

高性能、高可用服务化架构漫谈

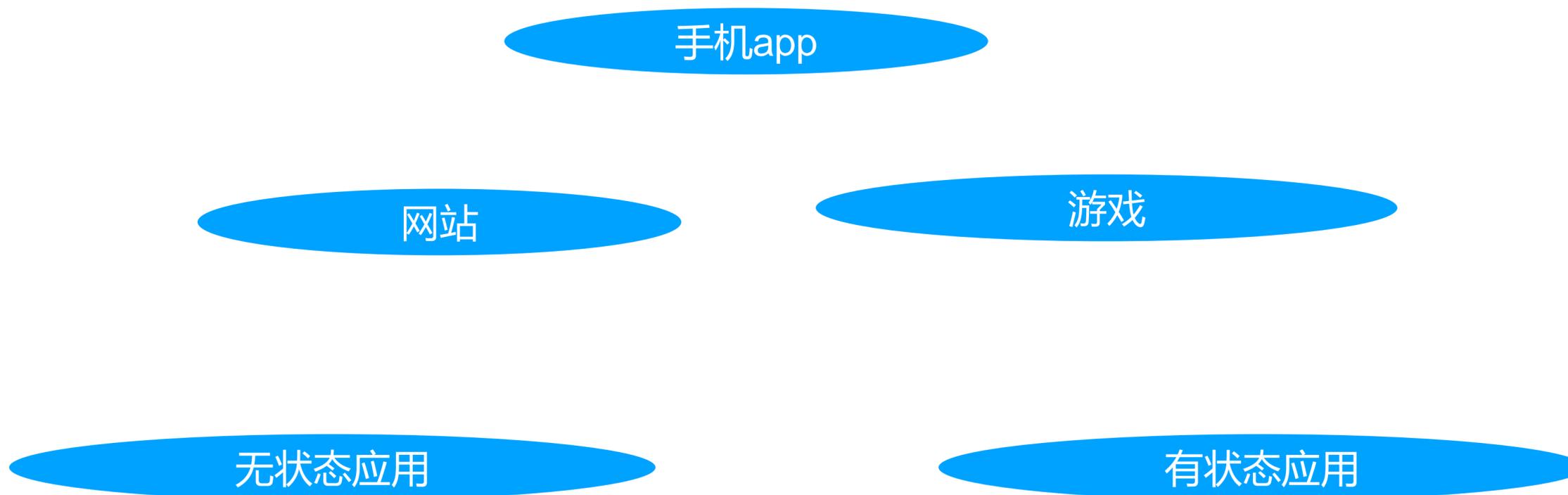
曾宇星 网易云资深解决方案架构师

目录

- 01 应用的常见架构，架构设计要素
- 02 服务化架构之高性能RPC 实现
- 03 架构怎么做到高可用？
- 04 架构未来演进，云端如何快速落地



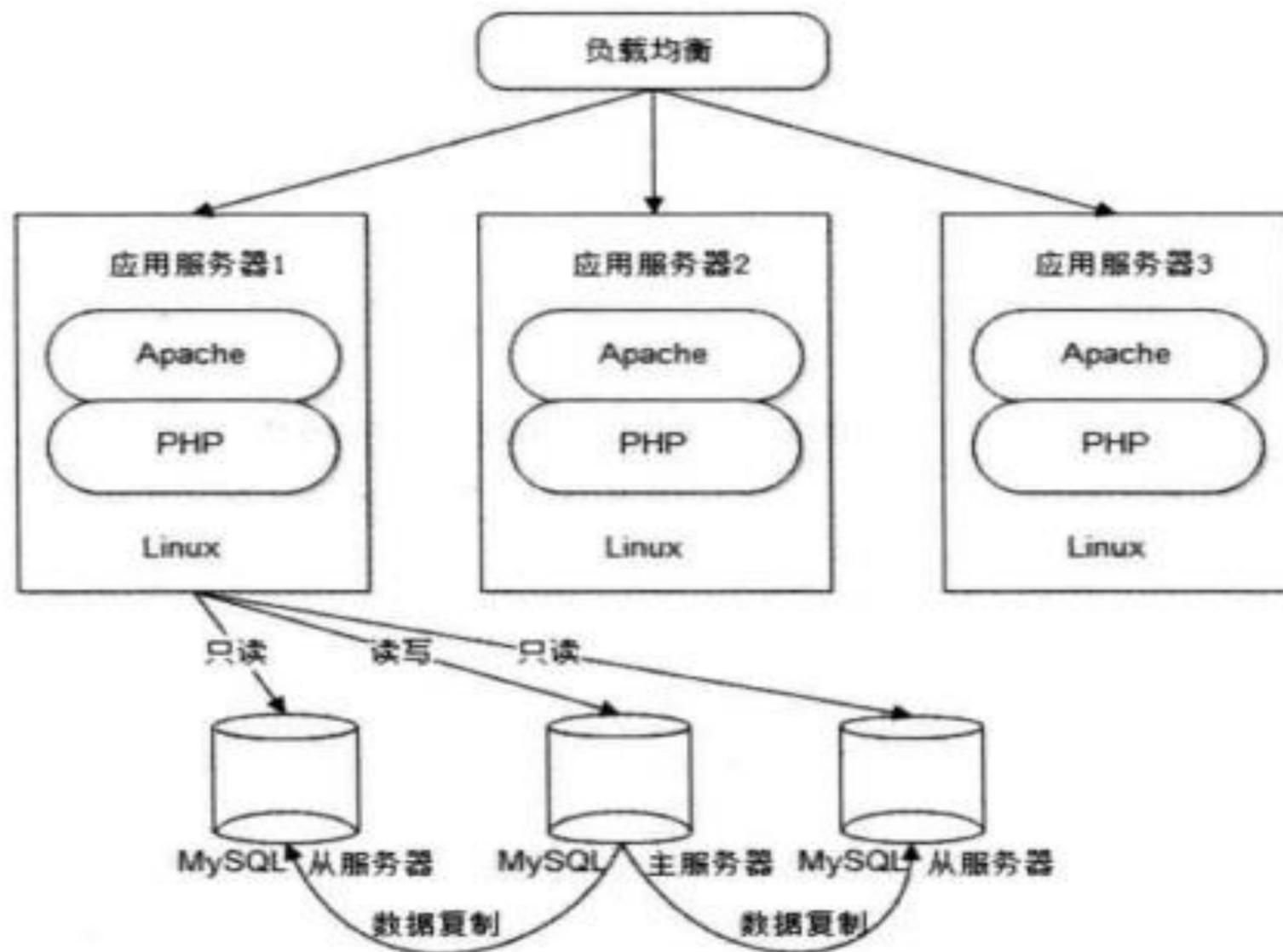
典型互联网应用



具体应用架构都不一样

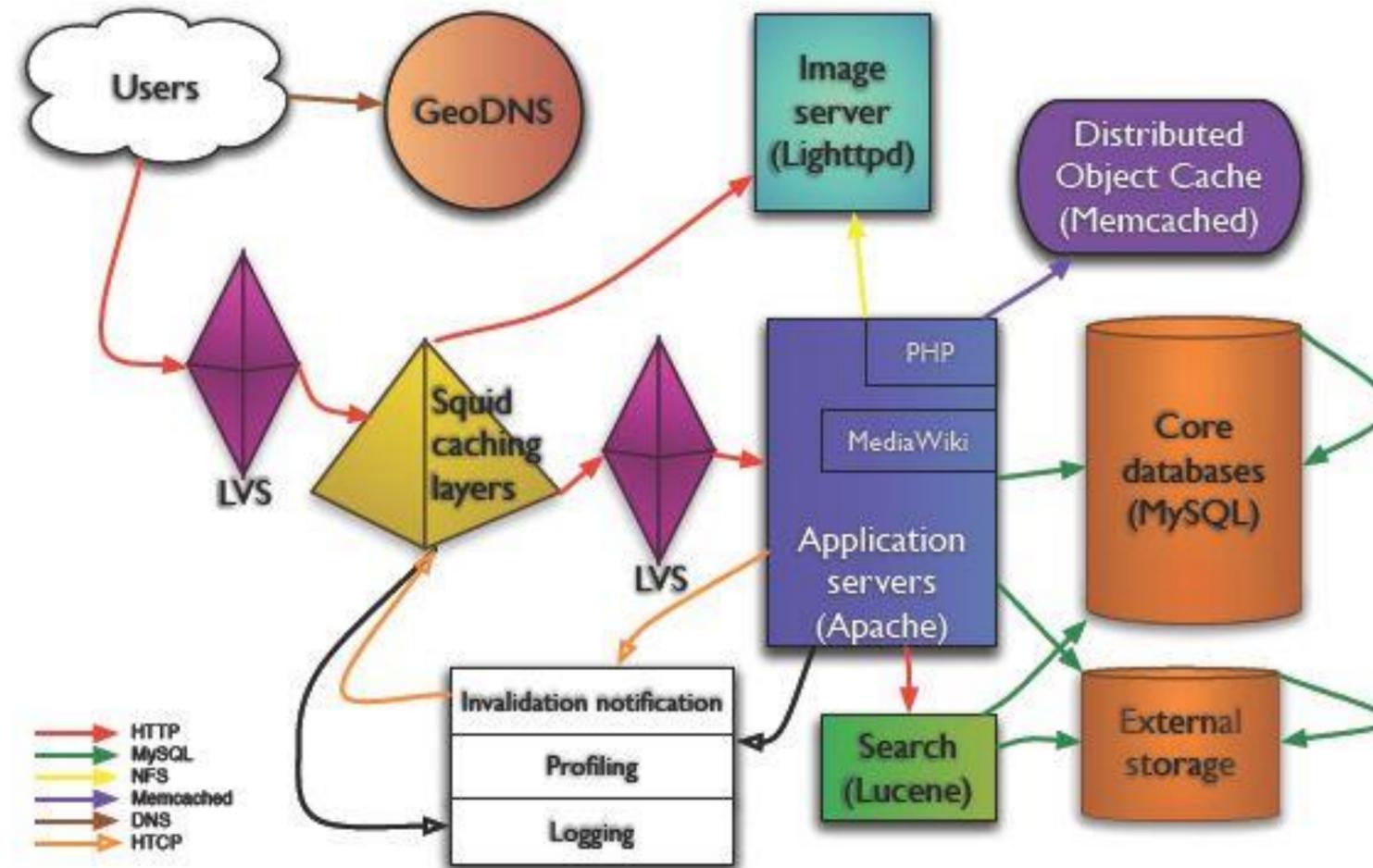


2003 年某宝网架构

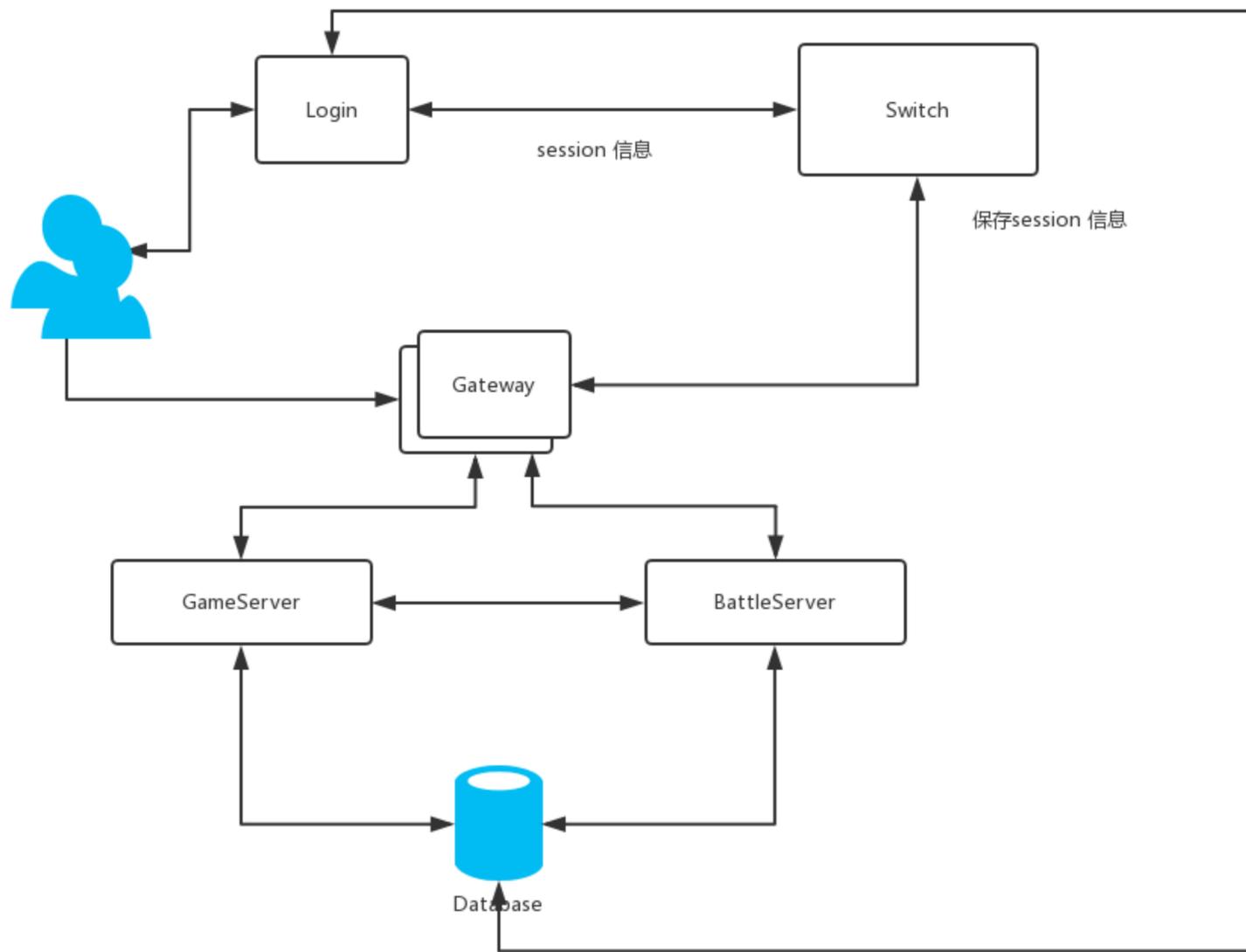




Wikipedia 架构

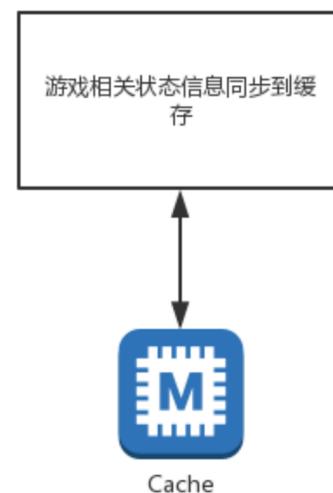


游戏-有状态应用



Tcp 长连接
短连接

Pvp
即时战斗



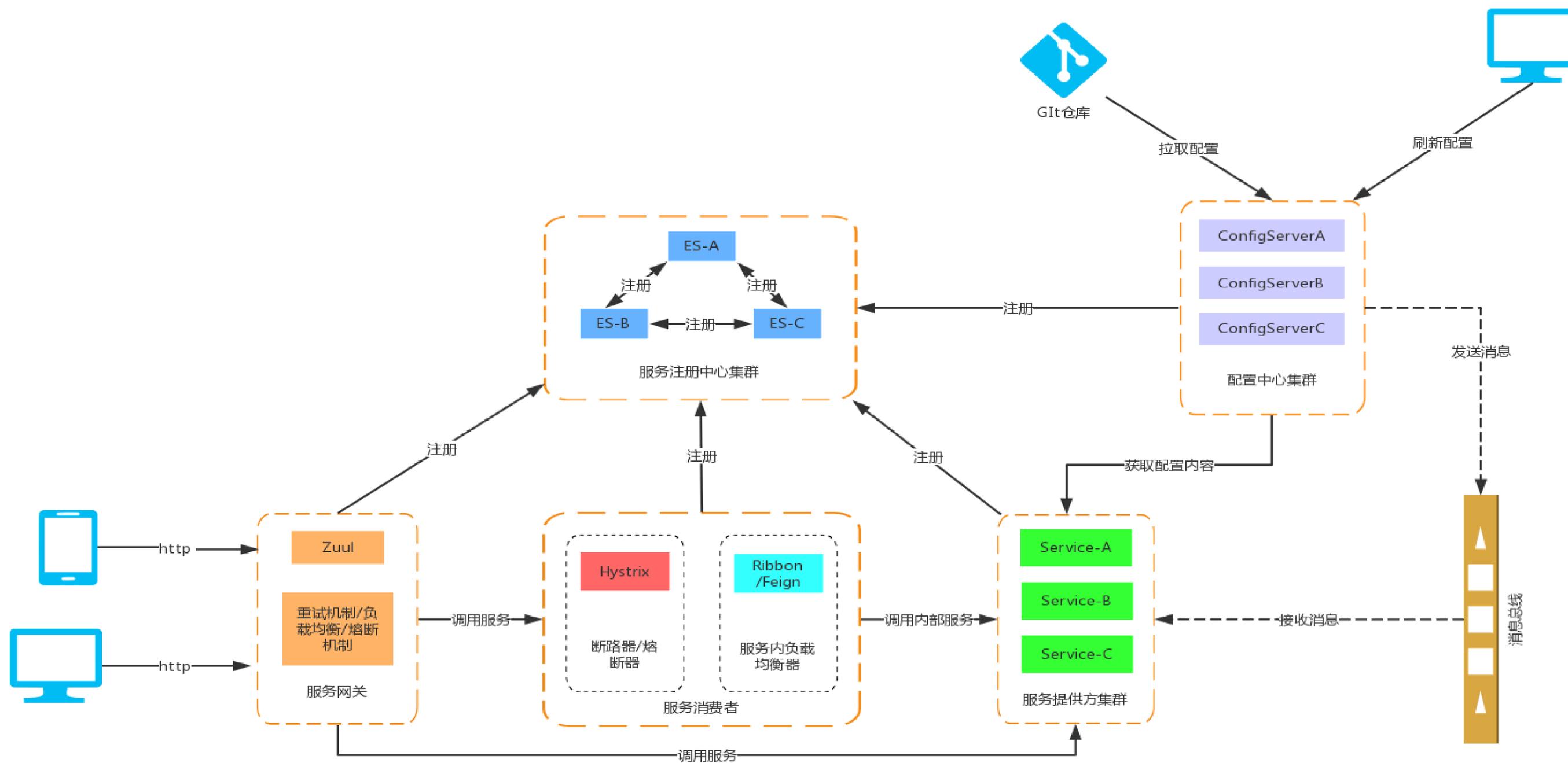
状态和应用剥离：

状态信息同步更新到缓存
Redis/MemCache
应用服务宕机，游戏状态信息可以立即从缓存恢复，重建应用服务

客户端应用：
连接断开重连处理，使玩家无感知

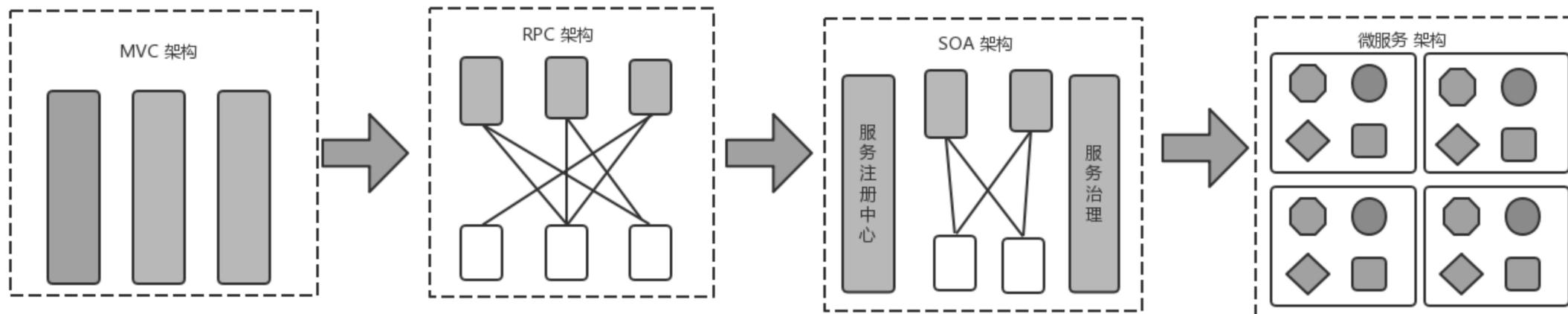


基于Spring Cloud 的微服务架构

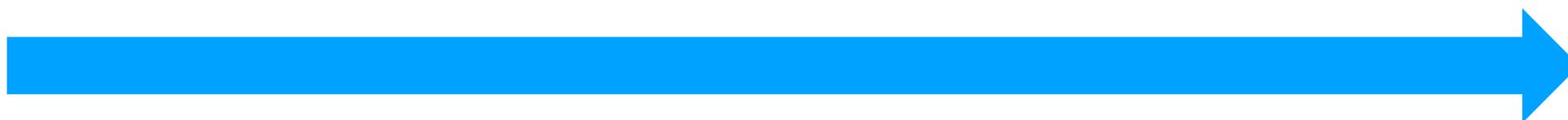




架构演变



关键：是否满足业务发展需求



业务发展、用户规模不断扩大

传统架构 VS 服务化架构

传统架构

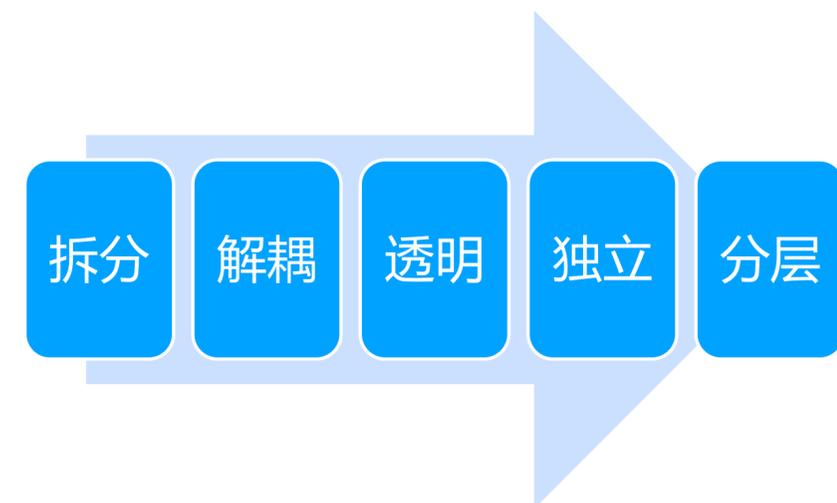
研发成本高

- 代码重复率高（顶多有公共类库，但公共类库维护及版本升级困难）
- 需求变更困难（user数据库变化，各个模块都需要修改）
- 无法满足新业务快速创新和敏捷交付

运维效率低

- 测试、部署成本高：业务运行在一个进程中，因此系统中任何程序的改变，都需要对整个系统重新测试并部署
- 可伸缩性差：水平扩展只能基于整个系统进行扩展，无法针对某一个功能模块按需扩展
- 可靠性差：某个应用BUG，例如死循环、OOM等，会导致整个进程宕机，影响其它合设的应用

解决对策-服务化架构





架构设计相关定律

墨菲定律

二八定律

康威定律



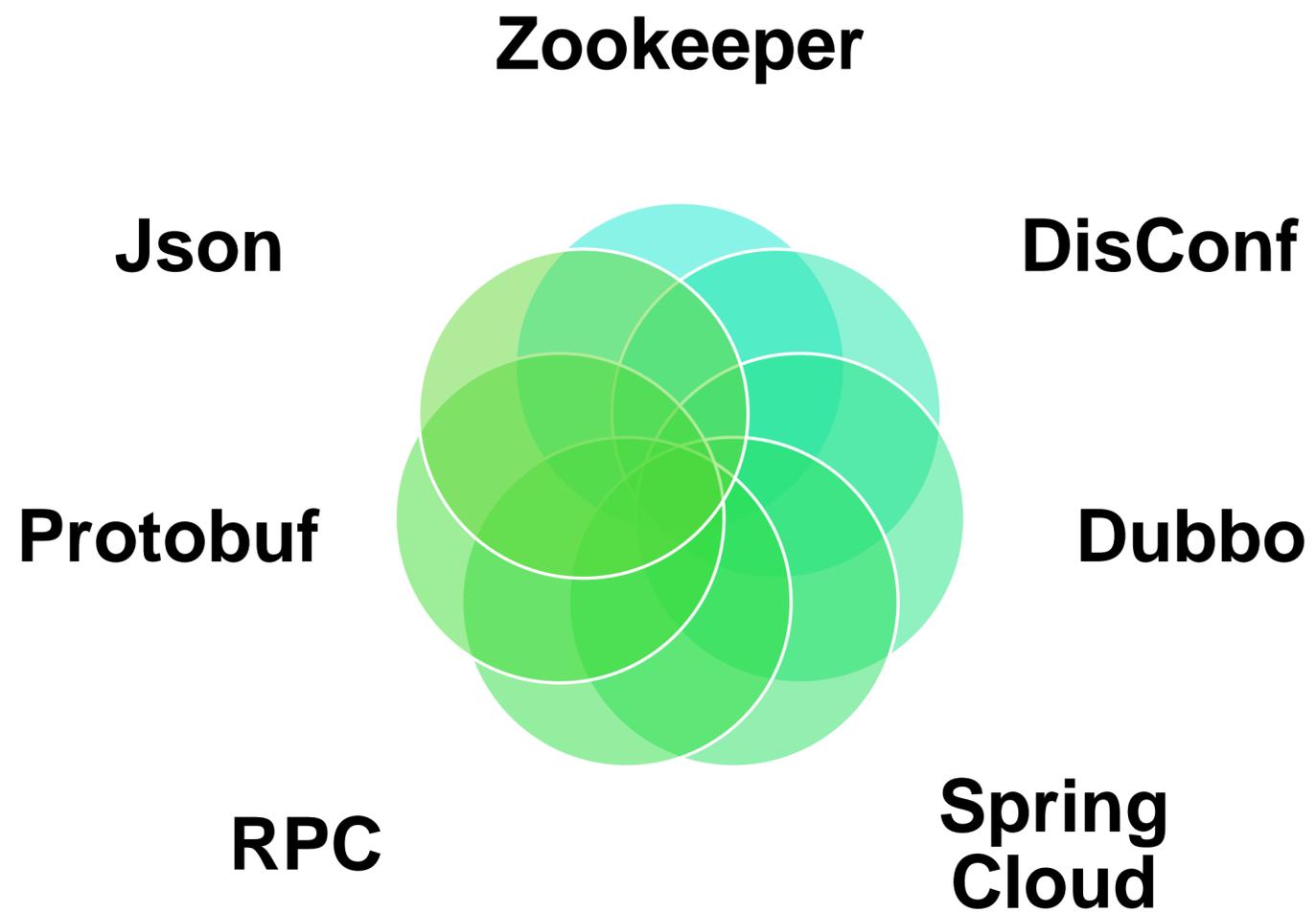
分布式架构要素



性能、稳定性、用户体验之间做平衡

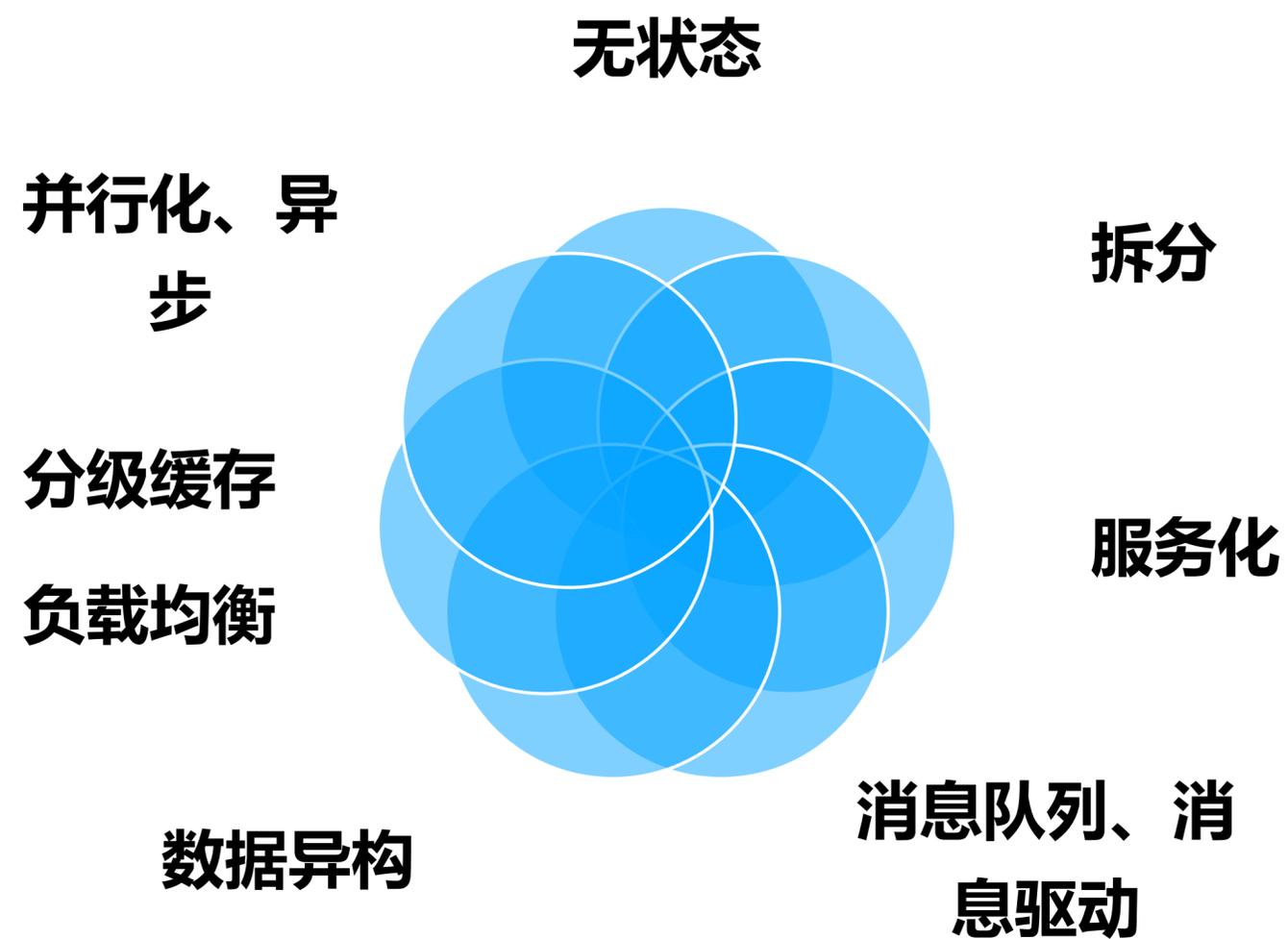


服务化架构-关键词





高性能/并发架构 原则





高性能服务器架构 (RPC)

第一，服务端尽可能多的处理并发请求
第二，同时尽可能短的处理完毕。



高性能服务器架构-同步&异步

IO模型：

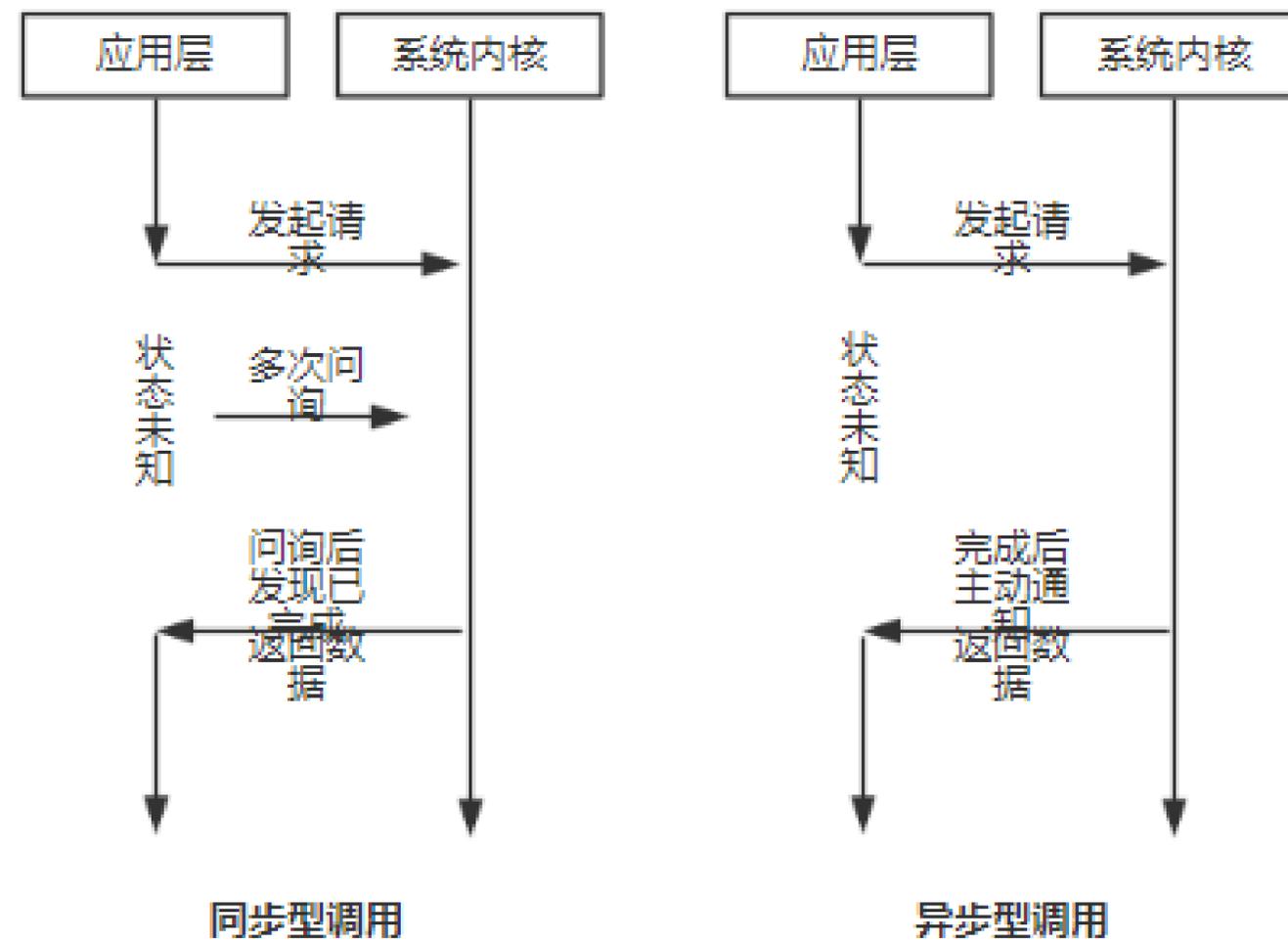
1. 传统的阻塞 I/O (Blocking I/O)
2. 非阻塞 I/O (Non-blocking I/O)
3. **I/O 多路复用，两种I/O复用设计模式，
Reactor、Proactor**
4. 异步 I/O (Asynchronous I/O)

高性能服务器架构-同步&异步

易混淆概念：

- ◆ 同步& 异步
- ◆ 阻塞& 非阻塞

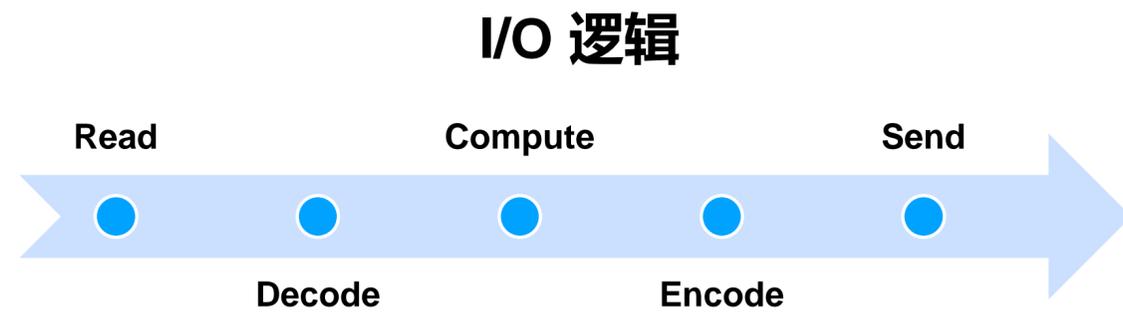
关注点不同，同步和异步关注的是多件事情是否可以并发，而阻塞和非阻塞关注的是程序等待调用结果时的状态。





高性能服务器架构-并发

进程 / 线程模型



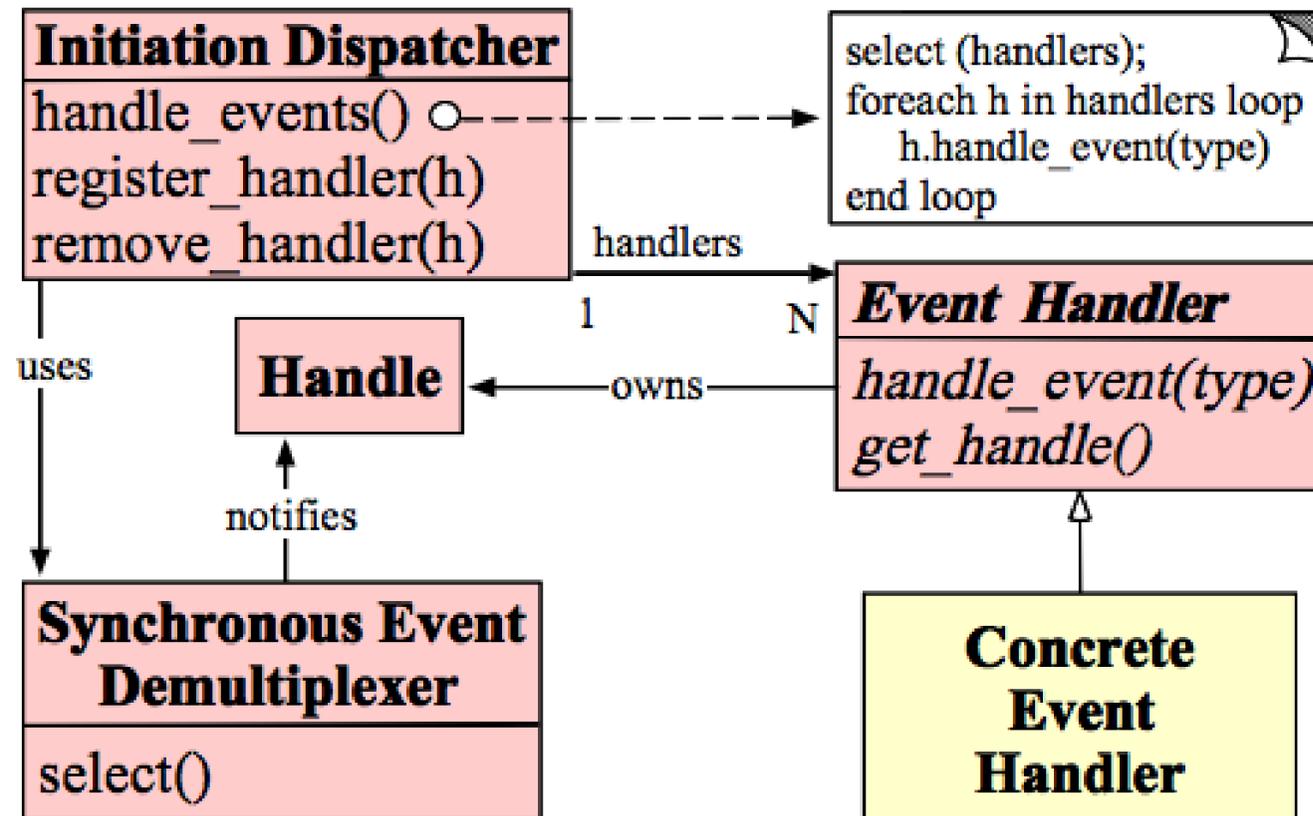
传统的阻塞 I/O + 线程池 -> C10K 问题

I/O 多路复用 -> Reactor 模式 (Redis、Nginx、Node.js、Netty)

异步I/O -> Proactor (Windows IOCP)



Reactor 模式



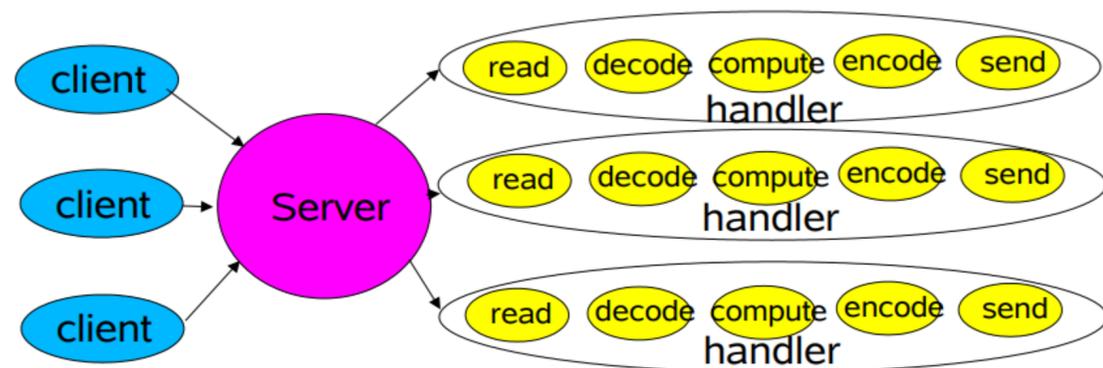
I/O 复用

- Select
- Poll
- Epoll
- kqueue

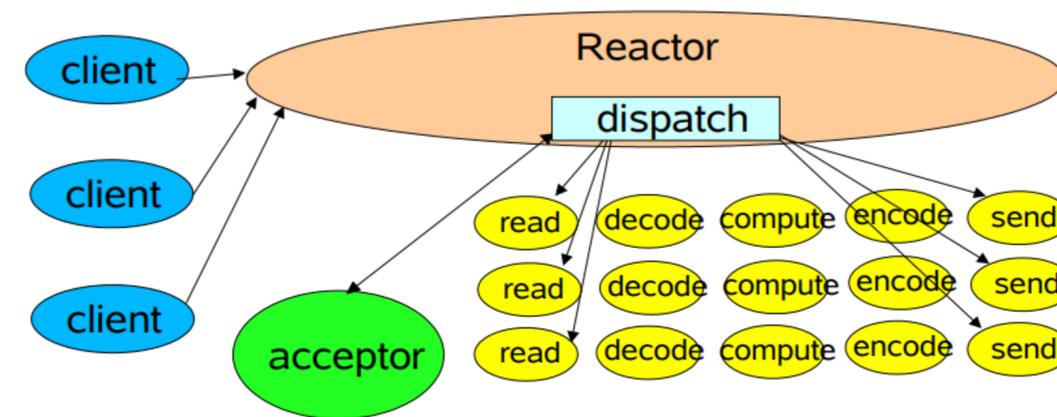
框架

- Libevent
- Netty

Reactor 模式-单线程



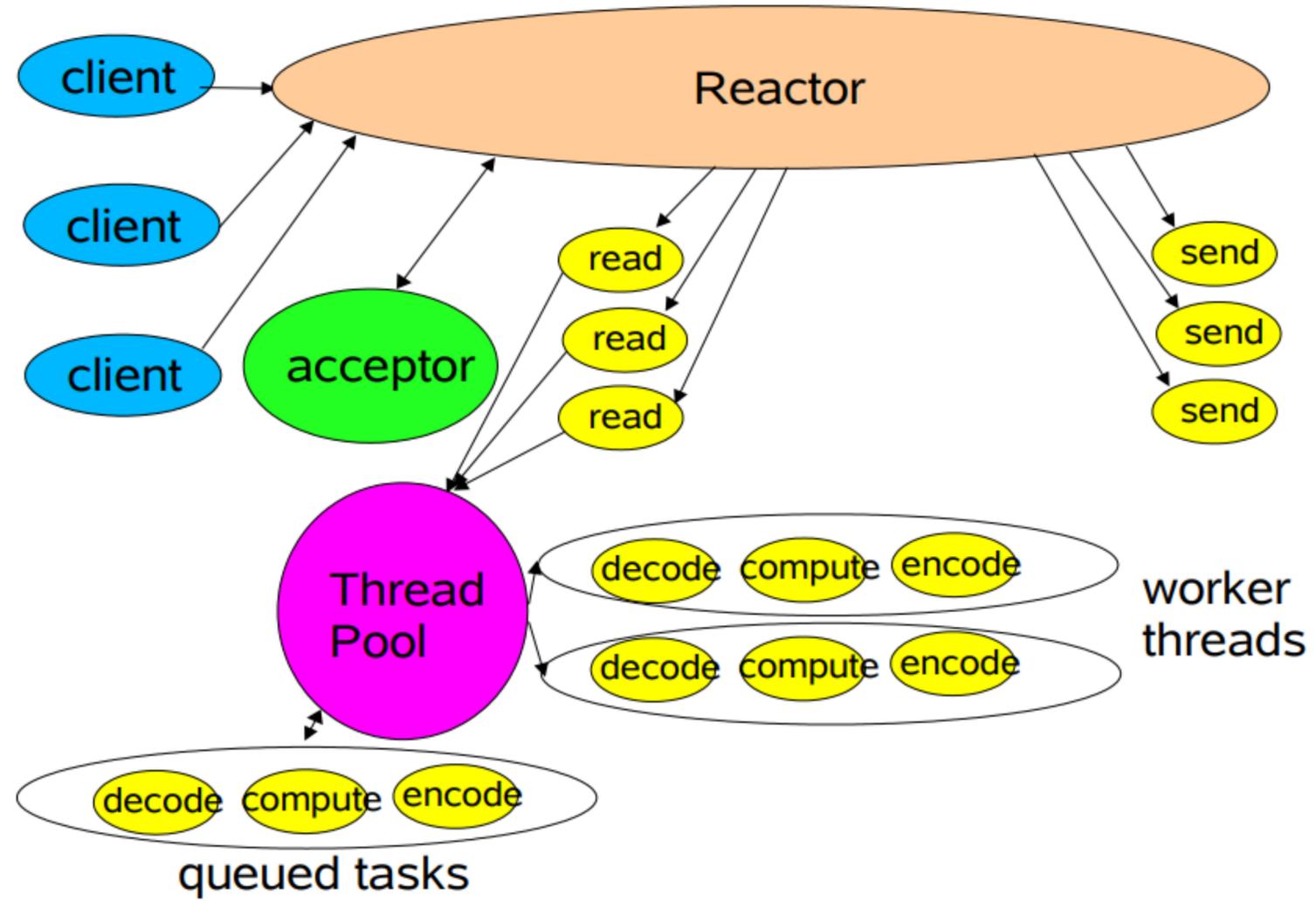
Each handler may be started in its own thread



Single threaded version

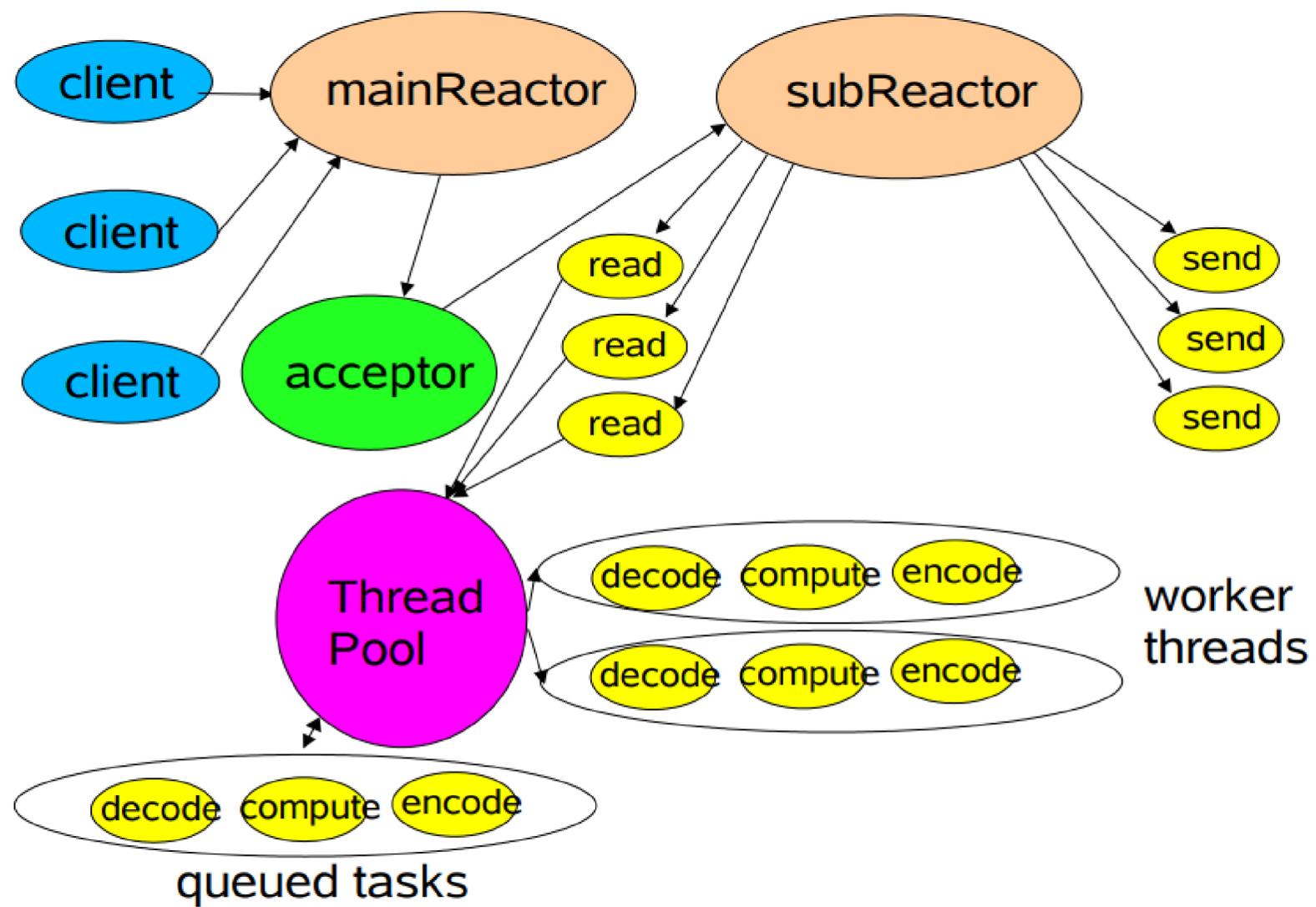


Reactor 模式-多线程



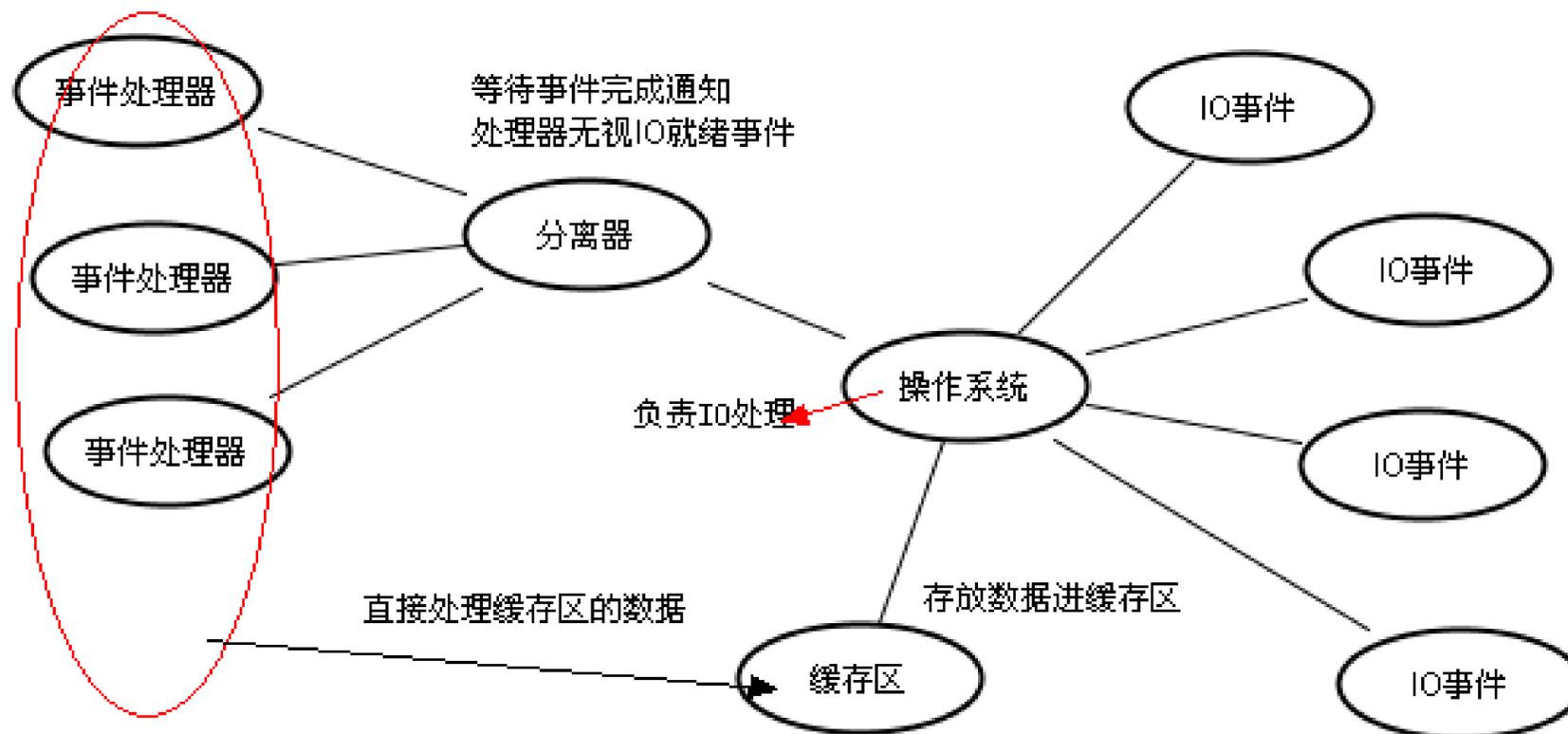


主从Reactor/ Multi-Reactor





Proactor 模式



Proactor模式：
操作系统必须支持异步IO



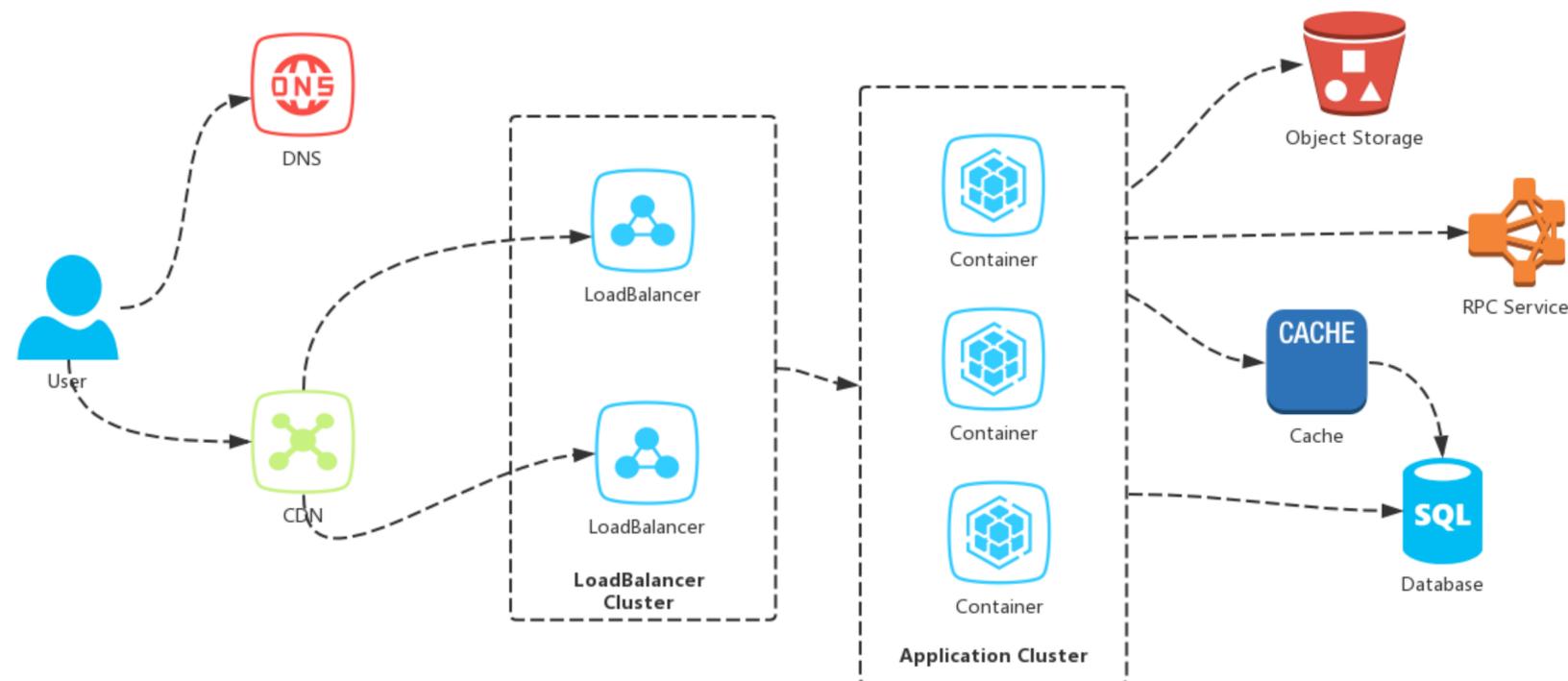
高性能服务化架构-分级缓存

按端划分：

- ◆ 客户端缓存（移动app、浏览器）
- ◆ 服务端缓存

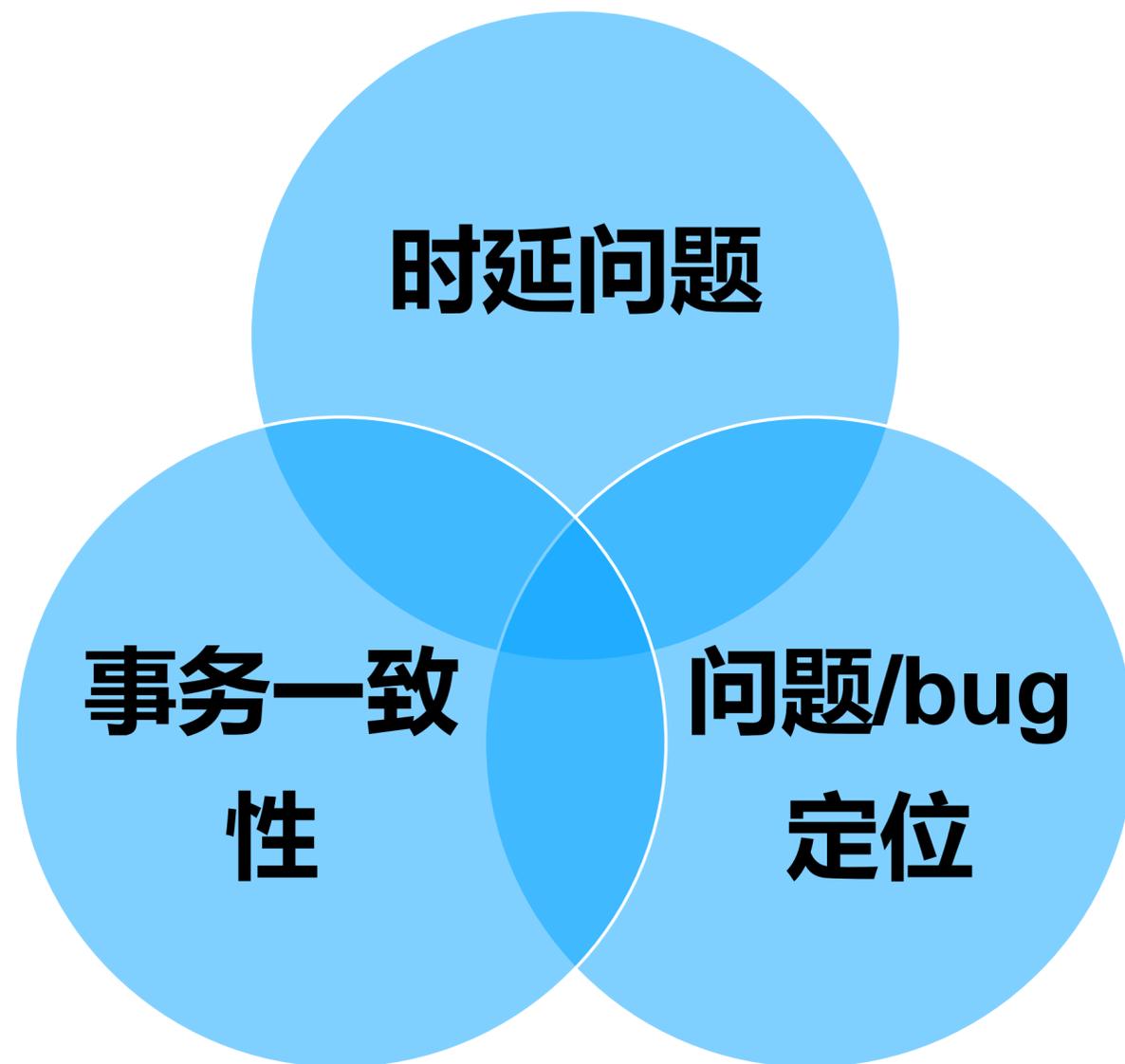
常见缓存：

- ◆ 浏览器缓存（http 缓存）
- ◆ CDN
- ◆ 反向代理（Nginx 缓存）
- ◆ 分布式缓存 redis、memcache等





服务化不是银弹





衡量系统可用性指标SLA

SLA：俗称几个9

不同公司的计算标准不一致

大体是：

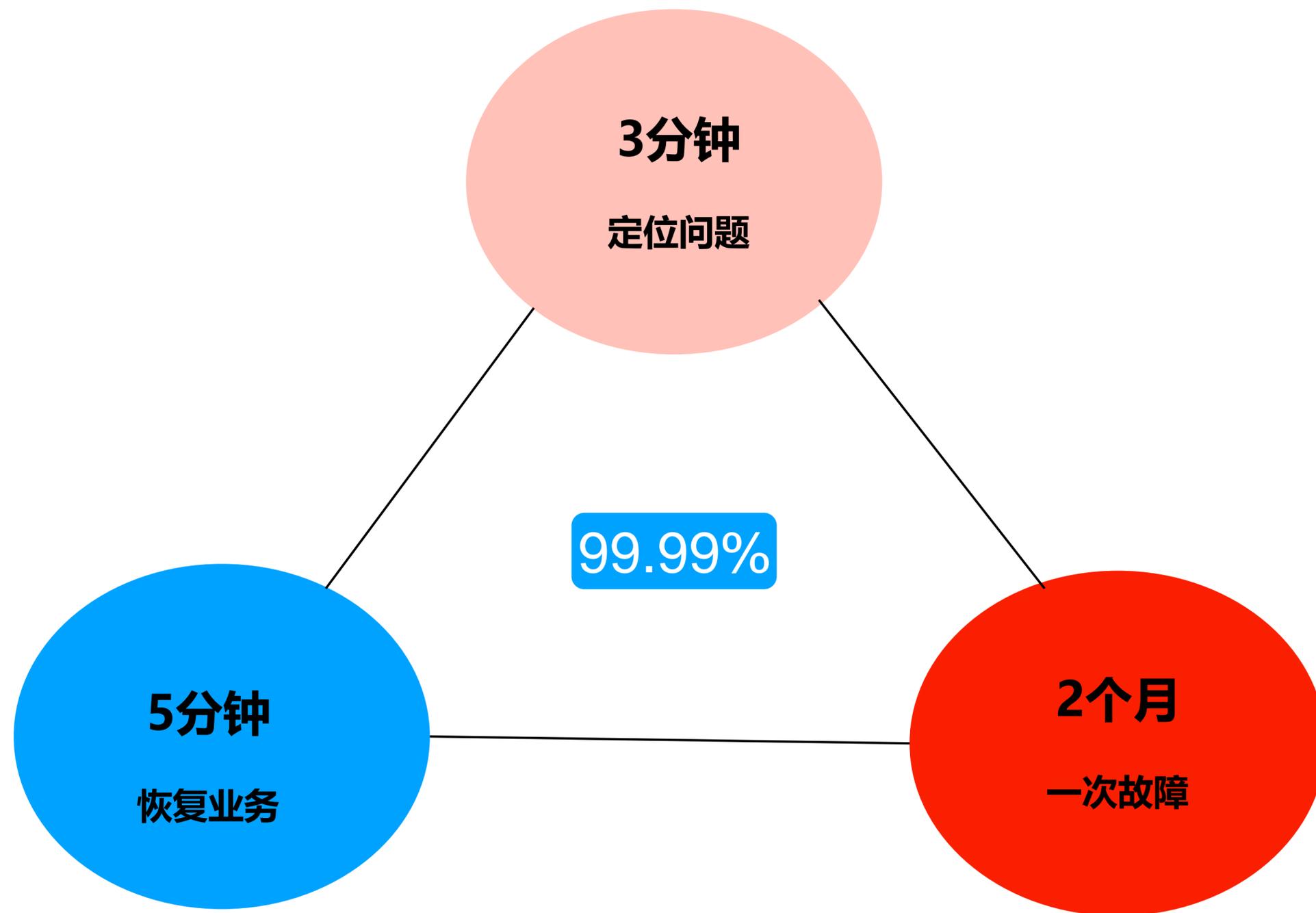
服务周期内所有可用时间 / 服务周期总时间。

Availability %	Downtime per year	Downtime per month	Downtime per week	Downtime per day
90% ("one nine")	36.5 days	72 hours	16.8 hours	2.4 hours
95% ("one and a half nines")	18.25 days	36 hours	8.4 hours	1.2 hours
97%	10.96 days	21.6 hours	5.04 hours	43.2 minutes
98%	7.30 days	14.4 hours	3.36 hours	28.8 minutes
99% ("two nines")	3.65 days	7.20 hours	1.68 hours	14.4 minutes
99.5% ("two and a half nines")	1.83 days	3.60 hours	50.4 minutes	7.2 minutes
99.8%	17.52 hours	86.23 minutes	20.16 minutes	2.88 minutes
99.9% ("three nines")	8.76 hours	43.8 minutes	10.1 minutes	1.44 minutes
99.95% ("three and a half nines")	4.38 hours	21.56 minutes	5.04 minutes	43.2 seconds
99.99% ("four nines")	52.56 minutes	4.38 minutes	1.01 minutes	8.64 seconds
99.995% ("four and a half nines")	26.28 minutes	2.16 minutes	30.24 seconds	4.32 seconds
99.999% ("five nines")	5.26 minutes	25.9 seconds	6.05 seconds	864.3 milliseconds
99.9999% ("six nines")	31.5 seconds	2.59 seconds	604.8 milliseconds	86.4 milliseconds
99.99999% ("seven nines")	3.15 seconds	262.97 milliseconds	60.48 milliseconds	8.64 milliseconds
99.999999% ("eight nines")	315.569 milliseconds	26.297 milliseconds	6.048 milliseconds	0.864 milliseconds
99.9999999% ("nine nines")	31.5569 milliseconds	2.6297 milliseconds	0.6048 milliseconds	0.0864 milliseconds

比如我们说月度99.95%的SLA，意味着每个月服务出现故障的时间只能占总时间的0.05%，如果这个月是30天，那么就是21.6分钟。



更直观的SLA



聚焦于业务
容易分解
衡量



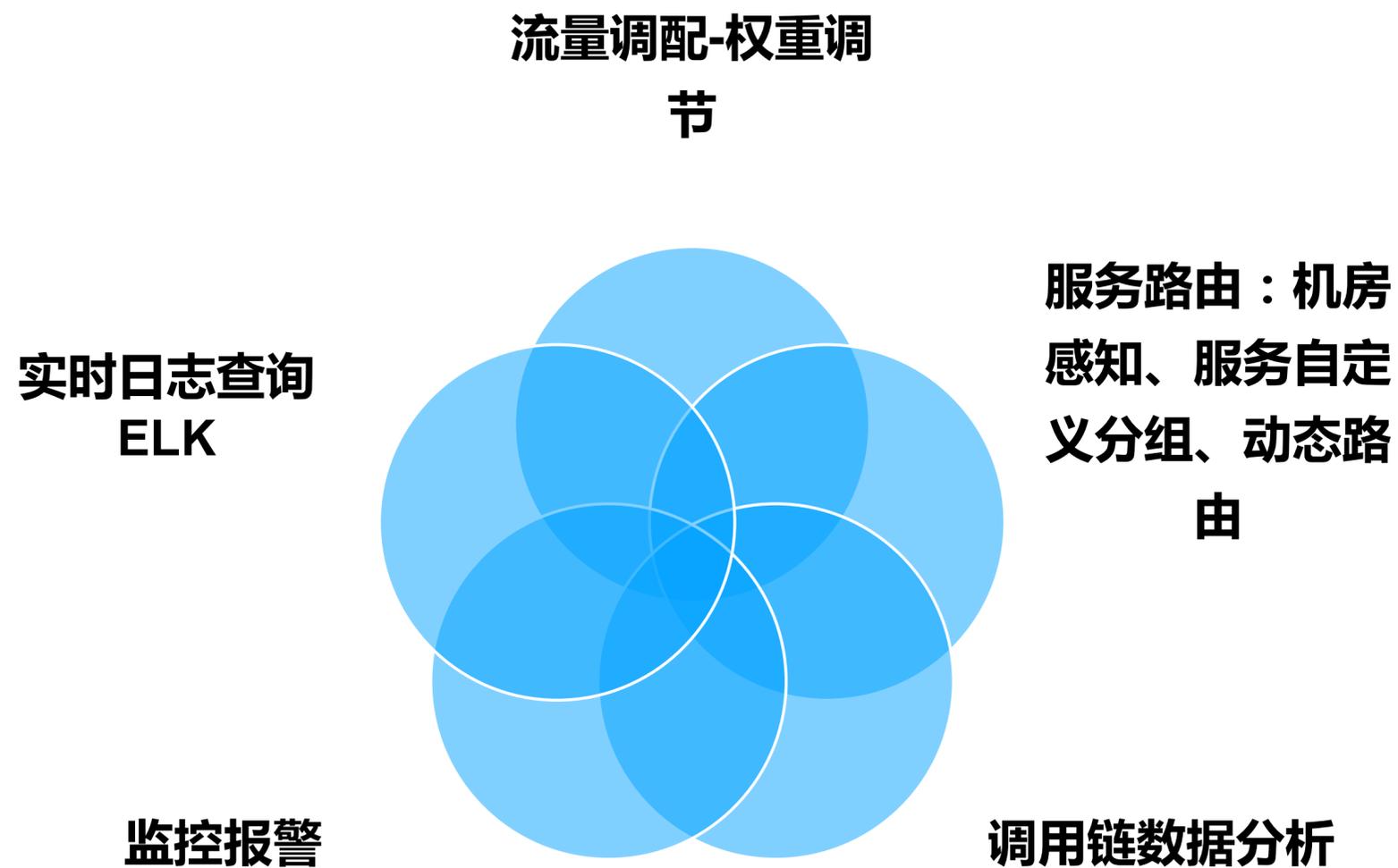
解决系统可用性的手段

◆减少故障出现次数
—— 不出问题

◆缩短故障处理时间
—— 尽快解决

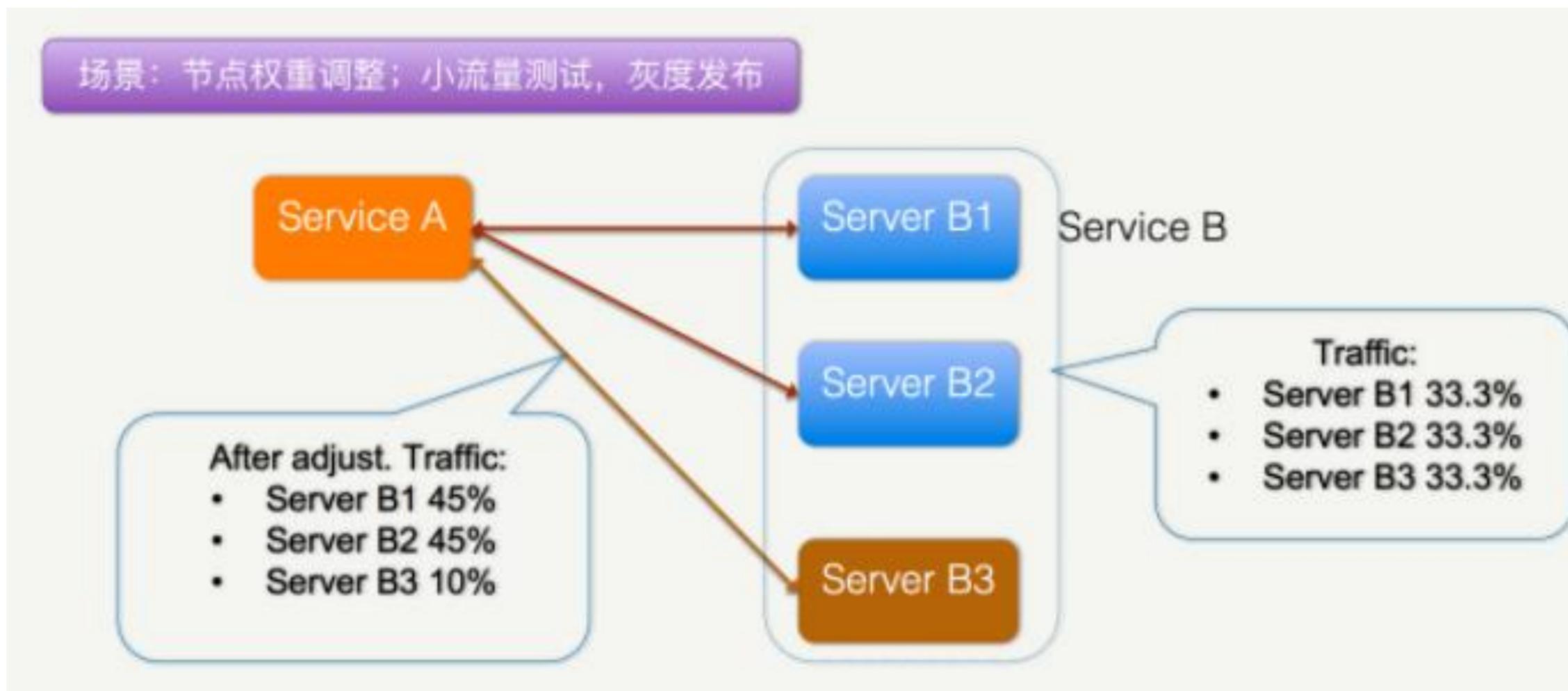


服务化实践-服务治理

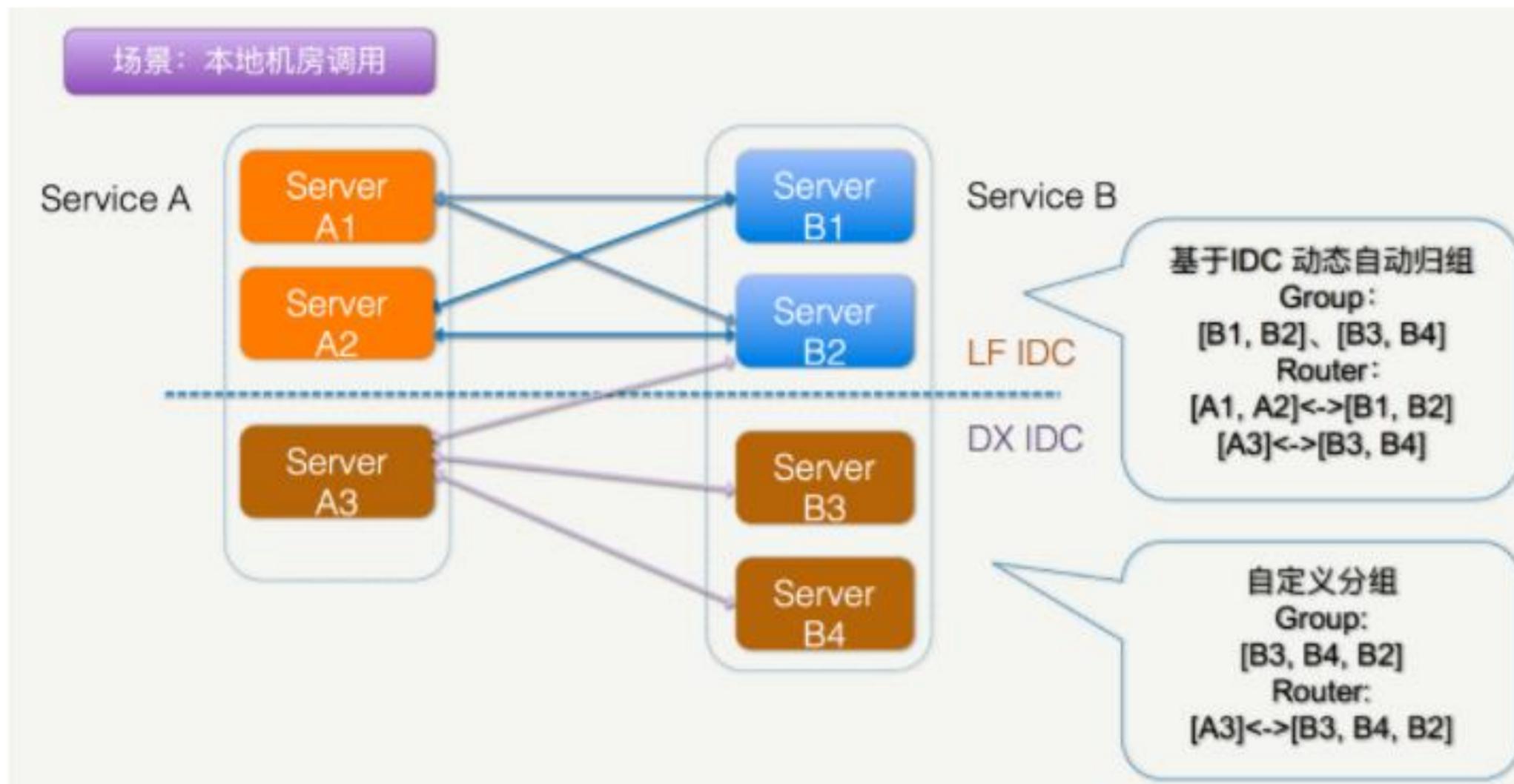




服务化实践-服务治理-权重调节

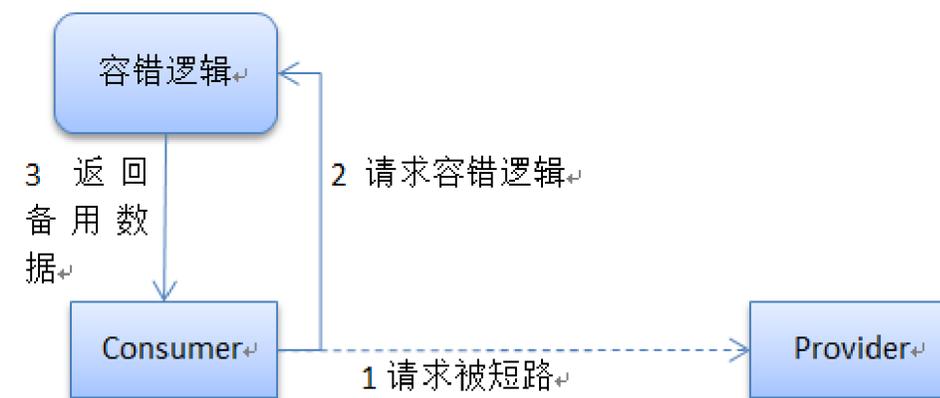
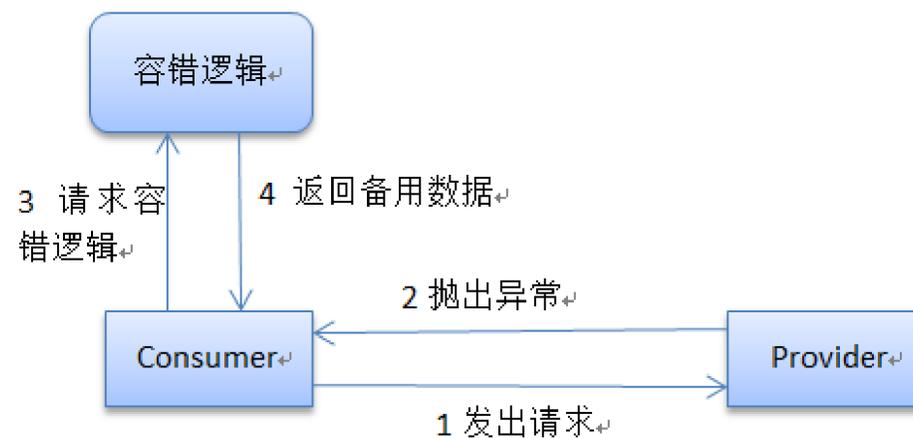
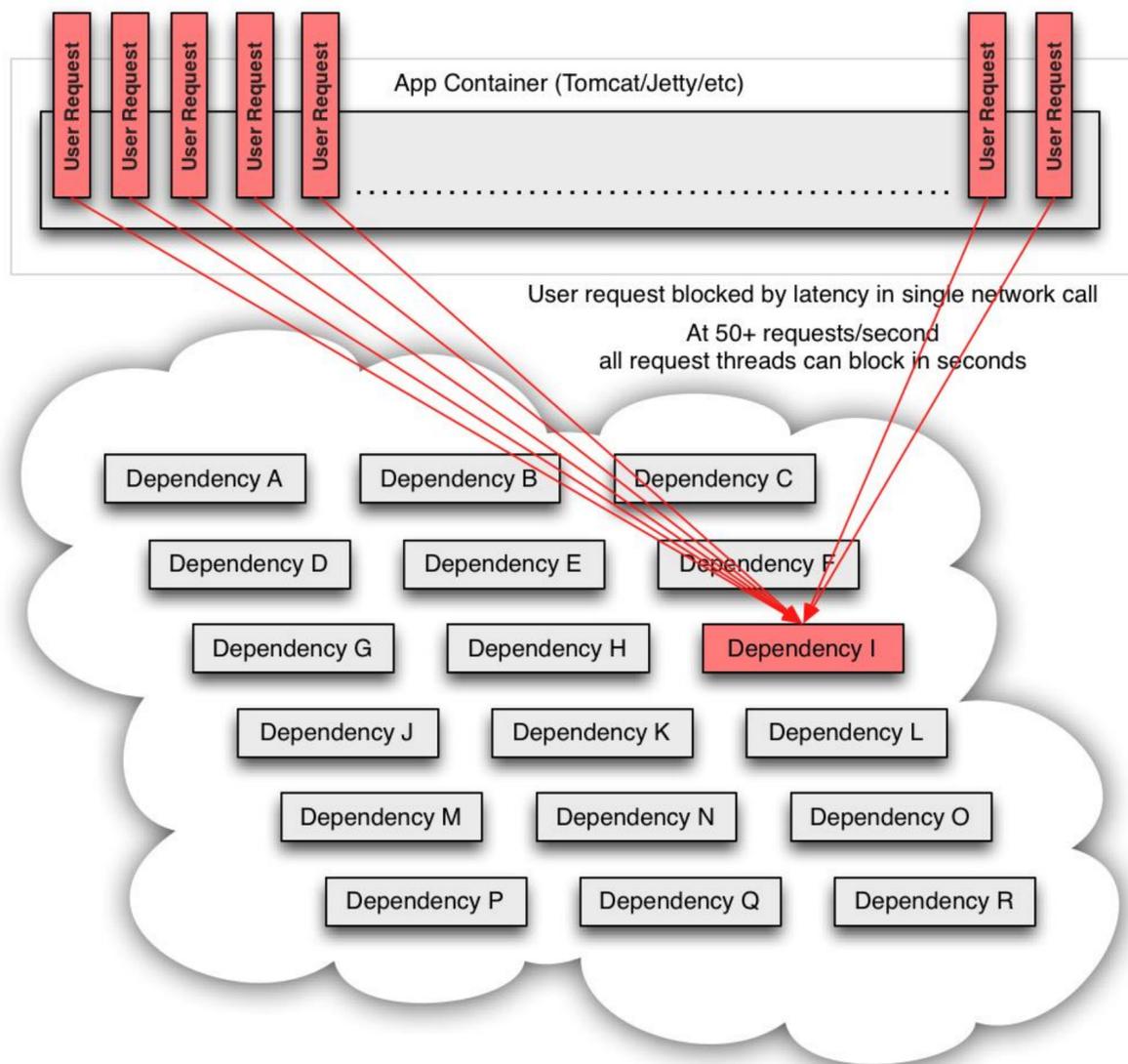


服务治理-本地机房调用





服务化治理-故障隔离





服务化实践-监控

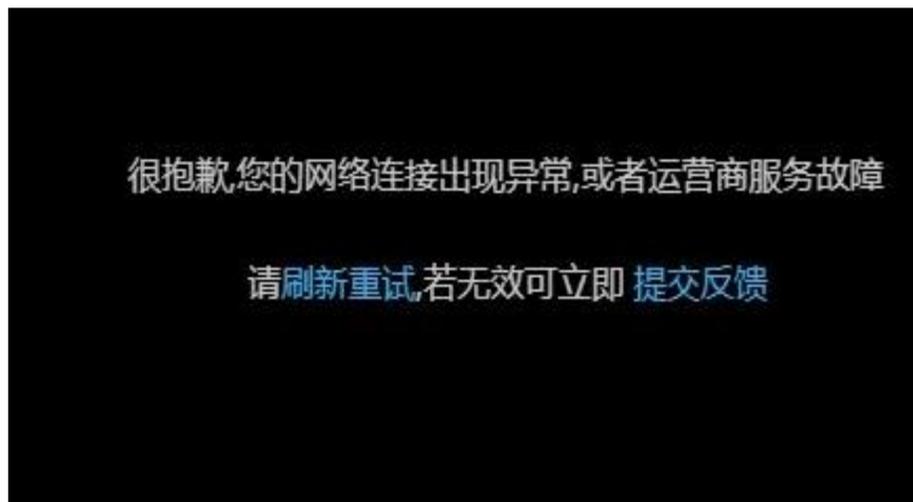




单实例故障

原因：

网络中断
机器故障
程序崩溃

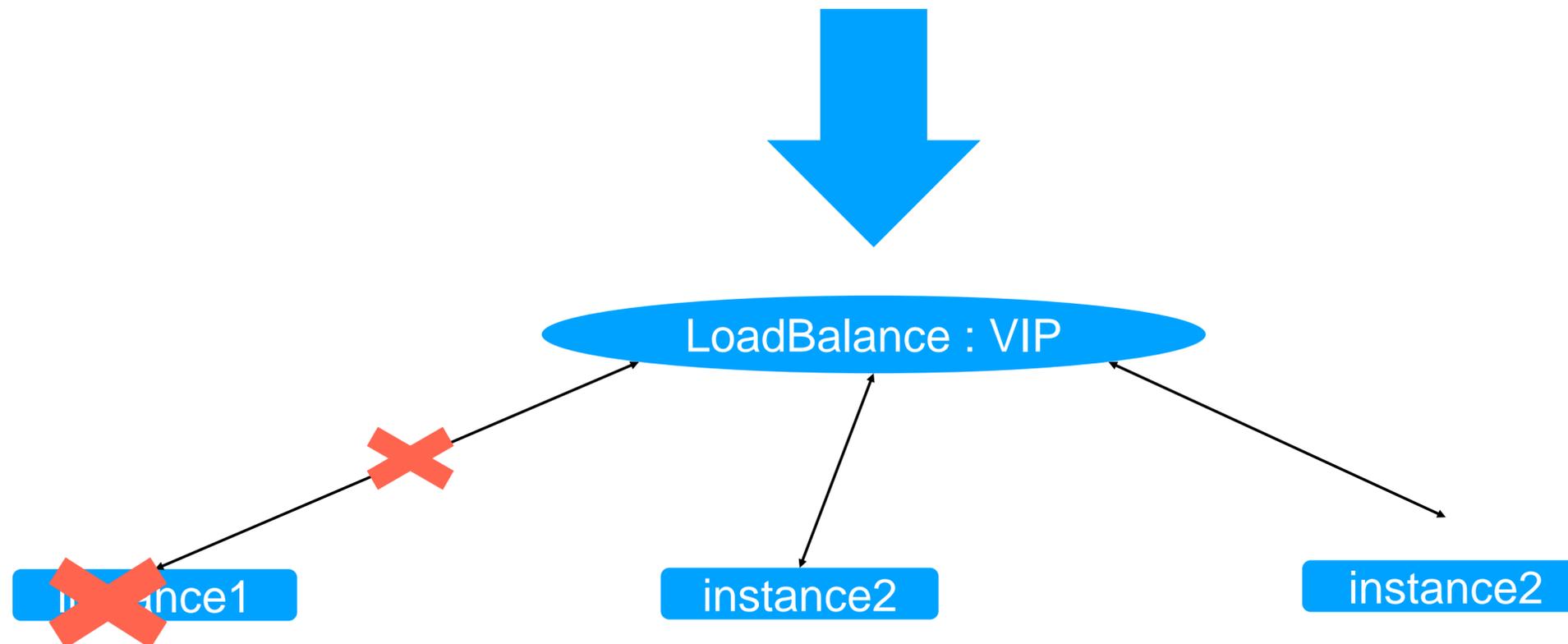


```
Core was generated by './larbin'.
Program terminated with signal 11, segmentation fault.
#0  newId (this=0xb7adae9c) at site.cc:366
366      strcpy(name, u->getHost());
(gdb) where
#0  newId (this=0xb7adae9c) at site.cc:366
#1  NamedSite::newQuery (this=0xb7adae9c) at site.cc:182
#2  0x08053bbd in fetchDns () at fetchOpen.cc:55
#3  0x08056b1b in main (argc=Cannot access memory at address 0x19
) at main.cc:101
(gdb)
```



无状态单实例故障

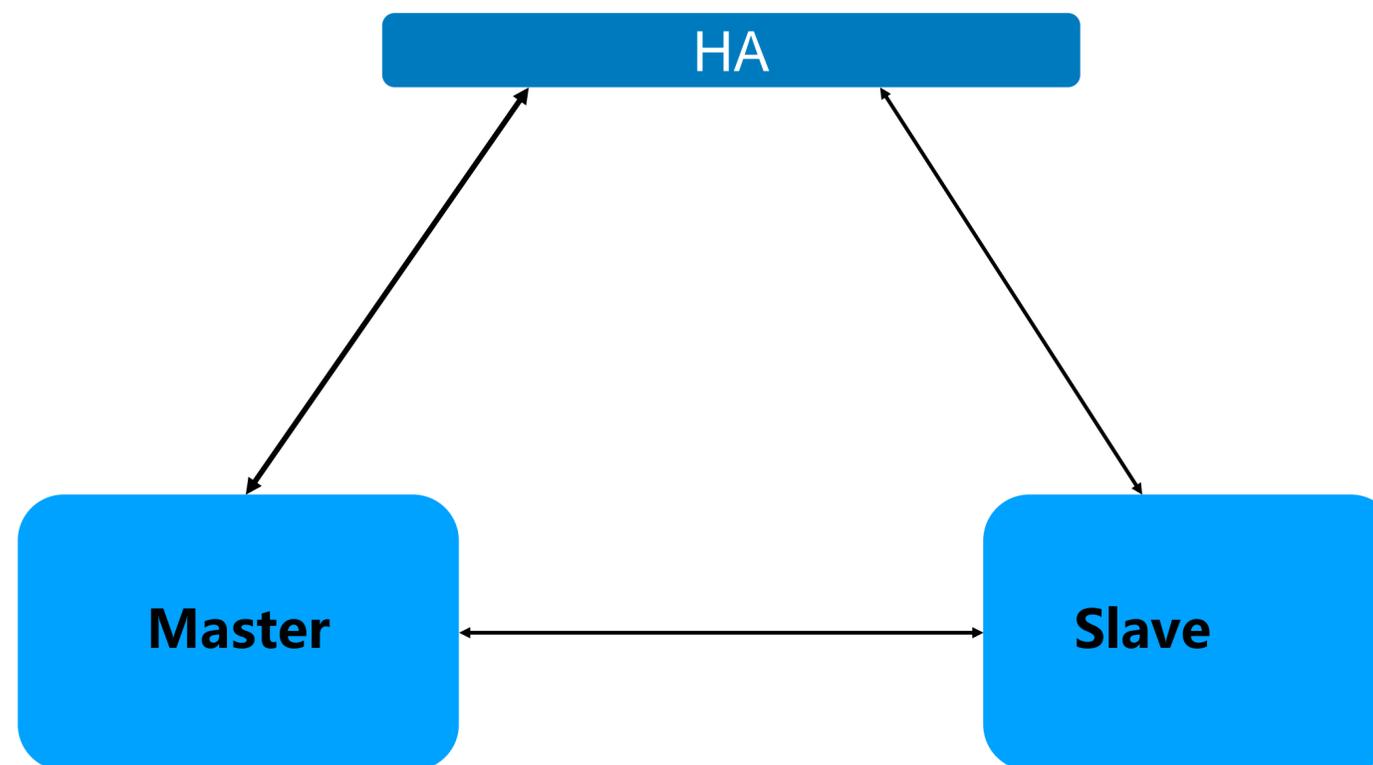
健康心跳
负载均衡
快速失败
超时时间
控制重试
自动恢复



有状态单实例故障

◆ 第三方仲裁者HA
数据库主备切换

◆ 自动选举
Paxos
Raft





服务集群故障

原因：

容量不足

变更引起

网络故障

程序BUG

...

解决办法：

快速弹性扩缩容、限流、熔断&降级

CI/CD DevOps

多机房多AZ（单元化部署）

Auto Test/ Code Review

...

服务集群故障-自我保护

限流：
按系统负载限流
按业务优先限流

降级：
依赖模块的降级
业务功能的降级

税费 预估 ¥ 42.48，实际税费请以提交订单时为准 税费收取规则 ▾
 运费 至 浙江/杭州市/上城区 ▾ 满88元包邮 ▾
 服务 本商品由 自营保税仓 发货
 数量 1 库存充足
 说明 ① 不支持7天无理由退货

立即购买

加入购物车

收藏

★★★★★ 99% 11148人评价 459人晒单

税费 预估 ¥ 42.48，实际税费请以提交订单时为准 税费收取规则 ▾
 运费 至 浙江/杭州市/上城区 ▾ 满88元包邮 ▾
 服务 本商品由 自营保税仓 发货
 数量 1 库存充足
 说明 ① 不支持7天无理由退货

立即购买

加入购物车

收藏

★★★★★ 99%



服务集群故障-变更

NDP 自动部署平台

我的产品 我的收藏

按应用搜索

NDP

ndp-solution

集群

搜索您的集群



帮助



个人设置

曾宇星

退出

ndp-solution > solution_test > 集群

集群操作

服务器管理

构建历史

发布历史

构建

发布

一键发布

回滚

重启

停止

上线

下线

构建状态: 构建成功

最新版本: master-708229c643073c68880e09e8132902bfcbc...

发布状态: 发布完成

<input type="checkbox"/>	主机	实例	分支	版本号	发布时间	发布状态	应用状态	上线状态
<input type="checkbox"/>	1 hzbdg-hzcloudsolution02.vpc.163.org	default	master	a3a1b15f90c4fe34f2...	2018-04-03 15:26:13	✓	运行	未设定
<input type="checkbox"/>	2 hzbdg-hzcloudsolution01.vpc.163.org	default	master	a3a1b15f90c4fe34f2...	2018-03-02 11:19:17	✓	运行	未设定
<input type="checkbox"/>	3 hzbdg-hzcloudsolution03.vpc.163.org	default	master	a3a1b15f90c4fe34f2...	2018-03-02 11:19:51	✓	运行	未设定
<input type="checkbox"/>	4 hzbdg-hzcloudsolution04.vpc.163.org	default	master	a3a1b15f90c4fe34f2...	2018-03-02 11:20:30	✓	运行	未设定



机房层面故障

网络核心故障
接入层故障
机房断电

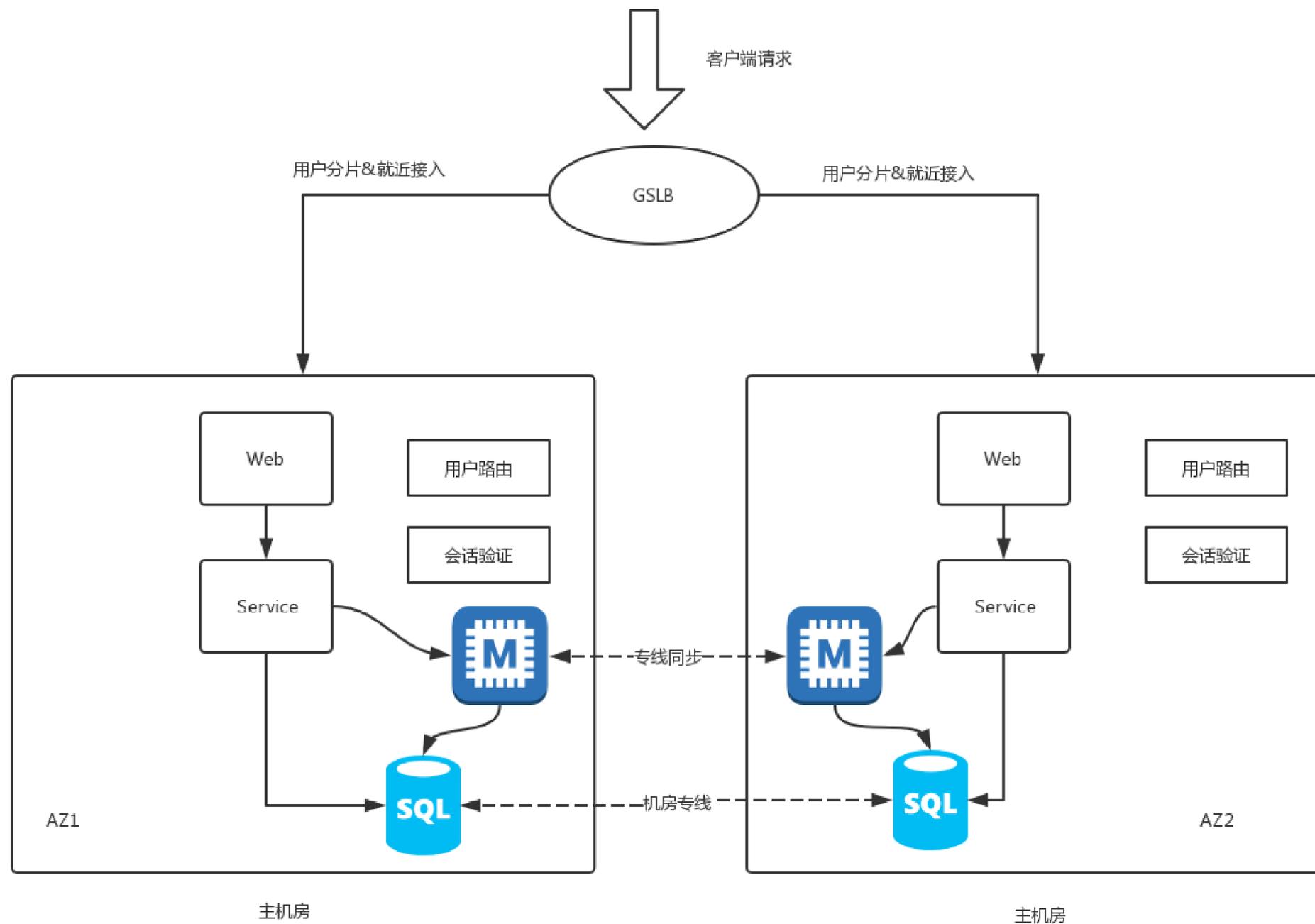
。 。 。



以不变应万变-----多机房/可用区容灾

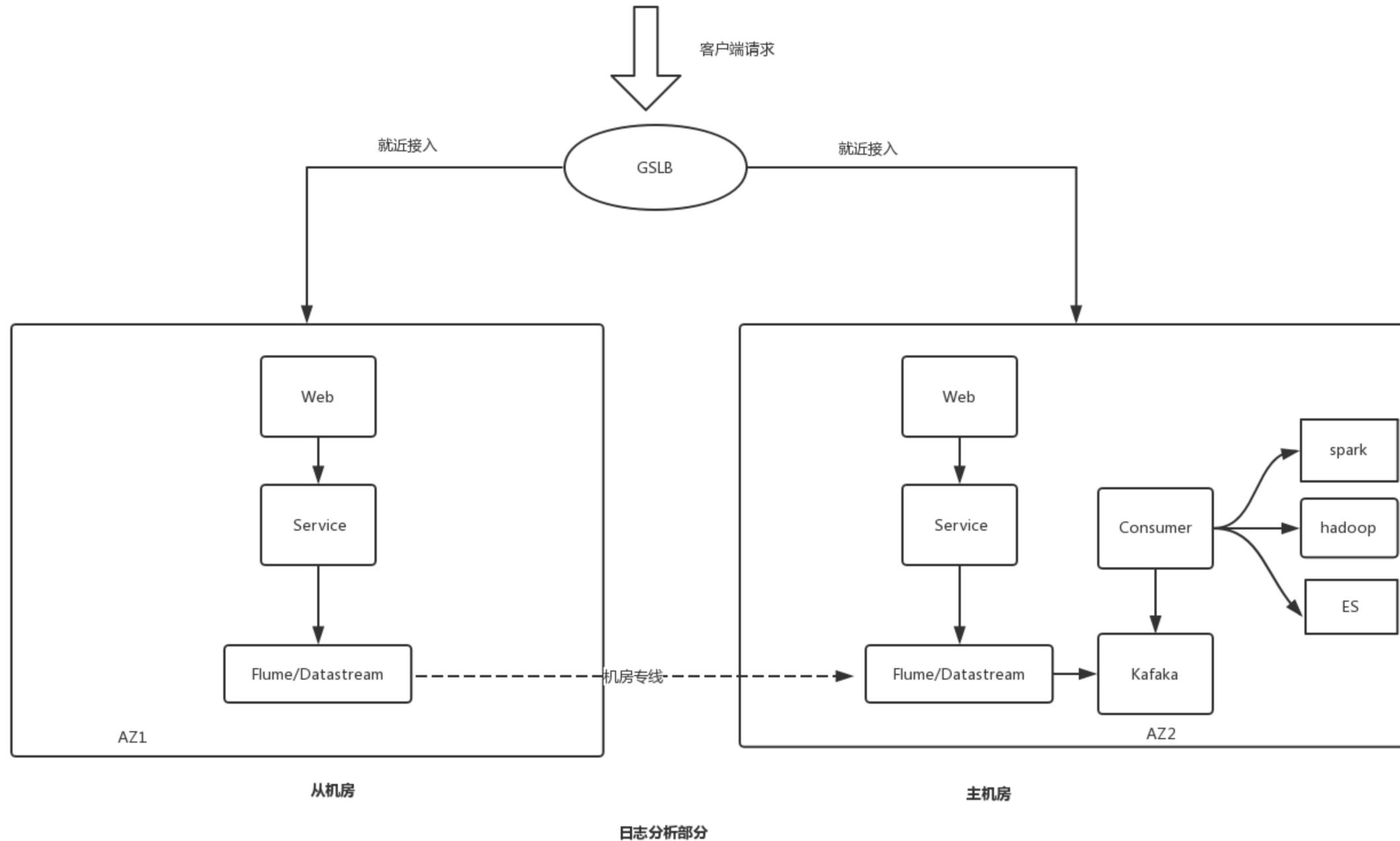


同城-多可用区部署架构-单元化、多活



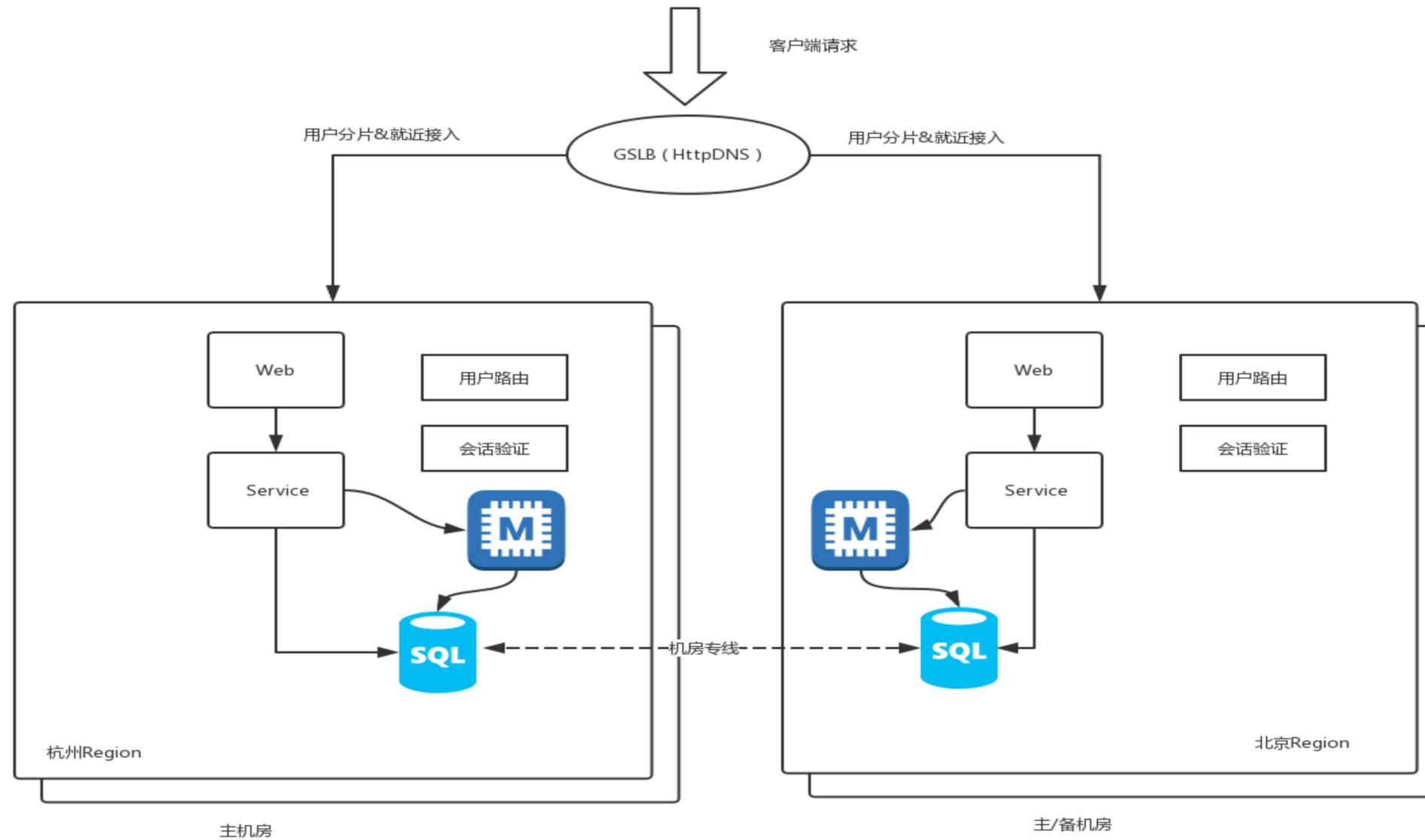


单元化-主从模式





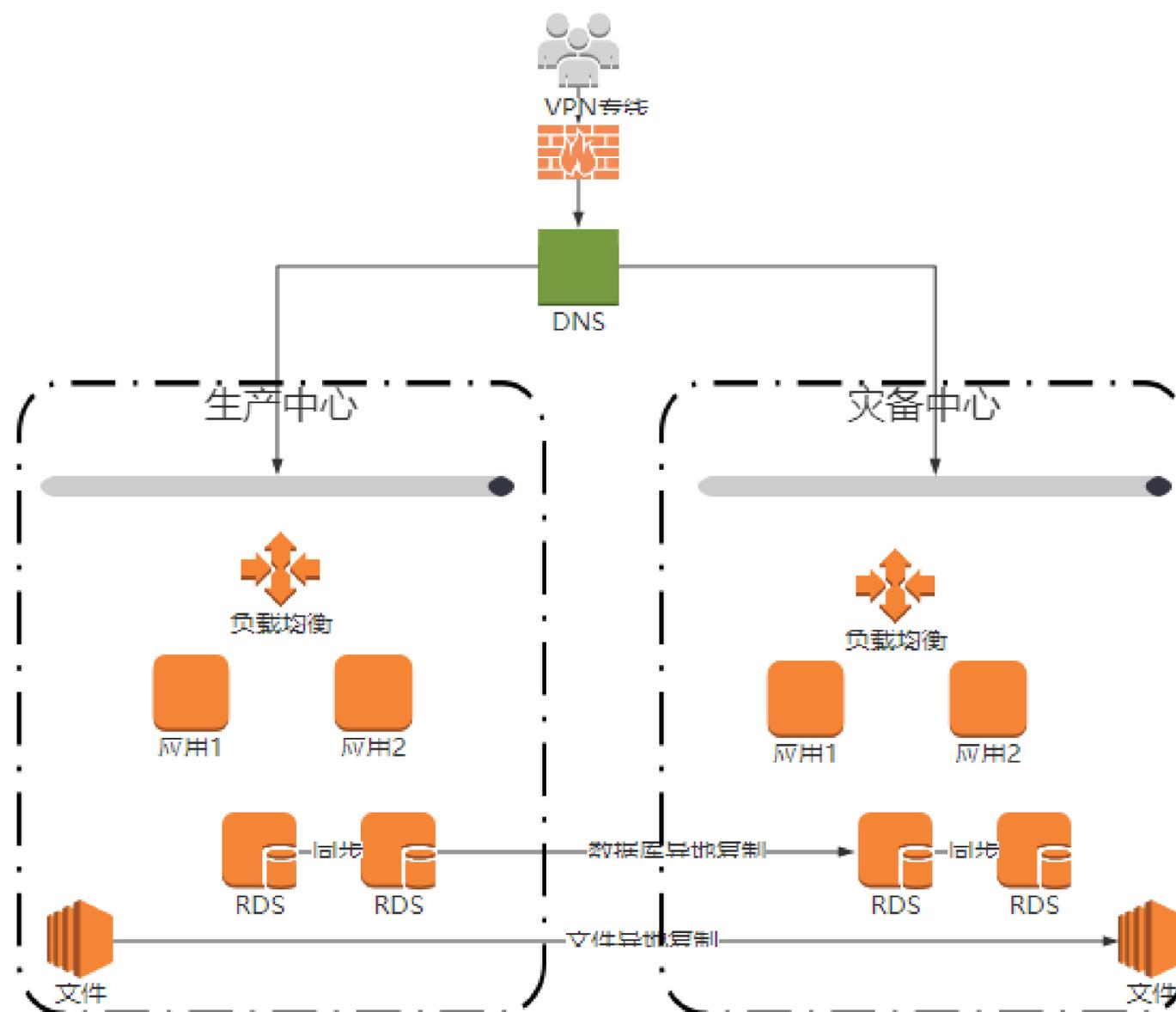
面向跨区域的设计-多活



跨Region应用是否多活-取决于数据延迟对业务的影响



跨Region-异地容灾





可用性

- Vm挂掉应被当做是 **【常态】**
- 针对不同的云服务，不同的容灾策略
 - 大原则：切、切、切
 - VM挂掉的处理
 - LB 挂掉的处理
 - 机房核心网络异常的处理

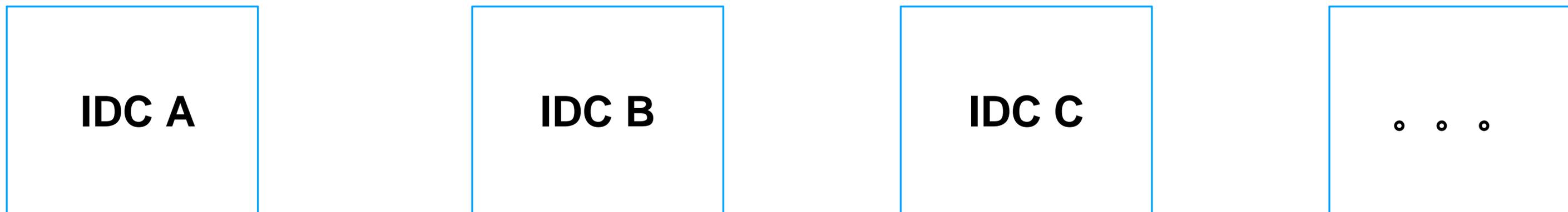
可用性



自动流量切换，业务几乎无影响



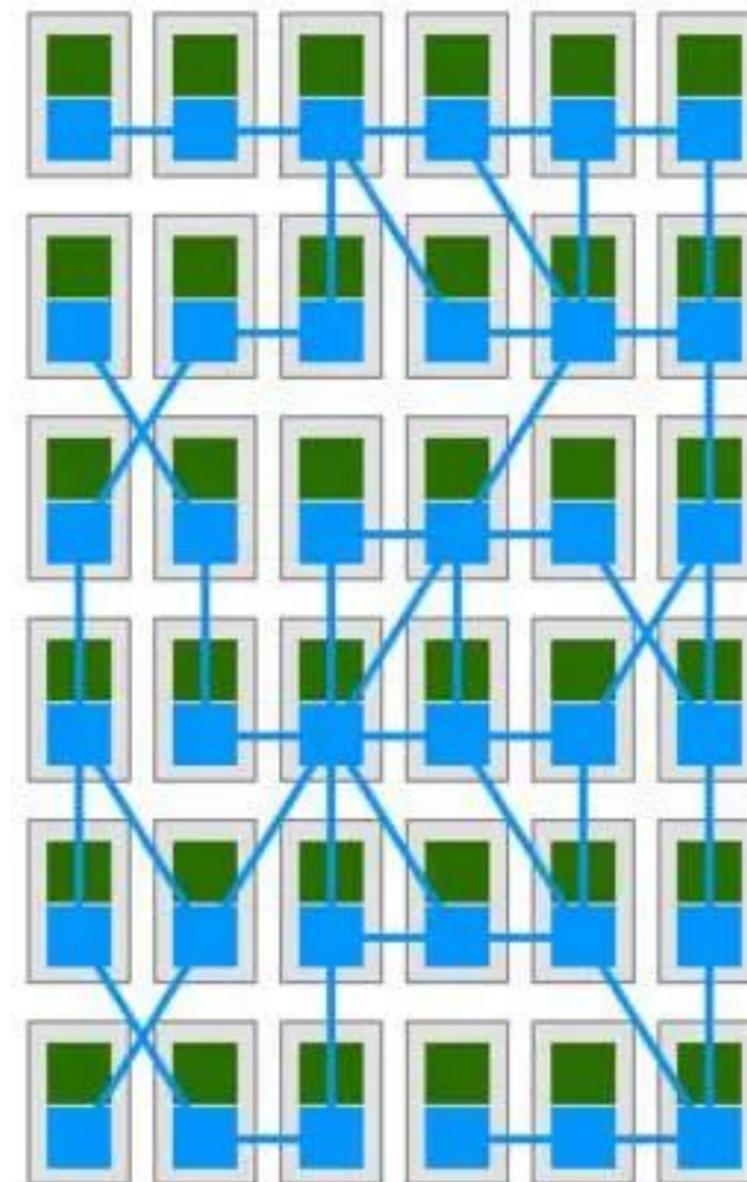
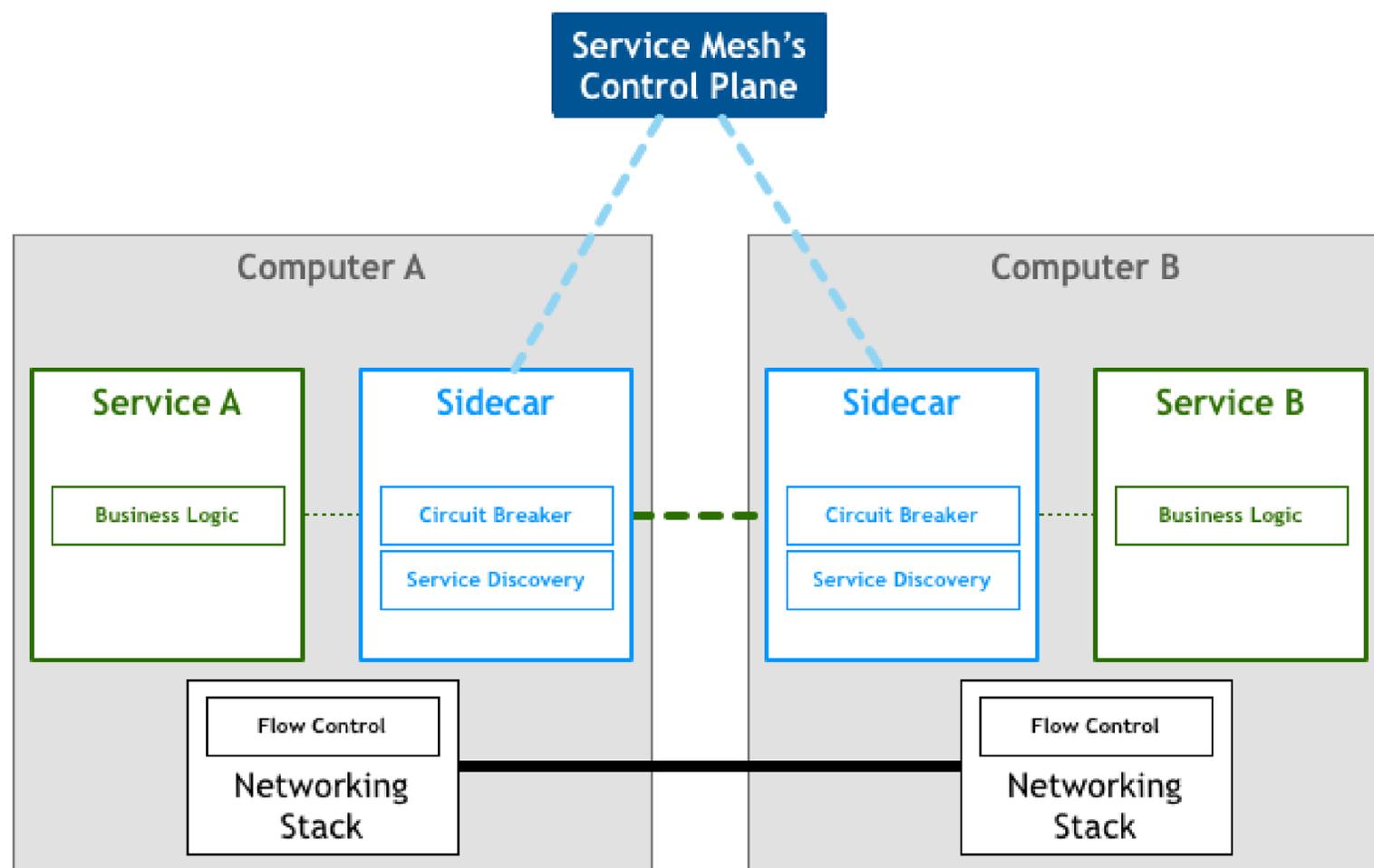
系统扩容



系统能够任意水平扩容

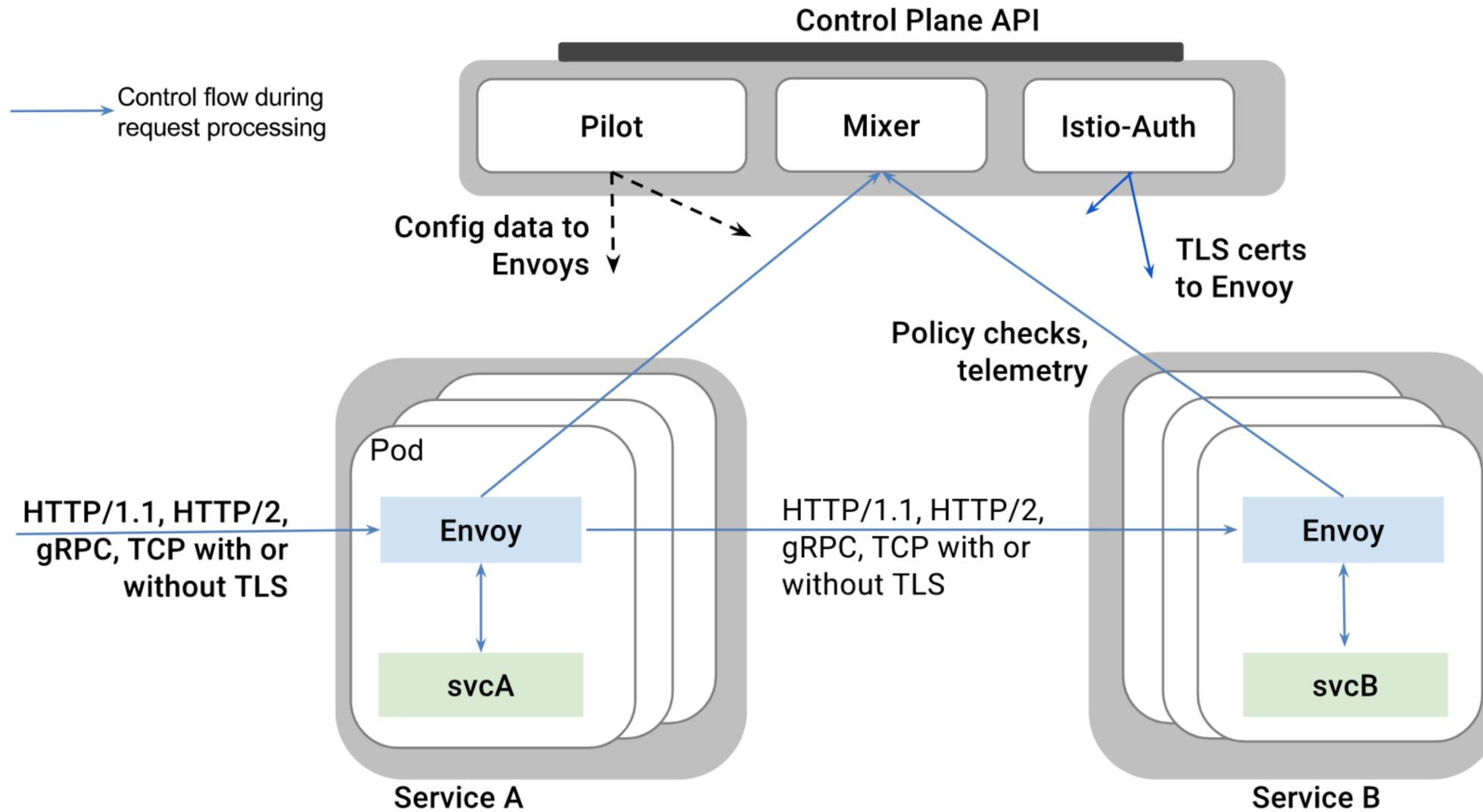


架构演进方向- Cloud Native (Service Mesh)

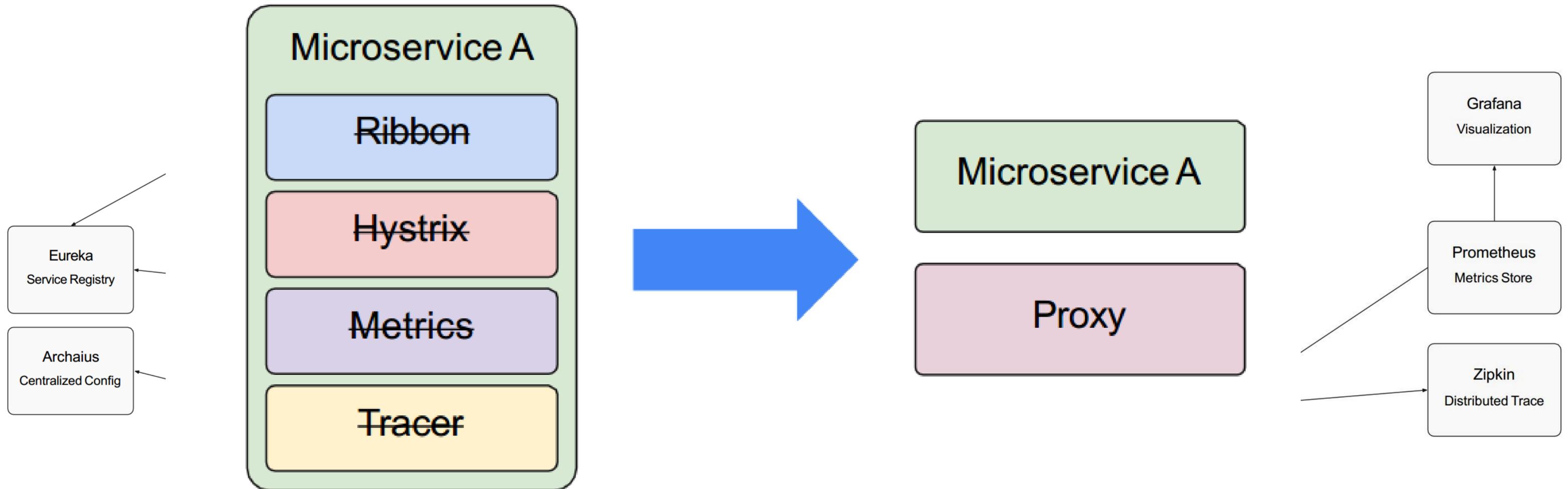




Service Mesh -- Istio



Service Mesh – envoy proxy & nginxMesh





网易云基础服务--总体架构全部基于开源平台

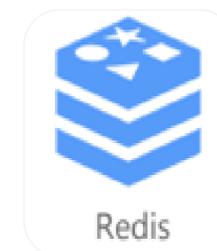
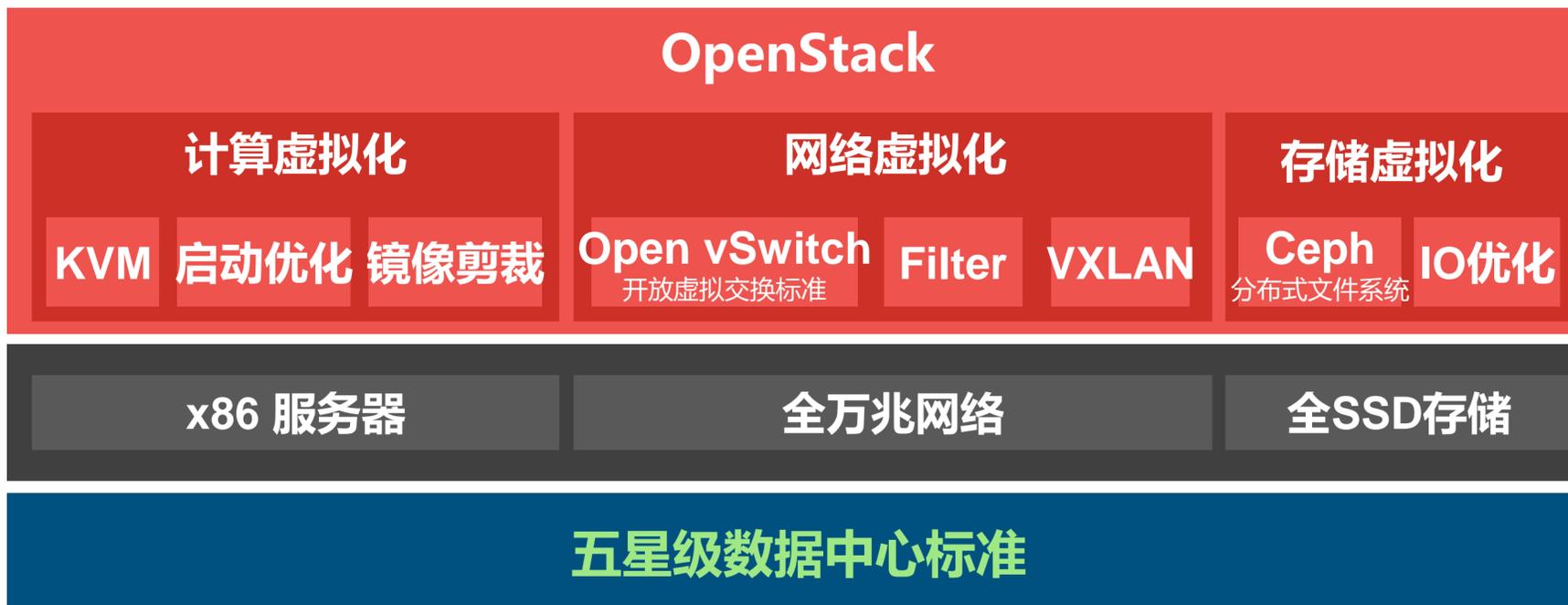
CaaS



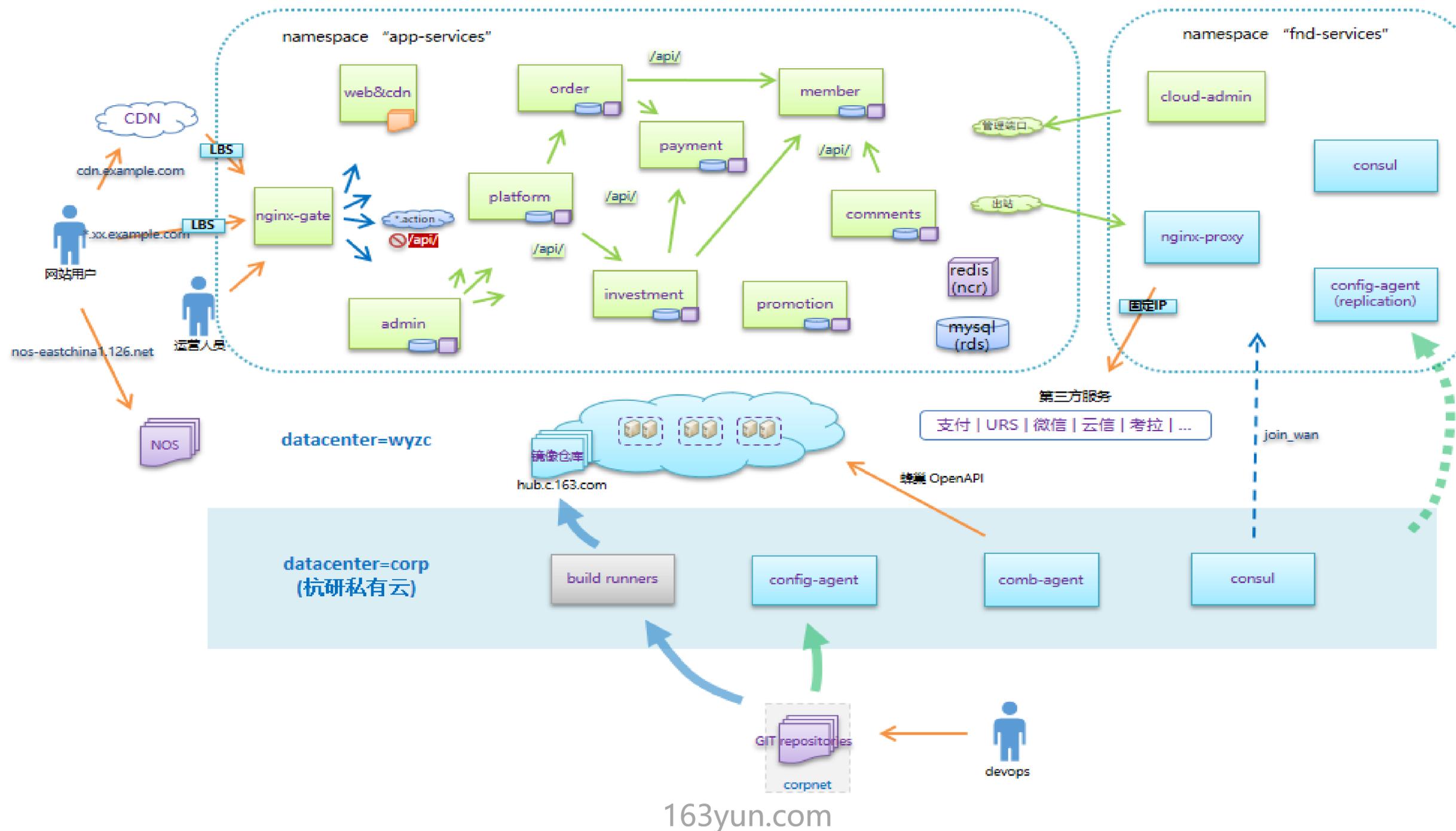
PaaS



IaaS



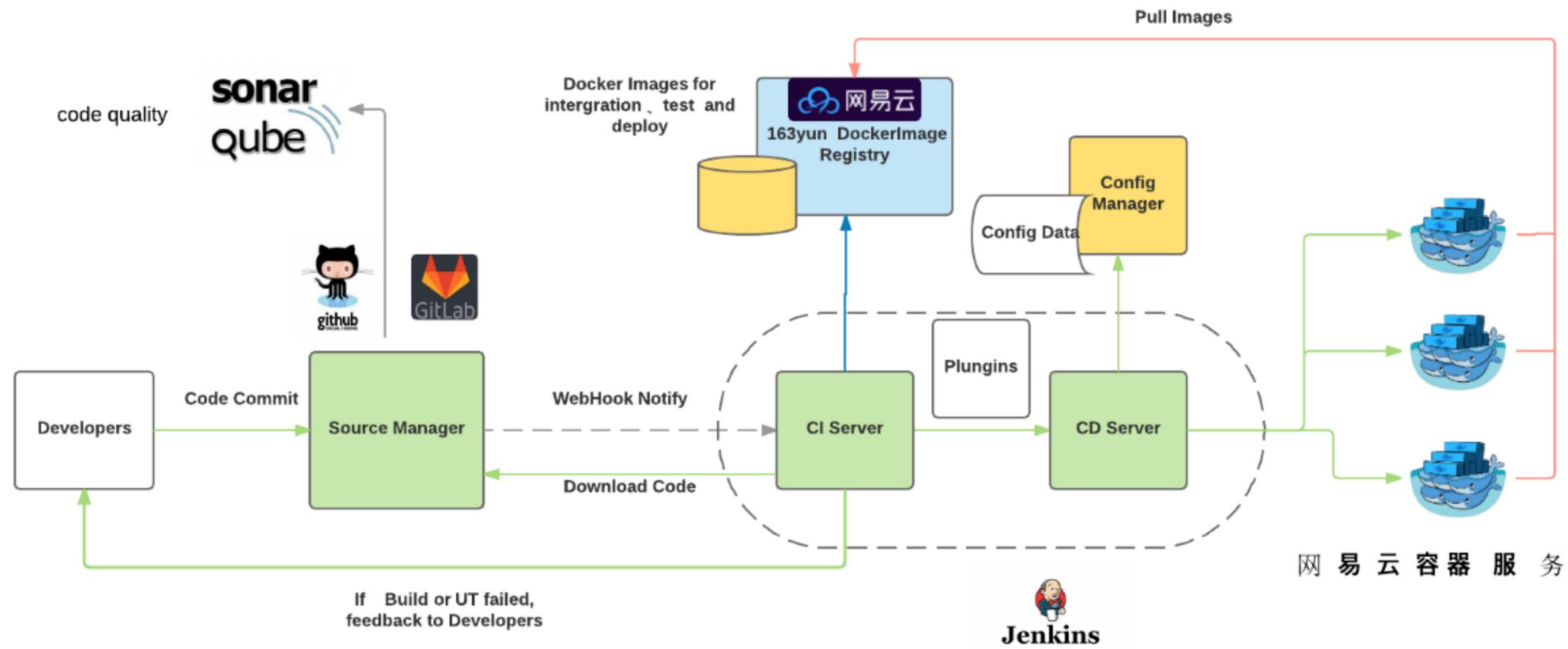
架构实践-云端微服务(Cloud Native)





持续交付-网易云最佳实践

163YUN-CICD





架构怎么快速落地？

如果你是网易云客户？

云计算不仅是提供
计算、存储、网络

还有Paas 和
SaaS

但光有这些还不够，还需要好的架构-> **架构师**



网易云

共创云上精彩世界