微服务注册发现

Eureka/Ribbon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 服务发现需求和模式

03 Netflix Eureka/Ribbon背景介绍

04 Eureka/Ribbon架构设计原理

05 Spring Cloud Eureka/Ribbon基础实验(Lab01)

06 Spring Cloud Eureka/Ribbon高级实验(Lab02)

07 Spring Cloud Eureka/Ribbon主要配置项

08 Eureka进阶~自保护模式

课程大纲

09 Eureka进阶~健康检查和蓝绿发布

10 Spring Cloud Zuul/Eureka/Ribbon集成实验(Lab03)

11 常见服务发现组件比较

12 ServiceMesh/Istio简介

13 基于Eureka/Zuul和容器云的持续交付架构

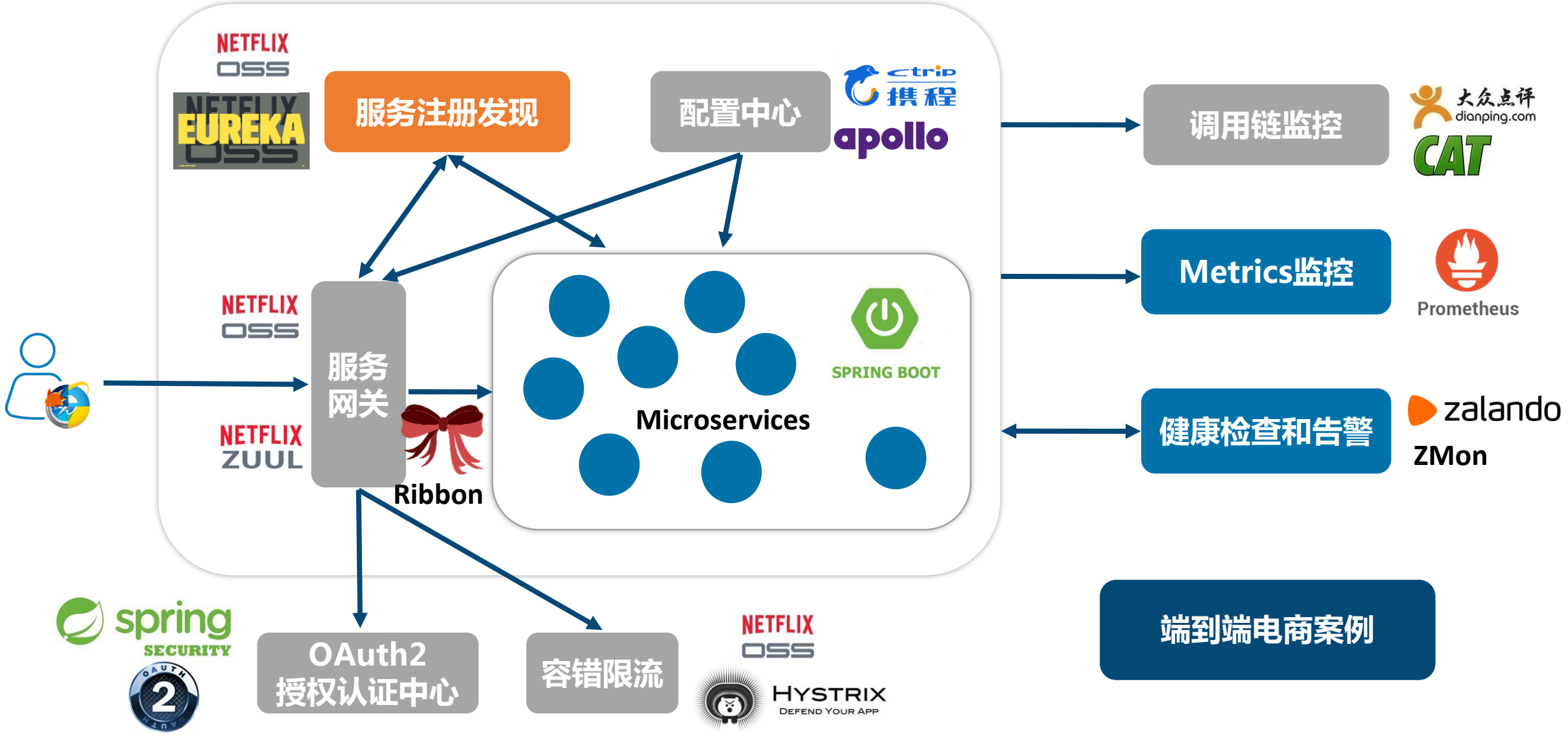
14 参考资源和后续课程预览

课程概述和亮点

1. 杨波的微服务基础架构体系的**第六个**模块
2. 服务发现**原理模式**剖析
3. Netflix开源服务发现组件**Eureka/Ribbon深度剖析**
4. Spring Cloud Eureka/Ribbon**案例和实验**
5. 基于Eureka/Zuul的**持续交付体系**
6. **常见服务发现组件比较**
7. **ServiceMesh/Istio**简介

杨波的微服务基础架构体系2018预览

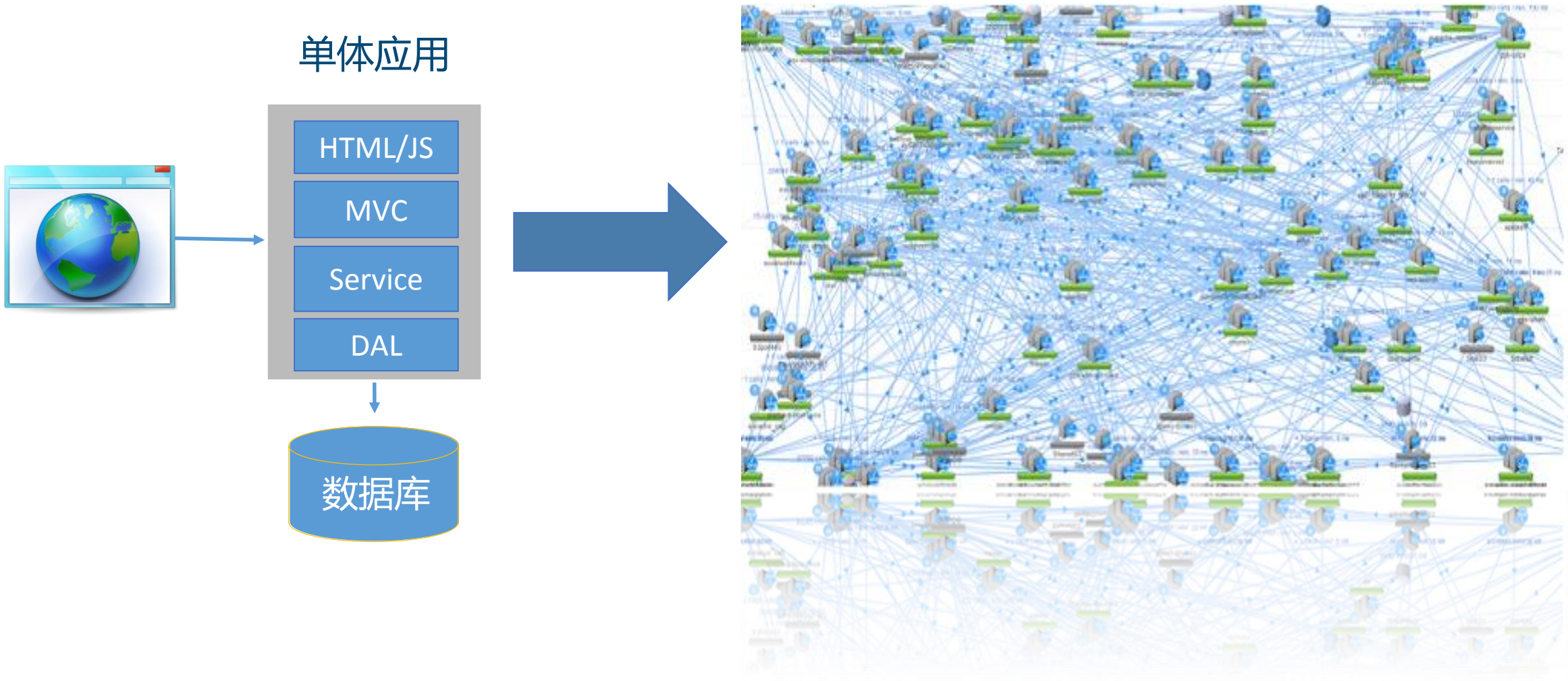
架构和技术栈预览


NETFLIX OSS
EUREKA
服务注册发现
配置中心
ctrip 携程
apollo
调用链监控
大众点评 dianping.com
CAT
NETFLIX OSS
服务网关
Metrics监控
Prometheus
NETFLIX ZUUL
Ribbon
Microservices
SPRING BOOT
健康检查和告警
zalando
ZMon
spring SECURITY
OAuth2 授权认证中心
容错限流
NETFLIX OSS
HYSTRIX
DEFEND YOUR APP
端到端电商案例

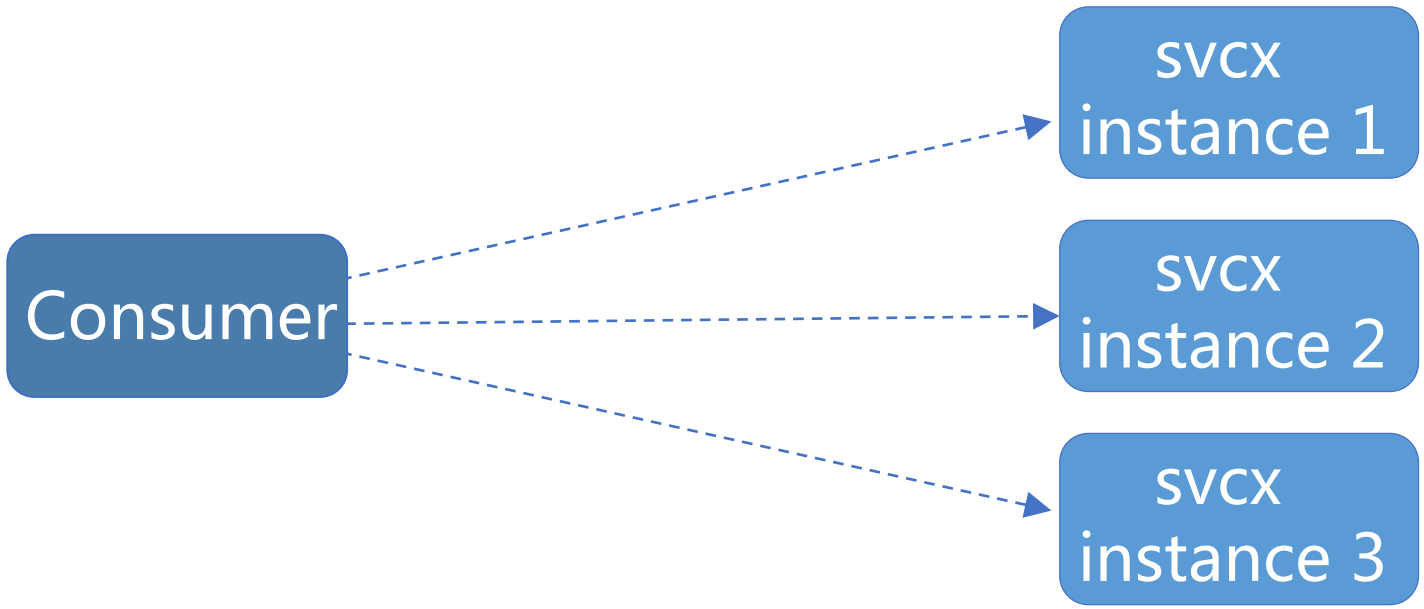
第 2 部分

服务发现需求和模式

从单块到微服务



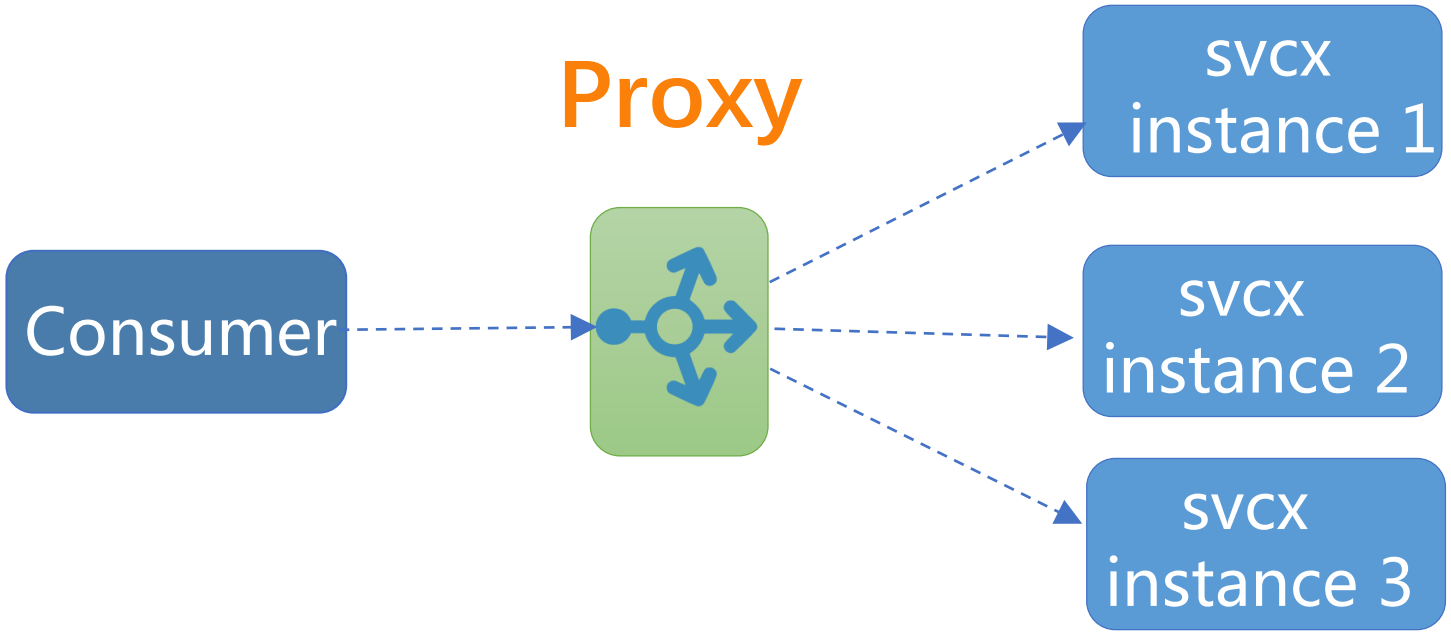
微服务架构的根本问题



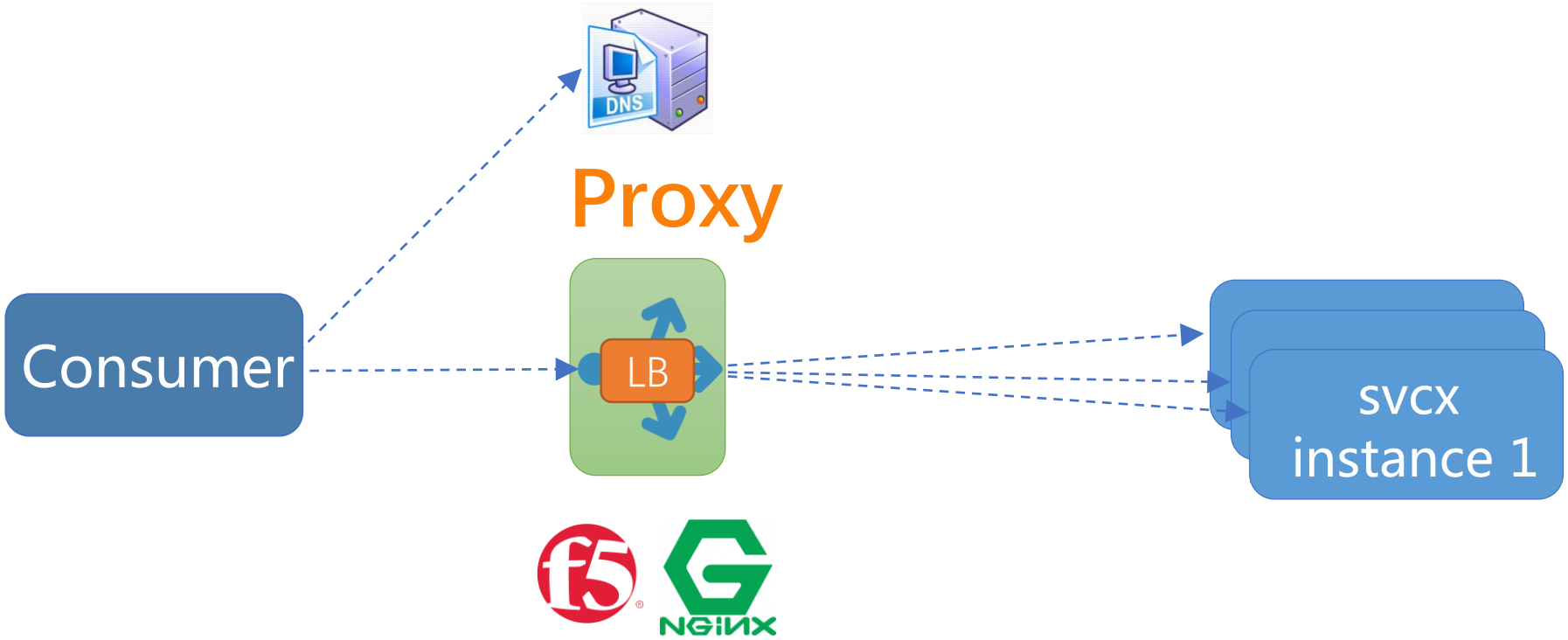
服务发现

负载均衡

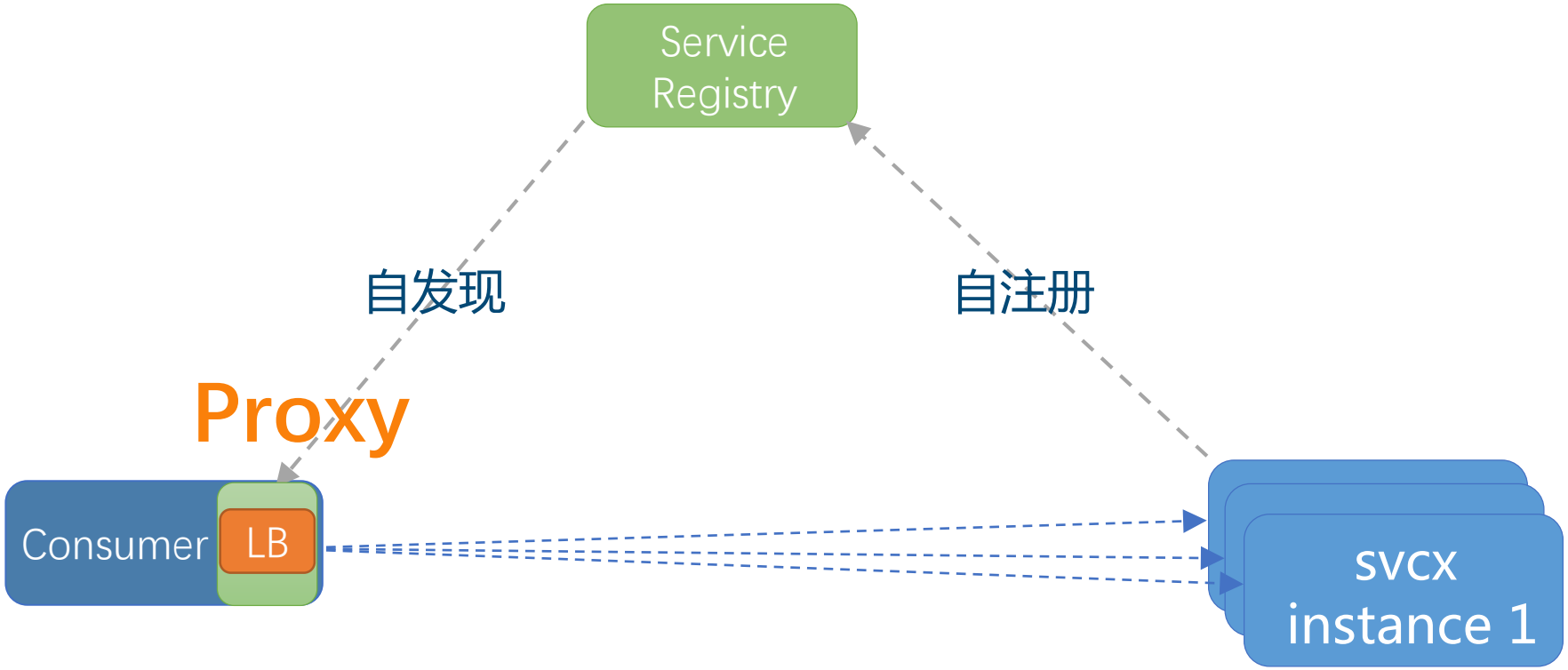
解决办法：代理


Proxy
Consumer
svcx
instance 1
svcx
instance 2
svcx
instance 3

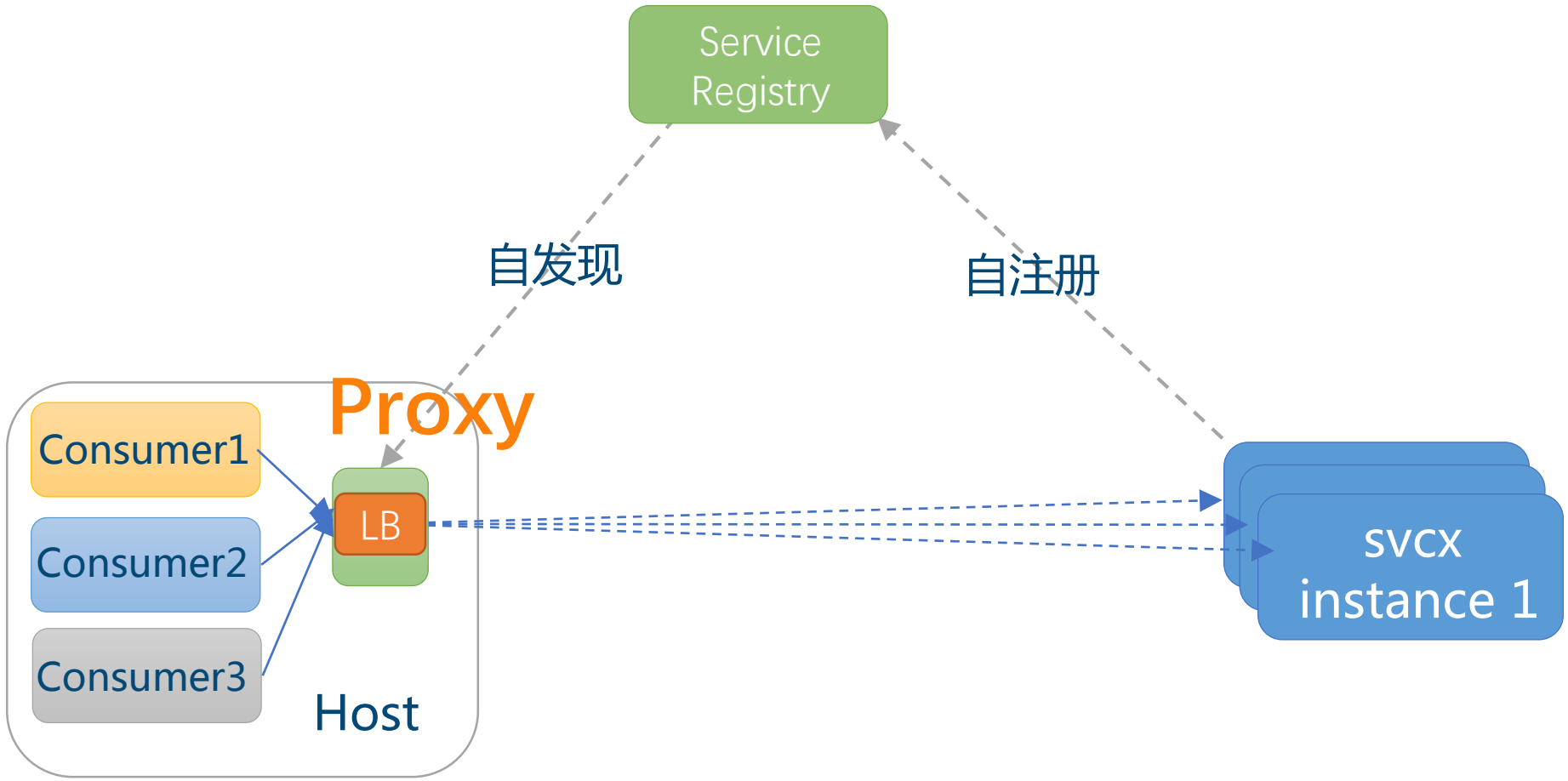
模式一： 传统集中式代理



模式二：服务注册中心+客户端嵌入式代理



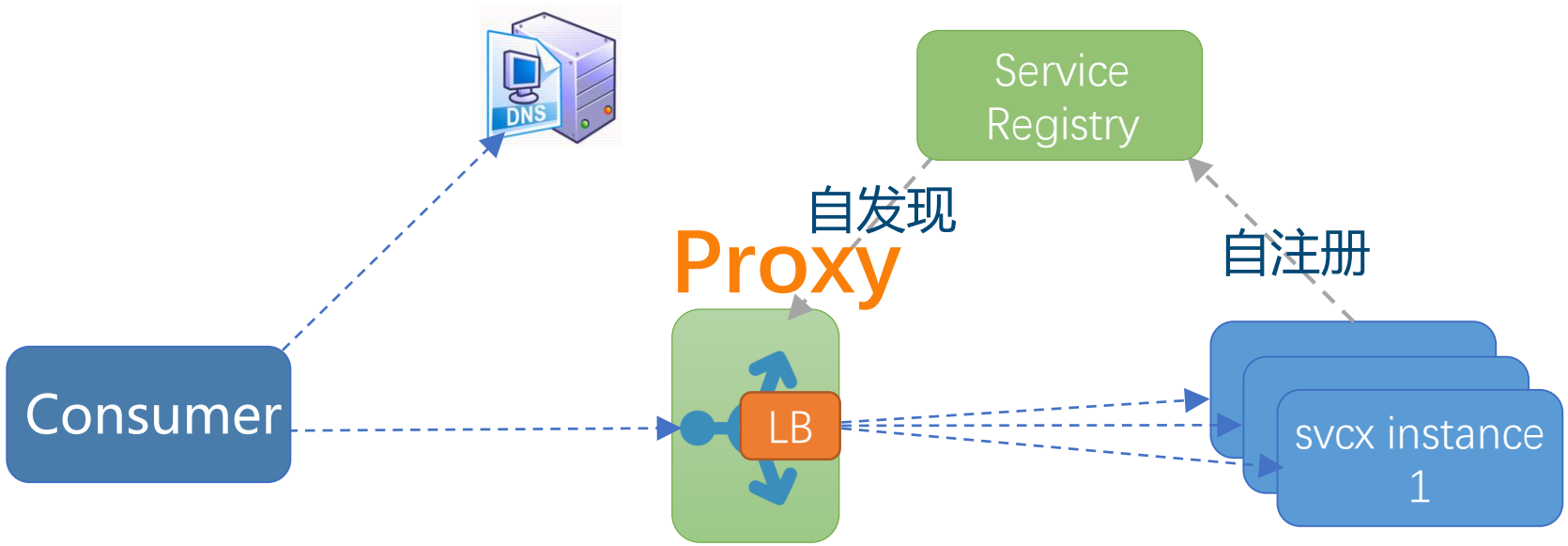
模式三：主机独立进程代理



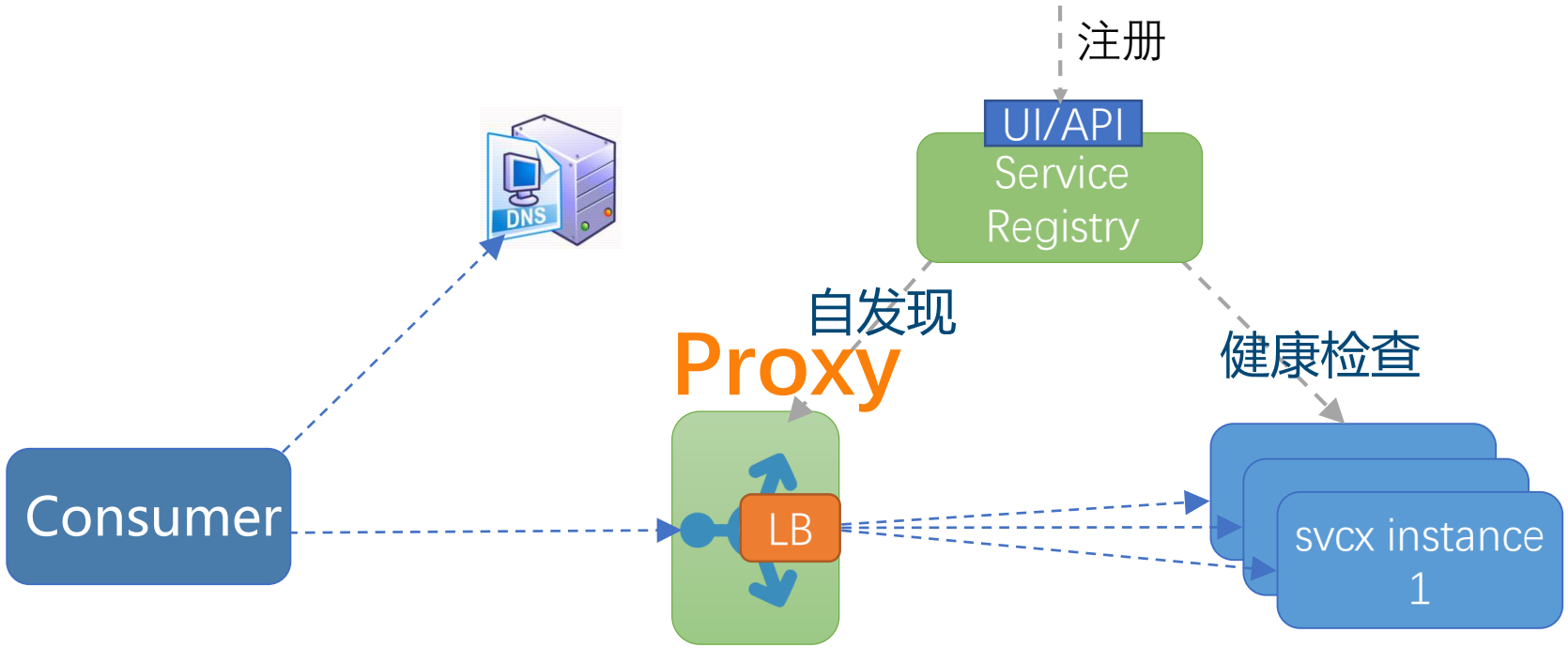
比较

模式	优	不足	适用场景	公司案例
集中式代理	运维简单 集中治理 语言栈无关	配置麻烦周期长 单点问题 多一跳有性能开销	中大小规模公司都适用，需一定运维能力	亿贝，携程，拍拍贷
客户端嵌入式代理	无单点，性能好	客户端复杂 多语言麻烦 治理松散	中大规模公司，语言栈较统一	Twitter Finagle，阿里Dubbo，Netflix Karyon, 新浪微博Motan
主机独立进程代理	折中方案	运维部署复杂	中大规模公司，运维能力强	Airbnb SmartStack，唯品会，新浪微博Motan，Istio ServiceMesh

模式一：变体1


DNS
Service
Registry
自发现
Proxy
自注册
Consumer
LB
svcx instance
1

模式一：变体2


注册
UI/API
Service
Registry
DNS
自发现
Proxy
健康检查
Consumer
LB
svcx instance
1

第 3 部分

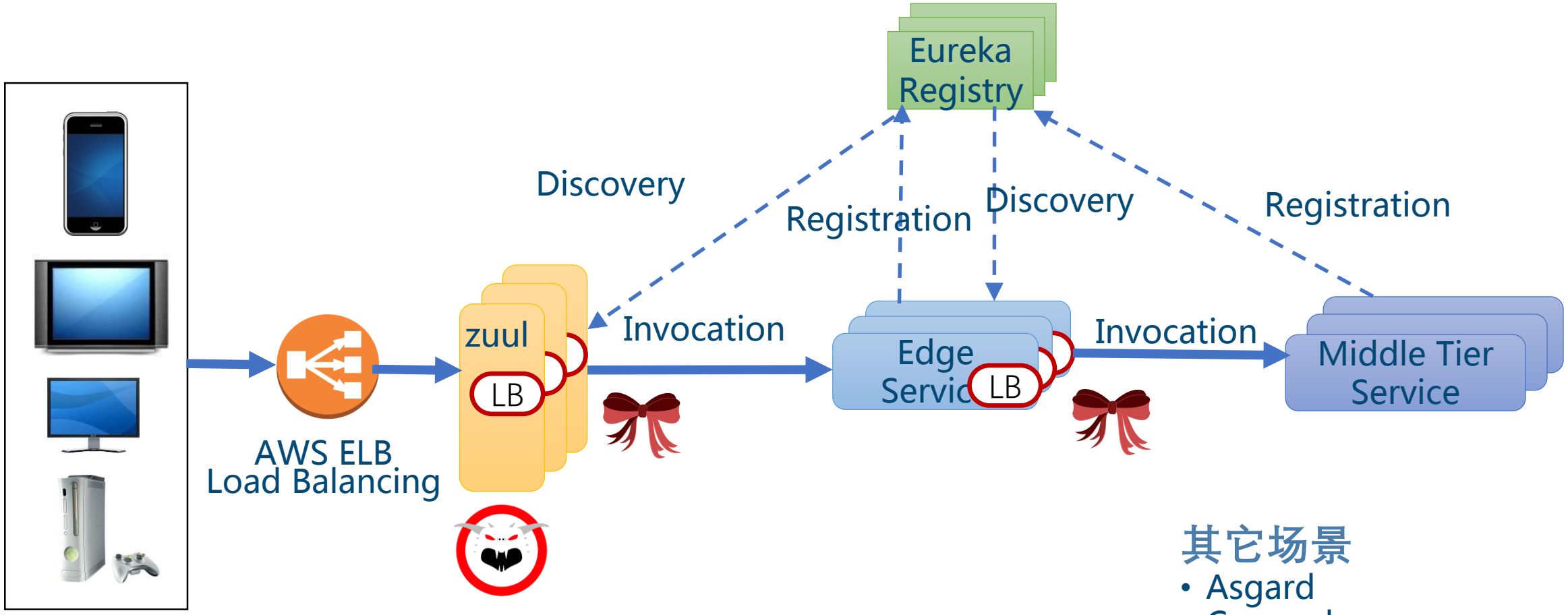
Netflix Eureka/Ribbon背景介绍

Eureka/Ribbon背景

- Eureka
  - 阿基米德发现浮力定律时发出的惊叹声，寓意微服务发现
  - 云服务发现组件，支持跨区域高可用(AP)
  - 提供Java based client，提供RESTful API支持多语言对接
  - 主要解决云中服务实例动态启停、漂移问题
  - 2012年开源，github > 6.2k星
    - https://github.com/Netflix/eureka
    - https://github.com/Netflix/eureka/wiki/Eureka-at-a-glance

- Ribbon
  - 蝴蝶结意思，寓意微服务联结
  - 客户端软负载组件，支持Eureka对接，支持多种可插拔LB策略
  - Java based
  - 2013年初开源，github > 2.3k星
    - https://github.com/Netflix/ribbon
    - https://medium.com/netflix-techblog/announcing-ribbon-tying-the-netflix-mid-tier-services-together-a89346910a62

主要场景@Netflix~中间层负载均衡

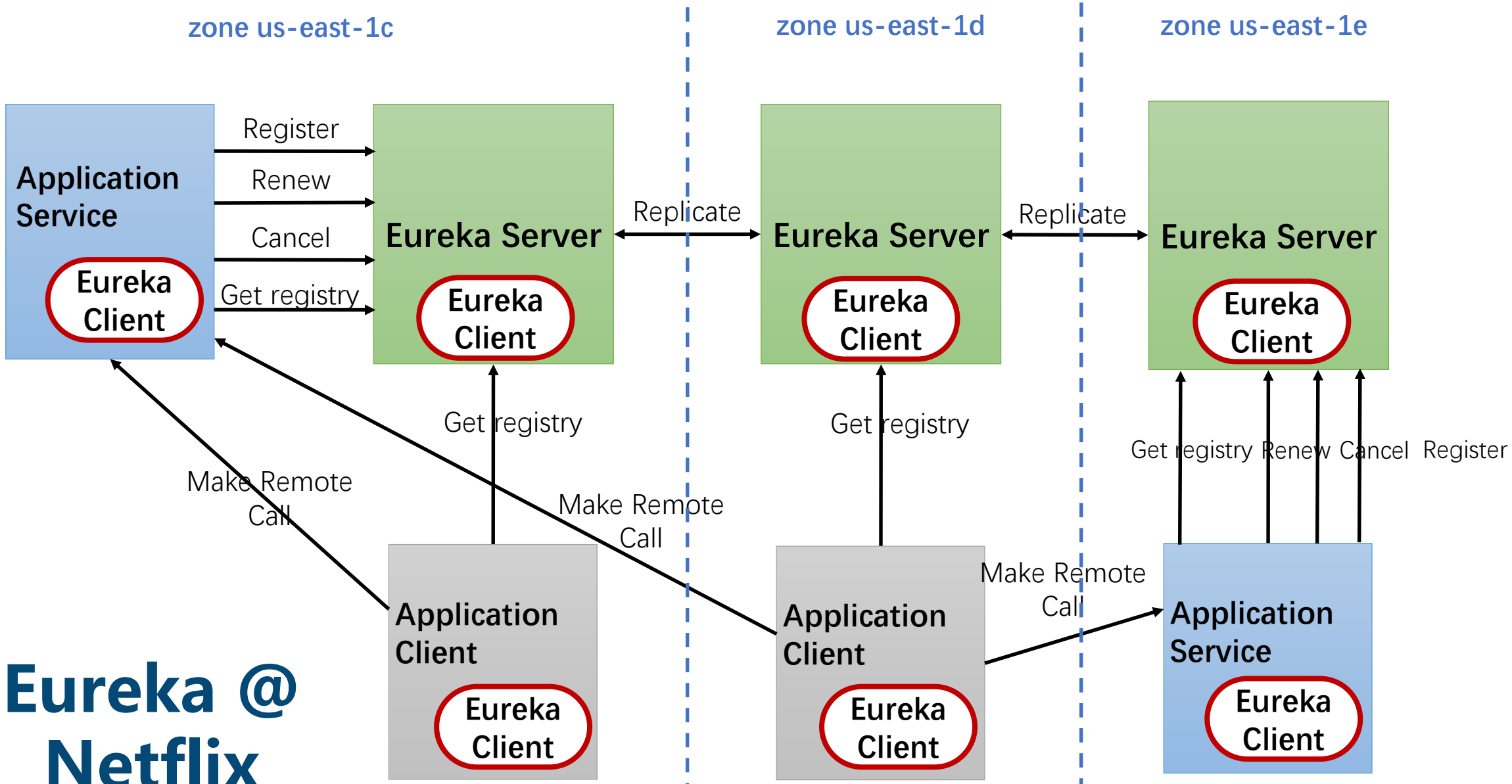


其它场景

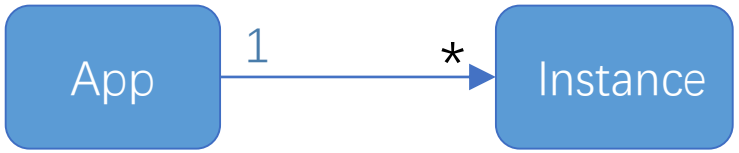
- Asgard
- Cassandra
- Memcached

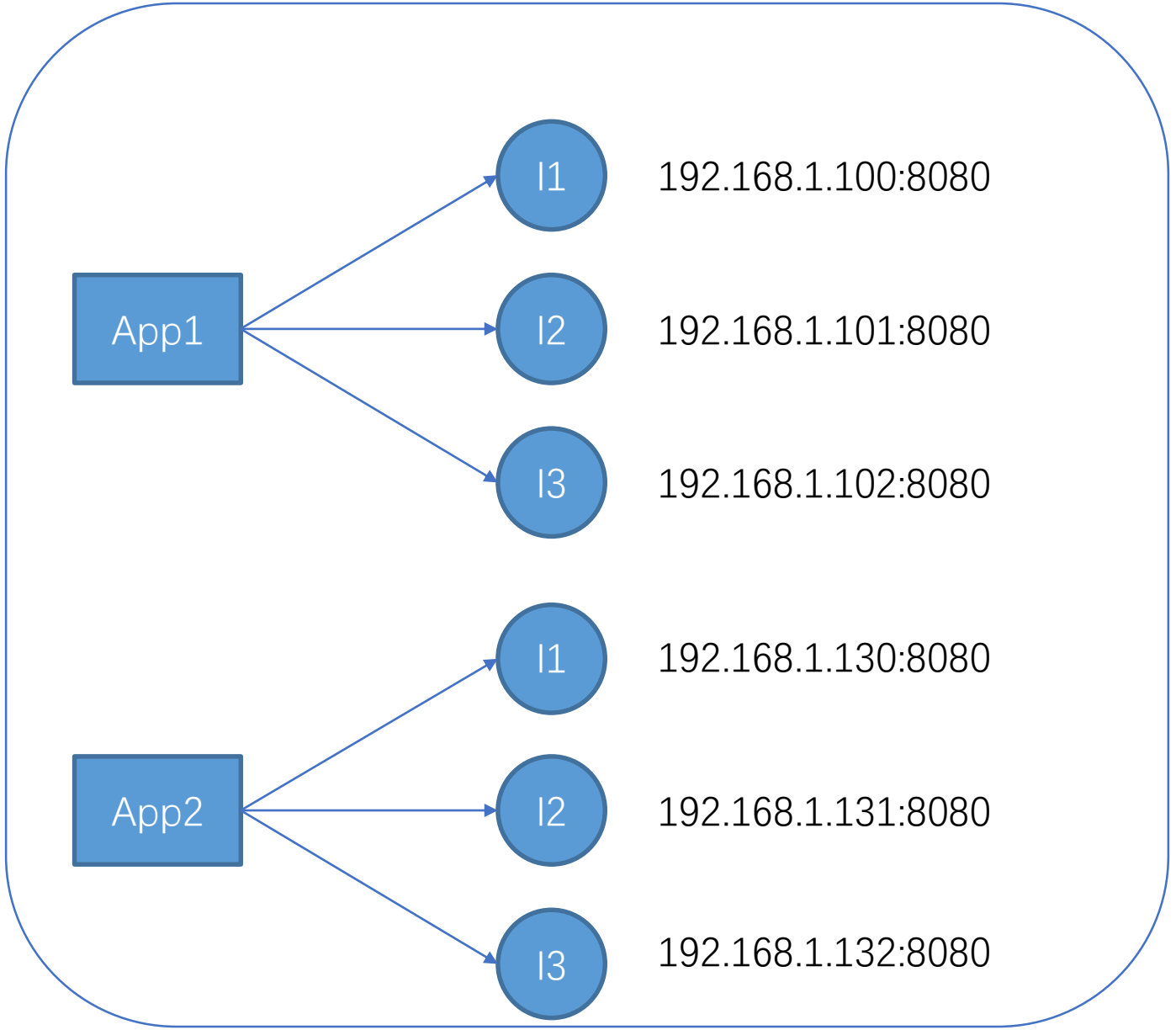
第 4 部分

Eureka/Ribbon架构设计原理

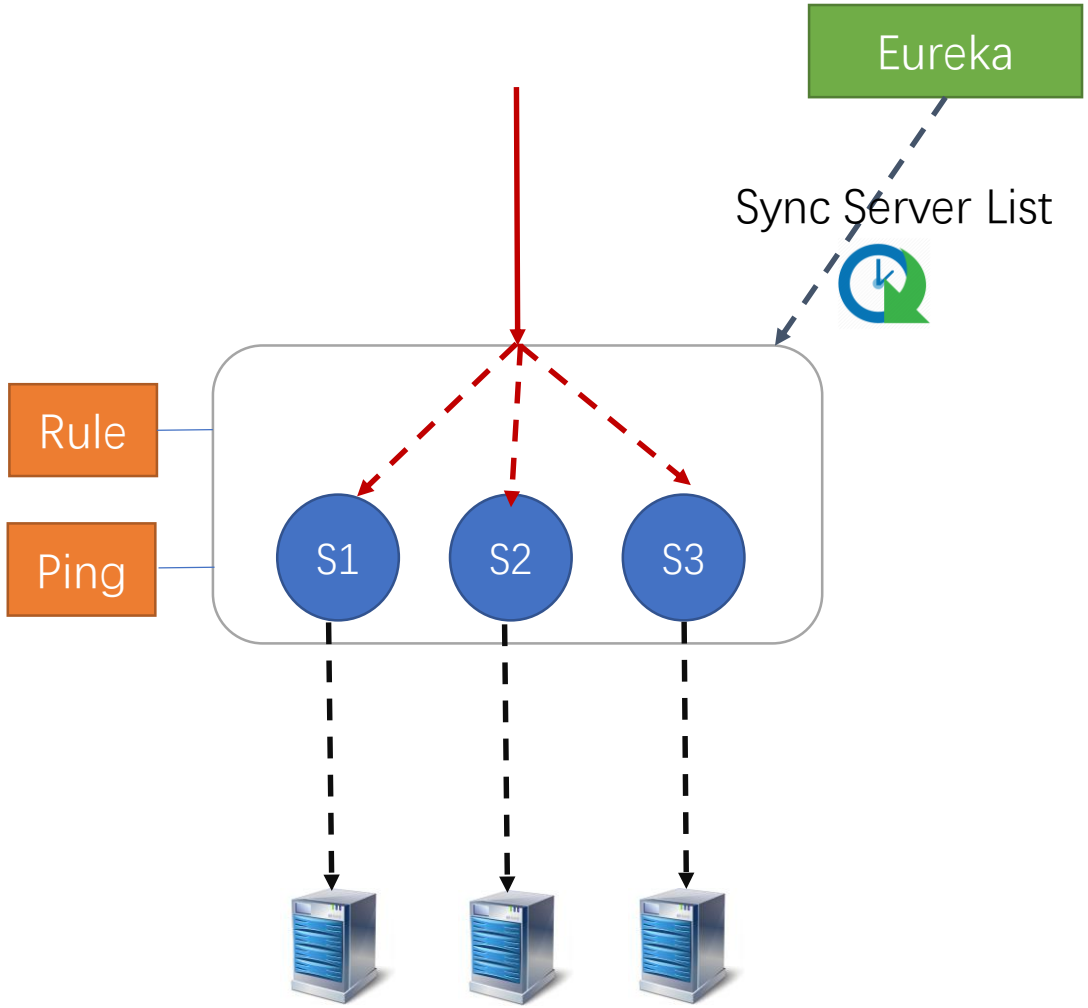

zone us-east-1c
zone us-east-1d
zone us-east-1e
Application Service
Eureka Client
Register
Renew
Cancel
Get registry
Eureka Server
Eureka Client
Replicate
Eureka Server
Eureka Client
Replicate
Eureka Server
Eureka Client
Get registry
Get registry
Get registry Renew Cancel Register
Make Remote Call
Make Remote Call
Make Remote Call
Application Client
Eureka Client
Application Client
Eureka Client
Application Service
Eureka Client
Eureka @ Netflix

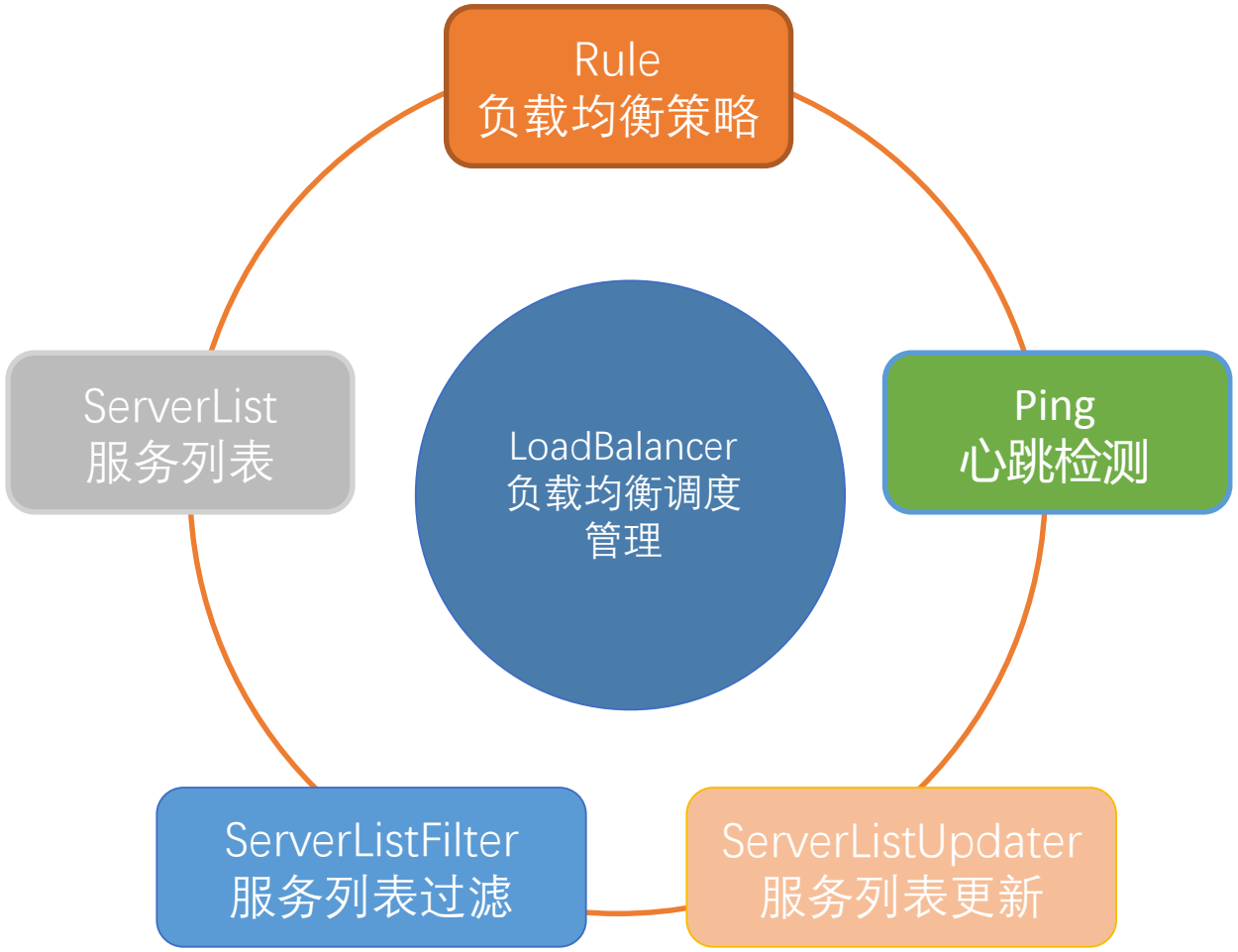
内存概念模型


App
1
*
Instance


App1
I1
I2
I3
192.168.1.100:8080
192.168.1.101:8080
192.168.1.102:8080
App2
I1
I2
I3
192.168.1.130:8080
192.168.1.131:8080
192.168.1.132:8080

Ribbon设计


Eureka
Sync Server List
Rule
Ping
S1
S2
S3


Rule
负载均衡策略
ServerList
服务列表
LoadBalancer
负载均衡调度
管理
Ping
心跳检测
ServerListFilter
服务列表过滤
ServerListUpdater
服务列表更新

参考

深度剖析服务发现组件~Netflix Eureka

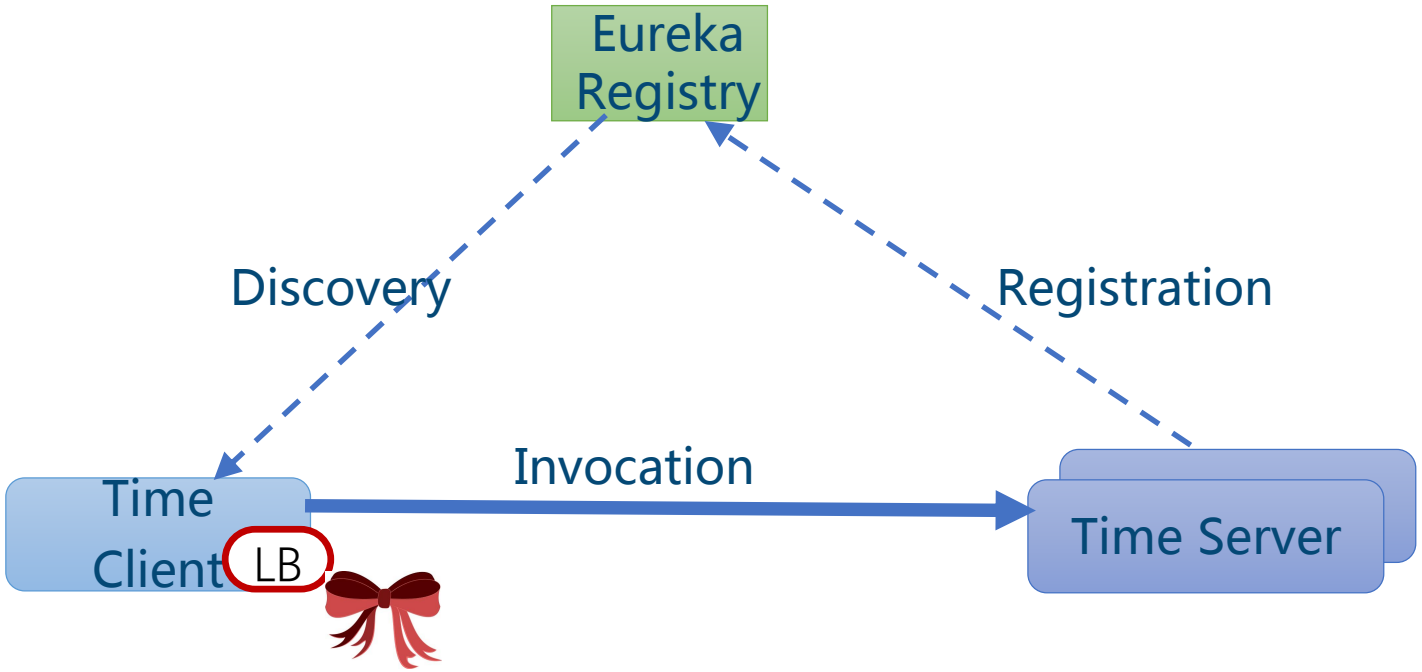
- https://zhuanlan.zhihu.com/p/24829766

深入理解Ribbon之源码解析

- https://blog.csdn.net/forezp/article/details/74820899

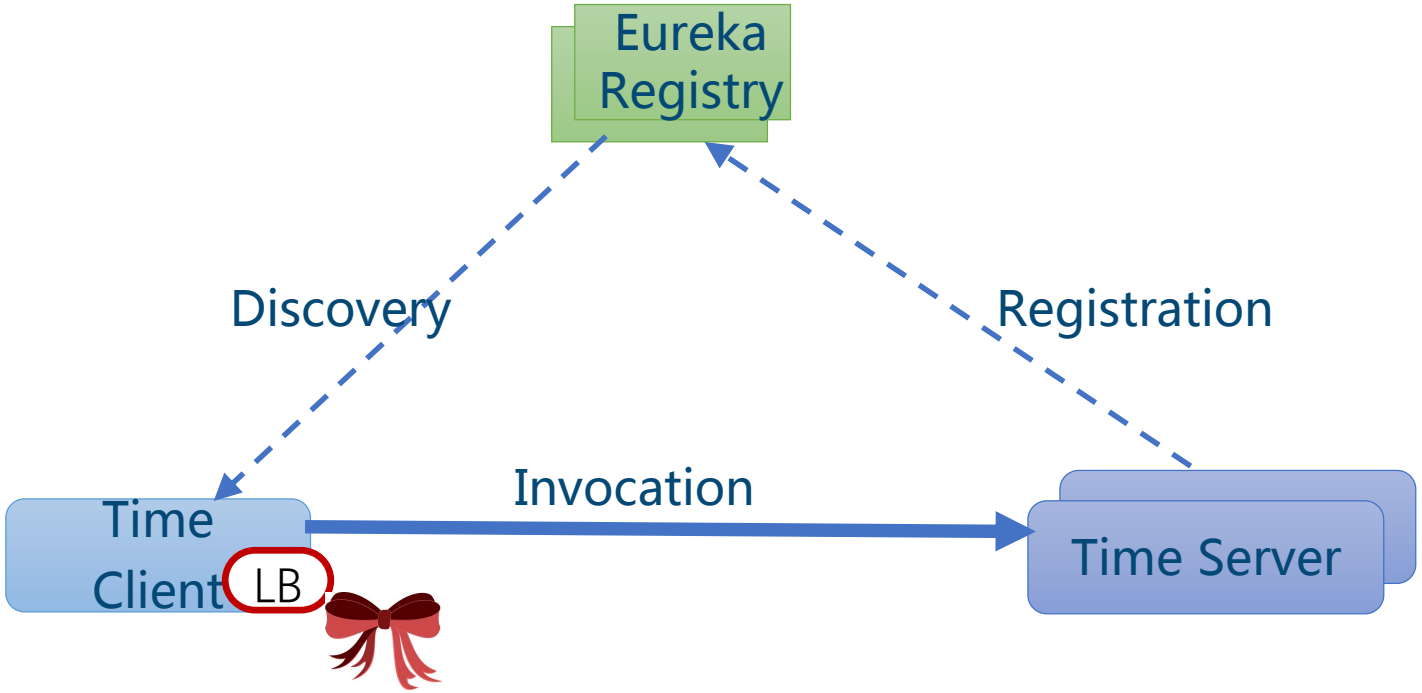
第5部分

Spring Cloud Eureka/Ribbon 基础实验(Lab01)


Eureka
Registry
Discovery
Registration
Invocation
Time
Client
LB
Time Server

第 6 部分

Spring Cloud Eureka/Ribbon 高级实验(Lab02)


Eureka
Registry
Discovery
Registration
Invocation
Time
Client
LB
Time Server

第 7 部分

Spring Cloud Eureka/Ribbon 主要配置项

Spring Cloud Eureka服务端主要配置项

配置参数(eureka.server.*)	默认值	说明
enableSelfPreservation	true	启用注册中心的自保护机制，Eureka 如果统计到15分钟之内损失>15%的微服务心跳，则将会触发自保护机制，不再剔除服务提供者。
waitTimeInMsWhenSyncEmpty	1000 * 60 * 5	在Eureka服务器获取不到集群里对等服务器上的实例时，需要等待的时间，单位为毫秒，默认为1000 * 60 * 5。单机开发模式建议设置为0。

仅供参考，以Spring Cloud具体版本为准！

Spring Cloud Eureka主要客户信息配置

配置项(eureka.client.*)	说明	默认值
serviceUrl	指定服务注册中心地址，类型为 HashMap，并设置有一组默认值，默认的Key为 defaultZone；默认的Value为 http://localhost:8761/eureka ，如果服务注册中心为高可用集群时，多个注册中心地址以逗号分隔。 如果服务注册中心加入了安全验证，这里配置的地址格式为： http://<username>:<password>@localhost:8761/eureka 其中 <username> 为安全校验的用户名；<password> 为该用户的密码	
fetchRegistry	是否从Eureka服务端获取注册信息	true
registryFetchIntervalSeconds	从Eureka服务端获取注册信息的间隔时间，单位为秒	30
registerWithEureka	是否要将自身的实例信息注册到Eureka服务端	true

仅供参考，以Spring Cloud具体版本为准！

Spring Cloud Eureka主要实例级别配置

配置项(eureka.instance.*)	说明	默认值
leaseRenewalIntervalInSeconds	Eureka客户端向服务端发送心跳的时间间隔，单位为秒	30
leaseExpirationDurationInSeconds	Eureka服务端在收到最后一次心跳之后等待的过期时间上限，单位为秒。超过该时间没有收到心跳，则服务端会将该服务实例从服务清单中剔除，从而禁止服务调用请求被发送到该实例上	90
appname	服务名，默认取spring.application.name的配置值，如果没有则为unknown	
hostname	主机名，不配置的时候将根据操作系统的主机名来获取	
instance-id	注册到eureka的实例id，推荐${spring.cloud.client.ip-address}:${spring.application.name}:${server.port}	主机名

仅供参考，以Spring Cloud具体版本为准！

Spring Cloud Ribbon主要配置项

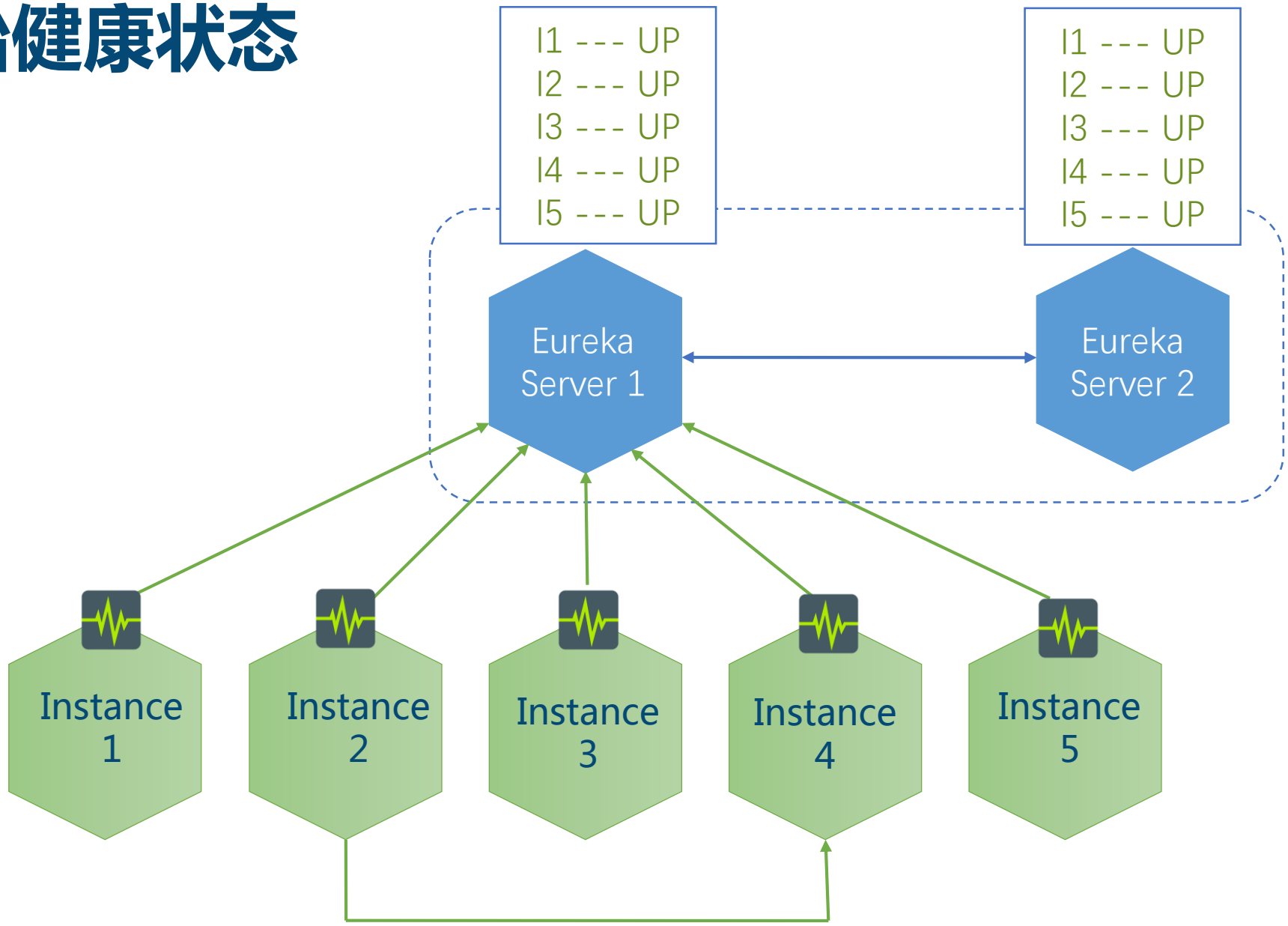
配置项({svc}.ribbon.*)	说明	默认值
ConnectionTimeout	连接超时时间	1000ms
ReadTimeout	读取超时时间	1000ms
ServerListRefreshInterval	刷新服务列表源的间隔时间	30秒
NFLoadBalancerClassName	定制ILoadBalancer实现	com.netflix.loadbalancer.ZoneAwareLoadBalancer
NFLoadBalancerRuleClassName	定制IRule实现	com.netflix.loadbalancer.ZoneAvoidanceRule
NFLoadBalancerPingClassName	定制IPing	com.netflix.loadbalancer.DummyPing
NIWSServerListClassName	定制ServerList	com.netflix.loadbalancer.ConfigurationBasedServerList
ServerListUpdateClassName	定制ServerListUpdater	com.netflix.loadbalancer.PollingServerListUpdater
NIWSServerListFilterClassName	定制SeverListFilter	com.netflix.loadbalancer.ZonePreferenceServerListFilter

仅供参考，以Spring Cloud具体版本为准！

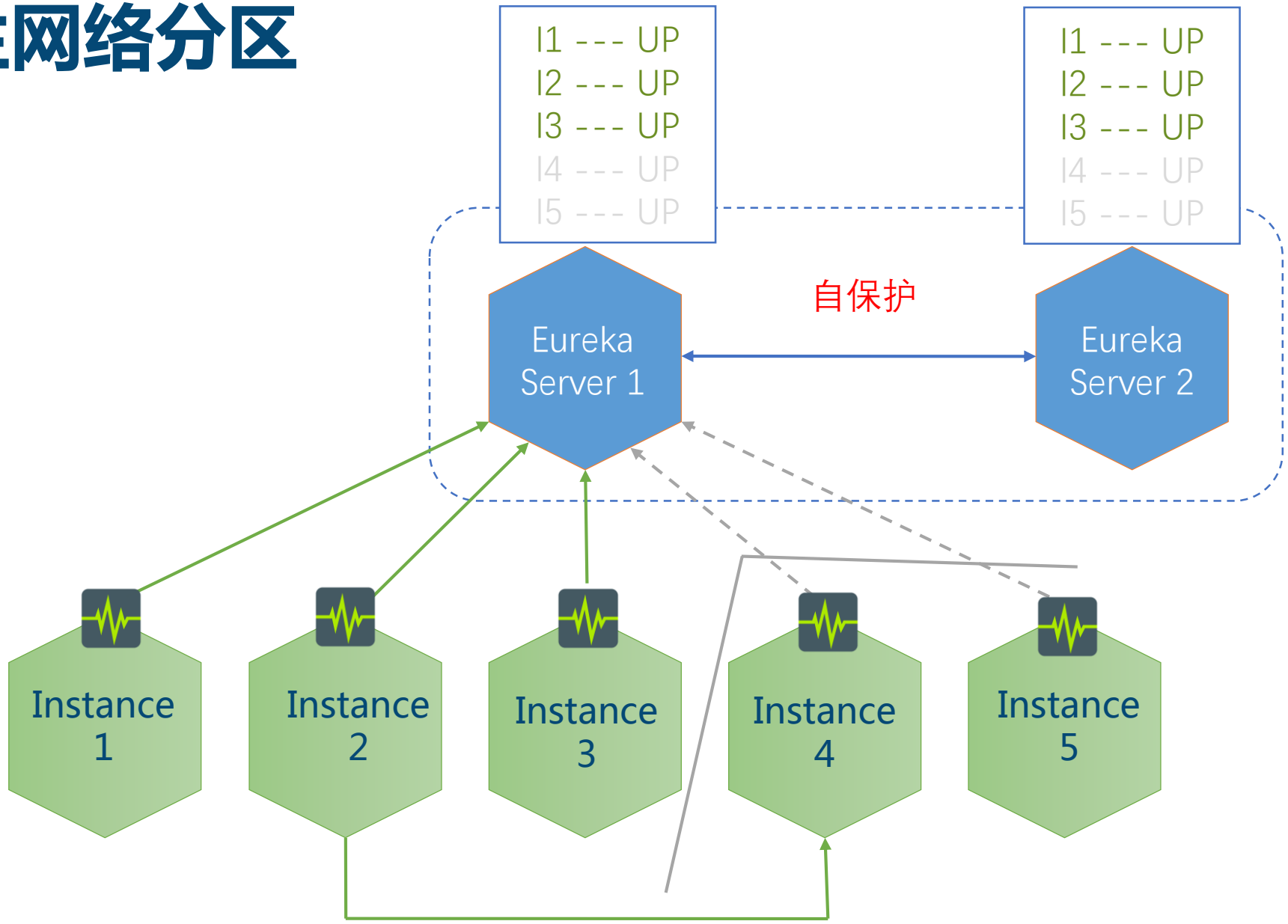
第 8 部分

Eureka进阶：自保护模式

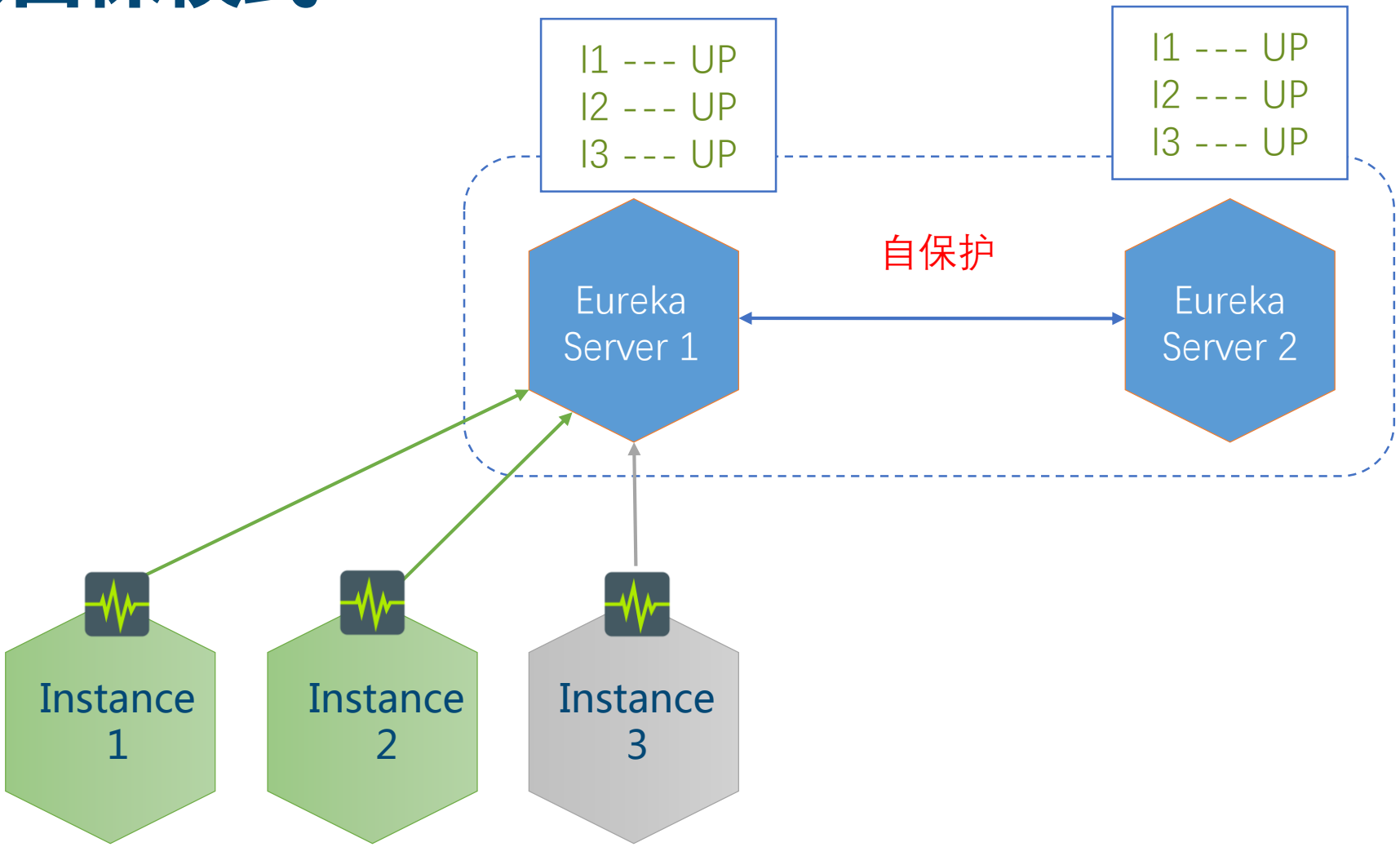
初始健康状态


I1 --- UP
I2 --- UP
I3 --- UP
I4 --- UP
I5 --- UP
I1 --- UP
I2 --- UP
I3 --- UP
I4 --- UP
I5 --- UP
Eureka Server 1
Eureka Server 2
Instance 1
Instance 2
Instance 3
Instance 4
Instance 5

发生网络分区


I1 --- UP
I2 --- UP
I3 --- UP
I4 --- UP
I5 --- UP
I1 --- UP
I2 --- UP
I3 --- UP
I4 --- UP
I5 --- UP
自保护
Eureka Server 1
Eureka Server 2
Instance 1
Instance 2
Instance 3
Instance 4
Instance 5

进入自保模式


I1 --- UP
I2 --- UP
I3 --- UP
I1 --- UP
I2 --- UP
I3 --- UP
自保护
Eureka Server 1
Eureka Server 2
Instance 1
Instance 2
Instance 3

参考

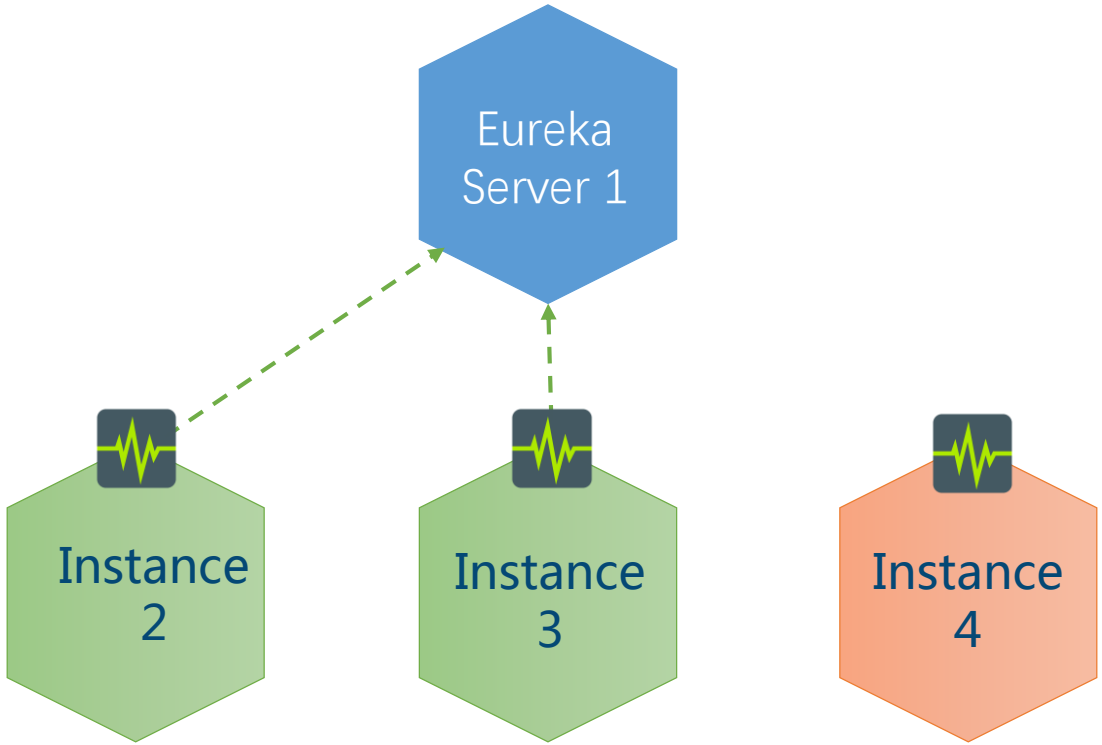
The Mystery of Eureka Self-preservation

- https://medium.com/@fahimfarookme/the-mystery-of-eureka-self-preservation-c7aa0ed1b799

第 9 部分

Eureka进阶：健康检查和蓝绿发布

报心跳式健康检查



PUT /eureka/apps/{appId}/{instanceId}?status=UP

PUT /eureka/apps/ORDER-SERVICE/localhost:order-service:8886?status=UP

HealthCheckHandler

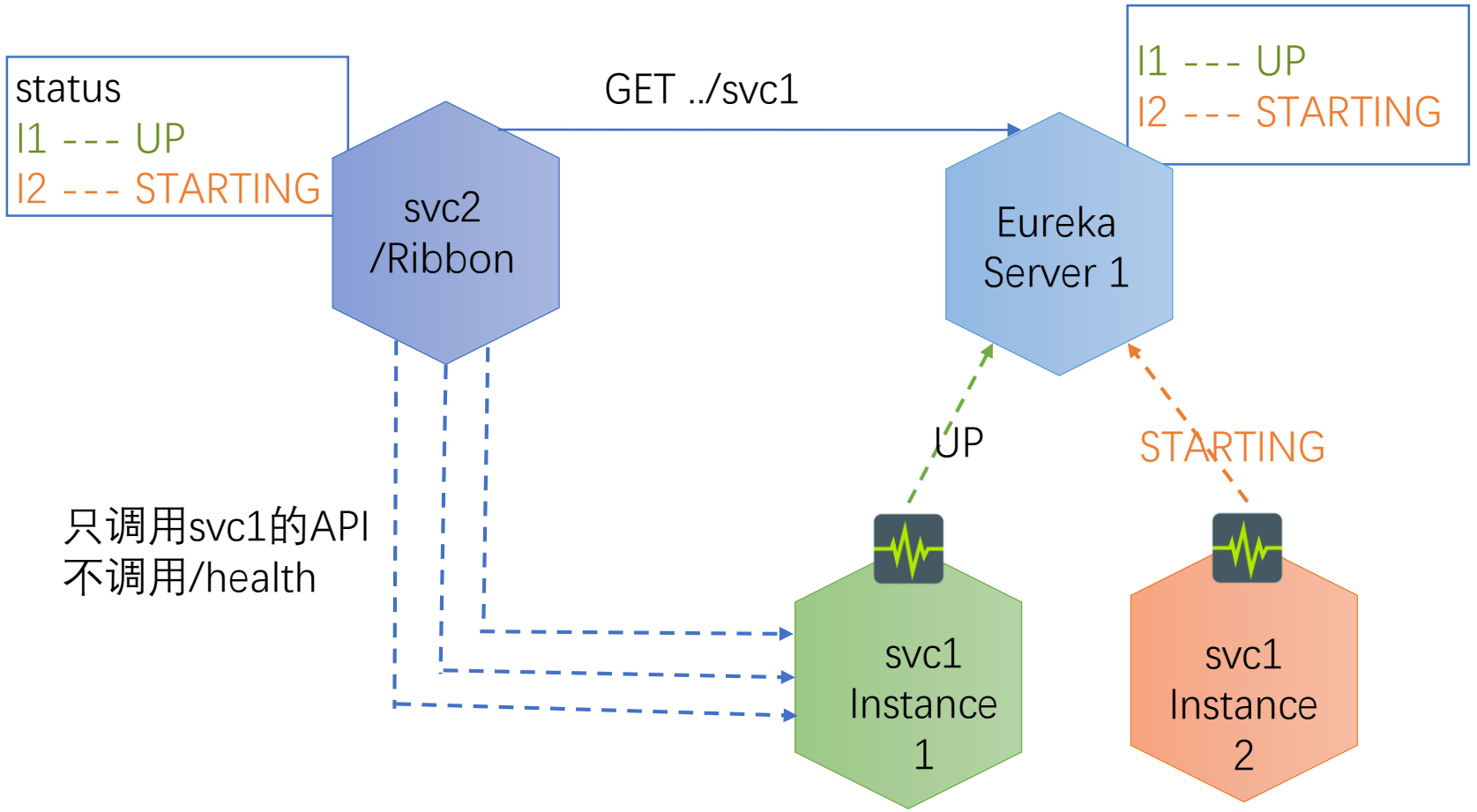
- **定制注册**
  - EurekaClient#registerHealthCheck
- **Spring Cloud**
  - eureka.client.healthcheck.enabled=true
  - EurekaHealthCheckHandler
    - DiskSpaceHealthIndicator
    - RefreshScopeHealthIndicator
    - HystrixHealthIndicator

获取应用信息

GET /eureka/apps/ORDER-SERVICE

```
<?xml version="1.0" encoding="UTF-8"?>
<application>
  <name>DISCOVERY-EUREKA-CLIENT</name>
  <instance>
    <instanceId>localhost:order-service:8886</instanceId>
    <ipAddr>192.168.1.6</ipAddr>
    <port>8886</port>
    <status>UP</status>
    <overriddenstatus>UP</overriddenstatus>
    <healthCheckUrl>http://localhost:8886/health</healthCheckUrl>
   ... ...
  </instance>
</application>
```

Ribbon软负载决策

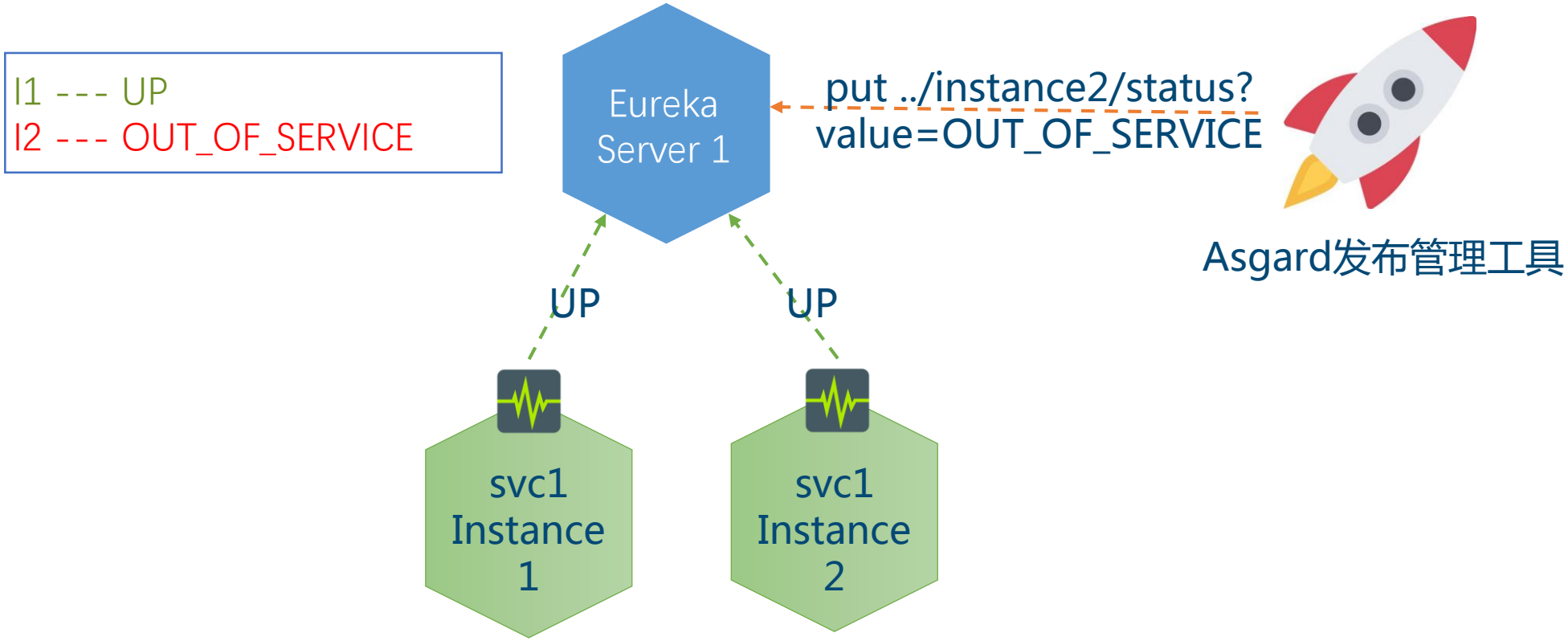


实例信息更新延迟

1. 注册延时（30秒）
2. Eureka服务器响应延迟（30秒）
3. Eureka客户端更新延迟（30秒）
4. Ribbon服务列表更新延迟（30秒）

最大可能有2分钟延迟

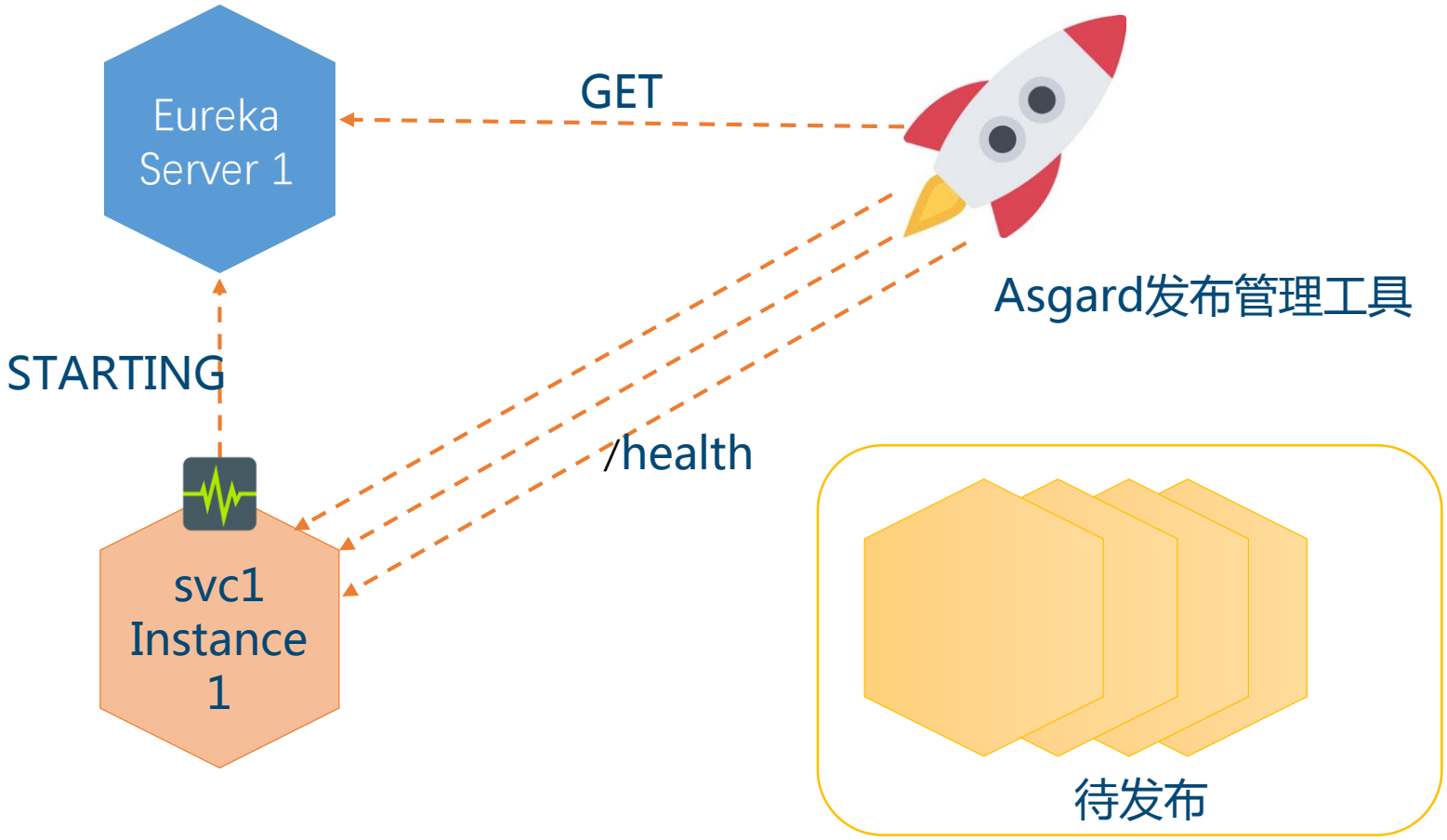
发布工具拉出



PUT /eureka/apps/{app id}/{instance id}/status?value={status}
DELETE /eureka/apps/{app id}/{instance id}/status

PUT /eureka/apps/ORDER-SERVICE/localhost:order-service:8886/status?value=OUT_OF_SERVICE

通过health端点检查快速发布



参考

The Mystery of Eureka Health Monitoring

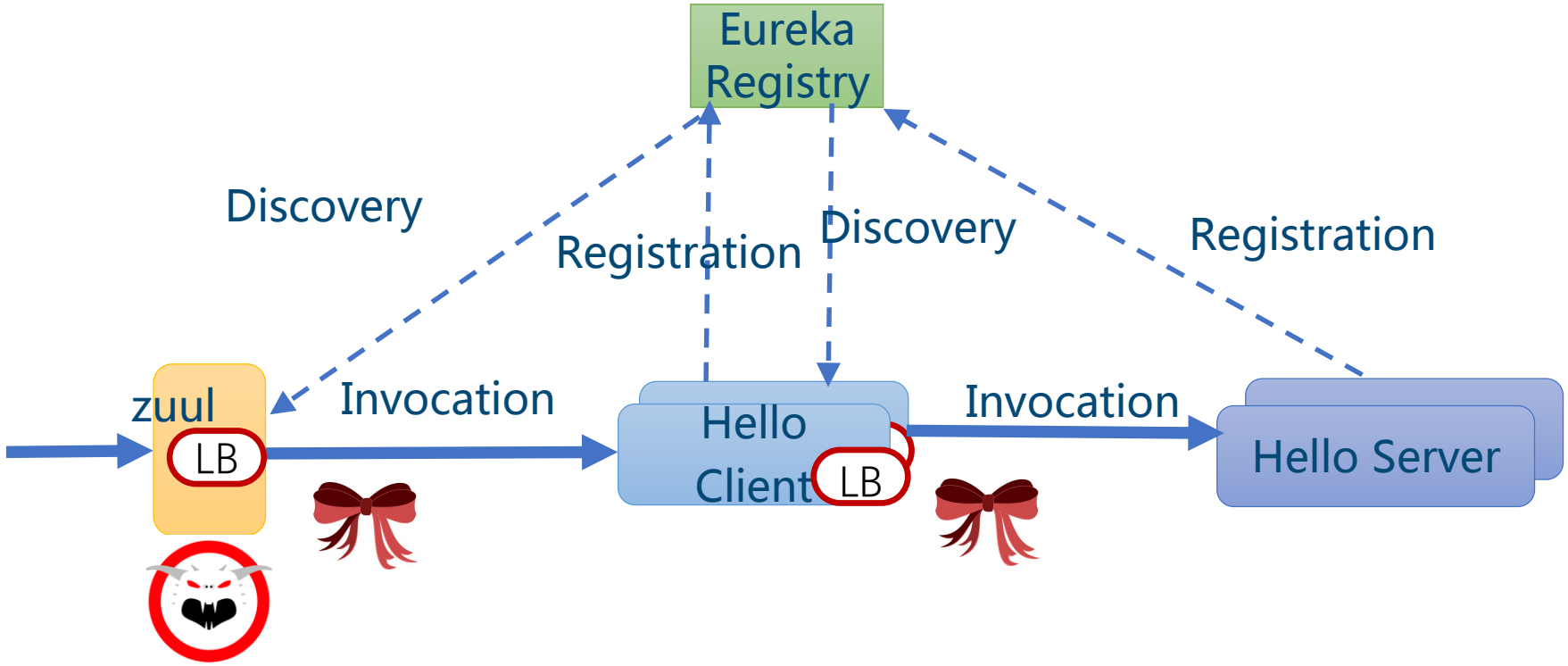
- https://medium.com/@fahimfarookme/the-mystery-of-eureka-health-monitoring-5305e3beb6e9

Spring Cloud Eureka – The Hidden Manual

- https://blog.asarkar.org/technical/netflix-eureka/

第10部分

Spring Cloud Zuul/Eureka/Ribbon集成实验(Lab03)


Eureka
Registry
Discovery
Registration
Discovery
Registration
zuul
LB
Invocation
Hello
Client
LB
Invocation
Hello Server

第11部分

常用服务发现组件比较

功能	Eureka	Consul	Zookeeper	Etcd
服务健康检查	客户主动报心跳	服务状态，内存，磁盘等	(弱)长连接，keepalive	连接心跳
多数据中心支持	支持	支持	不支持	不支持
Kv存储支持	不支持	支持	支持	支持
一致性协议	定制P2P	Raft	Paxos	Raft
CAP	AP	CA	CP	CP
客户端接口	Java/http	http/dns	客户端	http/grpc
watch支持	定期pull/大部分增量	全量/支持long pulling	支持	支持long pulling
自身监控	Metrics	Metrics	NA	Metrics
安全	NA	ACL/https	ACL	https支持(弱)
Spring cloud集成	已支持	已支持	已支持	已支持
社区流行度 （服务注册发现场景）	流行	流行	一般	一般

参考

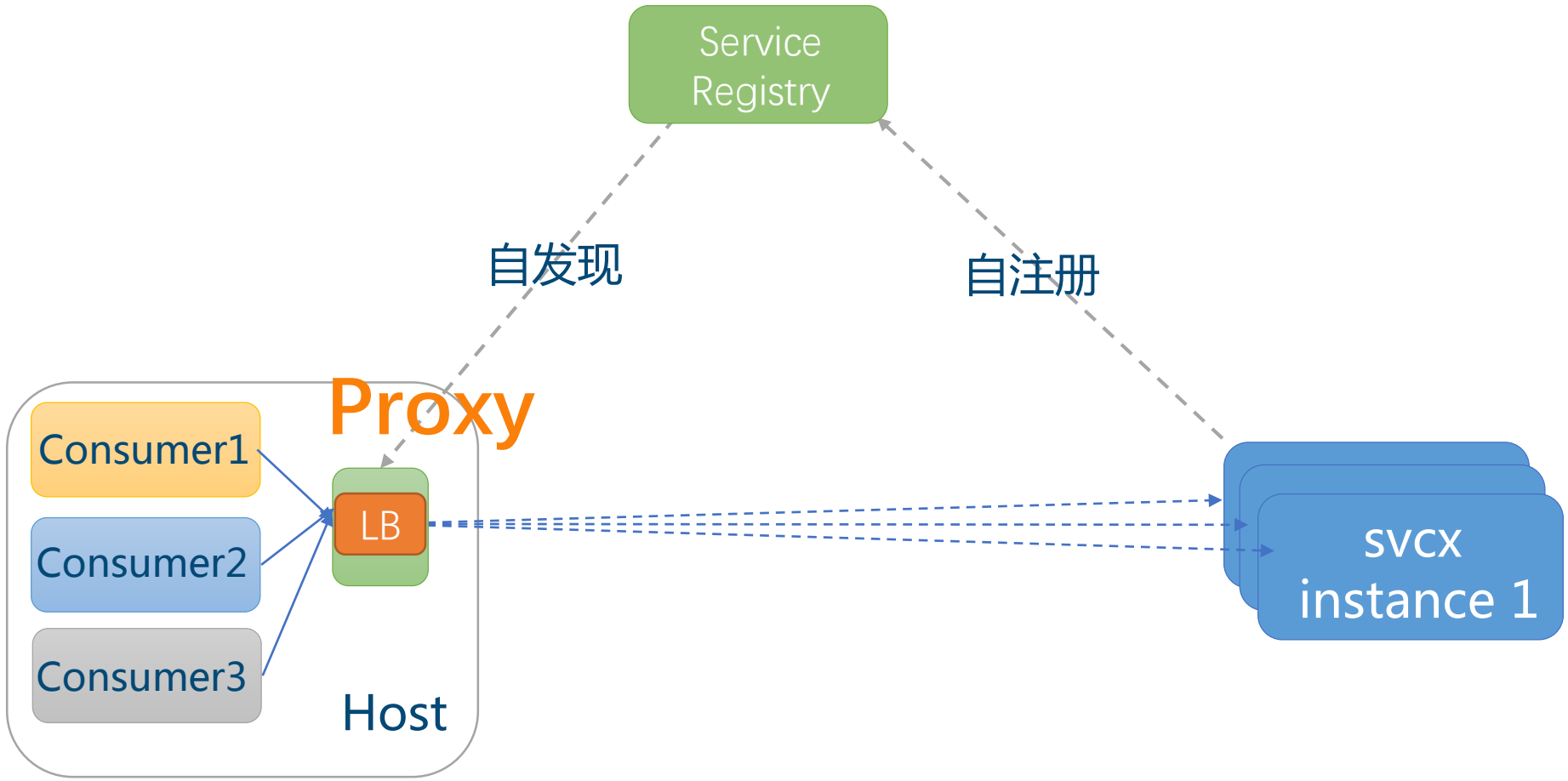
https://luyiisme.github.io/2017/04/22/spring-cloud-service-discovery-products/?utm_source=tuicool&utm_medium=referral

仅供参考，实际功能以具体版本为准！

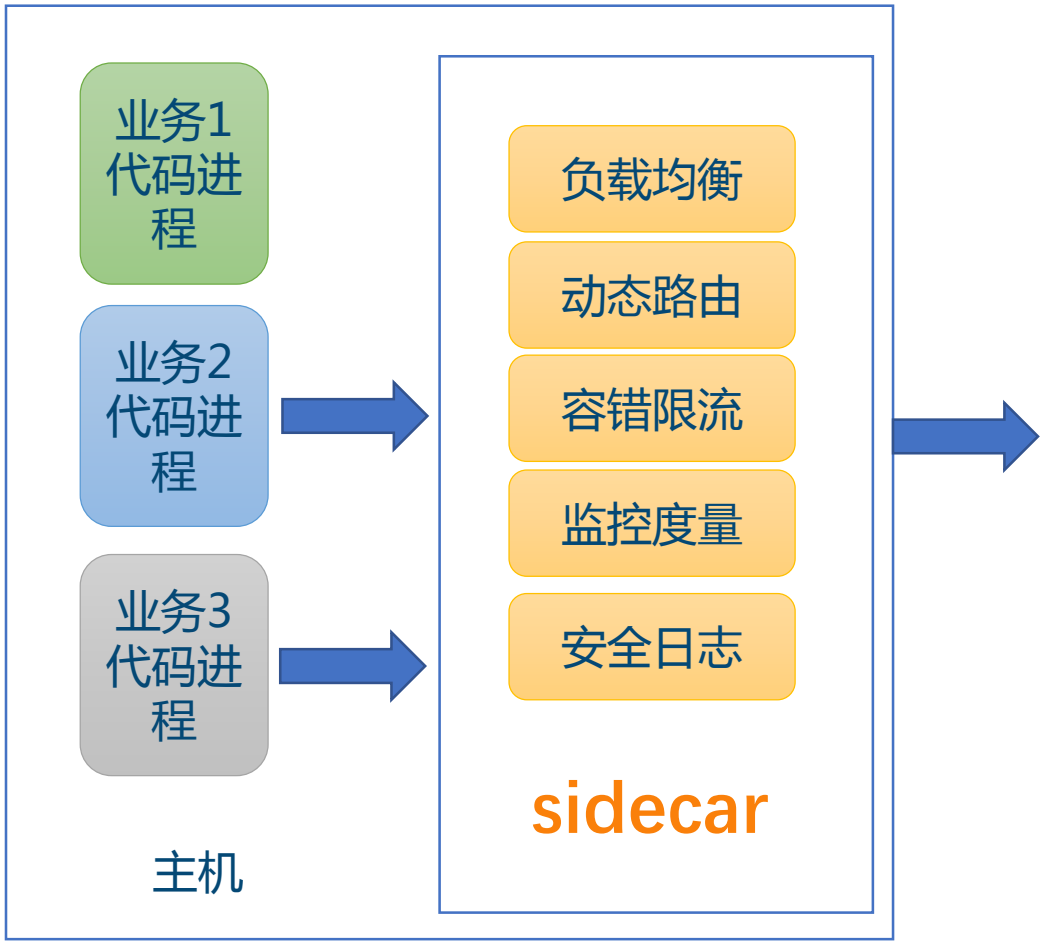
第12部分

ServiceMesh/Istio简介

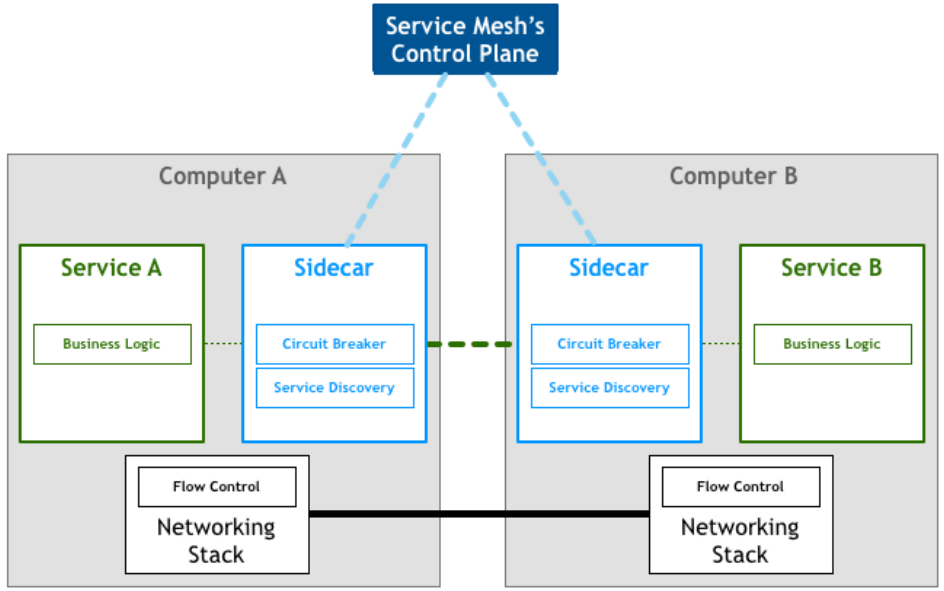
模式三回顾：主机独立进程代理

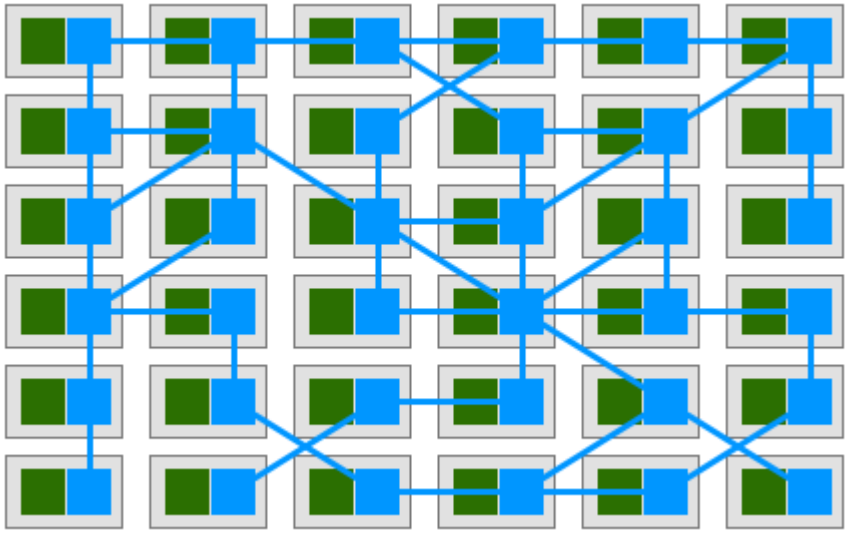


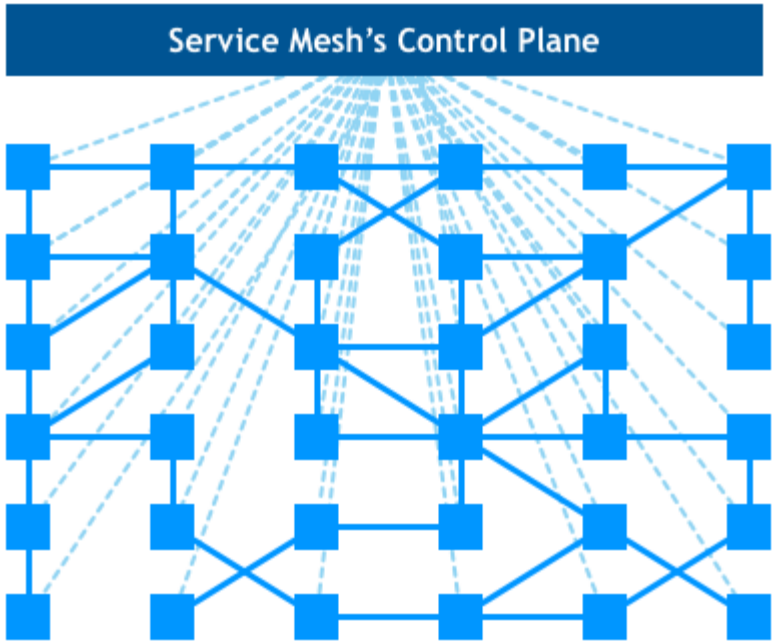
边车SideCar模式



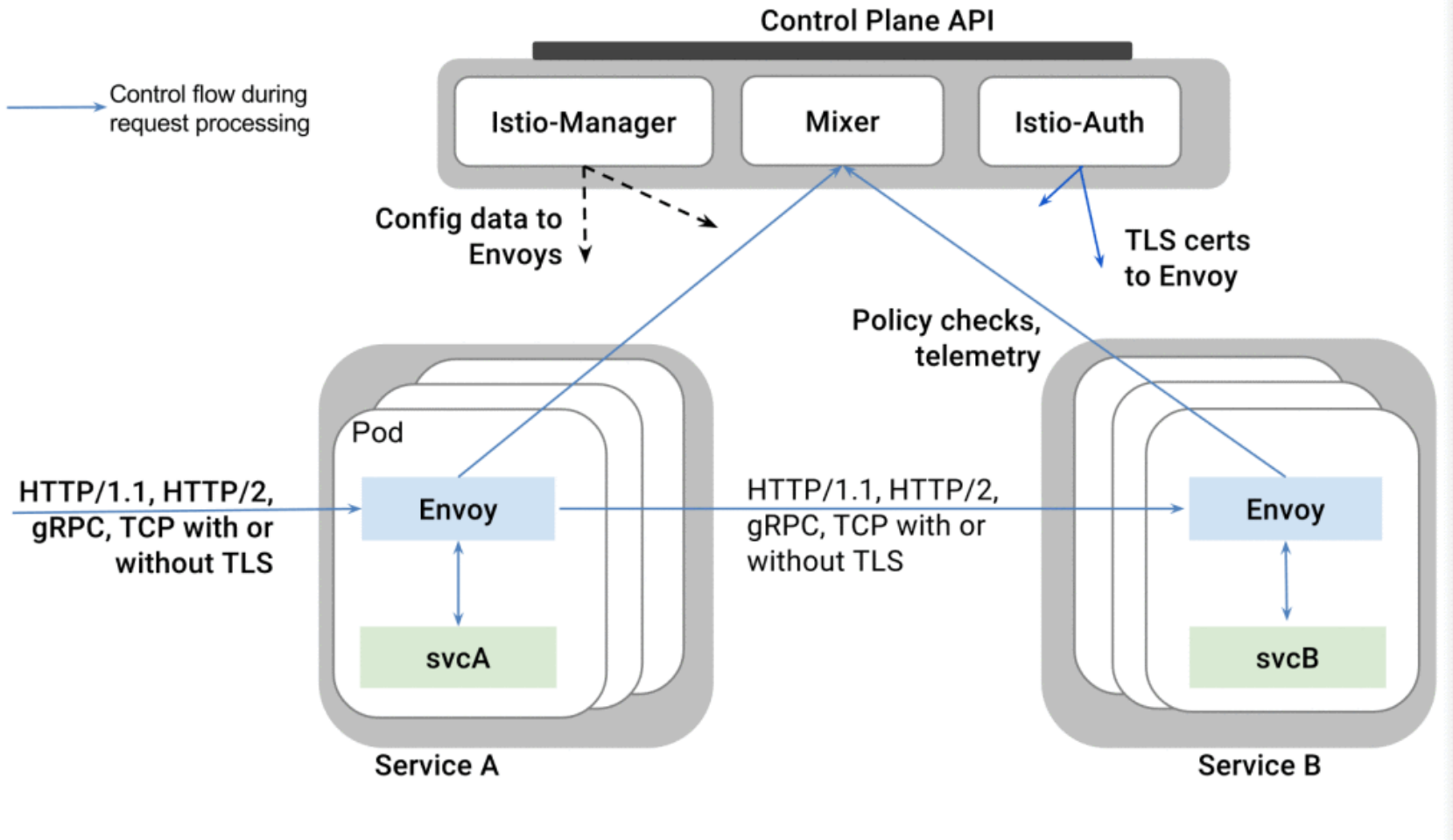
服务网格


Service Mesh's Control Plane
Computer A
Computer B
Service A
Sidecar
Sidecar
Service B
Business Logic
Circuit Breaker
Service Discovery
Circuit Breaker
Service Discovery
Business Logic
Flow Control
Networking Stack
Flow Control
Networking Stack




Service Mesh's Control Plane

Istio架构


Control Plane API
Control flow during request processing
Istio-Manager
Mixer
Istio-Auth
Config data to Envoys
TLS certs to Envoy
Policy checks, telemetry
Pod
HTTP/1.1, HTTP/2, gRPC, TCP with or without TLS
Envoy
svcA
HTTP/1.1, HTTP/2, gRPC, TCP with or without TLS
Envoy
svcB
Service A
Service B

产品

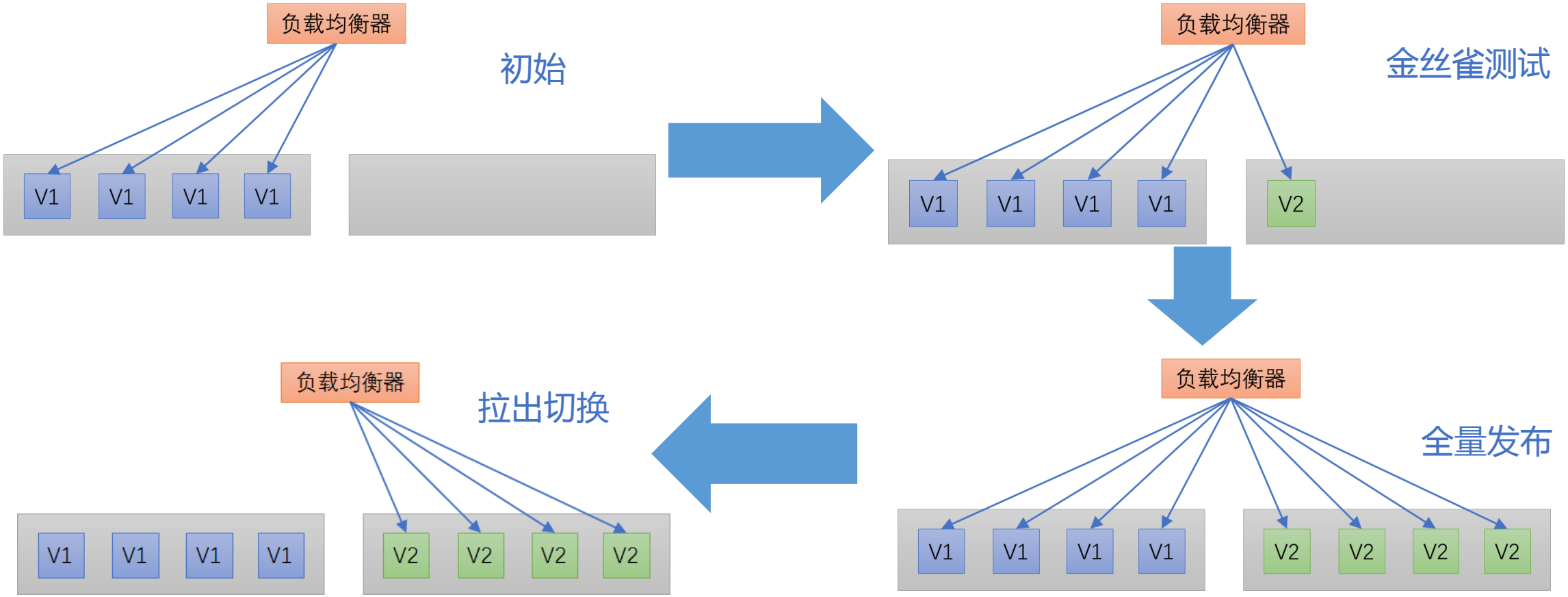
控制面板

数据面板

第13部分

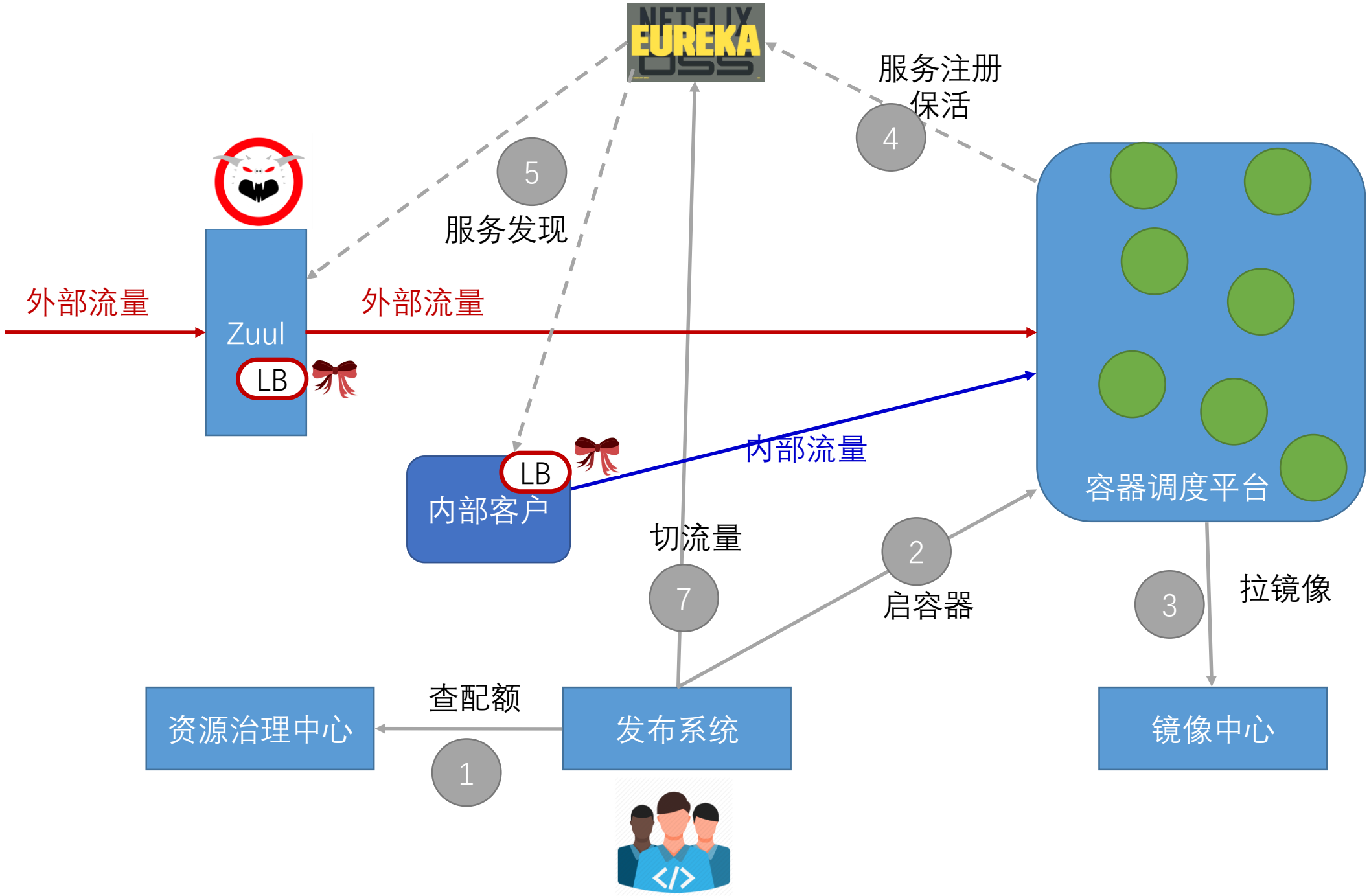
基于Eureka、Zuul和容器云的持续交付架构

发布术语：金丝雀+蓝绿部署



参考《现代发布技术的原理和实践》

https://mp.weixin.qq.com/s?__biz=MzIxMTA5NjQyMg==&mid=2647801956&idx=1&sn=742c72ab755d8b0ff8d10e1dd9f3c343&chksm=8f7c666db80bef7b17ea0f11021e1292b14350a09a0303532e7e59717a0d7f815452e7aad8ac&mpshare=1&scene=1&srcid=09257l1fwU60KnmuVuKYMm7a#rd


NETFLIX
EUREKA
服务注册
保活
4
5
服务发现
外部流量
Zuul
LB
外部流量
LB
内部客户
内部流量
容器调度平台
切流量
7
2
启容器
3
拉镜像
查配额
资源治理中心
发布系统
镜像中心
1

第14部分

参考资源和后续课程预览

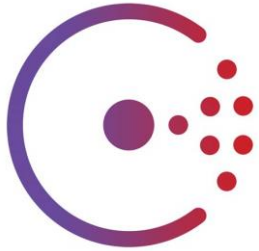
参考文章

- 深度剖析服务发现组件~Netflix Eureka
  - https://zhuanlan.zhihu.com/p/24829766
- 深入理解Ribbon之源码解析
  - https://blog.csdn.net/forezp/article/details/74820899
- Eureka at a glance
  - https://github.com/Netflix/eureka/wiki/Eureka-at-a-glance
- Announcing Ribbon: Tying the Netflix Mid-Tier Services Together
  - https://medium.com/netflix-techblog/announcing-ribbon-tying-the-netflix-mid-tier-services-together-a89346910a62

参考文章

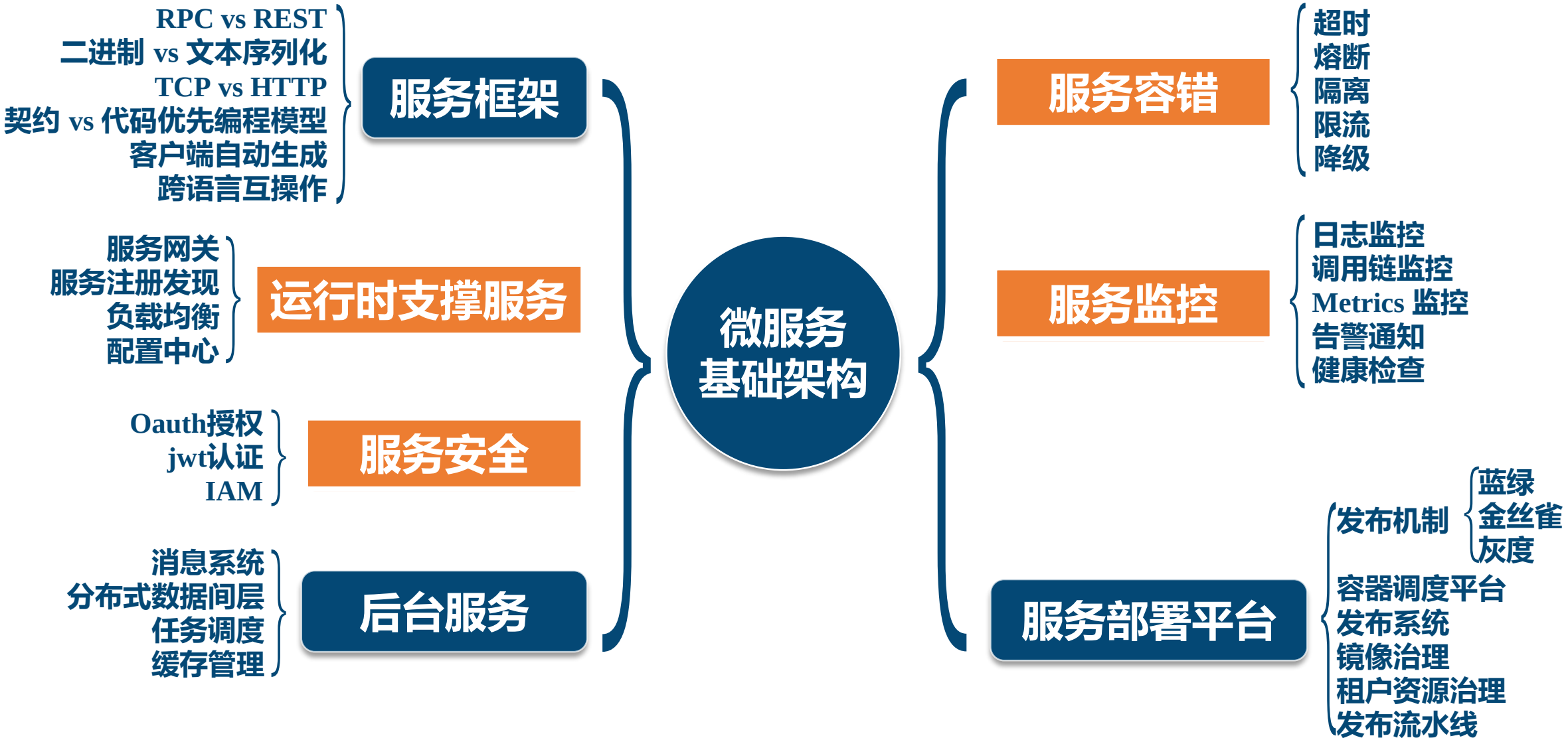
- The Mystery of Eureka Self-Preservation
  - https://medium.com/@fahimfarookme/the-mystery-of-eureka-self-preservation-c7aa0ed1b799
- The Myster of Eureka Health-Monitoring
  - https://medium.com/@fahimfarookme/the-mystery-of-eureka-health-monitoring-5305e3beb6e9
- Spring Cloud Eureka – The Hidden Manual
  - https://blog.asarkar.org/technical/netflix-eureka/

其它服务注册发现产品

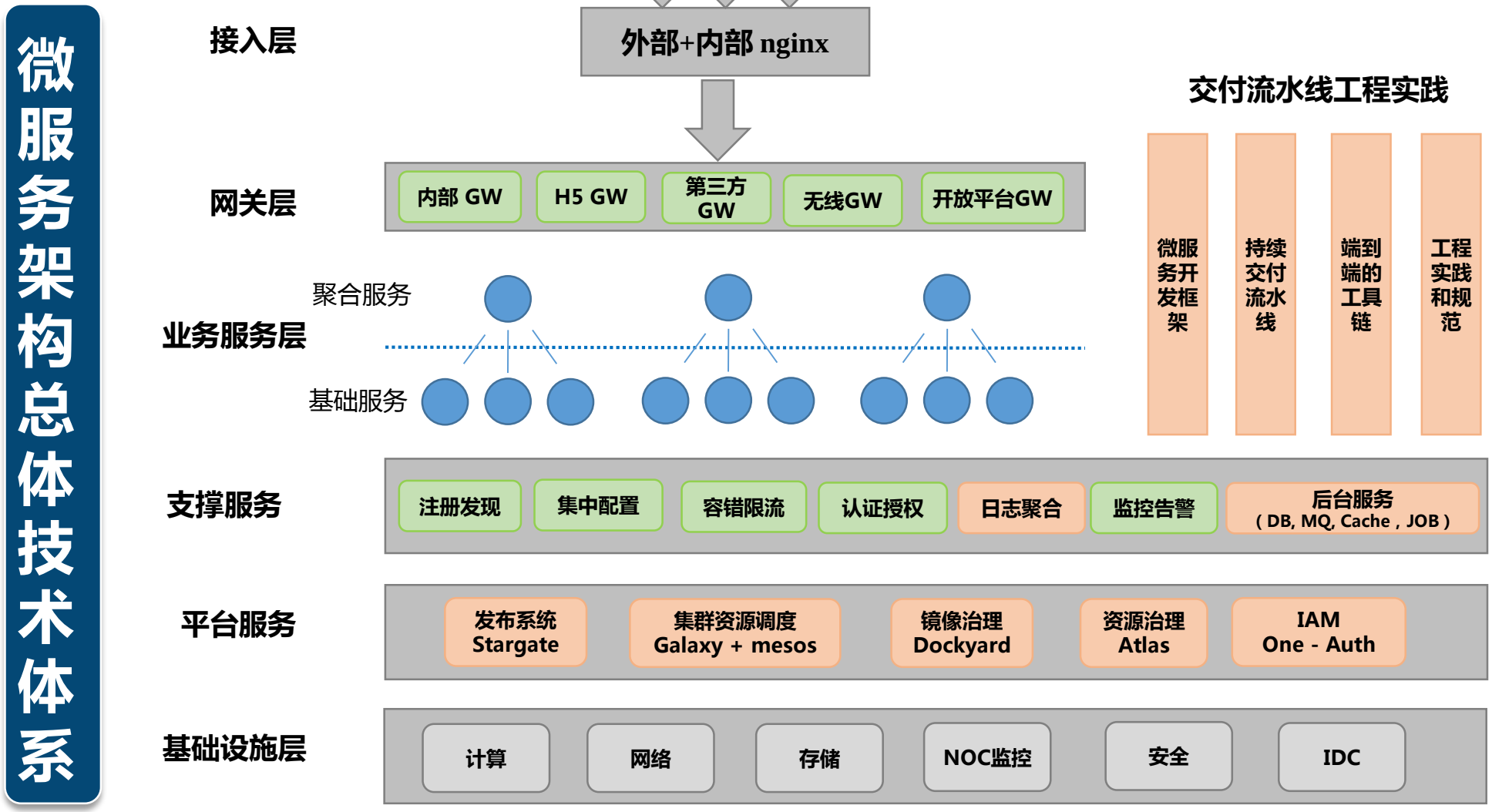


- Hashicorp Consul
  - https://www.consul.io/
- Etcd
  - https://github.com/etcd-io/etcd
- Apache Zookeeper
  - http://zookeeper.apache.org/

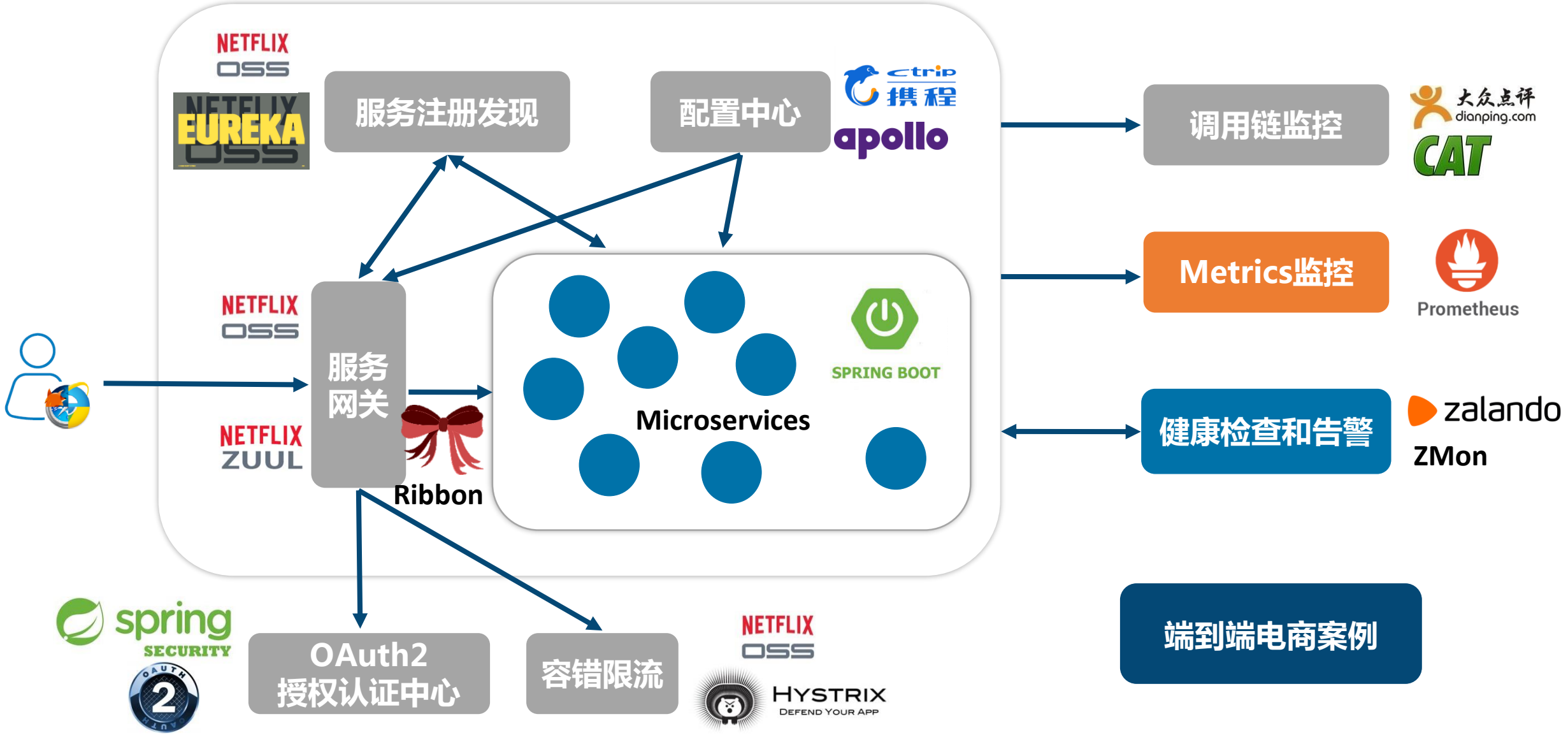
后续课程预览~2018课程模块



后续课程预览~技术体系


微服务架构总体技术体系
接入层
外部+内部 nginx
网关层
内部 GW
H5 GW
第三方 GW
无线GW
开放平台GW
聚合服务
业务服务层
基础服务
支撑服务
注册发现
集中配置
容错限流
认证授权
日志聚合
监控告警
后台服务
(DB, MQ, Cache , JOB)
平台服务
发布系统
Stargate
集群资源调度
Galaxy + mesos
镜像治理
Dockyard
资源治理
Atlas
IAM
One - Auth
基础设施层
计算
网络
存储
NOC监控
安全
IDC
交付流水线工程实践
微服务开发框架
持续交付流水线
端到端的工具链
工程实践和规范

架构和技术栈预览


NETFLIX OSS
EUREKA
服务注册发现
配置中心
ctrip 携程
apollo
调用链监控
大众点评 dianping.com
CAT
NETFLIX OSS
服务网关
Metrics监控
Prometheus
NETFLIX ZUUL
Ribbon
Microservices
SPRING BOOT
健康检查和告警
zalando
ZMon
spring SECURITY
OAuth2
授权认证中心
容错限流
NETFLIX OSS
HYSTRIX
DEFEND YOUR APP
端到端电商案例