微服务，设计先行

郑晔
推文科技 技术 VP

QCon+ 案例研习社

扫码了解 QCon+

学习前沿案例，向行业领先迈进

40 个
热门专题
—
行业专家把关内容筹备，
助你快速掌握最新技术发展趋势

200 个
实战案例
—
了解大厂前沿实战案例，
为 200 个真问题找到最优解

40 场
直播答疑
—
40 位技术大咖，每周分享最新
技术认知，互动答疑

365 天
持续学习
—
结合配套 PPT，学习社群引导，
畅学 365 天

程序员

写代码超过 20 年

Moco

专栏作者

10x 程序员工作法

软件设计之美

创业者

网文出海

自我介绍

你听说的微服务

独立部署

解决复杂性

不同的语言

独立扩展

单独的团队

你学习的微服务


服务注册与发现
负载均衡
服务网关
熔断与限流
Spring Cloud
Dubbo
微服务治理
配置中心
分布式事务
gRPC
服务调用追踪

实际中的微服务

不好测试

分布式事务

性能差

数据难一致

服务依赖

问题出在哪了？

微服务不是问题，设计才是

设计不好的问题，微服务解决不了

最好的设计是单体设计

易于测试

数据一致

一次性部署

进程内调用

微服务为何兴起?
复杂度

对抗软件复杂度
软件设计

解决复杂度问题
只能依赖于软件设计
微服务只是一种部署方案

今天怎么做设计?
领域驱动开发（DDD）

通用语言

模型驱动设计

战略设计

战术设计

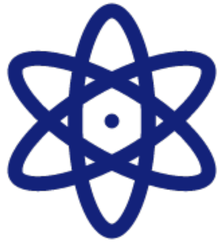
理解 DDD

开发人员

通用语言

业务人员

通用语言：同样的业务语言



核心域

支撑域

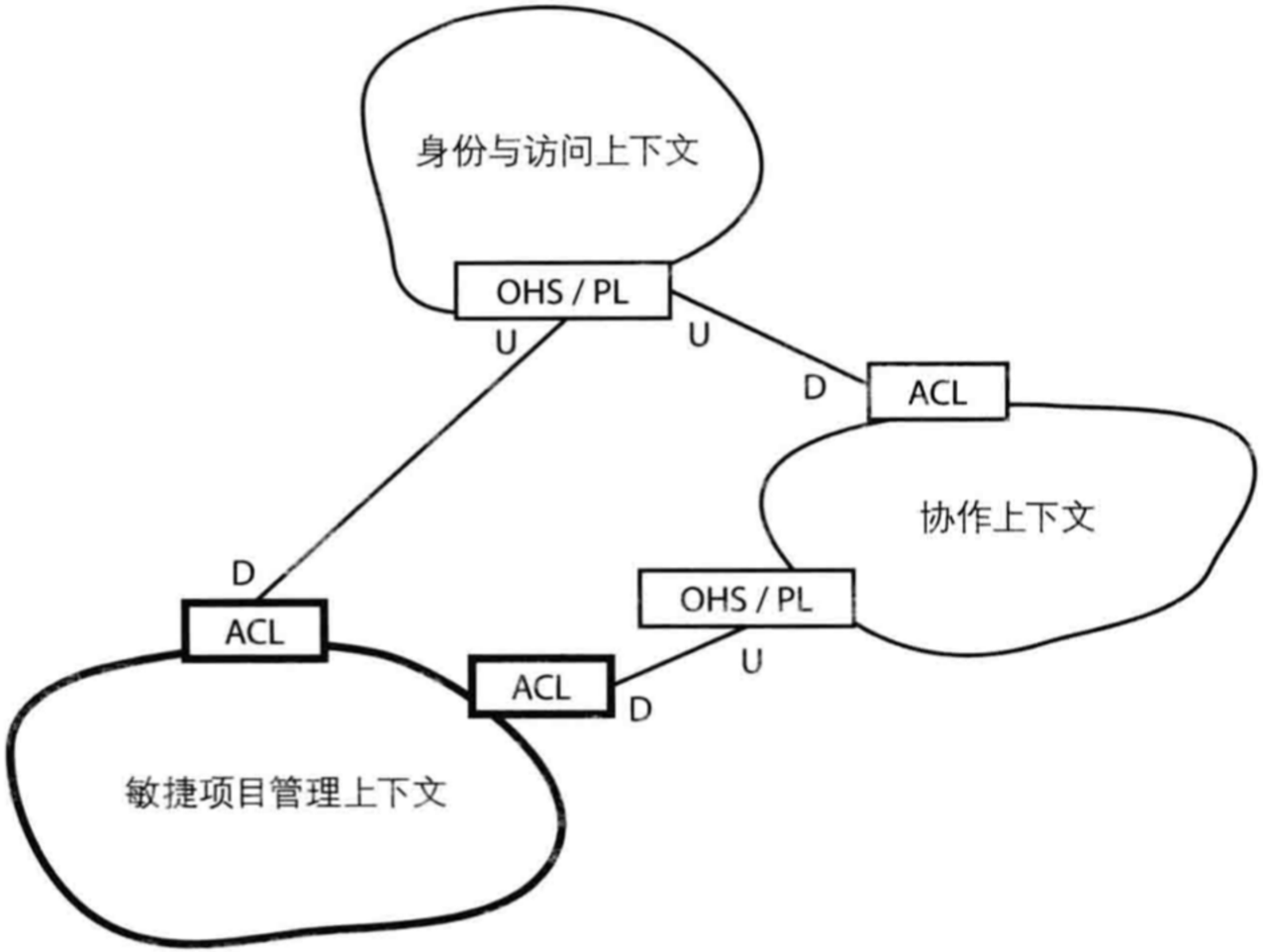
通用域

战略设计：切分子域

战略设计：组织子域

限界上下文

上下文映射图



角色

实体

值对象

关系

聚合

聚合根

互动

工厂

仓库

领域服务

应用服务

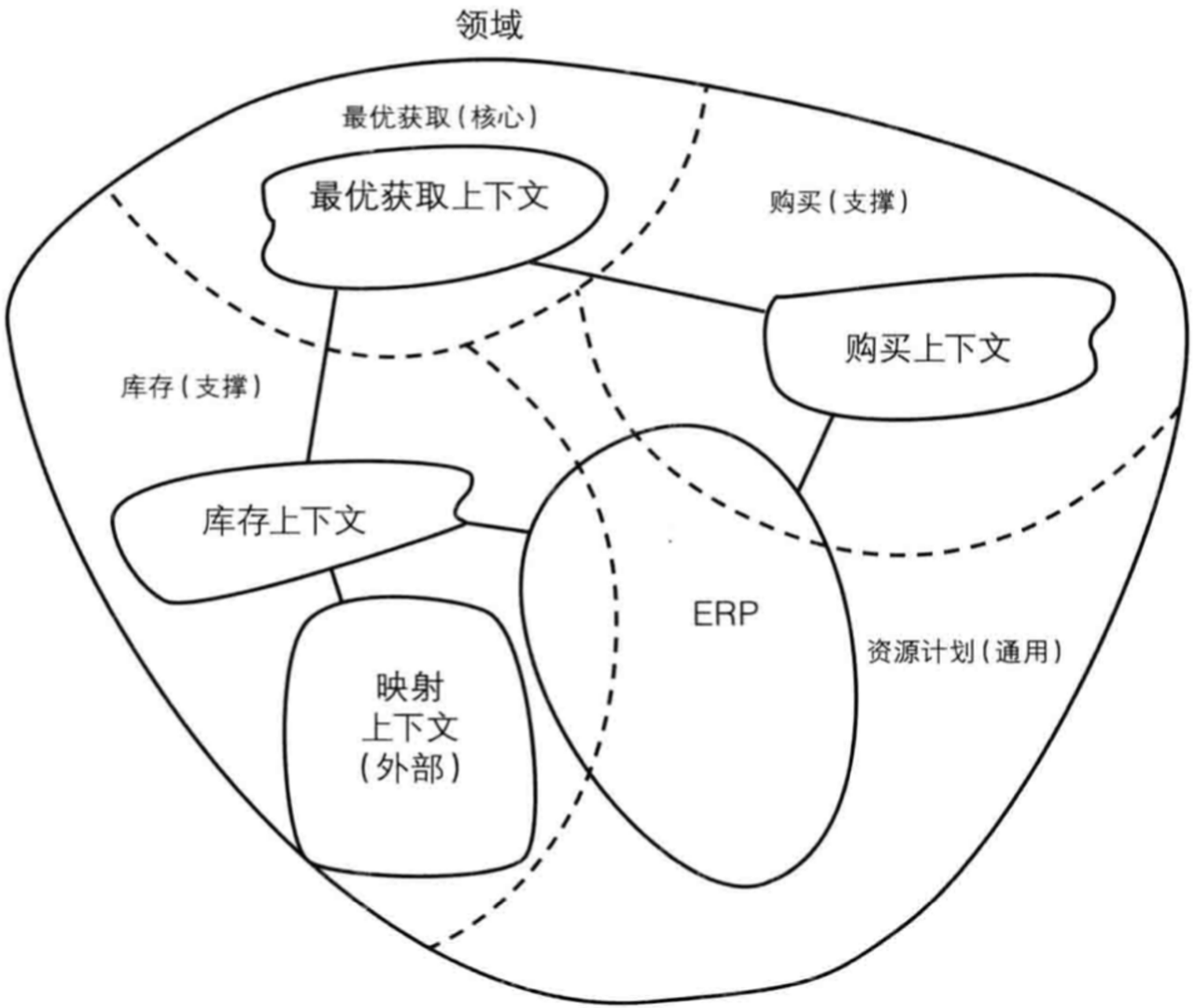
战术设计：按模板找对象

这和微服务有什么关系?

限界上下文

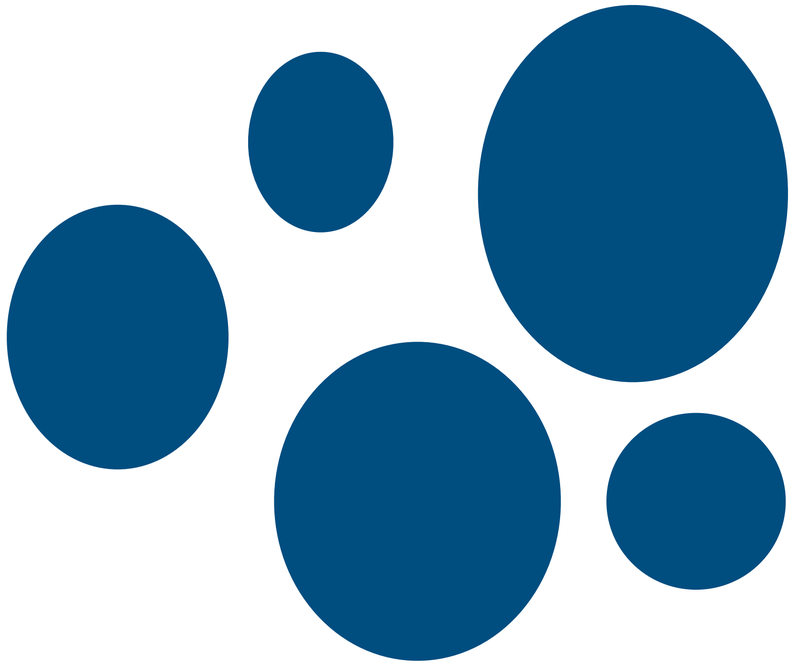
一个划分服务的恰当边界

完全独立的限界上下文



微服务是设计的结果

不是追求的目标

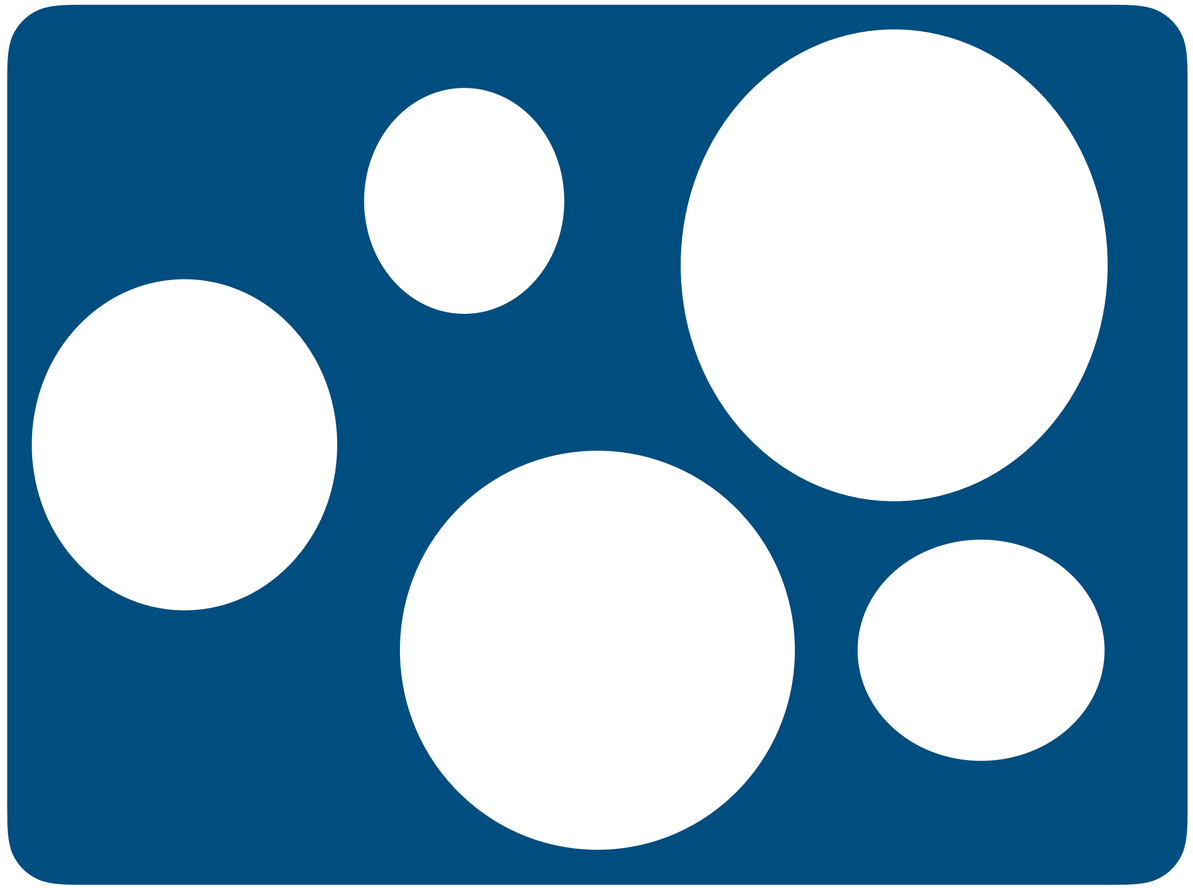


一次正常的“微”服务开发过程

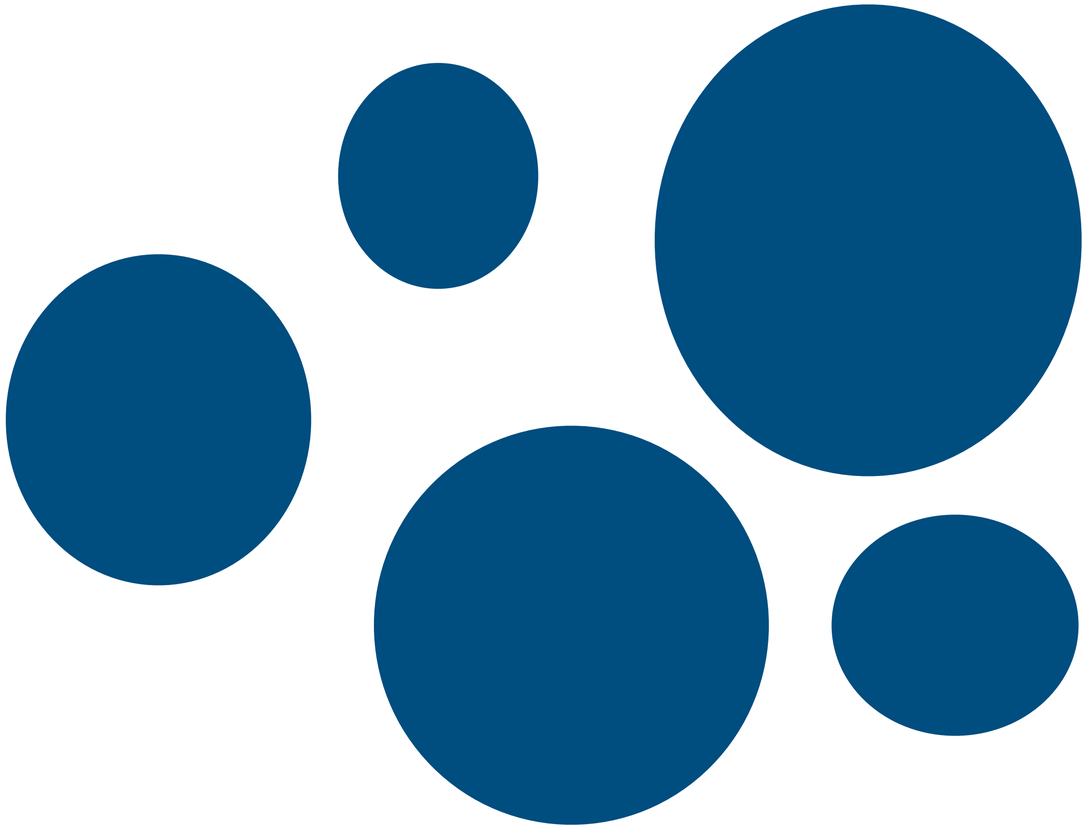
1. 开发一个单体应用

2. 按照战略设计，划分限界上下文

3.把不同的限界上下文划分到模块中



3.当业务成型之后，按需拆分服务



总结

- 微服务，理想很丰满，现实很骨感
- 微服务要解决复杂度的问题，但复杂度真正的解决只能依赖于设计
- 今天主流的设计方式是 DDD，包括建立通用语言、战略设计和战术设计
- 战略设计中的限界上下文是很好的服务边界
- 设计一个微服务，可以从设计一个单体应用开始演化

10x程序员工作法

软件设计之美

谢谢