

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>

F03\_a

# 观摩：SurfaceView小框架 的特殊性设计(a)

By 高煥堂

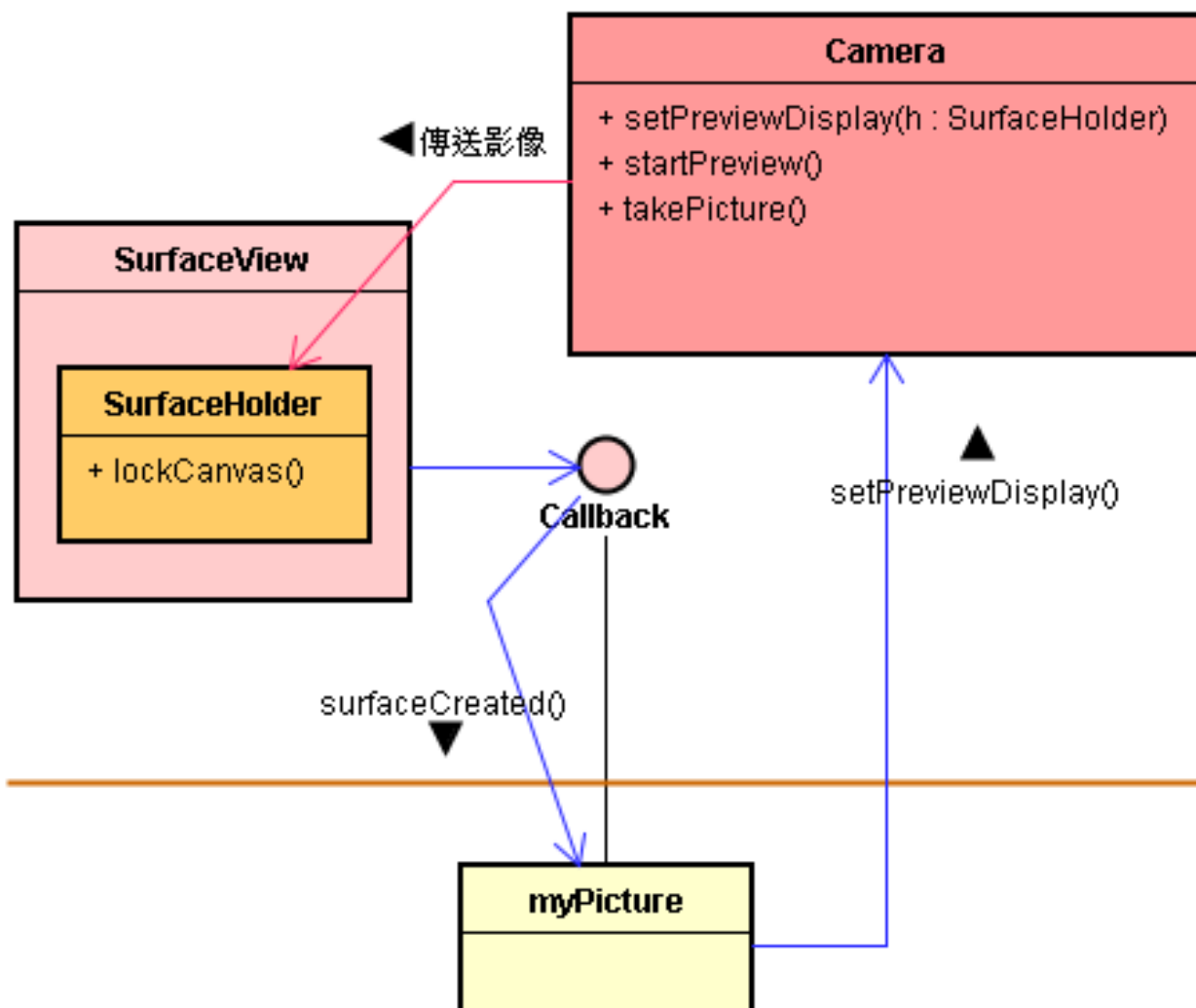
# 内容

1. 兼顾未来性与特殊性
2. 使用特殊性接口：Renderer
3. 范例：JNI + OpenGL ES C引擎
  - 范例(一)
  - 范例(二)

# 1、兼顾未来性与特殊性

回顾：未来性

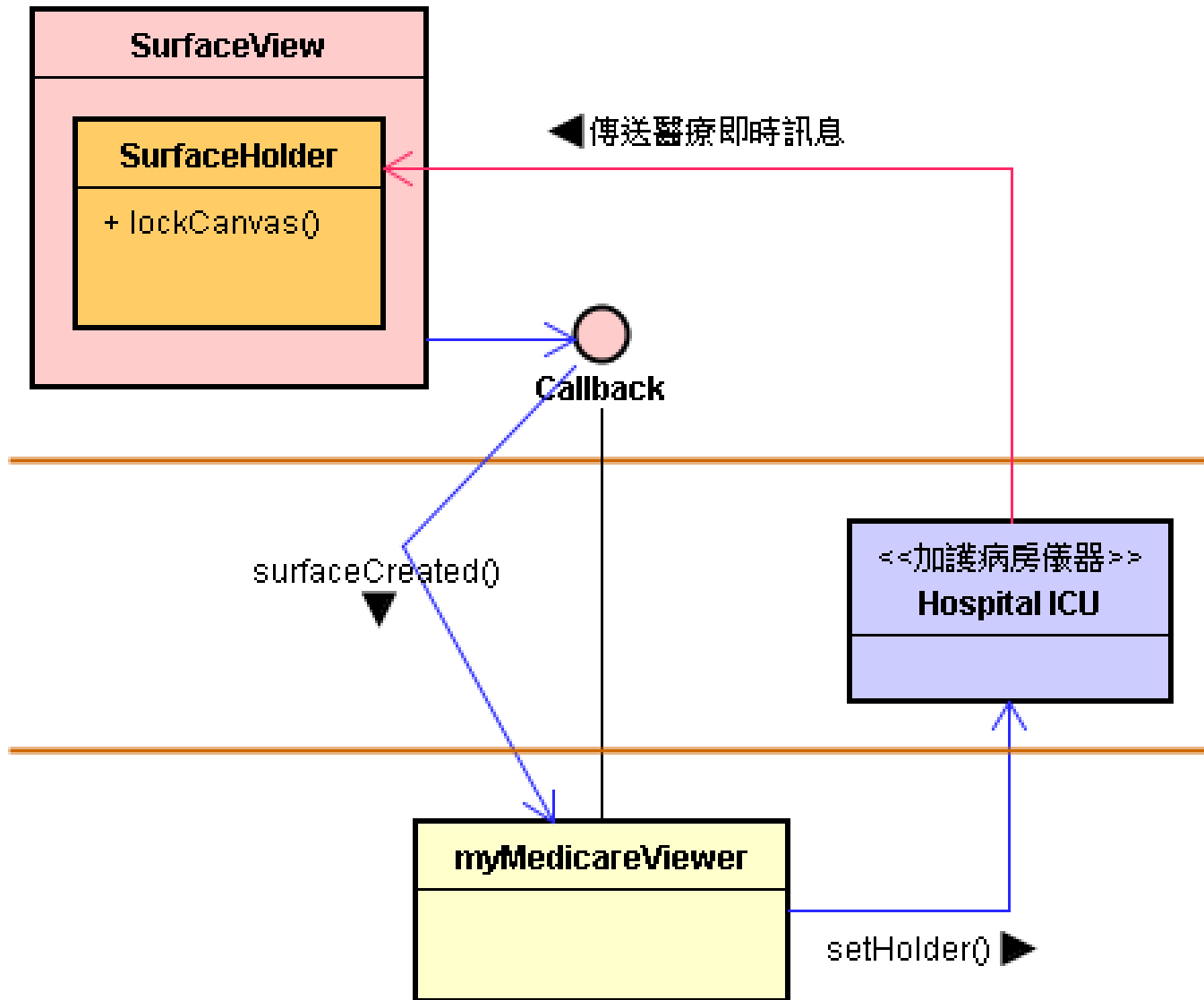
- 架构师专注的不是未来决策，而是目前决策的未来性。
- 于是，表现在软件架构设计(即决策)的未来性上，其最为明显之处就是：包容未来的变化；包括用户(User)未来可能的选择等等。  
。 例如：



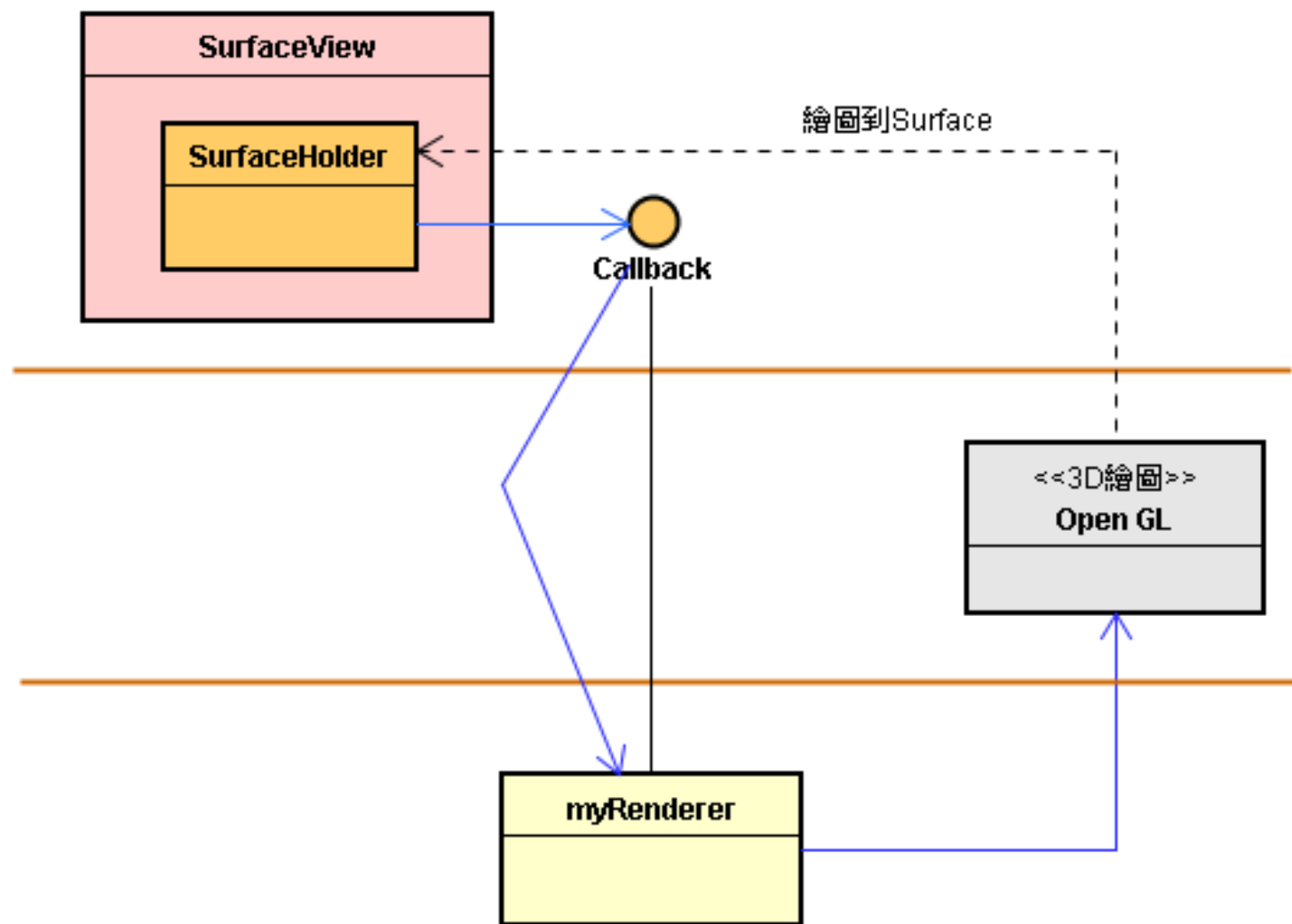
- 未来性大多是来自于通用性的接口。例如下图里的通用性Callback接口，创造了高度的未来性。
- 这通用性接口，让业主出现了，选择了Camera设备之后，App开发者才写myPicture应用子类。

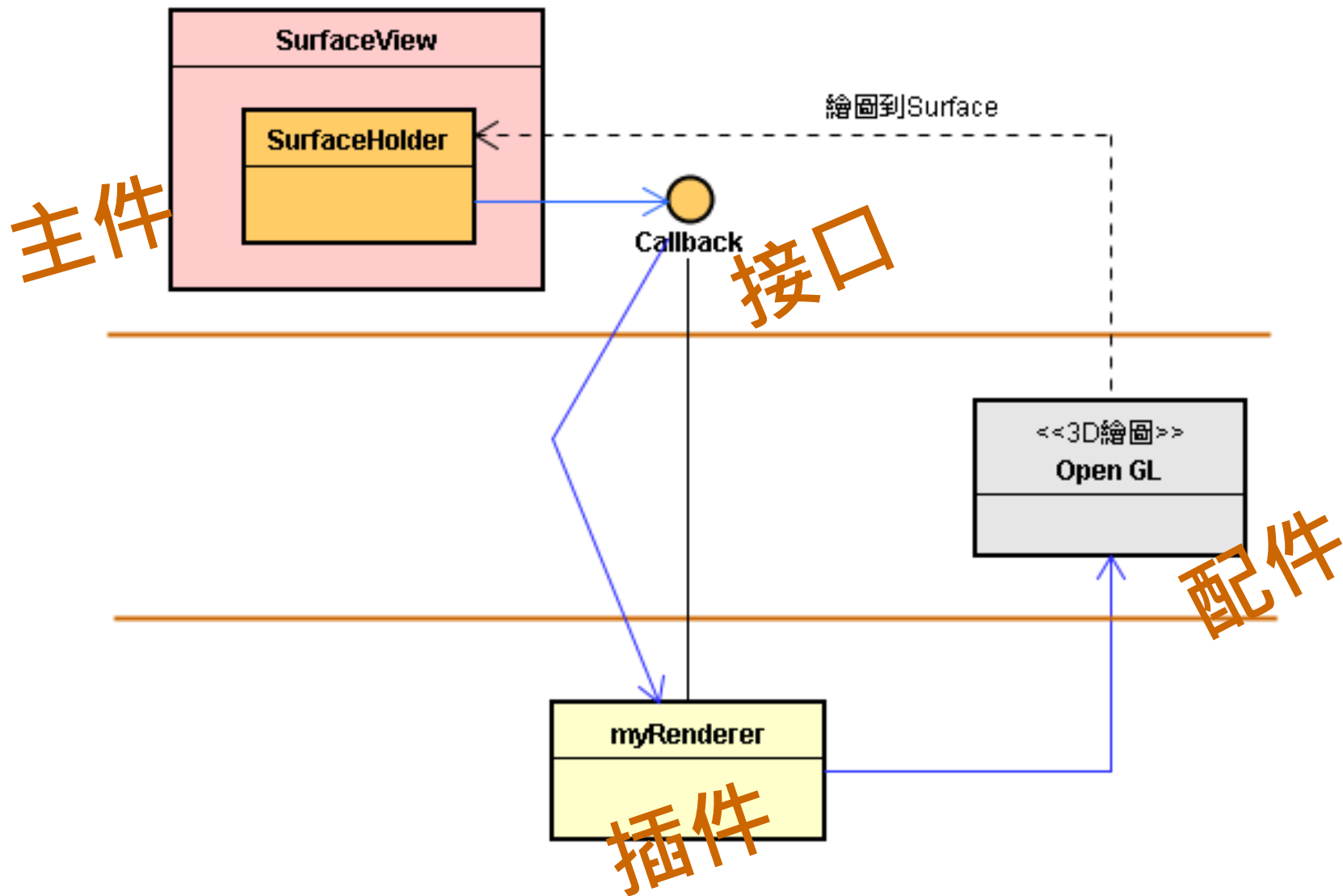


- 未来新业主出现了，选择了不一样的设备，App开发者才写新的myPicture子类。
- 例如，选择把护士的Pad链接到医院加护病房的仪器设备上



- 凡是业主选择了不一样的设备，App开发者就写新的myPicture子类。
- 例如，选择把Pad联接到OpenGL ES绘图引擎上。





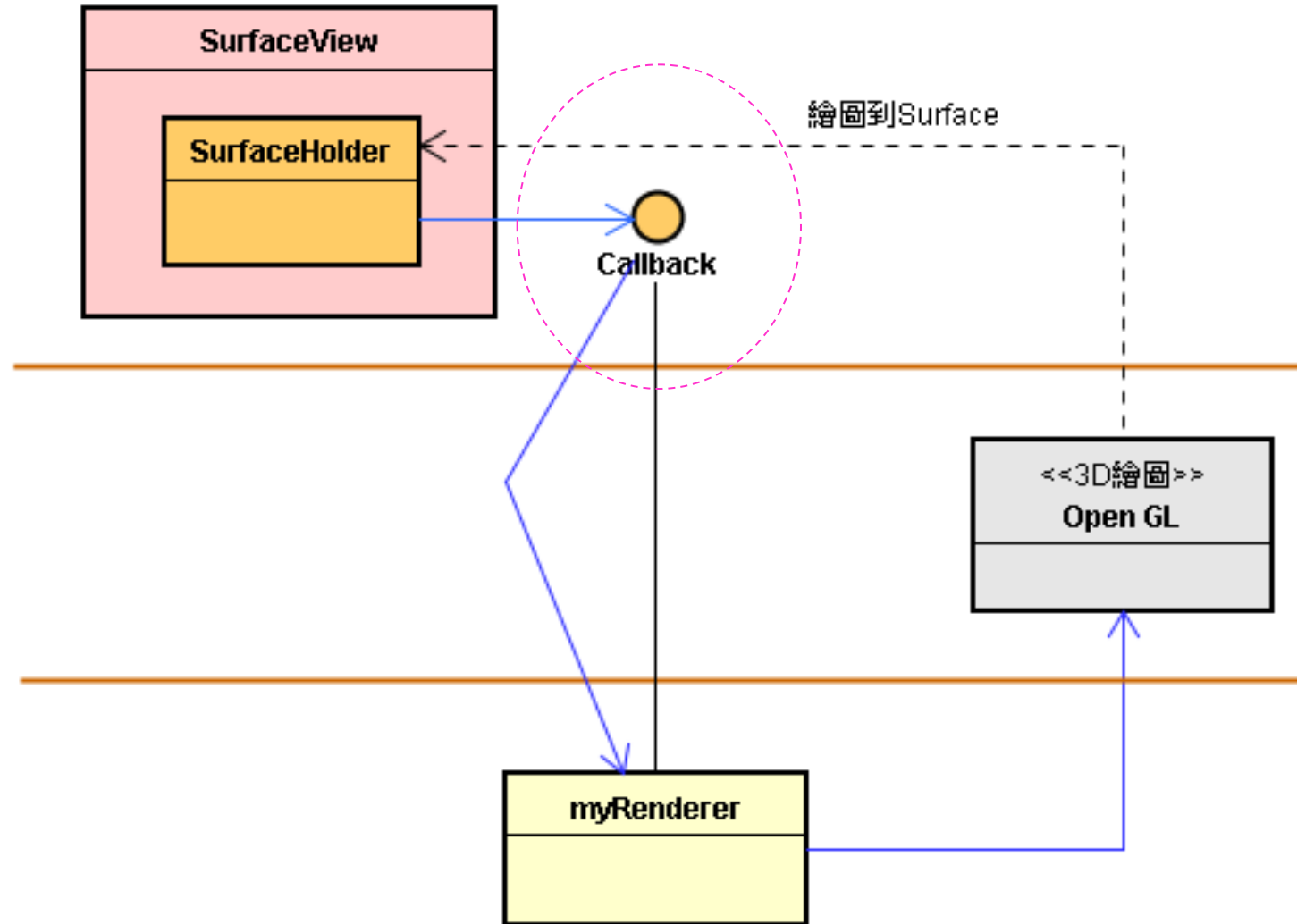
- SurfaceView基类和Callback接口，就是<E&I>，构成了一个小框架。这个框架就如同集装箱(container)。
- 它包容了形形色色的插件<T>和配件，亦即包容了业主需求的多变化，呈现了架构设计决策的未来性。

从未来性衍生出特殊性

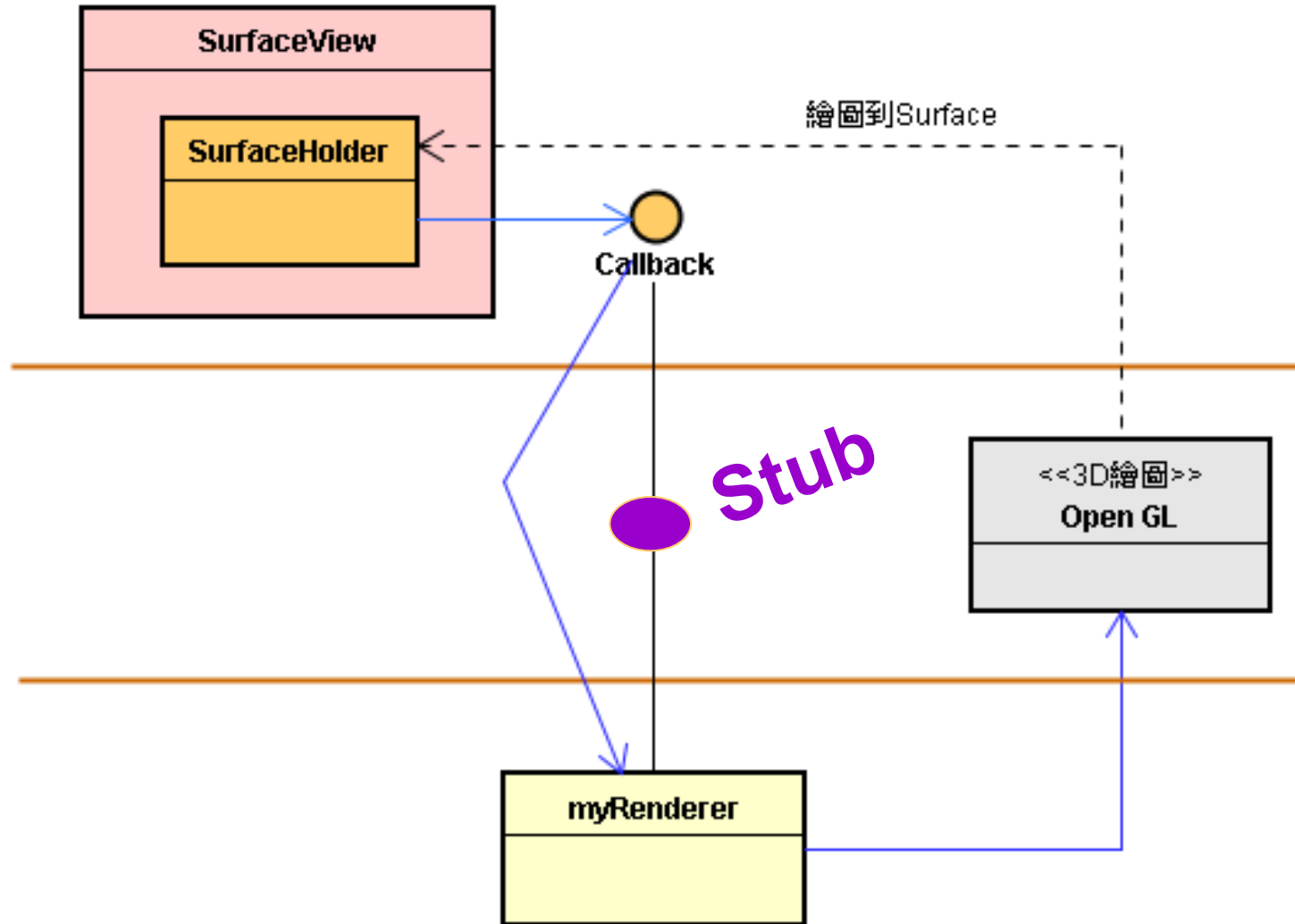
以Stub类改变接口

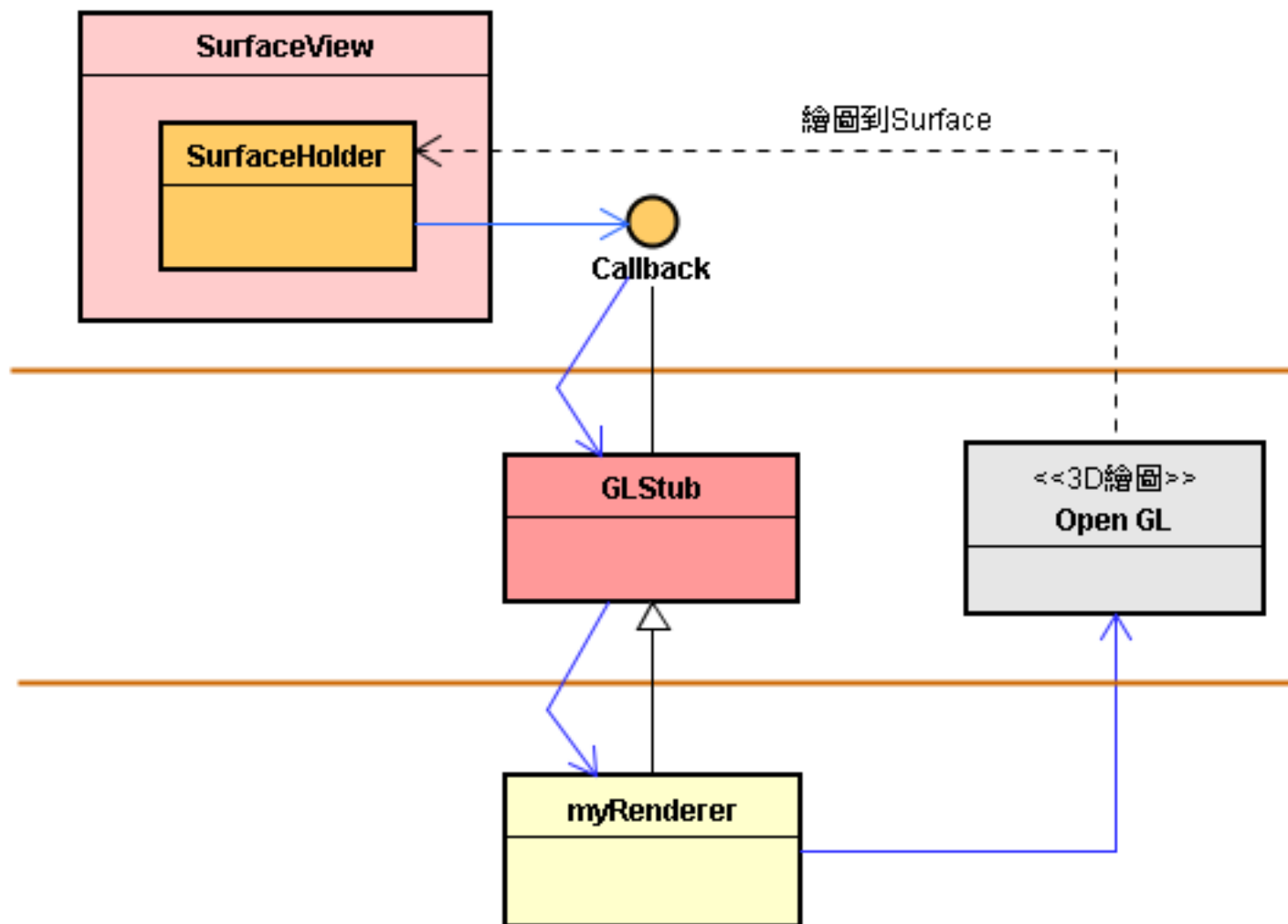
- 曹操类(Stub)能封装天子的接口，定义自己的新接口。
- 它通常封装了通用性(General)的接口，然后定义出比较特殊性(Special)新接口。
- 例如，将CameraService的通用性IBinder接口封装起来，同时定义出如ICamera、ICameraClient和ICameraService等比较特殊性的接口。



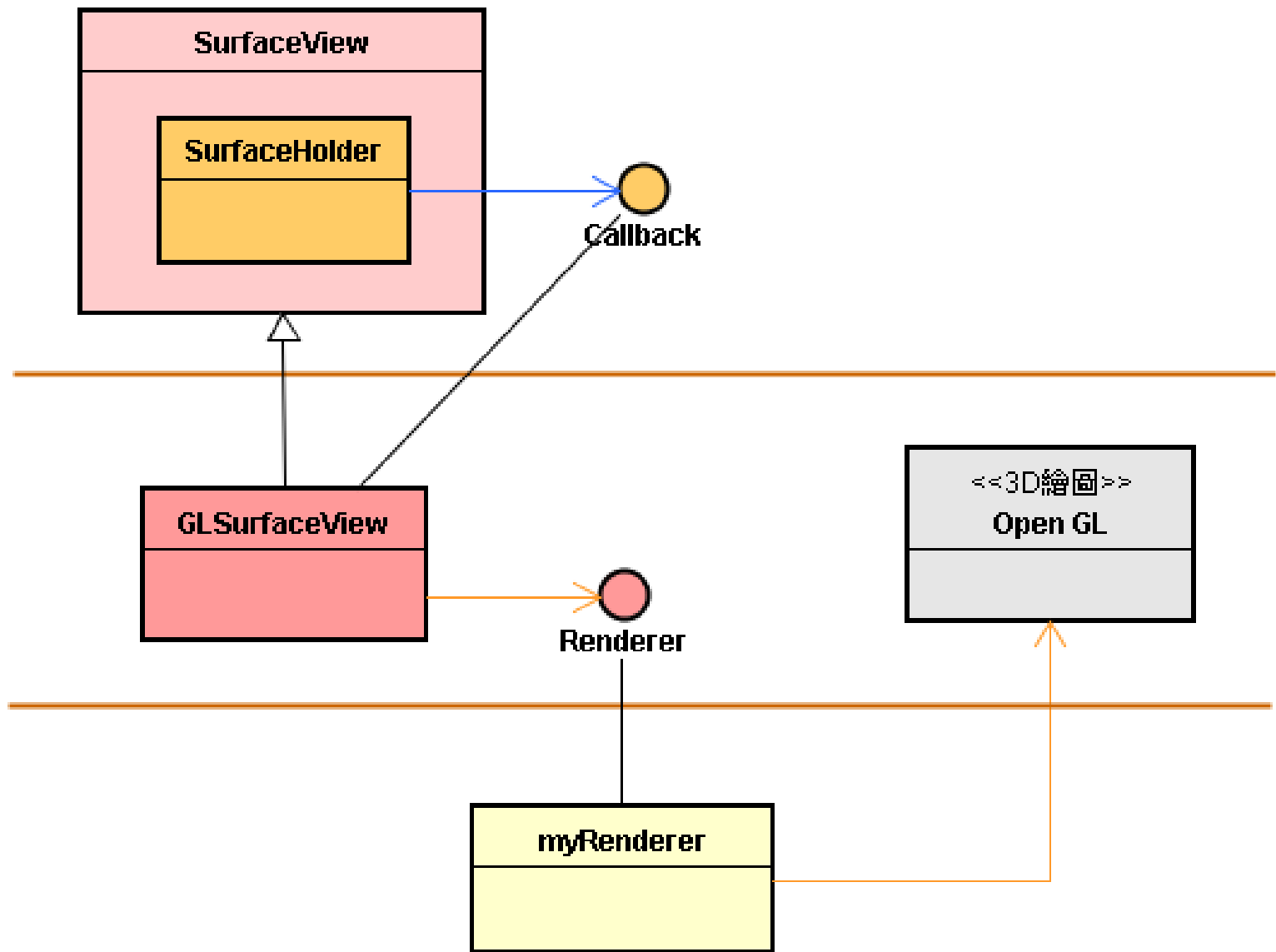


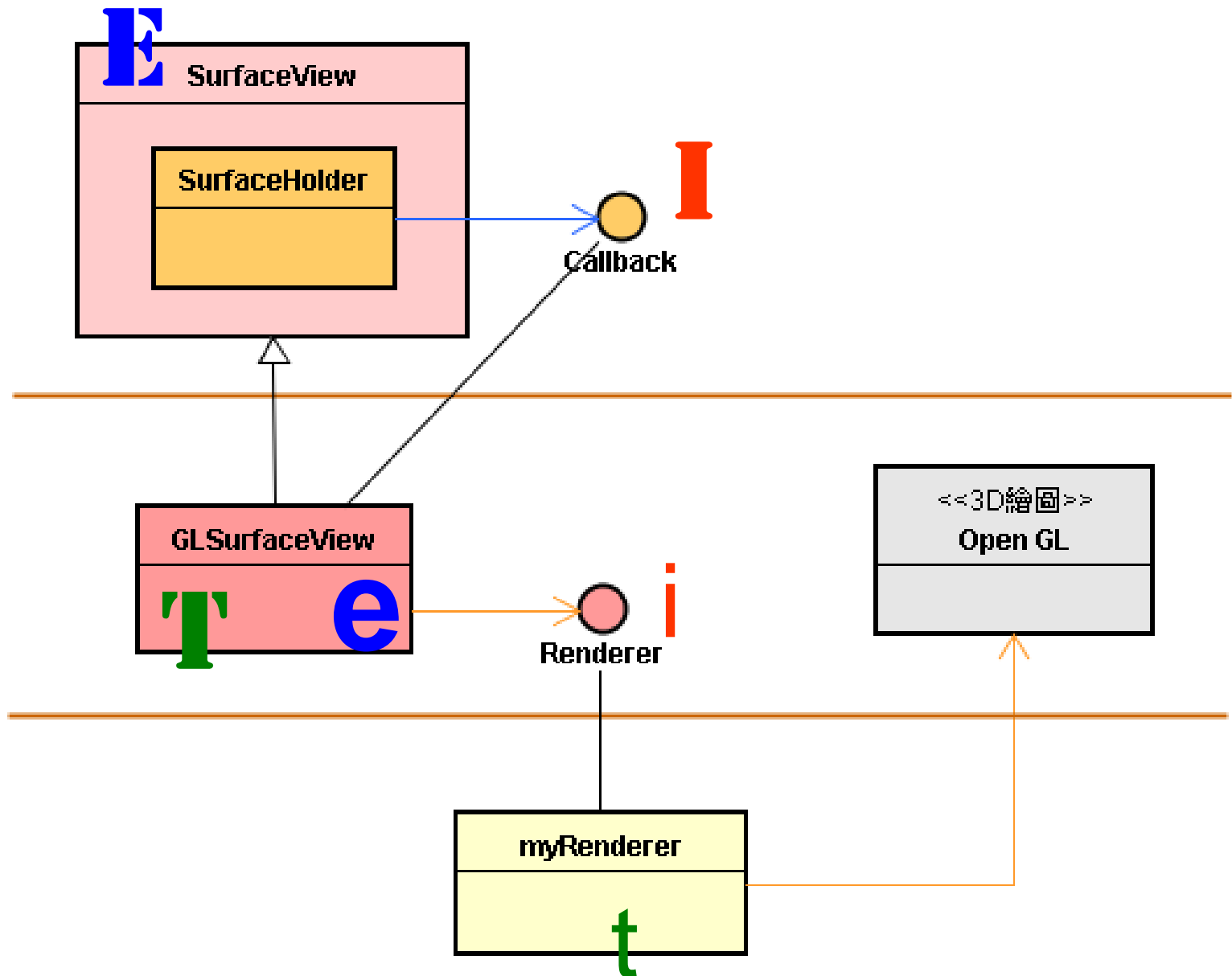
- 这里的Callback接口，就属于通用性接口。可以增添一个Stub类，来封装这个通用性接口，并提供较为特殊性的接口。



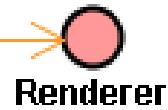
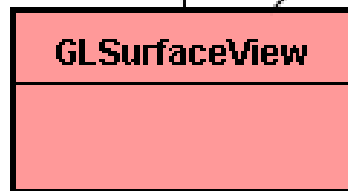
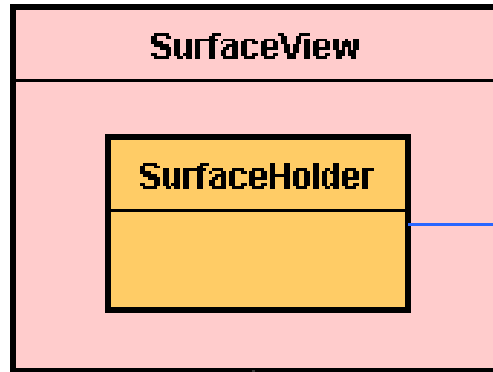


- 运用EIT的变形，建立起两层EIT造形。封装的上层EIT的<I>，定义出下层EIT的<T>。
- 目前Android平台，就提供了GLSurfaceView基类和Renderer接口的设计了。

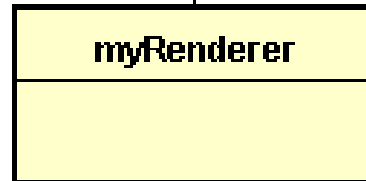




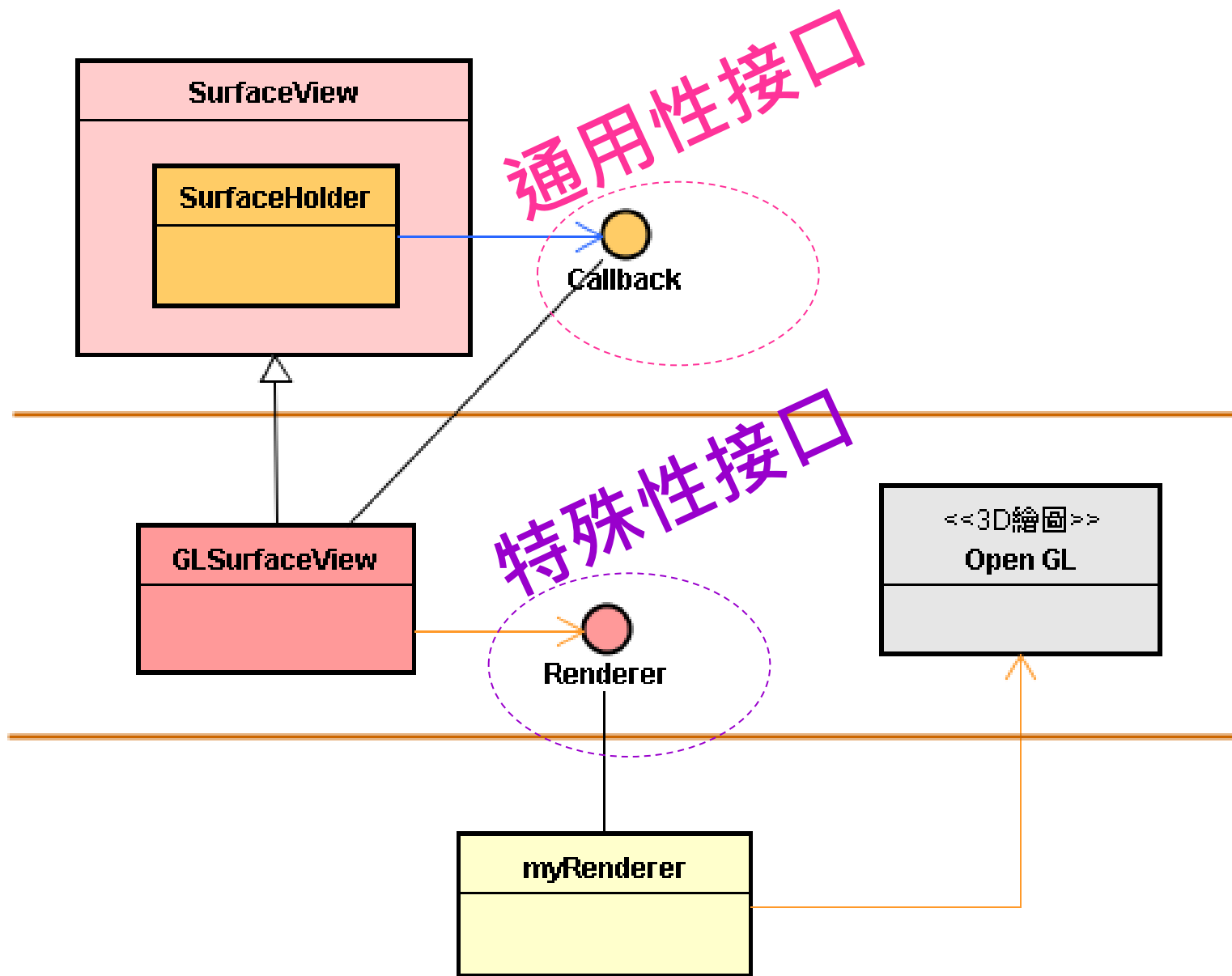
EIT



eit







- 这是兼顾未来性与特殊性的美妙架构设计。
  -



**~ Continued ~**