



G04_接口设计之美_多层级 EIT 造形和框架

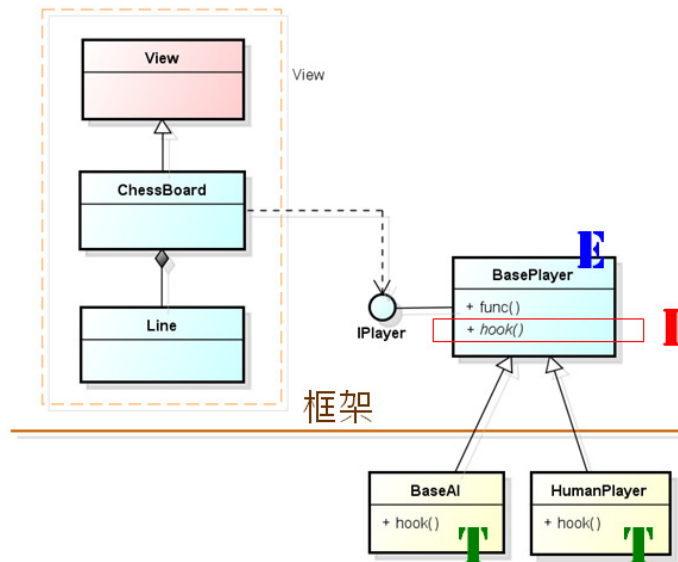
内容：

1. 建立多层 EIT 造形与多层框架
2. 基于多层框架，掌握 Android 架构的知识体系

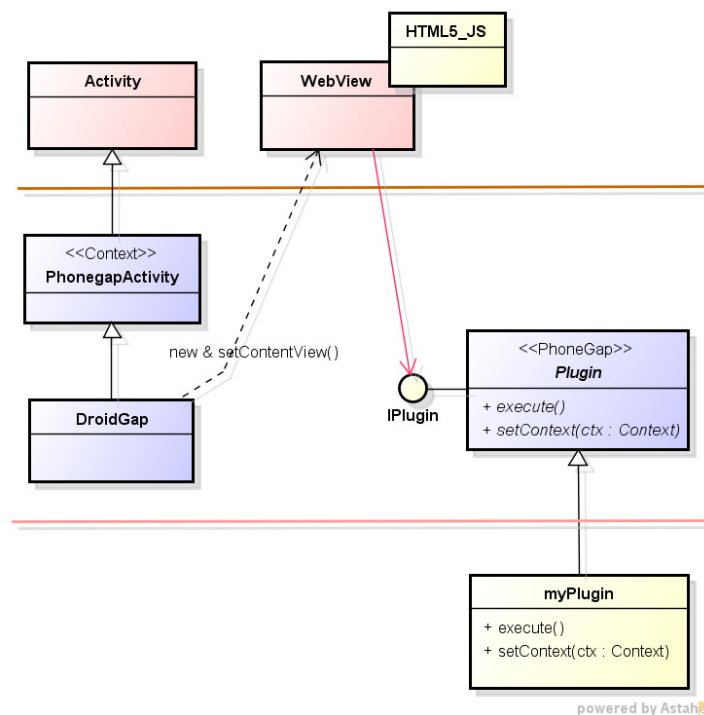
- ◇ 建立多层级 EIT 造形和框架体系
- ◇ 基於多层框架，掌握 Android 架构的知识体系

前言：

- 上一回合的<五子棋>框架是属于单层框架：



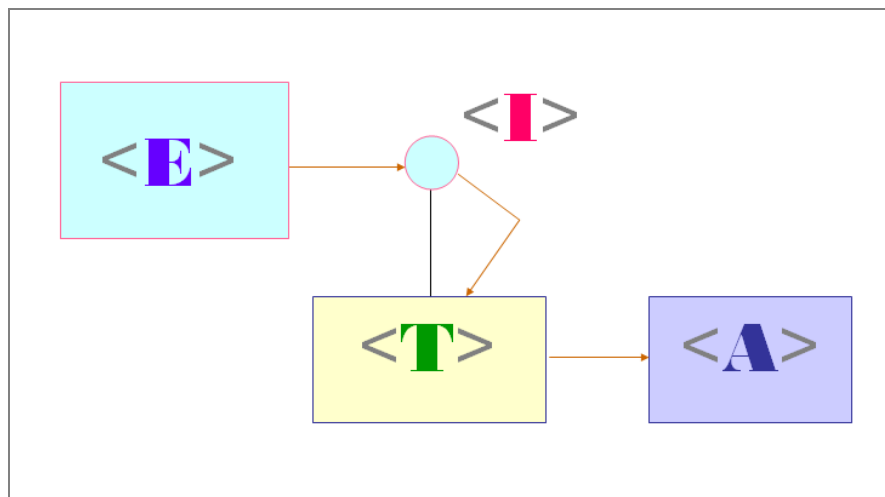
- EIT 造形可以重复组合，所以能组合出多层级的 EIT 造形体系；也因此而很容易设计出多层级框架体系。



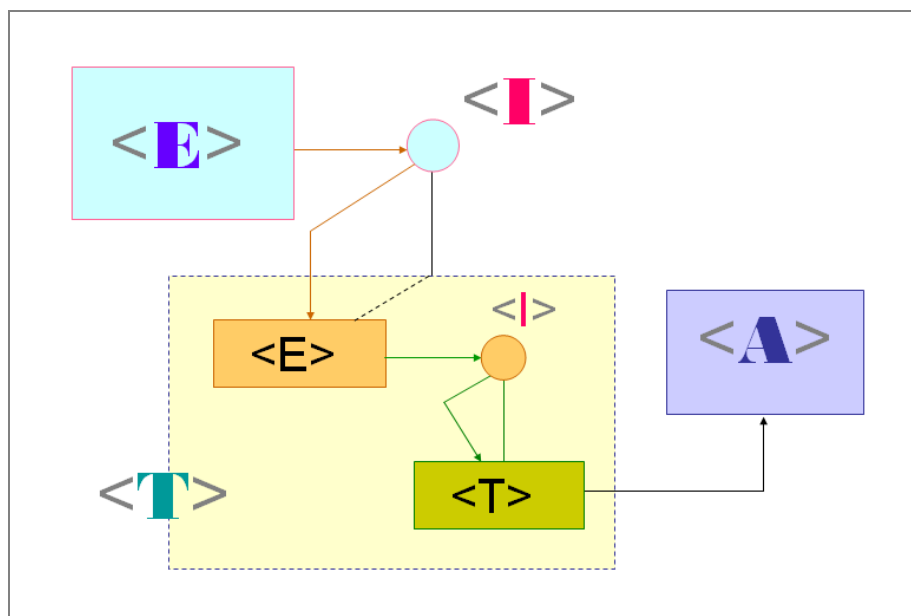
1. 建立多层 EIT 造形与框架体系

上一节已经说明了，如何基于 EIT 造形而建立一个框架。由于 EIT 造形本身具备重复组合的特性，其意味着，能够组合而成为多层级的 EIT 造形。然后，基于多层级的 EIT 造形，就能顺利建立(定义)出多层级(Multi-Layer)的框架体系了。

例如，有一个单层的 EIT 造形，如下图：

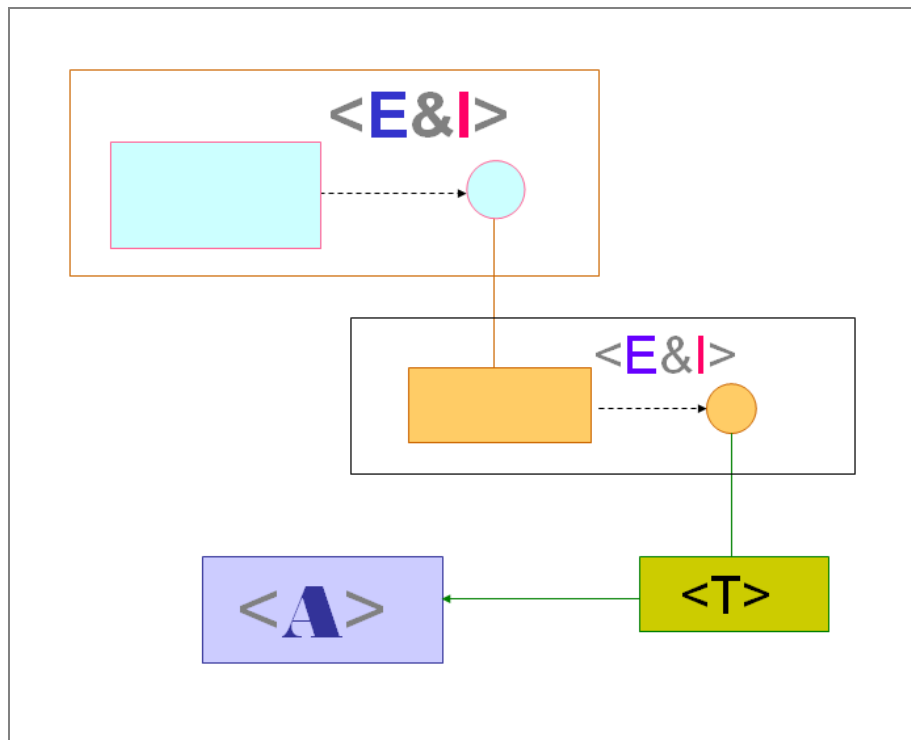


此时，只要让<T>里内含一个小的 EIT 造形，就成为双层的 EIT 造形了。

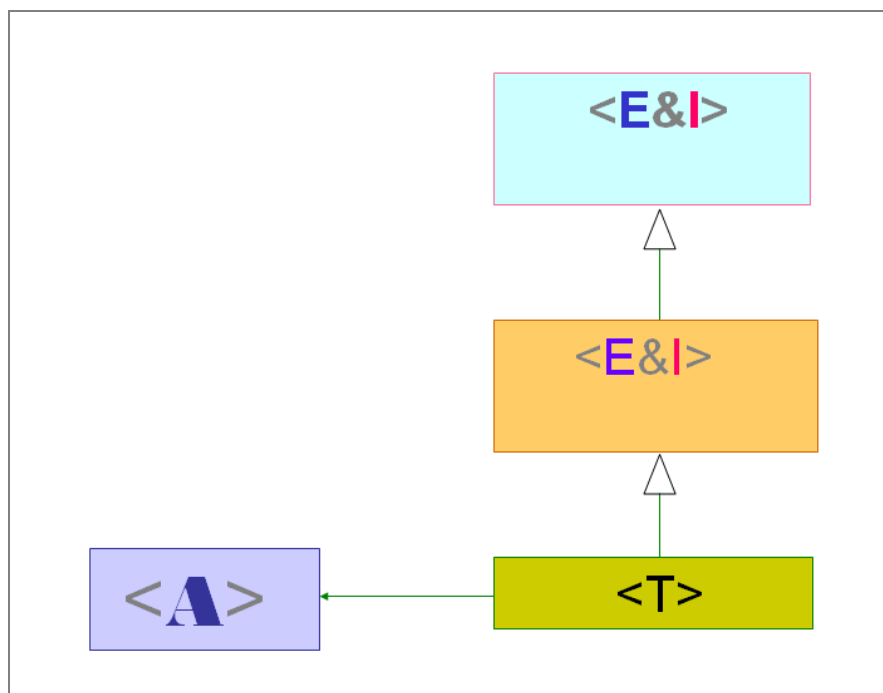


其中，最简单的结构就是：大 EIT 造形的<T>等于小 EIT 造形的<E>，就成

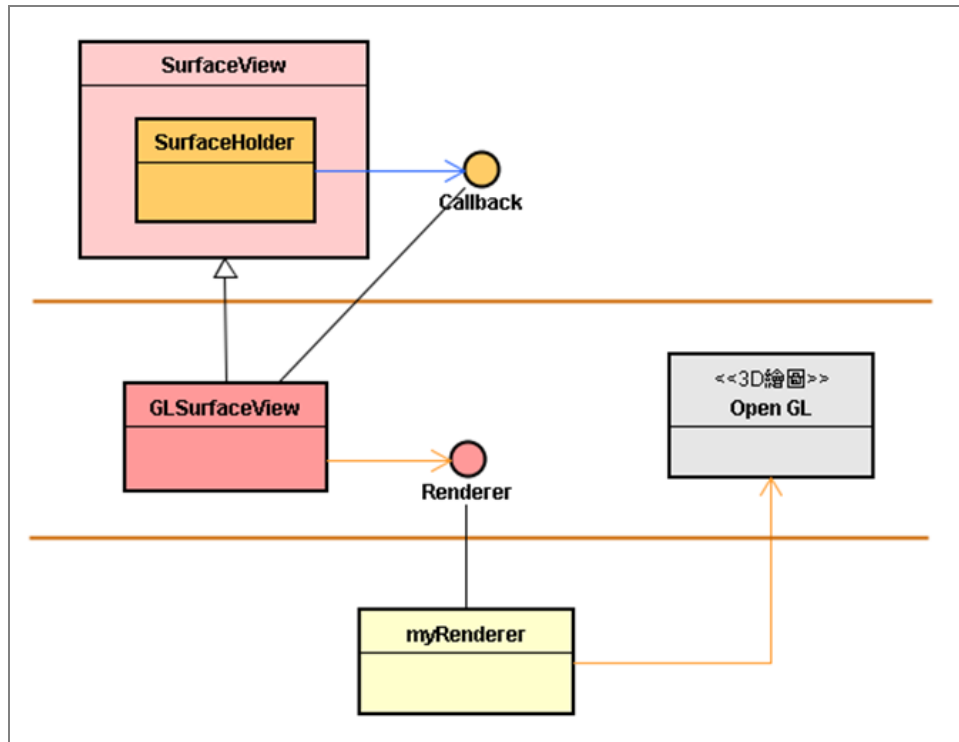
现出最简单的双层 EIT 造形了。



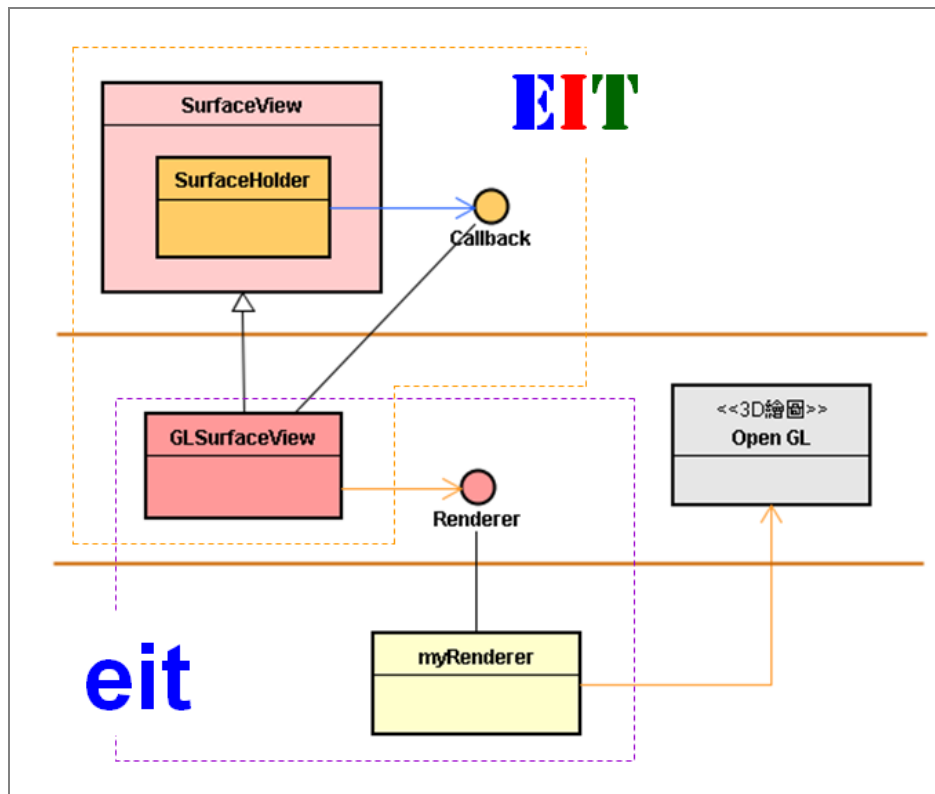
上图是采取一般的 EIT 造形结构。如果改为“子类 extends 基类”的结构，就如下图所示：



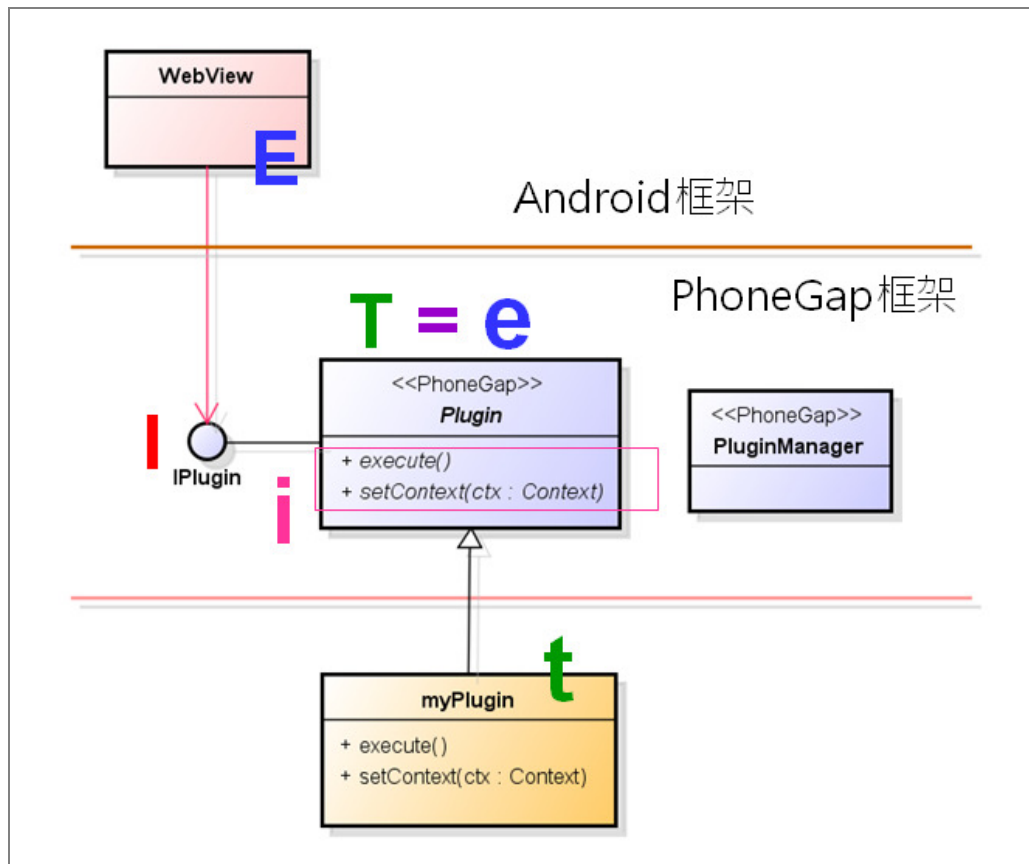
上述两个图的涵意是一致的。重复上述的规则和步骤，就能建立出更多层级的 EIT 造形体系了。就像在 Android 框架里，也处处可见到多层级的 EIT 造形结构，例如：



其中的 **GLSurfaceView** 既扮演上层 EIT 造形的 **<T>**，又扮演下层 EIT 造形的 **<E>**。就是一个简单的双层 EIT 造形了。

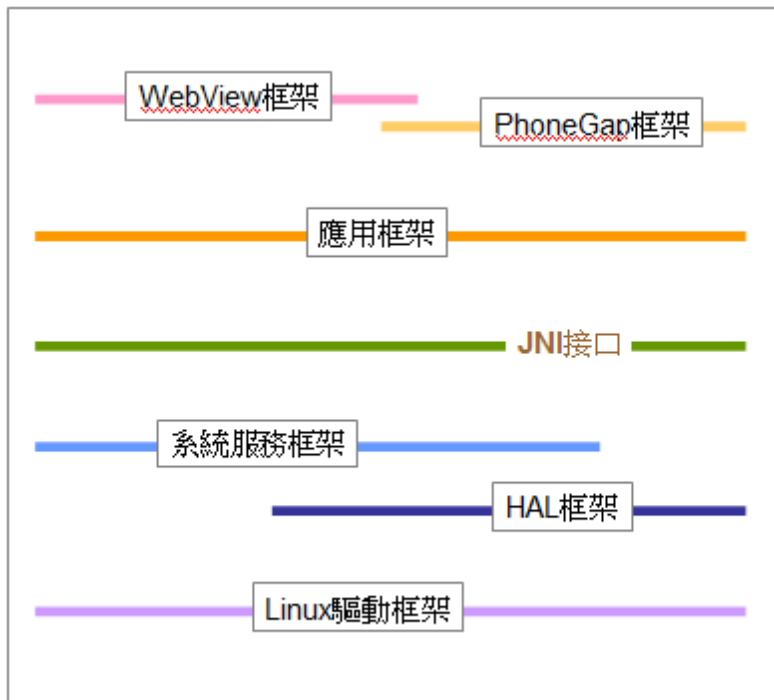


这两层 EIT 造形都属于 Android 平台的 Java 层应用框架(Application Framework)的内容 这两层 EIT 造形都是 Android 团队所开发的 由于 Android 是十分开源又开放的平台，人人(或各厂商)皆能接 Android 里的 EIT 造形，而扩充出自己的 EIT 造形，成为自己设计和开发的小框架。例如，PhoneGap 就是有第三方公司所扩充出来的小框架，来实践跨平台(如 Android、iOS 等)的功能；如下图所示的多层级 EIT 造形。

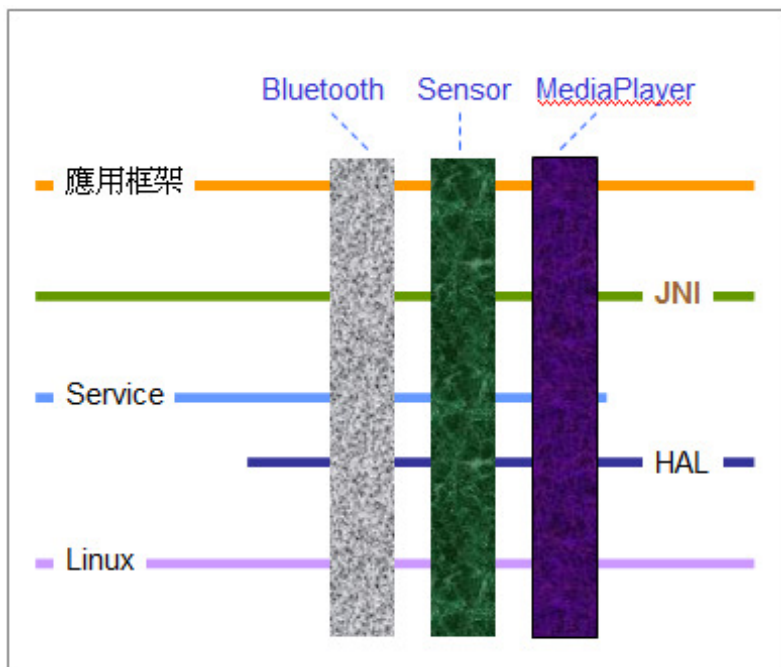


2. 基於多层框架，掌握 Android 架构的知识体系

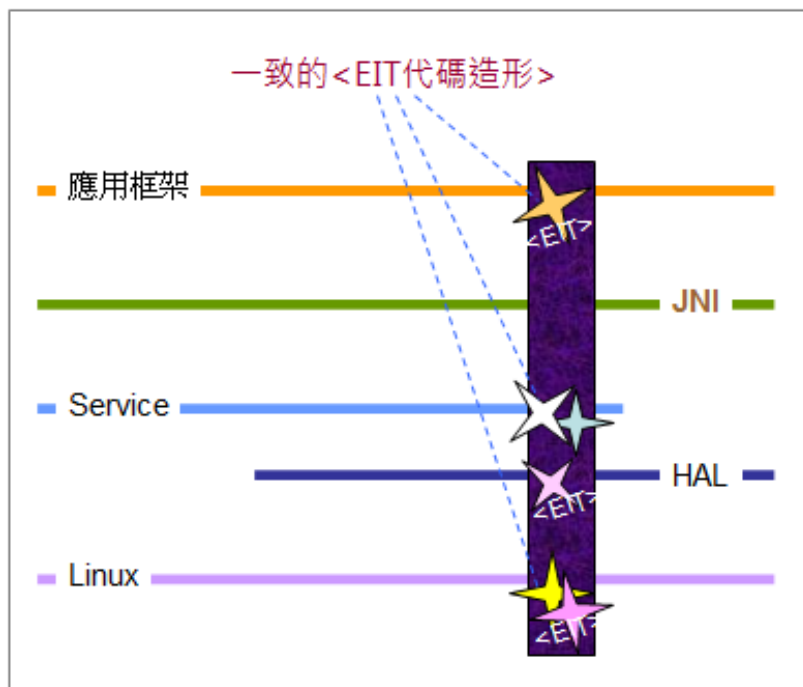
欲掌握 Android 的知识体系，从框架角度切入，可以找到它的甜点(Sweet Spot)。由于它是一个开源开放的架构，我们可以直接切入核心，看到树干结构，一目了然；而不必像 iOS、Win8 等封闭平台，只能从外部功能(树叶)去猜测底层架构。所以，欲掌握 Android 架构体系，从它的多层框架体系视角切入，是最有效的途径。其框架体系如下：



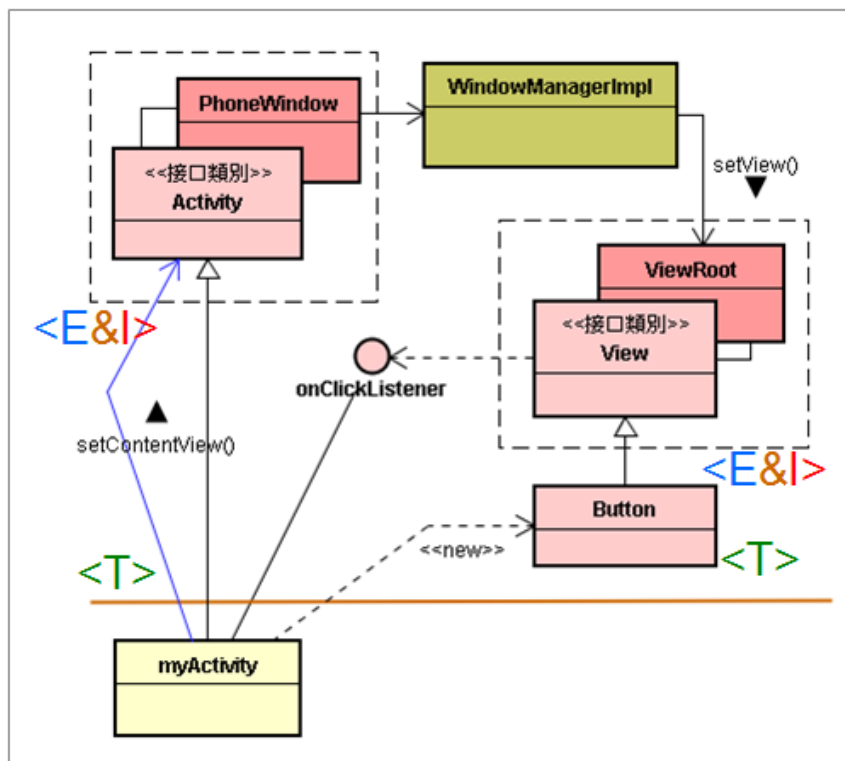
基于这个框架体系，再将众多功能(子系统)的模块添挂上去，每一项子系统就如同一棵完整的树，有树叶、树干、树根等。例如，MediaPlayer 播放功能，就含有 Java、JNI、系统服务、HAL 等完整的体系(即一棵完整的树)；如下图所示：



Android 就如同一座森林，由 Bluetooth、MediaPlayer 等众多的树所组成的。上图看来是蛮有规律的，然而更具规律性的是：各层框架的基本元素，其代码造形(Code Form)是一致的，如下图：



例如，在 Activity-View 架构看似很复杂，其实只是两个主要的 EIT 代码造形所组合而成的。如下图所示：



~ End ~