Nhập môn Công nghệ phần mềm

Tuần 6: Thiết kế kiến trúc phần mềm

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Nội dung của slide này được dịch và hiệu chỉnh dựa vào các slides của Ian Sommerville

Nội dung

1. Quyết định chọn kiến trúc thiết kế
2. Các góc nhìn về kiến trúc
3. Các kiến trúc mẫu
4. Các kiến trúc ứng dụng

Kiến trúc phần mềm

- Thiết kế kiến trúc liên quan đến việc hiểu một hệ thống được tổ chức như thế nào và thiết kế toàn bộ kiến trúc của hệ thống đó.
- Đầu ra: mô hình kiến trúc.

Thiết kế kiến trúc

- Là giai đoạn đầu tiên của một quy trình thiết kế hệ thống.
- Là cầu nối giữa yêu cầu phần mềm và thiết kế.
- Thực tế: Thiết kế kiến trúc thường được tiến hành song song với các hoạt động đặc tả.
- Bước này giúp nhận diện các component chính của hệ thống và cách thức giao tiếp giữa các component.

Các mức trừu tượng của kiến trúc

- Kiến trúc phần mềm nhỏ
  - Liên quan đến kiến trúc của các chương trình đơn lẻ.
  - Một chương trình được phân thành các component.
- Kiến trúc hệ thống lớn
  - Liên quan đến kiến trúc của một hệ thống phức tạp gồm nhiều hệ thống khác, chương trình và các component của chương trình.
  - Những hệ thống này được phân tán trên nhiều máy tính khác nhau, có thể được sở hữu và quản lý bởi nhiều công ty khác nhau.

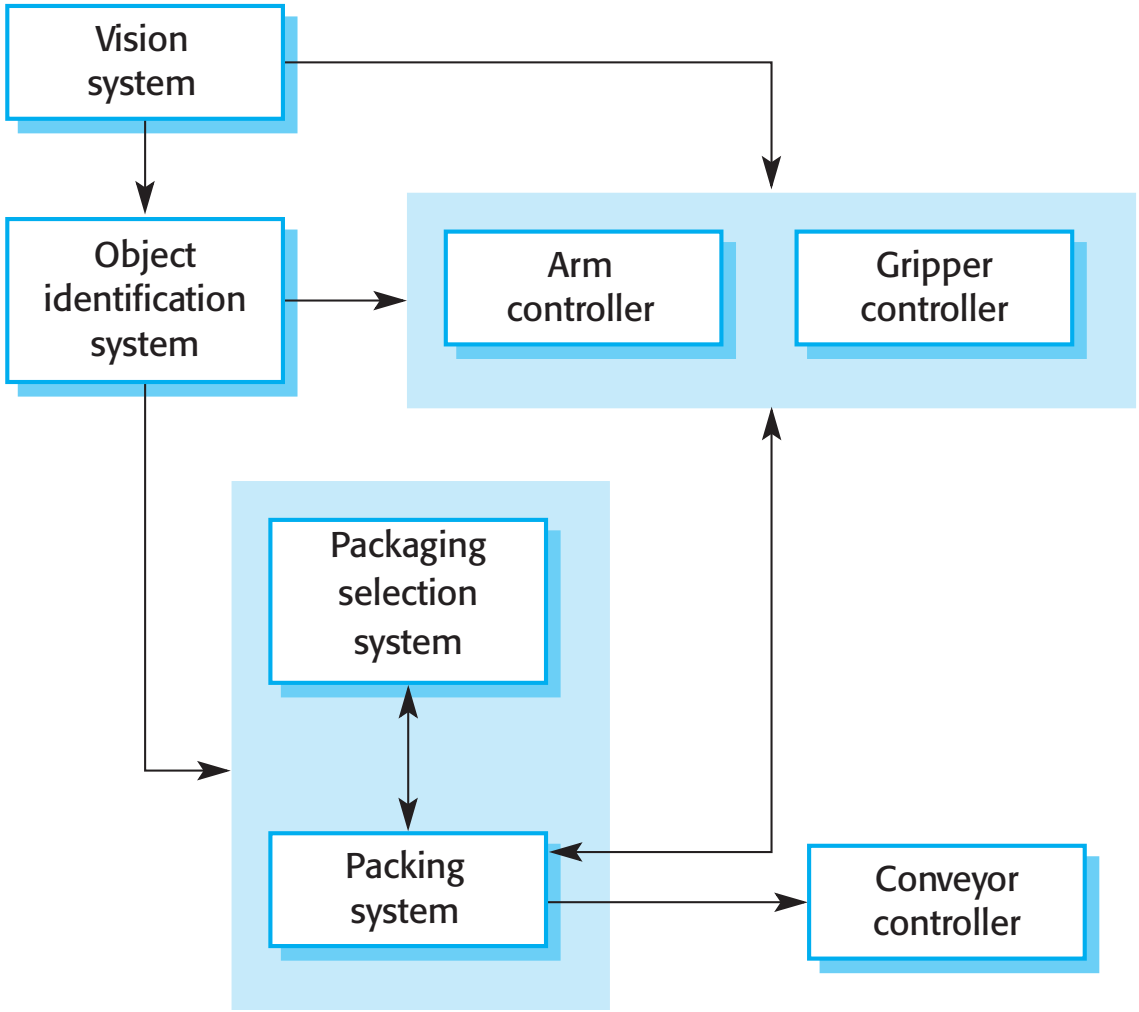
Ưu điểm của kiến trúc

- Giao tiếp với các stakeholder
  - Kiến trúc là biểu diễn mức cao của hệ thống
  - Được sử dụng để thảo luận với các stakeholder.
- Phân tích hệ thống
  - Là cách để phân tích xem liệu hệ thống có đáp ứng được các yêu cầu phi chức năng hay không.
- Tái sử dụng
  - Kiến trúc có thể được tái sử dụng cho nhiều hệ thống khác.

Biểu diễn kiến trúc

- Sử dụng biểu đồ khối [Hofmeister et al., 2000]
  - Đơn giản, không mang tính hình thức để chỉ ra các thực thể và quan hệ giữa chúng.
  - Biểu diễn một góc nhìn toàn cảnh về cấu trúc hệ thống: những người thuộc các lĩnh vực khác nhau vẫn có thể hiểu được.
- Việc sử dụng loại kiến trúc này bị chỉ trích trong một thời gian dài
  - Thiếu ngữ nghĩa, không chỉ ra được loại quan hệ giữa các thực thể và không chỉ ra các thuộc tính của thực thể trong kiến trúc. [Bass et al., 2003]

Kiến trúc của hệ thống điều khiển robot



Sử dụng các mô hình kiến trúc

- Là phương tiện để thảo luận về thiết kế hệ thống
  - Thiết kế kiến trúc ở mức cao có ích khi giao tiếp với các stakeholder và lên kế hoạch dự án vì nó không đi sâu vào chi tiết.
  - Các stakeholder có thể hiểu được mô hình trừu tượng của hệ thống ➔ hỗ trợ việc thảo luận về toàn bộ hệ thống mà không bị rối bởi việc quá đi sâu vào chi tiết.
- Là cách để viết tài liệu về kiến trúc đã được thiết kế
  - Mục tiêu: tạo ra một mô hình hệ thống hoàn chỉnh trong đó nó chỉ ra được các component khác nhau trong hệ thống, giao diện và sự kết nối của chúng.

Nội dung

1. Quyết định chọn kiến trúc thiết kế
2. Các góc nhìn về kiến trúc
3. Các kiến trúc mẫu
4. Các kiến trúc ứng dụng

Quyết định chọn kiến trúc thiết kế

- Thiết kế kiến trúc là một quy trình sáng tạo
  - Thiết kế tổ chức của một hệ thống thoả mãn được các yêu cầu chức năng và yêu cầu phi chức năng.
  - Các hoạt động trong quy trình phụ thuộc vào loại ứng dụng được phát triển, kinh nghiệm của người thiết kế kiến trúc và các yêu cầu cụ thể của hệ thống.
- Thiết kế kiến trúc được xem như là một chuỗi các quyết định hơn là một chuỗi tuần tự các hoạt động.

Quyết định chọn kiến trúc thiết kế

1. Có thể sử dụng kiến trúc ứng dụng tổng quát nào như là một template cho hệ thống sẽ được thiết kế không?
2. Hệ thống được phân tán trên nhiều core phần cứng hoặc processor như thế nào?
3. Có mẫu kiến trúc nào phù hợp?
4. Phương pháp nào được sử dụng để cấu trúc hoá hệ thống?
5. Hệ thống được phân rã thành các module như thế nào?
6. Chiến thuật nào được sử dụng để điều khiển hoạt động của các component trong hệ thống?
7. Kiến trúc được thiết kế như thế nào để thoả mãn tốt nhất các yêu cầu phi chức năng của hệ thống?
8. Kiến trúc được viết thành tài liệu như thế nào?

Tái sử dụng kiến trúc

- Các hệ thống có cùng lĩnh vực thường có cấu trúc tương tự nhau
  - Phản ánh những đặc điểm của lĩnh vực đó.
- Kiến trúc của một hệ thống có thể được thiết kế dựa vào một hoặc nhiều mẫu kiến trúc có sẵn (architectural pattern).
- Mẫu kiến trúc
  - Mô tả về kiến trúc của một hệ thống.
  - Chứa các đặc điểm chính của một kiến trúc đã được sử dụng qua các hệ thống phần mềm khác nhau.

Kiến trúc và đặc điểm của hệ thống

- Hiệu năng (Performance)
  - Định vị các chức năng quan trọng trong một số ít component và giảm thiểu giao tiếp. Những component này được triển khai trên cùng một máy tính.
- Bảo mật (Security)
  - Sử dụng kiến trúc phân tầng với các phần quan trọng được đặt ở các lớp trong cùng.
- An toàn (Safety)
  - Định vị các thao tác liên quan đến an toàn trong một số ít các hệ thống con.
- Tính thường trực (Availability)
  - Thiết kế sẵn các component dư thừa sao cho có thể thay thế hoặc cập nhật các component mà không phải dừng hệ thống, nghĩa là đảm bảo cho hệ thống hoạt động liên tục.
- Tính dễ bảo trì (Maintainability)
  - Sử dụng các component nhỏ, chi tiết, có thể thay thế được.

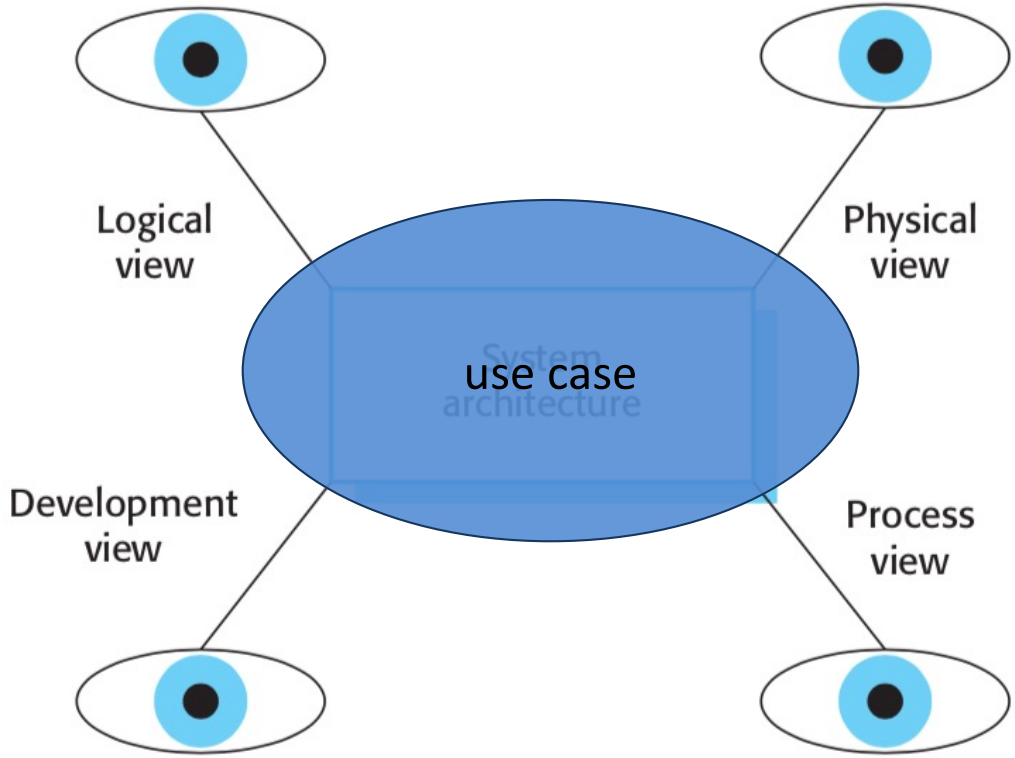
Nội dung

1. Quyết định chọn kiến trúc thiết kế
2. Các góc nhìn về kiến trúc
3. Các kiến trúc mẫu
4. Các kiến trúc ứng dụng

Các góc nhìn về mặt kiến trúc

- Mỗi mô hình kiến trúc chỉ thể hiện một góc nhìn về hệ thống.
- Khi thiết kế và viết tài liệu: cần biểu diễn hệ thống phần mềm ở nhiều góc nhìn khác nhau.

Mô hình kiến trúc phần mềm 4 + 1



Nội dung

1. Quyết định chọn kiến trúc thiết kế
2. Các góc nhìn về kiến trúc
3. Các kiến trúc mẫu
4. Các kiến trúc ứng dụng

Kiến trúc mẫu(Architectural pattern)

- Là một phương tiện để biểu diễn, chia sẻ và tái sử dụng lại các kiến thức về hệ thống phần mềm.
- Các kiến trúc mẫu là dạng mô tả trừu tượng, thường chứa thông tin :
  - Mô tả
  - Sử dụng khi nào
  - Ưu nhược điểm.
- Các mẫu được biểu diễn sử dụng bảng và mô tả đồ họa.

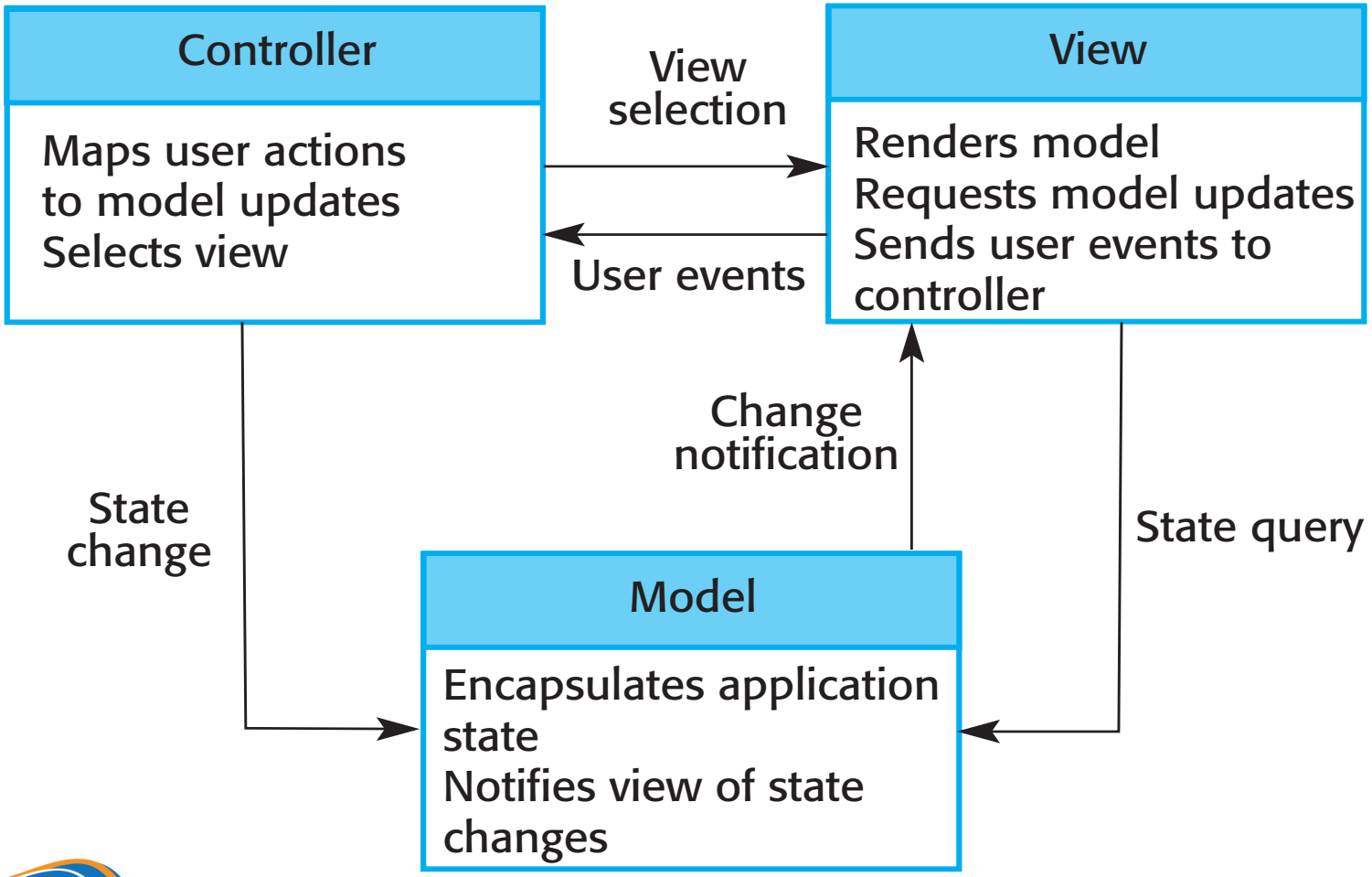
Một số mô hình kiến trúc mẫu

- Model-View-Controller (MVC)
- Kiến trúc phân tầng
- Repository
- Client–server
- Pipe and filter

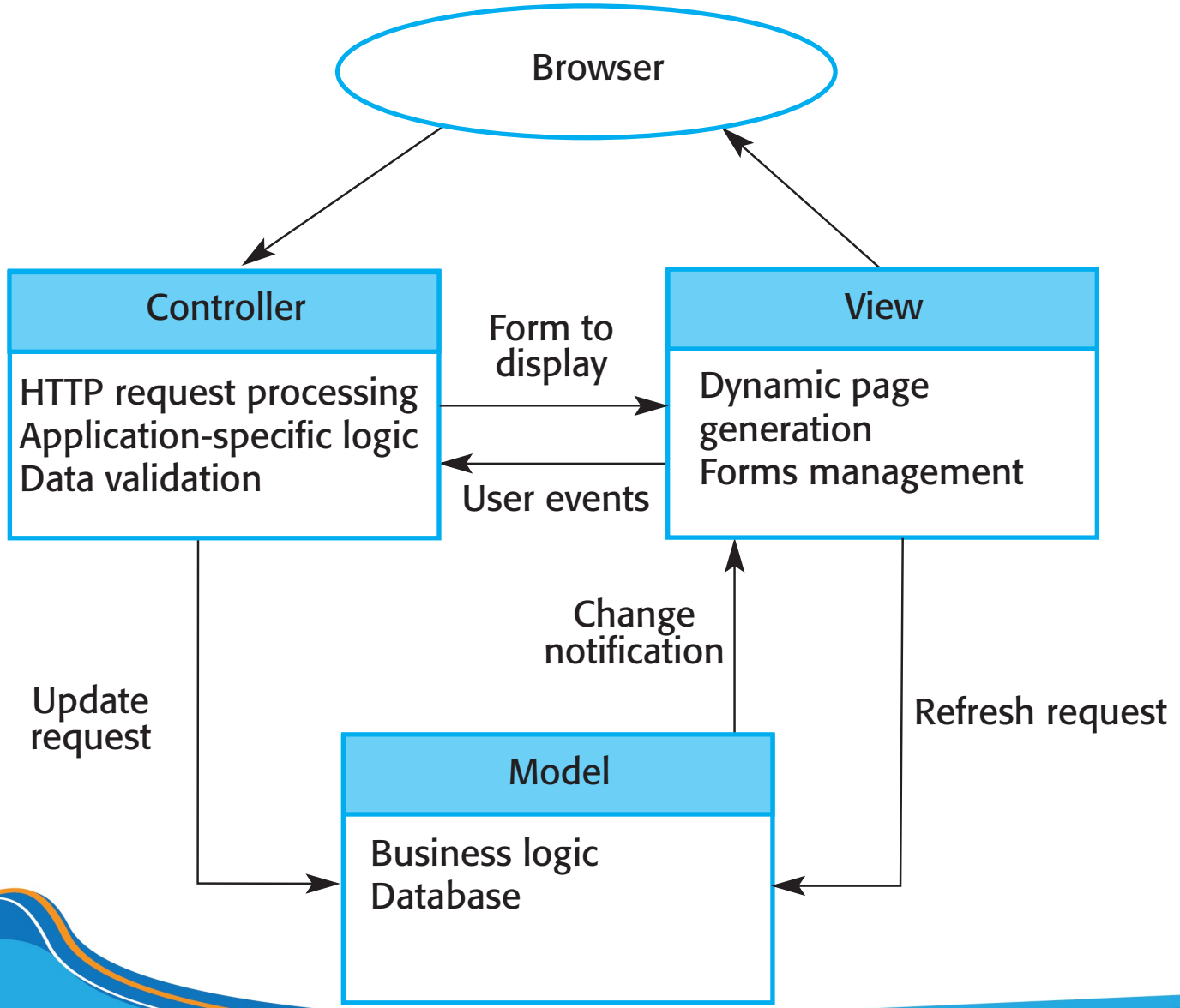
Mô hình MVC

Tên	Mô hình MVC (Model-View-Controller)
Mô tả	Tách riêng phần biểu diễn và phần tương tác ra khỏi dữ liệu hệ thống. Ba component tương tác với nhau. • **Model component**: quản lý dữ liệu hệ thống và các thao tác trên dữ trên dữ liệu đó. • **View component**: định nghĩa và quản lý cách dữ liệu được biểu diễn tới người dùng như thế nào. • **Controller component**: Quản lý tương tác người dùng (ví dụ như ấn phím, nhấp chuột, ...) và chuyển các tương tác này tới View và Model.
Sử dụng khi nào	• Khi có nhiều cách biểu diễn và tương tác với dữ liệu. • Khi chưa biết được các yêu cầu tương lai cho tương tác và biểu diễn dữ liệu.
Ưu điểm	Cho phép dữ liệu thay đổi độc lập với hiển thị và ngược lại. Hỗ trợ biểu diễn theo nhiều cách khác nhau trên cùng một dữ liệu.
Nhược điểm	Có thể chứa code bổ sung và code sẽ phức tạp hơn khi mô hình dữ liệu và mô hình tương tác đơn giản.

Tổ chức của mô hình MVC



Ví dụ: Kiến trúc ứng dụng Web sử dụng mô hình MVC



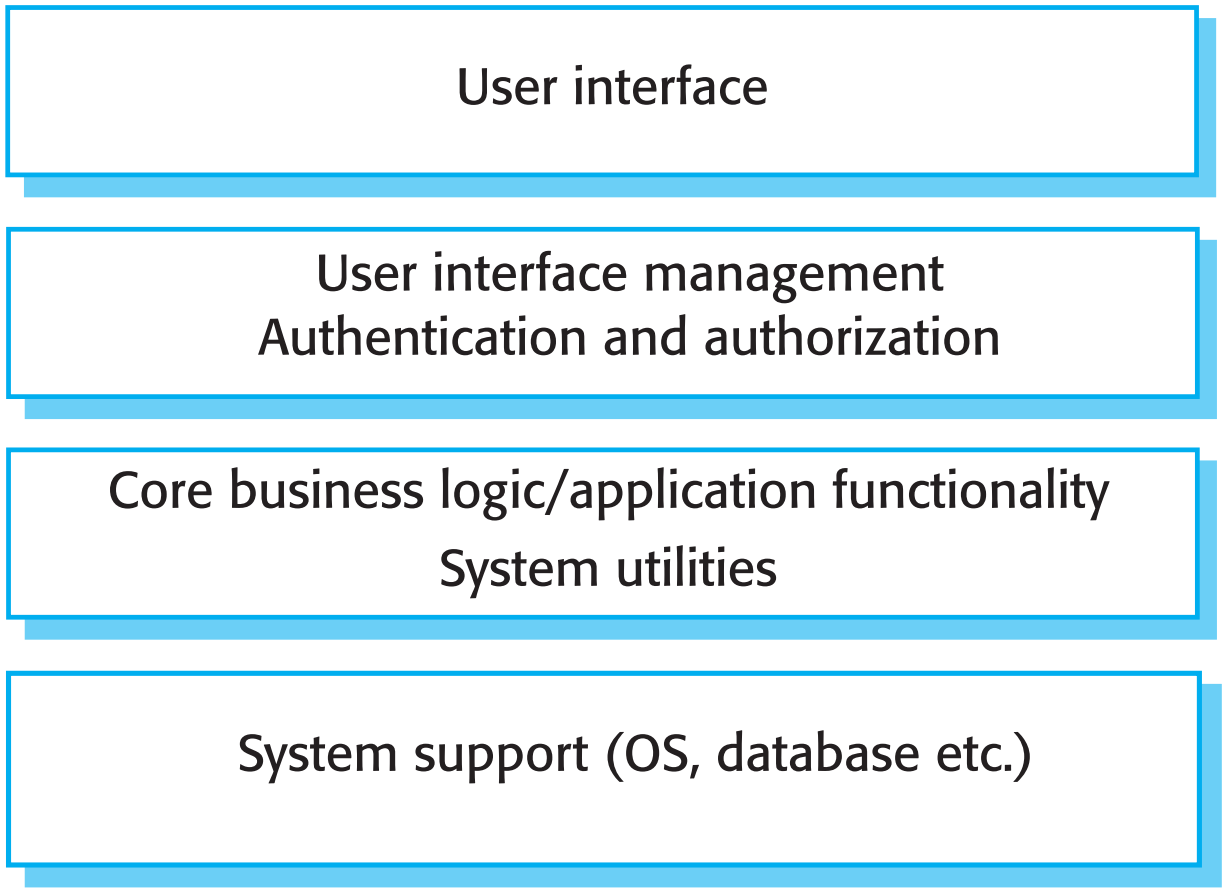
Kiến trúc phân tầng

- Được sử dụng để mô hình hóa giao diện của các hệ thống con.
- Tổ chức hệ thống thành một tập các tầng, mỗi tầng cung cấp một tập các dịch vụ.
- Hỗ trợ việc phát triển dần dần các hệ thống con trên các tầng khác nhau.
  - Khi giao diện của tầng thay đổi, chỉ các tầng lân cận mới bị ảnh hưởng.

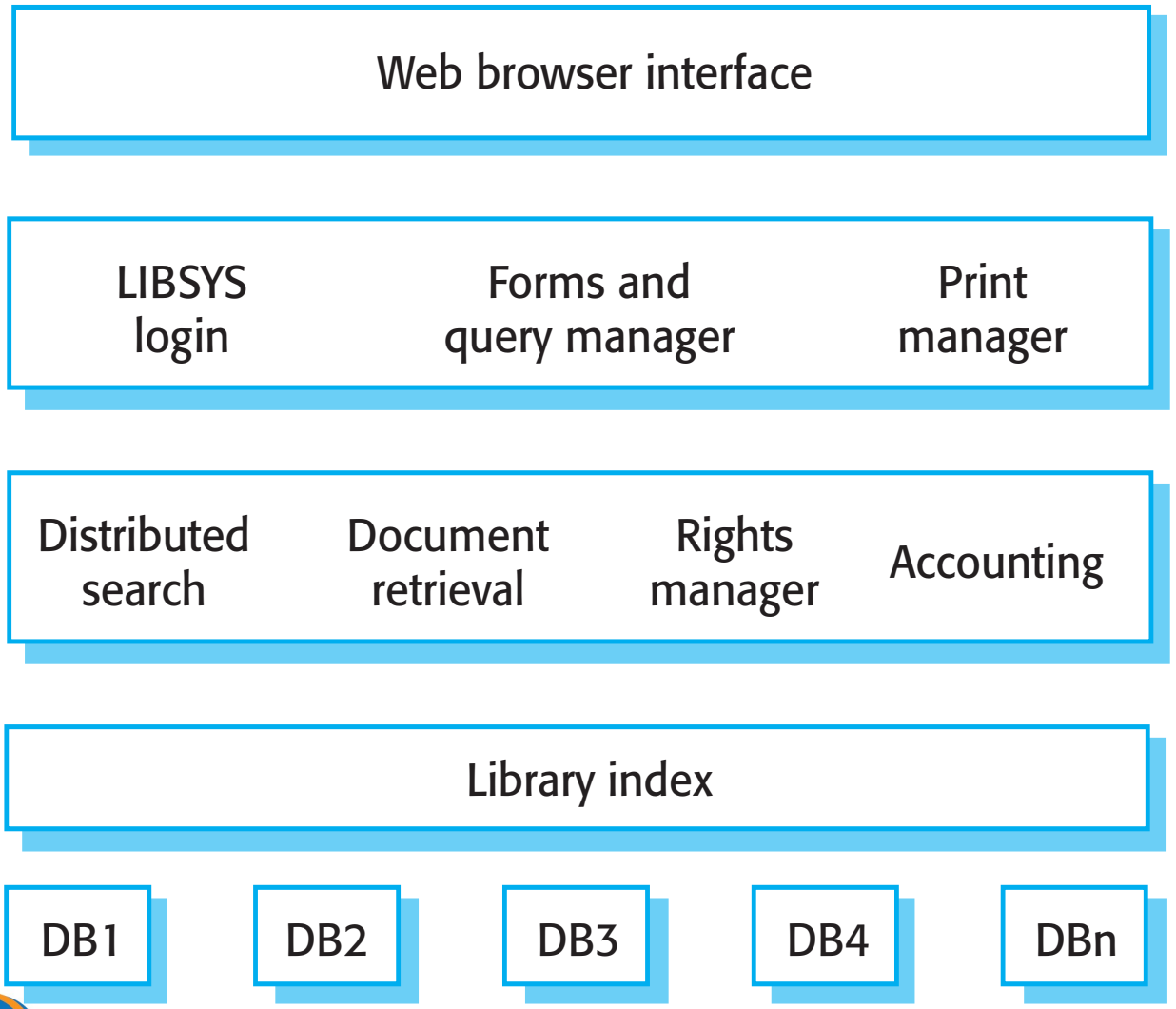
Mô hình kiến trúc phân tầng

Tên	Kiến trúc phân tầng
Mô tả	Tổ chức hệ thống thành các tầng, mỗi tầng chứa các chức năng liên quan đến nhau. Một tầng cung cấp các dịch vụ cho tầng trên của nó vì vậy các tầng thấp nhất biểu diễn các dịch vụ lõi được sử dụng trong toàn bộ hệ thống.
Sử dụng khi nào	• Khi xây dựng các tính năng mới dựa trên những hệ thống có sẵn; • Khi việc phát triển được dàn trải trên nhiều nhóm khác nhau và mỗi nhóm chịu trách nhiệm về chức năng của một tầng; • Khi có yêu cầu về bảo mật ở nhiều mức độ.
Ưu điểm	• Cho phép thay thế các phần miễn là interface được duy trì. • Các chức năng dư thừa (ví dụ như phân quyền) có thể được cung cấp ở mỗi tầng để tăng độ tin cậy của hệ thống.
Nhược điểm	• Thực tế: cung cấp một sự phân chia rõ rệt giữa các tầng thường rất khó khăn và tầng cao hơn có thể tương tác trực tiếp với tầng thấp hơn hơn là thông qua một tầng bên dưới nó. • Hiệu năng cũng có thể là một vấn đề vì có nhiều mức diễn giải của một yêu cầu dịch vụ khi nó được thực hiện tại mỗi tầng.

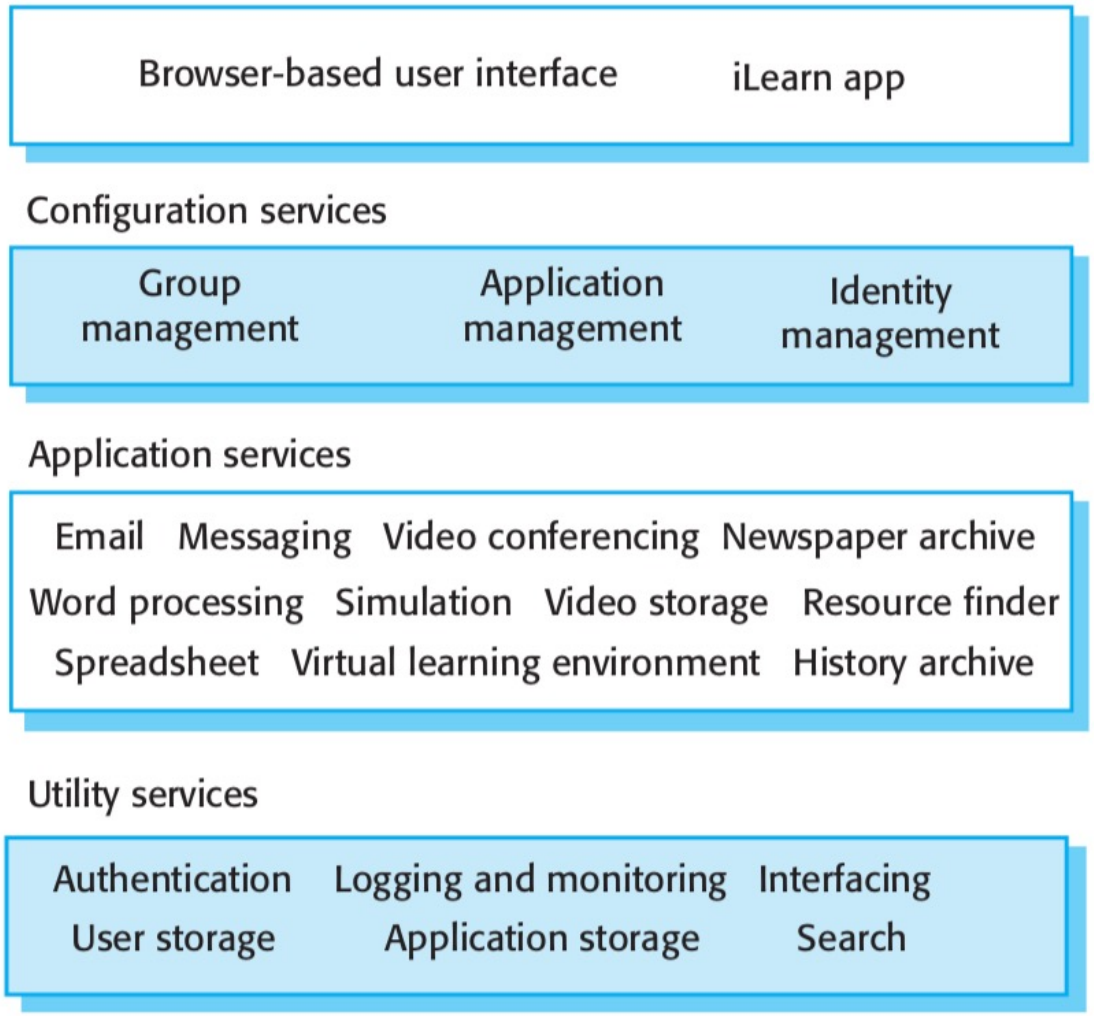
Một kiến trúc phân tầng tổng quát



Kiến trúc của hệ thống LIBSYS



Kiến trúc của hệ thống iLearn



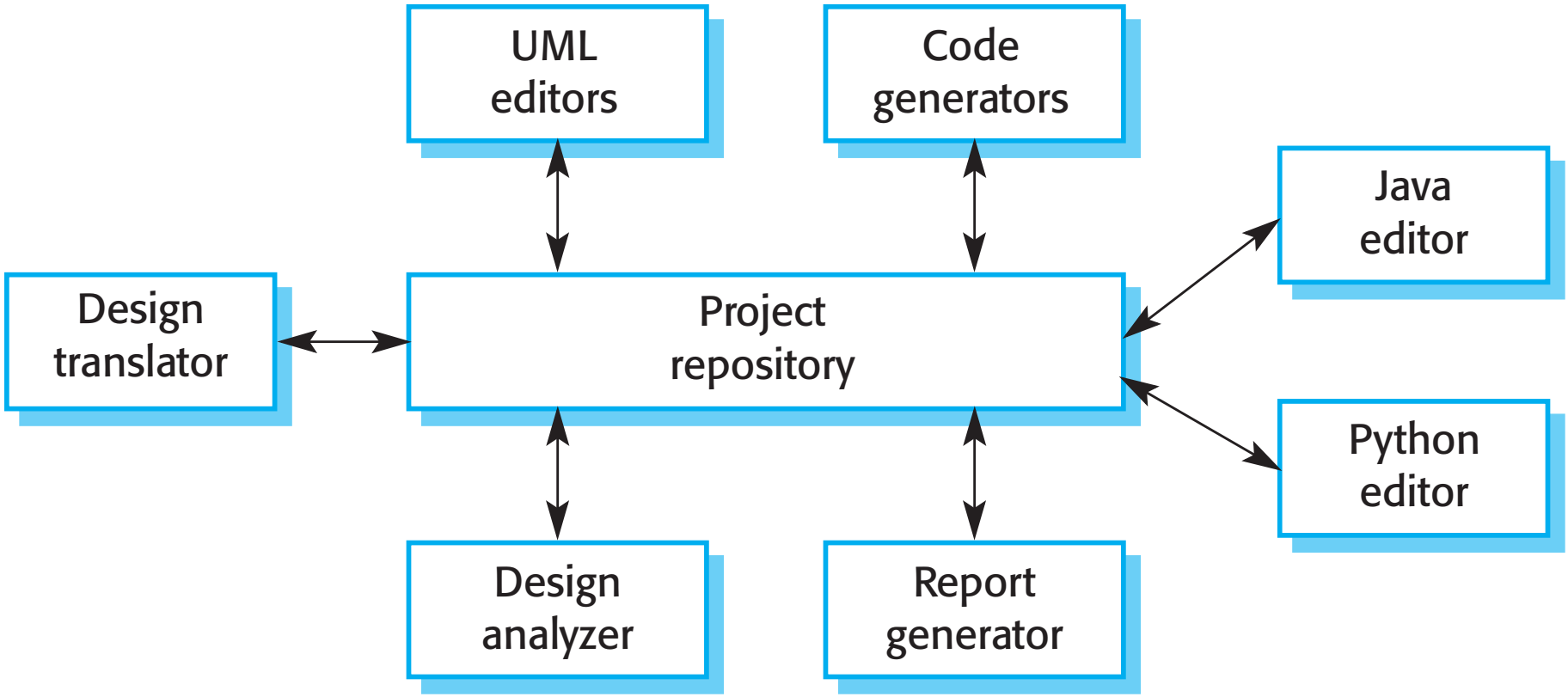
Kiến trúc repository

- Các hệ thống con phải trao đổi dữ liệu với nhau. Có hai khả năng:
  - Việc chia sẻ dữ liệu được thực hiện ở cơ sở dữ liệu trung tâm hay còn gọi là kho dữ liệu, kho này được truy cập bởi tất cả các hệ thống con;
  - Mỗi hệ thống con duy trì một cơ sở dữ liệu riêng và chuyển dữ liệu trực tiếp tới các hệ thống con khác.
- Khi có một lượng lớn dữ liệu cần chia sẻ: đây là mô hình phổ biến nhất và là cơ chế chia sẻ dữ liệu hiệu quả nhất.

Mô hình Repository

Tên	Mô hình Repository
Mô tả	• Tất cả các dữ liệu trong hệ thống được quản lý ở một kho trung tâm, kho này được truy cập bởi tất cả các component của hệ thống. • Các component không tương tác trực tiếp với nhau, chỉ thông qua kho chung thôi.
Sử dụng khi nào	• Khi ta có một hệ thống trong đó một lượng lớn thông tin sinh ra phải được lưu trữ trong một thời gian dài. • Sử dụng trong các hệ thống hướng dữ liệu
Ưu điểm	• Các component có thể độc lập với nhau – chúng không cần biết sự tồn tại của các component khác. Các thay đổi xảy ra ở một component không ảnh hưởng tới các component khác. • Tất cả các dữ liệu có thể được quản lý một cách nhất quán (ví dụ như backup dữ liệu được thực hiện đồng thời) vì tất cả dữ liệu được lưu trữ ở cùng một nơi.
Nhược điểm	• Các vấn đề xảy ra trên kho chung ảnh hưởng đến toàn hệ thống. • Có thể không hiệu quả trong việc tổ chức các giao tiếp thông qua kho. • Phân tán kho trên nhiều máy tính có thể khó khăn.

Kiến trúc repository cho một IDE



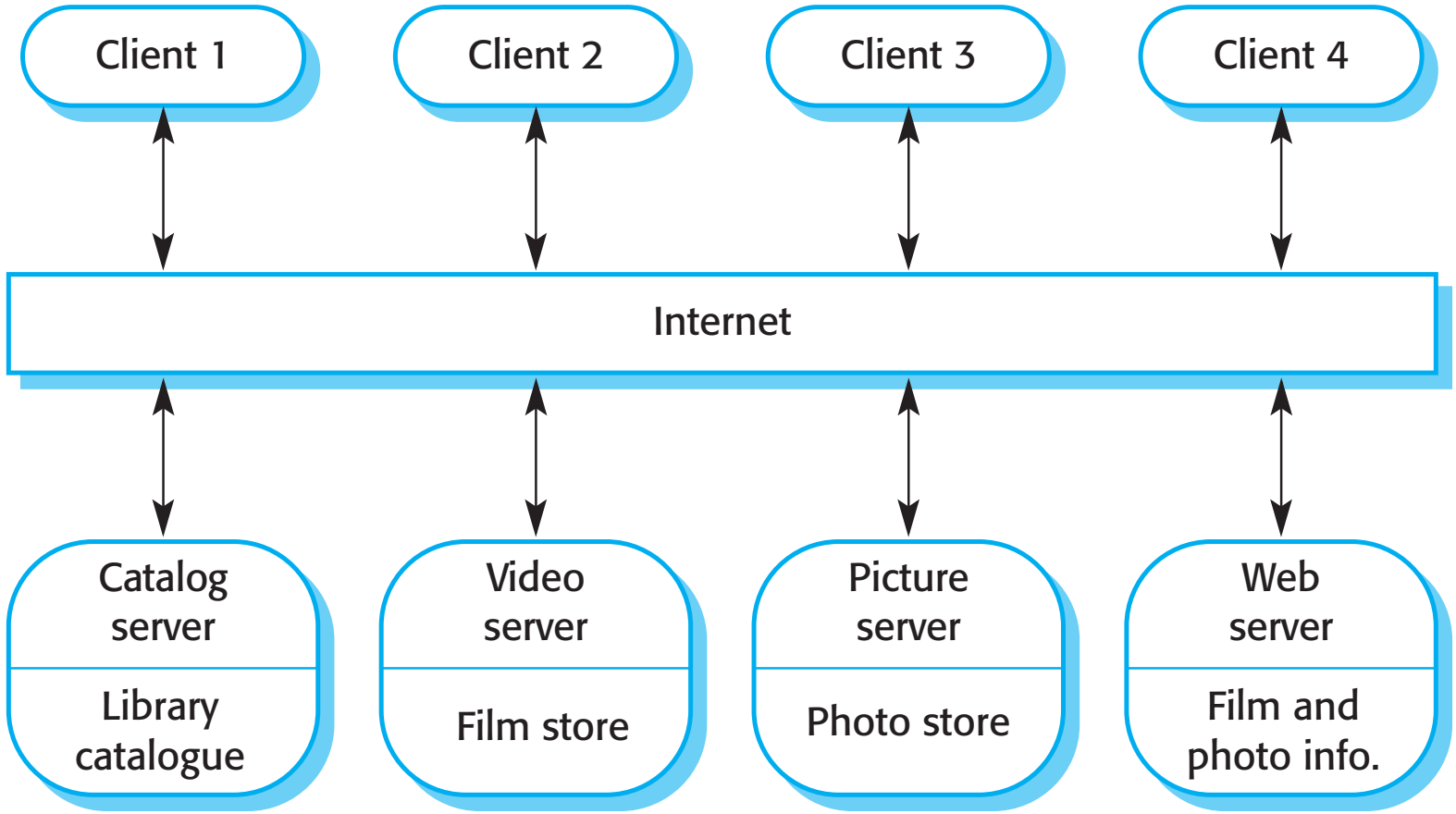
Kiến trúc client-server

- Các mô hình hệ thống phân tán chỉ ra cách dữ liệu và các xử lý được phân tán trên nhiều component như thế nào.
  - Có thể được cài đặt trên một máy đơn.
- Tập hợp các server độc lập cung cấp các dịch vụ cụ thể ví dụ như in ấn, quản trị dữ liệu, ...
- Tập hợp các khách hàng triệu gọi các dịch vụ này.
- Hệ thống mạng cho phép người dung truy cập vào các server.

Mô hình client–server

Tên	Mô hình client-server
Mô tả	• Chức năng của hệ thống được tổ chức thành các dịch vụ, mỗi dịch vụ được đặt trên một server riêng lẻ. • Khách hàng là người sử dụng các dịch vụ này và truy cập vào các server để sử dụng dịch vụ.
Sử dụng khi nào	• Khi dữ liệu trong một cơ sở dữ liệu chia sẻ phải truy cập từ nhiều nơi. Vì các server được truy cập từ nhiều nơi khác nhau, có thể được sử dụng khi tải trên hệ thống thay đổi.
Ưu điểm	• Server được phân tán trên mạng. • Chức năng chung (dịch vụ in ấn chẳng hạn) có thể có sẵn cho tất cả các khách hàng và không cần thiết phải cài đặt toàn bộ các dịch vụ.
Nhược điểm	• Mỗi dịch vụ là một điểm đơn gây lỗi vì vậy dễ bị tấn công từ chối dịch vụ hoặc lỗi server. • Hiệu năng có thể không dự đoán trước được do nó phụ thuộc vào mạng cũng như hệ thống. • Có thể có các vấn đề về quản lý nếu server được sở hữu bởi các tổ chức khác nhau.

Kiến trúc client–server cho một thư viện phim ảnh



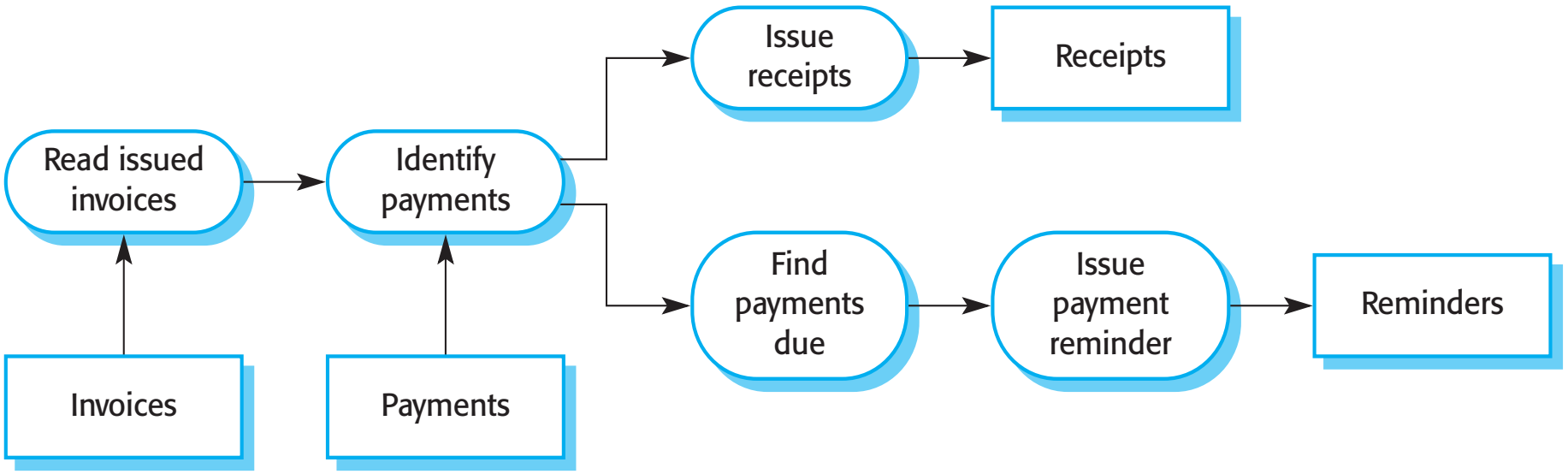
Kiến trúc pipe and filter

- Đây là mô hình tổ chức thời gian thực của hệ thống.
  - Các thao tác xử lý sẽ chuyển đổi các đầu vào và tạo ra các đầu ra.
- Các biến thể của phương pháp này rất phổ biến.
  - Khi các chuyển đổi là tuần tự, đây là mô hình xử lý khối tuần tự mà các hệ thống xử lý dữ liệu sử dụng.
- Không thật sự phù hợp với các hệ thống tương tác.

Mô hình pipe and filter

Tên	Mô hình pipe and filter
Mô tả	• Việc xử lý dữ liệu trong một hệ thống được tổ chức sao cho mỗi component xử lý (filter) là rời rạc và tiến hành một thao tác xử lý chuyển đổi dữ liệu. • Dòng dữ liệu (pipe) đi từ một component đến một component khác.
Sử dụng khi nào	• Trong các ứng dụng xử lý dữ liệu (cả ứng dụng xử lý khối và xử lý giao tác) trong đó các đầu vào được xử lý ở các giai đoạn rời rạc để tạo ra các đầu ra tương ứng.
Ưu điểm	• Dễ hiểu và hỗ trợ việc tái sử dụng các chuyển đổi. • Phù hợp với cấu trúc của của nhiều quy trình thương mại. • Thêm vào các chuyển đổi mới một cách dễ dàng. • Có thể cài đặt theo kiểu tuần tự hoặc song song.
Nhược điểm	• Format của dữ liệu truyền đi phải được chấp thuận trong việc giao tiếp giữa các chuyển đổi: Mỗi chuyển đổi phải phân tích cú pháp đầu vào của nó và chuyển nó thành đầu ra ở dạng được chấp nhận. → khó khăn trong việc tái sử dụng các hàm chuyển đổi khi cấu trúc dữ liệu không tương thích.

Ví dụ về kiến trúc pipe and filter



Nội dung

1. Quyết định chọn kiến trúc thiết kế
2. Các góc nhìn về kiến trúc
3. Các kiến trúc mẫu
4. Các kiến trúc ứng dụng

Các kiến trúc ứng dụng

- Các hệ thống ứng dụng được thiết kế để đáp ứng nhu cầu về công việc.
- Vì công việc trong một miền ứng dụng có nhiều điểm chung, các hệ thống ứng dụng cũng có xu hướng có kiến trúc chung phản ánh các yêu cầu ứng dụng.
- Kiến trúc ứng dụng tổng quát là một kiến trúc cho một loại hệ thống phần mềm được cấu hình và điều chỉnh để tạo ra một hệ thống đáp ứng các yêu cầu cụ thể.

Sử dụng các kiến trúc ứng dụng

- Như là điểm khởi đầu của thiết kế kiến trúc.
- Như là một checklist về thiết kế.
- Như một cách để tổ chức công việc của nhóm phát triển phần mềm.
- Như là một phương tiện để đánh giá việc tái sử dụng các component.
- Như là kho từ vựng để bàn về các loại ứng dụng.

Ví dụ về loại ứng dụng

- Ứng dụng xử lý dữ liệu
  - Các ứng dụng hướng dữ liệu trong đó xử lý dữ liệu khối mà không có sự can thiệp của người dùng trong suốt quá trình xử lý.
- Ứng dụng xử lý giao tác
  - Ứng dụng dữ liệu trung tâm trong đó xử lý các yêu cầu người dùng và cập nhật thông tin trong một cơ sở dữ liệu hệ thống.
- Hệ thống xử lý sự kiện
  - Ứng dụng trong đó các hoạt động của hệ thống phụ thuộc vào việc diễn giải các sự kiện từ môi trường hệ thống.
- Hệ thống xử lý ngôn ngữ
  - Ứng dụng trong đó ý định của người dùng được đặc tả trong các ngôn ngữ hình thức và được xử lý, diễn giải bởi hệ thống.

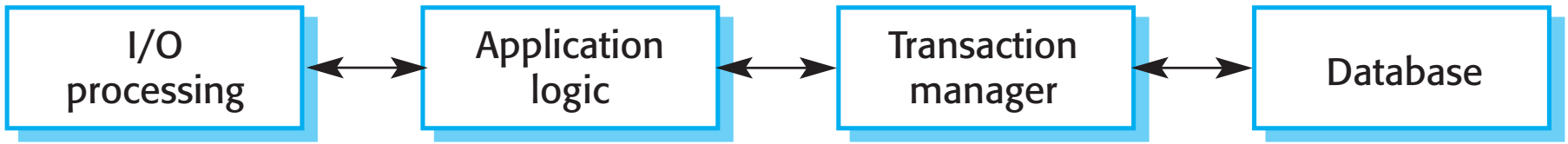
Ví dụ về loại ứng dụng

- Tập trung vào hệ thống xử lý giao tác và xử lý ngôn ngữ.
- Hệ thống xử lý giao tác
  - Hệ thống thương mại điện tử;
  - Hệ thống đặt chỗ.
- Hệ thống xử lý ngôn ngữ
  - Trình biên dịch;
  - Thông dịch lệnh.

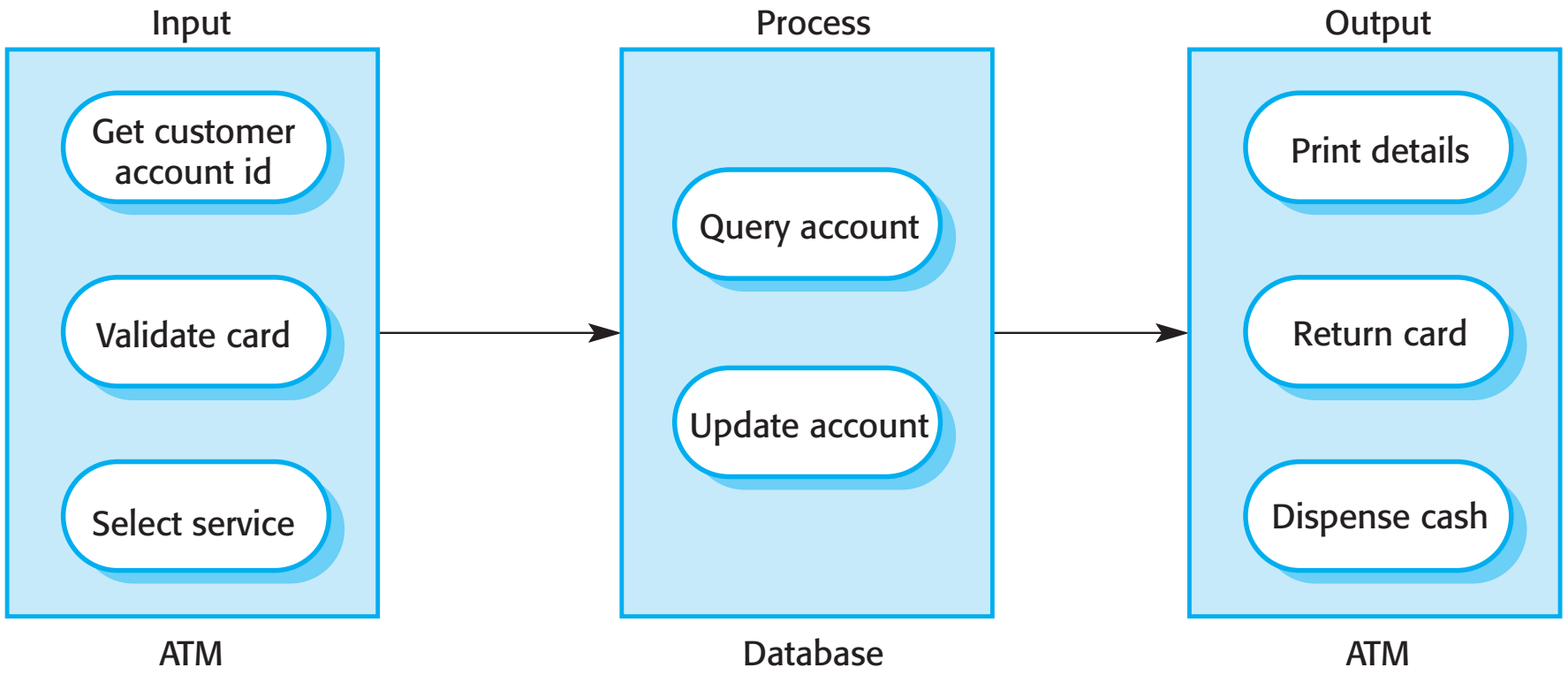
Hệ thống xử lý giao tác

- Xử lý các yêu cầu người dùng về thông tin từ một CSDL hoặc yêu cầu cập nhật CSDL.
- Từ góc độ của người sử dụng, một giao tác là:
  - Một chuỗi liên tục các thao tác để thỏa mãn một mục tiêu;
  - Ví dụ: tìm thời gian của các chuyến bay từ London tới Paris.
- Người dùng thực hiện các yêu cầu không đồng bộ về dịch vụ sau đó được xử lý bởi một bộ quản lý giao tác.

Cấu trúc của ứng dụng xử lý giao tác



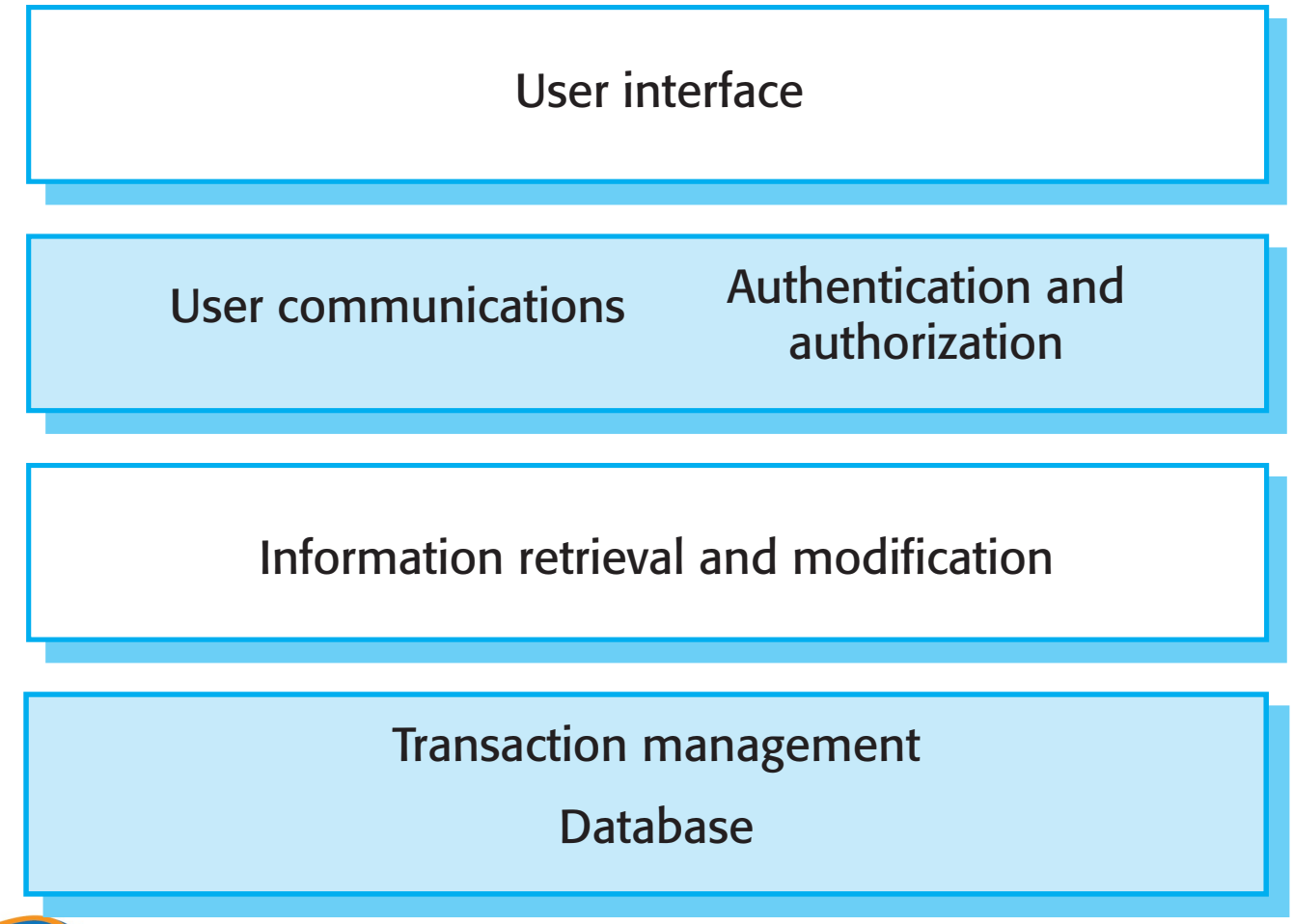
Kiến trúc của hệ thống ATM



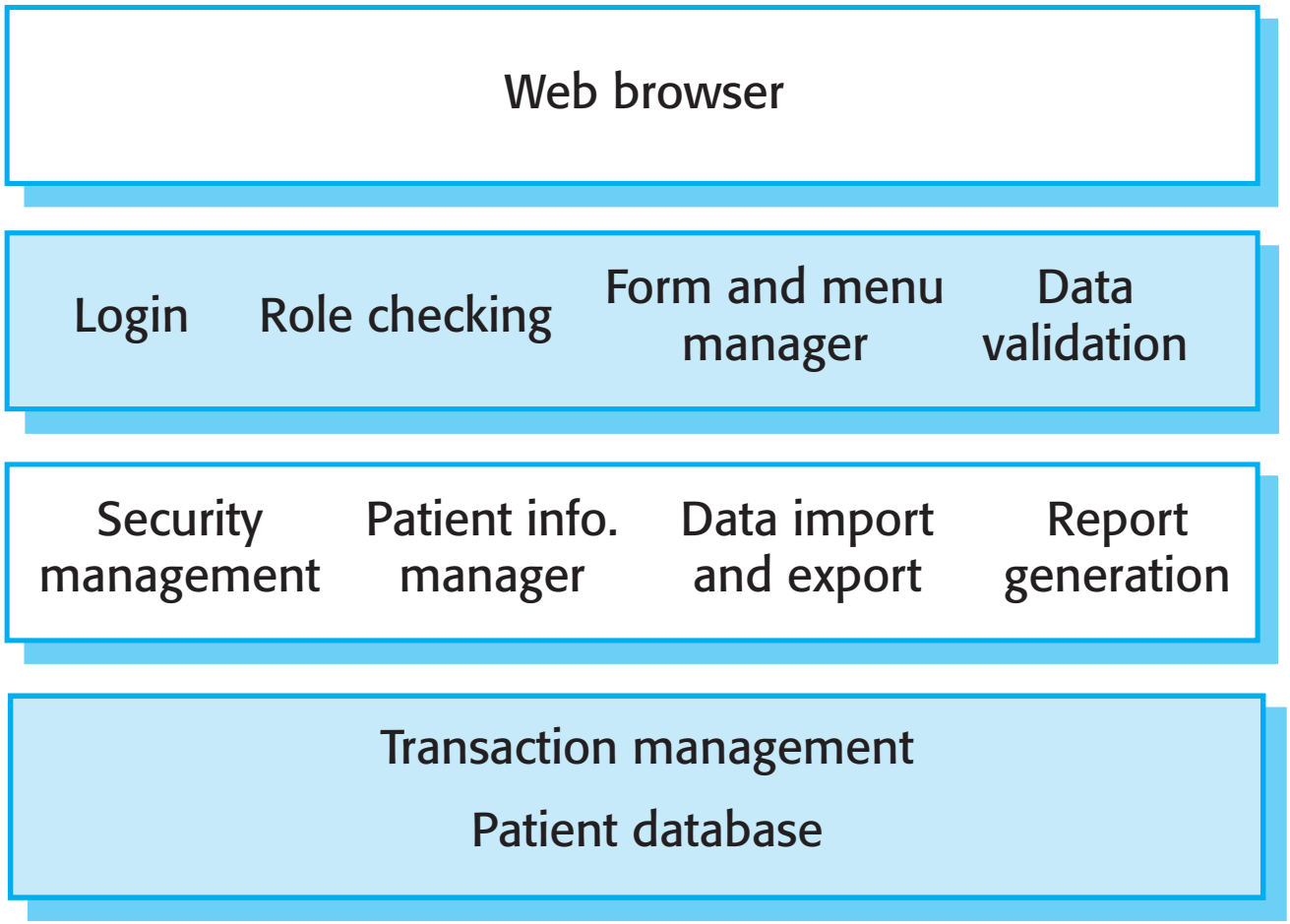
Kiến trúc hệ thống thông tin

- Các hệ thống thông tin có cấu trúc tổng quát được tổ chức theo kiểu cấu trúc phân tầng.
- Đây là những hệ thống dựa vào giao tác
  - vì tương tác với một CSDL được chia sẻ.
- Các tầng bao gồm:
  - The user interface
  - User communications
  - Information retrieval
  - System database

Kiến trúc phân tầng của hệ thống thông tin



Kiến trúc của hệ thống MHC-PMS



Các hệ thống thông tin dựa vào web

- Hệ thống quản lý tài nguyên và thông tin thường là các hệ thống dựa vào web
  - Giao diện người dùng được cài đặt trên web browser.
- Ví dụ: các hệ thống thương mại điện tử là các hệ thống quản trị tài nguyên dựa vào internet
  - Các đơn đặt hàng điện tử về hàng hóa hay dịch vụ được chấp nhận, các đơn hàng này sau đó được lên lịch giao cho khách hàng.
- Trong hệ thống thương mại điện tử, tầng ứng dụng chứa các tính năng bổ sung hỗ trợ "giỏ hàng" trong đó người dùng có thể đặt nhiều món hàng trên các giao tác khác nhau sau đó tính tiền chung trong một giao tác đơn.

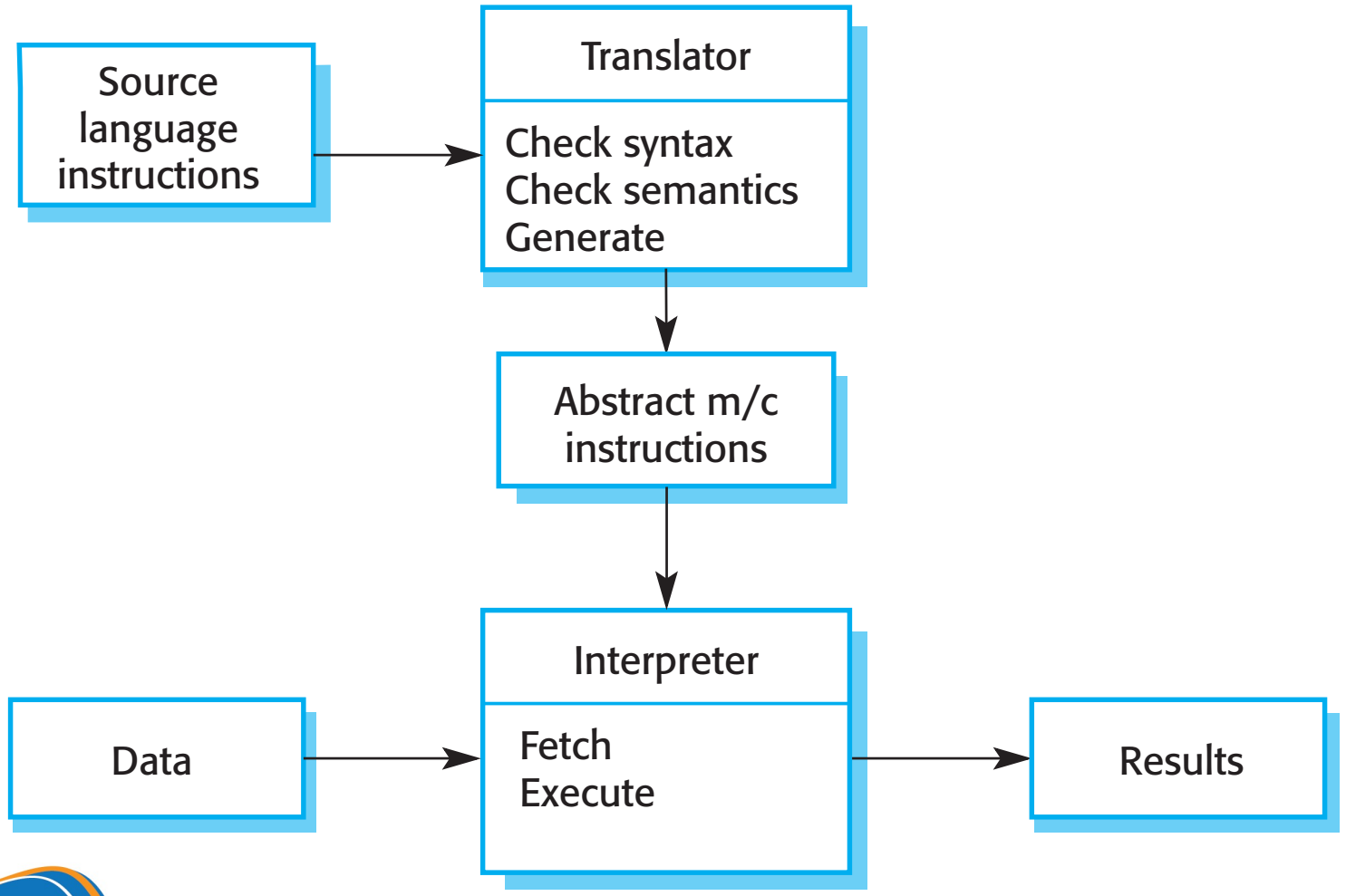
Cài đặt phía server

- Những hệ thống này thường được cài đặt theo kiểu kiến trúc đa tầng client-server
  - Web server chịu trách nhiệm giao tiếp với người dùng, giao diện người dùng được cài đặt sử dụng web browser;
  - Server ứng dụng chịu trách nhiệm cài đặt các chức năng ứng dụng cụ thể cũng như lưu trữ thông tin và truy vấn yêu cầu;
  - Server cơ sở dữ liệu chuyển thông tin từ và đến cơ sở dữ liệu và nắm quyền quản lý giao tác.

Hệ thống xử lý ngôn ngữ

- Đầu vào là ngôn ngữ tự nhiên hoặc ngôn ngữ nhân tạo và đầu ra là một dạng biểu diễn khác của ngôn ngữ đó.
- Có thể chứa bộ diễn giải để thực hiện các chỉ dẫn lệnh trong ngôn ngữ được xử lý.
- Ví dụ:
  - Trình biên dịch chuyển chương trình (C, Java, ...) thành mã máy.
  - Hệ thống dịch các mô tả CML thành lệnh để truy vấn cơ sở dữ liệu hoặc thành dạng biểu diễn XML khác.
  - Hệ thống từ điển

Kiến trúc của một hệ thống xử lý ngôn ngữ



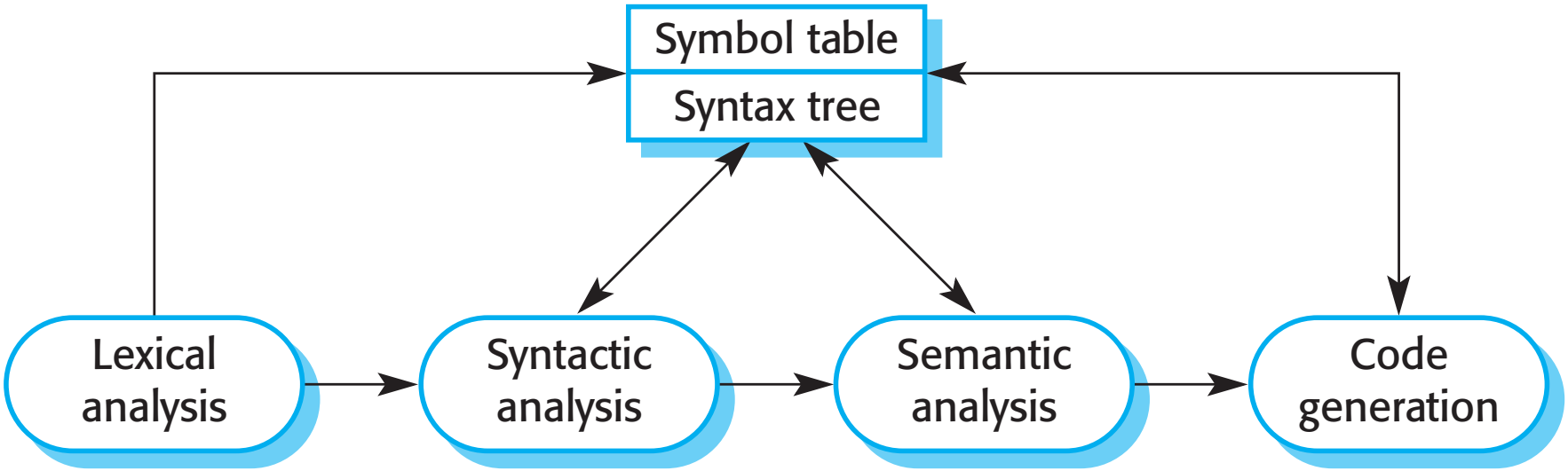
Các component của trình biên dịch

- Bộ phân tích từ vựng
  - Lấy các token đầu vào, chuyển đổi chúng thành một hình thức trung gian.
- Bảng ký hiệu
  - Lưu giữ thông tin về tên thực thể (biến, tên lớp, tên đối tượng, ...) được sử dụng trong văn bản cần biên dịch.
- Bộ phân tích cú pháp
  - Kiểm tra cú pháp của ngôn ngữ cần biên dịch.
- Cây cú pháp
  - Là một biểu diễn cấu trúc bên trong mà chương trình được dịch.

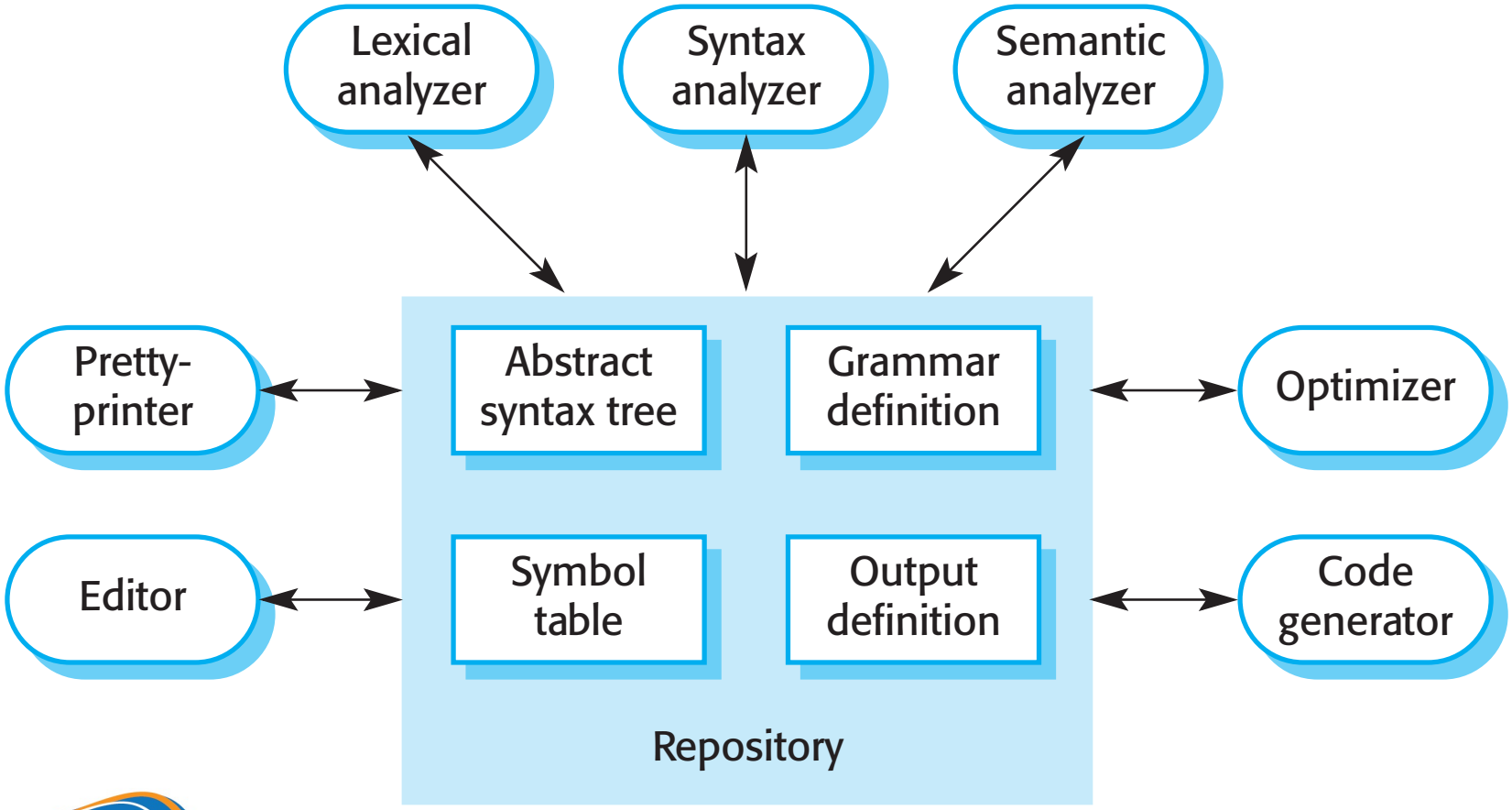
Các component của trình biên dịch

- Bộ phân tích ngữ nghĩa:
  - Sử dụng thông tin từ cây cú pháp và bảng ký hiệu để kiểm tra tính đúng đắn về ngữ nghĩa của văn bản ngôn ngữ đầu vào.
- Bộ phát sinh mã :
  - Duyệt qua cây cú pháp và sinh ra mã máy trừu tượng.

Kiến trúc pipe and filter của trình biên dịch



Kiến trúc repository cho hệ thống xử lý ngôn ngữ



Câu hỏi?

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN