Distributed & cloud computing: **Lecture 1**

Chokchai "Box" Leangsuksun
SWECO Endowed Professor,
Computer Science
Louisiana Tech University
box@latech.edu

CTO, PB Tech International Inc.
naibox@gmail.com

Class Info

- Class Hours: noon-1:50 pm T-Th
- Dr. Box's & Contact Info:
  - www.latech.edu/~box
  - Phone 318-257-3291
  - Email: naibox@gmail.com

Class Info

- Main Text:
  - 1) Distributed Systems: Concepts and Design By Coulouris, Dollimore, Kindberg and Blair Edition 5, © Addison-Wesley 2012
  - 2) related publications & online literatures

Class Info

- Objective:
  - the theory, design and implementation of distributed systems
  - Discussion on abstractions, Concepts and current systems/applications
  - Best practice on Research on DS & cloud computing

Course Contents

- The Characterization of distributed computing and cloud computing.
- System Models.
- Networking and inter-process communication.
- OS supports and Virtualization.
- RAS, Performance & Reliability Modeling.

Security.

Course Contents

- Introduction to Cloud Computing
- Various models and applications.
  - Deployment models
  - Service models (SaaS, PaaS, IaaS, Xaas)
- Public Cloud: Amazon AWS
- Private Cloud: openStack.
- Cost models (between cloud vs host your own)

-

Course Contents

case studies

- Introduction to HPC
- Multicore & openMP
- Manycore, GPGPU & CUDA
- Cloud-based EKG system
- Distributed Object & Web Services (if time allows)
- Distributed File system (if time allows)
- Replication & Disaster Recovery Preparedness

Important Dates

- Apr 10: Mid Term exam
- Apr 22: term paper & presentation due
- May 15: Final exam

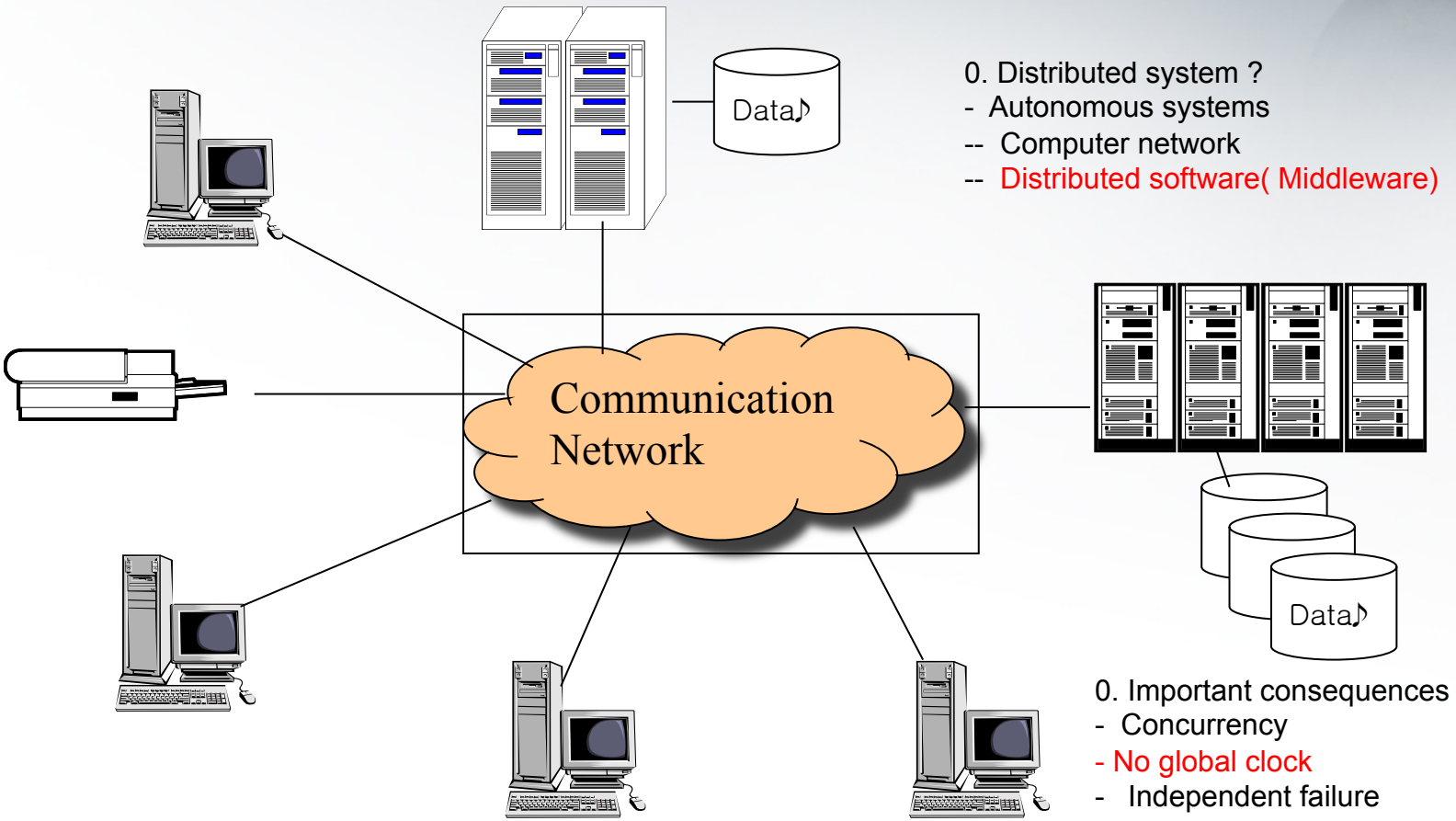
Evaluations

- 20% Lab Exercises/Quizzes
- 20% Programming Assignments
- 10% term paper
- 20% Midterm Exam
- 30% Final Exam

Intro to Distributed Computing

- Distributed System Definitions.
- Distributed Systems Examples:
  - The Internet.
  - Intranets.
  - Mobile Computing
  - Cloud Computing
- Resource Sharing.
- The World Wide Web.
- Distributed Systems Challenges.

Distributed Systems



0. Distributed system ?
- Autonomous systems
-- Computer network
-- Distributed software(Middleware)

0. Important consequences
- Concurrency
- No global clock
- Independent failure

Credit: Dr. Su-Chong Joo, Distributed Computing & Database Lab.

Why Distributed Systems?

- Main features
  - Geographical distribution of autonomous computers
  - Communication through cable/fiber/wireless/...connections
- Distributed system ?
  - A collection of independent computers that appears to its users as a single coherent system logically(called a single view system).
  - Advantages
    - interaction, co-operation, and sharing of resources
  - Benefits
    - reduced costs, improved availability and performanceScalability, resource sharing, fault tolerance.

Credit: Dr. Su-Chong Joo, Distributed Computing & Database Lab.

Distributed System Definitions

- A system in which hardware or software components located at *networked computers* communicate and coordinate their actions only by *message passing*.
  - Concurrency of components.
  - No global clock.
  - Independent failures.

Distributed System Definitions

- A system in which hardware or software components located at *networked computers* communicate and coordinate their actions only by *message passing*.
  - Concurrency of components.
  - No global clock.
  - Independent failures.
- A collection of two or more *independent* computers which coordinate their processing through the exchange of *synchronous or asynchronous* message passing.
- A collection of *independent* computers that *appear to the users* of the system as a single computer.

Computer Networks vs. Distributed Systems

- *Computer Network*: the independent computers are explicitly visible (can be explicitly addressed).
- *Distributed System*: existence of multiple independent computers is transparent.
- However,
  - many problems in common,
  - in some sense networks or parts of them (e.g., name services) are also distributed systems, and
  - normally, every distributed system relies on services provided by a computer network.

Distributed Systems: Characteristic features

- *Functional* distribution
- *Load* distribution / balancing
- Replication of *processing power*:
- *Physical* separation:
- *Economics*:

Distributed Systems: Characteristic features

- *Functional* distribution: computers have different functional capabilities (i.e., *sharing of resources* with specific functionalities).

- *Load* distribution / balancing: assign tasks to processors such that the overall system performance is optimized.

Distributed Systems: Characteristic features

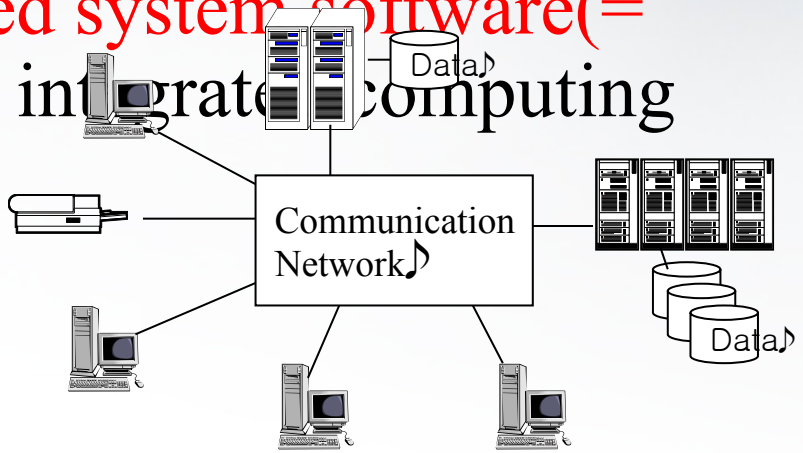
- Replication of *processing power*: independent processors working on the same task:
  - Distributed systems consisting of collections of microcomputers may have processing powers that no supercomputer will ever achieve.

- *Physical* separation: systems that rely on the fact that computers are physically separated (e.g., to satisfy reliability requirements).

- *Economics*: collections of microprocessors offer a better price/performance ration than large mainframes.
  - mainframes: 10 times faster, 1000 times as expensive

Definition of DS

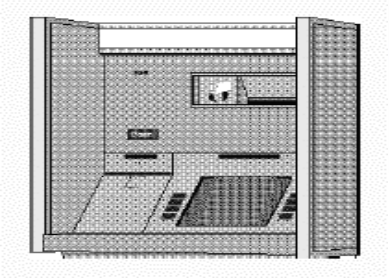
- A **distributed system** is a collection of autonomous computers interconnected by a computer network and equipped with distributed system software(= middleware) to form an integrated computing facilities.



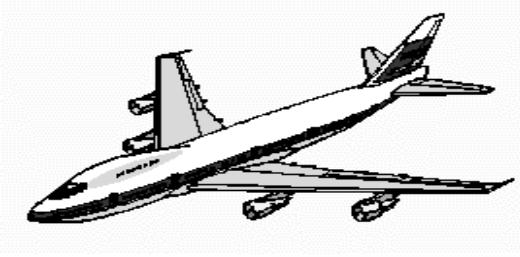
- Processes
  - execute concurrently
  - interact in order to co-operate to achieve a common goal
  - co-ordinate their activities and exchange information by means of messages transferred over a communication network

Importance of Distributed Computing

- Distributed (computer) systems are critical for functioning of many organizations



Banks



Transport

Telecommunications

- Distributed Application
  - **A set of processes** that are distributed across a network of machines and work together as an ensemble to solve a common problem

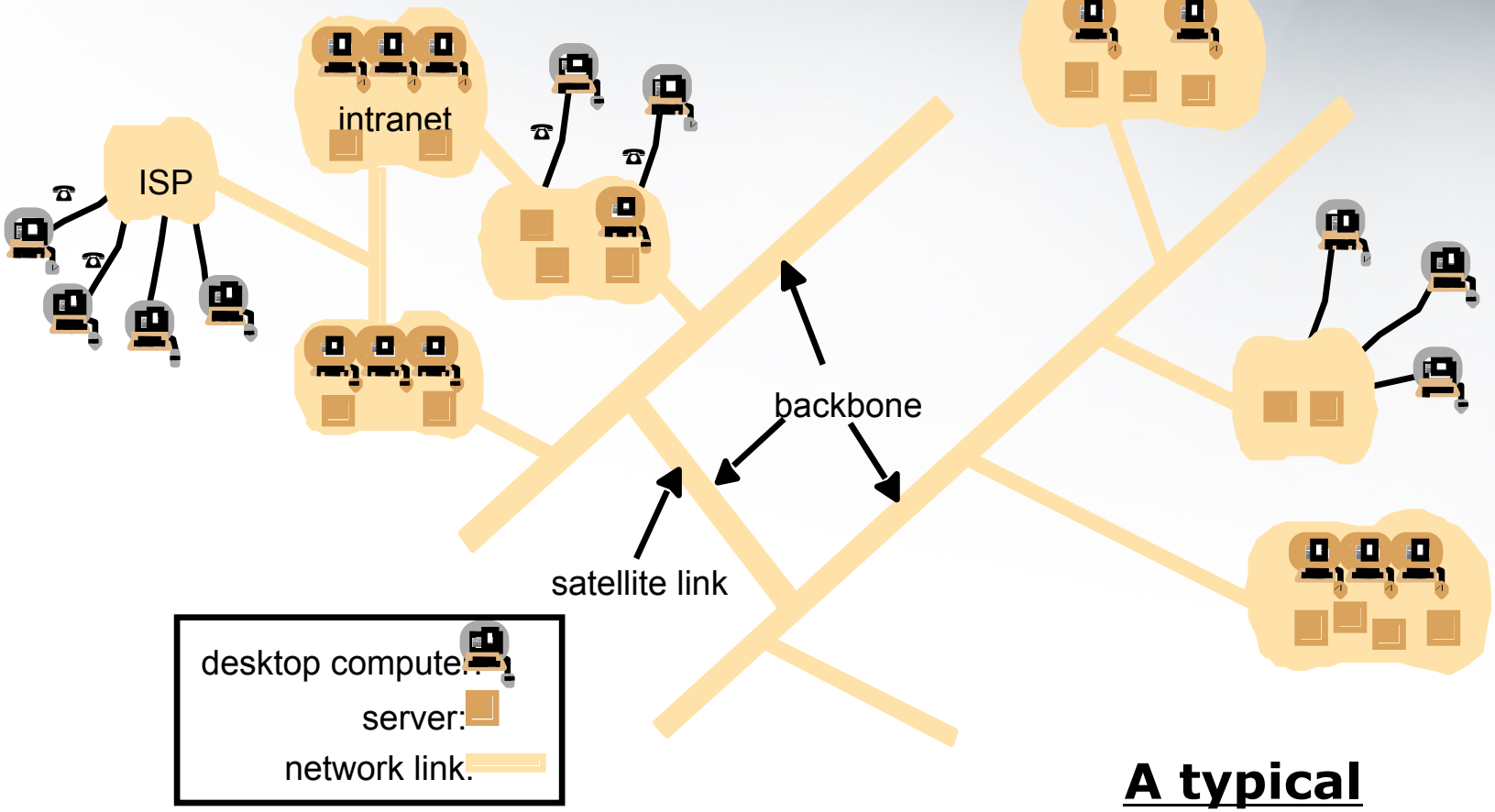
Typical examples

- **Internet**
  - global network of interconnected computers which communicate through IP protocols
- **Intranet**
  - a separately administered network with a boundary that allows to enforce local security policies
- **Mobile and ubiquitous computing**
  - laptops, PDAs, mobile phones, printers, home devices, ...
- **World-Wide Web**
  - system for publishing and accessing resources and services across the Internet
- Cloud
  - Computing offered as services, hardware, software, storage, everything??

Distributed Systems Examples
(The Internet vs intranet)

- The Internet is a vast interconnected collection of computer networks of many types.
- Its design enabling a program running anywhere to address messages to programs anywhere else.
- Allowing its users to make use of many services as: WWW, E-Mail, Web hosting, and File transfer.
- Its services can be extended by adding new types of service (*open-ended services*).
- Small organizations and individual users can to access internet services through *Internet Service Providers* (ISPs).
- Independent intranets are linked together by high transmission capacity circuits called *backbones*.

Distributed Systems Examples
(The Internet)

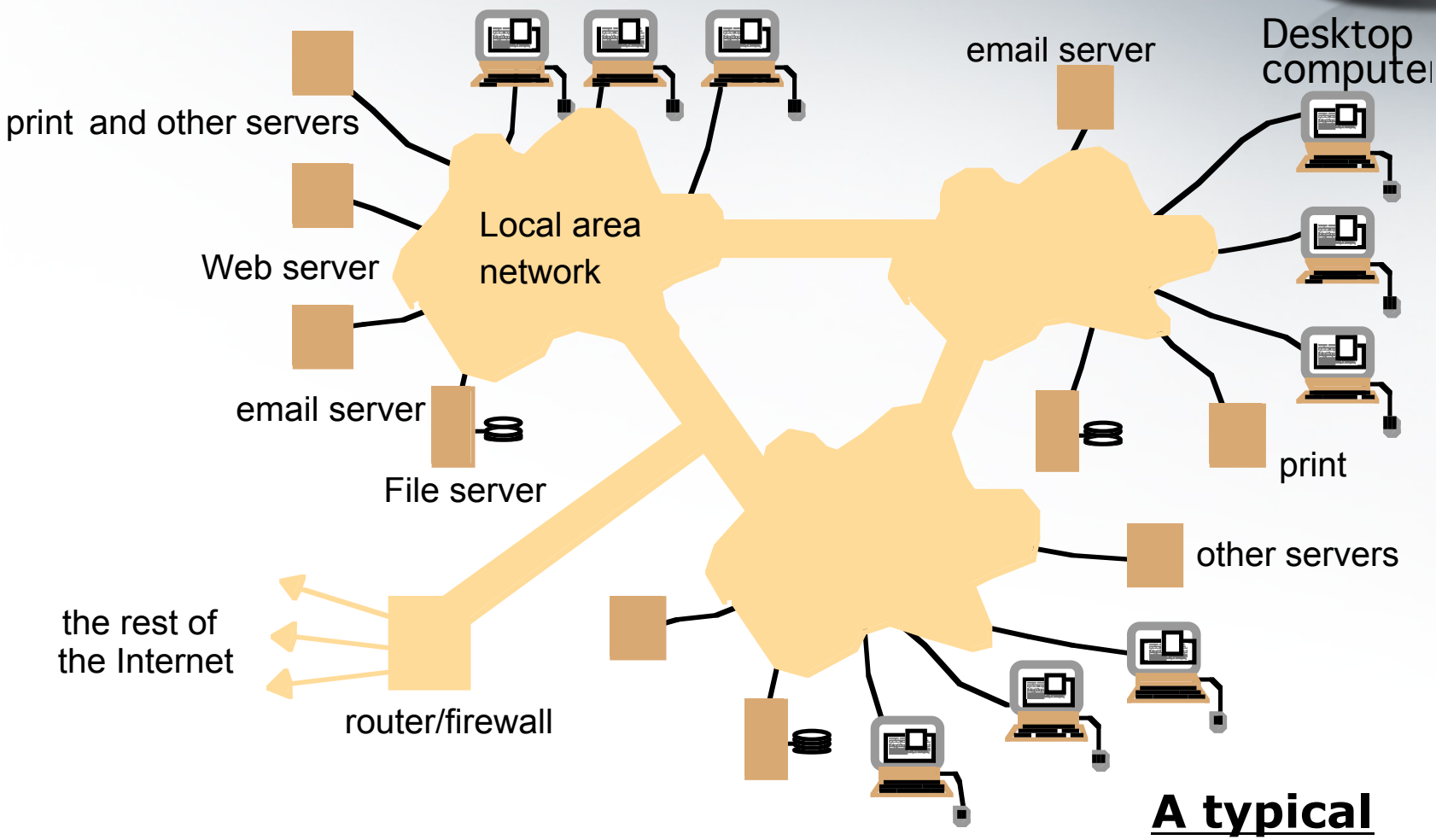


A typical portion of the Internet

Distributed Systems Examples (Intranets)

- An Intranet is a portion of the internet that is administrated separately and its local security policies are enforced by a configured boundary.
- Composed of several local area networks (LANs) linked by backbone connections to allow its users to access the provided services.
- Connected to the Internet via a *router* which allows its users to make use of the internet services elsewhere.
- Many organization protect their own services from unauthorized use by filtering incoming and outgoing messages using a *firewall*

Distributed Systems Examples (Intranets)

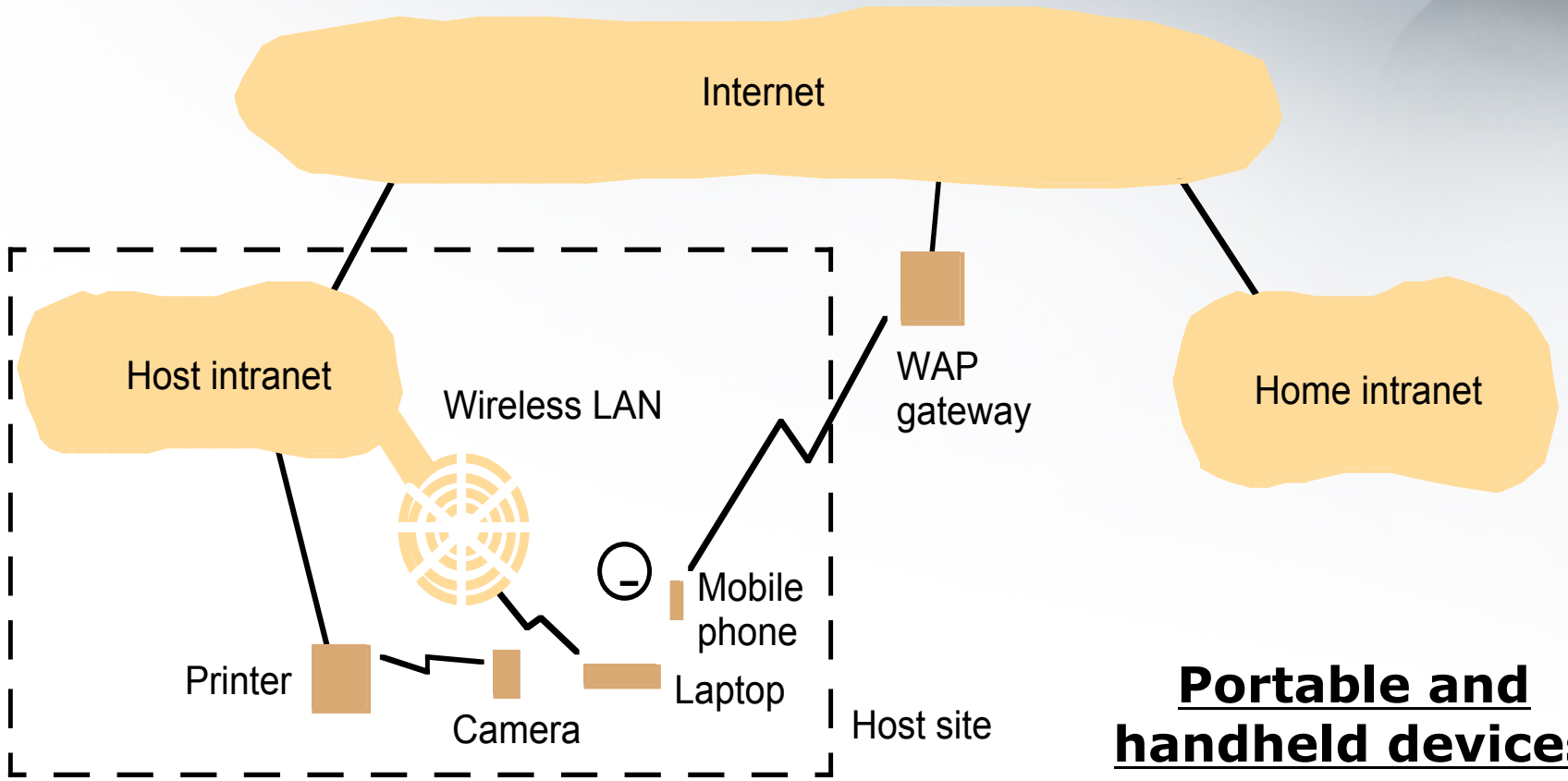


A typical Intranet

Distributed Systems Examples (Mobile Computing)

- The portability of many computing devices and the ability to connect to networks in different places makes mobile computing possible.
- *Mobile computing* is the performance of computing tasks while the users are on the move and away from their residence intranet but still provided with access to resources via the devices they carry with them.
- *Ubiquitous computing* is the harnessing of many devices that are present in user's physical environments.

Distributed Systems Examples
(Mobile and Ubiquitous Computing)

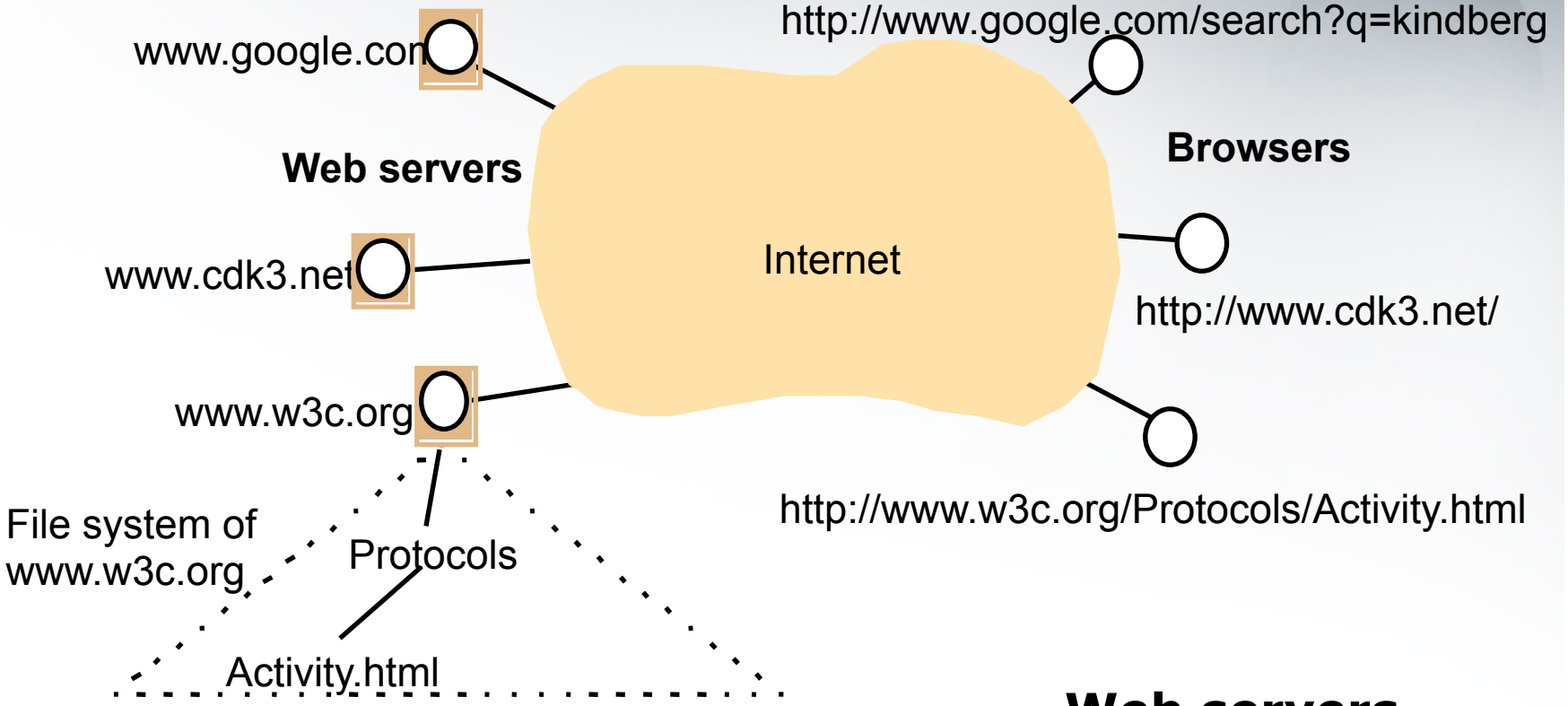


Portable and handheld devices in a distributed system

The World Wide Web (WWW)

- An executing web browser is an example of a client communicates with a web server to request web pages from it.
- WWW is an open client-server architecture implemented on top of the Internet.
- Users use the Web through available web browsers to retrieve and view documents of many types and interact with unlimited set of services.
- Web provides a *hypertext* structure among the documents that it stores. (i.e., the documents contain references, *links*, to other related documents and resources stored also in the web).

The World Wide Web (WWW)



Web servers and web browsers

WWW

- world-wide resource sharing over Internet or Intranet
- based on the following technologies:
  - HTML (HyperText Markup Language)
  - URL (Uniform Resource Locator)
  - Client-Server architecture
- Open system
  - Open-ended
  - can be extended, re-implemented, ...

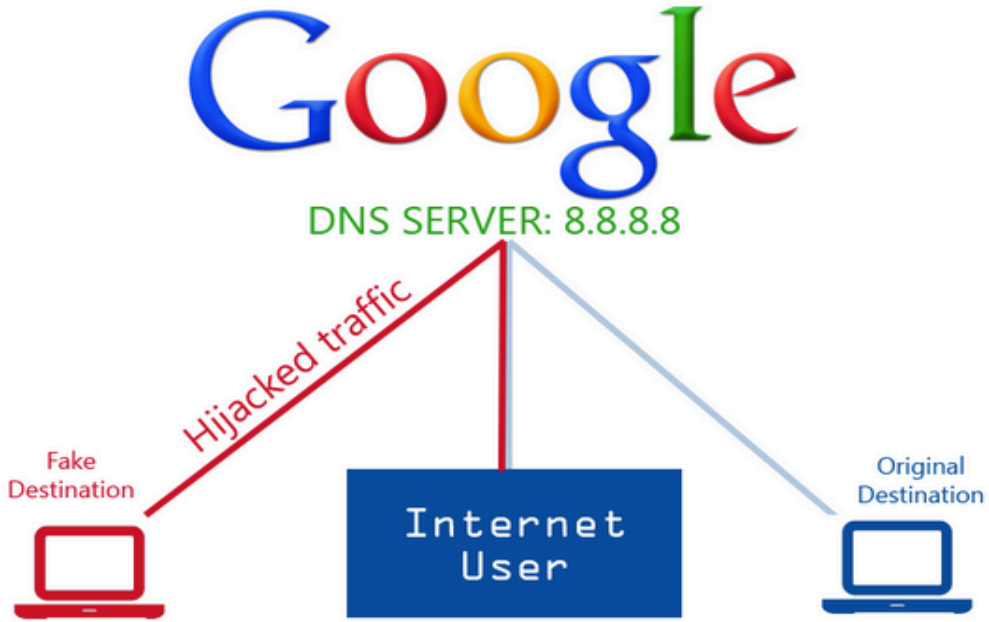
Challenges in DS

- Due to:
  - Complexity
  - Size
  - changing technologies
  - society's dependence
- Challenges posed by DSs
  - Heterogeneity
  - Openness
  - Security
  - Scalability
  - Fault handling
  - Concurrency
  - Transparency

Examples of challenges in real world

Google Public DNS Server Traffic Hijacked

Sunday, March 16, 2014 by Mohit Kumar



The Internet is becoming a dangerous place day-by-day and especially for those innocent web users who rely on 3rd party services. The latest bad news is that the World's largest and most widely used Google's free public DNS (Domain name system) resolvers raised security red flags yesterday.

DNS is the master address list for the Internet, which translates IP addresses into human readable form and vice versa. According to Internet monitoring firm *BGPmon*, Google's DNS server 8.8.8.8/32 was hijacked yesterday for 22 minutes.

The Google's DNS server handles around 150 billion queries a day and during the 22 minutes of hijacking, millions of Internet users, including Financial institutions, Governments were redirected to

Heterogeneity

- varying software and hardware
- OSs, networks, computer hardware, program languages, implementations by different developers
  - standards of protocols, middleware

- Heterogeneity and mobile code support
  - virtual machine approach (cf, Java applets)]

Openness

- independence of vendors
- publishable key interfaces
- E.g. CORBA(Common Object Request Broker Architecture)
- publishable communication mechanisms
  - E.g. Java RMI(Remote Method Invocation)

Security

- confidentiality (protect against disclosure)
  - cf, medical records
- integrity (protect against alteration and interference)
  - cf, financial data

⇒ Need encryption and knowledge of identity

1. Denial of service attacks
2. Security of mobile code

Scalability

- Design of scalable distributed systems
- Controlling the cost of physical resource(➔O(*n*), where *n* users)
  - Controlling the performance loss(➔O(*log n*), where *n* size of the set of data)
  - Preventing software resource running out(IPv4(32bits) ➔IPv6(128bits))
  - Avoiding performance bottleneck(➔DNS)

- E.g. IPv4 vs v6

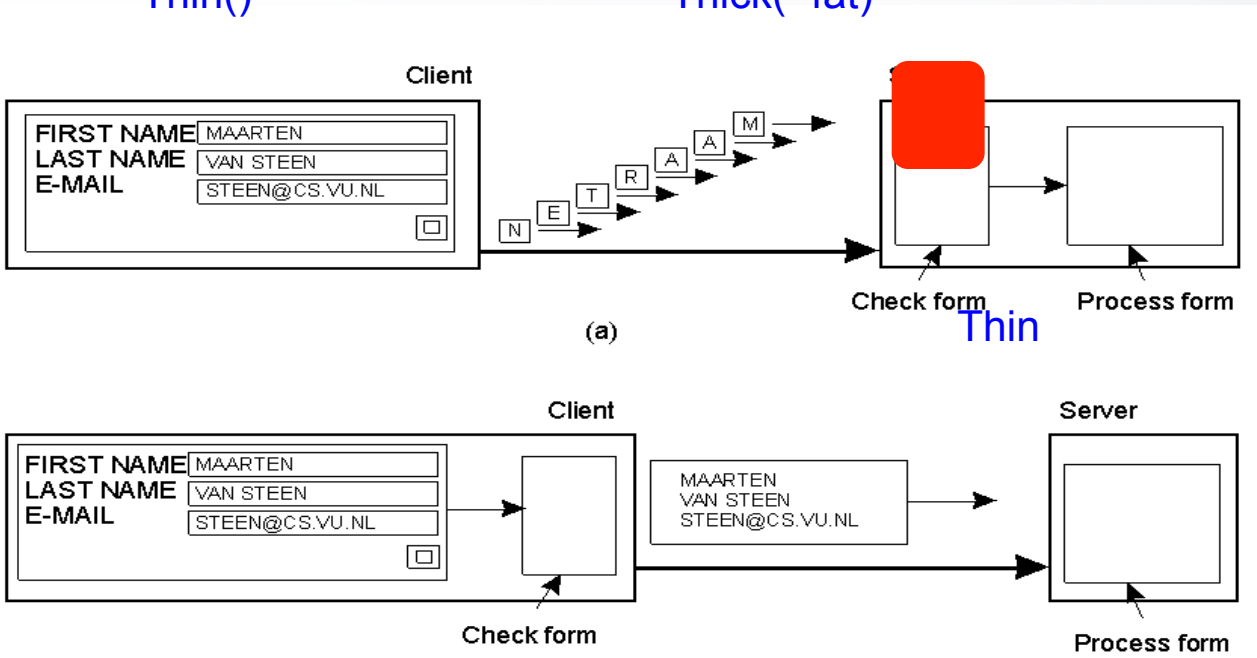
Scalability Techniques (1)

- The difference between letting: servers or clients

a) a server or

b) a client check forms as they are being filled

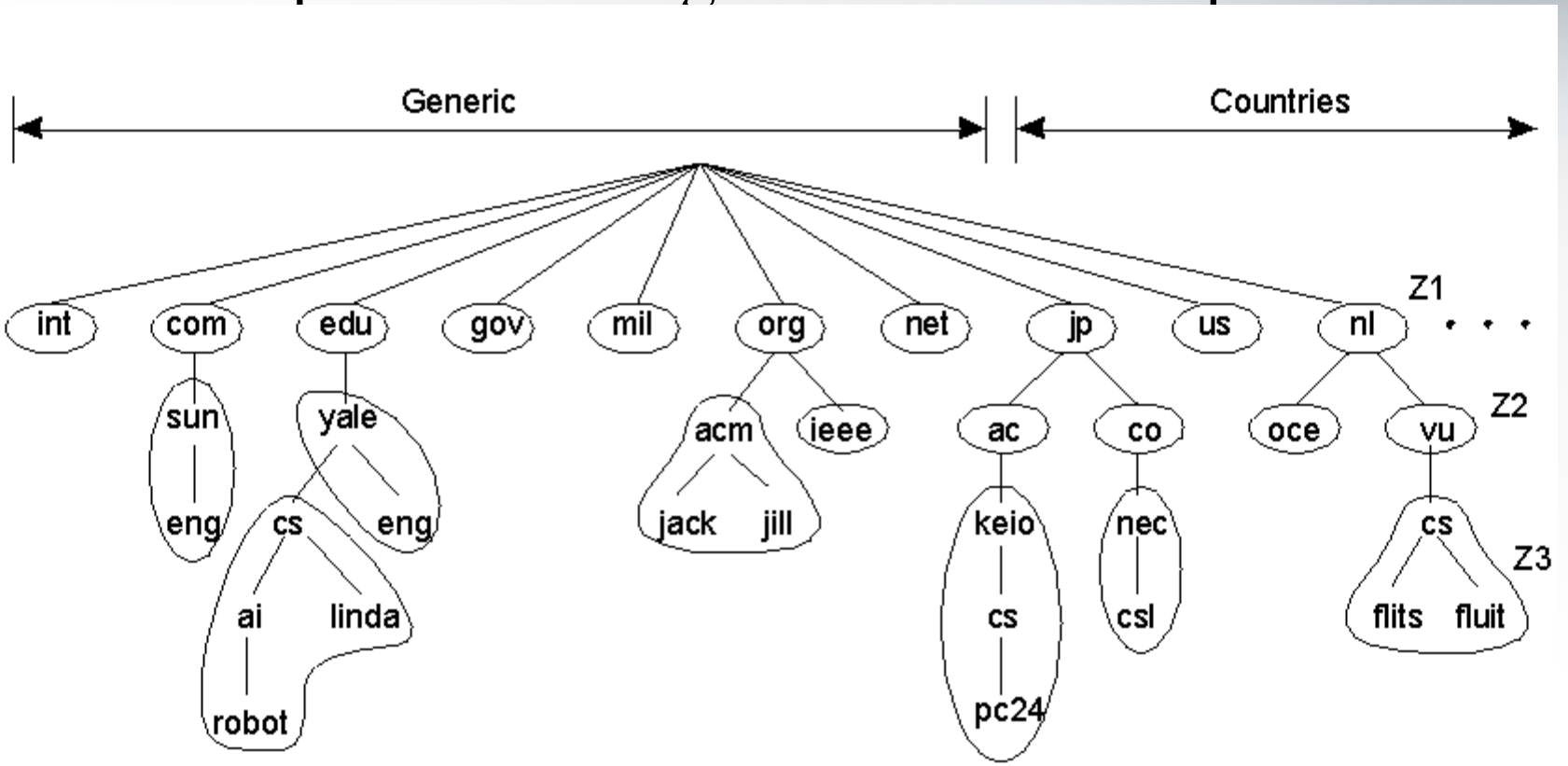
Thin() Thick(=fat)



Credit: Prof. Su-Chong Joo
scjoo@wonkwang.ac.kr
Distributed Computing & Database Lab.

Scalability Techniques (2)

- An example of dividing the DNS name space into



Failure handling

Ability to continue computation in the presence of failures.

- Detecting failures
- Masking failures(= hiding failure)
- Tolerate failures
- Recovery from failures
- Redundancy

Concurrency

Processes execute simultaneously and share resources.

- synchronization
- inter-process communication(IPC)

Transparency

Concealment of the separated nature of system from user/ programmer

=>Network transparency

- = Access transparency + Location Transparency
- cf .log on, email, …. on network

- Transparencies =>♪
  - Access transparency
  - Location transparency
  - Concurrency transparency
  - Replication transparency
  - Failure transparency
  - Mobile transparency(Migration transparency)
  - Performance transparency
  - Scaling transparency

Transparency

Access transparency: enables local and remote resources to be accessed using identical operations.

Location transparency: enables resources to be accessed without knowledge of their location.

Concurrency transparency: enables several processes to operate concurrently using shared resources without interference between them.

Replication transparency: enables multiple instances of resources to be used to increase reliability and performance without knowledge of the replicas by users or application programmers.

Failure transparency: enables the concealment of faults, allowing users and application programs to complete their tasks despite the failure of hardware or software components.

Mobility transparency: allows the movement of resources and clients within a system without affecting the operation of users or programs.

Performance transparency: allows the system to be reconfigured to improve performance as loads vary.

Scaling transparency: allows the system and applications to expand in scale without change to the system structure or the application algorithms.

Transparency (by A.S. Tanenbaum and M.V. Steen)

Transparency	Description
Access	Hide differences in data representation and how a resource is accessed
Location	Hide where a resource is located
Migration	Hide that a resource may move to another location
Relocation(mobility)	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource may be shared by several competitive users
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource
Persistence	Hide whether a (software) resource is in memory or on disk

Credit: Prof. Su-Chong Joo
scjoo@wonkwang.ac.kr
Distributed Computing & Database Lab.

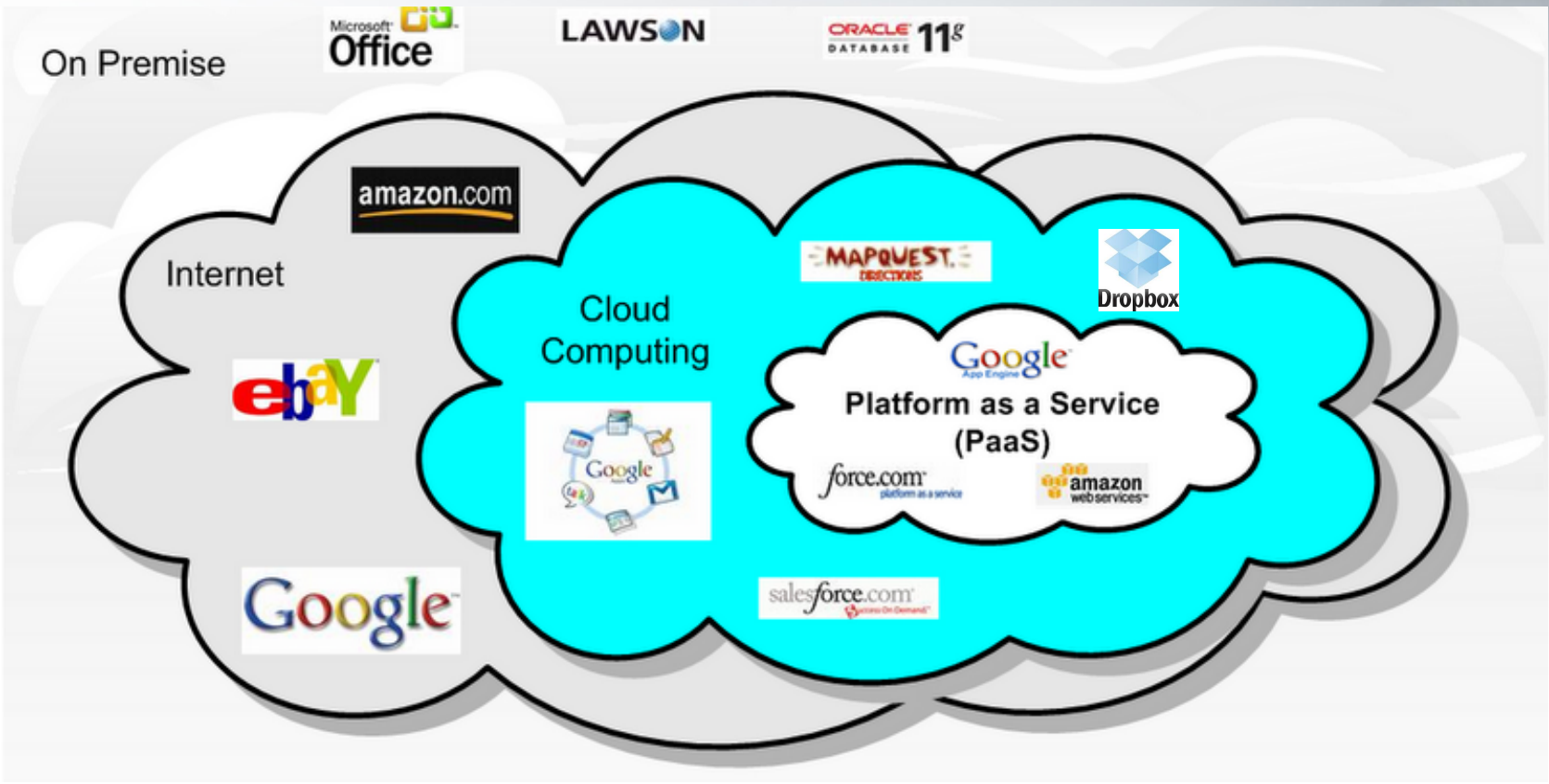
Summary

DSs

- pervasive in society
- use a variety of technologies
- understanding underlying concepts and issues important in their management, implementation, programming
- DS's challenges
  - Heterogeneity
  - Openness
  - Security
  - Scalability
  - Fault handling
  - Concurrency
  - Transparency

Cloud Computing: Distributed Systems Examples

Internet vs Cloud



What is Cloud Computing?

- - Every cloud vendor have their own definition of cloud.

- In General, Cloud computing is a Internet based (distributed) computing where hardware resources and software are exposed as a services.

What is Cloud Computing?

- These Services are exposed in a scalable manner so that the user can use those services and pay for only those services that are used.

- as on demand computing just like to get electricity we plug wire into socket.

- - According to the survey by IDC between 2008 and 2010, the main reason to adopt a cloud computing for the organization is low cost option

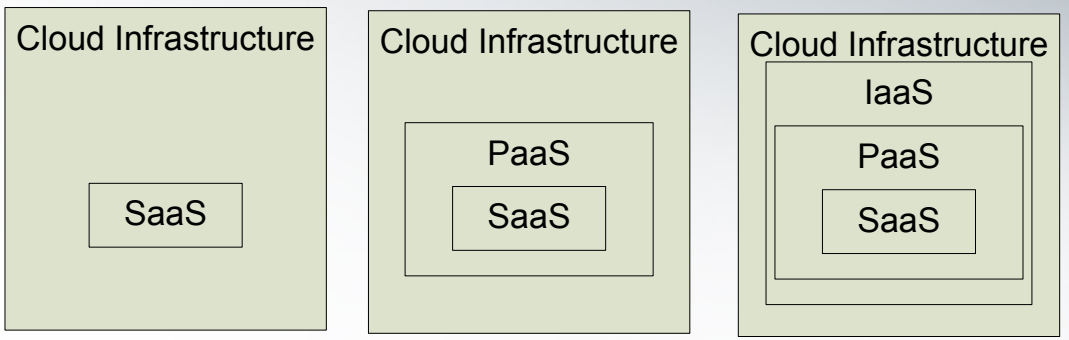
Cloud computing takes virtualization to the next step

- **You don't have to own the hardware**
- **Or consolidate many IT branches into a common share resource center(private)**
- You "rent" it as needed from a cloud
- There are public clouds
  - e.g. Amazon EC2, and now many others (Microsoft, IBM, Sun, and others ...)
- A company can create a private one
  - With more control over security, etc.

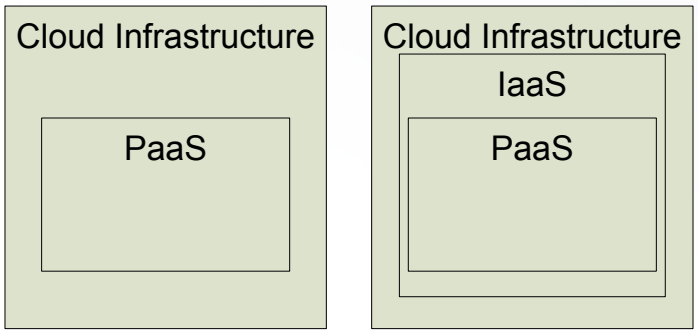
Examples of Cloud Services

- Email – instead of having a client sitting at home, you can check your e-mail wherever you are, even from other systems. (Yahoo, Gmail, Microsoft)
- Data Storage – dropbox..store MP3's, video, photos and documents online instead of at home.
- Virtualization – Amazon EC2
- Data Sharing – Google Doc's, allows quicker updates and faster project completion times
- *ASP's/SaaS, Paas

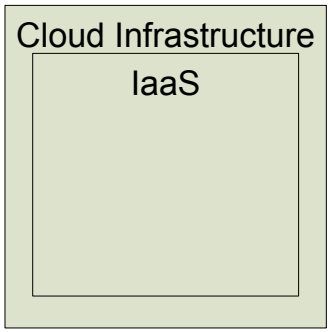
Service Model Architectures



Software as a Service (SaaS) Architectures

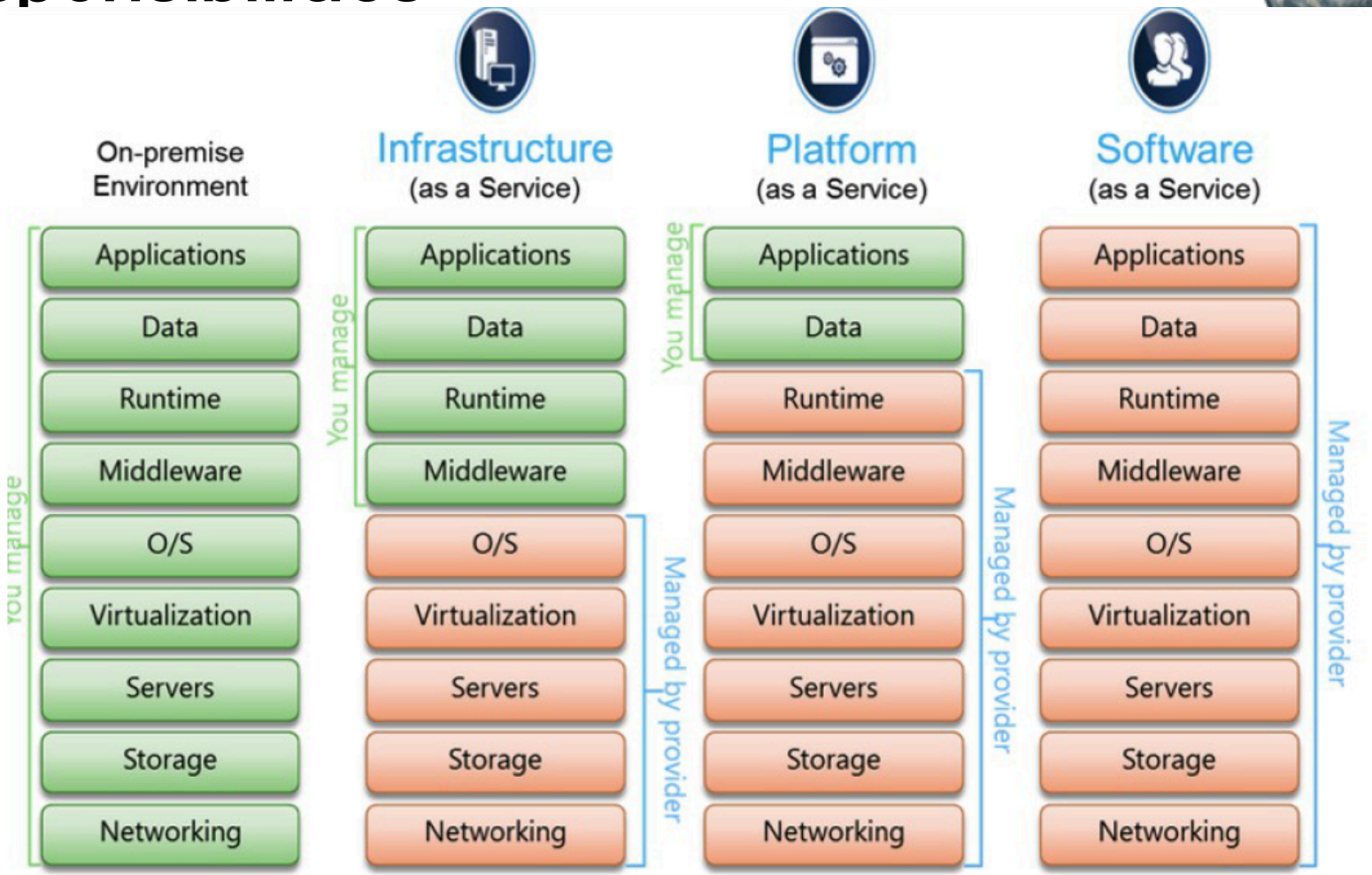


Platform as a Service (PaaS) Architectures



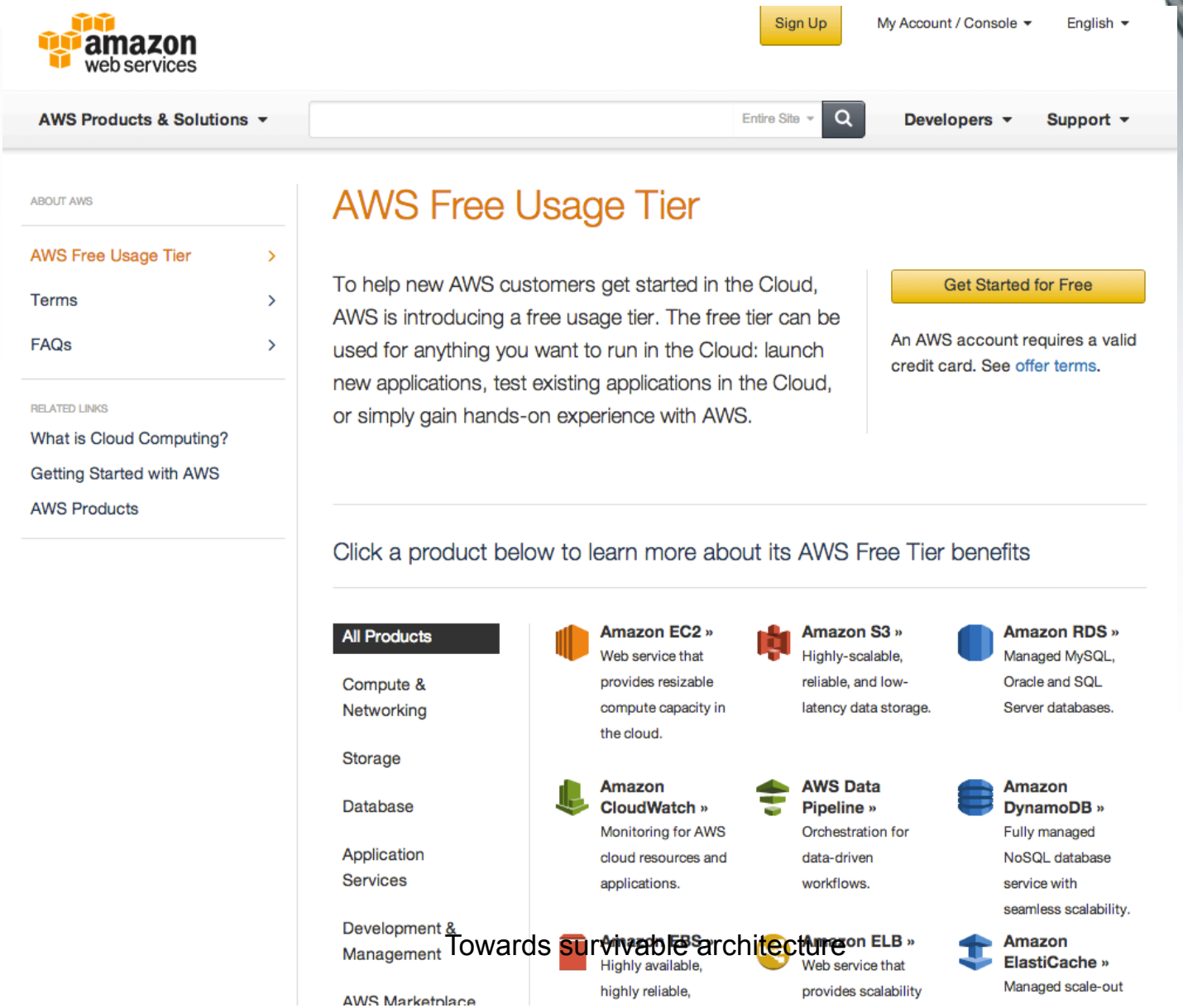
Infrastructure as a Service (IaaS) Architectures

Various service models & responsibilities

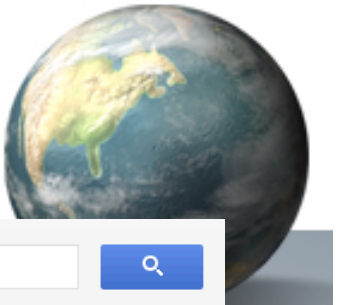


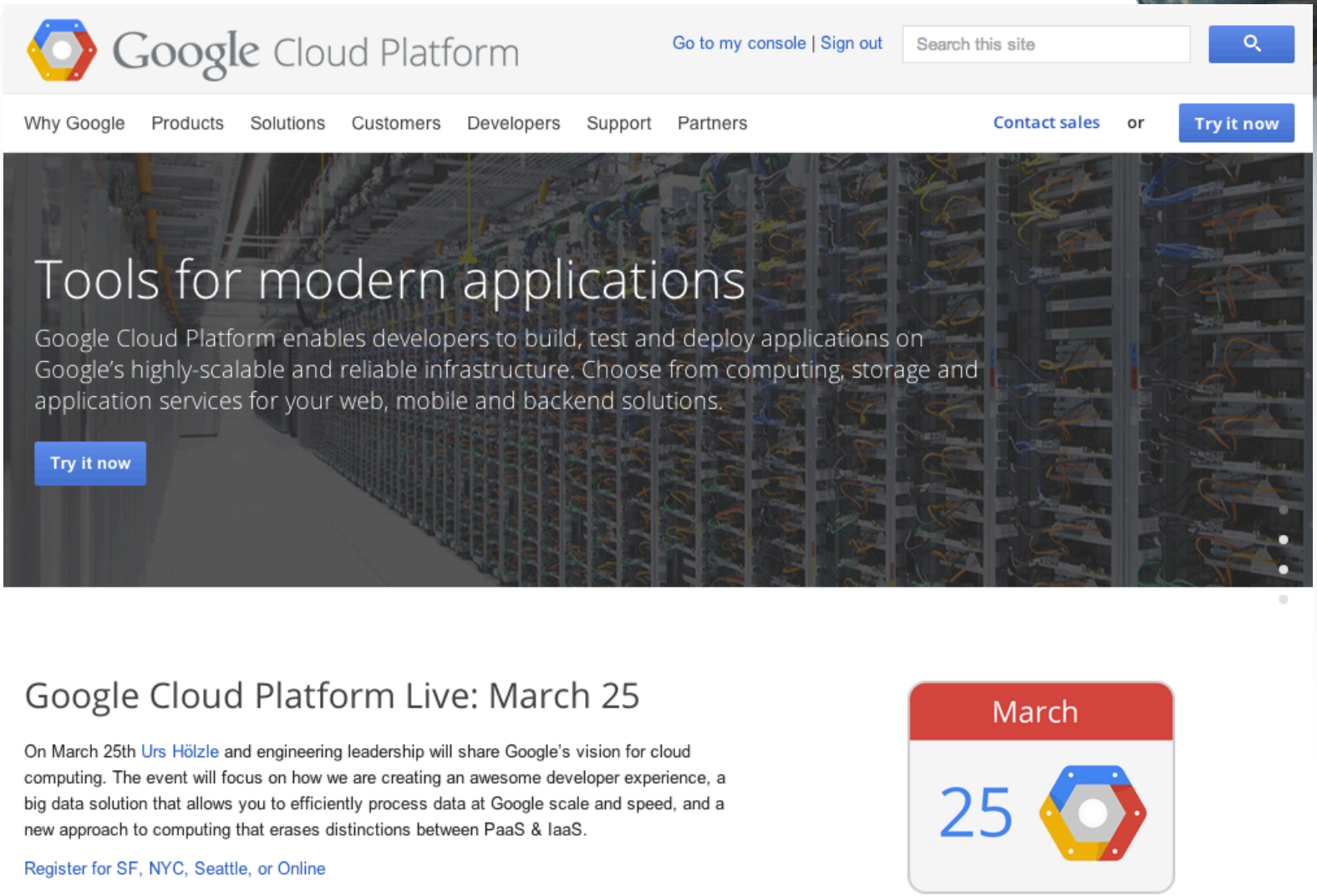
Source : http://acloudyplace.com/

Example of cloud vendors



Example of cloud vendors





Example of cloud vendors

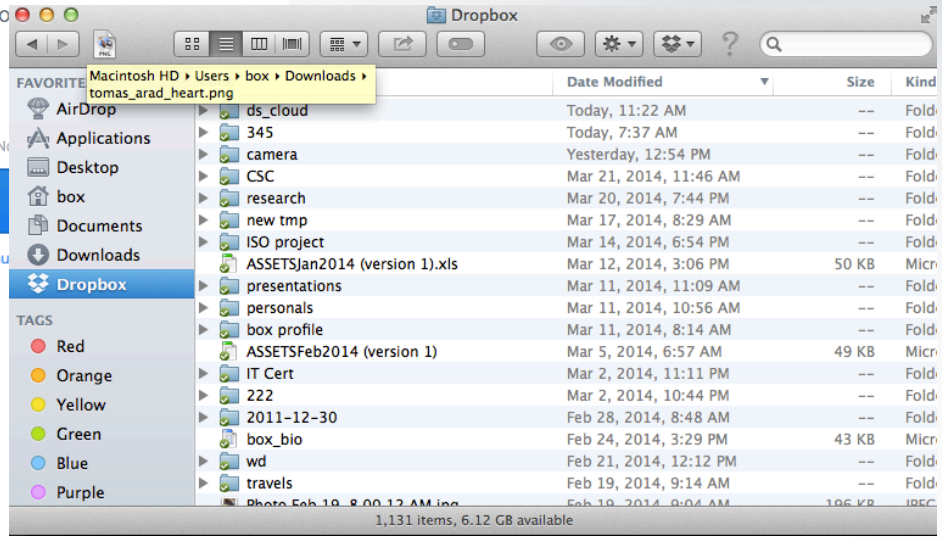
Dropbox for Business

Contact sales: Email 855-237-6726

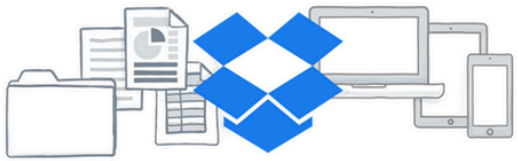
File sharing and storage made simple

More than 200 million people and 4 million businesses use Dropbox to wo

- Share with employees, vendors, and clients
- All the storage your business needs
- Access files from any device, anywhere
- Secure 256-bit AES encryption
- Admin console to manage permissions



How Dropbox for Business works



Dropbox for Business is the secure home for all your work. Anything your team adds to Dropbox will automatically sync to all their computers, mobile devices, and even the Dropbox website, so everyone can work together anywhere.

The NIST Cloud Definition Framework

Deployment Models

Hybrid Clouds

Private Cloud

Community Cloud

Public Cloud

Service Models

Software as a Service (SaaS)

Platform as a Service (PaaS)

Infrastructure as a Service (IaaS)

Essential Characteristics

- On Demand Self-Service
- Broad Network Access
- Rapid Elasticity
- Resource Pooling
- Measured Service

Common Characteristics

- Massive Scale
- Resilient Computing
- Homogeneity
- Geographic Distribution
- Virtualization
- Service Orientation
- Low Cost Software
- Advanced Security

Based upon original chart created by Alex Dowbor - http://ornot.wordpress.com

Conclusion

- Cloud computing is the promising technology where hardware resources and software are exposed as a services in a scalable manner so that the user can use those services and pay for only those services that are used.

- Cloud computing is not a silver bullet technology, we should make decision on a project by project basis and should be on the nature of the application or data that is being supported.

- Some Disadvantage such as security risks.

- Cloud computing has a potential to be a disruptive technology that may change how the IT business is done.