**What is N-Tier?**

An **N-Tier Application** program is one that is distributed among three or more separate computers in a distributed network.

The most common form of n-tier is the 3-tier Application, and it is classified into three categories.

* User interface programming in the user’s computer
* Business logic in a more centralized computer, and
* Required data in a computer that manages a database.

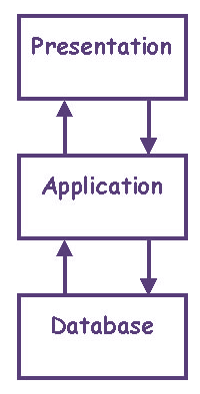
This architecture model provides Software Developers to create Reusable application/systems with maximum flexibility.

In **N-tier, “N”**refers to a number of tiers or layers are being used like – **2-tier, 3-tier or 4-tier, etc**. It is also called “**Multi-Tier** **Architecture”**.

**The n-tier architecture** is an industry-proven software architecture model. It is suitable to support enterprise level client-server applications by providing solutions to scalability, security, fault tolerance, reusability, and maintainability. It helps developers to create flexible and reusable applications.

**N-Tier Architecture**

A diagrammatic representation of an n-tier system depicts here – presentation, application, and database layers.

N Tier Architecture Diagram

These three layers can be further subdivided into different sub-layers depending on the requirements.

Some of the popular sites who have applied this architecture are

* MakeMyTrip.com
* Sales Force enterprise application
* Indian Railways – IRCTC
* Amazon.com, etc.

Some common terms to remember, so as to understand the concept more clearly.

* **Distributed Network:**It is a network architecture, where the components located at network computers coordinate and communicate their actions only by passing messages. It is a collection of multiple systems situated at different nodes but appears to the user as a single system.
  + It provides a single data communication network which can be managed separately by different networks.
  + An example of Distributed Network– where different clients are connected within LAN architecture on one side and on the other side they are connected to high-speed switches along with a rack of servers containing service nodes.
* **Client-Server Architecture:**It is an architecture model where the client (one program) requests a service from a server (another program) **i.e.** It is a request-response service provided over the internet or through an intranet.

In this model, **Client** will serve as one set of program/code which executes a set of actions over the network. While **Server**, on the other hand, is a set of another program, which sends the result sets to the client system as requested.

* + In this, client computer provides an interface to an end user to request a service or a resource from a server and on the other hand server then processes the request and displays the result to the end user.
  + An example of Client-Server Model– an ATM machine. A bank is the server for processing the application within the large customer databases and ATM machine is the client having a user interface with some simple application processing.
* **Platform:**In computer science or software industry, a platform is a system on which applications program can run. It consists of a combination of hardware and software that have a built-in instruction for a processors/microprocessors to perform specific operations.
  + In more simple words, the platform is a system or a base where any applications can run and execute to obtain a specific task.
  + An example of Platform – A personal machine loaded with Windows 2000 or Mac OS X as examples of 2 different platforms.
* **Database:**It is a collection of information in an organized way so that it can be easily accessed, managed and updated.
  + Examples of Database – MySQL,[SQL](https://www.guru99.com/sql.html)Server, and Oracle Database are some common Db’s.

**Types of N-Tier Architectures**

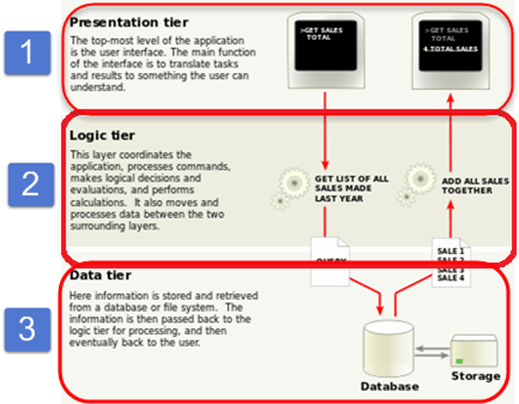
There are different types of N-Tier Architectures, like **3-tier Architecture, 2-Tier Architecture and 1- Tier Architecture.**

First, we will see 3-tier Architecture, which is very important.

**3-Tier Architecture**

By looking at the below diagram, you can easily identify that **3-tier architecture**has three different layers.

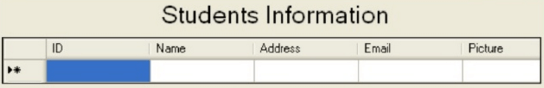
* Presentation layer
* Business Logic layer
* Database layer



3 Tier Architecture Diagram

Here we have taken a simple example of student form to understand all these three layers. It has information about a student like – Name, Address, Email, and Picture.

**User Interface Layer or Presentation Layer**



**Presentation Layer**

private void DataGrid1\_SelectedIndexChanged(object sender, System.EventArgs e)

{

// Object of the Property layer

clsStudent objproperty=new clsStudent();

// Object of the business layer

clsStudentInfo objbs=new clsStudentInfo();

// Object of the dataset in which we receive the data sent by the business layer

DataSet ds=new DataSet();

// here we are placing the value in the property using the object of the

//property layer

objproperty.id=int.Parse(DataGridl.SelectedItem.Cells[1].Text.ToString());

// In this following code we are calling a function from the business layer and

// passing the object of the property layer which will carry the ID till the database.

ds=objbs.GetAllStudentBsIDWise(objproperty);

// What ever the data has been returned by the above function into the dataset

//is being populate through the presentation laye.

txtId.Text=ds.Tables[0].Rows[0][0].ToString();

txtFname.Text=ds.Tables[0].Rows[0][1].ToString();

txtAddress.Text=ds.Tables[0].Rows[0][2].ToString();

txtemail.Text=ds.Tables[0].Rows[0][3].ToString();

**Code Explanation**

* The above code defines the basic designing of a front end view of applications as well as calling of the functions of other layers so that they can be integrated with each other.

**Business Access Layer –**

This is the function of the business layer which accepts the data from the application layer and passes it to the data layer.

* Business logic acts as an interface between Client layer and Data Access Layer
* All business logic – like validation of data, calculations, data insertion/modification are written under business logic layer.
* It makes communication faster and easier between the client and data layer
* Defines a proper workflow activity that is necessary to complete a task.

// this is the function of the business layer which accepts the data from the

//application layer and passes it to the data layer.

public class clsStudentInfo

{

public DataSet GetAllStudentBsIDWise(clsStudent obj)

{

DataSet ds=new DataSet();

ds=objdt.getdata\_dtIDWise(obj);// Calling of Data layer function

return ds;

}

}

**Explanation of code**

The code is using the function of business layer, which will accept the data for the application layer and passed it to the data layer. The Business layer codes act as a mediator between the functions defined in the presentation layer and data layer and calling the functions vice -versa.

**Data Access Layer**

This is the data layer function, which receives the data from the business layer and performs the necessary operation into the database.

// this is the datalayer function which is receiving the data from the business

//layer and performing the required operation into the database

public class clsStudentData // Data layer class

{

// object of property layer class

public DataSet getdata\_dtIDUise(clsStudent obj)

{

DataSet ds;

string sql;

sql="select \* from student where Studentld=" +obj.id+ "order by Studentld;

ds=new DataSet();

//this is the datalayer function which accepts the sql query and performs the

//corresponding operation

ds=objdt.ExecuteSql(sql);

return ds;

}

}

**Explanation of code**

The code defines in dataset layer above accepts the entire request: requested by the system and performing the required operations into the database.

**2-Tier Architecture:**

It is like Client-Server architecture, where communication takes place between client and server.

In this type of software architecture, the presentation layer or user interface layer runs on the client side while dataset layer gets executed and stored on server side.

There is no Business logic layer or immediate layer in between client and server.

**Single Tier or 1-Tier Architecture:**

It is the simplest one as it is equivalent to running the application on the personal computer. All of the required components for an application to run are on a single application or server.

Presentation layer, Business logic layer, and data layer are all located on a single machine.

**Advantages and Disadvantages of Multi-Tier Architectures**

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| * Scalability | * Increase in Effort |
| * Data Integrity | * Increase in Complexity |
| * Reusability |  |
| * Reduced Distribution |  |
| * Improved Security |  |
| * Improved Availability |  |

Hence, it is a part of a program which encrypts real-world business problems and determines how data can be updated, created, stored, or changed to get the complete task done.

**N-Tier Architecture Tips and Development**

Considering the software professionals must have a full control on all the layers of the architecture, tips on n-tier architecture are given as below

1. Try to decouple layers from another layer as much as possible by using a technique like soap XML.
2. Use some automated tools to generate a mapping between a business logic layer and a relational database layer (data layer). Tools that can help in modeling these mapping techniques are – Entity Framework and Hibernate for .Net etc.
3. In client presenter layer, put a common code for all the clients in a separate library as much as possible. This will maximize the code reusability for all types of clients.
4. A cache layer can be added into an existing layer to speed up the performance.