

# Kỹ thuật phần mềm ứng dụng

---

## *Chương 2*

### *Các pha trong phát triển phần mềm (Phần 7)*

#### *Pha thiết kế (thiết kế kiến trúc)*

# Kiến trúc phần mềm

---

- *Khái niệm:*

*“Là một cấu trúc bao gồm các thành phần phần mềm, các tính chất có thể thấy được từ bên ngoài của các thành phần này, và các liên kết giữa chúng” \**

- *Các thành phần phần mềm có thể gồm:*

- *Các module*
- *Các cấu trúc dữ liệu, cơ sở dữ liệu*

# Kiến trúc phần mềm

---

- *Mục đích sử dụng:*
  - *Để đánh giá tính hiệu quả của phần mềm trong việc đáp ứng các y k của hệ thống*
  - *Cân nhắc để chọn ra kiến trúc phù hợp nhất giữa các kiến trúc khác nhau*
  - *Giúp dự trù sớm và tương đối chính xác các tài nguyên cần chuẩn bị cho giai đoạn cài đặt phần mềm*
  - *Đóng vai trò như thiết kế tổng thể, làm nền tảng cho các thiết kế chi tiết sau đó*
  - *Giúp giảm thiểu các rủi ro trong quá trình xây dựng phần mềm sau này*

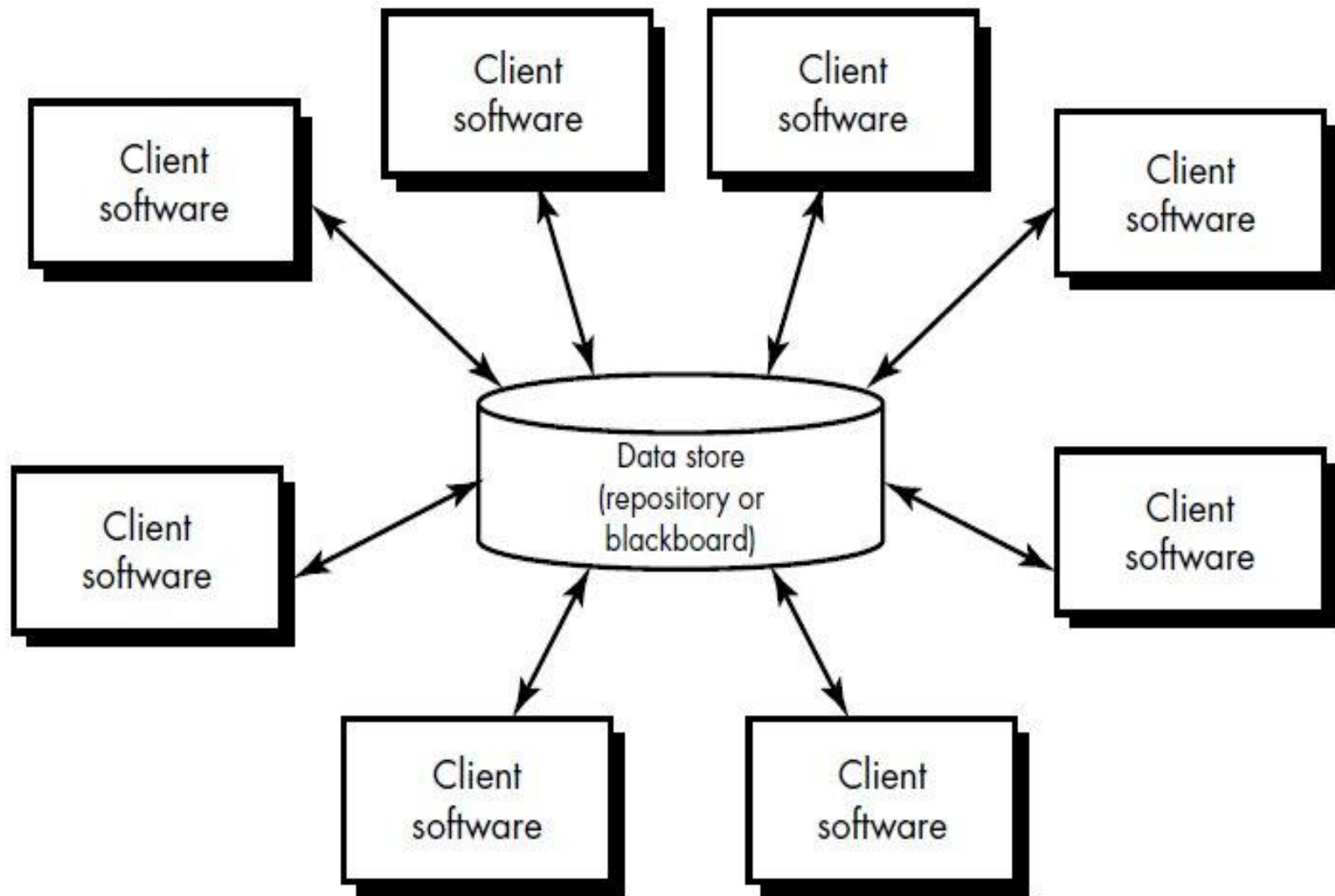
# Kiến trúc phần mềm

---

- *Các phong cách kiến trúc*
  - *Kiến trúc lấy dữ liệu làm trung tâm (data-centered architectures)*
  - *Kiến trúc luồng dữ liệu (data flow architectures)*
  - *Kiến trúc gọi và trả về (call and return architectures)*
    - *Main module / sub module*
    - *Remote procedure call*

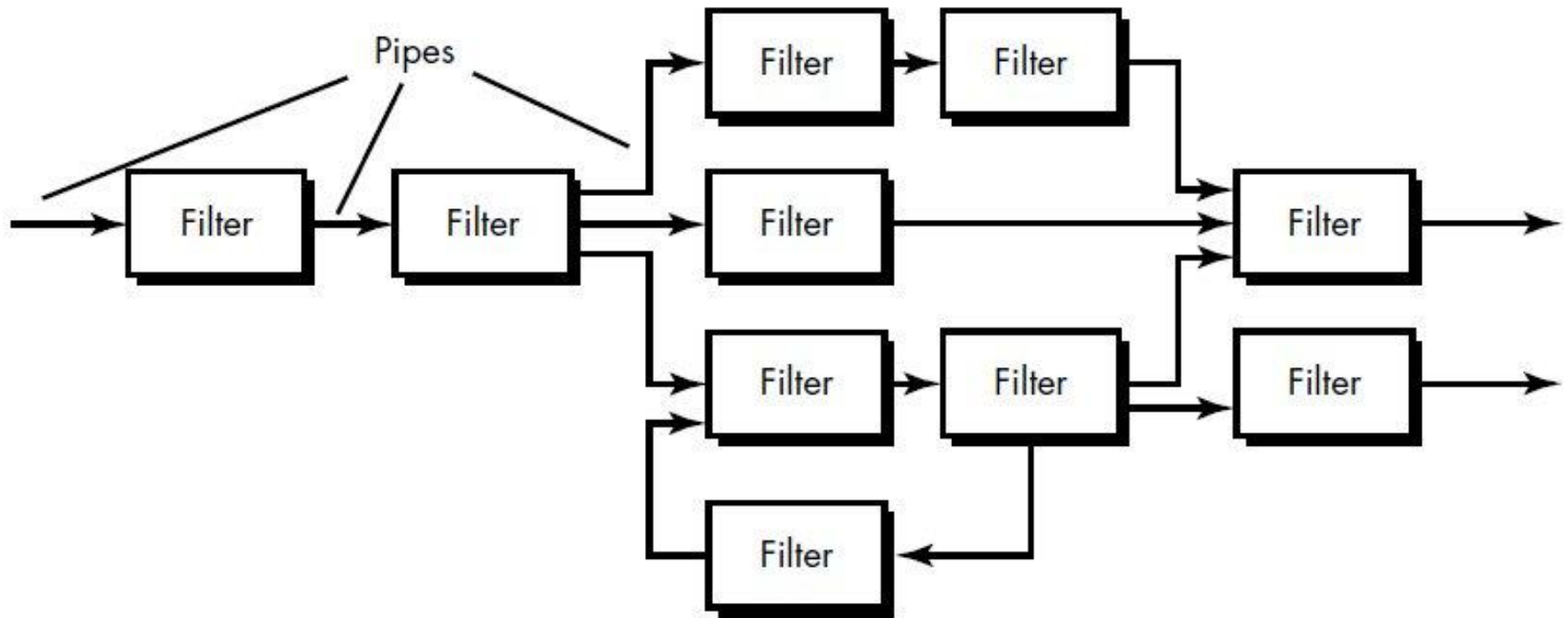
# Kiến trúc lấy dữ liệu làm trung tâm

---

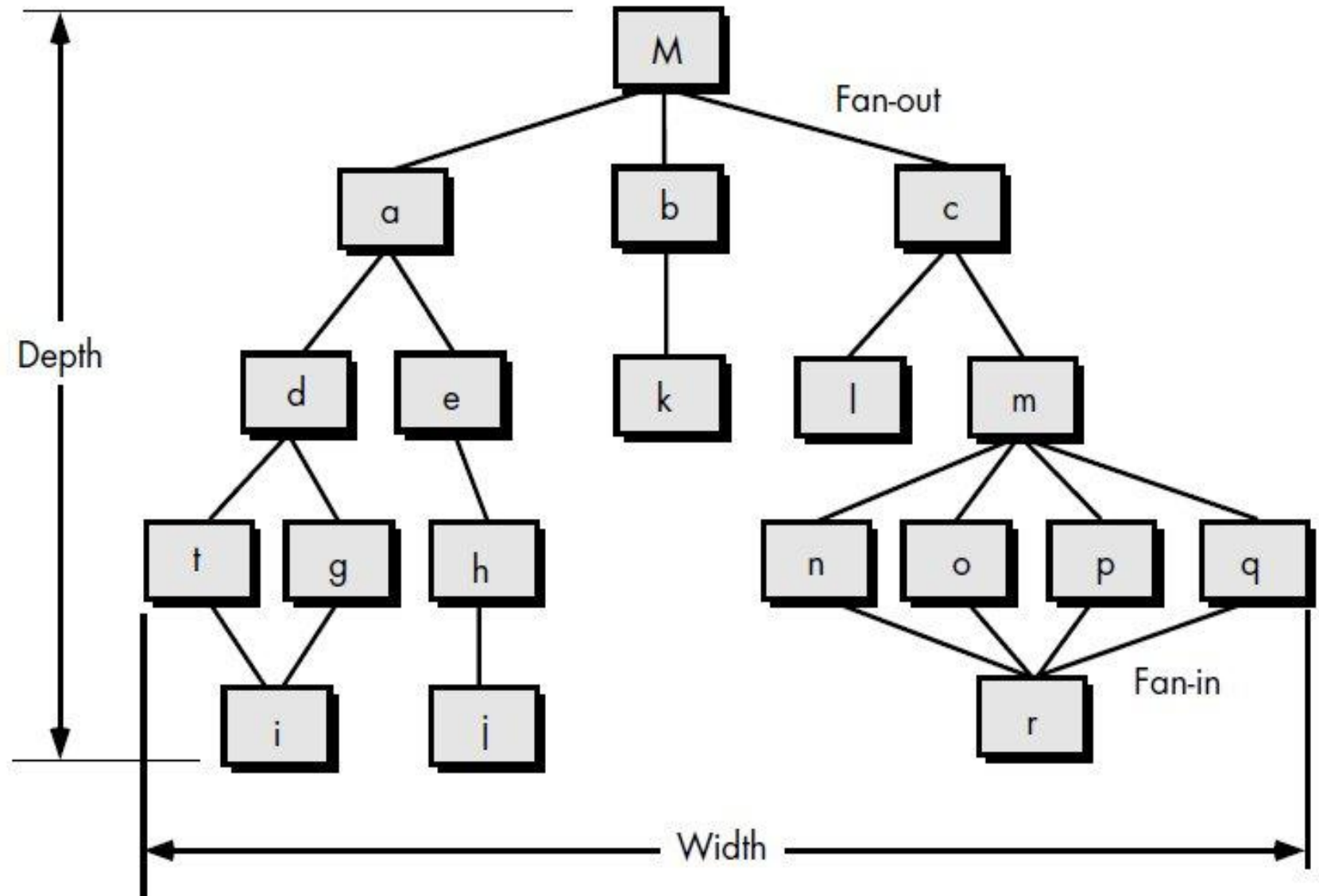


# Kiến trúc luồng dữ liệu

---



# Kiến trúc gọi và trả về



# Kiến trúc gọi và trả về

---

- *Kiến trúc này còn được gọi là “cấu trúc chương trình”, hay “phân cấp điều khiển”*
- *Một số khái niệm liên quan:*
  - ***Fan-in***: của một module là độ đo số lượng module khác mà điều khiển gọi module đó
  - ***Fan-out***: của một module là độ đo số lượng module mà module đó điều khiển gọi
  - ***Chiều sâu*** (*depth*): xác định số mức điều khiển gọi
  - ***Độ rộng*** (*width*): xác định phạm vi điều khiển gọi



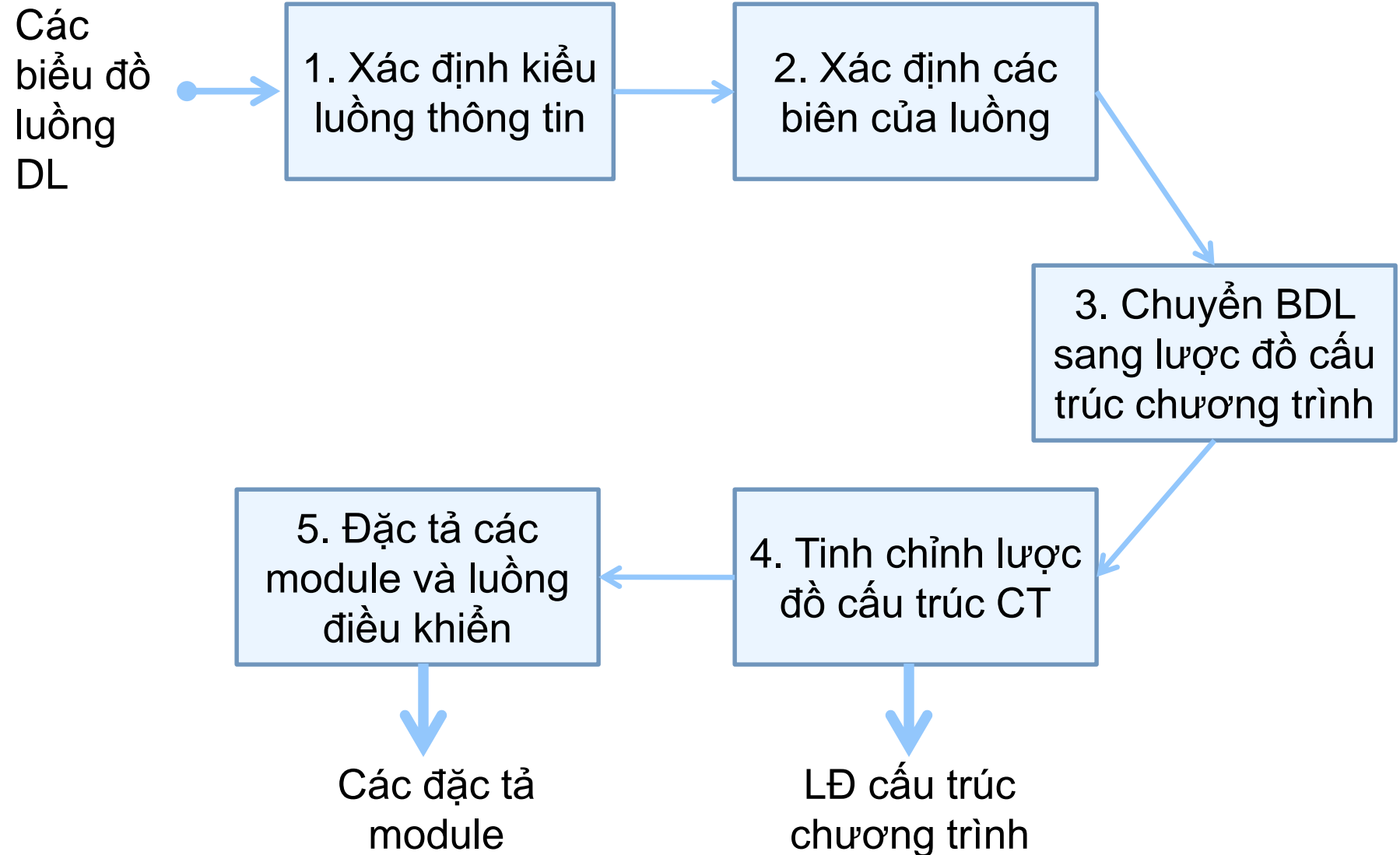
# Thiết kế kiến trúc phần mềm

---

- *Phương pháp được sử dụng:*  
***“Thiết kế có cấu trúc”*** (*structured design*)
- *Đặc điểm của phương pháp:*  
*Có hướng luồng dữ liệu, cung cấp cách thuận tiện để chuyển từ các biểu đồ luồng dữ liệu sang mô hình kiến trúc phần mềm*

# Thiết kế có cấu trúc

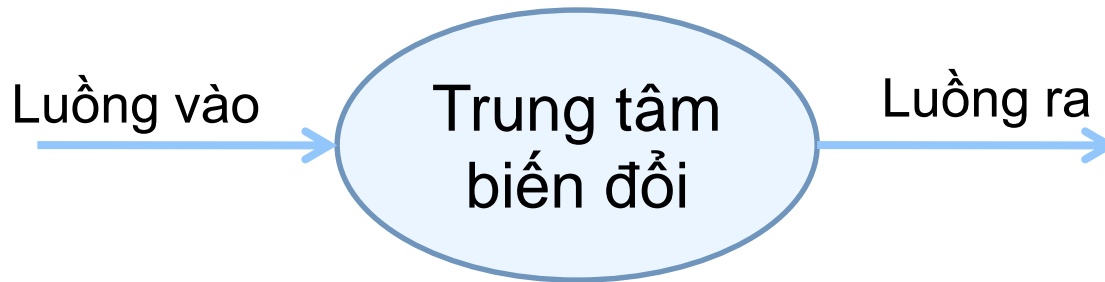
---



# Các kiểu luồng thông tin

---

- *Luồng biến đổi (transform flow)*

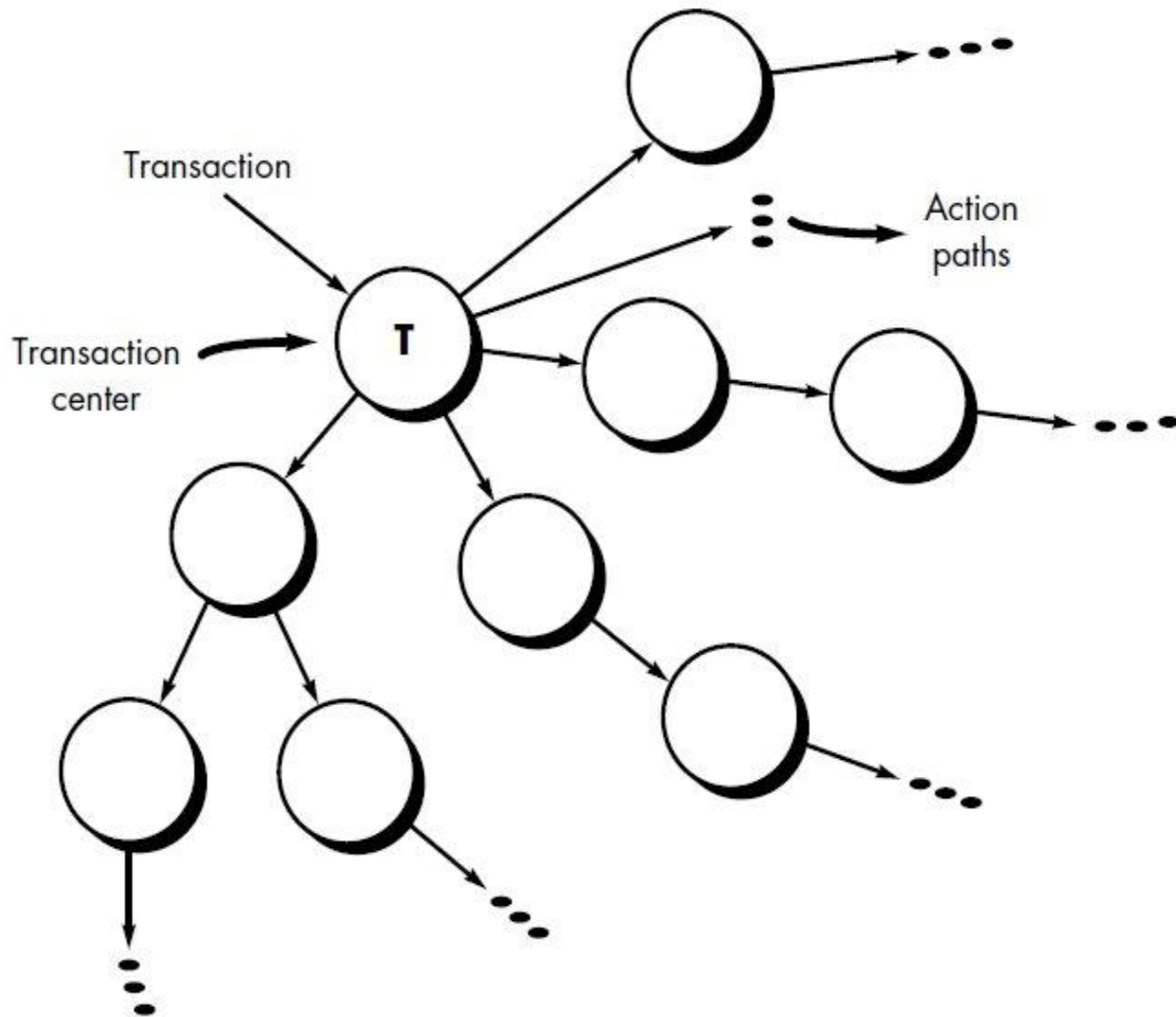


- *Luồng giao tác (transaction flow)*

- *Là loại luồng biến đổi đặc biệt, trong đó có một luồng vào và có nhiều luồng ra*

# Luồng giao tác

---



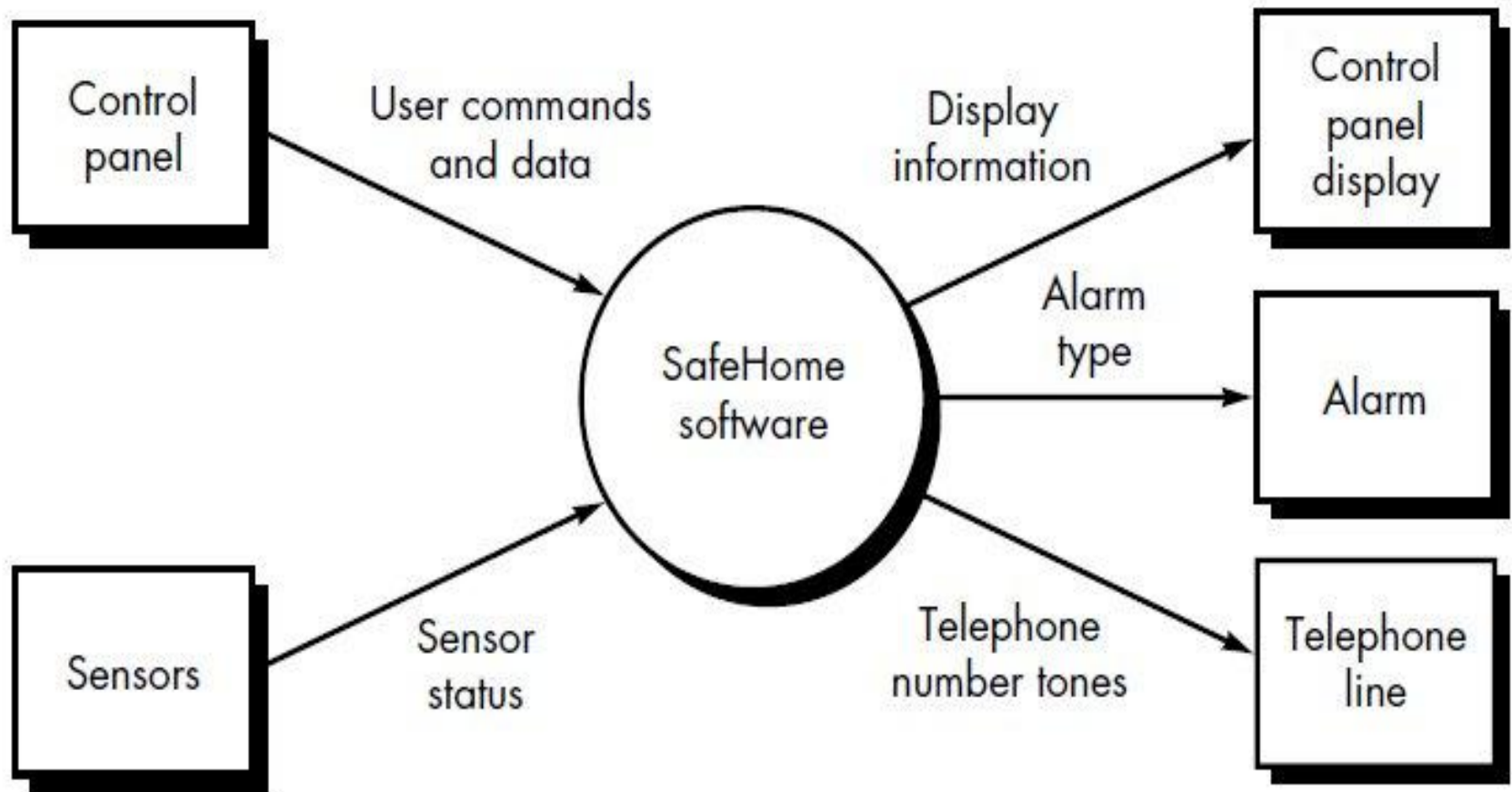
# Chuyển luồng biến đổi

---

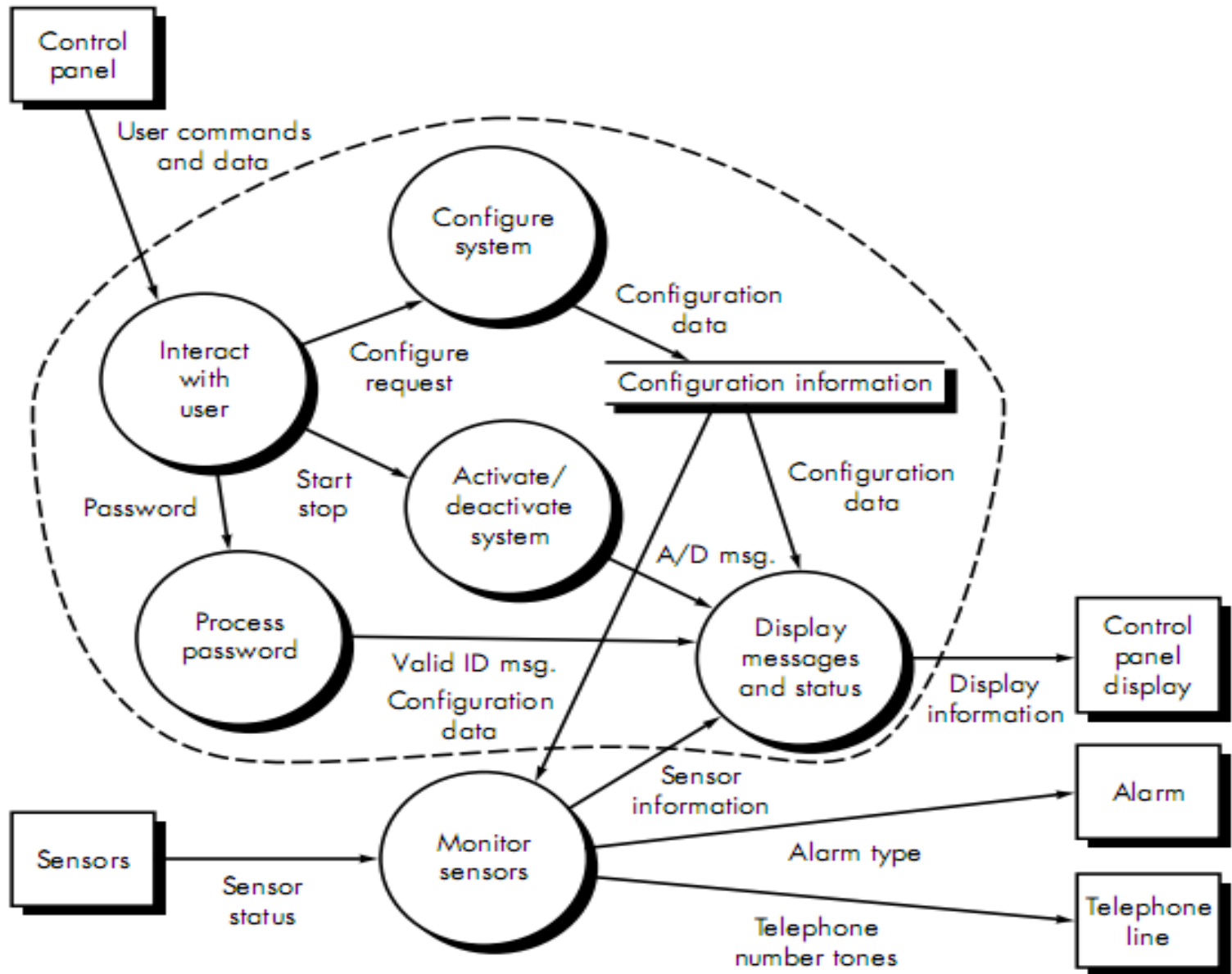
- *Bước 1: Làm mịn các biểu đồ luồng dữ liệu đến mức cần thiết*
- *Bước 2: Xác định các biên của luồng dữ liệu để xác định trung tâm biến đổi và các luồng vào và ra*
- *Bước 3: Chuyển đổi BDL sang lược đồ cấu trúc*
- *Bước 4: Tinh chỉnh LĐCT*

# Chuyển luồng biến đổi – Bước 1

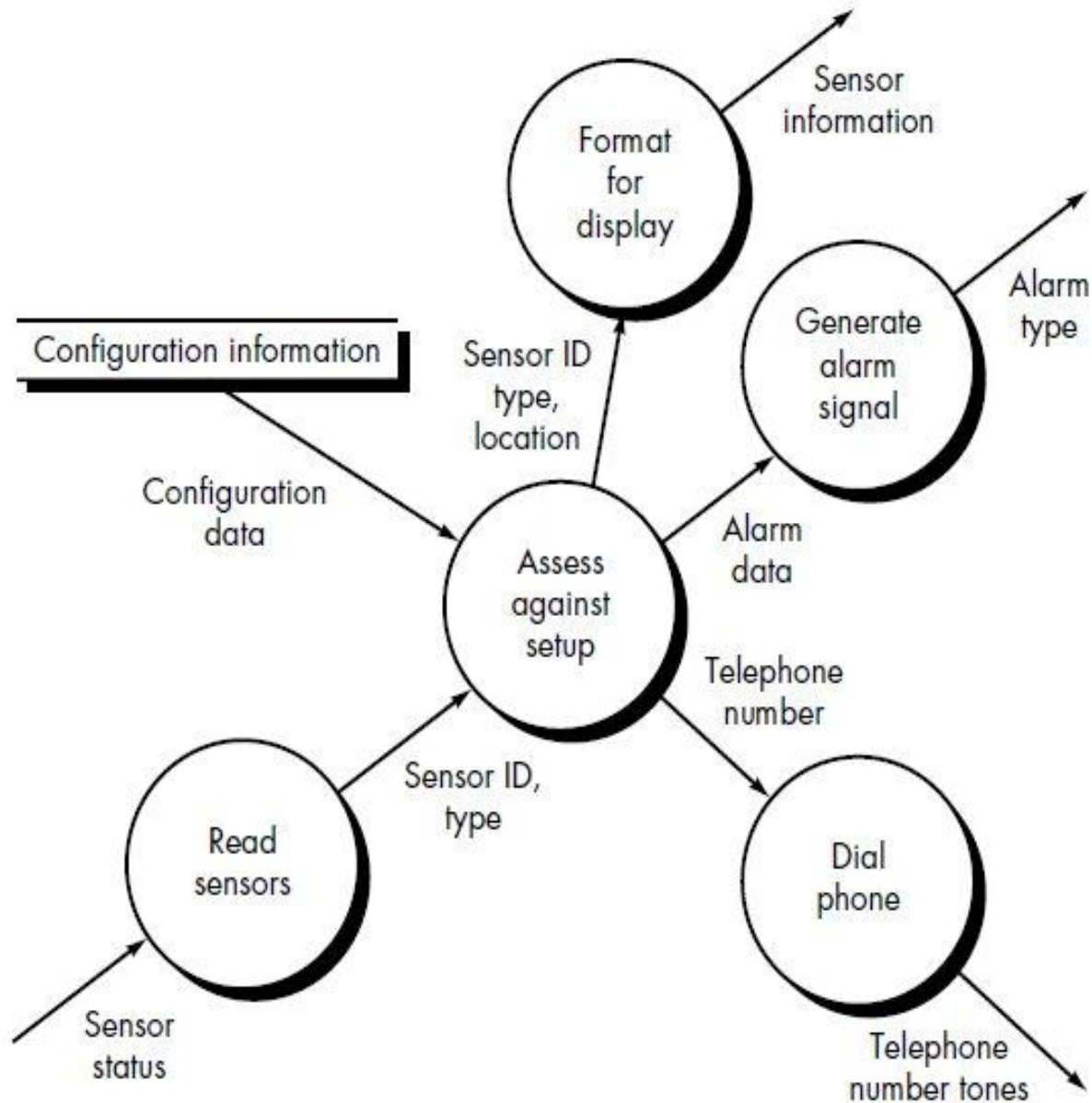
---



# Chuyển luồng biến đổi – Bước 1

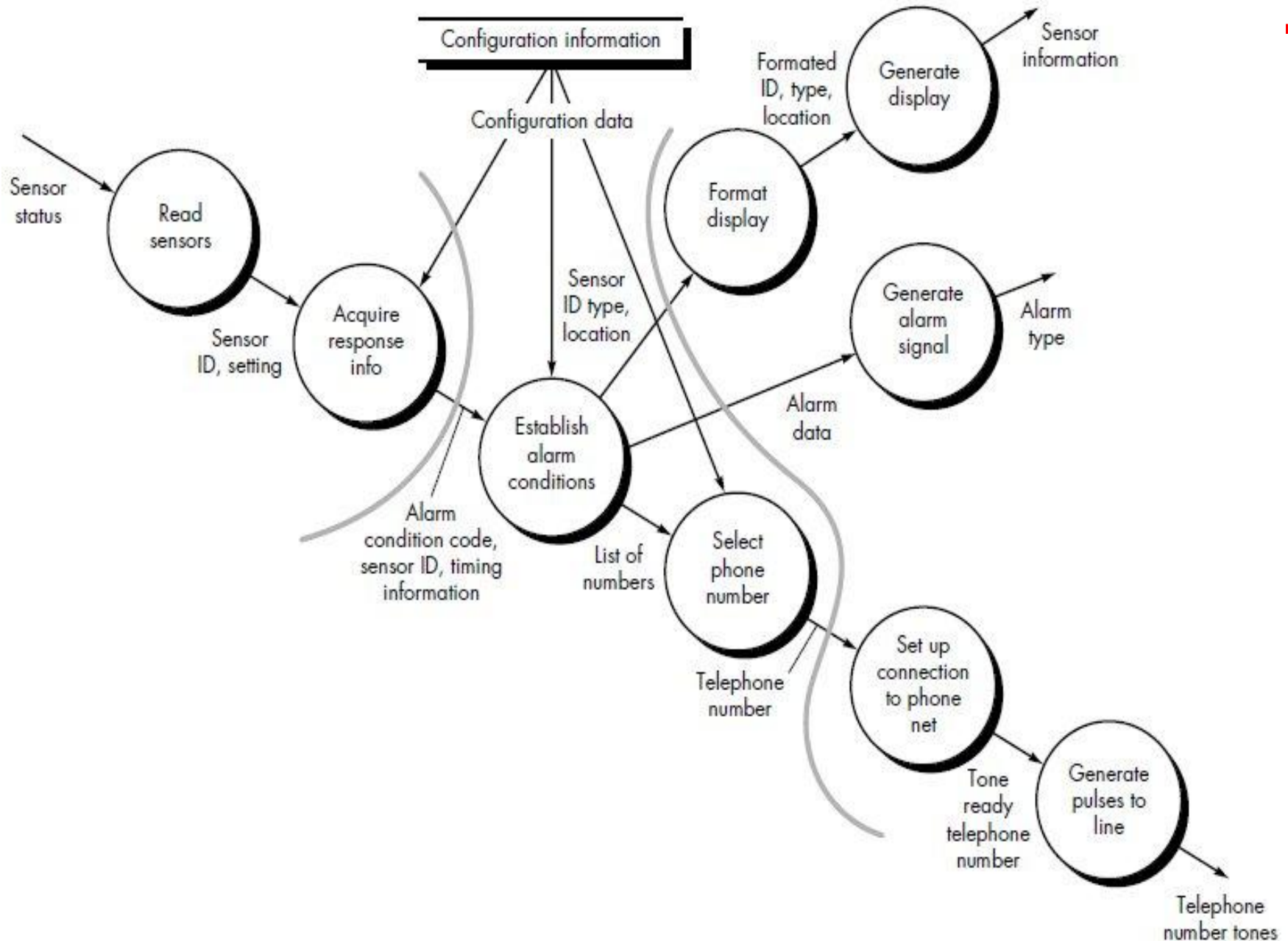


# Chuyển luồng biến đổi – Bước 1



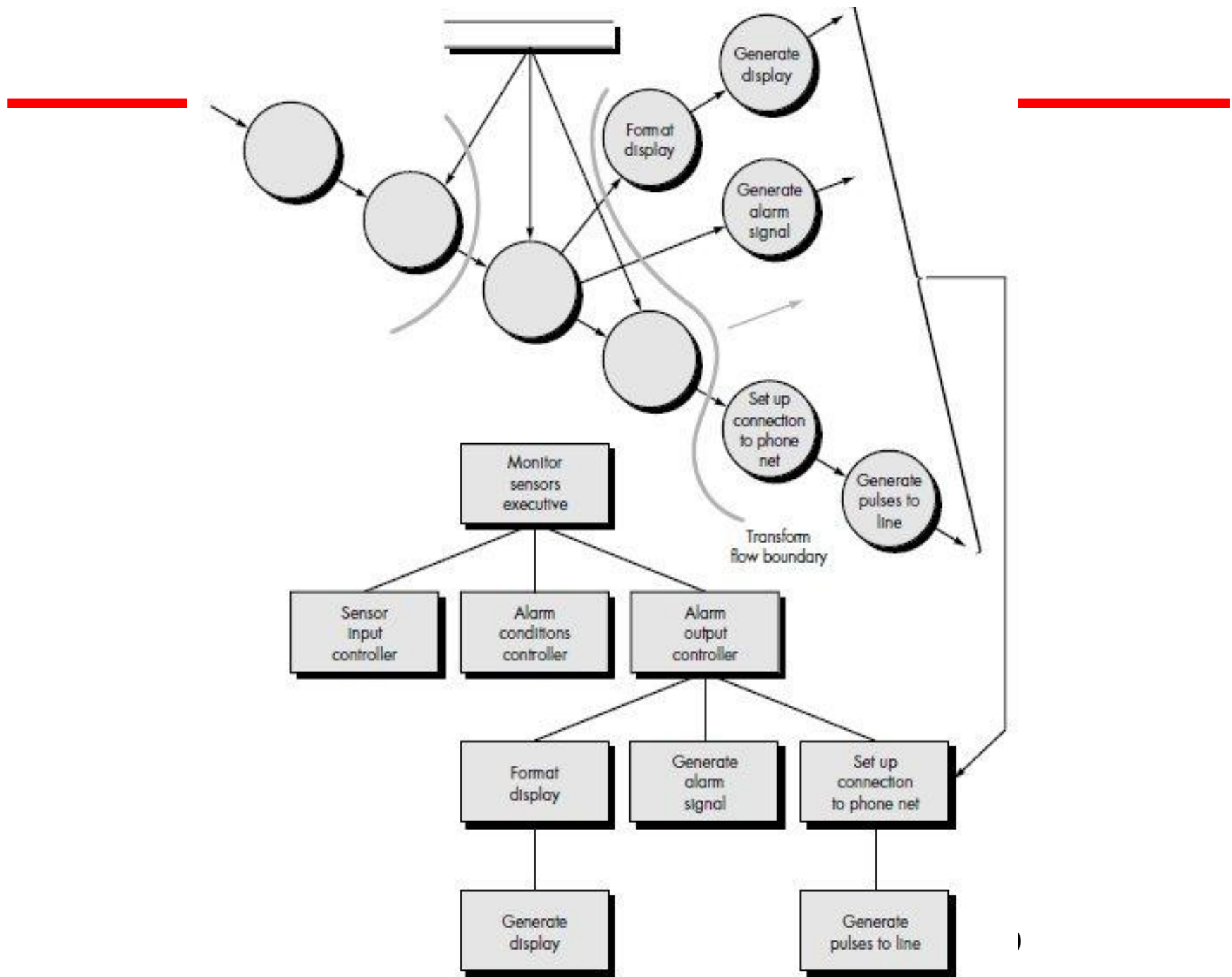


# Chuyển luồng biến đổi – Bước 2



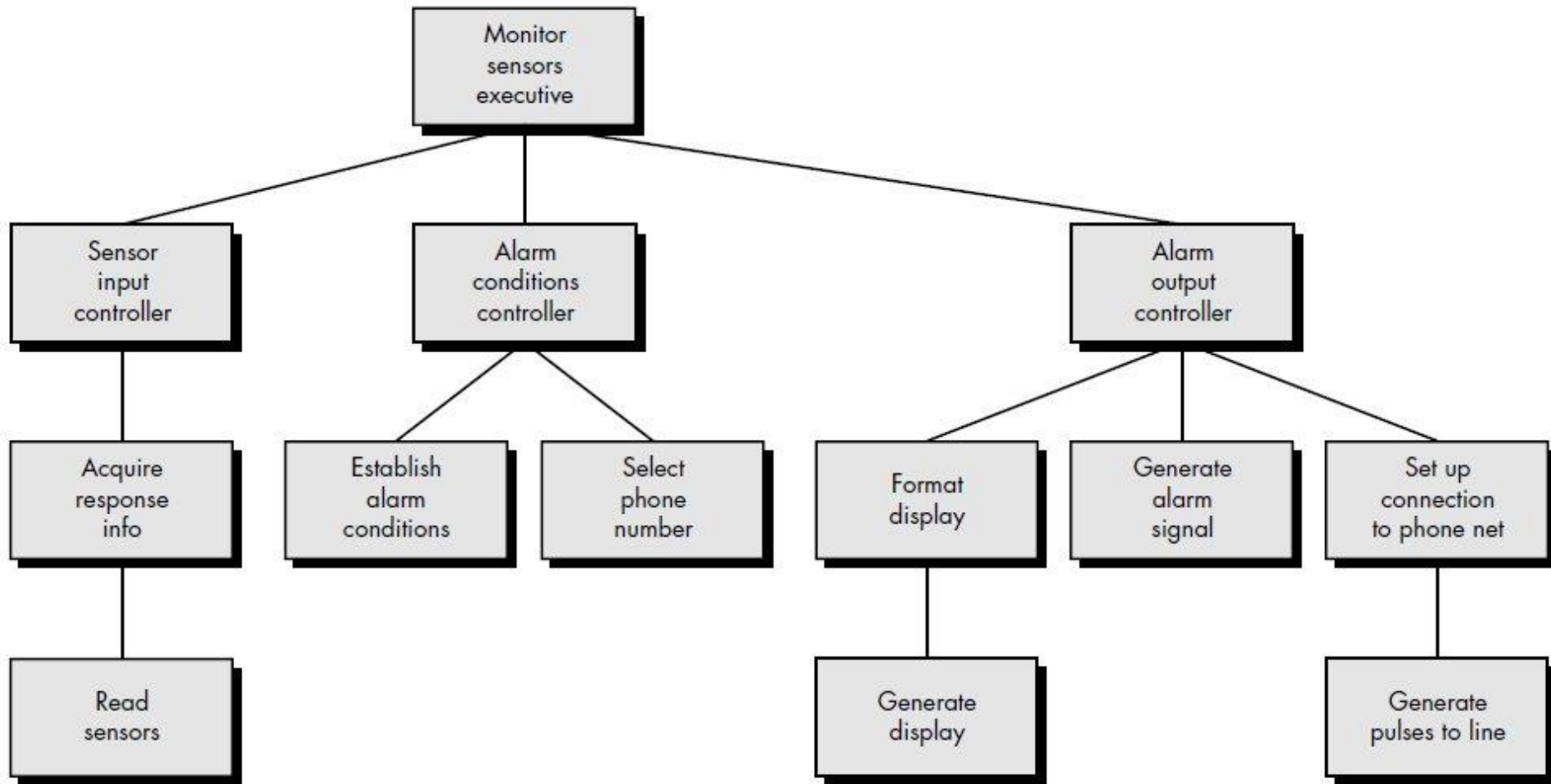
\_\_\_\_\_





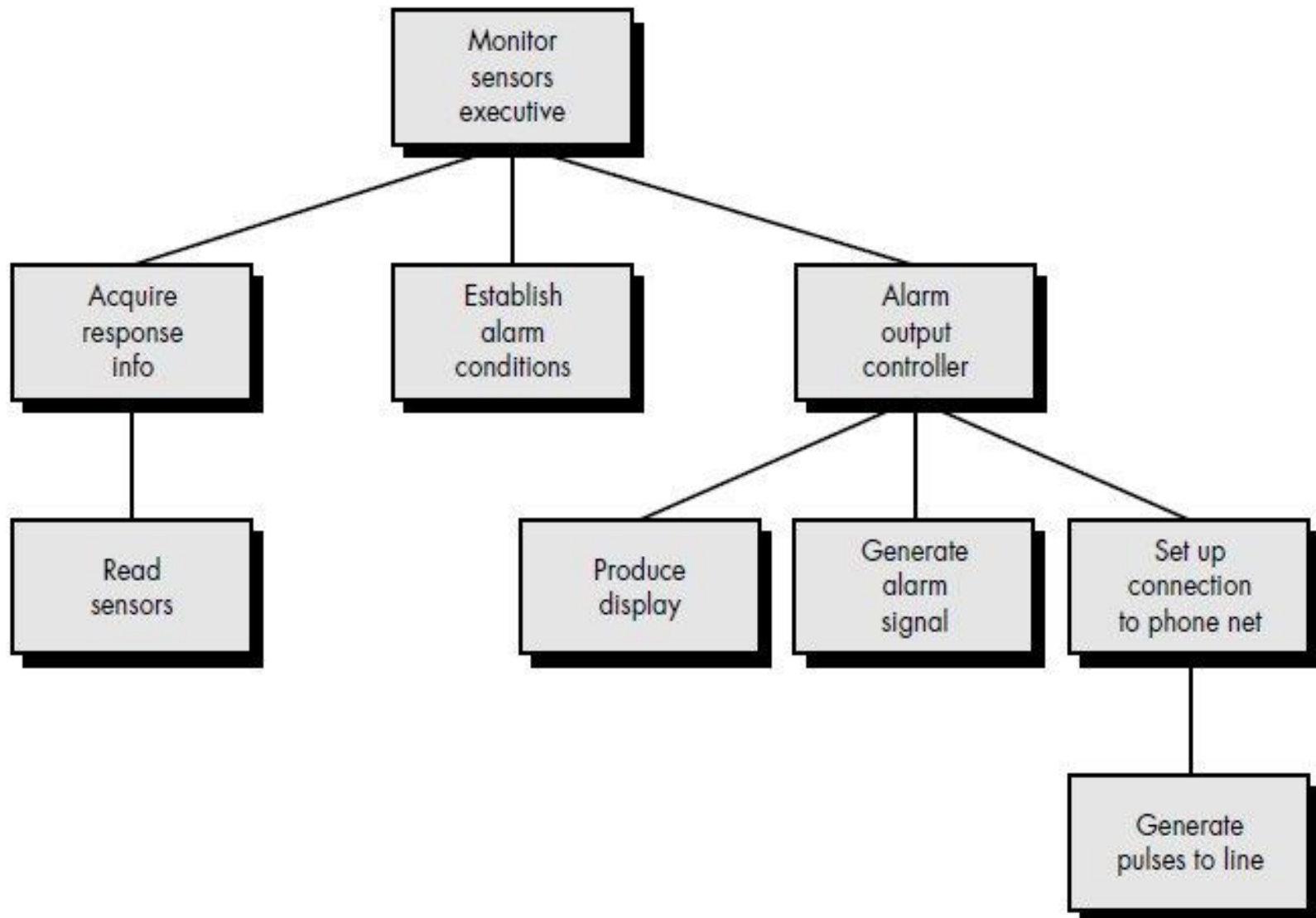
# Chuyển luồng biến đổi – Bước 4

---

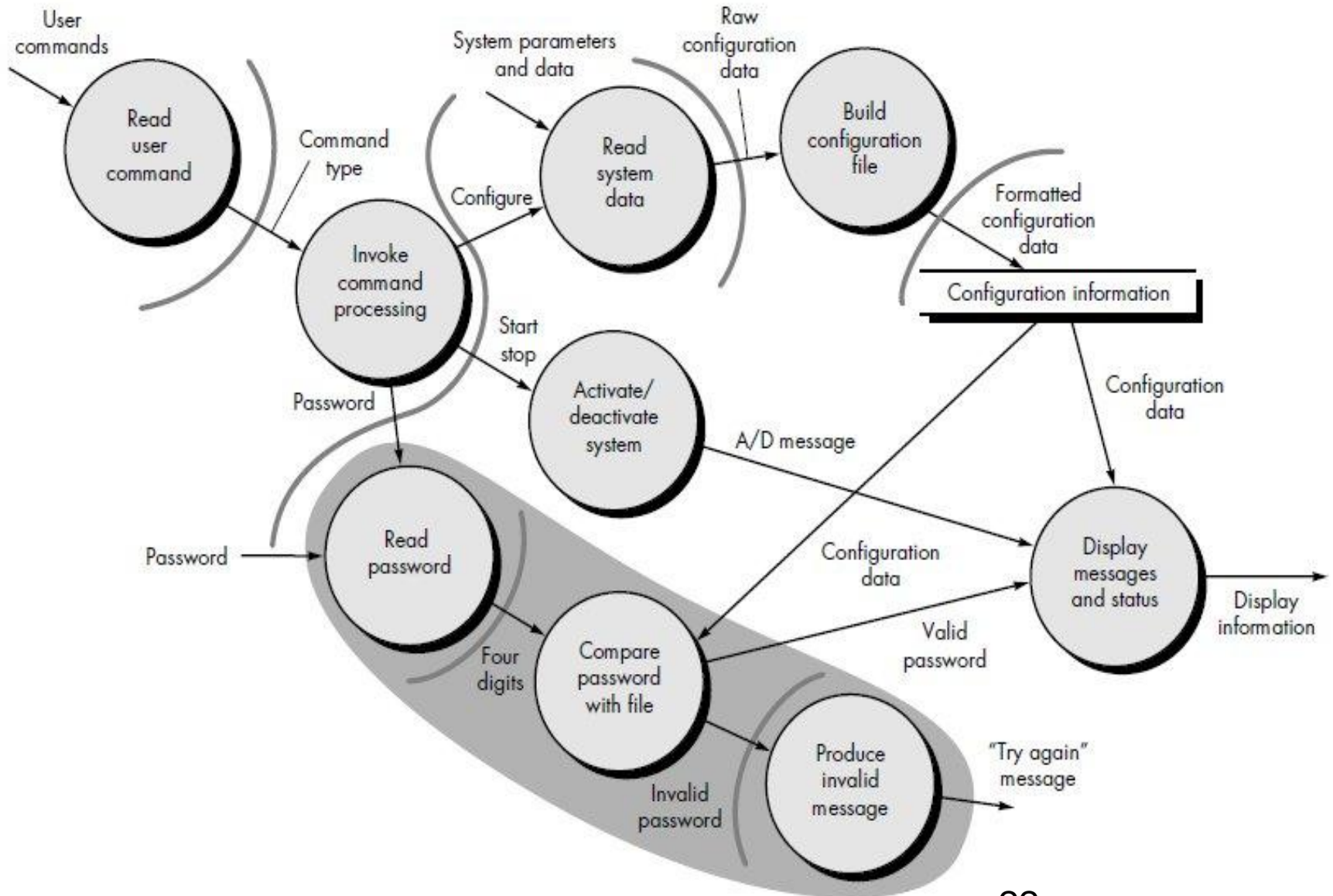


# Chuyển luồng biến đổi – Bước 4

---

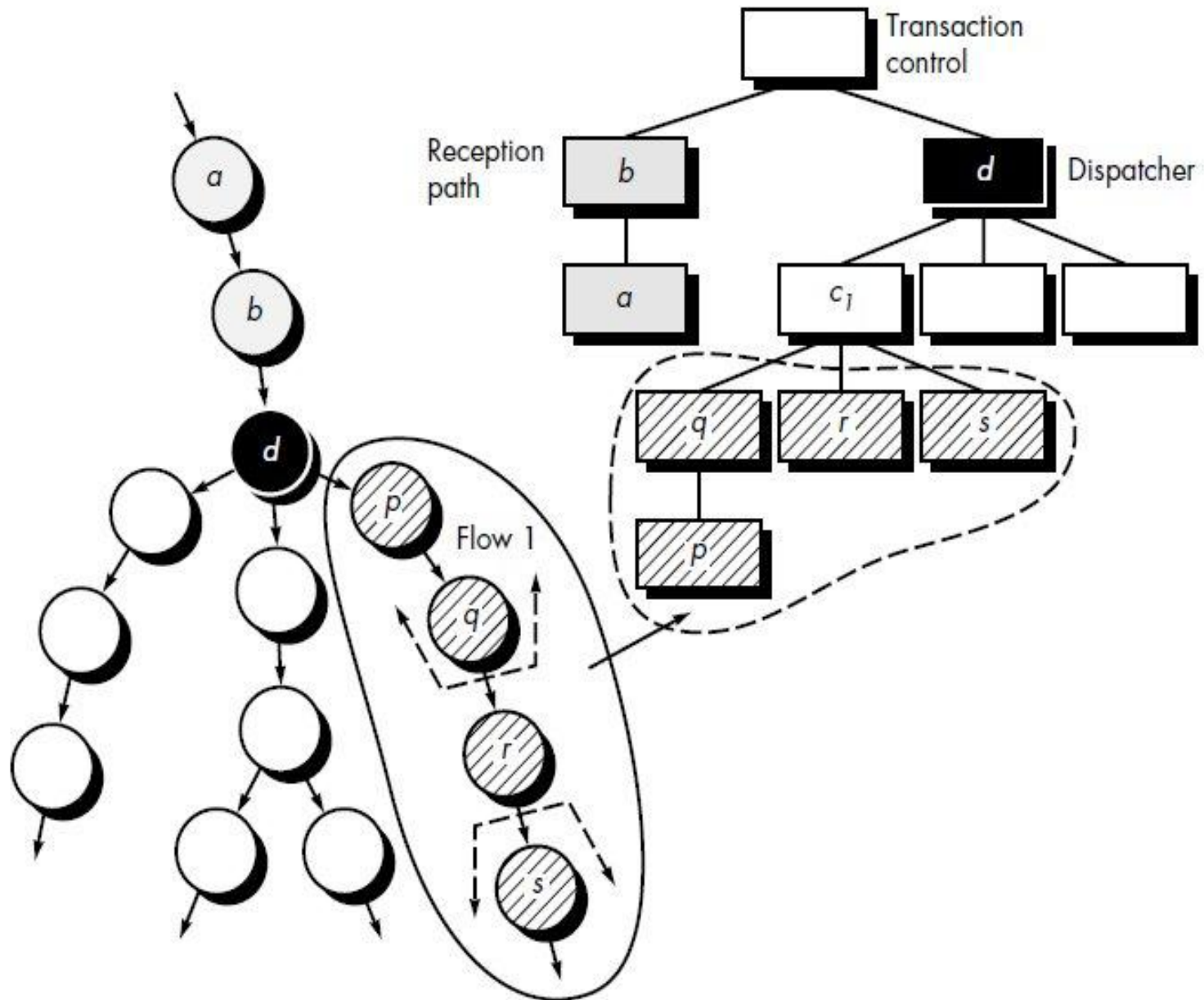


# Chuyển luồng giao tác – Bước 2



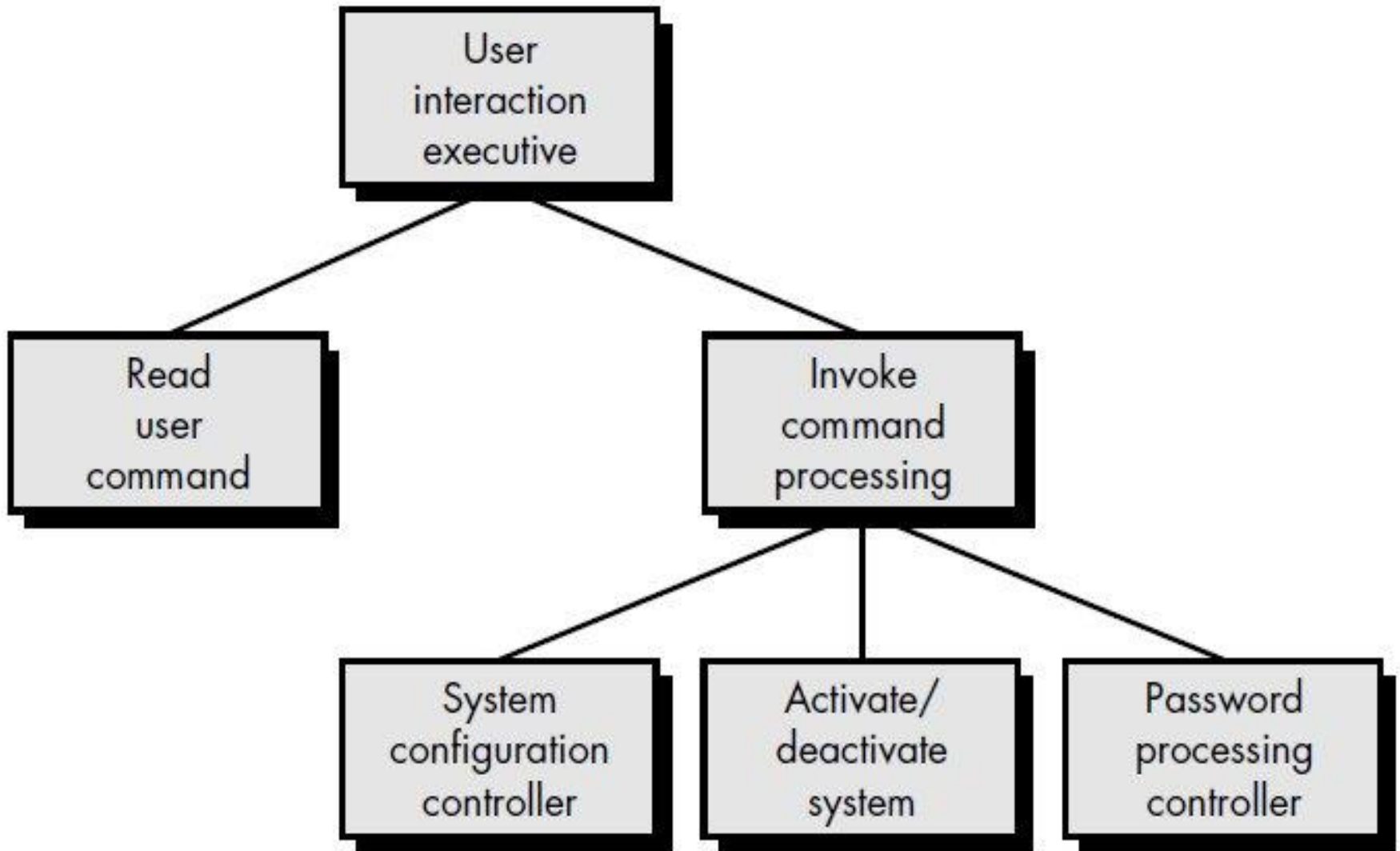


# Chuyển luồng giao tác – Bước 3



## Chuyển luồng giao tác – Bước 3

---





# Tóm tắt

---

- *Kiến trúc phần mềm và vai trò của việc thiết kế KTPM*
- *Các phong cách kiến trúc*
- *Phương pháp thiết kế có cấu trúc*
- *Các loại luồng dữ liệu*
  - *Luồng biến đổi*
  - *Luồng giao tác*
- *Cách chuyển đổi từ BDL sang lược đồ cấu trúc chương trình*