1. What is JNDI :

http://www.javaworld.com/article/2076888/core-java/jndi-overview--part-1--an-introduction-to-naming-services.html

The Java Naming and Directory Interface (JNDI) is an application programming interface (API) for accessing different kinds of naming and directory services. JNDI is not specific to a particular naming or directory service, it can be used to access many different kinds of systems including file systems; distributed objects systems like CORBA, Java RMI, and EJB; and directory services like LDAP, Novell NetWare, and NIS+.

JNDI is similar to JDBC in that they are both Object-Oriented Java APIs that provide a common abstraction for accessing services from different vendors. While JDBC can be used to access a variety of relational databases, JNDI can be used to access a variety of of naming and directory services. Using one API to access many different brands of a service is possible because both JDBC and JNDI subscribe to the same architectural tenet: Define a common abstraction that most vendors can implement. The common abstraction is the API. It provides an objectified view of a service while hiding the details specific to any brand of service. The implementation is provided by the vendor, it plugs into the API and implements code specific to accessing that vendor's product.

JNDI provides two APIs and one SPI. JNDI has a naming API that allows Java applications to access naming systems like CORBA's Naming services and a directory API that extends the naming service to provide access to directory services like LDAP. JNDI also has a SPI (Service-Provider Interface) which is a programming model that vendors use to write JNDI plug-ins or implementations for their specific product. Each vendor's plug-in is called a *service-provider. A service-provider implements the JNDI APIs so that a Java application can access that vendor's product. For the most part, JNDI hides the implementation details of the a service-provider so that Java developer that uses JNDI can use the same objects and method regardless of the brand of naming or directory service accessed. This is the real power behind APIs like JDBC and JNDI: They provide one programming model for accessing many different products; there is no need to learn a different programming model every time a different product is used.*