

TARS-PHP

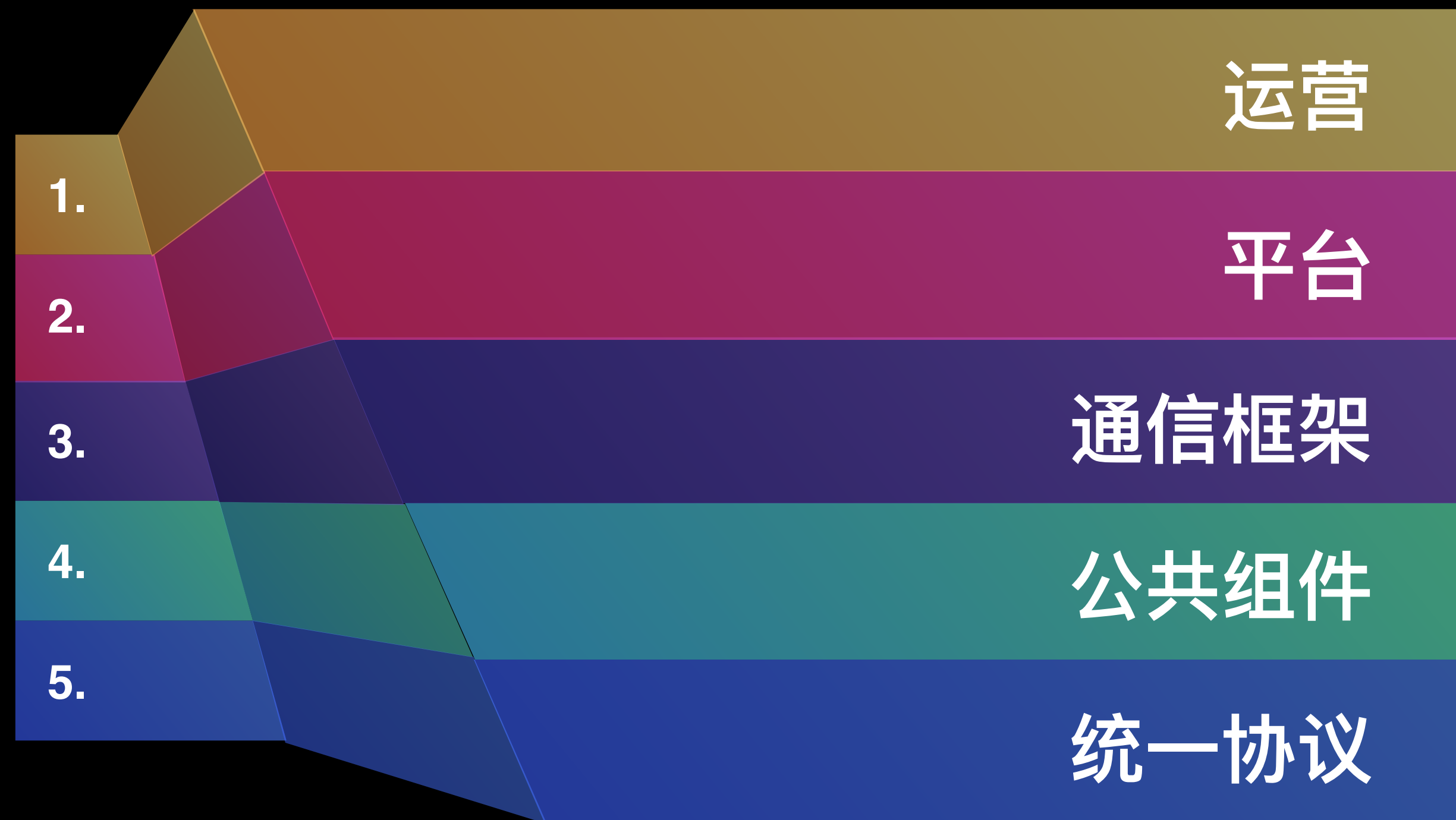
PHP构建高性能RPC框架



梁晨 · Ted

- 2007 ~ 2014 上海交通大学 本硕
- 2014 ~ 2016 腾讯 企业QQ&QQ公众号 Web后台开发
- 2016 至今 阅文集团 起点开发组 Web开发
- 腾讯开源项目TSF开发组成员
- 腾讯开源项目TARS贡献者
- liangchen@yuewen.com

TARS



TARS-运营与平台

TARS-服务部署

TARS



服务管理



运维管理



TARS ▼

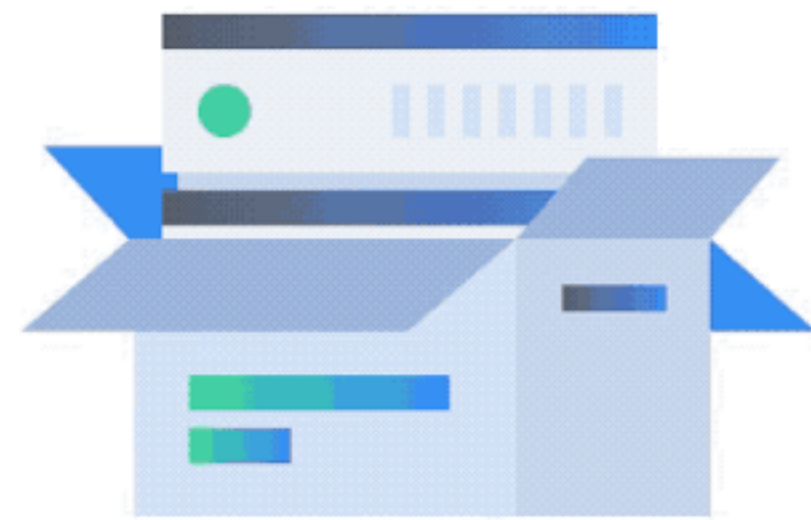
TARS

CLANG
DemoServer
HttpServer

Java
HelloServer

Nodejs
httpJson

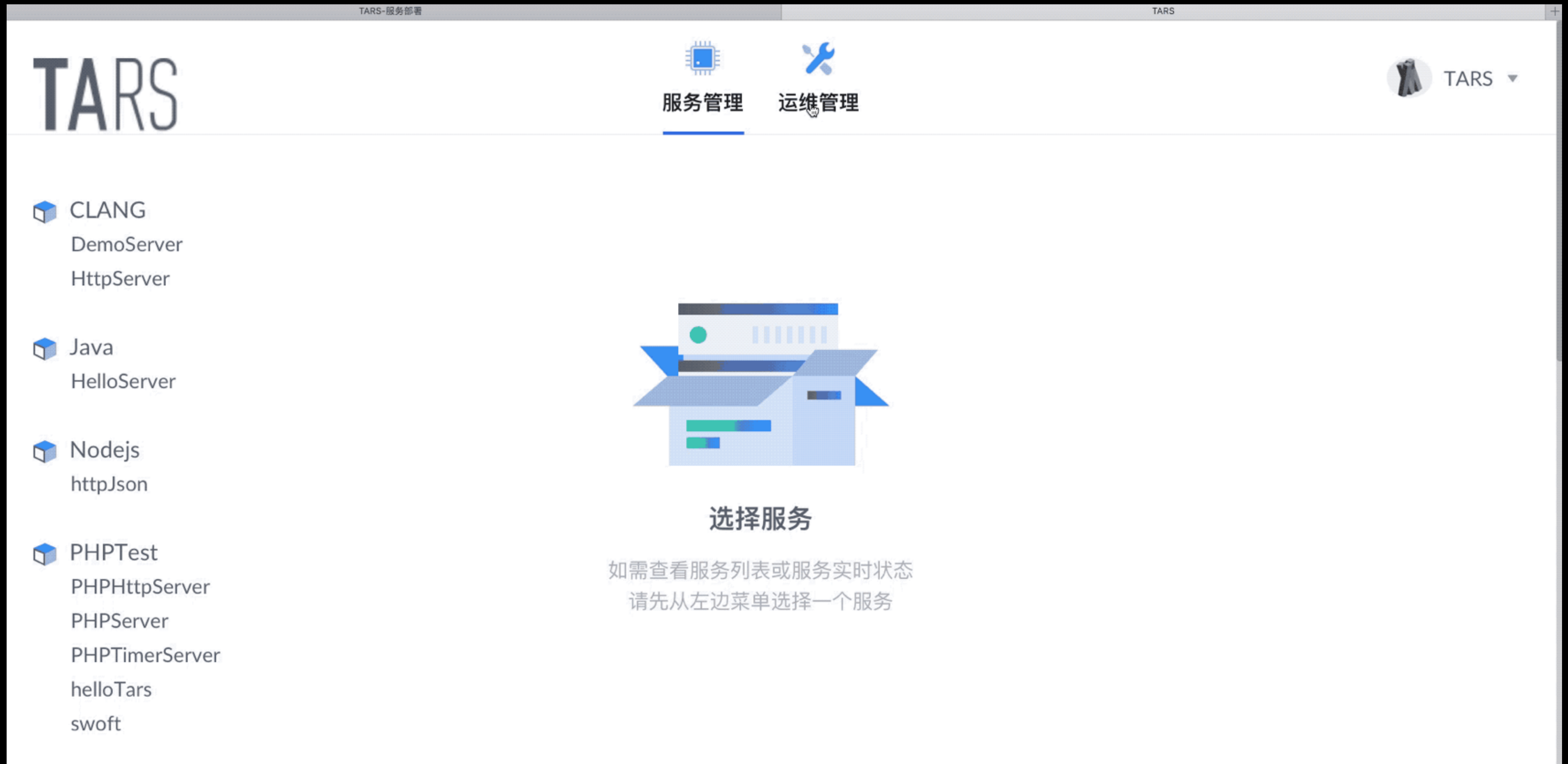
PHPTest
PHPHttpServer
PHPServer
PHPTimerServer
helloTars
swift



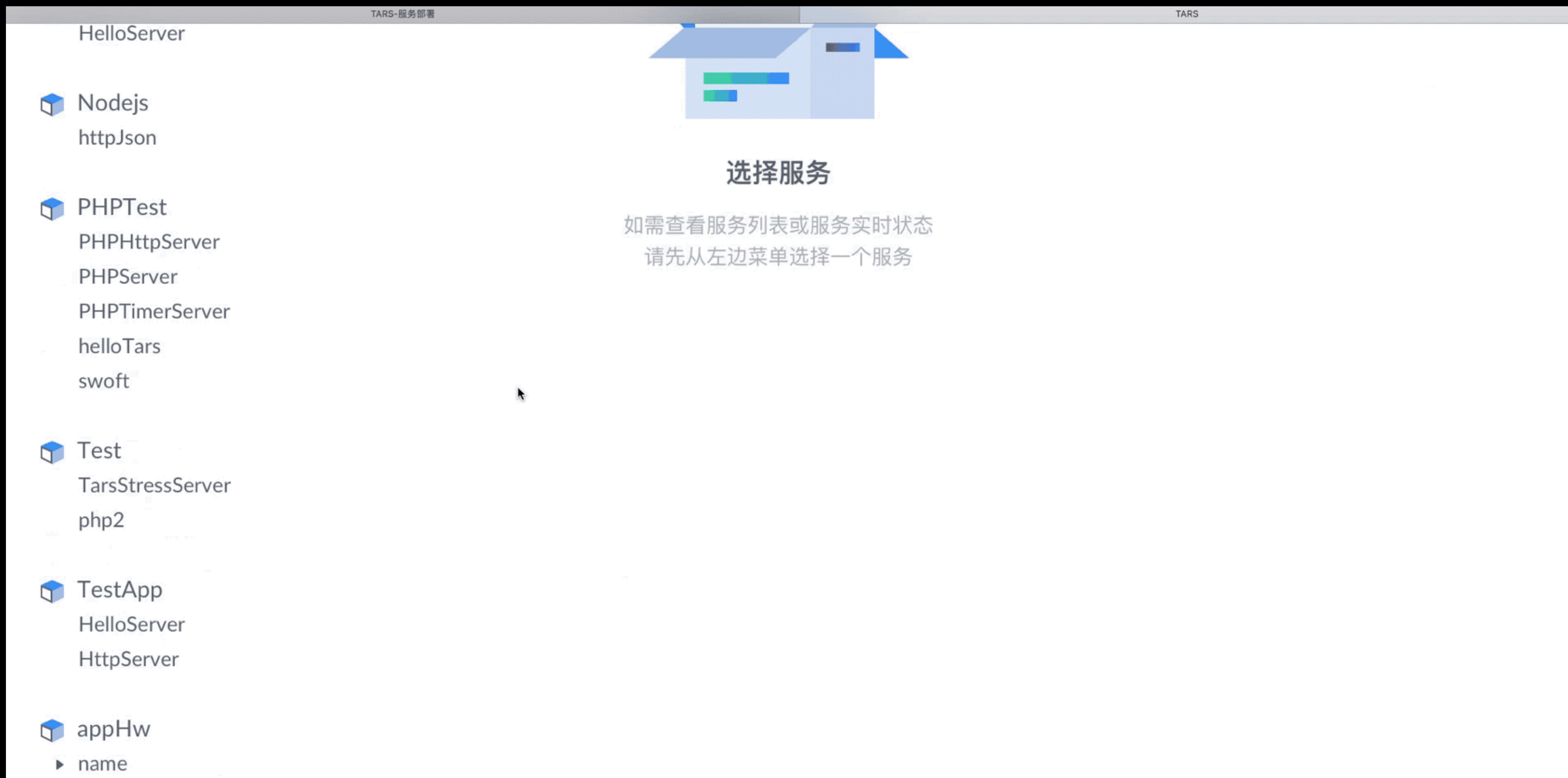
选择服务

如需查看服务列表或服务实时状态
请先从左边菜单选择一个服务

TARS-运营与平台



TARS-公共组件



TARS-协议



```
struct LoginInfo {  
    0 require string sid;  
    1 require string code;  
};  
  
struct ProfileInfo {  
    0 require string nick;  
    1 require int level;  
};  
  
interface TestServant {  
    int test(int qq, LoginInfo li, out ProfileInfo pi);  
}
```

二进制协议

语言无关

接口描述

工具转化代码

TARS-协议



```
struct RequestPacket {  
    1 require short iVersion;  
    2 optional byte cPacketType;  
    3 optional int iMessageType;  
    4 require int iRequetId;  
    5 require string sServantName;  
    6 require string sFuncName;  
    7 require vecotr<byte> sBuffer;  
    8 optional int iTimeout;  
    9 optional map<string,string> context;  
    10 optional map<string,string> status;  
};
```

协议版本

请求ID

服务路由信息

接口信息

二进制数据包

TARS-协议



```
struct ResponsePacket {  
    1 require short iVersion;  
    2 optional byte cPacketType;  
    3 require int iRequetId;  
    4 optional int iMessageType;  
    5 optional int iRet;  
    6 require vecotr<byte> sBuffer;  
    7 optional map<string,string> status;  
    8 optional string sResultDesc;  
};
```

协议版本

请求ID

返回码

二进制数据包

其他描述

TARS

 微服务

服务自动发现

智能调度

容灾容错

柔性熔断

路由与灰度

 协议

私有二进制协议

支持扩展协议

TARS

多语言

C++

JAVA

NodeJS

PHP

Python

运维体系

服务监控

主调上报

特性上报

命令下发

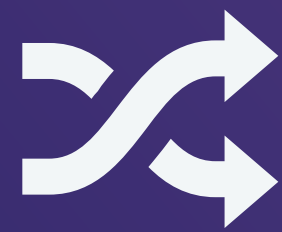
发布回滚

TARS-PHP



功能完善

对标现有C++、
Java、NodeJS体
系功能



灵 活

论灵活，谁与PHP
争锋？



轻 量

最轻量设计，点
到为止，即插即用



高 效

插上SWOOLE2.0
协程的翅膀，不得
不飞

TARS-PHP

TARS-CLIENT

配置统一

一键调用

多种网络能力

TARS-EXT扩展

高性能

易用

稳定



TARS-SERVER

服务三合一

灵活

轻量

开发效率

让代码写代码

省出来的时间去喝茶

TARS-CLIENT

✓ 自动寻址

无需关心服务地址

✓ SOCKET

后台系统一样接入

✓ 主调上报

服务情况一览无遗

✓ SWOOLE同步

SWOOLE网络能力

✓ 远程日志

无需本地文件写入

✓ SWOOLE协程

2.0协程更快



```
$config = new CommunicatorConfig();
```

```
$config->setLocator("tars.tarsregistry.QueryObj@tcp -h 172.16.0.161 -p 17890");
```

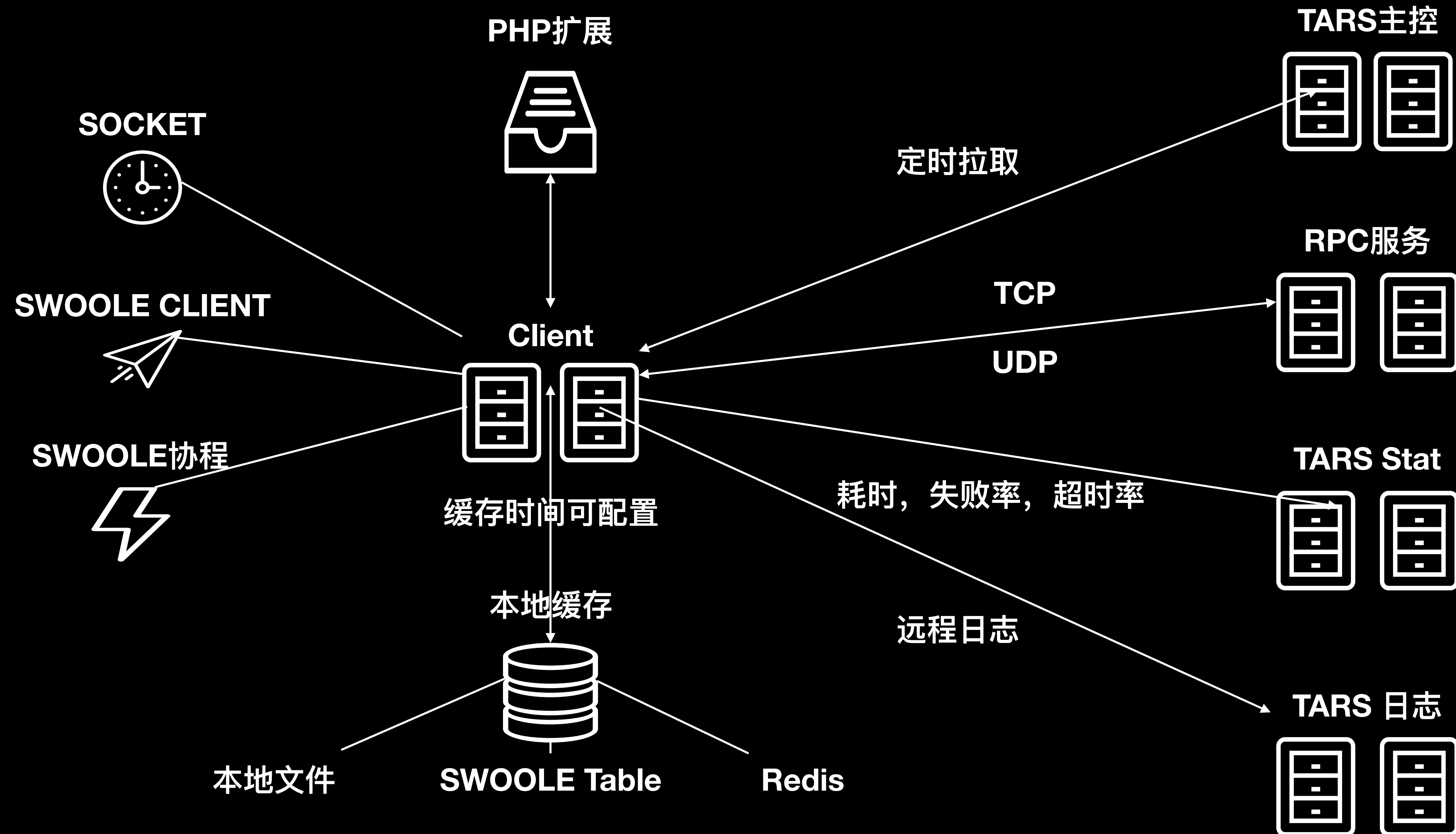
```
$config->setModuleName("PHPTest.PHPHttpServer");
```

```
$config->setSocketMode(SWOOLE_COROUTINE);
```

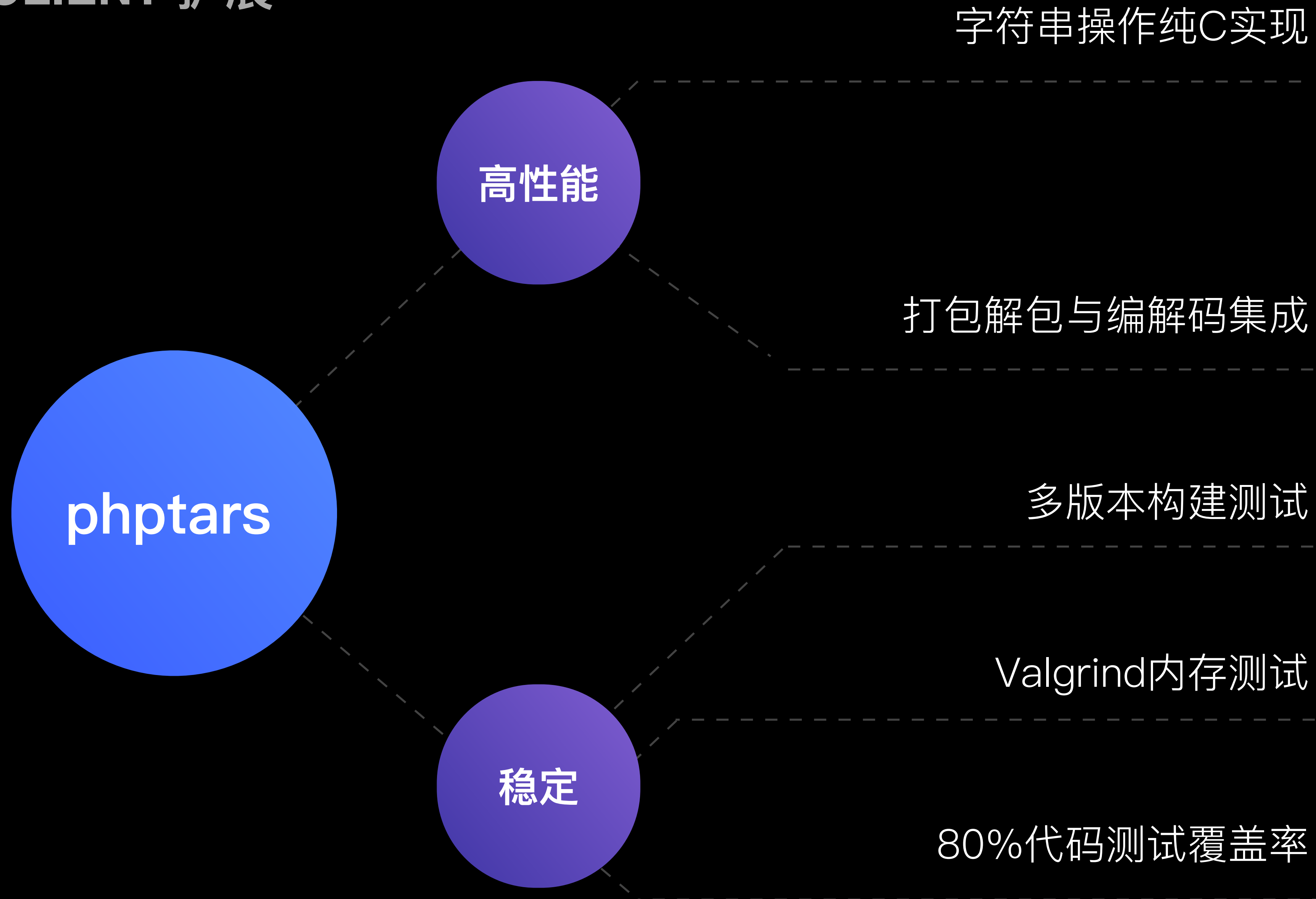
```
$servant = new TARSServant($config);
```

```
$result = $servant->rpcCall();
```

TARS-CLIENT



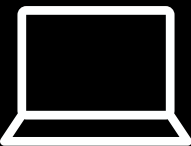
TARS-CLIENT-扩展



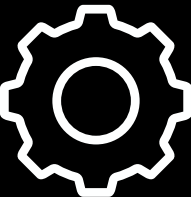
TARS-CLIENT

方式/100次	协议复杂度	打包时间(ms)	倍数	解包时间(ms)	倍数
phptars扩展	简单	0.69	1	1.18	1
纯PHP实现	简单	11.25	16	16.28	14
phptars扩展	复杂	1.17	1	1.55	1
纯PHP实现	复杂	14.5	12	15.1	9

TARS-CLIENT



流量统计



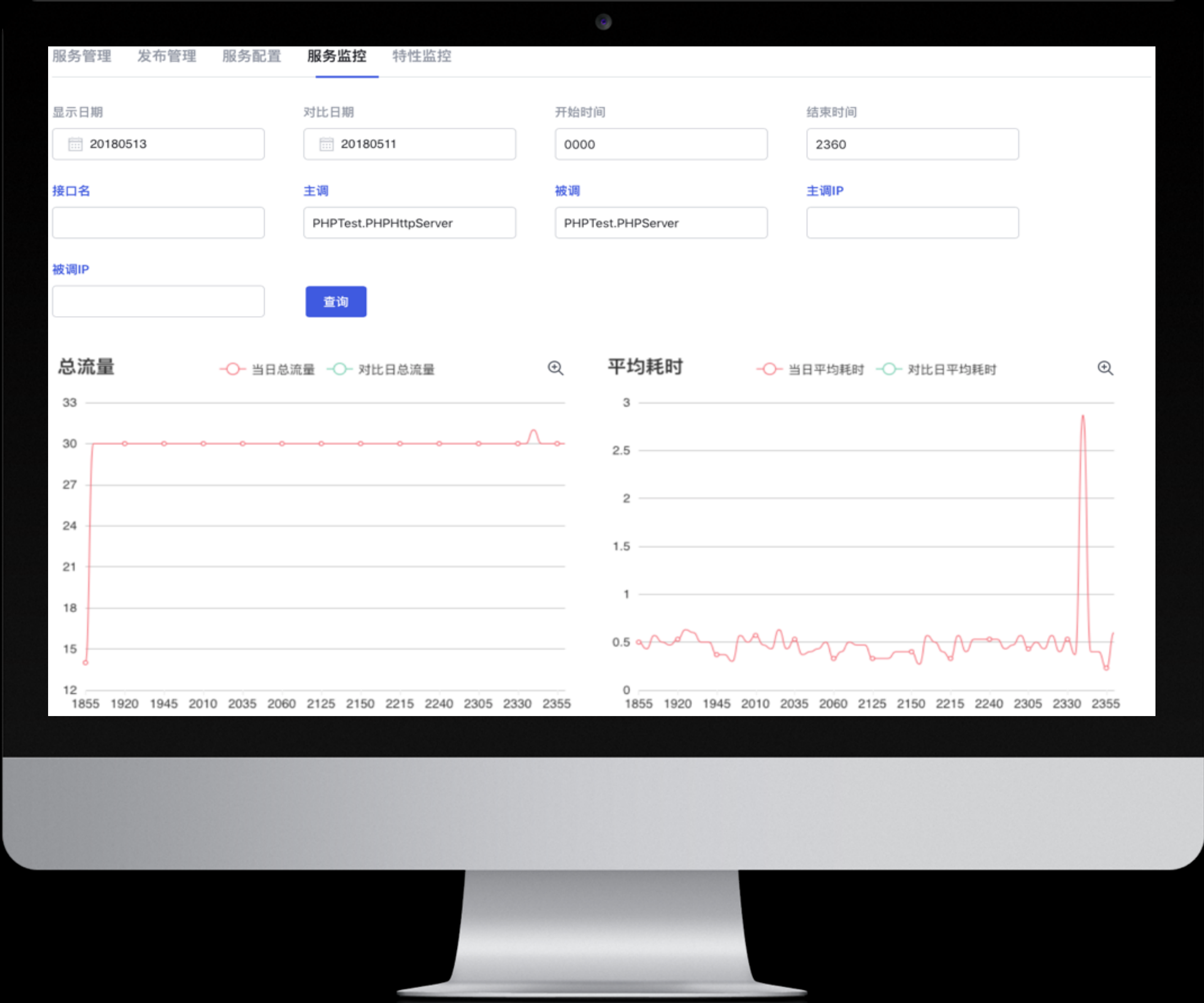
服务性能



接口耗时



时间对比



TARS-SERVER



TARS-SERVER

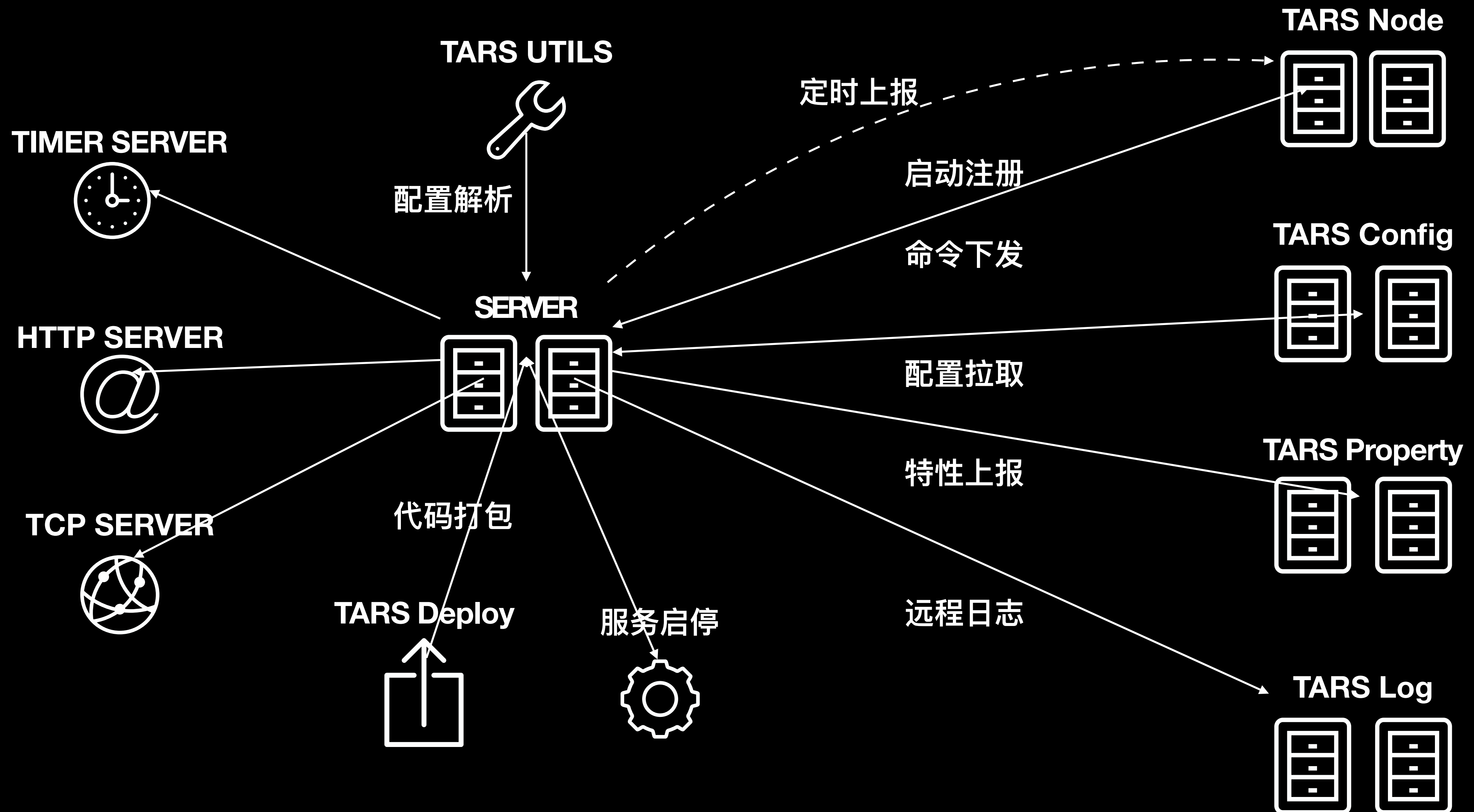
TARS-TIMER-SERVER

3 in 1

TARS-HTTP-SERVER

TARS-TCP-SERVER

TARS-SERVER



TARS-TCP-SERVER



SWOOLE

1.x

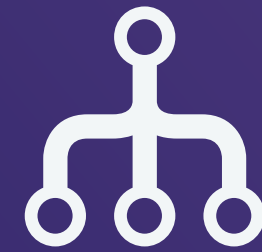
2.x – 推荐

PHP 5.6+



灵活配置

支持SWOOLE配置
服务入口灵活指定



基于注解路由

注解预生成
服务启动加载
运行时路由



管理端口

额外启动
配置下发
自定义命令



协议可扩展

通信协议
编码协议

TARS-HTTP-SERVER

HTTP
SERVER



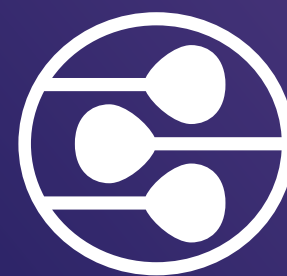
GET/POST

支持GET参数
POST FORM DATA
POST JSON



Cookie/Status

支持COOKIE、STATUS等
HTTP标准返回



ROUTE

极简路由/Index/Index
可扩展



FILE

文件上传
大小以SWOOLE配置为限



Service Detect

提供服务探测接口
结合服务本身上报

TARS-TIMER-SERVER



```
namespace TimerServer\timer;

class OneMinuteTimer {

    public $interval;

    public function __construct()
    {
        $this->interval = 60000;
    }

    public function execute()
    {
        error_log("Timer execute start at time:".time());

        // 业务逻辑

        error_log("Timer execute end at time:".time());
    }
}
```

- ✓ 时间灵活控制
- ✓ Timer统一管理
- ✓ Timer自启动
- ✓ Timer数量随意伸缩

TARS-SERVER

TARS

服务管理

运维管理

TARS

CLANG

DemoServer

HttpServer

Java

HelloServer

Nodejs

httpJson

PHPTest

PHPHttpServer

PHPServer

PHPTimerServer

helloTars

服务管理

发布管理

服务配置

服务监控

特性监控

服务列表

服务	节点	Se t	设置状态	当前状态	进程 ID	版本	发布时间	操作
PHPS erver	172.16.0.116	未 启 用	•Active	•Active	16564	171	2018-05-16 10:57:35	编辑 重启 停止 管理Servant 更多命令
PHPS erver	172.16.0.161	未 启 用	•Active	•Active	2501	171	2018-05-12 16:12:59	编辑 重启 停止 管理Servant 更多命令

服务实时状态

- ✓ 服务管理
- ✓ 发布管理
- ✓ 服务配置
- ✓ 服务监控
- ✓ 特性监控

TARS-SERVER 发布

TARS-服务部署

TARS

TARS

服务管理 运维管理

CLANG
DemoServer
HttpServer

Java
HelloServer

Nodejs
httpJson

PHPTest
PHPHttpServer
PHPServer
PHPTimerServer
helloTars
swift

服务管理 发布管理 服务配置 服务监控 特性监控

服务列表

服务	节点	Set	设置状态	当前状态	进程 ID	版本	发布时间	操作
PHPServer	172.16.0.116	未启用	• Off	• Off	0	171	2018-05-17 14:47:39	编辑 重启 停止 管理Servant 更多命令
PHPServer	172.16.0.161	未启用	• Active	• Active	22968	195	2018-05-17 14:47:56	编辑 重启 停止 管理Servant 更多命令

服务实时状态

TARS-SERVER 重启

TARS-服务部署

TARS

TARS

服务管理

运维管理

TARS

CLANG

DemoServer

HttpServer

Java

HelloServer

Nodejs

httpJson

PHPTest

PHPHttpServer

PHPServer

PHPTimerServer

helloTars

swift

服务管理

发布管理

服务配置

服务监控

特性监控

服务列表

服务	节点	Set	设置状态	当前状态	进程 ID	版本	发布时间	操作
PHPServer	172.16.0.116	未启用	• Off	• Off	0	171	2018-05-17 14:47:39	编辑 重启 停止 管理Servant 更多命令
PHPServer	172.16.0.161	未启用	• Active	• Active	27366	195	2018-05-17 14:47:56	编辑 重启 停止 管理Servant 更多命令

服务实时状态

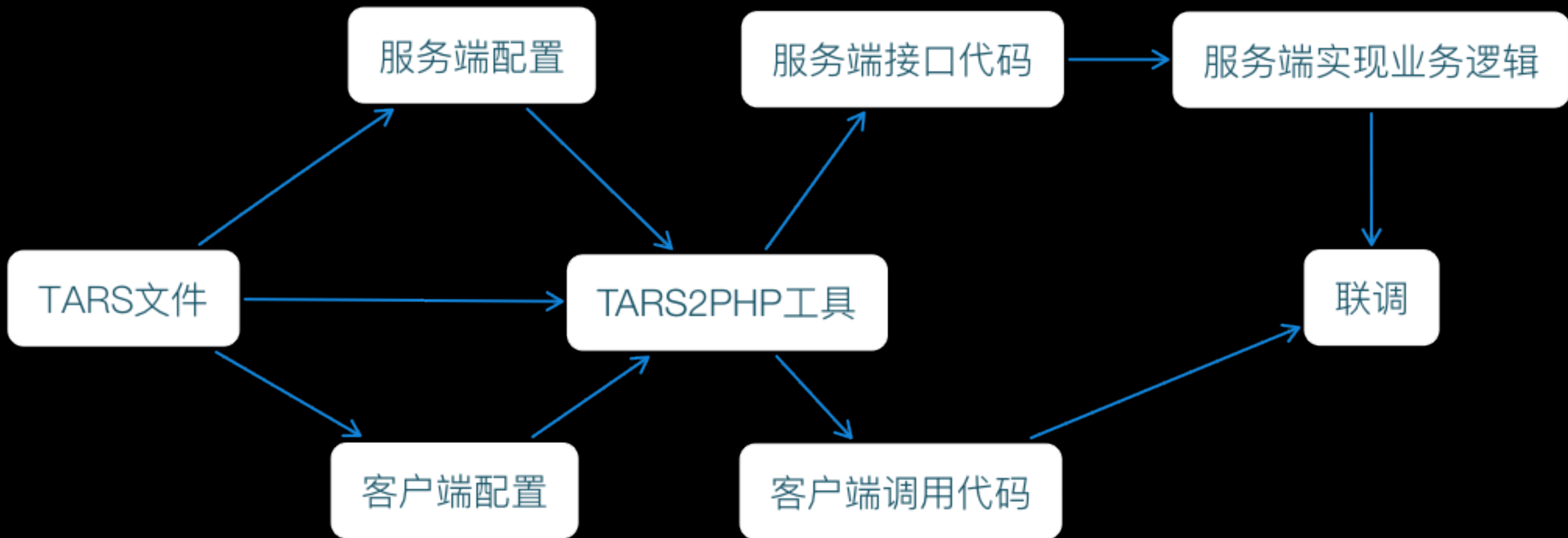
TARS-SERVER



TCP、HTTP
SERVER
性能评估
16核 64G

QPS	HTTP SERVER 5.6 同步CLIENT	HTTP SERVER 7.1 同步CLIENT	HTTP SERVER 7.1 协程CLIENT
服务空跑	19万	23万	
单次简单RPC	2万8	3万3	4万9
单次复杂RPC	1万8	2万1	3万
混合RPC调用	1万2	1万4	2万1

TARS2PHP - 开发效率



TARS2PHP - 开发效率

● ● ●

```
public function hello($name, $flag, OutStruct &$outStruct) {  
    try {  
        $requestPacket = new RequestPacket();  
        $requestPacket->_iVersion = $this->_iVersion;  
        $requestPacket->_funcName = __FUNCTION__;
```

● ● ●

```
/**  
 * @param string $name  
 * @param bool $flag  
 * @param struct $outStruct \Server\servant\PHPTest\PHPServer\obj\classes\OutStruct =out=  
 * @return int  
 */  
public function hello($name, $flag, OutStruct &$outStruct);
```

```
        $sBuffer = $this->_communicator->invoke($requestPacket,$this->_iTimeout);
```

```
        $ret = TUPAPIWrapper::getStruct("outStruct",3,$outStruct,$sBuffer,$this->_iVersion);
```

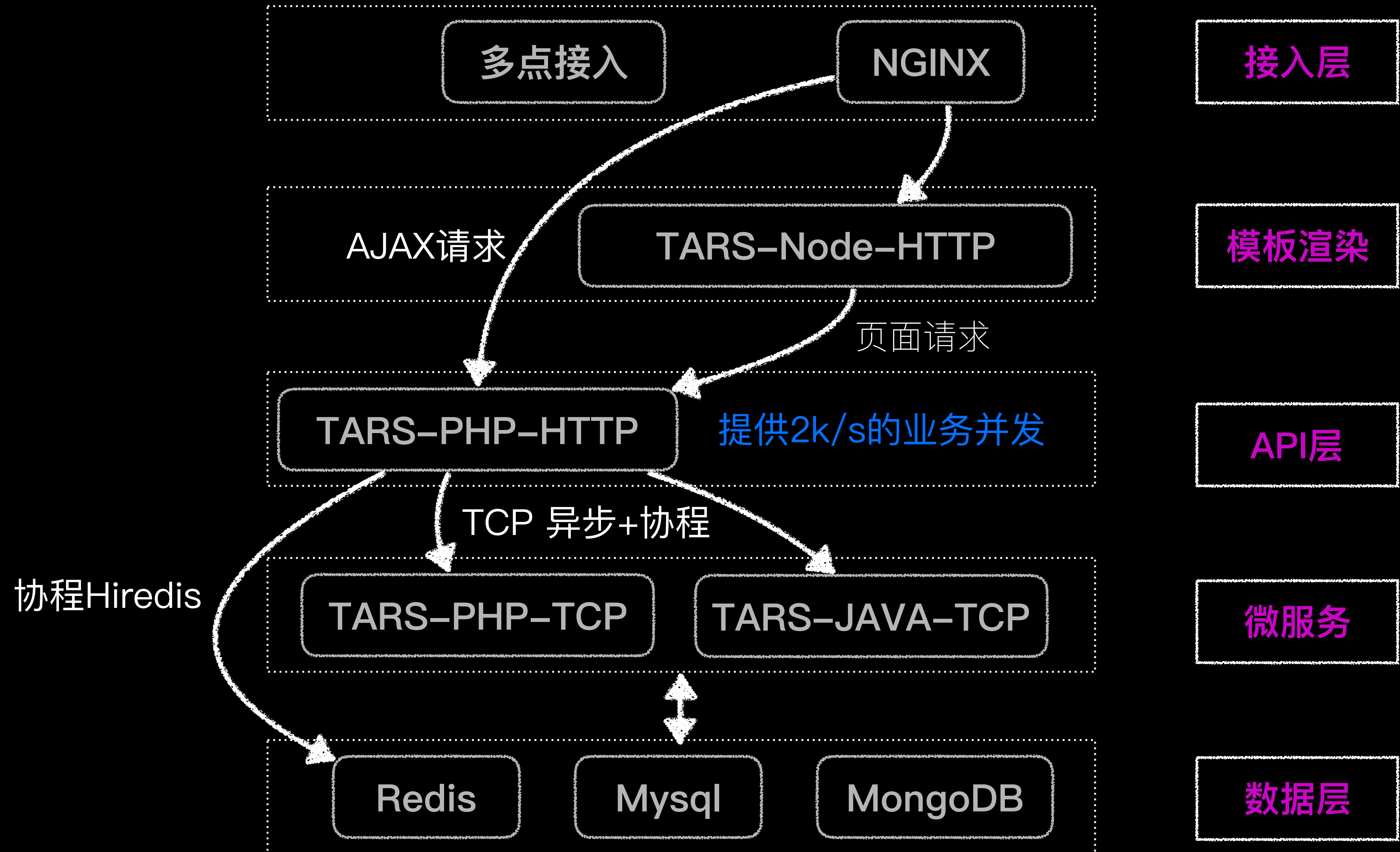
```
        return TUPAPIWrapper::getInt32("",0,$sBuffer,$this->_iVersion);
```

```
    }  
    catch (\Exception $e) {  
        throw $e;
```

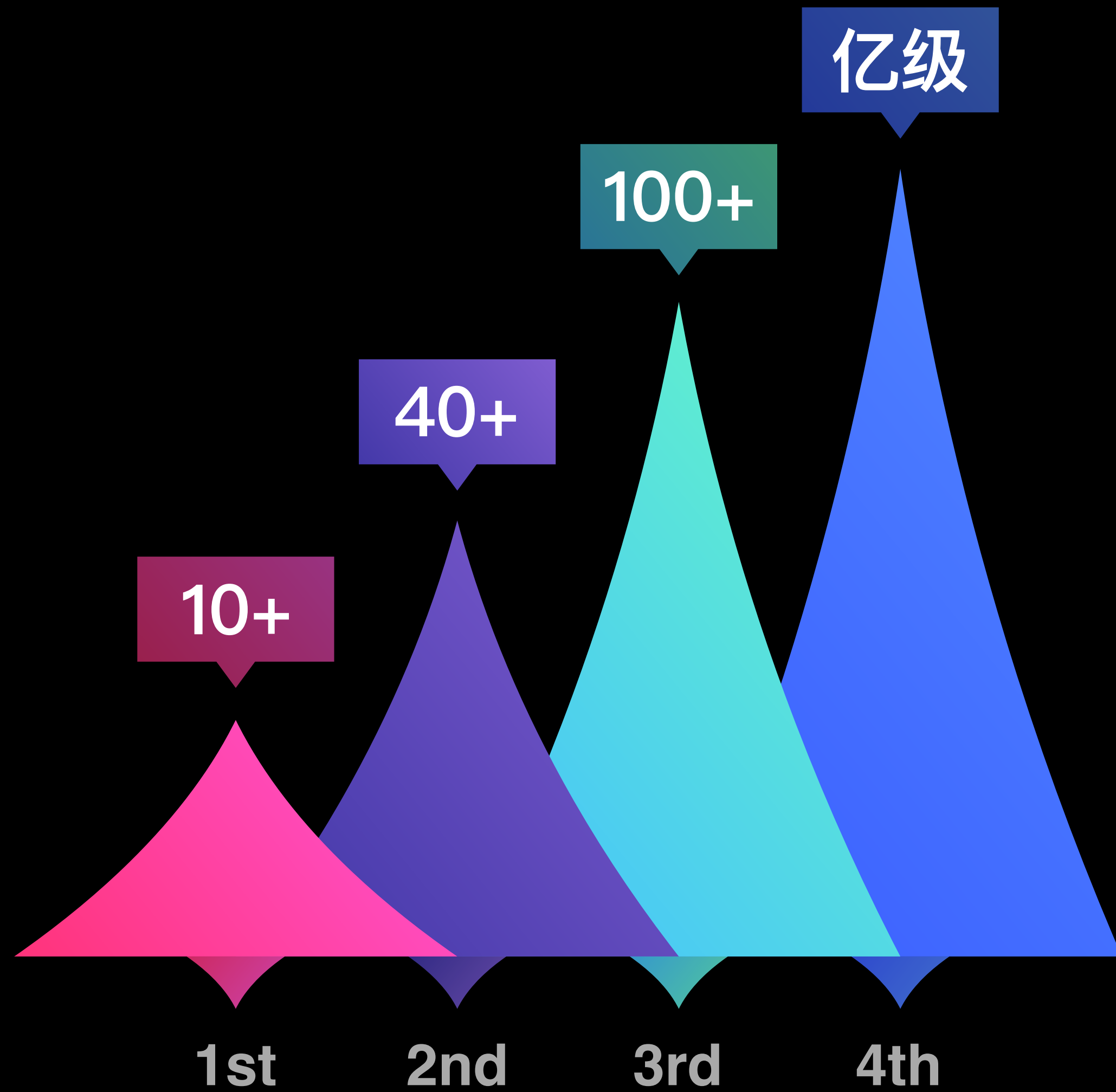
```
    }
```

```
}
```

TARS-PHP 在阅文



TARS-PHP 在阅文



- 模板拼接的Node HTTP服务10+
- 逻辑API层PHP服务与定时服务40+
- 后台TCP服务100+
- 每日亿级调用

TARS-PHP 模块化设计

phptars/tars*系列模块

phptars/tars-server

phptars/tars-client

phptars/tars-config 配置拉取

phptars/tars-log 远程日志

phptars/tars-monitor 监控上报

phptars/tars-registry 服务寻址

phptars/tars-report 服务保活

phptars/tars2php 代码生成

phptars/tars-utils 配置解析

TARS-PHP In Future

支持更多框架

提高稳定性

丰富CLIENT能力

SERVER优化

TARS-PHP



GITHUB搜索TARS

加入开源

共同建设

github.com/Tencent/Tars