
KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023



KubeCon | CloudNativeCon


OPEN SOURCE SUMMIT

China 2023

使用eBPF进行跨集群流量编排

张晓辉　Senior Architect，Evangelist，Flomesh

常　震　Software Engineer，Huawei Cloud

个人介绍

张晓辉

资深程序员，LFAPAC 开源布道师，CNCF Ambassador，微软 MVP，公众号“云原生指北”作者。

Flomesh 高级云原生架构师/布道师

常　震

Karmada 社区维护者

华为云软件工程师

引 言

云原生应用的现状和挑战

单集群规模受限

- 节点不超过5000
- Pod不超过15万
- 容器不超过30万
- 单节点不超过110 Pod

高可用部署需求

- 避免单点故障
- 两地三中心要求
- 服务弹性流量

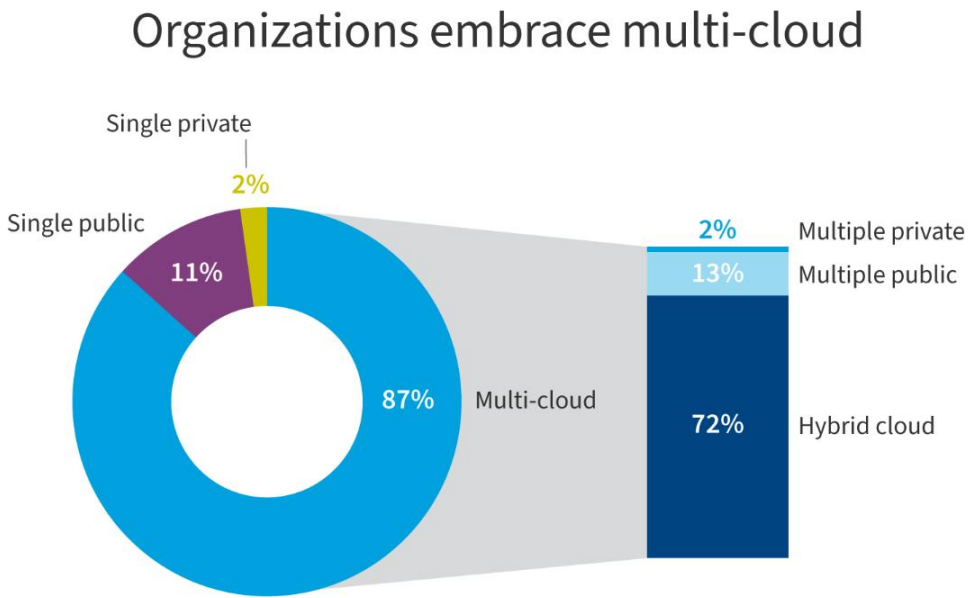
多云架构使然

- 属地化部署
- IDC+公有云弹性
- 避免厂商绑定
- 降本增效

业务场景隔离

- 业务隔离
- 团队隔离
- 开发流程隔离

多云多集群环境的崛起



N=750
Source: Flexera 2023 State of the Cloud Report

超过87%的企业受访者同时使用多个云服务商的服务。

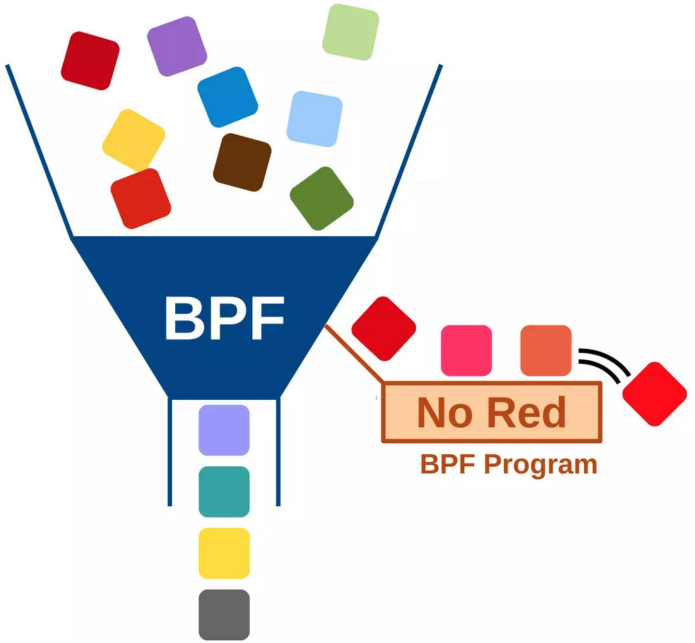
云原生技术和云市场不断成熟，未来将是编程式多云管理服务的时代。

eBPF 技术简介

Berkeley Packet Filter

来自于 1992 年的论文《The BSD Packet Filter: A New Architecture for User-level Packet Capture》

发明之初是用做网络包的过滤器，*tcpdump* 。



eBPF = extended Berkeley Packet Filter

Dynamically program the kernel for efficient networking, observability, tracing, and security.

- 稳定（DAG、可达性）
- 高效（JIT 本地机器码）
- 安全（verifier，有限的辅助函数）
- 热加载/卸载（无需重启）

内核可编程

Agenda

云原生多集群环境的趋势与需求

eBPF 在云原生流量处理中的角色

在多集群场景中应用和流量调度的实践与展望

结论与 **Q&A**

云原生多集群环境的趋势与需求

云原生多云多集群的发展趋势

一群孤岛

- 一致的集群运维
- 一致的应用交付
- 业务割裂，互不感知
- 数据孤岛、资源孤岛、流量孤岛

威尼斯水城

- 统一应用交付（部署运维）
- 统一应用访问（流量分发）
- 统一资源分配（编排调度）
- 少量、小压力的跨集群业务访问

We are here

大航海时代

实例、数据、流量：

- 自动调度
- 自由伸缩
- 自由迁移

云原生的多云充满挑战

多云容器集群管理的挑战

集群繁多

繁琐重复的集群配置

云厂商的集群管理差异

碎片化的API访问入口

业务分散

应用在各集群的差异化配置

业务跨云访问

集群间的应用同步

集群的边界限制

资源调度受限于集群

应用可用性受限于集群

弹性伸缩受限于集群

厂商绑定

业务部署的“黏性”

缺少自动的故障迁移

缺少中立的开源多集群编排项目

Karmada：开源的云原生多云容器编排平台

使用Karmada构建无限可扩展的容器资源池

让开发者像使用一个K8s集群一样使用多云

K8s原生API兼容

零改造从单集群升级为多集群
无缝集成K8s单集群工具链生态

开放中立

来自互联网、金融、制造业、
运营商、云厂商等联合发起

告别绑定

多云平台支持，自动分配，自由迁移
不绑定厂商的商业产品

开箱即用

面向多场景的内置策略集：
两地三中心、同城双活、异地容灾

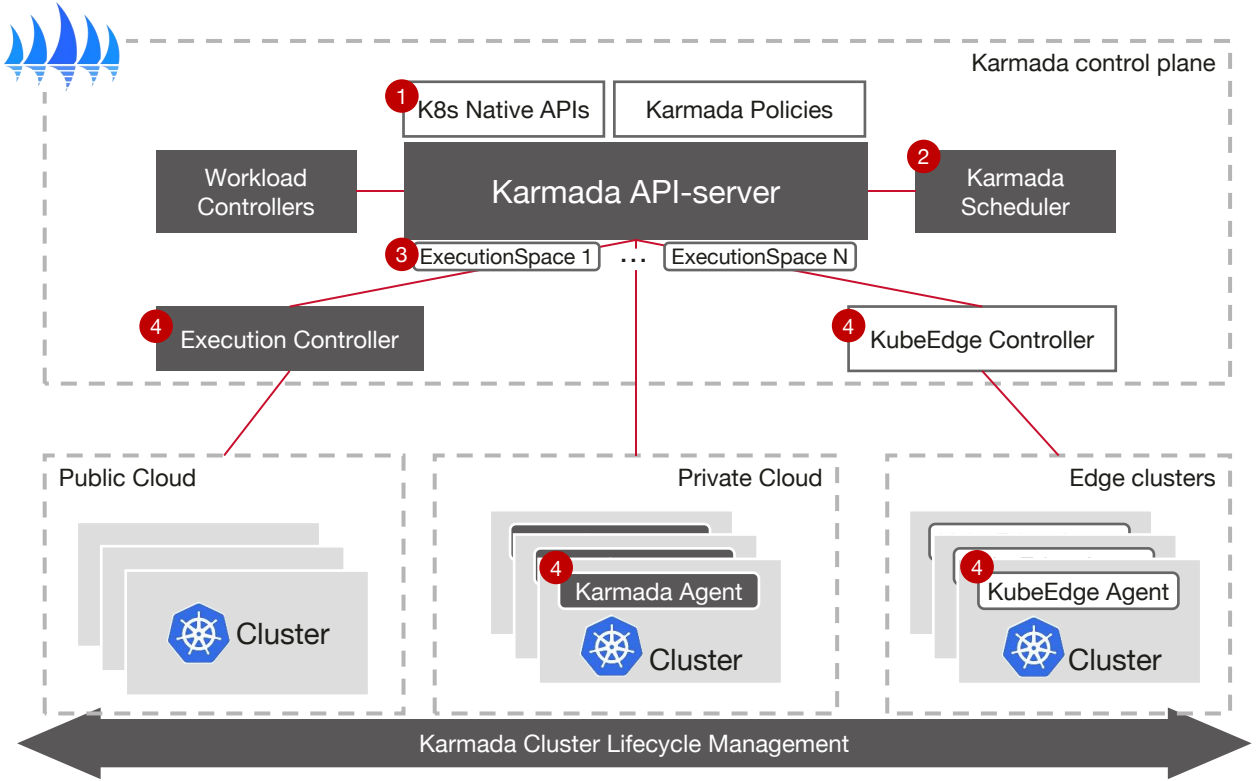
丰富的多集群调度

集群亲和性调度，多颗粒多集群高可用部署：
多Region、多AZ、多集群、多供应商

集中式管理

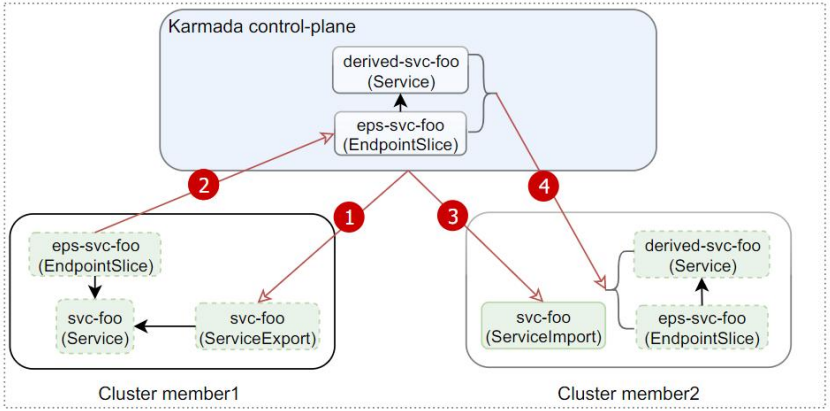
无需顾虑集群位置
支持公有云、私有云、边缘的集群

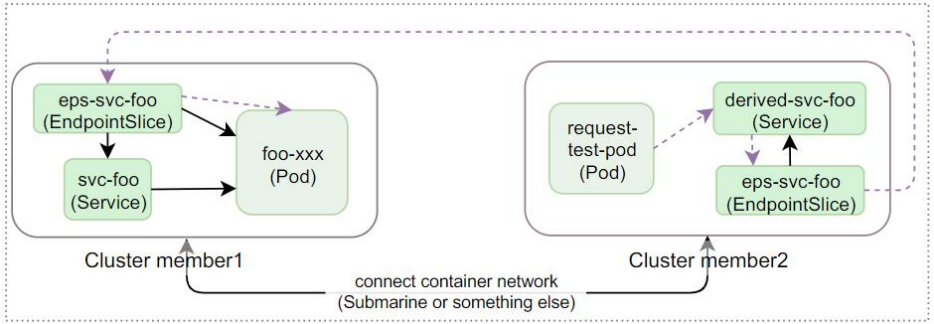
Karmada 架构


Karmada control plane
1
K8s Native APIs
Karmada Policies
Workload Controllers
Karmada API-server
2
Karmada Scheduler
3
ExecutionSpace 1
...
ExecutionSpace N
4
Execution Controller
4
KubeEdge Controller
Public Cloud
Cluster
Private Cloud
4
Karmada Agent
Cluster
Edge clusters
4
KubeEdge Agent
Cluster
Karmada Cluster Lifecycle Management

Karmada Adopters

多集群服务发现方案

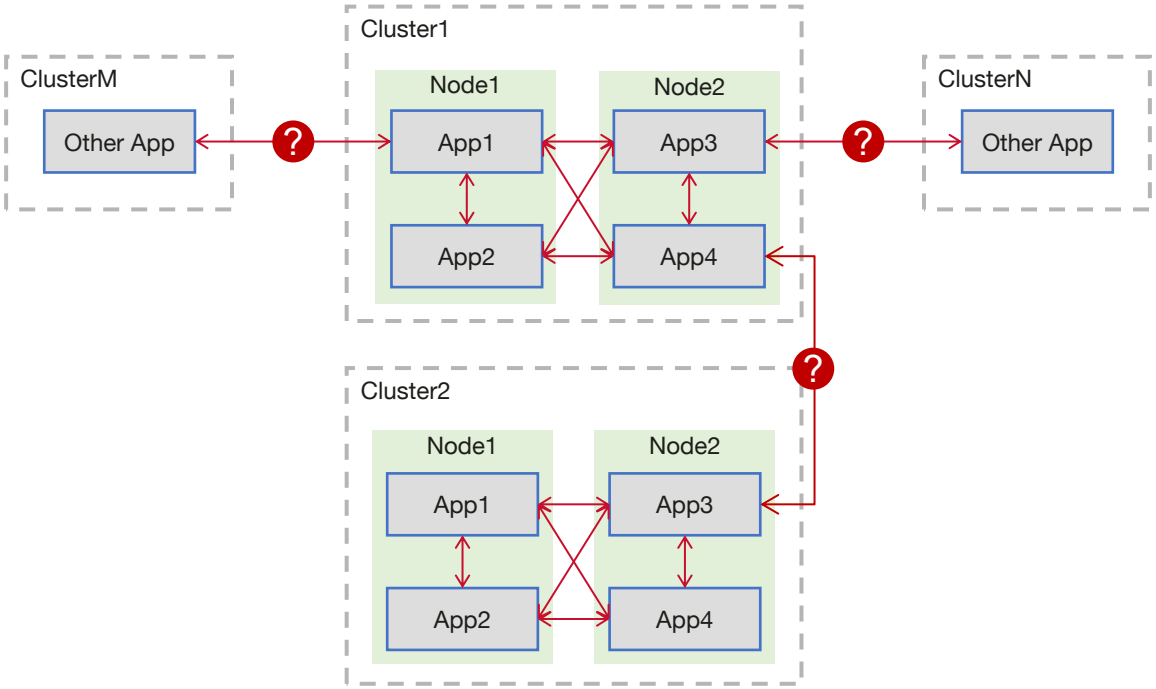

Karmada control-plane
derived-svc-foo
(Service)
eps-svc-foo
(EndpointSlice)
2
1
3
4
eps-svc-foo
(EndpointSlice)
svc-foo
(Service)
svc-foo
(ServiceExport)
svc-foo
(ServiceImport)
derived-svc-foo
(Service)
eps-svc-foo
(EndpointSlice)
Cluster member1
Cluster member2


eps-svc-foo
(EndpointSlice)
svc-foo
(Service)
foo-xxx
(Pod)
request-
test-pod
(Pod)
derived-svc-foo
(Service)
eps-svc-foo
(EndpointSlice)
Cluster member1
Cluster member2
connect container network
(Submarine or something else)

如何实现跨集群通信

1. 如果跨集群的容器网络没有打通呢？
2. 除了 mcs-api 之外，有没有其他方式实现跨集群服务发现？
3. …

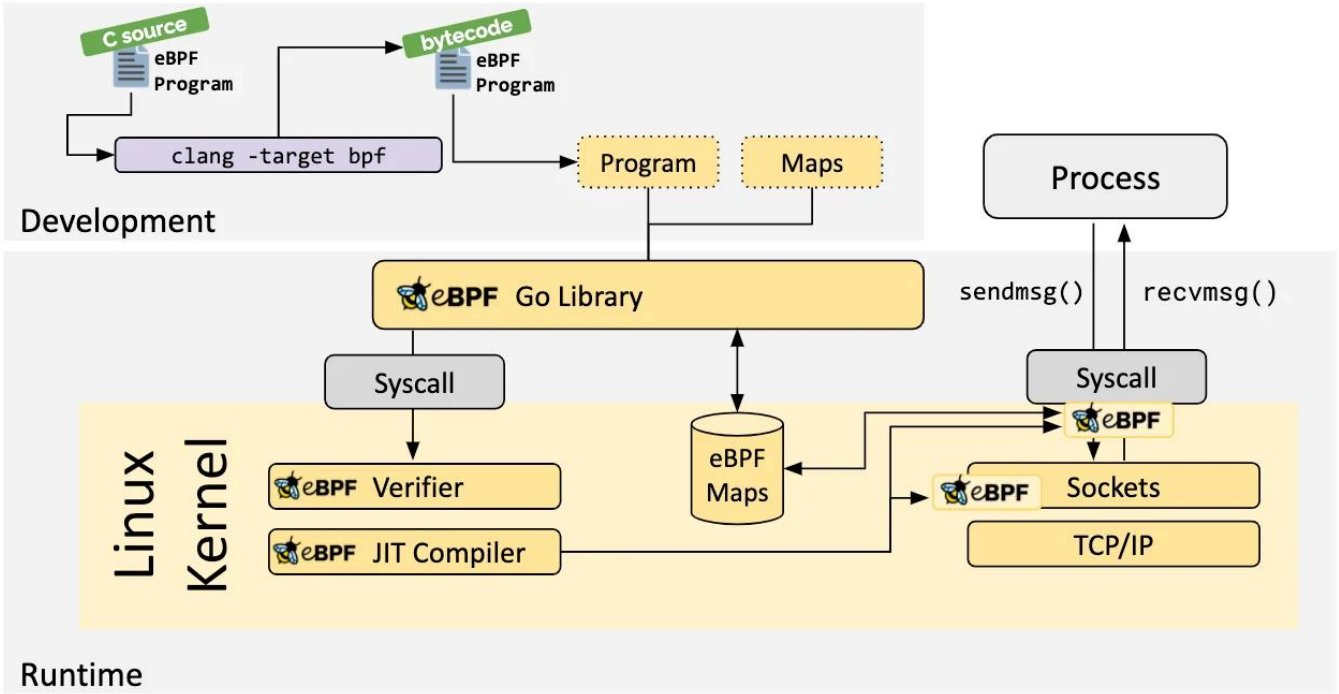
孤岛网络模型



孤岛网络模型 vs 平面网络模型

eBPF 在云原生流量处理中的角色

eBPF 加载器与验证器

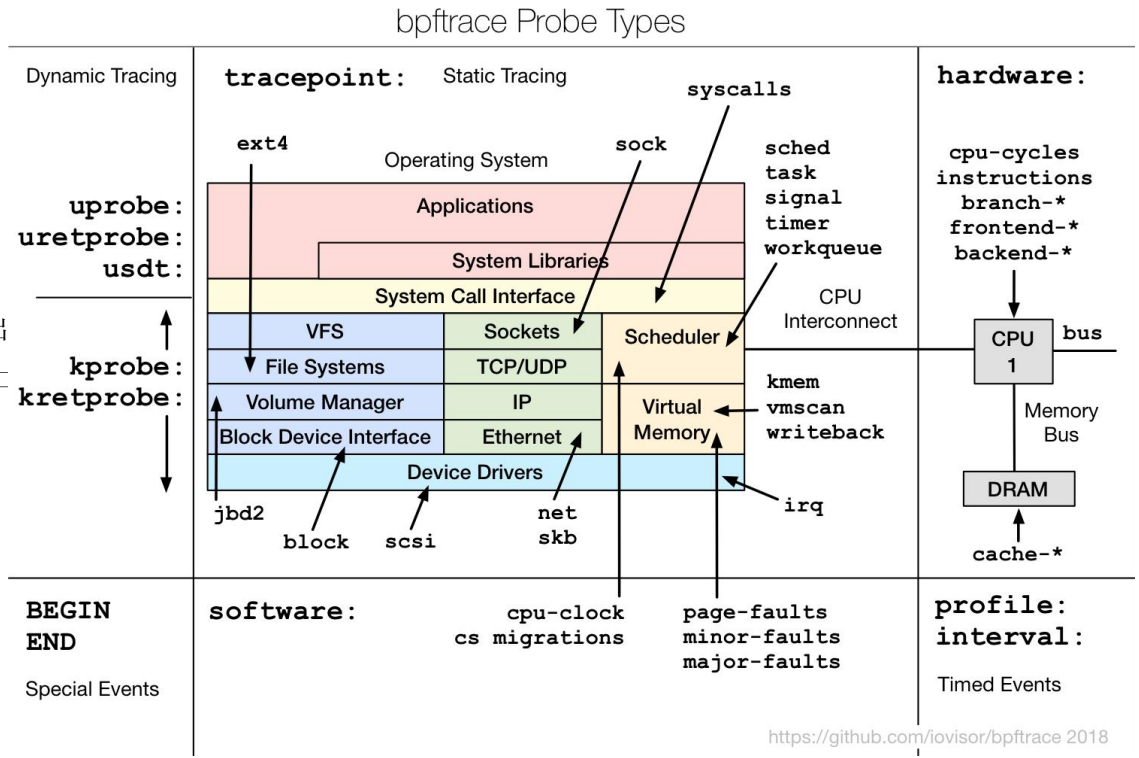


eBPF 事件驱动

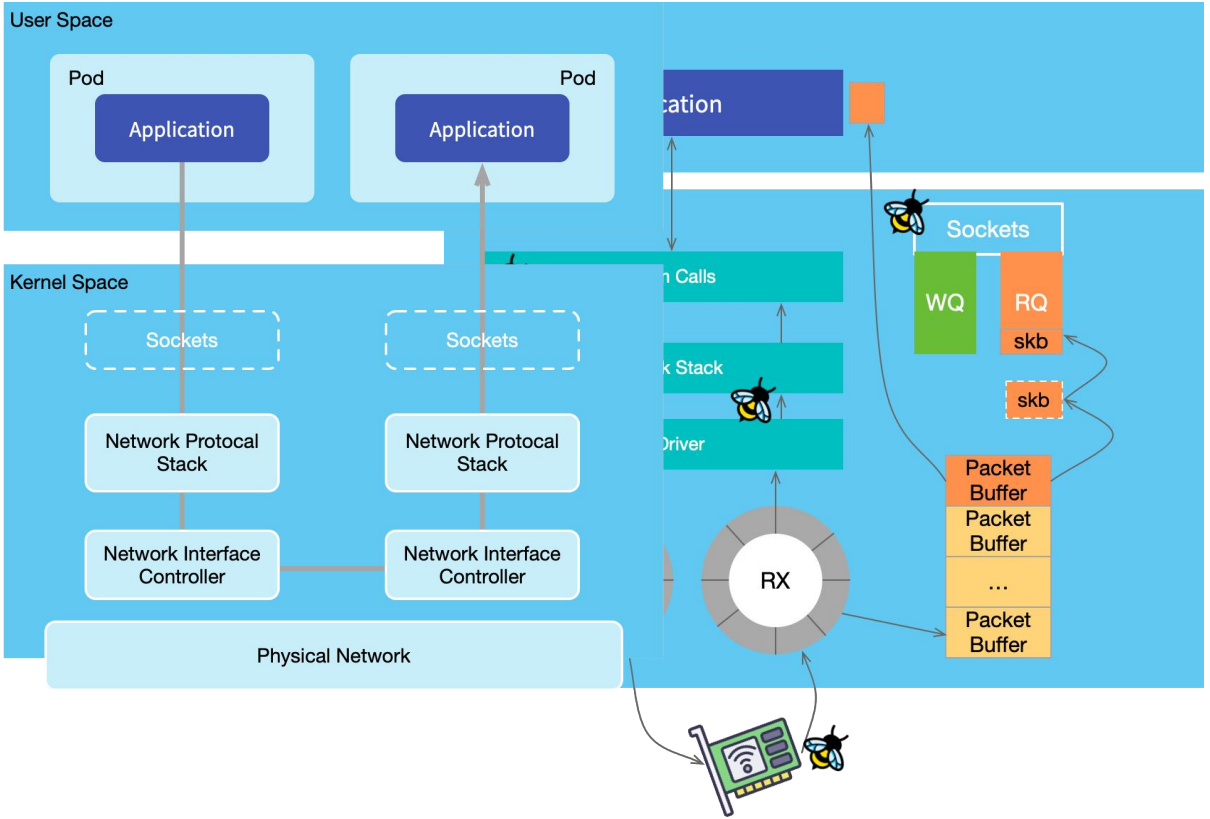
Event -> Action

事件：

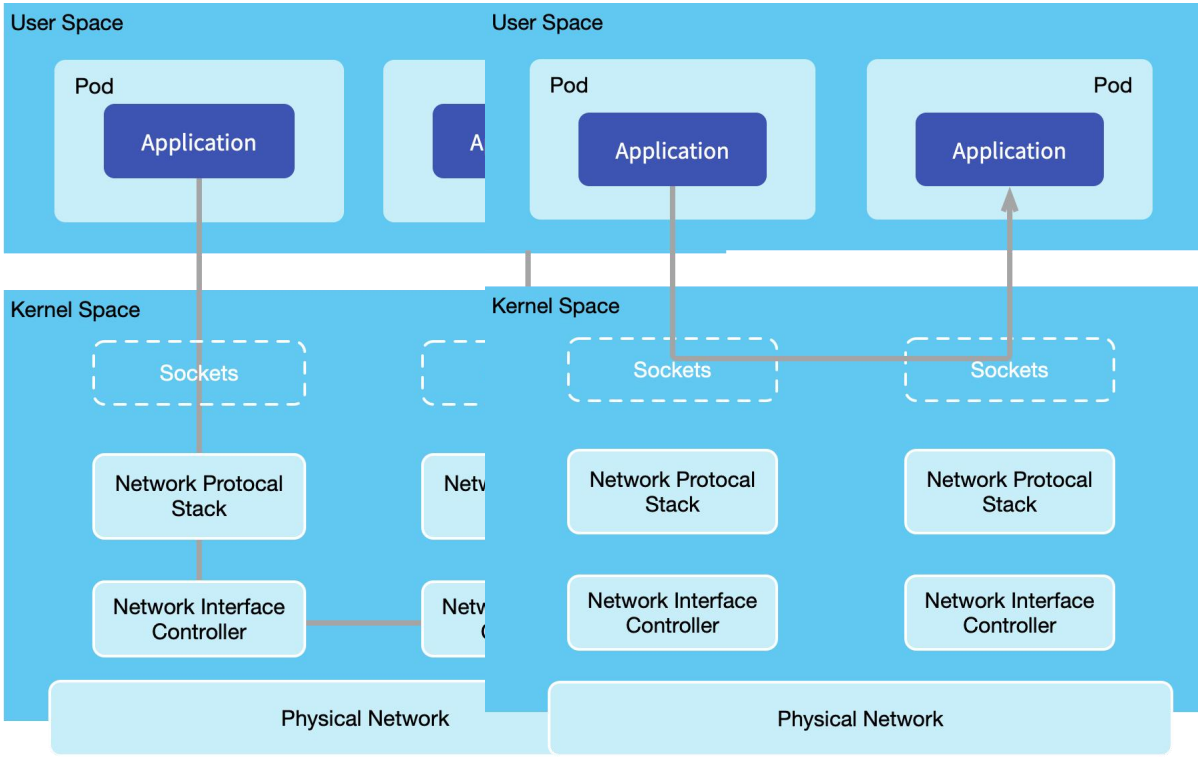
- Kprobe/Kretprobe（Kernel 函数入口和出
- Uprobe/Uretprobe（User 函数入口和出口
- XDP（eXpress Data Path）
- Tracepoint（特定事件时触发）
- Perf（性能事件，如 CPU 周期计数）



Pod 间的网络通信

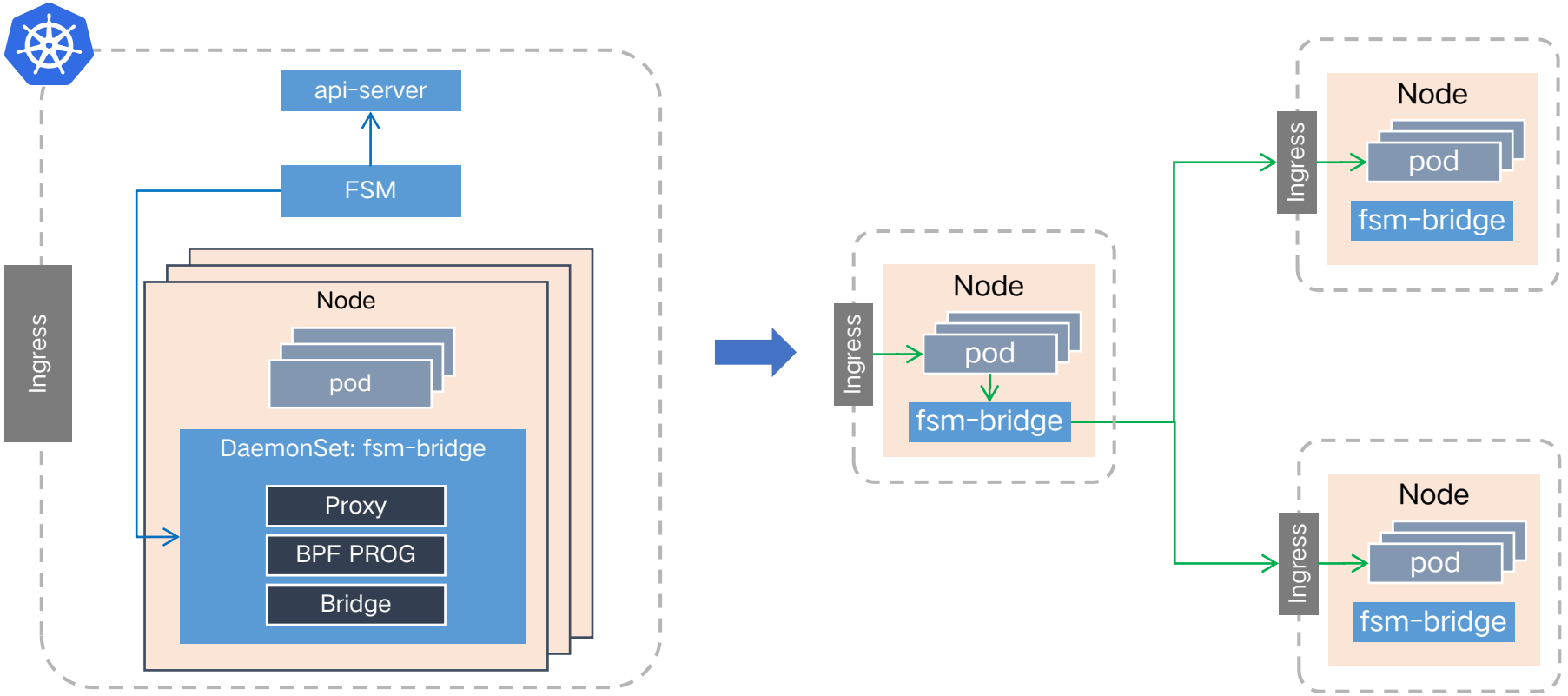

User Space
Pod
Application
Pod
Application
Kernel Space
Sockets
Sockets
Network Protocal Stack
Network Protocal Stack
Network Interface Controller
Network Interface Controller
Physical Network
Sockets
WQ
RQ
skb
skb
Packet Buffer
Packet Buffer
...
Packet Buffer
RX

网络 NAT

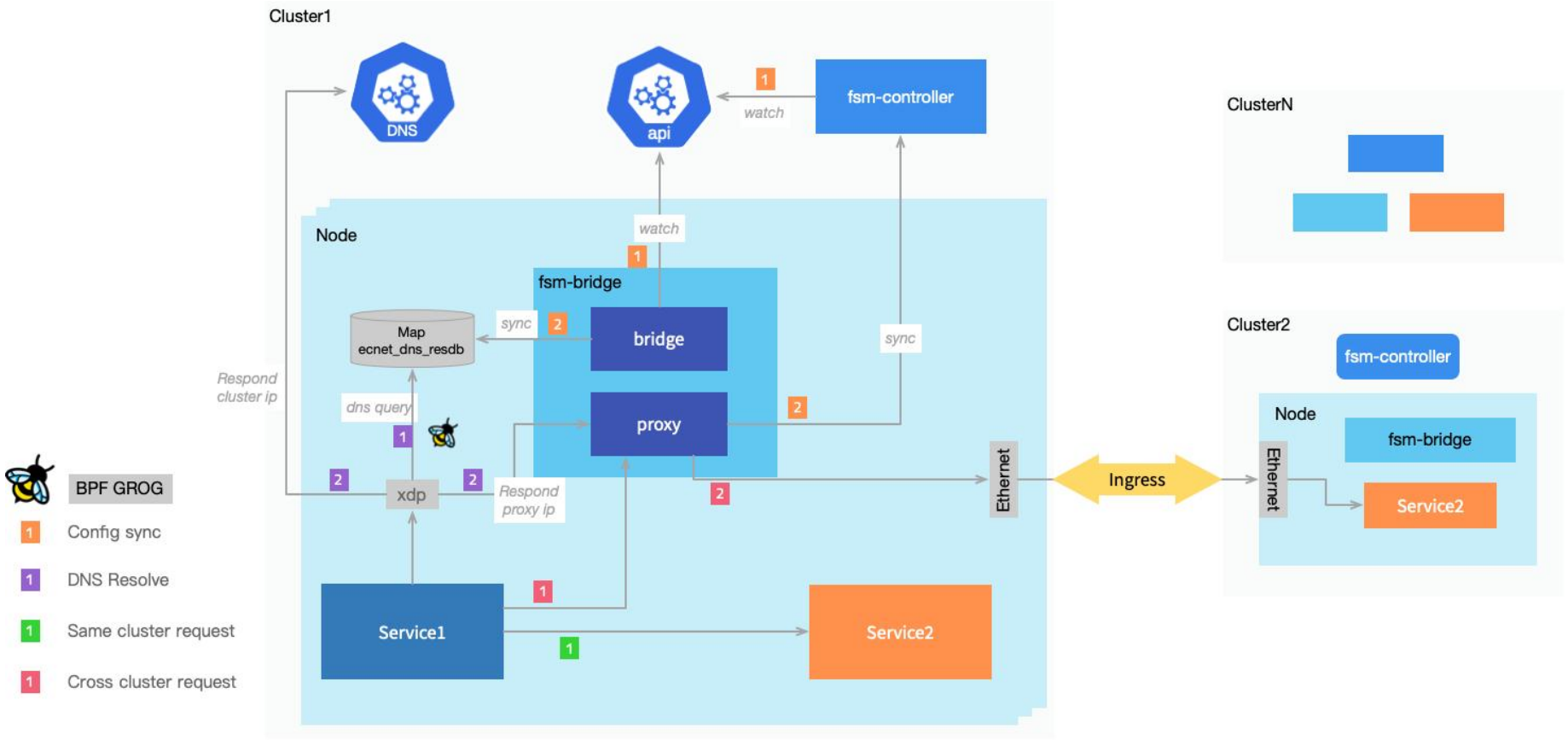

User Space
User Space
Pod
Application
Pod
Application
Pod
Application
Kernel Space
Kernel Space
Sockets
Sockets
Sockets
Network Protocal Stack
Network Protocal Stack
Network Protocal Stack
Network Interface Controller
Network Interface Controller
Network Interface Controller
Physical Network
Physical Network

多集群场景中应用和流量调度的实践

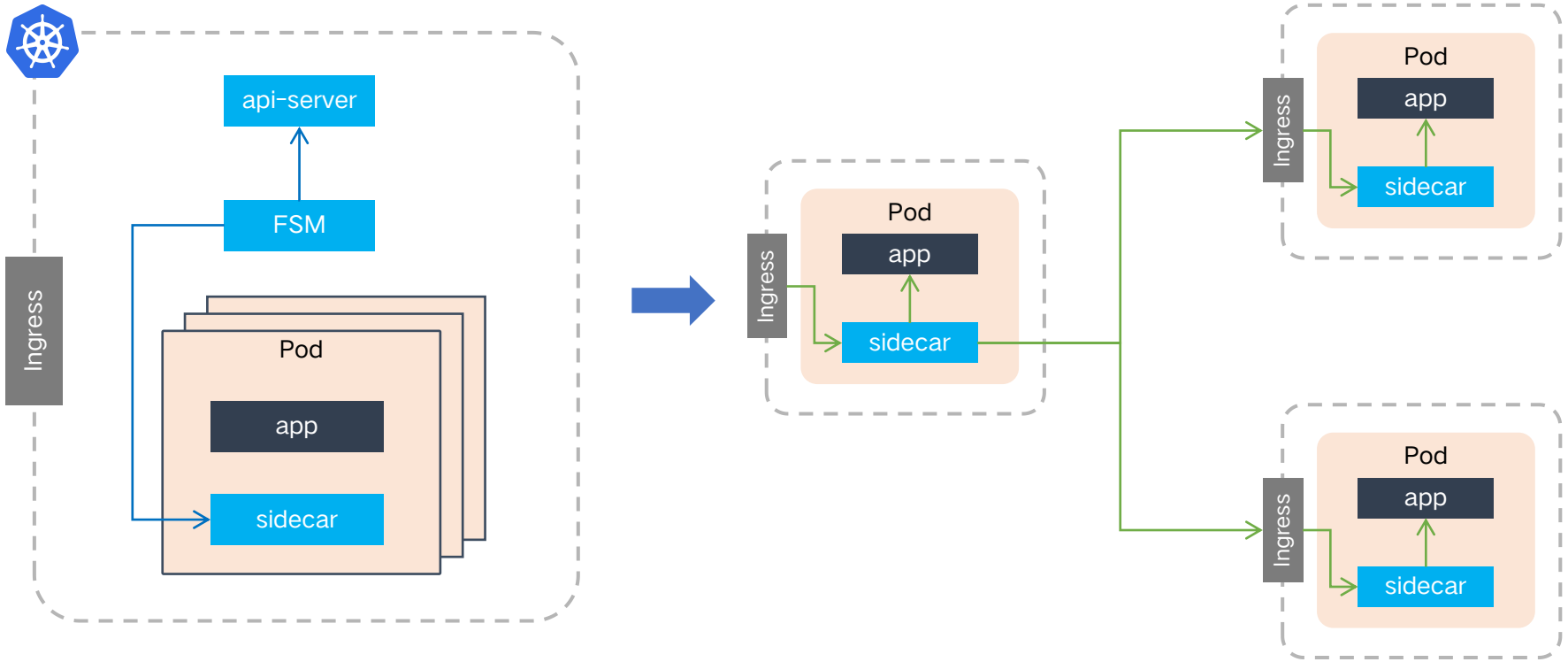
使用 eBPF 进行流量调度



使用 eBPF 进行流量调度



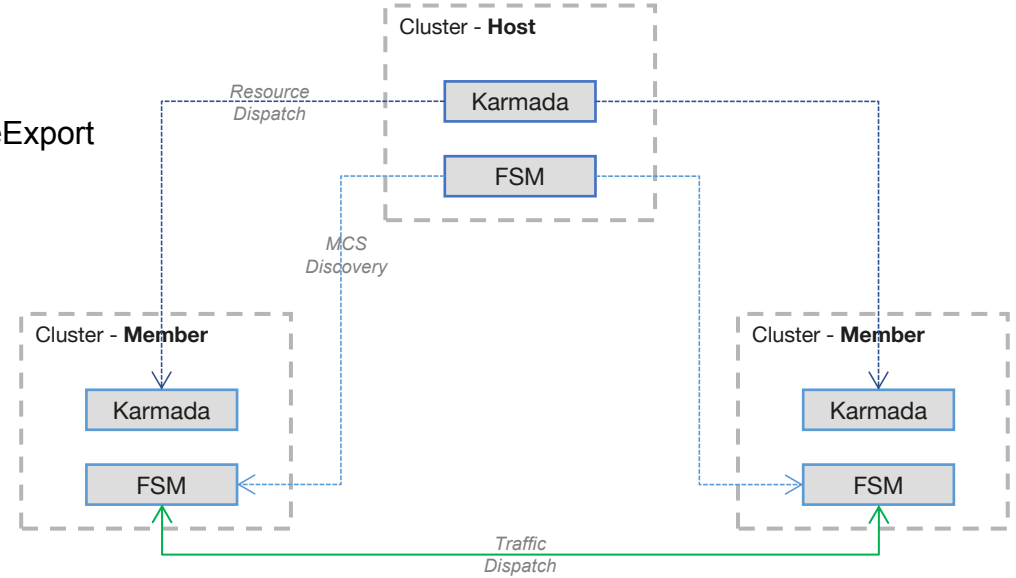
使用 Service Mesh 进行流量调度



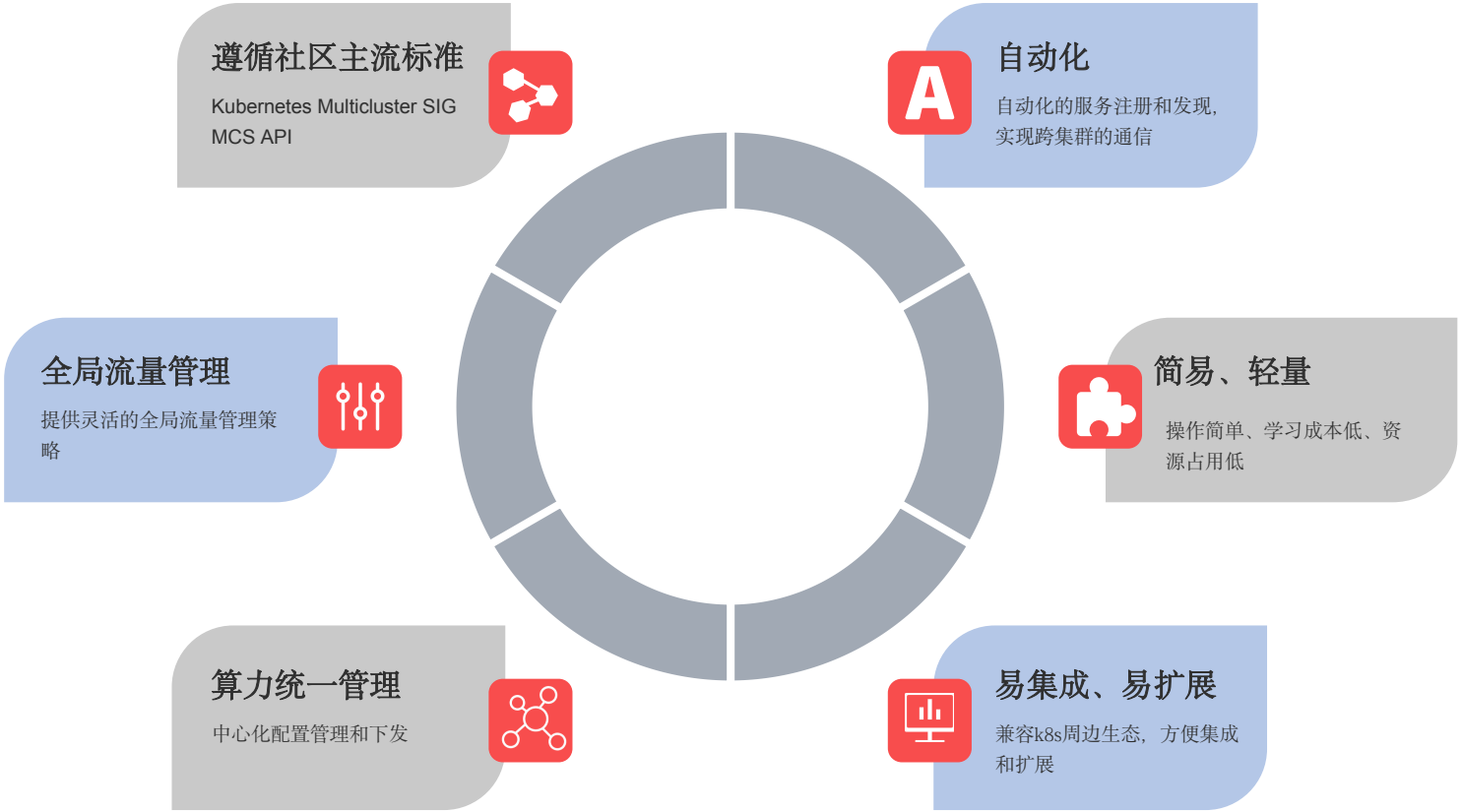
资源和流量的调度

资源调度： Deployment、Service、HPA、ServiceExport

多集群服务发现： ServiceExport、ServiceImport



方案亮点


遵循社区主流标准
Kubernetes Multicluster SIG MCS API
自动化
自动化的服务注册和发现，实现跨集群的通信
全局流量管理
提供灵活的全局流量管理策略
简易、轻量
操作简单、学习成本低、资源占用低
算力统一管理
中心化配置管理和下发
易集成、易扩展
兼容k8s周边生态，方便集成和扩展

加入 Karmada 社区

关注我们

https://karmada.io

https://github.com/karmada-io/karmada

https://slack.cncf.io (#karmada)

加入 Flomesh 社区

KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2023

关注我们

Flomesh flomesh.io

github.com/flomesh-io

flomesh-io.slack.com