

京东商城架构设计分享



明确设计目标

高扩展性

系统架构简单清晰，应用系统间耦合低，容易水平扩展，业务功能增改方便快捷

高可用性

自动化运维。整体系统可用性99.99%，单个系统可用性99.999%。全年故障时间整个系统不超过50分钟，单个系统故障不超过5分钟

02

低成本

增加服务的重用性，提高开发效率降低人力成本；利用成熟开源技术，降低软硬件成本；利用虚拟化技术，减少服务器成本

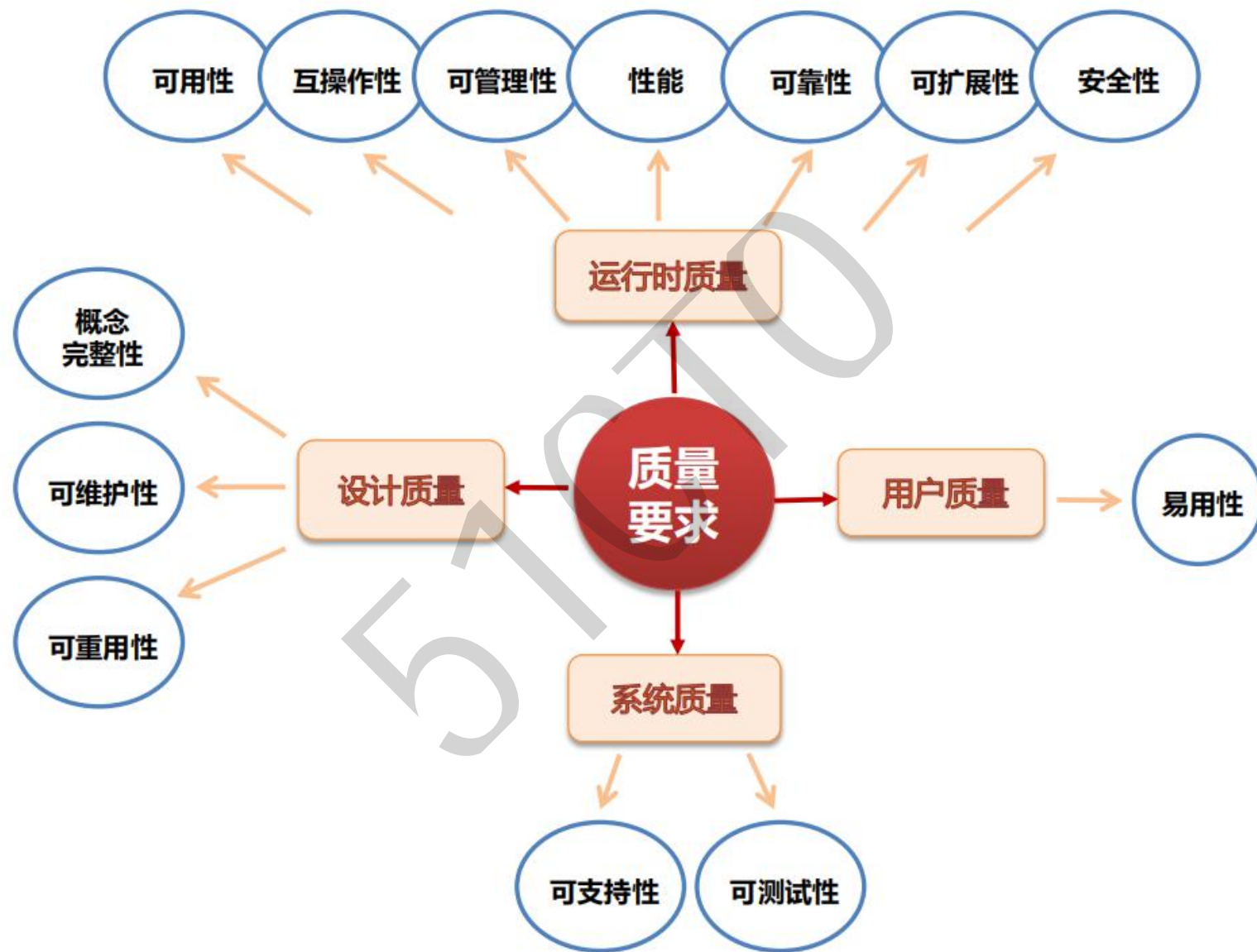
03

多快好省

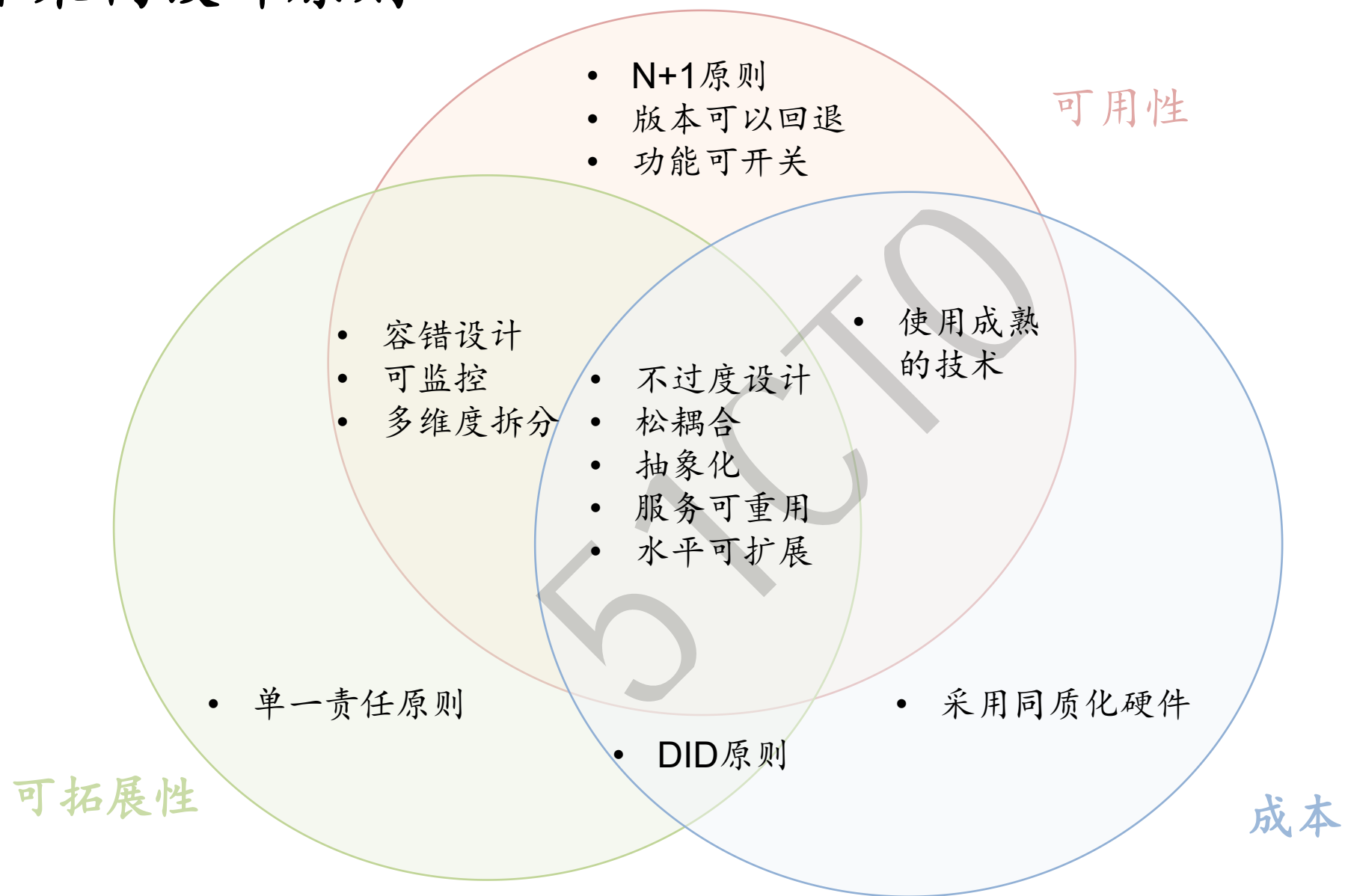
构建超大型电商交易平台，兼顾效率和性能，达到高人效、高时效和低成本的目标

04

架构设计的13个评测维度



总体架构设计原则



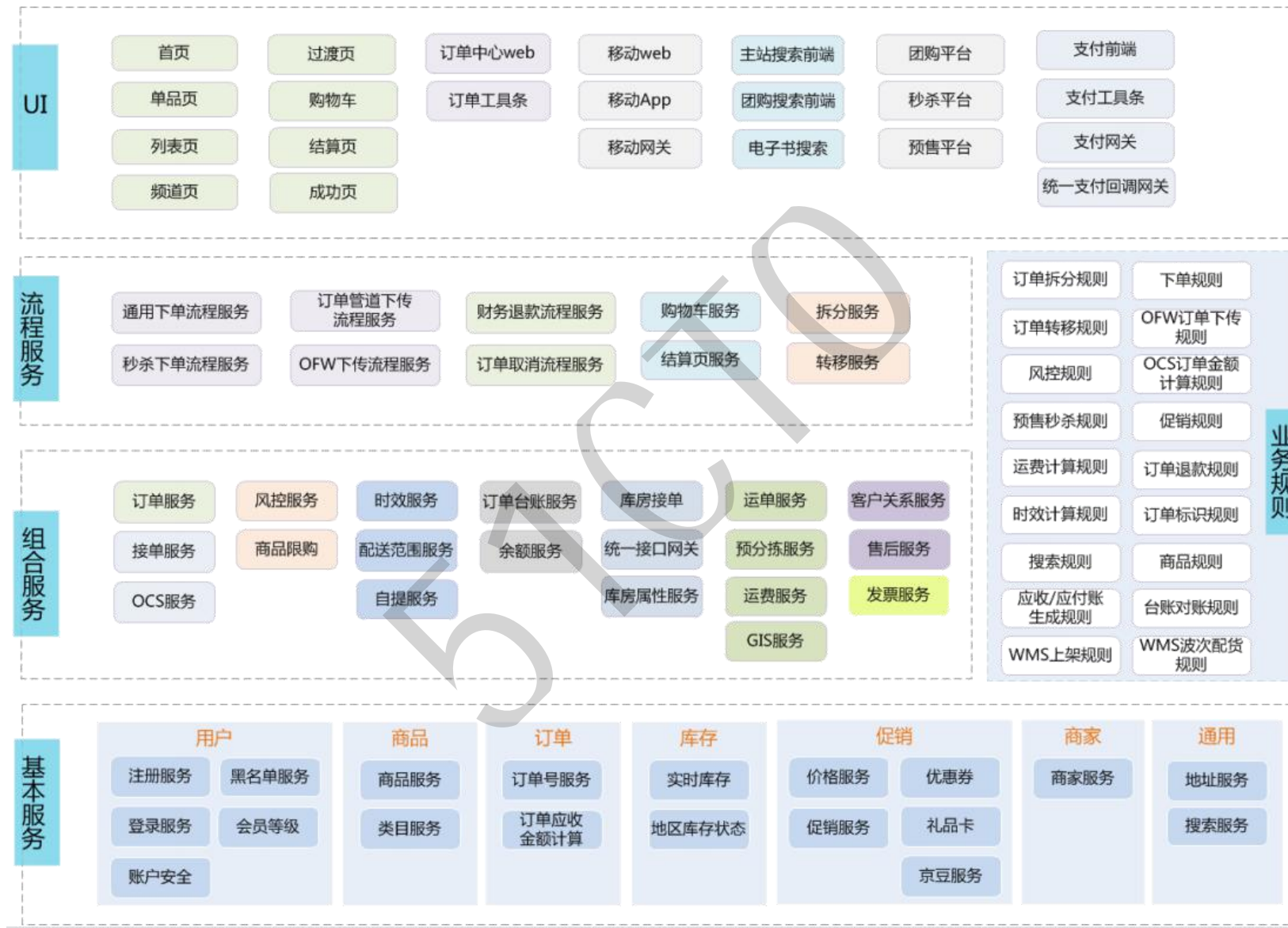
四种架构观测维度

齐老师，这个部分的内容感觉有点像企业架构的内容啦，对于一个技术方向的架构师的话是否也需要掌握这么多内容呀

如果是这样的话，感觉后面普适性的原则也有点多，因为很多原则性的东西在网络上可能也可以看到，那这样的话我们就需要把落脚点放在京东的架构设计上，可以讲一下京东的架构设计是如何满足这些内容的，有什么精妙之处，哪些我们可以参考。



业务架构总览



业务架构设计原则

1. 业务平台化

- 业务平台化，相互独立。如交易平台、仓储平台、物流平台、支付平台、广告平台等
- 基础业务下沉，可复用。如用户、商品、类目、促销、时效等

2. 核心业务、非核心业务分离

- 电商核心业务与非核心业务分离，核心业务精简（利于稳定），非核心业务多样化。如，主交易服务、通用交易服务



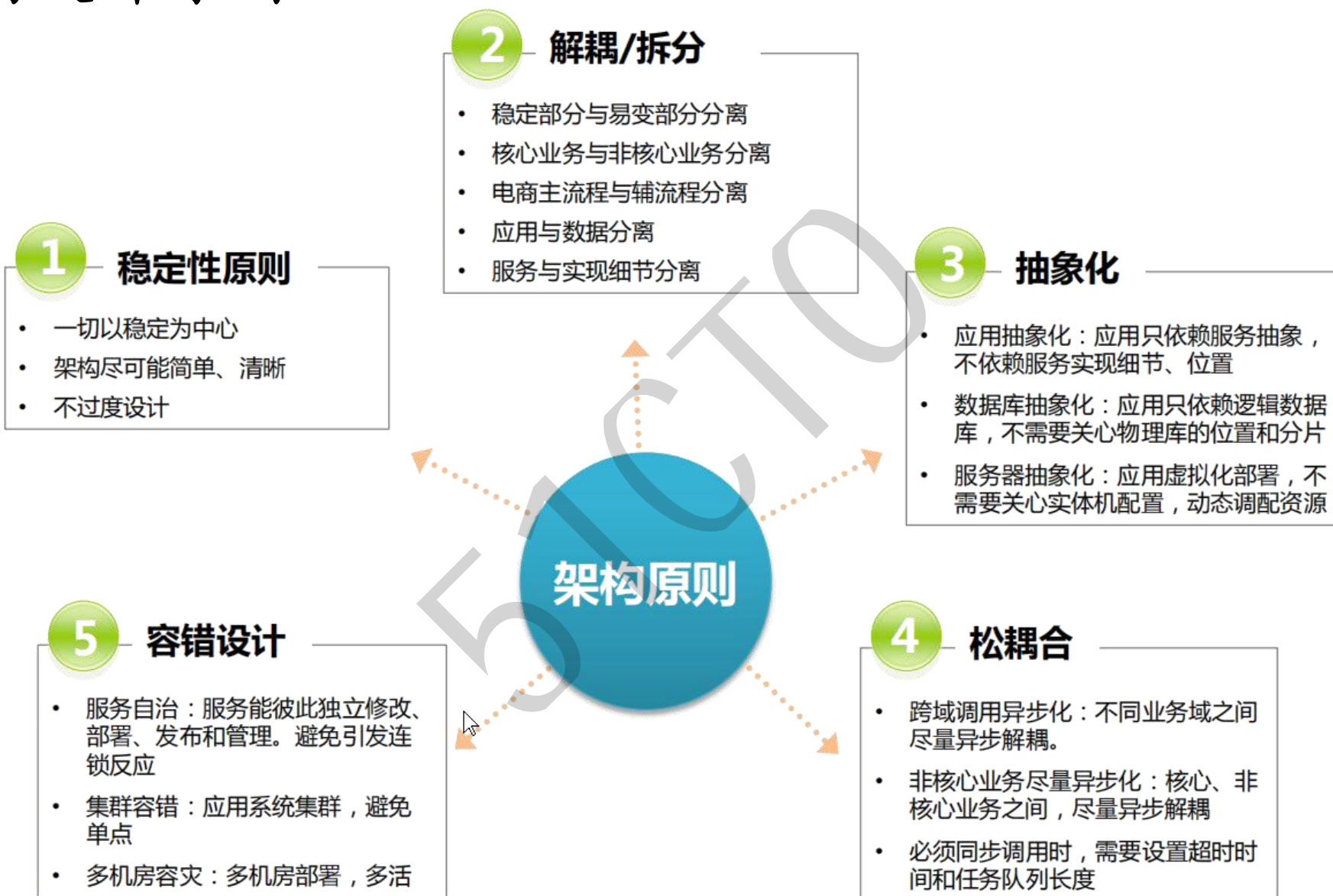
4. 区分主流程、辅流程

- 分清哪些是电商的主流程。运行时，优先保证主流程的顺利完成，辅流程可以采用后台异步的方式。避免辅流程的失败导致主流程的回滚。如，下单时，同步调用快照，异步通知台账、发票

3. 隔离不同类型的业务

- 交易业务是签订买家和卖家之间的交易合同，需要优先保证高可用性，让用户能快速下单
- 履约业务对可用性没有太高要求，可以优先保证一致性
- 闪购业务对高并发要求很高，应该跟普通业务隔离

应用架构设计原则



应用架构的分层设计



架构在不同维度下的要求



分层间的依赖原则

2. 跨域弱依赖

- 跨业务域调用时，尽可能异步弱依赖

1. 依赖稳定部分

- 稳定部分不依赖易变部分
- 易变部分可以依赖稳定部分
- 要求：避免循环依赖

6. 核心服务依赖

- 核心服务不依赖非核心服务
- 非核心服务可依赖核心服务
- 条件：核心服务稳定

5. 平台服务依赖

- 平台服务不依赖上层应用
- 上层应用可依赖平台服务
- 条件：平台服务稳定

3. 基本服务依赖

- 基本服务不能向上依赖流程服务
- 组合服务、流程服务可以向下依赖基本服务
- 条件：基本服务稳定

4. 非功能性服务依赖

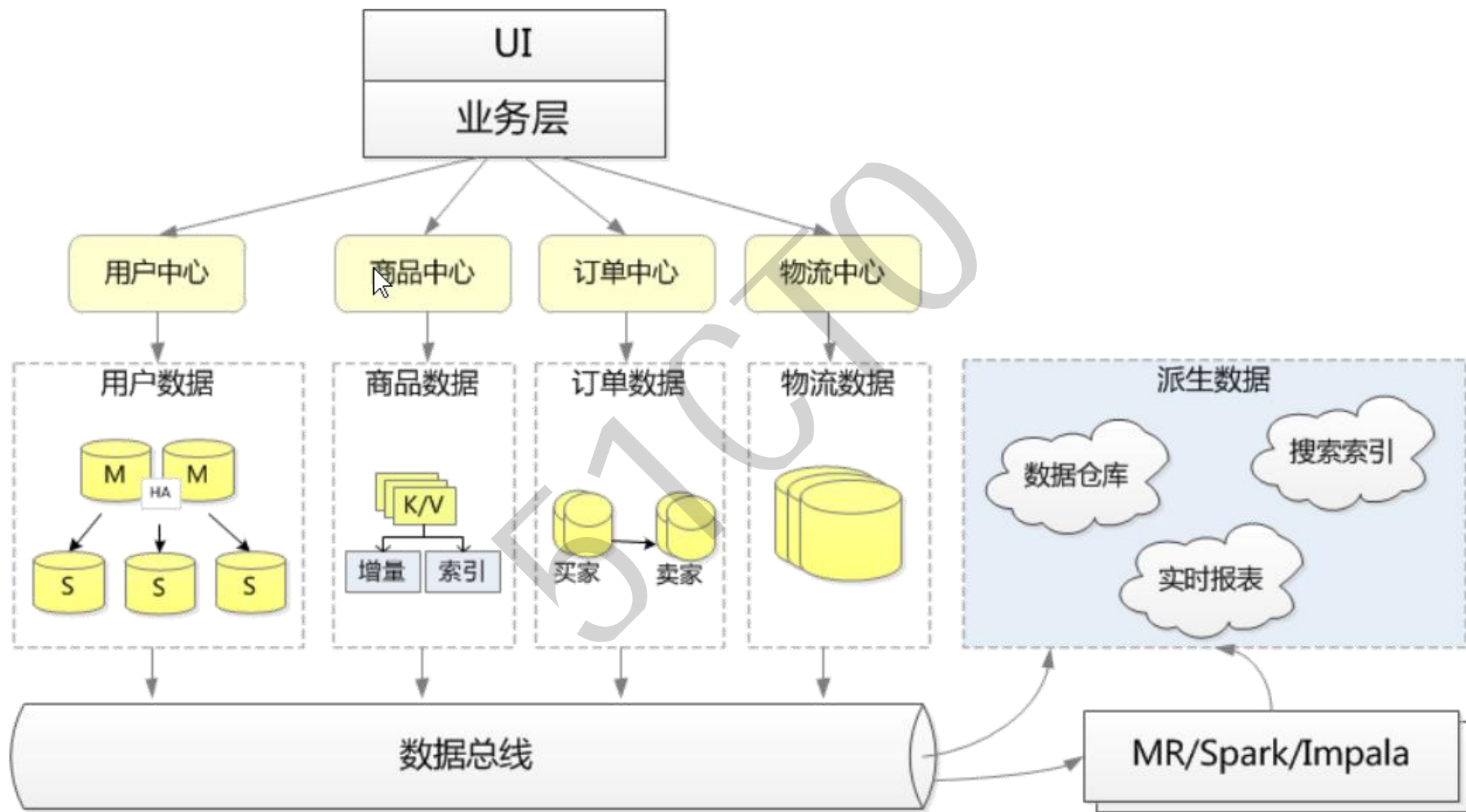
- 非功能性服务不依赖功能性服务
- 功能性服务可依赖非功能性服务
- 条件：非功能性服务稳定



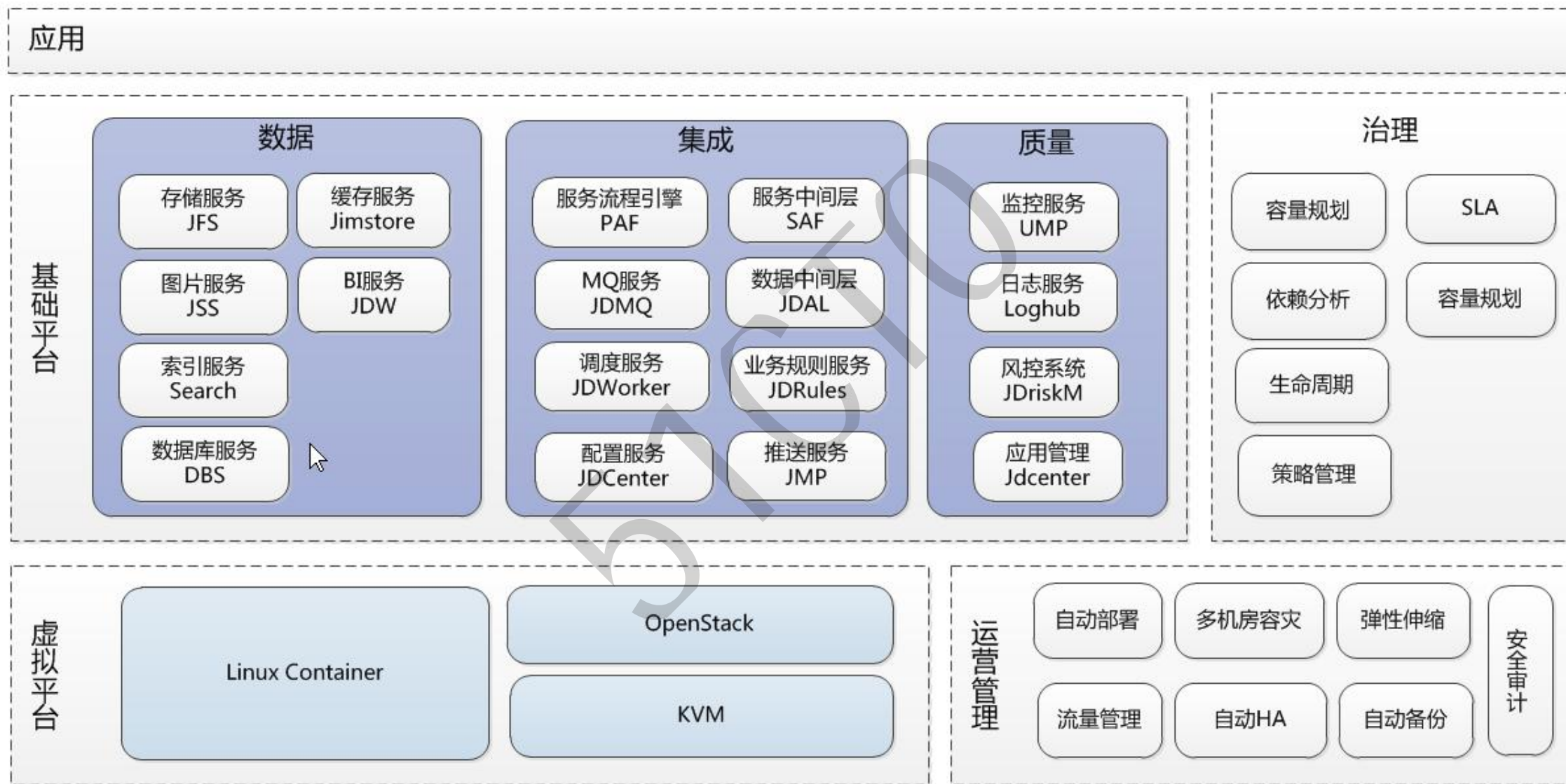
数据架构设计原则



数据架构示意图



技术架构图



技术架构运行时原则

1、可监控

- 服务的TPS和RT是否符合SLA
- 是否出现超预期流量

2、应用可回滚，功能可降级

- 应用出现问题时，要求能回滚到上一版本，或做功能开关或降级

3、在线扩容

- 超预期流量时，应用系统可选择在线水平扩展

运行时

6、可故障转移

- 多机房部署，发生故障时，能即时切换

4、安全保证

- 确保系统的保密性和完整性
- 具有足够的防攻击能力

5、可容错

- 核心应用要求多活，避免单点设计，并且自身有容错和修复能力。故障时间TTR小

系统部署原则

2 D-I-D原则

- 设计20倍的容量 (Design)
- 实现 3 倍的容量 (Implement)
- 部署1.5倍的容量 (Deploy)

1 N+1原则

- 确保为故障多搭建一套系统，避免单点问题。例如，多机房部署、应用系统集群、数据库主备等
- 功能开发与运维分开。系统开发完成后，交给专业的运维团队管理和运营。

5 业务子网

- 机房部署以业务域划分：基本服务和数据库，相同业务域的服务器部署在一起；不同业务域的服务器物理隔离

3 支持灰度发布

- 系统新上线，要求支持“灰度”发布，分步切流量，故障回滚

4 虚拟化部署

- 虚机部署：二级系统、三级系统采用虚拟机部署，节省资源和管理成本
- 虚拟化部署：一级系统应用服务器，采用虚拟化部署



谢谢!



51CTO 学堂

为梦想增值!

版权声明，本文档全部内容及版权归北京无忧创想信息技术有限公司所有。