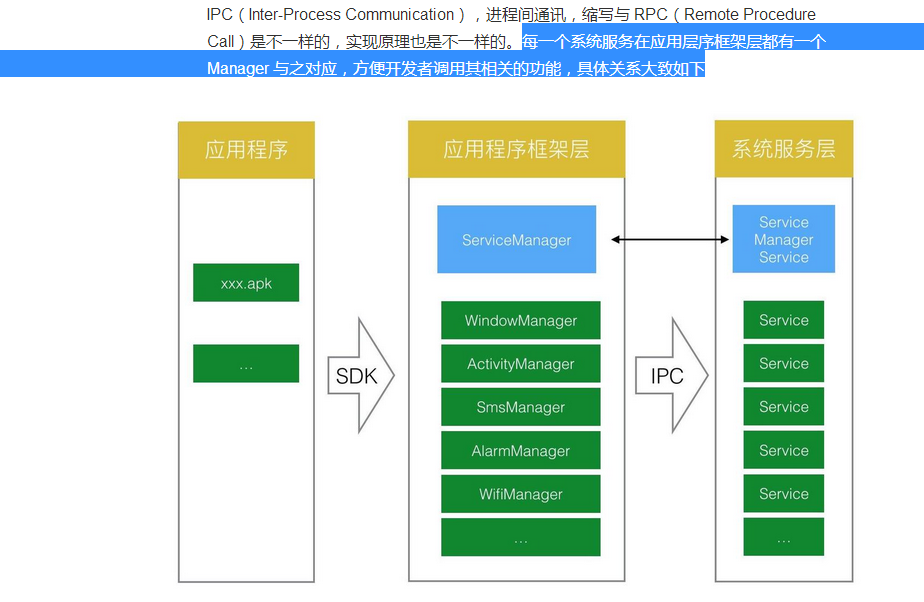
先从程序之间的调用和应用框架层、系统服务层之间的关系。

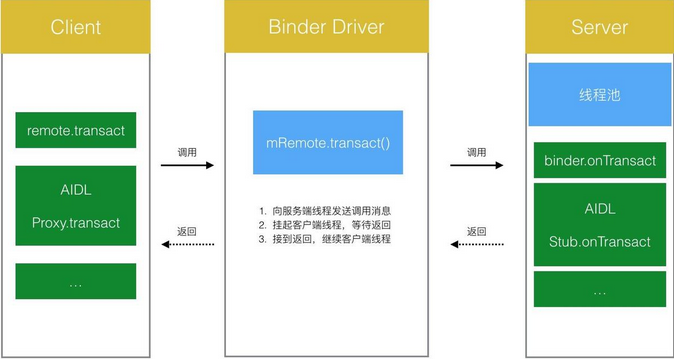


从这张图可以看出，要调用系统service，必须通过对应的manager，有了manager，我们就可以调用系统功能，所以说manager是一层保护service机制，而service只需要提供一个manager的接口。所以说基本流程是先定义manager，然后从对应service中来获取对应的manager。

所以说获取manager的通过写法是：

2.bmp

Binder的IPC（内部进程通信）。



注意这是个C/S结构，然后绿色的需要用户去实现，蓝色的是系统实现的。大概流程是client先调用binder driver，然后binder driver会调用server。这里的server是第一张图中service中onBind()方法返回的Ibinder对象，也就是说server的实现是有service返回来的对象来实现的。

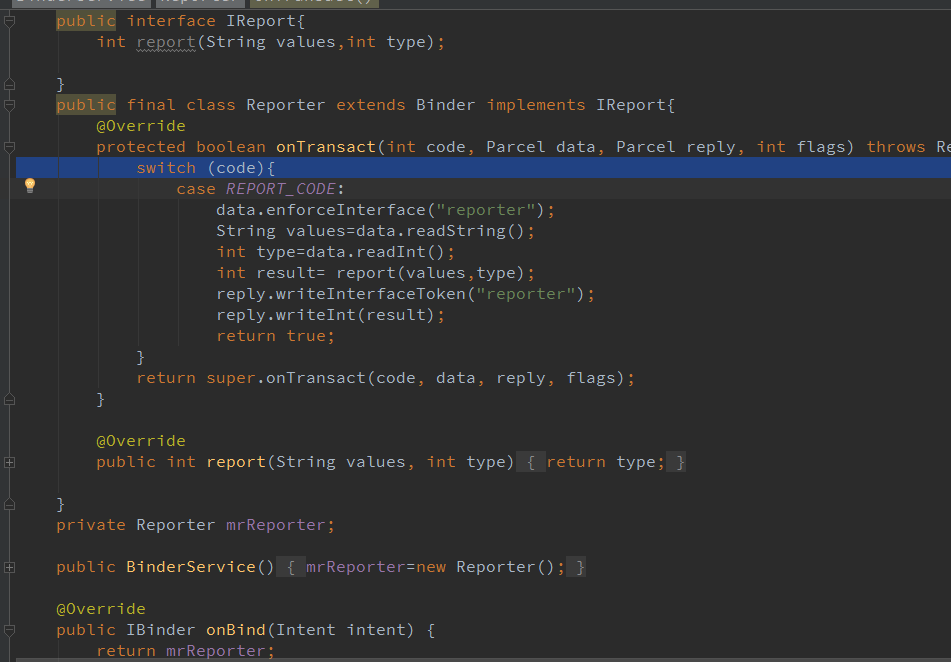
还有一个问题，就是server为什么用线程池而非单线程队列？

这个是因为如果多个客户端调用同一个service，如果单线程队列就会造成任务挤压，从而可能出现来不及响应的情况。这个是不允许的。（但是注意线程池是不安全的，而单线程是安全的，所以要做好多线程同步）

如何写一个完整的binder IPC机制，

首先从server入手，

server是service返回的Ibinder对象。那么我们就在service中onBind中返回这个binder，但是我们要自定义这个binder对象。





从这两方面看，客户端负责将数据组包，然后通过transact发送数据，然后server接到数据，解包，返回，然后客户端收到。

为什么socket性能传输比binder差，因为binder只有一个对象，而socket有client和server两个对象。