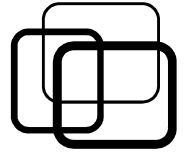
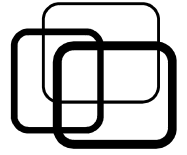


Kiến trúc phần mềm

GV. Nguyễn Minh Huy

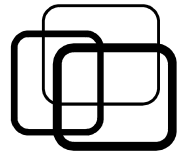


- Các khái niệm cơ bản.
- Các mô hình kiến trúc.
- Công nghệ phân tán.



- **Các khái niệm cơ bản.**
- Các mô hình kiến trúc.
- Công nghệ phân tán.

Các khái niệm cơ bản

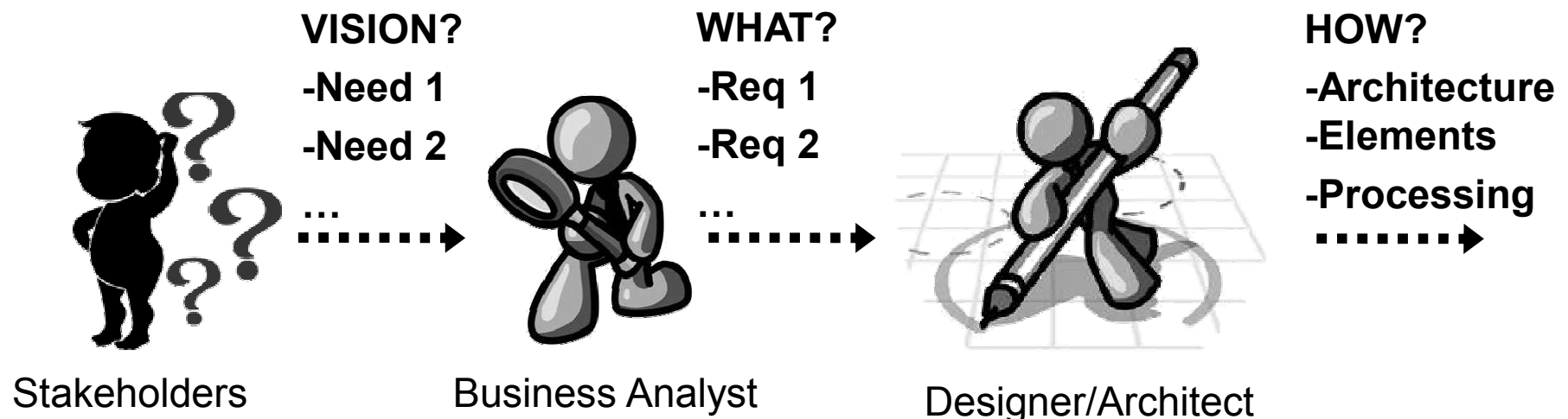


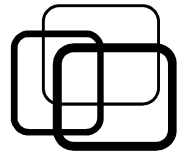
■ Thiết kế phần mềm là gì?

■ Trả lời câu hỏi **HOW**.

■ Phác thảo cách hiện thực hóa yêu cầu.

- Thiết kế kiến trúc: sub-systems, architecture model.
- Thiết kế thành phần: màn hình, đối tượng, dữ liệu.
- Thiết kế xử lý: hiện thực hóa kịch bản Use Case.





■ Kiến trúc phần mềm là gì?

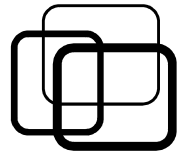
- Chương trình đơn giản → một thành phần.
- Chương trình phức tạp → nhiều thành phần con.
- Những câu hỏi về thành phần con:

- Được tổ chức, sắp xếp thế nào?
- Tương tác ra sao?
- Cấu tạo bên trong?

■ Kiến trúc phần mềm

→ Bản mô tả cấu tạo, tương tác bên trong của phần mềm.





■ Tầm quan trọng của kiến trúc:

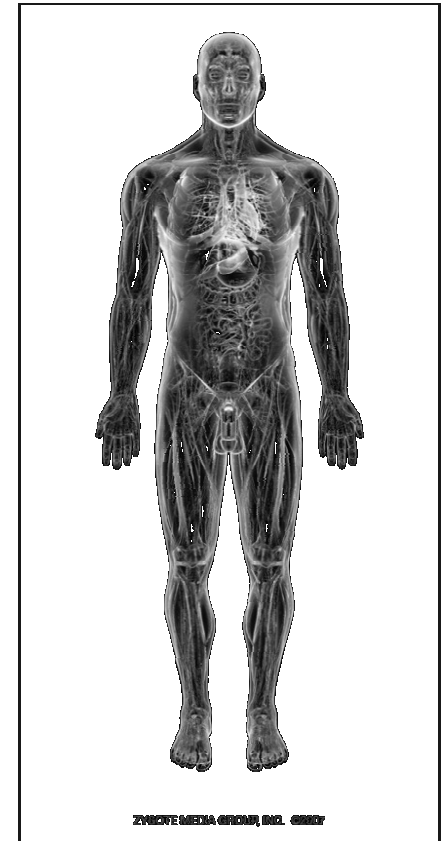
■ Ảnh hưởng hiệu quả hoạt động:

- Tốc độ xử lý (performance).
- Tính ổn định (reliability).
- Tính bảo mật (security).
- Tính chịu lỗi (fault-tolerance).

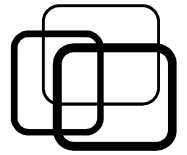
■ Ảnh hưởng chi phí:

- Khả năng triển khai.
- Khả năng vận hành.
- Khả năng bảo trì.

■ Ảnh hưởng thiết kế và cài đặt.

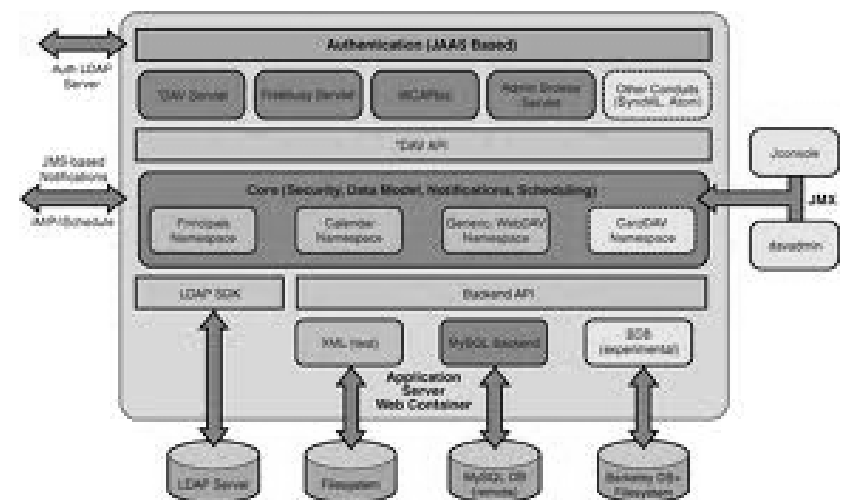


Các khái niệm cơ bản

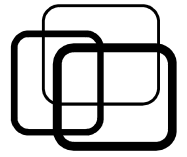


■ Thiết kế kiến trúc:

- Hoạt động đầu tiên của pha thiết kế.
- Xác định khung sườn phần mềm.
- Các bước thực hiện:
 - Bổ dọc: xác định sub-systems.
 - Bổ ngang: lựa chọn architecture model.



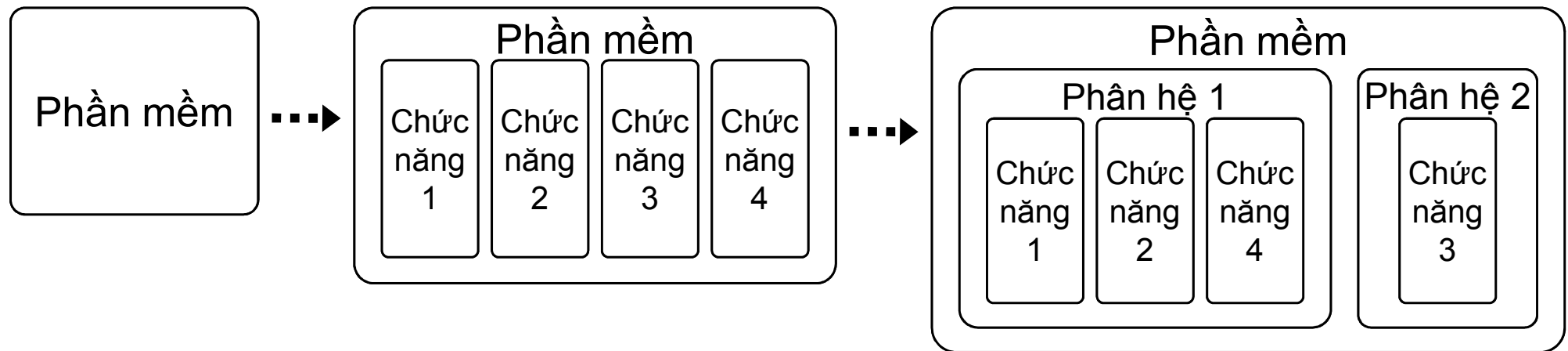
Các khái niệm cơ bản



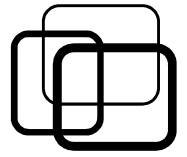
■ “Bổ dọc” phần mềm:

■ Chia phần mềm thành các sub-system.

- Hệ thống con nằm trong phần mềm.
- Phân hệ của phần mềm.
- Nhóm các chức năng liên quan với nhau.



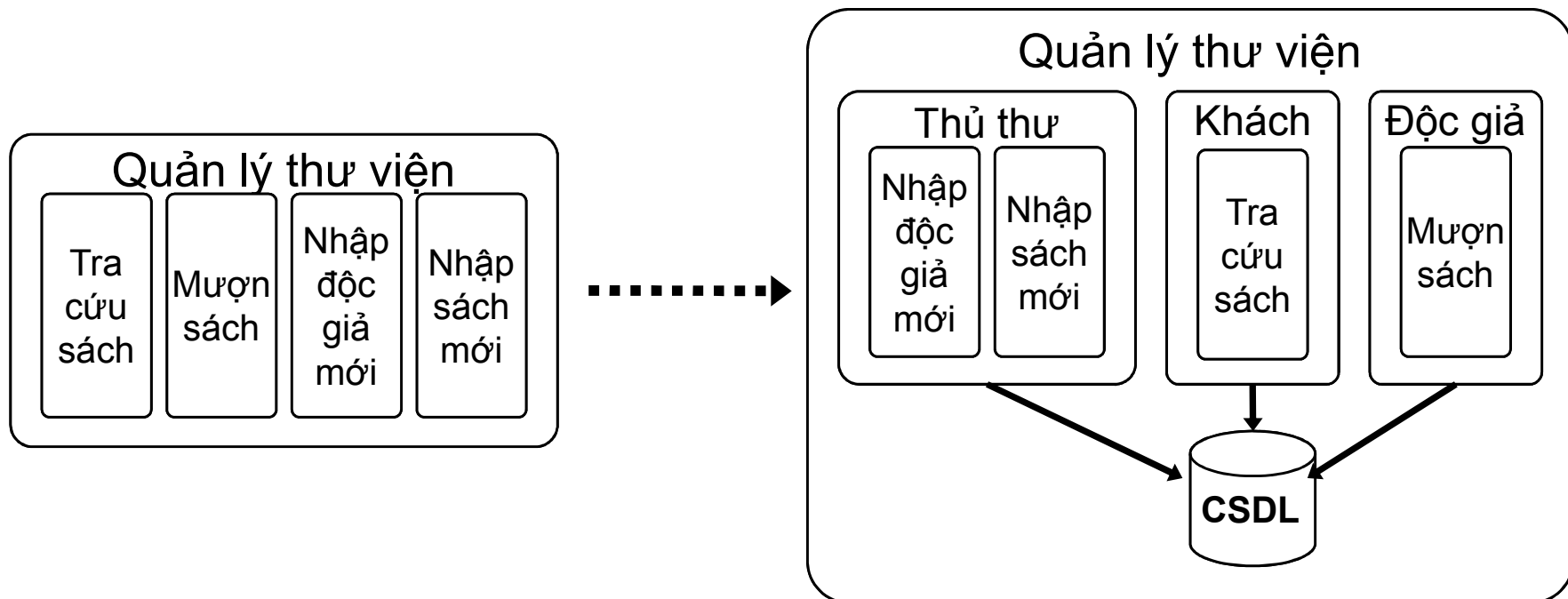
Các khái niệm cơ bản



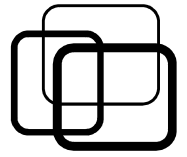
■ “Bổ dọc” phần mềm:

■ Tiêu chí phân chia sub-system:

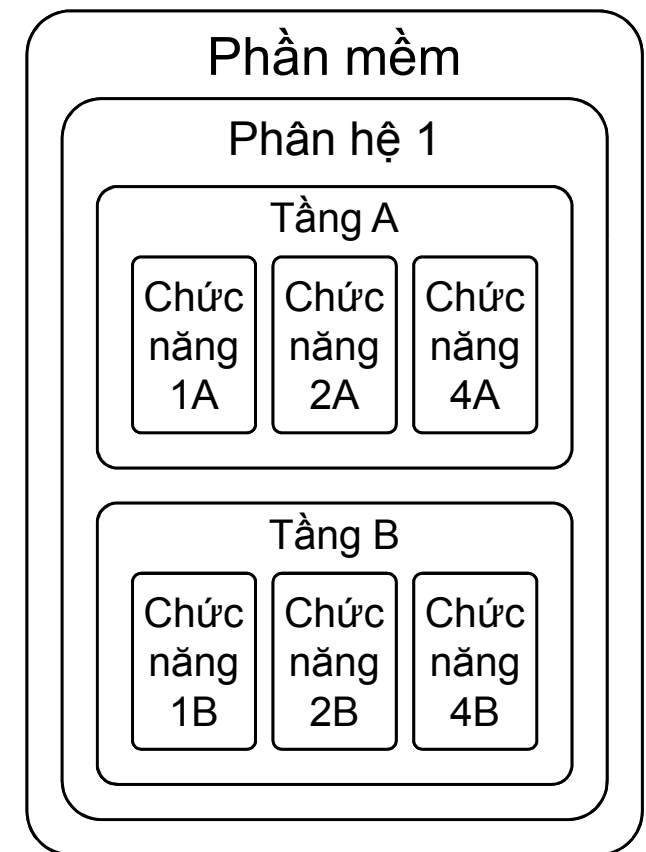
- Bảo mật người dùng.
- Chức năng dùng chung.

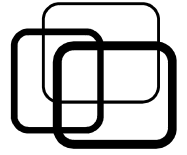


Các khái niệm cơ bản

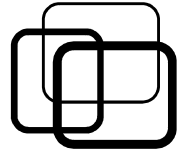


- “Bổ ngang” phần mềm:
 - Phân tầng xử lý các chức năng.
 - Dựa theo các mô hình kiến trúc.
- Các mô hình kiến trúc:
 - Mô hình đơn lập.
 - Mô hình phân tầng:
 - Mô hình 2-Tầng (Client-Server).
 - Mô hình 3-Tầng.
 - Mô hình Peer-To-Peer.





- Các khái niệm cơ bản.
- **Các mô hình kiến trúc.**
- Công nghệ phân tán.

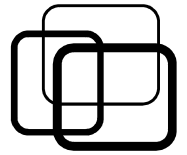


■ Mô hình đơn lập:

- Không phân tầng xử lý.
- Các thành phần tự do tương tác.
- Ưu điểm:
 - Dễ lập trình và triển khai.
 - Tốc độ xử lý.
- Nhược điểm:
 - Khó bảo trì, nâng cấp.
 - Khó chia sẻ dữ liệu.



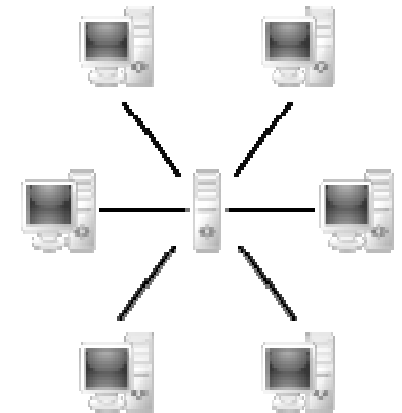
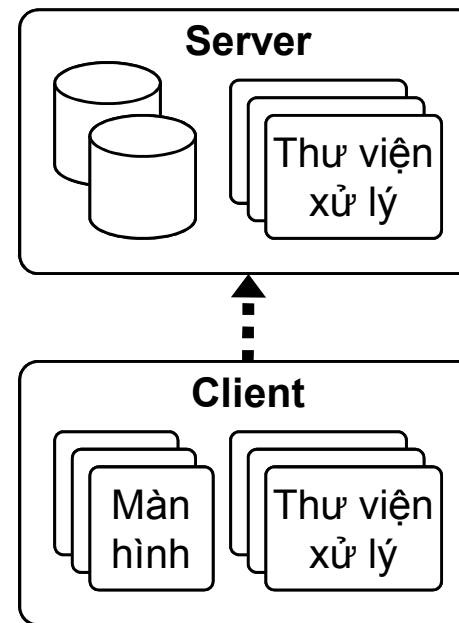
Các mô hình kiến trúc



■ Mô hình Client-Server:

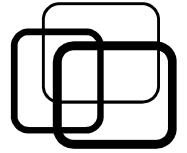
■ Phân làm hai tầng:

- Server:
 - Cung cấp dịch vụ.
 - ➔ Dữ liệu, thư viện.
 - Chia sẻ, dùng chung.
- Client:
 - Sử dụng dịch vụ.
 - ➔ Giao diện, thư viện.
 - Phân tán.



■ Tương tác giữa các thành phần:

- Trong phân hệ: tự do tương tác.
- Giữa hai phân hệ: tương tác 1 chiều từ client đến server.



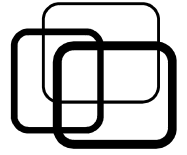
■ Mô hình Client-Server:

■ Mô hình Thin-Client:

- Server đảm trách dữ liệu + xử lý.
- Client chỉ lo giao diện người dùng.
→ Dump terminal.

■ Mô hình Fat-Client:

- Server đảm trách dữ liệu.
- Client lo toàn bộ xử lý + giao diện người dùng.



■ Mô hình Client-Server:

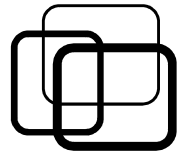
■ Ưu điểm:

- Dữ liệu chia sẻ và đồng bộ.
- Hạn chế tương tác
 - ➔ Cô lập lỗi client-server.
 - ➔ Dễ bảo trì, nâng cấp.

■ Nhược điểm:

- Chi phí triển khai.
- Tốc độ xử lý.

Các mô hình kiến trúc



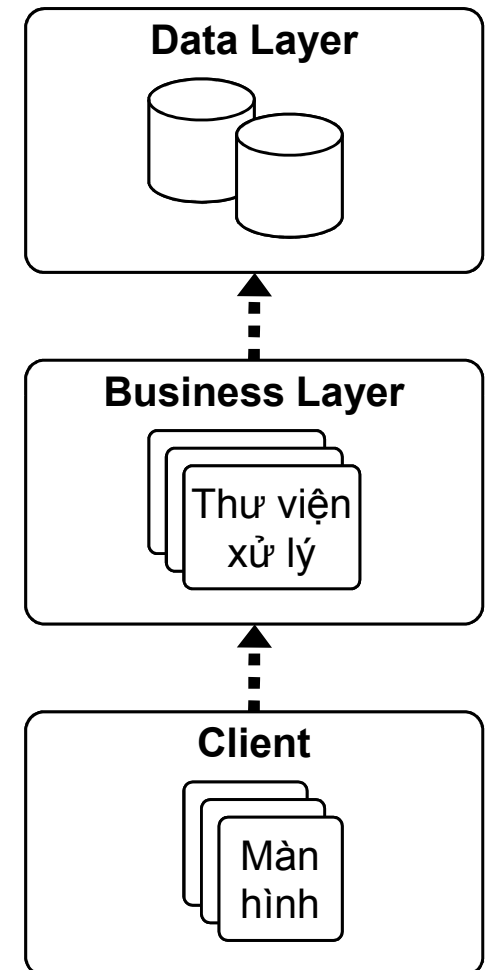
■ Mô hình 3-Tầng (3-Tiers):

■ Phân làm ba tầng:

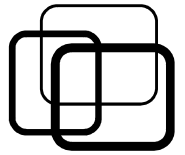
- Tầng dữ liệu (data layer).
 - Dịch vụ dữ liệu.
 - Data server.
- Tầng xử lý (business layer).
 - Thư viện xử lý.
 - Application server.
- Tầng giao diện (presentation layer).
 - Giao diện người dùng.
 - Thin-Client.

■ Quy tắc tương tác “thang máy”.

➔ Không tương tác vượt tầng.



Các mô hình kiến trúc



■ Mô hình 3-Tầng (3-Tiers):

■ Mô hình đa tầng:

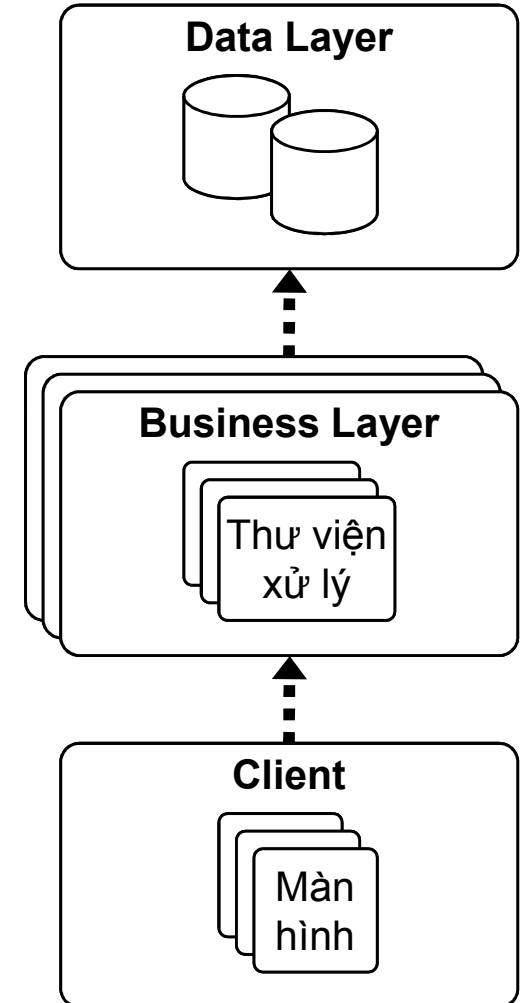
- Mở rộng mô hình 3-Tầng.
- Phân làm nhiều tầng xử lý.
- Dùng trong ứng dụng web.

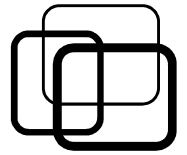
■ Ưu điểm:

- Tương tự mô hình Client-Server.
- Tầng xử lý riêng biệt.

■ Nhược điểm:

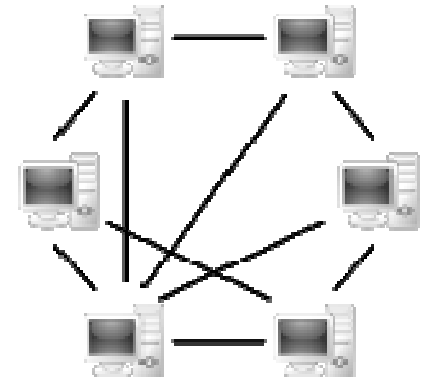
- Tương tự mô hình Client-Server.

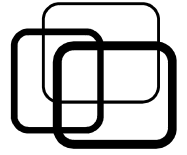




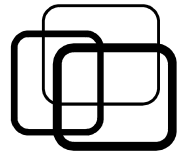
■ Mô hình Peer-to-Peer:

- Là mô hình đơn lập phân tán.
- Triển khai trên nhiều máy (nút).
 - Các nút tương tác được với nhau.
 - Mỗi nút đóng vai Client-Server.
 - Chia sẻ dữ liệu + xử lý trên toàn bộ nút.
- Ưu điểm:
 - Không cần server trung tâm.
 - Không gian lưu trữ và khả năng xử lý dàn trải.
 - Dễ triển khai.
- Nhược điểm:
 - Khó lập trình và quản lý dữ liệu.



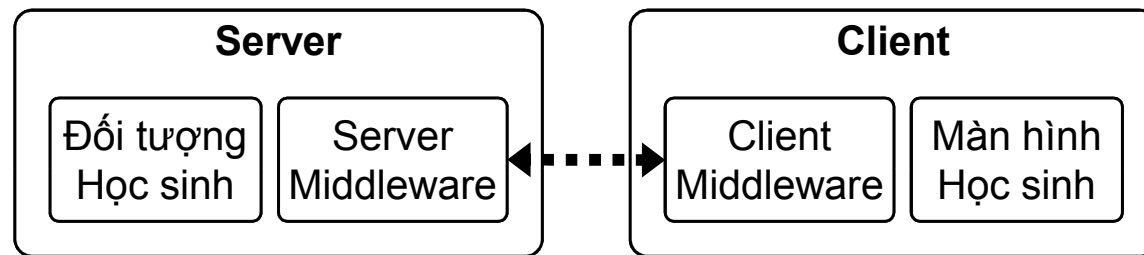


- Các khái niệm cơ bản.
- Các mô hình kiến trúc.
- **Công nghệ phân tán.**



■ Middleware:

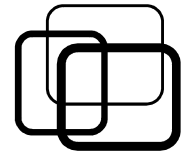
- Các thành phần trong hệ phân tán giao tiếp thế nào?
➔ Hệ thống đứng giữa điều phối.



■ Các chuẩn phổ biến:

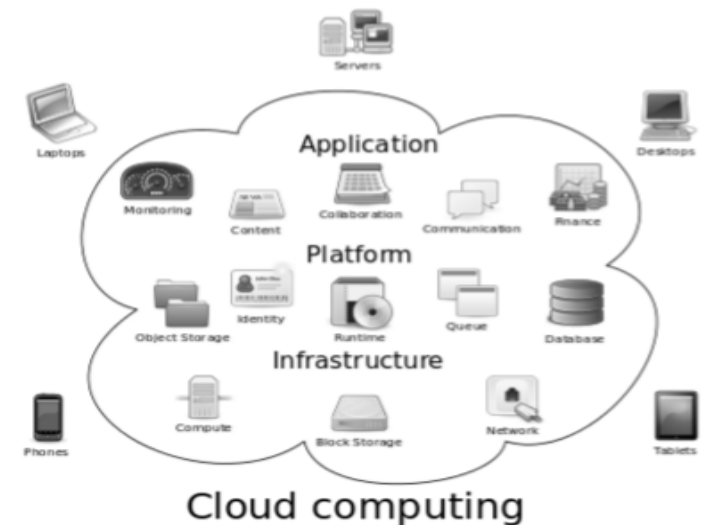
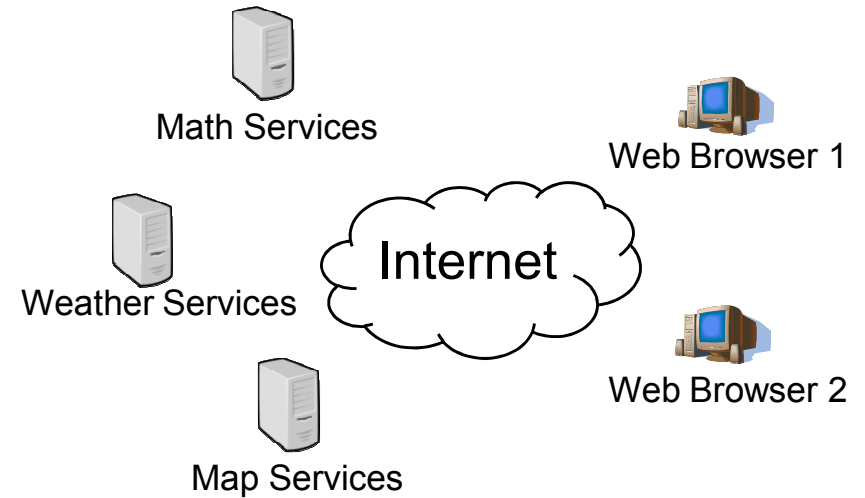
- CORBA (**C**ommon **O**bject **R**equest **B**roker **A**rchitecture).
- COM (**C**omponent **O**bject **M**odel).
- JavaBeans.

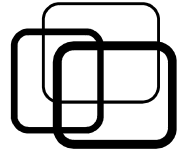
Công nghệ phân tán



■ Web Service:

- Thư viện lập trình online.
- Cung cấp dạng dịch vụ.
- Truy xuất qua internet.
- Các dịch vụ phổ biến:
 - Math services.
 - Google map.
 - Amazon services.
- Cloud computing.





■ Bài tập thiết kế kiến trúc:

■ Hệ thống “Online Bookstore”.

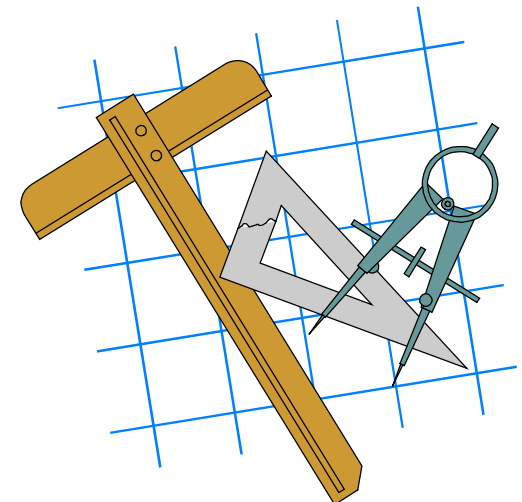
■ Công việc:

➤ Bổ dọc:

- Xác định stakeholders & yêu cầu tổng quan.
- Gom nhóm các yêu cầu để tăng bảo mật và tốc độ xử lý.
- Vẽ các phân hệ.

➤ Bổ ngang:

- Lựa chọn mô hình kiến trúc.
- Sử dụng ít nhất 1 web service.
- Vẽ kiến trúc hệ thống theo mô hình lựa chọn.



Ví dụ Kiến trúc phần mềm

