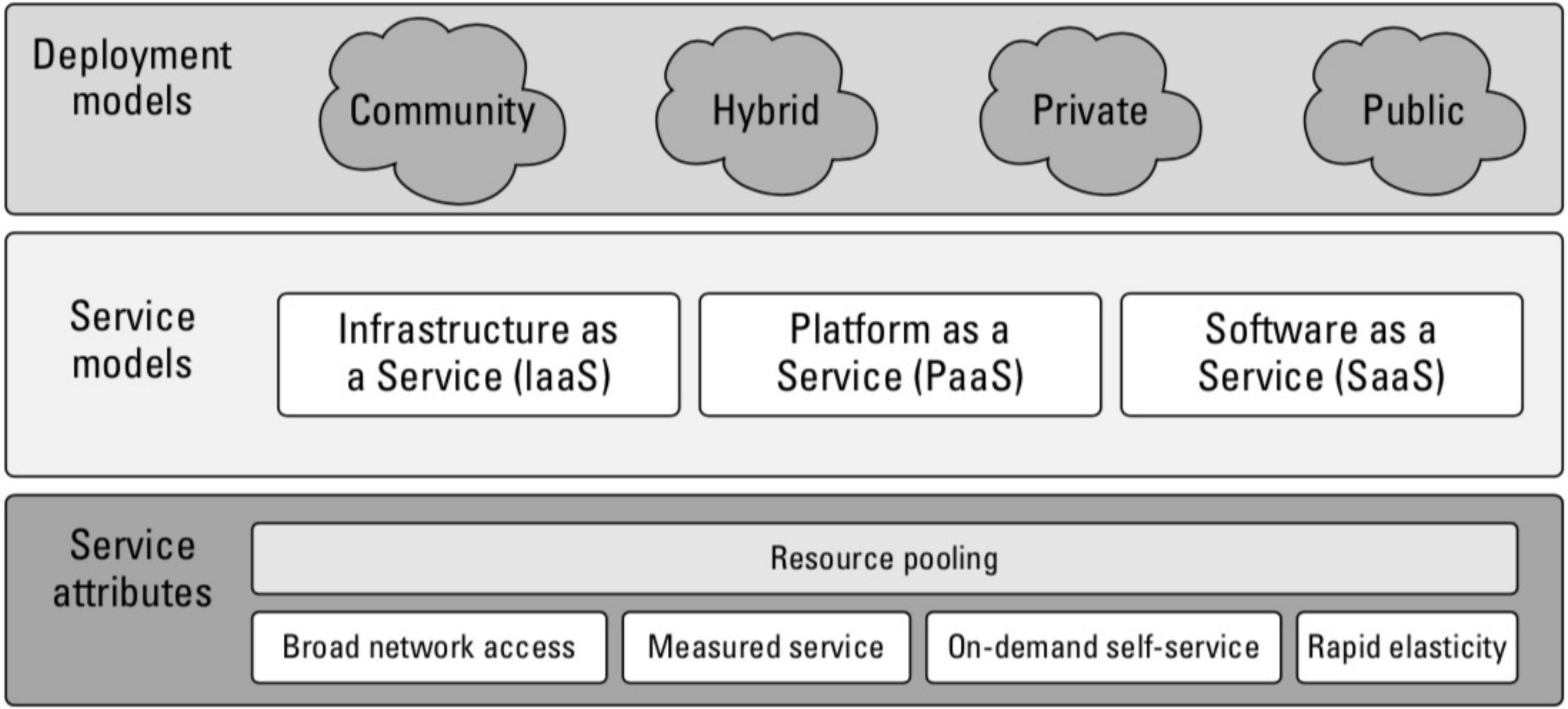
- A pay-as-you-go model for IT resources accessed over the Internet

Infrastructure as a Service (IaaS)

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


Level of
Abstraction
consume
SaaS
Facebook, Flikr, Myspace.com, Google maps API, Gmail
build
PaaS
Google App Engine, Force.com, Hadoop, Azure, Amazon S3, etc
host
IaaS
Amazon EC2, Joyent Accelerators, Nirvanix Storage Delivery Network, etc.

NIST Cloud Computing Definition



NIST, US Department of Commerce http://csrc.nist.gov/publications/nistpubs/800- 145/SP800- 145.pdf

Essential Features

On-demand self-service: Services can be provided on demand to consumers with no humans in the loop

Broad network access: Services are available over the network in real-time through standard mechanisms

Resource pooling: Resources are pooled to enable concurrent service provision while being adjusted to the actual demand

(Rapid) elasticity: Requests for extra resource are self-managed and automatic in relation to demand

Measured (quality of) service: Services leverage a quantitative and qualitative metering capability making usage-based billing

NIST, US Department of Commerce
http://csrc.nist.gov/publications/nistpubs/800- 145/SP800- 145.pdf

BACKGROUND TECHNOLOGIES

Essential Concepts

Virtualization:

- Cloud computing virtualizes systems by pooling and sharing resources. Systems and storage can be provisioned as needed from a centralized infrastructure, costs are assessed on a metered basis, multi-tenancy is enabled, and resources are scalable with agility.

Abstraction:

- Cloud computing abstracts the details of system implementation from users and developers. Applications run on physical systems that aren’t specified, data is stored in locations that are unknown, administration of systems is outsourced to others, and access by users is ubiquitous.

Sosinsky, B. (2010). *Cloud computing bible* (Vol. 762). John Wiley & Sons.

Cloud Computing: Background Technologies

VIRTUALIZATION

Virtualization

- Resources can be detached from their physical nature by means of *virtualization.*
- Virtualized resources for the cloud include servers, data stores, networks, and software.
- The basic idea is to pool physical resources, make them transparent, and manage them as a whole.
- Pools provide transparent access to resources.

Andrea Omicini “Cloud Computing: An Introduction” Dipartimento di Informatica – Scienza e Ingegneria (DISI)

Benefits for Providers

Resource usage: Load balancing towards full capacity of physical servers

Management: Resource pool management can be automated

Consolidation: Reducing the number of physical servers, increasing efficiency and cost effectiveness

Energy consumption: From consolidation

Less space required: From consolidation

Emergency planning: Moving virtual machines from one resource pool to another

Andrea Omicini “Cloud Computing: An Introduction” Dipartimento di Informatica – Scienza e Ingegneria (DISI)

Benefits for Customers

Dynamic behaviour: Resource pools can easily satisfy almost all requests dynamically—even when requests are peaking

Availability: Automated fault tolerance procedures make services highly available

Access: Isolation of virtual machines make it possible to safely delegate management functionality to the customer

Emancipation: Customers can purchase IT capabilities from a self-service portal, without humans in the loop

Andrea Omicini “Cloud Computing: An Introduction” Dipartimento di Informatica – Scienza e Ingegneria (DISI)

Drawbacks

Cost of abstraction: The abstraction layer requires resources; however, modern virtualisation techniques make such a cost more and more negligible

System management: The physical infrastructure grows in complexity; automation of procedures and availability of management tools requires however much less staff in the overall

Andrea Omicini “Cloud Computing: An Introduction” Dipartimento di Informatica – Scienza e Ingegneria (DISI)

Concepts of Virtualization

OS virtualization: Applications run within containers layered upon the host OS.

Platform virtualization: OS & applications run in a virtual environment, based on a VM monitor or hypervisor.

Storage virtualization: Dynamically scalable storage space as a service.

Network virtualization: Virtual IP addresses, Virtual LAN (VLAN).

Application virtualization: Software sales model for centralized management and distribution of applications through the network.

Andrea Omicini “Cloud Computing: An Introduction” Dipartimento di Informatica – Scienza e Ingegneria (DISI)

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.

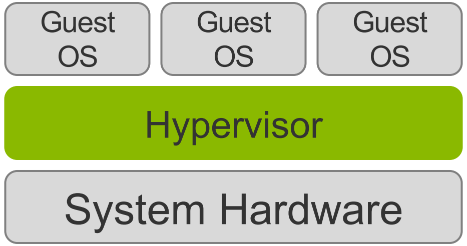
What is controlled by the hypervisor:

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

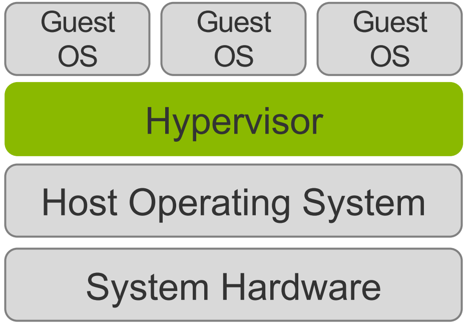
It validates all the guest-issued CPU instructions and manages any executed code that requires additional privileges.

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.

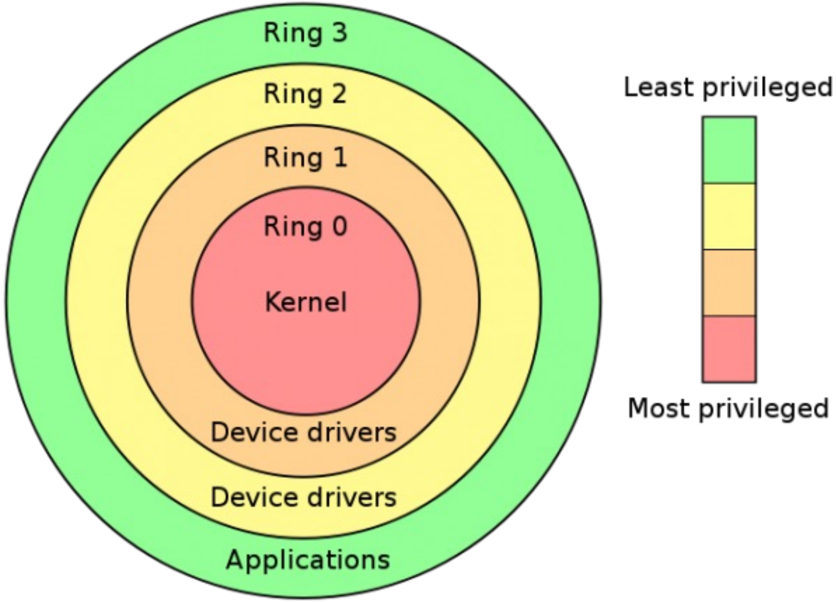


Needed Concept: Protection Rings

- ***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



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.

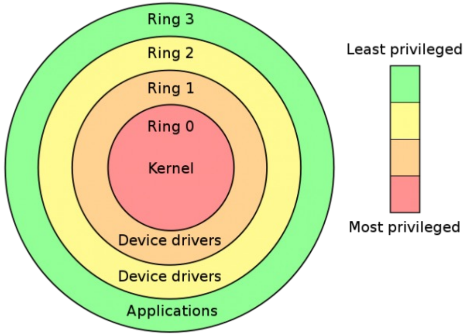
What runs where?

- 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.

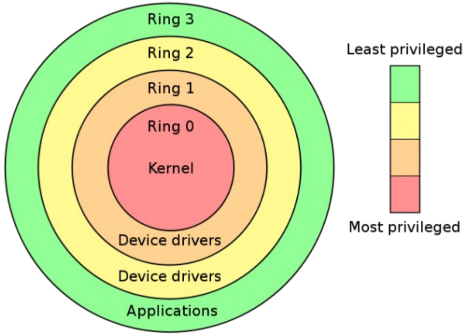
What runs where?

- 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

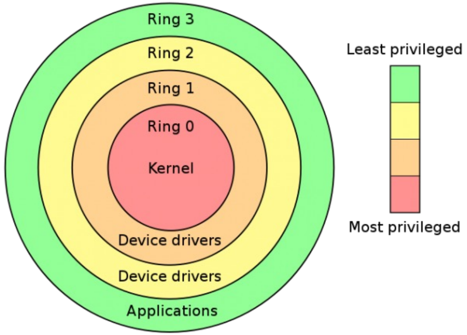


Paravirtualization

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: 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

Example: Linux KVM, Virtual Box

Cloud Computing: Background Technologies

ABSTRACTION

Service-Oriented Architectures

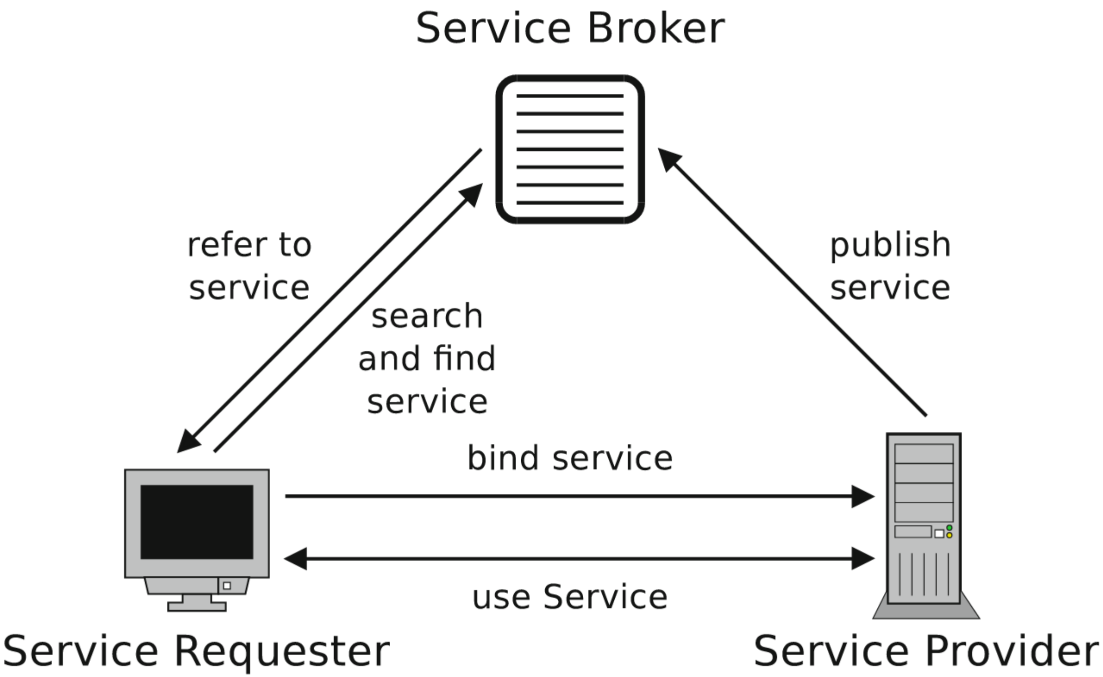
An SOA is a system architecture that represents varied, different, and possibly incompatible methods or applications as re-usable and openly accessible services and thus allows to use and re-use them in a platform- or language-independent manner.

Typical properties of an SOA are:

- It consists of distributed components, i.e. the services.
- Heterogeneous service consumers and service providers are interoperable across platforms.
- Services are loosely coupled and will be bound dynamically at runtime.

Marcel, C. B., Marcel, K., Mimis, J., & Tai, S. (2011). Cloud Computing: Web-based Dynamic IT Services.

SOA Participants and Actions



Marcel, C. B., Marcel, K., Mimis, J., & Tai, S. (2011). Cloud Computing: Web-based Dynamic IT Services.

Defining a Web Service

Generic definition: *Any application accessible to other applications over the Web.*

Definition of the UDDI consortium: *Web services are self-contained, modular business applications that have open, Internet-oriented, standards-based interfaces.* (http://www.uddi.org/pubs/UDDI_Executive_White_Paper.pdf)

Definition of the W3C: *A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.* (http://www.w3.org/TR/ws-arch/)

Characteristics of a Web Service

A web service interface generally consists of a collection of operations that can be used by a client over the Internet. The operations in a web service may be provided by a variety of different resources, for example, programs, objects, or databases.

The key characteristic of (most) web services is that they can process XML-formatted SOAP messages. An alternative is the REST approach.

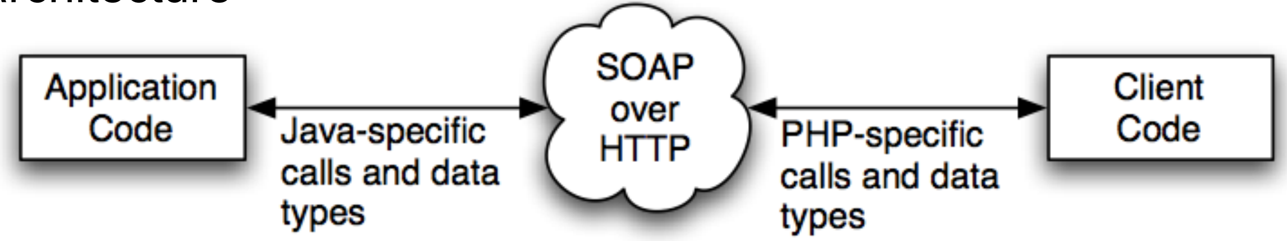
Each web service uses its own service description to deal with the service-specific characteristics of the messages it receives.

Recap: Simple Object Access Protocol (SOAP)

SOAP is designed to enable both client-server and asynchronous interaction over the Internet. It defines a scheme for using XML to represent the contents of request and reply messages as well as a scheme for the communication of documents.

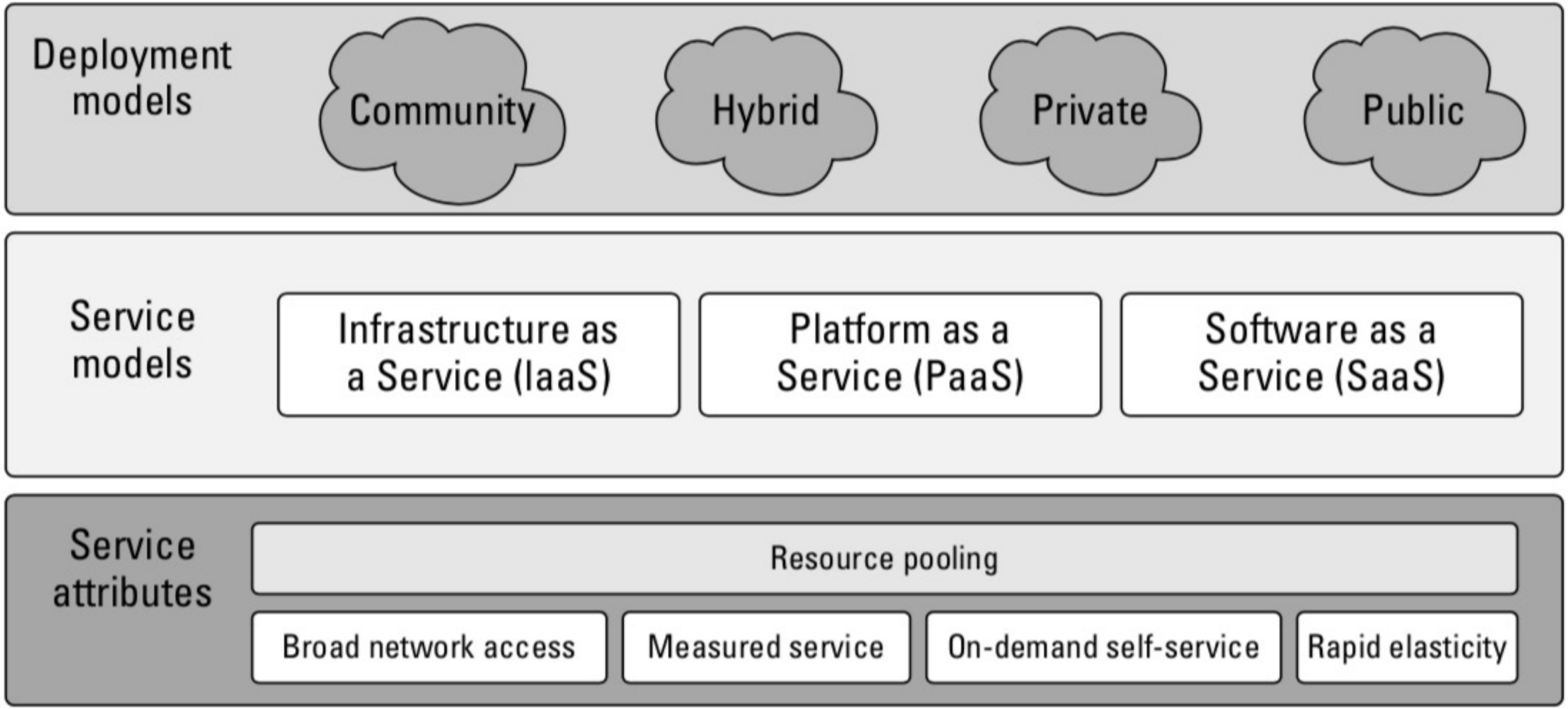
It is used for information exchange and RPC, usually (but not necessarily) over HTTP.

(Very) basic SOAP architecture


Application Code
Java-specific calls and data types
SOAP over HTTP
PHP-specific calls and data types
Client Code

DEPLOYMENT AND SERVICE MODELS

NIST Cloud Computing Definition



NIST, US Department of Commerce http://csrc.nist.gov/publications/nistpubs/800- 145/SP800- 145.pdf

Cloud Types

Deployment Models

This refers to the location and management of the cloud’s infrastructure.

Service Models

This consists of the particular types of services that you can access on a cloud computing platform.

Deployment Models

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

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.

Hybrid clouds combine private and public clouds.

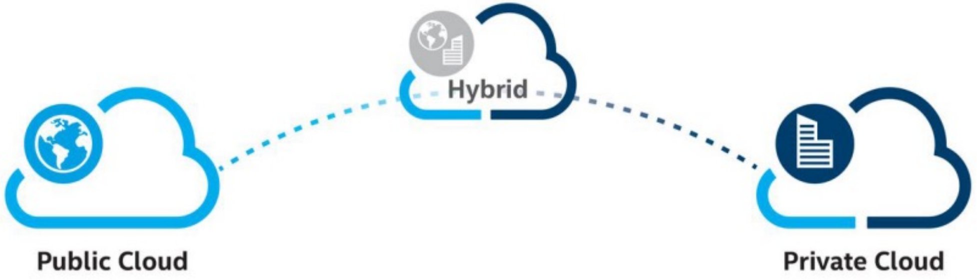
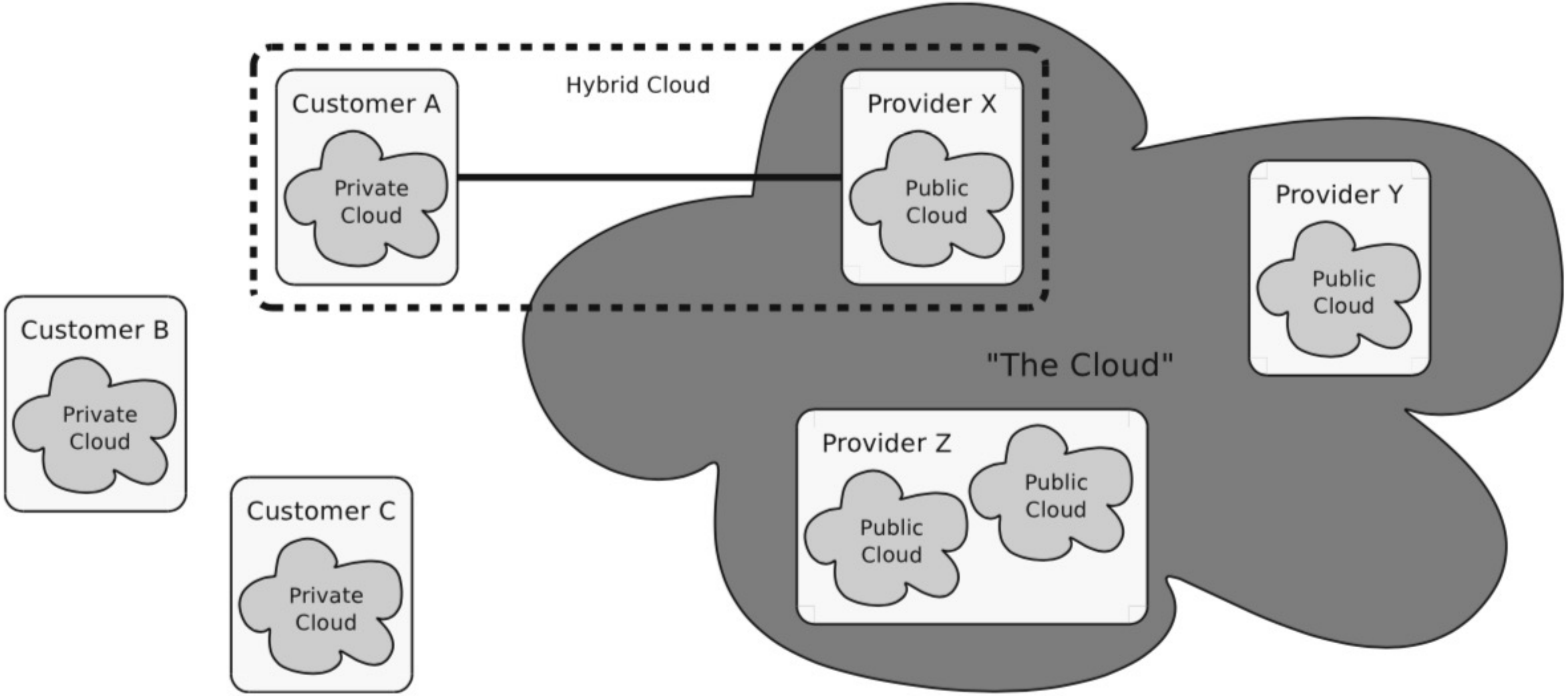


Image from: https://www.intel.de/content/www/de/de/cloud-computing/public-cloud/overview.html

Public Cloud, Private Cloud, and Hybrid Cloud



Marcel, C. B., Marcel, K., Mimis, J., & Tai, S. (2011). Cloud Computing: Web-based Dynamic IT Services.

Cloud Types

Deployment Models

This refers to the location and management of the cloud's infrastructure.

Service Models

This consists of the particular types of services that you can access on a cloud computing platform.

Everything-as-a-Service (XaaS)

Software as a Service (SaaS)

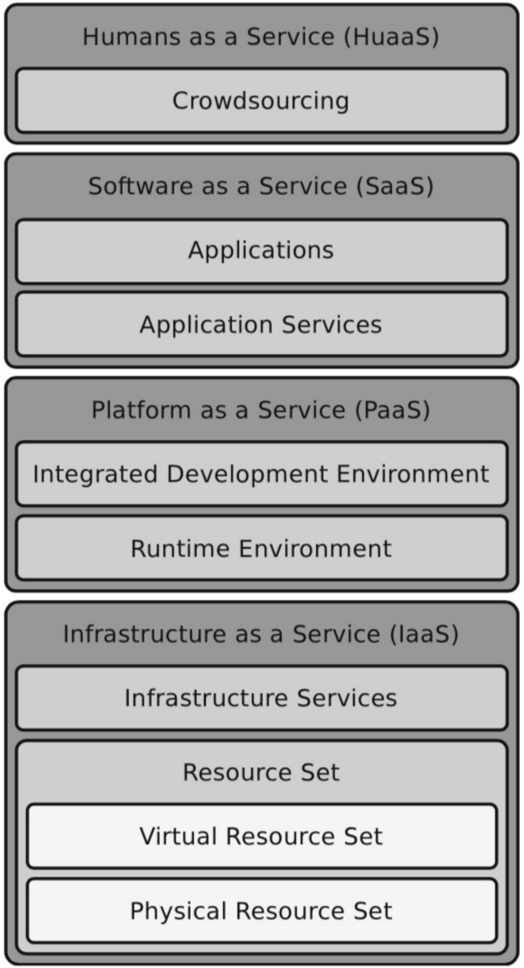
- A way to access applications hosted on the web through your web browser

Platform as a Service (PaaS)

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

Infrastructure as a Service (IaaS)

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


Humans as a Service (HuaaS)
Crowdsourcing
Software as a Service (SaaS)
Applications
Application Services
Platform as a Service (PaaS)
Integrated Development Environment
Runtime Environment
Infrastructure as a Service (IaaS)
Infrastructure Services
Resource Set
Virtual Resource Set
Physical Resource Set

Marcel, C. B., Marcel, K., Mimis, J., & Tai, S. (2011). Cloud Computing: Web-based Dynamic IT Services.

Infrastructure as a Service (IaaS)

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

The IaaS layer provides an abstract view on these physical resources.

- Items in the resource set can be subdivided into
  - physical resource set (PRS), representing and offering proprietary physical hardware, e.g., Emulab
  - virtual resource set (VRS), built on top of virtualisation technologies, and making virtual instances available, e.g., Amazon EC2, OpenNebula

Infrastructure services, too, belong to the IaaS layer, e.g., Hadoop MapReduce, Amazon S3, Dropbox

Example IaaS: 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/

Example IaaS: Simple Storage Service (S3)

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

Characteristics:

- Data are stored in buckets
- Buckets are the fundamental container in Amazon S3 for data storage
- No limit on number 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
- You cannot modify or append data to an existing object

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

amazon web services™

Platform as a Service (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 are Programming environments (PE) that can be based on specific types of development languages, application frameworks, or other constructs. Examples are Django Framework, Ruby on Rails.

Platforms are execution environments (EE) by offering tools and development environment to deploy applications on another vendor’s application. Examples are Google App Engine, Microsoft’s Azure - .NET

Example PaaS: Google App Engine

Google App Engine allows 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
- A local development environment

Example PaaS: Heroku

Heroku is a cloud platform that helps to build, deliver, monitor and scale apps.

These apps run in virtual containers which execute on a reliable runtime environment. Heroku calls these containers "Dynos."

Dynos can run code written in Node, Ruby, PHP, Go, Scala, Python, Java, or Clojure.

Heroku also provides custom buildpacks with which the developer can deploy apps in any other language.

Heroku lets the developer scale the app instantly just by either increasing the number of dynos or by changing the type of dyno the app runs in.

URL: http://www.heroku.com/

One Heroku alternative

- An open-source option for Heroku. It allows developers to run their applications on isolated, pre-configured containers provisioned with the necessary resources and dependencies.
- The core features include:
  - Container Platform
  - GitHub integration
  - Automatic deploys
  - Real-time Monitoring
  - Scalability
- Further information: https://blog.back4app.com/why-use-back4app/

Software as a Service (SaaS)

The SaaS layer directly provides users with software via cloud Applications are no longer installed locally, but are instead stored and managed by the providers, and downloaded when needed

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

Examples

- OpenID
- Google Apps - Gmail / Google Docs
- Apple’s MobileMe

Humans as a Service (HaaS)

HuaaS (Humans as a Service) is the top layer of the cloud computing stack. This shows that the cloud paradigm is not restricted to IT services, but can also be extended to include services provided by human beings as resources.

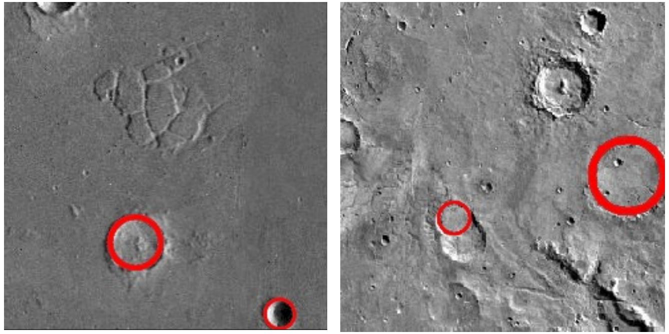
Since humans have certain capabilities that outperform computer systems, their technical integration as resources is a matter of particular interest. Some tasks, such as translation or design services, where creativity is an important asset, are better accomplished by this special resource.

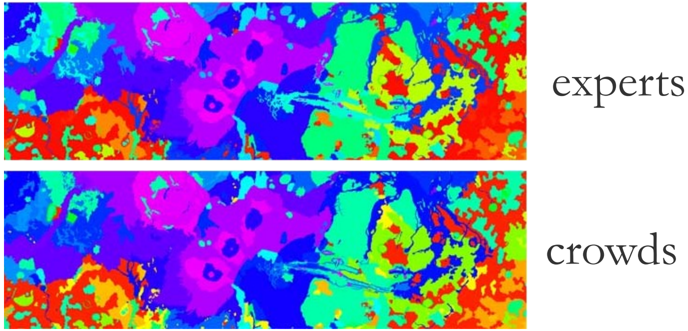
Examples:

- General purpose tasks, e.g. Amazon Mechanical Turk
- Specialized tasks, e.g. Figure Eight (formerly known as CrowdFlower)

Predecessor: Crowdsourcing

- Make tasks available for anyone online to complete
- Quickly access a large user pool, collect data, and compensate users
- Example: NASA Clickworkers
  - 100k+ volunteers identified Mars craters from space photographs (14k hours of free work)
  - Aggregate results “virtually indistinguishable” from expert geologists



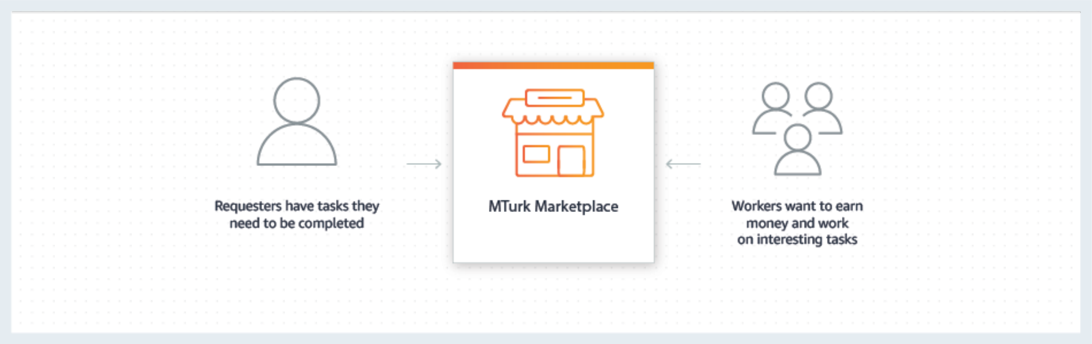


Example HaaS: Amazon Mechanical Turk

Provides a marketplace where *Requesters* can post tasks that *Workers* can perform

Specifically, tasks a computer can’t handle

- Requesters – Post “Human Intelligence Tasks” (AKA HITs) to be completed for a denoted payment consideration upon completion
- Workers – Select HITs to perform and are paid upon submission and Requester’s review & approval of completed work


Requesters have tasks they need to be completed
MTurk Marketplace
Workers want to earn money and work on interesting tasks

amazon mechanical turk

Amazon Mechanical Turk Use Case

Building, managing, and evaluating Machine Learning workflows

MTurk can be a great way to minimize the costs and time required for each stage of ML development.

It is easy to collect and annotate the massive amounts of data required for training machine learning (ML) models with MTurk.

Building an efficient machine learning model also requires continuous iterations and corrections. Another usage of MTurk for ML development is human-in-the-loop (HITL), where human feedback is used to help validate and retrain your model.

amazon mechanical turk

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 service oriented architecture.

Finally, we briefly discussed the four 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. Unviersity 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/