ThoughtWorks®

全球软件咨询公司

DDD CHINA

实战化领域驱动设计工作坊

概念讲解
(DDD China 2019 Conference)

讲师团队

胡皓

王岩

钟健鑫

朱海波

王瑞鹏

林宁

DDD CHINA

ThoughtWorks®

为何需要领域驱动设计？

现实世界的挑战

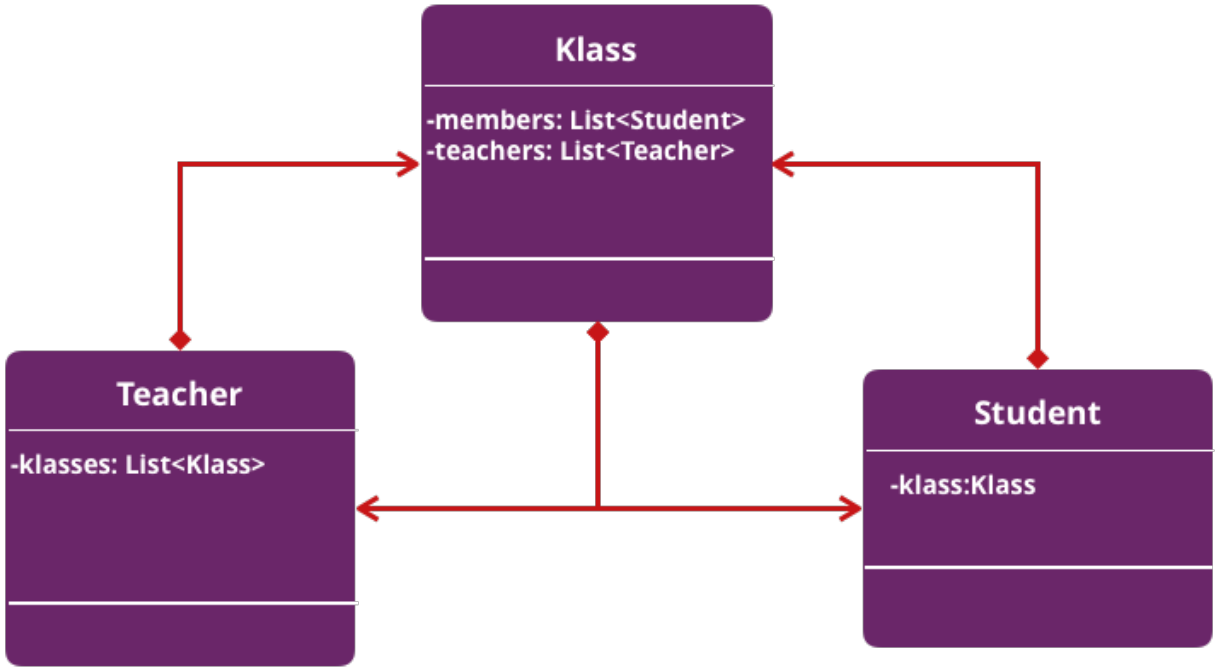
难以理解的代码

编程中最难的事情之一就是命名

```
int setContactInfofun2(ContactInfo contactInfo, int flag, String num)
{
    int len;
    int tmp1 = contactInfo.getCityId();
    String tmp2 = "";

    if (flag == 1) {
        tmp2 = contactInfo.getTelNum2();
        len = getfun(tmp1, tmp2);
        System.out.println( tmp2 + len + "-" + num);
    } else if (flag == 2) {
        tmp2 = contactInfo.getTelNum3();
        len = getfun(tmp1, tmp2);
        System.out.println( tmp2 + len + "-" + num);
    } else if (flag == 3) {
        tmp2 = contactInfo.getTelNum1();
        System.out.println( tmp2 + "-" + num);

    } else {
        return -1;
    }

    return 0;
}
```

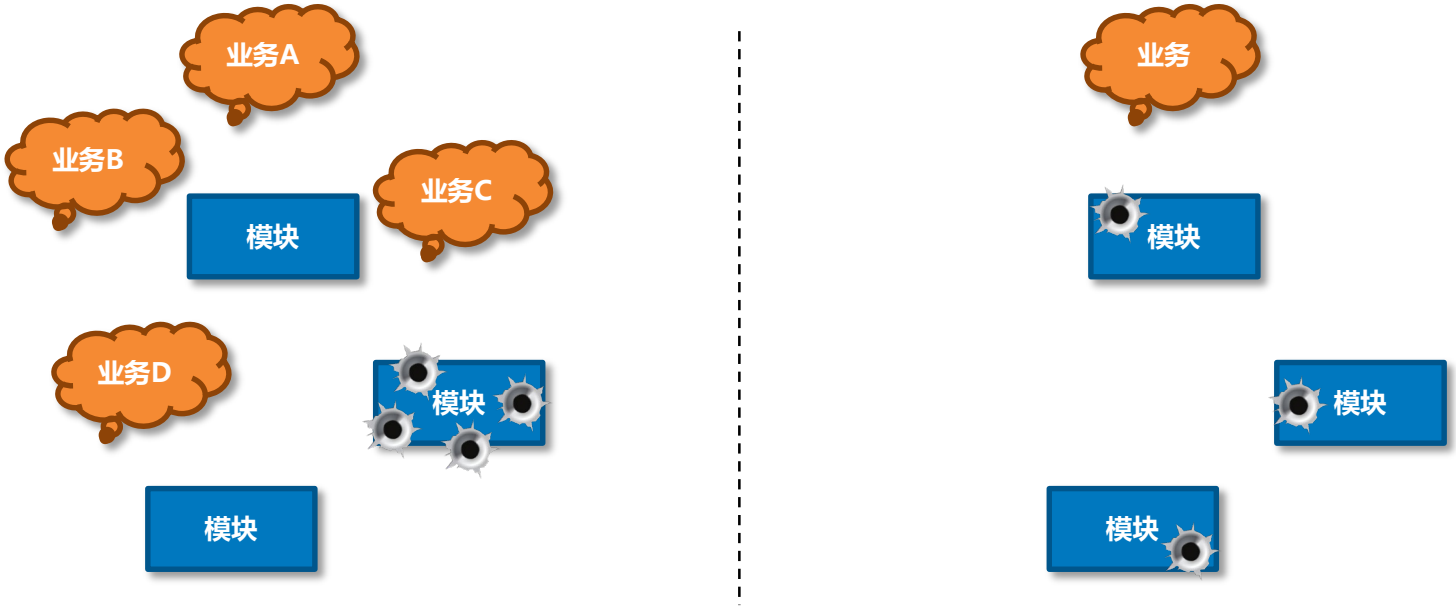
混乱不堪的模型

拍脑袋建模



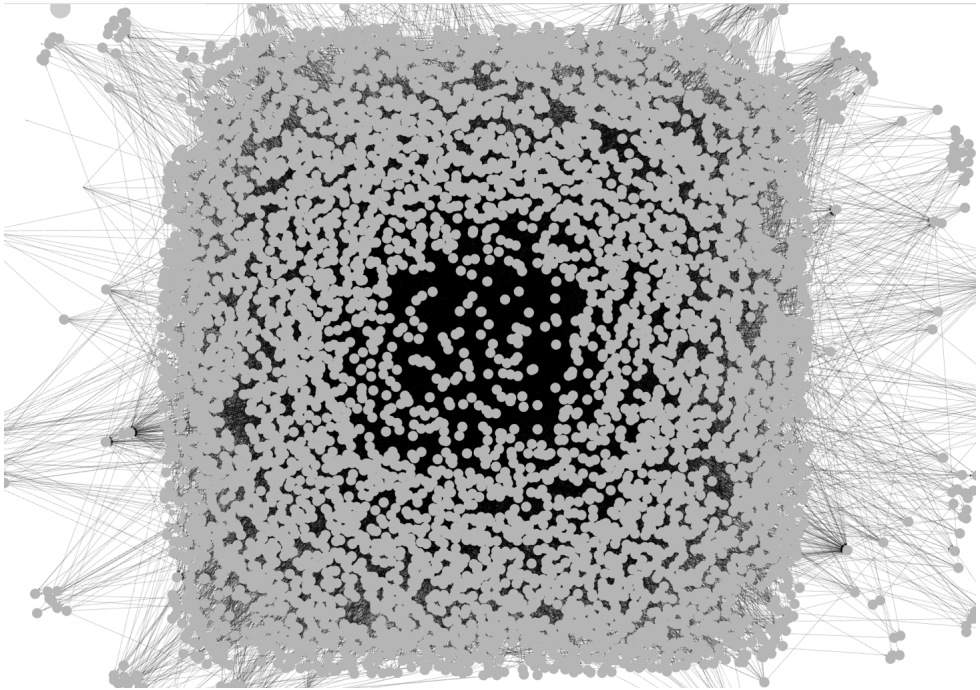
发散式变化 & 散弹式修改

依据变化的边界进行架构和编程——最简单也最困难



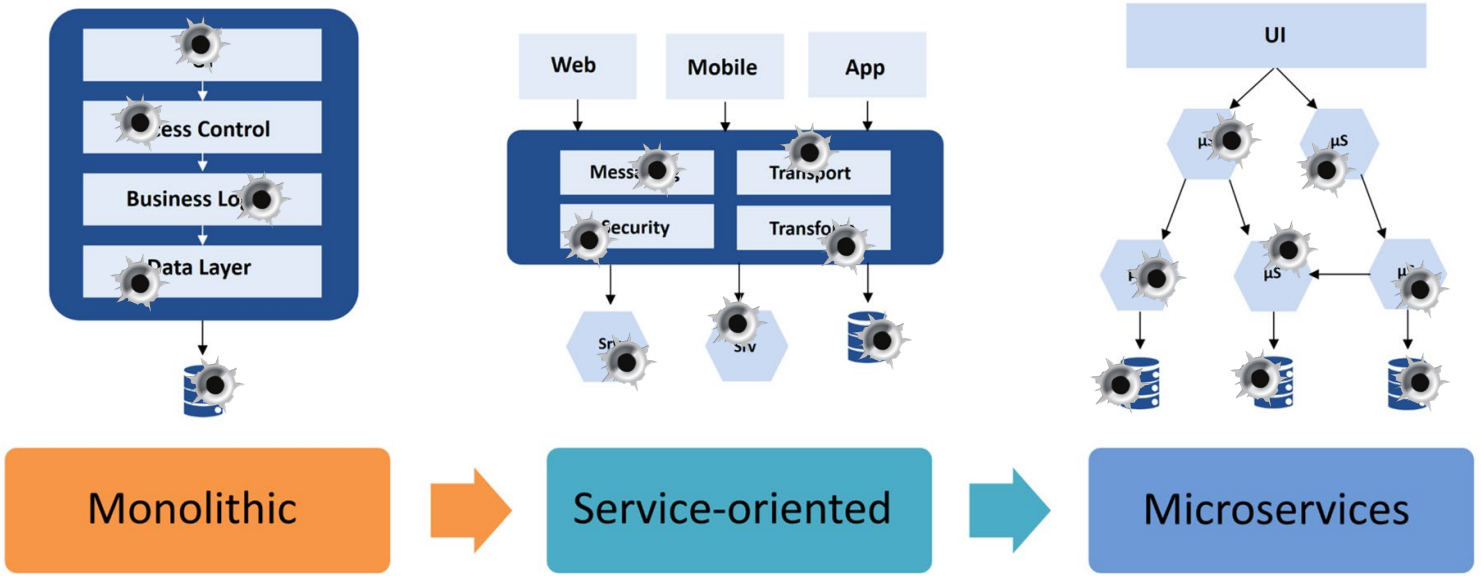
大泥球架构

技术与业务不能匹配的最终结果



微服务压垮了最后一根稻草

从大泥球到分布式泥坑

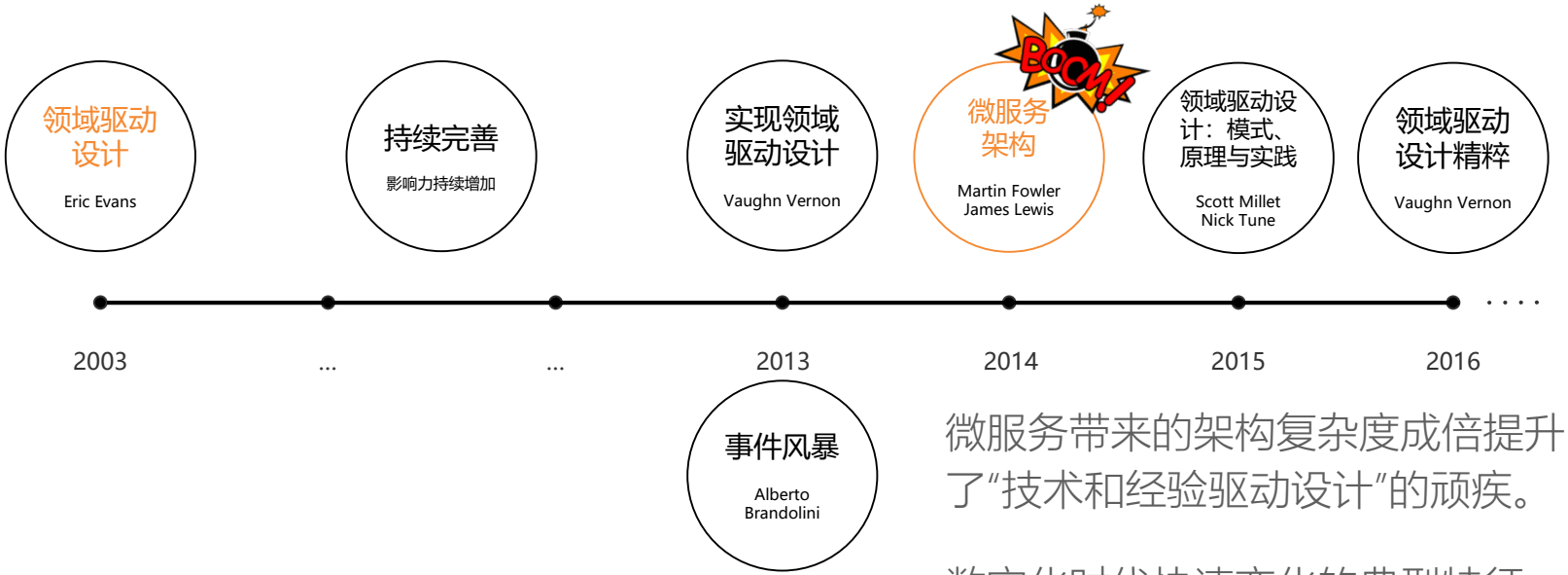


https://maveric-systems.com/blog/microservices-i-microservices-vs-soa

DDD CHINA

领域驱动设计的诞生

领域驱动设计
从低调沉稳到日渐火爆



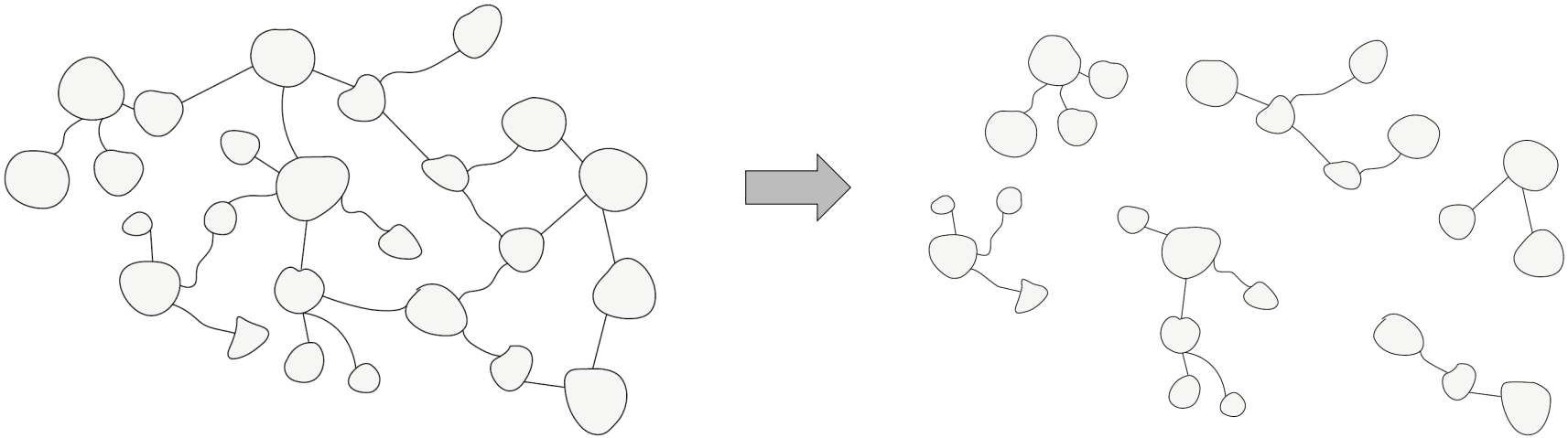
微服务带来的架构复杂度成倍提升，引爆了“技术和经验驱动设计”的顽疾。

数字化时代快速变化的典型特征，进一步敲响了低响应力架构的丧钟。

领域驱动设计解决问题的方式

分解大泥球

以子域、限界上下文为参考，通过聚合的方式进行建模



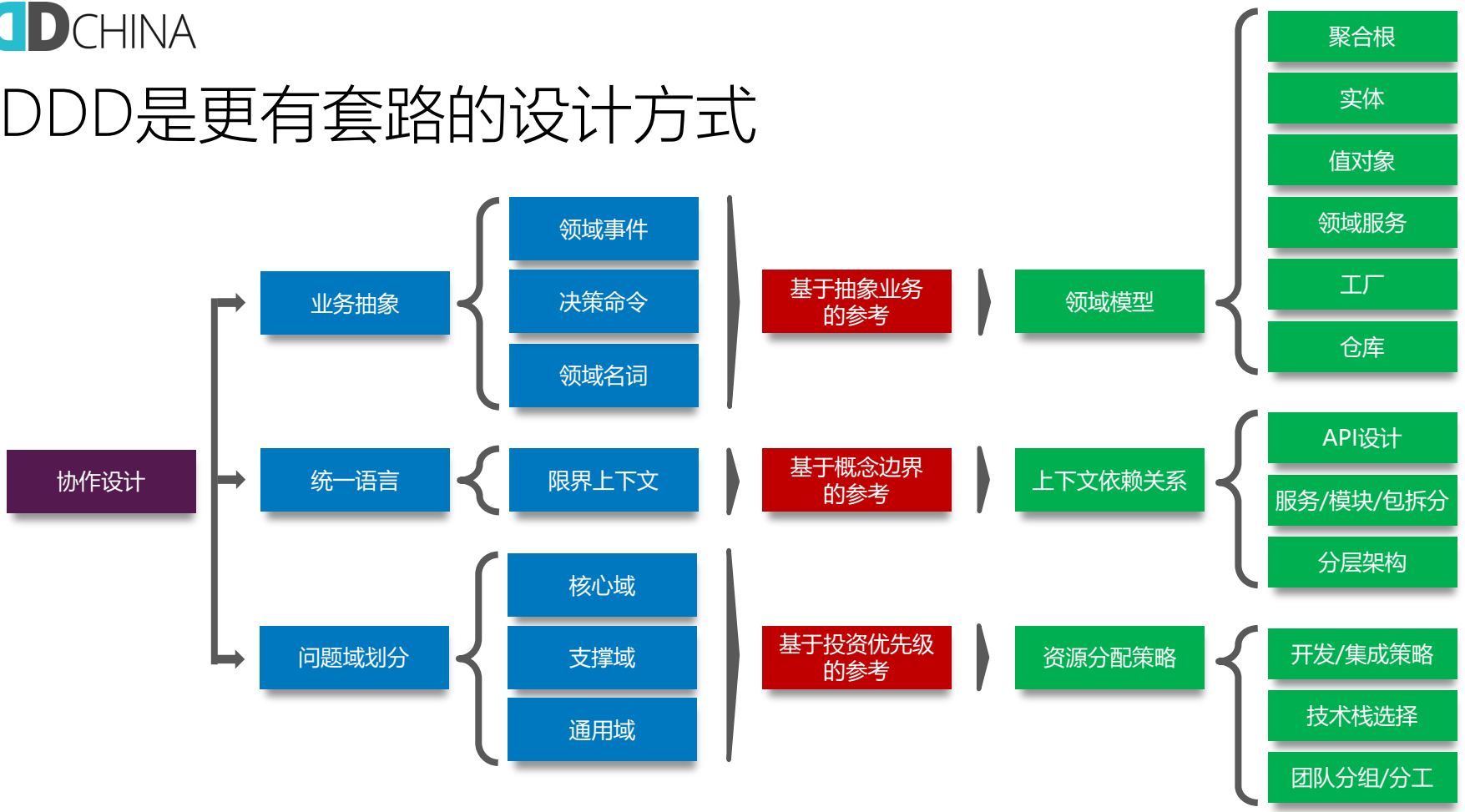
https://p2p.wrox.com/book-patterns-principles-practices-domain-driven-design-789/

DDD与传统设计方法的对比

大泥球架构

松耦合架构

DDD是更有套路的设计方式



核心原则

面向业务进行架构

聚焦核心域

澄清问题域，聚焦核心竞争力，优化资源投入

协作设计

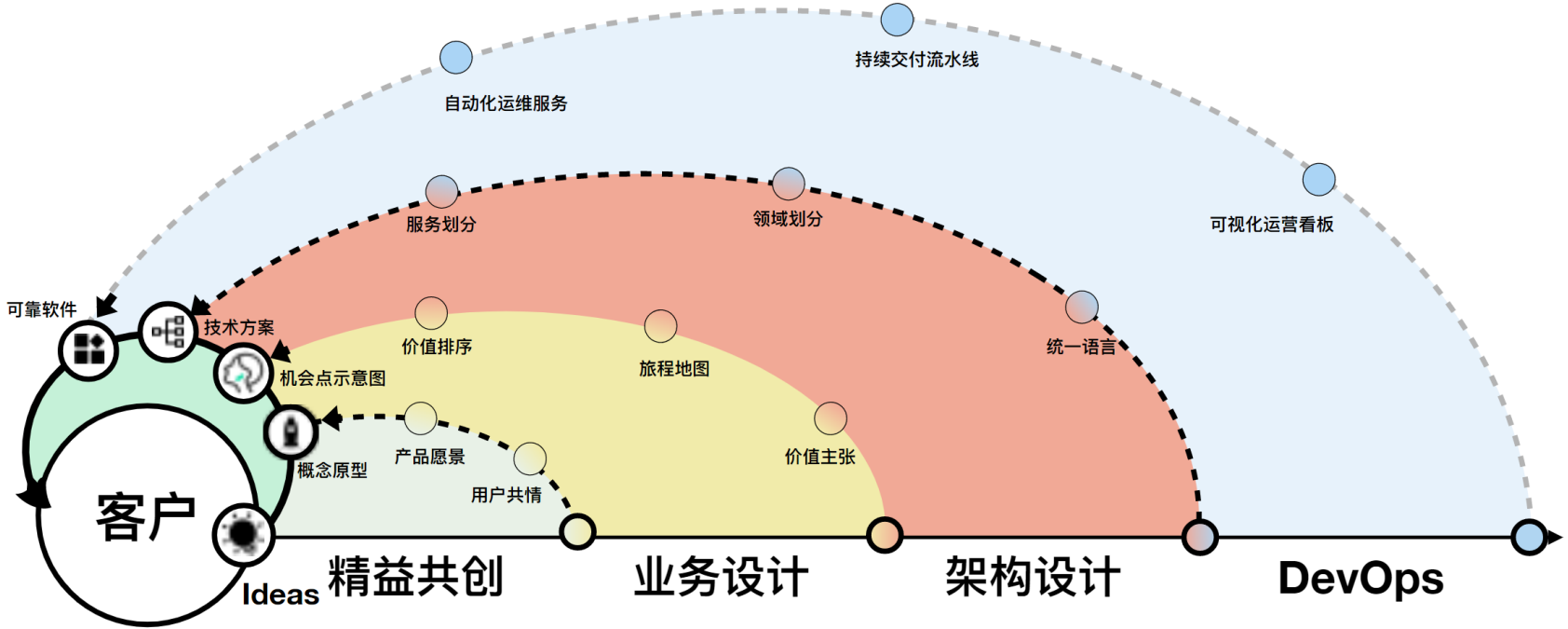
领域专家同软件专家通过创造性协作，迭代式的探索和发现模型

统一语言

利用明确且有边界的业务上下文统一语言

DDD在产品研发过程中的位置

Design Thinking + Domain Driven Design + DevOps



DDD CHINA

ThoughtWorks®

如何开展领域驱动设计？

分段式协作设计

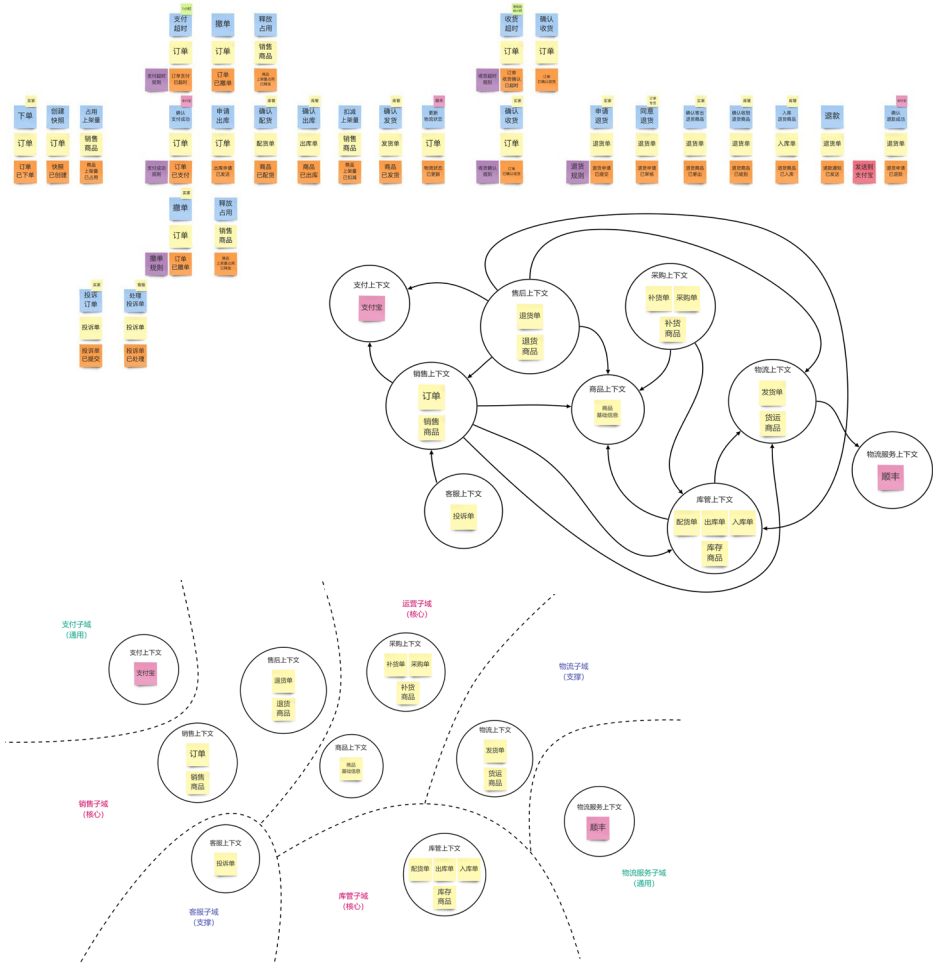
分段式协作设计
三个阶段

从问题出发，逐级抽象，层层深入和细化

- 战略设计
- 战术设计
- 技术实现

DDD CHINA

分段式协作设计
战略设计阶段



多说问题，少说方案

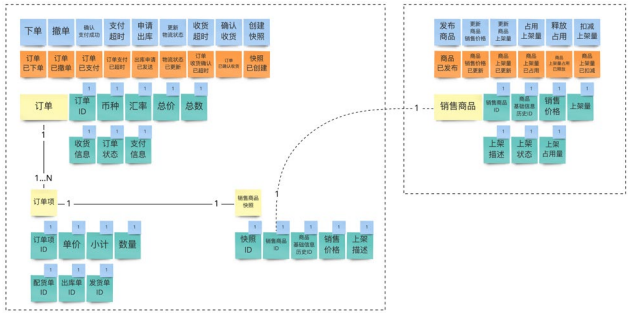
- 业务梳理和抽象
- 限界上下文识别
- 子域识别

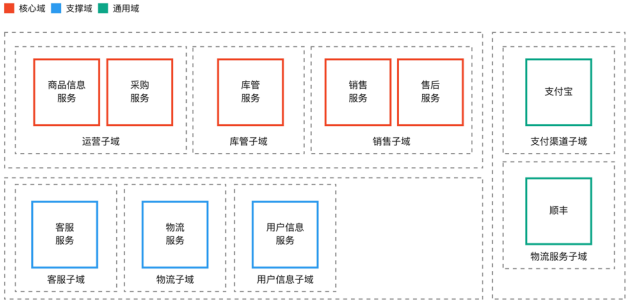
DDD CHINA

分段式协作设计
战术设计阶段

持续抽象，忽略技术细节

- 领域建模
- 业务服务识别
- 业务服务API能力识别

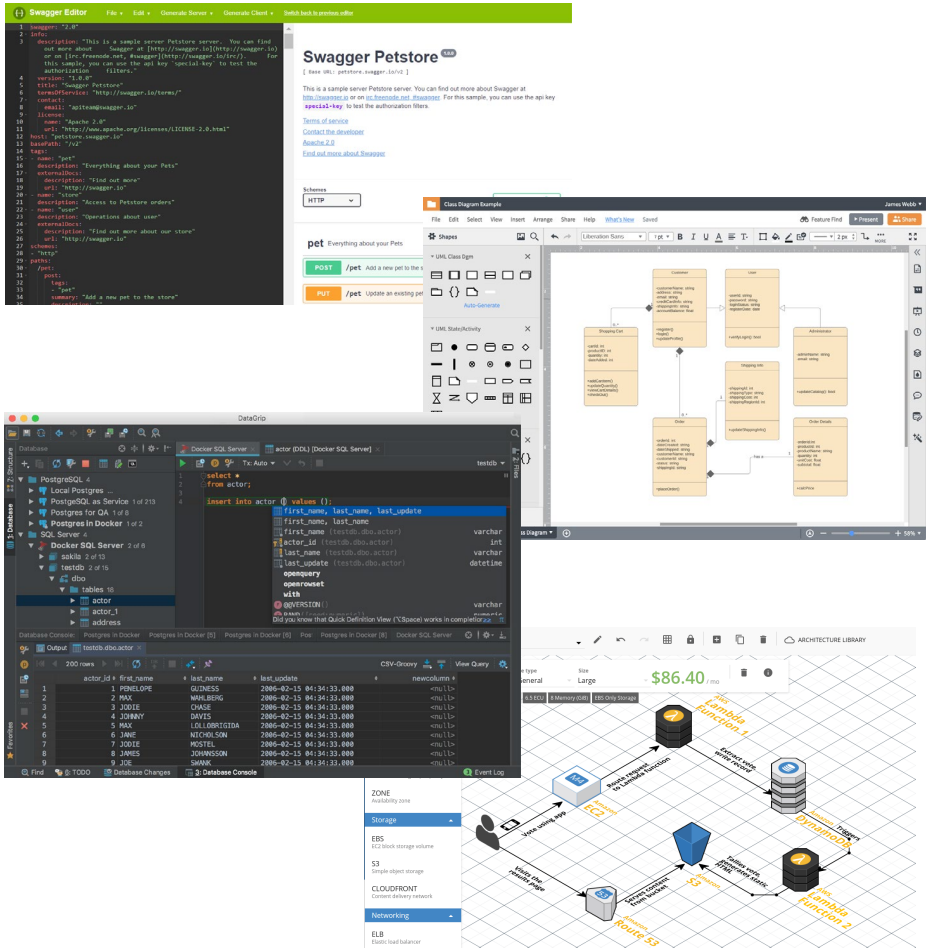




“销售服务”接口能力			
限界上下文	**聚合**	**接口能力**	**读写类型**
销售上下文	订单	下单	写
		获取订单	读
		撤单	写
		.....	.....
	销售商品	发布商品	写
		获取商品	读
		.....	.....

DDD CHINA

分段式协作设计
技术实现阶段



一切皆是细节

- API详细设计
- UML设计
- 数据库设计
- 部署与运维
- ……

DDD CHINA

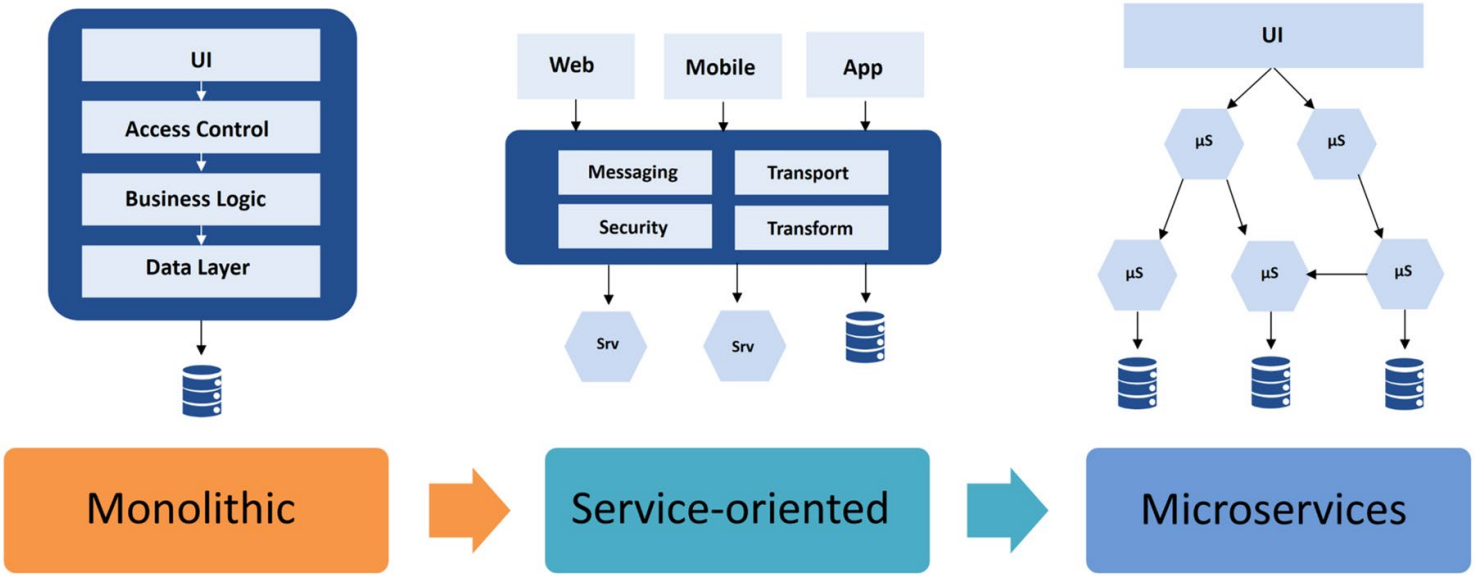
ThoughtWorks®

如何实现领域驱动设计？

松耦合架构

微服务设计

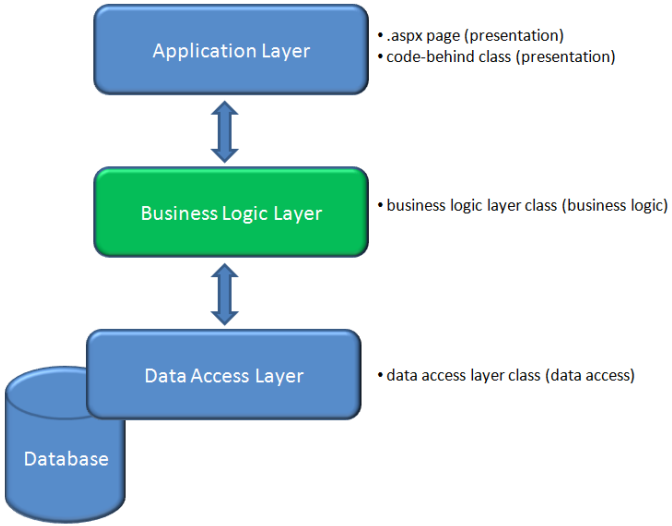
以业务边界为参考，实现架构与业务对齐



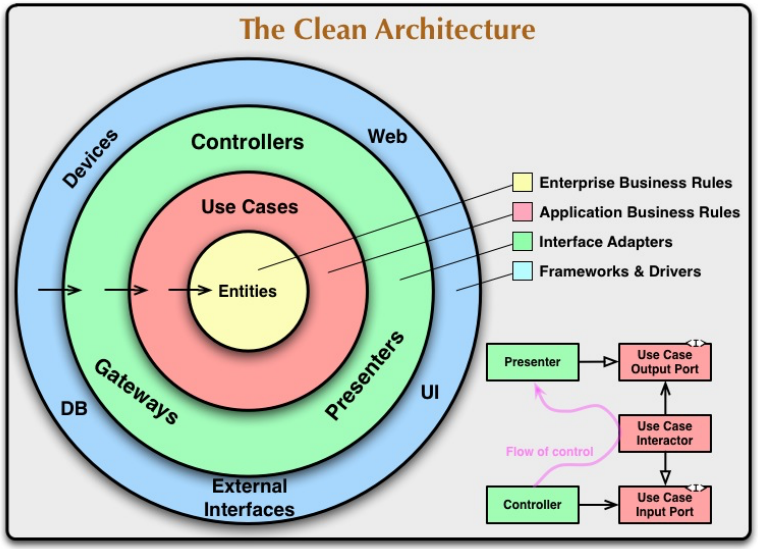
https://maveric-systems.com/blog/microservices-i-microservices-vs-soa

分层架构设计

以领域为核心，以变化的原因和聚合为边界，提供分层守护



https://maveric-systems.com/blog/microservices-i-microservices-vs-soa



https://blog.cleancoder.com/uncle-bob/2012/08/13/the-clean-architecture.html

代码实现样例

Talk is cheep, show me the code.

https://github.com/howiehu/ddd-architecture-samples

DDD CHINA

ThoughtWorks®

工作坊开始

通过假想的业务实际操作和体验领域驱动设计

一个简单粗暴的需求

嘿！我是你们的老板！我很有钱！我需要做一个产品干掉市场上那些倒霉的外卖产品！

电梯演讲 (Elevator Pitch)

- 使用便利贴在白纸上贴出电梯演讲的结构。
- 每个人为电梯演讲的每一行想象一个内容，通过便利贴的方式进行书写（每个人应该写出7个便利贴）。
- 每个人将自己的7个便利贴放在白纸上相应的位置，共同阅读并讨论以形成共同意见，让电梯演讲看上去更有吸引力，更可行且更通顺，必要的时候可以采取投票等方式达成一致（这只是一个练习，所以大家开心优于纠结😄）。
- 利用便利贴修改并调整电梯演讲到最终结果，然后每组将自己的电梯演讲分享给大家。

DDD CHINA

FOR [*target customer*]

WHO [*statement of the need or opportunity*]

THE [*product name*]

IS A [*product category*]

THAT [*key benefit, compelling reason to use*]

UNLIKE [*primary competitive alternative*]

OUR PRODUCT [*statement of primary differentiation*].

DDD CHINA

ThoughtWorks®

胡皓
高级技术顾问
hhu@thoughtworks.com | thoughtworks.com/profiles/hao-hu