

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>

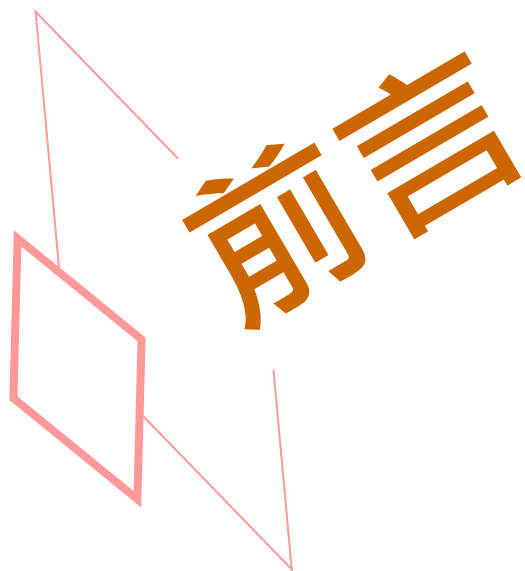
C02_d

认识JNI开发与NDK (d)

By 高煥堂

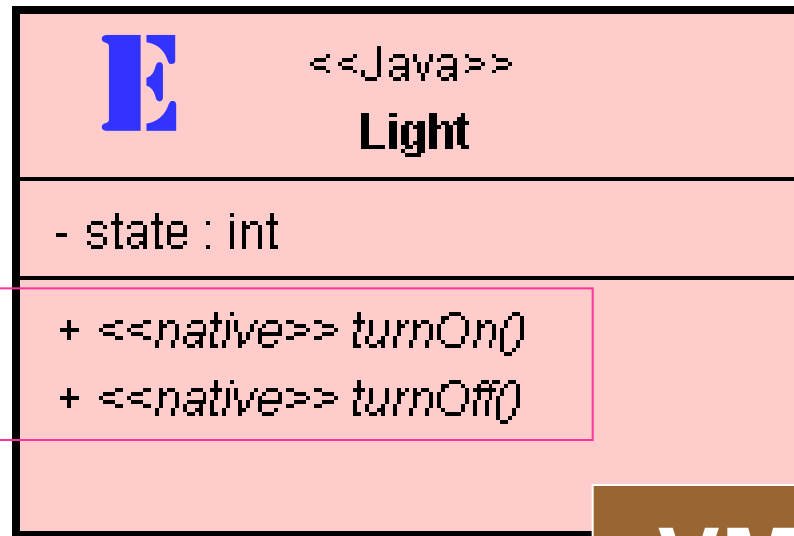
4、*.so的入口函数： JNI_OnLoad()

VM (virtual machine)
的角色



- Java代码在VM上执行。
- 在执行Java代码的过程中，如果Java需要与本地代码(*.so)沟通时，VM就会把*.so视为插件<Tn>而加载到VM里。
- 然后让Java函数呼叫到这插件<Tn>里的C函数。

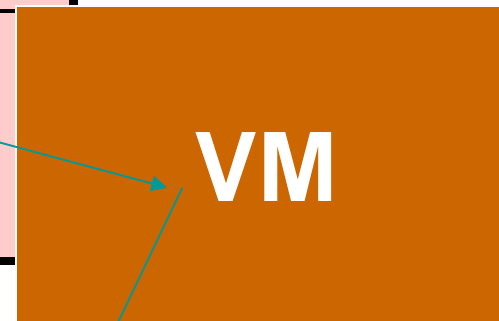
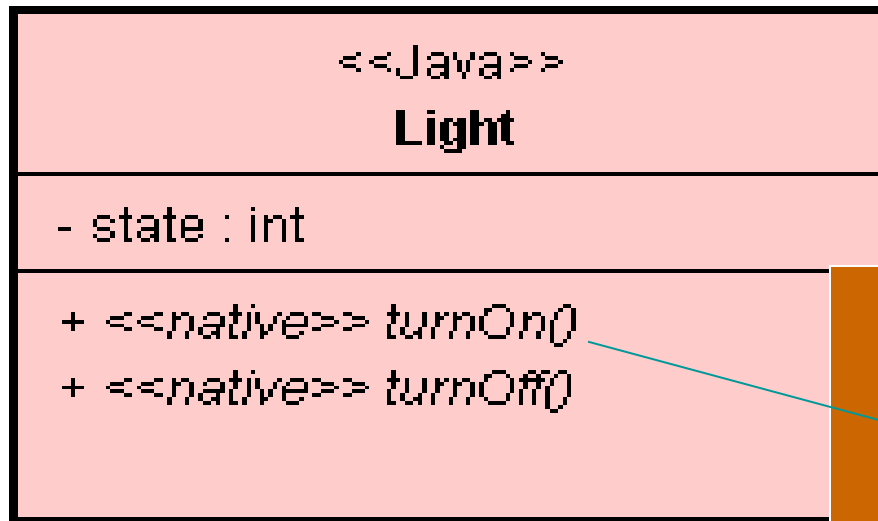
I



```
static void turnOn( object* this ){  
    // this->state = 1;  
    // printf("ON");  
}  
static void turnOff( object* this ) {  
    // this->state = 0;  
    // printf("OFF"); }
```

T

- 插件是由VM来管理的，实体上VM是*.so 插件的管理器(Plug-in Manager)。
- Java与C函数的调用，也是透过VM来对接的。



```
static void turnOn( object* this ){
    // this->state = 1;
    // printf("ON");
}
static void turnOff( object* this ) {
    // this->state = 0;
    // printf("OFF"); }
```


两个观点(视角)下的 EIT造形



実作

JNI接口層

```
// C函数
int add( int x, int y ) {
    return x+y;
}
```

E&I

<<Java層>>

addActivity

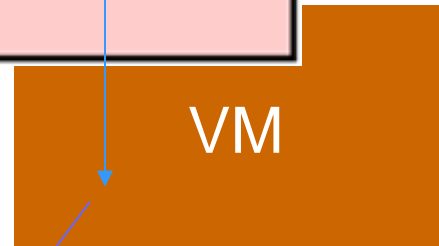
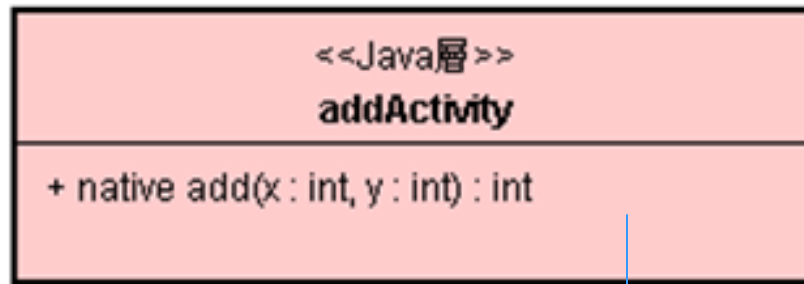
+ native add(x : int, y : int) : int

實作

JNI接口層

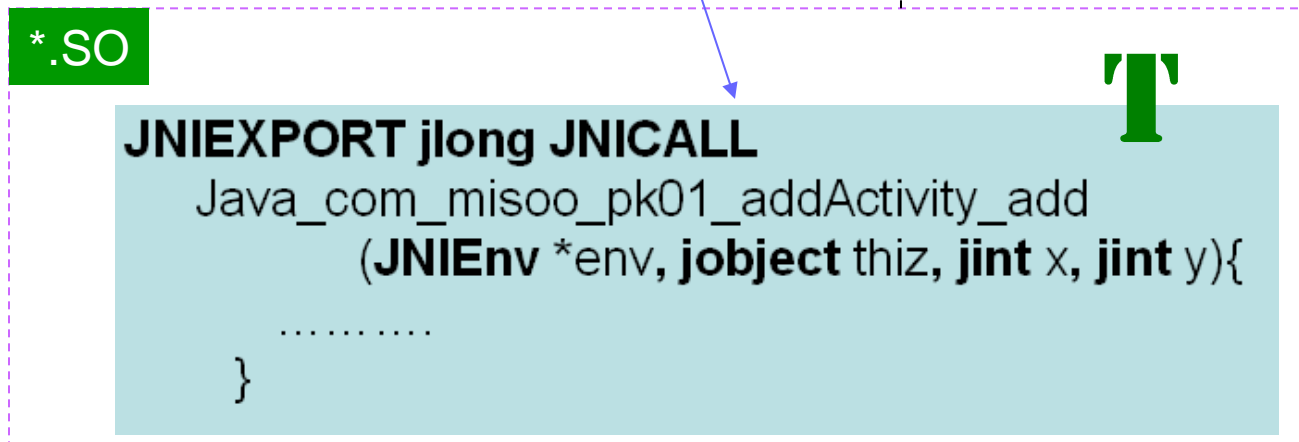
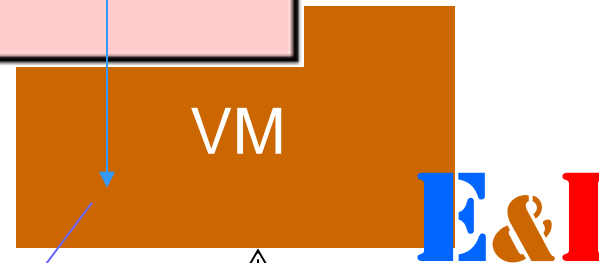
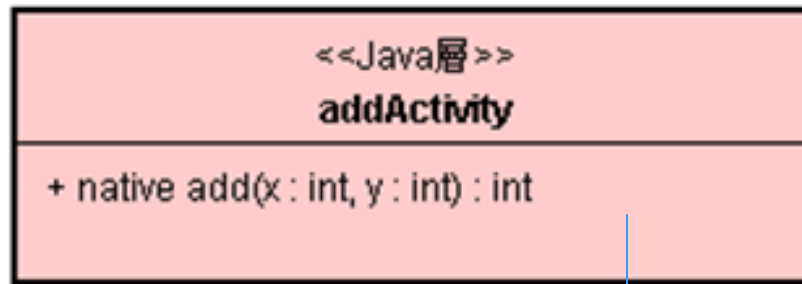
```
// C函數
int add( int x, int y ) {
    return x+y;
}
```

T



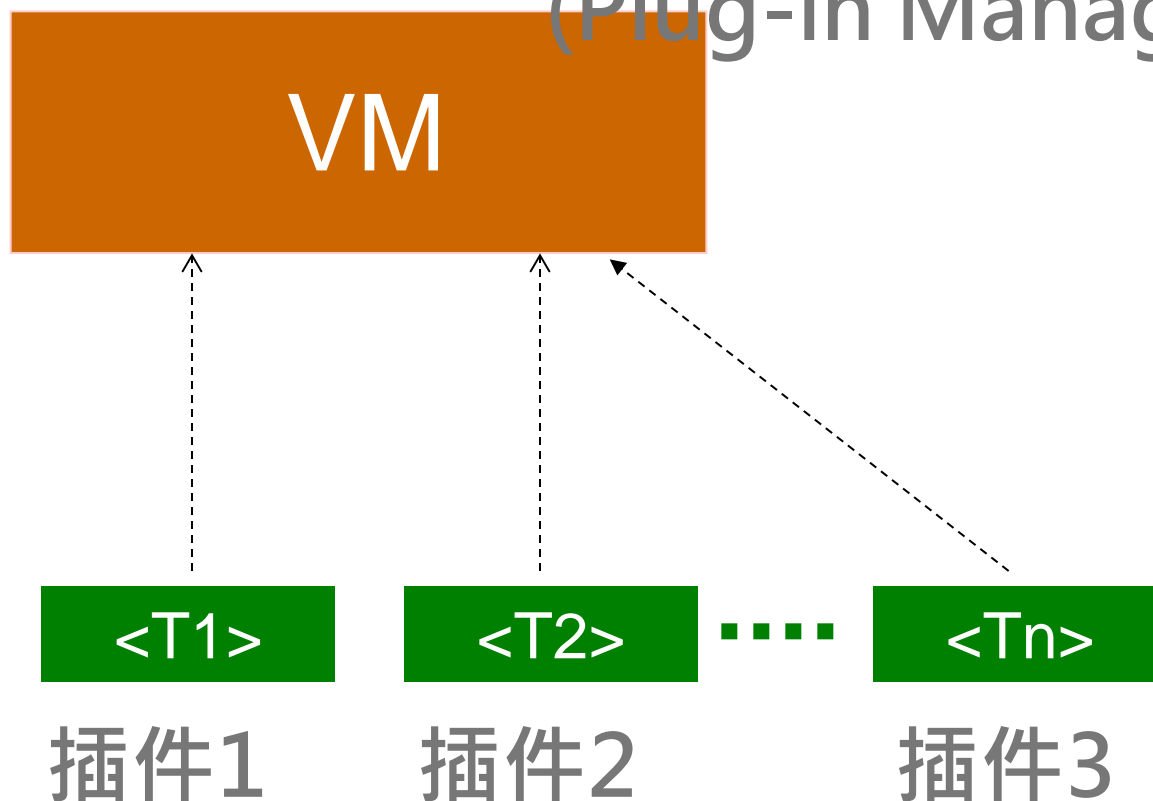
*.SO

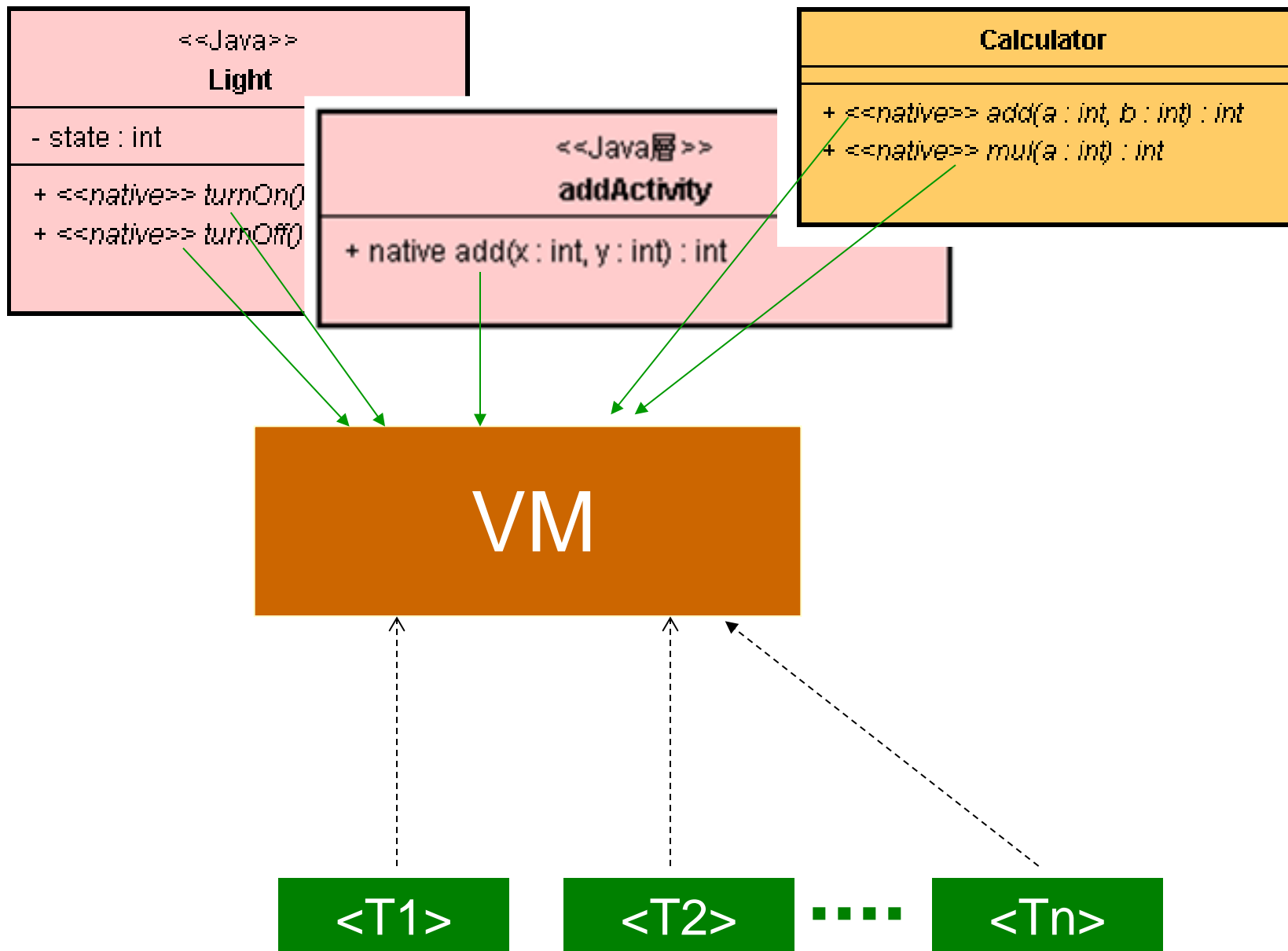
```
JNIEXPORT jlong JNICALL
Java_com_misoo_pk01_addActivity_add
(JNIEnv *env, jobject thiz, jint x, jint y){
    .....
}
```

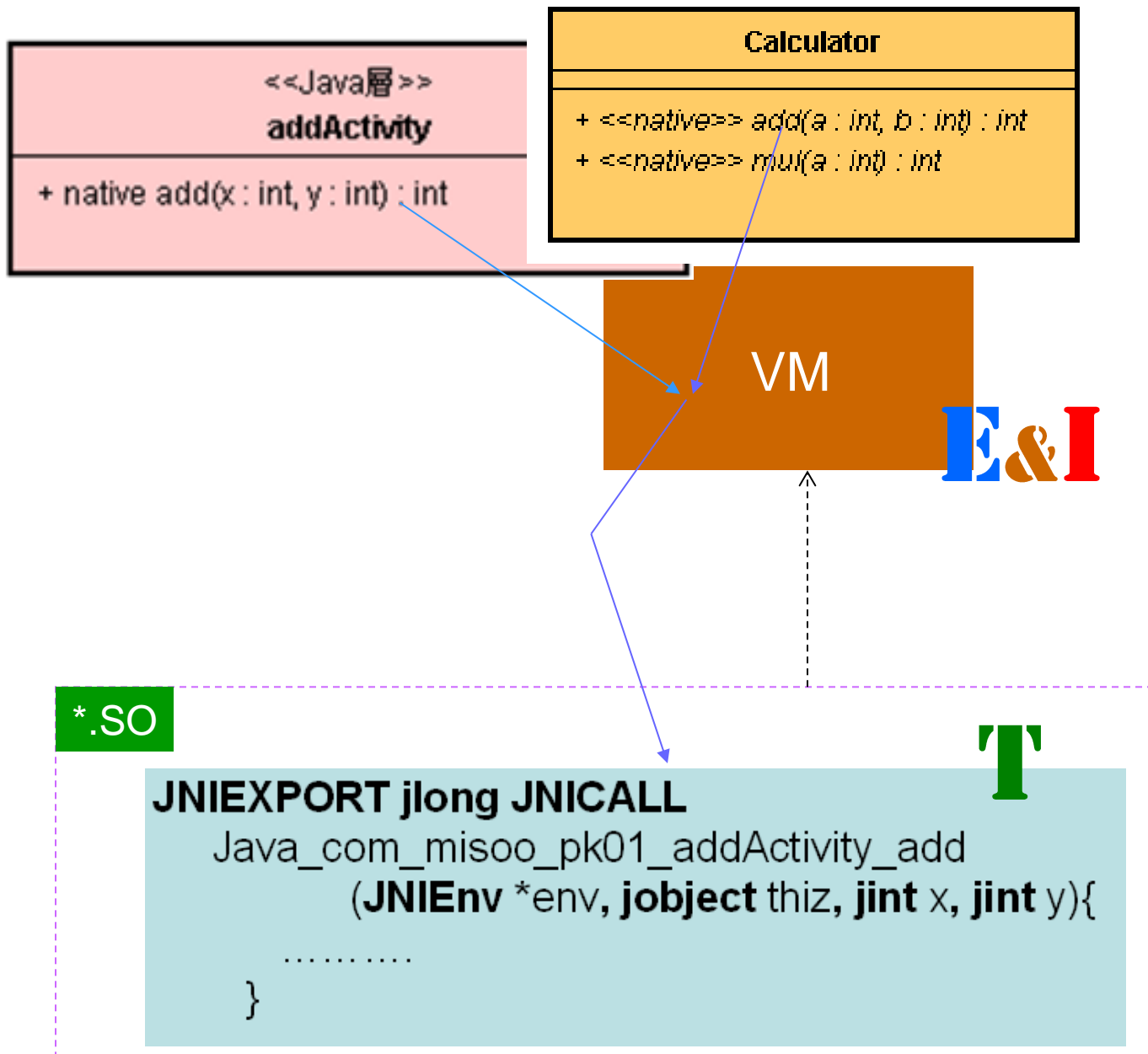


VM是插件管理器的角色

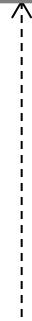
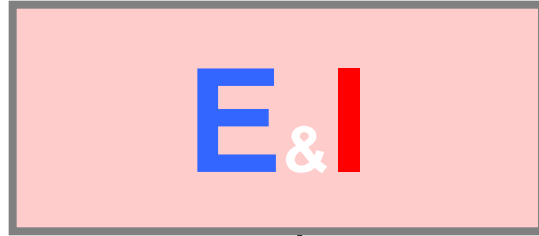
插件管理器 (Plug-in Manager)

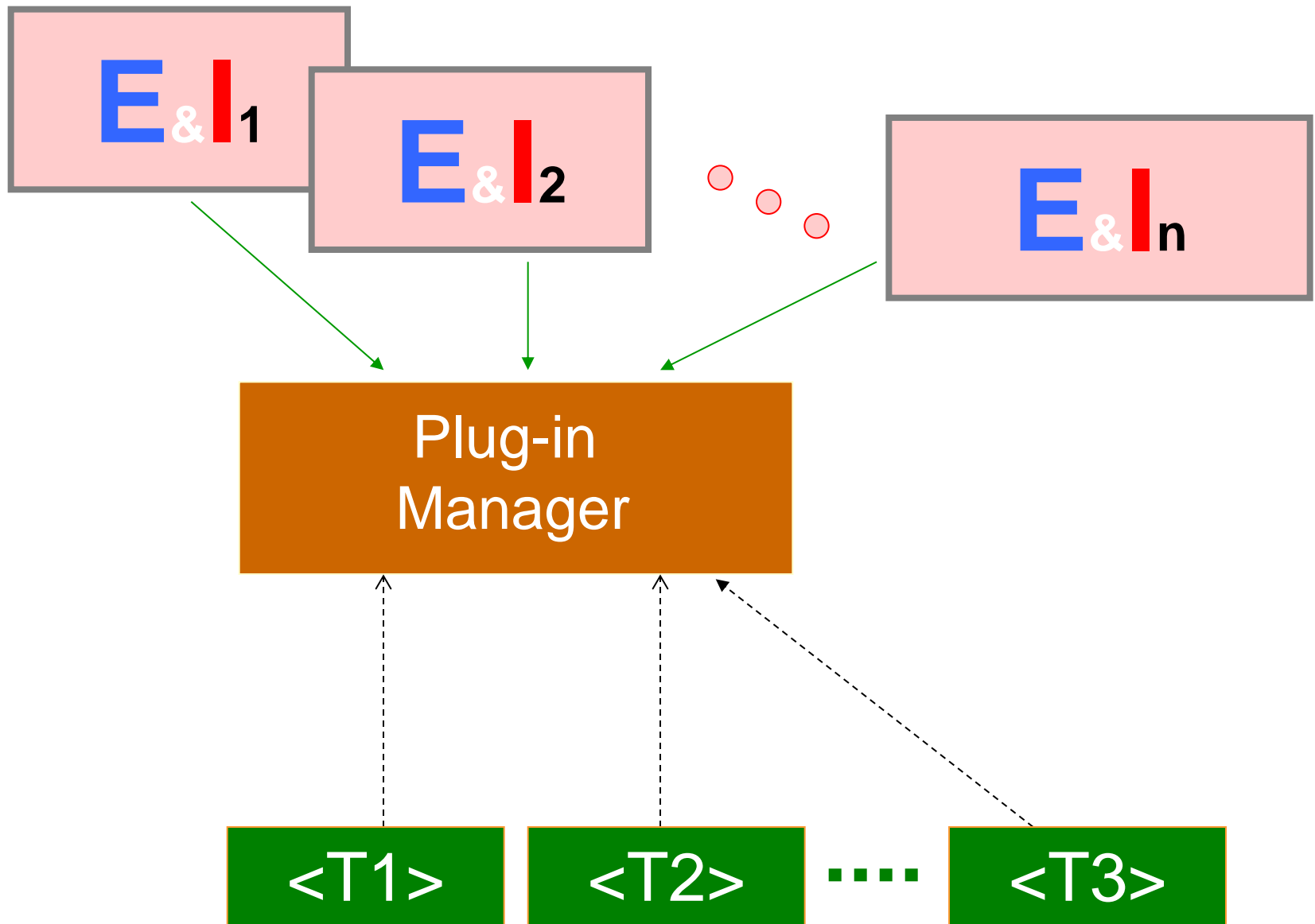






EIT造形平台化的基本思维







~ Continued ~