

# ỨNG DỤNG PHÂN TÁN

**Nhóm chuyên môn UDPT – Trường Công nghệ thông tin Phenikaa**  
(Trần Đăng Hoan, Đỗ Quốc Trường, Nguyễn Thành Trung, Phạm Kim Thành, Nguyễn Hữu Đạt)

# GIỚI THIỆU CHUNG HỌC PHẦN

- Học phần: Ứng dụng phân tán

Số tín chỉ	2 (1.5; 0.5; 4)
Tổng số tiết tín chỉ	37.5
- Số tiết lý thuyết	22.5
- Số tiết thảo luận/bài tập/thực hành	15
- Số tiết tự học	60
Số bài kiểm tra	1 (1 LT, 1 TH)

- Đánh giá giữa kỳ: B1. Bài kiểm tra + TL. Tiểu luận
- Đánh giá cuối kỳ: Thi trắc nghiệm trên máy

# GIỚI THIỆU CHUNG HỌC PHẦN

Mục tiêu (MT)	Miêu tả (mức độ tổng quát)	CĐR của CTĐT cấp độ 2
MT1	Nắm được các kiến thức chuyên ngành về hệ thống phân tán bao gồm: mô hình client/server, các giải thuật phân tán, truyền thông, tiến trình, định danh, đồng bộ hóa trong hệ phân tán.	1.3
MT3	Có khả năng làm việc nhóm, tổ chức nhóm	3.1

# GIỚI THIỆU CHUNG HỌC PHẦN

- 1. Tổng quan và Kiến trúc**
- 2. Tiến trình và Trao đổi thông tin trong Hệ phân tán**
- 3. Định danh**
- 4. Đồng bộ hóa**
- 5. Sao lưu và Nhất quán**
- 6. Chống chịu lỗi**

# HỆ PHÂN TÁN

1. Định nghĩa
2. Đặc điểm của hệ phân tán
3. Thành phần của hệ phân tán
4. Các loại hệ phân tán
5. Các vấn đề cần nghiên cứu trong hệ phân tán

# 1. Định nghĩa

---

1. Lịch sử phát triển
2. Các định nghĩa
3. Ví dụ

# 1. Định nghĩa

## 1.1. Lịch sử phát triển của các hệ thống máy tính

- Lịch sử phát triển các hệ thống máy tính
  - ✧ Thế hệ máy tính thứ nhất (1945 – 1956)
    - Bóng đèn chân không
    - ENIAC (Electronic Numerical Integrator And Computer)
  - ✧ Thế hệ thứ hai (1958-1964)
    - Transistor
  - ✧ Thế hệ thứ ba (1965-1971)
    - IC: Integrated Circuit
  - ✧ Thế hệ thứ tư (1972-ngày nay)
    - VLSI: Very Large Scale Integration
- Lịch sử phát triển Mạng máy tính
- Thay đổi về cách thức sử dụng máy tính

# 1. Định nghĩa

## 1.2. Định nghĩa

- Các máy tính độc lập
  - ✦ Không phụ thuộc lẫn nhau, có thể là các máy tính có kiến trúc khác nhau, có thể là các máy tính có phần mềm hệ thống khác nhau
- Kết nối lẫn nhau
  - ✦ Bằng mạng máy tính. Các phần mềm trên các máy tính khác nhau có khả năng phối hợp. Chia sẻ tài nguyên.
    - ✦ Thực hiện một nhiệm vụ chung
- Cung cấp dịch vụ một cách thống nhất
  - ✦ Thống nhất về giao diện, cách thức truy cập dịch vụ ở mức độ thống nhất
    - ✦ NSD không cần phải quan tâm tới các chi tiết của hệ thống

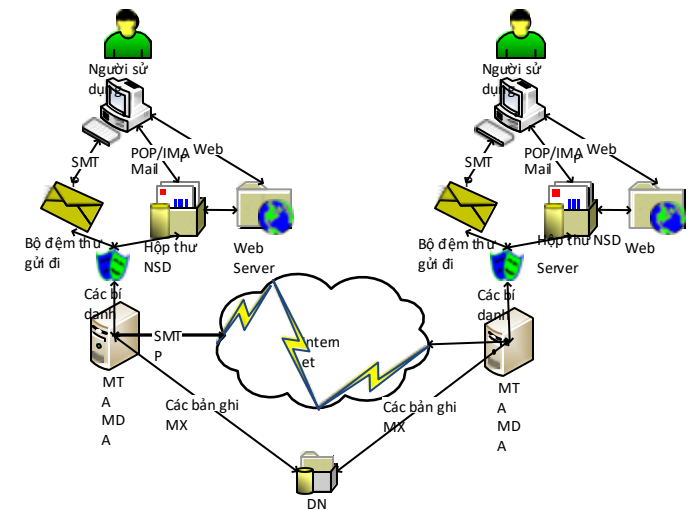


# 1. Định nghĩa

## 1.2. Định nghĩa

- Định nghĩa của Andrew Tannenbaum:

*Hệ phân tán là một tập hợp các máy tính độc lập mà, đối với người dùng, nó như thể một hệ thống đơn giản kết.*



## 2. Đặc điểm

1. Chia sẻ tài nguyên
2. Tính trong suốt  
(transparency)
3. Tính mở
4. Tính co giãn (scalability)

## 2. Đặc điểm

### 2.1. Chia sẻ tài nguyên

- Kết nối tài nguyên
- Giảm chi phí
- Tăng tính sẵn sàng
- Hỗ trợ làm việc nhóm
- Tăng rủi ro về an toàn thông tin

## 2. Đặc điểm

### 2.2. Giảm chi phí

- Nếu sử dụng phần cứng thông dụng cho các máy tính thành phần, thu được tỷ lệ giá/hiệu năng tốt hơn

## 2. Đặc điểm

### 2.3. Tính trong suốt (transparency)

- Hệ thống là duy nhất với NSD
  - ▣ Giao diện giống nhau
  - ▣ Cách thức truy cập giống nhau
- Trong suốt về qui mô và vị trí
- Che giấu tính phân tán của hệ phân tán
- Các loại trong suốt (slide sau)
- Mức độ trong suốt:
  - ▣ Cân bằng giữa hiệu năng và độ trong suốt

## 2. Đặc điểm

### Các loại trong suốt

Loại trong suốt	Mô tả
Truy cập	Che giấu sự khác nhau trong biểu diễn dữ liệu và cách thức truy cập tài nguyên.
Địa điểm	Che giấu vị trí của tài nguyên
Di trú	Che giấu việc tài nguyên chuyển đến địa điểm khác
Chuyển địa điểm	Che giấu việc tài nguyên chuyển đến địa điểm khác trong khi đang được sử dụng
Sao lưu	Che giấu việc dữ liệu được cung cấp từ nhiều bản sao khác nhau
Tương tranh	Che giấu việc tài nguyên được truy cập đồng thời bởi nhiều NSD
Thứ lỗi	Che giấu lỗi và quá trình phục hồi của tài nguyên
Bền vững	Che giấu việc tài nguyên/dữ liệu được lưu trữ bền vững (disk) hoặc không (RAM)

## 2. Đặc điểm

### 2.4. Tính mở

- Cho phép các thành phần có thể được sản xuất bởi các NSX khác nhau.
- Hệ phân tán mở cung cấp các dịch vụ theo các đặc tả về cú pháp và ngữ nghĩa của các dịch vụ, gọi là **giao diện**
- Thường được mô tả bằng ngôn ngữ tương tác dữ liệu (IDL)
- Tính đầy đủ của đặc tả
  - ✧ Quá chi tiết: phụ thuộc vào cài đặt cụ thể của dịch vụ
  - ✧ Không đủ chi tiết: Khi cài đặt phải bổ sung thêm: phụ thuộc vào cài đặt cụ thể của dịch vụ



## 2. Đặc điểm

### 2.4. Tính mở

- Khả năng phối hợp (interoperability)
- Tính khả chuyển (portability)
- Tính mềm dẻo + mở rộng được (flexibility, extensibility)
- Thực hiện: tách biệt chính sách và cơ chế

## 2. Đặc điểm

### 2.5. Tính co giãn

- Qui mô:
  - ▣ số lượng NSD và tài nguyên thay đổi
- Không gian địa lý
  - ▣ Qui mô vùng địa lý có tài nguyên và NSD thay đổi
- Tổ chức
  - ▣ Qui mô tổ chức thay đổi => tổ chức hệ thống thành các domain.

## 2. Đặc điểm

### Co giãn theo số lượng

- Mô hình tập trung
  - ▣ Dịch vụ: cổ chai
  - ▣ Dữ liệu: lưu trữ, xử lý
  - ▣ Giải thuật: thông tin vào ra, xử lý
- Mô hình không tập trung
  - ▣ Phức tạp, gặp các vấn đề về bảo mật và riêng tư
  - ▣ Quyết định cục bộ
  - ▣ Khó phát hiện được lỗi

## 2. Đặc điểm

### Co giãn theo không gian địa lý

- Gần: mạng cục bộ
  - ▣ quảng bá, tốc độ cao, tin cậy, độ trễ cố định
- Xa: mạng diện rộng
  - ▣ Điểm điểm, tốc độ thấp, không tin cậy, độ trễ thay đổi
- Khác nhau
  - ▣ Tốc độ truyền tin, độ trễ, ....
  - ▣ Đồng bộ/không đồng bộ
  - ▣ Các thao tác quảng bá
- Chủ yếu đảm bảo trao đổi thông tin trên mạng diện rộng như với mạng cục bộ

## 2. Đặc điểm

# Một số nguyên lý quan trọng đằng sau mọi HPT

- Kiến trúc hệ thống (system architecture)
- Liên lạc (communication)
- Sao lãp và tính nhất quán (replication & consistency)
- Đồng bộ hóa (synchronisation)
- Tên (naming)
- Chịu lỗi (fault tolerance)
- Bảo mật (security)

## 2. Đặc điểm

### Mô hình

Hầu hết các hệ thống phân tán được xây dựng dựa trên một mô hình cụ thể:

- Bộ nhớ dùng chung (shared memory)
- Đối tượng phân tán (distributed objects)
- Hệ thống file phân tán (distributed file system)
- Tài liệu dùng chung (shared documents)
- Cộng tác phân tán (distributed coordination)

### 3. Kiến trúc

Xem xét tổ chức của một Hệ Phân Tán: tách biệt giữa tổ chức logic và thực thi vật lý

- Thực thi vật lý: cách thức xếp đặt/cài đặt các thành phần phần mềm lên các thiết bị vật lý => Kiến trúc hệ thống
- Tổ chức logic: các thành phần phần mềm, cách thức kết nối, kiểu dữ liệu trao đổi => Kiến trúc phần mềm

## 3. Kiến trúc

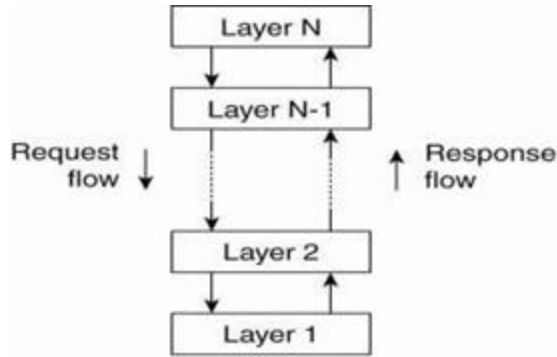
### Kiến trúc hệ thống

- Kiến trúc tập trung
- Kiến trúc không tập trung
- Kiến trúc hỗn hợp



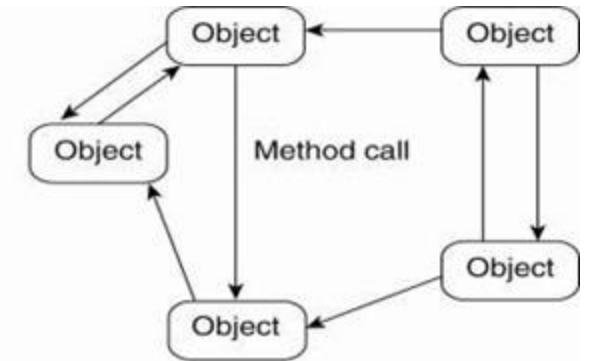
# 3. Kiến trúc

## Các phong cách kiến trúc

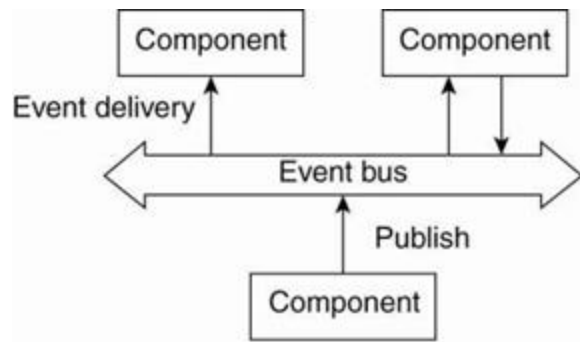


Phân tầng (layered)

Tách thành  
phần theo  
logic xử lý

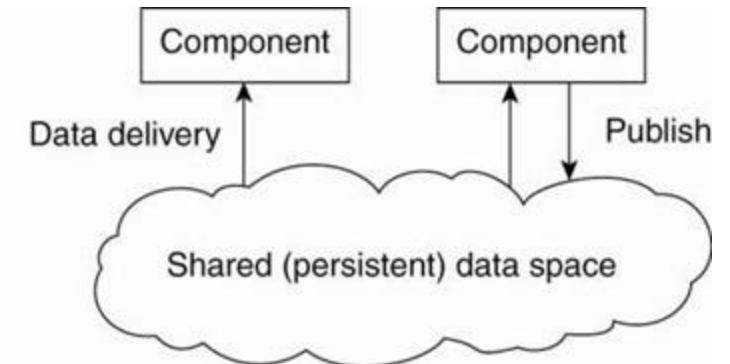


Hướng đối tượng (object-oriented) (b)



Dựa sự kiện (event-based)

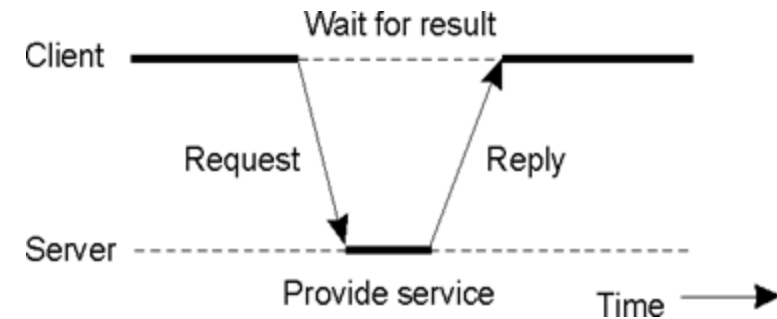
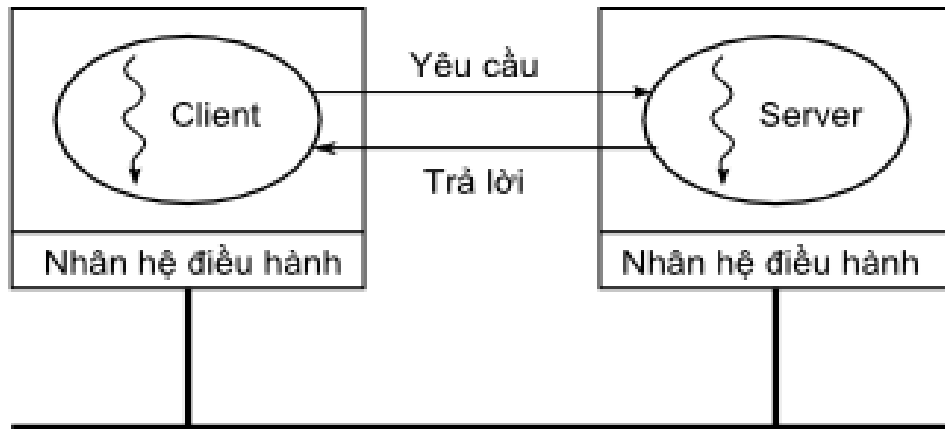
Tách thành  
phần theo  
không gian và  
thời gian



Lấy dữ liệu làm trung tâm (data-centered) (b)

### 3. Kiến trúc

#### Kiến trúc tập trung: Mô hình client-server cơ bản



- **Server** : tiến trình cung cấp dịch vụ
- **Client** : tiến trình sử dụng dịch vụ
- Client và server chạy trên các máy khác nhau
- Client sử dụng dịch vụ theo mô hình yêu cầu/trả lời

# 3. Kiến trúc

## Ví dụ client-server

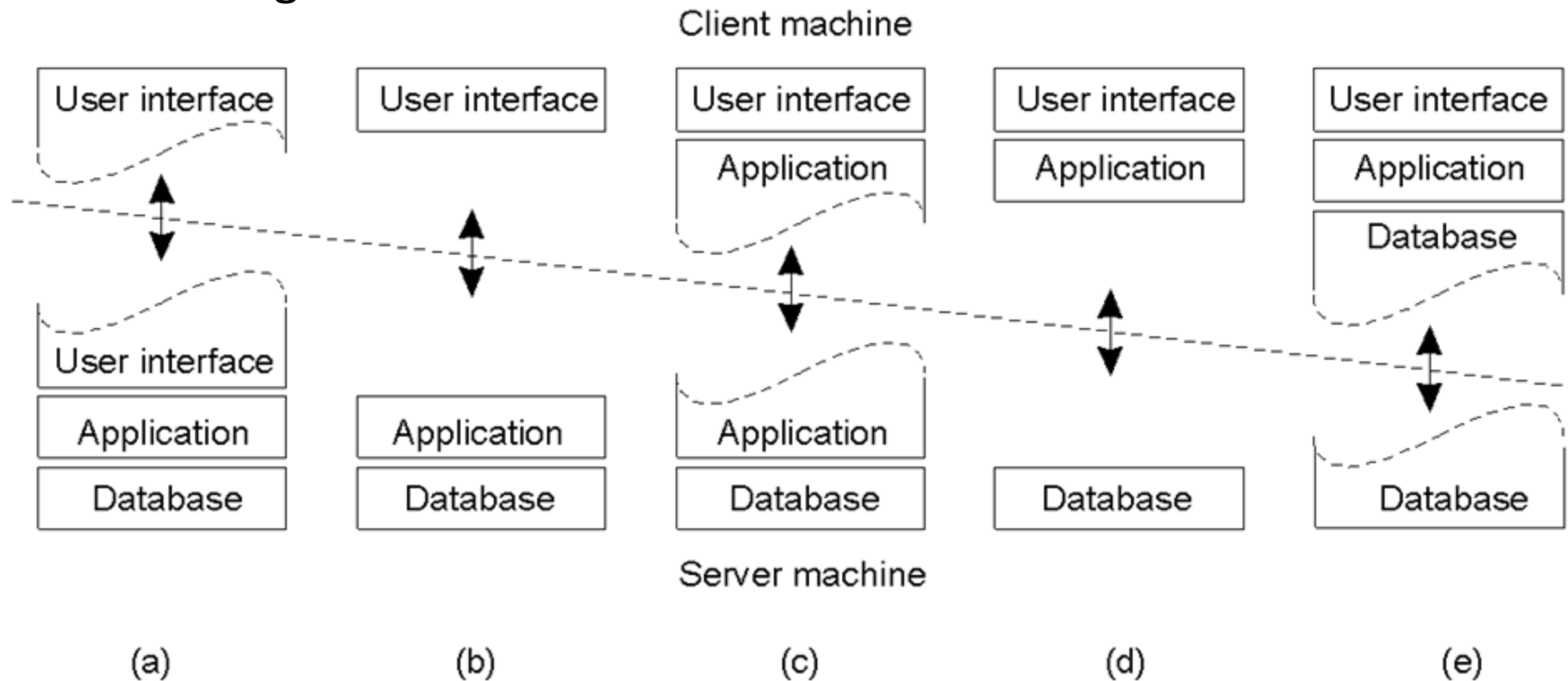
```
server(void) {  
    struct sockaddr_in cin,  
    sin; int sd, sd_client;
```

```
sd = socket(AF_INET, SOCK_STREAM, 0);  
bind(sd, (struct sockaddr *) &sin, sizeof(sin)); //đăng ký dịch  
vụ listen(sd, queuesize); //vòng lặp đợi yêu cầu của  
client while(true) {  
    sd_client = accept(sd, (struct sockaddr *)&cin, &addrlen));  
    recv(sd_client, buffer, sizeof(buffer), 0); //nhận yêu  
cầu DoService(buffer); //thực hiện yêu cầu, ghi kết quả vào  
buffer send(sd_client, buffer, strlen(buffer), 0); //gửi trả  
kết quả close(sd_client);  
}  
close(sd);
```

```
client(void) {  
    struct sockaddr_in sin; //địa chỉ server  
    char  
    buffer[bufsize];  
    int sd;  
  
    sd = socket(AF_INET, SOCK_STREAM, 0);  
    connect(sd, (void*) &sin, sizeof(sin));  
    send(sd, buffer, strlen(buffer),  
        0); //yêu cầu  
    recv(sd, buffer, bufsize, 0); // nhận kết quả  
    close(sd);  
}
```

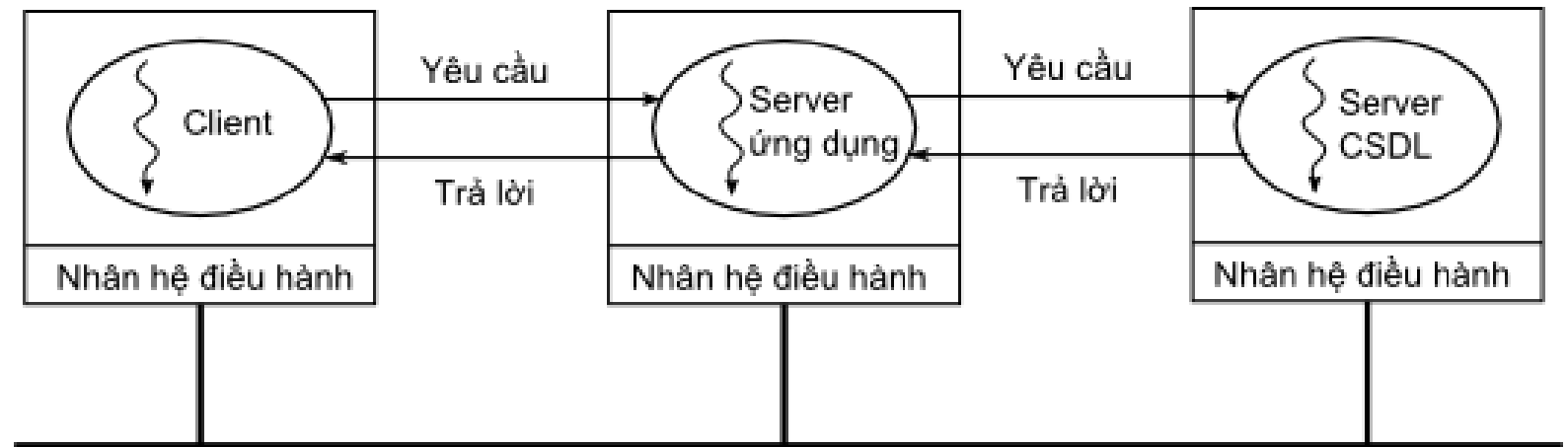
# 3. Kiến trúc

Các kiểu phân chia chức năng giữa client và server  
=> Kiến trúc đa tầng



### 3. Kiến trúc

## Phân tán theo chiều dọc (Multi-tier)



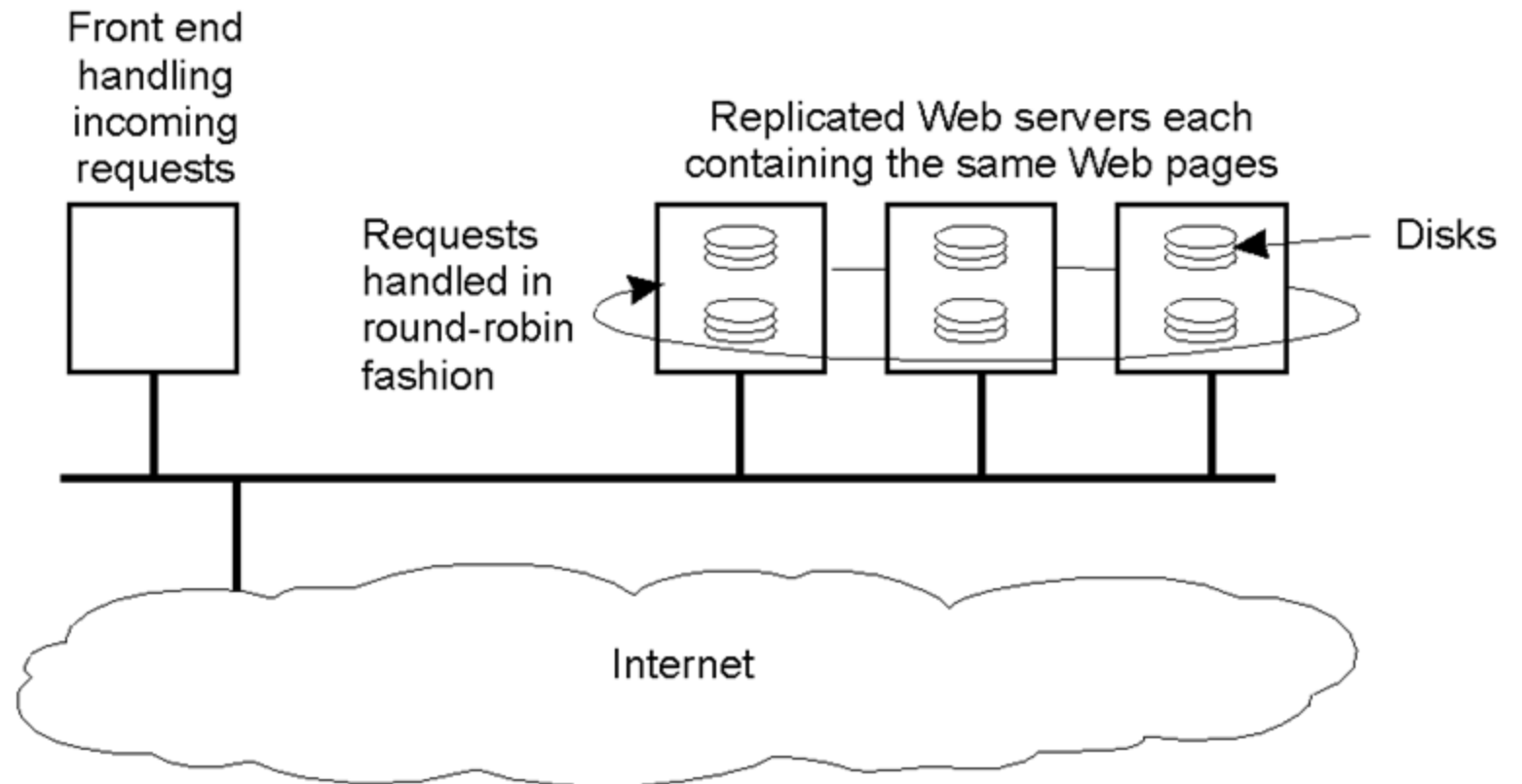
Ba mức phân tầng:

1. Giao diện người dùng
  2. Xử lý / logic ứng dụng
  3. Dữ liệu
- Các thành phần khác nhau về logic được đặt tại các máy khác nhau

### 3. Kiến trúc

## Phân tán theo chiều ngang

Ví dụ về phân tán theo chiều ngang ở một dịch vụ Web.



# Bài tập

Trong 2 trang A4, bằng sự tìm hiểu của cá nhân, hãy lấy một ví dụ cụ thể về một hệ phân tán và mô tả, phân tích theo 4 đặc điểm sau của một HPT

- Chia sẻ tài nguyên
- Tính trong suốt (transparency)
- Tính mở
- Tính co giãn (scalability)

# CẤU TRÚC CLIENT - SERVER ĐA TẦNG (MULTI-TIER CLIENT-SERVER)

- Kiến trúc đa tầng phát triển ở cuối thập niên 1980 và đầu thập niên 1990.
- **Kiến trúc đa tầng** (thường được gọi là *kiến trúc n-tier*) là một **kiến trúc Client-Server**, trong đó trình bày các quy trình, việc xử lý ứng dụng, và quản lý dữ liệu có logic riêng.
- Kiến trúc này cung cấp cho các tiêu chuẩn hóa giao diện rõ ràng giữa các cấp thành phần hợp lý-gọi là 'tầng'.



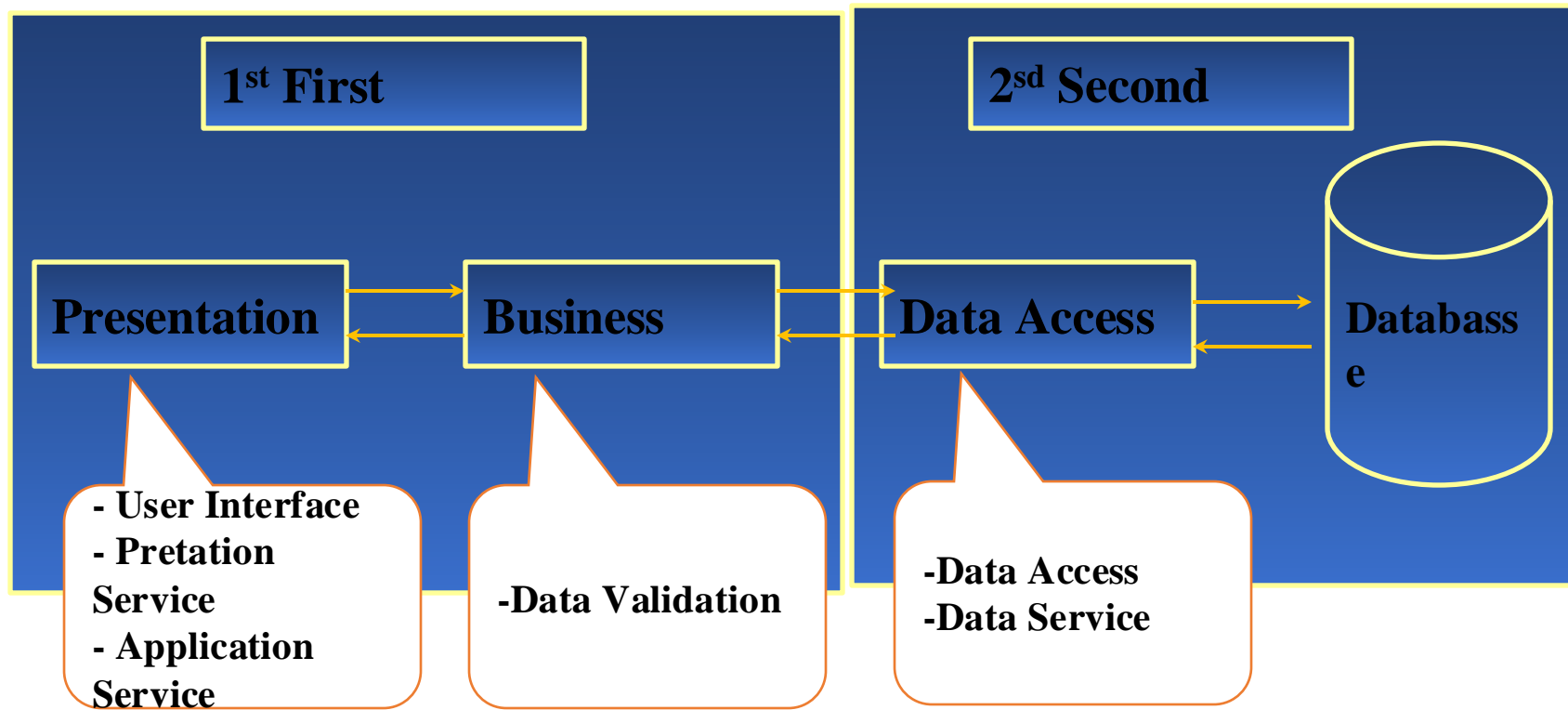
# CẤU TRÚC CLIENT - SERVER ĐA TẦNG (MULTI-TIER CLIENT-SERVER)

## Kiến trúc 2 tầng (Two-tier)

- Là kiến trúc phổ biến và đơn giản nhất của hệ thống client / server
  - **Tầng 1:** Định nghĩa các giao tiếp, các dịch vụ thể hiện, ứng dụng (Presentation, Business)....
  - **Tầng 2:** Cung cấp cơ sở dữ liệu cho các dịch vụ cho khách hàng (Data Access, Database).

# CẤU TRÚC CLIENT - SERVER ĐA TẦNG (MULTI-TIER CLIENT-SERVER)

## Kiến trúc 2 tầng (Two-tier)



# CẤU TRÚC CLIENT - SERVER ĐA TẦNG (MULTI-TIER CLIENT-SERVER)

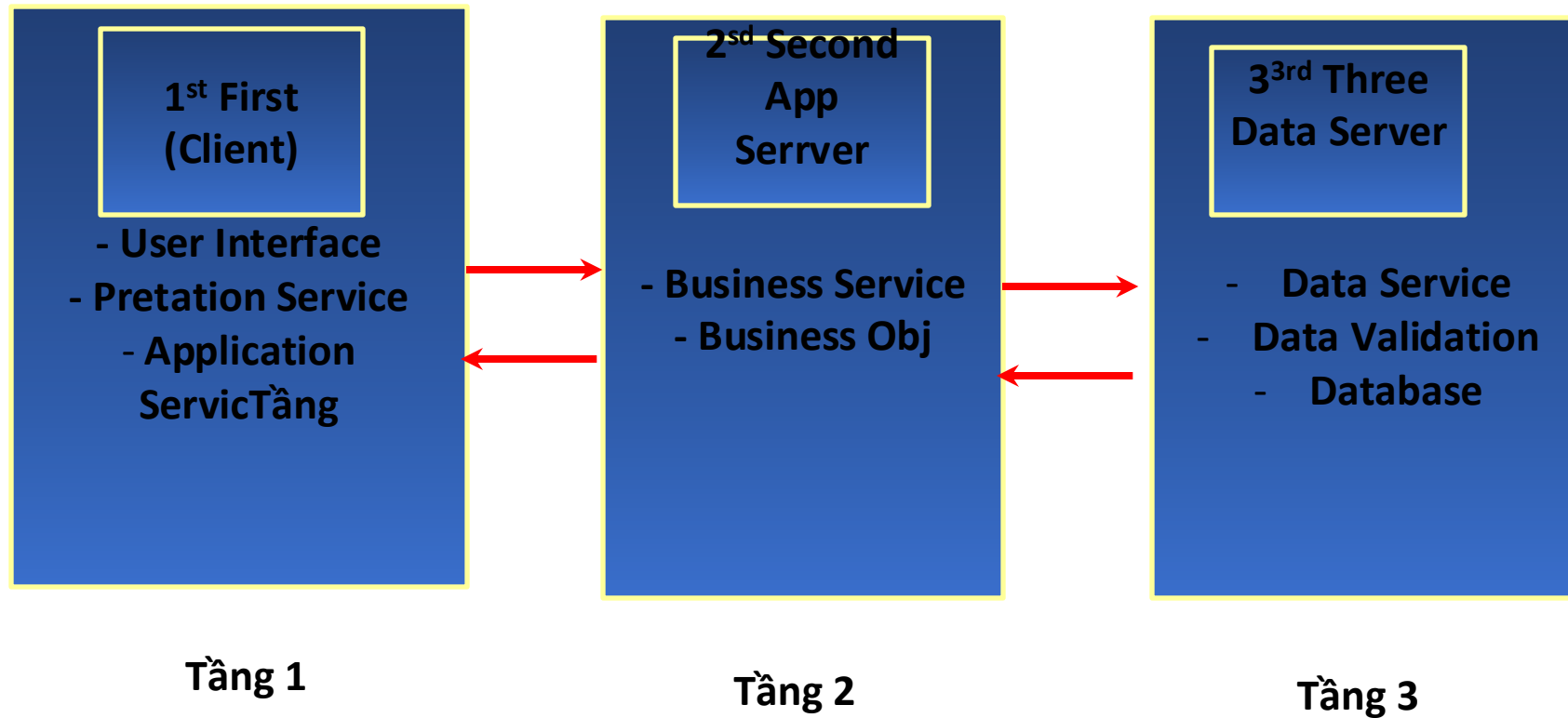
## Kiến trúc 3 tầng (Three-tier)

- Tách biệt rõ ràng 3 tầng khác nhau
  - **Tầng 1**: Cung cấp giao diện người sử dụng tiêu chuẩn
  - **Tầng 2**: Cung cấp dịch vụ được tạo sẵn cho ứng dụng. Đóng vai trò như 1 lớp trung gian cho phép Client có thể tương tác Database
  - **Tầng 3**: Cơ sở dữ liệu của ứng dụng
- Ưu điểm :

Cách tiếp cận này cho phép nhà triển đầu tư tận dụng các hệ thống hiện tại khi tạo ứng dụng mới có thể sử dụng các nguồn lực hiện có.

# CẤU TRÚC CLIENT - SERVER ĐA TẦNG (MULTI-TIER CLIENT-SERVER)

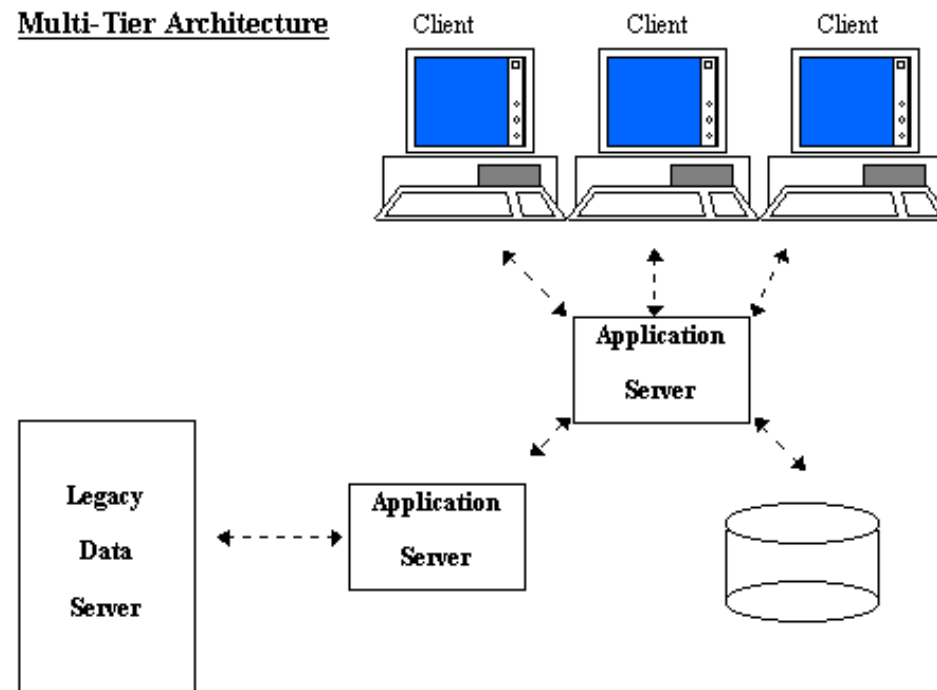
## Kiến trúc 3 tầng (Three-tier)



# CẤU TRÚC CLIENT - SERVER ĐA TẦNG (MULTI-TIER CLIENT-SERVER)

## Kiến trúc n tầng (n-tier)

- Kiến trúc đa tầng, thực hiện bằng cách mở rộng các tầng ở mô hình 3 tầng,
- Ví dụ :



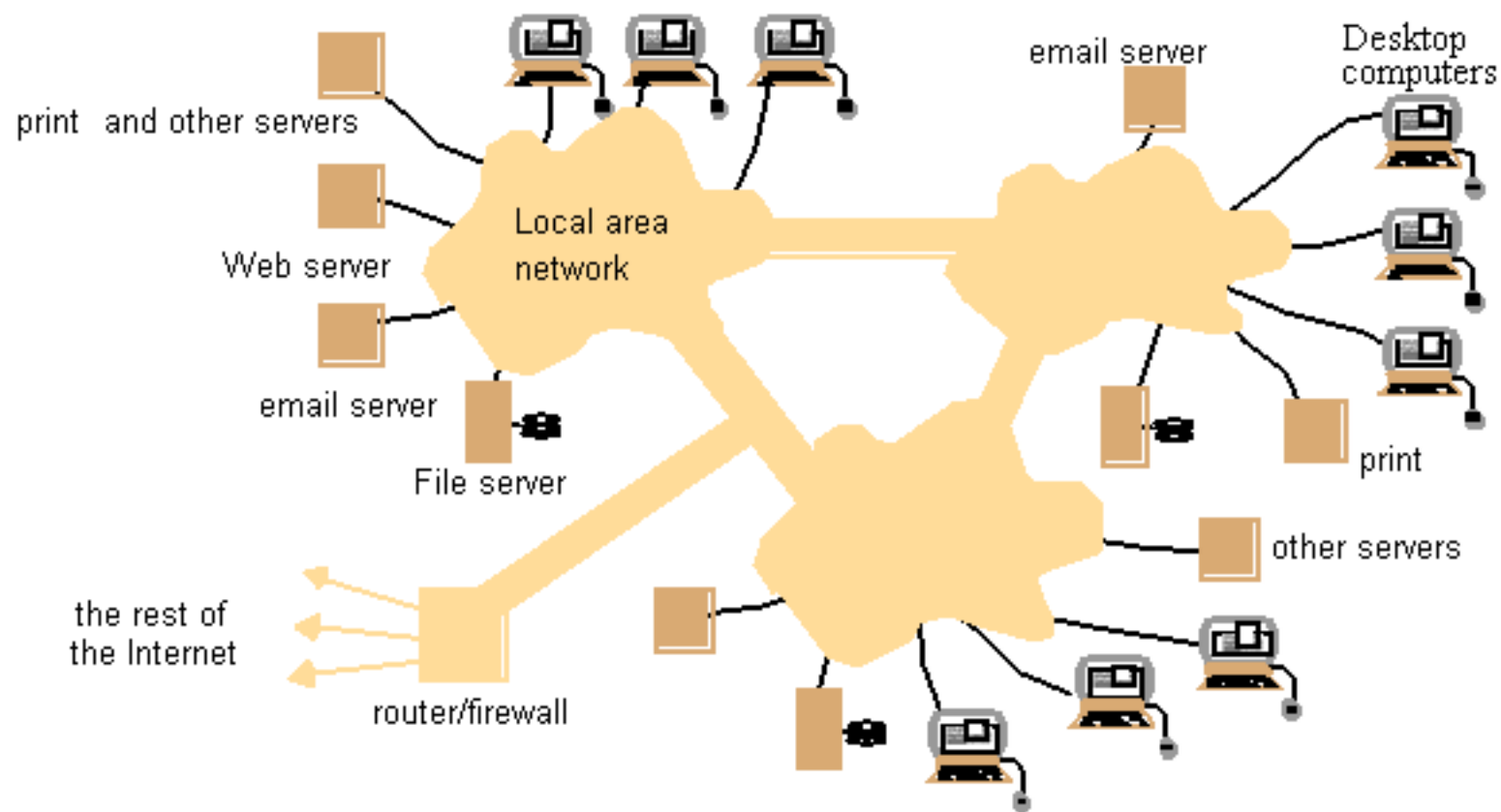
# CẤU TRÚC CLIENT - SERVER ĐA TẦNG (MULTI-TIER CLIENT-SERVER)

## Kiến trúc n tầng (n-tier)

- Thực hiện dịch vụ invokes từ các Application Server bên dưới và sau đó trả kết quả về cho Client
- Các dịch vụ trên máy chủ có thể : RPC, RMI,CORBA,DCOM,SOAP, Ajax, Web Service, REST v.v...

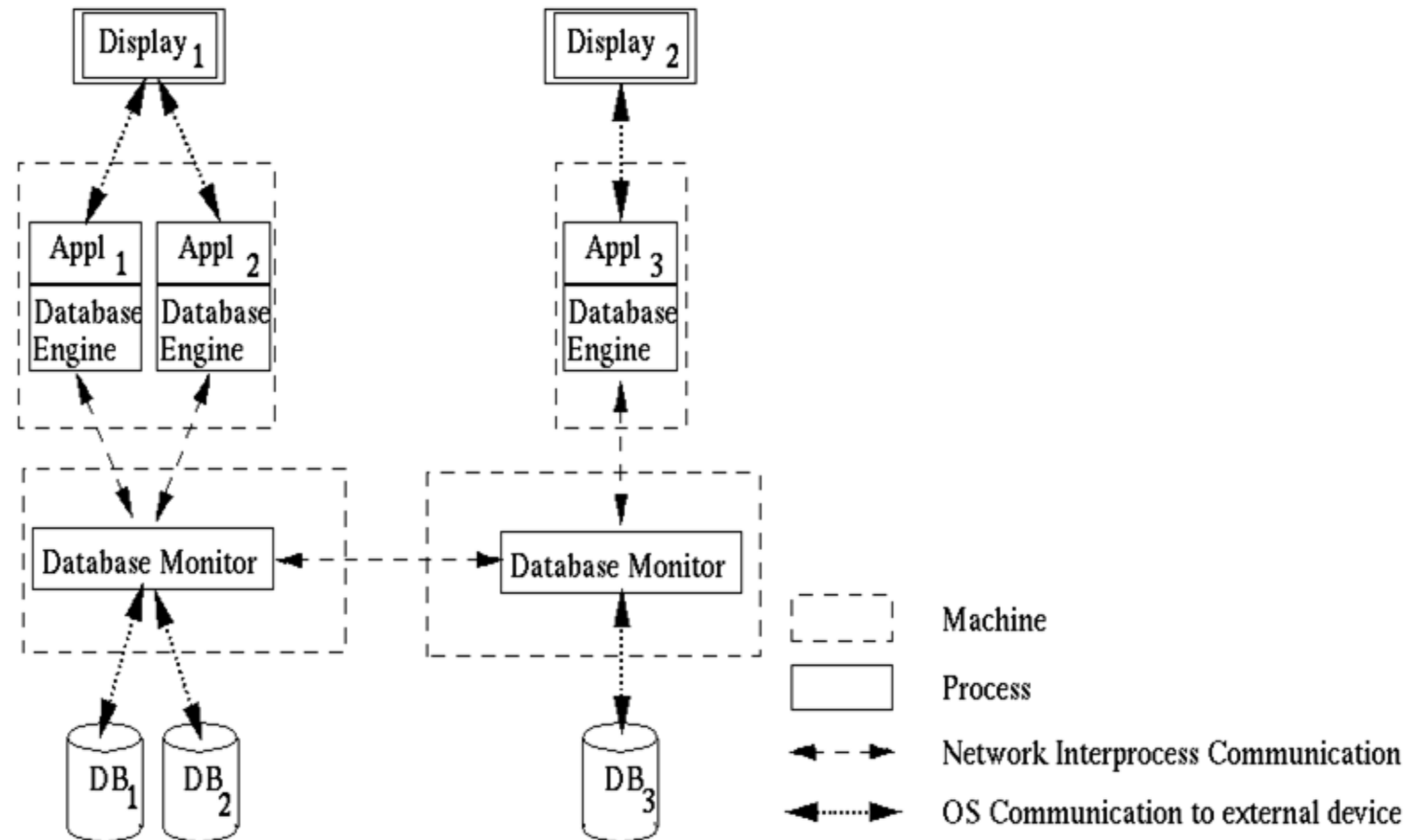
# VÍ DỤ VỀ HỆ PHÂN TÁN

## Mạng cục bộ (LAN Net)



# VÍ DỤ VỀ HỆ PHÂN TÁN

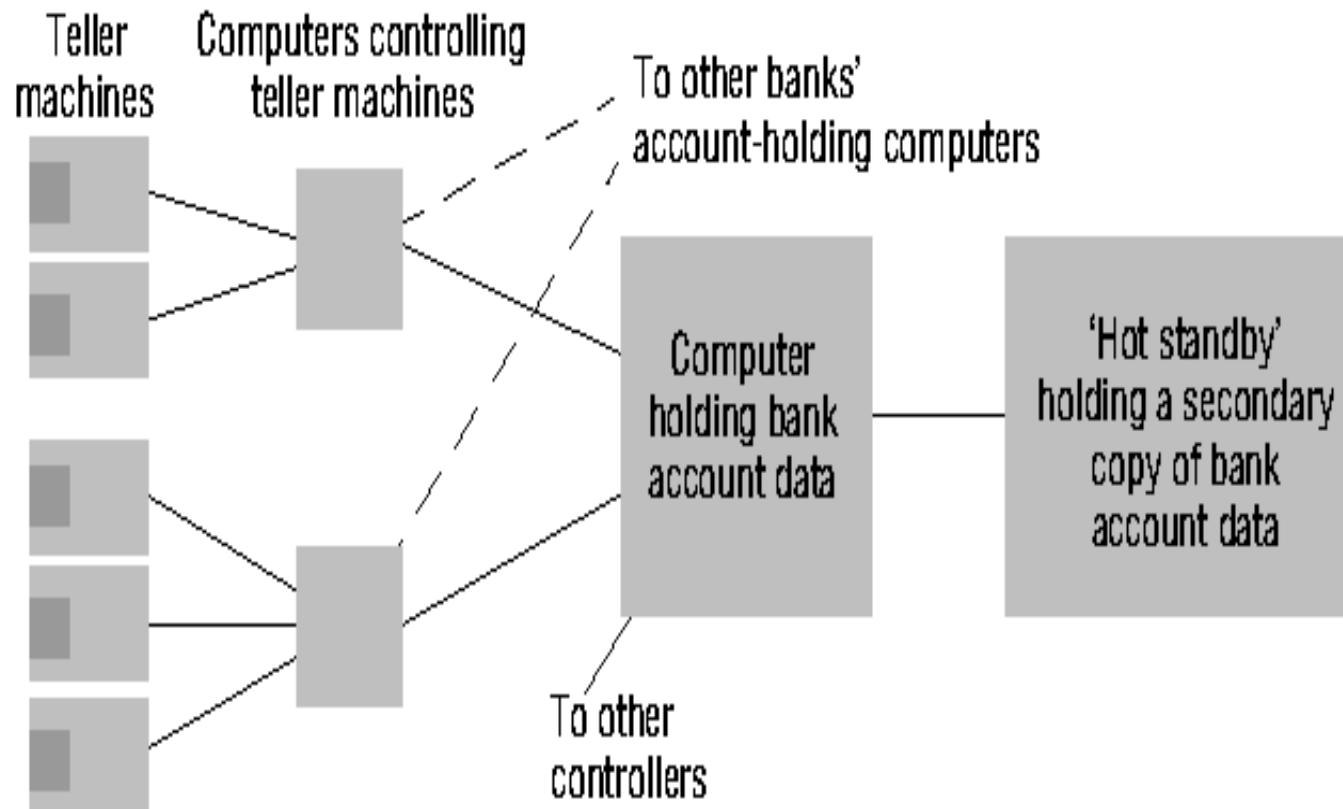
## Hệ thống quản trị CSDL





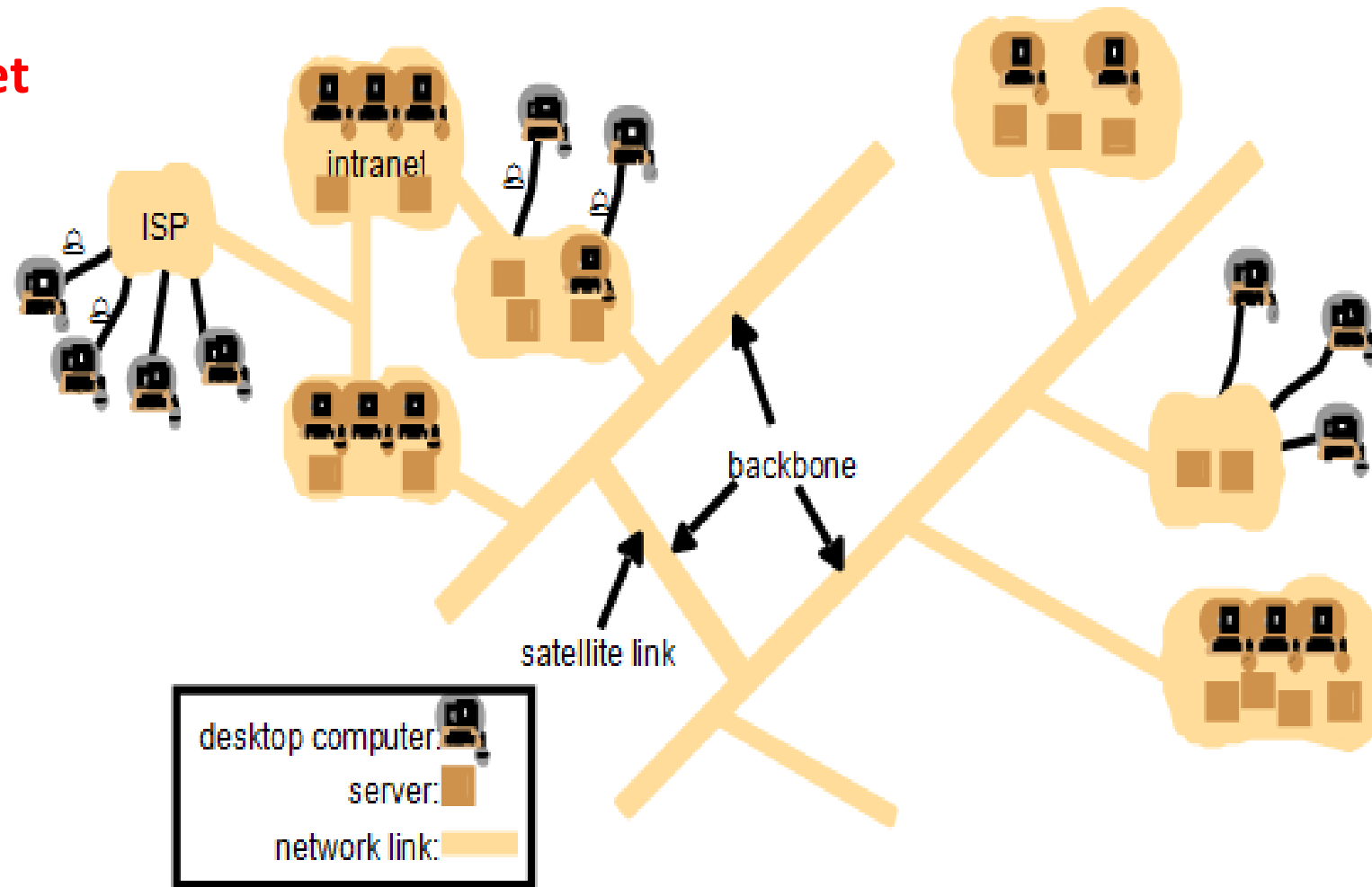
# VÍ DỤ VỀ HỆ PHÂN TÁN

## Cơ chế mạng điện thoại tự động (Automatic Teller Machine Network)



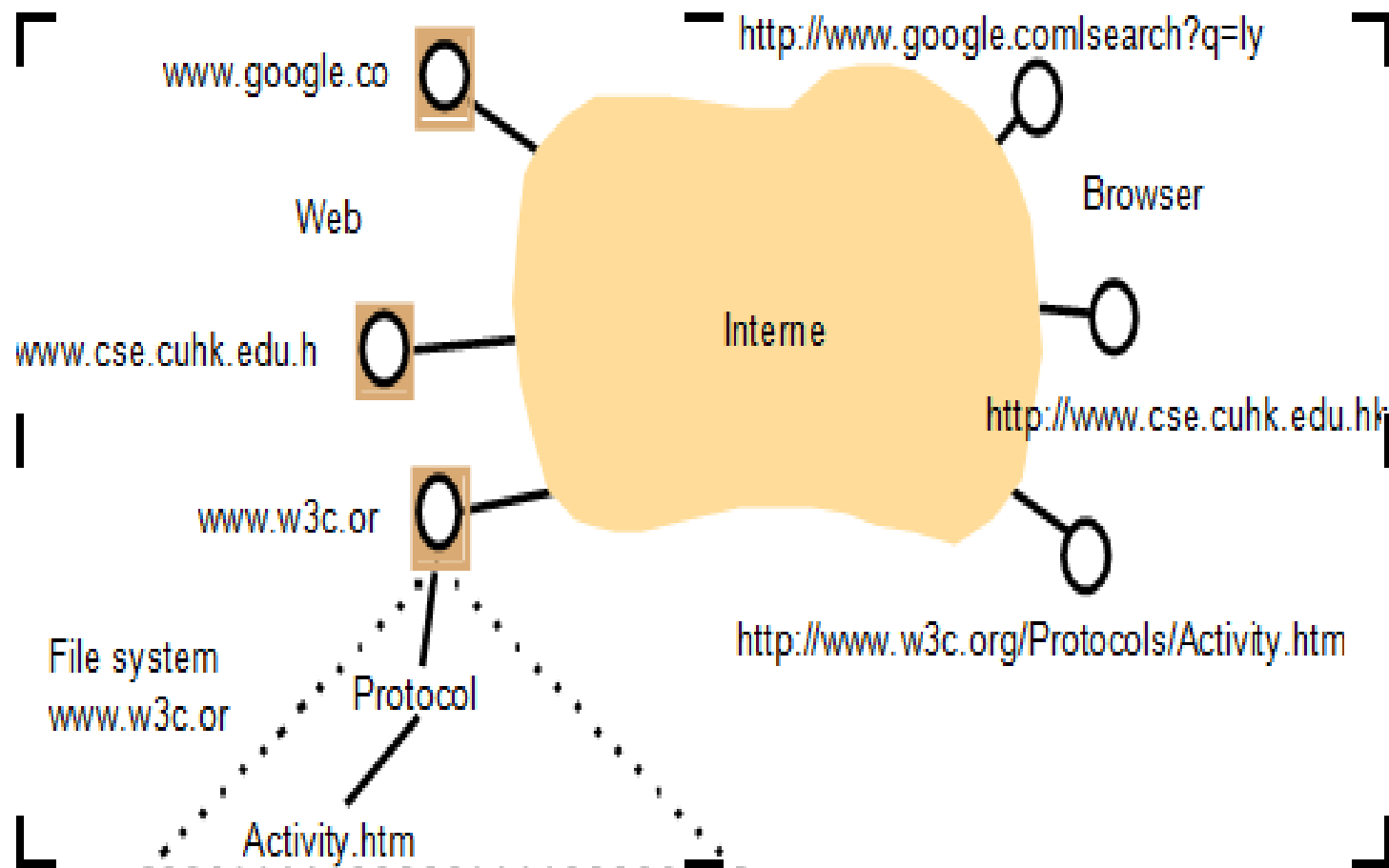
# VÍ DỤ VỀ HỆ PHÂN TÁN

## Mạng Internet



# VÍ DỤ VỀ HỆ PHÂN TÁN

## Ứng dụng Web Servers và trình duyệt Web



# VÍ DỤ VỀ HỆ PHÂN TÁN

## Searching Systems

