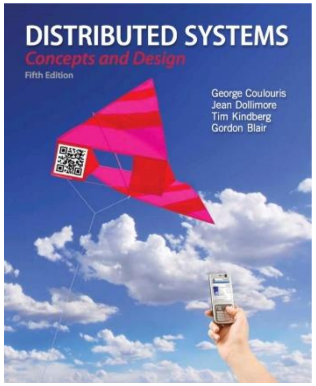
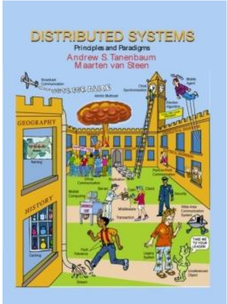
Distributed System Models



Most concepts are
drawn from Chapter 2
© Pearson Education

Dr. Rajkumar Buyya
Cloud Computing and **D**istributed **S**ystems (CLOUDS) Laboratory
Department of Computing and Information Systems
The University of Melbourne, Australia
http://www.cloudbus.org/652



Some ideas from Chapter 1
© Pearson Education

Outline

- Architectural Models
  - Introduction
  - Architectural elements
  - Architectural Patterns
    - Layering
    - Layering in distributed systems (platform, middleware)
    - Middleware types
    - Tiered architectures (two tier, three tier)
      - Example: Tiered architectures for web-based systems
    - J2EE tiered architectures

Outline

- Architectural Models
  - **Introduction**
  - Architectural elements
  - Architectural Patterns
    - Layering
    - Layering in distributed systems (platform, middleware)
    - Middleware types
    - Tiered architectures (two tier, three tier)
      - Example: Tiered architectures for web-based systems
    - J2EE tiered architectures

Architectural Models – Intro [1]

- The architecture of a system is its structure in terms of separately specified components.
  - Its goal is to meet present and likely future demands.
  - Major concerns are making the system reliable, manageable, adaptable, and cost-effective.
- Architectural Model:
  - Simplifies and abstracts the functions of individual components
  - The placement of the components across a network of computers – define patterns for the distribution of data and workloads
  - The interrelationship between the components – ie., functional roles and the patterns of communication between them.

Architectural Models

- The architecture of a system is its structure in terms of separately specified elements and their interrelationships.
- Hence, we will explain:
  - Architectural elements
  - Architectural patterns
  - Existing middleware platforms that support the above architectural elements and architectural patterns

Outline

- Architectural Models
  - Introduction
  - <u>**Architectural elements**</u>
  - Architectural Patterns
    - Layering
    - Layering in distributed systems (platform, middleware)
    - Middleware types
    - Tiered architectures (two tier, three tier)
      - Example: Tiered architectures for web-based systems
    - J2EE tiered architectures

Architectural Models – Architectural Elements

- What are the entities that are communicating in the distributed system?
  - Objects:
    - Objects have been introduced to enable and encourage the use of object-oriented approaches in distributed systems
    - Objects are accessed via interfaces, with an associated interface definition language (or IDL) providing a specification of the methods defined on an object.
    - Example?

Architectural Models – Architectural Elements

- What are the entities that are communicating in the distributed system?
  - Components
    - Problem-oriented abstractions for building distributed systems and are also accessed through interfaces
    - Assumptions?
    - Example?
  - Webservices
    - Web services represent an important paradigm for the development of distributed systems
    - Integrated into the world wide web

Architectural Models – Architectural Elements

- How do those entities communicate in a distributed system?
  - **Inter process communication** refers to the **relatively low-level support for communication** between processes in distributed systems, including message-passing primitives, direct access to the API offered by Internet protocols (Example ?)
  - **Remote invocation** covers a range of techniques based on a two-way exchange between communicating entities in a distributed system and **resulting in the calling of a remote operation, procedure or method** (Example ?)

Architectural Models – Architectural Elements

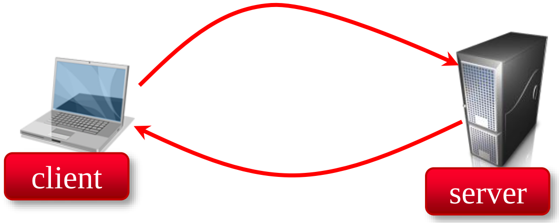
- How do those entities communicate in a distributed system?
  - **Indirect communication**
    - *Group communication*: Group communication is concerned with the delivery of messages to a set of recipients and hence is a multiparty communication paradigm supporting one-to-many communication.
    - *Publish-subscribe systems*
    - Message queues

Outline

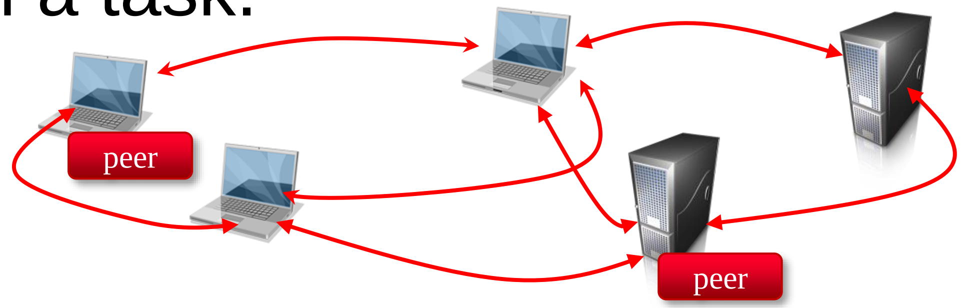
- Architectural Models
  - Introduction
  - Architectural elements
  - **Architectural Patterns**
    - Layering
    - Layering in distributed systems (platform, middleware)
    - Middleware types
    - Tiered architectures (two tier, three tier)
      - Example: Tiered architectures for web-based systems
    - J2EE tiered architectures

Architectural Models – Architectural Patterns

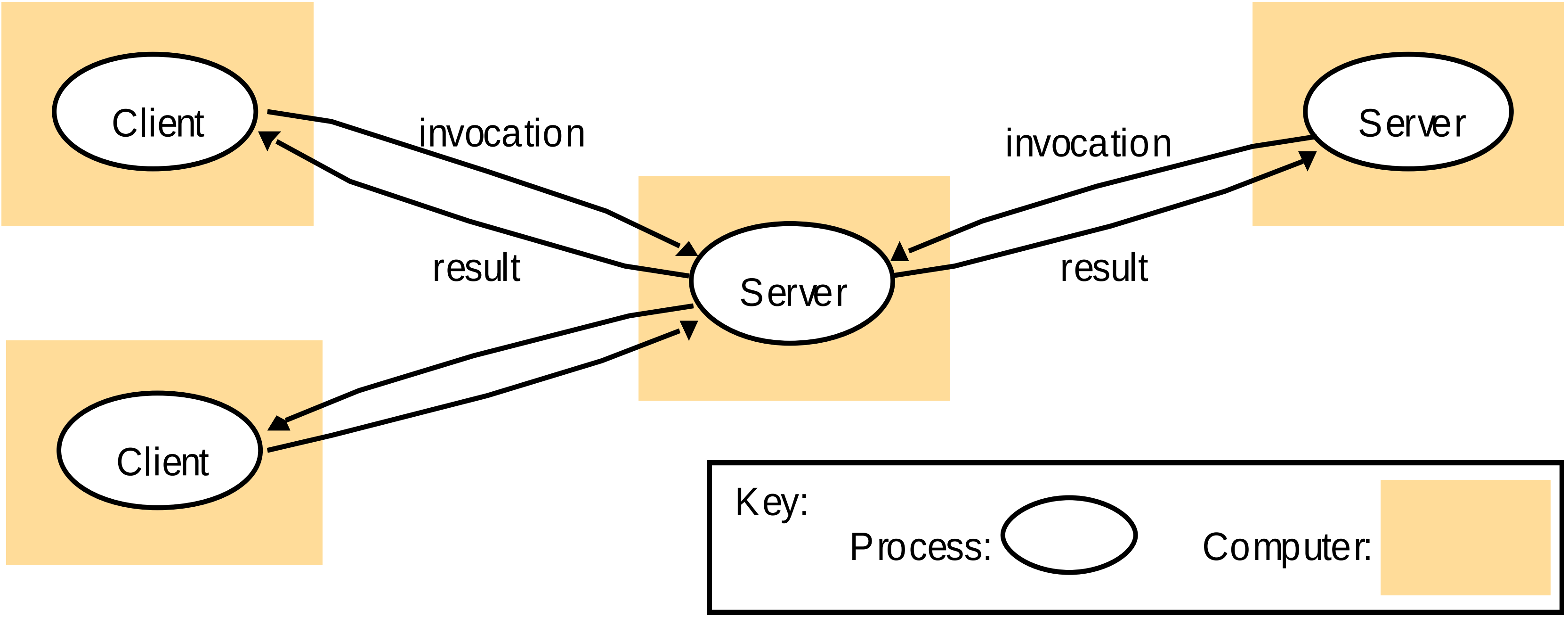
- One classification for architectural patterns stems from the role on processes within those patterns.
- An initial simplification is achieved by classifying processes as:
  - **Server processes**
  - **Client processes**
  - Peer processes
    - Cooperate and communicate in a symmetric manner to perform a task.



Dictionary server example.
Client-server or peer-to-peer?

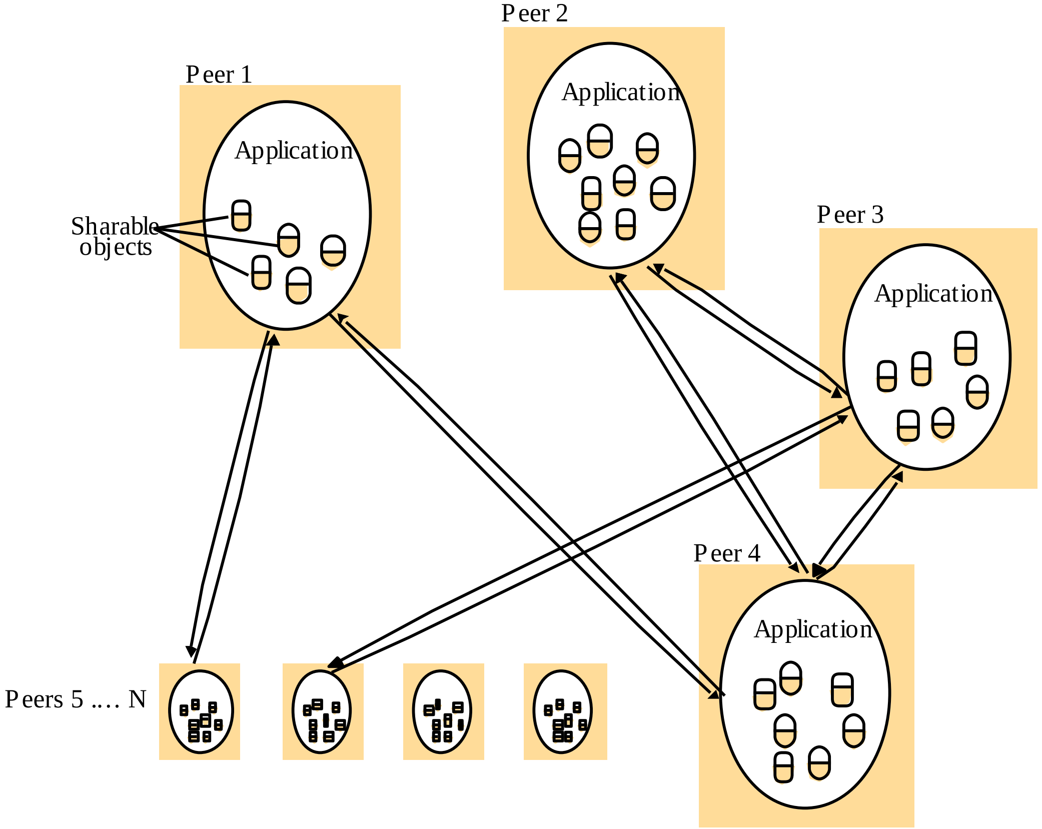


Client-Server Basic Model: Clients invoke individual servers



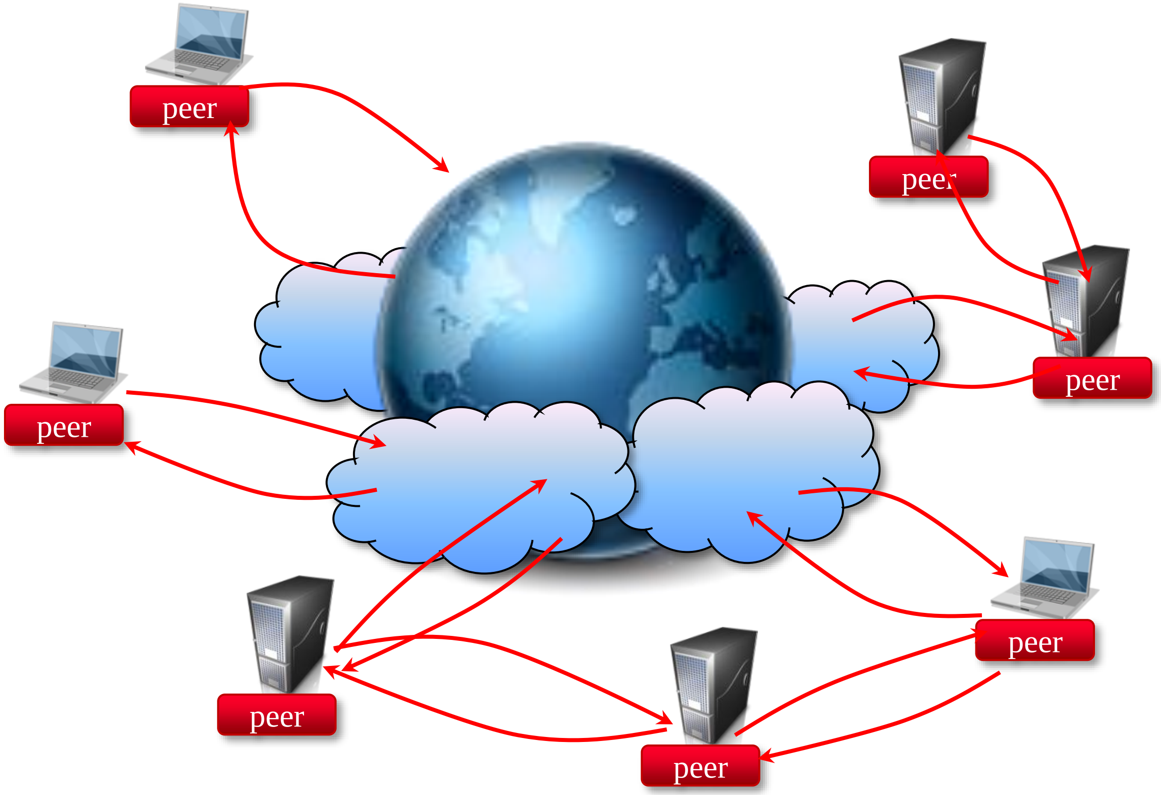
- Client processes interact with individual server processes in a separate computer in order to access data or resource. The server in turn may use services of other servers.
- Example?

Peer Processes: A distributed application based on peer processes



- **All of the processes play similar roles,** interacting cooperatively as peers to perform distributed activities or computations without distinction between clients and servers. (Examples?)
- Pros/Cons?

P2P with a Centralized Index Server (e.g. Napster Architecture)



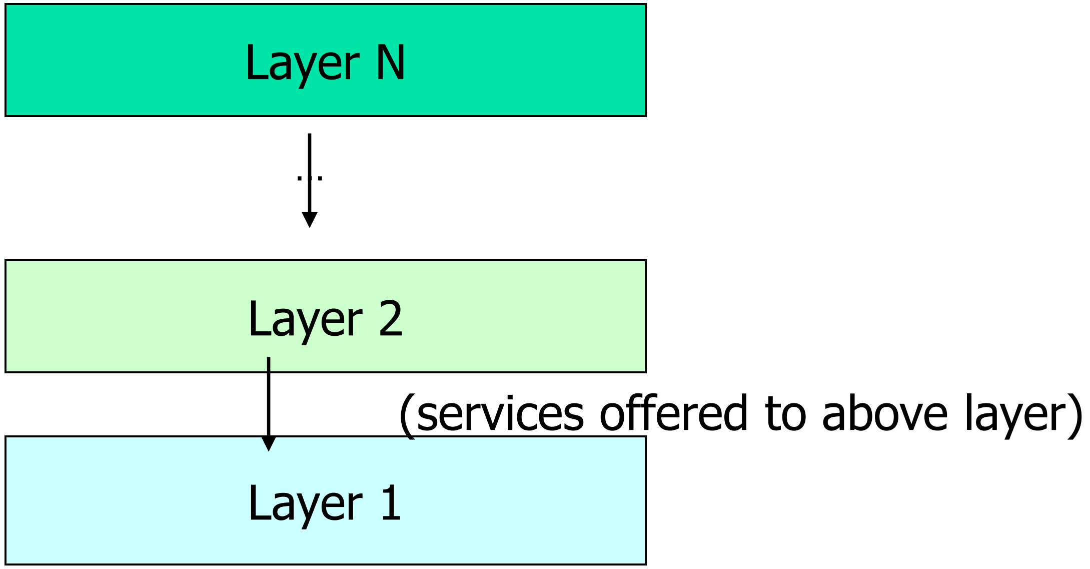
Outline

- Architectural Models
  - Introduction
  - Architectural elements
  - Architectural Patterns
    - **Layering**
    - Layering in distributed systems (platform, middleware)
    - Middleware types
    - Tiered architectures (two tier, three tier)
      - Example: Tiered architectures for web-based systems
    - J2EE tiered architectures

Architectural Patterns - Layering

- Breaking up the complexity of systems by designing them through layers and services
  - Layer: a group of related functional components
  - Service: functionality provided to the next layer.

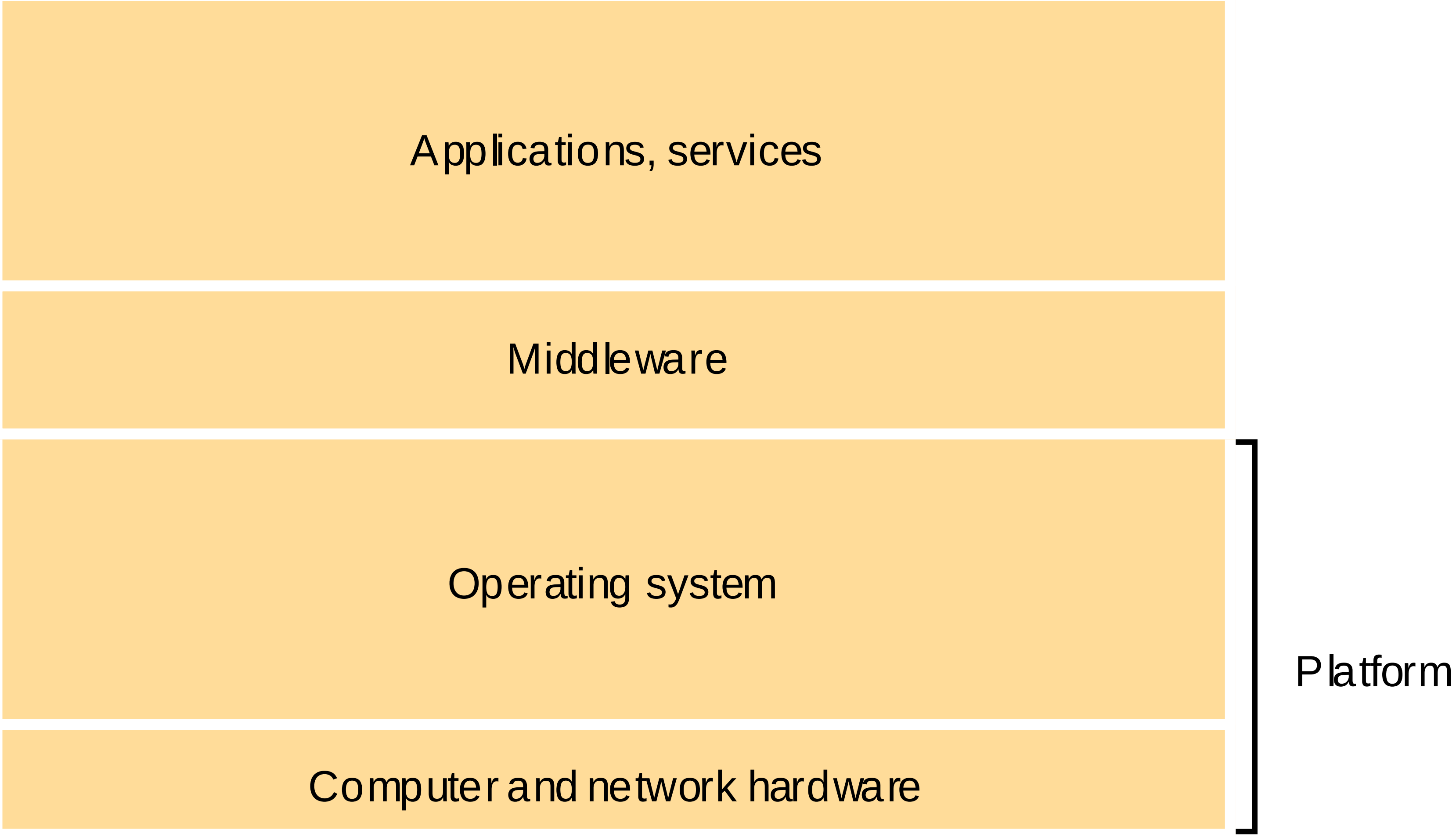
Lower level details awareness?



Outline

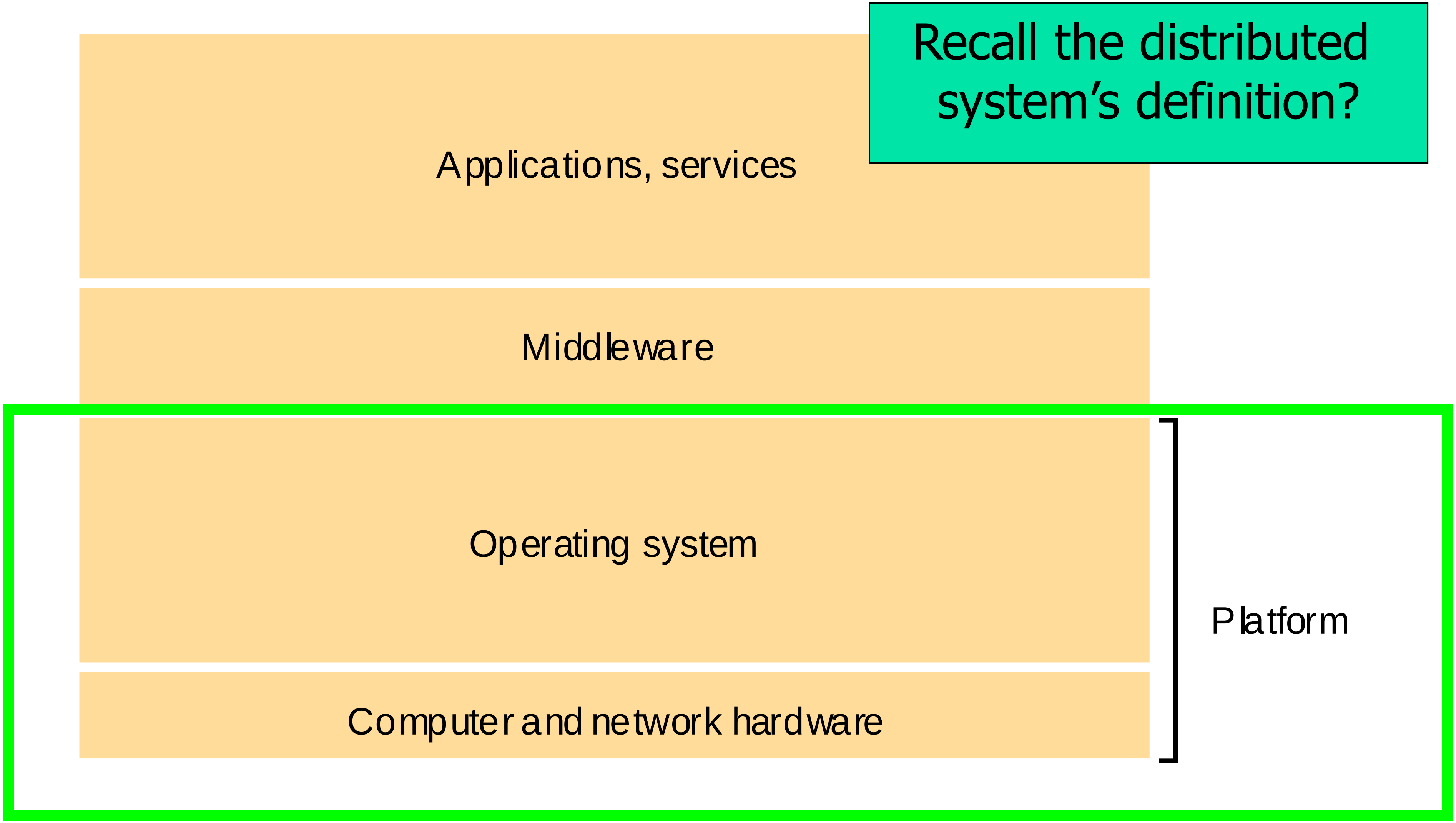
- Architectural Models
  - Introduction
  - Architectural elements
  - Architectural Patterns
    - Layering
    - **Layering in distributed systems (platform, middleware)**
    - Middleware types
    - Tiered architectures (two tier, three tier)
      - Example: Tiered architectures for web-based systems
    - J2EE tiered architectures

Software and hardware service layers in distributed systems



Software and hardware service layers in distributed systems

Recall the distributed system’s definition?



Applications, services

Middleware

Operating system

Computer and network hardware

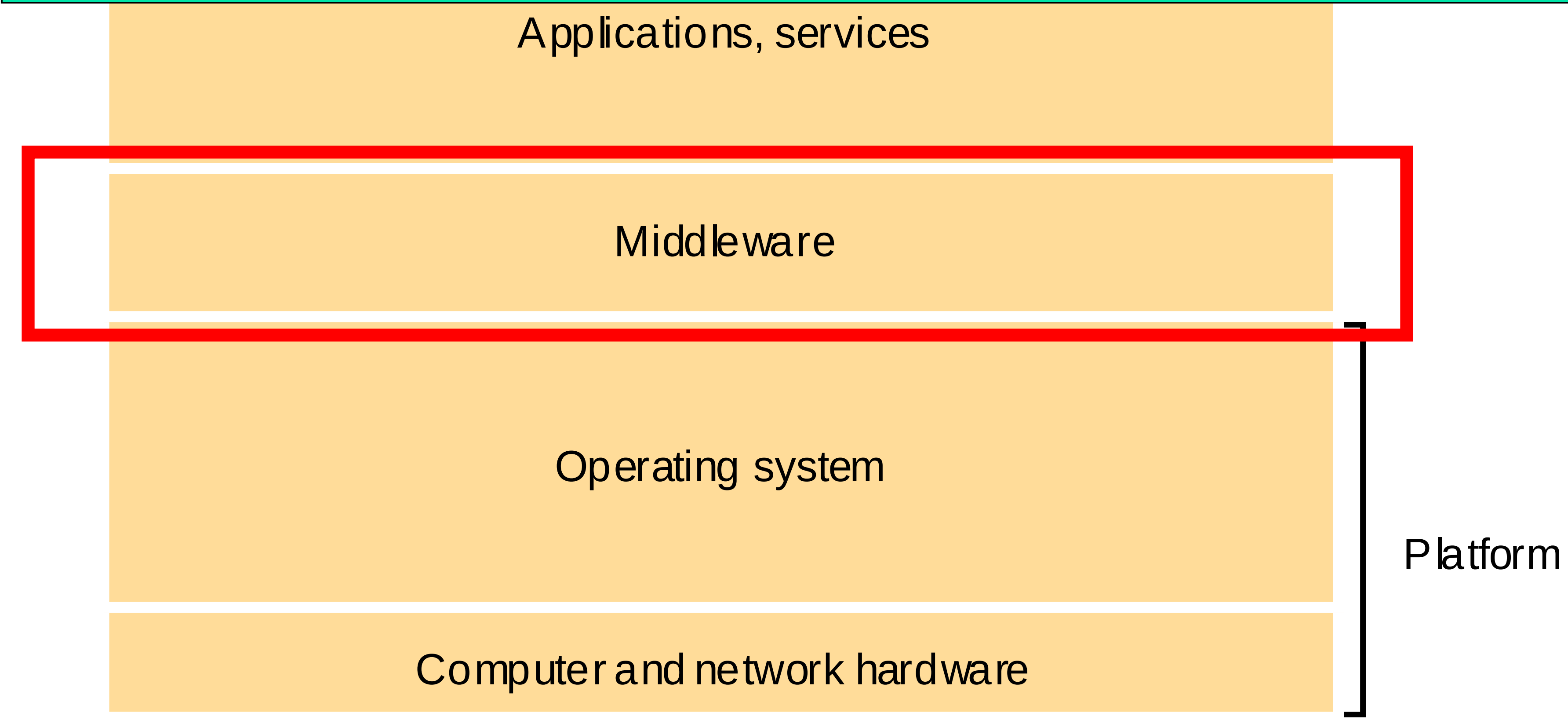
Platform

Platform

- The lowest hardware and software layers are often referred to as a platform for distributed systems and applications.
- These low-level layers provide services to the layers above them, which are implemented independently in each computer.
- Major Examples
  - Intel x86/Windows
  - Intel x86/Linux
  - Intel x86/Solaris
  - PowerPC/MacOS
  - iPhone/iOS
  - Samsung Galaxy/Android

Software and hardware service layers in distributed systems

A layer of software whose purpose is to **mask heterogeneity** and to provide a convenient programming model to application programmers



Applications, services

Middleware

Operating system

Computer and network hardware

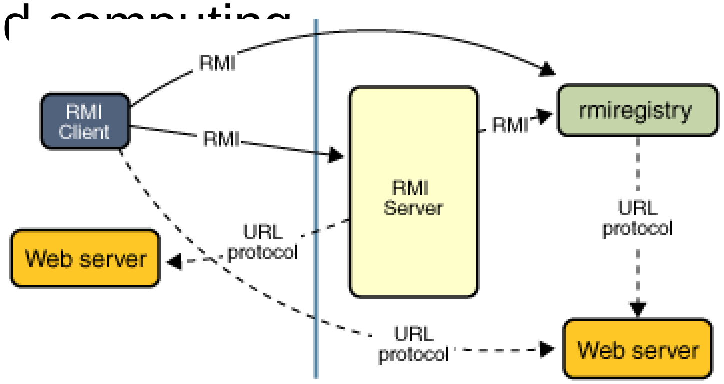
Platform

Middleware Prior Options

- What were the prior middleware options to get heterogenous machines to communicate?
  - E.g., a Windows machine and a Linux machine
  - What is we add a Solaris machine

Middleware

- A layer of software whose purpose is to mask heterogeneity present in distributed systems and to provide a convenient programming model to application developers.
- **Major Examples:**
  - **Sun RPC (Remote Procedure Call)**
  - OMG CORBA (Common Object Request Broker Architecture)
  - Microsoft D-COM (Distributed Components Object Model)
  - **Sun Java RMI (Remote Method Invocation)**
  - Modern Middleware Examples:
    - Manjrasoft Aneka– for Cloud computing
    - IBM WebSphere
    - Microsoft .NET
    - **Sun J2EE**
    - Google AppEngine
    - Microsoft Azure

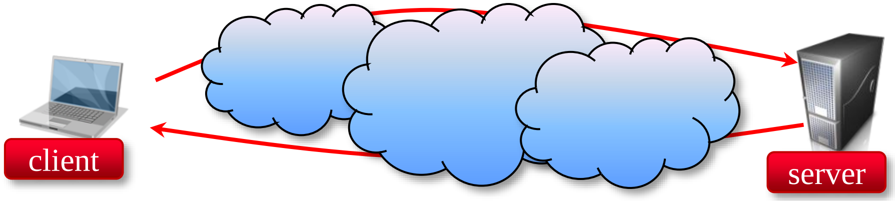


System Architecture

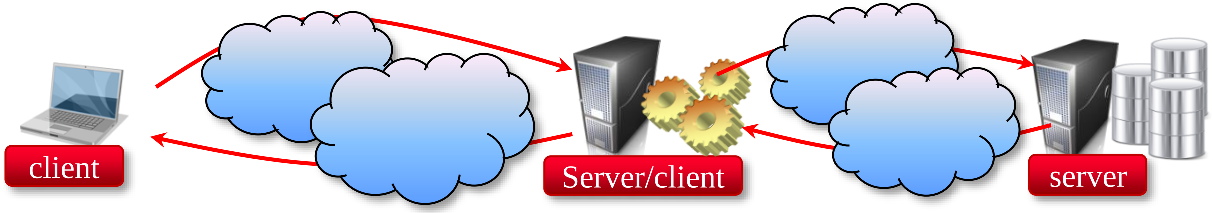
- The most evident aspect of DS design is the division of responsibilities between system components (applications, servers, and other processes) and the placement of the components on computers in the network.
- It has major implication for:
  - Performance, reliability, and security of the resulting system.

Client-Server Architecture Types
(Tier arch compliments layer architecture)

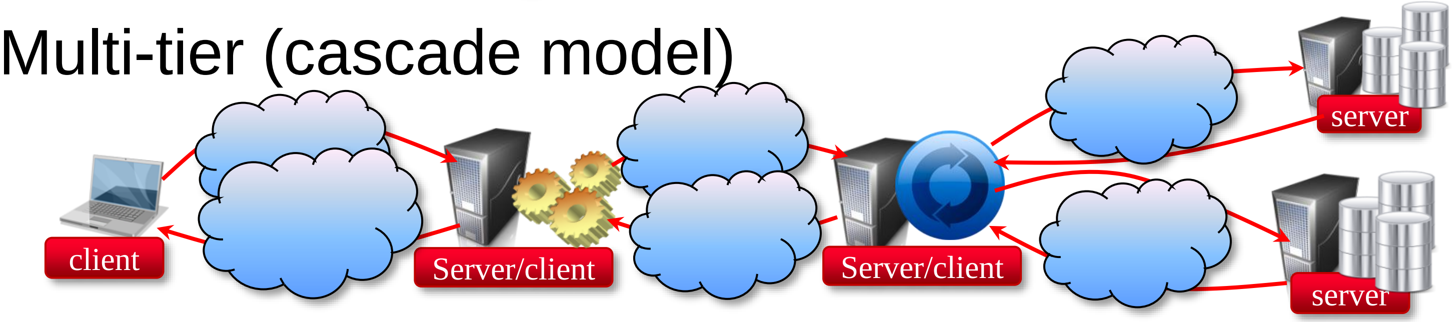
- Two-tier model (classic)



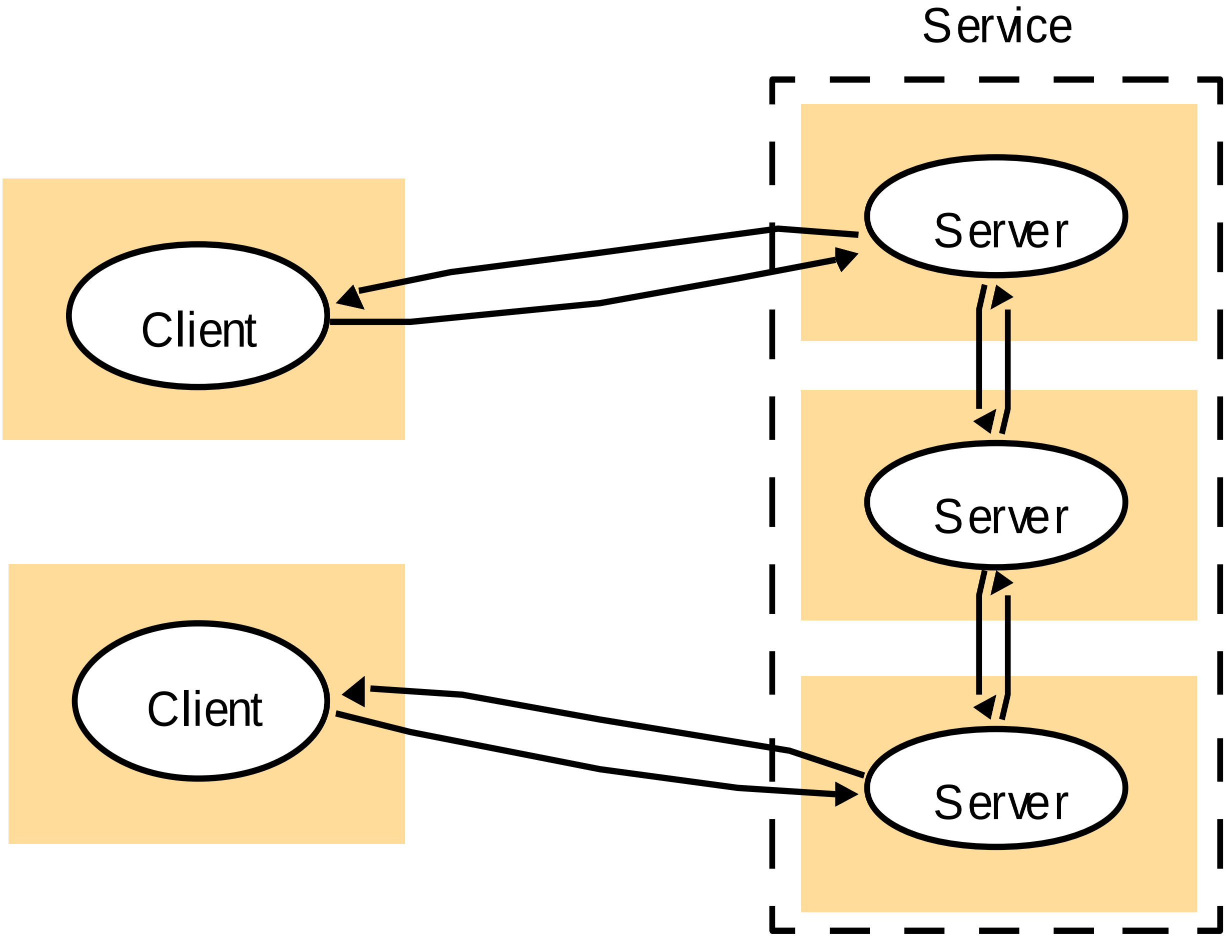
- Three-tier (when the server, becomes a client)



- Multi-tier (cascade model)



A service provided by multiple servers



- Services may be implemented as several server processes in separate host computers.
- Example: Cluster based Web servers and apps such as Google, parallel databases Oracle

Outline

- Architectural Models
  - Introduction
  - Architectural elements
  - Architectural Patterns
    - Layering
    - Layering in distributed systems (platform, middleware)
    - Middleware types
    - **Tiered architectures (two tier, three tier)**
      - Example: Tiered architectures for web-based systems
    - J2EE tiered architectures

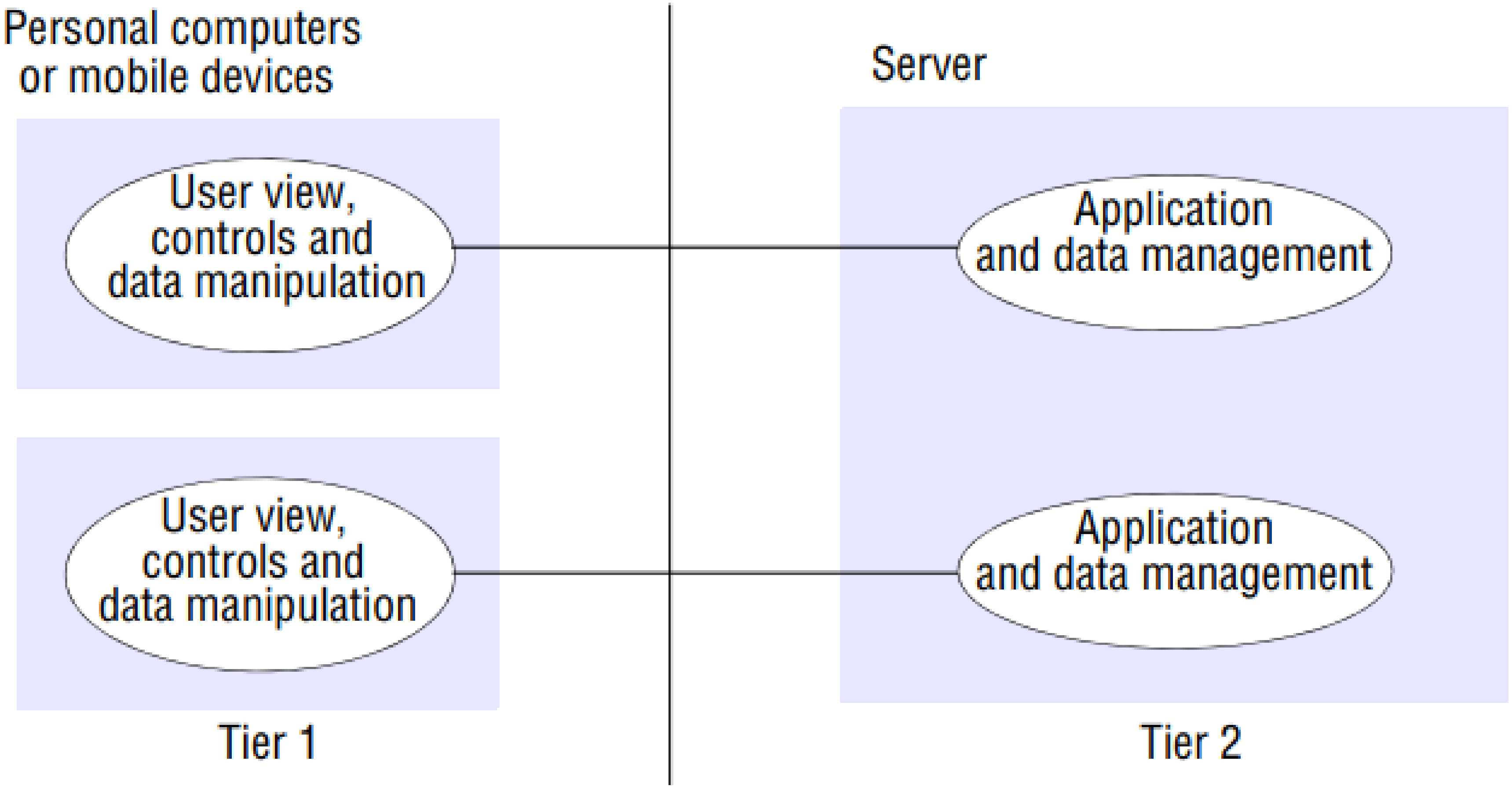
Architectural Patterns – Tiered Architecture

- Tiered architectures are complementary to layering (how?)
- Mostly associated with the organization of applications and services within the prior Layering figure
- Applies to all layers of a distributed system’s architecture.

Architectural Patterns – Tiered Architecture

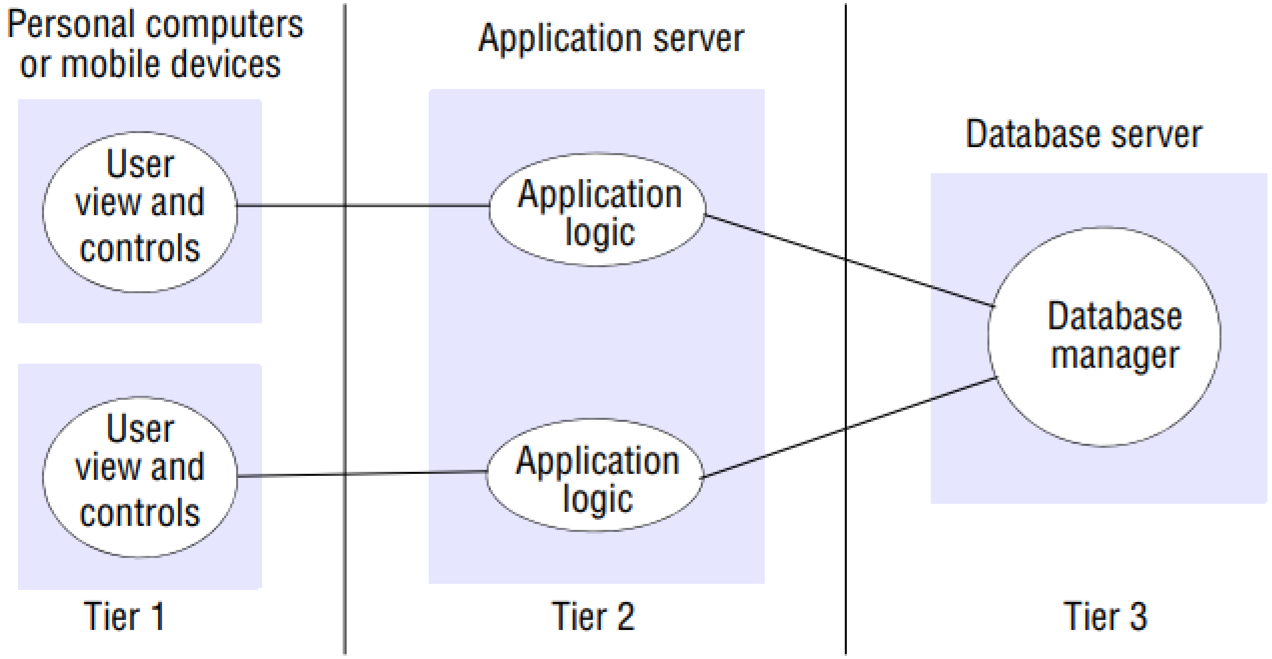
- Consider the functional decomposition of a given application, as follows:
    - the presentation logic
    - the application logic
    - the data logic

Architectural Patterns – Two Tiered Architecture


Personal computers or mobile devices
Server
User view, controls and data manipulation
User view, controls and data manipulation
Application and data management
Application and data management
Tier 1
Tier 2

- Pros/Cons
- Where would the middleware fit here (e.g., RMI)?

Architectural Patterns – Three Tiered Architecture



- Pros/Cons
- Where would the middleware fit here (e.g., RMI)?

Required Readings

- Chapter 2 from the text book: Coulouris, George F., Jean Dollimore, and Tim Kindberg. *Distributed systems: concepts and design*. Fifth Edition.