

WOTA

51CTO

World Of Tech 2017

全球架构与运维技术峰会

2017年4月14日-15日 北京富力万丽酒店

ARCHITECTURE



出品人及主持人：

**沈 剑**

58到家 高级技术总监

---

微服务架构实践

# 微博服务化的实践与演进

张雷-微博  
@rayzhang0603



**张雷**

新浪微博技术专家

MotanRPC框架技术负责人

**分享主题：**

微博服务化的实践与演进

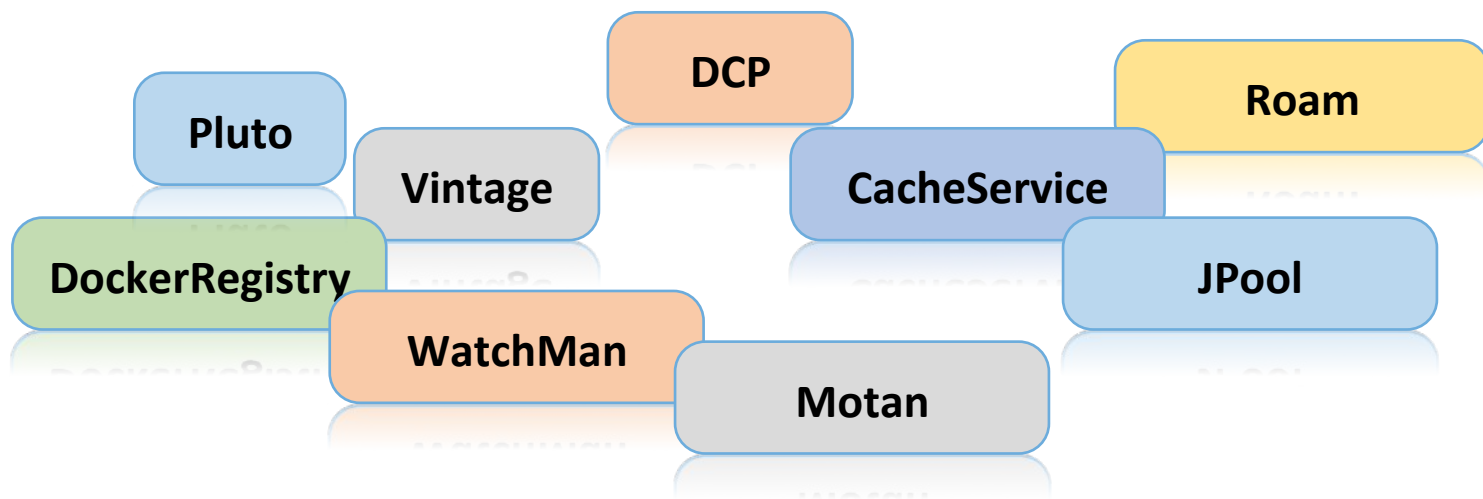
# 内容简介

- 微博服务化简介
  - 微博服务化简介
- 微博服务化发展历程
  - 敏捷优先的模块化服务
  - 基于Motan RPC框架的微服务
  - 混合云上的微服务化体系

# 微博服务化 简介

## 微博服务化简介

- 规模：百级业务线、千级服务、万级节点
- 服务调用：API百亿级、RPC万亿级
- 服务组件：





## 微博服务化简介

- 服务化、SOA、微服务





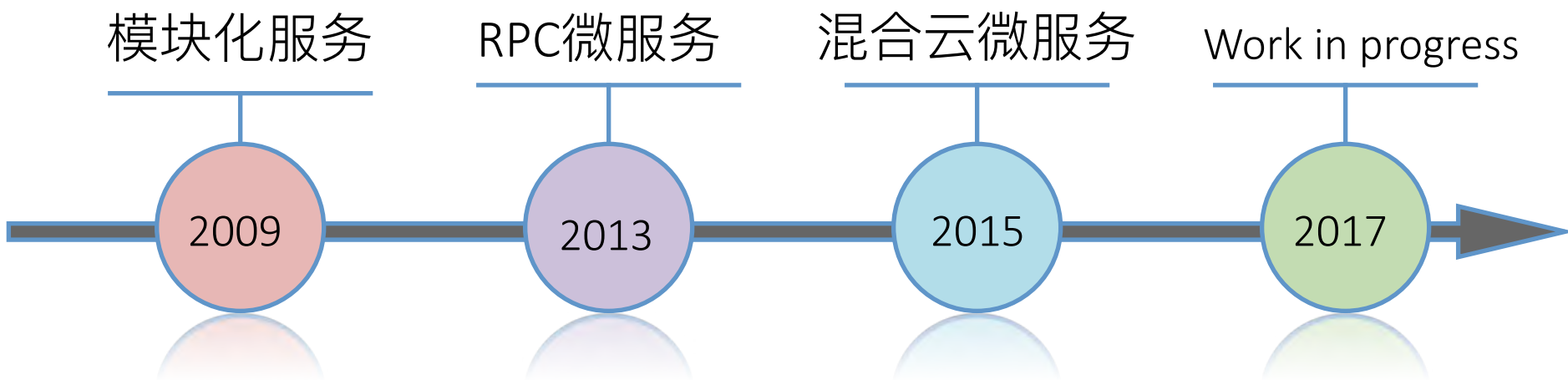
# 三个问题

- 为什么要做服务化
- 怎么实现服务化
- 实现服务化会遇到哪些问题



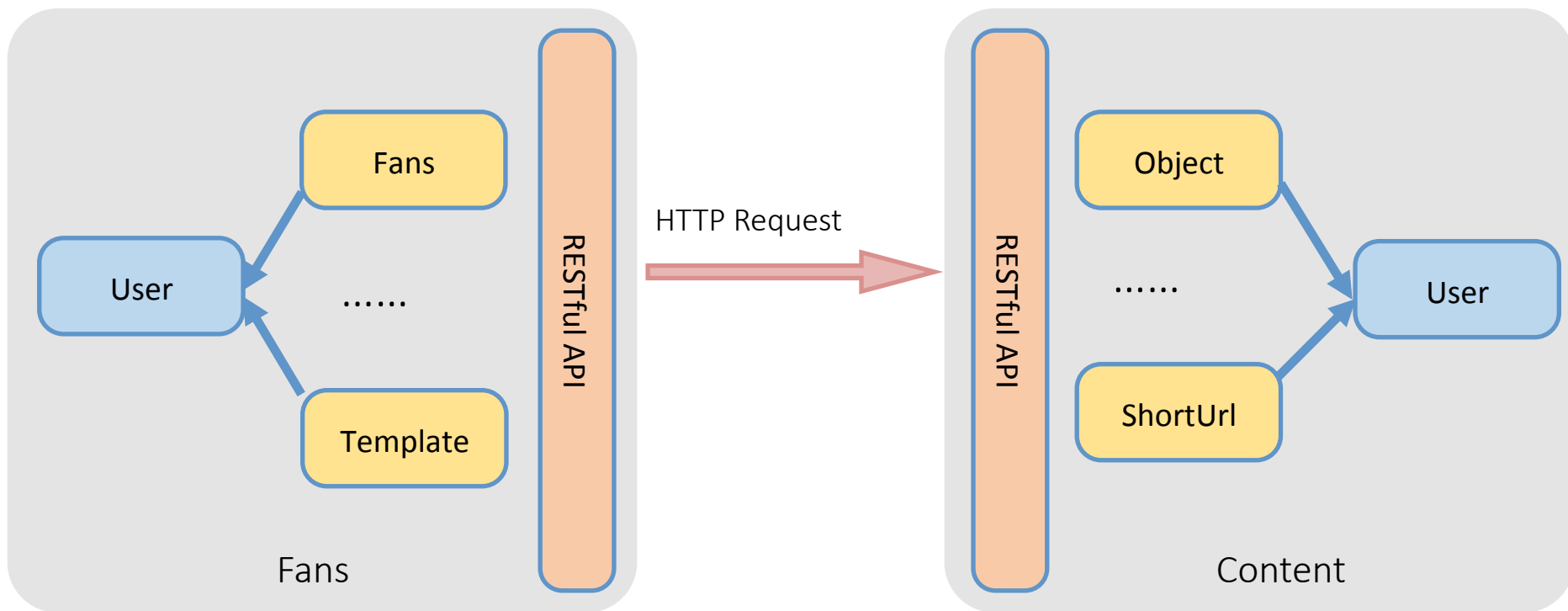
# 微博服务化发展 历程

# 微博服务化三个阶段



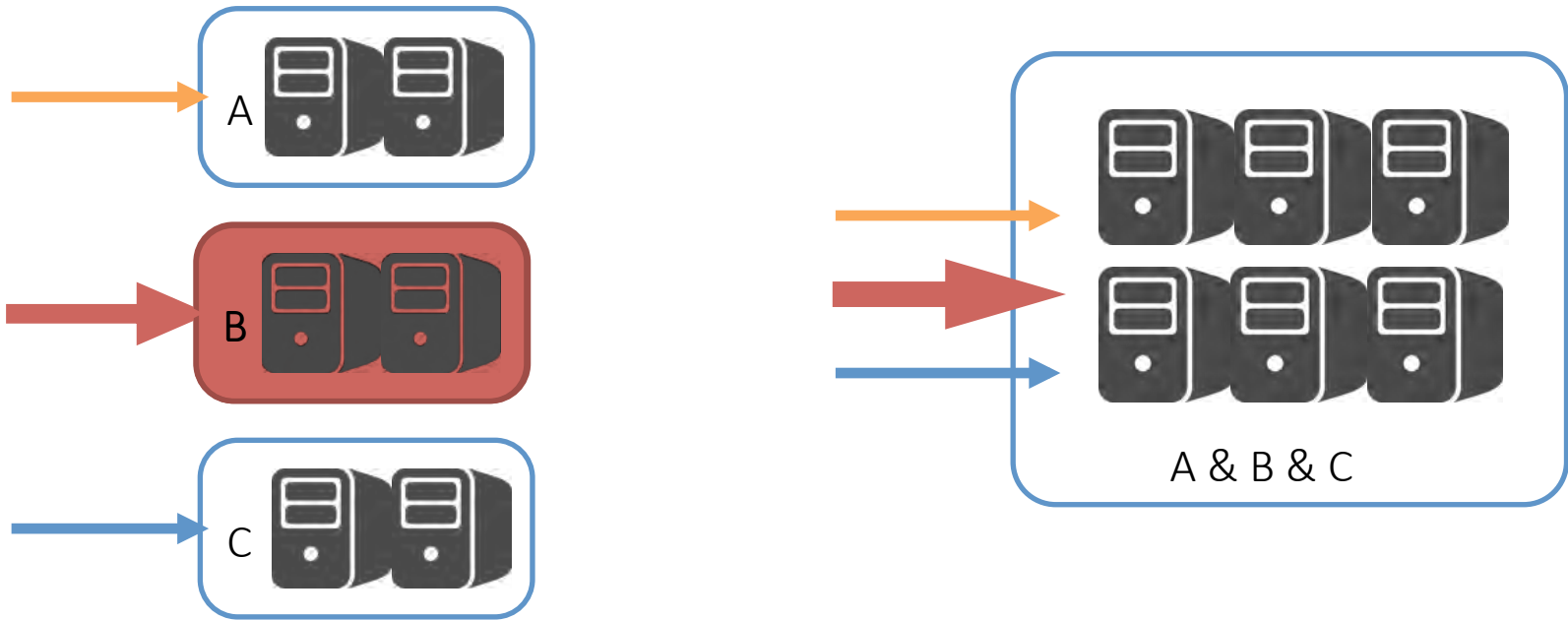
# 模块化服务阶段

- 时间：2013年以前
- 背景：初期高速发展
- 服务部署：大服务池
- 服务交互：API + JAR

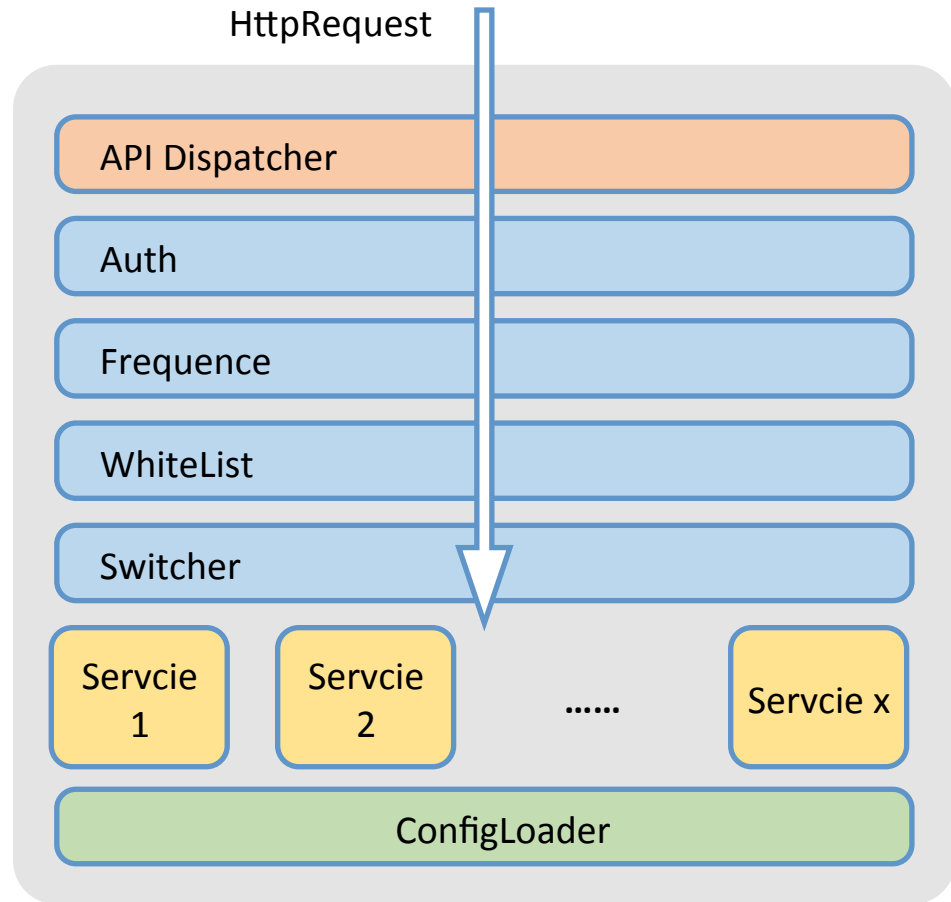


模块化服务交互示例

- 大服务池vs小服务池



## Cedrus RESTful Framework





# 减少服务之间相互影响

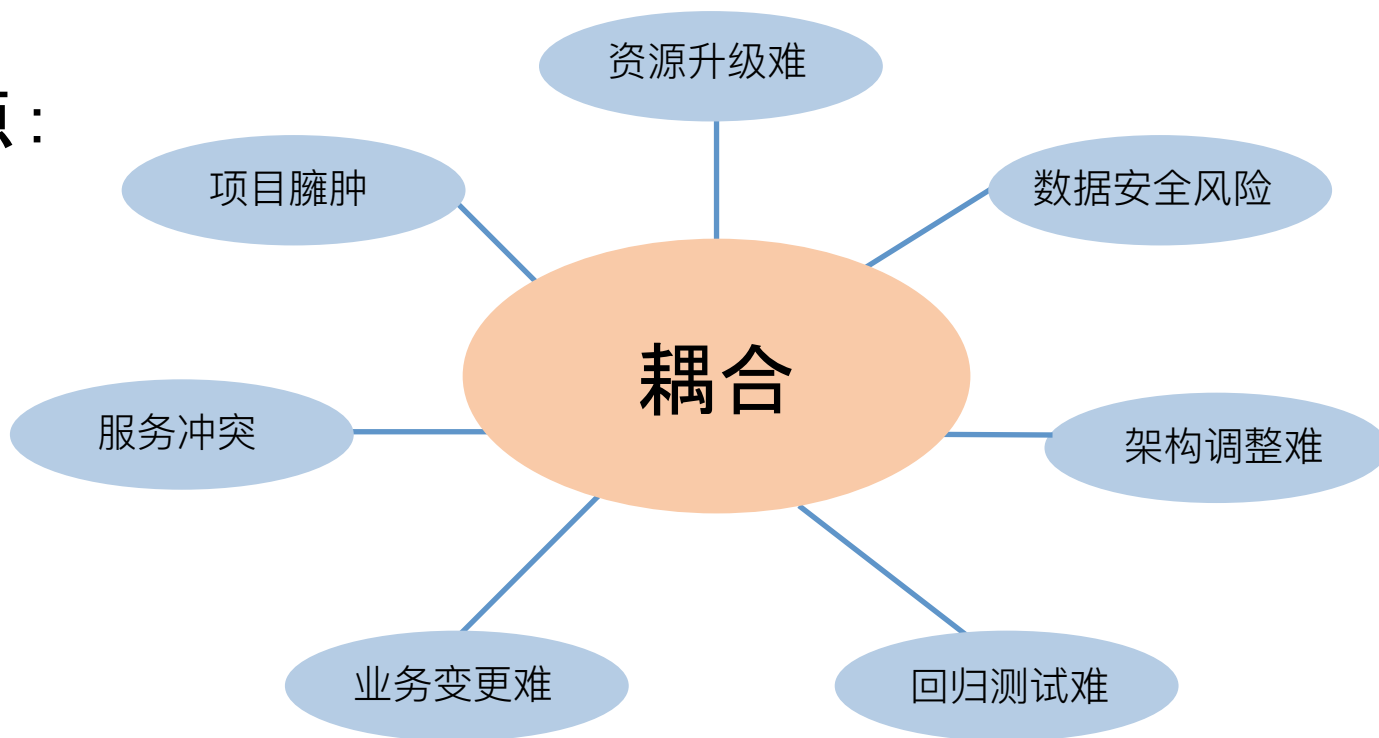
- API并发限制
- 接口、source、uid、认证的降级开关
- 黑白名单、接口频次限制

# 小结

优点:

- 合理利用资源
- 服务调用性能较高
- 开发相对独立

缺点：



# RPC微服务阶段

- 时间：2013年-2015年
- 背景：业务发展稳定，规模庞大
- 服务部署：独立部署
- 服务交互：API + Motan RPC

# 服务化面临的选择

- HTTP VS RPC
- 使用开源 VS 自研

# 对RPC的要求

- 优秀的性能
- 模块化设计，高扩展性
- 基础的服务治理功能
- 高可用性
- 定制化需求
- 与现有管理系统等无缝结合

# Motan RPC 框架

<https://github.com/weibocom/motan>



This repository

Search

Pull requests

Issues

Gist



[weibocom](#) / [motan](#)

Unwatch

420

★ Unstar

2,760

Fork

1,033

<> Code

Issues 124

Pull requests 5

Projects 0

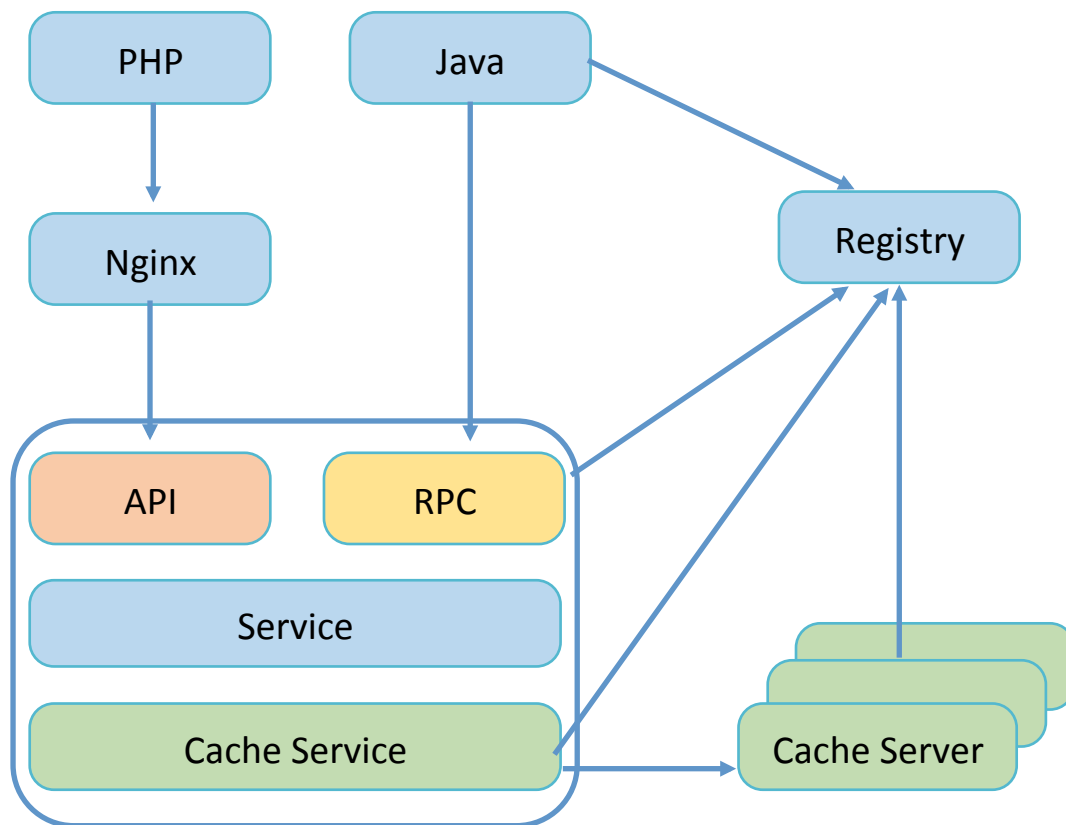
Wiki

Pulse

Graphs

Settings

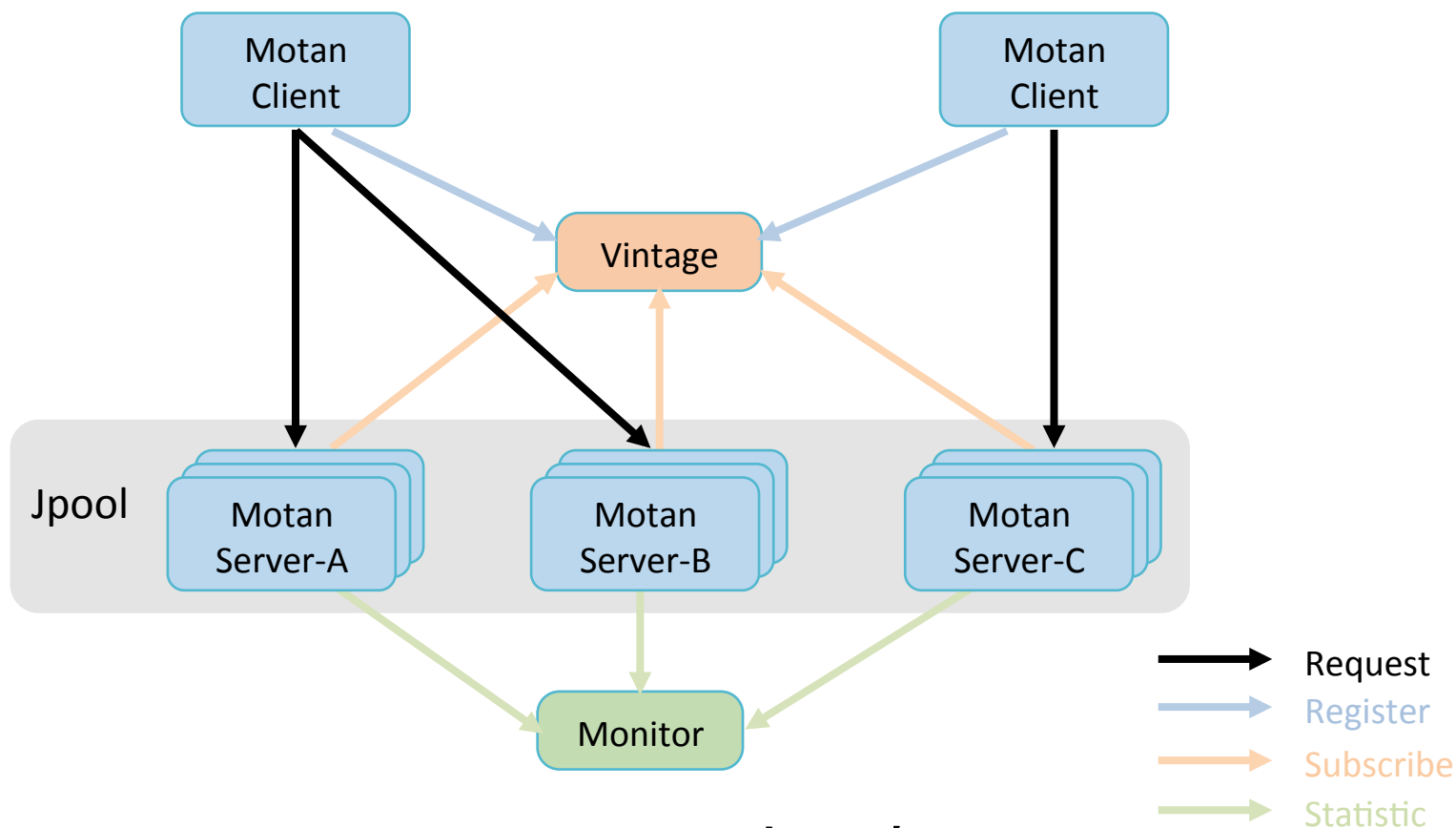




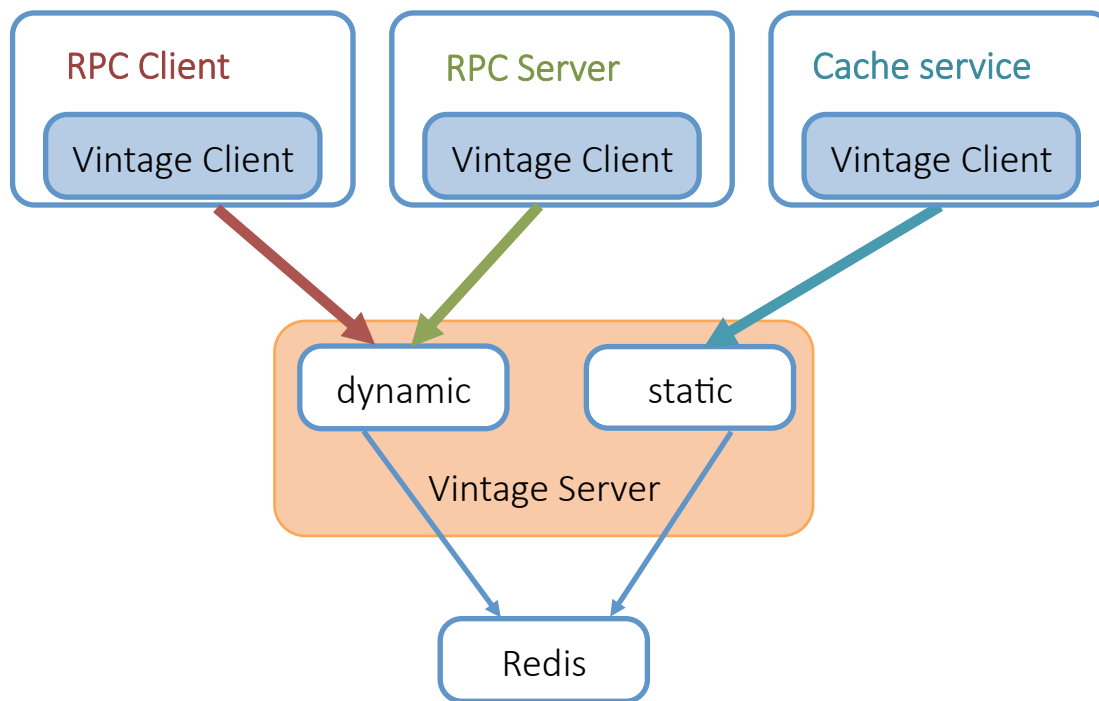
基于Motan的服务化框架

# 服务交互方式

- RESTful API
- Vintage + Motan
- Vintage + CacheService



Motan RPC 交互流程



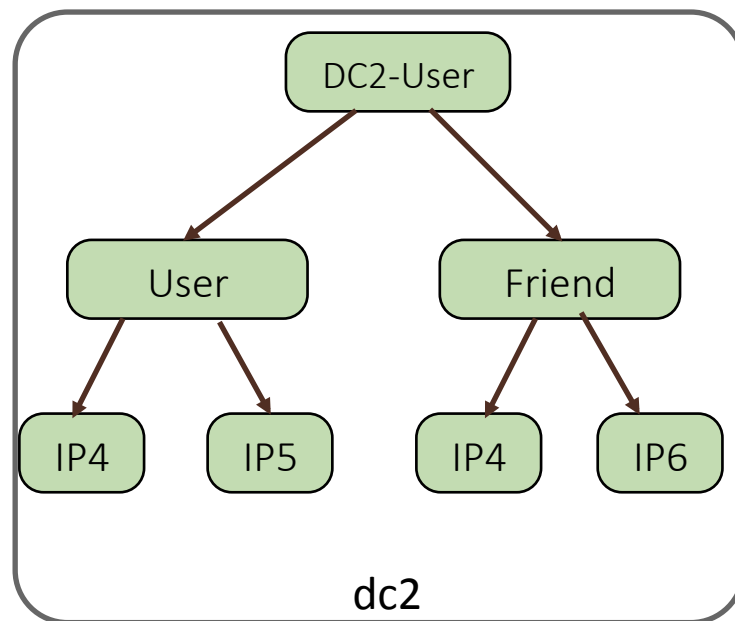
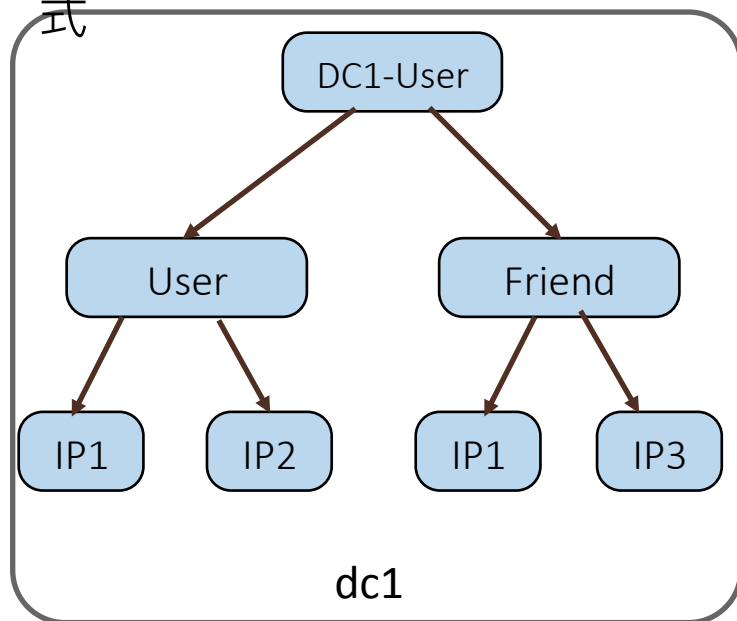
Vintage注册中心

# Vintage优化措施

- 减少网络风暴
  - 签名查询
  - 变更合并
  - 数据压缩
- 网络异常防护
  - 服务摘除防护
- 注册中心异常防护
  - 本地服务快照

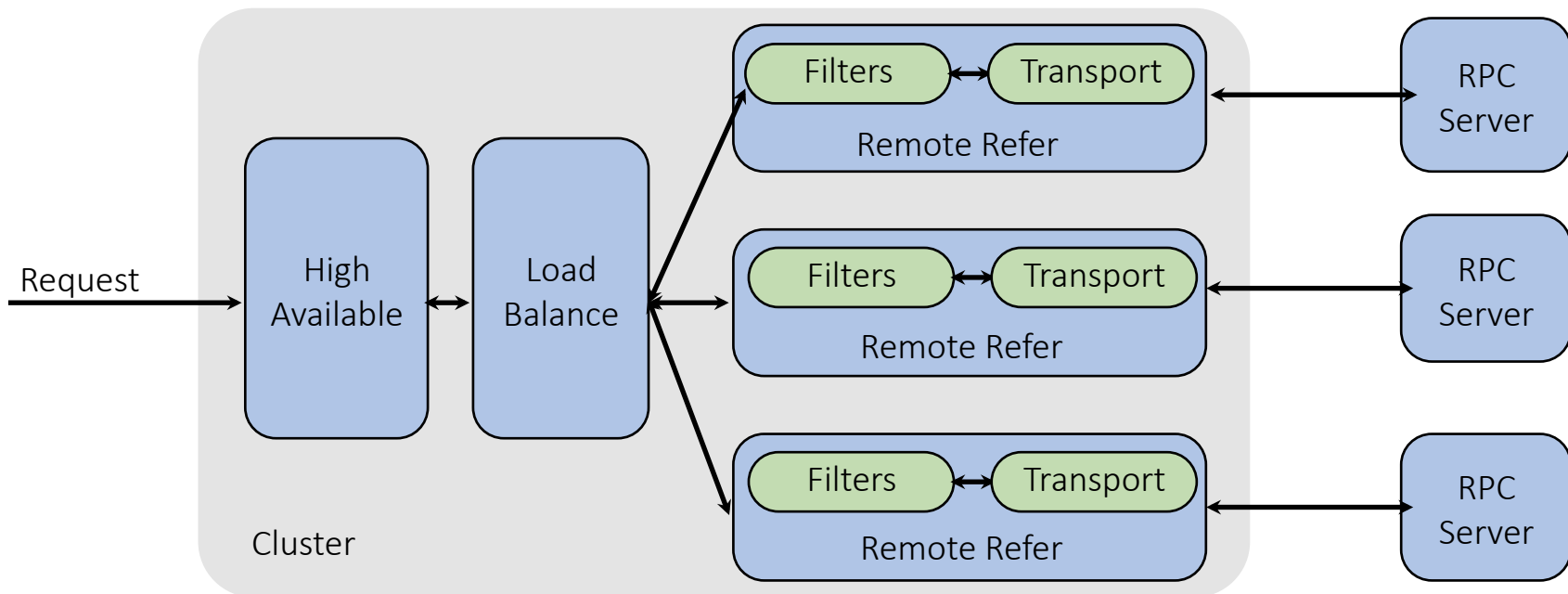
## Motan服务化服务管理方

式



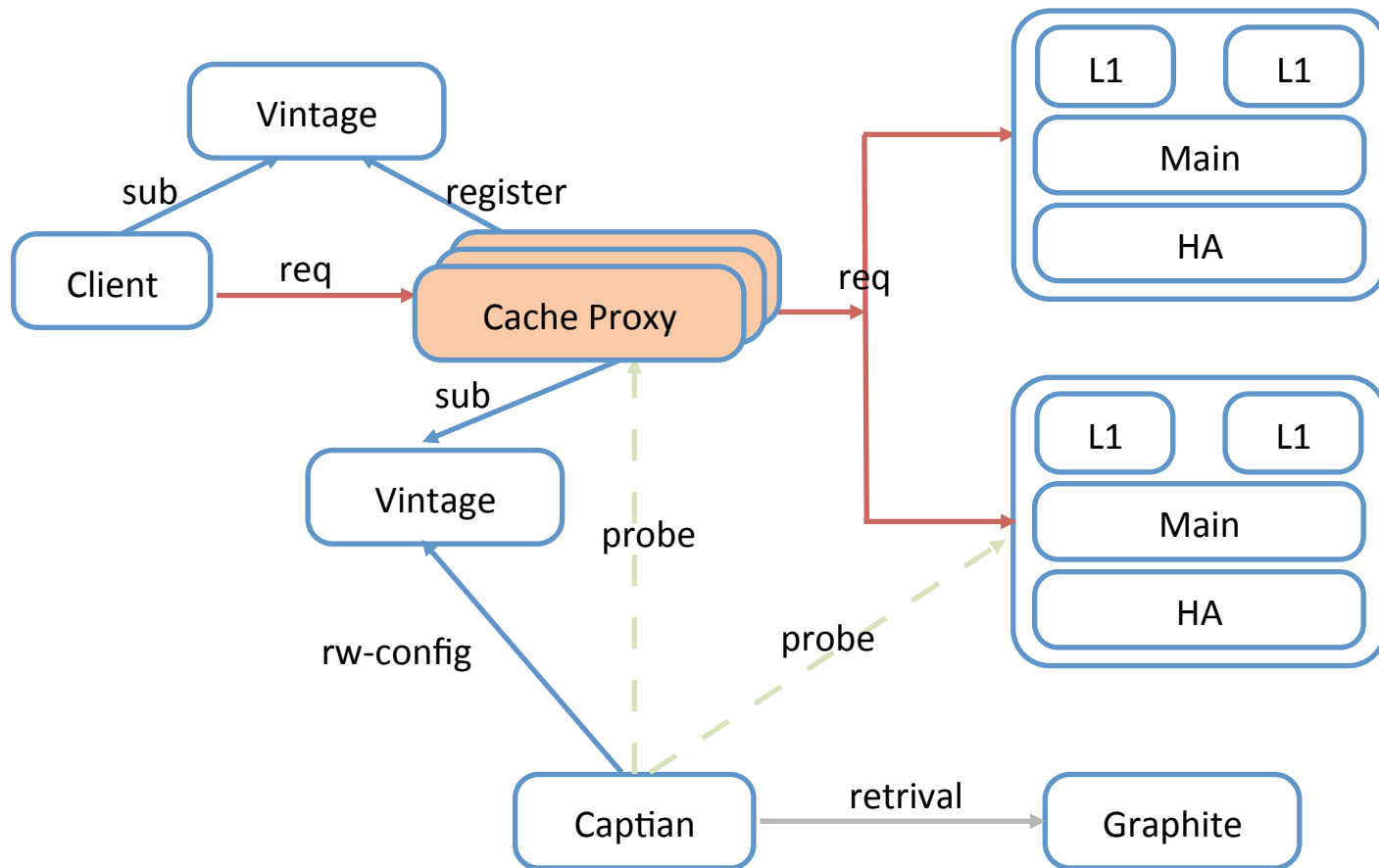
- 服务分组：机房 + 业务线

- 就近访问；服务隔离

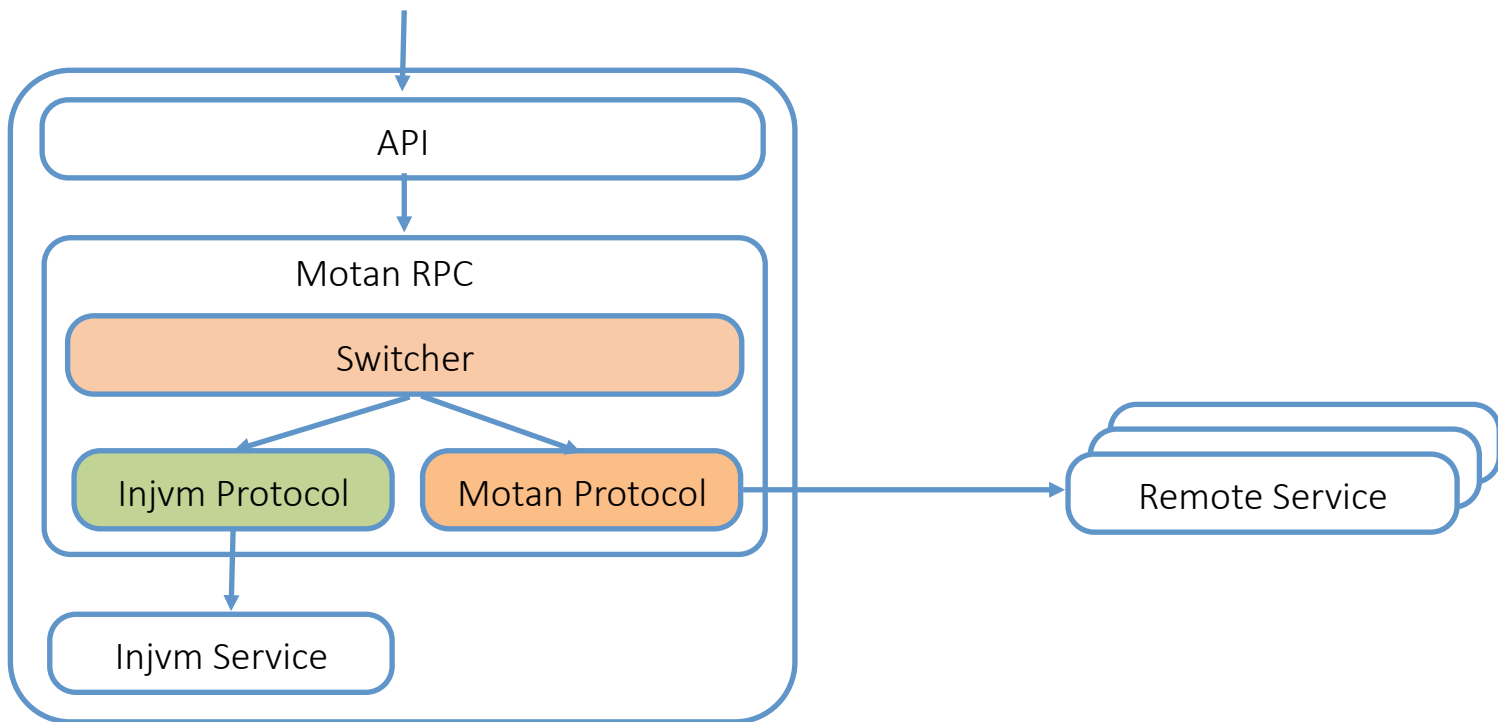


Motan的服务发现与负载均衡





资源服务化—CacheService



如何进行RPC服务化升级

# 小结

## 优点：

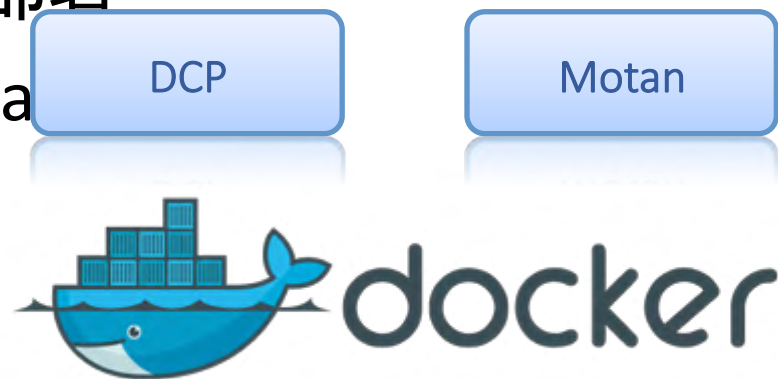
- 服务解耦
- 服务治理

## 缺点：

- 服务扩缩容较慢
- 服务粒度相对较粗

# 混合云微服务阶段

- 时间：2015年
- 背景：容器化服务部署、私有云体系快速发展
- 服务部署： Docker容器化部署
- 服务交互： API + Motan + Ya



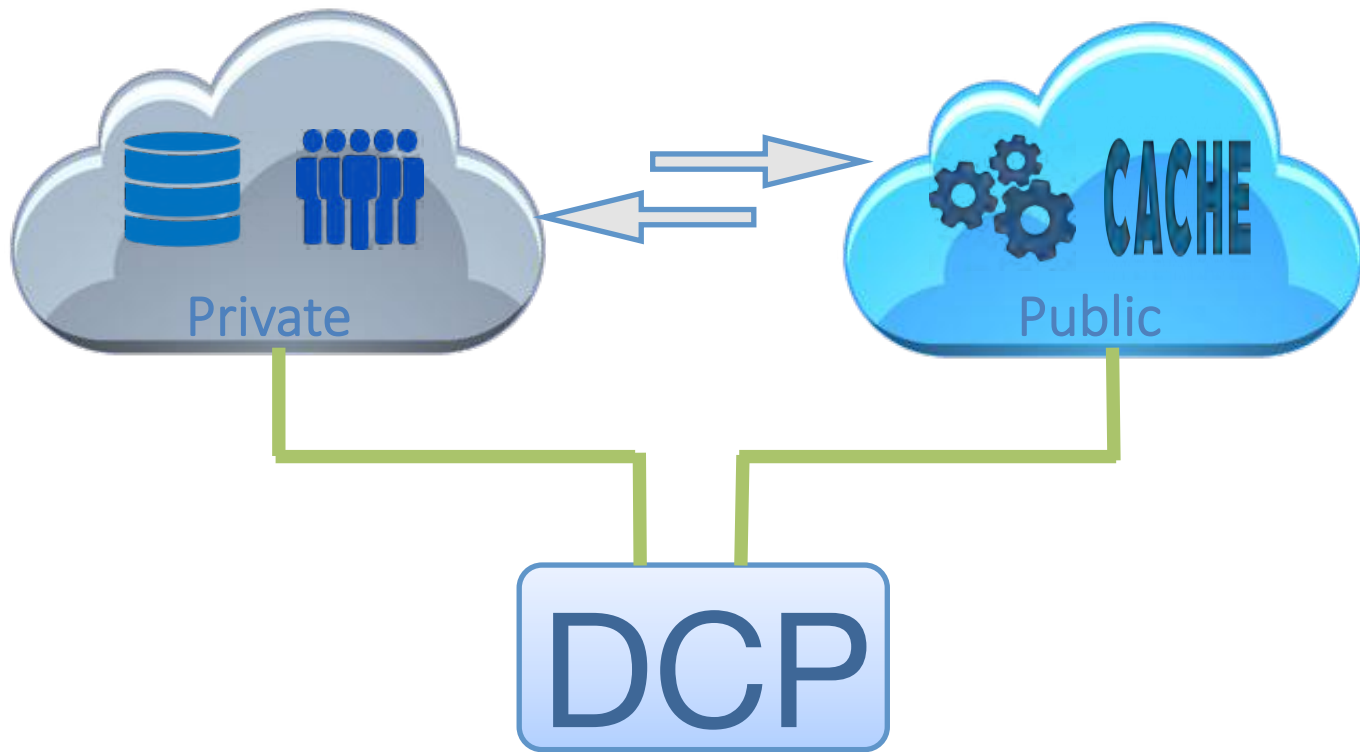
# DCP混合云平台简介

## DCP(Docker Container Platfrom)

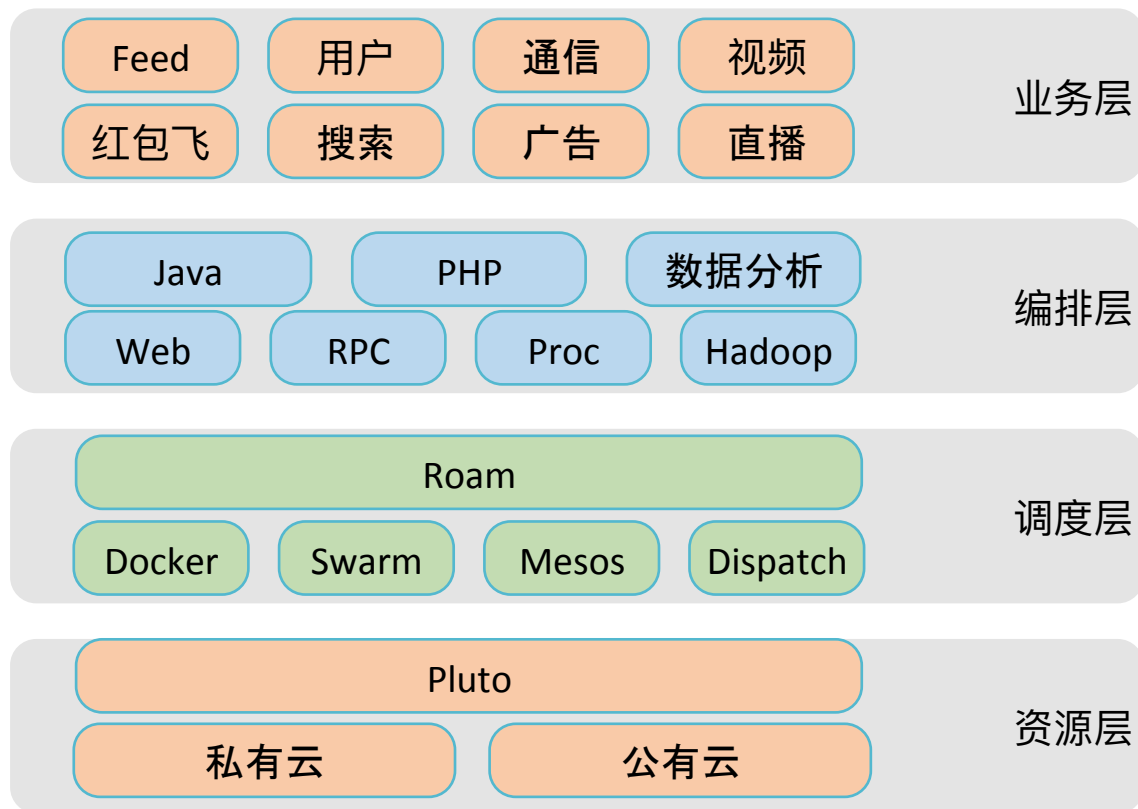
- 10分钟扩容1000+节点
- 每天600亿次的API调用
- 每天万亿次的RPC调用

## DCP特点：

- 私有云+公有云的混合云架构
- 弹性服务管理
- 服务监控
- 自动运维

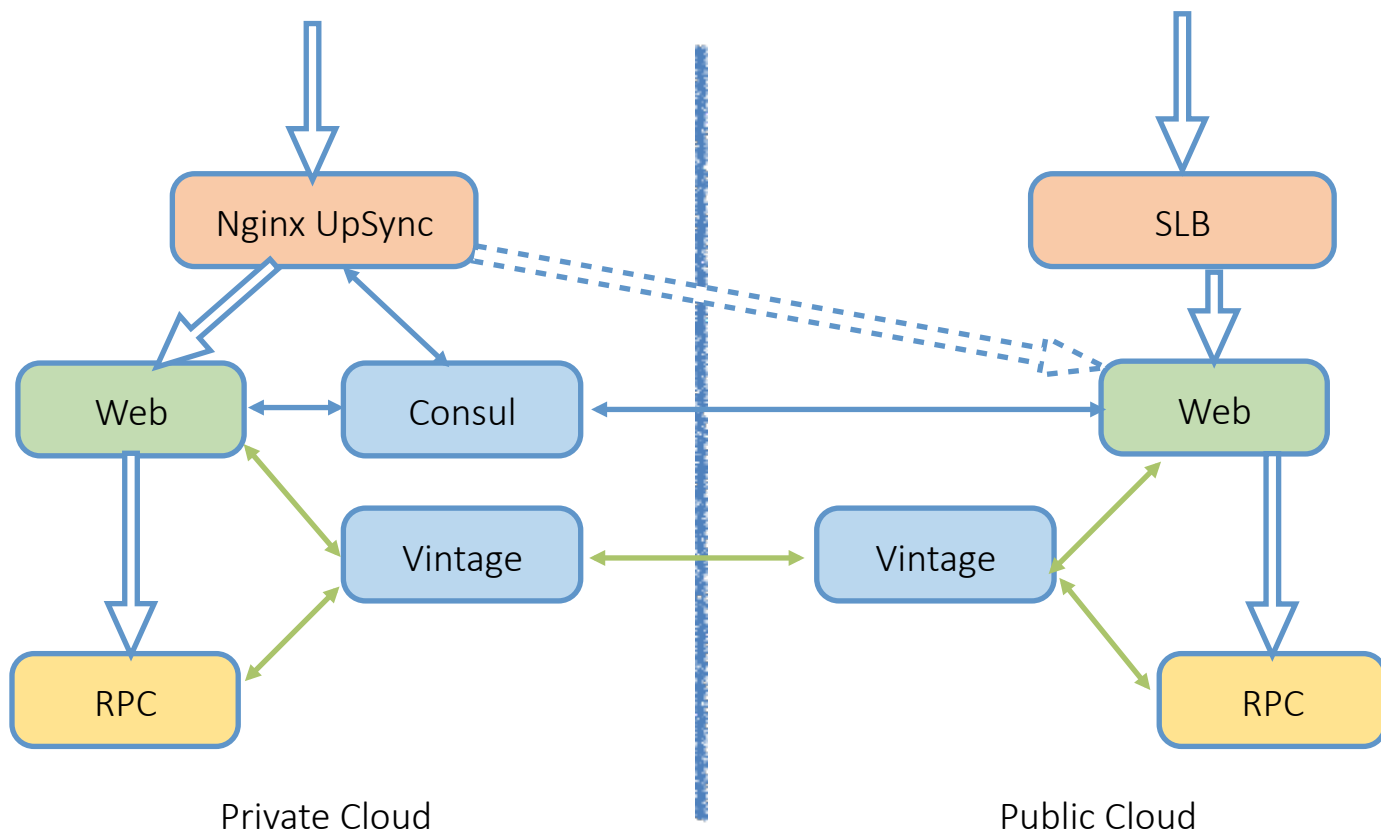


DCP混合云架构



DCP功能模块





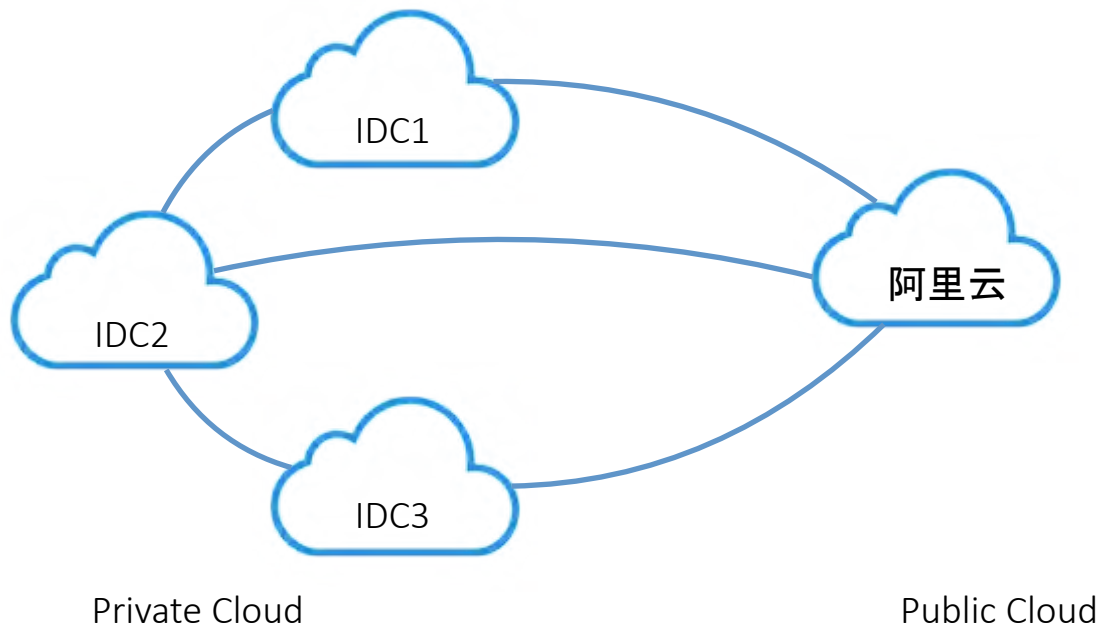
混合云服务交互流程

# 混合云服务化过程中的问题

- 跨云通信问题
- Web服务发现
- RPC流量控制与压缩
- 跨语言交互

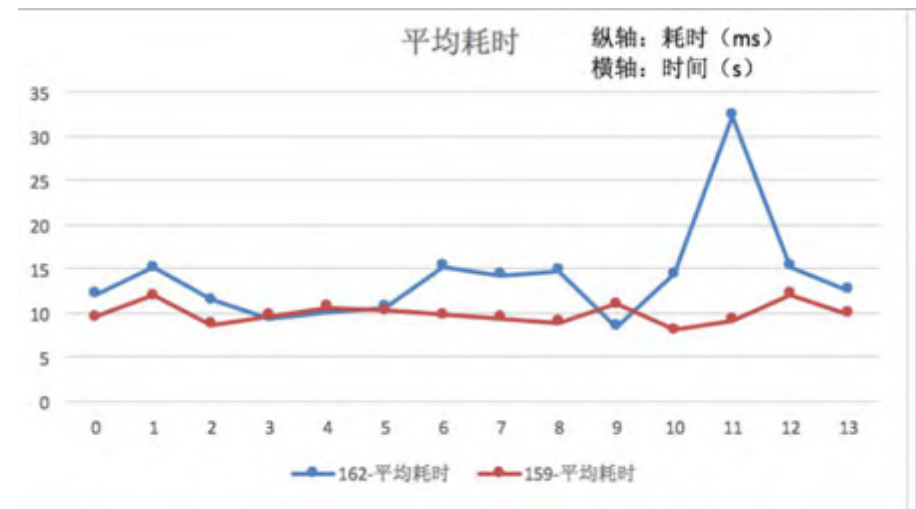
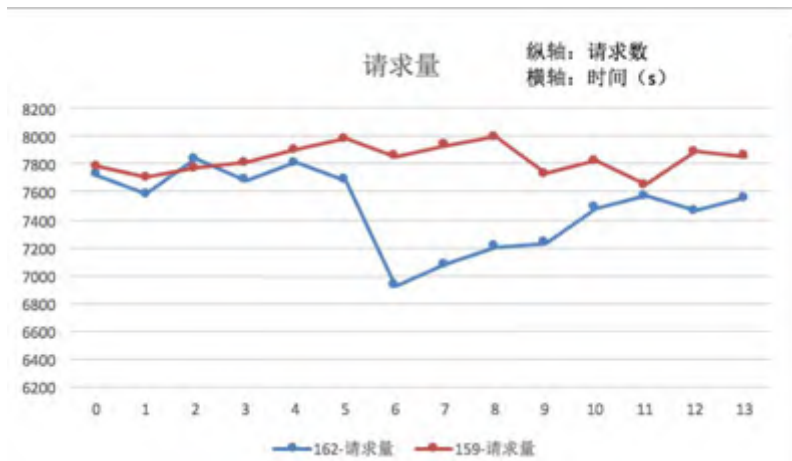
## 跨云通信问题

- 专线通信
- 多机房、多专线容灾



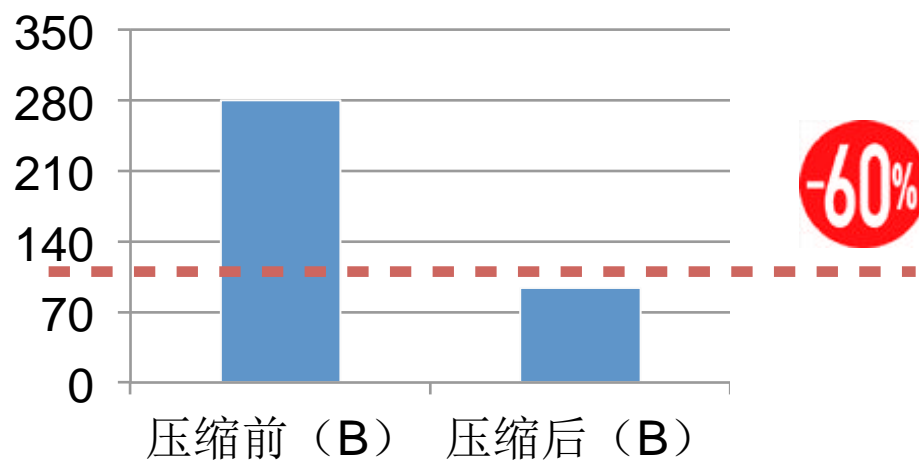
## 混合云服务发现

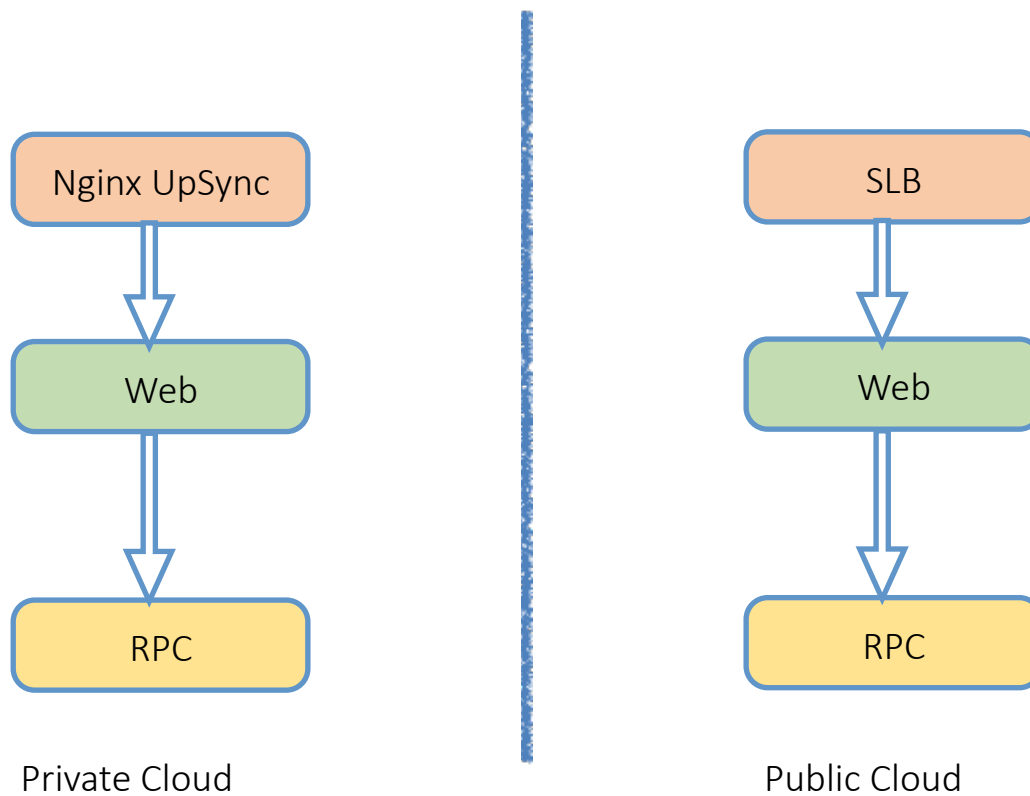
- Vintage云端部署
- Nginx Upsync模块HTTP服务动态发现



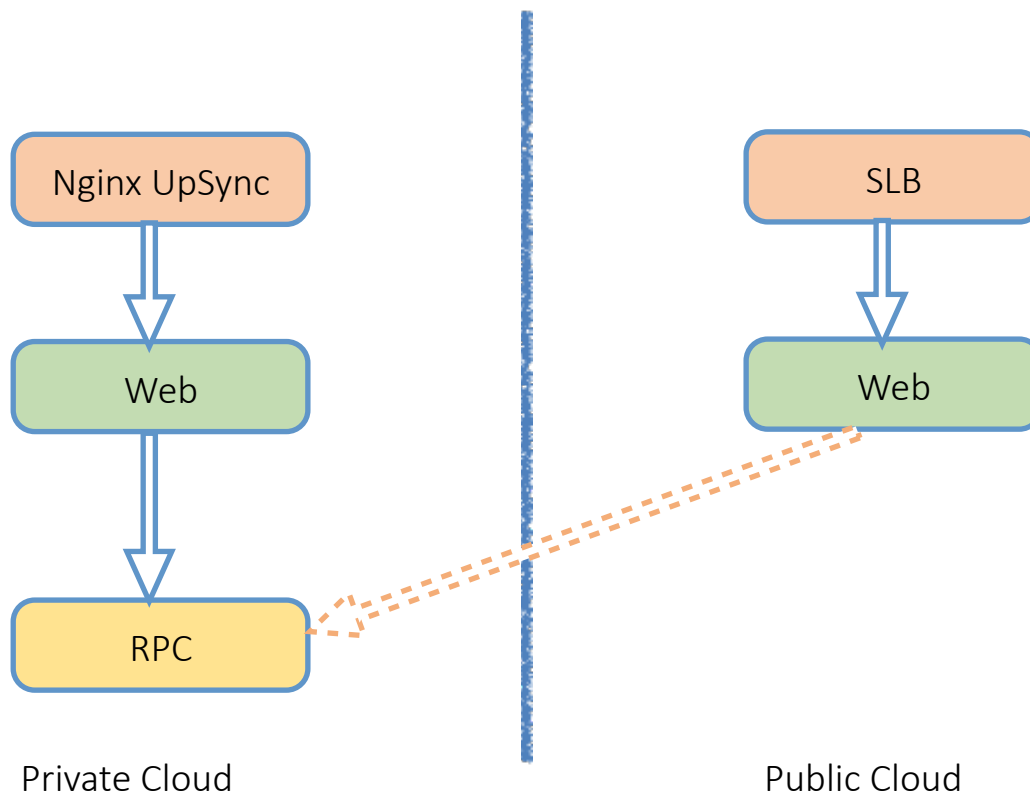
## RPC流量压缩—压缩协议

- 方法签名
- 固定信息签名
- 结果数据压缩

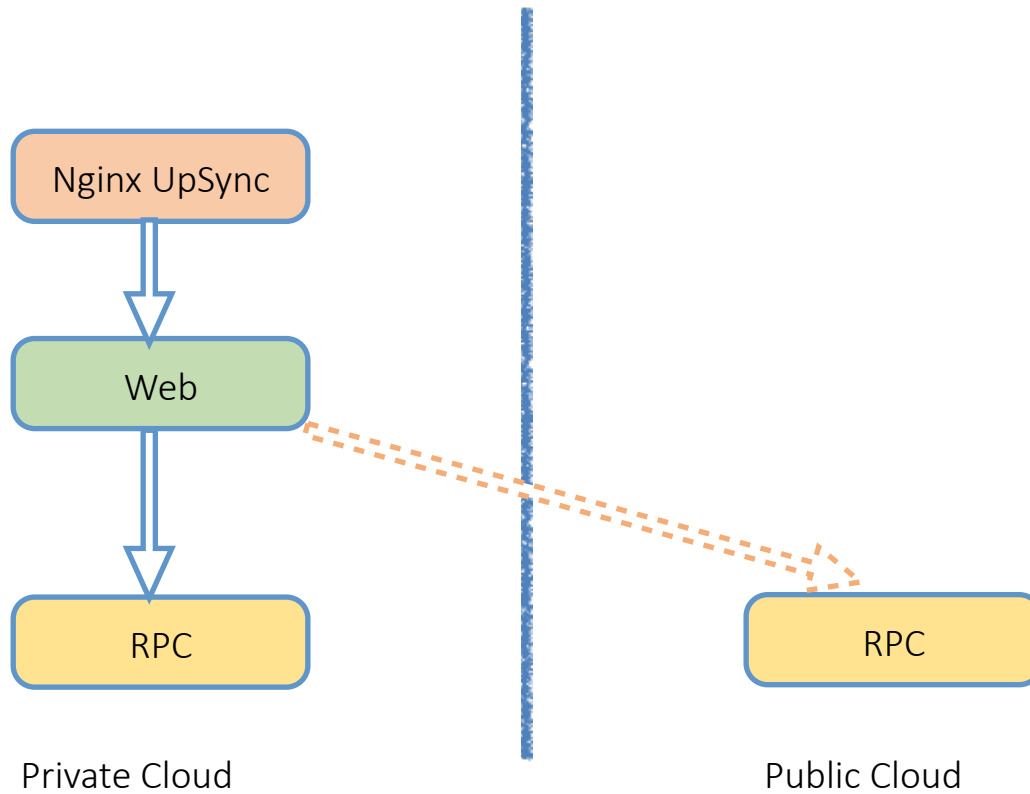




混合云服务扩容方式

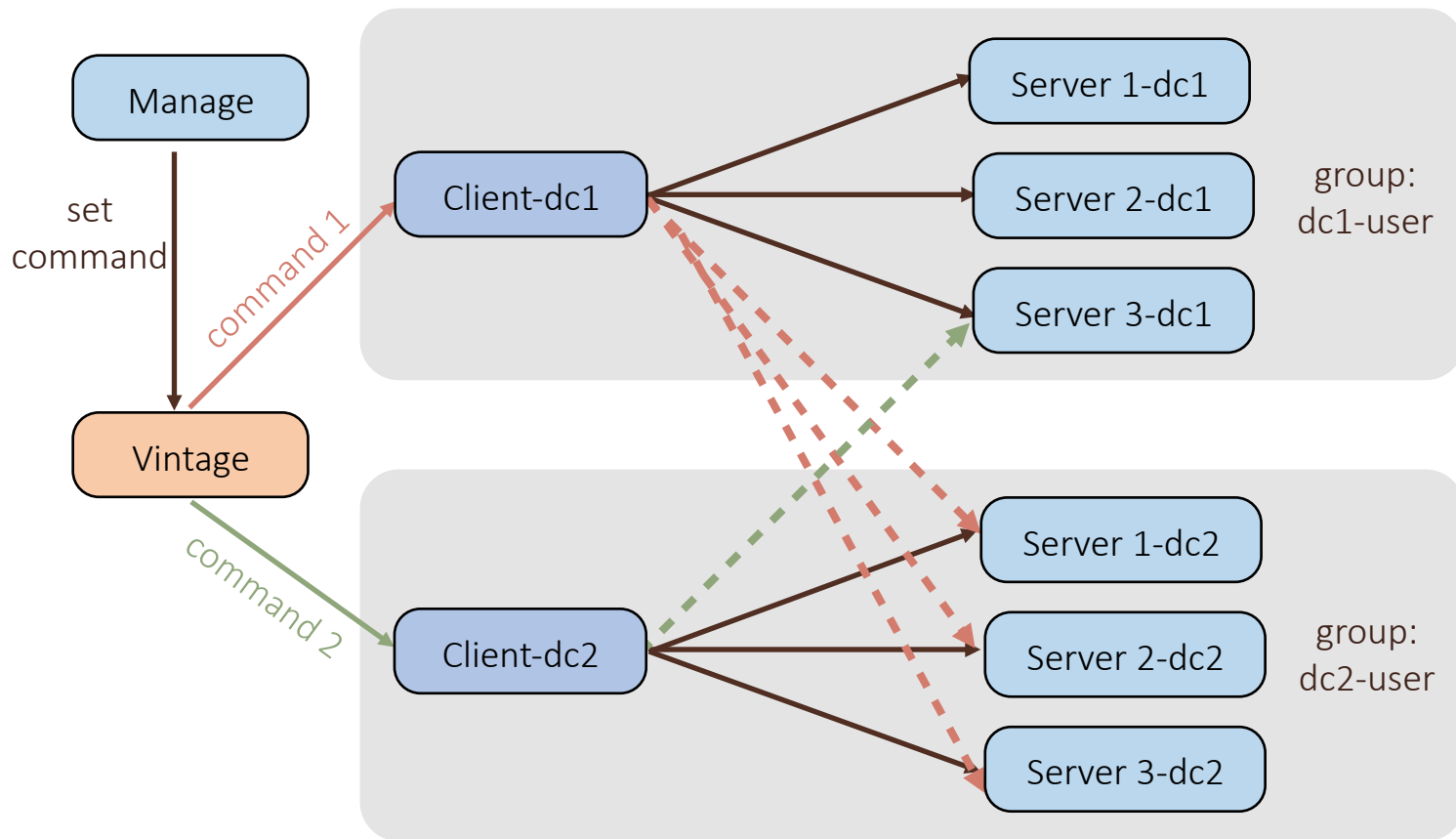


混合云服务扩容方式

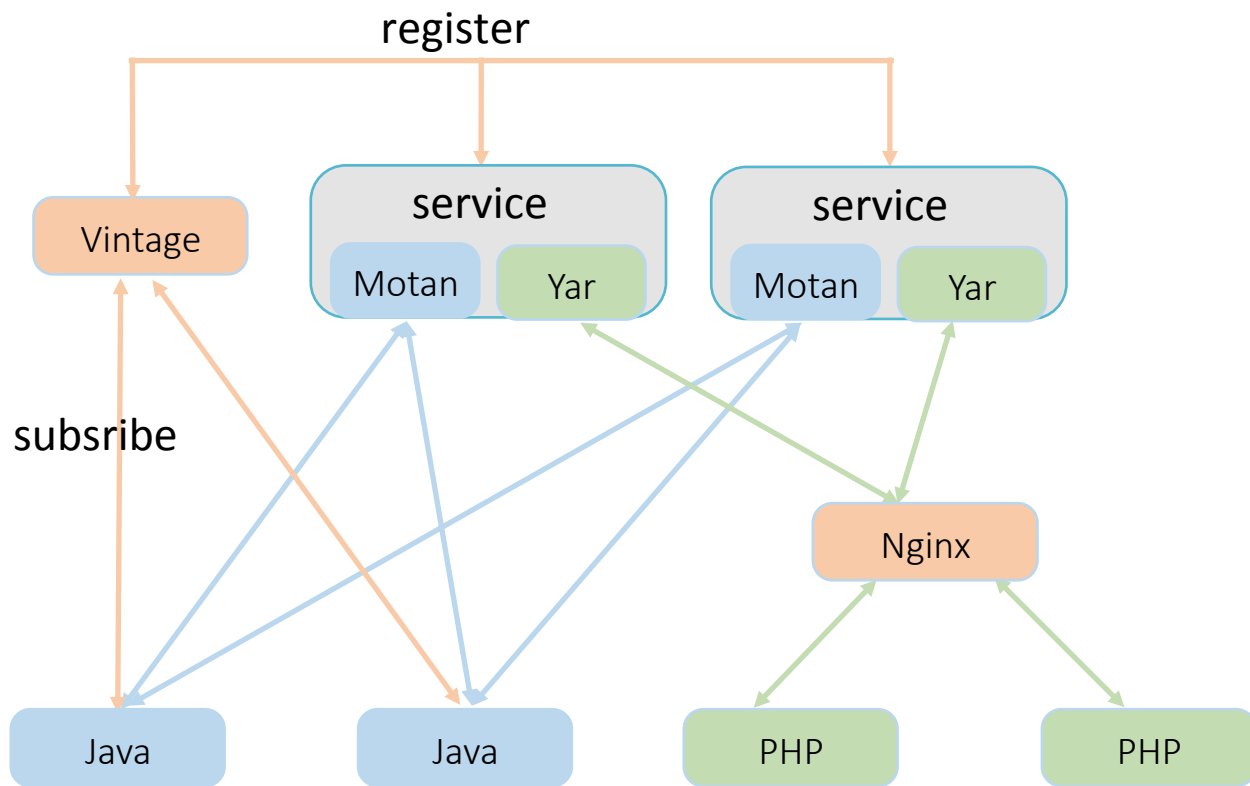


混合云服务扩容方式





流量控制与指令系统



## Motan多协议交互



- 跨语言调用—gRPC
- 服务治理标准化—Motan Agent

# 总结

- 第一阶段：资源利用最大化、服务防御
- 第二阶段：解耦、按需扩容、基础服务治理
- 第三阶段：动态扩容、弹性调度、无损抗峰
- 下一阶段：跨语言服务治理

Thank you !