# 分布式通信框架-RMI

## 被弃用原因

1、使用JRMP（java远程信息控制协议），不能跨语言，仅支持java语言，对于多语言项目不适用；

2、底层使用jdk自带的serialize序列化传输信息，效率低且由于是jdk自带的，所以不可以重写

3、底层使用socket通信，属于阻塞Io（BIO通信），没有使用NIO通信，对于并发情况下有弊端

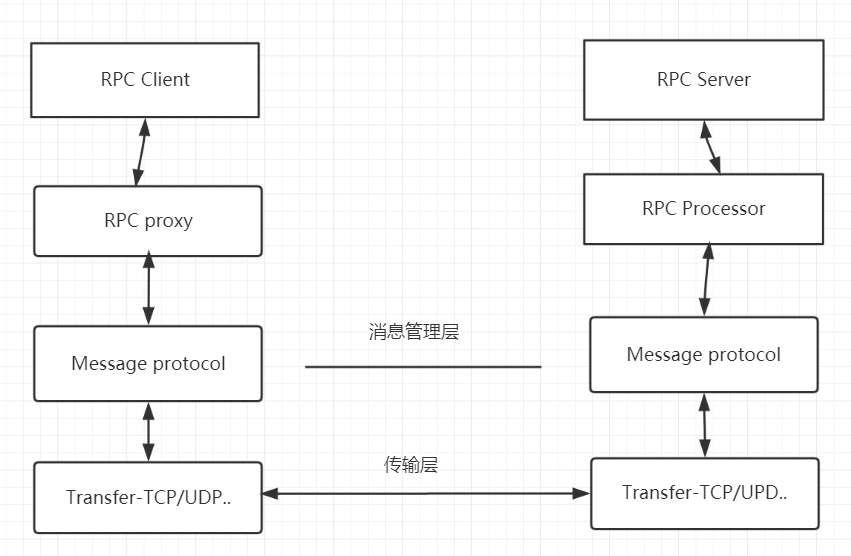
## 什么是RPC

Remote procedure call protocal 远程过程调用协议

RPC协议其实是一个规范。Dubbo、Thrif、RMI、Webservice、Hessain

网络协议和网络IO对于调用端和服务端来说是透明；

## 一个RPC框架包含的要素



# RMI的概述

RMI(remote method invocation) , 可以认为是RPC的java版本

RMI使用的是JRMP（Java Remote Messageing Protocol）, JRMP是专门为java定制的通信协议，所以踏实纯java的分布式解决方案

# 如何实现一个RMI程序

1. 创建远程接口， 并且继承java.rmi.Remote接口
2. 实现远程接口，并且继承：UnicastRemoteObject
3. 创建服务器程序： createRegistry方法注册远程对象
4. 创建客户端程序

# 如果自己要去实现一个RMI

1. 编写服务器程序，暴露一个监听， 可以使用socket
2. 编写客户端程序，通过ip和端口连接到指定的服务器，并且将数据做封装（序列化）
3. 服务器端收到请求，先反序列化。再进行业务逻辑处理。把返回结果序列化返回

