前隆科技
mobanker.com

微服务架构下的高可用网关

前隆科技 上海框架部 何雷

2018-03

CONTENTS

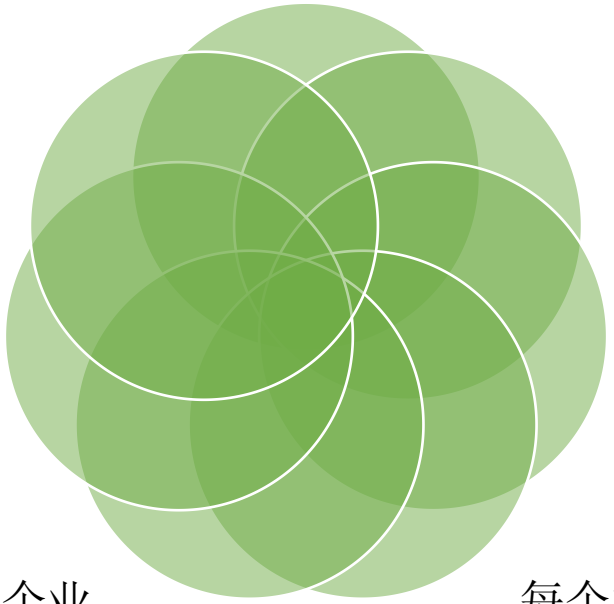
前隆网关的背景：面临的问题

前隆科技
mobanker.com

公司业务成倍增长，对接的商户越来越多，接口协议繁杂

平台级缺乏统一的外部流量统计和监控平台

产线增多，接口碎片化严重



不同的业务线有动态拦截请求的特定需求，需要做特殊处理

外部商户或请求和内部业务系统耦合紧密

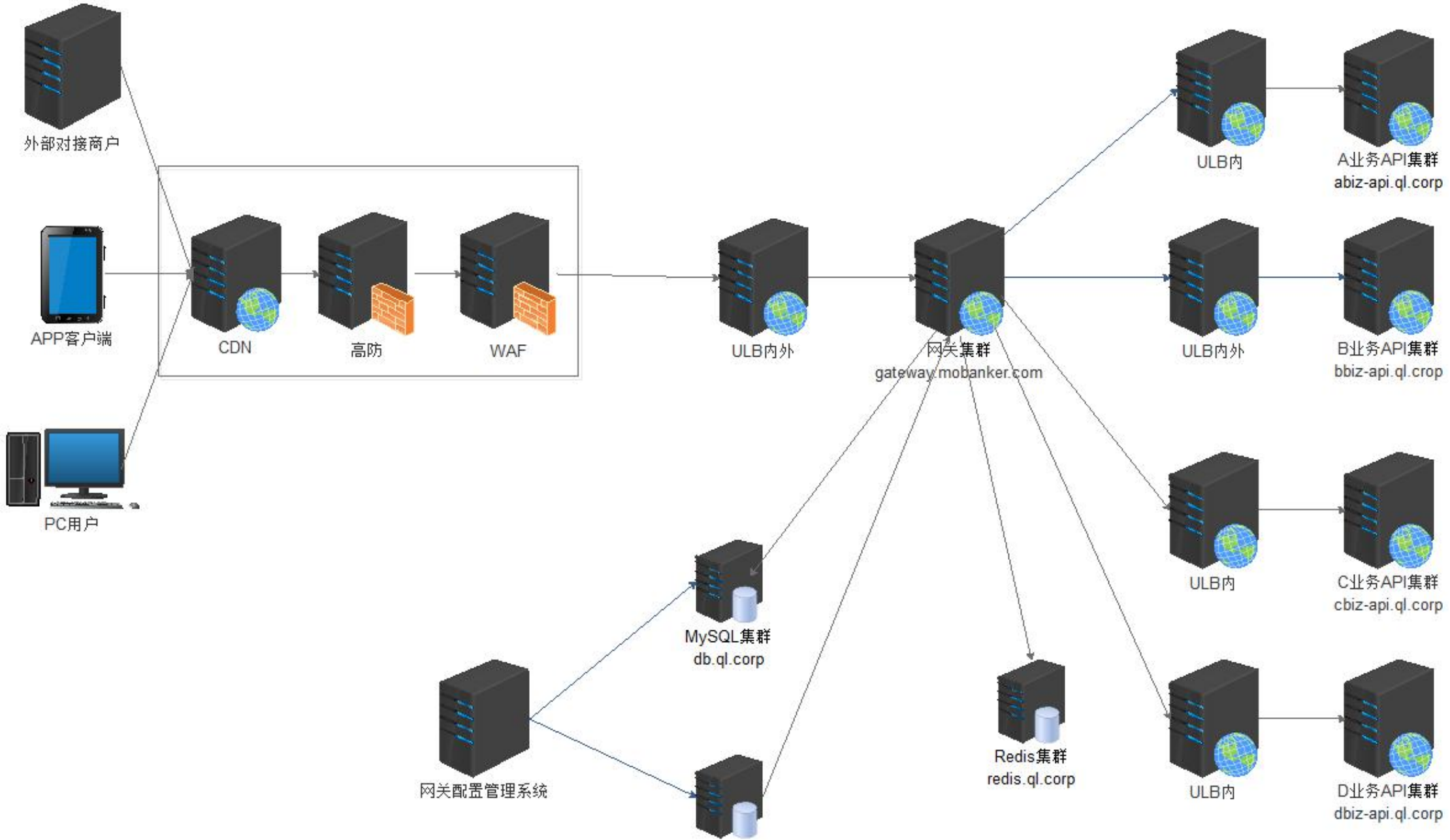
大流量下，每个业务系统急需一套限流、熔断、降级、线程隔离的框架

每个业务线都要实现安全、权限校验、反爬虫等一系列横切功能

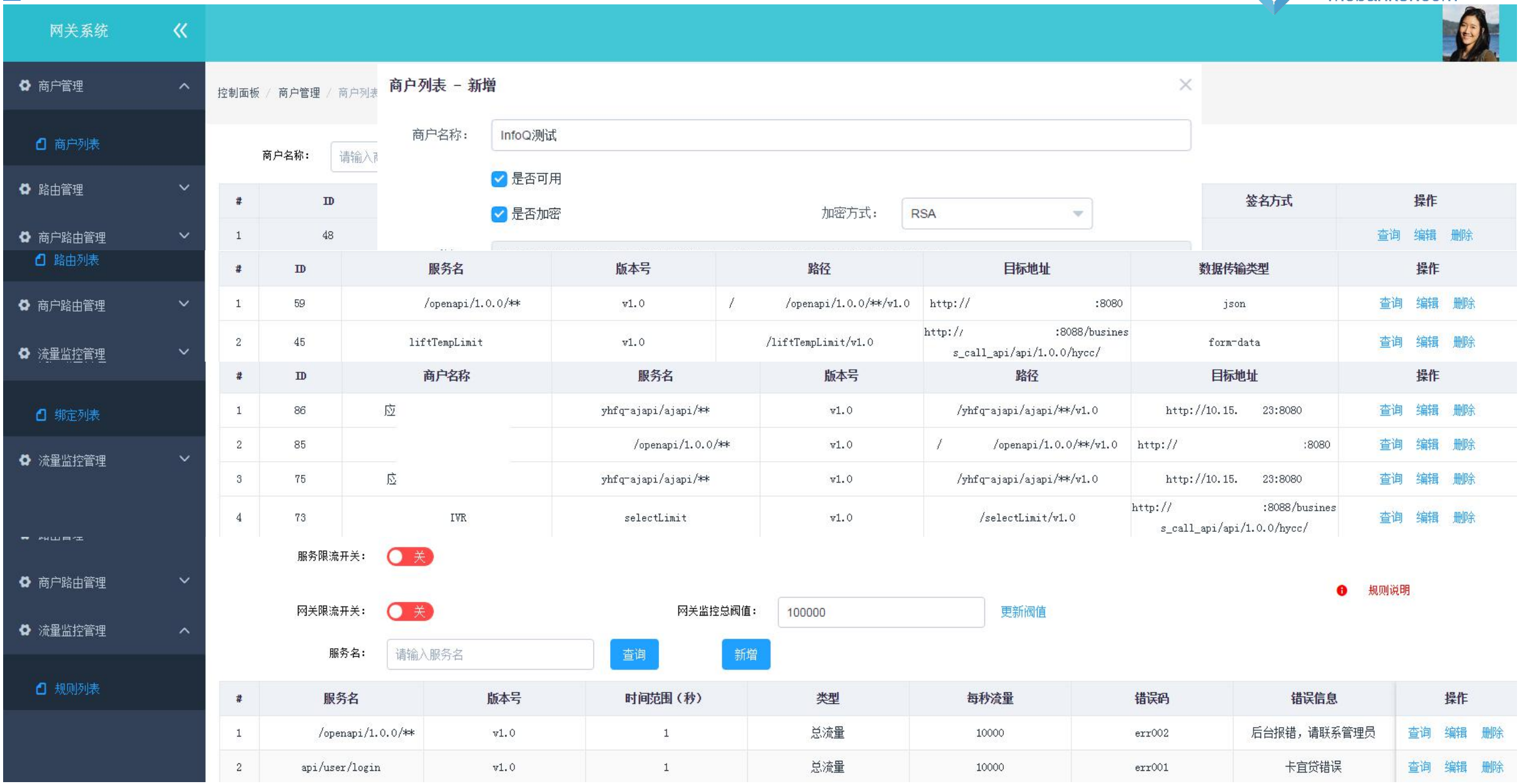
- 技术选型

	Zuul	Kong	Tyk
开发语言/基于平台	Java/Spring Cloud	Nginx	Go
优点	动态过滤 动态路由 验证与安全保障 监控与统计 限流熔断 线程隔离	授权 日志 ip限制 限流 api 统计分析 其他功能通过: lua编写插件实现	不符合当前技术体系
我们的选择	Zuul+自研		

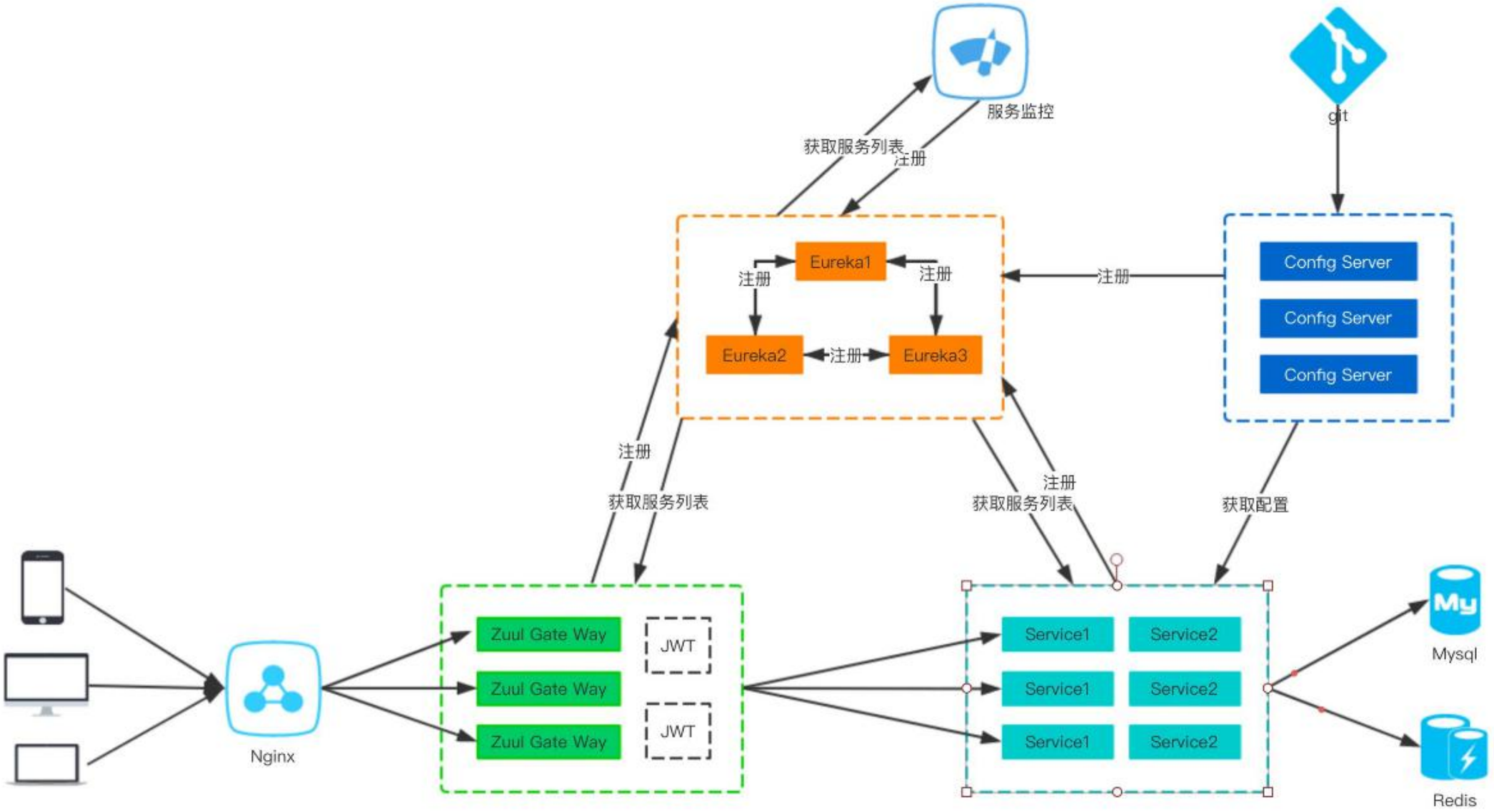
前隆网关的架构：物理部署图（部分业务）



前隆网关的架构：配置管理系统

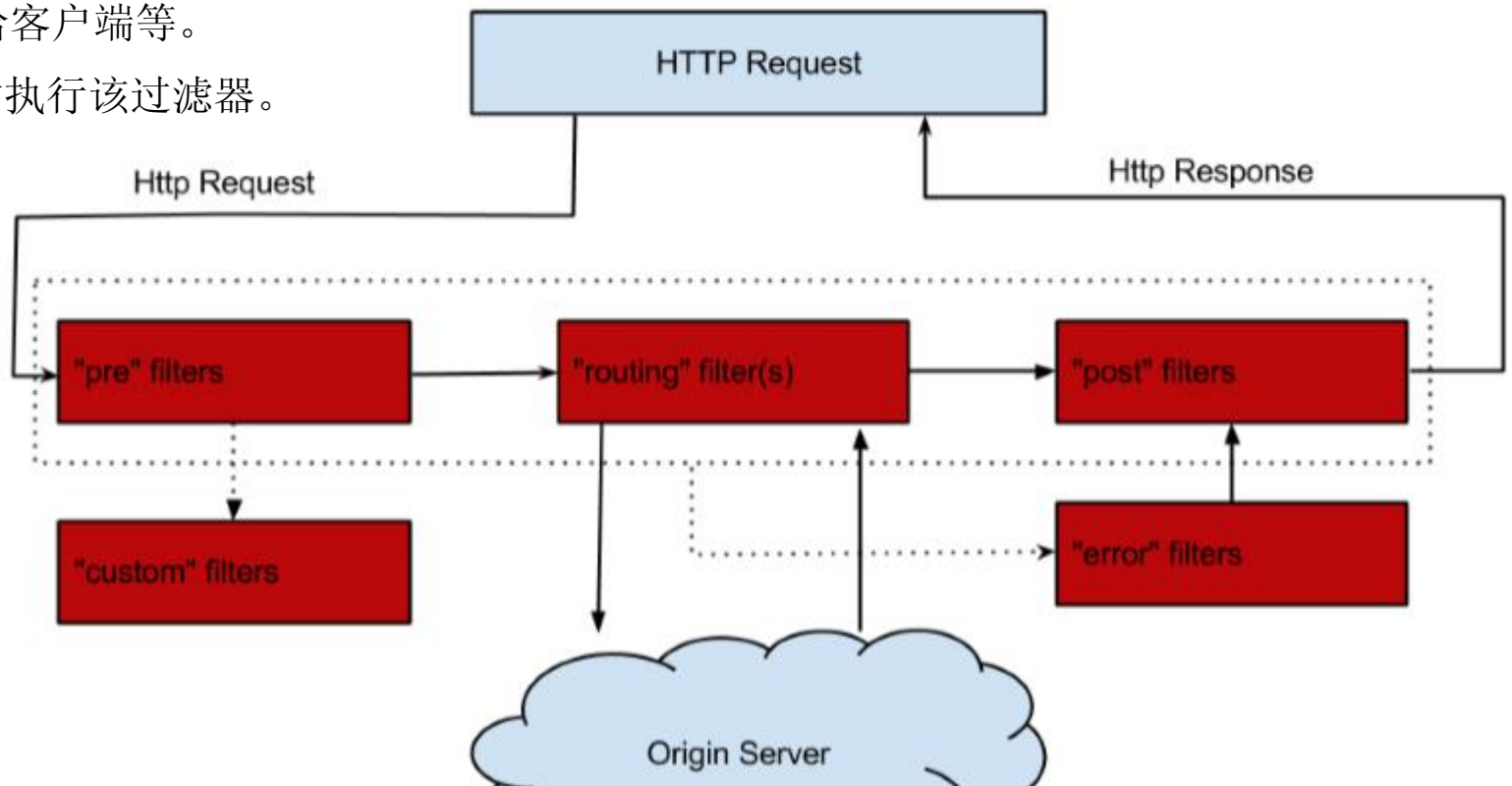


前隆网关的架构： Spring Cloud原生态的Zuul



前隆网关的架构： Zuul标准过滤器

- Zuul大部分功能都是通过过滤器来实现的。Zuul中定义了四种标准过滤器类型，这些过滤器类型对应于请求的典型生命周期。
- (1) PRE：这种过滤器在请求被路由之前调用。我们可利用这种过滤器实现身份验证、在集群中选择请求的微服务、记录调试信息等。
- (2) ROUTING：这种过滤器将请求路由到微服务。这种过滤器用于构建发送给微服务的请求，并使用Apache HttpClient或Netfilx Ribbon请求微服务。
- (3) POST：这种过滤器在路由到微服务以后执行。这种过滤器可用来为响应添加标准的HTTP Header、收集统计信息和指标、将响应从微服务发送给客户端等。
- (4) ERROR：在其他阶段发生错误时执行该过滤器。



网关与外部交互示例

前隆科技
mobanker.com

POST https://gateway.mobanker.com/openapi/ Params Send Save

Authorization Headers (1) Body Pre-request Script Tests Code

form-data x-www-form-urlencoded raw binary JSON (application/json)


```
{"merchantsNo":"893146","service":"                              SmsCode","sign":"b3bf714a1c            d2f3c1d6","signType":"md5","bizRequest":"lq67qBRDODvgjB1mGEdIIV+B29Hm8vycz4aDwYv
    ++iWd                  BAMEZF9JLz33WwSRNGw6ADJDw1I35AX615Ubndmr7rOvNn39SGXuXBQ8XVPzaoMu/POe1rGuhwVXiE                                                BNZTQER6cY3cC2zzG
    +HLZ8                  851+IlkQQI=","secret":"iSn2m+FZn+oagof4gkU0+3SD2f+9y6FHJ54DdfV7Ba2KzSRETds1w8t                                                pJEOmlAHAtA="
    ,"version":"v1.0","uuid":"c218fe92-cd43-4697-b17f-e         fa","timestamp":"2018-03-17 18:26:46"}
```


Body Cookies Headers (4) Test Results Status: 200 OK Time: 513 ms

Pretty Raw Preview JSON

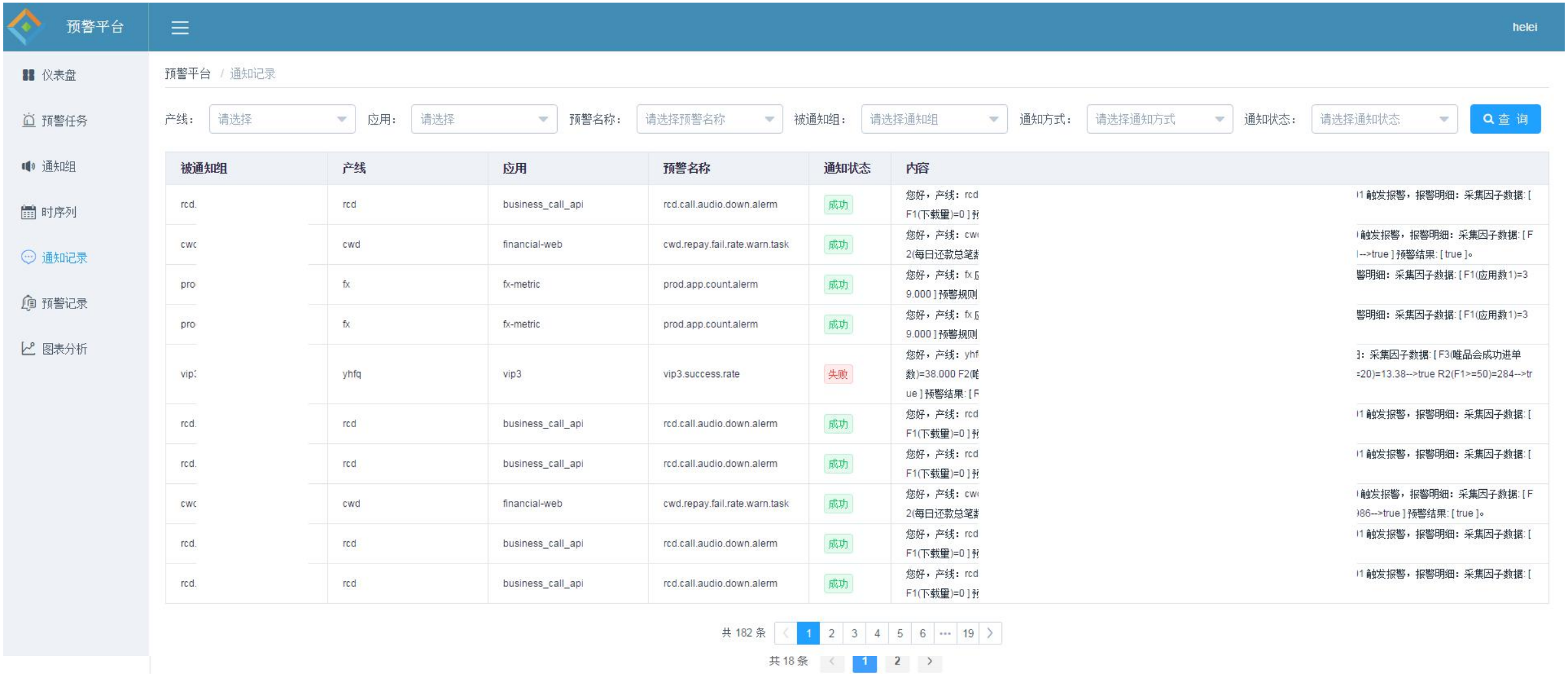

```
{
    "msg": "成功",
    "code": "00000000",
    "sign": "f02c45a9b35e            aa47955",
    "secret": "FJd3+z4Nnn            FztVdsPXUWoRiobYJZuY/DiPhBnaw                    tZEtYoGbSpW3fqRuj7e351TUVeBFWW9RauexZmrxDyKVrYkkq+e8yCKSBEQxKfxZh8+0u9ePhdkAQWVONIrUA=",
    "bizResponse": "vFq/+            5jzmpaVGz/LfLXIsXb11AyZZsSbir                    oAQQF6hpe3GQBNY"
}
```

前隆网关高可用措施：监控告警

前隆科技
mobanker.com

前隆网关高可用措施：度量平台与预警平台

前隆科技
mobanker.com



前隆网关高可用措施：多维度的认证授权、安全策略

前隆科技
mobanker.com

- 支持HTTP和HTTPS协议
- 只有授权的商户才能访问绑定过的API
- 针对商户对接：IP、商户的白名单、黑名单机制
- 公私钥算法保护API接口调用和数据传输安全
- 请求时，客户端动态生成DES3密钥，对业务数据加密，公钥对DES3密钥加密，服务端反之操作；返回时亦反之操作。
- 双向MD5加盐对业务数据验签防篡改
- H5请求基于动态生成的Token认证，有效期为24小时

前隆网关高可用措施：平台级和服务级限流、熔断、降级、线程隔离

前隆科技
mobanker.com

平台级限流：RateLimit，每台网关服务器控制所有业务线的流量请求。实现一个PreFilter，filterOrder为最先顺序执行，以拦截所有请求。请求拿到令牌才能后续执行，未拿到令牌直接返回网络繁忙提示，以保证现有的服务器在请求超过最大峰值时不被冲垮。

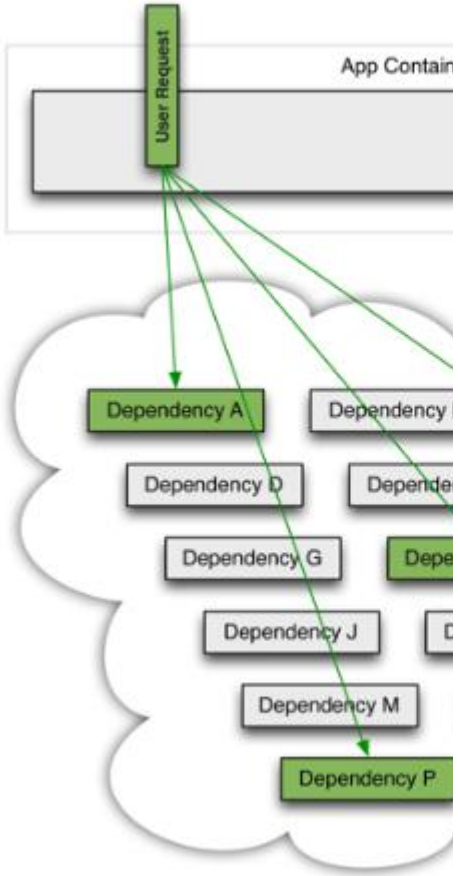
服务级限流：Hystrix，使用命令模式(继承HystrixCommand类)来实现具体的服务调用逻辑(run方法)，并在命令模式中添加了服务调用失败后的降级逻辑(getFallback)。

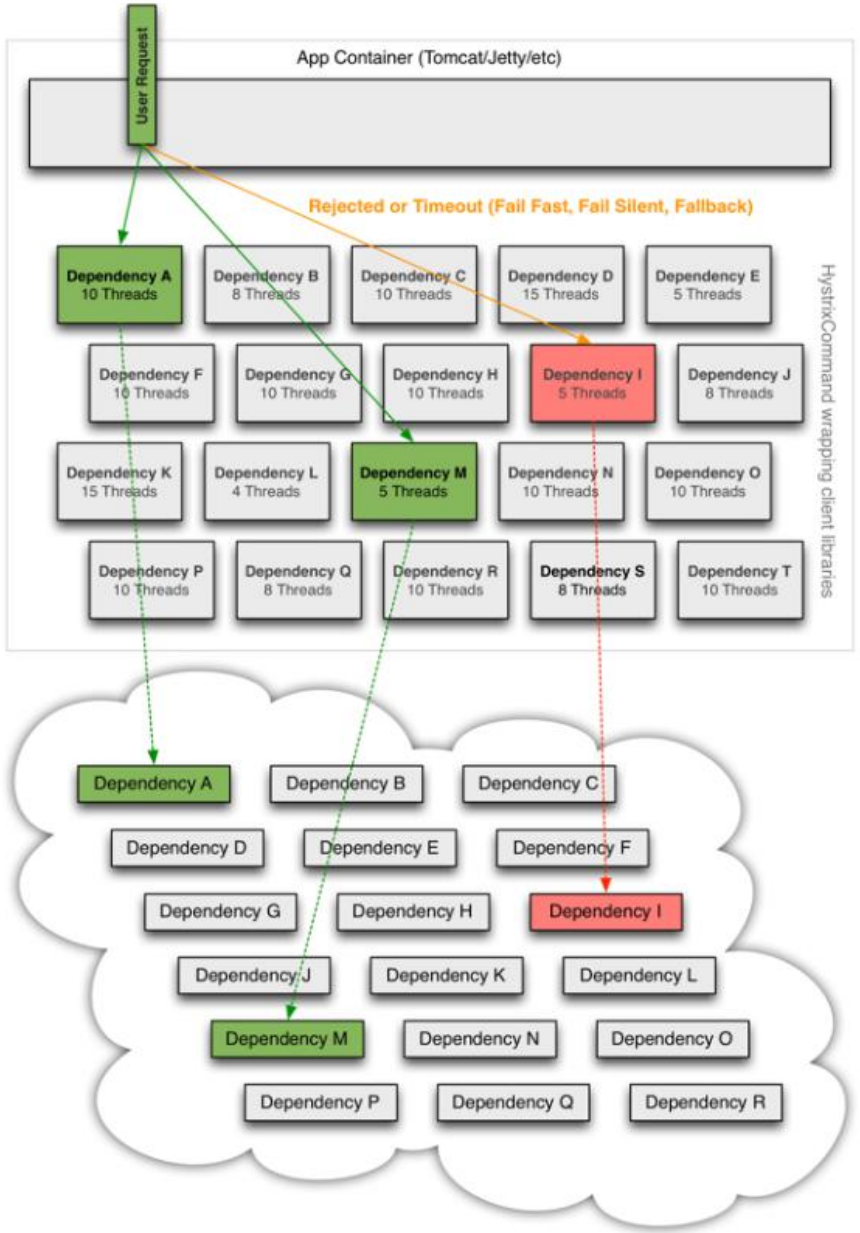
- 为什么需要Hystrix?
- 熔断器模式
- HystrixCircuitBreaker内部逻辑

为什么需要Hystrix?

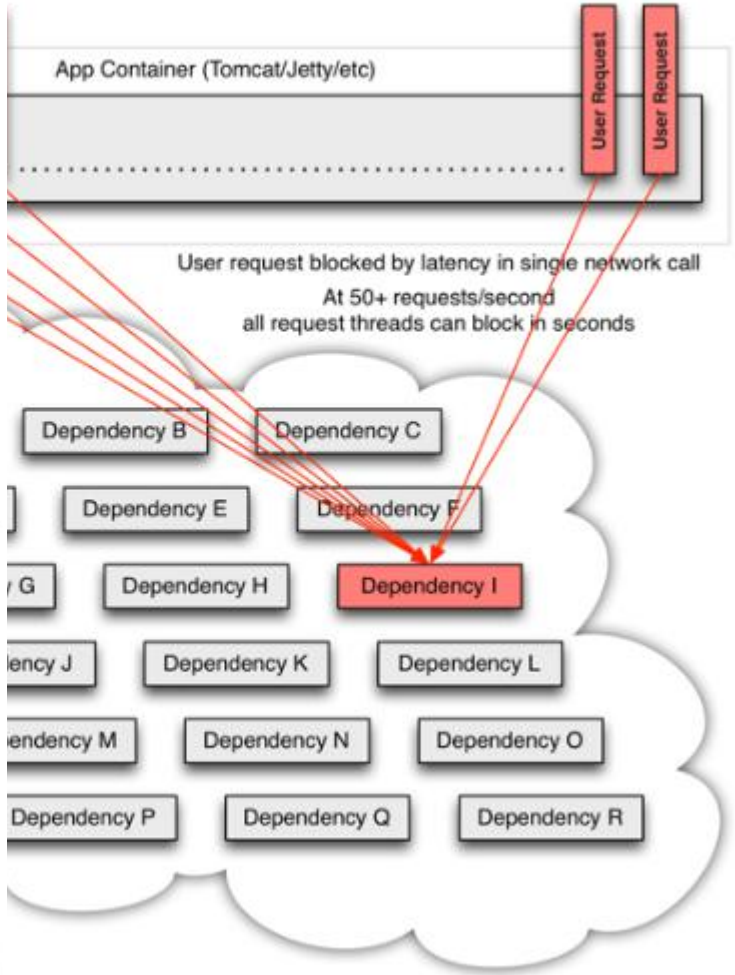
在大中型分布式系统中，通常系统很多依

在高并
忙，暂
如下图：



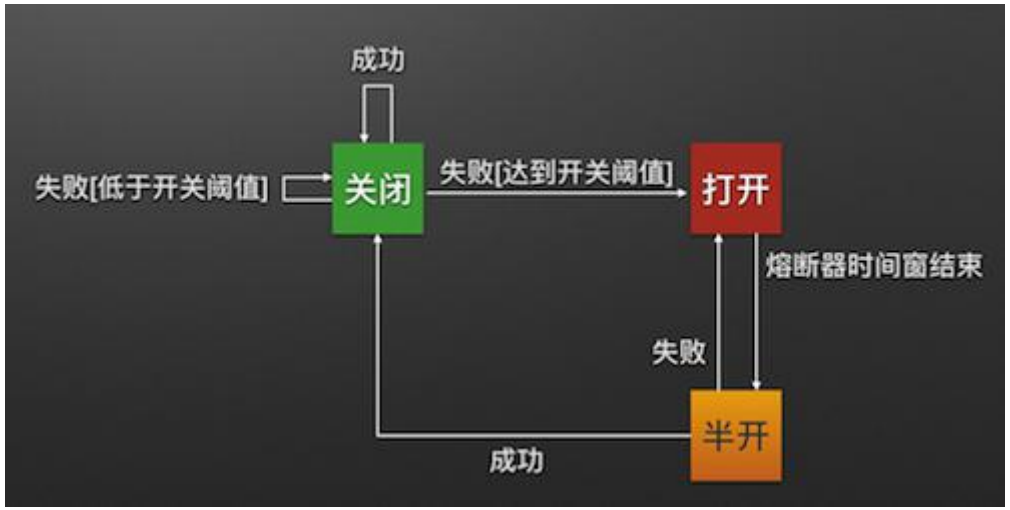


的线程池就出现阻塞(BLOCK),影响整个线上服务的稳定性.如下图:



熔断器模式

熔断器模式定义了熔断器开关相互转换的逻辑:

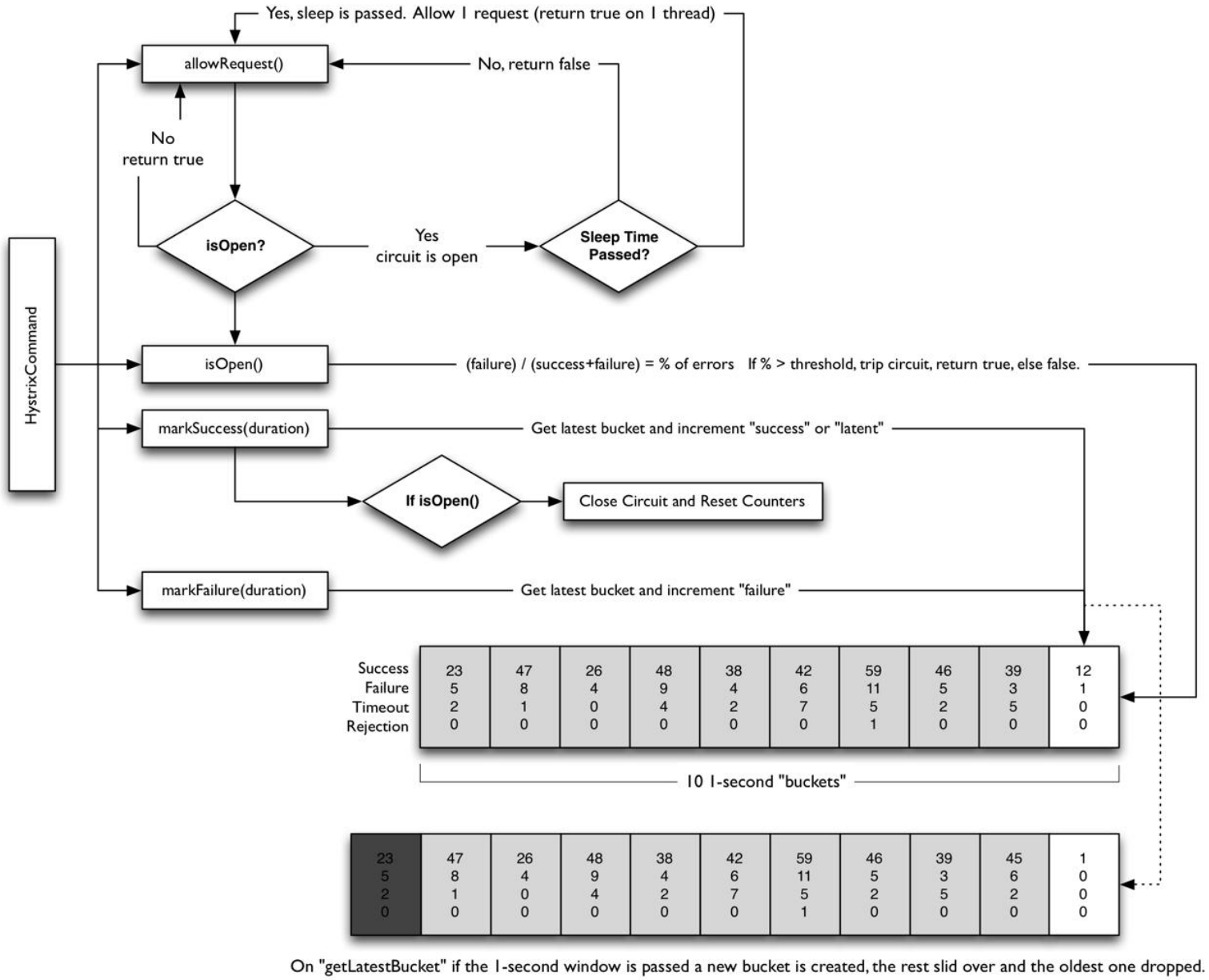


服务的健康状况 = 请求失败数 / 请求总数.
熔断器开关由关闭到打开的状态转换是通过当前服务健康状况和设定阈值比较决定的.

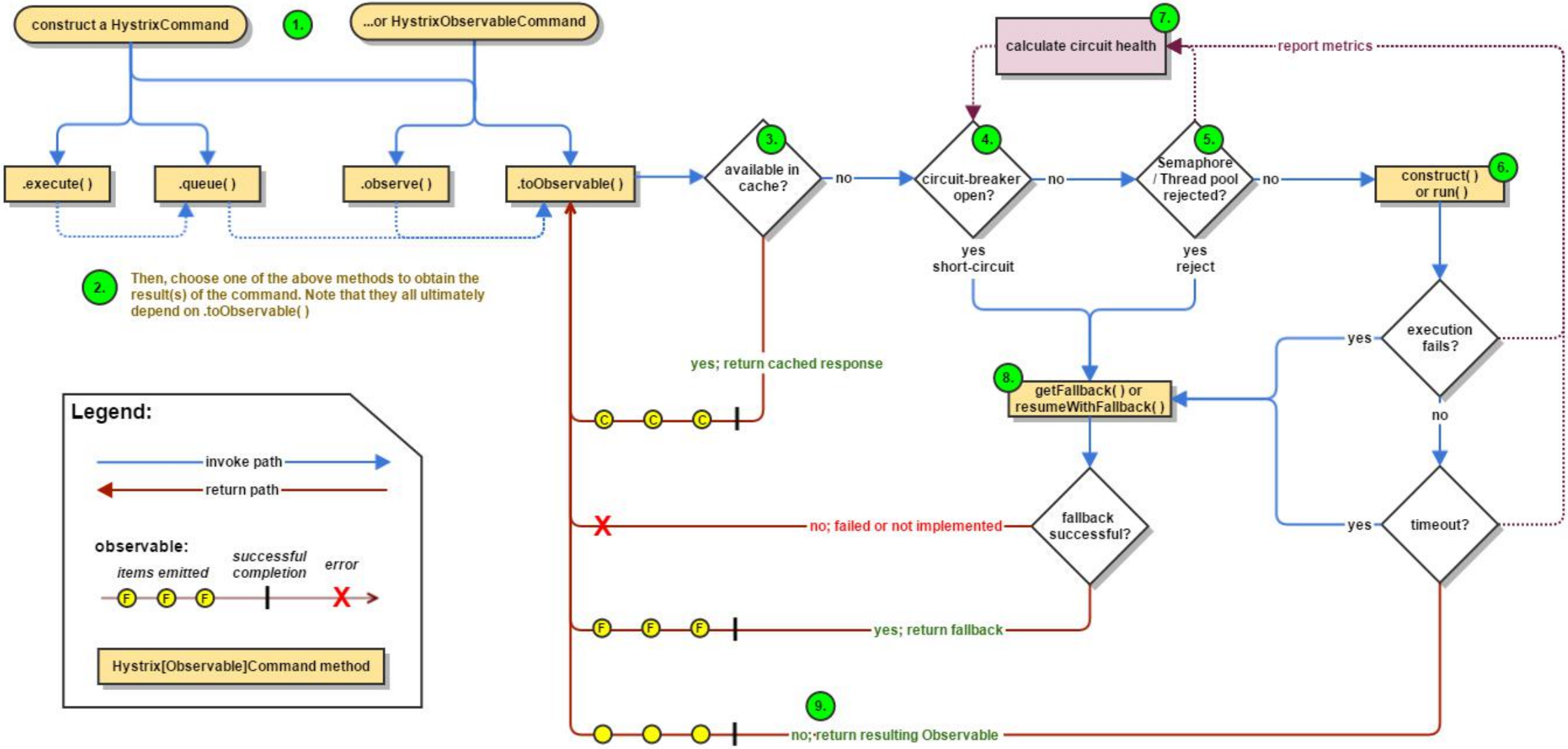
- ✓ 当熔断器开关关闭时, 请求被允许通过熔断器. 如果当前健康状况高于设定阈值, 开关继续保持关闭. 如果当前健康状况低于设定阈值, 开关则切换为打开状态.
- ✓ 当熔断器开关打开时, 请求被禁止通过.
- ✓ 当熔断器开关处于打开状态, 经过一段时间后, 熔断器会自动进入半开状态, 这时熔断器只允许一个请求通过. 当该请求调用成功时, 熔断器恢复到关闭状态. 若该请求失败, 熔断器继续保持打开状态, 接下来的请求被禁止通过.

熔断器的开关能保证服务调用者在调用异常服务时, 快速返回结果, 避免大量的同步等待. 并且熔断器能在一段时间后继续侦测请求执行结果, 提供恢复服务调用的可能.

HystrixCircuitBreaker内部逻辑



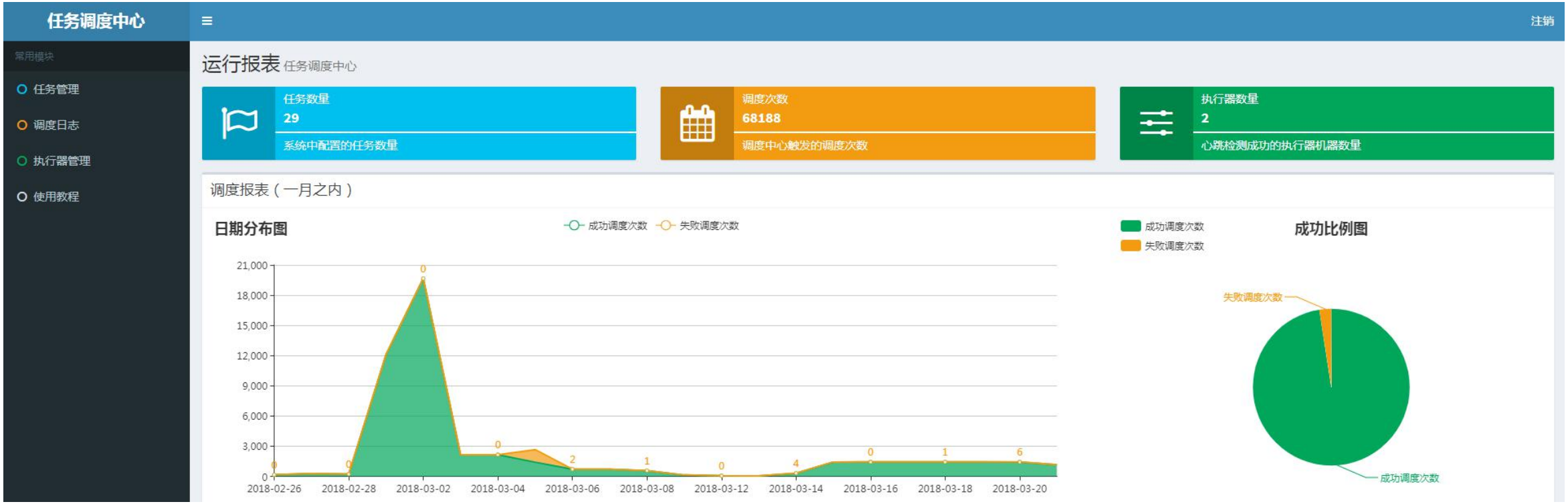
Hystrix工作流程



前隆网关高可用措施：实时生成公私钥影响性能的优化

前隆科技
mobanker.com

实时生成RSA密钥影响性能，通过任务调度中心在凌晨时预先批量生成公私钥放在缓冲中，请求来了直接取用。



前隆网关高可用措施：反爬机制措施

前隆科技
mobanker.com

检测Headers

通过校验Headers携带的User-Agent、Referer、Host、Cookie等信息识别爬虫

关键数据转成图片

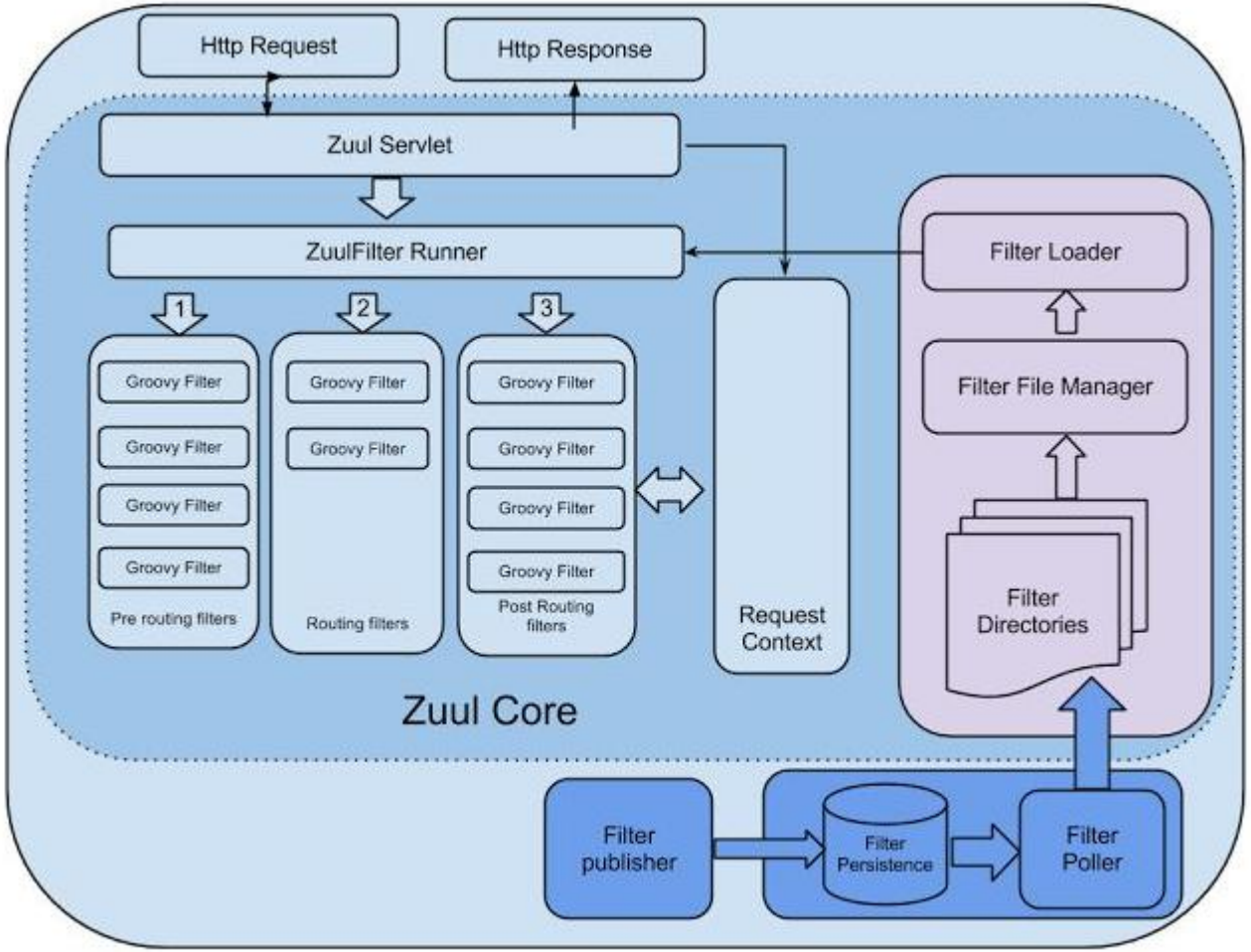
统计IP或账号

每次收到请求时记录下IP、访问时间、访问次数，通过判断访问时间、访问次数和阈值大小，识别是否爬虫。
同时，判定IP与上次请求IP是否在同一区域，不在则增加验证码、人工交互等手段拦截请求。

前端JS加密混淆

动态过滤器

Zuul的过滤器是由Groovy脚本写的，这些过滤器文件被放在Zuul Server上的特定目录下面，Zuul会定期轮询这些目录，修改过的过滤器会动态的加载到Zuul Server中以便过滤请求使用。




前隆科技
mobanker.com
谢谢聆听