

微服务的 理论模型与现实路径

京东·成都研究院

胡 峰

目录

- 定义
- 起源
- 特征
- 实施
- 角色

定义

- **小**： 微服务架构即是采用一组小服务来构建应用的方法。
- **独立进程**： 运行在独立的进程中，不同服务通过一些轻量级交互机制来通信
- **自动化**： 服务围绕业务能力来构建，并依赖自动部署机制来独立部署。

起源 - UNIX 哲学

- 小即是美
- 一个程序只做好一件事
- 尽可能早地创建原型
- 可移植性比效率更重要



微服务
就像把 UNIX 哲学应用到了分布式系统

起源 - SOA

You should instead think of **Microservices** as a specific approach for **SOA** in the same way that **XP** or **Scrum** are specific approaches for **Agile** software development.

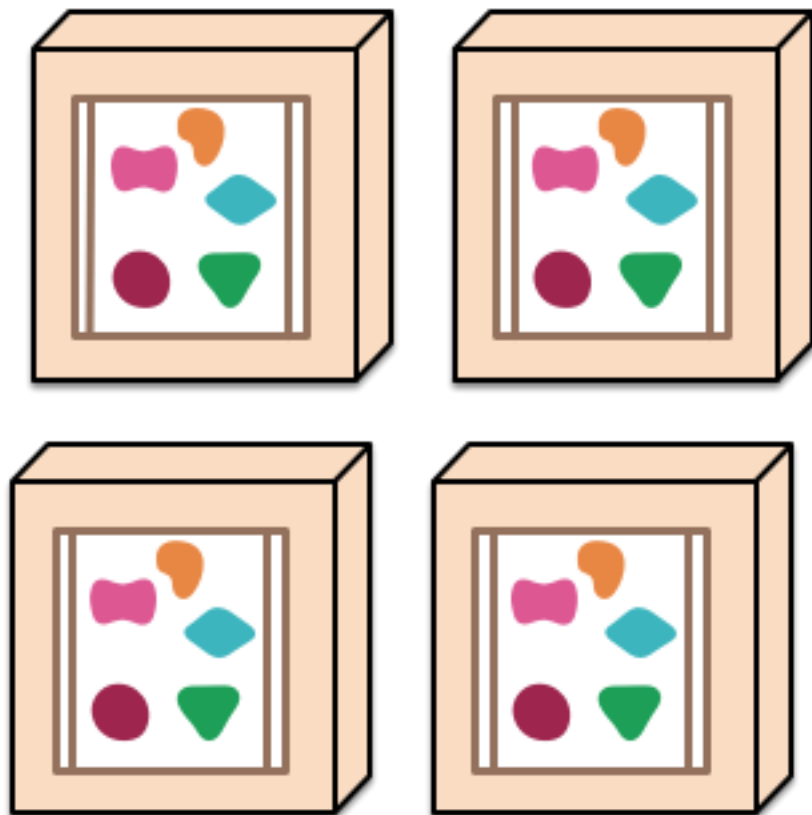
– *Building Microservices* by Sam Newman

特征 - 组件服务化

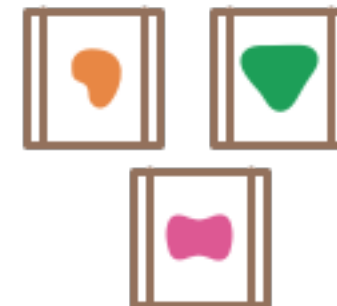
A monolithic application puts all its functionality into a single process...



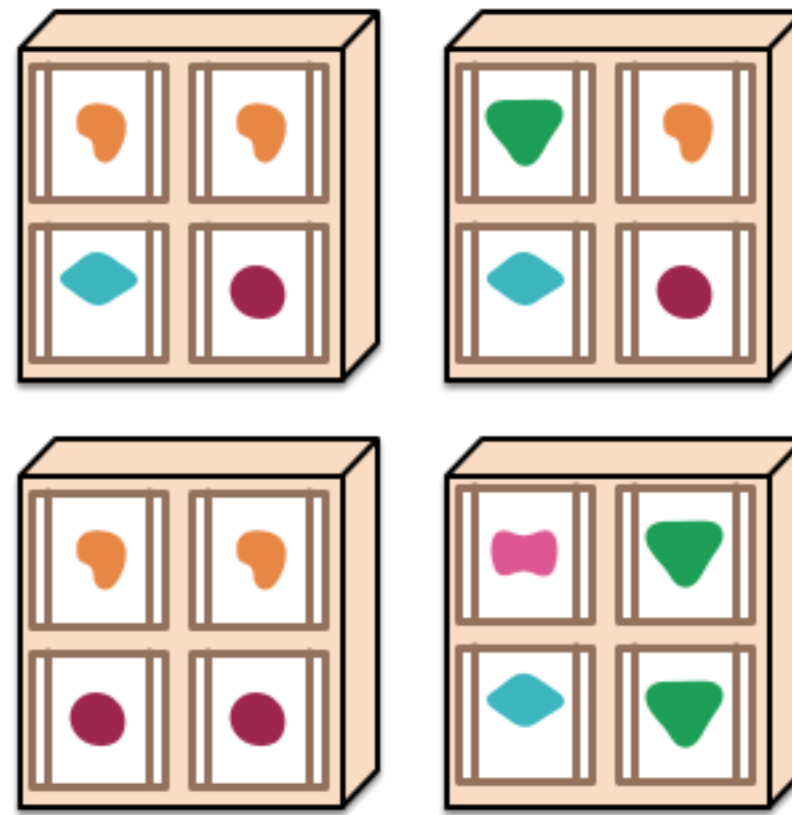
... and scales by replicating the monolith on multiple servers



A microservices architecture puts each element of functionality into a separate service...



... and scales by distributing these services across servers, replicating as needed.



特征 - 按业务能力组织服务

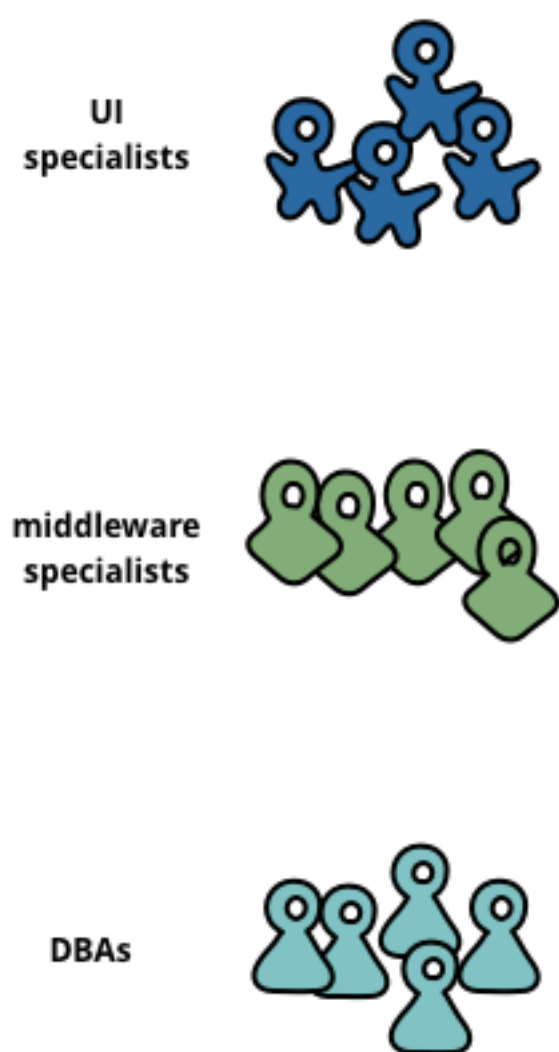
Organizations which design systems are constrained to produce designs which are copies of the communication structures of these organizations.

– Melvin Conway

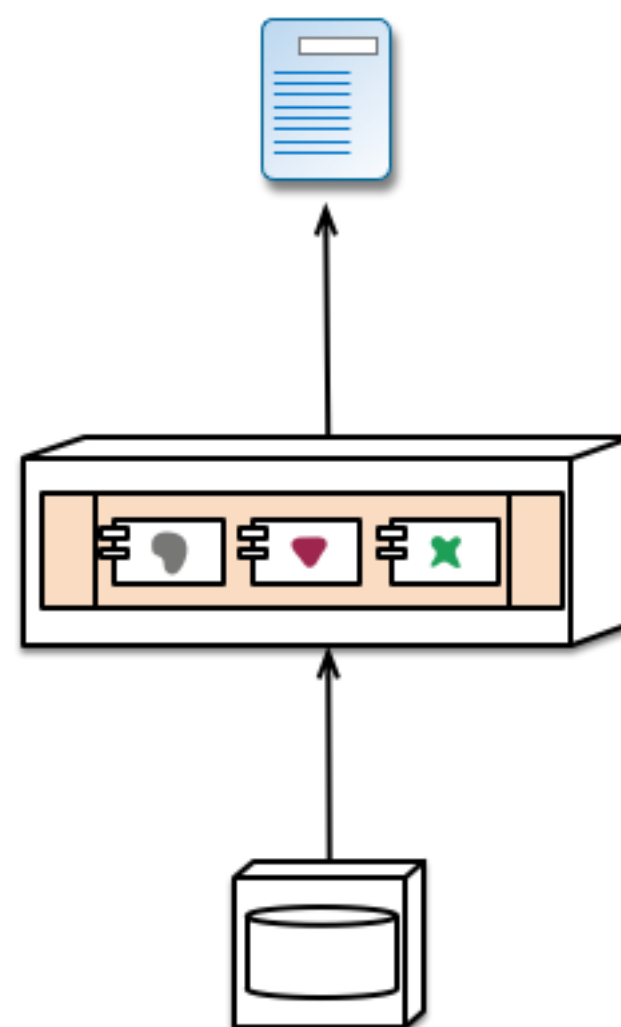
A red speech bubble with a tail pointing towards the text '– Melvin Conway'.

康威定律

特征 - 按业务能力组织服务

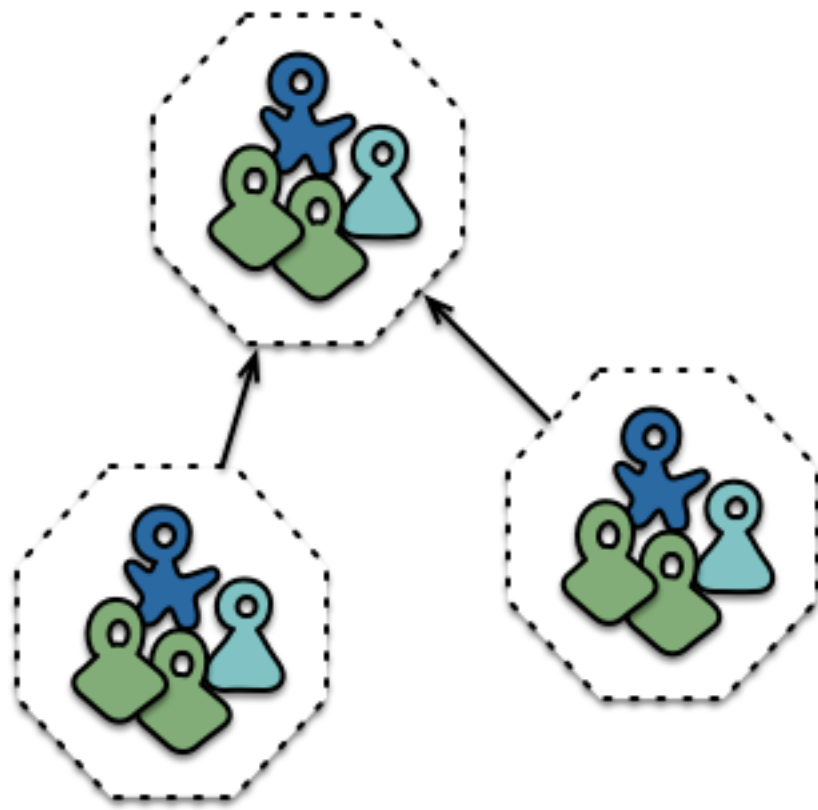


Siloed functional teams...

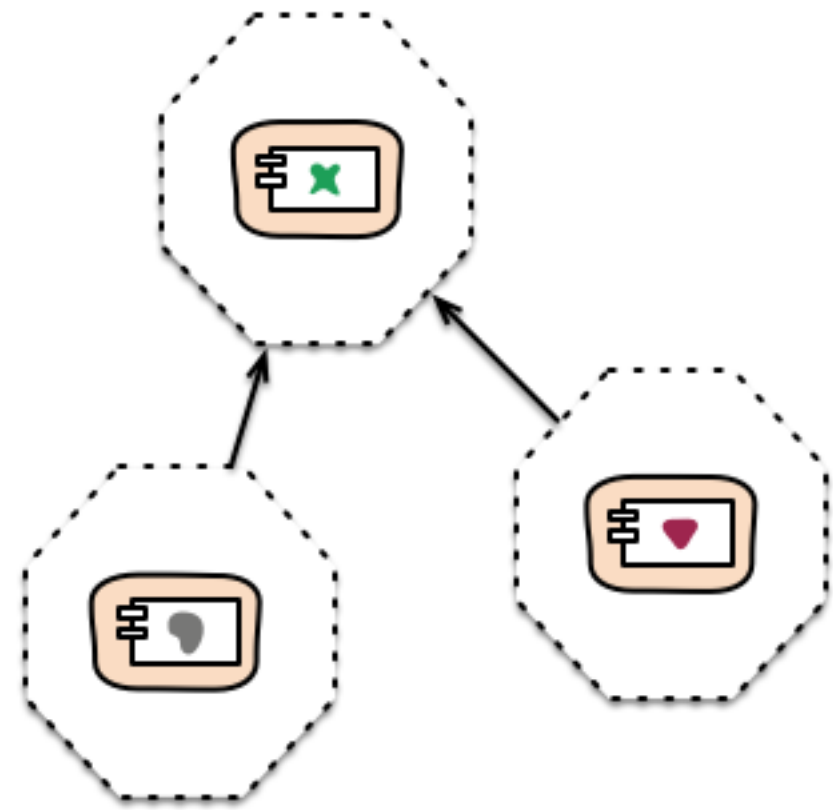


... lead to siloed application architectures.
Because Conway's Law

特征 - 按业务能力组织服务



Cross-functional teams...



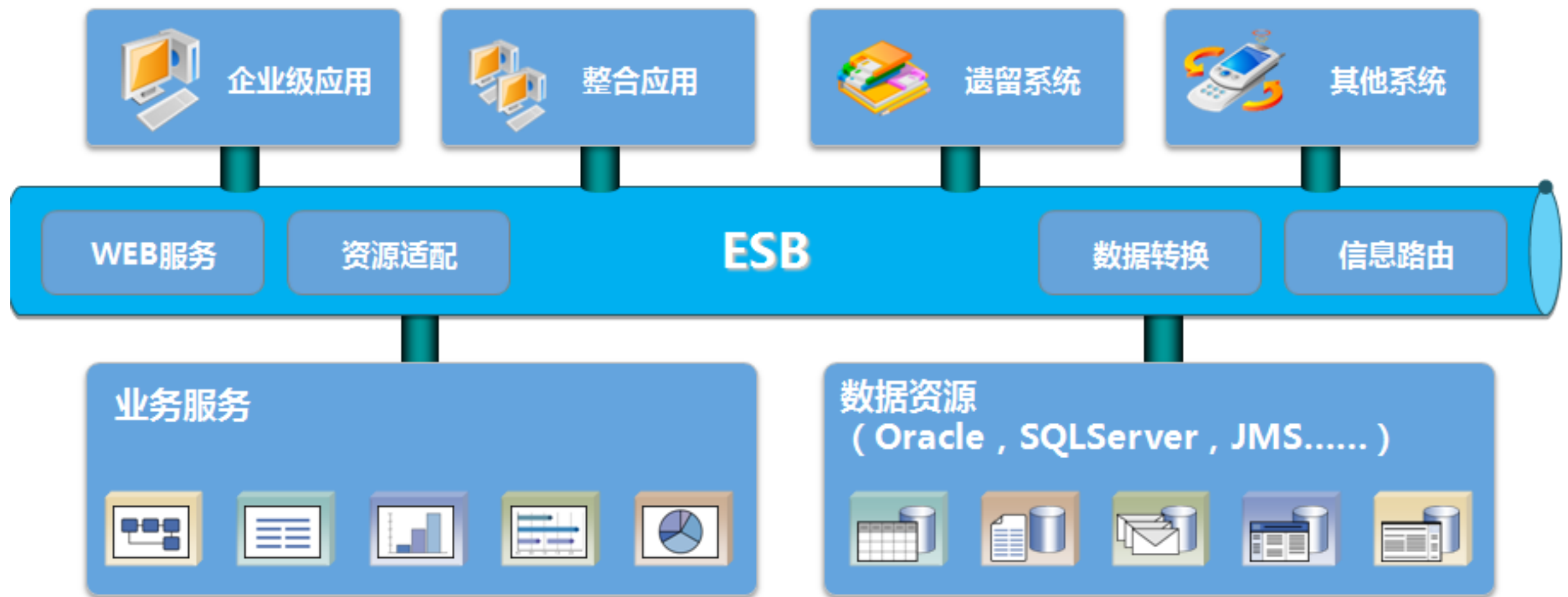
... organised around capabilities
Because Conway's Law

特征 - 服务即产品

You build it, you run it.

– Amazon

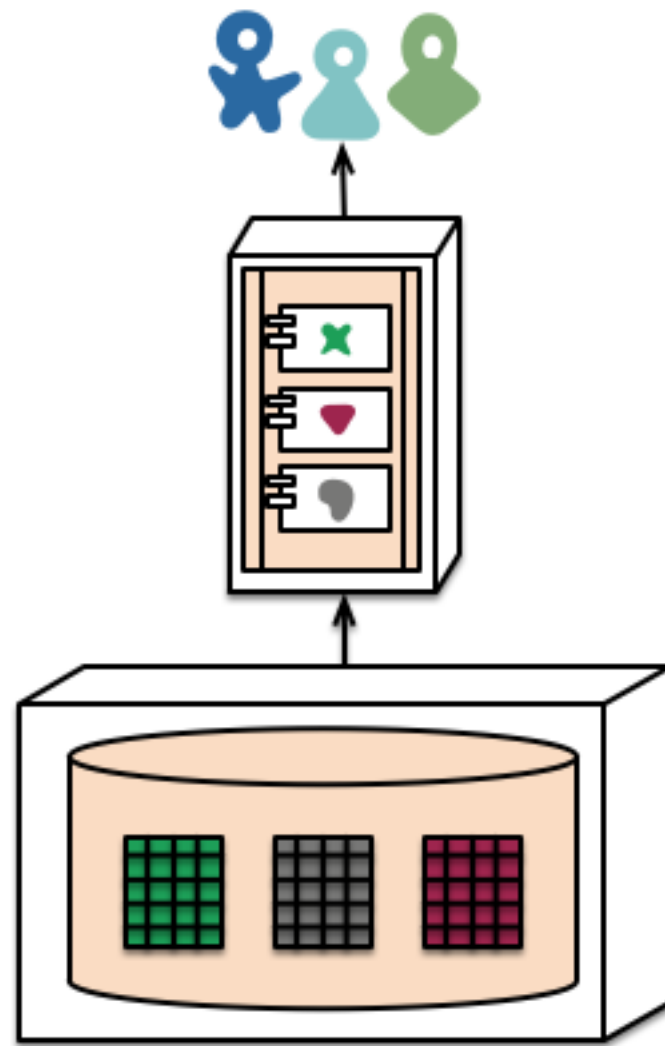
特征 - 智能终端与哑管道



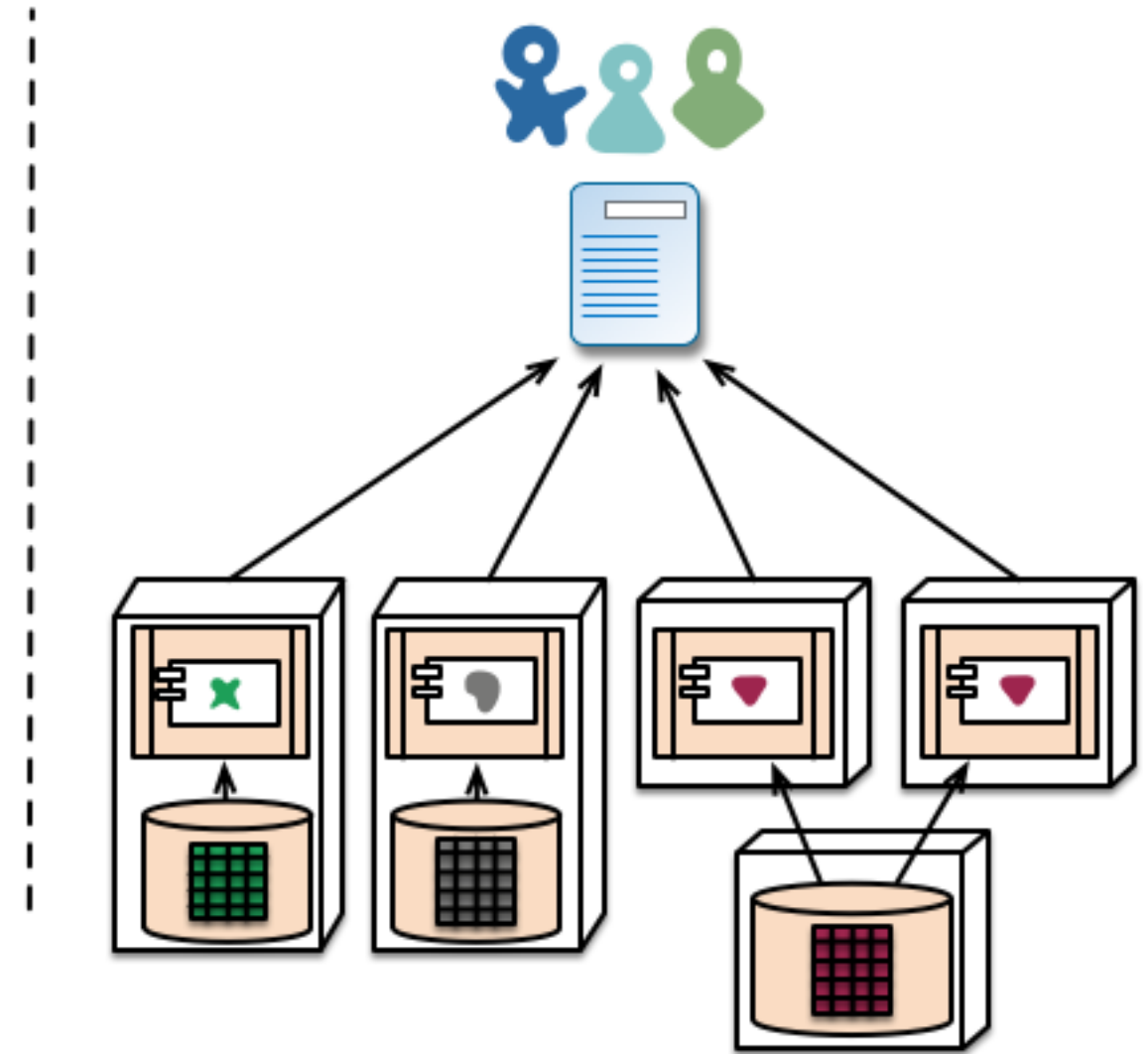
微服务架构抛弃了 ESB 过度复杂的业务规则编排、消息路由等

特征 - 去中心化

- 技术栈
- 数据存储



monolith - single database



microservices - application databases

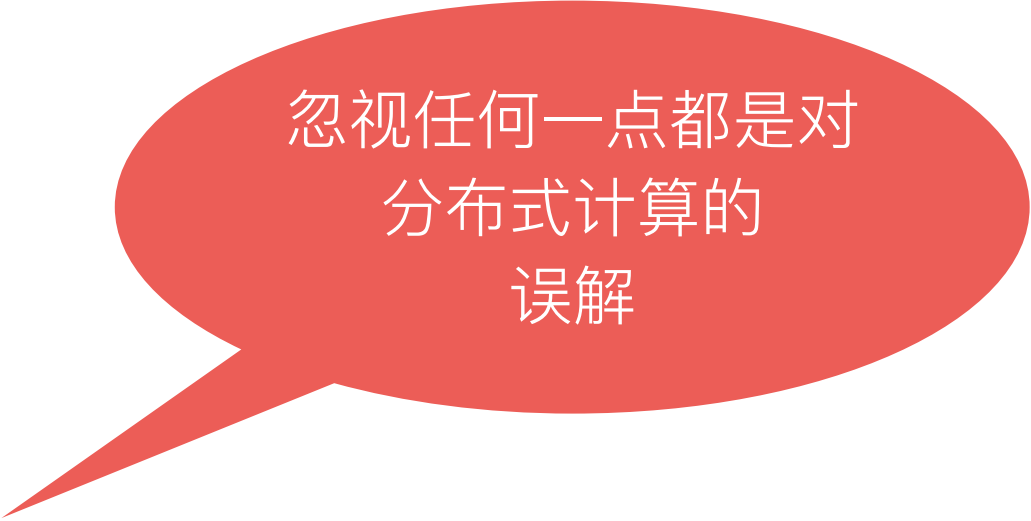
特征 - 基础设施自动化



- 1 个应用, 1台主机 => 部署复杂度 $O(1)$
- 1 个应用, 200台主机 => 部署复杂度 $O(200)$
- 1 个应用, 拆分为 50 个微服务, 再部署 200 台主机 => 部署复杂度 $O(10000)$

特征 - 容错设计

- 网络延迟
- 消息格式
- 负载均衡
- 错误处理



忽视任何一点都是对
分布式计算的
误解

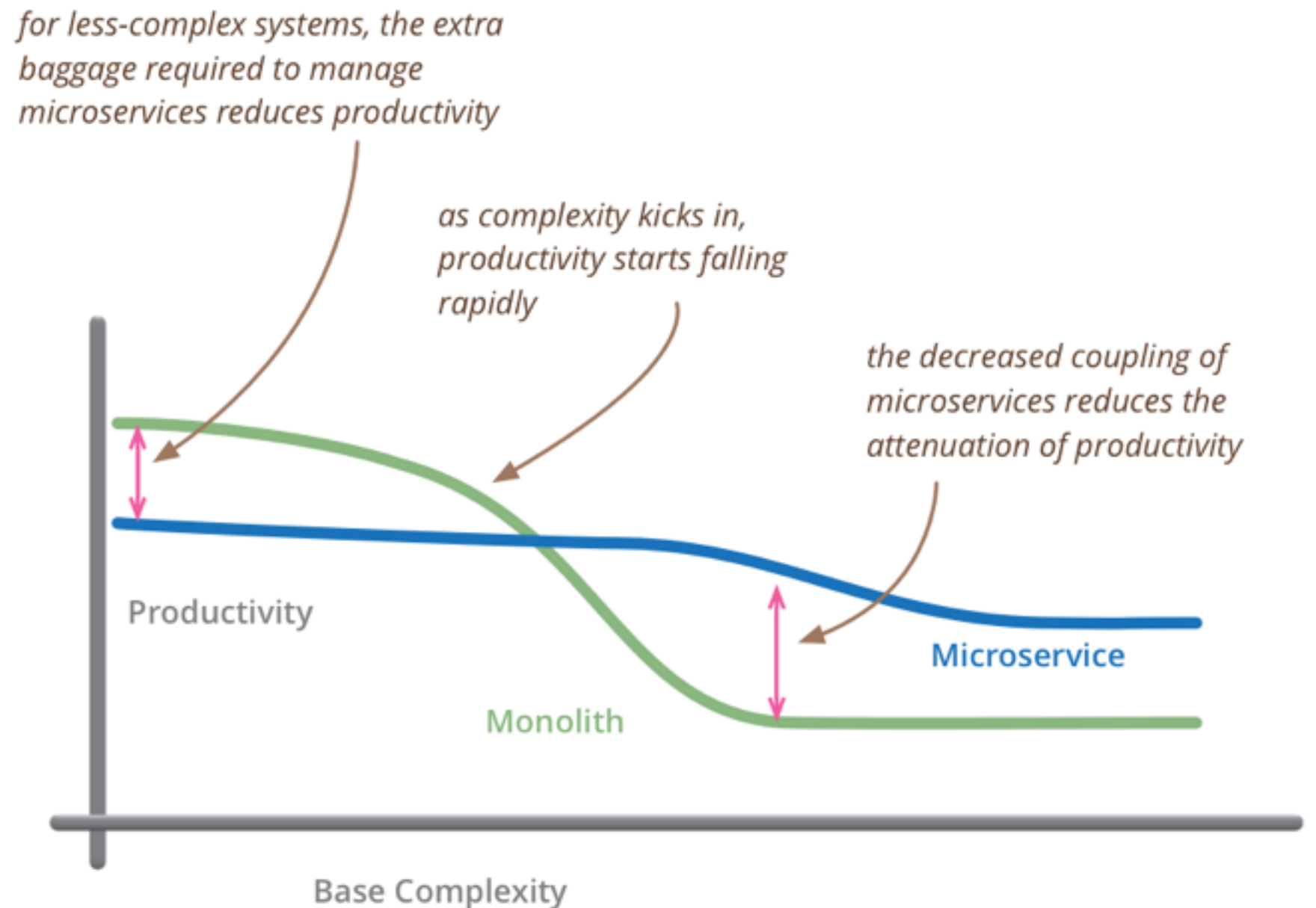
特征 - 兼容设计

Be **conservative** in what you send, be **liberal** in what you accept.

– Postel's law

实施 - 前提

- 1 ~ 2k 行, 从爬行到奔跑
- 2k ~ 20k 行, 开车
- 20k+, 开飞机



but remember the skill of the team will outweigh any monolith/microservice choice

实施 - 维度

京东主站

京东金融

京东微联

京东到家

京东众包

京东钱包

开普勒

众客服

C 端

WEB 端

M 端

SDK 端

PC 端

微信手Q端

定制服务

接口服务

插件服务

外部账号

外部证书

通用服务

账号服务

调度服务

风控服务

登录服务

鉴权服务

通知服务

聊天服务

聊天记录

广播服务

搜索服务

升级服务

表情服务

反馈服务

文件服务

客服服务

会话服务

入口服务

分配服务

评价服务

转接服务

组织架构

留言服务

寻址服务

运维服务

配置管理

调度管理

数据管理

运行管理

注册中心

缓存管理

运营服务

通知管理

表情管理

升级管理

风控管理

帮助中心

咚咚管家

B 端

商家 PC 端

商家移动端

基础服务

状态服务

接入服务

路由服务

定位服务

消息服务

数据服务

数据收集

数据处理

数据开放

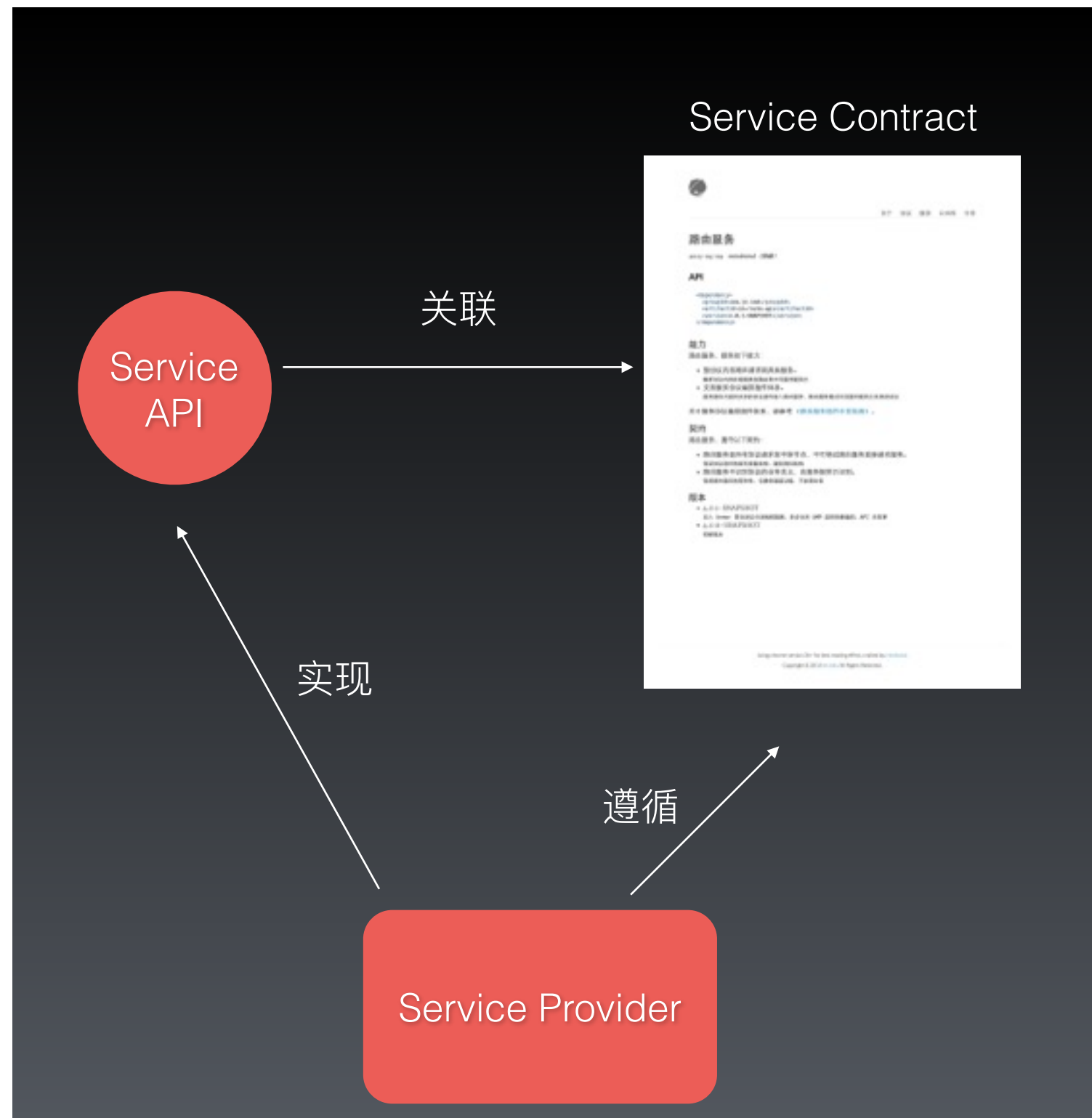
建模

- 业务能力
- 原子、正交化
- 抽象通用

实施 - 协作

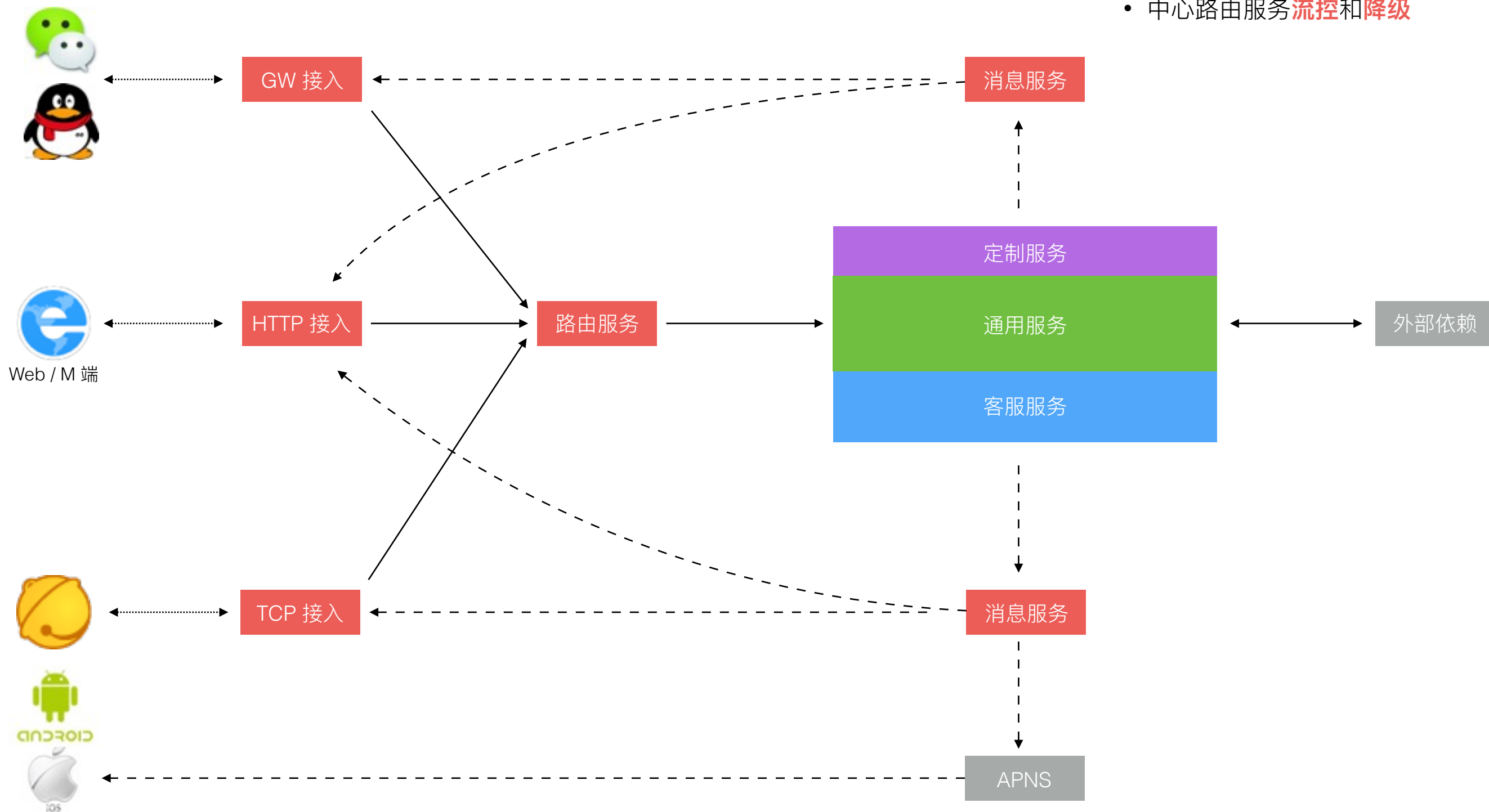
- 服务契约

- API
- 能力
- 契约
- 版本



实施 - 协作

- 服务交互异步化: **防雪崩效应**
- 中心路由服务**流控**和**降级**



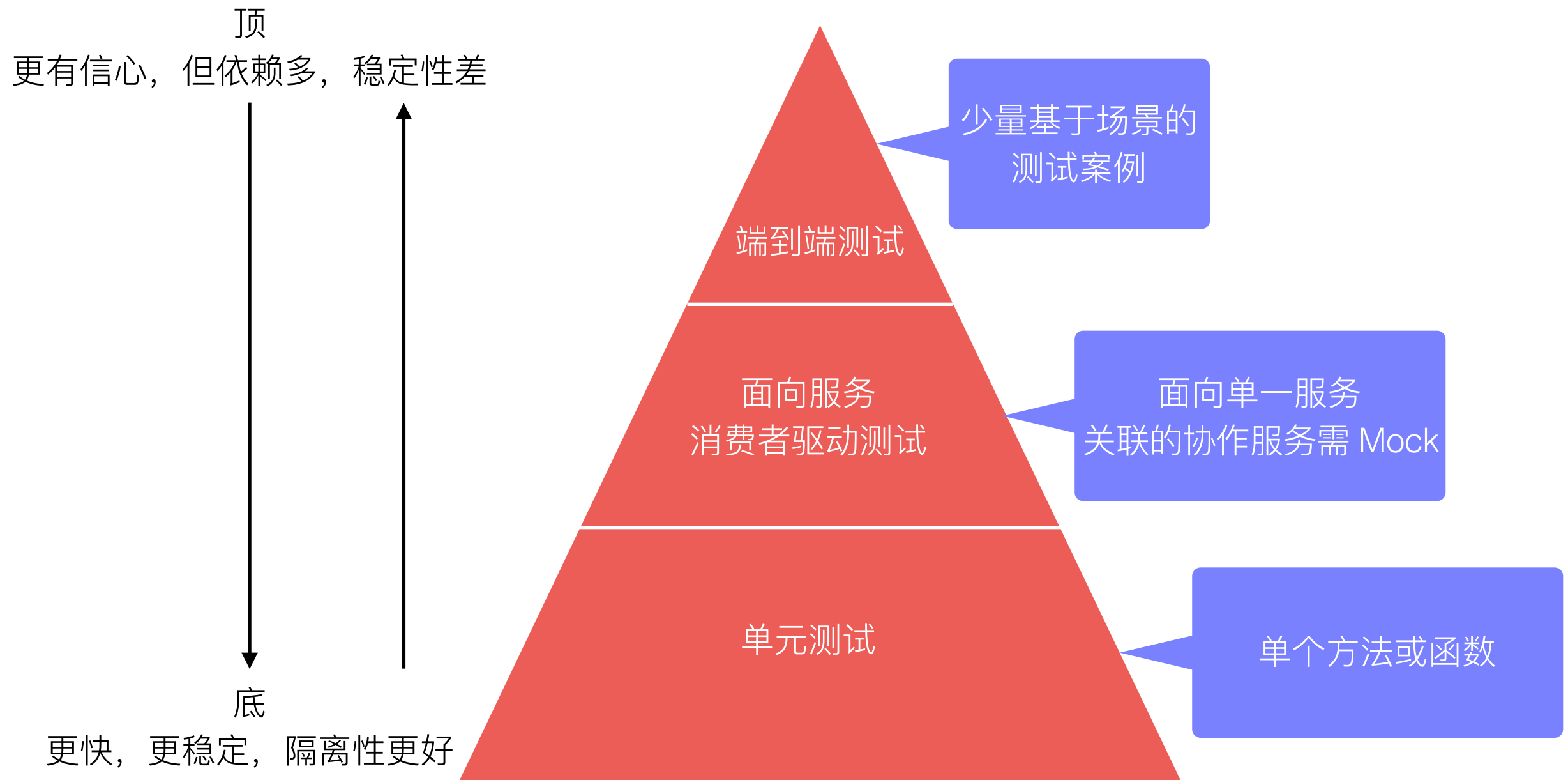
实施 - 测试

四象限



实施 - 测试

金字塔

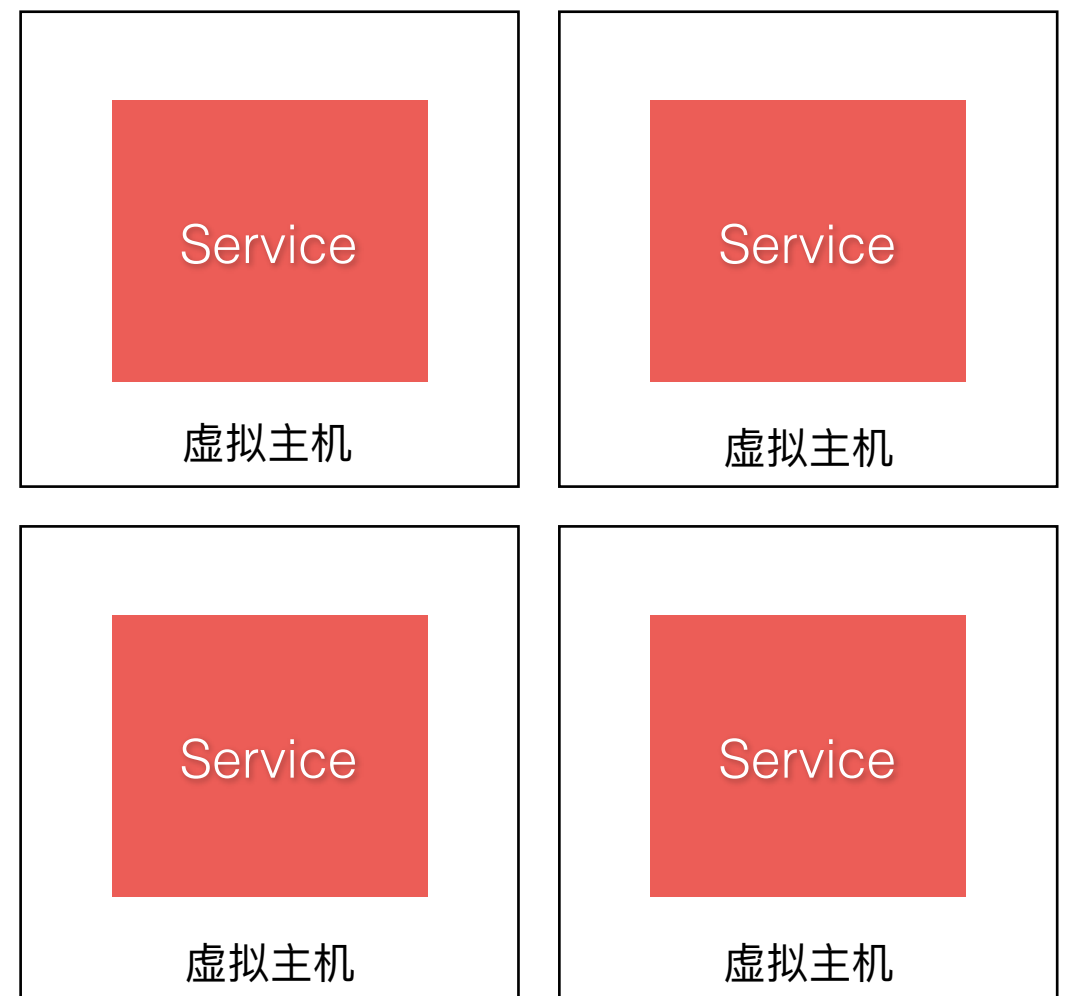


实施 - 部署

一主机多服务



一主机一服务



实施 - 监控

用户视角

业务监控

长周期趋势
多维度

服务监控

响应时间
流量 TPS
全链路
染色

基础监控

IDC
网络
主机
OS
VM

系统视角

实施 - 原则

战略目标

业务扩张

- 加机器不加入

业务开拓

- 快速
- 边际成本低
- 自服务

业务创新

- 试错成本

架构原则

契约化开发

- 契约变更通知
- 考虑消费方

自动化文化

- 代码模板生成
- 编译、测试、部署
- 日志收集
- 告警处理

反脆弱性

- 错误隔离
- 超时管理
- 断路器
- 隔离仓
- 独立部署
- 高度可监控

设计与交付实践

标准化接口

- Dubbo RPC (内部)
- JSF RPC (外部)
- REST (跨语言)

标准化输出

- 日志错误
- 报警提示

标准化配置

- 配置文件
- 启动脚本
- 环境参数

标准化监控

- AOP 接入
- 埋点约定

角色

- 球员 - 开发、测试工程师
- 教练 - 技术经理或架构师
- 经理 - 部门经理

A red speech bubble with a tail pointing towards the '教练' (Coach) role in the list.

类比像 NBA 职业球队

Q & A

谢 谢



微信公众号
瞬间之间