

# 微博Service Mesh实践

— Weibo Mesh

---

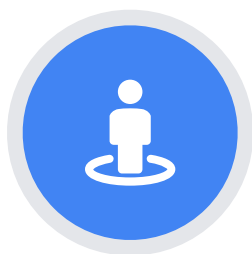
微博搜索/丁振凯

2018.07.29



# 内容提要

CONTENTS



## 1. 跨语言服务化的必要性及难点



## 3. WeiboMesh未来发展规划



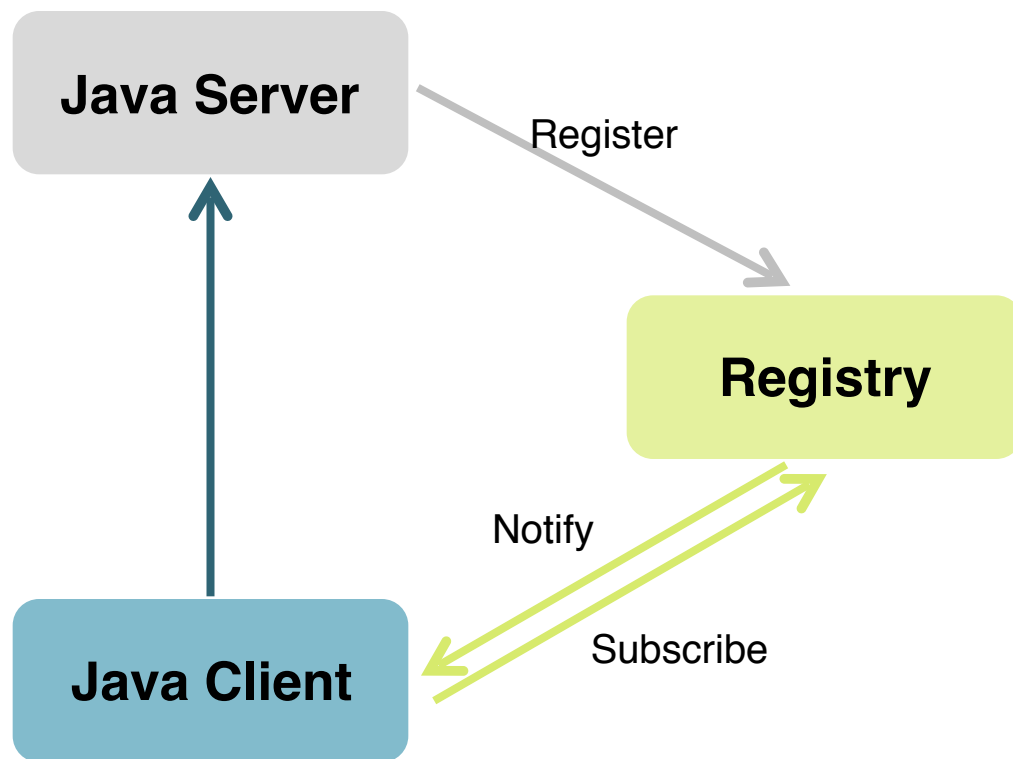
## 2. WeiboMesh方案介绍

# 1

## 为什么要做跨语言服务化

- 需求
- 趋势

# 平台体系



## 平台微服务相关建设比较完善

### Motan

- 服务治理
- 动态路由

### Vintage

- 注册中心

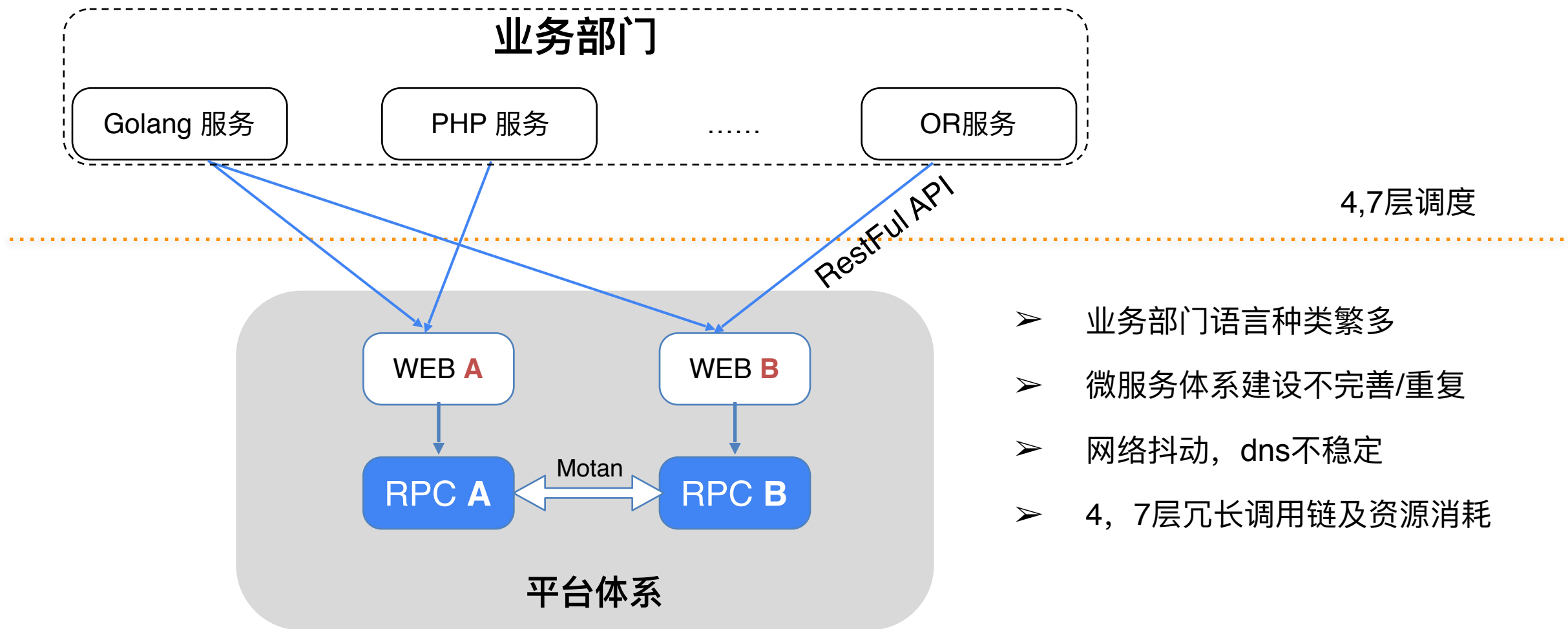
### Opendcp

- 智能弹性调度

### Graphite

- 实时统计监控

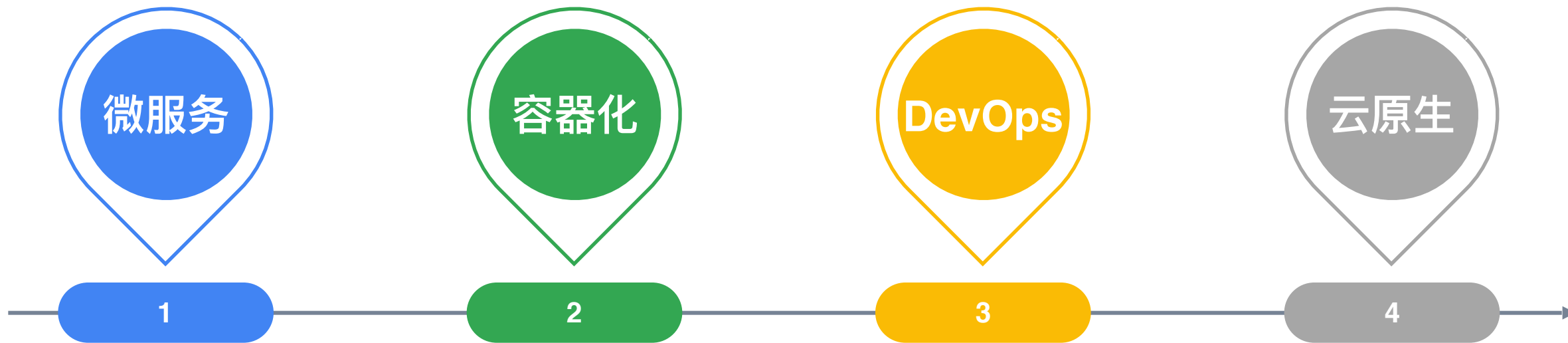
# 业务部门调用链



- 业务部门语言种类繁多
- 微服务体系建设不完善/重复
- 网络抖动, dns不稳定
- 4, 7层冗长调用链及资源消耗

# 趋势

---



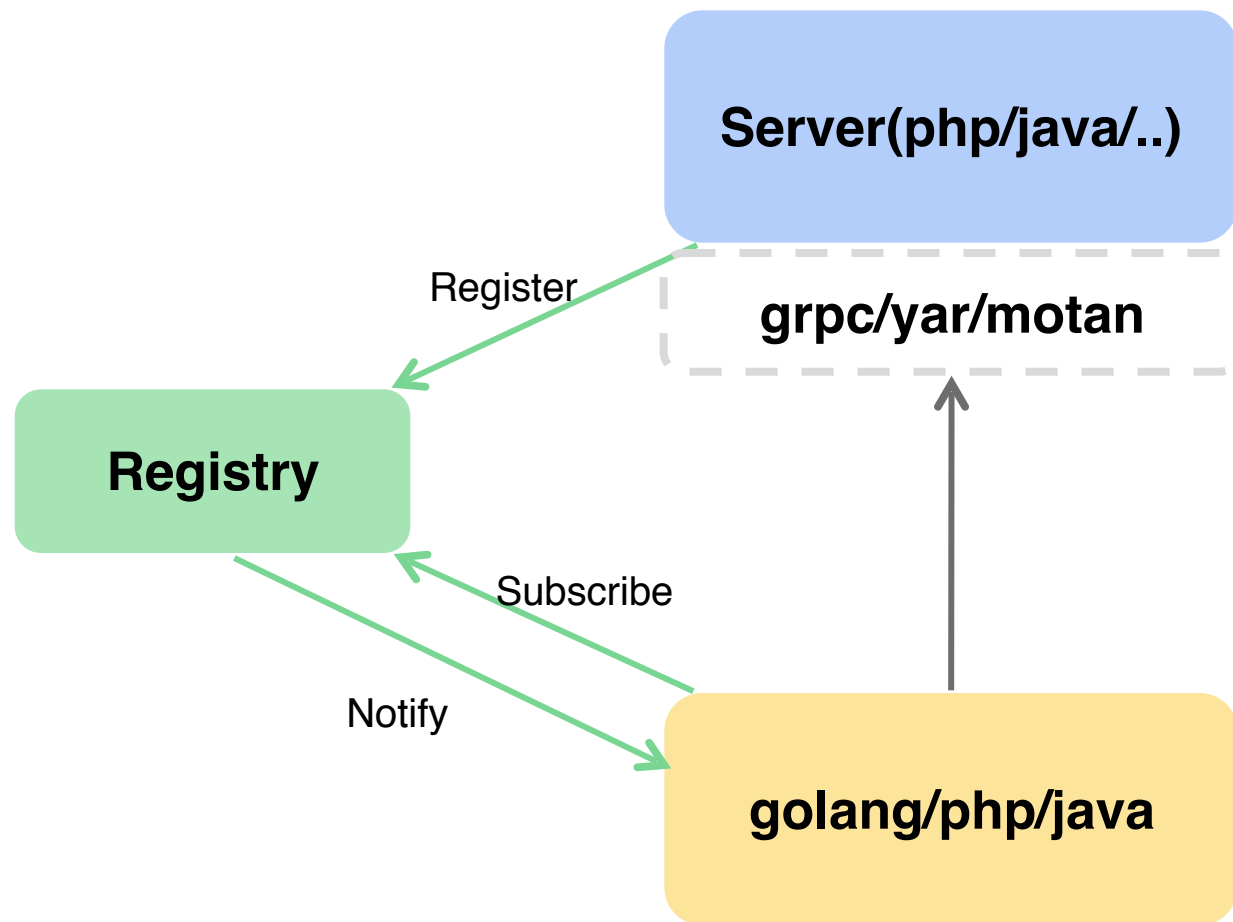
服务治理与业务逻辑解耦，可持续交付

# 2

## 跨语言服务化面临的问题

- 改造成本
- 服务治理

# 改造成本



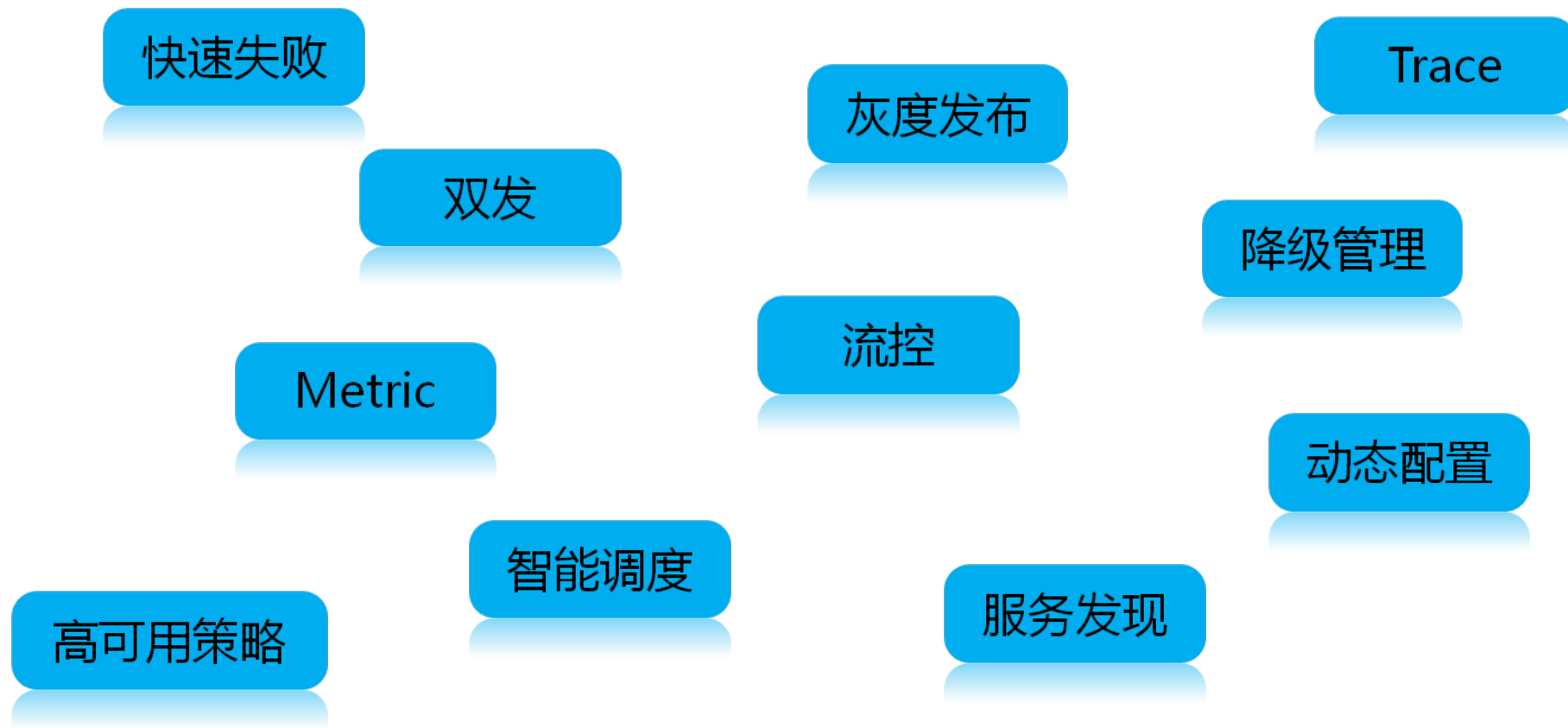
- 语言特性
- 历史积累
- 业务侵入较大, client太重
- 性能
- 扩展性差
- 推广困难



# 服务治理

---

相同的治理功能，不同语言的服务都要做一遍？

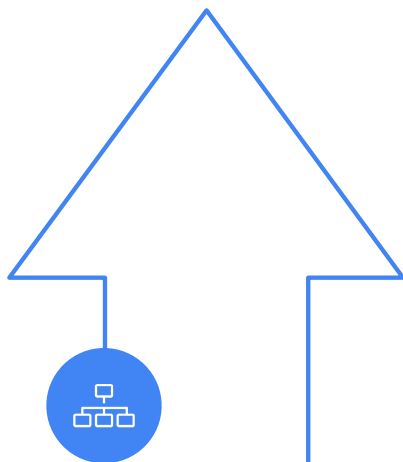


# 跨语言服务化的本质

---

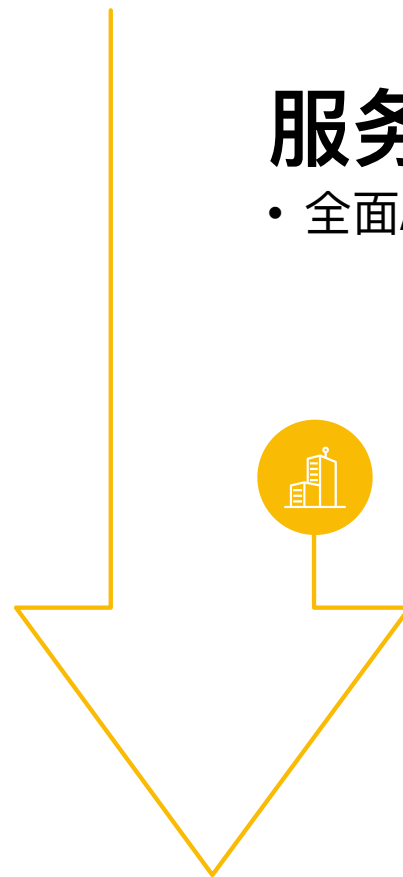
- 

**数据交互**  
协议中立/跨语言



**服务治理**

- 全面/灵活可扩展

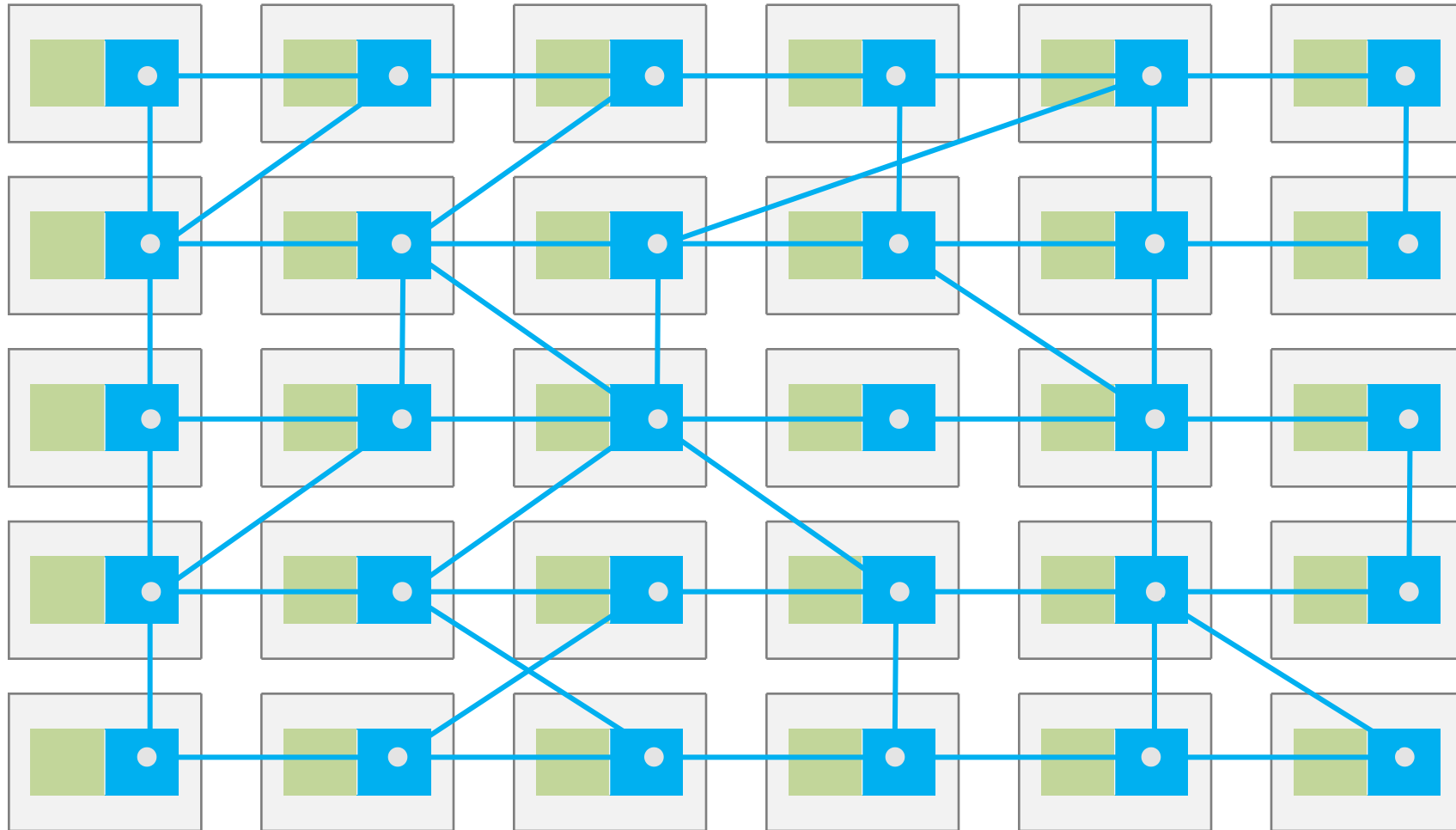


# 跨语言服务化方式对比

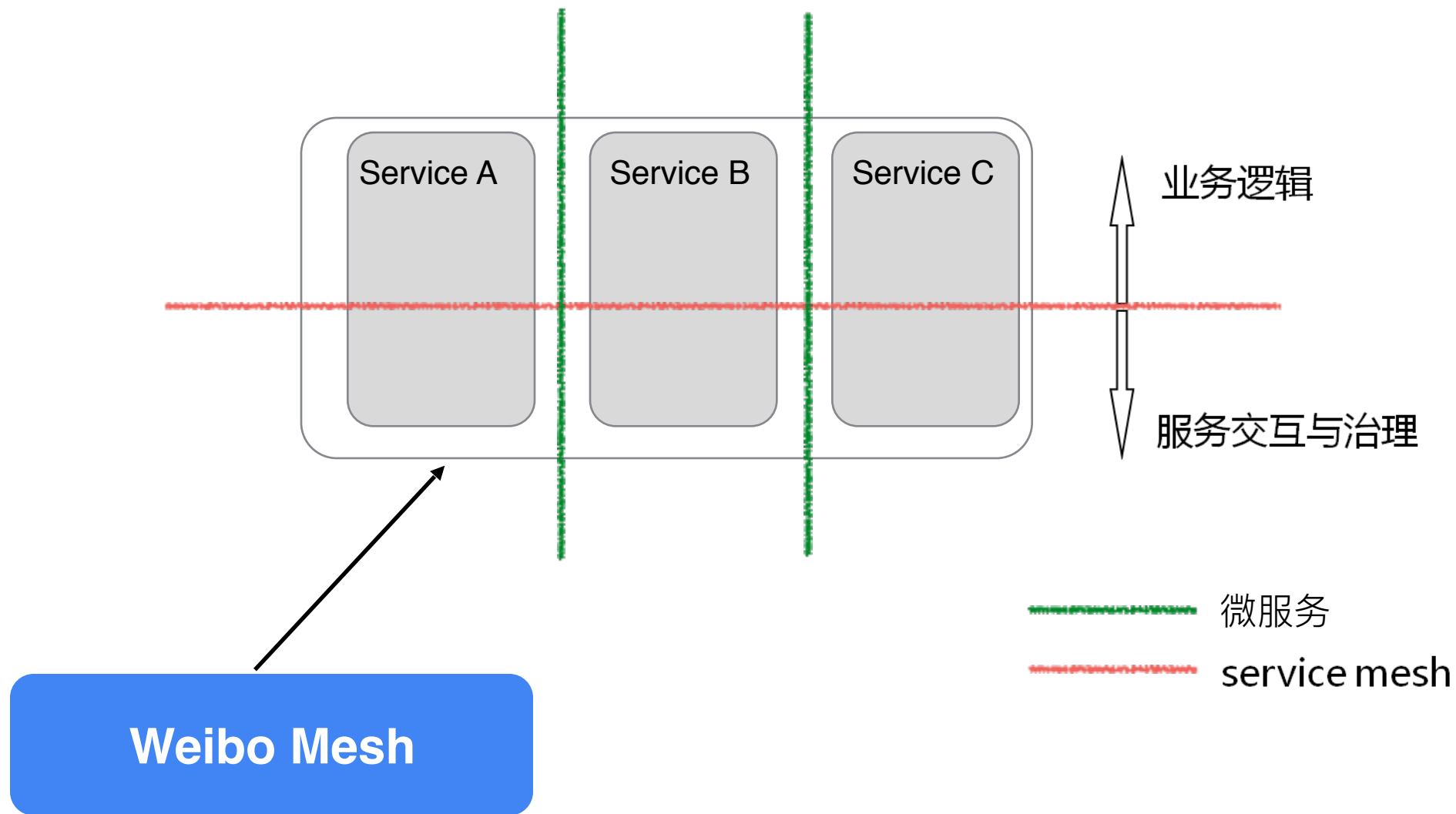
	Http代理	RPC模块	Agent代理
研发成本	低	高	中
维护成本	低	高	中
使用成本	低	低	中
治理功能	中	高	高
扩展能力	低	中	高

# Service Mesh

---



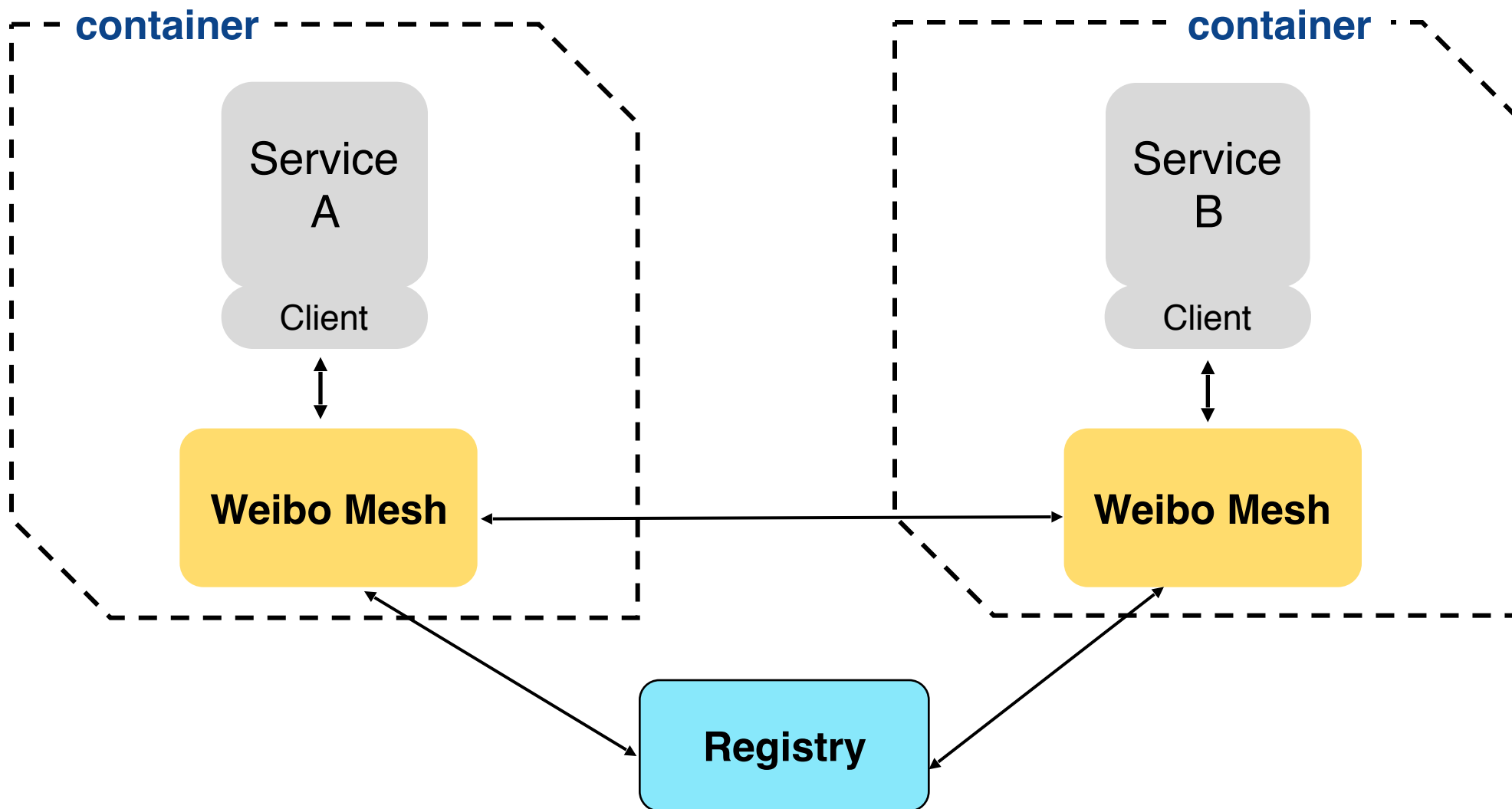
# Service Mesh与微服务的切分

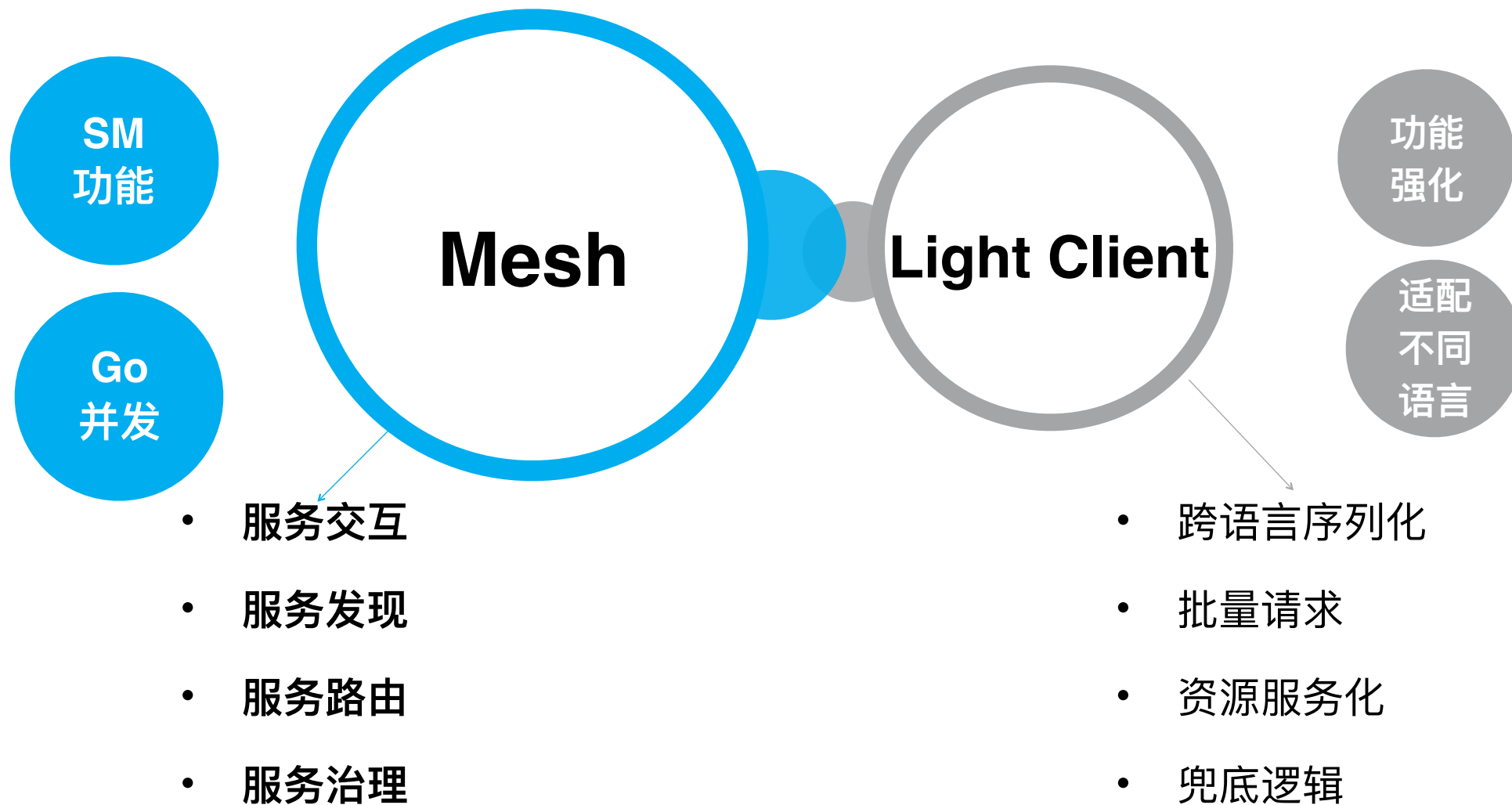


3

## Weibo Mesh方案介绍

# Weibo Mesh 总体架构





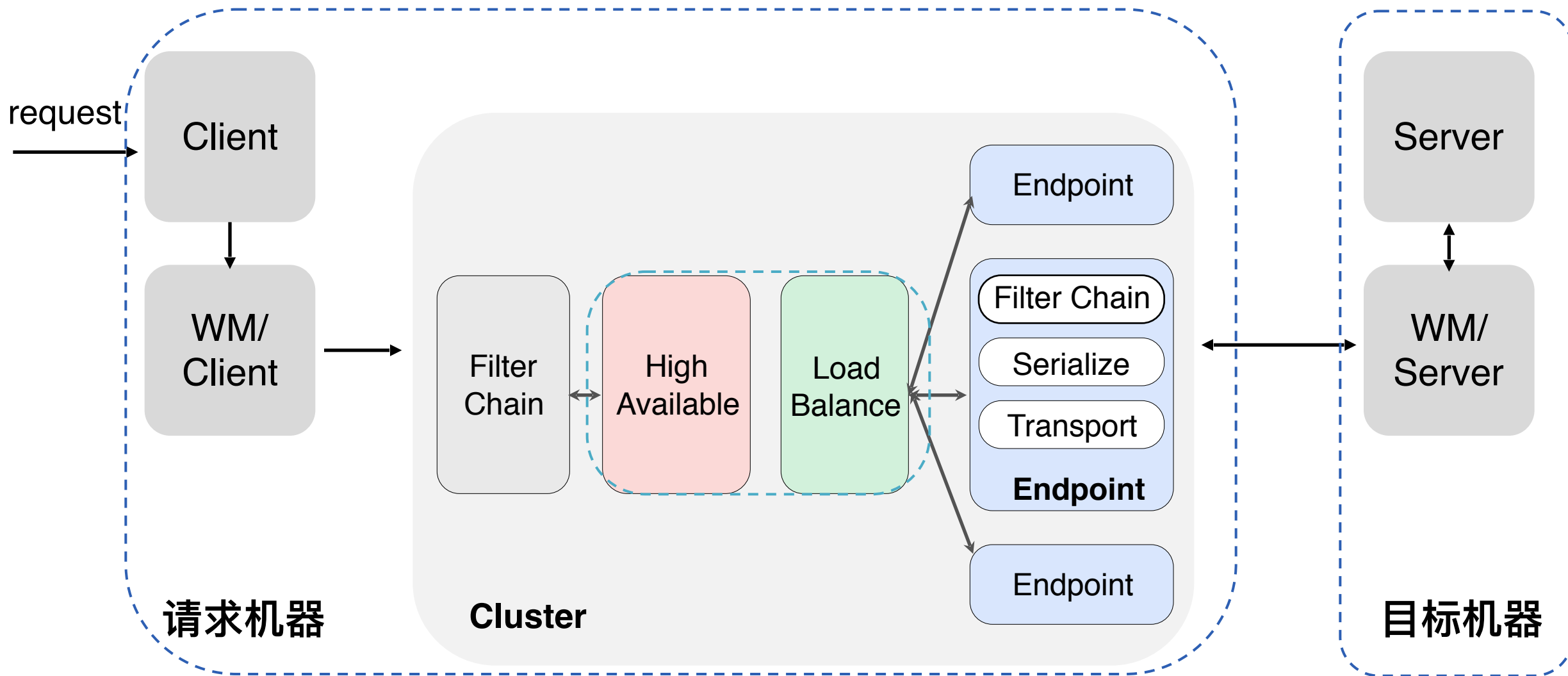


# Weibo Mesh 数据面

---

- Cluster (发现集群管理, group + path)
- HA (高可用策略)
- LB (负载均衡)
- Endpoint (服务节点的抽象)
- Protocol (Motan2/传输协议+Simple/序列化协议)

# Cluster 模块



# 高可用

---

## HA

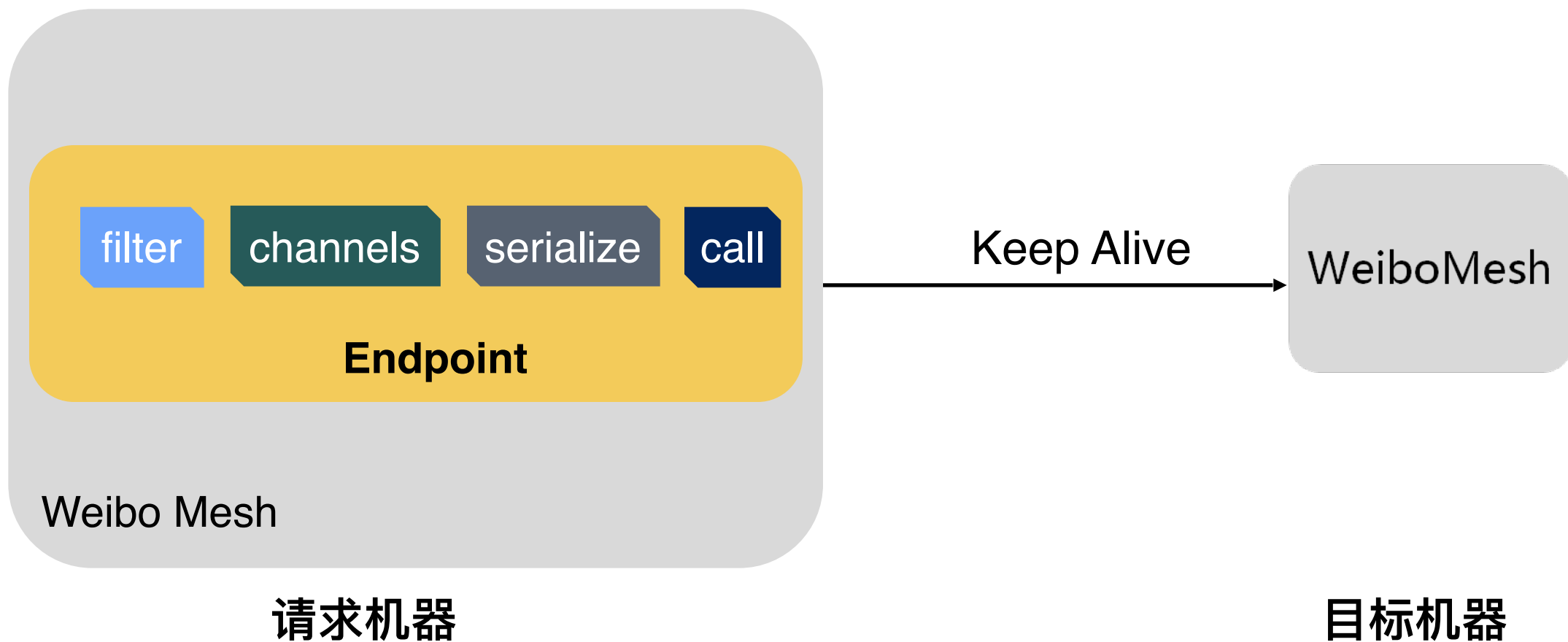
- Backup Request
- Failover
- ...

## LB

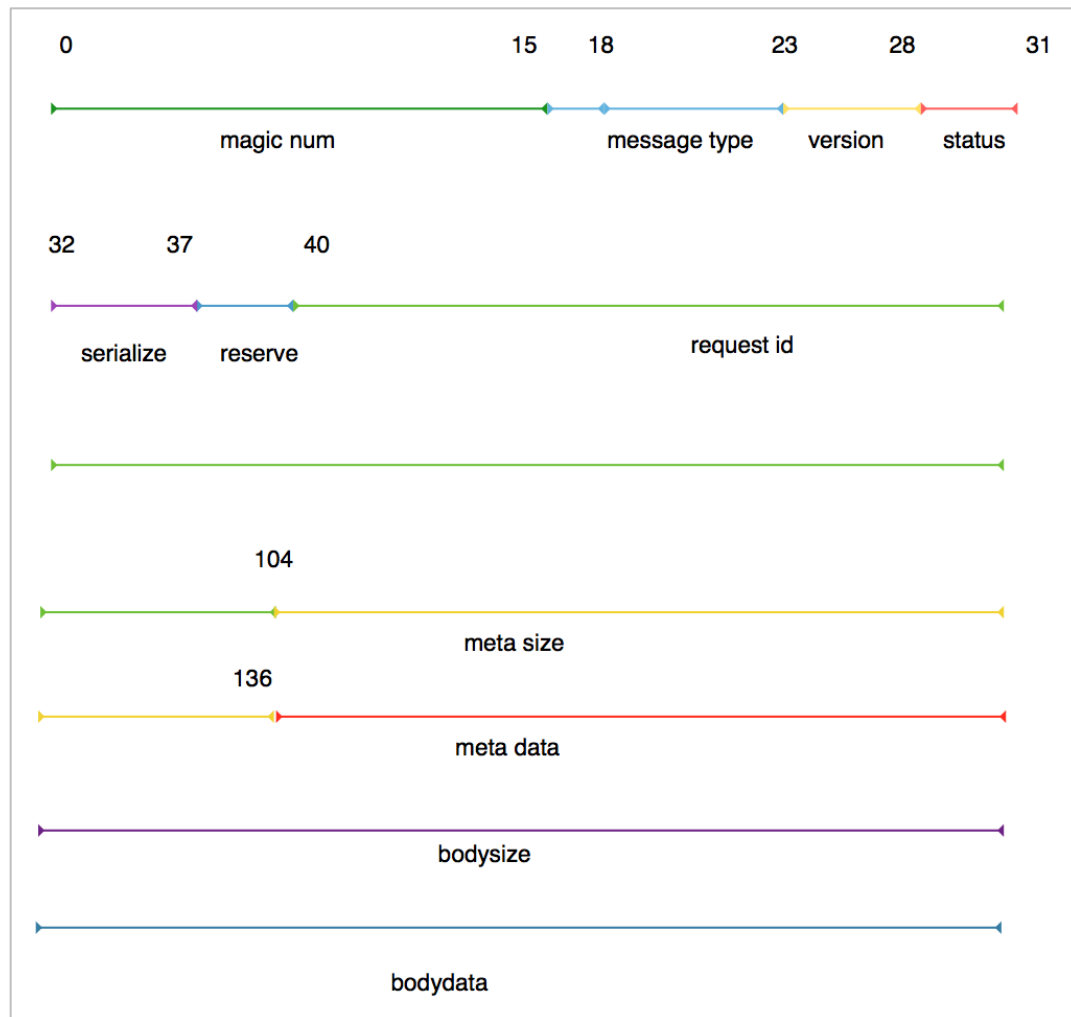
- Weight
- Roundrobin
- Random
- ...

# Endpoint

---



# Motan2 传输协议



## Header

- 消息类型
- 协议版本
- 序列化协议 (body)

## Metadata

- 服务名
- 方法名
- 系统参数及用户参数

## Body

- response
- Request

# Simple 序列化

null  
00

string "hello"  
01 00 00 00 05 68 65 6c 6c 6f

**type(1byte)+size(4byte)+content(\${size} byte)**

map {name:ray, code: xxx}  
02 00 00 00 1e 00 00 00 04 63 6f 64 65 00 00 00 03 78 78 78 00 00 00 04 6e  
61 6d 65 00 00 00 03 72 61 79

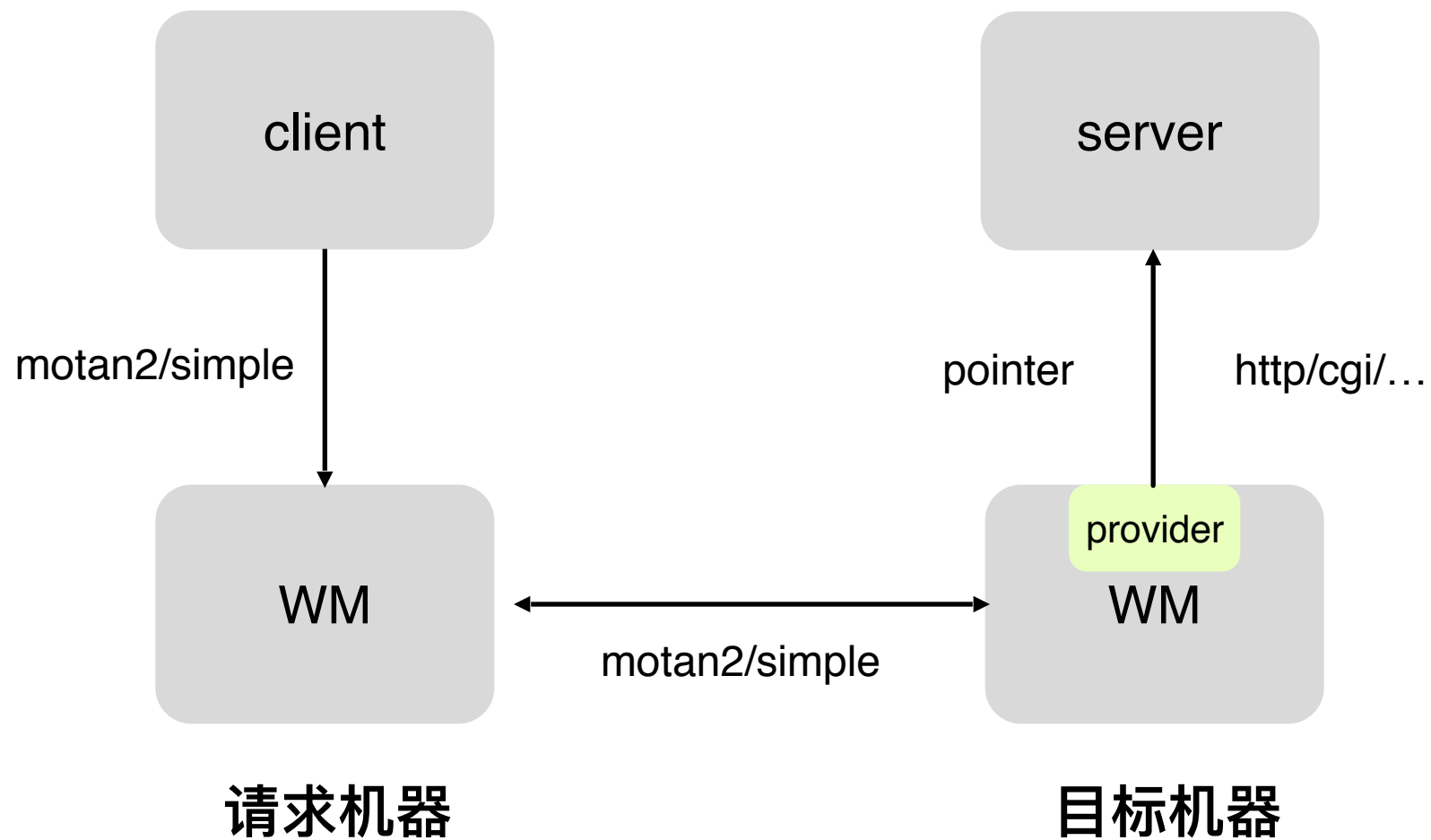
基础类型

复合类型

```
// serialize type
const (
    sNull = iota
    sString
    sStringMap
    sByteArray
    sStringArray
    sBool
    sByte
    sInt16
    sInt32
    sInt64
    sFloat32
    sFloat64

    // [string]interface{}
    sMap    = 20
    sArray = 21
)
```

# 协议投递过程



# Weibo Mesh 控制面

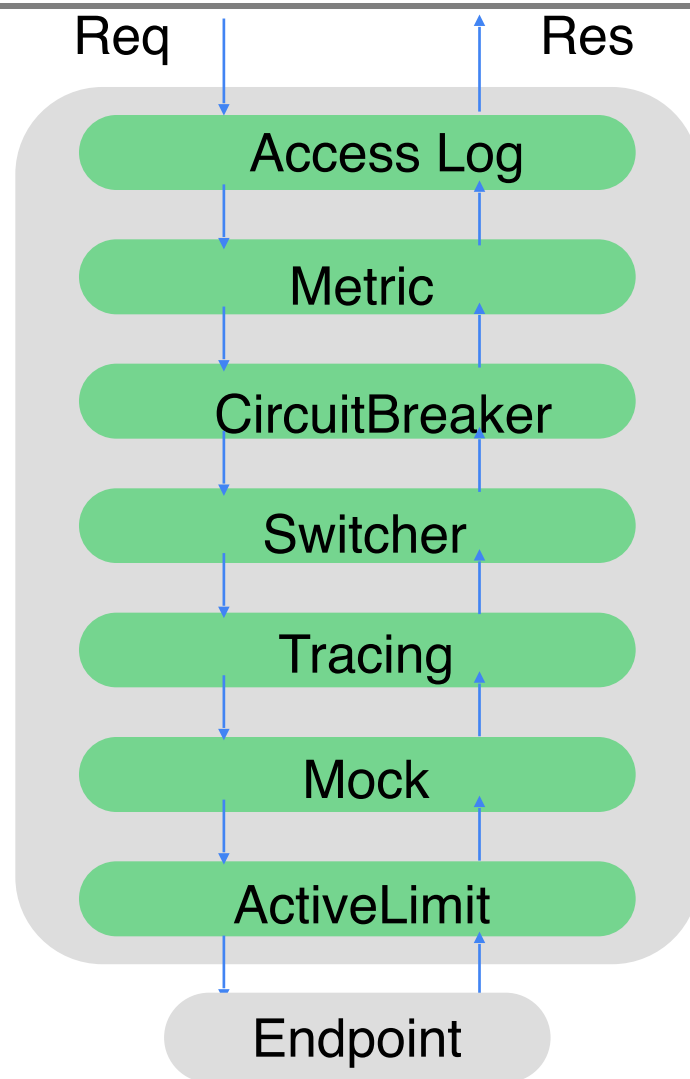
---

- 策略扩展: Filter Chain
- 流量调度: MCS (Mesh Command System)

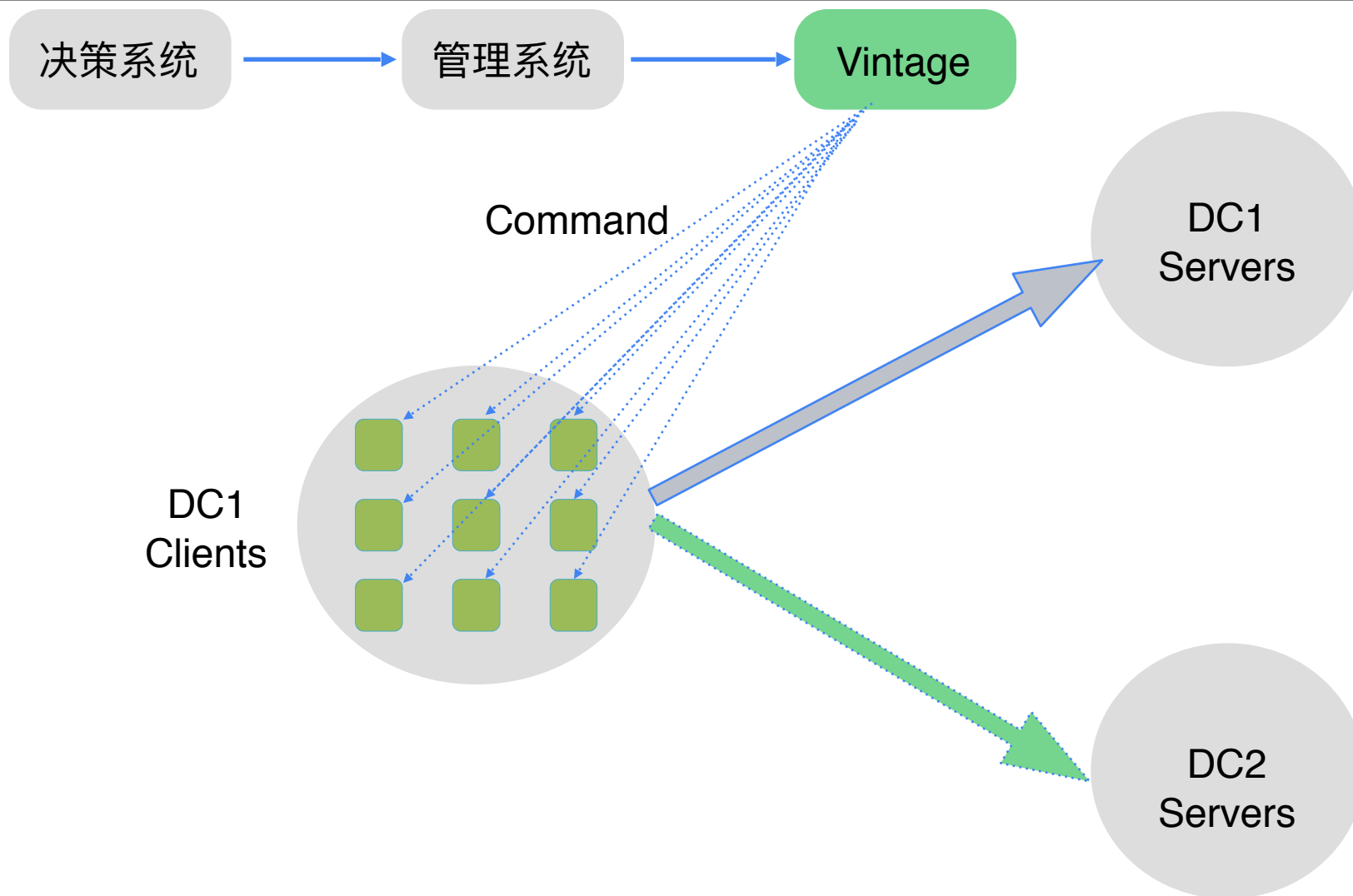


# Motan Filter Chain

插件化



# 基于MCS的流量调度

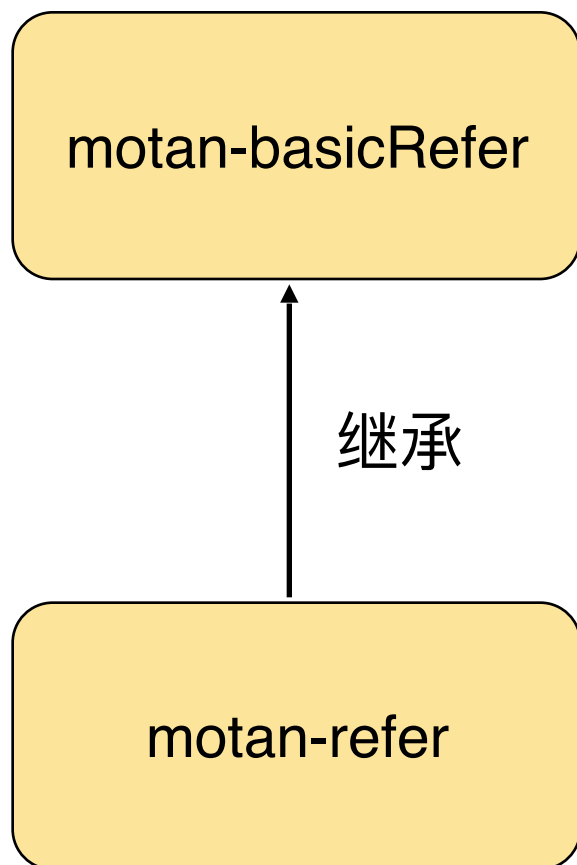


# 4

## 业务实战

- 正反向代理
- 收益及总结

# 正向代理配置

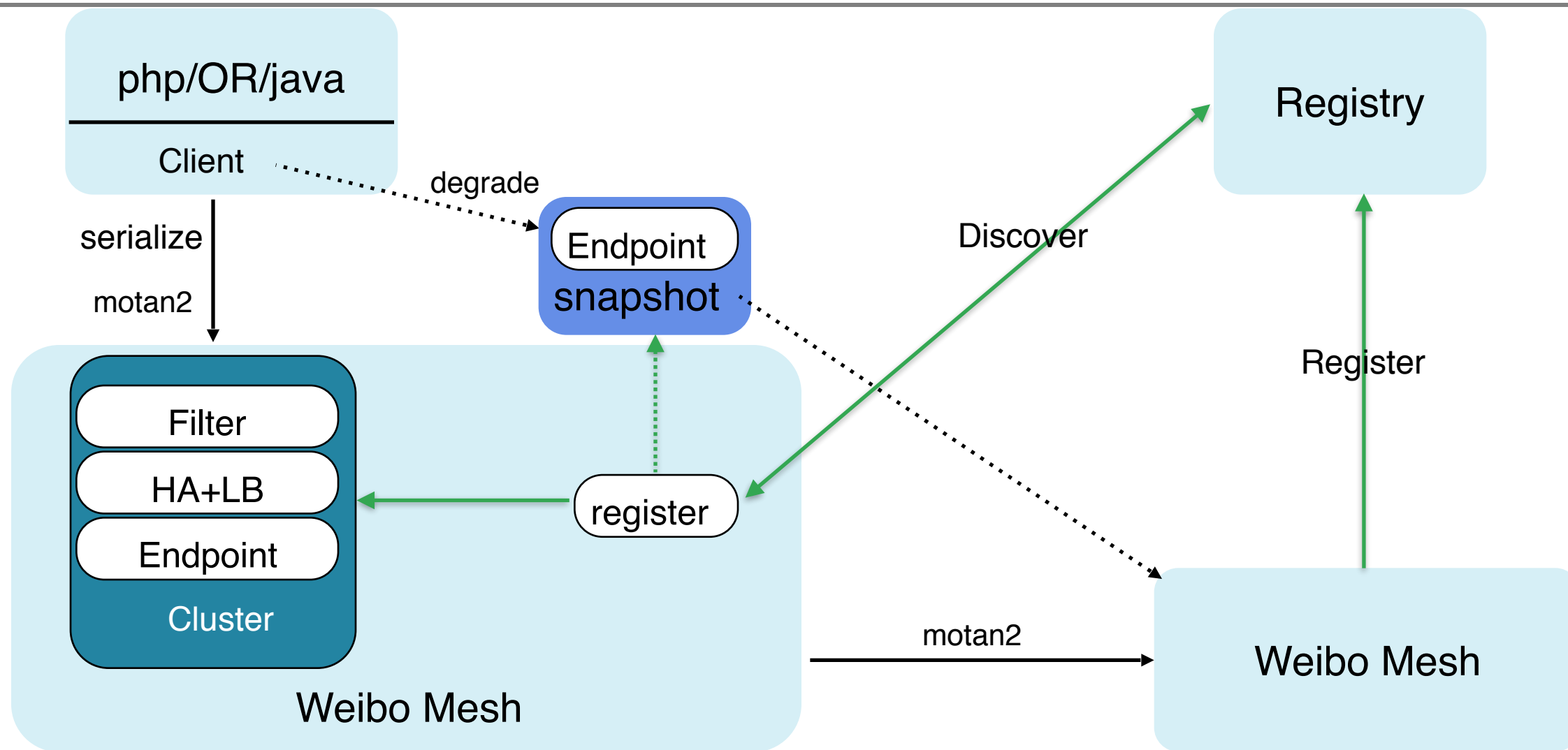


```
#conf of basic refers
motan-basicRefer:
  api-core-basicRefer:
    registry: vintage-online
    serialization: simple
    protocol: motan2
    version: 0.1
    requestTimeout: 1000
    haStrategy: backupRequest
    loadbalance: random
    filter: "accessLog,metrics,clusterMetrics"
    maxClientConnection: 10
    minClientConnection: 1
    retries: 1
    application: search # APP_NAME

#conf of refers
motan-refer:
  api-core-yf:
    group: yf-api-core
    path: com.weibo.api.FeedService
    basicRefer: api-core-basicRefer
```

*motan2://127.0.0.1:agent\_port/service=service?group=group*

# 正向代理流程

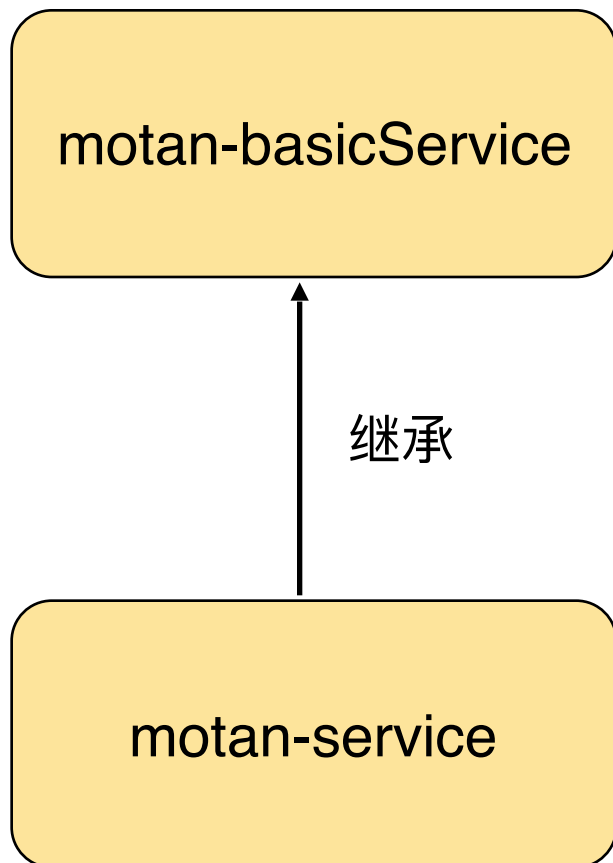


# 正向代理特点

---

- 业务平滑迁移
- agent故障转移，开关控制
- 多发，超时精确控制
- 适合非云，混合云快速迁移

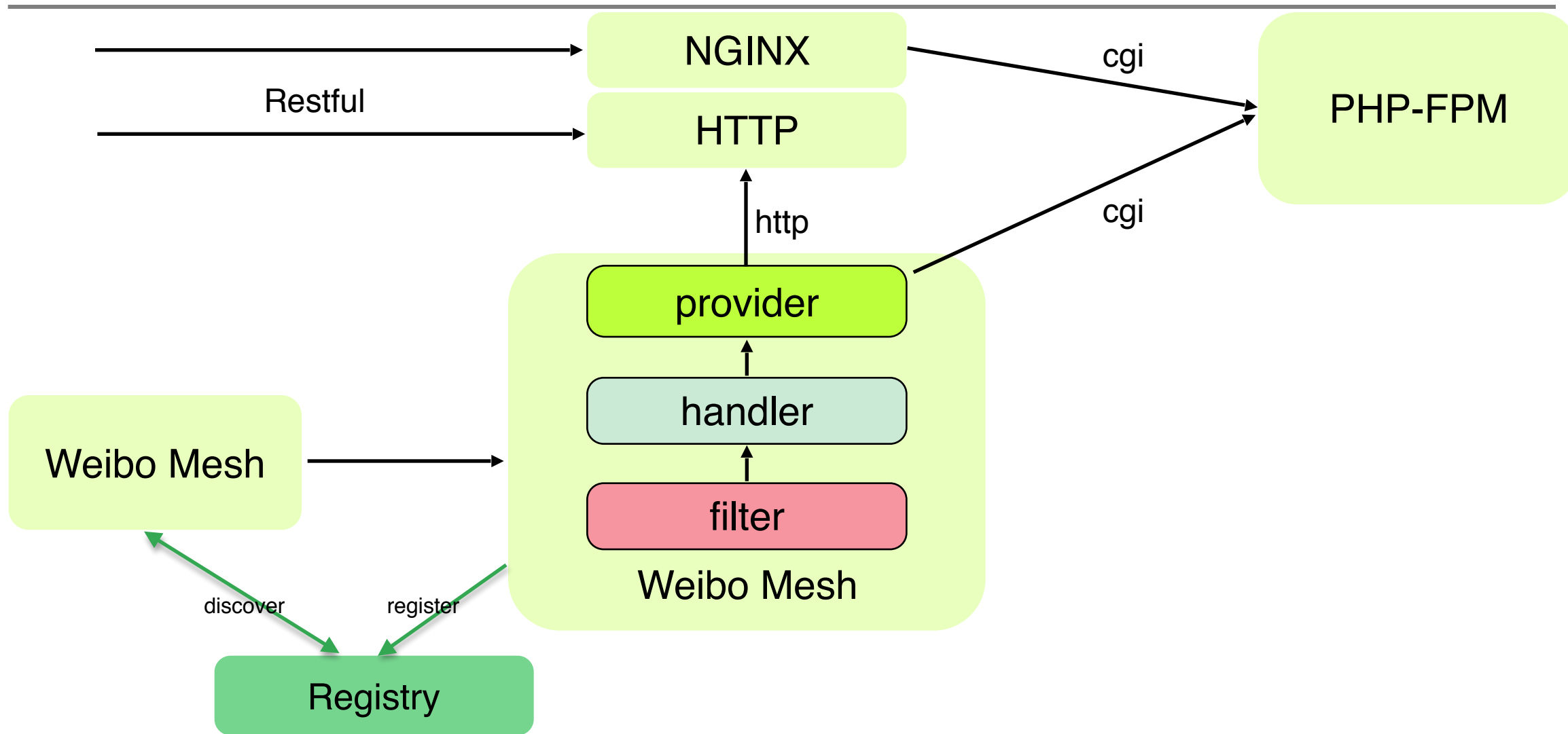
# 反向代理配置



```
#conf of basic services
motan-basicService:
  mesh-server-basicService: # basic refer id
    group: motan-server-mesh-search-test # group name
    protocol: motan2 # rpc protocol
    registry: "vintage-test" # registry id
    filter: "accessLog,metrics" # filter registered in extFactory
    serialization: simple
    nodeType: server
    application: search # APP_NAME

#conf of services
motan-service:
  cgi-mesh-example-helloworld:
    path: com.weibo.search.HelloService
    export: "motan2:9991"
    provider: cgi
    CGI_HOST: 10.210.239.117
    CGI_PORT: 9000
    CGI_REQUEST_METHOD: GET
    CGI_SCRIPT_FILENAME: /data1/nginx/htdocs/hello.php
    CGI_DOCUMENT_ROOT: /data1/nginx/htdocs
    group: motan-server-mesh-search-test
    basicService: mesh-server-basicService
```

# 反向代理流程





# 反向代理特点

---

- 提供HTTP/cgi provider，可扩展
- HTTP框架自动转RPC，业务无需开发新RPC框架
- 不影响原有服务的提供
- 改造成本极低

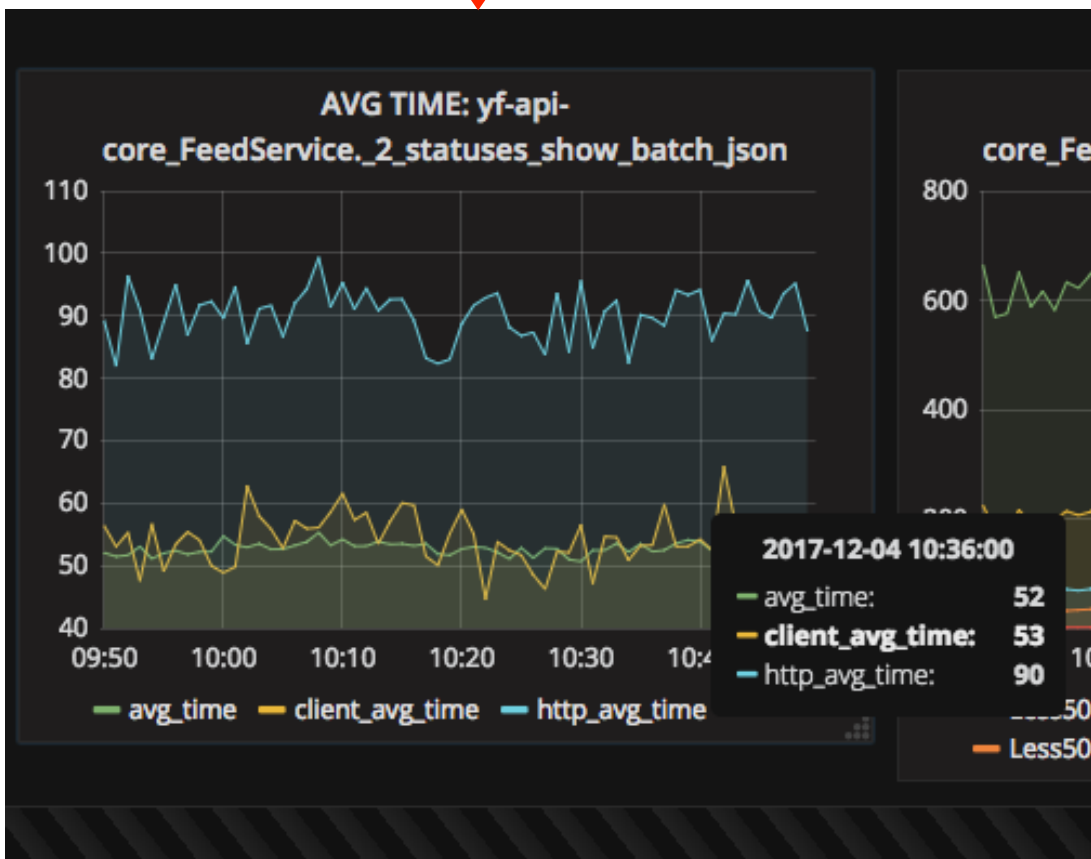
# Weibo Mesh的收益



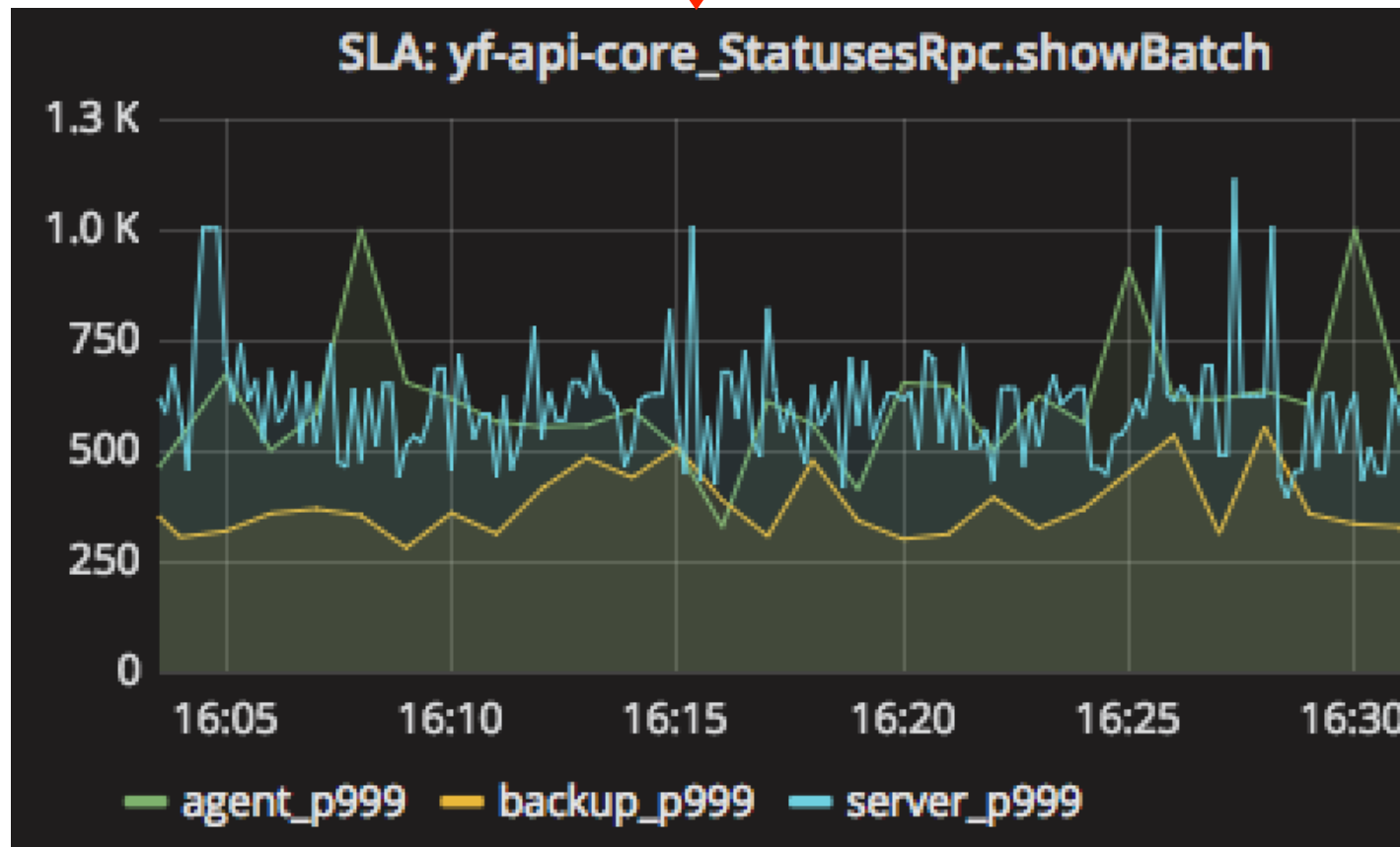
# Weibo Mesh在业务应用中的效果

平均耗时  20%~40%

SLA999  15%~50%

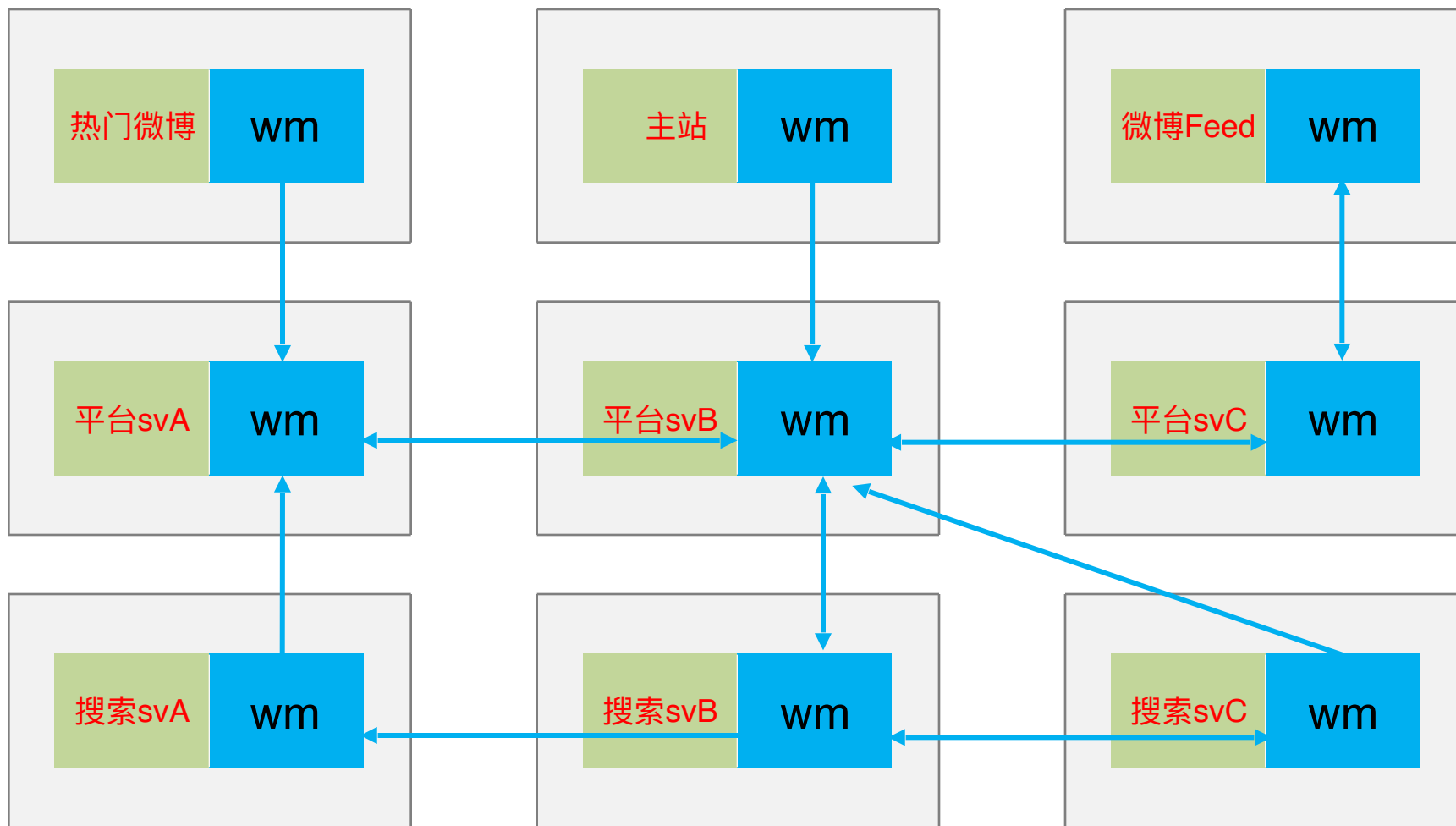


Mesh VS HTTP

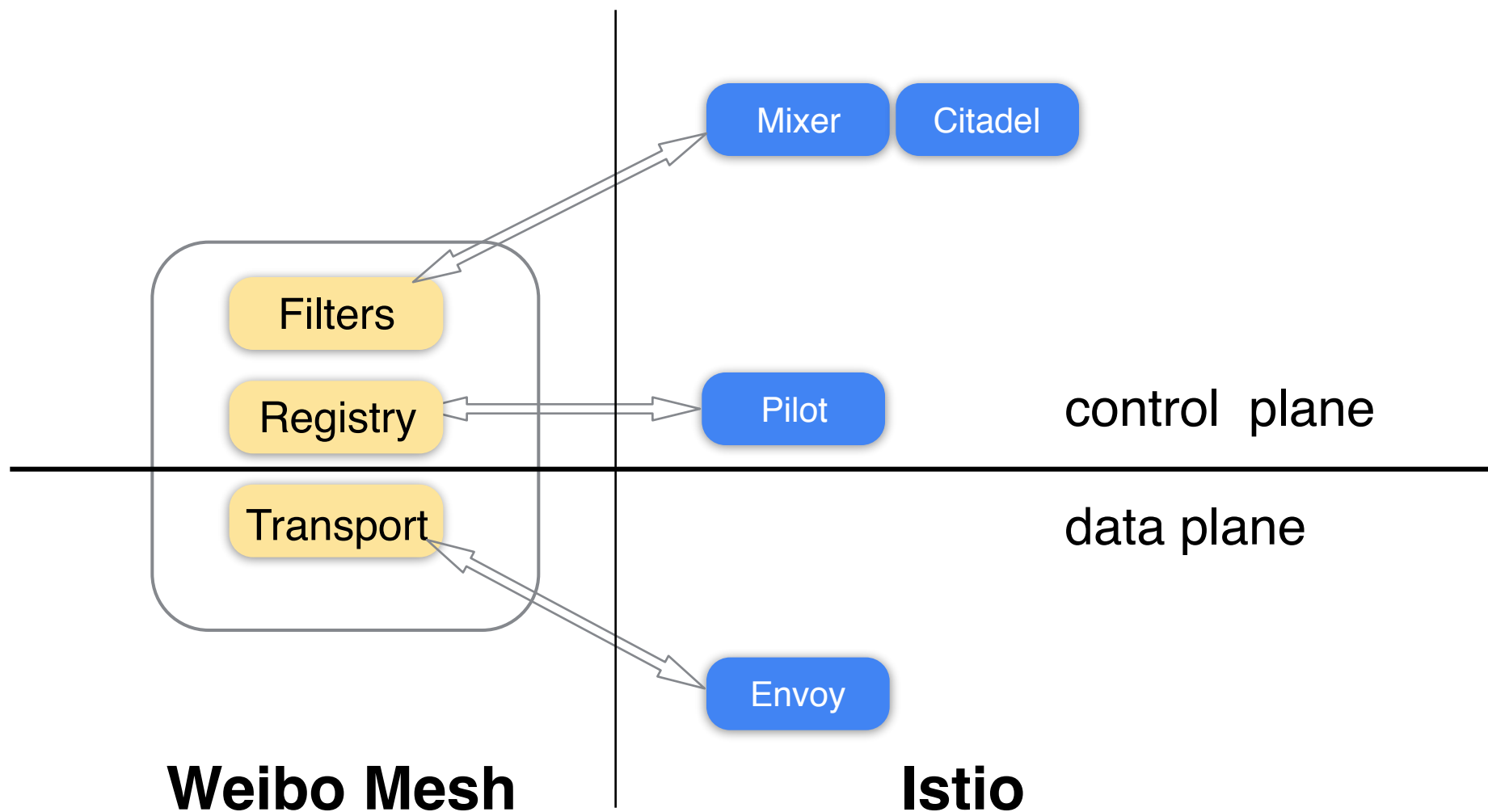


Backup Request 效果图

# Weibo Mesh集群

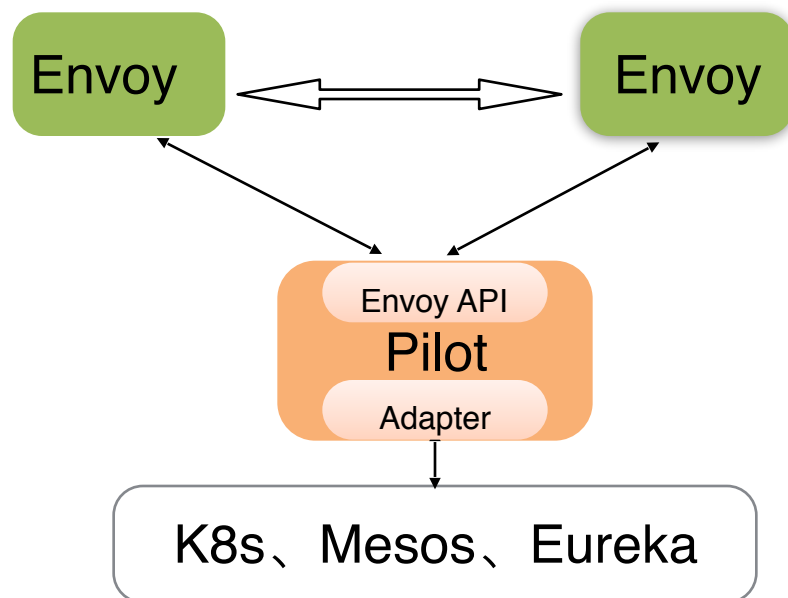


# 和Istio在数据面和控制面的区别

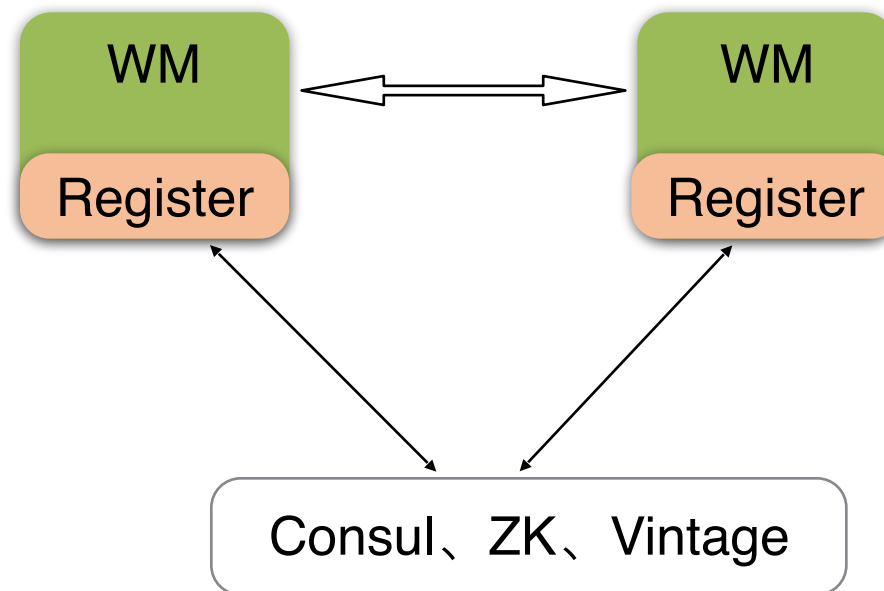


# 和Istio在Discovery上的区别

云 VS 非云



**Istio: Pilot适配云平台**



**WM: 注册中心**

# 和Istio在业务感知上的区别

---

## 服务透明 VS 模块耦合

- 云原生
- IPtables流量拦截
- 服务无感知

**Istio：对服务透明**

- 耦合度可选
- 定制化开发

**WM：模块化耦合**

# WM未来发展方向

---

## 泛服务化

服务协议

版本号

服务分组

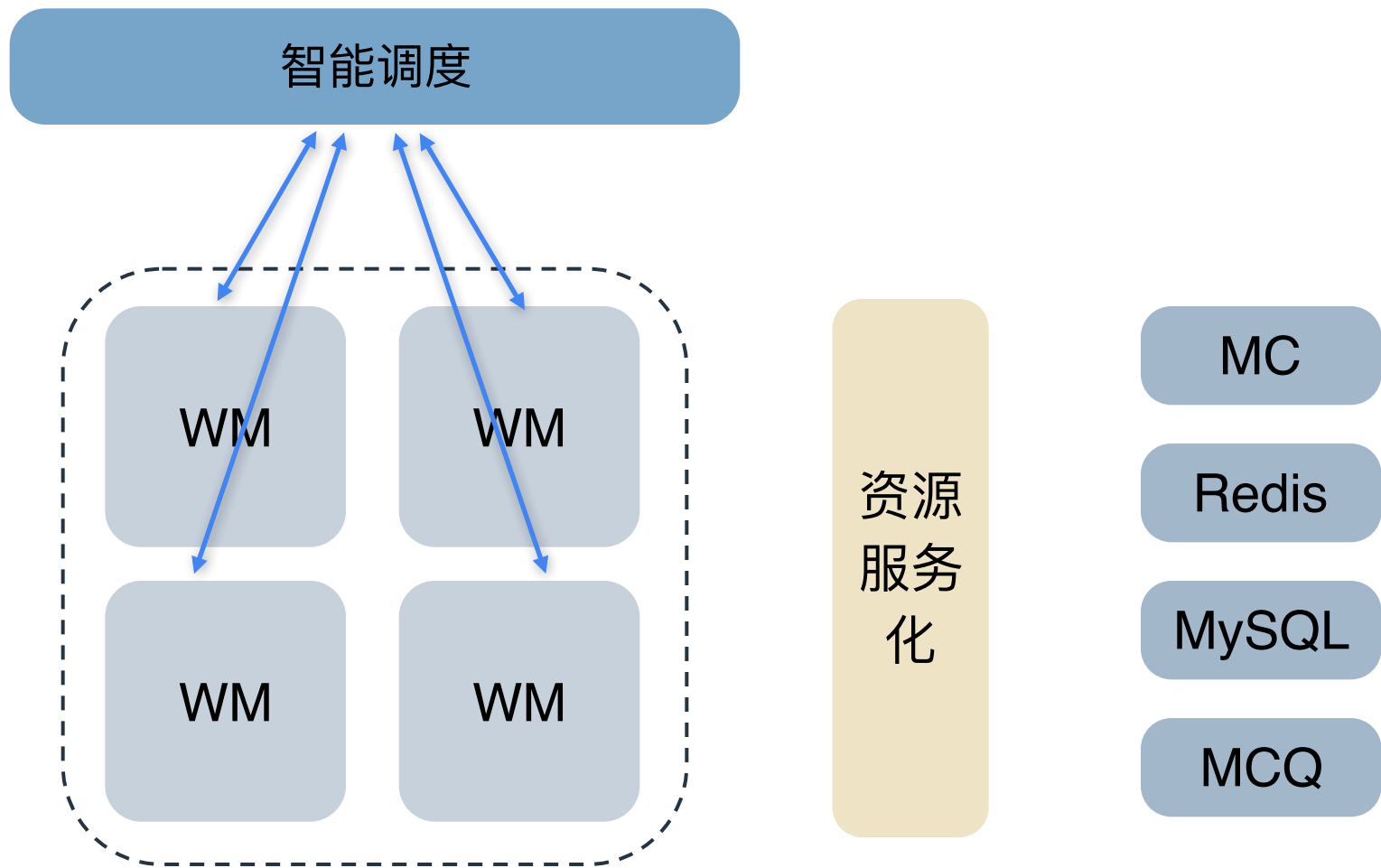
mcq://host:port/0.1/helloworld.Greeter?group=group-hello

目标节点

Service



# WM未来发展方向



# Weibo Mesh开源项目地址

---

WeiboMesh: <https://github.com/weibocom/motan-go>

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

OR: <https://github.com/weibocom/motan-openresty>

PHP: <https://github.com/weibocom/motan-php>

Examples: <https://github.com/motan-ecosystem/motan-examples>



# Thanks

## Q&A

微博搜索

丁振凯