MODULE - 2

→ CLASSIFICATION OF CLIENT / SERVER SYSTEMS
→ Open systems standards
→ TWO - TIER COMPUTING
→ MIDDLEWARE
→ THREE - TIER COMPUTING
→ MODEL VIEW CONTROLLER (MVC)
→ PRINCIPLES BEHIND CLIENT / SERVER SYSTEMS
→ CLIENT / SERVER TOPOLOGIES
→ EXISTING CLIENT / SERVER ARCHITECTURE
→ ARCHITECTURE FOR BUSINESS INFORMATION SYSTEM.

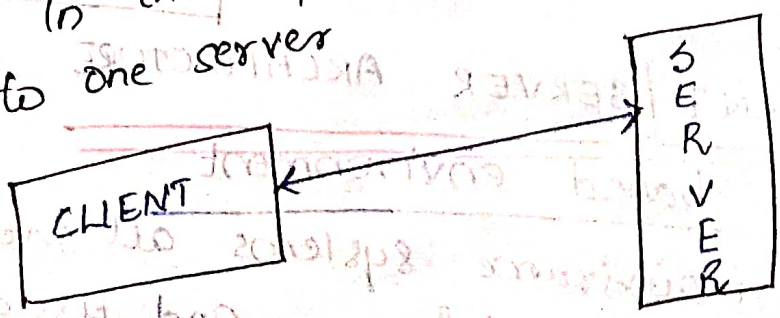
CLIENT - SERVER TOPOLOGIES.

A client/server topology refers to the physical layout of the client/server network in which all the clients and servers are connected to each other. The possible client/server topological design and strategies used are as follows:

(i) Single client, single server
(ii) Multiple clients, single server
(iii) Multiple clients, multiple servers.

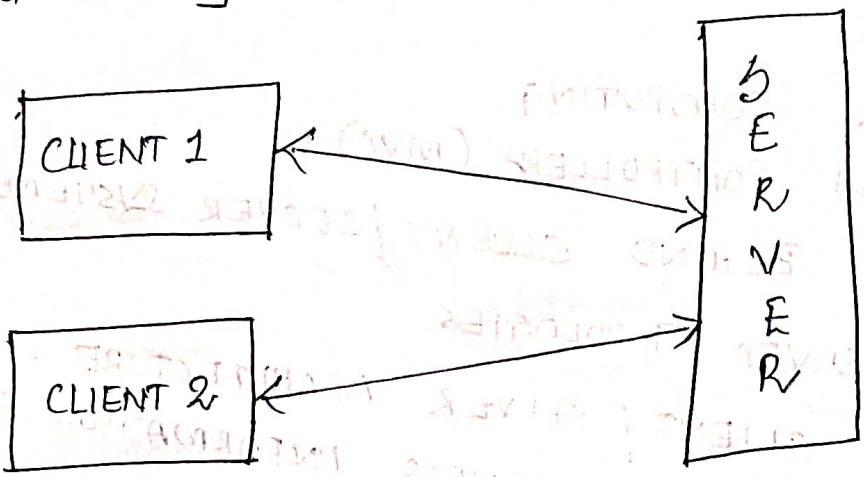
SINGLE CLIENT, SINGLE SERVER

↳ In this topology, one client is directly connected to one server



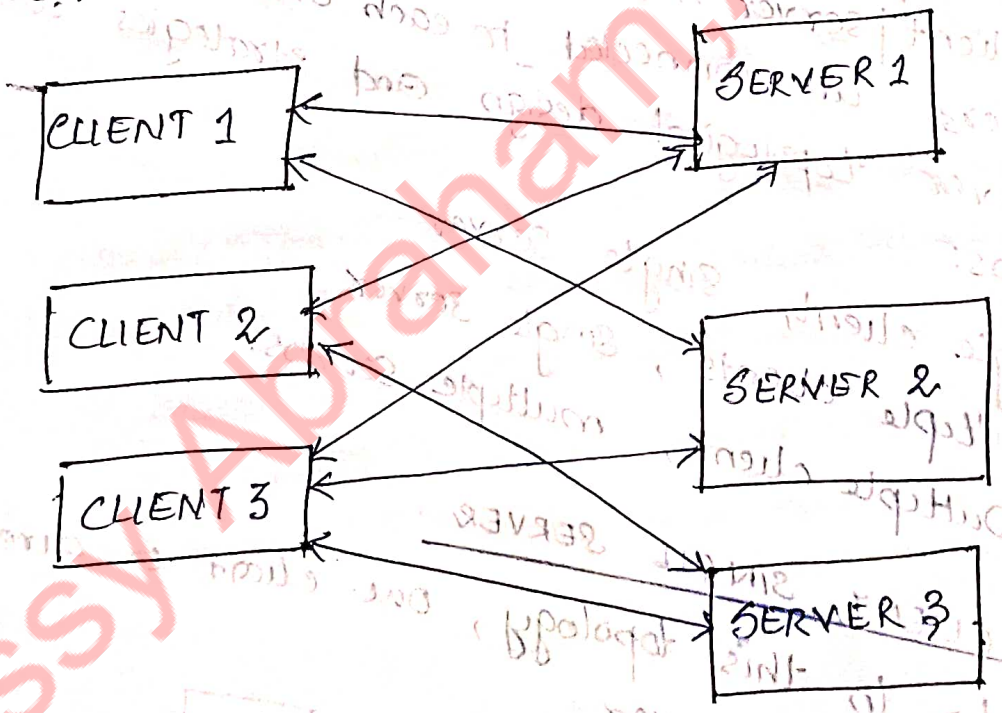
(ii) MULTIPLE CLIENTS, SINGLE SERVER

↳ In this topology, several clients are directly connected to only one server.



(iii) MULTIPLE CLIENTS, MULTIPLE SERVERS.

↳ In this topology, several clients are connected to several servers.

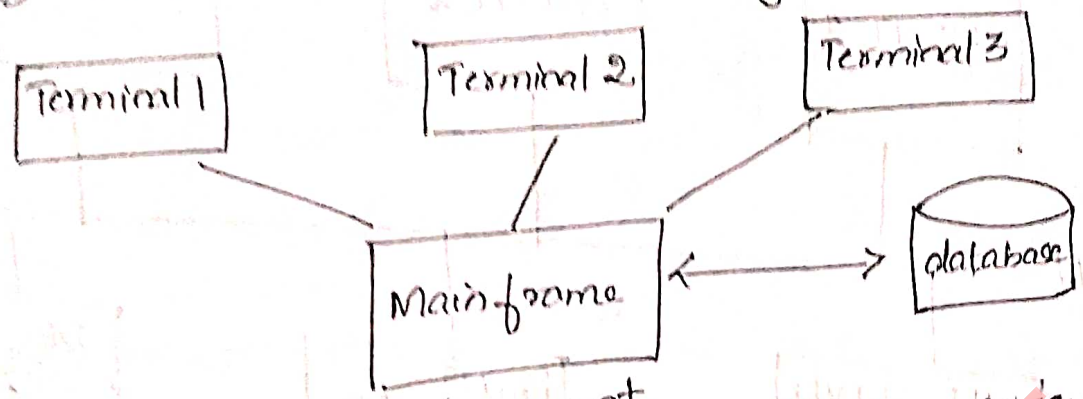


EXISTING CLIENT/SERVER ARCHITECTURE

1. Mainframe based environment

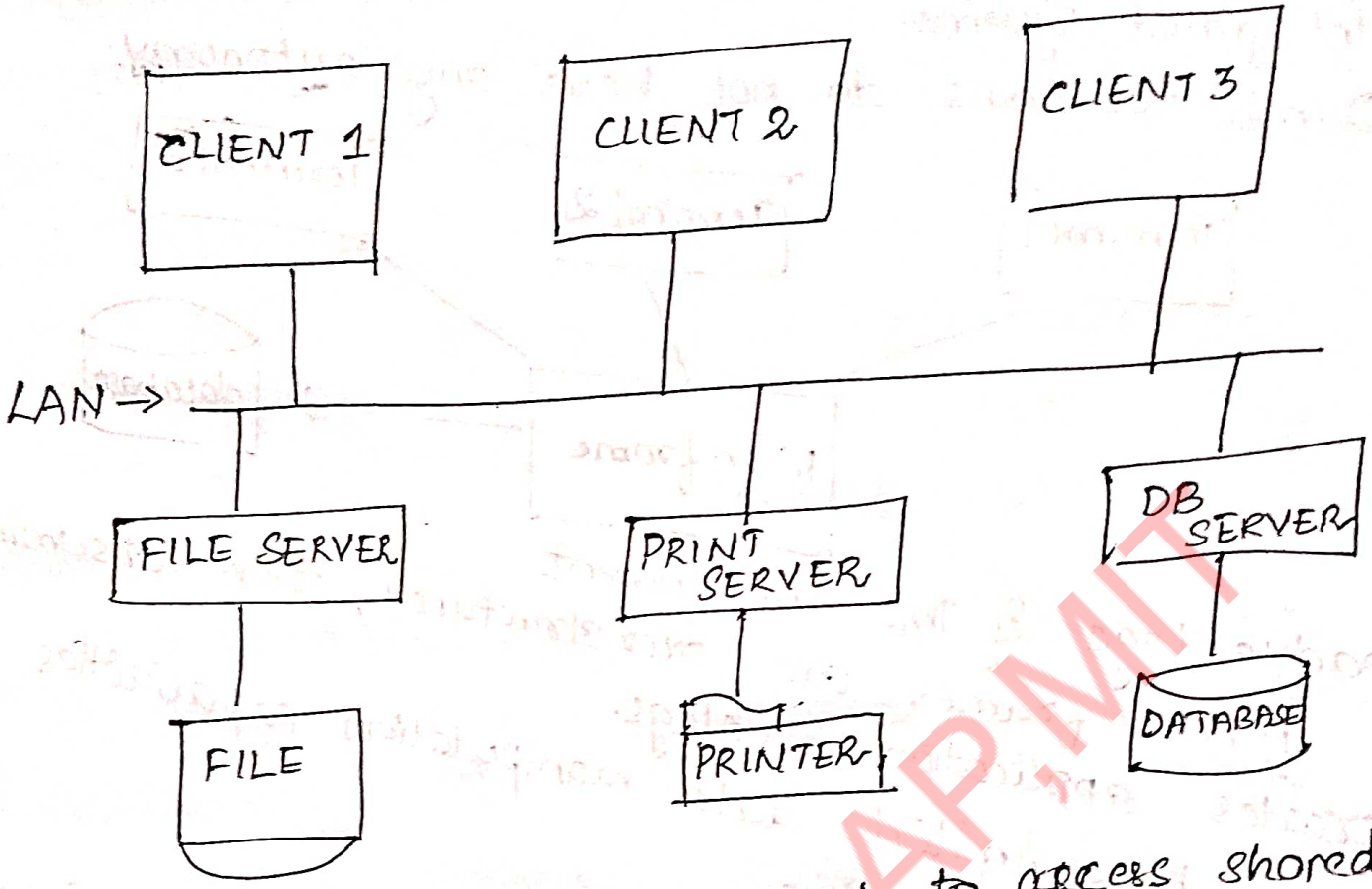
→ In mainframe systems all the processing takes place on mainframe and usually dumb terminals that are end user platform are used to display data on screens.

→ Main-frames are highly centralized and known as integrated systems.
→ Dumb terminals do not have any autonomy



Disadvantage of this environment
1) main frames are over structured, time consuming, creates application backlogs.
2) Has limited data manipulation capabilities.
3) Very inflexible.
4) Very expensive.
5) Unable to keep up with the demand for new applications.

2. LAN based environment
↳ In client/server LAN one or more stations called servers give services to other stations, called clients.
↳ The server version of network OS is installed on server, and client version of network OS is installed on clients.
↳ A network may have several servers, each dedicated for a particular task. These servers enable many clients to share access to the same resources and enables the use of high performance computer systems to manage the resources.

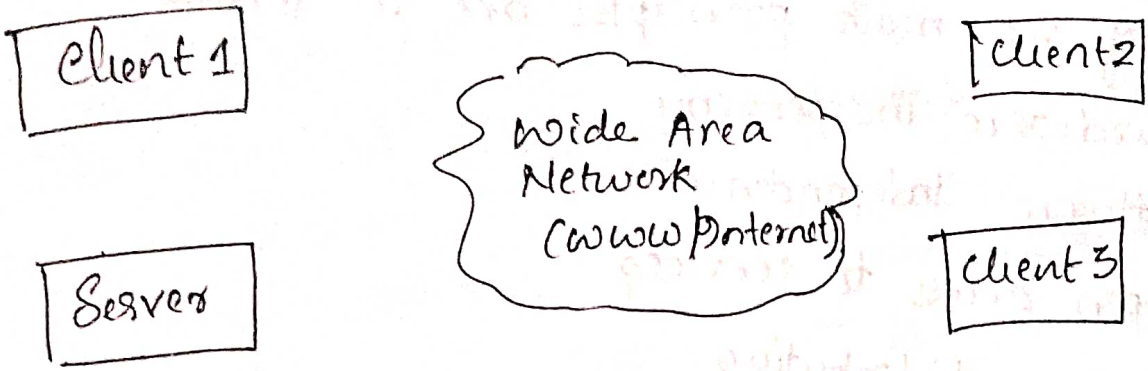


→ A file server allows the client to access shared data stored on the disk connected to the file server. Each client can send data to be printed to the print server, which then spools and print them. The file server station server runs a server file access program and a print server station a server print handling program.

→ Users, applications and resources are distributed and linked by single local area network.

3. Internet based environment.

→ Through internet vast amount of information becomes available from the same application and the interface.



→ Here web browsers are universal clients. A web browser is a minimalist client that interprets information it receives from a server, and displays it graphically to a user. The client takes complete responsibility for displaying the hypertext document, and for the user's response to it.

eg: Web browsers like Netscape and Spyglass interprets HTML commands. These browsers execute HTML commands and display text and images on a specific GUI platform; it can also navigate from one page to other using embedded ~~text~~ hypertext links.

→ The server system supplies multimedia document (pages) and runs some application programs (HTML forms and CGI programs) on behalf of client.

PRINCIPLES BEHIND CLIENT/SERVER SYSTEMS

The components of client/server architecture must conform to some basic principles, if they are to interact properly. These principles must be uniformly applicable to client, server and to communication middleware components.

Some of the main principles are as follows:

(i) Hardware independence

(ii) Software independence

(iii) Open access to services

(iv) Process distribution

(v) Standards.

Hardware independence:

→ Here the client, server and communication middleware should be able to run on multiple hardware platforms (IBM, DEC, Compaq, Apple and so on) without any functional differences.

Software independence:

→ Here the client, server and communication middleware processes should support multiple operating systems (such as Windows 98, Windows NT, Apple mac System, OS/2, Linux and Unix), multiple n/w protocols (such as IPX and TCP/IP) and multiple applications (Spreadsheet, database electronic mail and so on)

Open access to services.

→ All clients in the system must have open (unrestricted) access to all the services provided within the network and these services must not be dependent on the location of the client or the server.

→ Services should be provided on demand to the client.

Process distribution:

→ Processing of information is distributed among clients and servers.

→ The division of the application processing load must conform to the following rules

- Client and server processes must be autonomous entities. This properly enables to clearly define the functionality of each side and it enchances the modularity and flexibility of the system.
- Local utilization of resources (at both clients and server sides) must be maximized. The client and server process must fully utilize the processing power of the host computers. To utilize all resources, the server process must be shared among all the client processes, that is a server should service multiple requests from multiple clients.
- Scalability and flexibility requires that the client and server process be easily upgradable to run on more powerful hardware and software platforms. This provides additional capabilities and better performance for the client/server processes
- Interoperability and integration that client and server processes be integrated to form a system. Swapping a server process must be transparent to the client process.

Standards

→ All the principles that are formulated must be based on standards applied within the client/server architecture.

eg: Standard must govern the user interface, data acess, network protocols, interprocess communication and so on.

→ Standards ensure all the components interact in an orderly manner to acheive the desired results.

→ There is no universal standard for all the components.

eg: a) An application can be based on ODBC (Open Database Connectivity) instead of Integrated Database Application Programming Interface (IDAPI) for data acess.

[ODBC and IDAPI are database middleware components that enables the system to provide a data acess standard for multiple process].

b) The application might use IPX [Internet work Packet Exchange] instead of TCP/IP [Transmission Control Protocol or Internet Protocol], as the network protocol.

→ Standards should ensure that all the components (server, client and communication middleware) are able to interact as long as they use the same standards.

CLASSIFICATION OF CLIENT/SERVER SYSTEM

There are 2 types of client/server systems in existence.
(i) Two-tier (ii) Three-tier (iii) Multi-tier.

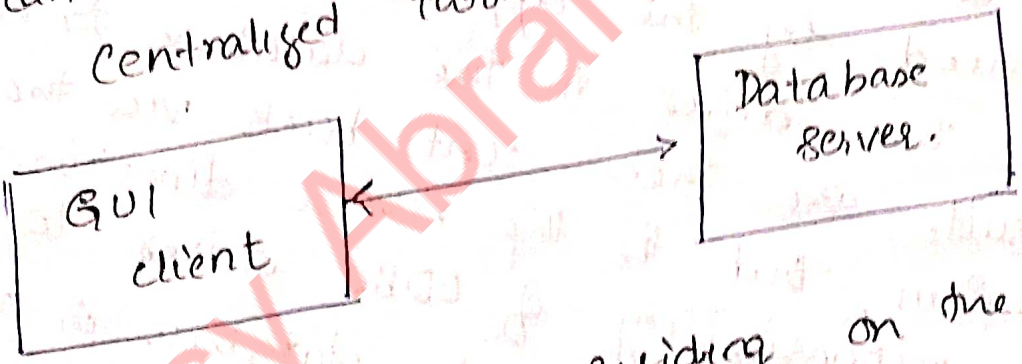
Two tier client/server model.

→ Application processing is done separately for database queries and updates and for business logic processing and user interface presentation.

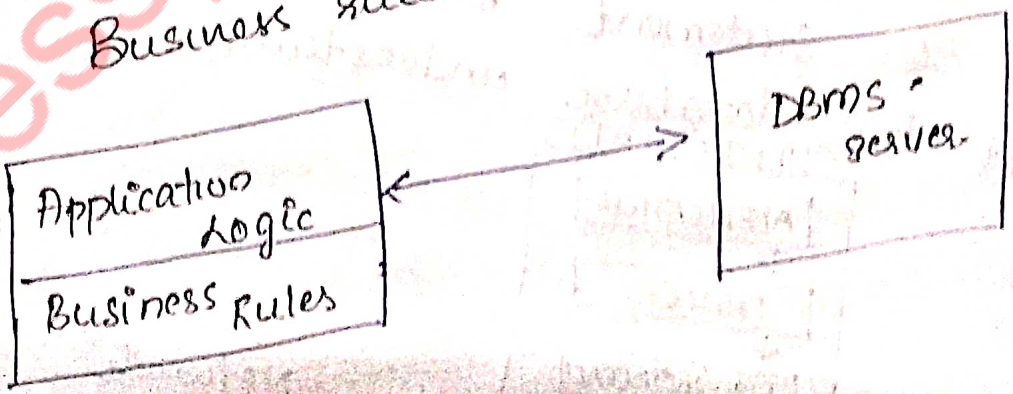
→ The application logic (the real business logic) is located on both the client program and in the database itself. Often business logic is merged into the presentation logic on the client side. On the database side, logic is often developed using stored procedures.

→ On a two tier architecture, if the client/server application has a number of business rules to be processed, then those rules can reside at either the client or at the server.

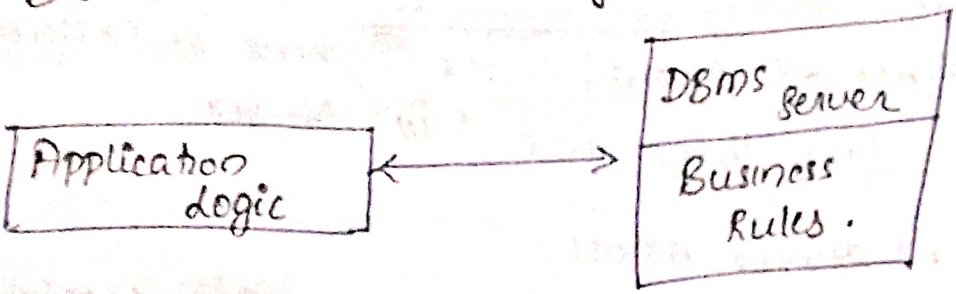
For centralised Two-tier representation



Business rules residing on the client



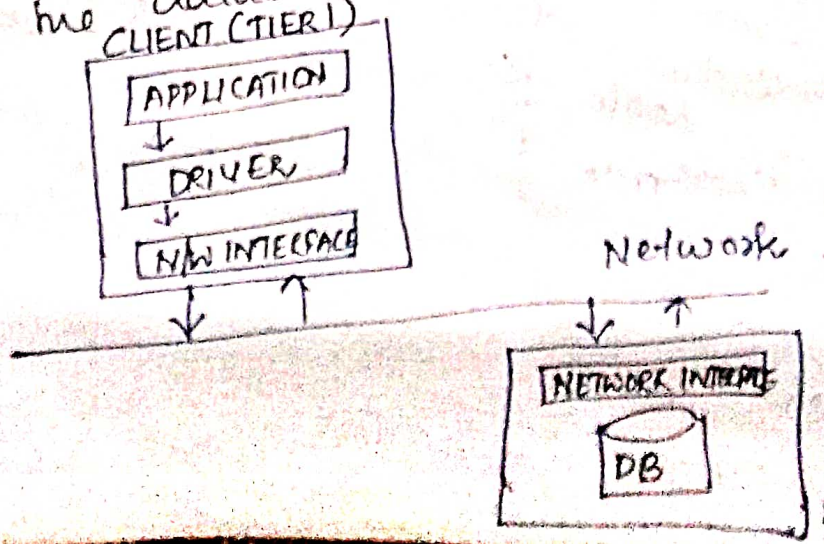
Business Rules residing on the server.



→ In 2 tier approach, the first tier is the client and the 2nd ~~tier~~ tier is the server. The client requests services directly from the server & client communicates directly with the server without the help of another server or server process.

→ To connect from client to server we use ODBC driver. ODBC is a software that comes along with dbms software and is installed in the client side.

→ In the 2 tier implementation, SQL statements are issued by the application and then handled on by the driver to the database for execution. The results are then sent back via the same mechanism, but in the reverse direction. It is the responsibility of ODBC driver to present the SQL statement to the database in a form that the database understands.



Advantages.

→ Availability of well integrated PC-based tools like PowerBuilder, MS Acces, 4GL tools provided by the RDBMS manufacturer, remote SQL, ODBC.

→ PC based tools show RAD [Rapid Application Development] ie Application can be developed in a comparatively short time.

→ Tools are inexpensive.

→ Least complicated to implement.

→ 2 tier client/server provides much more attractive graphical user interface (GUI) applications.

→ Architecture maintains persistent connection between client and database thereby eliminating overhead associated with the opening and closing of connections.

→ Faster than 3-tier

→ offers flexibility and simplicity in management.

Disadvantages.

→ Application development is done on client side, so maintainence cost of application and client side tools are expensive. As application development is done on client side, the 2-tier architecture client is called as fat client.

→ Increased network load → Since actual processing of data takes on the remote client, the data has to be transported over the network.

→ Applications are loaded on individual PC so the application logic cannot be reused.

→ Due to dynamic business scenario, business processes/ logic have to be changed. These changed processes have to be implemented in all individual PC's.

→ Software distribution procedure is complicated, expensive, ~~error~~ prone to errors and time consuming. As the application logic is executed on the PC's all these machine have to be updated in case of new release.

→ PC's are easy to crack so they are weak in terms of security.

→ Currently available drivers require native libraries to be loaded on client machine.

→ Some problem areas are encountered on implementing this architecture on the internet.

THREE TIER CLIENT-SERVER MODEL

→ To avoid embedding the application's logic at both the database side and the client side, a 3rd software tier is inserted in between.

→ On 3 tier architecture most of the business logic is located in the middle tier. On this structure, when the business activity or business rules change, only the middle tier must be modified.

→ On 3 tier architecture there are 3 main services

1) Presentation (GUI) or user services
2) Application services or business rules.
3) Database services or data services.

A. Presentation (GUI) or user services.

↳ Include maintaining the graphical user interface and generating what users see on the monitor.

↳ Presentation logic deals with screen formatting, windows management, input editing, what if analysis.

B. Application services or business rules:

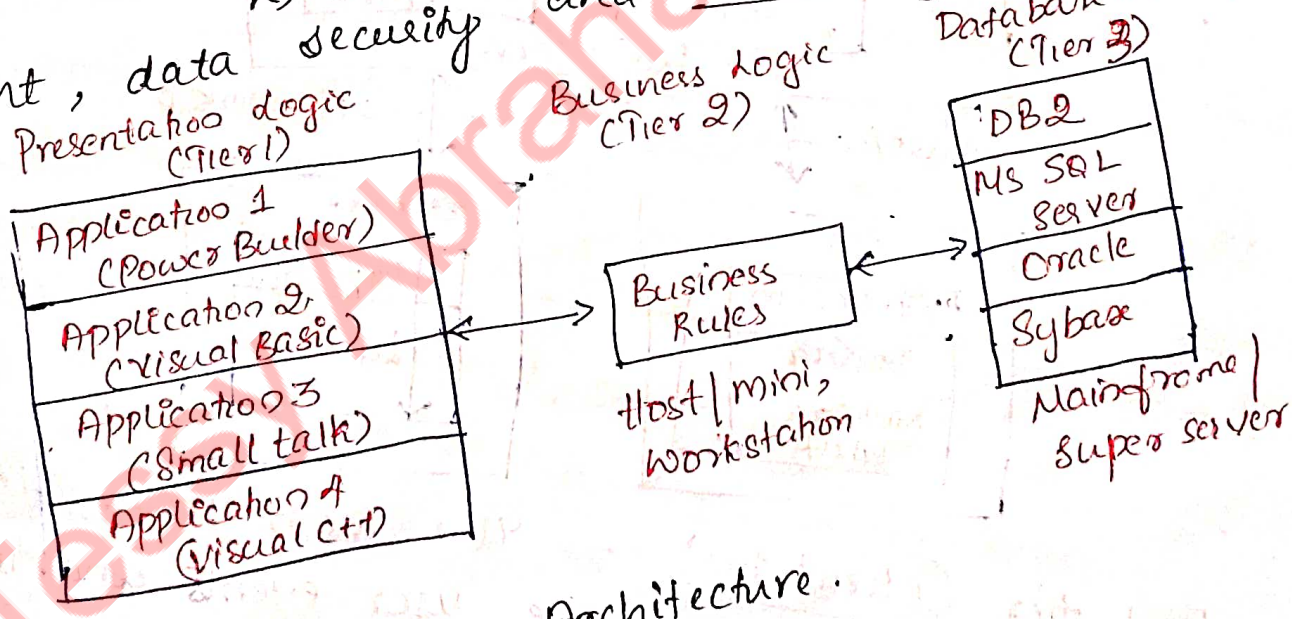
↳ Include executing applications and controlling program flow.

↳ Business logic deals with domain and range validation, data dependency validation, request/response architecture of InterProcess Communication level.

C. Database services or data server.

↳ It manages the underlying databases.

↳ It deals with: data access, data management, data security and SQL parsing. [calculates different ways how a query can run]

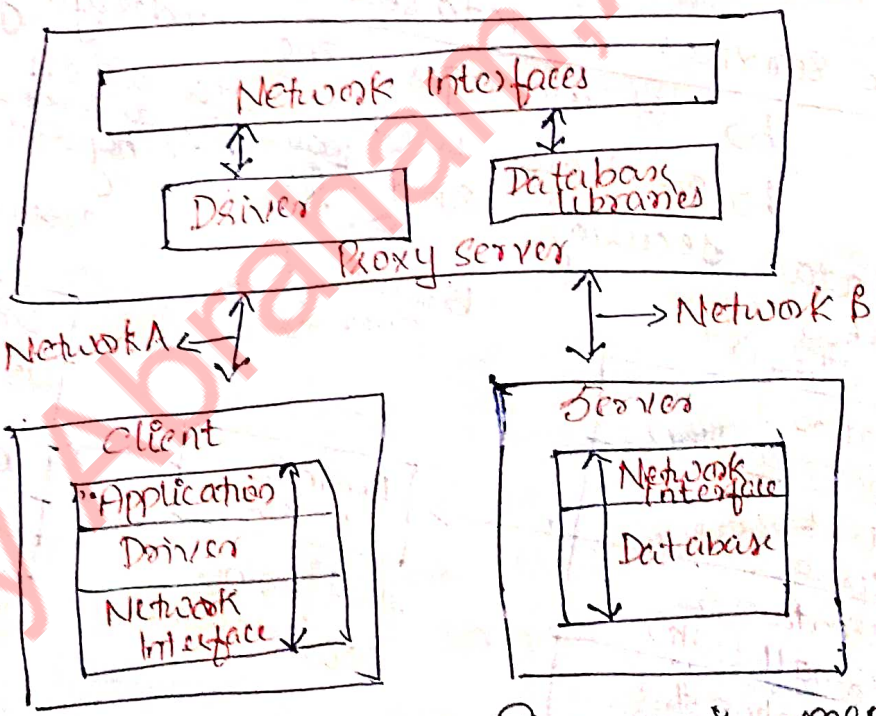


Three-tier Architecture.

→ In three tier model, a third server is employed to handle requests from the client and then pass them off to the database server. The third

Server acts as a proxy for all client requests. All client requests for the database are routed through the proxy server., thus creating more secure environment for your database.

→ Here the driver translates the request into a new protocol and then makes request via proxy server. The proxy server makes the database request on behalf of the client and passes the results back after they have been serviced by the database. This approach eliminates the need for DBMS to be located on the same server.



A. First tier (Client tier) → The main responsibility of this tier is to receive user events and to control the user interface and presentation of data. As most of the software is removed from the client, the client is called 'Thin Client' Mainly browser and presentation code

resides on this tier.

B. Second tier (application-server tier) → The complex application logic is loaded here. This tier can protect direct access of data.

C. Three tier (database-server tier) → This tier is responsible for data storage. This server mostly operates on a relational database.

Advantages of 3-tier model

1) Application maintenance is centralized with the transfer of the business logic for many end users into a single application server. This eliminates the concern of software distribution.

2) Clear seperation of user-interface control and data presentation from application logic; so more clients are able to have access to a wide variety of server applications.

3) Many users are able to access a wide variety of server application, as all application logic are loaded in the applications server.

4) As a rule servers are trusted systems. Their authorization is simpler. Data protection and security is simpler to obtain.

5) Redefinition of the storage strategy won't influence the clients. Switching from an RDBMS to an OODBMS won't influence the client.

6) Load balancing is easier with the seperation of the core business logic from the database server.

7) Dynamic load balancing → if bottlenecks in terms of performance occur, the server process can be moved to other servers at runtime.

8) Business objects and data storage should be kept as close together as possible; they should be together physically on the same server.

9) The need for less expensive hardware because the client is 'thin'.

10) Change management is easier and faster to execute. Business logic is implemented on the server rather than furnishing numerous PC's with new program versions.

11) The added modularity makes it easier to modify or replace one tier without affecting the other tier

12) Drivers can be managed centrally.

13) Database server does not have to be directly visible to the Internet.

eg of 3 tier is the web environment. web browser acts as a thin client; and a web server acting as a application server.

Disadvantages

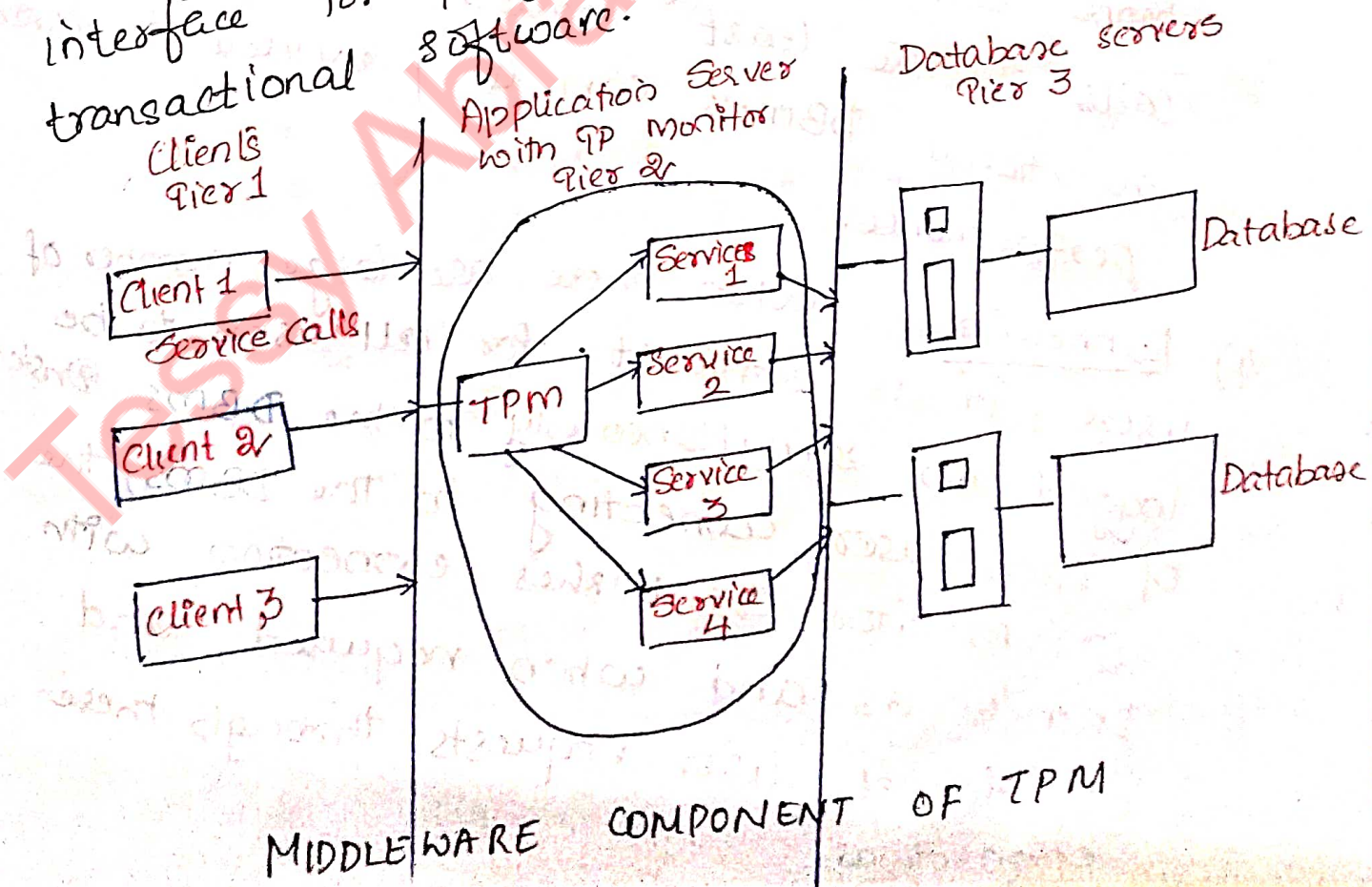
1) The client does not maintain a persistent database connection.

2) A seperate proxy server may be required.
3) The network protocol used by the driver may be proprietary.
4) There will be increased network traffic if a seperate proxy server is used.

Transaction Processing Monitors (TPM)

→ A transaction processing monitor is a program that controls data transfer between client and server inorder to provide consistent environment, particularly for online transaction processing (OLTP)

→ A transaction processing monitor or TP Monitor is a middleware component that provide access to the services of a number of resource managers (such as DBMS, operating systems, user interface. and messaging software) and provides a uniform interface for programmers who are developing transactional software.



Advantages of TP Monitors.

1) Transaction Routing → TP monitor can increase scalability by directing transaction to specific DBMS's

2) Managing distributed transaction → The TP Monitor can manage transactions that require access to data held in multiple, possibly heterogeneous, DBMS's. for eg, a transaction may require to update data item held in ORACLE DBMS at site 1, an Informix DBMS at site 2 and IMS DBMS at site 3. TP Monitors normally control transaction using X/open Distributed transaction processing (DTP) standards.

3) Load balancing → TP Monitor can balance client requests across multiple DBMS's on one or more computers by directing client services calls to the least loaded server. Also an additional DBMS's can be provided to increase performance.

4) Funneling :- When there are large number of users, it is difficult for all users to be logged on simultaneously to the DBMS. Instead of each user connecting to the DBMS, the TP monitor establishes connection with DBMS's as and when required, and can funnel user requests through these connections.

This allows a large number of users to access the available DBMS's with a potentially smaller number of connections, that is resource usage is less.

5) Increased reliability:- TP Monitor acts as a transaction manager, performing necessary action to maintain the consistency of database, with the DBMS acting as a resource manager. If the DBMS fails, the TP Monitor may be able to resubmit the transaction to another DBMS or can hold the transaction until DBMS becomes available again.

Egs of TP Monitors include CICS and Encina from IBM, Tuxido from BEA system.

Three tier with Message Server.

→ Messaging implements 3 tier architecture.
→ Messages are prioritized and processed asynchronously. Messages consist of header that contain priority information, the address and the identification number. The message server connects to the relational DBMS and other data sources.

Three tier Architecture and Internet

A typical web application uses the following 3 tier architecture.

* The user interface runs on the desktop as client
* The client is connected through

one or more immediate servers links) to a web server
· The web server is supported by database server
The web applications rely on Internet standards (HTTP, HTML, XML etc) as well as on distributed object programming languages.

Three tier with an Application Server

→ The main body of an application runs on a shared host rather than in the client environment.

→ The application server includes business logic, computations and data retrieval engines.

Advantages

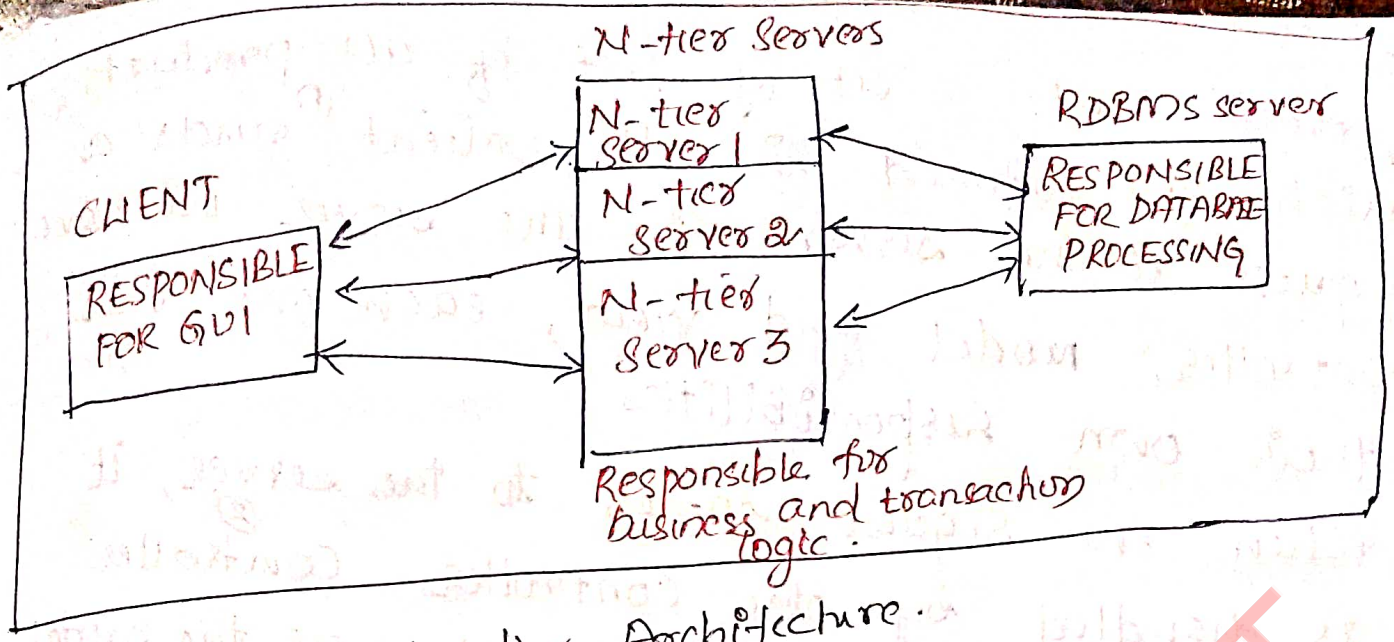
→ Scalable, support and installation costs are less., and also secure.

N-TIER CLIENT/SERVER MODEL

→ The most common approach used when designing N-tier system is the 3 tier architecture.

→ N-tier architecture provides finer-grained layers.

→ The splitting may be between application logic components, security logic, presentation logic, computation and also I/O intensive components.



N-tier Architecture.

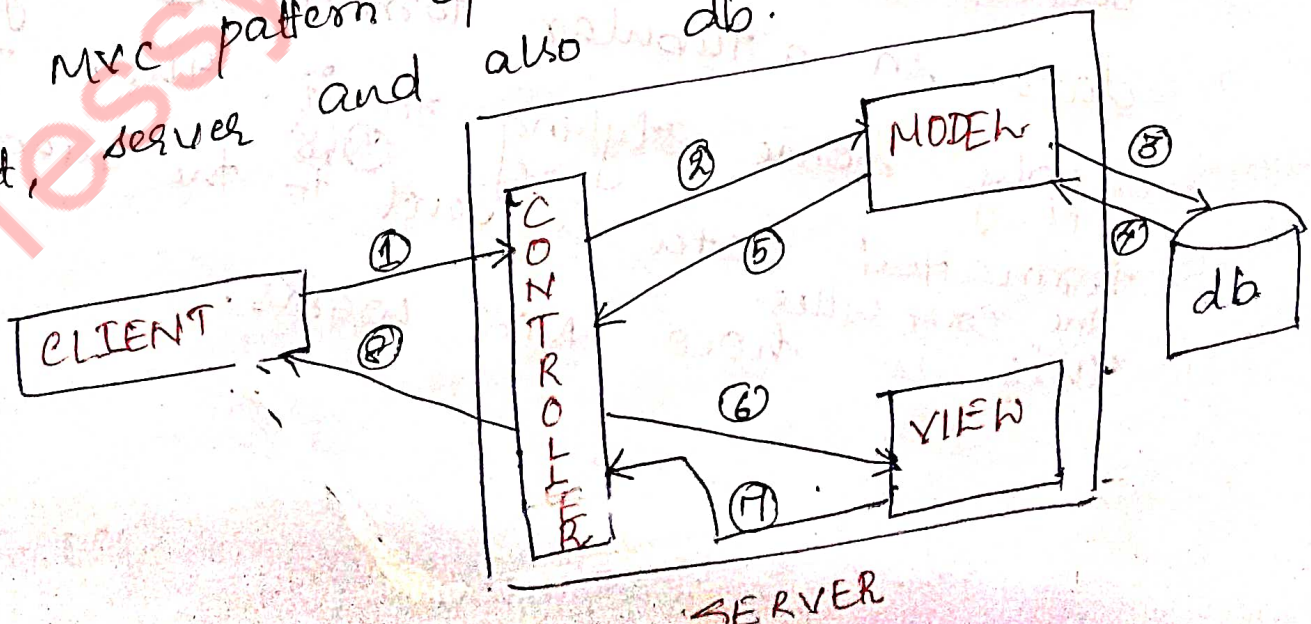
Advantages

→ Overall performance increases.
→ Business logic is centralized.
→ Enchanced security level

MODEL VIEW CONTROLLER (MVC)

→ MVC stands for Model View Controller.
→ It is an architectural pattern means it governs our whole architecture of the application.
→ MVC divides a software application into 3
a) MODEL b) VIEW c) CONTROLLER.

In a MVC pattern of a web application, there is client, server and also db.



If client wants a set of list of all products which are stored in db, client ① sends a request to the server. On the server we have controller, model and view, each one has their own responsibilities.

→ When the request comes to the server, it is handled by the controller. ② controller sends the request to model, as the model is ~~speati~~ specialized in handling the db. ③ Model interacts with the db and gets all the list of the products.

→ Model gives the data ④ from the database to the controller. But the data which came is in a raw format and is not in a human readable format. ⑤ So this data is sent to the view.

→ View formats the data ⑥ so it looks attractive to the user. It arranges the data in a tabular format or may apply some styling to it. ⑦⑧ Then the formatted data is send to the client through the controller.

This is how MVC works.

→ Each of the 3 components has specific responsibilities:

* Model → it interacts with db and also executes business logic. [Model basically deals with data]

* View → It generates User Interface (UI) for the user

* Controller → It takes the user inputs (also called request parameters). It also interacts with Model and View. [Controller is the middleman between model and view].

OPEN SYSTEMS AND STANDARDS.

→ client/server computing offers organizations the ability to distribute computing, storage and other resources as dictated by business needs. But the standard-based open systems provide the necessary flexibility and interoperability for maximum benefits.

→ IEEE Technical committee on Open systems offers the following definition of open systems.

A comprehensive and consistent set of international information technology standards and functional profiles that specify interfaces, services and supporting formats to accomplish interoperability and portability of applications, data and people.

Standard areas

Standards address 4 areas of client/server computing.

① Platforms :- These standards are developed by hardware and software vendors, usually in response to defacto standards, such as Intel chips, UNIX and DOS with windows.

② Networks :- Industry standard networking protocols such as OSI and TCP/IP is being used.

③ Middleware :- This is the software that sits between application and the OS including GUI's, databases, email systems etc.

④ Applications :- Organizations decide on standard applications to facilitate work group interaction and work-product compatibility.

Interoperability and portability are provided through adherence to standards. Portability means that software will run on other platforms without requiring modifications to application code. Interoperability means that the software can work with software (both itself and other vendors' software) on other platforms.

EXISTING STANDARDS

Currently there are recognized standards for the server operating system environment and for network protocols.

UNIX:

→ Developed by UNIX systems Laboratories (USL), a division of AT&T.

→ This is an OS for scientific, engineering and technical applications.

→ Written in C

→ It takes advantage of ~~RISC~~ power of RISC technology and offers multitasking and multi-user support. It offers excellent price/performance characteristics.

→ Vendors such as IBM, Digital and Microsoft offer their own versions of UNIX - AIX, Ultrix, XENIX respectively.

→ Latest version is UNIX System V Release 4 (SVR4). It uses less memory and needs less hard disk requirements so it can be used in client machines.

POSIX

→ Portable Operating System Interface (POSIX) from IEEE is a UNIX based specification, which is a standard for server OS.

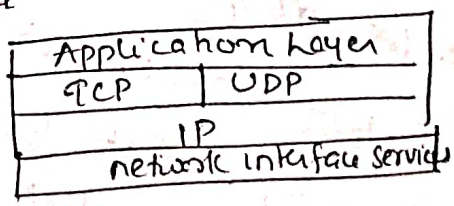
→ It is a uniform means for a C language application it can request service from an OS regardless of the underlying hardware architecture or operating environment.

→ Programmers can choose from a list of standard library functions and system header files.

TCP/IP

→ Transmission control protocol / Internet protocol (TCP/IP) is a defacto standard for interconnecting incompatible computers. It is used in a LAN based micros to connect to a UNIX host

→ TCP/IP is a set of network and transport protocol. TCP runs on top of IP, controls the packet delivery and IP takes care of interplatform and internet work communications.

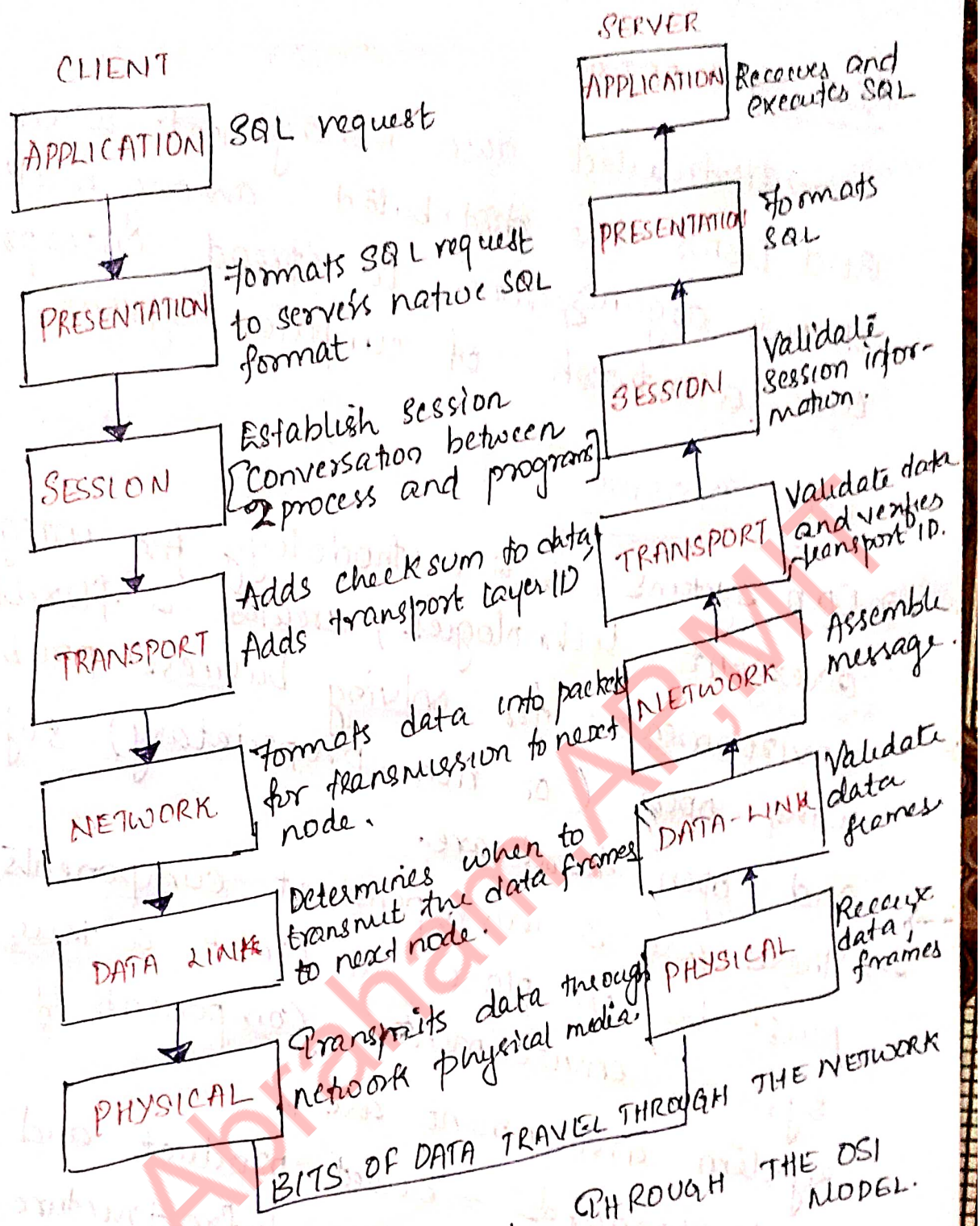


UDP → user Datagram

OSI

→ Open systems interconnection (OSI) model was developed by ISO [International Standards Organisation] to provide a common basis for communication system standards.

→ It can support heterogeneous communication.



FLOW OF INFORMATION THROUGH THE OSI MODEL.

→ OSI model provides a hierarchical layer structure in which each layer performs a specific n/w function, services the next high layer and accepts requests from the next lower layer.

RDA and DRDA.

→ distributed data management, when the data and DBMS are distributed among multiple systems allows organizations to spread enterprise data over a network of computer systems.

OPEN SYSTEMS

→ Open systems, is a methodology for integrating divergent technologies, creates a flexible environment for solving business problems using open (or non-proprietary) software and open hardware.

→ There are 3 interdependent components that must be in place to create a true open systems environment. Components of open system environment are :

(i) Standards - based product and technology
(ii) Open development Infrastructure.
(iii) Management directives.

✓ Standards based products and technology.

→ Open system technology provides portability and interoperability. Closed technologies will dead-end progress.

✓ Open development Infrastructure.
- Standards must ensure that current implementation of technology build on prior implementations and are able to support future implementations.

✓ Management directives
- These directives, ensure that technology does not benefit one group at the expense of the organization.

ARCHITECTURE FOR BUSINESS INFORMATION SYSTEMS

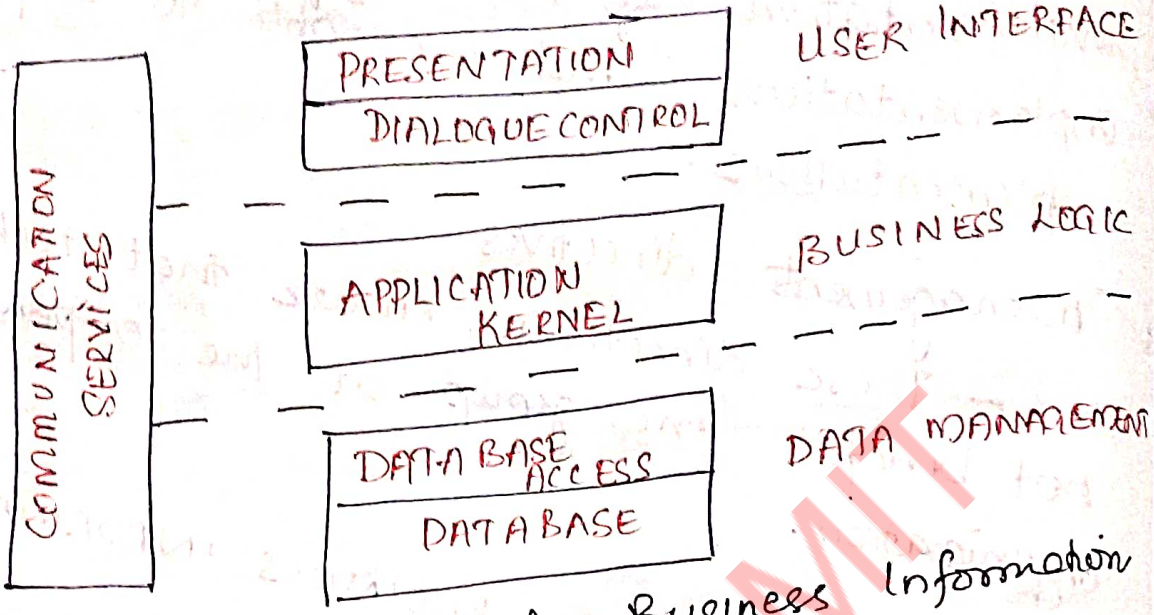
Business information system is an integrated set of components for collecting, storing and processing data and for providing information for business. Business firms rely on information for operations, interact with customers and manage the compete in the market place.

Three Layer Architecture

Typically, a business information system follows a three layer architecture.

The user interface handles presentational tasks and controls the dialogue. The application kernel performs the business

logics and the database access layer connects the application kernel functions to a database.



3-layer Architecture for Business Information System.

General Forces

- Business Needs Vs construction complexity

Allocating functionality and data to the places where it is actually needed supports distributed business process very well, but distributions raises a systems complexity. Few sources of complexity are GUI, ~~software~~ Middleware and heterogeneous OS environment.

- Processing styles

Different processing styles require different distribution decisions. Batch processing need

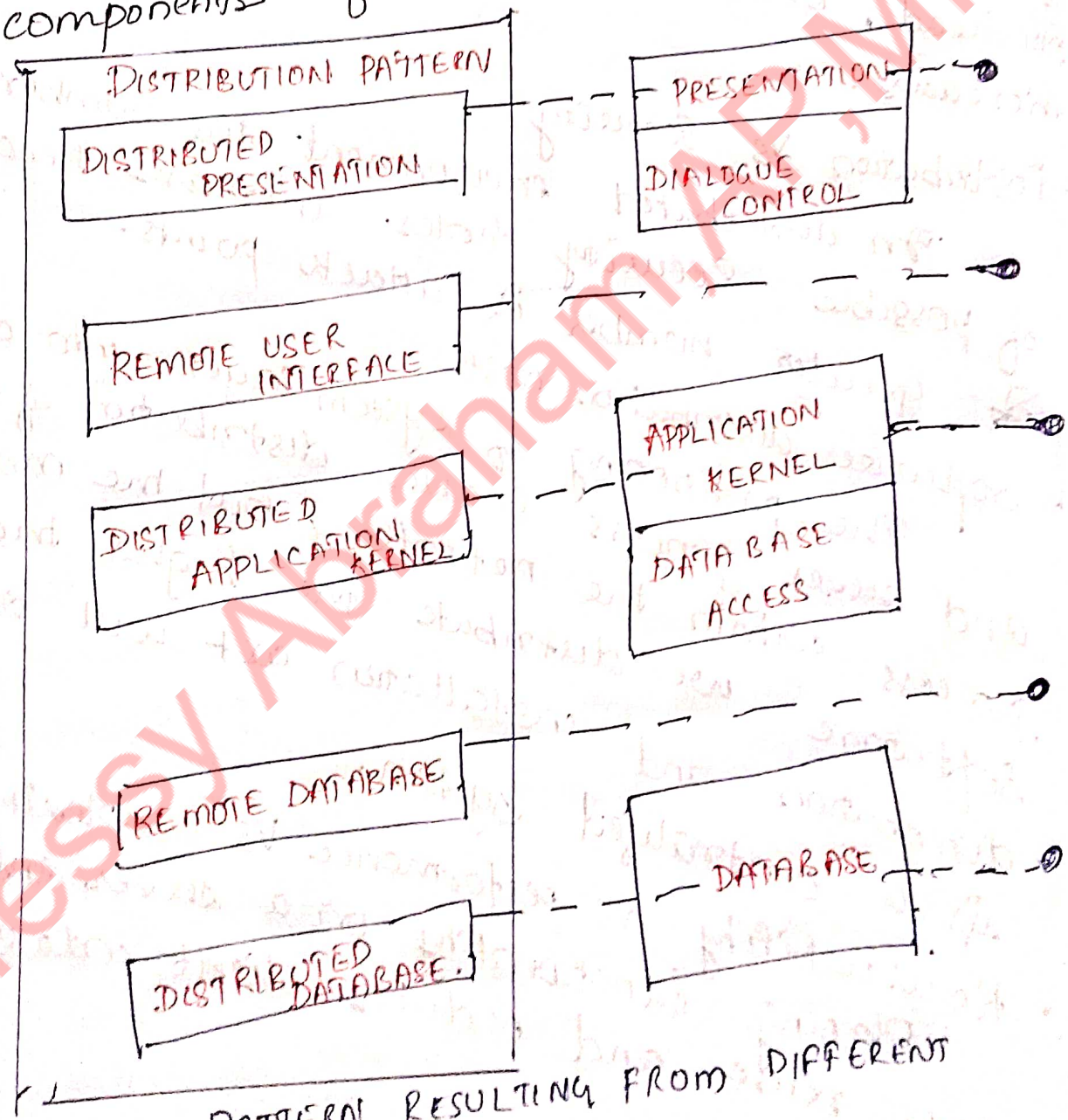
processing power close to the data and interactive processing should be close to input/output devices.

- Distribution Vs performance
  We gain performance by distributed processing units executing tasks in parallel, placing data close to the processing and balancing workload between several servers. But as the level of distribution increases the communication overhead, bottlenecks in communication network increases.
- Distribution Vs Security
  On distributed environment the number of possible security holes increases because of greater number of attack points.
- Software distribution cost
  The partitioning of system layers into client and server processes enables distribution of the process within the network, but the more software we distribute the higher the distribution and installation cost will occur in a centralized system.
- Reusability vs performance Vs complexity
  Placing functionality on a server enforces code reuse and reduces client code size,

but data must be shipped to the server and the server must enable the handling of request by multiple clients.

Distribution Pattern

There are several distribution styles to distribute an information system by assigning client and server roles to the components of the layered architecture.



PATTERN RESULTING FROM DIFFERENT CLIENT/SERVER CUTS.

Distributed presentation:

This pattern partitions the system within the presentation component. One part of the presentation component is packaged as a distribution unit and processed seperately from other part of the presentation, which can be packaged together with the other application layers.

Remote User Interface:

Instead of distributing presentation functionality the whole user interface becomes a unit of distribution and act as a client of the application kernel on the server side.

Distributed Application Kernel:

The pattern splits the application kernel into 2 parts which are processed seperately.

Remote database:

The database is a major component of business information system. Sometimes, several applications work on the same database. This pattern can locate the database component on a seperate node within the system's network.

Distributed database:

The database is decomposed into seperate database components, which can interact using IPC. With a distributed data base an application can integrate data from different database systems, or data can be stored more closely to the location where it is processed.

X ——————— X.