
Lec 10 – Client Server

Framework



- ✧ To understand client-server architecture
 - 2-tier
 - 3-tier

- ✧ To differentiate between thin and fat clients

Framework



- ✧ The application is modeled as a set of services that are provided by servers and a set of clients that use these services.
- ✧ Clients know of servers but servers need not know of clients.



Client-Server architecture: Examples

- ✧ **Clients**: web browsers, email clients, and online chat clients.

- ✧ **Servers**: web servers, ftp servers, application servers, database servers, name servers, mail servers, file servers, print servers, and terminal servers.
 - Most web services are also types of servers

Application layers

✧ Presentation layer

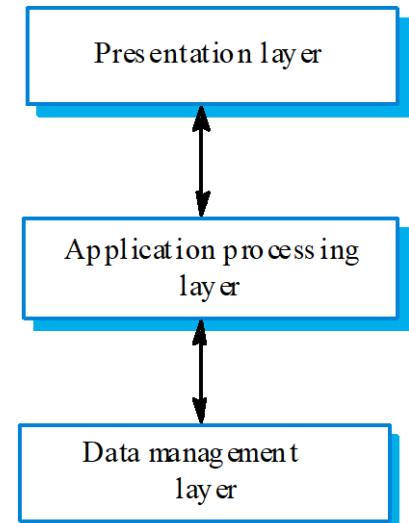
- Concerned with presenting the results of a computation to system users and with collecting user inputs.

✧ 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

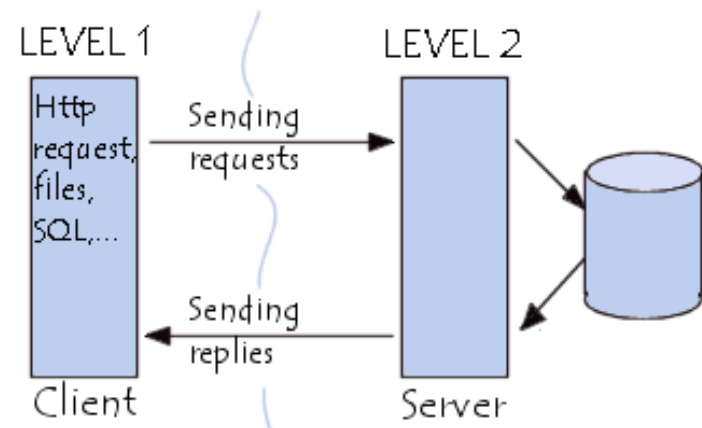
- Concerned with managing the system databases.



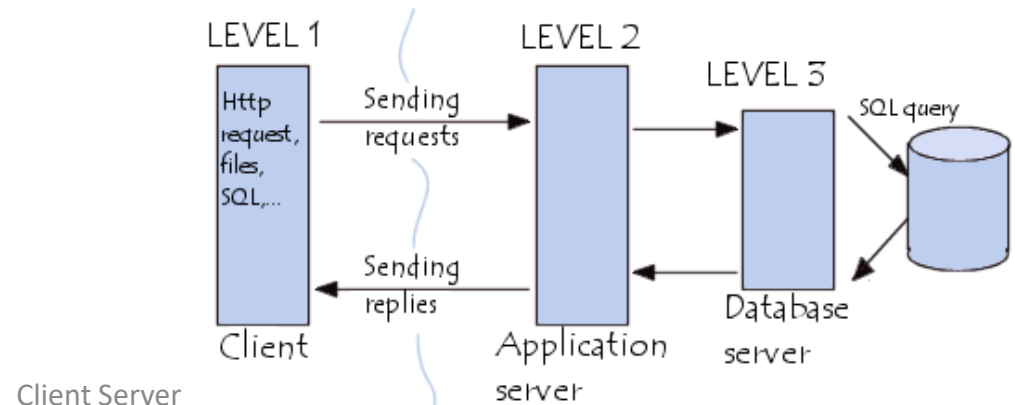
Client-server architectures: Types

✧ There are basically two types of client-server architectures

- Two tier architectures

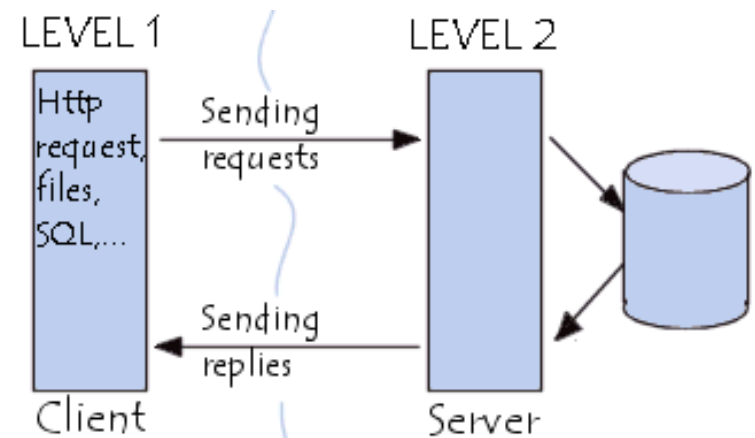


- Three tier architectures



Two tier architecture

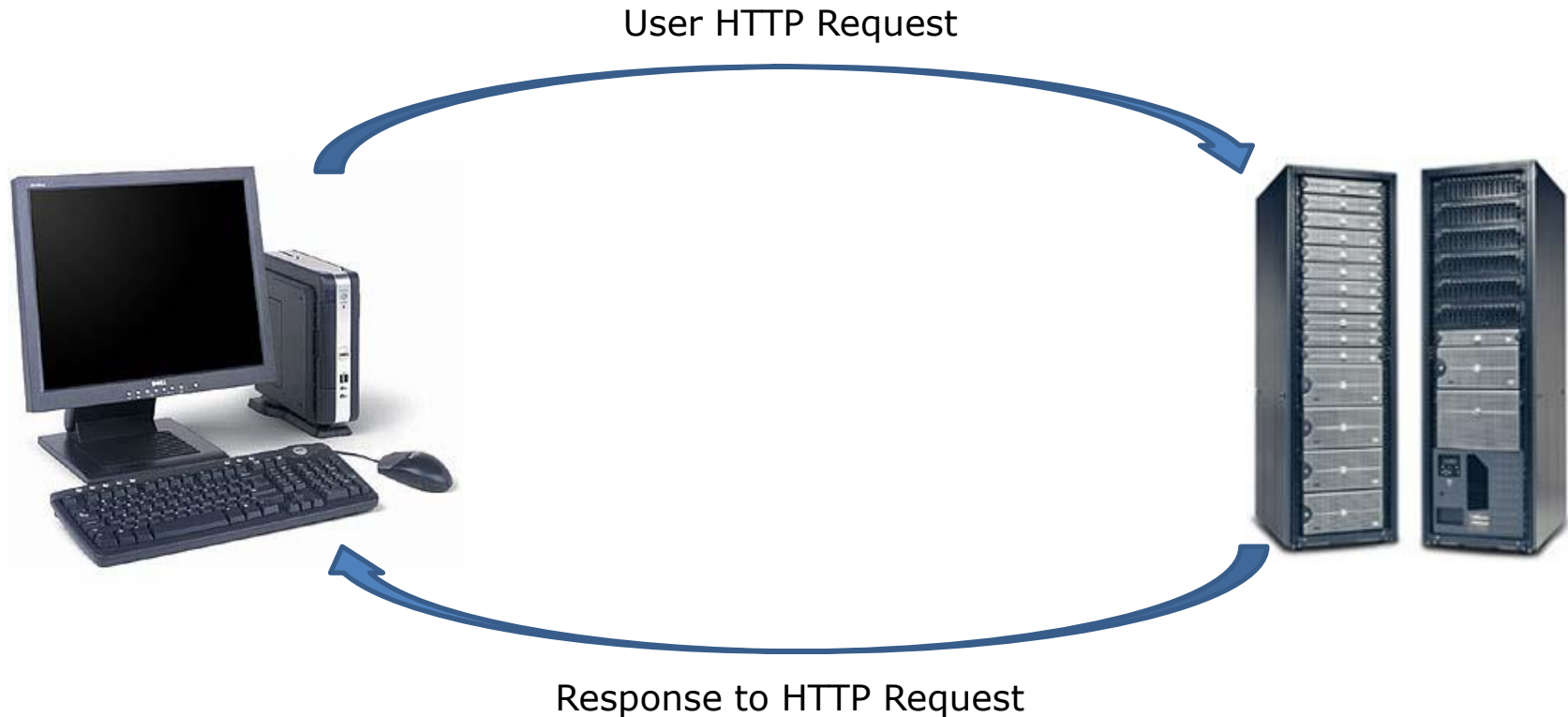
- ✧ Application components are distributed between the server and client software
- ✧ In addition to part of the application software, the server also stores the data, and all data accesses are through the server.
- ✧ The presentation (to the user) is handled strictly by the client software.
- ✧ The clients assume the bulk of the responsibility for the application logic.



Two tier architecture (cont...)

- ✧ The server assumes the bulk of the responsibility for data integrity checks, query capabilities, data extraction and most of the data intensive tasks, including sending the appropriate data to the appropriate clients.
- ✧ SQL is used on the clients to request appropriate subsets of data from the server.
- ✧ Data returned from the server to the clients is manipulated by the client software for reporting.

Example: HTTP Requests



Processing of **HTML code** takes place on the **client side**
and the **web page request** is processed on the **server side**



Two tier architecture (Cont.)

✧ Advantages

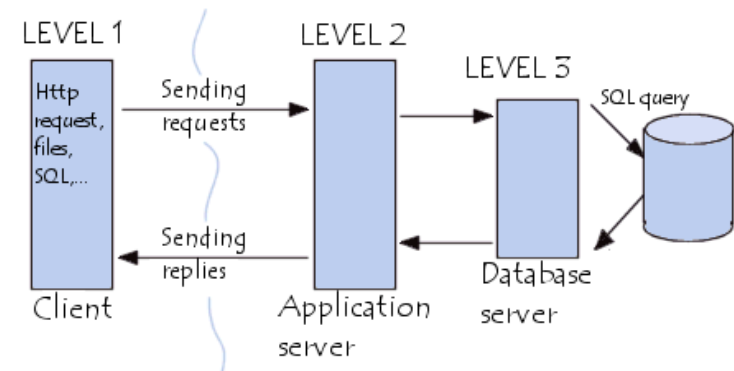
- Fast application development time
- Available tools are robust and provide fast prototyping.

✧ Disadvantages

- Not suitable for heterogeneous environments with rapidly changing business rules.
- Because the bulk of the application logic is on the client, there is the problem of client software version control and new version redistribution.

Three tier architecture

- ✧ 3-tier architectures attempt to overcome some of the limitations of the 2-tier architecture by separating presentation, processing, and data into 3 separate and distinct entities.
- ✧ Allows for better performance than a thin-client approach and is simpler to manage than a fat-client approach.
- ✧ The software in the client handles the presentation (to the user) using similar tools as in the 2-tier architecture.



Three tier architecture (cont...)

- ✧ When data or processing are required by the presentation client, a call is made to the middle-tier functionality server.
- ✧ This tier performs calculations, does reports, and makes any needed client calls to other servers (e.g.. a data base server).
- ✧ Middle tier servers may be multithreaded and can be accessed by multiple clients.
- ✧ The calling mechanism from client to server and from server to server is by means of Remote Procedure Calls (RPC's).

Three tier architecture advantages

- ✧ RPC calls from the presentation client to the middle tier provide greater generality (i.e. flexibility) than the SQL calls usually used in the 2-tier architecture.

- ✧ The 3-tier clients do not have to understand SQL.
 - This allows the organization, names, or even the overall structure of back end data to changes without requiring changes in the PC-based presentation clients.
 - This allows clients to access even non-relational databases and greatly simplifies introduction of new database technologies.



Three tier architecture advantages (cont...)

- ✧ Having separate functionality servers allows for the parallel development of individual tiers by application specialists.
- ✧ Can use reusable program modules for different applications, thus reducing development and migration costs.



Three tier architecture disadvantages

- ✧ Creates an increased need for network traffic management, server load balancing, and fault tolerance.
- ✧ Current tools are relatively immature and are more complex.
- ✧ Maintenance tools are currently inadequate for maintaining server libraries. This is a potential obstacle for simplifying maintenance and promoting code reuse throughout the organization.

Thin and fat clients

✧ Thin-client model (also called Slim-Client)

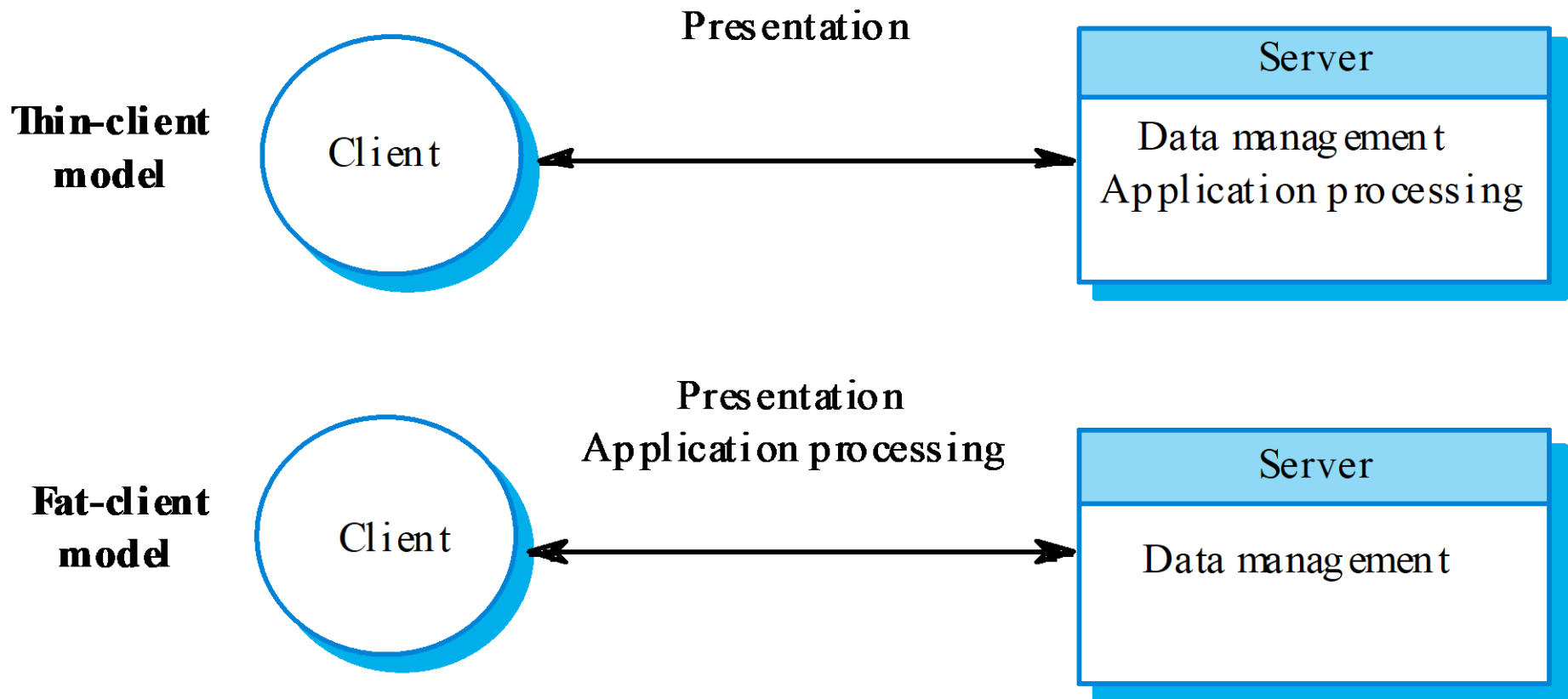
- In a thin-client model, all of the application processing and data management is carried out on the server. The client is simply responsible for running the presentation software.

✧ Fat-client model (Thick-Client)

- In this model, the server is only responsible for data management. The software on the client implements the application logic and the interactions with the system user.



Thin and fat clients



Use of Client-Server architectures

Architecture	Applications
Two-tier C/S architecture with thin clients	<p>Legacy system applications where separating application processing and data management is impractical.</p> <p>Computationally-intensive applications such as compilers with little or no data management.</p> <p>Data-intensive applications (browsing and querying) with little or no application processing.</p>
Two-tier C/S architecture with fat clients	<p>Applications where application processing is provided by off-the-shelf software (e.g. Microsoft Excel) on the client.</p> <p>Applications where computationally-intensive processing of data (e.g. data visualisation) is required.</p> <p>Applications with relatively stable end-user functionality used in an environment with well-established system management.</p>
Three-tier or multi-tier C/S architecture	<p>Large scale applications with hundreds or thousands of clients</p> <p>Applications where both the data and the application are volatile.</p> <p>Applications where data from multiple sources are integrated.</p>

Summary



✧ Two basic client-server architecture

- 2-tier
 - Thick clients: run some or all of the application logic.
 - Thin client: all of the application processing and data management is carried out on the server
- 3-tier
 - Allows for better performance than a thin-client approach and is simpler to manage than a fat-client approach.

