

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>

I04

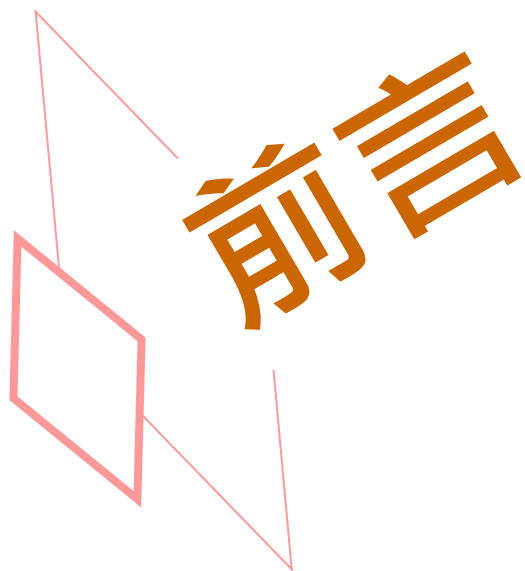
跨(芯片)小平台策略(c)

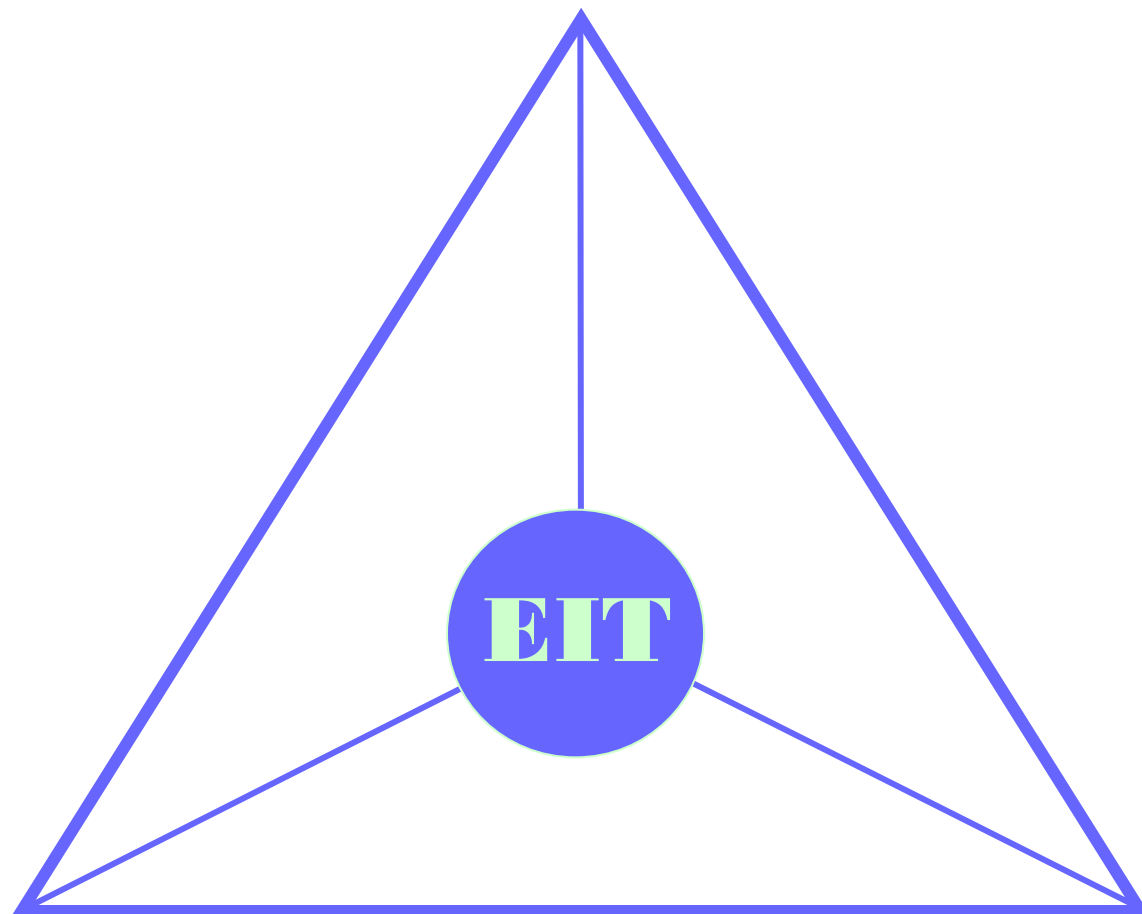
By 高煥堂

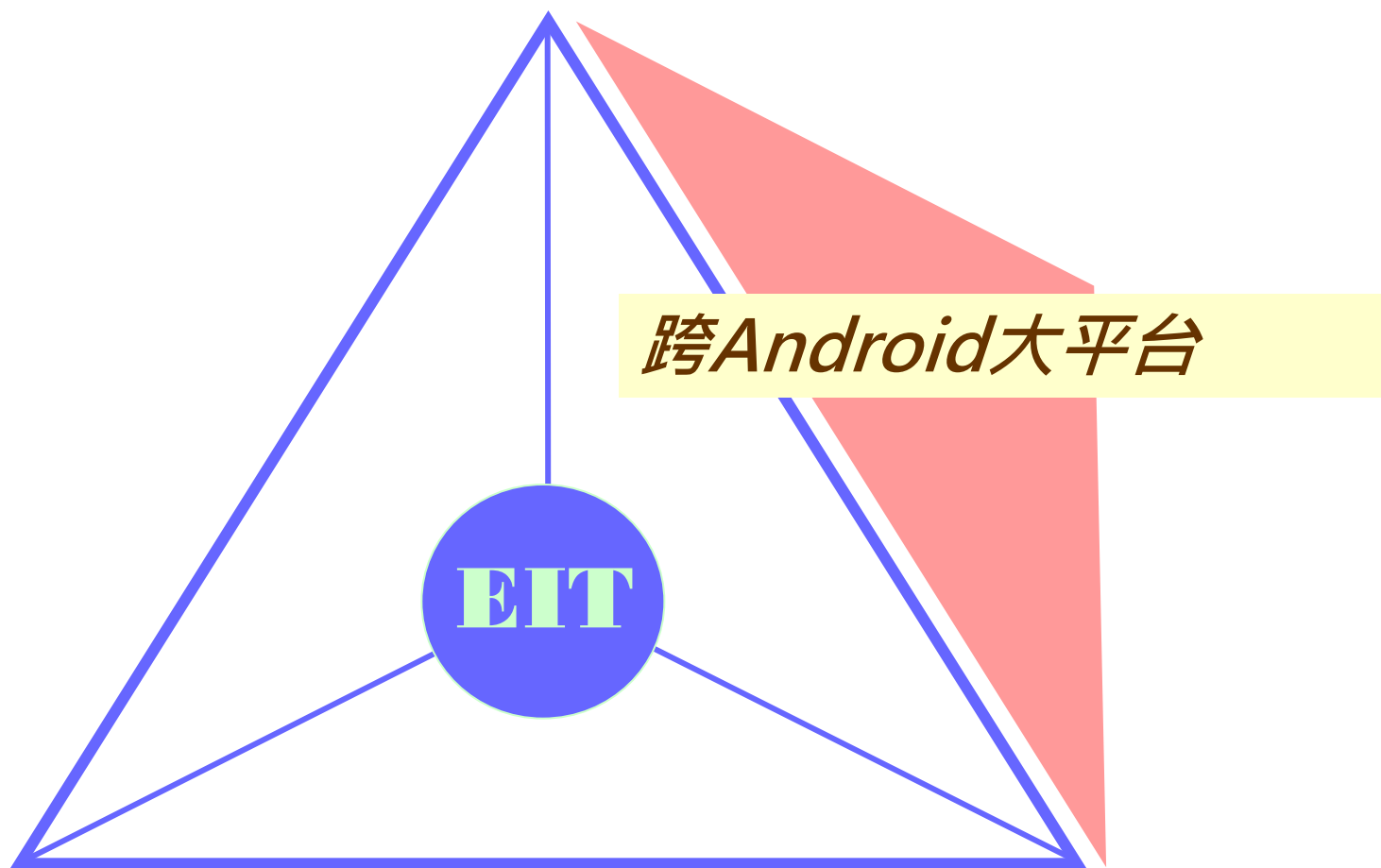
内容

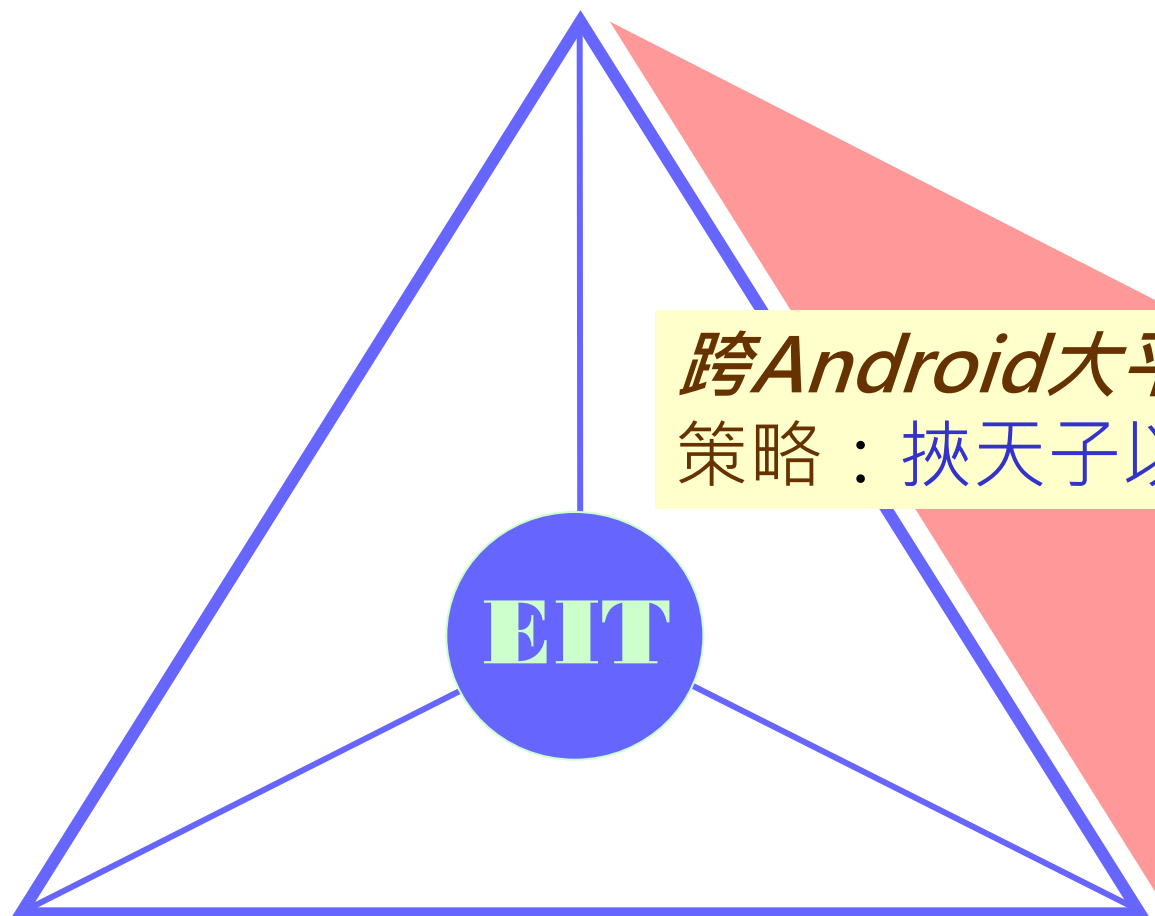
1. 策略-1：把它“ EIT(设计)” 了
2. 跨(芯片)小平台的 3种情境
3. 结语

1、策略-1：
把它“ EIT(设计)” 了









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

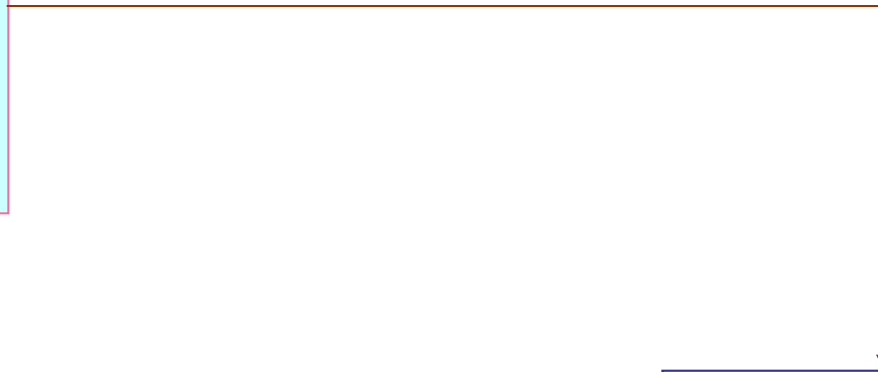
策略：建立中間件

把它” EIT(设计)” 了

- 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)

<車輛>

<地面>

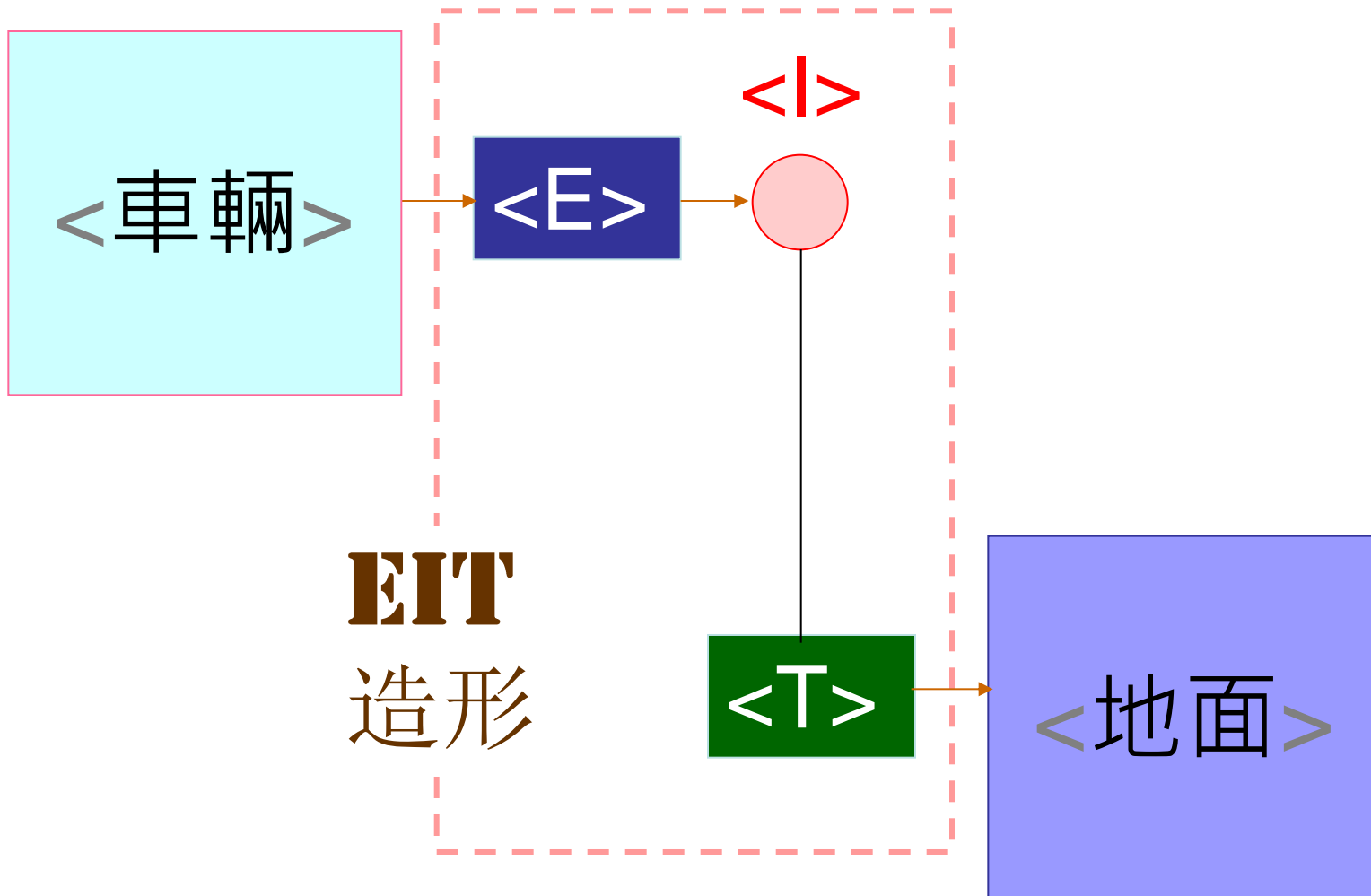


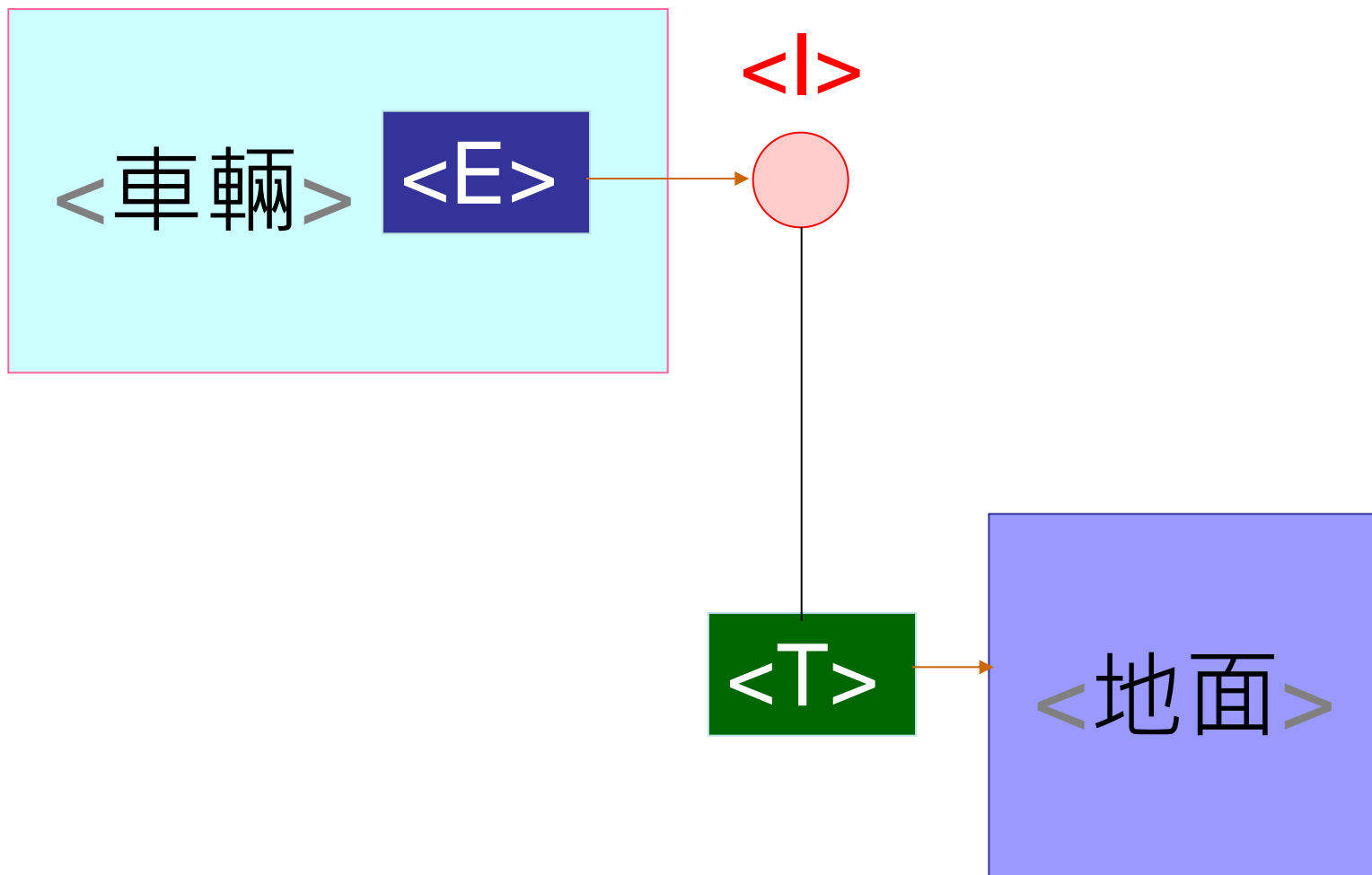
<車輛>

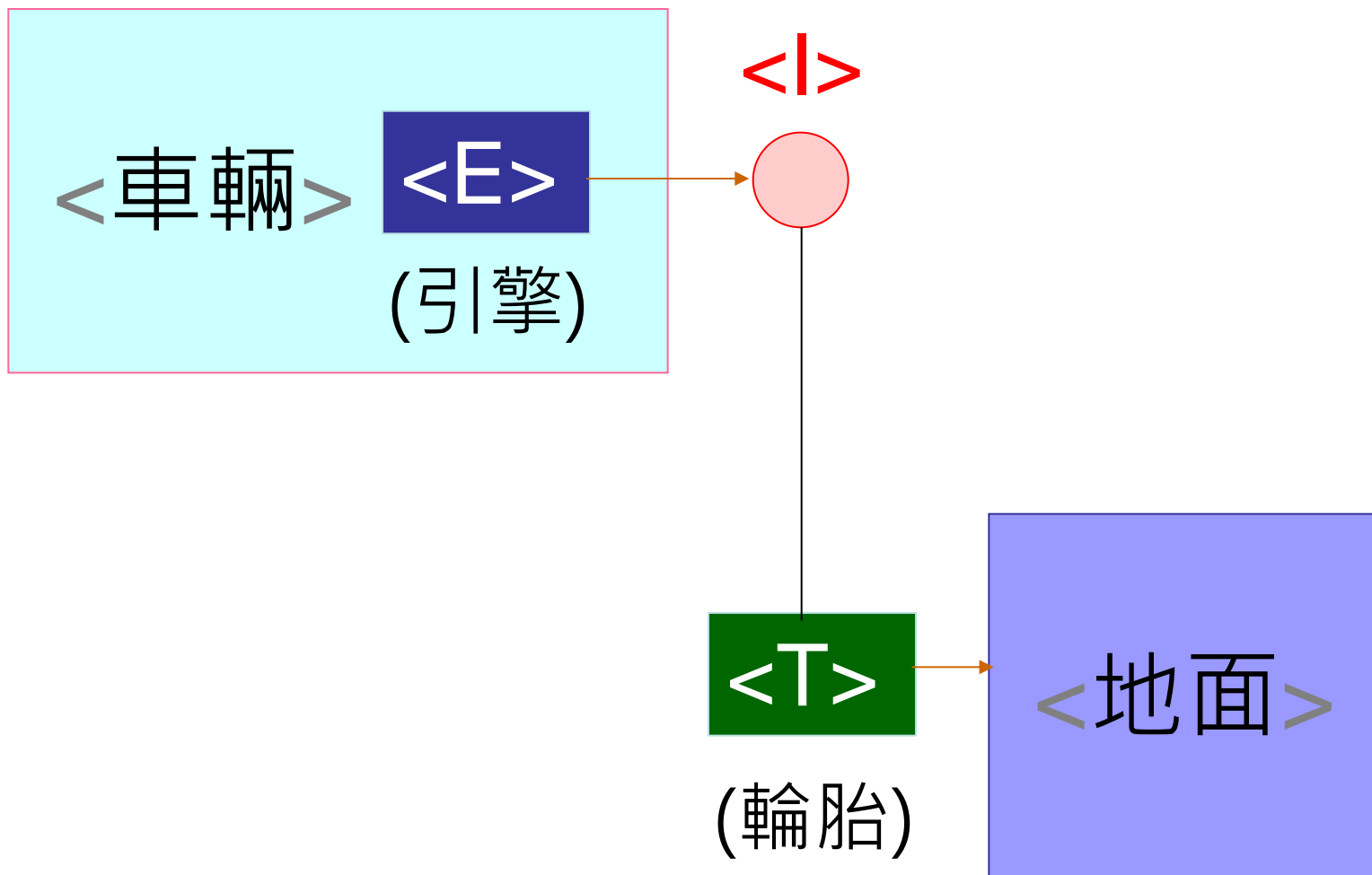
EIT
造形

<地面>





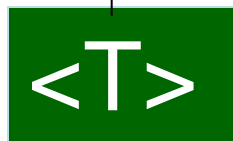
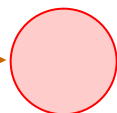






我的Module

<|>



別人的平台



<E>



<I>

接口



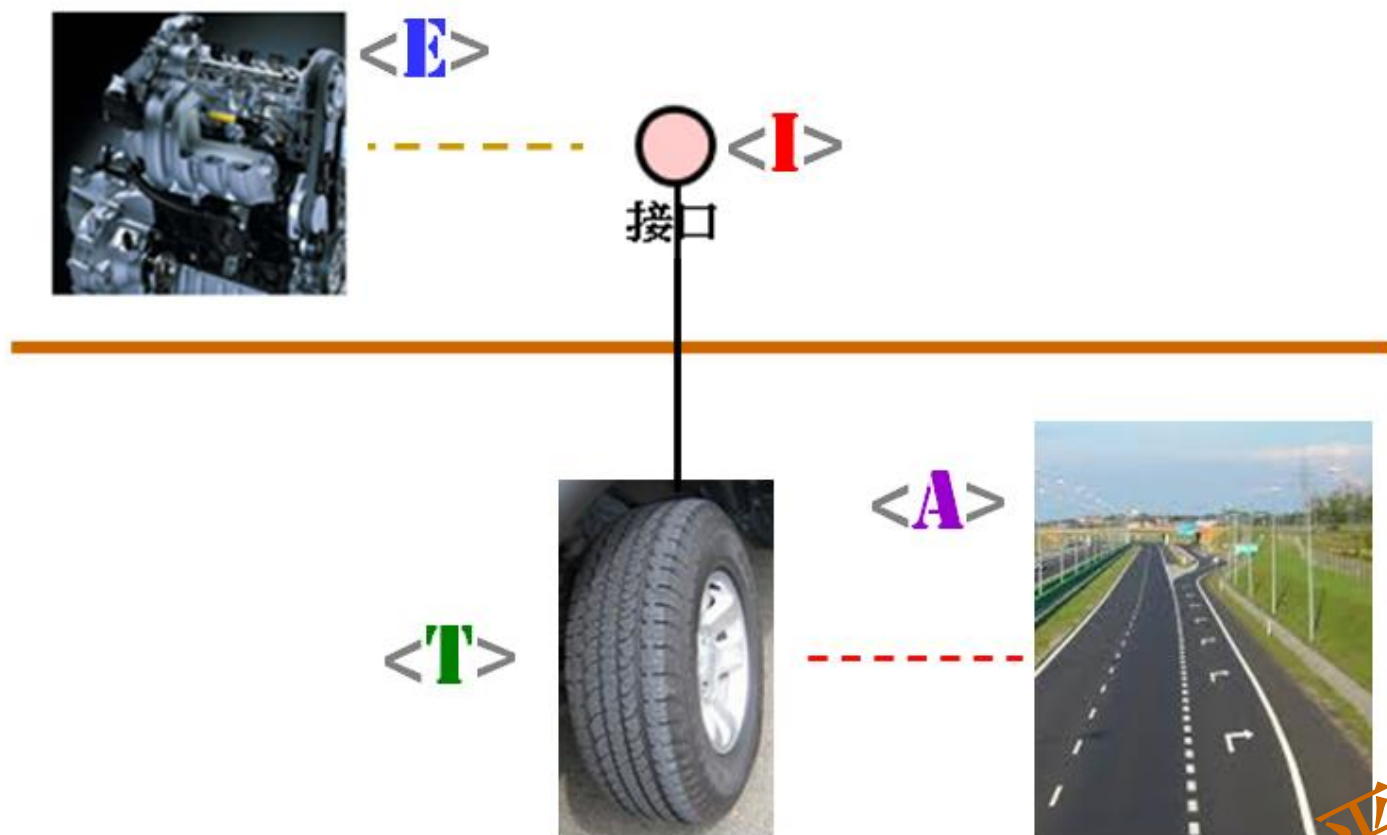
<T>

<A>

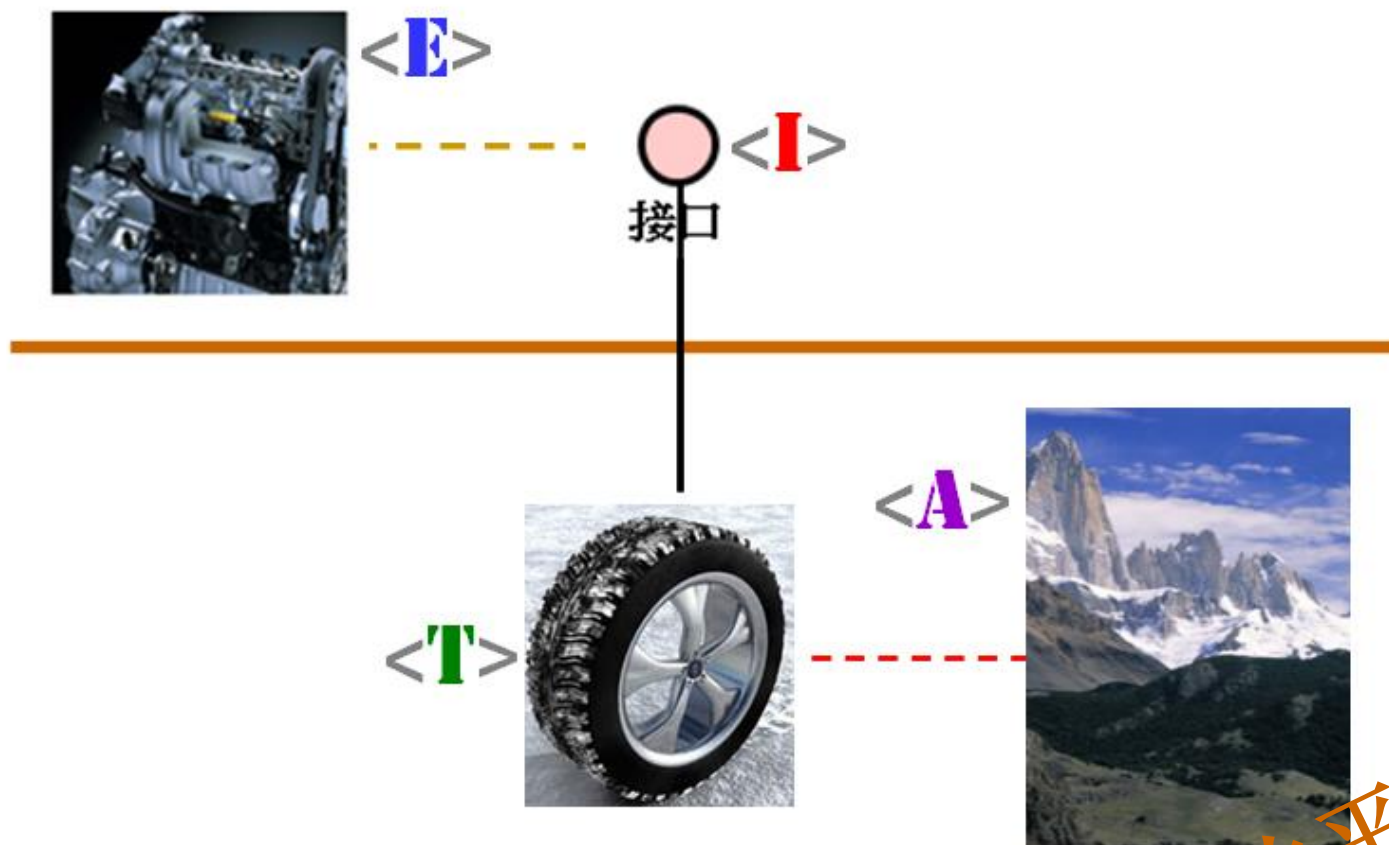


沙灘

別人的平台



別人的平台



別人的平台

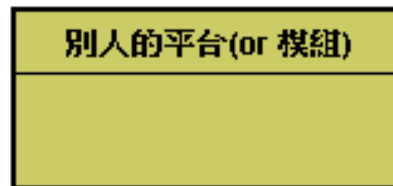
2、跨(芯片)小平台的 的 3种情境

情境-A

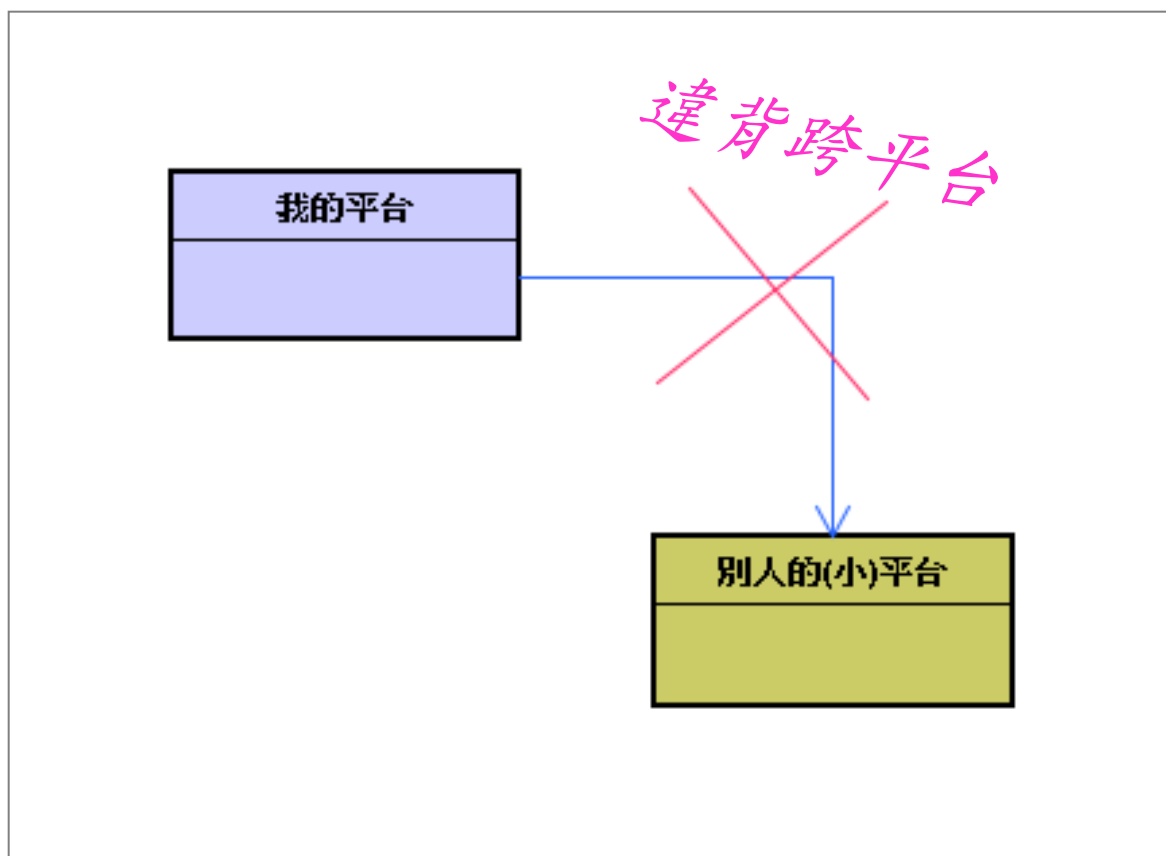
*先有别人的(小)平台，
然后才建立我的平台*

- ◎ 小平台是指别人的平台，该平台的变化决定于别人。例如，由供货商提供一个平台，如下图：

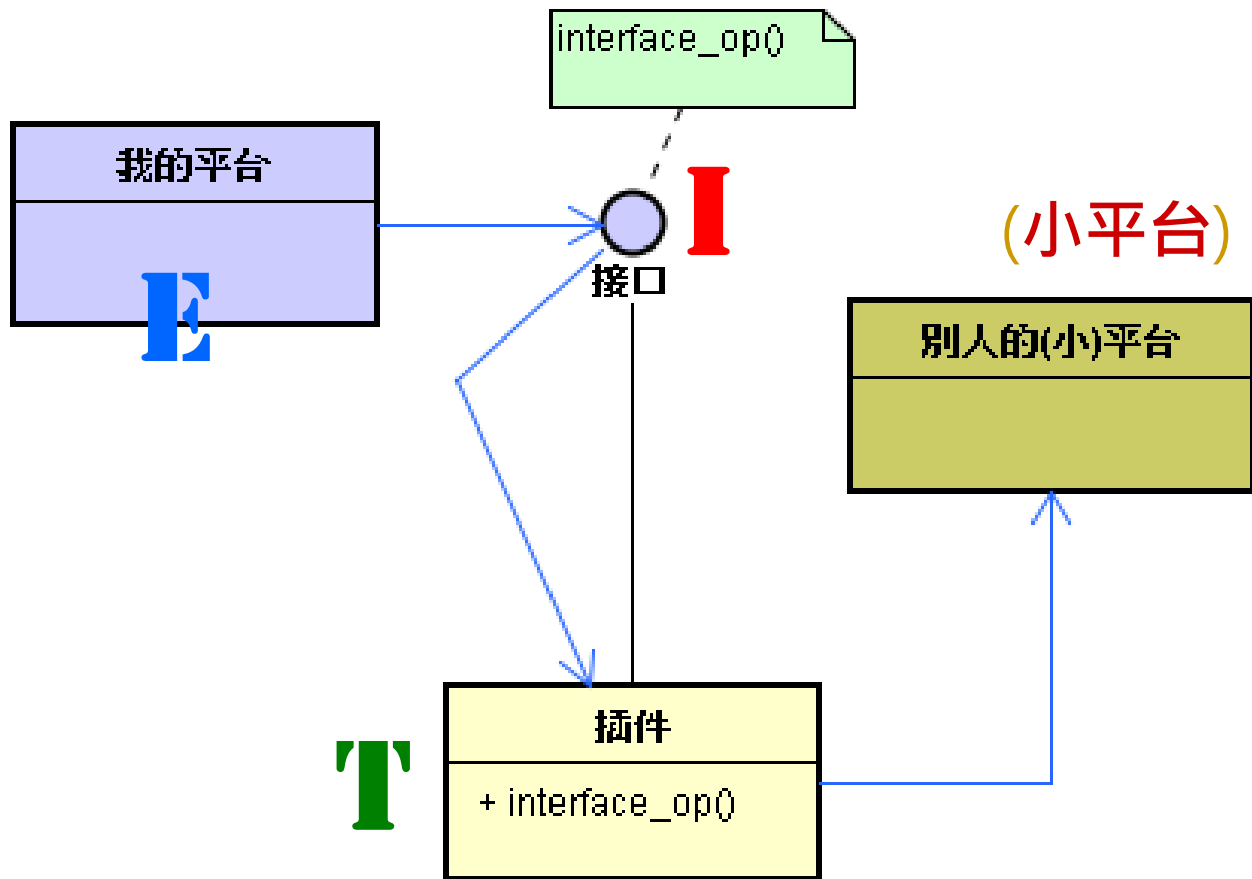
通常是供應商提供的

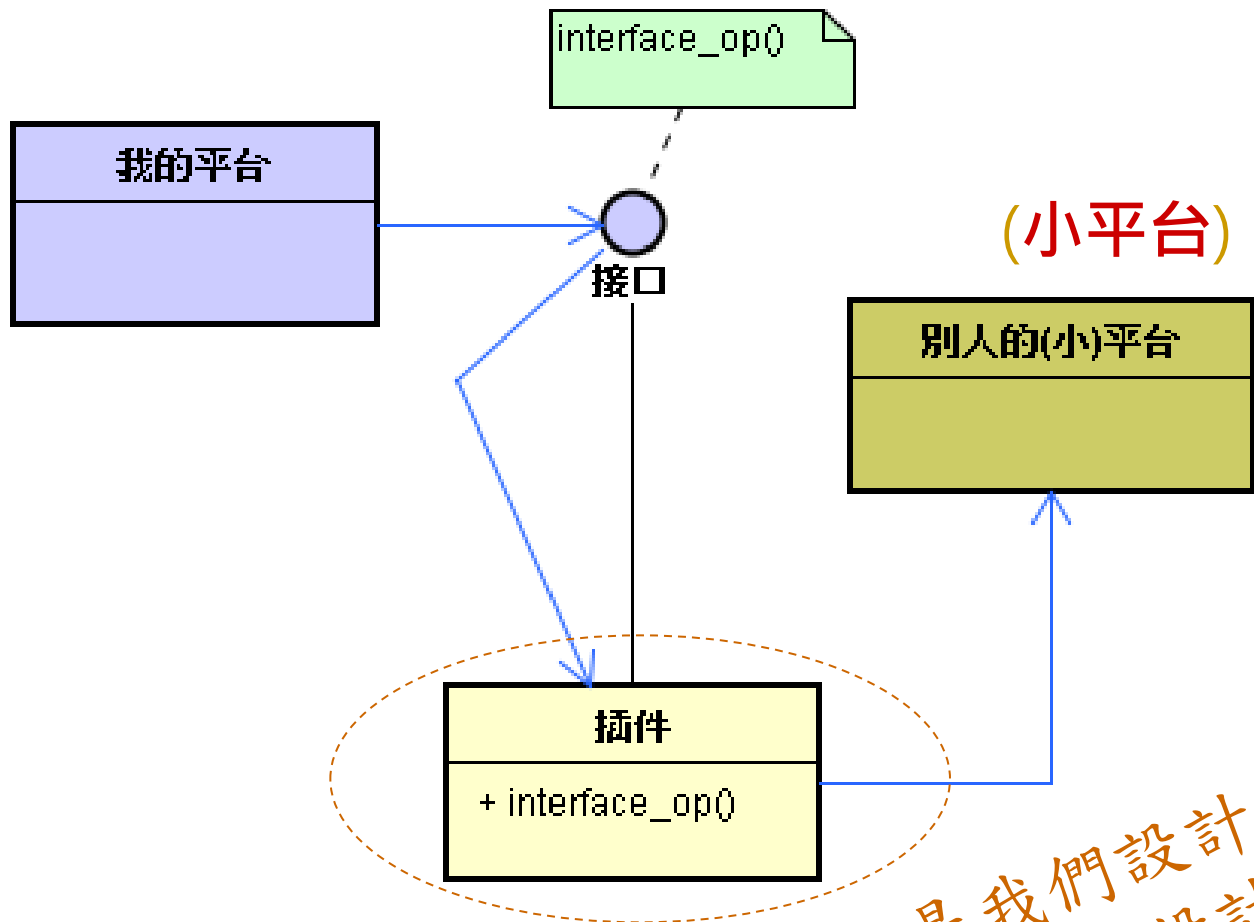


- ◎ 我们(自己)的平台或模块，直接使用了别人平台的API，如下图：



- 您设计<E&I>，而且设计<T>来包容别人平台的变化，这就称为：把它“ EIT(设计)”了。



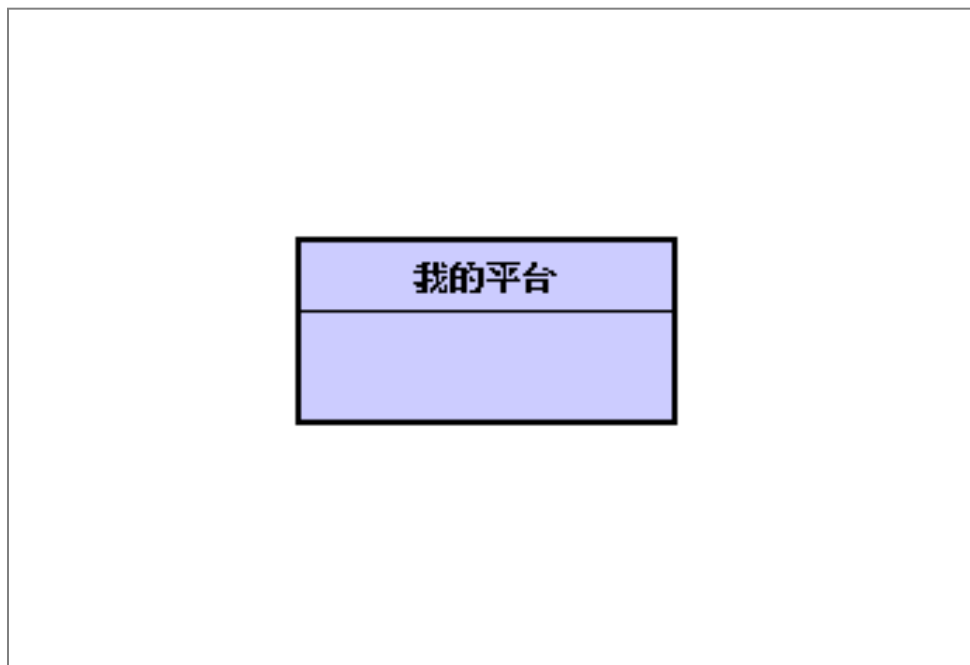


插件是我們設計的，
或是第三方設計的。

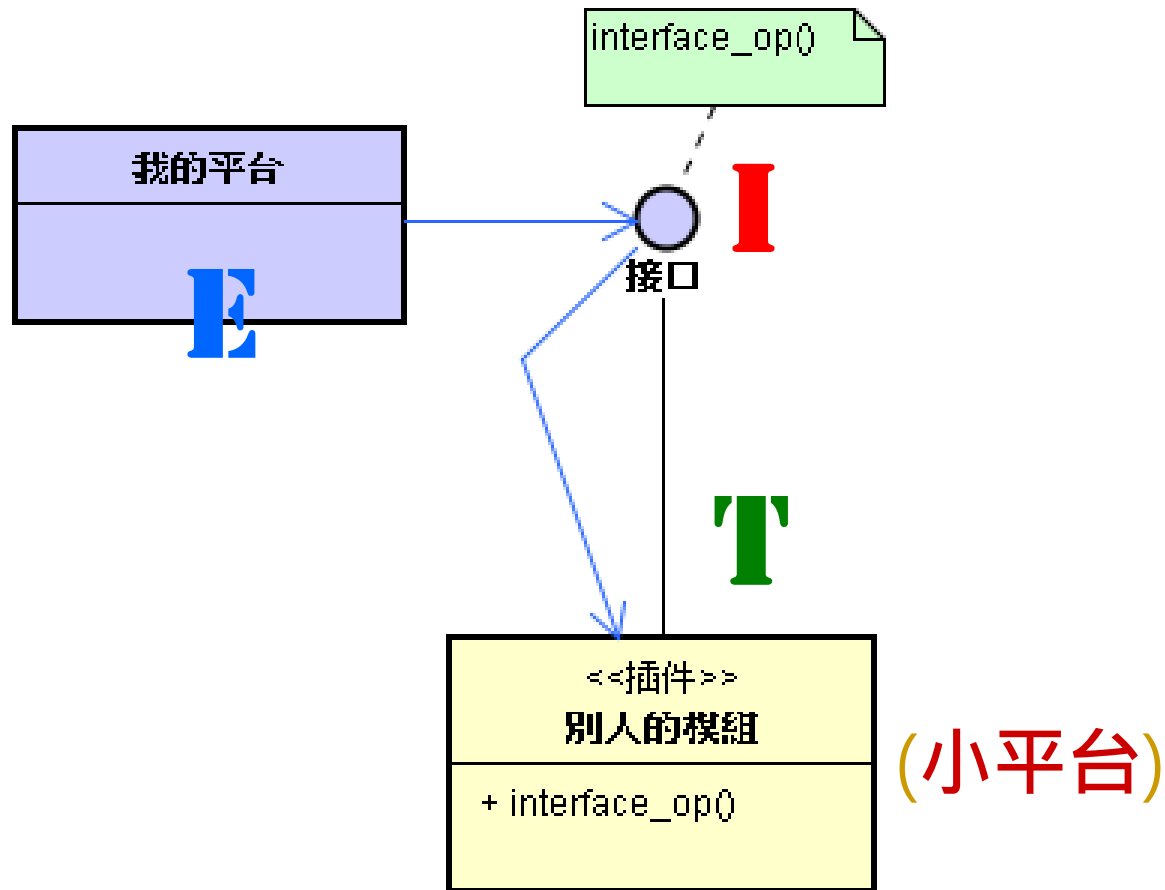
情境-B

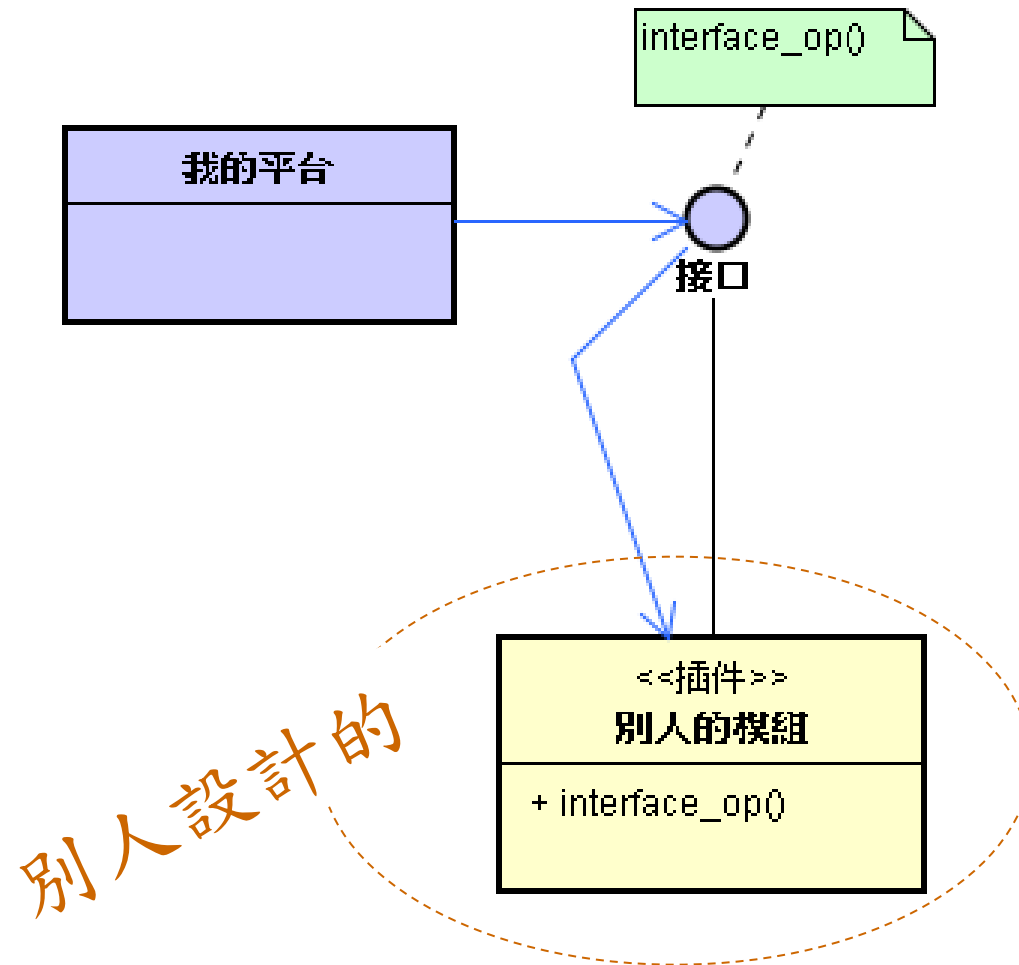
先建立我的平台，
然后才让别人来扩充(*Extend*)

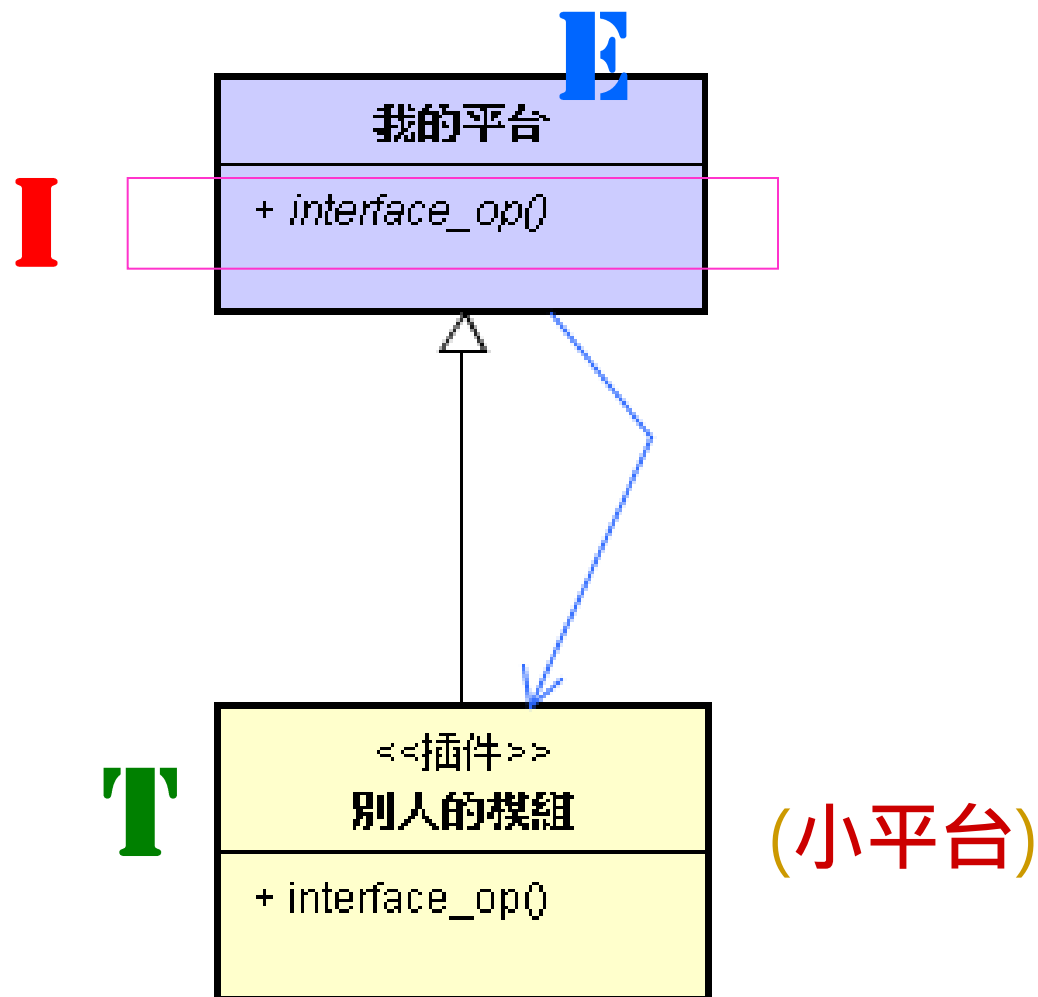
◎我的平台(E&I)已经先存在了。



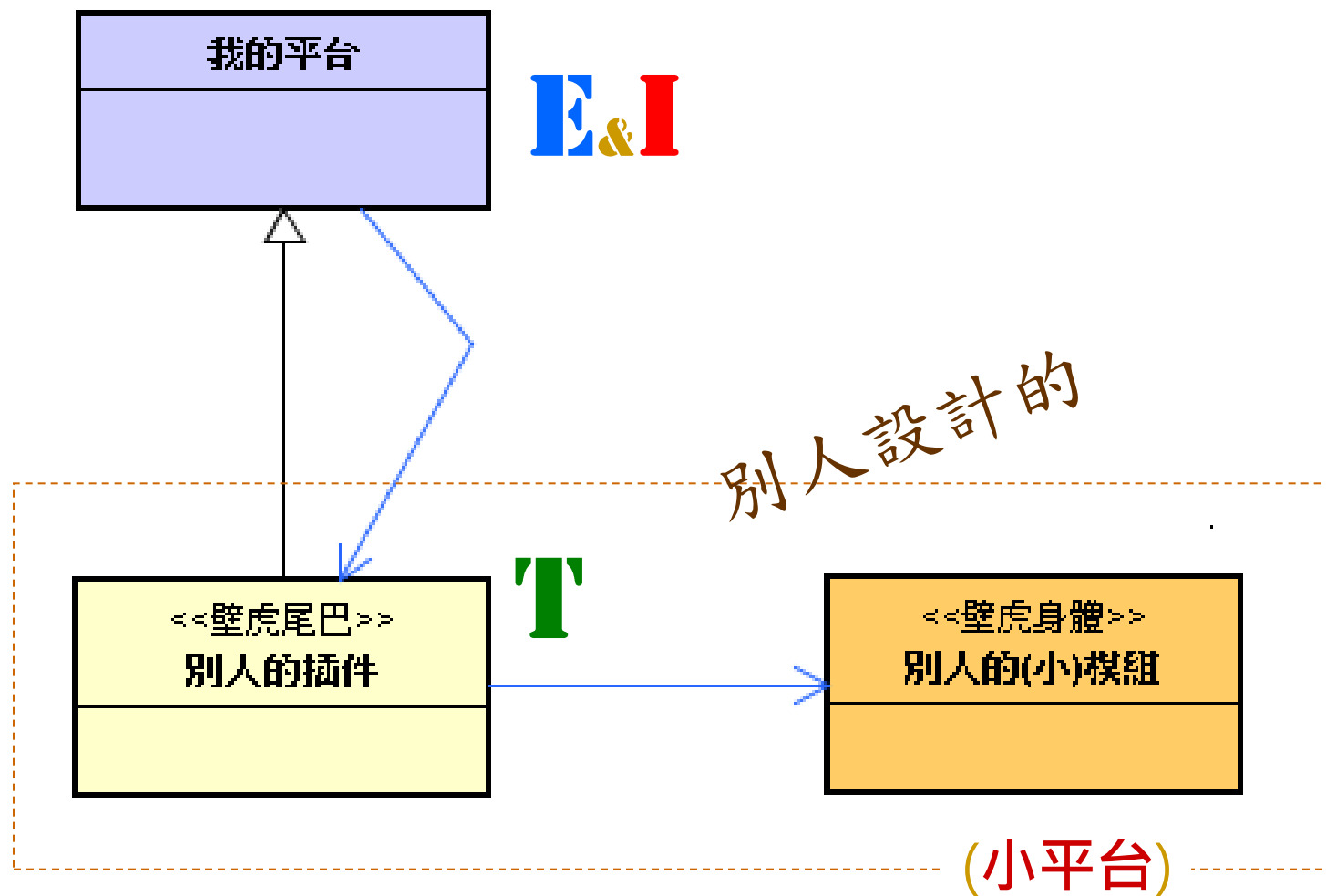
- ◎ 让别人设计插件<T>来扩充(extend)您的<E&I>。





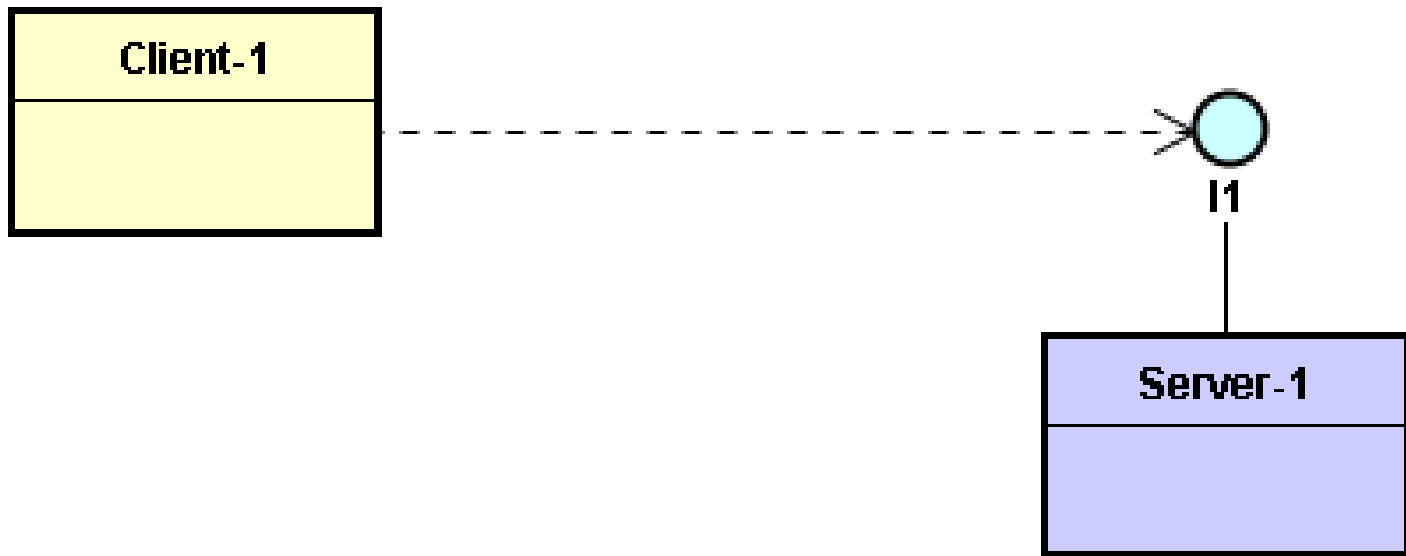


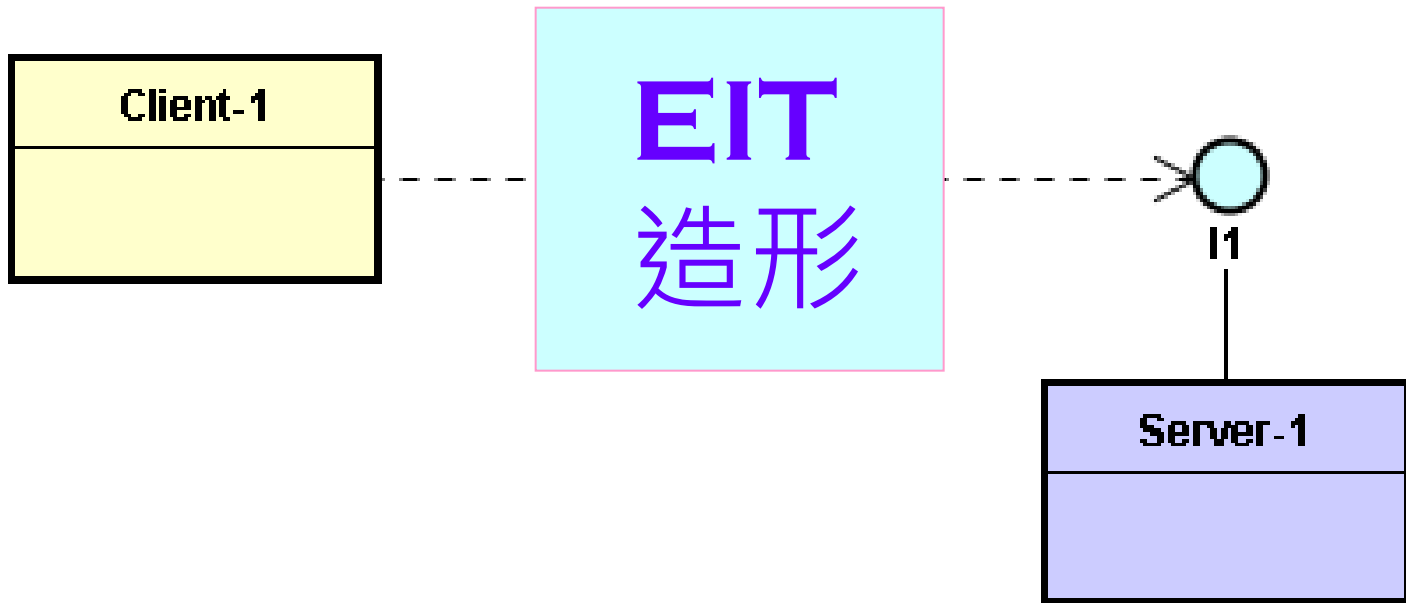
- 别人为了保护他自己，也会将插件分成两部分：<壁虎尾巴>与<壁虎身体>
- 万一您的<E&I>有变化时，这只壁虎(插件)便能弃尾求生，让<壁虎身体>跨您的<E&I>

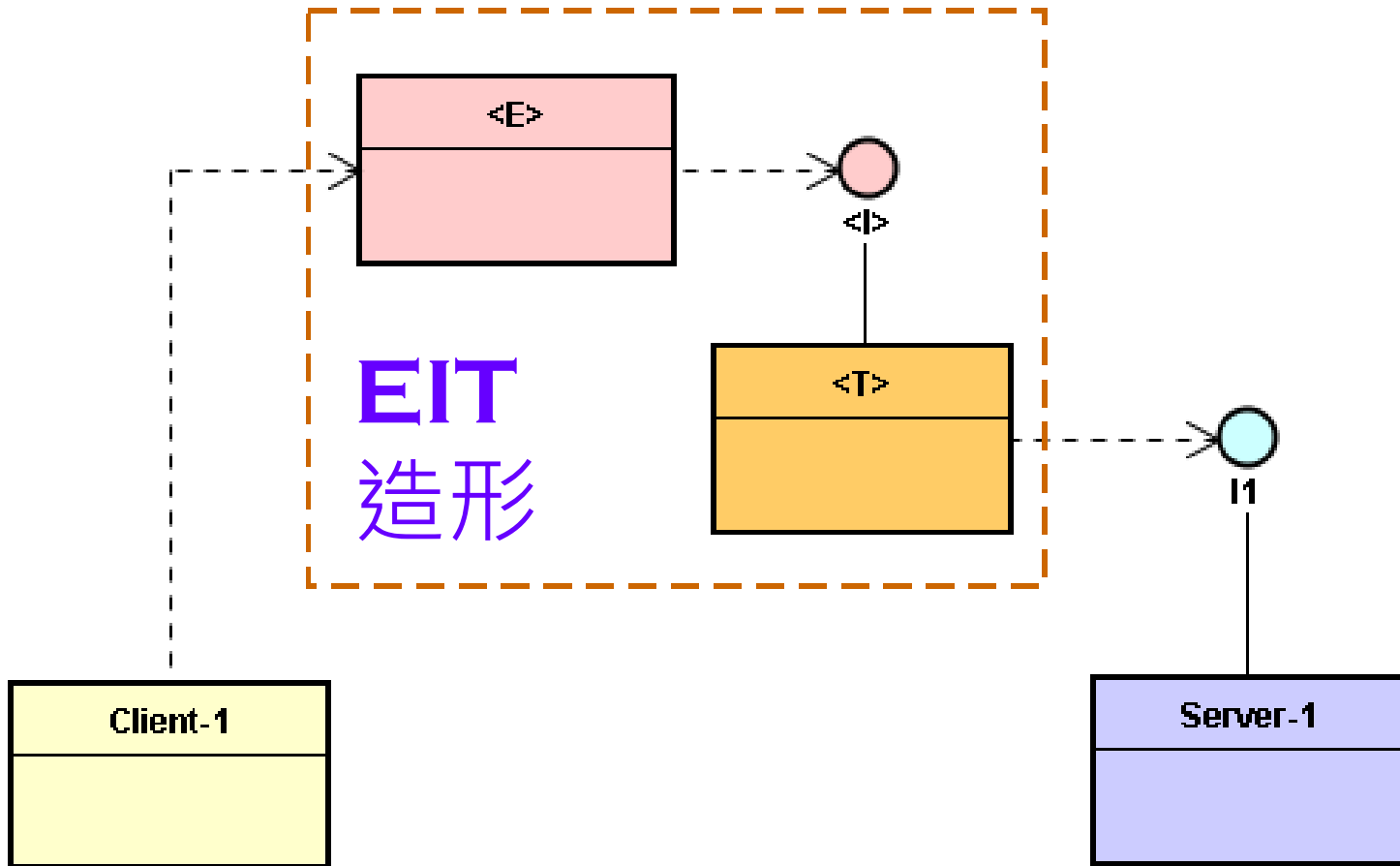


情境-C

*切入别人的Client-Server之间，
建立我的平台*

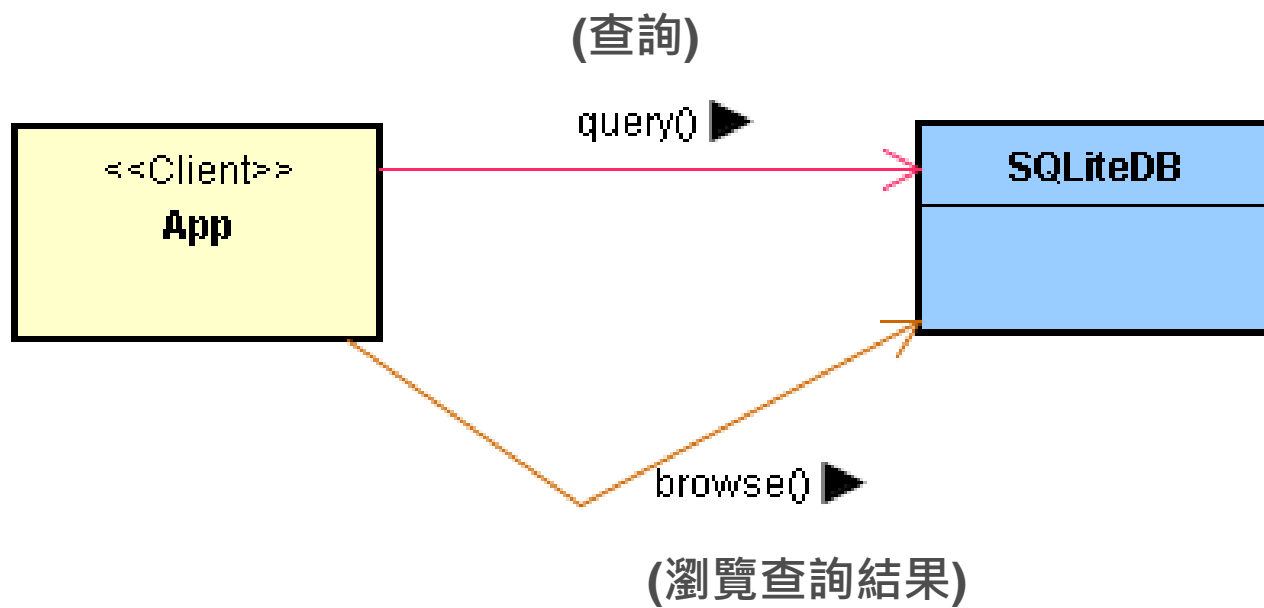


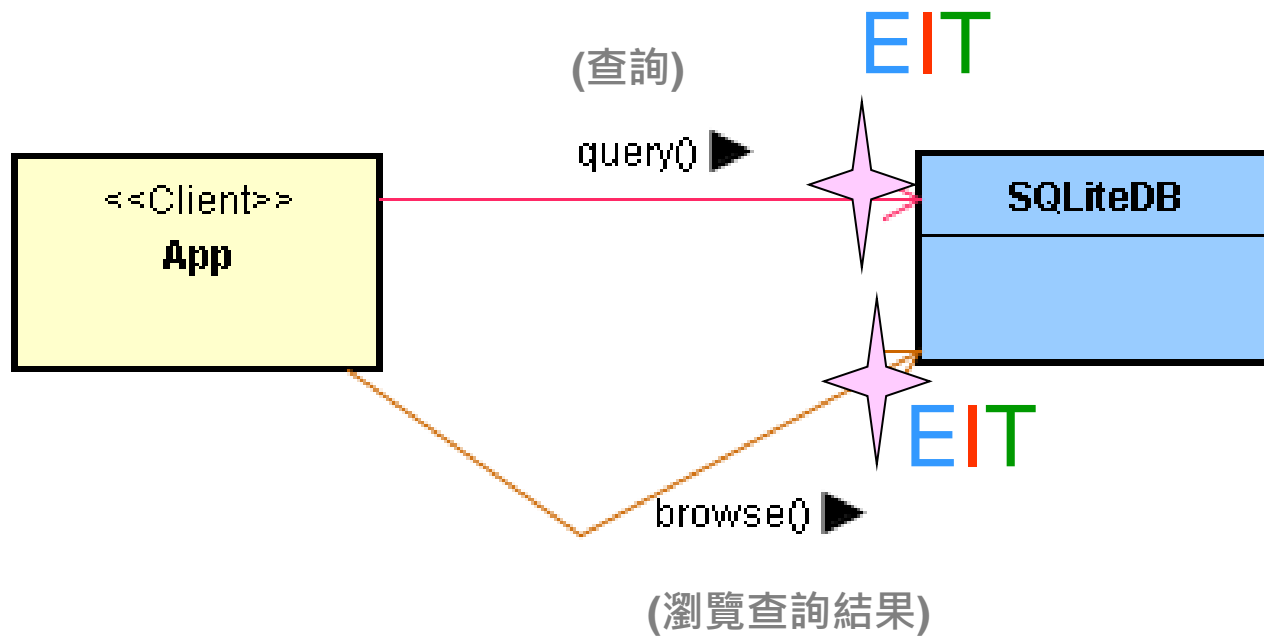


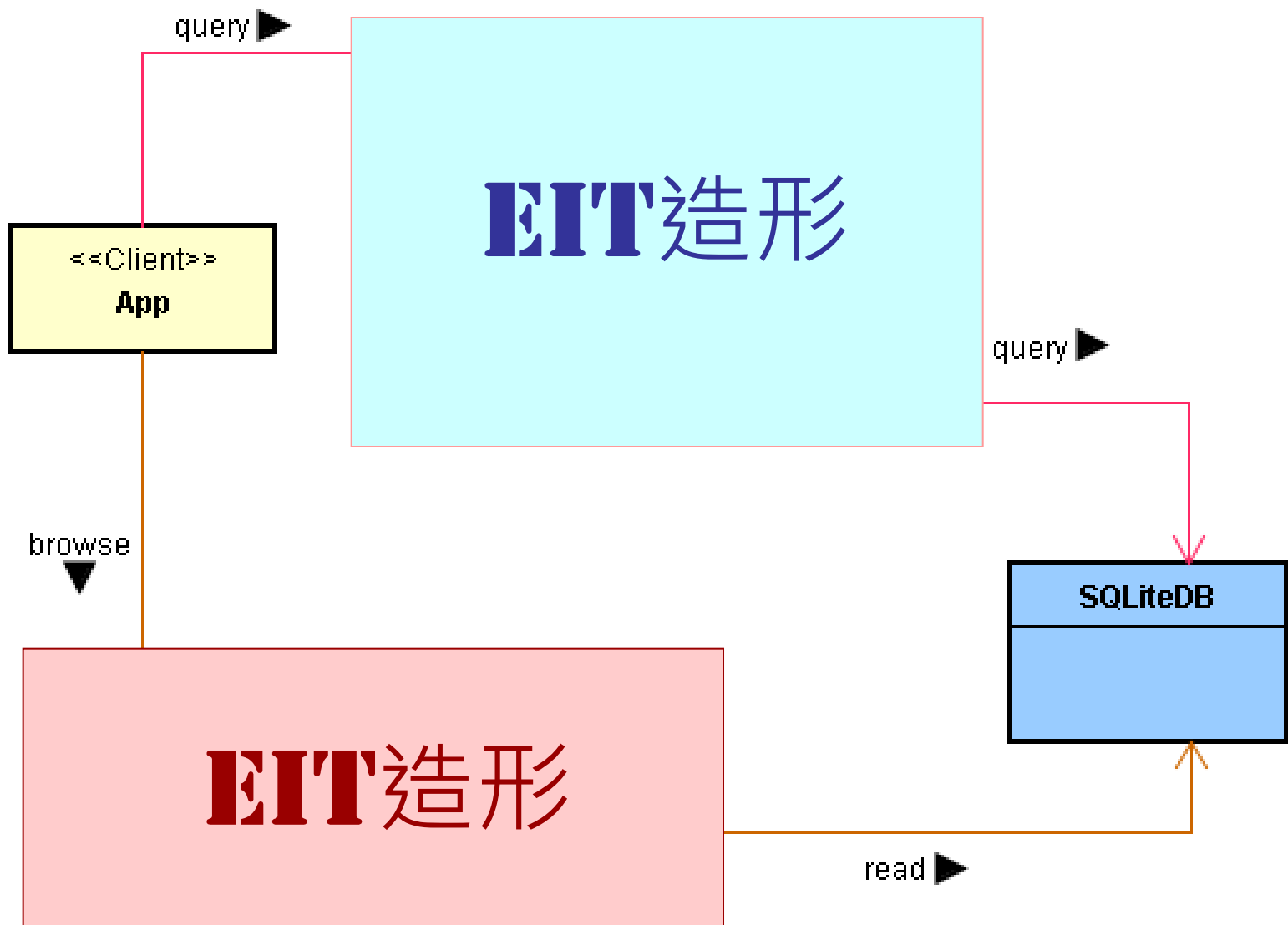


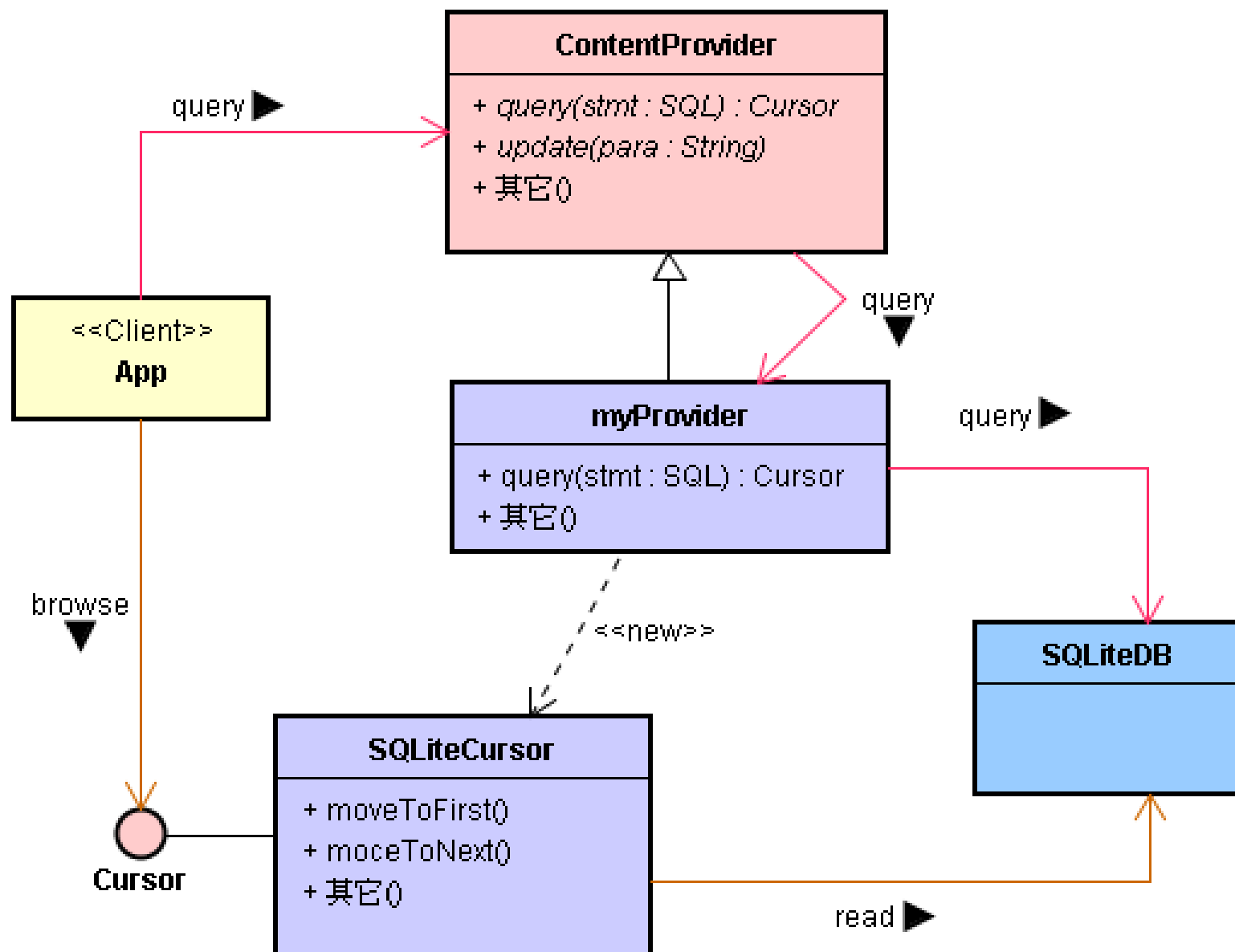
举例

- 例如，Android框架里，在其App与DB引擎(SQLite DB)之间，添加了EIT造形，让App可以跨越DB引擎(小)平台。

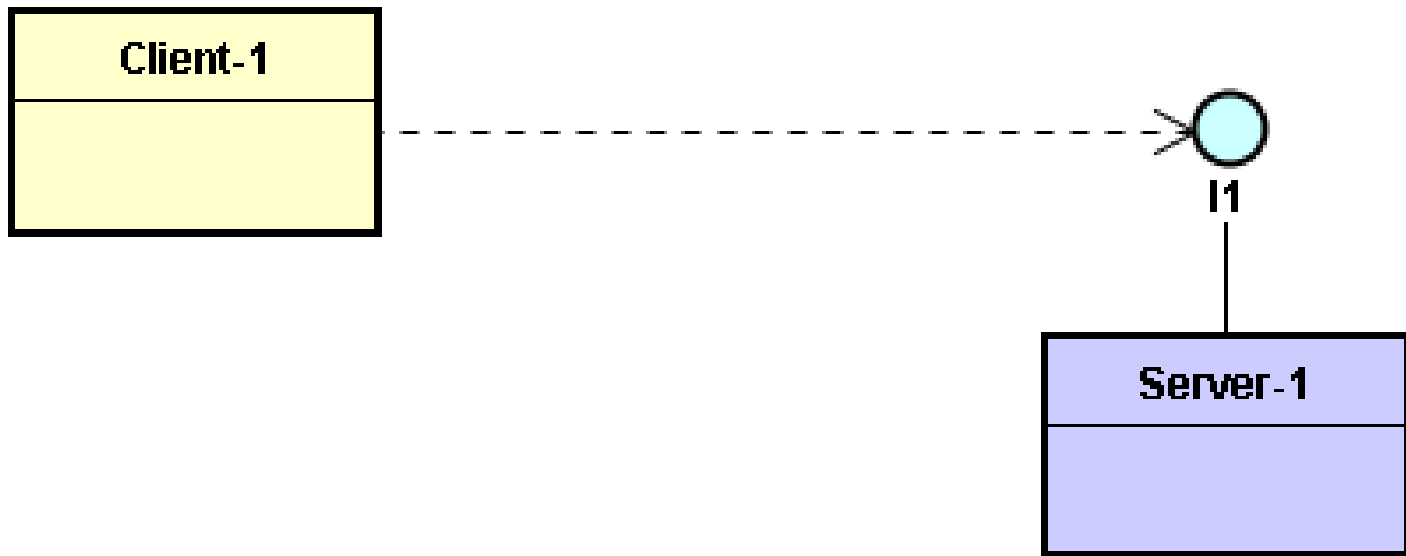


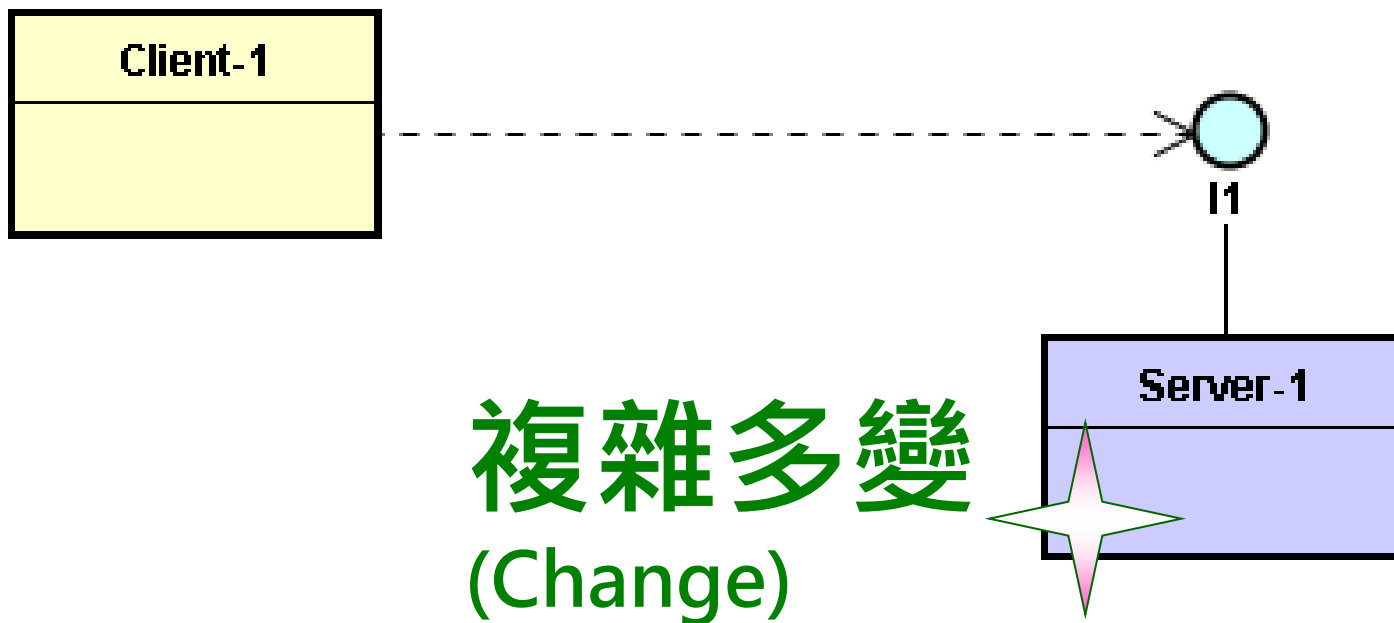




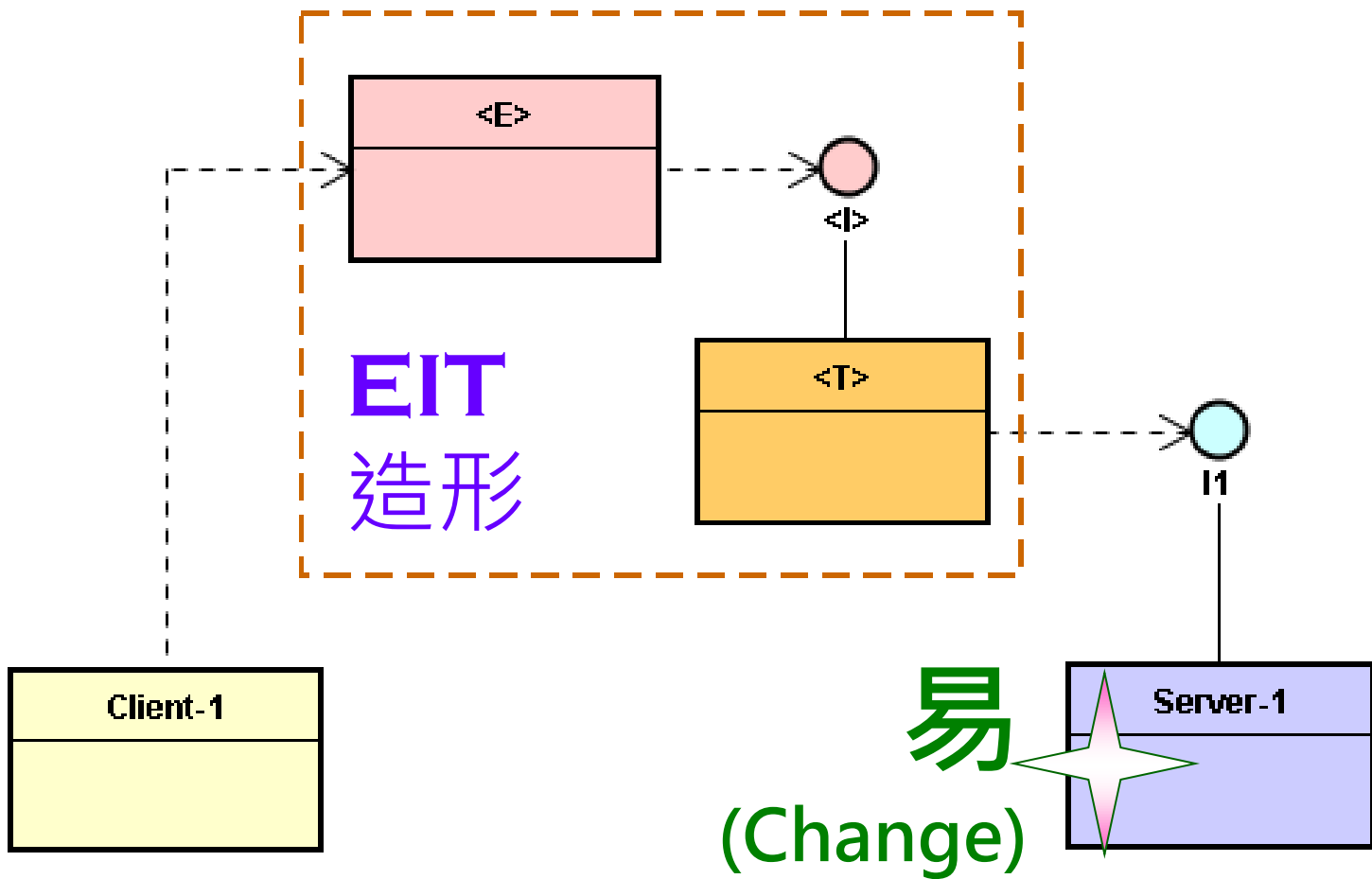


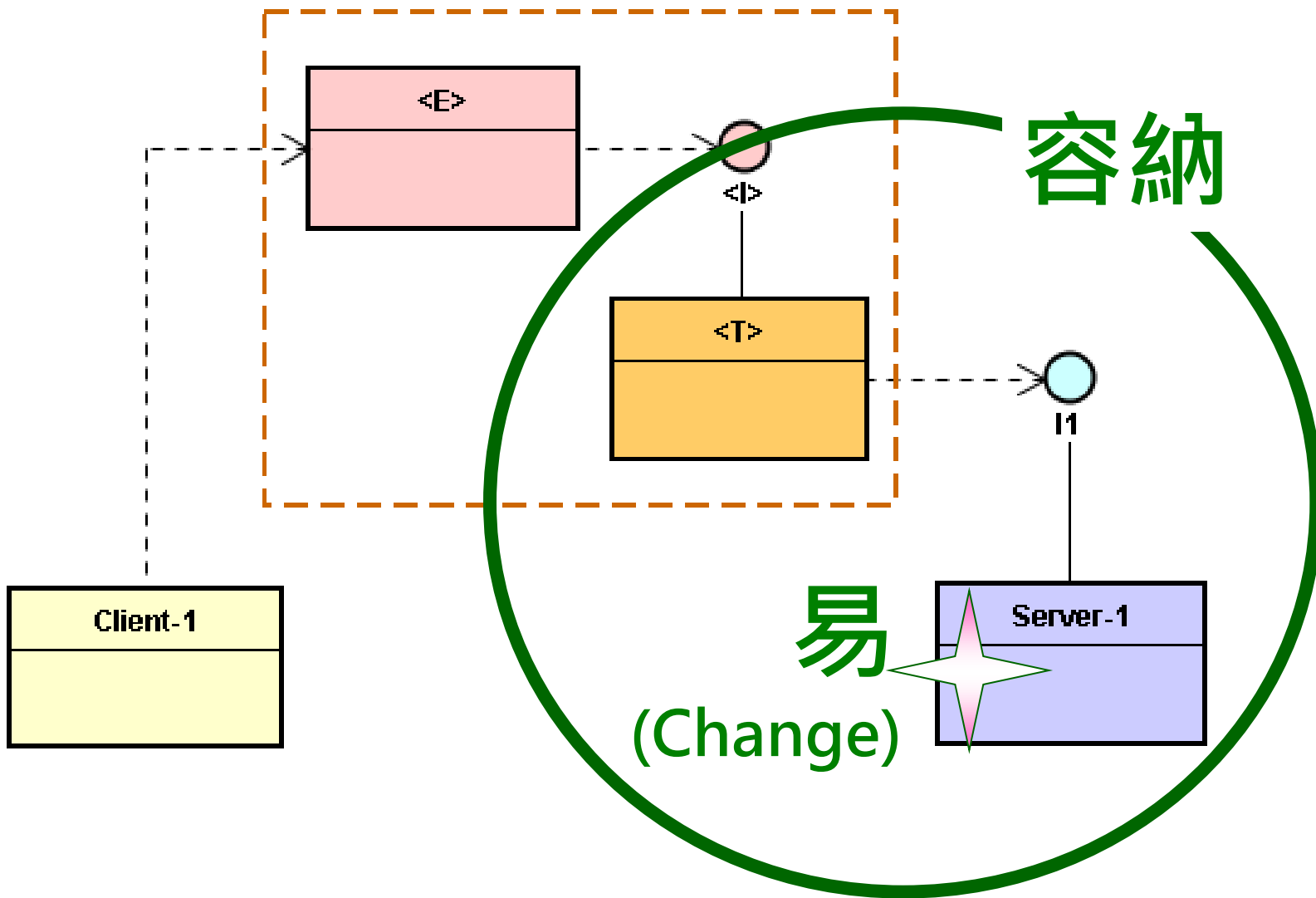
3、结语











容纳(平台的复杂多变)

= 容纳“ 善变”

= 容纳“ 易”

= 容易(Easy)

Thanks...

