1.1 COM

1).COM is the short form of Component Object Model.

2).COM is Microsoft platform for S/W componentary introduced by Microsoft in 1993.

3). Microsoft COM technology in the Microsoft Windows Family of the Operating system enables S/W component to communicate.

4).COM is used by developer to create re-usable s/w component , link component together to built applications and take advantage of window services.

5).COM is used in applications such as:- Microsoft Family of product.

6). It is the first time introduced with win'3.0 . This technology was the base of OLE(Object Linking & Embedding) which inturn was one of the Leading technology with win'3.0 .

7). Example:-

COM OLE technology allows word documents to dynamically link to data in Ms-

Excel , Spreadsheet and COM automation allow user to built script in their appl

ication to perform repeatitive task to control one application to another.

1.2 DCOM

1). Stands for Distributed Component Object Model.

2). It is the family of COM technology which includes COM+.net etc.

3). Microsoft's protocol that enable s/w component directory over the N/W in a reliable . It is also called N/W OLE extend.

4).DCOM is an addition to COM facilitate the transparent distribution of object over N/W and over the internet.

5).DCOM that allow communication and manipulation of object over a N/W.

1.3 History of COM/DCOM

1). One of the 1st method of inter-process communication in window was DDE (Dynamically Data Exchange) which allow sending and receiving message is so called conversion between Application.

2).DDE

This technology was the base of OLE which dynamically data exchange .(This technology was the base of OLE which dynamically link one s/w to another) i.e. the leading technology introduced with win'3.0 .

3). Text conversion or window messages could not to be Flexible as to allow sharing application feature in a Robust (strong) and Extensible.

4).OLE- 1991, OLE 1.0 was basically a method of handling compound component.

A compound document is a storing data in a Multiple formats such as:- Text , Graphics , Video and Sound Line.

5). By the time, version 3.1 of window was released . COM was created a new foundation and OLE change to OLE 2.

6). In 1993, Microsoft release the OLE2 which encode more than just compound document.

It supported an entire architecture of object-base services. If COM was the part of object-base services.

The foundation of OLE2 is named as COM.

7).COM – consist of set of standard that define interface for s/w . These standards helps the s/w manufacturer to add unique s/w function into re-usable s/w component.

8). At the same time, DCOM as a separate entity provided by Microsoft Propitary for the communication of s/w component across n/w computer and was called N/W OLE extend.

1.4 Benefits / Importance of COM/DCOM

1). For vendor COM, gives a single module with other application and distribution computing environment.

2). It allows the developer to built and distribute application more easily.

3). It gives greater range of s/w choices with better productivity of users.

4). It allow two or more program application or component or component to co-operate with one another even they are retain in different times by different vendor.

1.5 Component of COM

COM allows creation of independent and re-usable component. COM component attract with each other on the basis of client-server model. Based on this COM component can be categorized into 2(two) parts:

1).Client Component.

2).Server Component.

1).Client Component.

Client Component uses the services and functionality provided by other component.

2).Server Component.

Server Component is a COM component that exposes it's functionality and services so they can use it.

Example:-

Consider a situation where a user insert into a word document. Here, bitmap image exposing it's functionality to the word document is acting as a server document was the word document. Using the functionality of Bitmap image is acting as a client.

Think about other situation where a user want to insert a power-point slide into an Excel Worksheet. Here, the power-point slide is acting as a server as a Excel Worksheet is accessing the functionality of the power-point application.

1.6 There are two(2) types of COM component

a). Inprocess Server.

b). Outprocess Server.

a). Inprocess Server.

1). It means it’s code execute in the same process which space as a client.

2). It is implemented as DLL.

3).DLL stands for “Dynamic Link Libraries”. DLL stores a specific set of Function and Procedure separately for the executable.

4).DLL’s occupies the memory only when the program call for the function or procedure define in them.

5).DLL get executed in same memory area of the client.

6). As DLL’s occupy the same memory space that of the client they can communicate with the client at a faster speed.

7). Therefore, the best-suited for application that involve large amount of data transfer between the server and client, the n/w traffic will reduced.

b). Outprocess Server.

1). Outprocess Server means it’s code execute in another process at the same machine or remote machine.

2). Outprocess Server implemented as a standard alone executable having an .exe extension.

3). Outprocess Server is executed in different address space from that of the client.

4). They are executed in a separate space than of the client if they fail it does not affect the entire client application which is true in case of DLL. Therefore, exe are more stable than DLL.

5). They are slower in execution in DLL as data have to be transferred between client address and component address space.

1.7 COM/DCOM Issues.

1).Persistence of Code.

2).Sharing.

3).Scalability.

4).Language Independence.

5).Ease of Modification.

6).Versioning.

1).Persistence of Code.

A COM component once created can be used any no. of times, thus making develop a simple task. It is used number of time i.e. Re-usability.

2).Sharing.

This is one of the most wonderful features provided by COM/DCOM. It means that we can design our component to be either In-process, Out-process or Remote-Accessed but the calling application will interact with them in the same manner. The client application is never aware the component using in an In-process or Out-process or it has been accessed remotely because of their issue, a developer can develop and distribute component in a manner that makes the most sense to the client application using them.

3).Scalability.

Scalability refers to the ability of application cop with their increasing load in term of number of user using that application and also data moving across the N/W. A critical factor for a distributed application is its ability to grow with number of users, the amount of data and the required functionality. The application should be small and fast when the demands are minimum. But, it should be able to handle additional demands without Scalability, Performance or Reliability.

4).Language Independence.

COM provides Programming Language independence because thus, component created in one language and can be used in other language.

5).Ease of Modification

Components of COM created in COM are independent of each other because of these fact component can be upgraded without affecting other part of the program or other component.

6).Versioning.

COM component are self-versioning this means that whatever some few functionality is added to an old component. The version of this component changes automatically when this component is distributed to new client they can use new functionality. As well as new functionality provided by the new component while the old client can continue using old functionality.

2.1 Distributed

1). To be a true system model, an Architecture must allows a distributed involving system to support millions of object without risk. COM is such an application built.

2).COM support distributed object. i.e. allow application in a number of different component. Objects each of which can run on a different computer.

2.2 Scalable

1). It means that if new H/W is added to the system, the performance of the system should be improved automatically. This means that if an application shipped (change) from a single process or environment to a Multiprocess environment. The application should be able to take advantage of Multi-processing environment.

2).A scalable application can readly adopt to an increase work load without encouring H/W expenses or poor performance.

3). Scalability is a critical feature for those application that must support on expending enterprise.

2.3 COM Security

For a Distributed object system to be useful in real word it must provide a means for secure access to object and the data they encapsulated.(Single application).

COM provide security among several crucial dimension.

1).COM uses standard O.S permission to determine whether a client has write to start the code associated with a particular object.

2).COM uses O.S application permission to determine if a particular client can load the object at all or not. If so, whether they have read, write access it.

3).COM provide cross process and cross N/W object server with a standard security information about the client or the client that are using it. So, that a server can in more.

4).COM get security against unauthorized user to access the data.

2.4 Transactions and Databases

Transactions are atomic operation in which one part of the operation can succeed unless all part of the operation succeed.

A successful transaction is committed and any changes, it has made data store.

Unsuccessful transaction are rolled back so that all data are stored returned to their state prior to the transaction inception. This transaction approach to data modification helps develops guaranteed that data store in consistence state.

2.5 How Transaction are Performed

Here, we get the transaction of i.e. Modification in database if the data is modified it is the successful transaction and a successful transaction is committed and modified the data and stored in the database.

In the case of unsuccessful transaction. **There are two(2) cases cannot modified it is unchanged.**

1). Due to error in Transaction, Abortion of data are essential.

2). Due to system error data roll-back and cannot modify.

2.6 Integral Facets

Transaction have several Integral Facets known as **A C I D.**

A :-> Atomicity.

C :-> Consistency.

I :-> Isolation.

D :-> Durability.

Atomicity.

Atomicity refers to the fact that all element of a transaction are fitted as invisible. A transaction in all or nothing proposition. In a transaction where three database tables are updated all the tables will be updated all the tables will be updated or not.

Consistency

Property promises that the data store will remain stable. Manipulation of data will not corrupted by concurrent operation. Thus, data will not collide other of the system.

Isolation

Transaction of operation isolated or independent other user of the data stores involved. Therefore, no other element of the system will ever view a partial transaction. This means that no object will ever retrieve record from a table reflecting other part of the object.

Durable

Transaction are Durable integrating that once committed the operation will compute one way or another.

The durability property state that transaction are stable, after being committed the operation will complete one way or other.

2.7 Multi-Tier Architecture

Application for Business operation can usually have following processes.

1). Take the data entered by the user as input.

2). Process the input data as per the business logic.

3). Storing the data or outputting it to the user.

Therefore, all application are having three element within them.

1).User-Interface

Through which user interact with the application. It is used for accepting input from the user and present the output to the user.

2).Business Rule/Application Logic

It is the 2nd element of any application.

3).Data Storage Element

It is used to store and retrieve the data.

These three(3) element decide the Architecture/ Model of an application which are:-

1).Single-Tier Architecture.

2).Two- Tier Architecture.

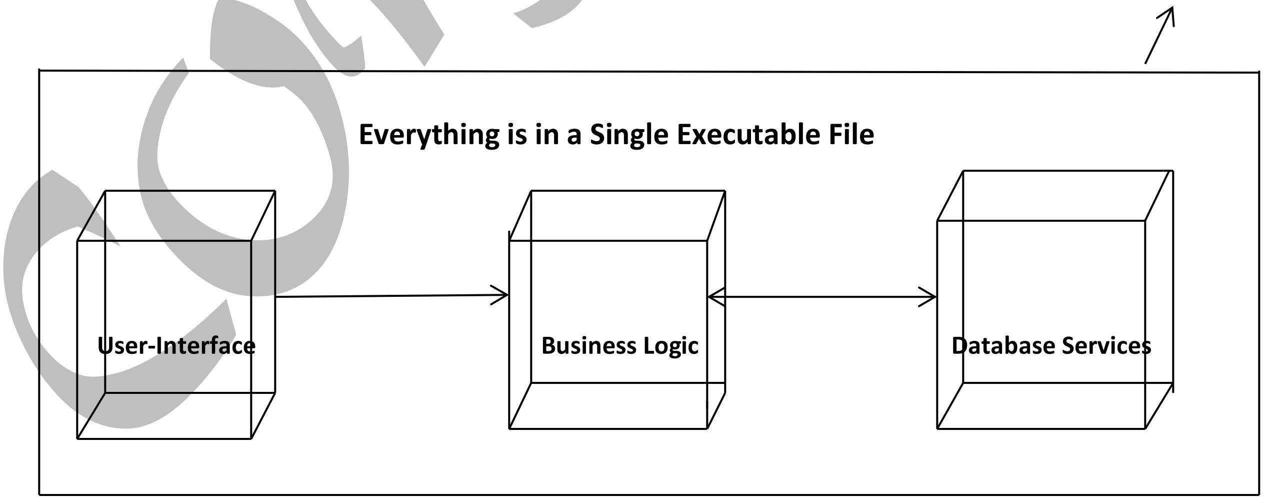
3).Three/Multi/n-tier Architecture.

1).Single-Tier Architecture.

OR

Monolithic Architecture.

Time Technique



1). Single-Tier Architecture are the most common is Desktop application.

2). It is also known as Monolithic Architecture.

3). In this architecture, everything runs on a Single-user Machine.

4). In Single-Tier Architecture all the three elements of an application :- User-Interface, Data Storage, Business Rule are packed in a Single executable.

5). If the Multi-user wants to use the application, the application has to installed all each and every user machine.

6). Even the database is copied to each and every machine. Because of the changes made in one database cannot be reflected in another database.

7). The machine on which these Monolithic application need to distributed have to be very powerful and highly configured because all the three layer(tier) have to be processed in a single machine.

2).*Two- Tier Architecture.*

OR

Client-Server Model

1). While Single-Tier Architecture designed for Desktop, whereas Two- Tier Architecture commonly designed for N/W. Where user access a Central Data Source.

2). This Architecture is also known as Client-Server Architecture.

3).With two-tier architecture the database don’t reside on the user’s computer. Instead, there is a database server that handles the management of data.

4).A client application, which resides the user’s machine is used to interact with the server.

5). The client-server relationship allows processing to be shared by both machines.

6). The database server takes care of storing and retrieve data for Multiple-user.

7). The difference from Single-user is that data resides on a different machines which accessed by the client.

8). In two-tier architecture business rule can either be applied on the client side or on the server side.

Based on this fact Two-tier Architecture is classified into two(2) types:-

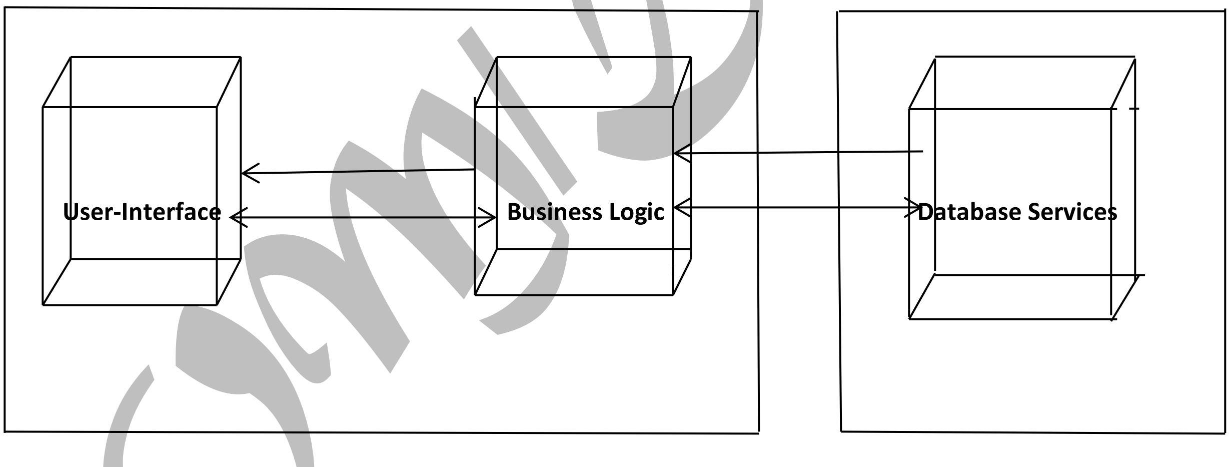
a).Thick Client.

b).Thin Client.

a).Thick Client.

1). If the business rule is applied by the client-side. This type of client-server model is known as Thick Client.

2).

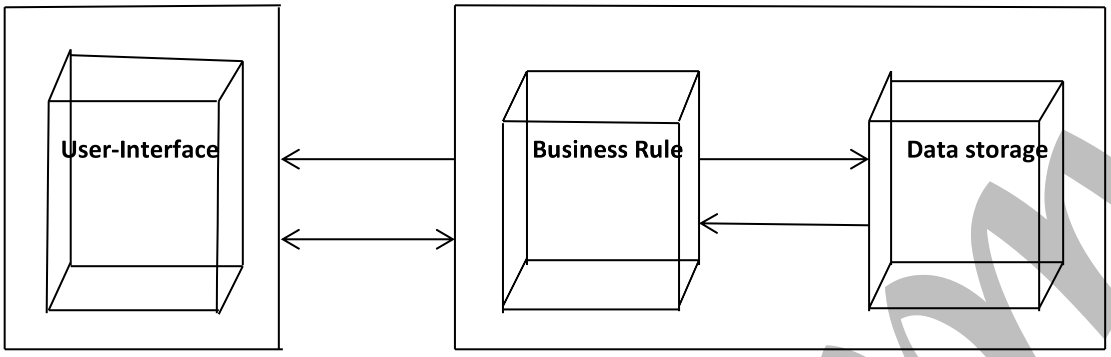


3). In Thick Client, server is generally only doing the job for retrieving and storing data while most of the validation are performed on the client itself.

b).Thin Client.

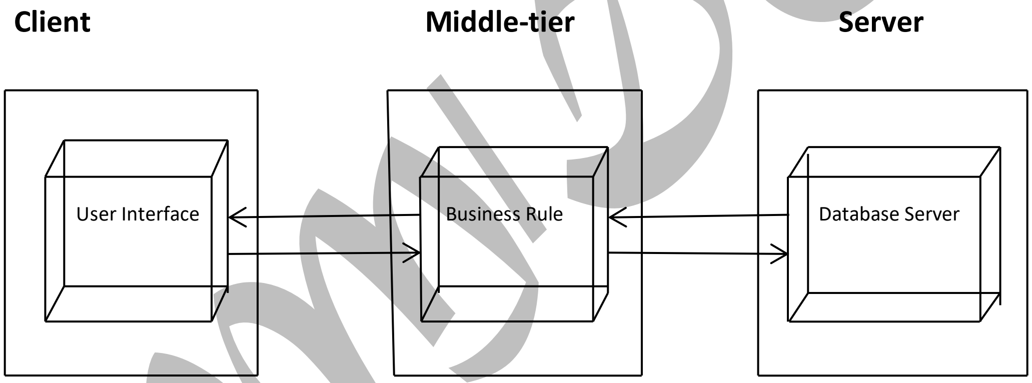
1). In Thin Client Architecture most of the business rules are applied on the server side and client is only handling user-interface issues.

2).



3). Thin Client Server Architecture enables the developer to change business rule with much more ease. Business rules needs to be change only on the server and all the client need not to be change.

3).*Three/Multi/n-tier Architecture*



1). To create a distributed application that avoid the problems of N/W bottle-necks at the data base server, we should consider using n-tier architecture.

2). This type of application split the solution across several computer.

3). The user interface still reside on the user's computer, and database remain on the data sever.

However, the business rule or data services objects are placed into a component or separate application that reside on the server. Such an application server the two computer perform.

4). In three-tier architecture application is divided into 3(three) elements.

User-Services, Data Services, Business Services where the user-service layer handles all the user-interface issue and handling of data import.

The business services layer manages the handling of business rule or Business logic applied and service layer including, storing and retrieving the data from the database.

2.8 Need for COM in Middle-Tier

In three-tier architecture business rule are generally applied on Middle-tier.

All the business rule cannot be grouped together into a single executable file. Since, the most big organization this strategy will not be suitable as even for a single component all the business rule have to be updated and re-compiled.

As distributed three-tier in architecture in nature, given popularity, COM emerge as the favoured technology.

Due to COM it was possible for developer to create separate component for separate services. These component are independent to each other.

In a three-tier architecture COM component are applied in Middle-tier business rules they are used to implement business services.

These component on independent of each other and have distinct functionality.

3.1 Messaging

1). Messaging is the process of sending completely encapsulated set of data between two application component.

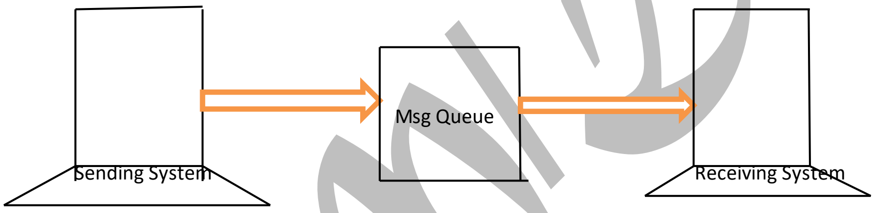
2). Messaging is the loose communication channel between two component.

3). Message can transfer in one or both direction.

4). Message component can exist in same process, in the different process on the same computer, in the different process on the different computer with completely encapsulated set of data.

3.2 Message Queue

1). Queue are the Buffering mechanism that makes Ms-MQueue (Microsoft-Message Queue) an effective tool for creating effective queue.



2). Queue store a message unit an appropriate application sees fit to retrieve the message.

3). Queue makes it possible for various component to communicate they even through they execute that different time at various location within enterprise or machine.

4). Most message should be placed from which standard message are received.

5). Message Queues are usually carried by application. However, some application might required that Ms-MQueue which create specific queue.

6). Message Queue can be public or private.

Public:- Public Queue are registered in the active direction.

Private:- Private Queue are visible only on the Ms-MQueue Machine on which they are created.

3.3 Ms-MQueue (Microsoft-Message Queue)

1). Ms-MQueue is the product of window based application development.

2). It provide protocol independent communication between queue.

3). Ms-MQueue is a system component which uses database technology to manage Robust Queue behavior and facility.

3.4 Clustering

1).The grouping of work-station (computer) in the N/W is called Cluster.

2).Cluster is a group of fully inter-connected active terminal with server that works together and appear as a single-user system to it's.

3).A grouping of inter-connected whole computer working together as a unified computing resource that can create the illusion of being one machine.

4).Clustering the most popular process for Distributed implementation system.

3.5 Benefits of Designing Cluster-based Distributed Computer.

1). In a Cluster, there is no limit on the number of machine. Cluster can be two or more than 100 computers.

2). Cluster are Flexible in nature. We can add any number of computer at any time without affecting the N/W.

3). Failure of one or more computer never affect cluster N/W because all the computer in cluster are active terminals i.e. Function of failure computer is taken over by some other computer.