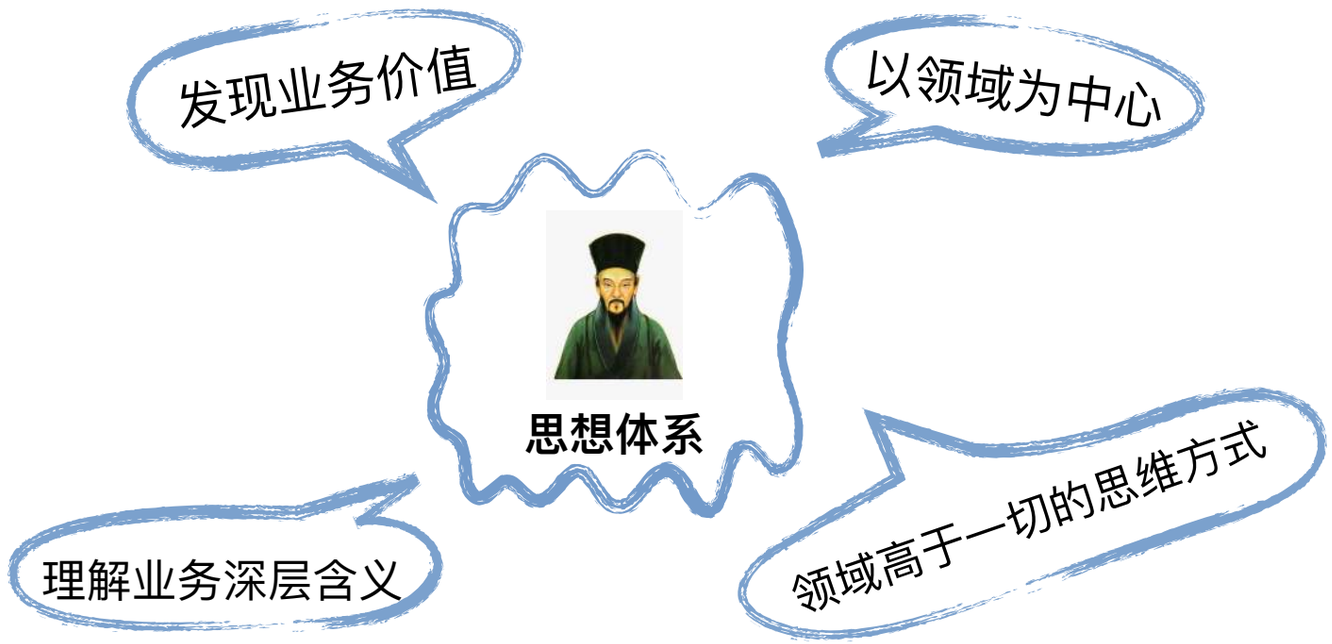
DDD在旅游电商架构演进中的实践

徐泼

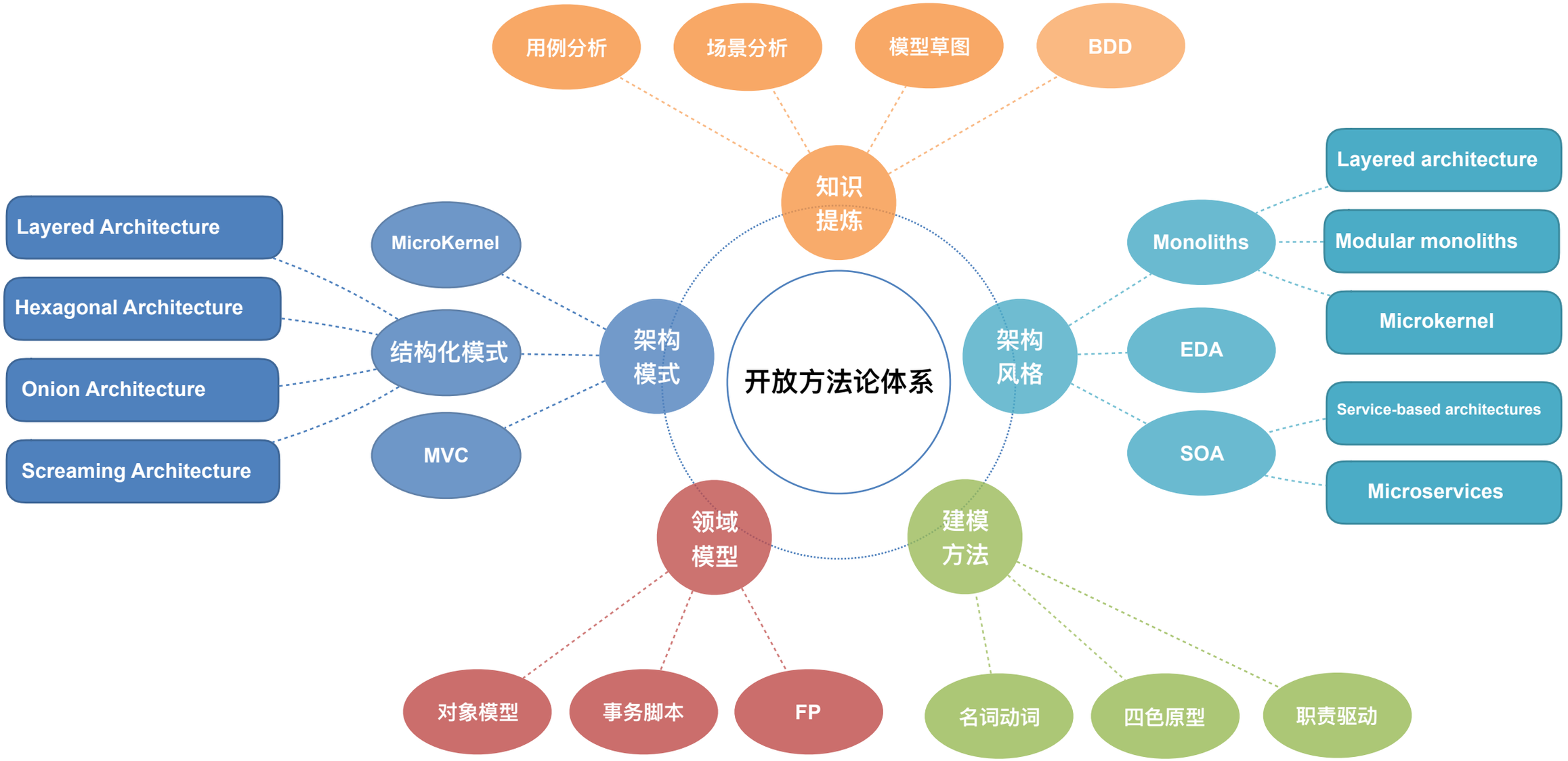
DDD CHINA

CONTENTS

领域驱动设计概述

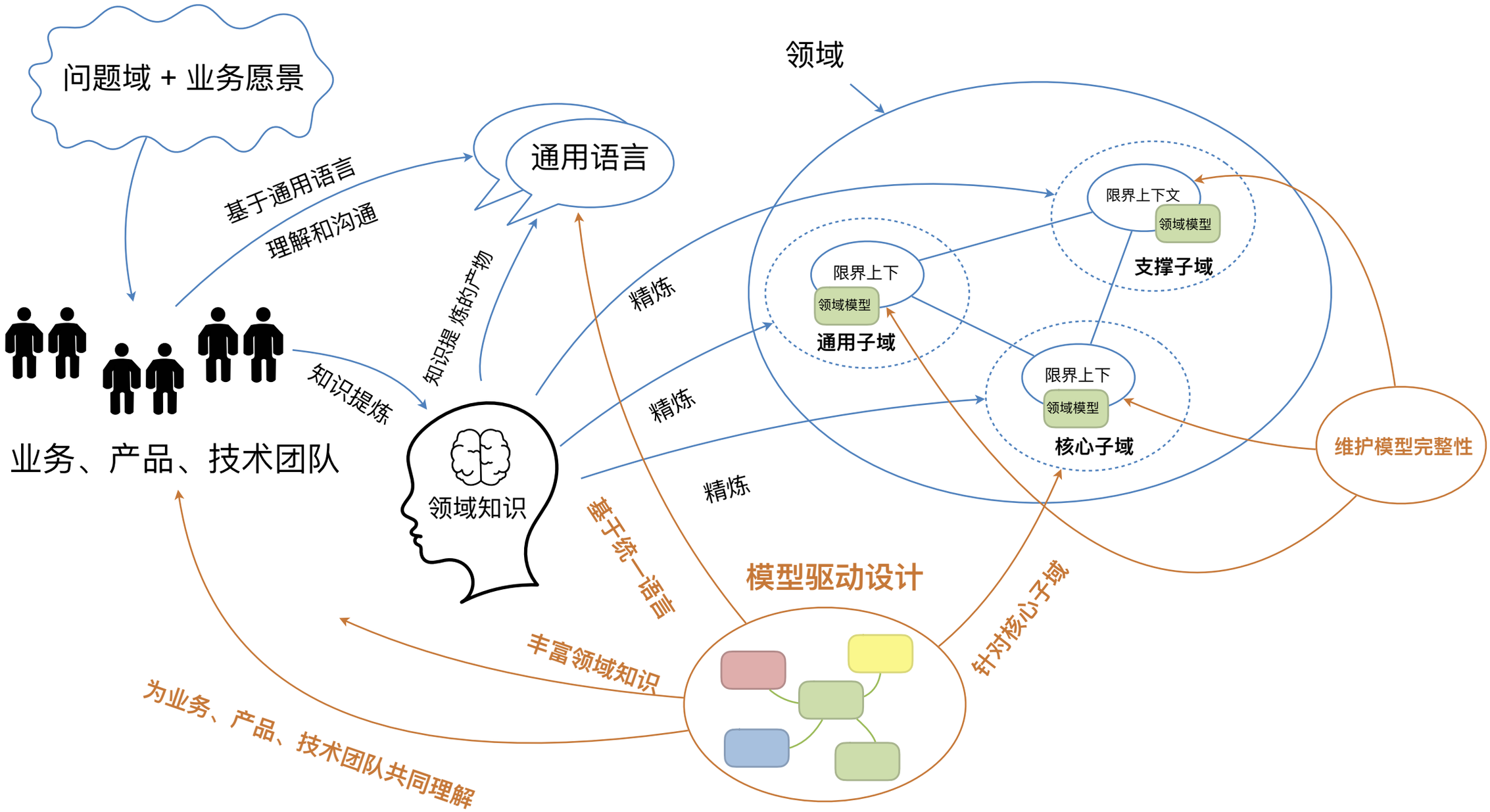


领域驱动设计和架构

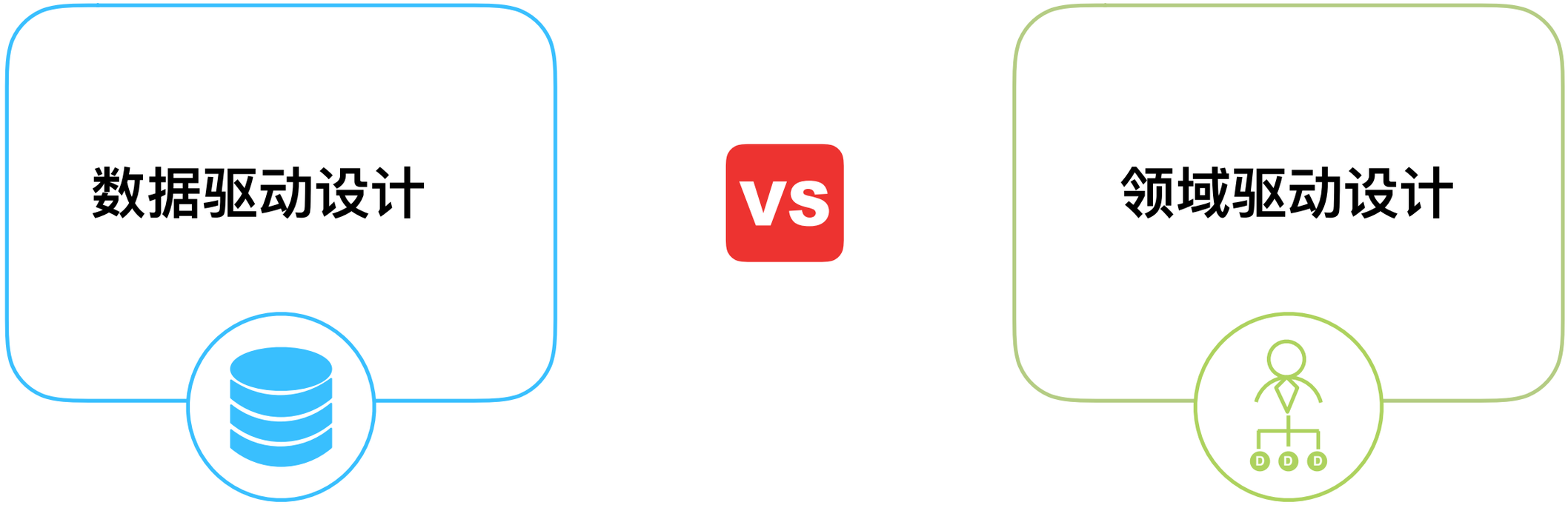


注：架构模式和架构风格的划分，一直是争论的热点，可以参考George Fairbanks和Pattern-oriented Software Architecture一书的观点

领域驱动设计过程


问题域 + 业务愿景
基于通用语言
理解和沟通
通用语言
知识提炼
知识提炼的产物
精炼
领域
限界上下文
领域模型
支撑子域
限界上下
领域模型
通用子域
限界上下
领域模型
核心子域
维护模型完整性
精炼
精炼
精炼
业务、产品、技术团队
领域知识
基于统一语言
模型驱动设计
针对核心子域
丰富领域知识
为业务、产品、技术团队共同理解

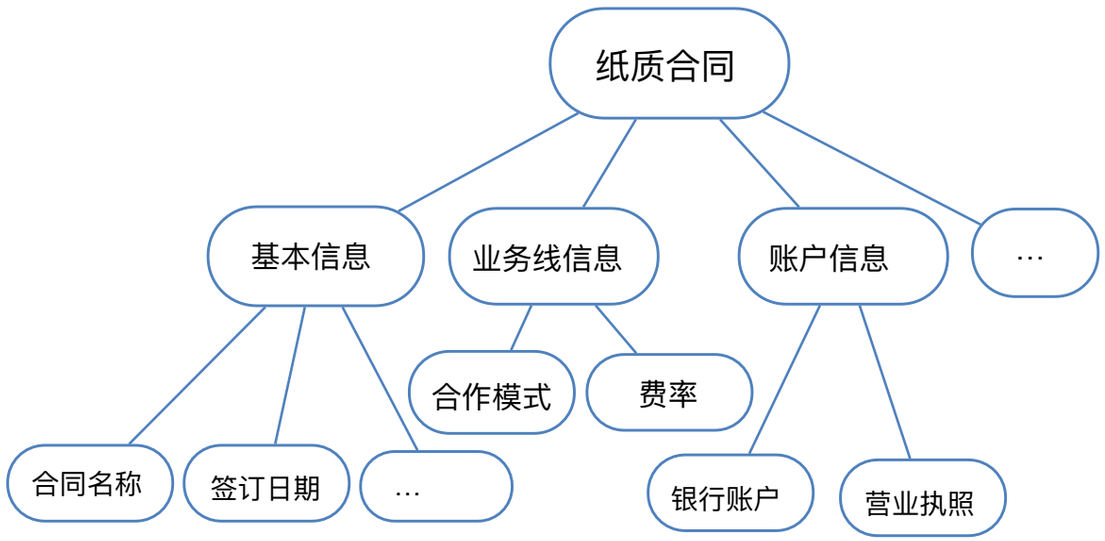
数据驱动 VS 领域驱动



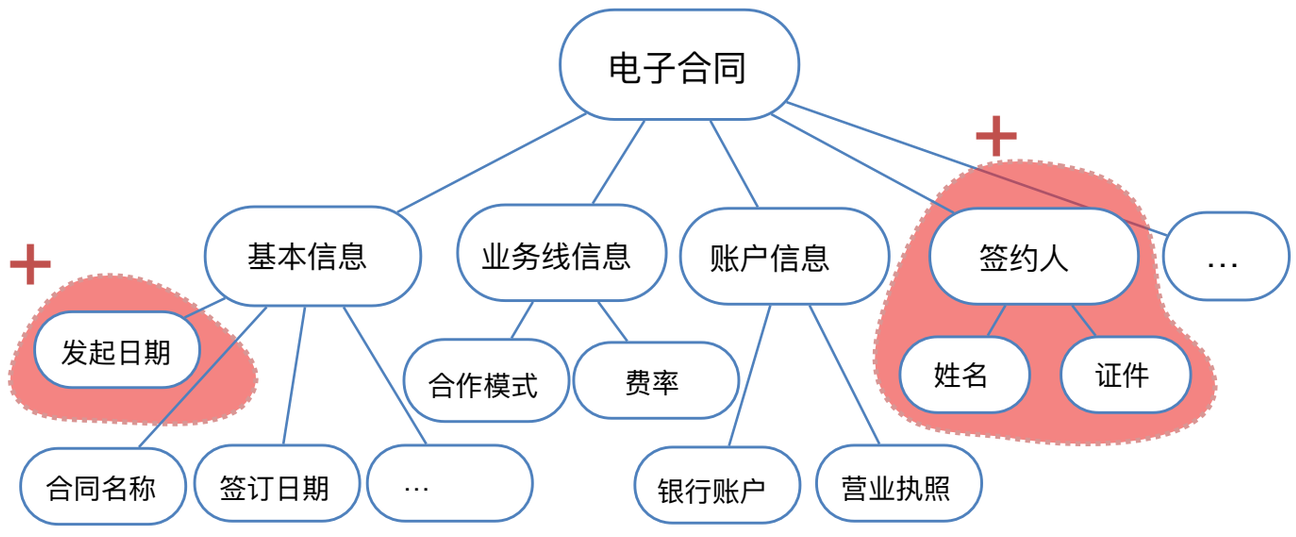
注：这里的数据驱动主要指以数据以及数据关系为核心的系统建模方法

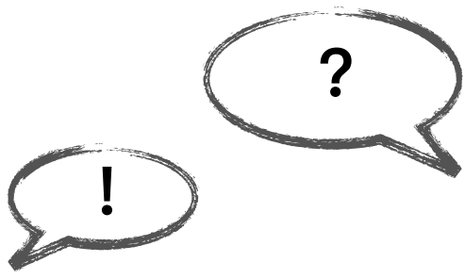
数据驱动 VS 领域驱动 - 数据分析

纸质合同管理



电子合同管理





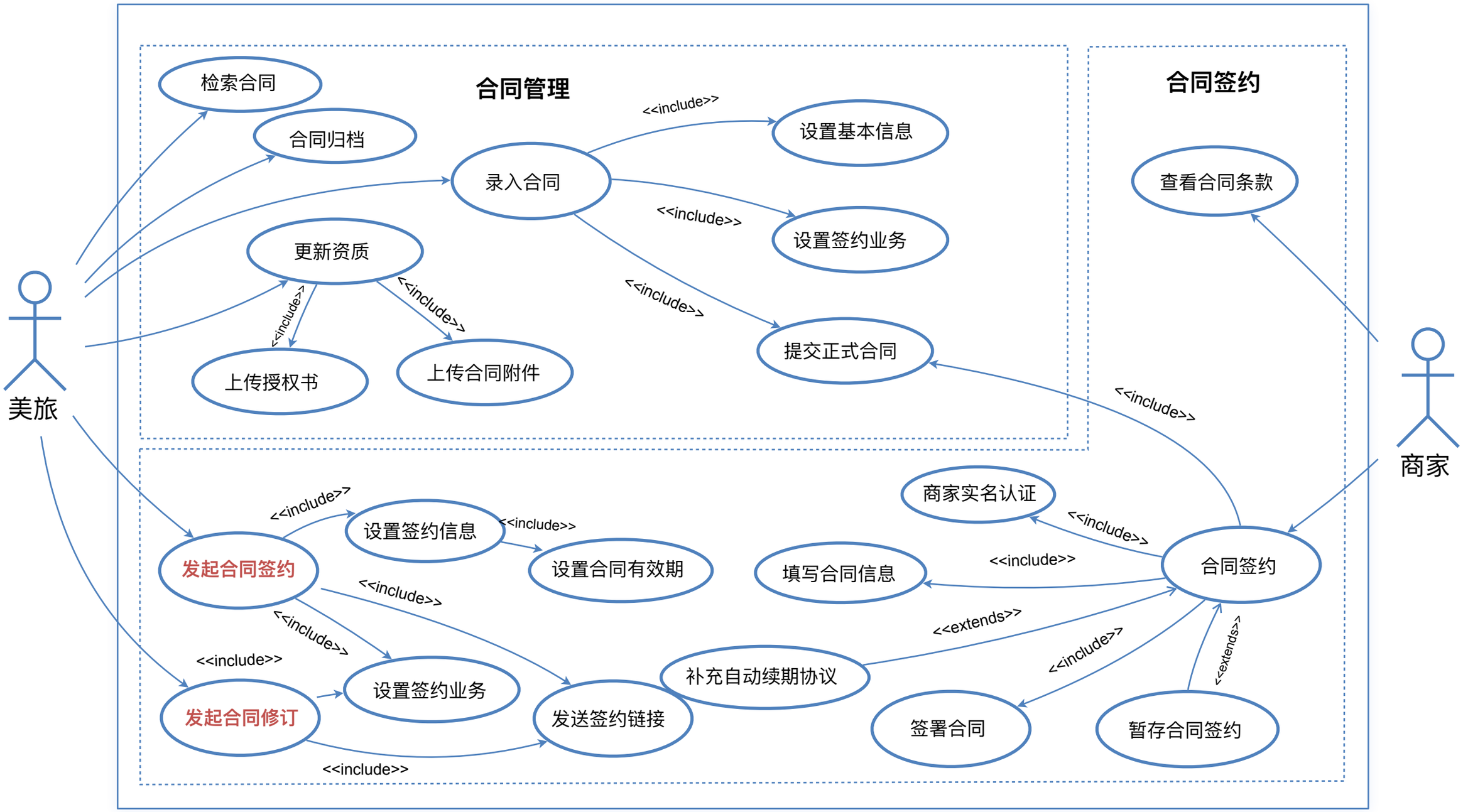
关注点：复用！复用！复用！

同一个数据模型处理纸质合同和电子合同的服务?

Software reuse is more like an organ transplant than snapping together Lego blocks.

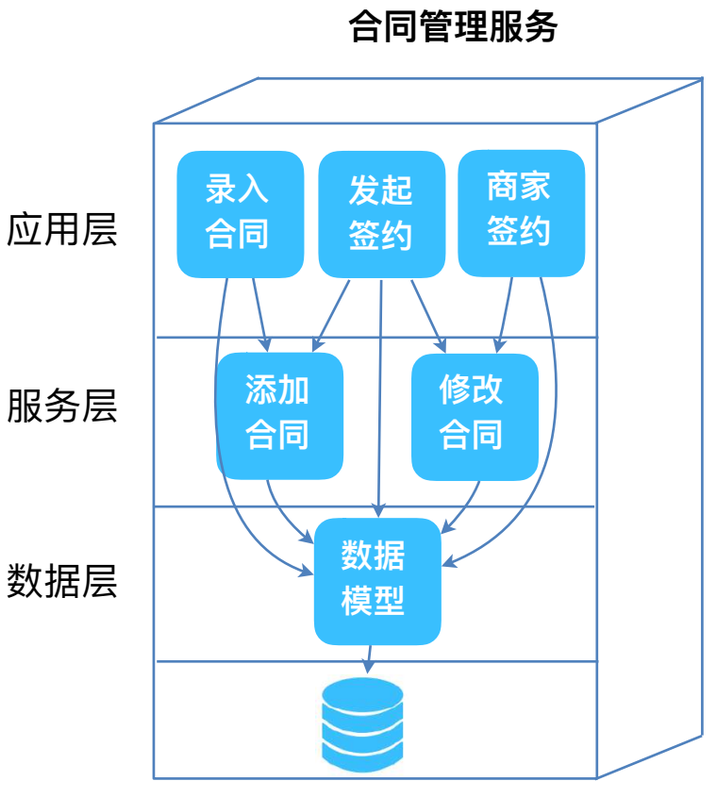
—John D. Cook

数据驱动 VS 领域驱动 - 用例分析

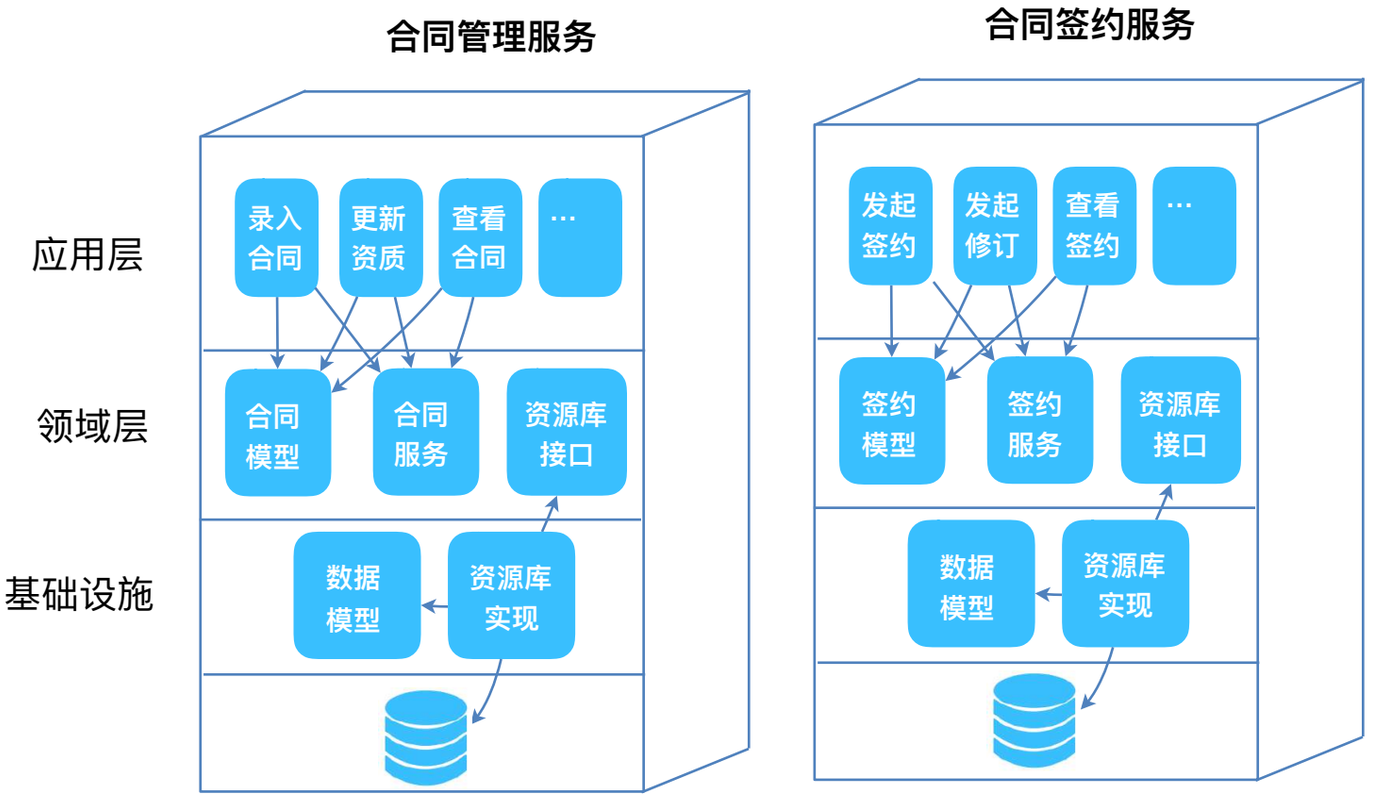


数据驱动 VS 领域驱动 - 设计结果

数据驱动设计结果

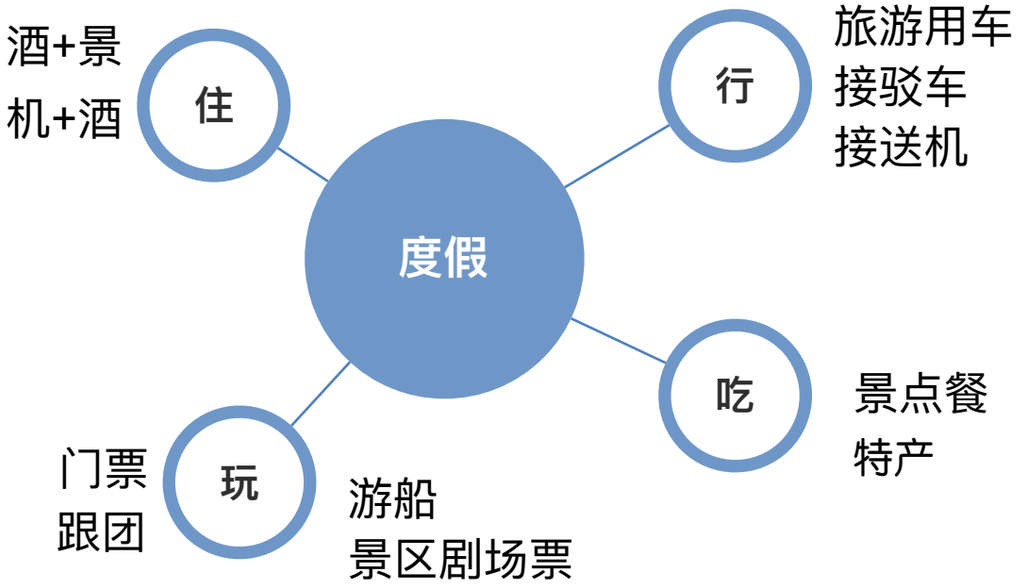


领域驱动设计结果



CONTENTS

旅游度假领域介绍



业务形态

- 购买
- 预订
- 团购
- 拼团
- 预约
- 预售
- …

业务模式

- 零售
- 代理
- 自营
- 分销

产品形态

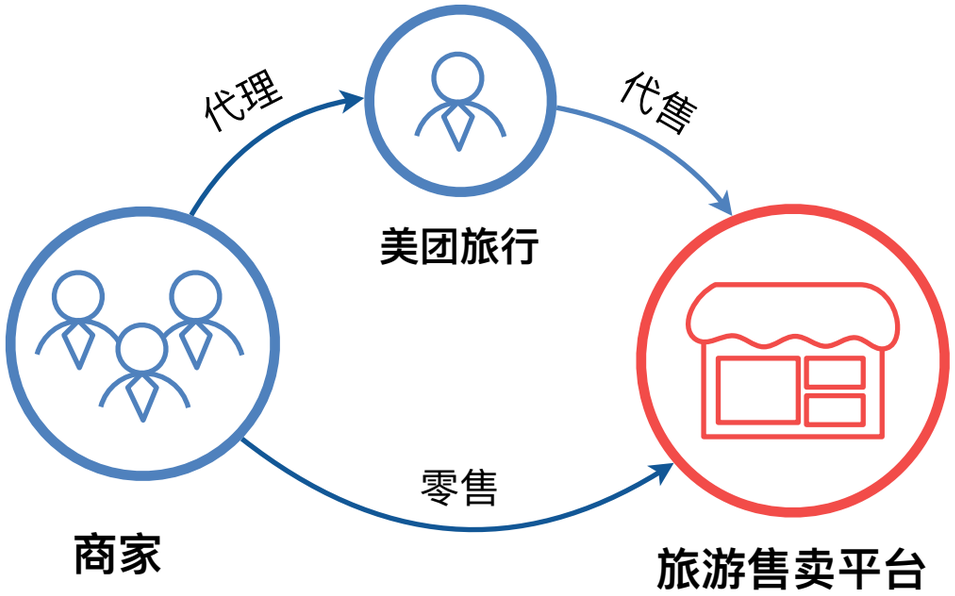
- 单品
- 预打死包
- 预打半活包
- 活包
- 搭售
- 购物车
- …

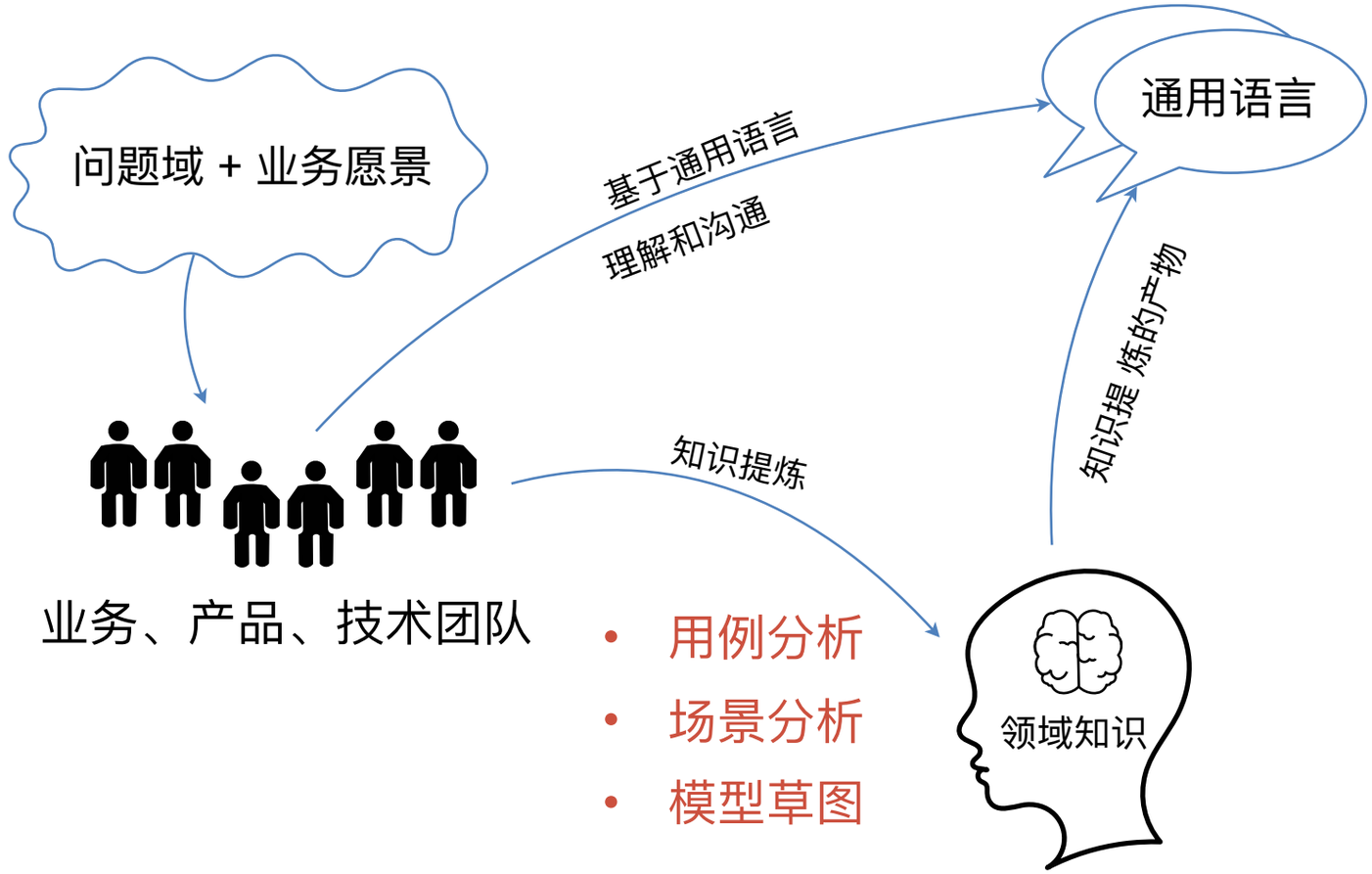
产品品类

- 门票
- 跟团
- 酒景
- 机酒
- 旅游用车
- 演出
- 特产
- …

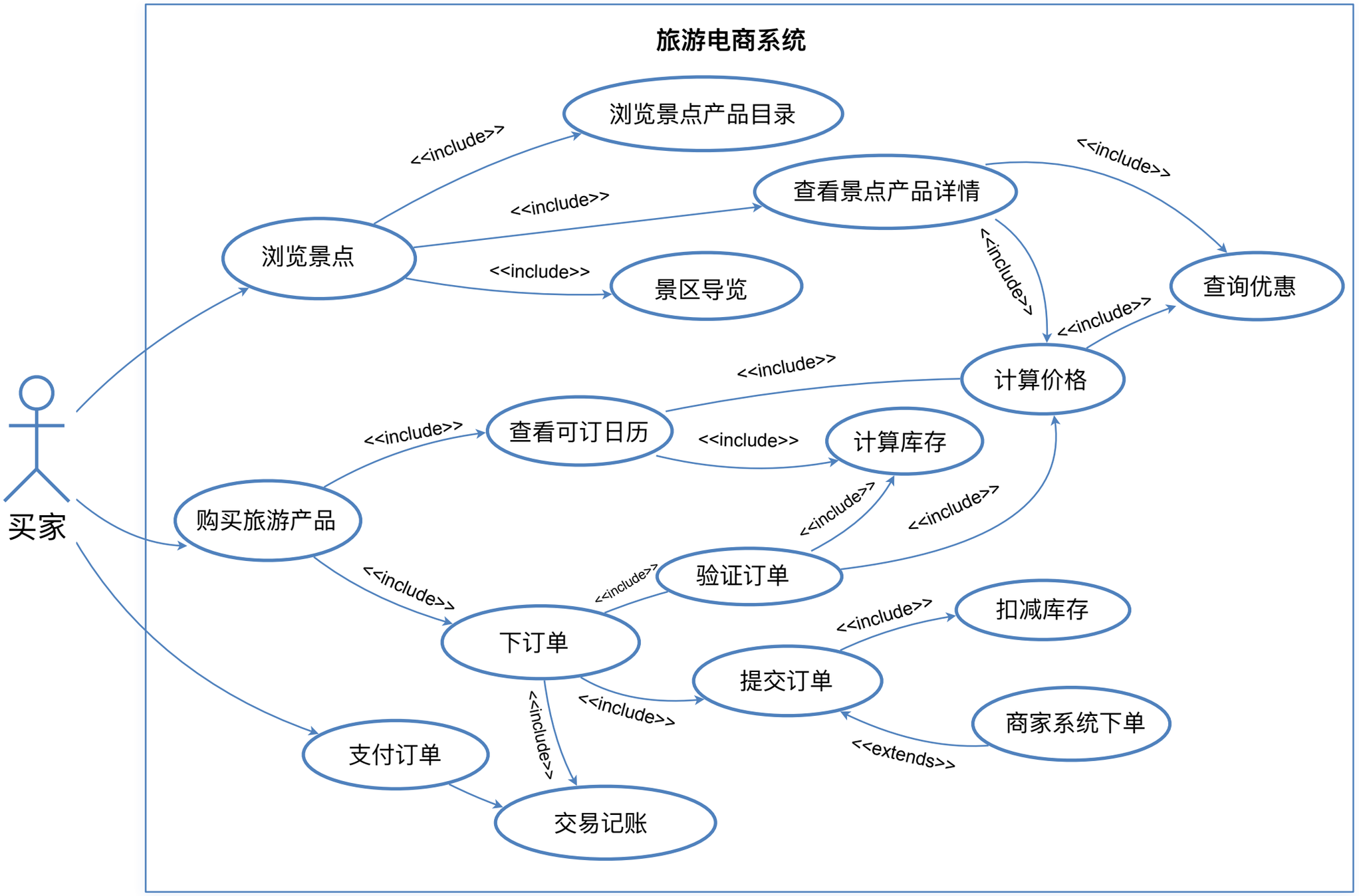
架构演进 - 阶段1 - 背景和知识提炼

- 为用户提供休闲度假的旅游产品购买和服务
- 为景区和旅游商家提供度假旅游产品一体化售卖平台



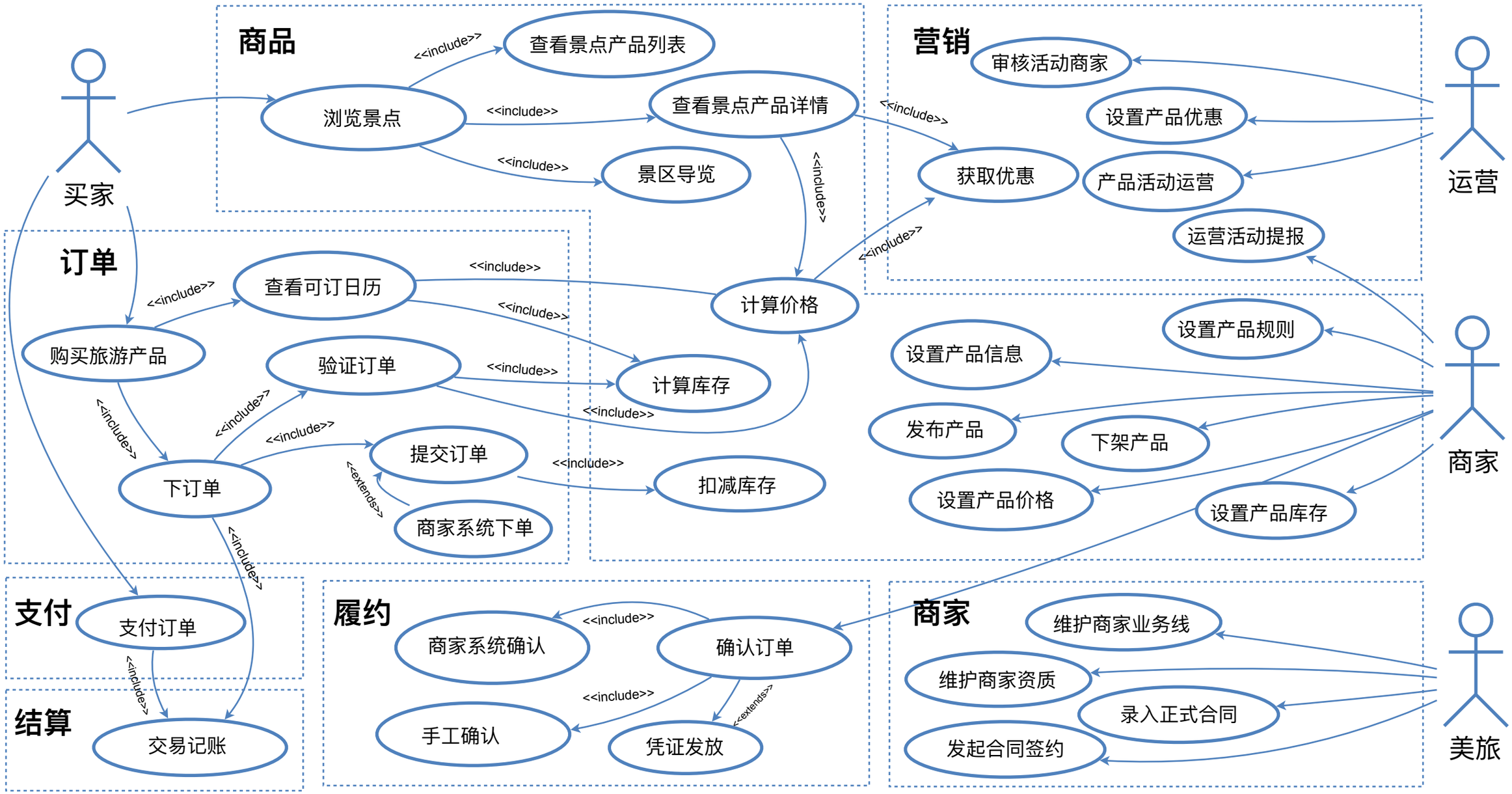


架构演进 - 阶段1 - 知识提炼 - 用例分析



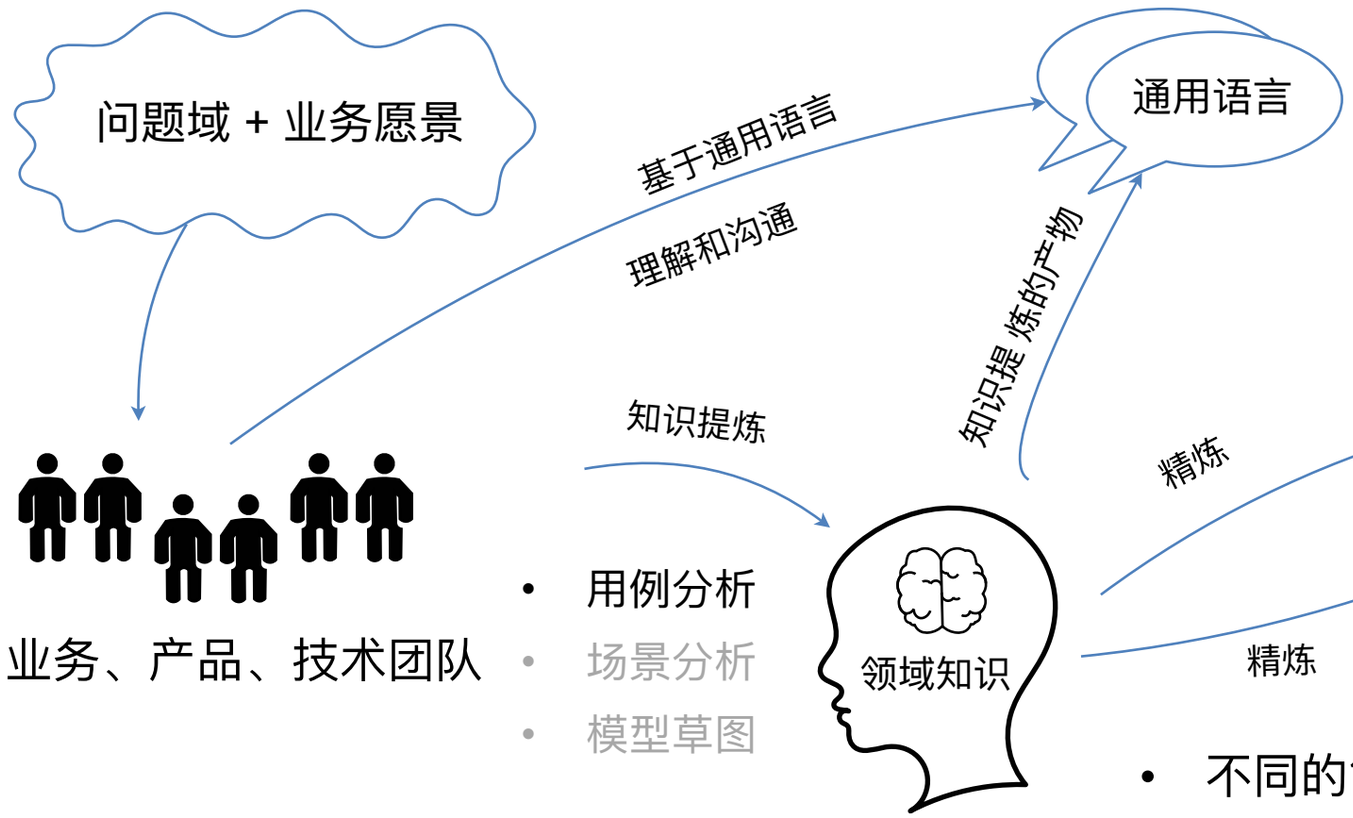
架构演进 - 阶段1 - 知识提炼 - 用例分析

架构演进 - 阶段1 - 知识提炼 - 用例分析

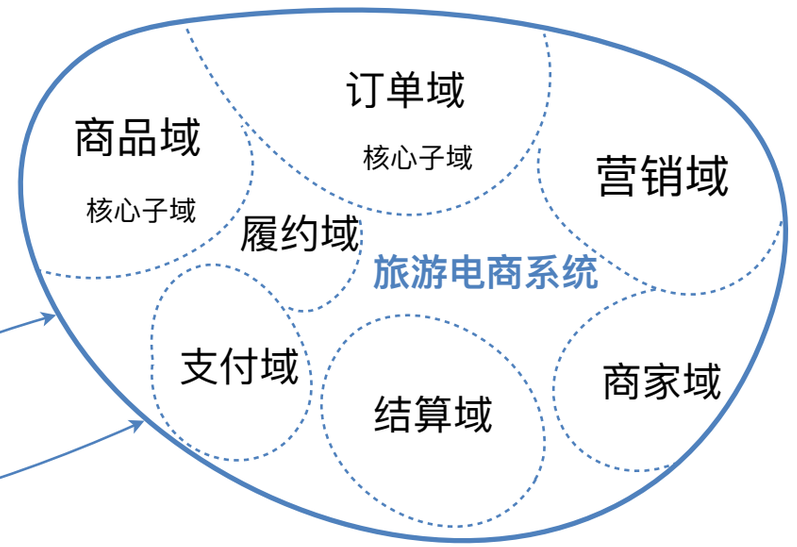


架构演进 - 阶段1 - 知识提炼 - 识别子域

- 为用户提供休闲度假的旅游产品购买和服务
- 为景区和旅游商家提供度假旅游产品一体化售卖平台



- 产品
- 订单
- 履约
- 优惠
- 买家
- 商家
- 支付
- 结算



- 不同的能力区域
- 不同的业务过程

- 用子域逻辑划分整个业务领域
- 理解大型复杂项目的问题空间

架构演进 - 阶段1 - 知识提炼 - 场景分析

跟团业务人员

用户提交跟团产品特定出团日期的成人购买数量、儿童购买数量
系统验证每单最多和最少购买限制，验证出团时间，验证抵用券信息，计算订单价格、验证剩余库存；验证订单通过后；生成订单；锁定库存；收到用户成功支付后，扣减库存。

门票业务人员

用户提交特定日期某票种的购买数量、系统验证每单最多和最少购买限制，验证手机号购买限制，验证入园时间；验证抵用券信息，计算订单价格、验证剩余库存；验证订单通过后扣减库存，并生成订单。等待用户支付结果消息。

酒景业务人员

用户提交酒景产品特定日期的某个套餐购买份数；
系统验证每套餐的最多和最少购买限制，验证套餐可使用时间；计算订单价格；验证剩余库存；验证订单通过后；生成订单；锁定库存；收到用户成功支付后，扣减库存。

下订单场景业务抽象

买家提交特定旅游产品**售卖单元**的购买数量；
系统验证数量**限购规则**和购买**时间限制规则**；
系统验证优惠信息；

计算订单价格、验证可用库存；

生成交易订单和订单项信息；

根据**库存扣减策略**操作所有订单项对应的售卖单元库存；

根据用户支付结果执行相关订单动作。

场景1：订单满足限购规则

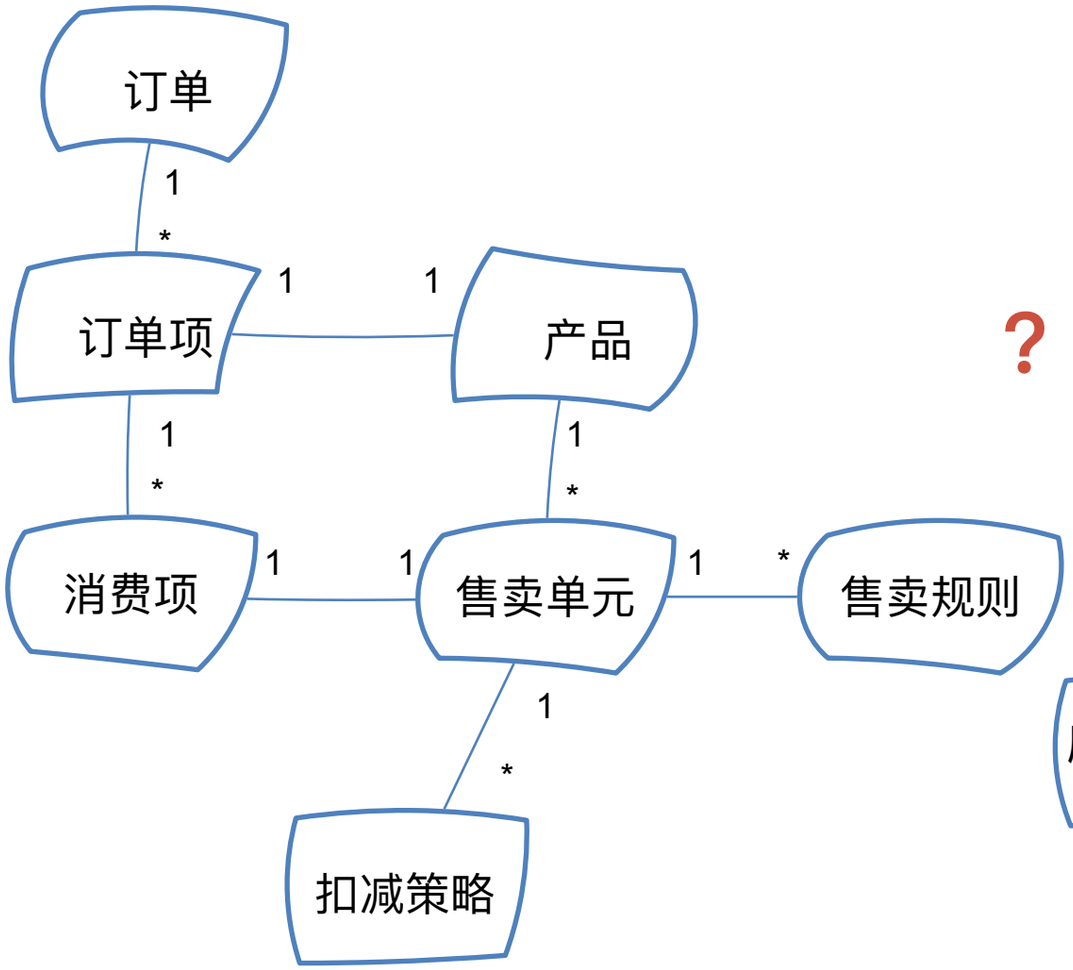
场景2：订单不满足限购规则

场景3：产品库存不足

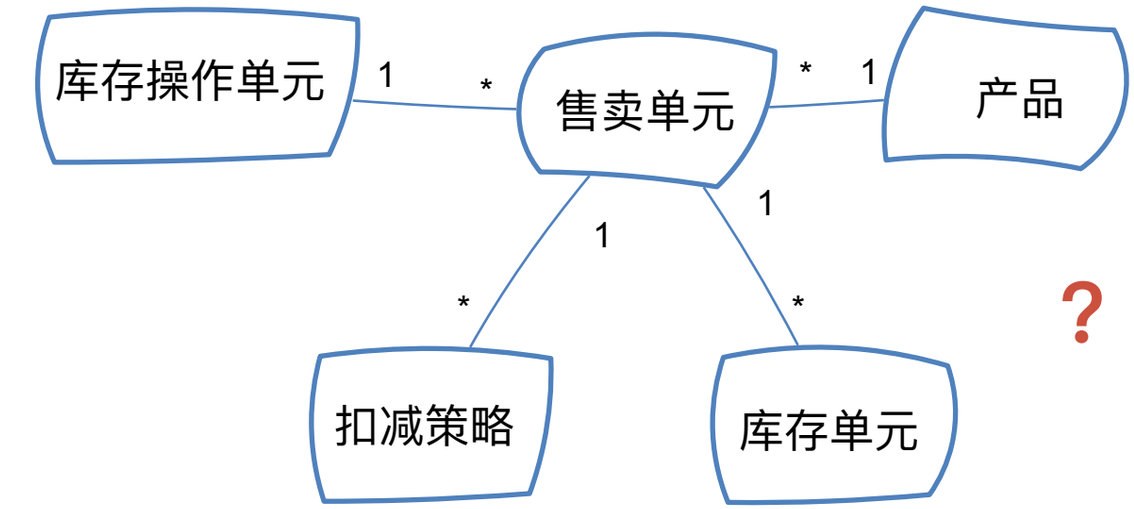
场景4：...

平台统一语言

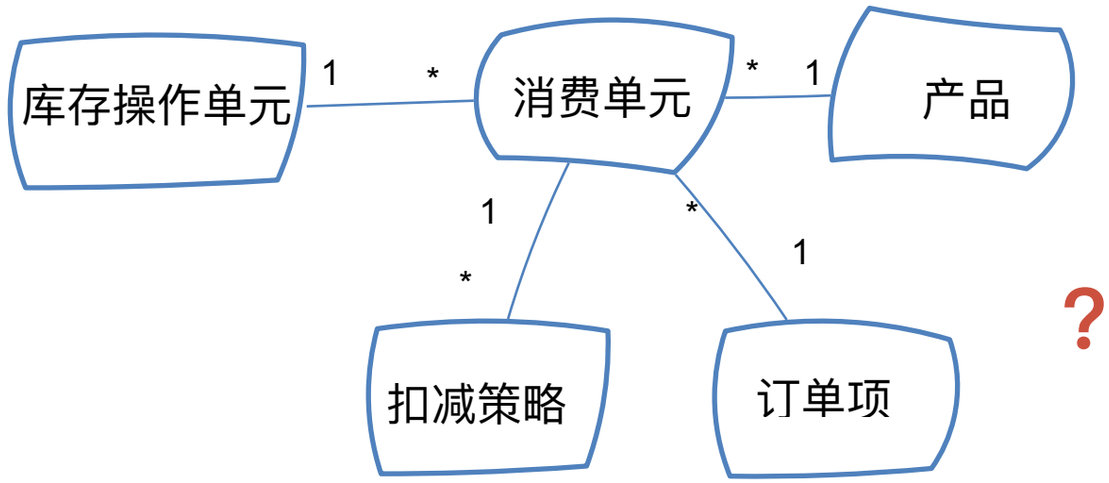
架构演进 - 阶段1 - 知识提炼 - 模型草图



?



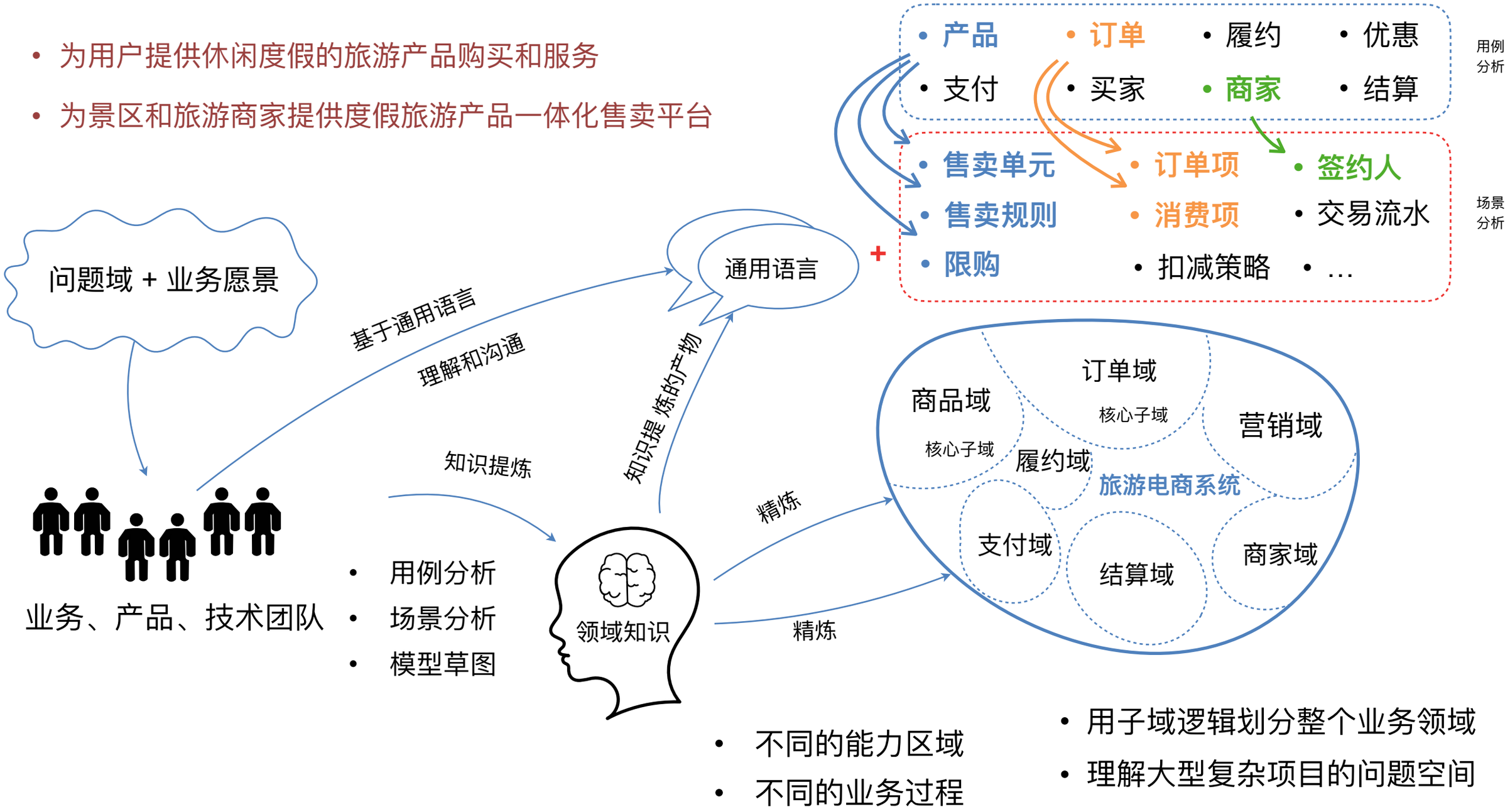
?



?

架构演进 - 阶段1 - 知识提炼 - 通用语言

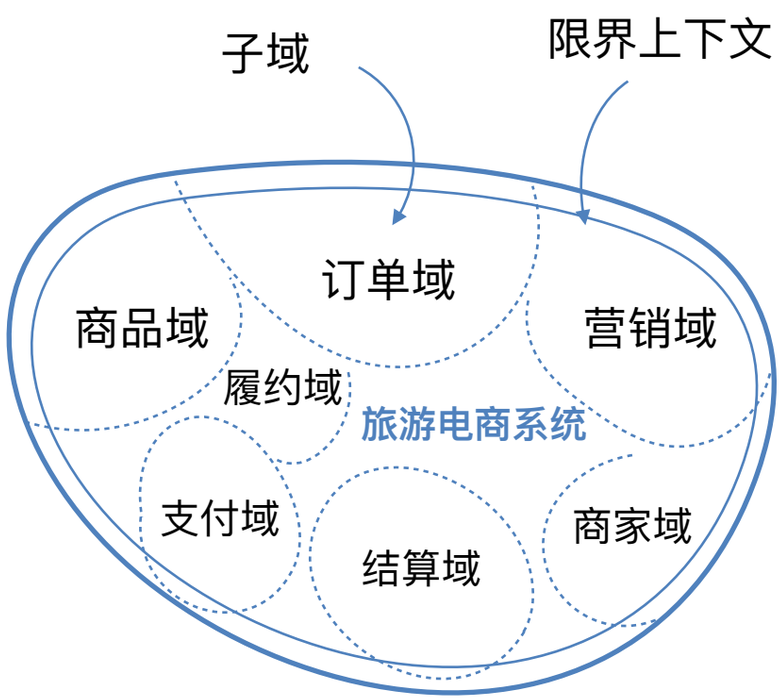
- 为用户提供休闲度假的旅游产品购买和服务
- 为景区和旅游商家提供度假旅游产品一体化售卖平台



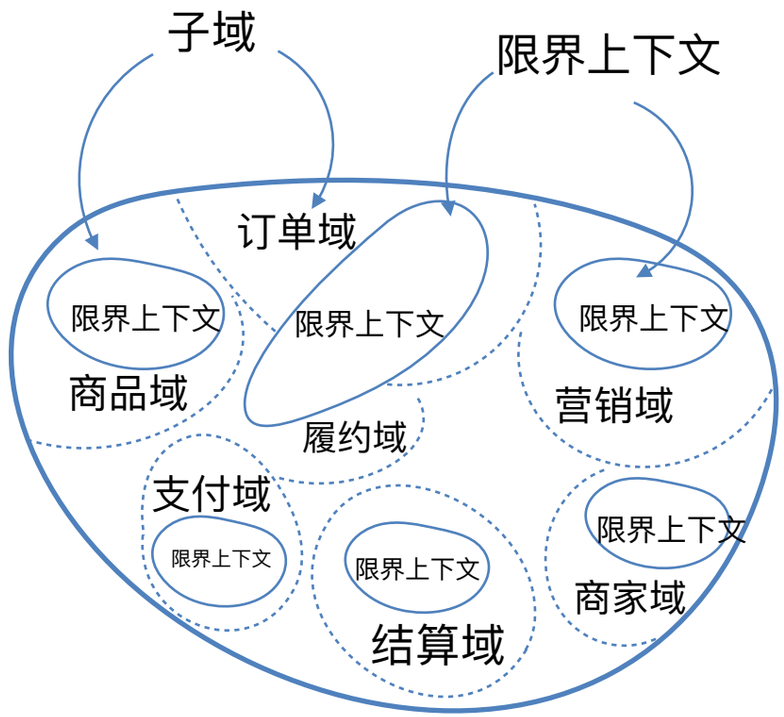
- 不同的能力区域
- 不同的业务过程

- 用子域逻辑划分整个业务领域
- 理解大型复杂项目的问题空间

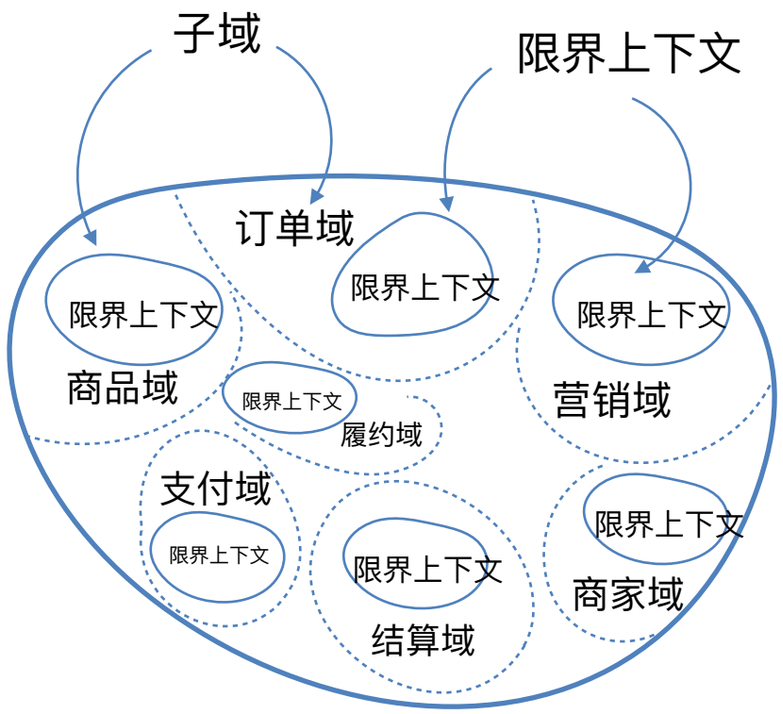
架构演进 - 阶段1 - 知识提炼 - 初步划分限界上下文



1 识别子域



2 初步划分限界上下文



3 如何进一步划分限界上下文?

架构演进 - 阶段1 - 进一步划分限界上下文

限界上下文划分依据

- **围绕语言定义边界**

Enforce linguistic boundaries to protect the validity of a domain term.

Linguistic boundaries are bounded context boundaries.

- **与业务能力保持一致**

business capabilities are often strong indicators of linguistic boundaries

- **围绕团队创建上下文**

一个限界上下文有一个**特性团队**负责：2PTS

from Millett S., Tune N.
- Patterns, Principles, and Practices of Domain-Driven Design

- **业务边界**

识别业务主题：利用语义相关性和功能相关性对用例归类

单一抽象层次原则+正交原则

- **工作边界**

避免限界上下文工作边界过大：2PTS

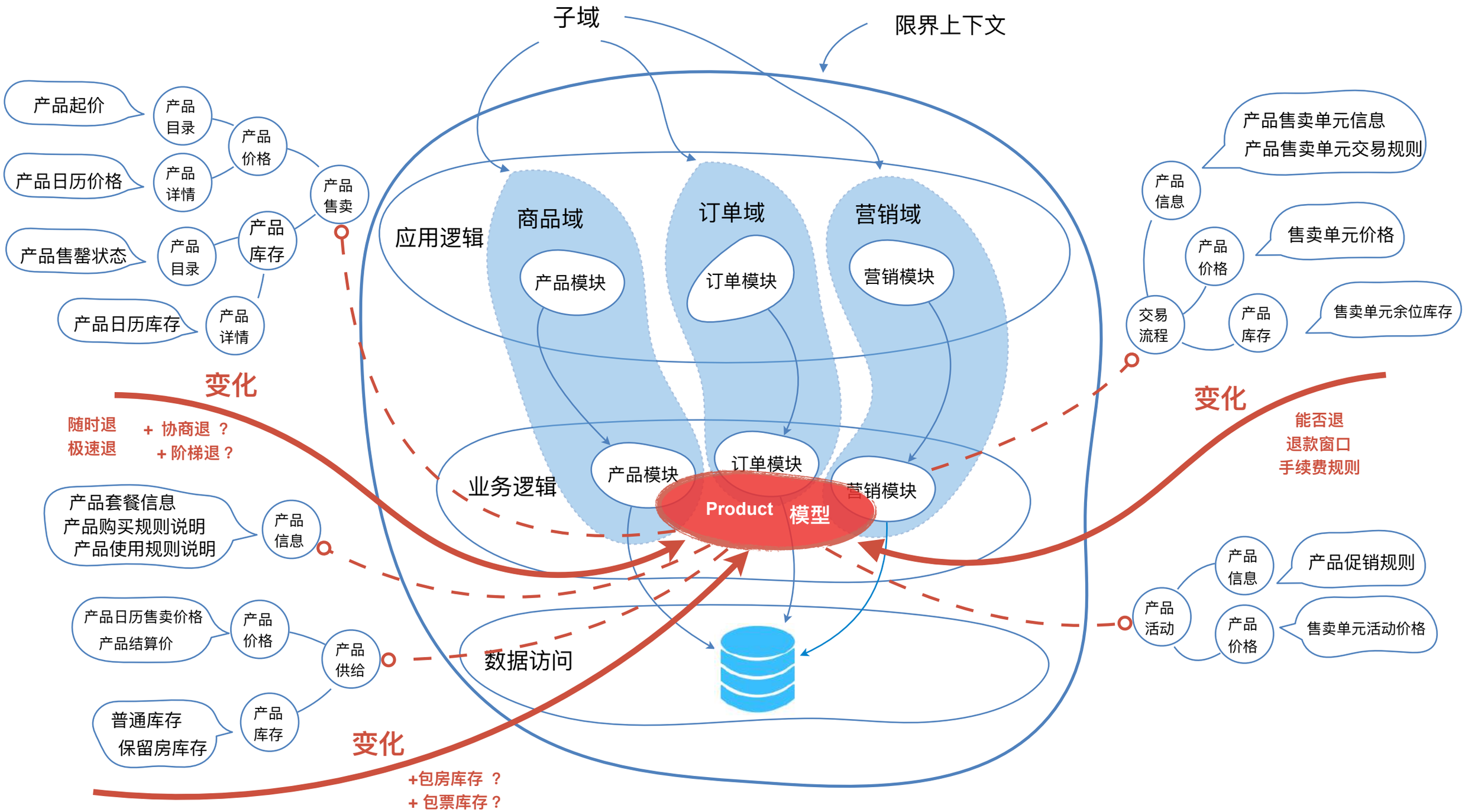
限界上下文允许并行开发的程度

- **应用边界**

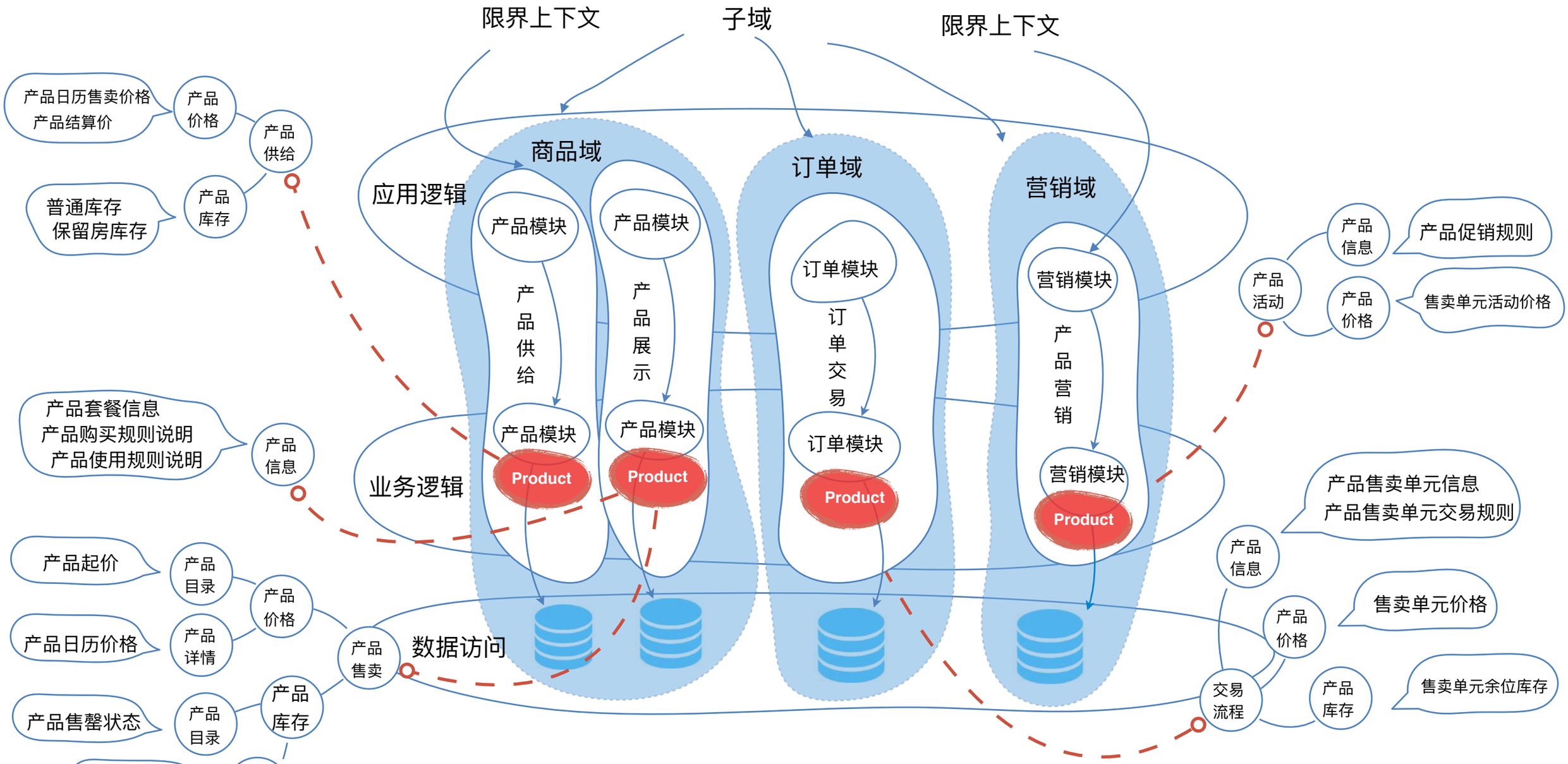
从技术角度考虑重用和变化的应对、遗留系统的集成

from 张逸
- 领域驱动战略设计实践

架构演进 - 阶段1 - 划分限界上下文 - 语言边界



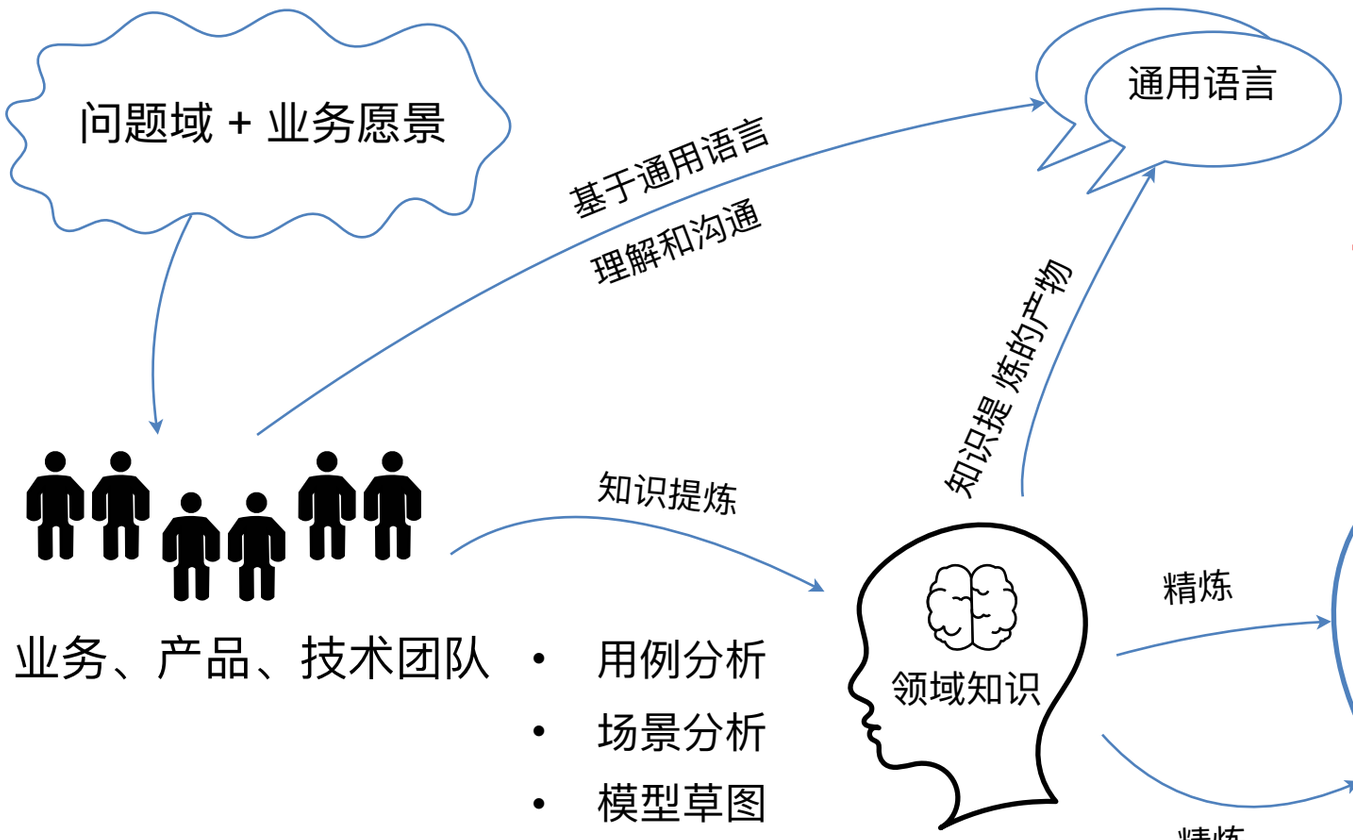
架构演进 - 阶段1 - 划分限界上下文 - 语言边界



用限界上下文消除大模型

架构演进 - 阶段1 - 知识提炼 - 结果

- 为用户提供休闲度假的旅游产品购买和服务
- 为景区和旅游商家提供度假旅游产品一体化售卖平台



用例分析

- 产品
- 订单
- 履约
- 优惠
- 支付
- 买家
- 商家
- 结算

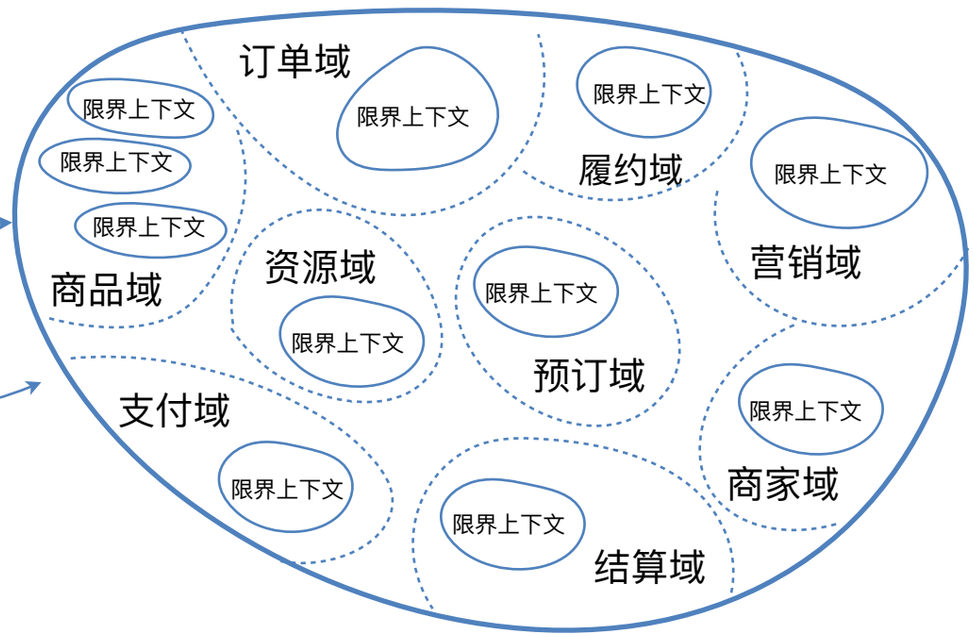
场景分析

- 售卖单元
- 订单项
- 签约人
- 售卖规则
- 消费项
- 交易流水
- 限购
- 扣减策略
- …

\+

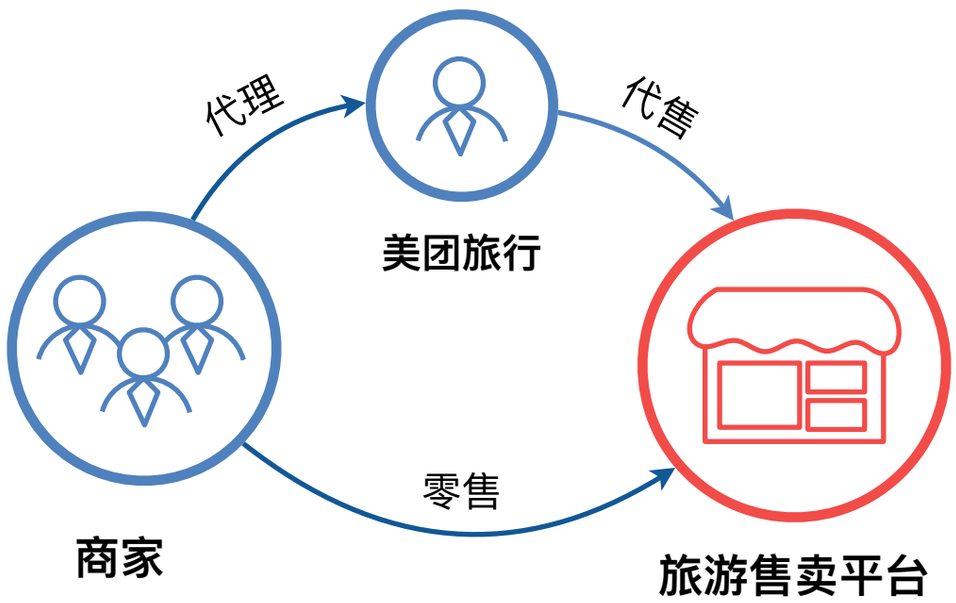
迭代结果

- 产品起价
- 售罄状态
- 价格模式
- 库存模式
- 分级库存
- 余位库存
- 占位库存
- …

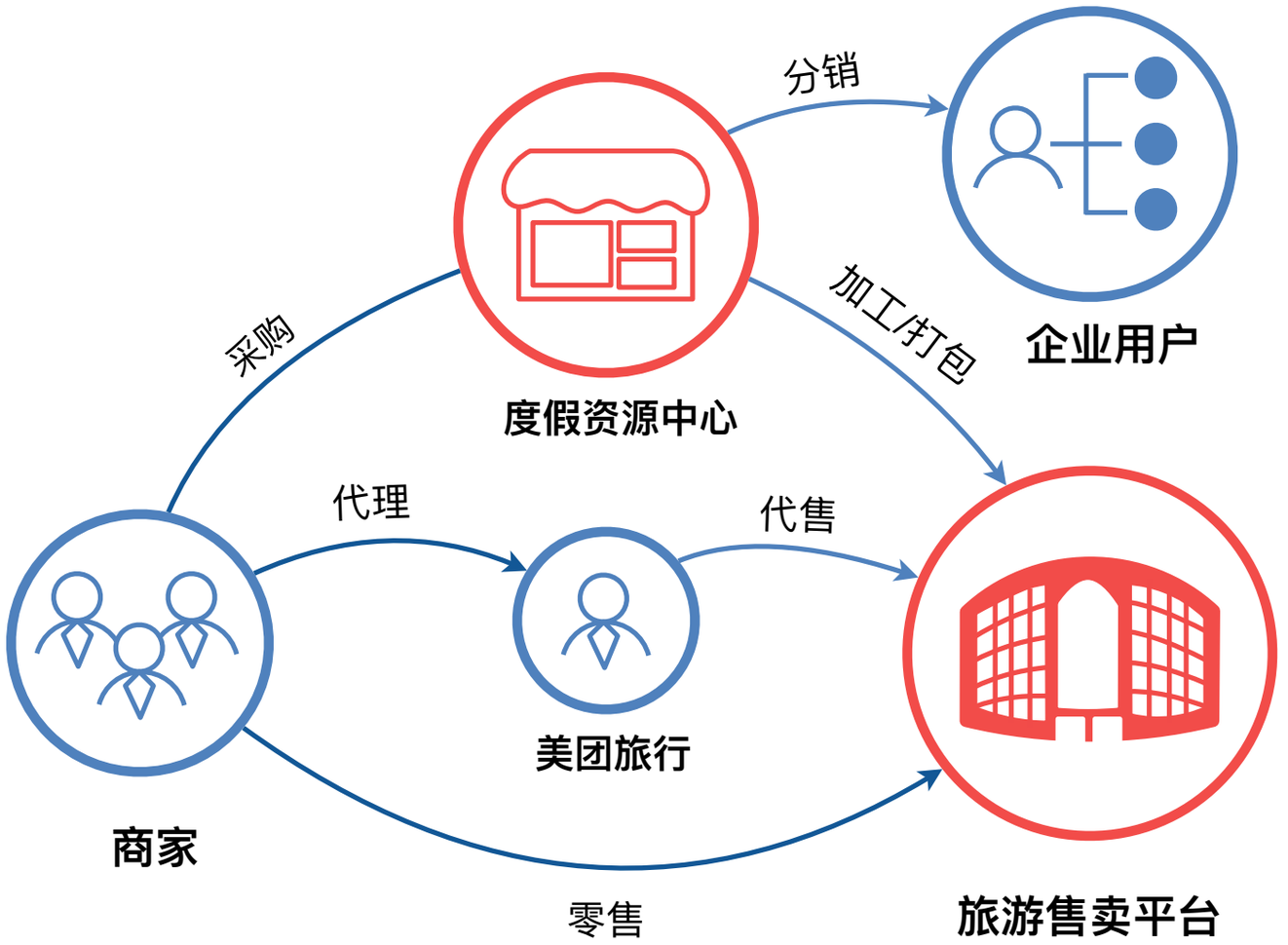


架构演进 - 阶段2 - 背景

阶段1



阶段2

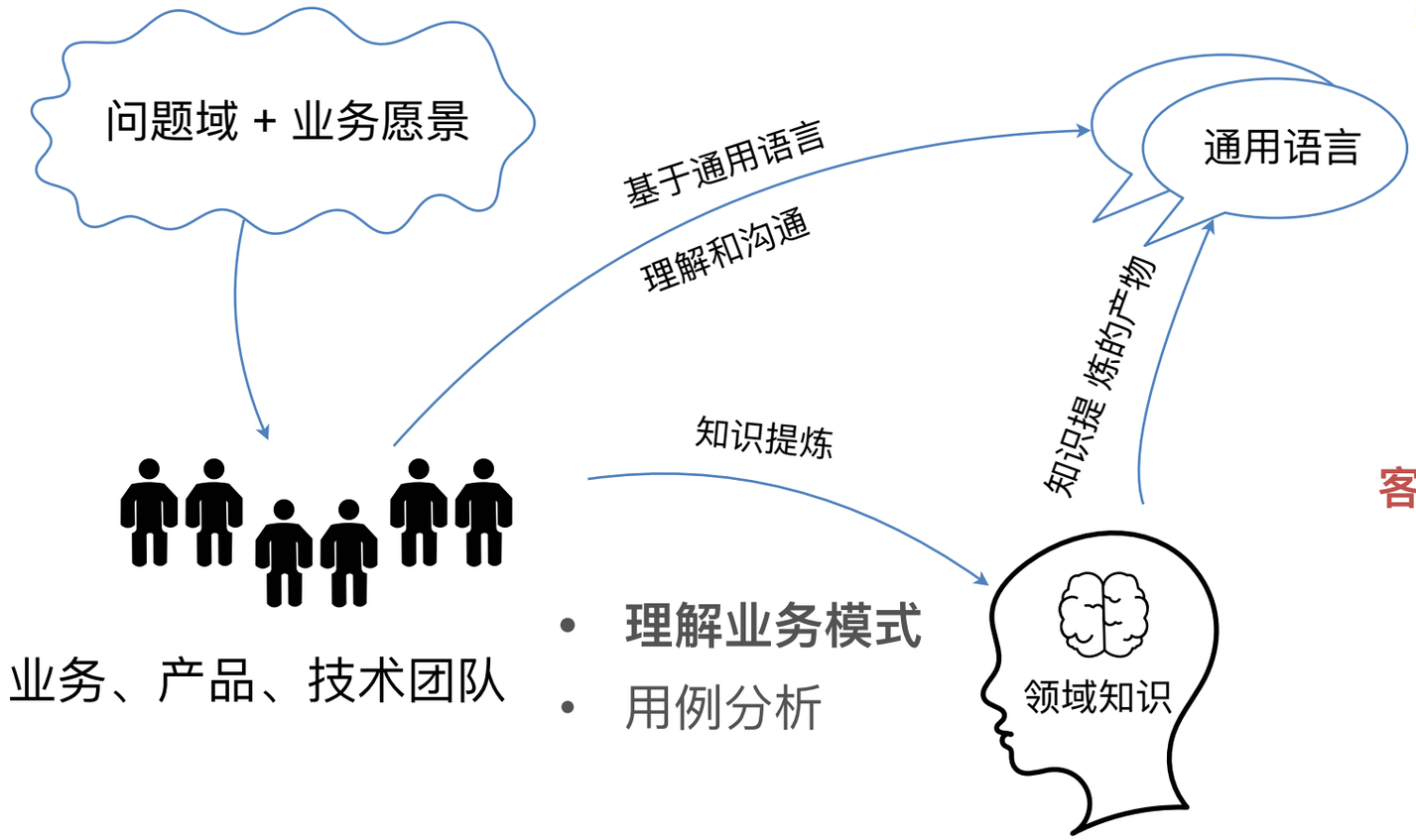


架构演进 - 阶段2 - 知识提炼

- 为用户提供休闲度假的旅游产品购买和服务
- 为用户提供多元化旅游场景服务和自主经营旅游产品服务
- 为景区和旅游商家提供度假旅游产品一体化产品售卖和分销平台

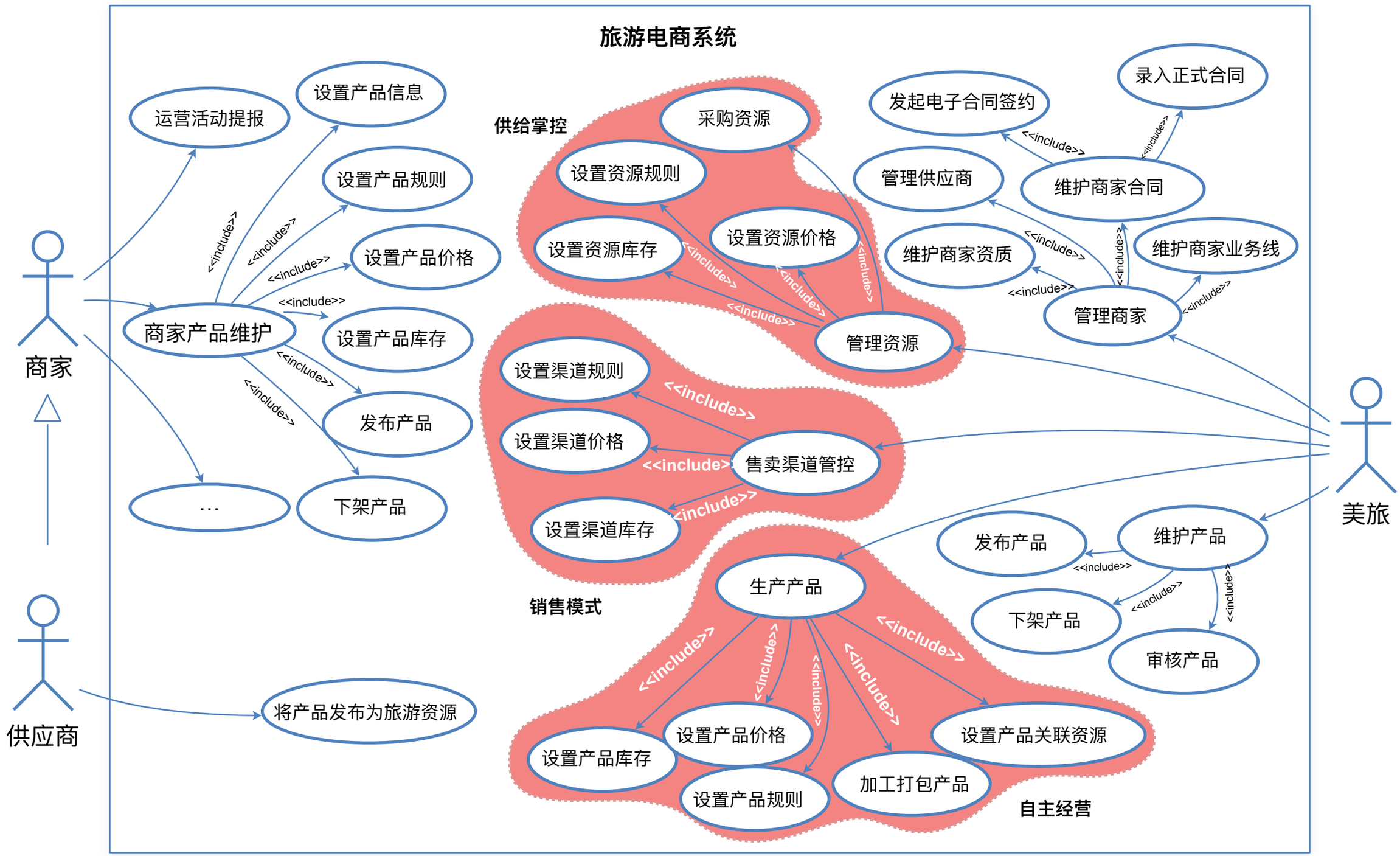
行业调研分析

注：图片来自艾瑞咨询

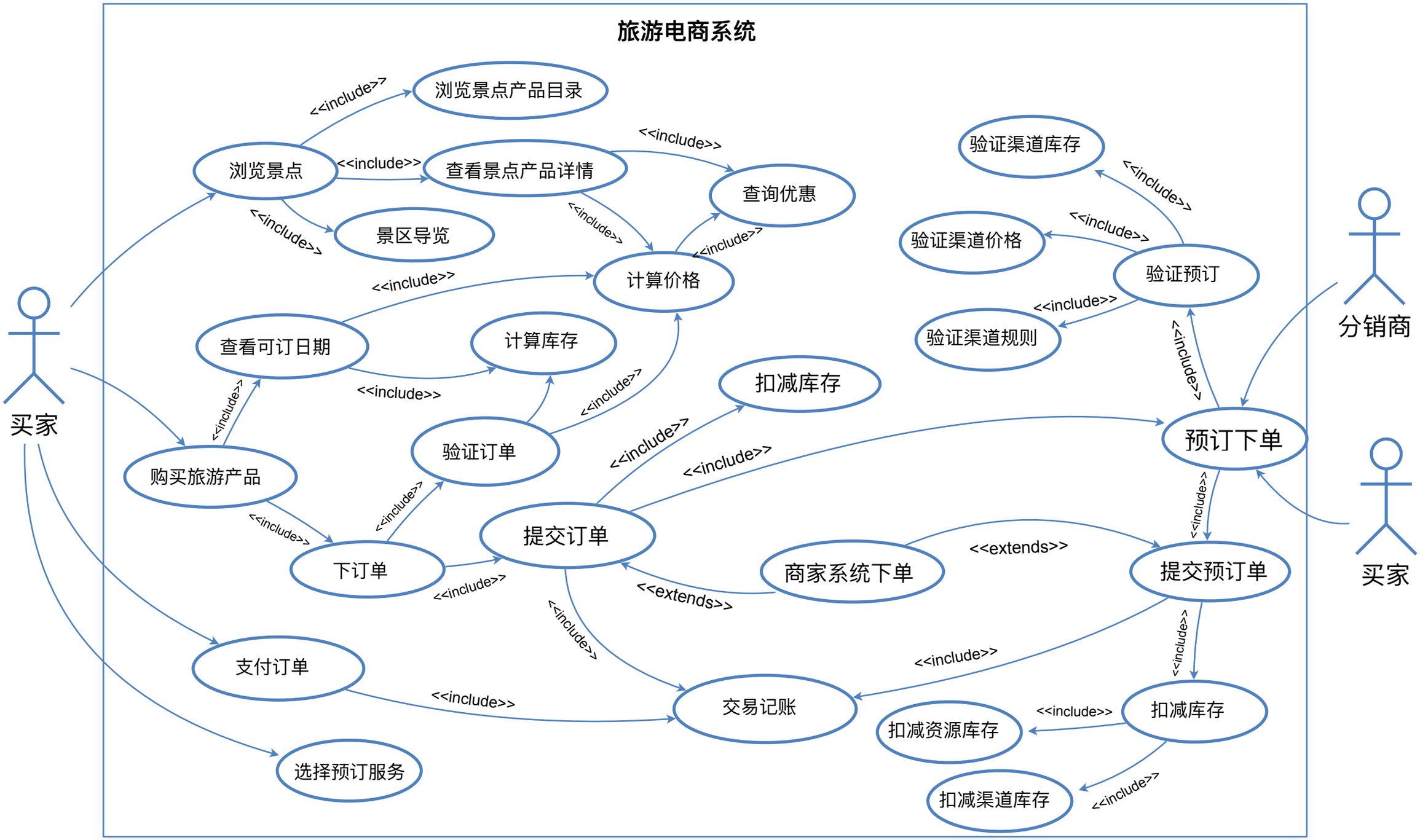


知识提炼 - 理解业务模式

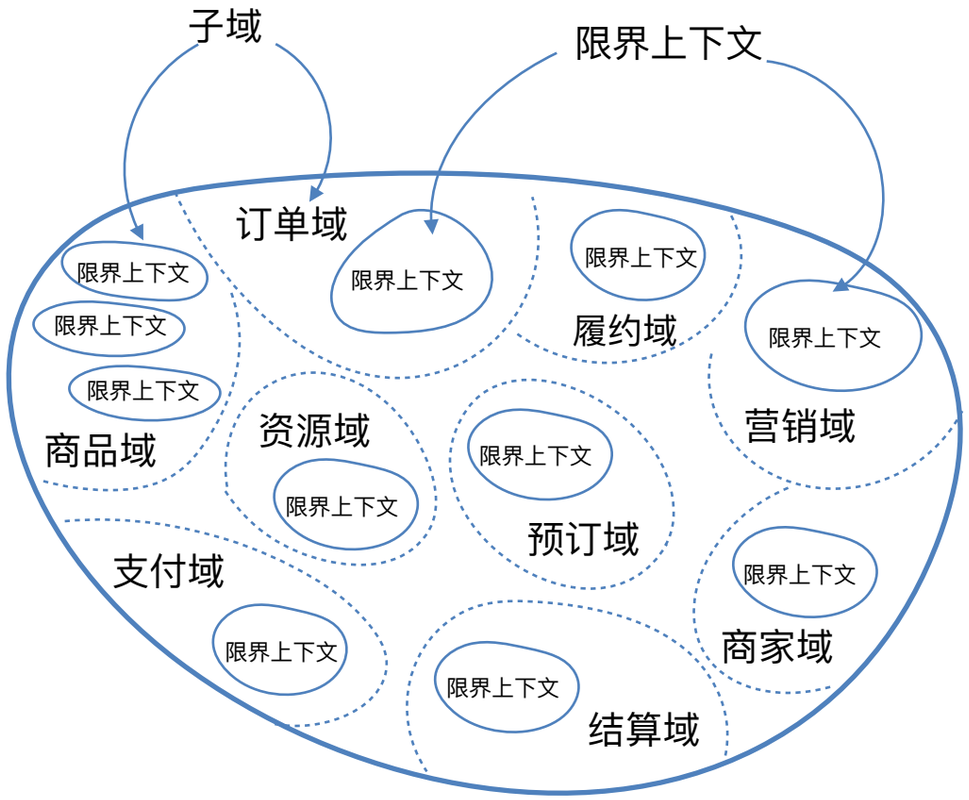
架构演进 - 阶段2 - 知识提炼 - 用例分析

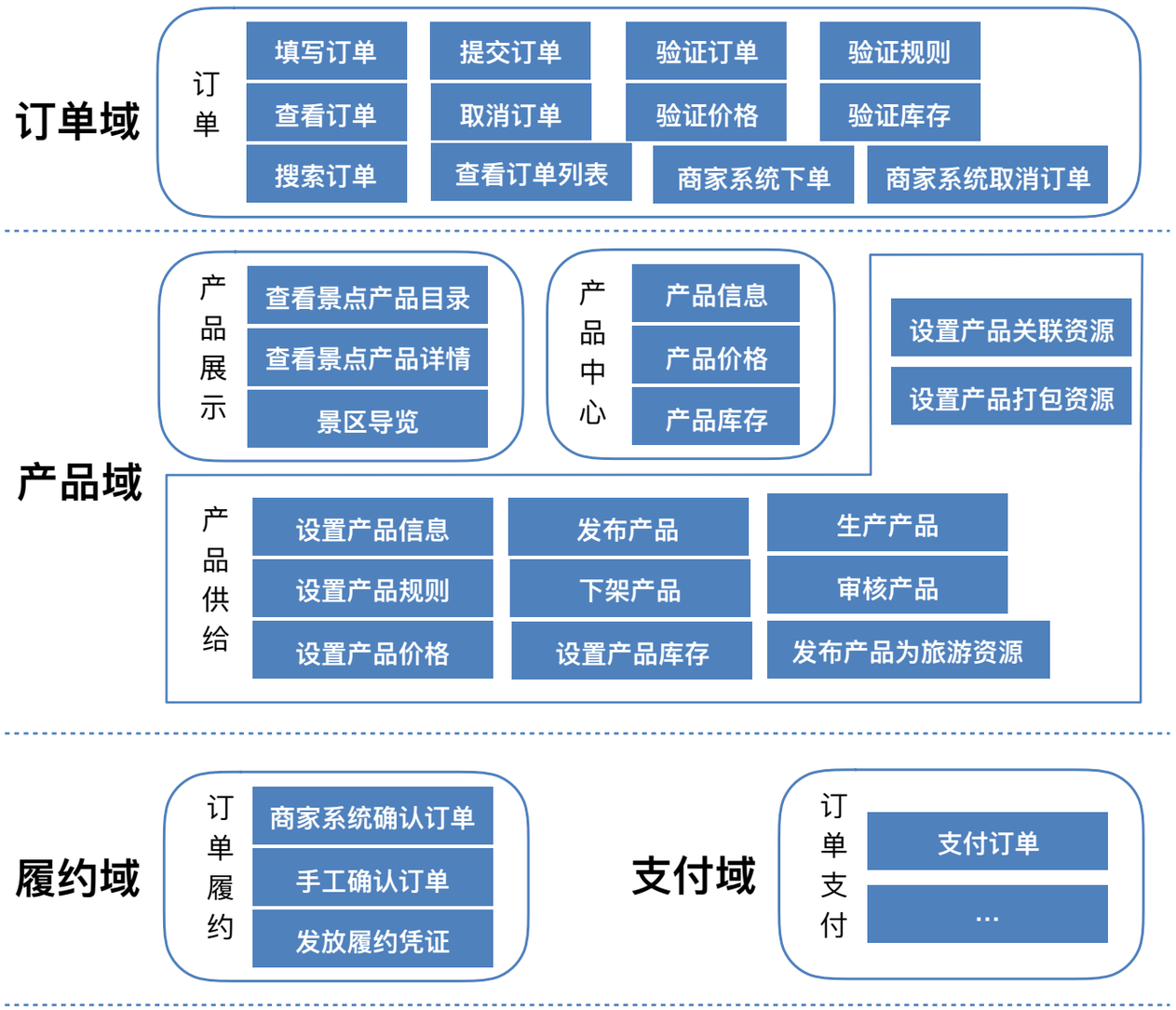


架构演进 - 阶段2 - 知识提炼 - 用例分析

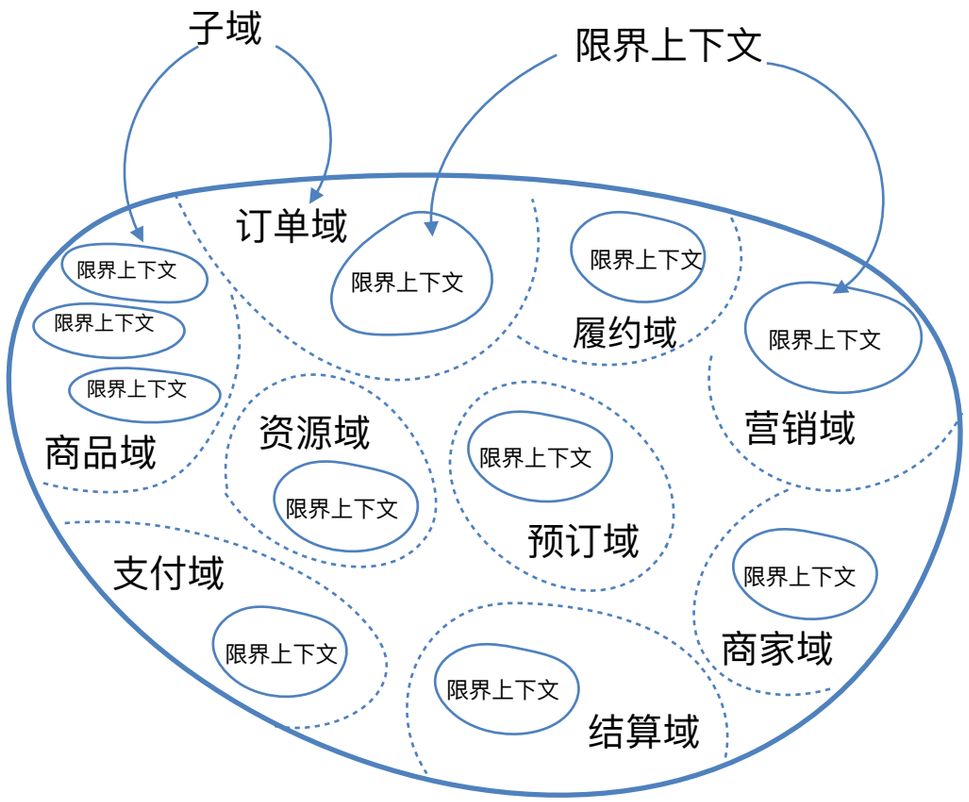


架构演进 - 阶段2 - 知识提炼 - 识别子域和限界上下文





架构演进 - 阶段2 - 知识提炼 - 识别子域和限界上下文



预订域

订单

选择预订服务 | 预订下单 | 验证预订 | 验证渠道规则
查看预订单 | 取消预订 | 验证渠道价格 | 验证渠道库存
供应商系统下单 | 供应商系统取消订单

资源域

资源中心

查询资源信息
查询资源价格 | 查询资源规则 | 查询渠道价格
查询资源库存 | 查询渠道规则 | 查询渠道库存

资源采购

采购资源 | 设置渠道规则
设置资源规则 | 设置渠道价格
设置资源价格 | 设置资源库存 | 设置渠道库存

商家域

供应商管理

维护供应商信息
维护供应商资质
维护供应商账号

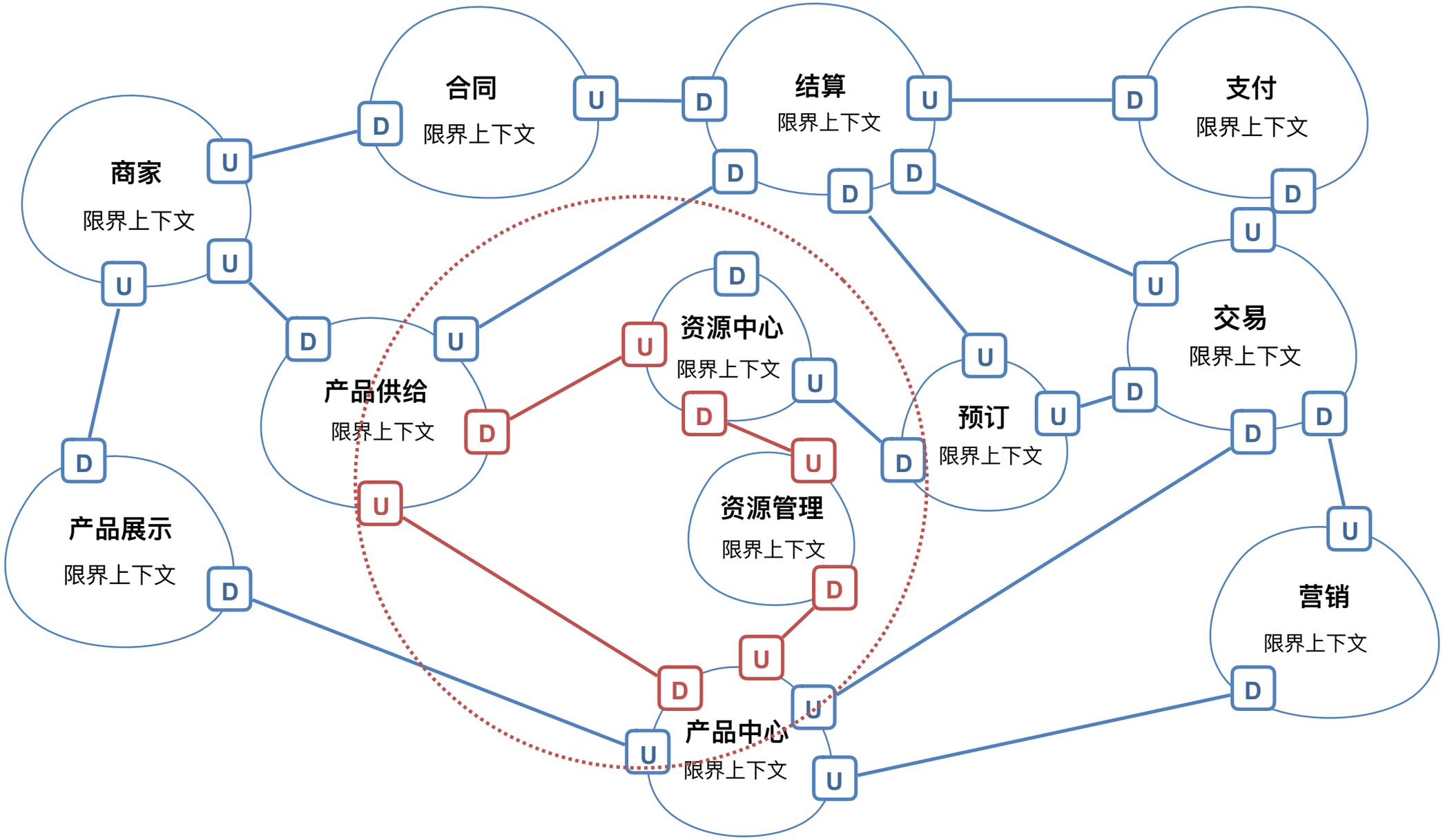
合同管理

维护合同信息
发起电子合同签约
电子合同签约

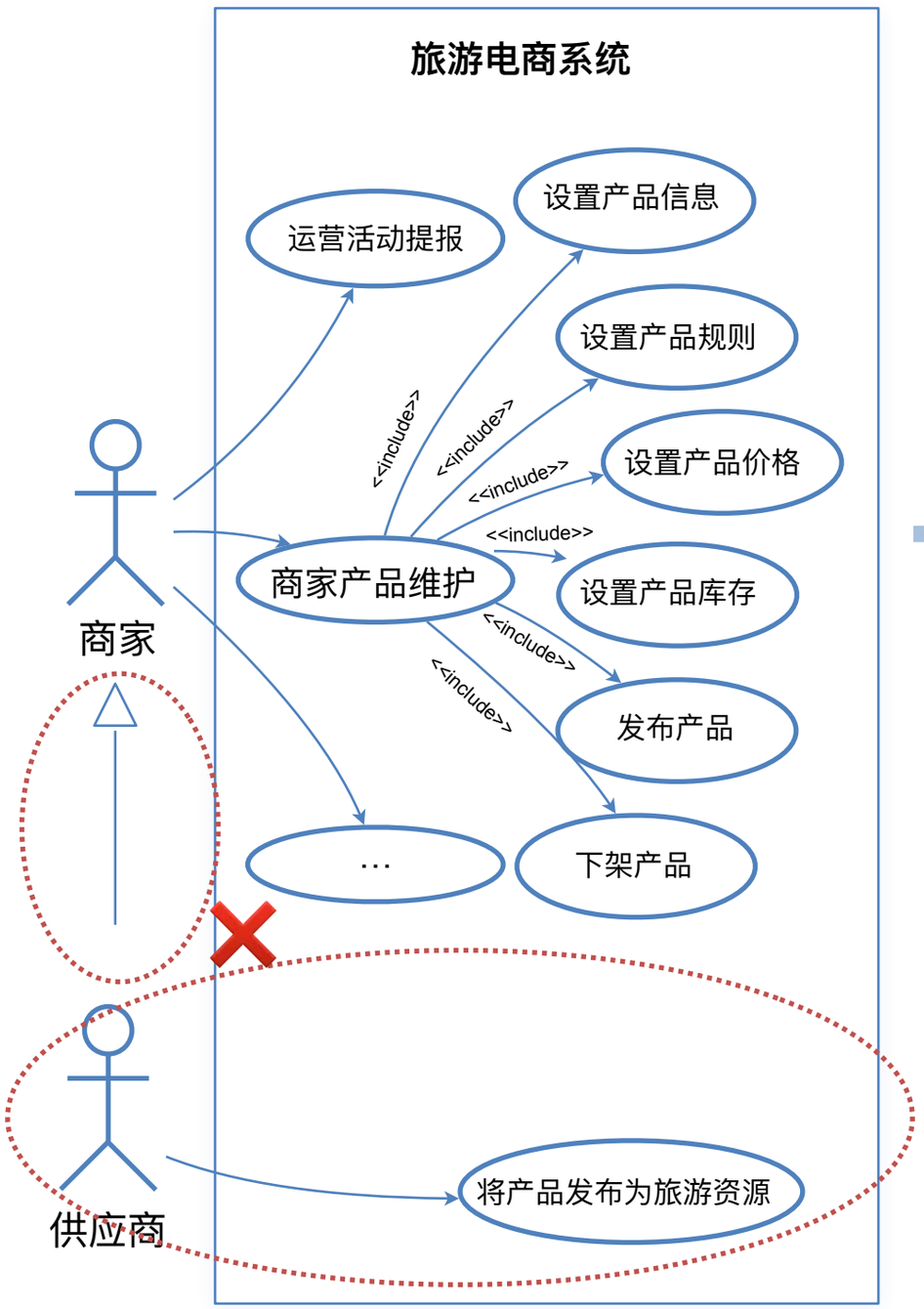
商家管理

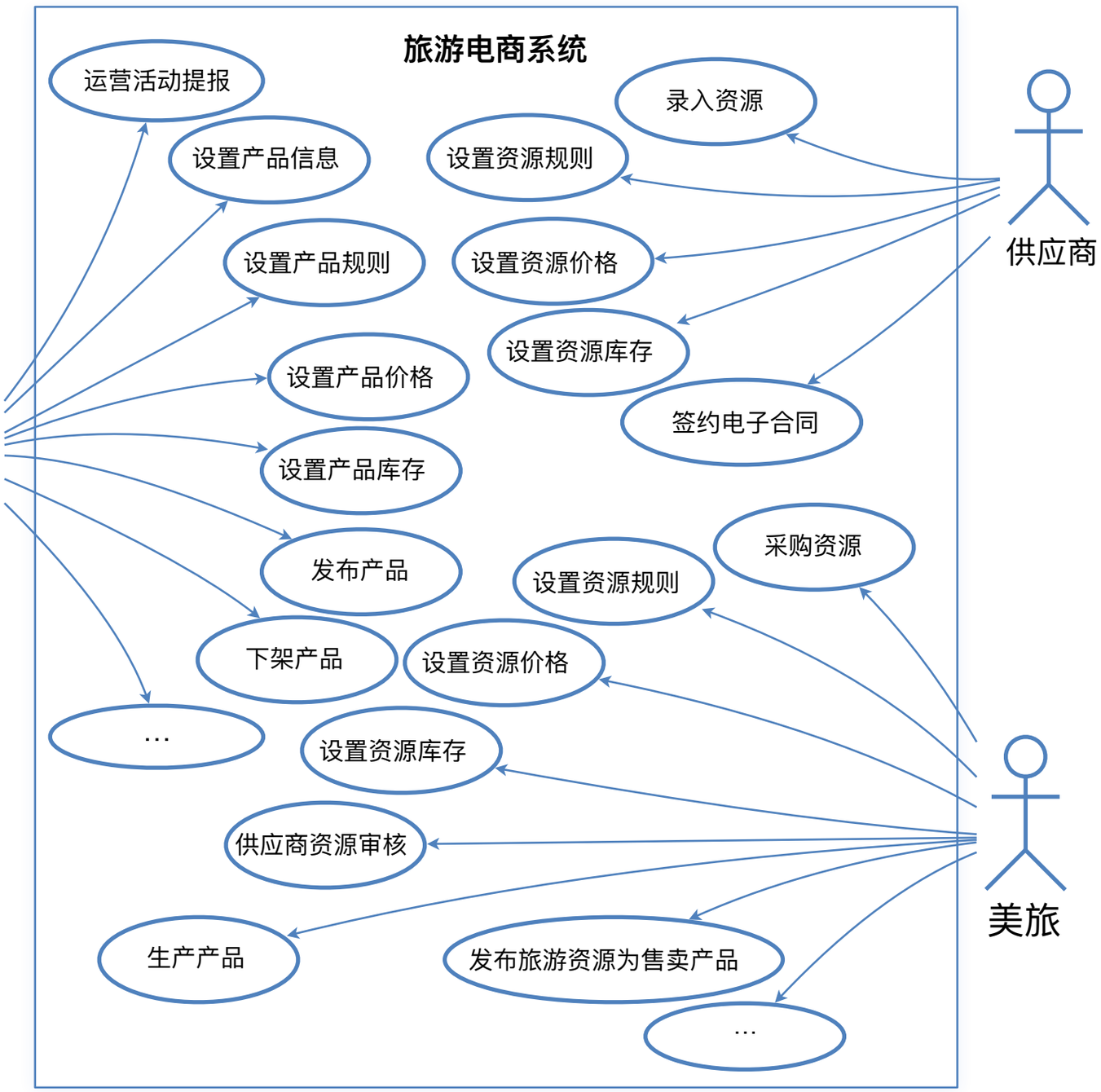
商家入驻
维护商家账号
维护商家信息

架构演进 - 阶段2 - 限界上下文映射

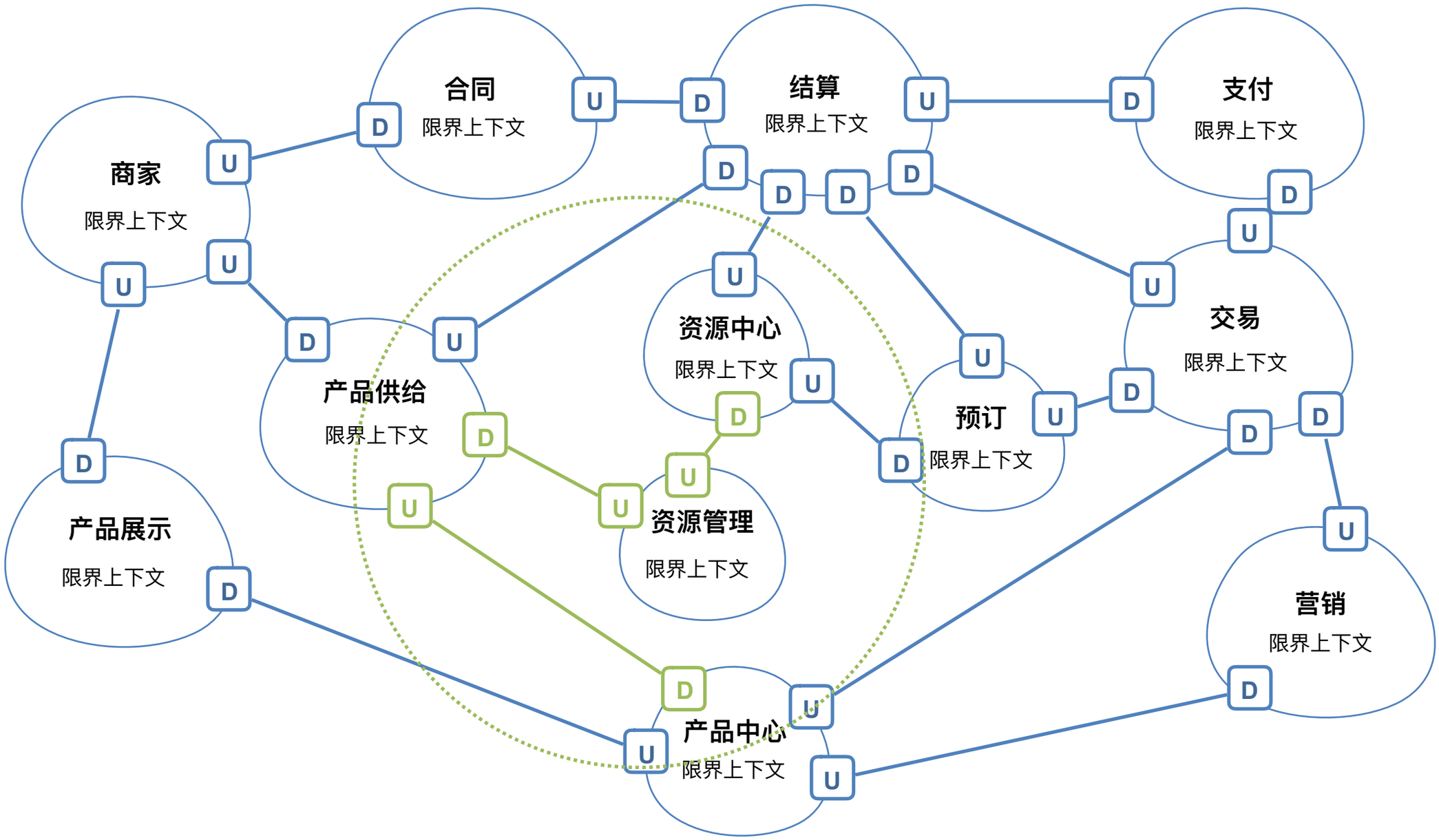


架构演进 - 阶段2 - 知识提炼 - 用例分析 - 迭代

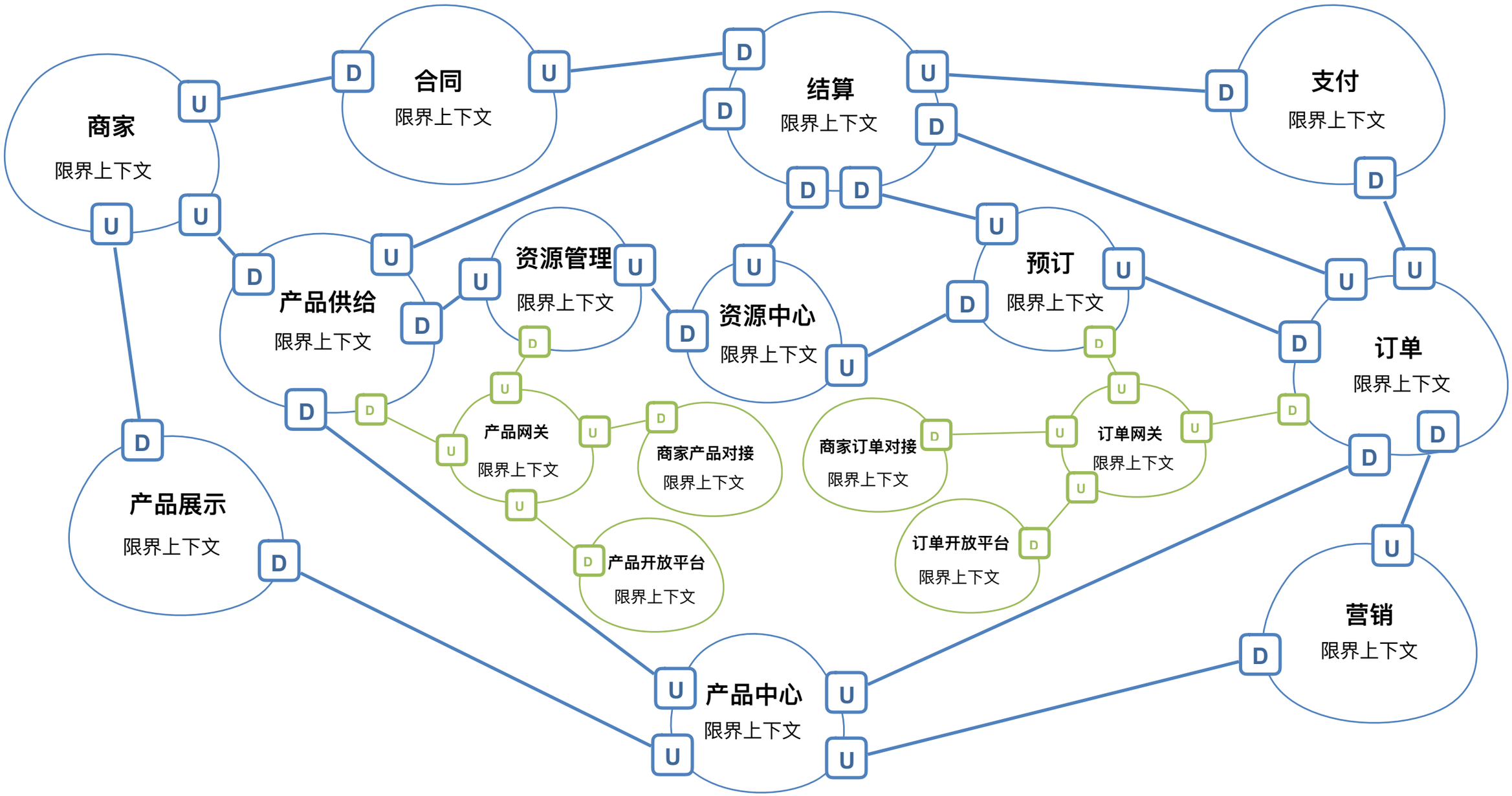




架构演进 - 阶段2 - 限界上下文映射 - 迭代

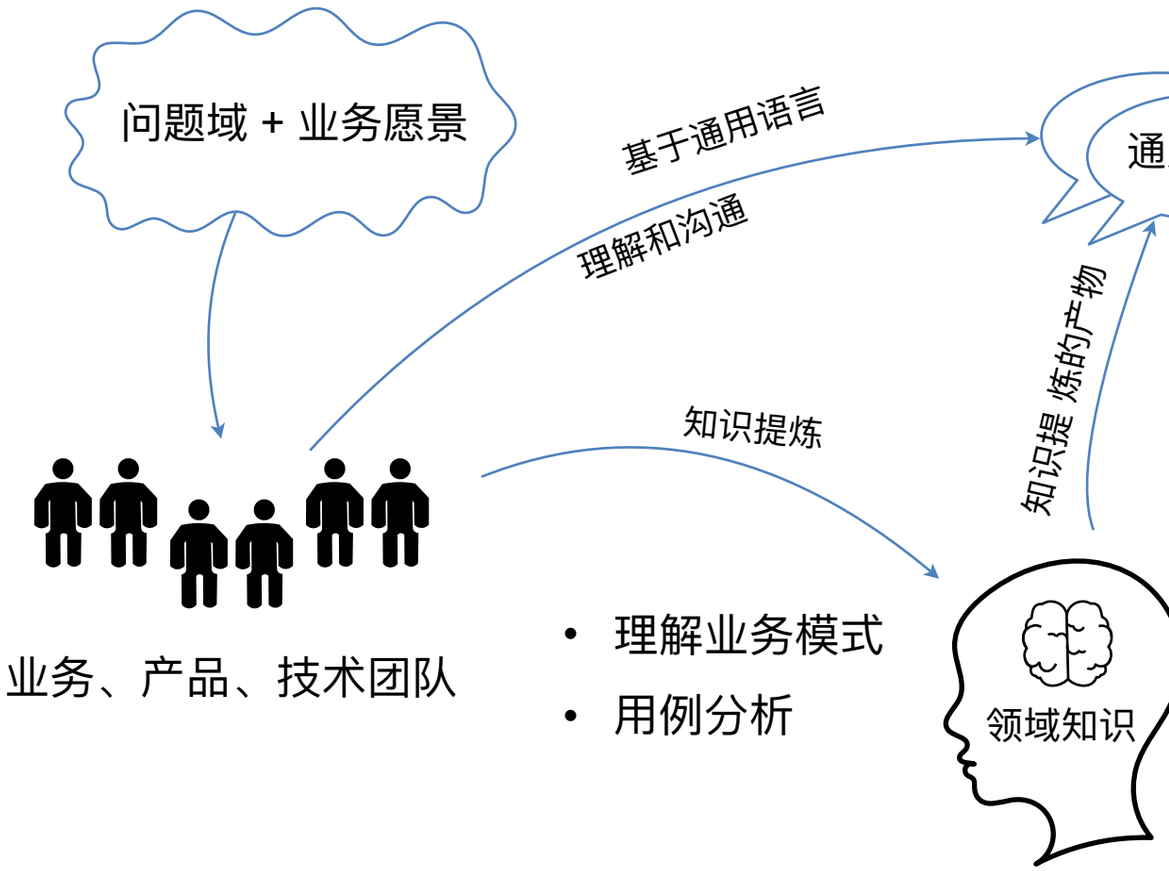


架构演进 - 阶段2 - 限界上下文映射 - 结果



架构演进 - 阶段2 - 通用语言

- 为用户提供休闲度假的旅游产品购买和服务
- 为用户提供多元化旅游场景服务和自主经营旅游产品服务
- 为景区和旅游商家提供度假旅游产品一体化产品售卖和分销平台



阶段1

- 产品
- 售卖规则
- 价格模式
- 售卖单元
- 售卖单元
- 库存模式
- 消费项
- 服务履约
- 扣减策略
- 订单项
- 交易流水
- 分级库存
- …

阶段2

- 打包
- 资源
- 拼团
- 交易
- 分销
- 预售
- 预订
- 预约
- 搭售
- 零售
- 代理
- 自营
- …

CONTENTS

架构语言

架构（Architecture）

模块（Module）

限界上下文（Bounded Context）

问题空间（Problem Space）

域（Domain）

服务（service）

微服务（MicroService）

解空间（Solution Space）

组件（Component）

架构量子（Architectural Quantum）

设计模式（Design Pattern）

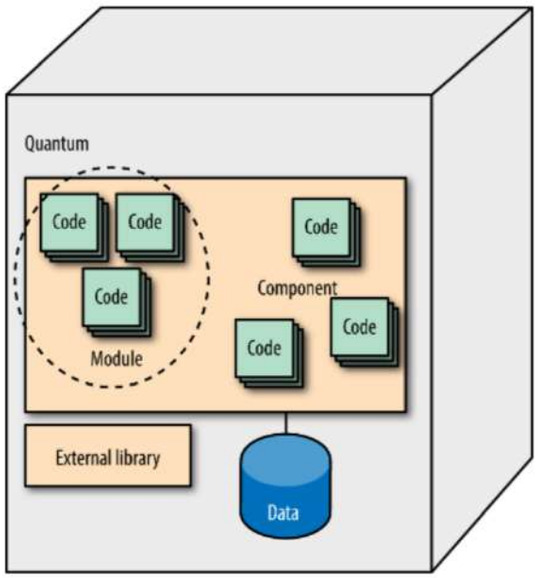
库（library）

架构模式（Architectural Pattern）

架构风格（Architectural Style）

限界上下文是模块？组件？服务？

架构语言



— from Building evolutionary architecture

应用架构
（Application architecture）

架构关注点

构建部署因素（进程内）

组件
（Component）

物理组织为

模块
（Module）

逻辑组织为

代码
（Code）

构建部署因素（进程间）

限界上下文
（Bounded Context）

架构量子
（Architectural Quantum）

库
（library）

包含

服务
（service）

微服务
（MicroService）

架构粒度
（Architectural Granularity ）

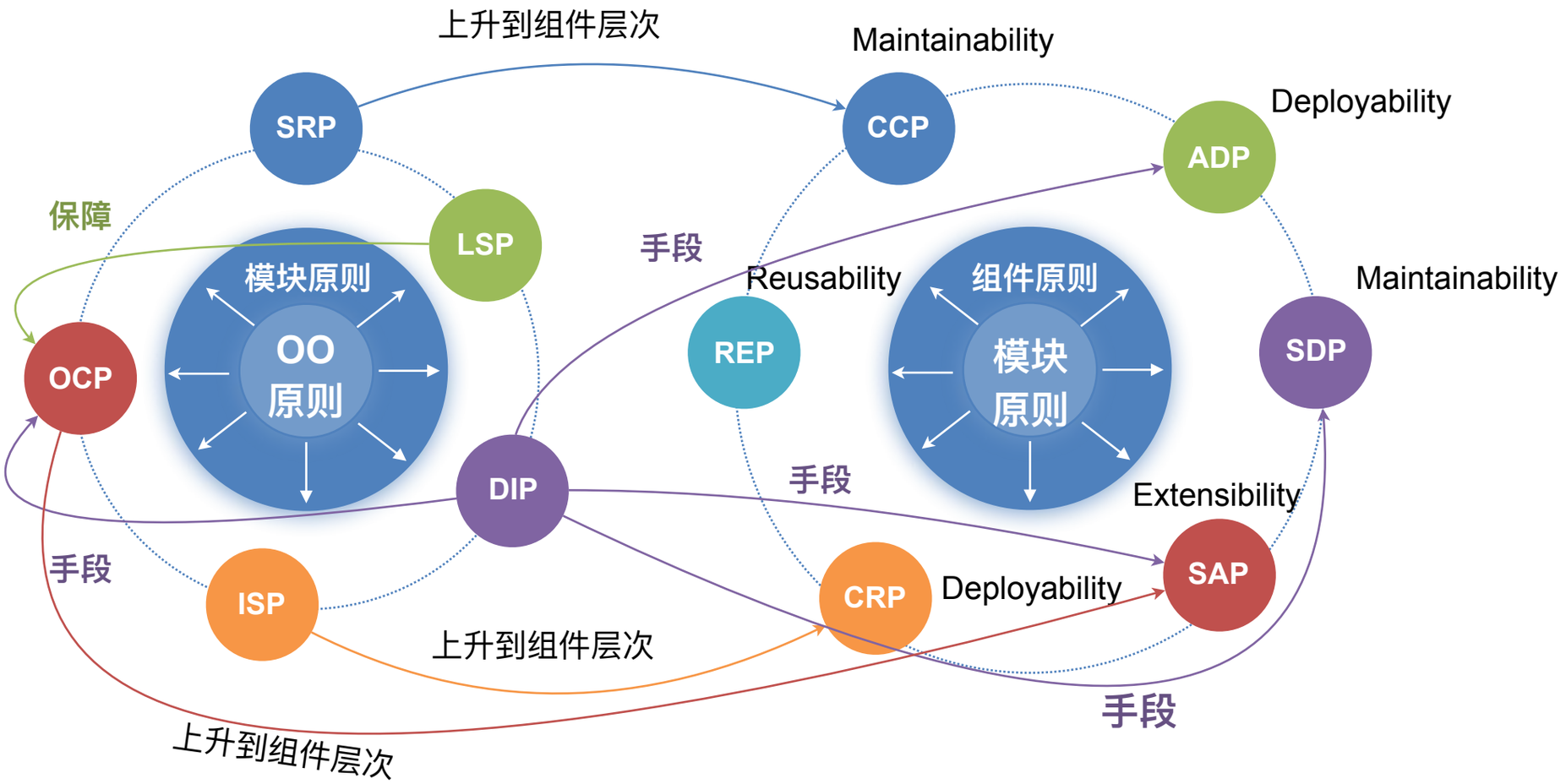
架构关注点

架构风格
（Architectural Style）

集成架构
（Integration architecture）

单体分层架构 Layered architecture monoliths → 单体模块化架构 Modular monoliths → 基于服务的架构 Service-based architectures → 微服务架构 Microservices

架构设计 - 原则



Robert C. Martin

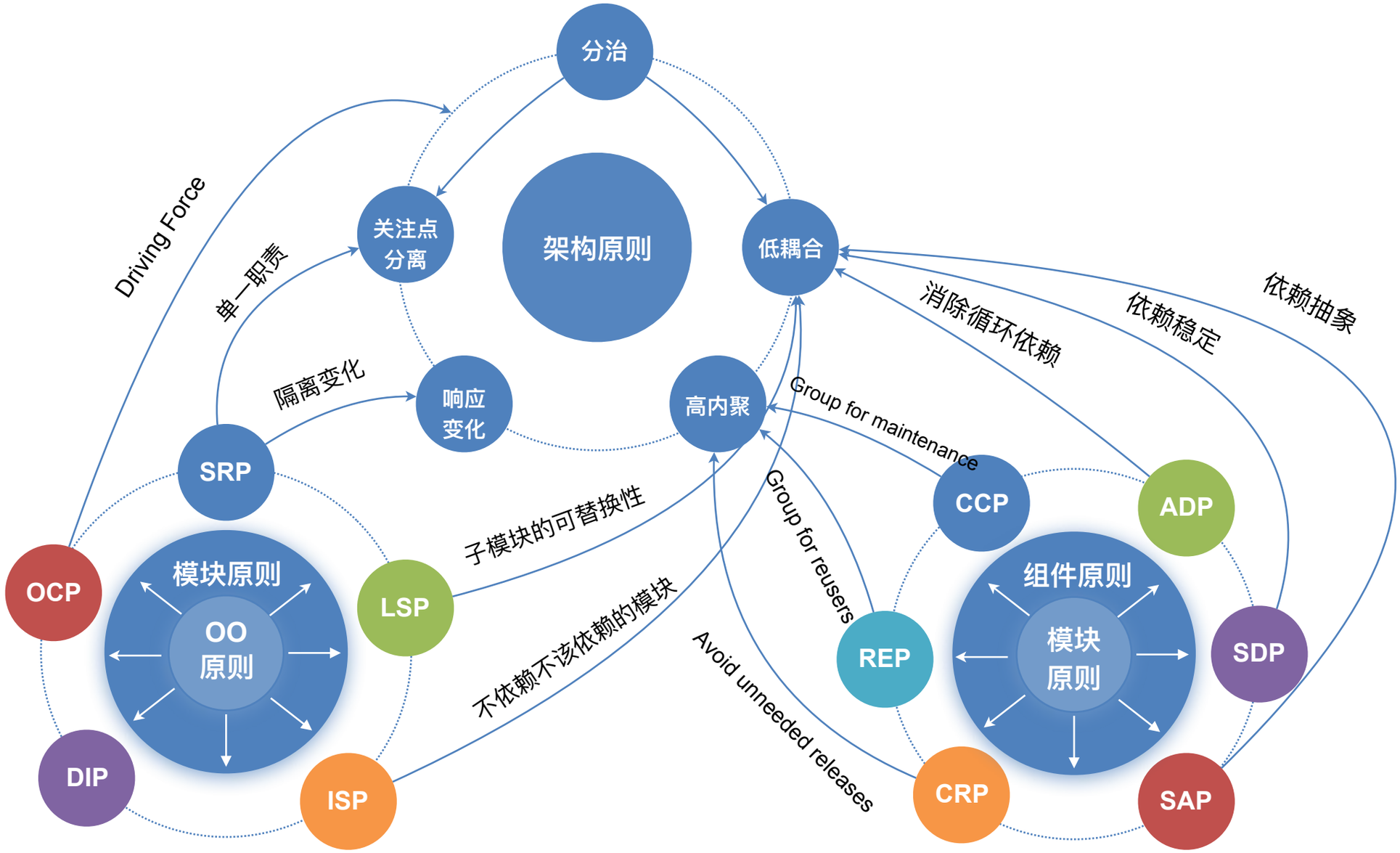
Agile Software Development, Principles, Patterns, and Practices, First Edition

Clean Architecture A Craftsman's Guide to Software Structure and Design

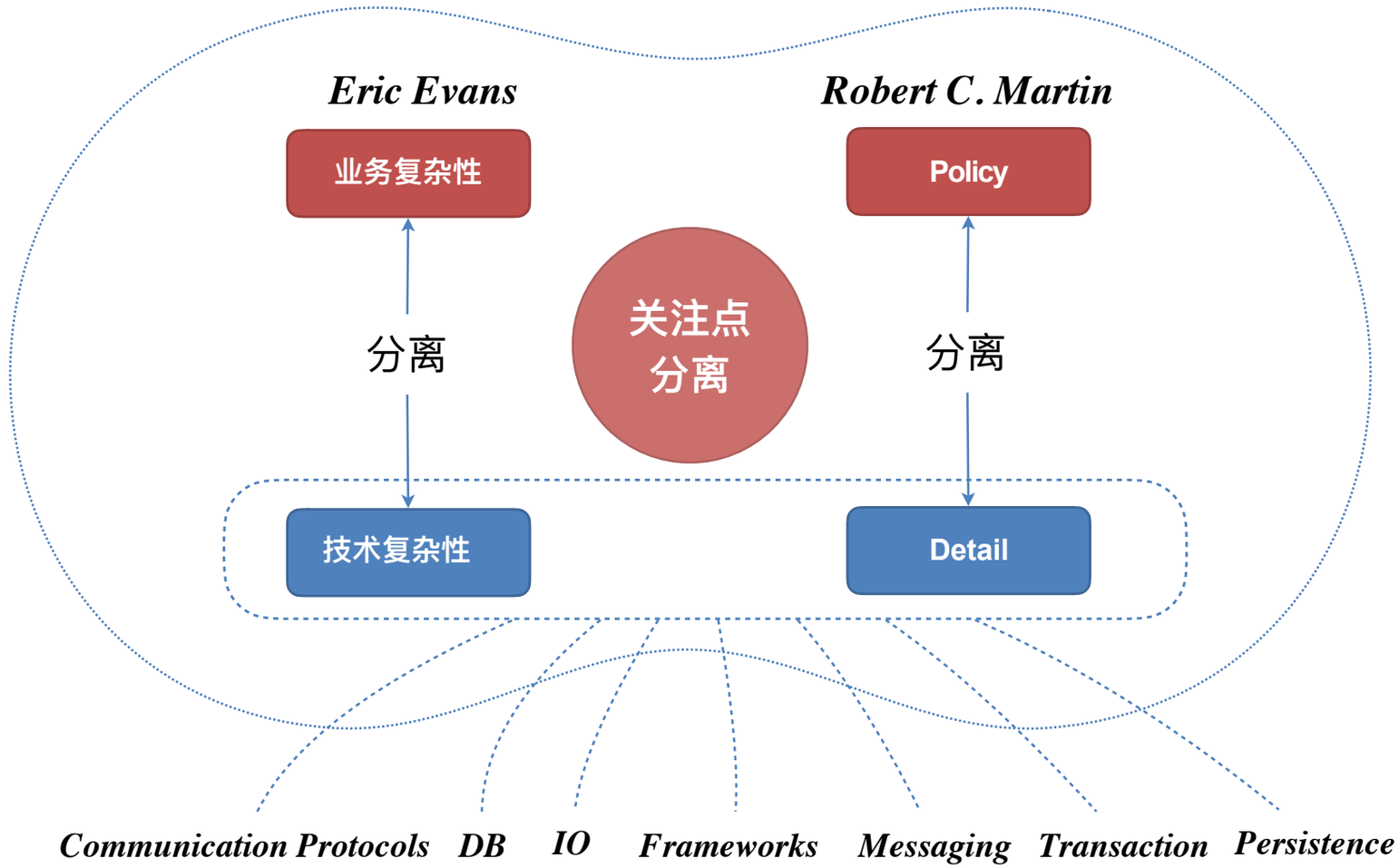
Package Principles

Component Principles

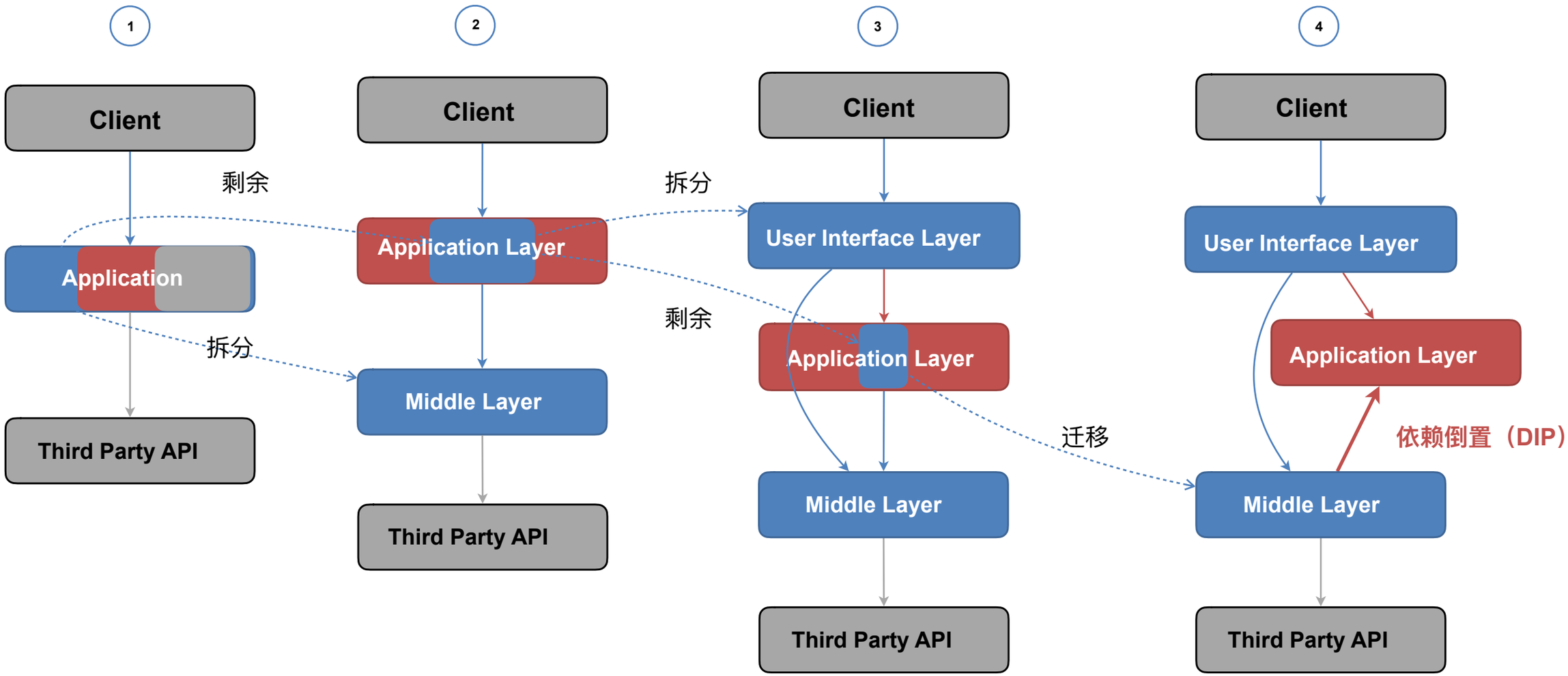
架构设计 - 原则


分治
架构原则
关注点
分离
低耦合
Driving Force
单一职责
隔离变化
响应
变化
高内聚
消除循环依赖
依赖稳定
依赖抽象
SRP
OCP
模块原则
OO
原则
LSP
DIP
ISP
子模块的可替换性
不依赖不该依赖的模块
Group for maintenance
Group for reusers
Avoid unneeded releases
CCP
ADP
REP
组件原则
模块
原则
SDP
CRP
SAP

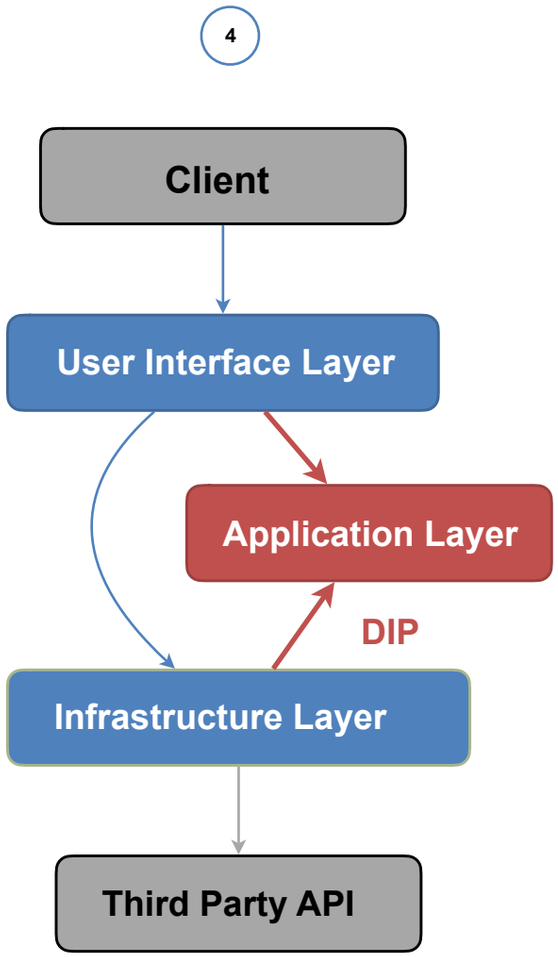
架构设计 - 原则

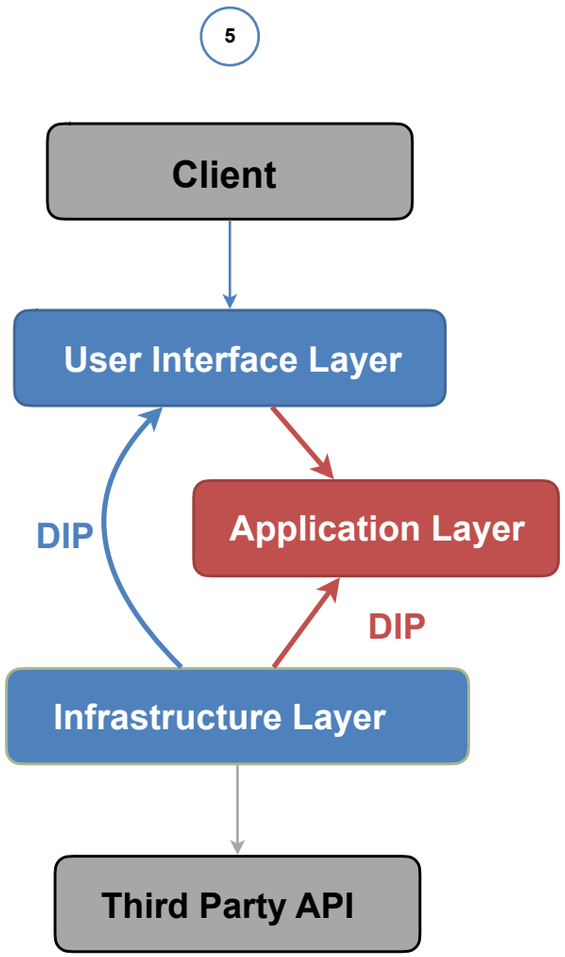

Eric Evans
Robert C. Martin
业务复杂性
Policy
关注点
分离
分离
分离
技术复杂性
Detail
Communication Protocols
DB
IO
Frameworks
Messaging
Transaction
Persistence

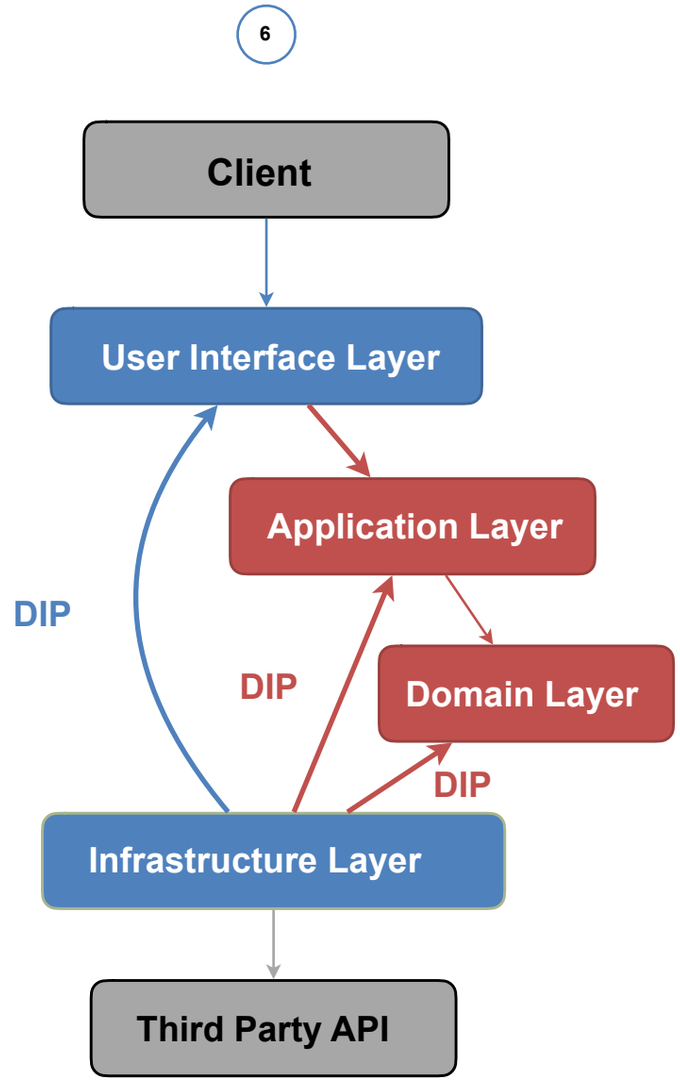
架构设计 - 模式 - 分层架构的演进



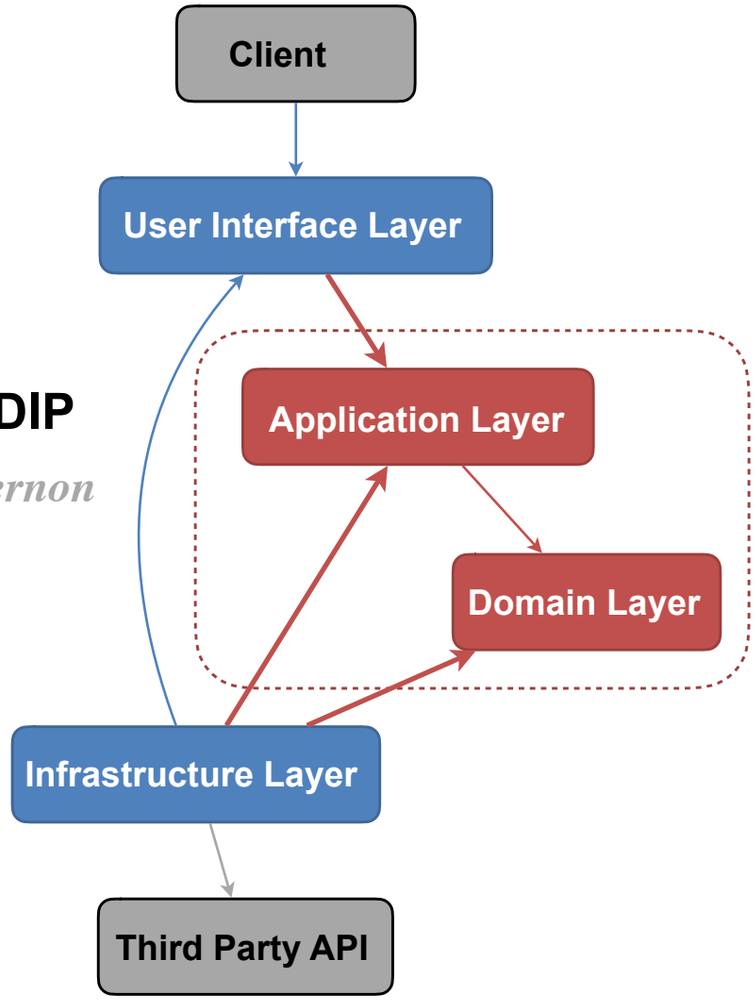
架构设计 - 模式 - 分层架构的演进





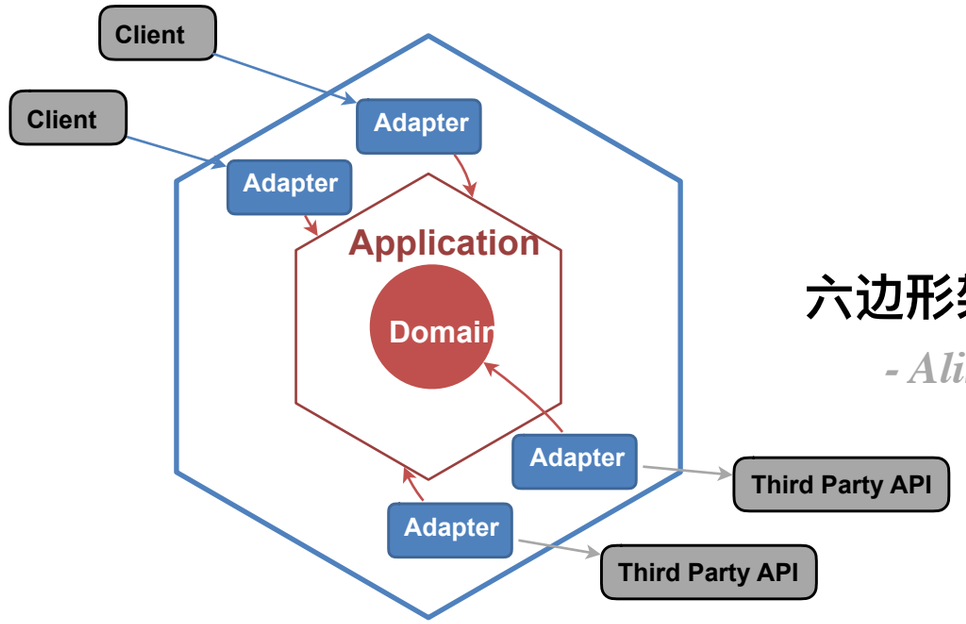


架构模式 - 应用架构



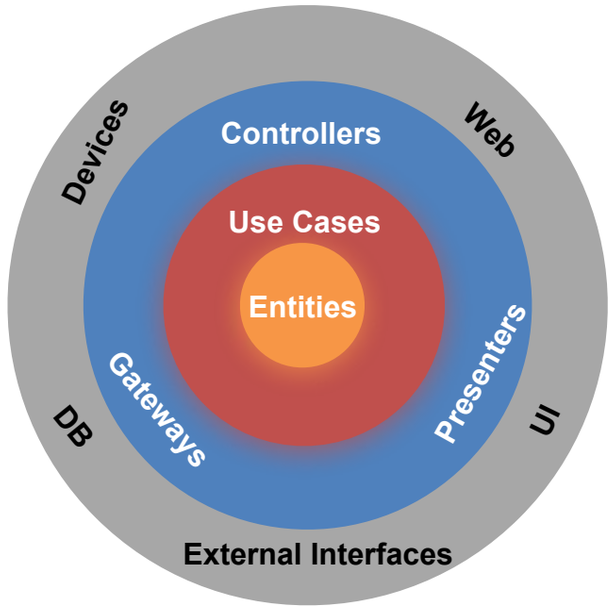
分层架构+DIP

- Vaughn Vernon



六边形架构

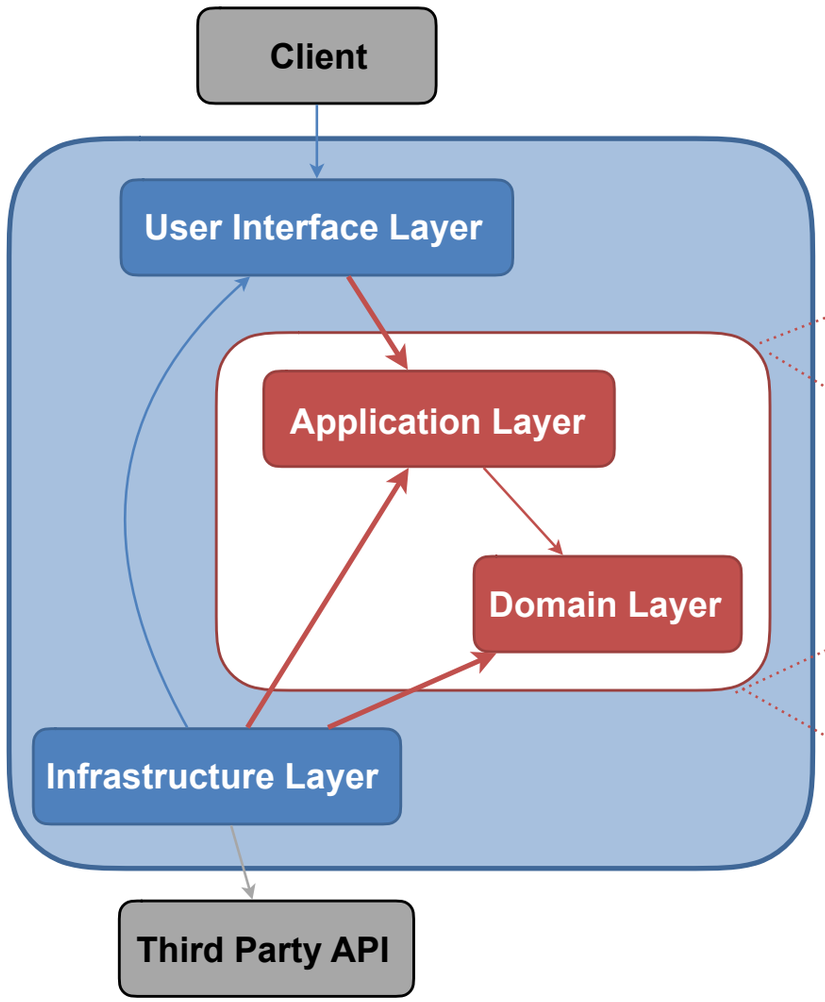
- Alistair Cockburn



整洁架构

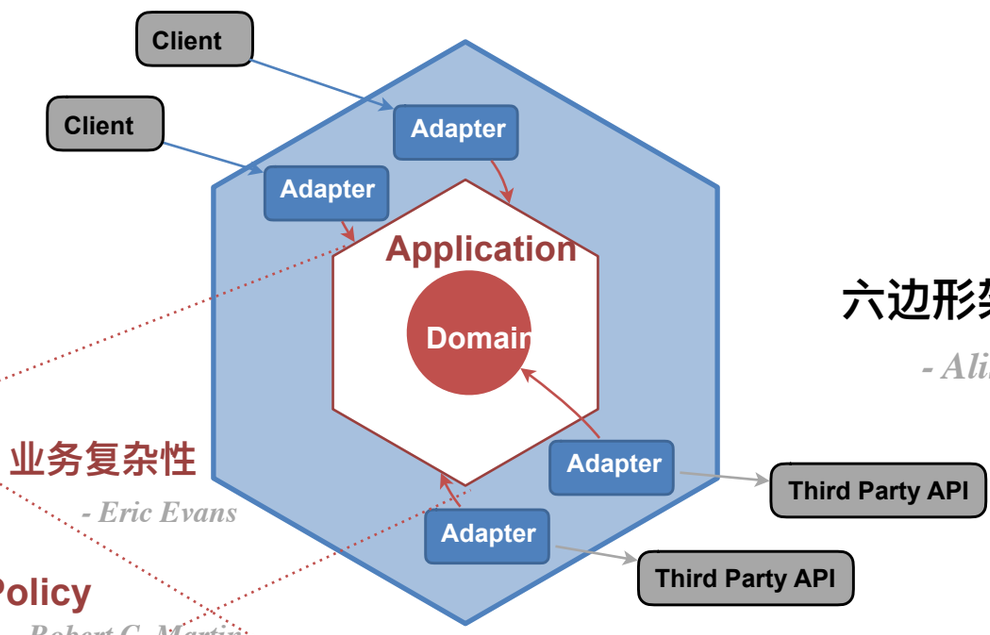
- Robert C. Martin

架构模式 - 应用架构 - 本质相同



分层架构+DIP

- Vaughn Vernon



业务复杂性

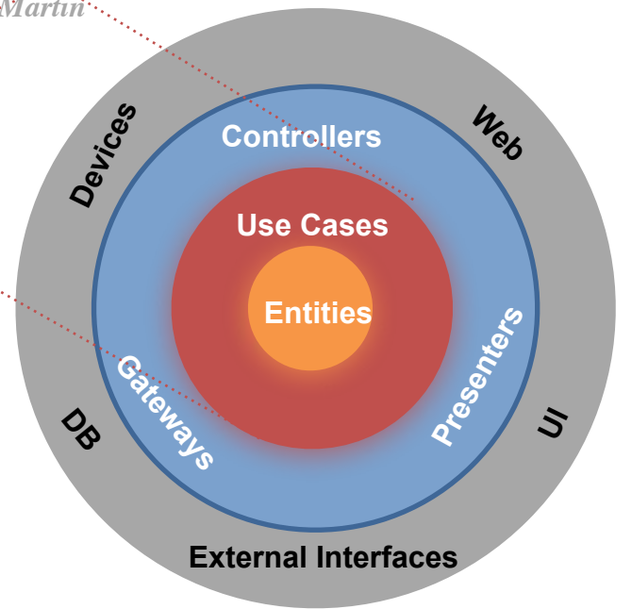
- Eric Evans

六边形架构

- Alistair Cockburn

Policy

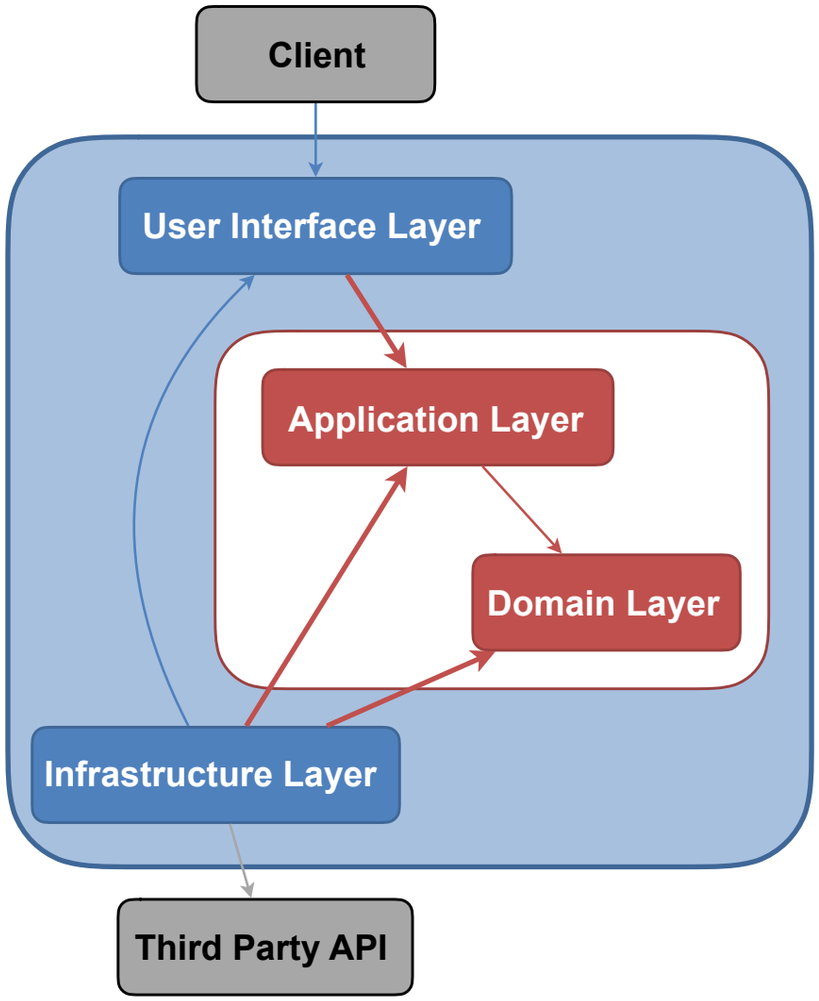
- Robert C. Martin



整洁架构

- Robert C. Martin

架构模式 - 应用架构 - 原则的体现



SRP

Separation of Concerns: separate changes at different rates, for different reasons.

The Axis of Change responsible for the creation of Architectural Boundaries.

LSP

Build Software systems from interchangeable parts.

OCP

Protect Policy from changes in Details

Protect Client from changes in Domain

DIP

Policy depend on Detail → Detail Depend on Policy

谢谢

DDD CHINA