



Swoole2.0原生协程高性能开发实践

chalesi



关于我

- 2011年-至今 腾讯上海 - SNG即通综合部
 - 企业QQ
 - QQ公众号
 - QQ看点
 - 腾讯开源联盟会(TOSA) PHP 发起人
- 开源社区
 - swoole核心开发组成员
 - swoole2.0开发小组leader
 - TSF创始人



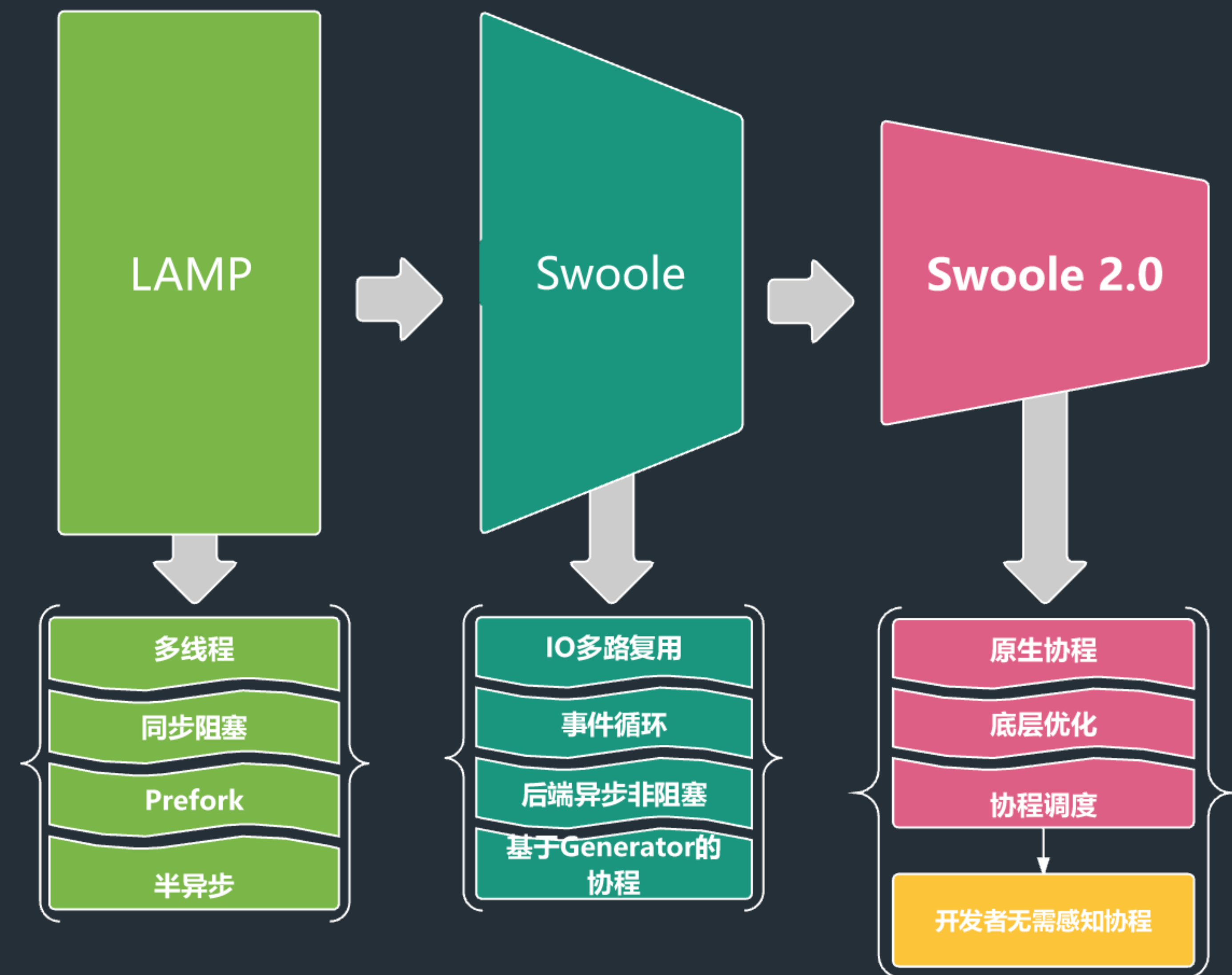


大纲

- Swoole2.0原生协程探索经验
- PHP高可用应用层框架建设
- QQ看点图文Web前后端优化之路



PHP的并发IO之路



$$\text{TPS} = \frac{\text{并发数}}{\text{平均响应时间}}$$

Generator

```
foo($params, function($params){  
    bar($params, function($params){  
        barr($params, function($params){  
            //.....  
        });  
    });  
});
```

```
yield foo($params);  
yield bar($params);  
yield barr($params);
```

?

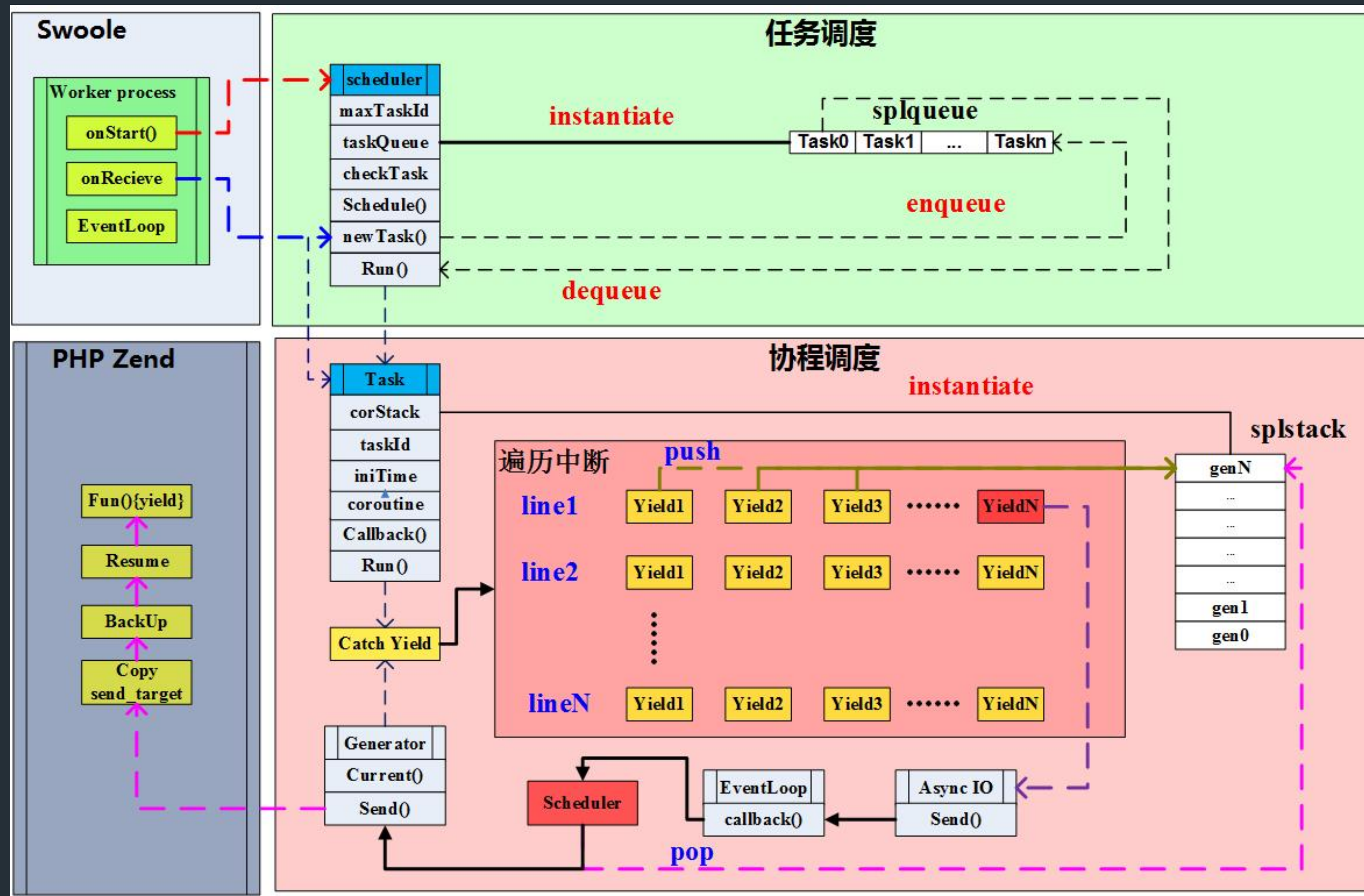


- 什么是协程?
- 为什么协程可以同步开发异步非阻塞程序?

```
$params = foo(' test' );  
$data = yield tcp_send($params);  
$return = barr($data);
```

- 基于yield如何实现协程?
 - PHP堆栈的分离, 保存
 - Yield实现协程切出
 - Generator双向通信

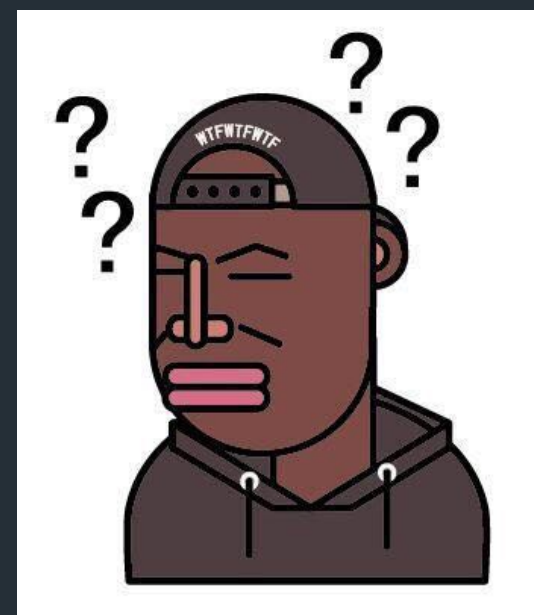
Yield协程





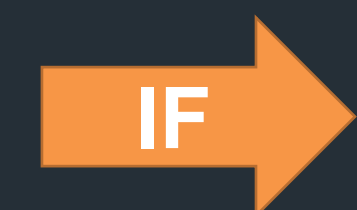
Yield问题

- 开发效率低，入门门槛高
 - 这里什么要yield？
 - 为什么代码没反应？
- 内存分配和运行低效

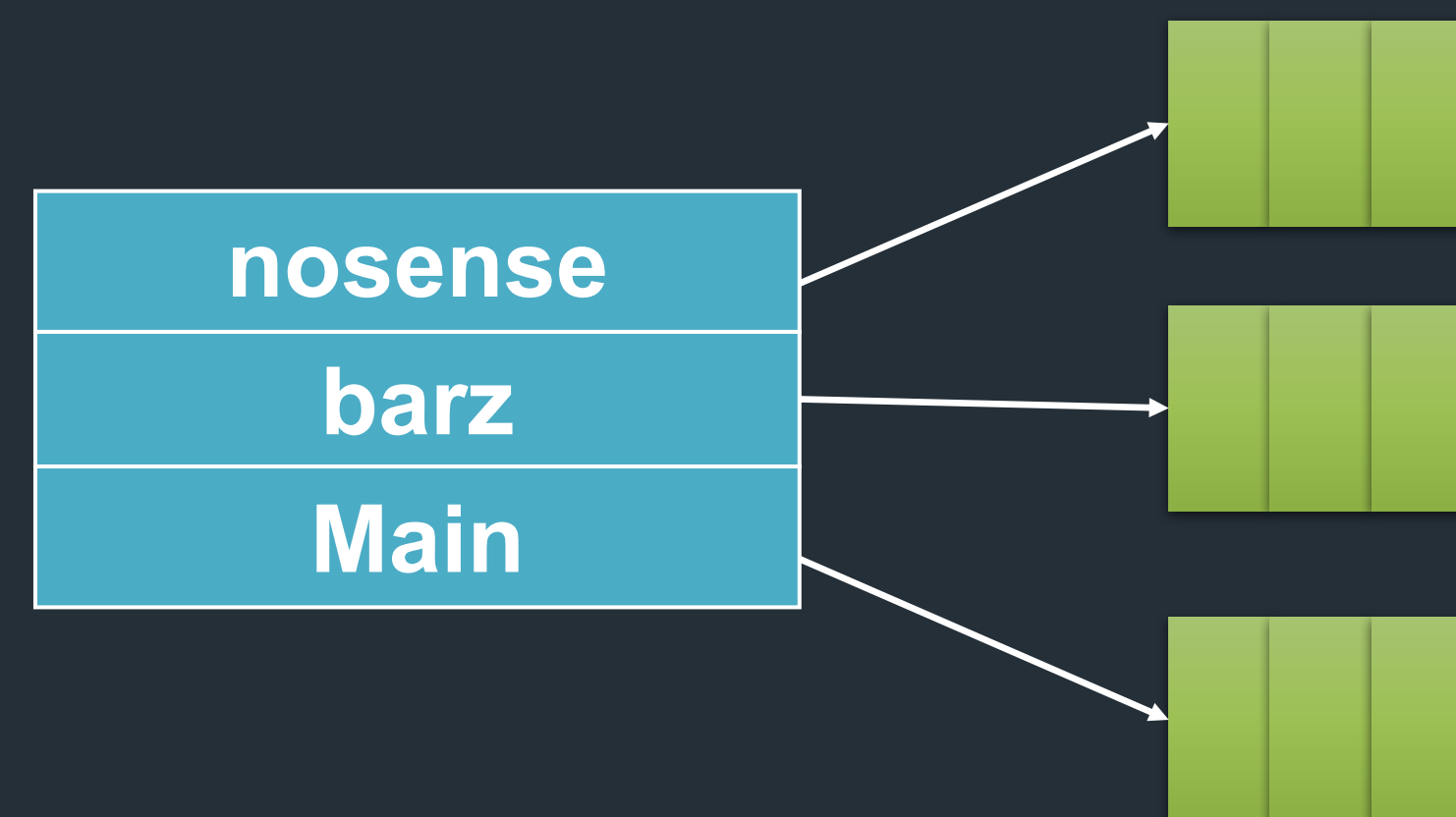


```
foo($params);  
yield bar($params);  
baz($params);
```

```
function bar($params)  
{  
    yield $client->send();  
}  
function baz($params)  
{  
    return nonsense($params);  
}  
Function nonsense($params)  
{  
    return "nonsense";  
}
```



```
Function nonsense($params)  
{  
    yield $client->send();  
}
```

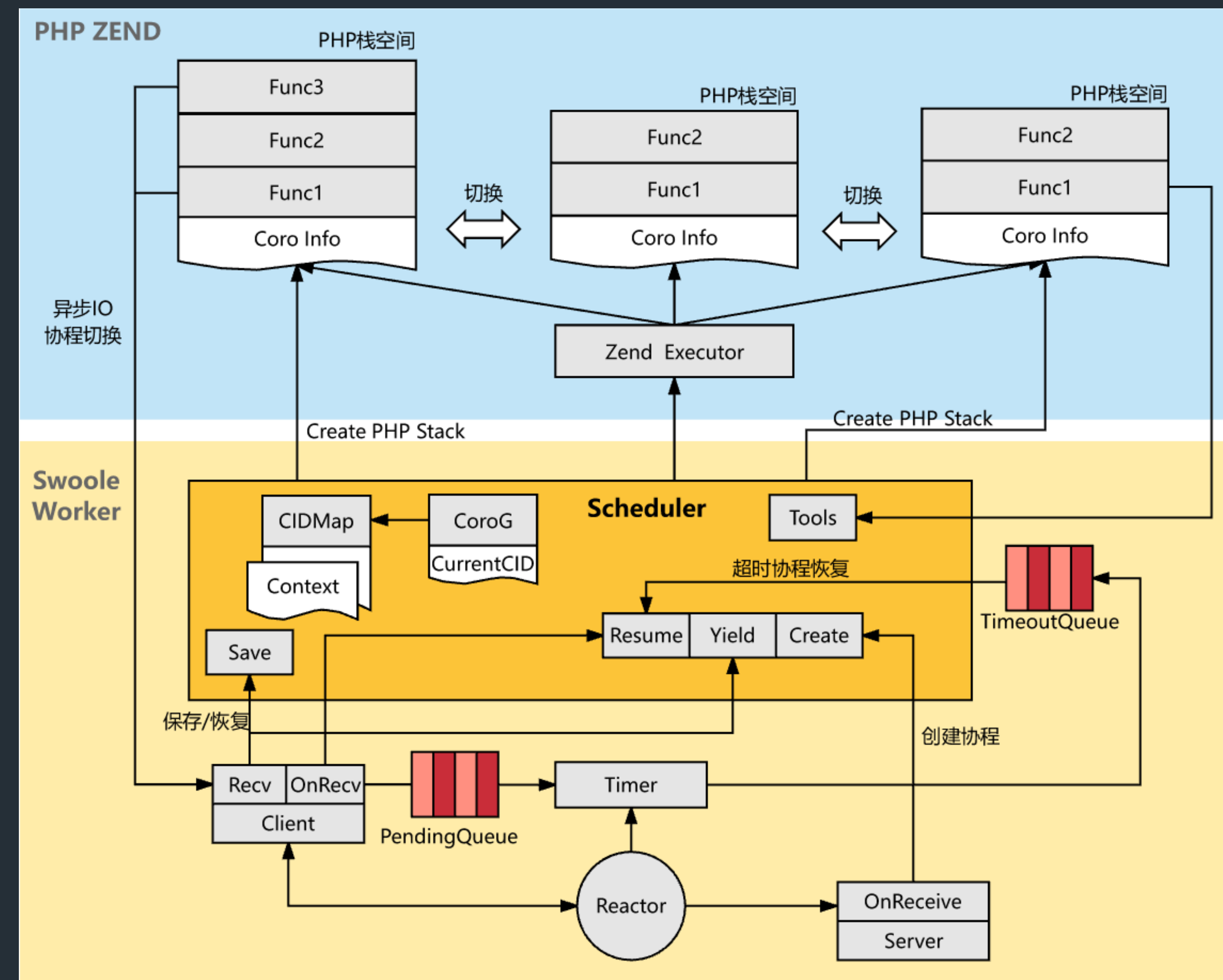


对协程的认知被转变成了对Generator语法和原理的认知



原生协程

- 栈分离—打破迭代器限制
- swoole底层分配协程栈
- 一个请求对应一个协程
- 为协程统一分配连续栈空间
- Scheduler控制协程切换





协程实践

TCP/UDP

```
$cli = new Swoole\Coroutine\Client(SWOOLE_SOCK_UDP);  
$ret = $cli->connect("127.0.0.1", 8888);  
$ret = $cli->send("hello world at the first time");  
$ret = $cli->recv(2);  
$cli->close();
```

Mysql

```
$cli = new Swoole\Coroutine\MySQL();  
$cli->connect(['host' => '127.0.0.1', 'user' => 'root',  
    'password' => '1', 'database' => 'test']);  
$ret = $cli->query('Select * from test limit 1');  
$cli->close();
```

HTTP

```
$cli= new Swoole\Coroutine\Http\Client('0.0.0.0', 9510);  
$cli->set(['timeout' => 1]);  
$cli->setHeaders([  
    "Host" => "api.mp.qq.com",  
    "User-Agent" => "Chrome/49.0.2587.3",  
]);  
$ret = $cliAA->get("/cn/token");
```

Redis

```
$redis = new Swoole\Coroutine\Redis();  
$res = $redis->connect('127.0.0.1', 6379);  
$res = $redis->set('key_tmp', 'Hello World');
```




协程实践

Multi-Call

```
$http_client= new Swoole\Coroutine\Http\Client('qq.com', 80);
$http_client->setDefer();
$http_client->get('/');
$mysql = new Swoole\Coroutine\MySQL();
$res = $mysql->connect(['host' => '192.168.244.128', 'user' => 'mha_manager',
                        'password' => 'mhapass', 'database' => 'tt']);

$mysql->setDefer();
$mysql->query('select sleep(1)', 2);
$res = $http_client->recv();
$mysql_res = $mysql->recv();
```



性能测试

压测环境： 8核+16G虚拟机 wrk c 200 d 30s HttpServer

Http长连接

	Swoole2.0	callback	PHP yield
Echo	540000	530000	80000
1次后端	83900	82400	41000
3次后端	31000	31000	20000

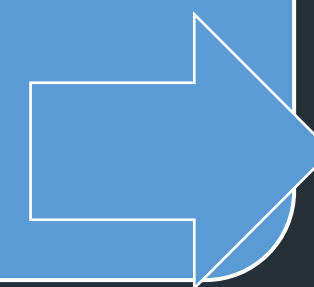
Http短连接

	Swoole2.0	callback	PHP yield
Echo	56000	55000	43000
1次后端	43600	43300	30000
3次后端	27000	27000	17000



Swoole

小而美



- 提供异步高性能IO
- 原生协程能力
- 通用原子能力封装

TSF

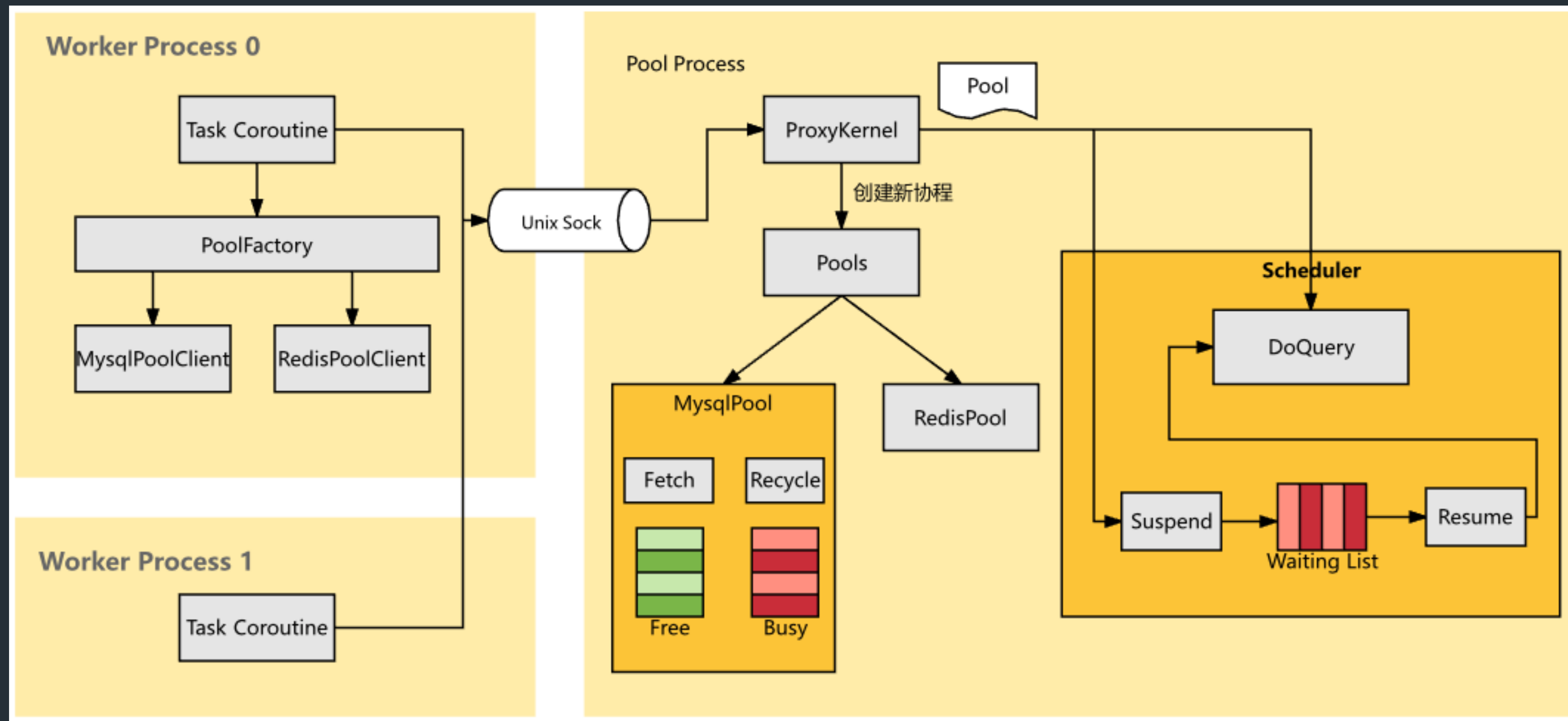
系统级高可用

- 快速搭建、高效开发
- 连接池方案
- 频控组件
- 过载保护
- 监控上报
- 组件扩展



连接池

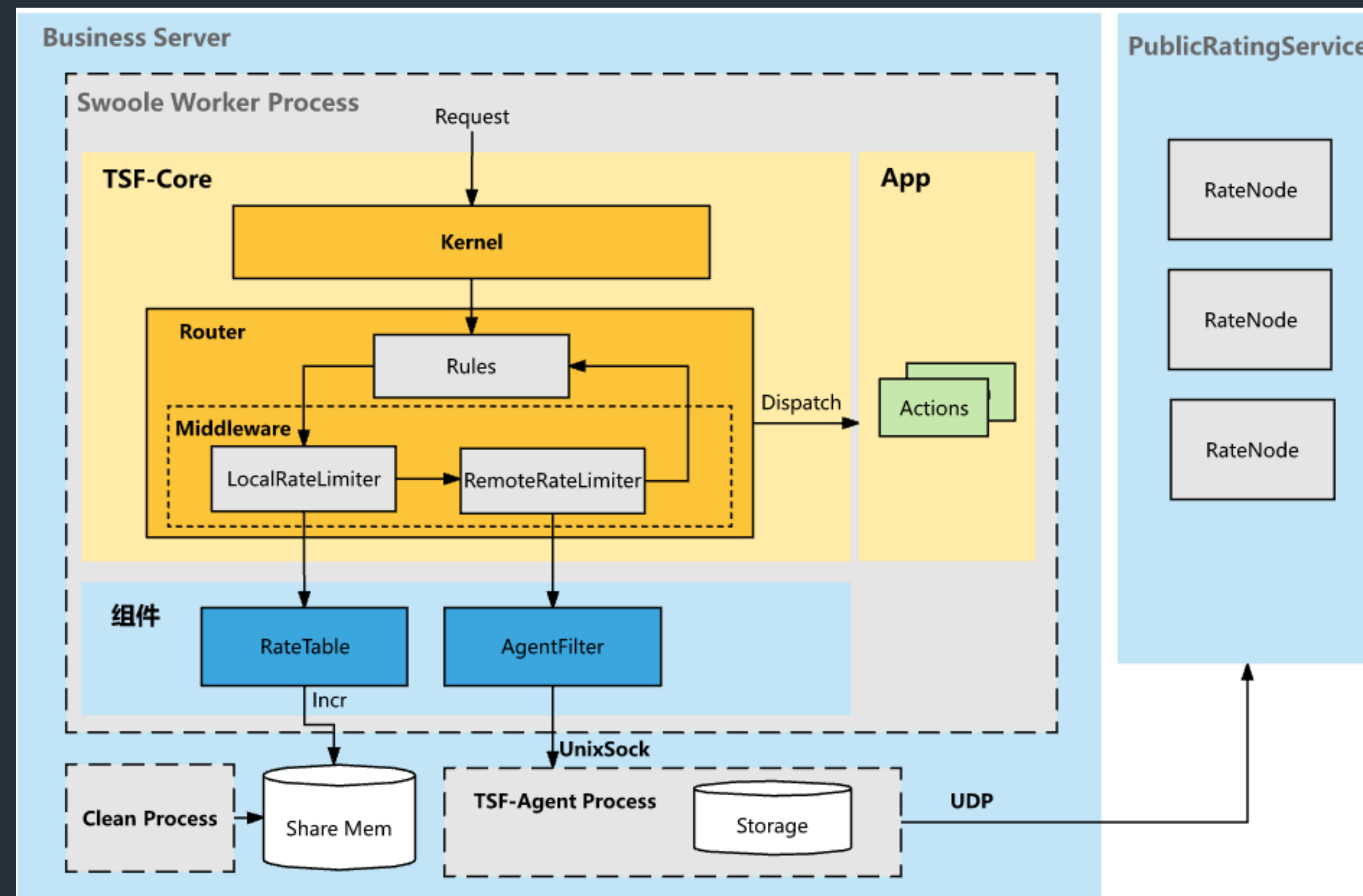
- 传统连接池为进程连接池，实现简单
 - 无法跨进程收敛连接
 - 服务定时重启带来短链攻击
- TSF3.0跨进程连接池，从worker层面收敛链接





频率控制

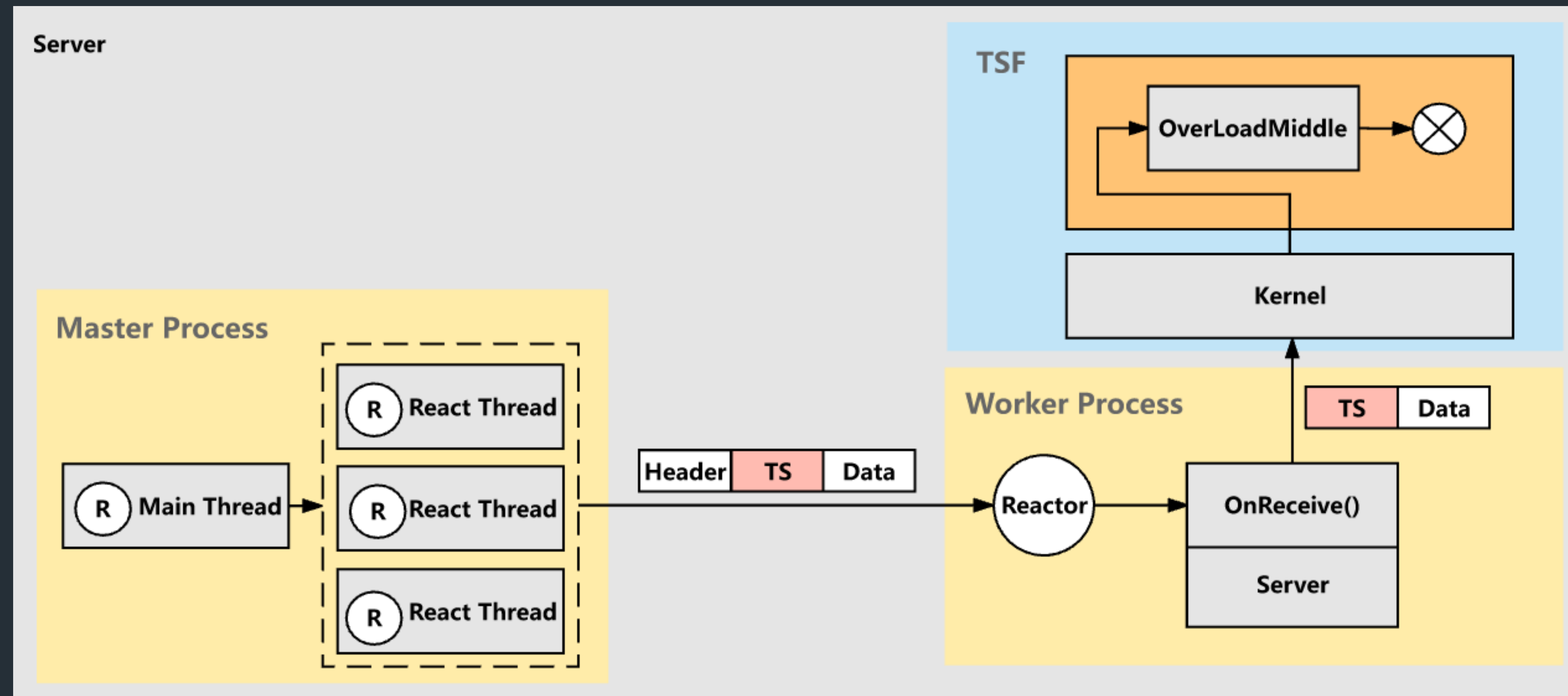
- 频率控制保护服务在有效负载范围内
- 保护后级服务免受流量冲击
- TSF3.0提供频控保护组件
 - 单机频控：防止单机高负载导致成功率下降
 - 联机频控：从服务模块维度控制访问频次





过载保护

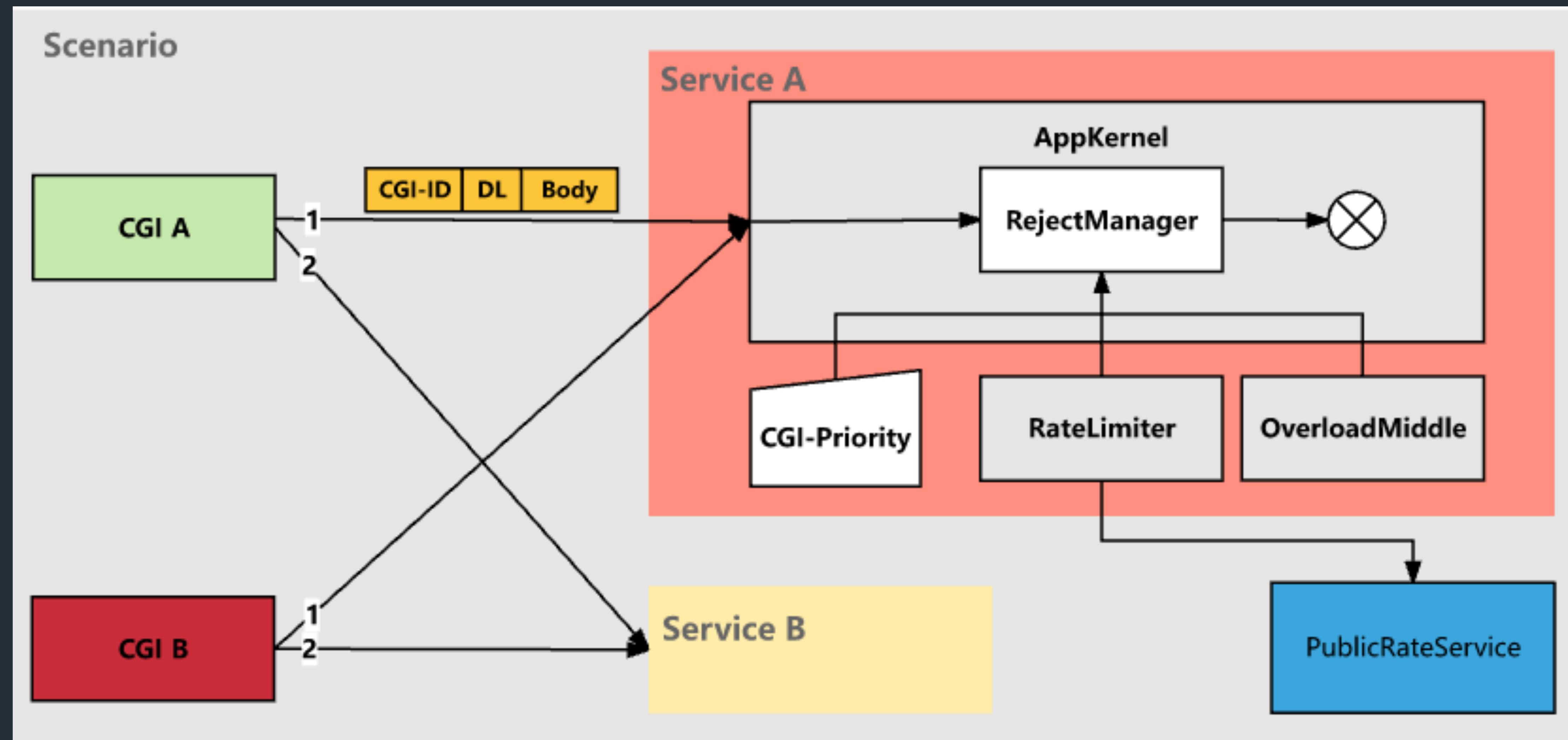
- 超负载高流量会导致整体服务雪崩
- 如何系统判断过载
- 单机过载保护：探测队列拥塞
 - Master进程标记请求收包时间
 - TSF根据收包时间戳判断队列拥塞情况





过载保护

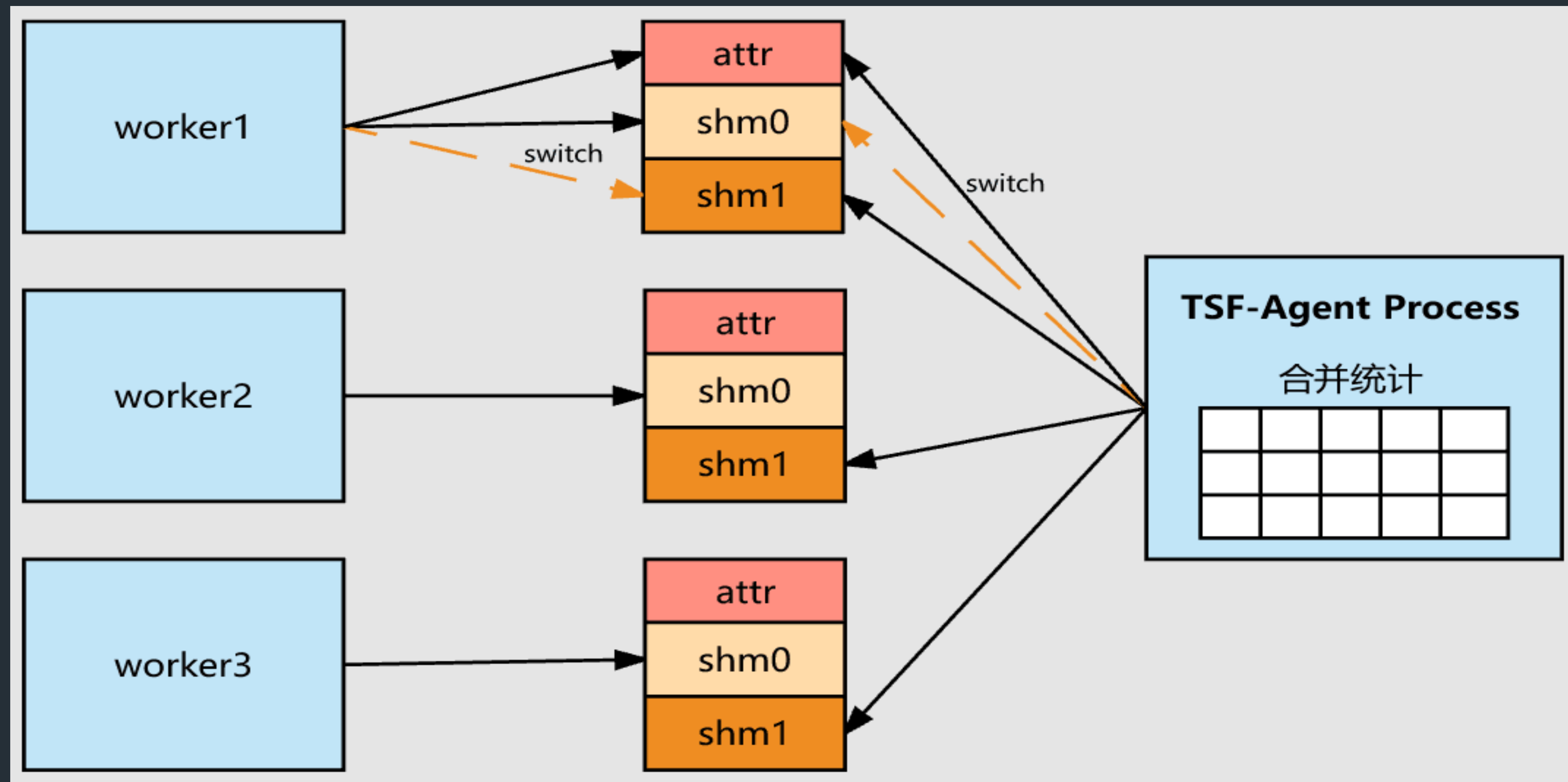
- 单机服务过载后粗暴丢弃
 - 后端依赖过载会导致整个调用链成功率下降
 - 单个CGI过载会导致其他CGI成功率降低
- 联机过载保护：从全局保证损失最小化
 - CGI优先级划分，保证高优先级能力
 - CGI协议指定后端deadline，放弃超时请求





服务监控

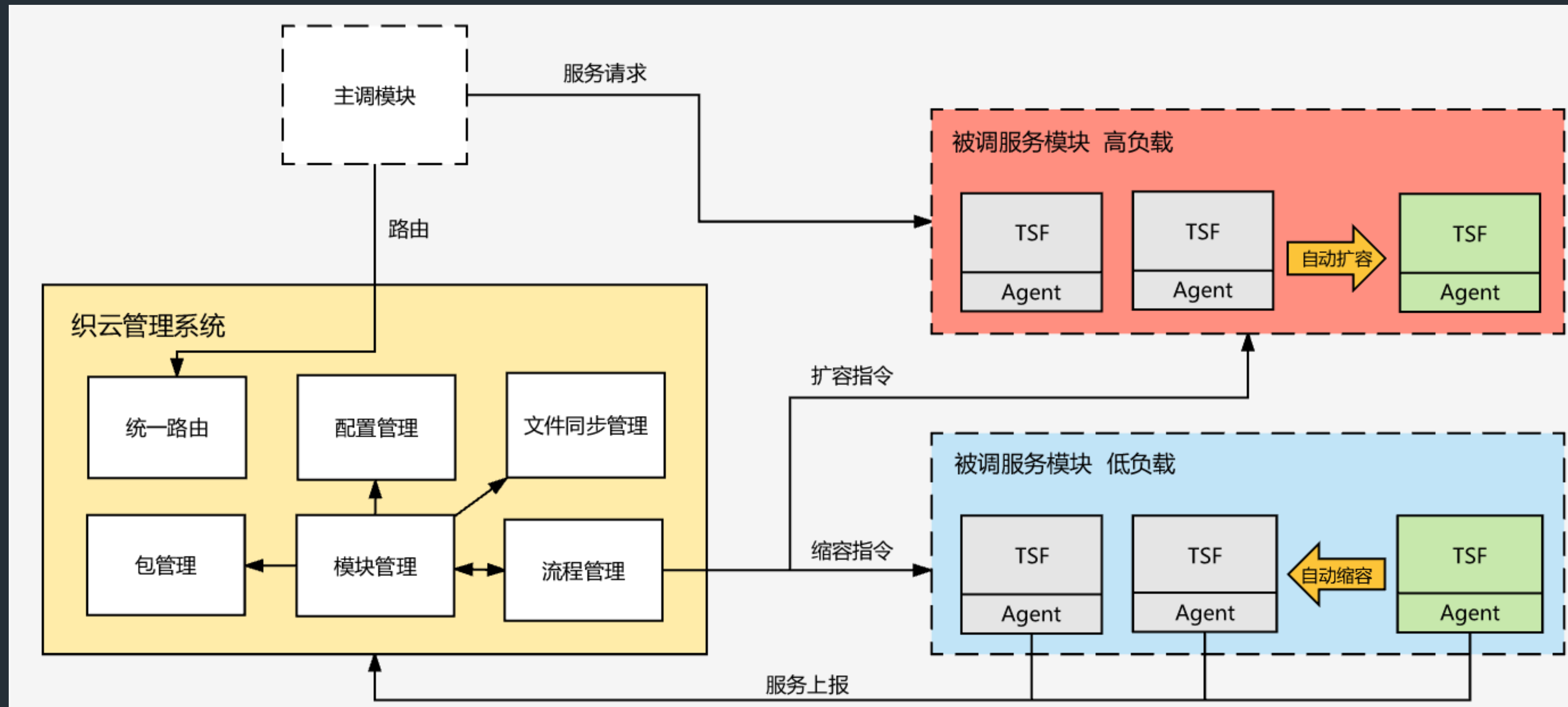
- 服务实时可用性由调用链上服务健康度决定
- TSF提供服务统一上报能力
- TSF-Agent读取worker的信息，高性能统计上报





集群动态管理

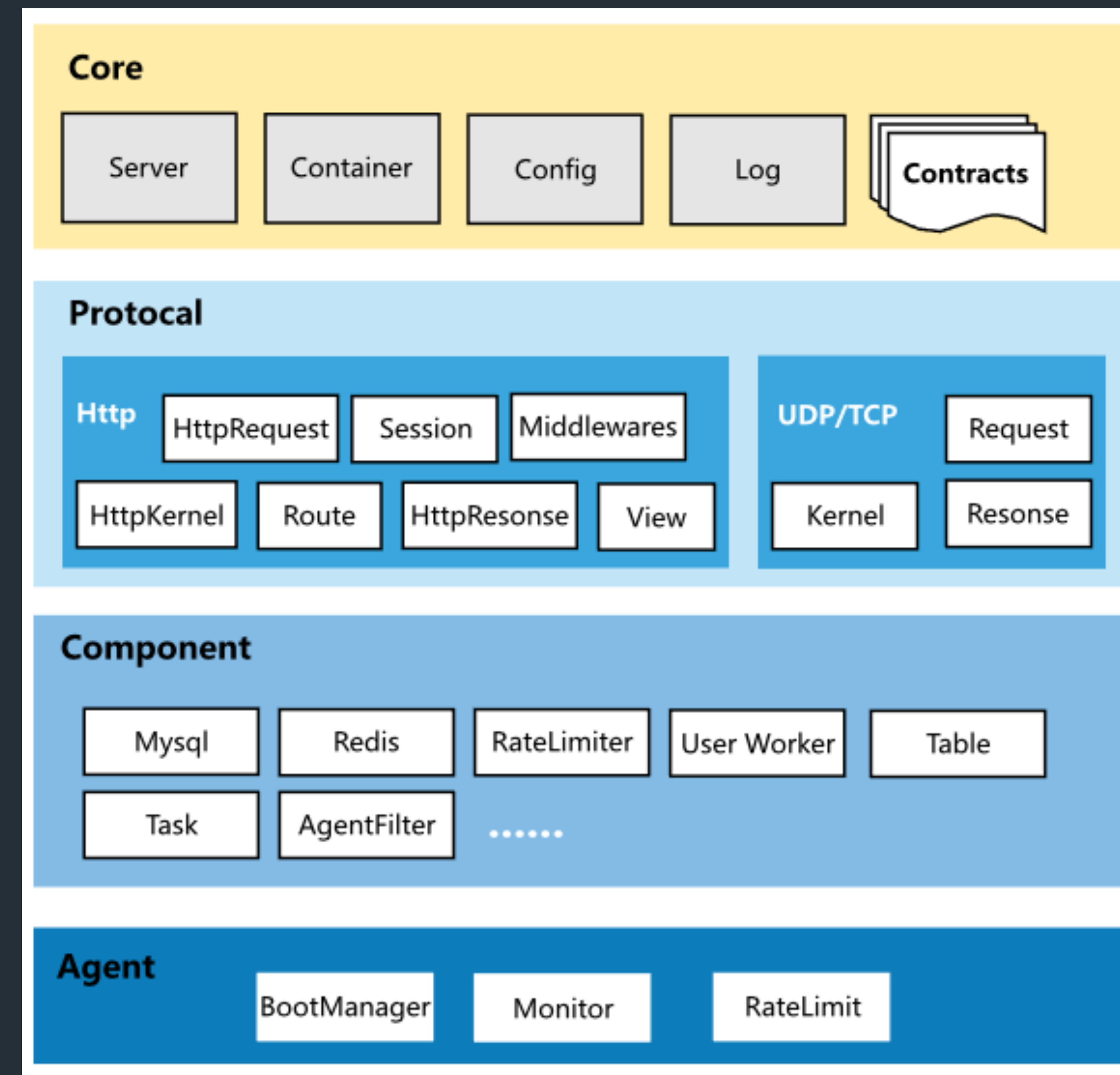
- TSF服务抽象为模块提供集群服务
- 模块内配置、系统一致化
- 织云系统提供管理服务
 - 模块监控
 - 自动部署
 - 弹性伸缩
 - 服务发现





TSF整体架构

- 核心层：抽象swoole基础网络能力
- 协议层：适配不同协议的网络接口
- 组件层：提供框架可扩展能力，封装具有独立能力的组件
- Agent服务：底层支持框架启停/监控/数据同步的能力





TSF实践

连接池:

```
'mysqlpool' => [  
    'conns' => [  
        'target_mysql' => [  
            'serverInfo' => [  
                'host' => '', 'user' => '', 'password' => '', 'database' => ''  
            ],  
            'maxSpareConns' => 10,  
            'maxConns' => 20  
        ],  
        'maxProxyConns' => 10,  
        'maxSpareProxyConns' => 5  
    ],  
],
```

```
$conn = TSF\Pool\Factory::fetch('mysql', 'target_mysql');  
$res = $conn->query('select version()');  
$conn->recycle();
```

单机频控:

```
Route::facade()->any('*', '*', [  
    'include' => [  
        'before' => ['TSF\Http\Middleware\RateLimiter'],  
    ]  
]);
```

User Worker:

```
"TSF\Component\UserWorker\UserWorkerManager" => [  
    [  
        "name" => "hippoworker",  
        "num" => 1,  
        "workerClass" => "\\App\UserWorker\HippoConsumer",  
    ]  
]
```




QQ看点图文页优化



性能

- 网络请求优化
- 终端预加载策略
- 加载策略优化

高可用

- 静态资源高可用
- 后台服务高可用
- 全链路监控

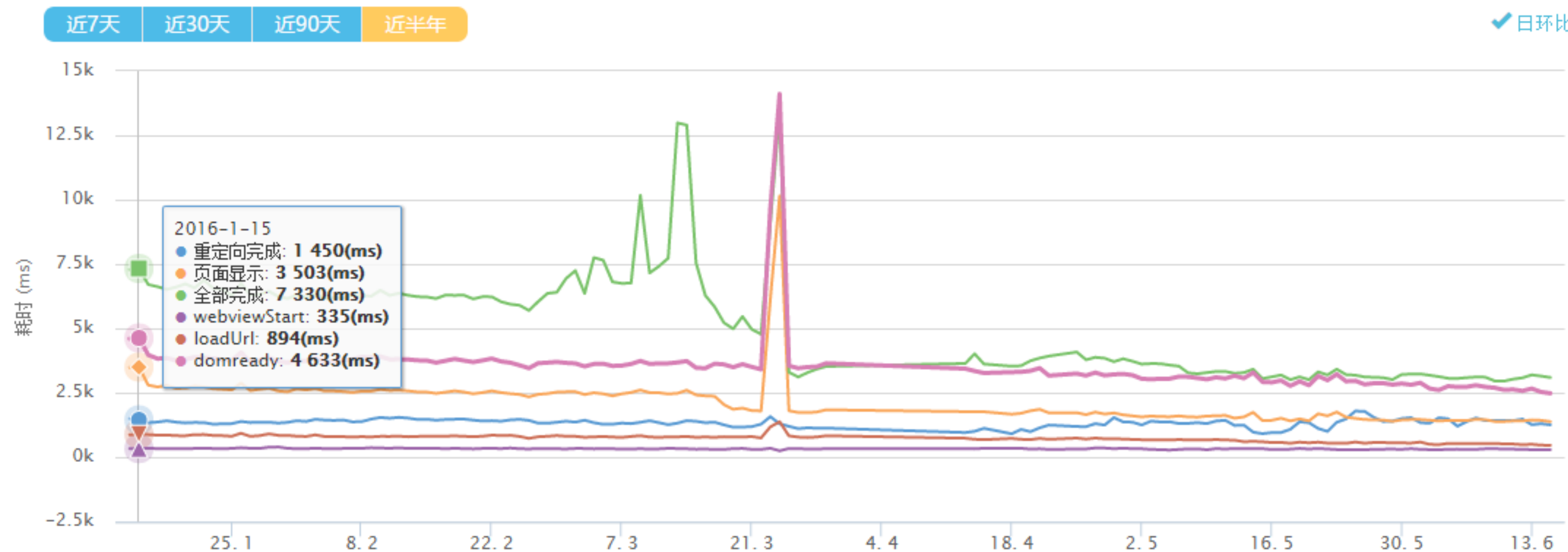


QQ看点图文页优化之前台性能篇

- H5常规优化

- 合并压缩CSS & JS
- 按需加载（首屏 + 响应式）
- 资源缓存
- 图片压缩
- GZIP开启
-

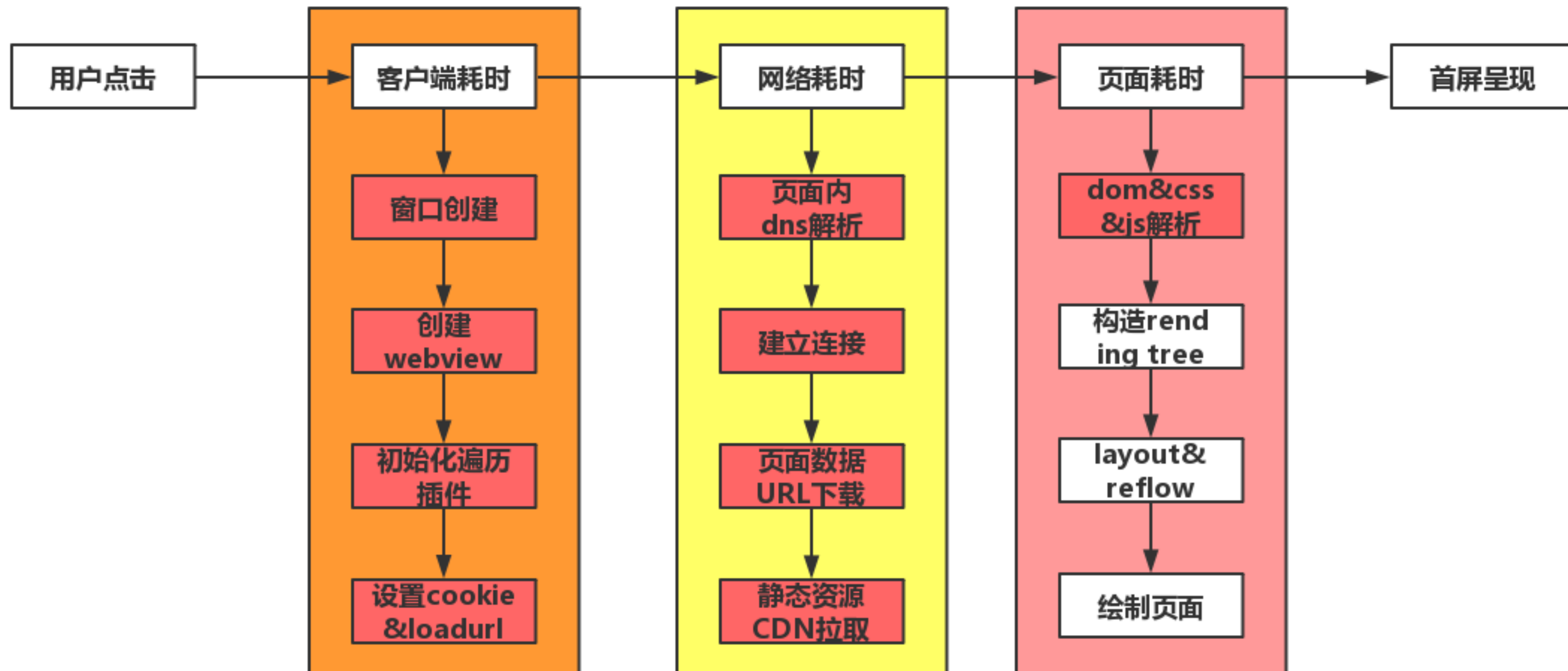
业务上报点性能





看点图文页优化之前台性能篇

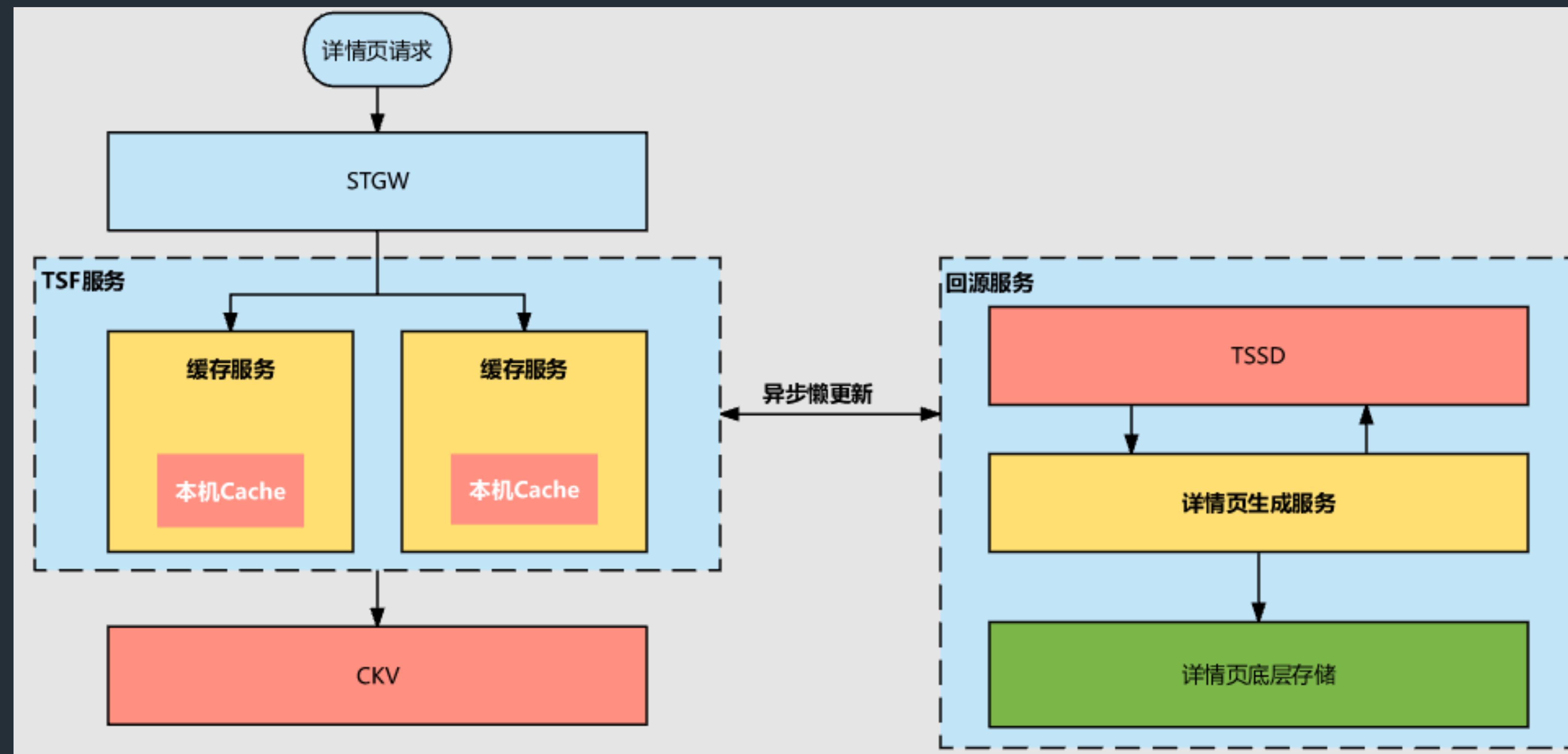
- 明确首屏直出目标：1s内（wifi/4g）
- 网络请求耗时全链路分析
- 网络请求路径优化
 - webview预启动
 - DNS预解析
 - 首屏sea.js内联
 - 首屏CSS内联
 - Wifi下页面预加载
 - 静态资源离线包缓存
 - 加载策略优化（首图预加载）





看点图文页优化之后台性能篇

- 多级缓存实现详情页高性能直出
 - 本机缓存
 - 链接缓存
 - TSSD存储
- 缓存异步懒更新





QQ看点图文页优化之前台高可用

web前台常见故障

好。而且作为iPhone 最大的销售国之一，iPhone 在国内的销量也同比下降不少。



为此很多人都将责任推给了库克，认为库克带领下的苹果，无能缺乏创新难以稳固自身地位，但是库克却并不这么认为，他表示：iPhone的用户体验是独一无二的，而且在各方面都有着非常独特的优势与口碑，虽然目前销量下滑，但这并不是问题，从长远来看iPhone 依旧有着很大的潜力。

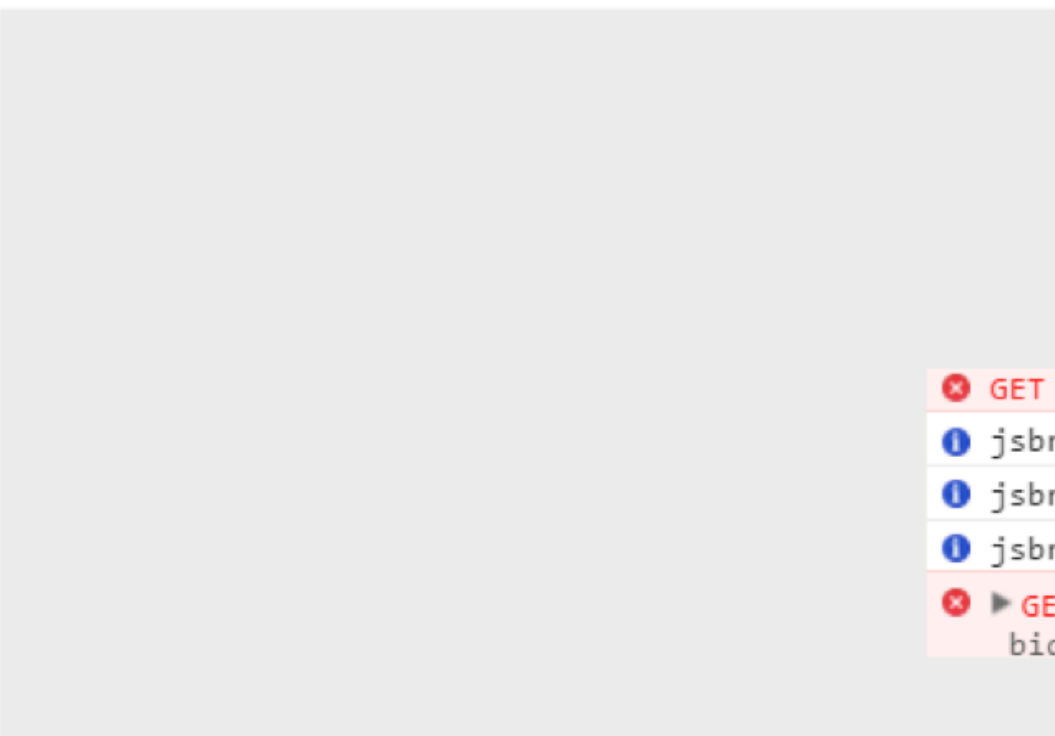
快报js出错无法运行

精彩推荐

正在加载...

在等待 mat1.gtimg...

2、TSW路由



微信js出错无法运行



好看你就多看点，点击阅读原文下载不得姐app，看更多有趣视频！

阅读原文

看点js出错页面正常

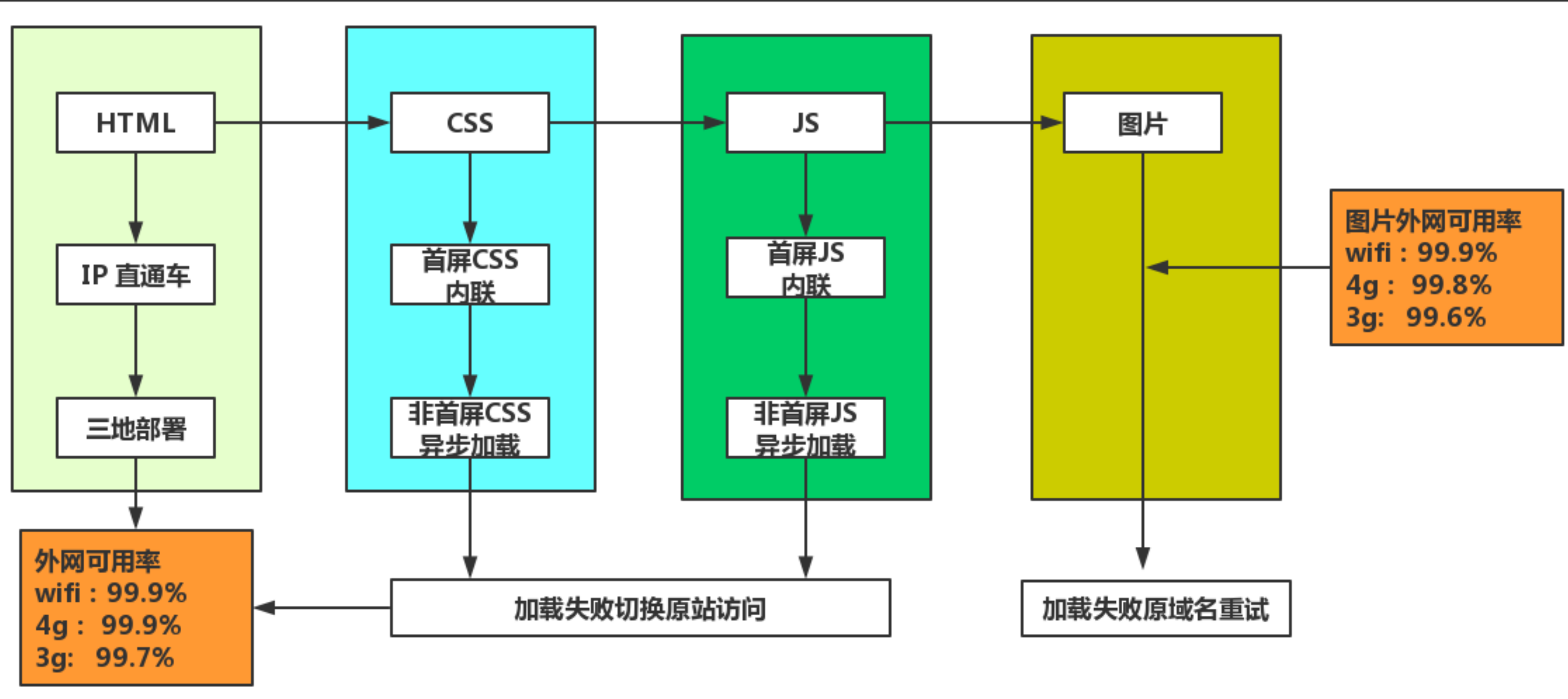
```
GET http://mp.gtimg.cn/sites/client/article/outer/aq_common.js?_bid=2258
jsbridge: version not match, apis ignored
jsbridge: version not match, apis ignored
jsbridge: version not match, apis ignored
GET http://mp.gtimg.cn/sites/client/article/page/atlas.js?_bid=2321&v=20161220
```

你可能还想看

至贱则无敌！很黄很暴力的死侍刷新你对超级英雄的...

12月18日 百思不得姐 [打开QQ查看](#)

全宇宙最不可思议的男女关系





- Web后台常见故障
 - 单机故障
 - 机房故障
 - 服务雪崩
 - 外部攻击
- Web后台台容灾方案
 - 故障容错
 - 多地部署
 - 过载保护
 - 频率控制

QQ看点图文页优化之后台高可用

