Service mesh для построения мультикластерной системы

Лукьянченко Александр
Lead engineer

Санкт-Петербург, 2019

DevOops

Avito

План

План

- Проблемы внедрения нескольких кластеров в одном окружении

План

- Проблемы внедрения нескольких кластеров в одном окружении
- Готовые решения, сравнение

План

- Проблемы внедрения нескольких кластеров в одном окружении
- Готовые решения, сравнение
- Service mesh подход

План

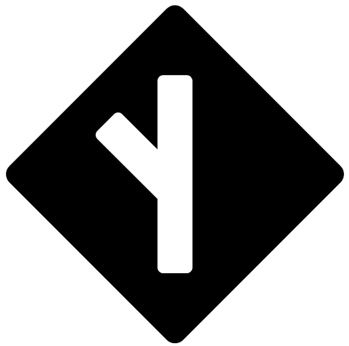
- Проблемы внедрения нескольких кластеров в одном окружении
- Готовые решения, сравнение
- Service mesh подход
- Наше решение

План

- Проблемы внедрения нескольких кластеров в одном окружении
- Готовые решения, сравнение
- Service mesh подход
- Наше решение
- Возможности

План

- Проблемы внедрения нескольких кластеров в одном окружении
- Готовые решения, сравнение
- Service mesh подход
- Наше решение
- Возможности
- Выводы



traffic control



?

Что хотим?

Что хотим?

- Прозрачный discovery на несколько кластеров

Что хотим?

- Прозрачный discovery на несколько кластеров
- Все DNS имена должны остаться прежними

Что хотим?

- Прозрачный discovery на несколько кластеров
- Все DNS имена должны остаться прежними
- Балансировка по нескольким кластерам

Что хотим?

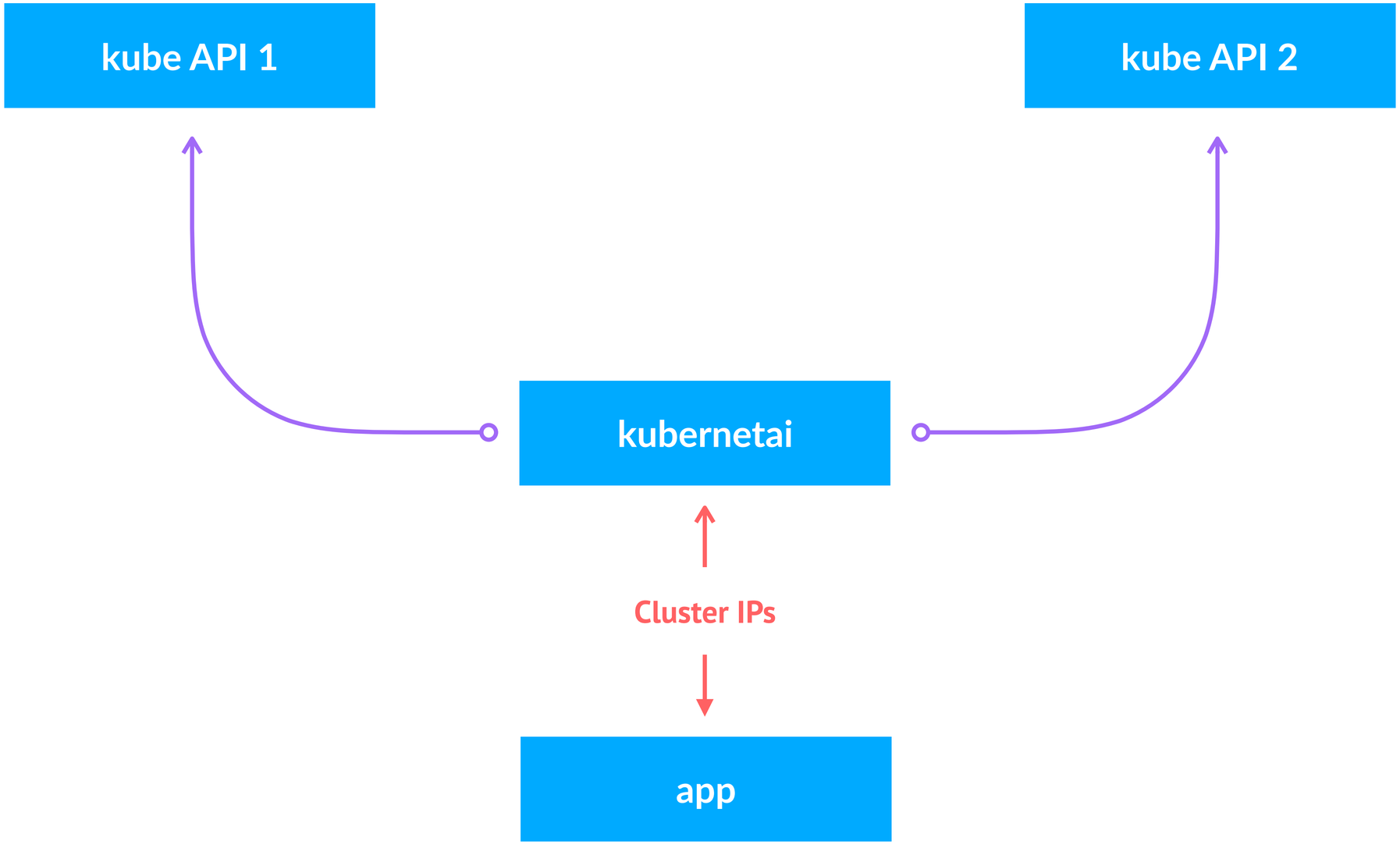
- Прозрачный discovery на несколько кластеров
- Все DNS имена должны остаться прежними
- Балансировка по нескольким кластерам
- Поддержка разных видов балансировки с сохранением локальности

Что хотим?

- Прозрачный discovery на несколько кластеров
- Все DNS имена должны остаться прежними
- Балансировка по нескольким кластерам
- Поддержка разных видов балансировки с сохранением локальности
- Canary релизы сквозь несколько кластеров

Что-нибудь готовое?

Kubernetai



Kubernetai

Kubernetai

- Использование native подхода для балансировки в kubernetes

Kubernetai

- Использование native подхода для балансировки в kubernetes
- Необходимо вручную делать схему с балансировкой на поды в другом кластере

- Использование native подхода для балансировки в kubernetes
- Необходимо вручную делать схему с балансировкой на поды в другом кластере
- На DNS лучше не завязываться для динамичной балансировки

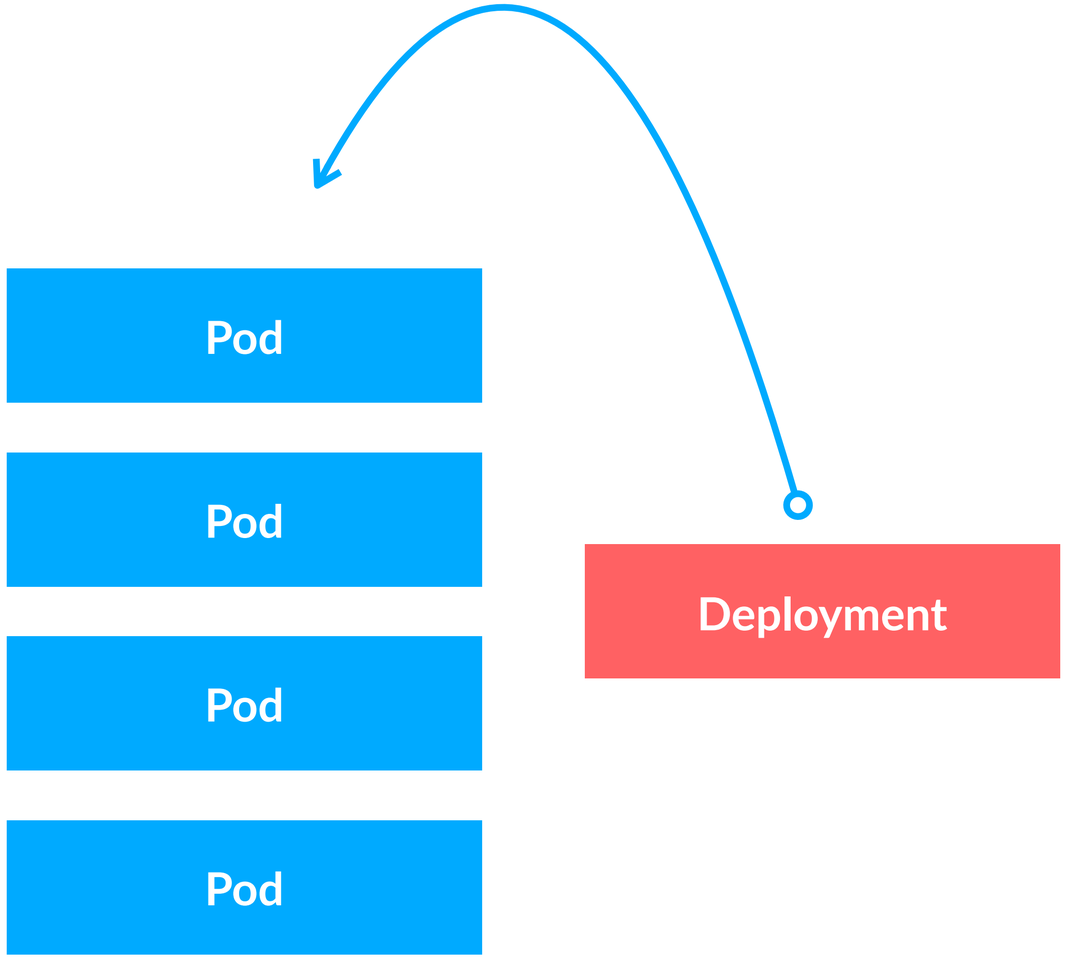
Kubernetai

- Использование native подхода для балансировки в kubernetes
- Необходимо вручную делать схему с балансировкой на поды в другом кластере
- На DNS лучше не завязываться для динамичной балансировки
- TTL, cache… :(

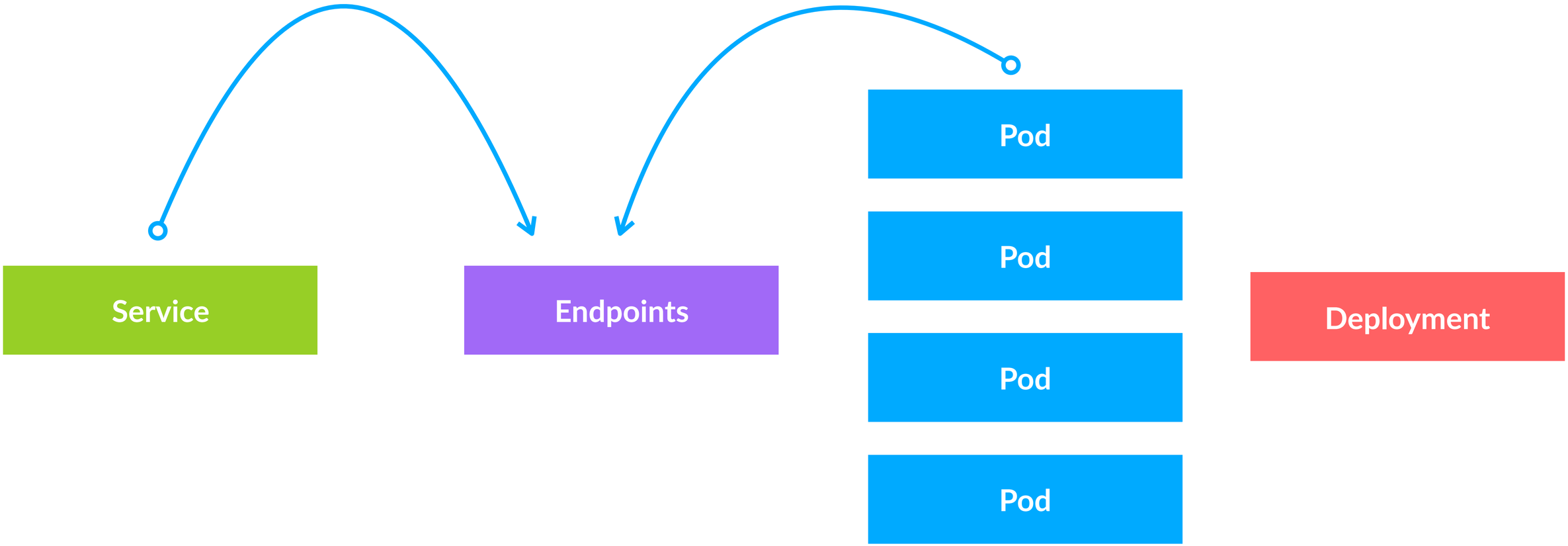
Controller manager

Deployment

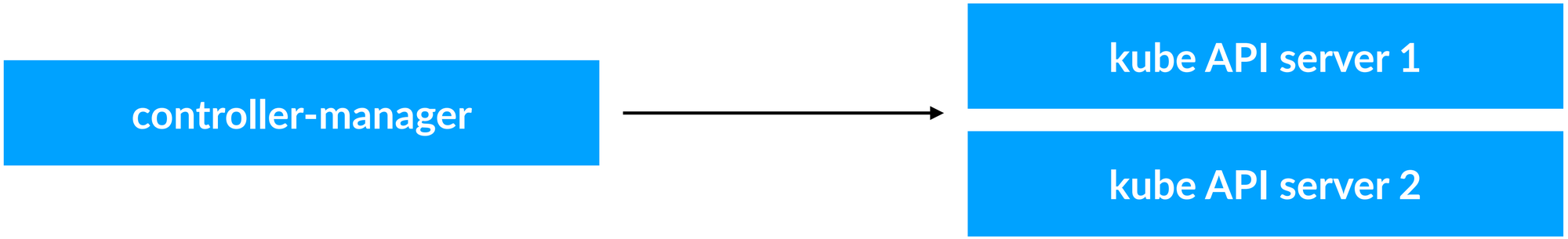
Controller manager



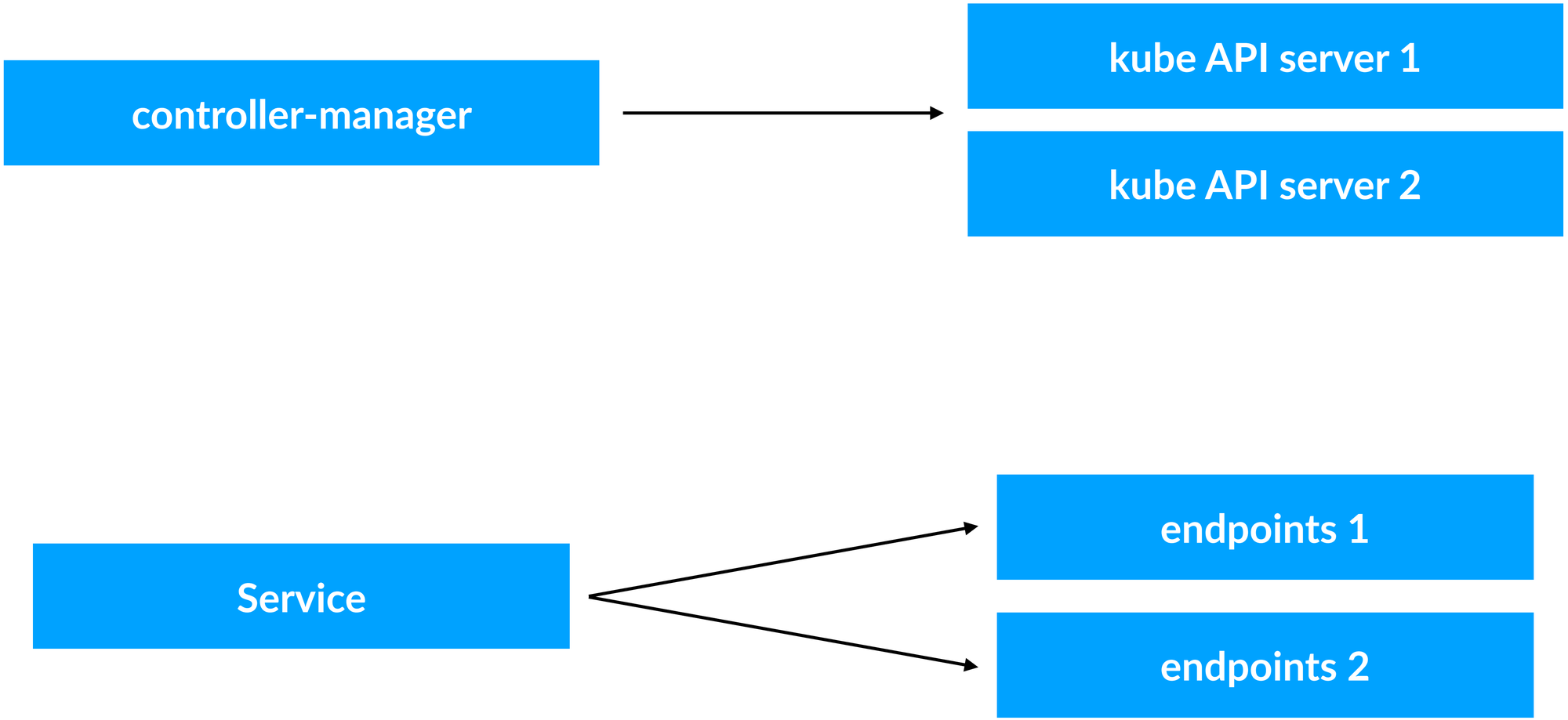
Controller manager



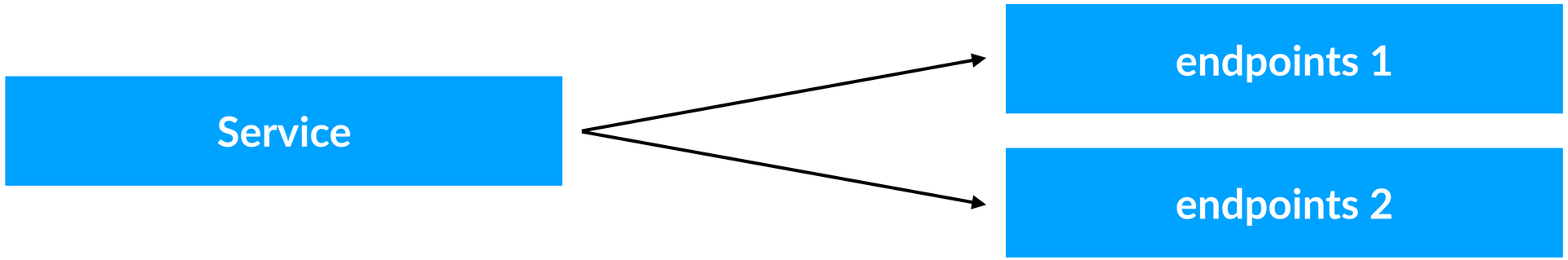
Прокачиваем controller-manager?



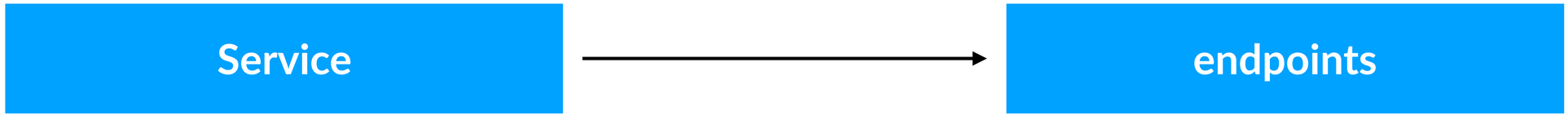
Прокачиваем controller-manager?



Прокачиваем controller-manager?



Прокачиваем controller-manager?



Прокачиваем controller-manager?

Прокачиваем controller-manager?

- Использование нативного подхода к балансировке в kubernetes

Прокачиваем controller-manager?

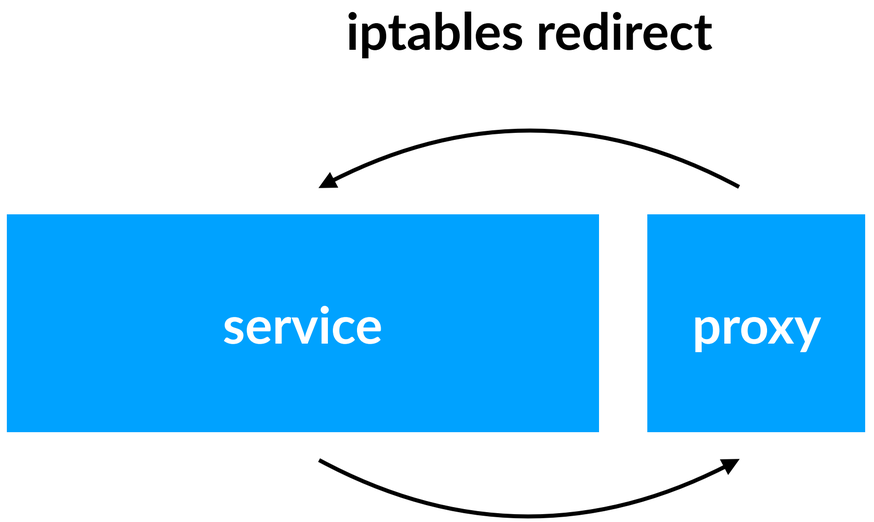
- Использование нативного подхода к балансировке в kubernetes
- Форк или своя имплементация

Прокачиваем controller-manager?

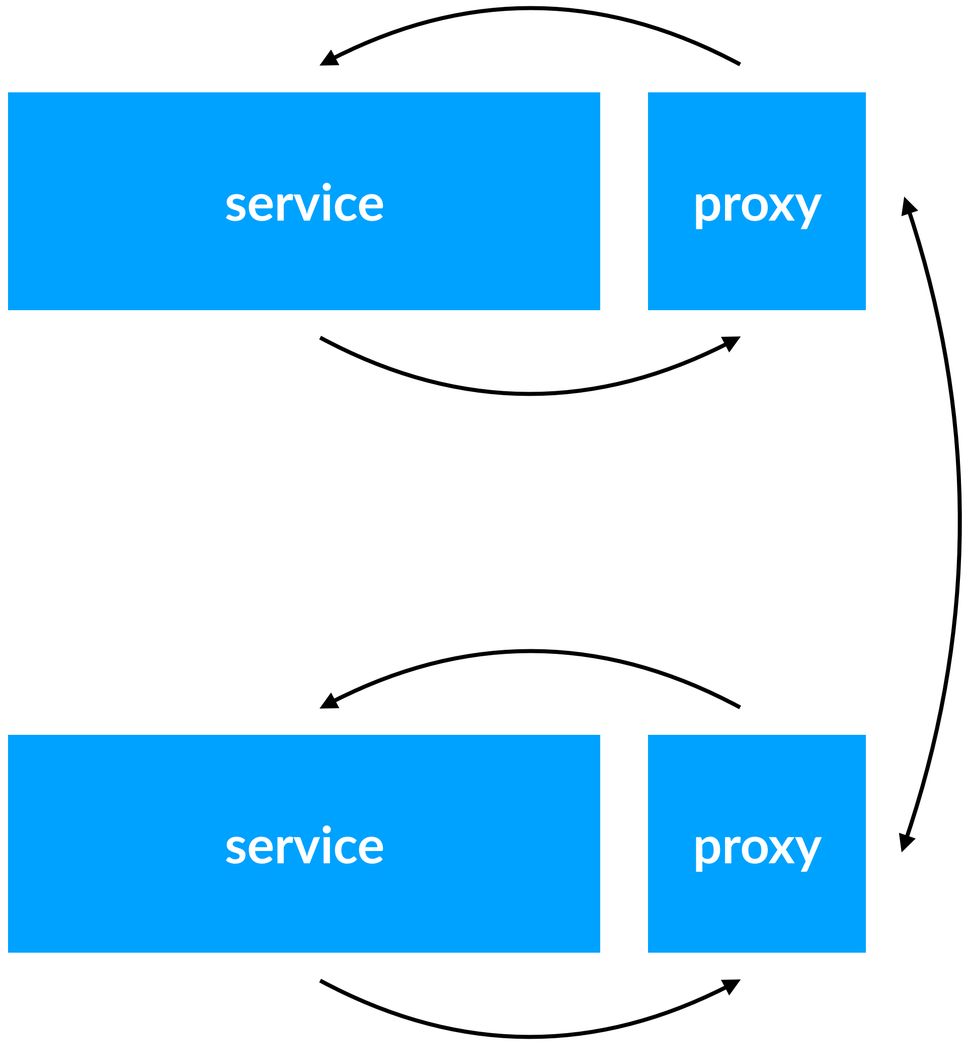
- Использование нативного подхода к балансировке в kubernetes
- Форк или своя имплементация
- Весовая балансировка затруднительна

Service mesh

Service mesh



Service mesh



Service mesh - зачем?

Service mesh - зачем?

- Добавление одинаковой логики в сетевые взаимодействия

Service mesh - зачем?

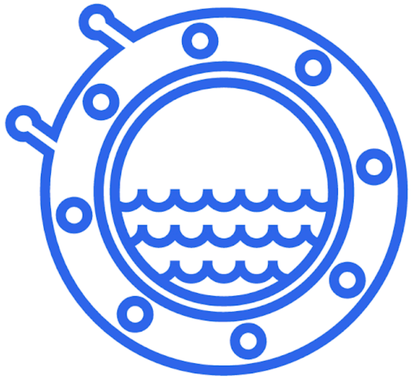
- Добавление одинаковой логики в сетевые взаимодействия
- Возможность внедрения в гетерогенных системах

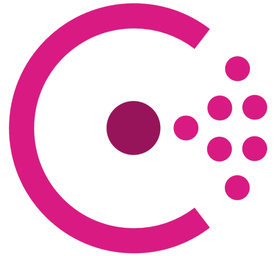
Service mesh - зачем?

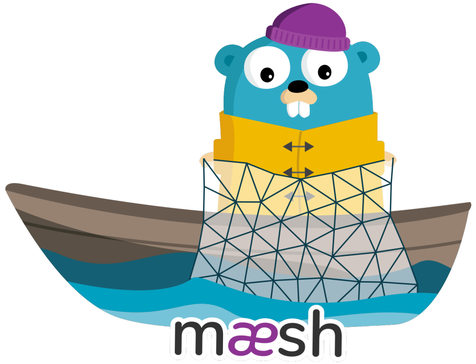
- Добавление одинаковой логики в сетевые взаимодействия
- Возможность внедрения в гетерогенных системах
- Гарантия выполнения сетевых политик

Service mesh - зачем?

- Добавление одинаковой логики в сетевые взаимодействия
- Возможность внедрения в гетерогенных системах
- Гарантия выполнения сетевых политик
- L7 логика

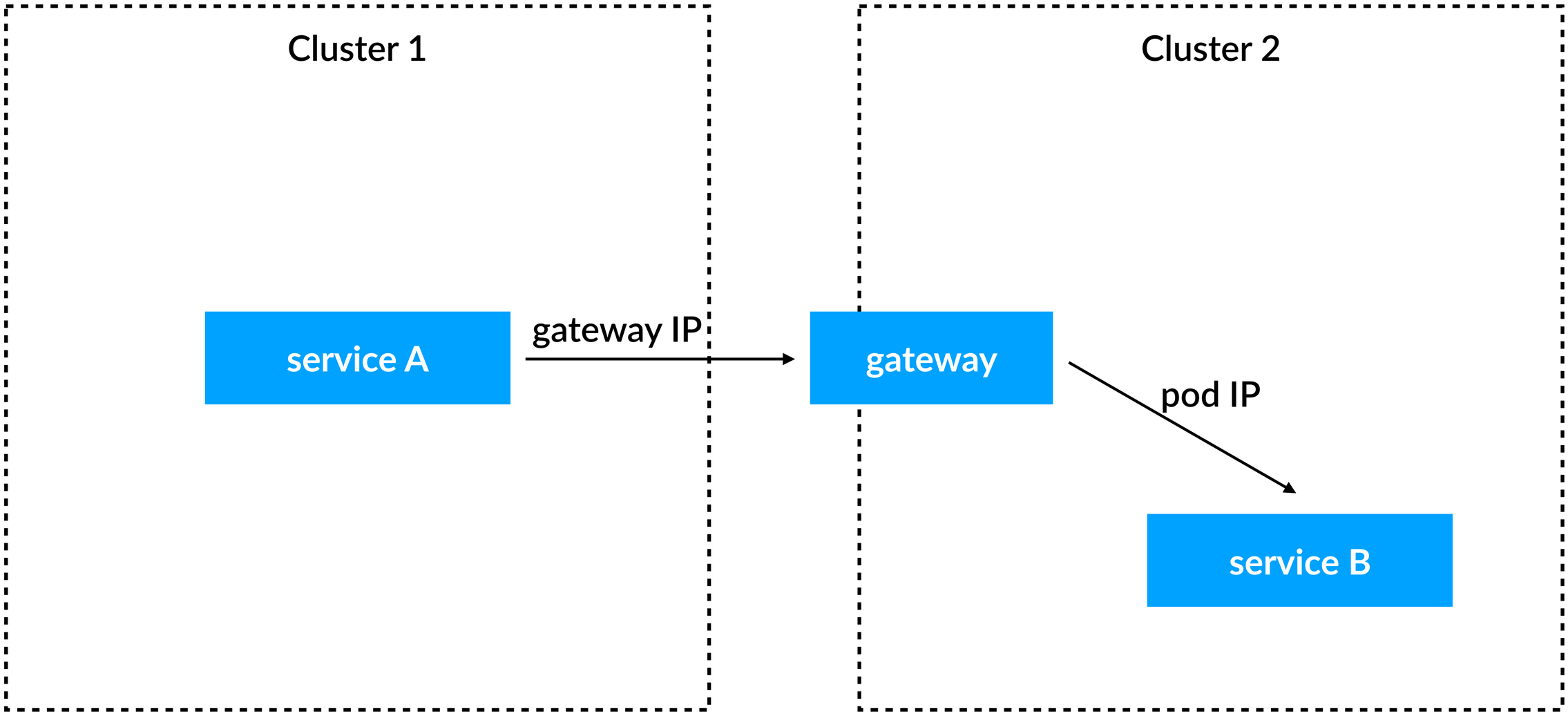




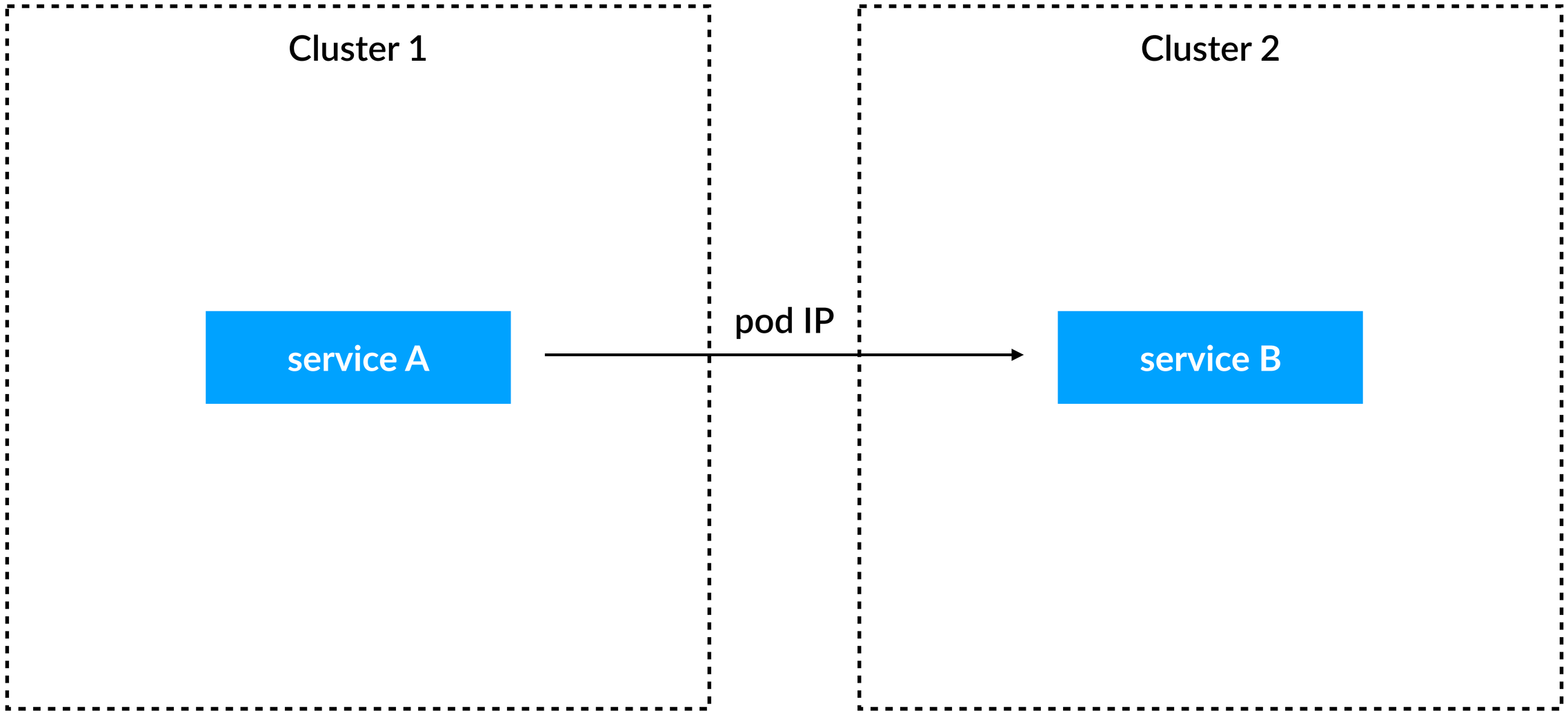


Kuma

Istio gateway



Istio VPN connectivity



Istio way

Istio way

- Поддержка нескольких кластеров из коробки

Istio way

- Поддержка нескольких кластеров из коробки
- Возможность соблюдения локальности

Istio way

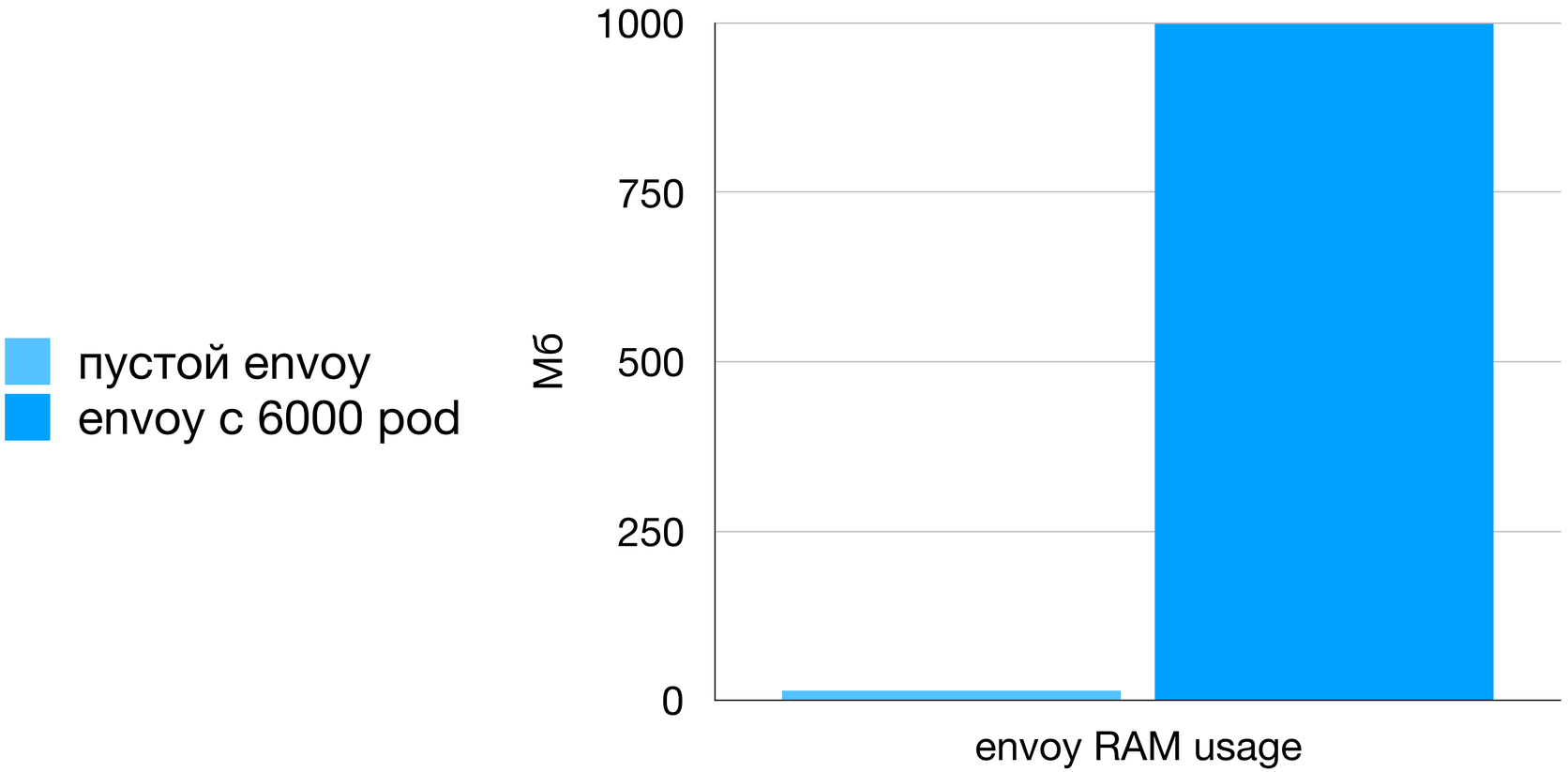
- Поддержка нескольких кластеров из коробки
- Возможность соблюдения локальности
- Гибкая настройка (Custom Resources)

Istio way

- Поддержка нескольких кластеров из коробки
- Возможность соблюдения локальности
- Гибкая настройка (Custom Resources)
- Не масштабируется :(

Сколько стоит ваш sidecar?

Потребление памяти



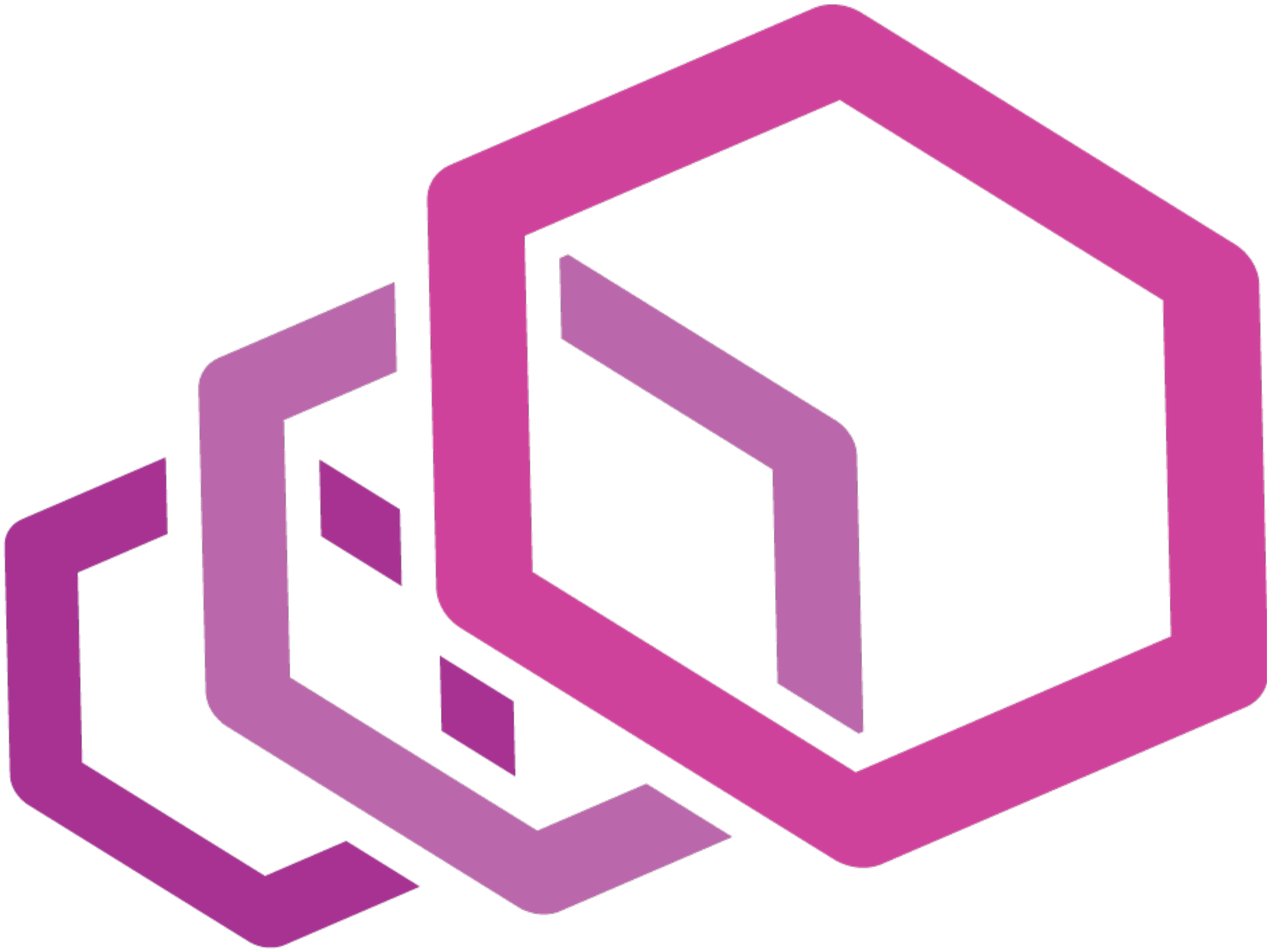
Общее потребление по системе

1 Gb * 6000 pod = 6 Tb

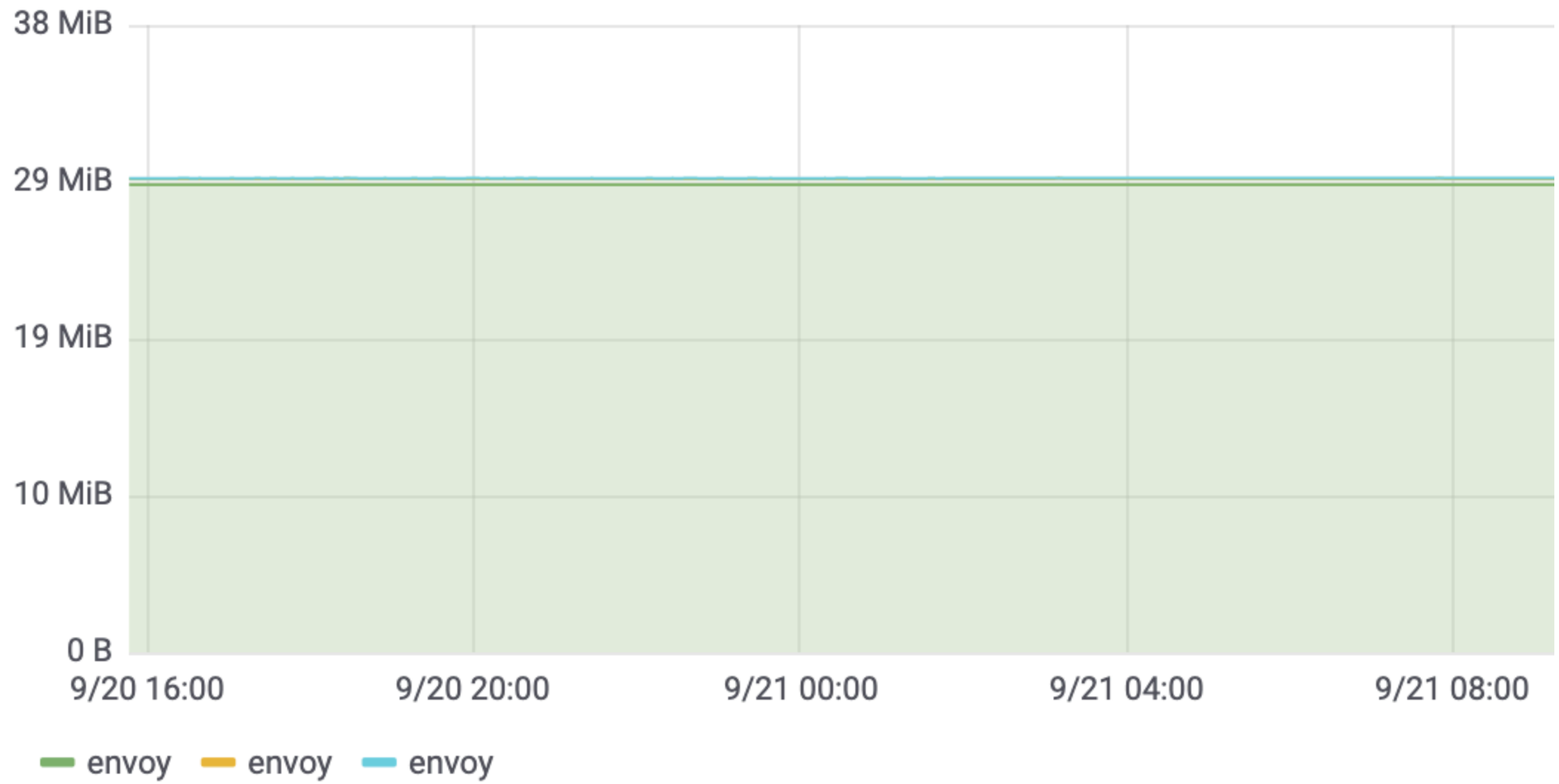
6Tb

после добавления sidecar к каждому сервису

Что делать?



Envoy RAM Usage

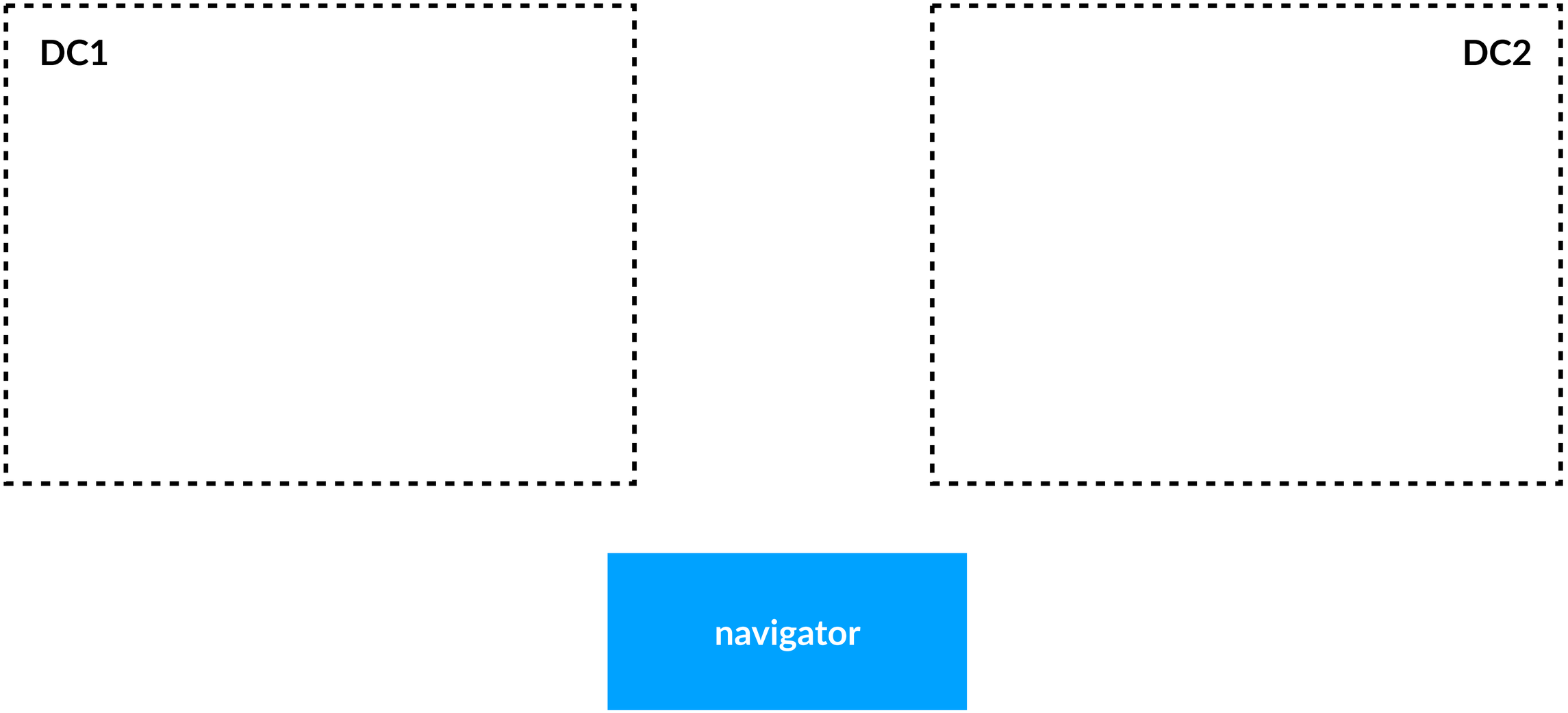


Общее потребление по системе

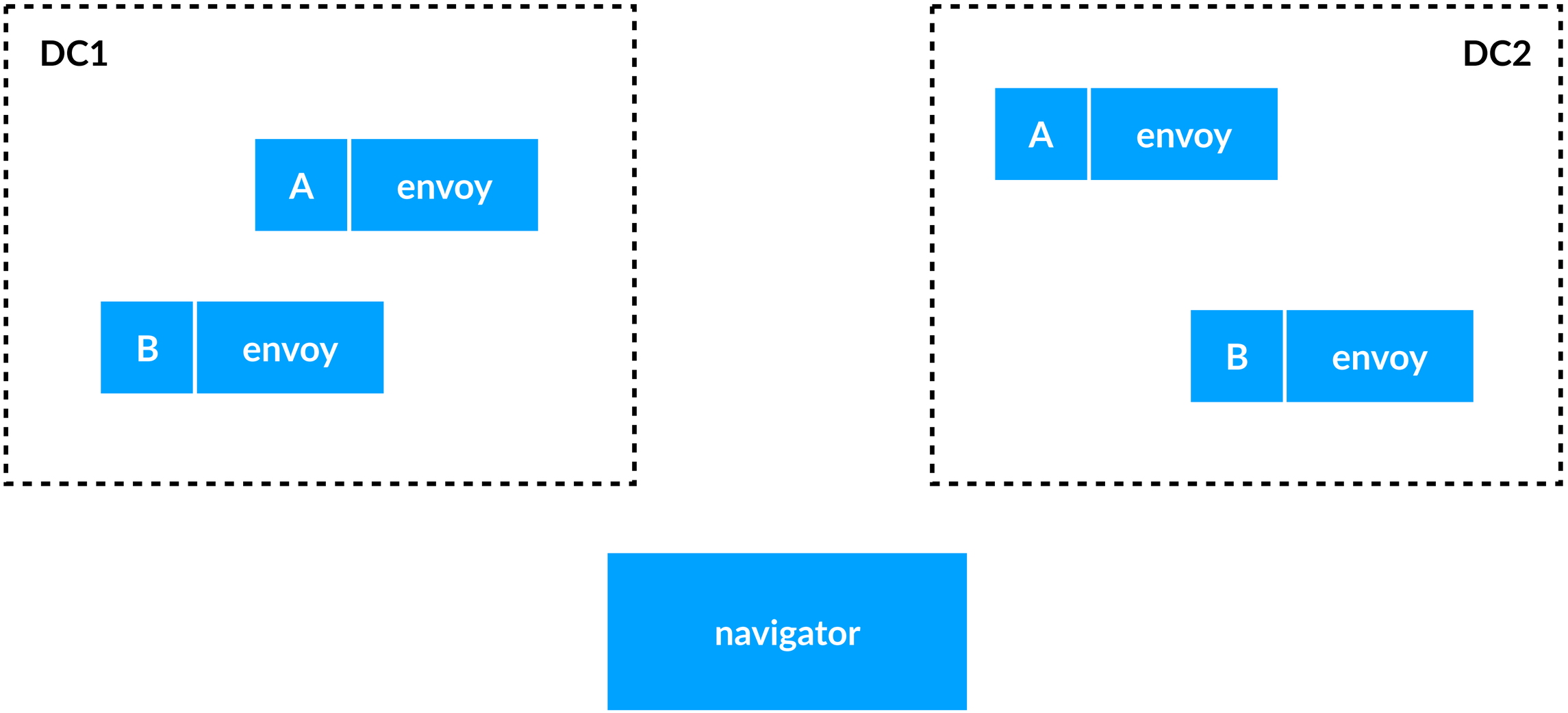
30 Mb * 6000 pod = 180 Gb

Как?

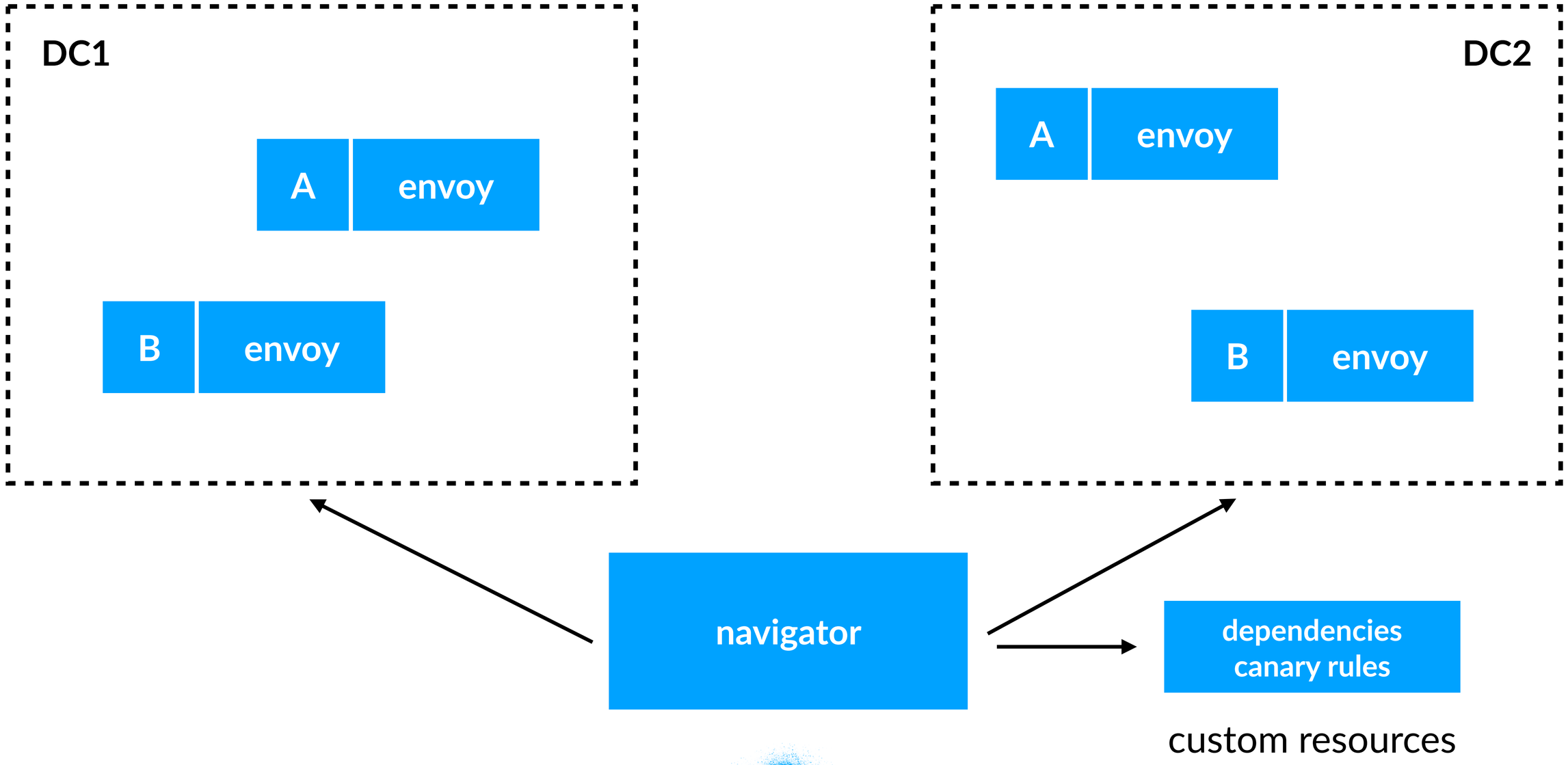
Navigator



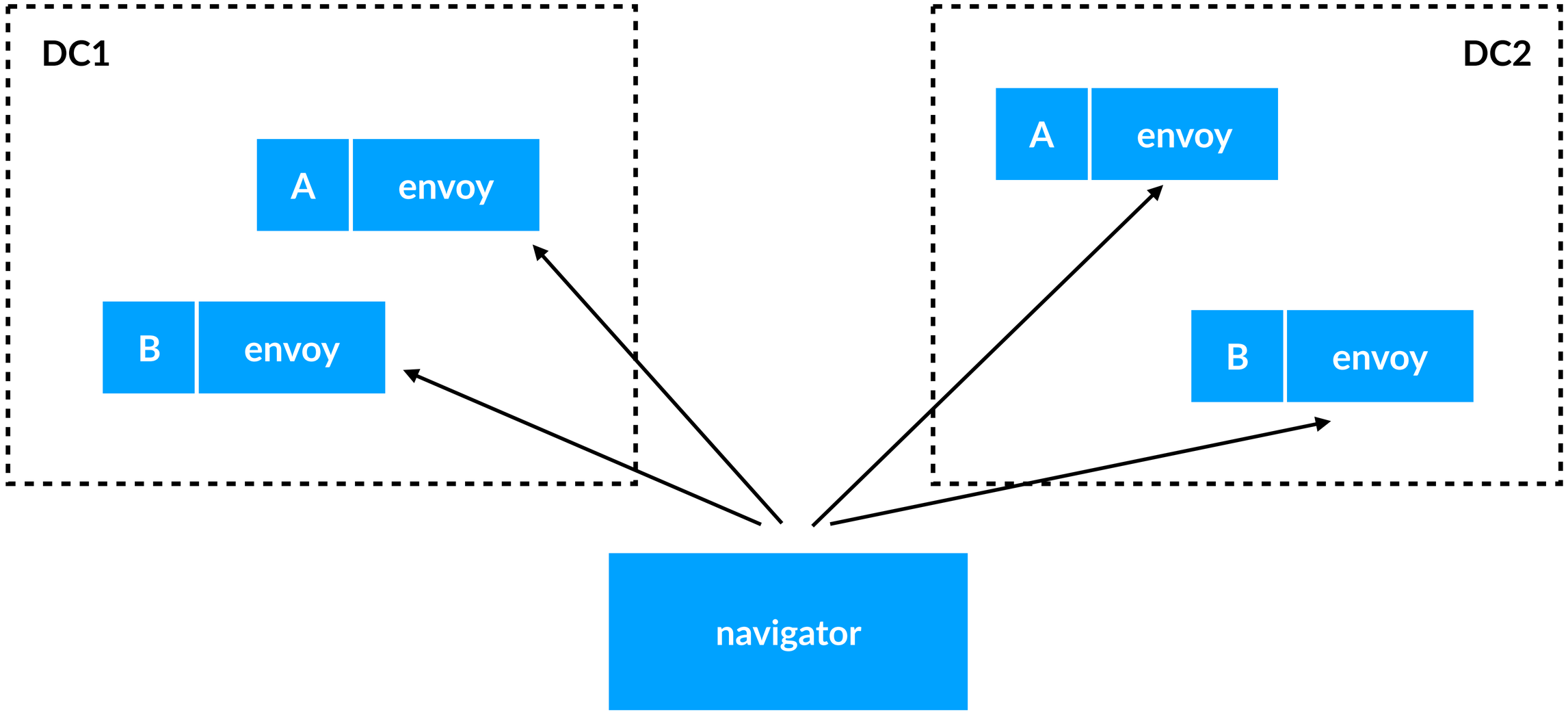
Navigator



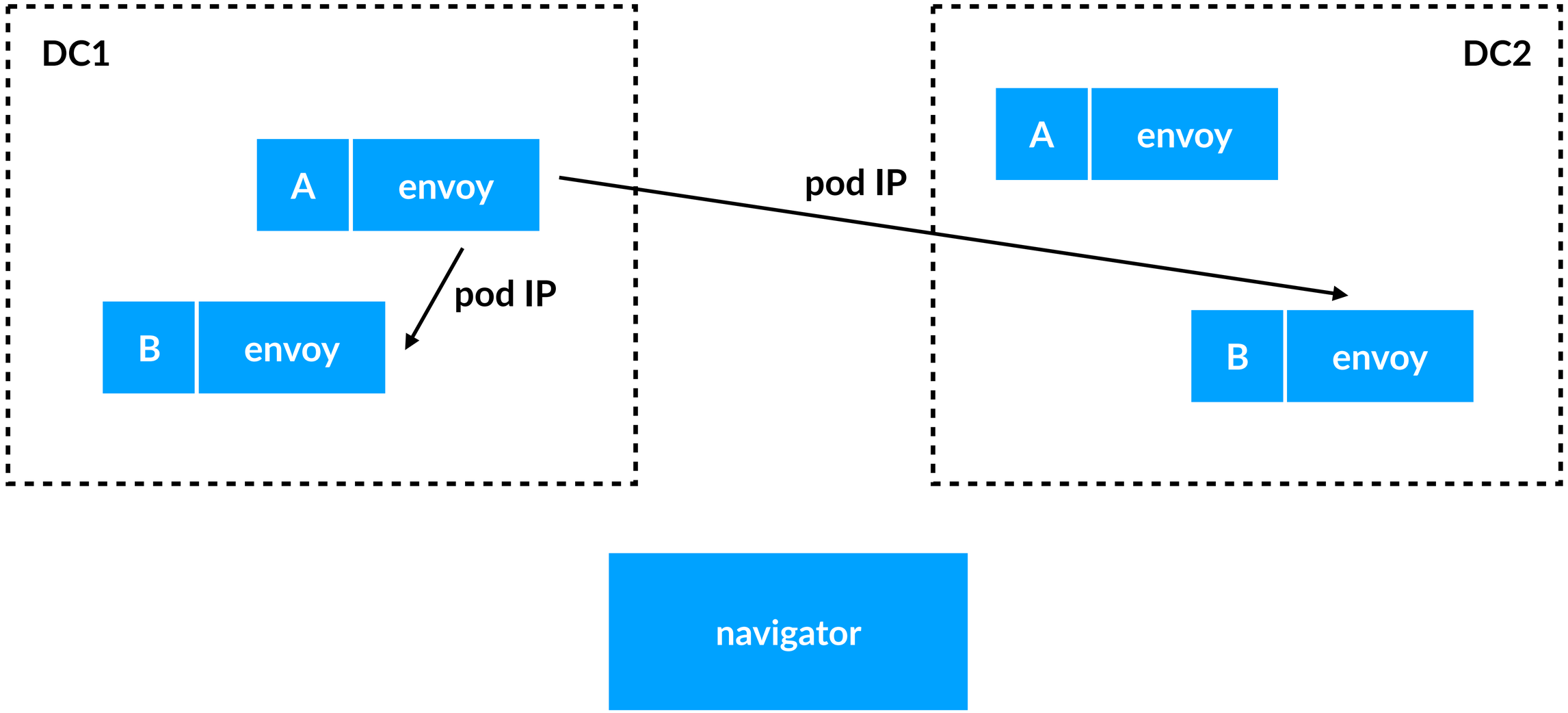
Navigator


DC1
A
envoy
B
envoy
DC2
A
envoy
B
envoy
navigator
dependencies
canary rules
custom resources

Navigator



Navigator

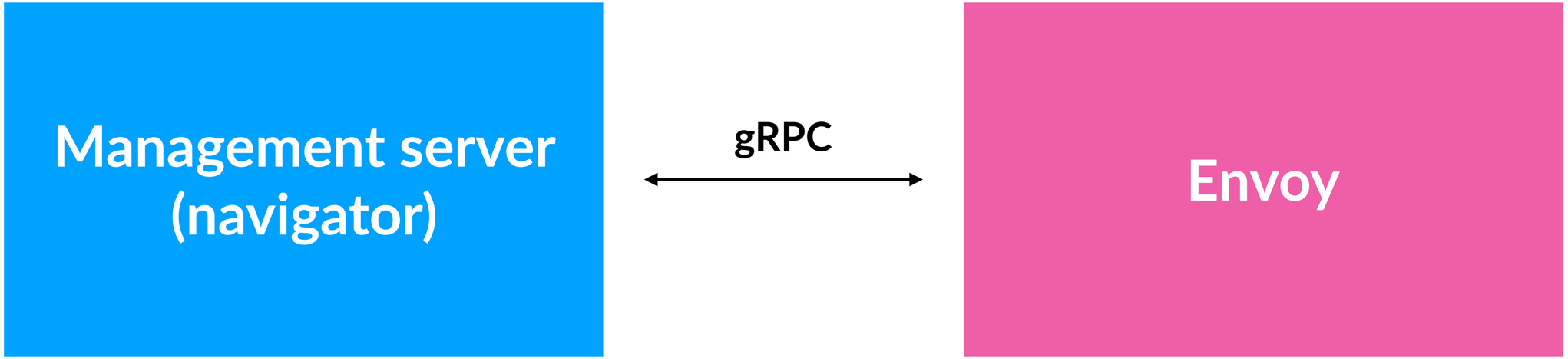


Kubernetes watchers

```
type ResourceEventHandler interface {
 OnAdd(obj interface{})
 OnUpdate(oldObj, newObj interface{})
 OnDelete(obj interface{})
}
```

https://github.com/kubernetes/client-go

Envoy XDS

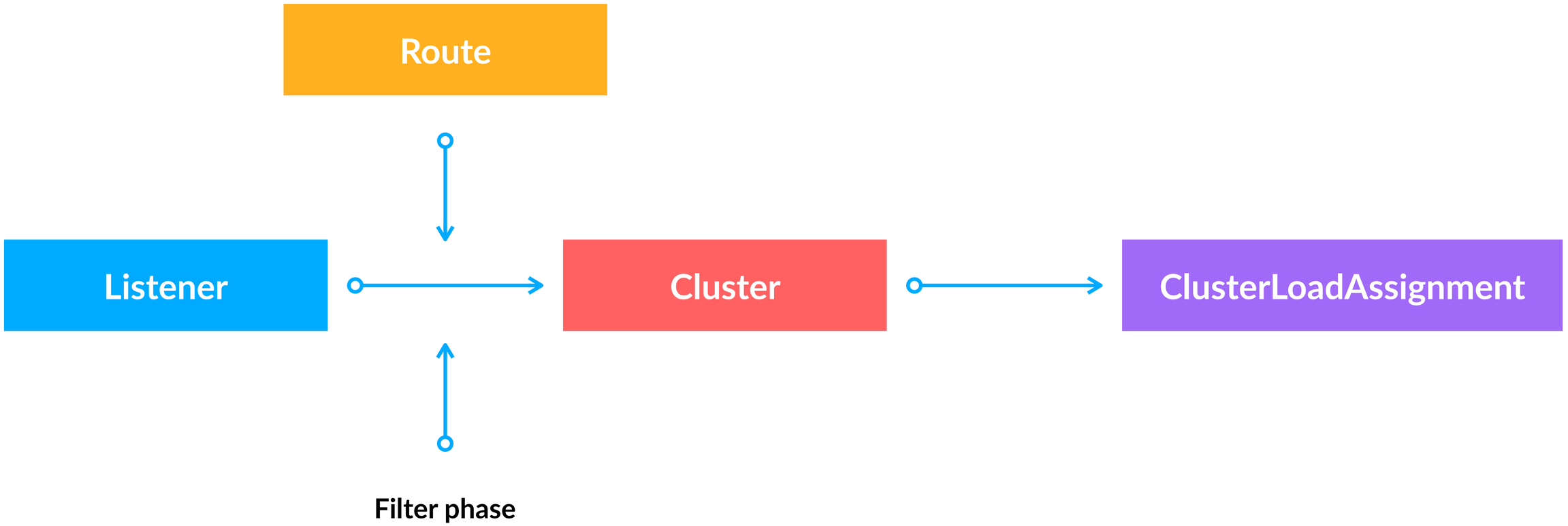


https://github.com/envoyproxy/go-control-plane

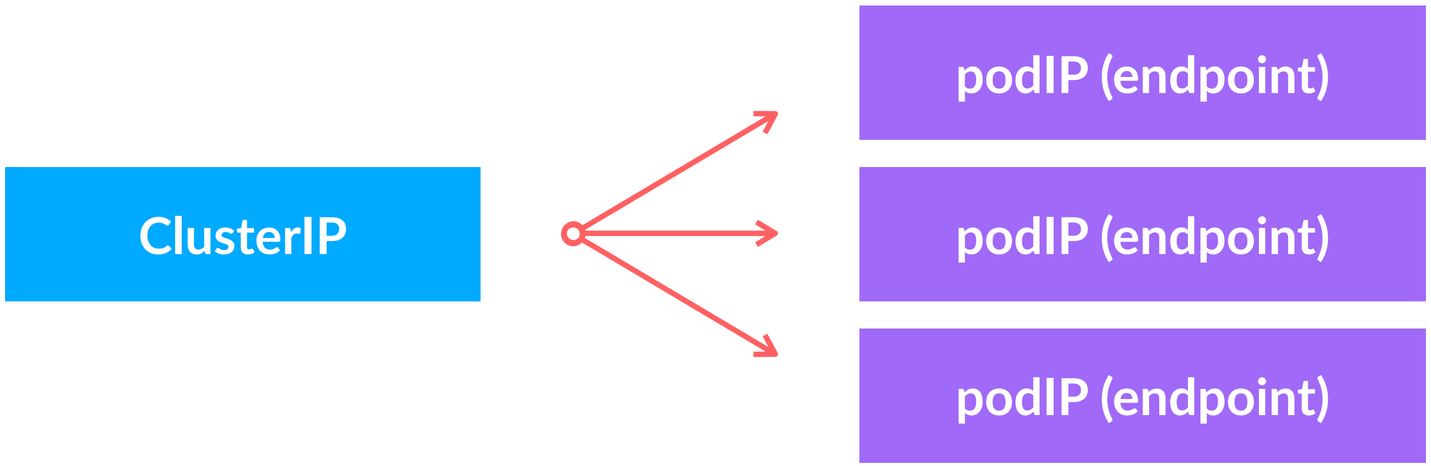
Envoy XDS

- LDS: envoy.api.v2.Listener
- CDS: envoy.api.v2.Cluster
- EDS: envoy.api.v2.ClusterLoadAssignment
- RDS: envoy.api.v2.Route

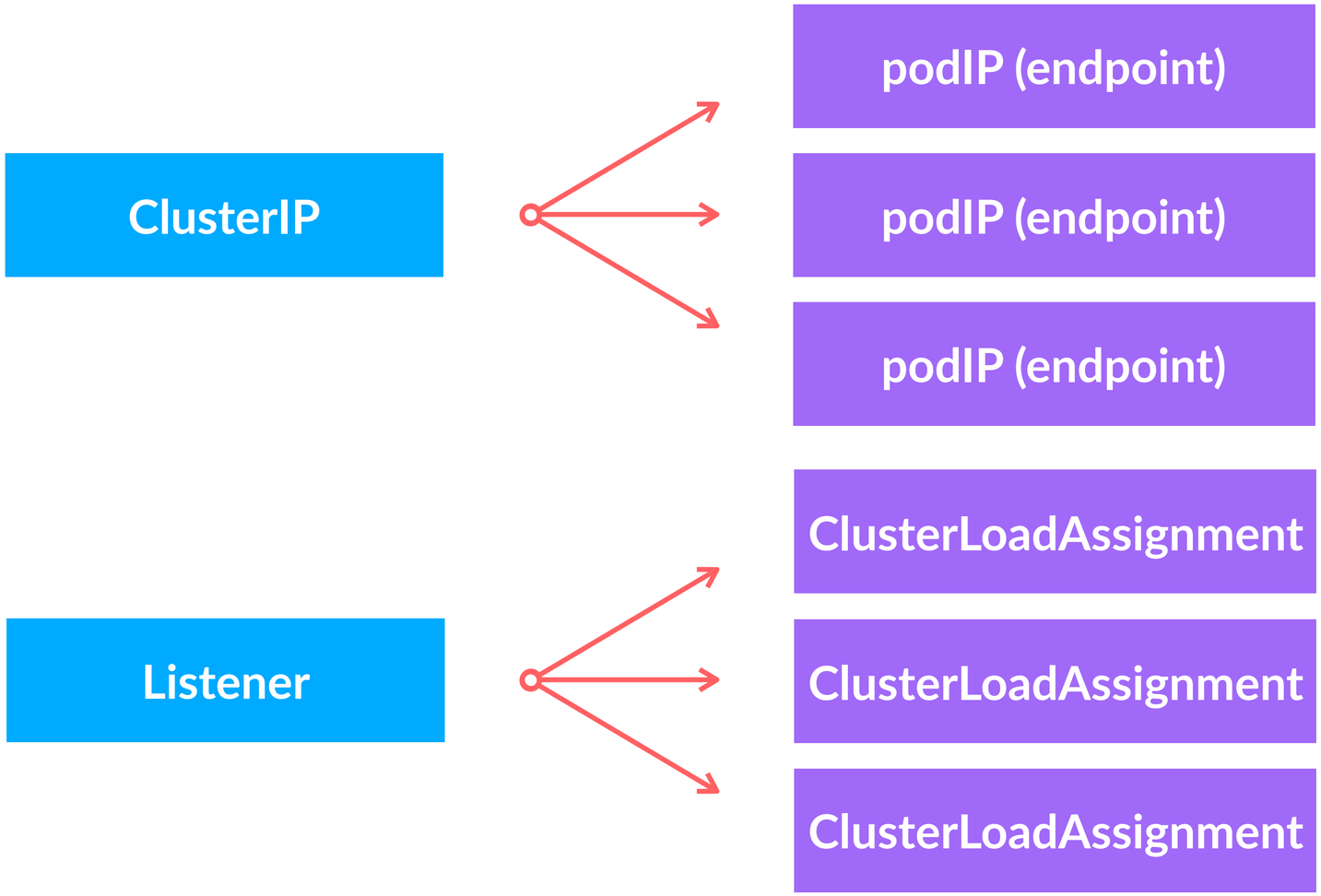
Envoy сущности



Kubernetes преобразование



Kubernetes преобразование и аналогичное в envoy



Kubernetes Service

```
apiVersion: v1
kind: Service
metadata:
  labels:
    service: item
  name: item
  namespace: item
spec:
  clusterIP: 10.200.255.180
  ports:
  - port: 8890
    protocol: TCP
    targetPort: 8890
  selector:
    service: item
```


Kubernetes Endpoints

```
apiVersion: v1
kind: Endpoints
metadata:
  labels:
    service: item
  name: item
  namespace: item
subsets:
- addresses:
  - ip: 10.21.148.210
    nodeName: srv1
    targetRef:
      kind: Pod
      name: item-57f4d65dd4-mt28j
      namespace: item
  - ip: 10.21.161.222
    nodeName: srv2
    targetRef:
      kind: Pod
      name: item-57f4d65dd4-26gsw
      namespace: item
```


Envoy listener

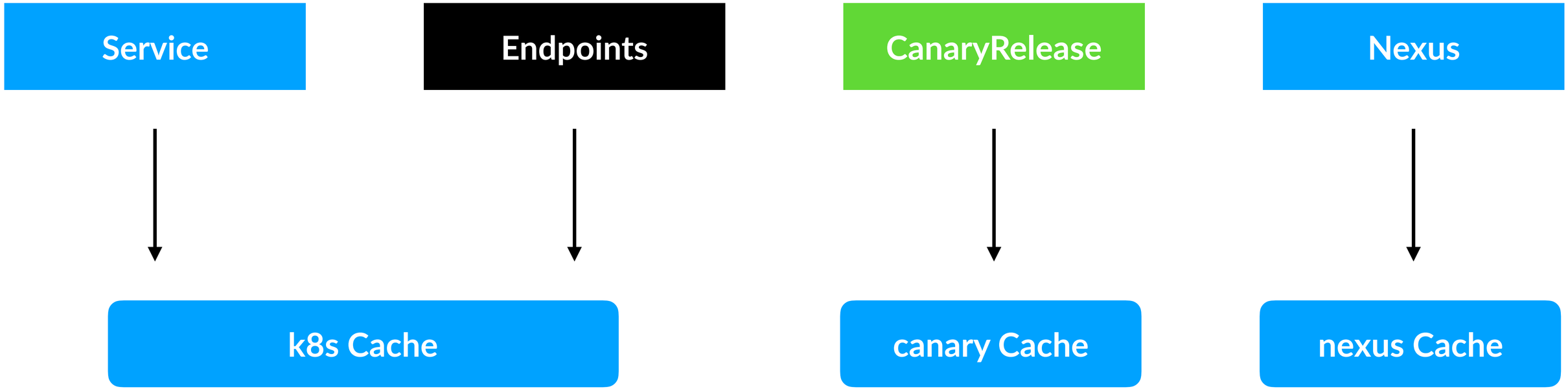
```
"listeners": [{
  "address": {
     "socket_address": { "address": "10.200.255.180", "port_value": 8890 }},
     "bind_to_port": false,
     "filter_chains": { … }
}]
```

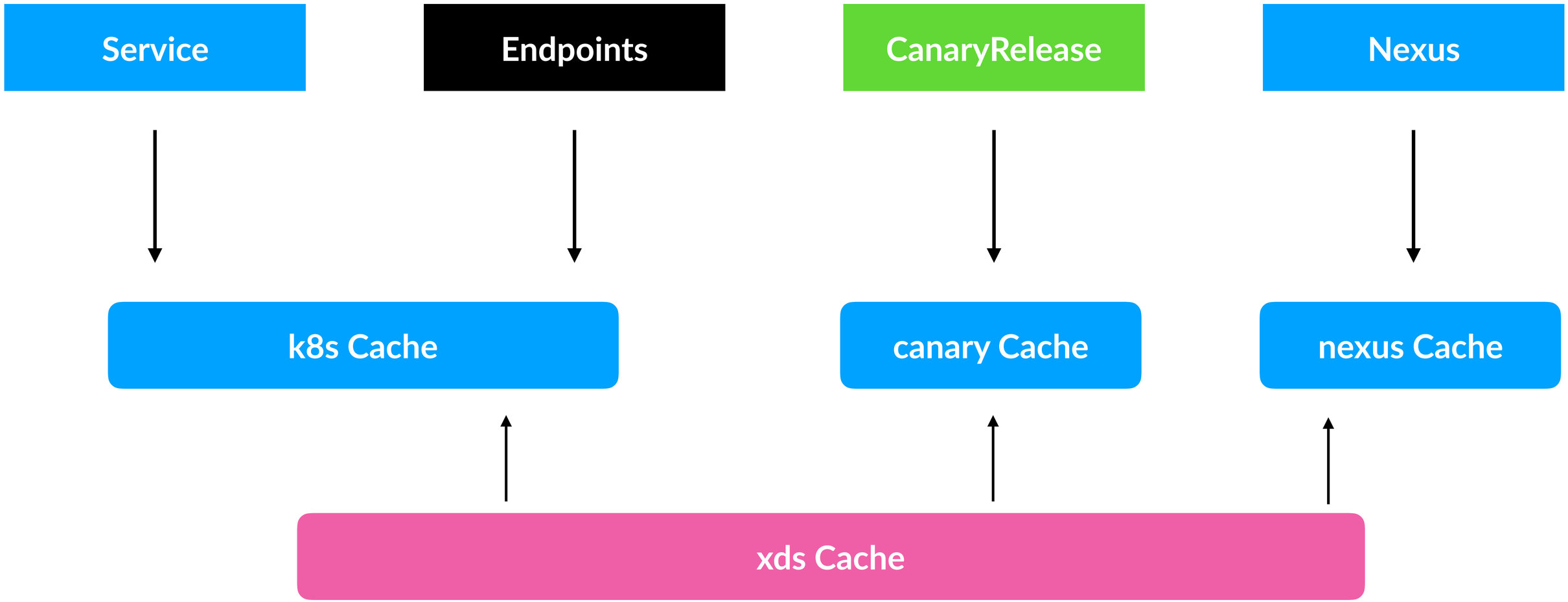
Cluster

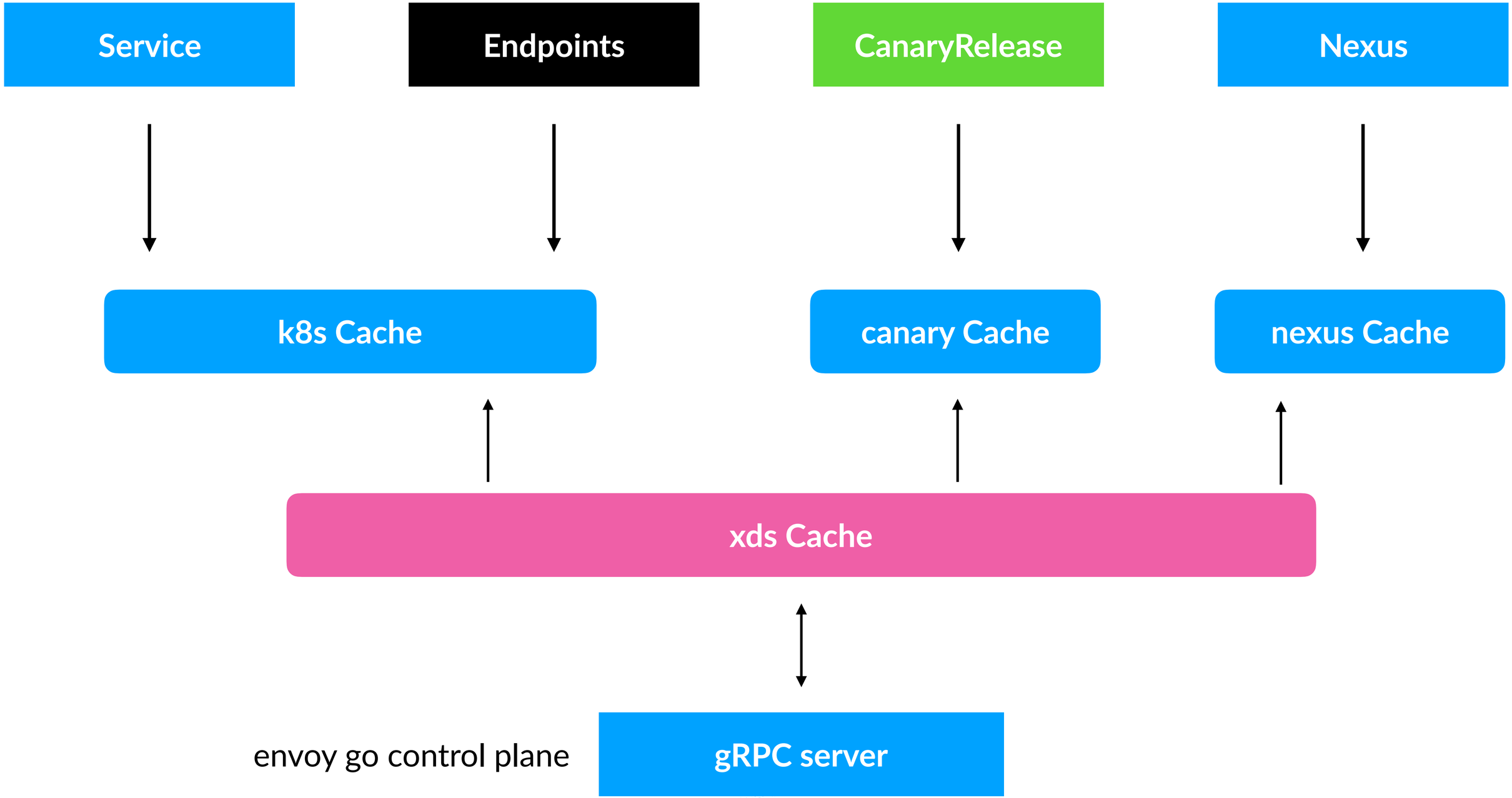
```
"clusters":[
    {
      "name": "item",
      "type": "STATIC",
      "connect_timeout": "1s",
      "hosts": [
        { "socket_address": { "address": "10.21.148.210", "port_value": 8890 } },
        { "socket_address": { "address": "10.21.161.222", "port_value": 8890 } },
      ]
    }
  ]
```

Посмотрим внутрь

kubernetes informers
(k8s client)


Service
Endpoints
CanaryRelease
Nexus
k8s Cache
canary Cache
nexus Cache


Service
Endpoints
CanaryRelease
Nexus
k8s Cache
canary Cache
nexus Cache
xds Cache



envoy go control plane

Ограничения

Ограничения

- Pod и Cluster IPs каждого кластера в разных подсетях

Ограничения

- Pod и Cluster IPs каждого кластера в разных подсетях
- Pod IP маршрутизируемы из любой точки системы

Ограничения

- Pod и Cluster IPs каждого кластера в разных подсетях
- Pod IP маршрутизируемы из любой точки системы
- Service сущности должны быть реплицированы по всем кластерам

Ограничения

- Pod и Cluster IPs каждого кластера в разных подсетях
- Pod IP маршрутизируемы из любой точки системы
- Service сущности должны быть реплицированы по всем кластерам
- Имена Service сущностей должны совпадать

Отказоустойчивость navigator

Отказоустойчивость navigator

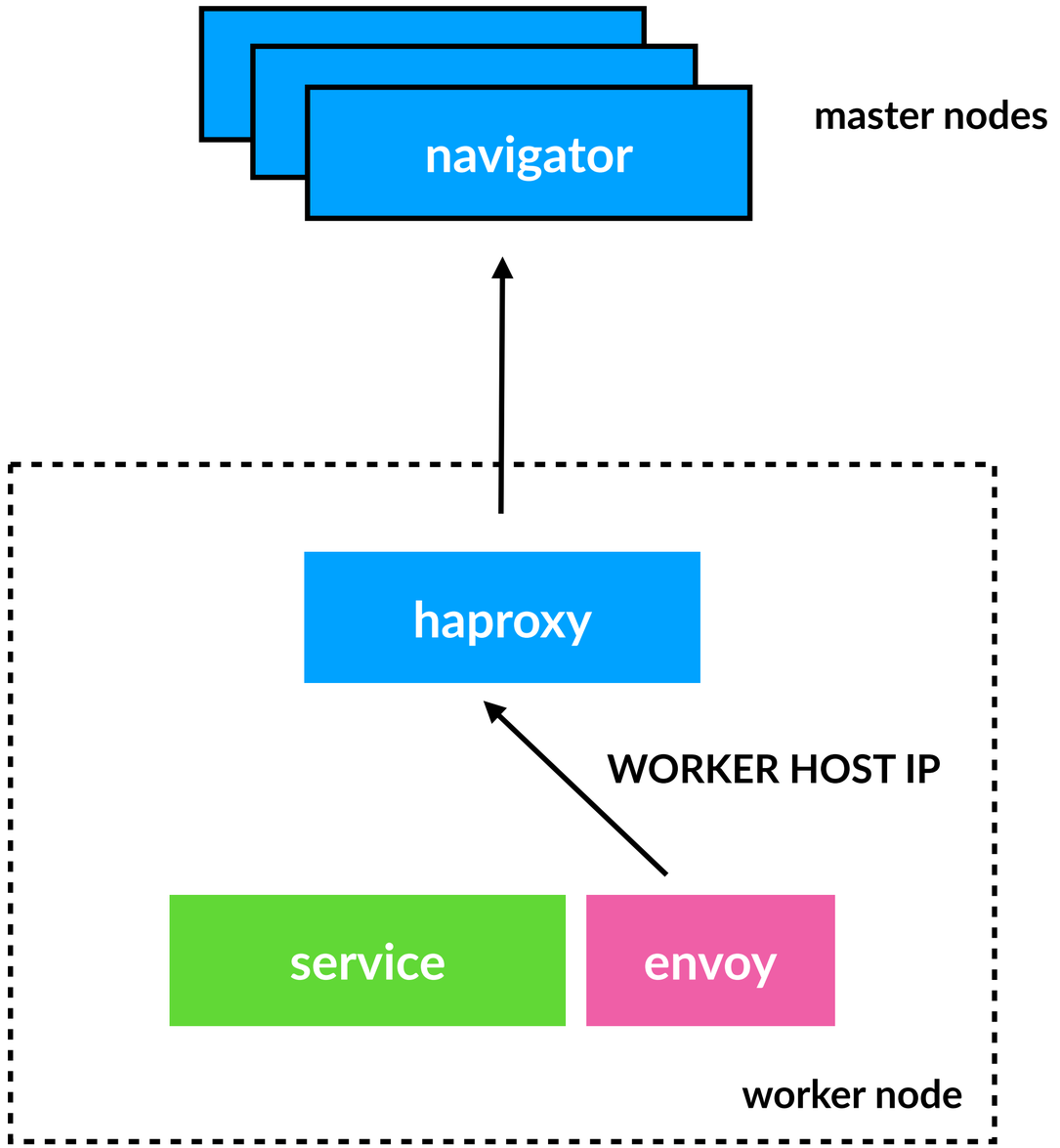
- Deploy на мастер ноды всех кластеров

Отказоустойчивость navigator

- Deploy на мастер ноды всех кластеров
- Балансировка в навигатор через хостовой балансер на каждой ноде

Отказоустойчивость navigator

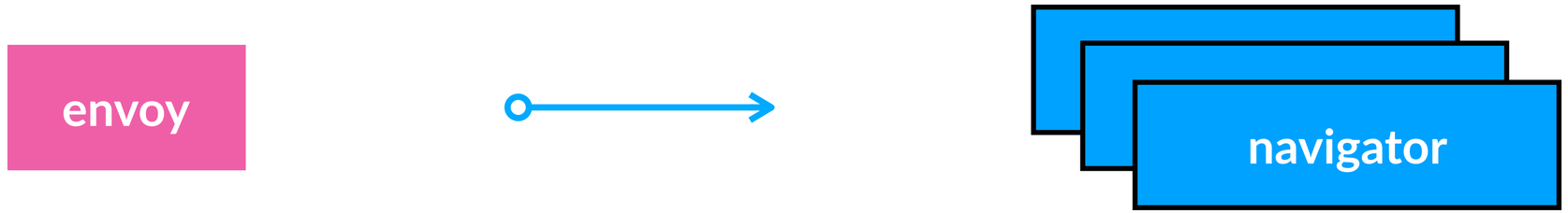
- Deploy на мастер ноды всех кластеров
- Балансировка в навигатор через хостовой балансер на каждой ноде
- Балансер находим через downward API


navigator
master nodes
haproxy
WORKER HOST IP
service
envoy
worker node

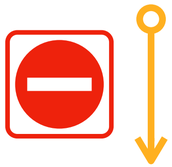
Отказоустойчивость navigator

Отказоустойчивость navigator

- Health check на envoy proxy через navigator


envoy
navigator

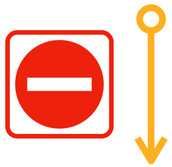
kube probe
/heathz



envoy

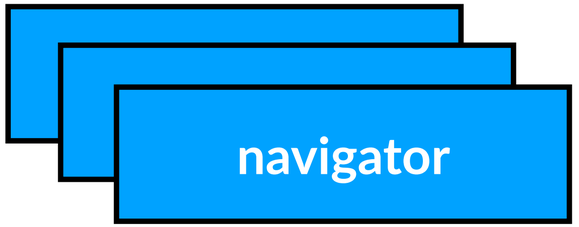
navigator

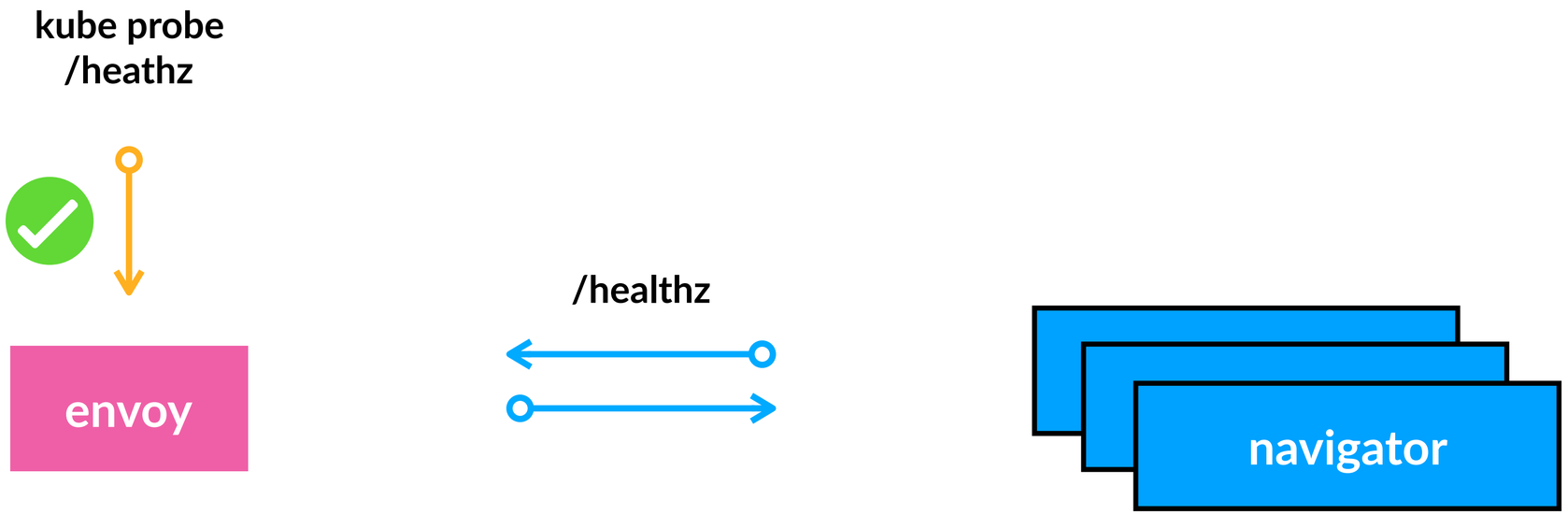
kube probe
/heathz



envoy

/healthz




kube probe
/heathz
envoy
/healthz
navigator

Возможности по балансировке

Возможности по балансировке

- Zone, region locality

Возможности по балансировке

- Zone, region locality
- Различные алгоритмы балансировки

Возможности по балансировке

- Zone, region locality
- Различные алгоритмы балансировки
- Active, passive health checks

Возможности по балансировке

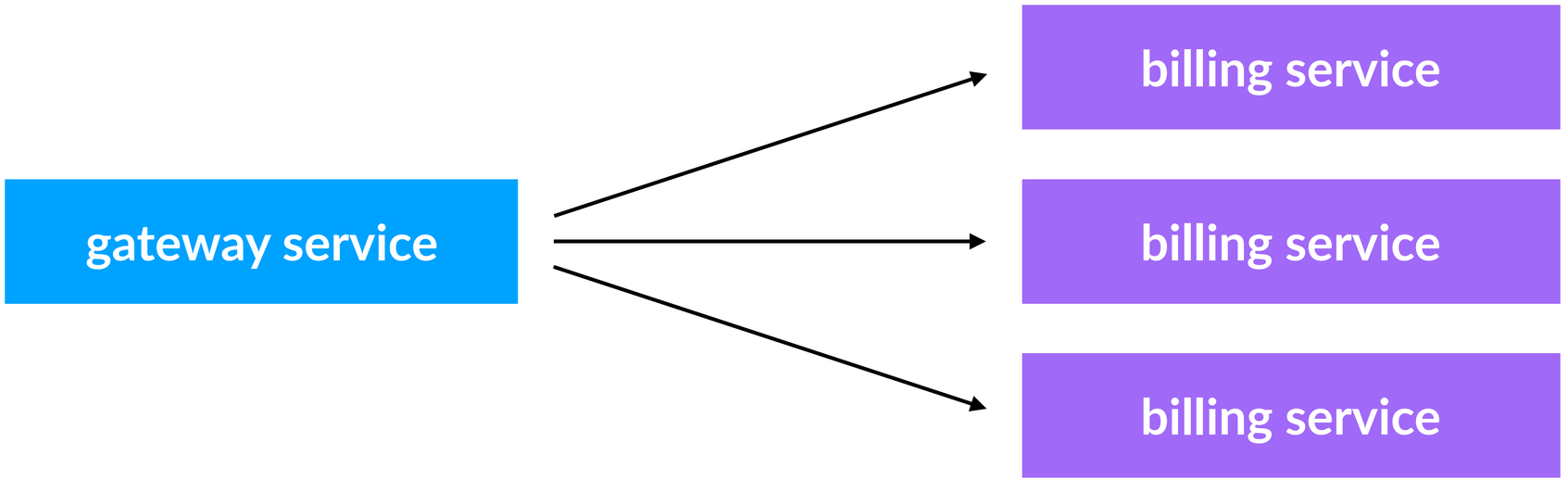
- Zone, region locality
- Различные алгоритмы балансировки
- Active, passive health checks
- L7 балансировка

Поддержка локальности

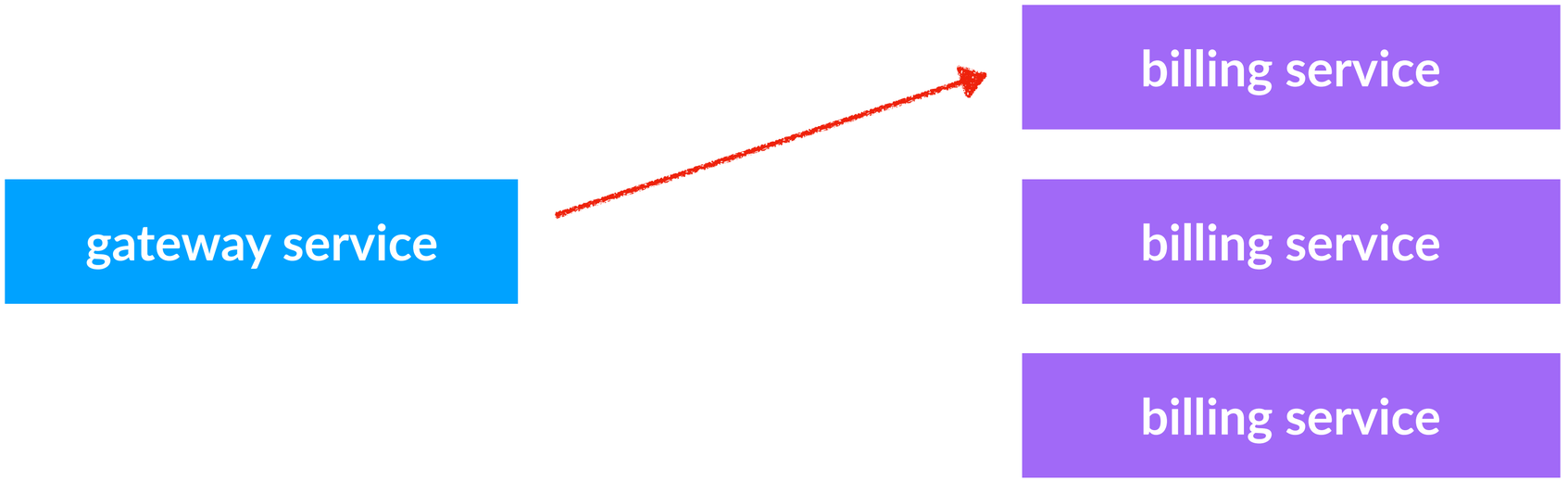
Priority levels

P=0 healthy endpoints	Traffic to P=0	Traffic to P=1
100 %	100 %	0 %
72 %	100 %	0 %
71 %	99 %	1 %
50 %	70 %	30 %
25 %	35 %	65 %
0 %	0 %	100 %

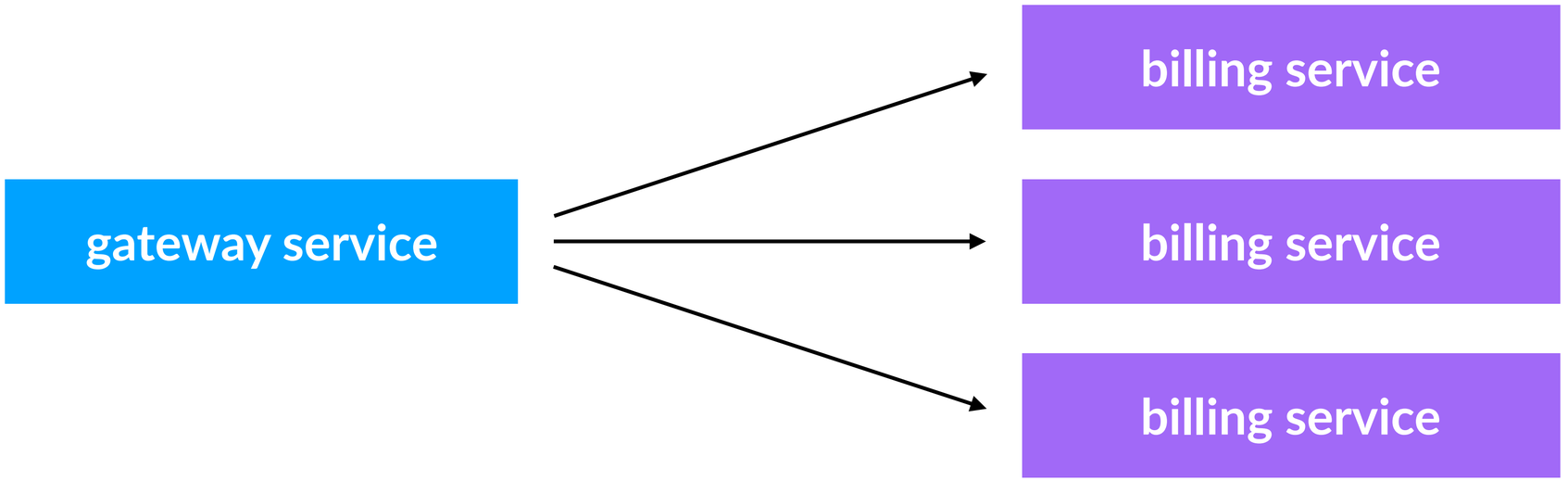
L4 балансировка



L4 балансировка



L7 балансировка



Canary deployments

Canary deploy

Canary deploy

- Поддержка multicluster

Canary deploy

- Поддержка multicluster
- Возможность использования того же sidecar (envoy)

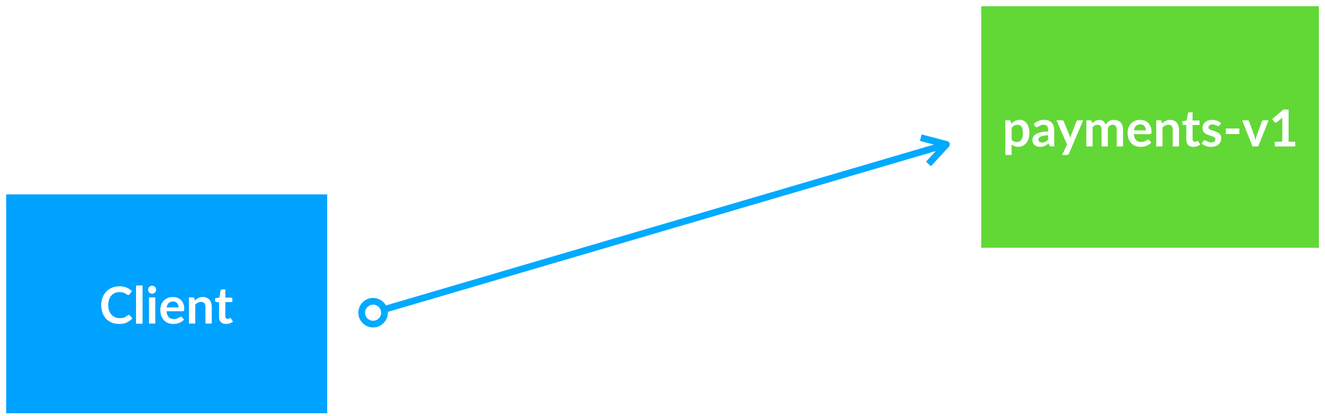
Canary deploy

- Поддержка multicluster
- Возможность использования того же sidecar (envoy)
- Единый control plane для service mesh

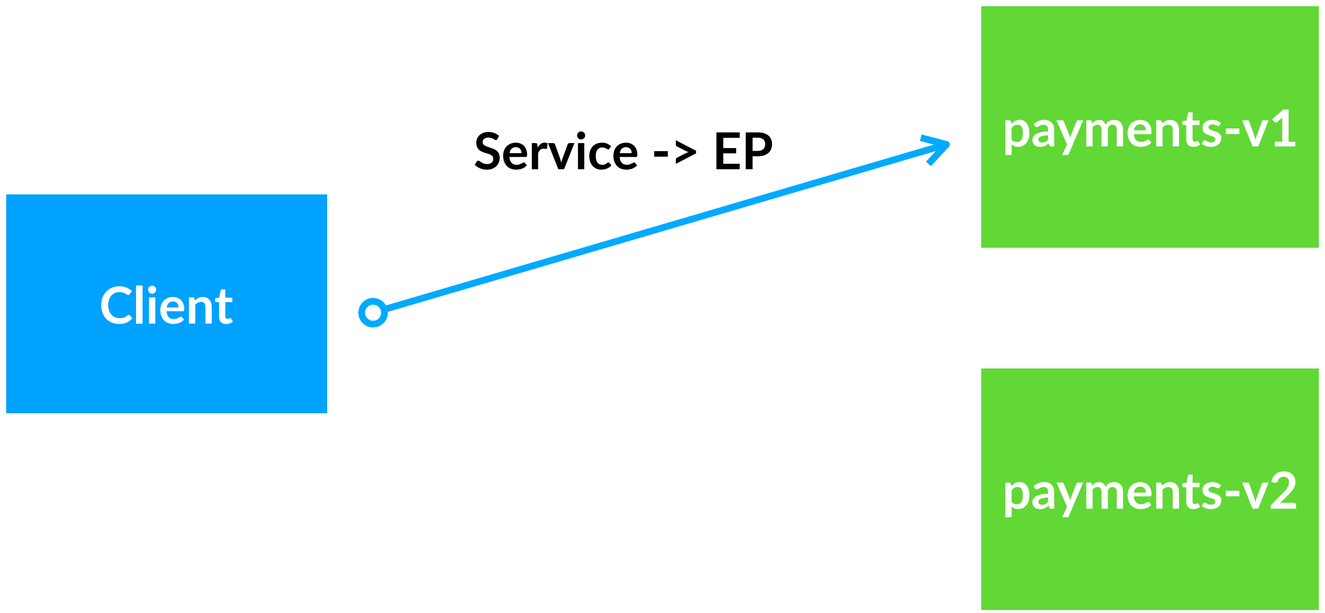
Canary

```yaml
apiVersion: "navigator.avito.ru/v1"
kind: CanaryRelease
metadata:
  name: payments-canary
  namespace: payments
spec:
  service: "payments"
  backends:
  - namespace: "payments"
    name: "payments-v2"
    weight: 30
  - namespace: "payments"
    name: "payments"
    weight: 70
```

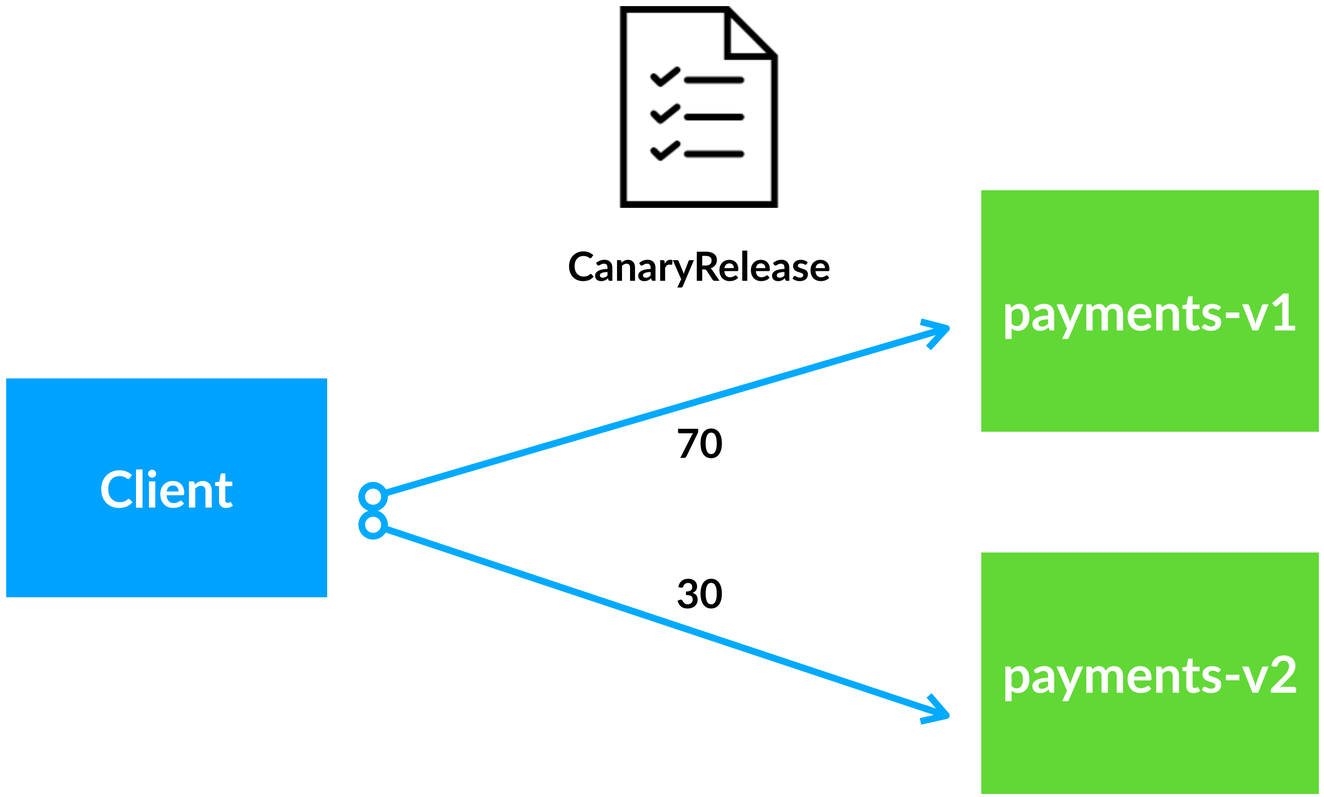

Canary deploy



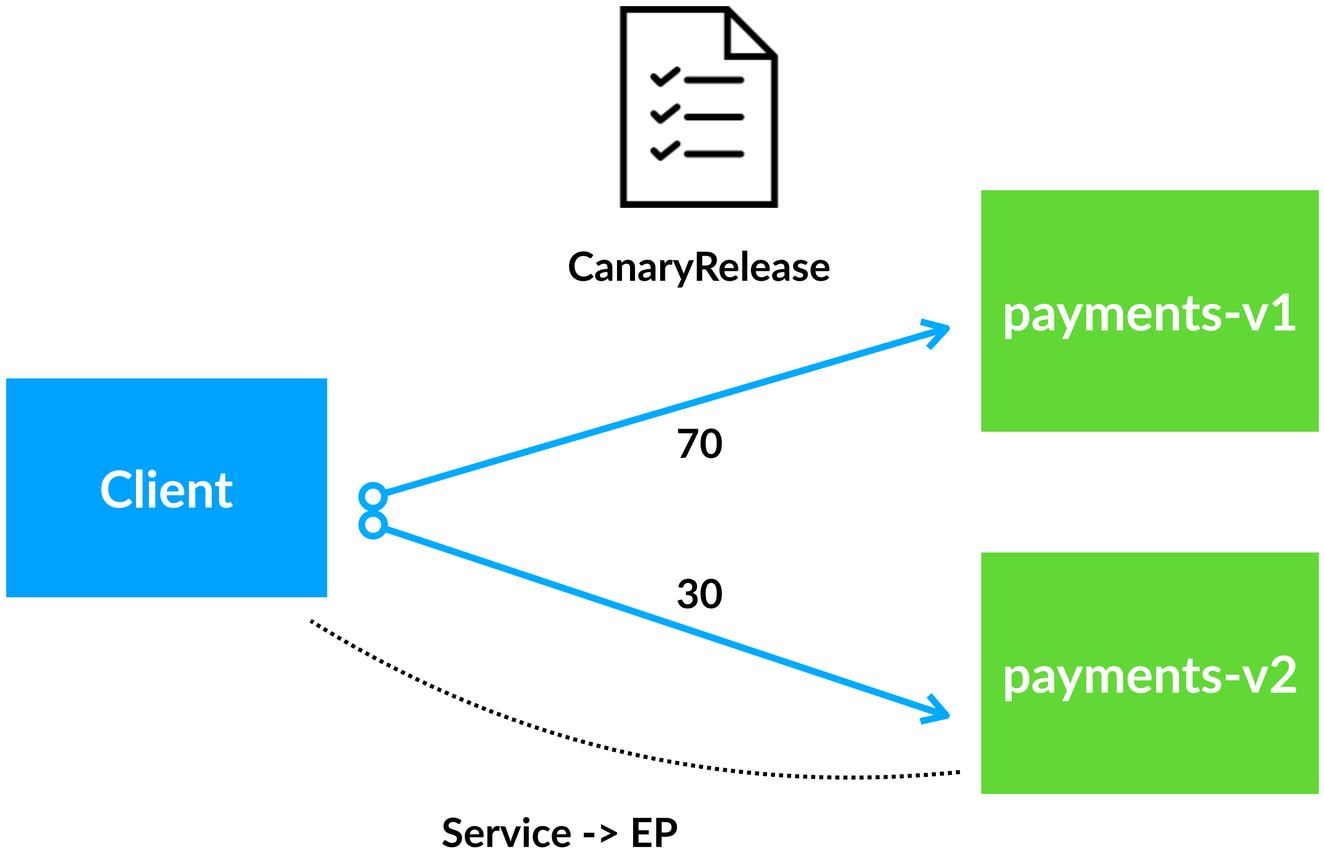
Canary deploy



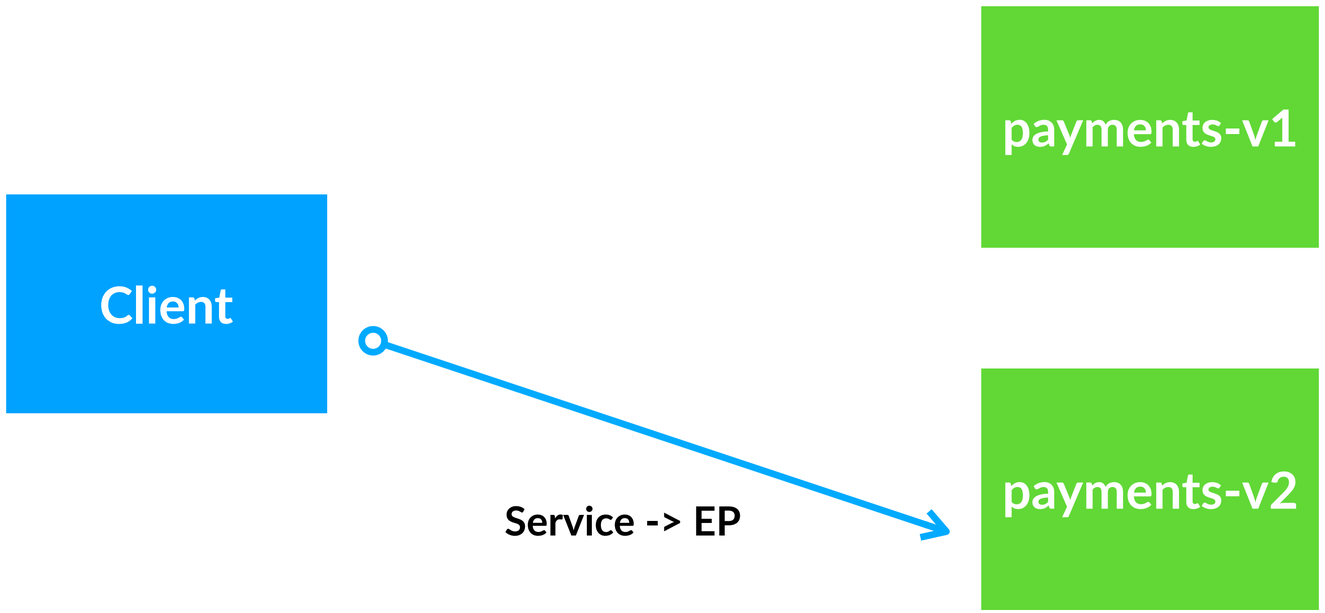
Canary deploy



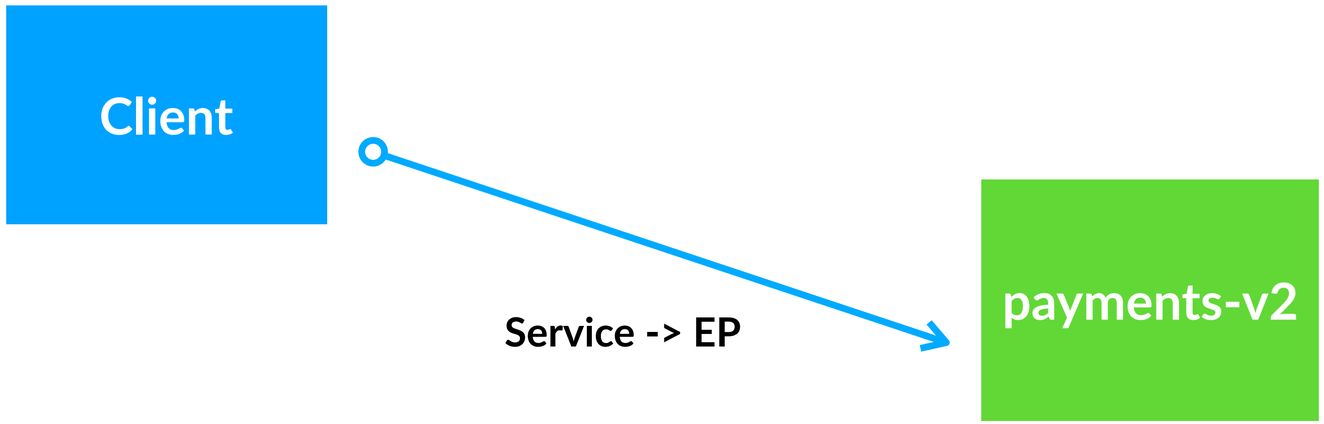
Canary deploy



Canary deploy



Canary deploy



Связи между сервисами

```yaml
apiVersion: "navigator.avito.ru/v1"
kind: Nexus
metadata:
  namespace: payments
  name: payments-v5
spec:
  appName: payments
  services:
  - namespace: "user"
    name: "" # all services in user NS
  - namespace: "billing"
    name: "processing"
```


Важные моменты о связях

Важные моменты о связях

- Держим параллельно связи нового релиза и предыдущего

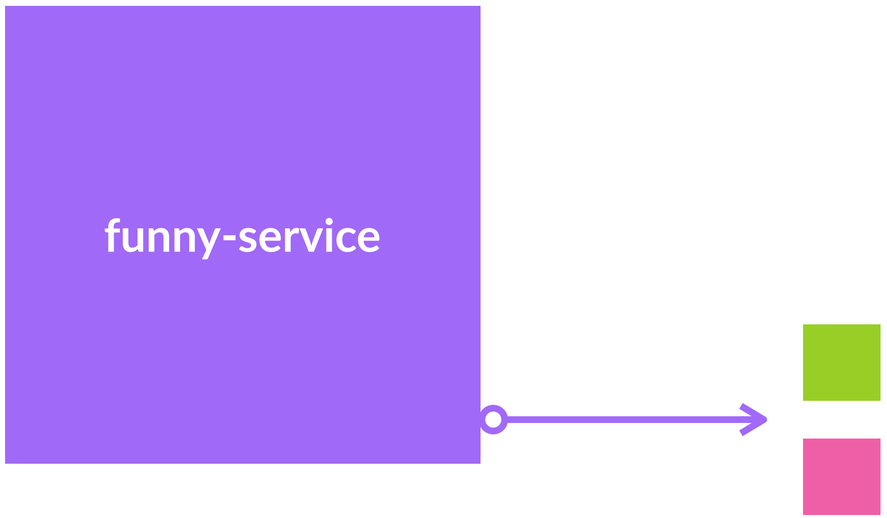
Важные моменты о связях

- Держим параллельно связи нового релиза и предыдущего
- Navigator делает merge всех связей для appName

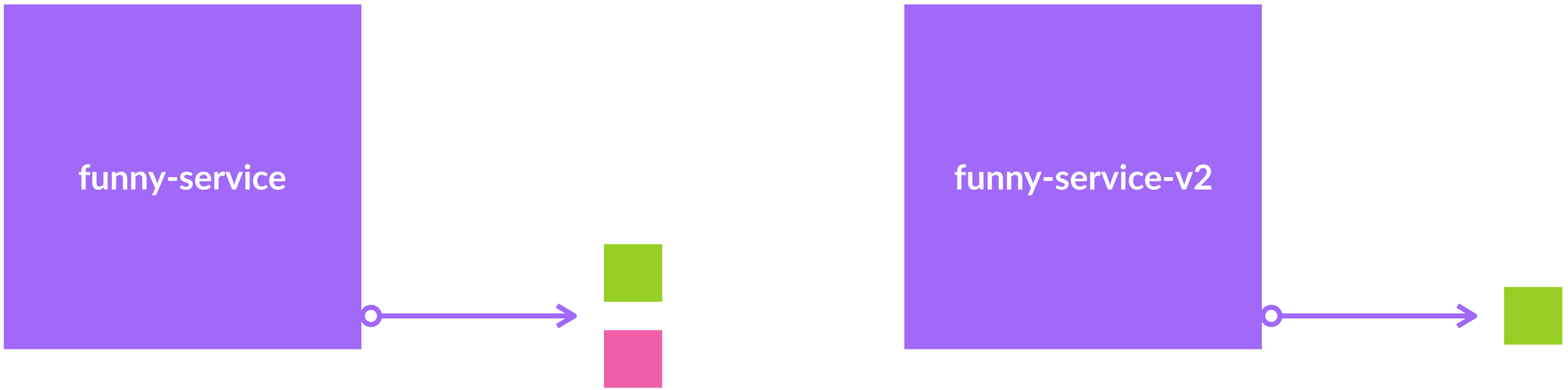
Важные моменты о связях

- Держим параллельно связи нового релиза и предыдущего
- Navigator делает merge всех связей для appName
- Система приходит в нужное состояние eventually

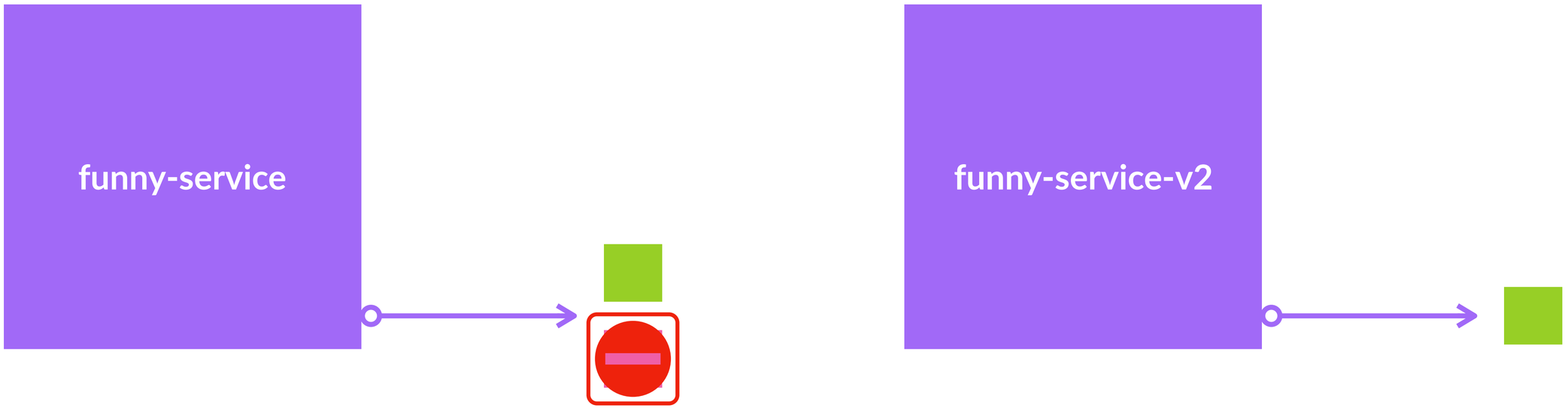
Зависимости при выкатках



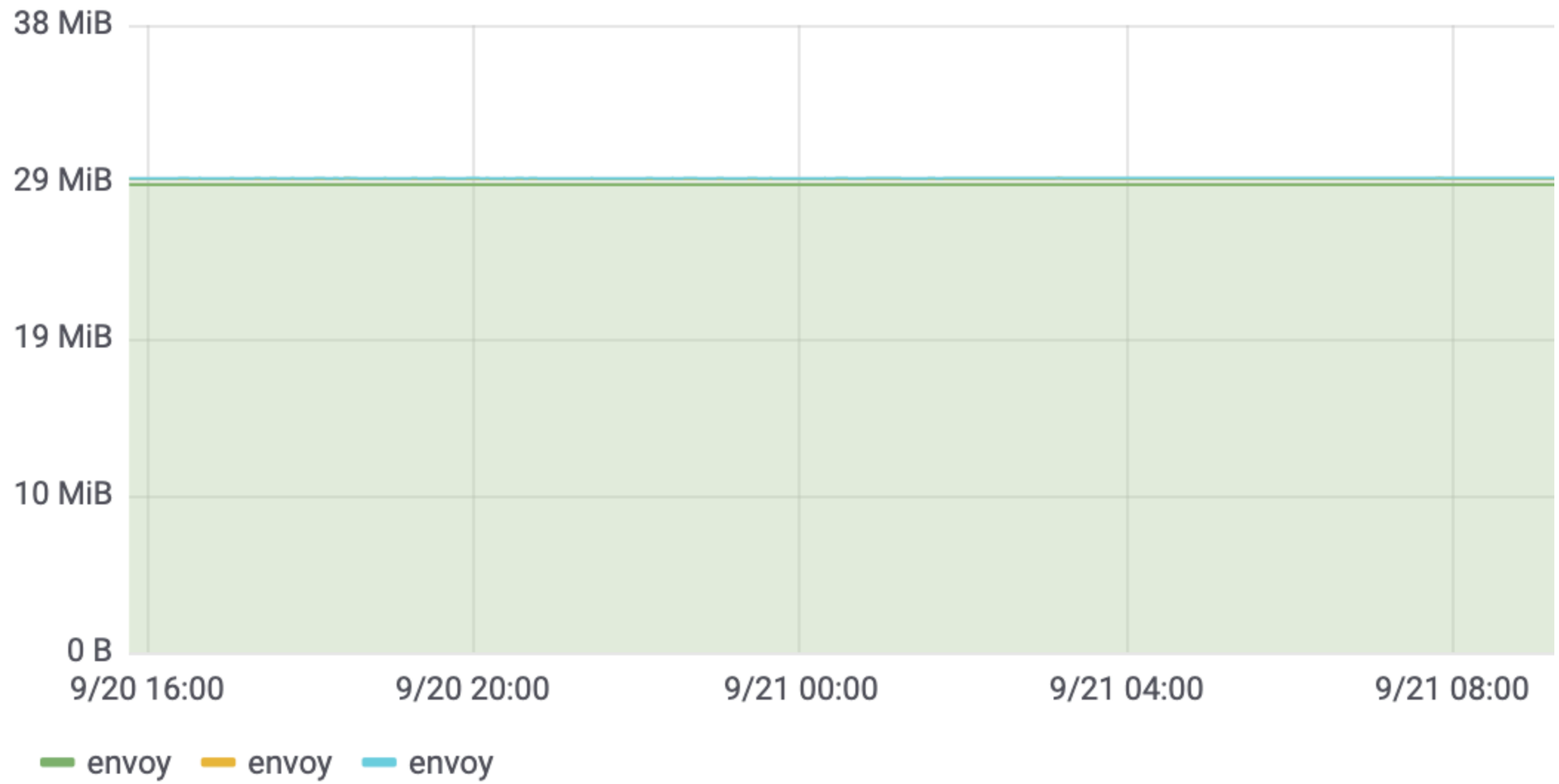
Зависимости при выкатках



Зависимости при выкатках



Envoy RAM Usage



Что еще получаем?

Что еще получаем?

- Сбор tracing информации

Что еще получаем?

- Сбор tracing информации
- Routing

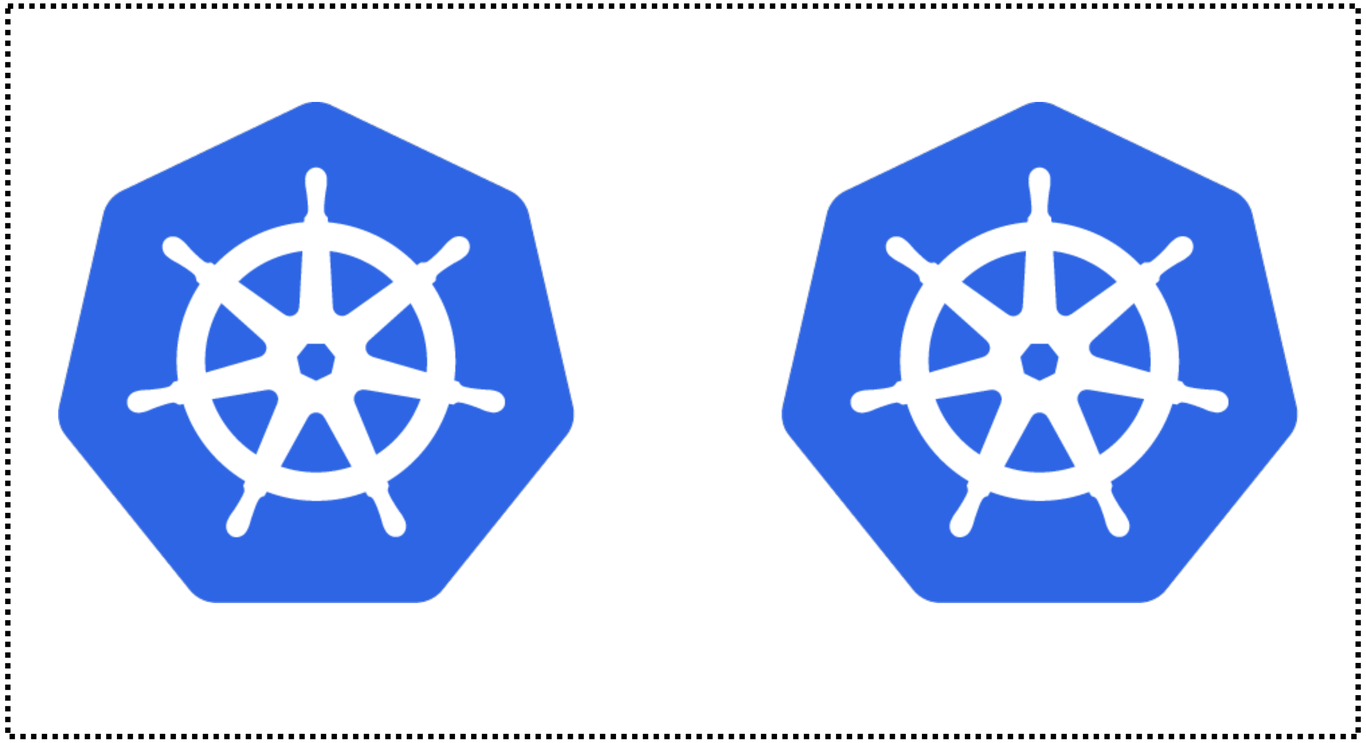
Что еще получаем?

- Сбор tracing информации
- Routing
- Мониторинг

Бонус. Обновление kubernetes

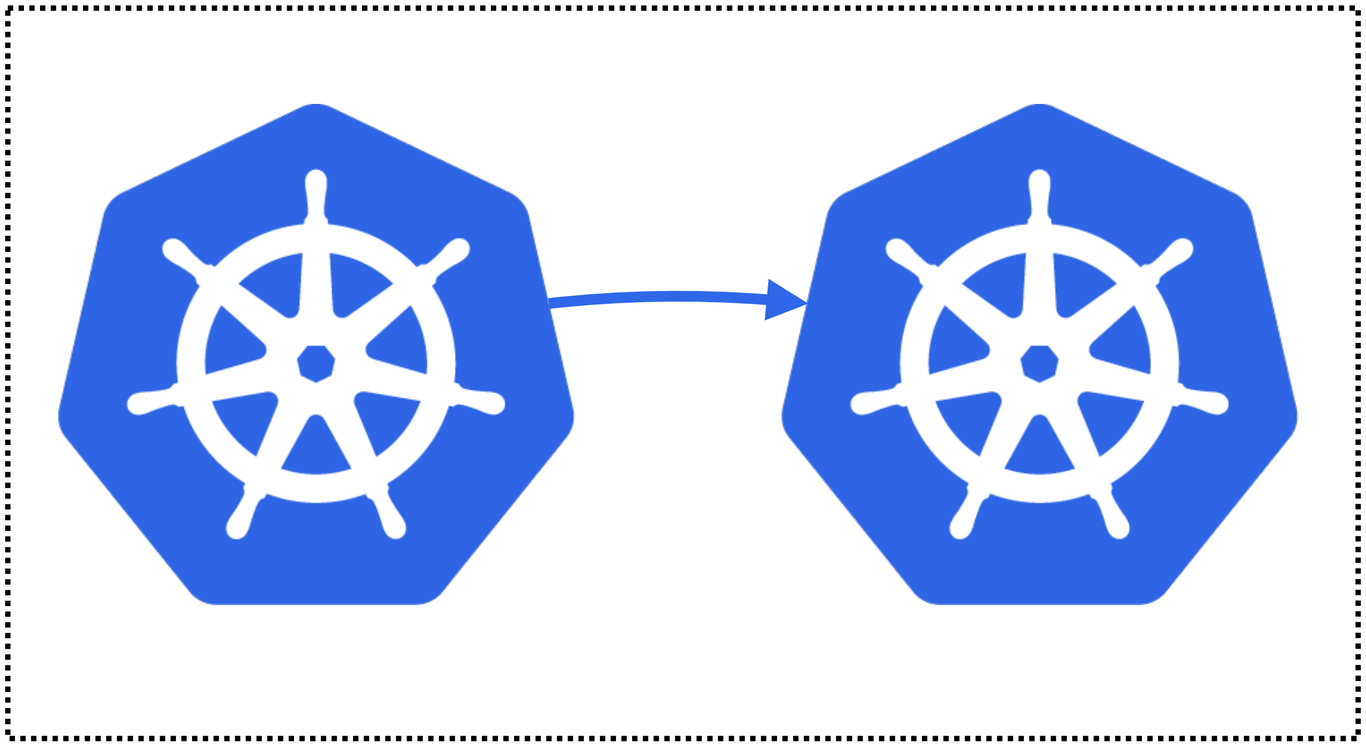
Бонус. Обновление kubernetes

Бонус. Обновление kubernetes



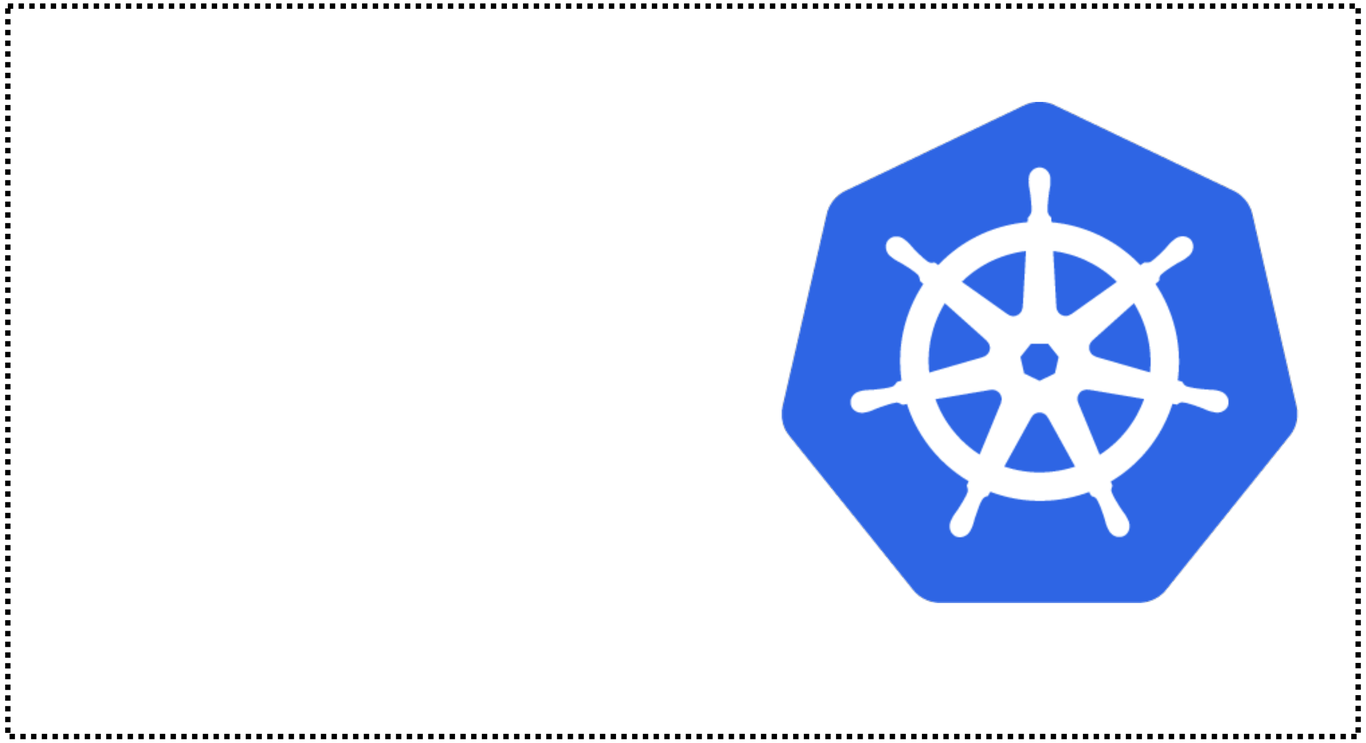
navigator

Бонус. Обновление kubernetes



navigator

Бонус. Обновление kubernetes

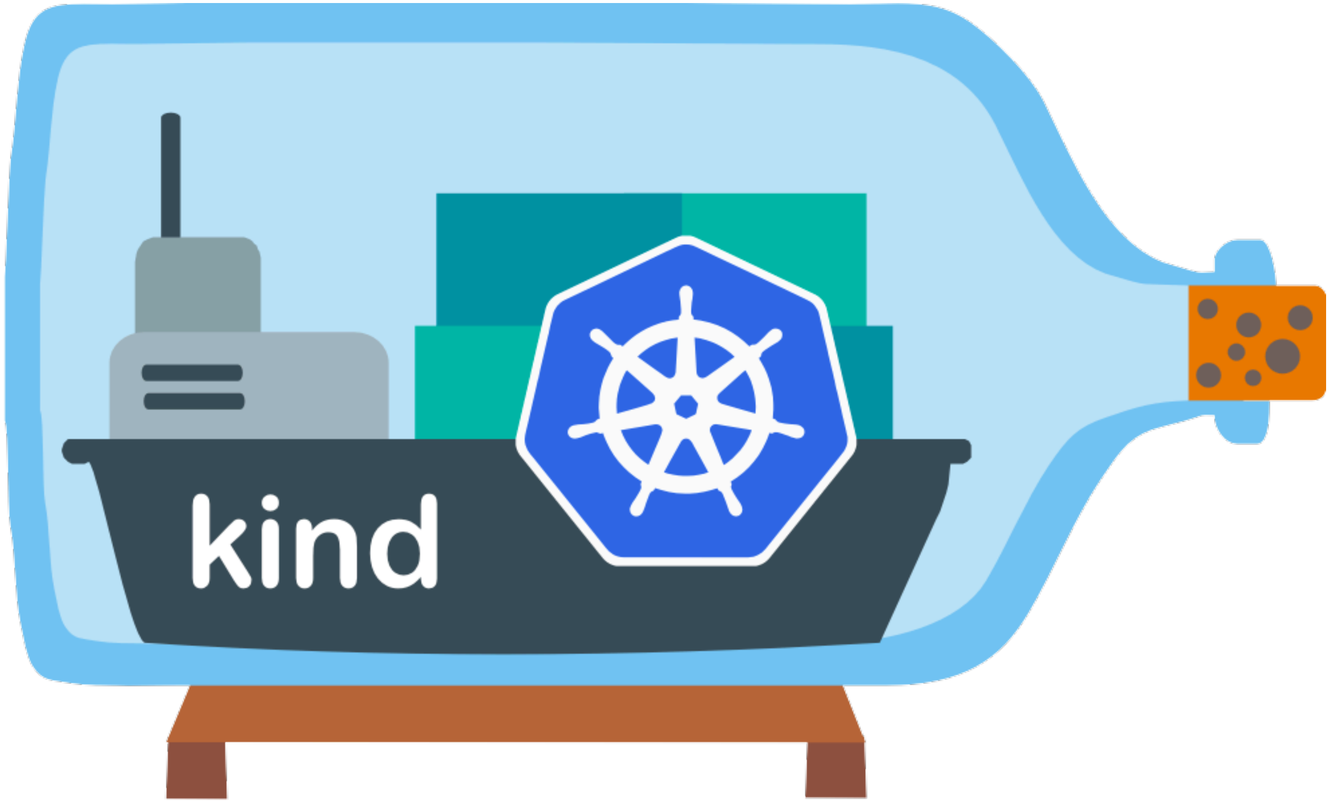


navigator

Бонус. Обновление kubernetes

navigator

Как тестируем?



Выводы

Выводы

- Service mesh позволяет удобно управлять трафиком извне приложений

Выводы

- Service mesh позволяет удобно управлять трафиком извне приложений
- Мы получаем возможность безболезненно использовать несколько изолированных кластеров и обновлять kubernetes

Выводы

- Service mesh позволяет удобно управлять трафиком извне приложений
- Мы получаем возможность безболезненно использовать несколько изолированных кластеров и обновлять kubernetes
- Внедрение политик возможно в любой гетерогенной среде

Выводы

- Service mesh позволяет удобно управлять трафиком извне приложений
- Мы получаем возможность безболезненно использовать несколько изолированных кластеров и обновлять kubernetes
- Внедрение политик возможно в любой гетерогенной среде
- Популярные инструменты все еще не позволяют с легкостью поднять необходимый сетап

Выводы

- Service mesh позволяет удобно управлять трафиком извне приложений
- Мы получаем возможность безболезненно использовать несколько изолированных кластеров и обновлять kubernetes
- Внедрение политик возможно в любой гетерогенной среде
- Популярные инструменты все еще не позволяют с легкостью поднять необходимый сетап
- Envoy предоставляет всю необходимую функциональность для построения service mesh

Спасибо

al.lukyanchenko@gmail.com
fb.com/alex.lukyanchenko
tg: @lookyan

Лукьянченко Александр
Lead engineer

Санкт-Петербург, 2019

Avito