## RMI (Remote method invocation)

远程方法调用 ejb基于此协议

## RPC (Remote Procedure Call)

远程过程调用 dubbo基于此协议

EJB

部署分布式应用程序 基于rmi实现

JMS (java message service)

java队列消息协议 消息由头(head) 属性(property) 主体(body) 组成支持两种消息传递模型: 点对点(ptp) 和 发布/订阅(pub/sub) 简单来说是一对一传递和一对多传递ActiveMQ基于此协议

## ActiveMQ

消息队列支持jms provider实现，支持 in-vm(管道通讯)，tcp,ssl,nio,Jgroup,jxta多种传输协议

## SOA

是一种粗粒度、开放式、松耦合的服务架构，一个面向服务的架构样式,支持异构,支持变化可理解为成型软件的oop思想 oop是开发的范式说得通俗一点就是，系统中分为三种角色：服务提供者、服务使用者和注册中心，提供者发布服务到注册中心，使用者通过注册中心发现所需服务，然后与该服务的提供者绑定，并调用服务。其实就是接口开发思想

## MAVEN

Maven是一个项目管理工具，包含项目对象模型(粗略理解为项目级的oop思想)，对于mavne来说 一个软件工程的唯一标识是由开发者(GroupId)、项目名(artifactid)、版本号(version)共同决定，每个工程都可以打包为jar、war、ear、或pom(打包成pom是给其他项目继承所用，一般将parent项目打包成pom)。

maven有聚合和继承两个概念，聚合就是我知道你们，你们不知道我，继承是你们都知道我，而我不知道你们。

maven有三个独立的生命周期 构建(build),清理(clean),建立项目发布站点(site)

maven基本命令:

install:将包安装到本地仓库,供其他mavne使用(其他maven引用的都是本地仓库的项目.jar)

clean:清除target目录下的文件(编译后的文件)

compile：编译项目中的代码

deploy：将包发布到远程的maven仓库，并提供给其他开发者使用。

package:将项目打包到项目的target目录下

## GIT

一个开源分布式版本控制系统，有工作区(日常编辑的部分),暂存区(git add之后的部分),版本库(git commit后的部分),HEAD(当前版本指针) 常用指令:git pull origin master/git push origin master git add \*,git commit,git --reset hard(放弃本地所有修改)git branch

## SVN

项目版本控制，协同开发，资料备份的工具，每次commit前先update

SVN和GIT的区别:

git是分布式 svn不是GIT跟SVN一样有自己的集中式版本库或服务器。但，GIT更倾向于被使用于分布式模式，也就是每个开发人员从中心版本库/服务器上chect out代码后会在自己的机器上克隆一个自己的版本库。可以这样说，如果你被困在一个不能连接网络的地方时，就像在飞机上，地下室，电梯里等，你仍然能够提交文件，查看历史版本记录，创建项目分支

GIT把内容按元数据方式存储，而SVN是按文件

git分支和svn分支不同(git切换分支方便快捷)  
 GIT没有一个全局版本号，svn有

GIT的内容完整性要优于SVN，因为GIT的内容存储使用的是[SHA-1](http://en.wikipedia.org/wiki/SHA-1" \t "http://blog.jobbole.com/31444/_new)哈希算法。这能确保代码内容的完整性，确保在遇到磁盘故障和网络问题时降低对版本库的破坏

## Socket

理解为网络上的io流，负责请求接收网络信息，tcp/ip协议的实现，处于osi的表现层和传输层之间的抽象层,负责网络通信

## Mina

一个基于Socket的封装网络通信框架，对c端s端都进行了封装，用于开发高性能，高可扩展性的网络应用程序,采用java 1.7新特性NIO，能自动发送心跳包(判断连接是否断开),能自动监听客户端断开

## mysql

c和c++编写，关系型数据库 轻量级开源 没啥特别的

innodb和myisam引擎区别

innodb 支持事务功能，myisam 不支持。

Myisam 的执行速度更快，性能更好。

InnoDB 为行级锁，myisam 为表级锁。

## Hibernate和Mybatis

hibernate是对象关系映射的持久层框架，对jdbc进行封装，完全遵循JPA的实现规范，掌握较慢

mybatis 是轻量级orm持久层框架，掌握较快  
hibernate是全自动化，mybatis是半自动化:

hibernate完全可以通过对象关系模型实现对数据库的操作，拥有完整的JavaBean对象与数据库的映射结构来自动生成sql。而mybatis仅有基本的字段映射，对象数据以及对象实际关系仍然需要通过手写sql来实现和管理。

hibernate数据库移植性远大于mybatis:

hibernate加强了对象和数据库的映射，降低了耦合度而mysql的sql则耦合度较高

hibernate拥有完整的日志系统，mybatis则欠缺一些：

hibernate日志系统非常健全，涉及广泛，包括：sql记录、关系异常、优化警告、缓存提示、脏数据警告等；而mybatis则除了基本记录功能外，功能薄弱很多。

从sql优化的角度讲，mybatis要比hibernate要好

mybatis可直接控制sql，灵活性强，但同时也降低了开发的效率，对自己的sql有信心的人 mybatis的效率远高于hibernate

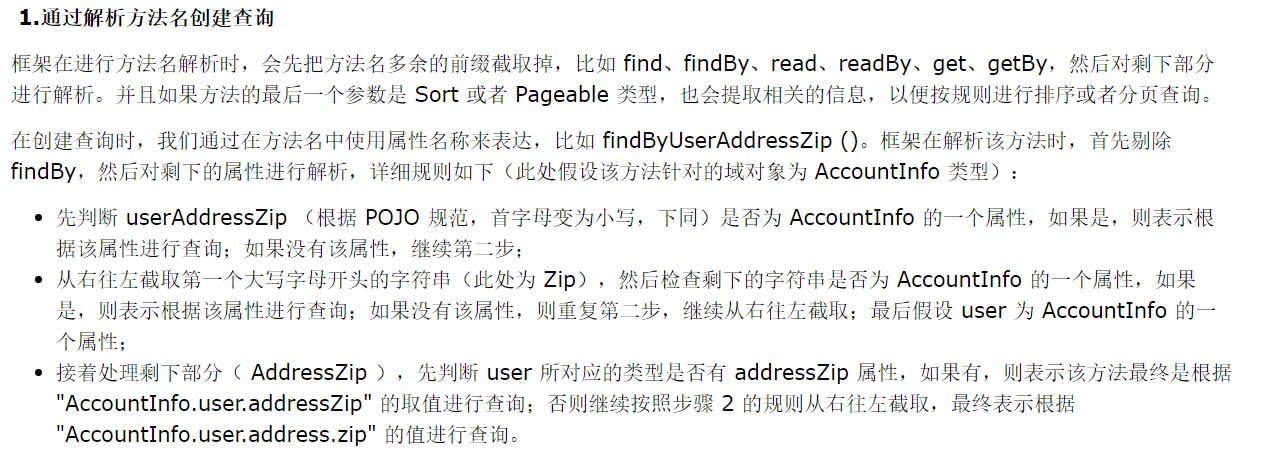
## Spring Boot

Spring web开发的一个子框架，强调约定大于配置来实现快速开发，该框架不需要以往xml配置，可达到引入jar包则用，如需定制一些配置可以配置在application.properties配置文件里。

spring boot 多用于创建独立的微服务

## Spring data jpa

spring data 下的一个jpa实现项目，严格遵循jpa标准，简化操作，实现repository<id类型,映射对象>(crudRepository)接口 在domain里面加上注解(和hibernate一样) 基于crud的操作不用再写，如需定制 ，findByUserId(String id) 意思就是 解析的时候先解析findBy后面的userid(整体) 如果该对象有这字段，则根据userid去查 参数即id ，如果里面没有userid属性，则从右往左解析到大写字母停止(Id)



## Spring mvc

轻量级web框架 没啥好说的 支持restful风格注解

## MongoDb 和 redis区别

都是非关系型数据库，不过mongodb更接近mysql 有索引，游标

mongodb新版本开始支持事务,redis也支持事务 不过比较弱，在大量的数据面前，redis打不过Mongodb 但这都不是瓶颈，都可以持久化，

mongodb 查询海量数据性能很好，可适用于大数据的存储，mongodb有空间索引，mongodb 集群比redis成熟