**ms-api-gateway项目设计**

|  |  |  |  |
| --- | --- | --- | --- |
| 信息分类 | 研发设计 | 涉密等级 | 机密 |
| 责任部门 | 智能网联研究院 | 责任人 | 傅建记 |
| 起草人 | 黄云 | 创建时间 | 2020.05.14 |
| 授权范围 | 智能网联研究院内部 | 涉密截止日期 | 2025.02.10 |

**修订记录：**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **日期** | **作者** | **修订内容** |
| V1.0 | 2020-05-14 | 黄云 | 初稿 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

[2 文档介绍 3](#_Toc26663)

[2.1 文档目的 3](#_Toc9641)

[2.2 读者对象 3](#_Toc30994)

[2.3 术语与解释 3](#_Toc12222)

[3 背景概述 4](#_Toc15091)

[4 需求分析 4](#_Toc1498)

[5 方案设计 5](#_Toc12547)

[5.1 技术架构 5](#_Toc22425)

[5.2 调用链路 6](#_Toc18576)

[5.3 功能实现 6](#_Toc28850)

[5.3.1 限流过滤器 6](#_Toc19370)

[5.3.2 黑白名单过滤器 7](#_Toc13146)

[5.3.3 鉴权授权过滤器 7](#_Toc31343)

[5.3.4 协议转换过滤器 8](#_Toc13043)

[5.3.5 ‘可扩展’过滤器 8](#_Toc1933)

[5.4 方案总结 8](#_Toc22779)

[5.5 附录 8](#_Toc23023)

[5.5.1 Redis数据设计 8](#_Toc3725)

[5.5.2 过滤器顺序 9](#_Toc17583)

[5.5.3 包名类名设计 9](#_Toc30468)

# 文档介绍

## 文档目的

本文档是对ms-api-gateway项目做的框架设计，主要包含技术架构、设计思路、及应用场景。本阶段，文档主要用于指导项目计划制定、系统/模块的开发、数据设计、测试方案、运维方案等。

## 读者对象

研发相关人员\评审相关人员\高层领导及公司规定的其他有阅读权限人员。

## 术语与解释

|  |  |
| --- | --- |
| **缩写、术语** | **解 释** |
| ms-api-gateway | 网关项目名称 |
| ms-user-auth | 鉴权微服务 |
|  |  |
|  |  |
|  |  |

# 背景概述



主要用于应用层（手机终端、展示大屏、后端管理网站、车主网站等）对微服务层（cpsp服务、用户中心、手机服务、日志服务、鉴权服务等）之间路由转发，同时包括对调用方的信息重新打包、鉴权、熔断、限流、日志记录、监控、缓存等操作。

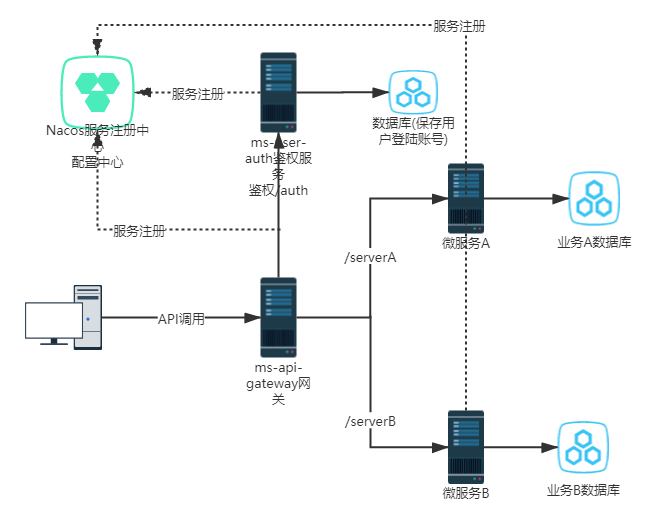
# 需求分析

ms-api-gateway作为各个微服务流量的入口，对应用端功能的调用是否能顺利完成起到至关重要的地位。在ms-api-gateway网关的设计方面，主要以对请求鉴权、黑白名单、统一监控、流量控制、请求路由、协议转换、缓冲才是需要重点考虑的方面。

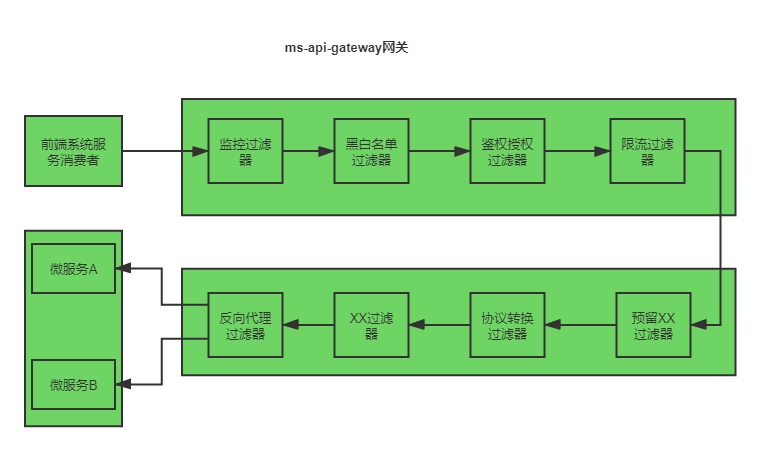
* 请求鉴权：对所以应用端请求都应做相应的鉴权（除相关忽略鉴权path外---可配置）
* 黑白名单：对相关ip/设备MAC/用户通过添加黑名单方式进行限制接口调用访问，防恶意攻击
* 统一监控：由于网关是外部服务的入口，所以我们可以在这里监控我们想要的数据，比如入参出参，链路时间
* 流量控制：熔断降级。对于流量控制，熔断降级非业务逻辑可以统一放到网关层
* 请求路由：在我们的系统中由于同一个接口新老两套系统都在使用，我们需要根据请求上下文将请求路由到对应的接口
* 协议转换：RPC 协议转成HTTP，由于在内部开发中我们都是以 RPC 协议(thrift or dubbo)去做开发，暴露给内部服务，当外部服务需要使用这个接口的时候往往需要将 RPC 协议转换成 HTTP 协议
* 缓存：缓存某些热点数据，当请求到网关可动态添加到请求头或请求实体

# 方案设计

## 技术架构



## 调用链路



注：实际调用顺序可能根据需要做相应更改

## 功能实现

### 鉴权授权过滤器

**作用：**对请求进行相应鉴权，鉴权成功后才能调用相应需要鉴权的uri

**实现**：所有非忽略鉴权的uri可配置（在yaml中配置）,通过调用ms-user-auth鉴权服务端授权接口经行授权，目前使用JSON Web Token（JWT）进行授权:

* 授权：ms-api-gateway网关端向ms-user-auth鉴权服务端转发授权请求（包括用户名和密码等必备参数，授权url应在忽略鉴权uri列表中或ms-user-auth相关接口直接忽略鉴权）
* 返回：ms-api-gateway网关端直接返回相关授权结果
* 业务请求：前端服务通过携带授权返回的access\_token进行业务请求，对于ms-api-gateway网关，在过滤器中对所有非忽略鉴权的uri（在yaml中配置）进行鉴权（调用ms-user-auth鉴权服务端鉴权接口）
* access\_token失效：对于access\_token失效，ms-user-auth鉴权服务端统一返回特定的resultCode，前端根据标识主动刷新access\_token后才能调用相关业务请求

**不足**：网关与鉴权服务调用频繁、是否影响性能及吞吐量

### 黑白名单过滤器

**作用**：防止恶意攻击，进行IP、用户、终端设备访问限制

**实现**：使用黑名单方式，通过配置文件可以配置进行拦截方式(可以组合)，根据拦截方式从redis读取相应黑名单列表（如果存在黑名单列表中，返回特定错误编码），key值设定参见[4.5.1-redis数据设计](#_Redis数据设计)章节，后期可考虑从可视化界面上进行设置相关拦截：

1. IP拦截：根据远程ip进行拦截过滤
2. 用户：根据请求用户id进行拦截过滤
3. 终端设备MAC：终端MAC拦截过滤（有缺陷，如需根据终端MAC限制,实现方式有两种：1、前端服务调用业务请求需携带设备MAC值；2、网关主动去拿取，但需要使用nbtstat -a ip 去获取客户端信息，涉及网关需开通访问外网权限，涉及安全问题 暂不考虑，推荐使用第一种方式）
4. 资源path：根据请求uri地址进行拦截过滤（较少用）

**不足**：依赖于redis中间件服务

### 日志记录过滤器

**作用：**主要记录相关入参出参及链路时间，为分析数据平台提供数据依据及黑白名单提供数据依据

**实现：**通过自定义过滤器实现，根据重写request订阅者订阅参数和重写response订阅者异步记录日志，并计算相应链路调用时间

### 限流过滤器

**作用：**在高并发的场景下，为了服务的可用性，需进行限流

**实现**：使用Redis和lua脚本实现了令牌桶方式进行限流，通过对配置文件对各微服务routeId进行设置，包括设置令牌桶总容量、令牌桶每秒填充平均速率，可根据远程请求ip、请求uri、用户进行限流

**不足**：依赖于redis中间件服务

### 请求路由

**作用：**做相应路由转发

**实现：**使用spring cloud gateway的routes方式实现

### 协议转换过滤器

**作用：**系统内部是通过RPC或者是dubbo方式调用，网关其实就是相当于将外部http方式调用转为内部RPC方式调用，或者将内部RPC转为http

**实现**：feign本身是个伪RPC，实则http，如果是使用feign的内部调用模式，无需添加额外代码，如需添加其它协议转换，需自定义相关过滤器

### 缓存

**作用**：做热点数据缓存

**实现**：比如通过授权成功后根据接口返回参数做用户基础信息的一个缓存，使用redis做数据缓存，当其它业务接口调用时根据需要从缓存读取数据并做相应参数添加调用

## 方案总结

本概要设计比较简单，只是基于目前需求情况设计，后续需根据需求迭代进行优化及完善。

## 附录

### Redis数据设计

存储黑白名单数据：可考虑set数据类型

**Key设计：**{微服务名}:{表/功能模块}:{主键}

|  |  |  |
| --- | --- | --- |
| 黑名单列表 | key | value |
| IP黑名单 | msgw:bl:ip | {ip值} |
| 设备MAC黑名单 | msgw:bl:mac | {mac值} |
| 用户id黑名单 | msgw:bl:userid | {userid值} |

如需考虑过期时间方案另定

### 过滤器顺序

order值越小越先执行,其中，自定义为需要自己实现的过滤器，默认为系统自带

|  |  |  |
| --- | --- | --- |
| 过滤器名称 | 作用 | order |
| LogRecorderGlobalFilter | 日志记录过滤器(自定义) | HIGHEST\_PRECEDENCE+100 |
| BlackOrWhiteListFilter | 全局-黑白名单过滤器(自定义) | HIGHEST\_PRECEDENCE+300 |
| AuthenticationFilter | 全局-鉴权授权过滤器(自定义) | HIGHEST\_PRECEDENCE+400 |
| RequestRateLimiter | 路由-限流过滤器（默认） | HIGHEST\_PRECEDENCE+500 |
| LoadBalancerClientFilter | 全局-负载均衡过滤器（默认） | 10100 |
| ProtocalTransferFilter | 全局-协议转换过滤器(自定义) | 10101 |
|  |  |  |

### 包名类名设计

参考：

[Java编码规范](http://10.24.3.242/wiki/pages/viewpage.action?pageId=6815768)：

<http://10.24.3.242/wiki/pages/viewpage.action?pageId=6815768>

[项目通用约定](http://10.24.3.242/wiki/pages/viewpage.action?pageId=10747997)：

<http://10.24.3.242/wiki/pages/viewpage.action?pageId=10747997>