

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>

I06\_a

# 跨自己平台策略(a)

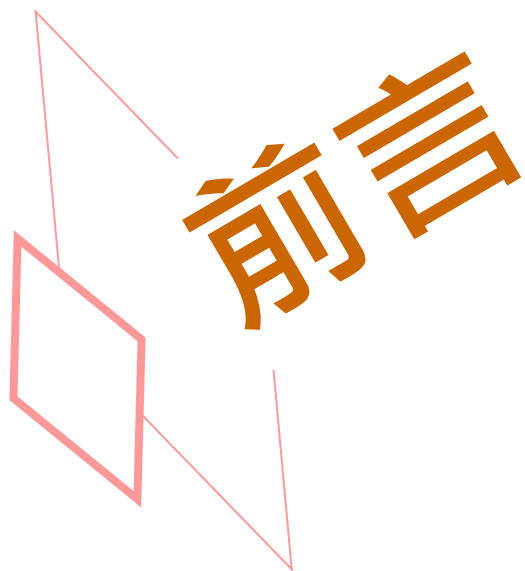
By 高煥堂

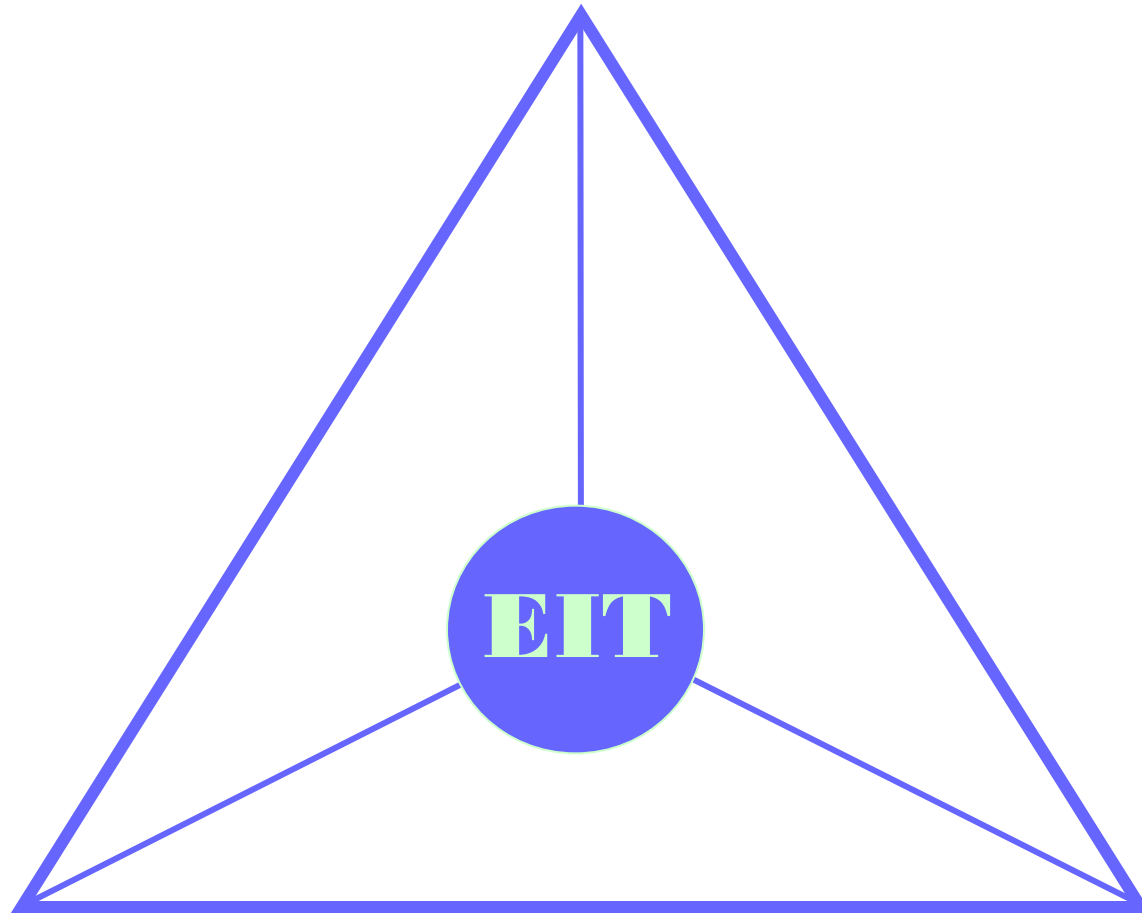
# 内容

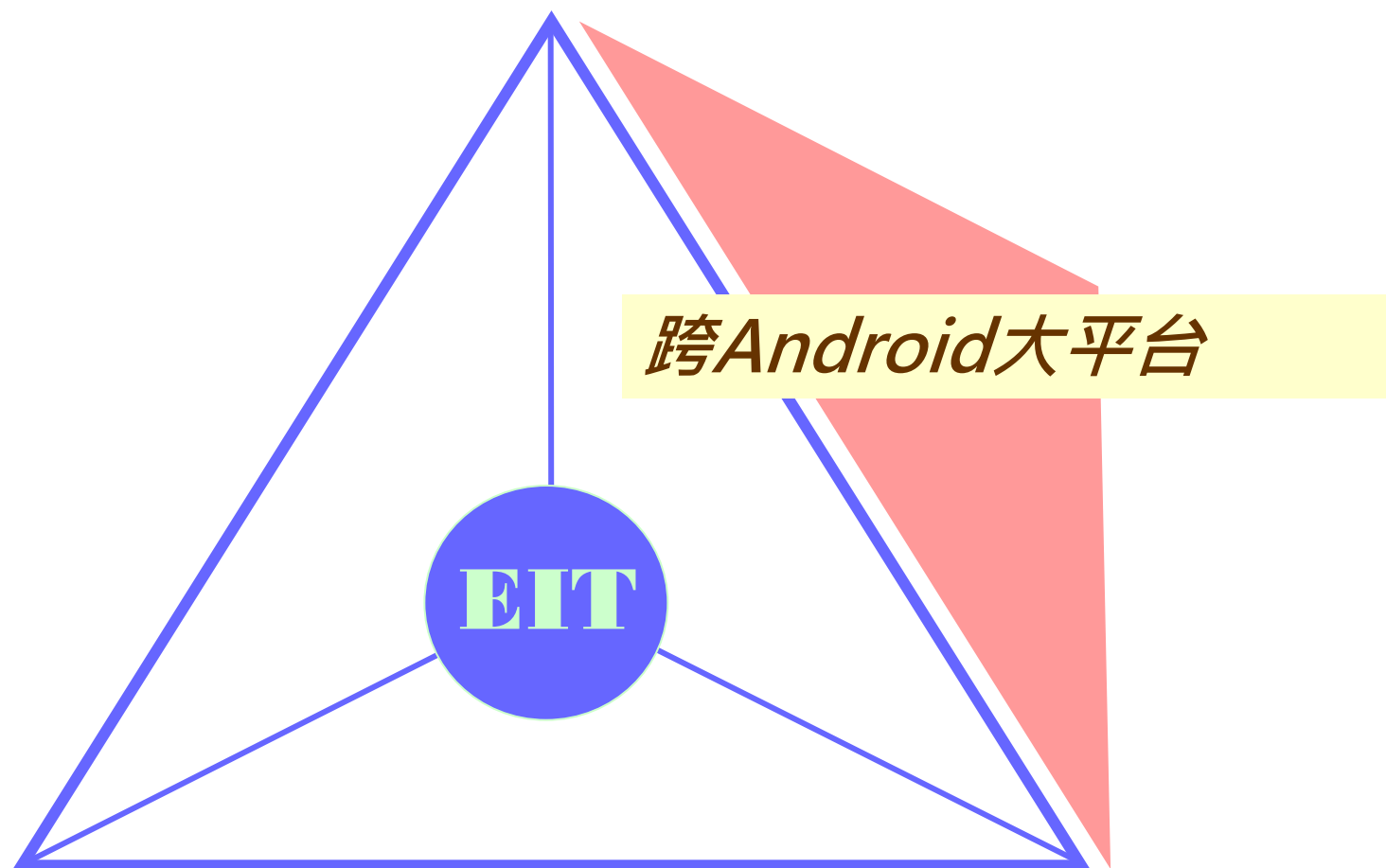
1. 策略-3：建立中间件(middleware)
2. <跨自己平台>的效益：没钱就改版，改版就有钱
3. 中间件跨平台的效益
4. 中间件跨客户端的效益
5. 中间件的设计

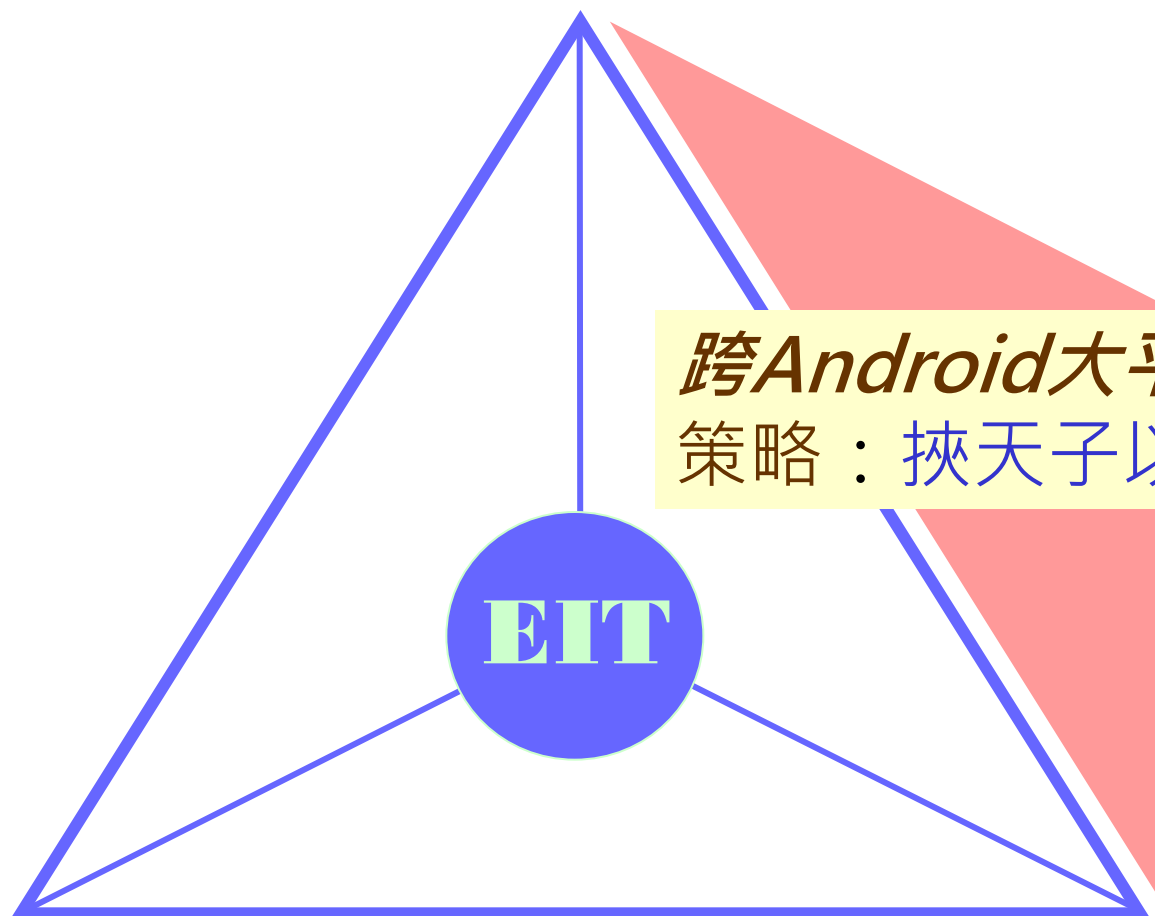
1、策略-3：

建立中间件(middleware)









*跨Android大平台*

策略：挾天子以令諸侯





跨芯片小平台

跨Android大平台

策略：挾天子以令諸侯

EIT



**跨芯片小平台**

策略：把它<EIT>了

**跨Android大平台**

策略：挾天子以令諸侯

**EIT**



**跨芯片小平台**

策略：把它<EIT>了

**跨Android大平台**

策略：挾天子以令諸侯

**EIT**

**跨自己的平台**



**跨芯片小平台**

策略：把它<EIT>了

**跨Android大平台**

策略：挾天子以令諸侯

**EIT**

**跨自己的平台**

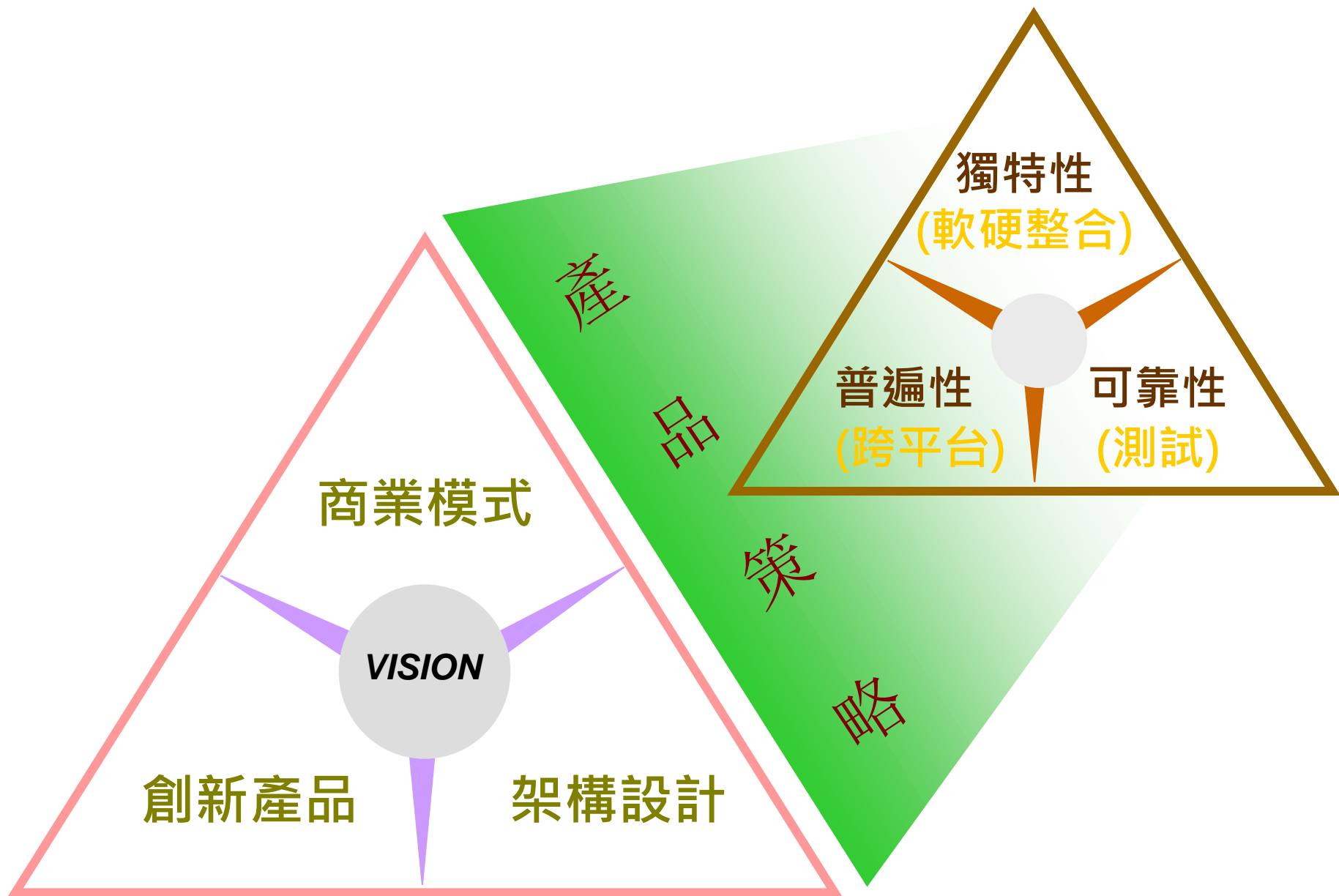
策略：建立中間件

# 建立中间件(middleware)

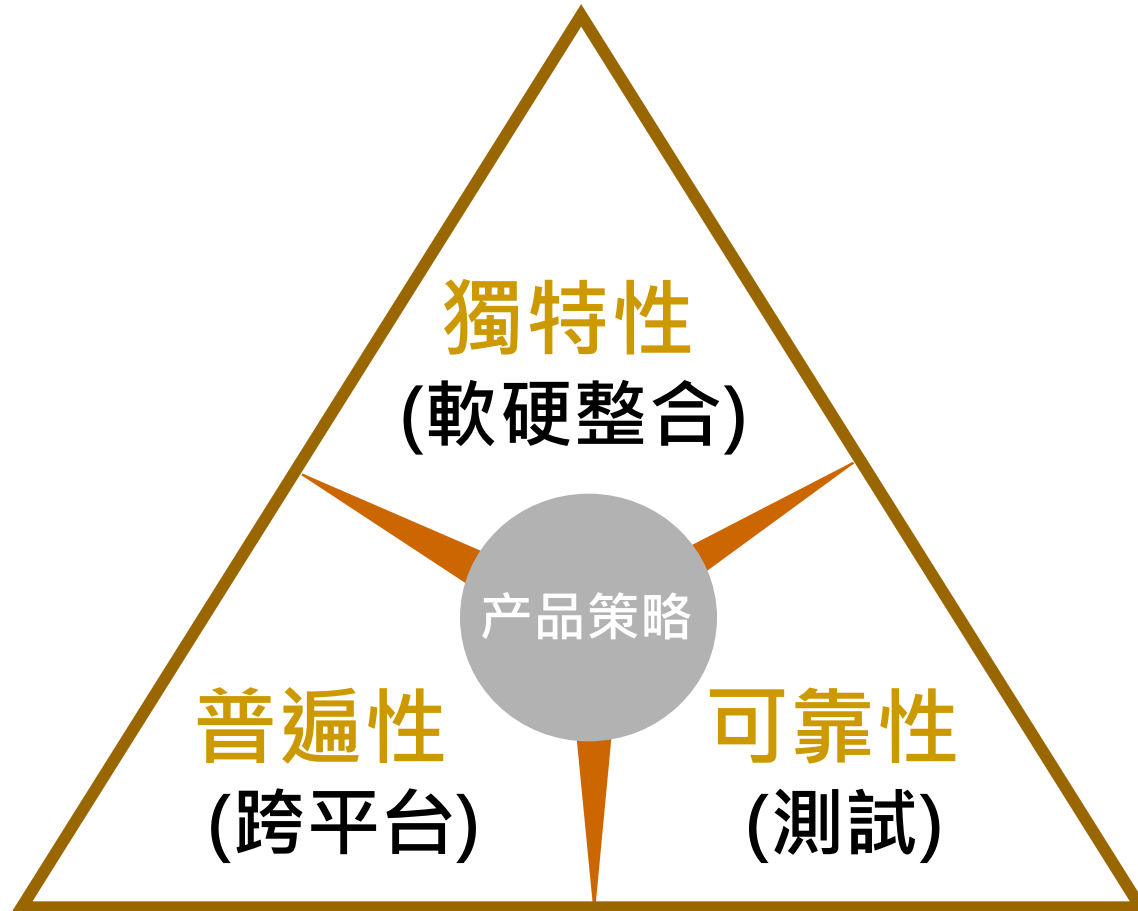
- 策略-1：把它“ EIT(设计)” 了
- 策略-2：挟天子以令诸侯
- 策略-3：**建立中间件(middleware)**

2、这个<策略-3>从哪里来？

反思心中” 不自觉的假设”







- **Assumption** : 只要能跨别人平台，又能跨自己的平台；就能摆脱平台的束缚，而获得产品的普遍性。

- 若<能跨别人平台，又能跨自己的平台>，  
則<能摆脱平台的束缚>。
- 若<能摆脱平台的束缚>，則<能获得产品的普遍性>。

Q

跨芯片(小)平台  
跨Android(大)平台

H

基于EIT造形的策略，  
可实现跨平台

P

基于EIT造形  
的跨平台策略



Q

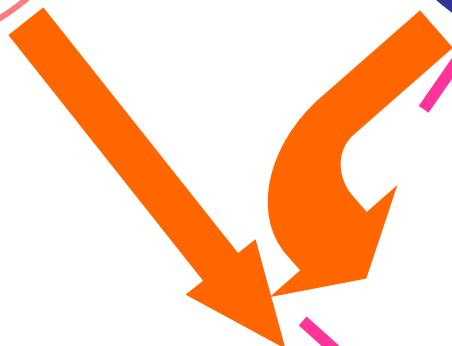
获得产品的普遍性

H

基于EIT造形的策略，  
可实现跨平台

P

基于EIT造形的  
跨平台策略



## H(假定)

假定：若拥有跨平台性，而且能跨客户端；就能拥有普遍性。

Q

获得产品的普遍性

H

若有跨平台性，且跨客户端；就能拥有普遍性。

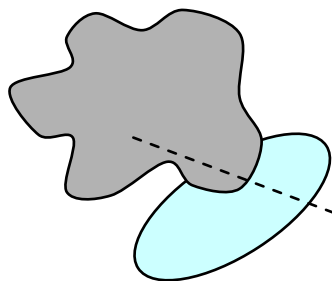
P

取得接口制定权，  
且提供通用性接口。

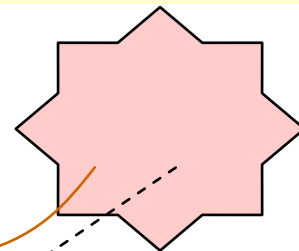
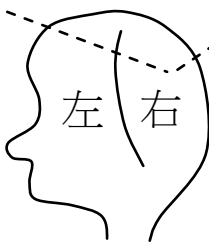
- 引导我们去探索、发现新事实。

拥有跨平台性，  
且跨客户端。

基于EIT造形



提供  
通用性接口

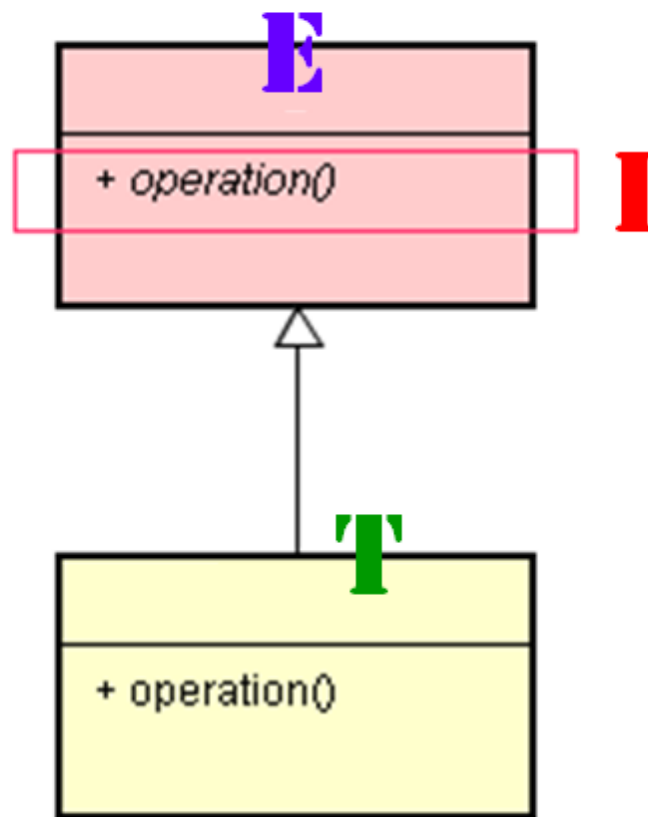




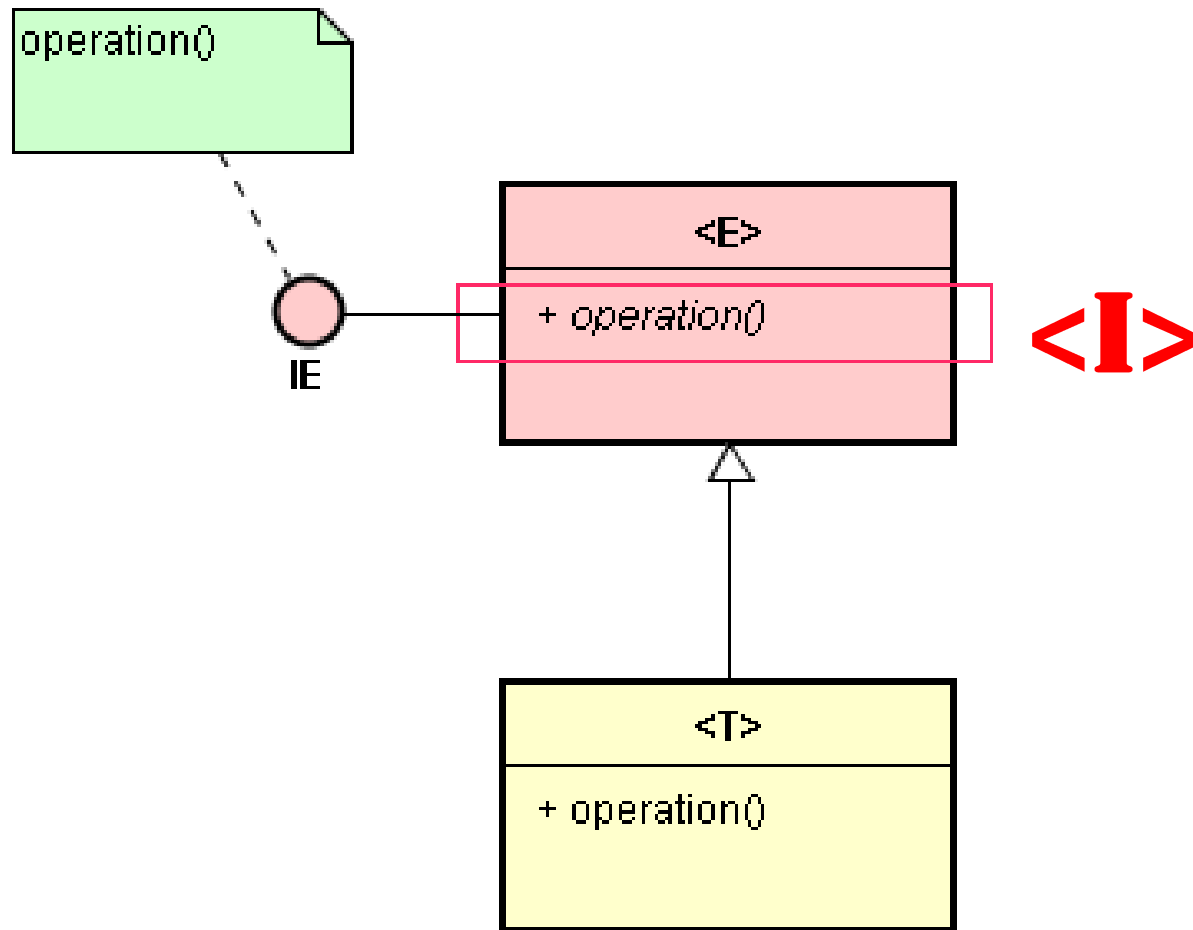
# 探索&学习未知事物

通用性接口的基本结构

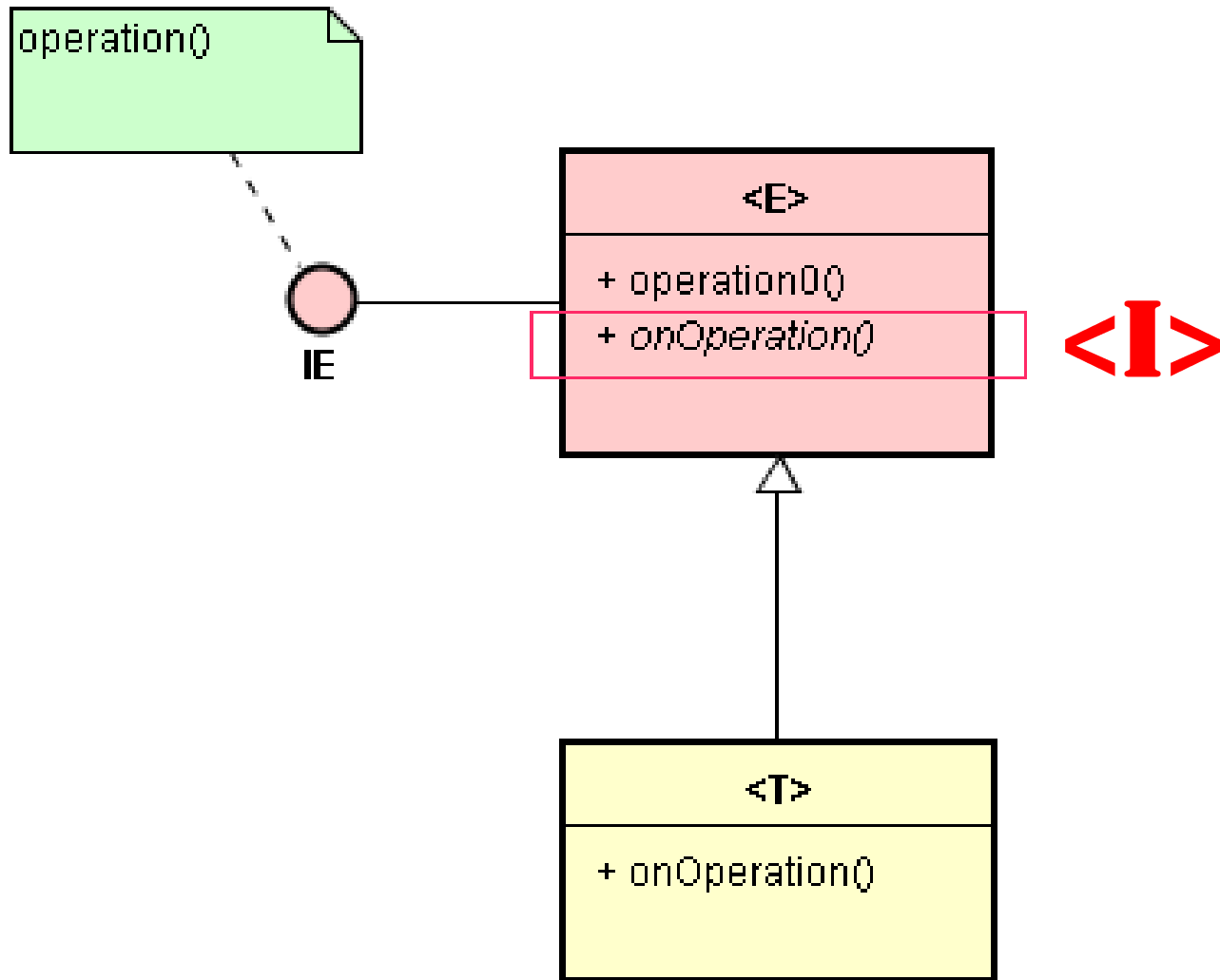
- 兹看看一个大家已经很熟悉的EIT造形：



- 这个<E>可以提供提供一个接口，如下图的IE：

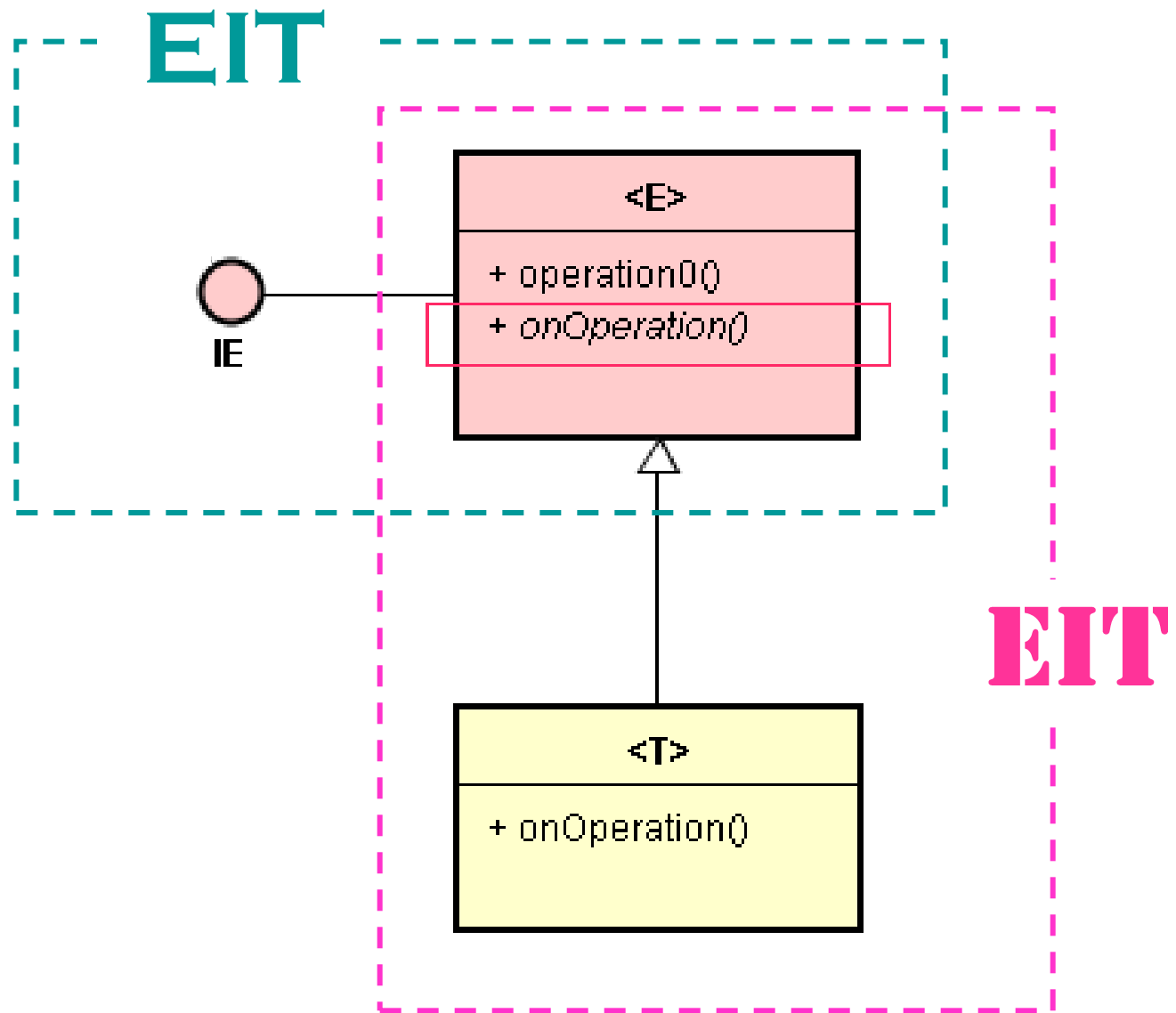


- <E> 提供了IE接口给Client使用；于是Client透过IE接口而调用<E>的operation()函数，由于它是一个抽象函数，所以就直接执行了<T>的operation()函数的实现代码。这里的IE接口与<I>接口是一致的。
- 它有一个更常见的变形：



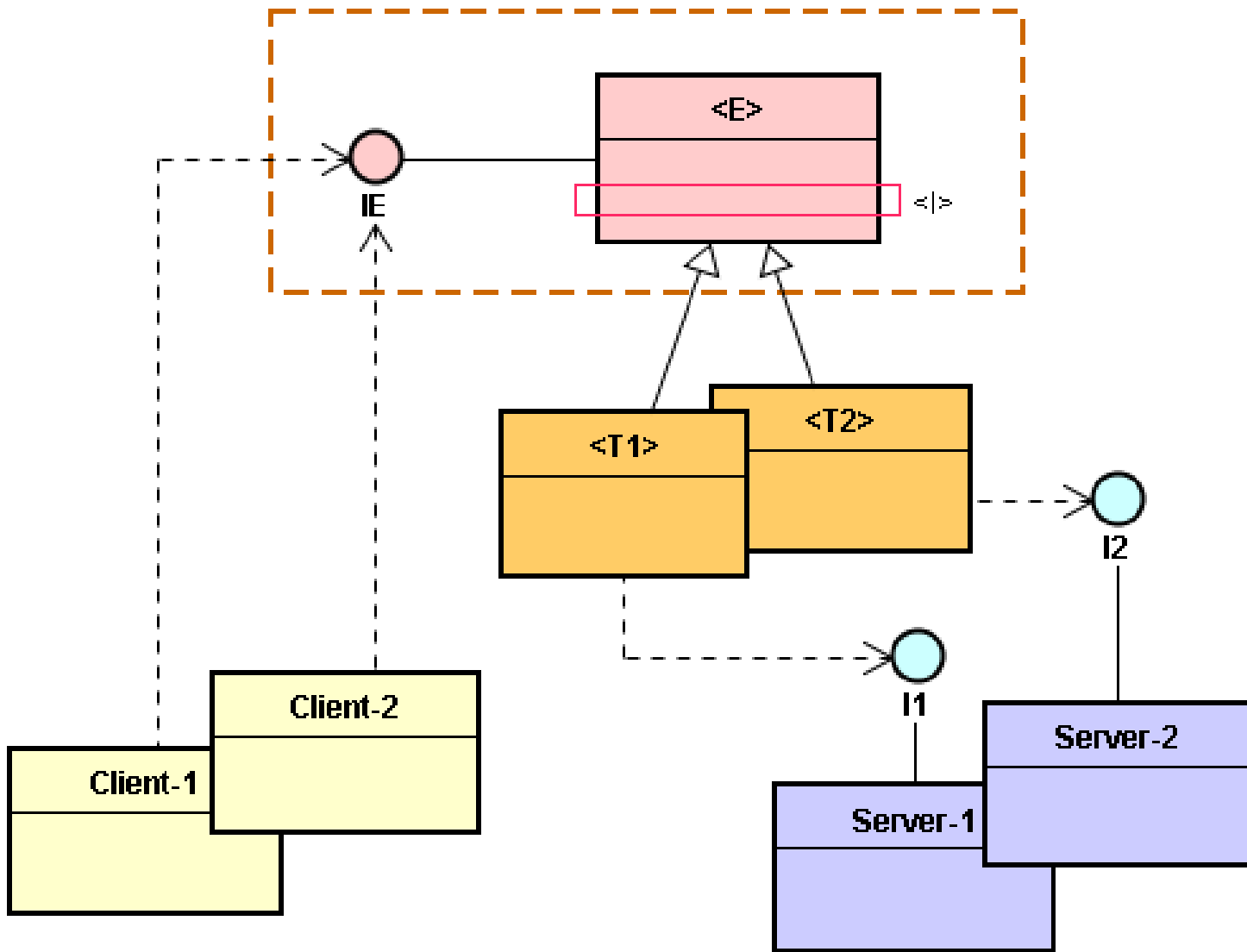
- 其中的operation()函数将IE接口转换成为<I>接口；于是Client透过IE接口而调用<E>的operation()具象函数，然后调用onOperation()抽象函数，就执行了<T>的onOperation()函数的实现代码。这里的IE接口与<I>接口是不一样的。

- 其实，上述这是接口设计模式，是由两个EIT造形所组成的，所以我们称之为“模式” (Pattern)，而不称为“造形” (Form)。

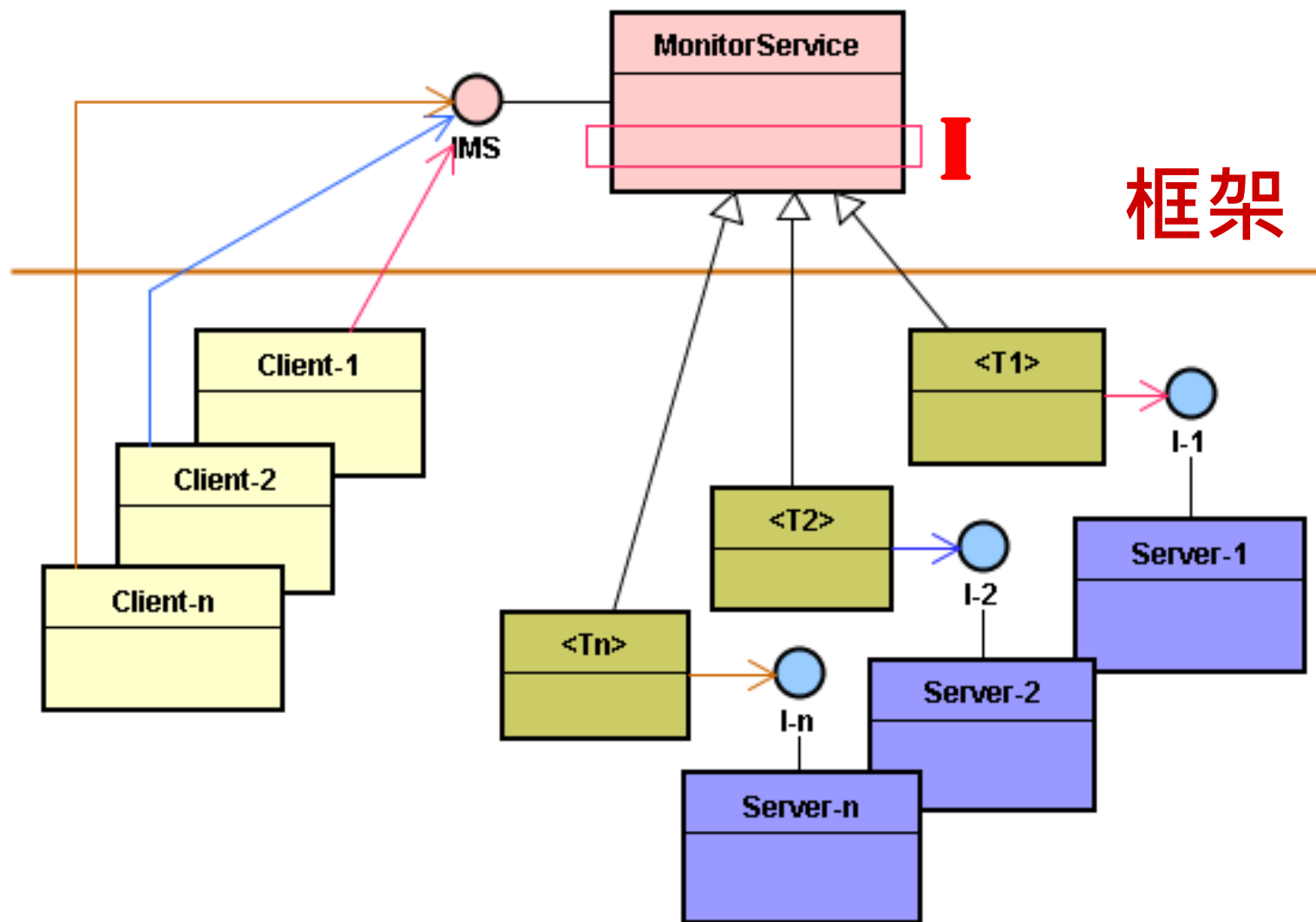




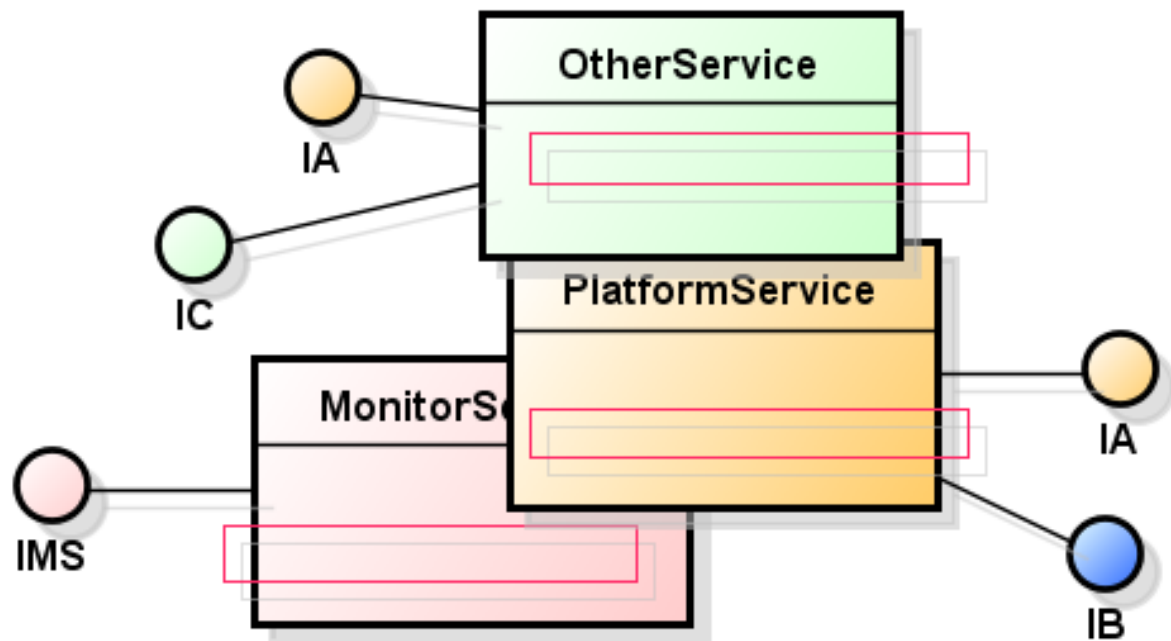
- 上面的EIT造形，是一个退化的EIT造形，没有限定它的<E>；也可以说，所有的Client都能扮演它的<E>角色，而是这个IE接口就成为“通用性接口”了。



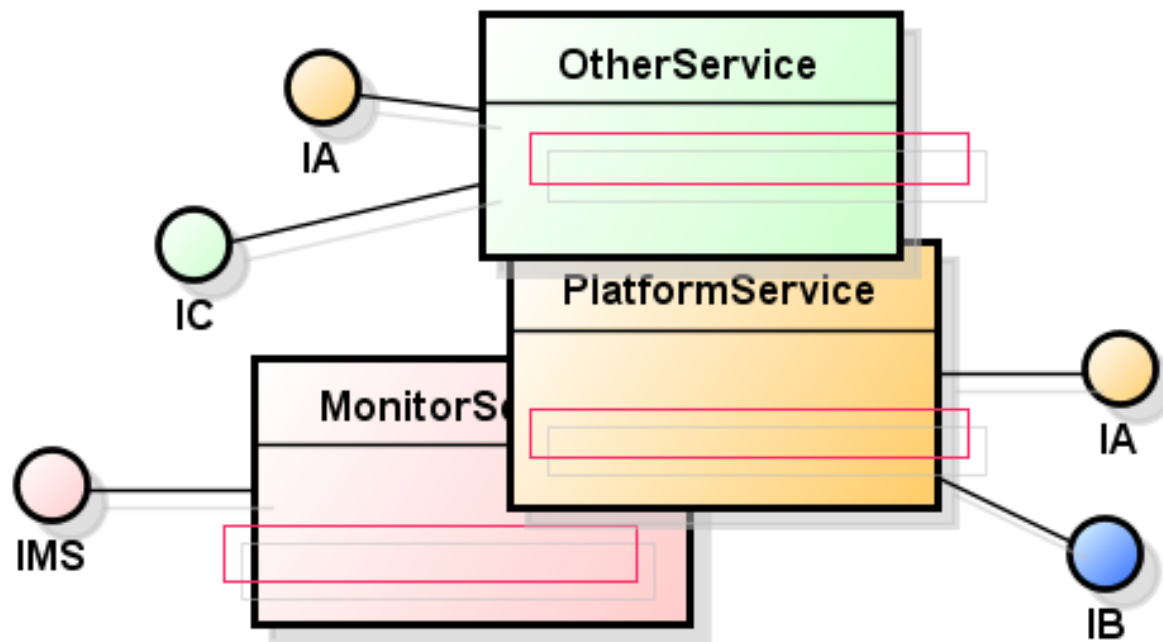
- 其实，通用性接口常常示成双成对的，例如下图里的<IMS>是对 Client端的通用性接口；此外，还常常设计一个对 Server端的通用性接口，它就是大家熟悉的EIT造形里的<I>了。



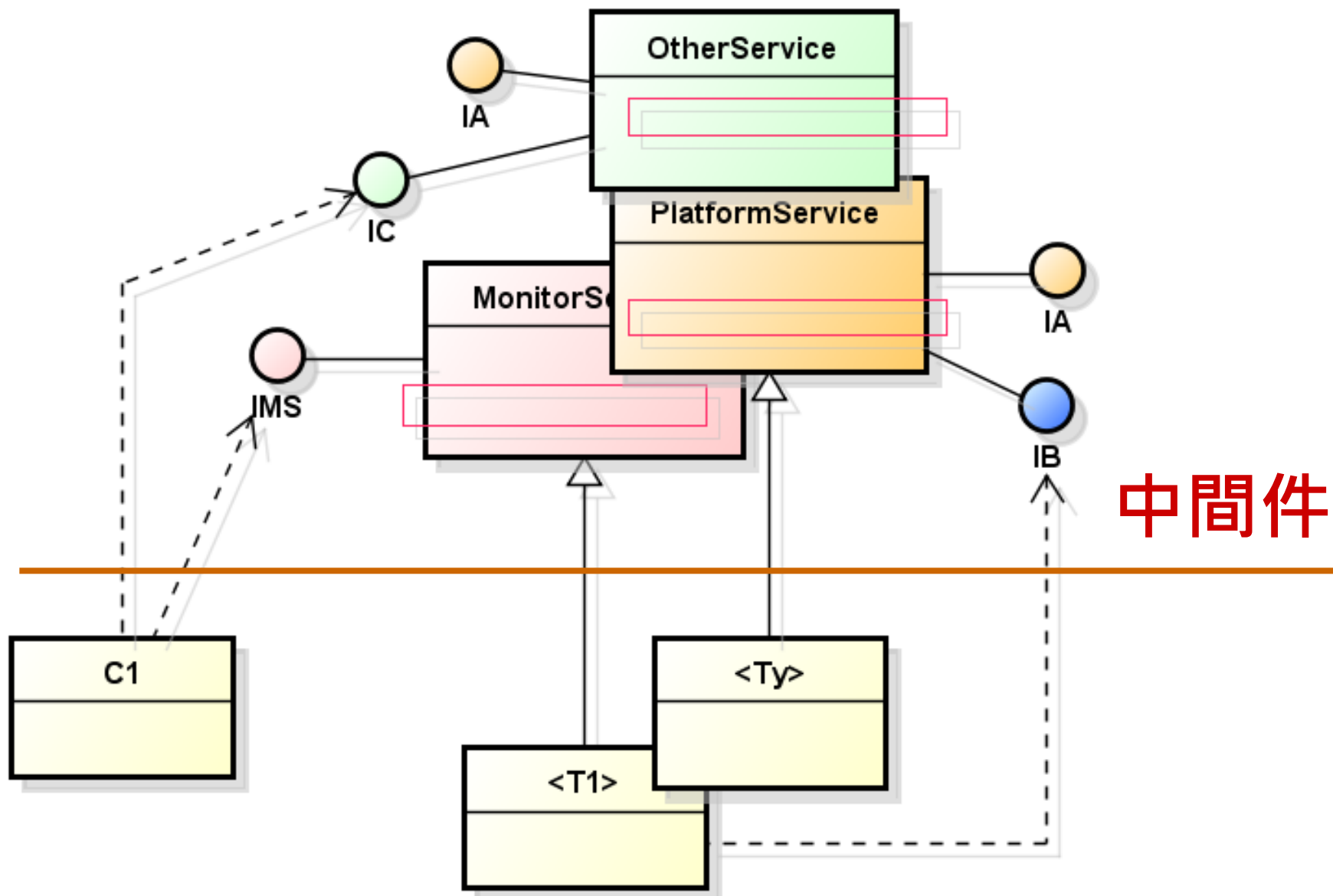
- 随着系统的成长，会设计出更多的通用性接口，就成为一个框架，或俗称的“中间件”。



框架



中間件





- 其实，通用性接口常常示成双成对的，其中一个是对 Client端的通用性接口，让我们的产品能跨客户端。另一个是对 Server端的通用性接口，它就是大家熟悉的EIT造形里的<I>了，让我们的产品能跨平台。

随着对未知事物的探索和学习，  
新事实显示支持我们的假定(H)。

Q

获得产品的普遍性

H

若有跨平台性，且跨客户端；就能拥有普遍性。

P

取得接口制定权，  
且提供通用性接口。

- 于是，我们继续探索这个<实践策略-3>。



~ Continued ~