Software Design and Architecture

Engr. Abdul-Rahman Mahmood

DPM, MCP, QMR(ISO9001:2000)

armahmood786@yahoo.com
alphapeeler.sf.net/pubkeys/pkey.htm
pk.linkedin.com/in/armahmood
www.twitter.com/alphapeeler
www.facebook.com/alphapeeler
abdulmahmood-sss alphasecure
armahmood786@hotmail.com
http://alphapeeler.sf.net/me

alphasecure@gmail.com
http://alphapeeler.sourceforge.net
http://alphapeeler.tumblr.com
armahmood786@jabber.org
alphapeeler@aim.com
mahmood_cubix 48660186
alphapeeler@icloud.com
http://alphapeeler.sf.net/acms/

Architectural Design

Software architecture

- The design process for identifying the sub-systems making up a system and the framework for sub-system control and communication is architectural design.
- The output of this design process is a description of the software architecture.

Architecture and system characteristics

- Performance
  - Localise critical operations and minimise communications. Use large rather than fine-grain components.
- Security
  - Use a layered architecture with critical assets in the inner layers.
- Safety
  - Localise safety-critical features in a small number of sub-systems.
- Availability
  - Include redundant components and mechanisms for fault tolerance.
- Maintainability
  - Use fine-grain, replaceable components.

Architectural design decisions

- During the architectural design process, system architects have to make a number of fundamental decisions that profoundly affect the system and its development process. Based on their knowledge and experience, they have to answer the following fundamental questions:
  - Is there a **generic application architecture** that can be used?
  - How will the system be **distributed**?
  - What **architectural styles** are appropriate?
  - What **approach** will be used to **structure the system**?
  - How will the system be decomposed into **modules?**
  - What **control strategy** should be used?
  - How will the architectural design be **evaluated**?
  - How should the architecture be **documented**?

Architectural styles

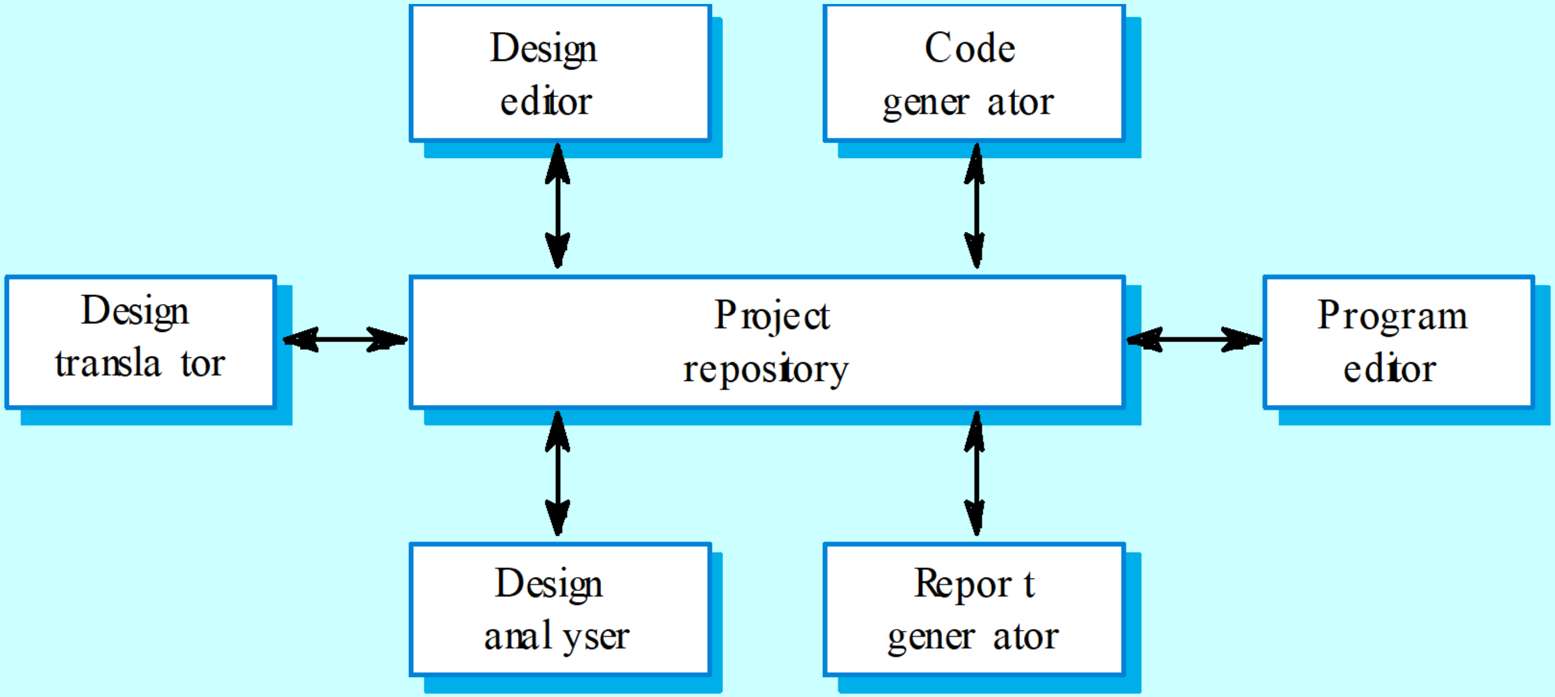
System organisation

- Reflects the basic strategy that is used to structure a system.
- **Three organisational styles are widely used**:
  - A **shared data** repository style;
  - A **shared services** and servers style;
  - An abstract machine **or layered style**.

The repository model

- **Sub-systems must exchange data**. This may be done in two ways:
  - Shared data is held in a **central database** or repository and may be accessed by all sub-systems;
  - Each **sub-system maintains its own database** and passes data explicitly to other sub-systems.
- When large amounts of data are to be shared, the repository model of sharing is most commonly used.
- This model is therefore suited to applications where data is generated by one sub-system and used by another.
- Examples of this type of system include **command and control systems**, **management information systems**, **CAD** systems and **CASE toolsets**.

CASE toolset architecture



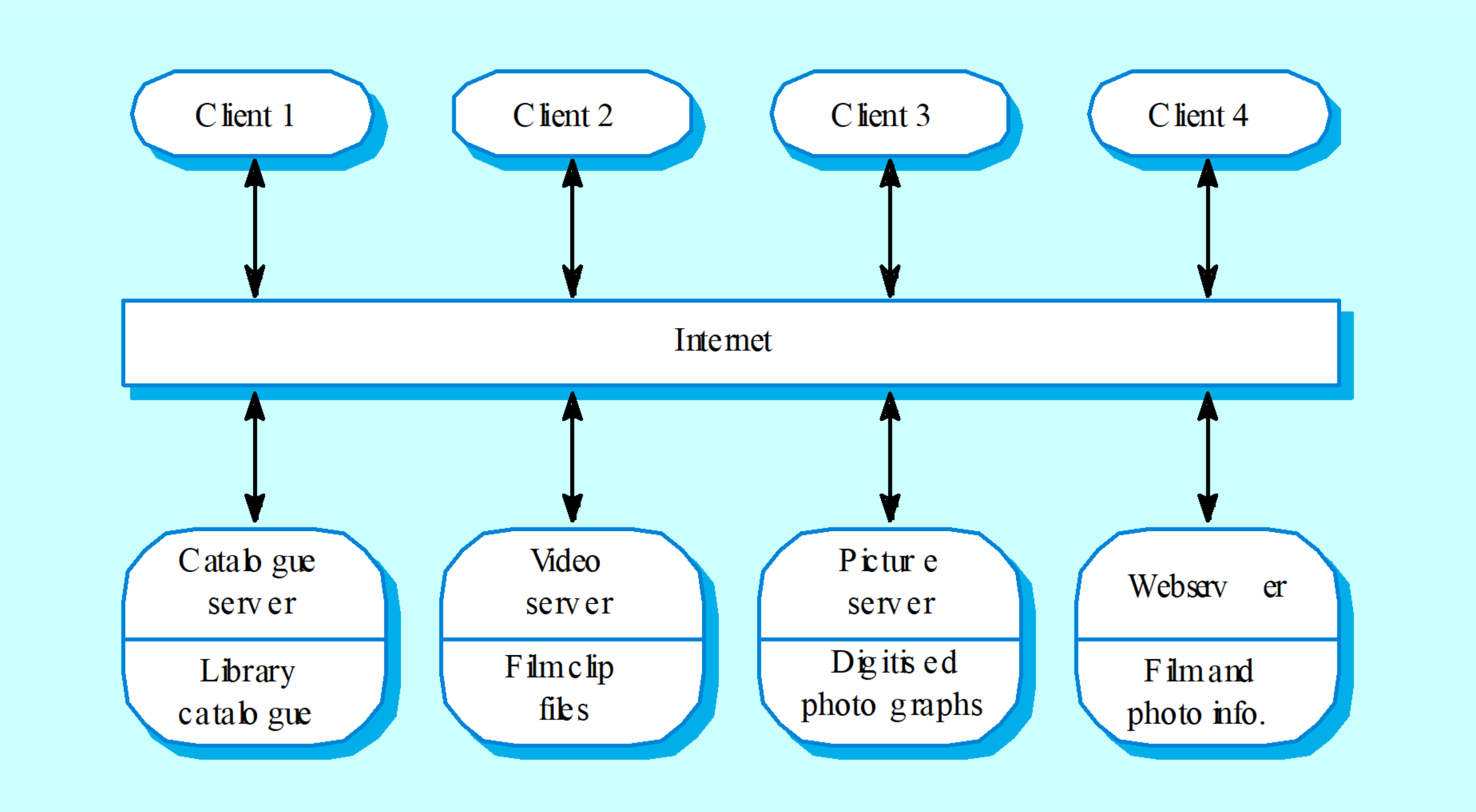
Repository model characteristics

- Advantages
  - Efficient way to share **large amounts of data**;
  - Sub-systems need not be concerned with how data is produced **Centralised management e.g. backup, security, etc**.
  - Sharing model is **published** as the **repository schema**.
- Disadvantages
  - Sub-systems **must agree on a repository data model**. Inevitably a compromise;
  - **Data evolution** is difficult and expensive;
  - No scope for specific **management policies**;
  - **Difficult to distribute** efficiently.

Client-server model

- **Distributed system model which shows how data and processing is distributed across a range of components.**
- The client–server architectural model is a system model where the system is organised as a set of services and associated servers and clients that access and use the services.
- The major components of this model are:
  - Set of **stand-alone servers** which provide specific services such as printing, data management, etc.
  - Set of **clients** which call on these services.
  - **Network** which allows clients to access servers.

Film and picture library



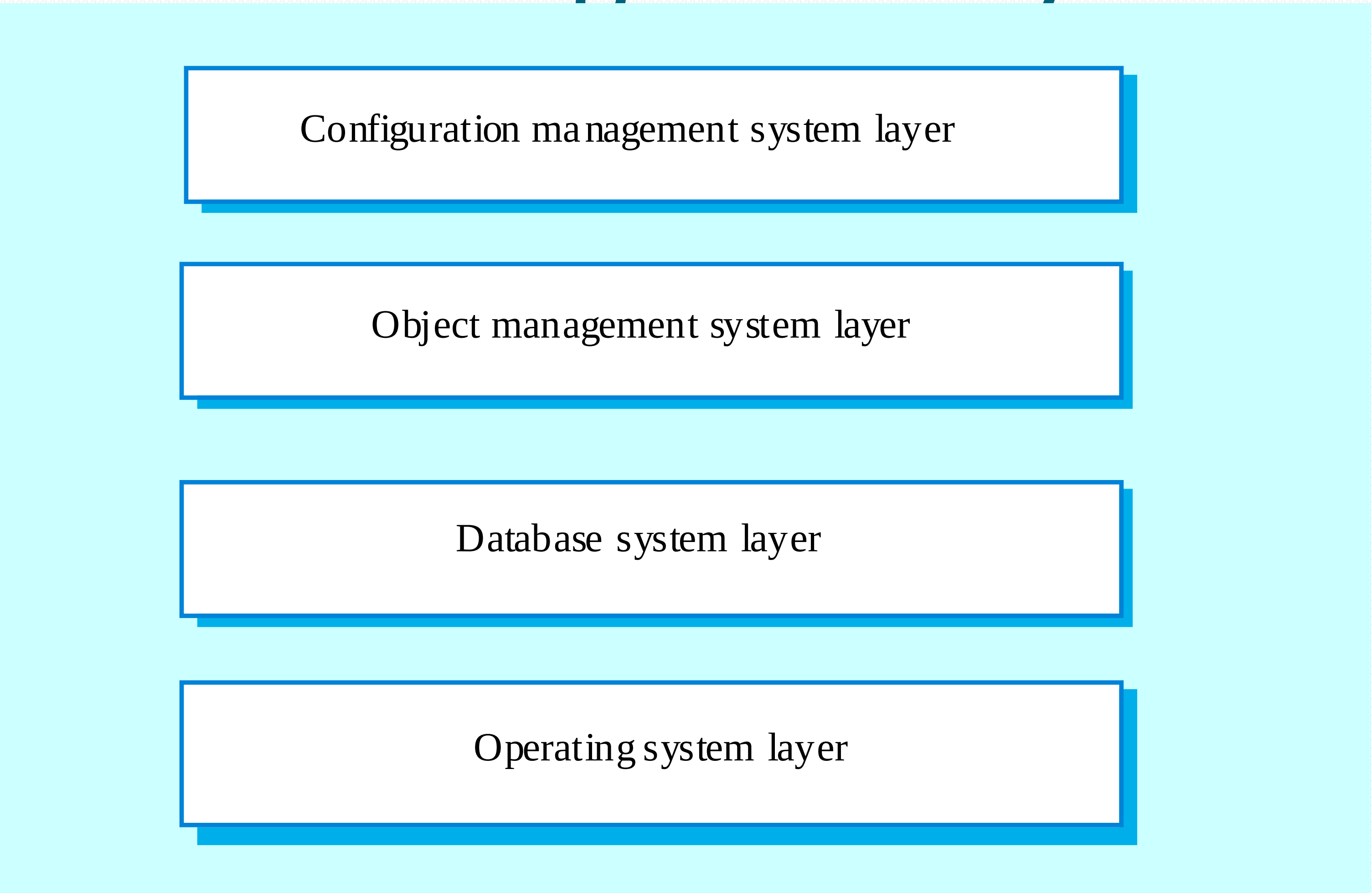
Client-server characteristics

- Advantages
  - Distribution of data is straightforward;
  - Makes effective use of networked systems. May require cheaper hardware;
  - Easy to add new servers or upgrade existing servers.
- Disadvantages
  - No shared data model so sub-systems use different data organisation. Data interchange may be inefficient;
  - Redundant management in each server;
  - No central register of names and services - it may be hard to find out what servers and services are available.

Abstract machine (layered) model

- The layered model of an architecture (an abstract machine model) **organizes a system into layers, each of which provide a set of services.**
- Used to model the interfacing of sub-systems.
- Organises the system into a set of layers (or abstract machines) each of which provide a set of services.
- Supports the incremental development of sub-systems in different layers. When a layer interface changes, only the adjacent layer is affected.
- However, often artificial to structure systems in this way.

Version management system



Configuration management system layer

Object management system layer

Database system layer

Operating system layer

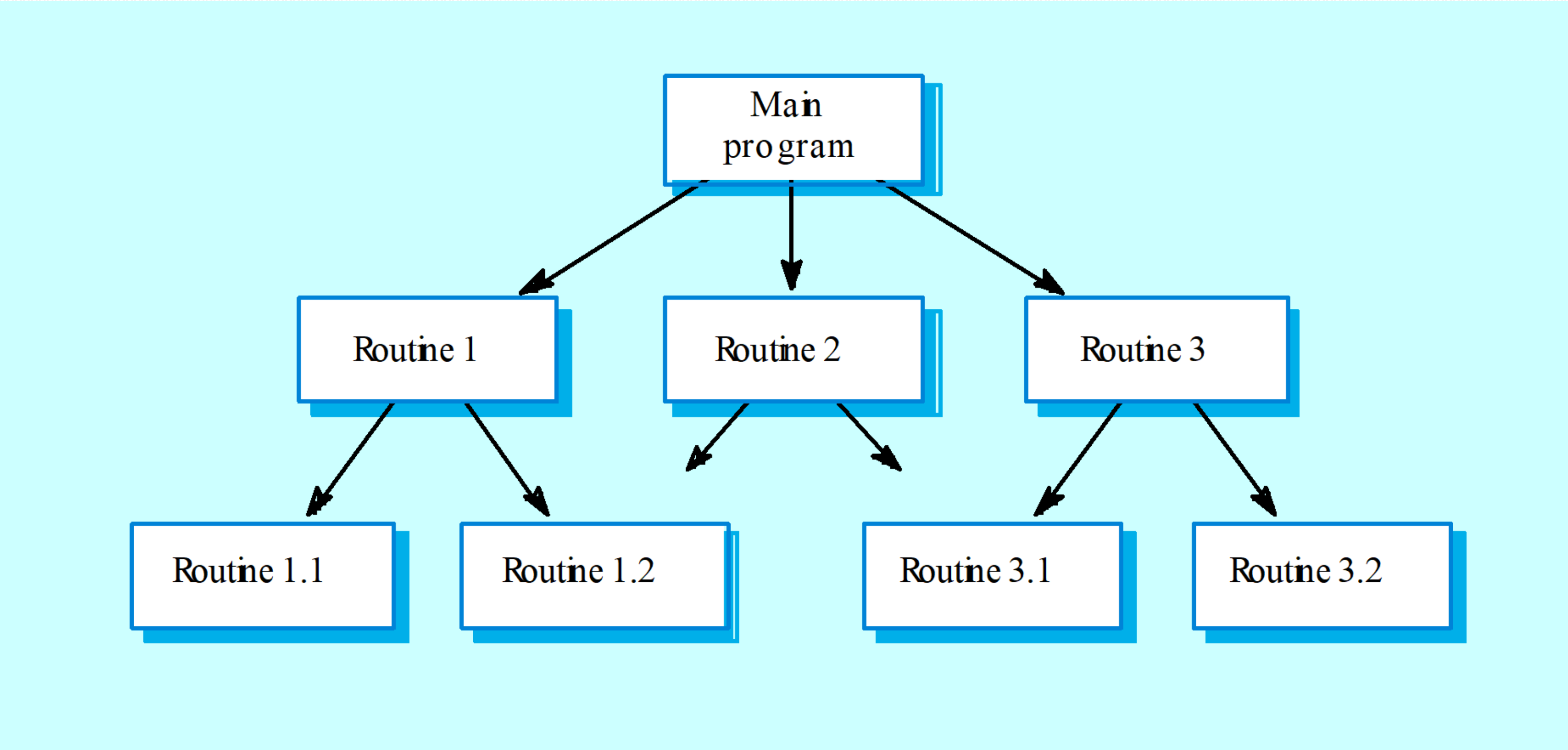
Control styles

- Are concerned with the control flow between sub-systems. Distinct from the system decomposition model.
- Centralised control
  - One sub-system has overall responsibility for control and starts and stops other sub-systems.
- Event-based control
  - Each sub-system can respond to externally generated events from other sub-systems or the system’s environment.

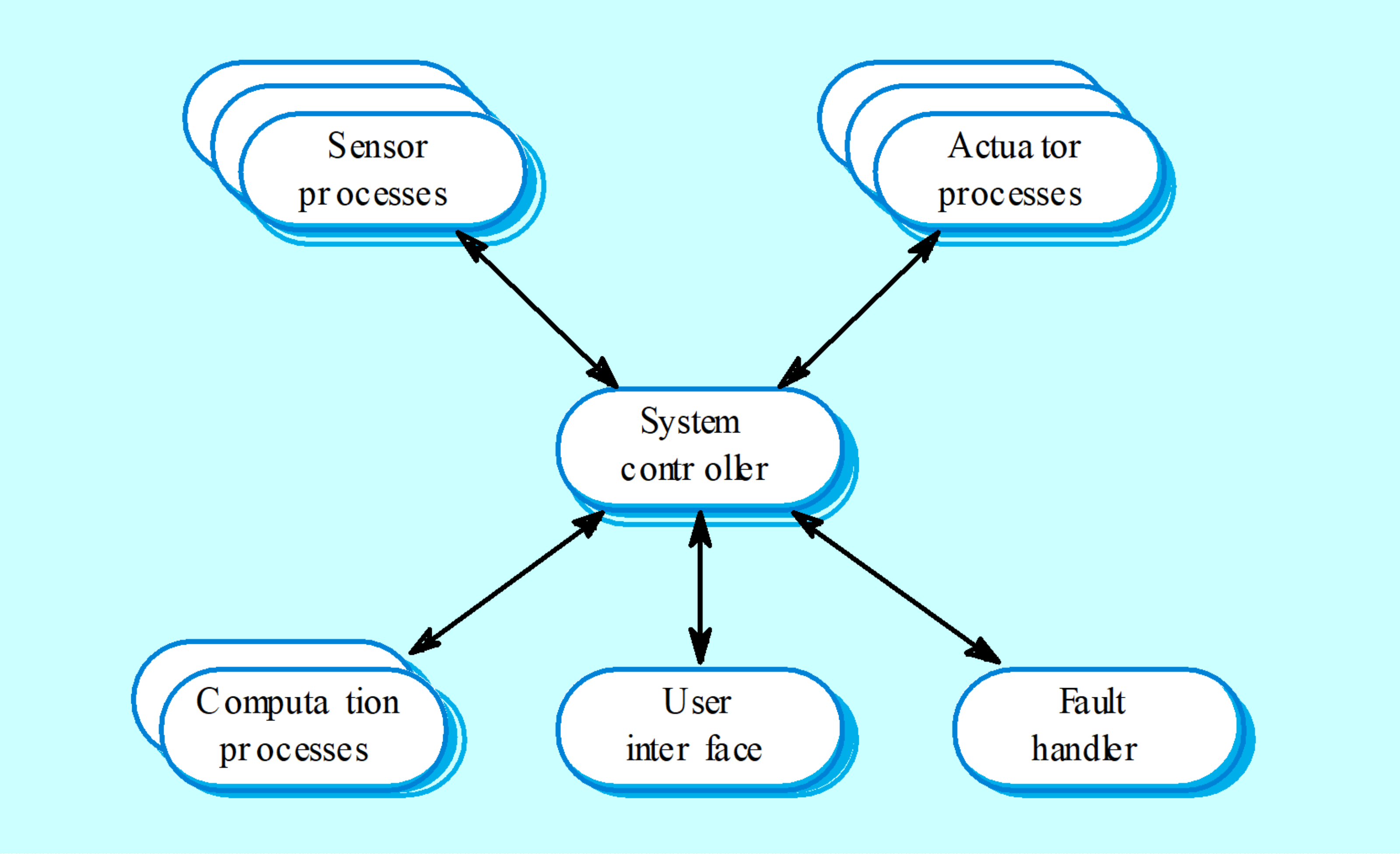
Centralised control

- A control sub-system takes responsibility for managing the execution of other sub-systems.
- Call-return model
  - Top-down subroutine model where control starts at the top of a subroutine hierarchy and moves downwards. Applicable to sequential systems.
- Manager model
  - Applicable to concurrent systems. One system component controls the stopping, starting and coordination of other system processes. Can be implemented in sequential systems as a case statement.

Call-return model



Real-time system control



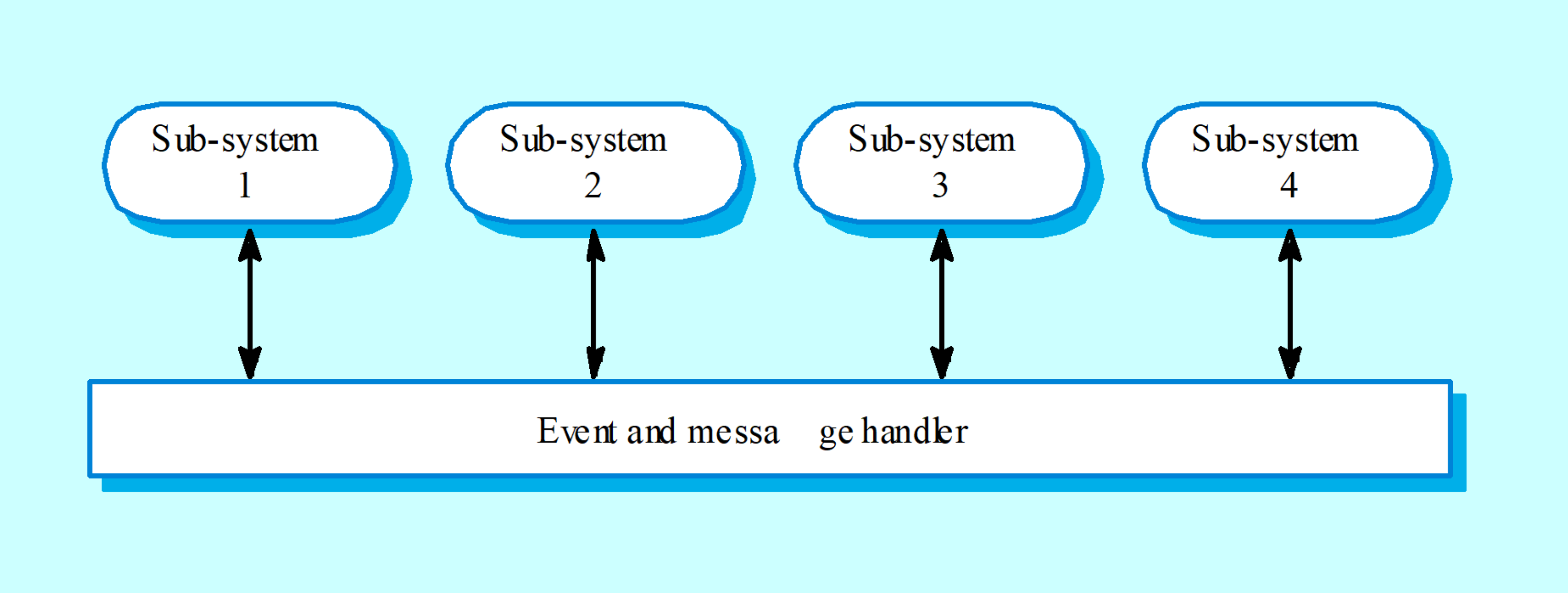
Event-driven systems

- Driven by externally generated events where the timing of the event is out of the control of the sub-systems which process the event.
- Two principal event-driven models
  - **Broadcast models.** An event is broadcast to all sub-systems. Any sub-system which can handle the event may do so;
  - **Interrupt-driven models.** Used in real-time systems where interrupts are detected by an interrupt handler and passed to some other component for processing.
- Other event driven models include spreadsheets and production systems.

Broadcast model

- Effective in integrating sub-systems on different computers in a network.
- Sub-systems register an interest in specific events. When these occur, control is transferred to the sub-system which can handle the event.
- Control policy is not embedded in the event and message handler. Sub-systems decide on events of interest to them.
- However, sub-systems don’t know if or when an event will be handled.

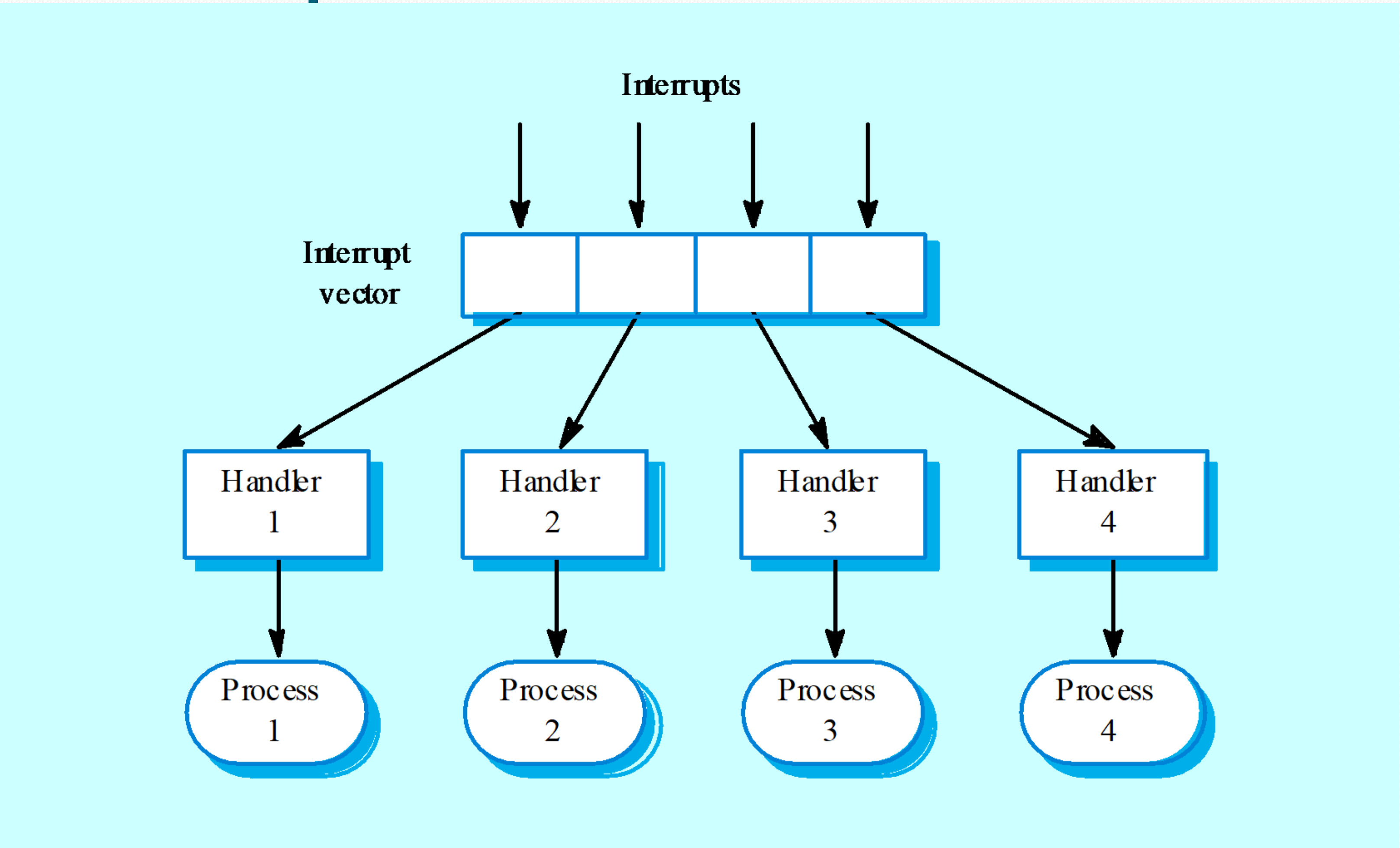
Selective broadcasting



Interrupt-driven systems

- Used in real-time systems where fast response to an event is essential.
- There are known interrupt types with a handler defined for each type.
- Each type is associated with a memory location and a hardware switch causes transfer to its handler.
- Allows fast response but complex to program and difficult to validate.

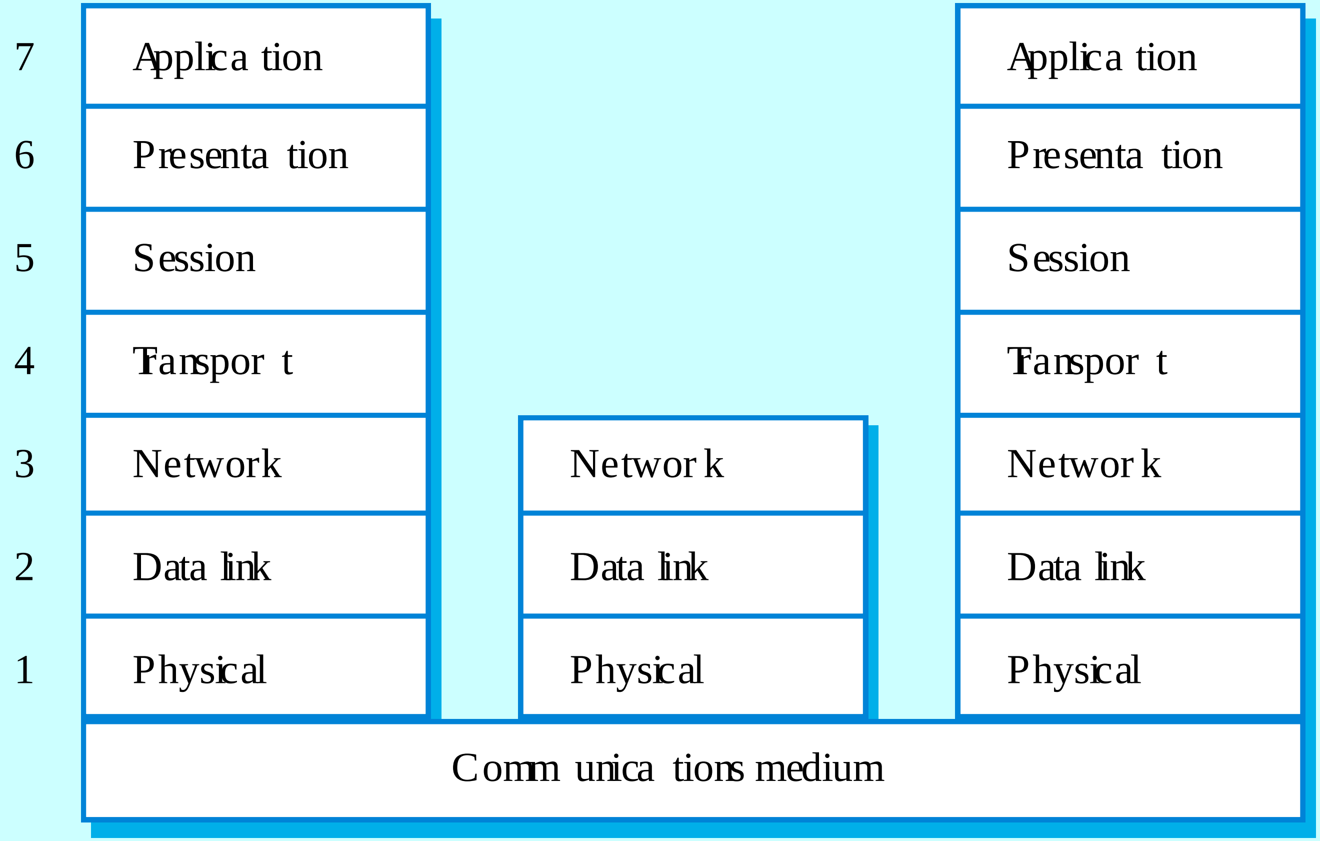
Interrupt-driven control



Reference architectures

- Reference models are derived from a study of the application domain rather than from existing systems.
- May be used as a basis for system implementation or to compare different systems. It acts as a standard against which systems can be evaluated.
- OSI model is a layered model for communication systems.

OSI reference model



7	Application		Application
6	Presentation		Presentation
5	Session		Session
4	Transport		Transport
3	Network	Network	Network
2	Data link	Data link	Data link
1	Physical	Physical	Physical

Communications medium