

Introduction of Internet of Things

Lecture 6: Supporting Technologies of IoT

Haitao Zhang

Beijing Key Lab of Intelligent Telecomm. Software and Multimedia
Beijing University of Posts and Telecommunications

Outline

- Cloud computing
- Security issues



6.1 Cloud Computing



Evolution of cloud computing



Evolution of Computing with Network

■ Network Computing

- Network is computer (client - server)
- Separation of Functionalities



■ Cluster Computing

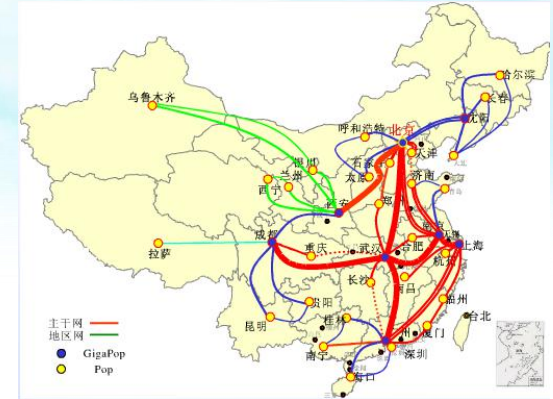
- Tightly coupled computing resources:
CPU, storage, data, etc. Usually connected within a LAN
- Managed as a single resource
- Commodity, Open source



Evolution of Computing with Network

■ Grid Computing

- Resource sharing across several domains
- Decentralized, open standards
- Global resource sharing



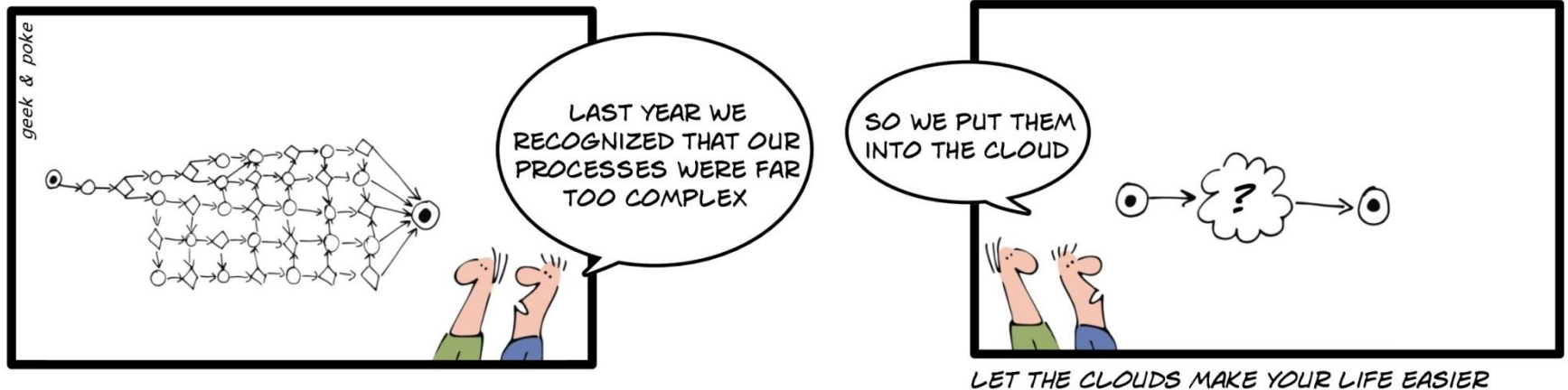
■ Utility Computing

- Don't buy computers, lease computing power
- Upload, run, download
- Ownership model

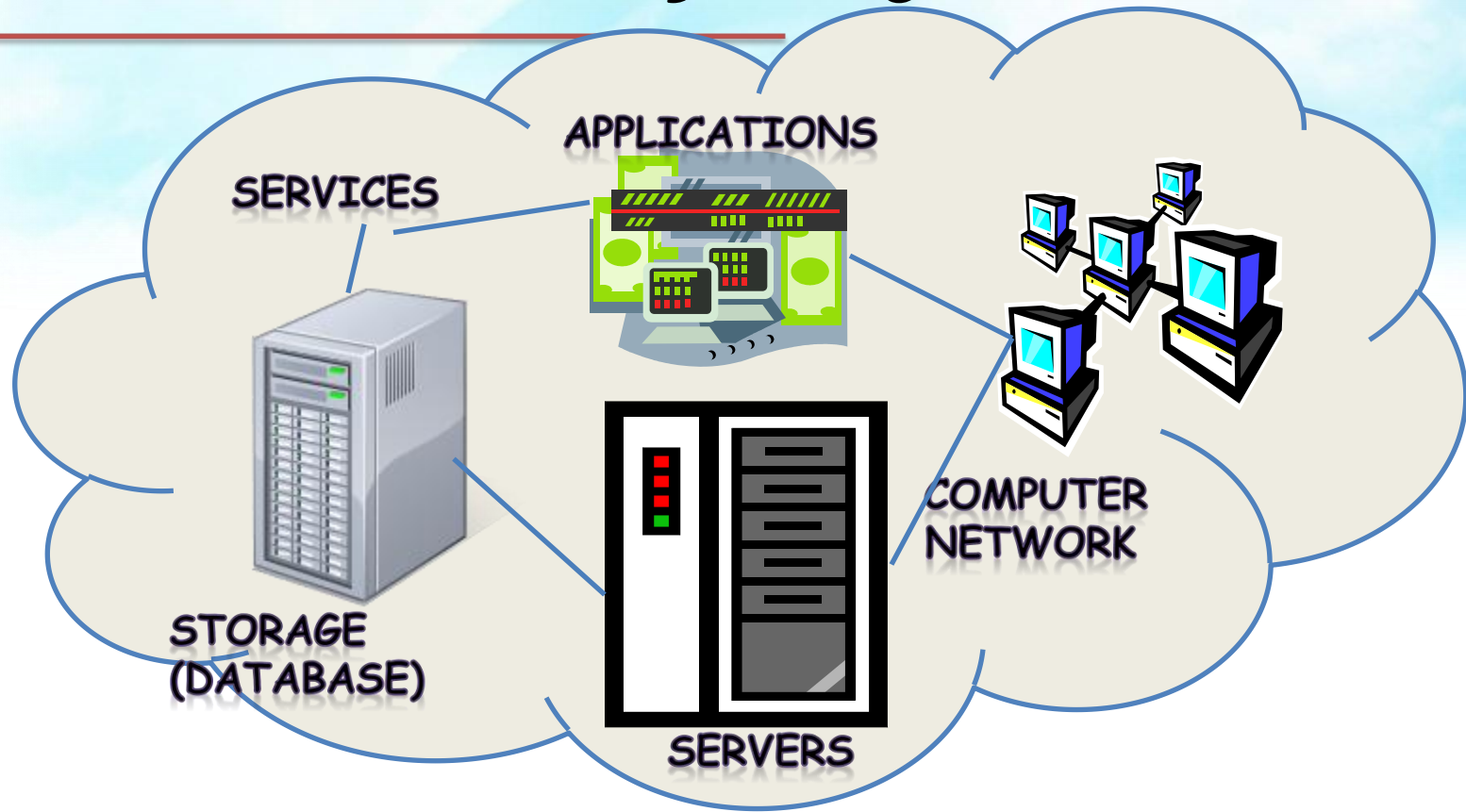


The Next Step: Cloud Computing

- **Service and data** are in the **cloud**, accessible with any device connected to the cloud with a browser.
- A key technical issue for developer:
 - **Scalability**
- Services are not known geographically.



What is Cloud Computing



- Shared pool of configurable computing resources
- On-demand network access
- Provisioned by the Service Provider



What is Cloud Computing?

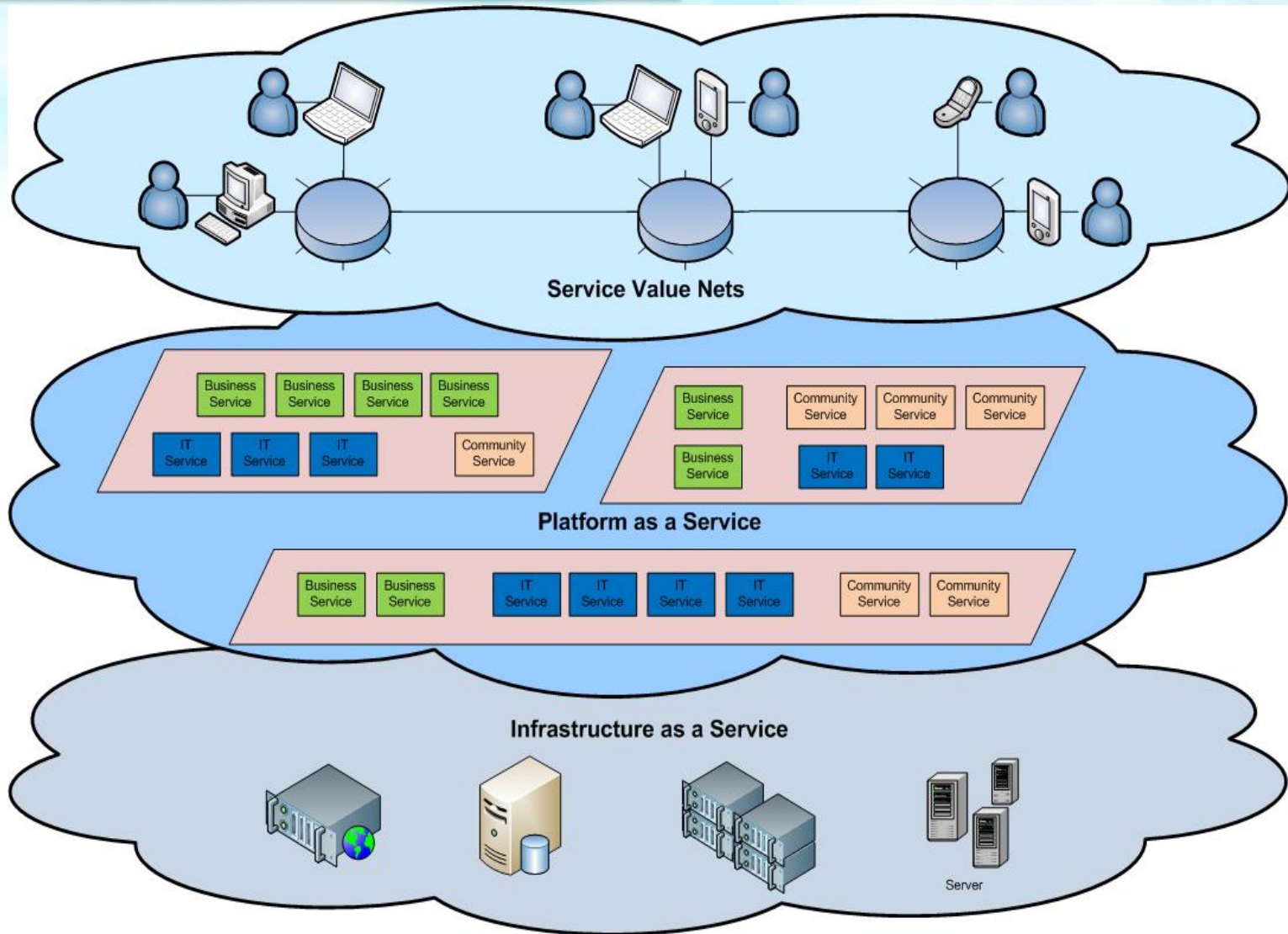
- **Cloud Computing** is a general term used to describe a new class of **network based computing** that takes place over the Internet, basically a step on from **Utility Computing**.
- In other words, this is a collection/group of integrated and networked hardware, software and Internet infrastructure (called a platform).
- Using the **Internet** for communication and transport provides hardware, software and networking services to clients.
- These platforms **hide the complexity** and details of the underlying infrastructure from users and applications by providing very simple graphical interface or API (Applications Programming Interface).

What is Cloud Computing?

- In addition, the platform provides on demand services, that are always on, **anywhere, anytime and any place**.
- **Pay for use** and as needed, elastic (scale up and down in capacity and functionalities).
- The **hardware and software services** are available to the general public, enterprises, corporations and businesses markets.



Cloud Architecture



Cloud Summary

- Cloud computing is an umbrella term used to refer to Internet based development and services.
- A number of characteristics define cloud data, applications services and infrastructure:
 - **Remotely hosted:** Services or data are hosted on remote infrastructure.
 - **Ubiquitous:** Services or data are available from anywhere.
 - **Commodified:** The result is a utility computing model similar to that of traditional utilities, like gas and electricity - you pay for what you would want!



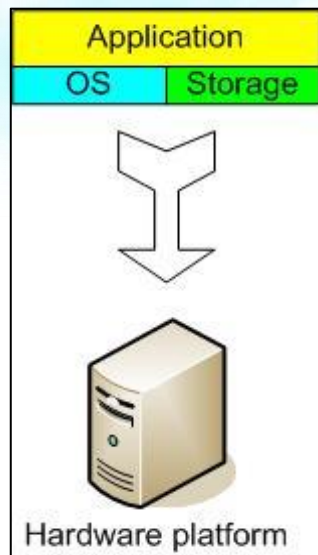
Comparison

Cloud computing shares characteristics with:

- Mainframe computer
- Client–server
- Grid computing
- Peer-to-peer



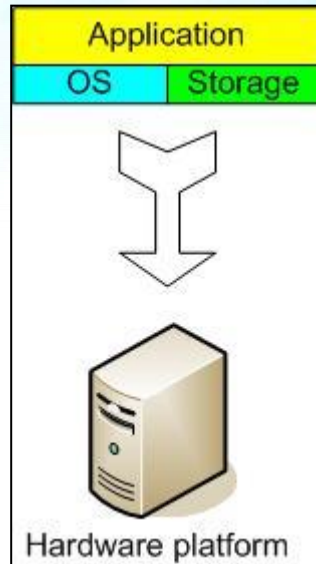
The Traditional Server Concept



Web Server

Windows

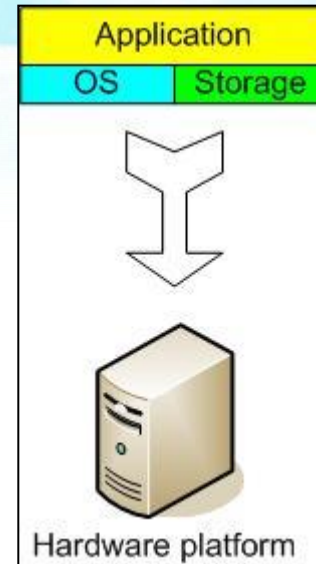
IIS



App Server

Linux

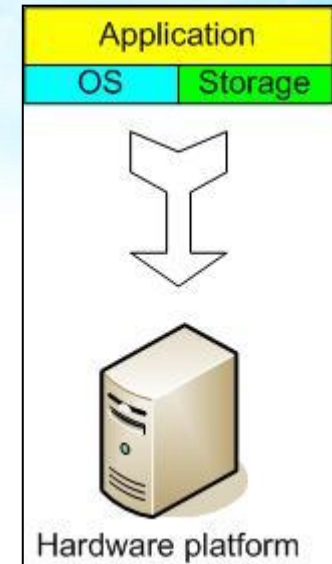
Glassfish



DB Server

Linux

MySQL



EMail

Windows

Exchange

Traditional Server Concept Explained

- Servers considered as a whole unit that includes the hardware, the OS, the storage, and the applications.
- Often referred to by their function i.e. the Exchange server, the SQL server, the File server, etc.
- If the File server fills up, or the Exchange server becomes overtaxed: must add in a new server.
- Unless there are multiple servers, if a service experiences a hardware failure, the service is down.
- Can implement clusters of servers to make them more fault tolerant.



Pros and Cons

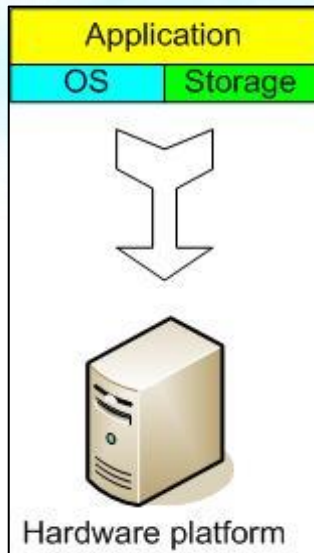
■ Pros

- Easy to conceptualize
- Fairly easy to deploy
- Easy to backup
- Virtually any application/service can be run from this type of setup

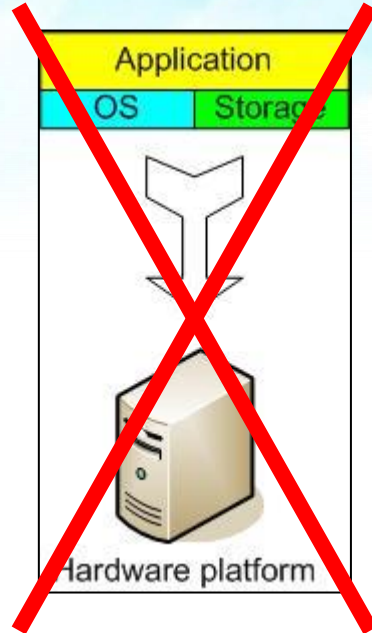
■ Cons

- Expensive to acquire and maintain hardware
- Not very scalable
- Difficult to replicate
- Redundancy is difficult to implement
- Vulnerable to hardware outages
- In many cases, processor is under-utilized

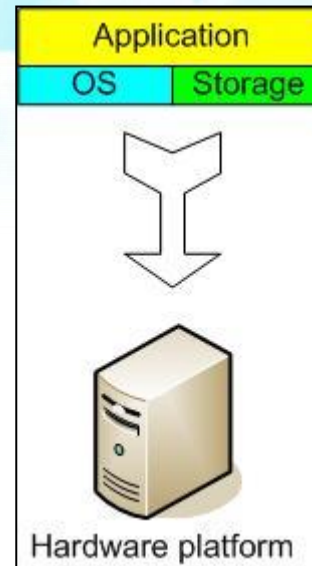
And if something goes wrong ...



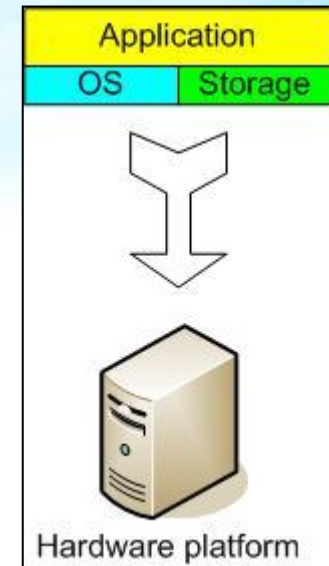
Web Server
Windows
IIS



App Server
DOWN!

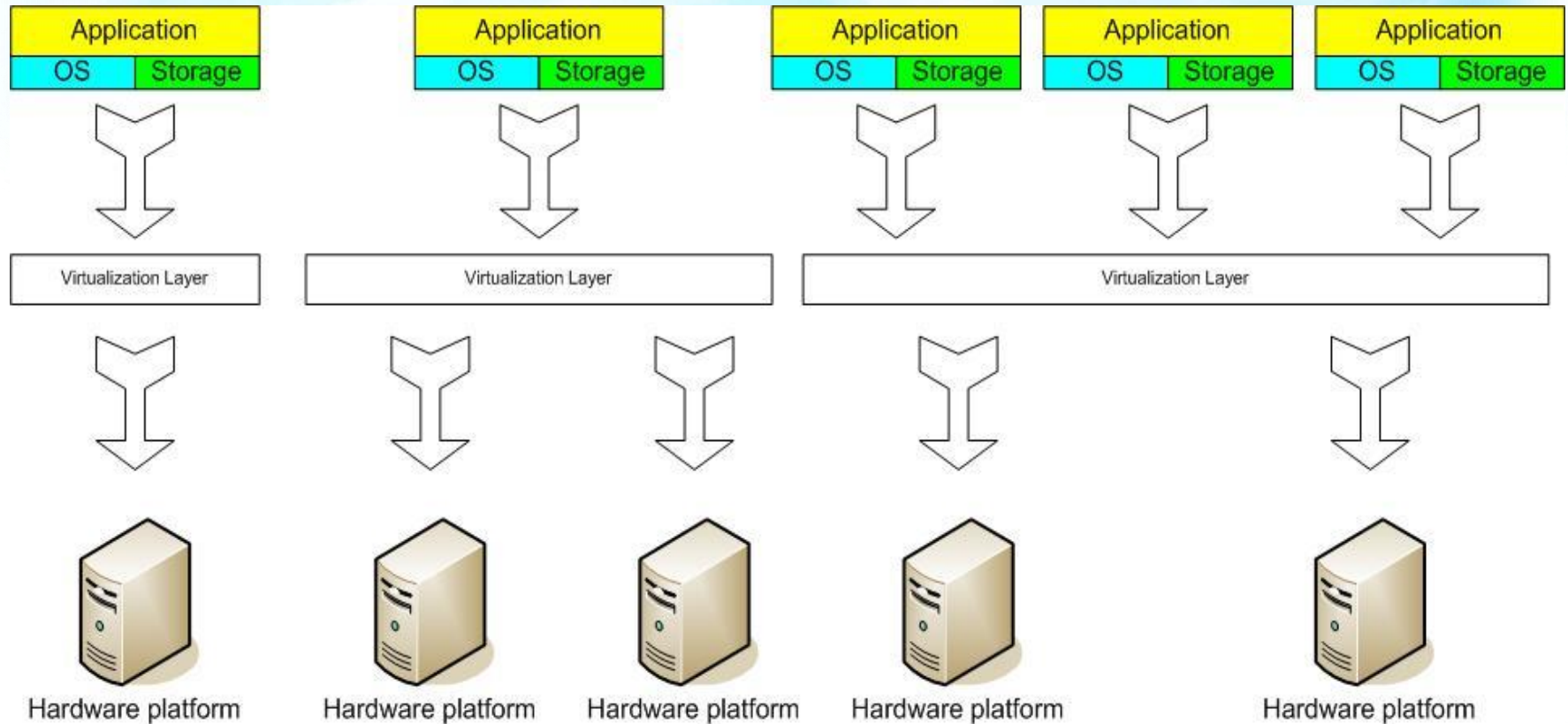


DB Server
Linux
MySQL



EMail
Windows
Exchange

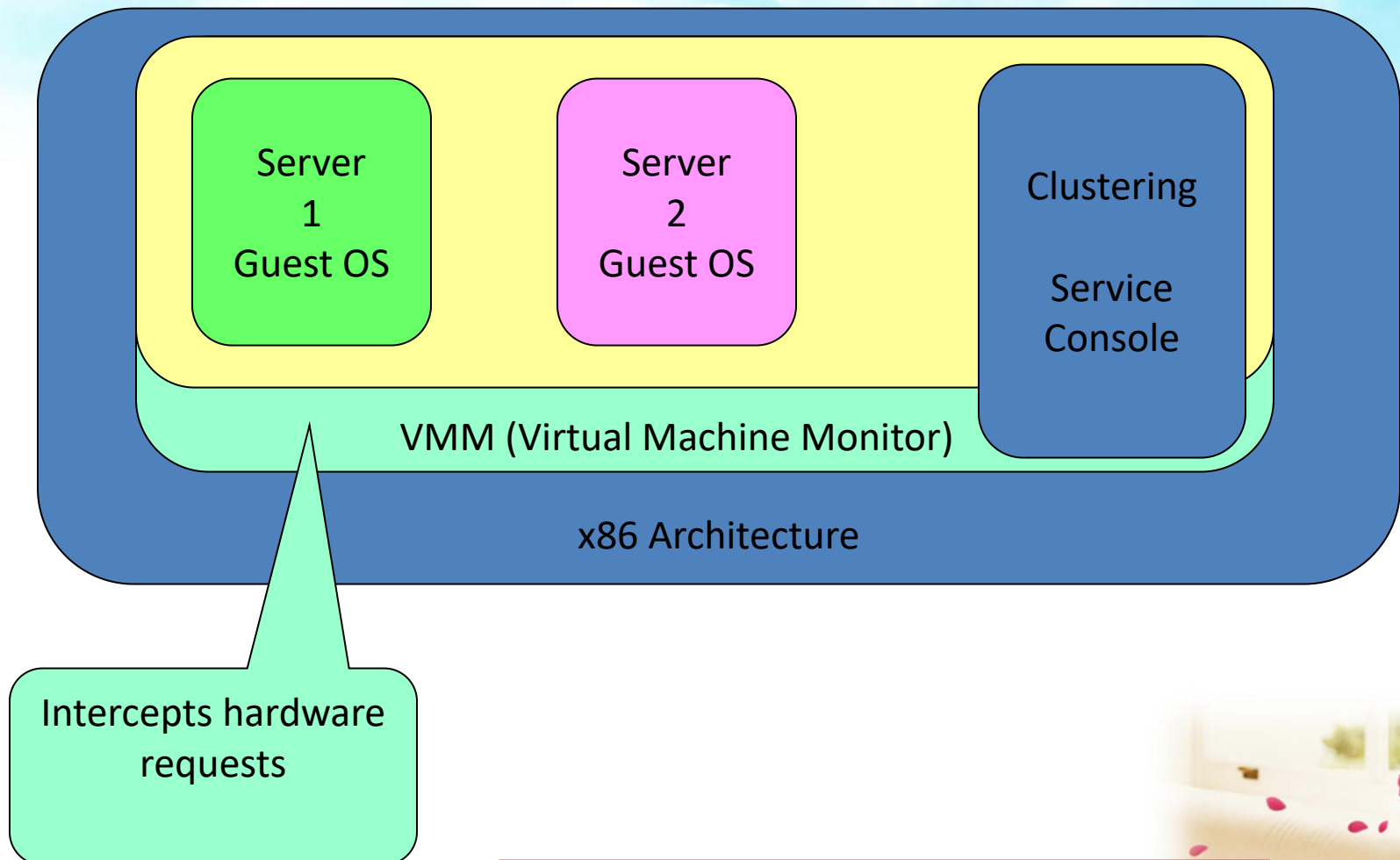
The Virtual Server Concept



Virtual Machine Monitor (VMM) layer between *Guest OS* and hardware

Close-up

* adapted from a diagram in VMware white paper, *Virtualization Overview*



Virtual Server Concept

- Virtual servers seek **to encapsulate the server software** away from the hardware
 - This includes the OS, the applications, and the storage for that server.
- **A virtual server can be serviced by one or more hosts, and one host may house more than one virtual server.**
- Virtual servers can still be referred to by their function i.e. email server, database server, etc.
- If the environment built correctly, virtual servers will **not be affected by the loss of a host.**



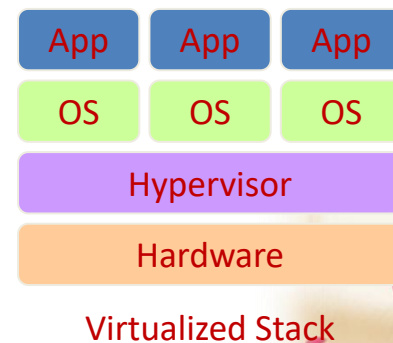
Virtual Server Concept

- Can be **scaled out easily**.
 - If the resources supporting a virtual server are being taxed too much, admin can adjust the amount of resources allocated to that virtual server.
- **Server templates** can be created in a virtual environment to be used to **create multiple, identical virtual servers**.
- Virtual servers themselves can be migrated from host to host almost at will.



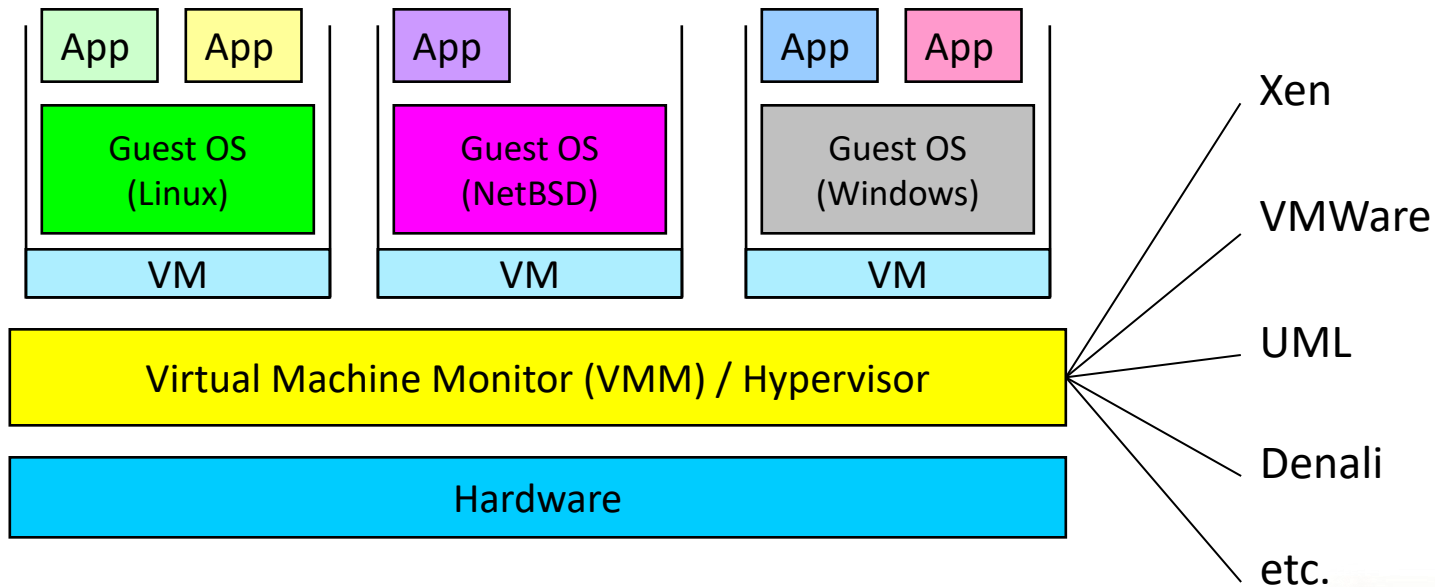
Virtualization

- Virtual workspaces:
 - An abstraction of an execution environment that can be made dynamically available to authorized clients by using well-defined protocols,
 - Resource quota (e.g. CPU, memory share),
 - Software configuration (e.g. O/S, provided services).
- Implement on Virtual Machines (VMs):
 - **Abstraction of a physical host machine,**
 - Hypervisor intercepts and emulates instructions from VMs, and allows management of VMs,
 - VMWare, Xen, etc.
- Provide infrastructure API:
 - Plug-ins to hardware/support structures



Virtual Machines

- VM technology allows multiple virtual machines to run on a single physical machine.



Performance: Para-virtualization (e.g. Xen) is very close to raw physical performance!

Virtualization in General

- Advantages of virtual machines:
 - Run operating systems where the physical hardware is unavailable,
 - Easier to create new machines, backup machines, etc.,
 - Software testing using “clean” installs of operating systems and software,
 - Emulate more machines than are physically available,
 - Timeshare lightly loaded systems on one host,
 - Debug problems (suspend and resume the problem machine),
 - Easy migration of virtual machines (shutdown needed or not).
 - Run legacy systems!



Pros and Cons

■ Pros

- Resource pooling
- Highly redundant
- Highly available
- Rapidly deploy new servers
- Easy to deploy
- Reconfigurable while services are running
- Optimizes physical resources by doing more with less

■ Cons

- Slightly harder to conceptualize
- Slightly more costly (must buy hardware, OS, Apps, and now the abstraction layer)

Layers

Client

Computer hardware and/or computer software relying on cloud computing for application delivery.

Application

Application services (SaaS).

Platform

Platform services (PaaS).

Infrastructure

Infrastructure services (IaaS).

Server

Computer hardware, software products specifically designed for delivery of cloud services.

Cloud Service Delivery Models

- **IaaS:** Infrastructure as a Service
 - provisions computing resources within provider's infrastructure upon which they can deploy and run arbitrary software, including OS and applications.
- **PaaS:** Platform as a Service
 - can create custom applications using programming tools supported by the provider and deploy them onto the provider's cloud infrastructure.
- **SaaS:** Software as a Service
 - use provider's applications running on provider's cloud infrastructure.

Cloud Service Models

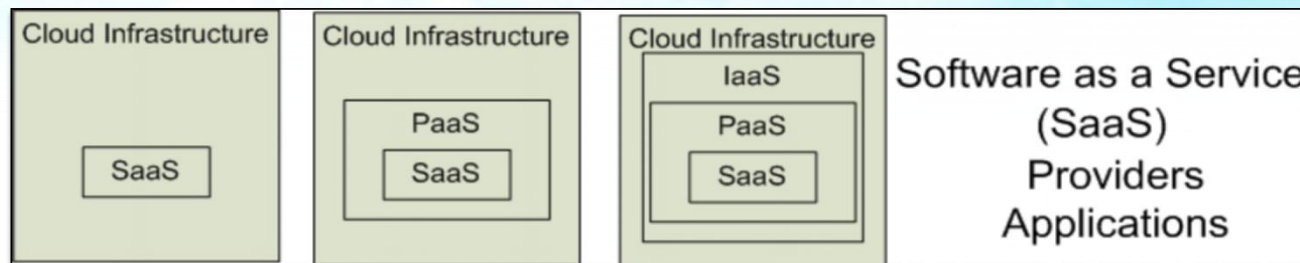
Software as a Service (SaaS)

Platform as a Service (PaaS)

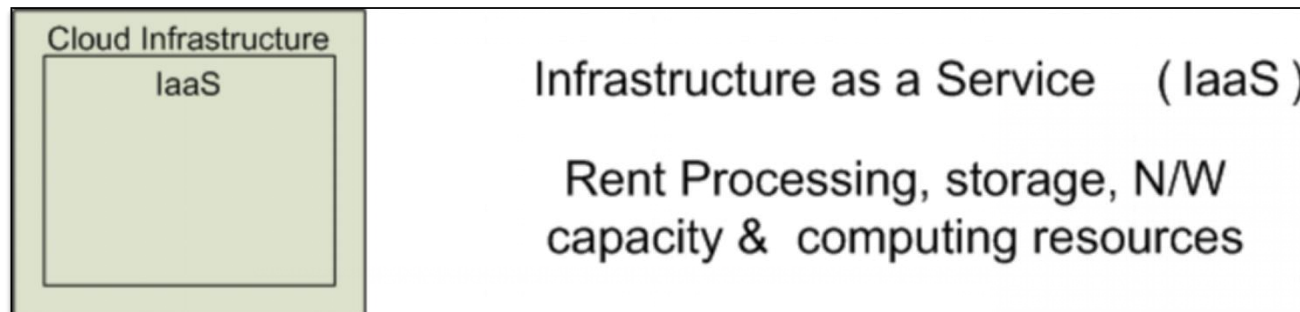
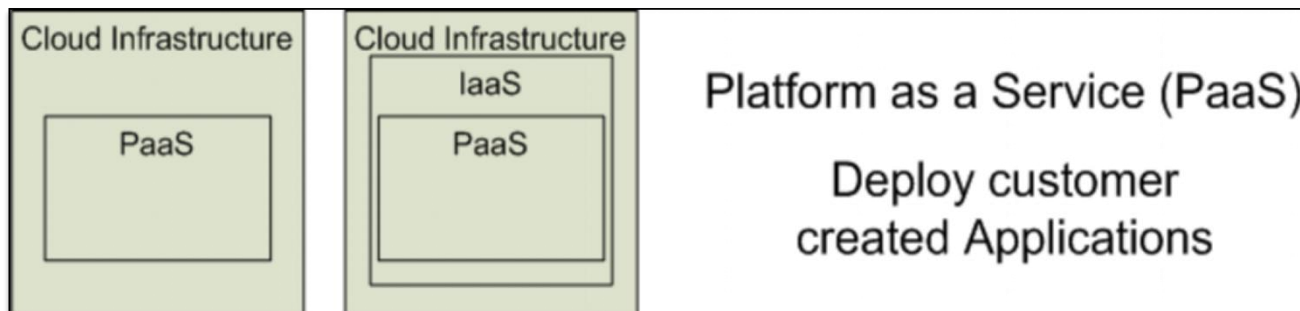
Infrastructure as a Service (IaaS)

SalesForce CRM

LotusLive



Google App Engine



Software as a Service (SaaS)

- SaaS is a model of software deployment where an application is hosted as a service provided to customers across the Internet.
- SaaS alleviates the burden of software maintenance/support
 - but users relinquish control over software versions and requirements.
- Terms that are used in this sphere include
 - **Platform as a Service (PaaS)** and
 - **Infrastructure as a Service (IaaS)**



Three Features of Mature SaaS Applications

- Scalable
 - Handle growing amounts of work in a graceful manner
- Multi-tenancy
 - One application instance may be serving hundreds of companies
 - Opposite of multi-instance where each customer is provisioned their own server running one instance
- Metadata driven configurability
 - Instead of customizing the application for a customer (requiring code changes), one allows the user to configure the application through metadata

Platform-as-a-Service (PaaS)

■ Definition

- Platform providing all the facilities necessary to support the complete process of building and delivering web applications and services, all available over the Internet
- Entirely **virtualized platform** that includes one or more servers, operating systems and specific applications



PaaS Example: Google App Engine

- Service that allows user to **deploy user's Web applications** on Google's very scalable architecture
- Providing user with **a sandbox** for user's Java and python application that can be referenced over the Internet
- Providing Java and Python APIs for persistently **storing and managing data** (using the Google Query Language or GQL)

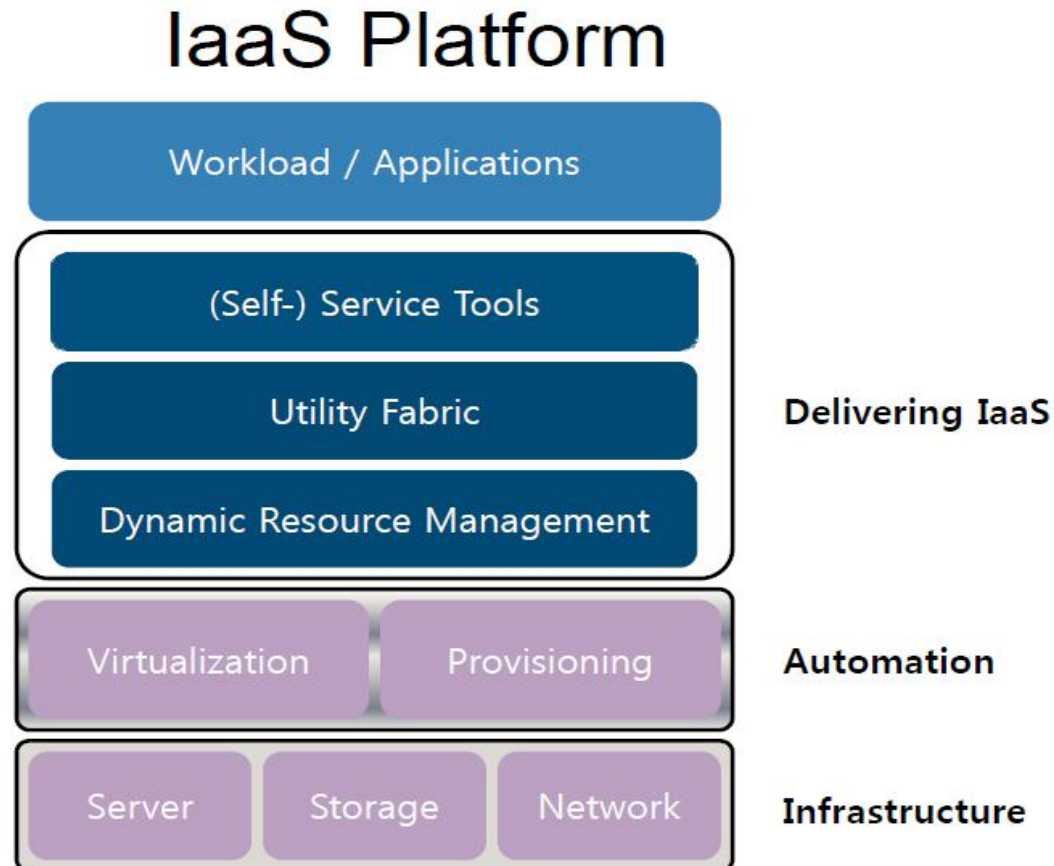


Infrastructure-as-a-Service (IaaS)

- Definition
 - Provision model in which an organization **outsources the equipment used to support operations**, including storage, hardware, servers and networking components.
 - Also known as Hardware as a Service (HaaS).
 - Service provider owns the equipment; responsible for housing, running and maintaining it.
 - Client typically **pays on a per-use basis**.



Infrastructure-as-a-Service (IaaS)









©Copyrights 2009 Seoul National University All Rights Reserved

Characteristics of Infrastructure-as-a-Service (IaaS)

- Utility computing and billing model
- Automation of administrative tasks
- Dynamic scaling
- Desktop virtualization
- Internet connectivity



Service Delivery Model Examples

	Amazon	Google	Microsoft	Salesforce
SaaS				
PaaS				
IaaS				

4 *Cloud Deployment Models*

- Private cloud
 - enterprise owned or leased
- Community cloud
 - shared infrastructure for specific community
- Public cloud
 - Sold to the public, mega-scale infrastructure
- Hybrid cloud
 - composition of two or more clouds

Web-Scale & Large data centers Problems

■ Characteristics:

- Definitely data-intensive
- May also be processing intensive

■ Examples:

- Crawling, indexing, searching, mining the Web
- “Post-genomics” life sciences research
- Other scientific data (physics, astronomers, etc.)
- Sensor networks
- Web 2.0 applications

How much data?

- Wayback Machine has 3 PB + 100 TB/month (2009)
- Google processes 20 PB a day (2008)
- “all words ever spoken by human beings” ~ 5 EB
- NOAA has ~1 PB climate data (2007)
- CERN’s LHC generates 15 PB a year (2010)



640K ought to be enough for anybody.

Large Data Centers

- Web-scale problems? Throw more machines at it!
- Clear trend: centralization of computing resources in large data centers
- Important Issues:
 - Redundancy
 - Efficiency
 - Utilization
 - Management



The “Cloud” = 10X Improvements

- Ease of Use
- Scalability
- Risk
- Reliability
- Cost



Ease of Use

- Deploy infrastructure with a mouse or API
 - Cloud computing providers **deliver applications via the internet**, which are accessed from web browsers and desktop and mobile apps
 - Do it yourself remotely from anywhere anytime



Scalability

- **Dynamic provisioning of resources** on a fine-grained, self-service basis near real-time, without users having to engineer for peak loads
- Control your infrastructure with your app



Risk

- Nothing to buy
- Cancel immediately
- Change instantly, even operating systems
- Rebuild it instantly after testing



Reliability

- Based on enterprise grade hardware
- Design for failures:
 - Automatically spin up replacements
 - Use multiple clouds



Cost Control

- Cost
 - Many systems have variable demands
 - Batch processing (e.g. New York Times)
 - Web sites with peaks (e.g. Forbes)
 - Startups with unknown demand (e.g. the Cash for Clunkers program)
 - Reduce risk
 - Don't need to buy hardware until you need it



Business Agility

- More than scalability - *elasticity*!
 - Ely Lilly in rapidly changing health care business
 - Used to take 3 - 4 months to give a department a server cluster, then they would hoard it!
 - Using EC2, about 5 minutes!
 - And they give it back when they are done!
- **Scaling back** is as important as scaling up



Google Cloud

- Started with Google Apps
- Platform as Service later on
- Replace office software
 - Gmail
 - Google Docs (word processing and spreadsheets)
 - Google video for business
 - Google sites (intranet sites and wikis)
- Google Cloud Connect
- 500,000+ organizations use Google Apps
- GE moved 400,000 desktops from Microsoft Office to Google Apps

Microsoft Azure Services

 Windows Live™

 Microsoft Office Live

Microsoft Exchange Online

Microsoft SharePoint Online

 Microsoft Dynamics CRM Online

Azure™ Services Platform

 Live Services

 Microsoft .NET Services

 Microsoft SQL Services

Microsoft SharePoint Services

Microsoft Dynamics CRM Services

 Windows® Azure™

Amazon Cloud

- Amazon cloud components
 - Elastic Compute Cloud (EC2)
 - Simple Storage Service (S3)
 - SimpleDB
- New Features
 - Availability zones
 - Place applications in multiple locations for failovers
 - Elastic IP addresses
 - Static IP addresses that can be dynamically remapped to point to different instances (not a DNS change)

Amazon Simple Storage Service (S3)

- Unlimited Storage.
- Pay for what you use:
 - \$0.20 per GByte of data transferred,
 - \$0.15 per GByte-Month for storage used,
 - Second Life Update:
 - 1TBytes, 40,000 downloads in 24 hours - \$200,



Utility Computing - EC2

- Amazon Elastic Compute Cloud (EC2):
 - Elastic, marshal 1 to 100+ PCs via WS,
 - Machine Specs...,
 - Fairly cheap!
- Powered by Xen – a Virtual Machine:
 - Different from Vmware and VPC as uses “para-virtualization” where the guest OS is modified to use special hyper-calls:
 - Hardware contributions by Intel (VT-x/Vanderpool) and AMD (AMD-V).
 - Supports “Live Migration” of a virtual machine between hosts.
- Linux, Windows, OpenSolaris
- Management Console/AP



EC2 - The Basics

- Load your image onto S3 and register it.
- Boot your image from the Web Service.
- Open up required ports for your image.
- Connect to your image through SSH.
- Execute you application...



Salesforce Cloud

- Started with **information management service** that could replace traditional business software technology
- Pioneered software-as-a-service market (esp. CRM tools)
- 5,000+ Public Sector and Nonprofit Customers use Salesforce Cloud Computing Solutions
- Moving beyond SaaS into the platform-as-a-service market



VMware Cloud (vCloud)

- Goal:
 - “**Federate resources** between internal IT and external clouds”
 - Application portability
 - **Elasticity and scalability**, disaster recovery, service level management
- vServices provide APIs and technologies



Case Study: IBM-Google Cloud

- Google and IBM plan to roll out a worldwide network of servers for a cloud computing infrastructure
- Initiatives for universities
- Architecture
 - Open source
 - Linux hosts
 - Xen virtualization (virtual machine monitor)
 - Apache Hadoop (file system)
 - “open-source software for reliable, scalable, distributed computing”



Facebook's Use of Open Source and Commodity Hardware

- 400 million users + 250,000 new users per day
- 100,000 transactions per second, 10,000+ servers
- Built on open source software
 - Web and App tier: Apache, PHP, AJAX
 - Middleware tier: Memcached (Open source caching)
 - Data tier: MySQL (Open source DB)
- Thousands of DB instances **store data in distributed fashion** (avoids collisions of many users accessing the same DB)



The Future

- Many of the activities loosely grouped together under cloud computing have already been happening and centralised computing activity is not a new phenomena
- However there are concerns that the mainstream adoption of cloud computing could cause many problems for users
- Many new open source systems appearing that you can install and run on your local cluster
 - should be able to run a variety of applications on these systems



优酷

优酷

——NHK情报翻译本部——

搬运: Jerome 听译: 青木瓜炖排骨
时间轴: 逝者善舞 校对: 与狼共舞
制作统括: 蓝旗营
www.veryed.com/groups/nhk

What Powers Cloud Computing in Google?

■ Commodity Hardware

- Performance: single machine not interesting
- Reliability
 - Most reliable hardware will still fail: fault-tolerant software needed
 - Fault-tolerant software enables use of commodity components
- Standardization: use standardized machines to run all kinds of applications

■ Infrastructure Software

- Distributed storage:
 - Distributed File System (GFS)
- Distributed semi-structured data system
 - BigTable
- Distributed data processing system
 - MapReduce



Open Source Cloud Software: Project Hadoop

- Google published papers on GFS('03), MapReduce('04) and BigTable('06)
- Project Hadoop
 - An open source project with the Apache Software Foundation
 - Implement Google's Cloud technologies in Java
 - HDFS(GFS) and Hadoop MapReduce are available.
Hbase(BigTable) is being developed
- Google is not directly involved in the development avoid conflict of interest



What is Hadoop?

- At Google MapReduce operation are run on a **special file system called Google File System (GFS)** that is highly optimized for this purpose.
- GFS is not open source.
- Doug Cutting and others at Yahoo! reverse engineered the GFS and called it Hadoop Distributed File System (HDFS).
- The software framework that supports HDFS, MapReduce and other related entities is called the project Hadoop or simply Hadoop.
- This is open source and distributed by Apache.

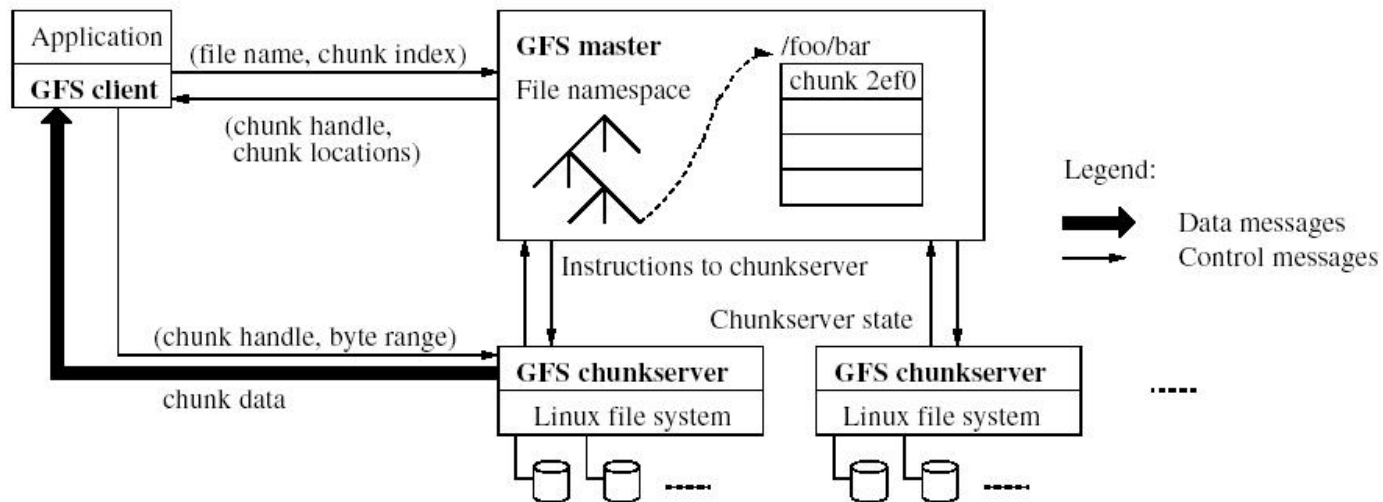


GFS Usage @ Google

- 200+ clusters
- Filesystem clusters of up to 5000+ machines
- Pools of 10000+ clients
- 5+ Petabyte Filesystems
- All in the presence of frequent HW failure

Google File System

- Files broken into chunks (typically 4 MB)
- Chunks replicated across three machines for safety (tunable)
- Data transfers happen directly between clients and chunkservers

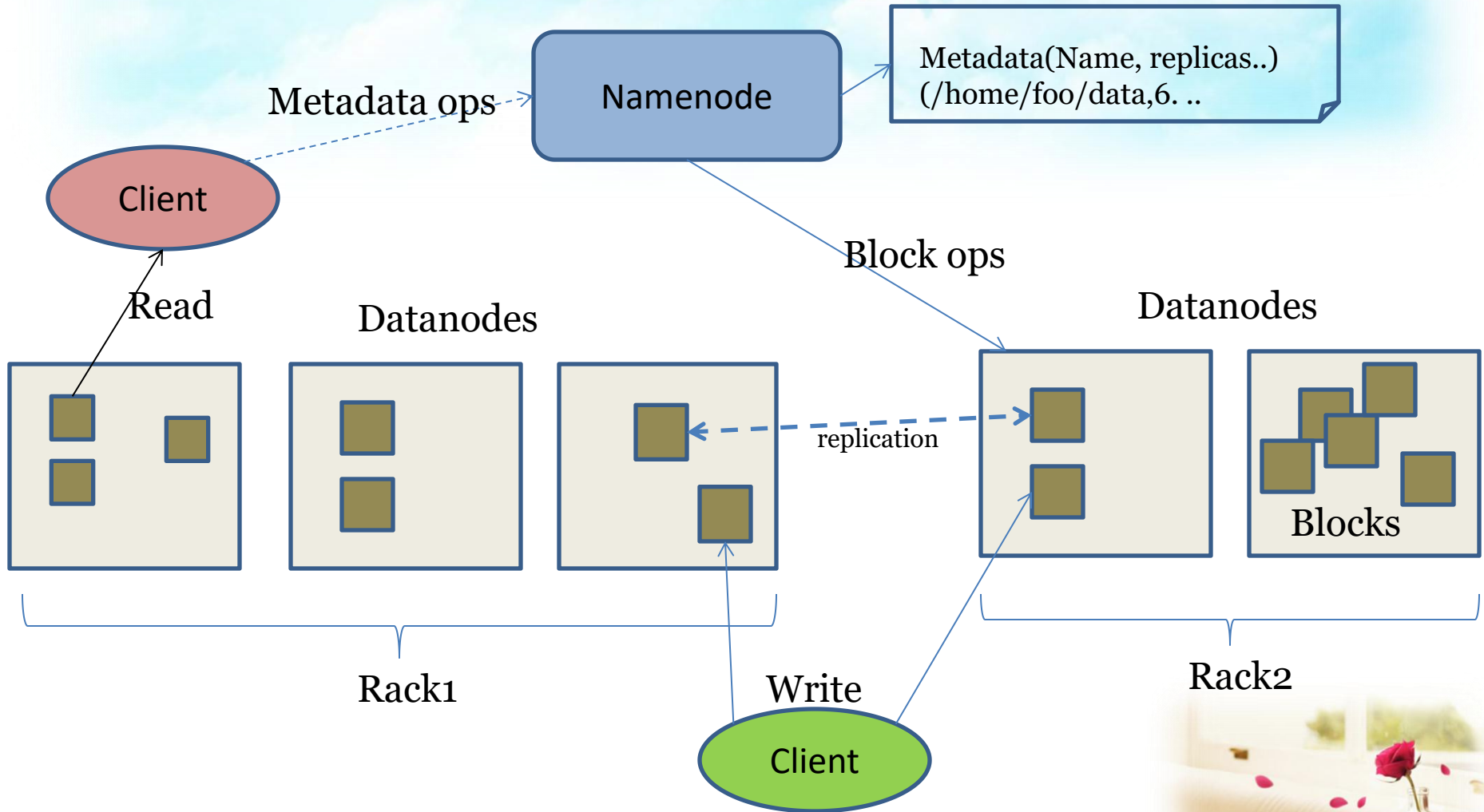


Fault tolerance

- Failure is the norm rather than exception
- A HDFS instance may consist of thousands of server machines, each storing part of the file system's data.
- Since we have huge number of components and that each component has non-trivial probability of failure means that there is always some component that is non-functional.
- **Detection of faults** and quick, **automatic recovery** from them is a core architectural goal of HDFS.

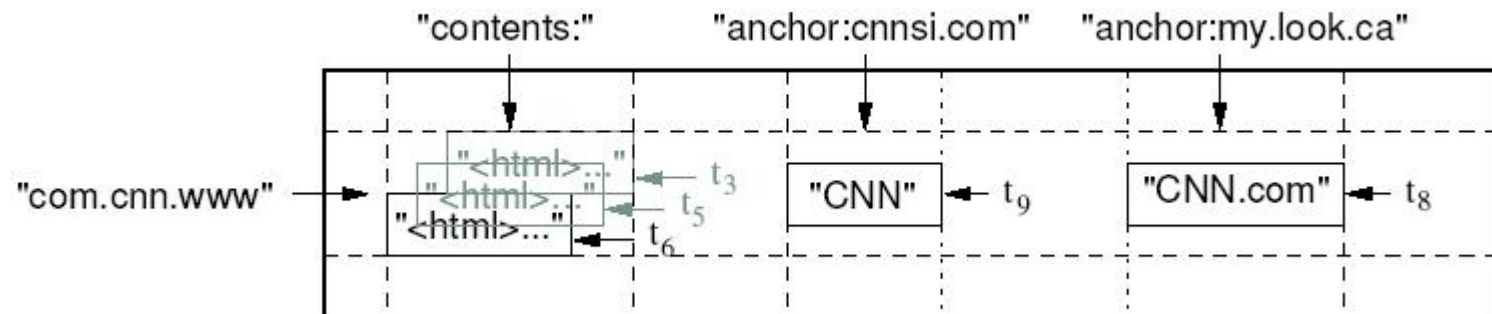


HDFS Architecture



BigTable

- Data model
 - (row, column, timestamp) \rightarrow cell contents



BigTable

- Distributed multi-level sparse map
 - Fault-tolerance, persistent
- Scalable
 - Thousand of servers
 - Terabytes of in-memory data
 - Petabytes of disk-based data
- Self-managing
 - Servers can be added/removed dynamically
 - Servers adjust to load imbalance

Why not just use commercial Database?

- Scale is too large or cost is too high for most commercial databases
- Low-level storage optimizations help performance significantly
 - Much harder to do when running on top of a database layer
 - Also fun and challenging to build large-scale systems



Distributed Data Processing

- Problem: How to count words in the text files?
 - Input files: N text files
 - Size: multiple physical disks
 - Processing phase 1: launch M processes
 - Input: N/M text files
 - Output: partial results of each word's count
 - Processing phase 2: merge M output files of step 1

```
Map(String input_key, String input_value):
```

```
    // input_key: document name
```

```
    // input_value: document contents
```

```
    for each word w in input_values:
```

```
        EmitIntermediate(w, "1");
```

```
Reduce(String key, Iterator intermediate_values):
```

```
    // key: a word, same for input and output
```

```
    // intermediate_values: a list of counts
```

```
    int result = 0;
```

```
    for each v in intermediate_values:
```

```
        result += ParseInt(v);
```

```
    Emit(result);
```



Technical issues

- File management: where to store files?
 - Store all files on the same file server → Bottleneck
 - Distributed file system: opportunity to run locally
- Granularity: how to decide N and M ?
- Job allocation: assign which task to which node?
 - Prefer local job: knowledge of file system
- Fault-recovery: what if a node crashes?
 - Redundancy of data
 - Crash-detection and job re-allocation necessary

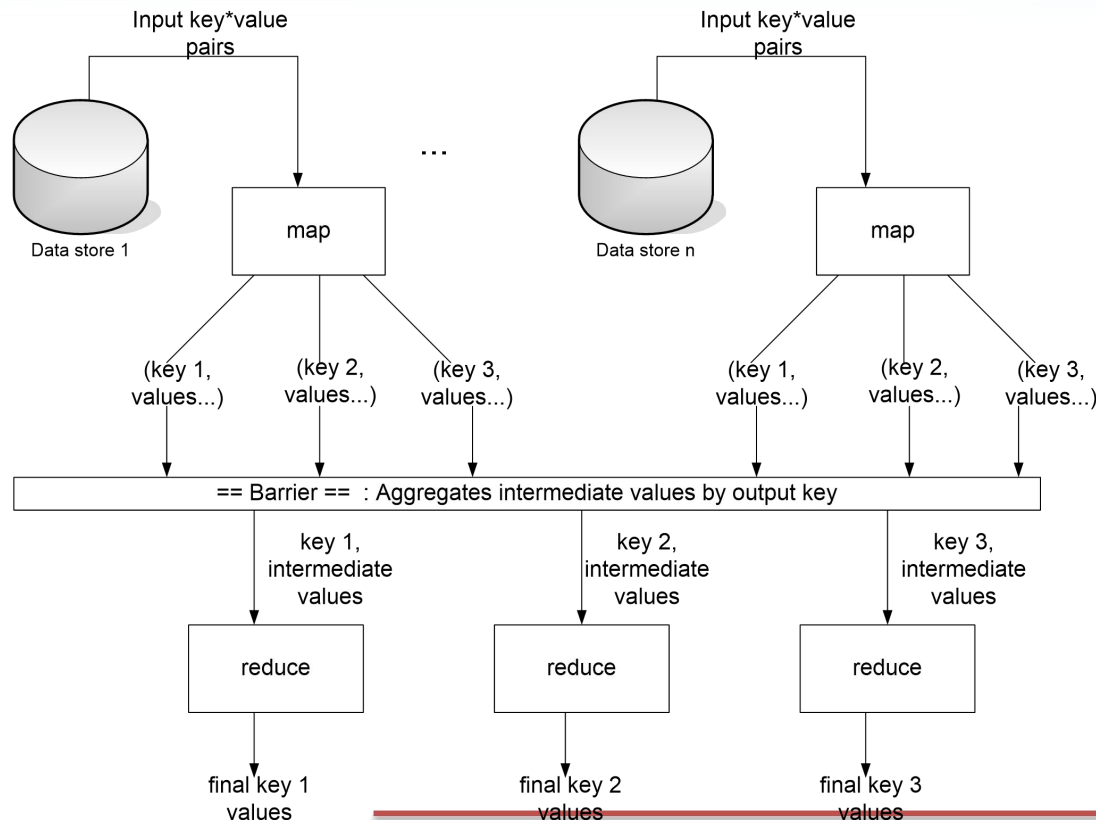
MapReduce

- A simple programming model that applies to many data-intensive computing problems
- Hide messy details in MapReduce runtime library
 - Automatic parallelization
 - Load balancing
 - Network and disk transfer optimization
 - Handle of machine failures
 - Robustness
 - Easy to use



MapReduce - A New Model and System

- Two phases of data processing
 - Map: $(in_key, in_value) \rightarrow \{(key_j, value_j) \mid j = 1 \dots k\}$
 - Reduce: $(key, [value_1, \dots, value_m]) \rightarrow (key, f_value)$



Example - WordCount (1/2)

- Input is files with one document per record
- Specify a map function that takes a key/value pair
 - key = document URL
 - Value = document contents
- Output of map function is key/value pairs. In our case, output (w,"1") once per word in the document

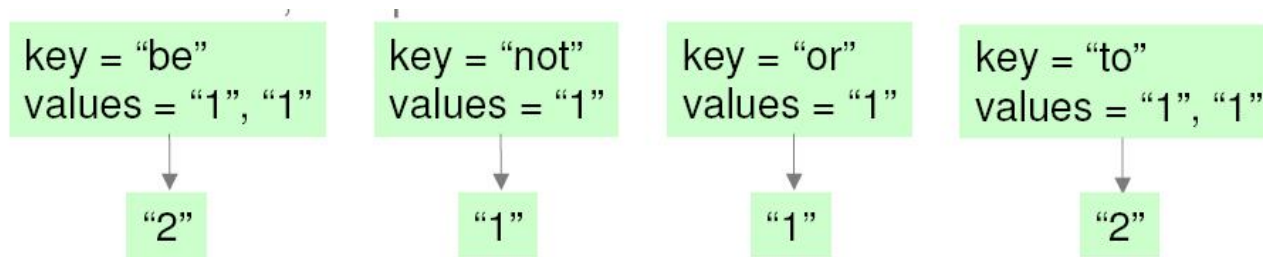
"document1", "to be or not to be"



"to", "1"
"be", "1"
"or", "1"
...

Example - WordCount (2/2)

- MapReduce library **gathers together all pairs with the same key**(shuffle/sort)
- The reduce function combines the values for a key. In our case, compute the sum



- Output of reduce paired with key and saved

```
"be", "2"  
"not", "1"  
"or", "1"  
"to", "2"
```

6.2 Security Issues



Information Security

- Achieving information security in an electronic society requires a vast array of **technical and legal skills**. There is, however, no guarantee that all of the information security objectives deemed necessary can be adequately met. The technical means can be provided through cryptography, etc.

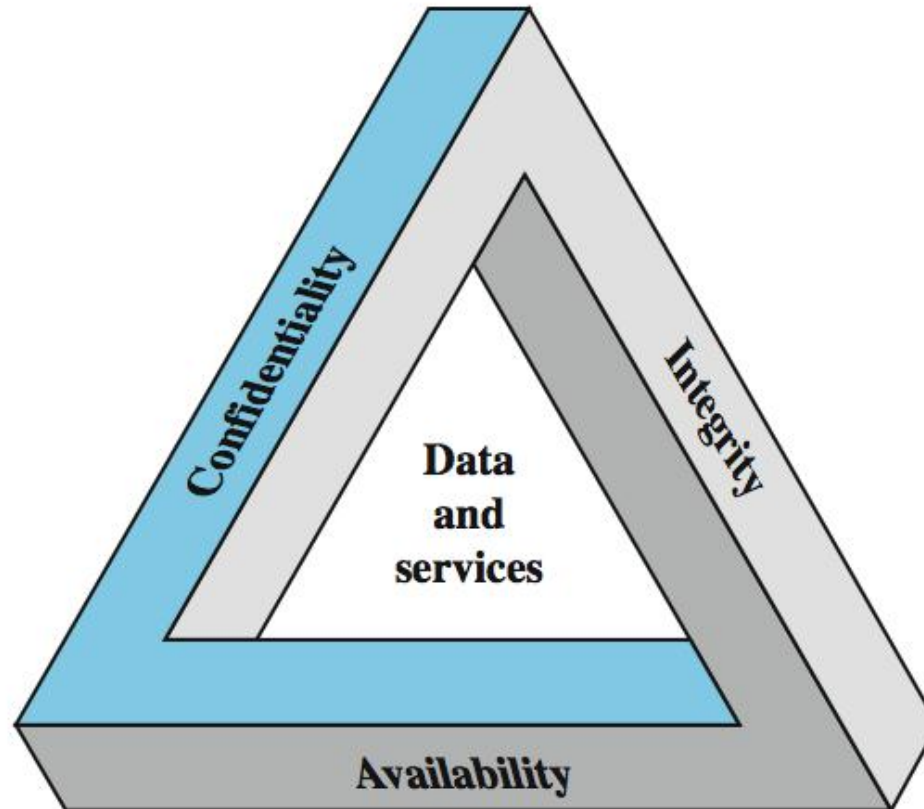


Computer Security

- Is defined as the protection afforded to an automated information system in order to attain the applicable objectives of preserving the **integrity**, **availability** and **confidentiality** of information system resources (includes hardware, software, firmware, information/data, and telecommunications)



CIA Triad



Key Objectives

■ Confidentiality

- Concealment of information or resources
- **Data Confidentiality**-information not disclosed to unauthorized individuals
- **Privacy**– individuals control how their information is collected, stored, shared

■ Integrity

- Trustworthiness of data or resources
- **Data Integrity**
- **System Integrity**

■ Availability

- Service not denied to authorized users
- Ability to use information or resources



Confidentiality

- Need for **keeping information secret** arises from use of computers in sensitive fields such as government and industry
- **Access mechanisms**, such as **cryptography**, support confidentiality
 - Example: encrypting income tax return
- Lost through **unauthorized disclosure** of information



Integrity

- Often requires **preventing unauthorized changes**
- Includes data integrity (content) and origin integrity (source of data also called **authentication**)
- Include **prevention mechanisms and detection mechanisms**
 - Example: Newspaper prints info leaked from White House and gives wrong source
- Includes both **correctness** and **trustworthiness**
- Lost through **unauthorized modification or destruction of information**



Availability

- Is an aspect of reliability and system design
- Attempts to block availability, called **denial of service attacks (DoS)** are difficult to detect
 - Example: bank with two servers –one is blocked, the other provides false information
- Ensures timely and reliable access to and use of information
- Lost through **disruption of access** to information or information system



Authenticity and Accountability

Two additional objectives:

- **Authenticity** - being genuine and able to be verified or trust; verifying that users are who they say they are
- **Accountability** -actions of an entity can be traced uniquely to that entity; supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention.



Levels of Impact

- We can define 3 levels of impact from a security breach:
 - Low
 - Moderate
 - High



Security Breach Low Impact

- Loss has **limited adverse effect**
- For example:
 - Effectiveness of the **functions** of an organization are noticeably reduced
 - Results in minor damage to organizational **assets**
 - Results in minor **financial** loss
 - Results in minor **harm** to individuals



Security Breach Moderate Impact

- Loss may have **serious adverse effect** on organizational operations, assets or individuals.
- For example:
 - Effectiveness of the **functions** of an organization are significantly reduced
 - Results in significant damage to organizational **assets**
 - Results in significant **financial** loss
 - Results in significant **harm** to individuals



Security Breach High Impact

- Loss is expected to have **severe or catastrophic adverse effect** on organizational operations, assets or individuals.
- For example:
 - Effectiveness of the **functions** of an organization are reduced so that the organization cannot perform its primary function(s).
 - Results in major damage to organizational **assets**
 - Results in major **financial** loss
 - Results in severe or catastrophic **harm** to individuals, involving loss of life or serious life-threatening injuries



Examples of Security Requirements

- **Confidentiality** – student grades
 - **High confidentiality** – grades
 - Only available to students, parents and employees (who need it to do their job)
 - **Moderate confidentiality** – enrollment
 - **Low confidentiality** – **Directory information**
 - Lists of departments, faculty, students
 - Available to the public
 - Often published on Web site



Examples of Security Requirements

- **Integrity**- patient information
 - **High requirement for integrity**
 - Medical database, if falsified or inaccurate, could cause harm (allergies, etc.)
 - **Medium requirement for integrity**
 - Web site that offers a forum for discussion of medical topics, not for research
 - **Low requirement for integrity**
 - Anonymous poll (such as a patient satisfaction)



Examples of Security Requirements

Availability - The more critical a component or service is, the higher the level of availability required:

- **High availability- authentication service**
 - Interruption of service results in being unable to access computing resources
- **Moderate availability- College web site**
 - Provides information but is not critical
- **Low availability- online phone directory**
 - Other sources of information are available



The Need for Security

- **Computer Security** - the collection of tools designed
 - to protect data and
 - to thwart hackers
- **Network security or internet security-** security measures needed to protect data during their transmission



Security

- Motivation: **Why do we need security?**
- **Increased reliance on Information technology** with or without the use of networks
- The use of IT has changed our lives drastically.
- We depend on E-mail, Internet banking, and several other governmental activities that use IT
- Increased use of E-Commerce and the world wide web on the Internet as a vast repository of various kinds of information (immigration databases, flight tickets, stock markets etc.)

Security Concerns

- Damage to any IT-based system or activity can result in **severe disruption of services and losses**.
- **Systems connected by networks are more prone to attacks** and also suffer more as a result of the attacks than stand-alone systems.
- Concerns such as the following are common
 - How do I know the party I am talking on the network is *really* the one I want to talk?
 - How can I be assured that no one else is listening and learning the data that I send over a network
 - Can I ever stay relaxed that no hacker can enter my network and play havoc?



Concerns continued...

- Is the web site I am downloading information from a legitimate one, or a fake?
- How do I ensure that the person I just did a financial transaction denies having done it tomorrow or at a later time?
- I want to buy some thing online, but I don't want to let them charge my credit card before they deliver the product to me.



That is why...

- ..we need security
 - To safeguard the **confidentiality, integrity, authenticity and availability** of data transmitted over insecure networks
 - Internet is not the only insecure network in this world
 - Many internal networks in organizations are prone to insider attacks



https://

(V.Shmatikov)

Wells Fargo Account Summary - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Favorites Print Home

Address https://online.wellsfargo.com/mn1_aa1_on/cgi-bin/session.cgi?sessargs=coAn76ax52xltPX8uoCT8rRBfMMdJldx Go Links Yahoo maps Mapblast Dictionary

Home | Help Center | Contact Us | Locations | Site Map | Apply | **Sign Off**

WELLS FARGO

Account Summary Last Log On: January 06, 2004

> Account Summary

Brokerage

Bill Pay

Transfer

Account Services

My Message Center

Stay organized with FREE 24/7 access to Online Statements. Sign up today.

Sign up for the Wells Fargo Rewards® program and get 2,500 points. Learn More.

Wells Fargo Accounts **OneLook Accounts**

Tip: Select an account's balance to access the Account History.

NEW [Enroll for Online Statements](#) [My Message Center](#)

Cash Accounts

Account	Account Number	Available Balance
Checking Add Bill Pay		
Total		

To end your session, be sure to Sign Off.

Account Summary | Brokerage | Bill Pay | Transfer | My Message Center | Sign Off

Home | Help Center | Contact Us | Locations | Site Map | Apply

© 1995 - 2003 Wells Fargo. All rights reserved.

However, in reality

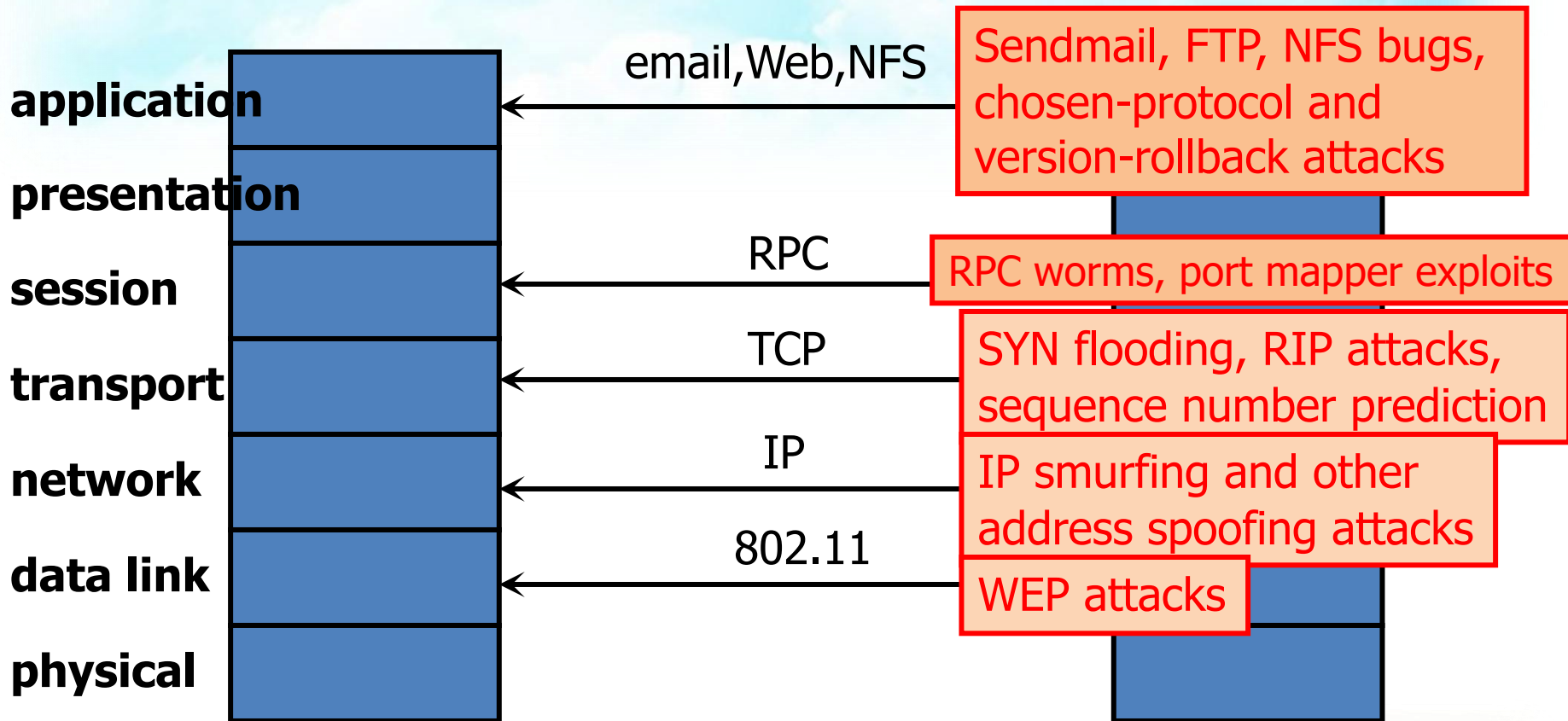
- Security is often **over looked** (not one of the top criteria)
- Availability, efficiency and performance tend to be the ones
- **Buggy implementations**
- Systems too complex in nature and rich in features can be filled with security holes.
- Incorporation of security into networks, not growing with the rapidly growing number and size of networks.
- **Attacking** is becoming so common and easy – there are books clearly explaining how to launch them
- Security and attacks are a **perpetual cat-and-mouse play**. The only way to avoid attacks is to keep up-to-date with latest trends and stay ahead of malicious netizens

OSI Security Architecture

- International Telecommunications Union (ITU) is a United Nations sponsored agency that develops standards relating to telecommunications and to Open system Interconnection (**OSI**)
- OSI Model: 7 Layer Model
- Describes the protocols and details of transmitting data at each layer.



OSI Network Stack and Attacks



Only as secure as the single weakest layer...

OSI Security Architecture

- **ITU-T Recommendation X.800 Security Architecture for OSI** which defines a systematic approach to assessing and providing security.
- The OSI security architecture focuses on security **attacks, mechanisms and services.**

Cryptography

- Definition **Cryptography** is the study of mathematical techniques related to aspects of information security such as **confidentiality, data integrity, entity authentication, and data origin authentication.**

Cryptographic Goals

- **Confidentiality** is a service used to keep the content of information from all but those authorized to have it.
- **Secrecy** is a term synonymous with confidentiality and privacy.
- There are numerous approaches to providing confidentiality, ranging from **physical protection** to **mathematical algorithms** which render data unintelligible.



Cryptographic Goals

- **Data integrity** is a service which addresses the **unauthorized alteration** of data. To assure data integrity, one must have the ability to detect data manipulation by unauthorized parties. Data manipulation includes such things as **insertion, deletion, and substitution.**

Cryptographic Goals

- **Authentication** is a service related to identification. This function applies to **both entities and information** itself.
- Two parties entering into a communication should identify each other.
- Information delivered over a channel should be authenticated as to **origin, date of origin, data content, time sent**, etc.
- For these reasons this aspect of cryptography is usually subdivided into two major classes: **entity authentication and data origin authentication**. Data origin authentication implicitly provides data integrity

Cryptographic Goals

- **Non-repudiation** is a service which prevents an entity from denying previous commitments or actions. When disputes arise due to an entity denying that certain actions were taken, a means to resolve the situation is necessary.
- For example, one entity may authorize the purchase of property by another entity and later deny such authorization was granted. A procedure involving a trusted third party is needed to resolve the dispute.



Aspects of Security

- Consider 3 aspects of information security:
 - **security attack**
 - **security mechanism**
 - **security service**

- Note terms:
 - *threat*
 - *attack*

Attacks, Services and Mechanisms

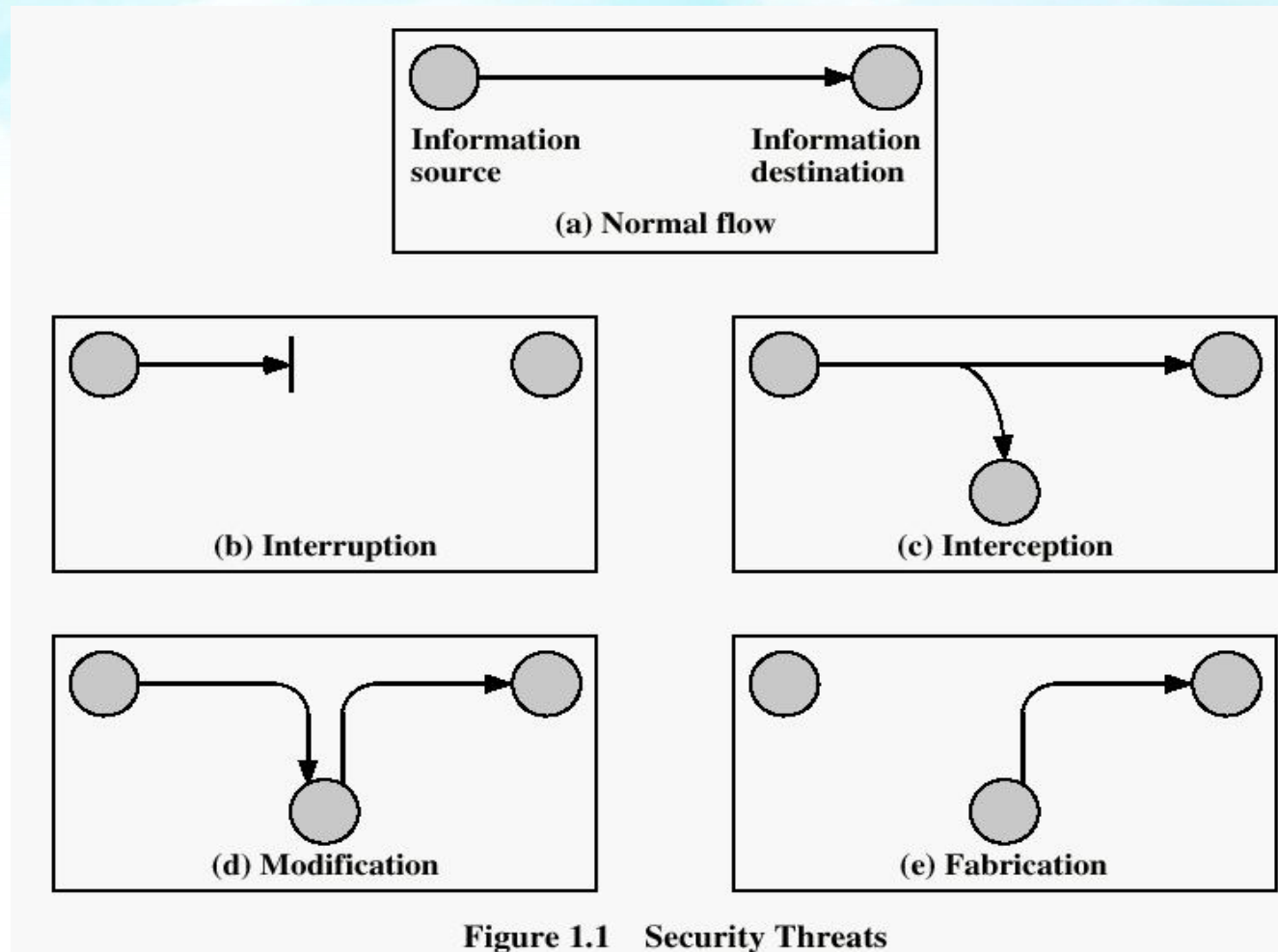
- **Security Attack:** Any action (active or passive) that compromises the security of information.
- **Security Mechanism:** A mechanism that is designed to detect, prevent, or recover from a security attack.
- **Security Service:** A service that enhances the security of data processing systems and information transfers. A security service makes use of one or more security mechanisms.



Threats and Attacks

- **Threat** - a potential for violation of security or a possible danger that might exploit a vulnerability.
- **Attack** - an assault on system security- an intelligent act that is a deliberate attempt to evade security services and violate the security policy of a system.

Security Threats/Attacks



Security Attacks

- **Interruption:** This is an attack on **availability**
 - Disrupting traffic
 - Physically breaking communication line
- **Interception:** This is an attack on **confidentiality**
 - Overhearing, eavesdropping over a communication line



Security Attacks

- **Modification:** This is an attack on **integrity**
 - Corrupting transmitted data or tampering with it before it reaches its destination
- **Fabrication:** This is an attack on **authenticity**
 - Faking data as if it were created by a legitimate and authentic party



Threats

- **Disclosure** – unauthorized access to information
- **Deception** – acceptance of false data
- **Disruption**- interruption or prevention of correct operation
- **Usurpation**- unauthorized control of some part of a system



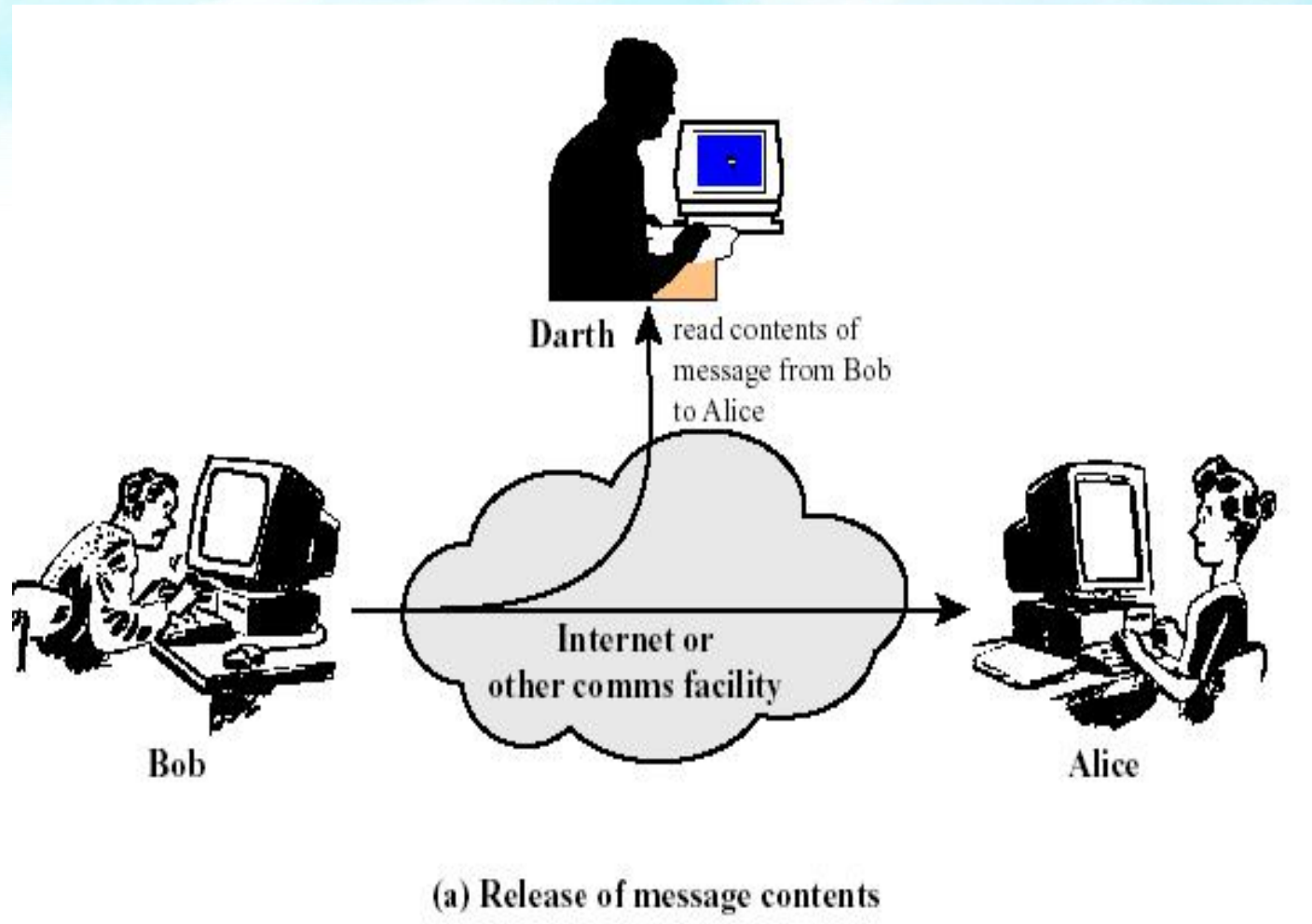
Passive and Active Attacks

- Security attacks are usually classified as passive or active:
- **Passive**- attempts to **learn or make use of information** from the system, but does not affect system resources.
- **Active**- attempts to **alter system resources or affect their operation**.

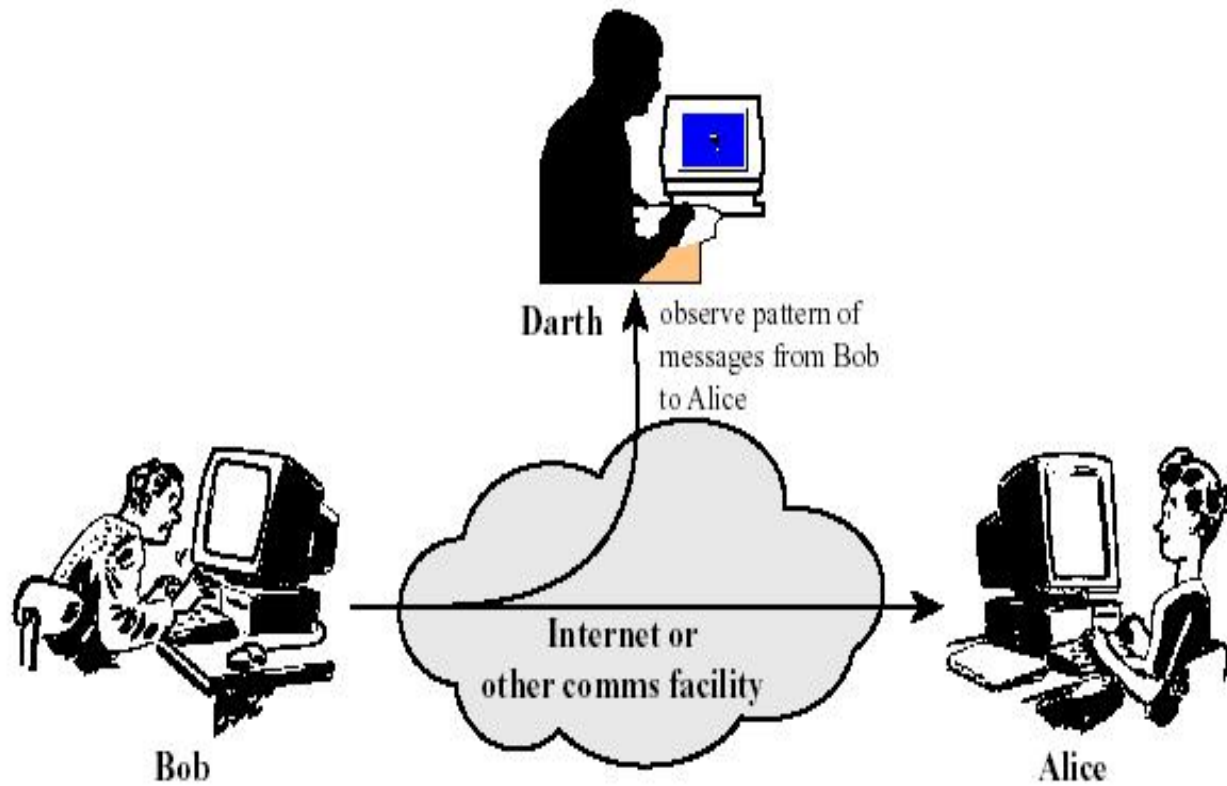
Passive and active attacks

- **Passive attacks-** goal to obtain information
 - No modification of content or fabrication
 - Eavesdropping to learn contents or other information (transfer patterns, traffic flows etc.)
 - Release of message contents
 - Traffic analysis
- **Active attacks-** modification of content and/or participation in communication to
 - Impersonate legitimate parties (Masquerade)
 - Replay or retransmit
 - Modify the content in transit
 - Launch denial of service attacks

Passive Attacks

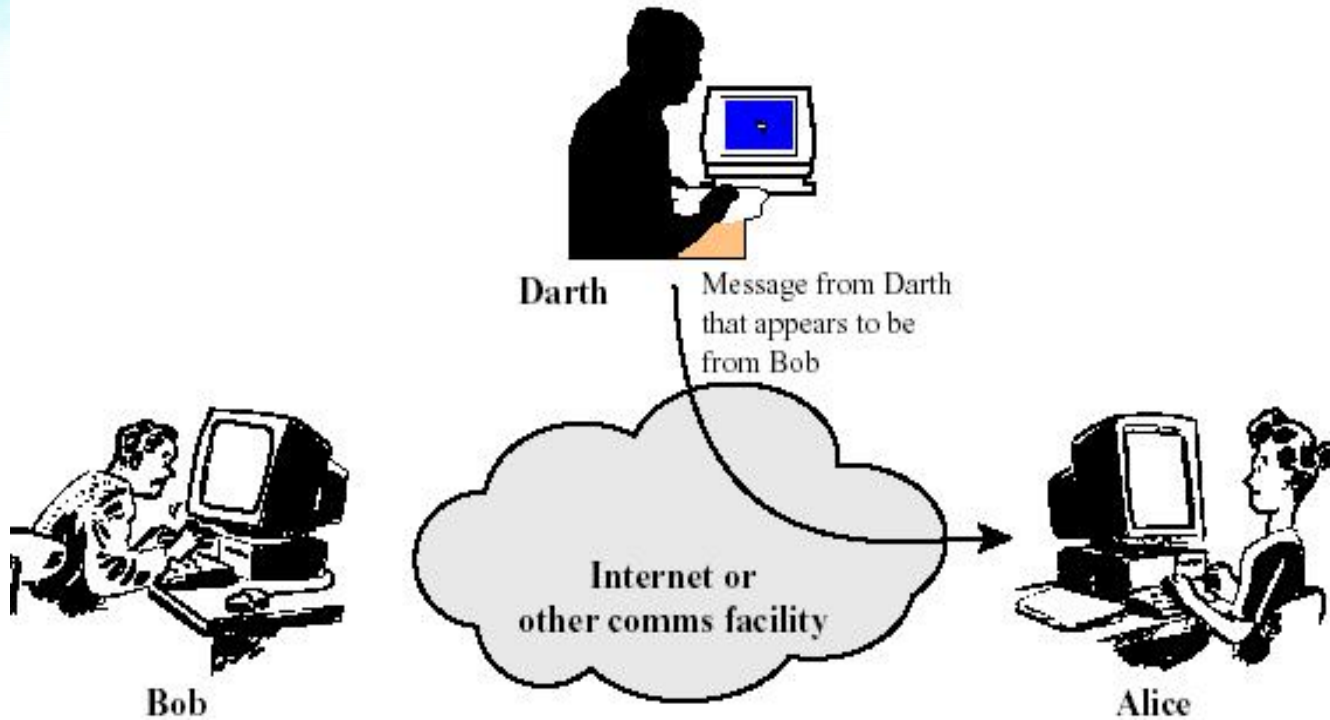


Passive Attacks



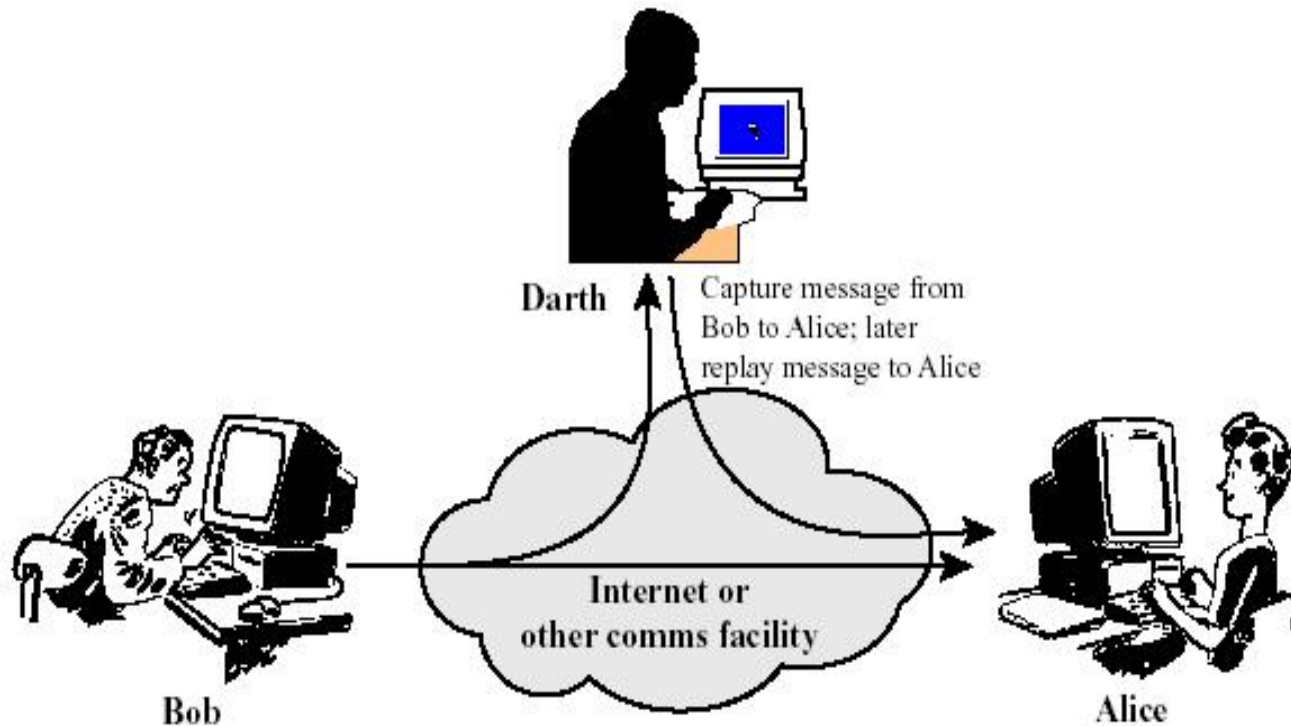
(b) Traffic analysis

Active Attacks



(a) Masquerade

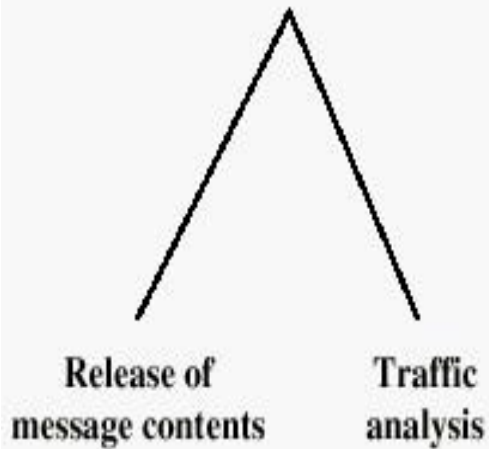
Active Attacks



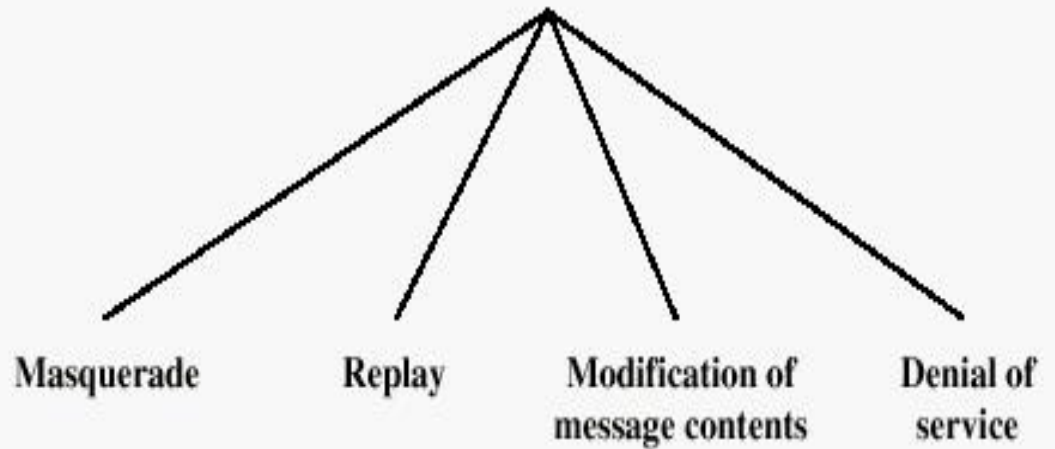
(b) Replay

Summary of Passive and Active Threats

Passive Threats



Active Threats



Security Models

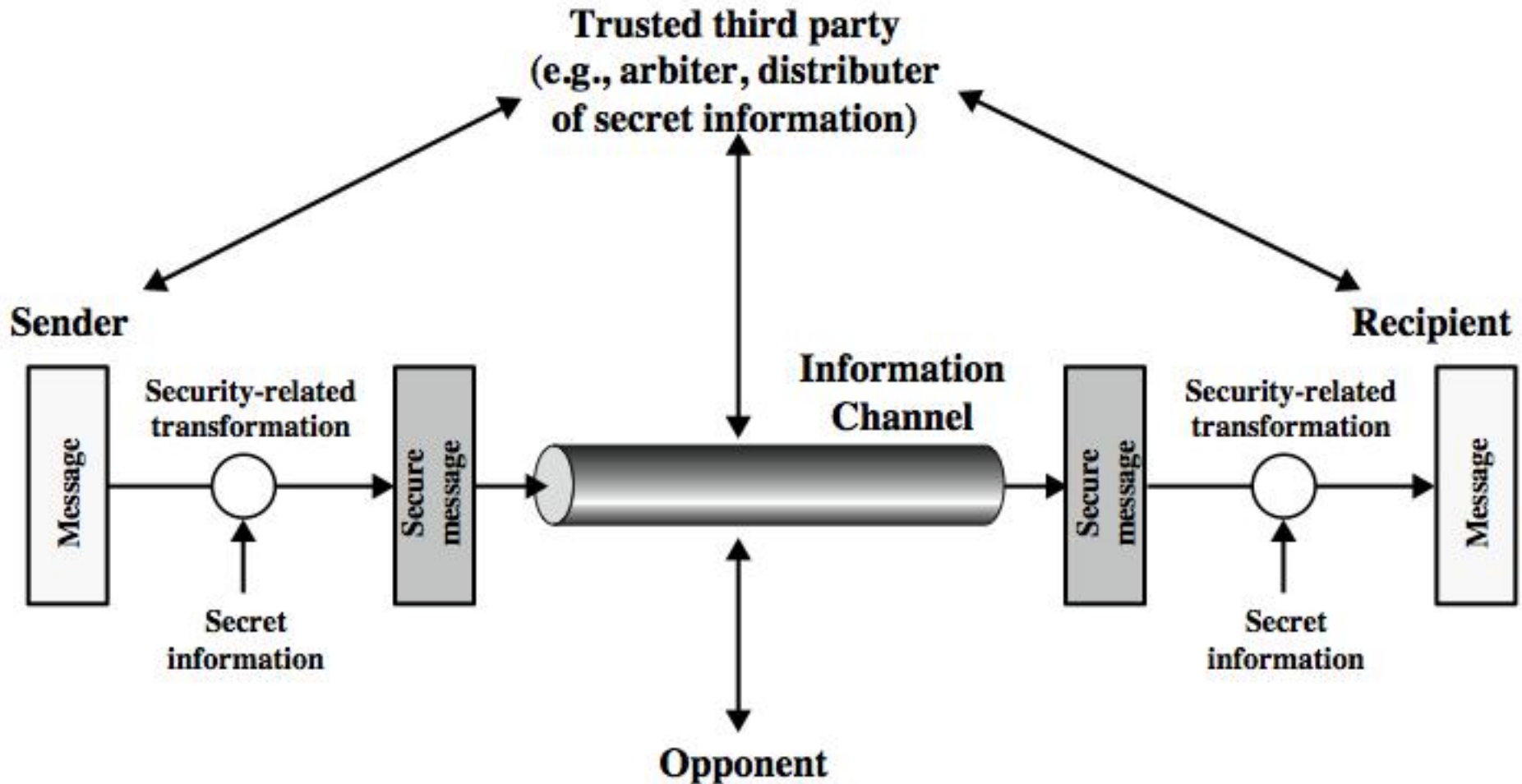
- A Network Access Security Model reflects the concern for protecting an information system from unwanted access, for example by hackers or malware (malicious programs).

Model for Network Security

- Models information flowing over an insecure communications channel, in the presence of possible opponents. Hence an appropriate **security transform (encryption algorithm)** can be used, with suitable **keys**, possibly negotiated using the presence of a **trusted third party**.



Model for Network Security

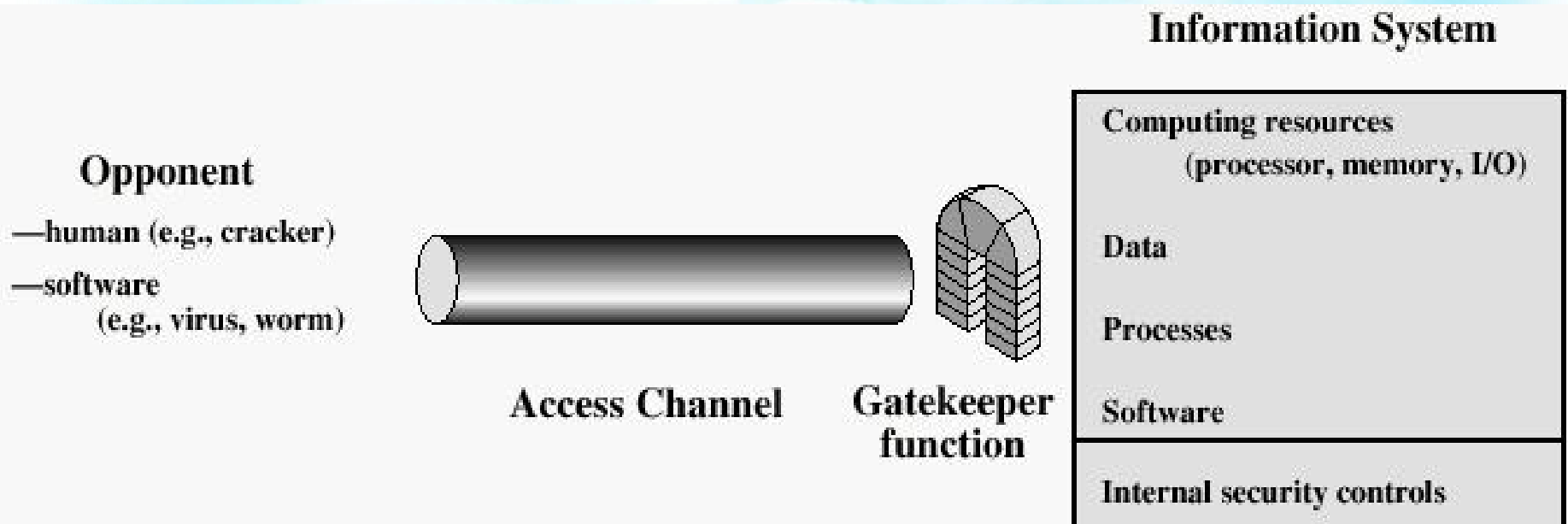


Model for Network Security

Using this model requires us to:

1. design a suitable **algorithm** for the security transformation
2. **generate** the secret information (**keys**) used by the algorithm
3. develop methods to **distribute and share the secret information**
4. specify a protocol enabling the principals to use the **transformation and secret information** for a security service

General Security Access Model

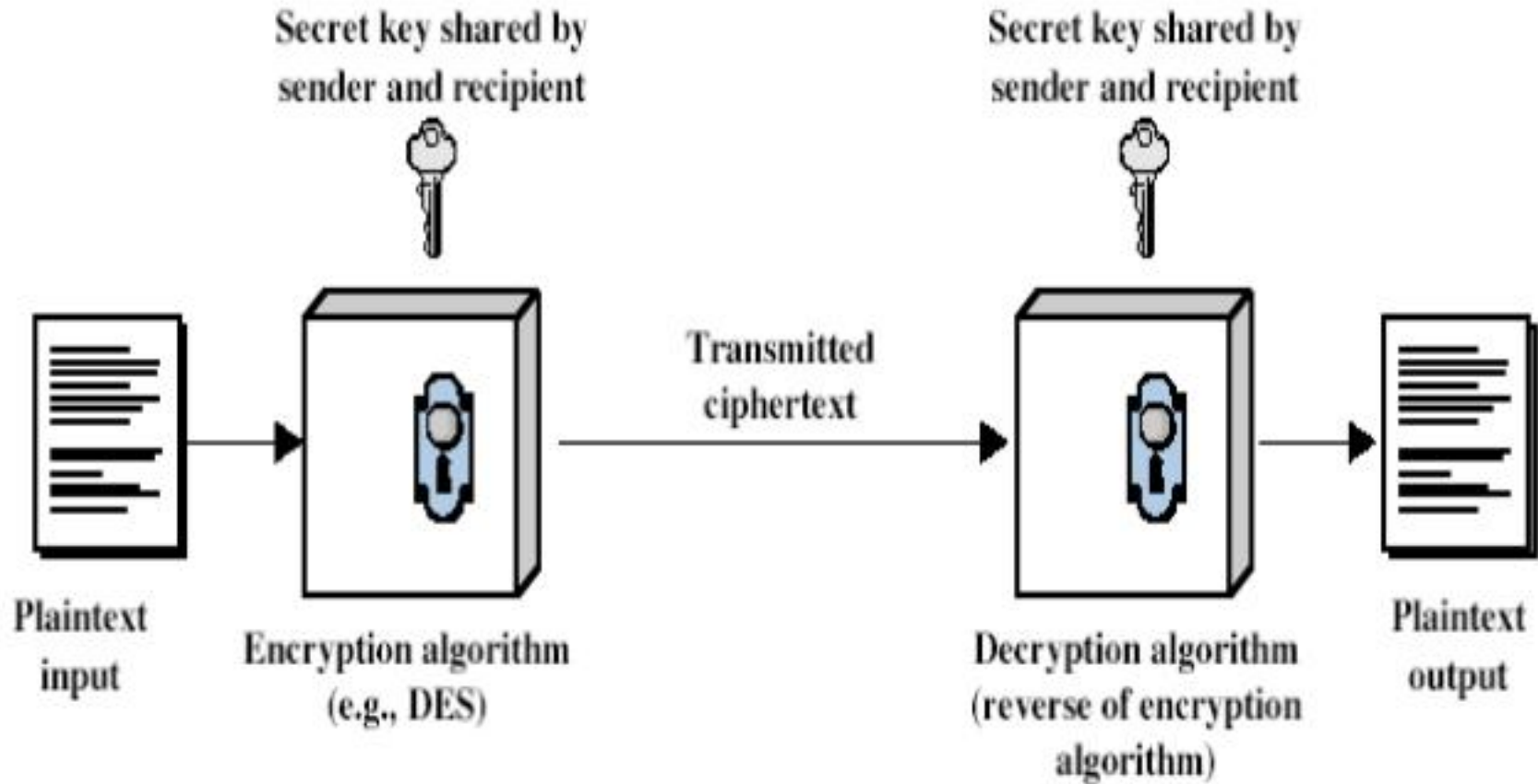


Model for Network Access Security

- Security mechanisms for controlling unwanted access fall into two categories.
- Using this model requires us to:
 1. select appropriate **gatekeeper** functions to identify users (for example, password-based login procedures) .
 2. implement security controls to ensure only **authorised users access designated information** or resources (for example, monitor activities and analyze stored information to detect the presence of intruders).



Symmetric Cipher Model



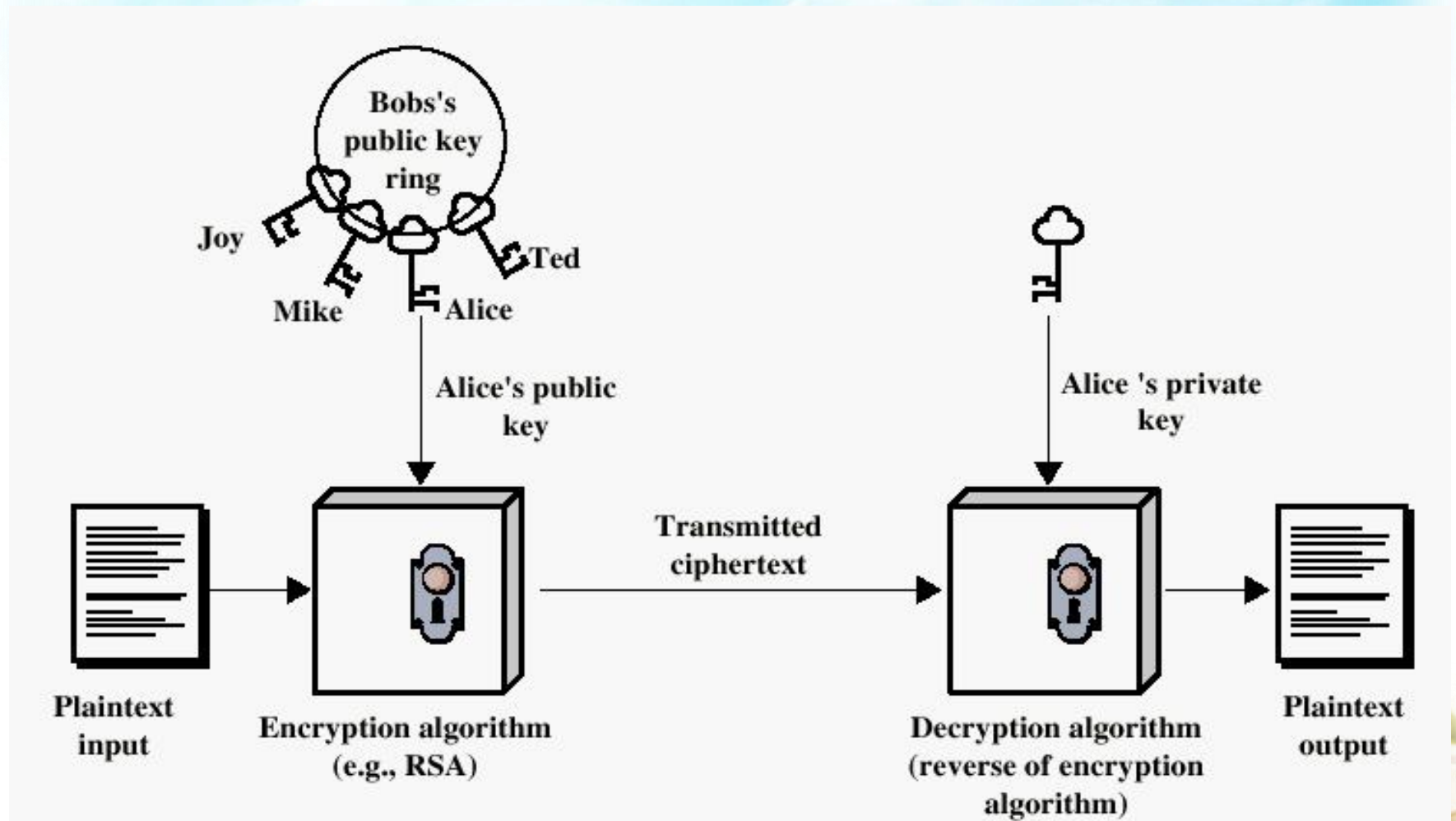
Public-Key Cryptography Principles

- The use of two keys has consequences in: key distribution, confidentiality and authentication.
- The scheme has six ingredients
 - Plaintext
 - Encryption algorithm
 - Public and private key
 - Ciphertext
 - Decryption algorithm



Encryption using Public-Key system

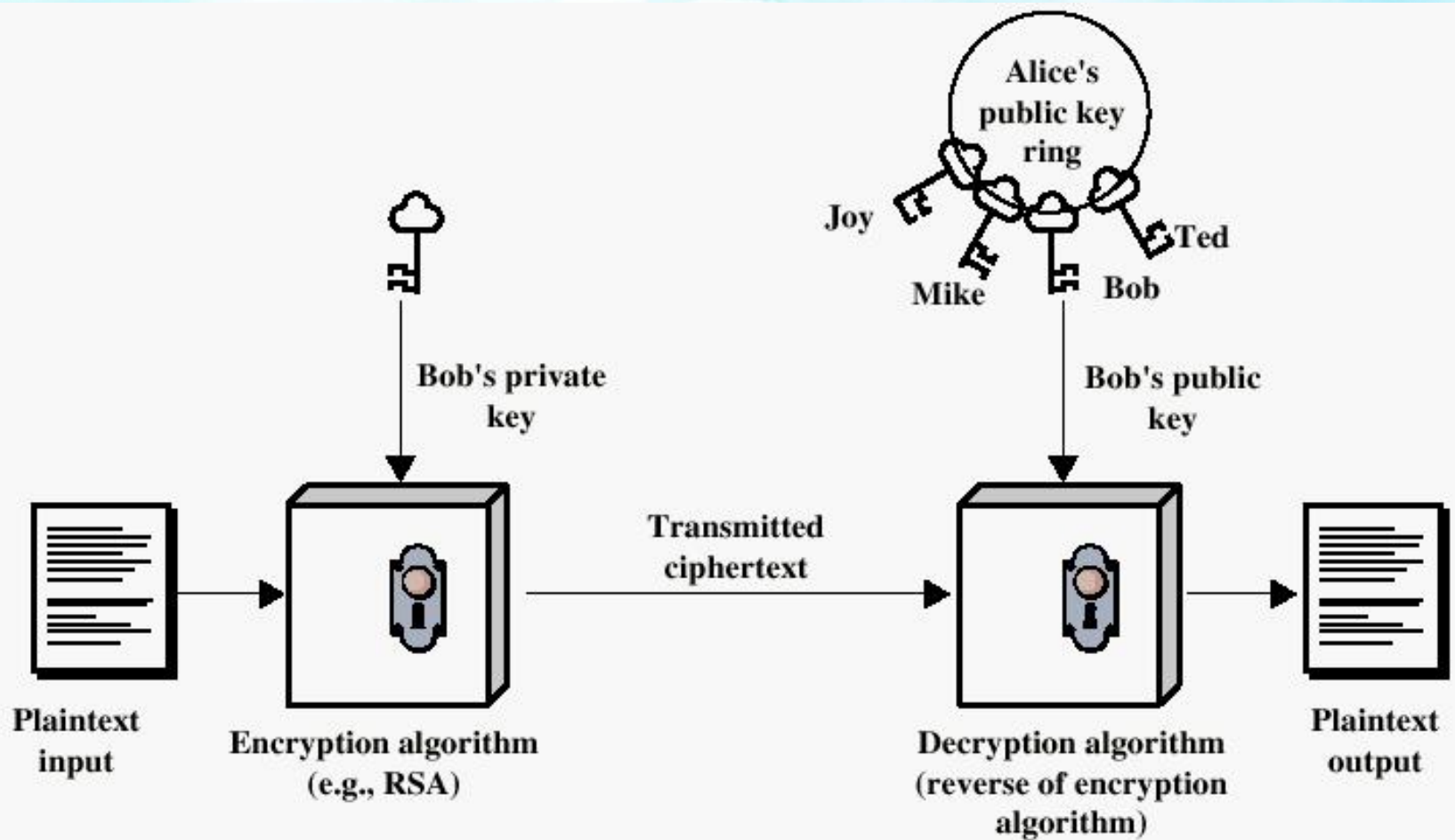
Bob -----> data -----> Alice



Authentication using Public-Key System

Bob

Alice



Applications for Public-Key Cryptosystems

- Three categories:
 - **Encryption/decryption:** The sender encrypts a message with the recipient's public key.
 - **Digital signature:** The sender "signs" a message with its private key.
 - **Key exchange:** Two sides cooperate to exchange a session key.

Symmetric key vs. Public-key

Advantages of symmetric key:

1. It can be designed for high rates of data throughput, may be using hardware implementations
2. Key lengths are relatively short
3. Can be used to produce stronger ciphers

Symmetric key vs. Public-key

Disadvantages of symmetric key:

1. Key must **remain secret at both ends**
2. In a large network, there are many key pairs to be managed. Effective key management requires use of an **unconditionally trusted third party**.
3. Digital signature schemes using private key cryptography requires large key.

Symmetric key vs. Public-key

Advantages of public key cryptography:

1. Only the private key to be kept secret
2. The administration of key requires only a functionally trusted TTP.
3. A private/public key pair may remain unchanged for a long time.
4. Gives relatively efficient digital signature schemes

Symmetric key vs. Public-key

Disadvantages of public-key cryptography:

1. Several orders of magnitudes slower
2. Key sizes are larger.
3. No public-key cryptosystem is proven to secure.