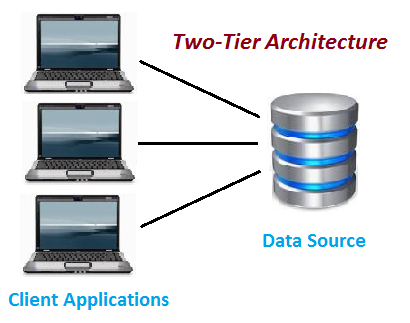
**Two-Tier Architecture:**

The two-tier is based on Client Server architecture. The two-tier architecture is like client server application. The direct communication takes place between client and server. There is no intermediate between client and server. Because of tight coupling a 2 tiered application will run faster.

[](http://www.softwaretestingclass.com/what-is-difference-between-two-tier-and-three-tier-architecture/two-tier-architecture/)

Two-Tier Architecture

The above figure shows the architecture of two-tier. Here the direct communication between client and server, there is no intermediate between client and server.

The Two-tier architecture is divided into two parts:

**1) Client Application (Client Tier)  
2) Database (Data Tier)**

On client application side the code is written for saving the data in the database. Client sends the request to server and it process the request & send back with data. The main problem of two tier architecture is the server cannot respond multiple request same time, as a result it cause a data integrity issue.

**Advantages:**

1. Easy to maintain and modification is bit easy
2. Communication is faster

**Disadvantages**:

1. In two tier architecture application performance will be degrade upon increasing the users.
2. Cost-ineffective

**Three-Tier Architecture:**

**Three-tier architecture** typically comprise a presentation tier, a business or data access tier, and a data tier. Three layers in the three tier architecture are as follows:

**1) Client layer**  
**2) Business layer**  
**3) Data layer**

**1) Client layer:**

It is also called as *Presentation layer* which contains UI part of our application. This layer is used for the design purpose where data is presented to the user or input is taken from the user. For example designing registration form which contains text box, label, button etc.

**2) Business layer:**

In this layer all business logic written like validation of data, calculations, data insertion etc. This acts as a interface between Client layer and Data Access Layer. This layer is also called the intermediary layer helps to make communication faster between client and data layer.

**3) Data layer:**

In this layer actual database is comes in the picture. Data Access Layer contains methods to connect with database and to perform insert, update, delete, get data from database based on our input data.

[](http://www.softwaretestingclass.com/what-is-difference-between-two-tier-and-three-tier-architecture/three-tier-architecture/)

Three-tier Architecture

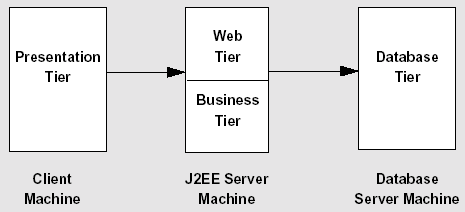
**Advantages**

1. High performance, lightweight persistent objects
2. Performance – Because the Presentation tier can cache requests, network utilization is minimized, and the load is reduced on the Application and Data tiers.
3. High degree of flexibility in deployment platform and configuration
4. Better Re-use
5. Improve Data Integrity
6. Improved Security – Client is not direct access to database.
7. Easy to maintain and modification is bit easy, won’t affect other modules
8. In three tier architecture application performance is good.

**Disadvantages**

1. Increase Complexity/Effort

**J2EE Architecture:**



The client machine supports web browsers, Java applets, and stand-alone applications. A client application may be as simple as a command line program running as an administrator client or a graphical user interface created from Java Swing or AWT (Abstract Window Toolkit) components. Regardless, J2EE applications encourage ***thin clients*** in the presentation tier. A thin client is a lightweight interface that does not perform database queries, implement business logic, or connect to legacy code. These types of "heavyweight" operations preferably belong to other tiers.

The J2EE server machine is the center of the architecture. This middle tier contains web components and business objects managed by the application server. The web components dynamically process user requests and construct responses to client applications. The business objects implement the logic of a business domain. Both components are managed by a J2EE application server that provides important system services for these components, such as security, transaction management, naming and directory lookups, and remote connectivity. By placing these services under control of the J2EE application server, client components focus only on presentation logic. And, business objects are easier for developers to write. Furthermore, the architecture *encourages* the separation of business logic from presentation logic (or model from view).

The database server machine handles the database back end. This includes mainframe transactions, databases, Enterprise Resource Planning (ERP) systems, and legacy code.