

# 技术为本，人性考量

-----一个架构师程序员眼中的软实力

演讲者：周恩

(版本：1.0.1.210729，部分删减，请勿传播)



## 周恩

华为技术专家兼软件教练，原部门首席程序员  
原京东交易平台架构师兼“京东GEEK发现代码之美”评审导师

设计实现过某东“我的关注”（日最大访问量百亿次级、服务化架构）、“Plus会员”等多个核心系统，是多个软件技术发明专利和著作权（含开发工具和商业软件）的唯一作者，修改过5+个开源软件。



(微信: isharedata)

## 纲要:

- 架构设计: 安生立命之根本
- 测试技术: 开发测试都重要
- 产品管理: 聚焦价值和演化
- 风险控制: 人、业务、技术
- 技术管理: 基于人性求效率
- 个人成长: 平庸到卓越蜕变

## 案例:

- “XX关注” 服务器数量降低XX倍反而性能更高更稳定
- 双十一促销时XX券系统流量暴增XX倍要怎么才扛得住
- 招聘部门XX%员工约XX个月内仅1人离职, 培养出多个专家和经理

软件，就是选用合适的技术，以可接受的成本和质量，实现准确的业务，并预见到未来可能的变化。

关键词：业务 技术 成本 质量 变化

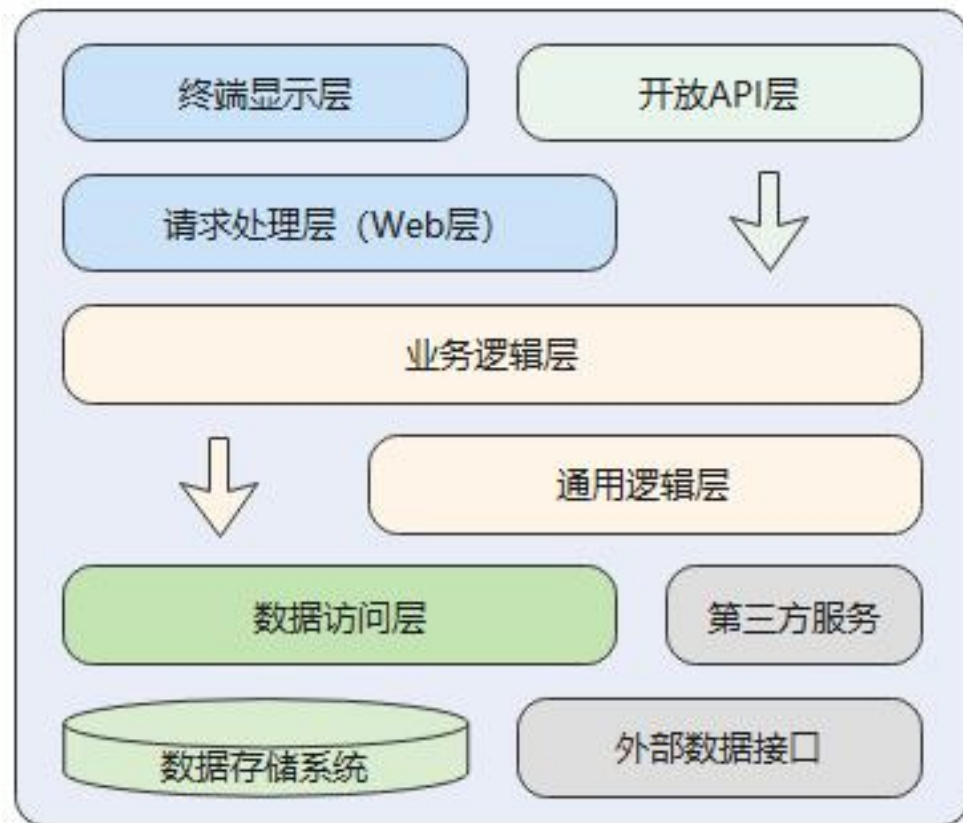
- 结构：代码的层次划分和模块组织
- 性能：数据量级下的性能和可靠性
- 粘合：模块、分层代码的有机关联
- 扩展：对业务、技术变化的适应性

架构设计主要解决容量、性能、可靠性问题，以适应业务和技术的演进

- 技术选型：授权协议、技术评估
- 结构分层：结构清晰、易于理解
- 基础代码：可靠重用、兼容变化
- 数据库表：设计重点、考虑容量

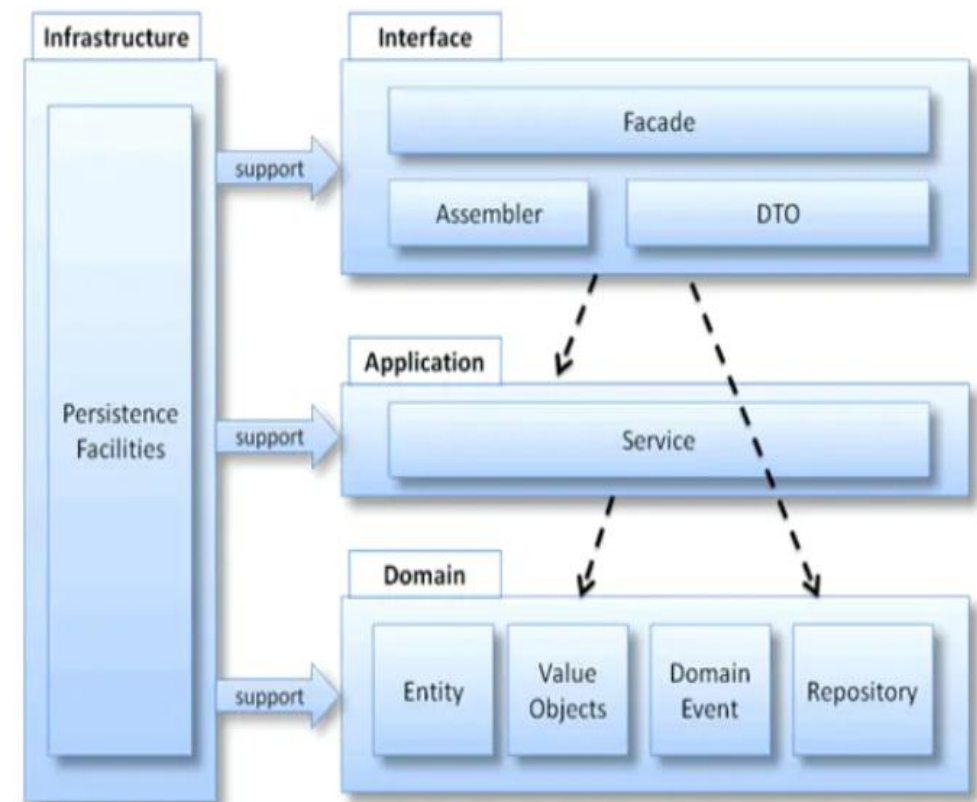
架构可以使用框架，但不等于框架，要避免把框架当做架构，形成强依赖

# 架构设计：什么是好的架构分层



经典分层结构

VS



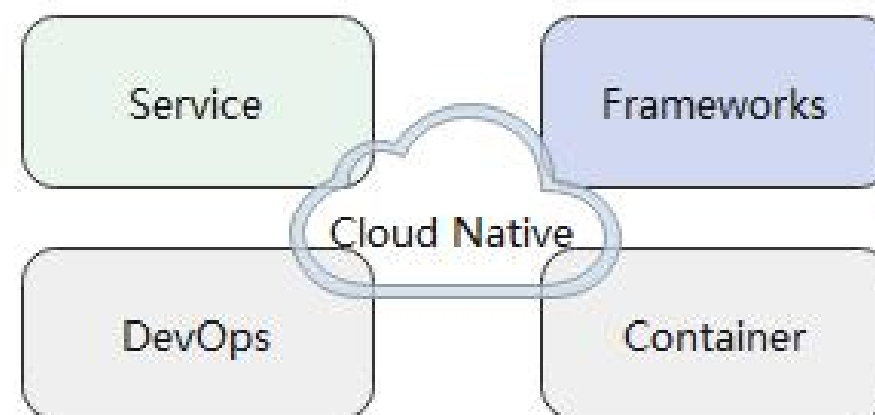
DDD 4层架构

- 结构清晰、概念简单，就低不就高
- 职责单一、避免杂糅，减少暴露面

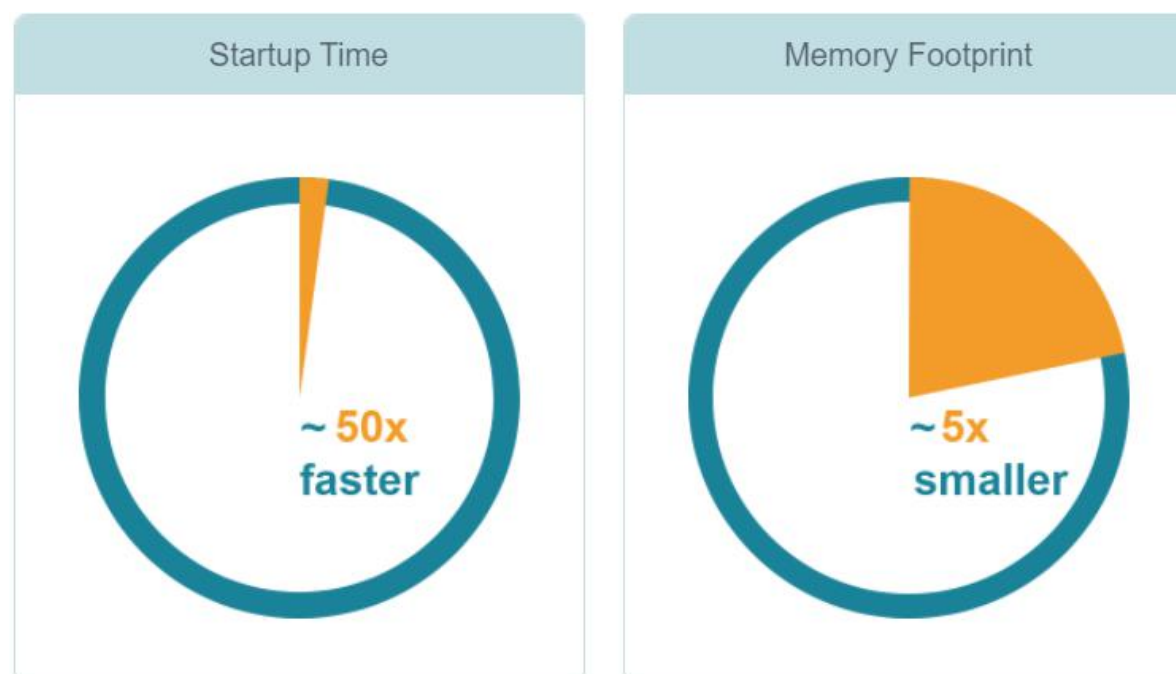
- 广义服务：微服务优先，不局限于微服务
- 服务拆分：先业务拆分，再进行技术拆分

服务化的本质是重用，RPC相对REST而言，可能有更好的性能和安全性





云原生优先，兼容物理机和虚机，不完全依赖XXXX



Microservices frameworks integrated with GraalVM



- 适度求新：技术有成熟期，优先使用标准技术
- 敏捷设计：需要适度设计，整体设计不可缺失
- 原型测试：新技术很必要，有效测试要有量级

■ 问题：用户关注商品、店铺等，数据读写（尤其读）性能不足，大促时容易挂掉

■ 解决方案：

- ✓ 简化技术：去掉HBase、Solr/Lucene、LevelDB等，仅用MySQL+Redis+消息
- ✓ 优化设计：对称数据库表设计，异步消息，分库分表扩容
- ✓ 数据服务：独立数据访问服务，减少数据库连接资源争抢
- ✓ 其他优化：缓存扩容、缓存部署模式和位置、均衡算法等

- 问题：如何低成本适应产品业务演化，兼容不同数据规模
- 解决方案：
  - ✓ 简化技术：标准技术、Apache Commons DbUtils、Spring/Spring Boot、.....
  - ✓ 兼容设计：面向接口设计，兼容非依赖Spring/Spring Boot，不用特有IOC
  - ✓ 数据容量：服务化+数据库分表，兼容不同数据量级，但不过度贪大

架构设计，要预先考虑兼容不同数据量级，而不是“到时候”再考虑高成本扩容

- 开发和测试是孪生关系，需要重视并减少对立
- 单元测试帮助传递技能，减少代码误用的风险
- 自动化测试要注意范围，对不稳定界面要慎重
- 压力测试展现技术实力，本质上可以提升信心

```
=====
测试结果:
测试名称:      [      ]
起止时间:      [      10:33:05] 至 [      10:34:55]
测试时间:      [120000000] 次测试共花费 [110476] 毫秒
并发循环:      [100] 个并发线程下循环 [10000] 次
每秒钟吞吐量: [108000] 次/每秒
每分钟吞吐量: [6480000] 次/每分钟
每小时吞吐量: [388800000] 次/每小时
=====
```

```
=====
测试结果:
测试名称:      [      ]
起止时间:      [      10:38:10] 至 [      10:57:22]
测试时间:      [120000000] 次测试共花费 [1151914] 毫秒
并发循环:      [100] 个并发线程下循环 [100000] 次
每秒钟吞吐量: [104000] 次/每秒
每分钟吞吐量: [6240000] 次/每分钟
每小时吞吐量: [374400000] 次/每小时
=====
```

```
=====
测试结果:
测试名称:      [      ]
起止时间:      [      09:42:55] 至 [      09:57:25]
测试时间:      [120000000] 次测试共花费 [869283] 毫秒
并发循环:      [100] 个并发线程下循环 [100000] 次
每秒钟吞吐量: [138000] 次/每秒
每分钟吞吐量: [8280000] 次/每分钟
每小时吞吐量: [496800000] 次/每小时
=====
```

```
=====
测试结果:
测试名称:      [      ]
起止时间:      [      11:59:00] 至 [      12:14:23]
测试时间:      [120000000] 次测试共花费 [922953] 毫秒
并发循环:      [100] 个并发线程下循环 [100000] 次
每秒钟吞吐量: [130000] 次/每秒
每分钟吞吐量: [7800000] 次/每分钟
每小时吞吐量: [468000000] 次/每小时
=====
```

测试机器：2011年产 X220 二手笔记本，i5 2代 2520M CPU，DDR3 16GB内存，SSD硬盘

低性能机器迫使你写高性能代码，高性能机器则是生产力工具

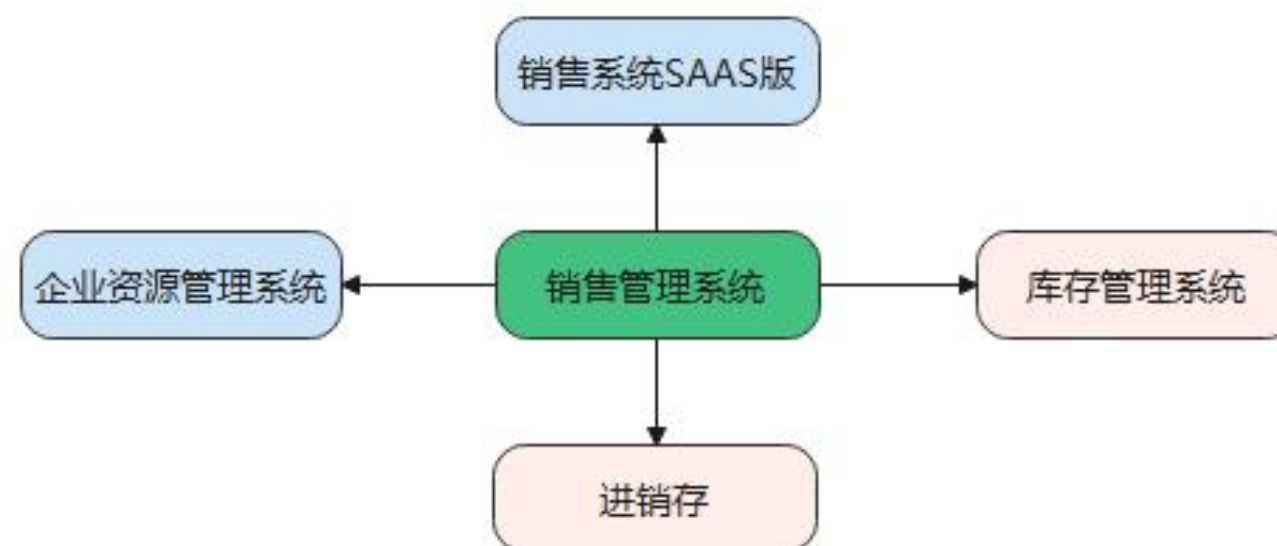
- 聚焦价值：提供客户价值，没有价值不能长久
- 版本演进：明确核心和外延价值，版本化演进
- 双轮驱动：业务体现价值，技术支撑价值实现
- 易用好用：简洁美，避免把功能和技术复杂化

产品化就是以技术实现价值版本化演进，一锤子买卖则是项目

- 业务规划：以核心价值为关联、外延，或聚焦核心做强做精
- 技术支持：架构设计，兼容支持不同规模、性能和可靠性等
- 平衡节奏：从商业保护、市场试探、实现成本等，取舍平衡

XX “服役一代、研制一代、预研一代” 思路值得借鉴





摘自：《销售管理系统版本规划》-周恩（2005）

## ■ 版本演化：

- ✓ 业务演化：业务复用
- ✓ 版本演化：版本节奏

版本演化的根本目的是商业复用，花一样钱补五样

- 人的风险：人是核心，管控和人性结合
- 业务风险：价值演进、成本收益要平衡
- 技术风险：技术，可能促进或破坏效率
- 管理风险：管理确保有序，领导不是领导岗位

人、业务、技术是软件生产核心，技术和管理可能有效，也可能是负向的

- 国际政治：全球背景下的阵营冲突
- 国家政策：符合国家大政方针导向
- 产权法律：知识产权以及商业合规
- 标准方向：标准不同，成本也不同

在食物链顶端，管理层的竞争格局尤为重要

- 问题：XX券领取从分散合并到领券中心后，大促时流量暴增，就容易雪崩
- 解决方案：
  - ✓ 技术应对：正常流量增加X倍，人性因素增加Y倍，按 $X*Y$ 倍计算
  - ✓ 其他处理：...

在大规模系统中，活着是第一优先考虑的！要假设一切都是不可靠的

- 问题：虚拟资产计算错误时，展示为0容易被投诉
- 解决方案：
  - ✓ 虚拟资产计算错误时，不展示（空白），会比0少引起投诉

0容易让人敏感和不满，用户基数大就会放大问题

- 团队组织：高低搭配，重视遴选，尊重人性
- 人才培养：**基础扎实、愿意上进、懂得感恩**
- 有效协作：团队协作、避免无效需求和返工
- 软件工程：过程要适配裁剪，敏捷也要设计

与优秀的人合作会让自己更优秀！**合作者厉害，意味着自己更厉害**

避免KPI伪需求、避免没想清楚就匆忙编码、避免高强度拼体力造成团队崩溃

# 案例：稳定上进的团队管理要考虑人性

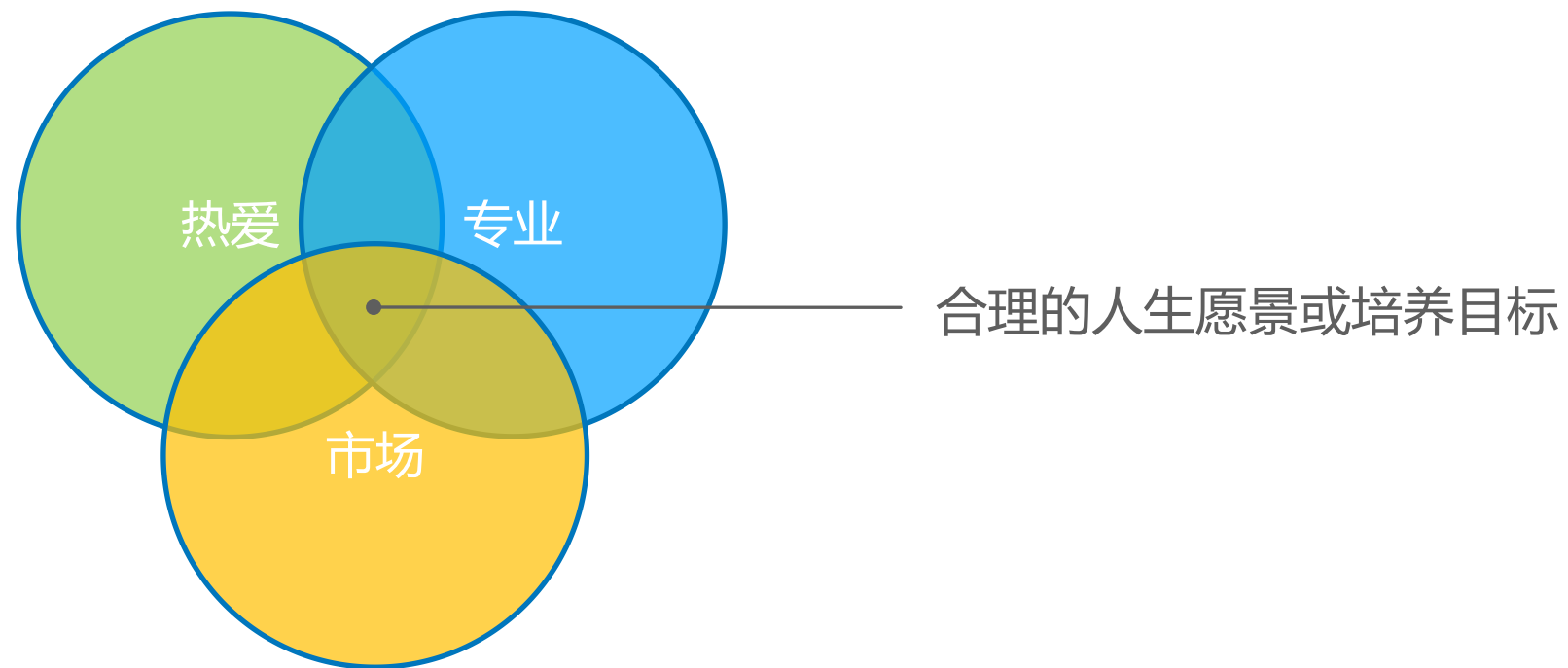
招聘部门XX%员工约XX个月内仅1人离职，培养出多个专家和经理

如何做到：

- ✓ **价值观：愉快、成长、挑战、成就感、最好钞票也合适**
- ✓ **领导力：领导不是领导岗位**，架构师也可以发挥领导力
- ✓ **内省力：以己及人**，人性考量的同理心，方可用人以心

**管理的初阶是管控，中阶是管控结合人性，最高境界是用人以心**

人性化管理需要组织支持，也对管理有更高要求，要专业和责任，搞得不好就会是放羊  
要追求质量和效率，看得长远，就不要拼时间搞内卷，除非真的只是搬砖



市场、热爱和专业，基本确定了一个人的高度  
热爱和坚持，可以把平庸变为专业；专业和市场，可以把无感变为热爱



- 方向正确：从价值、门槛、市场综合评估业务和技术
- 持续努力：方向选择很难，更难的是寂寞的持续努力
- 进化思维：要从二元对立心智模式，进化到整合思维

平衡选择：情商还是智商？全栈还是精专？年龄？ ...

# 感谢您的聆听，期待与您再相见！

## 内容回顾：

- 架构设计：安生立命之根本
- 测试技术：开发测试都重要
- 产品管理：聚焦价值和演化
- 风险控制：人、业务、技术
- 技术管理：基于人性求效率
- 个人成长：平庸到卓越蜕变

# 结交有格局、专业、友好的朋友！

个人理念：

- 架构师不仅需要技术，业务技术管理结合才是王道，包括人性的理解
- 用人以心是管理最高境界：愉快、成长、挑战、成就感、最好钞票也合适
- 有市场，有热爱，做到专业，就是成功！



(微信：isharedata)



麦思博(msup)有限公司是一家面向技术型企业的培训咨询机构，携手2000余位中外客座导师，服务于技术团队的能力提升、软件工程效能和产品创新迭代，超过3000余家企业续约学习，是科技领域占有率第1的客座导师品牌，msup以整合全球领先经验实践为己任，为中国产业快速发展提供智库。



高可用架构主要关注互联网架构及高可用、可扩展及高性能领域的知识传播。订阅用户覆盖主流互联网及软件领域系统架构技术从业人员。高可用架构系列社群是一个社区组织，其精神是“分享+交流”，提倡社区的人人参与，同时从社区获得高质量的内容。