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

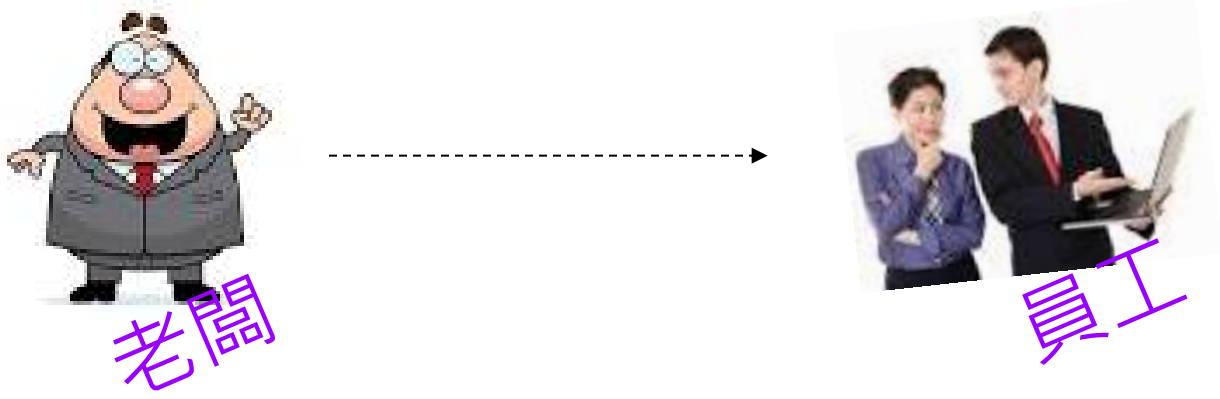
C03_b

JNI：从C调用Java函数 (b)

By 高煥堂

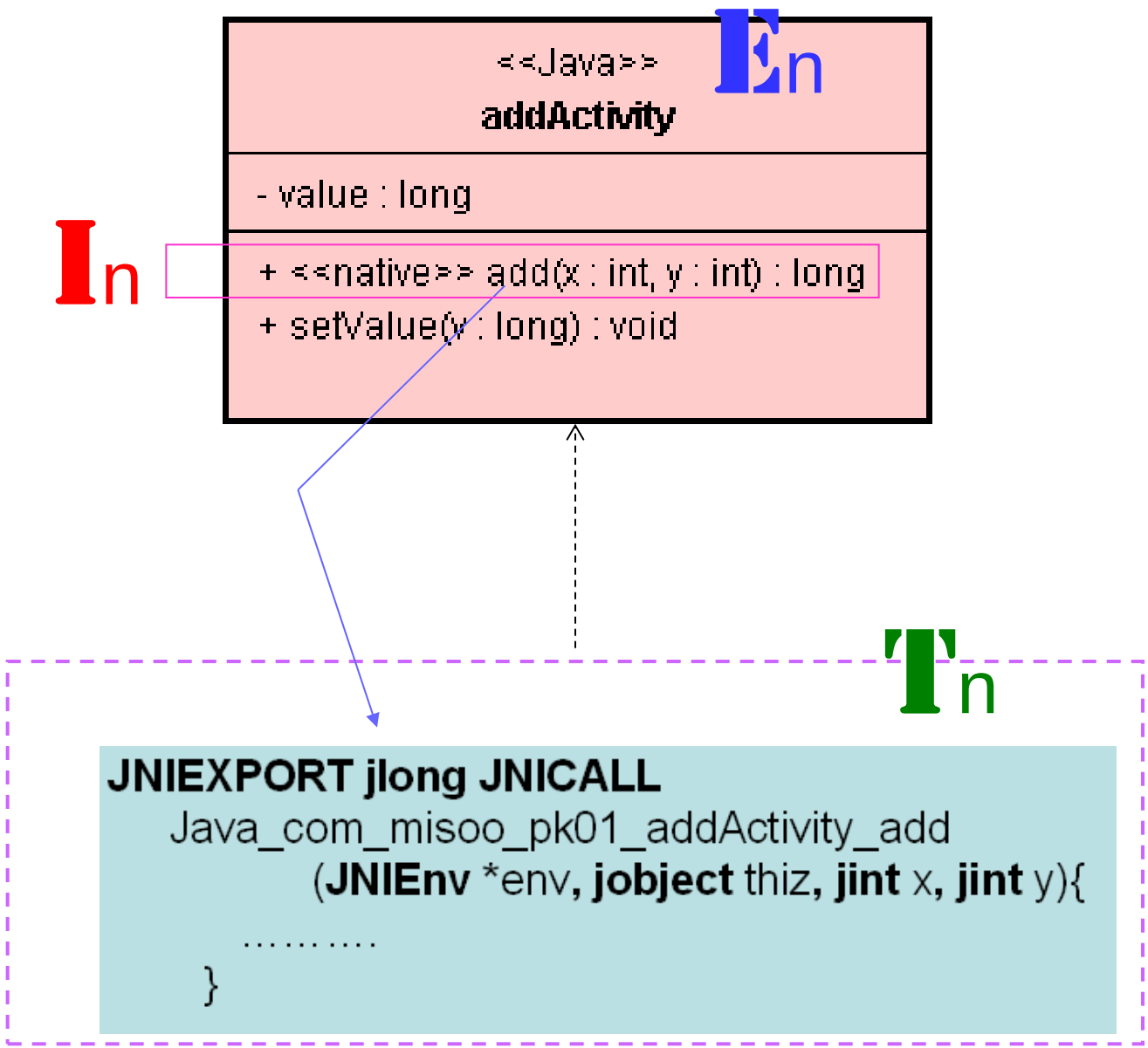
2、控制点与函数调用

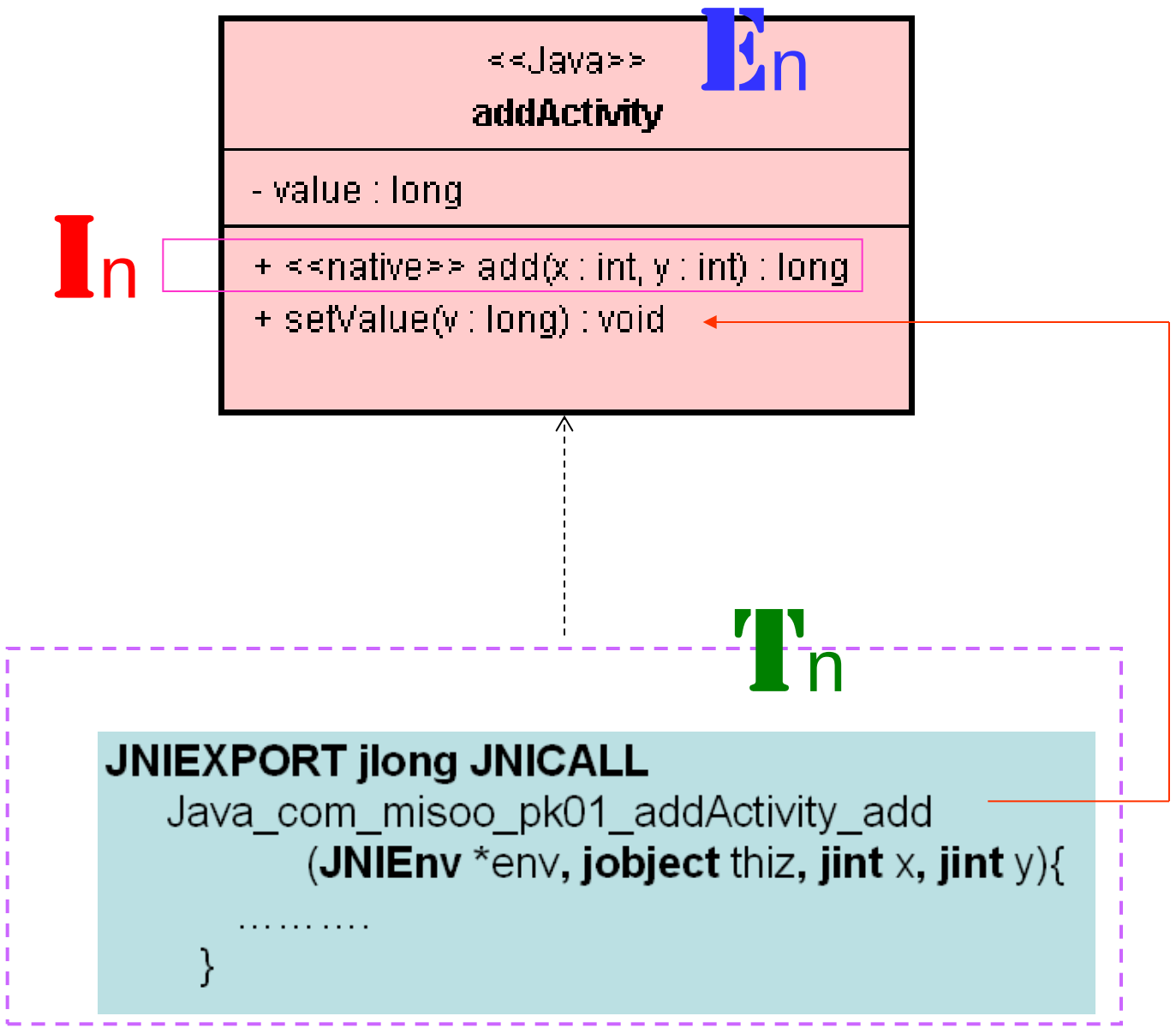
- C调用Java函数，并不一定表示C层拥有控制点。
- 但是，C层拥有控制点的必备表现是：C调用Java层函数



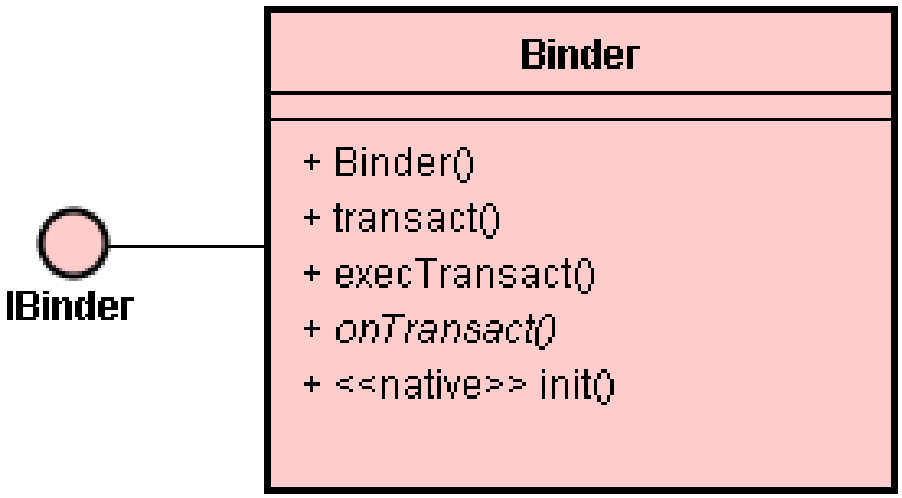
前言

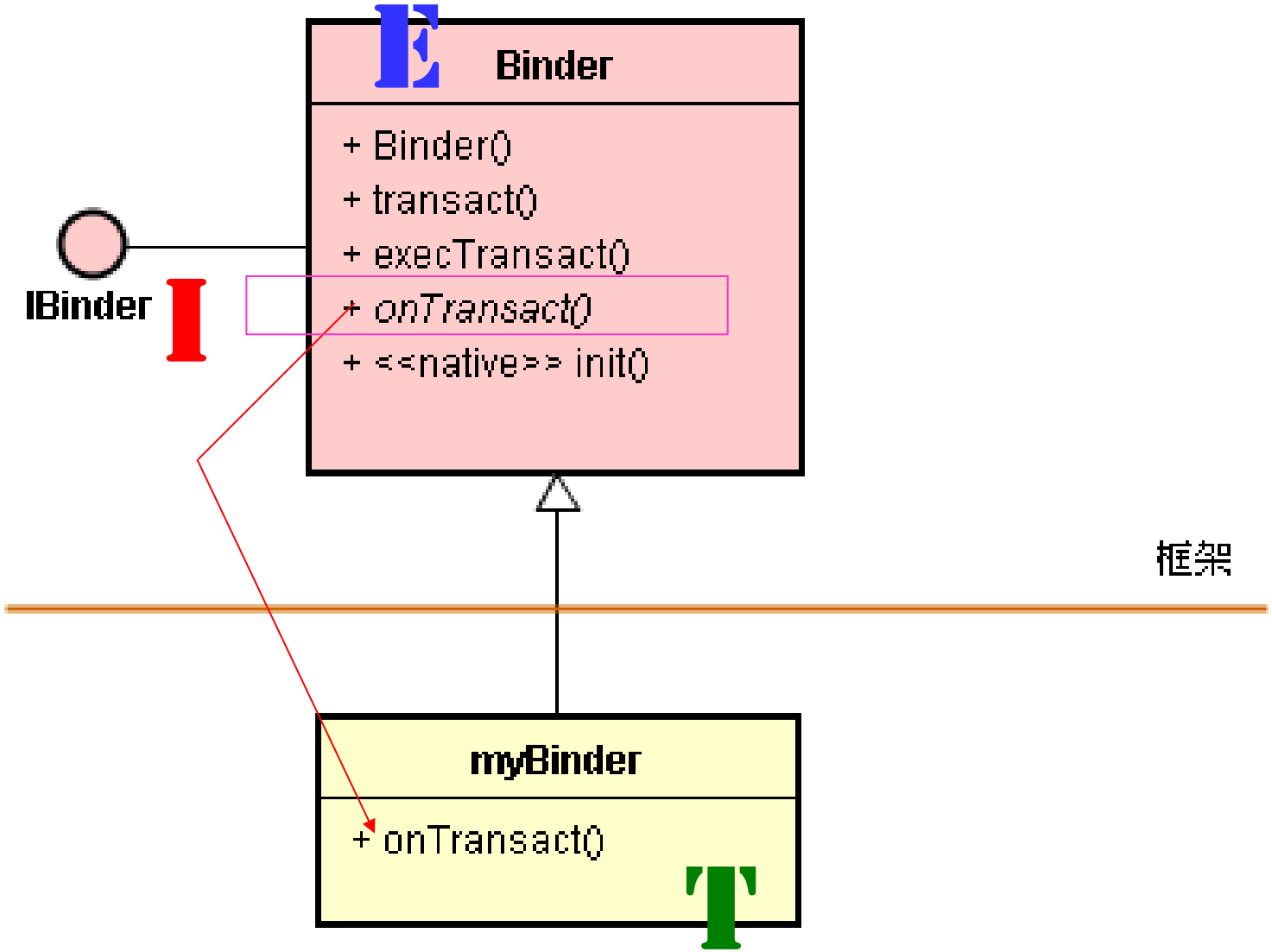
EIT造形的典型
反向调用：IoC

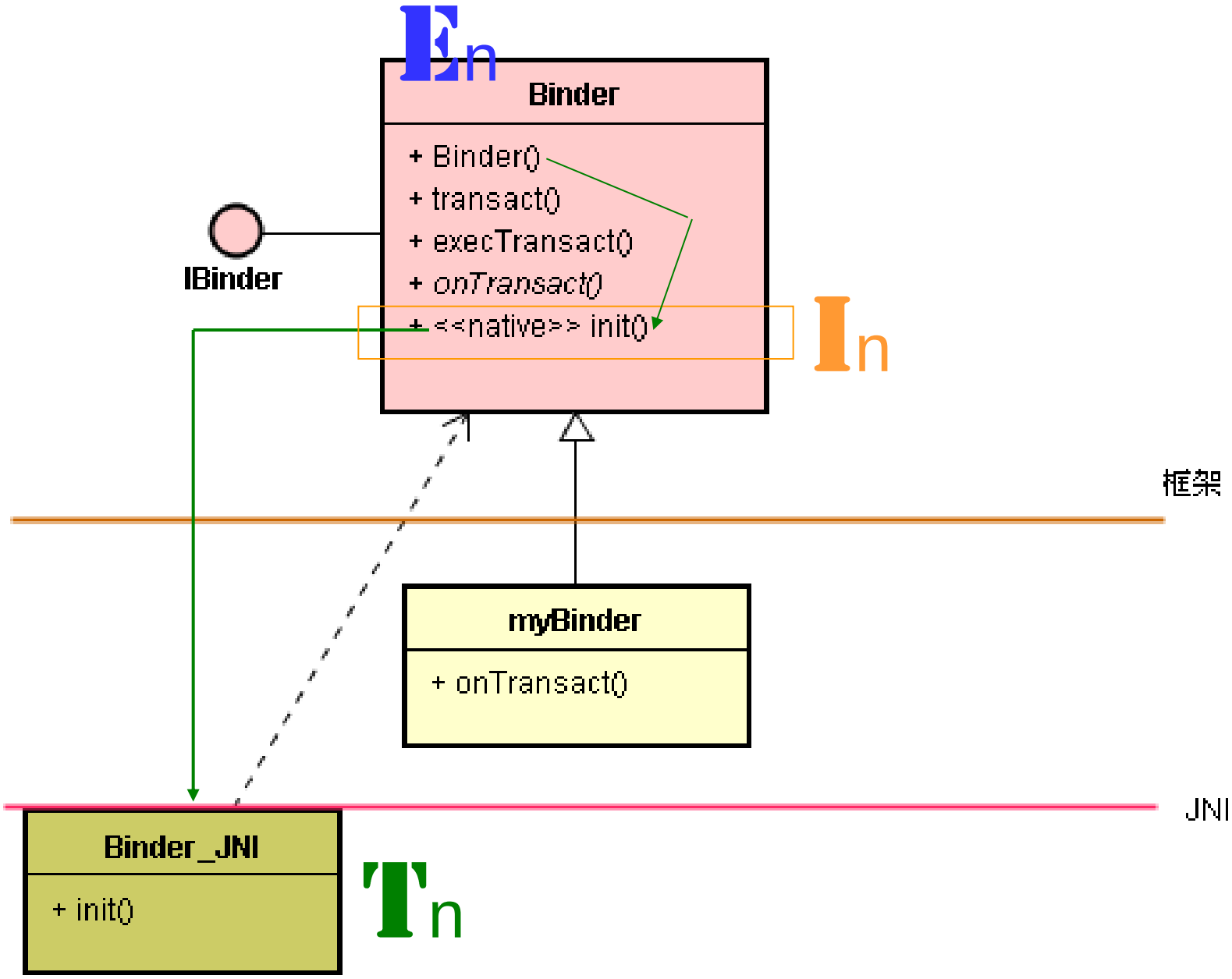

En
<<Java>>
addActivity
- value : long
In
+ <<native>> add(x : int, y : int) : long
+ setValue(v : long) : void
Tn
JNIEXPORT jlong JNICALL
Java_com_misoo_pk01_addActivity_add
(JNIEnv *env, jobject thiz, jint x, jint y){
.....
}

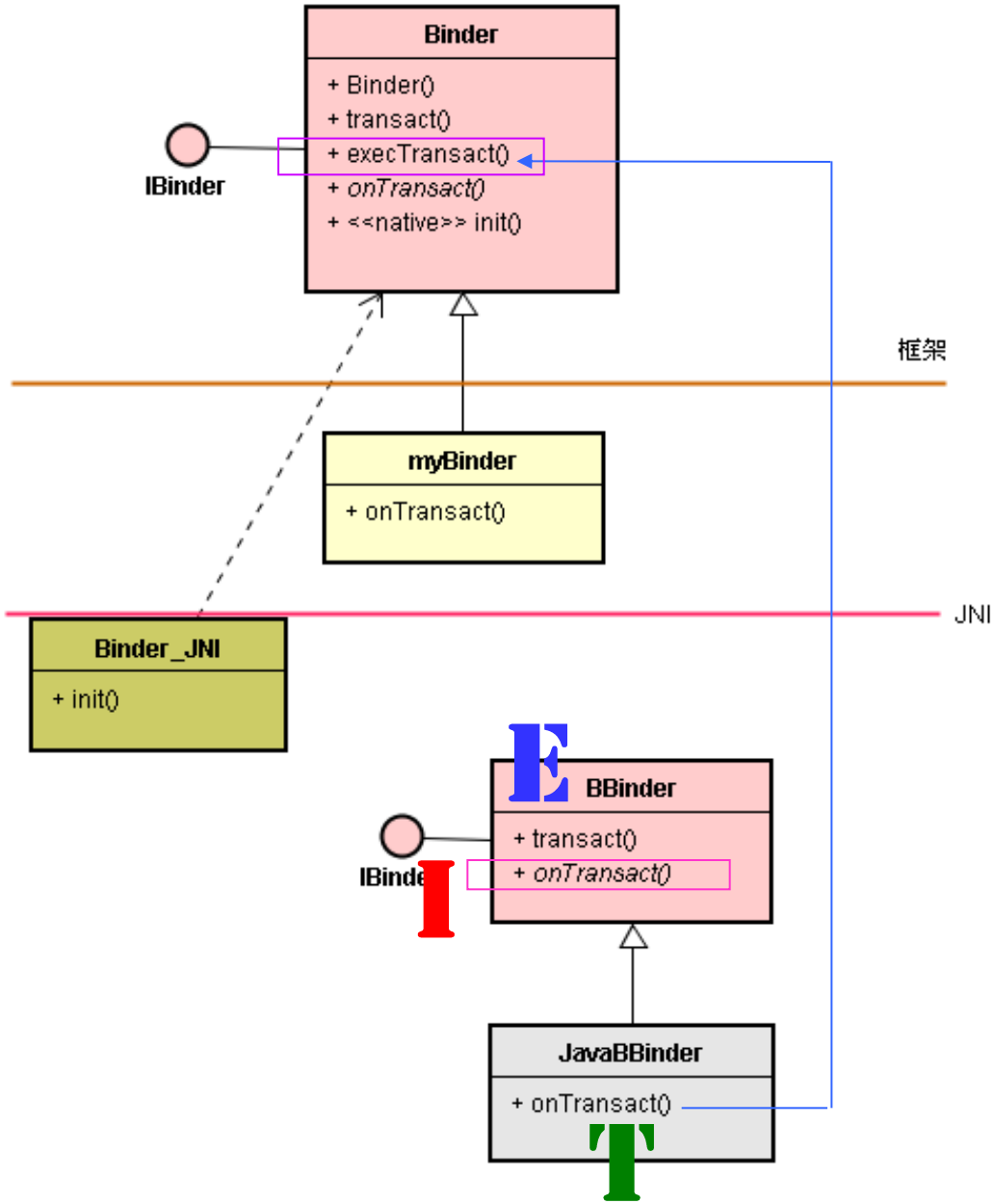

En
<<Java>>
addActivity
- value : long
In
+ <<native>> add(x : int, y : int) : long
+ setValue(v : long) : void
Tn
JNIEXPORT jlong JNICALL
Java_com_misoo_pk01_addActivity_add
(JNIEnv *env, jobject thiz, jint x, jint y){
.....
}

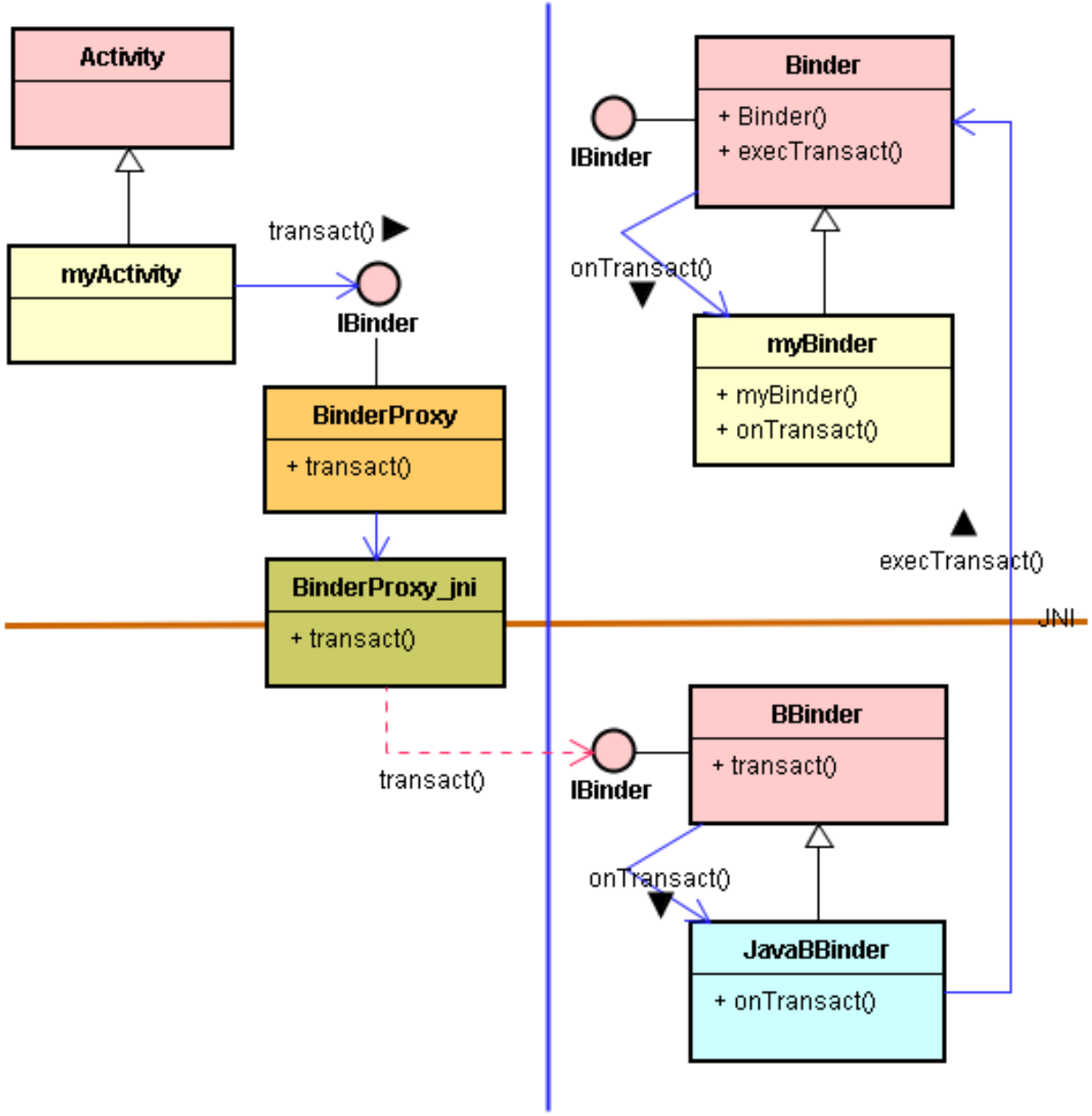
看看Android的范例


Binder
+ Binder()
+ transact()
+ execTransact()
+ onTransact()
+ <<native>> init()
IBinder


Binder
+ Binder()
+ transact()
+ execTransact()
+ onTransact()
+ <<native>> init()
IBinder
框架
myBinder
+ onTransact()

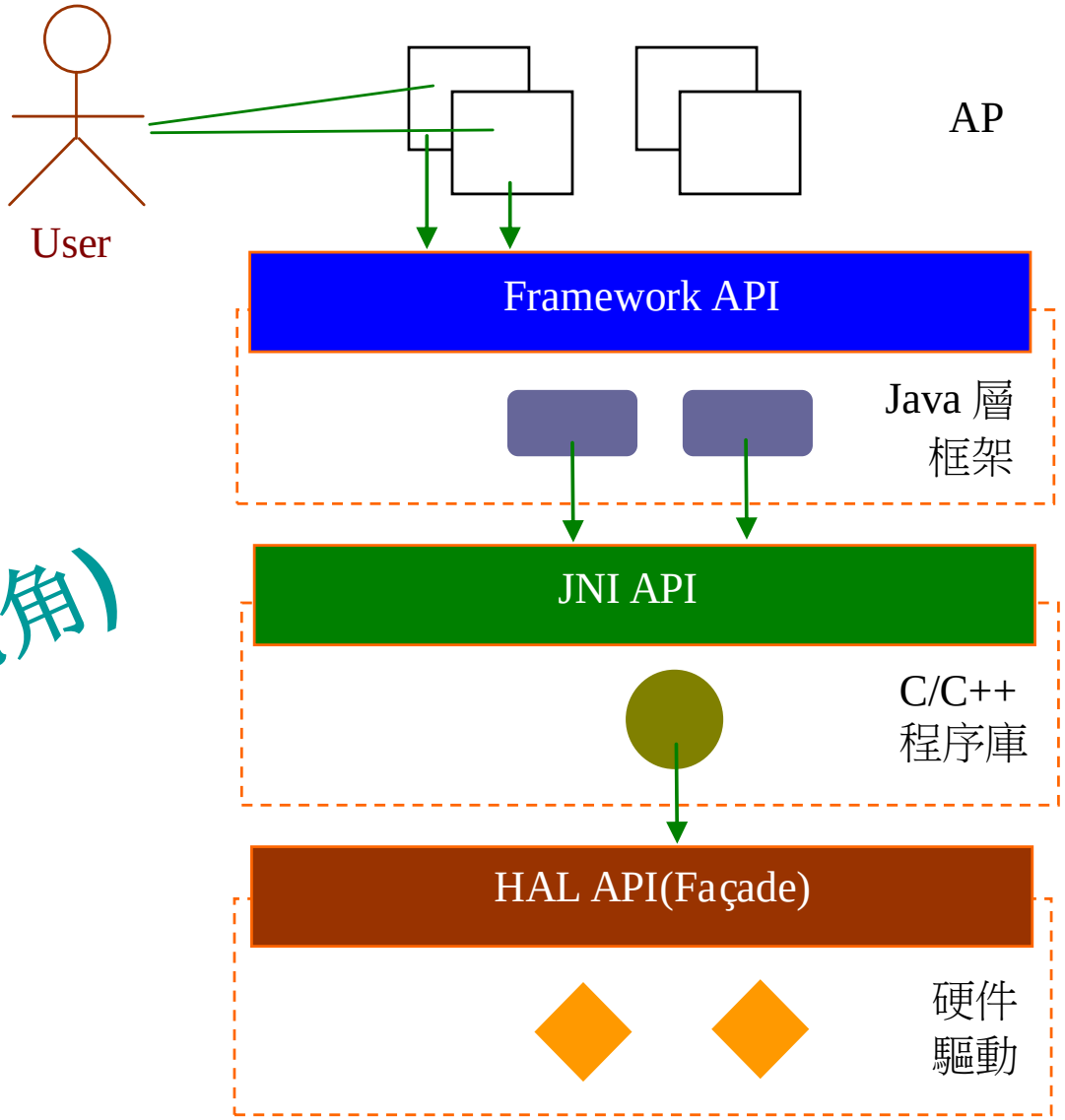

En
Binder
+ Binder()
+ transact()
+ execTransact()
+ onTransact()
+ <<native>> init()
IBinder
In
框架
myBinder
+ onTransact()
JNI
Binder_JNI
+ init()
Tn


Binder
+ Binder()
+ transact()
+ execTransact()
+ onTransact()
+ <<native>> init()
IBinder
框架
myBinder
+ onTransact()
JNI
Binder_JNI
+ init()
BBinder
+ transact()
+ onTransact()
IBinde
JavaBBinder
+ onTransact()


Activity
myActivity
transact()
IBinder
BinderProxy
+ transact()
BinderProxy_jni
+ transact()
transact()
Binder
+ Binder()
+ execTransact()
IBinder
onTransact()
myBinder
+ myBinder()
+ onTransact()
execTransact()
JNI
BBinder
+ transact()
IBinder
onTransact()
JavaBBinder
+ onTransact()


Java
APPLICATIONS
Alarm
Dialer
SMS/MMS
IM
Browser
Camera
Alarm
Home
Contacts
Voice Dial
Email
Calendar
Media Player
Albums
Clock
...
APPLICATION FRAMEWORK
Activity Manager
Window Manager
Content Provider
View System
Notification Manager
Package Manager
Telephony Manager
Resource Manager
Location Manager
...
C/C++
LIBRARIES
Surface Manager
Media Framework
SQLite
OpenGL|ES
FreeType
WebKit
SGL
SSL
Libc
ANDROID RUNTIME
Core Libraries
Dalvik Virtual Machine
HARDWARE ABSTRACTION LAYER
Graphics
Audio
Camera
Bluetooth
GPS
Radio(RIL)
WiFi
...
Kernel
LINUX KERNEL
Display Driver
Camera Driver
Bluetooth Driver
Shared Memory Driver
Binder (IPC) Driver
USB Driver
Keypad Driver
WiFi Driver
Audio Driver
Power Management

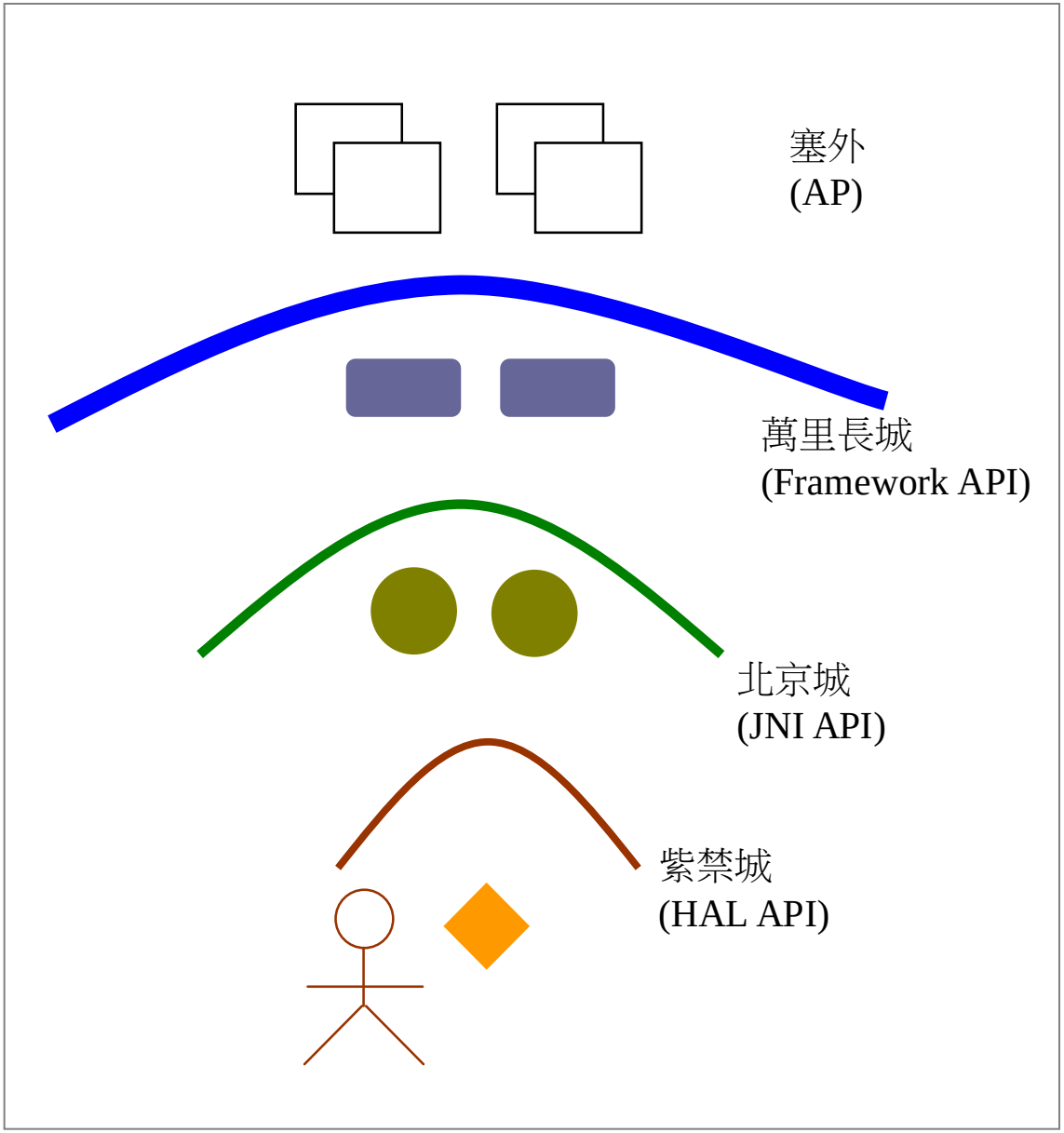
古典观点(视角)


User
AP
Framework API
Java 層
框架
JNI API
C/C++
程序庫
HAL API(Façade)
硬件
驅動

- 底层提供服务给上层调用。
- User希望AP稳定不变。
- AP希望Java框架不变。
- Java框架期待C/C++模块不变。
- C/C++模块期待驱动稳定不变。
- 人人都期待底下的“平台”不变。

古典观点

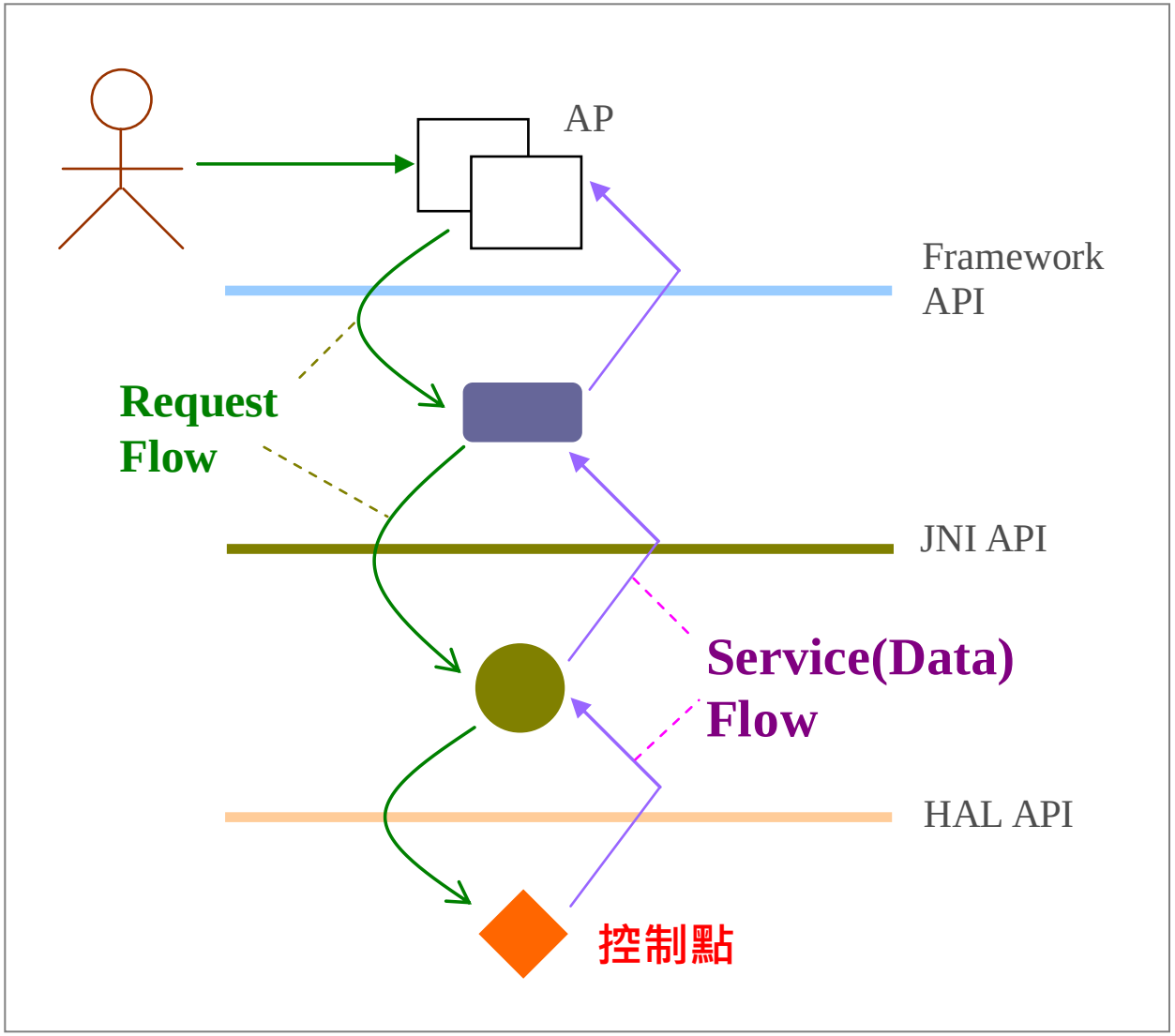
新观点



两种观点的差异

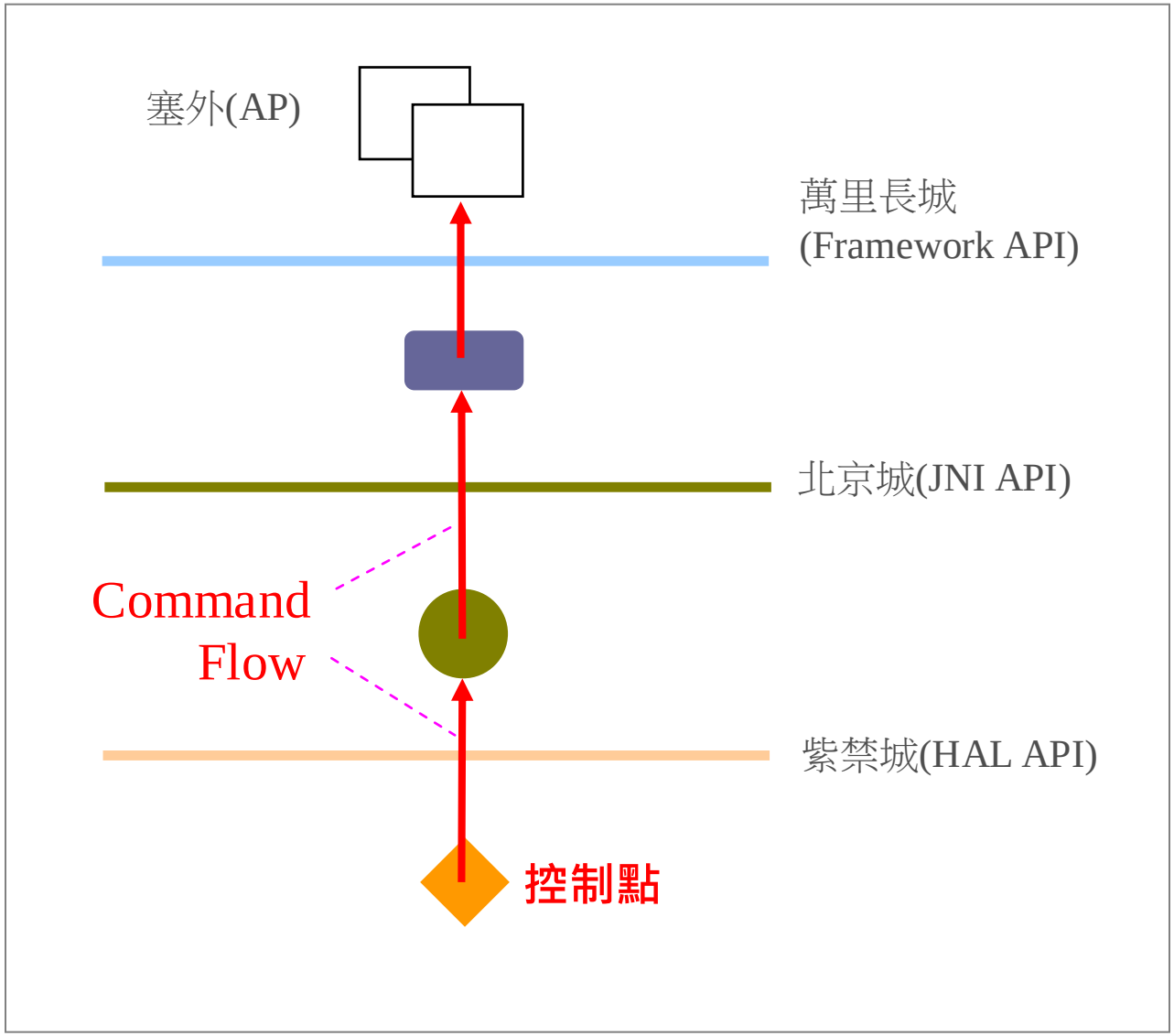
- 地位尊卑顺序相反。
- 行为决策相反。
- 老观点，人人争先恐后做 AP。
- 反之，新观点，争先恐后做框架和API。
- 新观点，底层先获利。万里长城让关内居民先获利。有利于软硬整合厂家。

古典观点下的两条流程

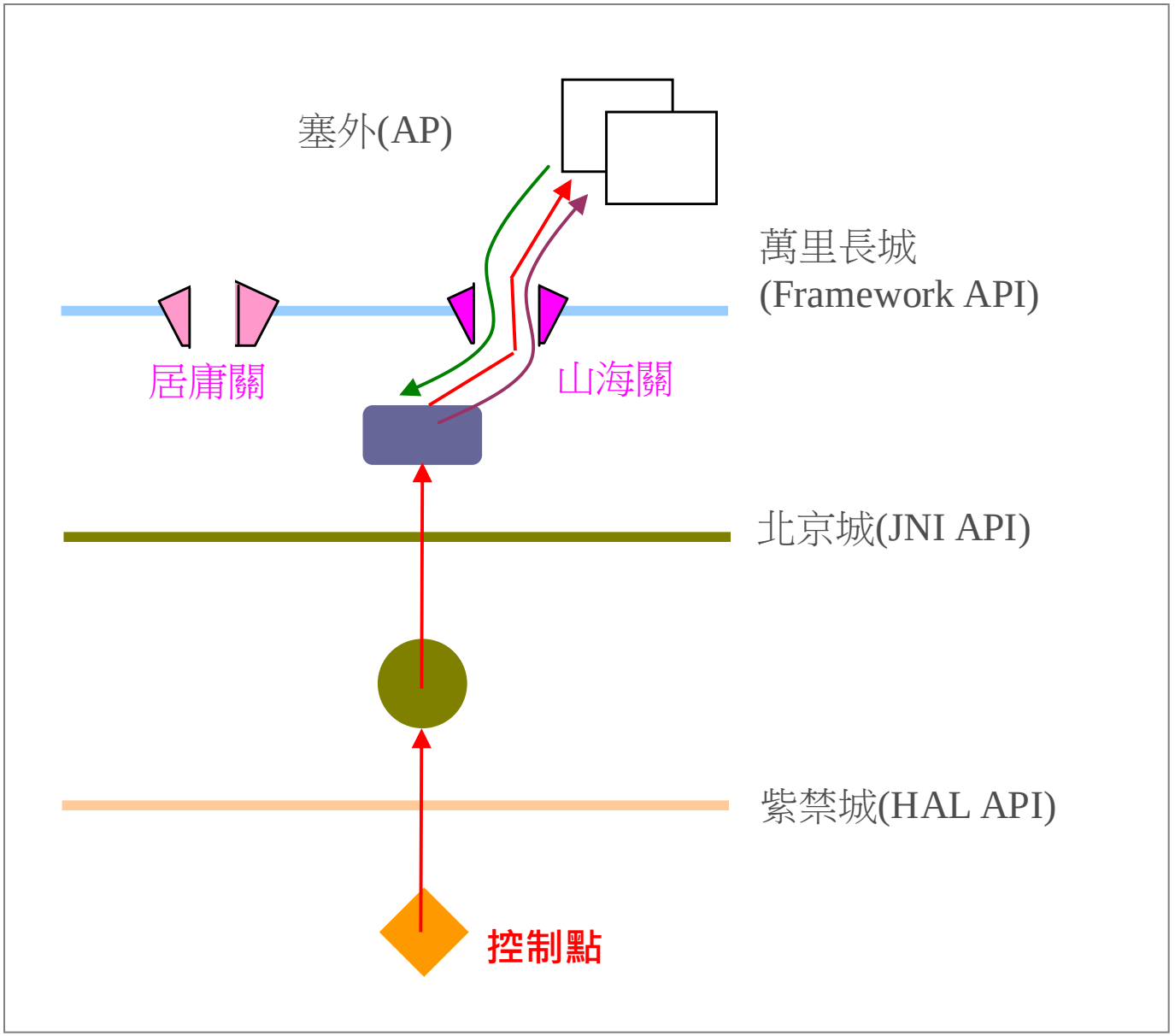

AP
Framework API
Request Flow
JNI API
Service(Data) Flow
HAL API
控制點

- 这不重视另一项流程：命令流程(Command Flow)。误认为请求流程 ==命令流程。
- 于是，命令来自业主或AP，底层硬件厂成为长工，难以实现软硬整合。
- 大家都知道命令的来源是紫禁城内，流向北京城外，再流到长城之外。

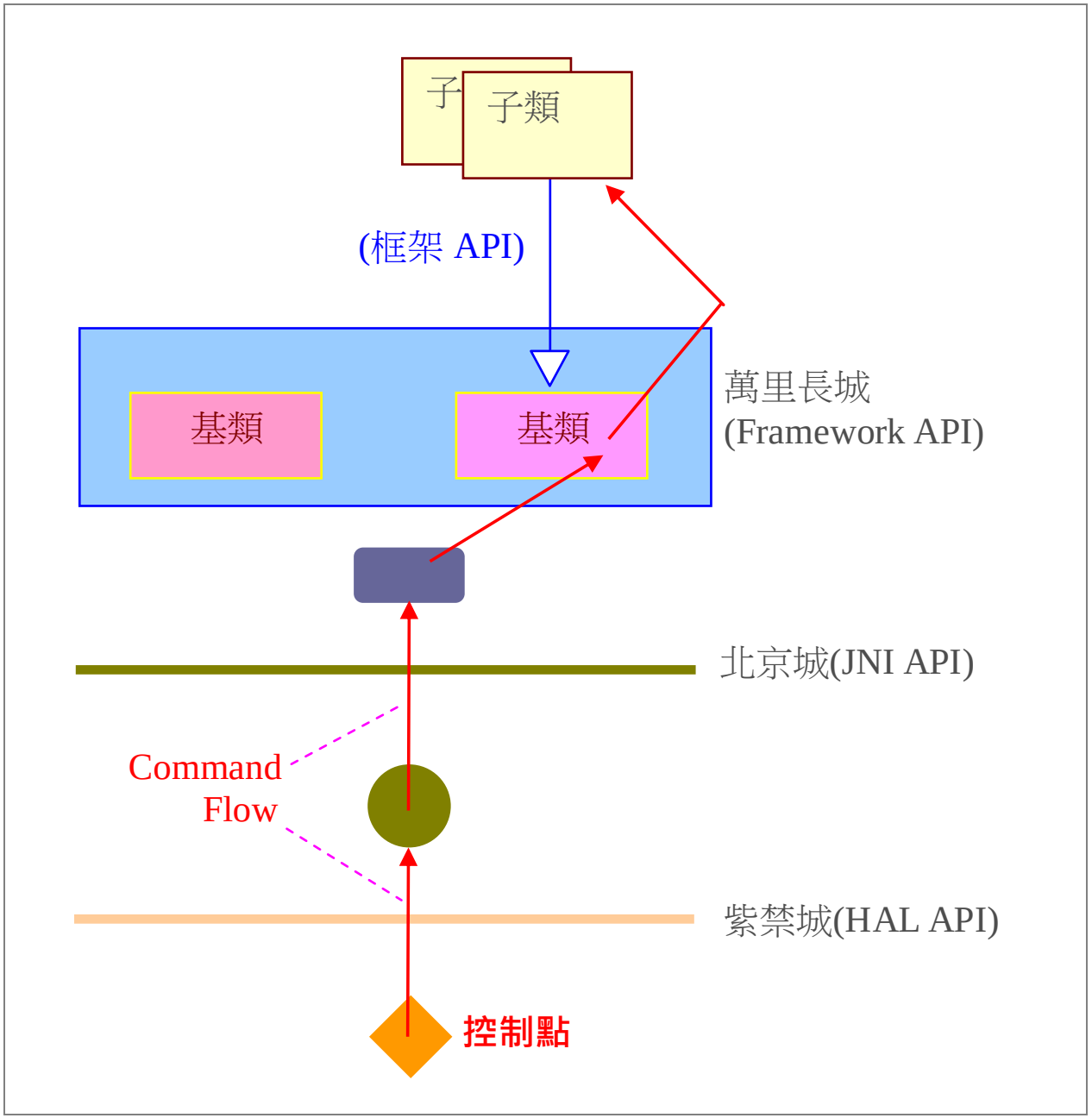
新观点下的
命令流程

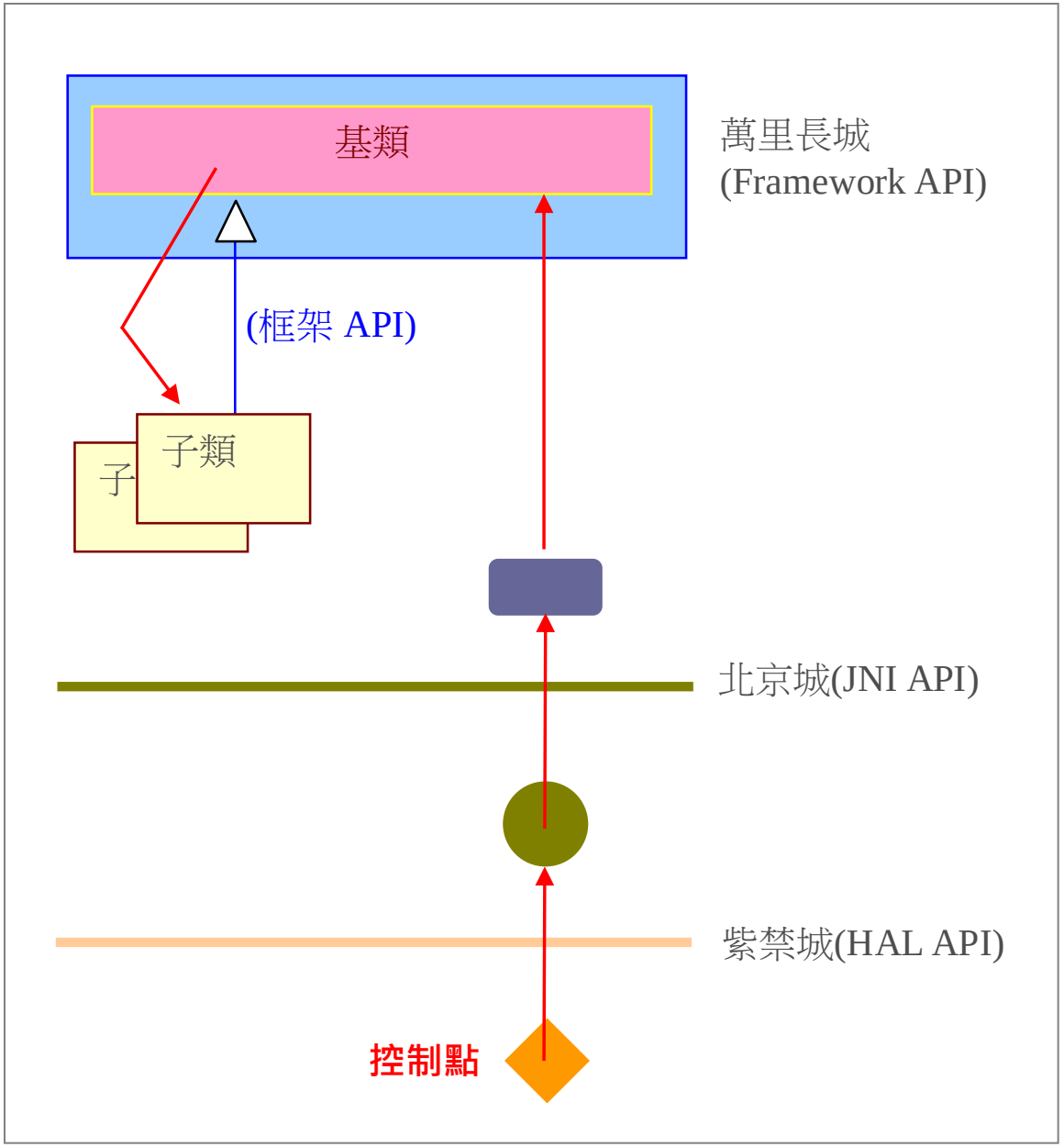

塞外(AP)
萬里長城
(Framework API)
北京城(JNI API)
Command
Flow
紫禁城(HAL API)
控制點

- 紫禁城内清朝皇帝的主导地位，必然会最第一道防线(即万里长城)设立关口，并重兵驻守，成为具有高度主导性的接口，例如山海关、居庸关等。
- 唯有主导性API(或称接口，或称关口)才能确保命令的传递和执行。在此平台里，硬件差异化才能凸显于API上，与App相会合。

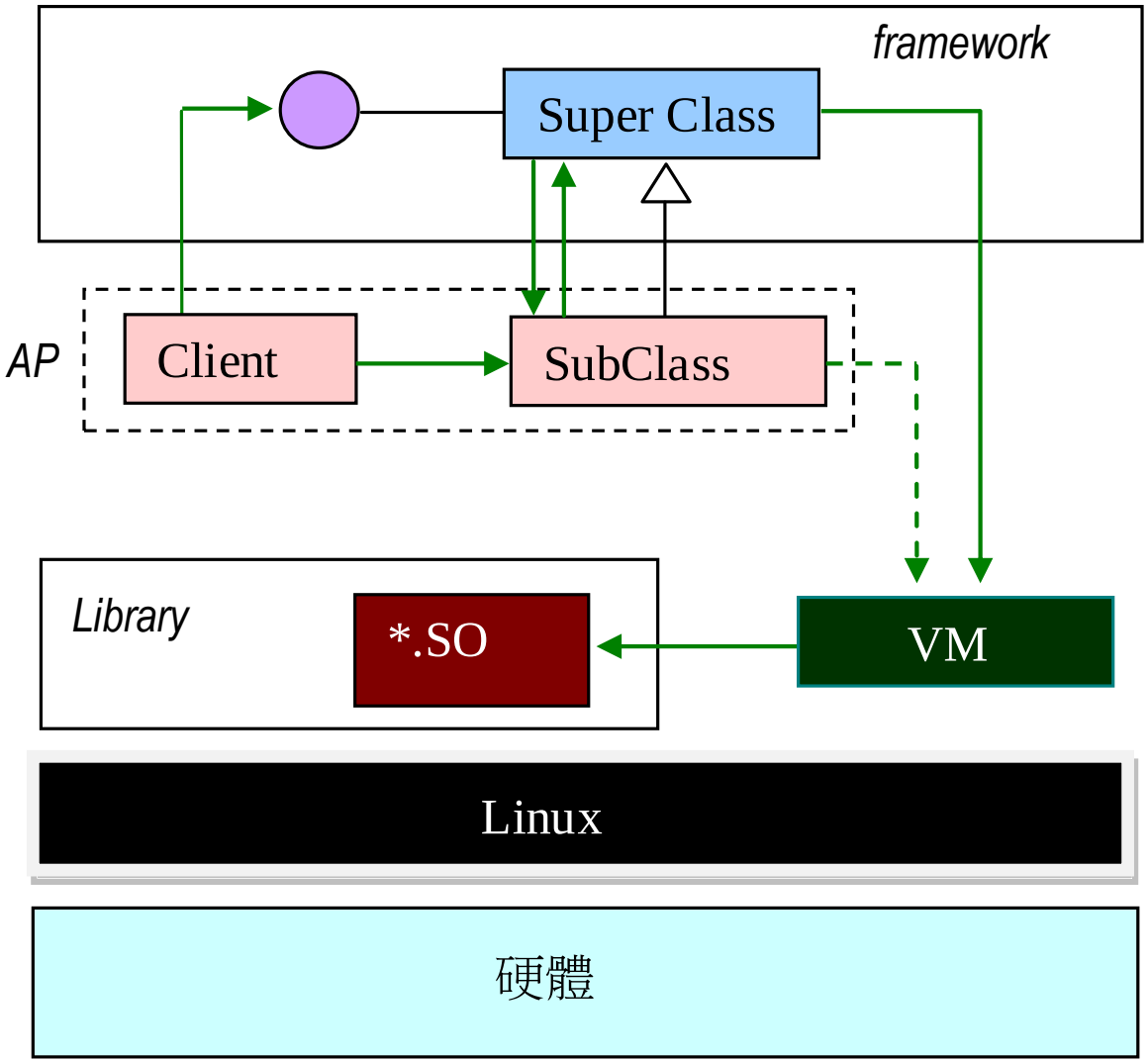

塞外(AP)
萬里長城
(Framework API)
居庸關
山海關
北京城(JNI API)
紫禁城(HAL API)
控制點

软件实践
技术：EIT


子
子類
(框架 API)
基類
基類
萬里長城
(Framework API)
北京城(JNI API)
Command
Flow
紫禁城(HAL API)
控制點


基類
萬里長城
(Framework API)
(框架 API)
子類
子
北京城(JNI API)
紫禁城(HAL API)
控制點

Android 框架


framework
Super Class
AP
Client
SubClass
Library
*.SO
VM
Linux
硬體

- C/C++掌握主导权(话语权)、拥有控制点的更多表现：

  除了C函数调用Java层函数之外，还有：

  1. C函数存取Java对象的属性值。
  2. C函数创建Java层的对象(object)。

~ Continued ~