CLIENT SERVER COMMUNICATIONS MIDDLEWARE COMPONENTS

The communication middleware software provides the means through which clients and servers communicate to perform specific actions. It also provides specialized services to the client process that insulates the front-end applications programmer from the internal working of the database server and network protocols. In the past, applications programmers had to write code that would directly interface with specific database language (generally a version of SQL) and the specific network protocol used by the database server.

**Multitier architecture**

In software engineering, multi-tier architecture (often referred to as n-tier architecture) is a client–server architecture in which the presentation, the application processing, and the data management are logically separate processes. For example, an application that uses middleware to service data requests between a user and a database employs multi-tier architecture. The most widespread use of multi-tier architecture is the three-tier architecture.

N-tier application architecture provides a model for developers to create a flexible and reusable application. By breaking up an application into tiers, developers only have to modify or add a specific layer, rather than have to rewrite the entire application over. There should be a presentation tier, a business or data access tier, and a data tier.

The concepts of layer and tier are often used interchangeably. However, one fairly common point of view is that there is indeed a difference, and that a layer is a logical structuring mechanism for the elements that make up the software solution, while a tier is a physical structuring mechanism for the system infrastructure.

**Three tiers**

Three-tier[2] is a client–server architecture in which the user interface, functional process logic ("business rules"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms. It was developed by John J. Donovan in Open Environment Corporation (OEC), a tools company he founded in Cambridge, MA.

The three-tier model is a software architecture and a software design pattern. Apart from the usual advantages of modular software with well-defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently as requirements or technology change. For example, a change of operating system in the presentation tier would only affect the user interface code.
Typically, the user interface runs on a desktop PC or workstation and uses a standard graphical user interface, functional process logic may consist of one or more separate modules running on a workstation or application server, and an RDBMS on a database server or mainframe contains the computer data storage logic. The middle tier may be multi-tiered itself (in which case the overall architecture is called an "n-tier architecture").

**Three-tier architecture has the following three tiers:**

**Presentation tier**
This is the topmost level of the application. The presentation tier displays information related to such services as browsing merchandise, purchasing, and shopping cart contents. It communicates with other tiers by outputting results to the browser/client tier and all other tiers in the network.

**Application tier** (business logic, logic tier, data access tier, or middle tier)
The logic tier is pulled out from the presentation tier and, as its own layer, it controls an application’s functionality by performing detailed processing.

**Data tier**
This tier consists of database servers. Here information is stored and retrieved. This tier keeps data neutral and independent from application servers or business logic. Giving data its own tier also improves scalability and performance.
A front-end web server serving static content, and potentially some cached dynamic content. In web based application, Front End is the content rendered by the browser. The content may be static or generated dynamically.
A middle dynamic content processing and generation level application server, for example Java EE, ASP.NET, PHP platform.
A back-end database, comprising both data sets and the database management system or RDBMS software that manages and provides access to the data.
TWO TIER VS. THREE TIER ARCHITECTURE

A two-tier client/server application architecture is implemented when a client talks directly to a server, with no intervening server. It is typically used in small environments of less than 50 users. Generally two-tier architecture separates the user interface and the business logic onto one computer (Tier1) and the database server is onto another computer (Tier2). In the case of an alignment database, “business logic” is more aptly phrased as “data logic”. Datalogic is defined as the set of rules and specifications that data or queries must meet before being submitted to the database server. The data logic tier may also include calculation, data analysis or anything that does not relate specifically to the user interface. A three-tier client/server application architecture separates the user interface, the data logic and the database server (data services) onto three separate tiers. Generally this
means that there are three computers involved, though it is possible to program the application so that the data logic part runs as an independent process on either the client or server computer.