MICROOH 麦可网

Android-从程序员到架构师之路

出品人：Sundy
讲师：高焕堂（台湾）

http://www.microoh.com

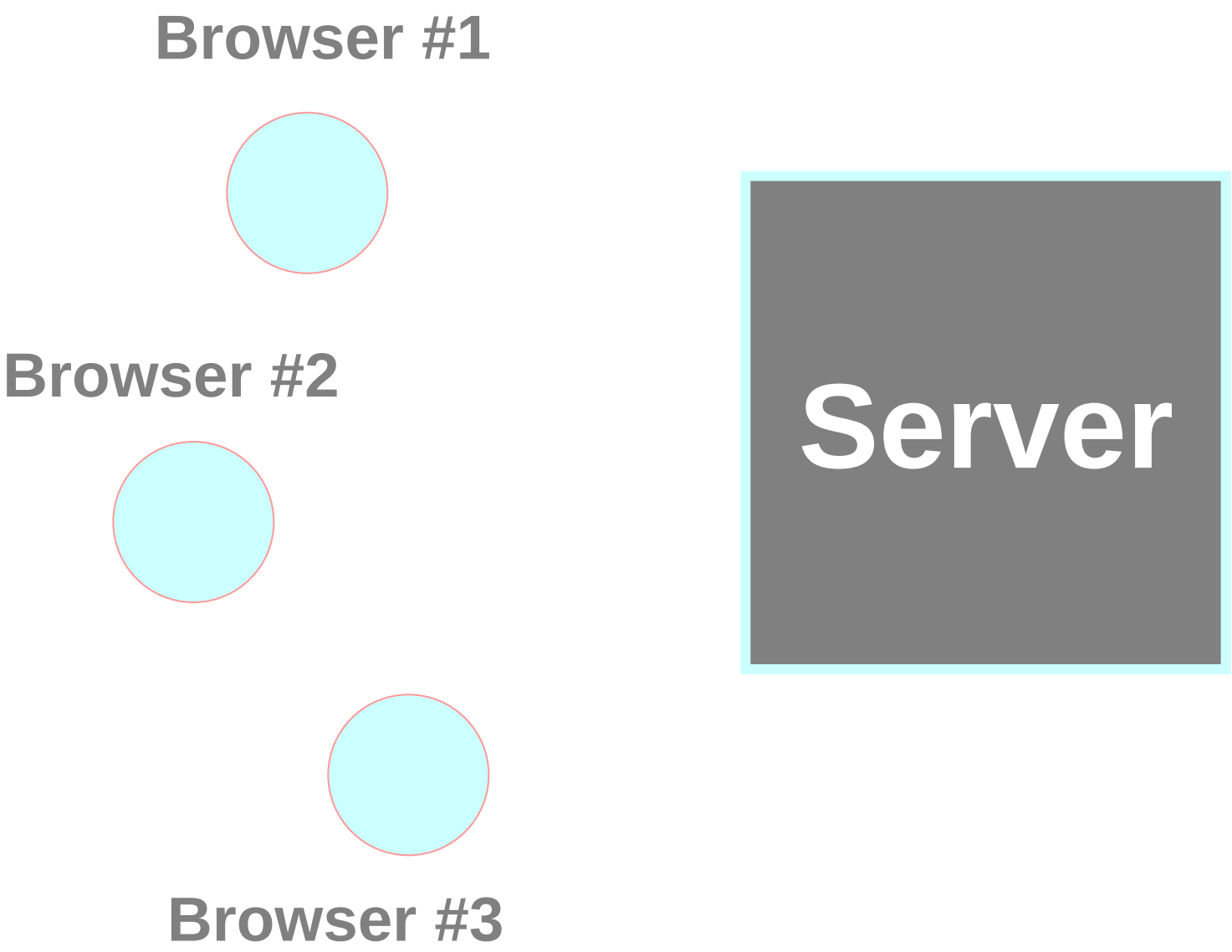
F01_b

观摩：Session模式与Proxy-Stub模式的搭配(b)

By 高煥堂

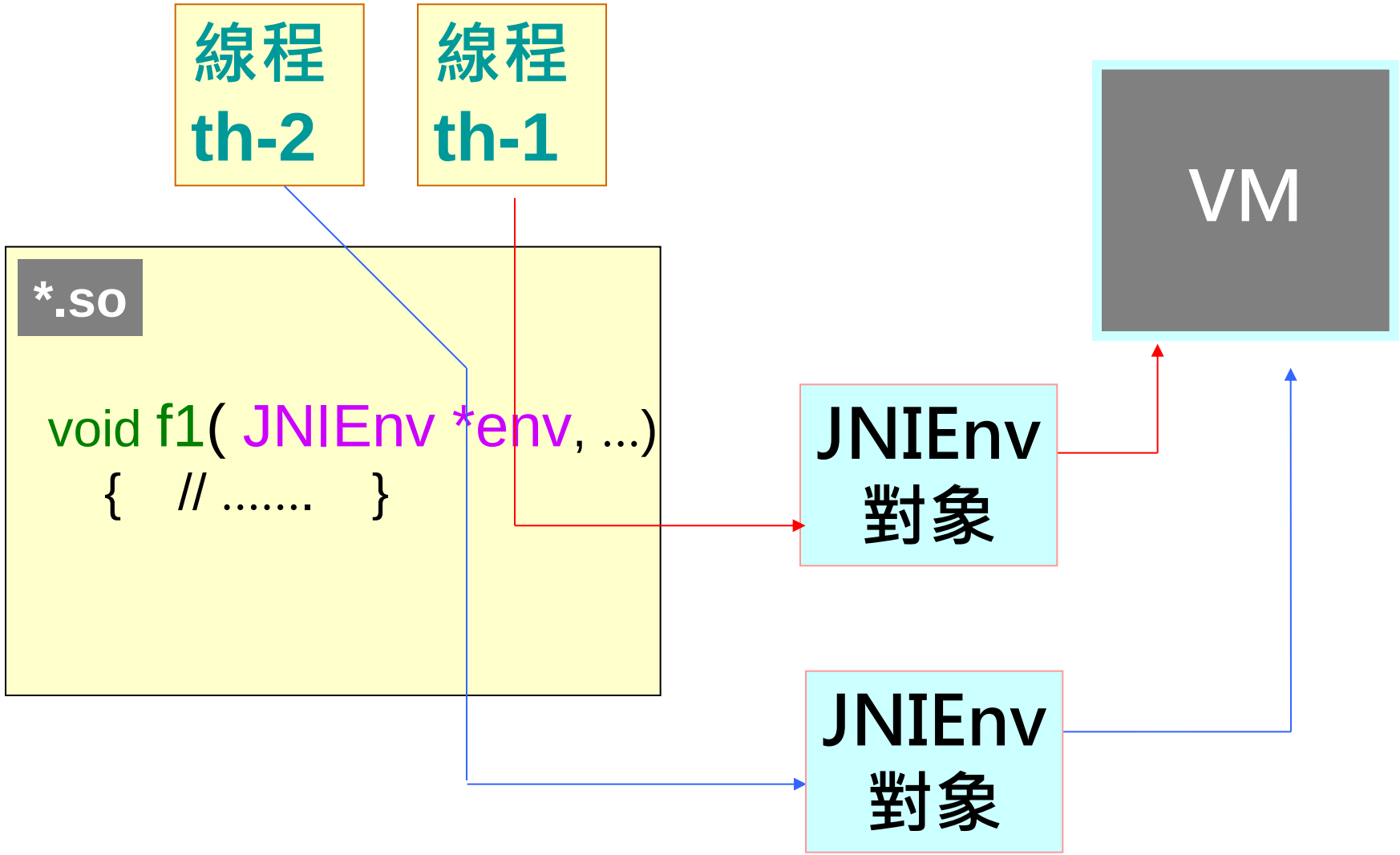
2、Session设计模式：以VM的JNIEnv对象为例

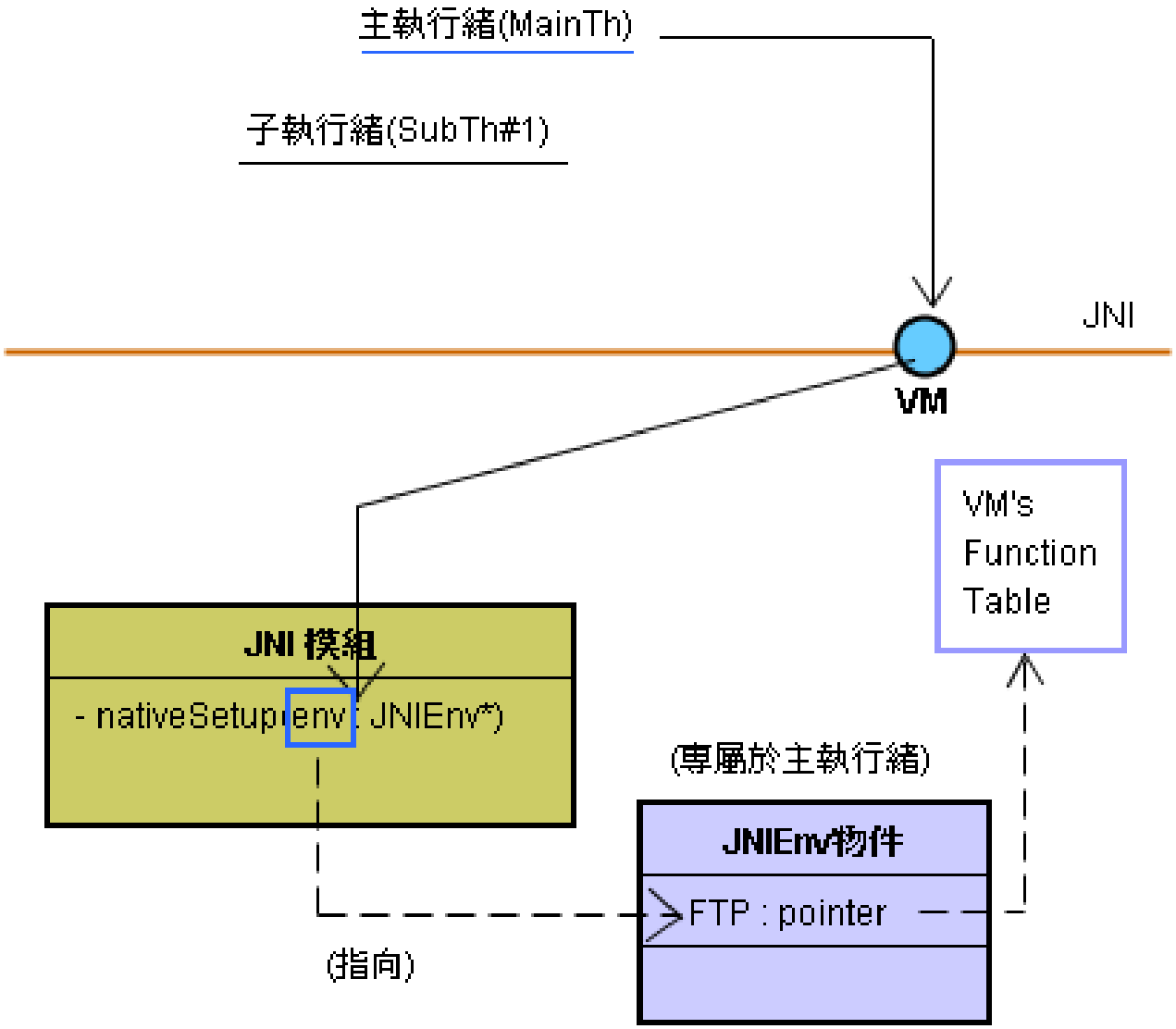
前言


Browser #1
Browser #2
Browser #3
Server

每一个Connection
都有一个私有的Session对象；

每一个线程进入VM
都有一个私有JNIEnv对象。


線程
th-2
線程
th-1
VM
*.so
void f1(JNIEnv *env, ...)
{ // ..... }
JNIEnv
對象
JNIEnv
對象

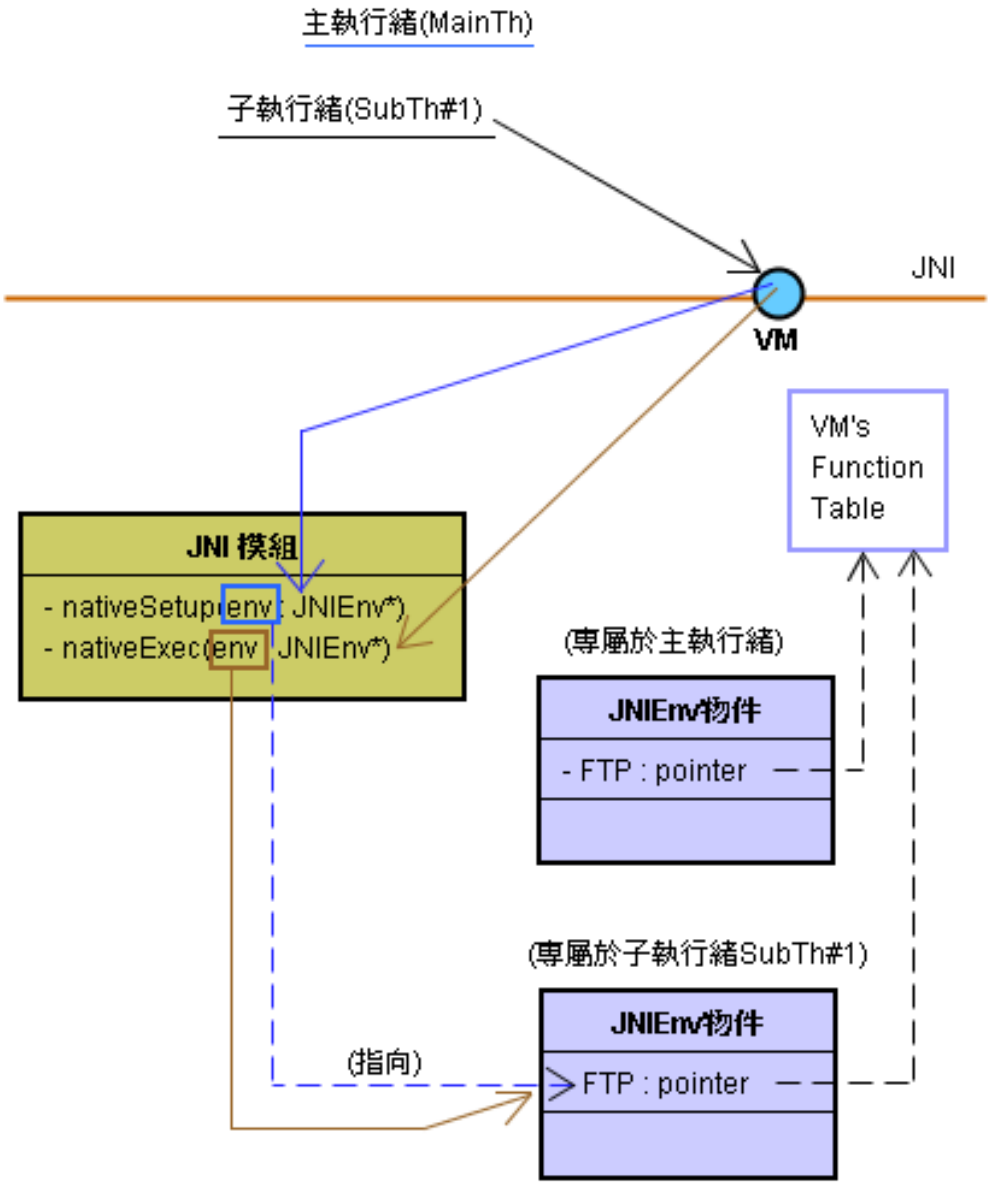

主執行緒(MainTh)
子執行緒(SubTh#1)
JNI
VM
VM's
Function
Table
JNI 模組
- nativeSetup(env : JNIEnv*)
(專屬於主執行緒)
JNIEnv物件
FTP : pointer
(指向)

- 每一个线程第一次进入VM调用本地函数时，VM会替它诞生一个相对映的JNIEnv对象。
- Java层的线程调用C层的本地函数时，该线程必然经过VM，且VM一定替它诞生相对映的JNIEnv对象。

线程不共享JNIEnv对象，成为”单线程”开发，不必烦恼线程安全问题，让本地函数的撰写单纯化了。

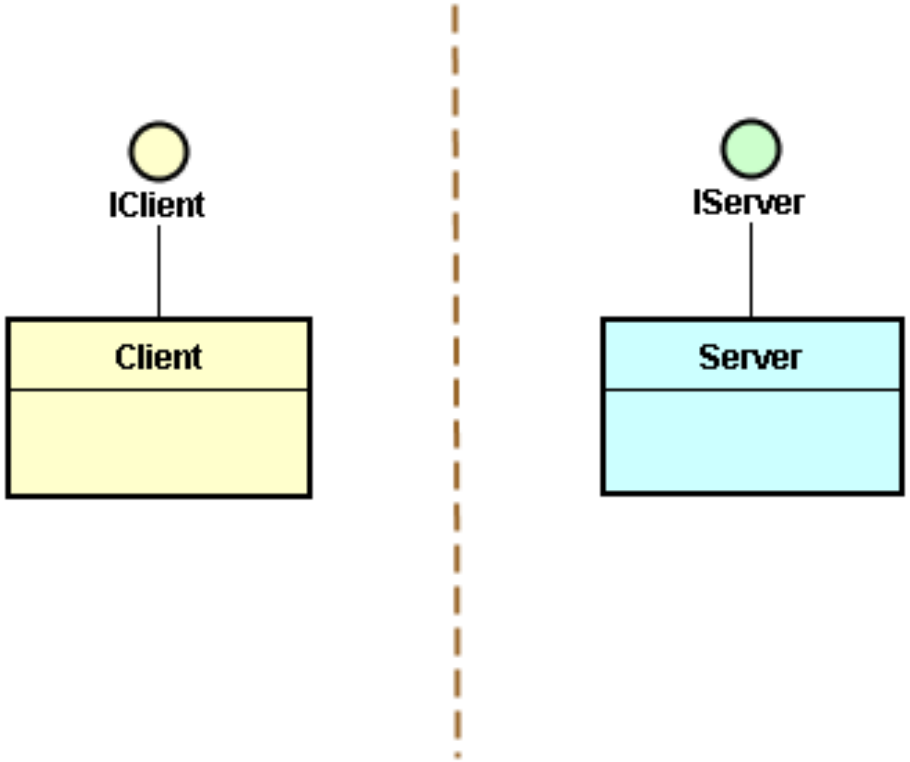
簡而言之…

- 在预设情形下，在某个线程第一次进入VM去执行JNI层C函数时，VM就会替它诞生专属的JNIEnv对象。只要该线程还存在着，就会一直保留它所专属的JNIEnv对象。

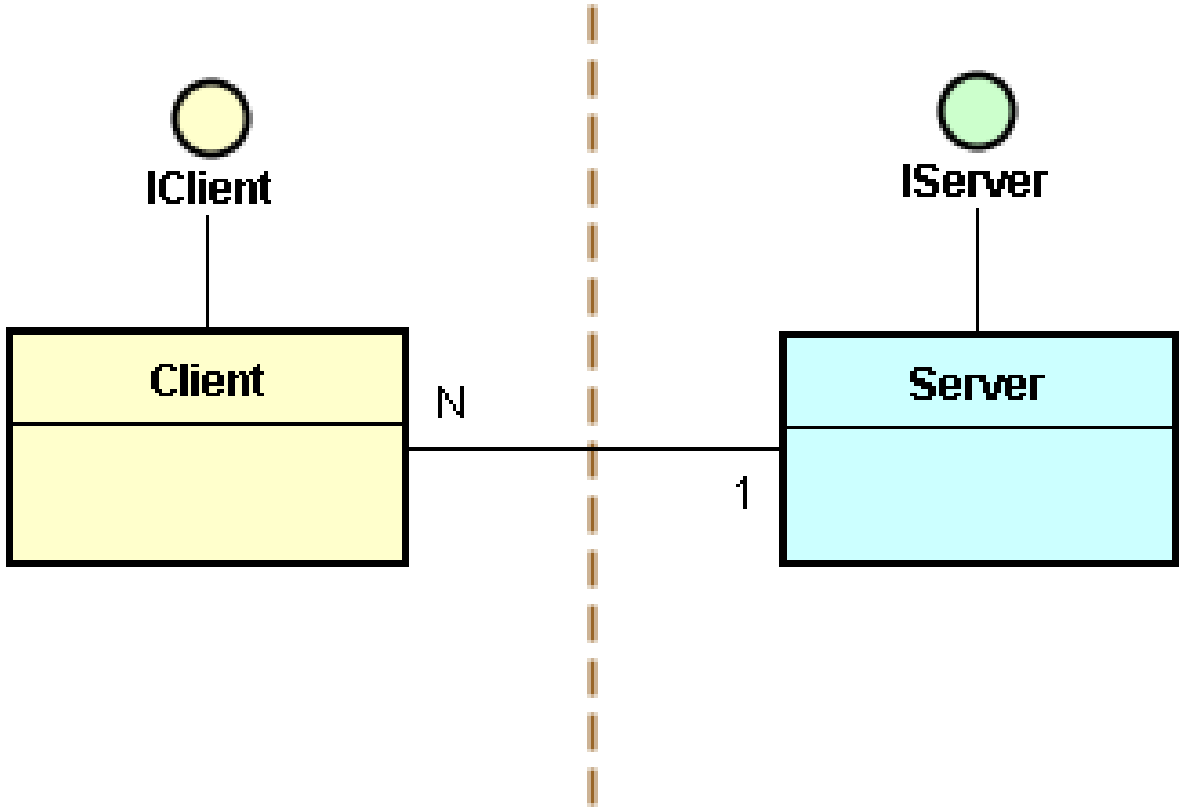

主執行緒(MainTh)
子執行緒(SubTh#1)
JNI
VM
VM's
Function
Table
JNI 模組
- nativeSetup(env : JNIEnv*)
- nativeExec(env : JNIEnv*)
(專屬於主執行緒)
JNIEnv物件
- FTP : pointer
(專屬於子執行緒SubTh#1)
JNIEnv物件
(指向)
FTP : pointer

3、Session设计模式：典型架构

- 首先复习一下Client-Server模式，其Client和Server各提供接口，如下图：



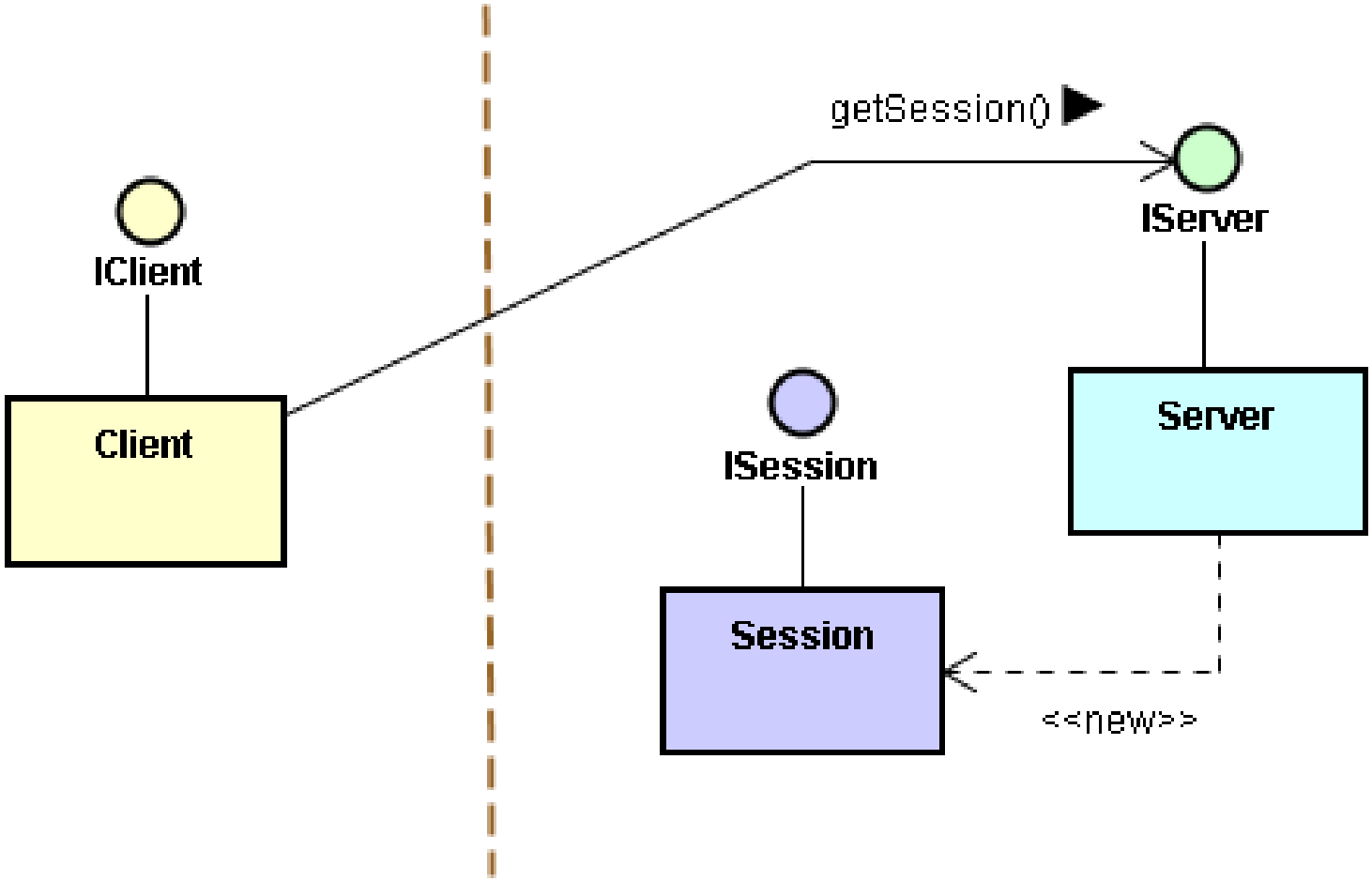
- 在这Client与Server之间透过接口互相沟通；而且多个 Client可同时与Server建立连结，取得Server的服务。所以，Client与Server之间是N:1的关系，如下图：


IClient
IServer
Client
N
Server
1

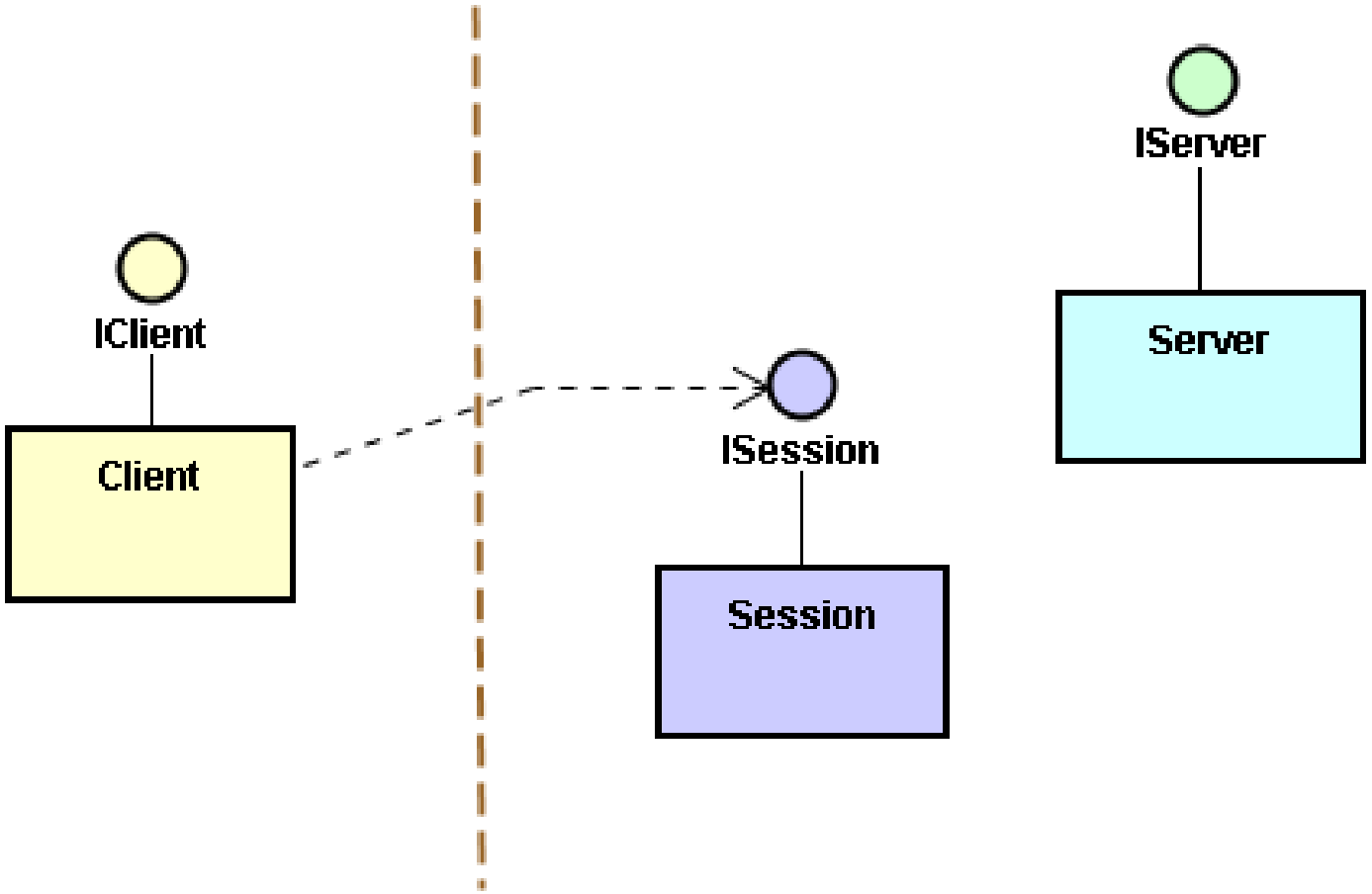
- 基于这个架构，可以建立Client与Server之间的各种连结(Connection)和沟通(Communication)。
- 例如，Client端的浏览器(Browser)会与Server建立连结，然后开起一段交谈(Session)。

- 首先，Client透过某项机制(例如，Android的ServiceManager)来绑定(bind)后台的Server。

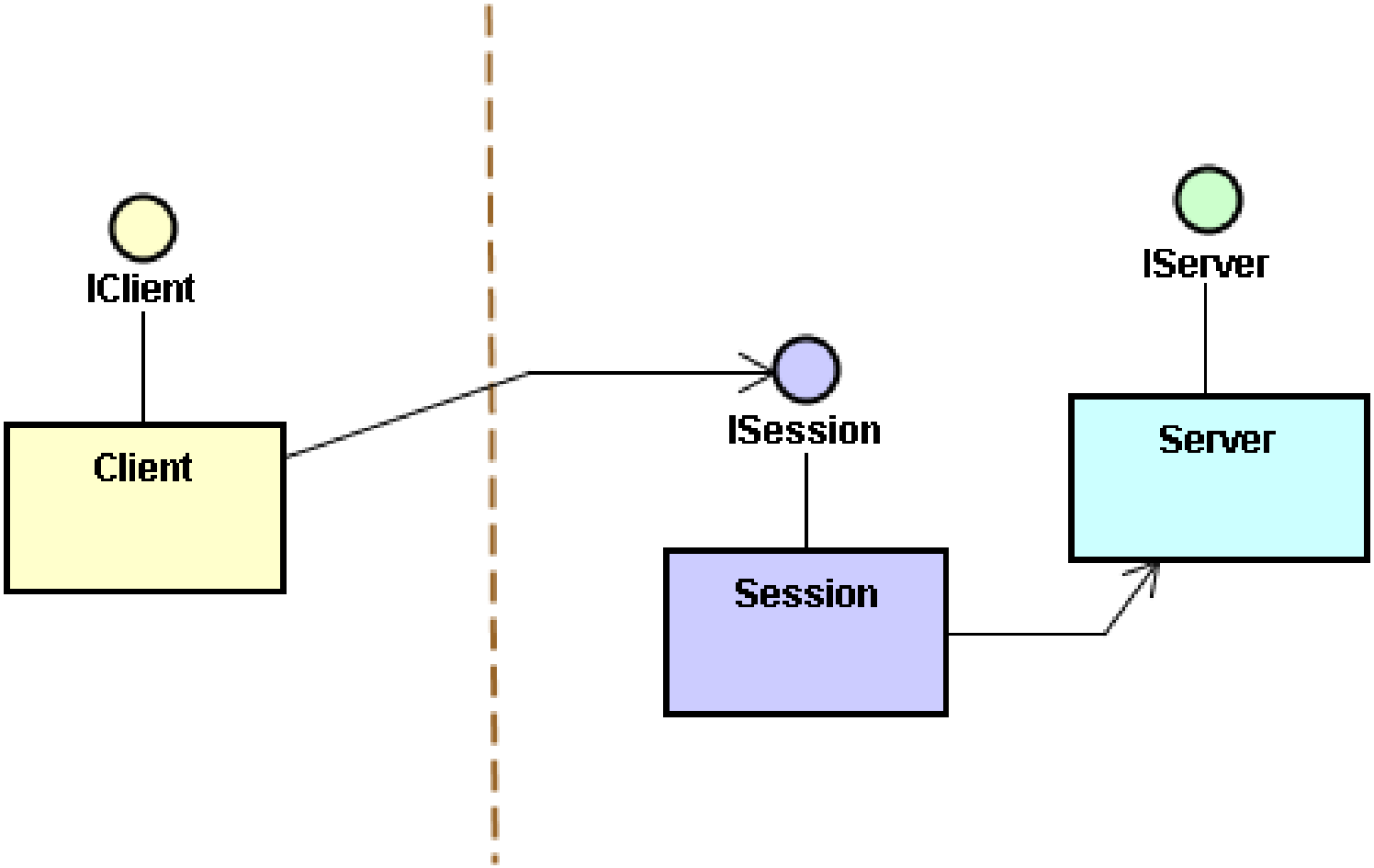
- 绑定(bind)之后，Client就能呼叫Server的getSession()函数，准备开启一段对话。
- 此时，Server就诞生一个Session对象，来作为这项连结的专属对象，可以记载对话过程所产生的信息。


getSession()
IClient
IServer
Client
ISession
Server
Session
<<new>>

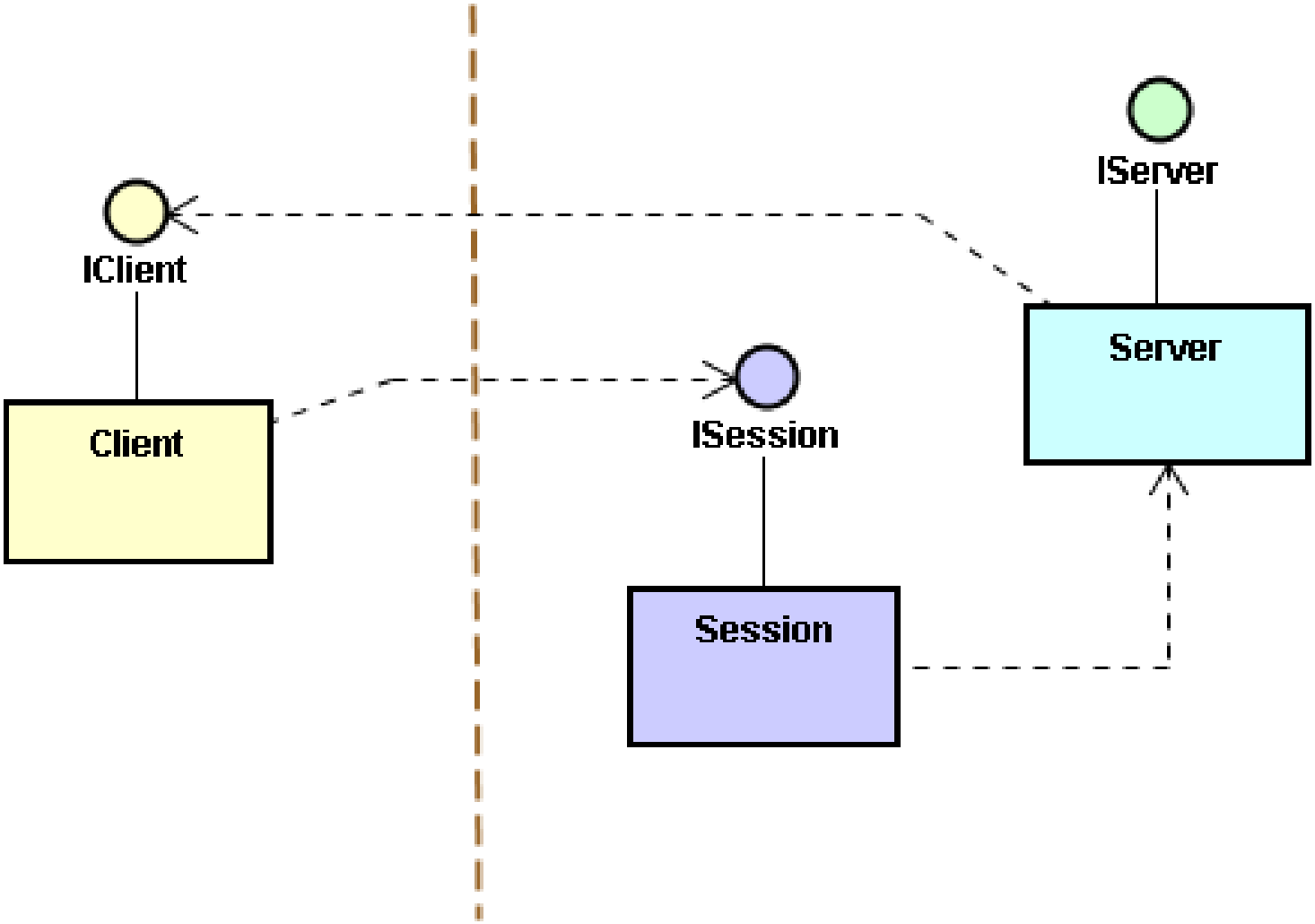
- 把ISession口回传给Client。如下图：



- Client掌握了ISession接口，就能透过ISession接口来呼叫Session的函数，然后由Session 来与Server沟通。如下图：

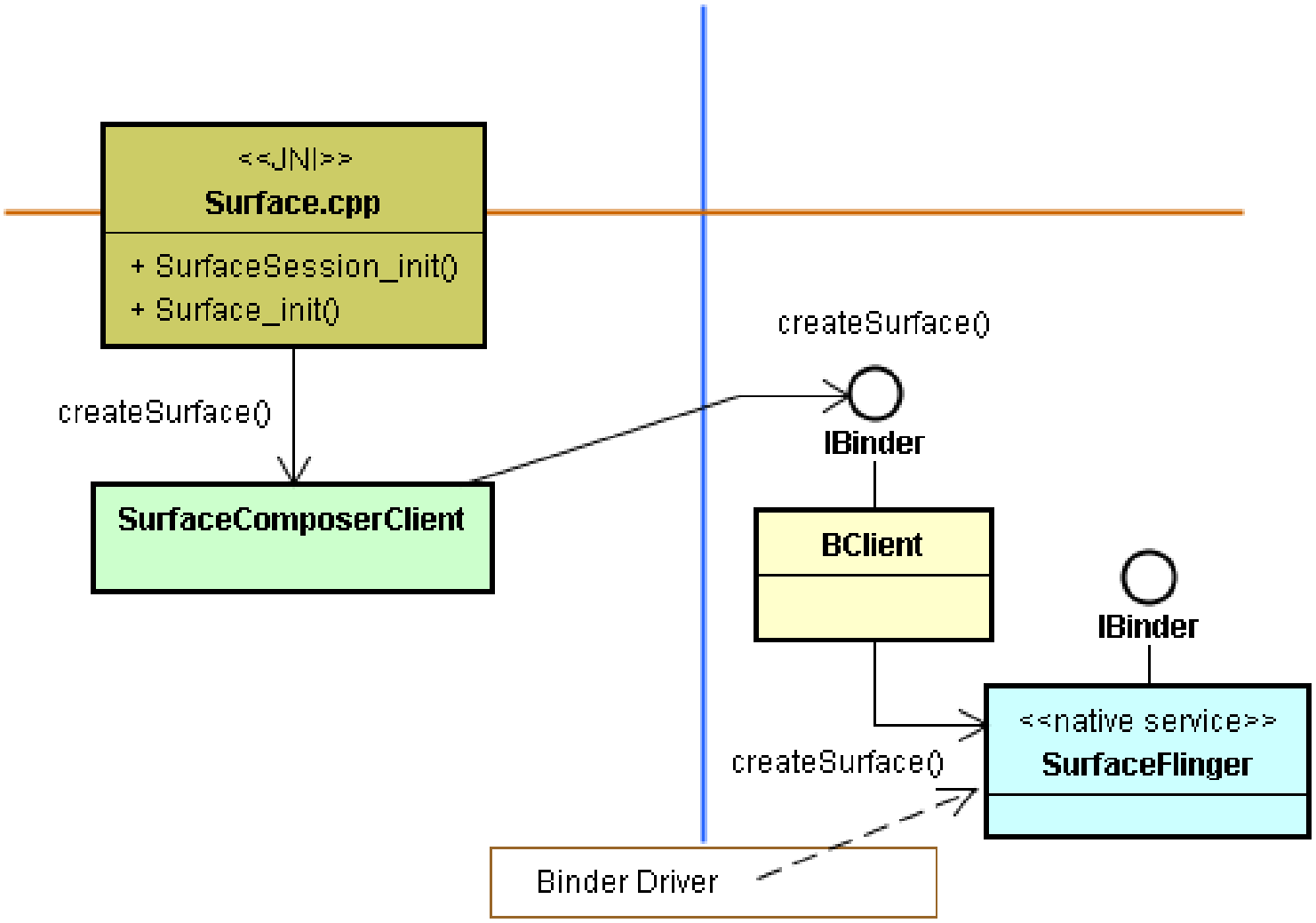

IClient
IServer
Client
ISession
Server
Session

- 在上图的呼叫时，Client可以将自己的IClient接口传递给Session或Server。让Session或Server就能透过IClient接口来呼叫Client的函数。于是建立了双向连结的架构，如下图：


IServer
IClient
Server
Client
ISession
Session

- 由于Client拥有ISession接口，就能透过Session来与Server沟通。
- 同时，Server拥有IClient接口，就能使用IClient接口来与Client沟通。

例如，
Android的SurfaceFlinger系统服务


<<JNI>>
Surface.cpp
+ SurfaceSession_init()
+ Surface_init()
createSurface()
createSurface()
IBinder
SurfaceComposerClient
BClient
IBinder
<<native service>>
SurfaceFlinger
createSurface()
Binder Driver

~ Continued ~