
KubeCon
CloudNativeCon
OPEN SOURCE SUMMIT
China 2023

Cross-Cluster Traffic Orchestration with eBPF

Xiaohui Zhang Senior Architect, Evangelist, Flomesh
Zhen Chang Software Engineer, Huawei Cloud

Speaker Introduction

Xiaohui Zhang

Senior Programmer, LFAPAC Open Source Evangelist,

CNCF Ambassador, Microsoft MVP

Senior Architect/Evangelist at Flomesh.

Zhen Chang

Approver of the Karmada Community

Software Engineer at Huawei Cloud.

Introduction

Status and Challenges of Cloud-Native Application

Single Cluster Limit

- The number of nodes does not exceed 5000
- Pod does not exceed 150,000
- No more than 300,000 containers
- No more than 110 Pods per node

HA Requirement

- Avoid single points of failure
- Requirements for three centers in two places
- Service elastic traffic

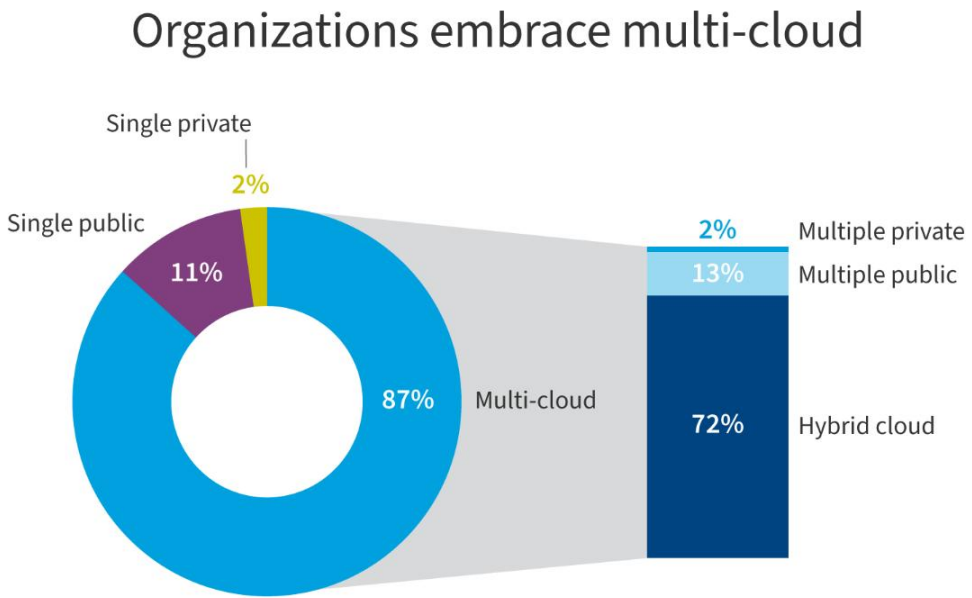
Multi-cloud Architecture

- Localized deployment
- IDC+Public Cloud Elasticity
- Avoid vendor binding
- Reduce costs and increase efficiency

Business Isolation

- Business isolation
- Team quarantine
- Development process isolation

Rise of Multi-Cloud and Multi-Cluster



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

More than 87% of enterprise respondents use the services of multiple cloud service providers at the same time.

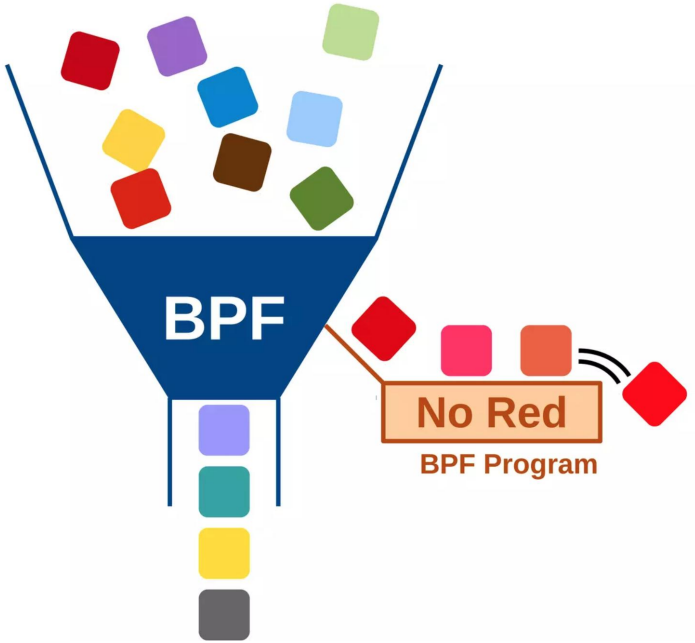
Cloud native technology and the cloud market continue to mature, and the future will be an era of programmatic multi-cloud management services.

eBPF Introduction

Berkeley Packet Filter

Originating from the 1992 paper titled "The BSD Packet Filter: A Novel Architecture for User-level Packet Capture"

Initially conceived as a network packet filter utilized in applications such as tcpdump.



eBPF = extended Berkeley Packet Filter

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

- Stability (DAG, reachability)
- Efficient (JIT native machine code)
- Security (verifier, limited helper function)
- Hot loading/unloading (no reboot required)

Programmable Kernel

Agenda

Trends and Requirements for Cloud-Native Multi-Cluster

eBPF in Cloud Native Traffic Management

Practice of Application and Traffic Scheduling

in Multi-Cluster Conclusion and Q&A

Trends and requirements for cloud-native multi-cluster environments

Trends and Requirements for Cloud-Native Multi-Cluster

Group of isolated islands

- Consistent cluster operations
- Consistent application delivery
- Businesses are separated and do not know each other.
- Data island, resource island, traffic island

Venice City

- Unified application delivery (deployment and operation)
- Unified application access (traffic distribution)
- Unified resource allocation (orchestration and scheduling)
- A small amount of cross-cluster business access with low
  sure

We are here

Age of Discovery

Instances, data, traffic:

- Automatic scheduling
- Free expansion and contraction
- free movement

Cloud-native Multi-Cloud is Challenging

Challenges of managing multi-cloud container clusters

Too Many clusters

Cumbersome and repetitive setup

Incompatible cluster lifecycle APIs

Fragmented API endpoints

Business fragmentation

Differentiated cluster configurations

Multi-cluster service discovery required

Sync apps between clusters

Cluster boundary restrictions

Resource scheduling

Application availability

Horizontal auto-scaling

Vendor locking

Deployment gravity

Lack of migration automation

Lack of independent, neutral, open source multi-cluster management projects

Karmada: OpenSource Multi-Cloud Container Orchestration

Build infinitely scalable container resource pools with Karmada

Let developers use multi-cloud like a K8s cluster

K8s native API compatible

Upgrade from single cluster to multiple clusters with zero modification
Seamlessly integrate K8s single cluster tool chain ecology

Open and neutral

From the Internet, finance, manufacturing,
Jointly initiated by operators, cloud vendors, etc.

No Locking

Multi-cloud platform support, automatic allocation, free migration
Not tied to commercial products from manufacturers

Ready out of the box

Built-in policy set for multiple scenarios: three centers in two places, active-active in the same city, remote disaster recovery

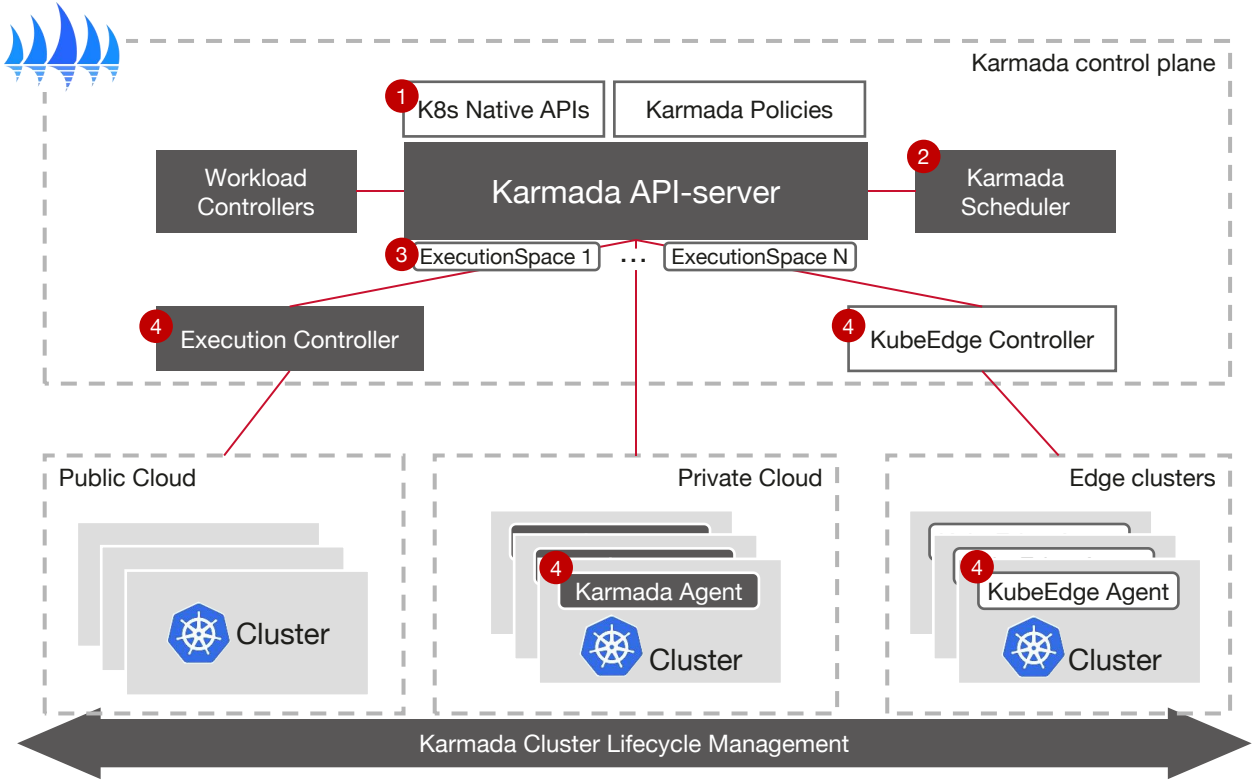
Rich multi-cluster scheduling

Cluster affinity scheduling, multi-granule multi-cluster high-availability deployment: multi-Region, multi-AZ, multi-cluster, multi-vendor

Centralized management

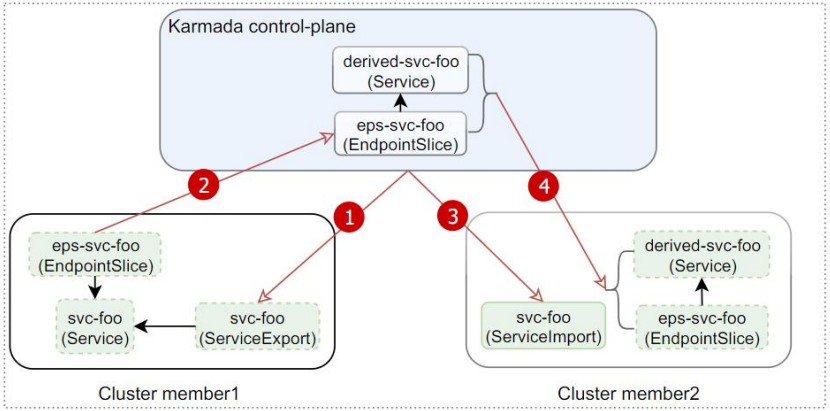
No need to worry about cluster location
Supports public cloud, private cloud, and edge clusters

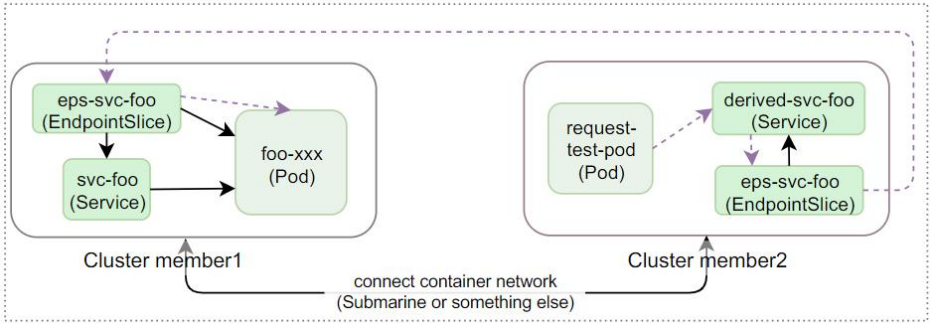
Karmada Architecture



Karmada Adopters

Multi-Cluster Service Discovery

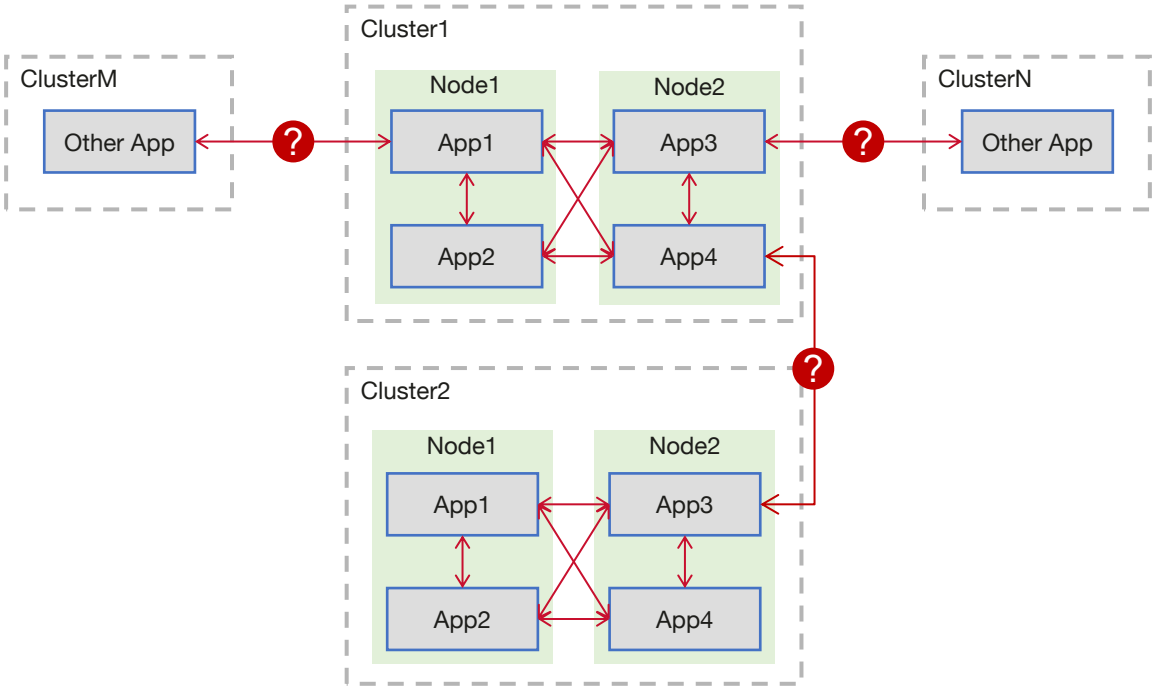




Cross-Cluster Communication

1. What if the cross-cluster container network is not connected?
2. In addition to mcs-api, is there any other way to achieve cross-cluster service discovery?
3. …

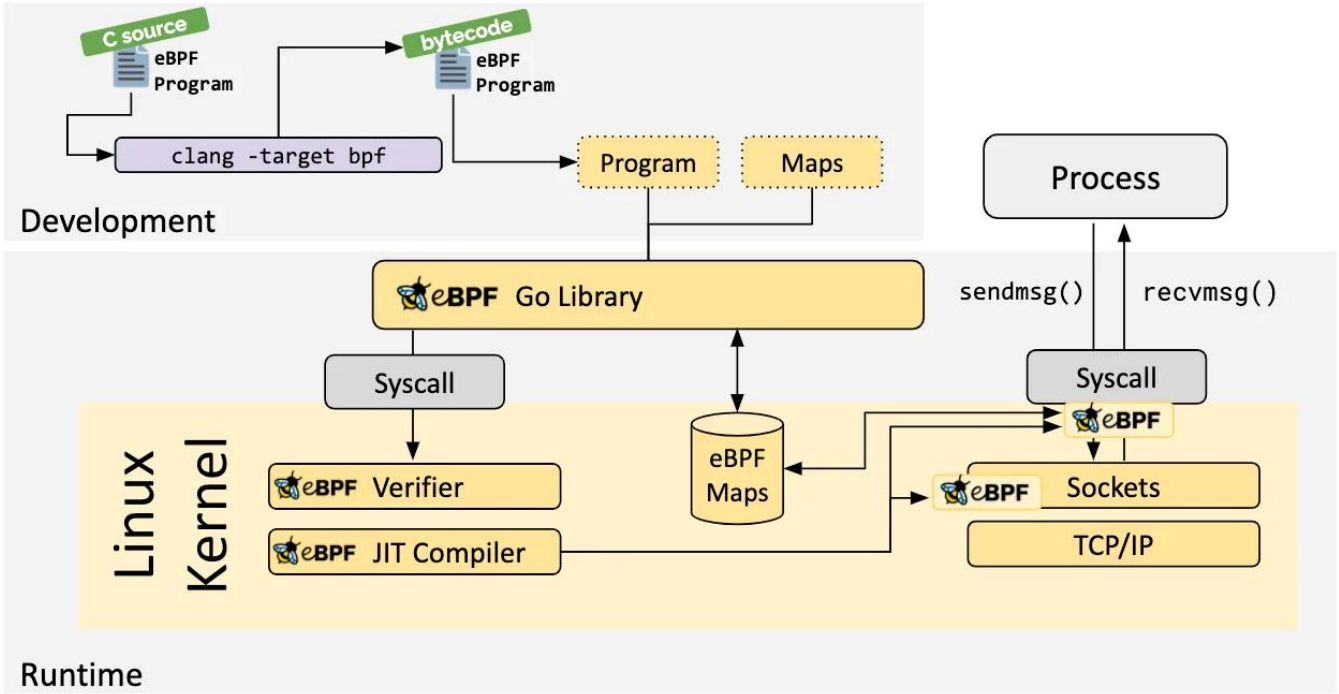
Island Network Model



Island Network Model vs Flat Network Model

eBPF in Cloud Native Traffic Management

eBPF Loader and Verifier

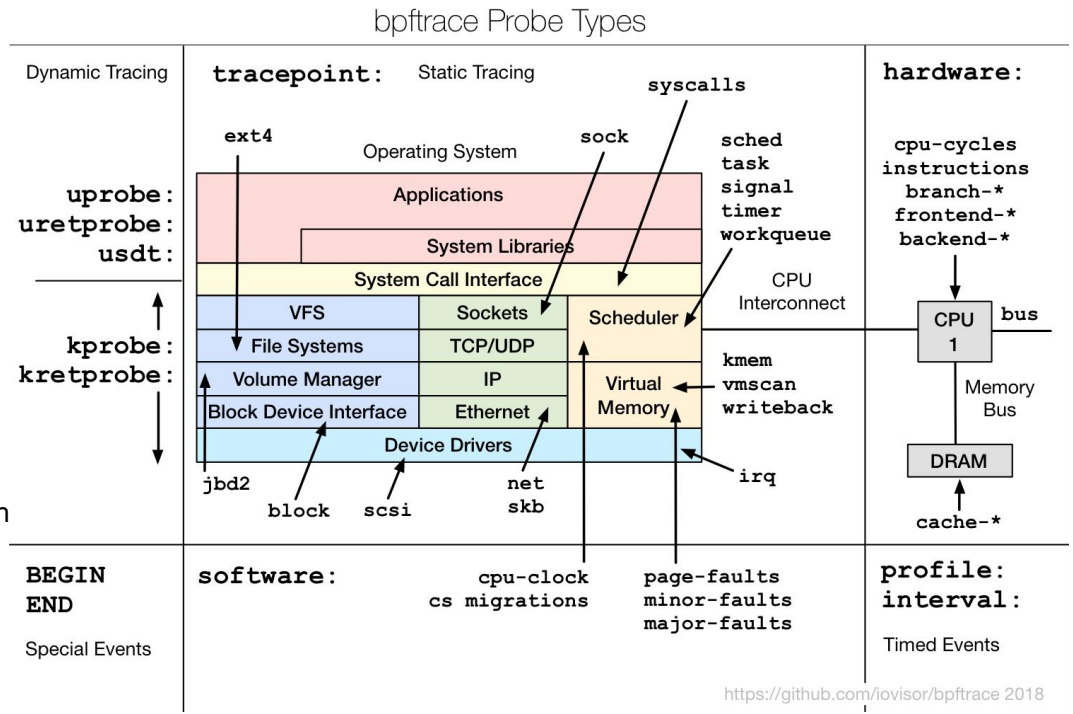


eBPF Event-Drivent

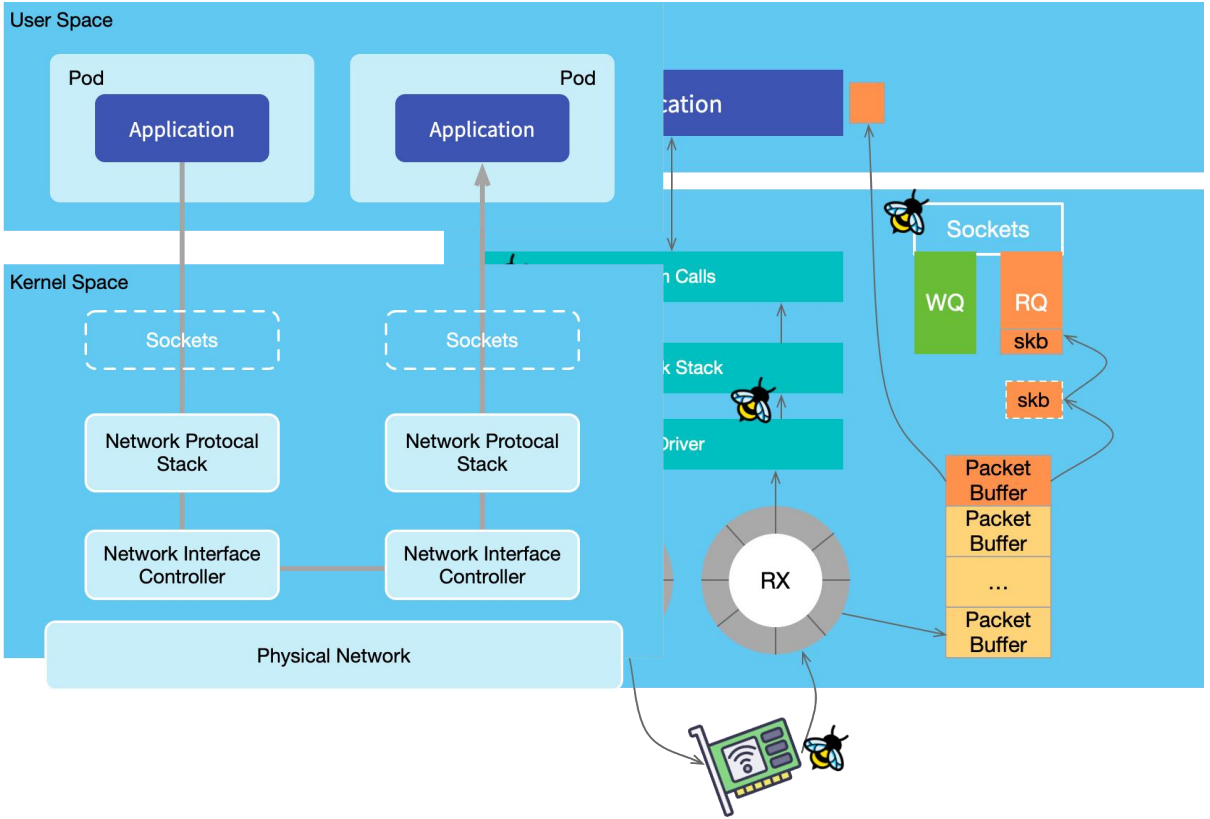
Event -> Action

Event:

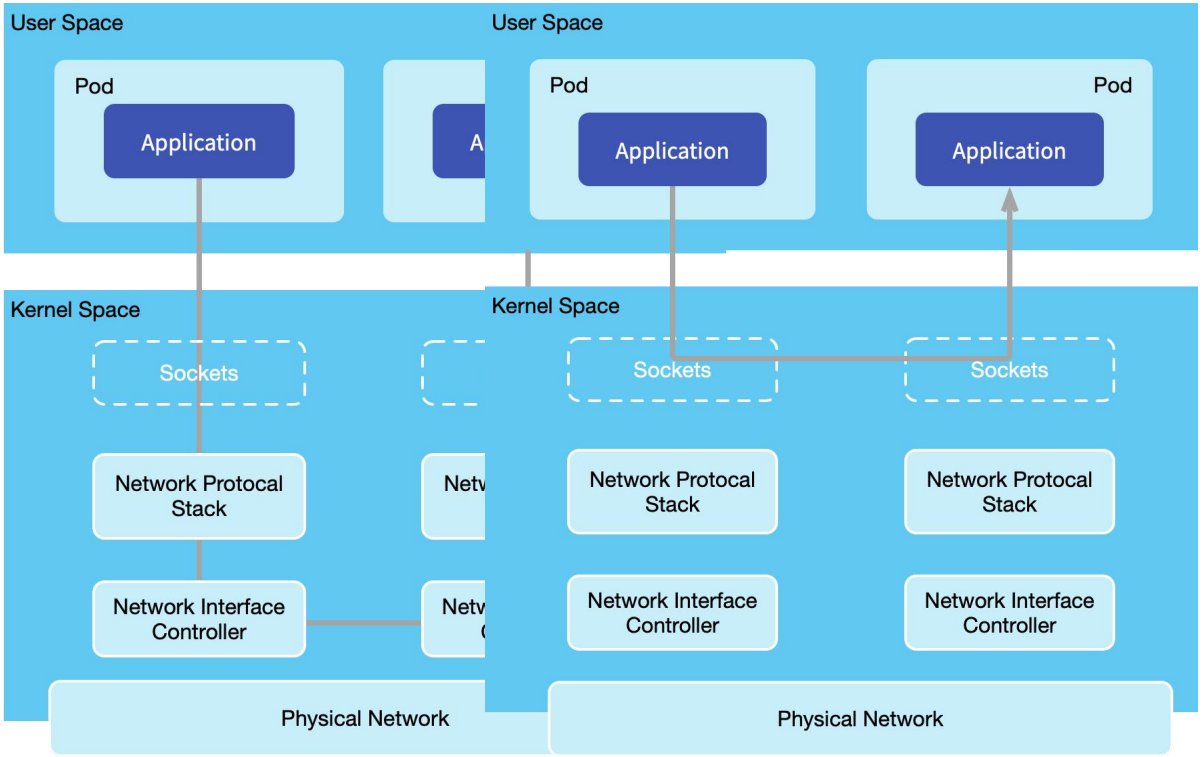
- Kprobe/Kretprobe (Kernel function entry and exit)
- Uprobe/Uretprobe (User function entry and exit)
- XDP (eXpress Data Path)
- Tracepoint (triggered on specific events)
- Perf (performance events such as CPU cycle coun



Network communication between Pods

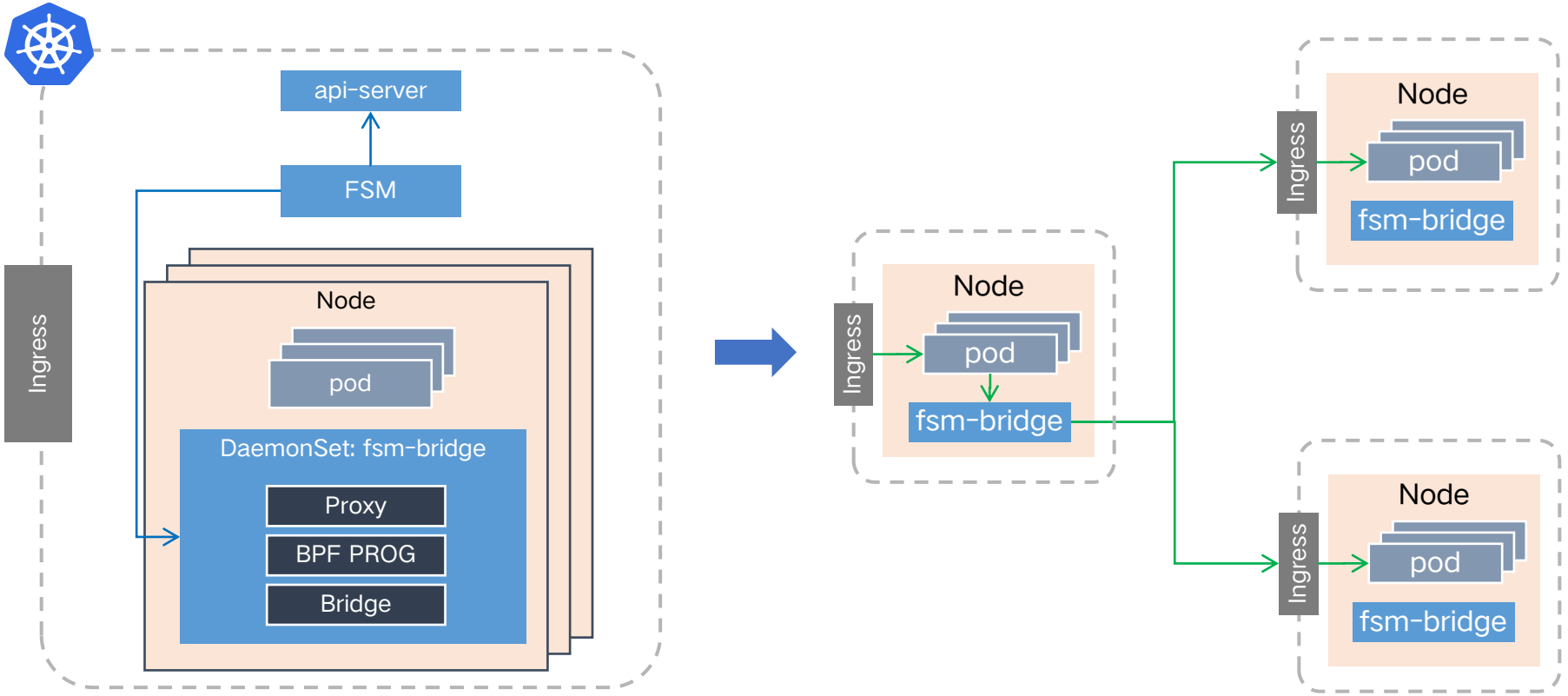


NAT

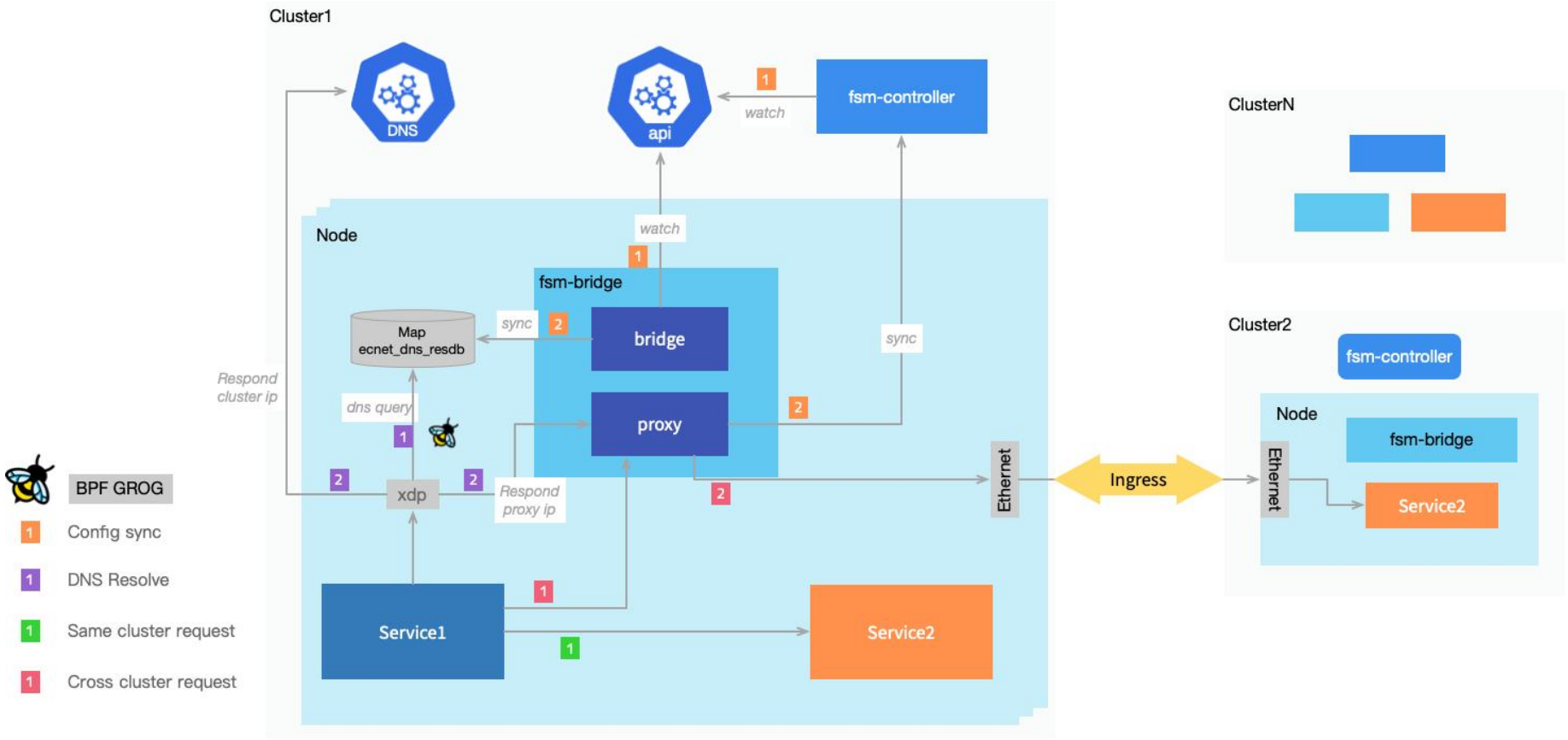

User Space
User Space
Pod
Application
Pod
Application
Pod
Application
Kernel Space
Kernel Space
Sockets
Sockets
Sockets
Network Protocal Stack
Network Protocal Stack
Network Protocal Stack
Network Interface Controller
Network Interface Controller
Network Interface Controller
Physical Network
Physical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

Practice of Application and Traffic Scheduling in Multi-Cluster

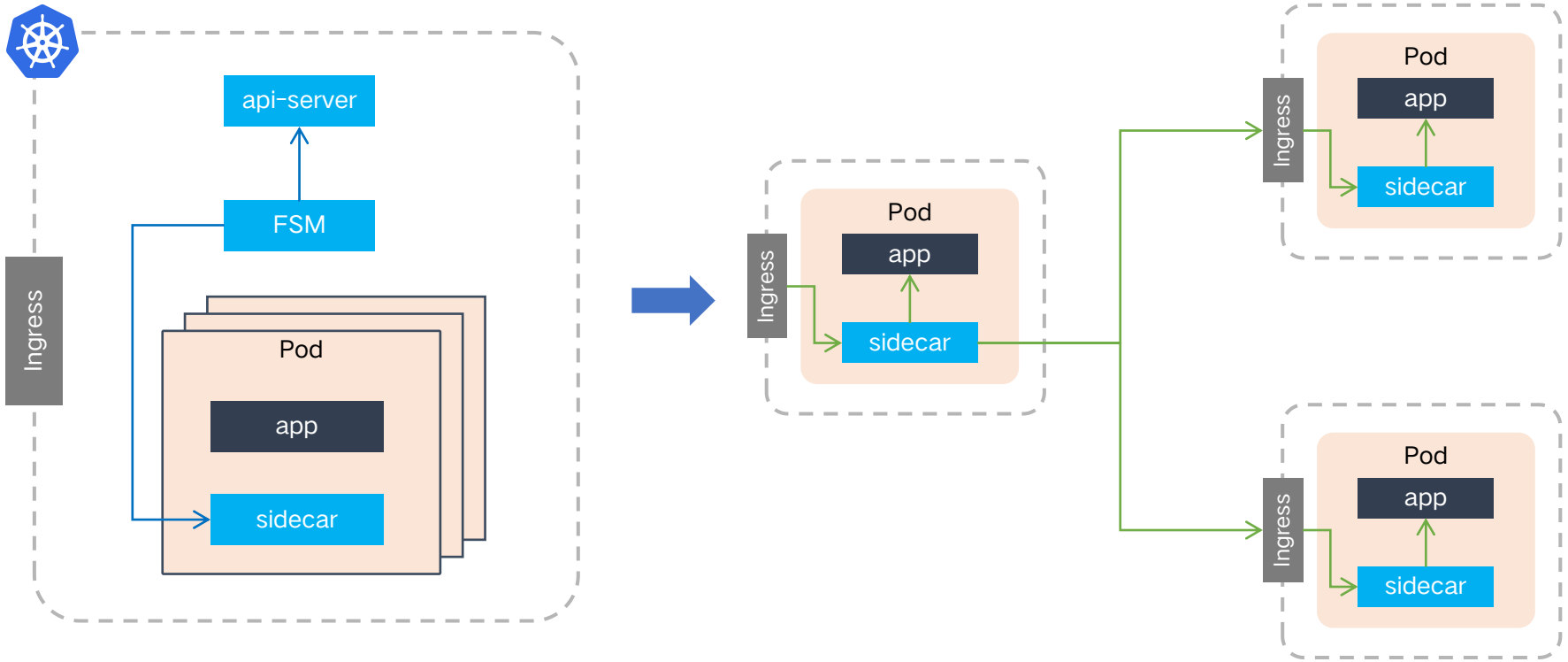
Traffic Dispatching with eBPF



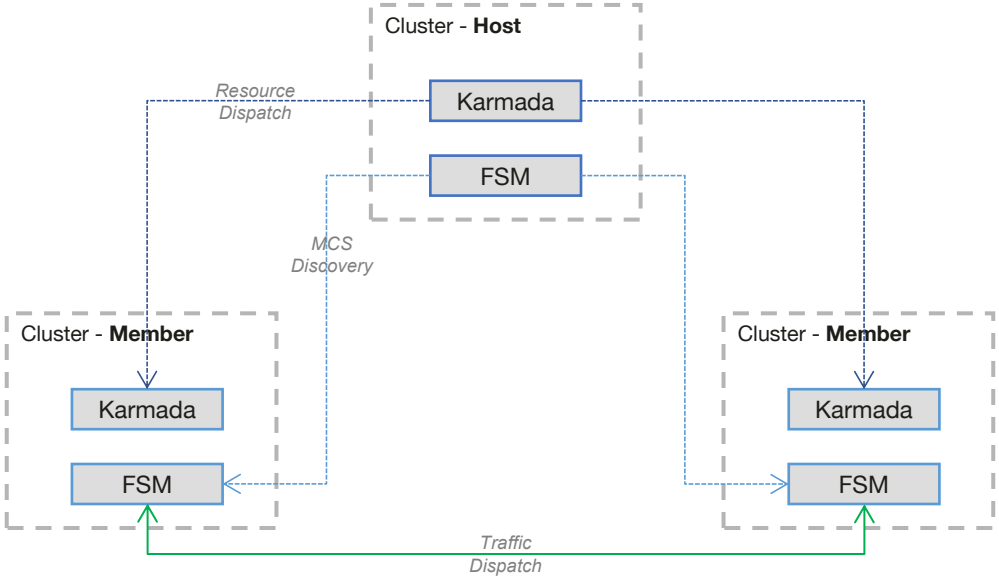
Traffic Scheduling with eBPF



Traffic Dispatching with Service Mesh



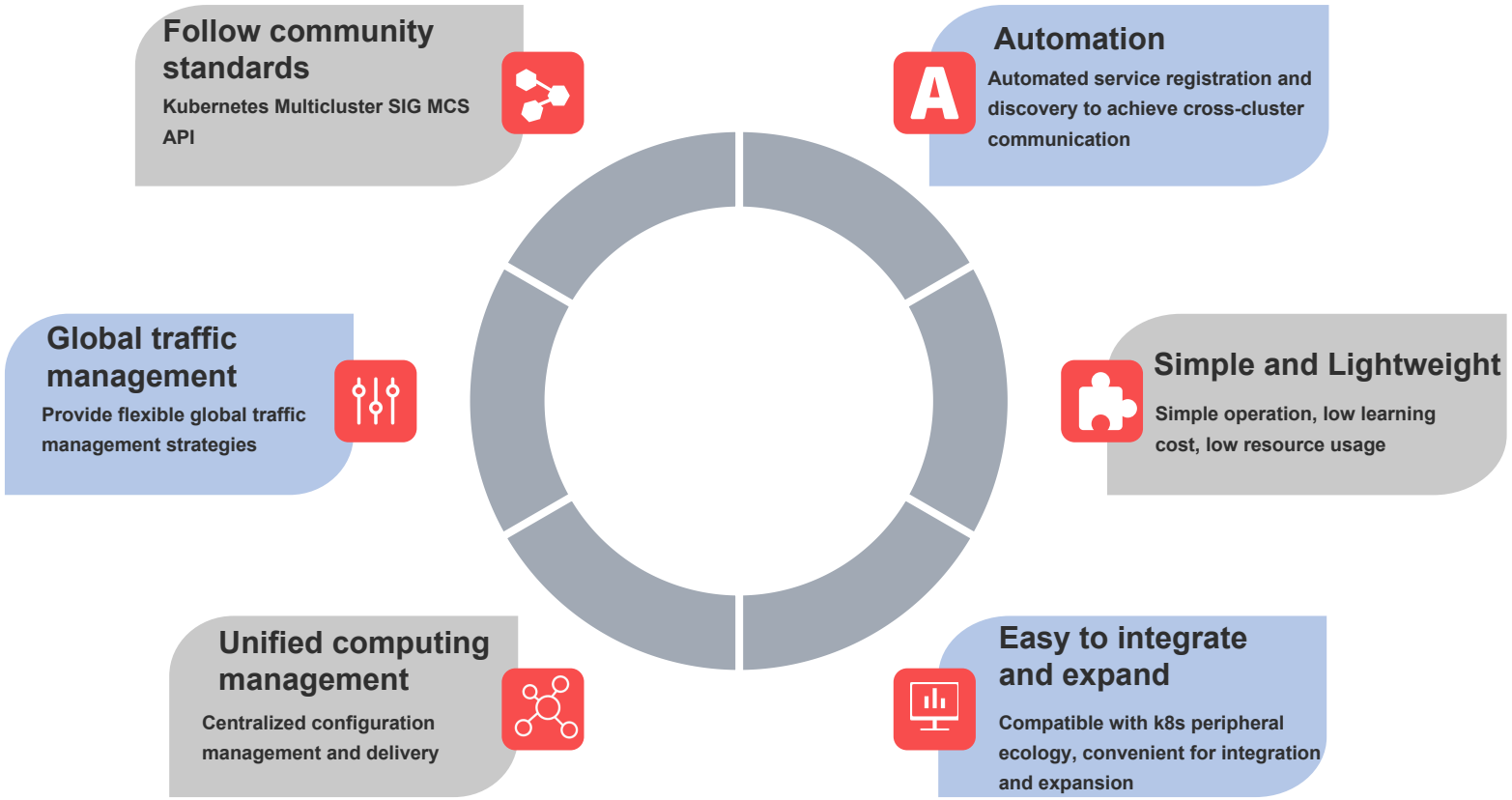
Resources and Traffic Scheduling



Resource Scheduling：Deployment、Service、HPA、ServiceExport

Service Discovery in MC：ServiceExport、ServiceImport

Solution Highlights



Join Karmada Community

KubeCon CloudNativeCon OPEN SOURCE SUMMIT China 2023

Follow us

https://karmada.io

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

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

Join Flomesh community

Follow us

flomesh.io

github.com/flomesh-io

flomesh-io.slack.com