



UIT

TRƯỜNG ĐẠI HỌC
CÔNG NGHỆ THÔNG TIN

PHƯƠNG PHÁP PHÁT TRIỂN PHẦN MỀM HƯỚNG ĐỐI TƯỢNG

Thiết kế hướng đối tượng (p1 – Kiến trúc)

Nội dung

1. Kiến trúc hệ thống
2. Sơ đồ package
3. Sơ đồ component
4. Sơ đồ deployment
5. Một số mô hình kiến trúc phần mềm
 - Layers
 - Tiers

Nội dung

- 1. Kiến trúc hệ thống**
2. Sơ đồ package
3. Sơ đồ component
4. Sơ đồ deployment
5. Một số mô hình kiến trúc phần mềm
 - Layers
 - Tiers

Kiến trúc hệ thống

- ❖ Physical architecture
- ❖ Gắn các module với các phần cứng vật lý thích hợp

Kiến trúc hệ thống

❖ Các thành phần của hệ thống

■ Phần mềm

- Data storage: lưu trữ dữ liệu hệ thống
- Data access logic: giao tiếp với thành phần lưu trữ
- Application logic: xử lý các nghiệp vụ
- Presentation logic: Xử lý giao diện

■ Phần cứng

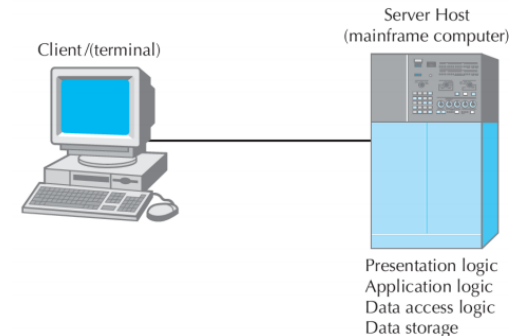
- Client: input/output (máy tính, smartphone, màn hình,...)
- Server: Trung tâm xử lý chính của hệ thống
- Network: kết nối client-server

Kiến trúc hệ thống

❖ Các kiểu kiến trúc vật lý

■ Server – Based

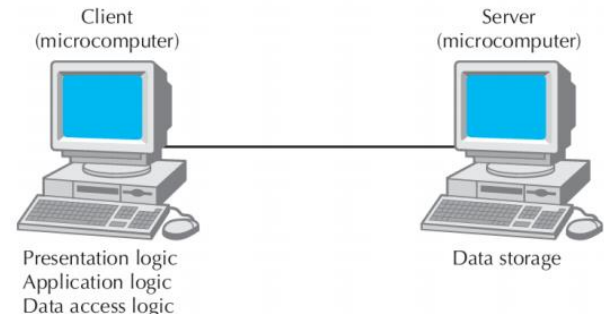
- Client: input/output
- Các xử lý chủ yếu trên server



■ Client – Based

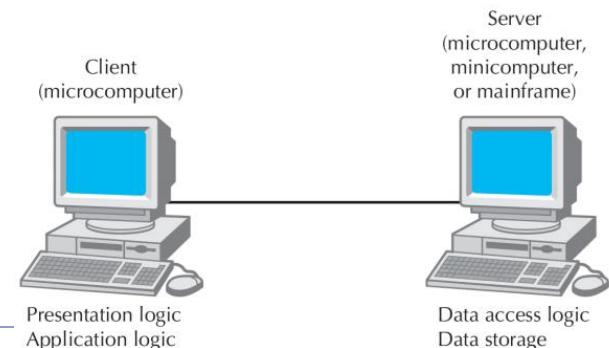
Các xử lý chủ yếu trên

- Client: Nơi xử lý chính
- Server đảm nhận lưu trữ



■ Client – Server

- Server: quản lý dữ liệu, xử lý quan trọng
- Client: xử lý đồ họa, và 1 số quy trình nghiệp vụ

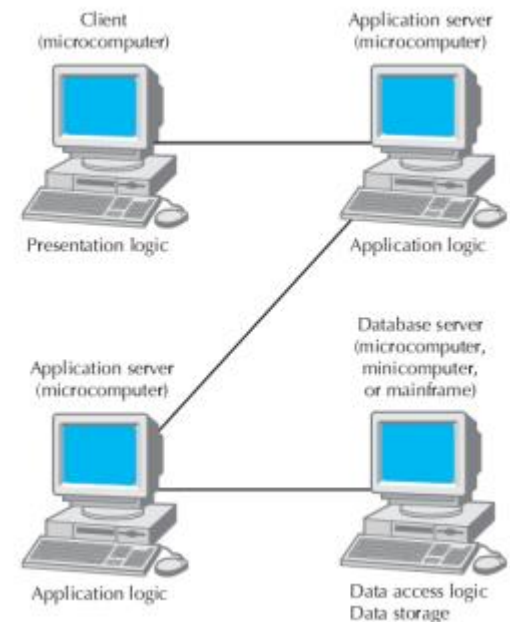
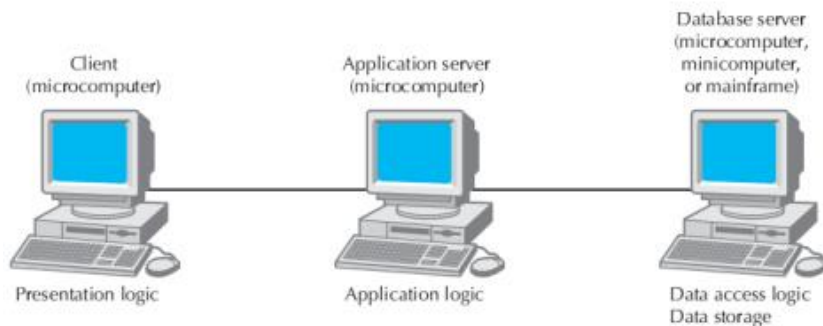


Kiến trúc hệ thống

❖ Các kiểu kiến trúc vật lý

▪ Client – Server nhiều tầng

- Client: xử lý đồ họa
- Server: gồm tầng để tăng tốc độ, hiệu năng xử lý
- Có thể có 3 tầng, 4 tầng hoặc n – tầng (n tiers)



Lựa chọn kiến trúc

❖ Chi phí của kiến trúc

- Phần cứng cần phải có trong kiến trúc
- Bảng thông, các dịch vụ, cài đặt khác

❖ Chi phí phát triển phần mềm

- Kiến trúc càng phức tạp phải có thêm nhiều lớp đảm nhận nhiệm vụ liên lạc giữa các bên
- Vấn đề bảo mật
- Khả năng mở rộng

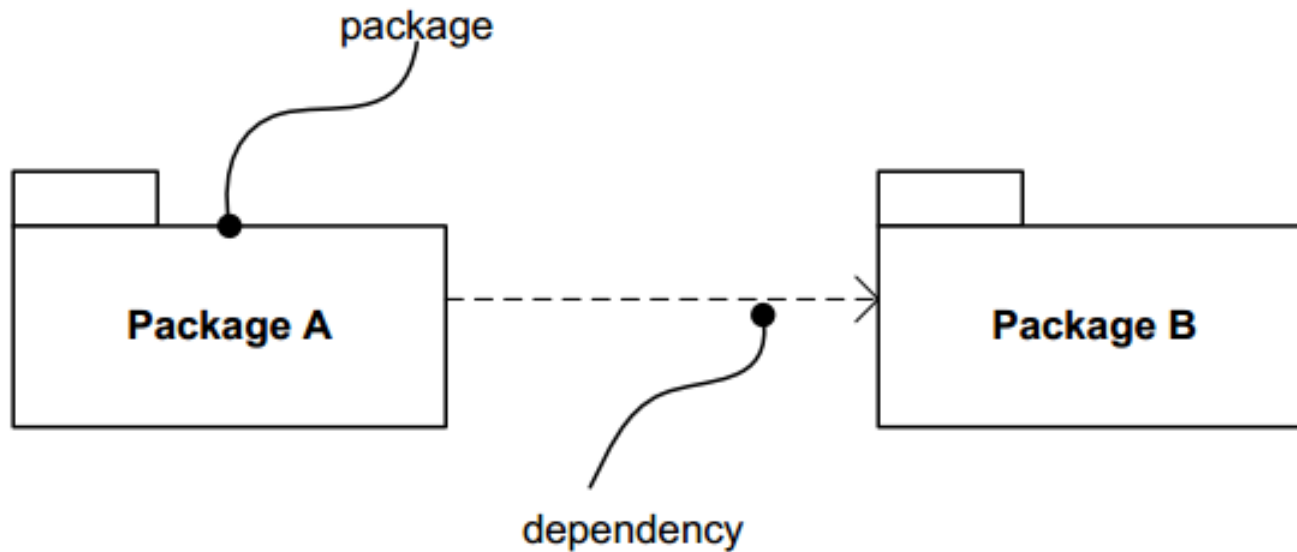
Nội dung

1. Kiến trúc hệ thống
- 2. Sơ đồ package**
3. Sơ đồ component
4. Sơ đồ deployment
5. Một số mô hình kiến trúc phần mềm
 - Layers
 - Tiers

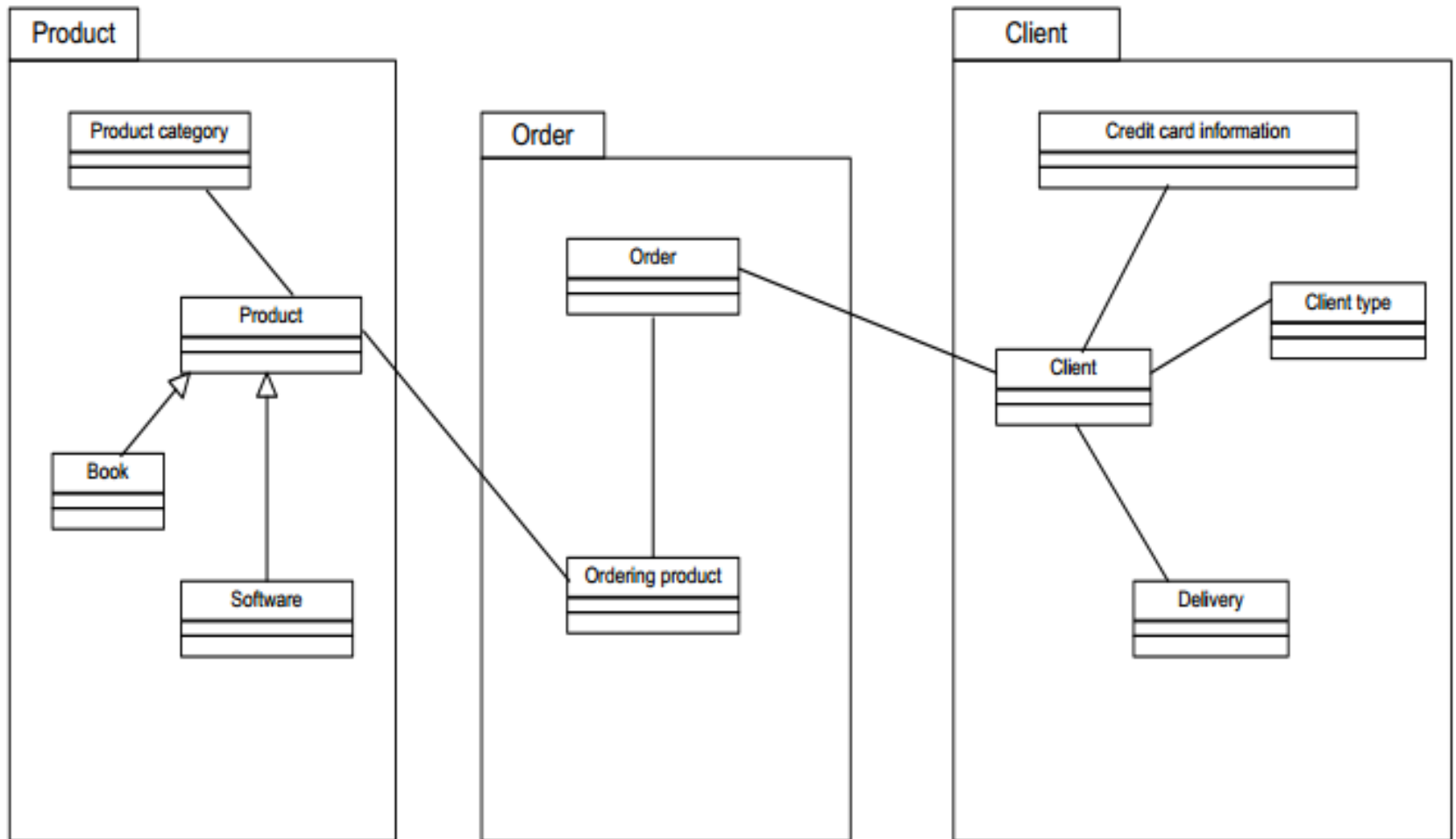
Package diagram

- ❖ Nhóm các thành phần cơ bản (lớp) vào các nhóm có cùng vai trò hoặc ý nghĩa
- ❖ Dễ bắt và bảo trì hệ thống dễ dàng, hiệu quả hơn
- ❖ Package có thể chứa các package khác

Package element

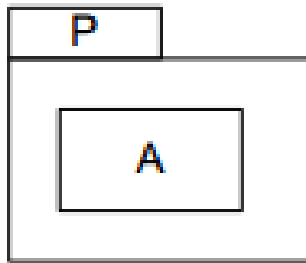
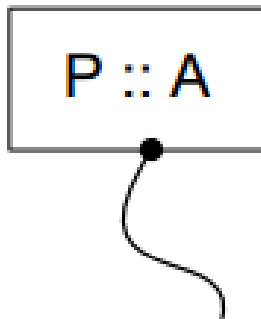


Package chứa package



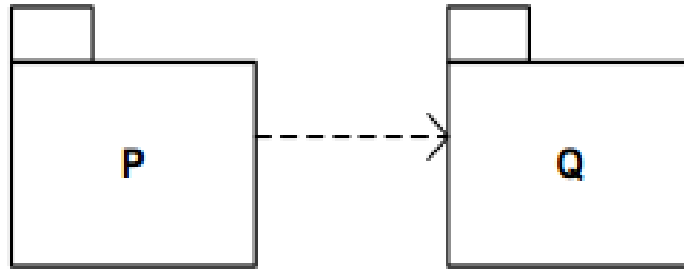
Đặt tên/gọi tên

- ❖ Tên class có thể đi kèm với tên package
- ❖ Cú pháp; [Tên package]::[tên class]

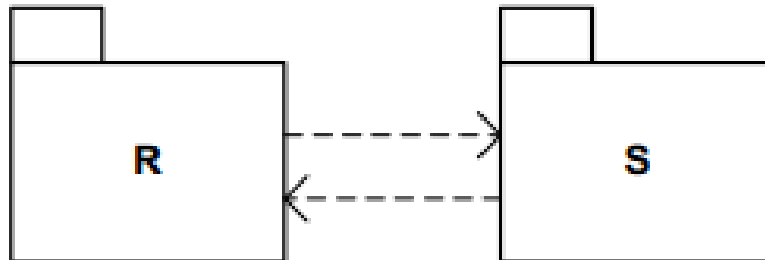


Mối quan hệ

❖ Package P sử dụng package Q

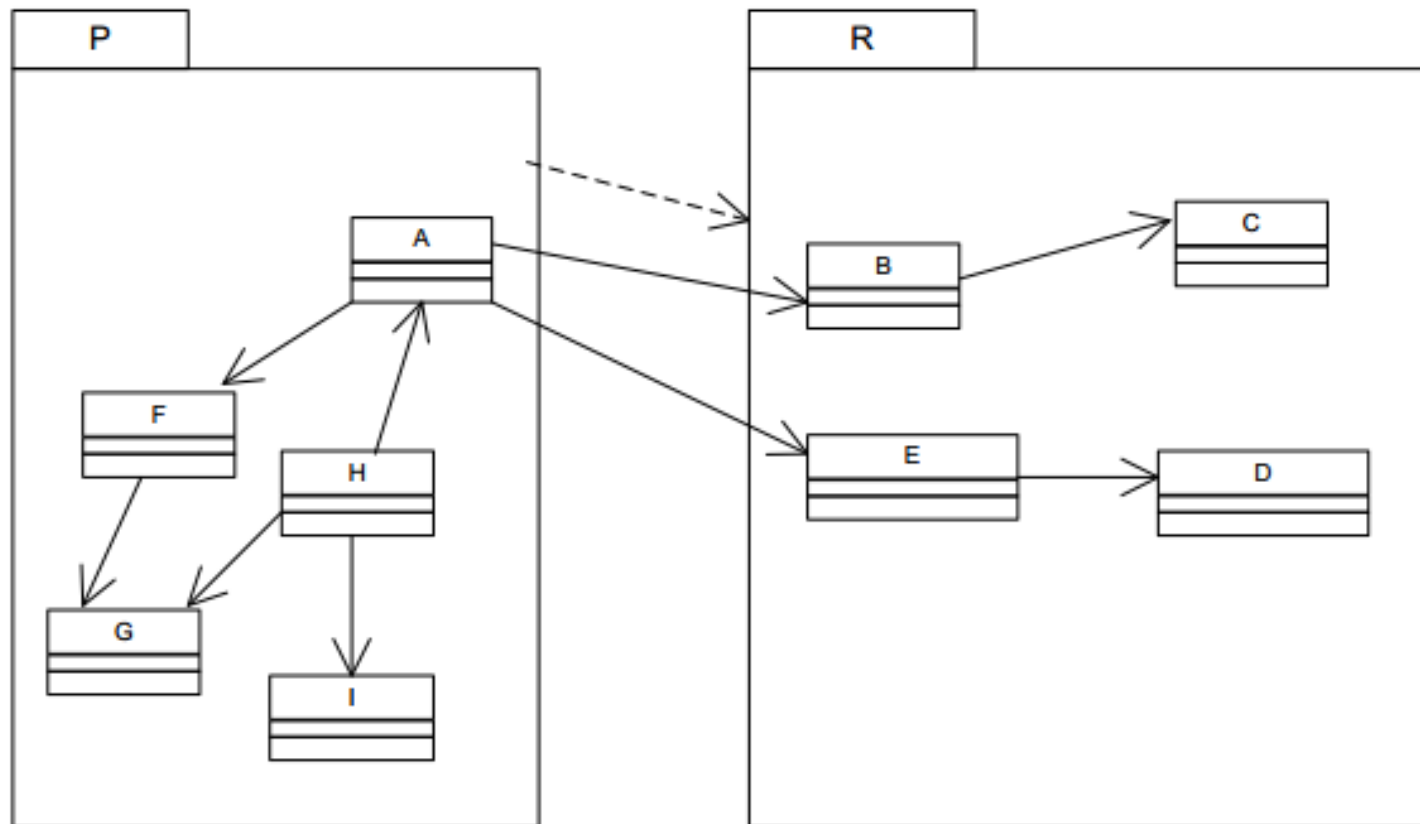


❖ R sử dụng S, S sử dụng R



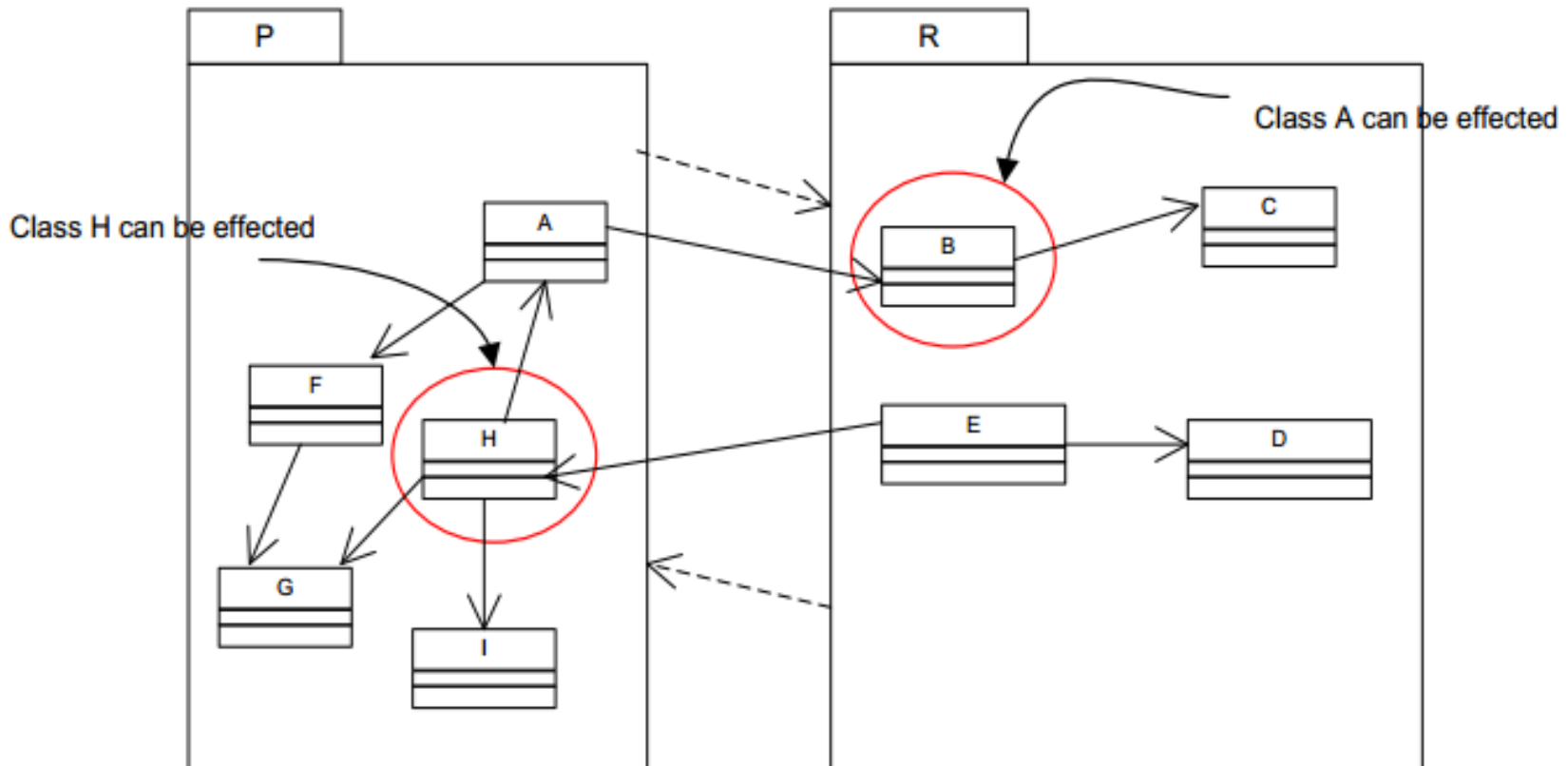
Mối quan hệ

- ❖ Though classes in package P have changes
- ❖ but package R is not effective

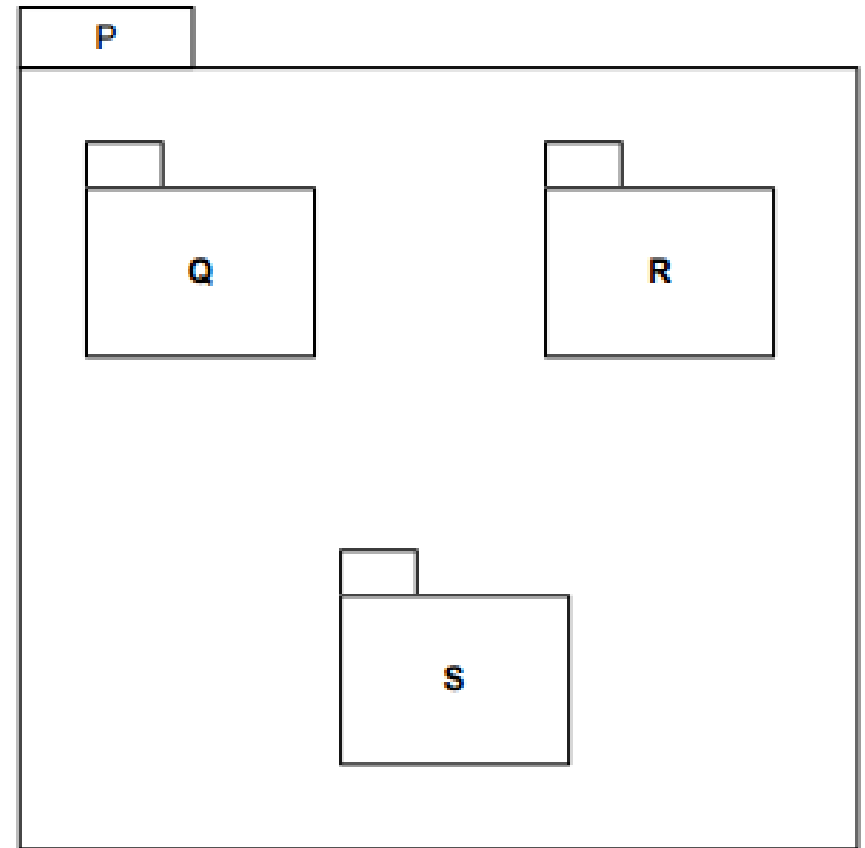
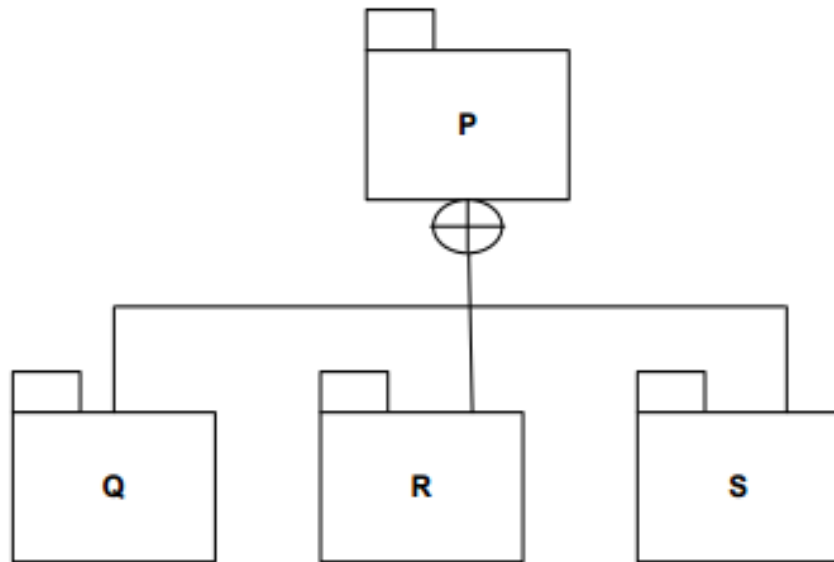


Mối quan hệ

- ❖ Class in package P changes, maybe it effects to other package



Thể hiện package con



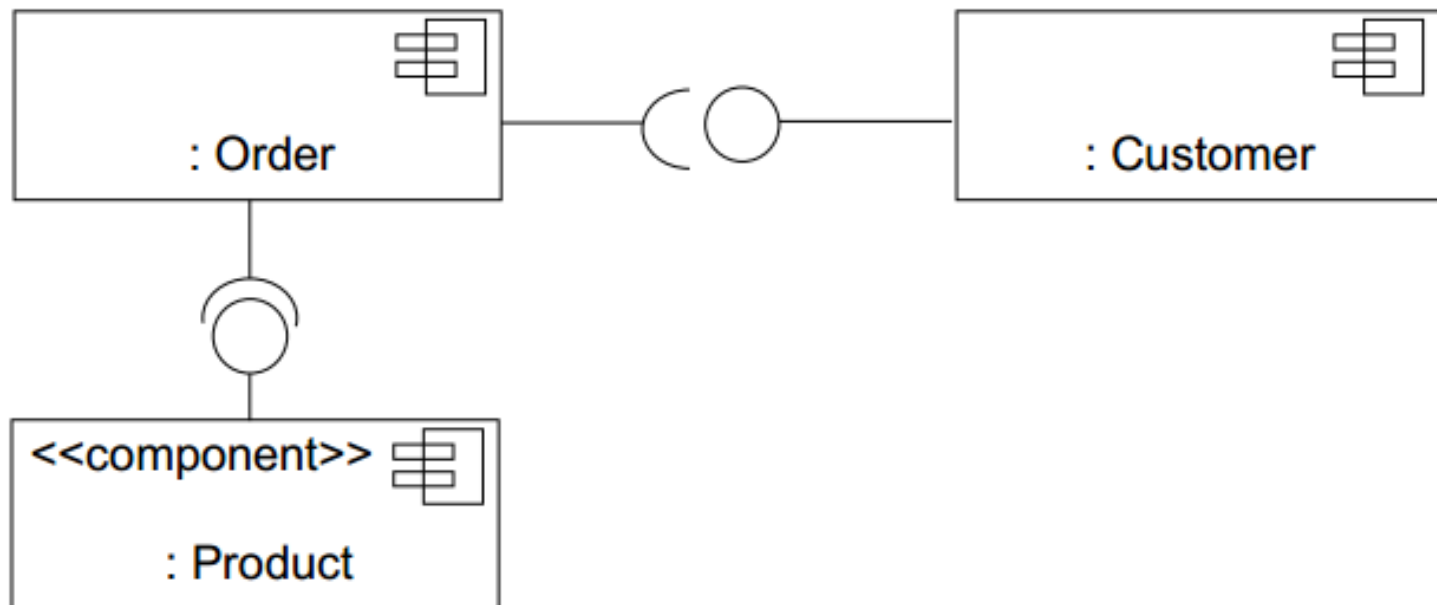
Can display the child package

Nội dung

1. Kiến trúc hệ thống
2. Sơ đồ package
- 3. Sơ đồ component**
4. Sơ đồ deployment
5. Một số mô hình kiến trúc phần mềm
 - Layers
 - Tiers

Component diagram

- ❖ Thể hiện cấu trúc, mối quan hệ giữa các component hoặc các hệ thống con



Component & Interface

❖ Component

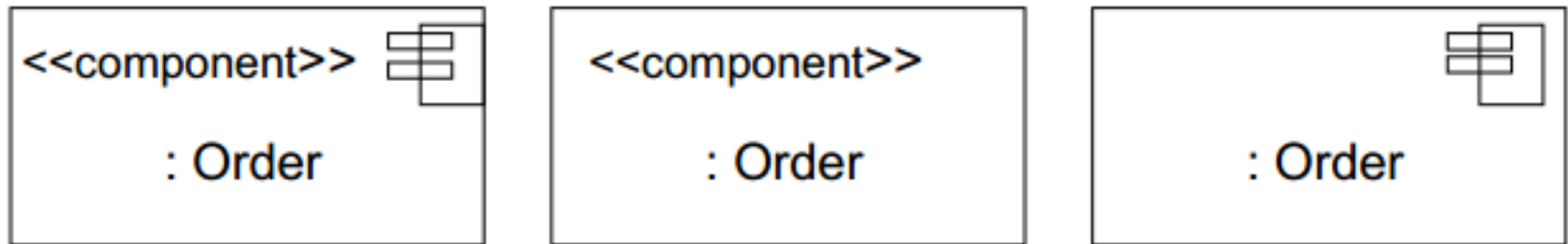
- Tự động đóng gói các đơn vị bên trong software
- Physical phase
 - business component
 - process component
- Logical phase: EJB, CORBA, COM+, .NET, WSDL
- Có ít nhất 1 interface public và giao tiếp với bên ngoài

❖ Interface

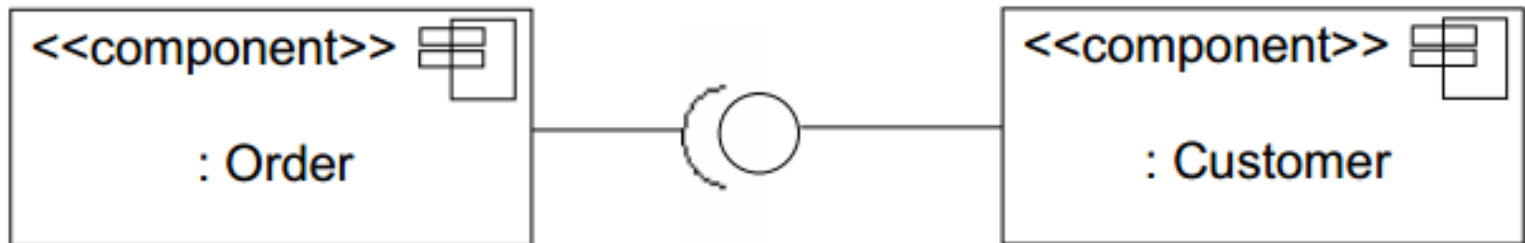
- Provided: hỗ trợ
- Required: yêu cầu

Ký hiệu

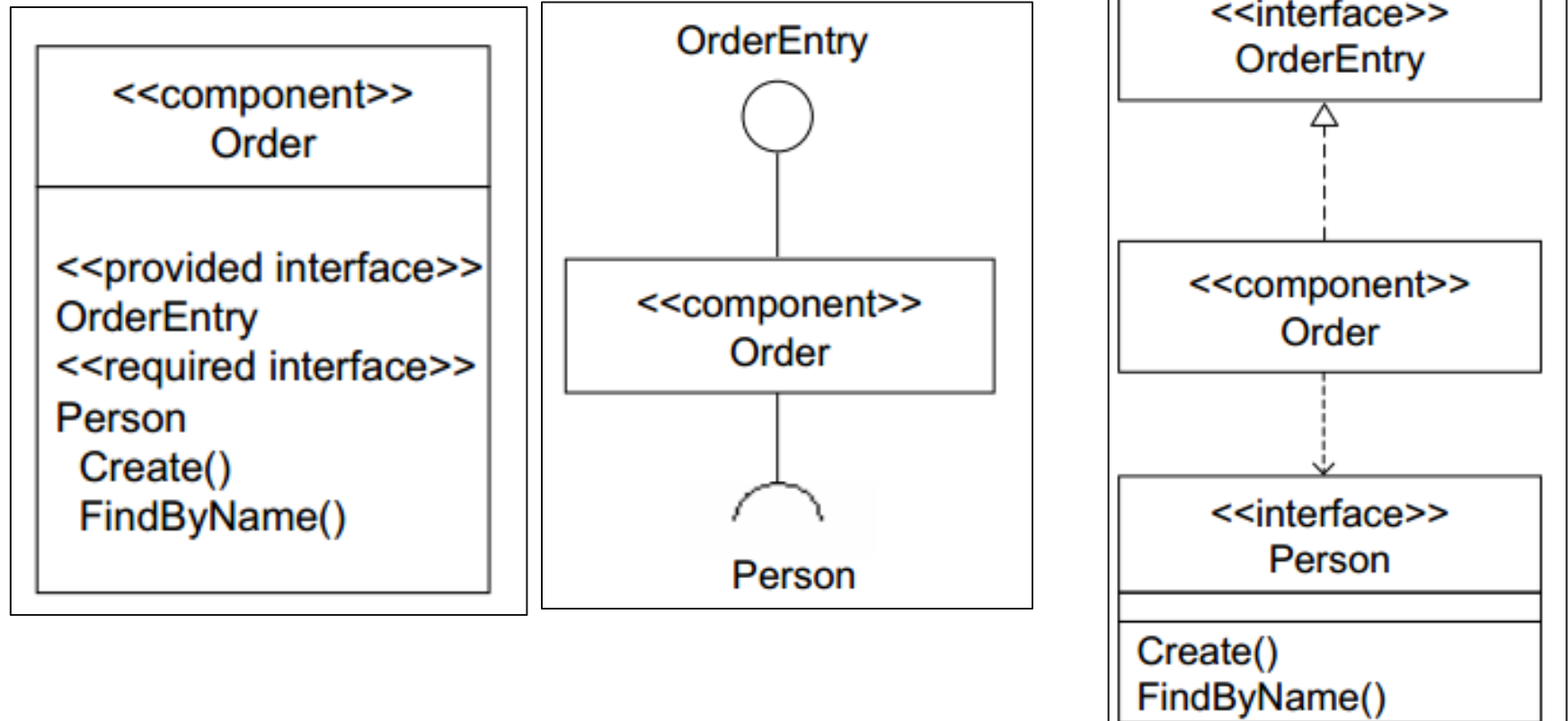
❖ Component



❖ Interfaces

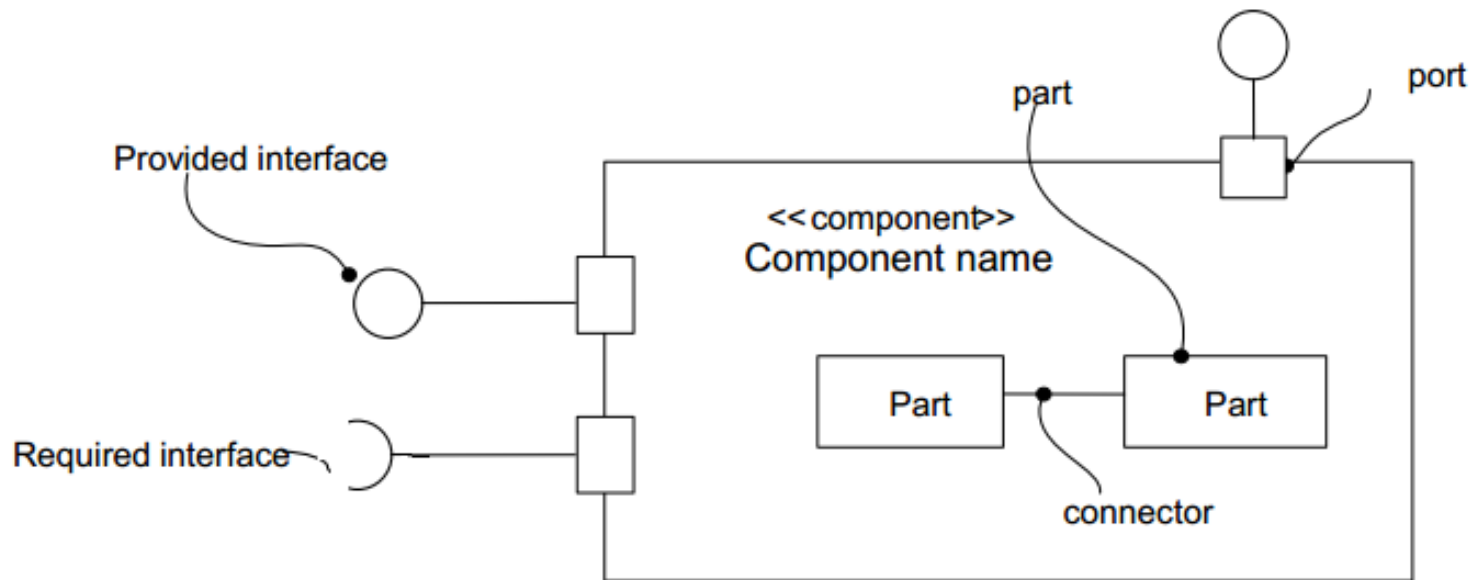


Interface



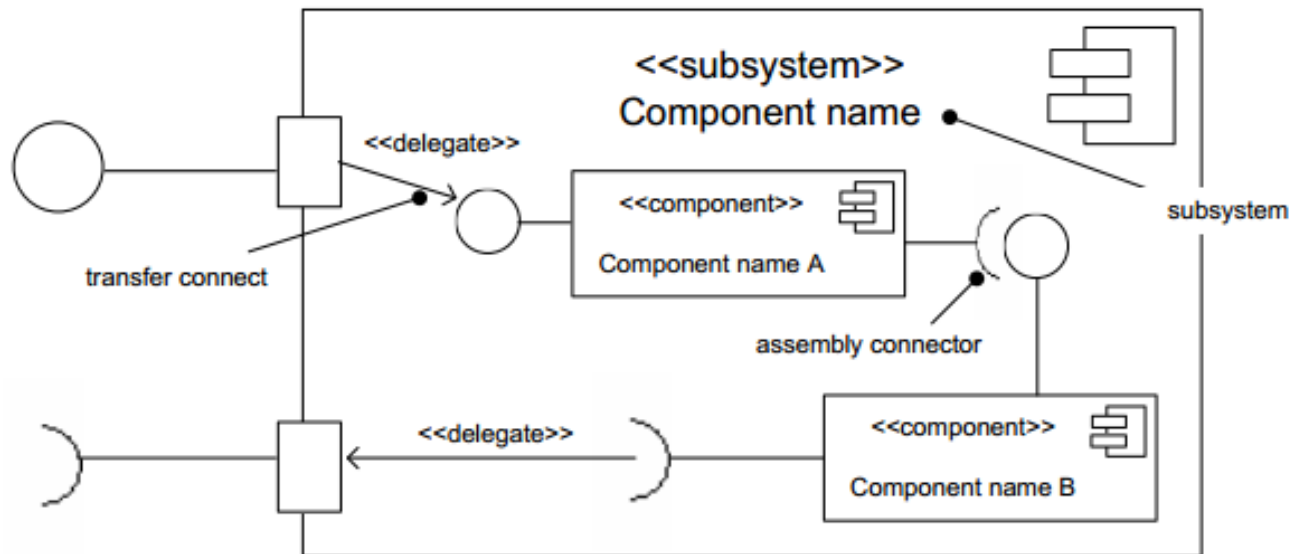
Bên trong 1 component

- ❖ **Part:** Thành phần chứa thành phần khác
- ❖ **Port:** Thành phần ở biên, giúp giao tiếp giữa thành phần bên trong và bên ngoài
- ❖ **Connector:** Nối giữa các part và có thể bổ sung chỉ số



Trong subsystem

- ❖ **Transfer connector:** Chuyển yêu cầu đến các interface public với bên ngoài hoặc interface bên trong
- ❖ **Assembly connector:** Liên kết giữa require interface và provide interface



Nội dung

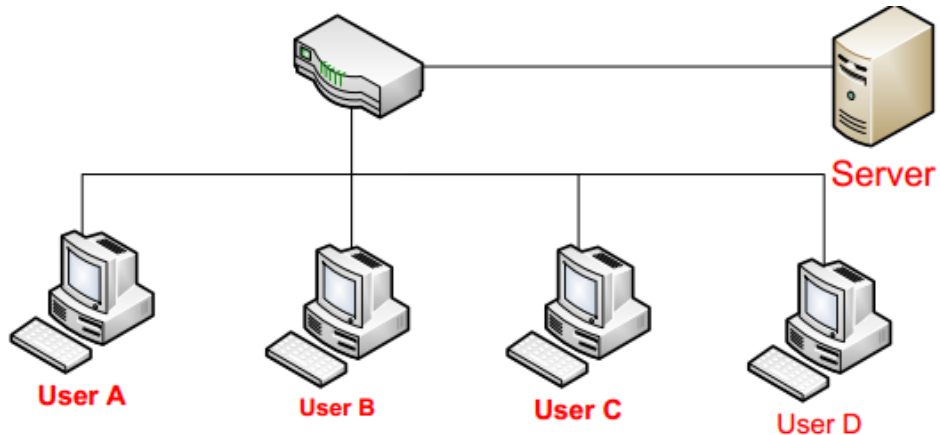
1. Kiến trúc hệ thống
2. Sơ đồ package
3. Sơ đồ component
- 4. Sơ đồ deployment**
5. Một số mô hình kiến trúc phần mềm
 - Layers
 - Tiers

Deployment diagram

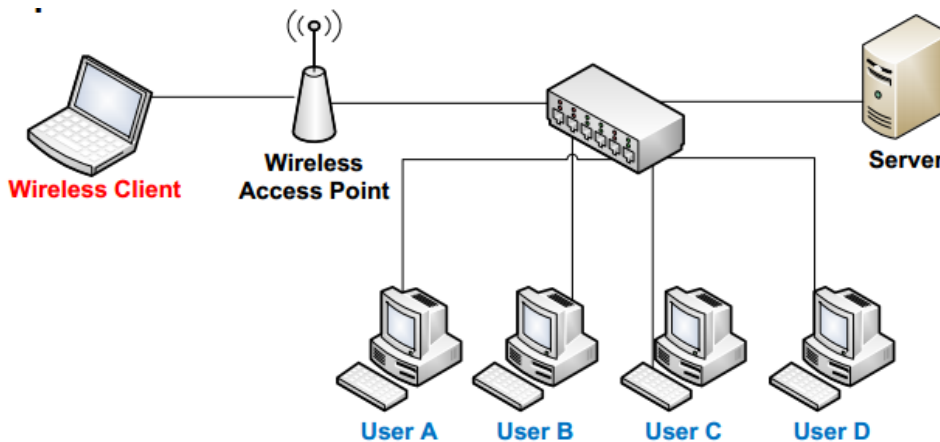
- ❖ Mô tả kiến trúc cài đặt vật lý các thành phần bên trong hệ thống và tương tác giữa chúng
- ❖ Cho biết mối quan hệ giữa phần mềm và phần cứng
- ❖ Được sử dụng trong Deployment View

Deployment diagram

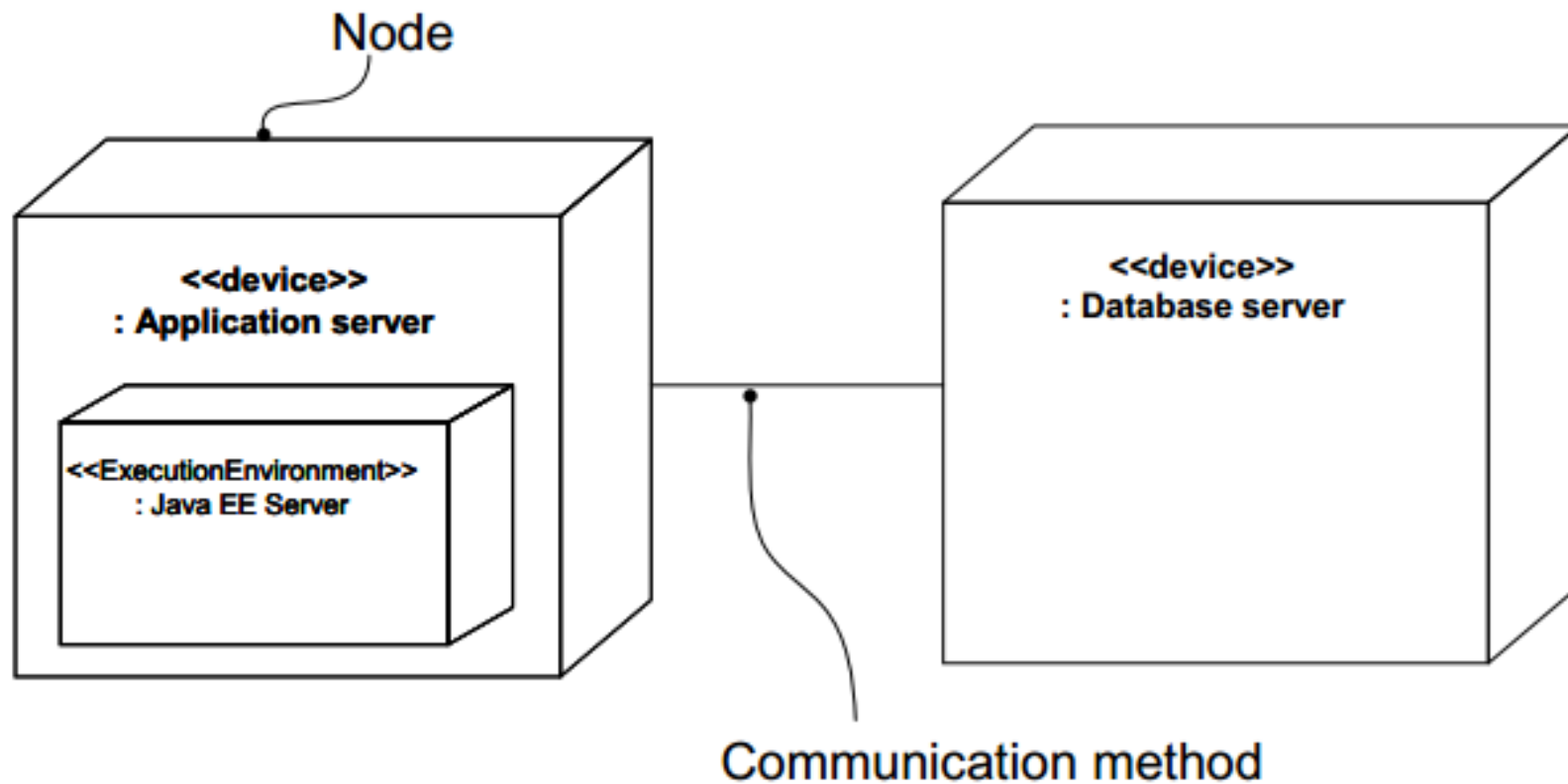
❖ LAN



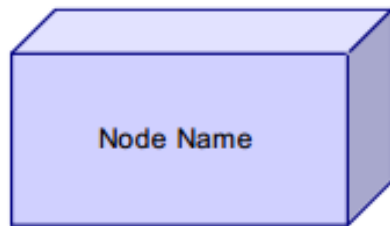
❖ LAN – WLAN



Element



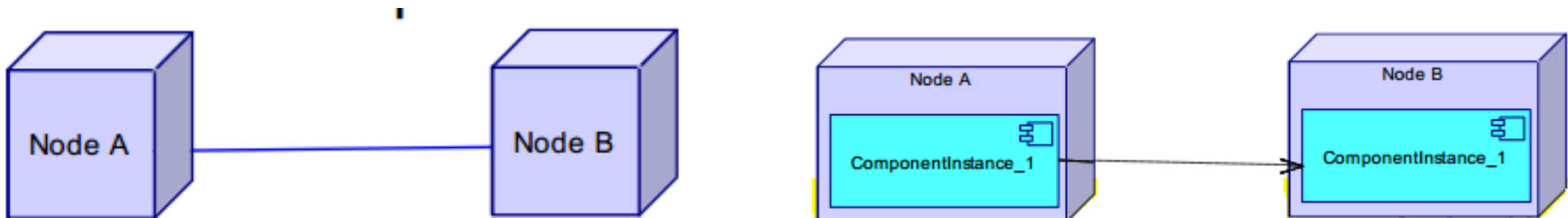
Element



Node



Component instance



Quan hệ giữa các thành phần

Node

❖ Là các tài nguyên trong hệ thống

- Server
- PC
- Printer
- Scanner
- Router

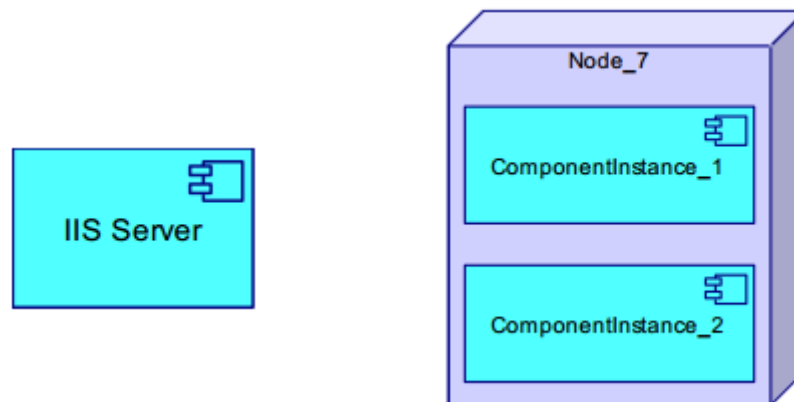


- ❖ Stereotype cho biết loại phần cứng: device, mobile device, application server, web server, database server, client workstation
- ❖ Mỗi node có thể có nhiều component

Node

❖ Component instance

- Là một thể hiện của một component (trong component diagram) sẽ được cài đặt trong node
- Thuộc tính của Component Instance
 - Chỉ số
 - Các component



Quan hệ giữa các thành phần

❖ Giữa

- Node - Node
- Node - Component Instance
- Component Instance và Component Instance

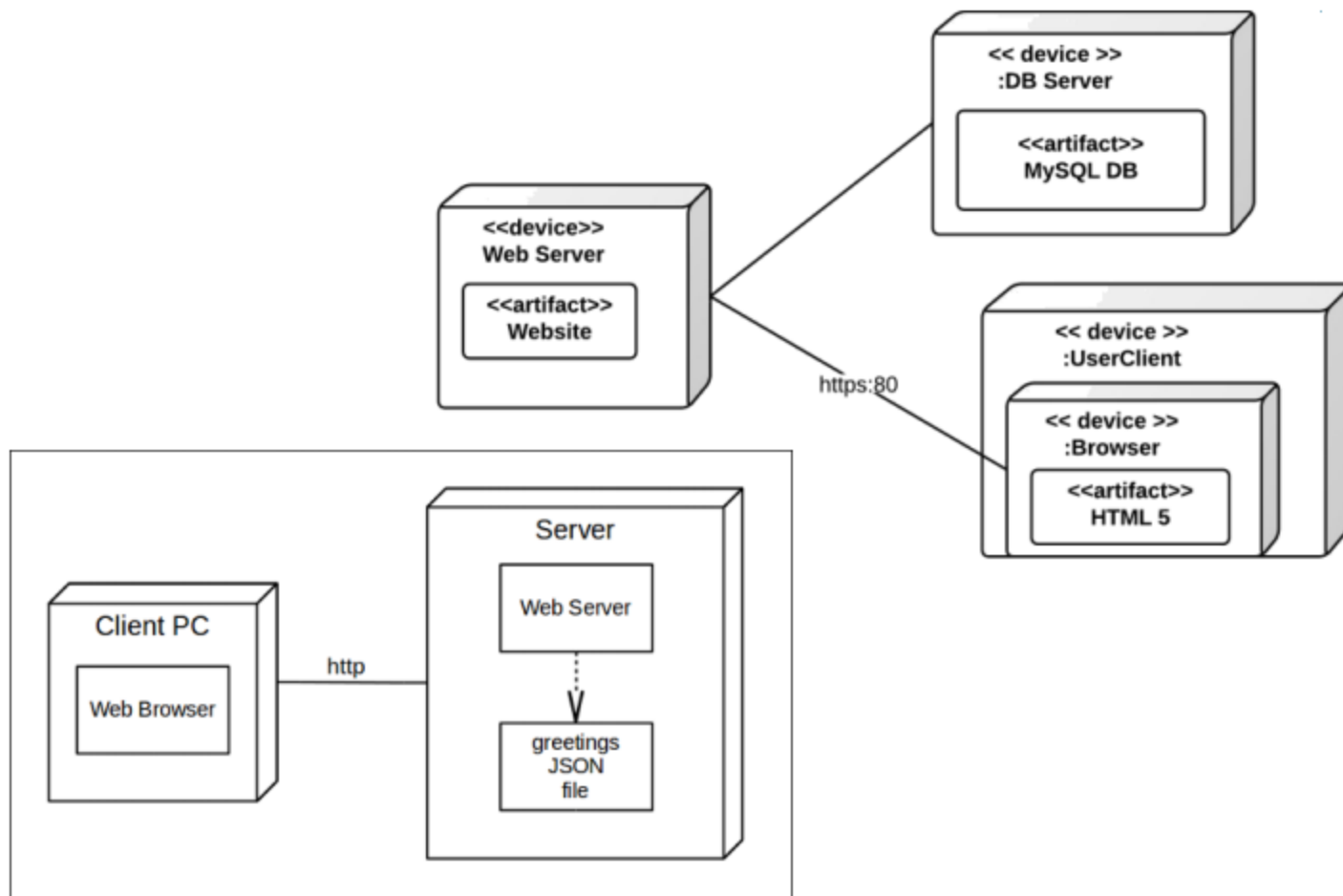
❖ Loại

- Association
- Dependency

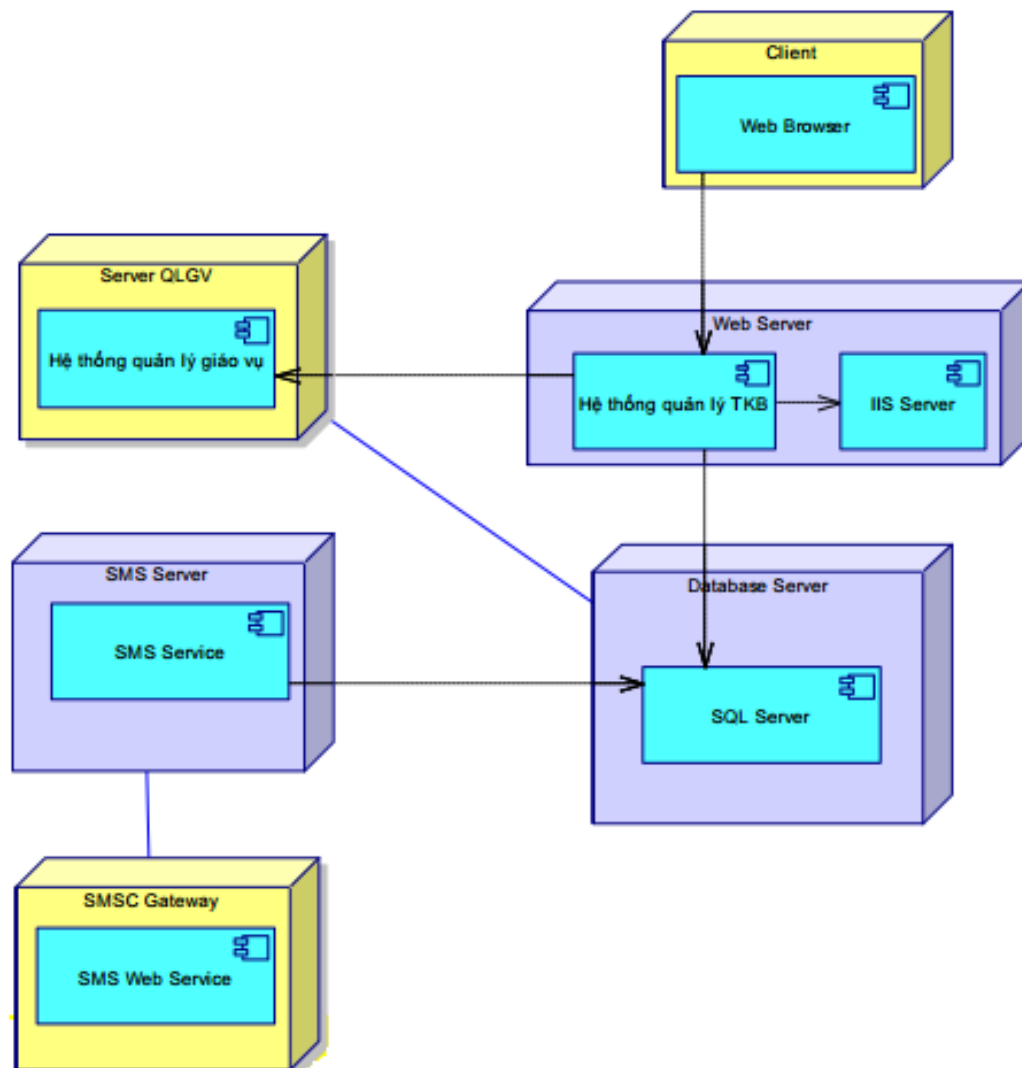
Ví dụ

- ❖ HT quản lý thời khóa biểu
 - Giáo vụ Khoa nhập thời khóa biểu, cập nhật thời khóa biểu của tất cả các lớp
 - Sinh viên tra cứu thời khóa biểu
 - Giảng viên tra cứu thời khóa biểu giảng dạy

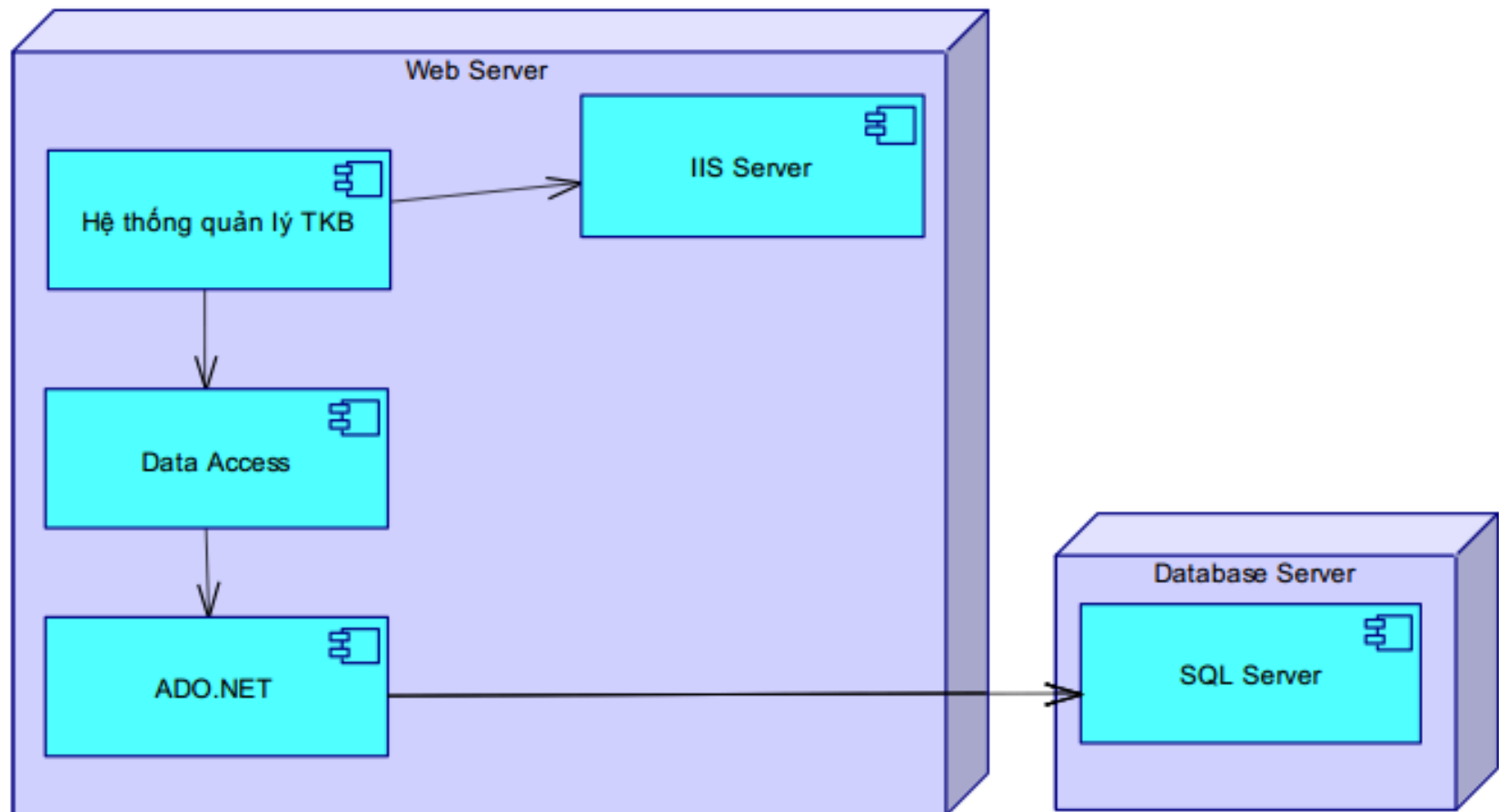
Ví dụ



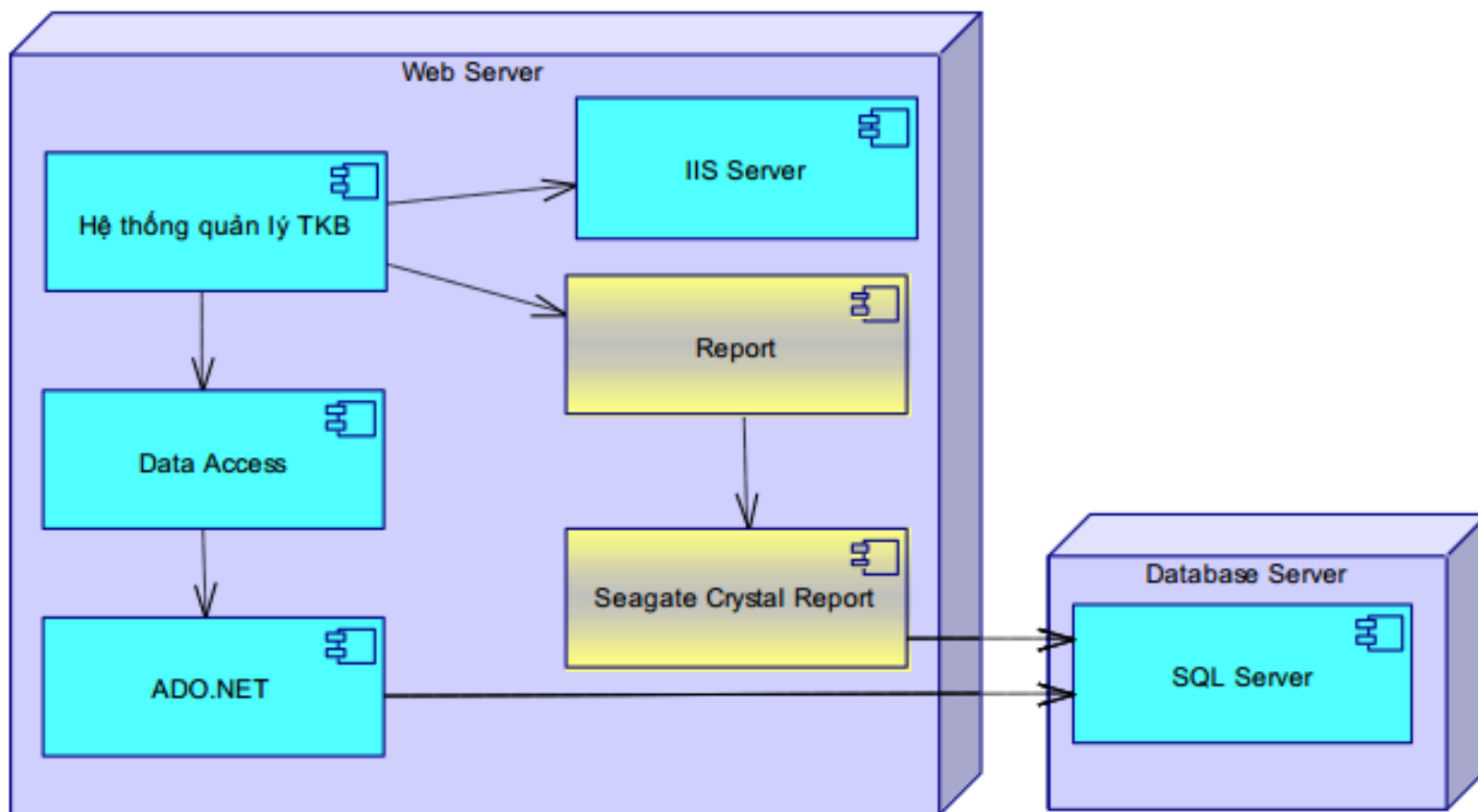
Ví dụ



Ví dụ



Ví dụ



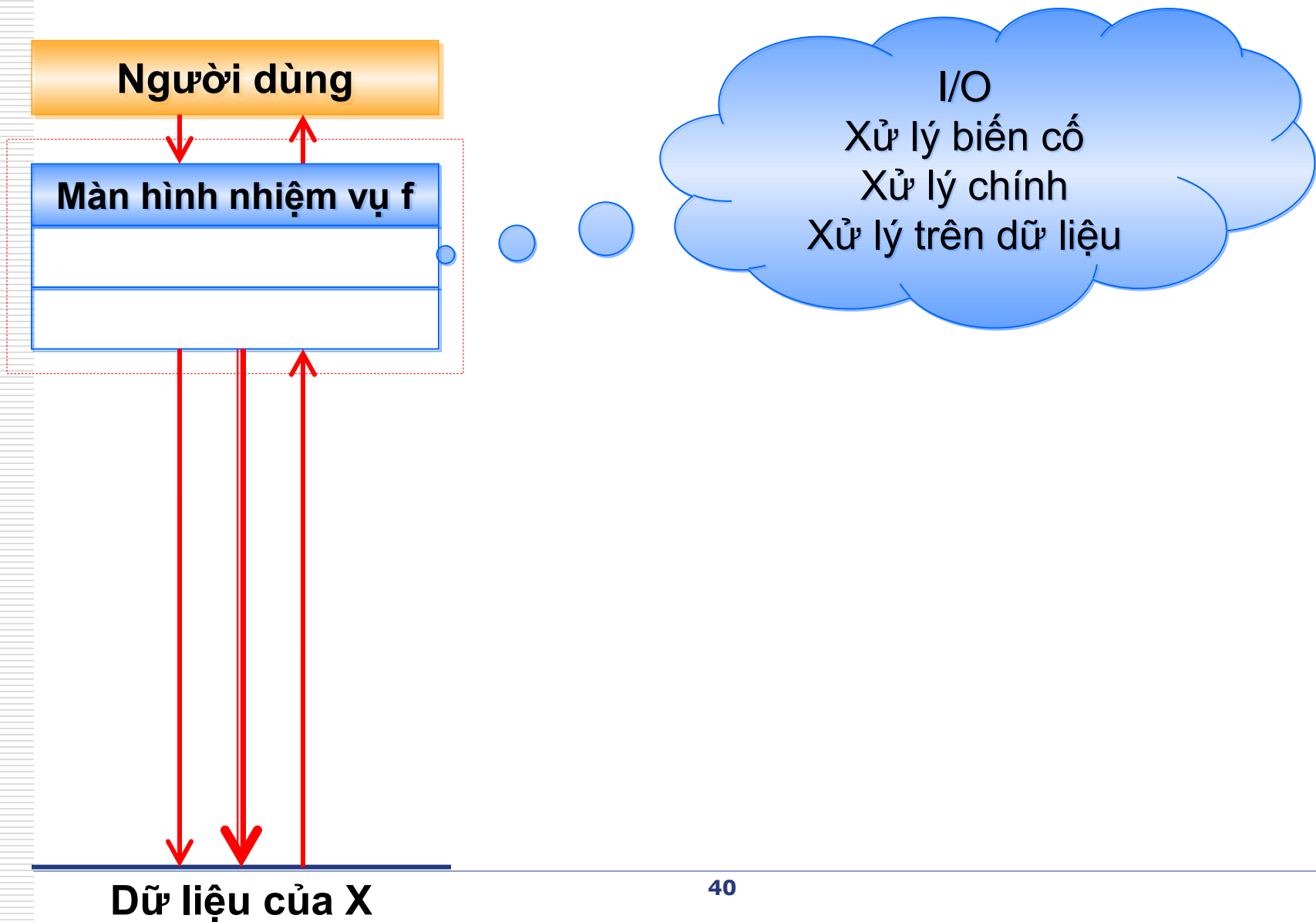
Nội dung

1. Kiến trúc hệ thống
2. Sơ đồ package
3. Sơ đồ component
4. Sơ đồ deployment
- 5. Một số mô hình kiến trúc phần mềm**
 - **Layers**
 - **Tiers**

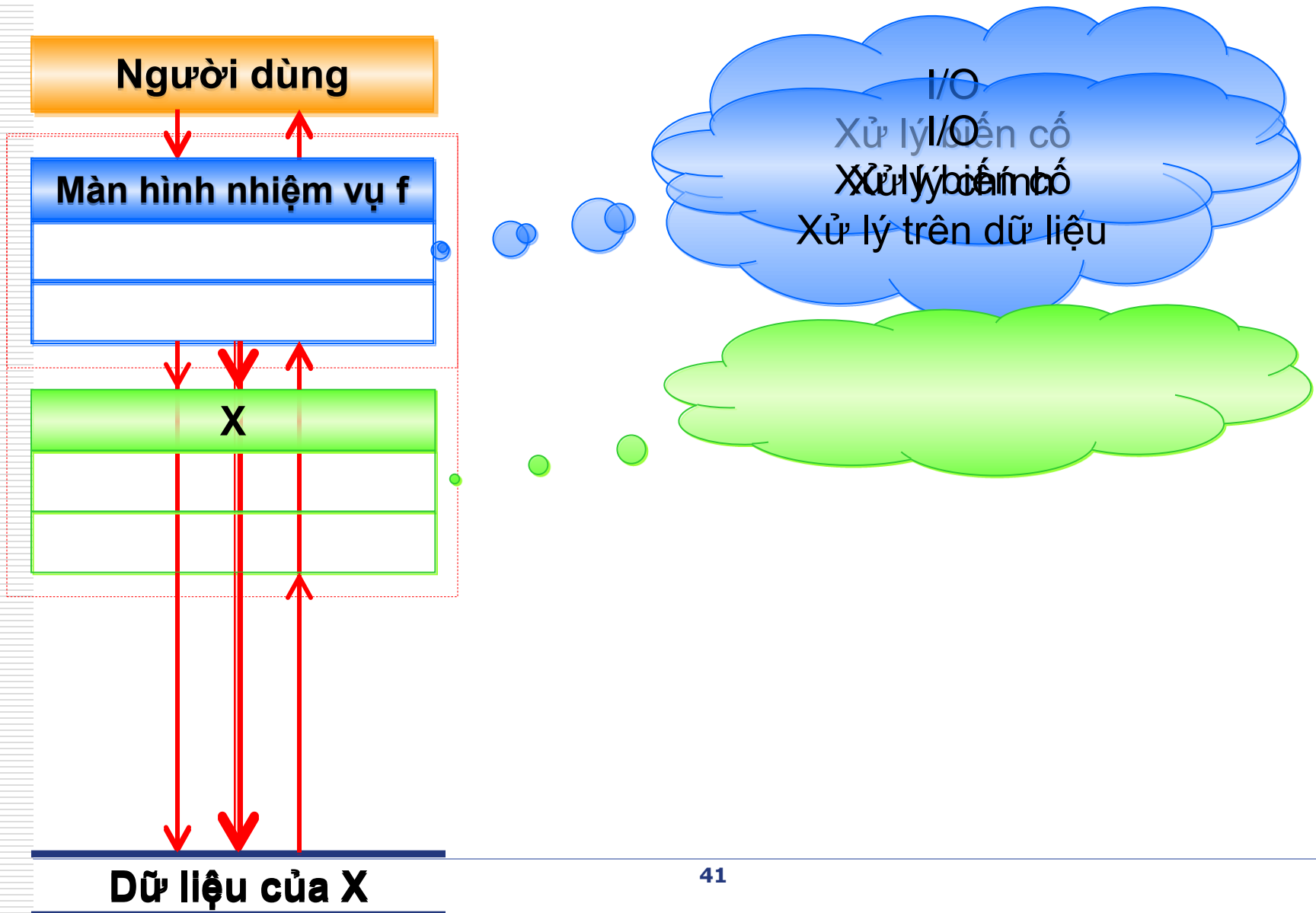
Nội dung

1. Kiến trúc hệ thống
2. Sơ đồ package
3. Sơ đồ component
4. Sơ đồ deployment
- 5. Một số mô hình kiến trúc phần mềm**
 - **Layers**
 - **Tiers**

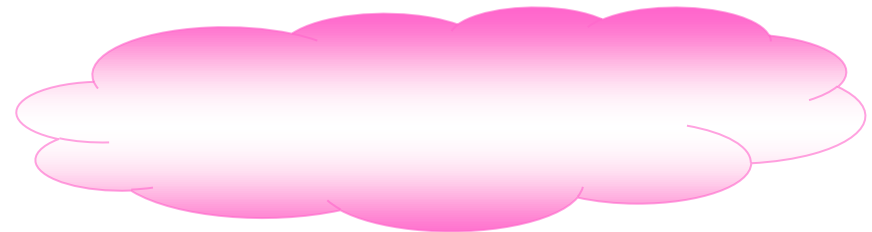
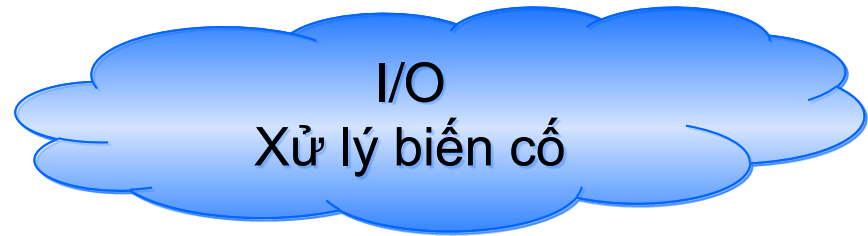
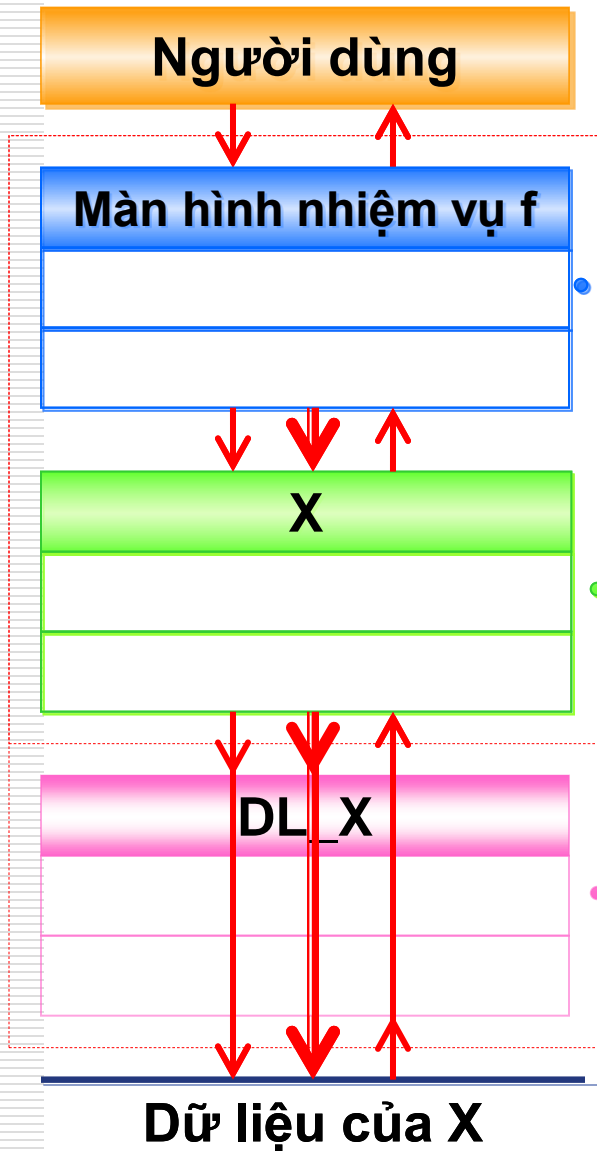
Kiến trúc 1 layer



Kiến trúc 2 layer



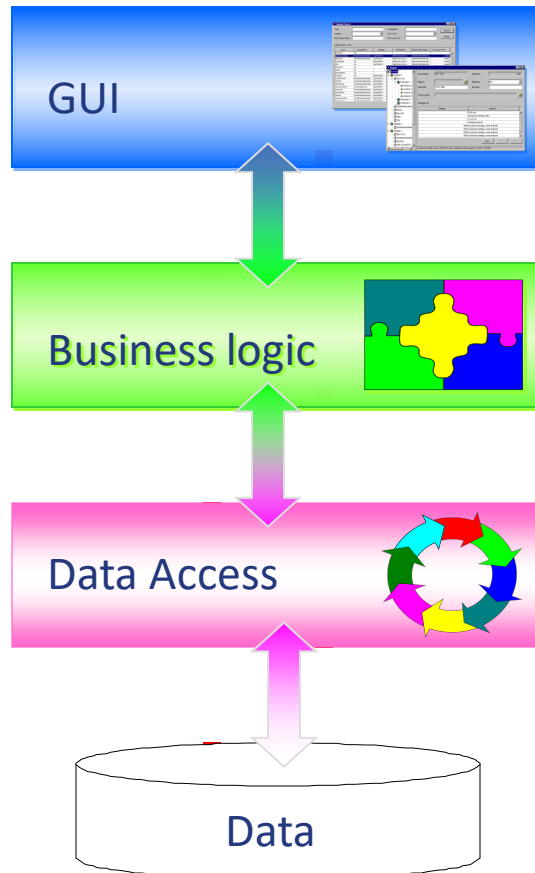
Kiến trúc 3 layer



Vai trò của các layer

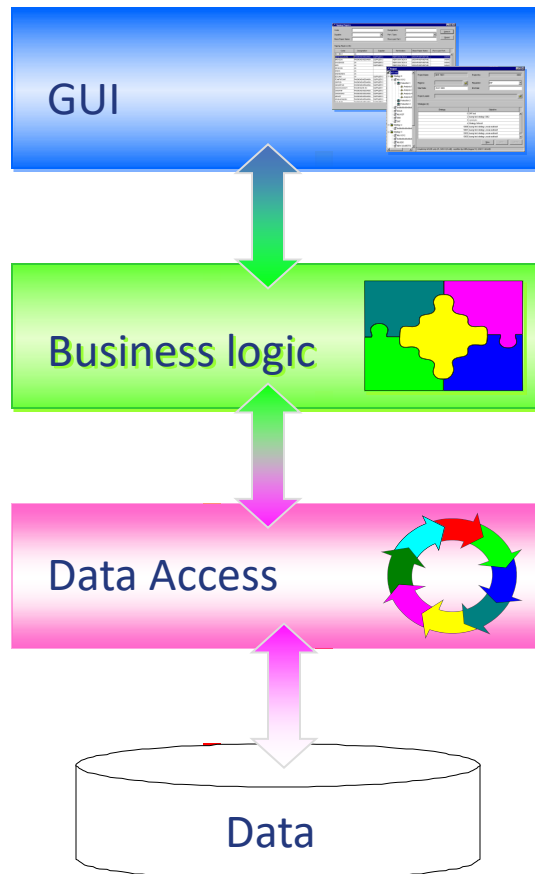
- ❖ GUI (Presentation) Layer: Nhập liệu và trình bày dữ liệu, có thể bao gồm các bước kiểm tra dữ liệu trước khi gọi Business Logic Layer.
- ❖ Business Logic Layer: Kiểm tra các yêu cầu nghiệp vụ trước khi cập nhật dữ liệu, quản lý các transaction, quản lý các concurrent access.
- ❖ Data Access Layer: Kết nối CSDL, tìm kiếm, thêm, xóa, sửa,...trên CSDL/XML

Việc trao đổi liên lạc giữa các layer



- Các giá trị, dòng, bảng

Việc trao đổi liên lạc giữa các layer

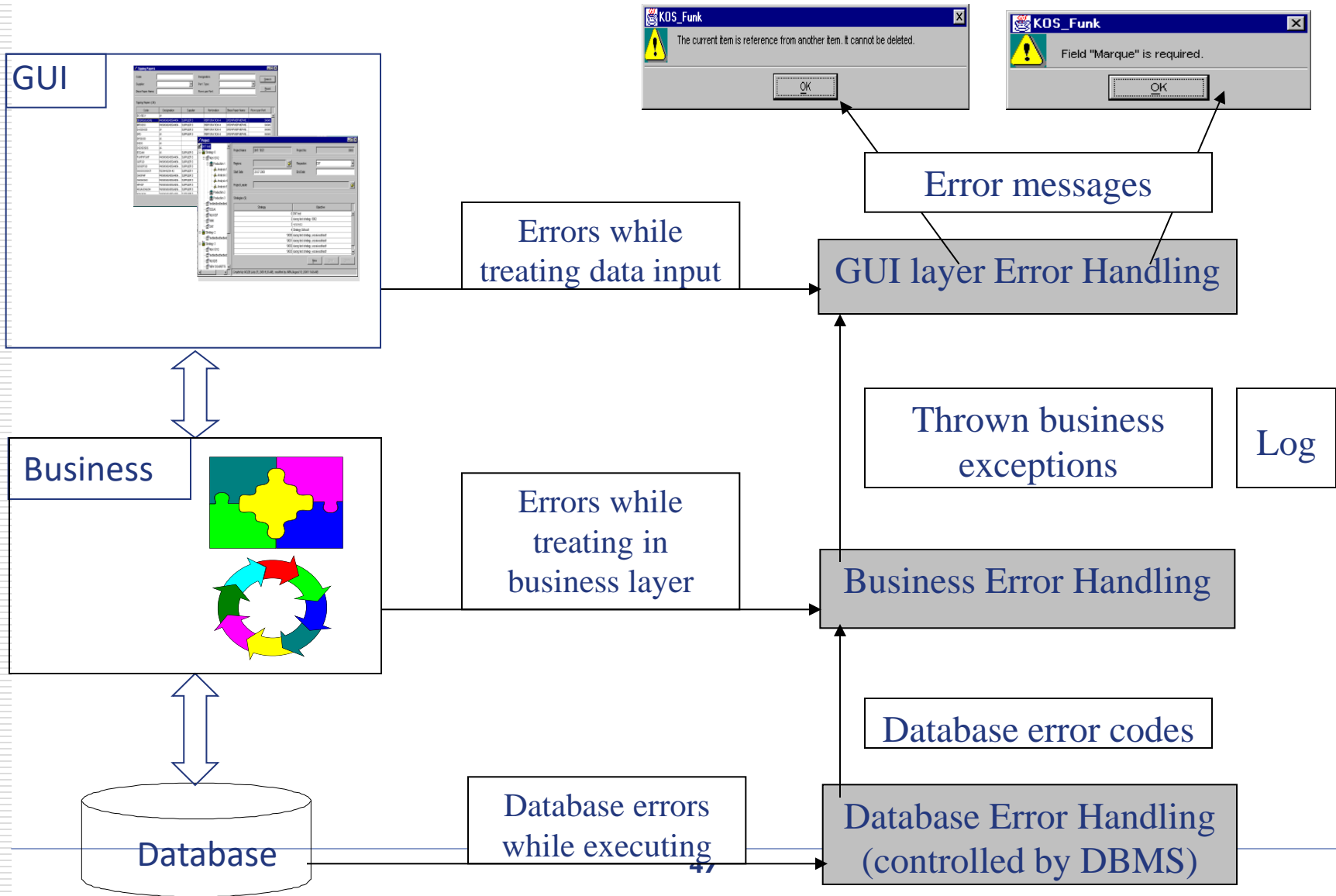


- Data Transfer Object (DTO)
- Các giá trị, dòng, bảng

Tính chất của mô hình 3-layer

- ❖ Giảm sự kết dính giữa các thực thể phần mềm (decoupling)
- ❖ Tái sử dụng
- ❖ Chia sẻ trách nhiệm

Quản lý ngoại lệ



Quản lý ngoại lệ

- ❖ Ngoại lệ có thể xảy ra ở bất kỳ layer nào
- ❖ Khi ngoại lệ xảy ra ở một layer thì:
 - Xử lý nội bộ trong layer đó
 - “Quăng” ngoại lệ lên layer “cao hơn”
 - Không xử lý
- ❖ Khi một layer nhận ngoại lệ từ một layer “thấp hơn”
 - Xử lý nội bộ
 - “Quăng” ngoại lệ lên layer “cao hơn”
 - Không xử lý

Nội dung

1. Kiến trúc hệ thống
2. Sơ đồ package
3. Sơ đồ component
4. Sơ đồ deployment
- 5. Một số mô hình kiến trúc phần mềm**
 - Layers
 - Tiers**

1-tier, 3-layer

Physical view

Application + Data

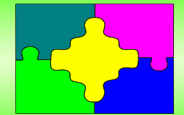


Logical view

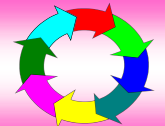
GUI



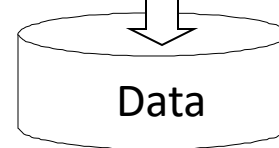
Business logic



Data Access

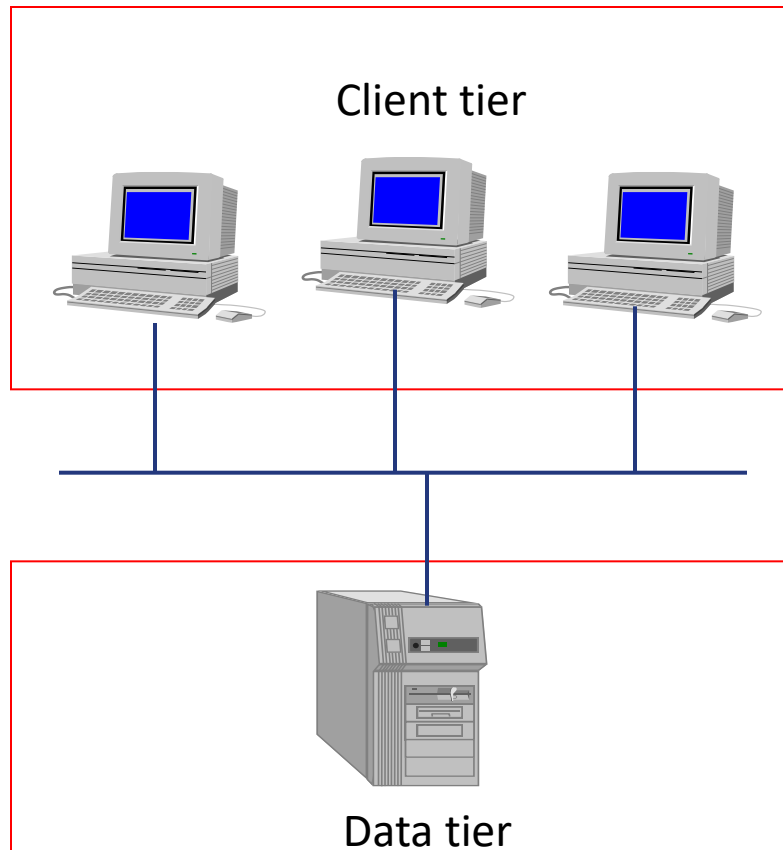


Data

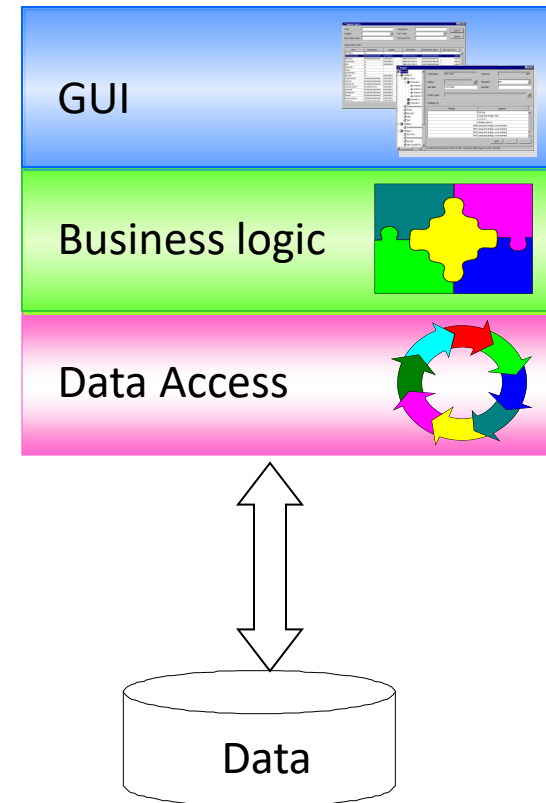


2-tier, 3-layer

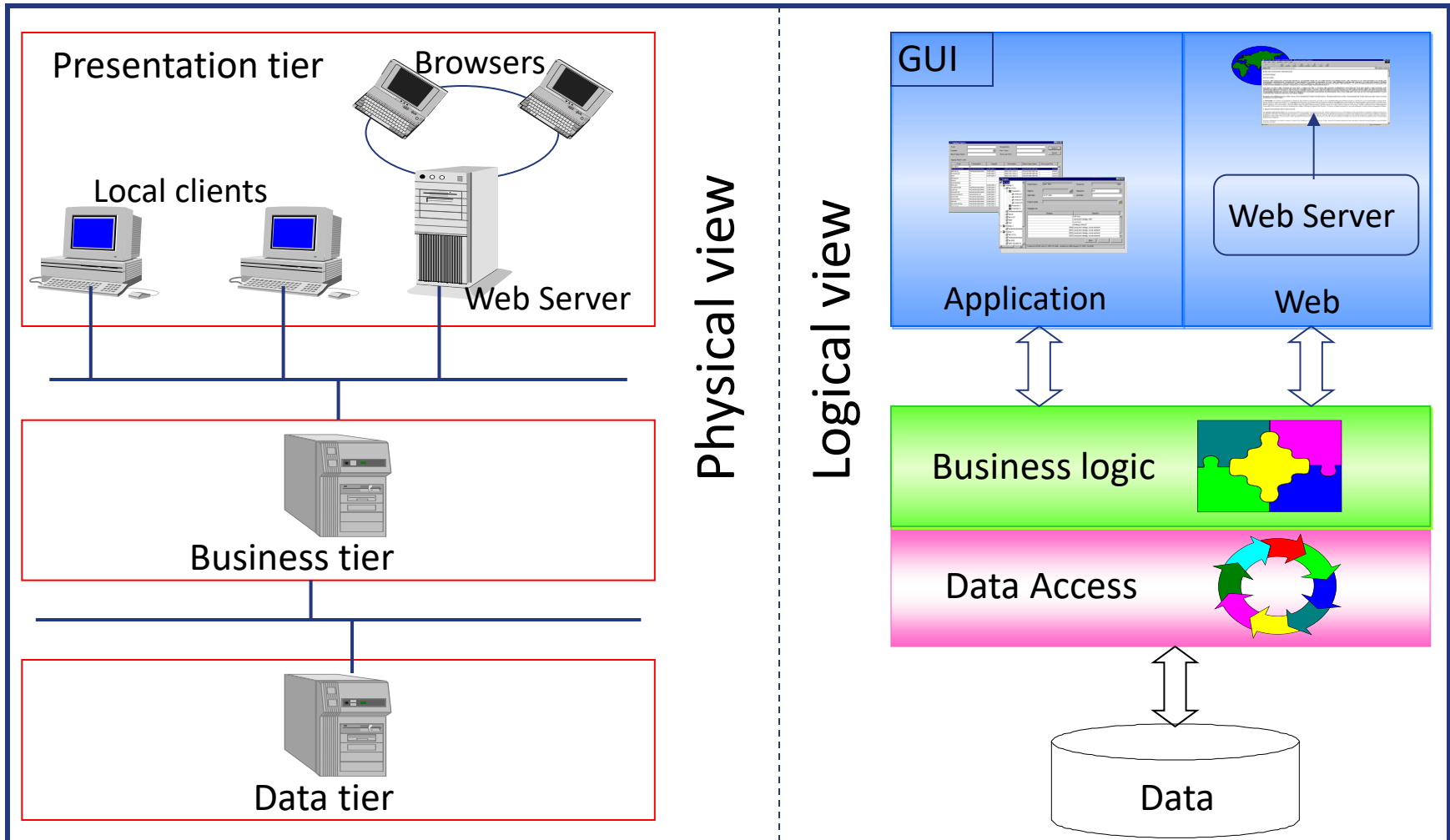
Physical view



Logical view



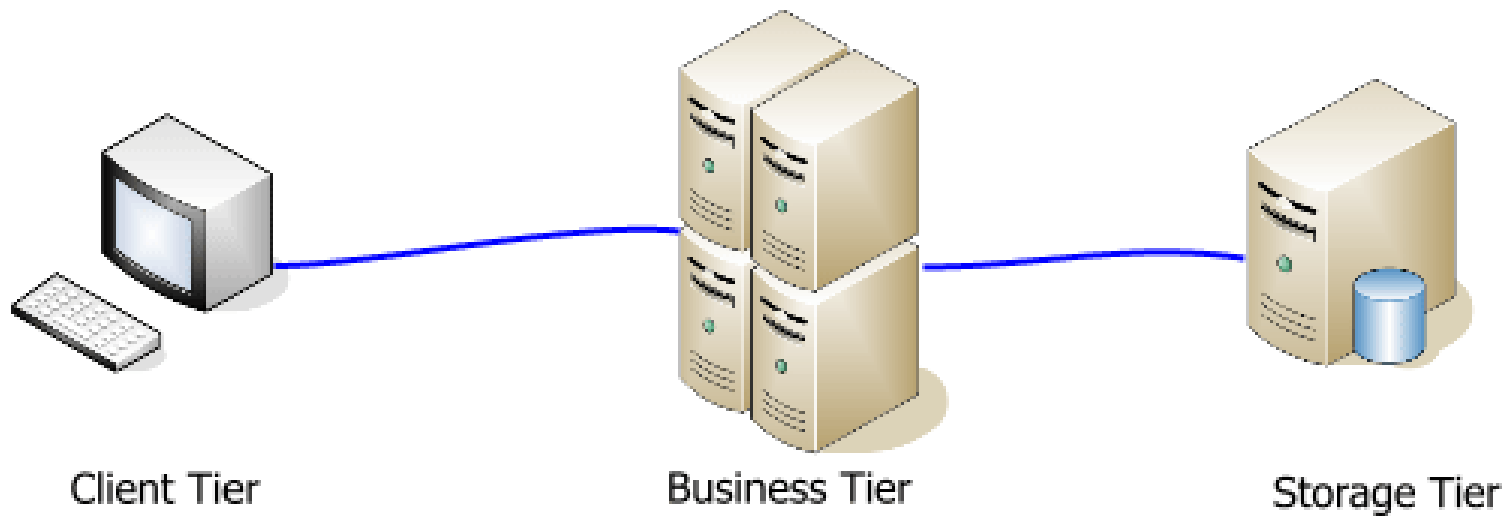
3-tier, 3-layer



Các lưu ý quan trọng

- ❖ Phân biệt vai trò Business Layer và khái niệm “xử lý”
- ❖ Mỗi Layer vẫn có xử lý riêng, đặc trưng của Layer đó
- ❖ Đôi khi việc quyết định 1 xử lý nằm ở layer nào chỉ mang tính chất tương đối

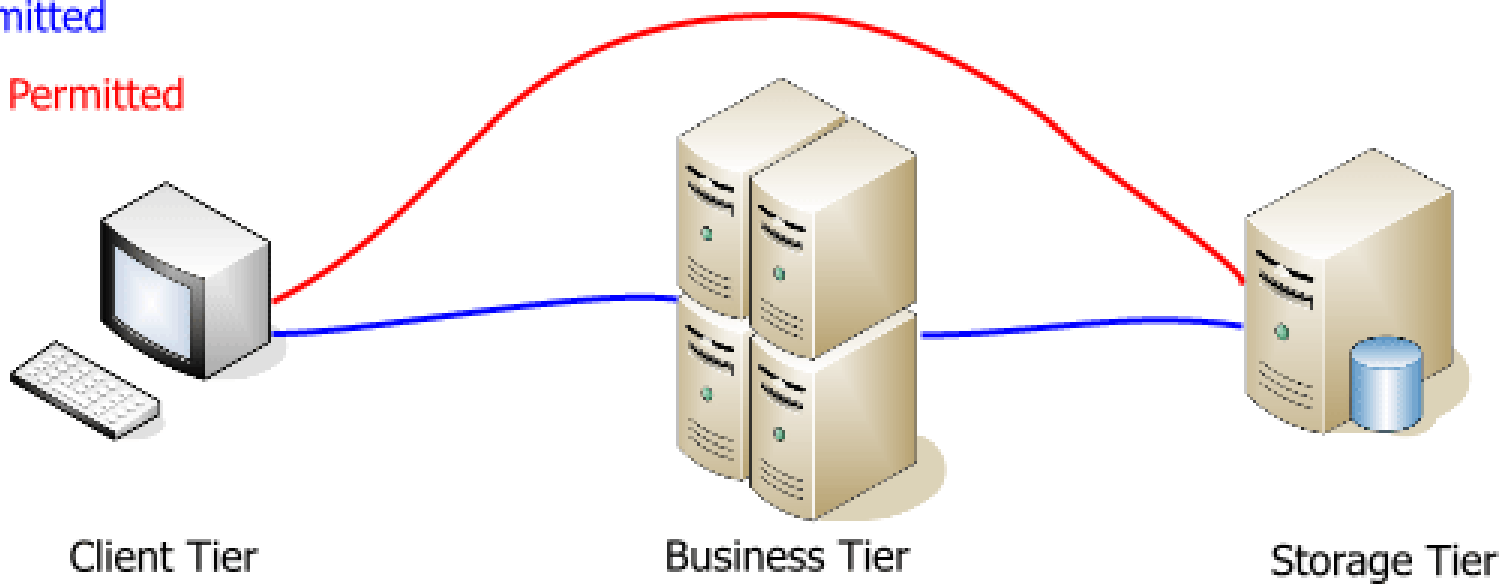
Physical tiers



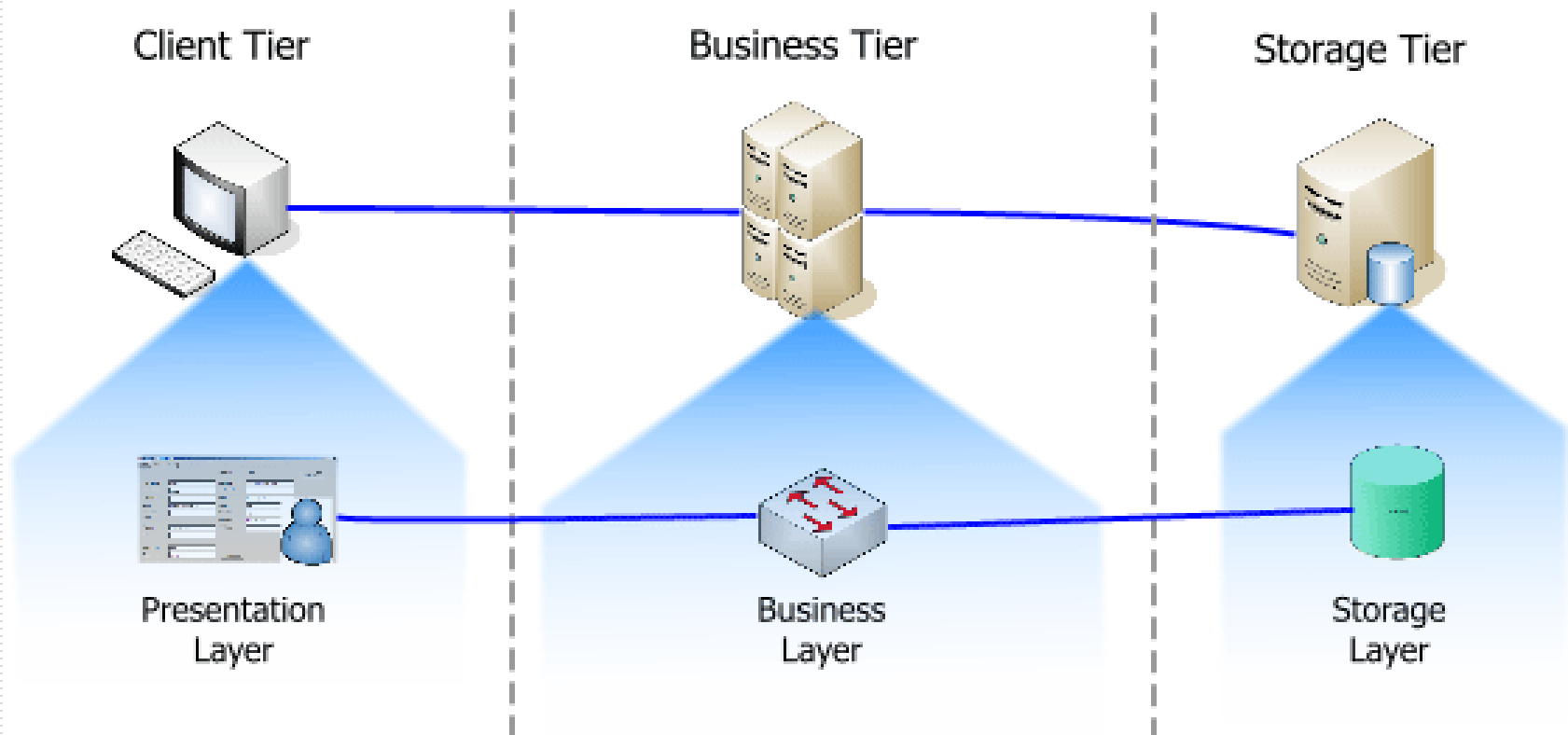
Physical tiers

Permitted

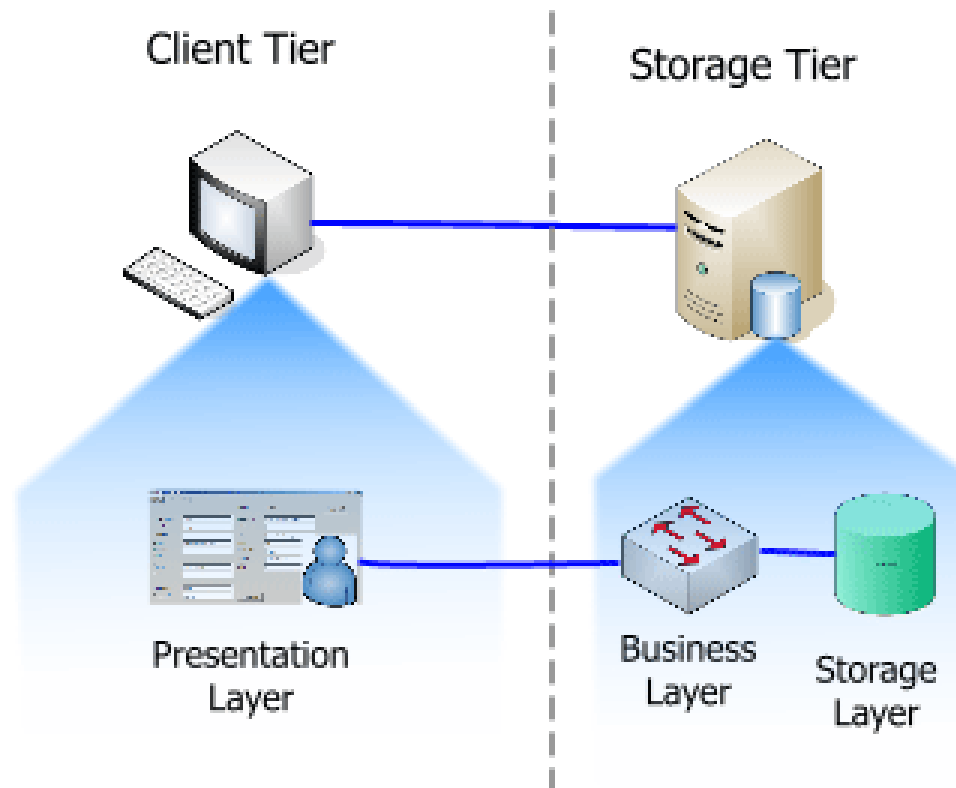
Not Permitted



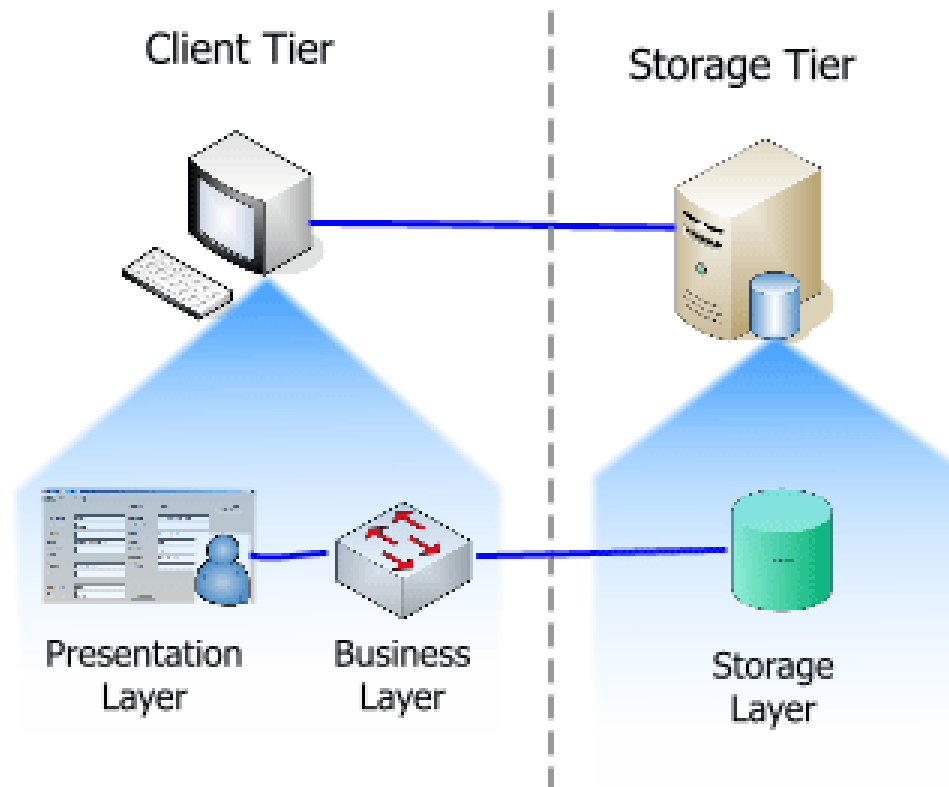
"Tiers of layers"



Business Layer đặt tại Storage Tier



Business Layer đặt tại Client Tier



Ví dụ về các thành phần tại mỗi Tier

