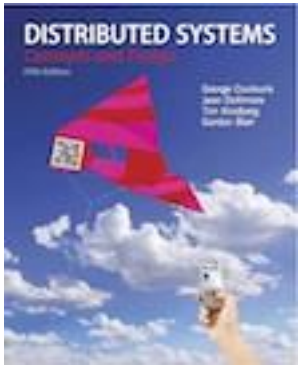
UNIT-I

Characterization of Distributed Systems



Distributed Systems:

Concepts and Design

By

Coulouris, Dollimore, Kindberg and Blair

1.1 Introduction:

- ✓ Network everywhere.(personal n/w, factory n/w, campus n/w, corporate n/w, mobile n/w, home n/w, in-car n/w and internet and so on..)

- ✓ **Distributed system** is the one in which hardware or software components located at networked computers communicate and coordinate their actions only by passing messages.

Characteristics of Distributed Systems are,

- ✓ ***Concurrency*:** Concurrent programs execution is the norm. I can do my work on my computer while you do your work on yours, sharing resources such as web pages or files when necessary. Resources can be added or removes easily.
- ✓ **No global Clock:** Close coordination often depends on a shared idea of the time at which the programs' actions occur. But it turns out that there are limits to the accuracy with which the computers in a network can synchronize their clocks – there is no single global notion of the correct time.
- ✓ **Independent Failure:** Faults in the network result in the **isolation** of the computers that are connected to it, but that doesn't mean that they stop running.

Application Areas of Distributed Systems: (i.e. where we use?)

Finance and commerce	eCommerce e.g. Amazon and eBay, PayPal, online banking and trading
The information society	Web information and search engines, ebooks, Wikipedia; social networking: Facebook and MySpace.
Creative industries and entertainment	online gaming, music and film in the home, user-generated content, e.g. YouTube, Flickr
Healthcare	health informatics, on online patient records, monitoring patients
Education	e-learning, virtual learning environments; distance learning
Transport and logistics	GPS in route finding systems, map services: Google Maps, Google Earth
Science	The Grid as an enabling technology for collaboration between scientists
Environmental management	sensor technology to monitor earthquakes, floods or tsunamis

1.2 Examples of Distributed Systems:

Typical examples of Distributed systems are,

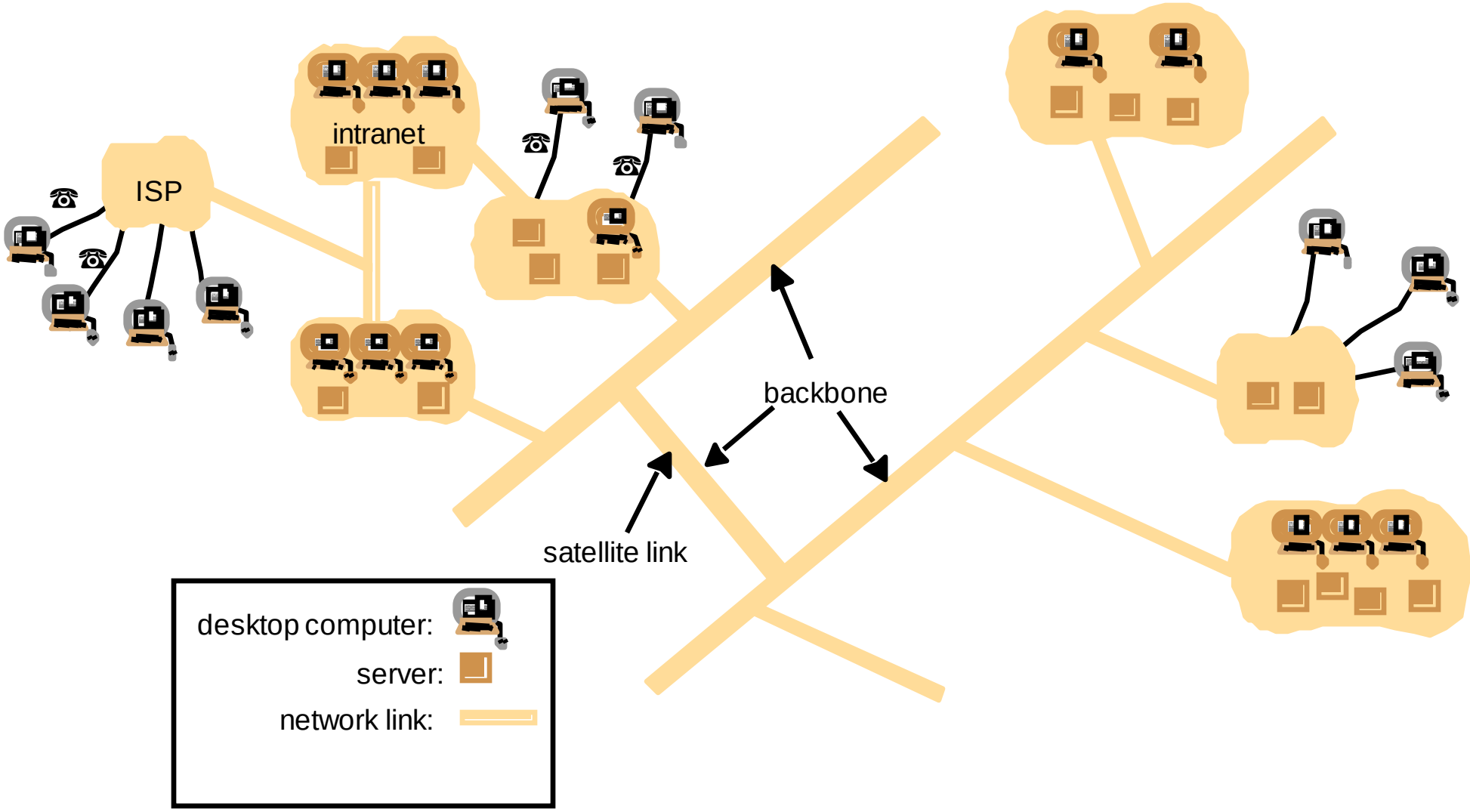
- **The Internet**
- **Intranets**
- **Mobile and Ubiquitous computing.**

The Internet:

✓ Internet is a very large distributed system. It enables users, wherever they are, to make use of services like www, email, file transfer. The set of services is open-ended.

✓ Refer figure below.

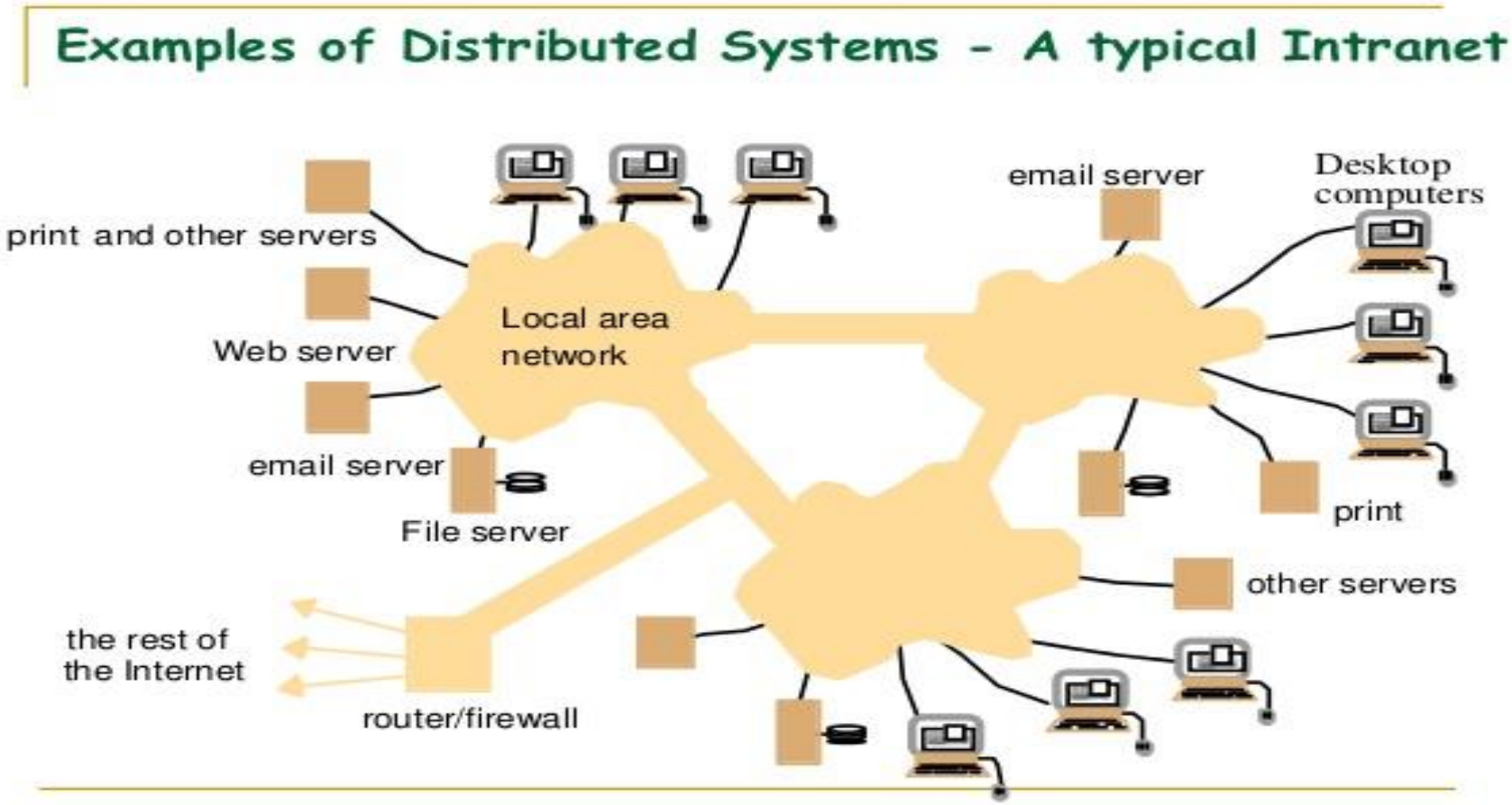
Figure . A typical portion of the Internet



Intranets:

- ✓ An intranet is a portion of the internet that is separately administered and has a boundary that can be configured to enforce local security policies.
- ✓ It may be composed of several LANs linked by backbone connections.
- ✓ The n/w configuration of a particular intranet is the responsibility of the organization that administers it.
- ✓ An intranet is connected to the Internet via router, which allows the users to use the services available in the Internet.
- ✓ Firewall is used to protect intranet by preventing unauthorized messages leaving or entering.

A typical Intranet



Intranets: contd..

- ✓ Some organizations do not wish to connect their internal networks to the Internet at all.
- ✓ E.g. police and other security and law enforcement agencies are likely to have at least some internal networks that are isolated from outside world.
- ✓ These organizations can be connected to Internet to avail the services by dispensing with the firewall.
- ✓ The main issues arising in the design of components for use in intranets are,
  - ✓ File services are needed to enable users to share data
  - ✓ Firewalls should ensure legitimate access to services.
  - ✓ Cost of installation and support should be minimum.

Mobile and Ubiquitous computing:

- ✓ Integration of portable computing devices like Laptops, smartphones, handheld devices, pagers, digital cameras, smart watches, devices embedded in appliances like refrigerators, washing machines, cars etc. with the distributed systems became possible because of the technological advances in device miniaturization and wireless networking.
- ✓ These devices can be connected to each other conveniently in different places, makes mobile computing possible.
- ✓ In mobile computing, users who are away from home intranet, are still allowed to access resources via the devices they carry.

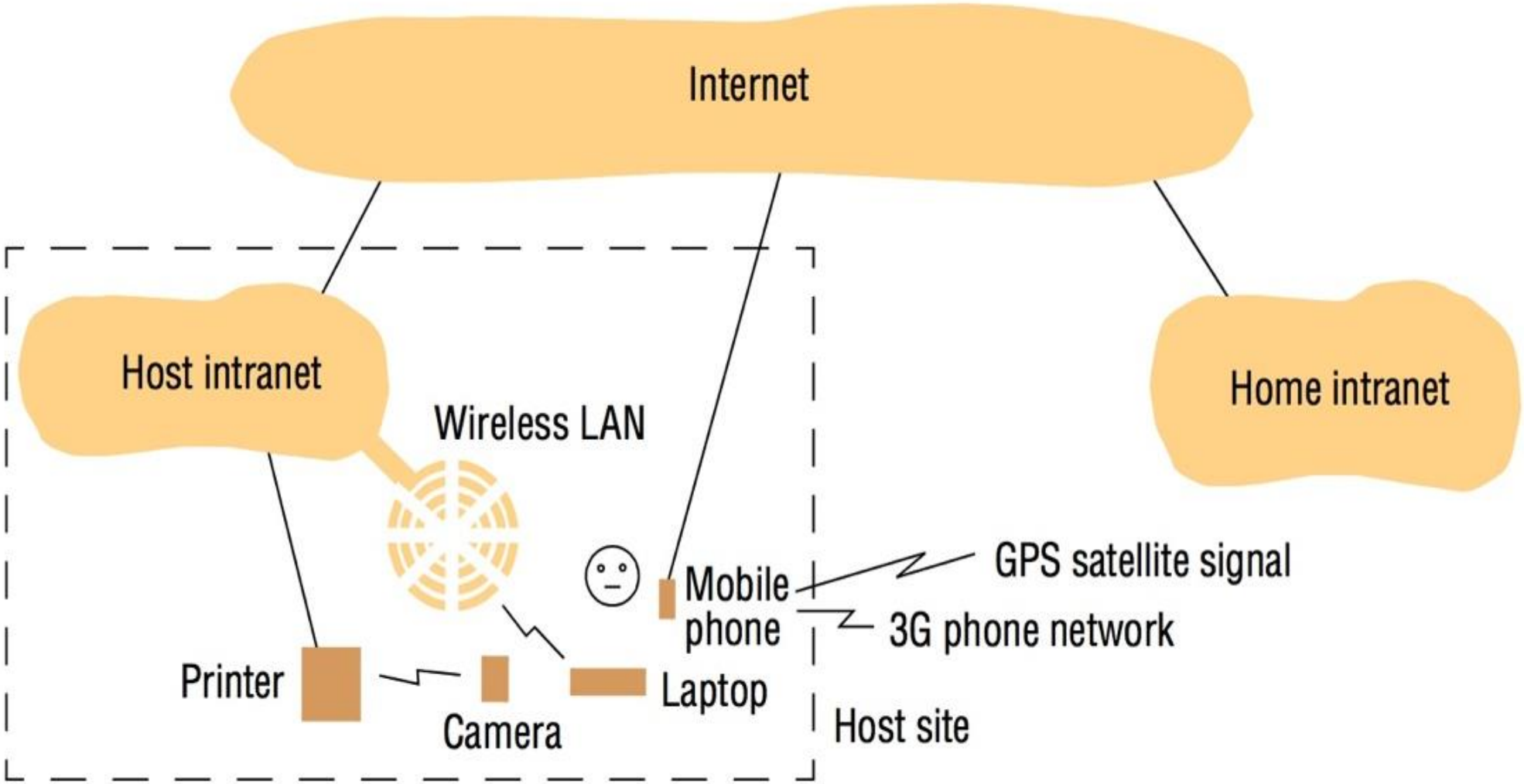
Mobile and Ubiquitous computing: contd..

- ✓ Ubiquitous computing is the harnessing of many small, cheap computational devices that are present in users physical environments, including home, office and others.
- ✓ The term ubiquitous is intended to suggest that small computing devices will eventually become so pervasive in everyday objects that they are scarcely noticed.
- ✓ The presence of computers everywhere is useful only when they can communicate with one another.
- ✓ E.g. it would be convenient for users to control their washing machine and hi-fi system using " Universal remote control" device at home.

Mobile and Ubiquitous computing: contd..

- ✓ The mobile user can get benefit from computers that are everywhere.
- ✓ Ubiquitous computing could benefit users while they remain in a single environment such as the home, office or hospital.
- ✓ Figure below shows a user who is visiting a host organization. The users home intranet and the host intranet at the site that the user is visiting. Both intranets are connected to the rest of the Internet.

Figure: Portable and handheld devices in a distributed system



1.4 Challenges:

HETEROGENEITY: The Internet enables users to access services and run applications over a heterogeneous collection of computers and networks.

Heterogeneity (that is, variety and difference) applies to all of the following:

- **Networks;**
- **Computer hardware;**
- **Operating systems;**
- **programming languages;**
- **Implementations by different developers.**

openness

- ✓ The openness of a computer system is the characteristic that determines whether the system can be extended and re-implemented in various ways.
- ✓ The openness of distributed system determined primarily by the degree to which new resource sharing devices can be added .
- ✓ Openness cannot be achieved unless the specification and documentation of the Key software interfaces of the components of a system are made available to software developers. In a word, the key interfaces are *published*.

Security

- ✓ Many of the information in distributed systems have a high intrinsic value to their users.
- ✓ Their security is therefore of considerable importance.
- ✓ Security for information resources has three components:
  - ✓ confidentiality (protection against disclosure to unauthorized individuals),
  - ✓ integrity(protection against alteration or corruption), and
  - ✓ availability (protection against interference with the means to access the resources).

For example:

1. A doctor might request access to hospital patient data or send additions to that data.

2. In electronic commerce and banking, users send their credit card numbers across the Internet.

scalability

- ✓ A system is described as *scalable* if it will remain effective when there is a significant increase in the number of resources and the number of users.
- ✓ The number of computers and servers in the Internet has increased dramatically.

Date	*Computers*	*Web servers*	*Percentage*
1993, July	1,776,000	130	0.008
1995, July	6,642,000	23,500	0.4
1997, July	19,540,000	1,203,096	6
1999, July	56,218,000	6,598,697	12
2001, July	125,888,197	31,299,592	25
2003, July	~200,000,000	42,298,371	21
2005, July	353,284,187	67,571,581	19

FAILURE HANDLING

- ✓ When faults occur in hardware or software, programs may produce incorrect results or may stop before they have completed the intended computation.
- ✓ Failures in a distributed system are partial – that is, some components fail while others continue to function.
- ✓ The following are techniques for dealing with failures.
  - Detecting failures
  - Masking failures
  - Tolerating failures
  - Recovery from failure
  - Redundancy

Concurrency

- ✓ Both services and applications provide resources that can be shared by clients in a distributed system.
- ✓ There is therefore a possibility that several clients will attempt to access a shared resource at the same time.
- ✓ For example, a data structure that records bids for an auction may be accessed very frequently when it gets close to the deadline time.
- ✓ The process that manages a shared resource could take one client request at a time. But that approach limits throughput.
- ✓ Therefore services and applications generally allow multiple client requests to be processed concurrently.

TRANSPARENCY

Transparency is defined as the concealment from the user and the application programmer of the separation of components in a distributed system, so that the system is perceived as a whole rather than as a collection of independent components.

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

Location transparency enables resources to be accessed without knowledge of their physical or network location (for example, which building or IP address).

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.

contd..

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.

Important Questions:

1. Define a distributed system and explain the same with two examples.
2. Analyze the different challenges of distributed system.
3. Discuss the types of hardware and software resources which can be shared in distributed system with an illustration.
4. Discuss the Software Layers of distributed system architectural model.
5. What are variations of client-server model?
6. Describe the interaction model of distributed system.
7. Write the design requirements for Distributed architectures.
8. Describe the failure model of distributed system.

End of UNIT I