Rpc原理：

1. 服务消费方（client）调用以本地调用方式调用服务；
2. client stub接收到调用后负责将方法、参数等封装(序列化)成能够进行网络传输的消息体
3. client stub找到服务地址，并将消息发送到服务端；
4. server stub收到消息后进行解码；
5. server stub根据解码结果调用本地的服务
6. 本地服务执行并将结果返回给server stub
7. server stub将返回结果打包成消息并发送至消费方
8. client stub接收到消息，并进行解码（反序列化）；
9. 服务消费方得到最终结果。

公司内使用百度的ral。

数据结构：

1）接口名称

2）方法名

3）参数类型以及相应的参数值；

4）超时时间

5）requestID(nio为了回掉而设置的参数)

序列化：

序列化就是将数据结构或对象转换成二进制串的过程

（由于网络协议是基于二进制的，内存中的参数的值要序列化成二进制的形式，也就是序列化）

Rcp：

1. 基于tcp（复杂）
2. 基于HTTP（效率低一些）

netty