

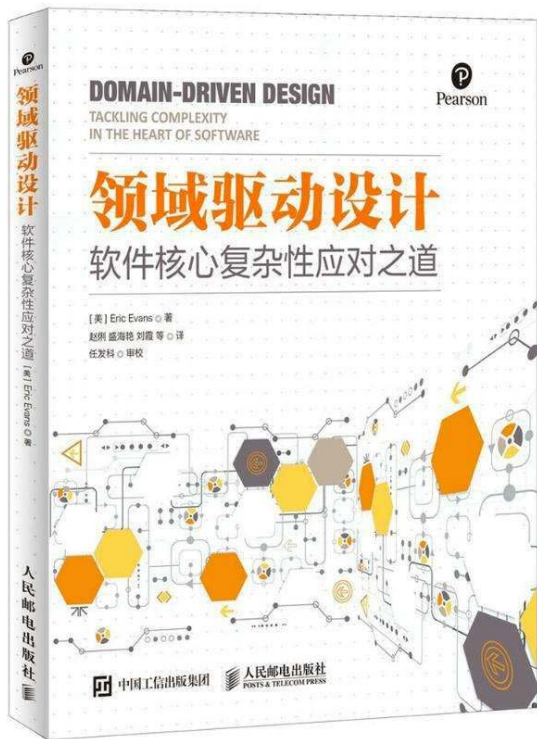
# DDD (Domain Driven Design) 的精髓



张建飞

阿里巴巴-新零售-零售通技术部

扫码关注【阿里巴巴云原生】公众号  
获取第一手技术干货



DDD的精髓是**应对软件复杂性**。

史蒂夫·迈克康奈尔：**软件的首要技术使命管理复杂度**。

**第一部分：DDD的复杂性应对之道**

**第二部分：基于DDD的应用架构-COLA**



统一语言：统一语言（概念和语义），可以降低认知成本。

巴门尼德对你说：存在者存在，非存在者不存在。

你对巴门尼德说：组合优于继承，约定胜于配置。

Stock

Inventory

Amount

库存

З а П а  
С Ы

# 统一语言：“一个团队，一种语言”

业务概念

PRD文档

日常沟通

设计文档

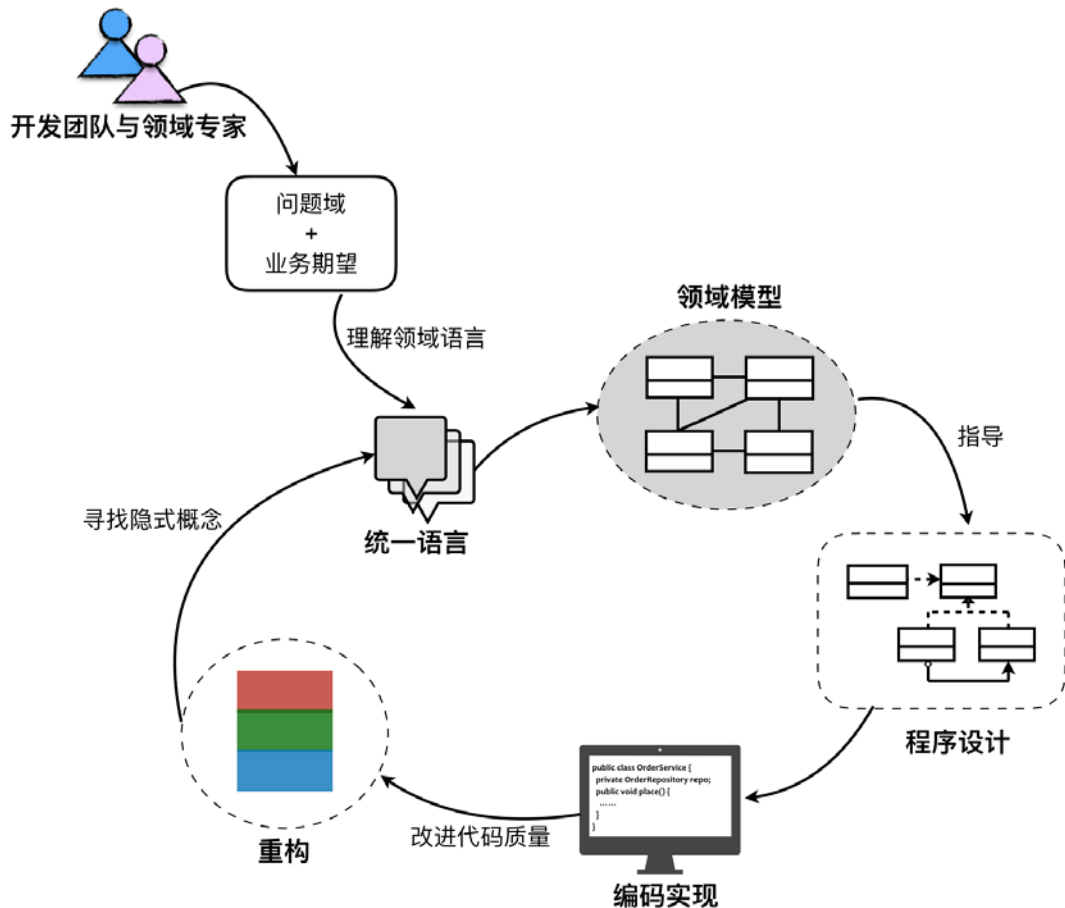
代码



## 核心领域词汇表

中文名	英文名	含义
建议零售价	Suggest Retail Price	建议小店对外零售的价格
供货价	Supply Price	商家设置的供货价格（也就是批发价）
红价	Red Price	买家在OfferDetail上看到的商品价格
成交价	Deal Price	实际成交价 = 红价 - 优惠金额

# 统一语言：是一个不断理解业务纵深，不断迭代的过程



## 统一语言：命名规范（一致性可以降低复杂度）

### 后置限定词

把限定词放在名词最后

- revenueTotal （总收入）
- expenseTotal （总支出）
- revenueAverage （平均收入）
- expenseAverage （平均支出）

### 事件命名

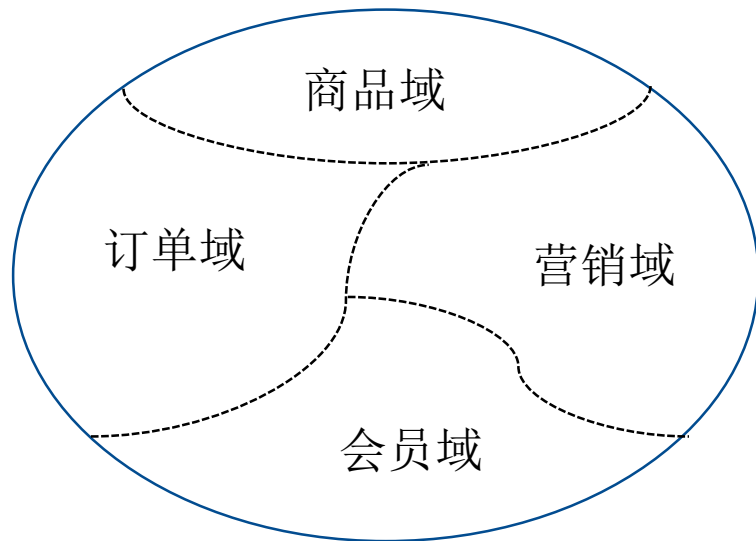
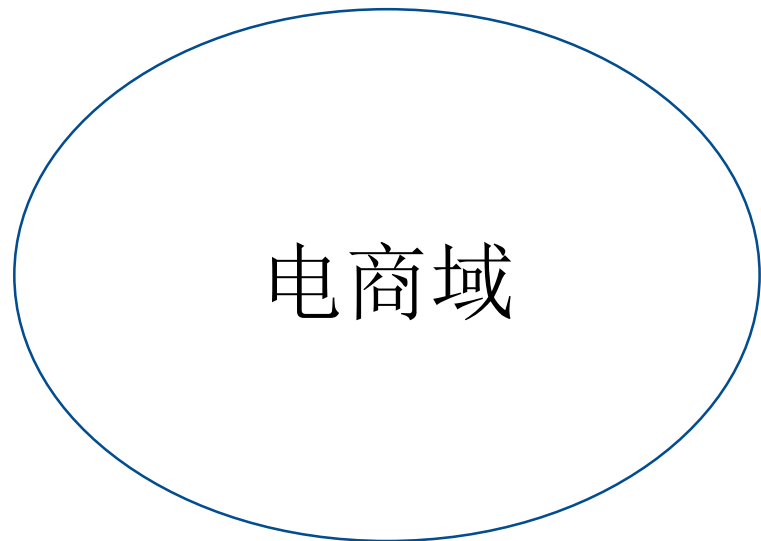
Domain Name + 动词的过去式 + Event

- CustomerCreatedEvent （创建客户事件）
- OrderPlacedEvent （下单事件）
- OrderPaidEvent （订单支付事件）

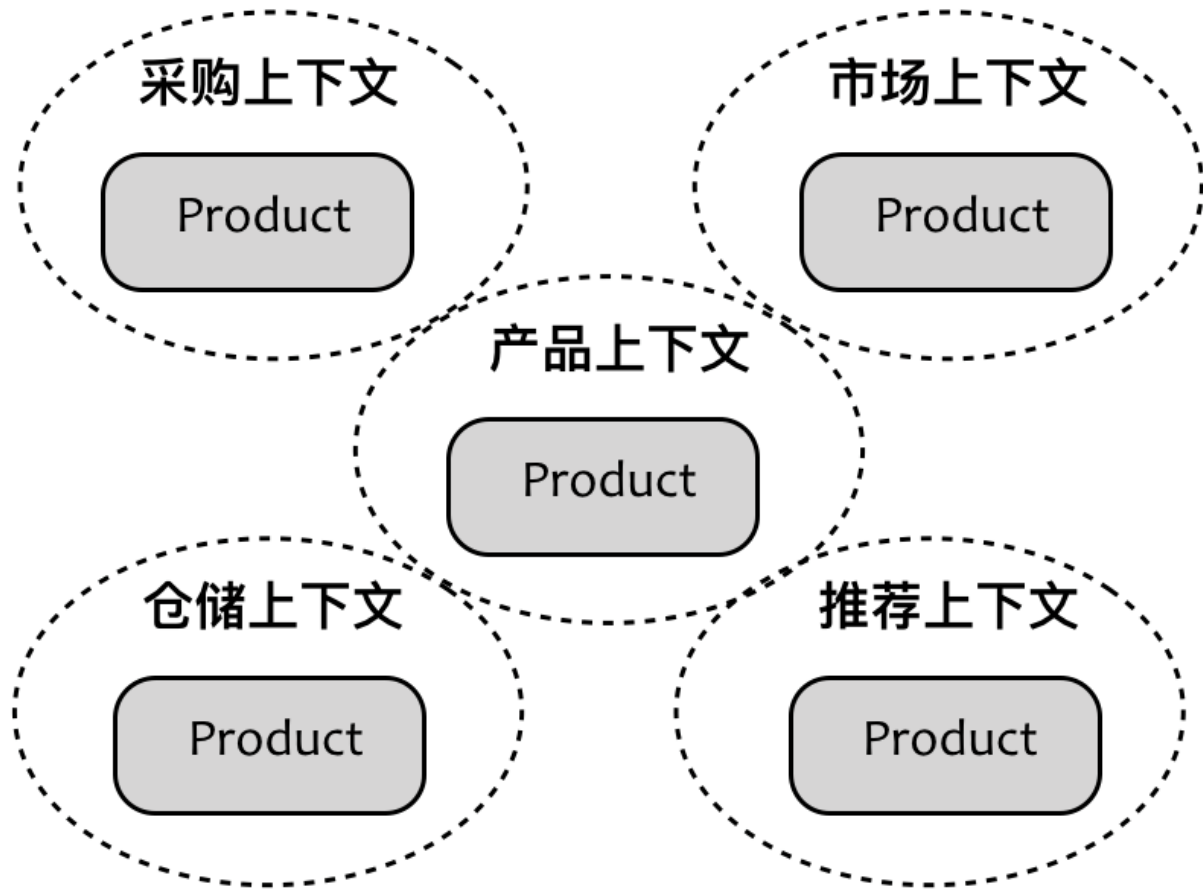




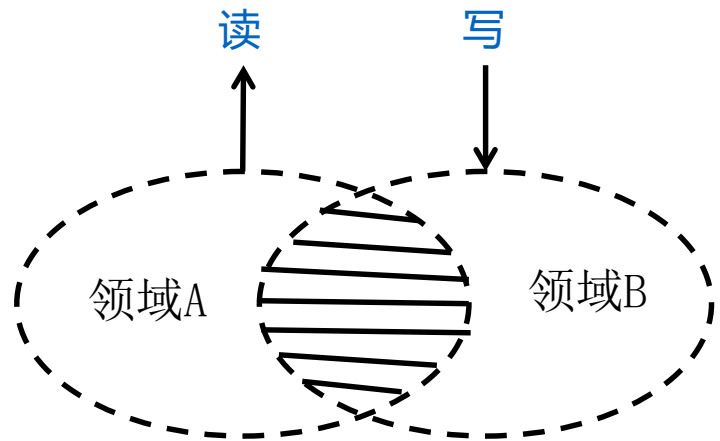
# 领域划分：分而治之



## 边界上下文 (Bounded Context)

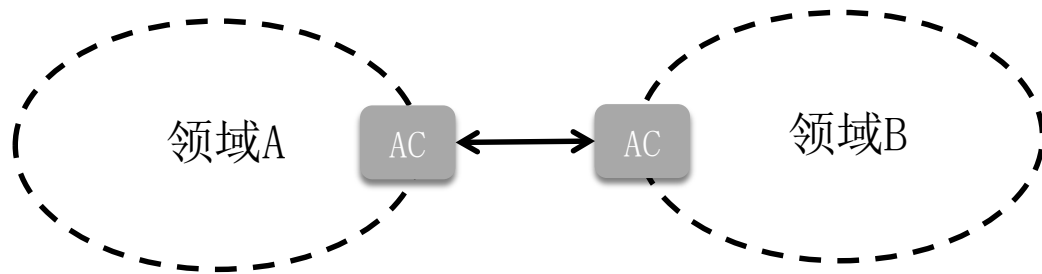


# 上下文映射 (Context Mapping)



## 1、共享内核 (Shared Kernel)

- Pro: 复用高
- Con: 耦合高

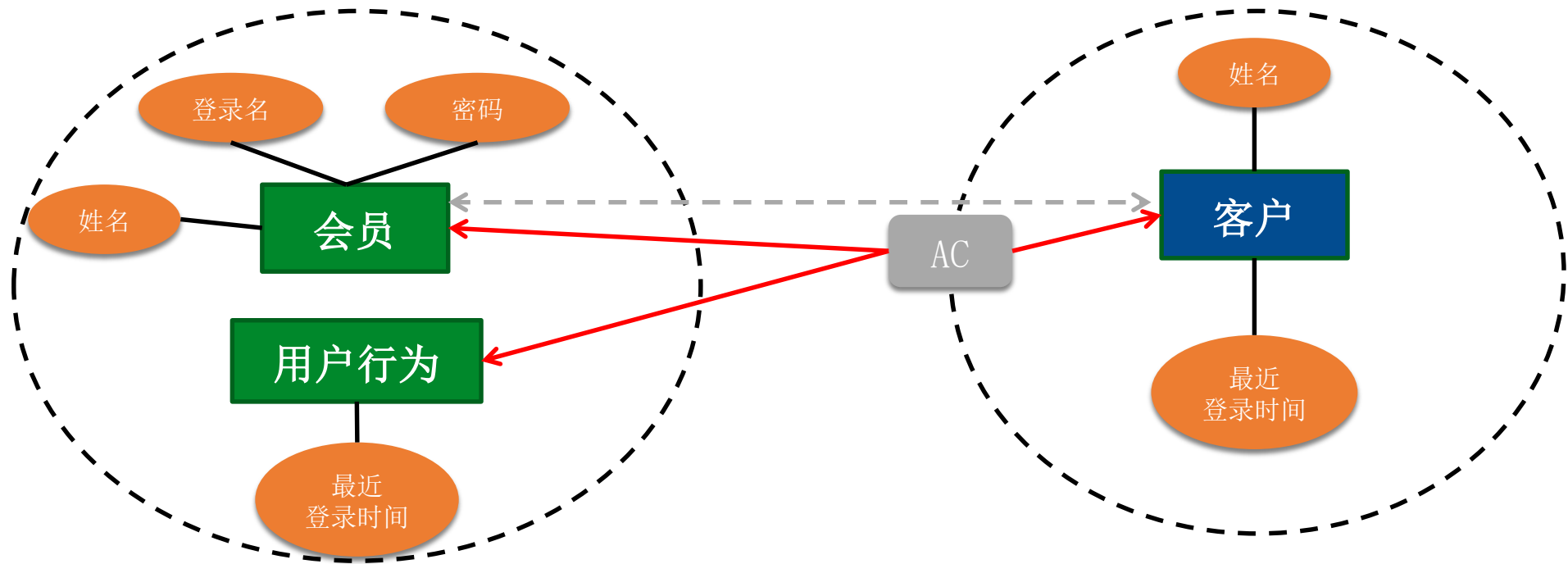


## 2、防腐层 (Anti-Corruption)

- Pro: 耦合低
- Con: 复用低

## 网站域

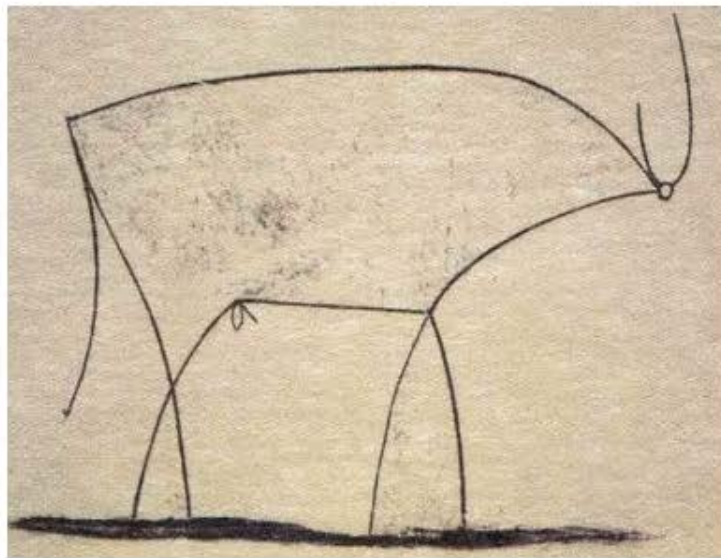
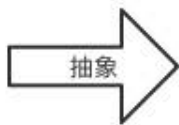
## CRM域







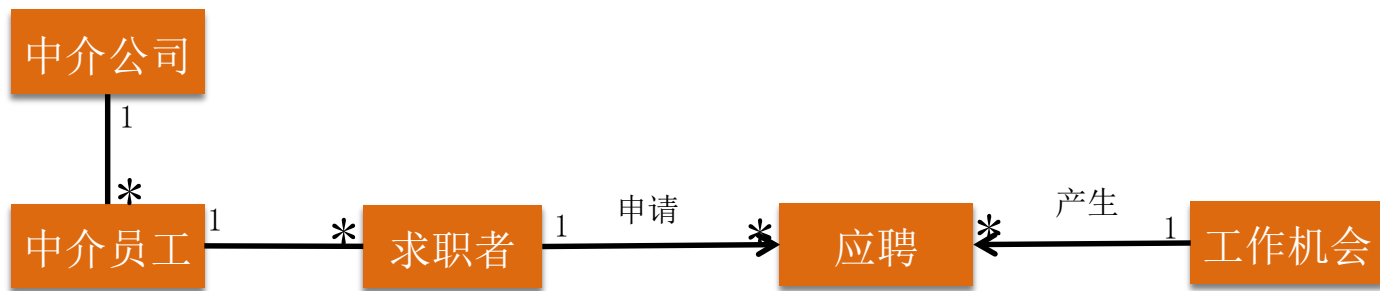
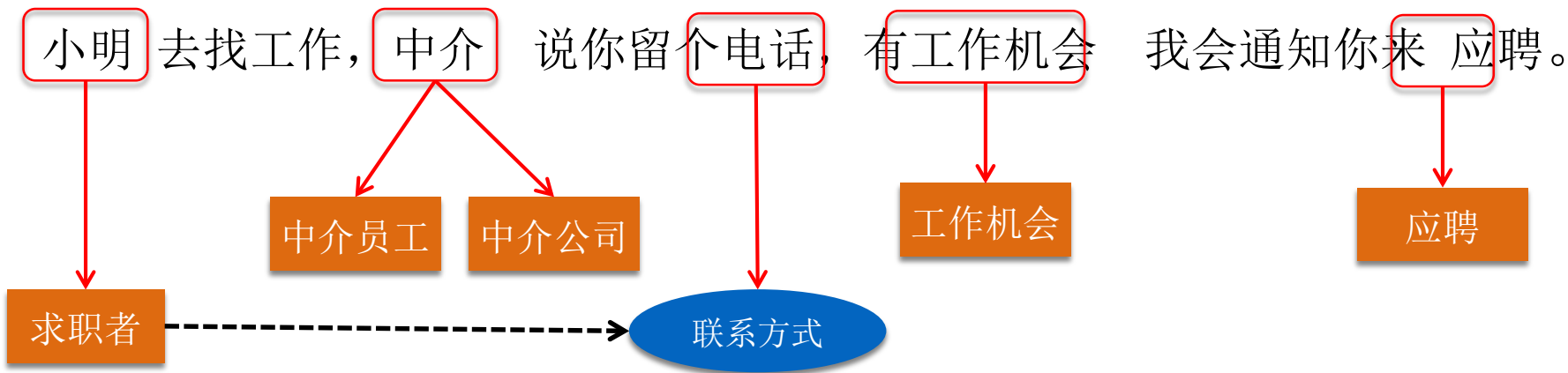
实体牛



抽象牛

**定义：“抽”就是抽取，“象”就是具象。是从具体事物抽取、概括出它们共同的方面、本质属性与关系等。抽象就是简化事物，抓住事物本质的过程。**

# 建模方法论：用例分析法

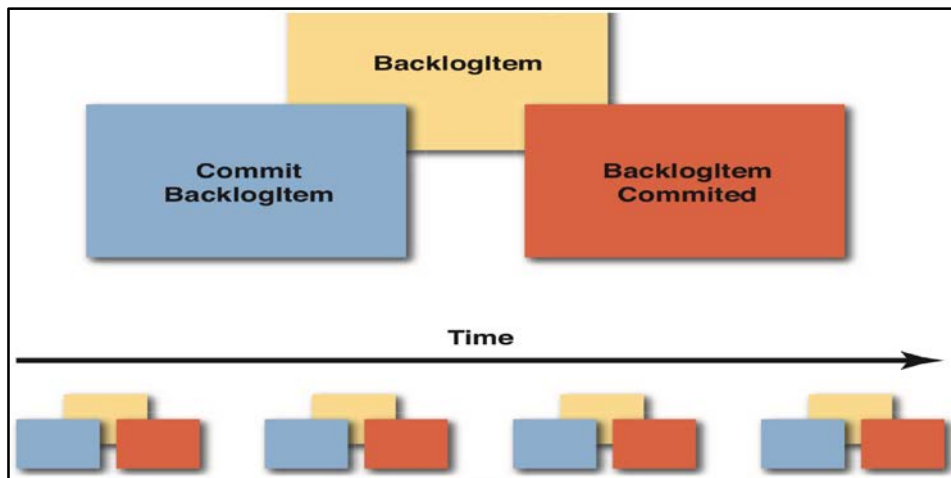




## 四色建模法



## 事件风暴



领域建模：建模也是一个不断迭代和探索的过程。



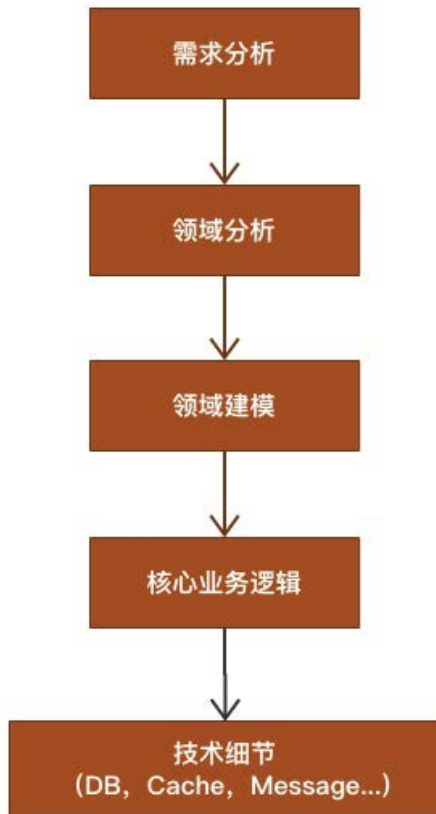
# 第二部分：基于DDD的应用架构-COLA

## 数据驱动

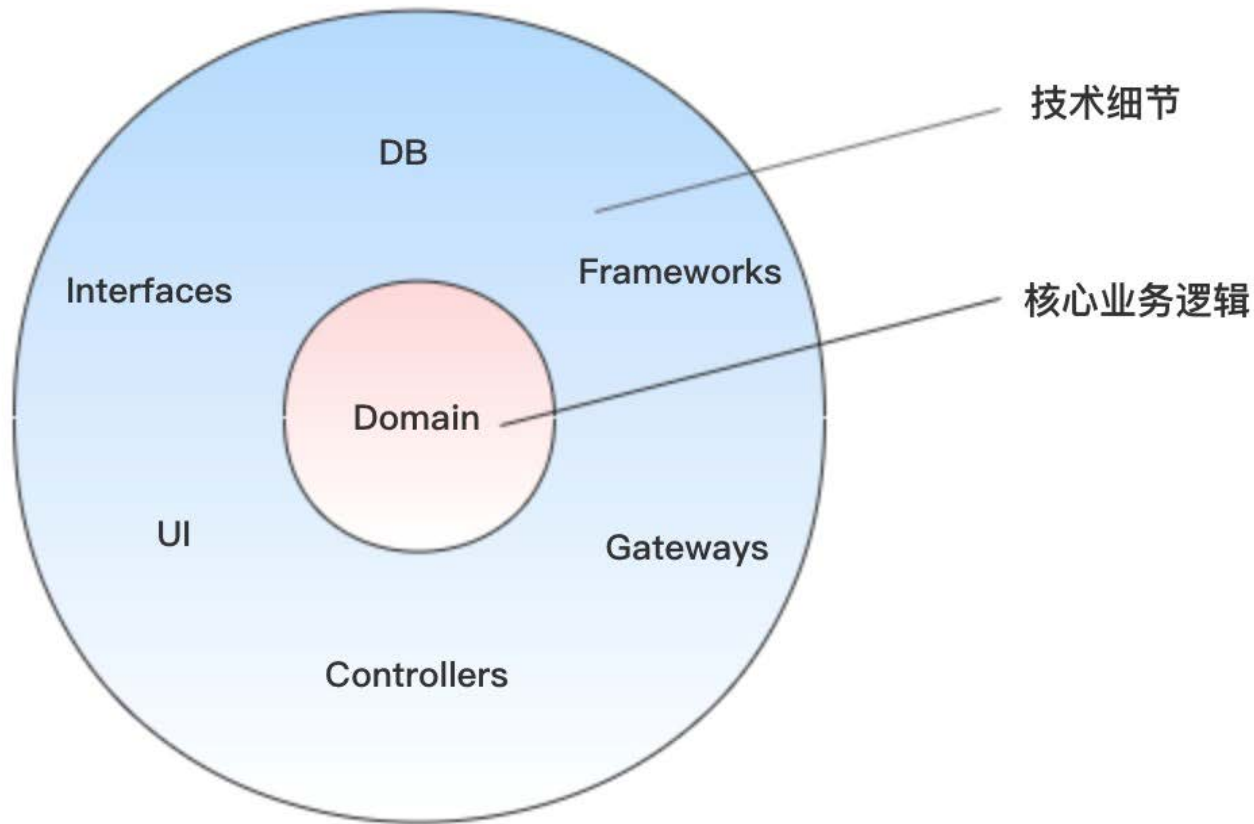


VS

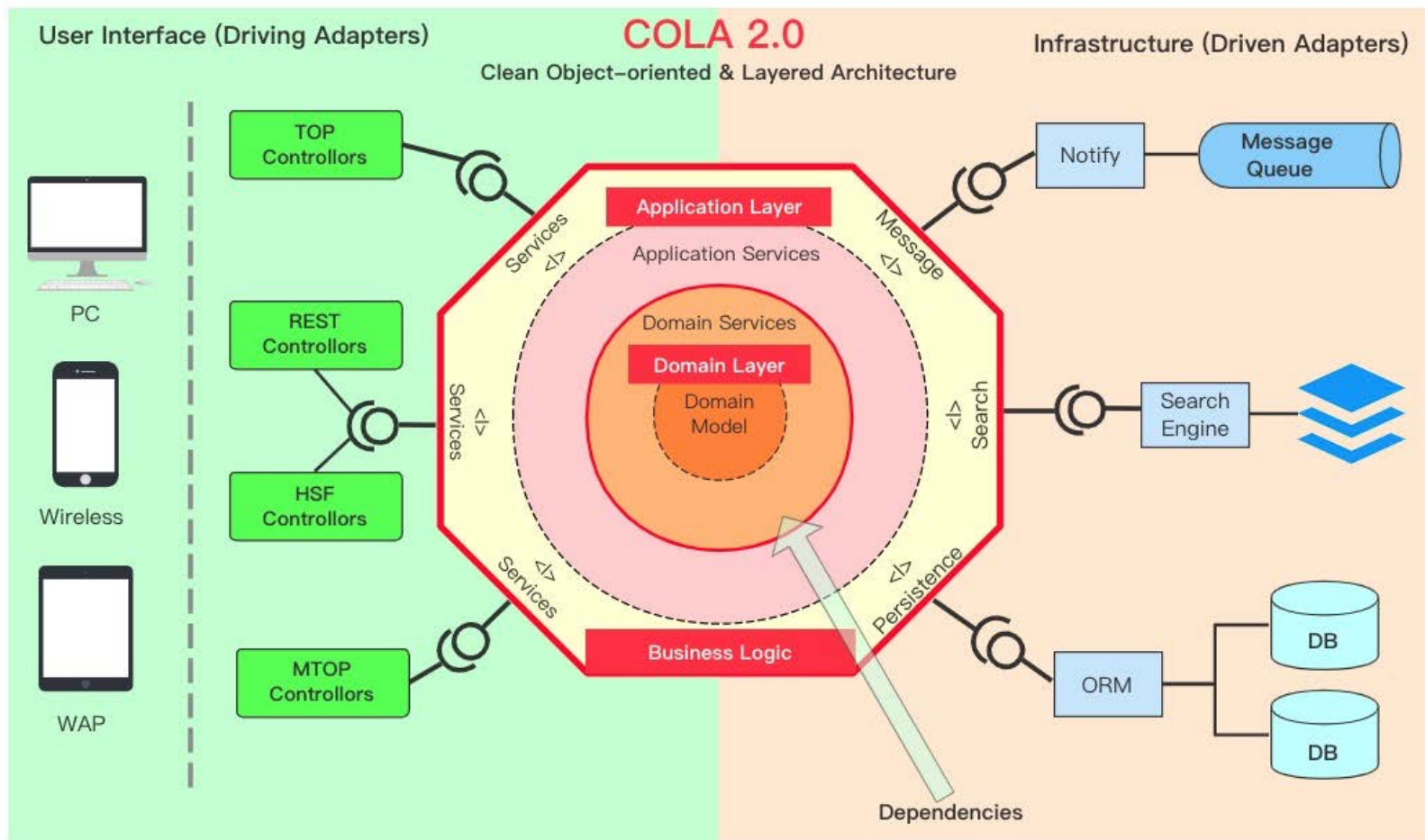
## 领域驱动



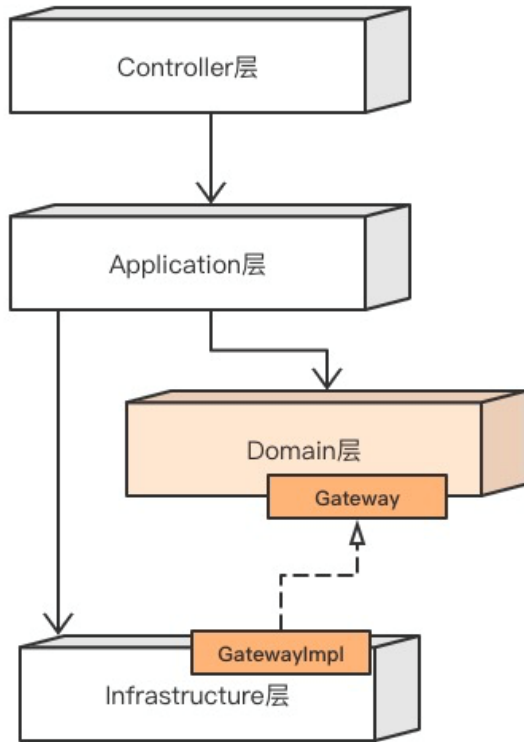
## 基于DDD应用架构的核心：分离业务复杂度和技术复杂度



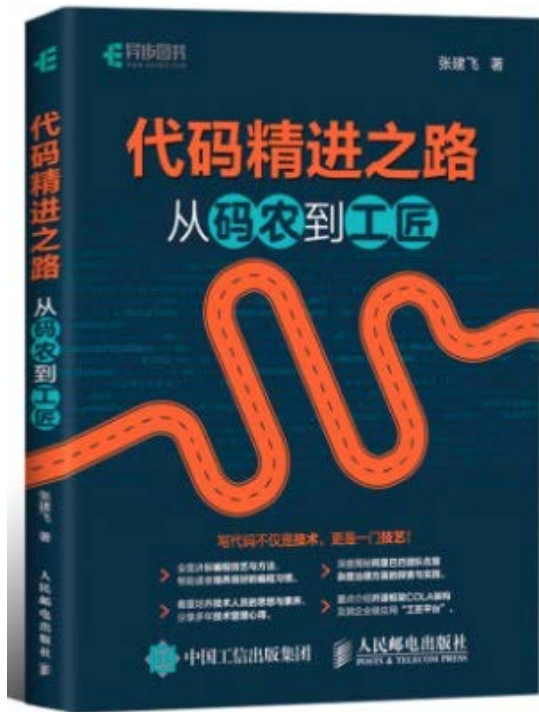
# DDD应用架构: COLA 2.0 (Clean Object-Oriented and Layered Architecture)



## DDD应用架构：使用COLA 2.0



<https://github.com/alibaba/COLA>



我的邮箱地址：[fulan.zjf@alibaba-inc.com](mailto:fulan.zjf@alibaba-inc.com)





奥运会全球指定云服务商



扫码关注【阿里巴巴云原生】公众号  
获取第一手技术干货