AiSuite: 안정적으로 멀티 테넌트 AI 플랫폼 제공하기

박정욱

백인준

NAVER SEARCH

NAVER DEVIEW 2023

CONTENTS

1. AiSuite: Kubeflow 기반의 멀티 테넌트 AI 플랫폼 in NAVER

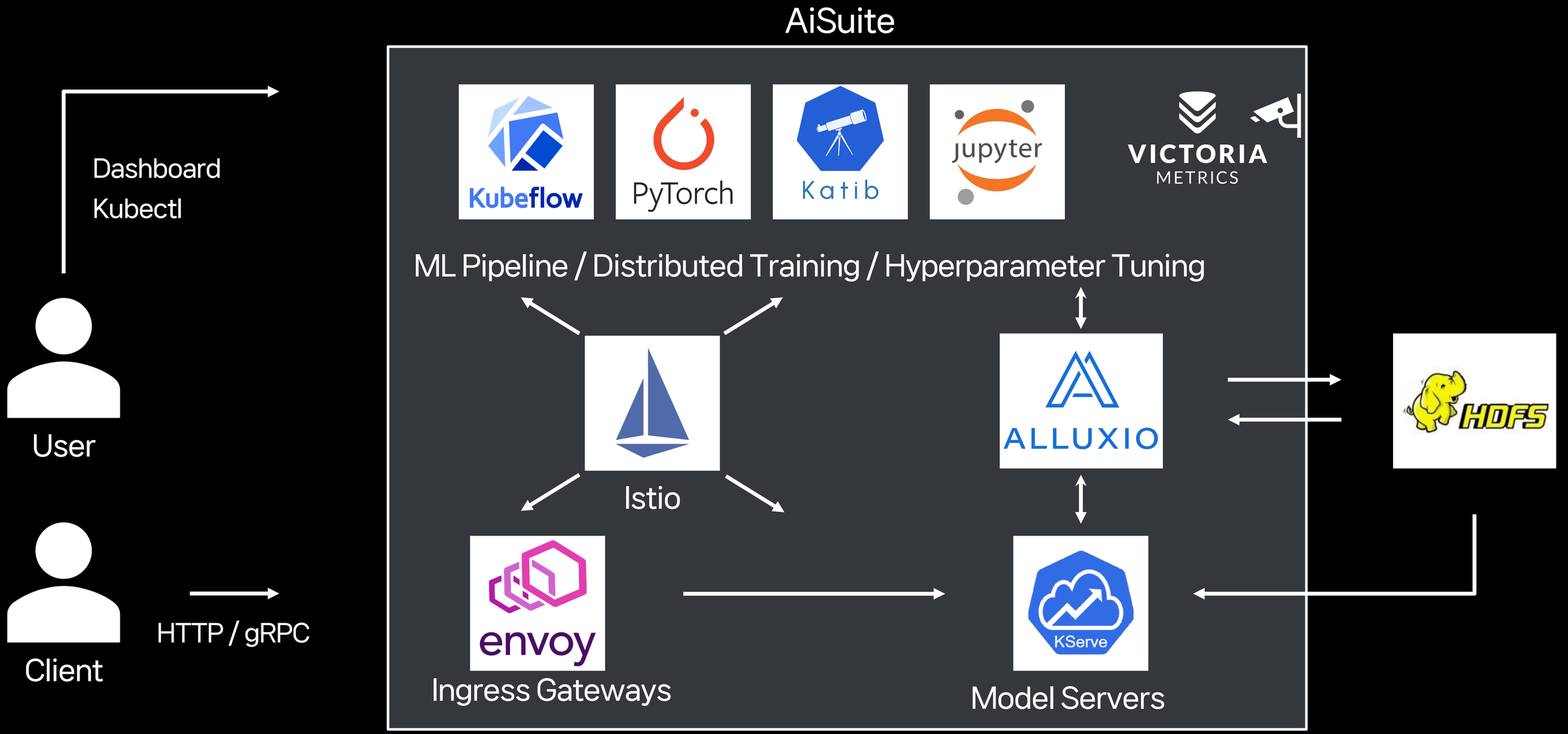
1.1 AiSuite

NAVER
DEVIEW
2023

Kubeflow 기반의 멀티 테넌트 AI 플랫폼

- 다양한 ML toolkit 지원 (Jupyter, Kubeflow Pipelines, Katib, MLflow)
- 모델 서빙 지원 (Kserve, Istio)
- 용도에 맞는 다양한 스토리지 지원 (Ceph RBD, HDFS, DDN Exascaler)
- 효율적인 데이터 사용을 위한 데이터 캐싱 레이어 지원 (Alluxio)
- Multi-Tenancy / HPC workload를 위한 커스텀 스케줄링 지원
- 분산 학습을 위한 고성능 네트워크 지원
- [DEVIEW 2021] AiSuite: Better AI Model Serving and MLOps with Kubeflow

1.2 AiSuite: Overview



1.2 AiSuite: Overview

NAVER
DEVIEW
2023

사용자 namespace에 배포되는 application

- Model Servers
- Notebooks
- Distributed Training pods
- Pipeline worker pods
- Model registry
- User Grafana dashboard
- ...

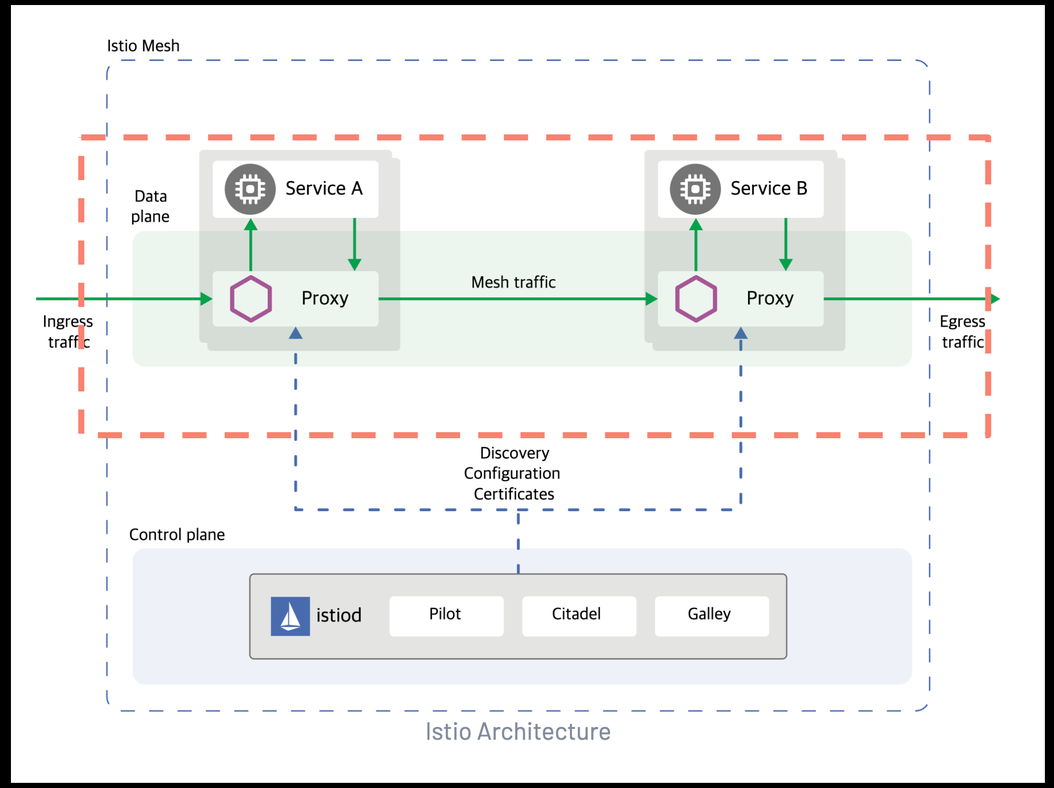
전체 클러스터가 공유하는 application

- Istio Control Plane (istiod)
- Ingress gateways
- Monitoring components
- Alluxio
- Operators
- ...

2. 멀티 테넌트 모델 서빙 안정적으로 제공하기: Istio 최적화

2.1 Istio 작동 방식 이해하기

NAVER
DEVIEW
2023



Istio Architecture

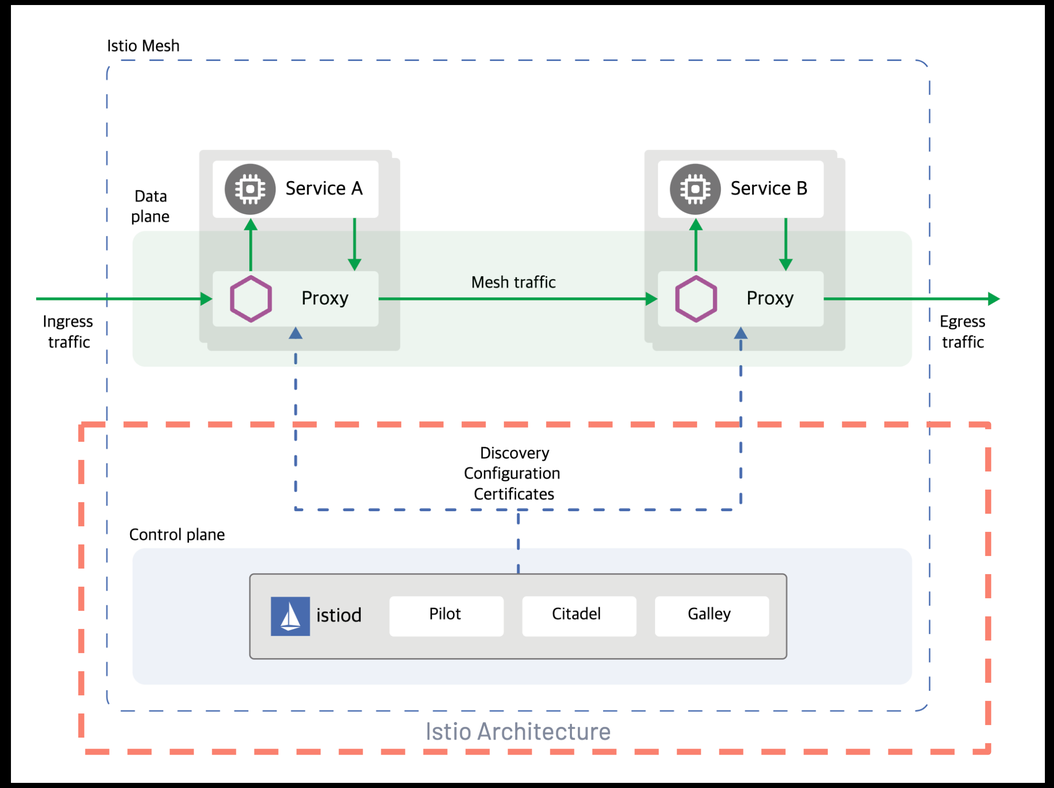
Data Plane

- Envoy proxy (L7 proxy)
- Kubernetes pod에 들어오고 / 나가는 트래픽을 통제

https://istio.io/latest/docs/ops/deployment/architecture/

2.1 Istio 작동 방식 이해하기

NAVER
DEVIEW
2023



클러스터 내 다른 서비스에 대한 정보, 라우팅 설정 등

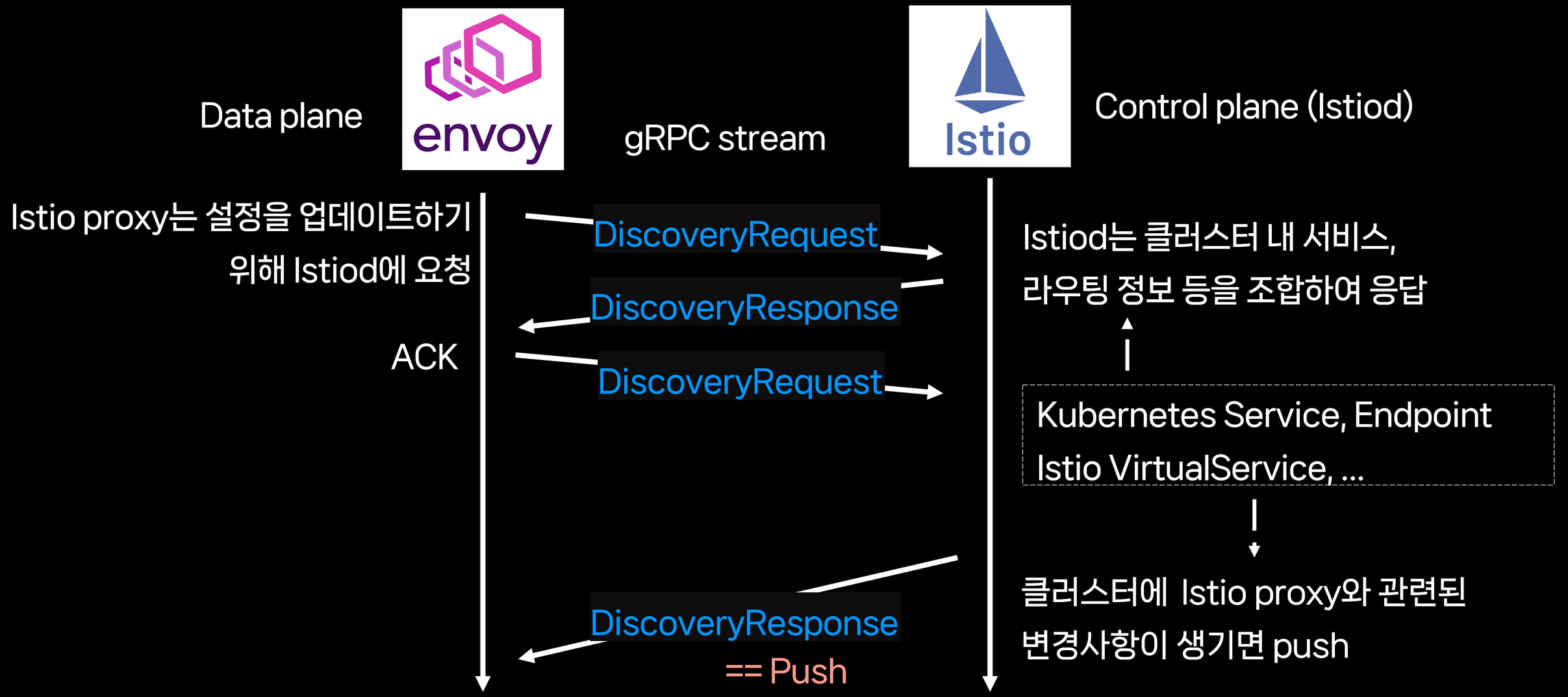
Control Plane (Istiod)

- Data plane 관리
- 클러스터 상태 / Istio CR에 따라 동적으로 data plane 설정 변경 (push)
- 동적으로 data plane 설정을 변경하는 API - xDS (* discovery service)

https://istio.io/latest/docs/ops/deployment/architecture/

2.1 Istio 작동 방식 이해하기 (xDS)

NAVER
DEVIEW
2023



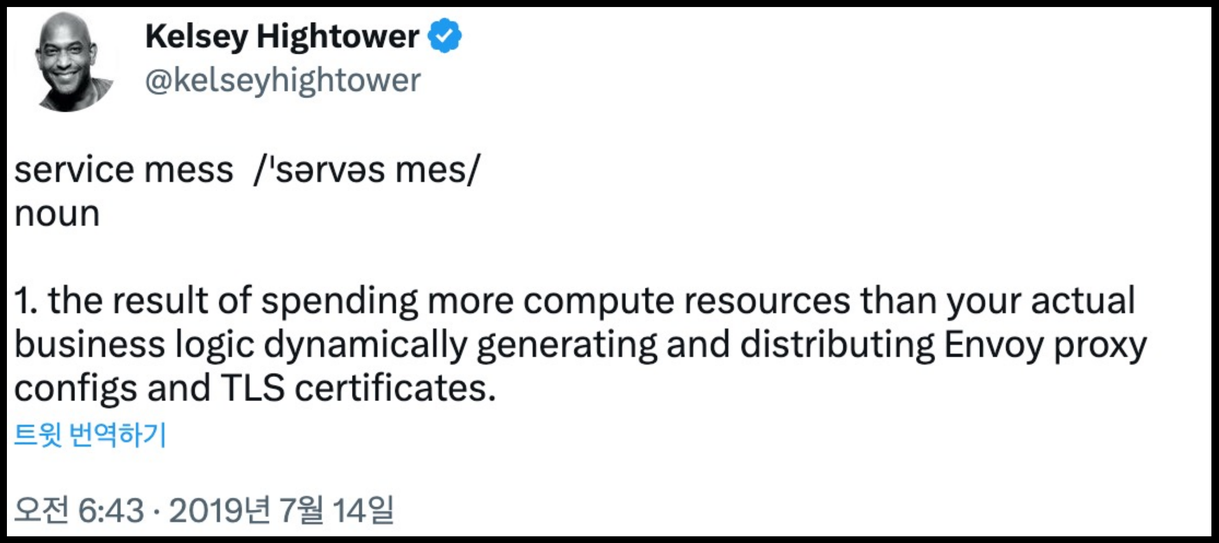
2.1 Istio 작동 방식 이해하기

NAVER
DEVIEW
2023

Istiod와 Istio proxy는 push를 처리하기 위해 클러스터의 컴퓨팅 리소스를 사용합니다.

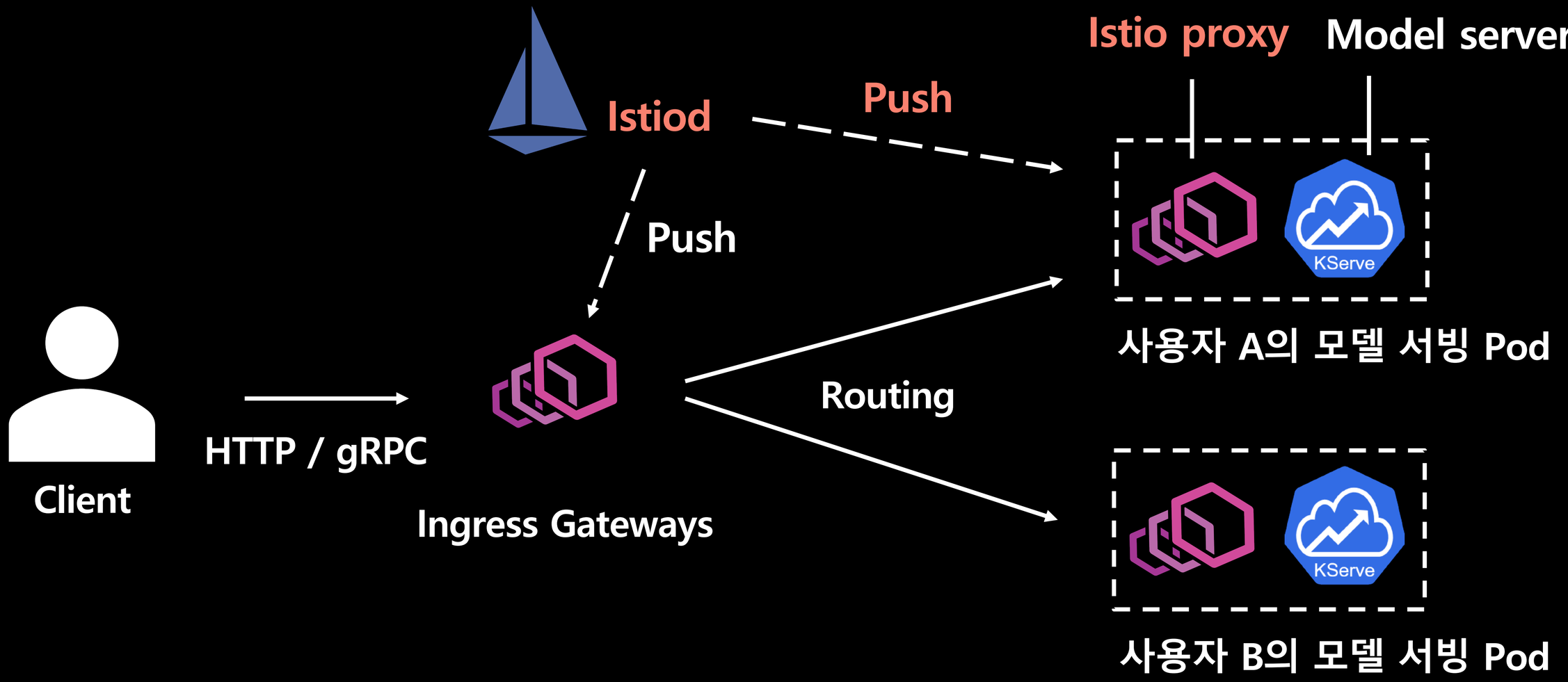
- Istiod는 필요한 설정들을 생성하고 Istio proxy들에게 이를 push해야 합니다.
- Istio proxy들은 Istiod로부터 설정을 받고, 이를 메모리에 유지해야 합니다.

→ CPU, memory, network bandwidth 사용



2.1 Istio in AiSuite Model Serving

NAVER
DEVIEW
2023

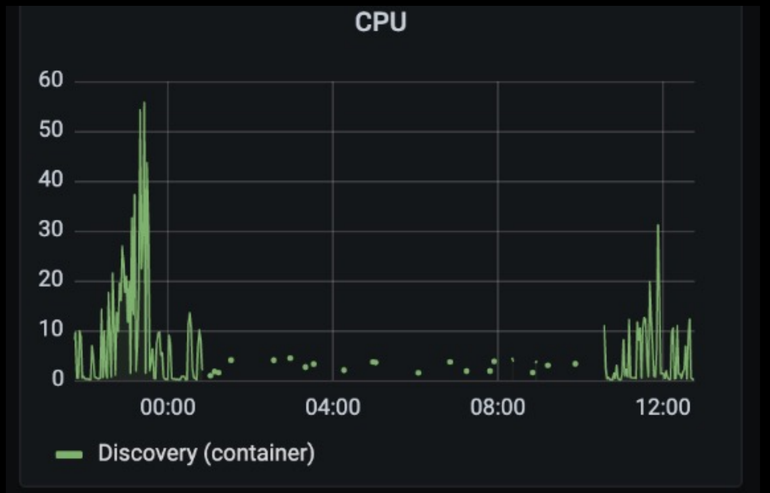


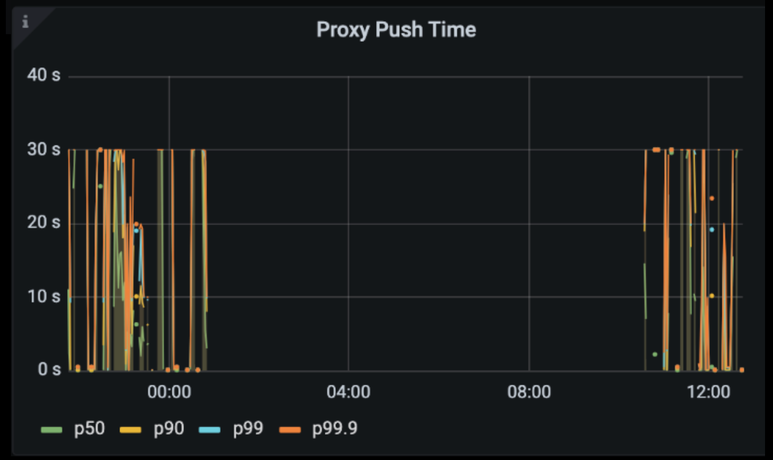
https://istio.io/latest/docs/ops/deployment/architecture/

2.2 문제: Istiod 부하 증가

NAVER
DEVIEW
2023

사용자 증가 → 서비스 배포 증가로 인해 Istiod 부하 증가

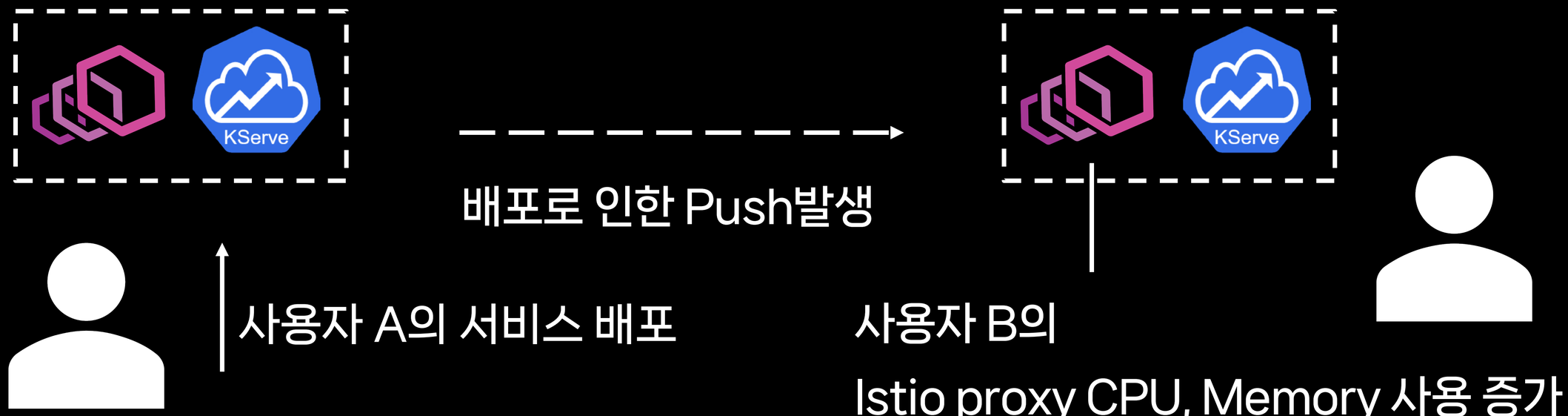




2.2 문제: Istio proxy의 리소스 사용량 증가

NAVER
DEVIEW
2023

다른 사용자의 서비스 배포가 모델 서빙 pod의 Istio proxy에 영향을 주는 문제가 있었습니다. 모델 서빙 이외의 용도로 Istio proxy가 리소스를 낭비하게 됩니다.



2.3 불필요한 Push 줄이기

NAVER
DEVIEW
2023

기본적으로 Istio proxy는 클러스터 안에 있는 다른 모든 서비스에 접근할 수 있도록 필요한 모든 설정들을 주입 (push) 받습니다. 불필요한 push는 클러스터의 리소스를 낭비하게 됩니다.

- 다른 Namespace에서의 변경 사항으로 인해 불필요한 push가 발생할 수 있습니다.
- Namespace 안에서도 관련 없는 서비스로 인해 불필요한 push가 발생할 수 있습니다.

2.3 불필요한 Push 줄이기

NAVER
DEVIEW
2023

Sidecar 리소스의 egress 설정을 통해 다른 namespace와 격리하고 불필요한 push를 줄일수 있습니다.

```yaml
apiVersion: networking.istio.io/v1beta1
kind: Sidecar
metadata:
  name: default
  namespace: istio-system
spec:
  egress:
  - hosts:
    - istio-system/*
    - kubeflow/*
    - ./* # Current namespace
```

host → namespace/dnsName 형식

Istio proxy가 배포된 namespace, kubeflow, istio-system namespace에 있는 리소스들이 변경될 때만 push를 받게 됩니다.

2.3 불필요한 Push 줄이기

NAVER
DEVIEW
2023

```
60  func checkProxyDependencies(proxy *model.Proxy, config model.ConfigKey) bool {
61          // Detailed config dependencies check.
62          switch proxy.Type {
63          case model.SidecarProxy:
64                  if proxy.SidecarScope.DependsOnConfig(config) {
65                          return true
66                  } else if proxy.PrevSidecarScope != nil && proxy.PrevSidecarScope.DependsOnConfig(config) {
67                          return true
68                  }
69          default:
70                  // TODO We'll add the check for other proxy types later.
71                  return true
72          }
73          return false
```

checkProxyDependencies – Proxy가 특정 리소스와 관련 있는지 체크하여 push를 여부를 결정하는 함수

https://github.com/istio/istio/blob/1.11.8/pilot/pkg/xds/proxy_dependencies.go#L60-L73

2.3 불필요한 Push 줄이기

NAVER DEVIEW 2023

```
60  func checkProxyDependencies(proxy *model.Proxy, config model.ConfigKey) bool {
61          // Detailed config dependencies check.
62          switch proxy.Type {
63          case model.SidecarProxy:
64                  if proxy.SidecarScope.DependsOnConfig(config) {
65                          return true
66                  } else if proxy.PrevSidecarScope != nil && proxy.PrevSidecarScope.DependsOnConfig(config) {
67                          return true
68                  }
69          default:
70                  // TODO We'll add the check for other proxy types later.
71                  return true
72          }
73          return false
```

Sidecar egress 설정에 따라 정해진다

Kubernetes Service 등의 리소스

checkProxyDependencies – Proxy가 특정 리소스와 관련 있는지 체크하여 push를 여부를 결정하는 함수

https://github.com/istio/istio/blob/1.11.8/pilot/pkg/xds/proxy_dependencies.go#L60-L73

2.3 불필요한 Push 줄이기

NAVER
DEVIEW
2023

Sidecar 리소스 설정 전후 비교 (100 push / s)

설정 전

- Memory footprint (proxy) : 300 MB
- P99 Push time (istiod): ~ 30s
- Avg. CPU usage (istiod): 15 ~ 20

설정 후

- Memory footprint (proxy) : 60 MB
- P99 Push time (istiod): ~ 5s
- Avg. CPU usage (istiod): ~ 5

2.3 불필요한 Push 줄이기

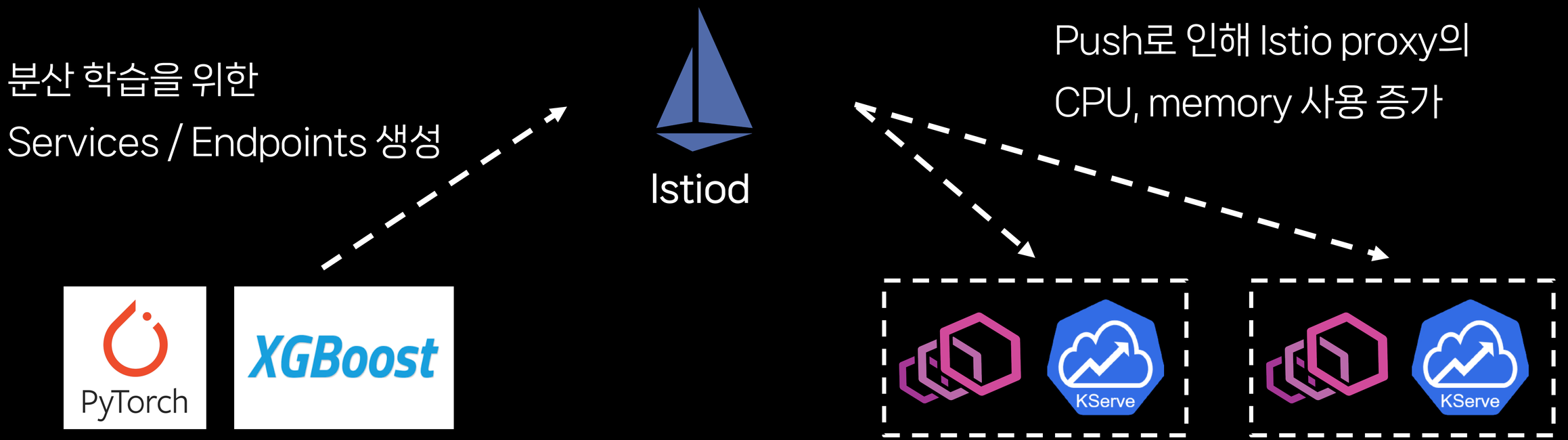
기본적으로 Istio proxy는 클러스터 안에 있는 다른 모든 proxy에 접근할 수 있도록 필요한 모든 설정들을 주입 (push) 받습니다. 불필요한 push는 클러스터의 리소스를 낭비하게 됩니다.

- 다른 Namespace에서의 변경 사항으로 인해 불필요한 push가 발생할 수 있습니다.
- Namespace 안에서도 관련 없는 서비스로 인해 불필요한 push가 발생할 수 있습니다.

2.3 불필요한 Push 줄이기

NAVER
DEVIEW
2023

분산 학습 용도로 생성된 서비스로 인해 불필요한 push가 생길 수 있습니다.



2.3 불필요한 Push 줄이기

NAVER
DEVIEW
2023

Sidecar 리소스로 더 세밀하게 proxy의 범위를 조절할 수 있지만, 사용자가 이를 하나하나 직접 설정하기는 어렵습니다.

```yaml
apiVersion: networking.istio.io/v1beta1
kind: Sidecar
metadata:
  name: default
  namespace: jwpark
spec:
  egress:
  - hosts:
     - ./post-processing-service
  workloadSelector:
    labels:
      app: predictor
```

Predictor service에 속한 Istio proxy는 같은 namespace 안의 post-processing-service 에만 영향 받습니다.

2.3 불필요한 Push 줄이기

NAVER DEVIEW 2023

분산 학습 서비스의 visibility를 "None" 으로 설정하여 불필요한 Push를 줄일 수 있습니다.

Istio mesh에 서비스가 노출되는 정도 (SidecarScope을 결정하는데 같이 사용됨)

```
25  type Instance string
26
27  const (
28          // Private implies namespace local config
29          Private Instance = "."
30          // Public implies config is visible to all
31          Public Instance = "*"
32          // None implies service is visible to no one. Used for services only
33          None Instance = "~"
34  )
```

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    networking.istio.io/exportTo: "~"
  name: private-service
spec:
  type: ClusterIP
```

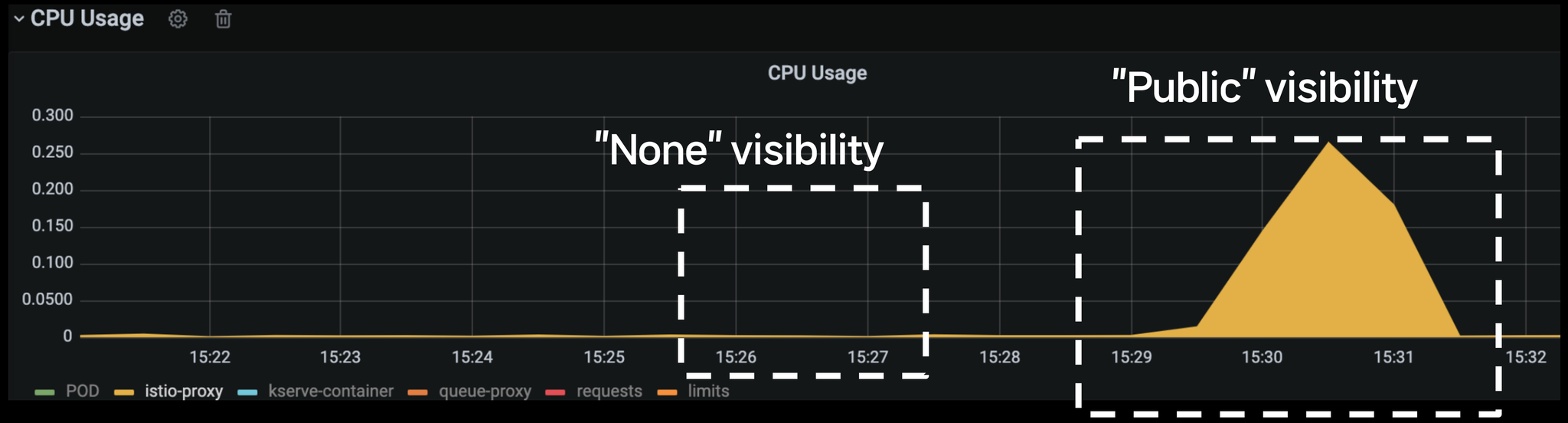
Istiod는 "private-service" 서비스에 대한 정보를 다른 proxy들에게 push하지 않습니다.

https://github.com/istio/istio/blob/1.11.8/pkg/config/visibility/visibility.go#L27-L34

2.3 불필요한 Push 줄이기

NAVER
DEVIEW
2023

"None" visibility의 효과 (Istio proxy)



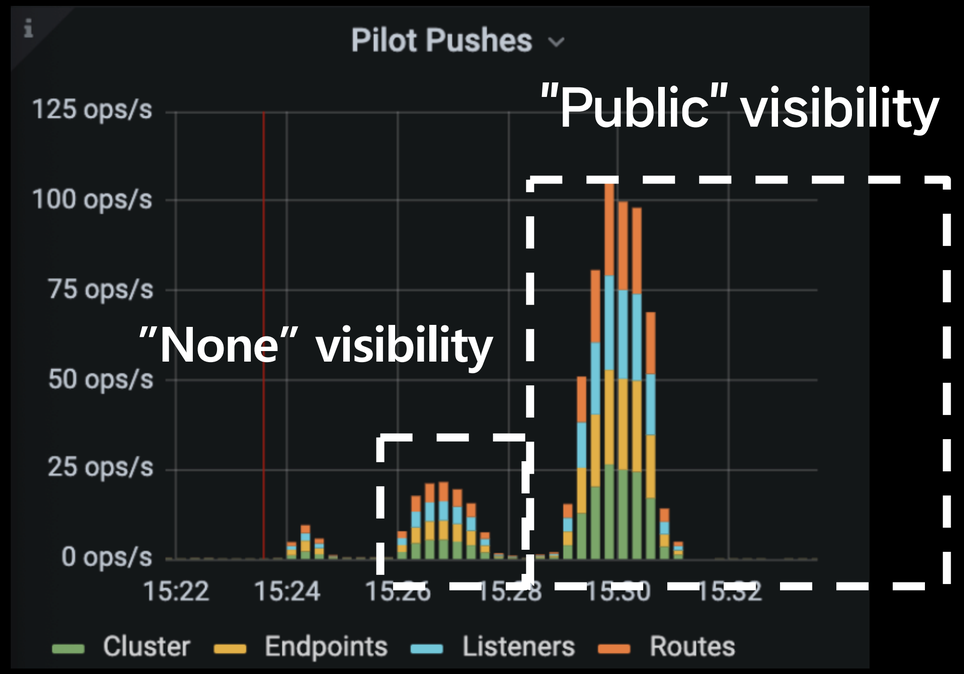
같은 namespace 안에서 visibility를 다르게 하여 수백 개의 서비스를 생성했을 때

Istio proxy의 CPU 사용량

2.3 불필요한 Push 줄이기

NAVER
DEVIEW
2023

"None" visibility의 효과 (Istiod)

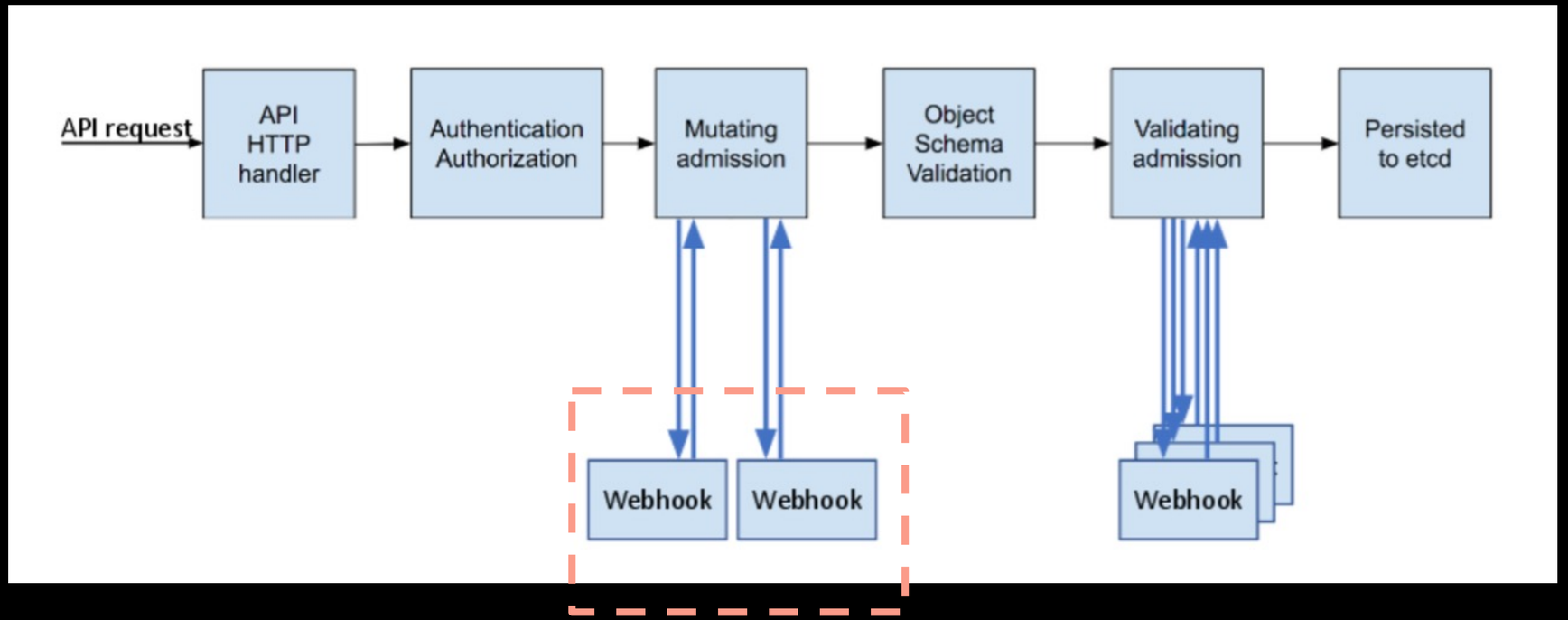


수백개의 서비스를 "None" / "Public" visibility 로 생성했을 때 Istiod push

2.3 불필요한 Push 줄이기

NAVER
DEVIEW
2023

사용자가 직접 서비스의 visibility를 신경쓰기는 어렵습니다. 서비스의 성격에 따라 visibility를 설정해주는 mutating admission webhook을 개발하였습니다.

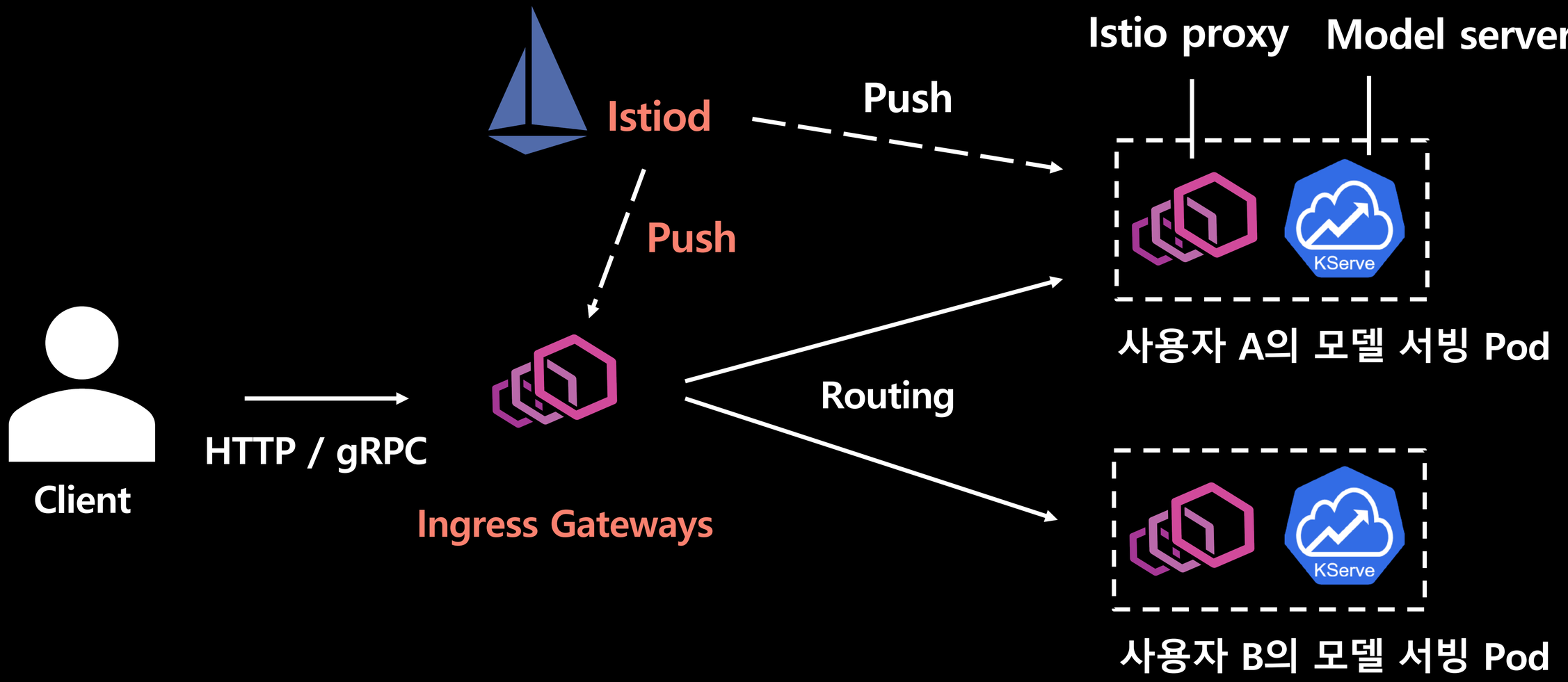


분산 학습 서비스의 경우 "None" visibility를 설정합니다.

https://kubernetes.io/blog/2019/03/21/a-guide-to-kubernetes-admission-controllers/

2.4 문제: Ingress Gateway Push로 인한 CPU 사용

NAVER DEVIEW 2023

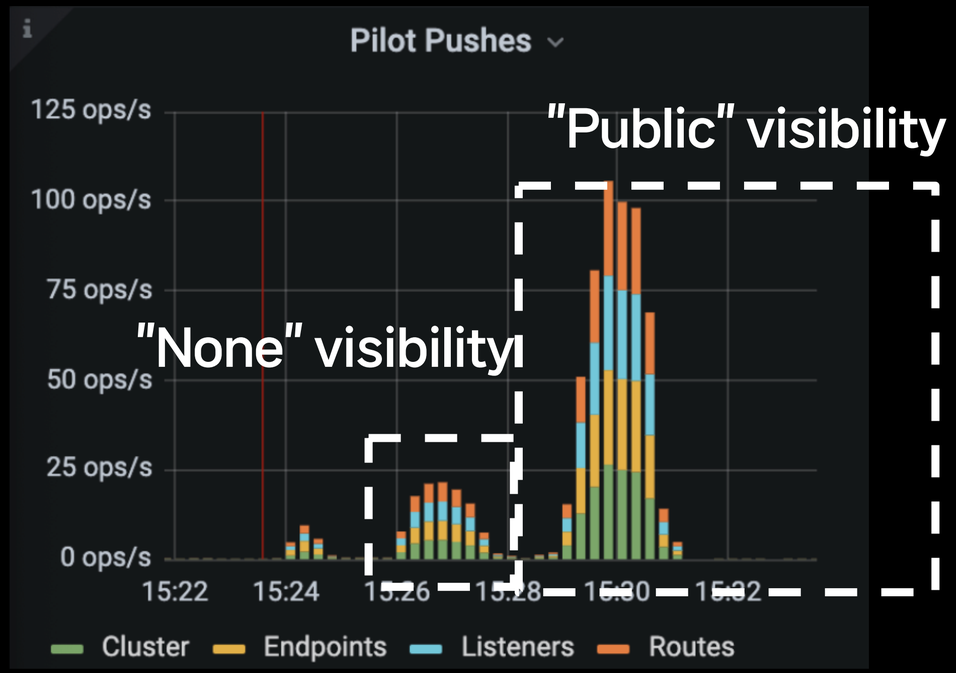


https://istio.io/latest/docs/ops/deployment/architecture/

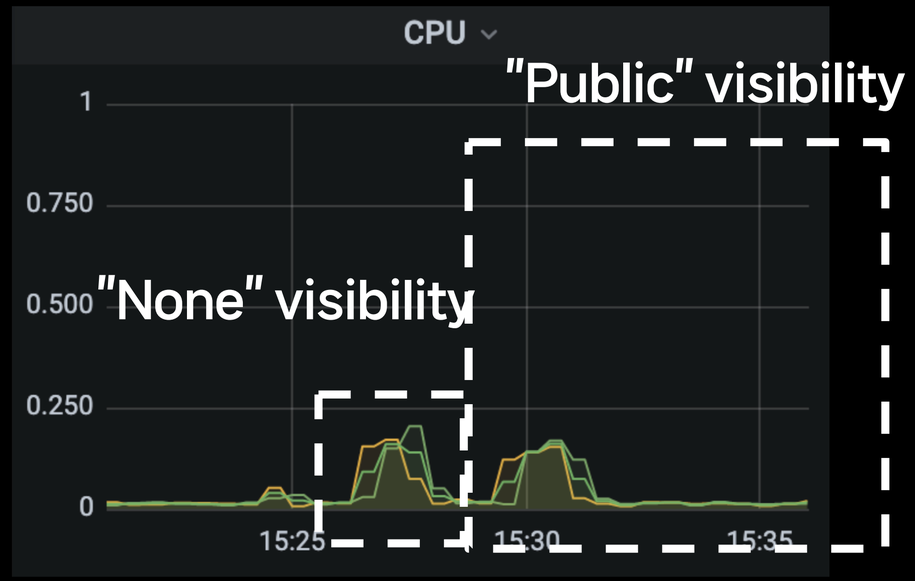
2.4 문제: Ingress Gateway Push로 인한 CPU 사용

NAVER DEVIEW 2023

"None" visibility로 서비스를 생성하더라도 Gateway에 무조건 push가 발생하는 문제가 있었습니다.



Istiod의 push

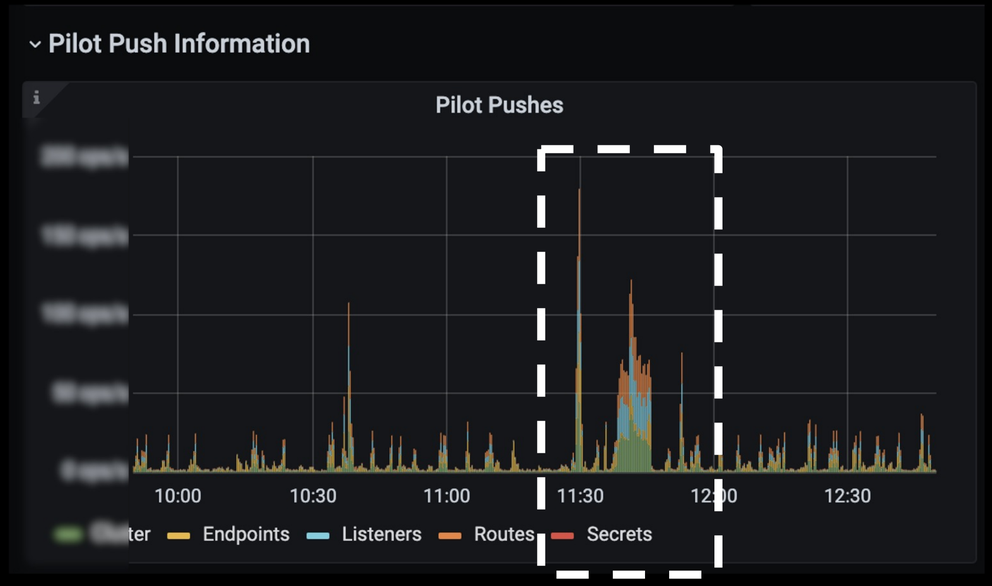


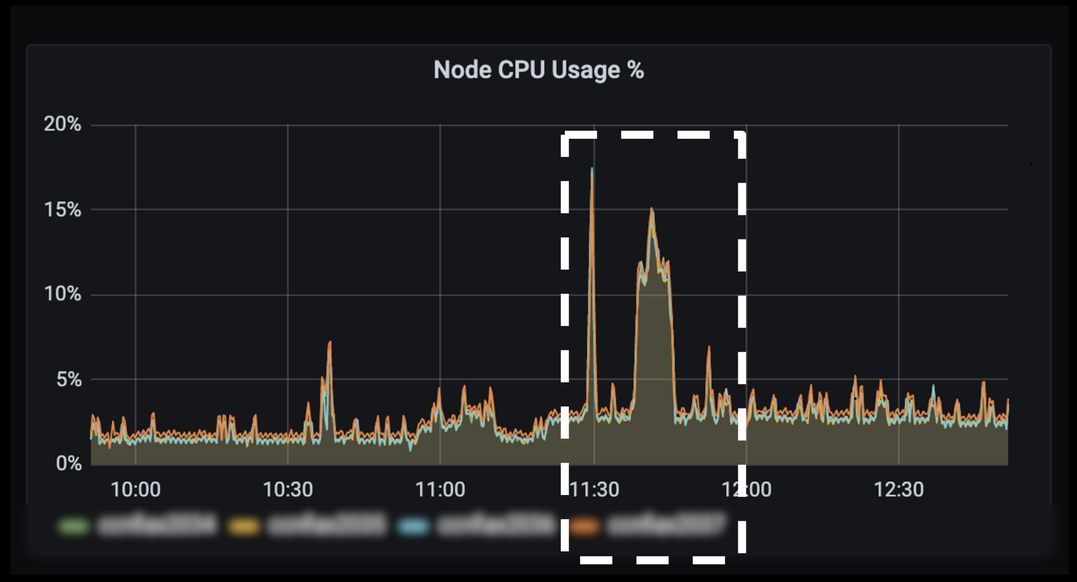
Push로 인해 증가한 Ingress gateway pod의 CPU 사용량

2.4 문제: Ingress Gateway Push로 인한 CPU 사용

NAVER DEVIEW 2023

Client 요청 처리와 관련 없이 push로 인해 많은 CPU를 사용하는 게 문제가 됩니다.





대규모 분산 학습 시 많은 수의 서비스 생성 → Pilot push 발생 → Ingress Gateway의 CPU 사용량 증가

2.4 Gateway Push 줄이기

NAVER DEVIEW 2023

```
60  func checkProxyDependencies(proxy *model.Proxy, config model.ConfigKey) bool {
61          // Detailed config dependencies check.
62          switch proxy.Type {
63          case model.SidecarProxy:
64                  if proxy.SidecarScope.DependsOnConfig(config) {
65                          return true
66                  } else if proxy.PrevSidecarScope != nil && proxy.PrevSidecarScope.DependsOnConfig(config) {
67                          return true
68                  }
69          default:
70                  // TODO We'll add the check for other proxy types later.
71                  return true
72          }
73          return false
74  }
```

checkProxyDependencies – Proxy가 특정 리소스와 관련 있는지 체크하여 push를 여부를 결정하는 함수

Gateway type proxy의 경우 push가 항상 일어나는 것을 확인할 수 있습니다.

https://github.com/istio/istio/blob/1.11.8/pilot/pkg/xds/proxy_dependencies.go#L60-L74

2.4 Gateway Push 줄이기

NAVER DEVIEW 2023

```
68          case model.Router:
69                  if config.Kind == kind.ServiceEntry {
70                          // If config is ServiceEntry, name of the config is service's FQDN
71                          svc, exist := push.ServiceIndex.HostnameAndNamespace[host.Name(config.Name)][config.Namespace]
72                          if exist {
73                                  if !push.IsServiceVisible(svc, proxy.Metadata.Namespace) {
74                                          return false
75                                  }
76                          }
77                  }
78                  return true
```

Improved the number of pushes to gateway proxies by not pushing when services are not visible from the gateway. (Issue #39110)

- Istio 1.15.0 에 기여

https://github.com/istio/istio/issues/39110

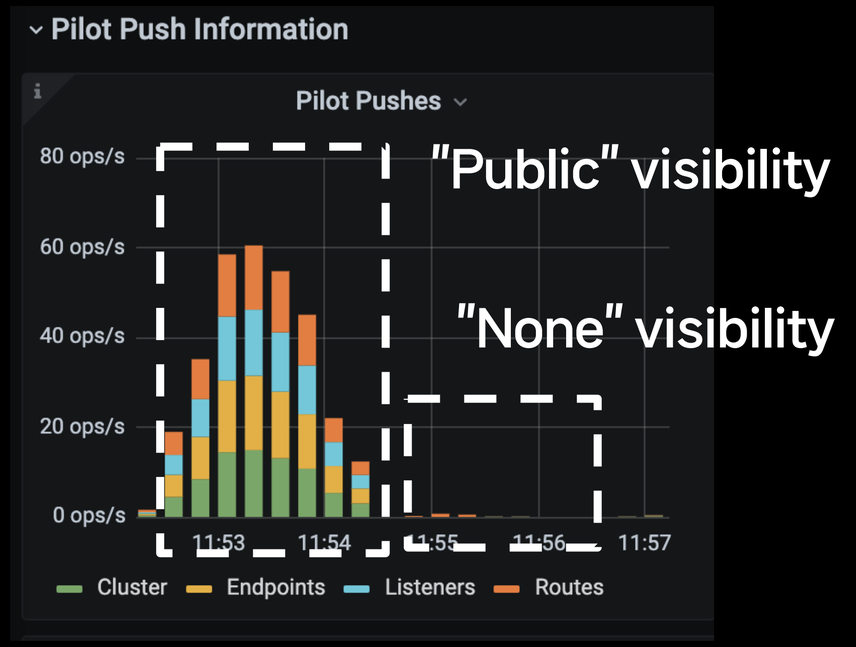
https://github.com/istio/istio/pull/39168

https://github.com/istio/istio/pull/42422

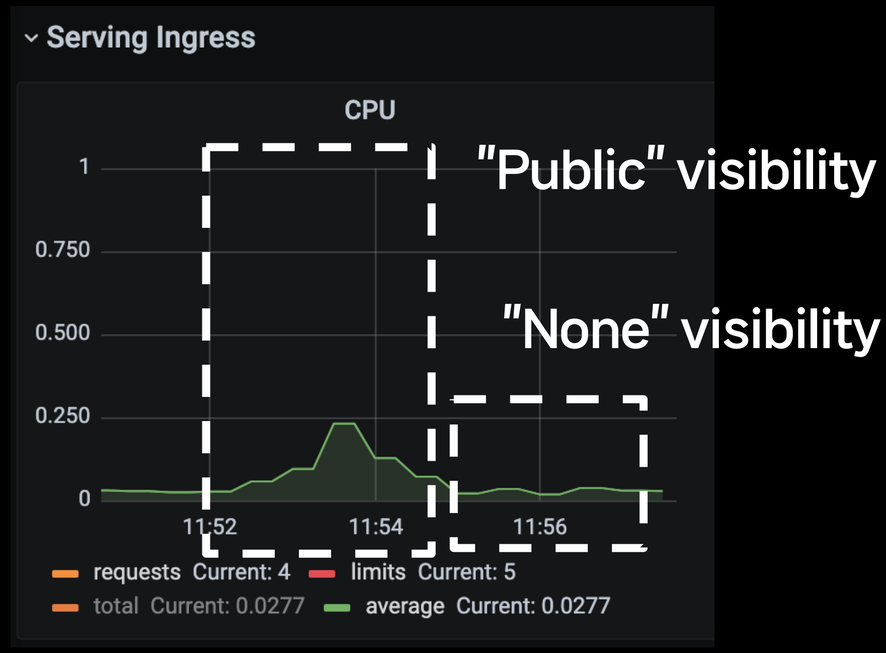
2.4 Gateway Push 줄이기

NAVER DEVIEW 2023

개선 이후 불필요한 gateway push가 줄어들었음을 확인했습니다.



Istiod의 push



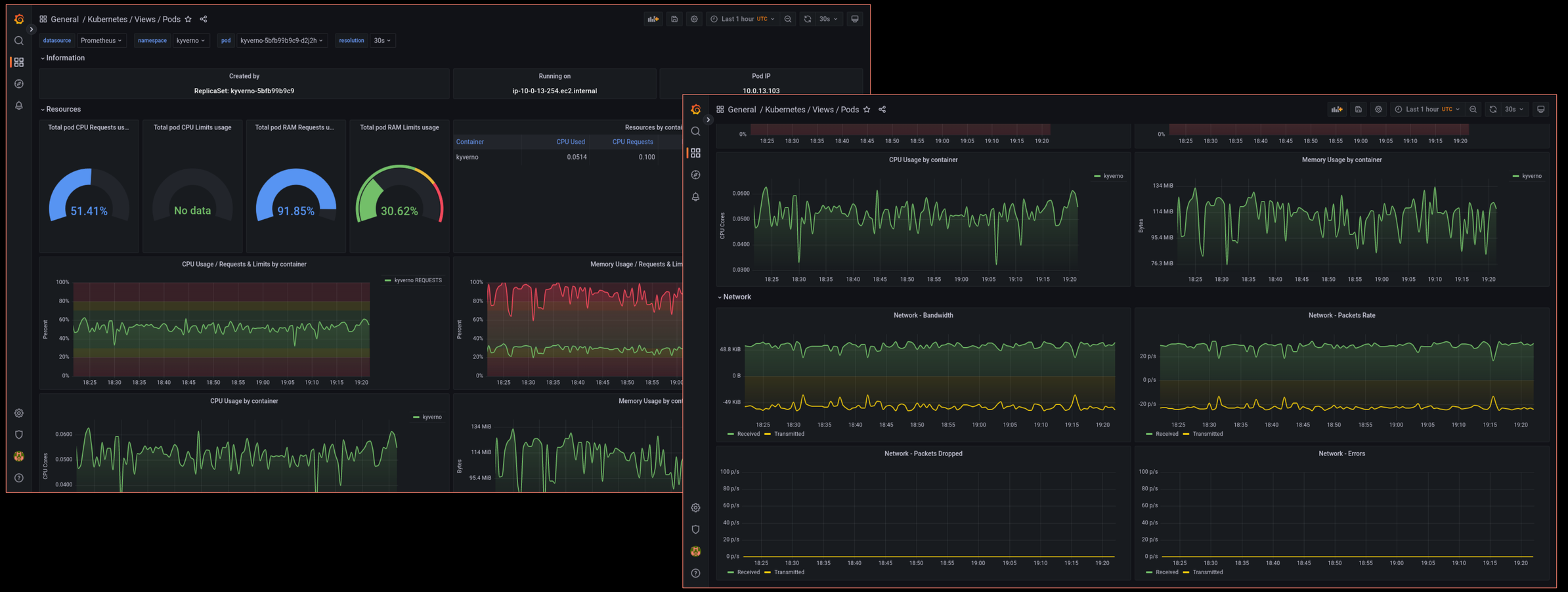
Ingress gateway pod의 CPU 사용량

3. 모델 서빙 모니터링 안정적으로 제공하기: 모니터링 시스템 개선

3.1 Prometheus 기반의 모니터링 시스템

NAVER
DEVIEW
2023

관리자들에게 꼭 필요한 대시보드

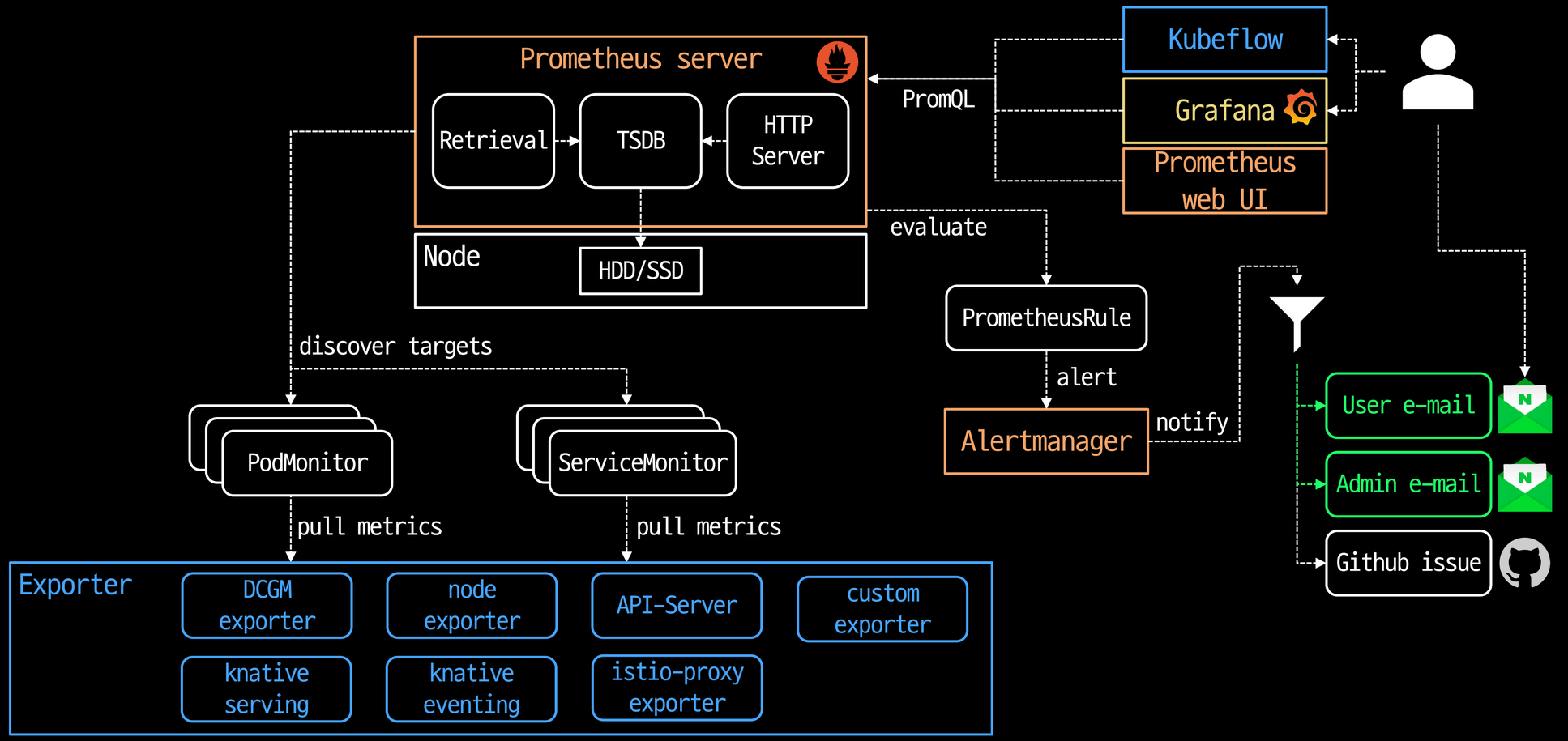


https://grafana.com/grafana/dashboards/15760-kubernetes-views-pods/

3.1 Prometheus 기반의 모니터링 시스템

NAVER
DEVIEW
2023

모니터링 시스템은 운영에 있어 중요한 부분을 담당합니다.



3.1 Prometheus 기반의 모니터링 시스템

NAVER
DEVIEW
2023

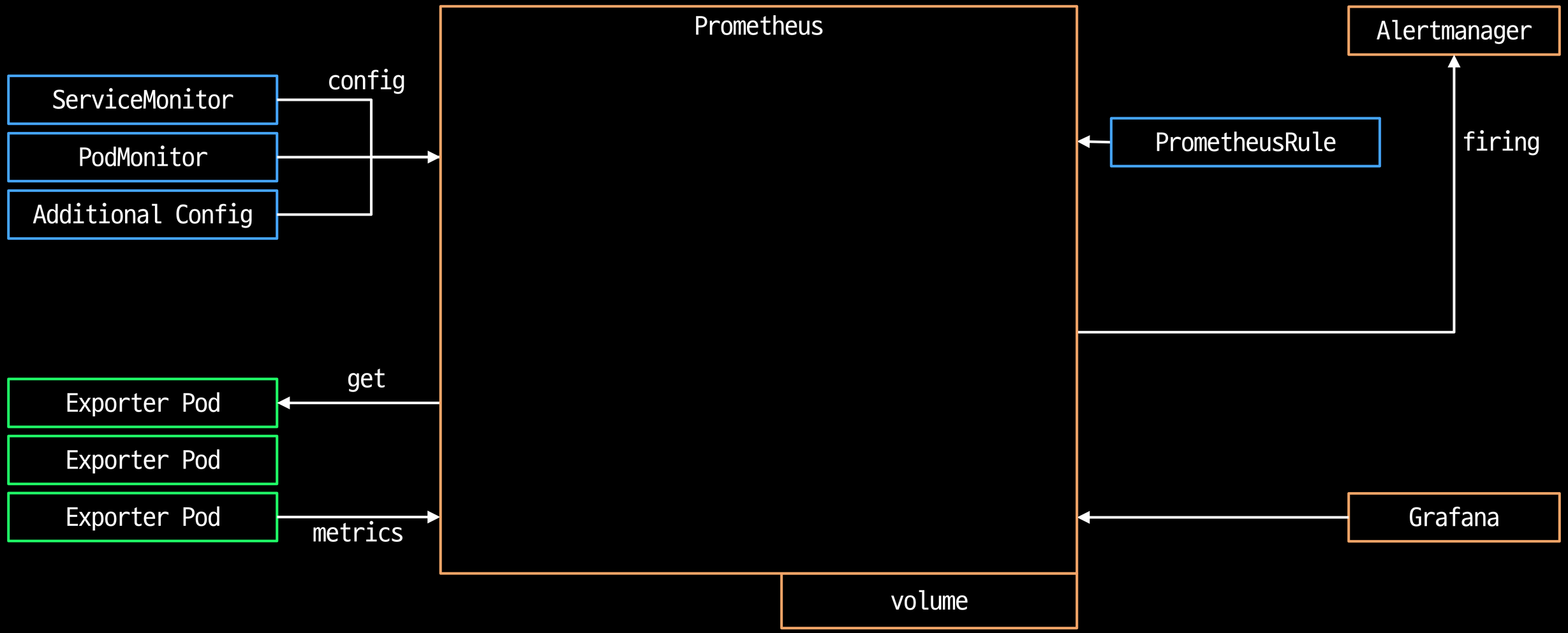
Prometheus Operator

- Kubernetes Custom Resource 배포를 이용
- 대상에 맞는 설정을 명시적으로 선언하여 간단하고, 자동으로 설정되도록 지원

```yaml
apiVersion: monitoring.coreos.com/v1
kind: Prometheus
metadata:
  name: sample
  namespace: monitoring
spec:
  retention: 30d
resources:
...
```

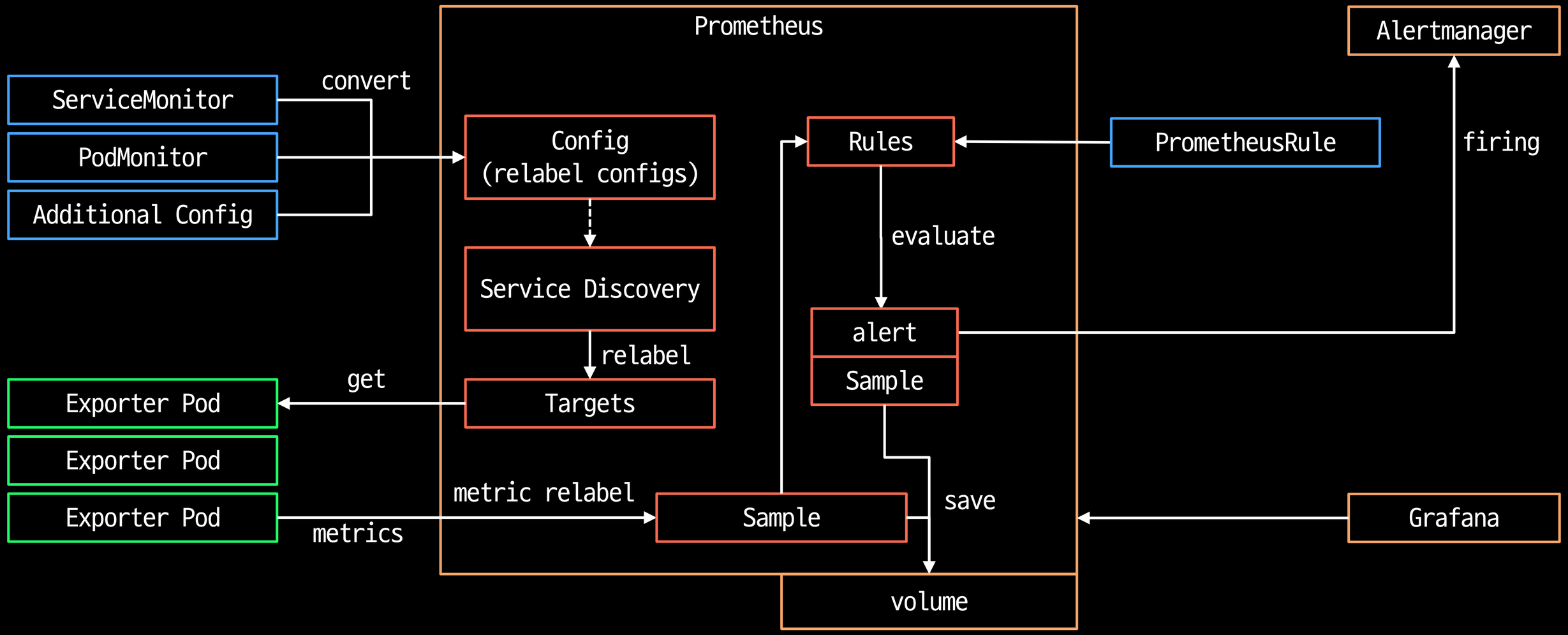
3.1 Prometheus 기반의 모니터링 시스템

NAVER
DEVIEW
2023



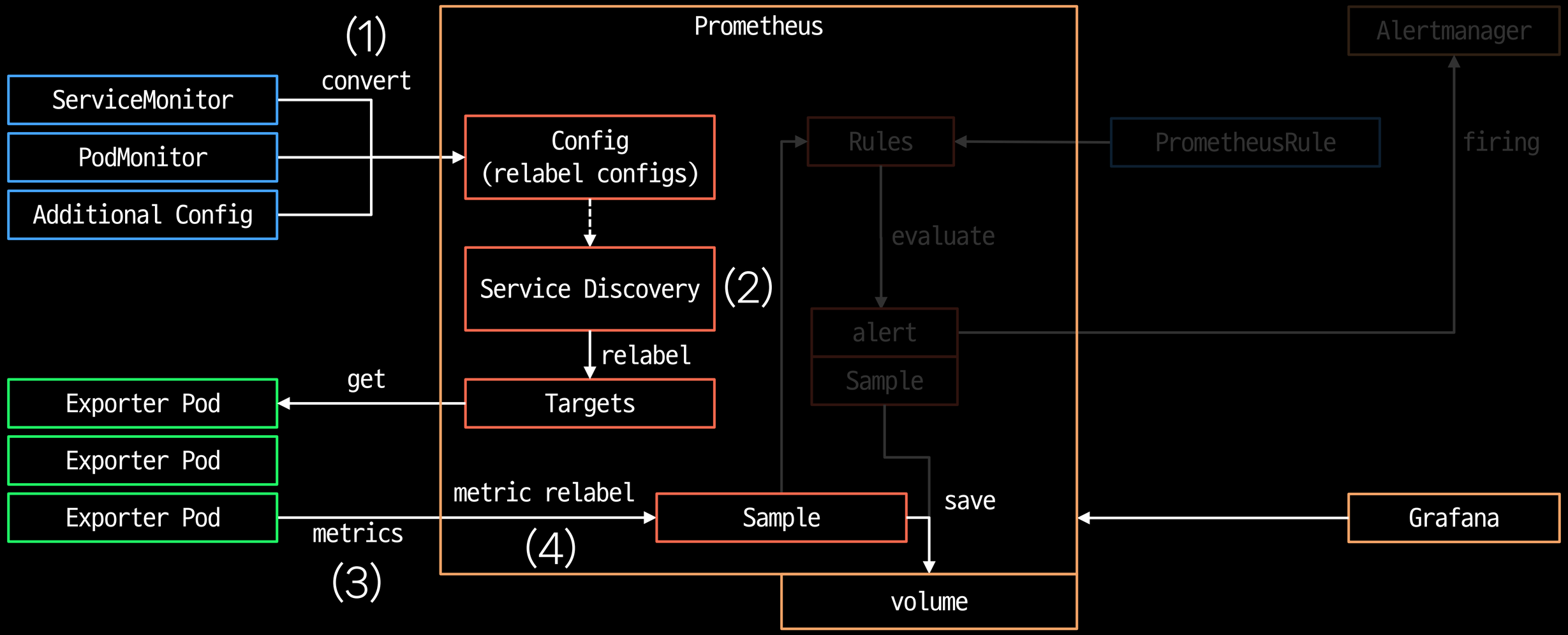
3.1 Prometheus 기반의 모니터링 시스템

NAVER
DEVIEW
2023



3.1 Prometheus 기반의 모니터링 시스템

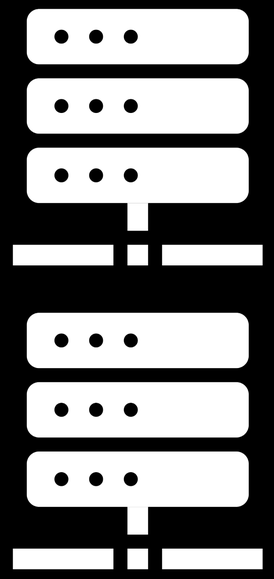
NAVER
DEVIEW
2023



3.1 Prometheus 기반의 모니터링 시스템

NAVER
DEVIEW
2023

AiSuite: AI 플랫폼 (다수의 GPU)



3.1 Prometheus 기반의 모니터링 시스템

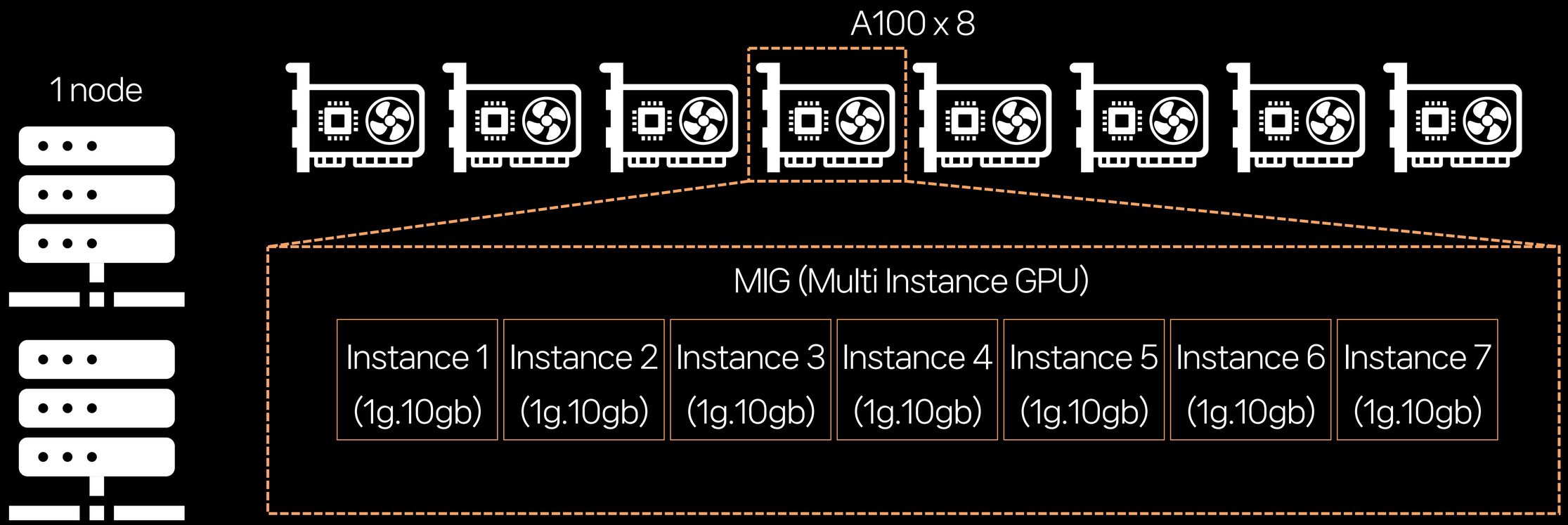
NAVER
DEVIEW
2023

AiSuite: AI 플랫폼 (다수의 GPU)

3.1 Prometheus 기반의 모니터링 시스템

NAVER
DEVIEW
2023

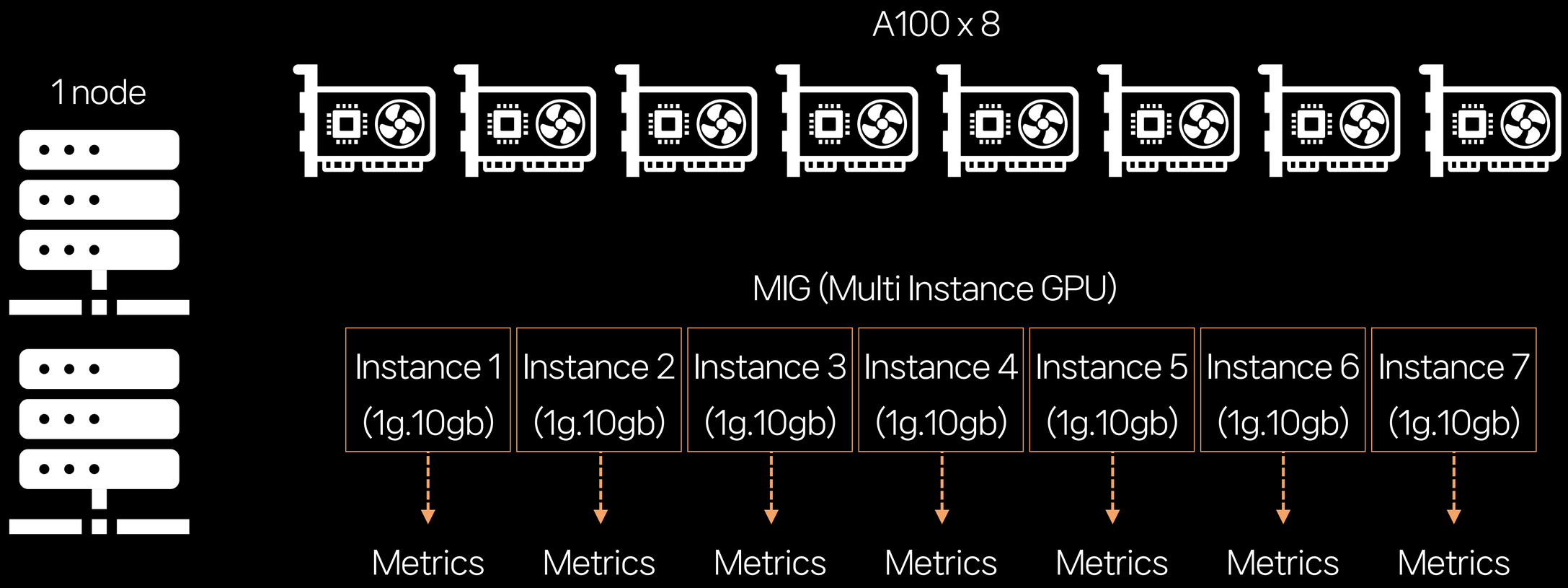
AiSuite: AI 플랫폼 (다수의 GPU)



3.1 Prometheus 기반의 모니터링 시스템

NAVER
DEVIEW
2023

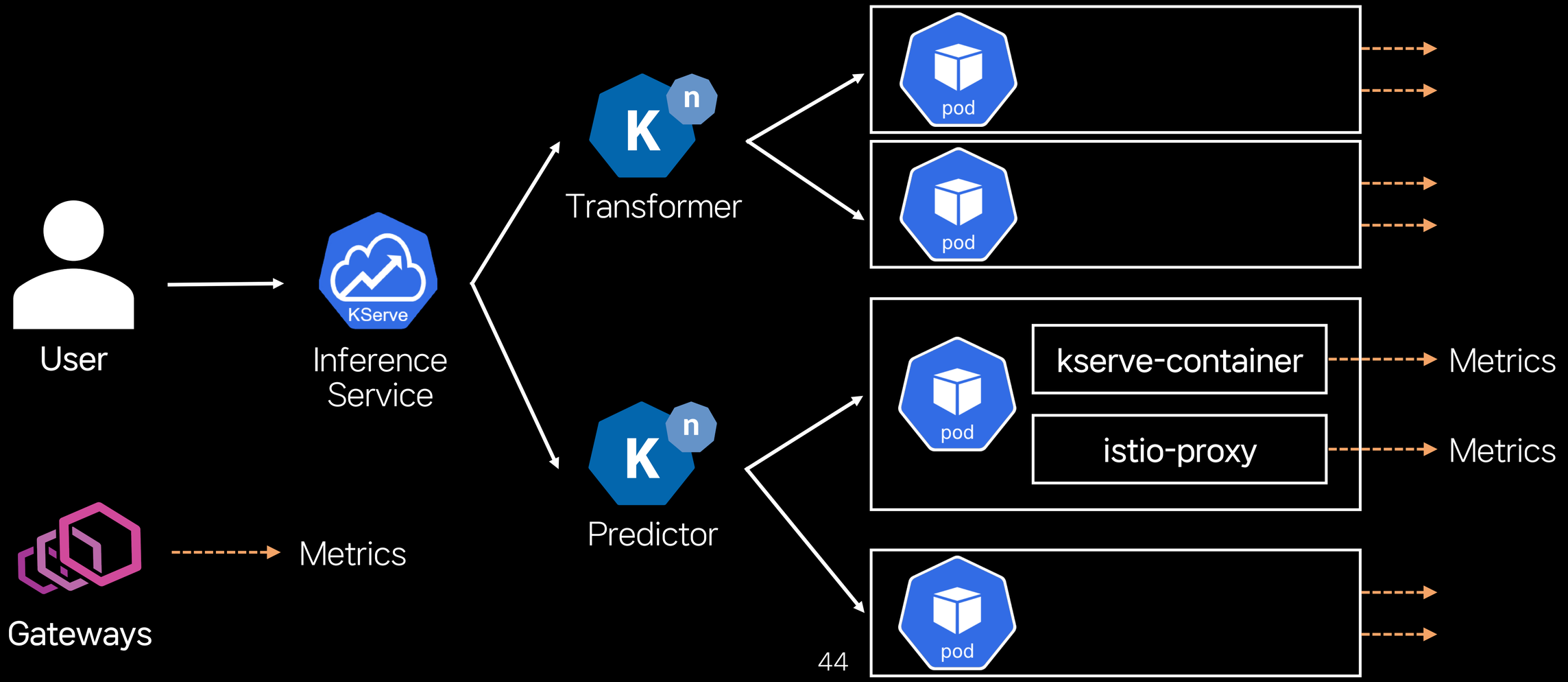
AiSuite: AI 플랫폼 (다수의 GPU)



3.1 Prometheus 기반의 모니터링 시스템

NAVER
DEVIEW
2023

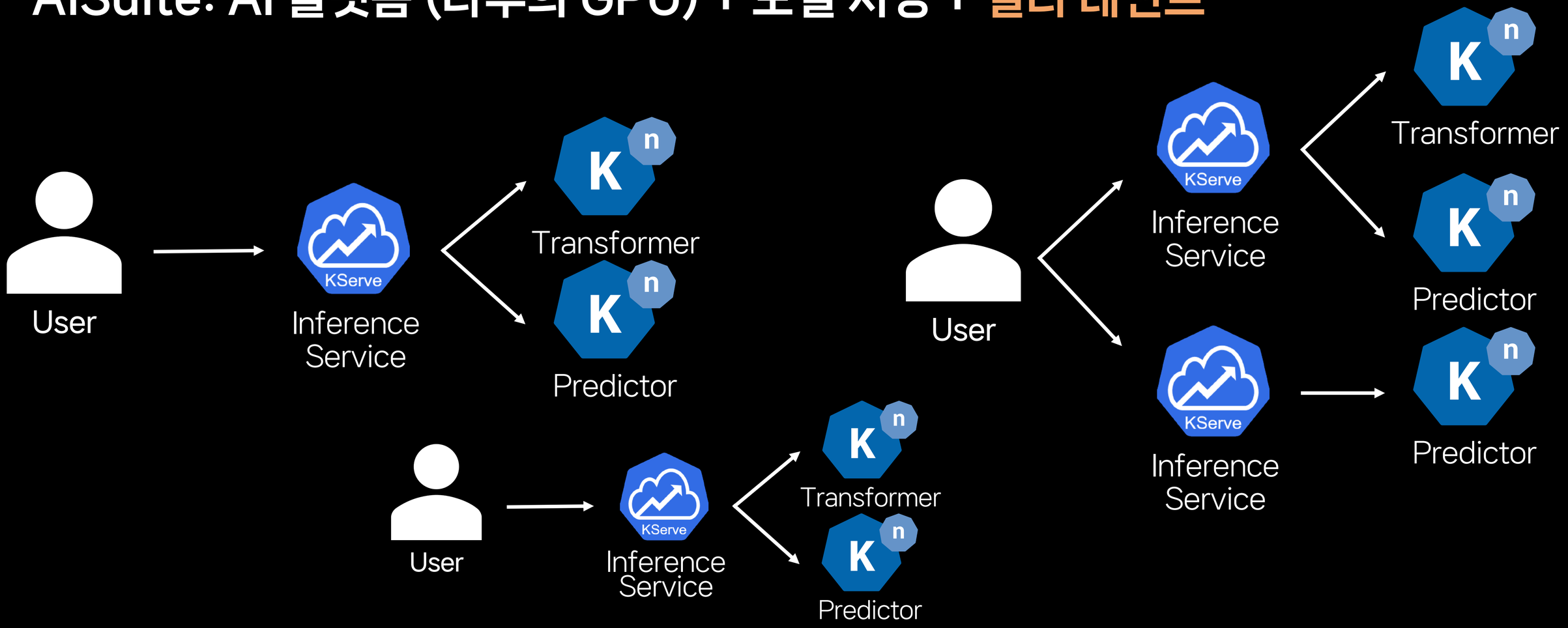
AiSuite: AI 플랫폼 (다수의 GPU) + 모델 서빙



3.1 Prometheus 기반의 모니터링 시스템

NAVER
DEVIEW
2023

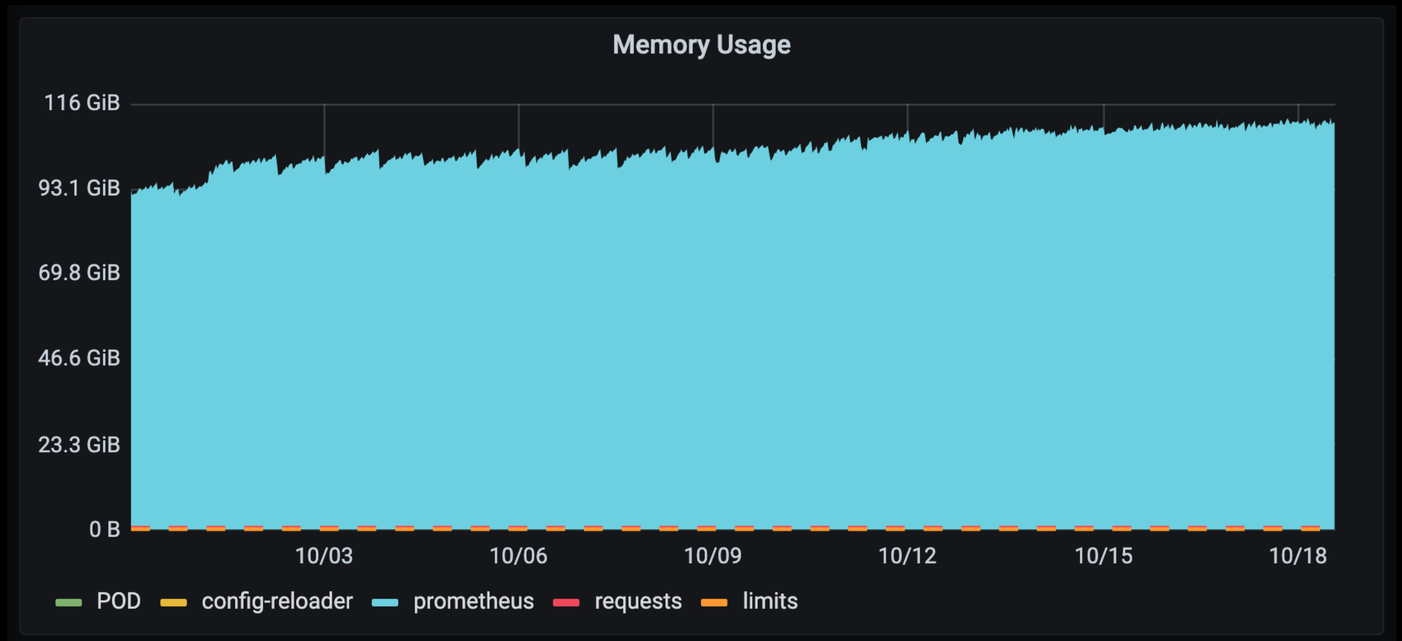
AiSuite: AI 플랫폼 (다수의 GPU) + 모델 서빙 + 멀티 테넌트



3.2 문제: 높은 메모리 사용량

NAVER
DEVIEW
2023

Prometheus는 기본적으로 3시간 가량의 지표를 Memory상에도 저장합니다.
수많은 Metrics는 높은 메모리 사용량을 유발합니다.



Head stats	
Number of Series	1천여만개 가량
Number of Chunks	2천여만개 가량
Number of Label Pairs	1십여만개 가량

3.2 문제: 높은 메모리 사용량

NAVER
DEVIEW
2023

Prometheus는 기본적으로 3시간 가량의 지표를 Memory상에도 저장합니다.
수많은 Metrics는 높은 메모리 사용량을 유발합니다.
Kubernetes 특성 상 높은 churn rate가 발생할 수 있습니다.

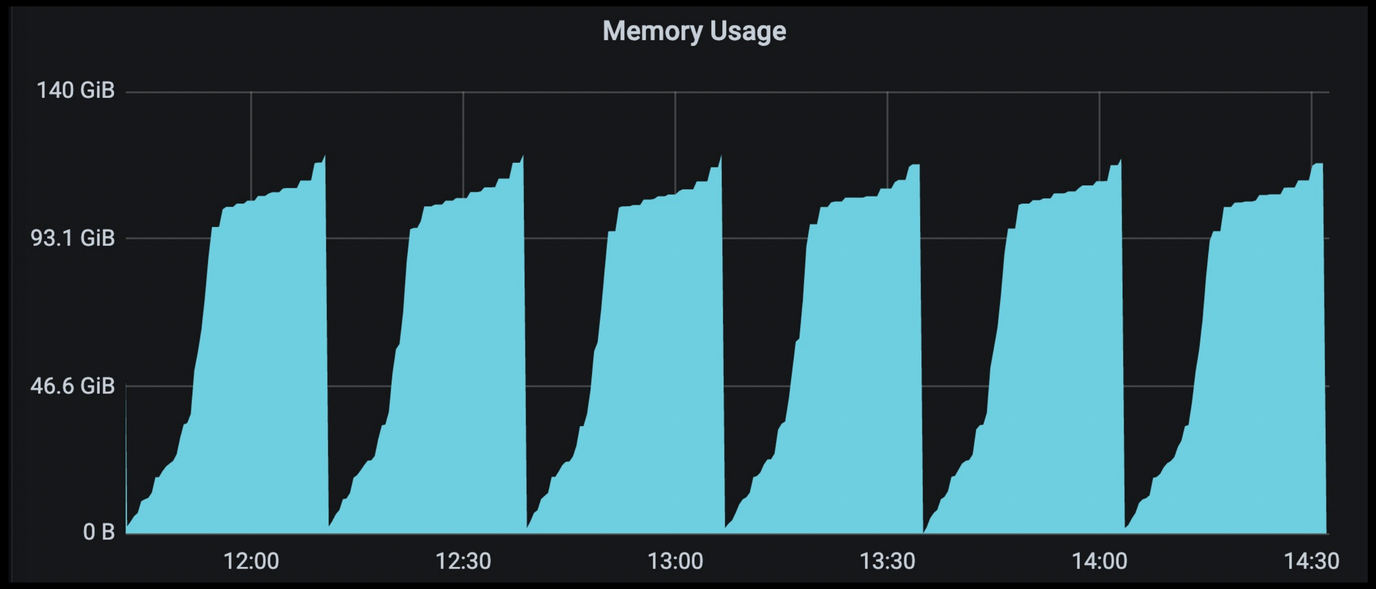
- 높은 churn rate → 높은 cardinality → OOM 유발


```
some_metric_name_1 {svc="service-1", pod="some-pod-old", ip="1.1.1.1"}
some_metric_name_1 {svc="service-1", pod="some-pod-new", ip="2.2.2.2"}
```

3.2 문제: 높은 메모리 사용량

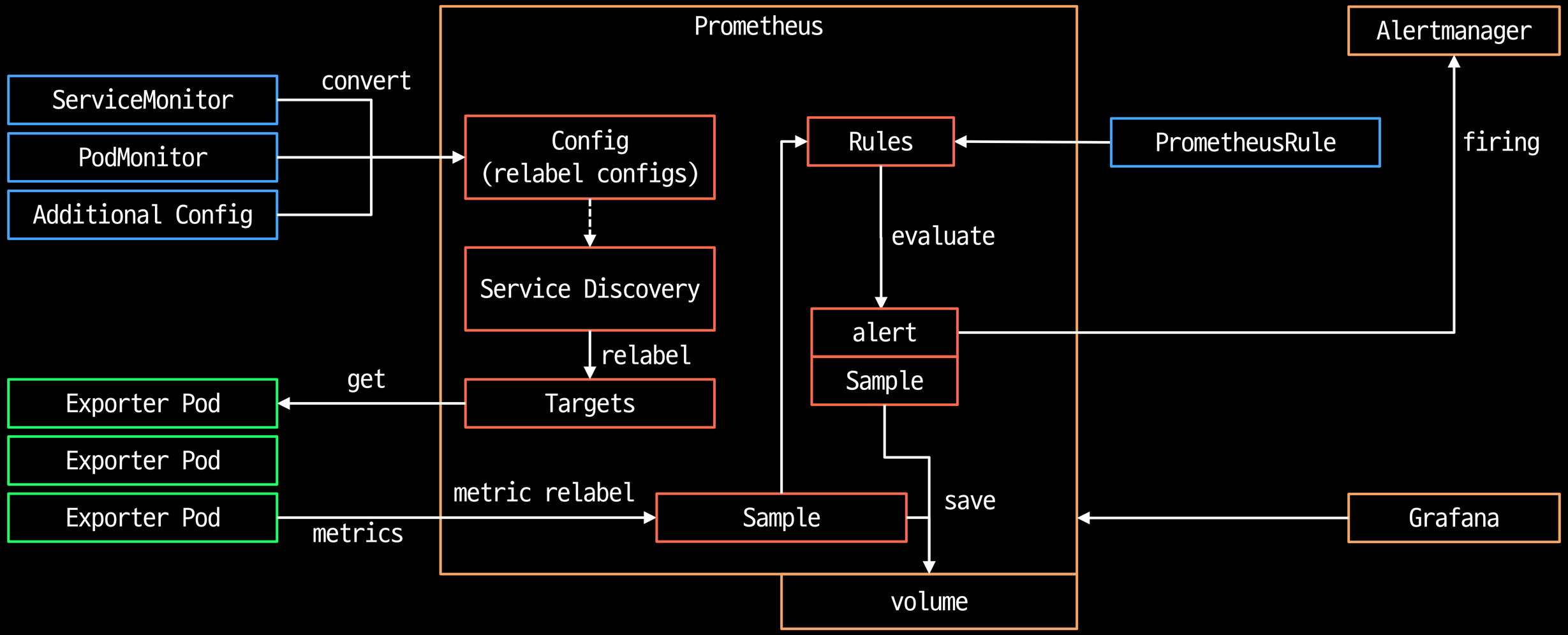
NAVER
DEVIEW
2023

Prometheus는 기본적으로 3시간 가량의 지표를 Memory상에도 저장합니다.
수많은 Metrics는 높은 메모리 사용량을 유발합니다.
Kubernetes 특성 상 높은 churn rate가 발생할 수 있습니다.
Prometheus에 OOM이 발생한다면 모니터링 시스템 전체에 문제가 발생합니다.



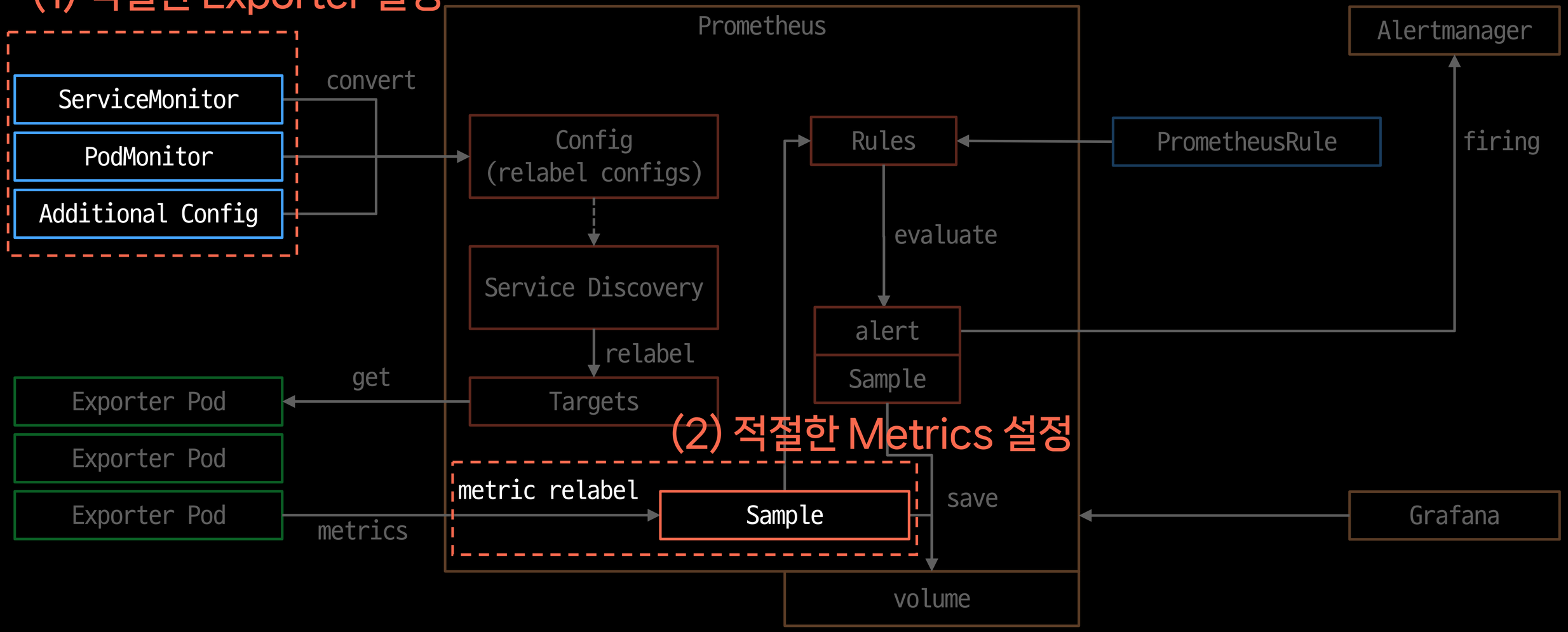
3.2 Metrics 줄이기

NAVER
DEVIEW
2023



3.2 Metrics 줄이기

NAVER
DEVIEW
2023



3.2 Metrics 줄이기

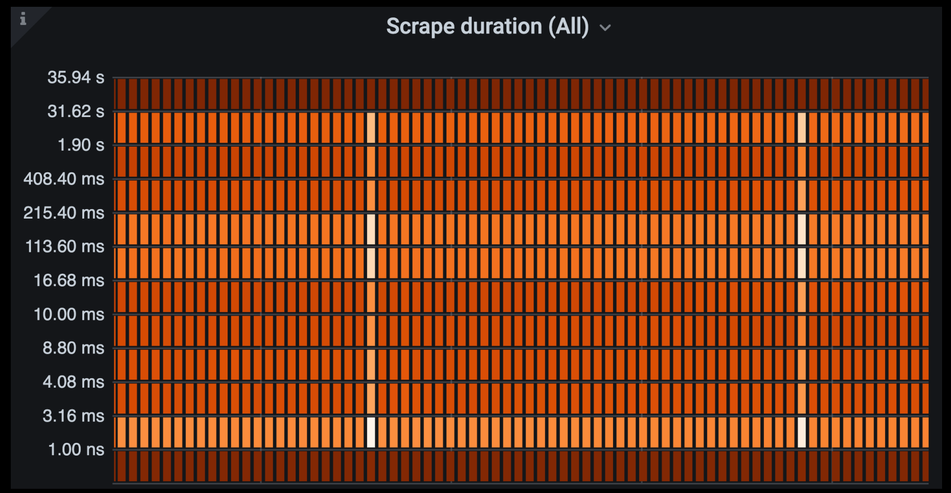
NAVER
DEVIEW
2023

Exporter를 통해 보여지는 지표들은 "전부"를 수집합니다.

- 향후 Metrics relabel 작업을 진행하더라도 "수집"은 이루어집니다.

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://1.2.3.4:15090/stats/prometheus	UP	container="istio-proxy" endpoint="http-envoy-prom" instance="1.2.3.4:15090" job="serving-gateway" namespace="istio-system" pod="some-pod"	1m 23s ago	61953ms	

- 높은 Duration은 time-out을 유발하고 ,
  수집 component에 부하를 가져옵니다.



3.2 Metrics 줄이기

NAVER
DEVIEW
2023

Exporter를 통해 보여지는 지표들은 "전부"를 수집합니다.

○ 설정을 통해 적절한 지표들만 내보내도록 조정이 필요합니다.

Enabled by default

Name	Description	OS
arp	Exposes ARP statistics from `/proc/net/arp`.	Linux
bcache	Exposes bcache statistics from `/sys/fs/bcache/`.	Linux
bonding	Exposes the number of configured and active slaves of Linux bonding interfaces.	Linux
btrfs	Exposes btrfs statistics	Linux
	Exposes system boot time derived from the	Darwin, Dragonfly, FreeBSD

eg) node-exporter --no-collector.<name>

3.2 Metrics 줄이기

NAVER
DEVIEW
2023

상당한 양의 지표는 부담을 줄 수 있습니다.

- 상황에 따라 특정 label을 삭제하여 cardinality를 줄일 수 있습니다.
- 상황에 따라 특정 지표를 수집하지 않게 하여 metrics 수를 줄일 수 있습니다.

```
some_metric_name_1 {pod="some-pod-1", ip="1.1.1.1", id="some-A"}
some_metric_name_1 {pod="some-pod-2", ip="1.1.1.2", id="some-A"}
some_metric_name_1 {pod="some-pod-3", ip="1.1.1.3", id="some-B"}


some_metric_name_2 {node="some-node-1", os="A"}
some_metric_name_2 {node="some-node-2", os="A"}
some_metric_name_2 {node="some-node-3", os="B"}
```

3.2 Metrics 줄이기

NAVER
DEVIEW
2023

상당한 양의 지표는 부담을 줄 수 있습니다.

- 상황에 따라 특정 label을 삭제하여 cardinality를 줄일 수 있습니다.
- 상황에 따라 특정 지표를 수집하지 않게 하여 metrics 수를 줄일 수 있습니다.

```
some_metric_name_1 {pod="some-pod-1", ip="1.1.1.1", id="some-A"}
some_metric_name_1 {pod="some-pod-2", ip="1.1.1.2", id="some-A"}
some_metric_name_1 {pod="some-pod-3", ip="1.1.1.3", id="some-B"}
```

```
some_metric_name_2 {node="some-node-1", os="A"}
some_metric_name_2 {node="some-node-2", os="A"}
some_metric_name_2 {node="some-node-3", os="B"}
```

3.2 Metrics 줄이기

NAVER
DEVIEW
2023

상당한 양의 지표는 부담을 줄 수 있습니다.

- 상황에 따라 특정 label을 삭제하여 cardinality를 줄일 수 있습니다.
- 상황에 따라 특정 지표를 수집하지 않게 하여 metrics 수를 줄일 수 있습니다.

```
some_metric_name_1 {pod="some-pod-1", ip="1.1.1.1", id="some-A"}
some_metric_name_1 {pod="some-pod-2", ip="1.1.1.2", id="some-A"}
some_metric_name_1 {pod="some-pod-3", ip="1.1.1.3", id="some-B"}
```

```
some_metric_name_2 {node="some-node-1", os="A"}
some_metric_name_2 {node="some-node-2", os="A"}
some_metric_name_2 {node="some-node-3", os="B"}
```

3.2 Metrics 줄이기

NAVER
DEVIEW
2023

상당한 양의 지표는 부담을 줄 수 있습니다.

- 상황에 따라 특정 label을 삭제하여 cardinality를 줄일 수 있습니다.
- 상황에 따라 특정 지표를 수집하지 않게 하여 metrics 수를 줄일 수 있습니다.

```
some_metric_name_1 {pod="some-pod-1", ip="1.1.1.1", id="some-A"}
some_metric_name_1 {pod="some-pod-2", ip="1.1.1.2", id="some-A"}
some_metric_name_1 {pod="some-pod-3", ip="1.1.1.3", id="some-B"}
```

```
some_metric_name_2 {node="some-node-1", os="A"}
some_metric_name_2 {node="some-node-2", os="A"}
some_metric_name_2 {node="some-node-3", os="B"}
```

3.2 Metrics 줄이기

NAVER
DEVIEW
2023

상당한 양의 지표는 부담을 줄 수 있습니다.

- **상황에 따라 특정 label을 삭제하여 cardinality를 줄일 수 있습니다.**

```
istio_requests_total{
  source_cluster="Kubernetes",
  destination_cluster="unknown",
  destination_canonical_revision="sample-predictor-default-00001",
  destination_canonical_service="sample-predictor-default",
  source_canonical_revision="sample-transformer-default-00001",
  source_canonical_service="sample-transformer-default",
  reporter="destination",
  response_code="200",
  response_flags="-",
  ...
  + 22 labels
}
```

불필요해 보이는 label

3.2 Metrics 줄이기

NAVER
DEVIEW
2023

상당한 양의 지표는 부담을 줄 수 있습니다.

- **상황에 따라 특정 label을 삭제하여 cardinality를 줄일 수 있습니다.**

```
istio_requests_total{
  source_cluster="Kubernetes",
  destination_cluster="unknown",
  destination_canonical_revision="sample-predictor-default-00001",
  destination_canonical_service="sample-predictor-default",
  source_canonical_revision="sample-transformer-default-00001",
  source_canonical_service="sample-transformer-default",
  reporter="destination",
  response_code="200",
  response_flags="-",
  ...
  + 22 labels
}
```

하지만 위의 Label이 없을 시,
Kiali와 같은 시스템을 사용중이라면 오류가 발생할 수 있어 주의가 필요합니다.

3.2 Metrics 줄이기

NAVER
DEVIEW
2023

상당한 양의 지표는 부담을 줄 수 있습니다.

- **상황에 따라 특정 지표를 수집하지 않게 하여 metrics 수를 줄일 수 있습니다.**


```
istio_requests_total{
  Reporter: source
  Response flag: -
  Status code: 200
}
```

```
istio_requests_total{
  Reporter: destination
  Response flag: -
  Status code: 200
}
```

3.2 Metrics 줄이기

NAVER
DEVIEW
2023

상당한 양의 지표는 부담을 줄 수 있습니다.

- **상황에 따라 특정 지표를 수집하지 않게 하여 metrics 수를 줄일 수 있습니다.**

중복된 내용을 갖고있어 한쪽은 불필요해 보입니다.

3.2 Metrics 줄이기

NAVER
DEVIEW
2023

상당한 양의 지표는 부담을 줄 수 있습니다.

- **상황에 따라 특정 지표를 수집하지 않게 하여 metrics 수를 줄일 수 있습니다.**


```
istio_requests_total{
  Reporter: source
  Response flag: UT
  Status code: 504
}
```

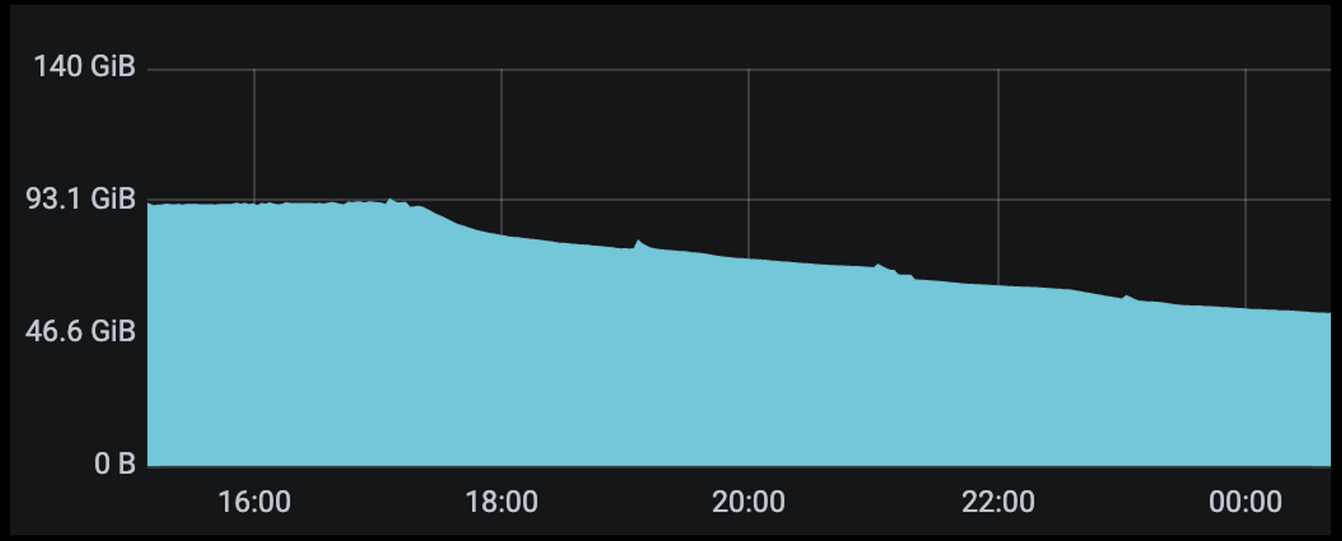
```
istio_requests_total{
  Reporter: destination
  Response flag: DC
  Status code: 0
}
```

하지만 문제 발생 시, 명확한 원인을 찾기 힘들 수 있습니다.

3.2 Metrics 줄이기

NAVER
DEVIEW
2023

적절한 label 제거, metrics 제거를 통해 메모리 사용량을 줄일 수 있습니다.



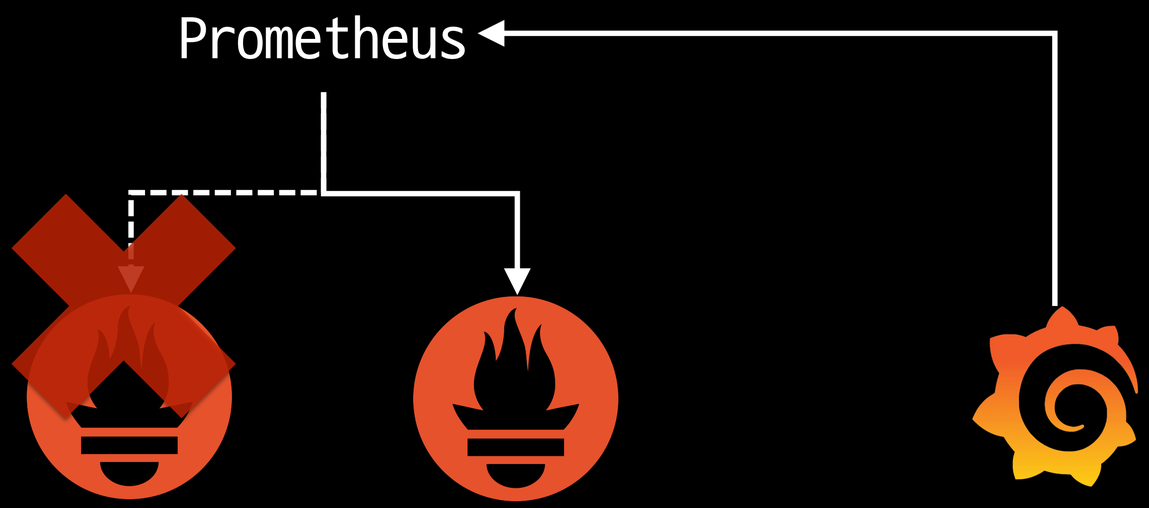
Head stats	
Number of Series	4백여만개 가량
Number of Chunks	1천여만개 가량

하지만,
클러스터의 크기가 커짐에 따라 메모리 사용량은 다시 증가할 수 밖에 없습니다.

3.3 문제: Scale-out 불가한 구조

NAVER
DEVIEW
2023

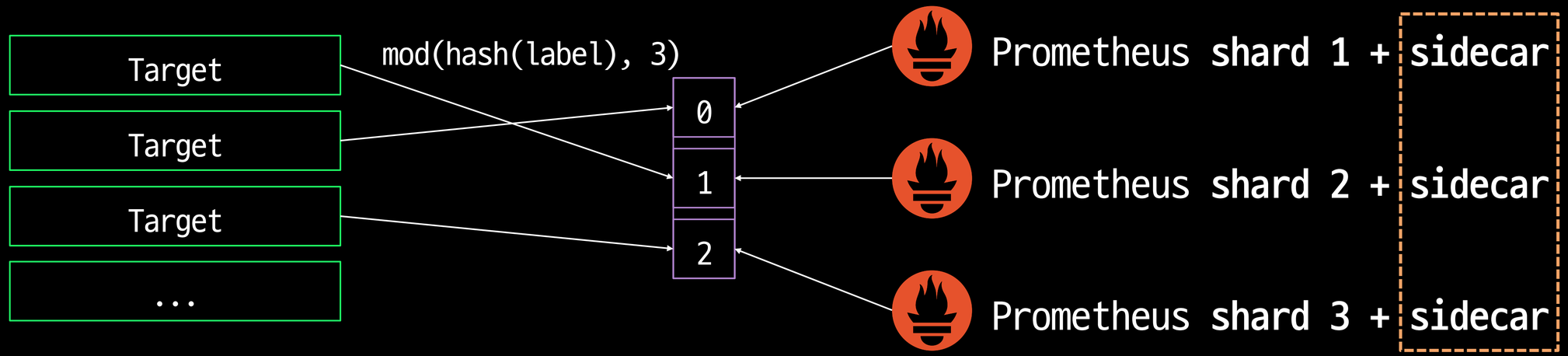
Prometheus는 **저장 + 조회 + 수집 + 알림** 까지 모두 담당하고 있습니다.
stand-by구조로 HA를 구성할 순 있지만, 단일 Prometheus가 받는 부담은 그대로입니다.



3.3 문제: Scale-out 불가한 구조

NAVER
DEVIEW
2023

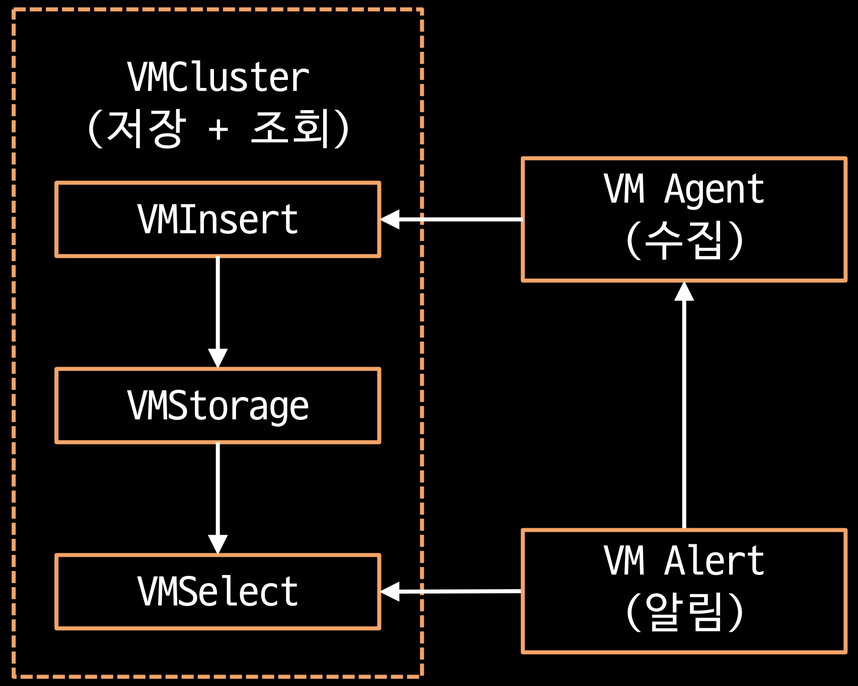
Prometheus는 **저장 + 조회 + 수집 + 알림** 까지 모두 담당하고 있습니다.
sharding을 구성할 순 있지만, Global Query를 위해서는 Thanos sidecar, Thanos querier 등의 부수적인 요소가 필수적입니다.



3.3 VictoriaMetrics 도입

NAVER
DEVIEW
2023

VictoriaMetrics는 **저장 + 조회 + 수집 + 알림** 을
각각의 Component가 나누어서 담당하고 있습니다.

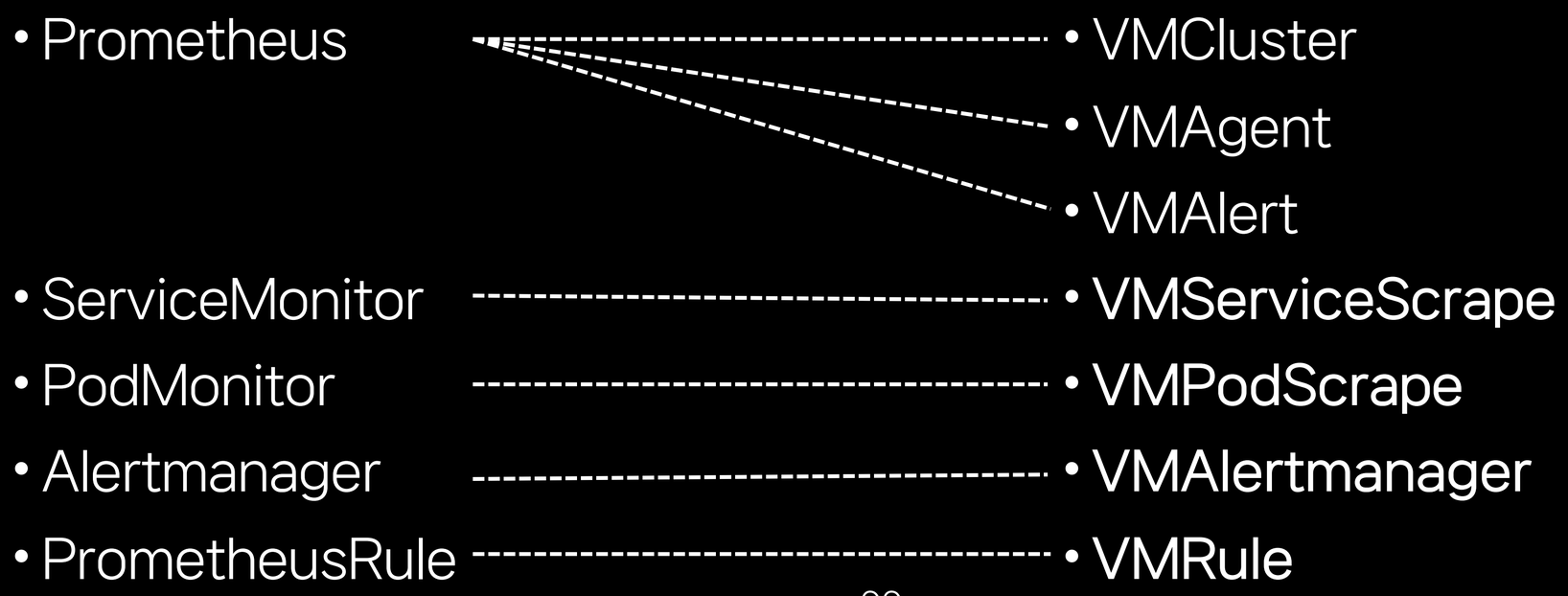


3.3 VictoriaMetrics 도입

NAVER
DEVIEW
2023

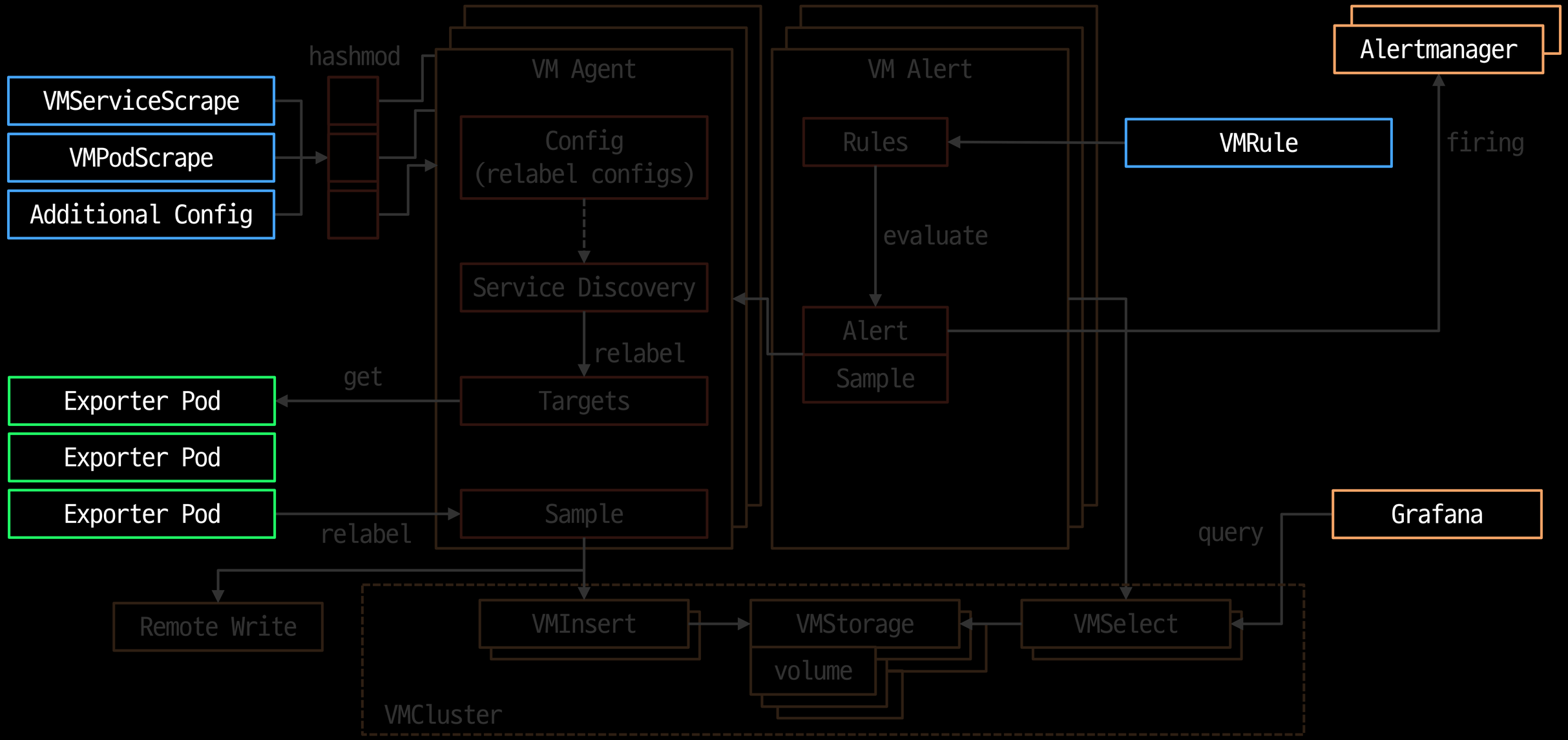
VictoriaMetrics Operator

- Prometheus와 마찬가지로 VictoriaMetrics도 Operator를 지원합니다.
- Compatibility를 지원하며 자동으로 변환까지 진행해줍니다.



3.3 VictoriaMetrics 도입

NAVER DEVIEW 2023



3.3 VictoriaMetrics 도입

NAVER
DEVIEW
2023

(저장 + 조회)

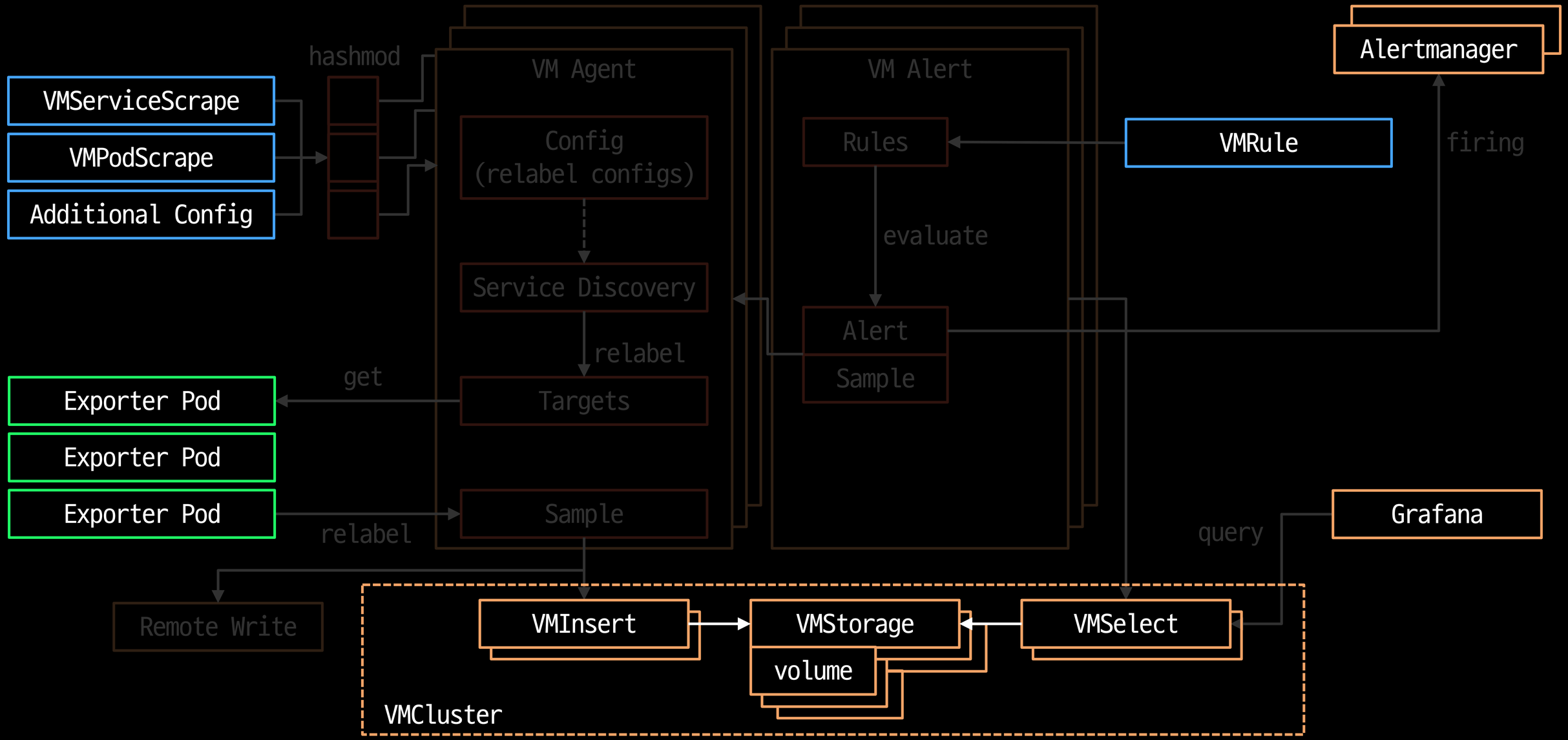
VictoriaMetrics cluster components:

- VMInsert (Deployment):
  유입된 지표를 적절한 VMStorage로 전달
- VMStorage (StatefulSet):
  지표를 저장
- VMSelect (StatefulSet):
  VMStorage로부터 데이터 조회

```yaml
apiVersion: operator.victoriametrics.com/v1beta1
kind: VMCluster
metadata:
  name: main
spec:
  retentionPeriod: 30d
  vmstorage:
    replicaCount: 4
    resources:
      requests:
        cpu: "1"
        memory: 1Gi
  vmselect:
    replicaCount: 2
    resources:
      requests:
        cpu: "1"
        memory: "1Gi"
  vminsert:
    replicaCount: 2
    resources:
      requests:
        cpu: "1"
        memory: "1Gi"
```

3.3 VictoriaMetrics 도입

NAVER DEVIEW 2023



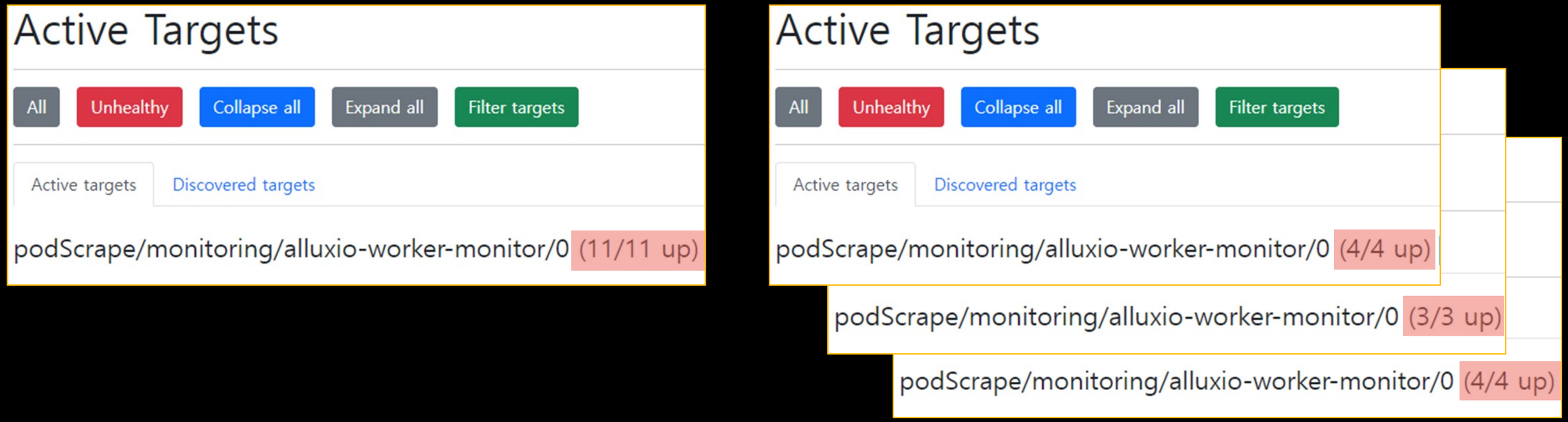
3.3 VictoriaMetrics 도입

NAVER
DEVIEW
2023

(수집)

VMAgent를 통해 수집을 진행합니다.

- sharding을 지원하여, hashmod를 통해 수집 대상을 분산할 수 있습니다.



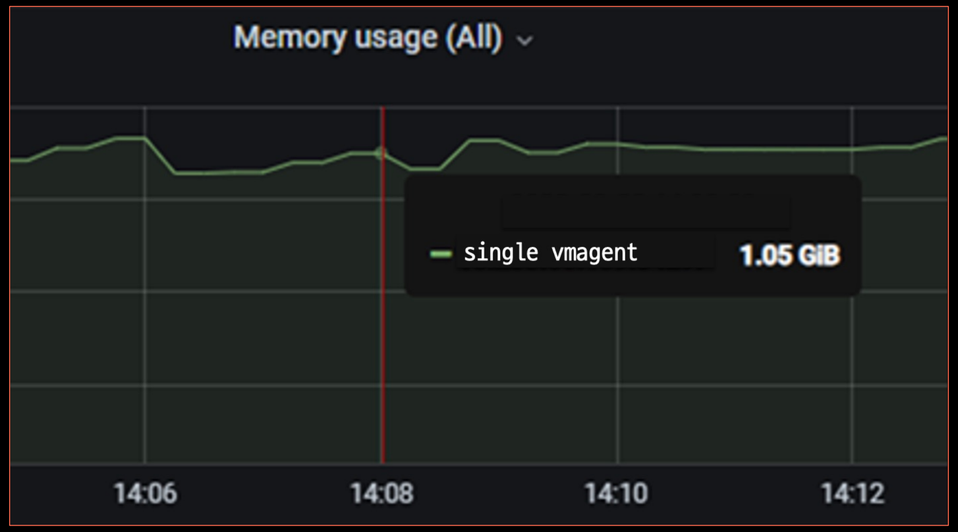
3.3 VictoriaMetrics 도입

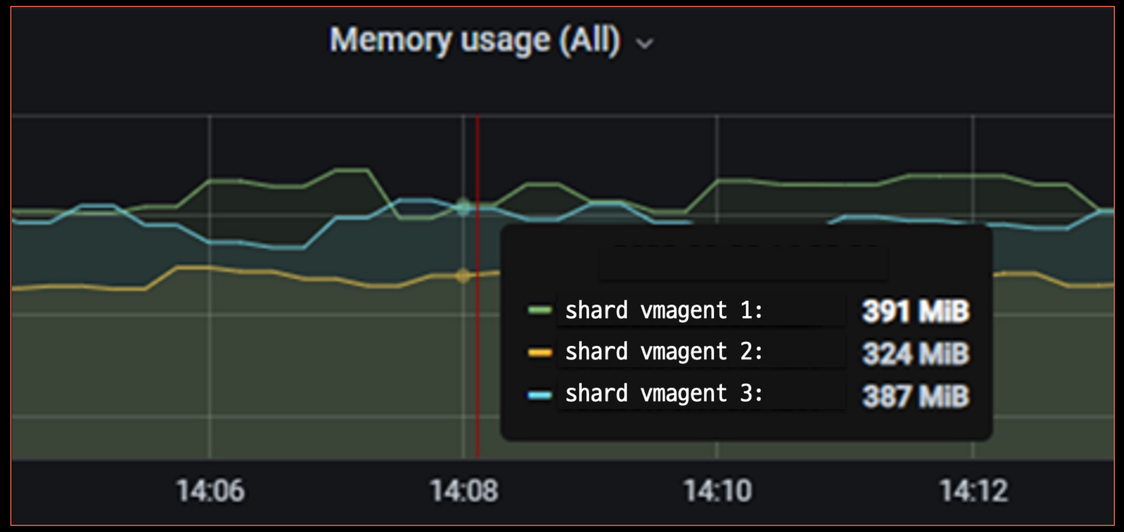
NAVER
DEVIEW
2023

(수집)

VMAgent를 통해 수집을 진행합니다.

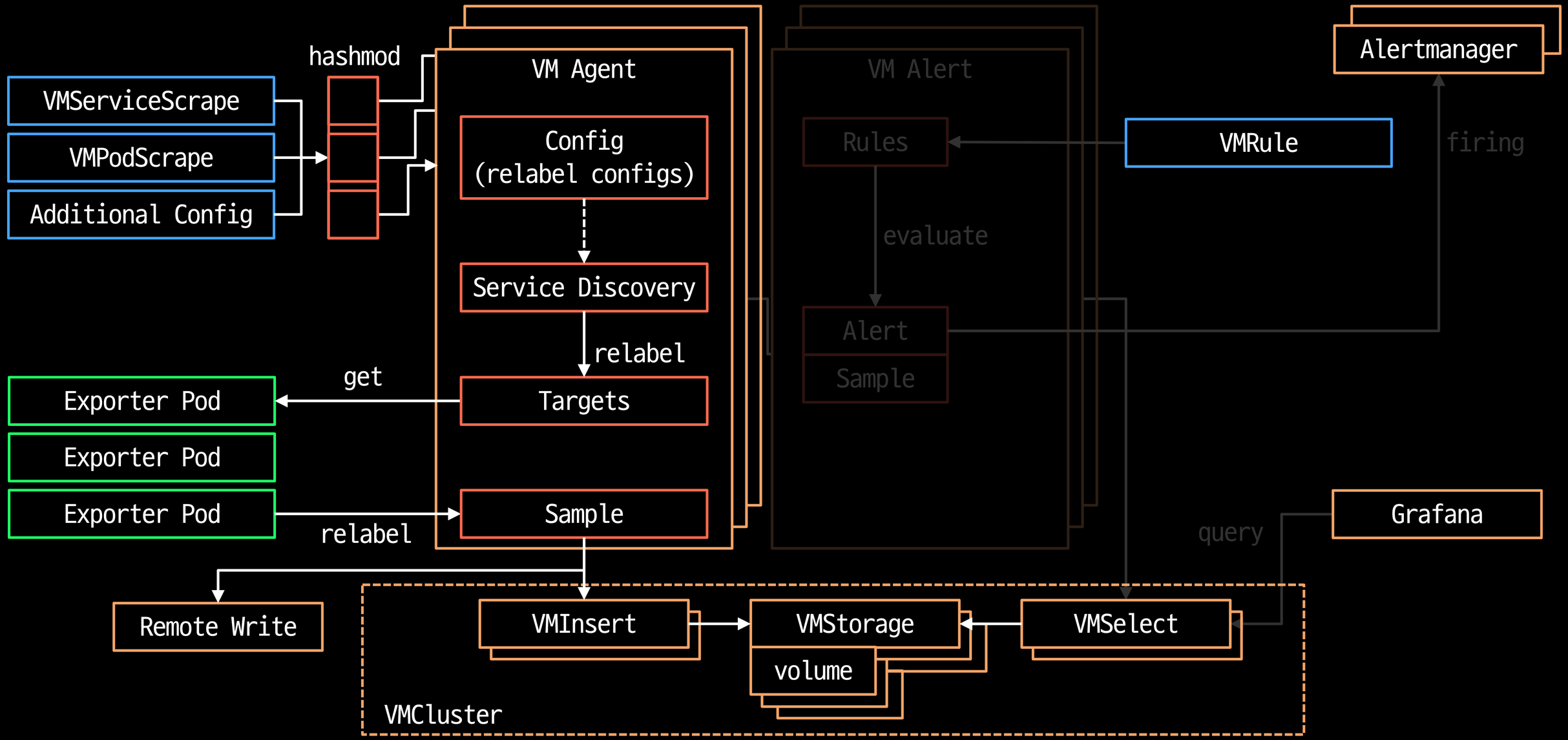
- sharding을 지원하여, hashmod를 통해 수집 대상을 분산할 수 있습니다.
- 리소스 관리와 문제 발생시 피해를 최소화 할 수 있습니다.





3.3 VictoriaMetrics 도입

NAVER DEVIEW 2023



3.3 VictoriaMetrics 도입

NAVER
DEVIEW
2023

(알림)

VMAlert을 통해 alert, recording rule을 수행할 수 있습니다.

alert: KubeContainerWaiting (for: 3600 seconds)

```
sum by (namespace, pod, container) (kube_pod_container_status_waiting_reason{job="kube-state-metrics"}) > 0
```

Labels: `severity=warning`

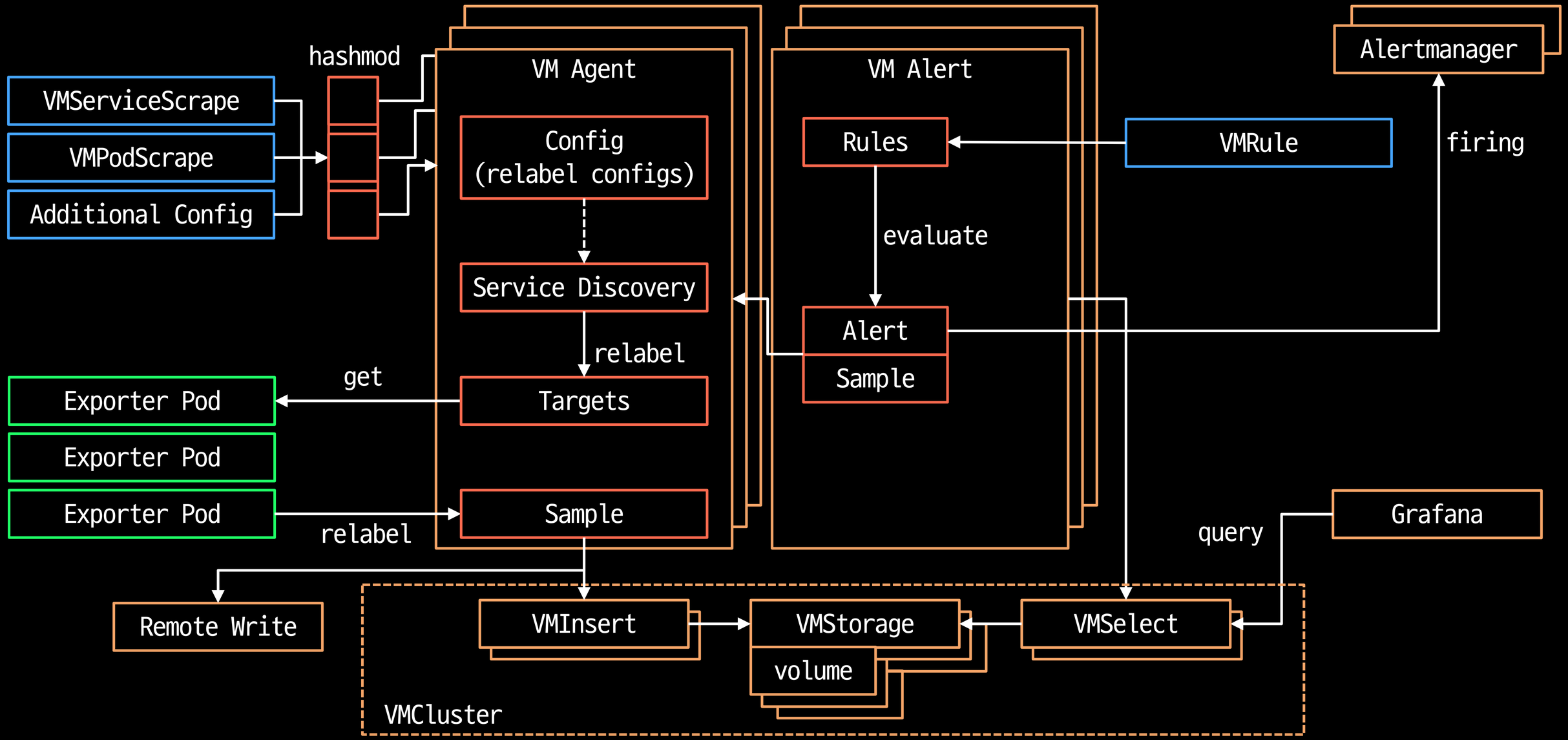
record: code_verb:apiserver_request_total:increase1h

```
sum by (code, verb) (increase(apiserver_request_total{job="apiserver",verb="GET",code=~"4.."}[1h]))
```

https://raw.githubusercontent.com/prometheus-operator/kube-prometheus/main/manifests/kubernetesControlPlane-prometheusRule.yaml

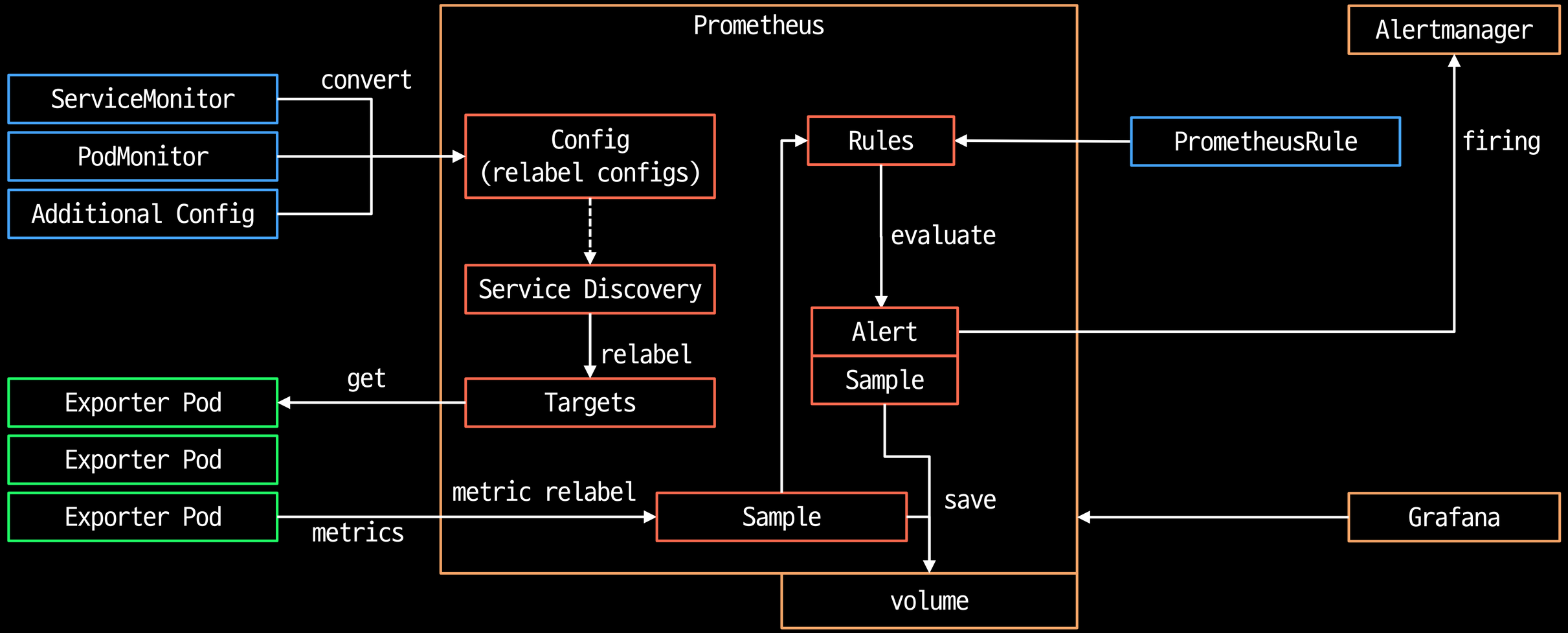
3.3 VictoriaMetrics 도입

NAVER DEVIEW 2023



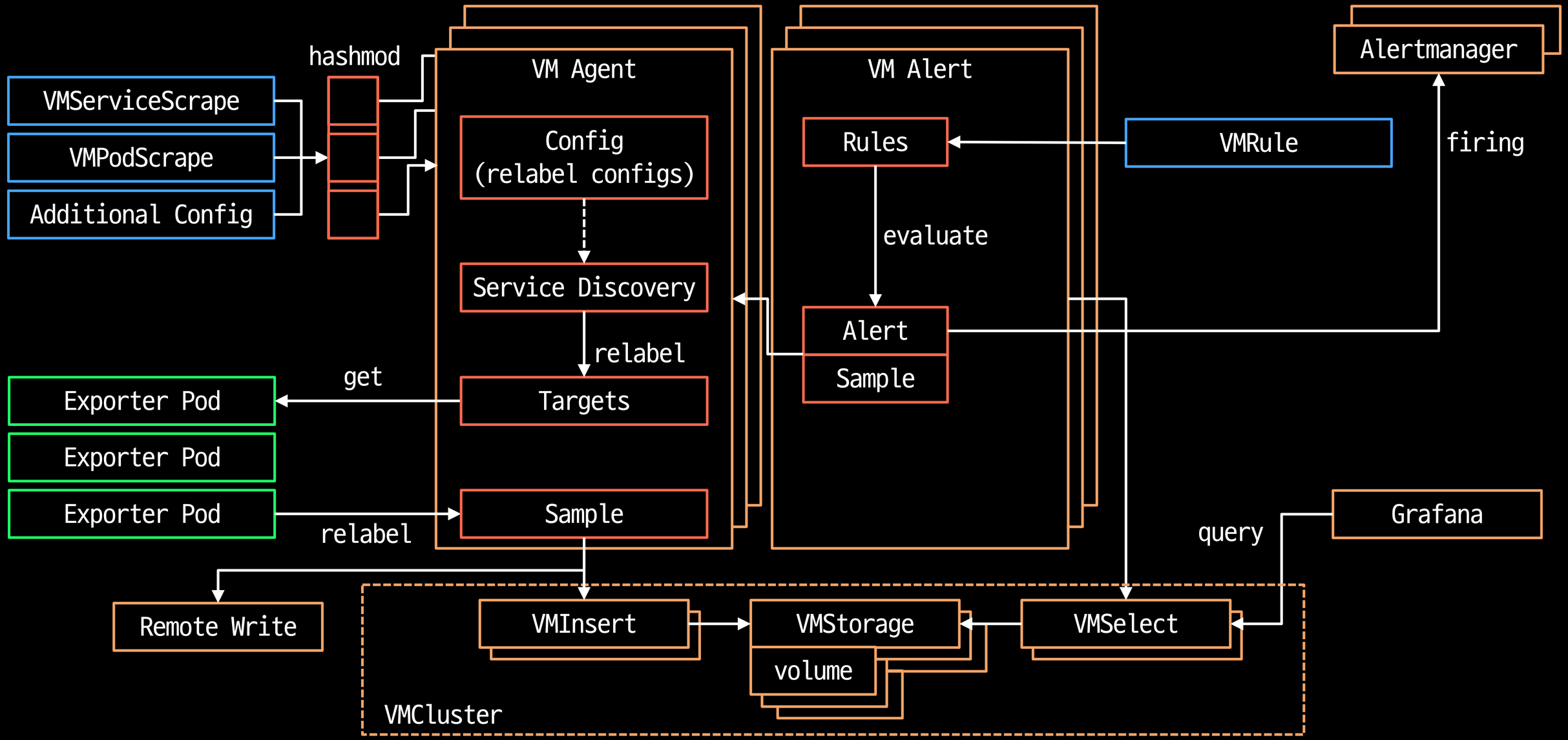
3.3 VictoriaMetrics 도입: AS-IS

NAVER DEVIEW 2023



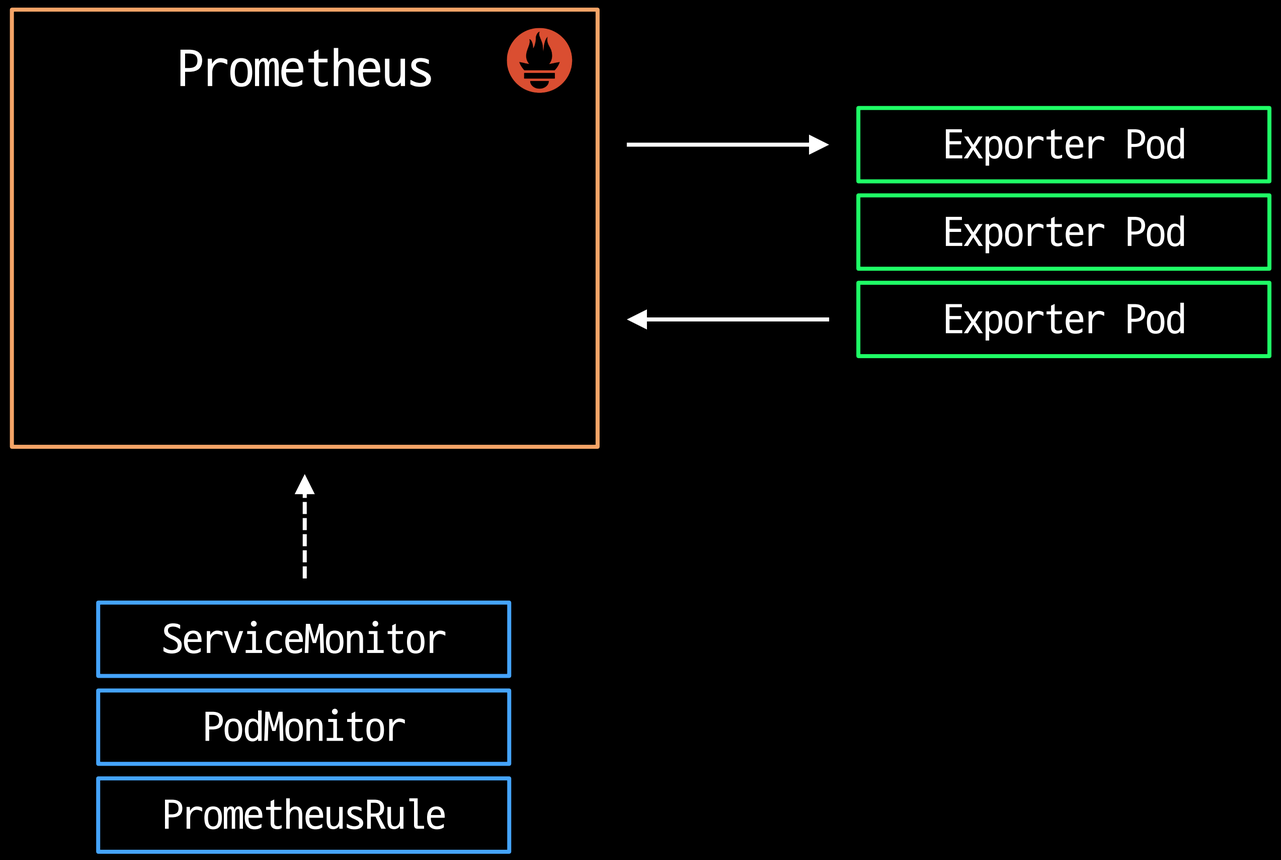
3.3 VictoriaMetrics 도입: TO-BE

NAVER DEVIEW 2023



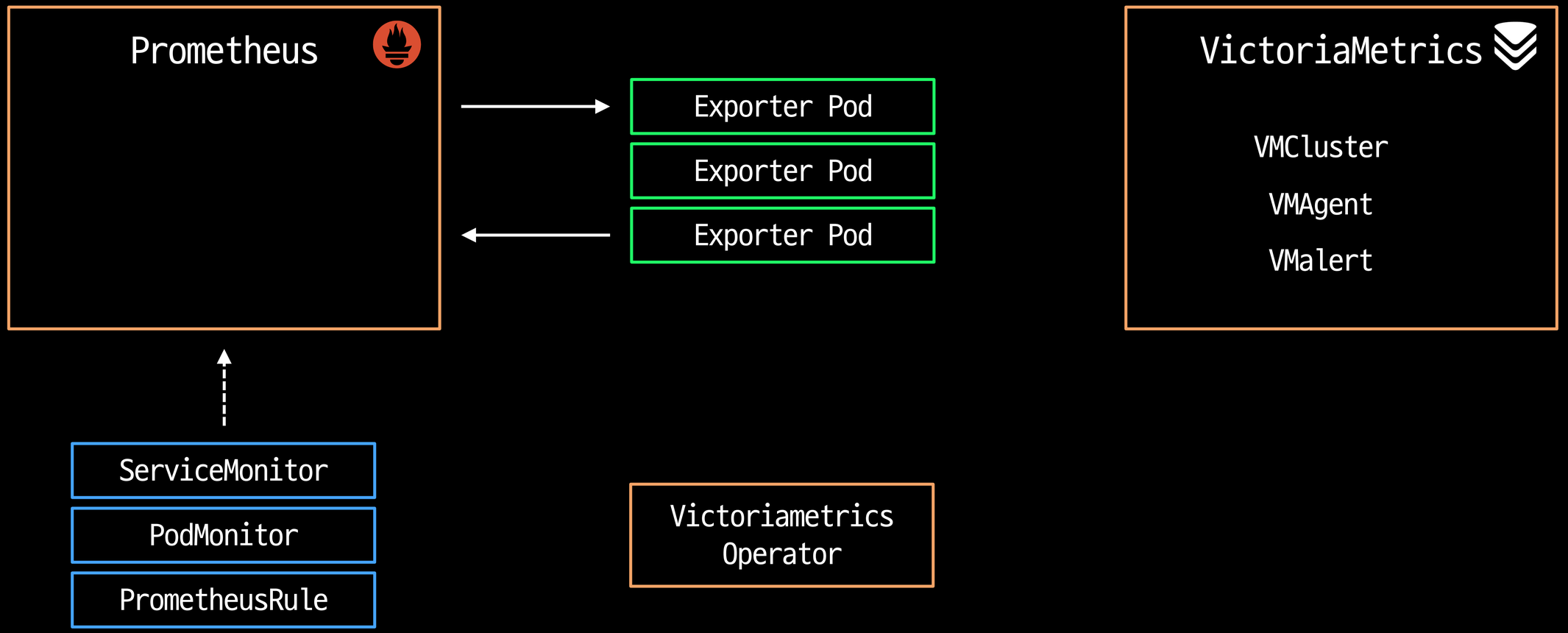
3.3 VictoriaMetrics 도입: Migration

NAVER
DEVIEW
2023



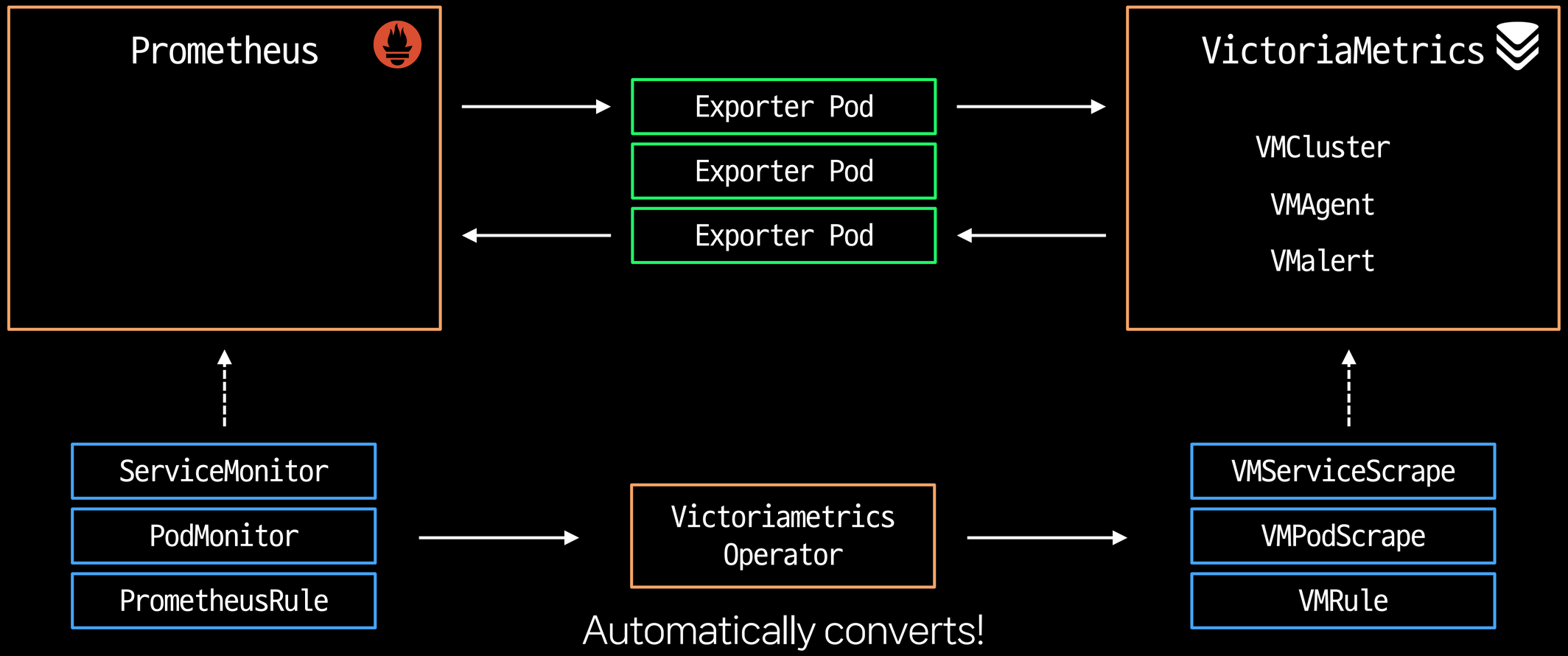
3.3 VictoriaMetrics 도입: Migration

NAVER
DEVIEW
2023



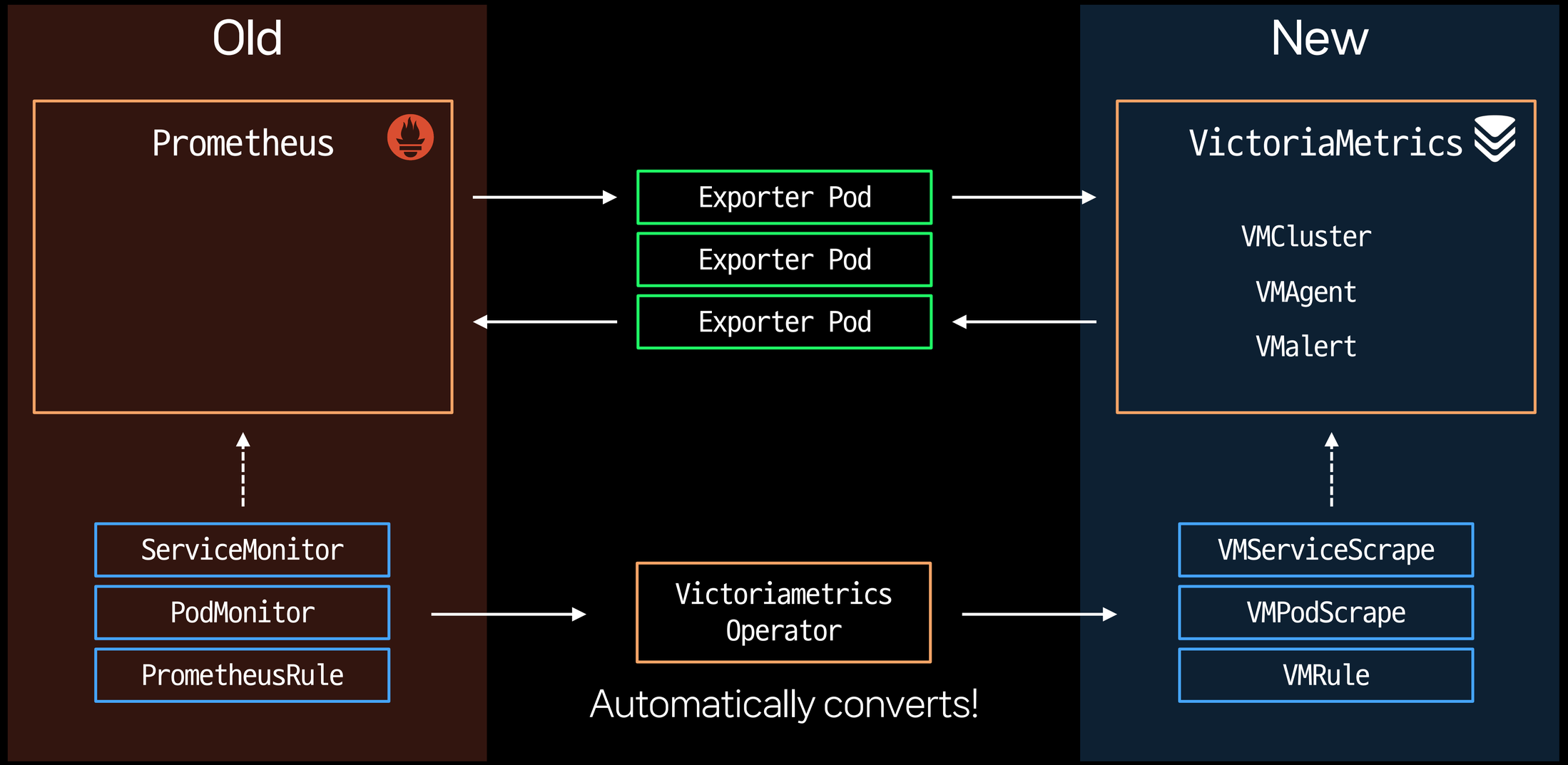
3.3 VictoriaMetrics 도입: Migration

NAVER
DEVIEW
2023



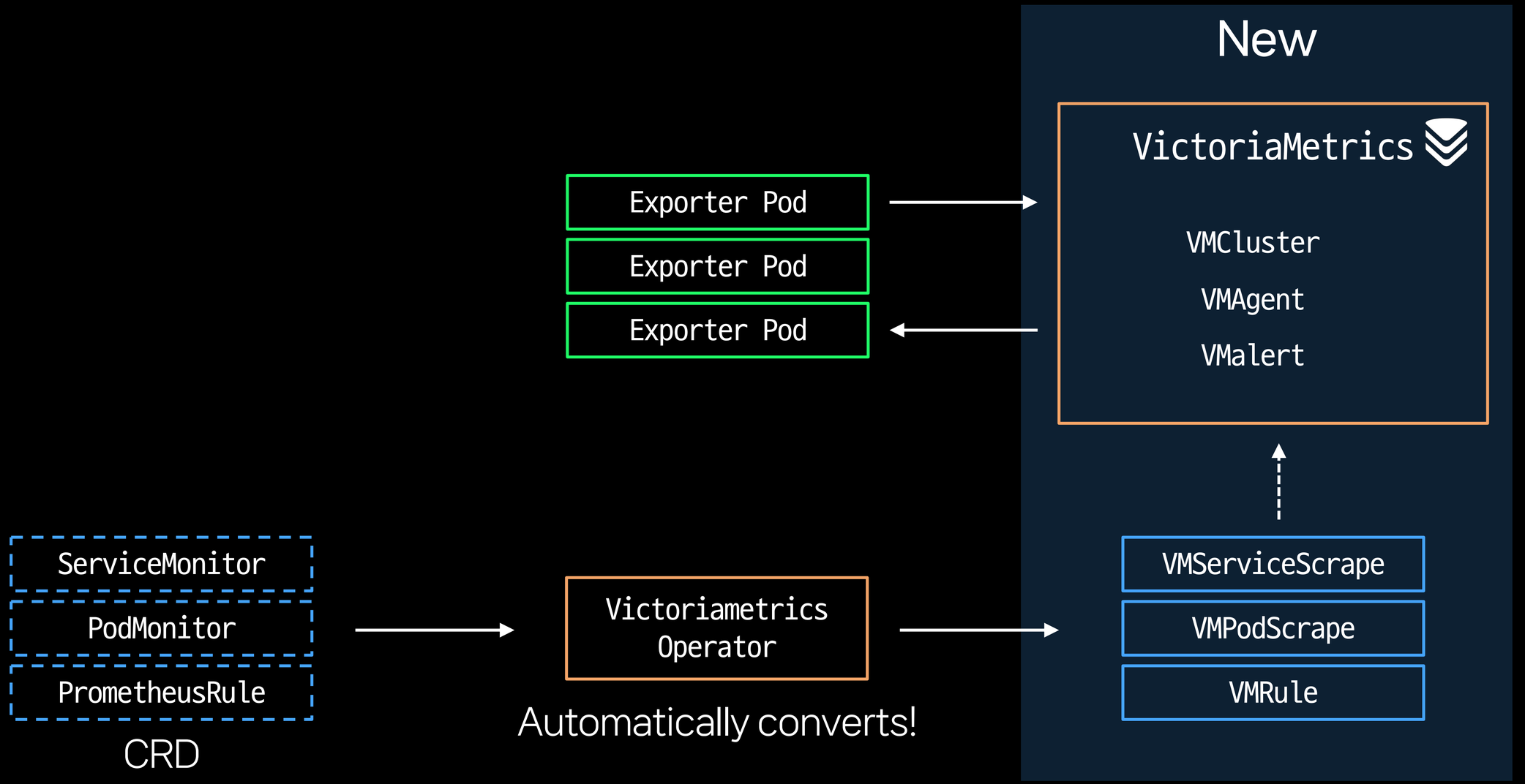
3.3 VictoriaMetrics 도입: Migration

NAVER
DEVIEW
2023



3.3 VictoriaMetrics 도입: Migration

NAVER DEVIEW 2023



3.3 VictoriaMetrics 도입

NAVER
DEVIEW
2023

- VMCluster(VMSelect + VMInsert + VMStorage), VMAlert , VMAgent 모든 Component를 Scale-out할 수 있습니다.
- 기존 Prometheus와 병행하여 운영이 가능합니다.
- 메모리 사용량 또한 눈에 띄게 감소하였습니다.

테스트 결과	Prometheus	VictoriaMetrics
CPU Usage	14.7	14.6
Memory Usage	104 GiB	52GiB
Receive Bandwidth	143 MB/s	173 MB/s
Transmit Bandwidth	2.20 MB/s	33 MB/s

4. 향후 계획

4.1 향후 계획

NAVER
DEVIEW
2023

더 안정적인 서빙을 위해...

- Istio RFC: Simplify Istio Multi Tenancy
- Sidecar없는 Service mesh (Istio ambient mesh, Cilium service mesh)
- Multi-Tenant ingress 지원
- 사용자마다의 모니터링 환경 제공

Q&A