



Zuul架构和实践

讲师：杨波

研发总监/资深架构师

第

1

部分

课程概述

课程大纲

- 01 课程概述
- 02 Zuul网关和基本应用场景
- 03 Zuul网关高级应用场景
- 04 Zuul网关架构剖析
- 05 Zuul网关代码剖析(Code Review)
- 06 Zuul网关过滤器管理工具(Code Review)
- 07 过滤器实验~前置过滤器(Lab01)
- 08 过滤器实验~路由过滤器(Lab02)
- 09 过滤器实验~后置过滤器(Lab03)

课程大纲

- 10 Zuul网关对接Apollo(Lab04)
- 11 Zuul网关生产部署实践
- 12 Zuul网关路由管理实践
- 13 基于网关的两层路由体系
- 14 Spring Cloud Zuul实验(Lab05)
- 15 Zuul2.0简介
- 16 Zuul网关最佳实践
- 17 参考资料和后续课程预览

课程概述和亮点

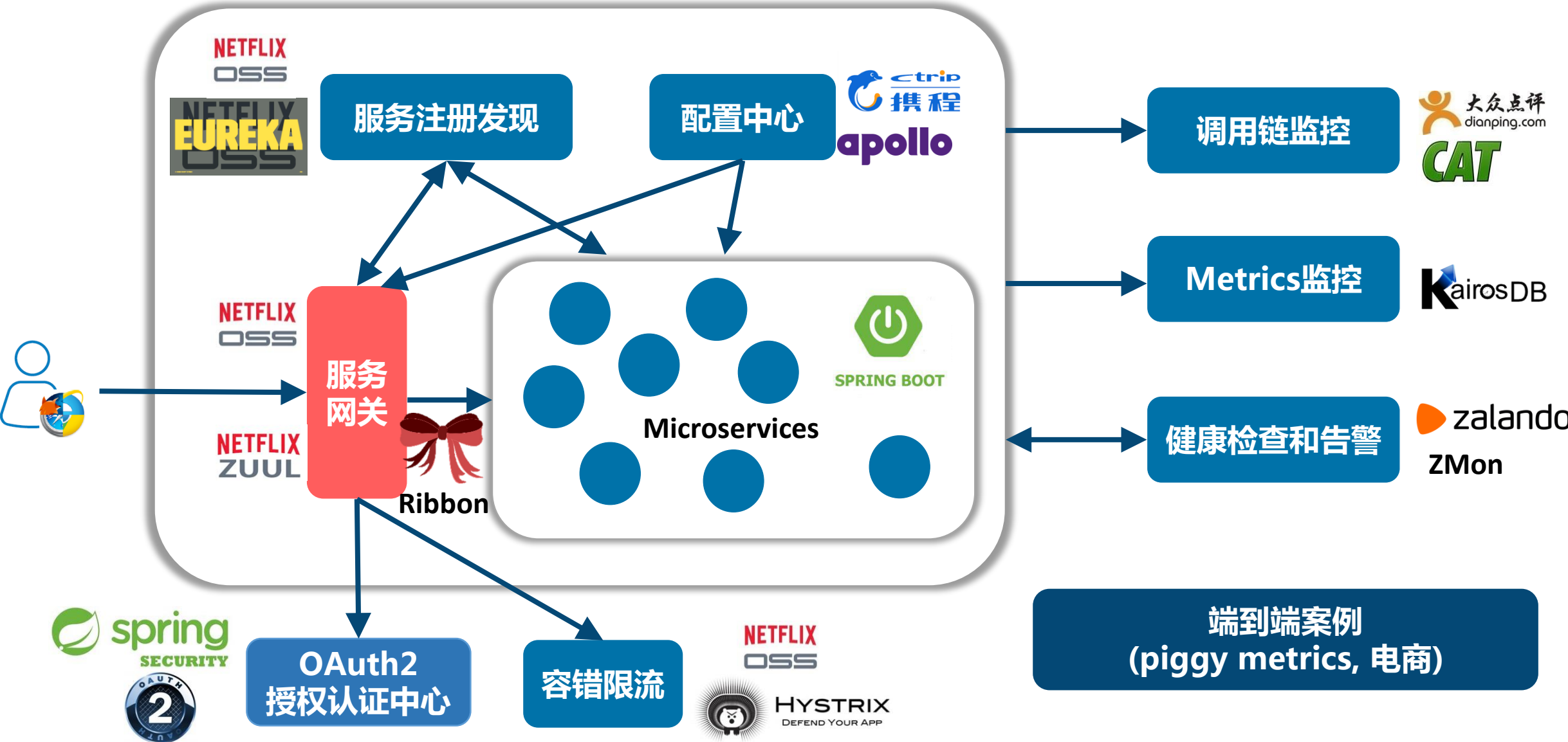


- 1 杨波的微服务基础架构体系的**第三个**模块
- 2 Netflix开源网关**Zuul****深度剖析**
- 3 面向**业务场景、架构和微服务**
- 4 **Zuul****核心源码分析**
- 5 **Zuul****定制扩展和集成**
- 6 结合**案例和实操**
- 7 结合**生产最佳实践**
- 8 **Spring Cloud Zuul****简介**

杨波的微服务基础架构体系2018预览(draft)



架构和技术栈预览



第

2

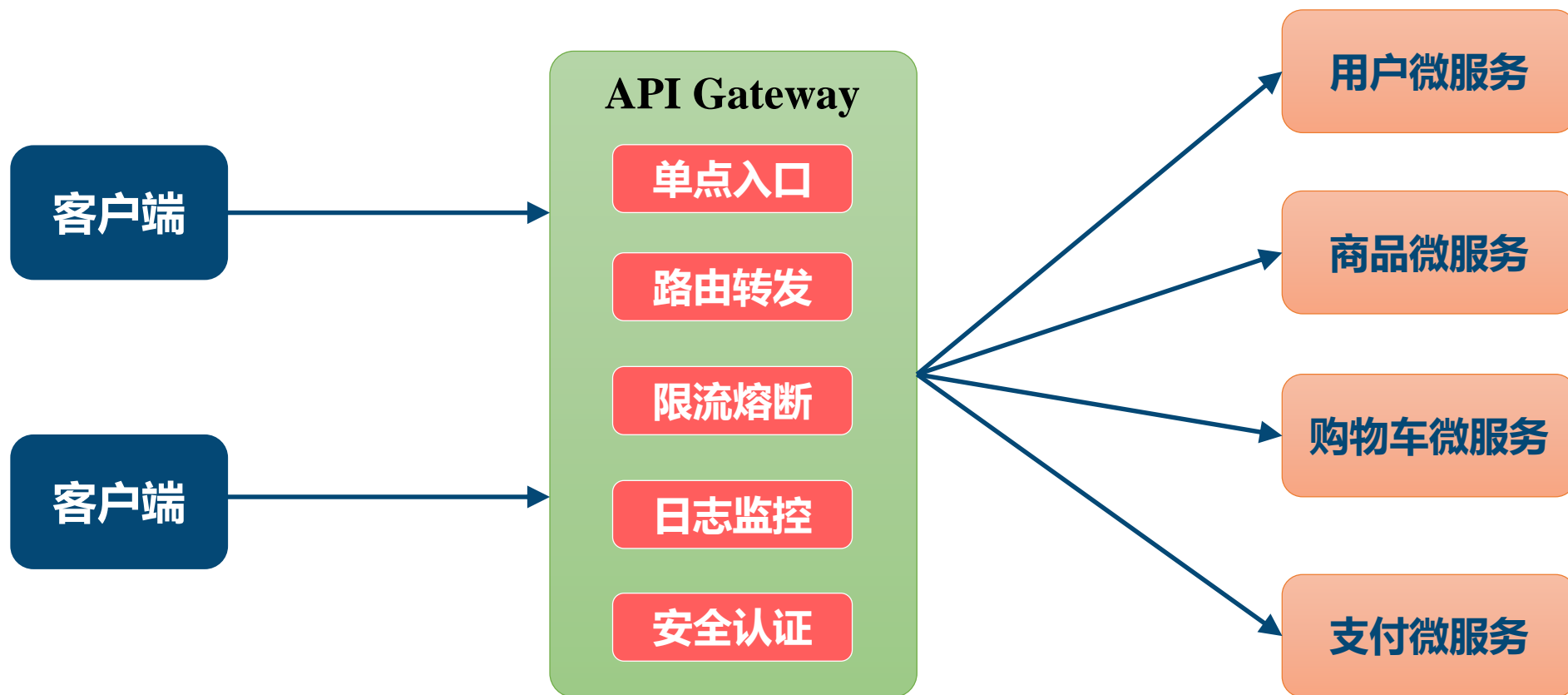
部分

Zuul网关和基本应用场景

API网关是如何演化出来的？



API网关基本功能



Netflix Zuul网关

Netflix于2012年初开源

- <https://github.com/Netflix/zuul>
- Zuul is an edge service that provides dynamic routing, monitoring, resiliency, security, and more
- 目前Github 5.3k星



亮点：可动态发布的过滤器机制

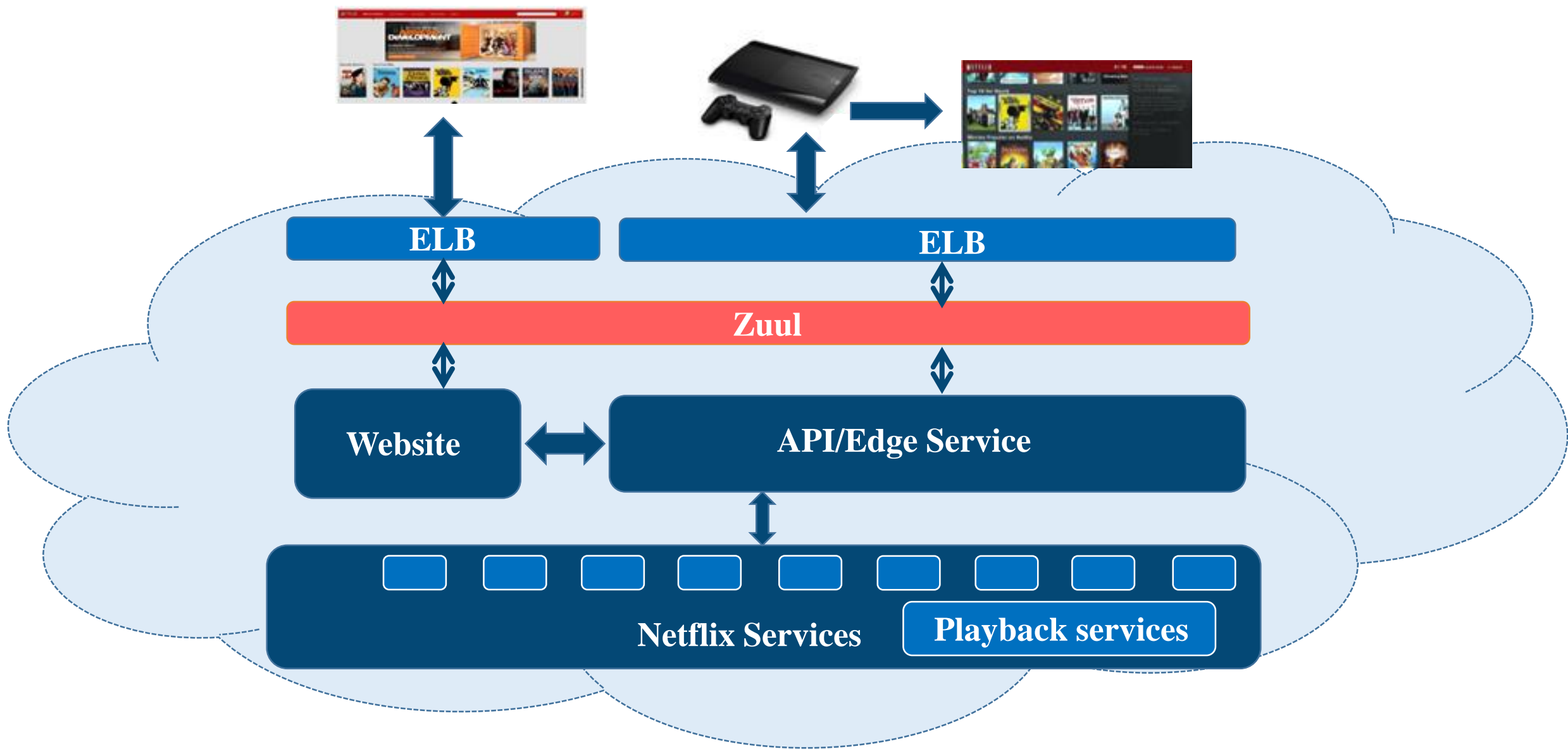
- Zuul在英文中是怪兽的意思，寓意看门神兽
- 2014年被Pivotal集成入Spring Cloud体系



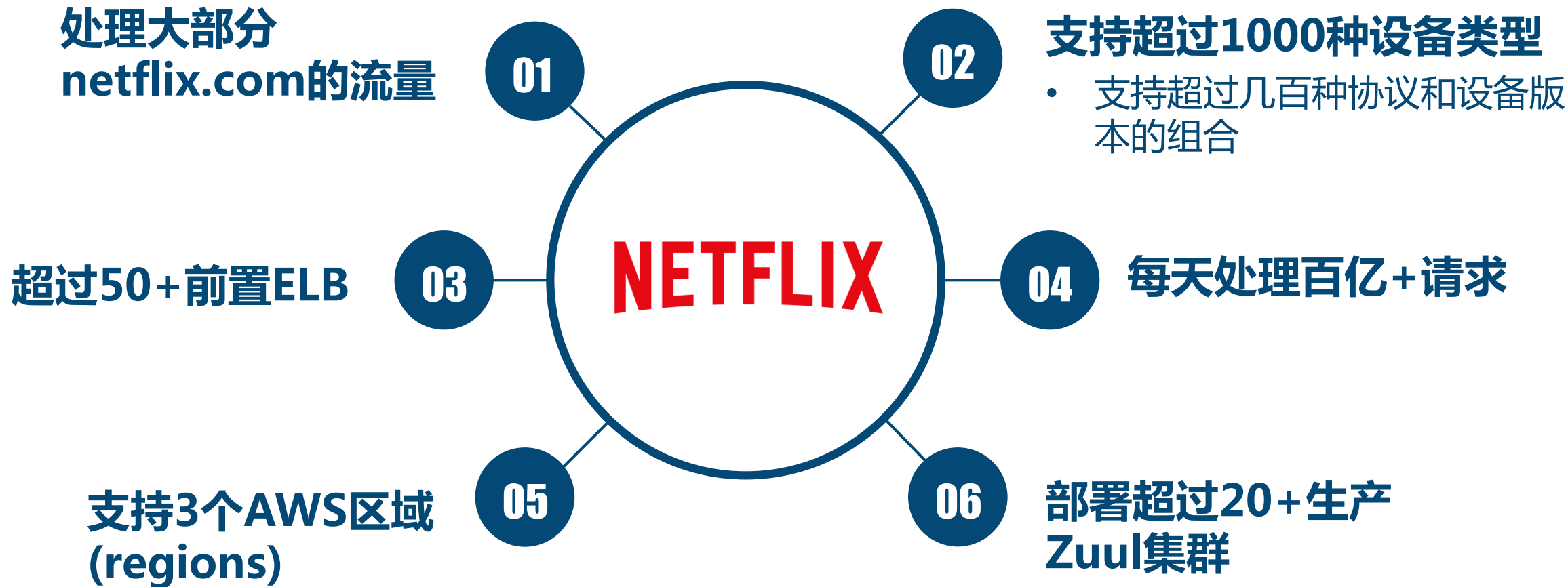
Netflix前总架构师Adrian Cockcroft评价

“One of our most powerful mechanisms and somewhat overlooked NetflixOSS projects is the Zuul gateway service.”

Netflix架构体系和Zuul



Netflix使用情况2017



国内公司落地案例



部署生产实例150+（分集群）

覆盖无线、H5、分销联盟、支付业务等场景

日流量**超50亿**



部署实例40+（分集群）

覆盖无线、H5、第三方开放平台、联盟商等场景

日流量超5亿

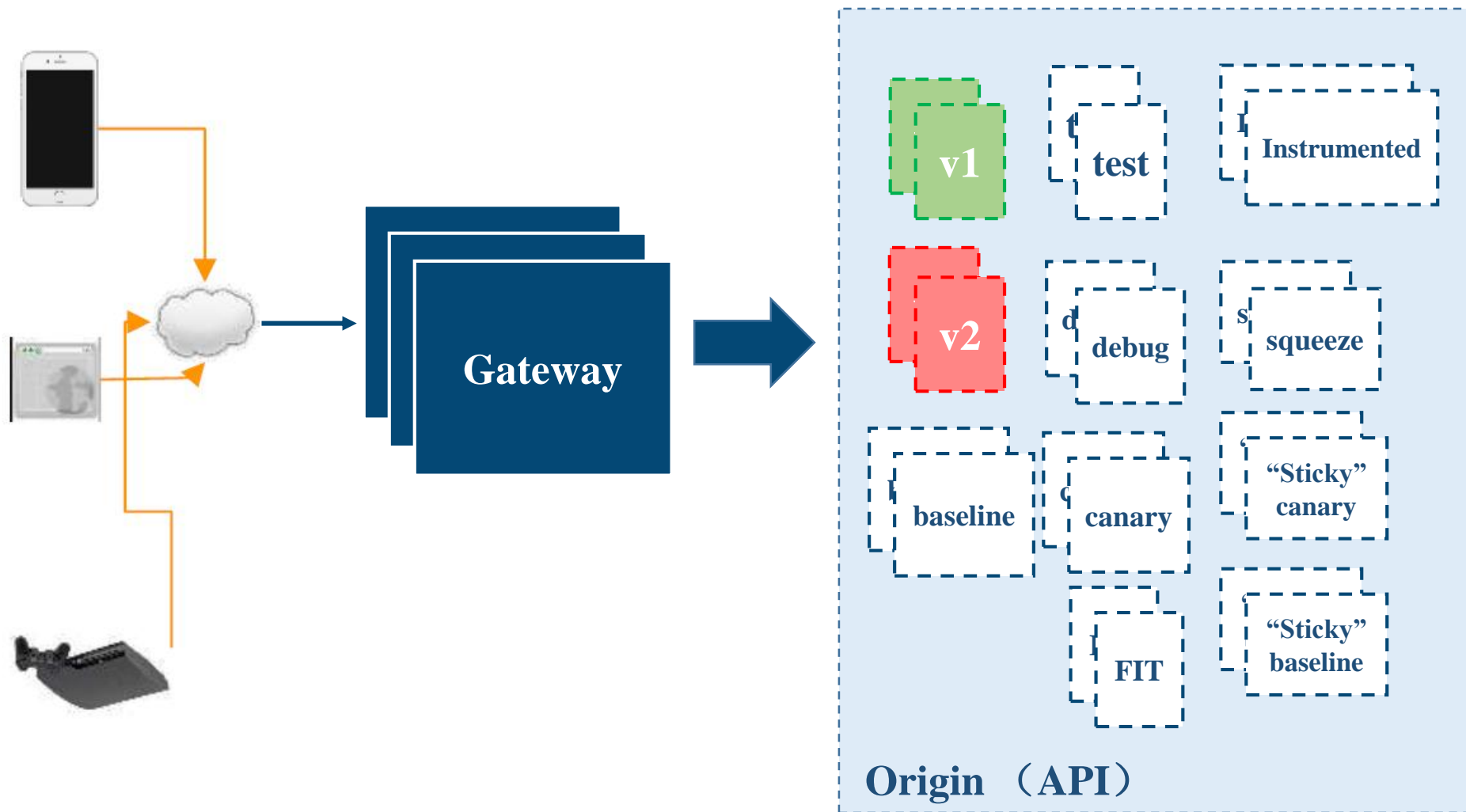
第

3

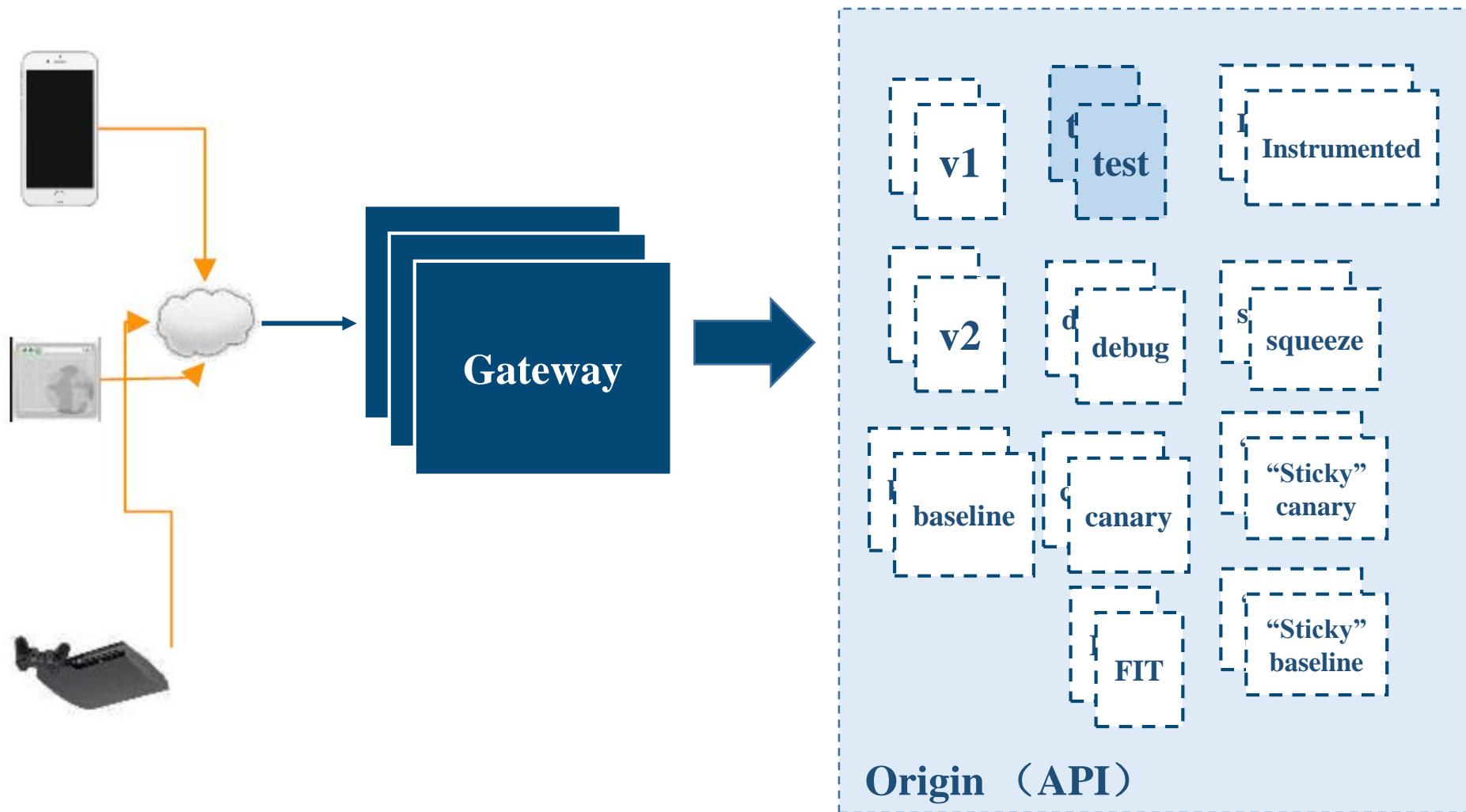
部分

Zuul网关高级应用场景

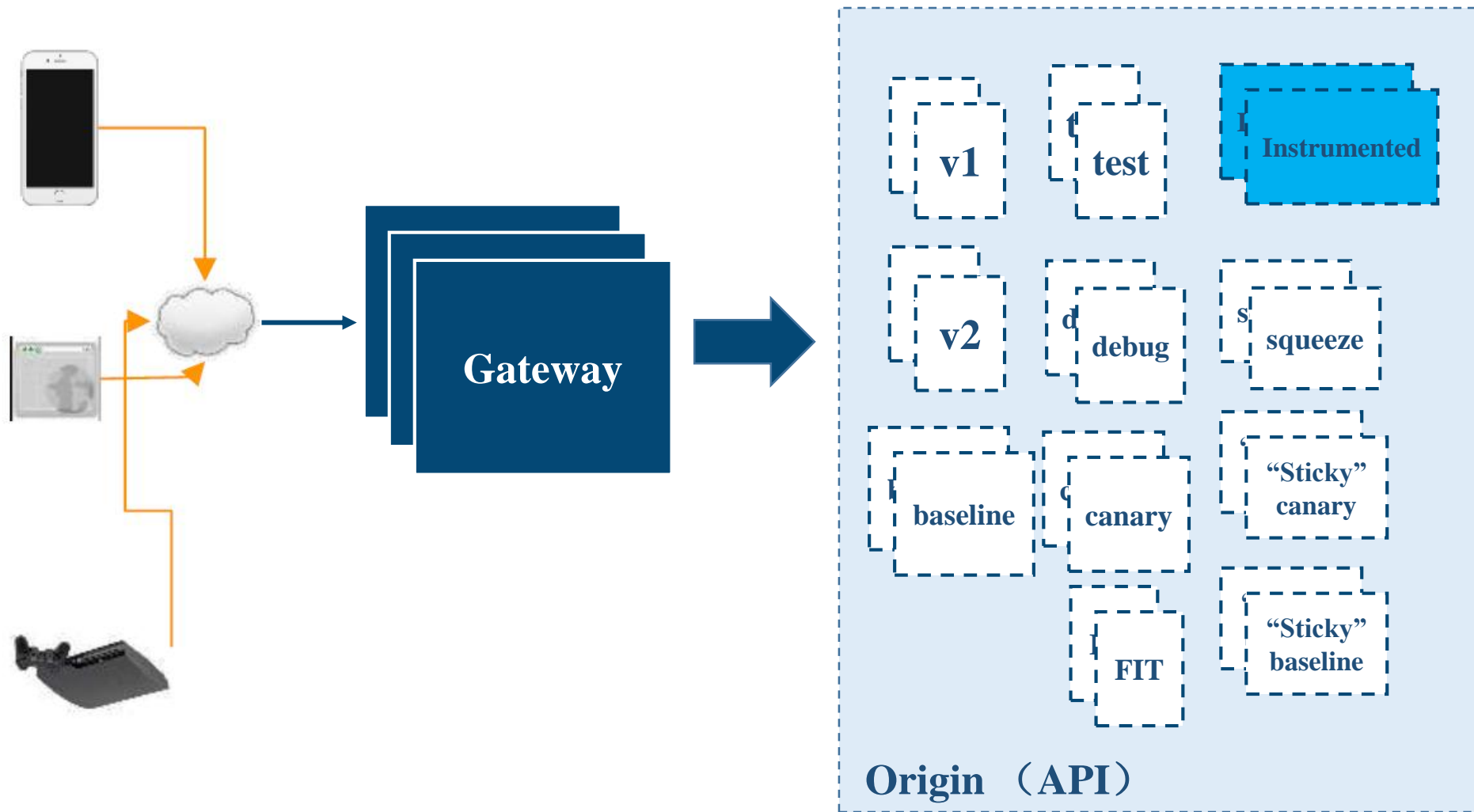
红绿部署



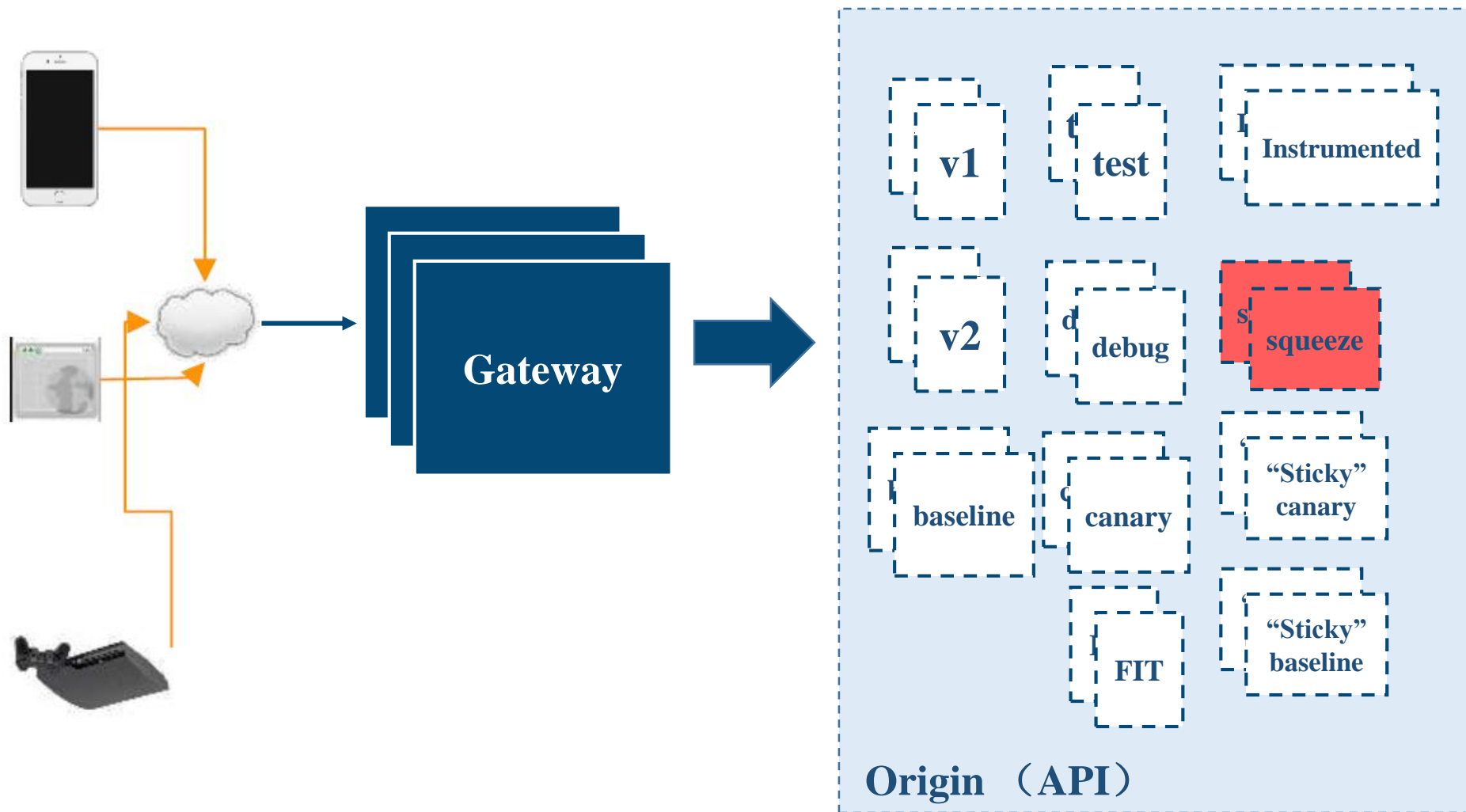
开发者测试分支



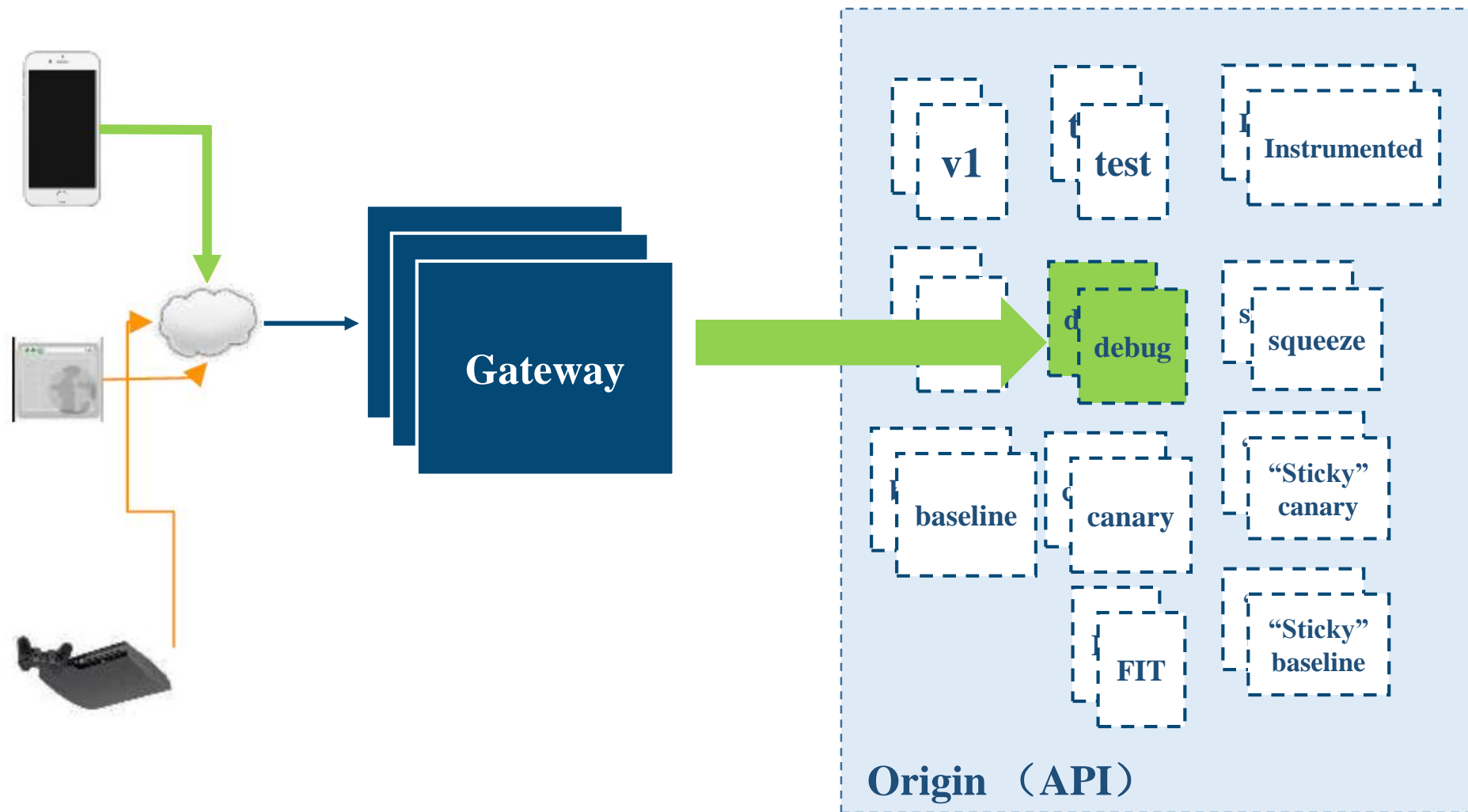
埋点测试



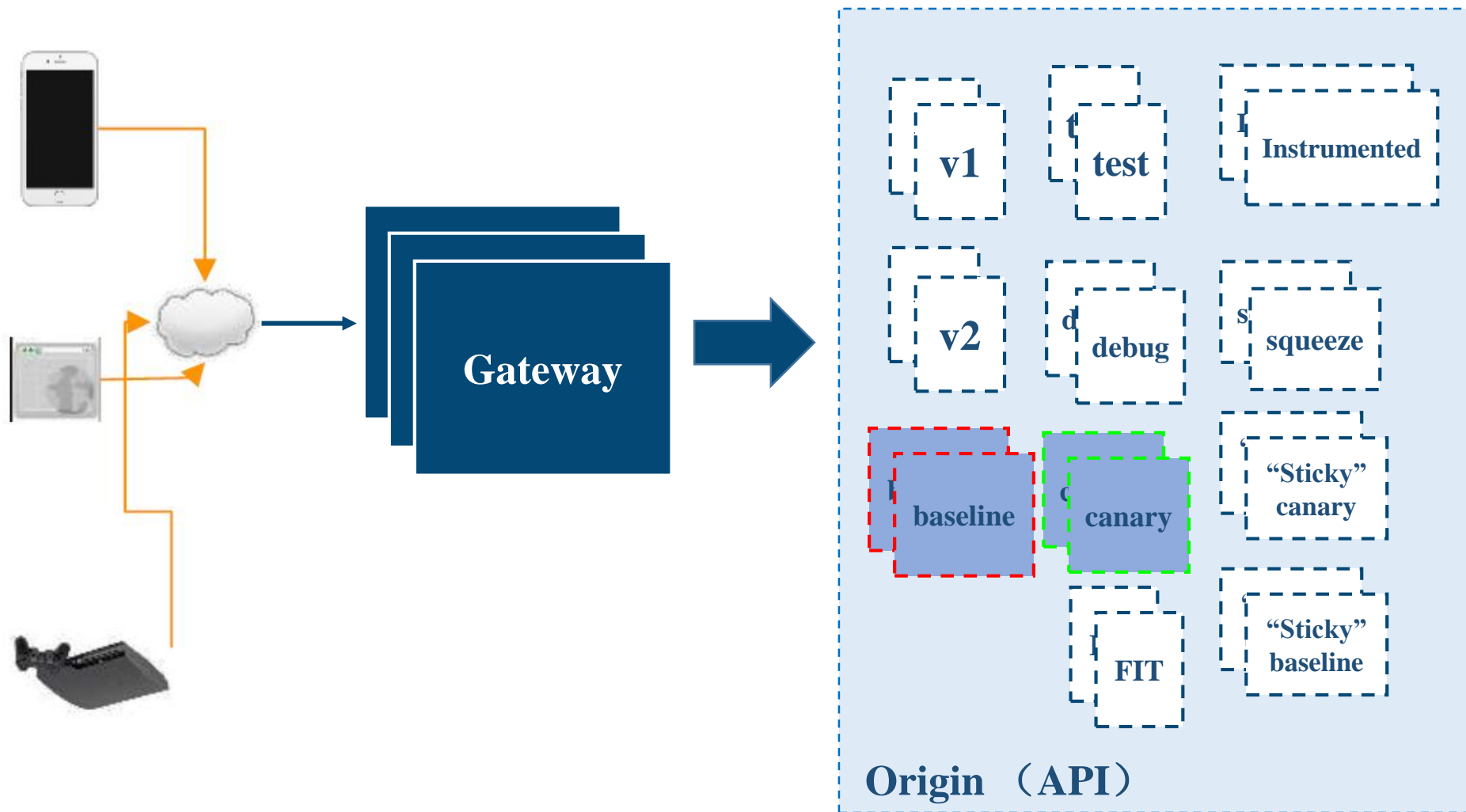
压力测试



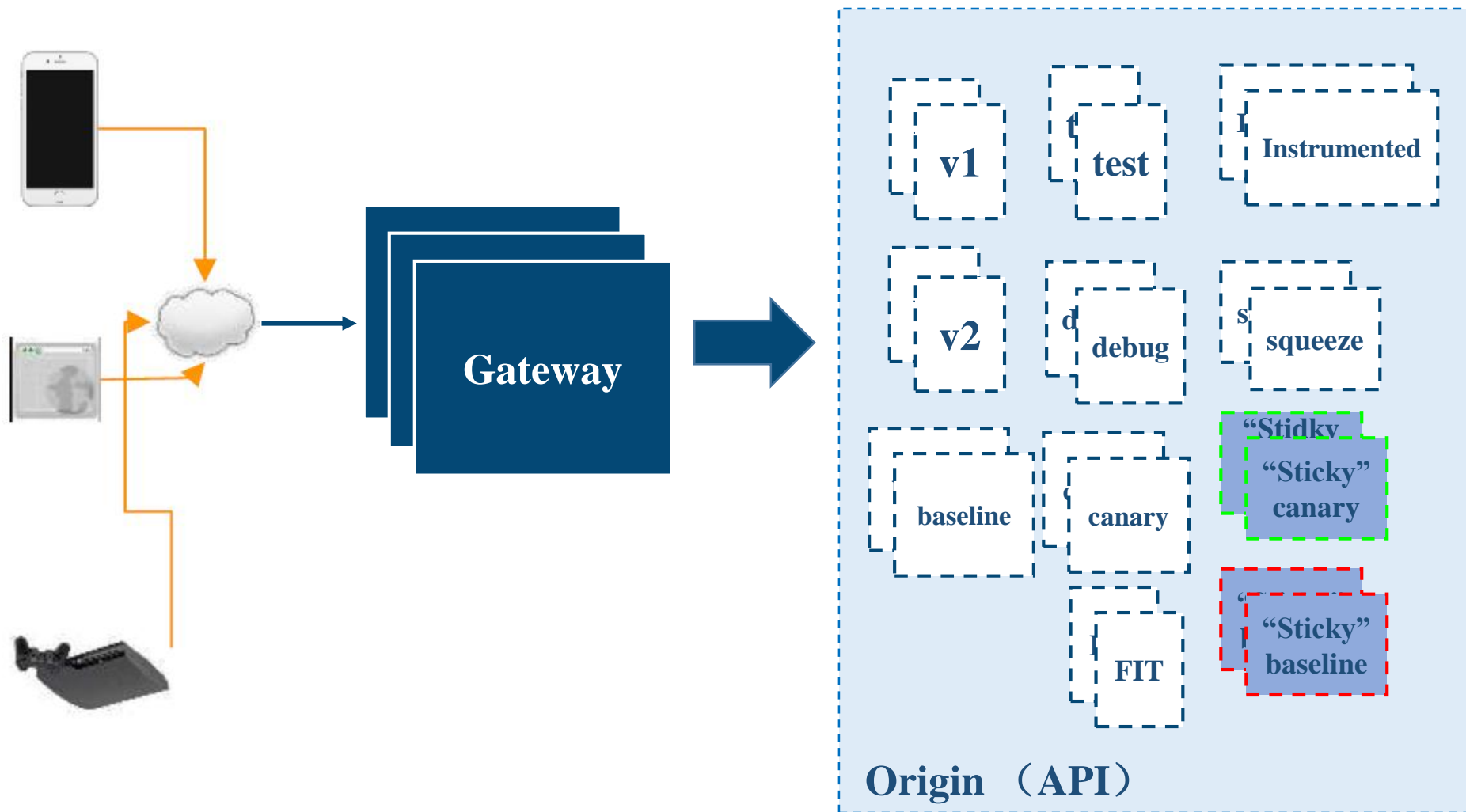
调试路由



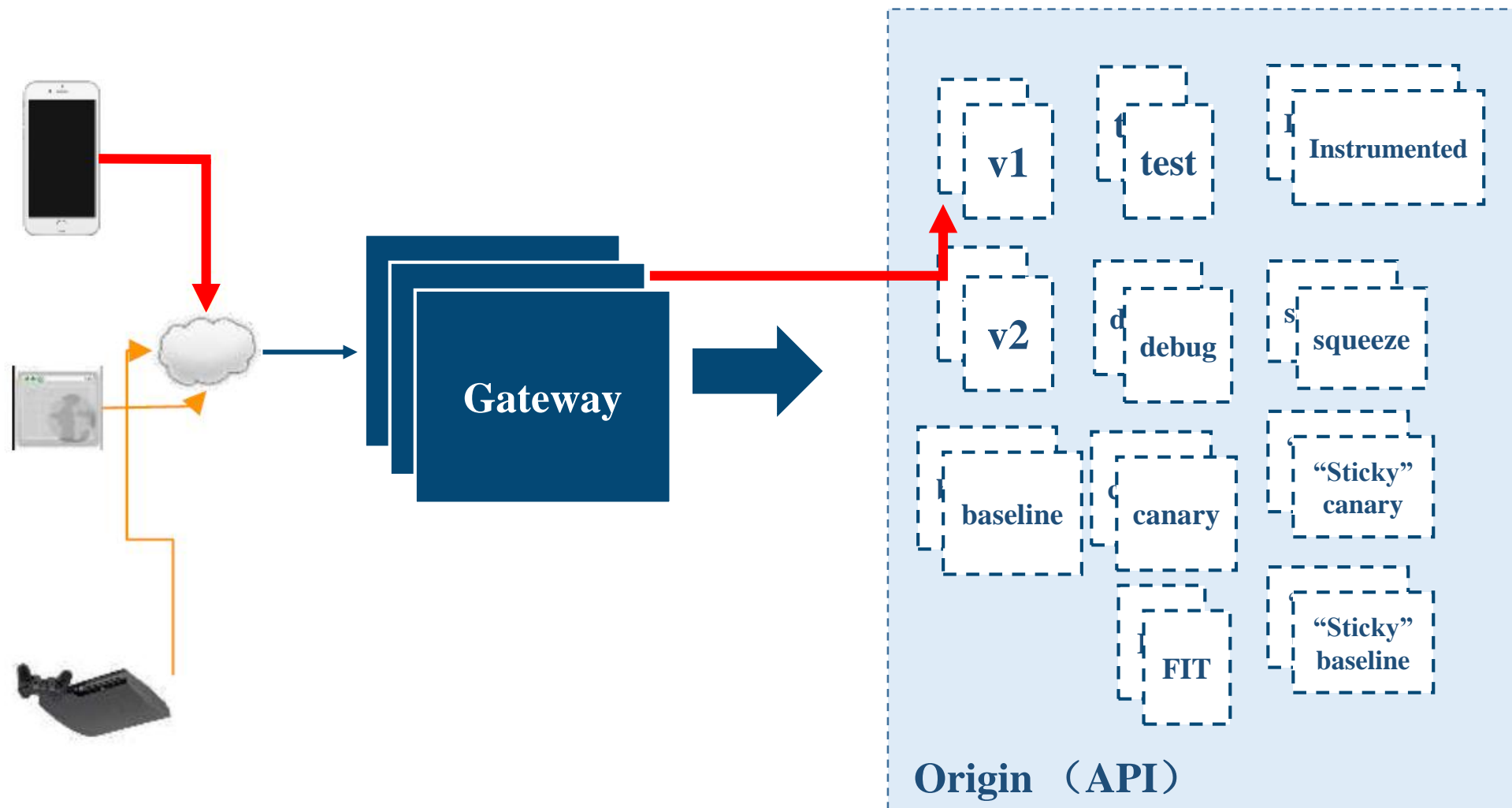
金丝雀测试



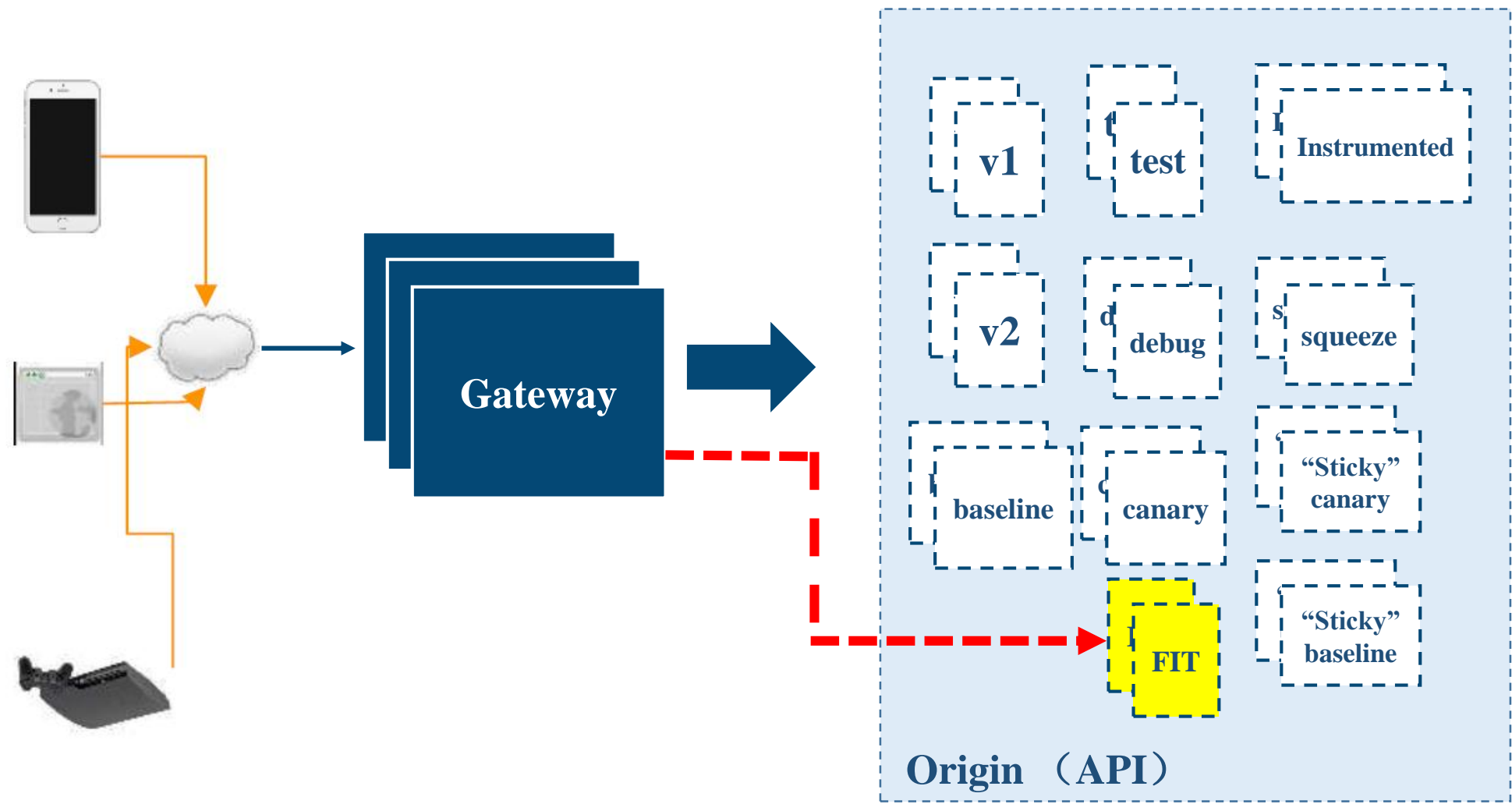
粘性金丝雀



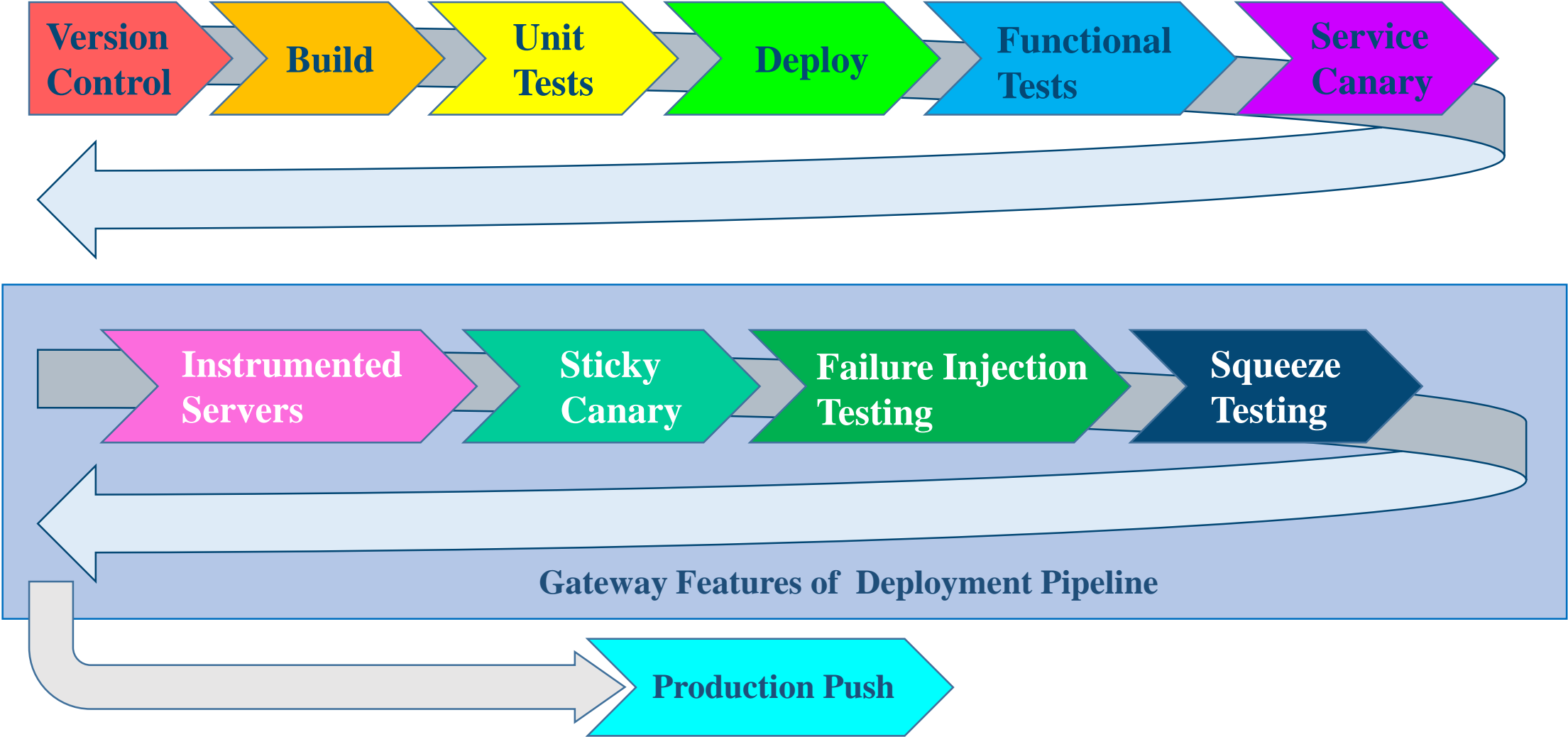
失败注入测试



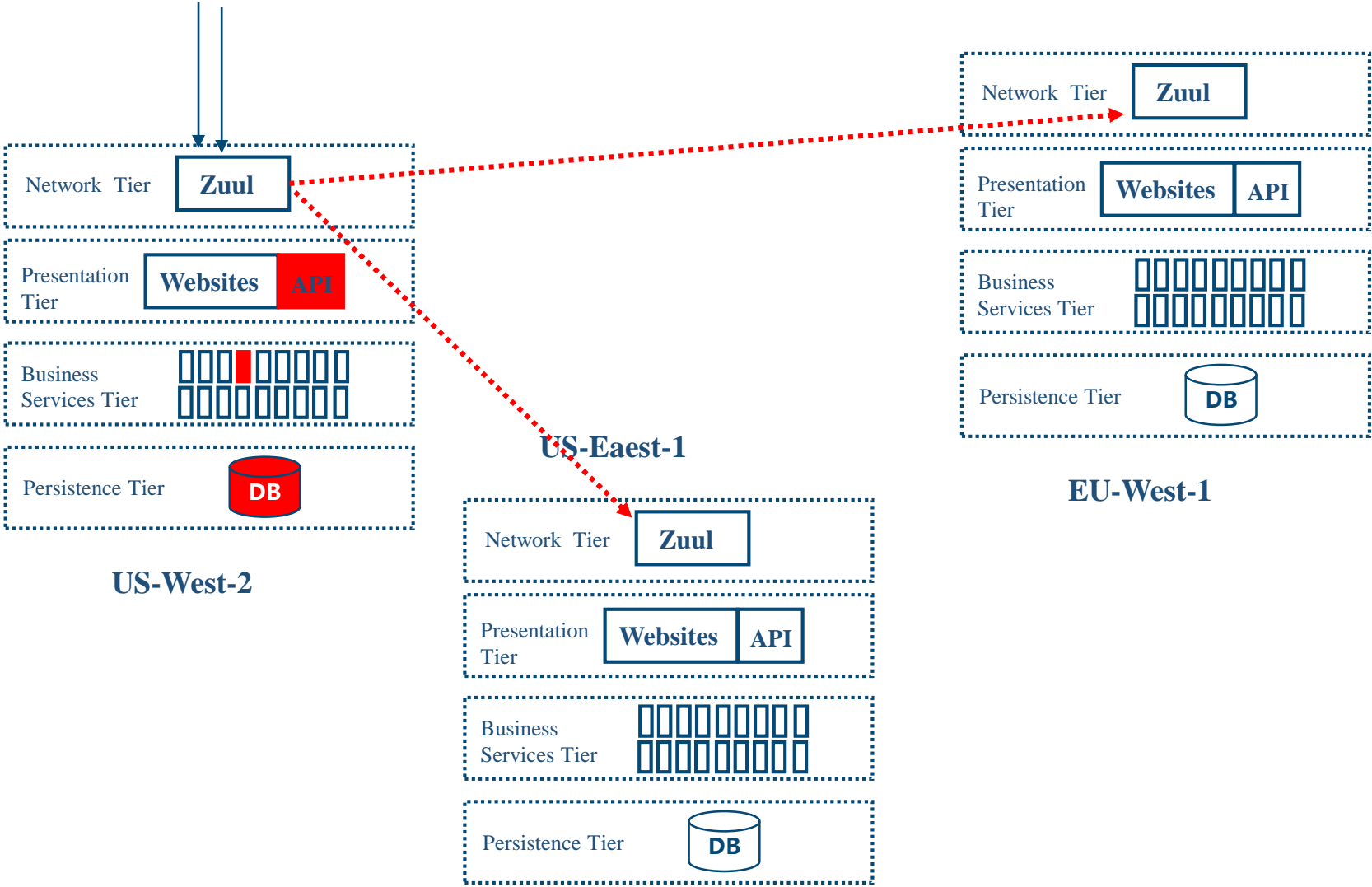
降级测试



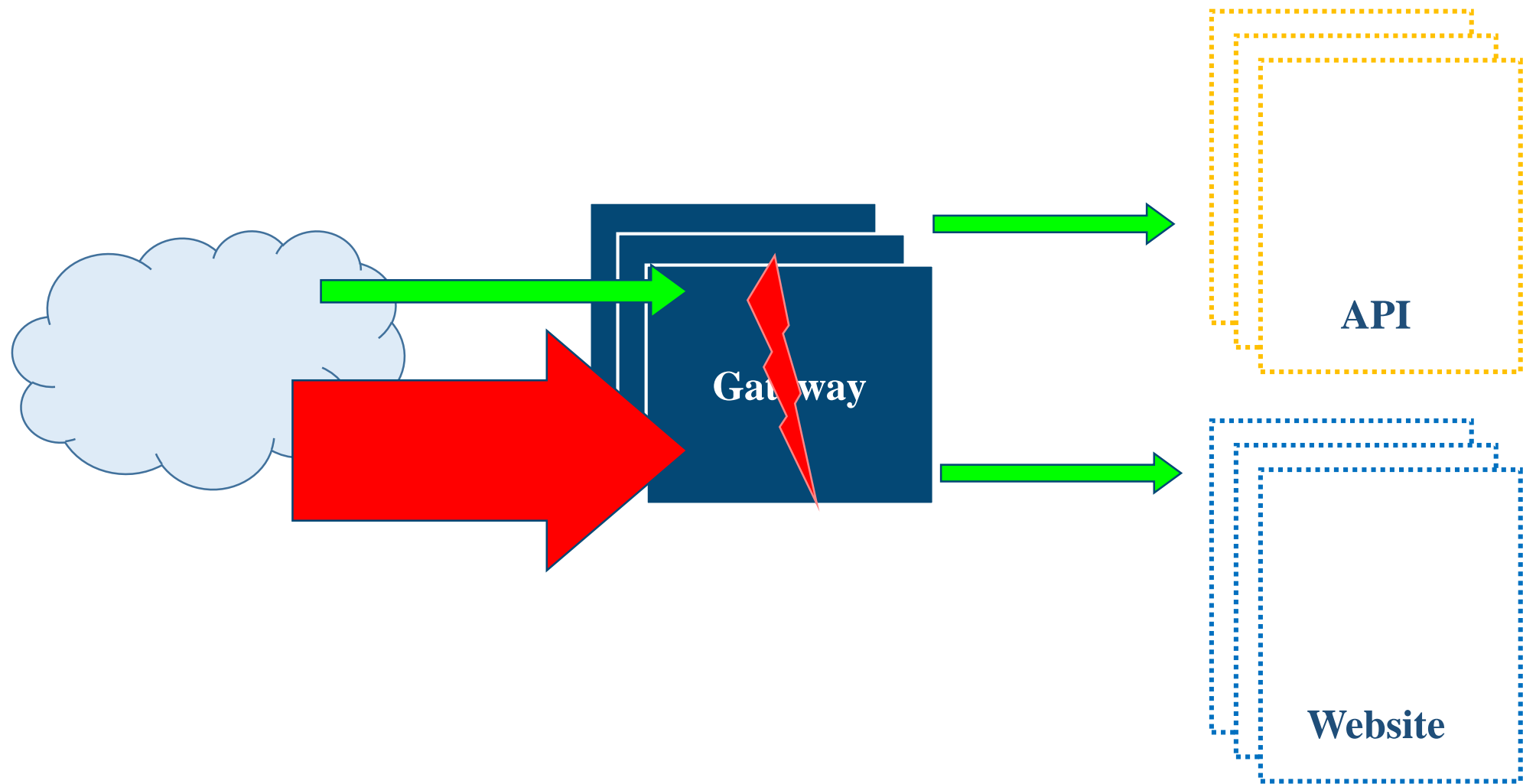
Netflix持续交付流水线



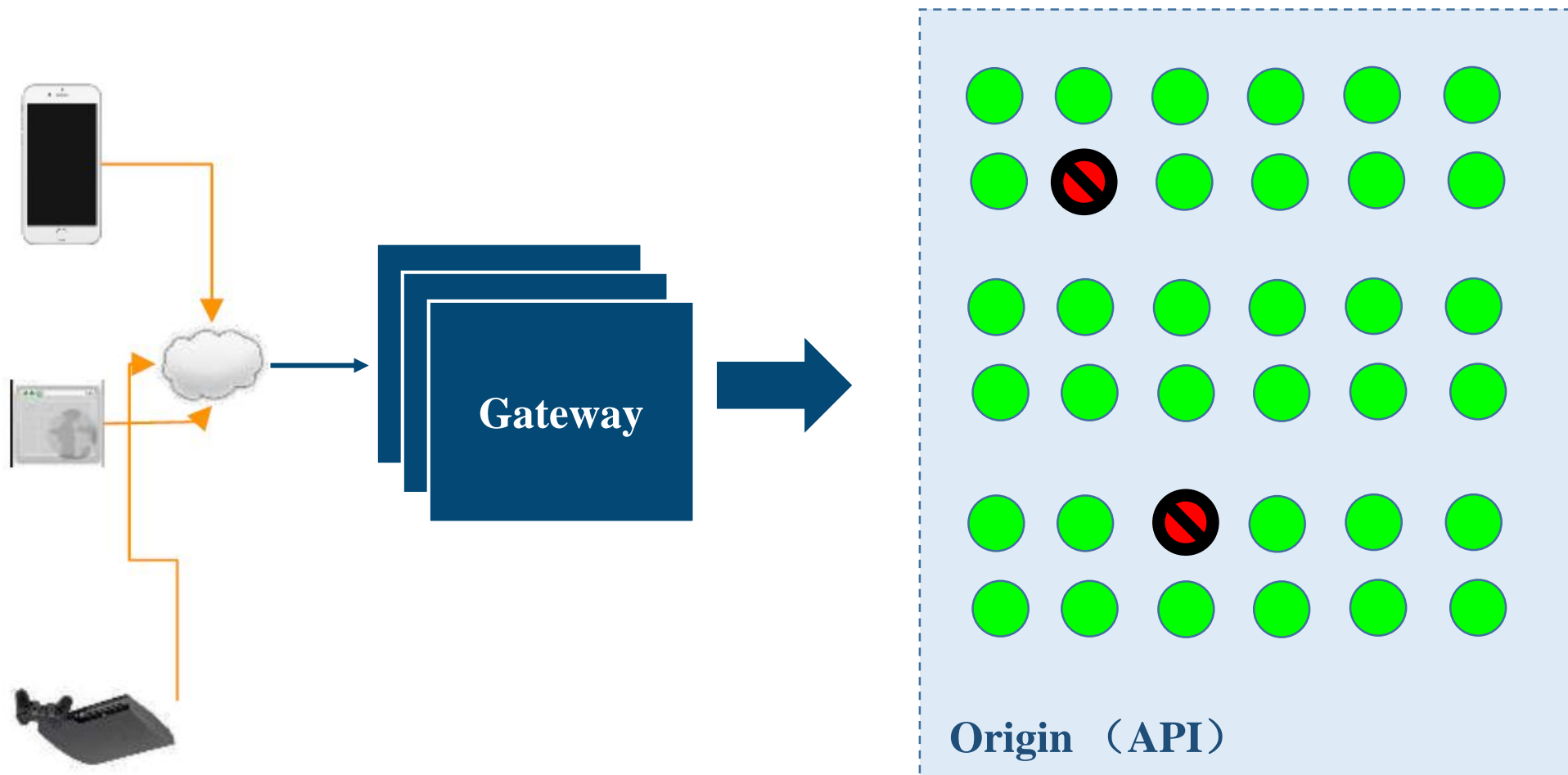
跨区域高可用



防爬防攻击



健康检查和屏蔽坏节点



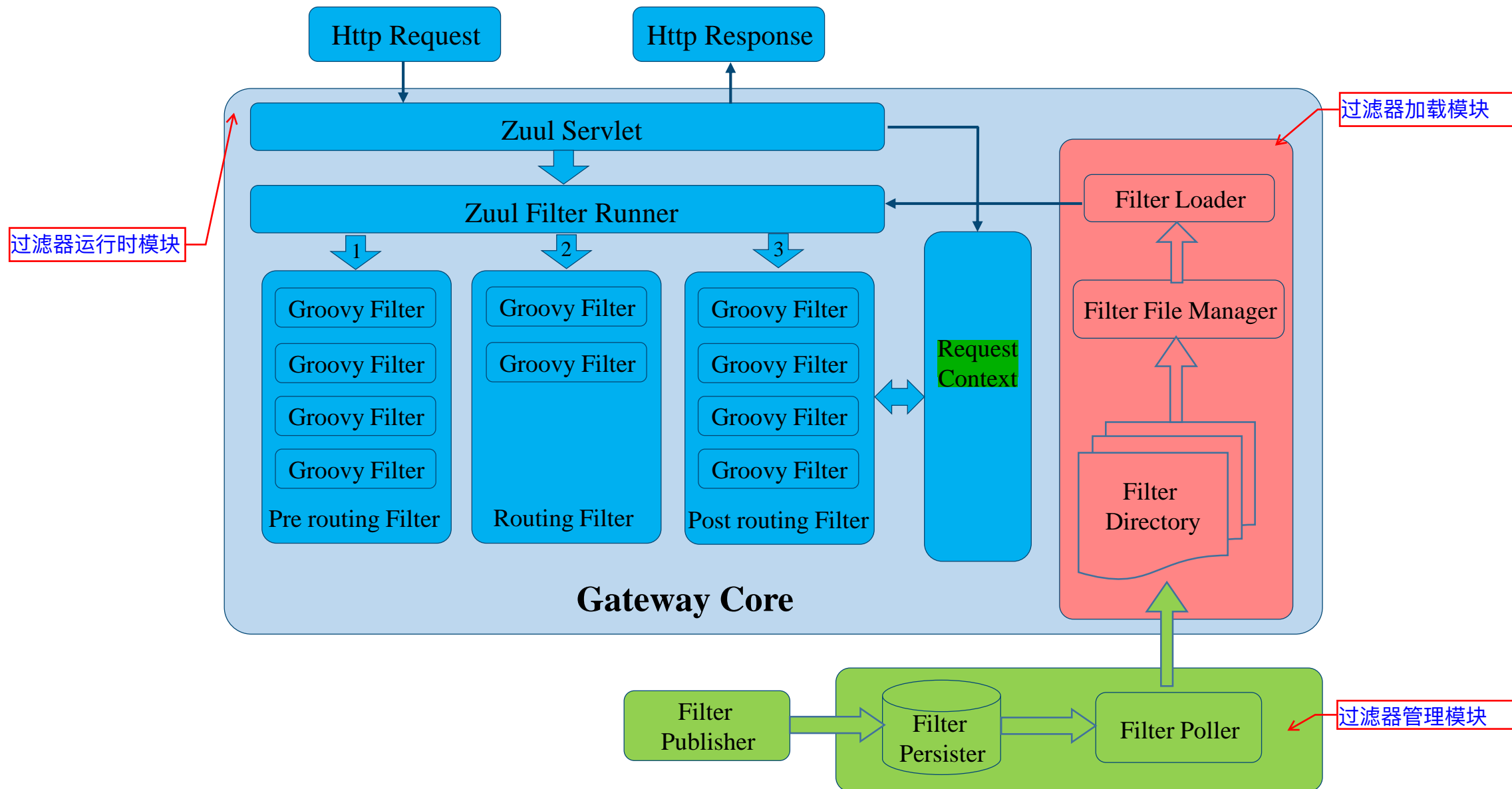
第

4

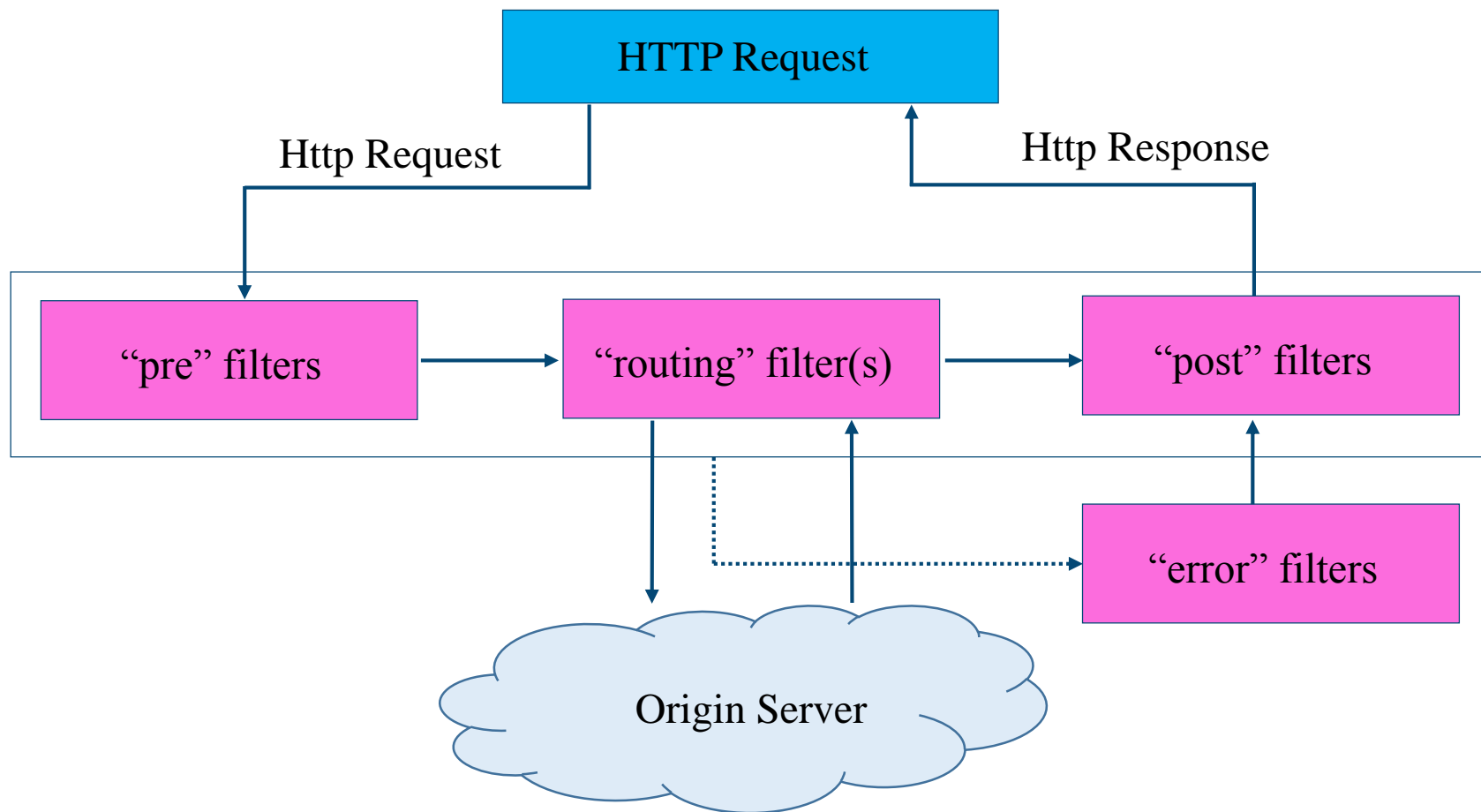
部分

Zuul网关架构剖析

Zuul网关架构



请求处理生命周期



过滤器关键概念

类型Type

定义在路由流程中，过滤器被应用的阶段

执行顺序Execution Order

在同一个Type中，定义过滤器执行的顺序

条件Criteria

过滤器被执行必须满足的条件

动作Action

如果条件满足，过滤器中将被执行的动作

标准过滤器类型

ROUTING :
处理将请求发送到
源服务器的过滤器

ERROR :
上述阶段中出现错
误要执行的过滤器

PRE :
在请求被路由到源服务
器前要执行的过滤器

- 认证
- 选路由
- 请求日志

POST :
在响应从源服务器返回
时要被执行的过滤器

- 对响应增加HTTP头
- 收集统计和度量
- 将响应以流的方式发送回客
户端

过滤器样例

```
class DeviceDelayFilter extends ZuulFilter {  
    def static Random rand = new Random()  
    @Override  
    String filterType() {  
        return 'pre'  
    }  
    @Override  
    int filterOrder() {  
        return 5  
    }  
    @Override  
    boolean shouldFilter() {  
        return RequestContext.getRequest().getParameter("deviceType")?.equals("BrokenDevice"):false  
    }  
    @Override  
    Object run() {  
        sleep(rand.nextInt(20000)) //Sleep for a random number of seconds between [0-20]  
    }  
}
```

Filter管理

H5 Gateway Filter List						
21 filters found				<button>Upload Filter</button>		
Name	Type	Order	Latest Revision	Test Environment	UAT Environment	Prod Environment
HealthCheck	Pre	0	4	Active (4 / -)	Active (4 / -)	Active (4 / -)
DebugModeSetter	Pre	1	2	Active (2 / -)	Active (2 / -)	Active (2 / -)
DebugRequest	Pre	10	1	Active (1 / -)	Active (1 / -)	Active (1 / -)
BotBlocker	Pre	20	10	Active (10 / -)	Active (10 / -)	Active (10 / -)
RoutingForNative	Pre	150	1	Active (1 / -)	Active (1 / -)	Inactive (- / -)
RoutingForSOA2	Pre	200	22	Active (22 / -)	Active (22 / -)	Active (22 / -)
RoutingForESB	Pre	300	8	Active (8 / -)	Active (8 / -)	Inactive (- / -)
RoutingForBusPhp	Pre	301	1	Active (1 / -)	Inactive (- / -)	Active (1 / -)
RoutingForSearch	Pre	306	2	Active (2 / -)	Active (2 / -)	Active (2 / -)
RoutingForGolf	Pre	308	1	Active (1 / -)	Active (1 / -)	Active (1 / -)
				<a>← Previous	Displaying 1 to 10 of 21	<a>Next →

第

5

部分

Zuul网关代码剖析(Code Review)

第

6

部分

Zuul过滤器管理工具(Code Review)

第

7

部分

过滤器实验~前置过滤器(Lab01)

第

8

部分

过滤器实验~路由过滤器 (Lab02)

第

9

部分

过滤器实验~后置过滤器 (Lab03)

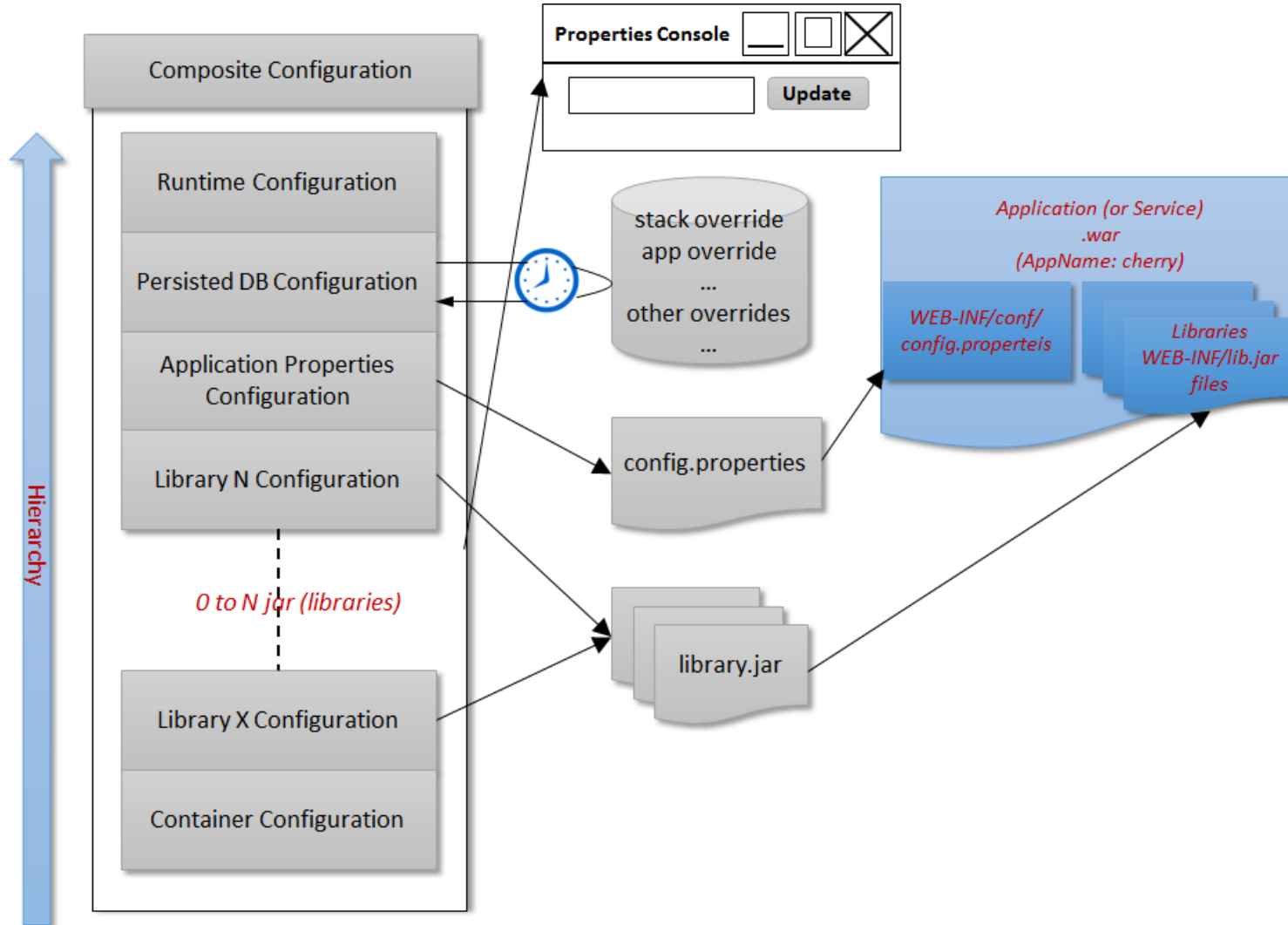
第

10

部分

Zuul网关对接Apollo (Lab04)

Netflix Archaius



Zuul对接Apollo

<https://github.com/netflix/archaius>



`{config_server_url}/configfiles/{appId}/{clusterName}/{namespaceName}`

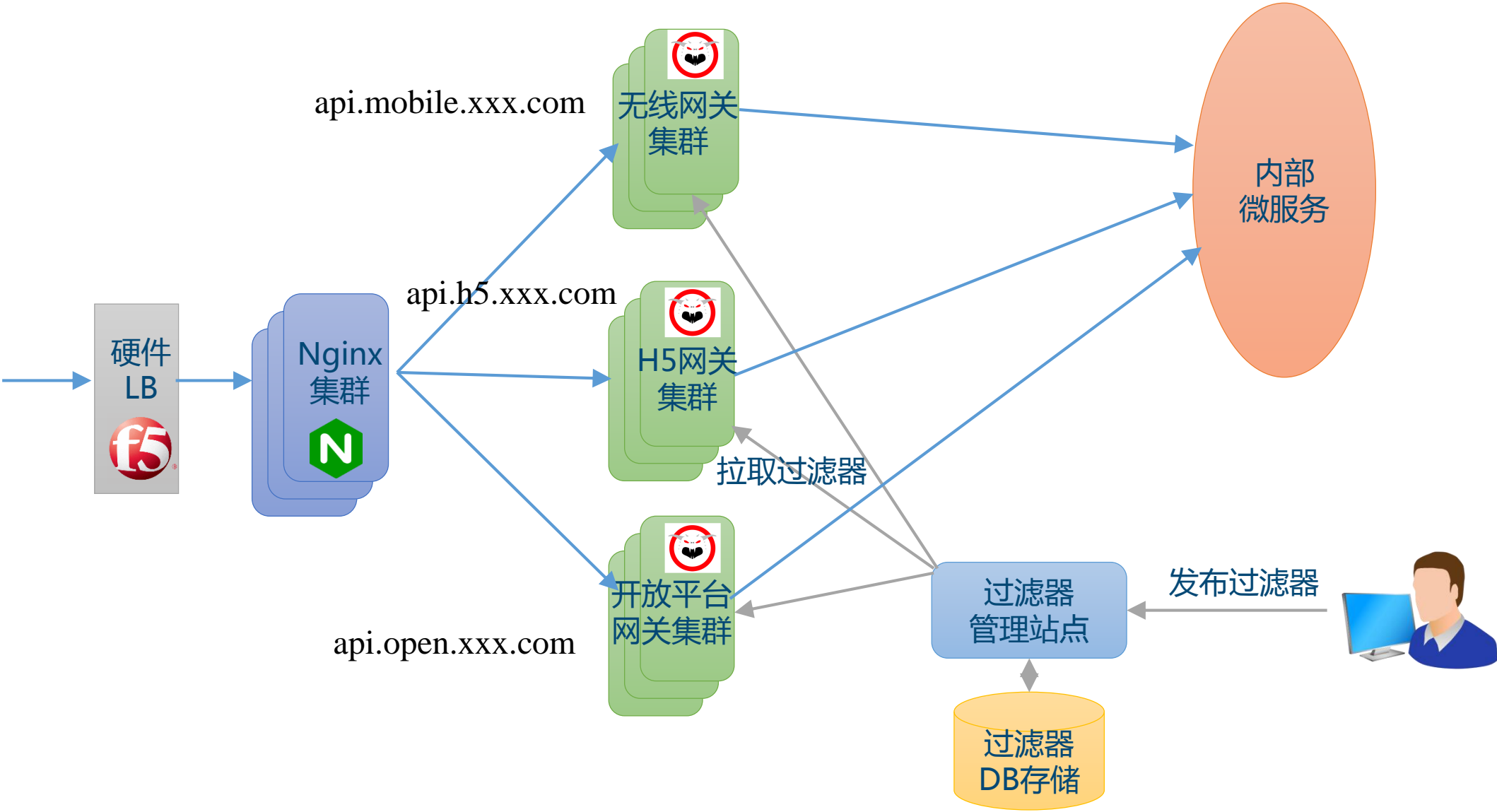
第

11

部分

Zuul网关生产部署实践

参考部署架构案例



分集群过滤器管理

Api管理系统

Filter管理

filter列表

上传文件位置：

请选择

上传filter文件

所有网关filter

无线网关

h5网关

第三方网关

ID	过滤器Id	名称	类型	Order	禁用属性	版本	状态	创建时间	操作
79	wirelessgate:SearchRoute:pre	SearchRoute	pre	20	gate.SearchRoute.pre.disable	2	<div>灰度</div> <div>上线</div> <div>下线</div>	2018-04-19 19:12:58	<div>详情</div> <div>删除</div>
65	wirelessgate:ExecuteRoute:route	ExecuteRoute	route	20	gate.ExecuteRoute.route.disable	2	<div>灰度</div> <div>上线</div> <div>下线</div>	2018-01-19 11:18:32	<div>详情</div> <div>删除</div>
62	wirelessgate:ValidateToken:pre	ValidateToken	pre	40	gate.ValidateToken.pre.disable	5	<div>灰度</div> <div>上线</div> <div>下线</div>	2017-11-28 16:19:44	<div>详情</div> <div>删除</div>
60	wirelessgate:ValidateToken:pre	ValidateToken	pre	40	gate.ValidateToken.pre.disable	3	<div>灰度</div> <div>下线</div>	2017-11-22 19:04:57	<div>详情</div> <div>删除</div>
55	wirelessgate:ValidateToken:pre	ValidateToken	pre	40	gate.ValidateToken.pre.disable	2	<div>灰度</div> <div>下线</div>	2017-09-21 16:19:19	<div>详情</div> <div>删除</div>
54	wirelessgate:BotBlocker:post	BotBlocker	post	-200	gate.BotBlocker.post.disable	1	<div>灰度</div> <div>上线</div> <div>下线</div>	2017-09-14 17:31:40	<div>详情</div> <div>删除</div>
38	wirelessgate:ExecuteRoute:route	ExecuteRoute	route	20	gate.ExecuteRoute.route.disable	1	<div>灰度</div> <div>下线</div>	2017-08-12 00:08:03	<div>详情</div> <div>删除</div>
37	wirelessgate:ValidateToken:pre	ValidateToken	pre	40	gate.ValidateToken.pre.disable	1	<div>灰度</div> <div>下线</div>	2017-08-12 00:08:00	<div>详情</div> <div>删除</div>
36	wirelessgate:ValidateTimeStamp:pre	ValidateTimeStamp	pre	30	gate.ValidateTimeStamp.pre.disable	1	<div>灰度</div> <div>上线</div> <div>下线</div>	2017-08-12 00:07:56	<div>详情</div> <div>删除</div>
35	wirelessgate:ValidateHeader:pre	ValidateHeader	pre	10	gate.ValidateHeader.pre.disable	1	<div>灰度</div> <div>上线</div> <div>下线</div>	2017-08-12 00:07:54	<div>详情</div> <div>删除</div>

共 20 条

10 条/页

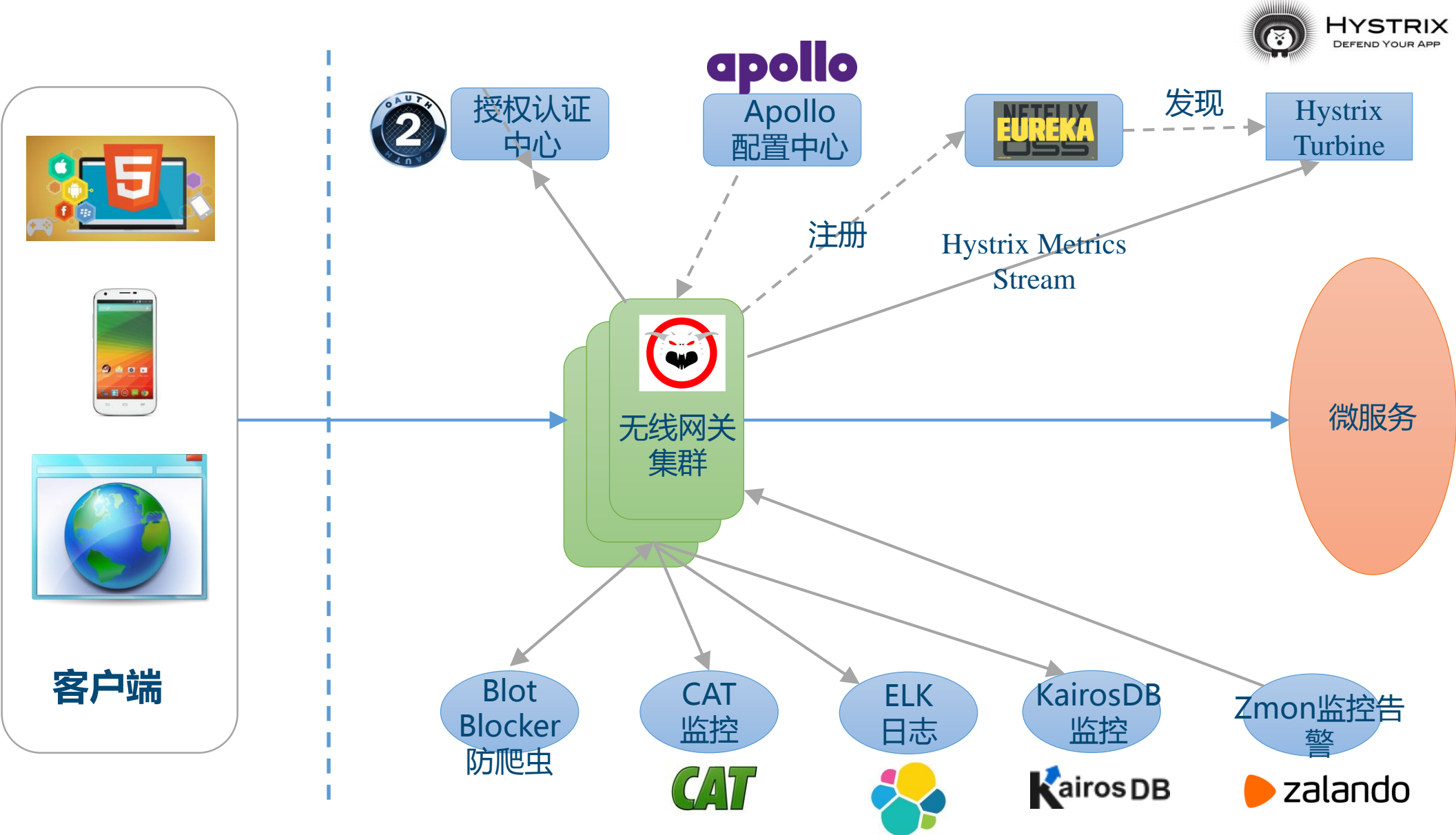
<12>

前往

1

页

网关生产级部署实践



Zuul网关集成Hystrix实时监控

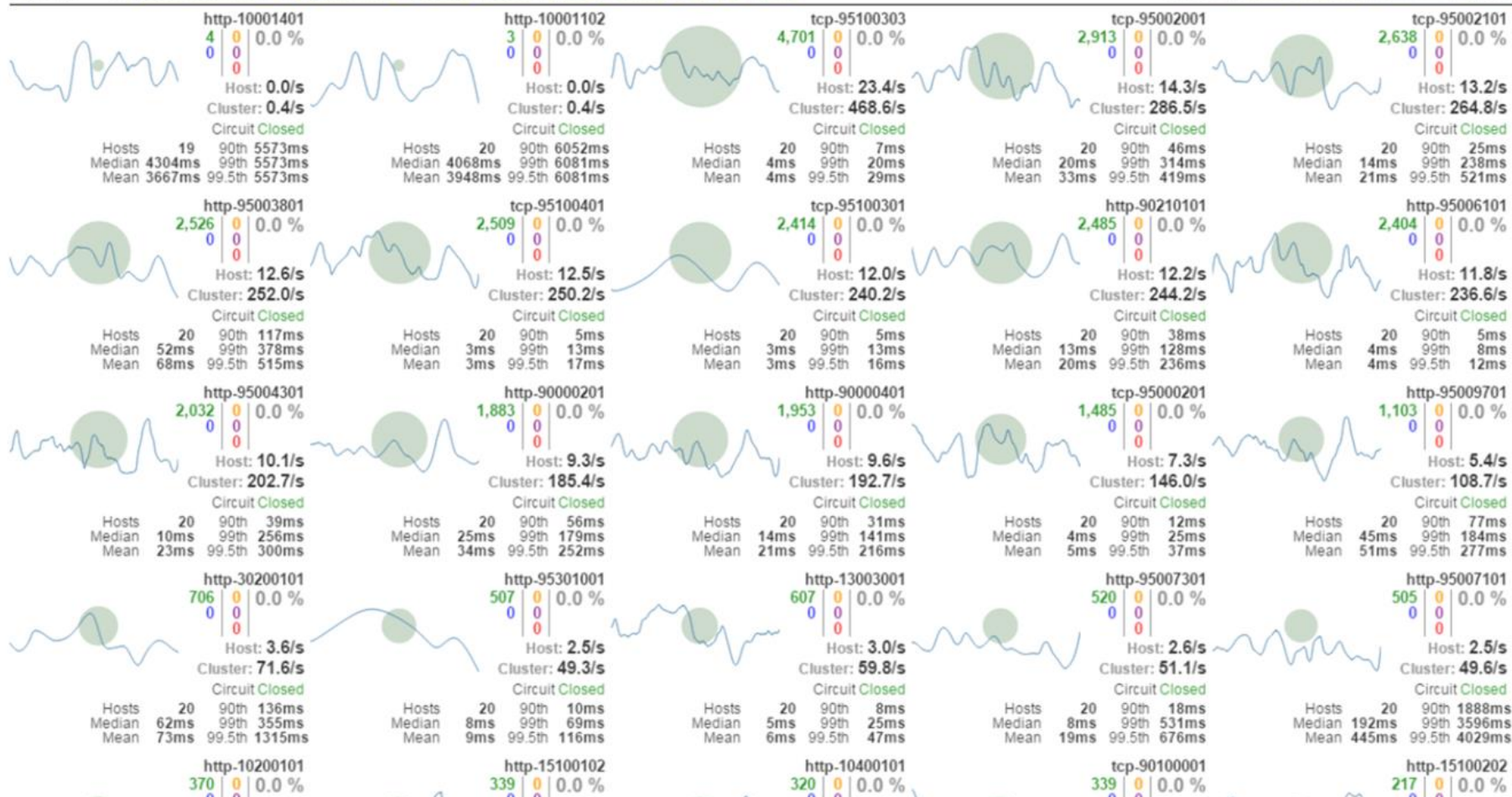
Hystrix Stream: <http://turbine.soa.ctripcorp.com/turbine/turbine.stream?cluster=tcp-gatekeeper>



HYSTRIX
DEFEND YOUR APP

Circuit Sort: [Error then Volume](#) | [Alphabetical](#) | [Volume](#) | [Error](#) | [Mean](#) | [Median](#) | [90](#) | [99](#) | [99.5](#)

[Success](#) | [Short-Circuited](#) | [Timeout](#) | [Rejected](#) | [Failure](#) | [Error %](#)



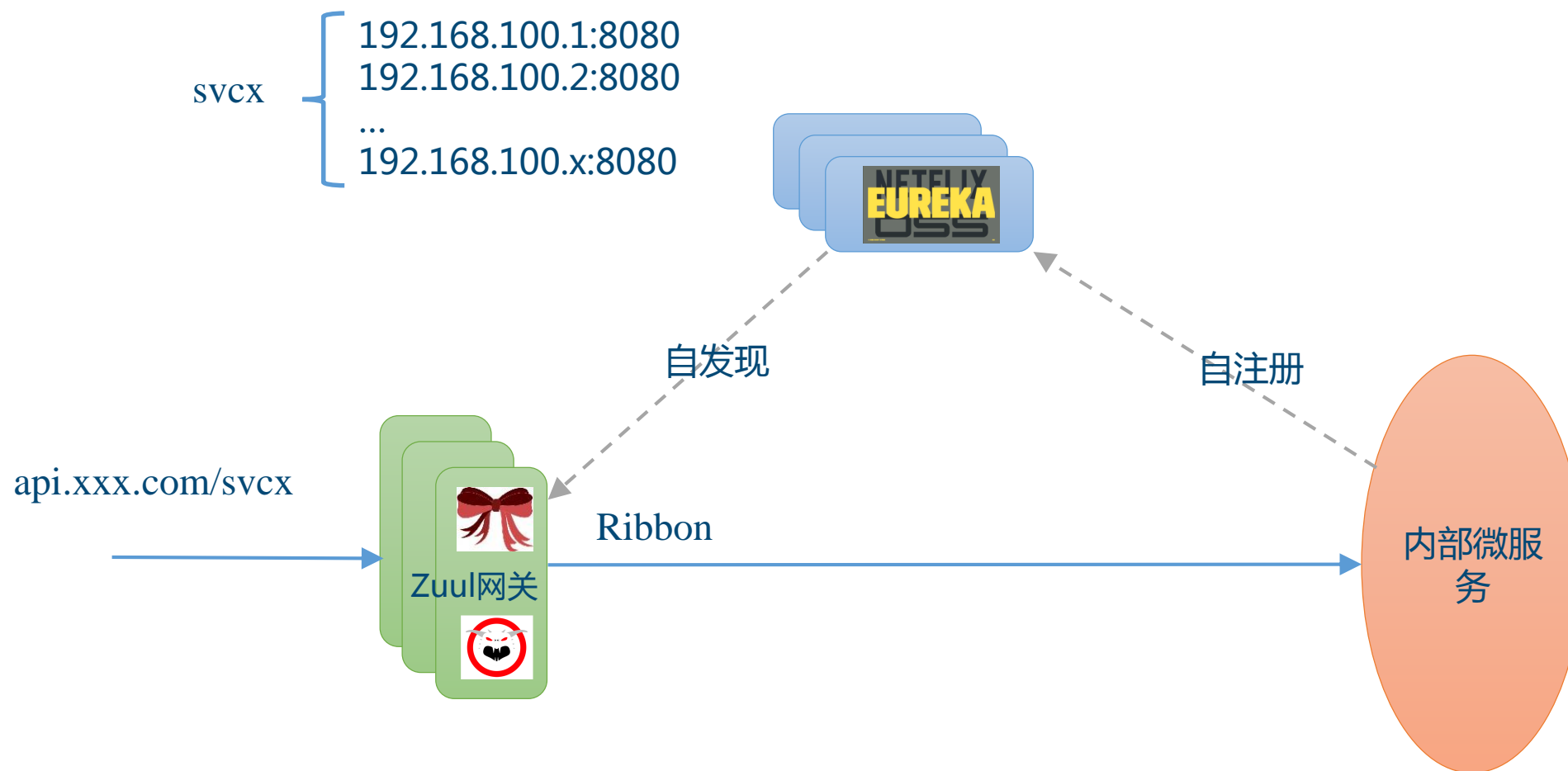
第

12

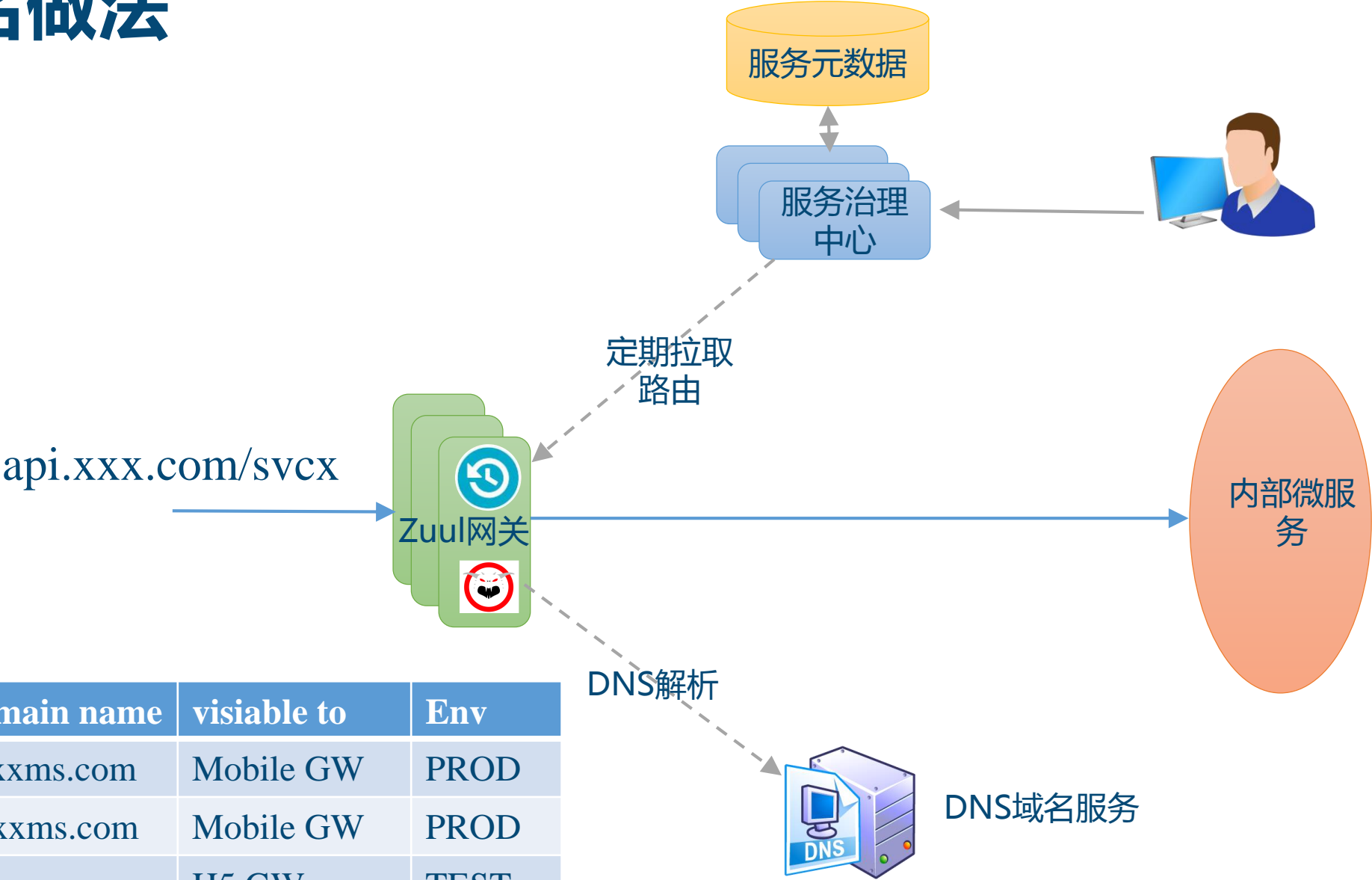
部分

Zuul网关路由管理实践

基于Eureka自发现(Netflix做法)

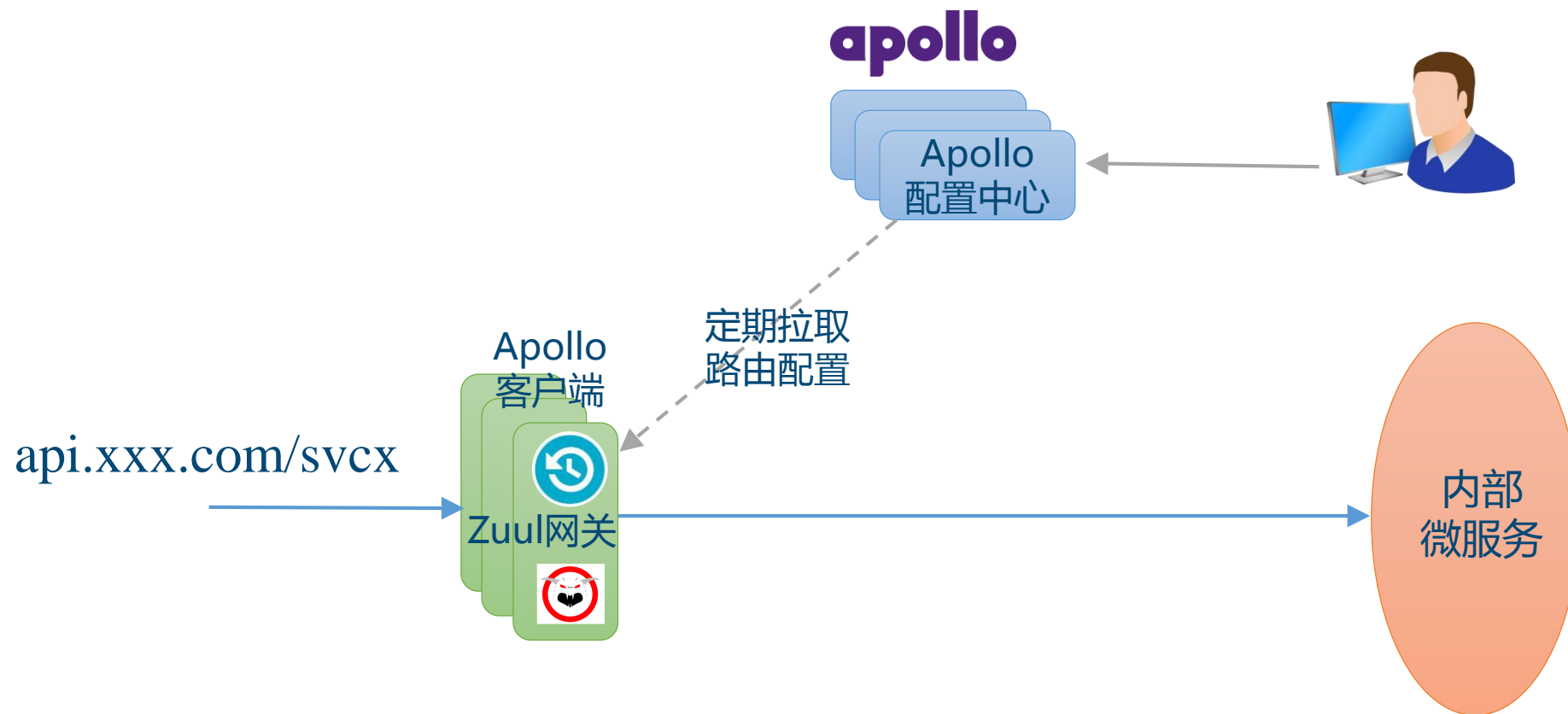


基于域名做法



svcName	svc domain name	visiable to	Env
svca	svca.xxxms.com	Mobile GW	PROD
svcb	svcb.xxxms.com	Mobile GW	PROD
Svcx	svcx.xxxms.com	H5 GW	TEST

简单基于Apollo做法



第

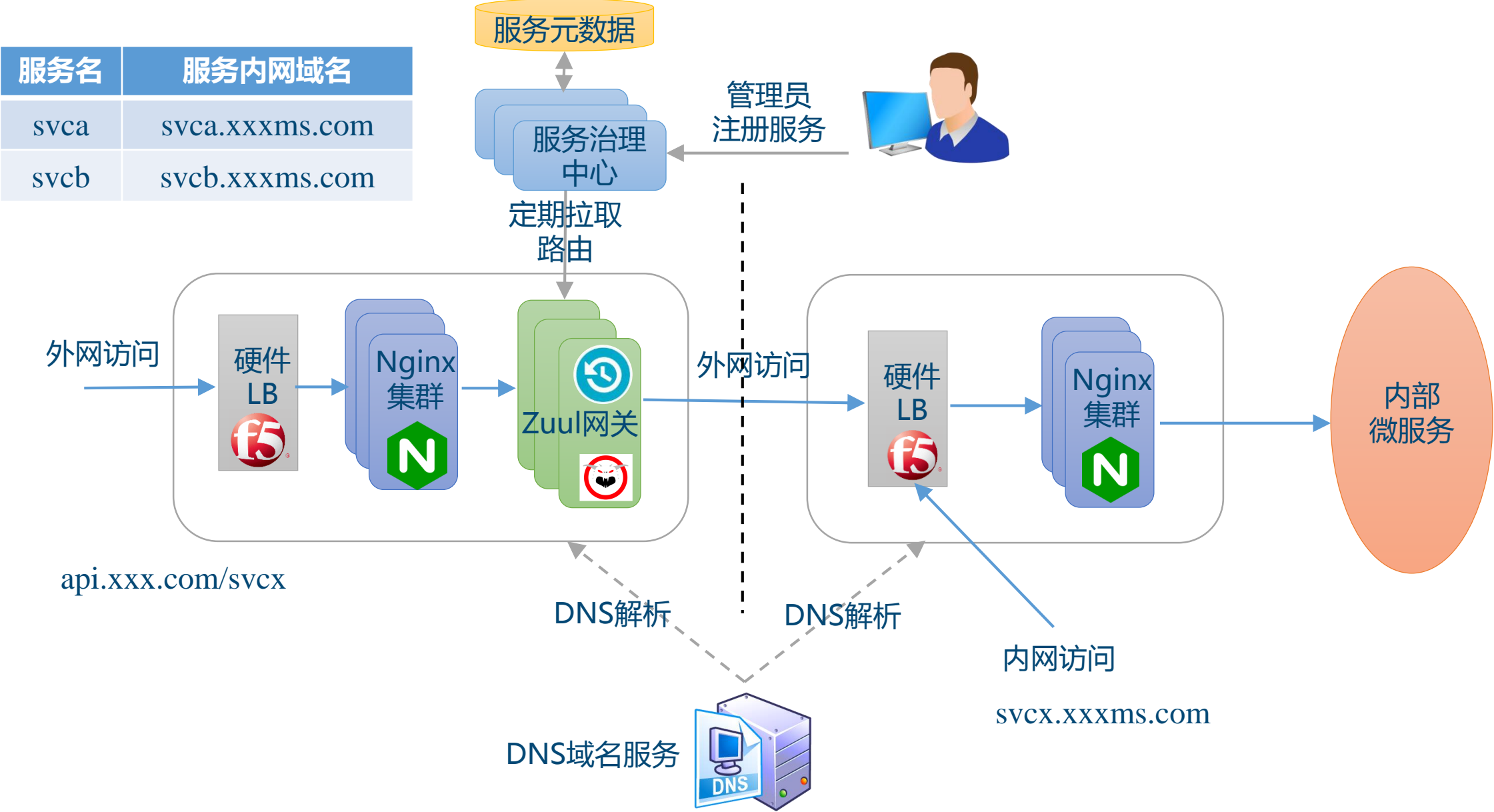
13

部分

基于网关的两层路由体系

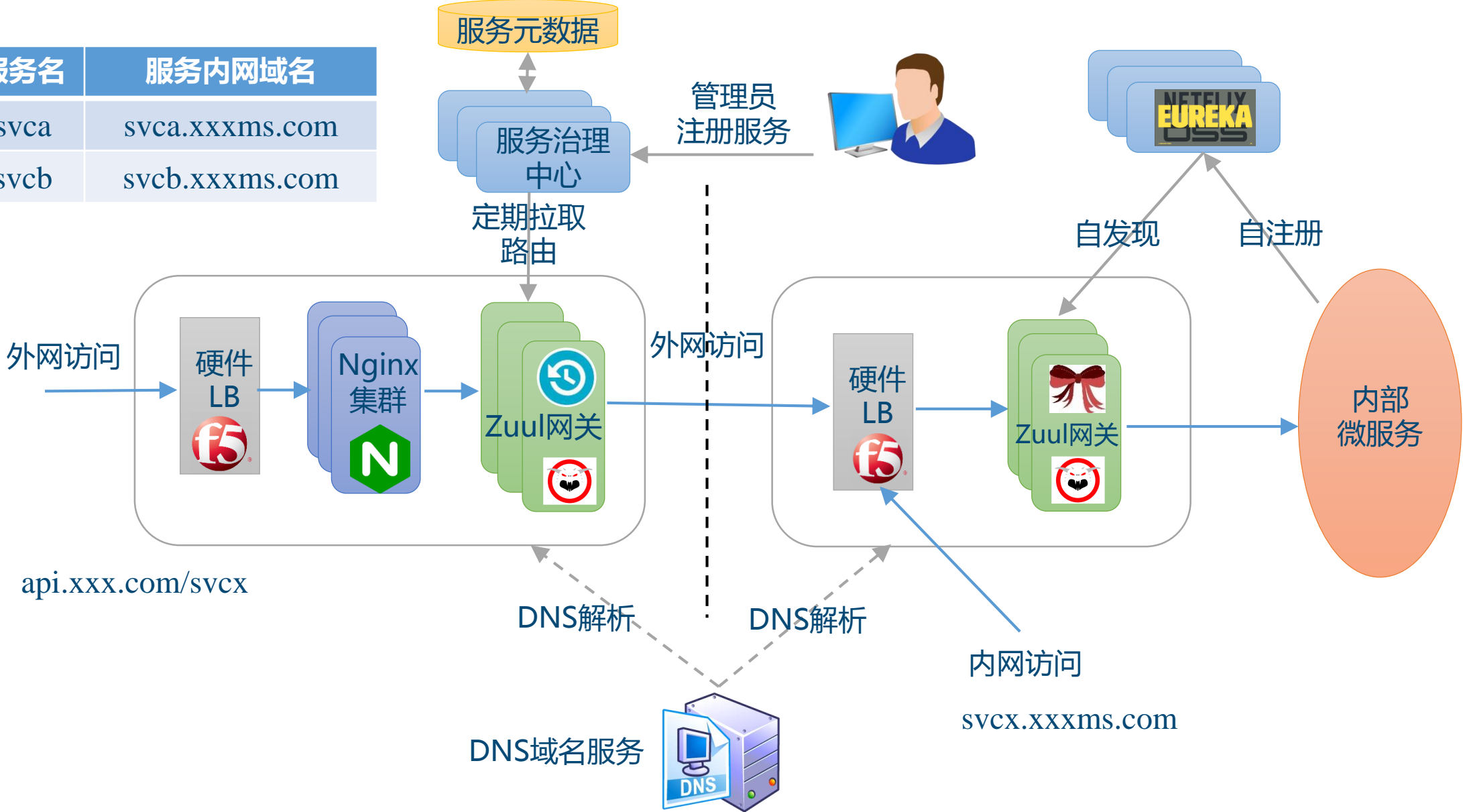
内网Nginx静态配置做法

服务名	服务内网域名
svca	svca.xxxms.com
svcb	svcb.xxxms.com



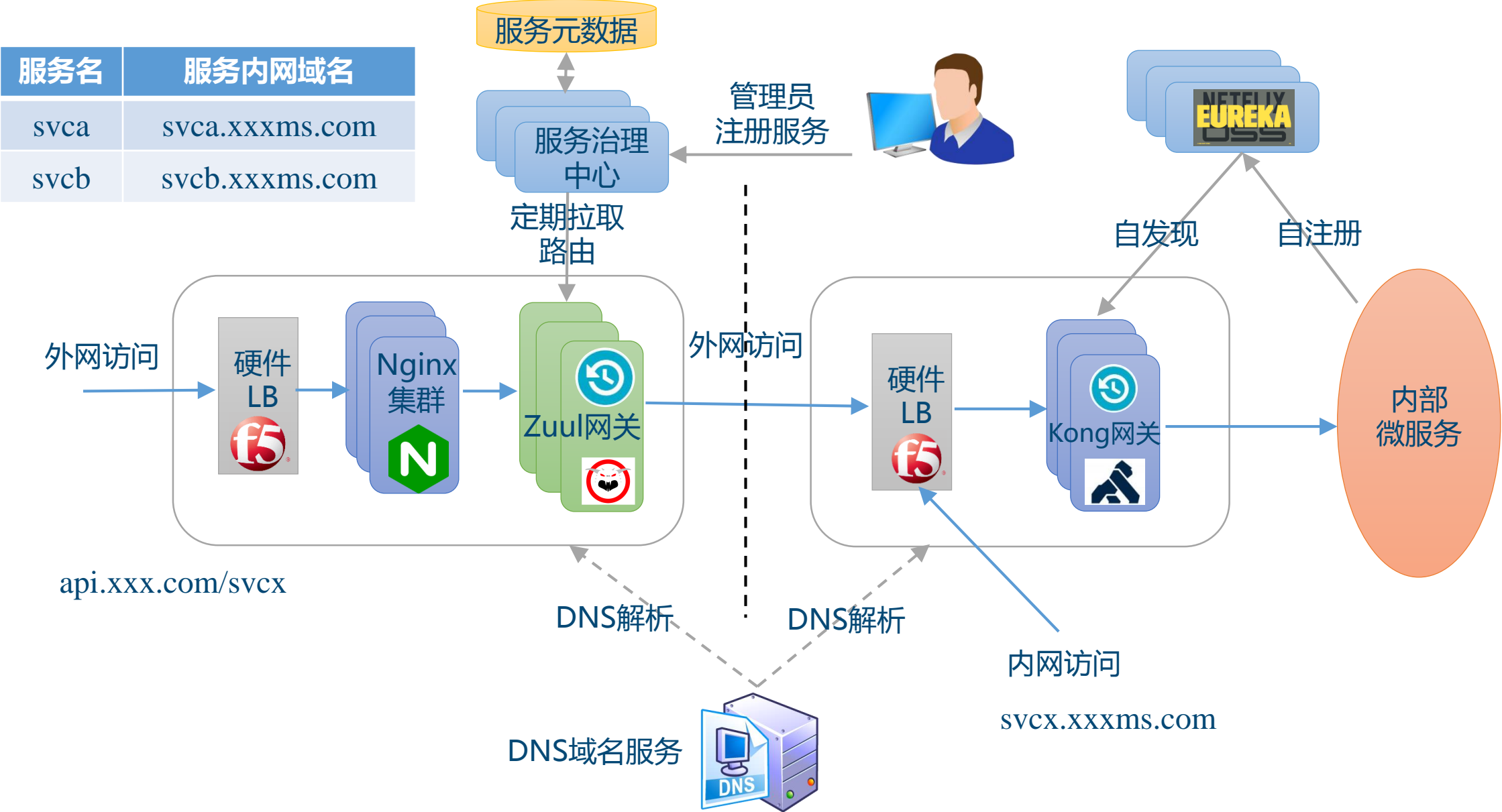
内网Zuul动态配置做法

服务名	服务内网域名
svca	svca.xxxms.com
svcb	svcb.xxxms.com



内网kong动态配置做法

服务名	服务内网域名
svca	svca.xxxms.com
svcb	svcb.xxxms.com



第

14

部分

Spring Cloud Zuul (Lab05)

Spring Cloud Zuul简介

Netflix Zuul内核+Spring Boot

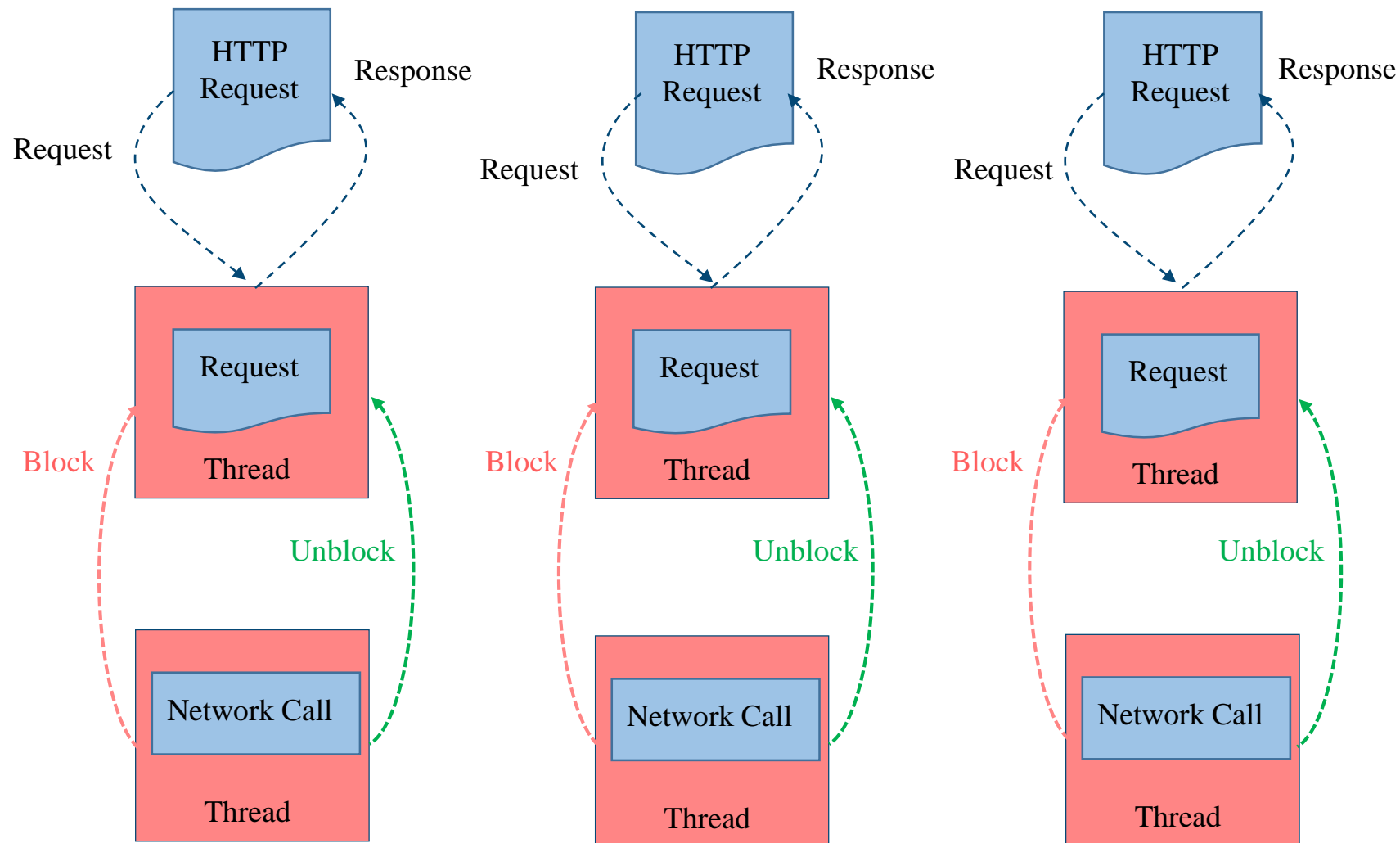
EnableZuulProxy标注

去掉了动态过滤器加载！

第 15 部分

Zuul 2.0简介

Zuul 1~阻塞多线程模式



阻塞同步模式优劣

优

- 编程模型简单
- 开发调试运维简单



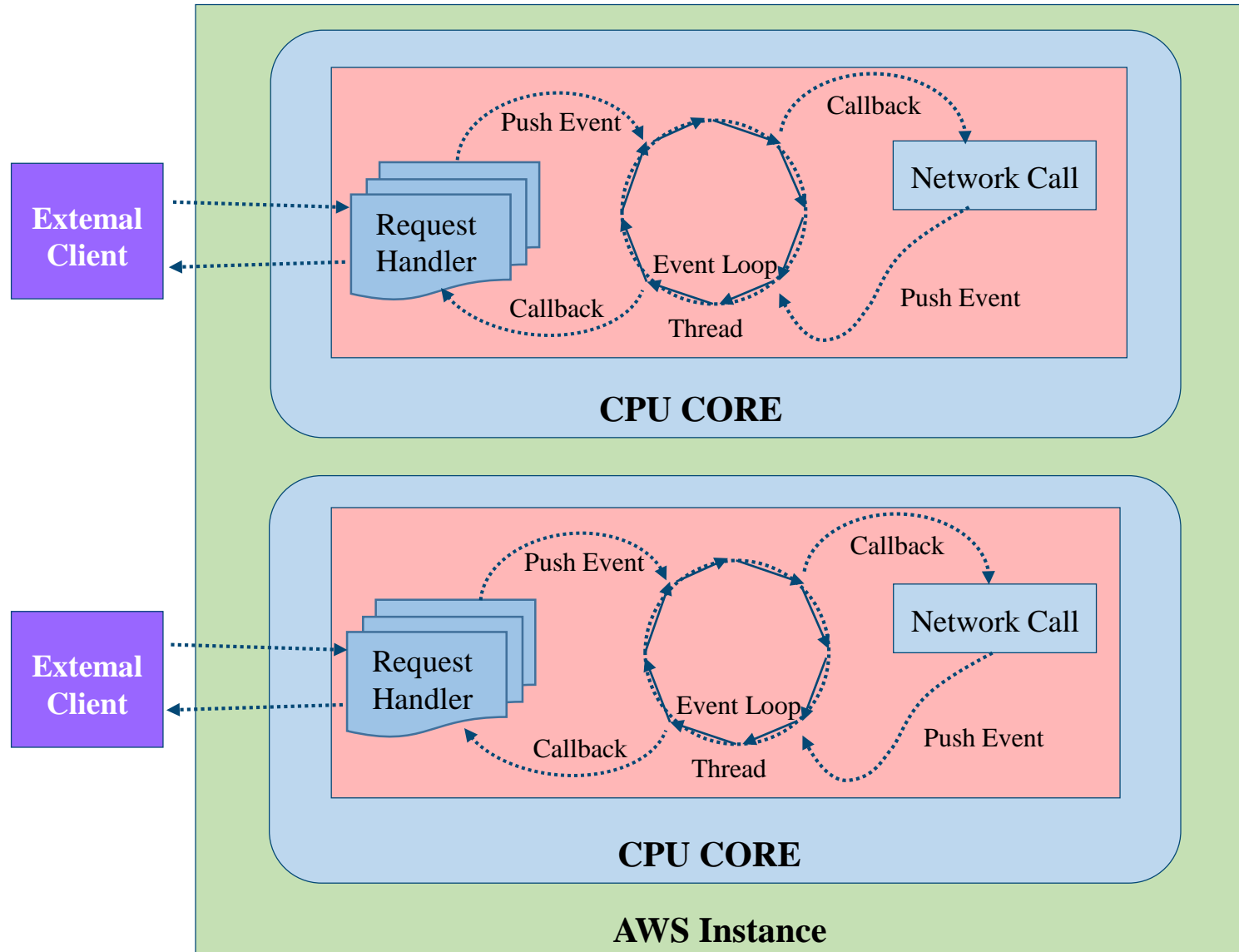
劣

- 线程上下文切换开销
- 连接数限制
- 延迟阻塞耗尽线程连接资源

VS

适用: 计算密集型(CPU bound)场景

Zuul 2 ~ 非阻塞异步模式



非阻塞异步模式优劣

优

- 线程开销少
- 连接数易扩展



劣

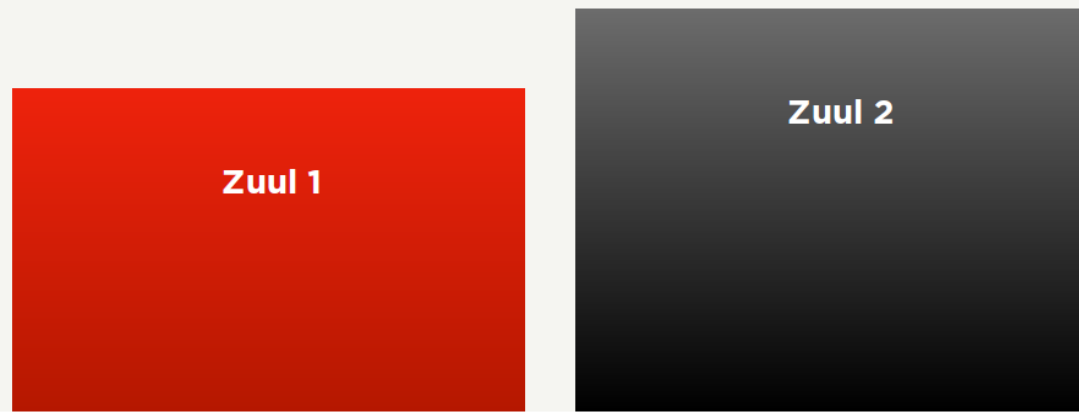
- 编程模型复杂
- 开发调试运维复杂
- ThreadLocak不work

VS

适用: IO密集型(IO bound)场景

性能比对

Real World Performance



Request per second per node

参考

<https://github.com/strangeloop/StrangeLoop2017/blob/master/slides/ArthurGonigberg-ZuulsJourneyToNonBlocking.pdf>

Zuul 2额外功能亮点

HIGHLIGHTS

服务器协议

- HTTP/2
- Mutual TLS

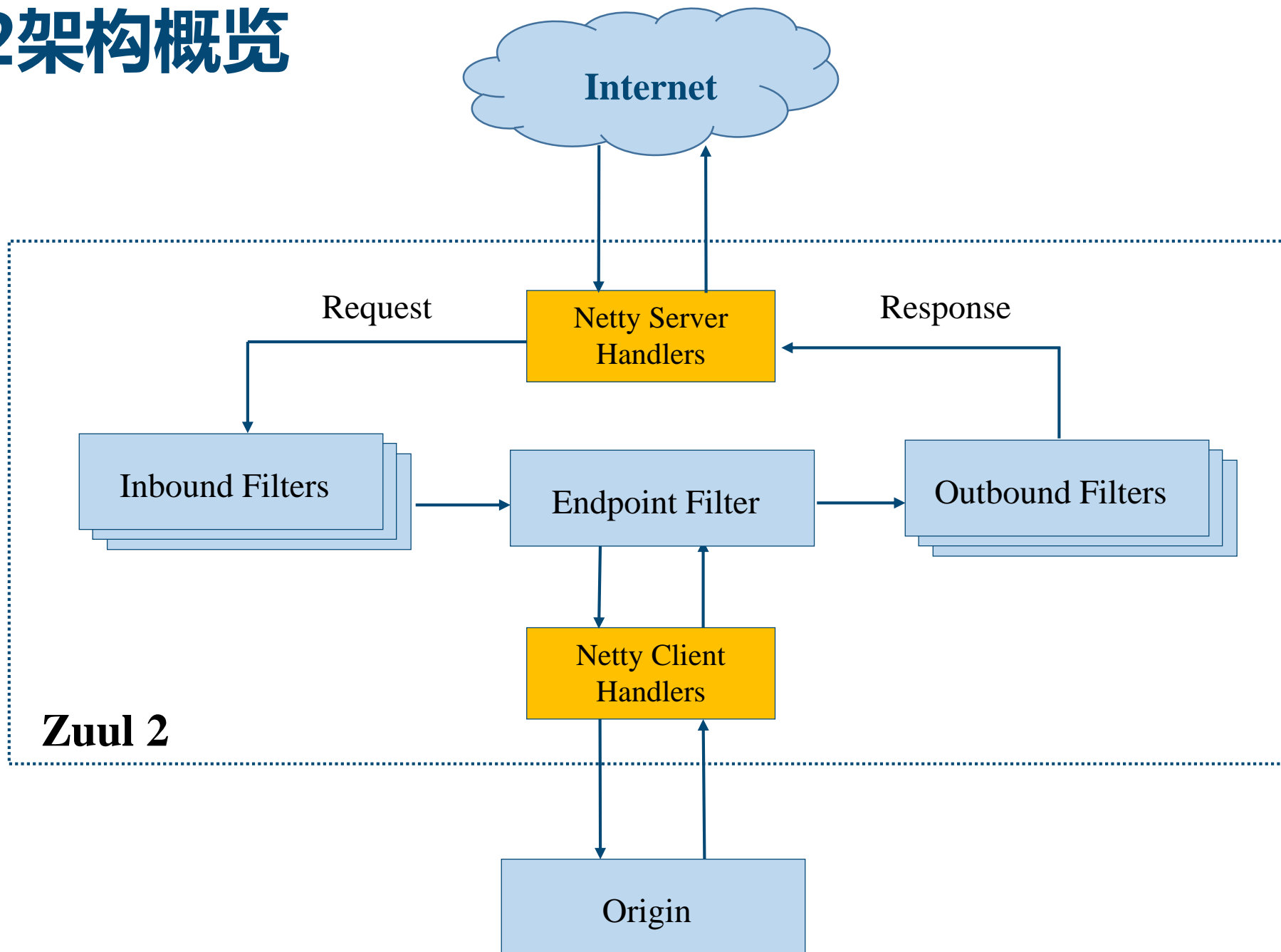
弹性

- Adaptive Retries
(自适应重试)
- Origin Concurrency
Protection(源并发保
护)

运维

- Request Passport
- Status Categories
(状态码分类)
- Request Attempts

Zuul 2架构概览



个人建议

01

生产使用Zuul1.0

- 同步模型简单
- 容易监控埋点(CAT)
- 稳定成熟
- 大部分公司量级OK
- 可以集成Hystrix熔断
- 可以使用AsyncServlet优化

关注和试验Zuul2.0

02

第

16

部分

Zuul网关生产最佳实践

Zuul网关最佳实践



异步AsyncServlet优化连接数

Apollo配置中心集成动态配置

Hystrix熔断限流

- 信号量隔离

连接池管理

CAT和Hystrix监控

过滤器调试技巧

网关无业务逻辑

自助路由(需定制扩展)

第

17

部分

参考资源和后续课程预览

参考文章

Announcing zuul edge service in the cloud

<https://medium.com/netflix-techblog/announcing-zuul-edge-service-in-the-cloud-ab3af5be08ee>

Zuul 2 : The Netflix Journey to Asynchronous, Non-blocking Systems

<https://medium.com/netflix-techblog/zuul-2-the-netflix-journey-to-asynchronous-non-blocking-systems-45947377fb5c>

Open Sourcing Zuul 2

<https://medium.com/netflix-techblog/open-sourcing-zuul-2-82ea476cb2b3>

Spring Cloud Zuul

http://cloud.spring.io/spring-cloud-netflix/1.4.x/multi/multi__router_and_filter_zuul.html

Netflix's Global Edge Architecture

- <https://www.slideshare.net/MikeyCohen1/edge-architecture-ieee-international-conference-on-cloud-engineering-32240>

Zuul @ Netflix

- <https://www.slideshare.net/MikeyCohen1/zuul-netflix-springone-platform>

Zuul's Journey to Non-Blocking

- <https://github.com/strangeloop/StrangeLoop2017/blob/master/slides/ArthurGonigberg-ZuulsJourneyToNonBlocking.pdf>

源码

<https://github.com/Netflix/zuul>

Netflix Zuul

<https://github.com/spring2go/s2g-zuul>

Spring2go定制版Zuul

其它开源网关产品



Kong(核心开源)

<https://github.com/Kong/kong>



Tyk(核心开源)

<https://github.com/TykTechnologies/tyk>



悟空API网关(部分开源+商业支持)

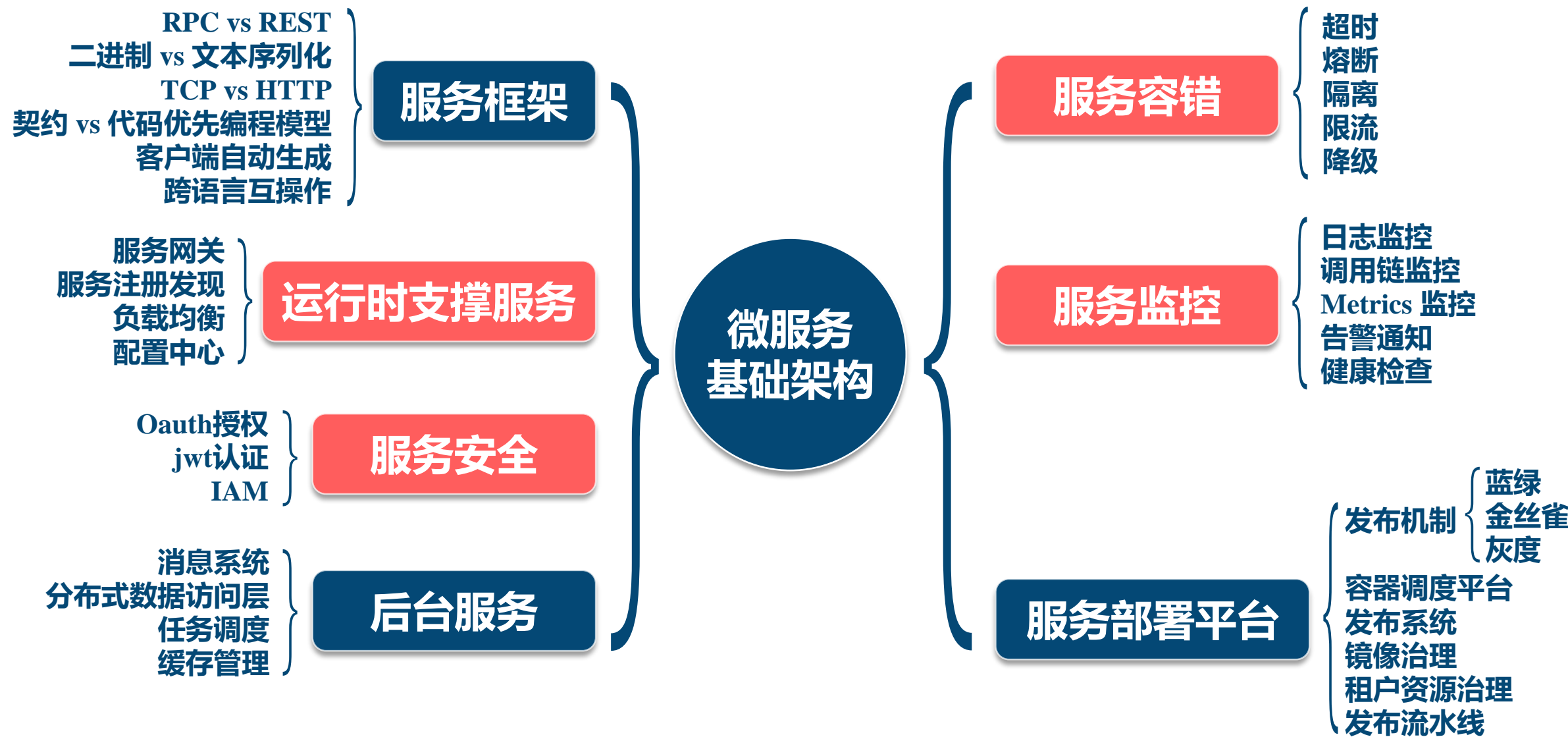
<https://github.com/eolinker/GoKu-API-Gateway>



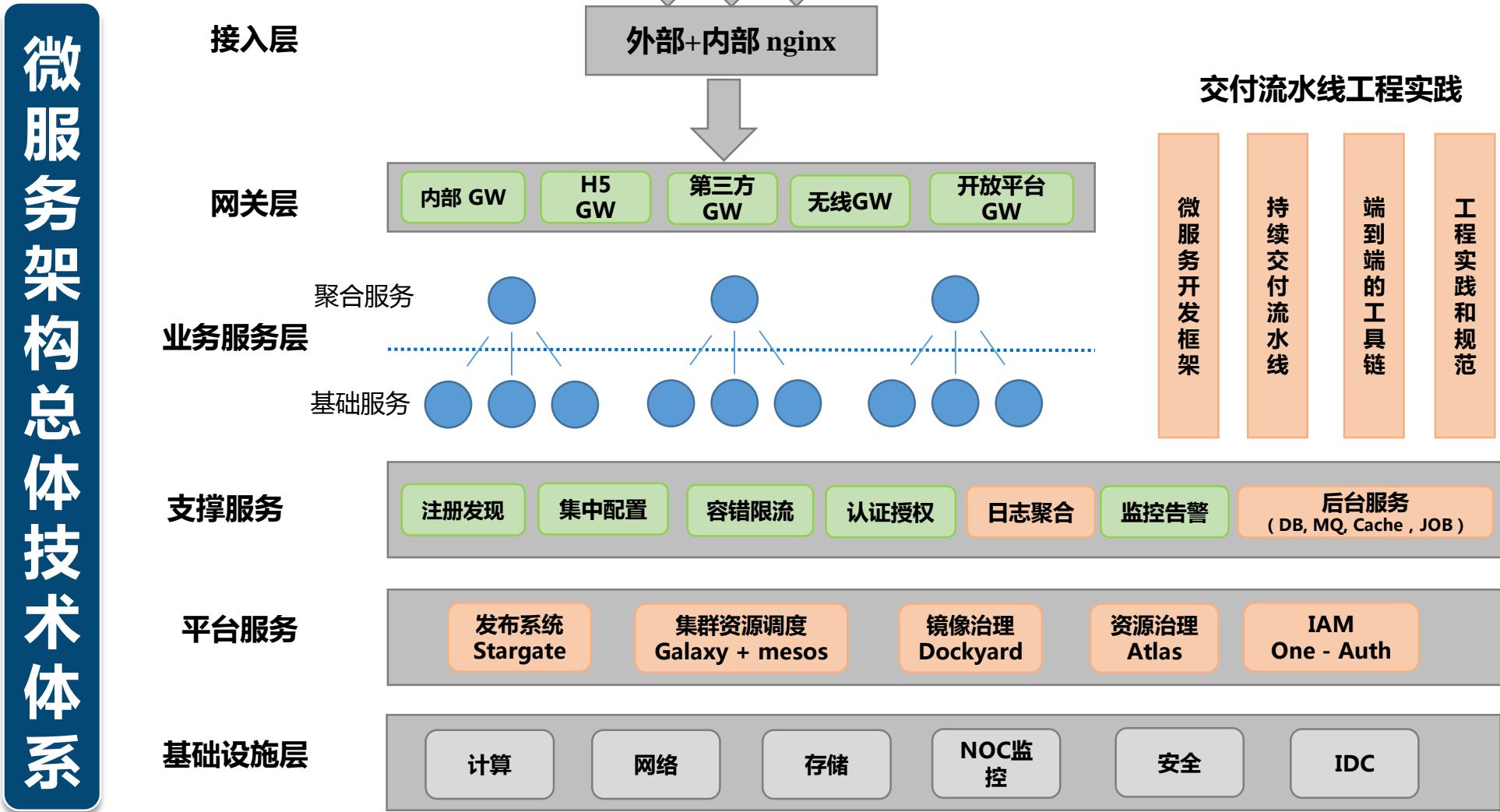
小豹API网关(商业)

<http://www.xbgateway.com/>

后续课程预览~2018课程模块



后续课程预览~技术体系



架构和技术栈预览

