



GOPS 2016
Shenzhen



全球运维大会

2016

深圳站

会议时间：3月25日-3月26日

会议地点：深圳·南山区 圣淘沙酒店(翡翠店)

主办单位： 开放运维联盟
OOPSA Open OPS Alliance  高效运维社区
GreatOPS Community

指导单位： 数据中心联盟
Data Center Alliance

协办单位：中国新一代IT产业推进联盟





GOPS 2016
Shenzhen



全球运维大会 2016


深圳站

阿里游戏高可用架构设计实践

李运华 @阿里游戏



目录

- 
- 1 总体架构
 - 2 HTTP-DNS
 - 3 架构解耦
 - 4 业务降级
 - 5 异地多活
 - 6 360°监控



背景 - 疲于奔命的运维

机柜断电

服务器宕机

交换机宕机

程序bug



1个月4次大故障，运维背了好大一口锅！！！！



分析 - 运维太low ?

硬件太烂?

运气不好?

测试不足?

运维经验不足?

流程不完善?



根因分析：技术太弱

根本策略：把运维的锅，让研发去背



高可用目标 – 传统方法

Unavailability Durations Based on Availability Percentages

Availability	Unavailability Per Year
98%	7.3 days
99%	3.65 days
99.8%	17 hrs 31 min
99.9%	8 hrs 45 min
99.99%	52.5 min
99.999%	5.25 min
99.9999%	31.5 sec

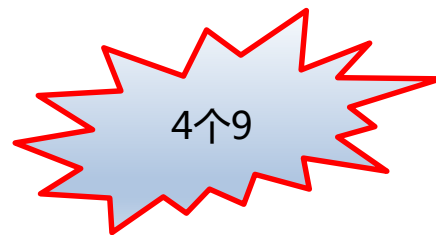
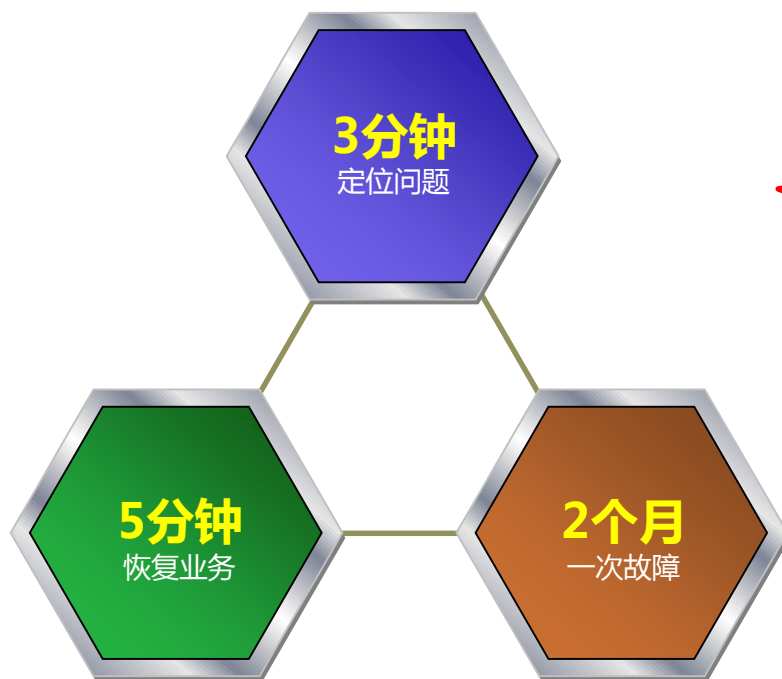
俗称“几个9”

优点：业界通用，方便对比

缺点：无法直观的理解



高可用目标 – 面向业务



优点：1) 聚焦于业务 2) 容易分解 3) 容易衡量



高可用总体架构



- 1) 面向业务
- 2) 立体化

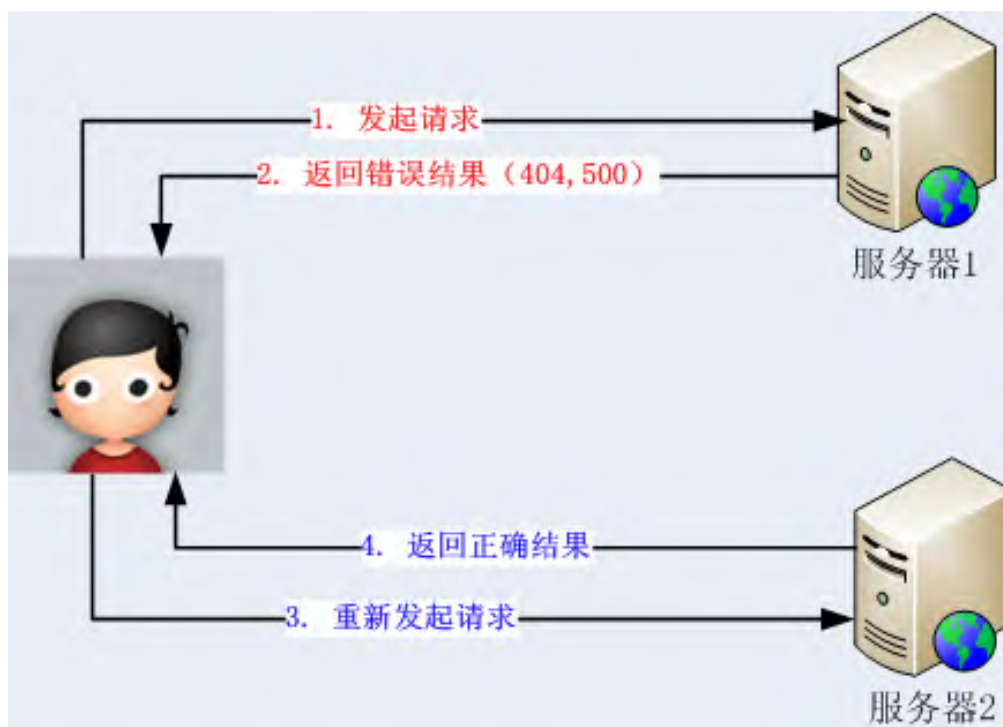


目录

- 1 总体架构
- 2 HTTP-DNS
- 3 架构解耦
- 4 业务降级
- 5 异地多活
- 6 360°监控



客户端重试



让看得见炮火的前线快速做决策

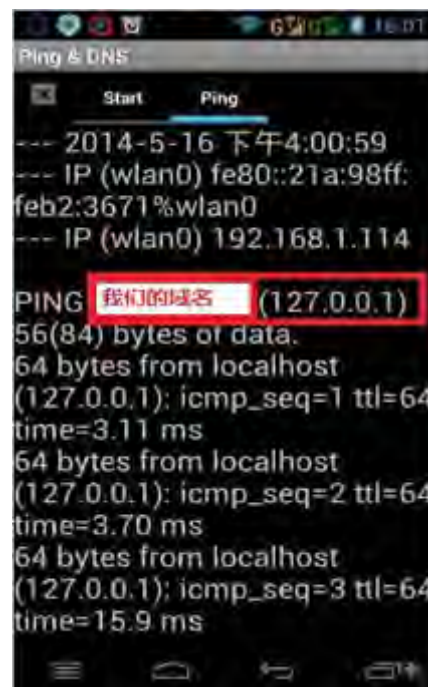


传统DNS问题

DNS劫持

DNS污染

DNS缓存



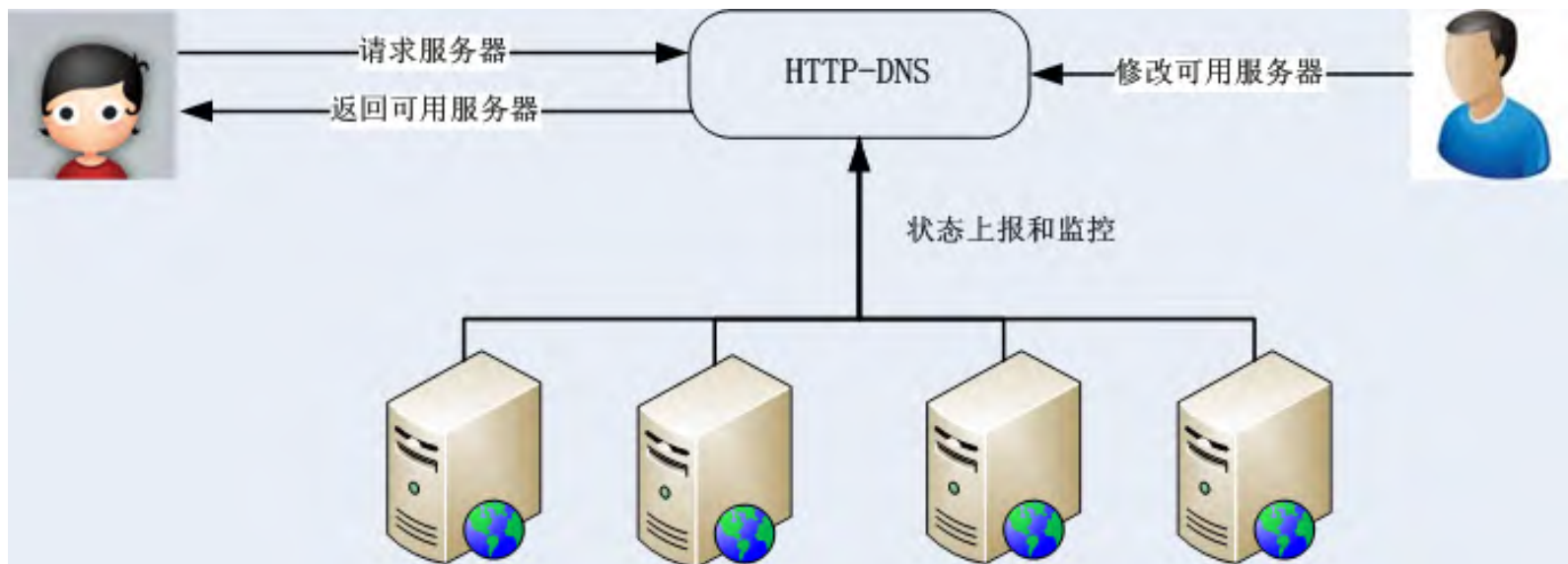
```
Ping & DNS
Start Ping
--- 2014-5-16 下午4:00:59
--- IP (wlan0) fe80::21a:98ff:
feb2:3671%wlan0
--- IP (wlan0) 192.168.1.114

PING 我们的域名 (127.0.0.1)
56(84) bytes of data.
64 bytes from localhost
(127.0.0.1): icmp_seq=1 ttl=64
time=3.11 ms
64 bytes from localhost
(127.0.0.1): icmp_seq=2 ttl=64
time=3.70 ms
64 bytes from localhost
(127.0.0.1): icmp_seq=3 ttl=64
time=15.9 ms
```

传统DNS无法满足要求



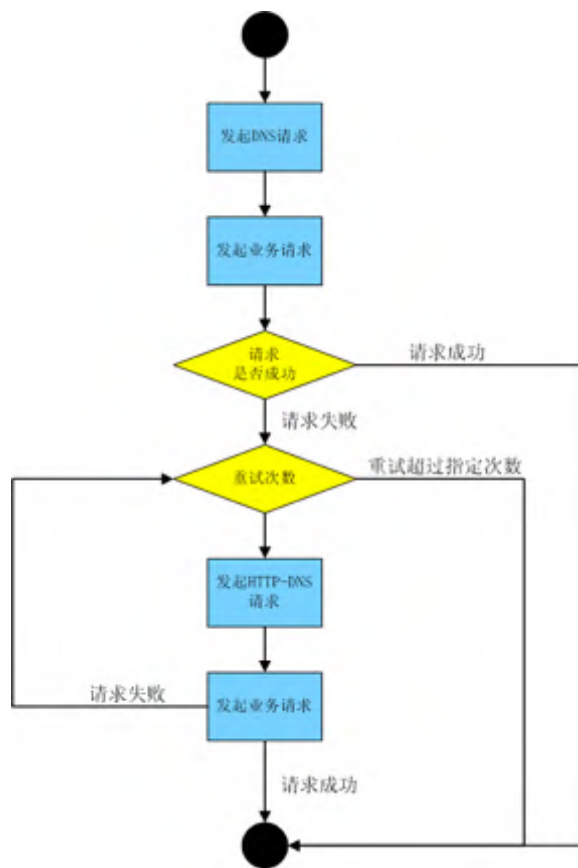
HTTP-DNS



优点：1) 灵活；2) 快速；3) 方便




客户端重试 + HTTP DNS



正常走传统DNS，异常走HTTP-DNS

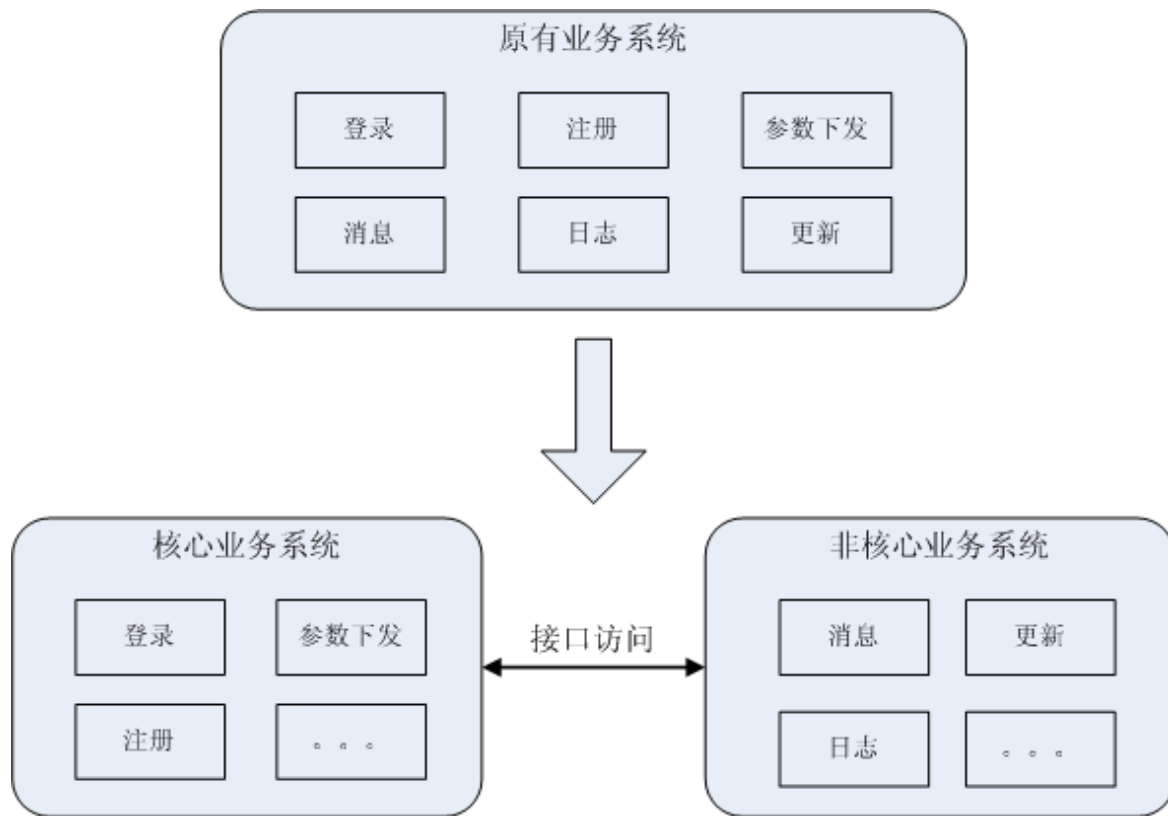


目录

- 1 总体架构
- 2 HTTP-DNS
-  3 架构解耦
- 4 业务降级
- 5 异地多活
- 6 360°监控



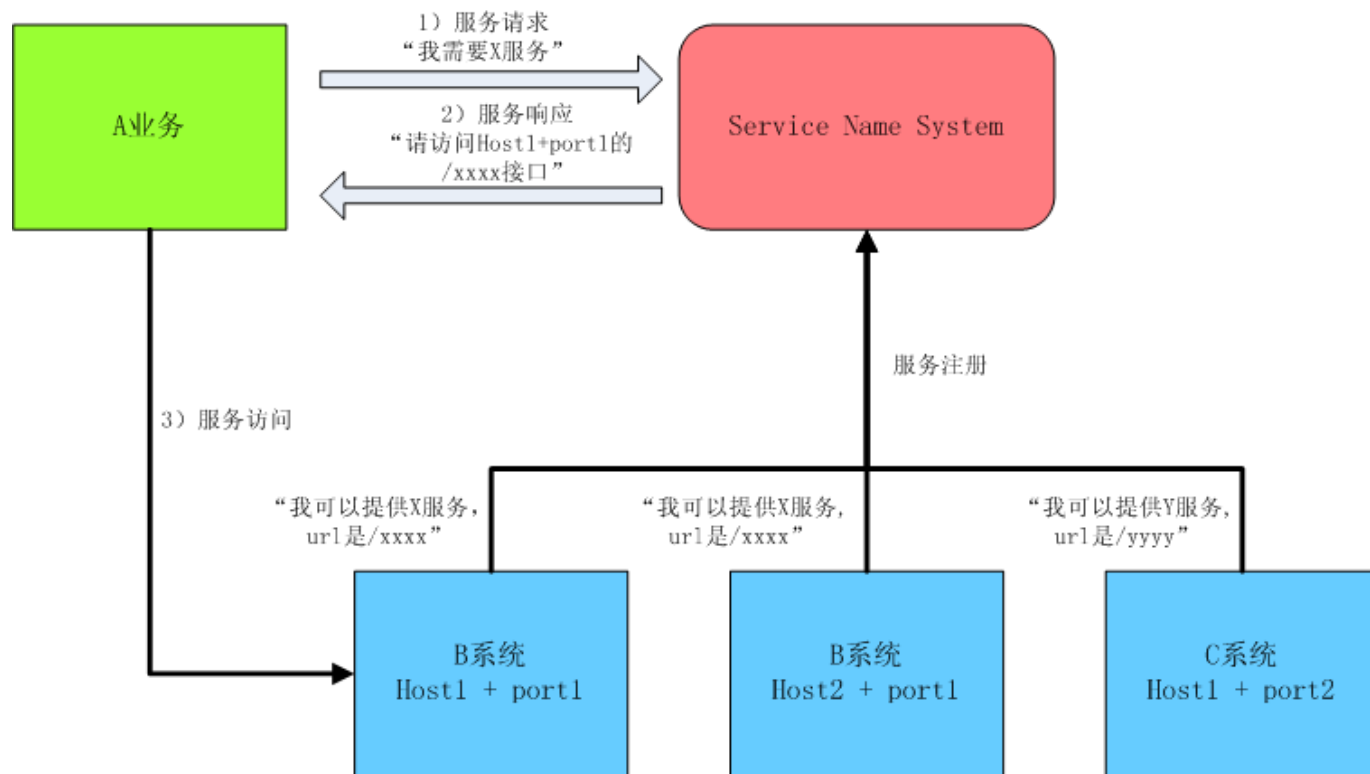
业务分离



根据业务优先级拆分系统，避免互相影响




服务中心



服务中心类似DNS，实现服务寻址和调度功能

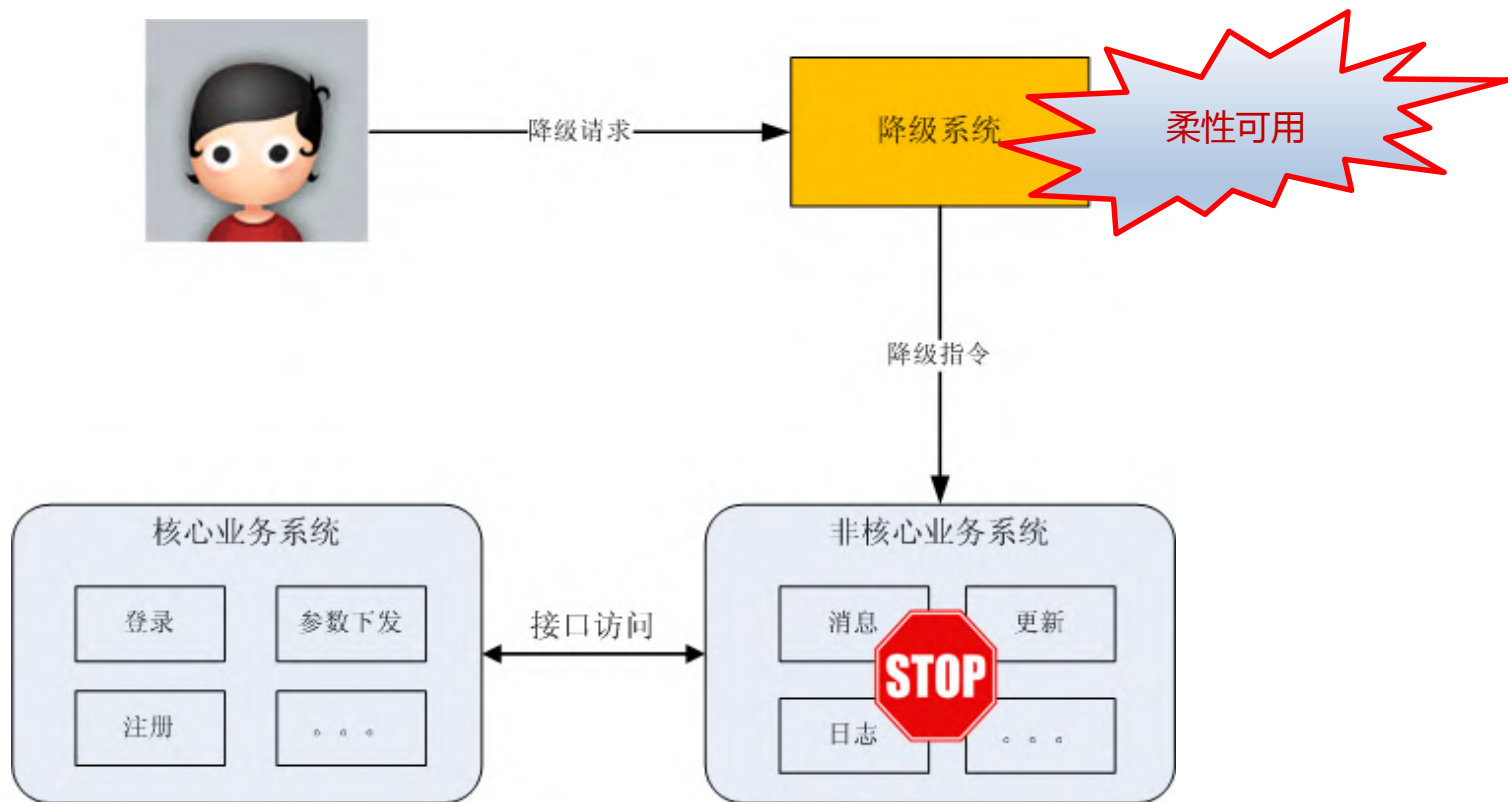


目录

- 1 总体架构
- 2 HTTP-DNS
- 3 架构解耦
-  4 **业务降级**
- 5 异地多活
- 6 360°监控



业务降级



丢卒保车



目录

1

总体架构

2

HTTP-DNS

3

架构解耦

4

业务降级



5

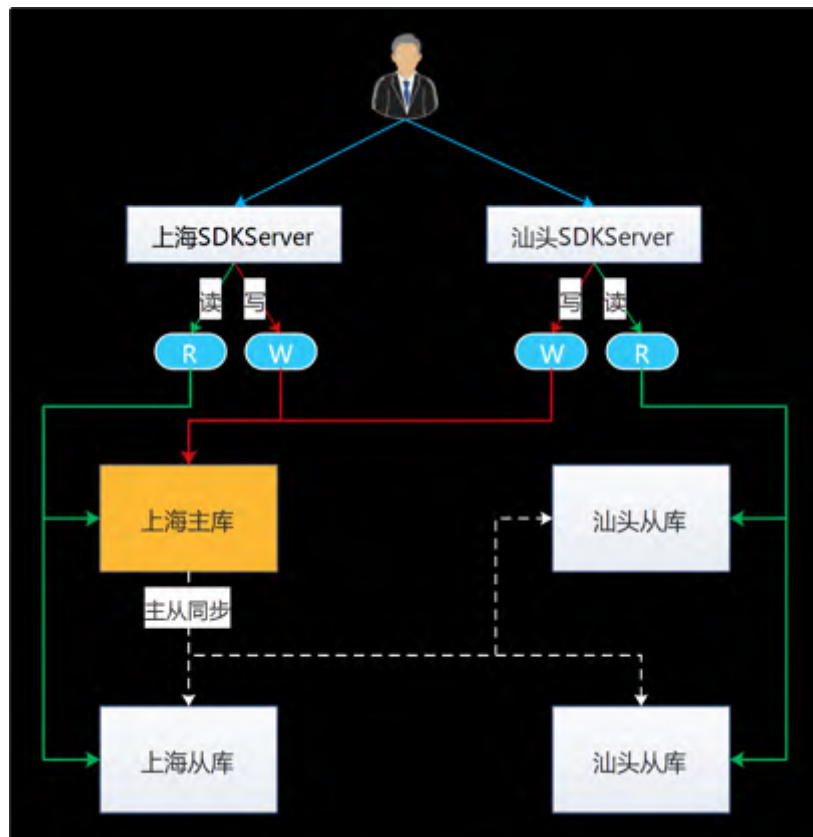
异地多活

6

360°监控



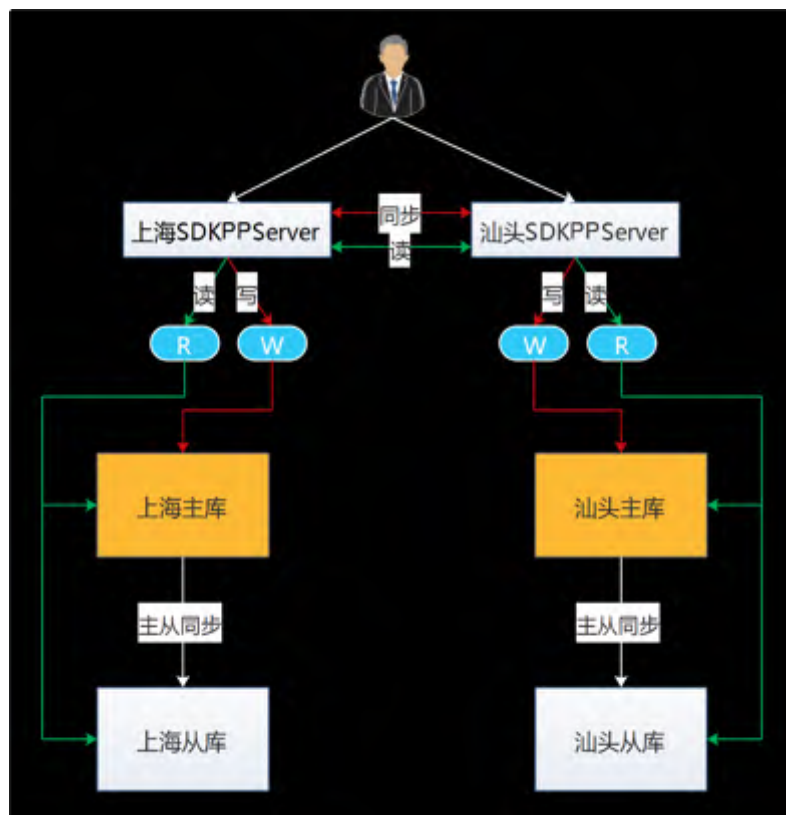
老架构



存在全局单点、跨机房同步时延问题



新架构



关键：1) 业务层数据同步；2) 二次读取；3) 可重复生成全局唯一数据



目录

1

总体架构

2

HTTP-DNS

3

架构解耦

4

业务降级

5

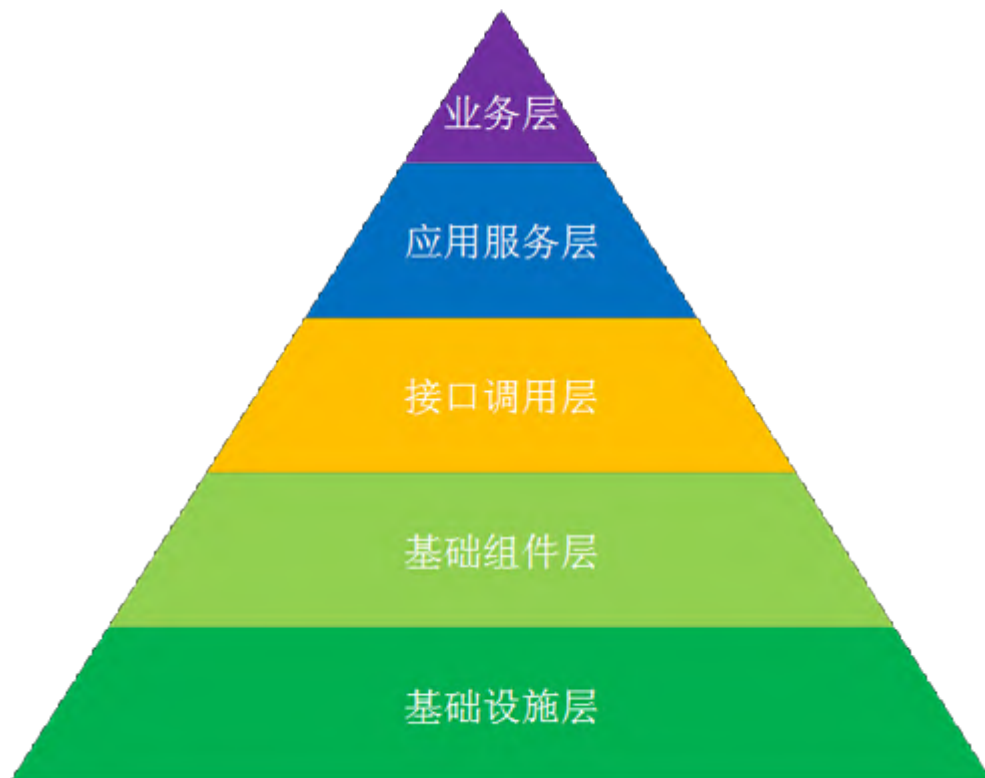
异地多活

6

360°监控



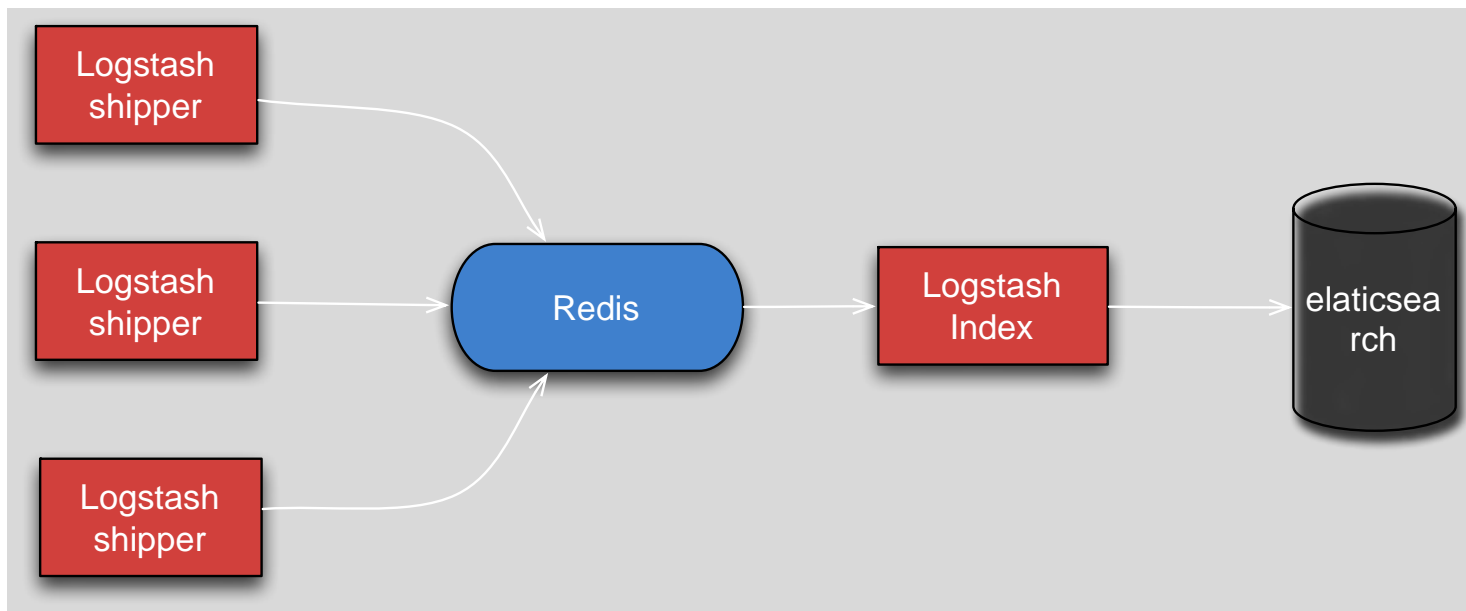
立体化



分层监控，不留死角



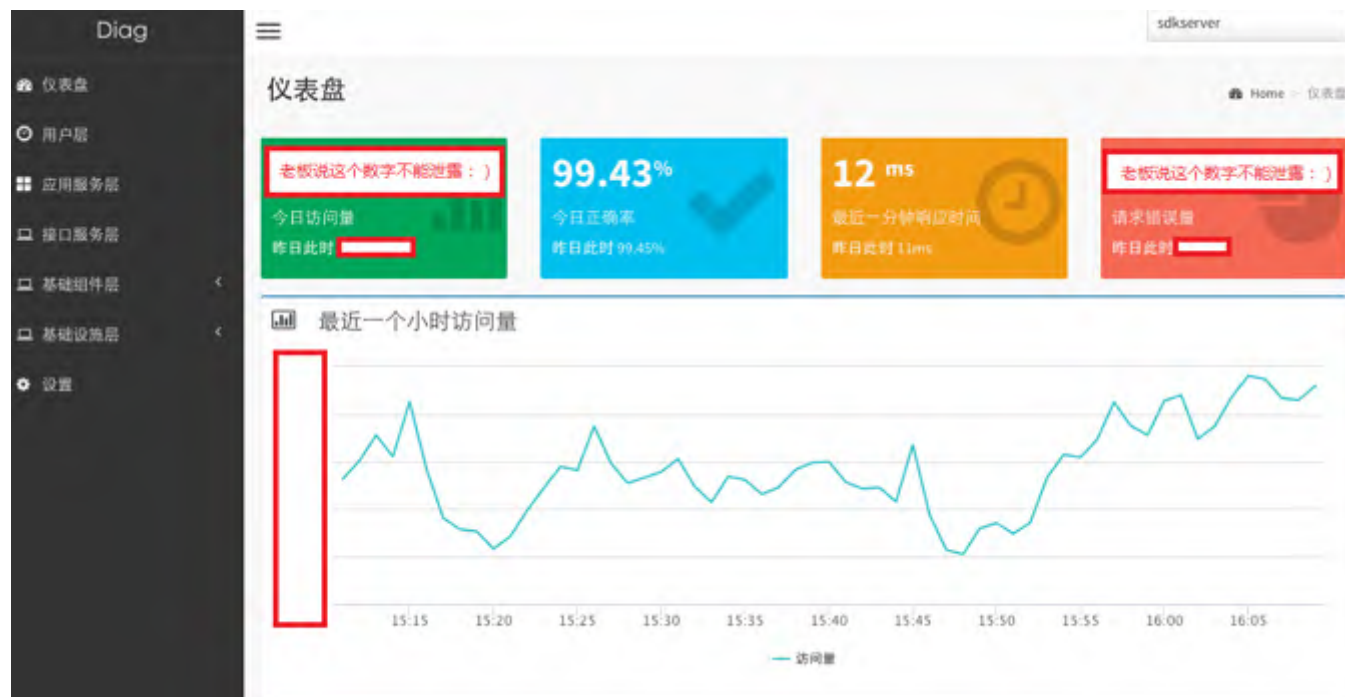
自动化



实时采集和分析，无需人工参与



可视化



一目了然，无需太多临时分析



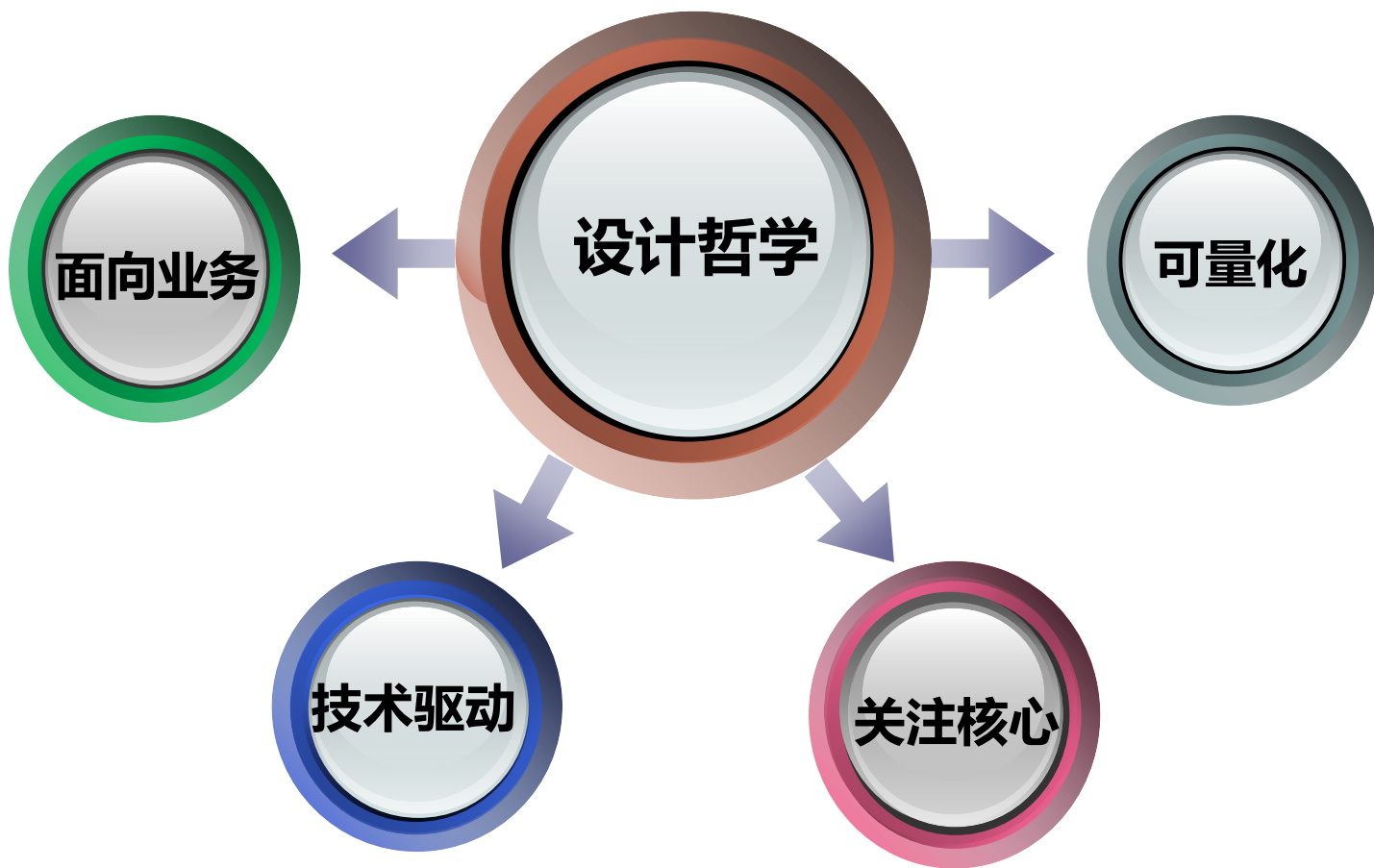
效果

现象	原因	处理
请求量每5分钟飙升5倍	APP日志每5分钟上报 (app.android.User.reportOnlineStatus)	APP修改定时机制，改为随机延迟上报
FGC时间过长：2.5s，导致系统有时候进入过载保护	并行垃圾回收器，中断时间较长	改为CMS，中断降为200ms以内 过载保护没有了，不再丢包
每天00:00机器过载	APP有个日志接口 (app.android.Log.statistics) 每天凌晨定时上报日志	APP修改为随机上报
支付宝错误响应飙升	支付宝断网演练	无需处理
新机器4倍流量，过载丢包很多	新机器硬件性能很强，一开始分配了4倍的流量，但实际上应用程序并不能完全利用机器资源，导致过载	修改为2倍流量，没有过载了

既能够发现大问题，也能够发现隐藏的问题



设计哲学



愿景



天下无贼



运维无锅



谢谢

