

微博跨语言服务化与 WeiboMesh

— 微博研发中心 / 周晶

内容提要

- ▷ 为什么要做跨语言服务化
- ▷ PHP 跨语言服务化所面临的挑战
- ▷ WeiboMesh 与 泛服务化 附能 PHP 微服务

为什么要做跨语言服务化

为什么要做跨语言服务化

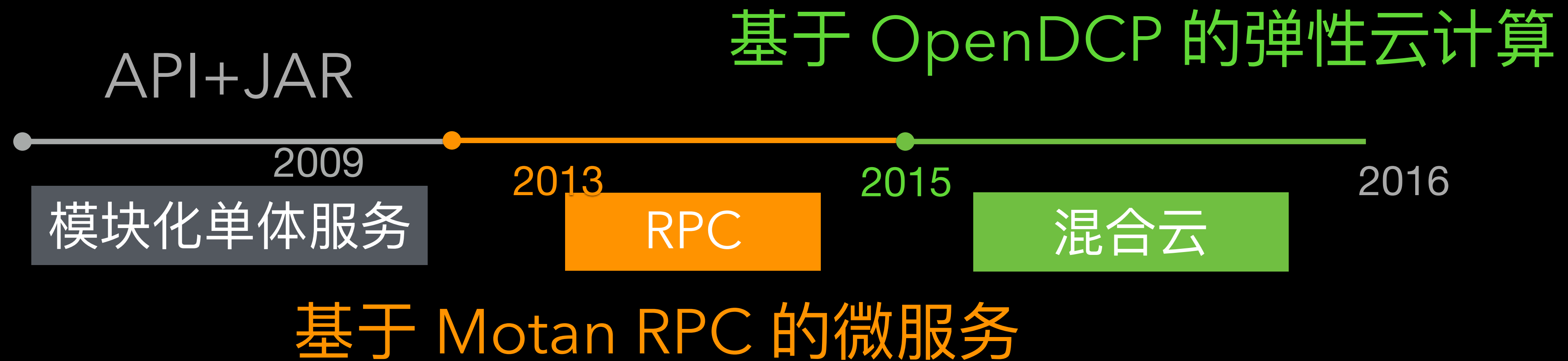
▷ 痛点

▷ 趋势

痛点

需求 · 性能 · 成本

平台体系



平台体系

服务注册、发现

负载均衡、降级、限流

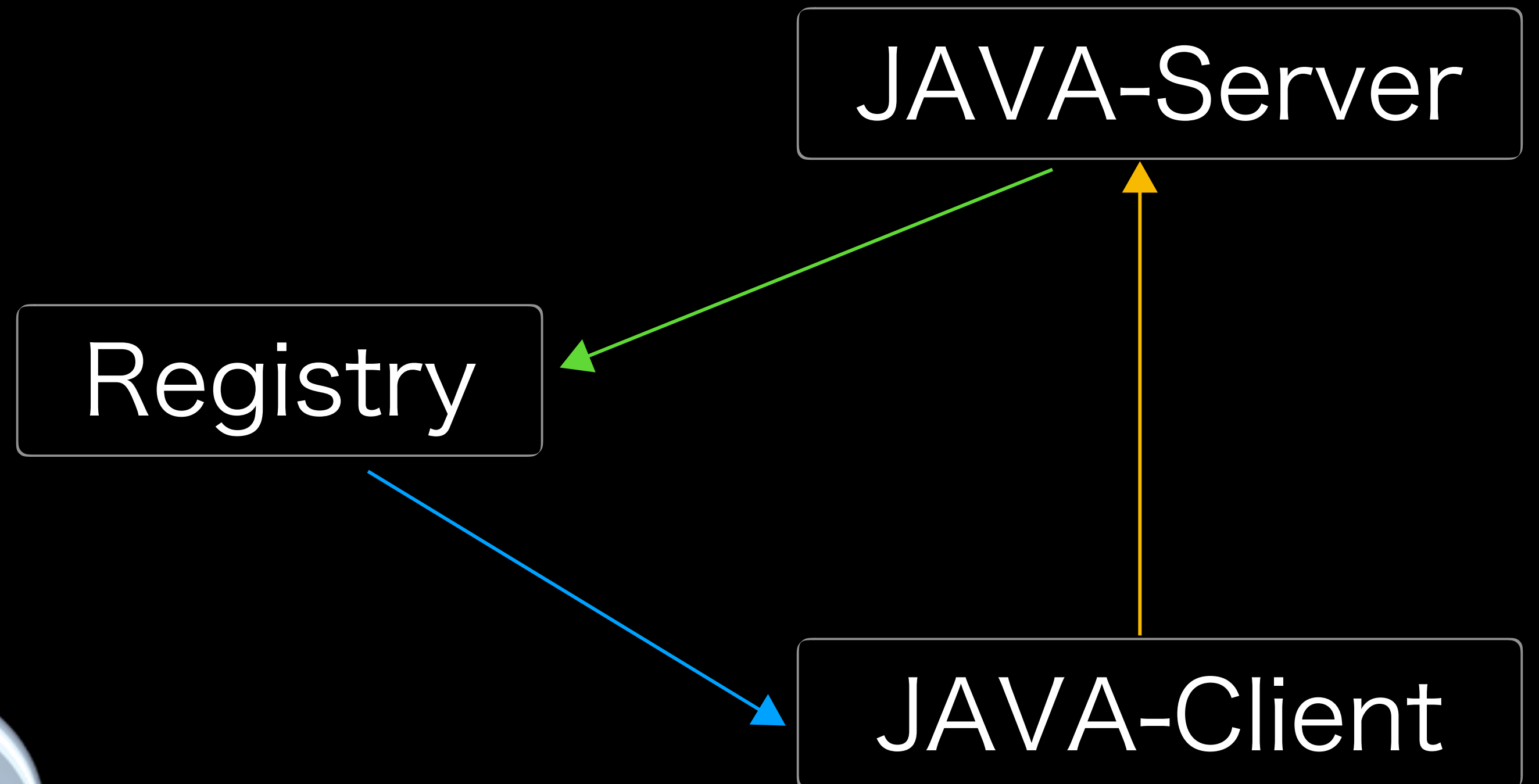
秒级实时监控

自动化容量评估

智能弹性调度

自动扩缩容

请求动态路由





微博广告

红包

微博话题

MAPI

微博搜索

热门微博

微博主站

...

平台体系

Restful API

↓
Java RPC

趋势

微服务 · 容器化 · 云化 · 标准化 · 工程化

更快的交付

更低的成本

服务与实现方式不应该有强耦合关系

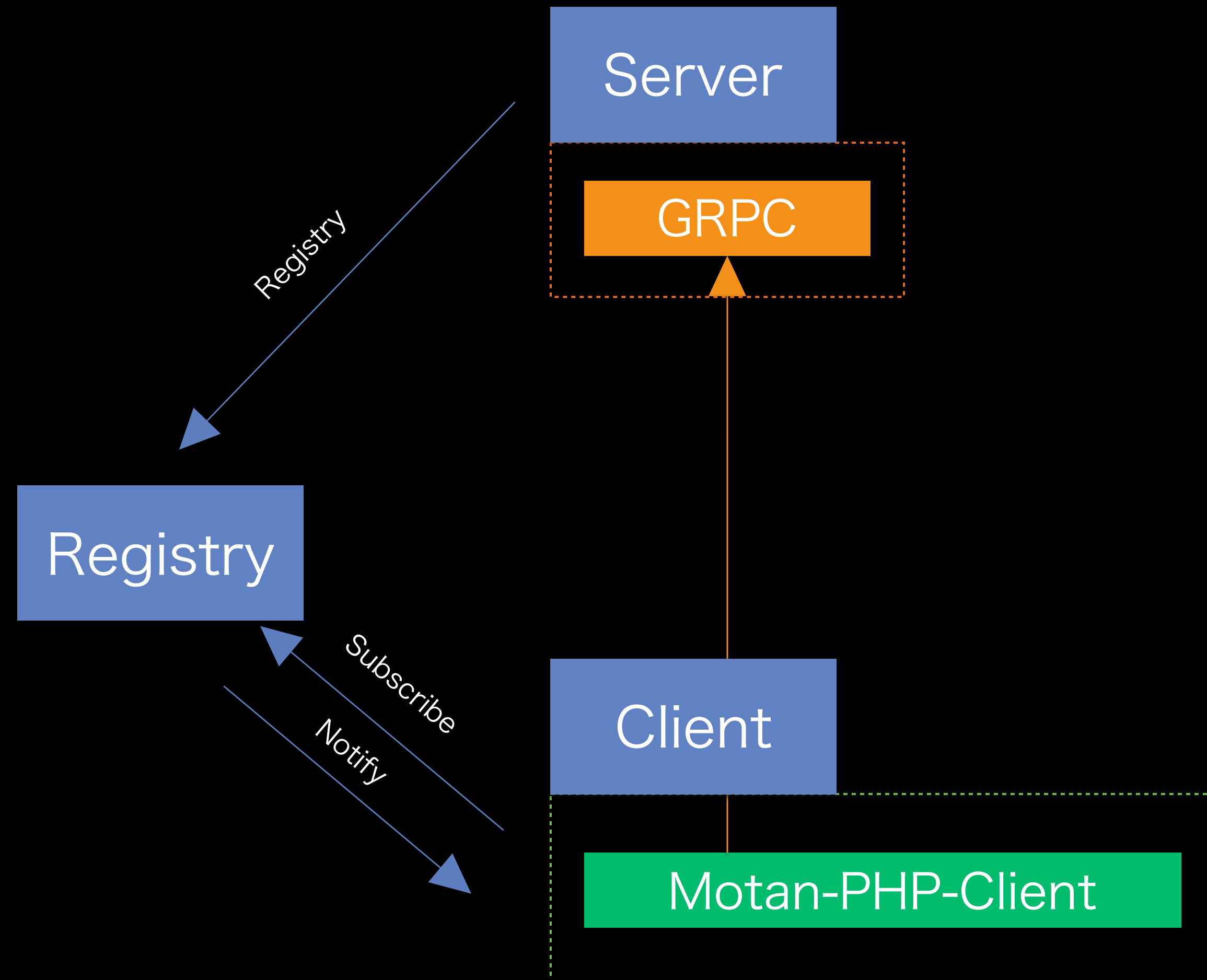
PHP 跨语言服务化所面临的挑战

跨语言服务化核心问题

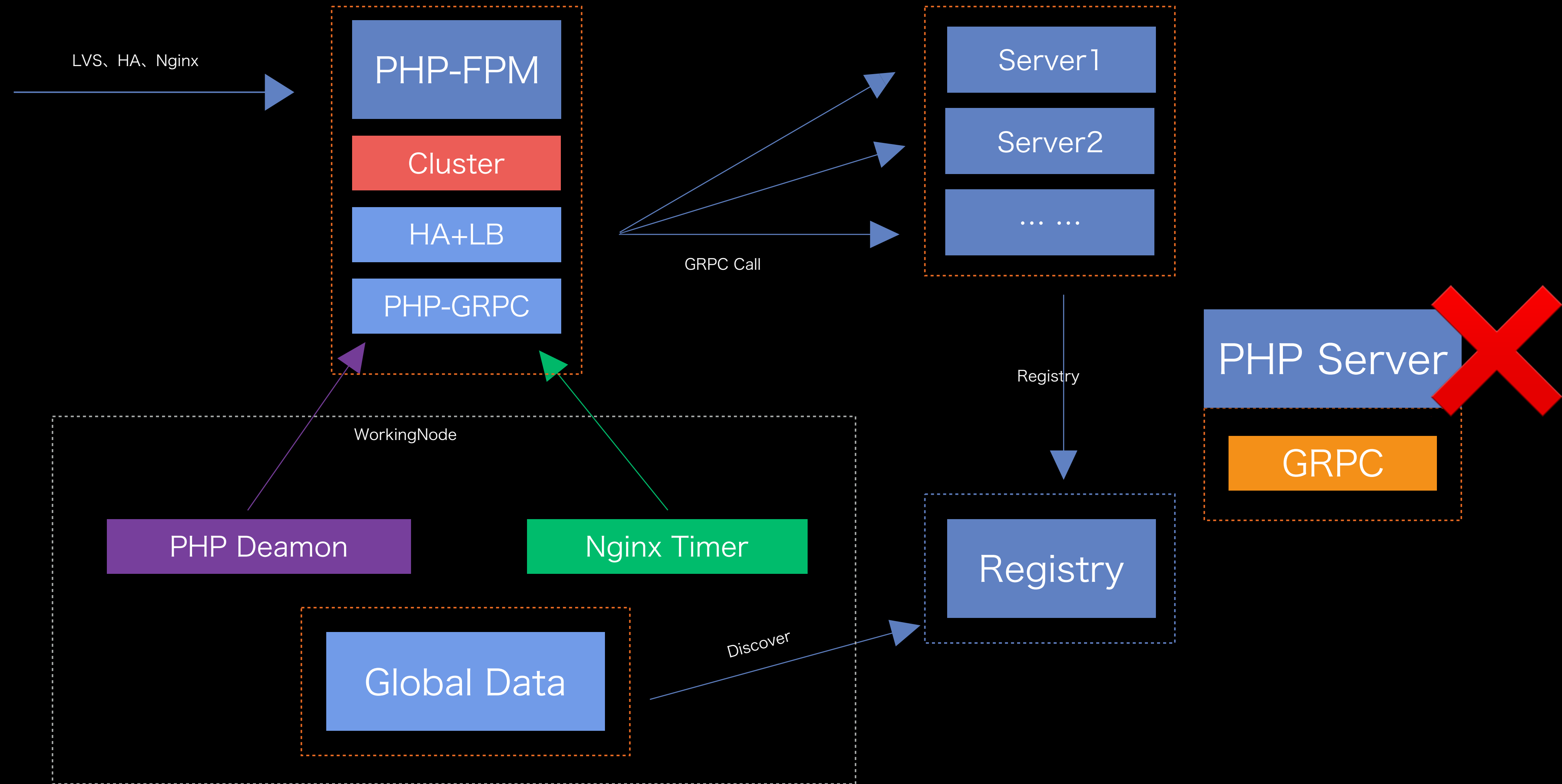
交互（语言中立、多语言）：HTTP / gRPC (PB) / ...

治理（灵活可扩展）：服务注册与发现 / 负载均衡 / ...

挑战一：迁移成本与性能



挑战二：服务治理与 RPC Server



实现成本
治理能力

经验总结

PHP 服务化的现状: LN(OR)MP 架构 / 交付效率 / 迁移成本 / ...

PHP 语言本身的局限性: 性能 / 常驻能力 / ...

灵活可扩展的服务治理: 非中心化 / 非重 Client / ...

我们需要适合自己的跨语言解决方案

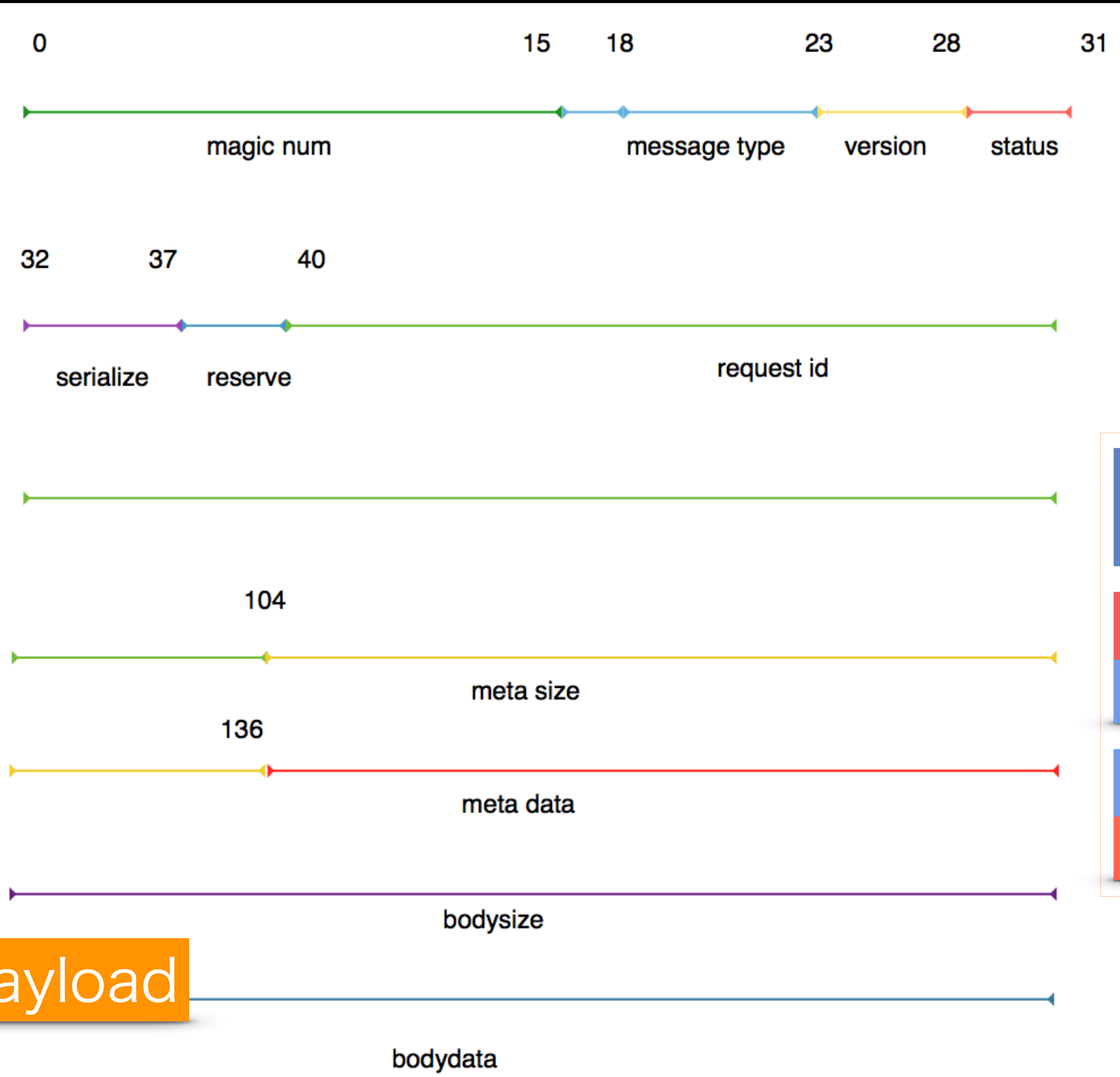
交互 · 治理

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
治理能力	中	高	高
扩展能力	低	中	高

交互（语言中立、多语言）

协议



MotanHeader	
Meta Data	Size
Size	Data
Req Body	Size
Size	Data

Header Payload

序列化

null

00

string "hello"

01 00 00 00 05 68 65 6c 6c 6f

基础类型

复合类型

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

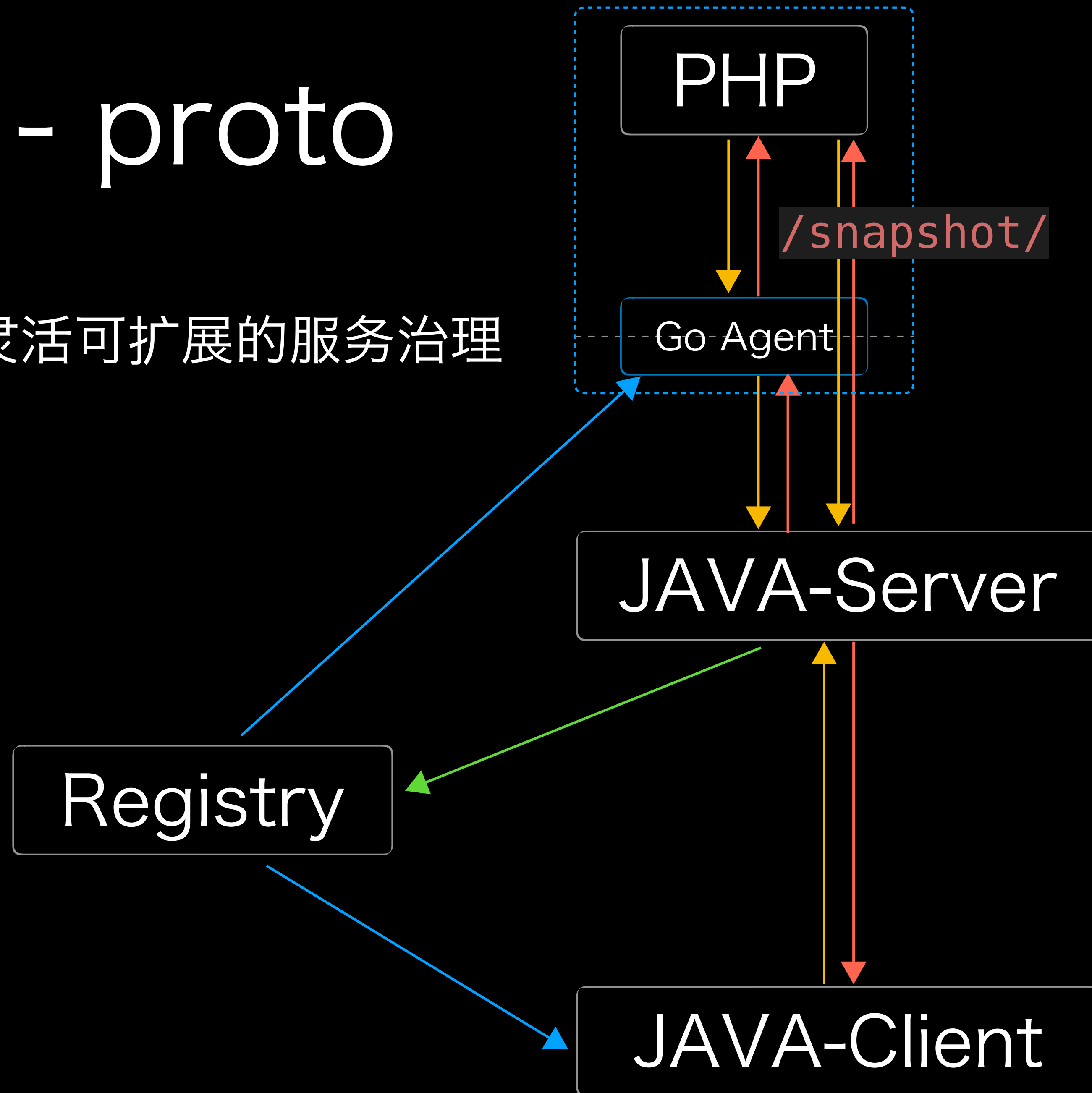
```
// serialize type
const (
    sNull = iota
    sString
    sStringMap
    sByteArray
    sStringArray
    sBool
    sByte
    sInt16
    sInt32
    sInt64
    sFloat32
    sFloat64

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

治理（灵活可扩展）

WM - proto

灵活可扩展的服务治理



Motan-PHP 跨语言调用

```
if ($connection->buildConnection($agent_addr)){  
    $this->_url_obj  
    ->setEndpoint(Constants::ENDPOINT_AGENT);  
    $this->_endpoint = new Agent($this->_url_obj);  
} else {  
    $this->_endpoint = new Cluster($this->_url_obj);  
}
```

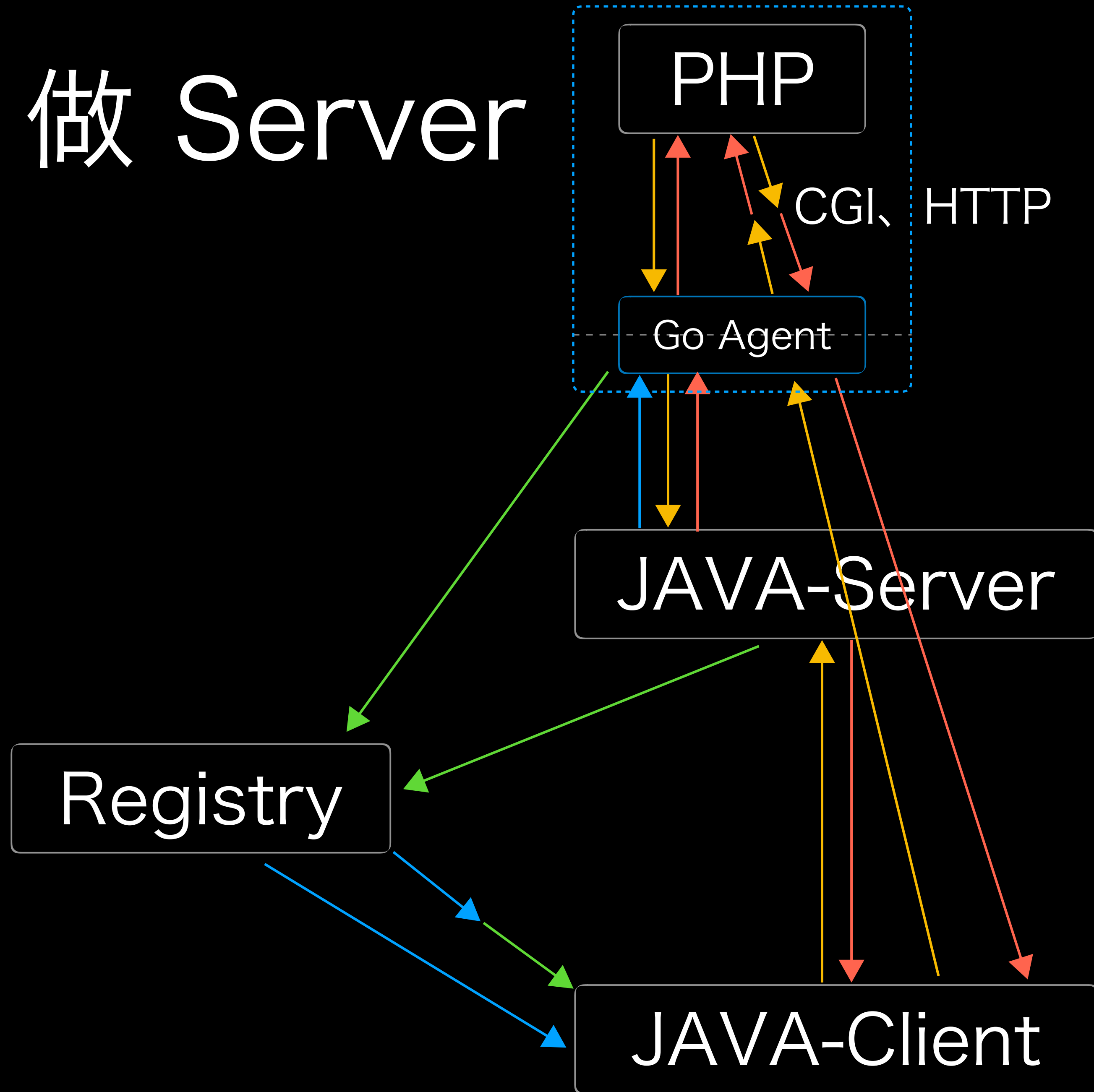
Motan-PHP Cluster 直连策略

```
public function call()
{
    return $this->_ha_strategy->call($this->_load_balance);
}
```

```
AGENT_RUN_PATH . "/snapshot/" . $this->_group . '_' . $this->_service_str;
```

Endpointer 扩展保底

PHP 做 Server



Motan-Go Provider 与 PHP

```
extFactory.RegistExtProvider(CGI, func(url *motan.URL)
motan.Provider {
    return &CgiProvider{url: url}
})
```

```
extFactory.RegistExtProvider(HTTP, func(url *motan.URL)
motan.Provider {
    return &HTTPProvider{url: url}
})
```

MOTAN_*

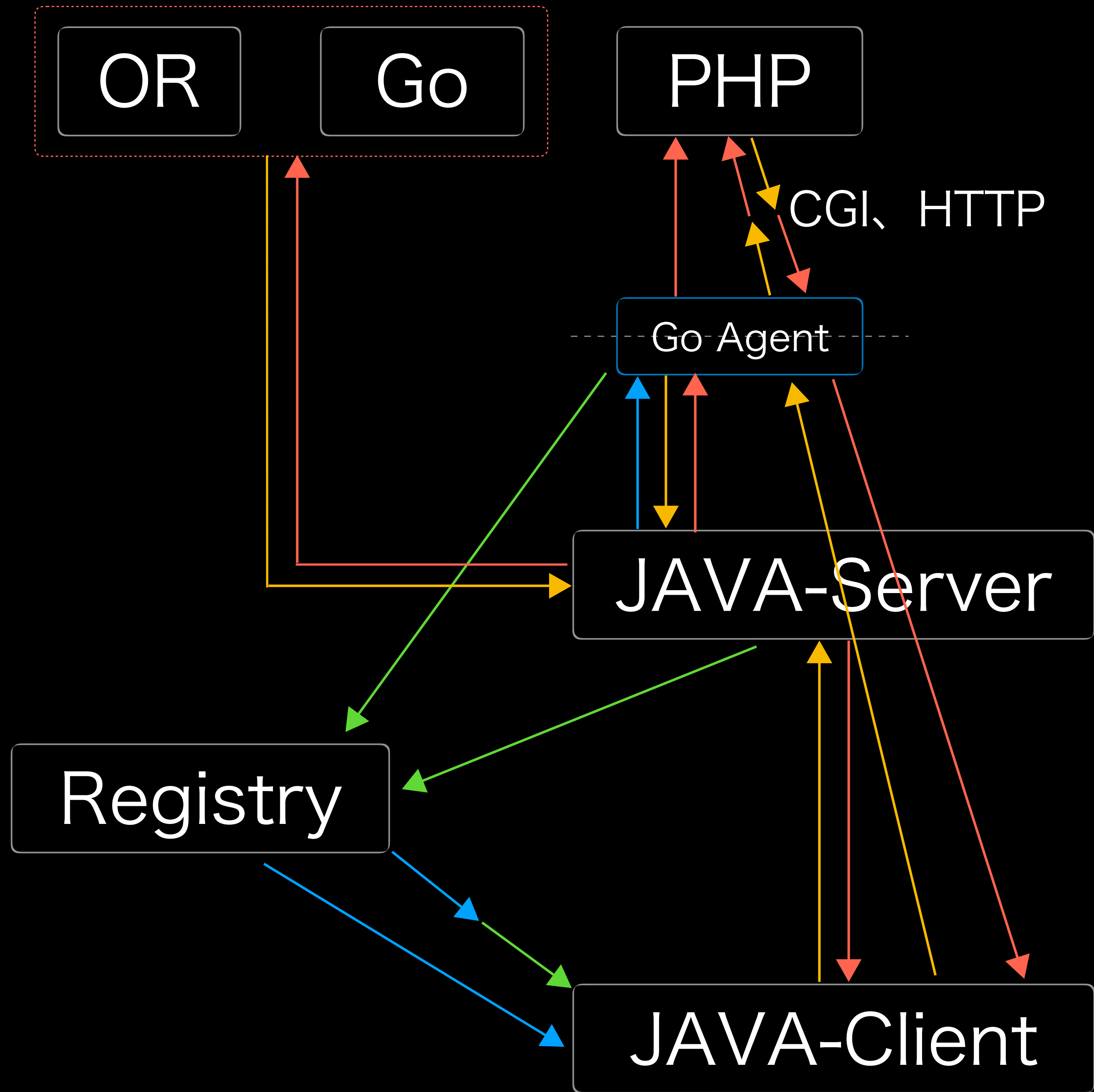
[MOTAN_M_p] => com.weibo.motan.HelloWorldService

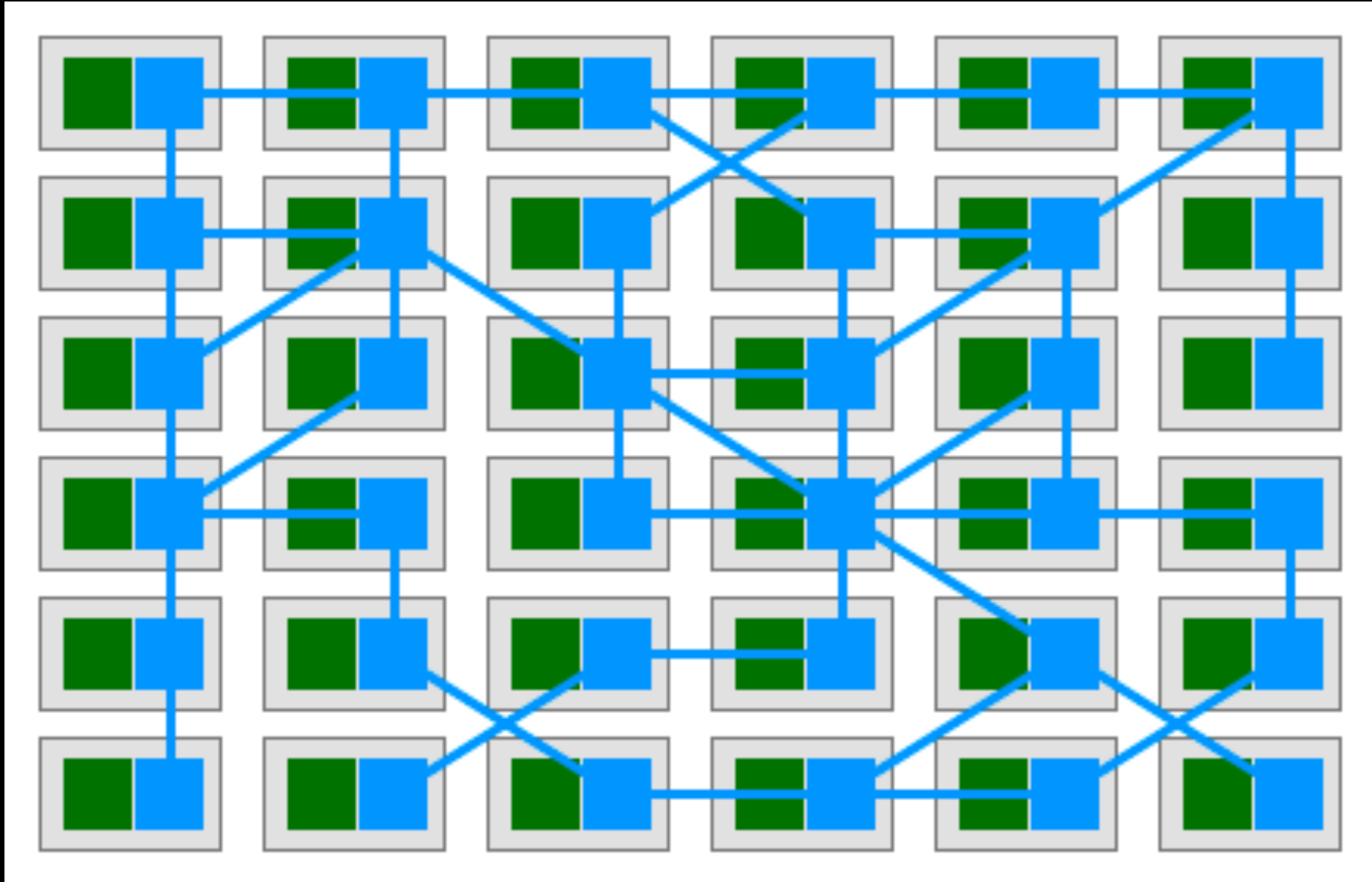
[MOTAN_SERIALIZATION] => simple

[MOTAN_M_g] => motan-server-mesh-example

[MOTAN_M_m] => Hello

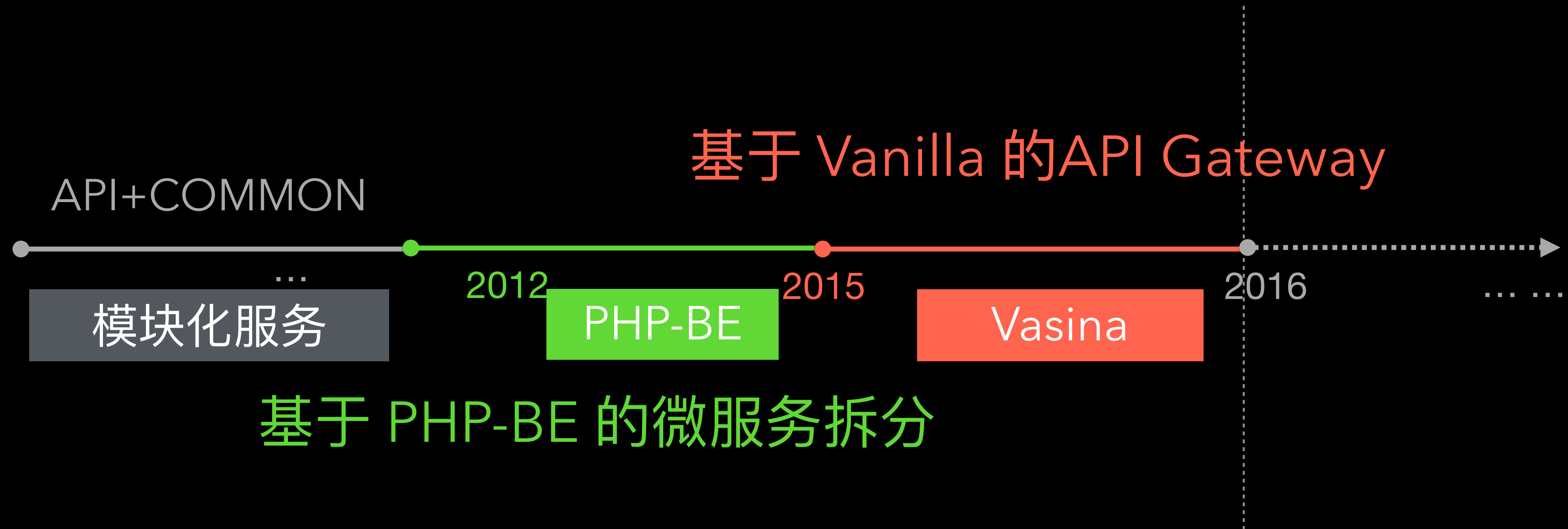
[MOTAN_M_pp] => motan2



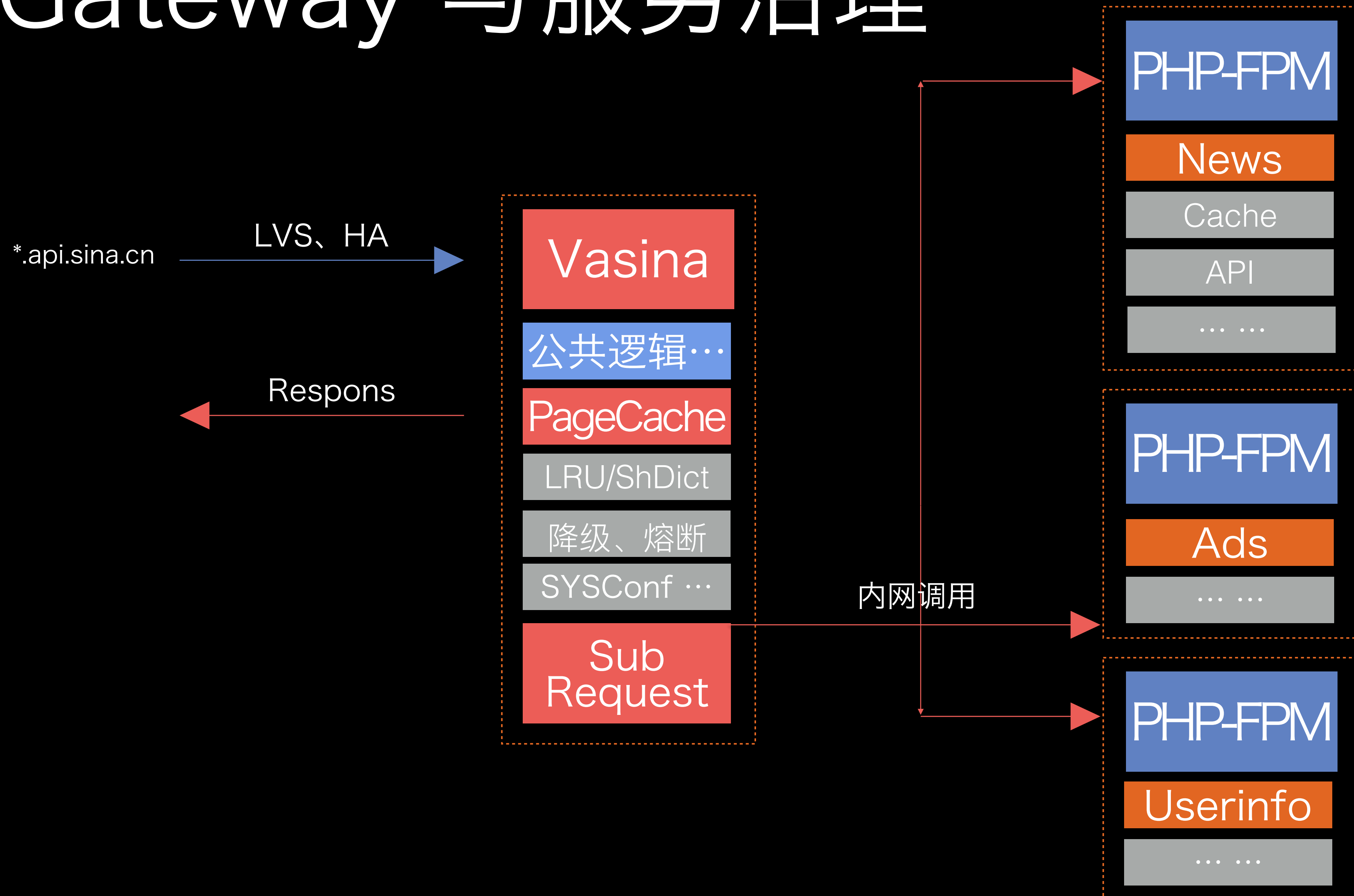


WeiboMesh 与 泛服务化 附能 PHP 微服务

传统 PHP 服务化演进



APIGateway 与服务治理



Service Mesh 是什么

A service mesh is a **dedicated infrastructure layer** for **handling service-to-service communication**. It's responsible for the reliable delivery of requests through the **complex topology of services** that comprise a modern, cloud native application. **In practice**, the service mesh is typically implemented as an array of **lightweight network proxies** that are deployed alongside application code, **without the application needing to be aware**.

Service Mesh 是什么

dedicated infrastructure layer **handling service-to-service communication**

complex topology of services

In practice

lightweight network proxies

without the application needing to be aware

APIGateway

Service
A

Service
B

Service
C

APIGateway

micro Service

A

Service

B

Service

C

APIGateway

micro Service

A

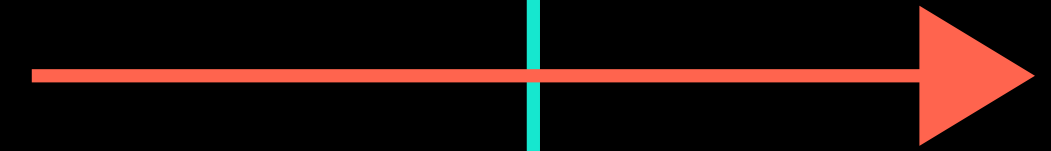
Service

B

Service

C

Service Mesh



APIGateway

micro Service

A

Service

B

Service

C

Service

Service Mesh

Service Mesh

解耦·下沉

通信和服务治理 dedicated infrastructure layer

业界 Service Mesh 实现

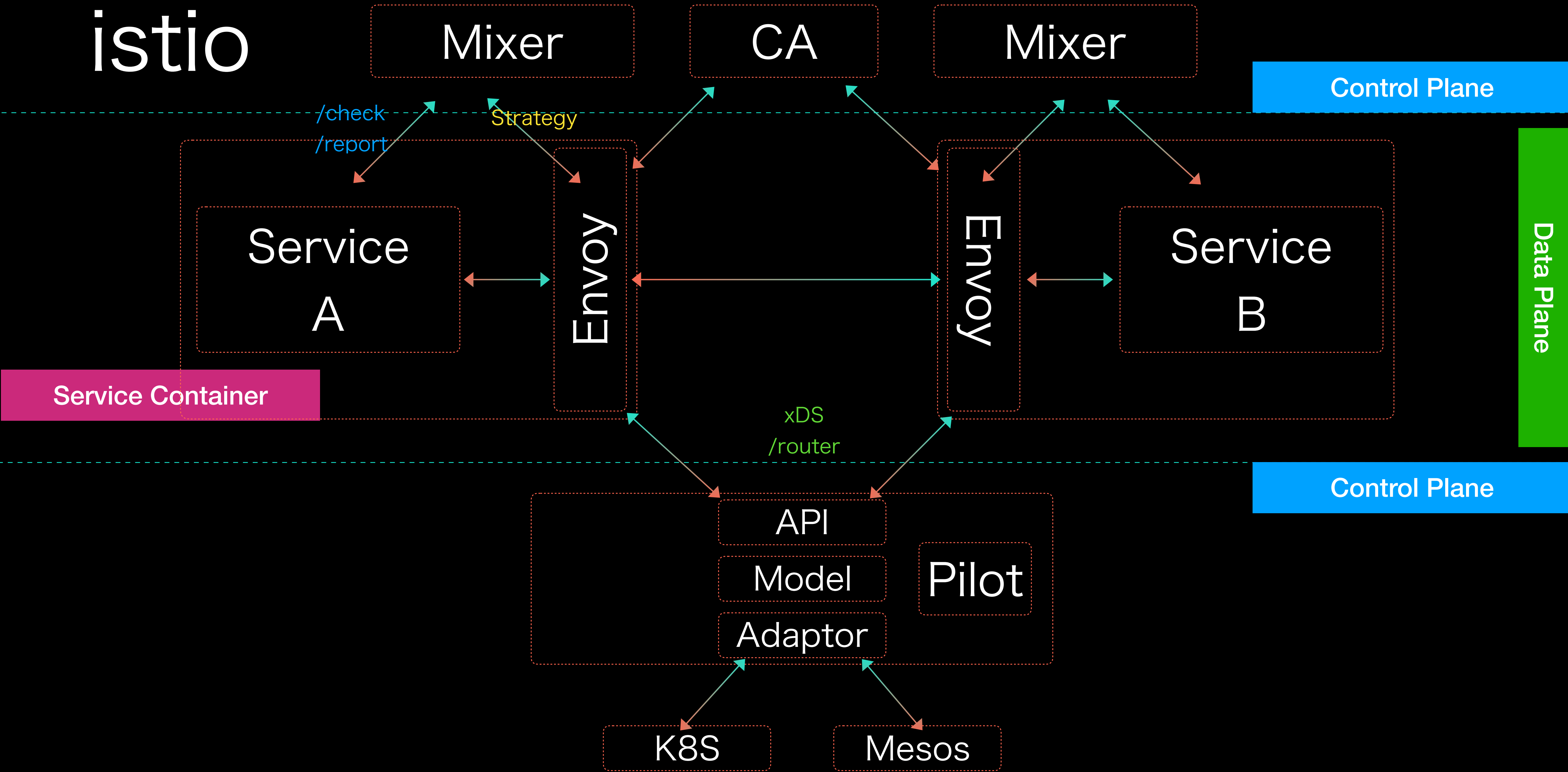
名称	厂商	版本	基金
istio	Google、IBM、Lyft	0.7.1	CNCF
conduit	Buoyant	0.4.1	/
nginmesh	Nginx	0.7.2	/
Linkerd	Buoyant	1.4.0	CNCF
serviceComb	HUAWEI	1.0.0	Apache

Service Mesh 事实规范

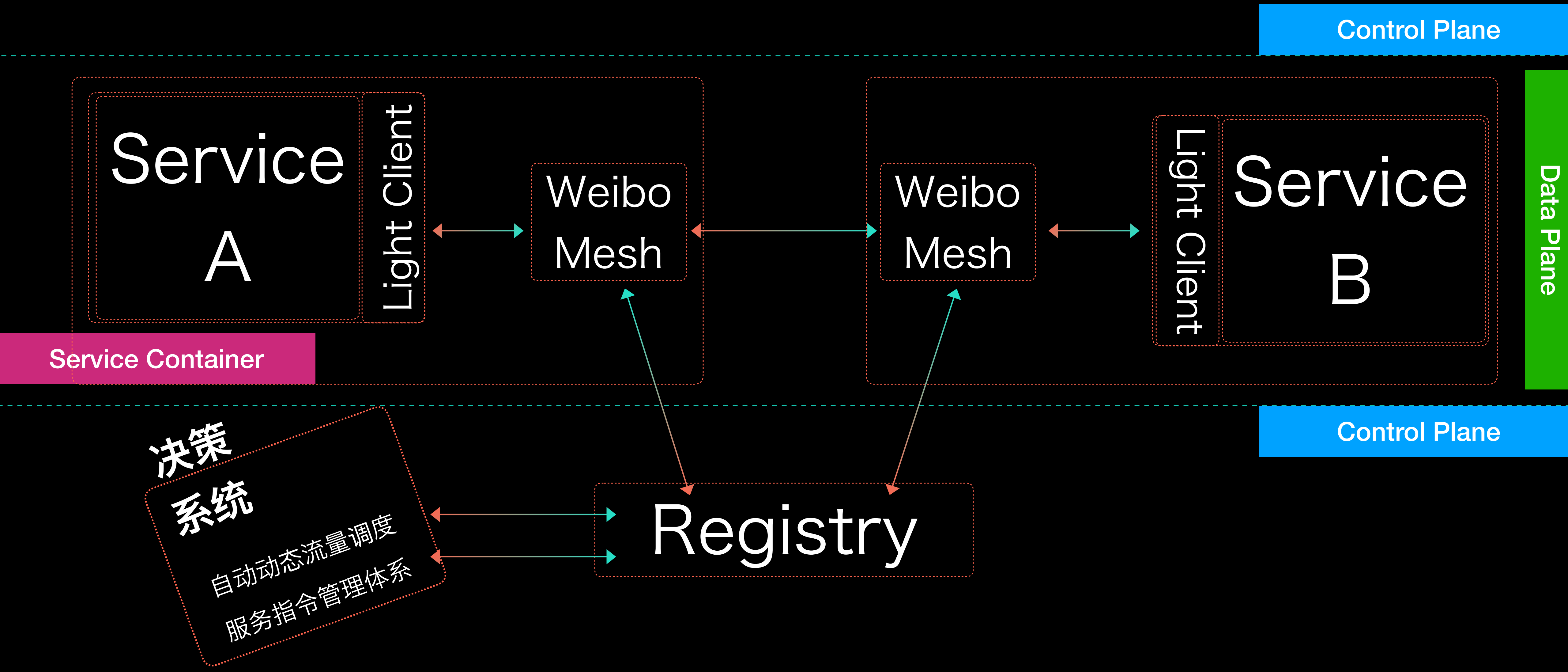
Data Plane: 专注 Service 间请求数据传输, 完成服务之间的交互

Control Plane: 专注服务治理、流量路由、执行策略等

istio



Weibo Mesh



WeiboMesh 泛服务化

Server: Go / OR / JAVA / PHP / MC / Redis / ...

Motan2 / MC / Redis / ...

Weibo Mesh

Motan2

Weibo Mesh

Motan2 / ...

Client: Go / OR / JAVA / PHP / ...

服务的刻画 / Motan Schema

服务协议

版本号

Service

grpc://host:port/0.1/helloworld.Greeter?group=idevz-test

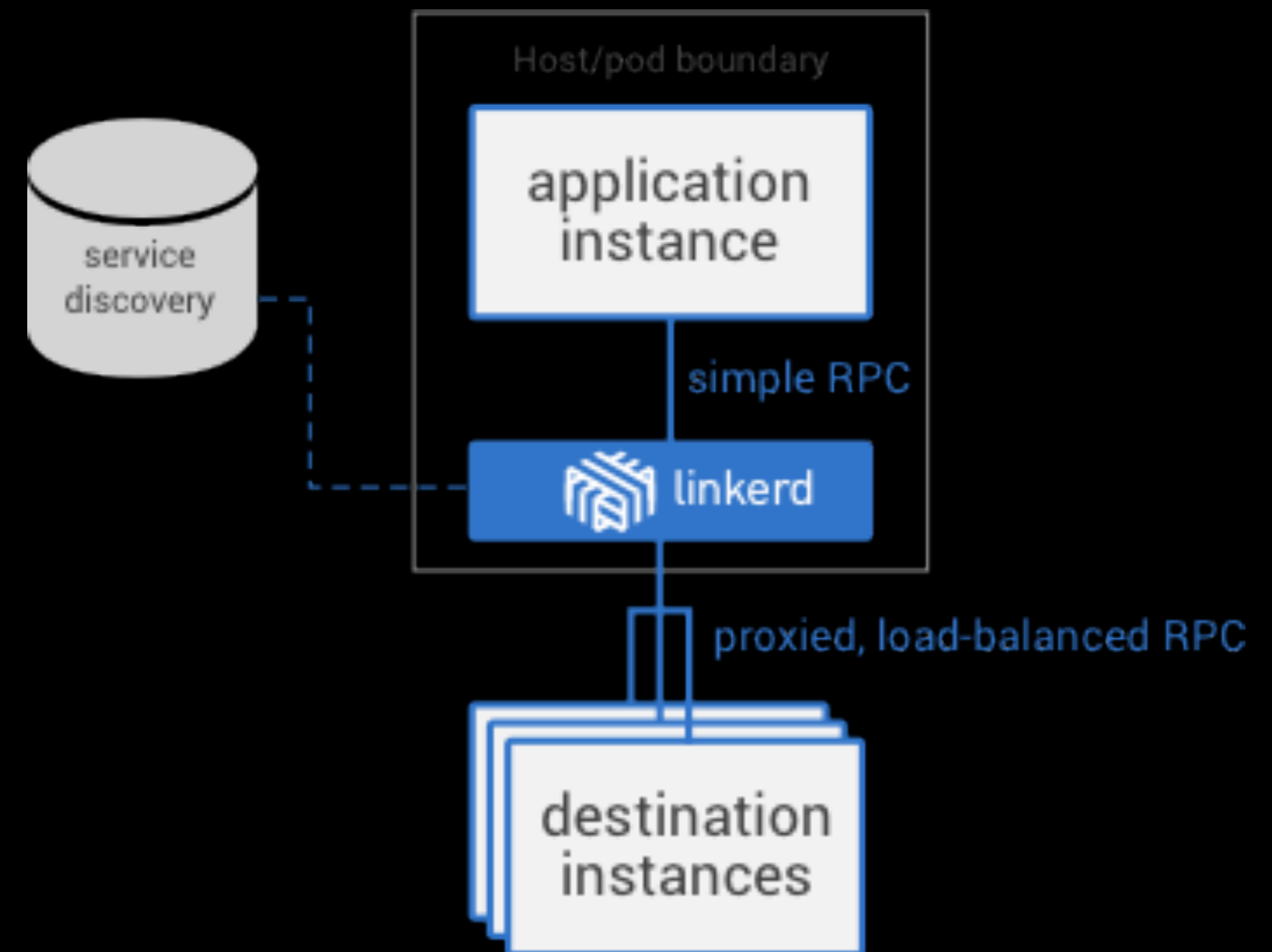
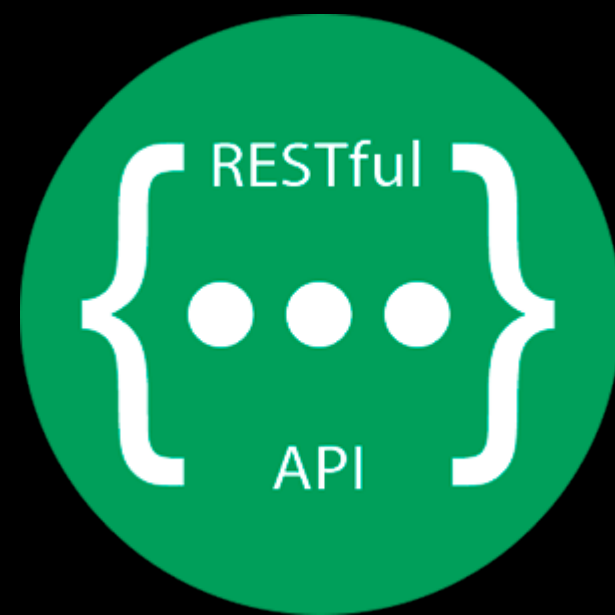
目标节点

服务分组

URL

唯一刻画

我们依赖的只有服务



附能 PHP 微服务

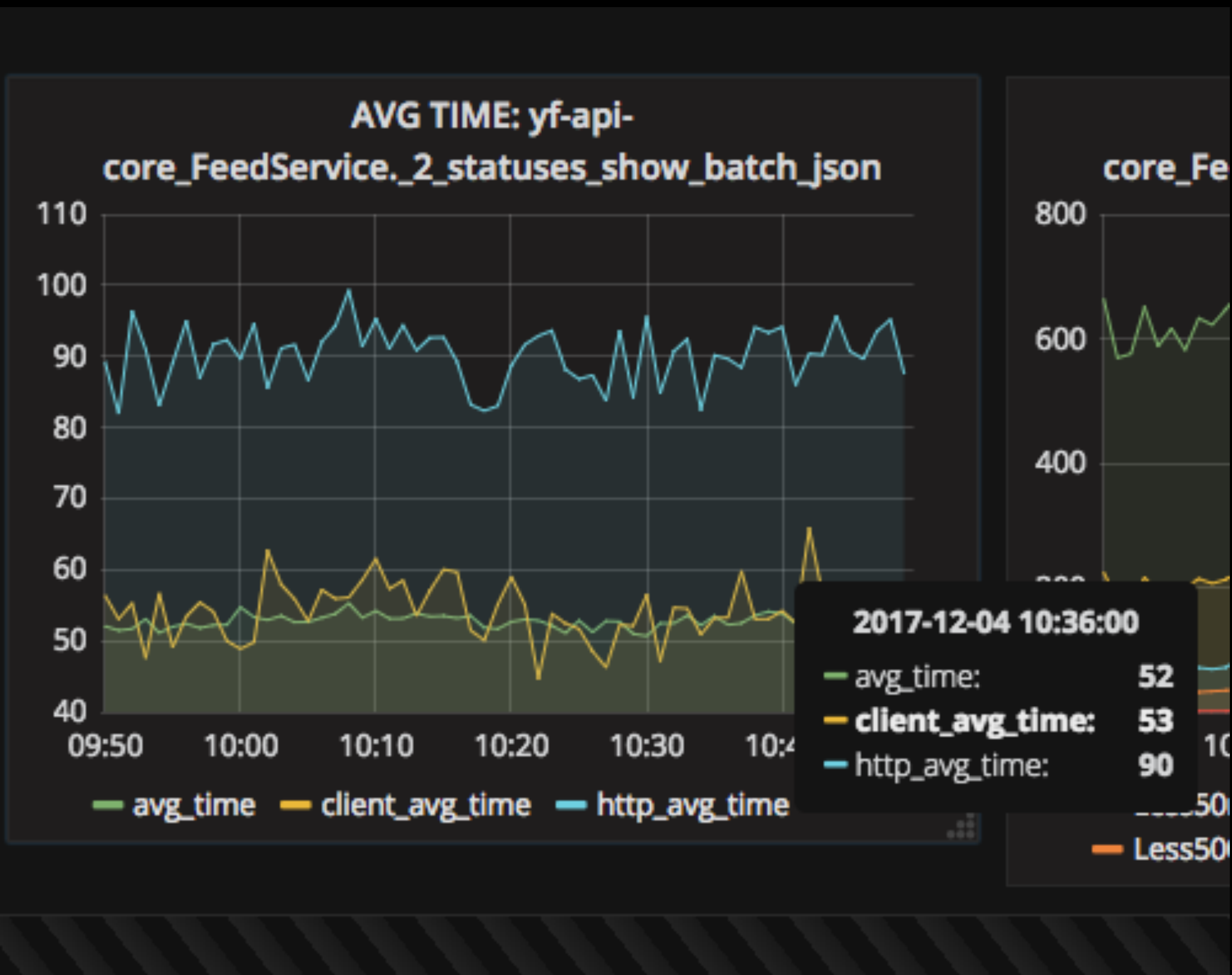
跨语言服务化能力

提高性能（点对点直连、长链接）

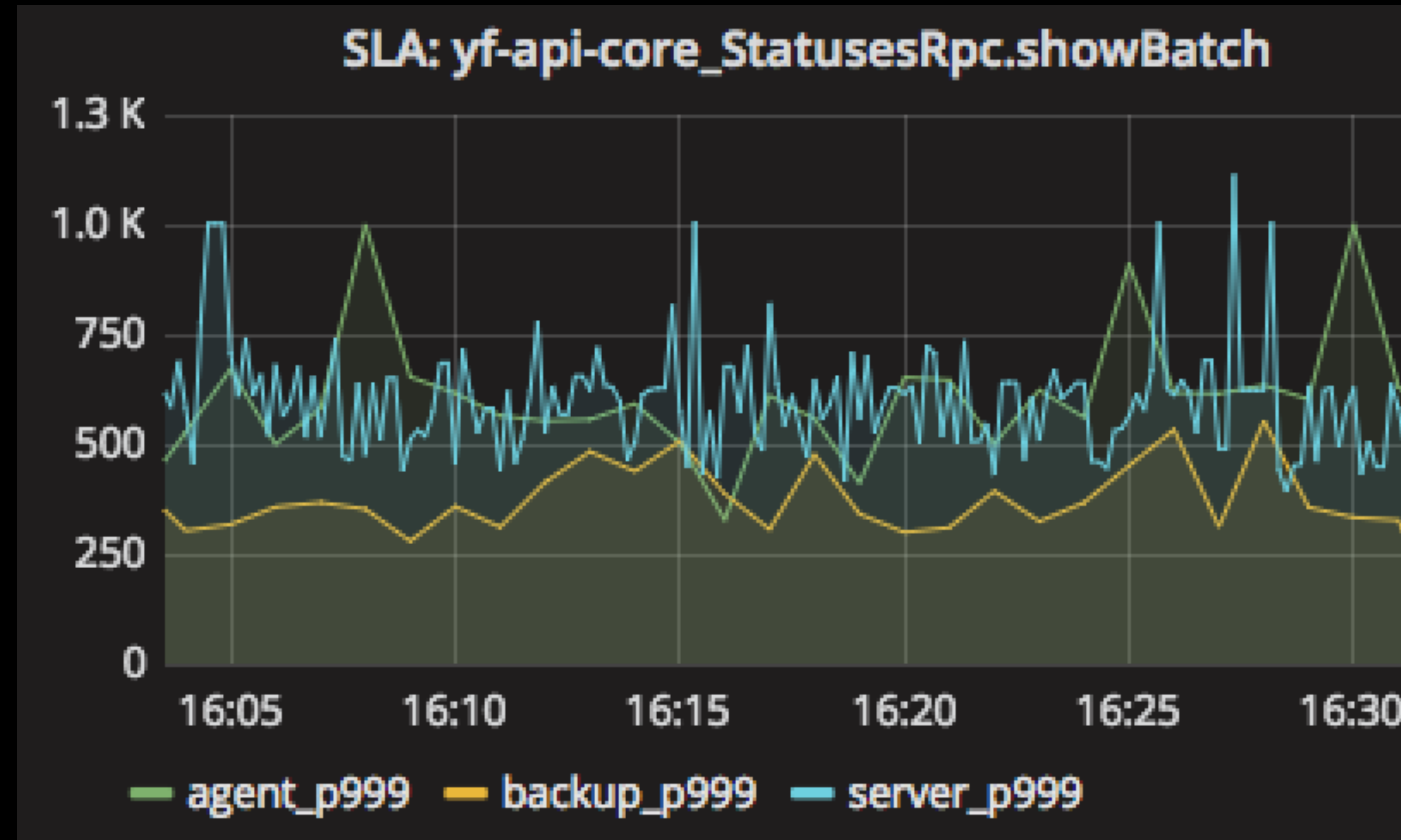
统一服务治理能力：流量调度、双发、多发、

资源服务化（redis、mc、kafka、hbase、cacheservice等）

无感知持续功能更新



Mesh VS HTTP



Backup Request

