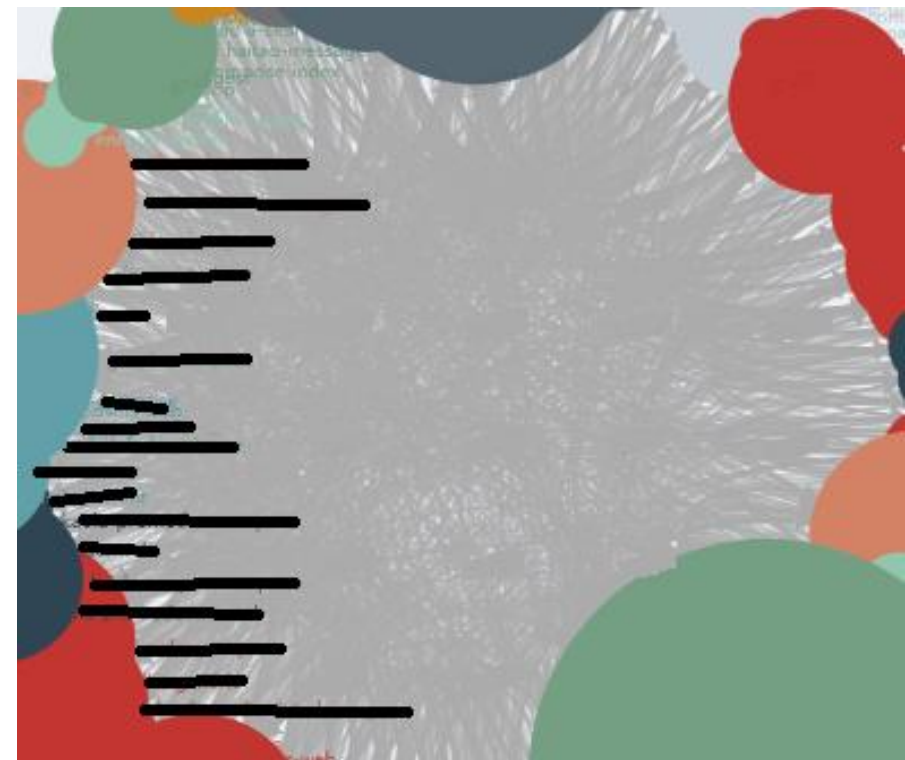


微服务治理中心设计与实践

刘超 网易研究院云计算技术部首席架构师

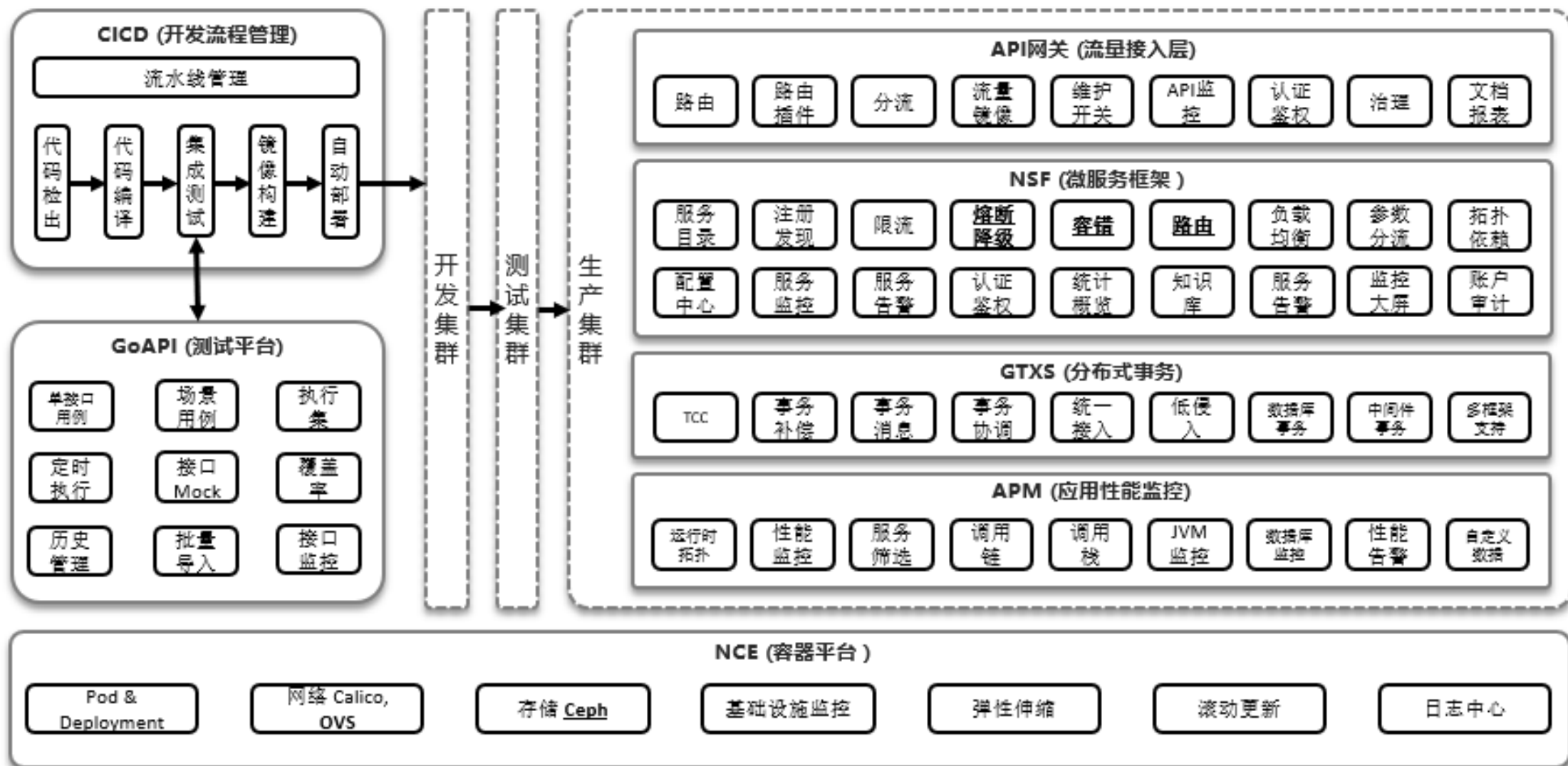
为什么说微服务是一件复杂的事情

考拉服务治理平台——密密麻麻的调用关系

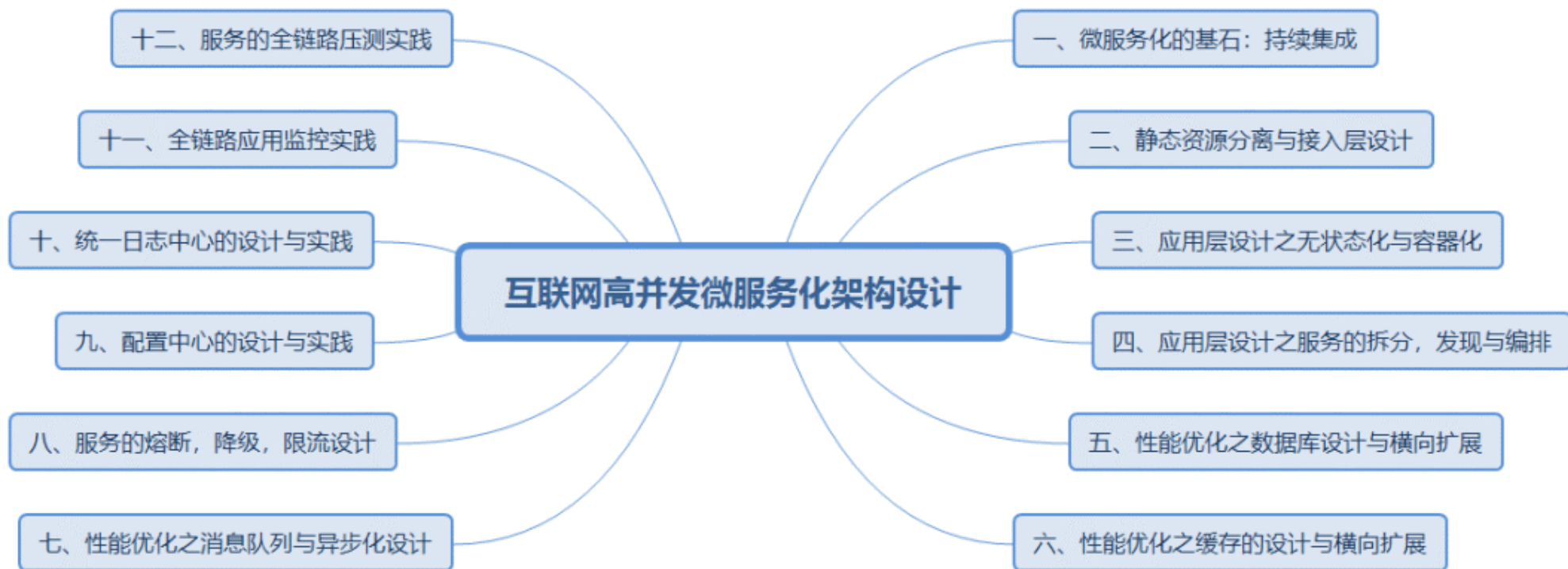


服务调用关系

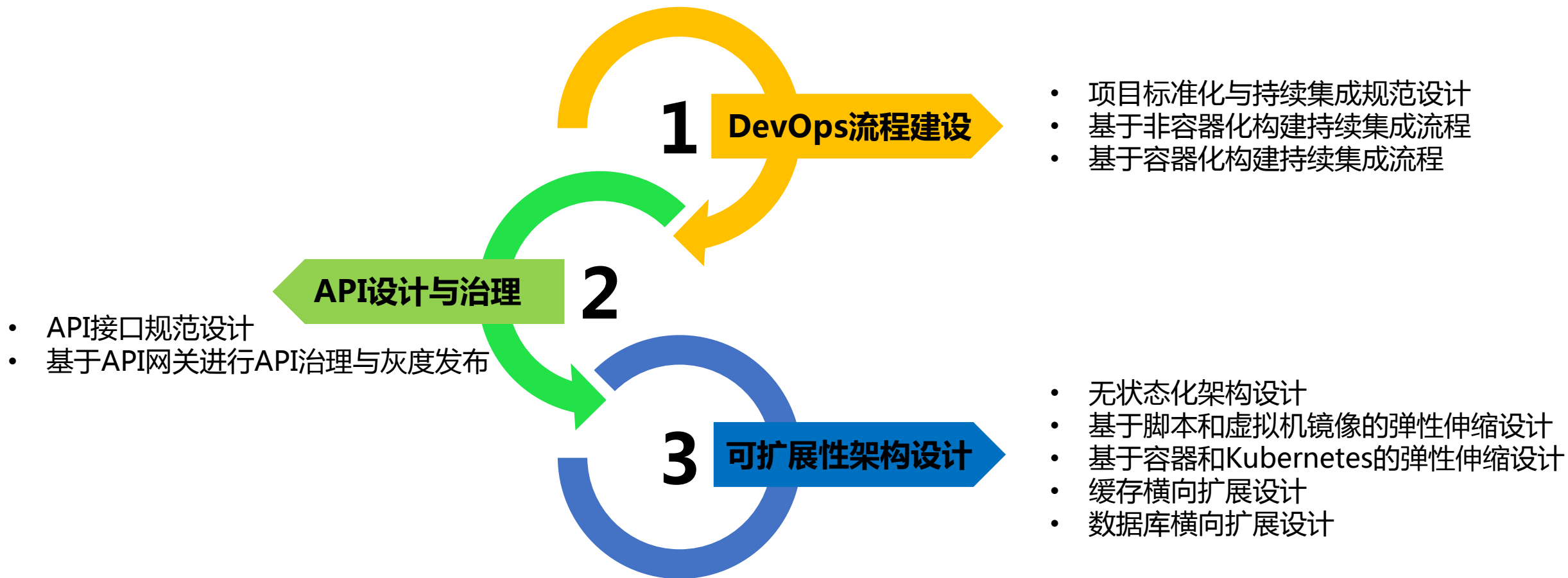
七大系统



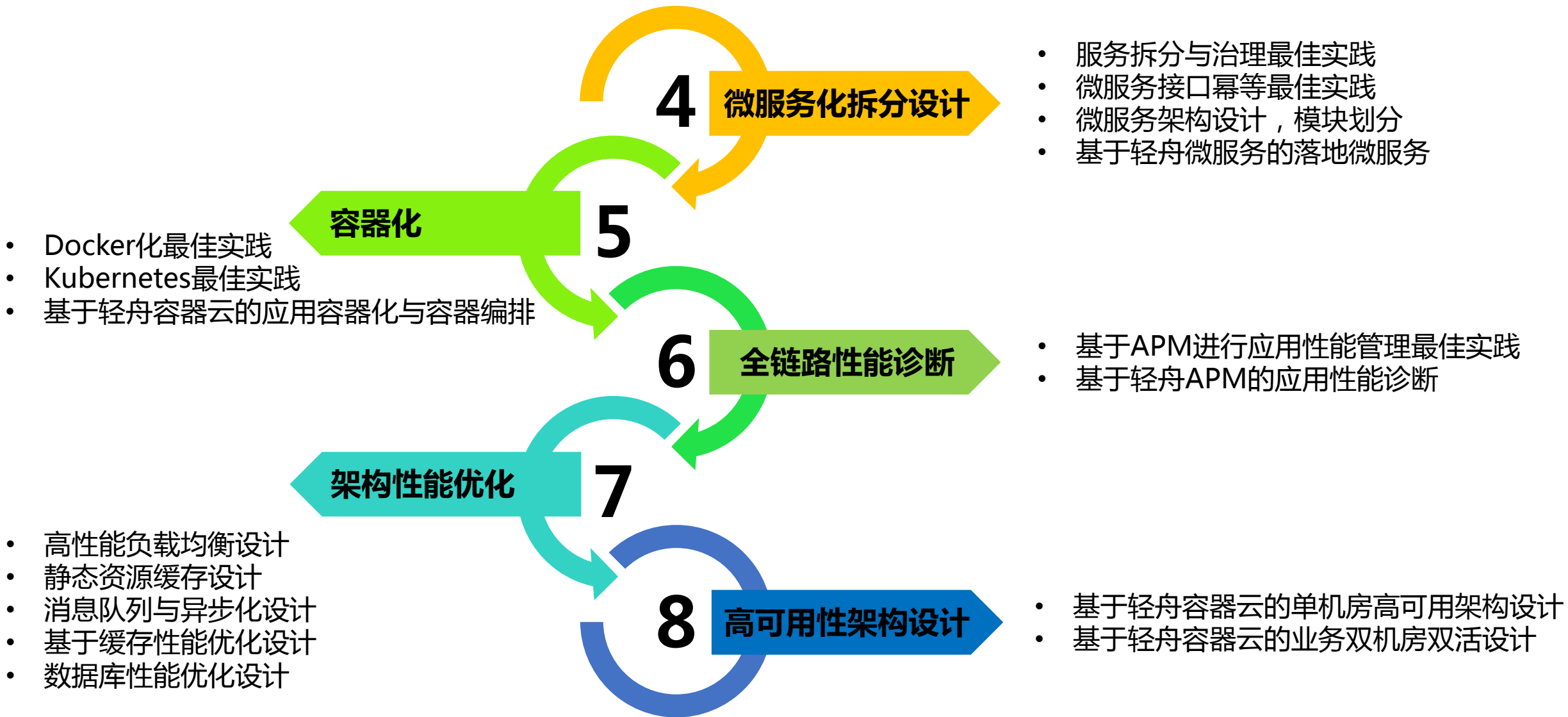
十二个过程



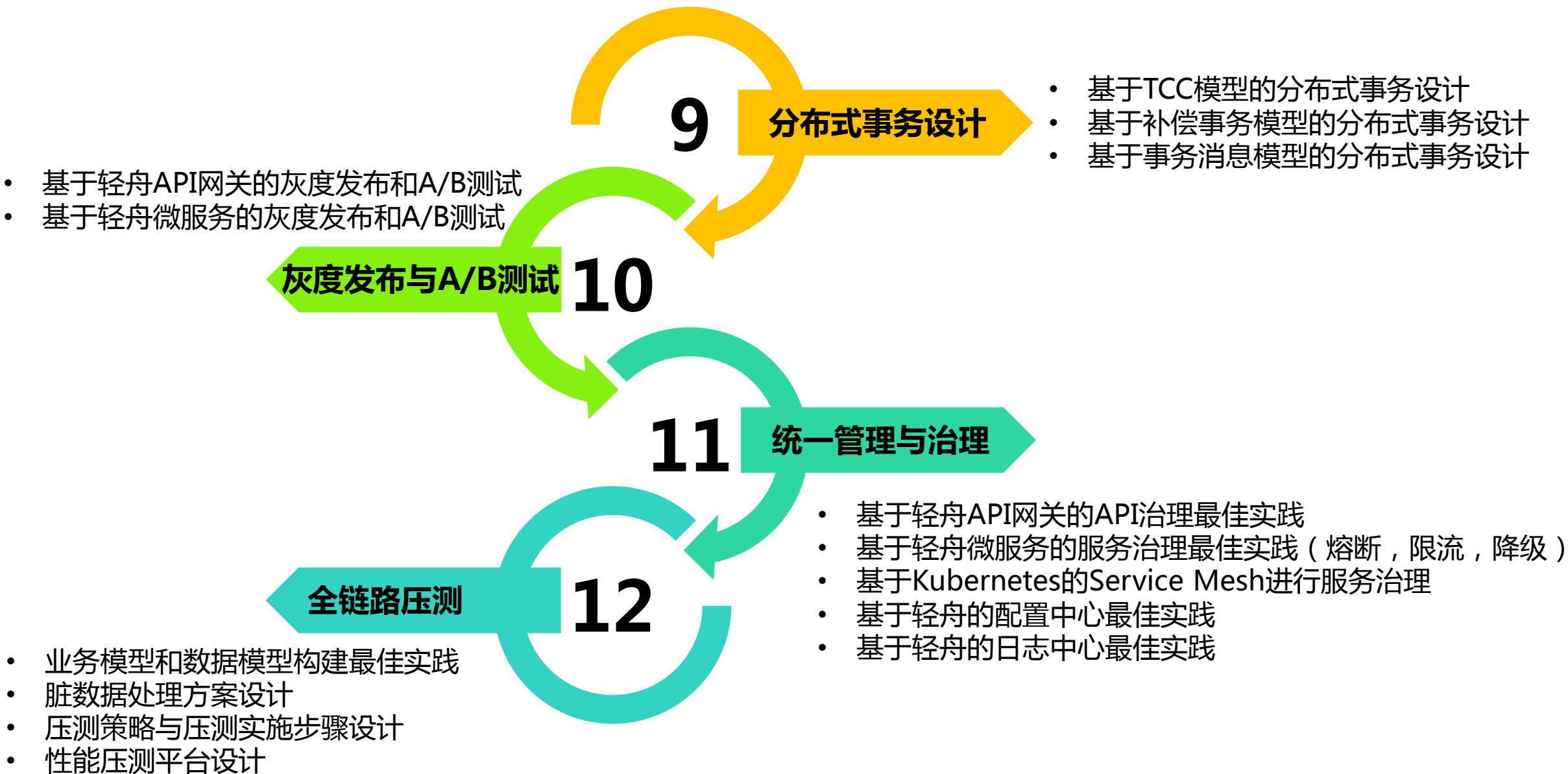
十二个过程：初创期



十二个过程：成长期



十二个过程：成熟期

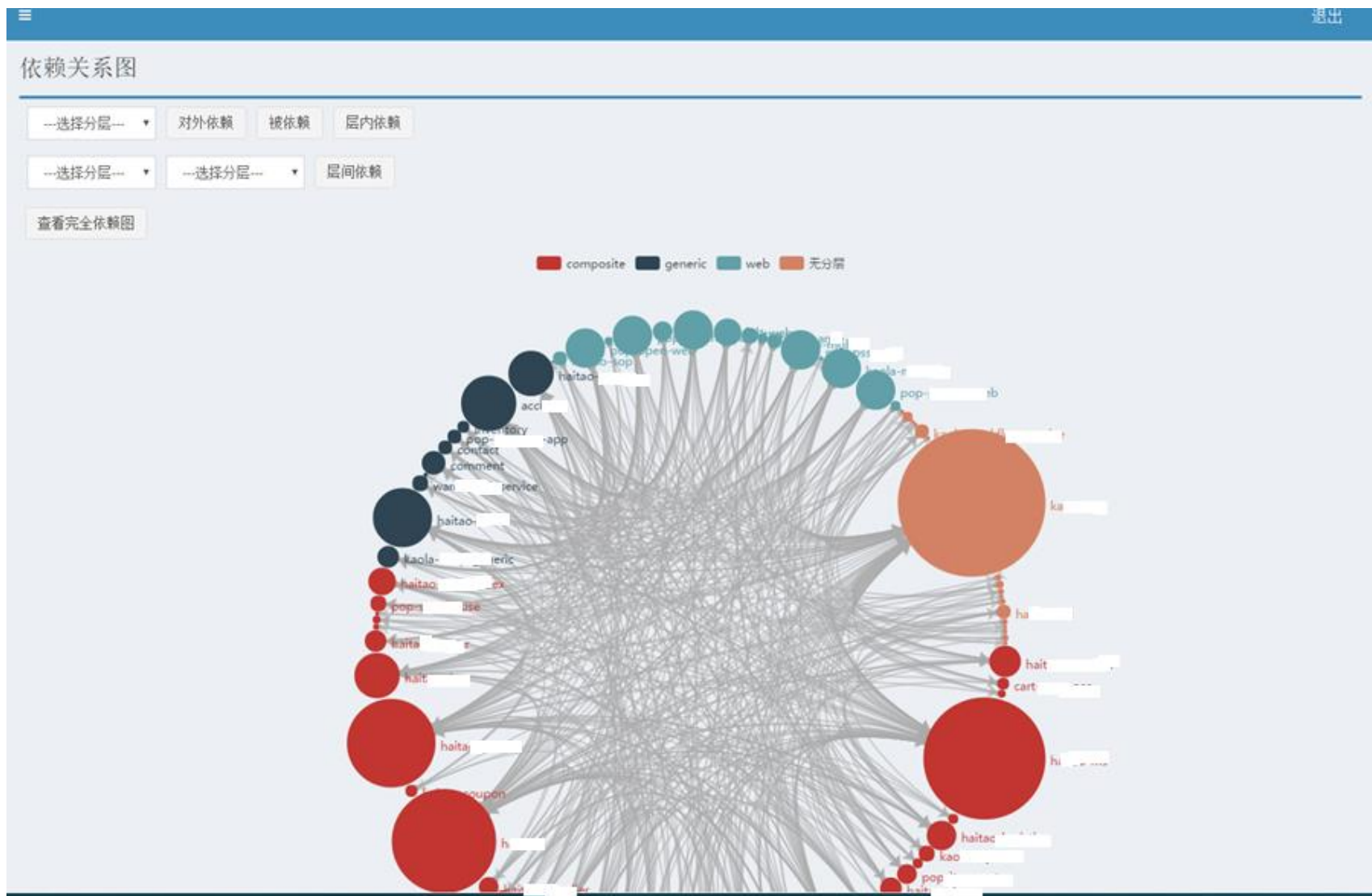


服务治理中心的设计

■ 服务治理中心应该解决的问题

- 服务依赖管理：服务间直接调用，依赖混乱
- 服务调用统计：调用记录无迹可寻，调用统计与分析无从谈起
- 服务接口规范：环境与接口规范缺失，维护困难
- 服务安全管理：安全靠白名单各自为战
- 服务治理能力：大量重复代码实现路由，分流，熔断，降级
- 服务接口测试：拆分过程中接口行为不一致，隐藏Bug
- 服务灰度发布：上线功能实现灰度借助大量if-else
- 服务压力测试：对于峰值压力无历史数据，靠运气
- 服务调用链分析：当服务请求缓慢，难以定位问题点

服务依赖管理



服务调用统计



服务接口规范



认证鉴权 注册，发现，调用都提供鉴权

知识库 接口文档统一维护
文档与运行时一致
减少调用沟通成本

账户审计 根据平台、租户、项目三个层次区分权限作用域
操作记录，审计日志，事件查询

服务安全管理



路由 可配置多条规则，按优先级匹配

可配置消费端黑白名单：只有A服务能访问B服务，只有IP1能访问B服务，更加安全

可配置服务端黑白名单：A服务版本1访问B服务版本1，A服务版本2访问B服务版本2，更灵活

服务治理能力



熔断

粒度更细：可指定服务版本，类，方法级别

配置灵活：可配置检测粒度为每M毫秒N个请求P%的错误率

指标多样：RT值，错误率，线程池参数

容错

粒度更细：可指定调用者和被调用者服务版本，支持failover、failfast、failback容错机制。

配置灵活：支持自定义超时时间和重试次数。

可自行定制：通过暴露自定义异常NSFExcetion解决任意业务方法的容错，支持超时、failover，failfast容错。

服务治理能力



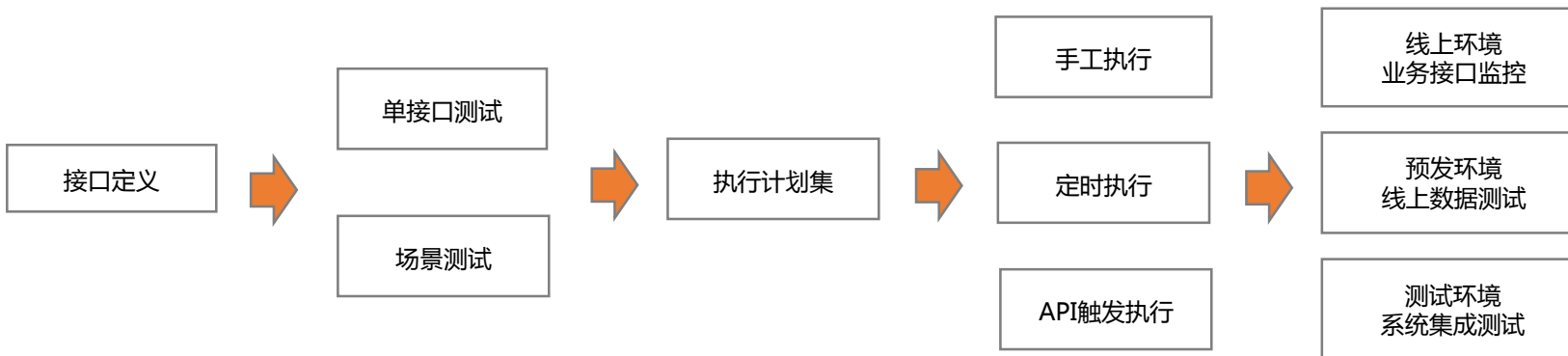
参数分流

负载均衡规则的高级补充，可以通过参数判断，决定请求流量的流向。
 参数取模时，可针对 Cookie、HTTP Header 或 Query String 的参数进行取模运算。
 名单分流时，名单支持正则表达式。

A用户永远只访问A服务v1

VIP用户访问A服务V2，非VIP用户访问A服务V1

服务接口测试



杭州空气质量

complete 04/11 19:24

执行人: 张明

执行环境: 测试环境

校验结果: ✓

GET http://api.help.bj.cn/apis/aqi3/?id=hangzhou

请求 响应 断言

耗时: 345 ms

HEADERS

BODY

▼ "{...}":

"city": "杭州"

"level": "三级 (轻度污染)"

"update": "2018-04-11 18:00:00"

"aqi": "106"

"pm25": "79"

"pm10": "154"

"co": "1.01"

"no2": "51"

"o31h": "175"

"o38h": "154"

"so2": "16"

▶ "data": 11 items

Step1 验证body的参数类型

Step2 验证URL的参数类型

Step3 暂停 等待 3 秒

Step4 数组场景校验

Step5 暂停 等待 1 秒

Step6 验证URL的参数类型

Step7 暂停 等待 3 秒

Step8 验证post url参数获取

Step9 验证body的参数类型

Step10 数组场景校验

Step11 暂停 等待 6 秒

用例

等待

场景

【大回归】POP大回归执行集

单用例/场景用例/覆盖接口: 256 / 7 / 194

类型: 测试回归 创建人: 葛庆阳

描述:

执行记录: [Progress Bar]

【大回归】【商品】商品大回归集

单用例/场景用例/覆盖接口: 32 / 2 / 23

类型: 测试回归 创建人: 陈伟

描述: 商品大回归集

执行记录: [Progress Bar]

【促销】优惠券后台

单用例/场景用例/覆盖接口: 0 / 27 / 5

类型: 测试回归 创建人: 丁鹏

描述:

执行记录: [Progress Bar]

【POP】合作伙伴系统回归用例集

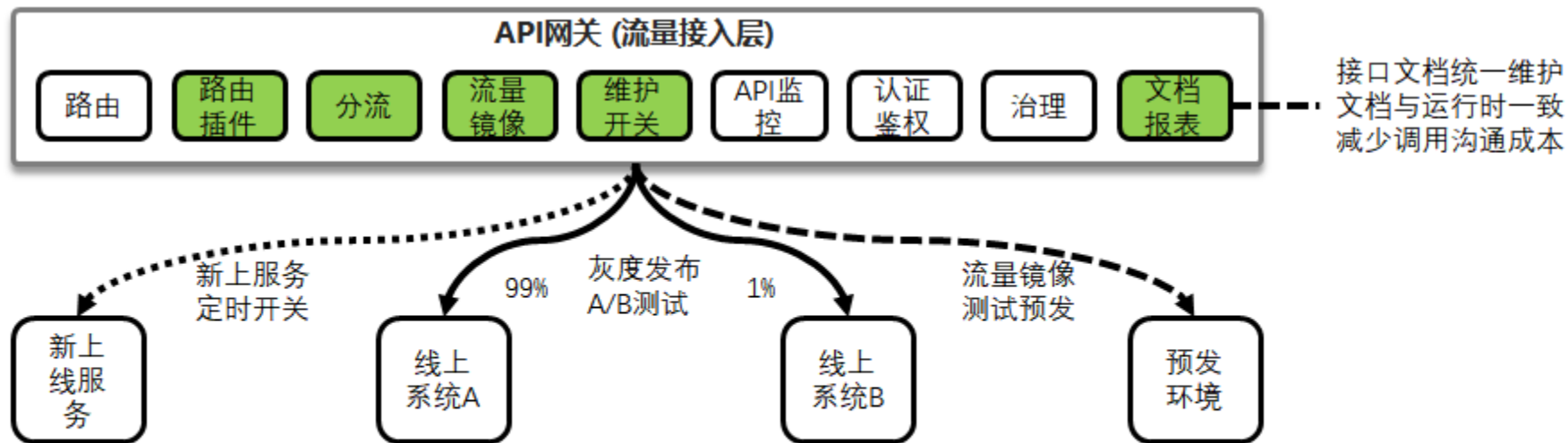
单用例/场景用例/覆盖接口: 1 / 0 / 1

类型: 测试回归 创建人: 葛庆阳

描述:

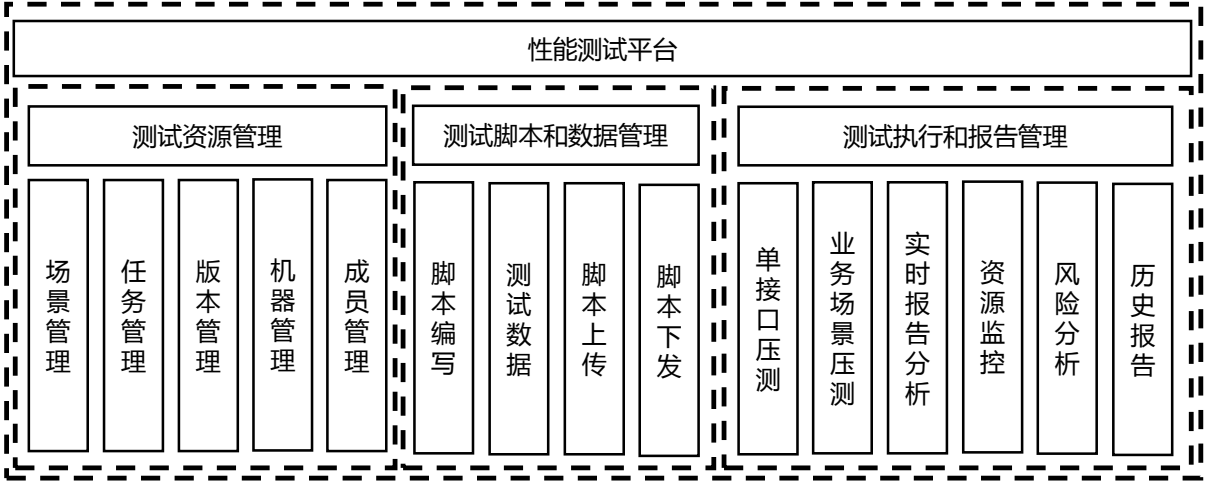
执行记录: [Progress Bar]

服务灰度发布



可自行定制：路由插件，可开发插件拦截请求，进行定制化

服务压力测试



容量测试

• 采用梯度压力，看服务的性能变化情况，评估出服务的最大容量值。

摸高压测

• 在达到停止条件之后，继续增加压力，检验服务集群在失效状态下的表现。

峰值稳定性测试

• 在峰值压力下，保持30分钟（可讨论）稳定

秒杀场景测试

• 针对秒杀类业务，制定秒杀测试场景

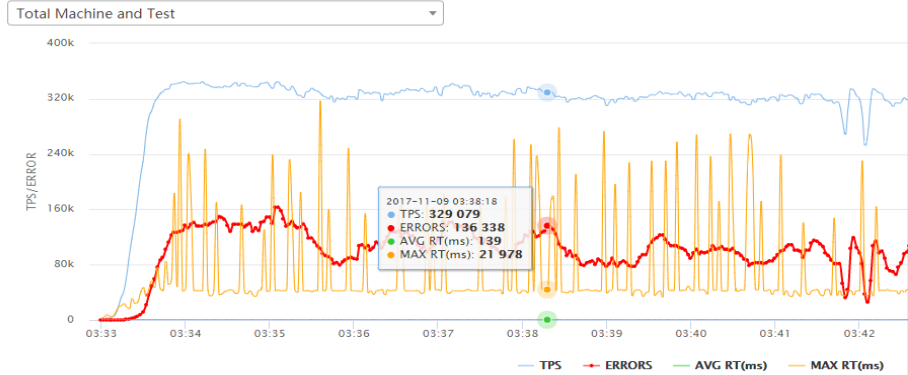
限流演练

• 多级限流，保护系统稳定提供服务

降级演练

• 非核心业务降级，提升整体服务能力

TPS-RT曲线

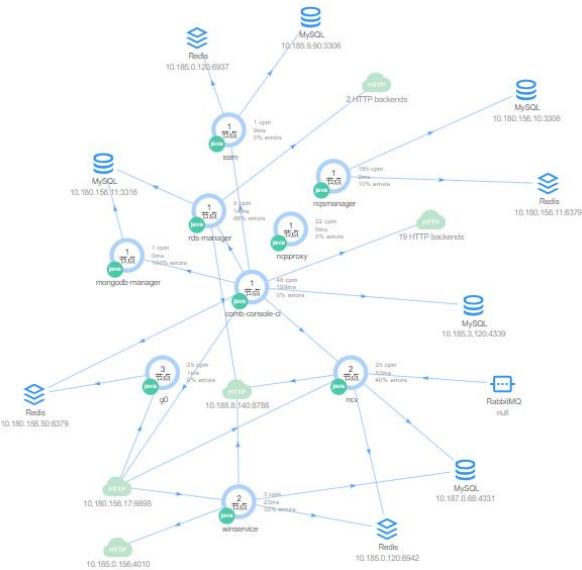


HTTP响应字节数（KB/s）曲线

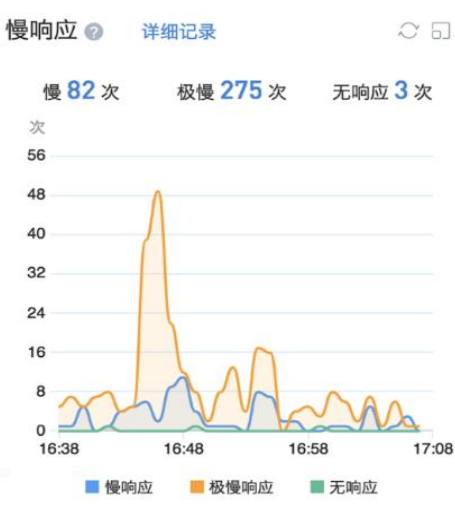


服务调用链分析

服务拓扑



异常检测



请求快照定位问题

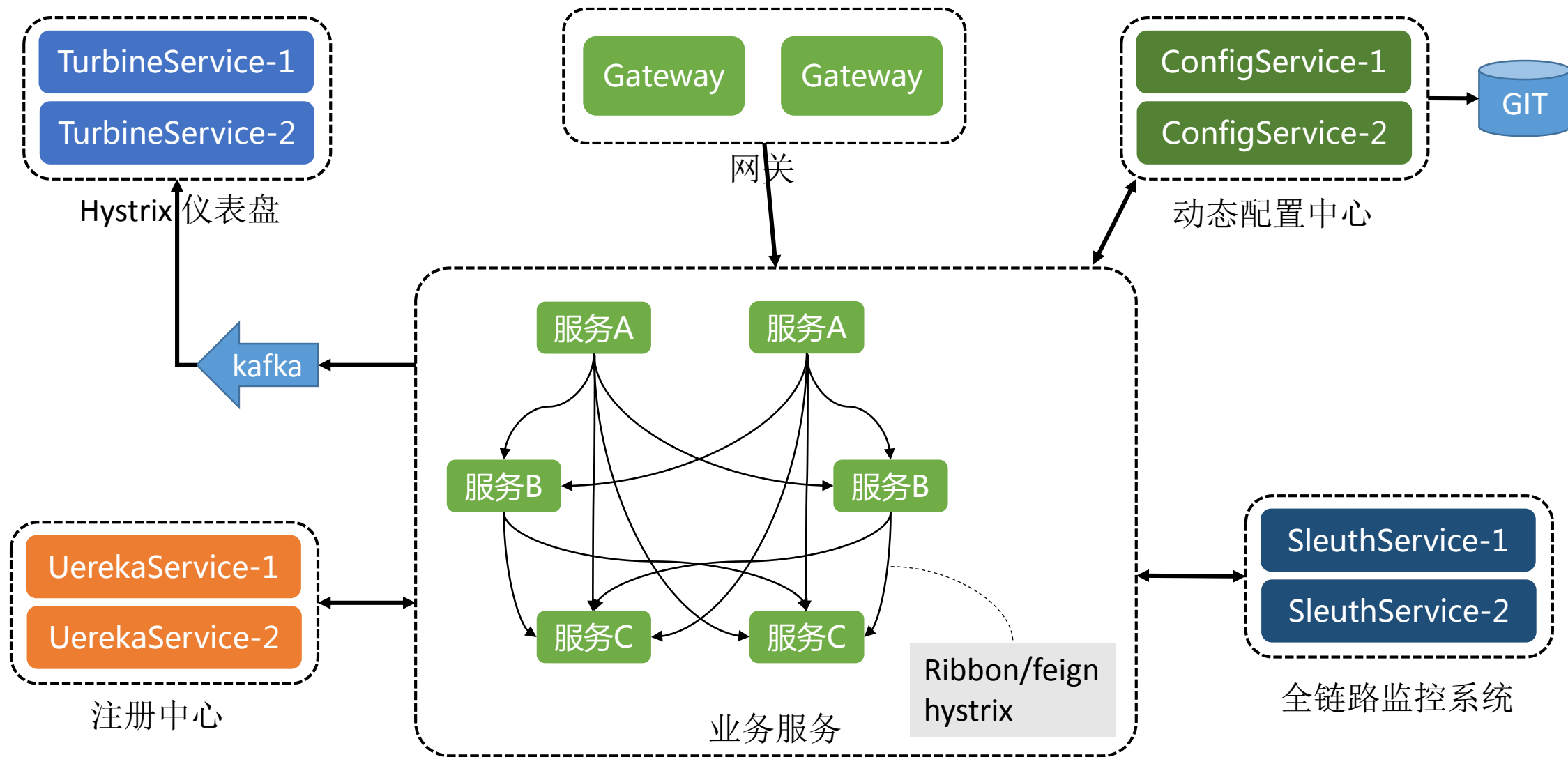


查看调用链

时间	类型	请求	运行时间	服务	节点	业务ID	操作
2018-09-25 11:54:59	极慢响应	http://10.18.192.75:8080/javademo/mockpath	4.61s	two	java-agent-two	-	查看调用链 查看日志
2018-09-25 11:54:56	错误	http://10.18.192.17:8080/javademo/memcache...	6ms	one	java-agent	-	查看调用链 查看日志
2018-09-25 11:46:34	错误	http://10.18.192.17:8080/javademo/memcache...	82ms	one	java-agent	-	查看调用链 查看日志
2018-09-25 11:46:29	极慢响应	http://10.18.192.17:8080/javademo/memcache...	5.24s	one	java-agent	-	查看调用链 查看日志

服务治理中心实践

基于开源Spring Cloud的微服务架构



既然是开源，为什么不自己做？（微服务比想象的复杂的多）



微服务组件的使用成本

服务发现(Eureka)

1. 添加 eureka 依赖

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-eureka</artifactId>
</dependency>
```

2. 添加 @EnableEurekaClient 注解

```
@EnableEurekaClient
public class Application
```

3. 在 application.yml 或者 application.properties 中添加配置

```
eureka:
  instance:
    leaseRenewalIntervalInSeconds: 1
    leaseExpirationDurationInSeconds: 2
  client:
    serviceUrl:
      defaultZone: http://127.0.0.1:8761/eureka/
```

错误容忍(Hystrix)

```
@Configuration
public class HystrixConfiguration {

    @Bean
    public HystrixCommandAspect hystrixAspect() {
        return new HystrixCommandAspect();
    }
}

public class UserService {
    ...
    @HystrixCommand
    public User getUserById(String id) {
        return userResource.getUserById(id);
    }
    ...
}
```

负载均衡(Ribbon)增强示例

```
@Configuration
public class Config {
    @LoadBalanced
    @Bean
    public RestTemplate restTemplate() {
        return new RestTemplate();
    }
}
```

```
@Autowired
private RestTemplate restTemplate;

public MessageWrapper<Customer> getCustomer(int id) {
    Customer customer = restTemplate.exchange( "http://customer-service/customer/{id}", H
    return new MessageWrapper<>(customer, "server called using eureka with rest template"
}
```


The diagram illustrates the GRPC architecture, divided into two main planes: the **数据面 (Data Plane)** and the **控制面 (Control Plane)**.

数据面 (Data Plane):

- 应用服务 (Application Service):** Consists of **NSF Agent** and **Envoy** components.
- 调用 (Invocation):** Shows requests being sent from the application service to the NSF Agent and Envoy.
- 其他外部服务 (Other External Services):** Represented by a cloud icon, showing external service interactions.
- GRPC双向通信 (GRPC Bidirectional Communication):** Indicated by green arrows between the NSF Agent and Envoy.
- 上报服务信息 (Report Service Information):** Indicated by a green arrow from the NSF Agent to the Service Registration Center.
- 服务注册与服务发现 (Service Registration and Discovery):** Indicated by a blue dashed arrow from the Service Registration Center to the NSF Agent.
- 下发治理策略 (Distribute Governance Policy):** Indicated by a green arrow from the NSF Server to the NSF Agent.

控制面 (Control Plane):

- 服务注册中心 (Service Registration Center):** A blue box responsible for service registration and discovery.
- 服务控制中心 (NSF Server):** A blue box responsible for service governance and policy distribution.
- ETCD:** A purple box representing the distributed key-value store used for configuration synchronization.
- 多节点高可用 (Multi-node High Availability):** Indicated by multiple stacked boxes for the Service Registration Center, NSF Server, and ETCD.
- 多实例配置同步 (Multi-instance Configuration Synchronization):** Indicated by a double-headed arrow between the NSF Server and ETCD.

NSF Agent Components:

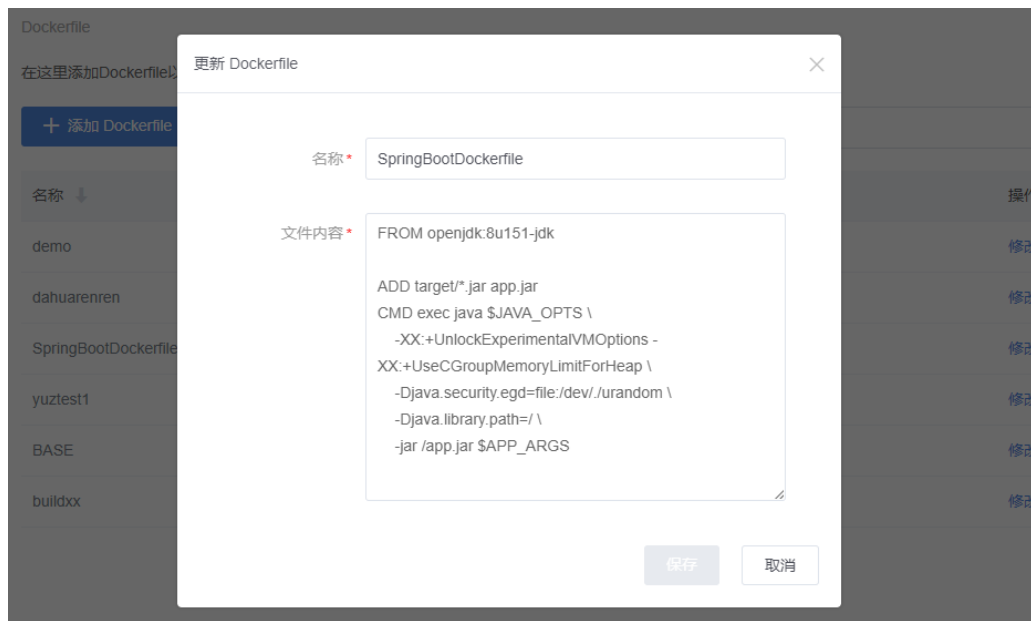
- 熔断模块 (hystrix):** Circuit breaker module.
- 流控模块 (Flow Control Module):** Flow control module.
- 服务注册 (Eureka client):** Service registration module.
- 负载均衡 (Ribbon):** Load balancing module.
- Java Assist:** Java assistance module.
- 统计 (Statistics):** Statistics module.
- 监控 (Monitoring):** Monitoring module.

- 以微服务架构为设计目标，同样解决服务治理难的痛点
- 业务代码与服务治理的解耦开发和运维
- 数据面代理设计，无侵入方式对应用进行改造
- 提供统一的控制面，开箱即用
- 兼容多数据平面



微服务框架使用实践

与Kubernetes 如何优雅集成？



```
apiVersion: extensions/v1beta1
```

```
kind: Deployment
```

```
metadata:
```

```
  labels:
```

```
    nce-app: nsf-stock-viewer
```

```
  name: nsf-stock-viewer
```

```
  namespace: nsf
```

```
spec:
```

```
  template:
```

```
    metadata:
```

```
      labels:
```

```
        nce-app: nsf-stock-viewer
```

```
  containers:
```

```
    - name: tomcat
```

```
      env:
```

```
        - name: NCE_PORT
```

```
          value: "8790"
```

```
        - name: NCE_JAVA_OPTS
```

```
          value: "-javaagent:/home/nsf-agent/nsf-agent.jar=stock-viewer"
```

```
      image: hub.c.163.com/qingzhou/stock-viewer:online-20180813-193644
```

```
      imagePullPolicy: IfNotPresent
```

```
      volumeMounts:
```

```
        - mountPath: /home/nsf-agent
```

```
          name: nsf-agent
```

```
      dnsPolicy: ClusterFirst
```

```
      hostNetwork: true
```

```
      restartPolicy: Always
```

```
      volumes:
```

```
        - hostPath:
```

```
          path: /opt/ncf/nsf-demo-stock/nsf-stock-viewer/nsf-javaagent-home
```

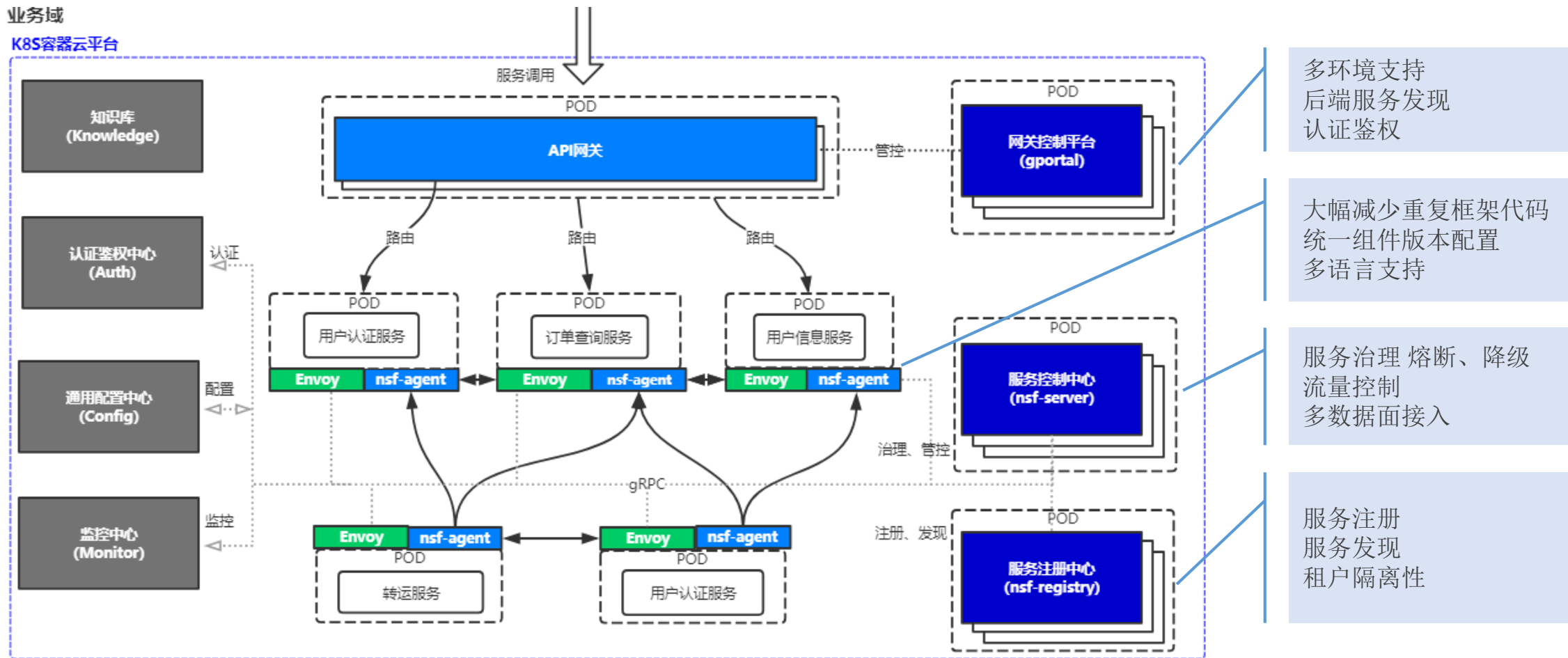
```
          type: ""
```

```
        name: nsf-agent
```



It just works!

微服务应用整体架构

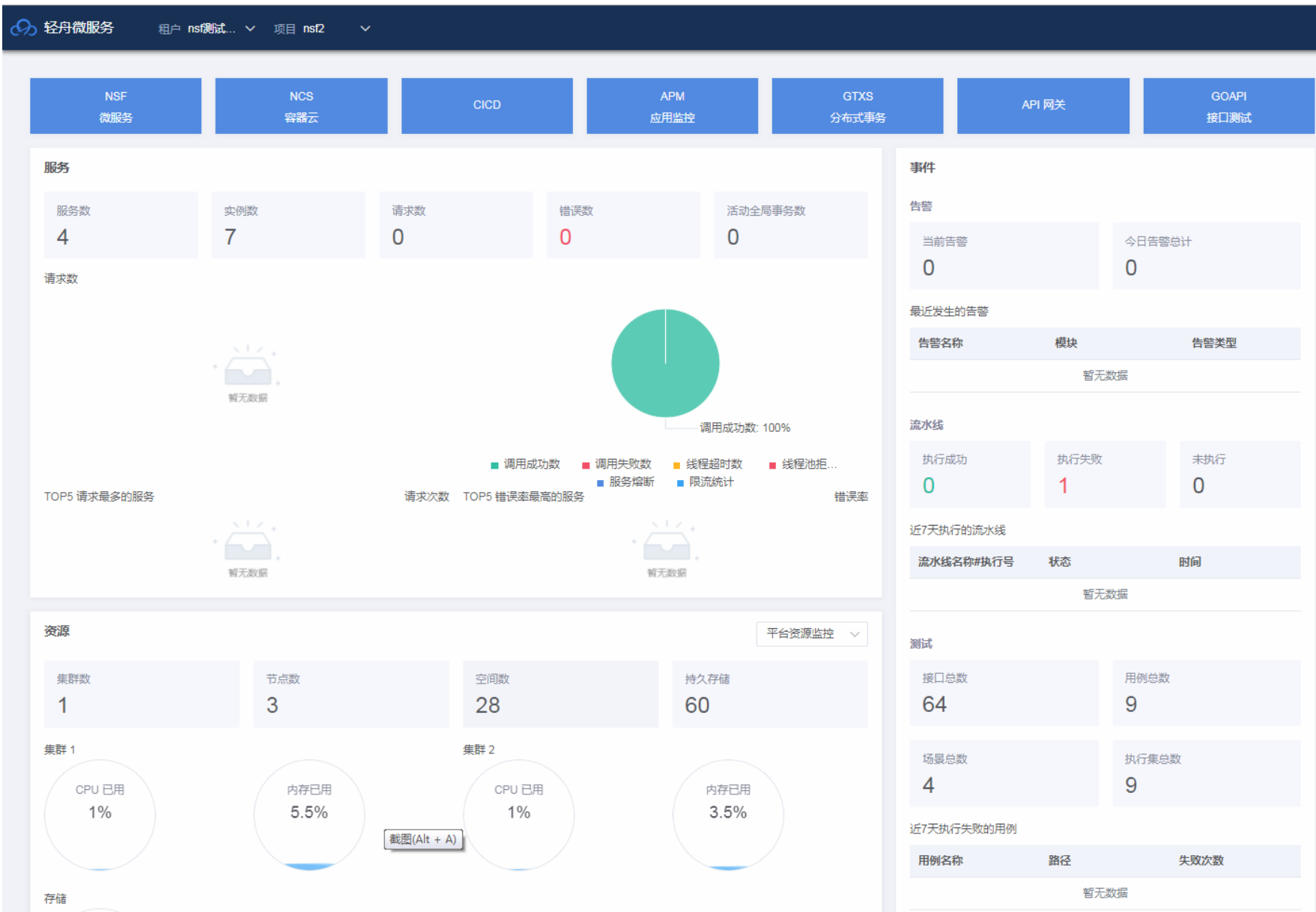


解决了哪些问题？



- + 应用减负：通过Agent 和Sidecar 技术，对应用无成本增强
- + 开发减负：以微服务治理框架为设计目标，大幅减少重复框架代码，避免重复造轮子；
- + 版本控制：统一组件版本配置，避免隐性问题
- + 兼容性：兼容的HTTP、RPC调用。兼容非java应用
- + 服务治理：根据业务线场景选择治理支持方法级别治理粒度
- 高性能：更低性能损耗，并提供更细粒度的服务治理；

轻舟微服务平台



服务依赖管理

轻舟微服务 · NSF

项目 nsf2

环境 test

当前环境 test

应用 全部

Q 请输入名称搜索

sync-test

nsf-demo-stock-viewer

nsf-demo-stock-advisor

nsf-demo-stock-provider

NS

nsf-demo-stock-viewer 已上线

创建时间 2019-01-23 21:35:00

负责人 qatest

可访问类型 认证关闭

更多

描述 -

依赖关系

监控

服务实例

服务治理

流量控制

服务配置

访问权限

提供者服务

服务名称	所属应用	负责人	最近调用时间
nsf-demo-stock-advisor	stock	qatest	2019-01-31 16:21:30 519
nsf-demo-stock-provider	stock	admin	2019-01-31 15:07:41 940

消费者服务

服务名称	所属应用	负责人	最近调用时间
暂无数据			

服务调用统计



服务接口规范

轻舟微服务 · NSF

项目 nsf2

环境 test

超级管理员

当前环境 test

创建服务

NS

nsf-demo-stock-viewer

创建时间 2019-01-23 21:34:49

最近更新时间 2019-01-23 21:34:49

所属应用 stock

负责人 qatest

描述 -

更多

接口信息

发布信息

+ 创建 API

请输入搜索关键字

接口信息	备注	创建时间	修改时间	操作
<div>RESTFUL DELETE</div> <div>删除_接口-1</div> <div>/deleteApi/{id}</div>	-	2019-01-26 13:35:58	2019-01-26 13:36:47	<div>查看文档</div> <div>编辑</div> <div>删除</div>
<div>RESTFUL PUT</div> <div>putApi</div> <div>/path1/path2</div>	-	2019-01-26 13:32:59	2019-01-26 13:32:59	<div>查看文档</div> <div>编辑</div> <div>删除</div>
<div>RESTFUL POST</div> <div>postInterface</div> <div>/user</div>	post接口	2019-01-26 13:31:10	2019-01-26 13:32:30	<div>查看文档</div> <div>编辑</div> <div>删除</div>
<div>RESTFUL GET</div> <div>getApi</div> <div>/getApi</div>	测试GET API	2019-01-26 13:26:41	2019-01-26 13:30:39	<div>查看文档</div> <div>编辑</div> <div>删除</div>

20

条/页

共4项

1

服务管理

服务实例

全局配置

认证管理

全平台

知识库

告警管理

系统管理

开发工具

sync-test

nsf-demo-stock-viewer

nsf-demo-stock-advisor

nsf-demo-stock-provider

test-zbj

test-sync

sssss

www.163yun.com

服务安全管理

轻舟微服务 · NSF

项目 nsf2

环境 test

当前环境 test

应用 全部

请输入名称搜索

sync-test

nsf-demo-stock-viewer

nsf-demo-stock-advisor

nsf-demo-stock-provider

NS nsf-demo-stock-viewer 已上线

创建时间 2019-01-23 21:35:00

可访问类型 认证开启

负责人 qatest

描述 -

依赖关系

监控

服务实例

服务治理

流量控制

服务配置

访问权限

+ 添加授权

名称

NS nsf-demo-stock-viewer 已上线

创建时间 2019-01-23 21:35:00

可访问类型 认证关闭

负责人 qatest

描述 -

依赖关系

监控

服务实例

服务治理

流量控制

服务配置

访问权限

+ 路由规则

负载均衡

路由

参数分流

规则名称 RouterRule-1

当前服务 nsf-demo-stock-viewer

服务版本 全部

策略优先级 0

策略类型 白名单 黑名单

消费端匹配 实例名 project-55:10.182.2.107:nsf-demo-stock-viewer:8790

提供端过滤 服务 nsf-demo-stock-advisor

保存 取消

添加授权

类型 NSF服务 外部服务

选择服务 租户

t_1551093208571

t_1551092071873

t_1551069785664

platform-test

edu-tenant

k8s

chenziyu

gateway

yangjinke

gtxs

hedu

交互访谈专用

服务端测试平台

暂无可选项目

暂无可选服务

添加 取消

服务治理能力

NS

nsf-demo-stock-viewer

已上线

创建时间

2019-01-23 21:35:00

负责人

qatest

可访问类型

认证关闭

更多

依赖关系

监控

服务实例

服务治理

流量控制

服务配置

访问权限

限流

降级

容错

修改容错规则

规则名称

FaultToleranceRule-1

当前服务

nsf-demo-stock-viewer

服务版本

全部

容错范围

nsf-demo-stock-advisor

容错方式

☐ Failover
 ☐ Failfast
 ☐ Failback
 ☒ 自定义

同服务重试次数

5

新服务尝试个数

5

保存

取消

修改限流规则

规则名称

RateLimitRule-1

当前服务

nsf-demo-stock-viewer

规则范围

0.0.1

☐ HttpClientRequestServiceImpl
 ☐ echoObject

☐ RandomServiceImpl
 ☐ getRandomNumber
 ☐ getRanomString

☒ StockServiceImpl
 ☐ getStockList
 ☐ getStockByld

限流维度

qps

阈值

10000000 / 秒

保存

取消

修改降级规则

规则名称

FailRule-1

当前服务

nsf-demo-stock-viewer

规则范围

0.0.1

☐ HttpClientRequestServiceImpl
 ☐ echoObject

☐ RandomServiceImpl
 ☐ getRandomNumber
 ☐ getRanomString

☒ StockServiceImpl
 ☒ getStockList
 ☒ getStockByld

熔断方式

☐ 手动熔断
 ☒ 自动熔断

熔断最小调用

10000 个 / 500 ms

超时阈值

12000 ms

错误率

100 %


熔断恢复窗口

500 ms

保存

取消

服务接口测试


GoAPI

轻舟网关
切换产品

帮助
后台管理

产品概述

接口定义

接口测试

场景测试

执行集

资源管理

执行历史

搜索...

+

创建接口

批量操作

请输入搜索关键词...

HTTP接口类型

service1

alarm

stock-provider

stock-advisor

moco

demo

fsk-demo-srv-api-gw

测试服务API认证

接口名称	优先级	路径	用例数	创建人	最新状态	执行记录	操作
获得股票列表	P0	GET /provider/stocks	2	admin	09-18 15:05	<div></div>	编辑 执行 删除
熔断测试	P0	GET /provider/sleepgw	0	admin	11-02 15:22	<div></div>	编辑 执行 删除
获得股票详情	P0	GET /provider/stocks/#{stockIds}	2	admin	09-18 15:12	<div></div>	编辑 执行 删除
echo	P0	GET /provider/echo	0	admin	10-24 19:57	<div></div>	编辑 执行 删除
股票情况	P0	GET /api/stocks	1	admin	01-30 10:20	<div></div>	编辑 执行 删除

共 5 条 < 1 >

www.163yun.com