

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>

C05_b

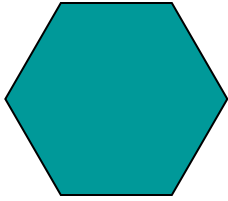
JNI : 多个Java线程 进入本地函数(b)

By 高煥堂

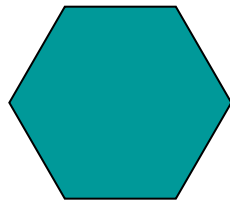
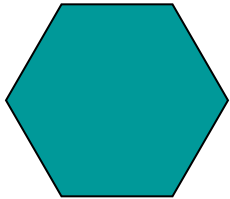
2、从Session概念认识 JNIEnv对象

- 只要你写过WebService应用，你就会有Session(对象)的概念。
- 由于Client 与 Server是 N : 1关系所以Server替每一个Client的Connection准备一个Session对象，让各Connection使用自己专属的对象，避免共享对象的数据安全问题。

Thread t1



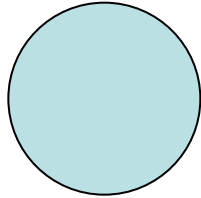
Thread t2



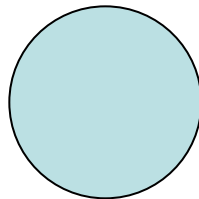
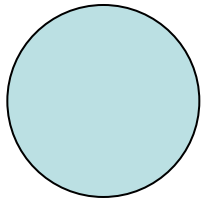
Thread tn



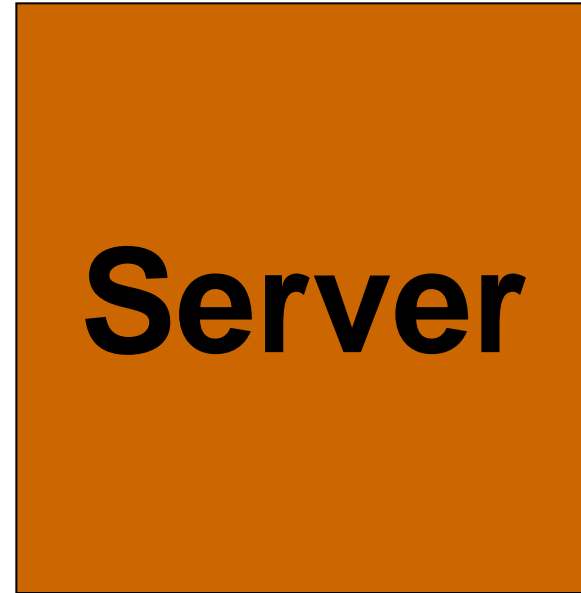
Browser Client



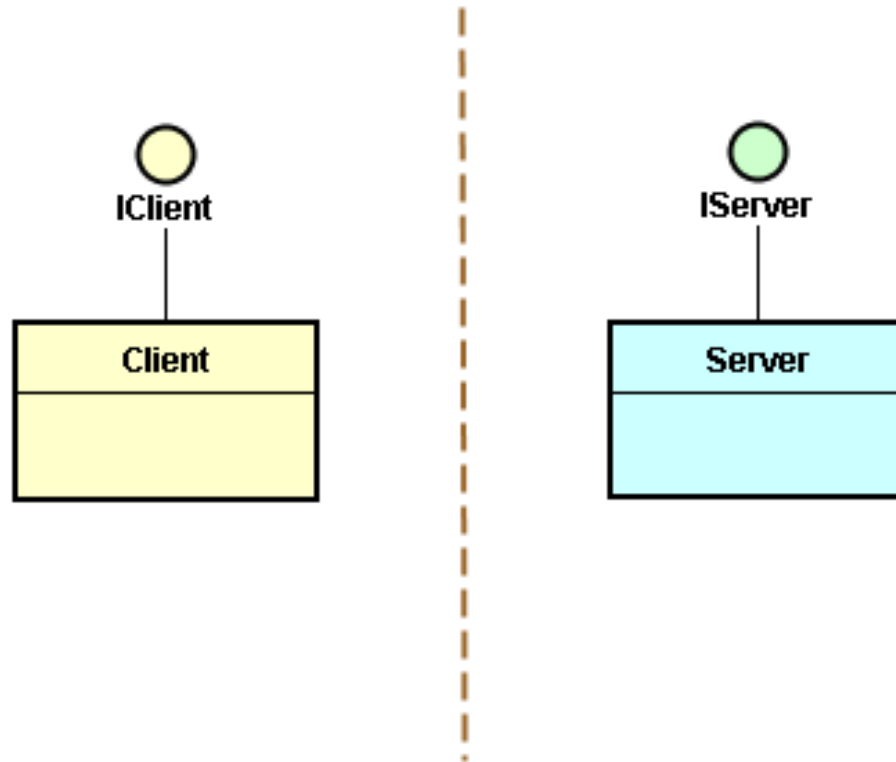
Browser Client



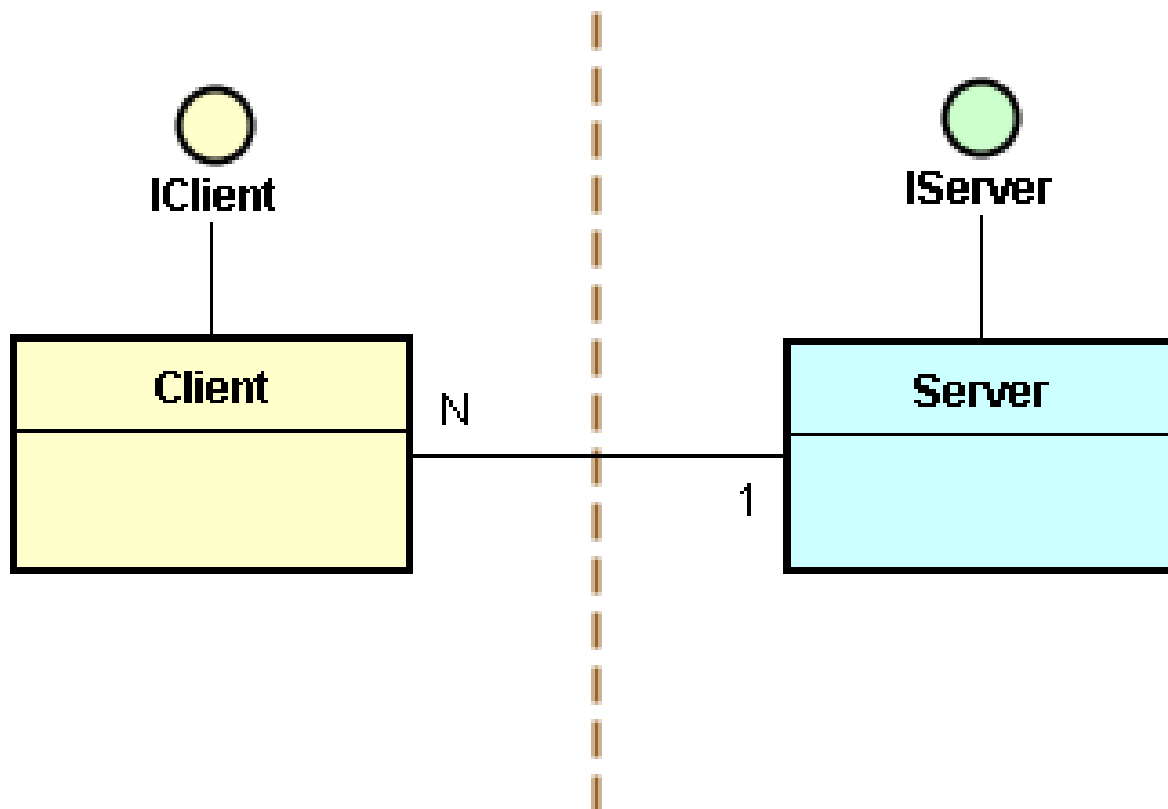
Browser Client



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



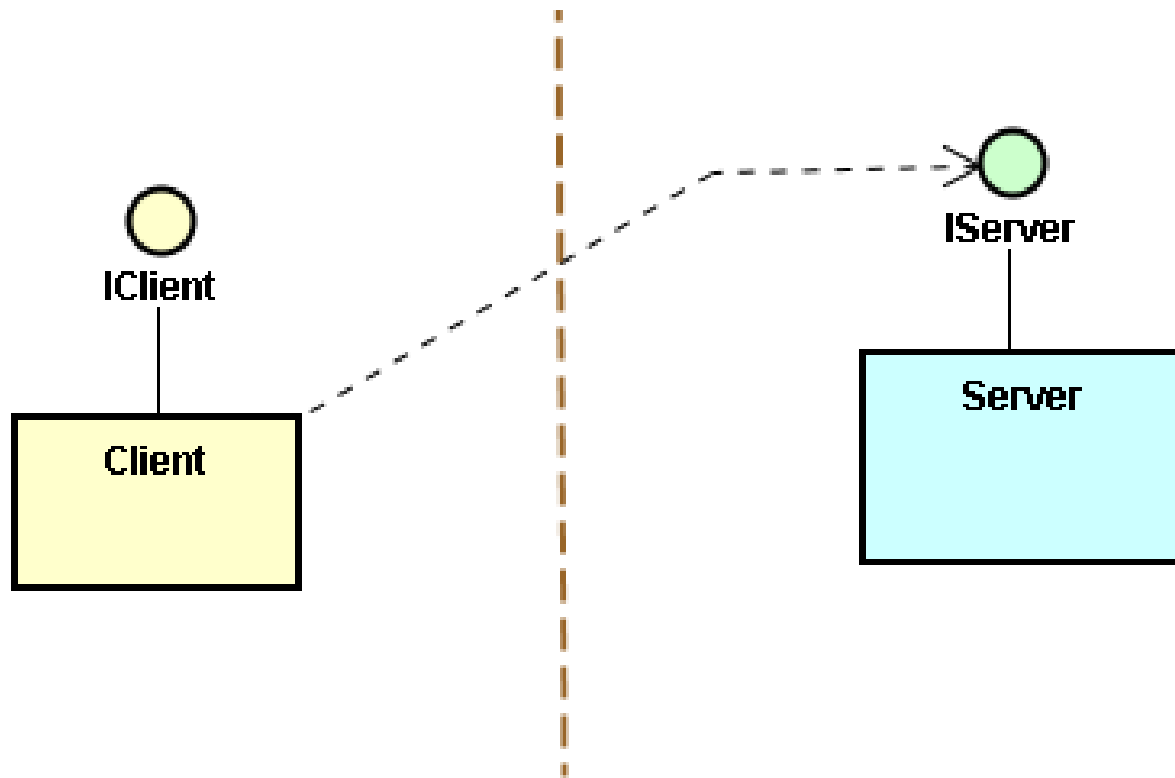
- 在这Client与Server之间透过接口互相沟通；而且多个 Client可同时与Server建立连结，取得Server的服务。所以，Client与Server之间是N:1的关系，如下图：



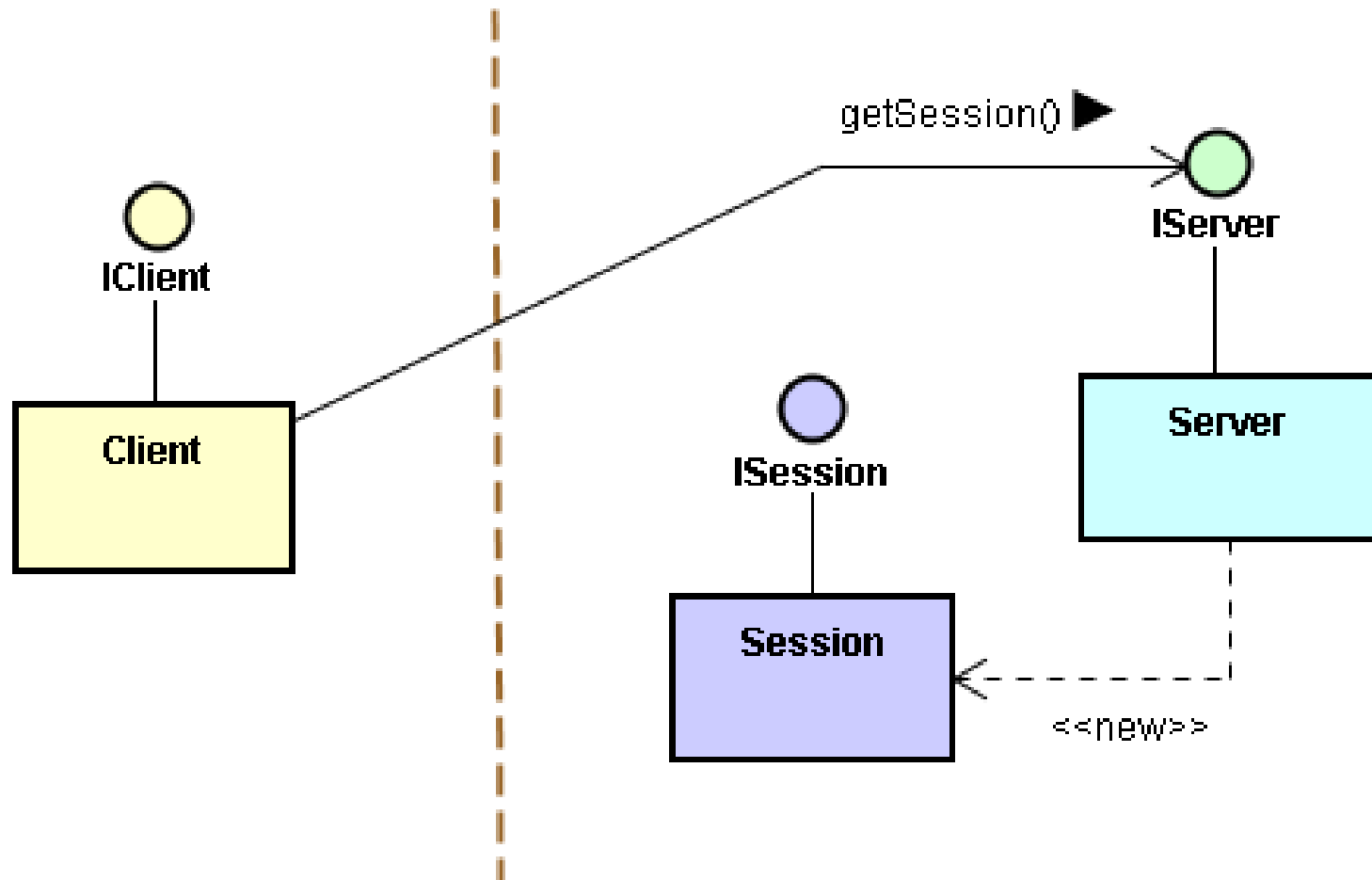
- 基于这个架构，可以建立Client与Server之间的各种连结(Connection)和沟通(Communication)。
- 例如，Client端的浏览器(Browser)会与Server建立连结，然后开起一段交谈(Session)。

- 首先，Client透过某项机制(例如，呼叫公用的getConnection()函数等)来建立与Server之间的连结，此时Server就把它的接口(即IServer)回传给Client，如下图：

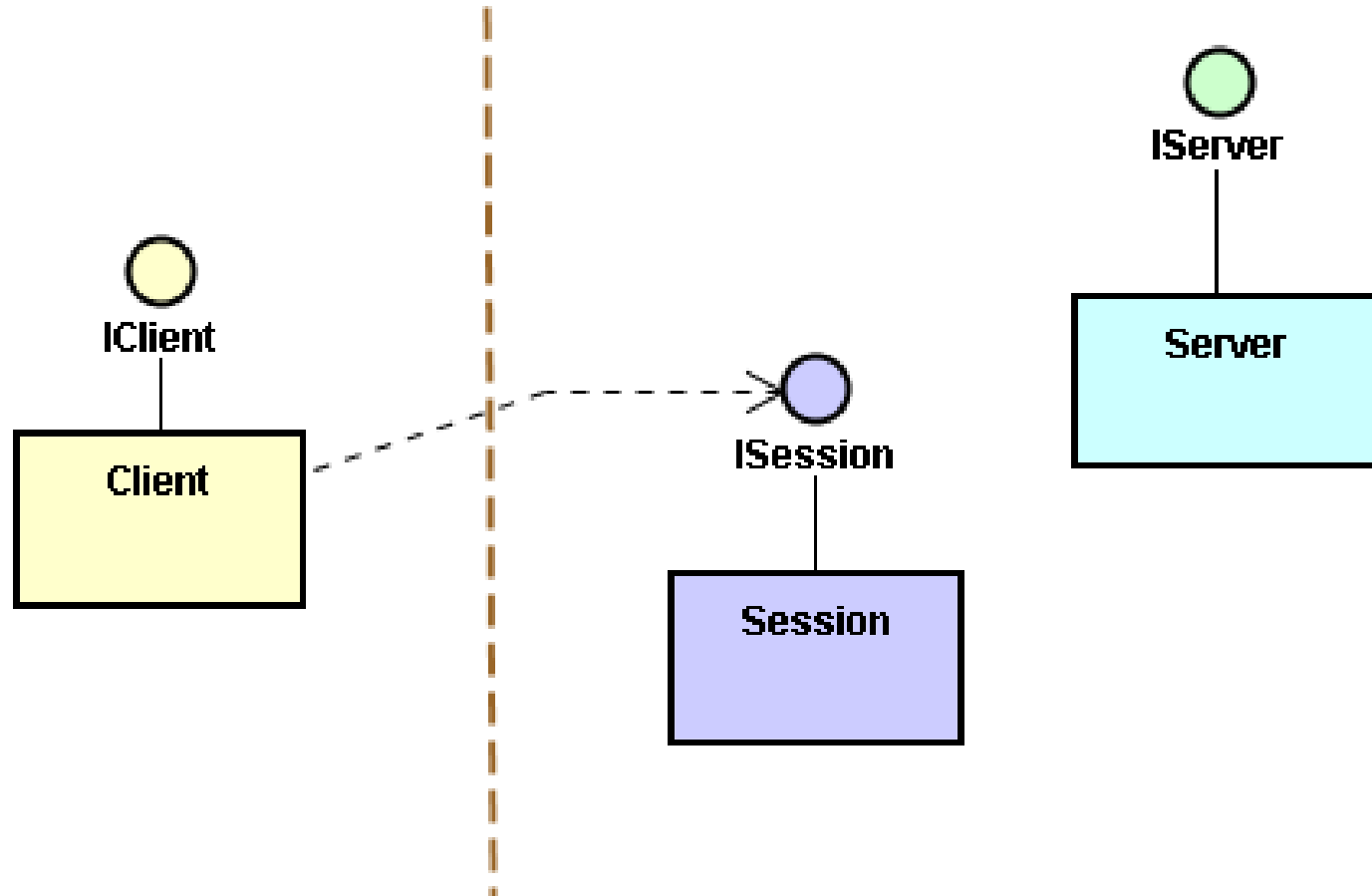
getConnection()



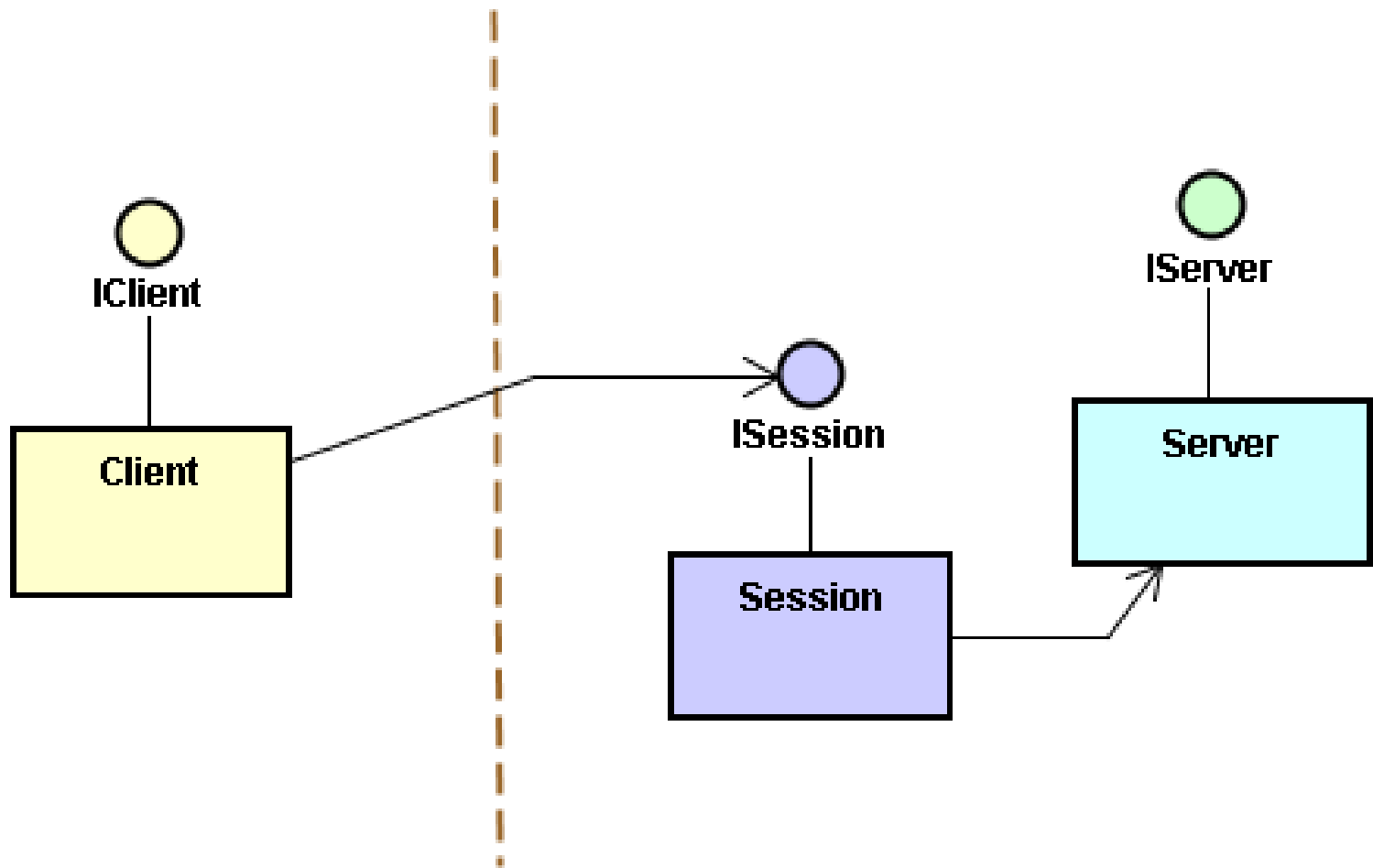
- 透过刚才所建立的连结关系，Client就能呼叫Server的getSession()函数，准备开启一段对话。
- 此时，Server就诞生一个Session对象，来作为这项连结的专属对象，可以记载对话过程所产生的信息。



- 把ISession口回传给Client，让Client直接与Session沟通；才间接与Server沟通。如下图所示：



- 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：



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

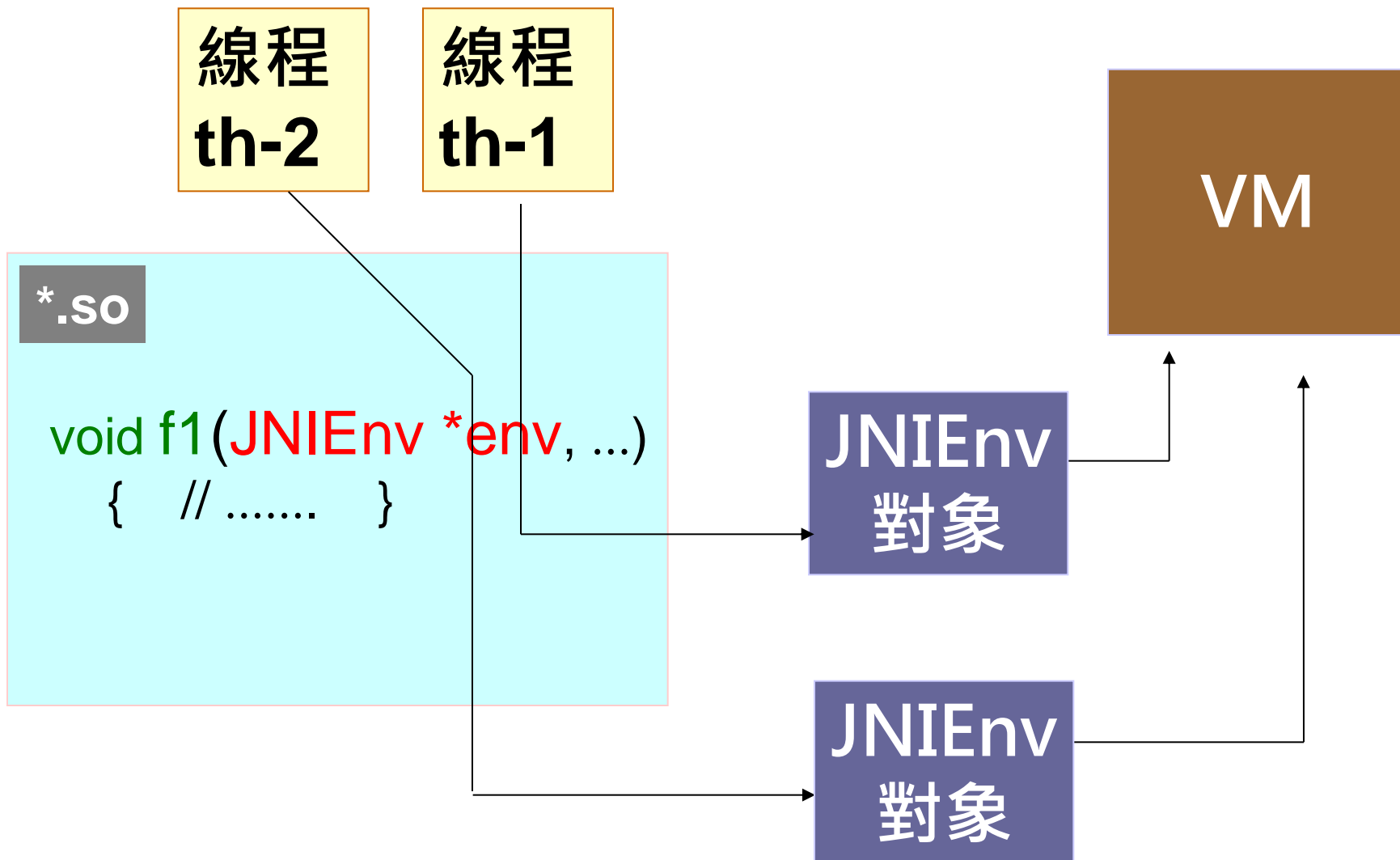
*.so

```
void f1(JNIEnv *env, ...)  
{ // ..... }
```

JNIEnv
對象

JNIEnv
對象

VM





~ Continued ~