



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

BEIJING 2017

# 微博混合云架构应用实践之路

付稳 @it\_fuwen



促进软件开发领域知识与创新的传播



关注InfoQ官方信息  
及时获取QCon软件开发者  
大会演讲视频信息



扫码，获取限时优惠

**ArchSummit**  
全球架构师峰会 2017 [深圳站]

2017年7月7-8日 深圳·华侨城洲际酒店

咨询热线：010-89880682

**QCon**

全球软件开发大会 [上海站]

2017年10月19-21日

咨询热线：010-64738142

# 大纲

- 一、**峰值流量带来架构挑战**
- 二、**混合云DCP平台演进**
- 三、**混合云多机房架构演进**
- 四、**春晚峰值流量实战应对**





# Part 1

## 峰值流量带来架构挑战



# 微博业务现状

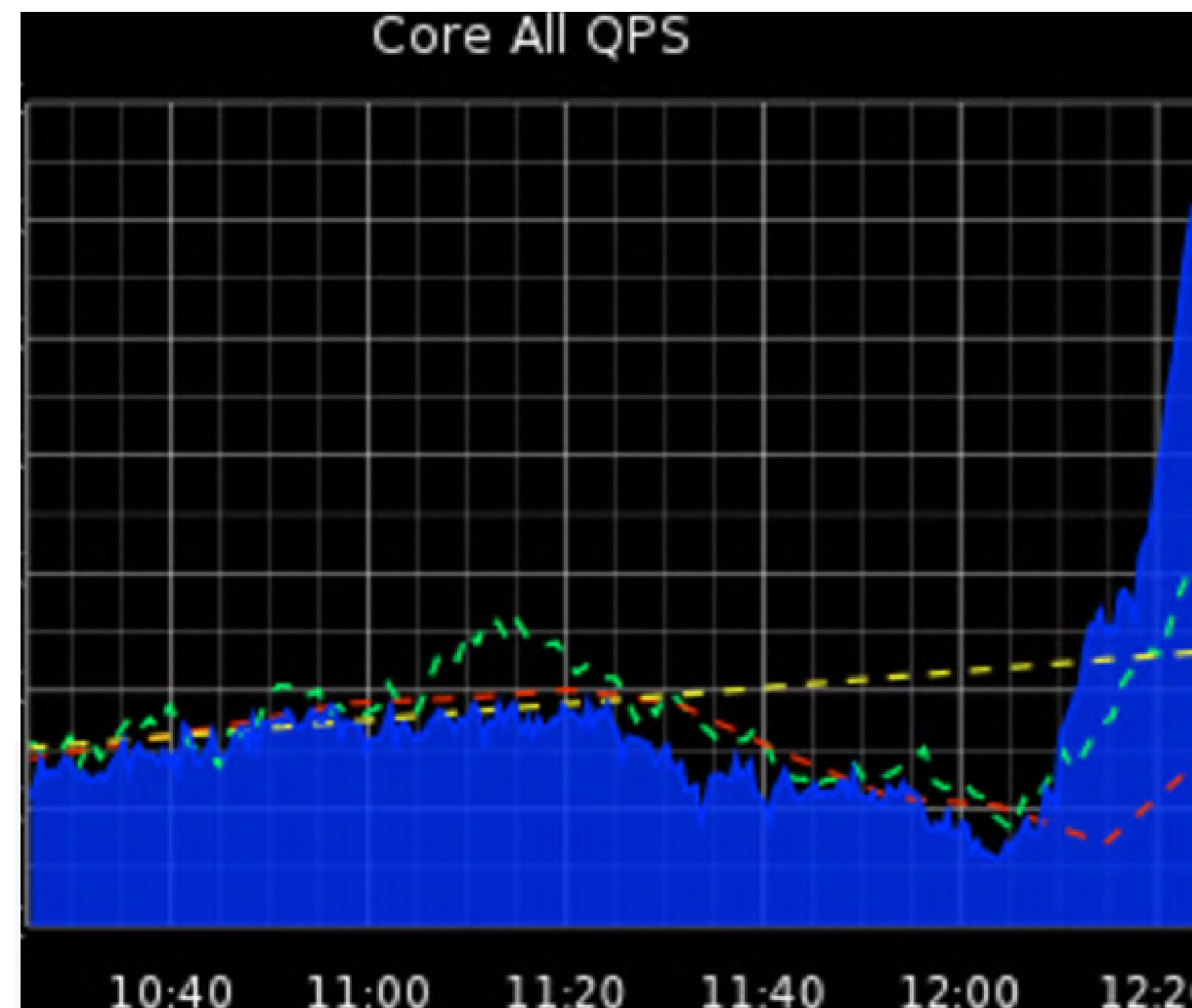
## •春晚峰值流量

- 机架位不足，上千台服务器库存不足
- 千万级采购成本巨大
- 采购周期长，运行三个月只为一晚

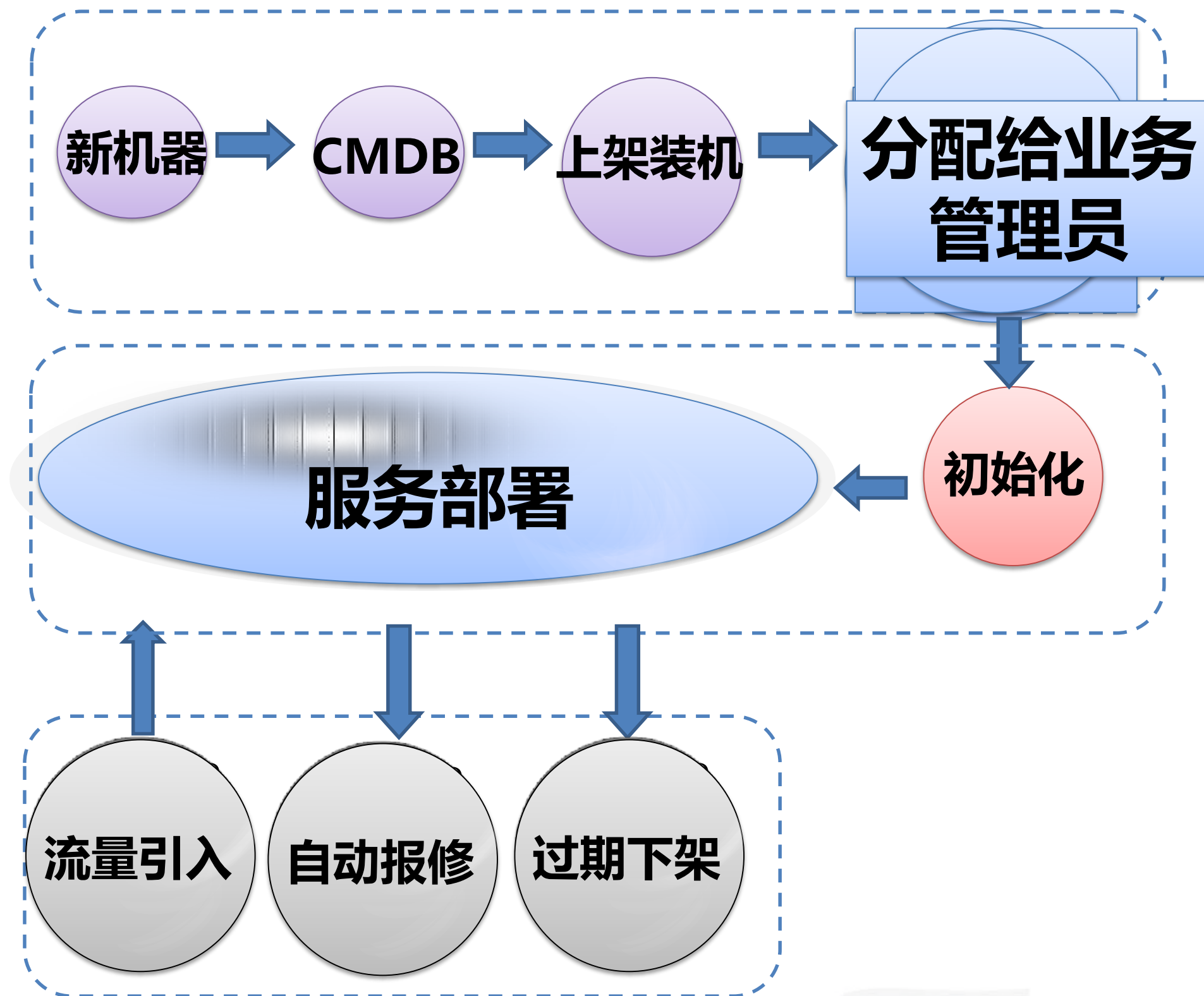
## •白百合、晨冰恋等娱乐热点突发事件

- 突发性强无预期、无准备
- 瞬时极端峰值
- 互动周期短

明星“出轨、在一起！”



# 峰值应对—传统手段



## 业务运维可控

### Step1 设备申请

- 设备申请，项目评审
- 入CMDDB，装机

### Step2 机器初始化

- 设备录入资源池，初始化

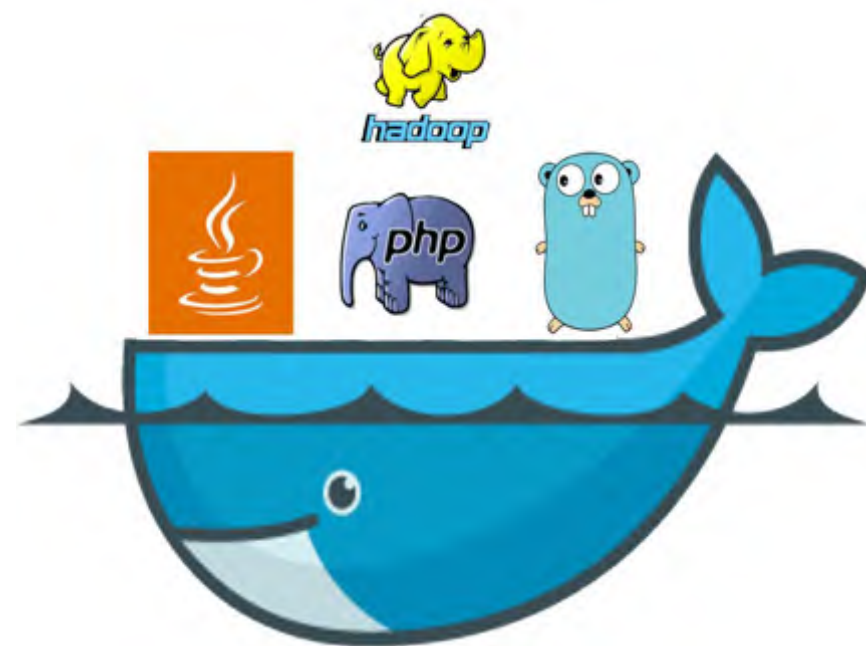
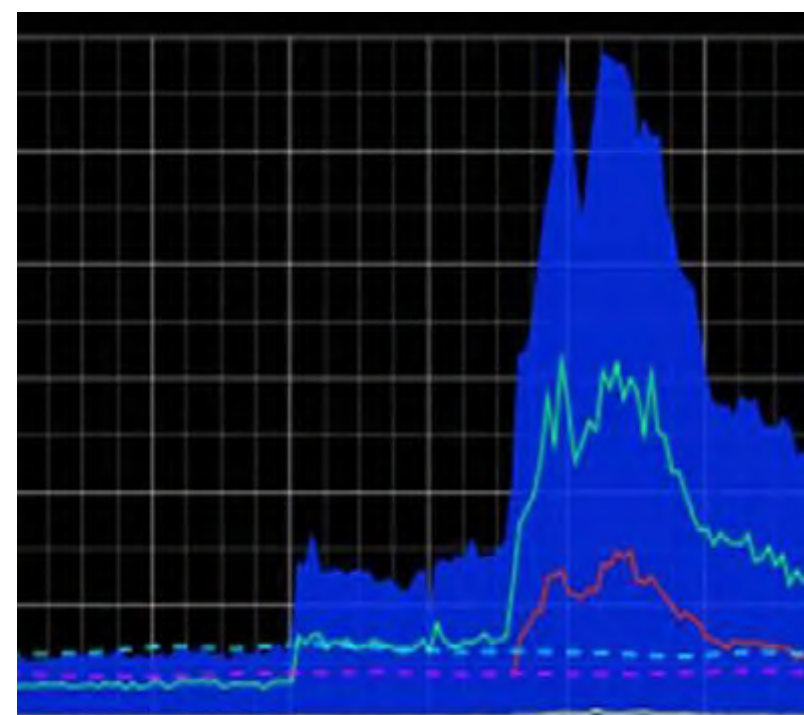
### Step3 服务部署

- 环境部署
- 监控部署
- 服务部署（代码 & confs）
- 流量引入

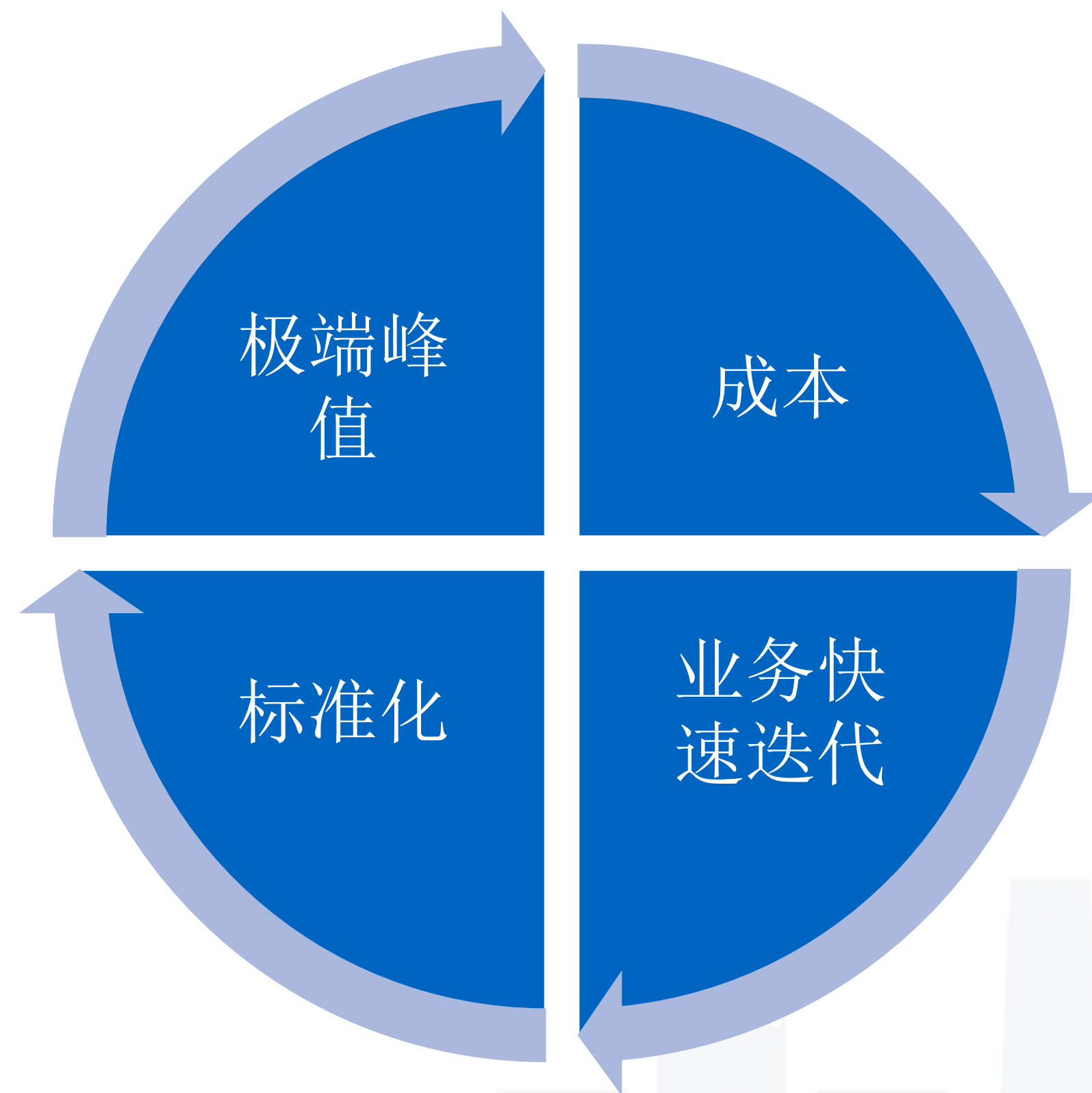
### Step4 自动报修&下架

- 服务自动上下线
- 机器置换或下架

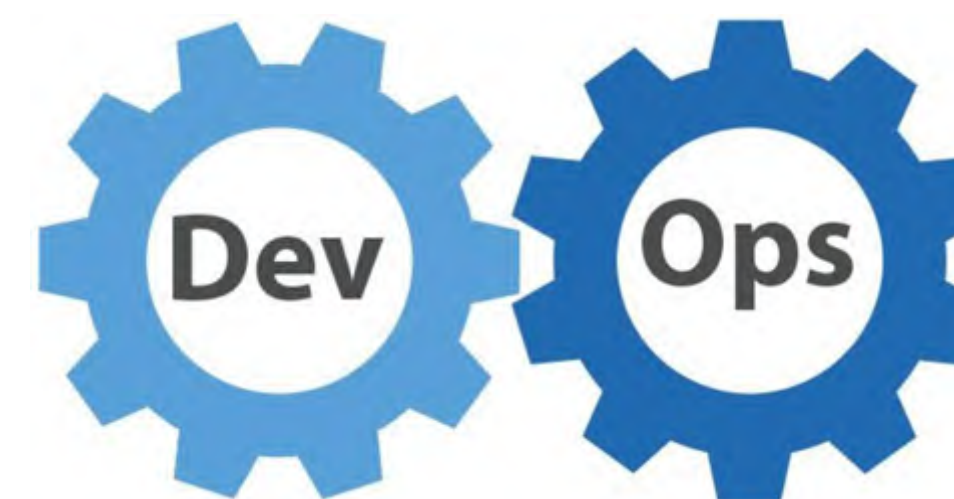
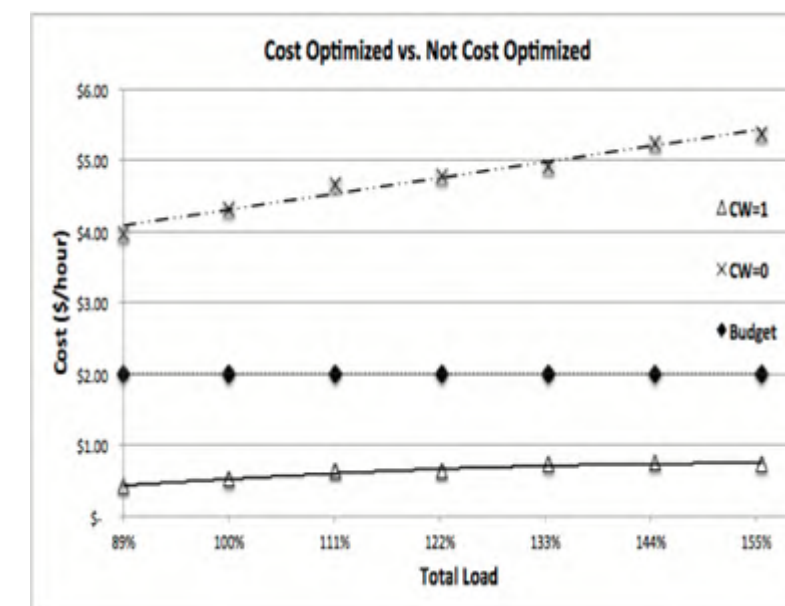
- 弹性调度
- 多机房架构改造



- 打破语言环境差异
- 建立全公司统一平台

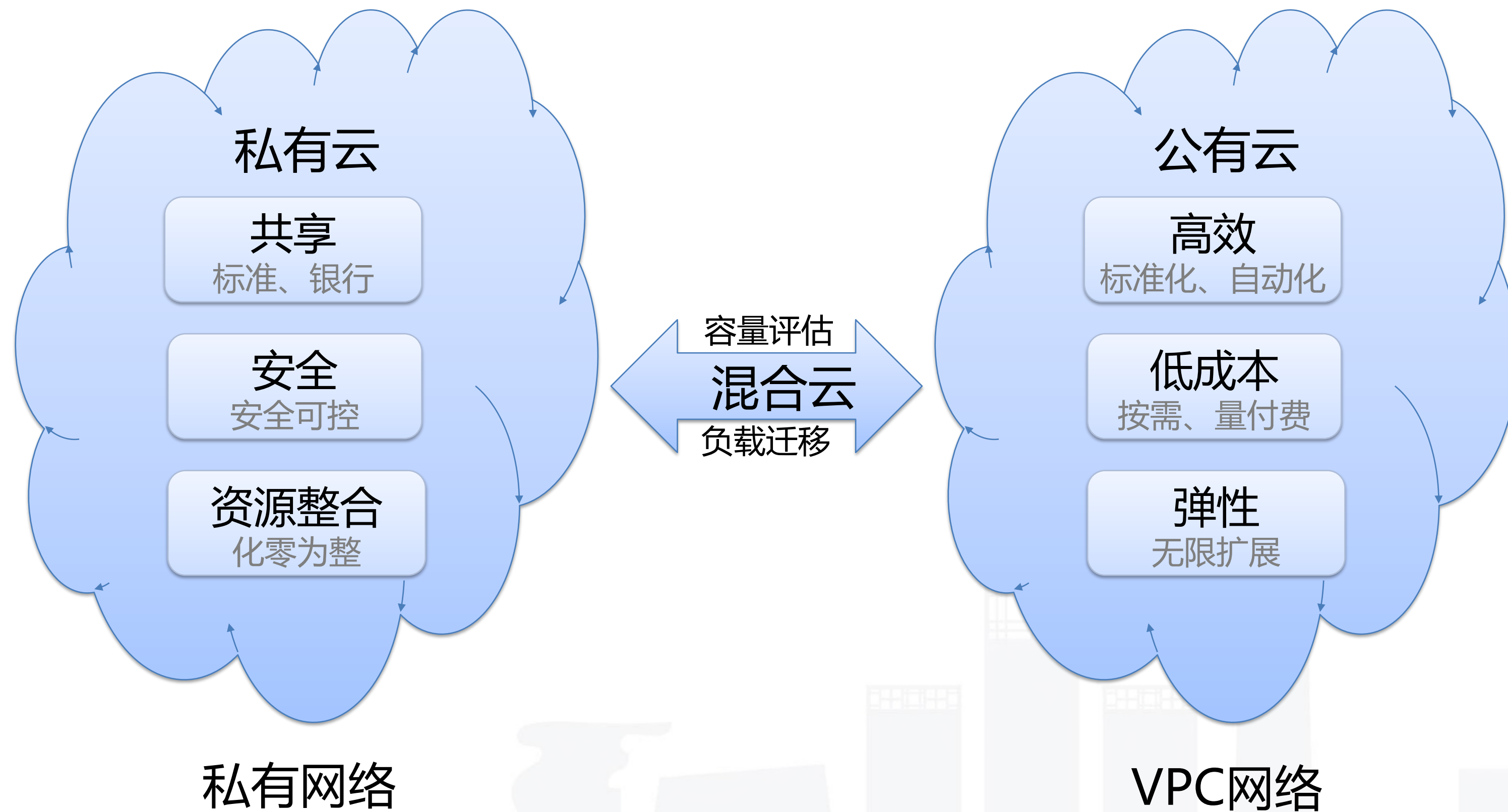


- 公有云弹性调度
- 私有云资源整合



- DevOps提高研发及发布效率

# 混合云趋势





# 混合云架构应用实践

混合云DCP ( Docker Container Platform ) 平台演进

多机房架构改造演进

业务服务保障体系

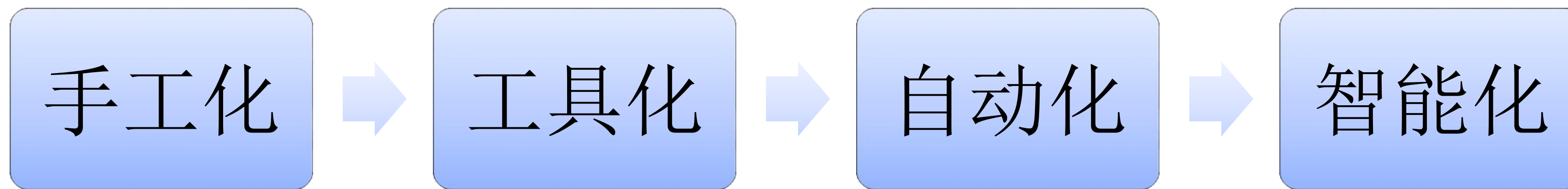


## Part 2

# 混合云DCP平台演进



# 混合云DCP平台



# 工具化-Docker化

- 应用部署

- 不同语言

- Java、PHP

- 环境差异

- 依赖基础环境OpenStack、公有云（OS、JDK、Nginx等）

- 依赖脚本、基础环境配置（启动脚本、定时任务等）

- 目录结构等

- 不同语言环境导致工具化很难实现

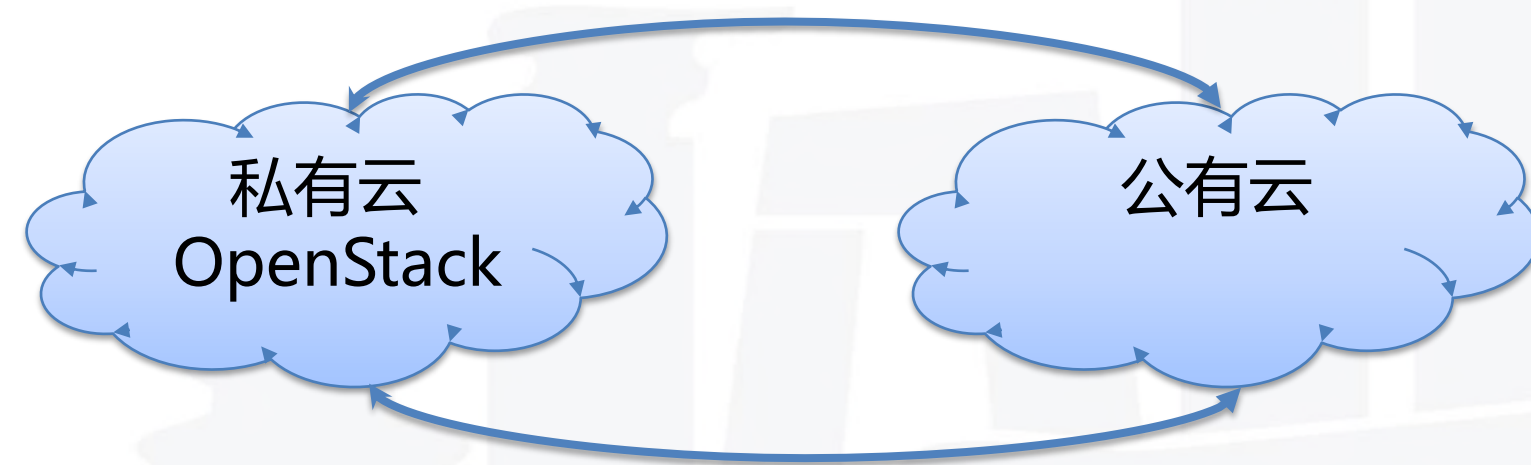
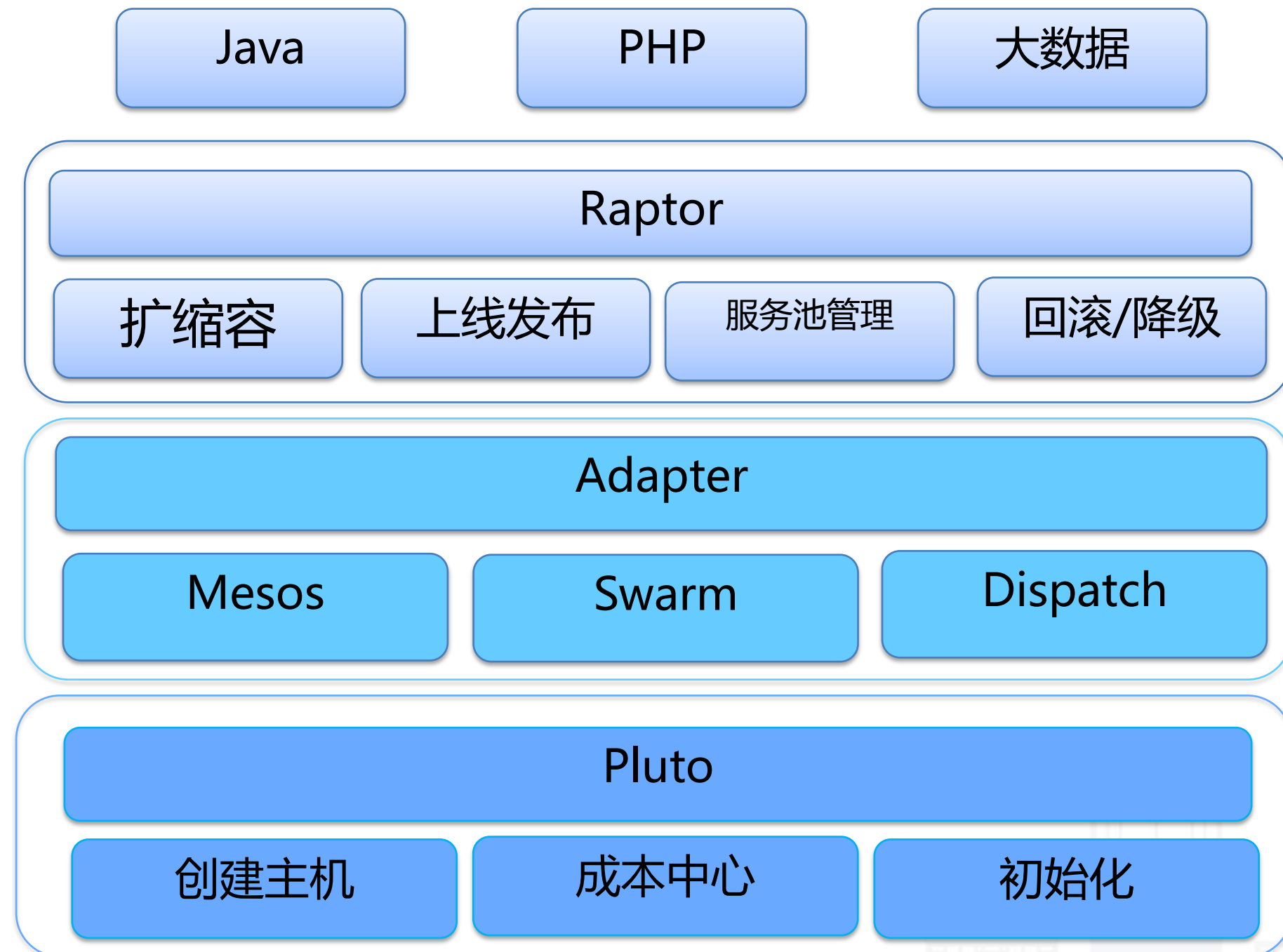
- 运维研发效率低 —> Devops

- 触发Docker化原因

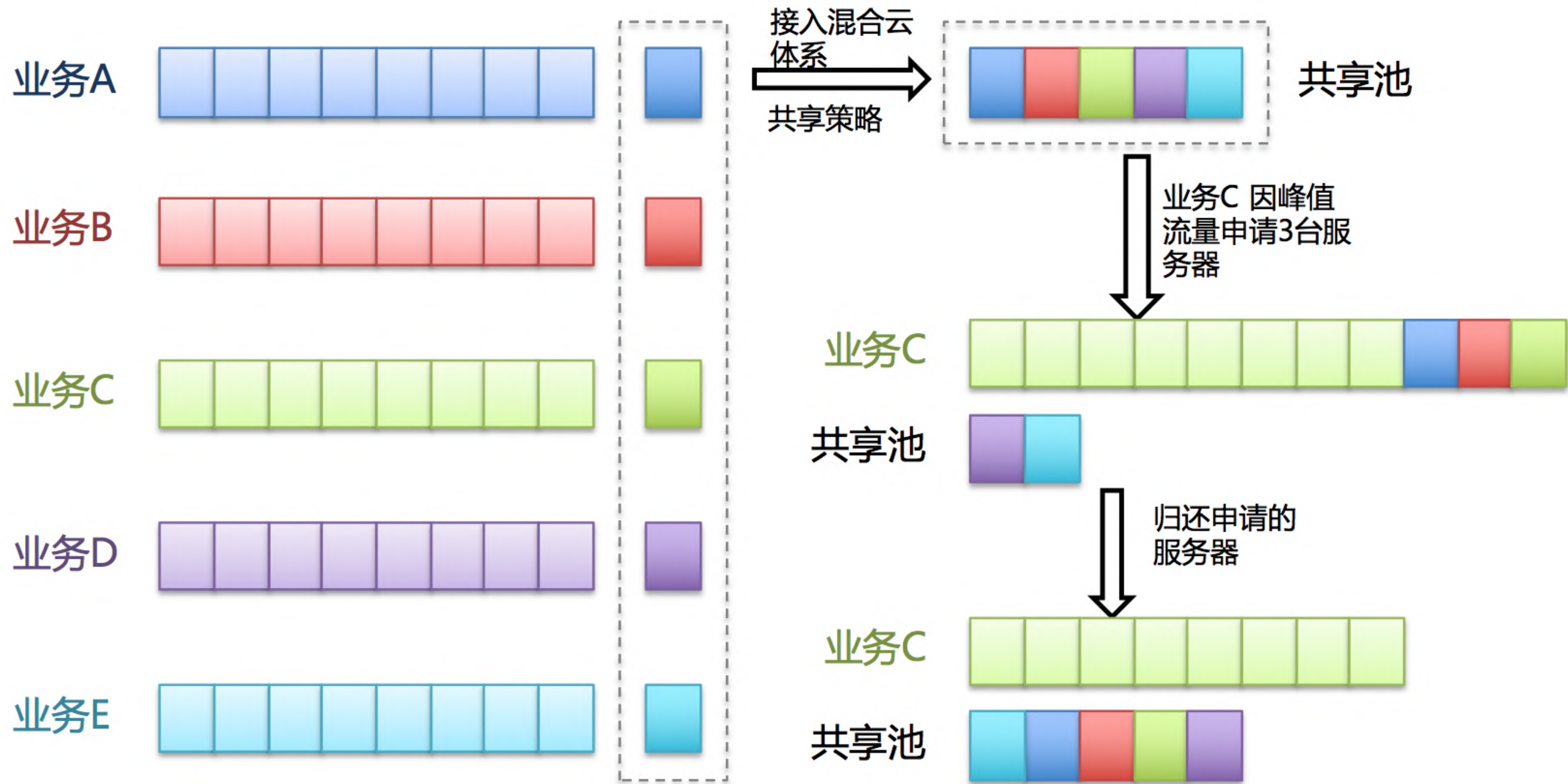
# 工具化

- 设备从哪里来？快速弹性扩容！
- 基础架构模块拆解
  - 物理主机层 – 基础设施跨云
  - 服务调度编排 – 基于Docker弹性调度
  - 依赖基础设施：Docker Registry、服务发现、DNSServer等
- 面临的问题
  - 业务方推广
  - 工具目标
    - 解决业务痛点问题
    - 简化生成力，而不是增加复杂度
  - 大规模下高可用、成功率

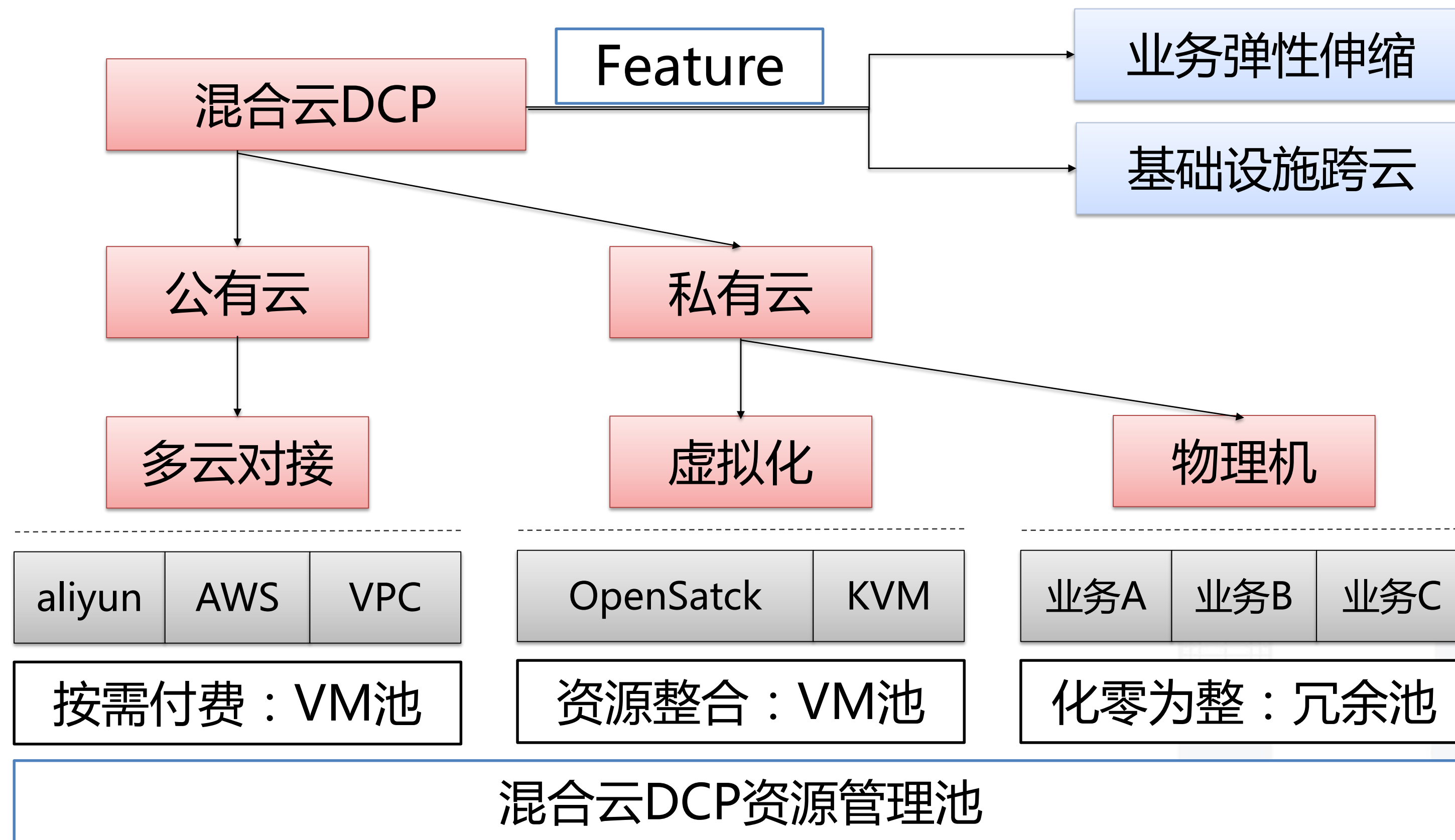
# 混合云DCP平台



# 混合云架构-资源共享



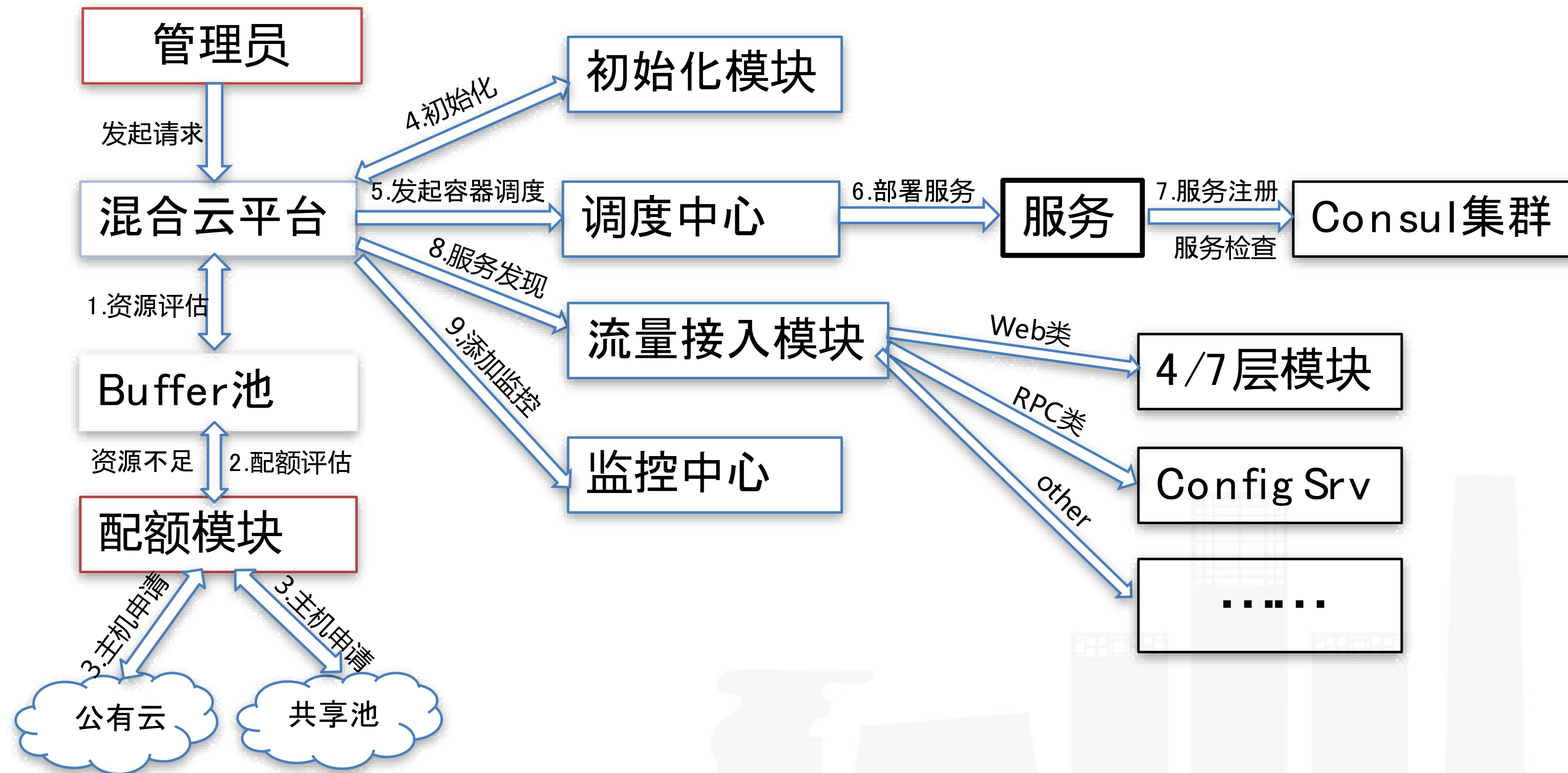
# 混合云DCP平台





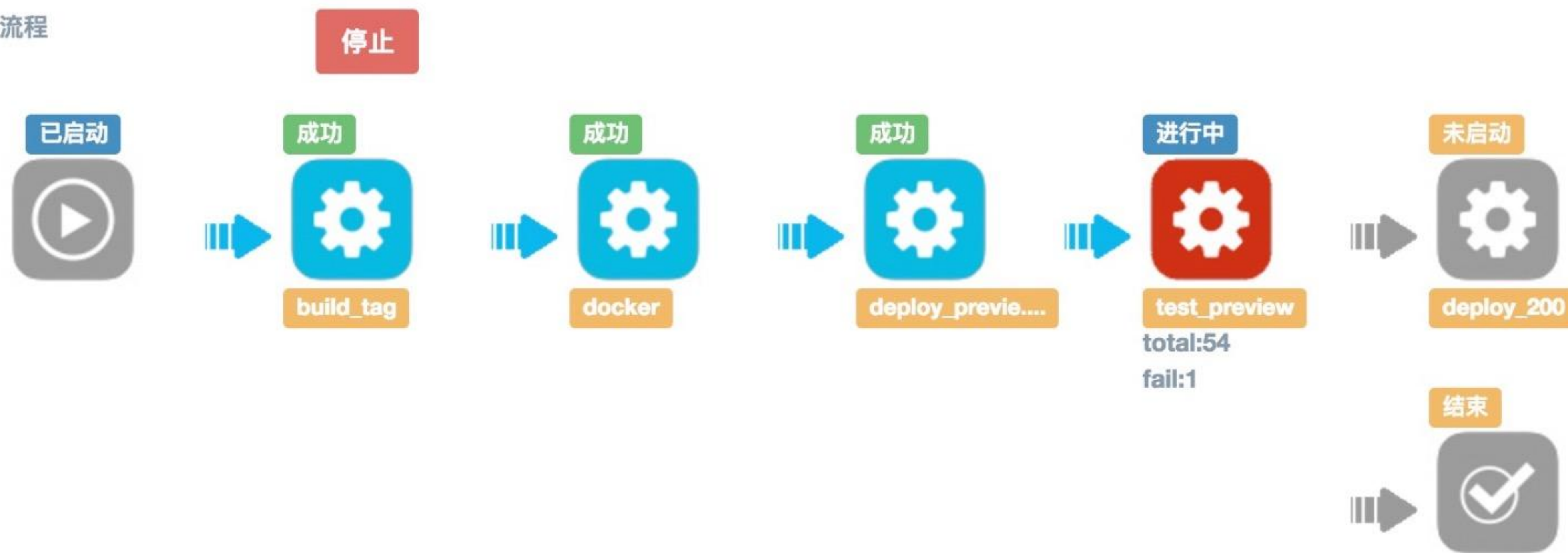
# 弹性扩容

一键扩容



# DevOps

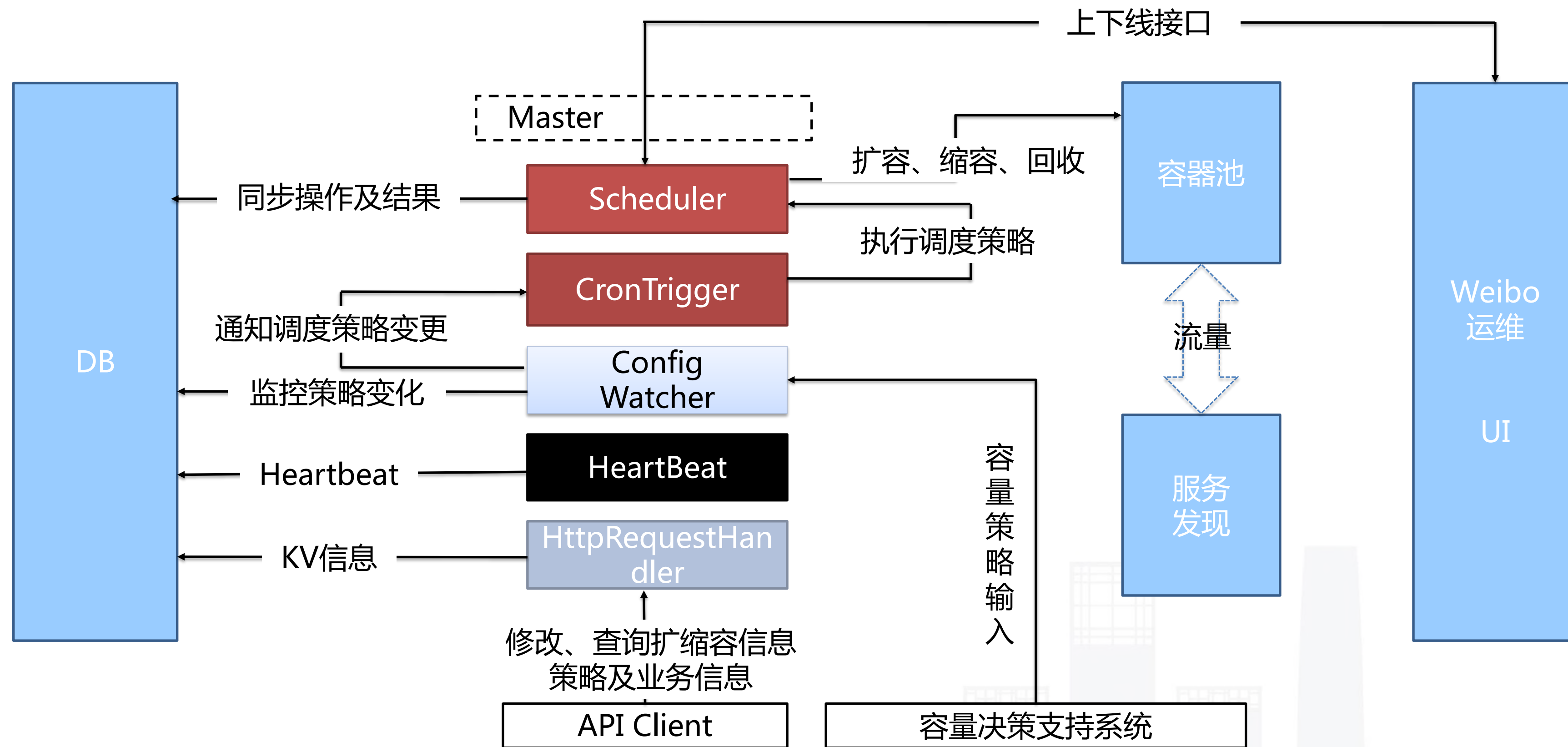
流程



# 自动化

- 工具化后技术同学幸福感还不够？
  - 弹性扩容涉及多个环节（主机创建、初始化、弹性调度、服务发现等），运维同学是崩溃的...
  - 人工工具操作导致出现问题增多
  - 10分钟1000节点扩容能力无法满足
- 自动化目标：**从工具中人参与多个环节演进到无人值守**
- 主要问题：非常难
  - 成功率
  - 服务依赖
  - 架构支撑能力

# 自动化任务调度框架-Schedule



# 自动化

- 批次处理
  - 单台失败导致整批失败
  - 依赖失败导致任务失败
- 流式处理
  - 单任务以单台为粒度
  - 增加重试、容错
- 高可用保障
  - 每日晚高峰定时扩容演练
  - 双主、监控报警机制

# “无人值守” 调度框架-Schedule UI

弹性调度 社区 文档

调度系统任务信息 task & ip List

[Readme](#) [Log](#)

任务列表

[添加任务](#) [删除任务](#) [修改任务](#)

No.	<input type="checkbox"/>	Sid	任务名称	创建者	调度方式	依赖关系	修改时间	上一个job	下一个job	当前总数	失败数
1	<input type="checkbox"/>	1604212050051001	aliyun-user-rpc-action	guansheng	depend		2016-07-08T14:31:02	--	--	0台	0台
2	<input type="checkbox"/>	1604212051101002	aliyun-v4-action	guansheng	crontab	1604212050051001	2016-07-12T01:05:40	00:42 0台	20:02 5台	0台	0台
3	<input type="checkbox"/>	1604251659071262	aliyun-user-rpc-status	guansheng	depend		2016-07-08T14:27:10	--	--	0台	0台
4	<input type="checkbox"/>	1604251700301263	aliyun-v4-status	guansheng	crontab	1604251659071262	2016-07-12T01:06:02	00:43 0台	20:06 5台	0台	0台
5	<input type="checkbox"/>	1606121152540001	aliyun-user-rpc-action-tc	guansheng	depend		2016-07-08T14:27:27	--	--	0台	0台
6	<input type="checkbox"/>	1606121205490002	aliyun-v4-action-tc	guansheng	crontab	1606121152540001	2016-07-09T21:57:48	23:45 0台(昨)	21:03 2台	0台	0台
7	<input type="checkbox"/>	1606121207320003	aliyun-user-rpc-status-tc	guansheng	depend		2016-07-08T14:27:38	--	--	0台	0台
8	<input type="checkbox"/>	1606121208130004	aliyun-v4-status-tc	guansheng	crontab	1606121207320003	2016-07-12T01:06:52	00:44 0台	21:02 10台	0台	0台
9	<input type="checkbox"/>	1606121209540005	aliyun-user-rpc-core-tc	guansheng	depend		2016-07-11T20:14:14	--	--	0台	0台
10	<input type="checkbox"/>	1606121210550006	aliyun-v4-core-tc	guansheng	crontab	1606121209540005	2016-07-11T20:13:51	21:55 0台(昨)(Disable)	21:01 5台(Disable)	0台	0台
11	<input type="checkbox"/>	9030820217062000	aliyun-ansible	guansheng	crontab		2016-07-07T18:29:29	22:55 0台(昨)	20:02 3台	0台	0台
12	<input type="checkbox"/>	9030836500065000	aliyun-registry	guansheng	crontab		2016-07-08T21:05:21	22:55 0台(昨)	20:02 3台	0台	0台
13	<input type="checkbox"/>	9030965188405000	aliyun-user-rpc-core	guansheng	depend		2016-07-08T14:28:24	--	--	0台	0台

ScheduleType:  crontab  depend  once  auto

Jobs:

Time:	00:50	InstanceNum:	0	
Time:	12:01	InstanceNum:	15	
Time:	12:51	InstanceNum:	0	
Time:	20:01	InstanceNum:	20	
Time:	21:01	InstanceNum:	30	
Time:	23:50	InstanceNum:	0	

+ 添加job

Dependencies:

Sid: 1612222119300000 Ratio 0.3

ElasticCount: 0

+ 添加依赖

Sconfigs:

Name: aliyun AddRetry: 0

# “无人值守” 弹性流式扩容

整点开始操作

【日常】100台服务器扩容流程，总时间10分钟



# 智能化-容量评估系统

## ●两种决策方式

### ◆自动压测：训练模型

- 压测方法：减少服务池可服务的实例数
- 压测机制：503.sh/200.sh
- 压测机指标：数据来源于监控中心（粒度10s）

系统	Load	<12	12<X<24	>24
	Cpu idle	<30%	10%<X<30%	<10%
	Iowait	<20%	20%<X<35%	>50%
	Swap	<500M	1G<X<2G	>2G
业务	5xx错误比率	<1%	1%<x<5%	>5%
	接口平均耗时	<100ms	100-500ms	>1s

### ◆容量预估：数据来源于业务量监控

- 同比分析
- 环比分析

## ●容量评估产出：

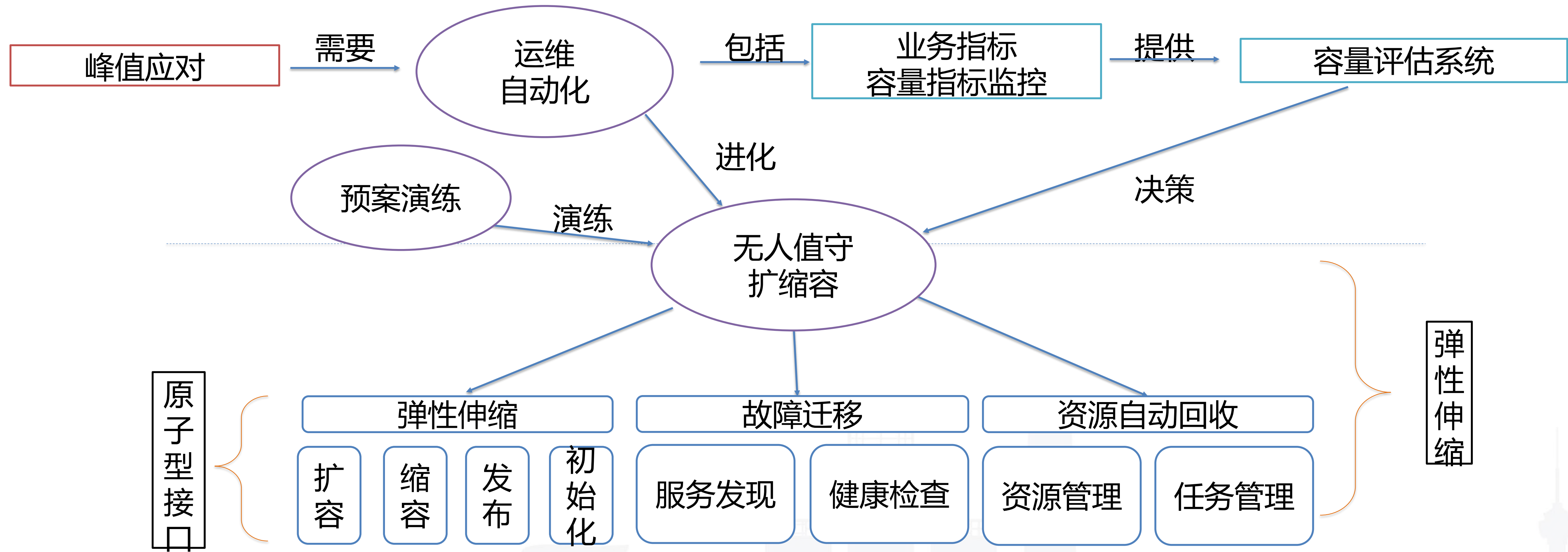
- 水位预警
- 容量报表
- 容量API

## ●容量评估系统





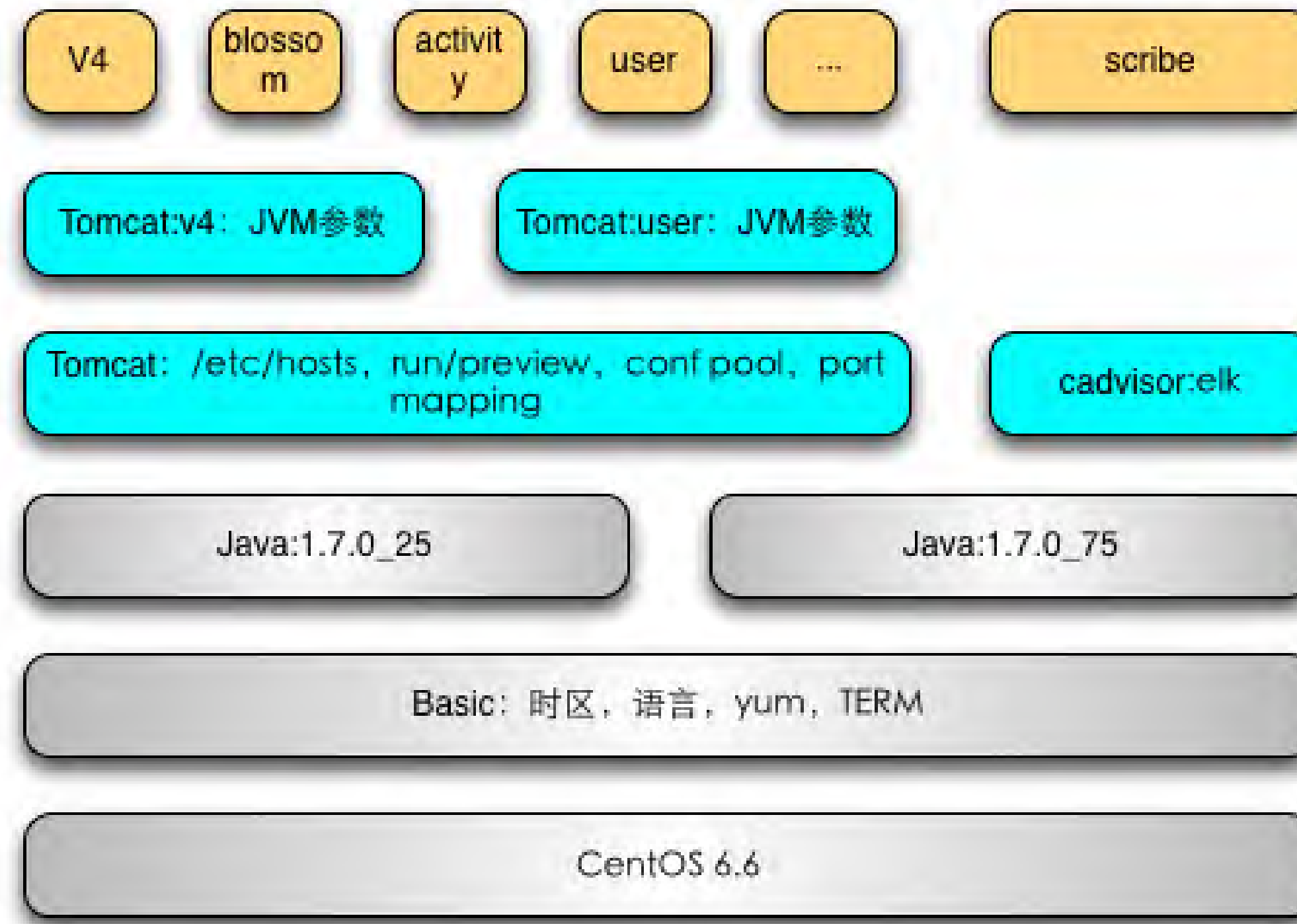
# 目标“无人值守”弹性扩缩容



# 大规模实践 – 镜像仓库 – 镜像分层服务

## ● 镜像服务

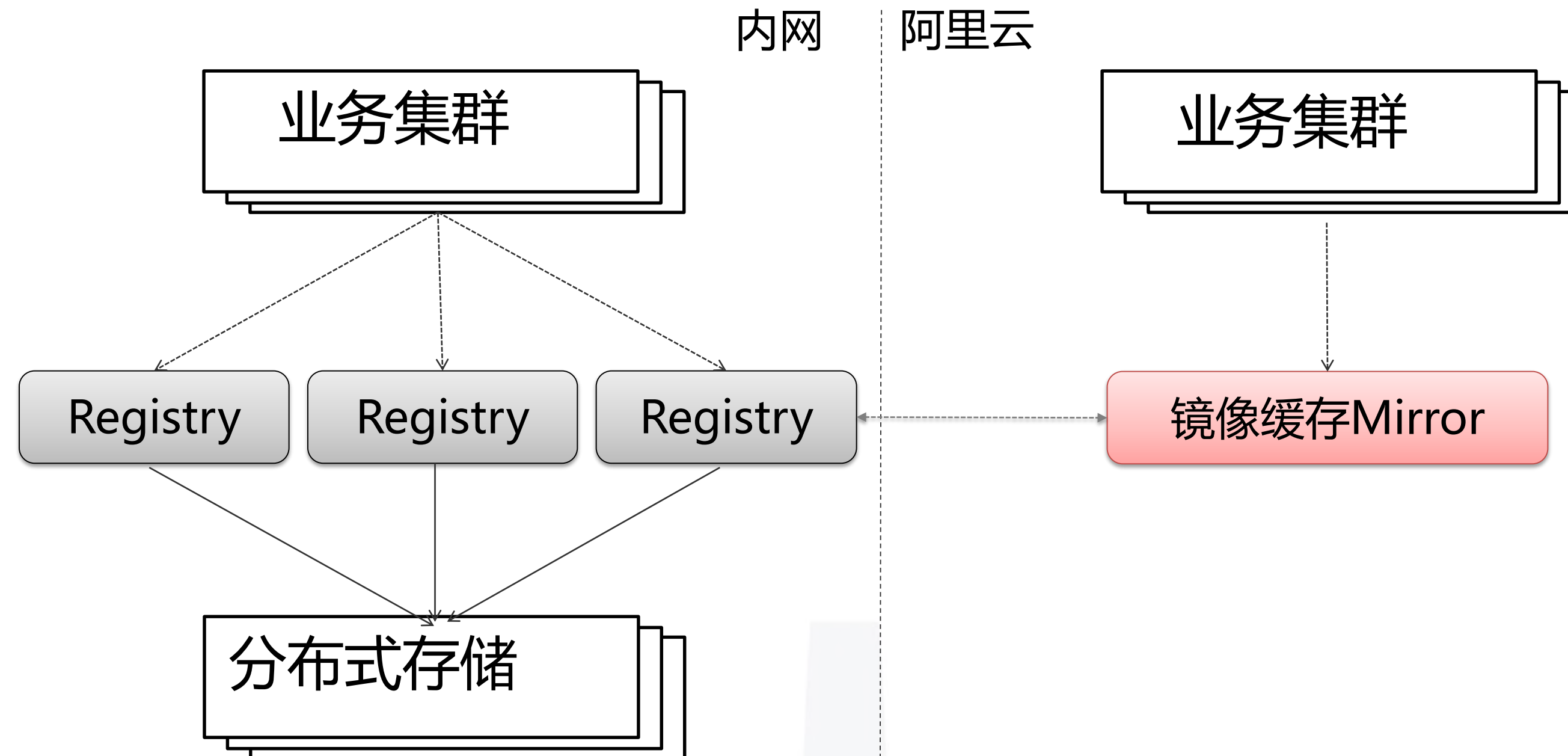
- 分层设计，逐层复用
- 基础环境/运行时/容器/业务
- 优化大小，dockerignore
- 禁止使用latest



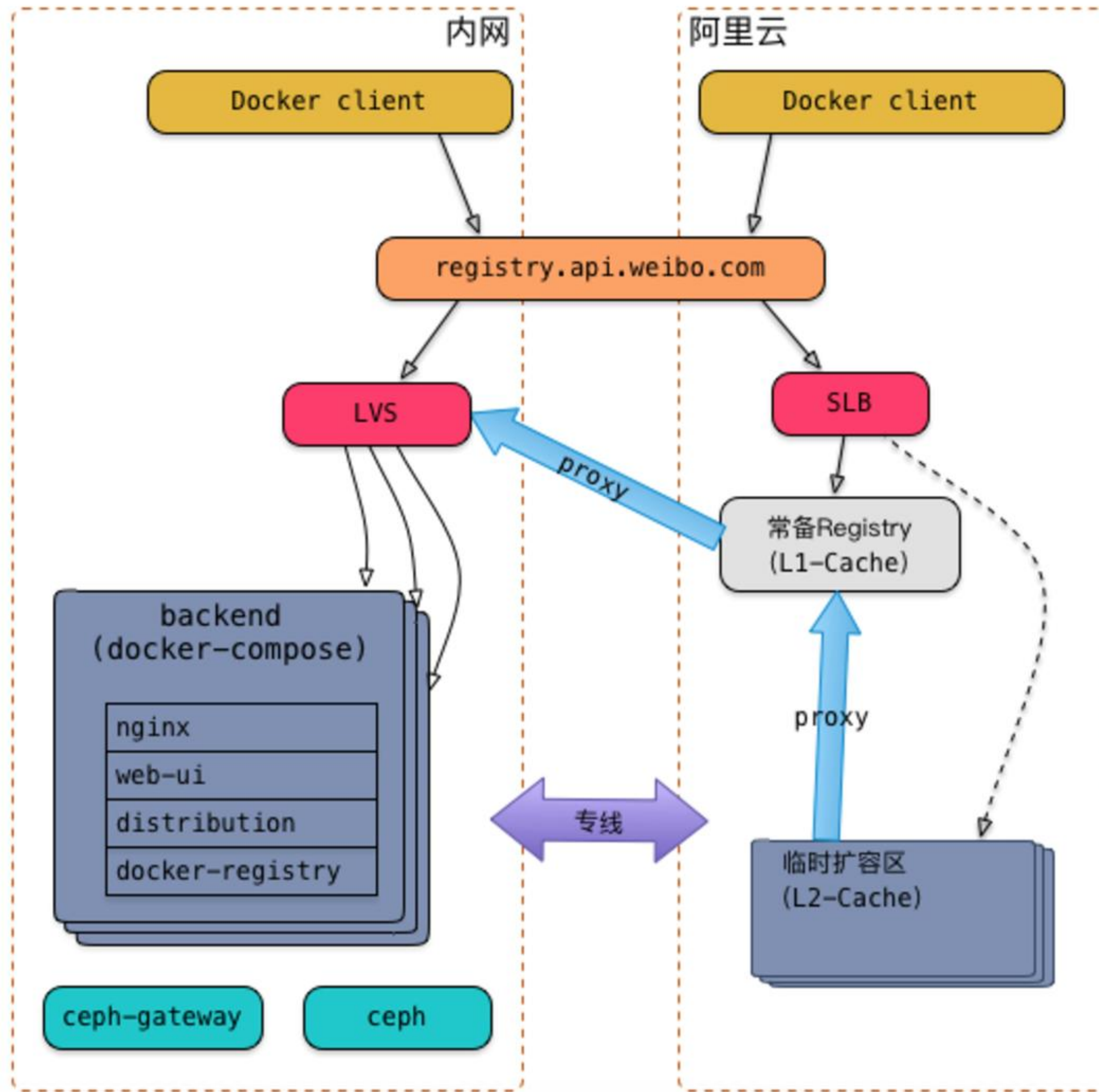
# 大规模实践 – 镜像仓库 – Docker Registry

## ●构建私有Registry Hub

- docker-registry : V1 -> V2
- storage driver : Ceph
- docker-registry-frontend : Nginx

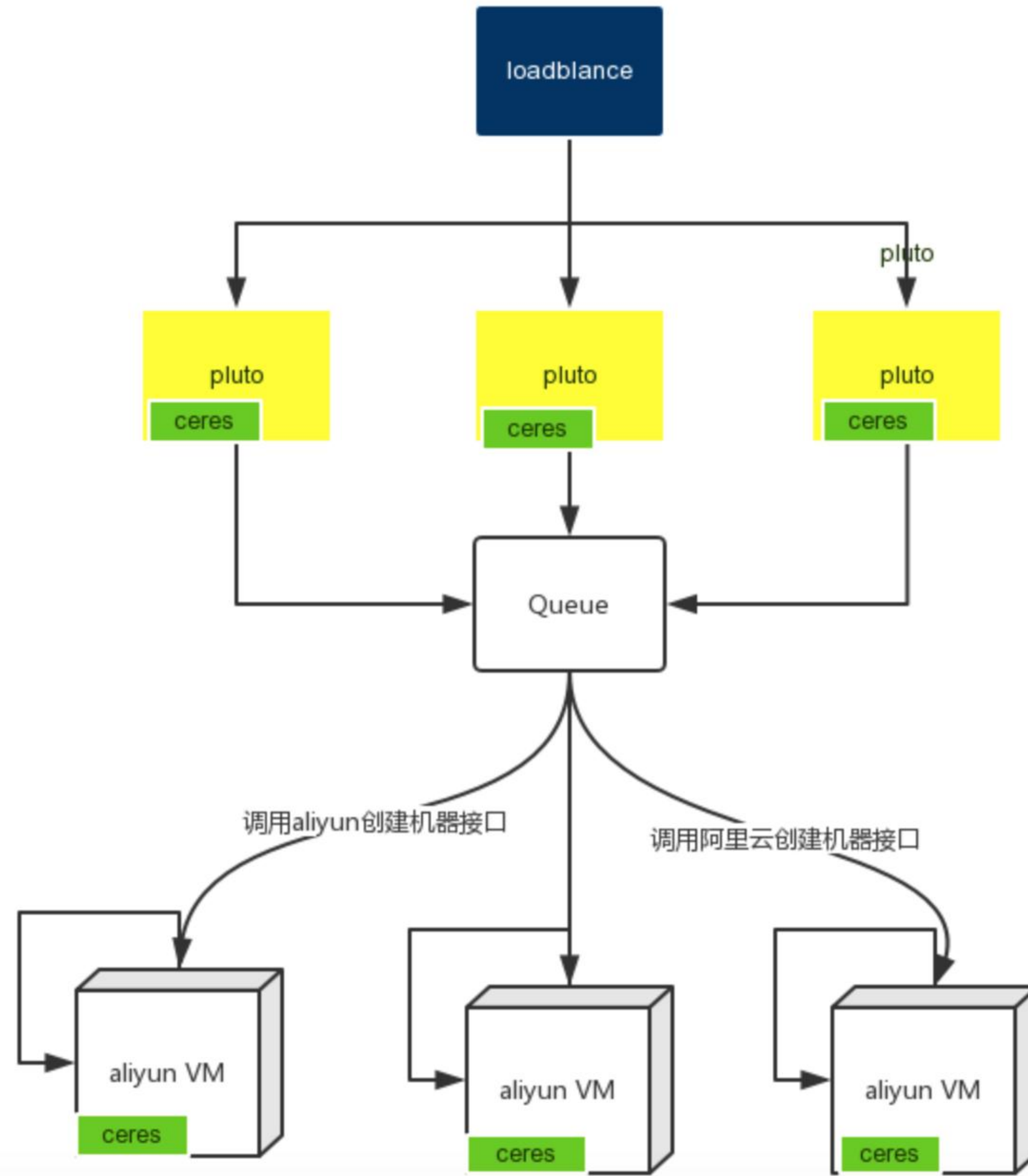


# 大规模实践-Docker Registry部署架构



# 大规模实践-基于Ansible配置分发

- SSH开启pipelining和ControlPersist
- Ansible前端增加调度队列，单机控制并发数
- 自定义callback，异步向队列中写入结果
- 分布式去中心化改造





# Part 3

## 微博多机房架构演进

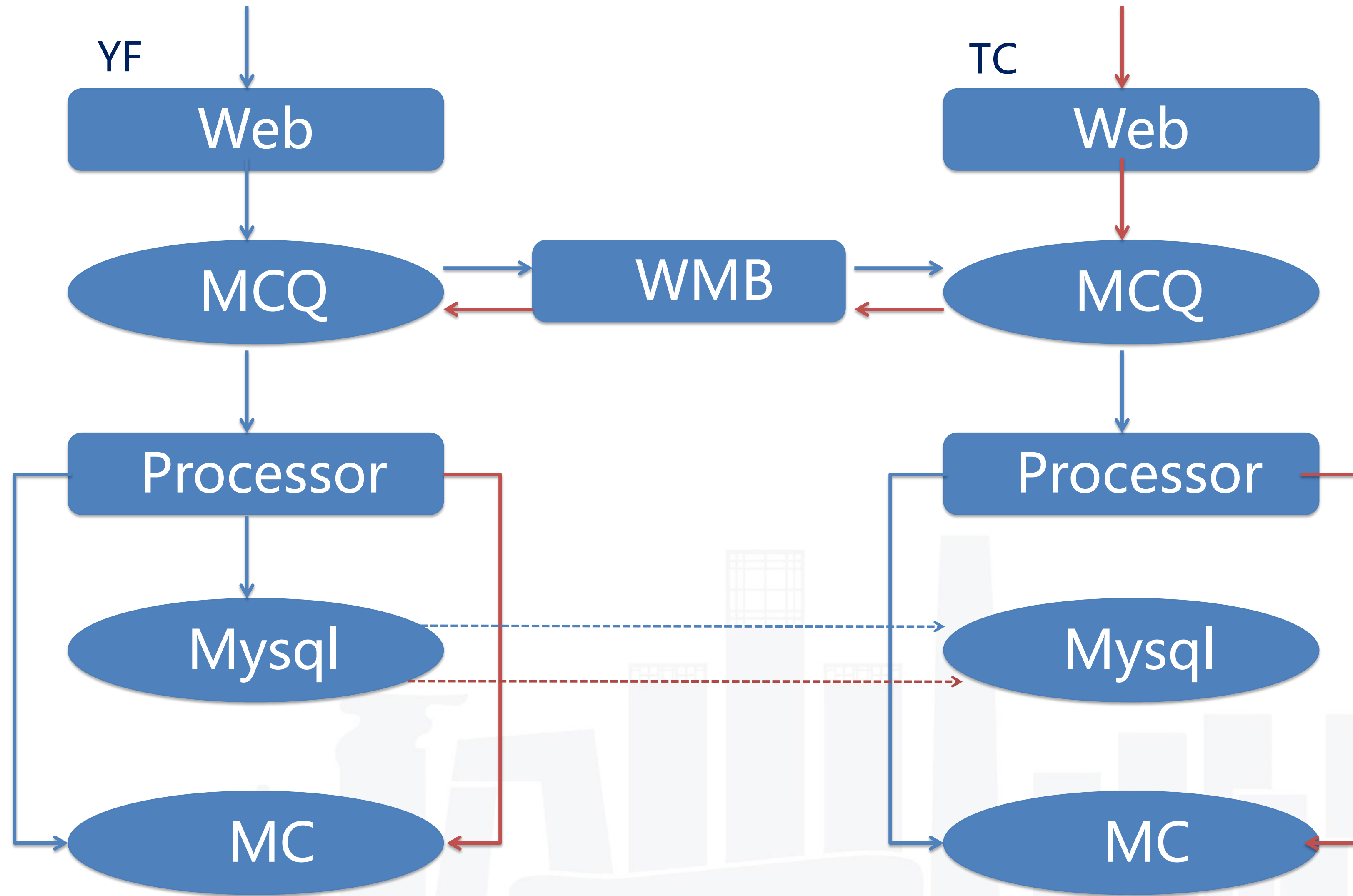


# 混合云多机房架构改造

- 性能延迟 < 200ms
  - 核心缓存部署
  - 业务架构改造：依赖链路长（用户关系、Card、UVE等）
    - 微服务改造：依赖服务部署
- 专线带宽优化（单台 < 30Mb）
  - Nginx、RPC数据压缩
  - 静态数据local化
  - **跨IDC流量规划**

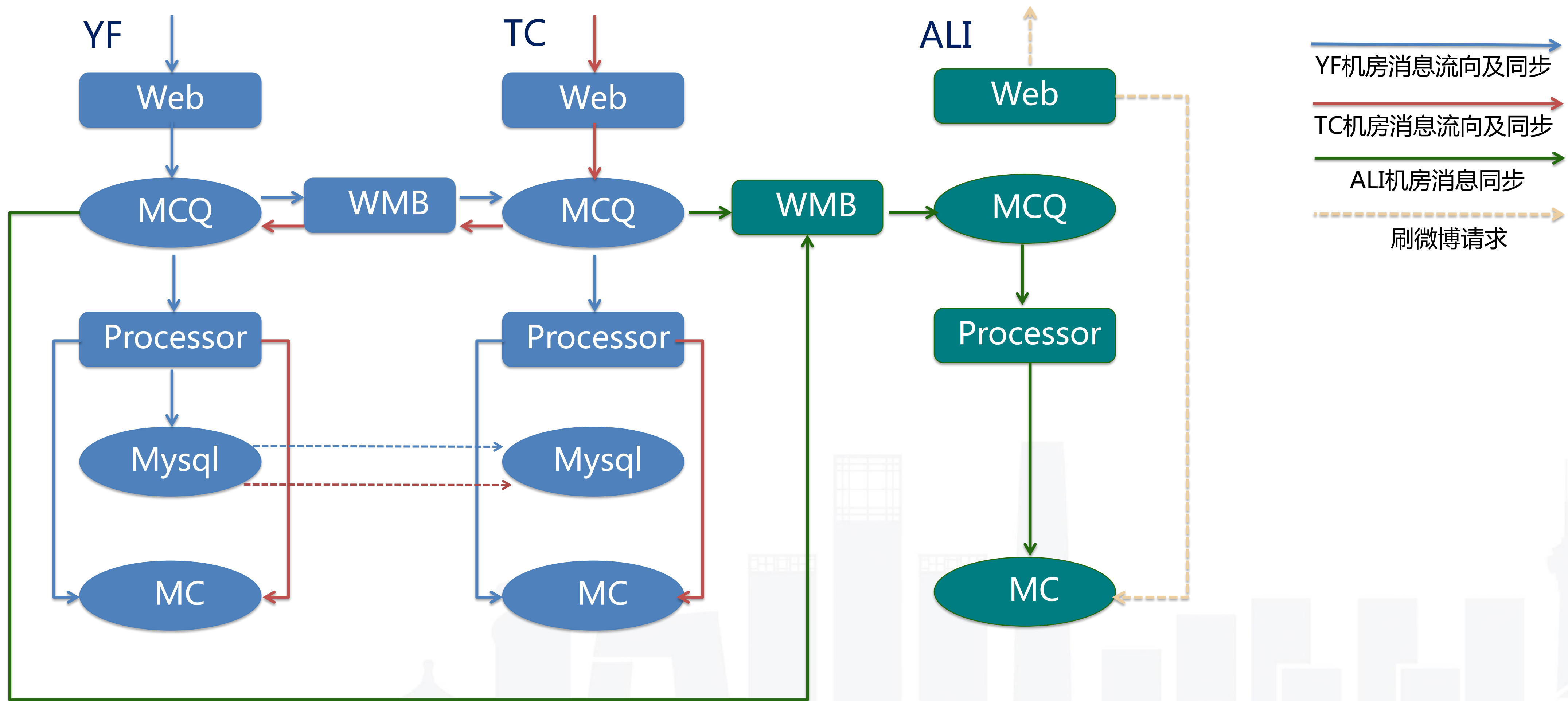


# 微博双机房架构

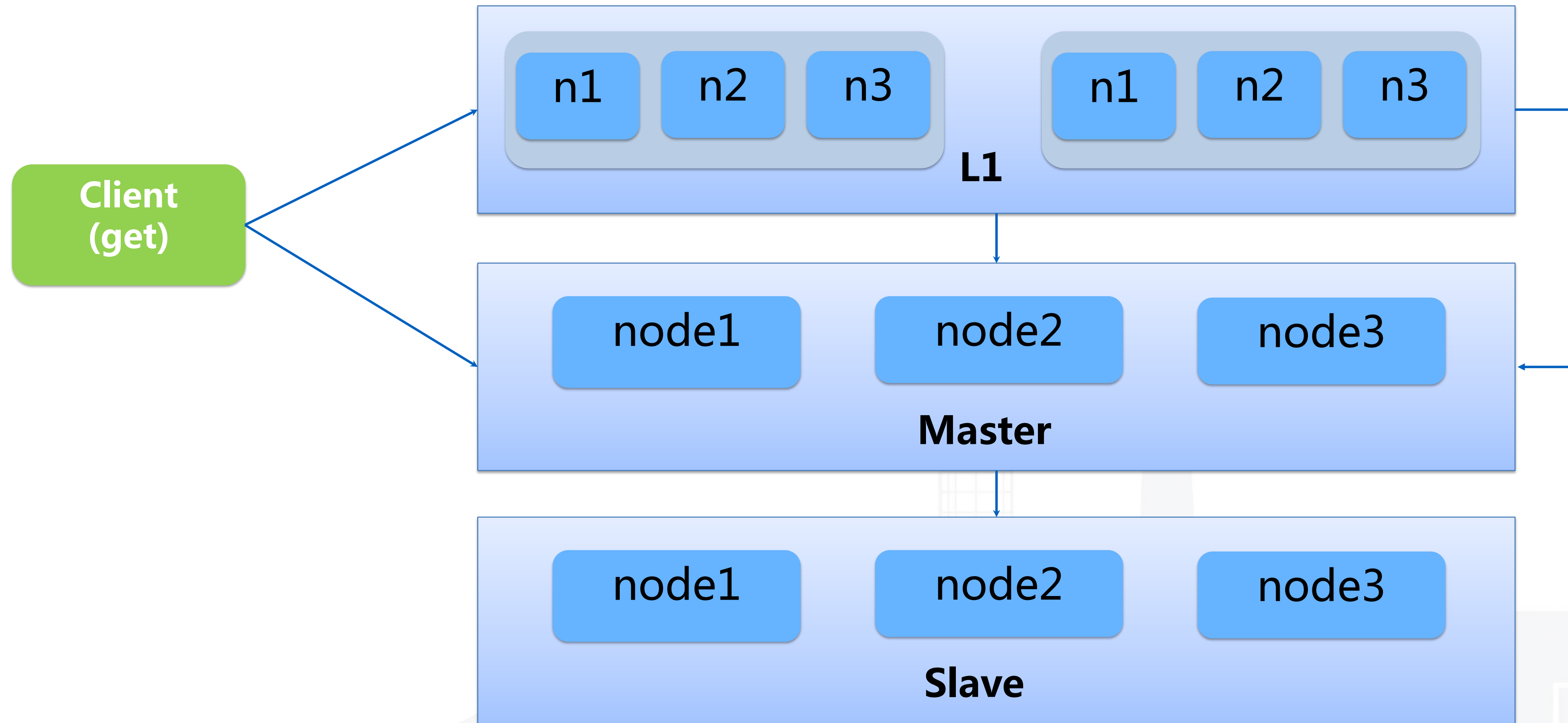




# 轻量级多机房架构

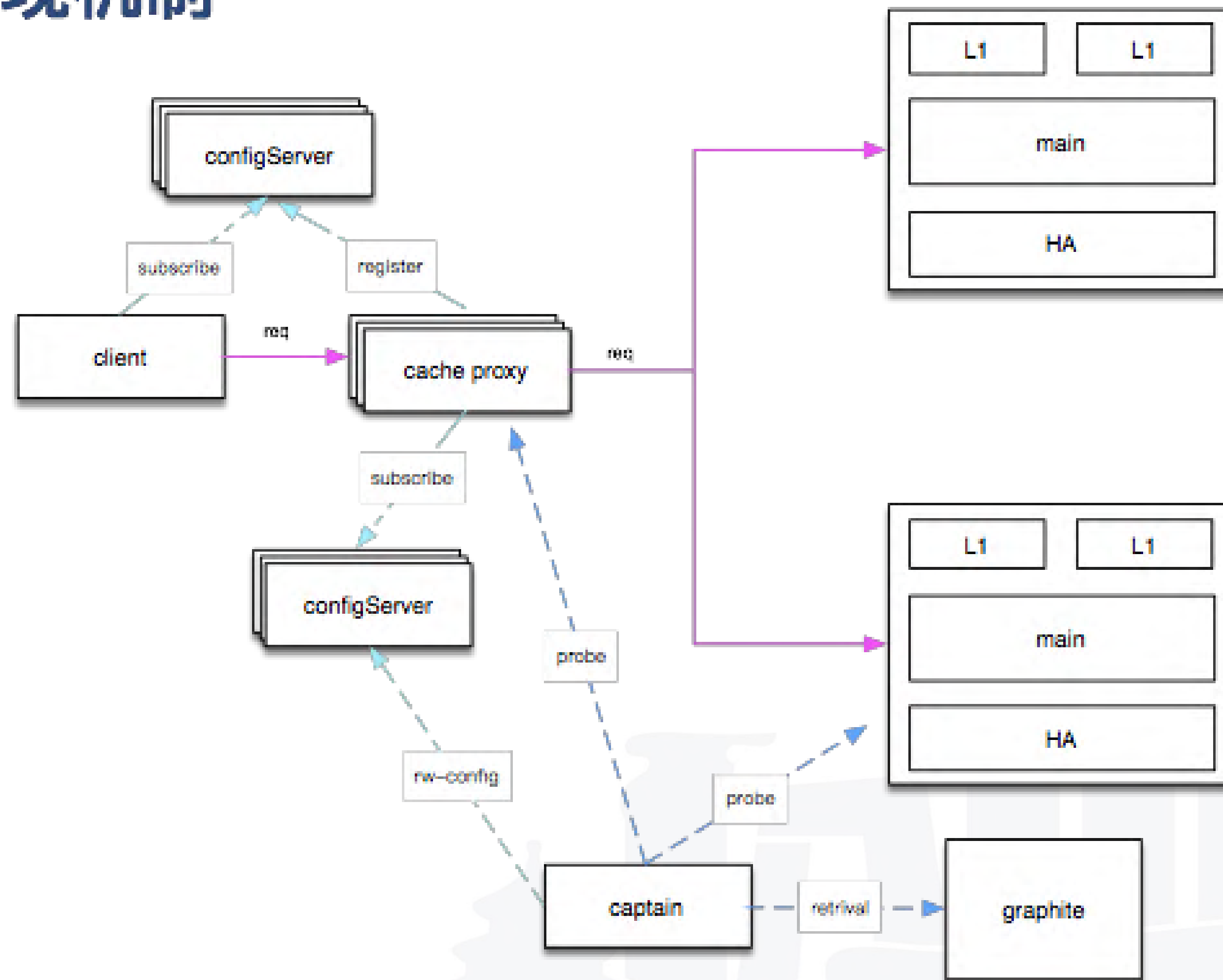


# 高可用、可扩展缓存集群

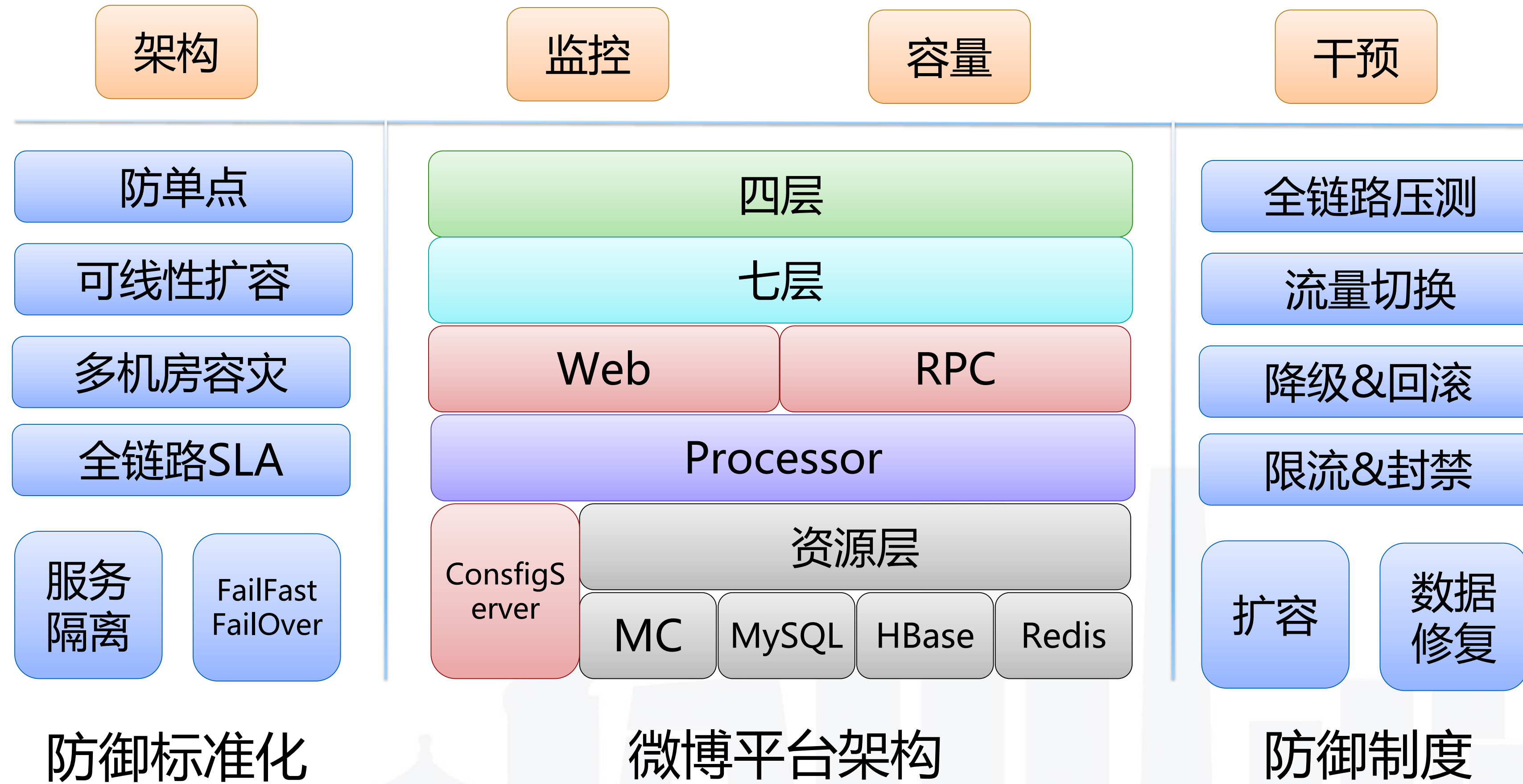


# 缓存服务化

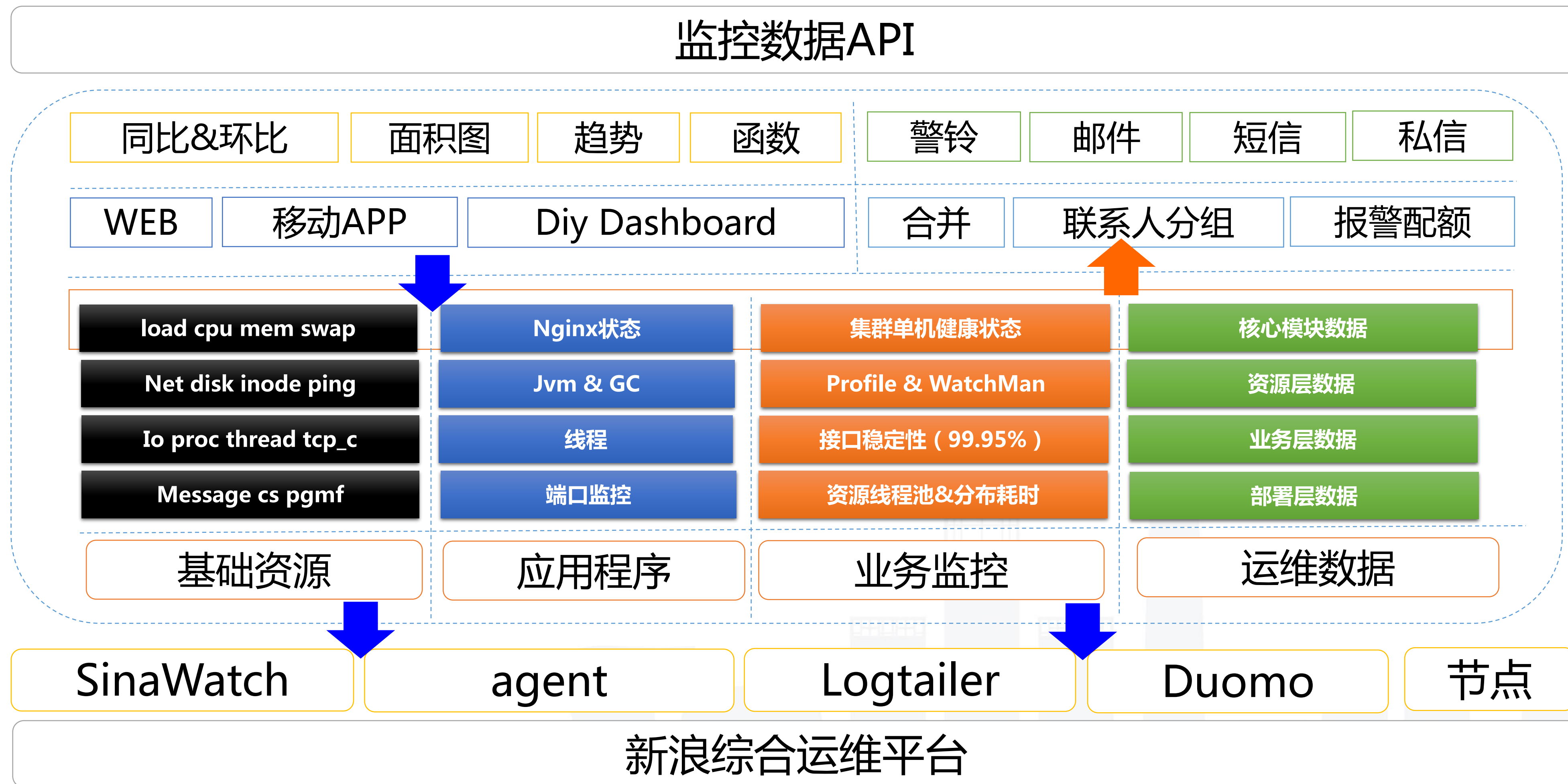
## 实现机制



# 防御体系框架



# 主动防御-监控体系



# DCP-监控分类案例

## 作战图类

云上服务器数量

IDC专线带宽

实时容量

## 监控类

QPS监控

avgtime

4××、5××

系统指标

## 问题定位类

单机slow top

资源slow top

分布耗时

网络质量监控

## 实时报警类

接口监控超过  
阈值告警

队列消息堆积

# 大规模应用实践-全链路压测-问题

- 公有云应用环境差异
  - 主机性能差异：PPS、IOPS等
  - 性能不稳定
- 环网流量优化
  - 网络链路优化
  - 业务架构优化





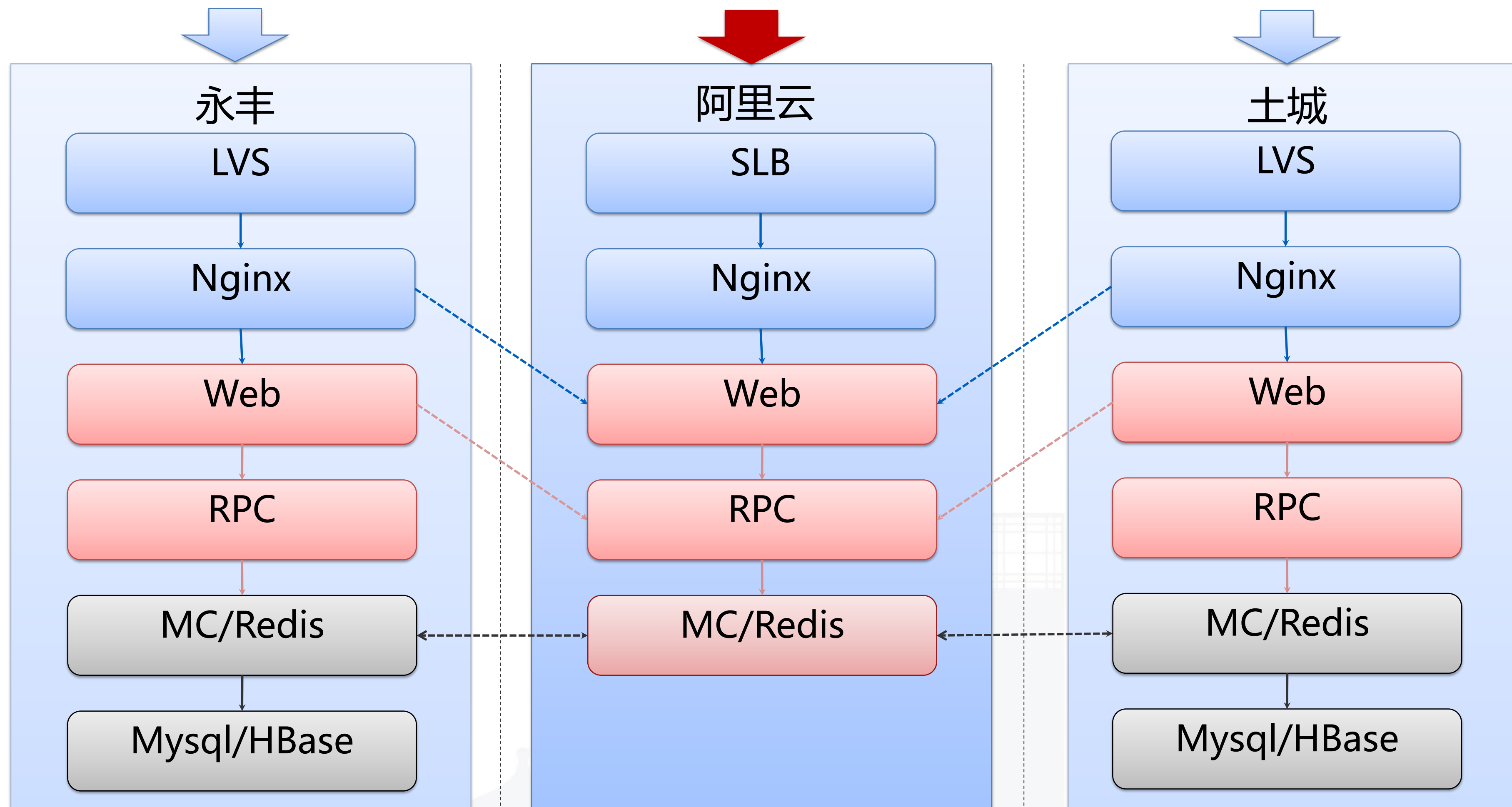
# Part 4

## 春晚峰值流量实战应对





# 混合云部署架构



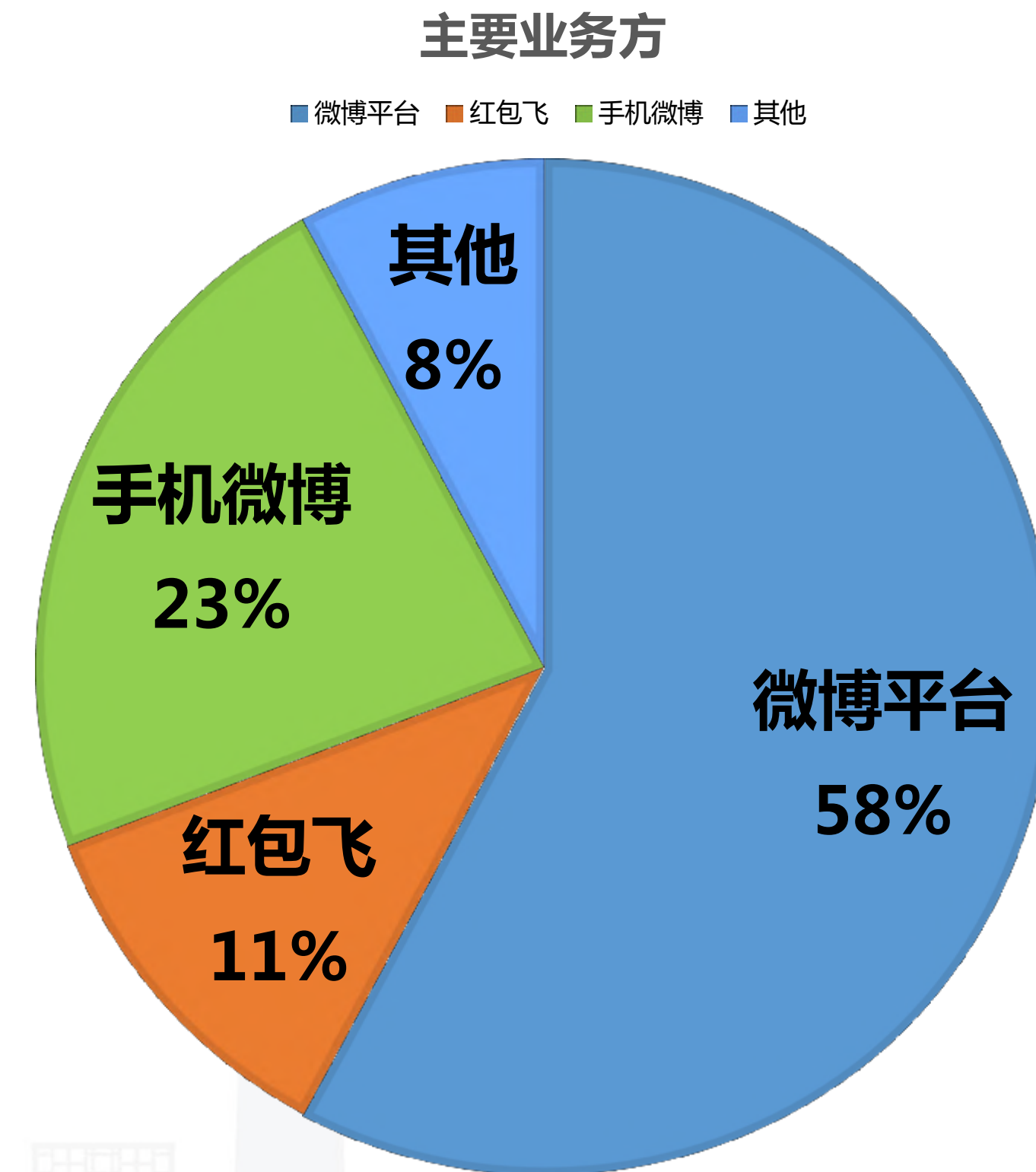
# 微博混合云成果

## ●混合云进展：

- 容器数：5000+
- 全国跨公司级最大混合云项目

## ●春晚备战：

- **10分钟混合云扩容1000节点技术能力**
- 2017春晚峰值历史新高，当天完成**4700台**阿里云ECS扩容，实现全程无操作无降级平稳度过，公有云高峰支持微博50%主体流量。完成e-ed流、红包飞、手机微博公司各主要业务线均完成上云支持



# OpenDCP : 基于Docker技术的混合云管理平台

OpenDCP : <https://github.com/weibocom/opendcp>

- 综合性的运维管理平台。涵盖运维配置、发布、上线变更等运维管理主要功能，而不局限于容器集群管理，可适配Kubernetes、Mesos、Swarm等
- 功能覆盖镜像市场、多云对接、服务编排、服务发现等云资源管理主要环节
- 支持阿里云、AWS、私有云等主流云厂商
- 支持Nginx、SLB等服务发现方式
- 支持Java、PHP、C/C++、Go等主流语言

欢迎沟通交流：

[itfuwen@163.com](mailto:itfuwen@163.com) @it\_fuwen





关注QCon微信公众号，  
获得更多干货！

# Thanks!



主办方 **Geekbang** > **InfoQ**  
极客邦科技