

Phân tích và thiết kế hệ thống

Giảng viên: Nguyễn Bá Ngọc

Hà Nội-2021

Kiến trúc hệ thống

Nội dung

- Mô-đun hóa hệ thống
- Các thành phần kiến trúc hệ thống
- Mẫu kiến trúc
- Các sơ đồ UML
 - Sơ đồ gói
 - Sơ đồ thành phần
 - Sơ đồ triển khai

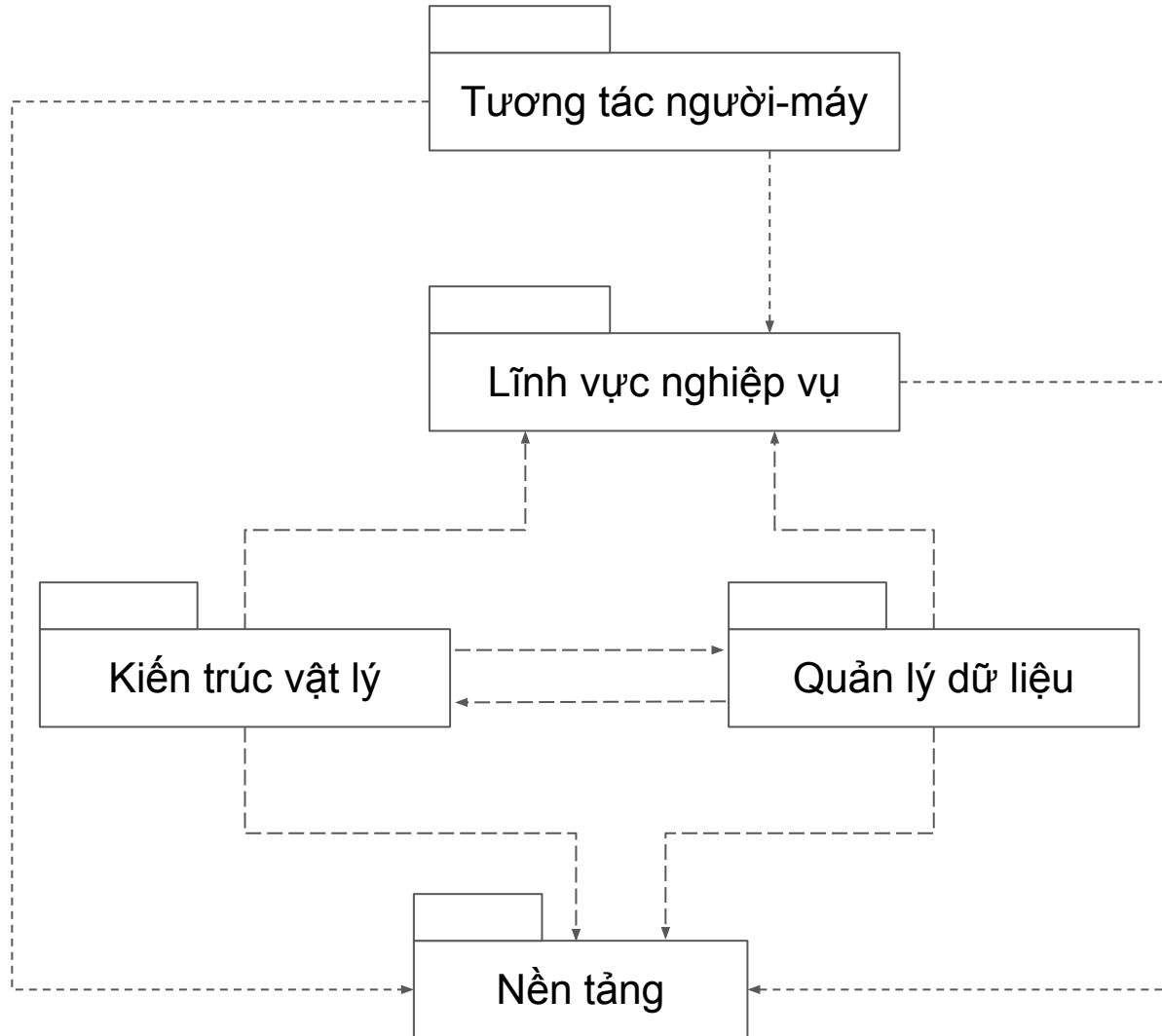
Nội dung

- Mô-đun hóa hệ thống
- Các thành phần kiến trúc hệ thống
- Mẫu kiến trúc
- Các sơ đồ UML

Mô-đun hóa hệ thống

- Mô-đun hóa: Chia nhỏ hệ thống thành nhiều mô-đun - các cấu phần hệ thống.
- Có thể vận dụng các nguyên lý thiết kế, và thử nghiệm với các kịch bản thay đổi.
- Có thể dựa trên các tương tác và các mối liên hệ:
 - Các sơ đồ lớp
 - Các sơ đồ giao tiếp
 - Ví dụ, các thông điệp và liên kết trong sơ đồ giao tiếp
 - Nếu có nhiều thông điệp được trao đổi giữa 2 lớp thì đưa cả 2 lớp vào cùng 1 mô-đun.
 - Ma trận CRUDE
 - V.V..
- Tạo mô-đun từ các phần tử có bậc liên kết cao.

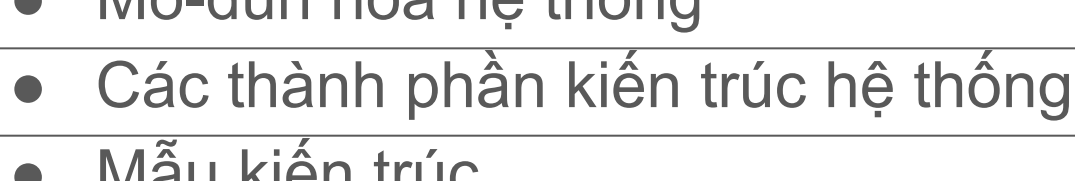
Ví dụ 1. Kiến trúc 5 tầng



Ví dụ 1. Kiến trúc 5 tầng₍₂₎

- **Tầng nền tảng:** Cung cấp các phần tử cơ bản để xây dựng hệ thống. Ví dụ: Array, Map, ...
- **Tầng nghiệp vụ:** Các biểu diễn và xử lý nhằm đáp ứng các hoạt động nghiệp vụ, đã được đề cập tới ở pha phân tích, và tiếp tục được phát triển ở pha thiết kế. Ví dụ lớp: Order, Customer
- **Quản lý và truy cập dữ liệu:** Các xử lý liên quan đến lưu trữ cố định. Ví dụ: CustomerDAM, FileInputStream
- **Tương tác người-máy:** Các giao diện người dùng, thường là các giao diện đồ họa. Ví dụ: Form, Button

Nội dung

- Mô-đun hóa hệ thống
 - Các thành phần kiến trúc hệ thống
 - Mẫu kiến trúc
 - Các sơ đồ UML
- 

Các thành phần kiến trúc

- Hạ tầng
 - Máy tính, mạng máy tính và hình trạng mạng, và phần mềm hệ thống
 - Hình thành hạ tầng để triển khai phần mềm và cung cấp dịch vụ.
- Phần mềm
 - Chương trình ứng dụng, dịch vụ Web, CSDL, v.v..., hỗ trợ tác nghiệp.
 - Được triển khai trên hạ tầng, cài đặt trên các thiết bị phần cứng và kết nối với nhau qua các giao thức và đường truyền mạng.

Mục đích thiết kế kiến trúc tổng quan là xác định cách bố trí các thành phần phần mềm trong các thiết bị phần cứng.

Thiết bị tính toán

- Máy chủ - Cung cấp các tài nguyên dùng chung cho người dùng và các máy tính khác.
 - Hoạt động liên tục, độ tin cậy cao.
- Máy khách / Máy tính cá nhân
 - Máy bàn, máy tính xách tay, máy tính bảng, điện thoại thông minh, v.v..
 - Người dùng sử dụng để tương tác với hệ thống.

Điện toán đám mây

- Mô hình cho thuê phần cứng
 - Máy chủ ở trên "Đám mây"
 - Máy khách là thiết bị của người dùng
- "Đám mây"
 - Trung tâm dữ liệu, nội bộ hoặc công khai; hoặc
 - Dịch vụ được cung cấp bởi các đối tác.
 - Tích hợp nhiều công nghệ:
 - Ảo hóa
 - Kiến trúc hướng dịch vụ
 - Mạng lưới tính toán / Grid computing
 - V.V...

Phần mềm như dịch vụ (SaaS)

- Thường được thực hiện với hạ tầng điện toán đám mây
- Phần mềm được triển khai trên máy chủ thay vì thiết bị của người dùng.
- Các dịch vụ ứng dụng được truy cập từ xa (thường qua trình duyệt).
- Dữ liệu người dùng được cô lập và được lưu trên máy chủ dịch vụ.

Ứng dụng Web

- Được triển khai dựa trên các quy chuẩn Web.
- Truy cập qua URL.
- Dữ liệu được trao đổi dựa trên giao thức mạng.
- Các xử lý được thực hiện trên máy chủ.
- Dữ liệu được trả về trong trang Web.
- V.V..

Hạ tầng mạng

- Bao gồm thiết bị mạng, phần mềm mạng.
- Hạ tầng Internet
 - Các đường truyền băng thông lớn và máy tính tốc độ cao
 - Được sở hữu bởi các chính phủ và các công ty viễn thông
- Mạng địa phương (LAN)
 - Mạng nhỏ cho 1 hệ thống thông tin nội bộ.
- Mạng toàn cầu (WWW)
 - Tất cả các tài nguyên được kết nối và truy cập qua Internet.

Các giao thức

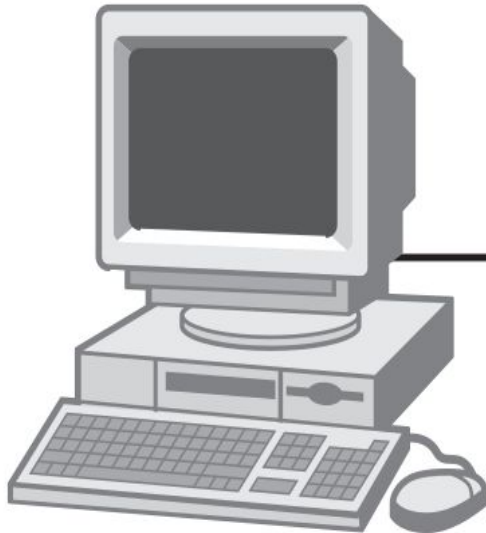
- Giao thức:
 - Một tập ngôn ngữ và quy tắc đảm bảo giao tiếp và trao đổi dữ liệu giữa phần cứng và phần mềm
- Các giao thức mạng:
 - Mạng riêng tư ảo (VPN)
 - Tạo mạng riêng tư trên Internet bằng cách sử dụng các công nghệ bảo mật và mã hóa

Các cấu phần hệ thống tiêu biểu

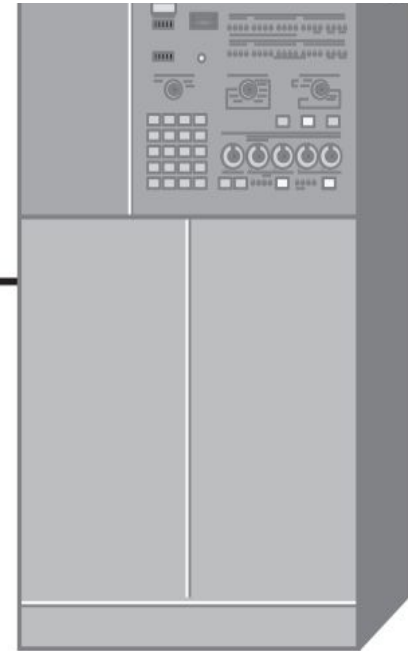
- Thiết bị phần cứng
 - Máy khách / máy tính cá nhân
 - Máy chủ
 - Hạ tầng mạng
- Thành phần phần mềm
 - Thành phần lưu trữ dữ liệu
 - Thành phần truy cập dữ liệu
 - Thành phần ứng dụng
 - Thành phần trình diễn

Kiến trúc dựa trên máy chủ

Máy khách/Cổng
thao tác



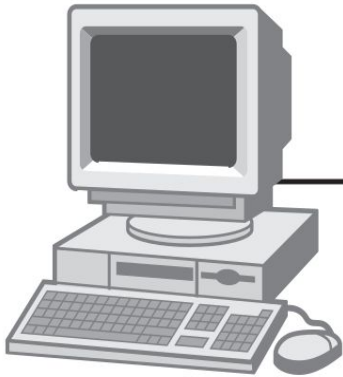
Máy chủ



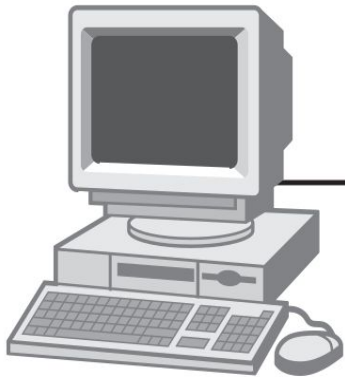
Trình diễn
Ứng dụng
Truy cập dữ liệu
Lưu trữ dữ liệu

Kiến trúc dựa trên máy khách

Các máy khách

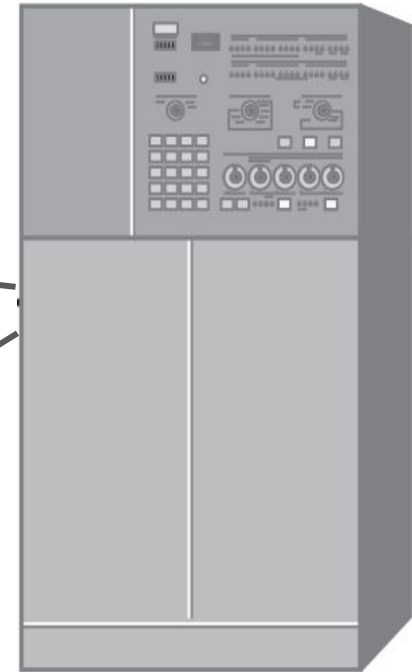


Trình diễn, ứng dụng,
truy cập dữ liệu



Trình diễn, ứng dụng,
truy cập dữ liệu

Máy chủ



Lưu trữ dữ liệu

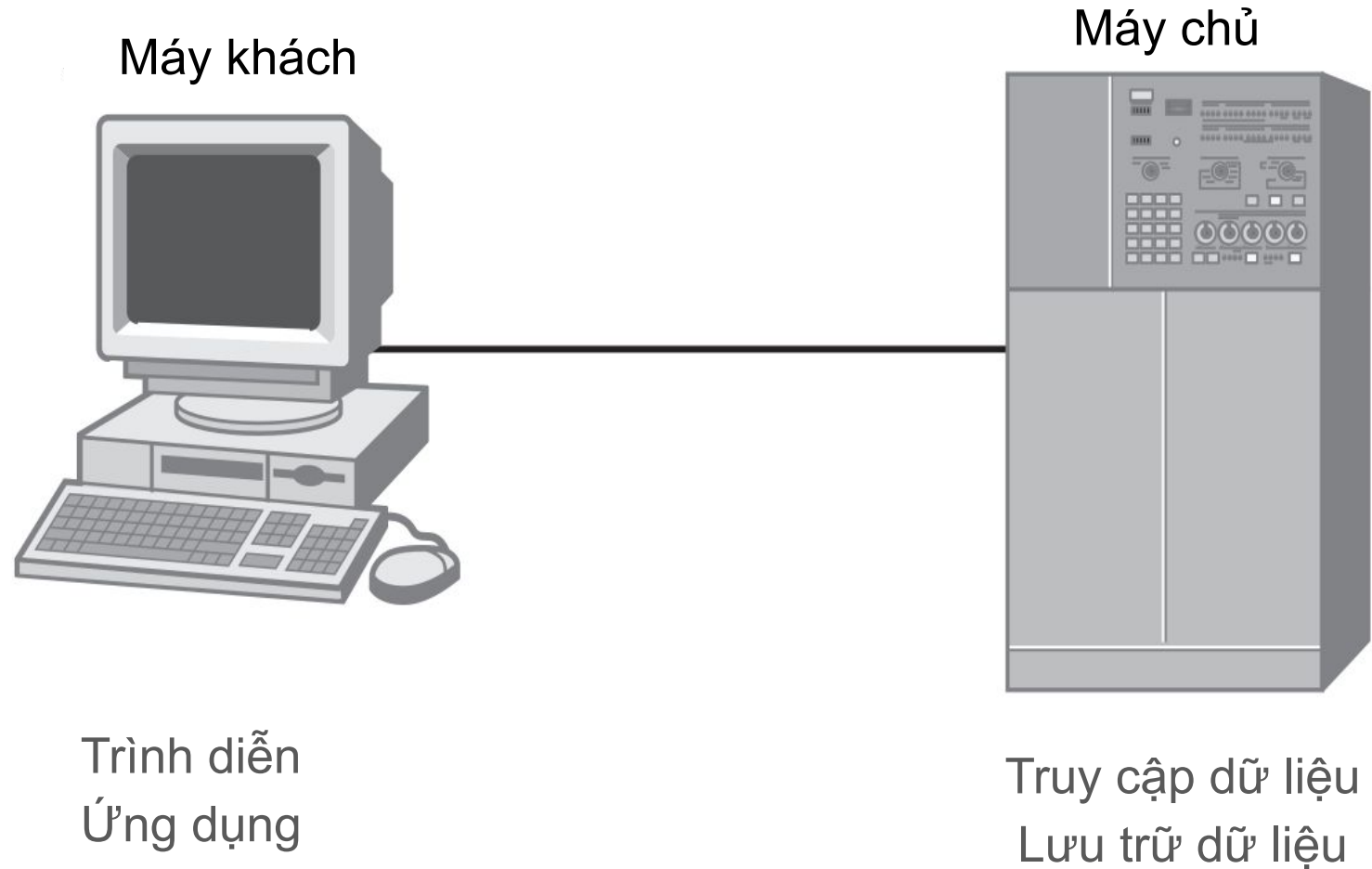
Kiến trúc máy khách-máy chủ

- Hệ thống được thiết kế với 1 phần ở máy chủ và 1 phần ở máy khách, cố gắng phân chia hợp lý khối lượng tính toán
- Kiến trúc phổ biến nhất trong các hệ thống hiện đại
- Khối lượng tính toán của máy khách dao động
 - Máy khách mỏng chỉ thực hiện lô-gic trình diễn
 - Máy khách dày thực hiện lô-gic trình diễn và lô-gic ứng dụng
- Khả năng mở rộng cao với chi phí riêng từng phần.
- Phức tạp hơn so với phát triển ứng dụng theo kiến trúc dựa trên máy khách và kiến trúc dựa trên máy chủ, và cần hỗ trợ tương tác trong môi trường phân tán.

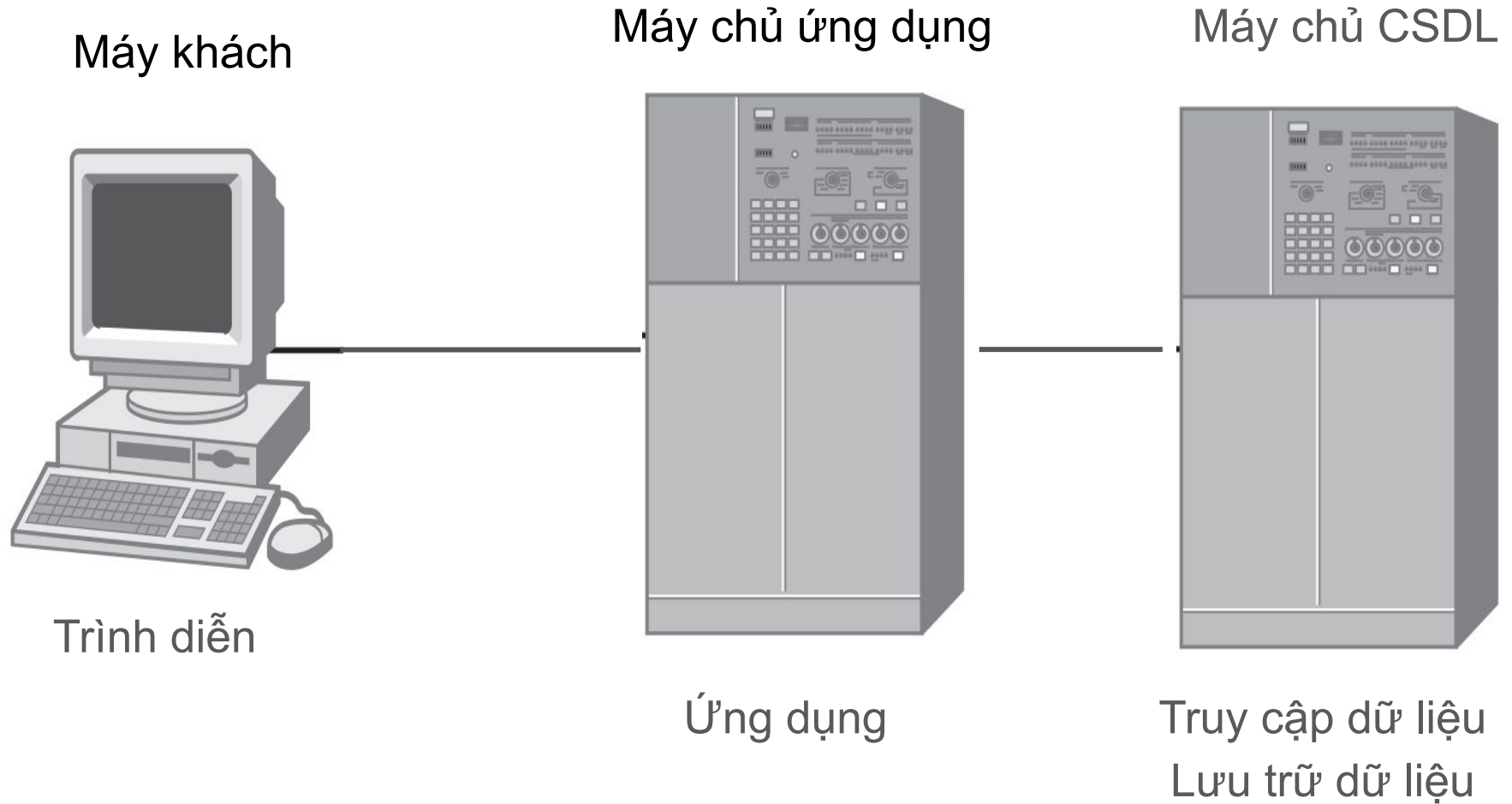
Các mô hình kiến trúc máy khách-máy chủ

- 2-dãy: 1 máy chủ chịu trách nhiệm lưu trữ và quản lý truy cập dữ liệu; máy khách có trách nhiệm xử lý lô-gic ứng dụng và lô-gic trình diễn
- 3-dãy: Lô-gic truy cập và lưu trữ dữ liệu trên 1 máy chủ; lô-gic ứng dụng trên 1 máy chủ khác; máy khách có trách nhiệm xử lý lô-gic trình diễn.
- n-dãy: Lô-gic ứng dụng được phân chia trên nhiều máy chủ, lô-gic dữ liệu trên 1 máy khác chủ khác
 - Phổ biến trong các ứng dụng thương mại điện tử
 - Cân bằng tải tốt hơn
 - Khả năng mở rộng cao hơn hệ thống 2 hoặc 3 dãy
 - Nhu cầu sử dụng mạng cao hơn
- Các dãy máy khách-máy chủ được xác định dựa trên lô-gic chia nhỏ hệ thống.

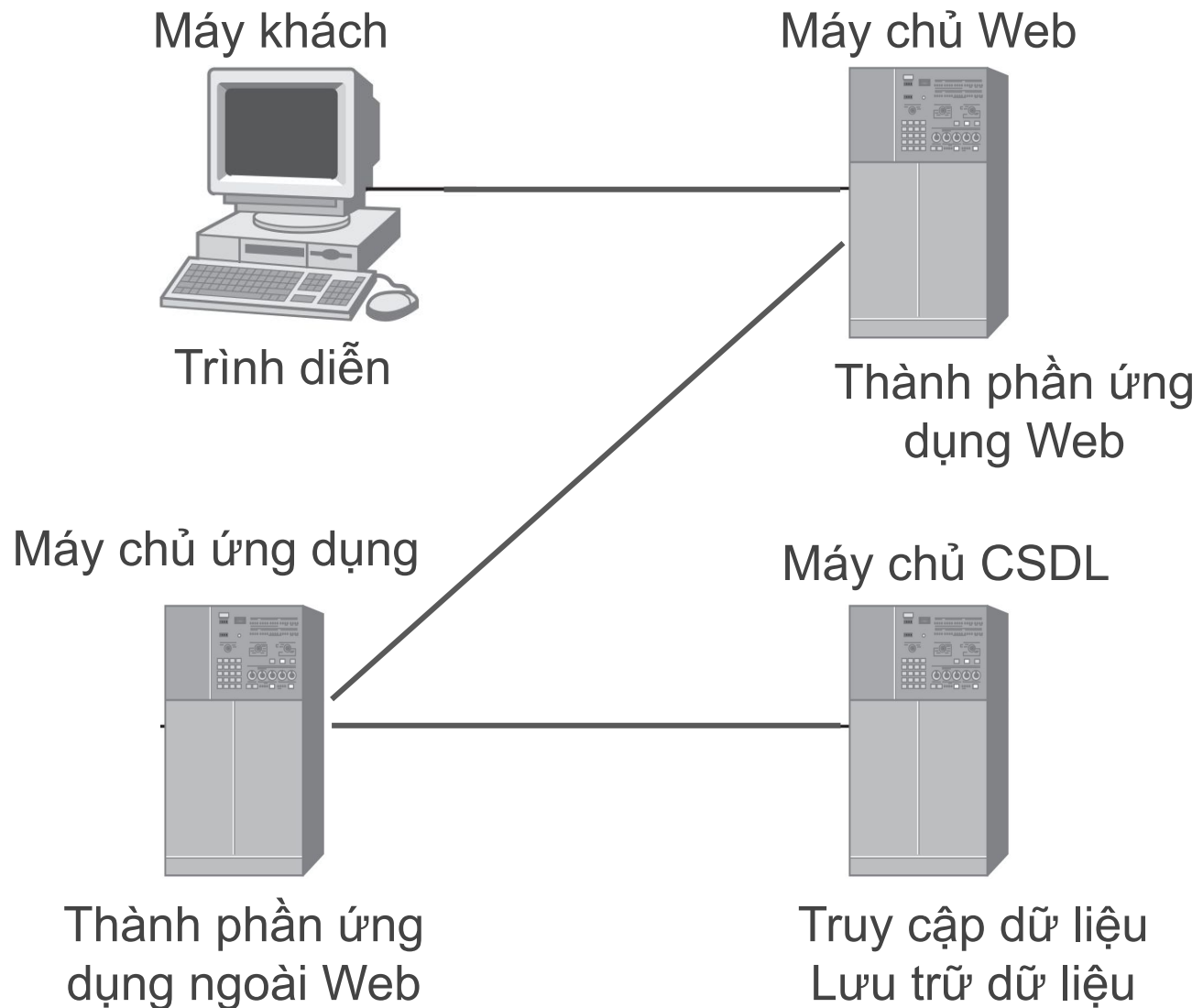
Kiến trúc máy khách-máy chủ 2-dãy



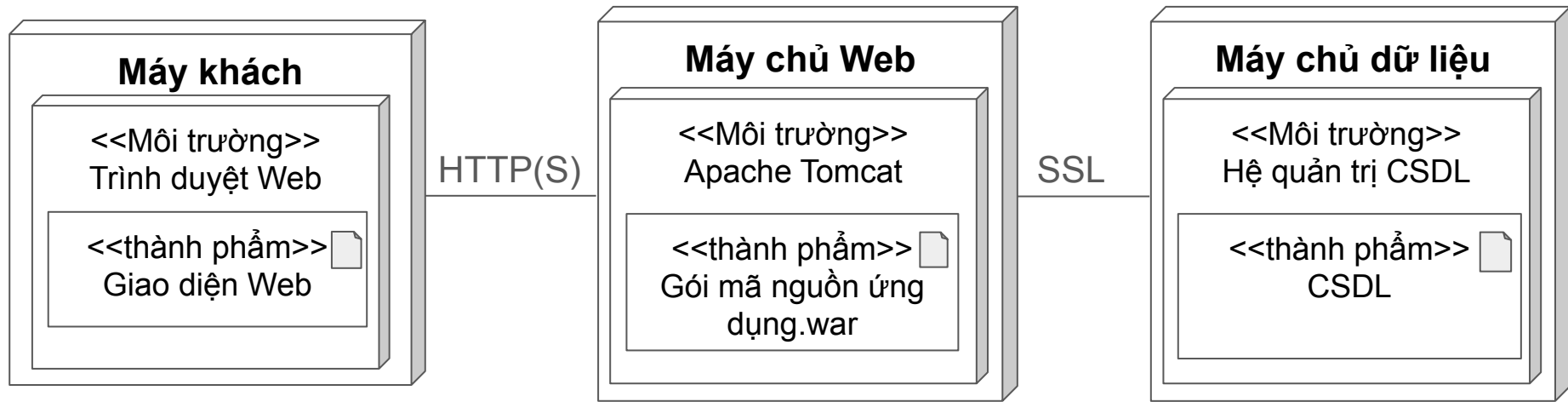
Kiến trúc máy khách-máy chủ 3-dây



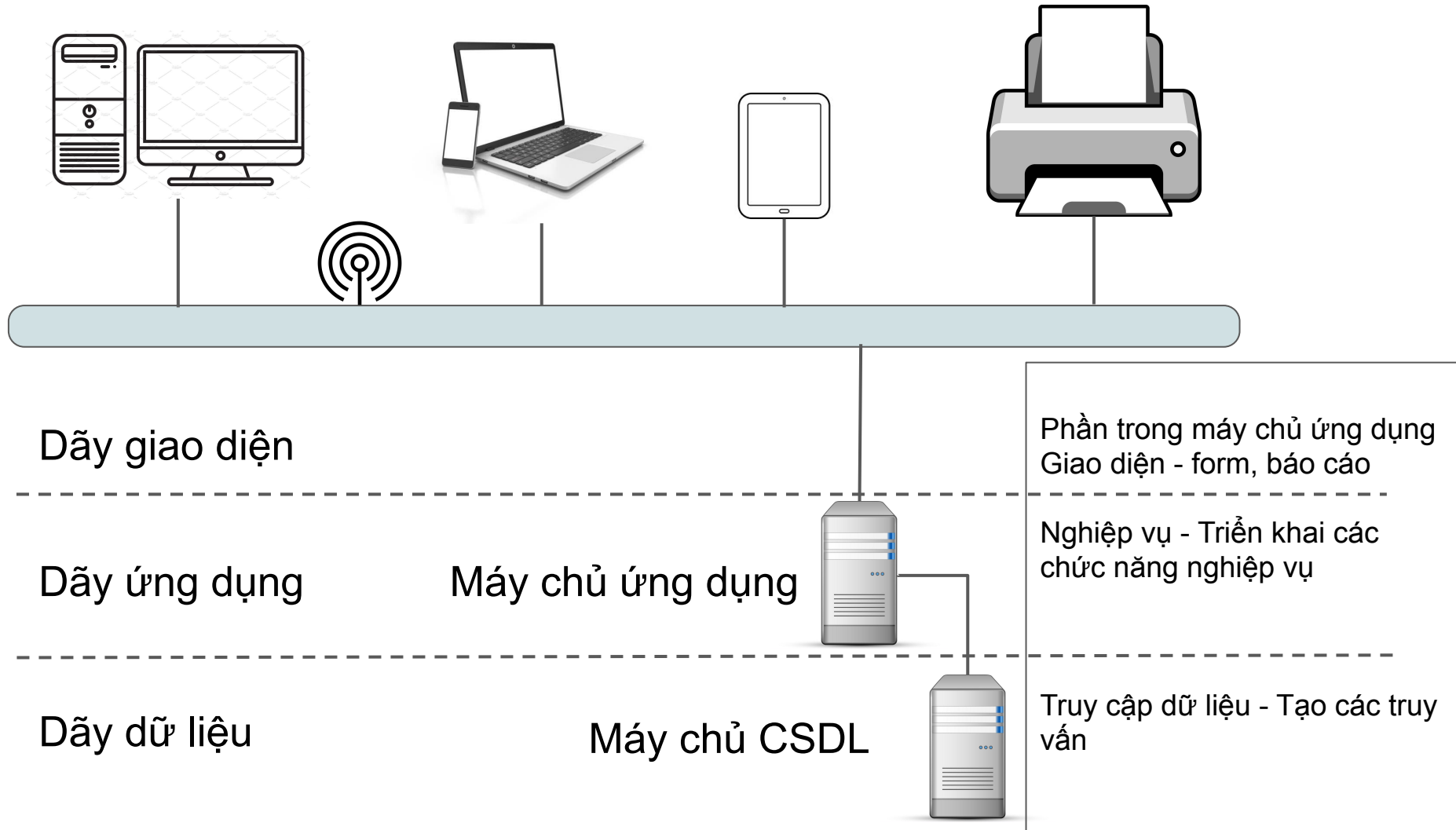
Kiến trúc máy khách-máy chủ 4-dãy



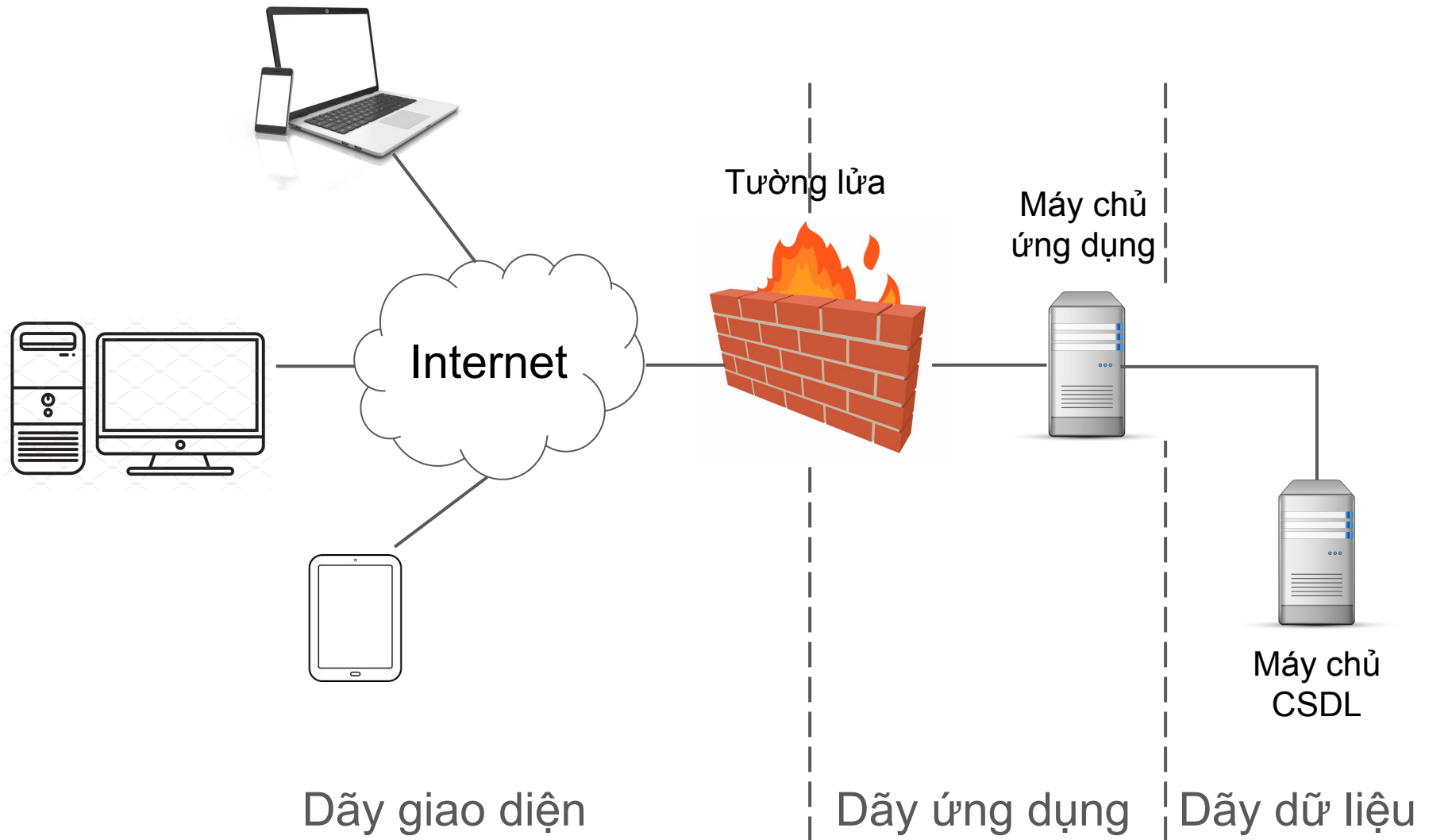
Sơ đồ triển khai hệ thống 3-dãy



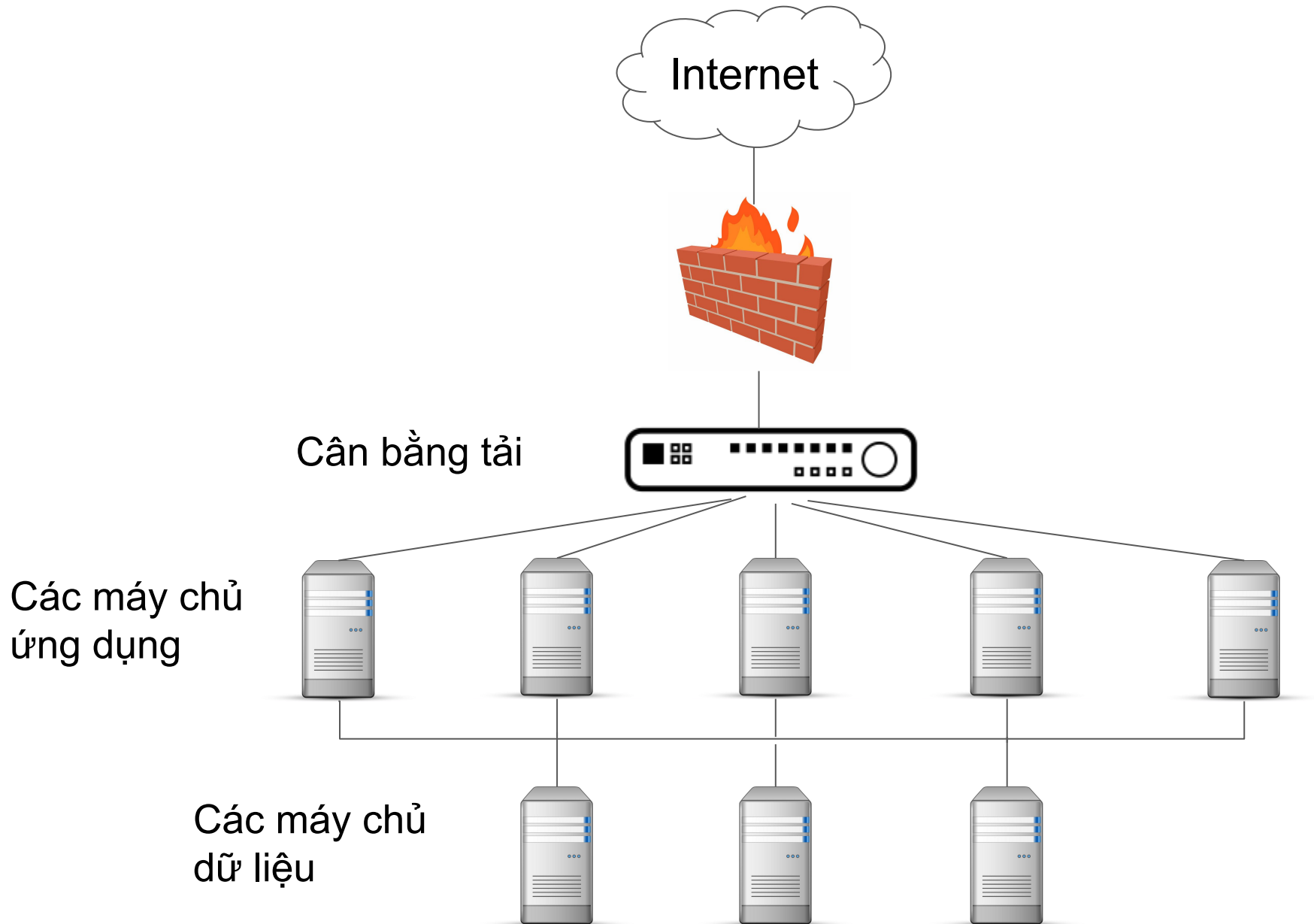
Kiến trúc 3-dãy: Triển khai nội bộ



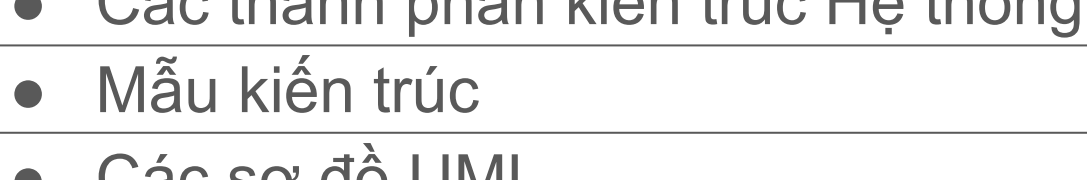
Kiến trúc 3-dãy: Triển khai trên internet



Kiến trúc 3-dãy: Mở rộng hệ thống



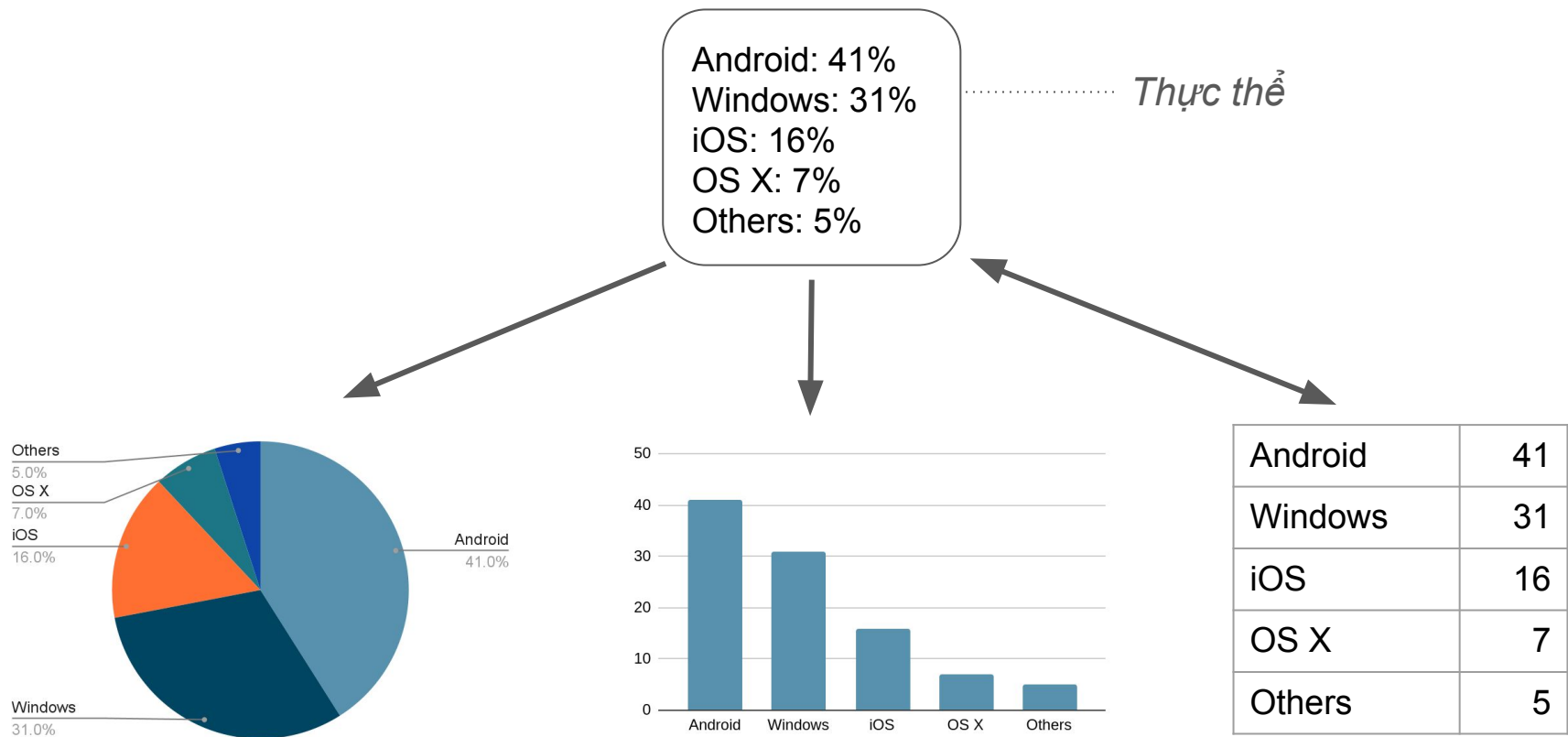
Nội dung

- Mô-đun hóa hệ thống
 - Các thành phần kiến trúc Hệ thống
 - Mẫu kiến trúc
 - Các sơ đồ UML
- 

Kiến trúc MVC

- Mẫu kiến trúc Model-View-Controller (MVC) phân chia ứng dụng phần mềm thành 3 phần:
 - Thực thể/Model biểu diễn dữ liệu và xử lý nghiệp vụ.
 - Khung nhìn/View giao diện đồ họa, tương tác người dùng.
 - Điều khiển/Controller tiếp nhận và điều hướng thông điệp.
- Điều khiển có thể được tích hợp trong giao diện.
- Các thay đổi trong thực thể có thể được thông báo và cập nhật theo các cơ chế:
 - Đẩy / Push: Khi thực thể thay đổi dữ liệu được đẩy lên khung nhìn.
 - Kéo / Pull: Khi thực thể thay đổi, khung nhìn được thông báo và nó tự yêu cầu các thành phần dữ liệu.

Ví dụ 2. Biểu diễn thị phần hệ điều hành

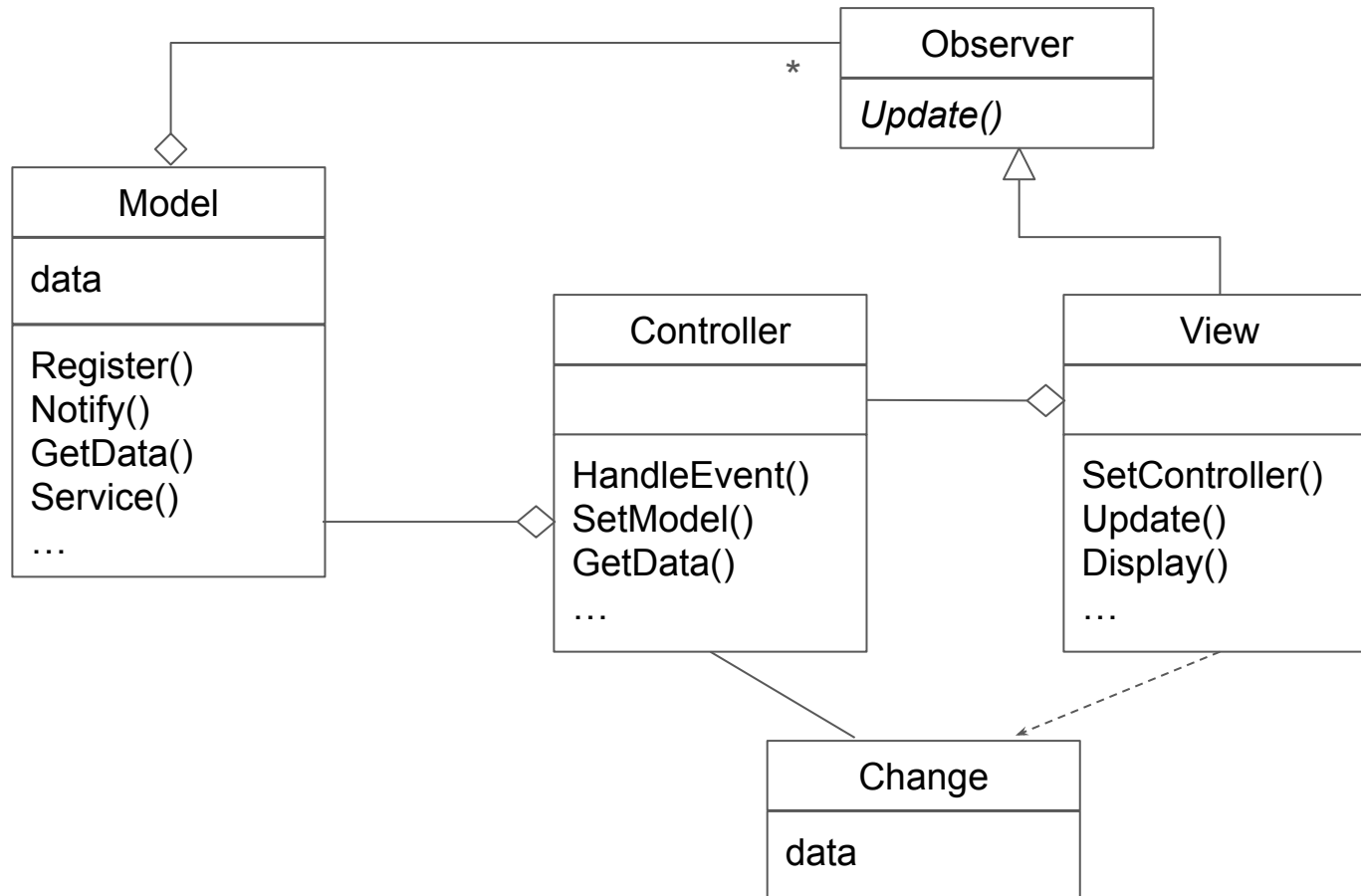


Tùy chỉnh giao diện đồ họa

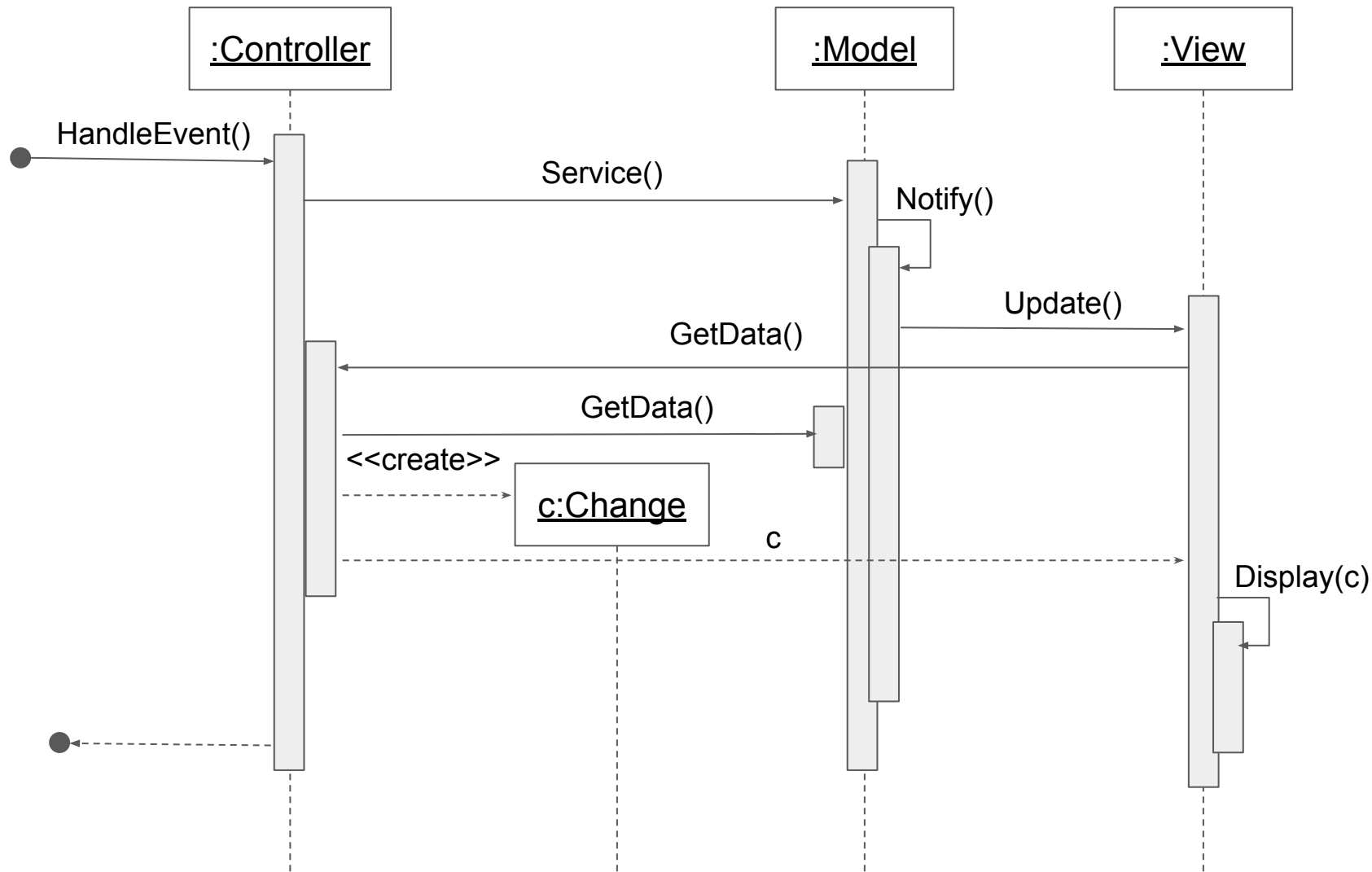
- Giao diện người dùng là phần hệ thống có nhiều nguyên nhân để thay đổi:
 - Mở rộng chức năng của hệ thống,
 - Các yêu cầu khác nhau của người dùng, v.v.
- Xây dựng phần mềm theo kiến trúc MVC cho phép giữ nguyên lô-gic ứng dụng (Thực thể) khi thay đổi giao diện, sử dụng đồng thời nhiều giao diện, và bổ xung giao diện mới.

Ví dụ 3. MVC theo cơ chế đẩy dữ liệu

Thực thể sẽ thông báo khung nhìn và điều khiển khi có thay đổi. Điều khiển sẽ tạo ra đối tượng dữ liệu cho khung nhìn.



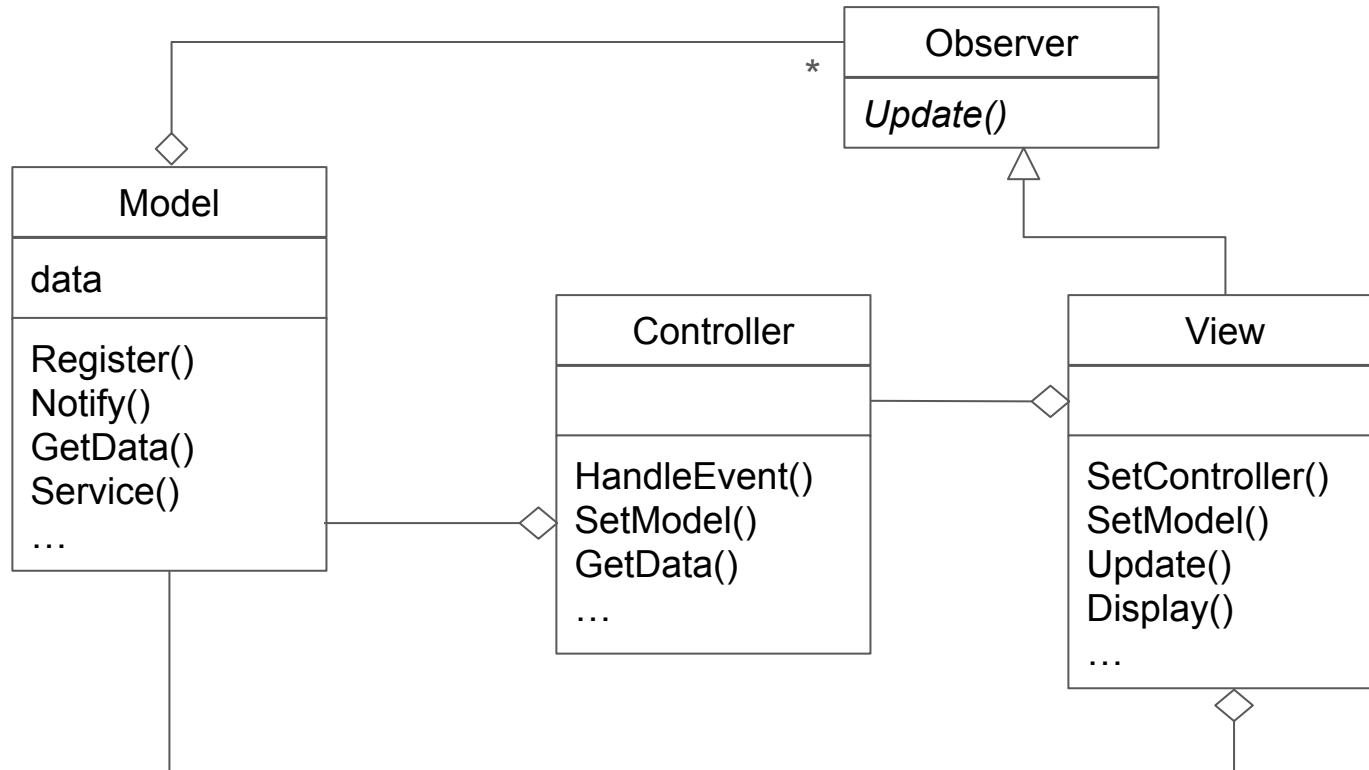
Ví dụ 3. MVC theo cơ chế đẩy dữ liệu⁽²⁾



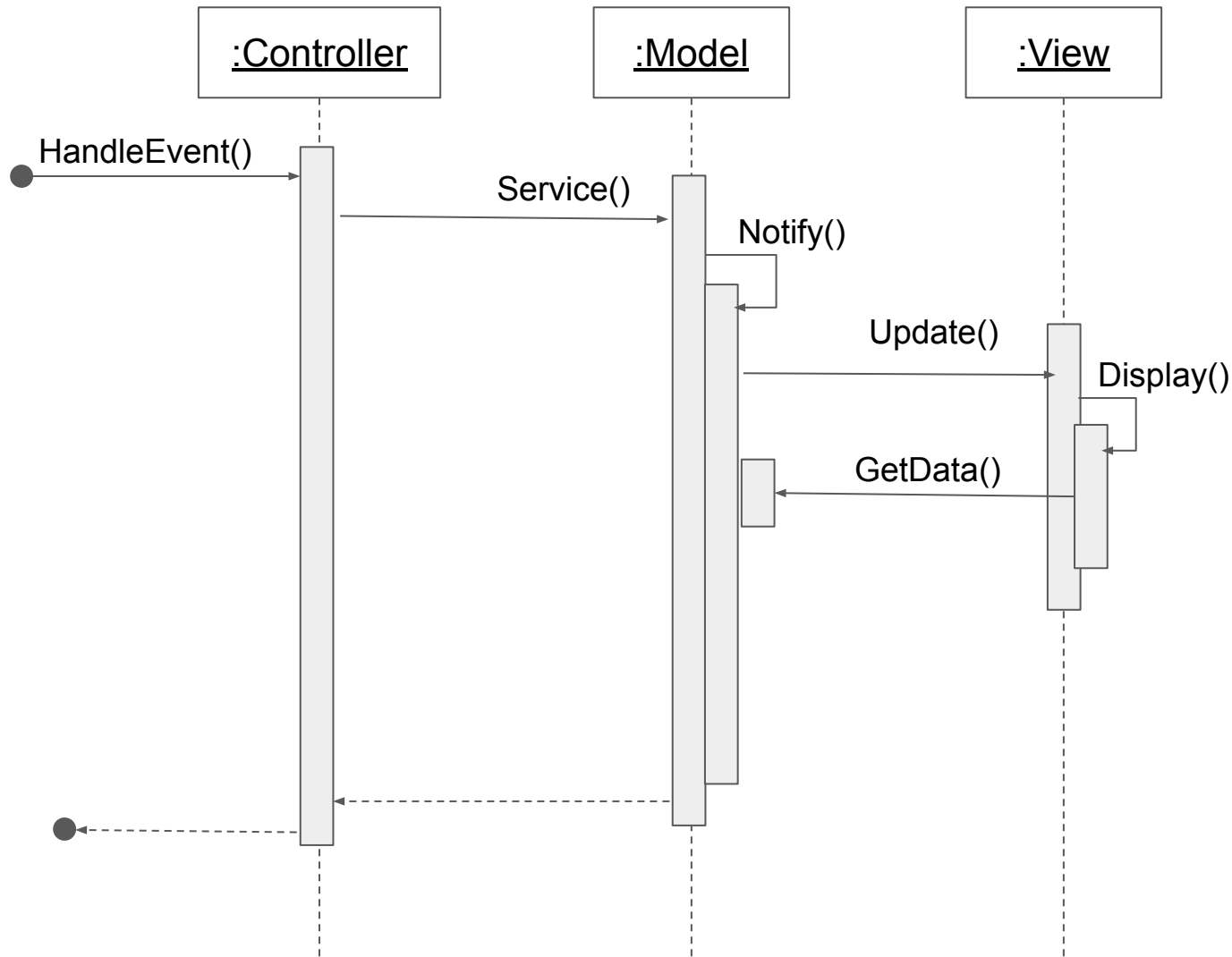
Khung nhìn lấy dữ liệu từ đối tượng được tạo bởi điều khiển

Ví dụ 4. MVC theo cơ chế kéo dữ liệu

Khung nhìn đọc dữ liệu từ thực thể khi có thay đổi



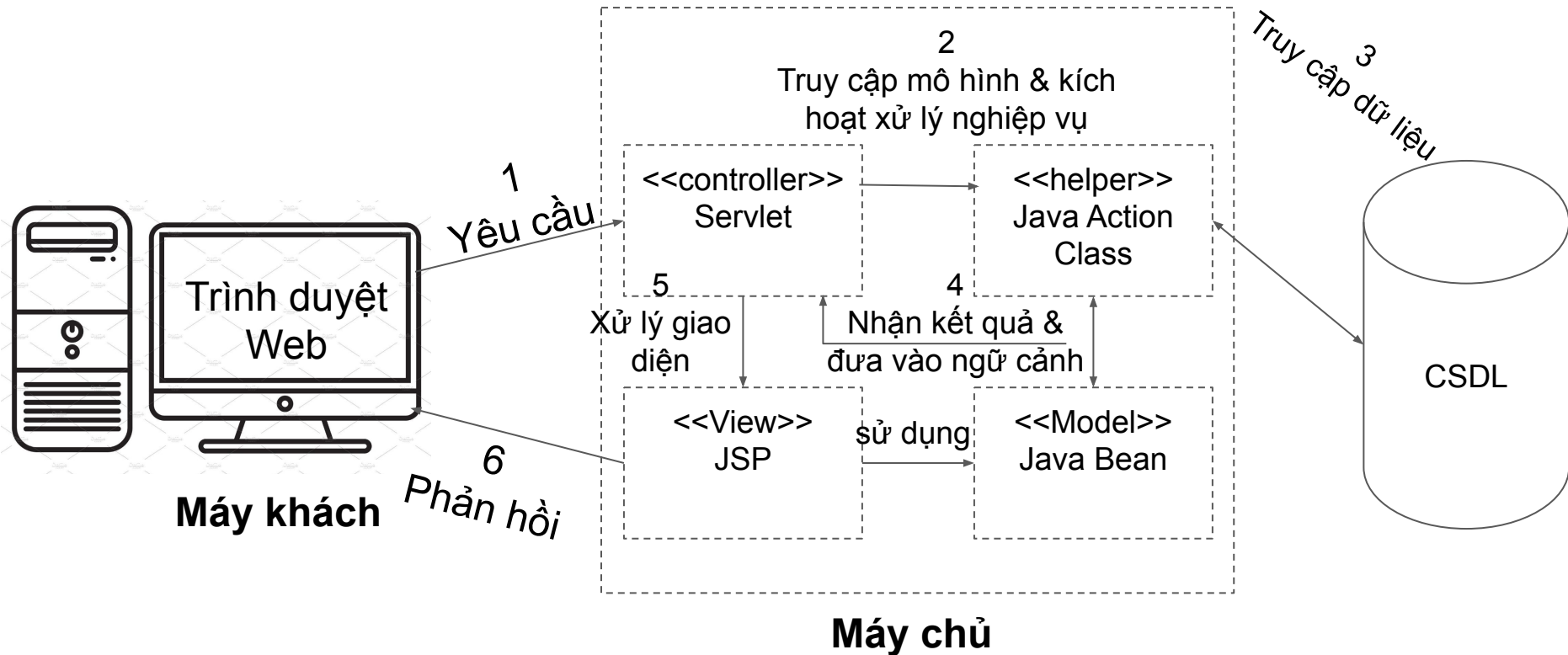
Ví dụ 4. MVC theo cơ chế kéo dữ liệu₍₂₎



Khung nhìn đọc dữ liệu từ thực thể khi cần cập nhật

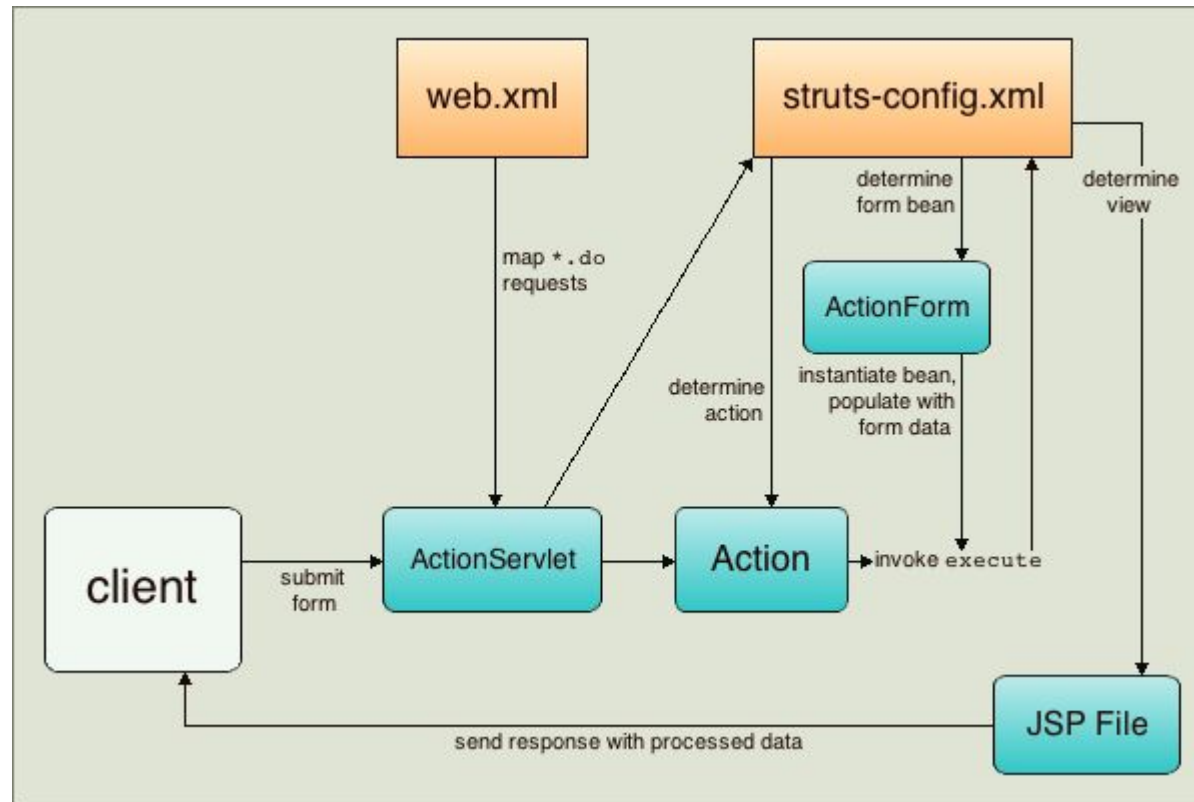
Kiến trúc JSP Model 2

Tách biệt lô-gic trình diễn và lô-gic ứng dụng tương tự như trong MVC.



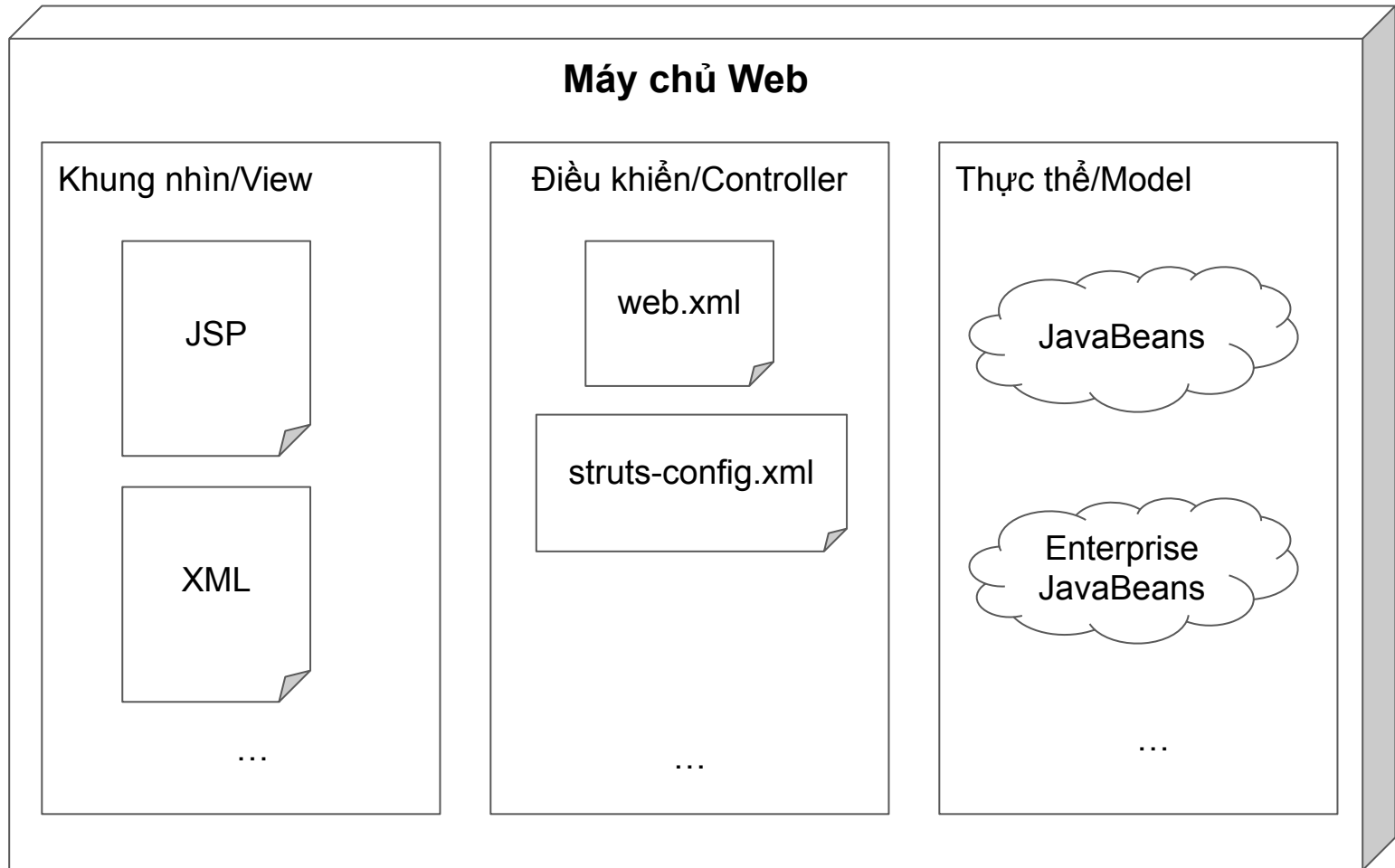
** JSP Model 2 không định nghĩa định dạng cụ thể của Model*

Nền tảng Struts

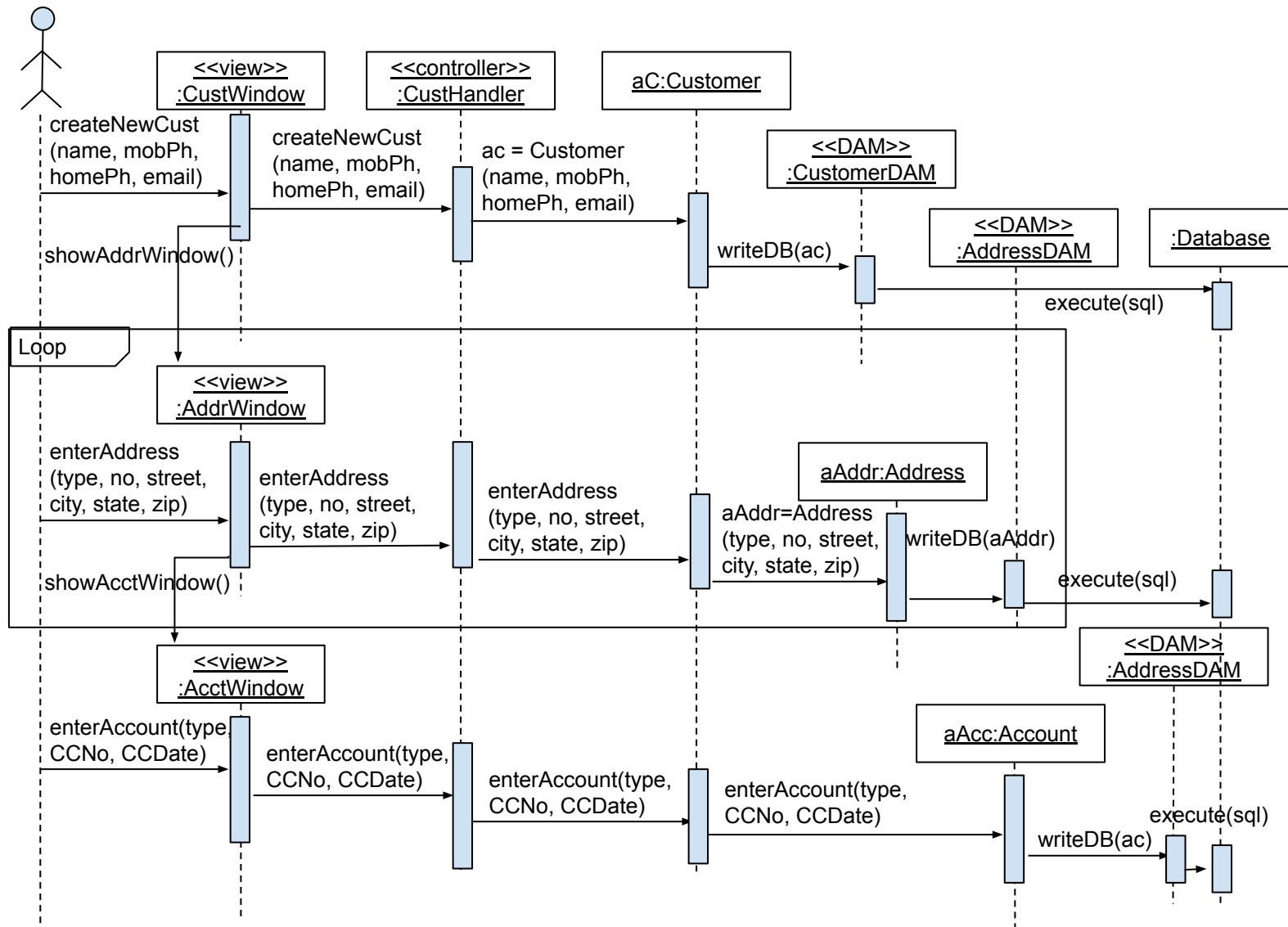


[apache.org]

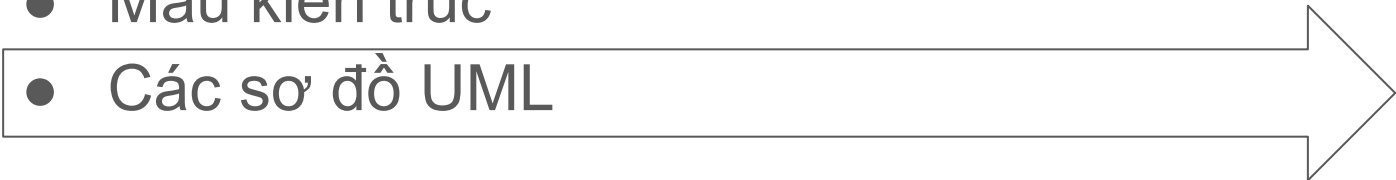
JSP Model 2 và MVC



Sơ đồ tương tác thực tế



Nội dung

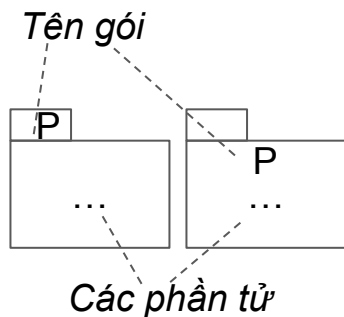
- Mô-đun hóa hệ thống
 - Các thành phần kiến trúc Hệ thống
 - Mẫu kiến trúc
 - Các sơ đồ UML
- 

Nội dung

- Mô-đun hóa hệ thống
- Các thành phần kiến trúc Hệ thống
- Mẫu kiến trúc
- Các sơ đồ UML
 - Sơ đồ gói
 - Sơ đồ thành phần
 - Sơ đồ triển khai

Sơ đồ gói: Hệ ký hiệu

Sơ đồ gói biểu diễn các gói cùng với các mối quan hệ

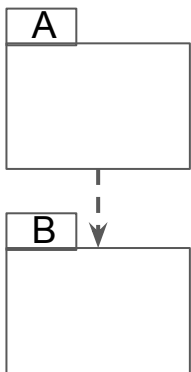


- **Gói/Package:**

- Nhóm lô-gic của nhiều phần tử, là không gian tên của các phần tử.
- Có thể gom nhiều phần tử liên quan thành 1 mô-đun bậc cao, giúp giản lược mô hình.

- **Quan hệ phụ thuộc/Dependency:**

- Nếu B thay đổi thì A cũng sẽ thay đổi theo.
- A phụ thuộc vào B được biểu diễn bằng mũi tên nét đứt từ A tới B.



Sơ đồ gói: Hệ ký hiệu₍₂₎

<<merge>>
----->

- Quan hệ hợp nhất/Merge:
 - Nội dung của gói đích (theo mũi tên) được hợp nhất với nội dung của gói nguồn.
 - Các phần tử cùng tên cũng được hợp nhất.

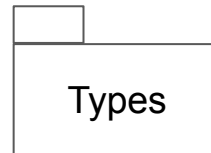
<<import>>
----->

- Quan hệ nhập/Import
 - Thêm các phần tử của gói đích vào gói nguồn.
 - Các phần tử được thêm vào có thể được nhìn thấy từ bên ngoài / nhập công khai.

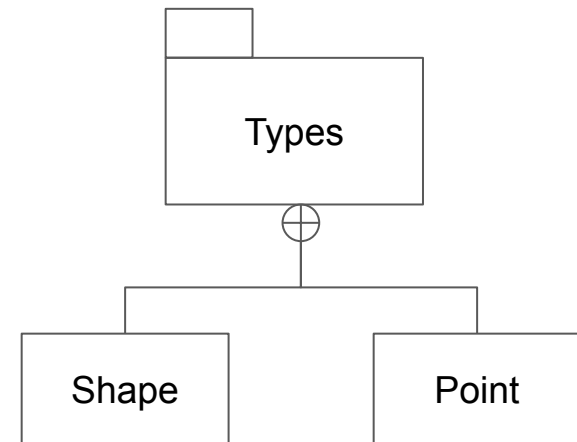
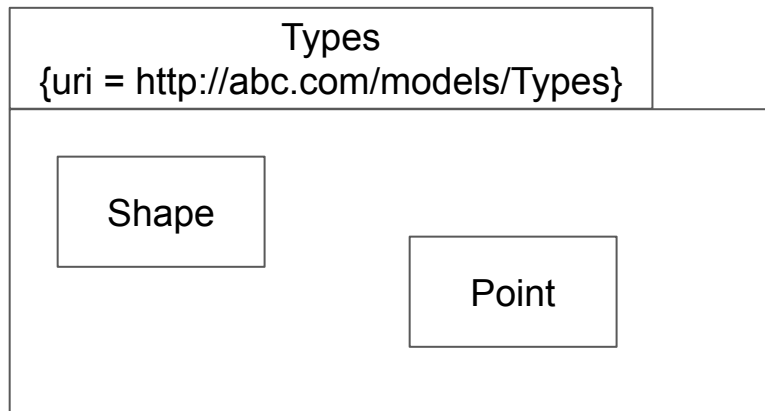
<<access>>
----->

- Quan hệ truy cập/access
 - Thêm các phần tử của gói đích vào gói nguồn
 - Các phần tử được thêm vào không được nhìn thấy từ bên ngoài / nhập riêng tư

Biểu diễn phần tử thuộc gói

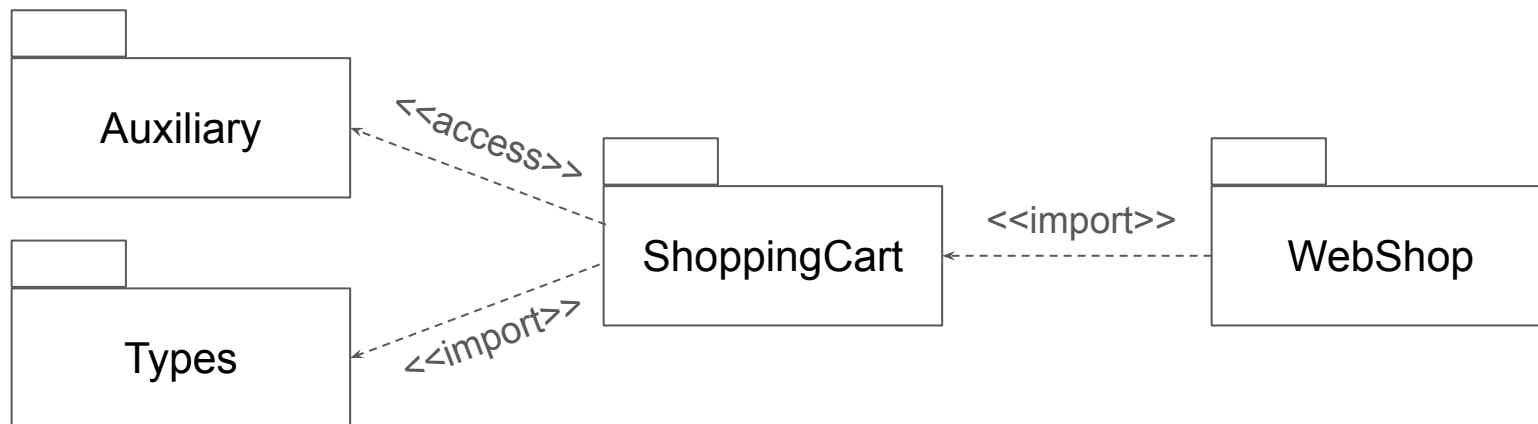


Ảnh nội dung của
gói Types



Các lựa chọn biểu diễn Shape và Point thuộc gói Types

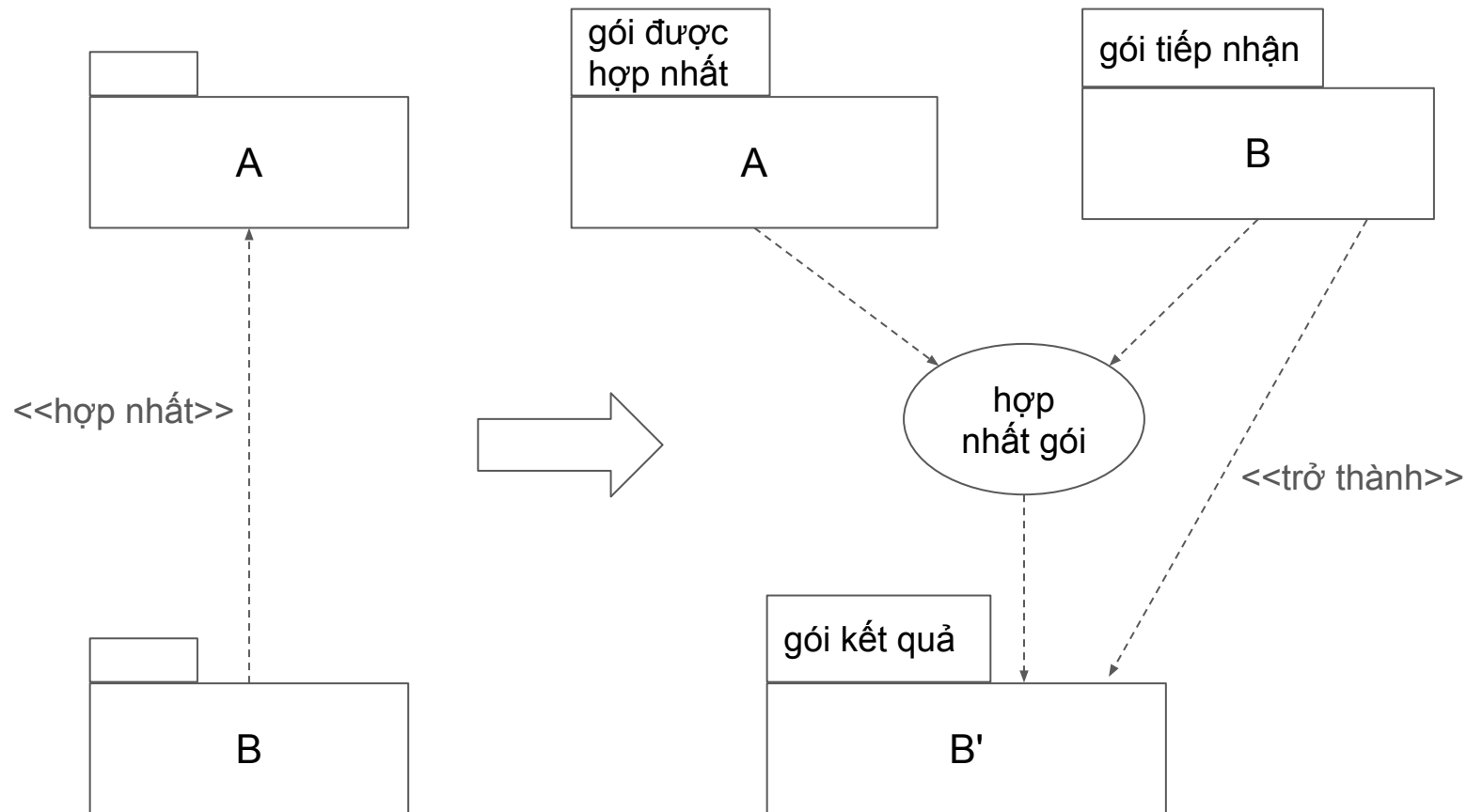
Quan hệ nhập - <<import>>



Các phần tử của gói Types được nhập vào gói ShoppingCart và sau đó được nhập vào WebShop.

Các phần tử của Auxiliary chỉ có thể truy cập được từ ShoppingCart, nhưng không thể được từ WebShop.

Quan hệ hợp nhất - <<merge>>



Hợp nhất các phần tử của A vào trong B.

Nội dung

- Mô-đun hóa hệ thống
- Các thành phần kiến trúc Hệ thống
- Mẫu kiến trúc
- Các sơ đồ UML
 - Sơ đồ gói
 - Sơ đồ thành phần
 - Sơ đồ triển khai

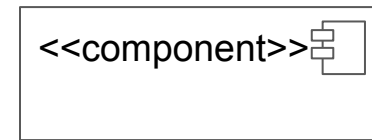
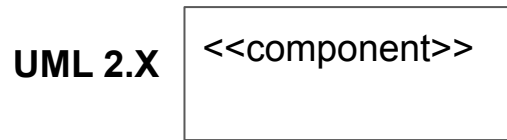
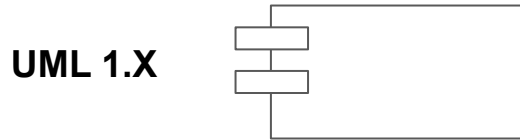
Sơ đồ thành phần

Sơ đồ thành phần biểu diễn các thành phần khác nhau của hệ thống cùng với các giao diện và các mối quan hệ.

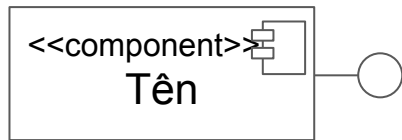
- Thành phần/component biểu diễn một mô-đun hệ thống, đóng gói đầy đủ nội dung của nó.
- Hành vi của thành phần được xác định bằng các giao diện được cung cấp và được yêu cầu.
- Mỗi thành phần trong môi trường của nó có thể được thay thế bằng thành phần khác với các giao diện tương thích.
- Mỗi thành phần có thể được thể hiện như một hoặc nhiều thành phẩm.

Sơ đồ thành phần: Hệ ký hiệu

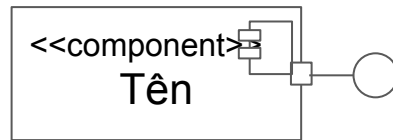
- Biểu diễn thành phần:



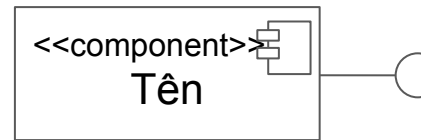
- Các giao diện được biểu diễn bằng hình tròn (cây kẹo mứt - cung cấp) và nửa đường tròn (ổ cắm - yêu cầu).



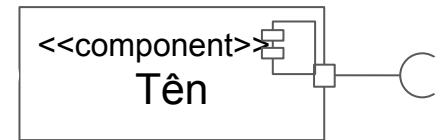
Triển khai giao diện



Cung cấp cổng

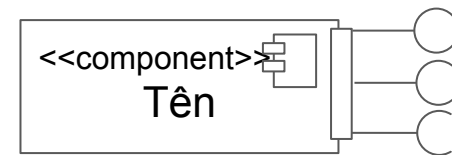


Sử dụng giao diện



Yêu cầu cổng

Giao diện có thể được gắn với cổng



Thành phần với cổng phức tạp

- Các quan hệ:

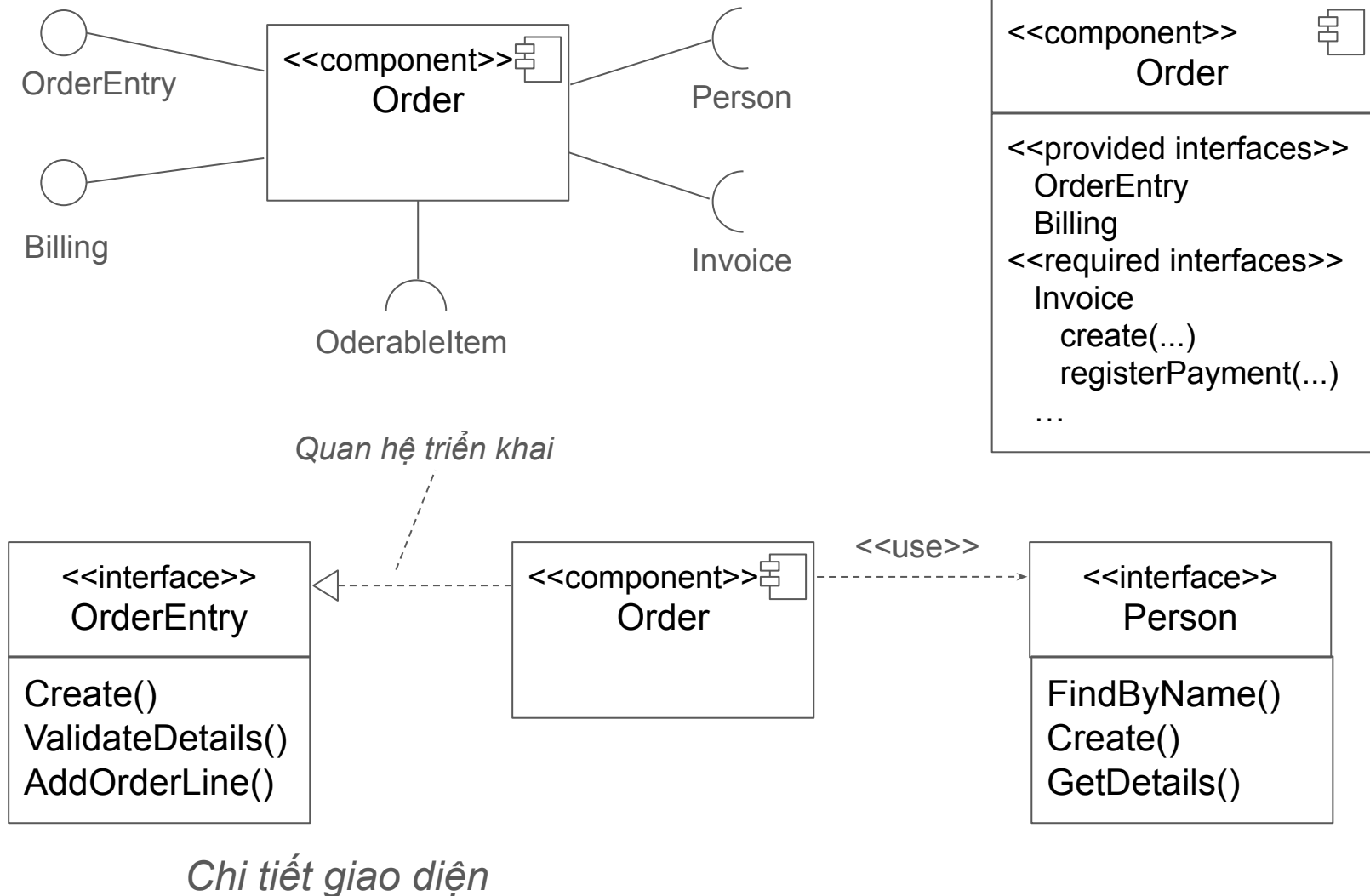
-----> Phụ thuộc/ Sử dụng

—()— Lắp ráp

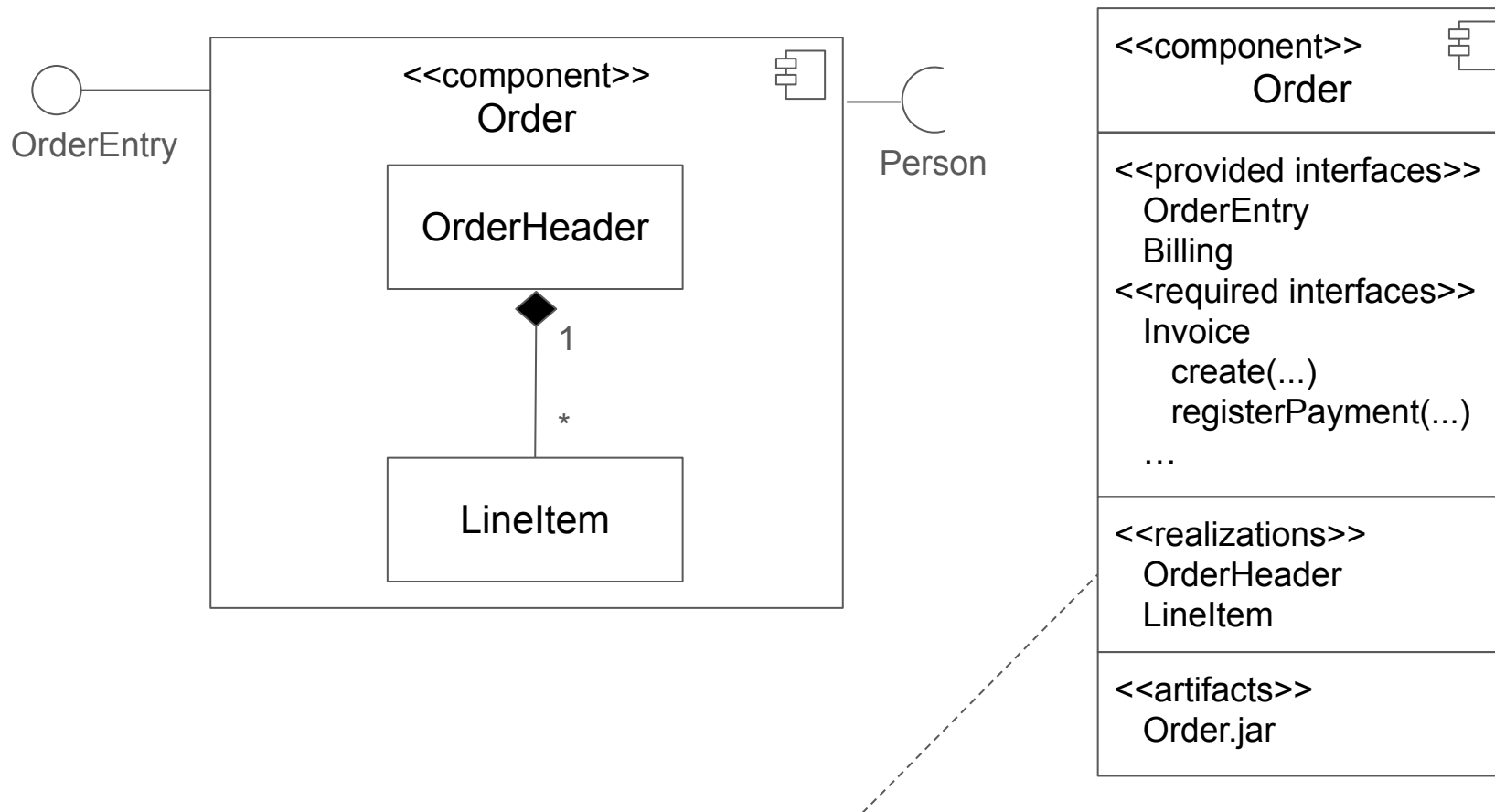
-----> Triển khai

———— Chuyển tiếp

Biểu diễn thành phần và các giao diện

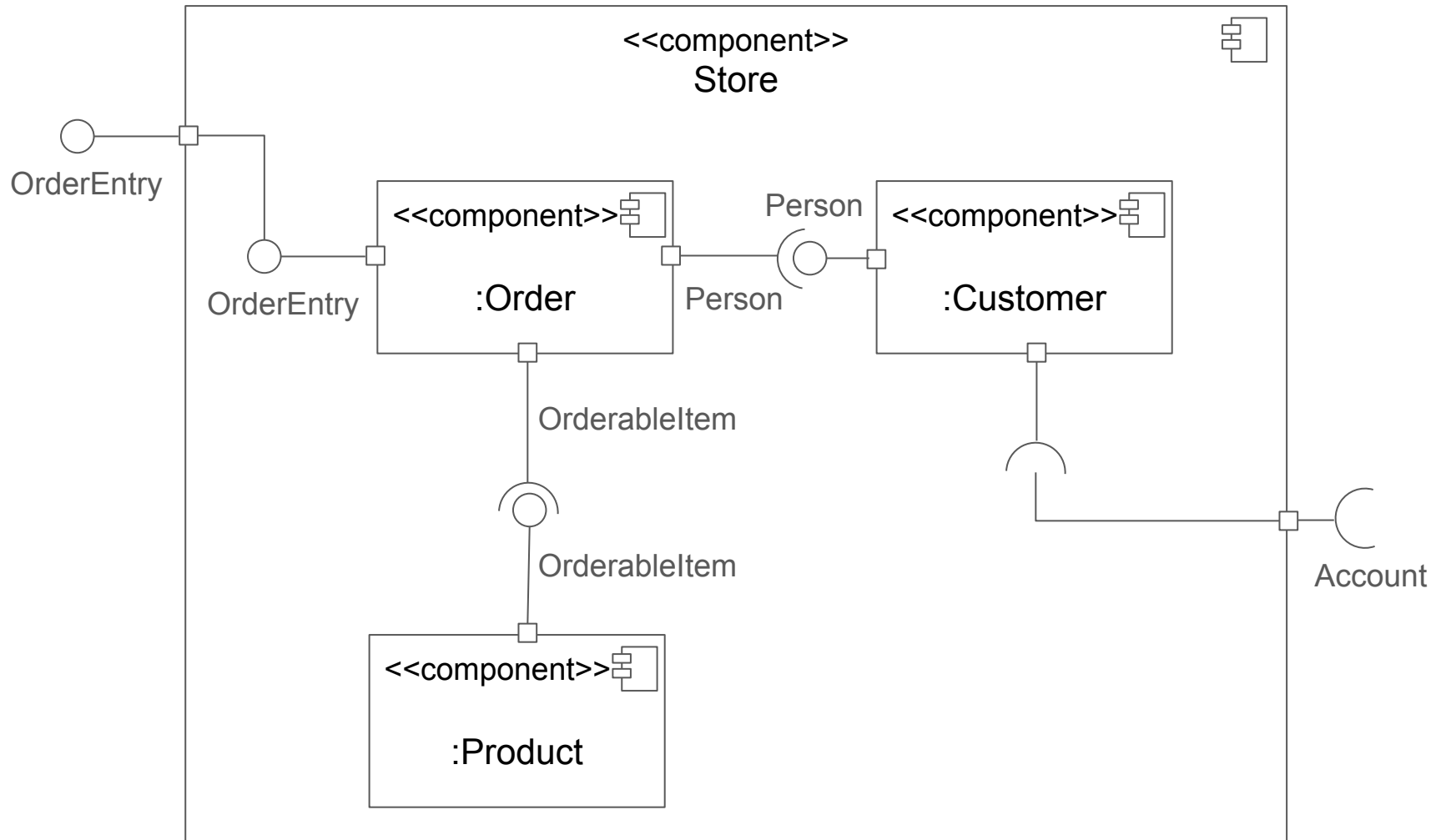


Biểu diễn kết cấu bên trong thành phần

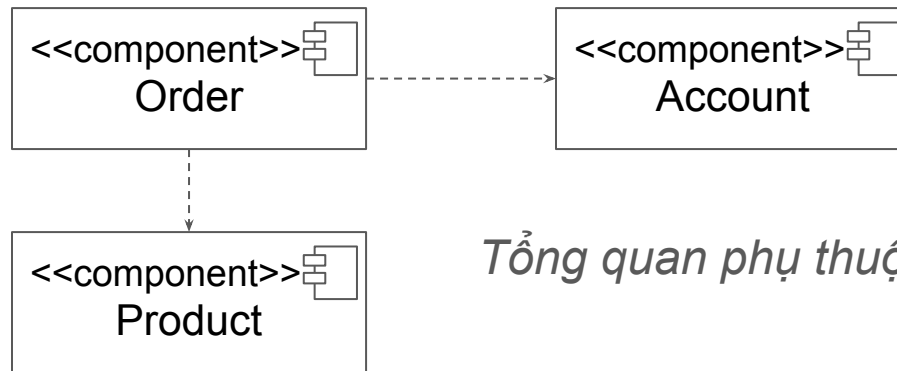


Các ngăn được bổ xung cho các khía cạnh khác nhau

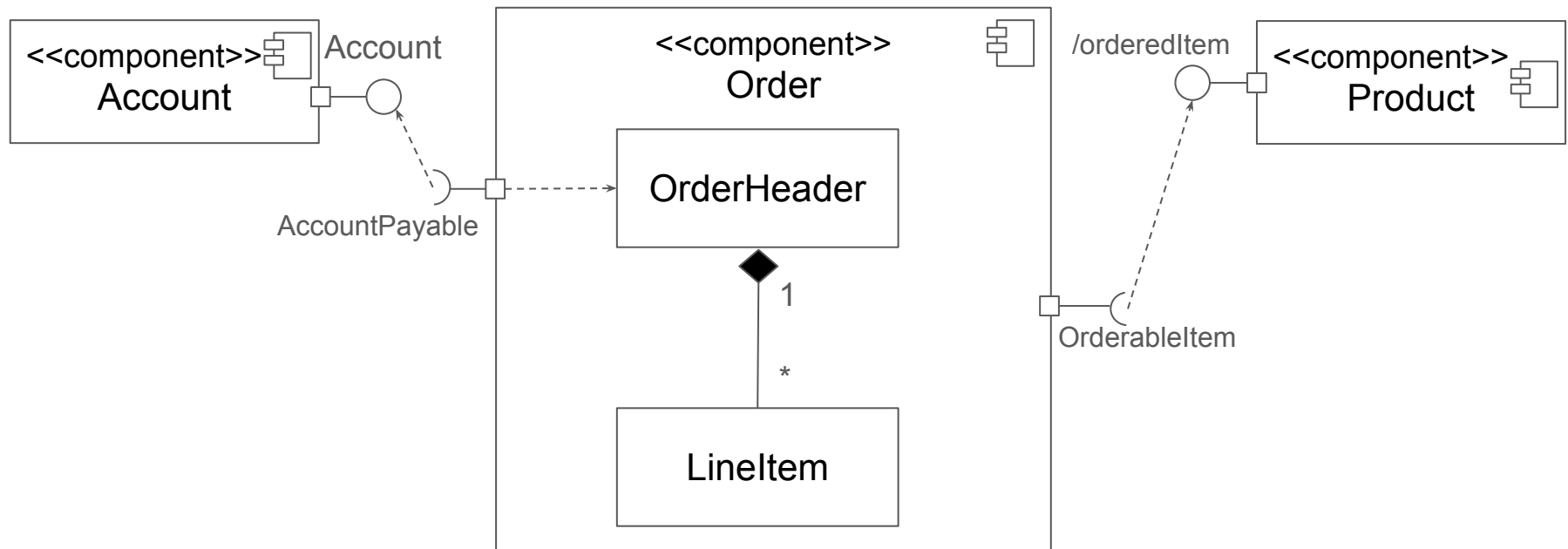
Chuyển tiếp giao diện bên trong thành phần



Biểu diễn phụ thuộc

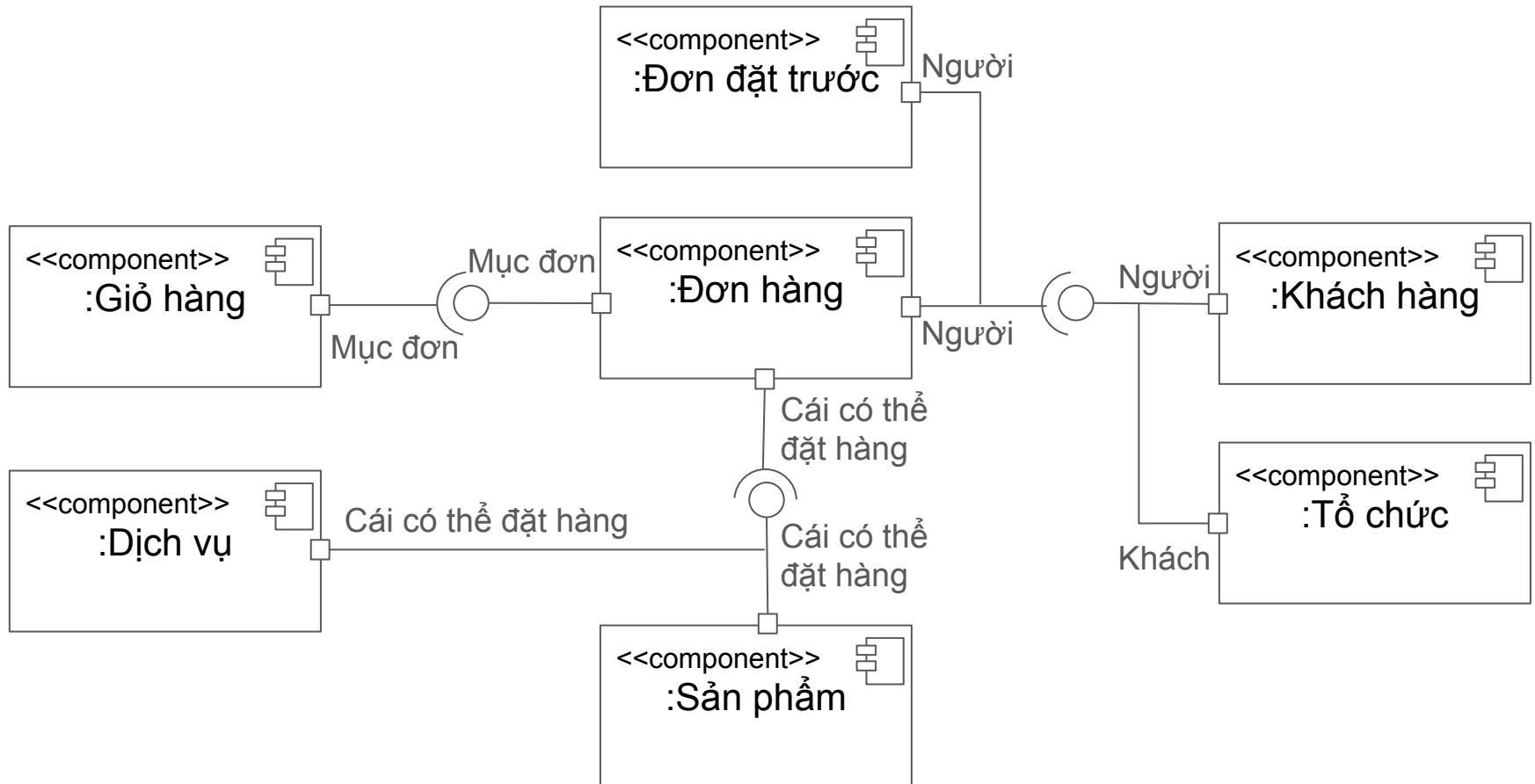


Tổng quan phụ thuộc

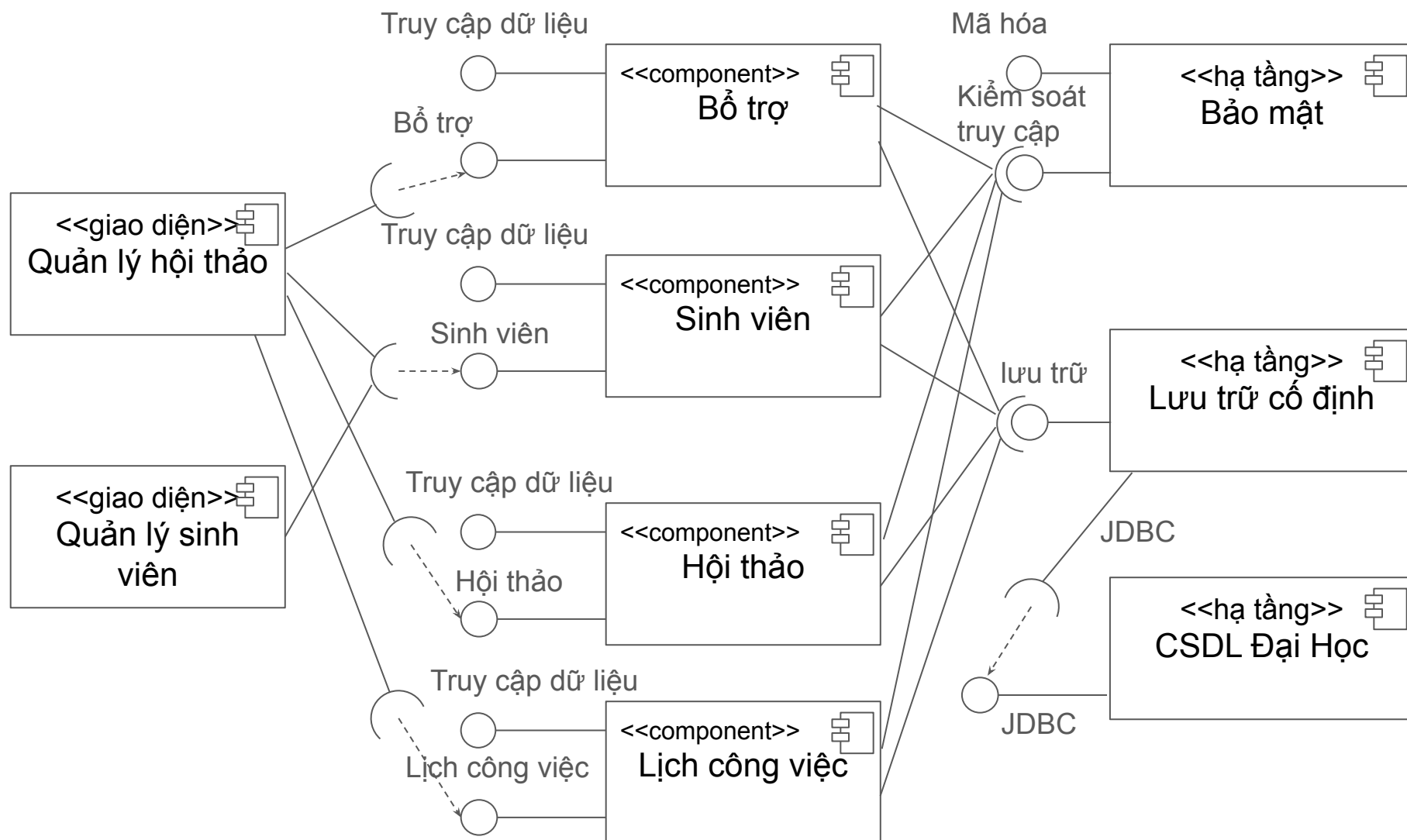


Giao diện được yêu cầu phụ thuộc vào giao diện được cung cấp

Ví dụ 5. Kết cấu từ nhiều thành phần



Ví dụ 6. Sơ đồ thành phần trong UML 2.x



Nội dung

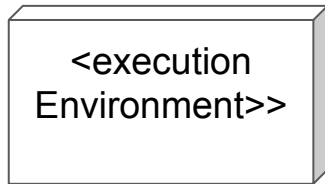
- Mô-đun hóa hệ thống
- Các thành phần kiến trúc Hệ thống
- Mẫu kiến trúc
- Các sơ đồ UML
 - Sơ đồ gói
 - Sơ đồ thành phần
 - Sơ đồ triển khai



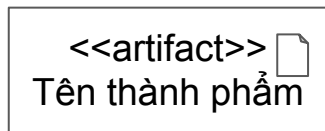
Sơ đồ triển khai

- Biểu diễn kiến trúc vật lý, mối quan hệ giữa phần cứng và phần mềm của hệ thống được triển khai, mô tả cách hệ thống tương tác với môi trường bên ngoài.
- Các nút
 - Thường là các thiết bị tính toán: Máy chủ, Máy vi tính, điện thoại thông minh, v.v..
 - Môi trường thực thi: Linux, macOS, Windows, ...
 - Chứa các thành phần phần mềm
- Các liên kết
 - Biểu diễn kênh trao đổi dữ liệu, kết nối mạng, giao thức.

Sơ đồ triển khai: Hệ ký hiệu

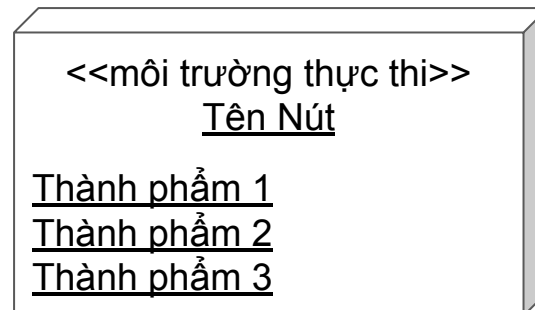
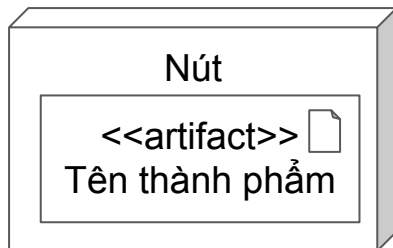


- Nút là tài nguyên tính toán:
 - Thiết bị hoặc
 - Môi trường thực thi,
 - Có thể được kết nối để biểu diễn các kênh trao đổi thông tin theo hình trạng mạng



- Thành phẩm / Artifact:
 - Cấu phần cụ thể của hệ thống
 - Các định dạng:
 - `<<mã nguồn>>` / `<<source>>`, `<<tệp thực thi>>` / `<<executable>>`, `<<jar>>`, v.v..

- Nút với thành phẩm được triển khai:



Sơ đồ triển khai: Hệ ký hiệu₍₂₎

- Đặc tả triển khai

- Đặc tả 1 tập thuộc tính xác định các tham số thực thi của thành phẩm được triển khai trên nút.
- Đặc tả triển khai có thể chứa thuộc tính.

<<đặc tả triển khai>> Tên

<<đặc tả triển khai>> Tên
execution: execKind transaction: Boolean

<<đặc tả triển khai>> <u>Tên</u>
execution: thread transaction: true

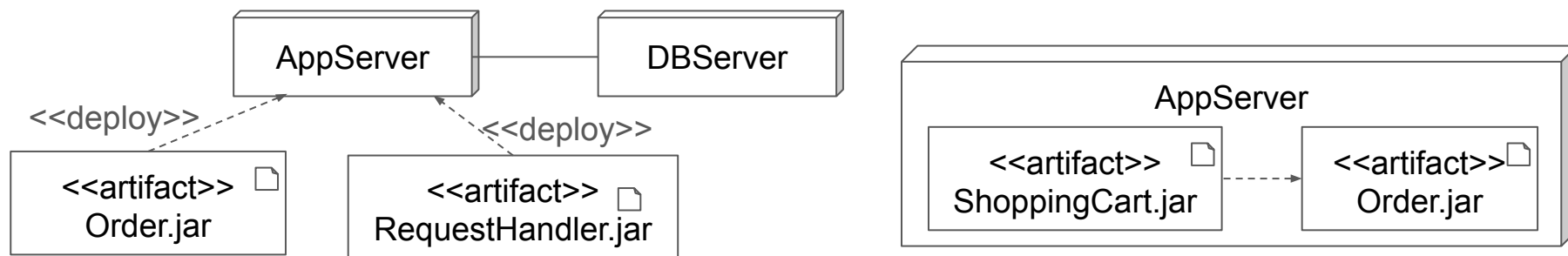
- Thành phẩm phức tạp có thể chứa thực bản của đặc tả triển khai

<<artifact>> Tên thành phẩm {execution = thread, transaction = true}

Sơ đồ triển khai: Hệ ký hiệu₍₃₎

`<<deploy>>`
----->

- Triển khai thành phẩm trên nút



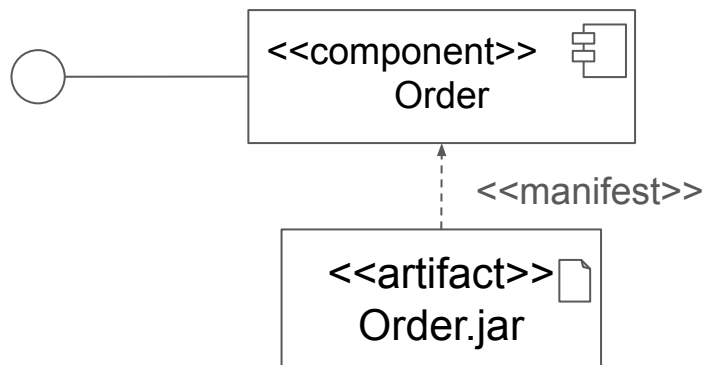
HTTP

- Đường truyền và giao thức được biểu diễn bằng quan hệ liên kết

`<<manifest>>`

----->

- Quan hệ manifest/xuất bản: Cho biết cơ sở của thành phẩm, cái hình thành nên thành phẩm.



Sơ đồ triển khai: Thành phẩm

Thành phẩm/Artifact biểu diễn một đầu ra thực tế, được triển khai trong các nút.

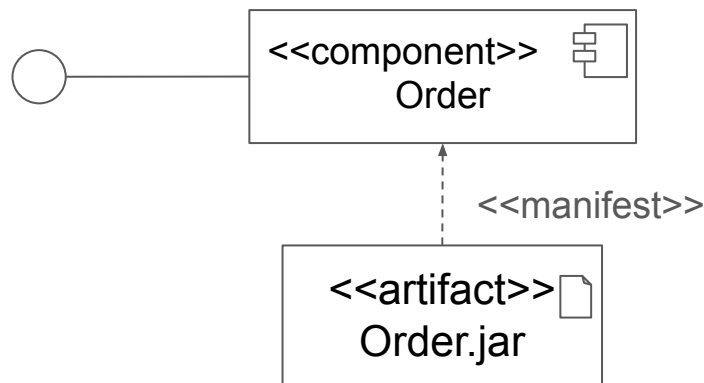


Một số định dạng thành phẩm tiêu chuẩn:

- Tập - <<file>> một tập cụ thể trong hệ thống tập
- Tài liệu - <<document>> một tài liệu, có thể là mã nguồn hoặc tập thực thi
- Thư viện - <<library>> một thư viện tĩnh hoặc động
- Tập thực thi - <<executable>> một chương trình có thể được thực thi trên máy tính.

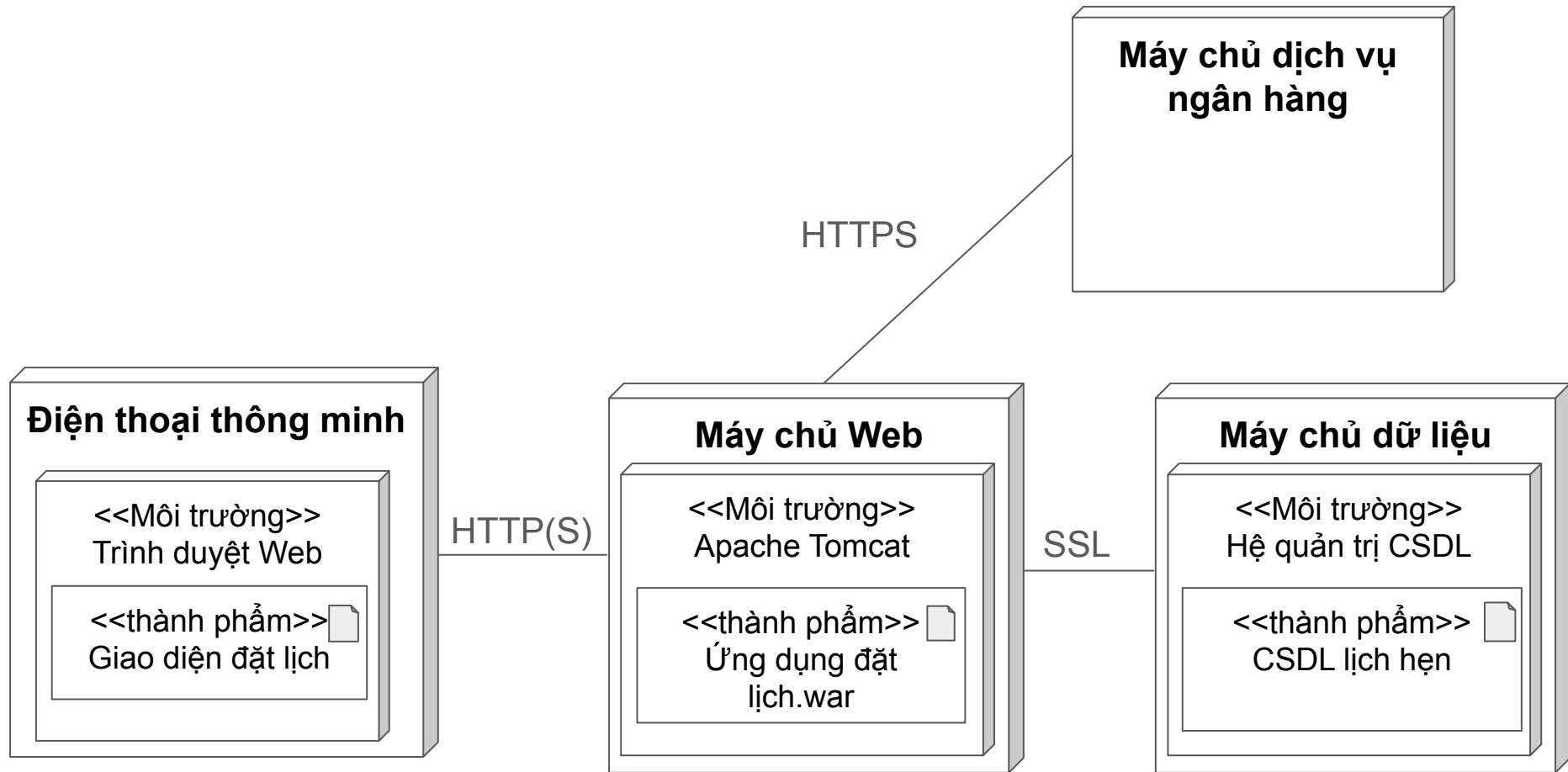
Quan hệ manifest

- Biểu diễn triển khai vật lý của 1 hoặc nhiều cấu phần của hệ thống như các thành phẩm.
- Thường được sử dụng để biểu diễn triển khai thành phần như thành phẩm:
 - Mỗi thành phần có thể được triển khai như 1 thành phẩm
 - Mỗi thành phẩm có thể bao gồm nhiều thành phần

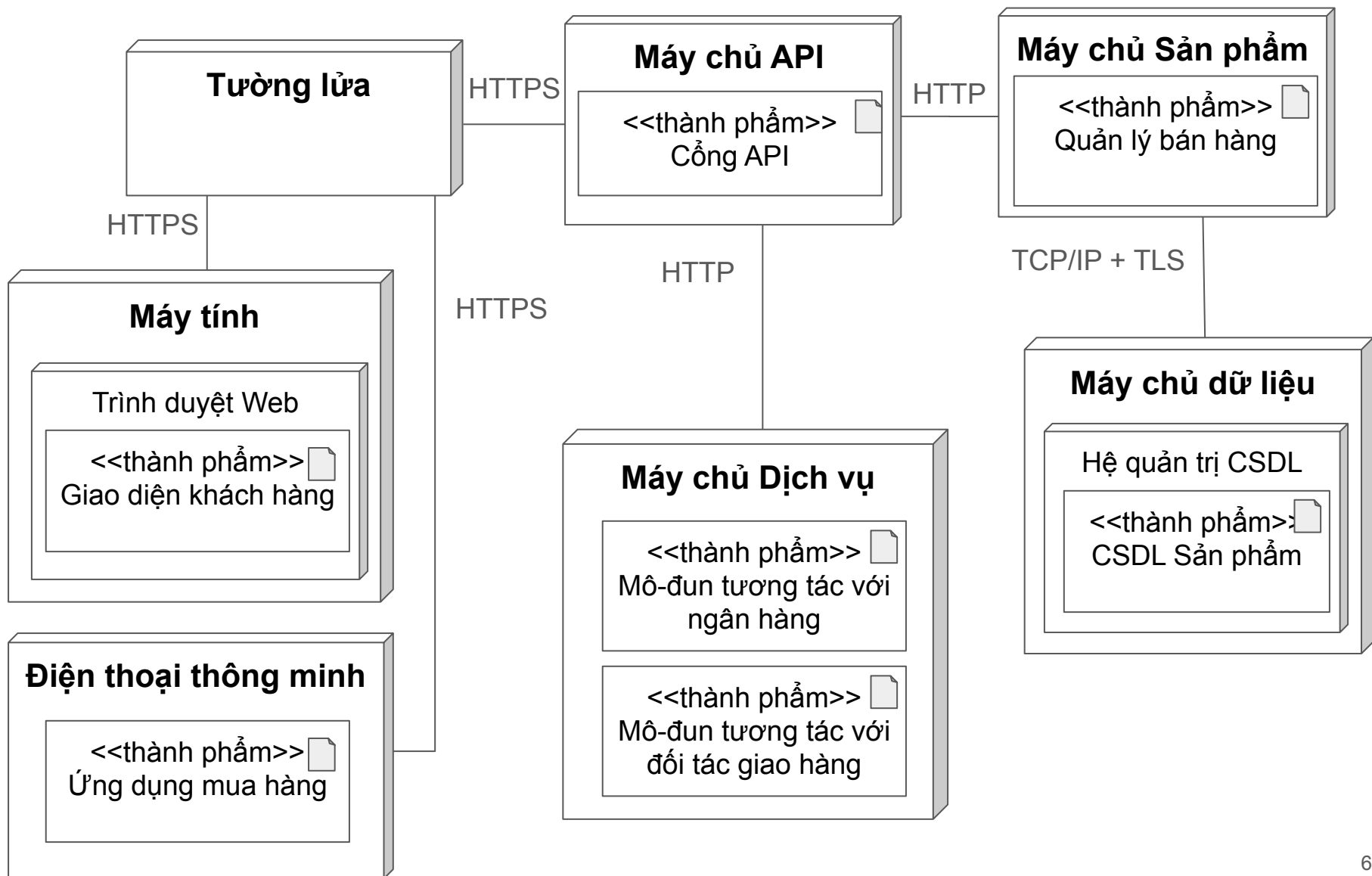


- Ngoài ra có thể sử dụng với bất kỳ phần tử nào khác nếu có thể đóng gói / `PackageableElement`

Ví dụ 7. Sơ đồ triển khai hệ thống 3-dãy



Ví dụ 8. Sơ đồ triển khai hệ thống bán hàng



Bài tập

Phân tích các vấn đề với kết cấu sau và phác thảo giải pháp hướng tới liên kết lỏng giữa các phần tử: Vẽ sơ đồ lớp và sơ đồ tuần tự.

