Information Processing Techniques

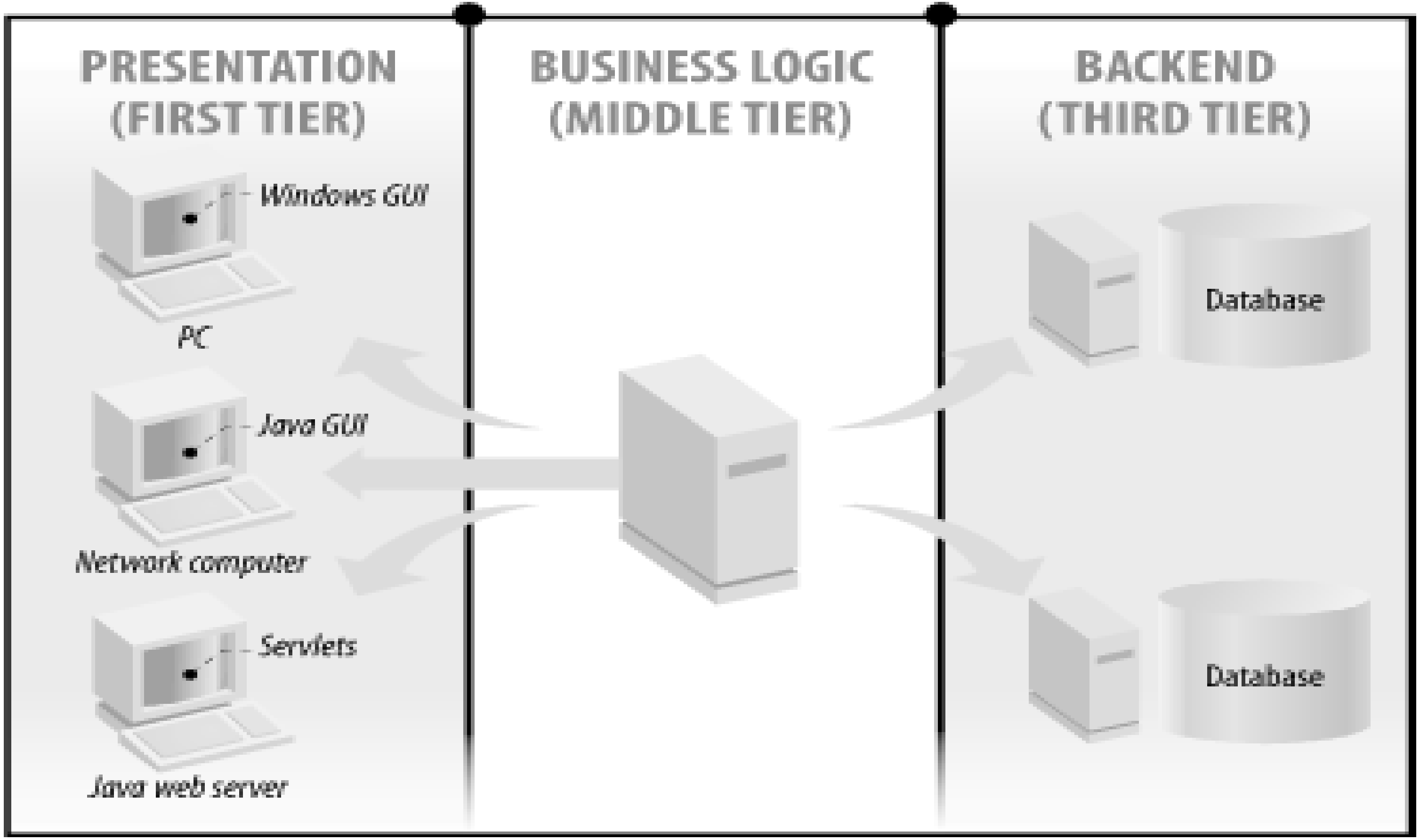
Week 07

MURTAZA MUNAWAR FAZAL

Distributed Systems

- Virtually all large computer-based systems are now distributed systems.
- Information processing is distributed over several computers rather than confined to a single machine.
- Different distributed systems architectures
  - client-server
  - peer-to-peer
  - parallel/cluster/grid computing
  - distributed object architectures -CORBA; .NET; J2EE
  - service-oriented architectures

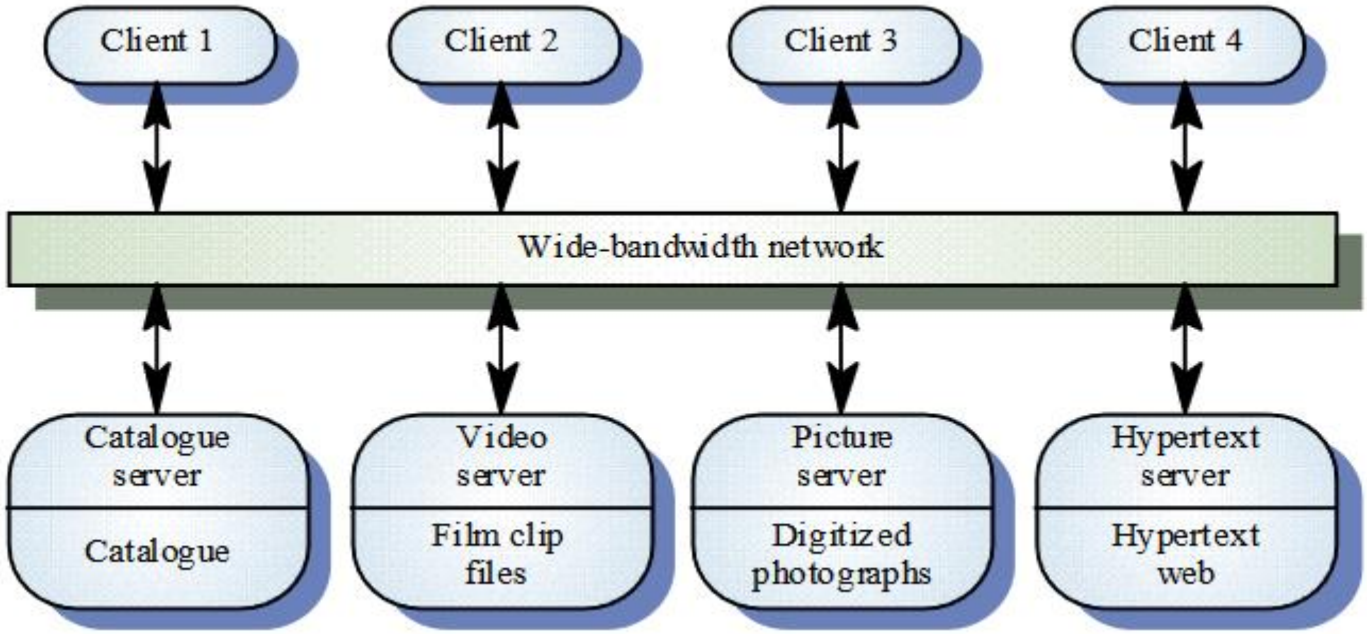
Distributed Object Architectures



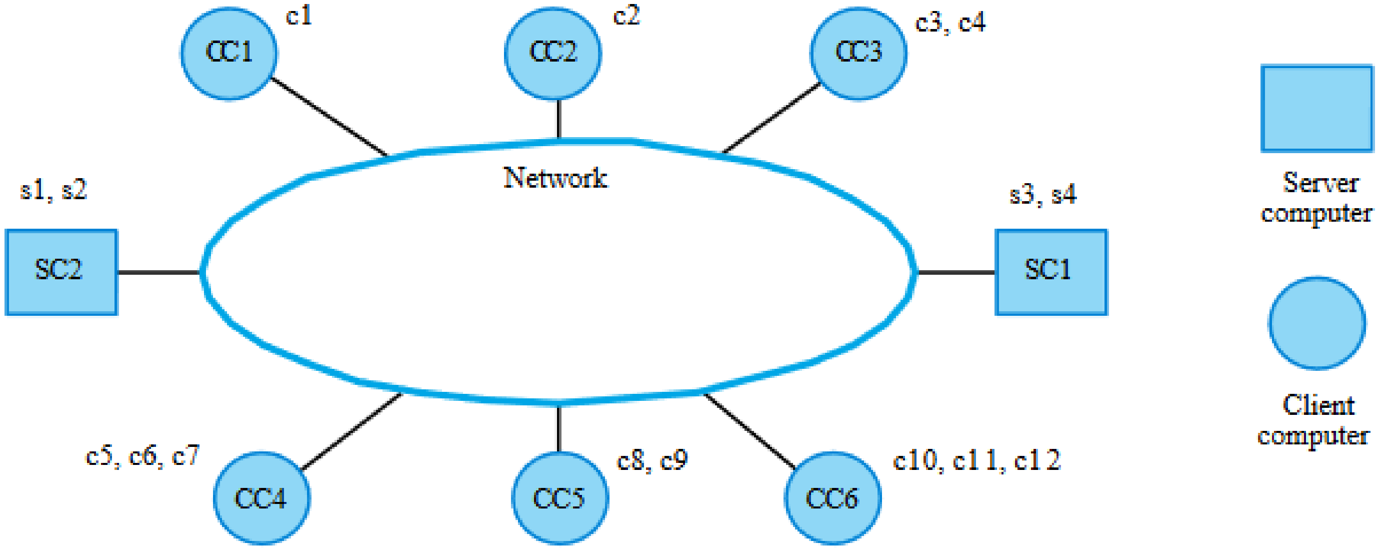
Client-server architecture

- Distributed system model where data and processing is distributed across a range of components
- Set of 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
- Clients know of servers but servers need not know of clients

Example: Film and picture library



Computers in a C/S network



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 organization-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

Middleware

- Software that manages and supports the different components of a distributed system. In essence, it sits in the middle of the system.
- Middleware can be off-the-shelf rather than specially written software
- Examples
  - Transaction processing monitors;
  - Data converters;
  - Communication controllers.

Layered application architecture

- Presentation layer
  - Concerned with presenting the results of a computation to system users and with collecting user inputs
- Application processing layer
  - Concerned with providing application specific functionality e.g., in a banking system, banking functions such as open account, close account, etc.
- Data management layer
  - Concerned with managing the system databases

Tiering

- A two-tier architecture is one where a client talks directly to a server, with no intervening server
  - This type of architecture is typically used in small environments with less than 50 users
- A three-tier architecture introduces another server (or an "agent") between the client and the server
  - The role of the middle-tier agent is many-fold - it can provide translation services as in adapting a legacy application on a mainframe to a client/server environnent
  - A plethora of software technologies have evolved to fill the middle tier - middleware

Thin and fat clients

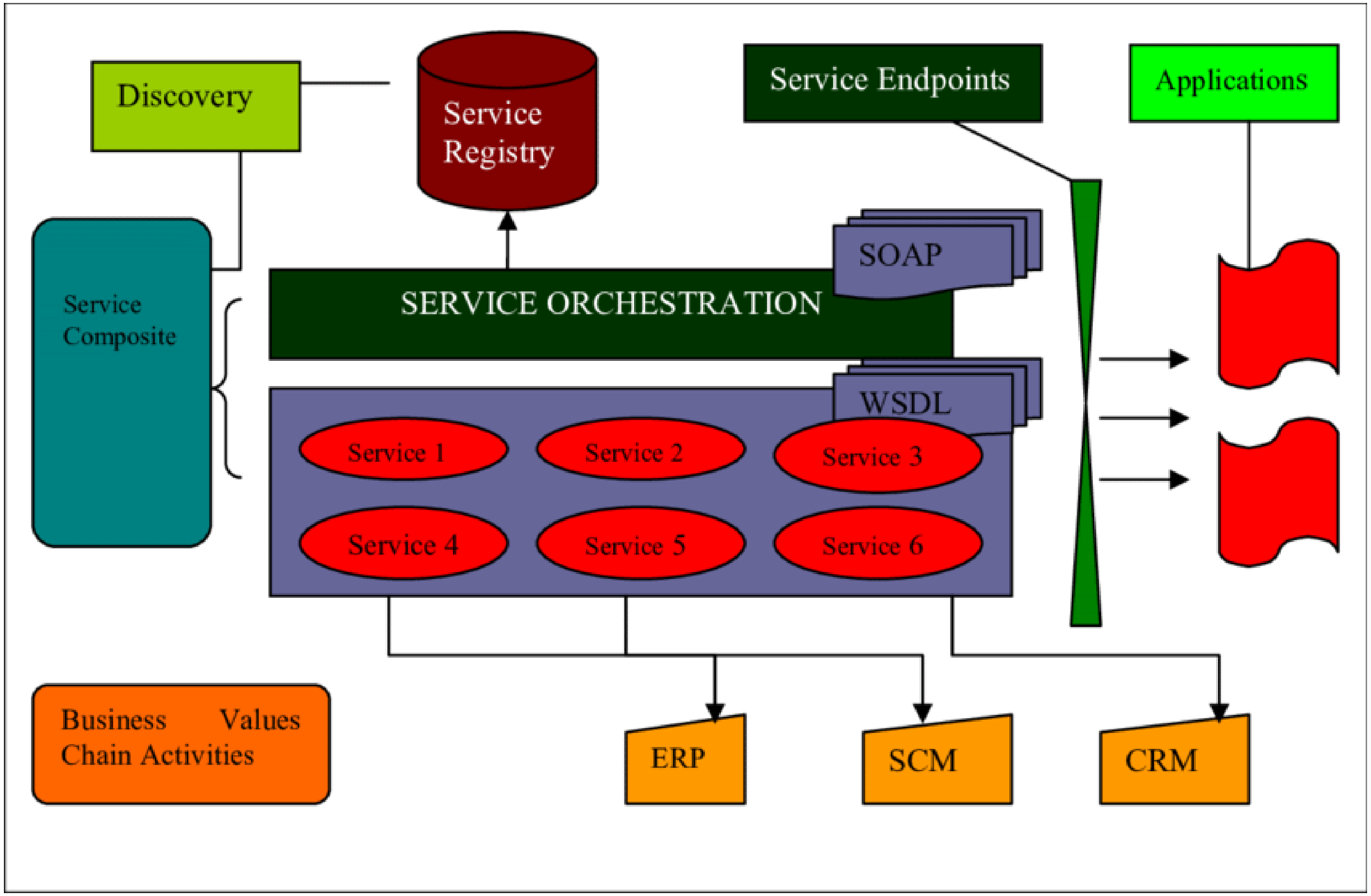
- Thin-client model
  - all of the application processing and data management is carried out on the server
  - client is simply responsible for running the presentation software
- Fat-client model
  - the server is only responsible for data management.
  - software on the client implements the application logic and the interactions with the system user

Scaling

Environment is typically heterogeneous and multi-vendor

- Client-server systems can be scaled horizontally or vertically
  - Horizontal scaling means adding or removing client workstations with only a slight performance impact. (Adding more machines)
  - Vertical scaling means migrating to a larger and faster server machine or multi servers. (adding hardware resources to the same machines)
- The front-end task and back-end task have fundamentally different requirements for computing resources
  - servers typically expensive, fast, multi-processor systems, with large storage capacity
  - clients less expensive

Service Oriented Architecture


Discovery
Service Registry
Service Endpoints
Applications
SOAP
Service Composite
SERVICE ORCHESTRATION
WSDL
Service 1
Service 2
Service 3
Service 4
Service 5
Service 6
Business Values Chain Activities
ERP
SCM
CRM