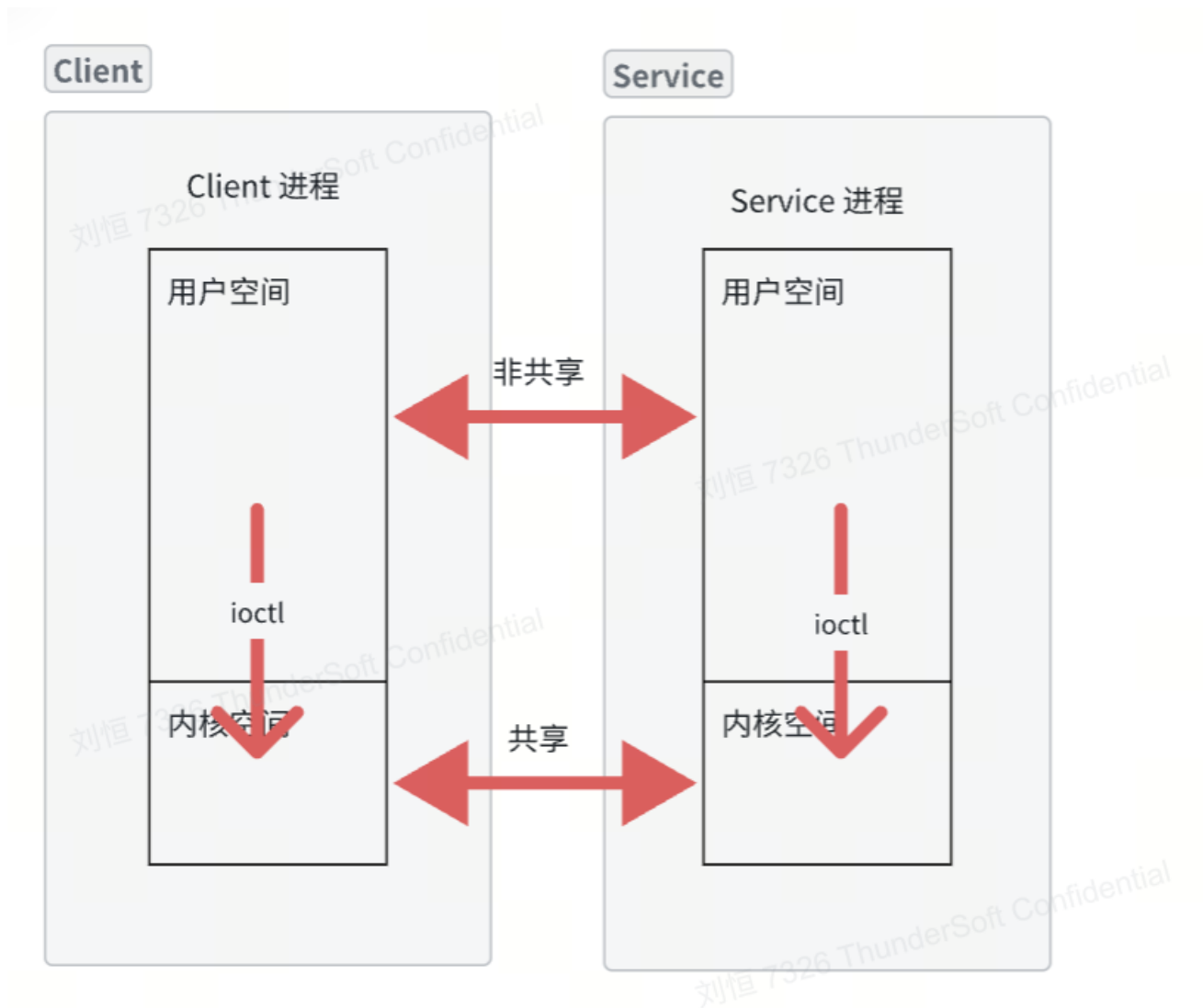


Android 平台上的一种跨进程通信（IPC）机制
note:Inter-Process Communication，进程间通信

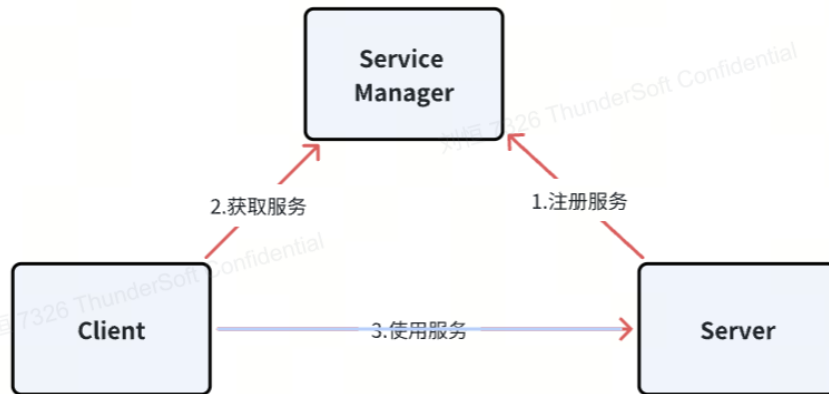
1.Ipc 原理



每个Android进程，只能运行在自己的进程所拥有的虚拟地址空间

Client进程与Service进程通信，利用进程间可共享的内核内空间来完成底层通信工作

2.Android C/S通信机制

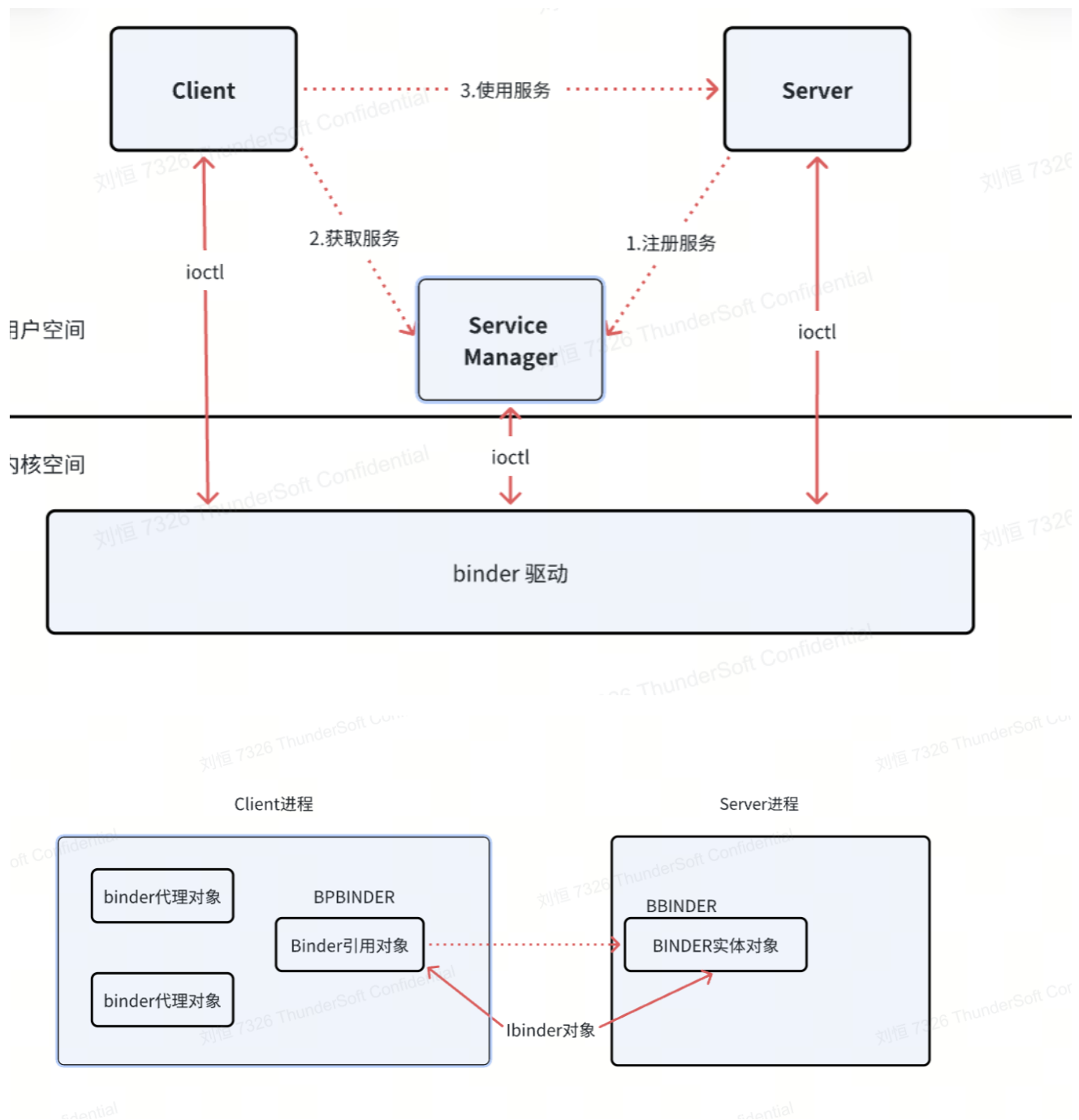


1.Server是service的提供者，提供的service需要在service manager进行注册。

Server可以注册一个或者多个service

2.Client是Service的使用者，使用前需要向Service manager申请服务

3.Client 获取到service后，可以调用Server服务了



binder机制的关键概念:

binder实体对象: binder服务的提供者, 类型是bbinder, 也称bbinder对象。(BNbinder)

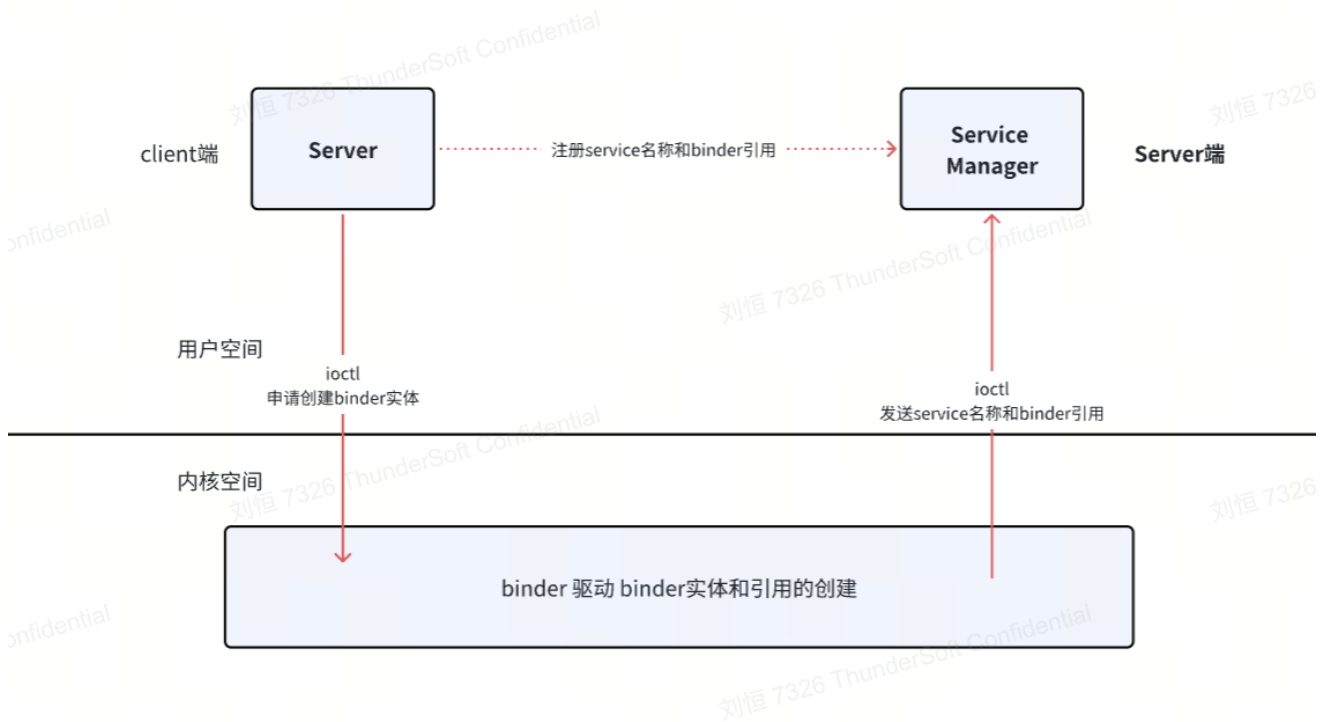
binder引用对象: binder实体对象在客户端进程的代表, 类型是bpbinder, 也称bpbinder对象

binder代理对象: 代理对象也称接口对象, 主要是为客户端的上层应用提供接口服务, 类型是 IInterface;

Android将binder引用对象和代理对象分开的好处是一个binder引用对象可以有多个代理对象, 方便应用层使用

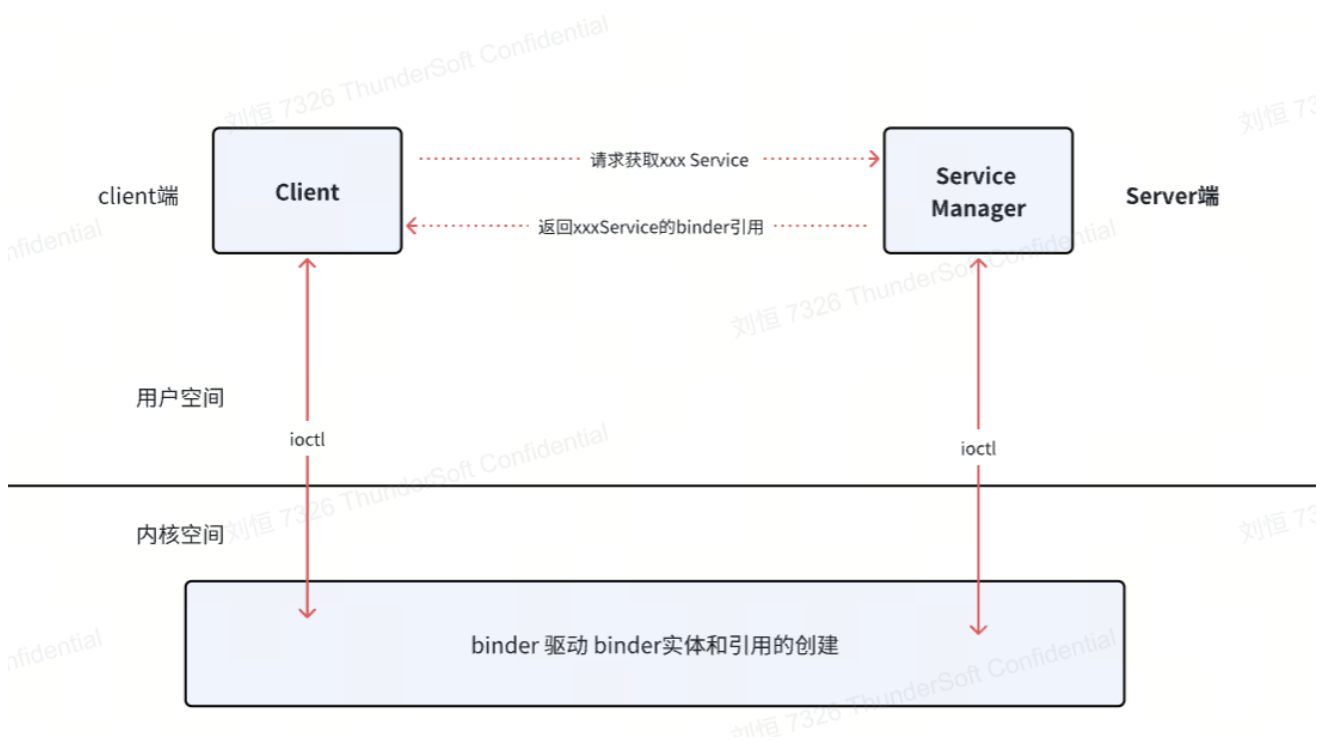
Ibinder对象: 实体对象和引用对象的统称, 因为bbinder和bpbinder都继承自Ibinder

1.注册服务



1. 首先server在自己的进程向binder驱动申请创建service的binder实体
2. binder驱动为这个Service创建位于内核的binder实体节点和binder引用
3. server通过binder驱动将Service名字和Service的binder引用打包发送给service manager。
4. service manager收到数据包之后，取出service的名字和binder引用，填入一张表内。

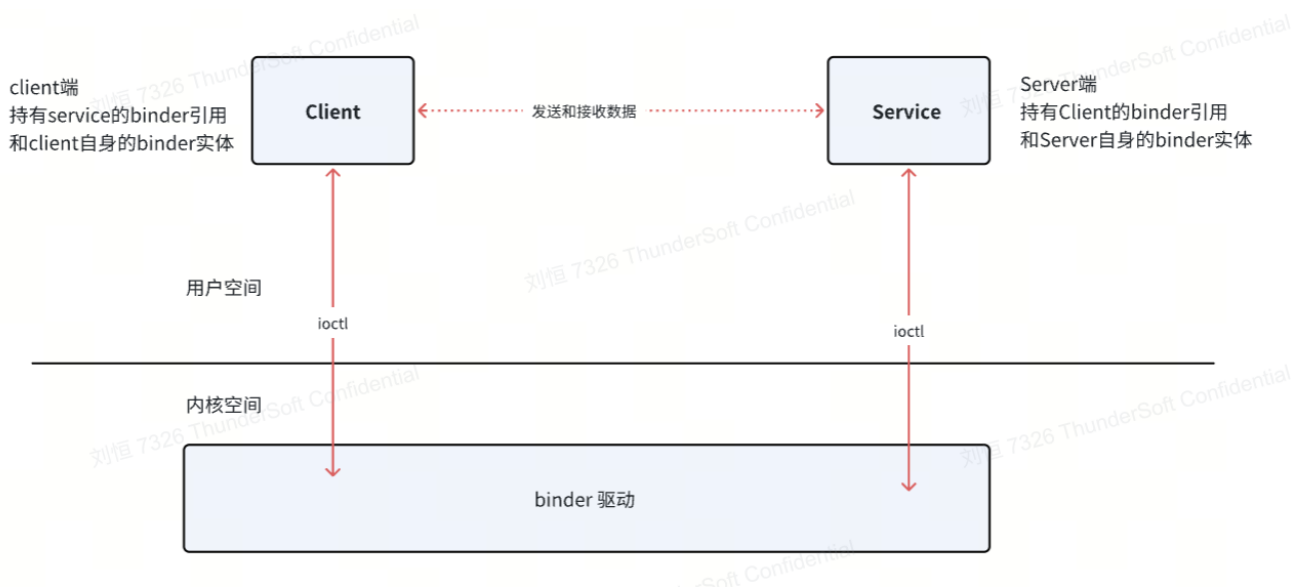
2. 获取服务



1. Client通过hanler值为0的引用找到service manager
2. 向service manager发送xxxservice的访问请求

- 3.Service manager从请求包中获取到xxx service的名称，通过表取出binder 引用，
- 4service manager返回binder引用。

3.使用服务



- 1.client 和service既是发送方也是接收方
- 2.发送方通过自身的binder实体进行发送操作，把数据通过接收方binder引用发送给接收方
- 3.binder驱动会处理这个发送请求，利用内核空间进程共享的机制：
 - 1.把发送方的数据放入写缓存（binder_wirte_read.wirte_buffer）（对于接收方来讲为读缓存区）
 - 2.接收方之前一直在阻塞状态中，当写缓存有数据，则会读取数据执行命令。
 - 3.接收方执行完成之后，会把返回结果同样采取写入缓存区（对于发送方，为读缓存区）