

简单的 Service Mesh -- Flomesh

蔡书(caishu@polaristech.io)

2019-10



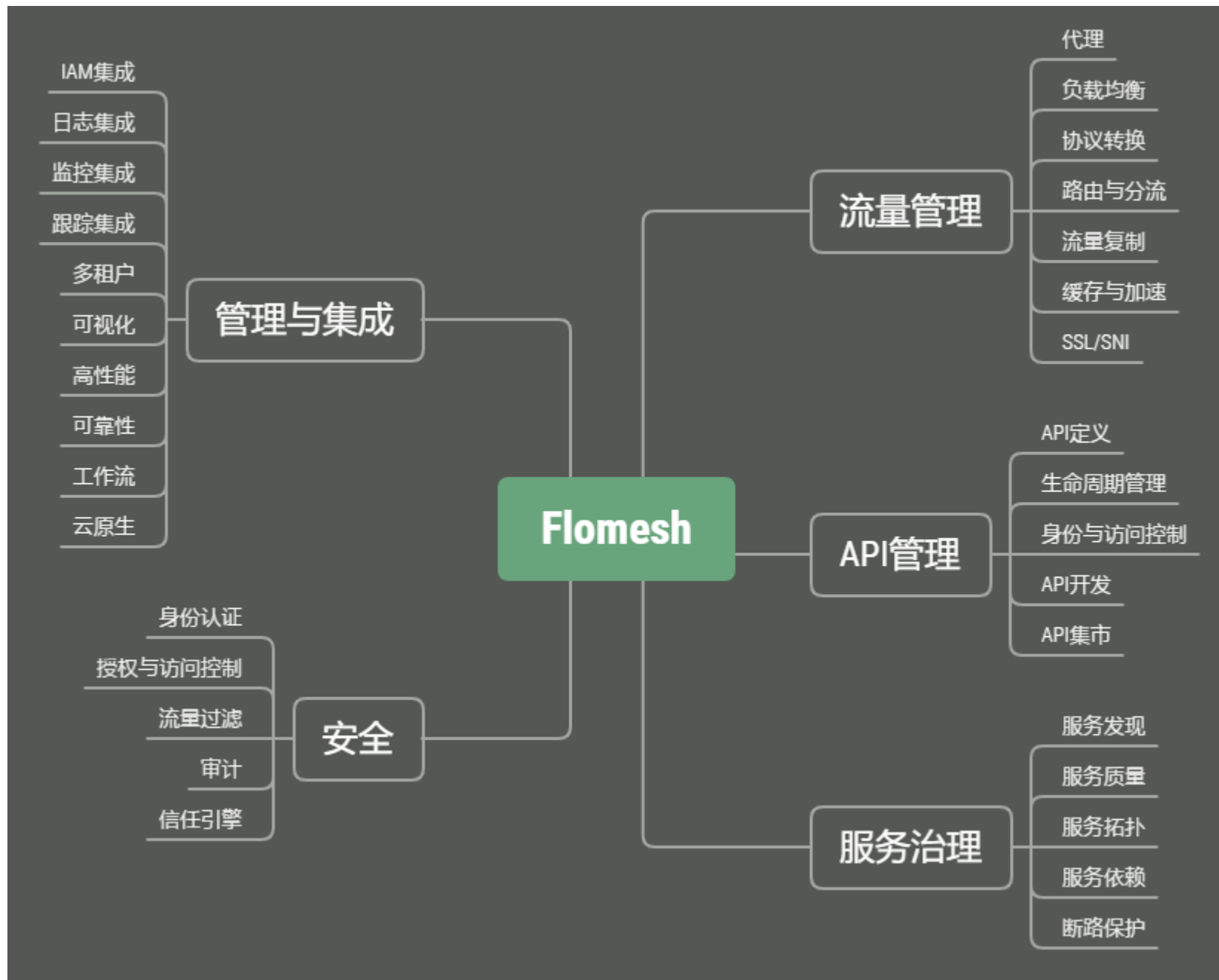
简介

大概在2017年，在实践中客户要求我们协助部署istio，用于管理该客户内部超过100个微服务之间的流量。但是在实际工作过程中，我们发现虽然istio所描述的功能是客户所期望的；但是在实现上，istio并不能充分的满足客户的需求，并且随着类似需求的客户的增多，需求的范围进一步扩大，istio的局限也更加明显，我们按照客户对istio的期望，参考istio和linkerd自己实现了一系列的功能，并最终形成Flomesh这个产品。因为istio和linkerd被认为是Service Mesh的代表，所以我们称Flomesh是Service Mesh的一个实现。

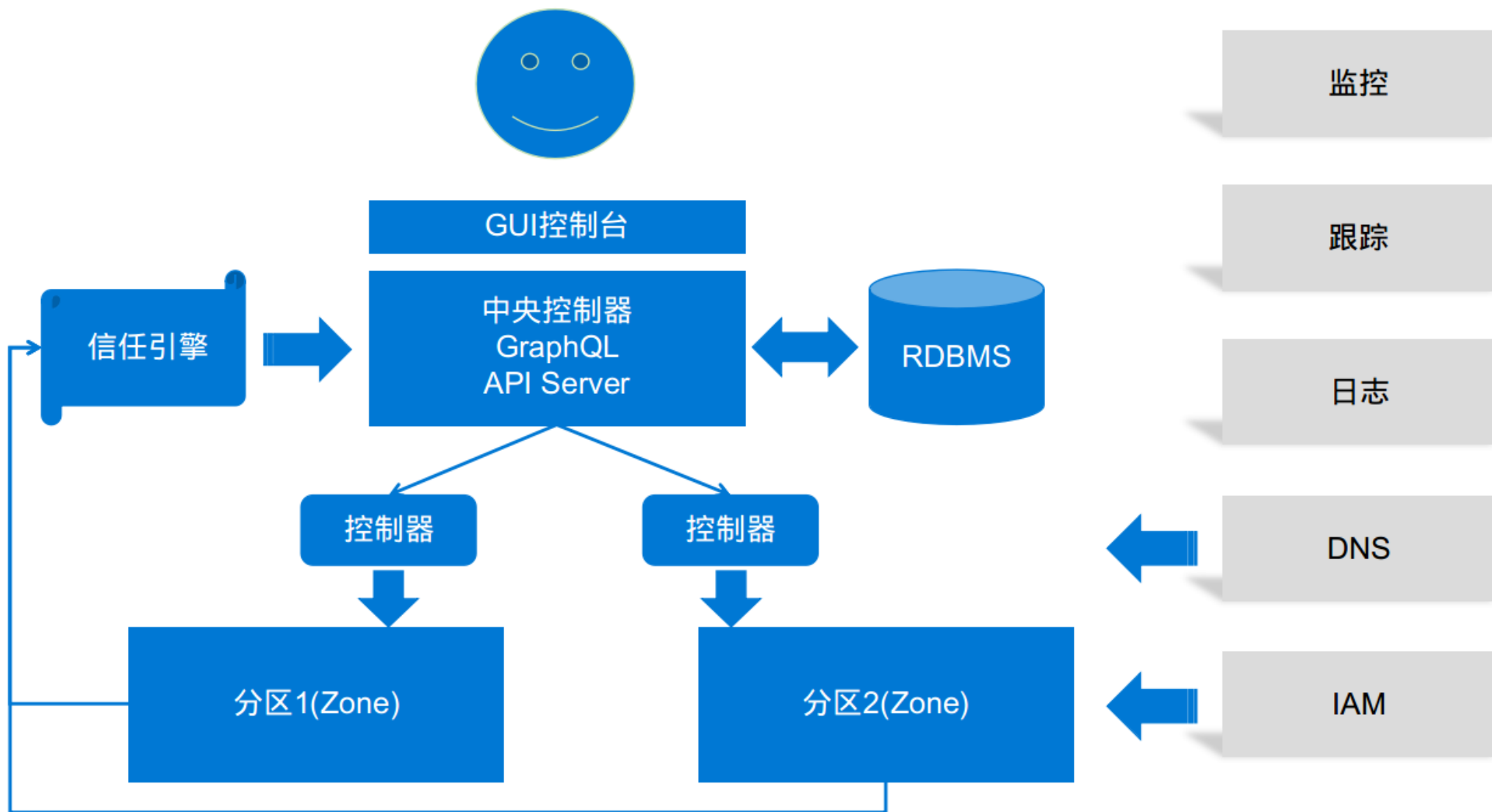
我们对于Service Mesh的定义是：Service Mesh是软件定义的应用程序网络，是一种SDN；Service Mesh包括数据平面和控制平面；通过网关的编排，Service Mesh实现了应用流量的管理。

2018年Openresty杭州meetup，我们分享了flomesh的设计思路；2019年，我们把这个设计思路产品化落地。

功能



架构



服务网格和网关

Kubernetes是“计算资源”调度；Service Mesh是“应用流量”调度。

Cloud

Kubernetes

Docker

Kernel(Cgroups/namespace/selinux)

Flomesh

Aiur

Gateway(Kong/Piped)

Nginx

对比istio和Flomesh

在线文档



对比项	Istio	Flomesh	描述
部署环境	kubernetes	容器、Kubernetes、虚拟机、物理机	
是否支持k8s多集群?	否	是	
是否支持混合多集群(k8s/vm/dubbo/spring-cloud)?	否	是	
代理模式	sidecar, 每个容器一个代理	四种: 1. sidecar, 每个容器一个代理; 2. 每个namespace/project一个代理; 3. 每个节点一个代理; 4. 全局集中代理	
支持Metrics采用的技术	Prometheus	Prometheus+内存实时统计	
支持服务依赖采用的技术	跟踪系统, 比如zipkin/jaeger	无依赖	
支持的跟踪系统	Zipkin, Jaeger	Zipkin, Jaeger, Skywalking	
是否支持Dubbo RPC流量管理?	未知	支持Dubbo RPC	
是否支持response缓存?	否	是	
支持的网关组件	Envoy	Kong, Piped	
服务认证方式	SSL	多种: HTTP BASIC; Token; JWT; HMAC	
是否支持WAF(owasp top 10)流量过滤?	未知	支持	
控制平面开发语言	Go	Type Script	
控制平面接口	gRPC, REST	graphql, REST	
数据平面开发语言	Envoy(C++)	Kong(OpenResty), Piped(C++)	
数据平面接口	gRPC	REST	
支持的微服务框架	kubernetes	kubernetes; SpringCloud/Eureka; Dubbo/Zookeeper	可以自动生成服务注册信息, 维护动态路由信息
灰度发布/金丝雀发布	支持	可视化	
HTTP正向代理	未知	支持	
配置导入/导出	未知	支持	导入导出Json或者Yaml
支持windows环境?	未知	支持	包括运行在windows里的服务; 以及组件是否可以部署在windows中
多租户/分角色/分权限	未知	支持	
配置变更审批与 workflows	未知	支持	
内置管理控制台图形界面?	未知	支持	
多环境统一管理 (Dev/Test/Stage, etc)	未知	支持	
跨DC、跨AZ、跨Region 部署与流量转发	未知	支持	比如支持两地三中心多活高可用部署
流量管理扩展	未知	插件, 以及lua在线编辑在线生效	

演示

反向代理与负载均衡

1 设计 2 审核 3 运行 4 停止

审核

配置的分區 +

访问端点
方法 (多个) (none)
协议 http, https
正则匹配优先级 0
选项
主机名称 (多个)
路径 (多个)
/ok
添加访问端点

超时
链接 60000
读 60000
写 60000
认证
方法
IP 地址限制
启用 No
限流
方法
响应缓存
启用 No
金丝雀发布
类型
地理位置 IP
启用 No
跟踪
类型
流量复制

服务提供者 (主机)
名称 反向代理演示
协议 http
主机名称 flomesh.cn
端口 80
添加服务提供者

Flomesh 项目 分区 管理 A

更改名称

spring

1 个代理 0 个负载均衡 0 个发现

最近 5 分钟 最近 15 分钟 最近 1 小时 最近 6 小时 最近 1 天 最近 7 天 最近 30 天

每秒请求次数

requests/sec

20 18 16 14 12 10 8 6 4 2 0

0:46 AM 0:47 AM 0:48 AM 0:49 AM 0:50 AM

Error All

数据流量

6.0K 5.5K 5.0K 4.5K 4.0K 3.5K 3.0K

Egress Ingress

访问控制与流控

方法 (多个) (none) 链接 60000 名称 bookinfo-ratings

编辑 API

基础设置

超时

认证

IP 地址限制

限流

响应缓存

金丝雀发布

地理位置 IP

跟踪

流量复制

脚本

特定分区设置 [选择分区](#) ▼

选择一个分区以显示其特定设置

方法

Basic

Limit By

IP

Policy

Local

Second

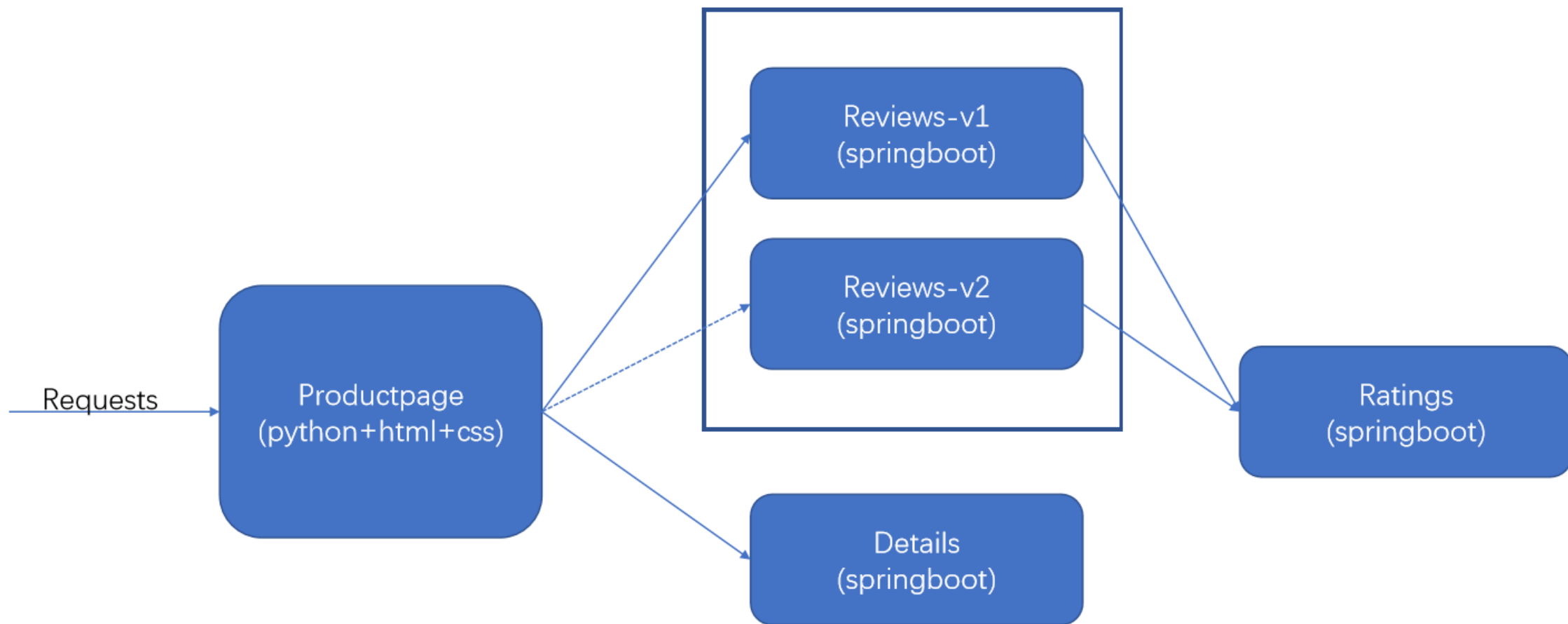
100

Minute

0

Hour

Spring微服务



灰度发布

方法 (多个) (none) 链接 00000 名称 bookinfo-ratings

编辑 API

超时 基础设置 特定分区设置 [选择分区](#) ▼

认证 类型 选择一个分区以显示其特定设置

IP 地址限制 ▼

限流 Variable Name

响应缓存 Hash ▼

金丝雀发布 | Steps ▼

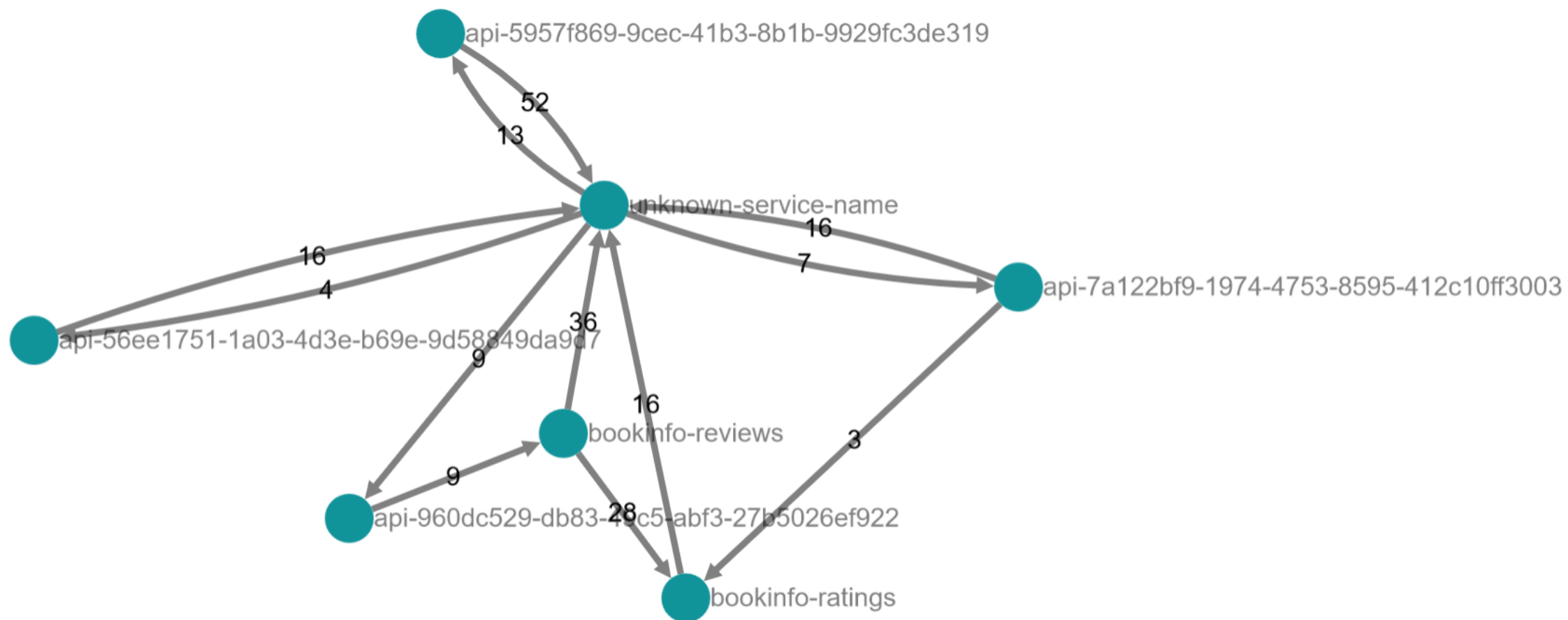
地理位置 IP Start

跟踪 Duration

流量复制

脚本

服务拓扑与分布式跟踪



网关Perf测试

=====

[flomesh]

Name=flomesh

[Baseurl=http://repo.flomesh.cn](http://repo.flomesh.cn)

Enabled=1

Gpgcheck=0

=====

Yum -y install piped

Cp /etc/piped/test/echo/config /etc/piped/

Systemctl start piped

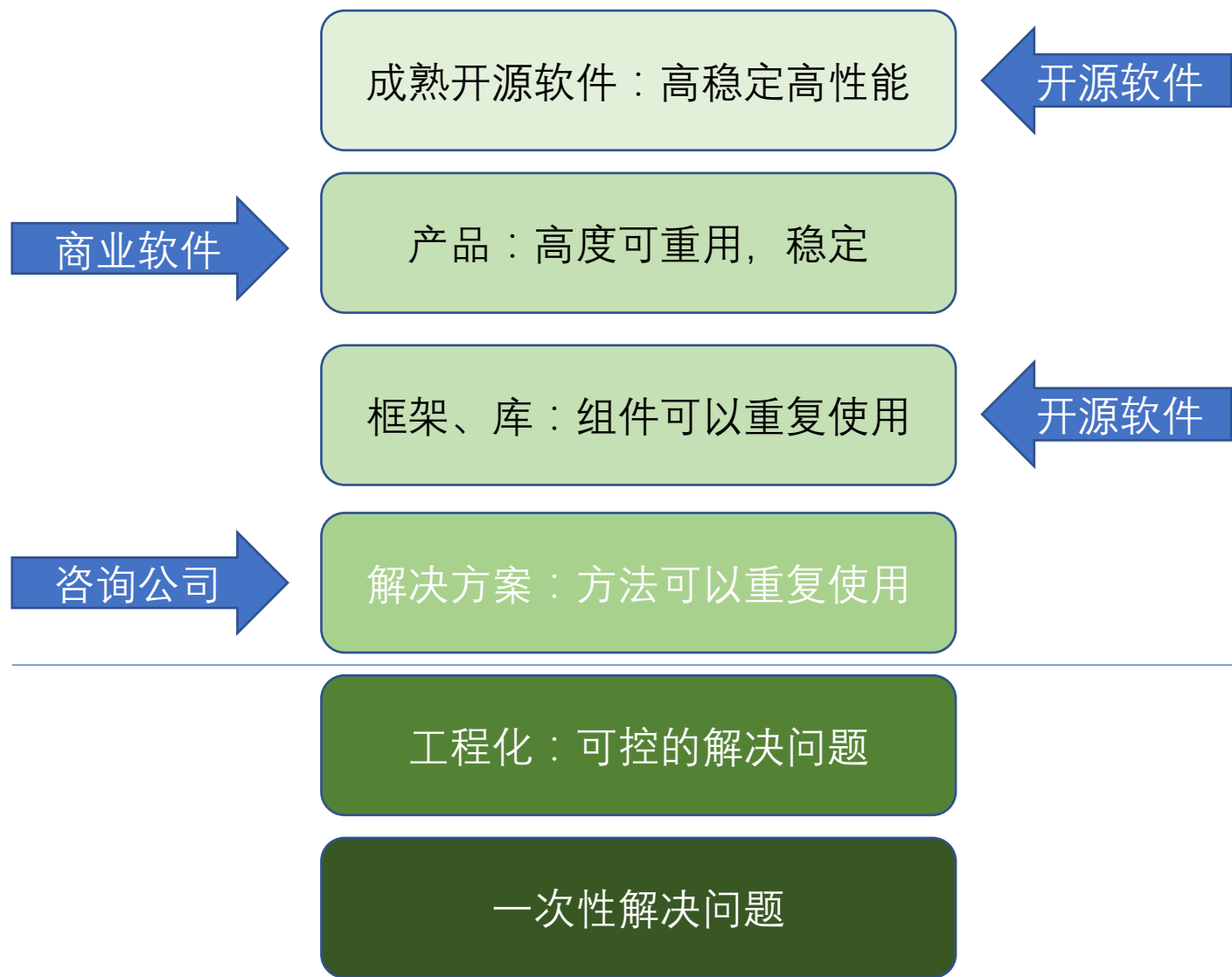
- Nginx : Ab -c 10 -n 100000 <http://localhost/>
- Piped : Ab -c 10 -n 100000 <http://localhost:8000/>

- Nginx : wrk -t10 -c100 -d10s <http://localhost/>
- Piped : wrk -t10 -c100 -d10s <http://localhost:8000/>

Q&A



关于开源



- ✓ 开源软件在框架和库这层，有领先的优势：一方面这个环节不需要太多的人员，有几个“大牛”就可以；另一方面，这个领域的商业回报不高
- ✓ 成熟开源软件非常难于形成，需要团队的技术能力、推广能力、商业认可等多因素。凤毛麟角
- ✓ 每个层级的跃迁都需要专业人才的反复试错，以及高昂的成本：金钱、时间、以及试错失败带来的挫败感