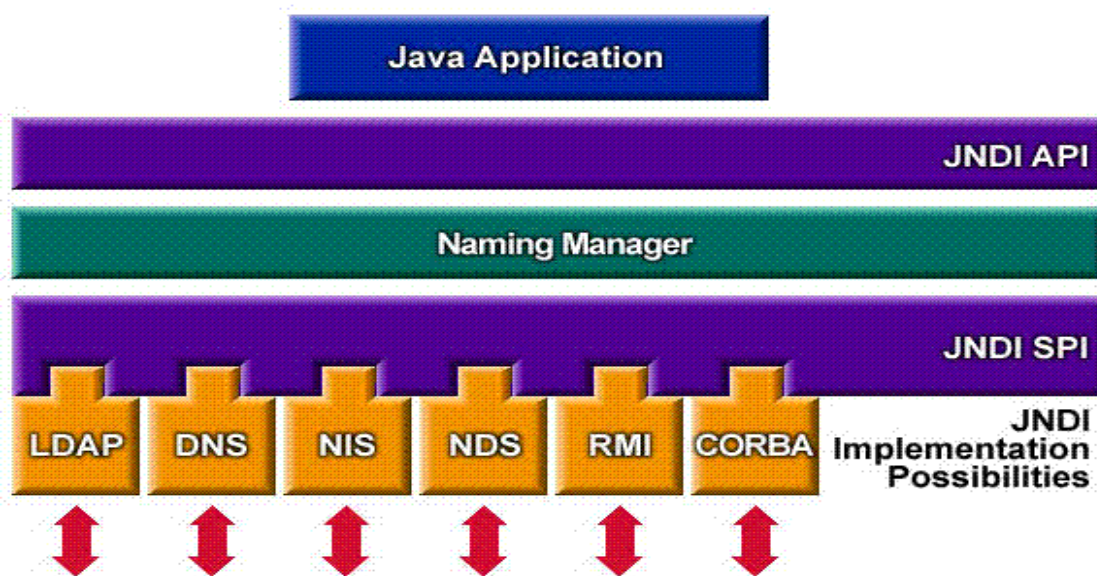


## 题目：JNDI 原理

jndi 是 Java 命名和目录接口(Java Naming and Directory Interface, JNDI) 的简称.从一开始就一直是 Java 2 平台企业版的核心技术之一。在 JMS, JMail,JDBC,EJB 等技术中,就大量应用的这种技术。JNDI 可访问的现有的目录及服务有: DNS、XNam 、Novell 目录服务、LDAP(Lightweight Directory Access Protocol 轻型目录访问协议)、 CORBA 对象服务、文件系统、Windows XP/2000/NT/Me/9x 的注册表、RMI、DSML v1&v2、NIS。

jndi 诞生的理由似乎很简单。随着分布式应用的发展,远程访问对象访问成为常用的方法。虽然说通过 Socket 等编程手段仍然可实现远程通信,但按照模式的理论来说,仍是有其局限性的。RMI 技术, RMI-IIOP 技术的产生,使远程对象的查找成为了技术焦点。JNDI 技术就应运而生。JNDI 技术产生后,就可方便的查找远程或是本地对象。

JNDI 的架构



JNDI 的架构与 JDBC 的架构非常类似.JNDI 架构提供了一组标准命名系统的 API,这些 API 在 JDK1.3 之前是作为一个单独的扩展包。jndi.jar(通过这个地址下载),这个基础 API 构建在与 SPI 之上。这个 API 提供如下五个包

```
javax.naming
javax.naming.directory
```

`javax.naming.event`

`javax.naming.ldap`

`javax.naming.spi`

在应用程序中,我们实际上只使到用以上几个包的中类.具体调用类及通信过程对用户来说是透明的.JNDI API 提供了访问不同 JNDI 服务的一个标准的统一的实现,其具体实现可由不同的 **ServiceProvider** 来完成。前面讲的为第一层 JNDI API 层.最下层为 JNDI SPI API 及其具体实现。它包括了几个增强和下面的命名/目录服务提供者:

LDAP(Lightweight Directory Access Protocol)服务提供者

CORBA COS (Common Object Request Broker Architecture Common Object Services) 命名服务提供者

RMI(Java Remote Method Invocation)注册服务提供者

DNS(Domain Name System)服务提供者.

FSSP(File System Service Provider)文件系统服务提供者

典型的运用场景：数据源

tomcat 中配置一个数据源，程序就可以通过 `java.sql` 接口去访问数据库，不管底层的数据库是什么类型。