

# **Session 2:**

## **Distributed System Challenges**

**Course Leader: Jishmi Jos Choondal**



# Objectives

At the end of the lecture, the student will be able to

- Gain a good understanding of the challenges related to *heterogeneity*
- *openness*
- *security*
- *scalability*
- *failure handling*
- *concurrency* and *transparency* as they apply to distributed systems



# Contents

- Heterogeneity
- Openness
- Security
- Scalability
- Failure handling
- Concurrency
- Transparency



# Heterogeneity

- Heterogeneity in DS:
  - Different networks, hardware, operating systems, programming languages and implementations by different developers
- Networks
  - internet protocols are set up to solve network heterogeneities to communicate between different sorts of network
  - A computer on a different sort of network will need an implementation of the internet protocols for that network.
- Hardware
  - Data type representation in different ways in different sorts of hardware
- OS
  - Calls for exchanging messages in Unix are different from the calls in Windows



# Heterogeneity

- Programming Languages
  - Use different representations for characters and data structures such as array and records
- Implementations by different developers
  - Cannot communicate with one another unless they use common standards

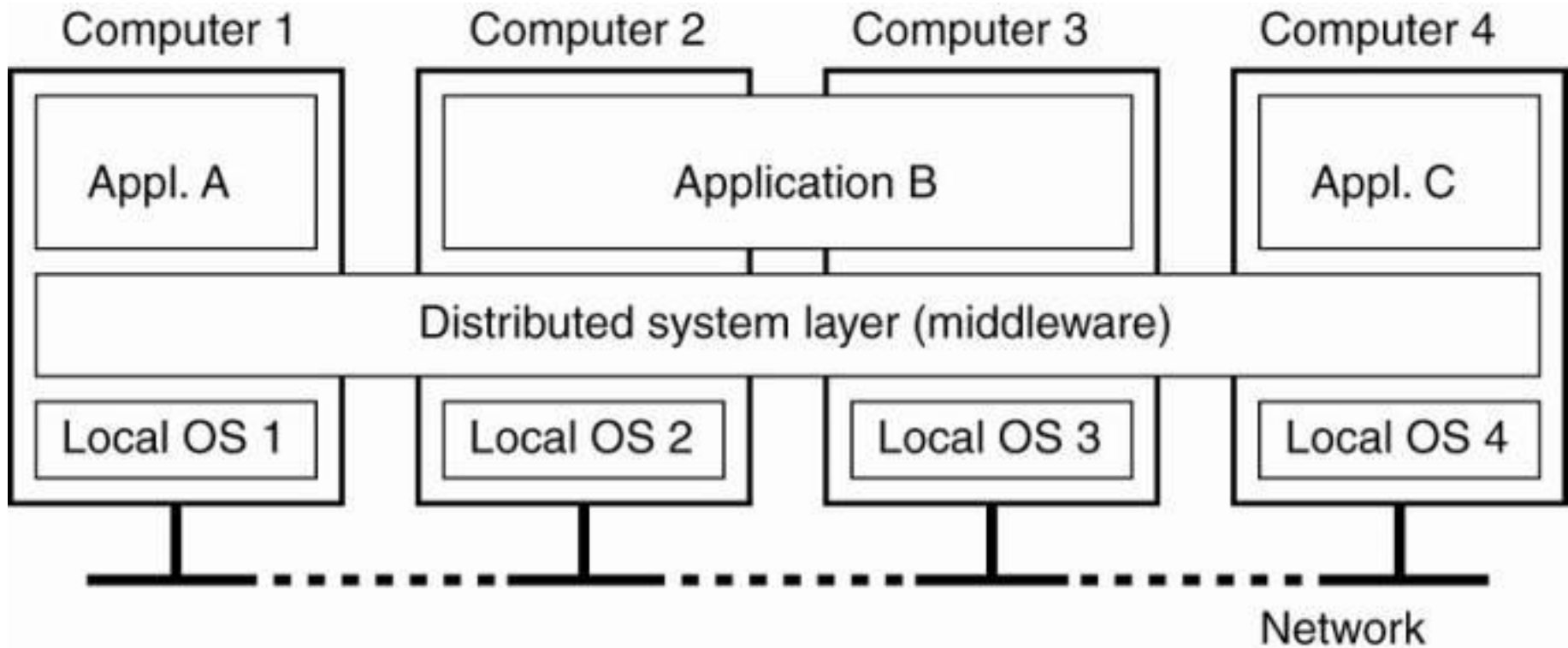


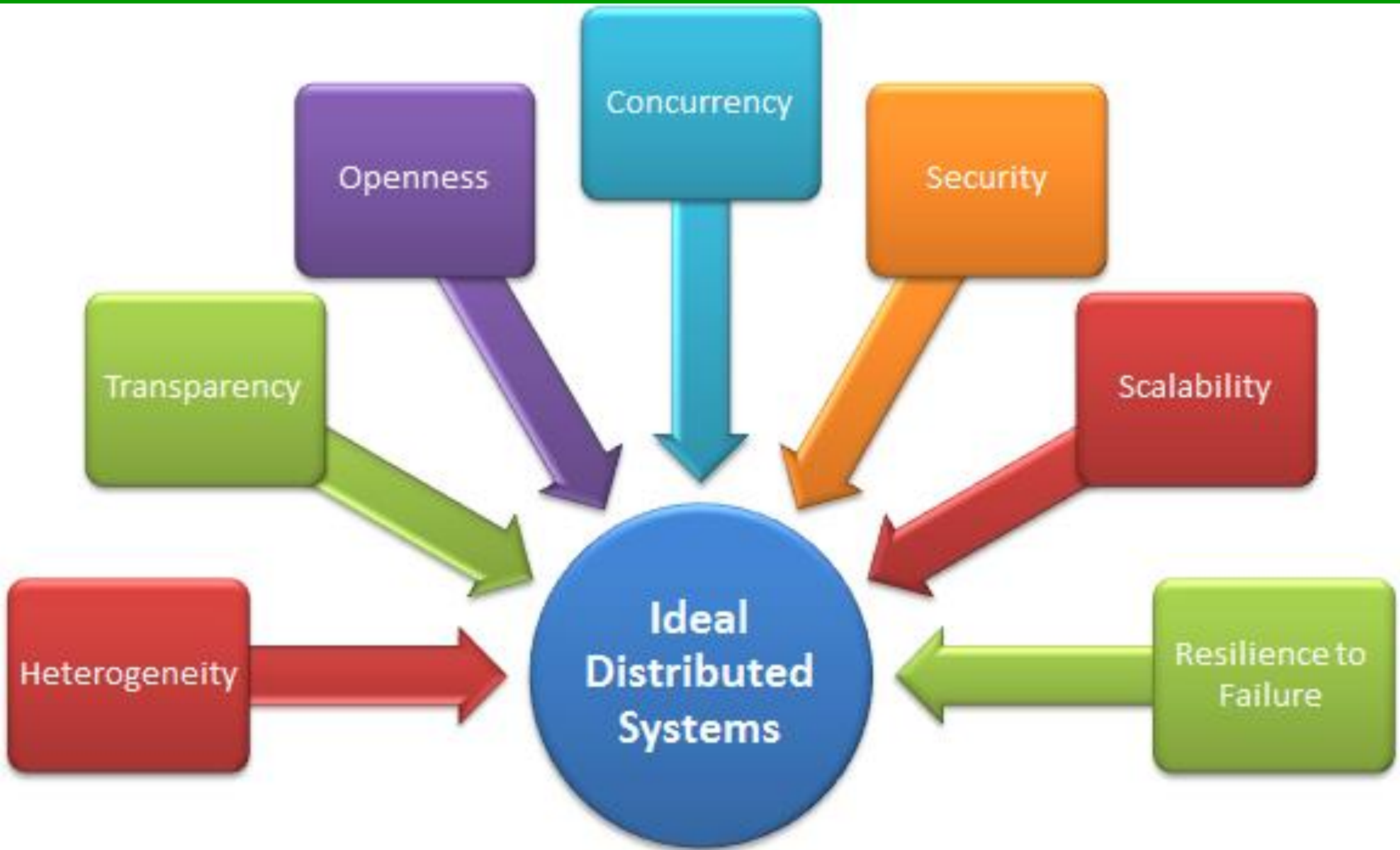
# Middleware

- A software layer that provides a programming abstraction as well as masking the heterogeneity of the underlying networks, hardware, OS and Programming Languages. E.g., Java RMI
- It also provides uniform computational models for use by the programmers of servers and distributed applications.
- Possible models include
  - Remote Object Invocation (CORBA)
    - An object in a program running on one computer to invoke a method of an object in a program running on another computer
  - Remote event notification
  - Remote SQL access
  - Distributed transaction processing



# Middleware







# Heterogeneity and mobile code

- Mobile code: Program code that can be transferred from one computer to another and run at the destination. E.g., Java applets
  - Inclusion of Java script programs in some web pages loaded into client browsers
- Virtual Machine: provides a way of making code executable on a variety of host computers
  - The compiler for a particular language generates code for a virtual machine instead of a particular hardware order code. E.g., Java Compiler
  - It is not applicable to programs written in other languages



# Openness

- A characteristic that determines whether the system can be extended and reimplemented in various ways
- Openness of DS is determined primarily by the degree to which new resource-sharing services can be added and be made available for use by a variety of client programs
- Open DS are based on the provision of a uniform communication mechanism and published interfaces for access to shared resources
- They can be constructed from heterogeneous hardware and software
- The conformance of each component to the published standard must be carefully tested and verified if the system is to work correctly



# Security

- Security for information resources has three components:
  - *Confidentiality*: protection against disclosure to unauthorized individuals
  - *Integrity*: protection against alteration or corruption
  - *Availability*: protection against interference with the means to access the resources
- Firewalls, Encryption Techniques are used to ensure security



# Security

- Two security challenges:
  - Denial of service attacks (DoS)
    - A user may wish to disrupt a service for some reason
    - This can be achieved by bombarding the service with such a large number of pointless requests that the serious users are unable to use it
  - Security of mobile code
    - The possible effects of running the program are unpredictable



# Scalability

- A system is described as scalable if it remains effective when there is a significant increase in the number of resources and the number of users. E.g., the Internet
- Challenges of scalable DSs:
  - Controlling the cost of physical resources
  - Controlling the performance loss
  - Preventing software resources from running out
  - Avoiding performance bottlenecks—by partitioning the name table between servers: DNS



# Failure Handling

- When faults occur in hardware or software
  - Programs may produce incorrect results or
  - They may stop before they have completed the intended computation
- Failures in a distributed system are partial
  - Some components fail while others continue to function



# Failure Handling

- Techniques for dealing with failures:
  - Detecting failures
    - Checksum
  - Masking failures
    - Message retransmission
    - Data written to a pair of disks
  - Tolerating failures
    - Leaving them free to try again later
  - Recovering from failures
    - State of permanent data can be recovered or rolled back
  - Providing redundancy
    - At least 2 different routes between any two routers
    - In the DNS, every name table is replicated
    - Database may be replicated in several servers



# Concurrency

- It will arise when several clients will attempt to access a shared resource at the same time.
- Any object that represents a shared resource in a DS must be responsible for ensuring that operates correctly in a concurrent environment
  - applies to servers and to objects in applications
- For an object to be safe in a concurrent environment, its operation must be synchronized in such a way that its data remains consistent. E.g., using semaphores





# 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
- Eight forms of 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



# Transparency, Cont'd.

- **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
- **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
- **Mobility transparency:** Allows the movement of resources and clients within a system without affecting the operation of users or programs



# Summary

- Challenges and design issues of the distributed system can be classified under
  - Heterogeneity
  - Openness
  - Security
  - Scalability
  - Failure handling
  - Concurrency and
  - Transparency



# Thank you

- <http://lycog.com/distributed-systems/challenges-distributed-systems/>

