**Basic Points of Java**

Java is a platform independent, more powerful, secure, high performance, multithreaded programming language. Here we discuss some points related to Java.

**Define byte**

**Byte code** is the set of optimized instructions generated during compilation phase and it is more powerful than ordinary pointer code.

**Define JRE**

The **Java Runtime Environment (JRE)** is part of the Java Development Kit (JDK). It contains a set of libraries and tools for developing Java application. The Java Runtime Environment provides the minimum requirements for executing a Java application.

**Define JVM**

**JVM** is set of programs developed by sun Micro System and supplied as a part of the JDK for reading line by line line of byte code and it converts into a native understanding form of operating system. The Java language is one of the compiled and interpreted programming language.

**Garbage Collector**

The **Garbage Collector** is the system Java program which runs in the background along with a regular Java program to collect un-Referenced (unused) memory space for improving the performance of our applications.

**Note:**Java programming does not support destructor concept in place of destructor, we have garbage collector program.

**Define an API**

An **API (Application Programming Interface)** is a collection of packages, a package is the collection of classes, interfaces and sub-packages. A sub-package is a collection of classes, Interfaces and sub sub packages etc.

Java programming contains user friendly syntax so that we can develop effective applications. in other words if any language is providing user friendly syntax, we can develop error free applications.

**Definition of JIT**

JIT is the set of programs developed by SUN Micro System and added as a part of JVM, to speed up the interpretation phase.

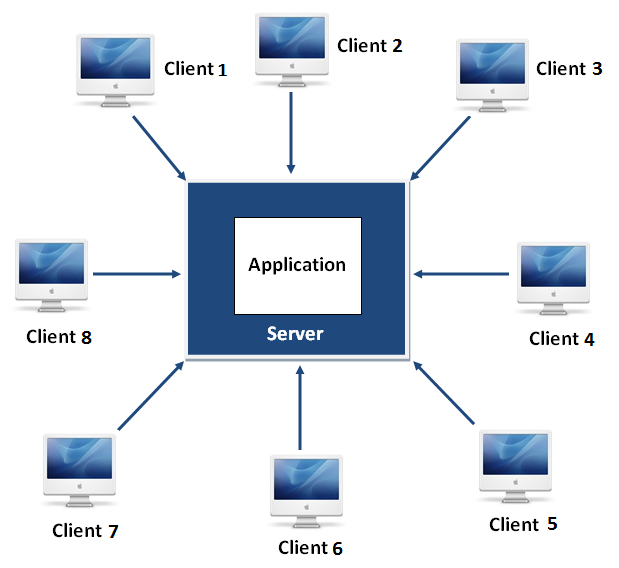
**Network based application**

Network based application are mainly classified into two types.

* Centralized Applications
* Distributed Applications

**Centralized applications**

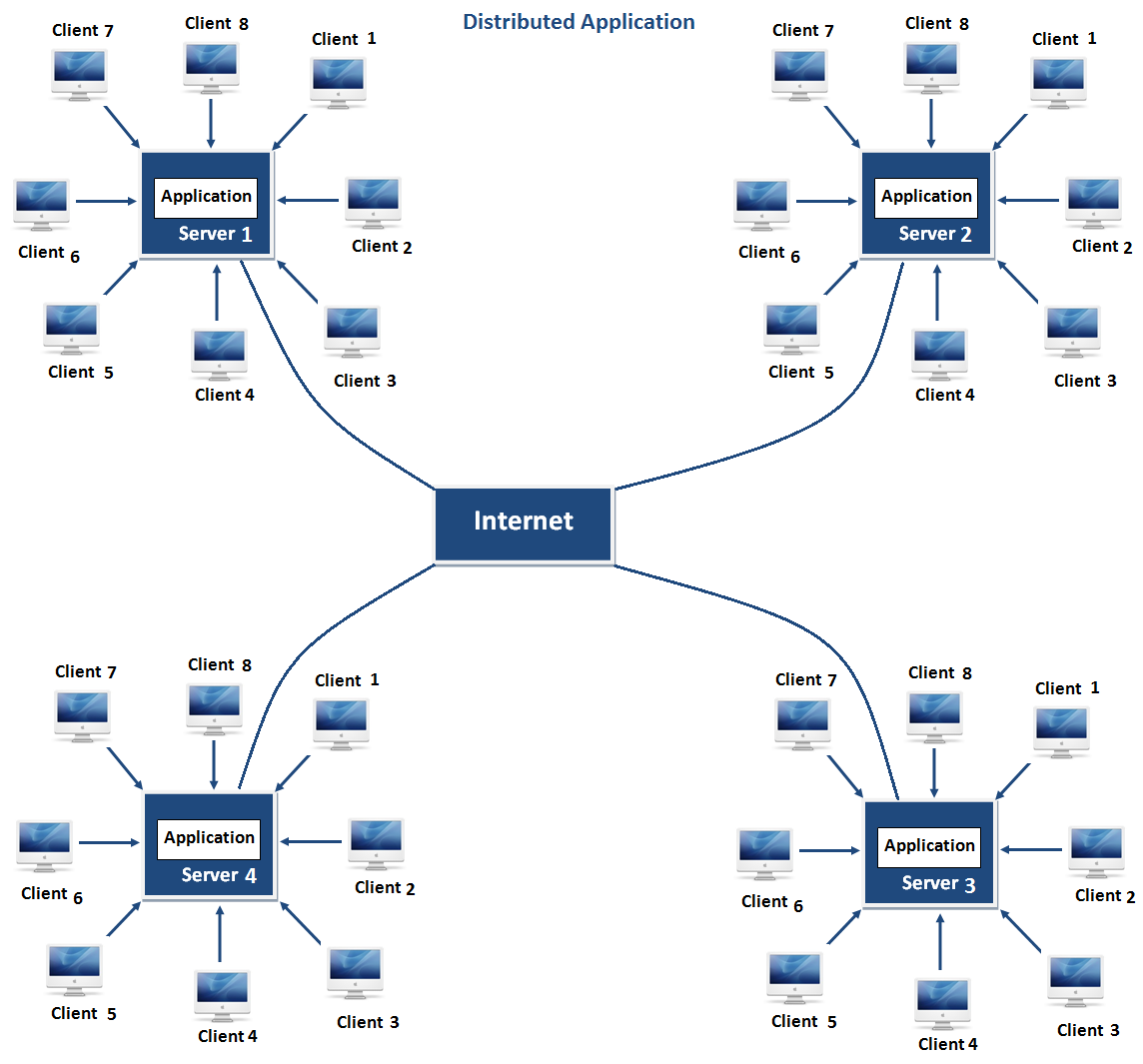
In this scenario multiple client system depends on single server system.



The major drawback in this architecture is if any problem occurred on server system that will be reflected on every client system.

**Distributed applications**

In this scenario multiple client system are depends on multiple server system so that even problem occurred in one server will never be reflected on any client system.



**Note:**In this architecture same application is distributed in multiple server system.

Java is a very powerful language can be used to developed both client server architecture and distributed architecture based application.