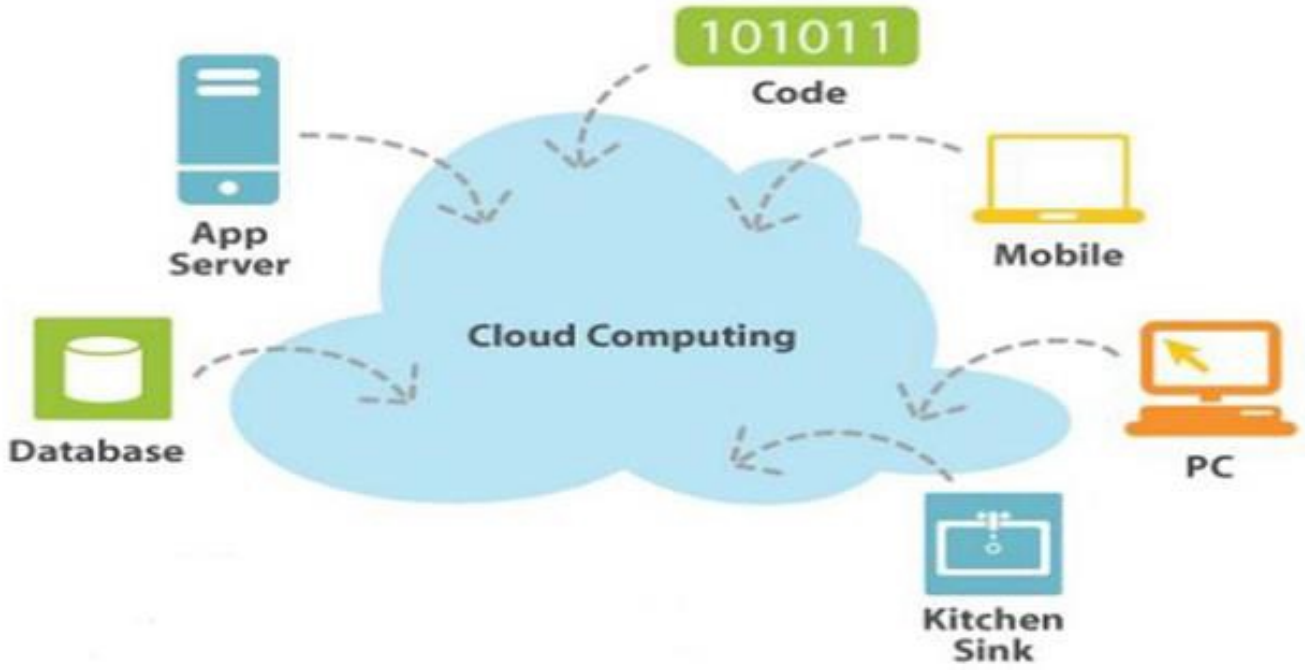
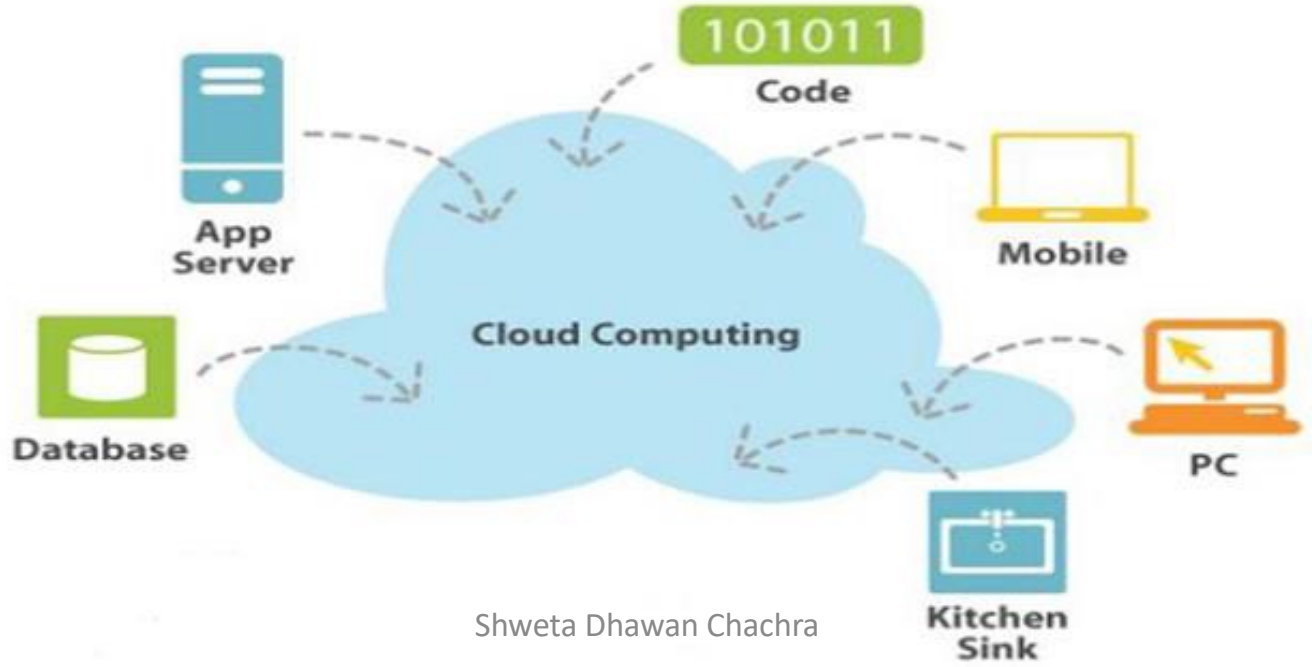
Cloud Computing

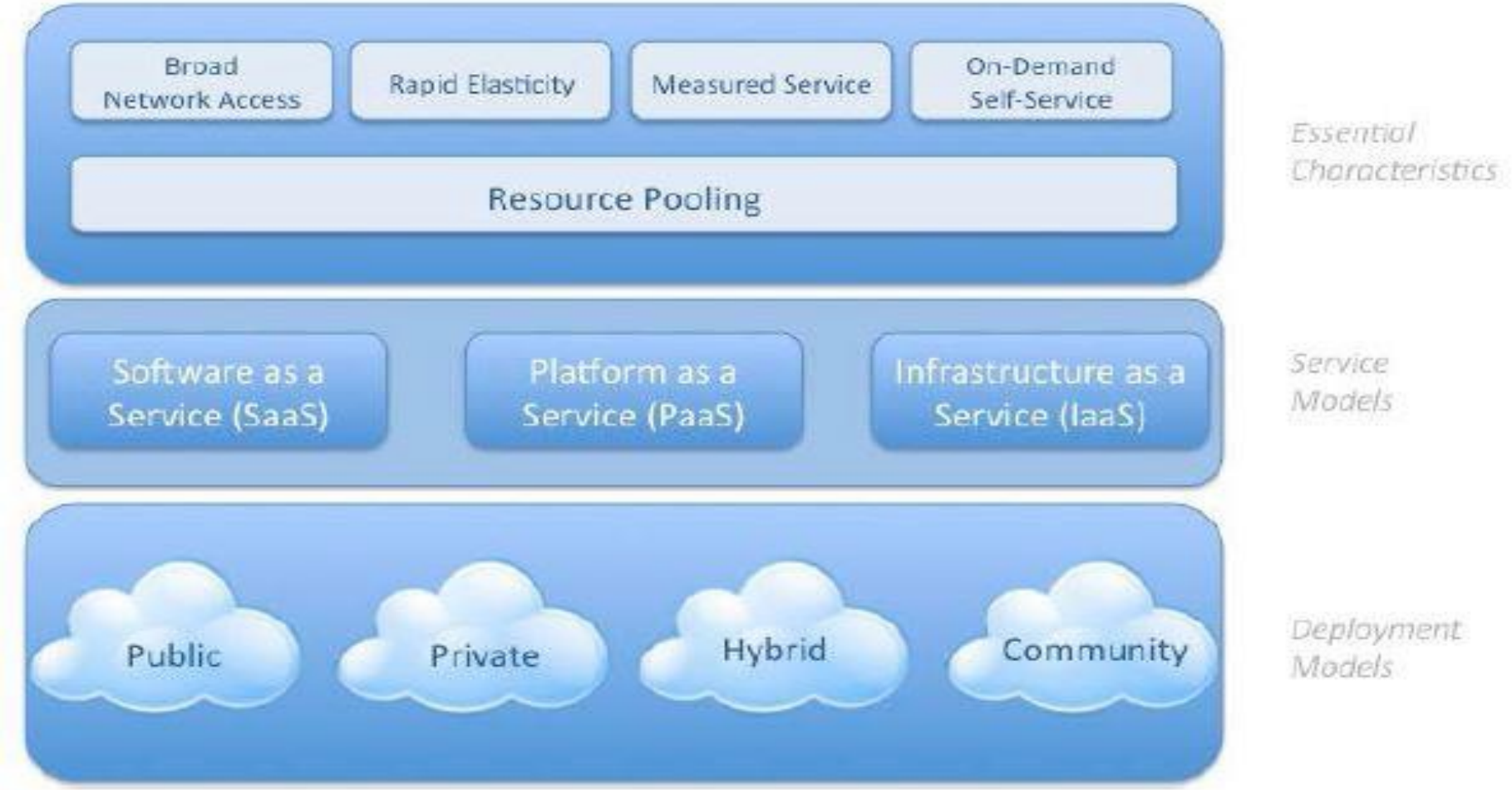
- US **National Institute of Standards and Technology (NIST)'s 3-4-5 definition**



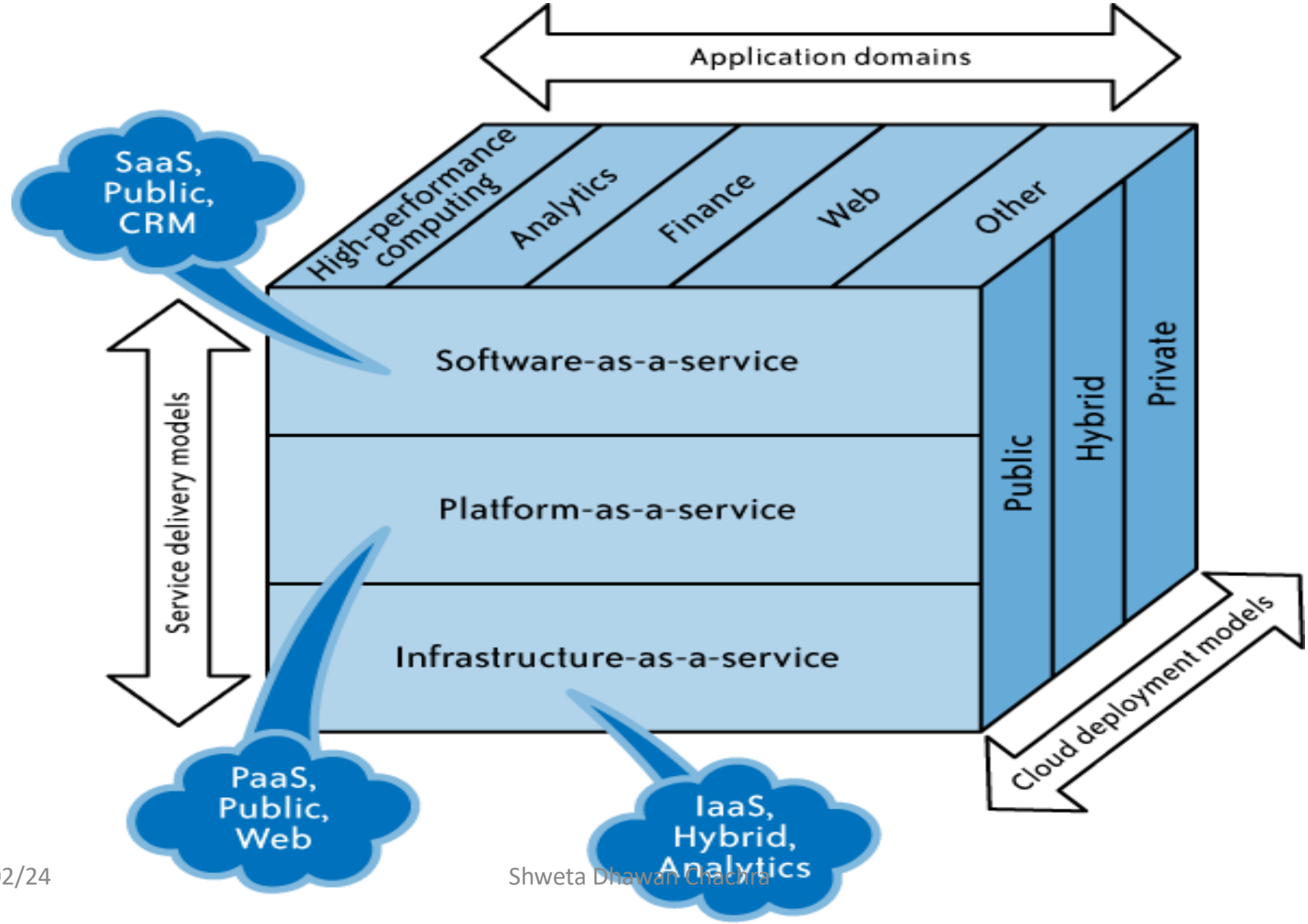
- US **National Institute of Standards and Technology (NIST)** defines Computing as:
- " Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. "



NIST Cloud Definition – Pictorial representation



Cloud Computing (Definition)



Essential Characteristics

On-demand self-service

A consumer can unilaterally provision computing capabilities as needed automatically without requiring human interaction with each service provider.

Essential Characteristics

Broad network access

Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations).

Essential Characteristics

Resource pooling

The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand.

Essential Characteristics

Measured Service

Cloud systems automatically control and optimize resource use by leveraging a metering capability at some level of abstraction appropriate to the type of service .

Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service.

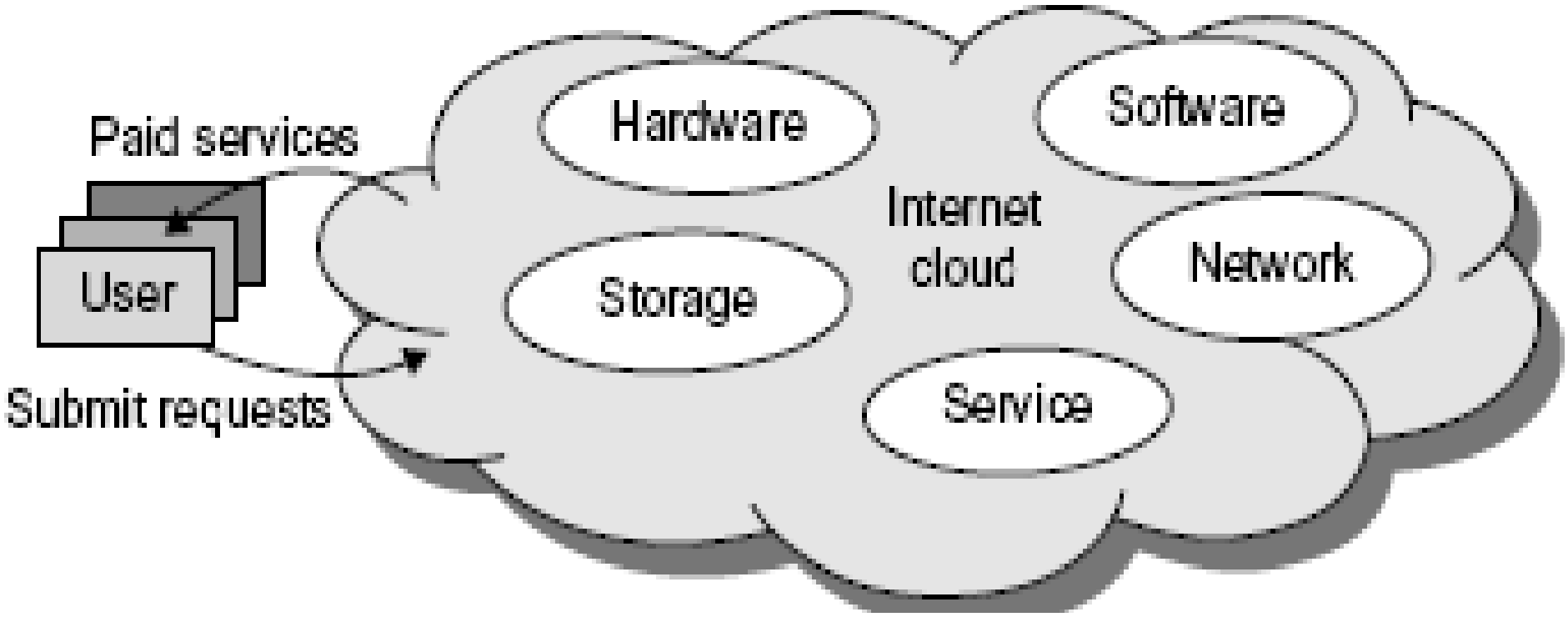
Essential Characteristics

Rapid elasticity

Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand.

To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time.

Basic Concept of Internet Clouds



Cost Model

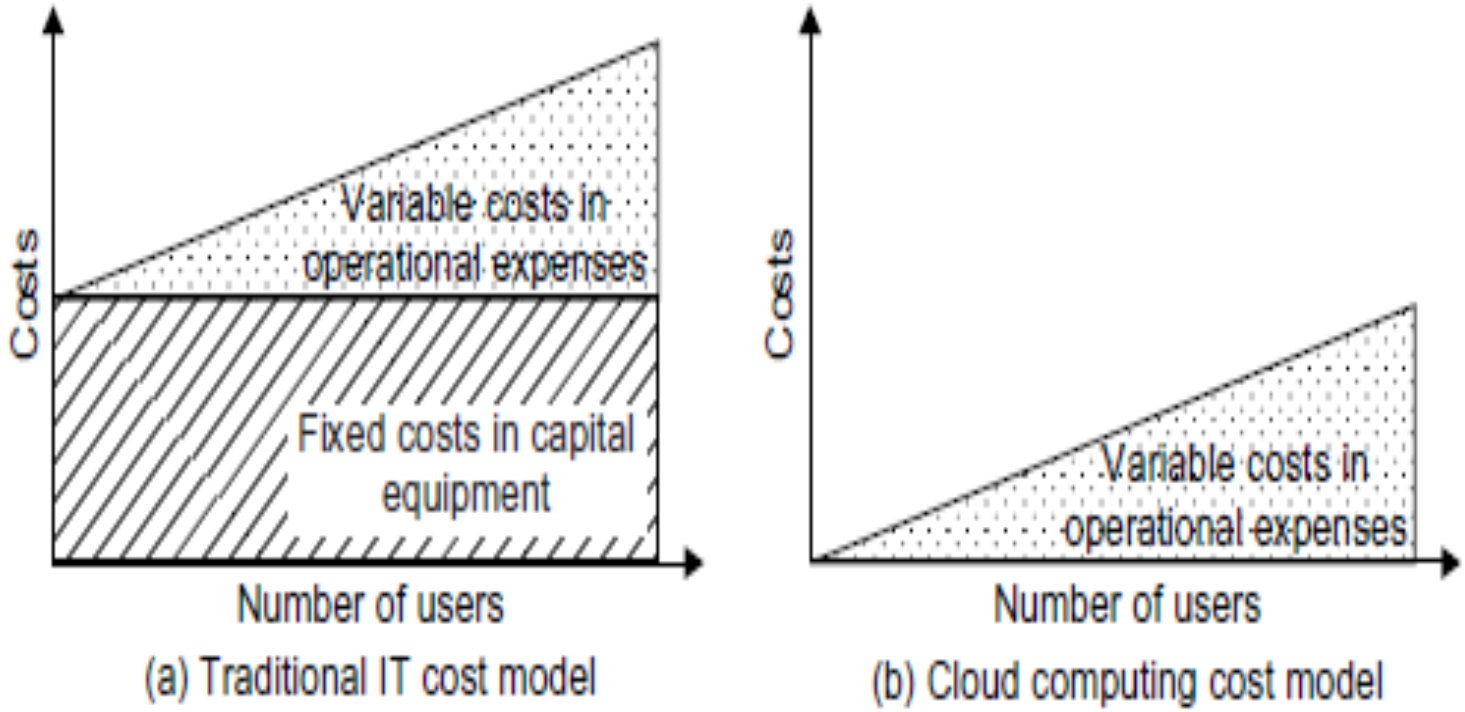


FIGURE 4.3

Computing economics between traditional IT users and cloud users.

Traditional Cost Model

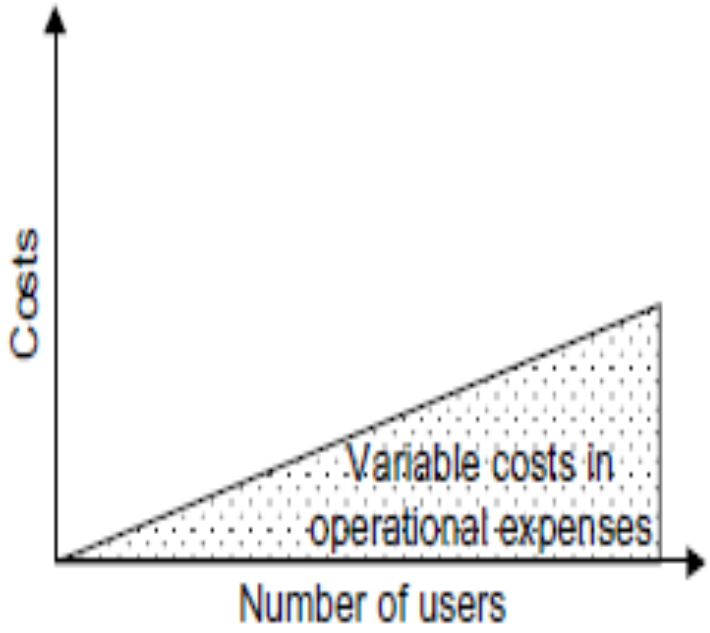
- Users must acquire their own computer and peripheral equipment as capital expenses.
- In addition, they have to face operational expenditures –
  - in operating and – **(Power & cooling)**
  - maintaining the computer systems,
  - including personnel and service costs.

- **The operational costs may increase sharply with a larger number of users.**

- **Therefore, the total cost escalates quickly with massive numbers of users.**

Cloud Computing Cost Model

- Cloud computing applies a **pay-per-use business model,** in which user jobs are **outsourced to data centers.**
  - To use the cloud, **one has no up-front cost in hardware acquisitions.**
  - **Only variable costs are experienced by cloud users**



(b) Cloud computing cost model

The Next Revolution in IT -Cloud Computing

Cloud Ecosystem

Classical Computing

–Buy & Own
  – Hardware, System Software, Applications often to meet peak needs.
–Install, Configure, Test, Verify, Evaluate
–Manage
–..
–Finally, use it
–$$$$....$(High CapEx)

Cloud Computing

–Subscribe
–Use
–$ - pay for what you use, based on QoS

Cloud Design Objectives

The following list highlights six design objectives for cloud computing:

- Shifting computing from desktops to data centers
  - Computer processing, storage, and software delivery is shifted away from desktops and local servers and toward data centers over the Internet.
- Service provisioning and cloud economics
  - Providers supply cloud services by signing SLAs with consumers and end users. The services must be efficient in terms of computing, storage, and power consumption. Pricing is based on a pay-as-you-go policy.
- Scalability in performance
  - The cloud platforms and software and infrastructure services must be able to scale in performance as the number of users increases.

Cloud Design Objectives

- Data privacy protection
  - Can you trust data centers to handle your private data and records? This concern must be addressed to make clouds successful as trusted services.
- High quality of cloud services
  - The QoS of cloud computing must be standardized to make clouds interoperable among multiple providers.
- New standards and interfaces
  - This refers to solving the data lock-in problem associated with data centers or cloud providers.
  - Universally accepted APIs and access protocols are needed to provide high portability and flexibility of virtualized applications.

Advantages of Cloud Computing

- **Lower computer costs**:

–No need of a high-powered and high-priced computer to run cloud computing's web-based applications.

–Since applications run in the cloud, not on the desktop PC, your desktop PC does not need the processing power or hard disk space demanded by traditional desktop software.

–When you are using web-based applications, your PC can be less expensive, with a smaller hard disk, less memory, more efficient processor...

–In fact, your PC in this scenario does not even need a CD or DVD drive, as no software programs have to be loaded and no document files need to be saved.

Advantages of Cloud Computing

- **Improved performance:**

–With few large programs hogging your computer's memory, you will see better performance from your PC.

–Computers in a cloud computing system boot and run faster because they have fewer programs and processes loaded into memory.

- **Reduced software costs:**

–Instead of purchasing expensive software applications, you can get most of what you need for free.

- most cloud computing applications today, such as the Google Docs suite.

–better than paying for similar commercial software

- which alone may be justification for switching to cloud applications.

Advantages of Cloud Computing

- **Instant software updates**

–Another advantage to cloud computing is that you are no longer faced with choosing between obsolete software and high upgrade costs.

–When the application is web-based, updates happen automatically available the next time you log into the cloud.

–When you access a web-based application, you get the latest version without needing to pay for or download an upgrade.

- **Improved document format compatibility.**

–You do not have to worry about the **documents you create on your machine being compatible with other users' applications or OS.**

–There are less format incompatibilities when everyone is sharing documents and applications in the cloud.

Advantages of Cloud Computing

- **Unlimited storage capacity**

–Cloud computing offers **virtually limitless storage.**

–**Your computer's current 1 Tera Bytes hard drive is small compared to the hundreds of Peta Bytes available in the cloud.**

- **Increased data reliability**

–Unlike desktop computing, in which if **a hard disk crashes and destroy all your valuable data, a computer crashing in the cloud should not affect the storage of your data.**

  - **if your personal computer crashes, all your data is still out there in the cloud, still accessible**

Advantages of Cloud Computing

- **Universal information access**

–That is not a problem with cloud computing, **because you do not take your documents with you.**

–**Instead, they stay in the cloud, and you can access them whenever you have a computer and an Internet connection**

- **Latest version availability**

–**When you edit a document at home, that edited version is what you see when you access the document at work.**

–The cloud always hosts the latest version of your documents as long as you are connected, you are not in danger of having an outdated version.

Advantages of Cloud Computing

- **Easier group collaboration**

–**Sharing documents leads directly to better collaboration.**

–Many users do this as it is an important advantages of cloud computing **multiple users can collaborate easily on documents and projects**

- **Device independence**

–**You are no longer tethered to a single computer or network.**

—Move to a portable device, and your applications and documents are still available.

Disadvantages of Cloud Computing

Requires a constant internet connection

–Cloud computing is **impossible if you cannot connect to the Internet.**

–Since you use the Internet to connect to both your applications and documents, if you do not have an Internet connection you cannot access anything, even your own documents.

–**A dead Internet connection means no work and in areas where Internet connections are few or inherently unreliable,**

this could be a deal-breaker.

Disadvantages of Cloud Computing

Does not work well with low-speed connections

–Similarly, **a low-speed Internet connection, such as that found with dial-up services, makes cloud computing painful** at best and often impossible.

–**Web-based applications require a lot of bandwidth to download, as do large documents.**

Disadvantages of Cloud Computing

Features might be limited

–This situation is bound to change, but today **many web-based applications simply are not as full-featured as their desktop-based applications.**

For example, you can **do a lot more with Microsoft PowerPoint than with Google Presentation's web-based offering**

Disadvantages of Cloud Computing

Can be slow

–Even with a fast connection, web-based applications can sometimes be slower than accessing a similar software program on your desktop PC.

–Everything about the program, from the interface to the current document, has to be sent back and forth from your computer to the computers in the cloud.

–If the cloud servers happen to be backed up at that moment, or

–If the Internet is having a slow day, you would not get the instantaneous access you might expect from desktop applications.

Disadvantages of Cloud Computing

Stored data might not be secured

–With cloud computing, all your data is stored on the cloud.

•The questions is **How secure is the cloud?**

–**Can unauthorized users gain access to your confidential data ?**

Stored data can be lost!

–Theoretically, data stored in the cloud is safe, **replicated across multiple machines.**

–But on the off chance that your data goes missing, you have no physical or local backup.

•Put simply, relying on the cloud puts you at risk if the cloud lets you down.

Disadvantages of Cloud Computing

HPC Systems

–Not clear that you can run compute-intensive HPC applications that use MPI/OpenMP!

–Scheduling is important with this type of application as you want all the VM to be co-located to minimize communication latency!

General Concerns

–Each **cloud systems uses different protocols and different APIs**

- **may not be possible to run applications between cloud based systems**

–Amazon has created its own DB system (not SQL 92), and workflow system (many popular workflow systems out there)

- so your normal applications will have to be adapted to execute on these platforms.

Cloud Computing Challenges:

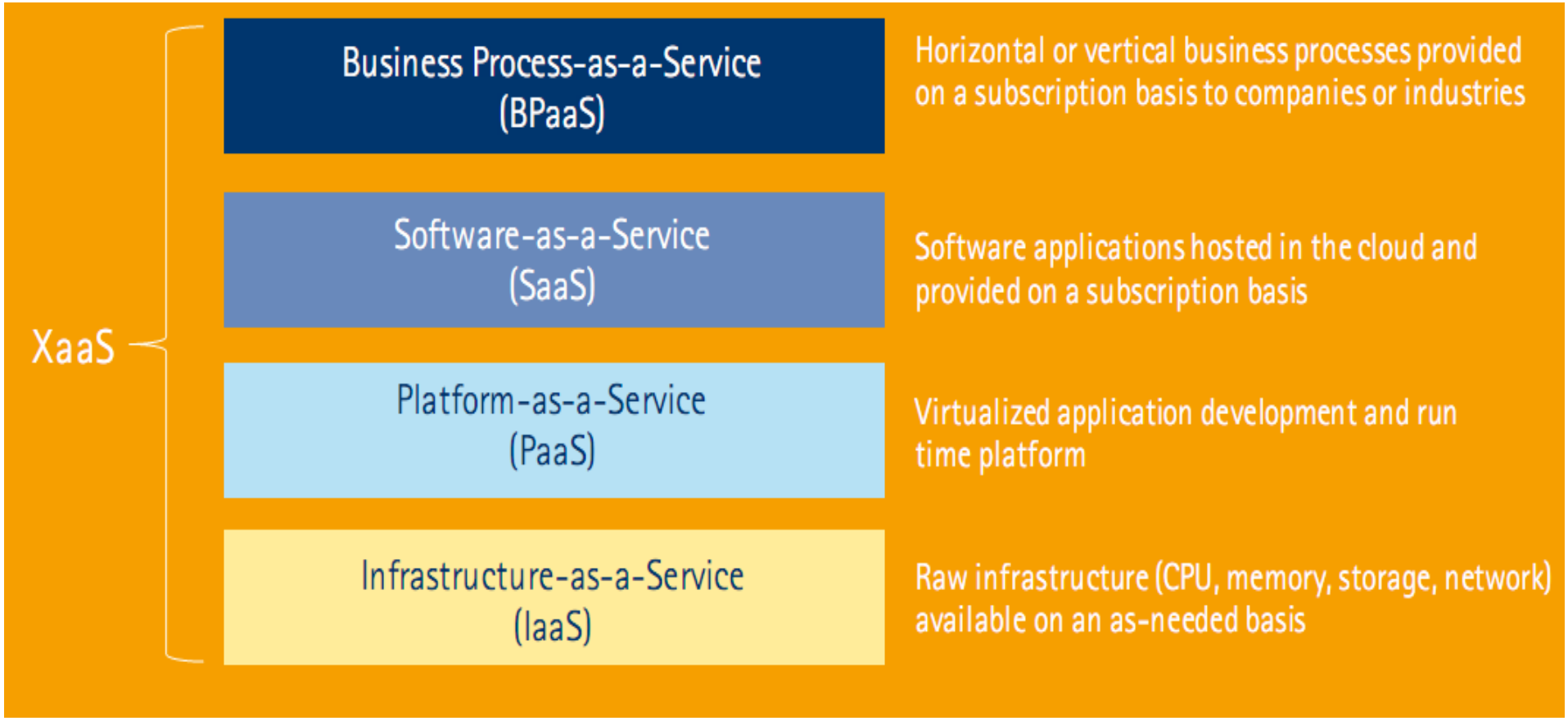
Dealing with too many issues (Courtesy of R. Buyya)

Service Models (XaaS)

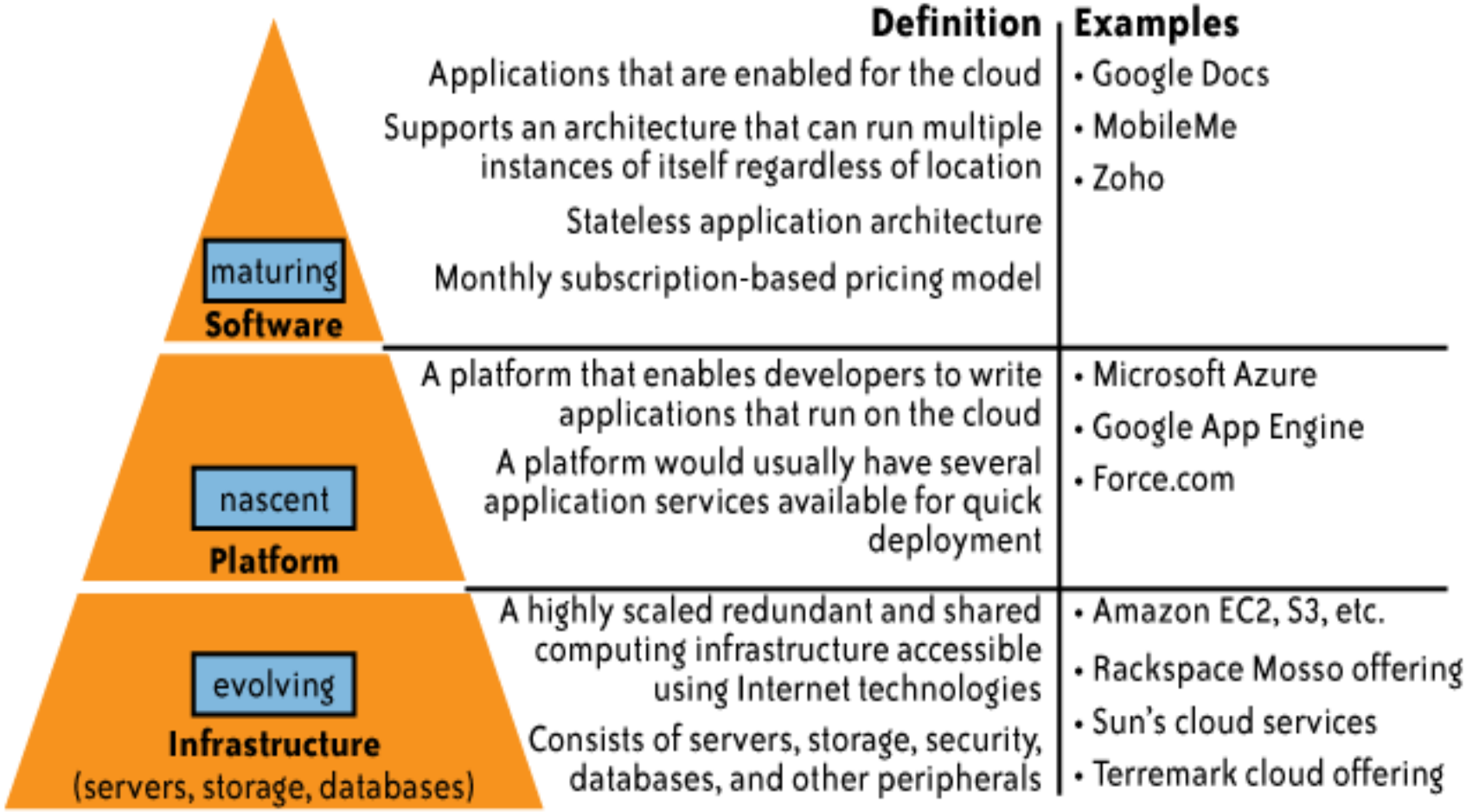
Combination of Service-Oriented Infrastructure (SOI) and cloud computing realizes to XaaS.

- **X as a Service (XaaS) is a generalization for cloud-related services**
- **XaaS stands for "anything as a service" or "everything as a service“**
- **XaaS refers to an increasing number of services that are delivered over the Internet** rather than provided locally or on-site
- XaaS is the essence of cloud computing.

Service Models (XaaS)



Service Models (XaaS)



Basic Service Models

SaaS - Software-as-a-Service Where the business subscribes to an application it accesses over the Internet.

PaaS - Platform-as-a-Service Where a business can create its own custom applications for use by all in the company.

IaaS - Infrastructure-as-a-Service

Where players like Amazon, Microsoft, Google, and Rackspace provide a backbone that can be "rented out" by other companies.

(IaaS is the game changer as it allows clients to fully outsourced service on demand rather than purchasing servers, software, datacenter space or network equipment.)

DaaS - Desktop-as-a-Service

DaaS is a cloud service where the back-end of a virtual desktop is hosted by a cloud provider.

Service Models (XaaS)

Most common examples of XaaS are

- Software as a Service (SaaS)
- Platform as a Service (PaaS)
- Infrastructure as a Service (IaaS)

Other examples of XaaS include

- Business Process as a Service (BPaaS)
- Storage as a service (another SaaS)
- Security as a service (SECaaS)
- Database as a service (DaaS)
- Monitoring/management as a service (MaaS)
- Communications, content and computing as a service (CaaS)
- Identity as a service (IDaaS)
- Backup as a service (BaaS)
- Desktop as a service (DaaS)

Simplified description of cloud service models

- **SaaS** applications are designed for end users and are delivered over the web
- **PaaS** is the set of tools and services designed to make coding and deploying applications quickly and efficiently
- **IaaS** is the hardware and software that powers it all – servers, storage, network, operating systems

Software as a Service

- SaaS is defined as software that is deployed over the internet.

- With SaaS, a provider licenses an application to customers
  - a service on demand,
  - Either through a subscription, in a “pay-as-you-go” model, or
  - (increasingly) at no charge when there is opportunity to generate revenue from streams other than the user, such as from advertisement or user list sales.

- E.g. of SaaS: Sales Force Customer Relationship Management (CRM) software

Applications where SaaS is used

- Applications where there is significant interplay between organization and outside world. √

E.g. email newsletter campaign software

- Applications that have need for web or mobile access. √

E.g. mobile sales management software.

- **Software that is only to be used for a short term need.** √

- **Software where demand spikes significantly.** √

E.g. Tax/Billing softwares.

Applications where SaaS may not be the best option

- Applications where extremely fast processing of real time data is needed √

- **Applications where legislation or other regulation does not permit data being hosted externally** √

- **Applications where an existing on-premise solution fulfills all of the organization's needs** √

Platform as a Service

- PaaS can be defined as a computing platform that allows
  - **creation of web applications quickly and easily and**
  - **without the complexity of buying and maintaining the software and infrastructure underneath it.**

Platform as a Service

- PaaS is analogous to SaaS except that,

Rather than being software delivered over the web, it is a platform for the creation of software, delivered over the web.

PaaS Examples: Microsoft Azure, Google App Engine

Scenarios where PaaS is used

- PaaS is especially useful in any situation where
  - **multiple developers will be working on a development project**
  - **Or where other external parties need to interact with the development process .**

- PaaS is useful where developers **wish to automate testing and deployment services.**

Scenarios where PaaS is used

- The popularity of agile software development, a group of software development **methodologies based on iterative and incremental development,**
  - **will also increase the uptake of PaaS as it eases the difficulties around rapid development and iteration of software.**

Scenarios where PaaS is not ideal

- **Where proprietary languages or approaches would impact on the development process** √
- **Where a proprietary language would hinder later moves to another provider** – concerns are raised about **vendor lock in** √
- Where application performance requires **customization of the underlying hardware and software** √

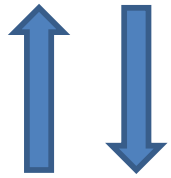
Infrastructure as a Service

- Infrastructure as a Service (IaaS) is a way of **delivering Cloud Computing infrastructure – servers, storage, network and operating systems** – as an on-demand service.

- Rather than purchasing servers,
  - software,
  - datacenter space or
  - network equipment,
  - clients instead buy those **resources as a fully outsourced service on demand.**

Scenarios where IaaS makes sense

- Where demand is very volatile – any time there are significant spikes and troughs in terms of demand on the infrastructure
- For **new organizations without the capital** to invest in hardware
- Where the **organization is growing rapidly and scaling** hardware would be problematic



Scenarios where IaaS makes sense

- Where there is **pressure on the organization to limit capital expenditure** and to move to operating expenditure
- For specific line of business, **trial or temporary infrastructural needs**

Scenarios where IaaS may not be the best option

- Where **regulatory compliance** makes the **offshoring or outsourcing of data storage and processing difficult.** √

- Where the **highest levels of performance** are required, and **on-premise or dedicated hosted infrastructure has the capacity** to meet the organization's needs √

SaaS providers

Provider	Software	Pricing model
Salesforce.com	CRM	Pay per use
Google Gmail	Email	Free
Process Maker Live	Business process management	Pay per use
XDrive	Storage	Subscription
SmugMug	Data sharing	Subscription
OpSource	Billing	Subscription
Appian Anywhere	Business process management	Pay per use
Box.net	Storage	Pay per use
MuxCLoud	Data processing	Pay per use

Feature comparison of PaaS providers

Provider	Target to Use	Programming language, Frameworks	Programming Models	Persistence options
Aneka	.NET enterprise applications, Web applications	.NET	Threads, Task, MapReduce	Flat files, RDBMS
AppEngine	Web applications	Python, Java	Request-based Web programming	BigTable
Force.com	Enterprise applications	Apex	Workflow, Request-based Web programming, Excel-like formula language	Own object database
Azure	Enterprise applications, Web applications	.NET	Unrestricted	Table/BLOB/queue storage, SQL Services
Heroku	Web applications	Ruby on Rails	Request-based Web programing	PostgreSQL, Amazon RDS
Amazon Elastic MapReduce	Data processing	Hive and Pirg, Cascading, Java, Ruby, Perl, Python, PHP, C++	MapReduce	Amazon S3

Feature comparison of IaaS providers

Provider	Geographic distribution of data centers	User interfaces and APIs	Hardware capacity	Guest operating systems	Smallest billing unit
Amazon E2C	US Europe	CLI, WS, Portal	CPU: 1_20 EC2 compute units Memory: 1.7-15 GB Storage: 160-1690 GB, 1 GB – 1 TB (per ESB units)	Linux Windows	Hour
Flexiscale	UK	Web console	CPU: 1-4 Memory: 0.5-16 GB Storage: 20-270 GB	Linux, Windows	Hour
GoGrid		REST, Java, PHP, Python, Ruby	CPU: 1-6 Memory: 0.5-8 GB Storage: 30-480 GB	Linux, Windows	Hour
Joyent	US		CPU: 1/16-8 Memory: 0.25-32.5 GB Storage: 5-100GB	OpenSolaris	Month
RackSpace	US	Portal, REST, Python, PHP, Java, .NET	CPU: Quad-core Memory: 0.25-16 GB Storage: 10-620 GB	Linux	Hour

Deployment Models

- Public Cloud
- Private Cloud
- Hybrid Cloud
- Community Cloud

Types of Cloud (Deployment Models)

Private cloud

The cloud infrastructure is operated solely for an organization.

e.g Window Server 'Hyper-V'.

Community cloud

The cloud infrastructure is shared by several organizations and supports a specific goal.

Public cloud

The cloud infrastructure is made available to the general public

e.g Google Doc, Spreadsheet,

Hybrid cloud

The cloud infrastructure is a composition of two or more clouds (private, community, or public)

e.g Cloud Bursting for load balancing between clouds.

Public Cloud

- Cloud infrastructure is provisioned for open use by the general public.
  - It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them.
  - It exists on the premises of the cloud provider.

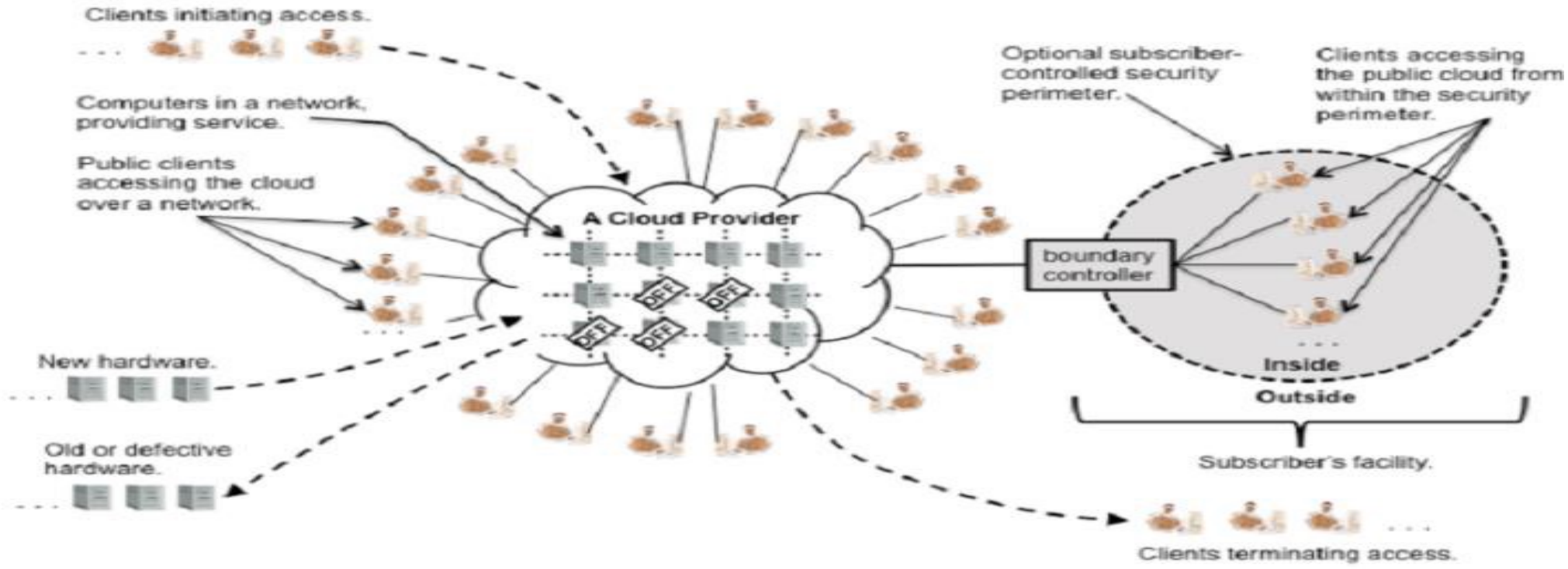
Public Cloud

Examples of Public Cloud:

- Google App Engine
- Microsoft Windows Azure
- IBM Smart Cloud
- Amazon EC2

Public Cloud

- In Public setting, the provider's computing and storage resources are potentially large;
  - the communication links can be assumed to be implemented over the public Internet; and
  - the cloud serves a diverse pool of clients (and possibly attackers).



Private Cloud

- The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units).
  - It may be owned, managed, and operated by the organization, a third party, or some combination of them, and
  - it may exist on or off premises.

Private Cloud

Contrary to popular belief, private cloud may exist off premises and can be managed by a third party. Thus, two private cloud scenarios exist, as follows:

- On-site Private Cloud

– Applies to private clouds implemented at a customer's premises.

- Outsourced Private Cloud

– Applies to private clouds where the server side is outsourced to a hosting company.

Private Cloud

Examples of Private Cloud:

- Eucalyptus
- Ubuntu Enterprise Cloud - UEC
- Amazon VPC (Virtual Private Cloud)
- VMware Cloud Infrastructure Suite
- Microsoft ECI data center.

Community Cloud

- Cloud infrastructure is provisioned for exclusive use by a specific community of consumers
- from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations).
  - It may be owned, managed, and operated by
  - one or more of the organizations in the community,
  - a third party, or
  - some combination of them, and
  - it may exist on or off premises.

Community Cloud

Examples of Community Cloud:

- Google Apps for Government
- Microsoft Government Community Cloud

Hybrid Cloud

- The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities,
  - but are bound together by standardized or proprietary technology that enables data and application portability

Hybrid Cloud

Examples of Hybrid Cloud:

–Windows Azure (capable of Hybrid Cloud)

–VMware vCloud (Hybrid Cloud Services)

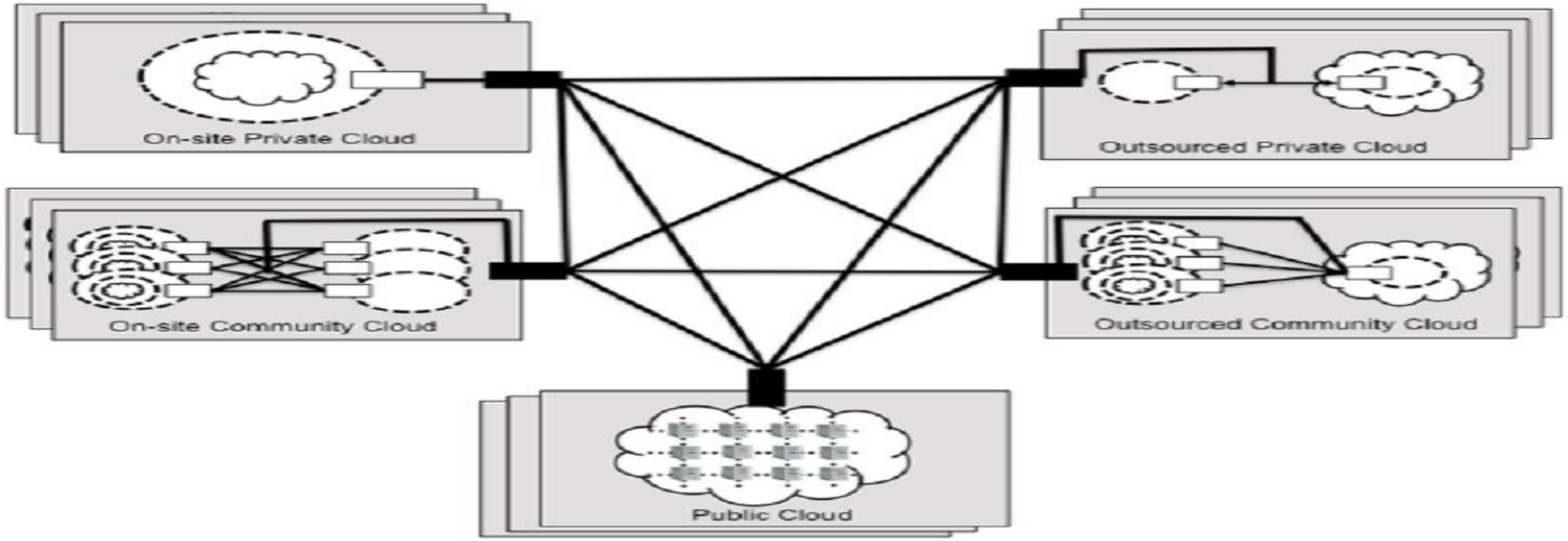
- Cloud Bursting for Load Balancing

Cloud Bursting for Load balancing

- Cloud bursting is a configuration which is set up between a private cloud and a public cloud to deal with peaks in IT demand.
  - If an organisation using a private cloud reaches 100 percent of its resource capacity,
  - the overflow traffic is directed to a public cloud so there is no interruption of services.
- It is the ability to allocate resources across various public and private clouds as an organization's needs change.

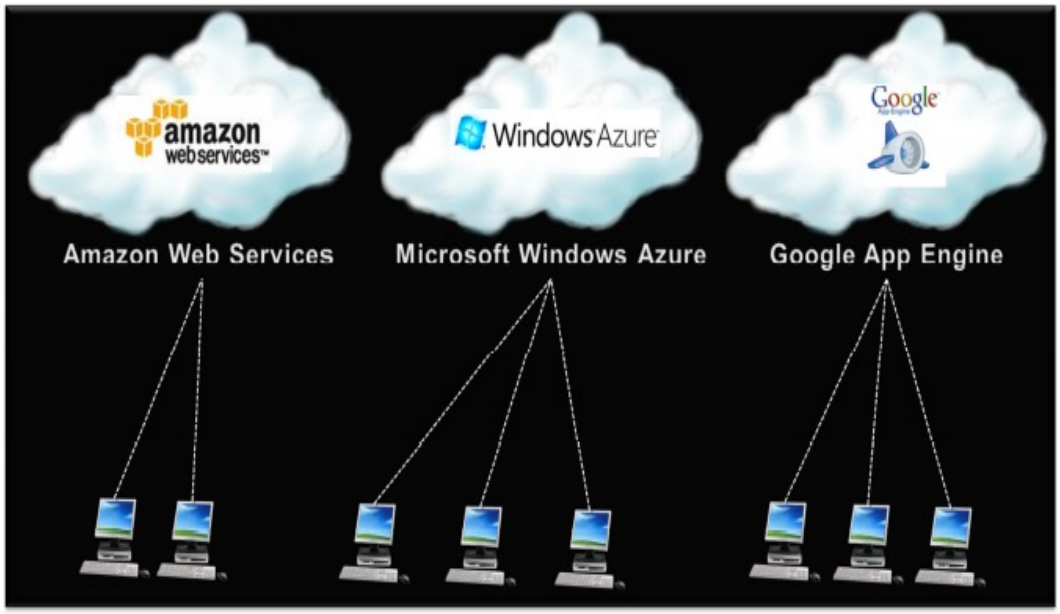
Hybrid Cloud

- A hybrid cloud is composed of two or more private, community, or public clouds.
- They have significant variations in performance, reliability, and security properties
  - depending upon the type of cloud chosen to build hybrid cloud.

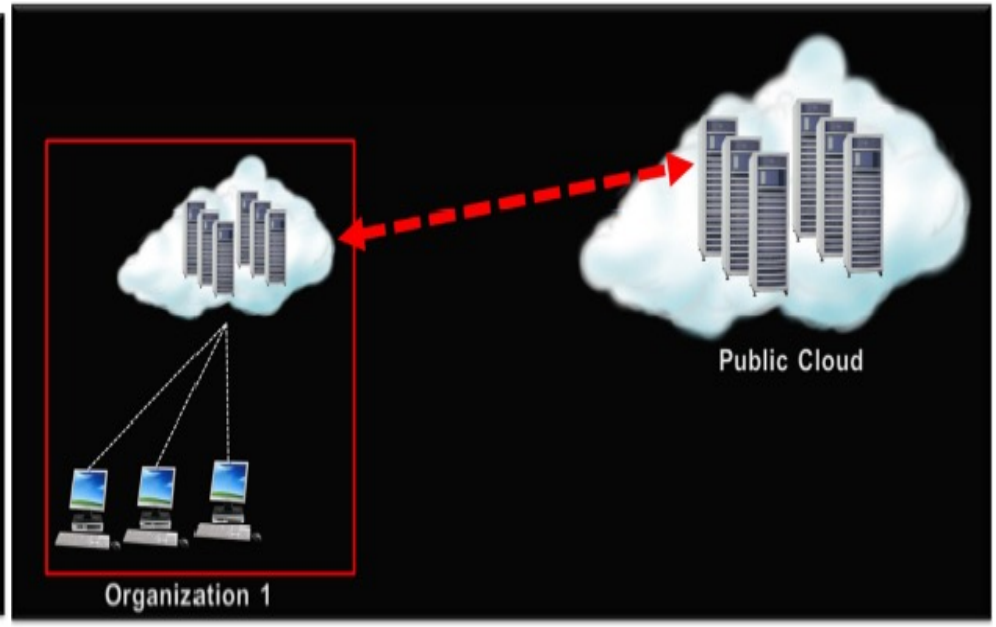


Hybrid Cloud

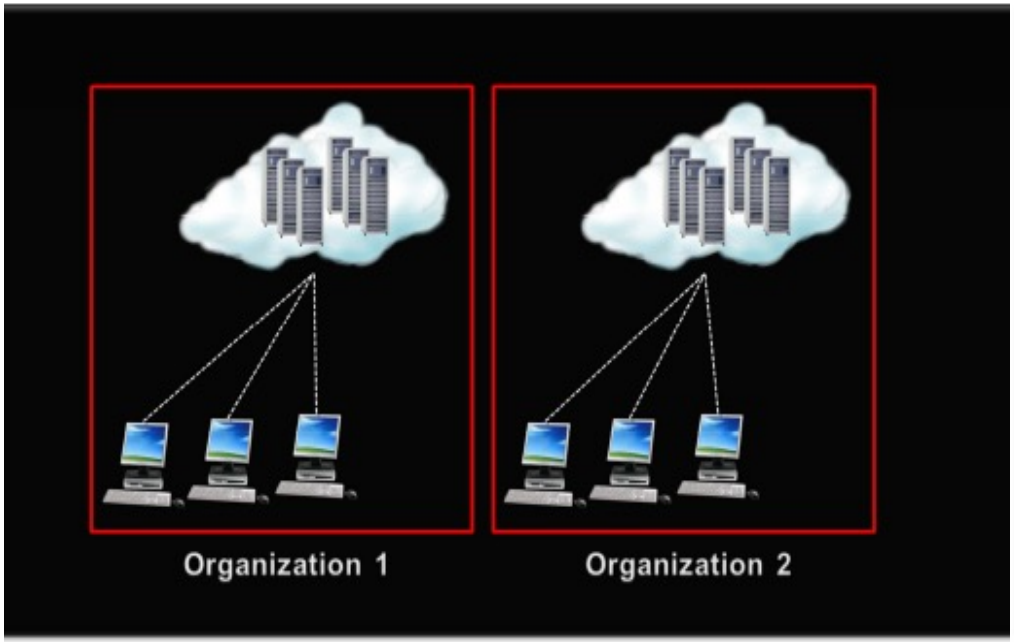
- A hybrid cloud can be extremely complex
- A hybrid cloud may change over time with constituent clouds joining and leaving.



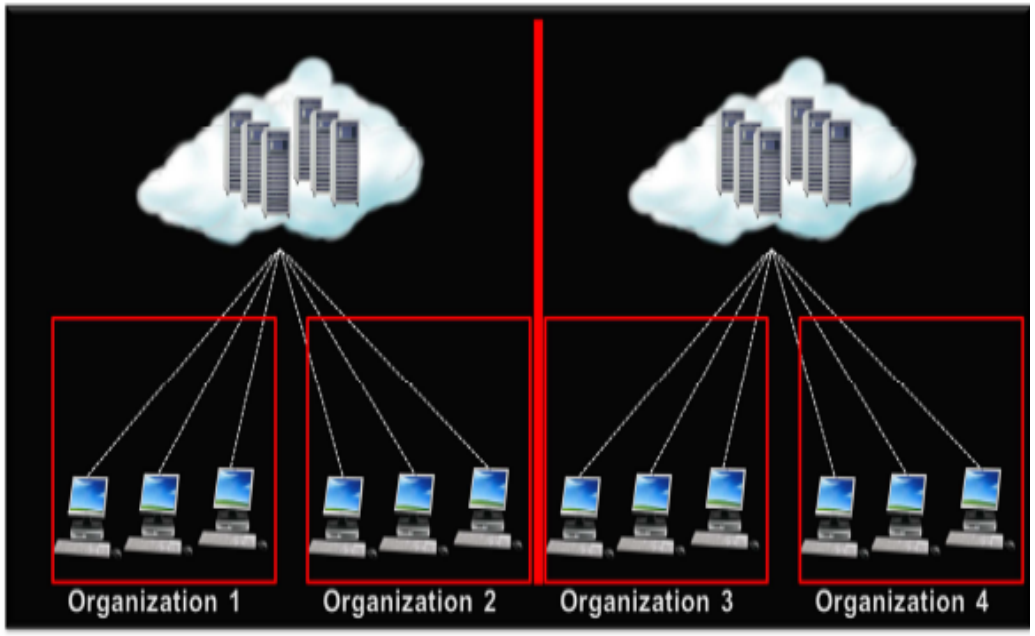
Public Cloud



Hybrid Cloud



Private Cloud



Community Cloud

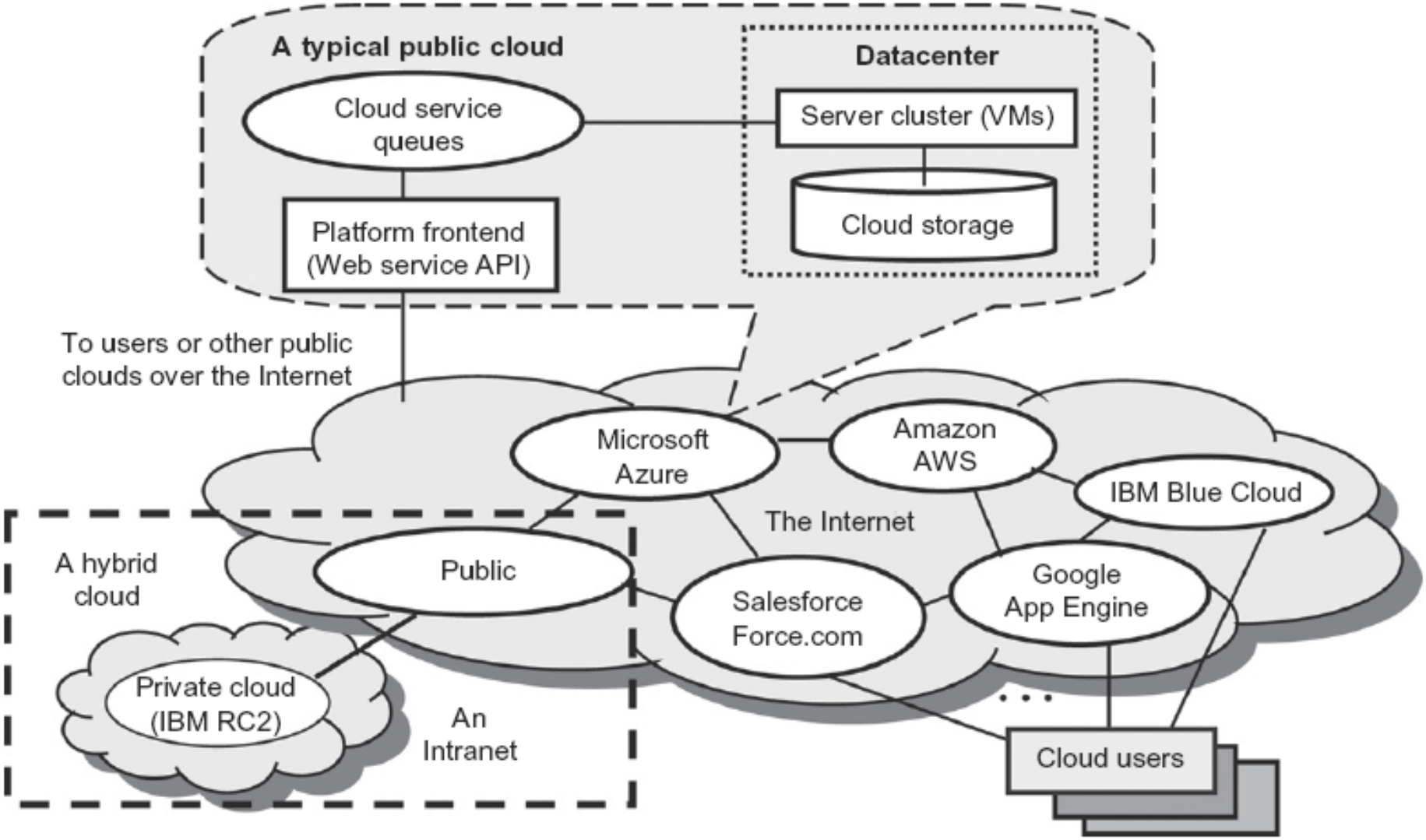
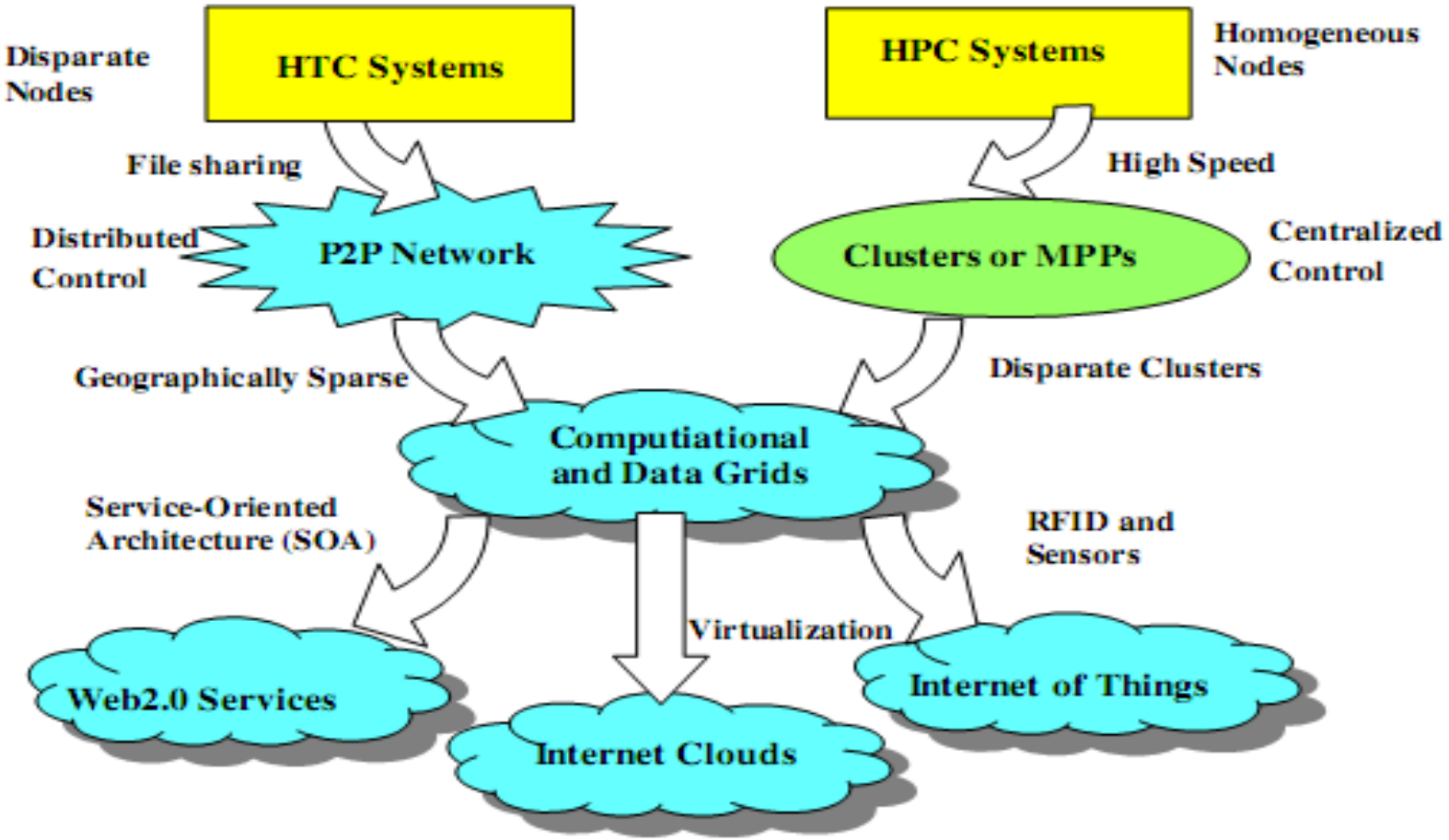


FIGURE 4.1

Public, private, and hybrid clouds.

Evolutionary trend



HPC

- HPC systems emphasize the raw speed performance.
- The speed of HPC systems has increased from Gflops (early 1990s) to Pflops (2010).

FLOPS-

In computing, floating point operations per second (FLOPS) is a measure of computer performance, useful in fields of scientific computations.

Name	Unit	Value
kiloFLOPS	kFLOPS	10^3
megaFLOPS	MFLOPS	10^6
gigaFLOPS	GFLOPS	10^9
teraFLOPS	TFLOPS	10^{12}
petaFLOPS	PFLOPS	10^{15}
exaFLOPS	EFLOPS	10^{18}
zettaFLOPS	ZFLOPS	10^{21}
yottaFLOPS	YFLOPS	10^{24}

HTC

- High Throughput Computing.
- HTC pays more attention to High flux computing.
- High flux computing application is in Internet searches and web services by millions or more users simultaneously.
- The performance goals thus shifts to measure high throughput or the number of tasks completed per unit of time.

HPC vs HTC

- HPC tasks are characterized as needing **large amounts of computing power for short periods of time,**
- HTC tasks also require **large amounts of computing, but for much longer times (months and years, rather than hours and days).**

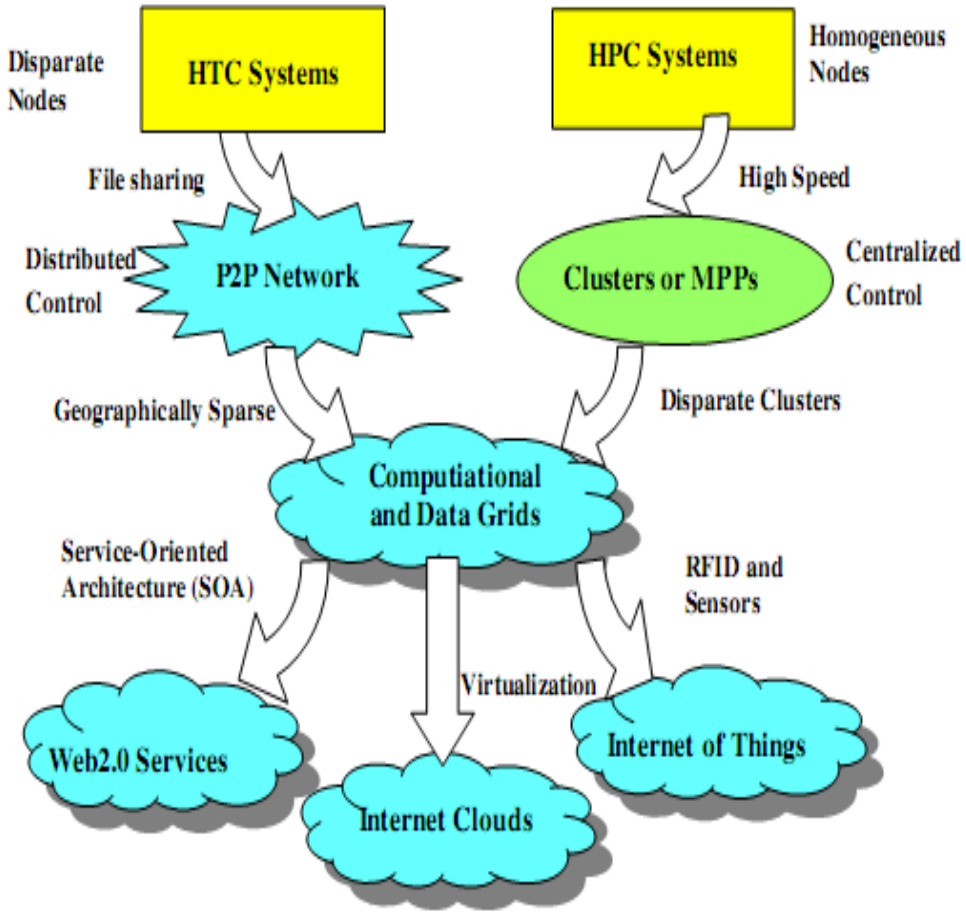
- HPC environments are often **measured in terms of FLOPS.**
- The HTC community, however, is not concerned about operations per second, but **rather operations per month or per year.**

- Therefore, the HTC field is more interested in **how many jobs can be completed over a long period of time instead of how fast.**

Courtesy: https://en.wikipedia.org/wiki/High-throughput_computing

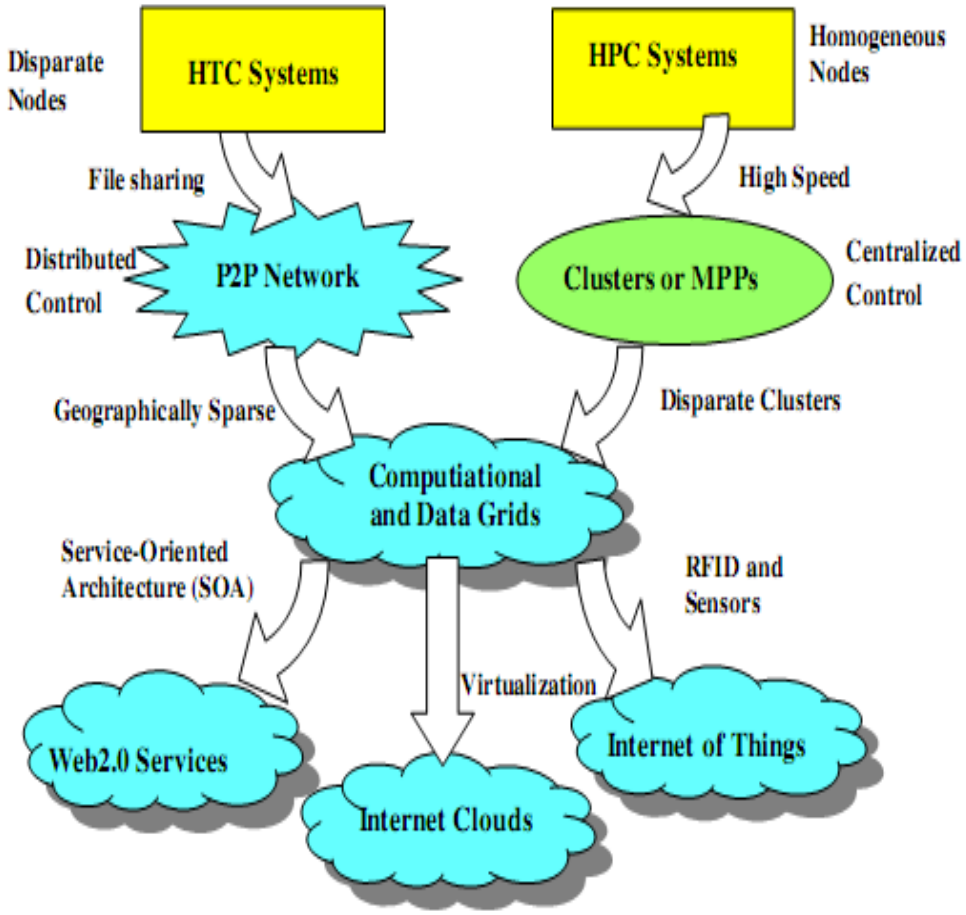
On HPC Side-

- On the HPC side, Supercomputers or MPPs (Massively Parallel Processors) are gradually replaced by clusters of Co-operative computers out of desire to **share computing resources.**



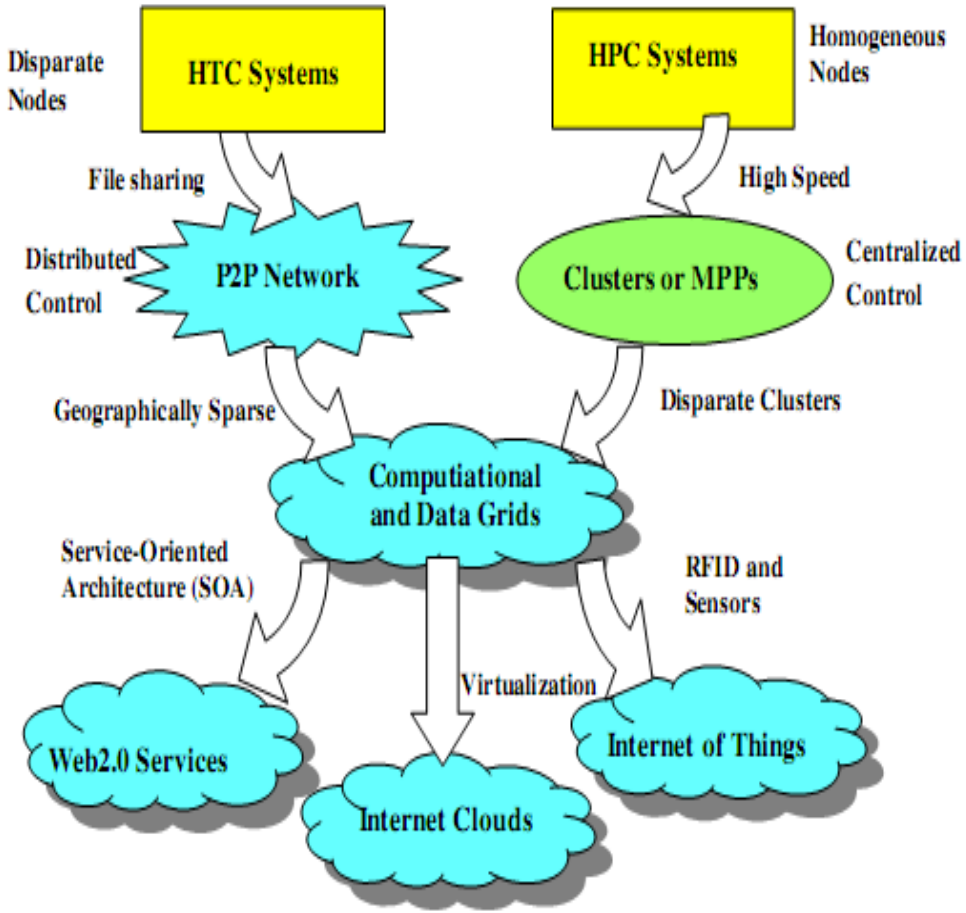
On HPC Side-

- Clusters are collection of homogenous compute nodes that are physically connected in **close range to one another.**

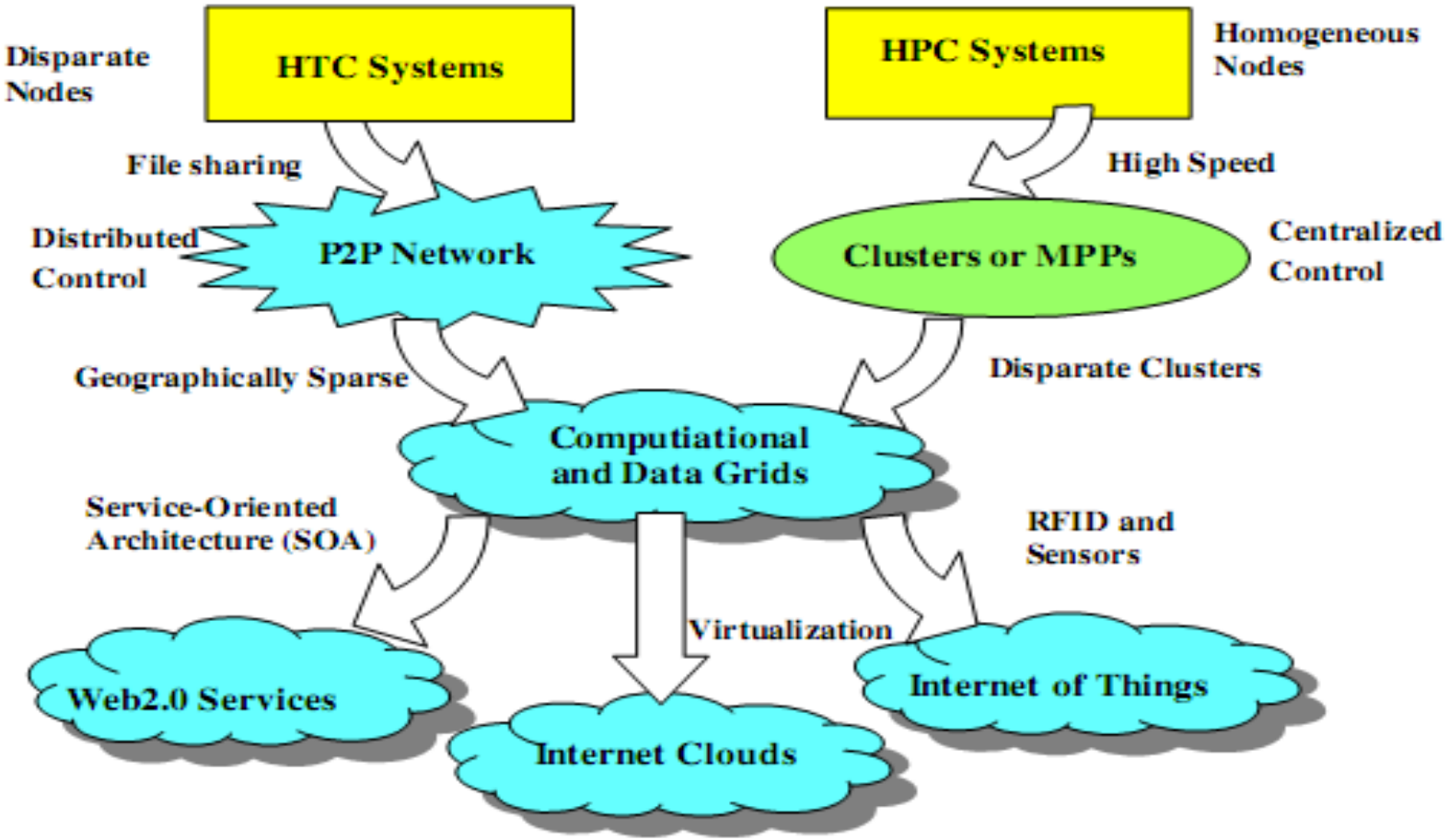


On HTC Side-

- P2P(Peer 2 Peer networks) are formed for distributed file sharing and content delivery applications.
- P2P is built over many client machines and **<u>are globally distributed in nature.</u>**
- P2P,Cloud computing and Web Services platforms are more focussed on HTC applications.



- **Clustering and P2P lead to development of Computational and data grids.**



Compute grid

- Compute grids allow you to take a computation, optionally split it into multiple parts, and execute them on different grid nodes in parallel.
- Computation will perform faster as it now can use resources from all grid nodes in parallel.
- Features include-
  - **load balancing**
  - **fail-over**
  - **grid events**
  - **node metrics**

Data grid

- Data grids allow you to distribute your data across the grid.
- The main goal of data grid is to provide as much data as possible from memory on every grid node and to ensure data coherency.
- Features include-
  - **data replication**
  - **data backups**
  - **data affinity/partitioning**

How is Grid Computing Used?

- All computing resources do not have to work on the same specific task, but can work on sub-tasks that collectively make up the end goal.
- **For example,**
- **A research team might analyze weather patterns in the North Atlantic region,**
- **Another team analyzes the south Atlantic region,**
- **Both results can be combined to deliver a complete picture of Atlantic weather patterns.**

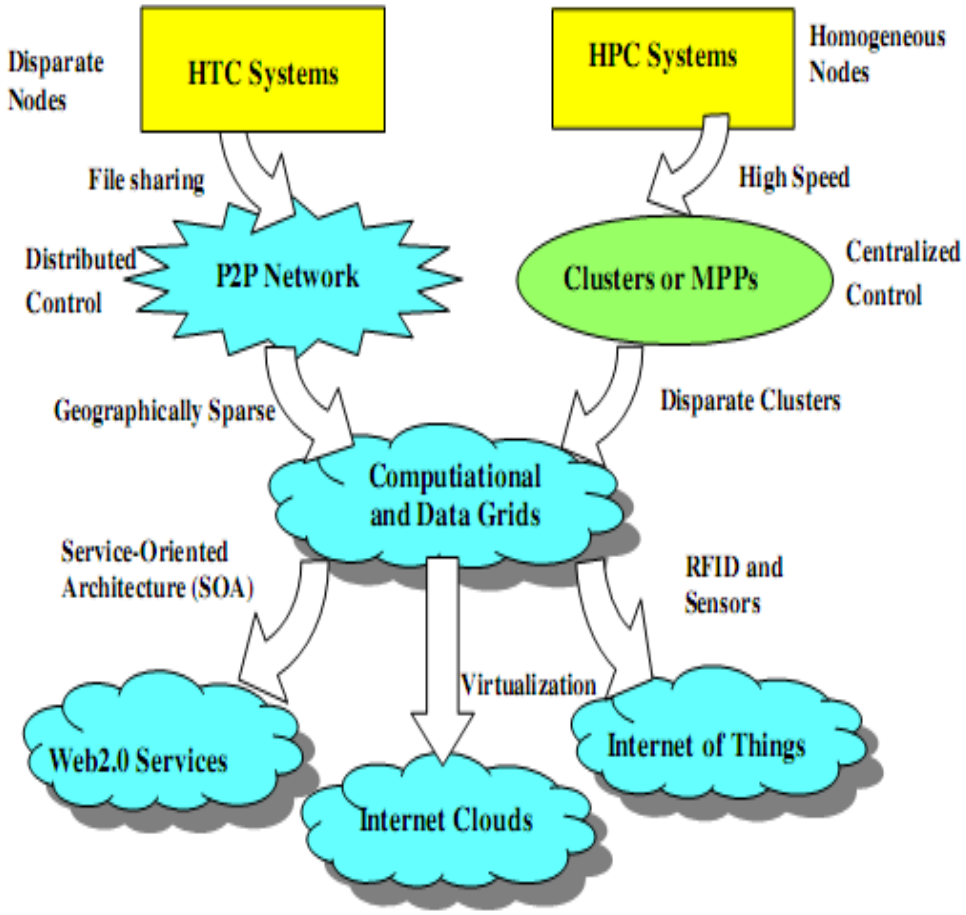
How is Grid Computing Used?

- Grid computing is especially useful when
  - different subject matter experts need to collaborate on a project but
  - do not necessarily have the means to immediately share data and computing resources in a single site.

- By joining forces despite the geographical distance,
- the distributed teams are able to leverage their own resources that contribute to a bigger effort.

3 New Computing Paradigms

- Internet Clouds
- Internet of things(IOT)
- Web 2.0 services



3 New Computing Paradigms

- The maturity of Radio frequency Identification(RFID), Global Positioning System(GPS) and sensor technologies has triggered the development of the Internet of things(IOT)

- With the Introduction of SOA, Web Services became available.

- Advances in Virtualization led to the growth of Internet Clouds.

Building cloud computing environments

The creation of cloud computing environments encompasses both

- **Application development**
- **Infrastructure and system development**

Building cloud computing environments

The creation of cloud computing environments encompasses both

- **Application development**
  - the development of applications and systems that leverage cloud computing solutions and
- **Infrastructure and system development**
  - the creation of frameworks, platforms, and infrastructures delivering cloud computing services

Application development

- Applications that leverage cloud computing benefit from its **capability to dynamically scale on demand.**
- One class of applications is that of **Web applications.**
- Their performance is mostly **influenced by the workload generated by varying user demands.**
- The Web has become a platform for developing rich and complex applications, **including enterprise applications for service delivery and user interaction.**

Application development

- Another class of applications that can potentially gain considerable advantage by leveraging cloud computing is represented by **resource-intensive applications**.
- These can be either **data intensive or compute-intensive applications.**
- In both cases, **considerable amounts of resources are required to complete execution in a reasonable timeframe.**
- It is worth noticing that these **large amounts of resources are not needed constantly or for a long duration.**

Application development

- For example, **scientific applications can require huge computing capacity to perform large-scale experiments once in a while**, so it is not feasible to buy the infrastructure supporting them.
- In this case, cloud computing can be the solution.
- **Resource-intensive applications are not interactive and they are mostly characterized by batch processing.**

Infrastructure and system development

- Cloud computing provides a solution for on-demand and dynamic scaling across the entire stack of computing.
- This is achieved by

(a) providing methods for renting compute power, storage, and networking;

(b) offering runtime environments designed for scalability and dynamic sizing; and

(c) providing application services that mimic the behavior of desktop applications but that are completely hosted and managed on the provider side.

Infrastructure and system development

- All these capabilities leverage **service orientated architecture,** which allows a simple and seamless integration into existing systems.
- Developers access such services via simple Web interfaces, often implemented through **representational state transfer (REST) Web services**

Infrastructure and system development

- The core technologies enabling the provisioning of cloud services -
  - **Distributed computing**
  - **Virtualization**
  - **Service orientation**
  - **Web 2.0/Web Services**
- Developing applications and systems that leverage the cloud requires knowledge across all these technologies.

What is a Web Service ?

Web service is a means by **which computers talk to each other over the web** using HTTP and other universally supported protocols.

A Web service is an application that:

- Runs on a Web server
- Exposes Web methods to interested callers
- Listens for HTTP requests representing commands to invoke Web methods
- Executes Web methods and returns the results

Web Services

- A web service is any piece of software that makes itself available over the internet and uses a standardized XML messaging system.
- XML is used to encode all communications to a web service.
  - For example, a client invokes a web service by sending an XML message, then waits for a corresponding XML response.
- As all communication is in XML, web services are not tied to any one operating system or programming language—Java can talk with Perl; Windows applications can talk with Unix applications.

Web Services

- “Programmable application logic accessible using Standard Internet Protocols…” – Microsoft

Software components that can be spontaneously discovered, combined, and recombined to provide a solution to the user's problem/request … " - SUN

Validation Web Services

http://www.webservicex.net/CreditCard.asmx?WSDL - validate a credit card number

http://www.webservicex.net/ValidateEmail.asmx?WSDL - validate email address

http://www.tpisoft.com/smartpayments/validate.asmx?WSDL - validate credit card number

http://ws.cdyne.com/emailverify/Emai...mail.asmx?wsdl - validate email

http://ws.cdyne.com/phoneverify/phoneverify.asmx?wsdl - validate phone number

Business Web Services

http://www.webservicex.net/stockquote.asmx?WSDL - get stock quote of a particular company

"Software application identified by a URI, whose interfaces and bindings are capable of being defined, described, and discovered as XML artifacts" – W3C Web Services Architecture Requirements, Oct. 2002

Scientific Web Services

http://www.webservicex.net/ConvertAc...tion.asmx?WSDL - Convert acceleration

http://www.webservicex.net/CovertPressure.asmx?WSDL - Convert Pressure

http://www.webservicex.net/ConvertDensity.asmx?WSDL - Convert Density

http://www.webservicex.net/ConverPower.asmx?WSDL - Convert Power

http://www.webservicex.net/ConvertAngle.asmx?WSDL - Convert Angles

http://www.webservicex.net/ConvertTorque.asmx?WSDL - Convert Torque

http://www.webservicex.net/convertMe...ight.asmx?WSDL - Convert Weight

http://www.webservicex.net/convertVolume.asmx?WSDL - Convert Volume

http://www.webservicex.net/ConvertTemperature.asmx?WSDL - Convert Temperature

http://www.webservicex.net/convertFrequency.asmx?WSDL - Convert Frequency

http://www.webservicex.net/Astronomical.asmx?WSDL - Convert Astronomical Values (i.e. - the speed of light)

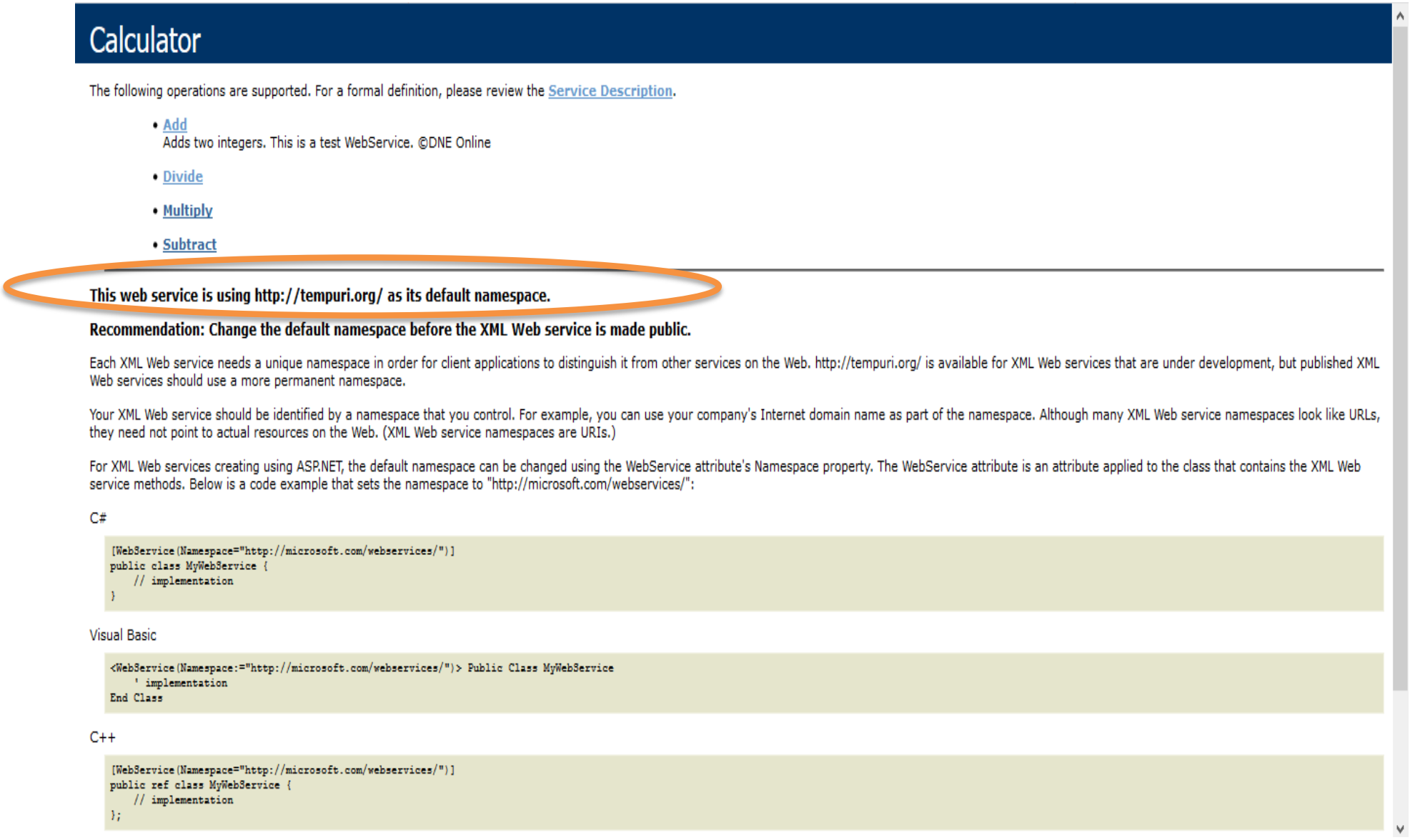
http://www.webservicex.net/ConvertForec.asmx?WSDL - Convert Force

http://www.webservicex.net/ConvertEnergy.asmx?WSDL - Convert Energy

http://www.webservicex.net/ConvertArea.asmx?WSDL - Convert Area

http://www.webservicex.net/ConvertCooking.asmx?WSDL - Convert Cooking units

Some Free Web services Available Online-Calculator Web Service



Calculator

The following operations are supported. For a formal definition, please review the Service Description.

- Add
  Adds two integers. This is a test WebService. ©DNE Online
- Divide
- Multiply
- Subtract

This web service is using http://tempuri.org/ as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://tempuri.org/ is available for XML Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to "http://microsoft.com/webservices/":

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementation
}
```

Visual Basic

```
<WebService(Namespace:="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementation
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // implementation
};
```

Processes XML messages framed using SOAP

Calculator

Click here for a complete list of operations.

Add

Adds two integers. This is a test WebService. ©DNE Online

Test

The test form is only available for requests from the local machine.

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /calculator.asmx HTTP/1.1
Host: www.dneonline.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/Add"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Add xmlns="http://tempuri.org/">
      <intA>int</intA>
      <intB>int</intB>
    </Add>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <AddResponse xmlns="http://tempuri.org/">
      <AddResult>int</AddResult>
    </AddResponse>
  </soap:Body>
</soap:Envelope>
```

Processes XML messages framed using SOAP-Temperature Conversion Web Service

TempConvert

Click here for a complete list of operations.

CelsiusToFahrenheit

Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
Celsius:	23

Invoke

SOAP 1.1

The following is a sample SOAP 1.1 request and response. The placeholders shown need to be replaced with actual values.

```
POST /xml/tempconvert.asmx HTTP/1.1
Host: www.w3schools.com
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "https://www.w3schools.com/xml/CelsiusToFahrenheit"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CelsiusToFahrenheit xmlns="https://www.w3schools.com/xml/">
      <Celsius>string</Celsius>
    </CelsiusToFahrenheit>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CelsiusToFahrenheitResponse xmlns="https://www.w3schools.com/xml/">
      <CelsiusToFahrenheitResult>string</CelsiusToFahrenheitResult>
    </CelsiusToFahrenheitResponse>
  </soap:Body>
</soap:Envelope>
```

What is a Web Service ?

Web Services is based on:

- **HTTP (Hypertext Transport Protocol)**
- **SOAP (Simple Object Access Protocol)**
- **UDDI (Universal Description, Discovery and Integration)**
- **WS-POLICY (Web Services Policy)**

Most Web services expect their Web methods to be invoked using

HTTP requests containing SOAP messages.

SOAP is an XML-based vocabulary for performing remote procedure calls using HTTP and other protocols.

Web Service Components

- **XML** – eXtensible Markup Language

Uniform data representation and exchange mechanism.

- **SOAP** – Simple Object Access Protocol

A standard way for communication.

- **WSDL** – Web Services Description Language

A standard meta language to describe the services offered.

- **UDDI** – Universal Description, Discovery and Integration specification

A mechanism to register and locate WS based application.

	Web Service	Website
1.	A web service doesn't have a user interface	A website has a user interface or GUI
2.	Web services are meant for other applications to be interacted with over internet	Websites are meant for use by humans
3.	Web services are platform independent as they use open protocols	Websites are cross platform as they require tweaking to operate on different browsers, operating systems etc.
4.	Web services are accessed by HTTP methods - GET, POST, PUT, DELETE etc	Websites are accessed by using their GUI components - buttons, text boxes, forms etc.
5.	E.g. Google maps API is a web service that can be used by websites to display Maps by passing coordinates to it	E.g. ArtOfTesting.com is website that has collection of related web pages containing tutorials

Service Oriented Architecture (SOA)

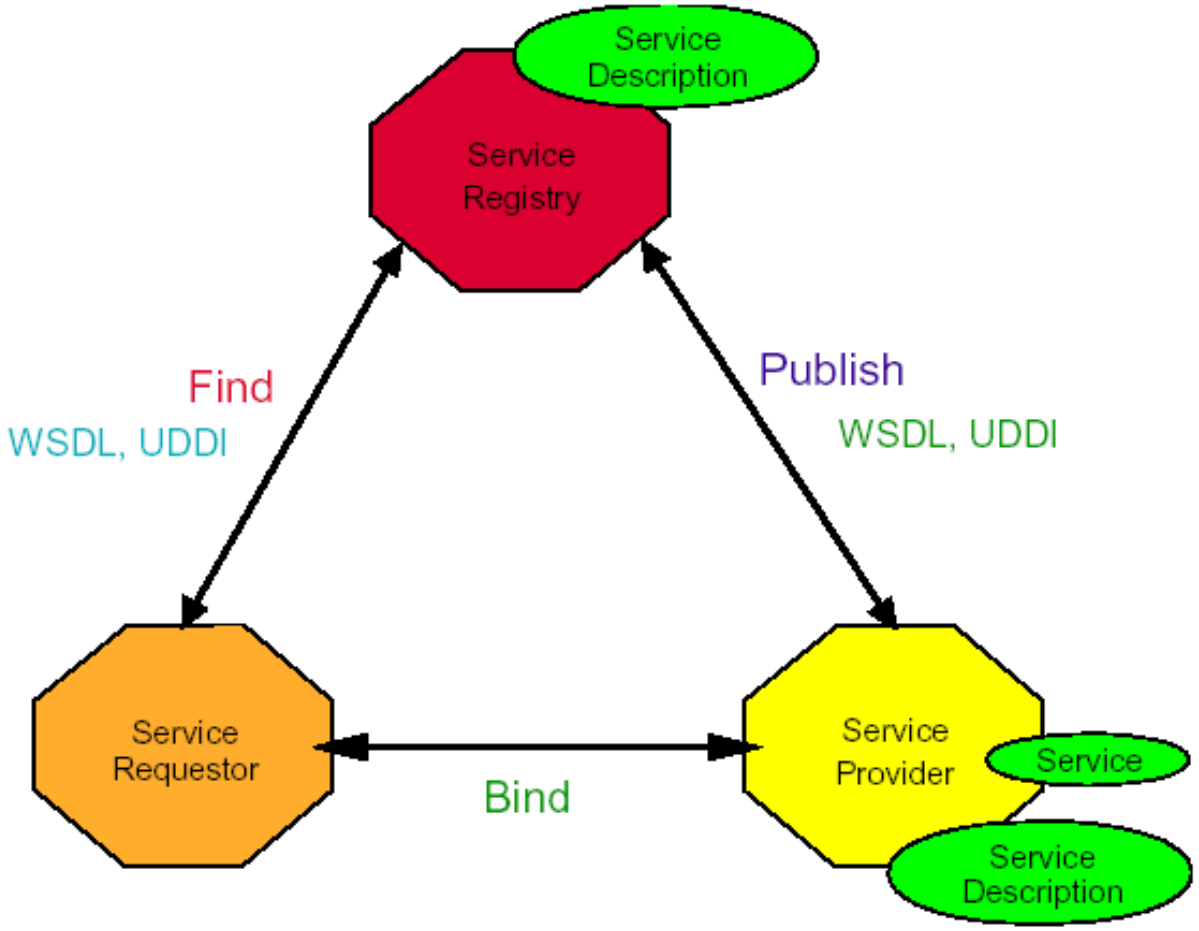
IBM has created a model to show Web services interactions which is referred to as a Service-Oriented Architecture (SOA) consisting of relationships between three entities:

- A service provider;
- A service requestor;
- A service broker

IBM's SOA is a generic model describing service collaboration, not just specific to Web services.

See: http://www-106.ibm.com/developerworks/webservices/

Service Oriented Architecture (SOA)



Service Oriented Architecture (SOA)

Roles in Web Service architecture

- Service provider –
  - Owner of the service
  - Platform that hosts access to the service
- Service requestor
  - Business that requires certain functions to be satisfied
  - Application looking for and invoking an interaction with a service
- Service registry
  - Searchable registry of service descriptions where service providers publish their service descriptions

Service Oriented Architecture (SOA)

Operations in a Web Service Architecture

- Publish

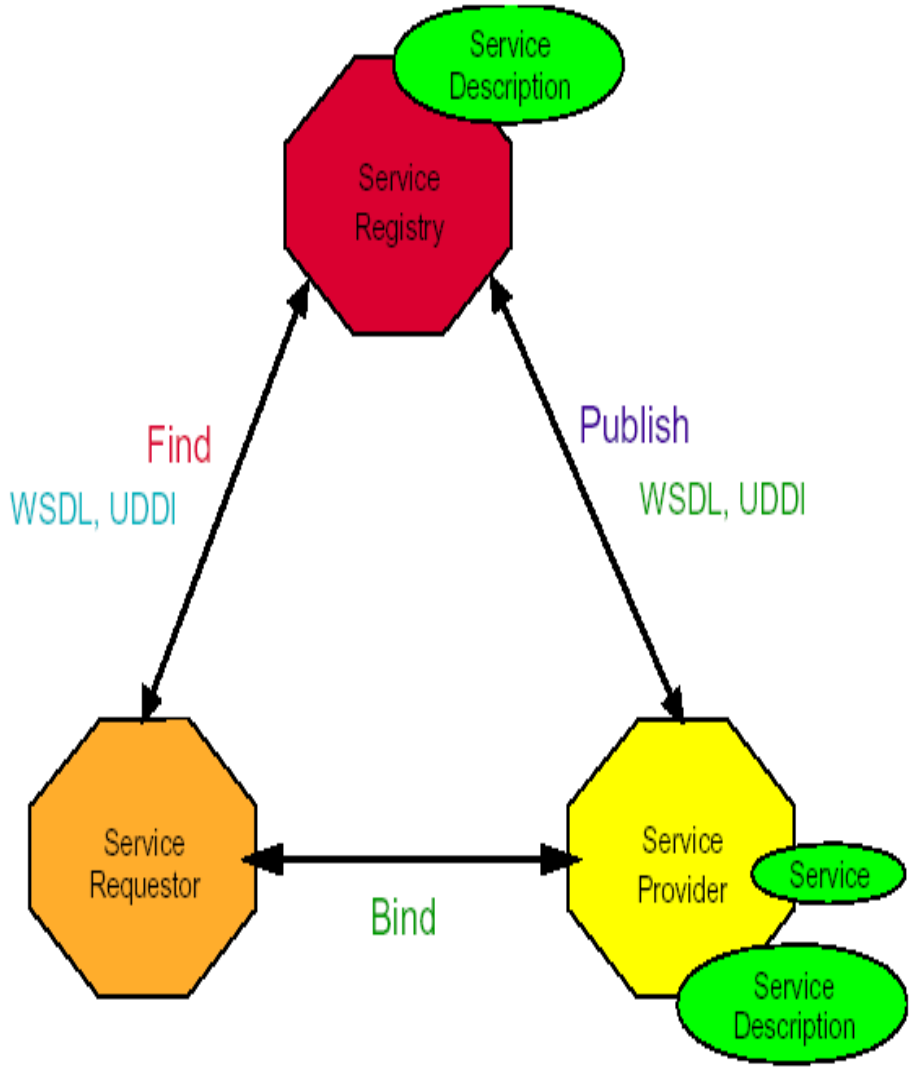
Service descriptions need to be published in order for service requestor to find them

- Find

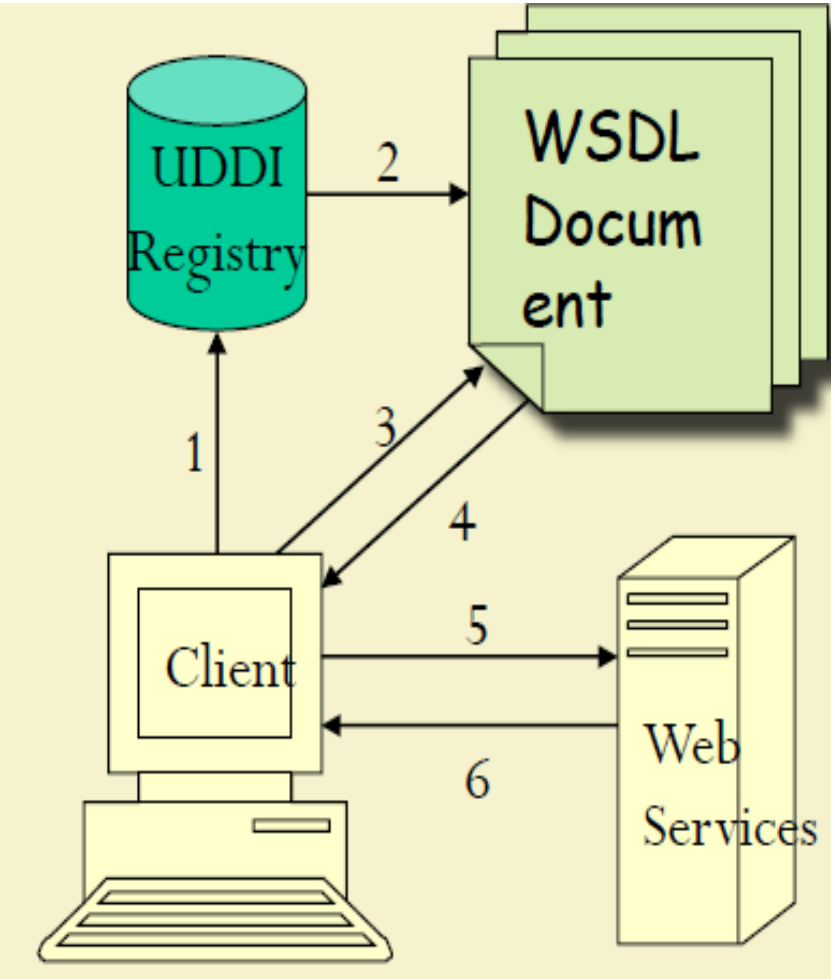
Service requestor retrieves a service description directly or queries the service registry for the service required

- Bind

Service requestor invokes or initiates an interaction with the service at runtime



Steps of Operation



1.Client queries registry to locate service.

2.Registry refers client to WSDL document.

3.Client accesses WSDL document.

4.WSDL provides data to interact with Web service.

5.Client sends SOAP-message request.

6.Web service returns SOAP-message response.

Computing platforms and technologies

- Development of a cloud computing application happens by **leveraging platforms and frameworks that provide different types of services**

Amazon Web Services (AWS)

- Amazon has been a leader in providing public cloud services (http://aws.amazon.com/).
- Amazon applies the IaaS model in providing its services.

Amazon web services (AWS)

- Amazon web services (AWS) AWS offers comprehensive cloud IaaS services ranging from virtual
  - compute,
  - storage, and
  - networking to complete computing stacks.

Amazon Web Services

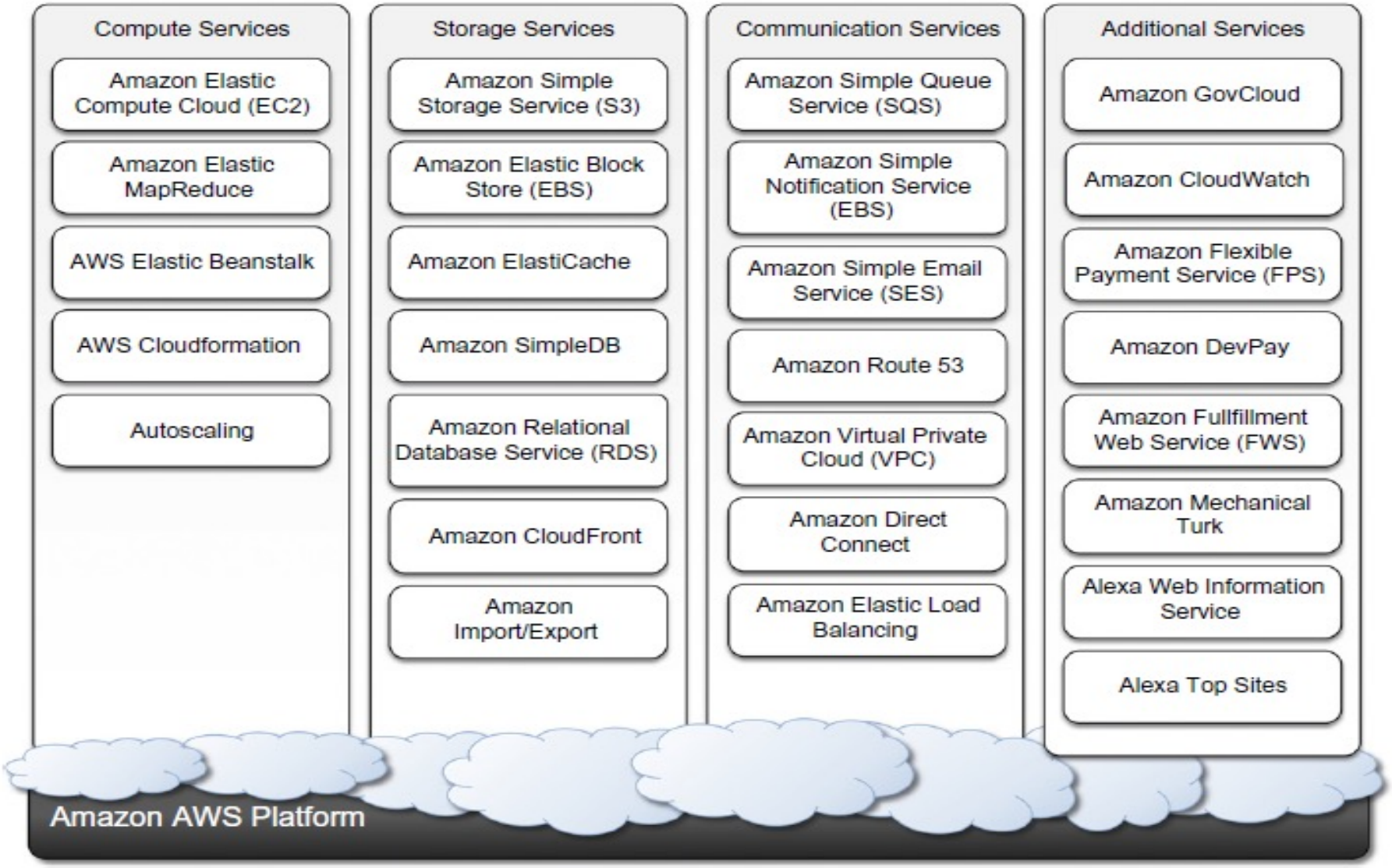


FIGURE 9.1

Amazon Web Services ecosystem.

Amazon web services (AWS)

- AWS is mostly known for its compute and storage-on demand services, namely
  - Elastic Compute Cloud (EC2) and
  - Simple Storage Service (S3)

AWS Offerings

Compute

Amazon Elastic Compute Cloud (EC2)

Amazon Elastic MapReduce

Auto Scaling

Content Delivery

Amazon CloudFront

Database

Amazon SimpleDB

Amazon Relational Database Service (RDS)

E-Commerce

Amazon Fulfillment Web Service (FWS)

Messaging

Amazon Simple Queue Service (SQS)

Amazon Simple Notification Service (SNS)

Monitoring

Amazon CloudWatch

Networking

Amazon Virtual Private Cloud (VPC)

Elastic Load Balancing

Payments & Billing

Amazon Flexible Payments Service (FPS)

Amazon DevPay

Storage

Amazon Simple Storage Service (S3)

Amazon Elastic Block Storage (EBS)

AWS Import/Export

Support

AWS Premium Support

Web Traffic

Alexa Web Information Service

Alexa Top Sites

Workforce

Amazon Mechanical Turk

AWS

- EC2 (Elastic Compute Cloud) provides the virtualized platforms to the host VMs where the cloud application can run.
- S3 (Simple Storage Service) provides the object-oriented storage service for users.
- EBS (Elastic Block Service) provides the block storage interface which can be used to support traditional applications.

Google App Engine (GAE)

- Google's App Engine (GAE) offers a PaaS platform supporting various cloud and web applications.

Google App Engine (GAE)

- Google App Engine (often referred to as GAE or simply App Engine) is a cloud computing **platform as a service for developing and hosting web applications in Google-managed data centers.**

-Definition

Google AppEngine

- AppEngine provides both a secure execution environment and a collection of services that simplify the development of scalable and high-performance Web applications.
- The languages currently supported are Python, Java, and Go.

Google App Engine (GAE)

- Google has hundreds of data centers and has installed more than 460,000 servers worldwide.
- For example, 200 Google data centers are used at one time for a number of cloud applications.

- Data items are stored in text, images, and video and are replicated to tolerate faults or failures.

- Here we discuss Google's App Engine (GAE) which offers a PaaS platform supporting various cloud and web applications.

Architecture

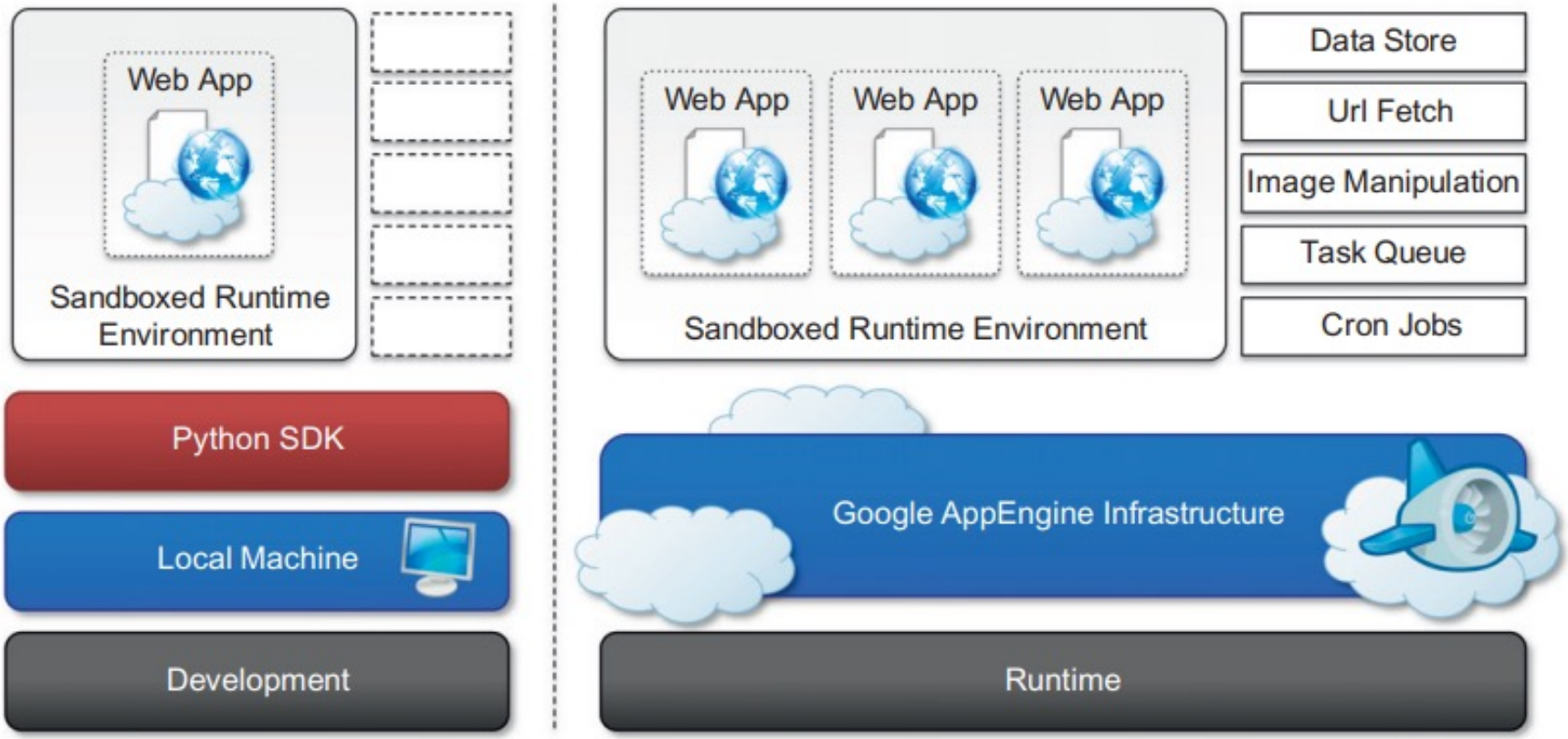


FIGURE 9.2

Google AppEngine platform architecture.

Functional Modules of GAE

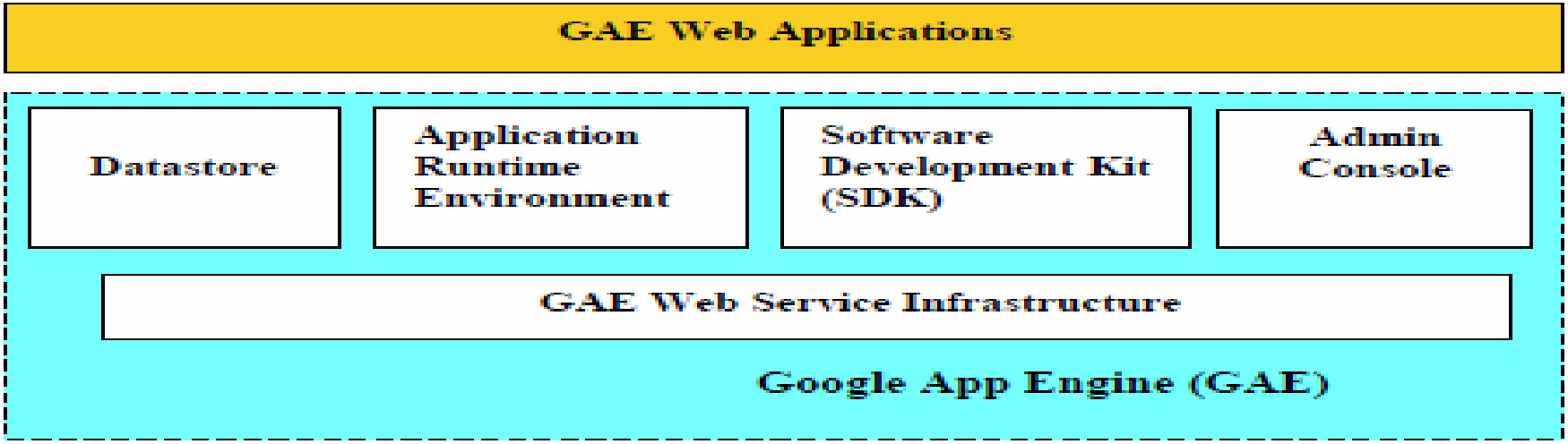


Figure 7.24 Functional components in the Google App Engine (GAE)
(Courtesy of Google, http://code.google.com/appengine/)

- The application runtime environment offers a platform for scalable web programming and execution. It supports two development languages: Python and Java.

Google AppEngine

- Developers can build and test applications on their own machines using the AppEngine software development kit (SDK),
  - which replicates the production runtime environment and
  - helps test and profile applications.
- Once development is complete, developers can easily migrate their application to AppEngine,
  - set quotas to contain the costs generated, and
  - make the application available to the world.

Microsoft Azure

- Microsoft Azure is a cloud operating system and **a platform for developing applications in the cloud.**
- It provides a scalable runtime environment for Web applications and distributed applications in general.

Microsoft Azure

- Applications in Azure are organized around the concept of roles, which identify a distribution unit for applications and embody the application's logic.
- Currently, there are three types of role:
  - **Web role,**
  - **worker role, and**
  - **virtual machine role.**

Microsoft Azure

- The Web role is designed to host a Web application
- The worker role is a more generic container of applications and can be used to perform workload processing
- The virtual machine role provides a virtual environment in which the computing stack can be fully customized, including the operating systems.

Microsoft Azure

- Besides roles, Azure provides a set of additional services that complement application execution, such as
- support for storage (relational data and blobs),
- networking,
- caching,
- content delivery, and others.

Parallel and Distributed Programming Models

- Map Reduce
- Hadoop

MapReduce

- Applied mainly in web-scale search and cloud computing applications.

- Web programming model for scalable data processing on large clusters over large data sets.

- Parallel programming abstraction
  - Used by many different parallel applications which carry out large-scale computation involving thousands of processors

MapReduce

- MapReduce is a programming model and an associated implementation for **processing and generating big data sets with a parallel, distributed algorithm on a cluster.**

- A MapReduce program is composed of **a map procedure, which performs filtering and sorting (such as sorting students by first name into queues, one queue for each name), and a reduce method, which performs a summary operation (such as counting the number of students in each queue, yielding name frequencies).**

MapReduce Model

- Two phases of MapReduce:
  - –Map operation
  - –Reduce operation

- A configurable number of M ‘mapper’ processors and R ‘reducer’ processors are assigned to work on the problem

- Computation is coordinated by a single master process

MapReduce Model Contd…

Map phase:

- –Each mapper reads approximately *1/M* of the input from the global file system, using locations given by the master

$$\text{Map:} \quad (k_1, v_1) \rightarrow [(k_2, v_2)].$$

- –Map operation consists of transforming one set of key-value pairs to another:

MapReduce Model Contd…

Map phase:

For eg-

3 Mappers, 2 Reducers,

Every mapper creates Two intermediate files, one per reducer

–Files are sorted by a key and stored to the local file system

–The master keeps track of the location of these files

MapReduce Model Contd…

Reduce phase:

–The master informs the reducers where the partial computations have been stored on local files of respective mappers

–Reducers make remote procedure call requests to the mappers to fetch the files

MapReduce Model Contd…

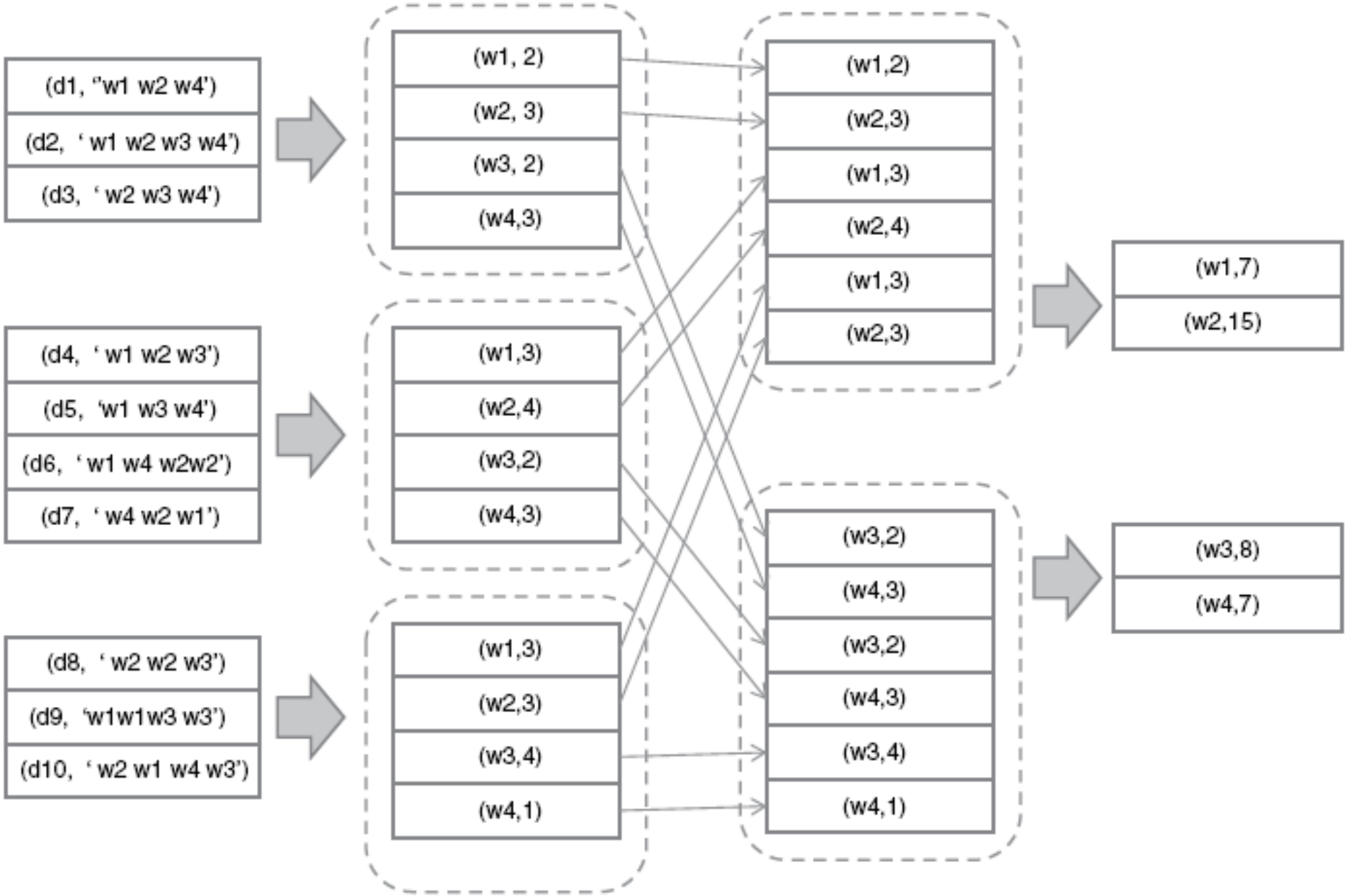
Reduce phase:

–Each reducer groups the results of the map step using the same key and performs a function *f* on the list of values that correspond to these key value:

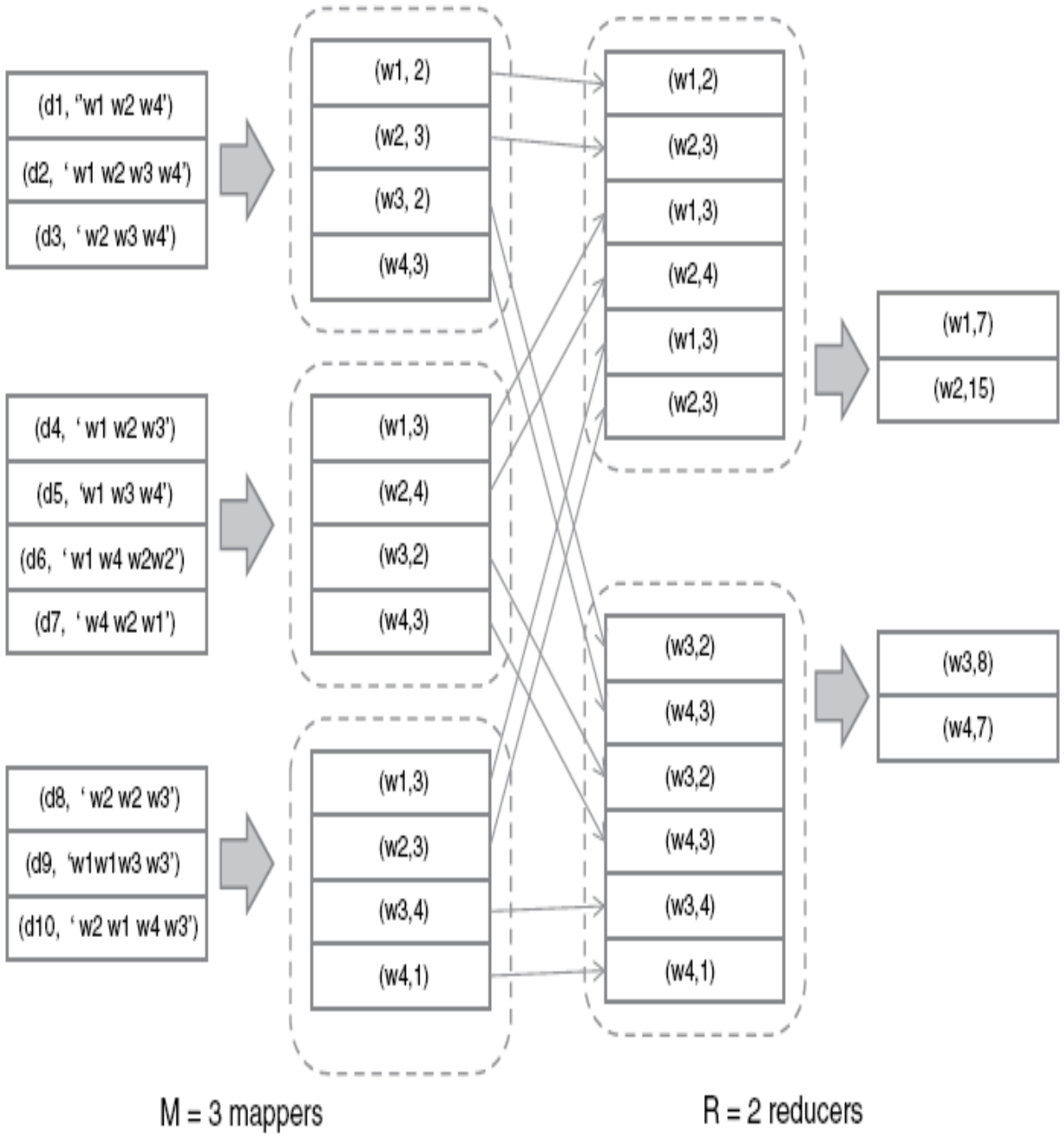
$$\text{Reduce:} \quad (k_2, [v_2]) \rightarrow (k_2, f([v_2])).$$

–Final results are written back to the file system.

MapReduce: Example??



MapReduce: Example



- 3 mappers; 2 reducers
- Map function:

$$(d_k, [w_1 \ldots w_n]) \rightarrow [(w_i, c_i)].$$

- Data d, Set of words w1,w2,w3…..wn
- For every wi it produces count of it.
- Reduce function:

$$(w_i, [c_i]) \rightarrow \left(w_i, \sum_i c_i\right)$$

Hadoop Library

- MapReduce: programming model developed at Google
- Hadoop: open source implementation of MapReduce ,developed at Yahoo!

Hadoop Library

- Hadoop provides the runtime environment, and
- developers need only provide the input data and specify the map and reduce functions that need to be executed

Hadoop Library

- Yahoo!, the sponsor of the Apache Hadoop project, has put considerable effort into transforming the project into an enterprise-ready cloud computing platform for data processing.
- Hadoop is an integral part of the Yahoo! cloud infrastructure and supports several business processes of the company.
- Currently, Yahoo! manages the largest Hadoop cluster in the world, which is also available to academic institutions

Hadoop Library

- A Software platform to write and run applications over the vast amounts of distributed data.
- Users can easily scale Hadoop to store and process petabytes of data in the web space.

Force.com

- Force.com is a cloud computing platform for developing social enterprise applications.
- PAAS soln
- PAAS platform that helps developers and business users to build powerful enterprise applications.

- Force.com allows developers to create applications
  - by **composing ready-to-use blocks;**
  - a complete set of components **supporting all the activities of an enterprise are available**
  - It is also possible to develop your own components or integrate those available in AppExchange into your applications.

SalesForce.com

- SaaS product,
- Contains in-built Out of Box (OOB) functionalities that enables a CRM system to automate your sales, marketing, services, etc.

SalesForce.com

- Force.com platform is the basis for SalesForce.com,
- SalesForce.com Is a Software-as-a-Service solution for customer relationship management.

SalesForce.com

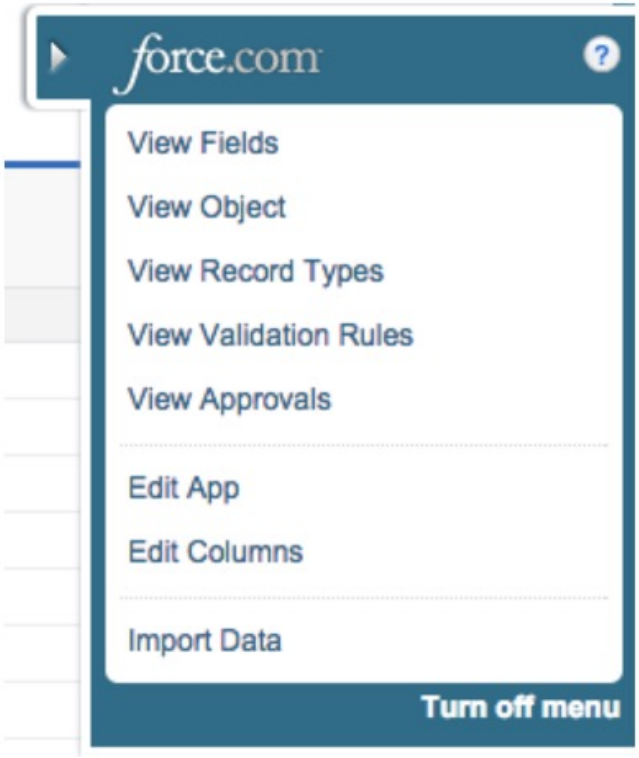
- Salesforce.com is an out of the box solution with their three core products, **Sales Cloud, Service Cloud and Marketing Cloud.**
- You could go out and buy the Sales Cloud today and starting logging leads, opportunities, running reports etc.

SalesForce.com

- In contrast, you can go out today and buy some Force.com licenses and you would not have an out of the box solution, no access to leads, accounts or opportunities.

- You would have to build Apps, Tabs, Custom Objects, Workflow Rules all yourself to design the kind of system that you want to use.

- So when you are building Custom Objects and adding in Workflows, Approval Processes etc, this is all using Force.com.



Manjrasoft Aneka

- Aneka is a platform and a framework for **developing distributed applications on the Cloud.**
- **It harnesses the spare CPU cycles of a heterogeneous network of desktop PCs and servers or datacenters on demand.**
- Aneka provides developers with a rich set of APIs for transparently exploiting such resources and expressing the business logic of applications by using the preferred programming abstractions.

Manjrasoft Aneka

- A cloud application platform for
  - rapid creation of scalable applications and
  - their deployment on various types of clouds in a seamless and elastic manner.

- **PAAS soln**

- Manjrasoft Aneka is a **.NET-based platform and framework designed for building and deploying distributed applications on clouds.**
- Aneka is also a market-oriented cloud platform since it **allows users to build and schedule applications, provision resources, and monitor results using pricing, accounting, and QoS/SLA services in private and/or public cloud environments.**

Manjrasoft Aneka

- Developers can choose different abstractions to design their application:
  - tasks,
  - distributed threads, and
  - map-reduce
- These applications are then executed on the distributed service-oriented runtime environment, which can dynamically integrate additional resource on demand