Cluster, Grid, Cloud & SOA

Lokeswari - AP/ CSE

SSN College of Engineering

Reference: Distributed and Cloud Computing
K. Hwang, G. Fox and J. Dongarra

Overview

- System Models for Distributed and Cloud Computing
  - Cluster and co-operative computers
  - Grid computing Infrastructures
  - Peer –to-Peer Network
  - Cloud Computing over Internet.
    - 3 Cloud Service Models
    - Cloud Computing Challenges
  - IoT
  - Service Oriented Architecture (SOA)

Distributed and Cloud Computing

- Distributed and cloud computing systems are built over large number of computer nodes
- These node machines are interconnected by SANs, LANs, or WANs.
- One can build a massive systems with millions of computers connected to network.
- Massive systems are considered highly scalable.
- Massive systems are classified as clusters, P2P networks, computational grids and internet of clouds.

System Models for Distributed and Cloud Computing

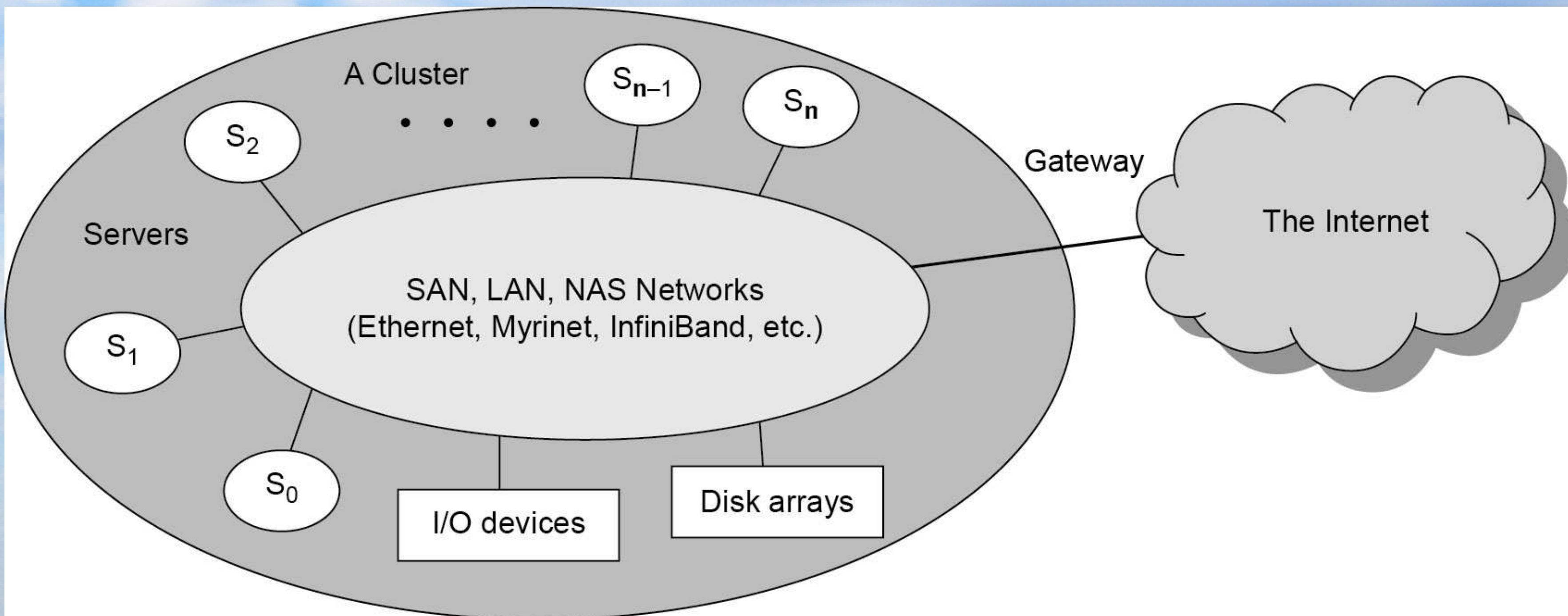
Table 1.2 Classification of Distributed Parallel Computing Systems

Functionality, Applications	Multicomputer Clusters [27, 33]	Peer-to-Peer Networks [40]	Data/Computational Grids [6, 42]	Cloud Platforms [1, 9, 12, 17, 29]
Architecture, Network Connectivity and Size	Network of compute nodes interconnected by SAN, LAN, or WAN, hierarchically	Flexible network of client machines logically connected by an overlay network	Heterogeneous clusters interconnected by high-speed network links over selected resource sites.	Virtualized cluster of servers over datacenters via service-level agreement
Control and Resources Management	Homogeneous nodes with distributed control, running Unix or Linux	Autonomous client nodes, free in and out, with distributed self-organization	Centralized control, server oriented with authenticated security, and static resources	Dynamic resource provisioning of servers, storage, and networks over massive datasets
Applications and network-centric services	High-performance computing, search engines, and web services, etc.	Most appealing to business file sharing, content delivery, and social networking	Distributed super-computing, global problem solving, and datacenter services	Upgraded web search, utility computing, and outsourced computing services
Representative Operational Systems	Google search engine, SunBlade, IBM Road Runner, Cray XT4, etc.	Gnutella, eMule, BitTorrent, Napster, KaZaA, Skype, JXTA, and .NET	TeraGrid, GriPhyN, UK EGEE, D-Grid, ChinaGrid, etc.	Google App Engine, IBM Bluecloud, Amazon Web Service(AWS), and Microsoft Azure,

Cluster of Cooperative computers

- Computing cluster consists of inter-connected stand-alone computers
- They work cooperatively as a single integrated resource.
- They produce impressive results in handling heavy workloads with large data sets.

A Typical Cluster Architecture



Cluster of Cooperative computers

- Single System Image (SSI)
  - Gref Pfister [2] : An ideal cluster should merge multiple system images into a SSI.
  - Cluster Designers desire a Cluster Operating System or some middleware to support SSI at sharing of CPU, Memory (DSM) and IO devices.
  - SSI makes a cluster to appear like a single machine to a user.
  - A cluster with multiple images is a collection of independent computers.
  - Most of clusters have Linux OS.
  - Challenge: Unfortunately, a cluster-wide OS for complete resource sharing is not yet available. Middleware is required. Otherwise cluster nodes cannot work together to produce cooperative results.

Cluster (Single System Image)

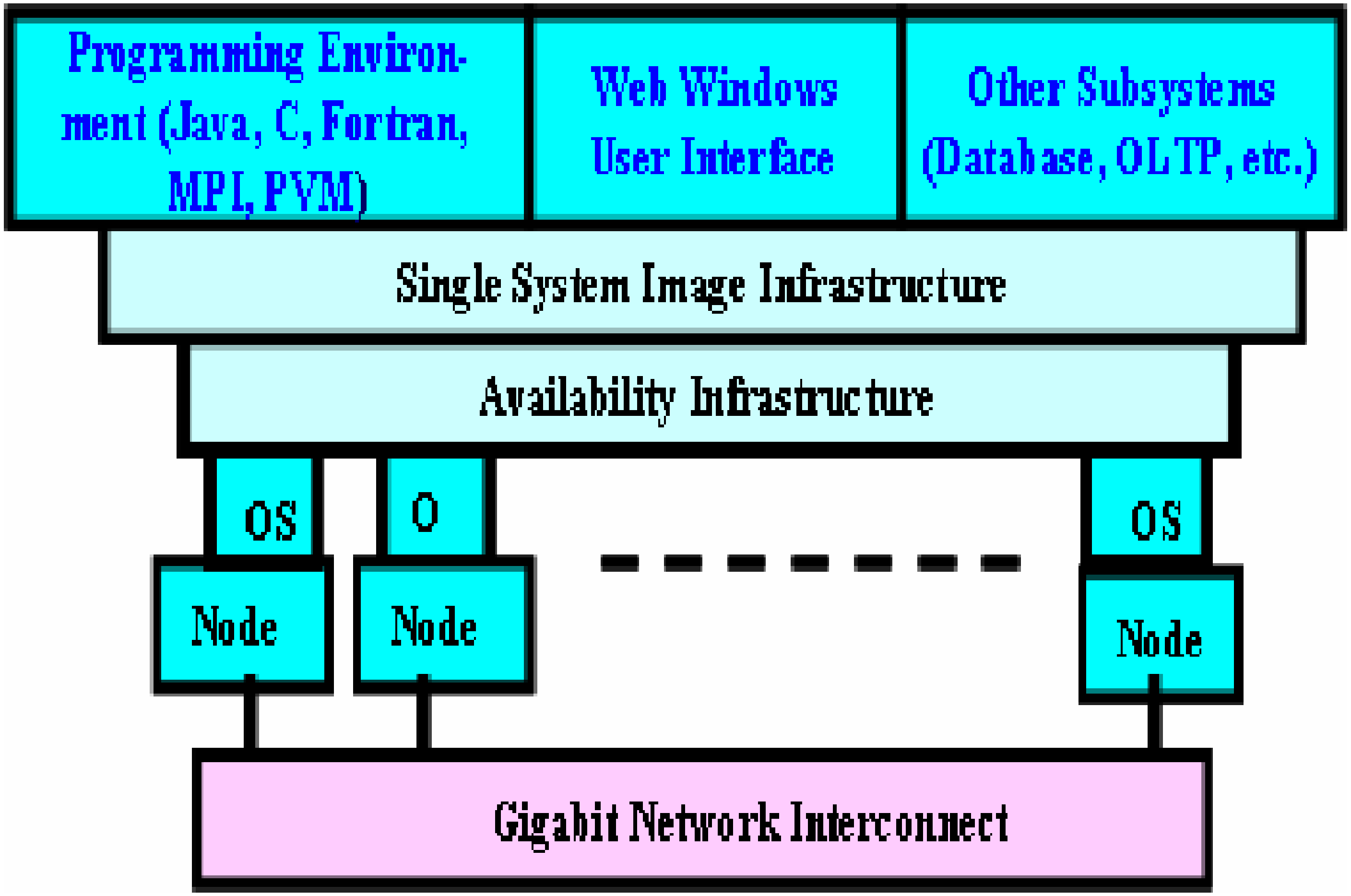


Figure 1.10 The architecture of a working cluster with full hardware, software, anAd middleware support for availability and single system image.

Cluster Features

Table 1.7 Critical Cluster Design Issues and Feasible Implementations

Features	Functional Characterization	Feasible Implementations
Availability Support	Hardware and software support for sustained high availability in cluster	Failover, failback, checkpointing, roll back recovery, non-stop OS, etc
Hardware Fault-Tolerance	Automated failure management to eliminate all single points of failure	Component redundancy, hot swapping, RAID, and multiple power supplies, etc.
Single-System Image (SSI)	Achieving SSI at functional level with hardware and software support, middleware, or OS extensions.	Hardware mechanisms or middleware support to achieve distributed shared memory (DSM) at coherent cache level.
Efficient Communications	To reduce message-passing system overhead and hide latencies	Fast message passing , active messages, enhanced MPI library, etc.
Cluster-wide Job Management	Use a global job management system with better scheduling and monitory	Apply single-job management systems such as LSF, Codine, etc
Dynamic Load Balancing	Balance the workload of all processing nodes along with failure recovery	Workload monitory, process migration, job replication and gang scheduling, etc.
Scalability and Programmability	Adding more servers to a cluster or adding more clusters to a Grid as the workload or data set increases	Use scalable interconnect, performance monitory, distributed execution environment, and better software tools

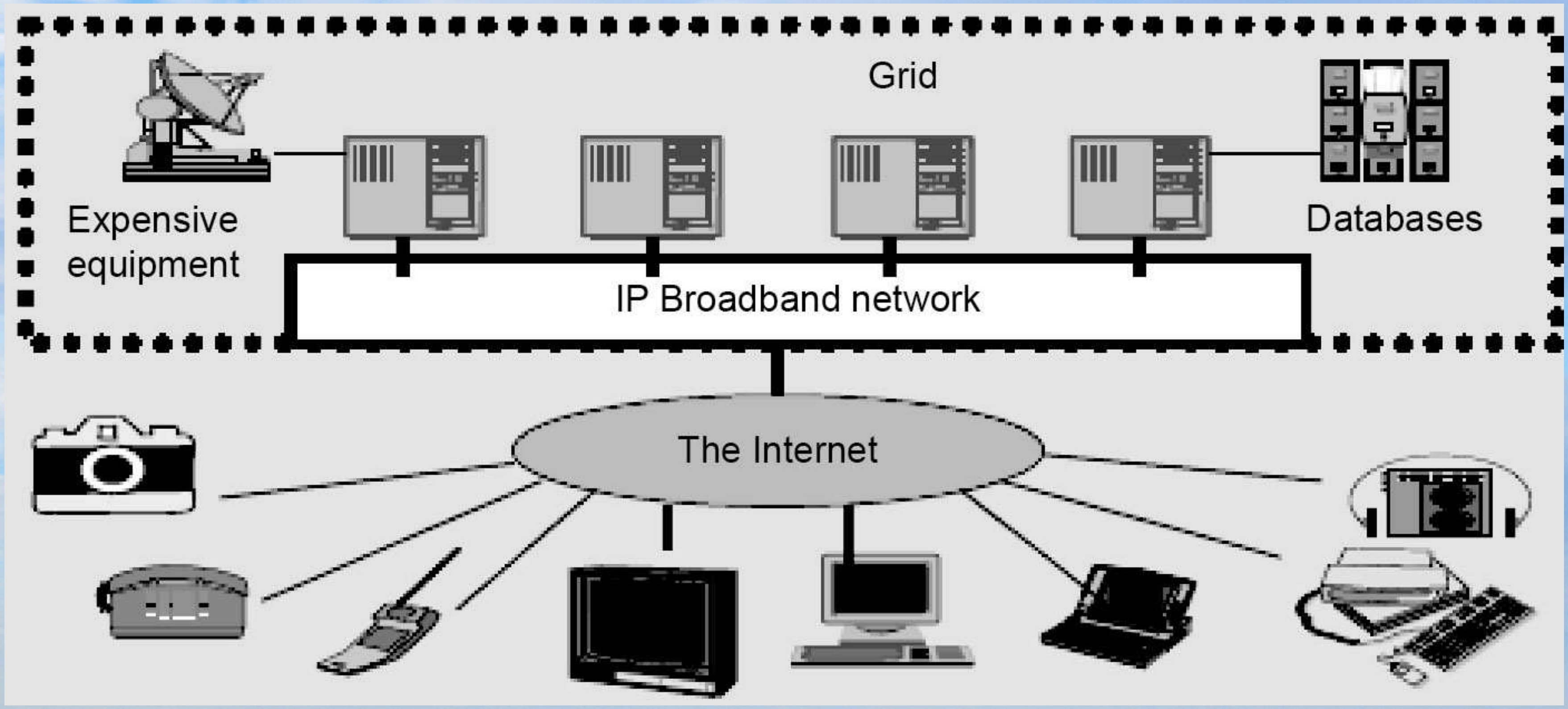
GRID Computing Infrastructure

- Internet services enables local computers to connect to remote computes.
  - Telnet commands.
  - HTTP web service.
- Grid allows close interaction among applications running on distant computers simultaneously.

Computational Grid

- A computing grid offers an infrastructure that couples computers, software, middleware, special instruments, people and sensor together, which is usually constructed across LAN, WAN, or Internet backbone.
- 2 categories: computational/data grids and P2P grids.

Computational or Data Grid



Computational or data grid providing computational utility, data and information services through resource sharing cooperation among participating organizations

A Typical Computational Grid

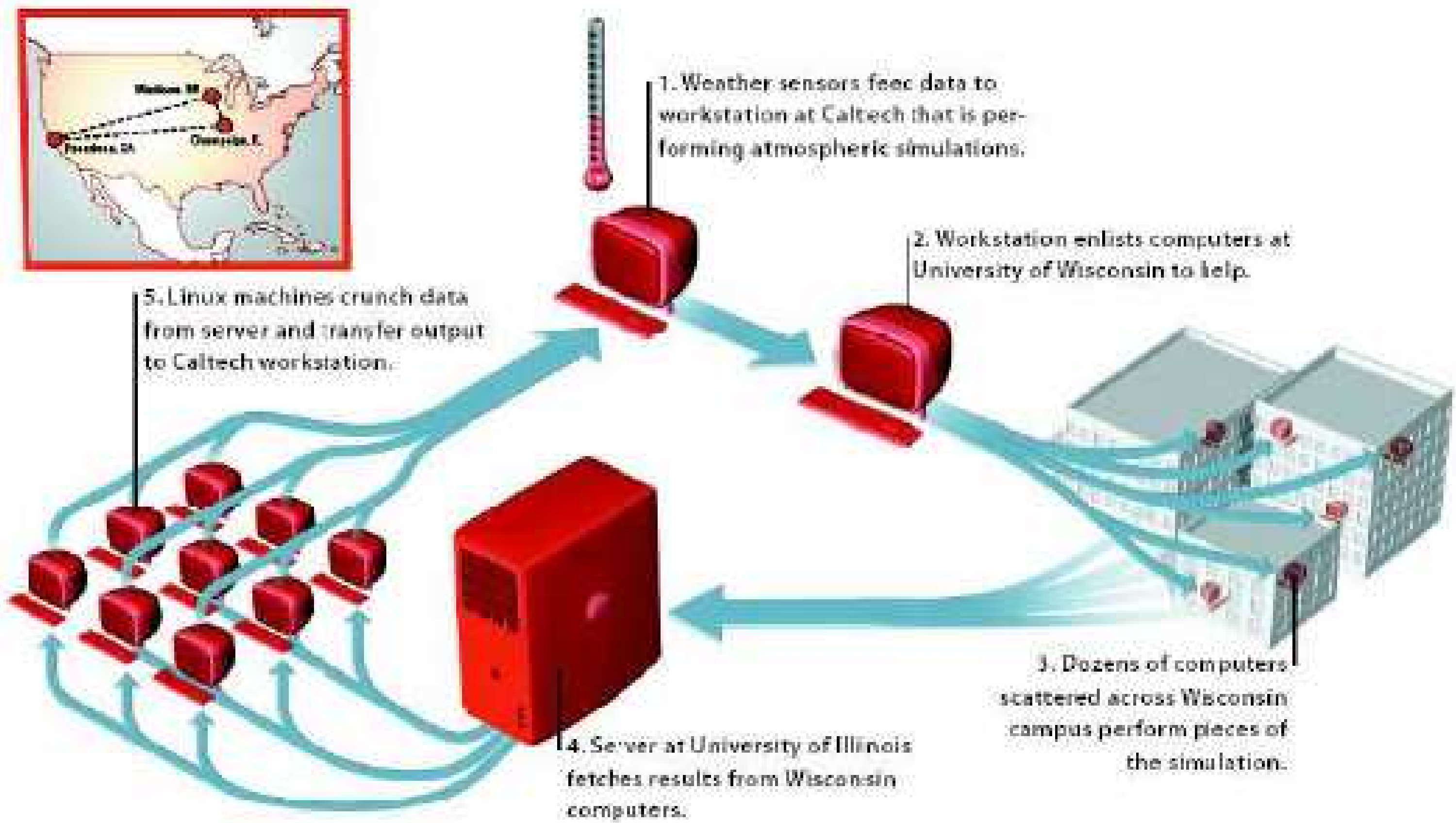


Figure 1.17 An example computational Grid built over specialized computers at three resource sites at Wisconsin, Caltech, and Illinois. (Courtesy of Michel Waldrop, "Grid Computing", IEEE *Computer Magazine*, 2000. [42])

Grid Computing Representatives

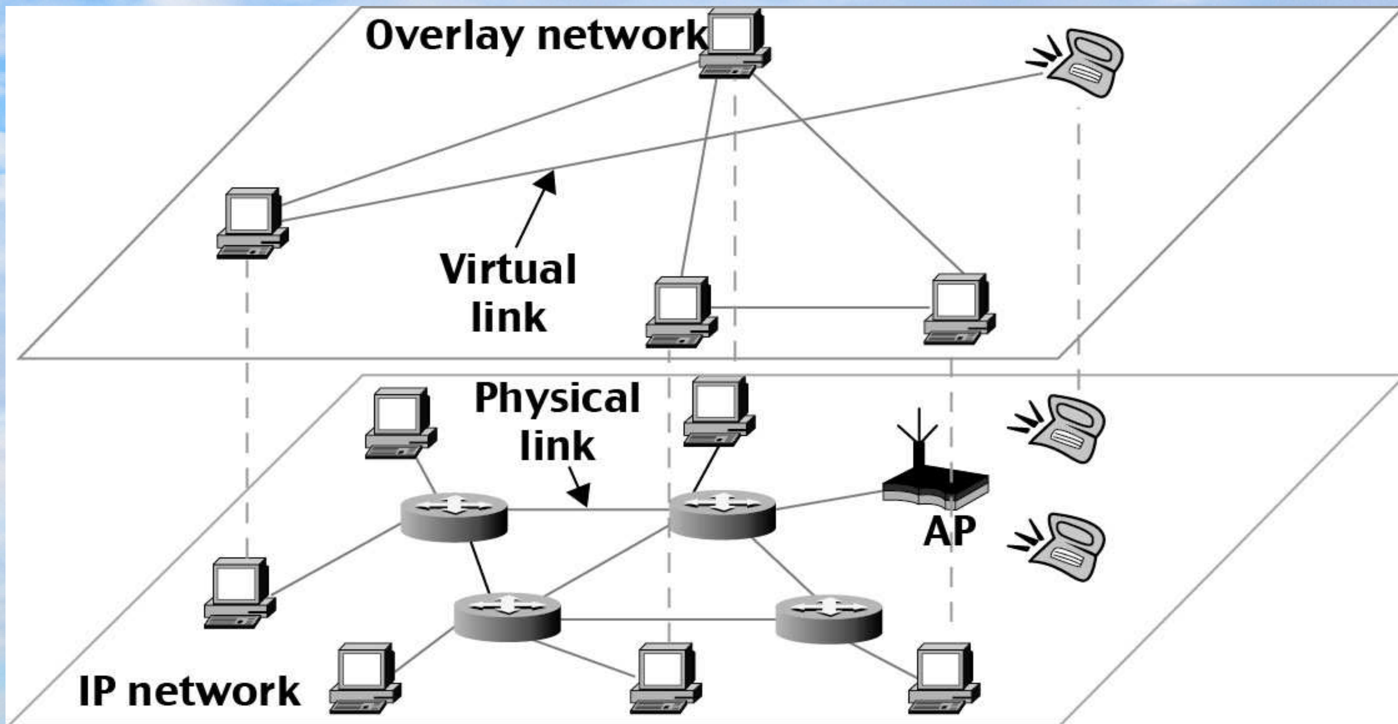
Table 1.8 Two Grid Computing Infrastructures and Representative Systems

Design Issues	Computational and Data Grids	P2P Grids
Grid Applications reported	Distributed Supercomputing, National Grid Initiatives, etc	Open grid with P2P flexibility, all resources from client machines
Representative Systems	TeraGrid in US, ChinaGrid, UK e-Science, etc.	JXTA, FightAid@home, SETI@home
Development Lessons learned	Restricted user groups, middleware bugs, rigid protocols to acquire resources	Unreliable user-contributed resources, limited to a few apps.

Peer-to-Peer (P2P) Network

- A distributed system architecture
- Each computer in the network can act as a client or server for other network computers.
- No centralized control or No central database.
- Typically many nodes, but unreliable and heterogeneous
- Physical network is a ad-hoc network formed using TCP/IP.
- Take advantage of distributed, shared resources (bandwidth, CPU, storage) on peer-nodes
- Self-organizing
- Operate in dynamic environment, frequent join and leave the system freely. (No Master- Slave relationship among peers)

Peer-to-Peer (P2P) Network



Overlay network - computer network built on top of another network.

- Nodes in the overlay can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links, in the underlying network. (Unstructured and structured Overlays)
- For example, distributed systems such as cloud computing, peer-to-peer networks, and client-server applications are overlay networks because their nodes run on top of the Internet.

P2P Virtual Mapping

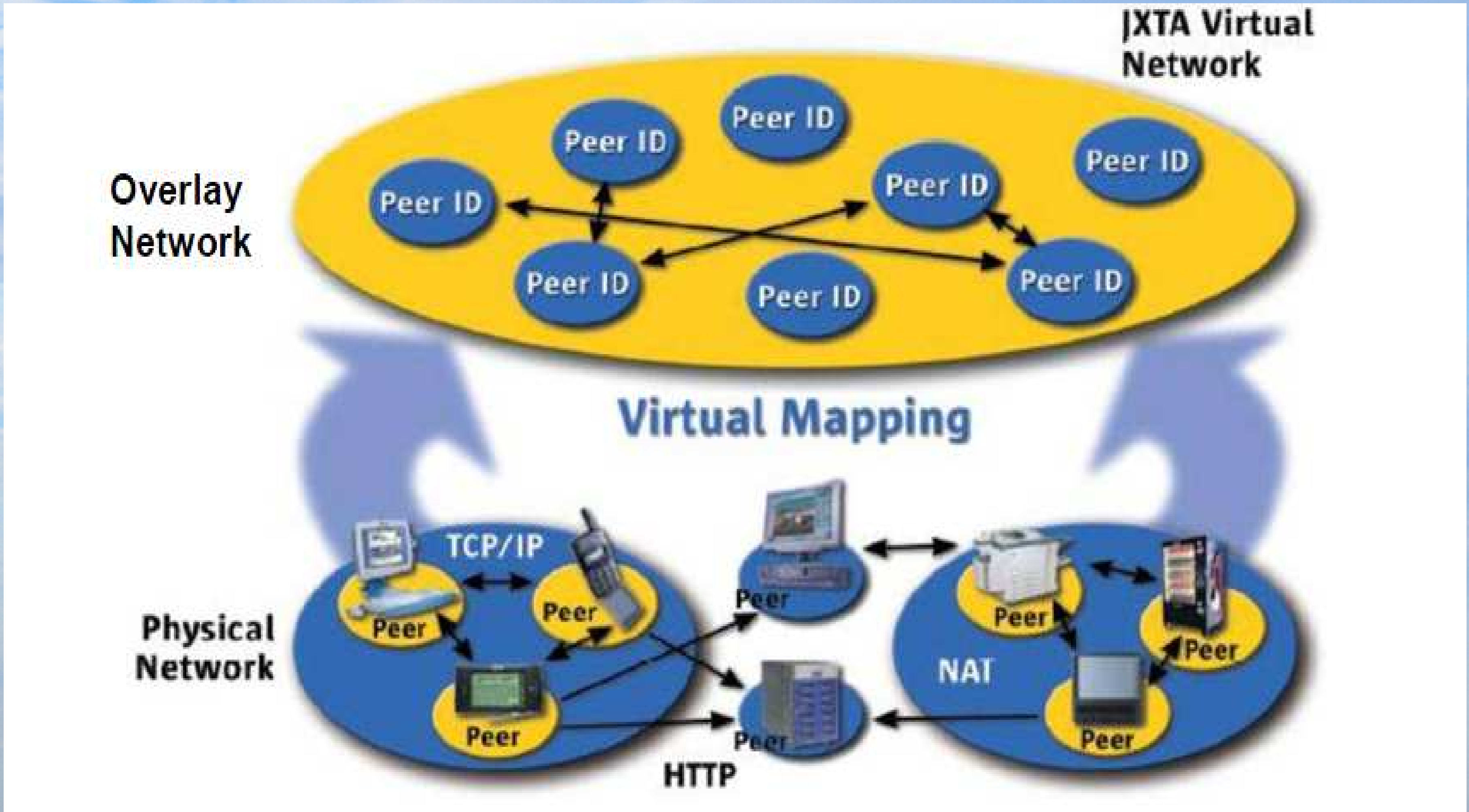


Figure 1.15 The structure of a peer-to-peer system by mapping a physical network to a virtual overlay network (Courtesy of JXTA, http://www.jxta.com)

P2P network Families

Table 1.5 Major Categories of P2P Network Families [42]

System Features	Distributed File Sharing	Collaborative Platform	Distributed P2P Computing	P2P Platform
Attractive Applications	Content distribution of MP3 music, video, open software, etc.	Instant messaging, collaborative design and gaming	Scientific exploration and social networking	Open networks for public resources
Operational Problems	Loose security and serious online copyright violations	Lack of trust, disturbed by spam, privacy, and peer collusion	Security holes, selfish partners, and peer collusion	Lack of standards or protection protocols
Example Systems	Gnutella, Napster, eMule, BitTorrent, Aimster, KaZaA, etc.	ICQ, AIM, Groove, Magi, Multiplayer Games, Skype, etc.	SETI@home, Geonome@home, etc.	JXTA, .NET, FightingAid@home, etc.

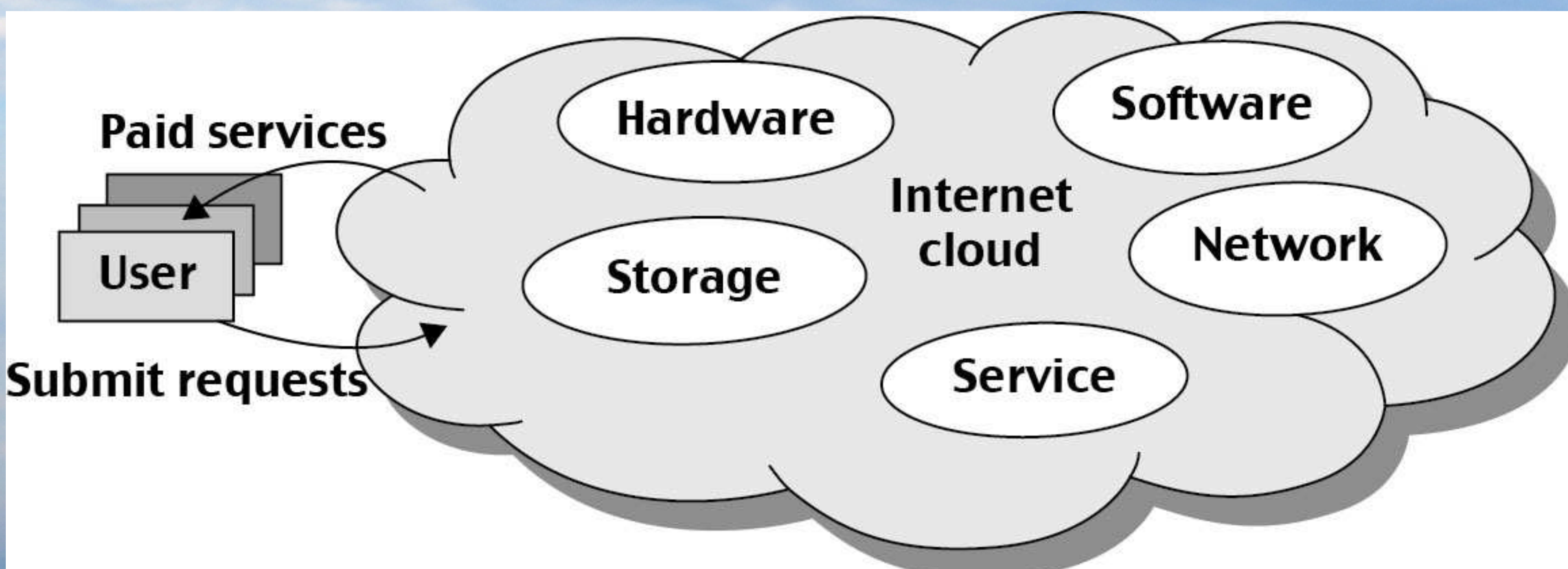
P2P Design Challenges : Hardware, software and network
Too many hardware models and architectures to select, incompatibility between software, OS and different network connection protocols.
Performance affected by routing efficiency and self-organizing nature of peers. Management is difficult, lacks security, No fault-tolerance, No load balancing, Not virus free.

The Cloud

- Historical roots in today's Internet apps
  - Search, email, social networks
  - File storage (Live Mesh, Mobile Me, Flicker, …)
- Computational science is changing to be data-intensive.
- Working with large data sets will mean sending the computations (programs) to the data, rather than copying the data to the workstations.
- A cloud infrastructure provides a framework to manage scalable, reliable, on-demand access to applications
- A cloud is the "invisible" backend to many of our mobile applications
- A model of computation and data storage based on "pay as you go" access to "unlimited" remote data center capabilities

Basic Concept of Internet Clouds

- **Cloud computing** is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet).
- **A cloud is a pool of virtualized computer resources, which can host a variety of different workloads, including batch and interactive applications**
- Cloud ecosystem must be designed to be secure, trustworthy and dependable.



The Next Revolution in IT
Cloud Computing

- Classical Computing
  - Buy & Own
    - Hardware,
    - System Software,
    - Applications often to meet peak needs.
  - Install, Configure, Test, Verify, Evaluate
  - Manage
  - ..
  - Finally, use it
  - $$$$....$(High Capital Investment)

Every 18 months?

- Cloud Computing
  - Subscribe
  - Use

  - $ - pay for what you use, based on QoS

(Courtesy of Raj Buyya, 2012)

Cloud Service Models (1)

Infrastructure as a service (IaaS)

- Most basic cloud service model
- Cloud providers offer computers, servers, storage, network and data centers as physical or more often as virtual machines.
- Virtual machines are run as guests by a hypervisor, such as Xen or KVM.
- Cloud users deploy their applications by then installing operating system images on the machines as well as their application software.
- Cloud providers typically bill IaaS services on a utility computing basis, that is, cost will reflect the amount of resources allocated and consumed.
- Examples of IaaS include: Amazon CloudFormation (and underlying services such as Amazon EC2), Rackspace Cloud, Terremark, and Google Compute Engine.

Cloud Service Models (2)

Platform as a service (PaaS)

- Cloud providers deliver a computing platform typically including operating system, programming language execution environment, database, and web server.

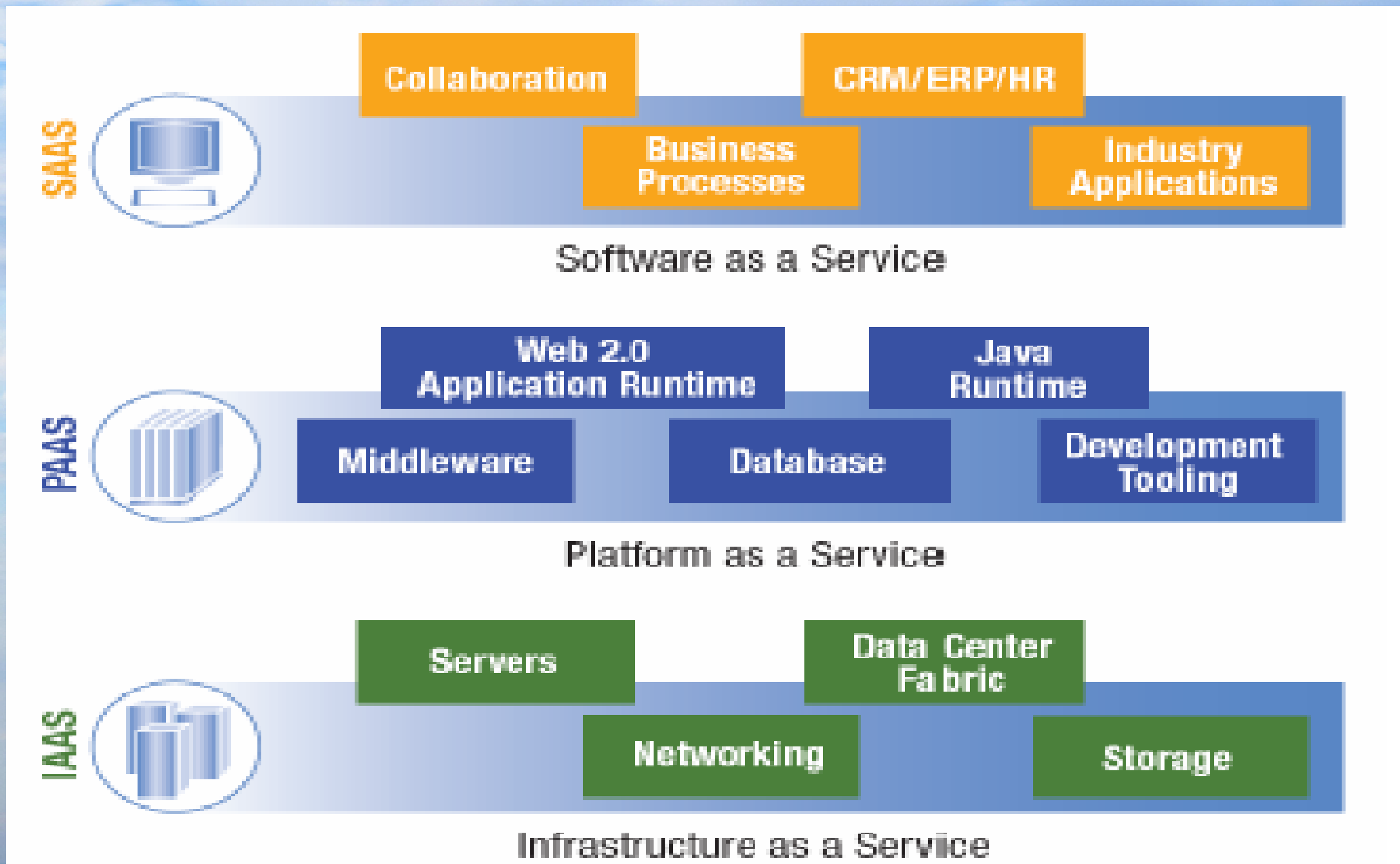
- Application developers develop and run their software on a cloud platform without the cost and complexity of buying and managing the underlying hardware and software layers.
- Examples of PaaS include: Amazon Elastic Beanstalk, Cloud Foundry, Heroku, Force.com, EngineYard, Mendix, Google App Engine, Microsoft Azure and OrangeScape.

Cloud Service Models (3)

Software as a service (SaaS)

- Cloud providers install and operate application software in the cloud and cloud users access the software from cloud clients.
- The pricing model for SaaS applications is typically a monthly or yearly flat fee per user, so price is scalable and adjustable if users are added or removed at any point.
- Examples of SaaS include: Google Apps, innkey pos, QuickBooks Online, Limelight Video Platform, Salesforce.com, and Microsoft Office 365.

Three Cloud Services



Three Cloud Services

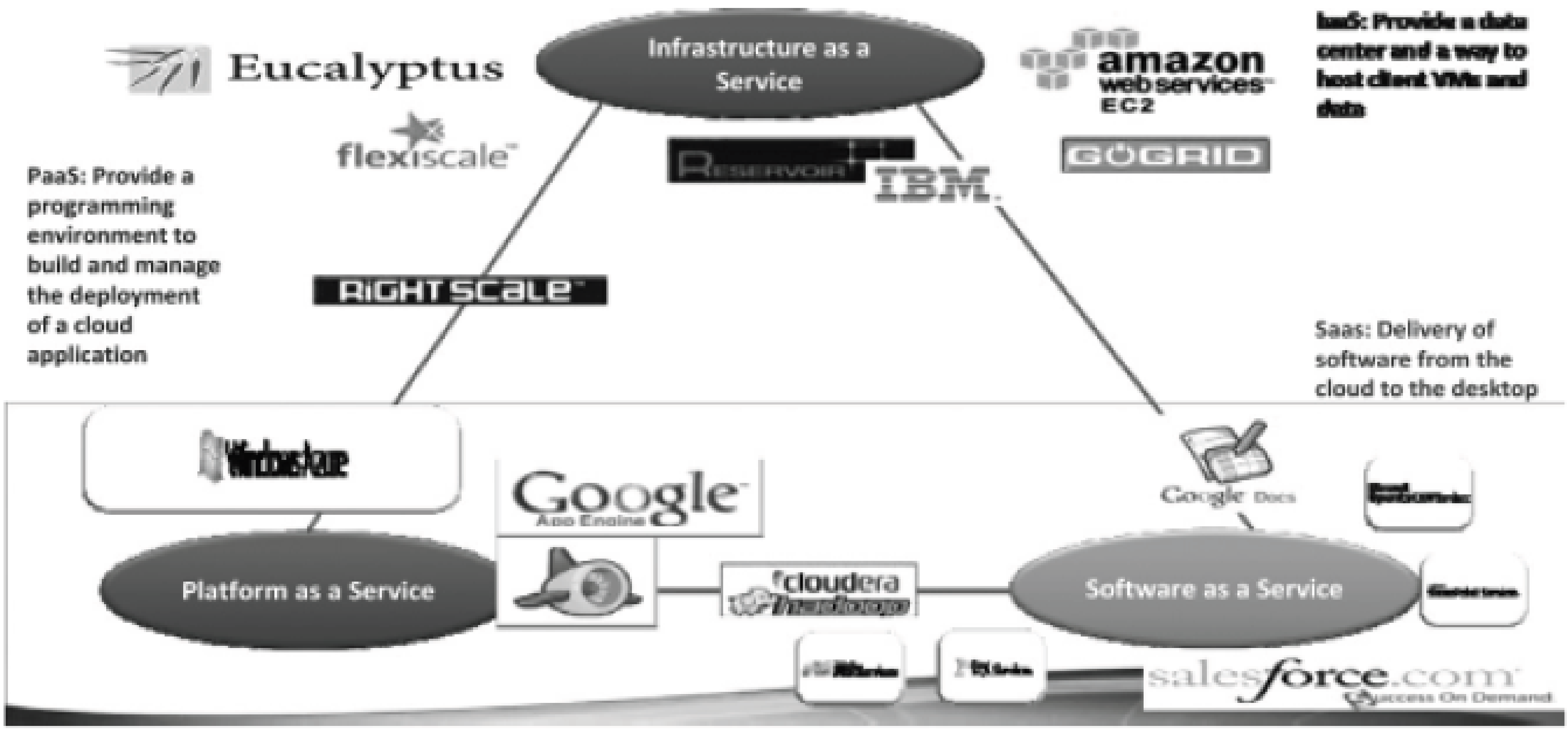
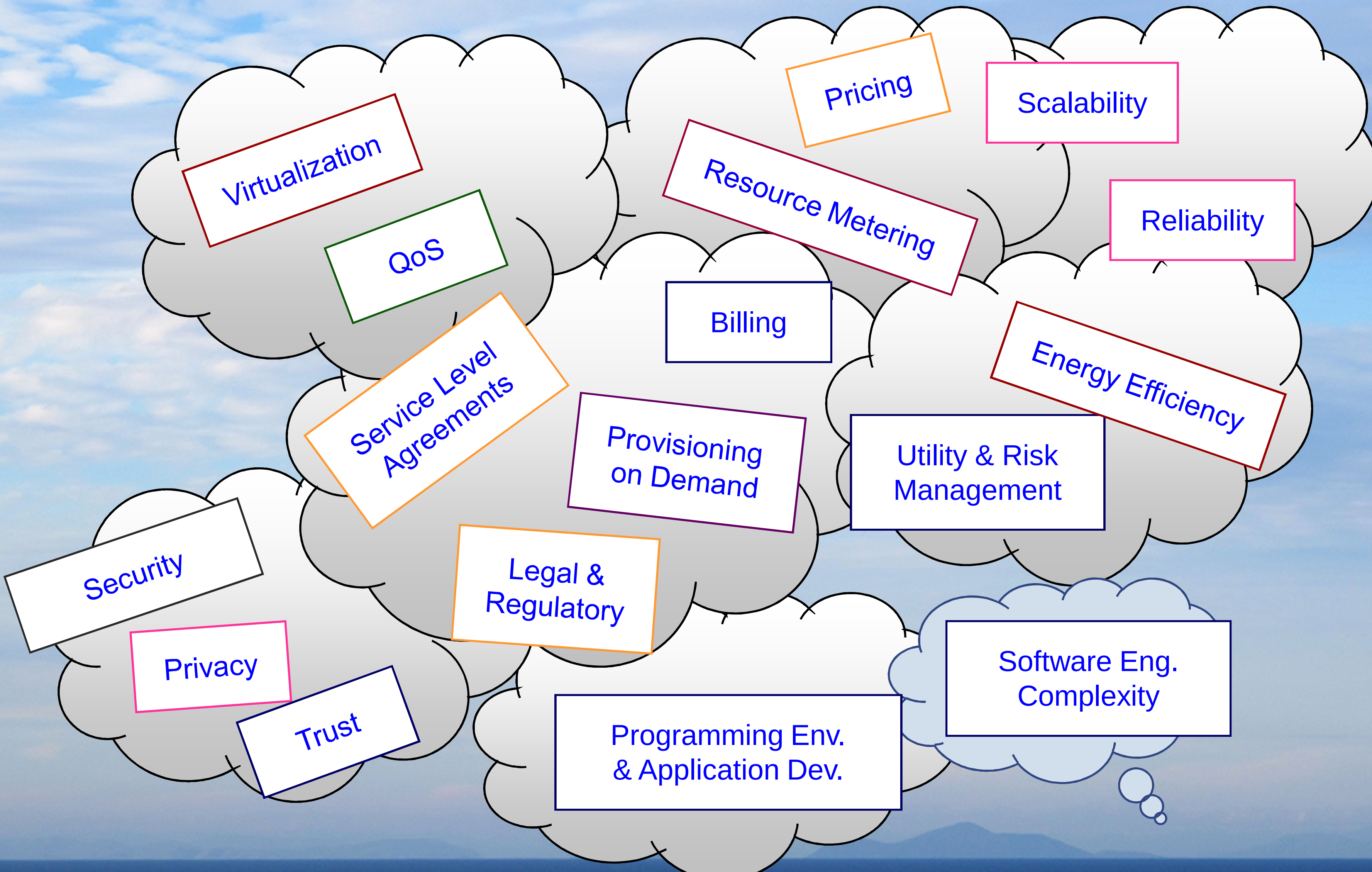


FIGURE 1.19

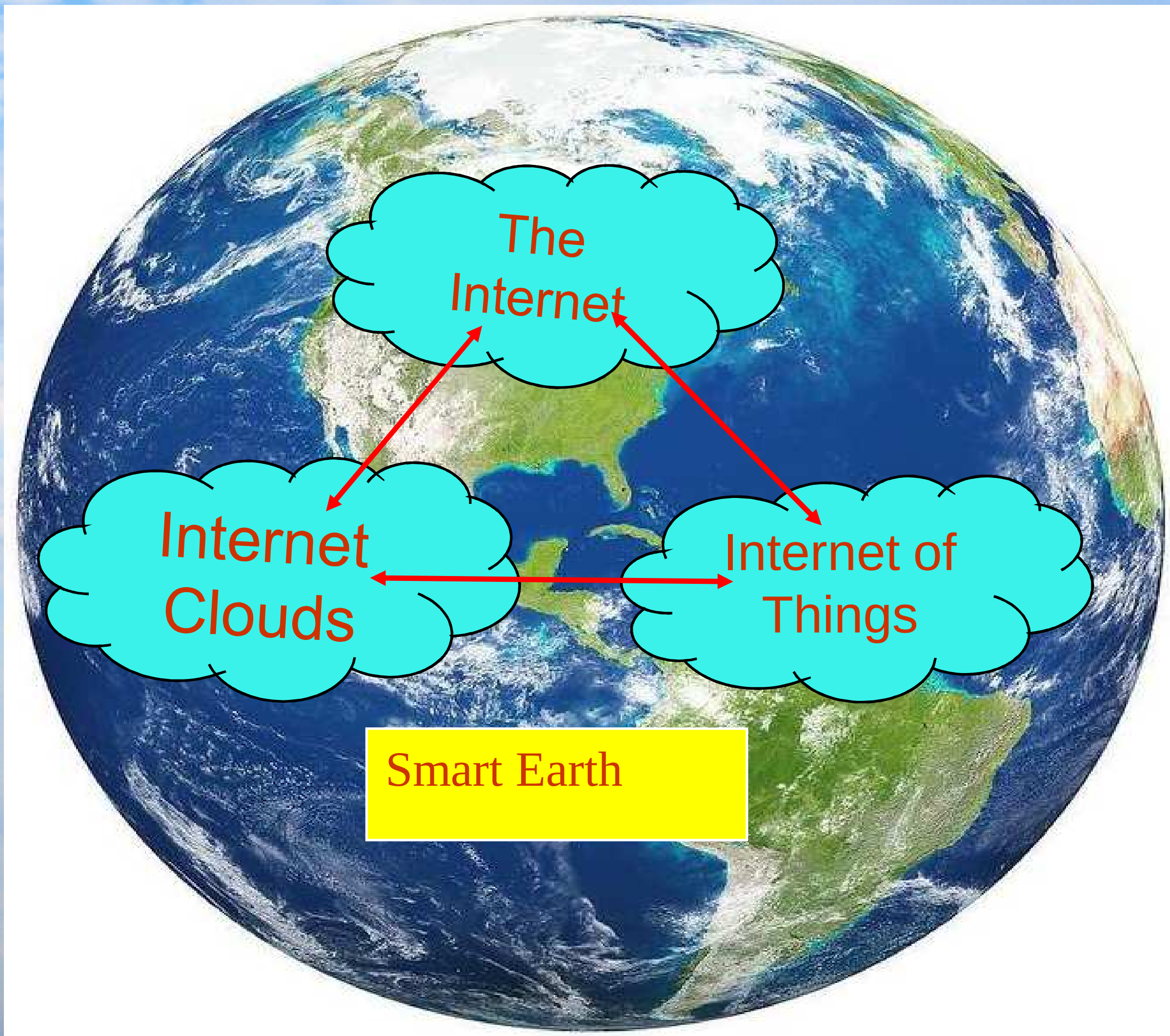
Three cloud service models in a cloud landscape of major providers.

(*Courtesy of Dennis Gannon, keynote address at Cloudcom2010 [19]*)

Cloud Computing Challenges:
Dealing with too many issues (Courtesy of R. Buyya)



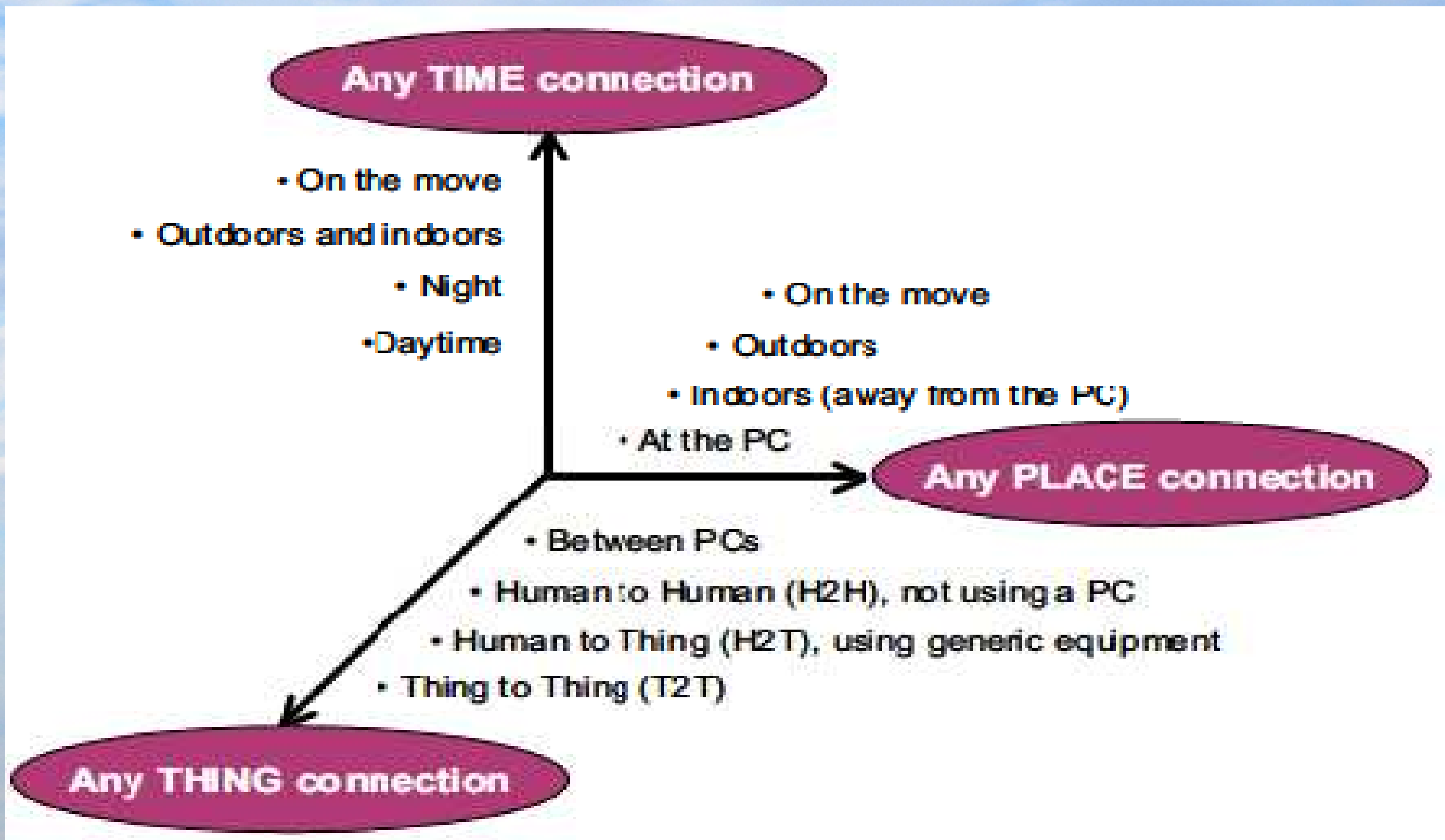
The Internet of Things (IoT)



Smart Earth:

An IBM Dream

Opportunities of IoT in 3 Dimensions



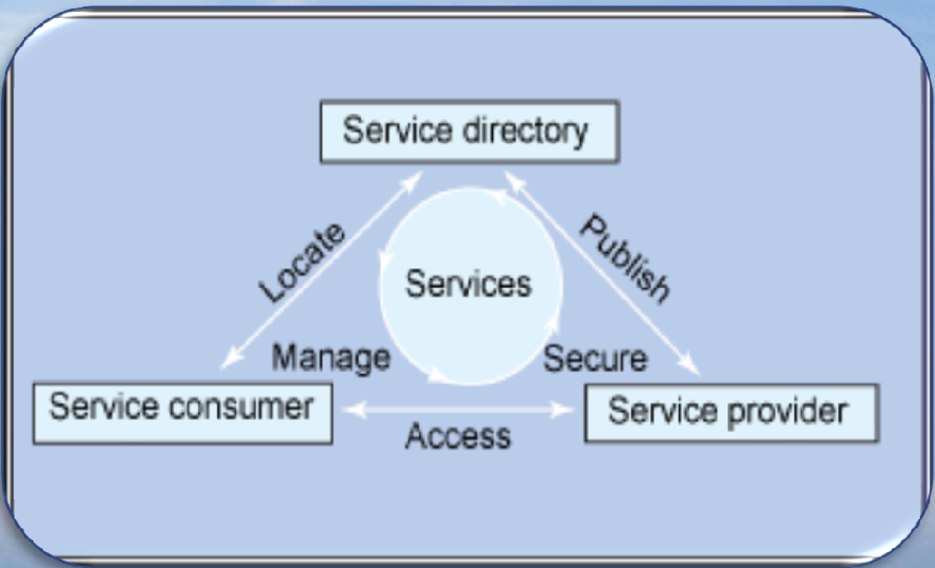
(courtesy of Wikipedia, 2010)

Service-oriented architecture (SOA)

- SOA is an evolution of distributed computing based on the request/reply design paradigm for synchronous and asynchronous applications.
- An application's business logic or individual functions are modularized and presented as services for consumer/client applications.
- Key to these services - their loosely coupled nature;
  - i.e., the service interface is independent of the implementation.
- Application developers or system integrators can build applications by composing one or more services without knowing the services' underlying implementations.
  - For example, a service can be implemented either in .Net or J2EE, and the application consuming the service can be on a different platform or language.

SOA

- A service-oriented architecture (SOA) is an **architectural pattern** in computer software design in which **application components** provide **services** to other **components** via a **communications protocol**, typically over a **network**. The principles of service-orientation are **independent** of any **vendor**, **product** or **technology**.



SOA key characteristics:

- SOA services have self-describing interfaces in platform-independent XML documents.
  - Web Services Description Language (WSDL) is the standard used to describe the services.
- SOA services communicate with messages formally defined via XML Schema (also called XSD).
  - Communication among consumers and providers or services typically happens in heterogeneous environments, with little or no knowledge about the provider.
  - Messages between services can be viewed as key business documents processed in an enterprise.
- SOA services are maintained in the enterprise by a registry that acts as a directory listing.
  - Applications can look up the services in the registry and invoke the service.
  - Universal Description, Definition, and Integration (UDDI) is the standard used for service registry.
- Each SOA service has a quality of service (QoS) associated with it.
  - Some of the key QoS elements are security requirements, such as authentication and authorization, reliable messaging, and policies regarding who can invoke services.

Transparent Cloud Computing Environment

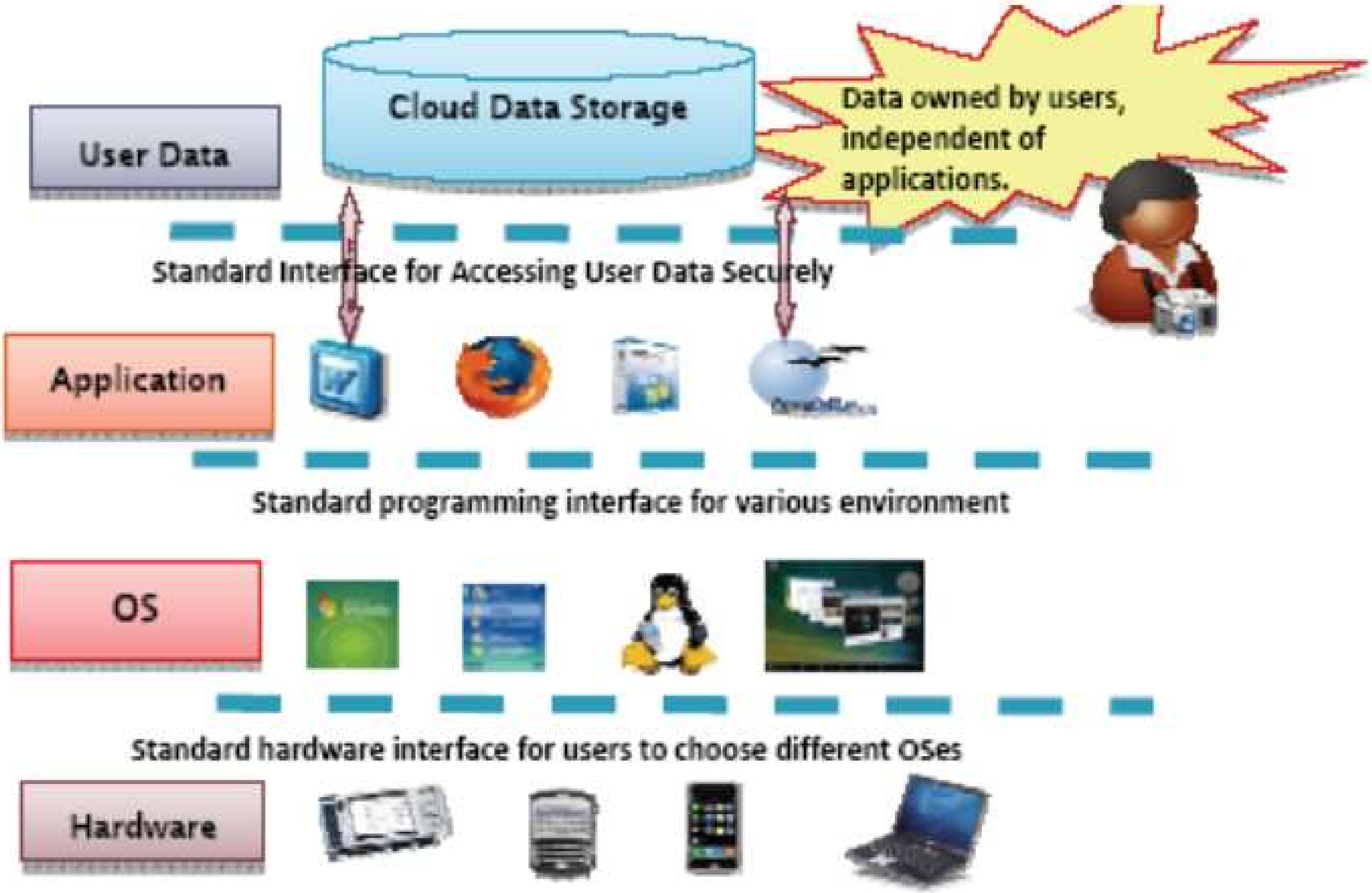


Figure 3 Transparent computing that separates the user data, application, OS, and hardware in time and space – an ideal model for future Cloud platform construction

Parallel and Distributed Programming

Table 1.7 Parallel and Distributed Programming Models and Tool Sets

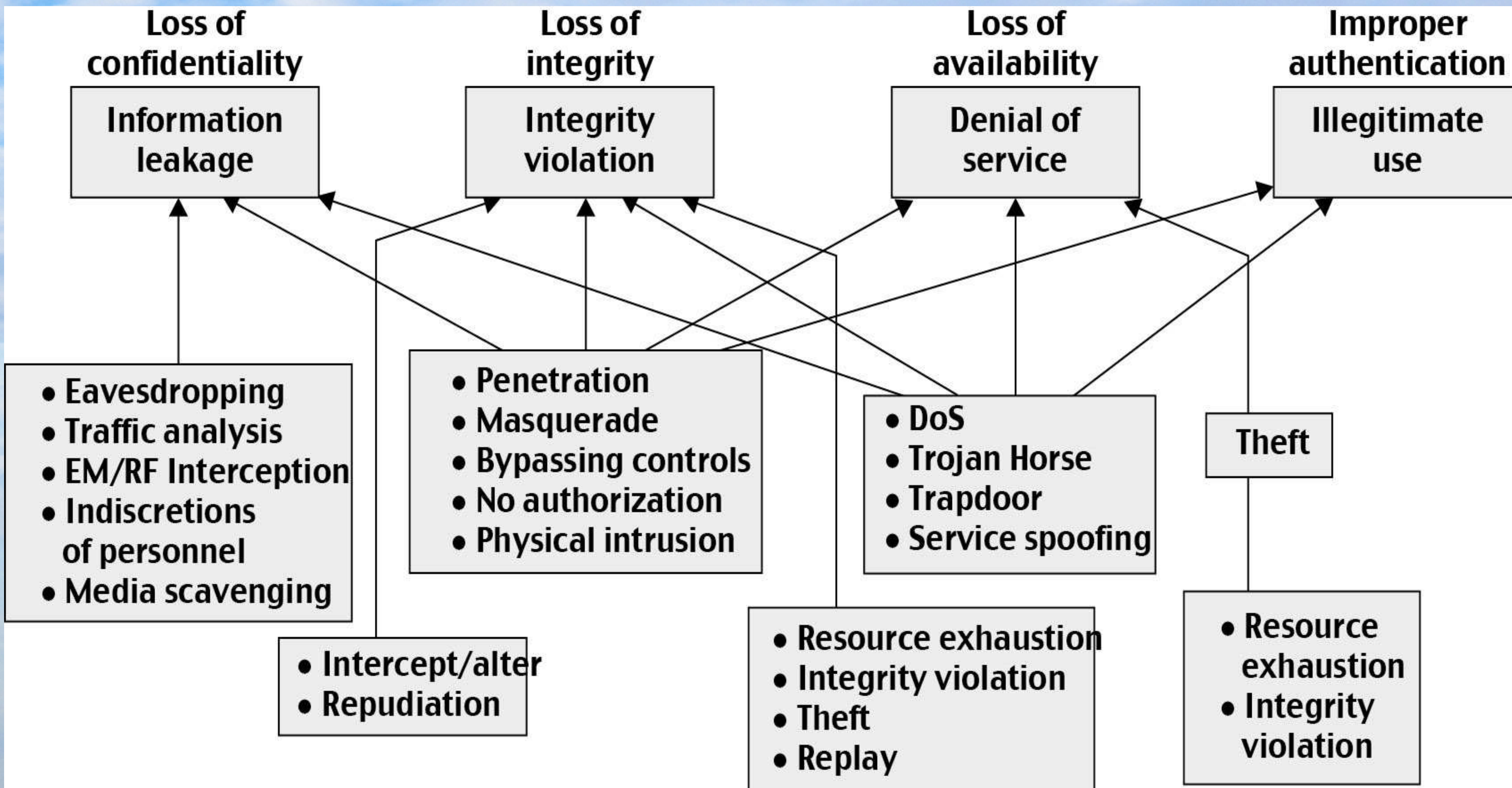
Model	Description	Features
MPI	A library of subprograms that can be called from C or FORTRAN to write parallel programs running on distributed computer systems [6,28,42]	Specify synchronous or asynchronous point-to-point and collective communication commands and I/O operations in user programs for message-passing execution
MapReduce	A Web programming model for scalable data processing on large clusters over large data sets, or in Web search operations [16]	*Map* function generates a set of intermediate key/value pairs; *Reduce* function merges all intermediate values with the same key
Hadoop	A software library to write and run large user applications on vast data sets in business applications (http://hadoop.apache.org/core)	A scalable, economical, efficient, and reliable tool for providing users with easy access of commercial clusters

Grid Standards and Middleware

Table 1.9 Grid Standards and Toolkits for scientific and Engineering Applications

Grid Standards	Major Grid Service Functionalities	Key Features and Security Infrastructure
OGSA Standard	Open Grid Service Architecture offers common grid service standards for general public use	Support heterogeneous distributed environment, bridging CA, multiple trusted intermediaries, dynamic policies, multiple security mechanisms, etc.
Globus Toolkits	Resource allocation, Globus security infrastructure (GSI), and generic security service API	Sign-in multi-site authentication with PKI, Kerberos, SSL, Proxy, delegation, and GSS API for message integrity and confidentiality
IBM Grid Toolbox	AIX and Linux grids built on top of Globus Toolkit, autonomic computing, Replica services	Using simple CA, granting access, grid service (ReGS), supporting Grid application for Java (GAF4J), GridMap in IntraGrid for security update.

Security: System Attacks and Network Threads



http://www.go4expert.com/articles/types-of-attacks-t7685/

https://www.owasp.org/index.php/Repudiation_Attack

Summary

- Cluster and co-operative computers
- Grid computing Infrastructures
- Peer –to-Peer Network
- Cloud Computing over Internet.
  - 3 Cloud Service Models
  - Cloud Computing Challenges
- IoT
- Service Oriented Architecture (SOA)

Reference Books

1. K. Hwang, G. Fox, and J. Dongarra, *Distributed and Cloud Computing: from Parallel Processing to the Internet of Things Morgan Kauffmann Publishers*, 2011

2. Greg F Pfister, In search of clusters, Second ed., Prentice hall , 2001

3. R. Buyya, J. Broberg, and A. Goscinski (eds), Cloud Computing: Principles and Paradigms, ISBN-13: 978-0470887998, Wiley Press, USA, February 2011.

4. T. Chou, *Introduction to Cloud Computing: Business and Technology*, Lecture Notes at Stanford University and at Tsinghua University, Active Book Press, 2010.

5. T. Hey, Tansley and Tolle (Editors), *The Fourth Paradigm : Data-Intensive Scientific Discovery*, Microsoft Research, 2009.