蔡學鏞 @ ModernWeb Conference

三維軟體架構

顛 覆 你 的 軟 體 系 統 架 構 認 知

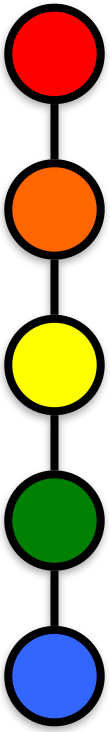
一句話說明軟體架構是什麼

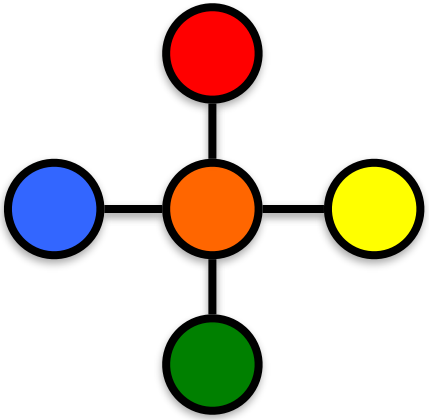
軟體架構就是...

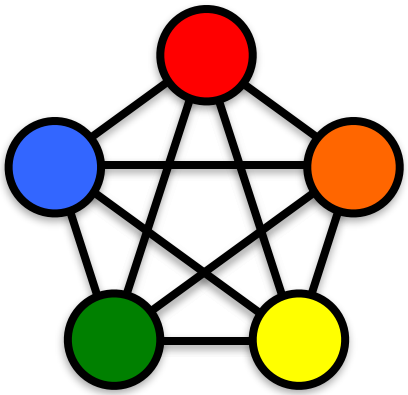
「關鍵」的「程式模組」間
「組織」與「互動」的方式

每種架構都有各自的優缺點

說說這三種架構方式各自的優缺點



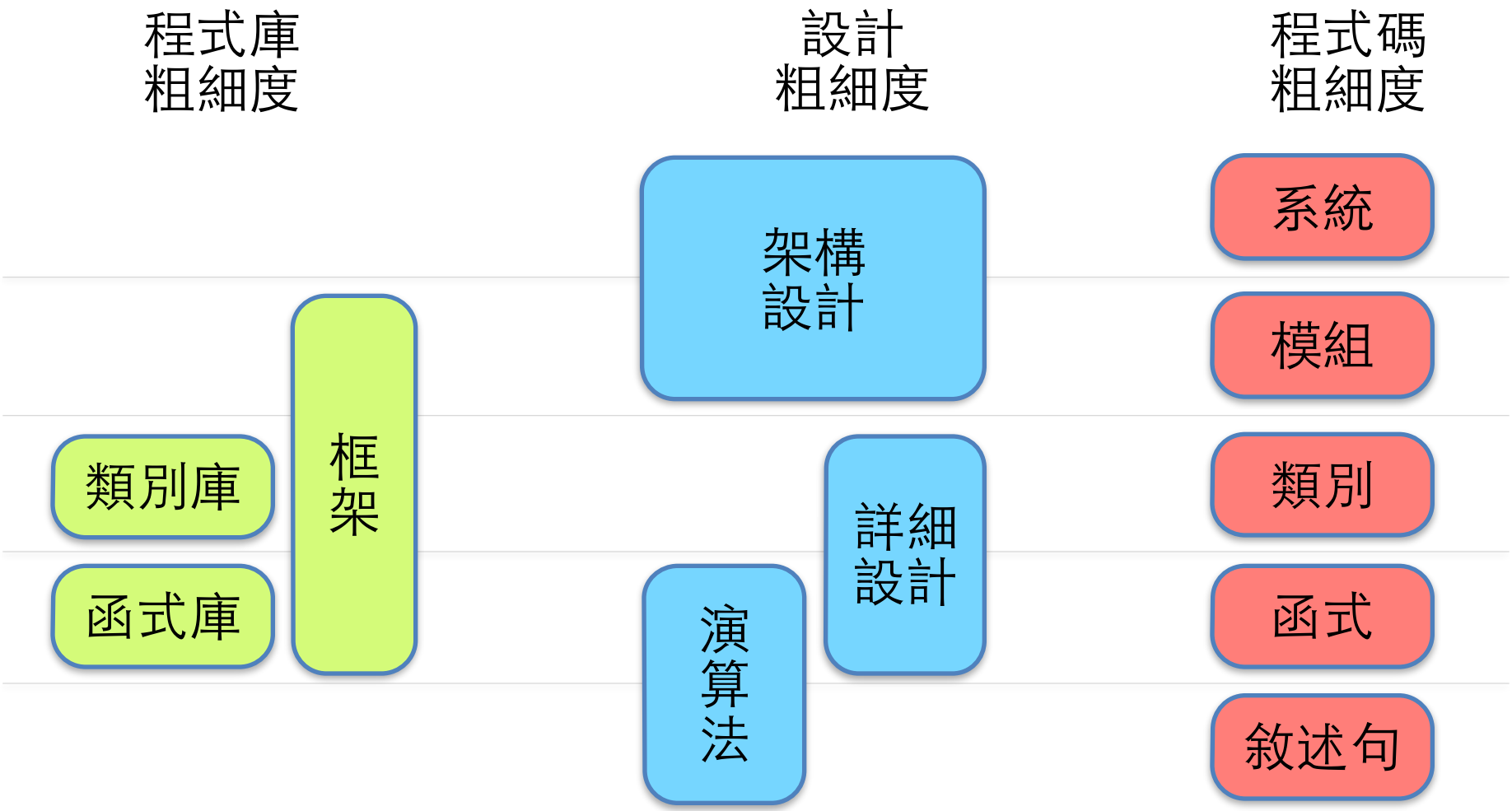




結論：分清楚每個架構方式的優缺點，
及其各自的適用場景

軟體架構不是什麼

許多人常將架構與設計模式和框架混為一談，這是錯的



架構師只關心大格局的需求

【架構師】
關心大格局的
設計需求

架構
設計

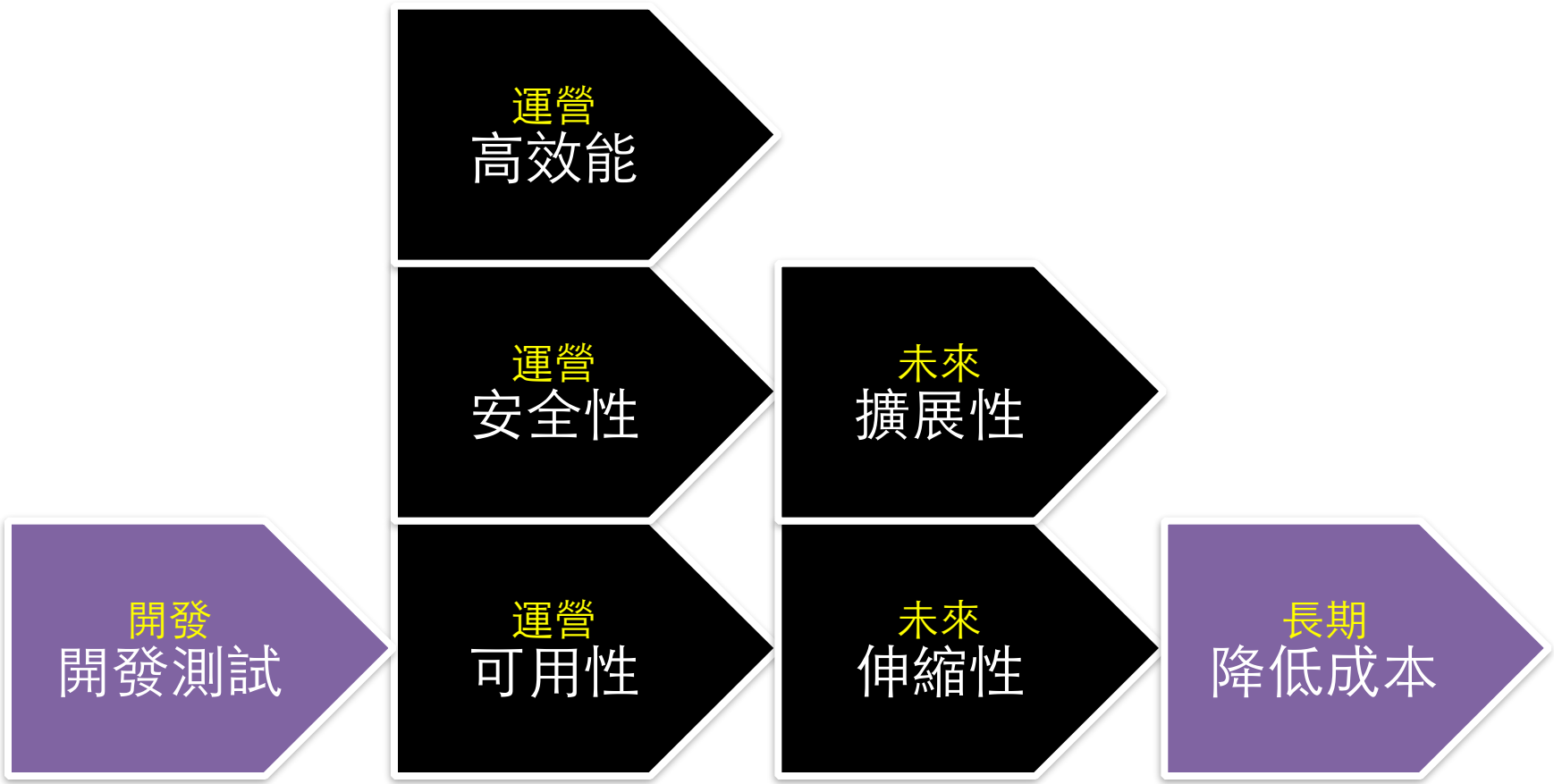
【系統設計師】
關心小的局部的
設計需求

詳細
設計

【演算法工程師】
【開發工程師】
關心具體問題的程
式法解決方法，關
心效率

演
算
法

什麼是「大格局」的需求?

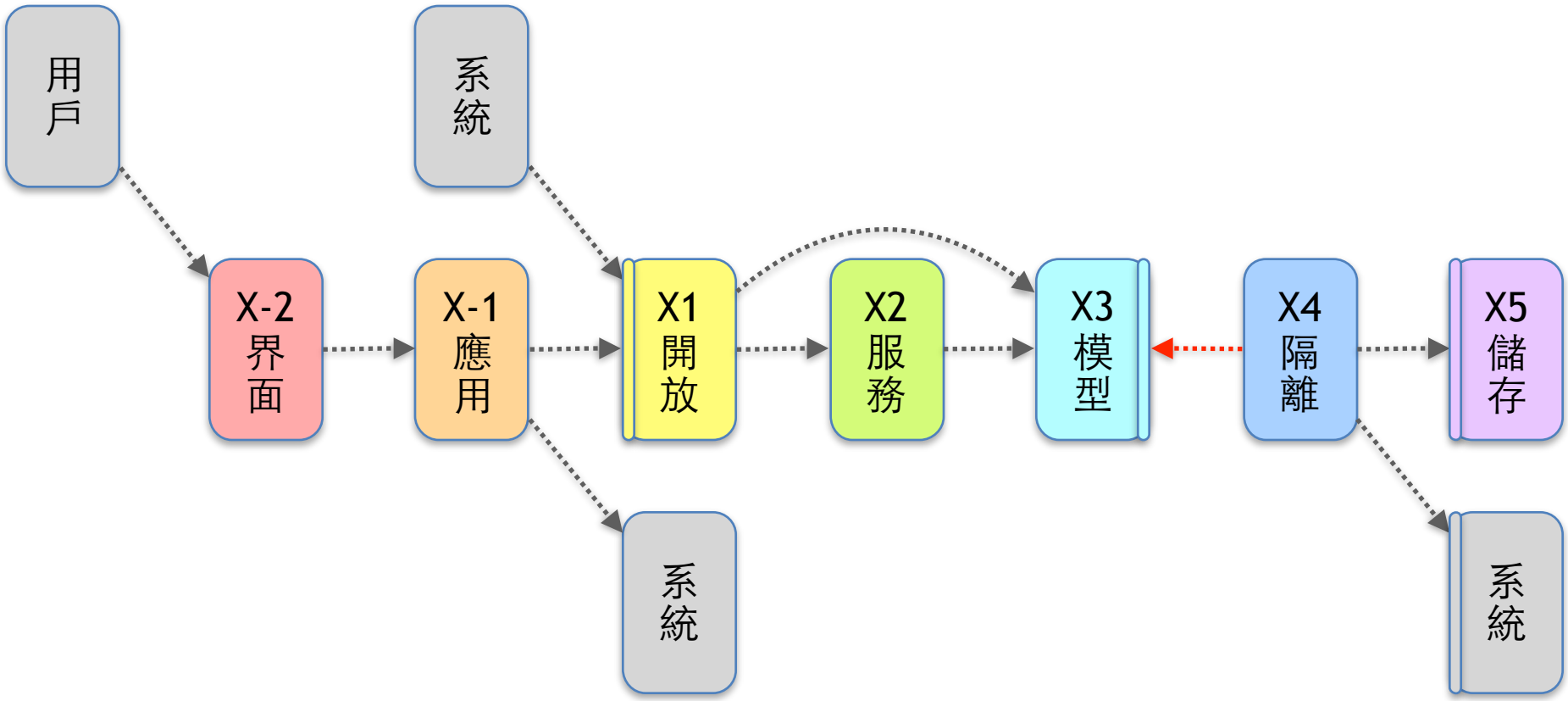


一個軟體系統怎麼可能如此完美?

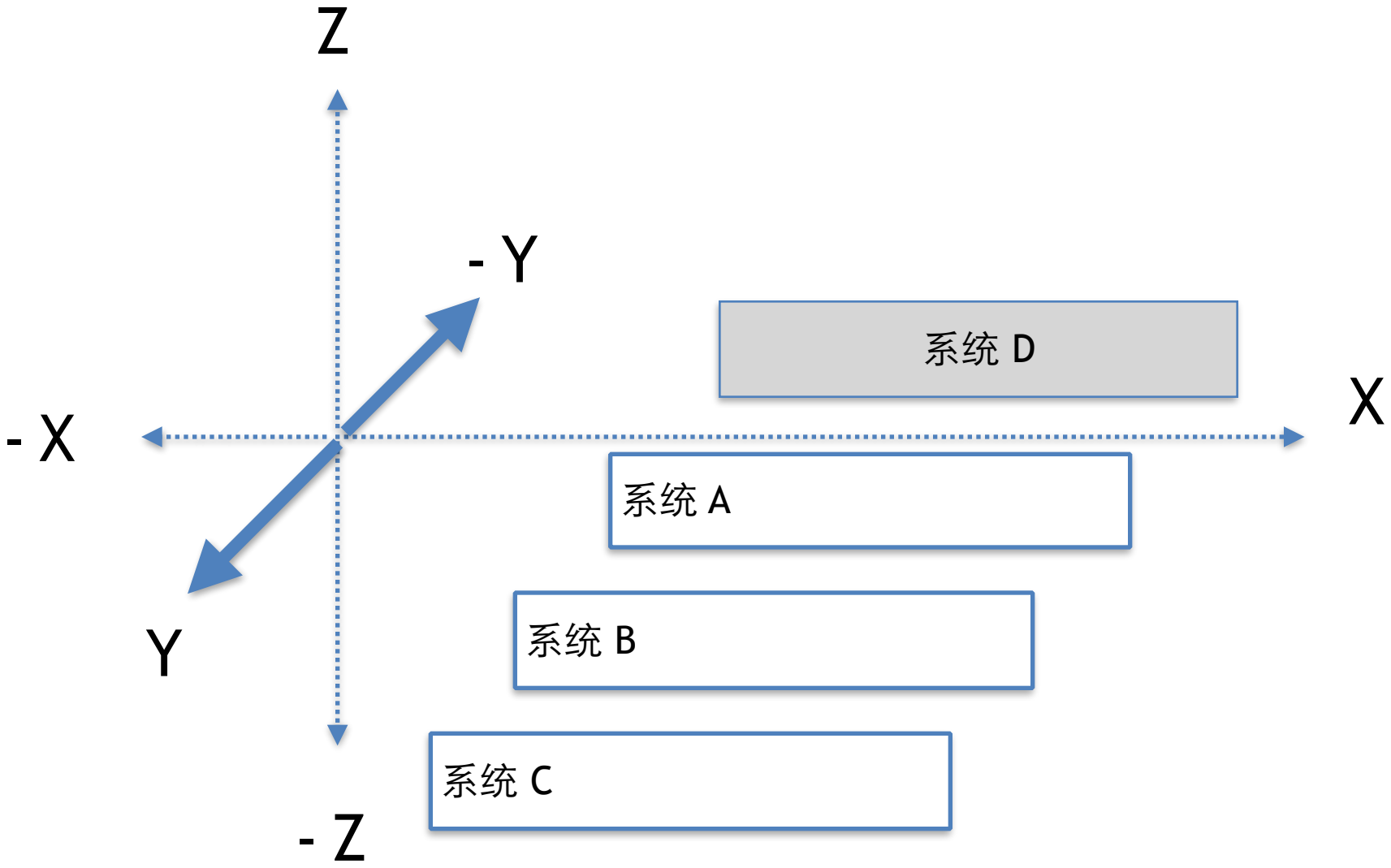
答案是：把整体的目标分发到局部

后面介绍架構的三维座标分割法

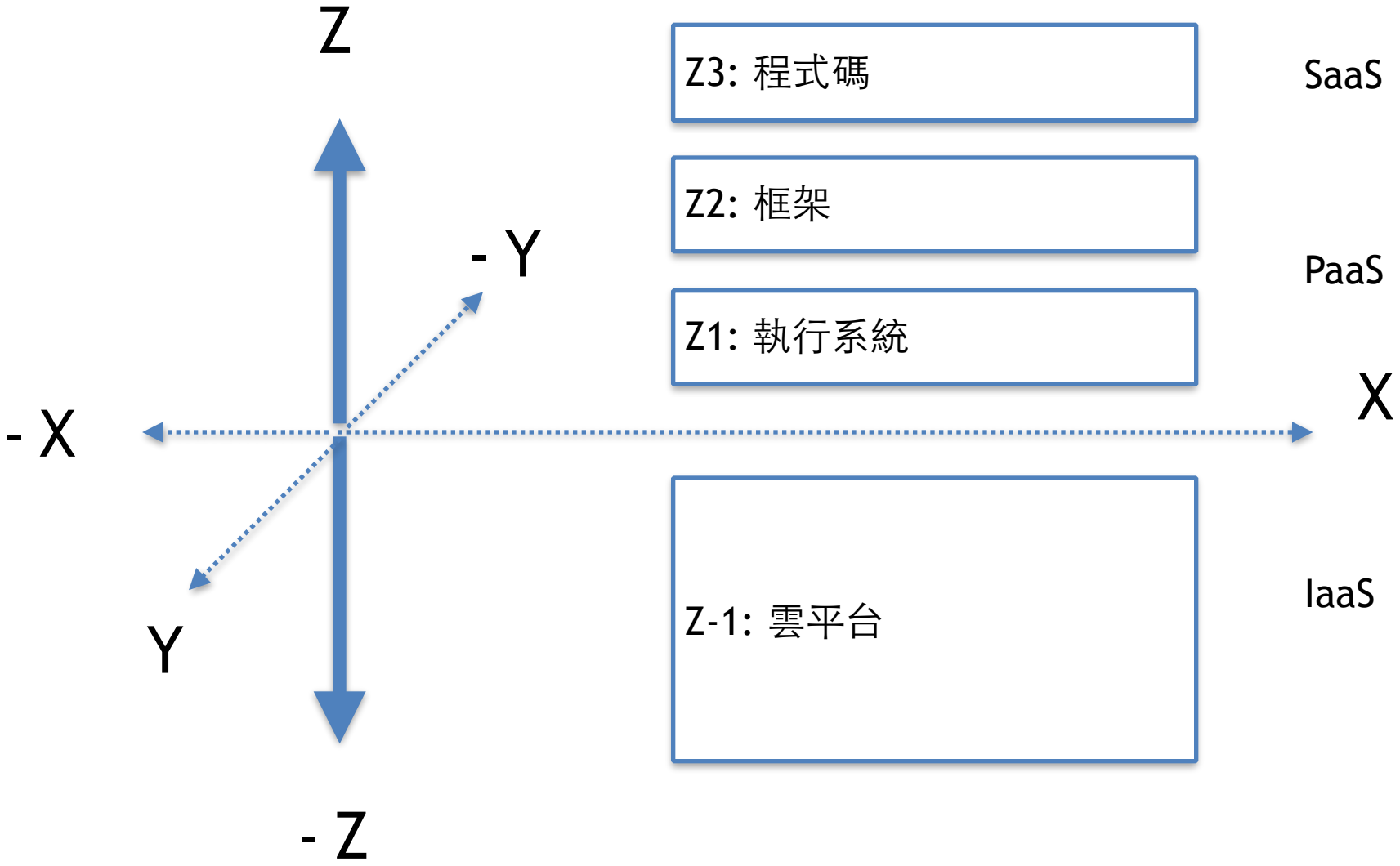
X 座標分為 7 層



Y座標分為 N 個系統



Z 座標分為 4 層



界面

了解用户，且具有审美观

HTML/CSS/JavaScript/HTML/Android/iOS/PHP

应用

了解市场与用户，具有产品设计能力

PHP /Python/Ruby/Dart/NodeJS

开放

了解领域和公司的战略

人格特质+
专业能力

服务

有比较强的业务知识与详细设计能力。
重构，设计模式，持续集成，敏捷，DDD

模型

Java/Rust/C++

隔离

了解领域与合作夥伴

Java/Rust/C++

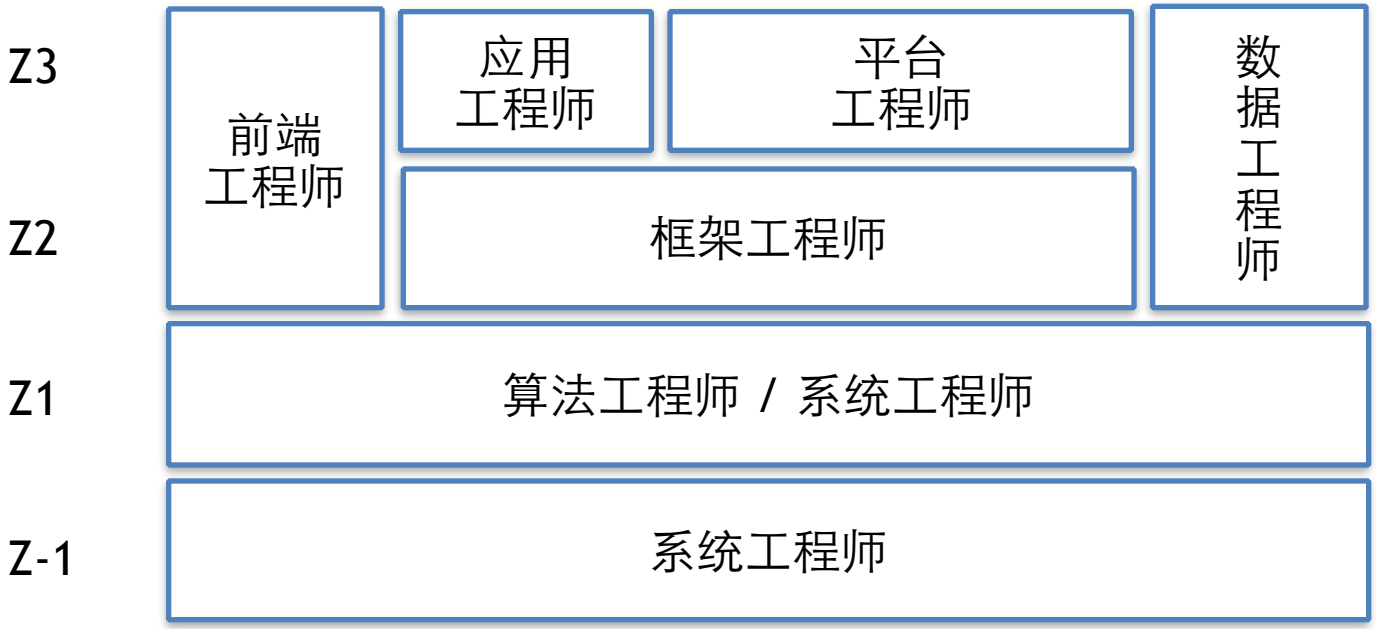
数据

了解领域与数据库

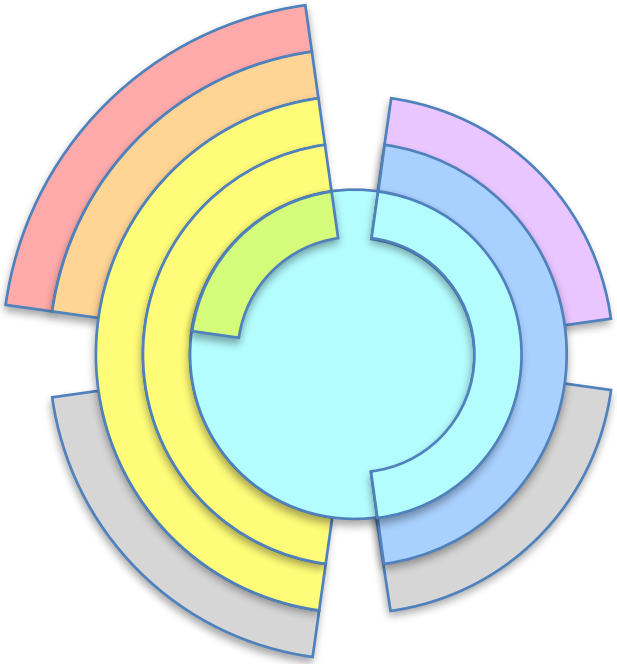
NoSQL / RDBMS / MoM / Cache

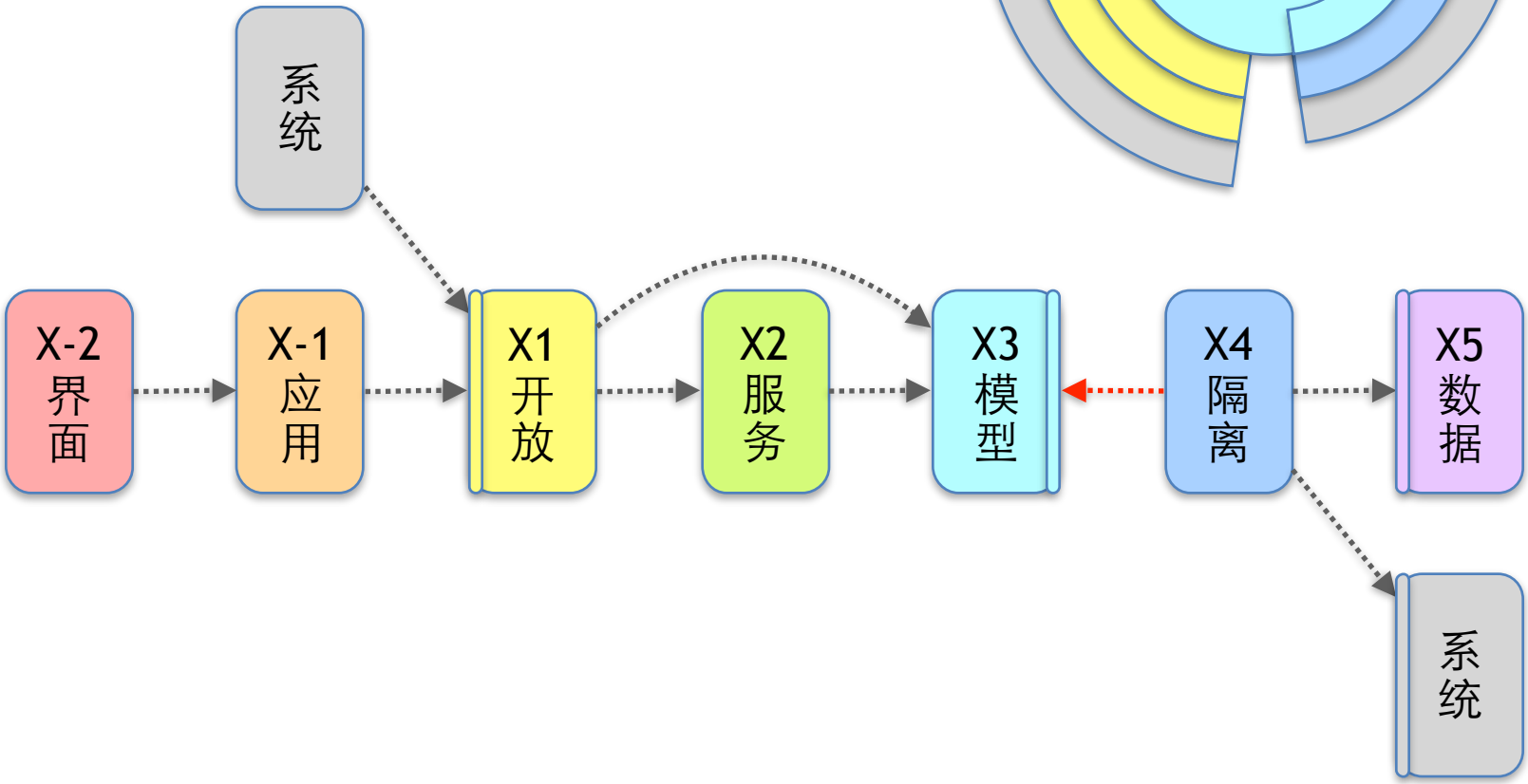
几类工程师的职业生涯

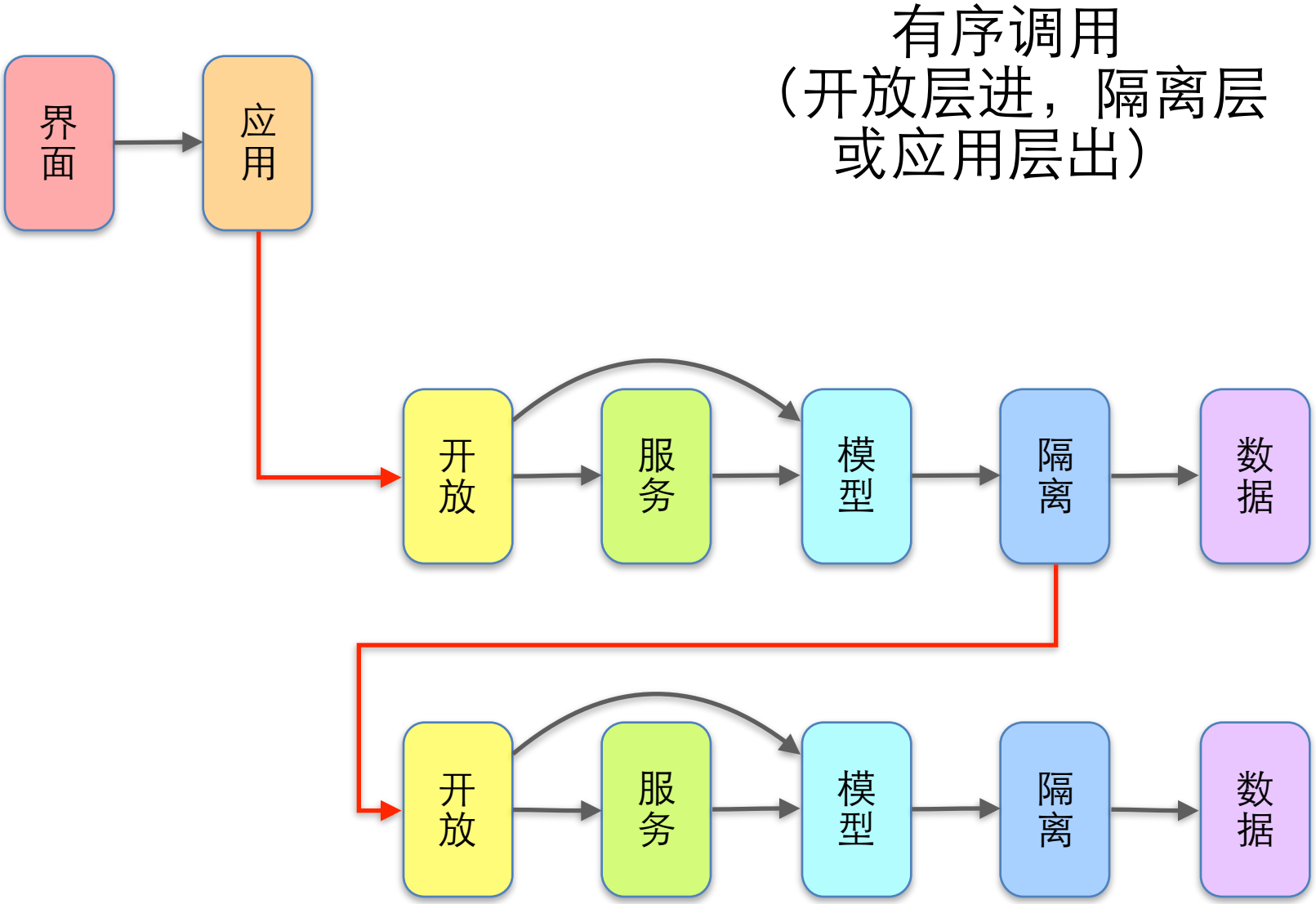
- 跳槽薪资的涨幅，通常何者比较高?
- 职业生涯，通常何者可以维持稳定比较久?



符合六邊形架構的理念

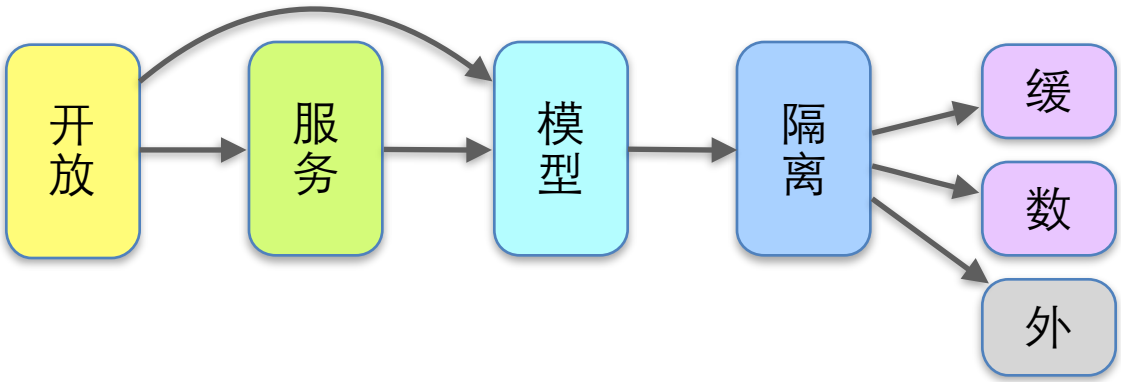




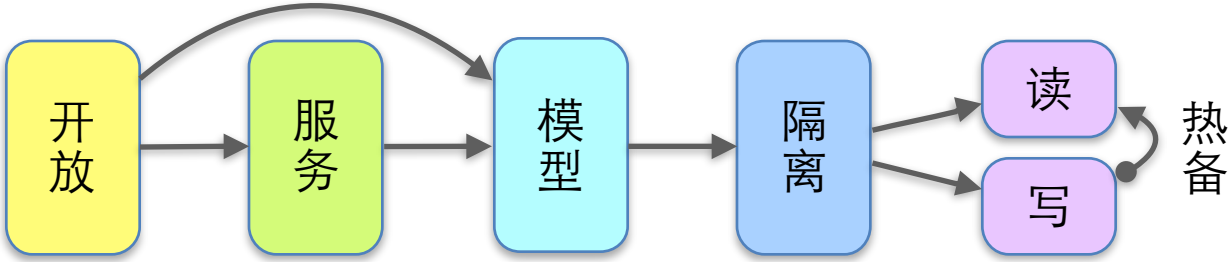

有序调用
（开放层进，隔离层
或应用层出）
界面
应用
开放
服务
模型
隔离
数据
开放
服务
模型
隔离
数据

解决数据瓶颈

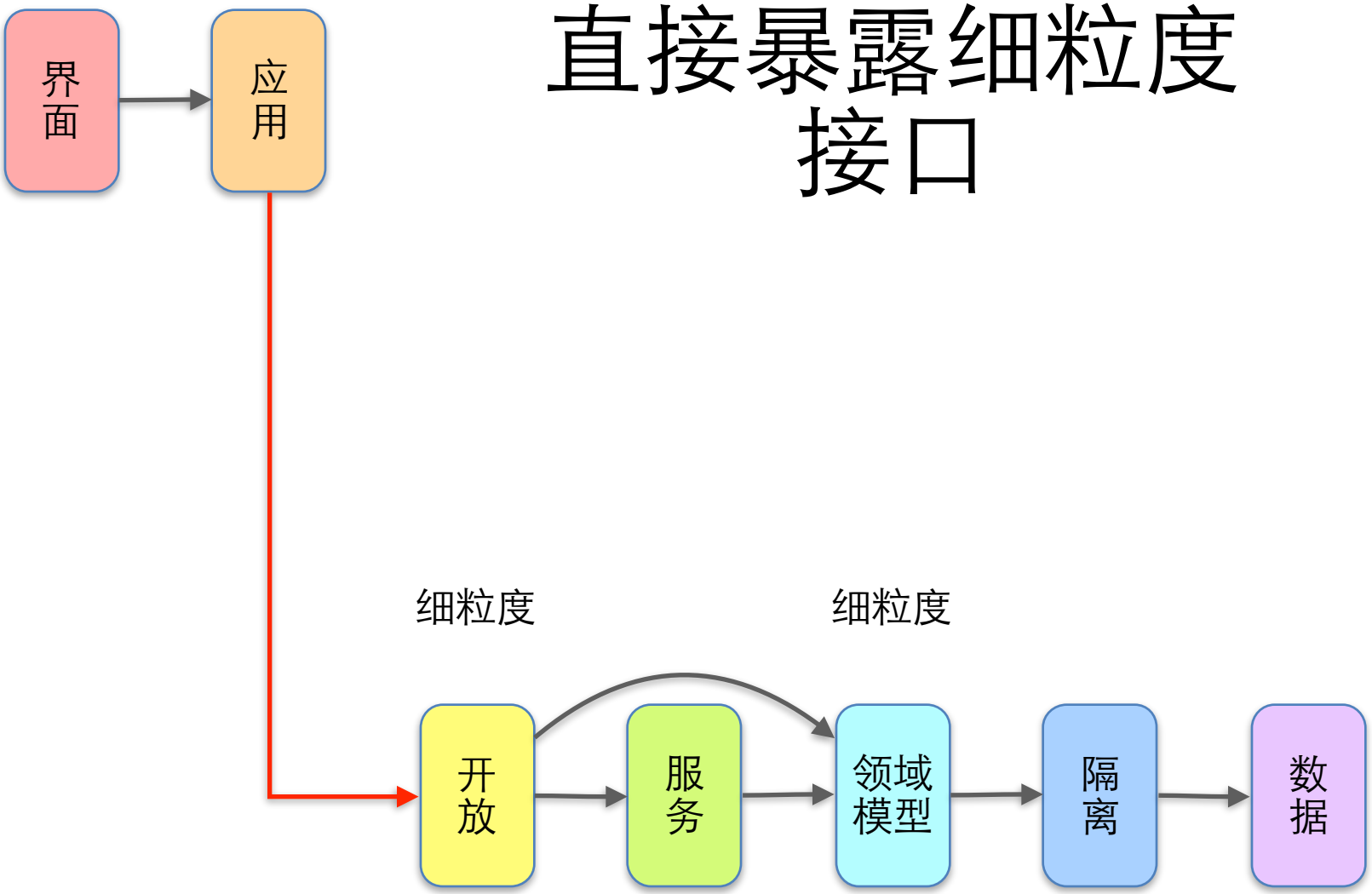
数据缓存



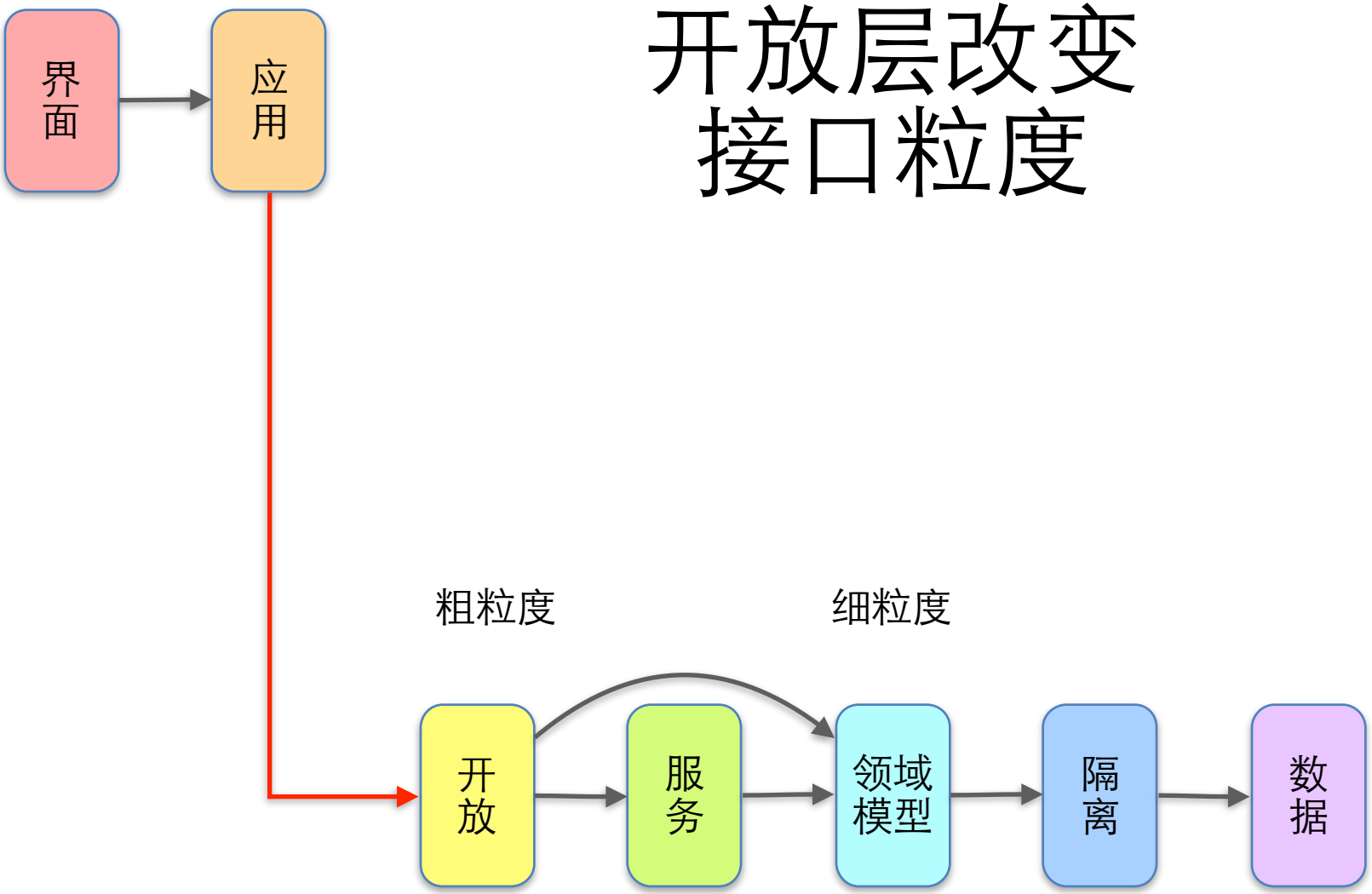
数据读写分离



直接暴露细粒度接口

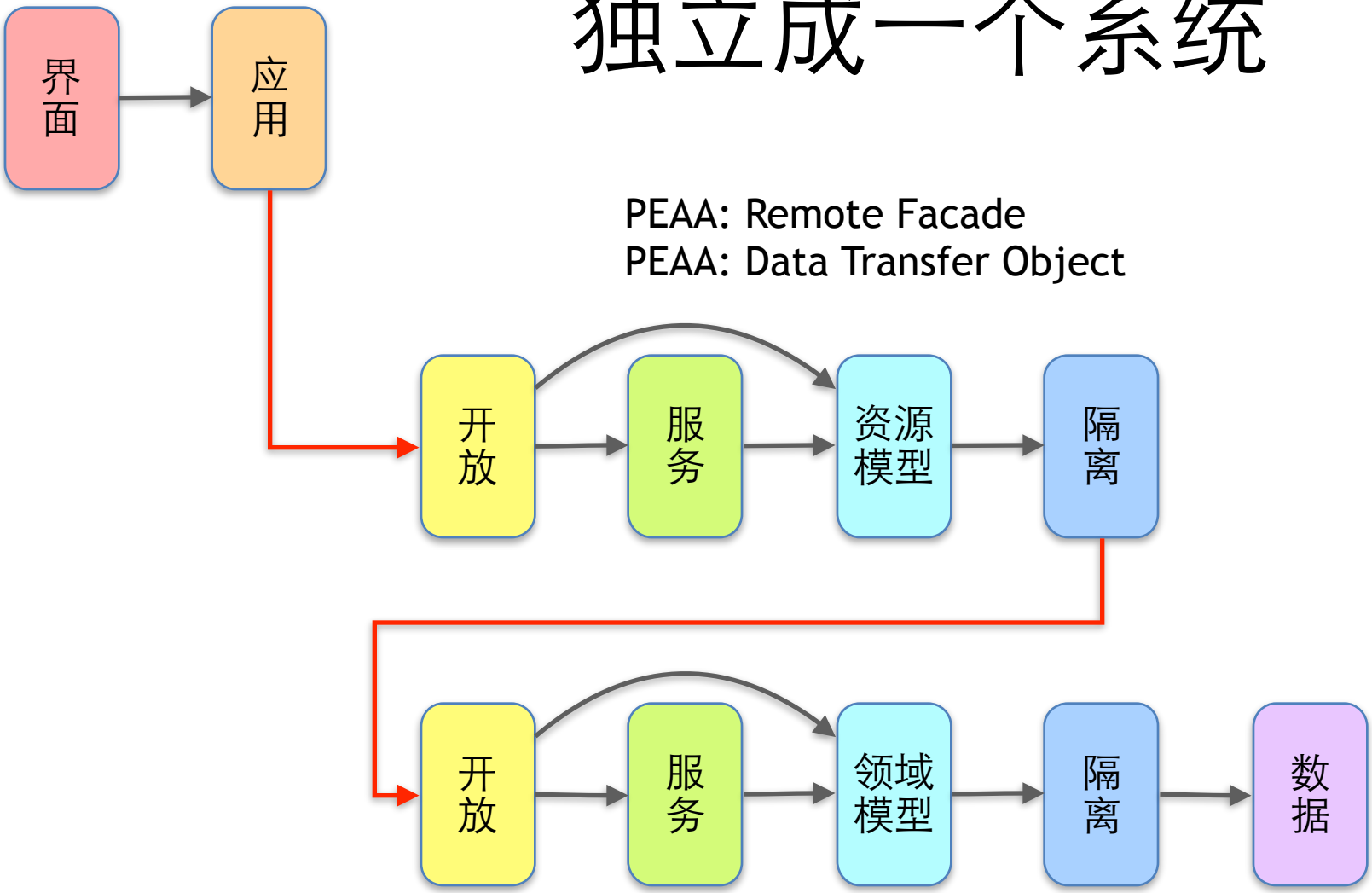


开放层改变接口粒度

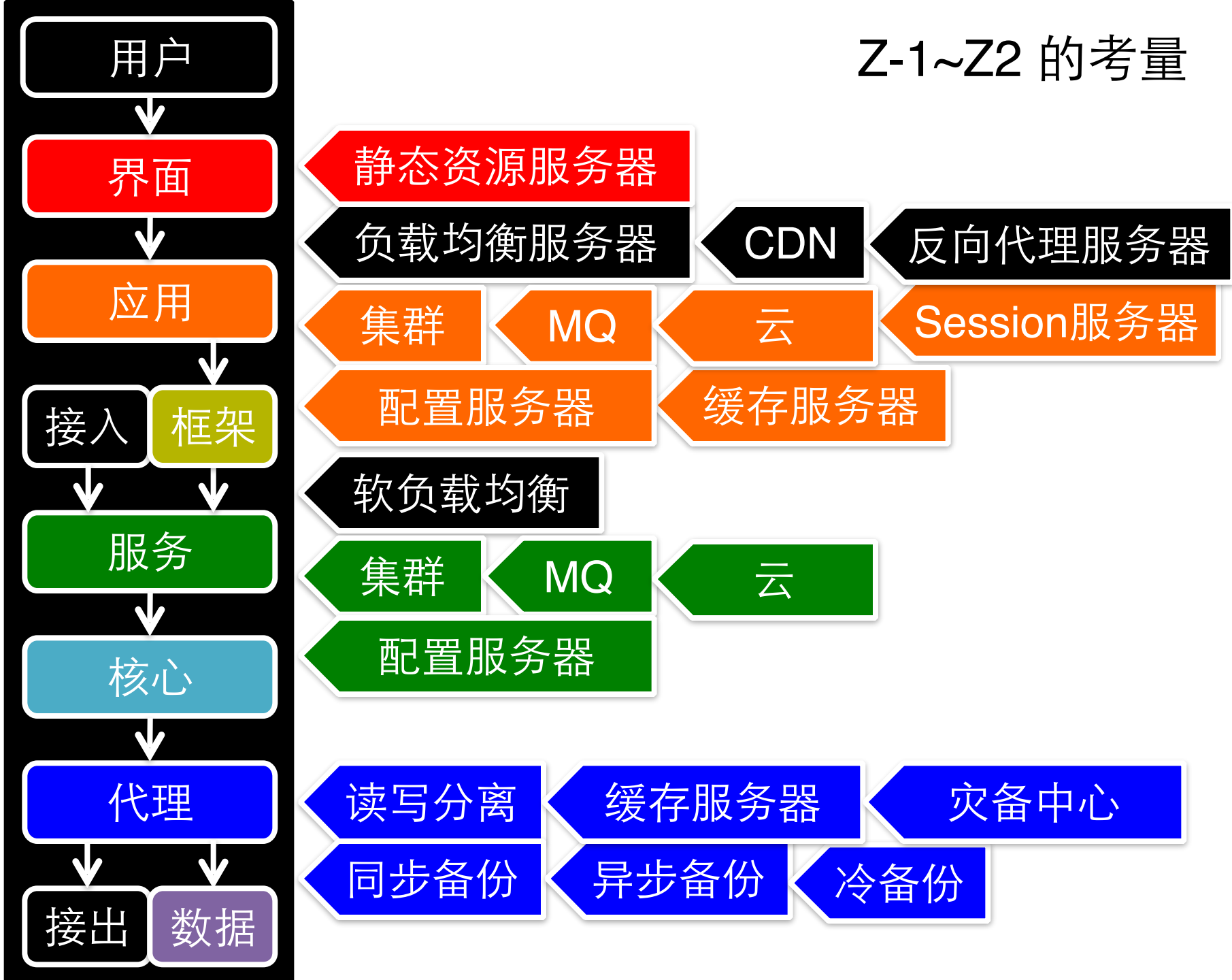


PEAA: Remote Facade
PEAA: Data Transfer Object

把Remote Facade独立成一个系统



Z-1~Z2 的考量

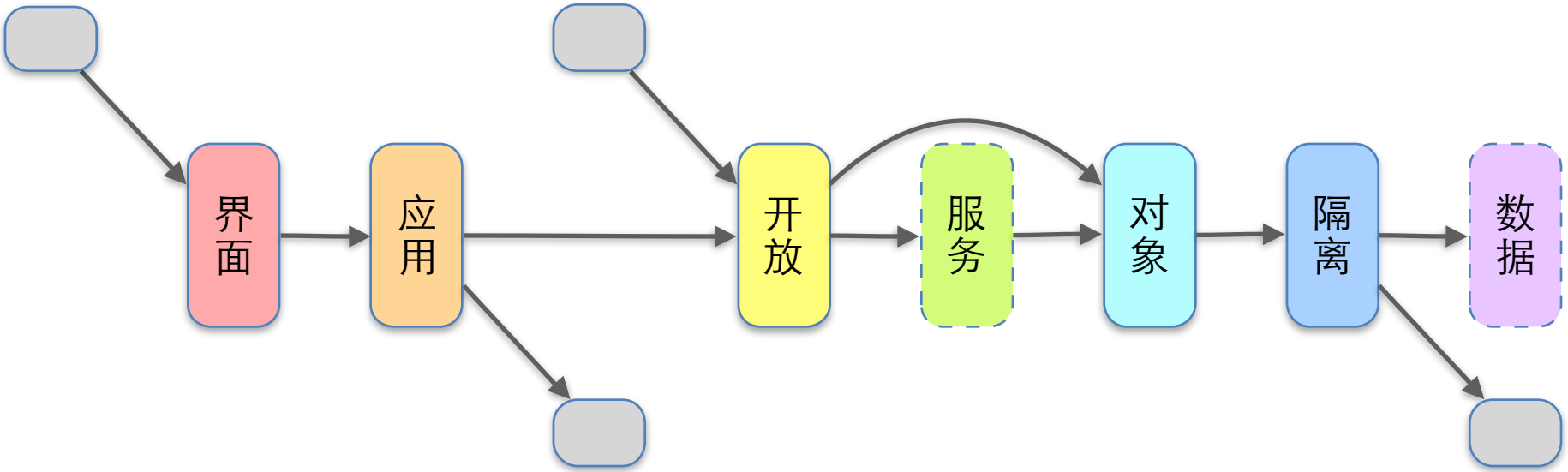

用户
界面
静态资源服务器
负载均衡服务器
CDN
反向代理服务器
应用
集群
MQ
云
Session服务器
配置服务器
缓存服务器
接入
框架
软负载均衡
服务
集群
MQ
云
配置服务器
核心
代理
读写分离
缓存服务器
灾备中心
同步备份
异步备份
冷备份
接出
数据

企业设计模式和七层架构的对应

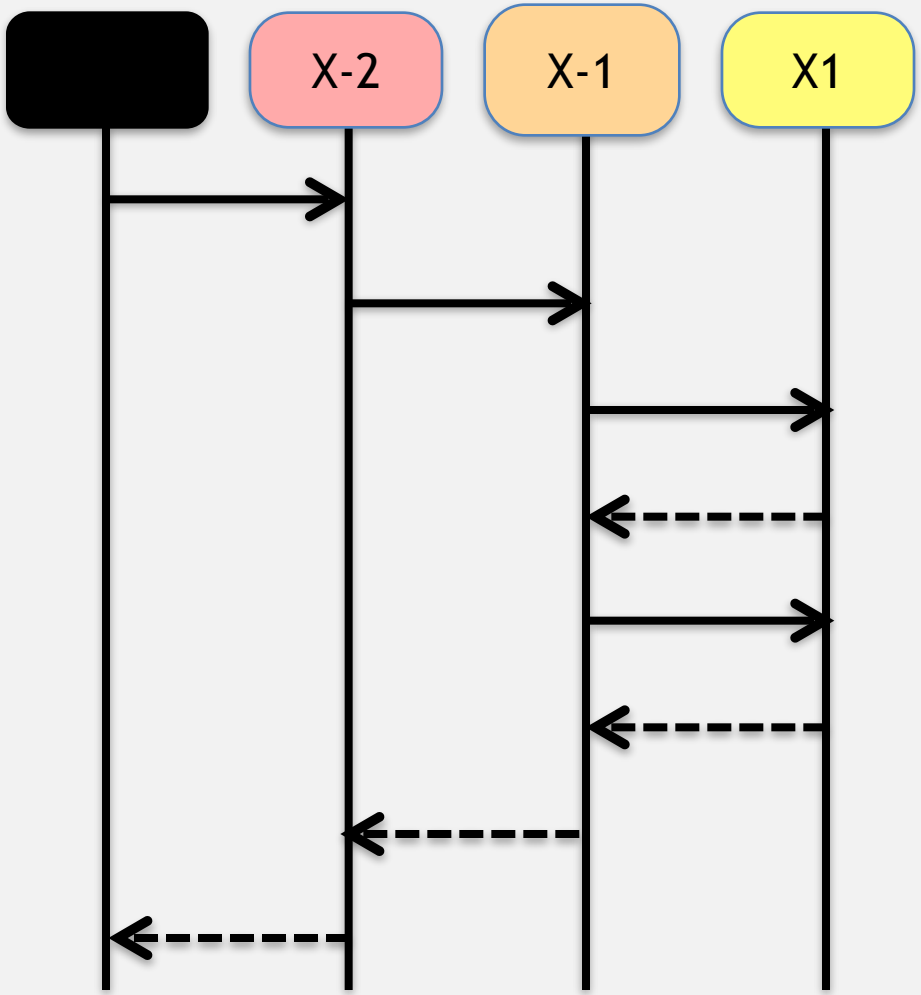
- 界面层：Page Controller, Front Controller, Template View, transform View, Two-Step View
- 应用层：Transaction Script, MVC, Application Controller, Client Session State, Sever Session State, Database Session State
- 开放层：Remote Facade, Data Transfer Object,
- 服务层：Service Layer
- 领域层：Domain Model, Coarse-Grained Lock, Value Object, Money, Special Case, Plug in, Mediator
- 隔离层：Table Module, Table Data Gateway, Raw Data Gateway, Active Record, Data Mapper, Unit of Work, Lazy Load, Identity Field, Foreign Kay Mapping, Gateway, Registry, Service Stub, Optimistic Offline Lock, Pessimistic Offline Lock, Implicit Lock

根据业务拆分系统

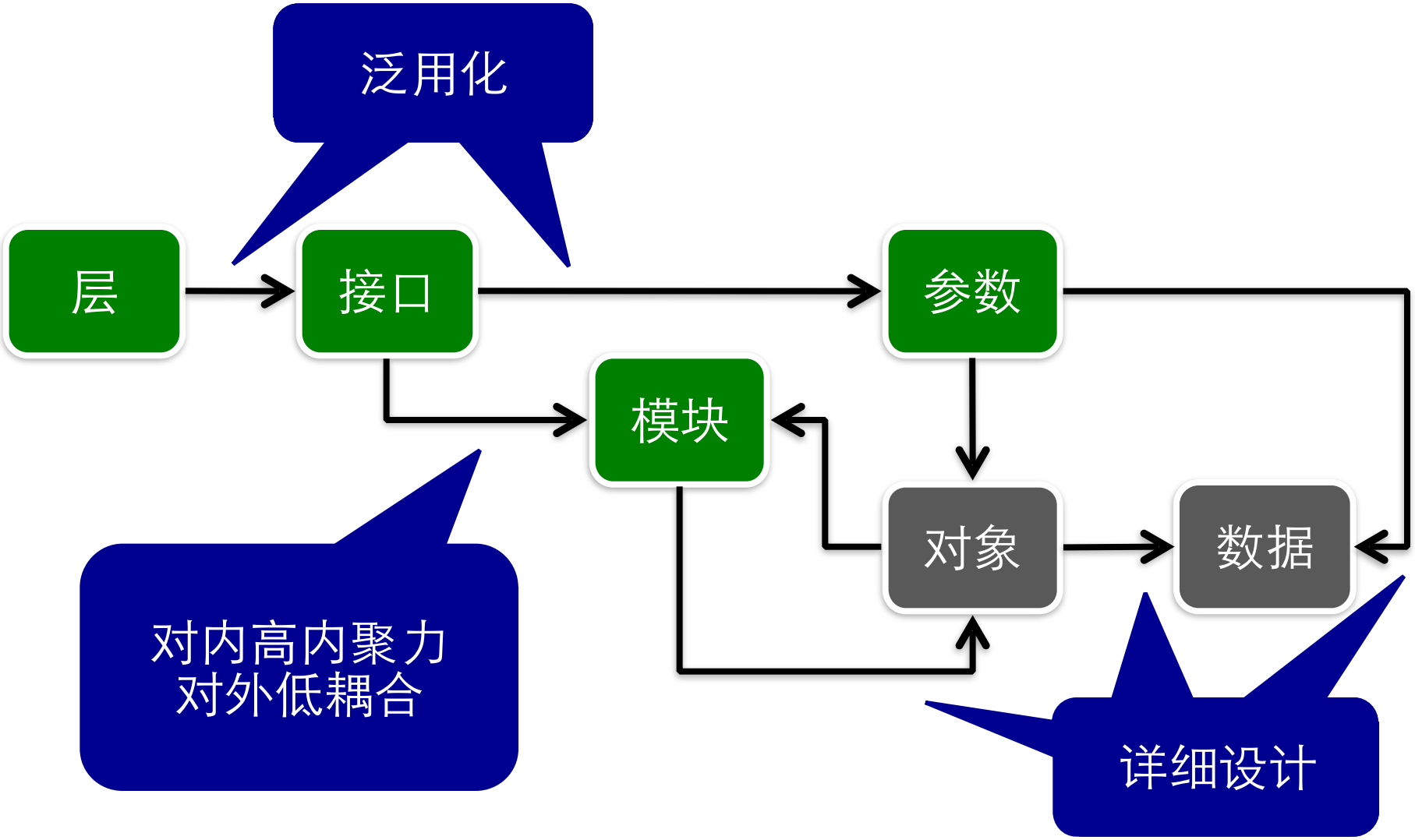
角色名称	人?	系统名称	用例描述

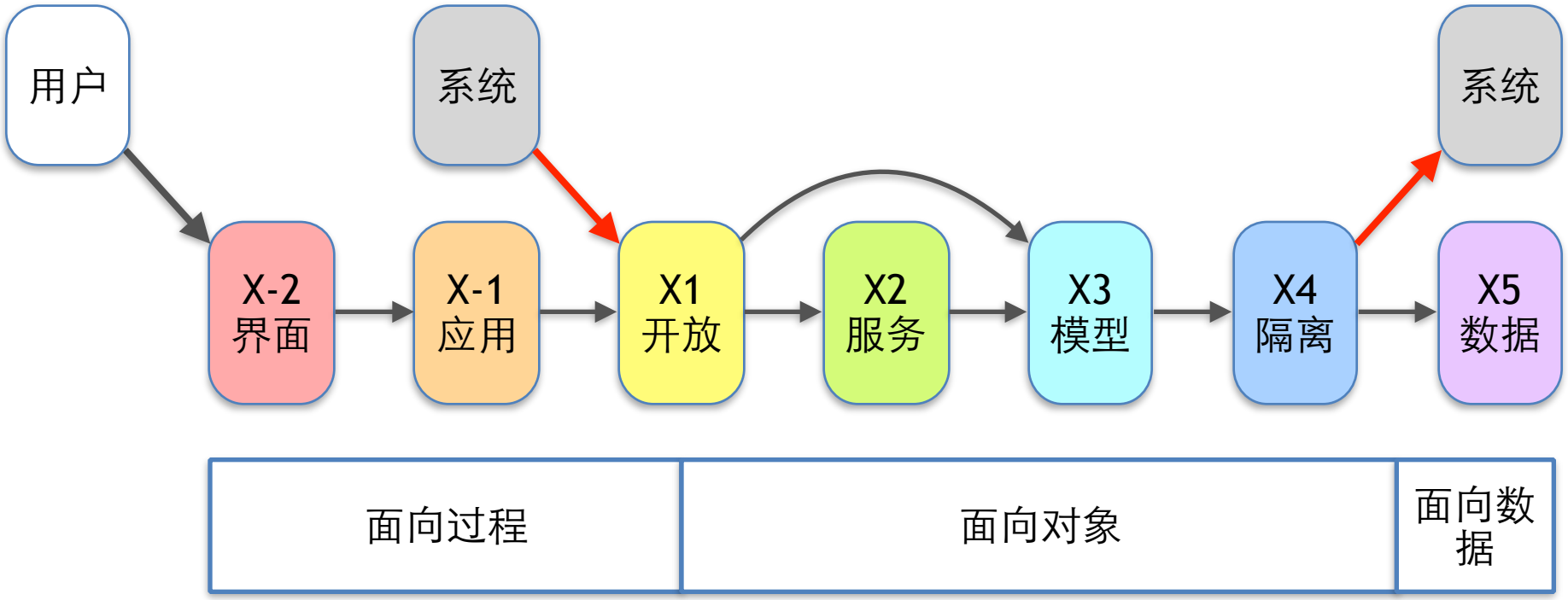


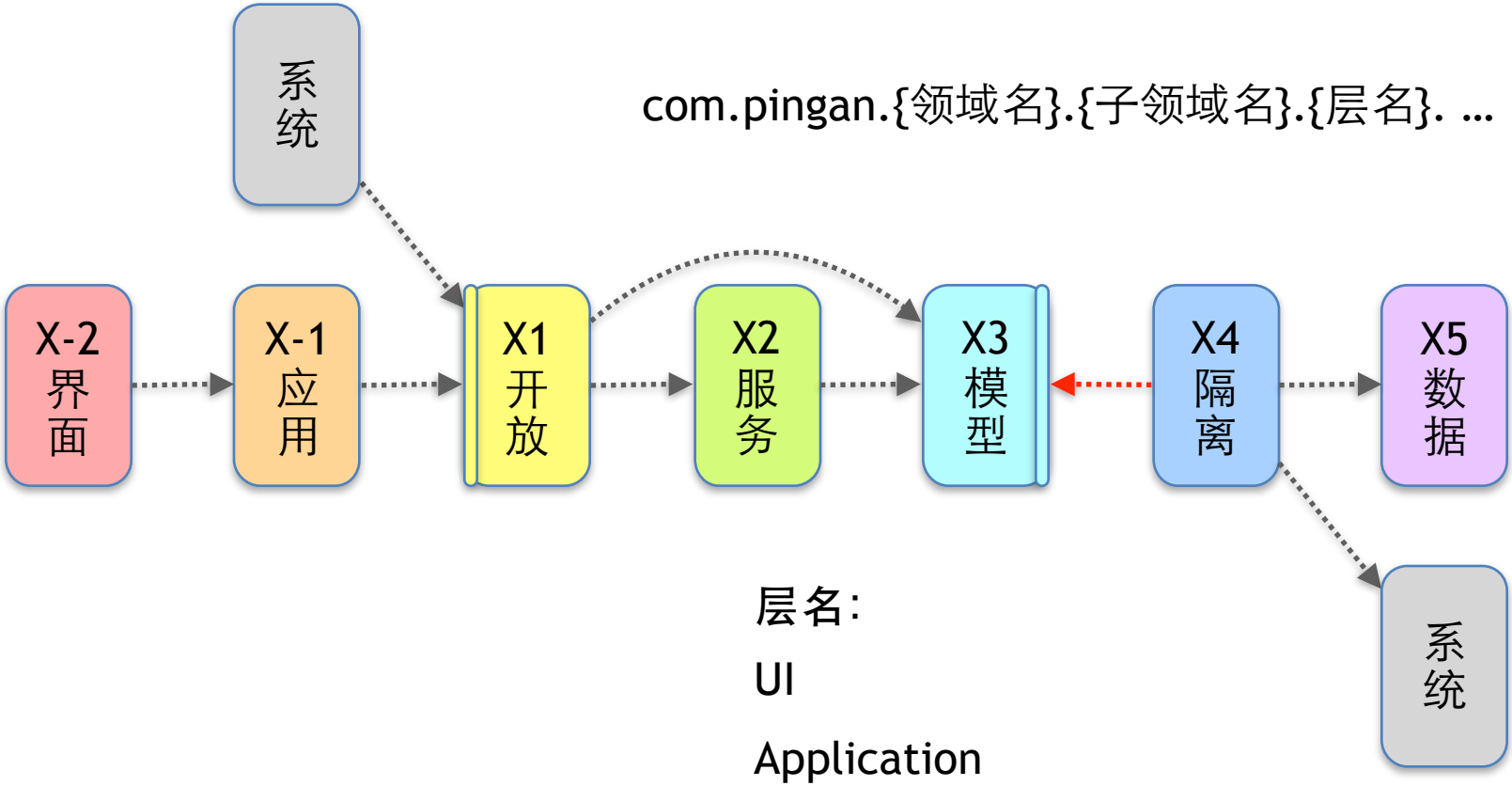
业务设计（找出接口与参数）



业务需求的推导过程




用户
系统
系统
X-2
界面
X-1
应用
X1
开放
X2
服务
X3
模型
X4
隔离
X5
数据
面向过程
面向对象
面向数据



com.pingan.{领域名}.{子领域名}.{层名}. ...

层名:

UI

Application

API

Open Host

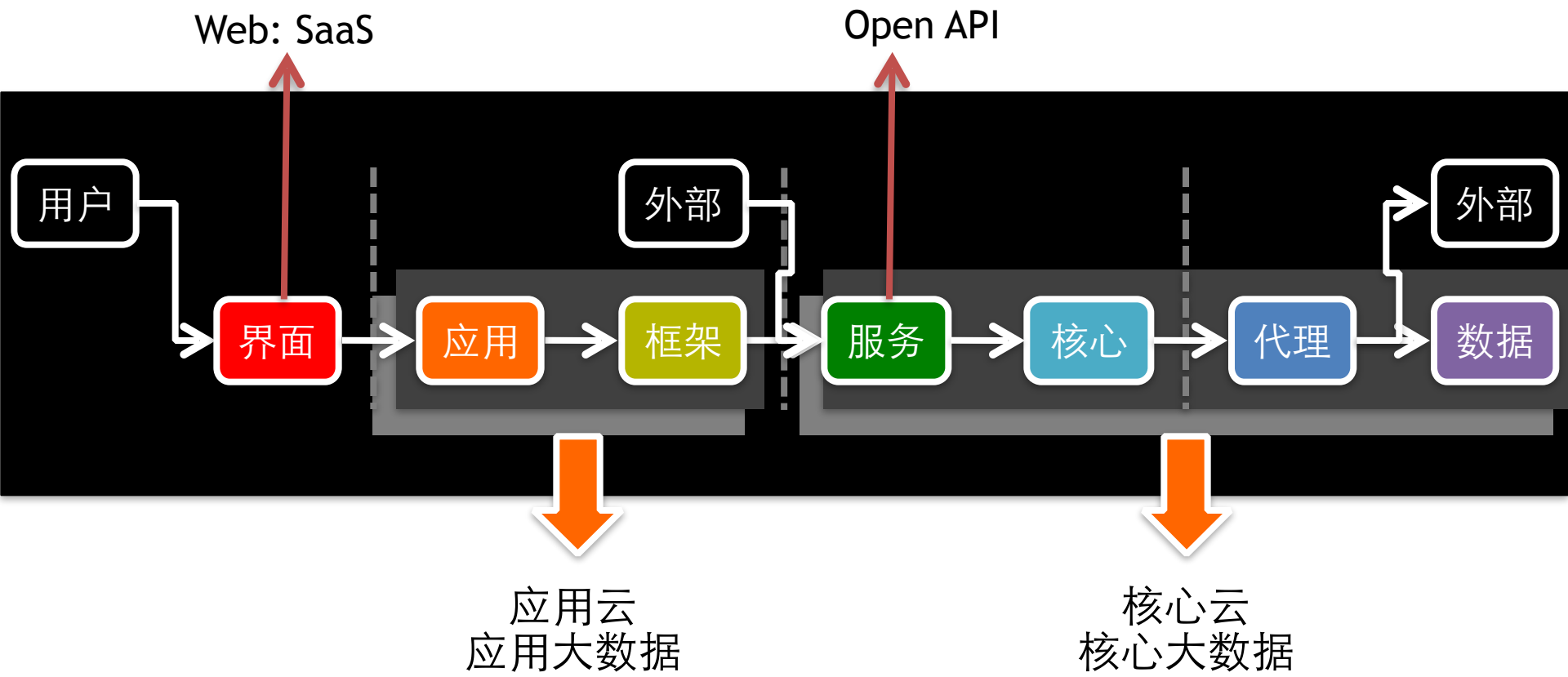
Service

Model

SPI

Anticorruption (Isolation)

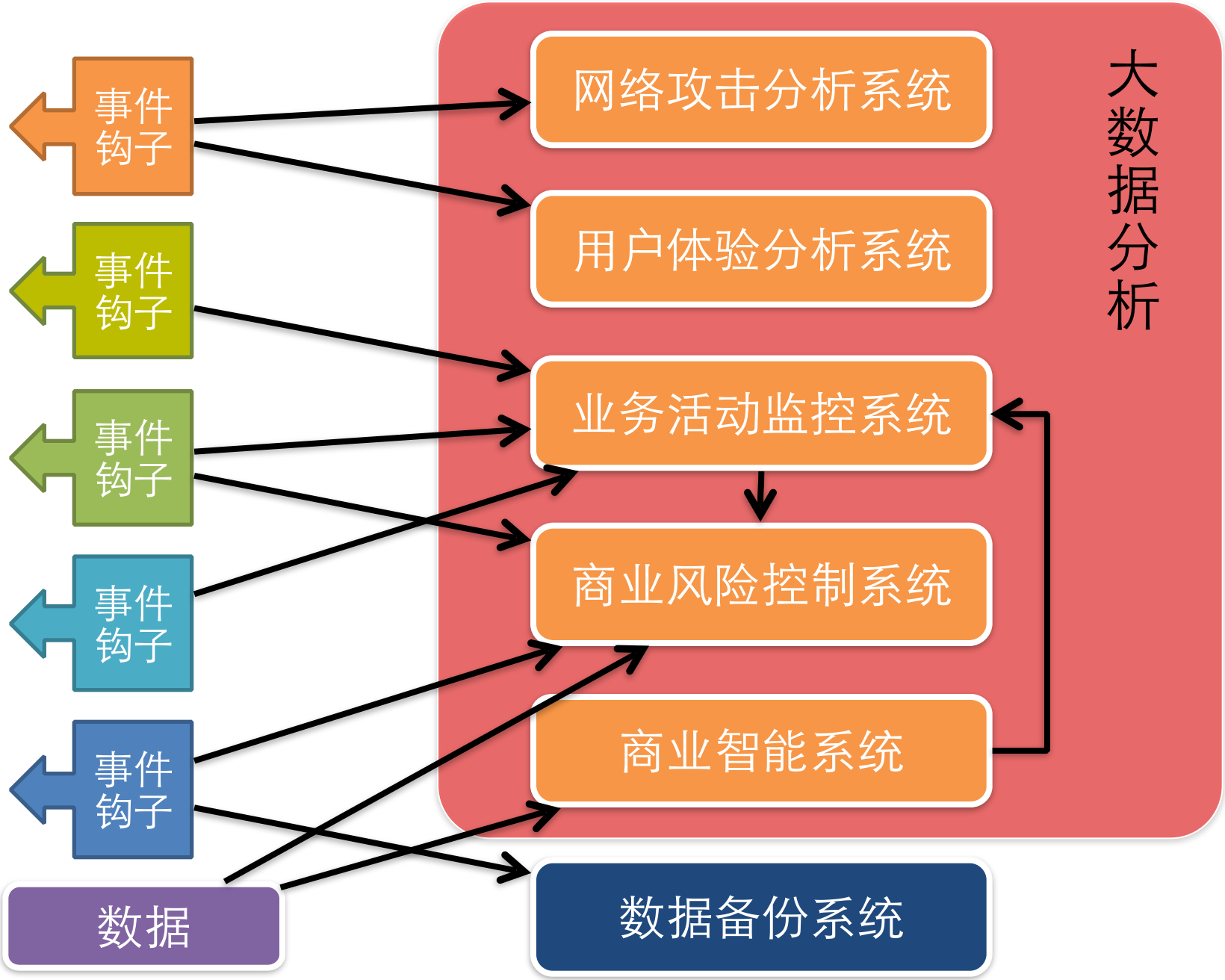
XZ 座标和云平台的关系



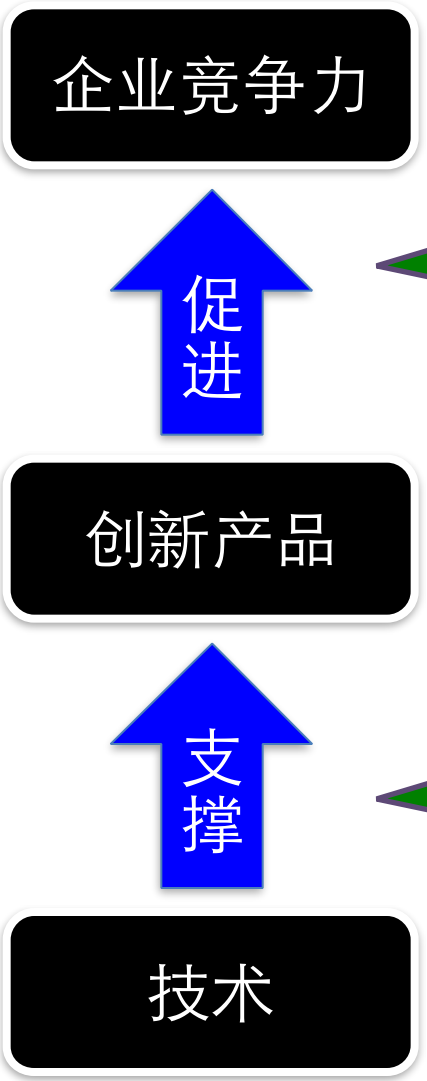
所有业务系统的服务层以下都要收归集团统一管理。优点：

- 只有内部（服务以下就是内部）可以依赖
- 数据方便管理
- 方便管理与外部系统的关系
- 可以逐渐形成统一平台

事件信息流向与数据分析系统



产品创新需要技术支撑

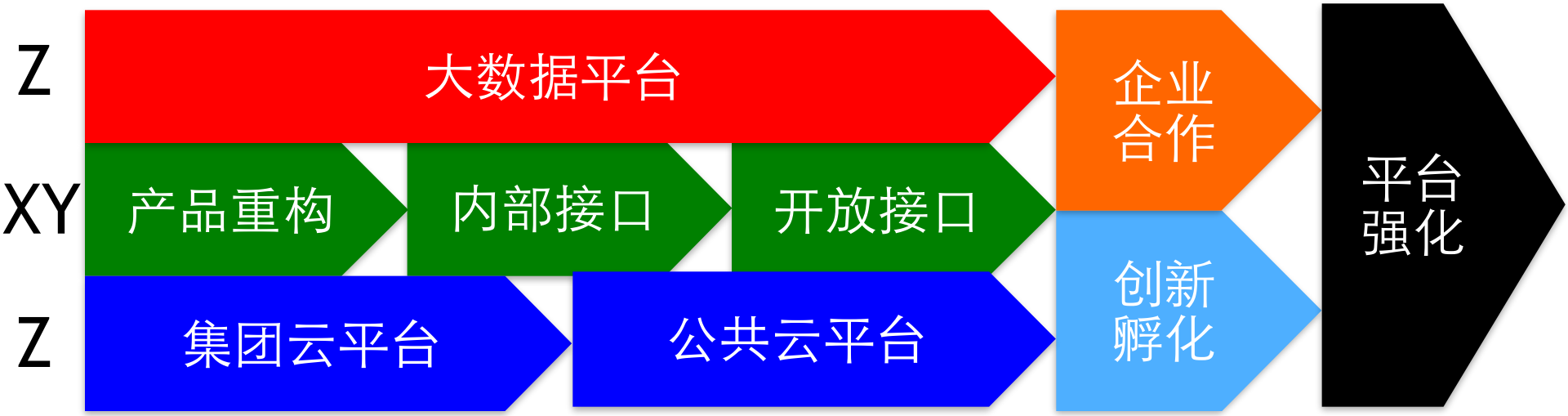


唯有持续推出创新的产品，才能维持企业竞争力

但…

滞后的技术与架构，无法支撑产品的创新

平台化促进合作与创新，进而巩固平台实力



蔡学镛 软件架构设计
Copyright © 2014 - 2015 Jerry Tsai (蔡学镛)
JerryTsai1218@Gmail.com
微博 @蔡学镛

Thank You