Cloud Computing

IFTS 19/20: modulo "Industria 4.0 – Tecnologie Informatiche di Riferimento"

Giovanni Ciatto
giovanni.ciatto@unibo.it

Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum – Università di Bologna a Cesena

February 14, 2020

Material from the Books

- these slides contain material from
  [Coulouris et al., 2012, Baun et al., 2011, Hill et al., 2013]
- some text and some pictures, too, were extracted from the books
- every problem or mistake contained in these slides, however, should be attributed to the sole responsibility of the authors of these slides

Next in Line. . .

History – Mainframes

- Late 1950s: need for computing power
- Computing mostly used for heavy data analyses required by (and affordable for) big players such as scientific or military organisations:
  - weather forecast
  - aeronautic or ballistic simulations
  - particle accelerators data analysis
  - support for biological experiments
- Before computer networks (1970s), mainframes as the only way to satisfy requirements

History – Mainframes

Figure: An ancient mainframe

History – Commodity computers

- Mainframes required very large investments to be built and maintained
- Late 1960s: microprocessors made it possible to adopt computers in commerce and manufacturing
- Late 1970s: computer networks, such as the Internet, enabled novel means for improving business processes and creating novel ones
- 1980s: computers as commodity having a market segment by their own

History – ARPANET

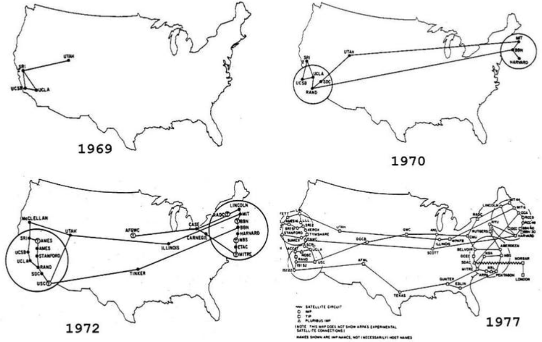


Figure: ARPANET topology evolution

History – Networking

- 1980s: networking stacks such as TCP/IP (i.e. "the Internet" as we know it) and applications (e.g. E-Mail). This enabled for instance:
  - computers from the *same* organisation to talk with each others
  - computers from *different* organisations to talk with each other
    - B2B applications

- When TCP/IP became the *de facto* standard for networking it also became the way for companies to join the global market

History – Network as computer

- 1984, John Gage of Sun Microsystems:

  "The network is the computer"

- Cloud computing was still at least a decade away, but the idea was there

History – Cluster and Grid

- the computational power of mainframes could be approximated by combining several commodity computers interconnected through computer networks
- cluster- and grid-computing paradigms, parallel and distributed computing
- powerful idea from the economical and engineering perspective: the design, development, and deployment processes of clusters and grids were far more scalable than in the mainframe case
  - lower "time to market"
  - investments may be incremental and deferred in time
  - graceful degradation: systems are robust w.r.t. partial failures

History – Cluster-computing

Figure: RACK Cluster

History – The World Wide Web

- Early 1990s: Sir Tim Berners-Lee proposed the idea of sharing knowledge stored on multiple servers to be made available to the world via “client” computers
    - Documents would contain hypertext: text with metadata representing location information to reference other documents
- The World Wide Web (WWW, or the Web) was born, and it dramatically lowered the technological barrier keeping “non-experts” out of the online market
    - it enabled, for instance, stores to expose their products online 24/7
- Quite immediately, Web contents providers soon recognised they desired their Web pages to show dynamic, interactive content driven by user (or costumers)’s inputs
- Soon, virtually any business exposed its products on the Web.

History – The WWW

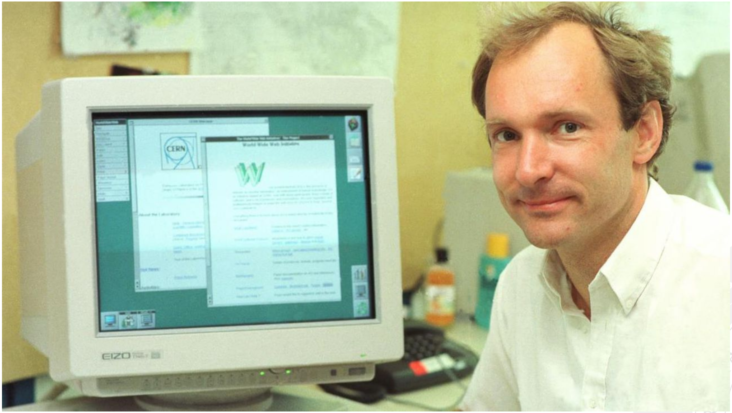


Figure: Sir Tim Berners-Lee

History – Amazon

- Amazon came into being at that time
- From this point in the mid-1990s, they built one of the largest and most powerful network of computers by the early 2000s
- According to their white-paper:

  "By 2005, [we] had spent over a decade and hundreds of millions of dollars building and managing the large-scale, reliable, and efficient IT infrastructure that powered the operation of one of the world's largest online retail platforms."

History – Web business (1)

- Doing business with the Web used to have at least 2 sources of costs:

  CapEx *(Capital Expense)* — i.e. investments needed to create the computational infrastructure supporting the business

  - this comprehends costs for both hardware & software purchase / production

  OpEx *(Operational Expense)* — i.e. costs related to the management and maintenance of the infrastructure

History – Web business (2)

In brief

In the early 2000s, if you had a cool idea – e.g. Facebook – to profit on the Web, you had to spend a considerable amount of money (CapEx + OpEx) just to be *technically* able to even try

- investors did not want their infrastructure to become obsolete as soon as they get more clients
- this implies that higher CapEx were exploted to operate an infrastructure that may be underused most of the time

History – The Cloud

Problem: What did all that equipment do during the summer?

It sat idle.

- Why not "sublease" this equipment to try and recuperate costs?
- In mid-2006, Amazon launched Amazon Web Services (AWS), selling their computing power
- Other companies joined the game (Google, Microsoft, Salesforce, VMware, etc.)

Next in Line...

Focus on. . .

Distributed Computing

Problem: How long will it take before a computation is done?

In most useful cases, a lot

Distributed Computing

- Idea: split algorithms and run them in parallel
  - on several machines
- The focus is still on producing an output . . .
  - but far more quickly
- . . . or efficiently store data for (later) shared access
  - a lot of data

Distributed Computing – Example: CERN

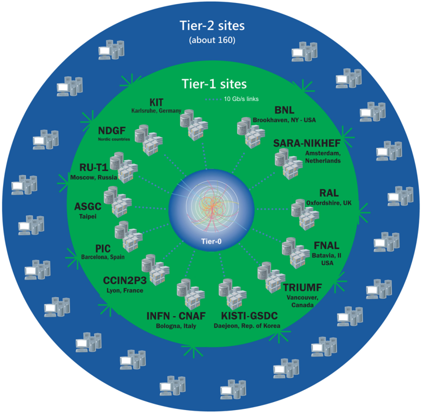


Figure: CERN multi-tier architecture

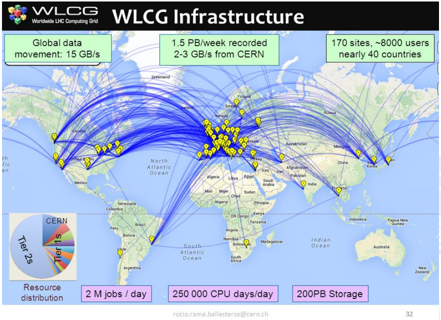


Figure: CERN tier locations

Distributed Systems — pt 1

- In some cases the goal is keeping a system alive for an indefinitely long time
  - e.g. Databases, Social Networks, Robots, etc

Distributed **System**

"A collection of independent computers that appear to users as a single coherent system" — A.S. Tenembaum

or, more commonly

"[a system] in which the failure of a computer you didn't even know existed can render your own computer unusable" — L. Lamport

Distributed Systems — pt 2

- Why would people be interested in Distributed Systems?
    - political, administrative, practical, functional reasons
        - could Facebook even make sense without distribution?
    - they usually come with desirable properties:
        - fault tolerance, graceful degradation
        - robustness, redundancy
        - scalability
        - better response time, remote access
        - ...
- Why should YOU be interested in Distributed Systems?
    - because Cloud systems are :)

Focus on...

WWW – The World Wide Web

Original Definition

- The World Wide Web (WWW, or just Web) is a way of exchanging information on the Internet
- It is the network of resource representations such as images, texts, sounds, or any other data chunk, which can be visited using a browser software

- Clients are willing to access resources, through their representations
- Servers are providing resource representations upon requests performed by clients
- Clients' requests are always directed towards an URL
- Servers' responses usually carry some resource representation, in some standard format such as HTML, JSON, XML, MP3, MP4, etc
- Both requests and responses are structured according to the *Hyper-Text Transfer Protocol* (HTTP), i.e. the WWW core

WWW – Overview

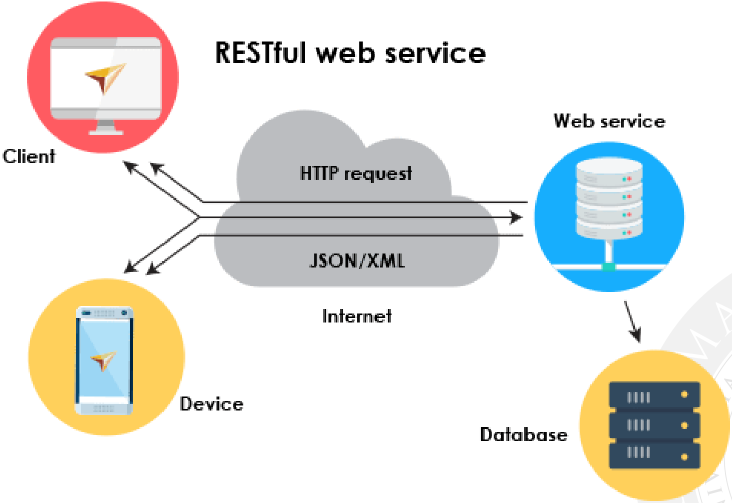


Figure: The interaction among clients and servers on the Web

Focus on. . .

Web Services

- Web Services introduced a strong, paradigmatic principle: separation of concerns
    - the capabilities of the company were expressed in term of services, made accessible through the Web
- The software that was previously distributed through arbitrary technologies, languages, and conventions is now made accessible through Web services
    - using the standard technologies, languages, and conventions which are common for the Web

Web Services — pt. 2

- A service is provided by some provider, and consumed by some requester
- A service has a description which can be dynamically found by consumers, after it has been published by producers on a service registry
    - such description states the service's functioning
- Services interact through messages, which are the basic units of communication carrying requests and responses

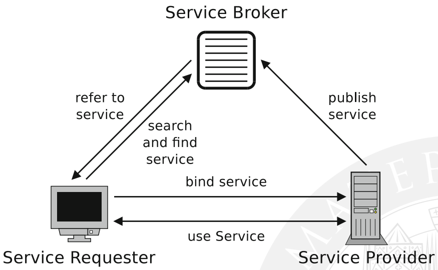


Figure: Main roles in the Web Services arena [Baun et al., 2011]

Web Services — pt. 3

- Services are reusable
  - regardless of whether immediate reuse opportunities exist, services are designed to support potential reuse
- Services share a formal contract
  - for services to interact, they need not share anything but a formal contract that describes each service and defines the terms of information exchange
- Services are composable, i.e., services may compose other services
  - this allows logic to be represented at different levels of granularity and promotes reuse and the creation of abstraction layers

! Services abstract underlying logic and technology
  - the only part of a service that is visible to the outside world is what is exposed via the service contract, the underlying logic is invisible and irrelevant to service requesters

- For all these reasons, Web Services are the perfect means for exposing an organisation's computational capabilities to the public

Web Services – Example

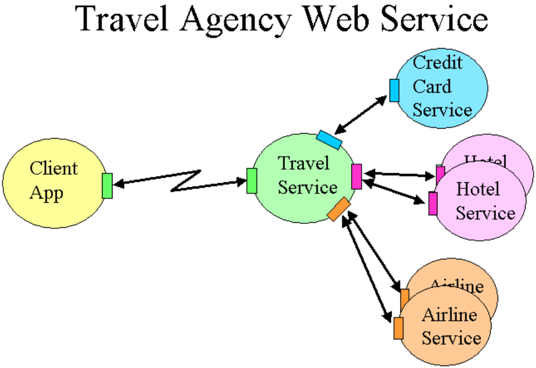


Figure: What happens behind the scenes of a site such as `https://www.volagratis.com`

Focus on. . .

From *X-on-Premise* ...

- In the past, until the advent of Web Services, software used to be distributed on premise
- End users or companies used to buy / download a specific version of a software, to install / deploy it locally, and use it for as long as possible
- This approach had some inherent issues
  - problems used to arise for instance during: set-up, update, etc,
  - every single deployment was different from the other ones
  - making things work was sometimes a mess, and so was fixing them up
- All such issues were a pain both for software producers and for their costumers

. . . to *X-as-a-Service*

- After the advent of Web Services, more and more software have been distributed as a Service
- End users just need to navigate the Web, through their browsers, towards the Web Service URL—an sometimes pay some subscription
- No installation / deployment / updates are required—just bare Internet access and a Web Browser
  - e.g. GMail, Google Docs, Dropbox, Spotify, Netflix, etc
- *X* may be virtually anything that can be conceived in terms of services
  - e.g. emails, documents, data, music, movies, etc
- The Cloud is technically a means for providing computing or storage, as a Service

Focus on. . .

The rise of Virtualisation

- The problem of legacy is of particular interest for software. Software gets old very quickly
    - and so do operative systems, programming languages, software / hardware architectures, etc
- When operative systems or hardware architectures stop being supported, several problems arises for critical products relying on them
    - banks or hospitals software as prominent examples
- Some companies – most notably, IBM – soon realised they could:
    - stop spending money in keeping oldest software compatible with newer operating systems / hardware architectures
    - invest in finding a way to emulate oldest operating systems / hardware architectures thus keeping oldest software unchanged

Traditional VS Virtual

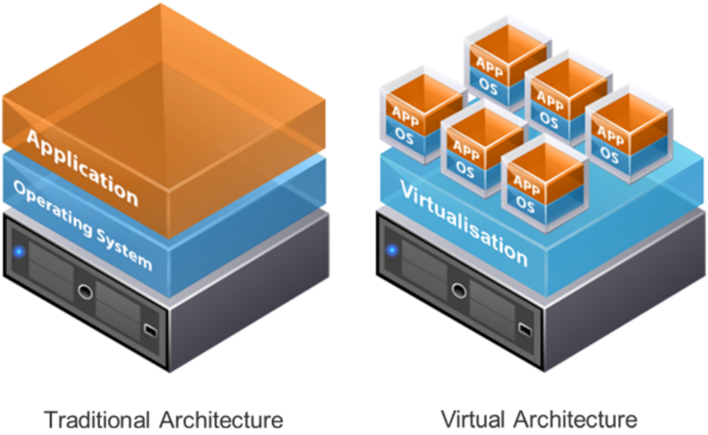


Figure: Traditional VS Virtualised Architecture

VMware: an example of Virtualisation Technology

Figure: Running Windows 10 within MacOS through VMware

Virtualised resources in the Cloud

- Resources can be detached from their physical nature through virtualisation
- Virtualised resources may include operative systems, data hard disks, networks, and software runtimes
- Economy of scale: the basic idea is to pool physical resources, factorising operational costs
- Pools provide transparent access to resources, w.r.t. users' perception

Spoiler alert!

Cloud computing is essentially a way for some costumers to access virtualised resources exposed by some providers

Benefits for providers

Resource Usage — load balancing towards full capacity of physical servers

Management Costs — resource pool management can be automated

Consolidation — reducing the number of physical machines, increasing efficiency and cost effectiveness

Energy consumption — because of consolidation

Less space required — because of consolidation

Emergency Planning — moving virtual machines from one resource pool to another

Benefits for customers

Elasticity — 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

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

Take away

Virtualisation comes with drawbacks, but they are more than compensated by the advantages of the economy of scale, and the standardised management processes

Concepts

OS virtualisation — applications run within *containers* layered upon the host OS

Platform virtualisation — OS & applications run in a virtual environment, based on a VM monitor or *hypervisor*

Storage virtualisation — dynamically scalable storage space as a service

Network virtualisation — virtual IP addresses, virtual LAN (VLAN)

Application virtualisation — software model for distribution of applications through the network

Next in Line. . .

Resources as commodities [Coulouris et al., 2012]

- Distributed resources as a commodity or utility
  - drawing the analogy between distributed resources and other utilities such as water or electricity
- With this model, resources are provided by appropriate service suppliers and effectively rented rather than owned by the end user
- This model applies to both physical resources and more logical services

Physical resources [Coulouris et al., 2012]

- Physical resources such as storage and processing can be made available to networked computers
  - ! removing the need to own such resources on their own

- In the simplest case, a user may opt for a remote storage facility for file storage requirements

Logical resources [Coulouris et al., 2012]

- Software services can also be made available across the global Internet using this approach

- Many companies offer a comprehensive range of services for effective rental, including services such as email and distributed calendars—e.g., Google Apps

- This development is enabled by agreed standards for software services, for example as provided by Web Services

Cloud computing [Coulouris et al., 2012]

- The term cloud computing is used to capture this vision of *computing as a utility*
- A Cloud 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, often paid for on a per-usage basis rather than purchased
- Note that cloud computing reduces requirements on users' devices, allowing very simple desktop or portable devices to access a potentially wide range of resources and services

Next in Line. . .

Focus on...

Baun's Definition

Definition [Baun et al., 2011]

By using virtualized computing and storage resources and modern Web technologies, cloud computing provides scalable, network-centric, abstracted IT infrastructures, platforms, and applications as on-demand services. These services are billed on a usage basis.

Remarks

- Even though cloud services usually rely on distributed systems, the definition do not specify this
  - in principle, it might be provided also by a single, high-performance server, such as a mainframe
  - also, cloud management is typically determined in a central (and proprietary) manner by a single provider
  - unlike grid computing, where distributed nodes are usually autonomous
- Also, a fundamental issue for cloud computing is its economic impact
  - being strictly service-oriented and Web-standard-based, cloud computing is an amenable choice fur a huge number of application scenarios
  - many IT approaches and solutions might exploit cloud services and get economical benefits

Focus on. . .

NIST Definition

Definiton [Mell and Grance, 2011]

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.

Five essential features according to NIST

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 to multiple users (multi-tenant model), while being adjusted to the actual demand of each user

(Rapid) Elasticity — requests for extra resource are self-managed and automatic in relation to demand; from the consumer's perspective, the supply of resources has no limit

Measured (quality of) service — services leverage a quantitative and qualitative metering capability making usage-based billing and validation of the service quality available

Three service models according to NIST

Infrastructure-as-a-Service (IaaS) — clients access (virtualised) machines, along with their (virtualised) hardware, network facilities, etc—infrastructure, in one word

- system administrators are common clients here

Platform-as-a-Service (PaaS) — clients access ready-to-program software runtimes, supporting most common programming languages, frameworks, and patterns—platforms, in one word

- software developers are common clients here

Software-as-a-Service (SaaS) — clients access ready-to-use applications, e.g. GMail

- end users are common clients here

Four deployment models according to NIST

Public/external cloud — the Cloud provider and its clients belong in different organisations

Private/internal cloud — the Cloud provider and its clients belong to the same organisation

Community cloud — a public Cloud where clients are from the same community or consortium

Hybrid cloud — any combination of the previous ones

Next in Line...

Perspectives on Cloud

Technical vs. organisational viewpoints

- A technical viewpoint focuses on the functional features of cloud architectures
- However, this is not the only one available and meaningful perspective: the way in which the cloud is deployed is also relevant
- In particular, the different roles played by providers and customers is essential for the organisational viewpoint

Focus on. . .

Public Cloud — pt. 1

- A public cloud (or, *external* cloud) is offered by a provider to users outside its organisation boundaries
- The cloud service is made publicly available through an automated service (typically, a Web portal) with no humans in the loop
- Obligations are pre-defined in terms of the service, and automatically negotiated
- Billing based on actual usage

Public Cloud — pt. 2

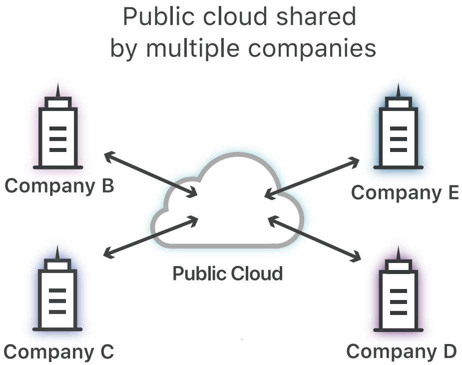


Figure: Organisations act as cloud consumers when accessing cloud services and IT resources made available by different cloud providers

Private Cloud — pt. 1

- Providers of a private cloud (or, *internal* cloud) belong to the same organisation as the users
- The cloud service is made available through an automated service (typically, a Web portal) with no humans in the loop
- The main aim of private cloud is security, by keeping control of organisation data

Private Cloud — pt. 2

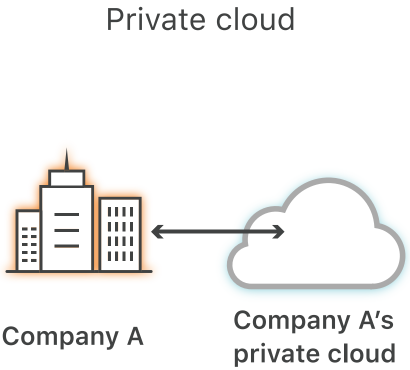


Figure: A cloud service consumer in the organization's on-premise environment accesses a cloud service hosted on the same organization's private cloud via a virtual private network

Hybrid Cloud — pt. 1

- In order to exploit the same tech & tools of public clouds
- And to scale automatically when private cloud is not enough
- Hybrid Clouds are built by bringing together resources from both private and public clouds
- There, normal operation is handled by the private cloud, while load peaks are typically transferred to the public cloud, obviously excluding data and resources that are critical for the organisation

Hybrid Cloud — pt. 2

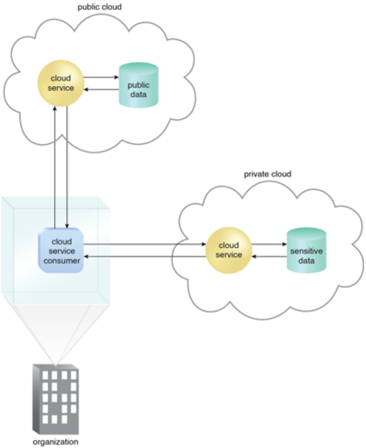
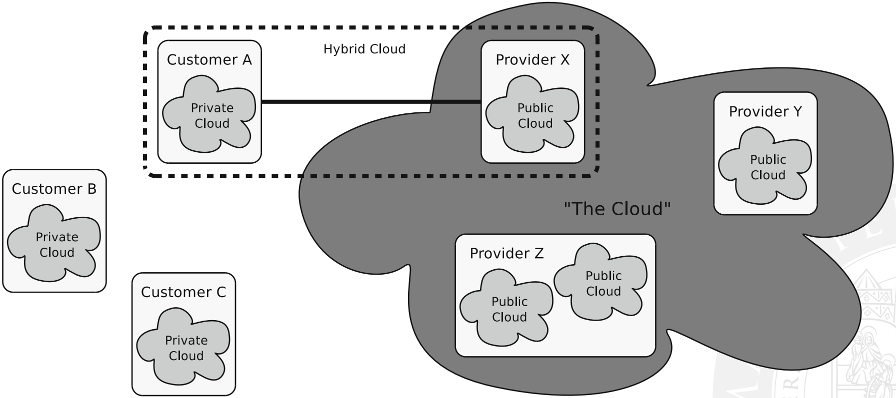


Figure: An organization using a hybrid cloud architecture that utilizes both a private and public cloud

Public / Private / Hybrid Clouds



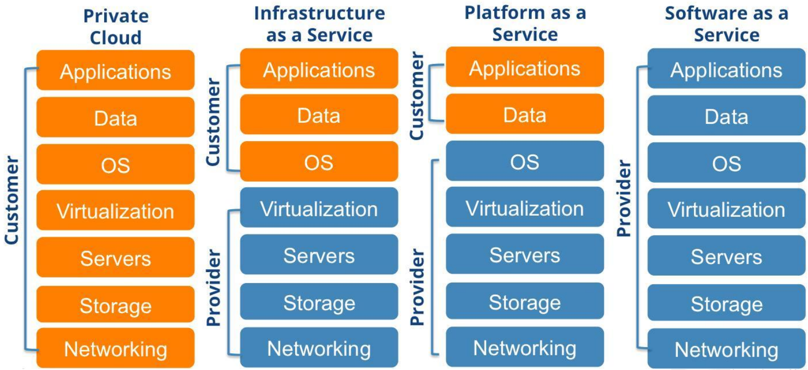
[Baun et al., 2011]

Focus on. . .

Everything as a Service (XaaS)

- Through the cloud, so many diverse infrastructures, services, and applications are made available as Web services
- As a result, cloud services present with diverse functions and goals
- The cloud offering can be then depicted as an architecture layered according to different layers of abstractions
- The three main layers follow the Everything-as-a-Service paradigm (XaaS): IaaS, PaaS, and SaaS

Cloud Service Models – Overview



Infrastructure as a Service (IaaS)

- The IaaS layer provides an abstract view on the physical resources—computers, mass storage systems, networks, etc.
- A user interface is provided to manage resources, allowing users allocate and use them
- Resources may be either:
  - physical resources, representing and offering proprietary physical hardware—e.g., Emulab, iLO
  - virtual resources, built on top of virtualisation technologies, and making virtual instances available—e.g., Amazon EC2, Nimbus, OpenNebula
- The cloud provider is only responsible for the correct functioning of the resource layer
  - OS configuration is up the costumers

Platform as a Service (PaaS)

- The *PaaS layer* provides developers with tools for the cloud
- Programming environments (PE) and execution environments (EE) for programming the cloud

PE e.g., Django Framework, Sun Caroline

EE e.g., Google App Engine (Python, Java), Azure by Microsoft (multilanguage)

- The cloud provider is only responsible for the correct functioning of PE and EE
  - data and application development / management is up to the costumers

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
- Application services, such as Google Maps
- Full-fledged applications, such as Google Docs, Microsoft Windows Live
- The cloud provider is responsible for every aspect of applications management

Value Visibility Pyramid

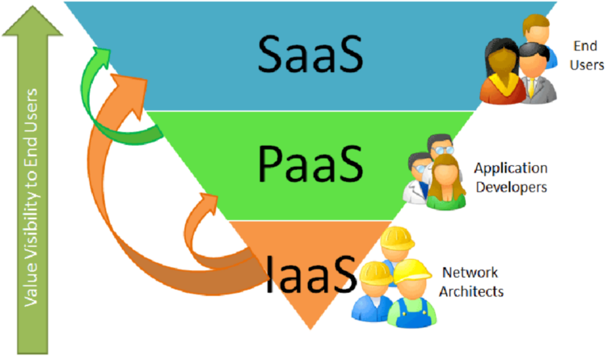


Figure: Service models come with different intended targets. The perceived value for the end user is also different

Hierarchical nature of Service Models

- Notice that higher layers are usually built on top of the lower ones
- PaaS providers may leverage on some third-party IaaS provider, in fact in PaaS:
  - costumers are usually app developers
  - providers are usually big cloud providers

  e.g. Heroku is built on top of Amazon
- SaaS providers may leverage on some third-party PaaS provider, in fact in SaaS:
  - costumers are usually common people
  - providers are usually app developers

Next in Line...

Service Level Agreements (SLAs) — pt. 1

- A service level agreement is an agreement between a service provider and a service consumer related to the service level (quality of service, QoS)
- The SLA implies a mutual agreement with respect to security, priorities, responsibilities, guarantees, and billing modalities
- Furthermore, SLA specifies metrics such as availability, throughput, response times, and others
- Two phases for service level management
  - agreement on the quality of service
  - ! service monitoring at runtime

Service Level Agreements (SLAs) — pt. 2

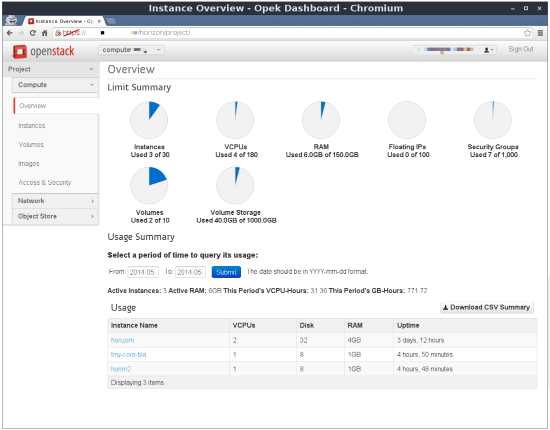
Figure: Features for good SLA

Lifecycle & Automation

- Cloud services have a lifecycle:
  - made available by the provider
  - selected and used by the customer
  - closed and detached at the end
  - finally accounted for
- Cloud costumers may automate most phases by means of template languages and orchestrators
- Cloud systems usually let costumers submit a formal, declarative description (i.e. a template) of the infrastructure / application to be instantiated
- Such description is interpreted by the Cloud system's orchestrator in order for the system to be automatically instantiated

Management Services & Tools

- Many tools are made available by providers, either via CLI or via Web for *monitoring*, *control*, and *development*, like, for instance, Web-based dashboards:



Security Management

- Exploiting Cloud services implies delegating some security-related aspects to the Cloud provider
- This is usually good, since smaller costumers cannot usually afford highest security standard—whereas Cloud providers commonly can and should
- But it also means those security-related aspects are out of the control of the costumer
- Keep in mind that Cloud providers have (and must have) access to all the costumers data stepping through the Cloud services
- When choosing a Cloud provider, it is wise to figure out how it defends costumers against both external and internal treats
  - and what happens in case of violation—through SLA

Risk Management

- Again, cloud does not add much in the risk management perspective

- Risks are more or less the same of any outsourcing process:
    - data transfer
    - service disruption
    - provider bankruptcy

- The vendor lock-in problem is arguably the most common, and subtle one

Legal Compliance

- Laws are applicable to cloud computing as to renting (w.r.t. paid services) or lending (w.r.t. unpaid services)
- Geographical location of providers affects applicable law
- e.g., according to article 4 of the European Data Protection Directive, european people' data must by stored within EU borders—which is why cloud services are typically provided region-based
- Particular attention should be devoted to legal aspects related to personal data processing—e.g., GDPR
- Special legislation takes care of sensitive data in scenarios such as healthcare and finance

Next in Line. . .

Recent Trends

- Containers, Micro-services, and CaaS
- Storage-aaS
- Machine-Learning-aaS
- ...

Some Links

- https://console.cloud.google.com/home/dashboard
- https://cloud.google.com/pricing/list
- https://aws.amazon.com/it/ec2/pricing

https://www.heroku.com/pricing

References

Baun, C., Kunze, M., Nimis, J., and Tai, S. (2011).
Cloud Computing. Web-Based Dynamic IT Services.
Springer Berlin Heidelberg.

Coulouris, G., Dollimore, J., Kindberg, T., and Blair, G. (2012).
Distributed Systems. Concepts and Design.
Pearson, 5th edition.

Hill, R., Hirsch, L., Lake, P., and Moshiri, S. (2013).
Guide to Cloud Computing. Principles and Practice.
Computer Communications and Networks. Springer London.

Mell, P. and Grance, T. (2011).
The NIST Definition of Cloud Computing.
https://csrc.nist.gov/publications/detail/sp/800-145/final.

Cloud Computing

IFTS 19/20: modulo "Industria 4.0 – Tecnologie Informatiche di Riferimento"

Giovanni Ciatto
giovanni.ciatto@unibo.it

Dipartimento di Informatica – Scienza e Ingegneria (DISI)
Alma Mater Studiorum – Università di Bologna a Cesena

February 14, 2020