Chapter 6
Architectural Design

Lesson Objectives

- Discuss why architectural design of software is important
- Explain the 3 main activities in architectural design – overall system organization, modular decomposition and control modeling
- Distinguish the models/styles involve in System Organization and Control Modeling

Why Architectural Design is Important?

Introduction

- Large systems can be decomposed into sub-systems that provide some related set of services.
- The initial design process of identifying these sub-systems and establishing a framework for sub-system control and communication is called architectural design.
- Architectural design is the first stage in the design process and usually comes before detailed system specification.

Advantages of Explicit Architecture

- Stakeholder communication
  - used during discussion to show high-level presentation of system
- System analysis
  - check whether the system can meet critical requirements
- Large-scale reuse
  - The architecture may be reusable across a range of systems since it shows us how a system is organized and how the components interoperate.

Advantages of Explicit Architecture

4. Negotiation

✓ Serve as design plan that can be used for negotiation & discussion

Advantages of Explicit Architecture

5. Complexity Management

✓ Act as essential tool for complexity management

Architectural Design Activities

1. System Organization
2. Modular Decomposition
3. Control Modeling

1. System Organization/Structuring

1. System Organization/Structuring

Represented in Block Diagram

Sub-systems

1. System Organization/Structuring

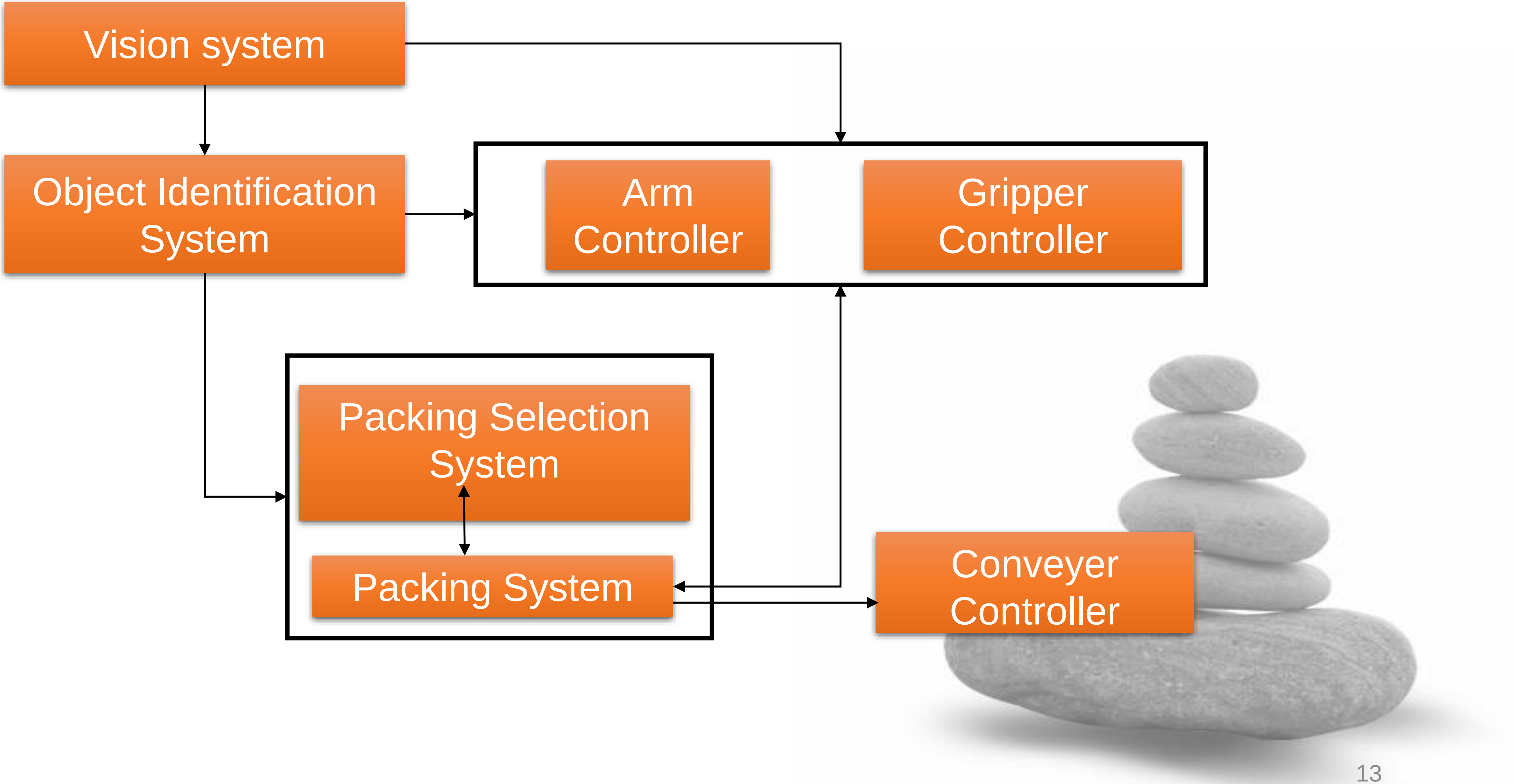
E.g. packing robot system

1. System Organization/Structuring

- This is the first phase of the architectural design activity.

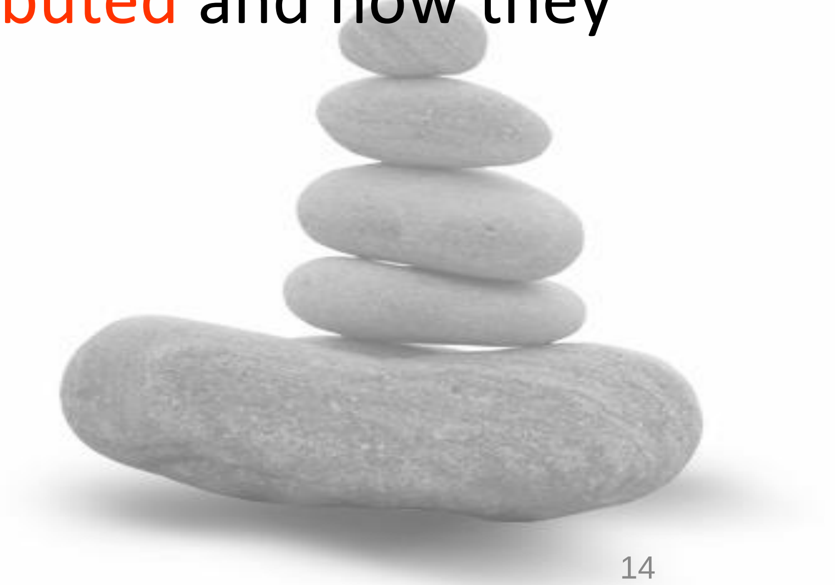
- At its most abstract level, an architectural design may be depicted as a block diagram in which each box represents a sub-system, as shown in the following figure: (next slide)

1. System Organization/Structuring



1. System Organization/Structuring

- The system is structured into a number of principal sub-systems where a sub-system is an *independent software unit*.
- Communications between sub-systems are also identified.
- More specific models of the structure may be developed which show how sub-systems share data, how they are distributed and how they interface with each other.



1. System Organization/Structuring

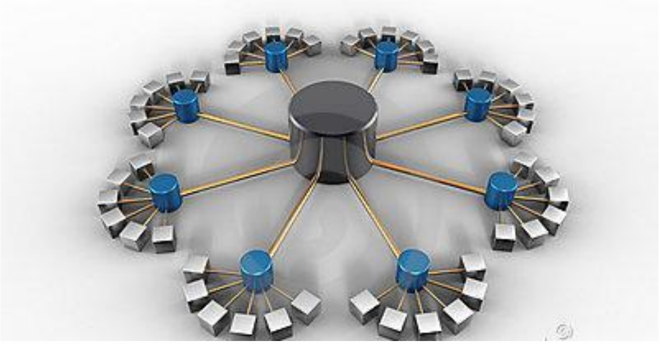
✓ 3 widely used system organization styles/models are:

i) Repository model

ii) Client-server model

iii) Layered model

1. System Organization/Structuring

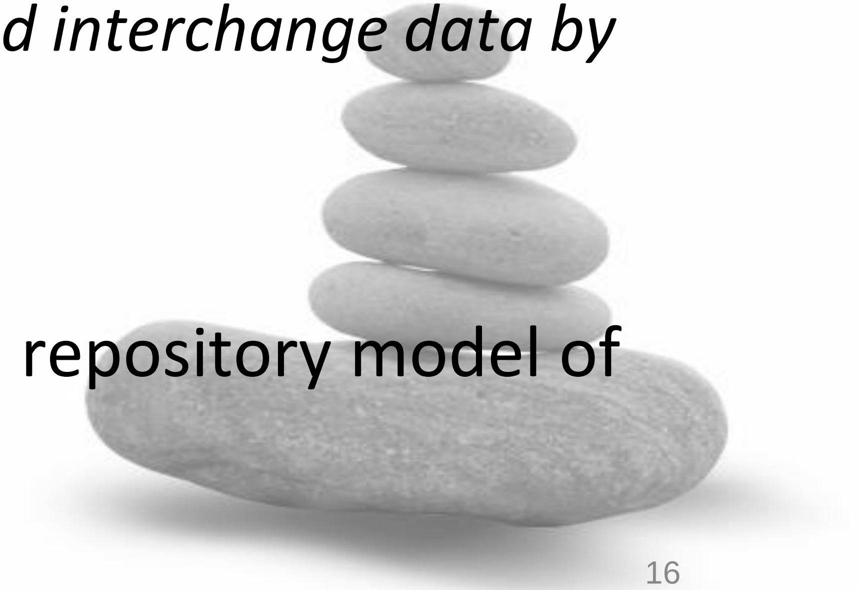


i) Repository Model

- Sub-systems need to exchange data.

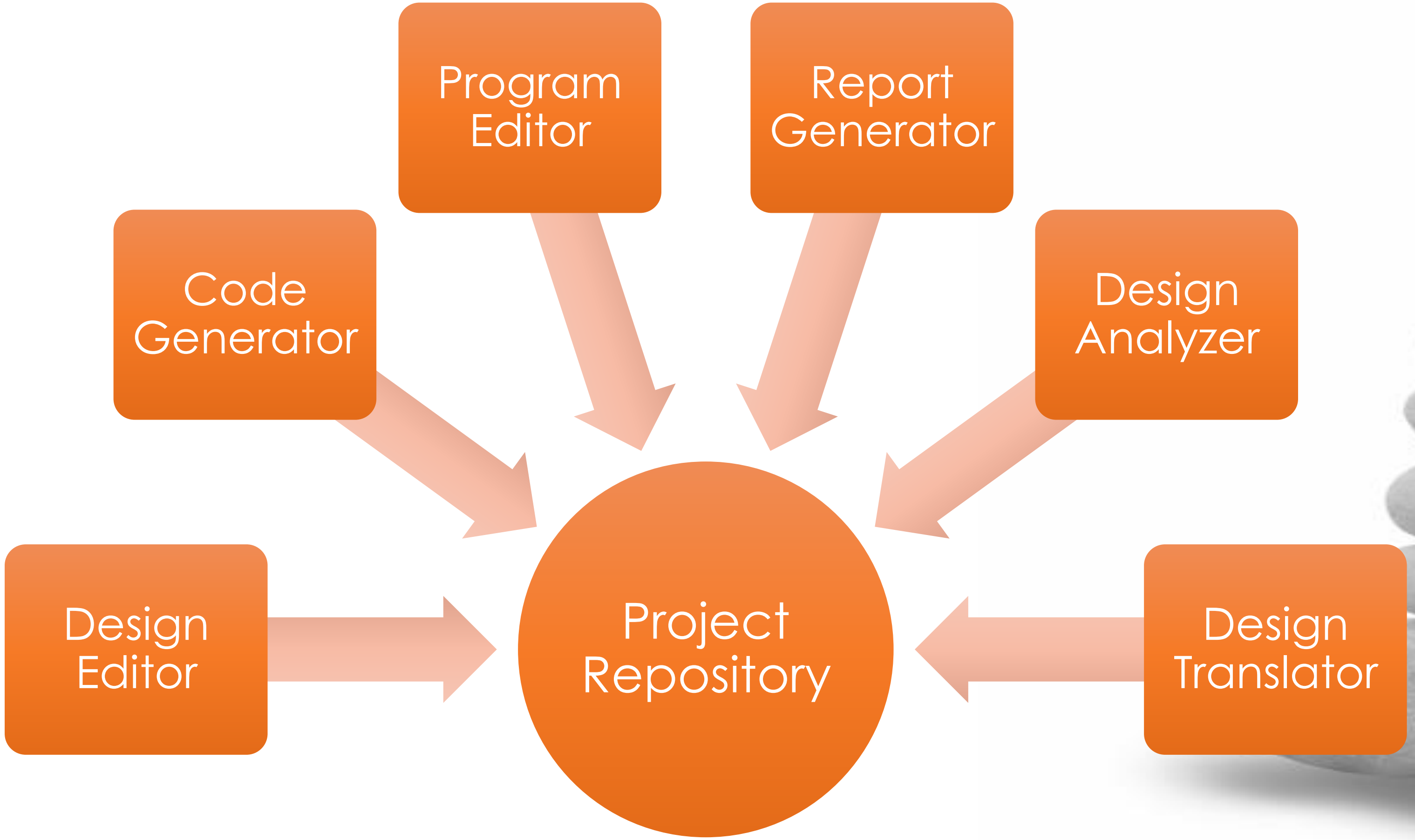
  This may be done in two ways:
  - *All shared data is held in a central database.*
  - *Each sub-system maintain its own database and interchange data by passing messages*

- When to use?
  - large amounts of data are to be shared, the repository model of sharing is most commonly used.



1. System Organization/Structuring

i) Repository Model -The architecture of an integrated CASE toolset



Program Editor

Report Generator

Code Generator

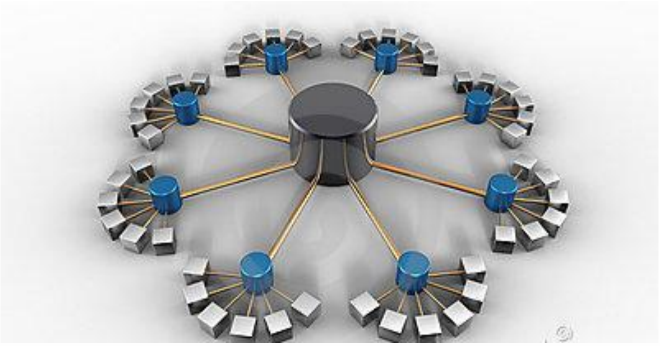
Design Analyzer

Design Editor

Project Repository

Design Translator

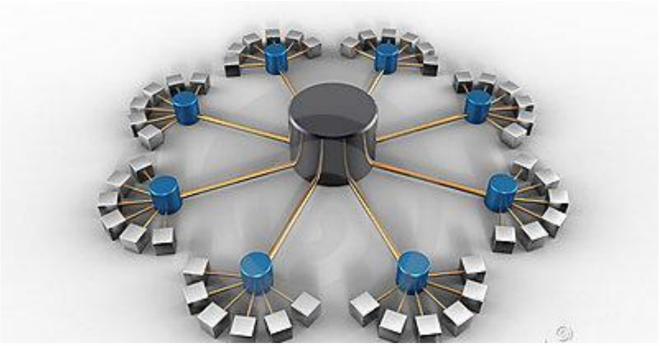
1. System Organization/Structuring



i) Repository Model

- 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.

1. System Organization/Structuring



i) Repository Model

- 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

1. System Organization/Structuring

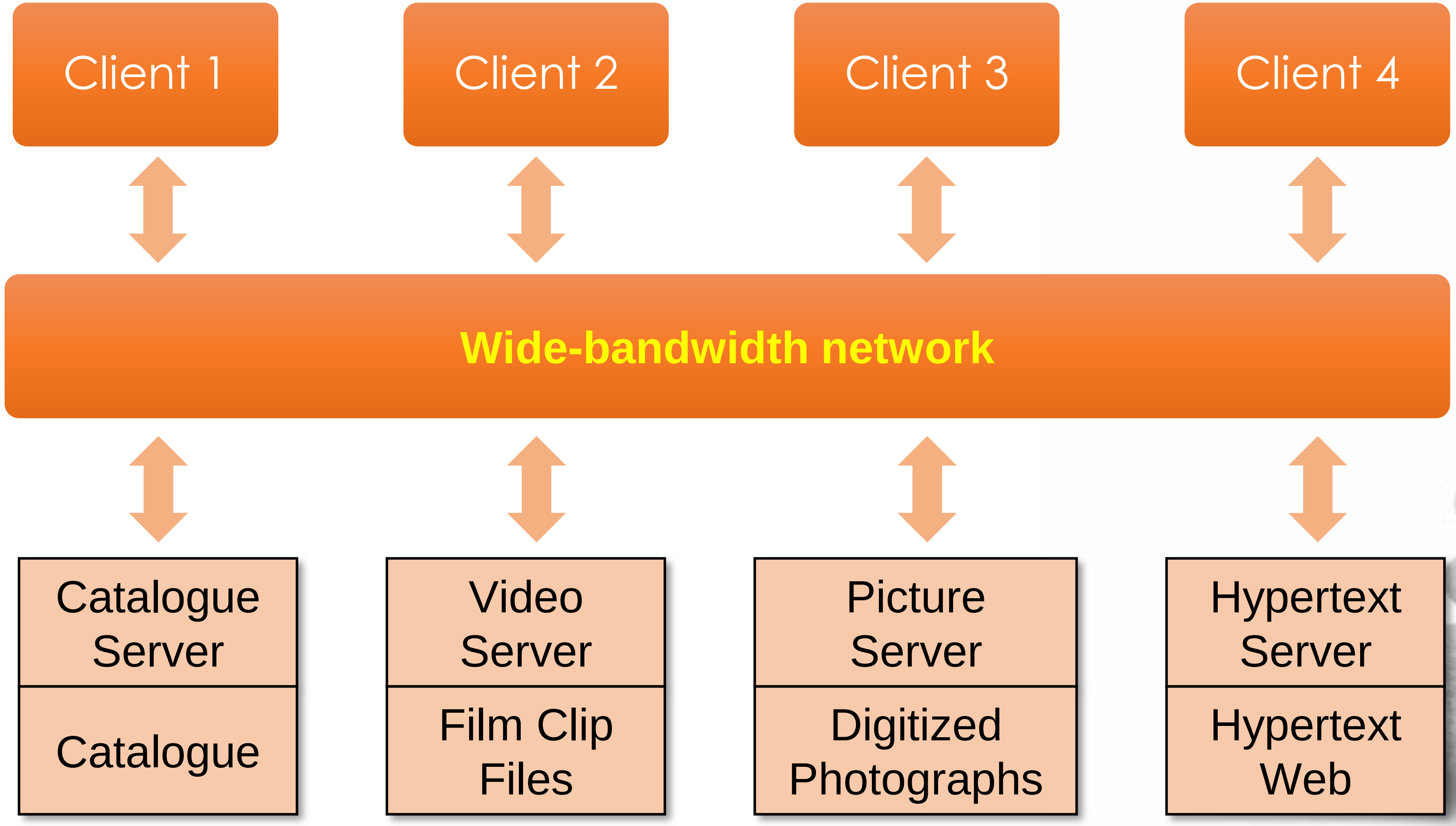
Client-Server Model

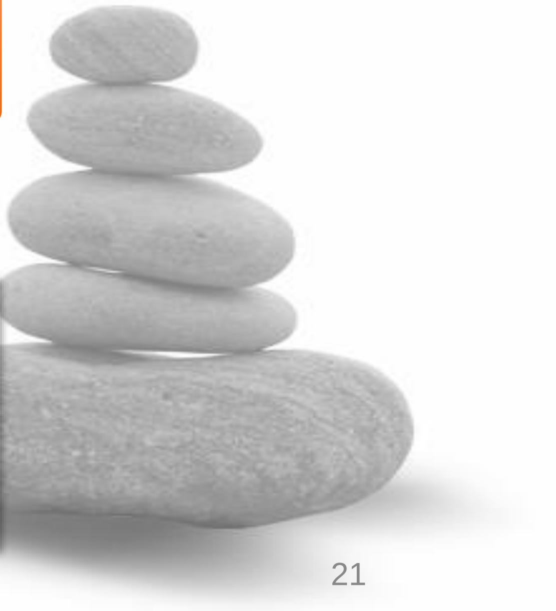
ii) Client-Server Model

- Distributed system model which shows how data and processing is distributed across a range of components.
- 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.

1. System Organization/Structuring

ii) Client-Server Model - The architecture of a film and picture library system





1. System Organization/Structuring

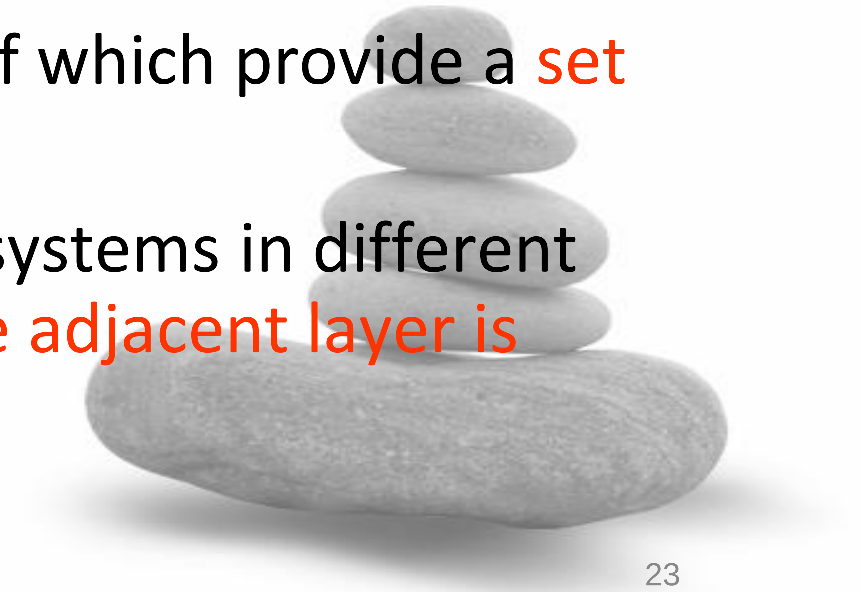
ii) Client-Server Model

- 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.

1. System Organization/Structuring

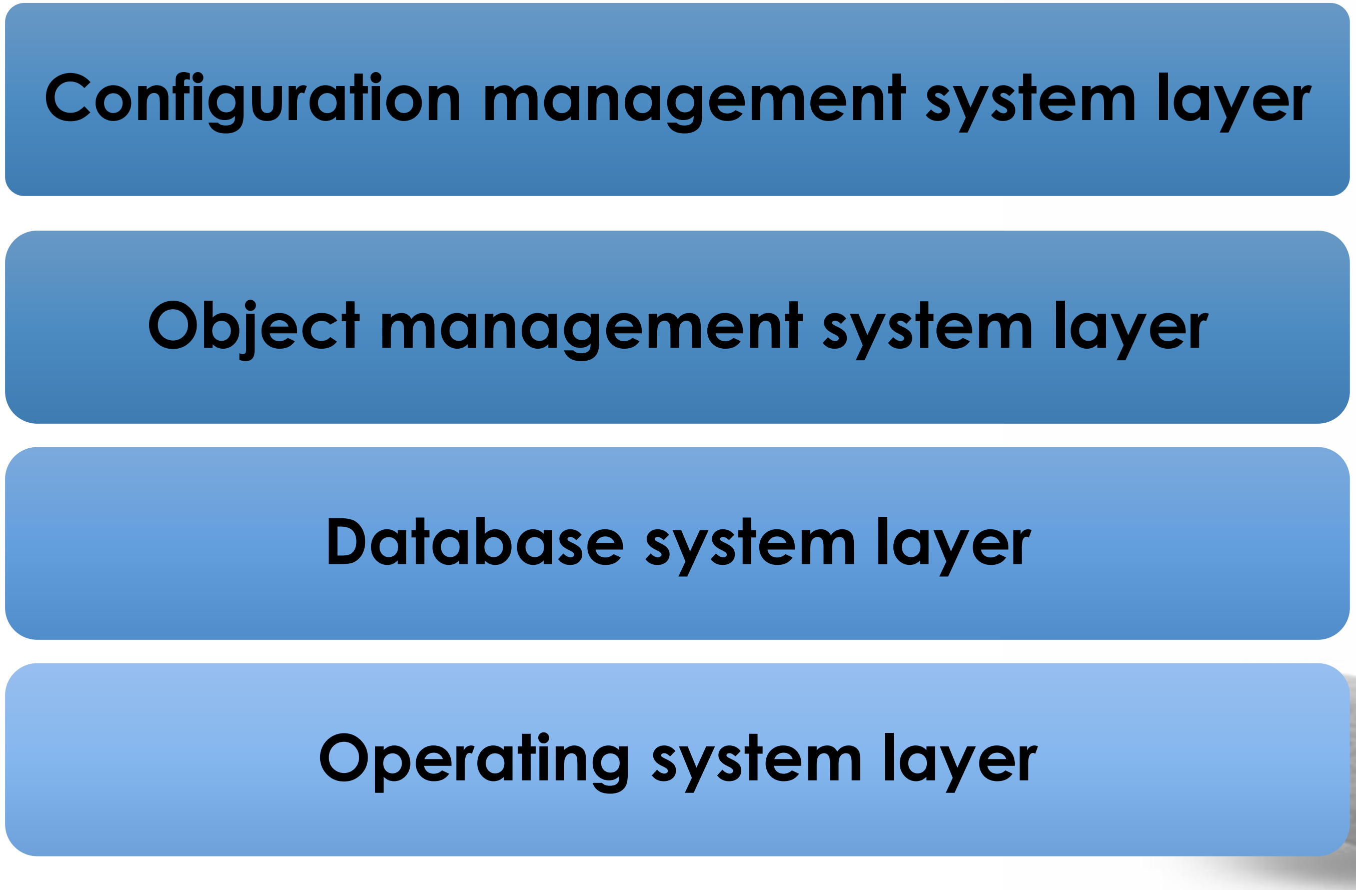
iii) Layered Model

- Also *called abstract machine model*
- Used to model the interfacing of sub-systems.
- Organises the system into a set of layers 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



1. System Organization/Structuring

iii) Layered Model – Version Management System



Configuration management system layer

Object management system layer

Database system layer

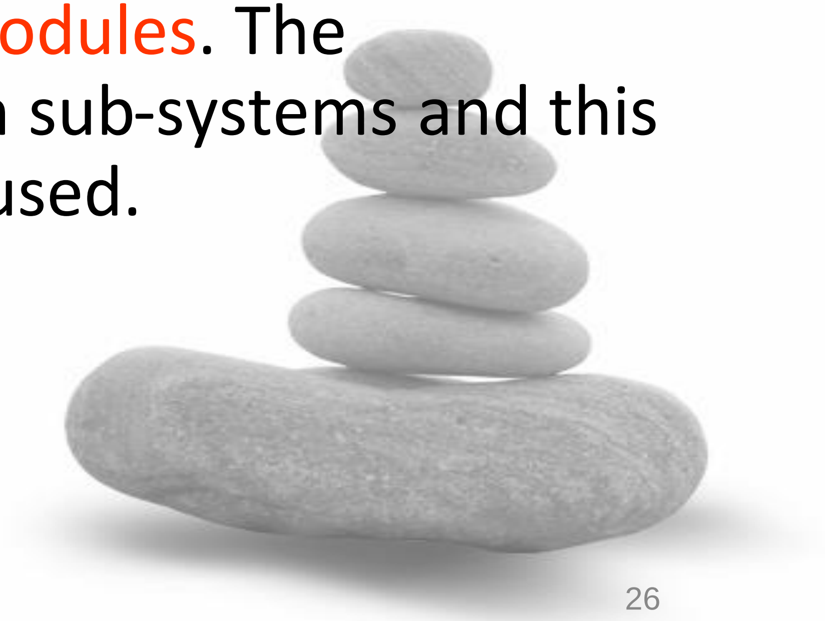
Operating system layer

2. Modular Decomposition

2. Modular Decomposition

- After an overall system organization has been chosen, you need to make a decision on the approaches to be used to decompose sub-systems into modules.

- This is the decomposition of sub-systems into modules. The components in modules are usually smaller than sub-systems and this allows alternative decomposition models to be used.



2. Modular Decomposition

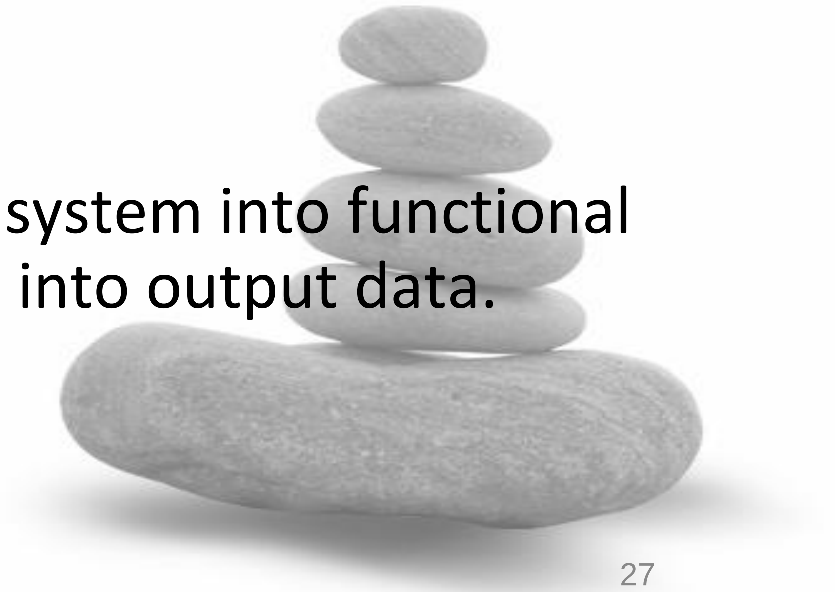
- 2 main strategies on decomposing a sub-system into modules:

i) **Object-oriented decomposition**

– an object model where decompose a system into a set of communicating objects

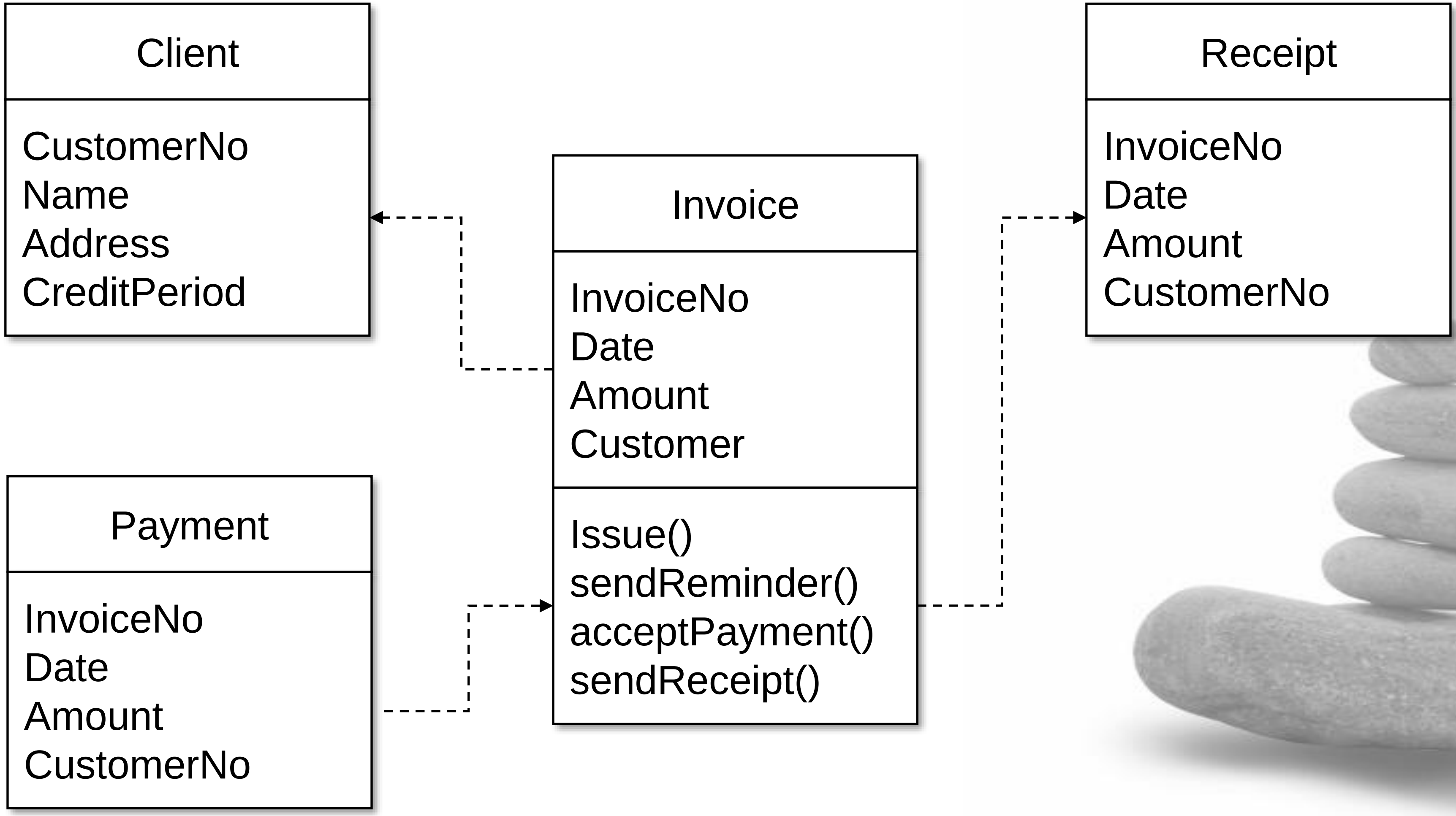
ii) **Function-oriented decomposition**

– an pipeline/ data-flow model where decompose a system into functional modules that accept input data and transform them into output data.



2. Modular Decomposition

Object-Oriented Decomposition – Invoice Processing System



Client

CustomerNo
Name
Address
CreditPeriod

Invoice

InvoiceNo
Date
Amount
Customer

Issue()
sendReminder()
acceptPayment()
sendReceipt()

Receipt

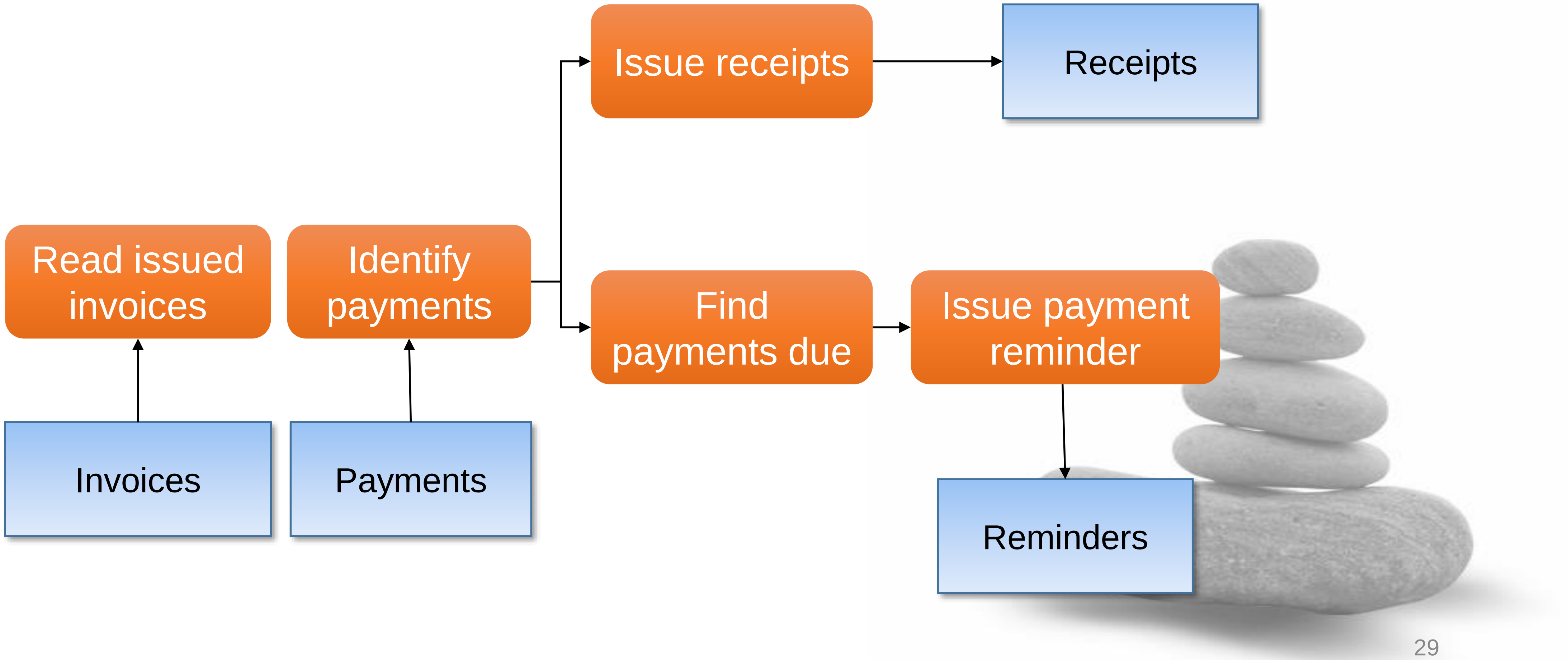
InvoiceNo
Date
Amount
CustomerNo

Payment

InvoiceNo
Date
Amount
CustomerNo

2. Modular Decomposition

Function-Oriented Decomposition – Invoice Processing System



3. Control Modeling

3. Control Modeling

- The models for structuring a system are concerned with how a system is decomposed into sub-systems. However, to work as a system, sub-systems must be controlled so that their services are delivered to the right place at the right time.
- Structural/ Organizational models do not include control information.
- Hence, a general model of the control relationships between the parts of the system is established
  - Concerned with the control flow between sub-systems

3. Control Modeling

i) Centralized control

- Call-return model
- Manager model

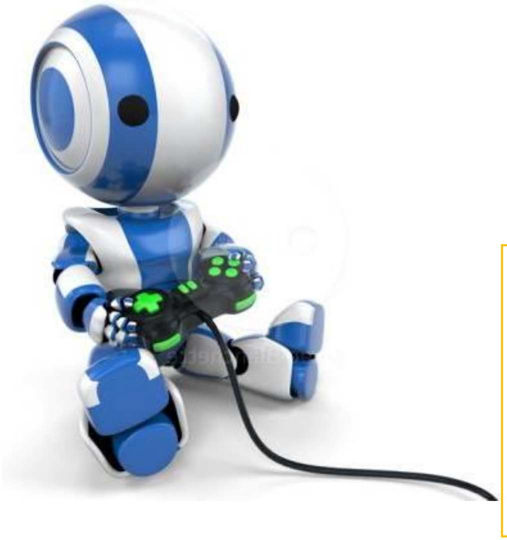
2 general control styles

ii) Event-based control

- Broadcast model
- Interrupt-driven model

3. Control Modeling

i) Centralized Control



One sub-system has overall responsibility for control and starts and stops other sub-systems.

3. Control Modeling

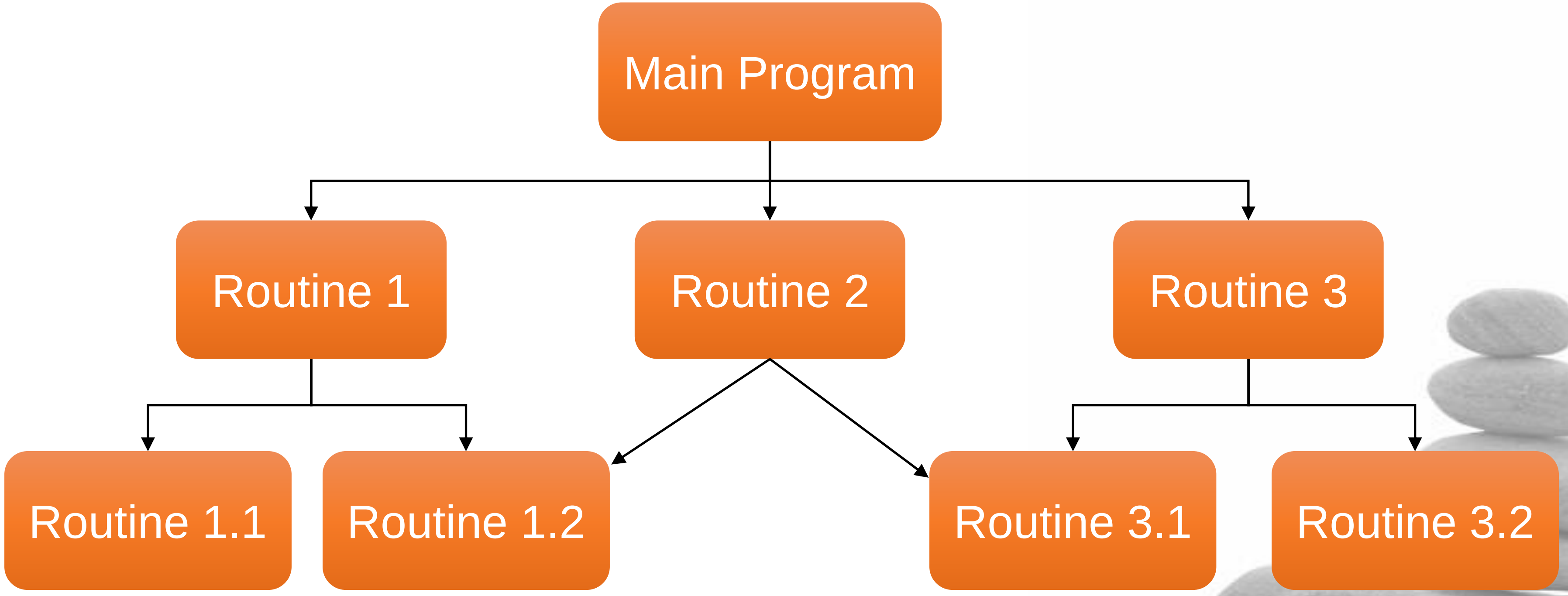
i) Centralized Control – Call-Return Model

Top-down subroutine model

Sequential systems

3. Control Modeling

i) Centralized Control – Call-Return Model



Main Program

Routine 1

Routine 2

Routine 3

Routine 1.1

Routine 1.2

Routine 3.1

Routine 3.2

□ top down sub-routine model

□ applicable for ***sequential systems***

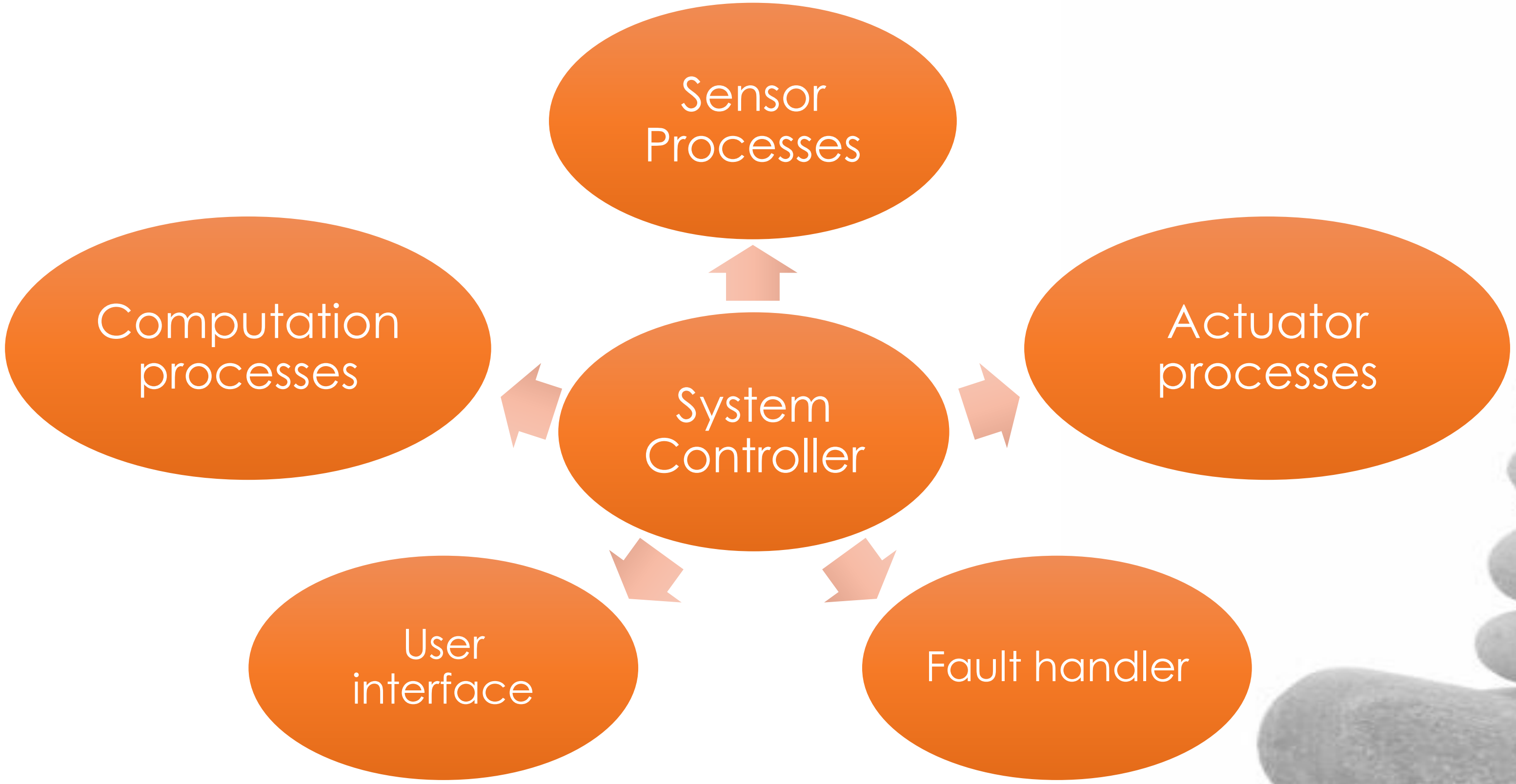
3. Control Modeling

i) Centralized Control – Manager Model

- One system component is designated as system manager i.e. controls the stopping, starting and coordination of other system processes.
- Applicable to concurrent systems.
- Can be implemented in sequential systems as a case statement.

3. Control Modeling

i) Centralized Control – Manager Model



- a system component is designated as system manager
- processes(sub-systems) can execute concurrently
- applicable for ***concurrent systems***

3. Control Modeling

ii) Event-based Control

- each sub-system can respond to externally generated events that might come from other sub-systems or environment.

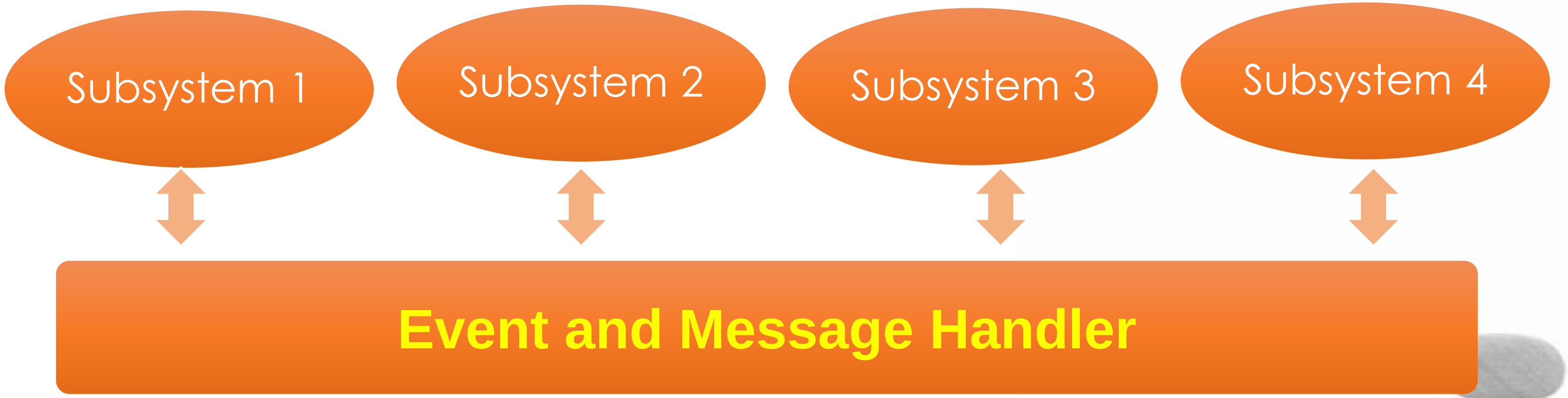
3. Control Modeling

ii) Event-based Control – 1. Broadcast Model

- An event is broadcast to all sub-systems. Any sub-system which can handle the event may respond to it.
- Effective in integrating sub-systems distributed across different computers on a network

3. Control Modeling

ii) Event-based Control – 1. Broadcast Model



Event	Subsystem
E1	SS1
E2	SS2
E3	SS3
E4	SS4

- □ an event is broadcast to all sub-systems
- □ effective in integrating sub-systems distributed across different computers on a network

3. Control Modeling

ii) Event-based Control –

2. Interrupt-Driven Model

- Used in real-time systems where interrupts are detected by an interrupt handler and passed to some other component for processing.

Revision

- System organization
    - Structure the system into main sub-systems & identify their communication. What are the Models?
- Modular Decomposition
    - Decompose sub-systems into modules. What are the models?
- Control Modeling
    - Establish control relationship between sub-systems/ modules. What are the Models?