

# Công nghệ phần mềm

## **Thiết kế kiến trúc**

# Nội dung

---

- Định nghĩa kiến trúc phần và một số khái niệm
- Thiết kế kiến trúc
- Một số kiểu kiến trúc

# Nội dung

---

- Định nghĩa kiến trúc phần mềm và Một số khái niệm
- Thiết kế kiến trúc
- Một số kiểu kiến trúc

# Định nghĩa KTPM

---

- Có nhiều trường phái
- Richard N. Taylor và đồng nghiệp
  - “*A software system’s architecture is **the set of principal design decisions** about the system*”
  - Kiến trúc phần mềm là bản thiết kế cho việc xây dựng và tiến hóa phần mềm
  - Các quyết định thiết kế bao gồm nhiều khía cạnh của hệ thống đang được phát triển
    - Cấu trúc
    - Hành vi
    - Tương tác
    - Thuộc tính chất lượng

# Định nghĩa KTPM

---

- Len Bass và đồng nghiệp

*“The software architecture of a system is the set of **structures** needed to reason about the system, which comprise **software elements**, **relations** among them, and **properties** of both”*

- Những định nghĩa khác:

<http://www.sei.cmu.edu/architecture/definitions.html>

- Chúng ta sẽ sử dụng định nghĩa của Len Bass và đồng nghiệp

# Sự quan trọng của KTPM

---

- Hỗ trợ cho giao tiếp giữa các bên liên quan (stakeholders)
- Xác định các ràng buộc cho việc hiện thực hóa
- Dự đoán chất lượng hệ thống
- Nâng cao độ chính xác của việc dự đoán chi phí và thời gian xây dựng hệ thống

# Một số khái niệm

---

- Cấu trúc (structure): tập các phần tử (element) và cách tổ chức
- 3 loại cấu trúc
  - Cấu trúc mô đun (module structures)
  - Cấu trúc thành phần – kết nối (component-and-connector structures)
  - Cấu trúc phân phối (allocation structures)
- Mỗi loại cấu trúc thể hiện một khía cạnh khác nhau của hệ thống
- Liên quan đến nhau

# Một số khái niệm

---

- Mẫu kiến trúc (architectural patterns): kiến trúc được sử dụng để giải quyết một/vài vấn đề khi xây dựng hệ thống và đã được sử dụng ở nhiều hệ thống
- Ví dụ: phân lớp (layered pattern), chia sẻ dữ liệu (shared-data), khách-chủ (client-server)



# Một số khái niệm

---

- Thuộc tính chất lượng
  - Tính linh hoạt: Chúng ta có thể thay đổi thành phần X không?
  - Tính khả chuyển: Chúng ta có thể triển khai trên một máy khác không?
  - Tính sử dụng lại: Chúng ta có thể sử dụng lại một phần hay toàn bộ cho ứng dụng khác không?

# Kiến trúc và thuộc tính chất lượng

---

- Để có hiệu năng cao
  - Phân rã thành các tiến trình chạy song song
  - Quản lý lượng và tần suất dữ liệu truyền nhận giữa các tiến trình
  - Xác định “nút cổ chai” về hiệu năng trong hệ thống
- Để có an ninh tốt
  - Phân chia chức năng cần và không cần login
  - Cấu trúc phần mềm thành nhiều tầng

# Các thuộc tính chất lượng

---

- Thuộc tính thiết kế (Design qualities)
- Thuộc tính thời gian thực thi (Run-time qualities)
- Thuộc tính hệ thống (System qualities)
- Thuộc tính người dùng (User qualities)

*Nguồn: <https://msdn.microsoft.com/en-us/library/ee658094.aspx>*

# Nội dung

---

- Định nghĩa kiến trúc phần mềm và một số khái niệm
- **Thiết kế kiến trúc**
- Một số kiểu kiến trúc

# Các nguyên lý thiết kế

---

- Phân tách các khía cạnh quan tâm (Separation of concerns): chia ứng dụng thành các phần càng ít sự chồng chéo về chức năng càng tốt. Cố gắng hạn chế tối đa sự tương tác giữa các thành phần nhằm có giảm sự phụ thuộc và tăng cường sự kết dính (cohesion) trong từng thành phần
- Trách nhiệm đơn: Mỗi thành phần chỉ thực hiện một chức năng hoặc một tập các chức năng gắn kết chặt chẽ
- Hiểu biết tối thiểu: Các thành phần không cần biết chi tiết bên trong của các thành phần khác

# Các nguyên lý thiết kế

---

- Không lặp lại: Mỗi một chức năng chỉ được hiện thực hóa bởi một thành phần
- Hạn chế thiết kế trước: chỉ thiết kế khi cần và có đủ thông tin

# Các mối quan tâm chính

---

Đây là những quyết định quan trọng khi thiết kế

- Kiểu ứng dụng
- Chiến lược triển khai
- Các công nghệ phù hợp
- Các thuộc tính chất lượng
- Một số các yếu tố “cắt ngang” (crosscutting concerns)

# Thiết kế kiến trúc

---

- Đầu vào: yêu cầu chức năng và phi chức năng, các ràng buộc
- Đầu ra: bản thiết kế kiến trúc
- Là quá trình lặp với 5 bước chính

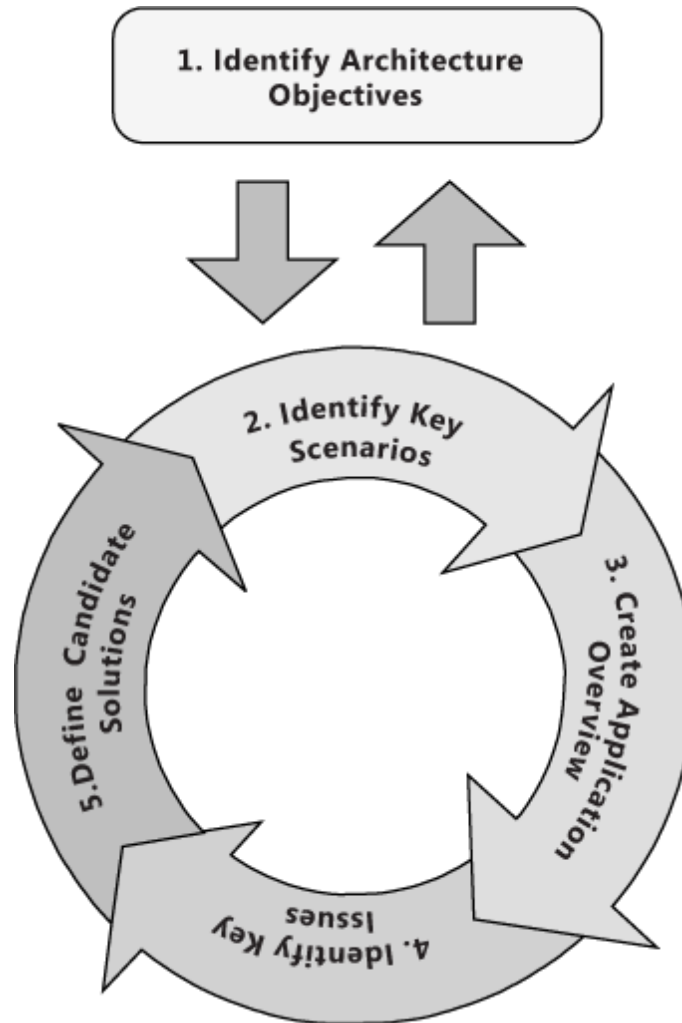


# Các bước chính

---

1. Xác định mục tiêu
2. Xác định các hoạt cảnh sử dụng chính
3. Xác định tổng quan về ứng dụng
4. Xác định các vấn đề chính
5. Xác định các giải pháp chính

# Các bước chính



# Xác định mục tiêu

---

- Thiết kế kiến trúc để làm gì
- Cho ai?
- Các ràng buộc là gì?



Phạm vi và thời gian thực hiện

# Hoạt cảnh sử dụng chính

---

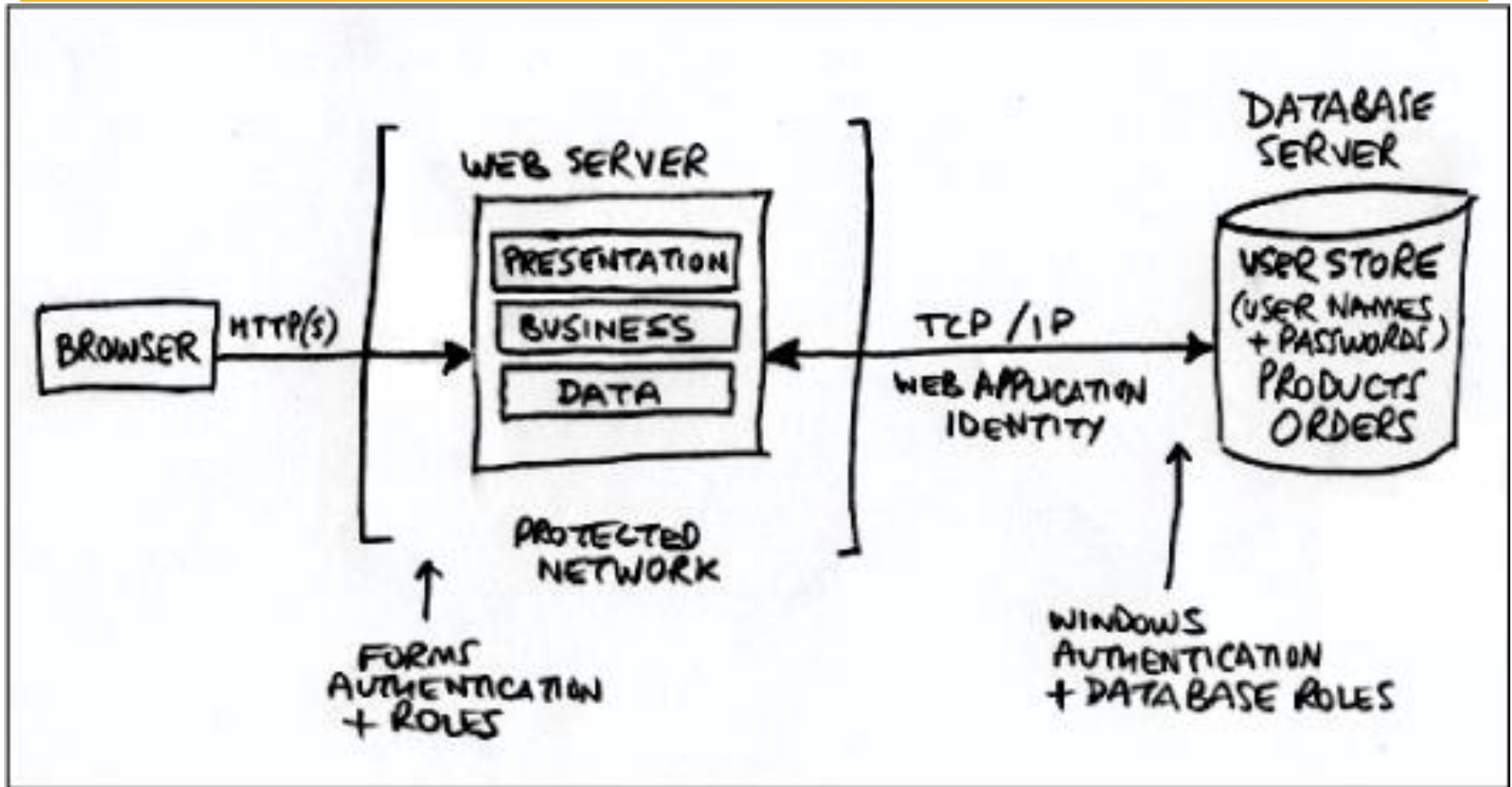
- Một hoạt cảnh là sự tổng quát của nhiều ca sử dụng (use case) tương tự
- Các hoạt cảnh chính
  - Có ảnh hưởng lớn, được sử dụng nhiều
  - Thể hiện sự “đánh đổi” giữa các thuộc tính chất lượng

# Tổng quan về ứng dụng

---

- Xác định kiểu ứng dụng
- Xác định các ràng buộc triển khai
- Các kiểu kiến trúc có thể sử dụng
- Các công nghệ liên quan

# Phác thảo kiến trúc

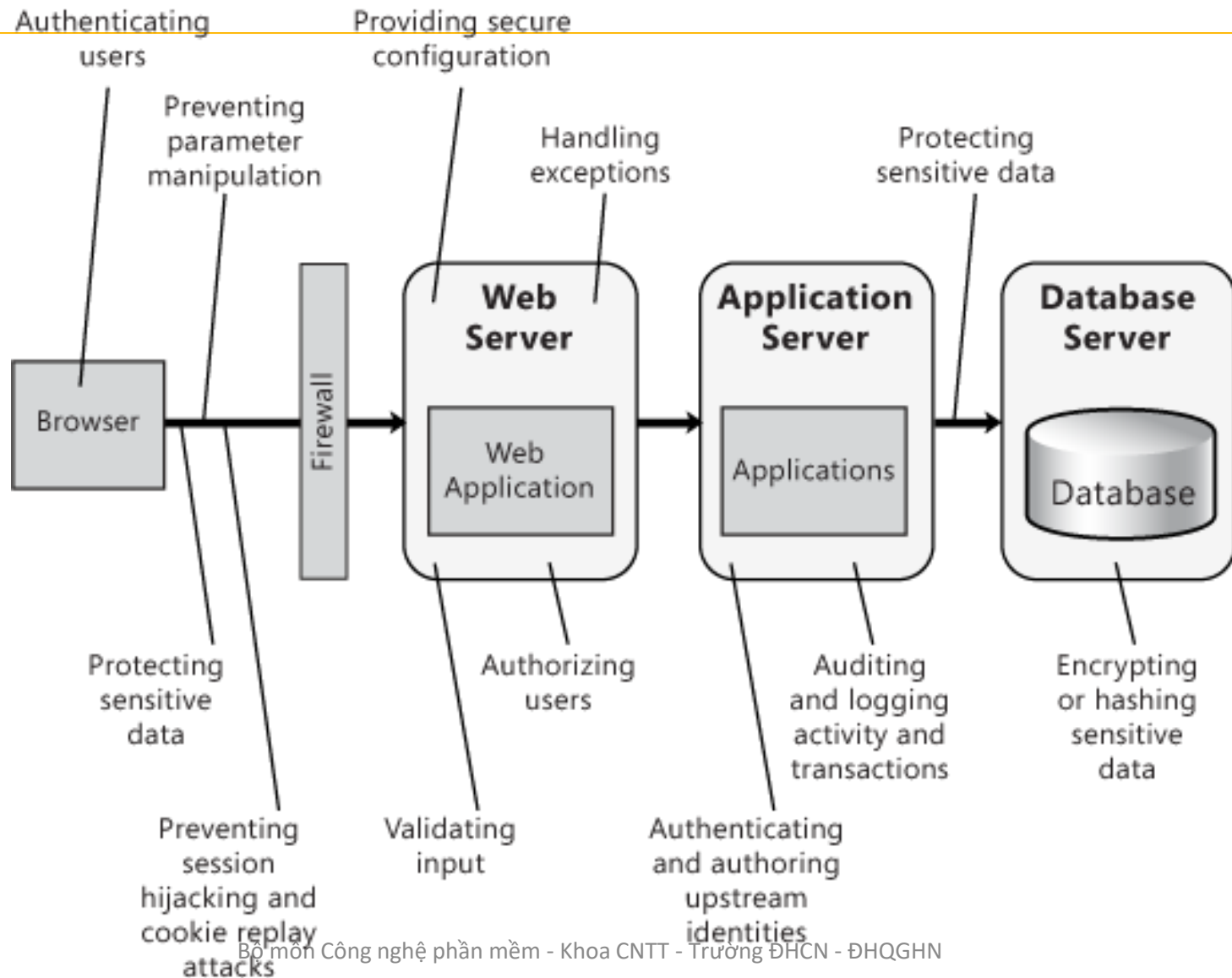


# Xác định các vấn đề chính

---

- Thuộc tính chất lượng: performance, extensibility,...
- Khía cạnh quan tâm chung: authentication, authorization, caching, exception handling,...

# Xác định các vấn đề chính





# Giải pháp dự kiến

---

- Là bản thiết kế kiến trúc ở mức cao
- Bao gồm cả kiểu ứng dụng, kiến trúc triển khai, kiểu kiến trúc, công nghệ được lựa chọn, thuộc tính chất lượng, các khía cạnh khác
- Kiểm tra các yêu cầu, ràng buộc

# Nội dung

---

- Định nghĩa kiến trúc phần mềm
- Một số khái niệm
- Thiết kế kiến trúc
- **Một số kiểu kiến trúc**

# Kiểu Khách/Chủ

---

- Cho hệ thống phân tán với hai thành phần riêng biệt Khách (client) và Chủ (server)
- Cấu hình 1-1, 1-n, n-1
- Trước đây, Client thường là ứng dụng chạy trên desktop; hiện nay có thêm Web browser và ứng dụng di động

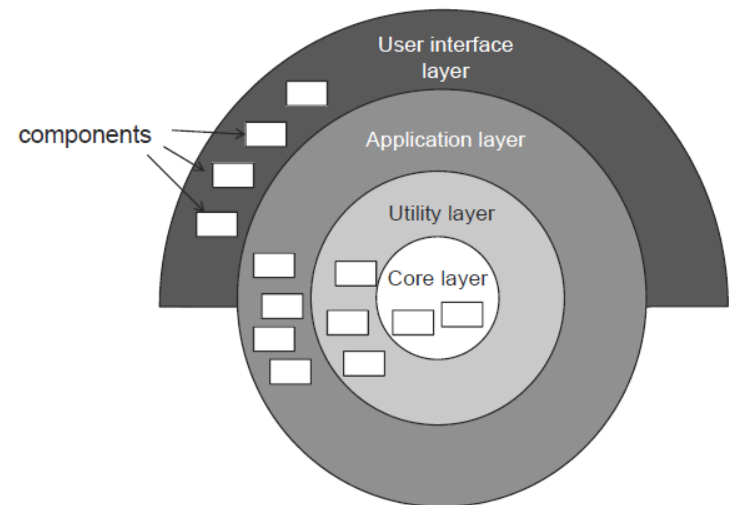
# Kiểu Khách/Chủ

---

- Một số biến thể
  - Client-Queue-Client: cho phép các clients có thể giao tiếp thông qua hàng đợi. Server chỉ đóng vai trò là hàng đợi để chứa dữ liệu. Còn được gọi là kiến trúc hàng đợi bị động
  - Peer-to-Peer: Client và server có thể đổi vai trò cho nhau
  - Application server: một dạng khách/chủ đặc biệt với phía chủ chứa các ứng dụng và dịch vụ cho các ứng dụng khách dạng Web hoặc dạng khác

# Kiểu phân lớp

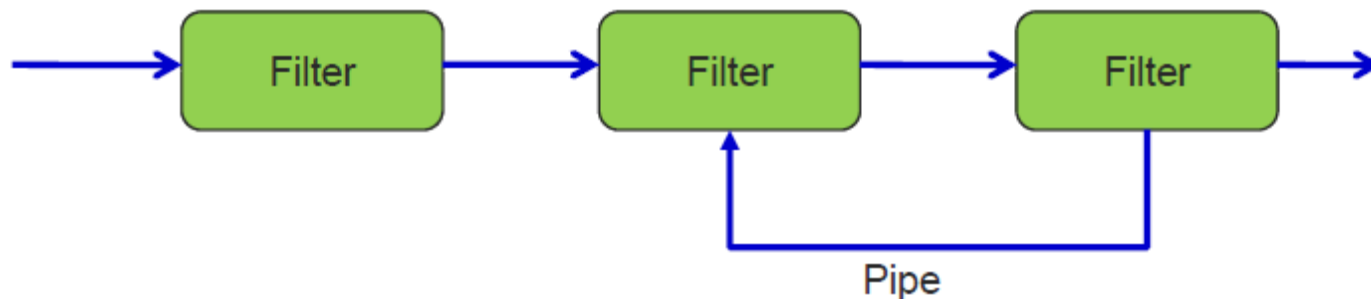
- Nhóm các chức năng có liên quan chặt chẽ vào chung một lớp
- Các lớp được sắp xếp chồng lên nhau
- Chỉ lớp trên mới sử dụng chức năng của lớp dưới



# Kiểu ống dẫn và bộ lọc

---

- Pipe and Filter
- Thể hiện tương tác trong thời gian chạy
- Dữ liệu truyền qua các ống dẫn để đến và được xử lý ở bộ lọc



# Kiểu thành phần

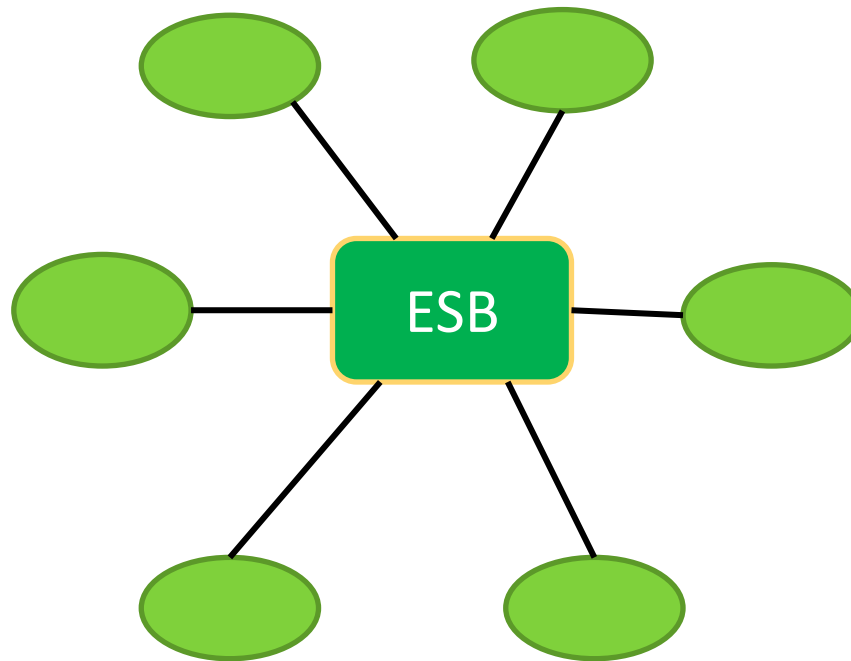
---

- Hệ thống được chia thành các thành phần độc lập, tương tác với nhau thông qua các giao diện
- Các thành phần có thể thực thi phân tán

# Kiểu bus thông điệp

---

Tương tác giữa các thành phần trong hệ thống được thực hiện bằng cách truyền thông điệp trên một bus chung





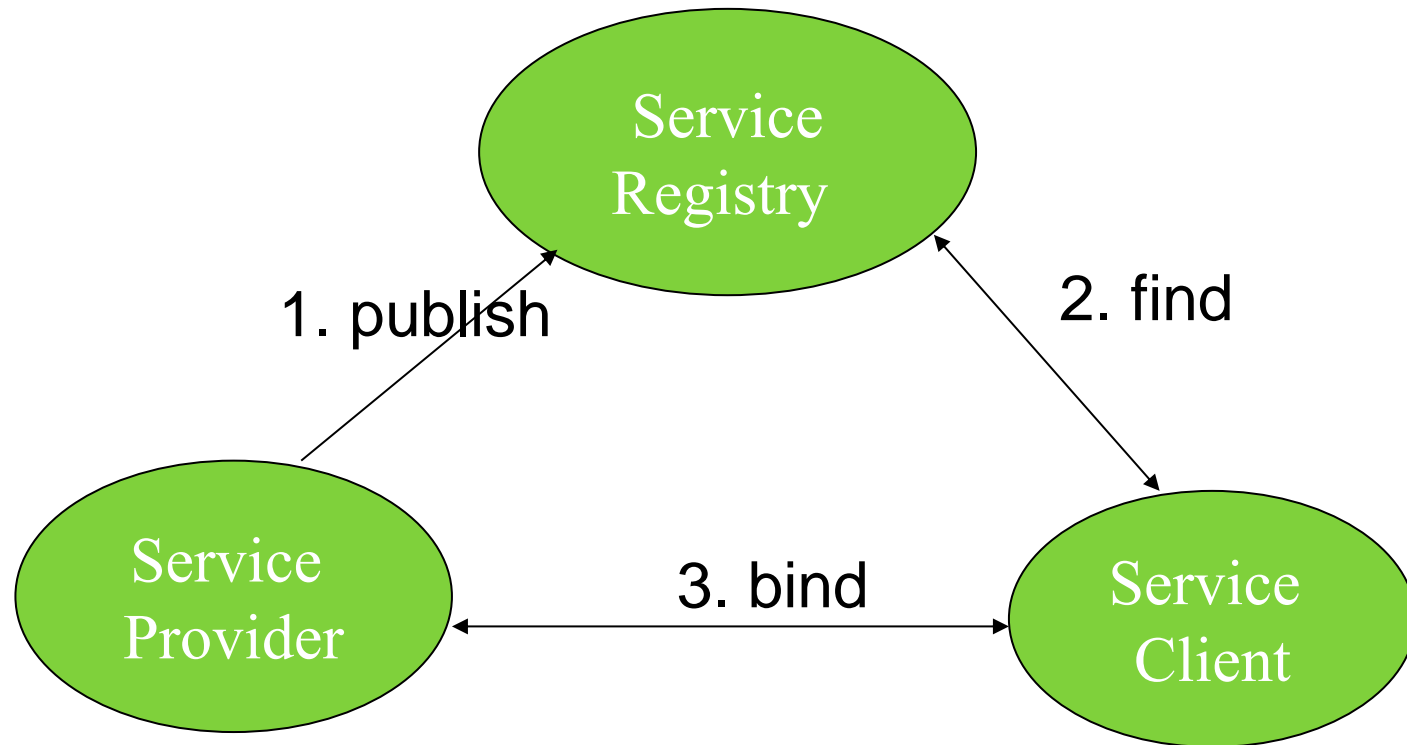
# Kiểu hướng dịch vụ

---

- Chức năng của các ứng dụng được cung cấp dưới dạng dịch vụ
- Các ứng dụng có thể được xây dựng dựa trên các dịch vụ (sử dụng các dịch vụ có trước)
- Các dịch vụ có kết nối mềm dẻo, giao diện được chuẩn hóa, có thể được xuất bản (published), tìm kiếm (discovered), gọi (invoked)

# Kiểu hướng dịch vụ

---



# Tổng kết

---

- Định nghĩa kiến trúc phần mềm
- Một số khái niệm
- Thiết kế kiến trúc
- Một số kiểu kiến trúc

# Hỏi - Đáp

# Tài liệu tham khảo

---

- Len Bass et al., Software architecture in practice, 3<sup>rd</sup> Edition
- Microsoft, Microsoft Application Architecture Guide, 2<sup>nd</sup> Edition