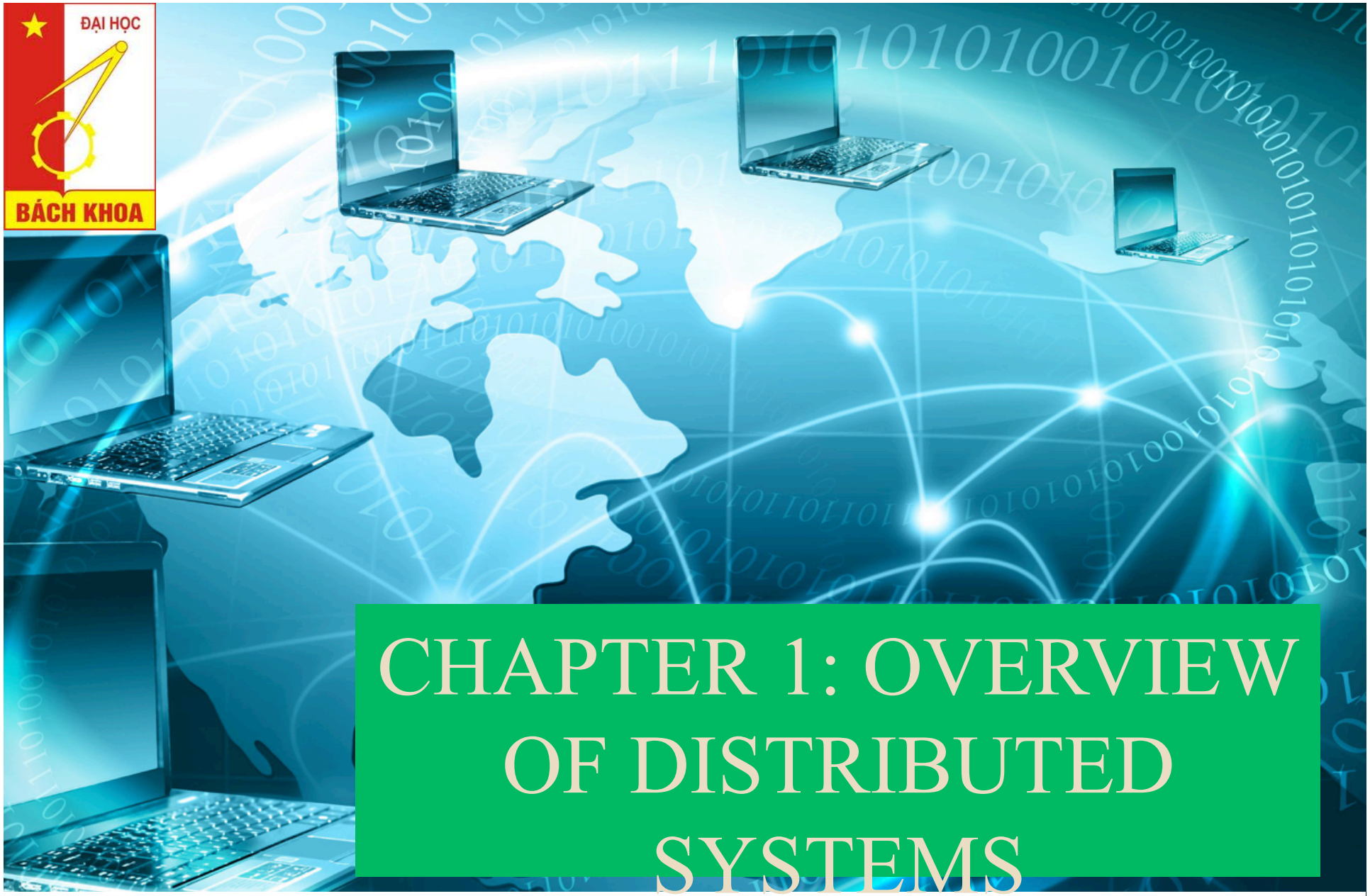


CHAPTER 1: OVERVIEW OF DISTRIBUTED SYSTEMS

Dr. Trần Hải Anh

Outline

1. Introduction
2. Characteristics of Distributes Systems
3. Components of Distributes Systems
4. Topics in Distributed Systems

1. Introduction

1.1. Brief history

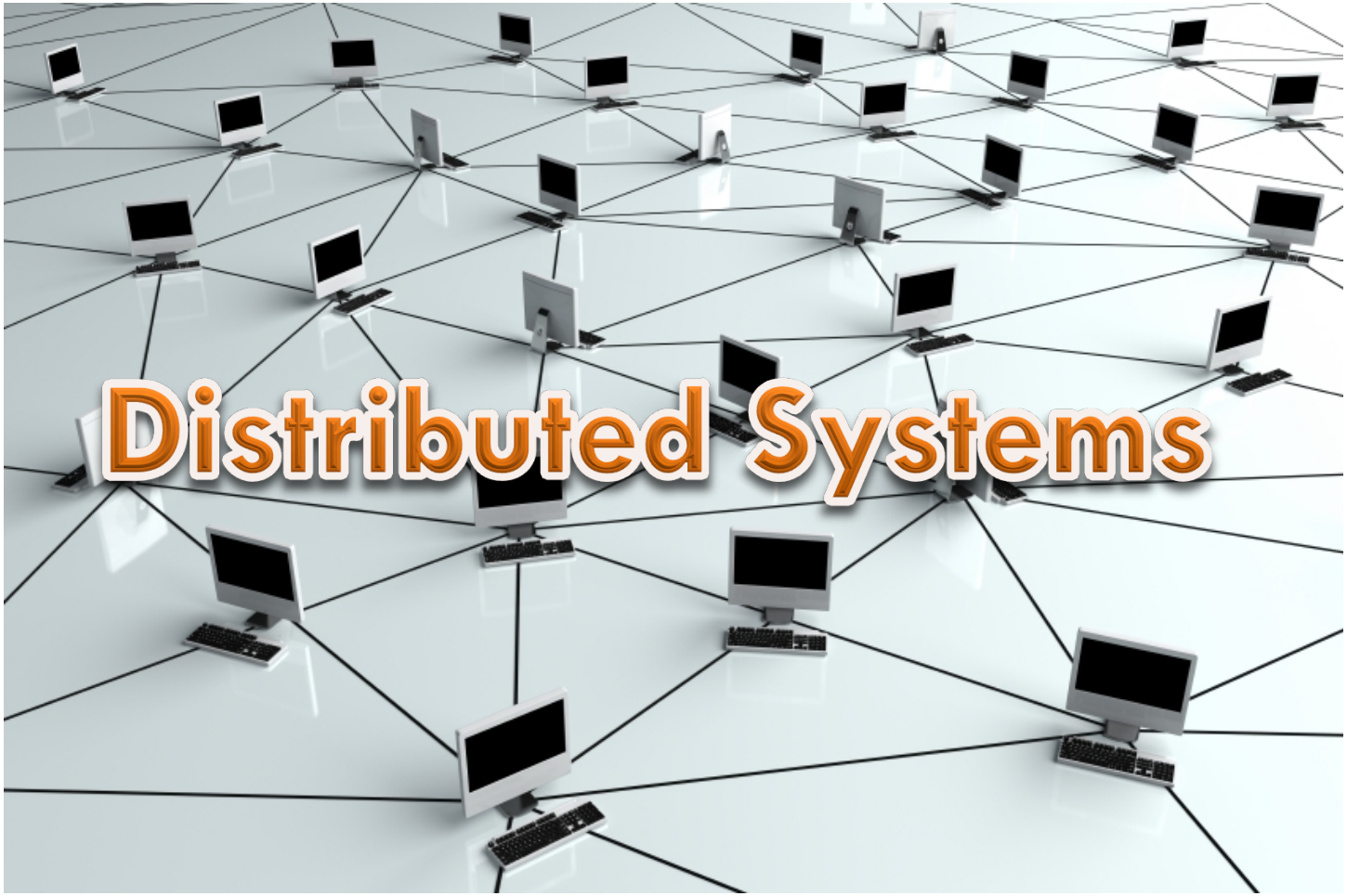
1.2. Definition

1.3. Examples

1.1. History

- History of computer
  - First generation (1937-1946)
    - vacuum tubes
    - ENIAC (Electronic Numerical Integrator And Computer)
  - Second generation: (1947 – 1962)
    - Transistor
    - Universal Automatic Computer (UNIVAC 1).
  - Third generation: 1963 - present
    - IC: Integrated Circuit
    - MS-Dos
    - IBM PC
- History of Computer Network
- Change the way of using PC

Distributed Systems



1.2. Definition

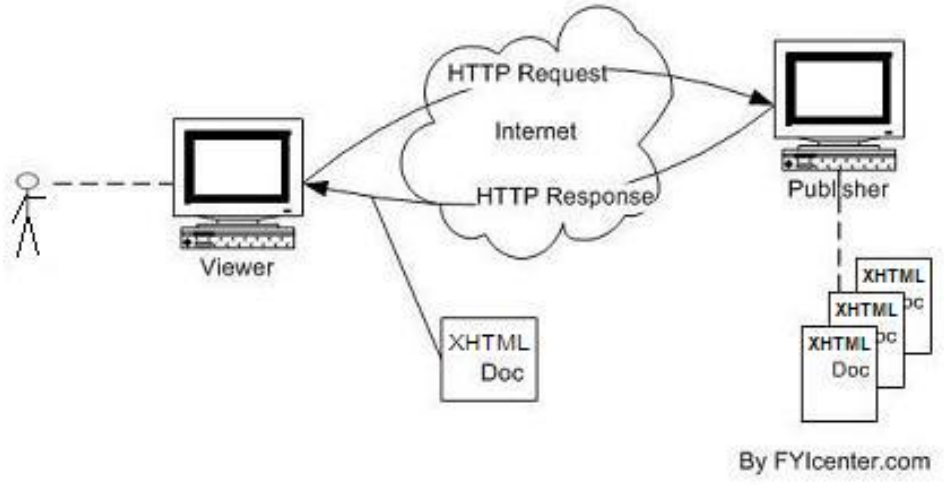
- Independent computers
  - They don’t depend on each others. Different on hardware and software architecture.
- Connected
- Provide common service uniformly
- Users don’t need to care about system’s details
- ***A collection of independent connected computers that provides services to its users as a single coherent system. [Tanenbaum 2006]***

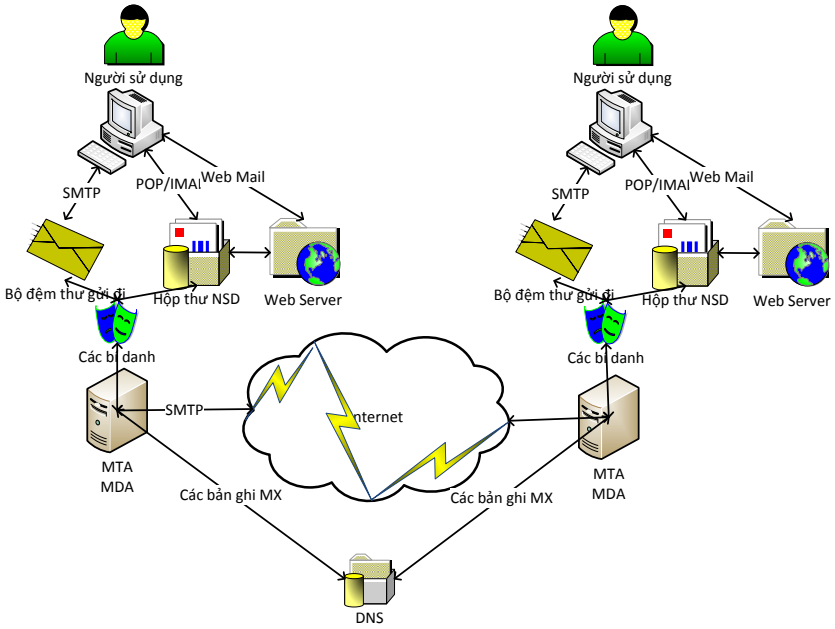
Distributed vs. Ubiquitous Systems

- Networked computer system: appears as many machines
- Distributed computer system: appears as single system
- Ubiquitous system: appears as no computer system

1.3. Examples

- WWW
- Email system
- Etc.





2. Goals

2.1. Making resources accessible

2.2. Distribution transparency

2.3. Openness

2.4. Scalability

2.1. Making resources accessible

- Easy to access remote resources
- Resources: anything (printers, computers, storage facilities, data, files, web pages, etc.)
- Example:
  - Sharing printer
  - Sharing supercomputer, high-performance storage system
  - Other expensive peripherals
- Working together: groupware
- Security problems: eavesdropping, intrusion on communication, etc.

2.2. Distribution Transparency

- Hide the fact that its processes and resources are physically distributed across multiple computers
- Appear as a single computer system → transparent

Types of transparency

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	Hide that a resource may be moved to another location while in use
Replication	Hide that a resource is replicated
Concurrency	Hide that a resource may be shared by several competitive users
Failure	Hide the failure and recovery of a resource

Degree of transparency?

Attempting to completely hide all distribution aspects from users is not a good idea

2.3. Openness

- **Open distribution system** is a system that offers services according to standard rules that describe the syntax and semantics of those services.
- Protocols
- Services are specified through **interfaces.**
- Interface Definition Language (IDL).
- Interoperability
- Portability
- Extensible

2.4. Scalability

- Size
  - Add more users and resources
- Geographical scalability
  - Users and resources may lie far apart
- Administrative scalability
  - It spans many independent administrative organizations

Scalability problems

- Size:
  - Centralized services
  - Centralized data
  - Centralized algorithms
- Geographical scalability
  - LAN → wide area network
  - Broadcasting
  - Reliable communication
- Administrative scalability
  - Resource usage
  - Management
  - Security

Scaling techniques

- Asynchronous communication
- Distribution
- Replicate
- Caching

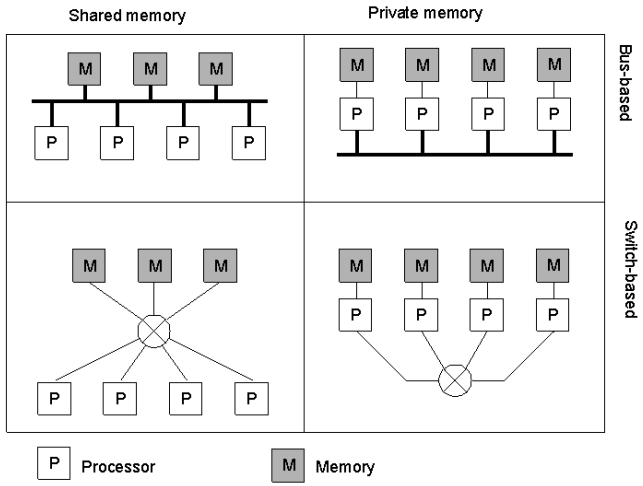
3. Components of Distributed Systems

3.1. Hardware

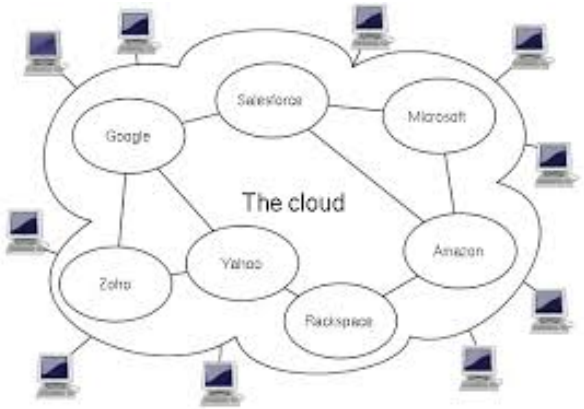
3.2. Software

3.3. Middleware

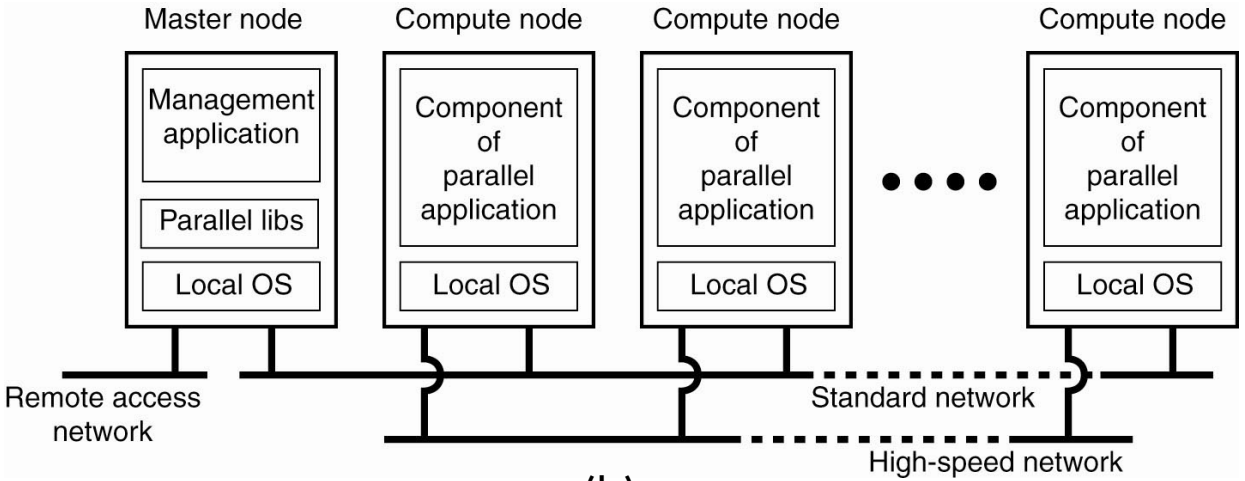
3.1. Hardware



(a)



(c)



(b)

3.2. Software

System	Description	Main Goal
DOS	Multicomputer, multiprocessors	Transparency
NOS	NOS on local machines	Local services for other machines
Middleware	Provide basic services to develop apps	Distributed transparency

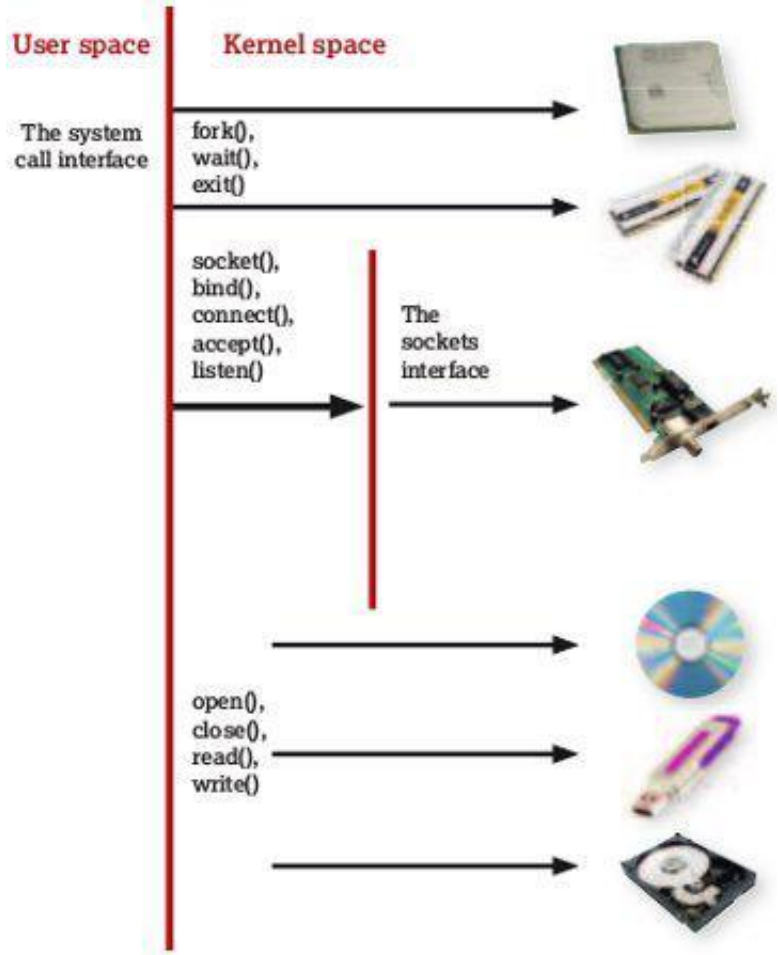
- DS is similar to OS
  - Handle the resources
  - Hide the complexity and heterogenity
- 2 types:
  - tightly-coupled systems (DOS)
  - loosely-coupled systems (NOS)

3.2.1. Distributed Operating Systems (DOS)

- Multiprocessor OS
- Multicomputer OS
- Like the uniprocessor OS, but handling multi processors.

Uniprocessor OS

- Main goal: sharing the resource
- Set the access permission for each app/process
- 2 modes: kernel mode & user mode



User mode & Kernel mode

- *User mode:*
  - Không gian nhớ ảo riêng
  - Giới hạn truy cập
- *Kernel mode*:
  - Không gian nhớ ảo đơn
  - Không giới hạn truy cập

Monolithic kernel vs Microkernels

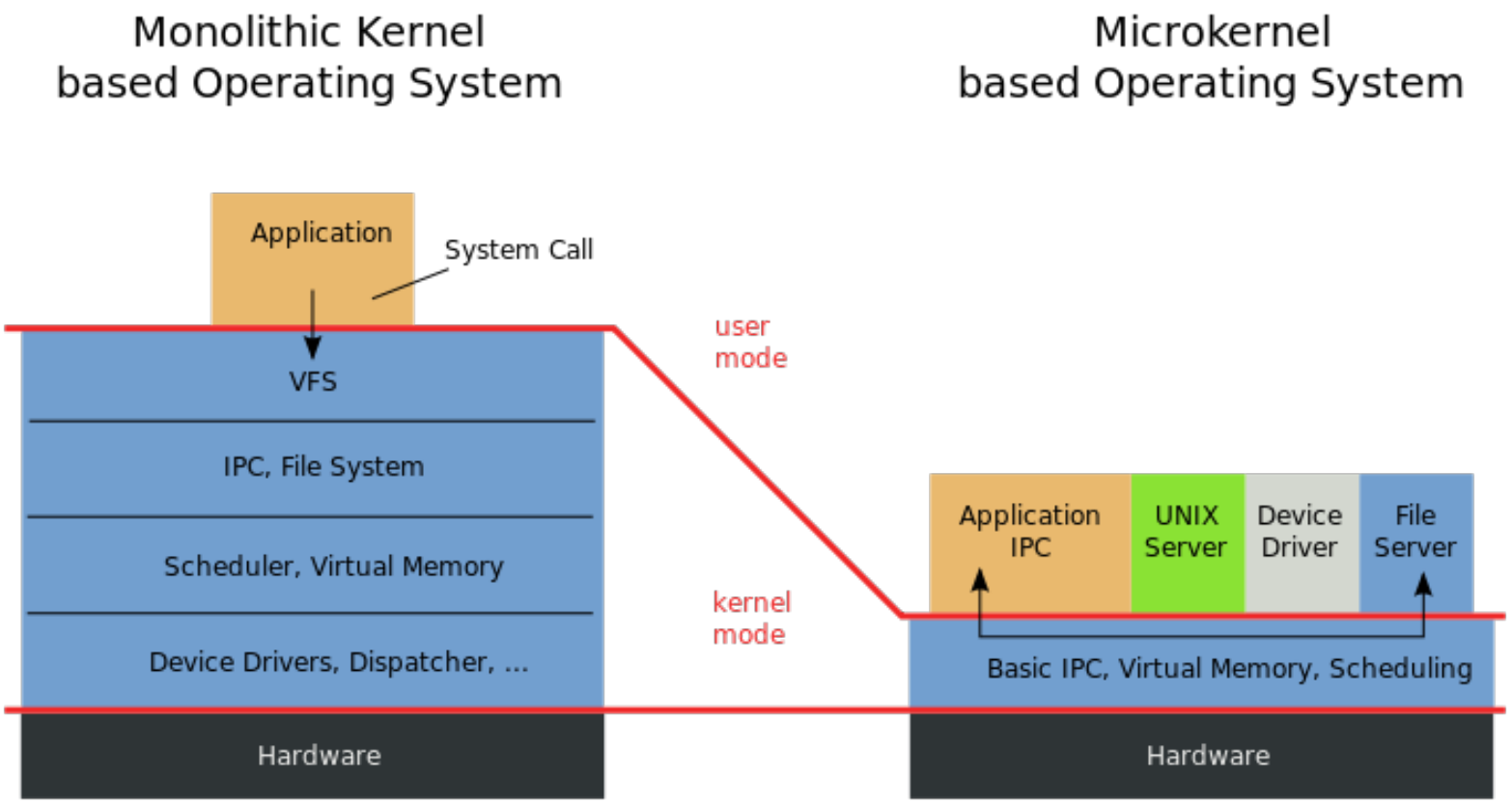
Monolithic kernel

- Single process runs on a single memory space
- All services run on one kernel memory space
- E.g: MS-DOS, UNIX, Linux.

Microkernels

- Kernel is divided into separated processes
- Processes run on either user-space or kernel-space
- Processes run separately on separate memory spaces
- E.g: QNX, L4, HURD, MINIX

Microkernel

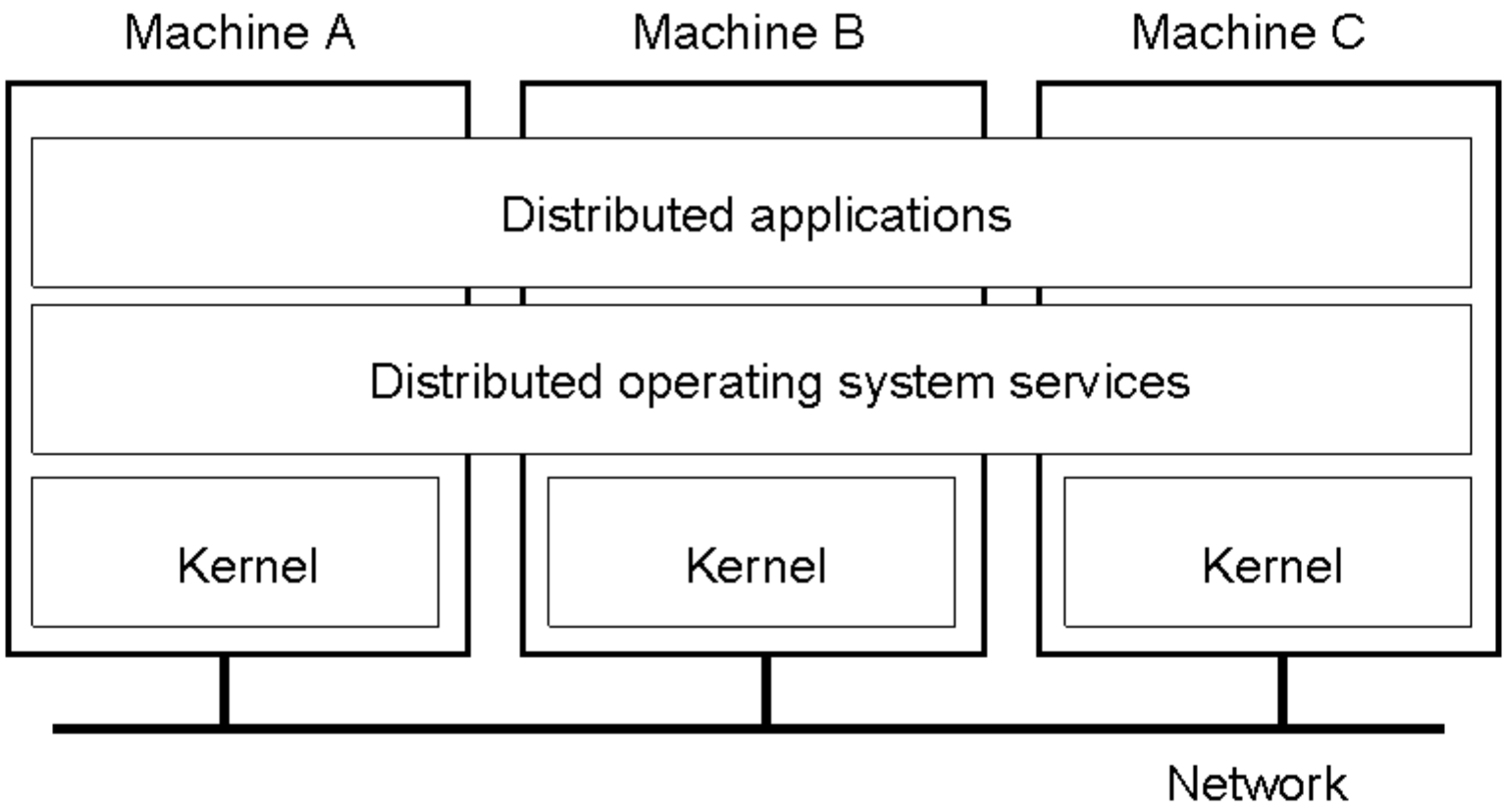


Separate app from OS code in using microkernel

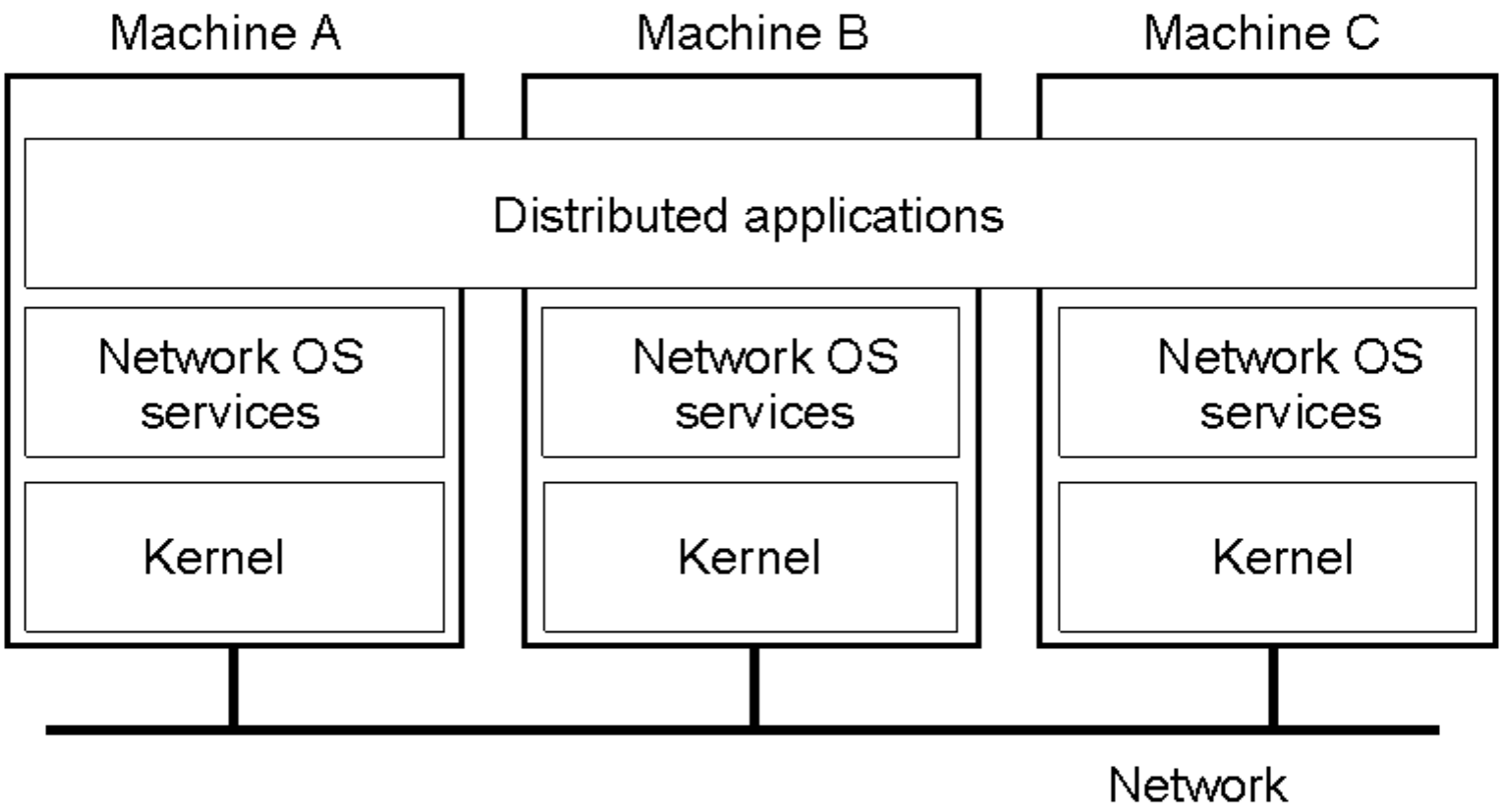
Multiprocessor OS

- Use the common shared memory
- OS supports multi-processor, transparent to apps.
- Avoid concurrent accesses
  - Semaphore
  - Monitor

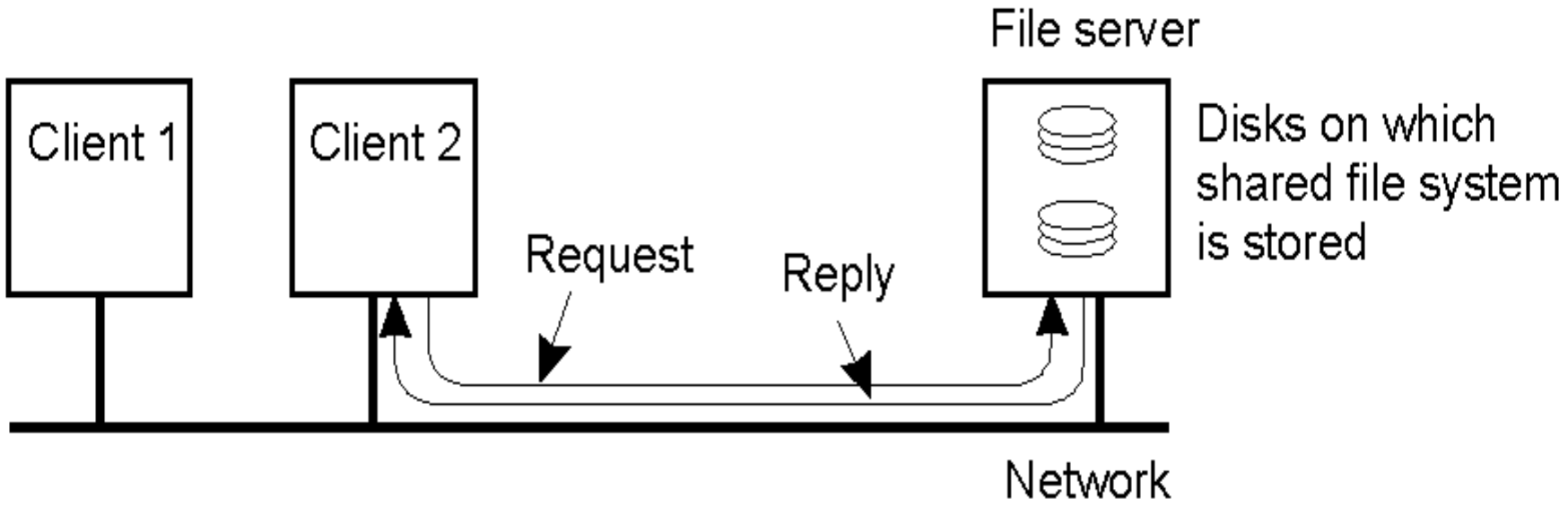
Multicomputer OS



3.2.2. Network OS

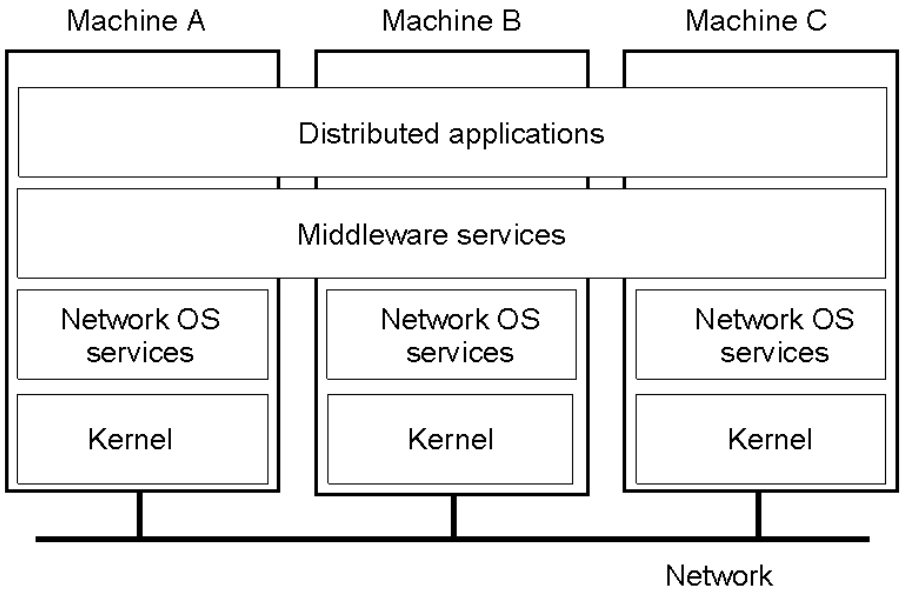


Network OS


File server
Client 1
Client 2
Request
Reply
Disks on which
shared file system
is stored
Network

3.3. Middleware

- Combine advantages of DOS and NOS
- Middleware
- E.g:
  - File system in UNIX
  - RPC
- Middleware service:
  - Transparent access
  - High level communication facilities



Homework

- Do the assignment on the LMS
- Watch the online course for the chapter 1