



QCon 全球软件开发大会
INTERNATIONAL SOFTWARE
DEVELOPMENT CONFERENCE

BEIJING 2017

高可用实践：从淘宝到上云的差异

SPEAKER / 王晨纯（沐剑）

About Me

- 王晨纯，花名沐剑
- 2011~2012 淘宝评价系统
- 2012~2015 阿里店铺平台，负责浏览及RPC服务业务支撑、性能优化、双11稳定性
- 2015~2016 聚石塔EWS系统架构师，负责公有云服务架构及整体高可用方案
商家事业部双11作战项目研发PM
- 2017~至今 负责商家营销平台

连续5年双11核心作战小组成员

去IOE、异地多活、全链路压测、安全混合云、容器服务VPC化等项目的设计和实施

- Blog: hesey.net

Agenda

1. 淘宝店铺的稳定性体系

- 浏览型系统的基础链路
- 缓存的设计和部署
- 容灾设计

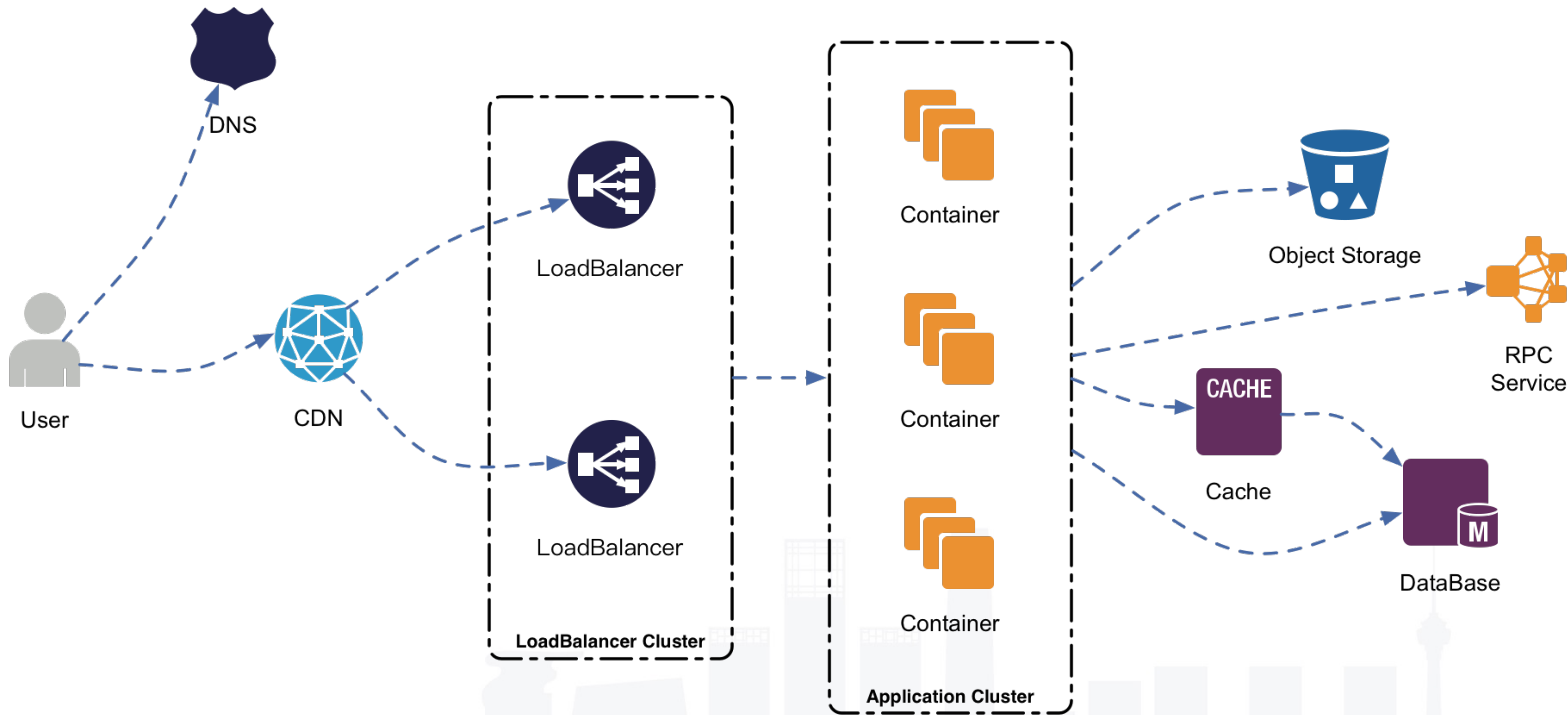
2. 基于公有云的高可用设计

- 云相关的经典故障
- 面向云的高可用架构设计
- 容灾部署架构

淘宝/天猫 店铺

- 一个典型的高并发浏览型系统（读）
- 每秒钟20万次Web请求
 - 对应后端的请求量约每秒400万次
 - 缓存、数据库、分布式远程调用
- 在性能、稳定性、体验之间的平衡





浏览型系统的基础链路

性能和稳定性之间的关系



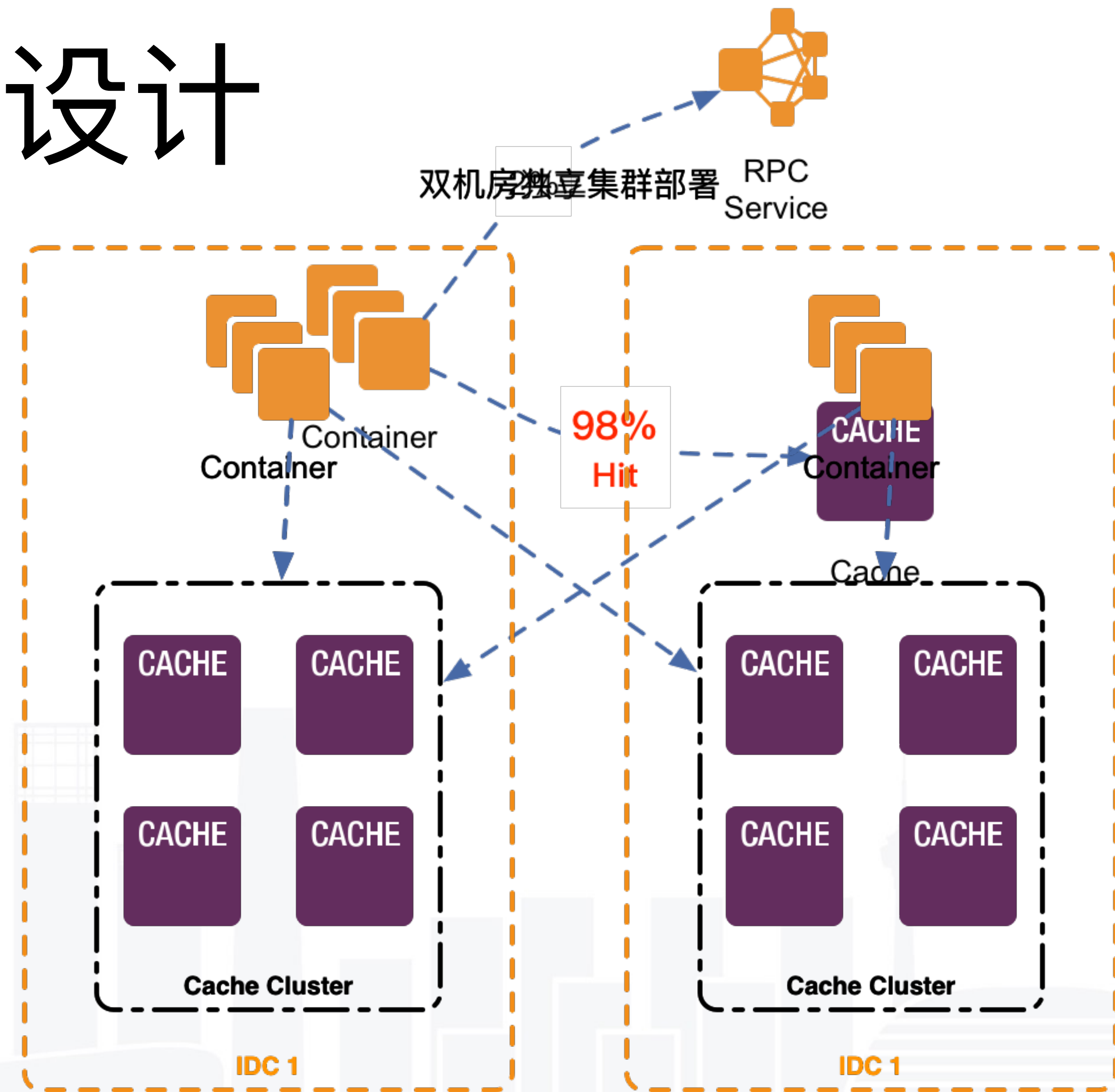
当应用优化已无法满足需求

- JVM层优化
 - perf
 - <https://github.com/jrudolph/perf-map-agent>
 - Exception开销
 - `_ZN19java_lang_Throwable19fill_in_stack_trace`
 - StringTable冲突
 - CodeCache GC问题
 - JVM warmup问题
 - Alibaba JDK
 - Zing ReadyNow
 - J9 AOT

```
0.26% lib64libc-2.12.so      [.] 0x0000000000008ddb2
0.20% binbash               [.] 0x0000000000001e941
0.20% libjvm.so              [.] typeArrayKlass::allocate(int, Thr
0.15% [kernel]               [k] _spin_lock
0.13% libpython2.6.so.1.0    [.] 0x000000000000a5a56
0.12% lib64libpthread-2.12.so [.] 0x000000000000c495
0.11% [kernel]              [k] find_busiest_queue
0.11% [kernel]              [k] copy_page_c
0.10% [kernel]              [k] find_next_bit
0.09% perf-181218.map        [.] 0x00007fdc44f8cd42
0.08% [kernel]              [k] tg_load_down
0.08% [kernel]              [k] schedule
0.07% [kernel]              [k] update_curr
0.06% [kernel]              [k] cpumask_next_and
0.06% gawk                  [.] interpret
0.06% [kernel]              [k] thread_return
0.06% [kernel]              [k] _spin_lock_irqsave
0.06% [kernel]              [k] update_shares
0.05% perf-53410.map         [.] 0x00007fb6950e6dd3
0.05% [kernel]              [k] __audit_syscall_exit
0.05% [kernel]              [k] apic_timer_interrupt
0.05% [kernel]              [k] page_fault
0.05% libjvm.so              [.] void ParScanClosure::do_oop_work<
```


缓存设计

- 缓存前置
 - 富客户端
 - 节约应用服务器
 - case: 某服务调用100亿优化到20亿
- 为容灾设计的部署架构
 - 简化为双机房模型
 - 双机房共享集群
 - 双机房独立集群
 - 失效的实现区别



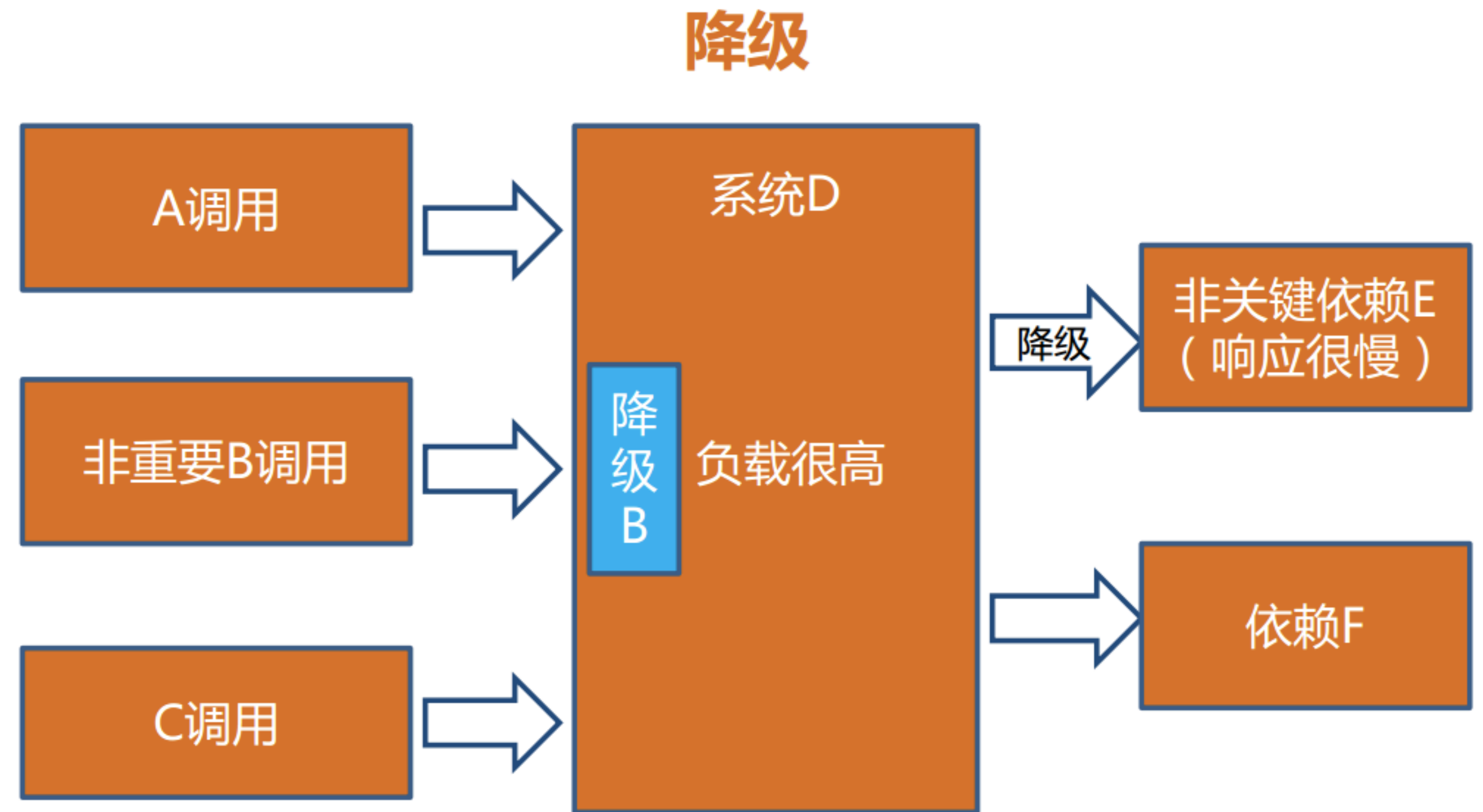
限流 & 降级

当流量超出你的掌控时如何让自己成为幸存者



限流降级

- 限流降级考验的究竟是什么
 - 故障自恢复能力
 - 下游依赖出问题
 - case: 断网演练
- 开关降级
 - 避免回滚导致的慢恢复
 - case: 如何在大变更下减少故障(如底层中间件的升级)



将高可用设计融入方案



容灾设计

- 一切都有可能挂，特别是新引入的依赖
 - 如何降级，如何解决
- 发布计划
- 特别关注的问题
 - 是否有数据迁移
 - 是否有对账，如何保证一致性
 - 发布顺序是否有依赖
 - 是否需要停机
 - 如何回滚
 - 是否需要挂公告通知外部用户



直觉很重要
但手艺要能化为产品/工具

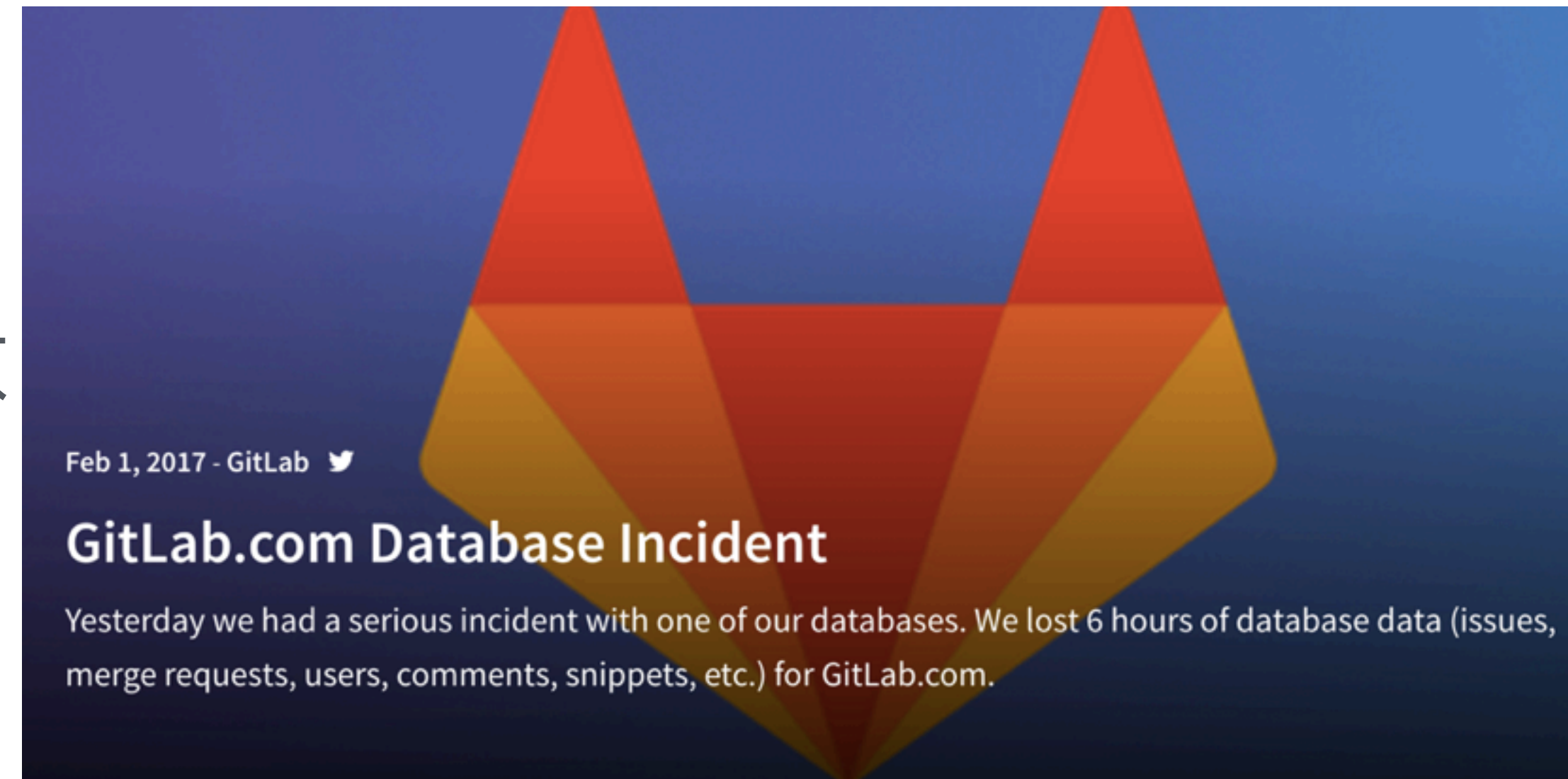


公有云高可用架构



故障案例

- Gitlab
 - 错误删除生产数据库
 - LVM Snapshot每24小时备份一次，最新数据是6小时前的
 - 常规备份由于pg_dump客户端版本问题失效
 - Azure Disk snapshot未启用
 - 数据库同步会导致webhook删除，所以webhook只能从备份中恢复
 - S3 备份未生效，bucket为空
 - 糟糕的备份流程，并且没有明确的文档



容灾没有定期演练 对依赖的云备份原理不够了解



故障案例

- AWS S3
 - 敲错命令误下线核心服务
 - 对象索引服务
 - 位置服务系统
 - 改进措施
 - 静默期
 - 最小资源数

对架构可用性
云服务的威胁竟然
我们该怎么办



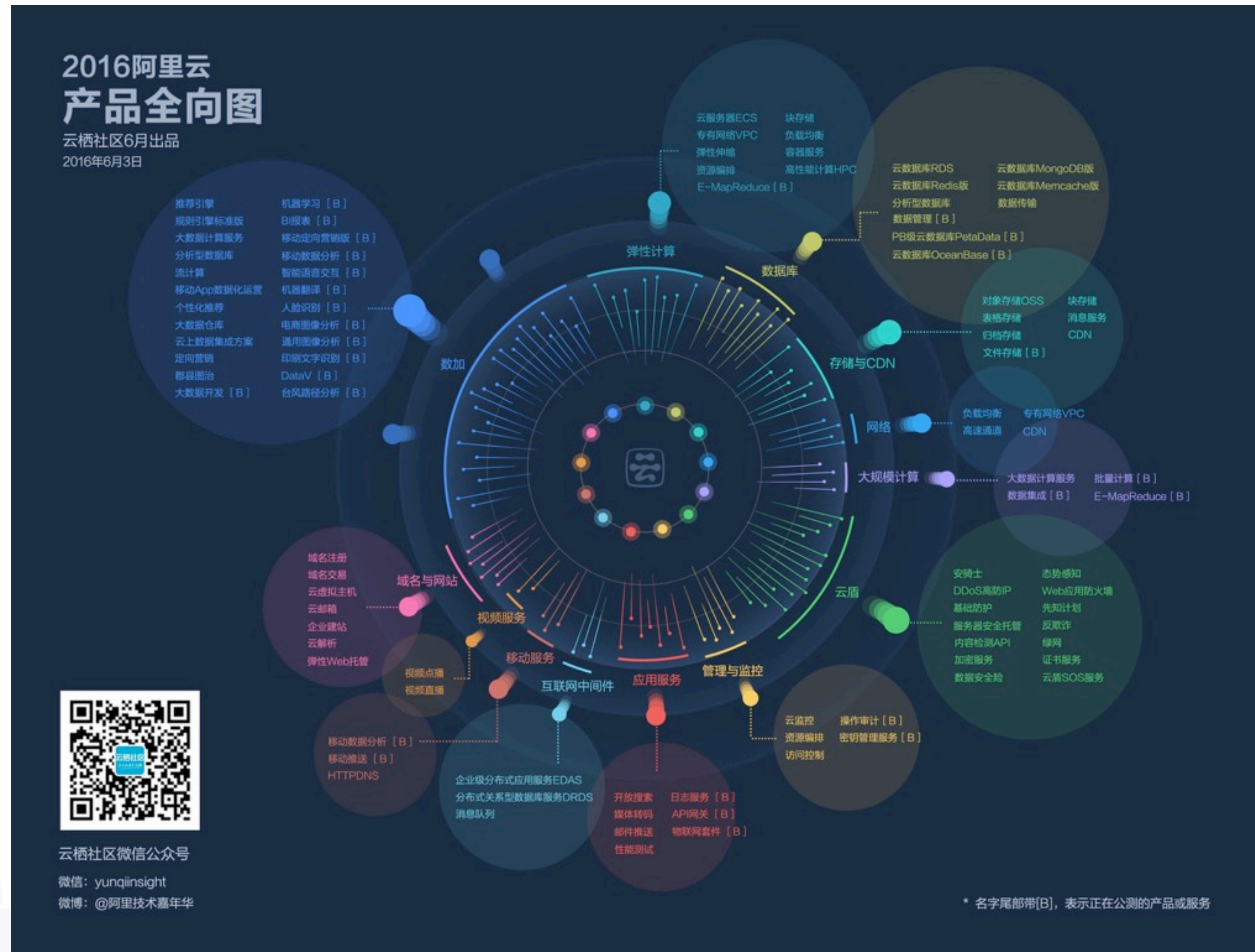
面向云的高可用架构

软件定义基础架构 SDI(Software-Defined-Infrastructure)



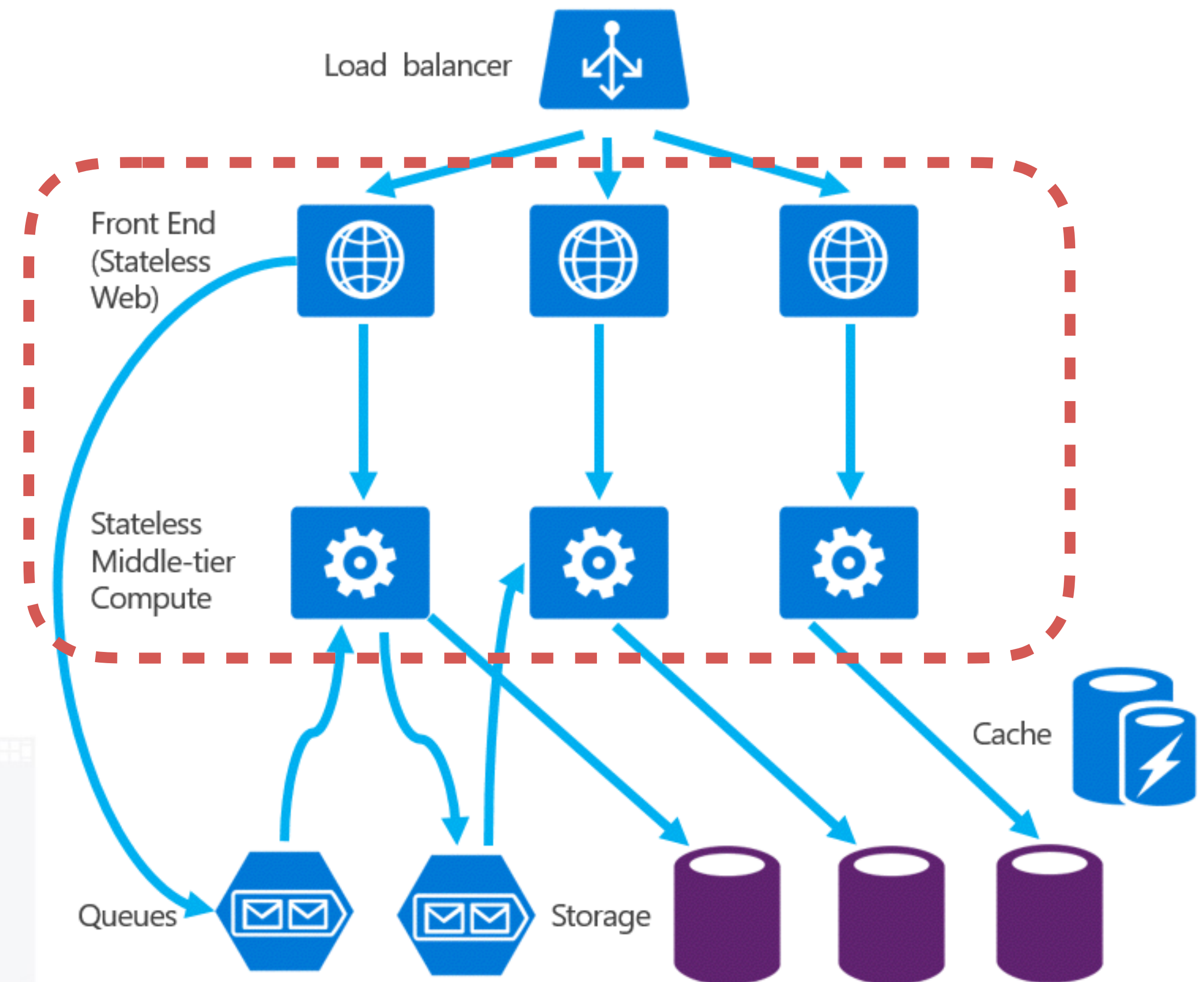
面向弹性的设计

- 资源是容易获得的
 - 能用加机器解决问题其实很厉害
- 丰富多样的资源
 - 虚拟机
 - 对象存储
 - 负载均衡
 - 数据库（读写分离）
 - 消息队列
 - 分布式缓存
- 可扩展的架构



面向弹性的设计

- 容器化
- VM挂掉应被当做是『常态』
- 针对不同的云服务的不同容灾策略
 - 大原则：切 切 切
 - LB挂掉的处理
 - VM挂掉的处理



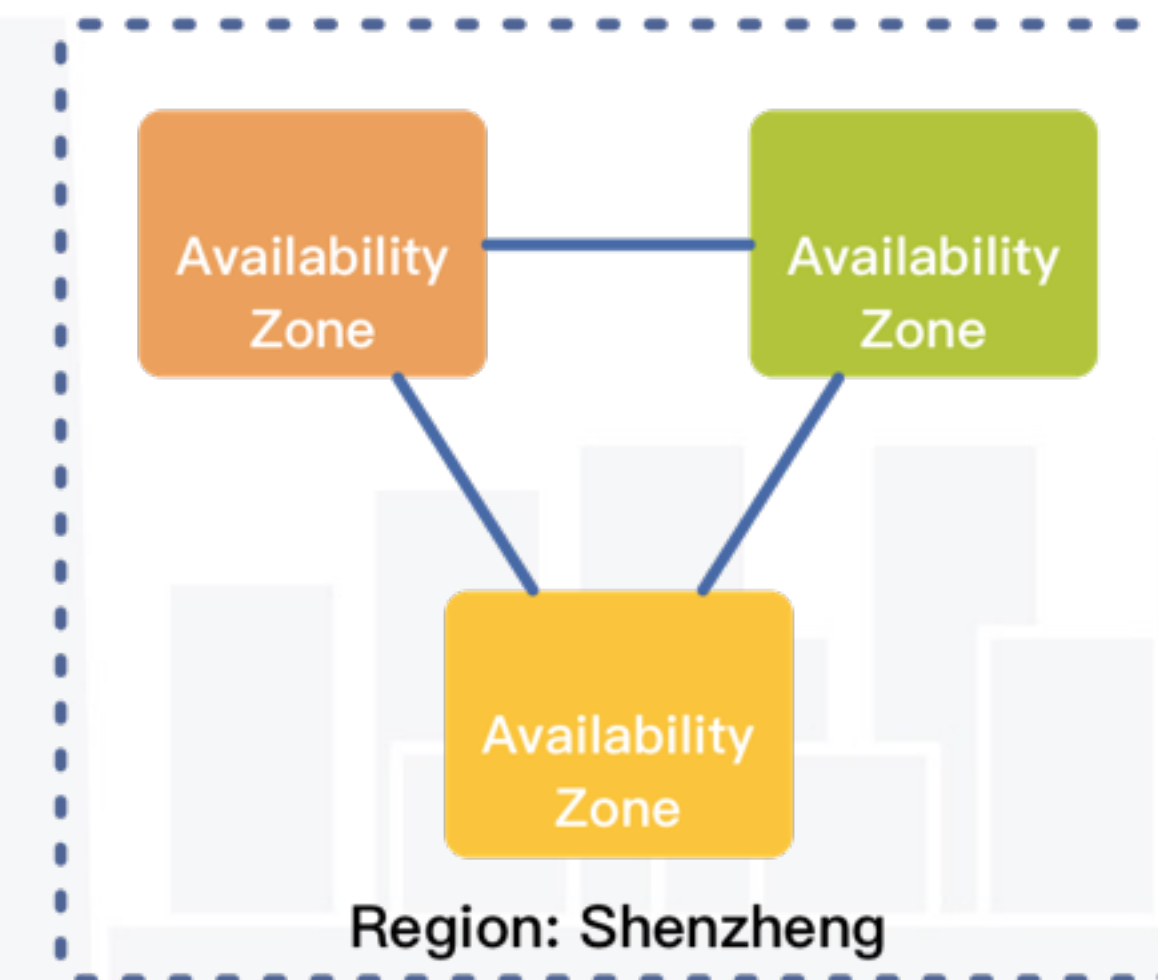
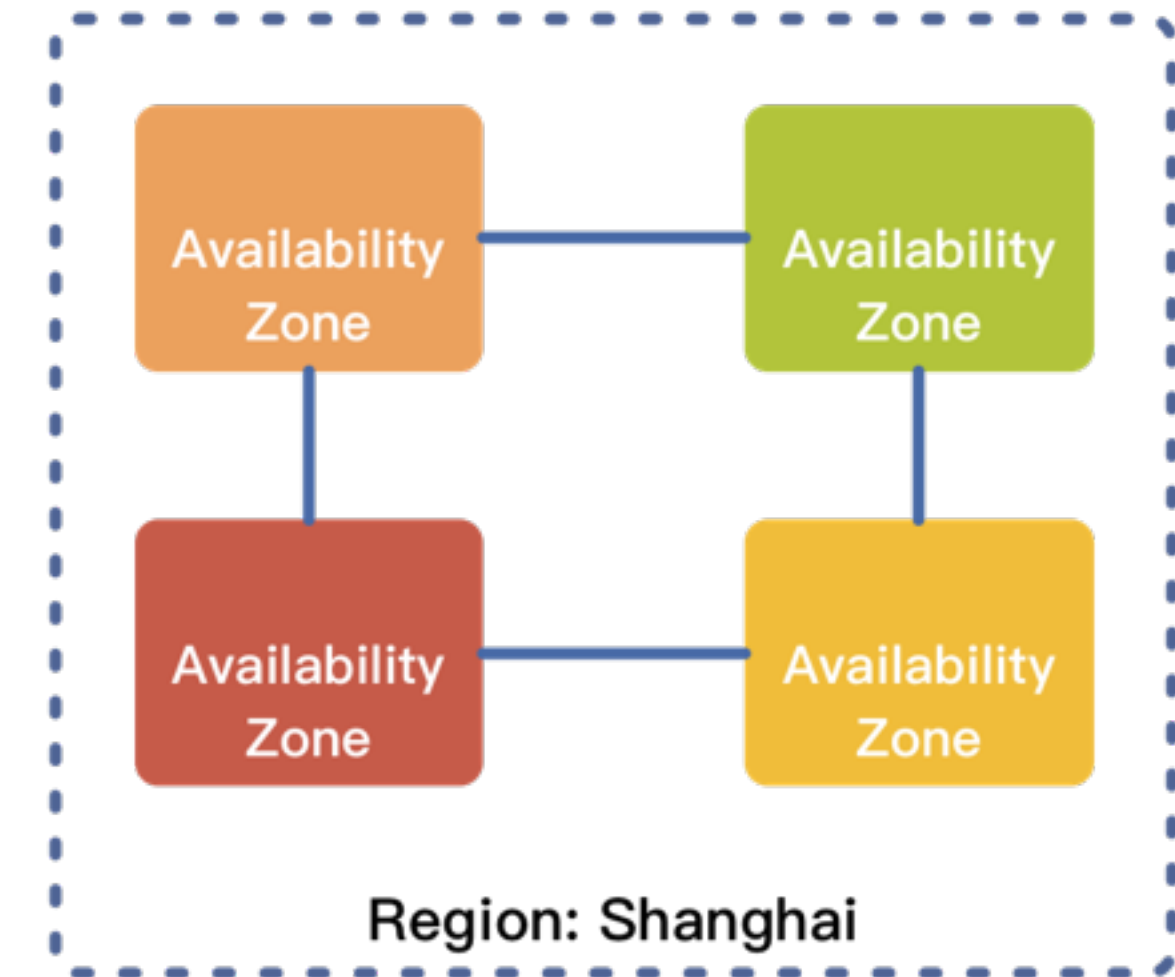
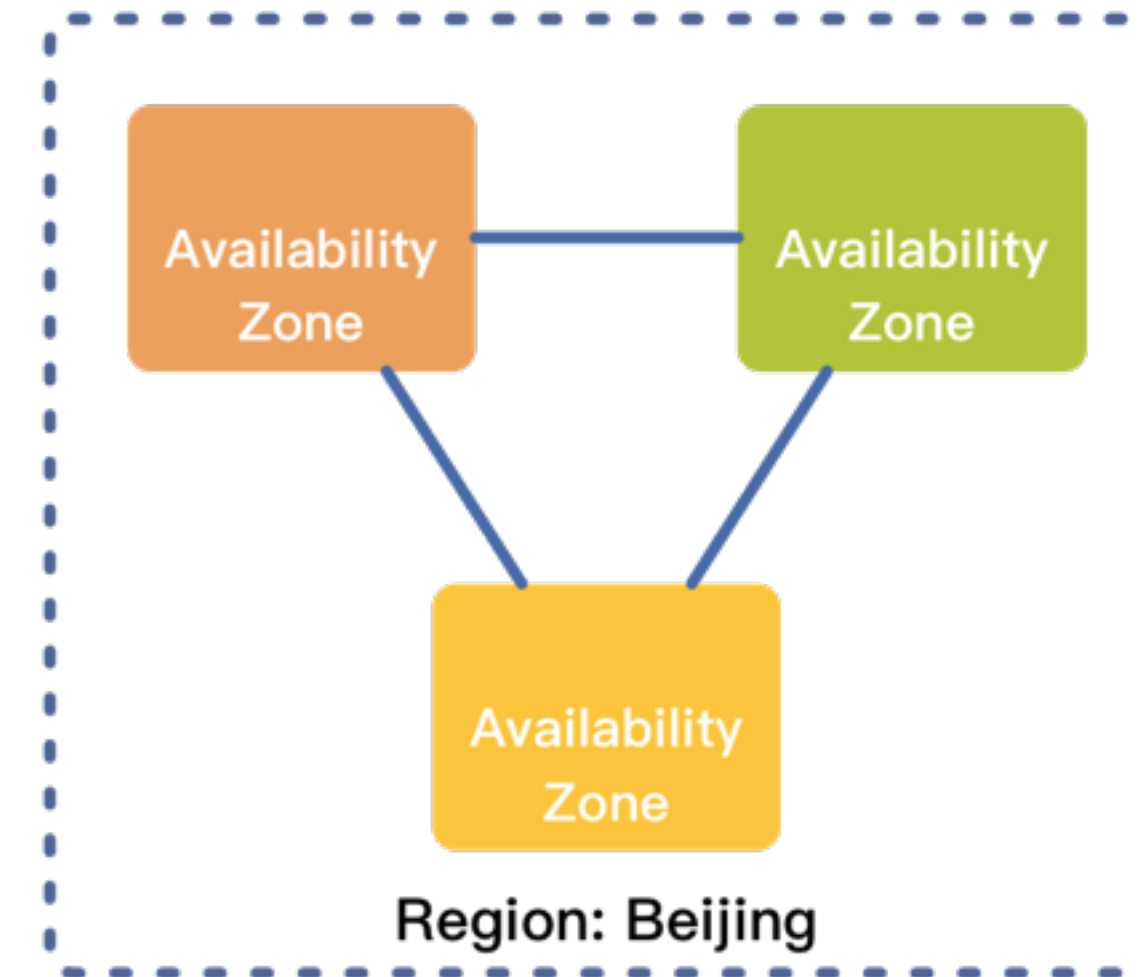
弹性是万能的吗

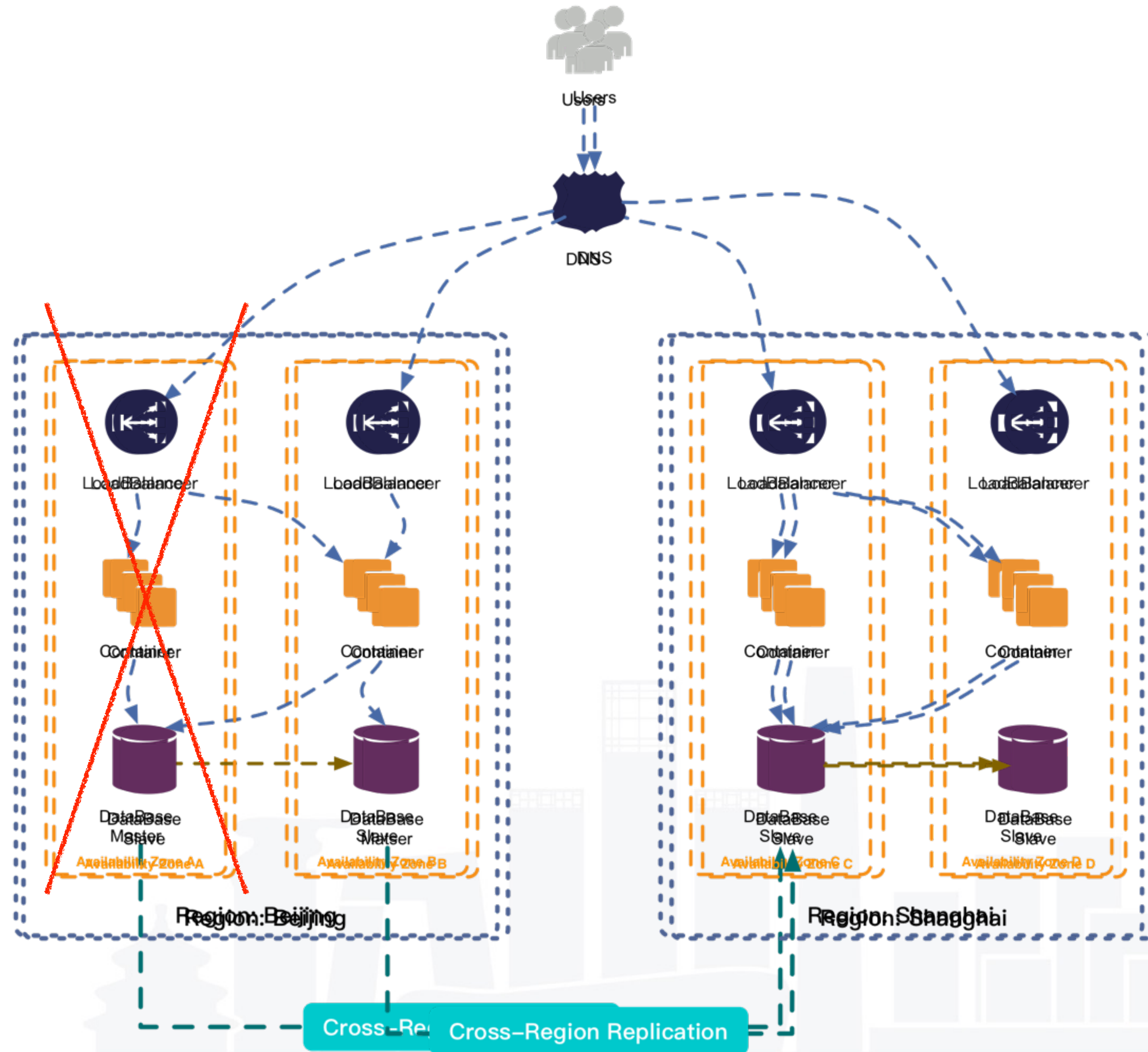
- Auto Scaling 不是银弹
- 容量规划仍然有必要性
- 弹性的开销
- 弹性期的SLA

弹性 + 限流 = 组合拳

面向跨区域/可用区的设计

- Region 和 Availability Zone
- 几个数字
 - 1.45: 光纤折射率
 - 2ms: 逻辑上被视为同机房的延时
- 基于多Region/AZ的架构
 - 应用层设计
 - 数据层设计
 - case: Netflix如何在S3故障中幸免于难
 - 负载均衡层设计





寻找一起奔跑的伙伴

- Java资深工程师、技术专家、高级技术专家
- 你能得到的：
 - 阿里最前线的作战经验，直接参与2017年双11大促作战
 - 超赞的技术氛围，鲁肃、毕玄、褚霸等阿里全集团顶尖专家分享（2周一次）
 - 飞速成长的发动机，一对一师兄带领，快速度过适应期
 - 新零售、新技术，创造下一个时代的商业文明，培养兼具技术和商业sense
- 请拿简历砸我
 - mujian.wcc@alibaba-inc.com



关注QCon微信公众号，
获得更多干货！

Thanks!



INTERNATIONAL SOFTWARE DEVELOPMENT CONFERENCE

主办方 **Geekbang**  **InfoQ**
极客邦科技