@ThanosMetrics

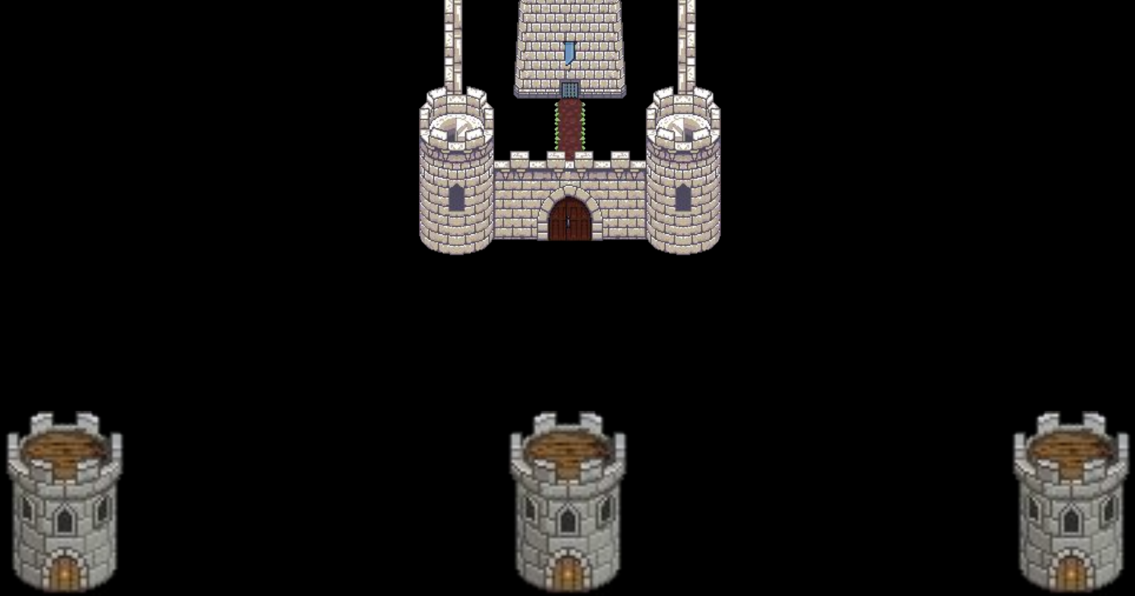
KubeCon + CNCF San Diego - November 19, 2019

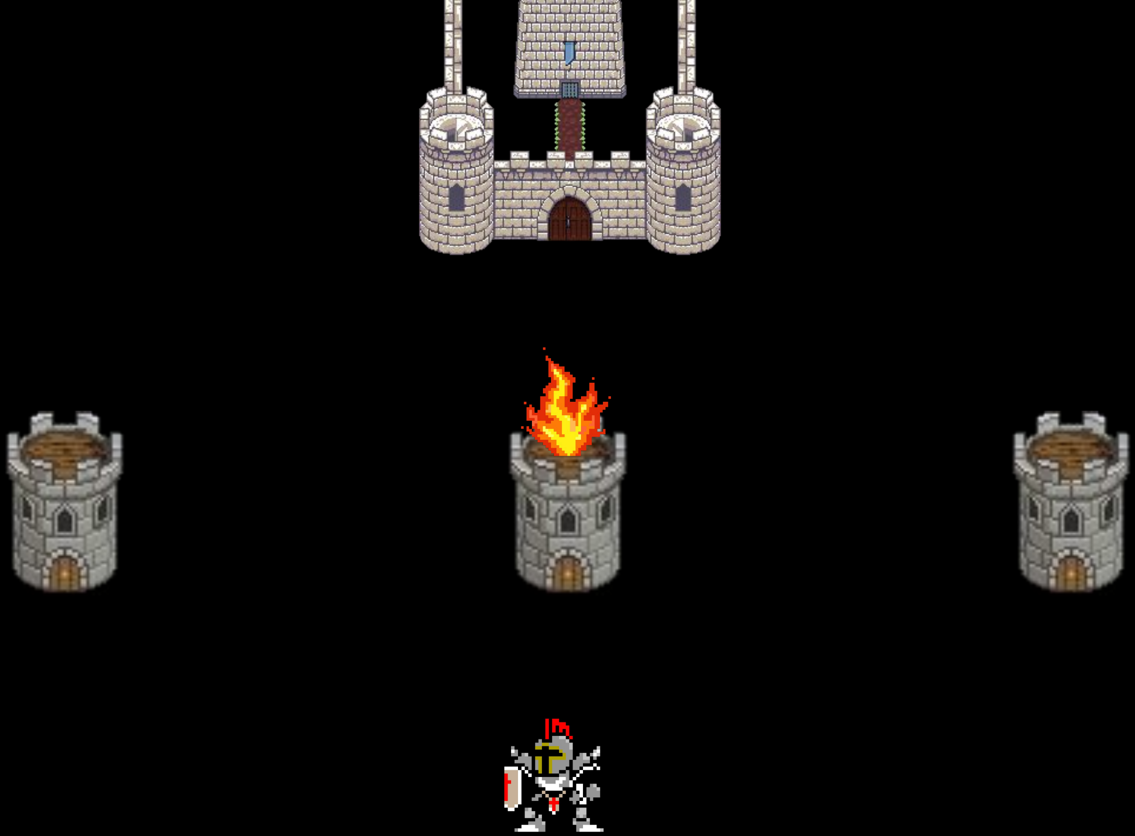
Thanos

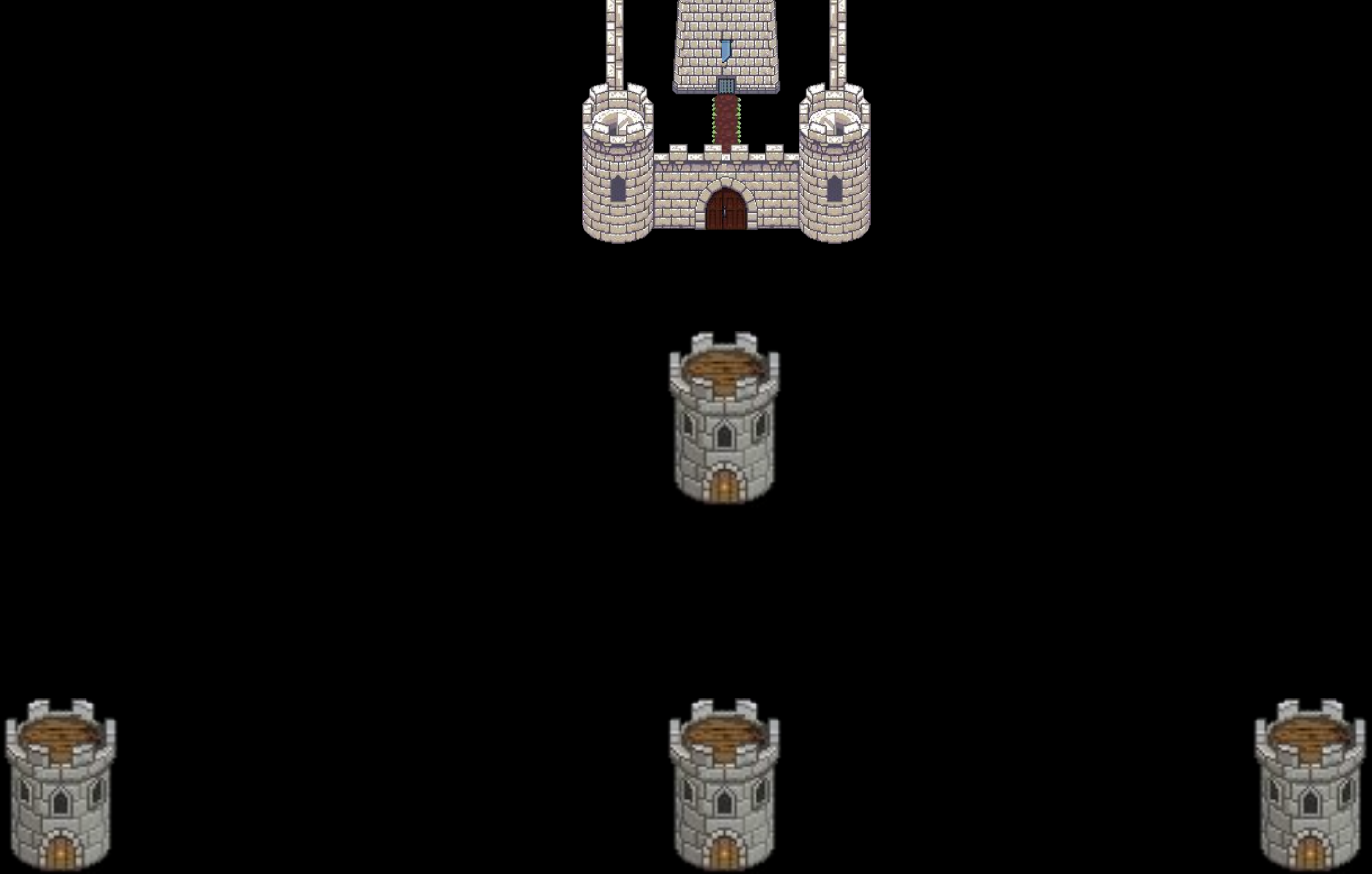
Introduction to Thanos

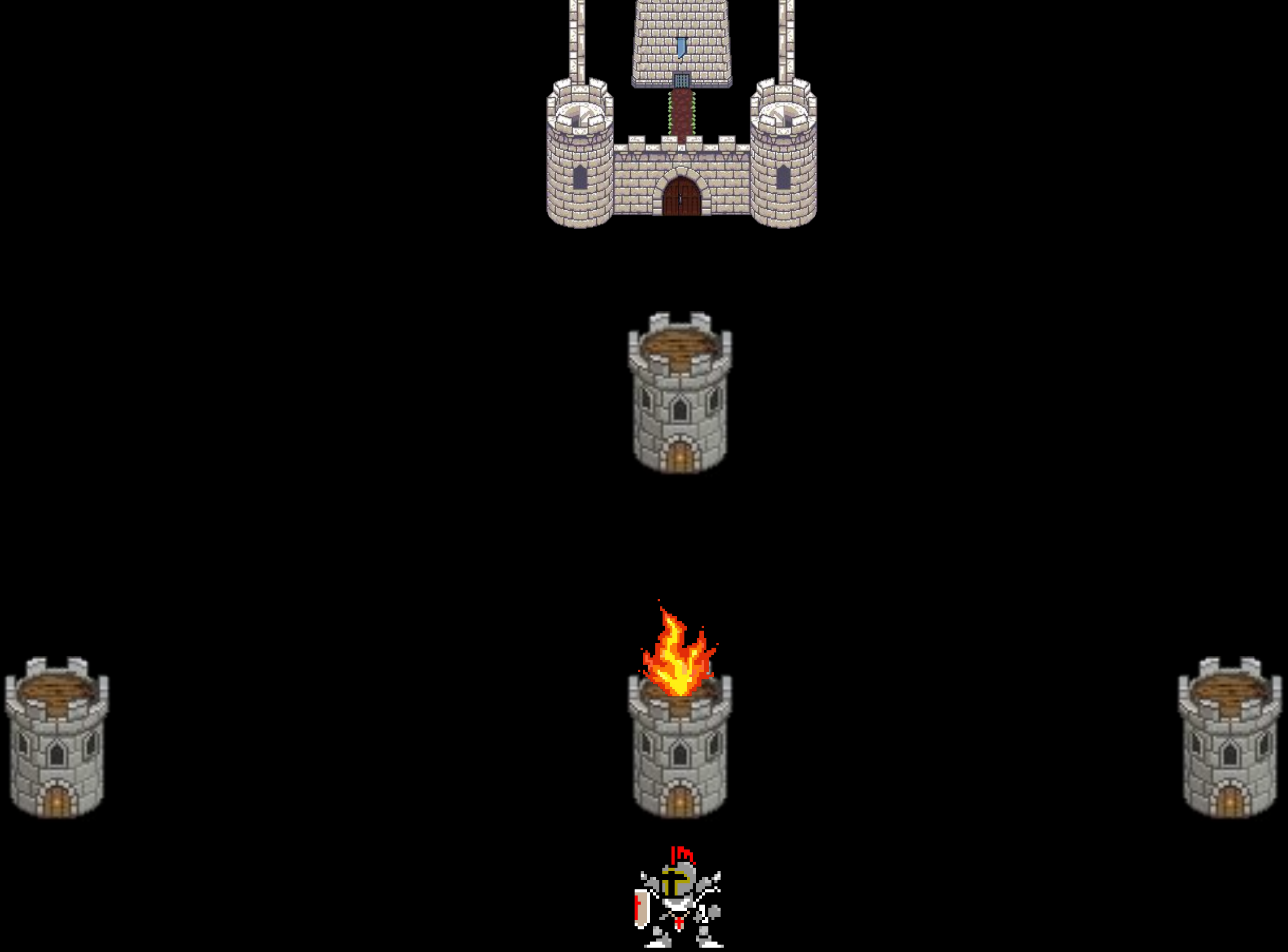
Dominic Green, Software Engineer (Improbable)
dom@improbable.io
domgreen

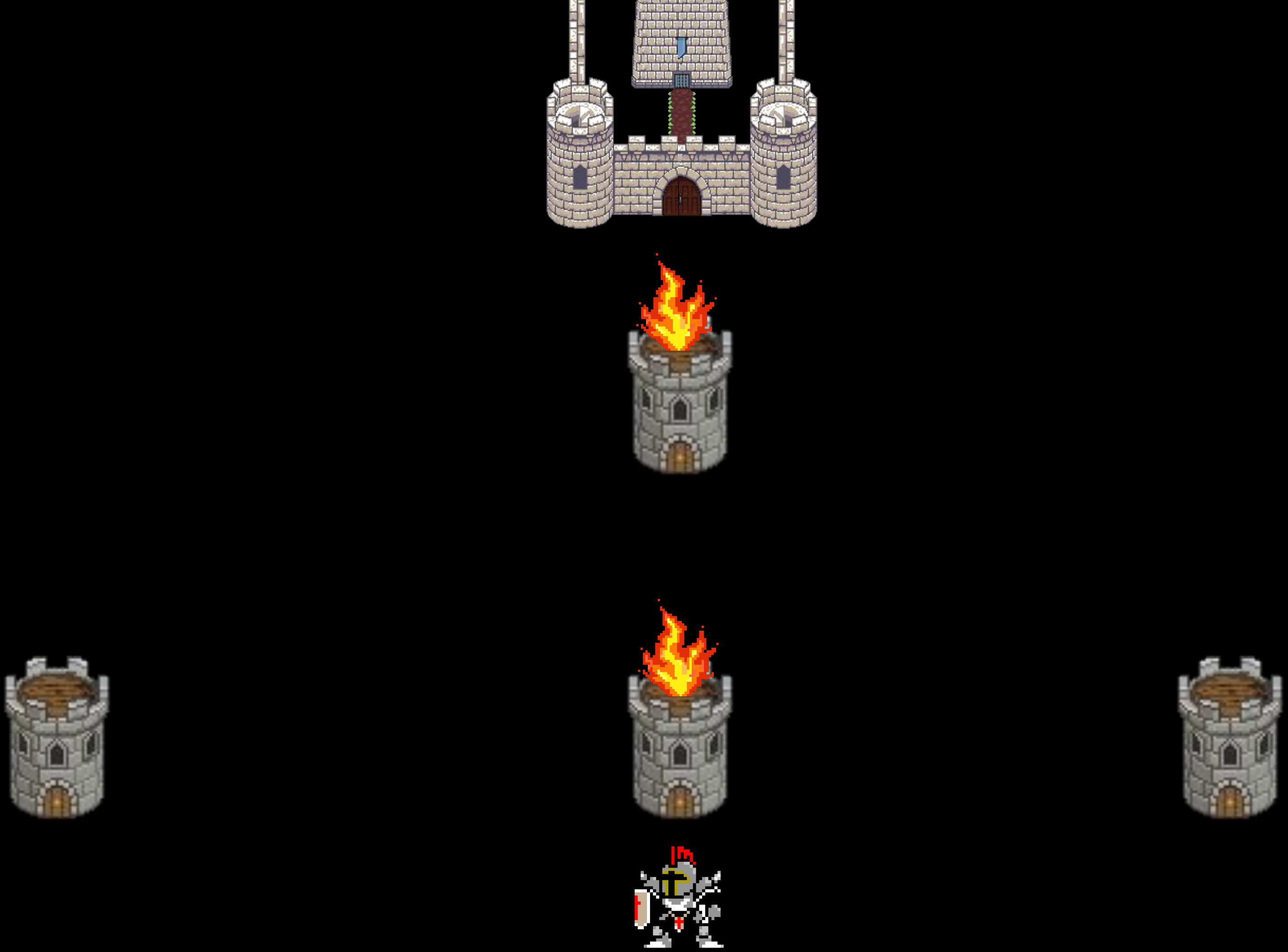
Lucas Servén, Senior Software Engineer (Red Hat)
lserven@redhat.com
squat

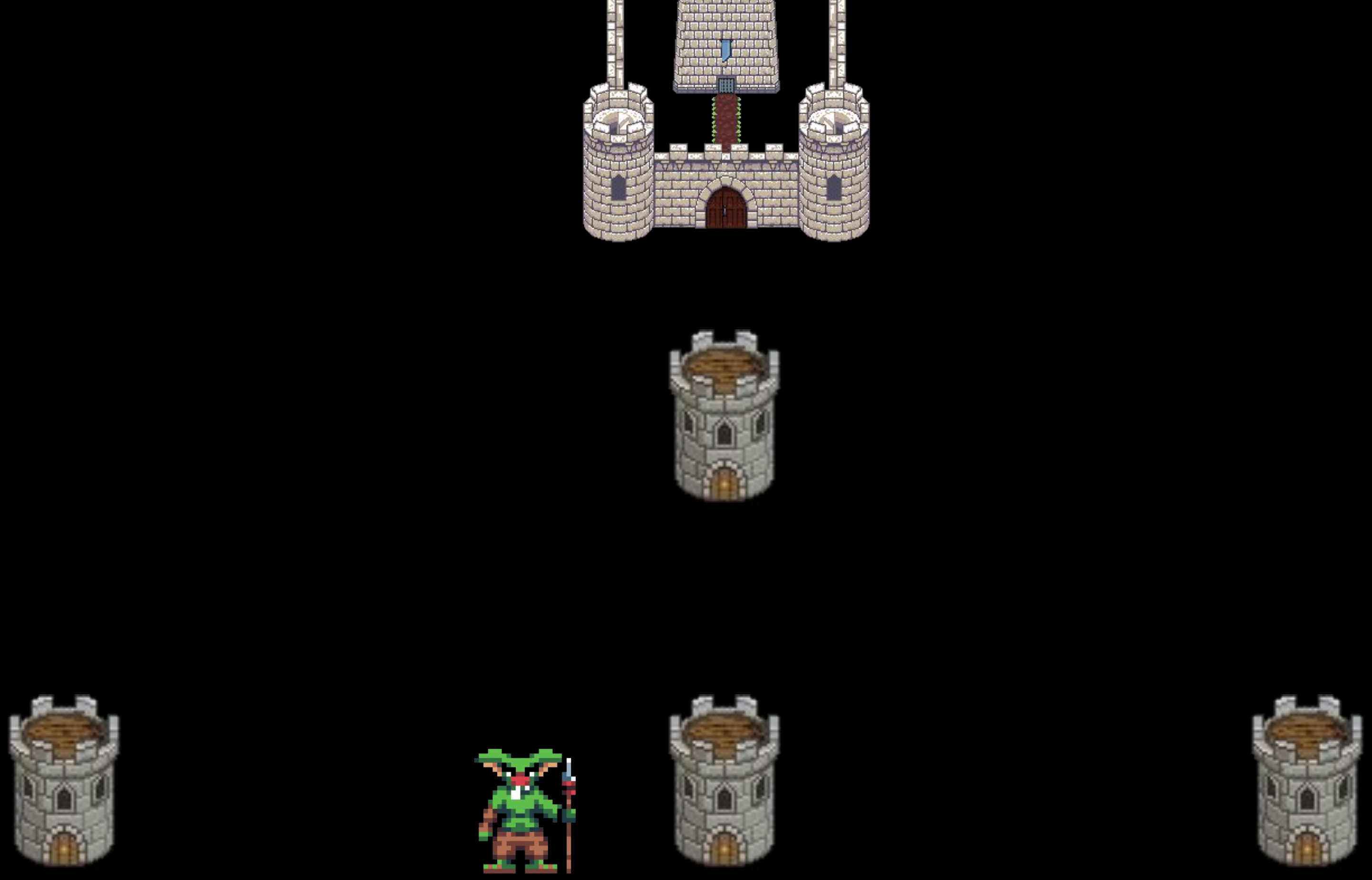


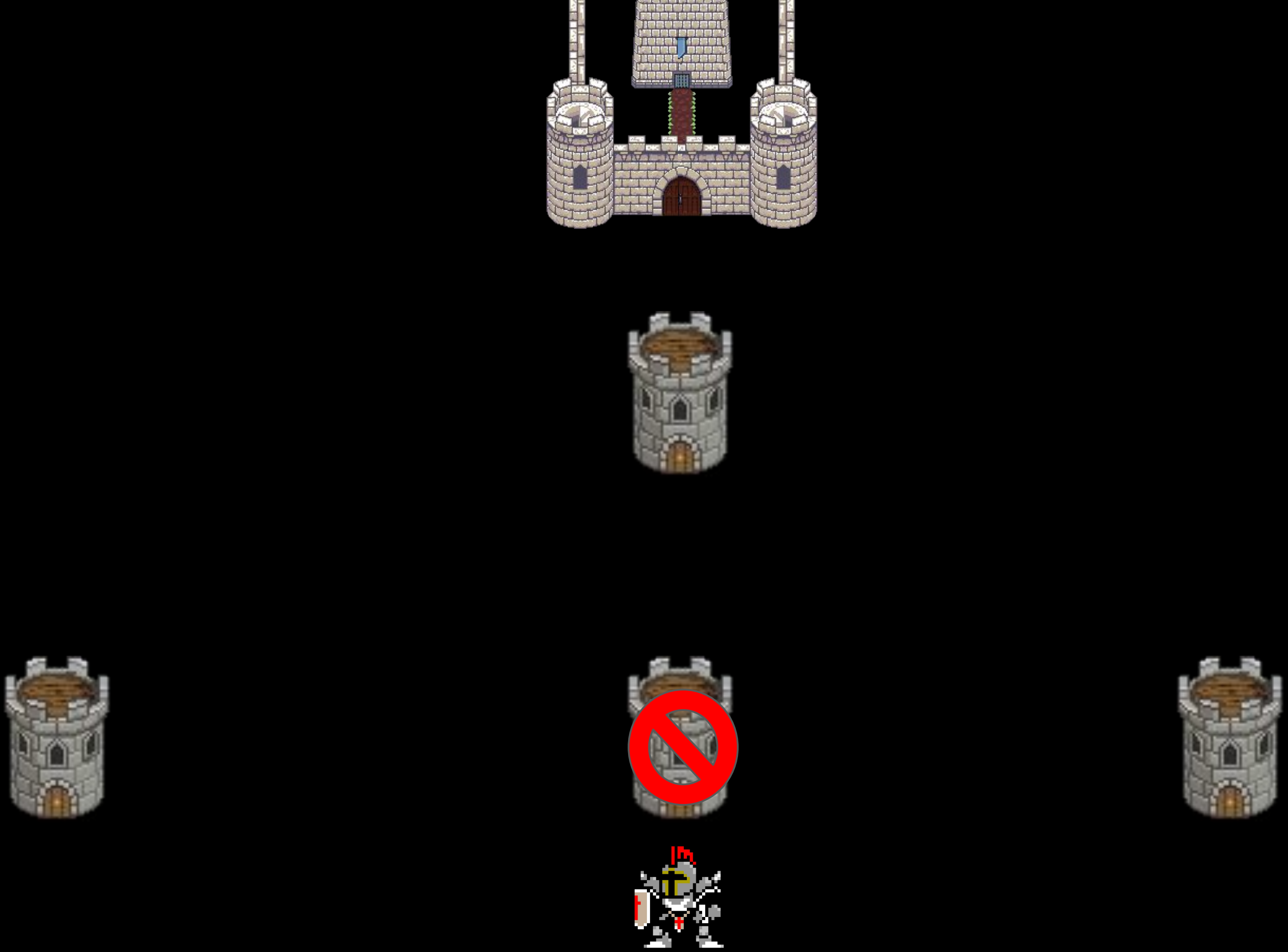


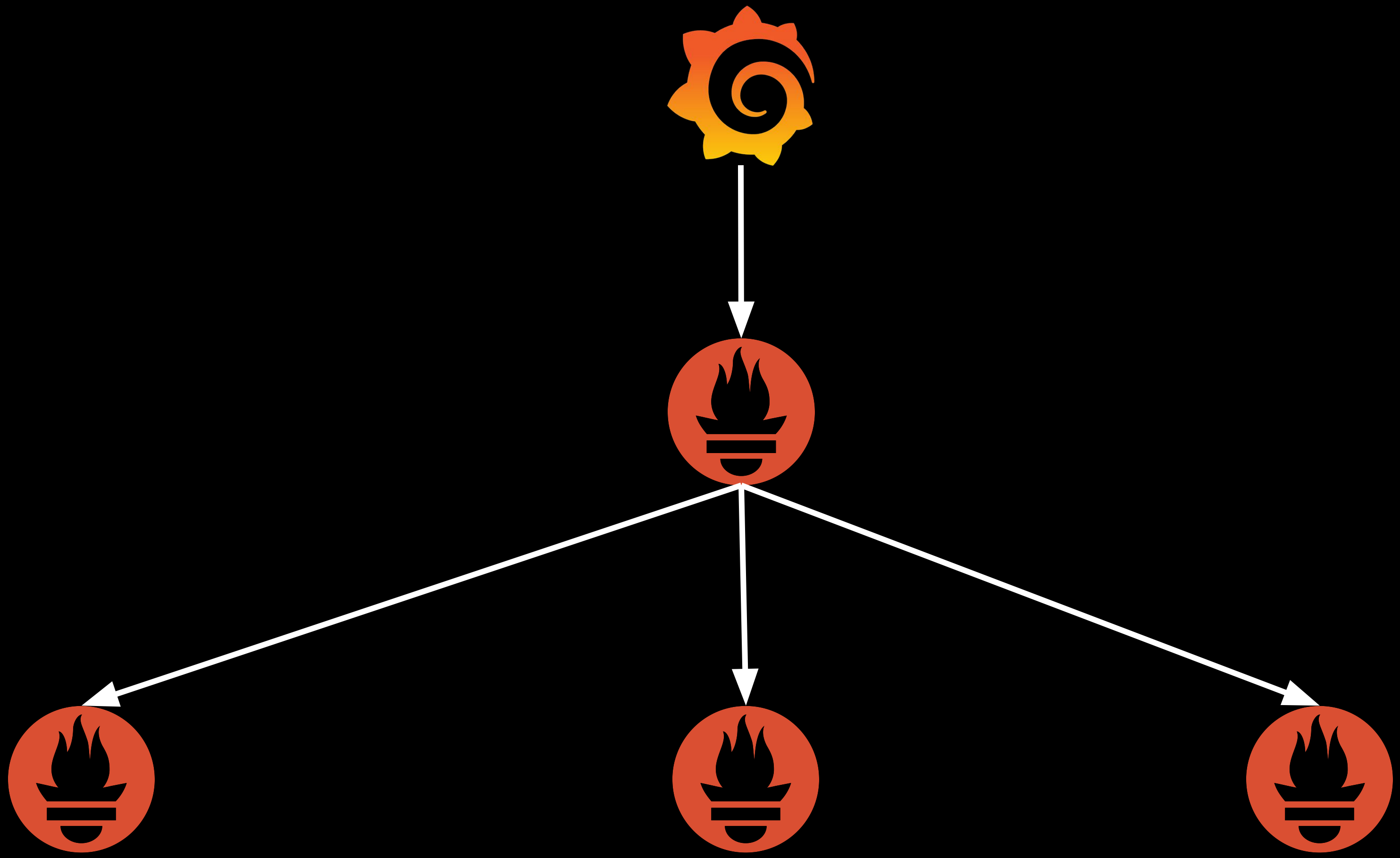












Dominic Green

- Software Engineer @ Improbable
- Observability Team
- OSS Contributor
  - Thanos
  - go-grpc-middleware
  - go-httpwares
- Meetup Organiser
  - Prometheus London
  - London Gophers

Lucas Servén Marín

- Senior Software Engineer @ Red Hat
- OpenShift Monitoring Team
- OSS Contributor
  - Thanos
  - Prometheus Operator
  - Kilo
- Twin

@ThanosMetrics

Thanos

Thanos Community

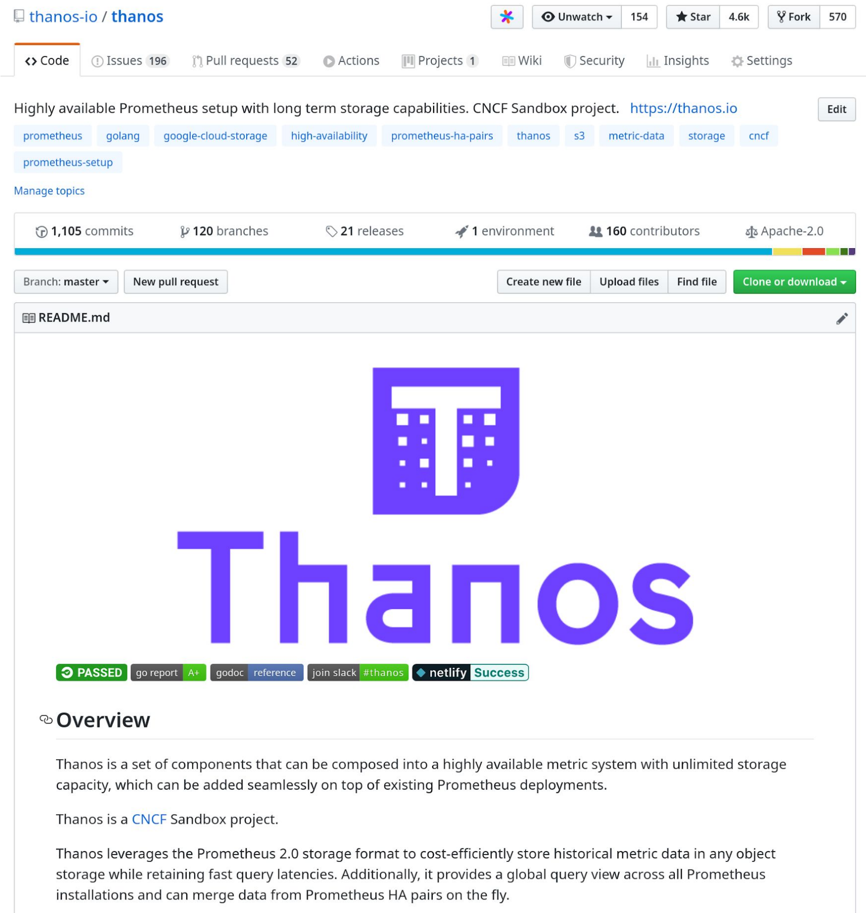
@ThanosMetrics

- Fully open source from start
- Started in Nov 2017
- Part of **CNCF Sandbox**

- 4600+ Github stars
- 162+ contributors
- ~500 slack users
- 8 maintainers, 3 triagers from 7 different companies.

- Transparent Governance

- Prometheus Ecosystem



Highly available Prometheus setup with long term storage capabilities. CNCF Sandbox project. https://thanos.io

Overview

Thanos is a set of components that can be composed into a highly available metric system with unlimited storage capacity, which can be added seamlessly on top of existing Prometheus deployments.

Thanos is a CNCF Sandbox project.

Thanos leverages the Prometheus 2.0 storage format to cost-efficiently store historical metric data in any object storage while retaining fast query latencies. Additionally, it provides a global query view across all Prometheus installations and can merge data from Prometheus HA pairs on the fly.

Production Users

Adobe

Alibaba.com™

BlaBlaCar

ebay™

GitLab

Hello FRESH

IMPROBABLE

monzo

Red Hat

🤔

Define: Monitoring

> “
> Collecting, processing, aggregating, and displaying real-time quantitative data about a system, such as query counts and types, error counts and types, processing times, and server lifetimes.
> ”

https://landing.google.com/sre/sre-book/chapters/monitoring-distributed-systems/

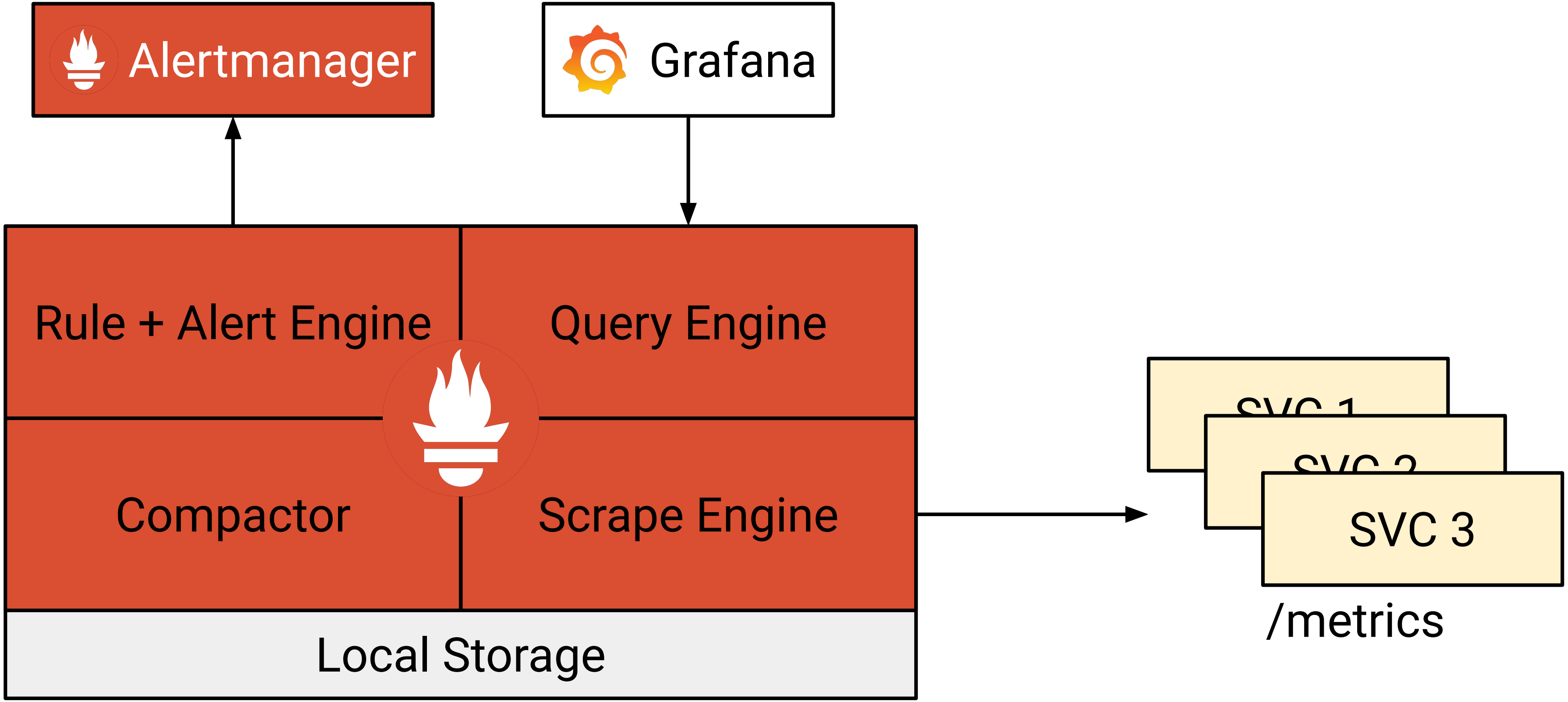
Prometheus

Prometheus

/metrics

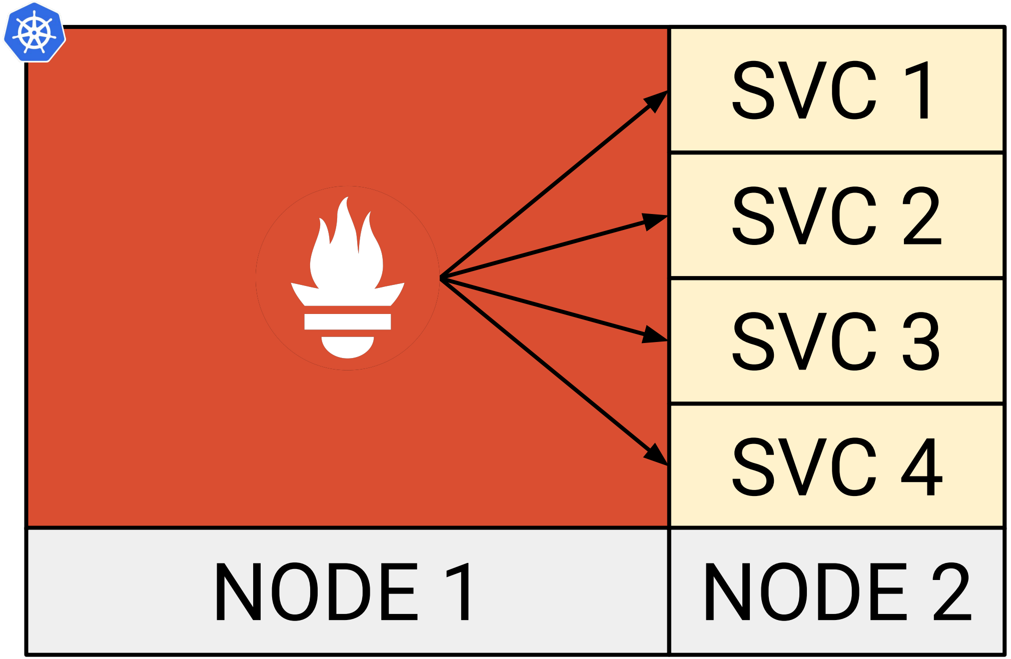
```
# TYPE counter
app_request_total 1337

# TYPE gauge
app_request_in_flight_total 3

# TYPE histogram
app_request_duration_ms_bucket {le="0.005"} 500
app_request_duration_ms_bucket {le="0.01"} 213
```

Prometheus


Alertmanager
Grafana
Rule + Alert Engine
Query Engine
Compactor
Scrape Engine
Local Storage
SVC 1
SVC 2
SVC 3
/metrics

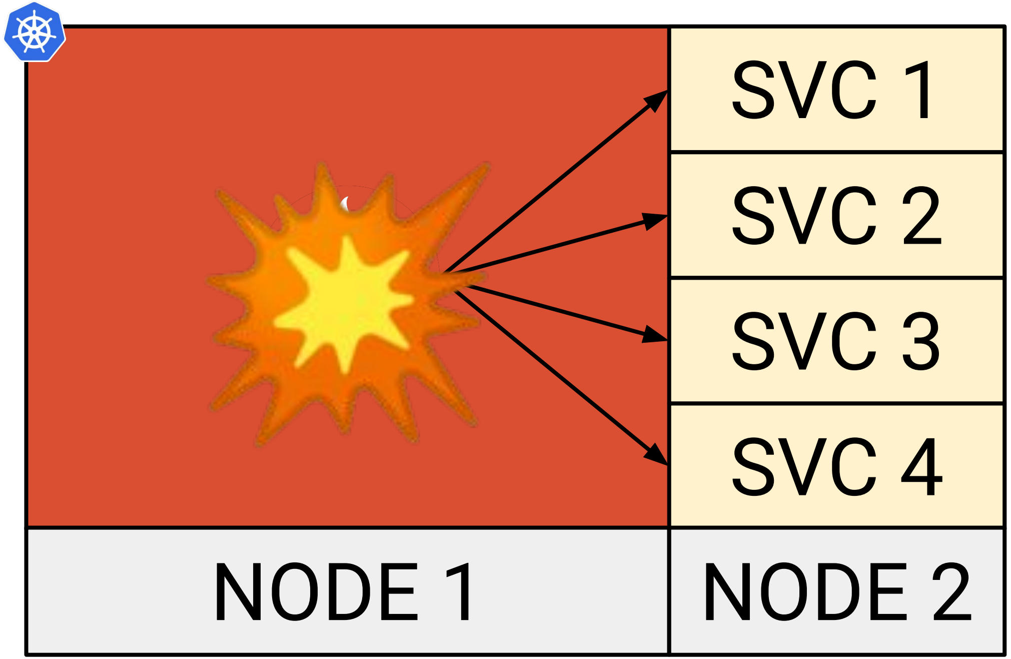
Prometheus Limitations

@ThanosMetrics

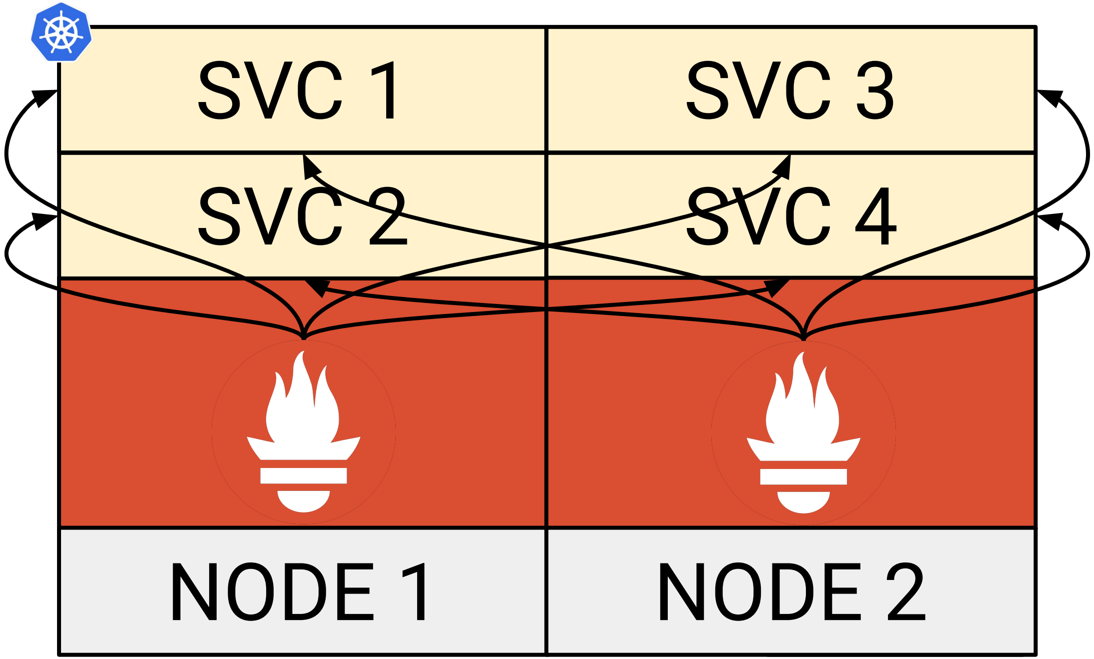


Prometheus Limitations

@ThanosMetrics

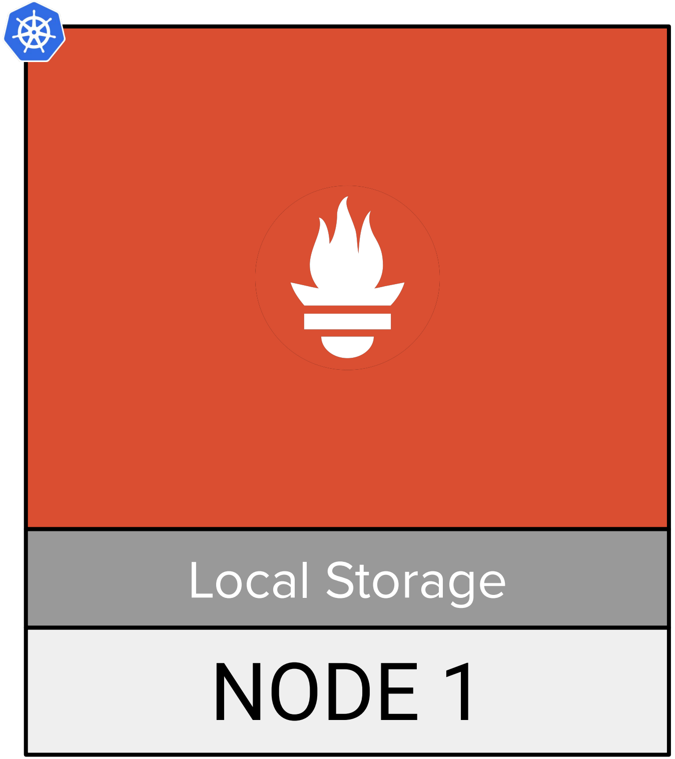


Prometheus Limitations



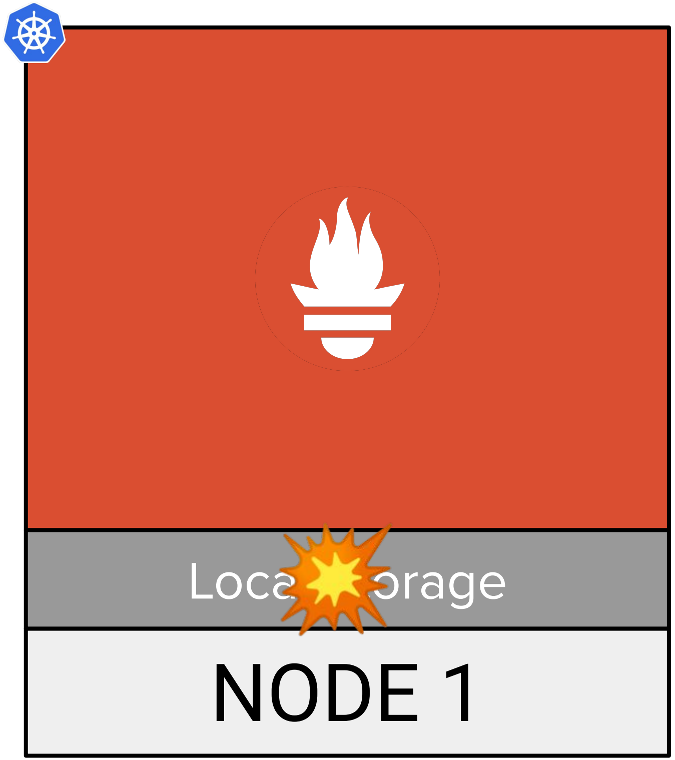
Prometheus Limitations

@ThanosMetrics

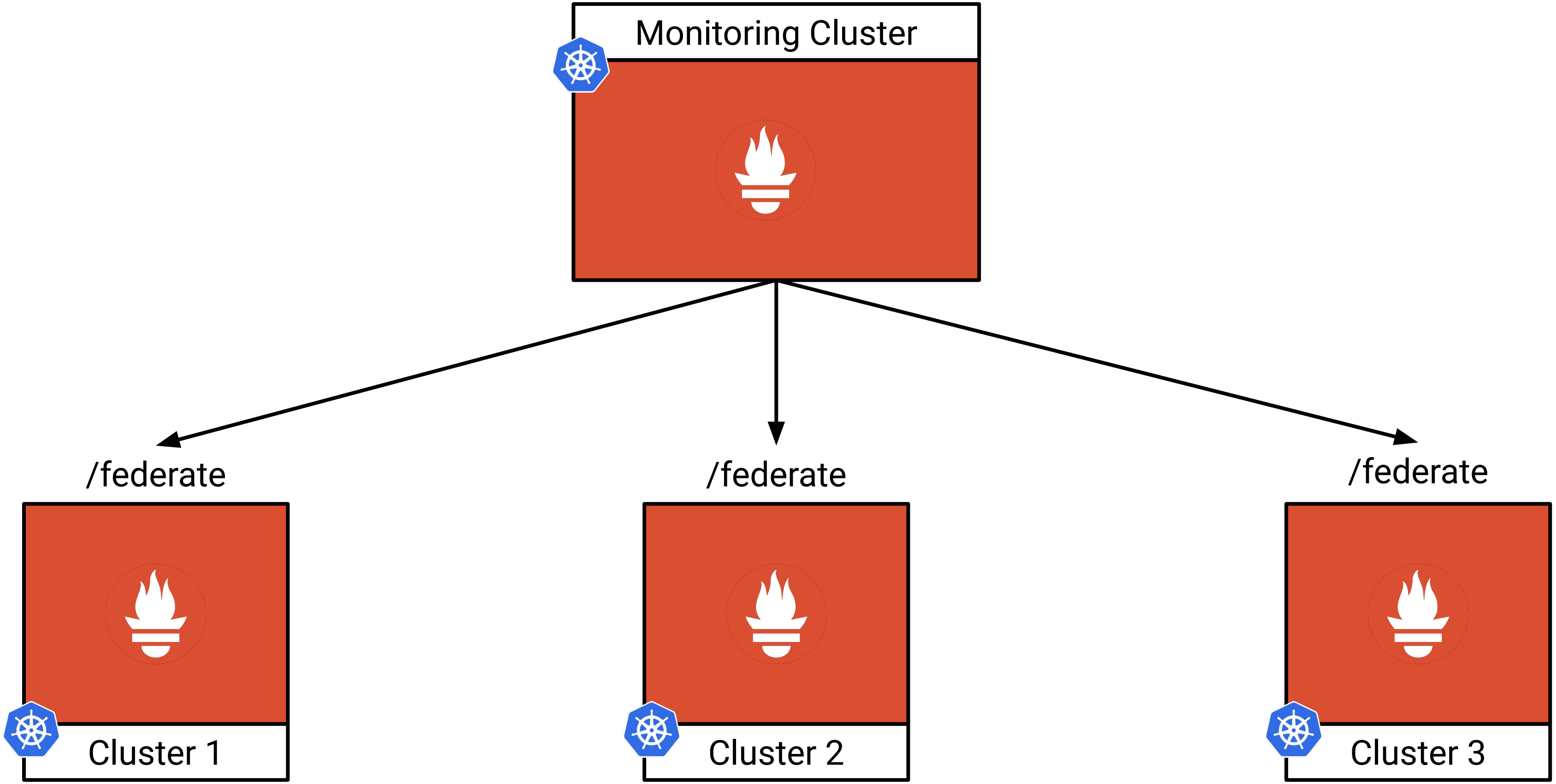


Prometheus Limitations

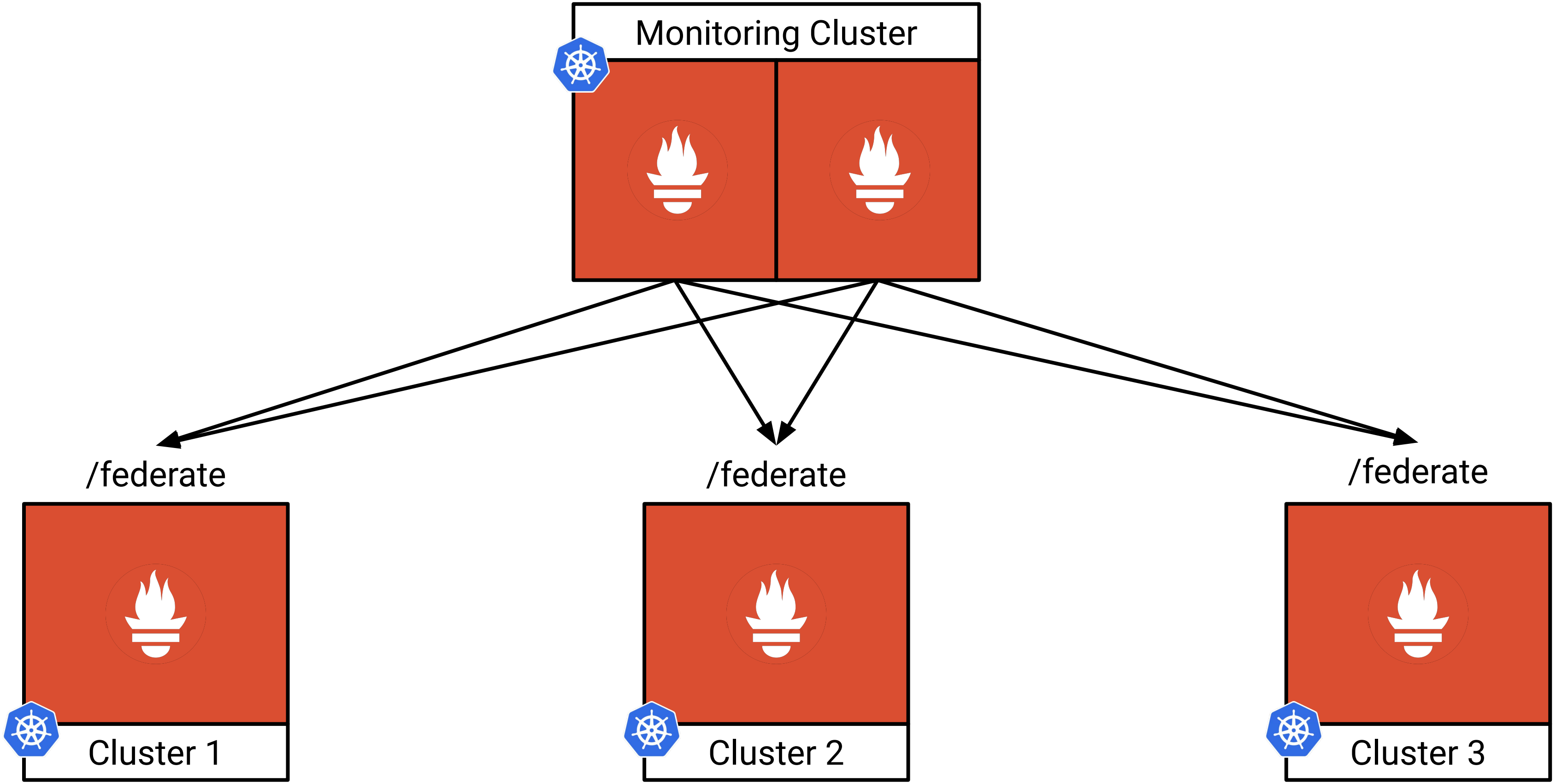
@ThanosMetrics



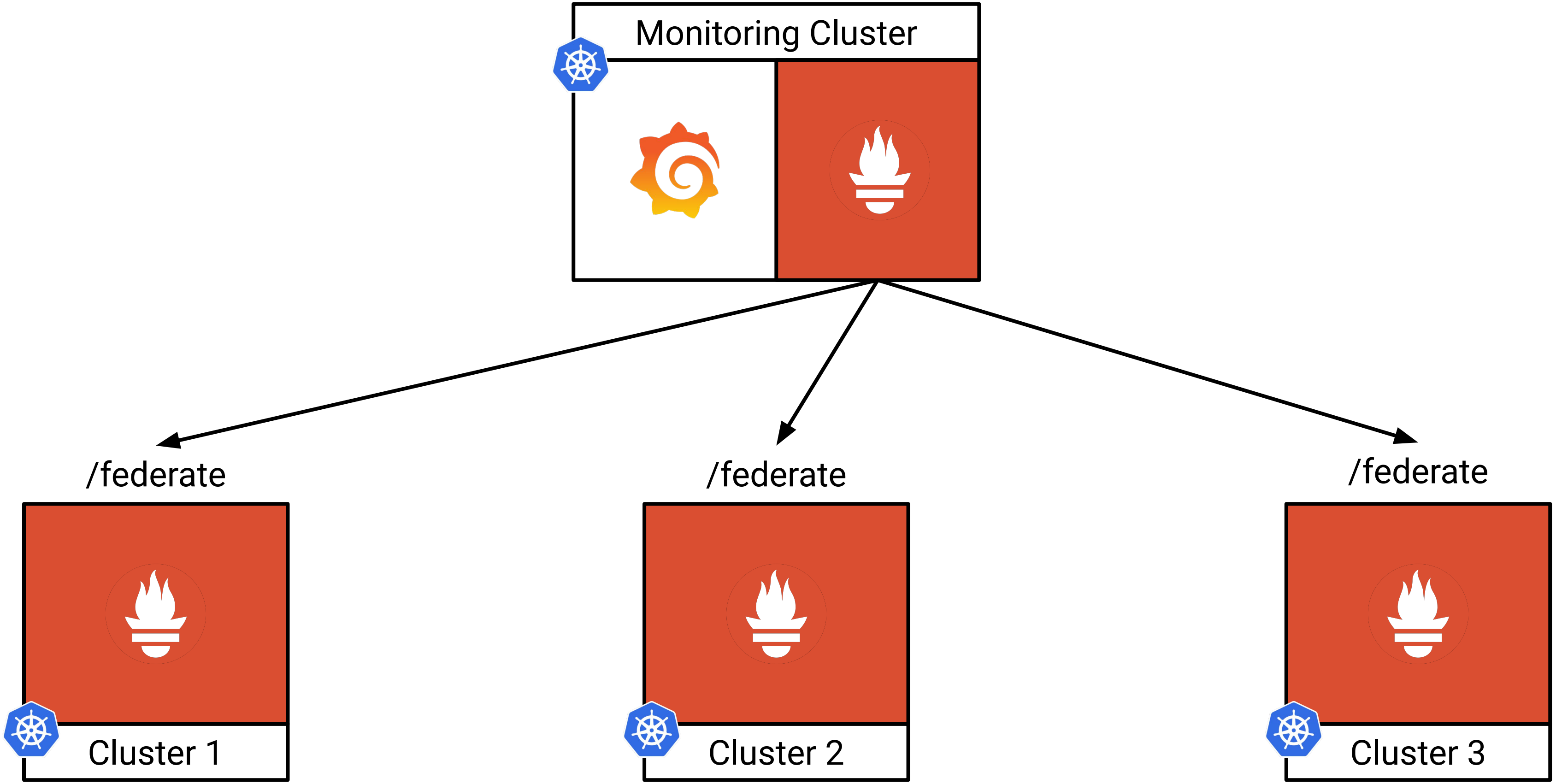
Prometheus Limitations



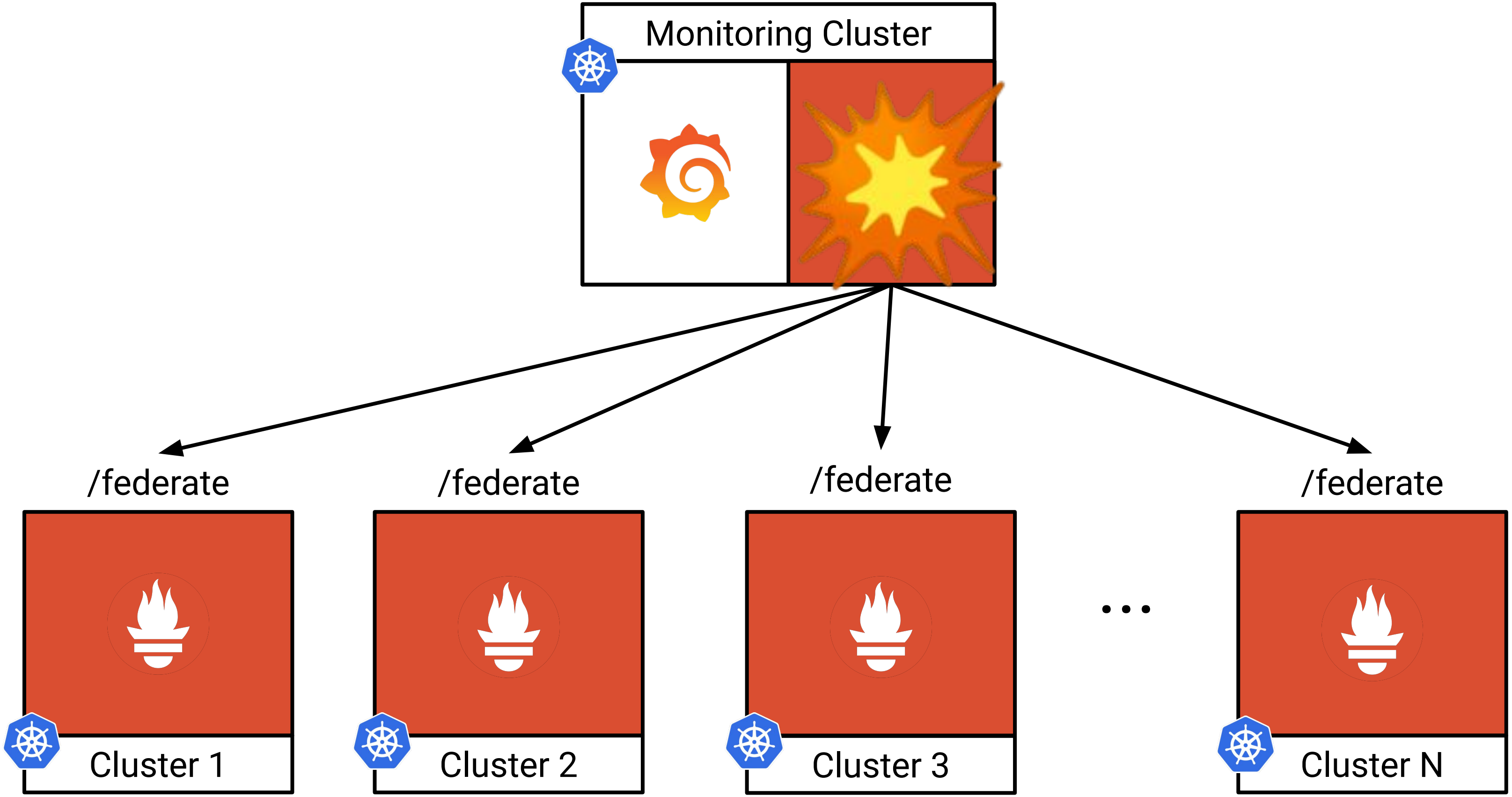
Prometheus Limitations



Prometheus Limitations



Prometheus Limitations



Monitoring Cluster

/federate

/federate

/federate

/federate

Cluster 1

Cluster 2

Cluster 3

...

Cluster N

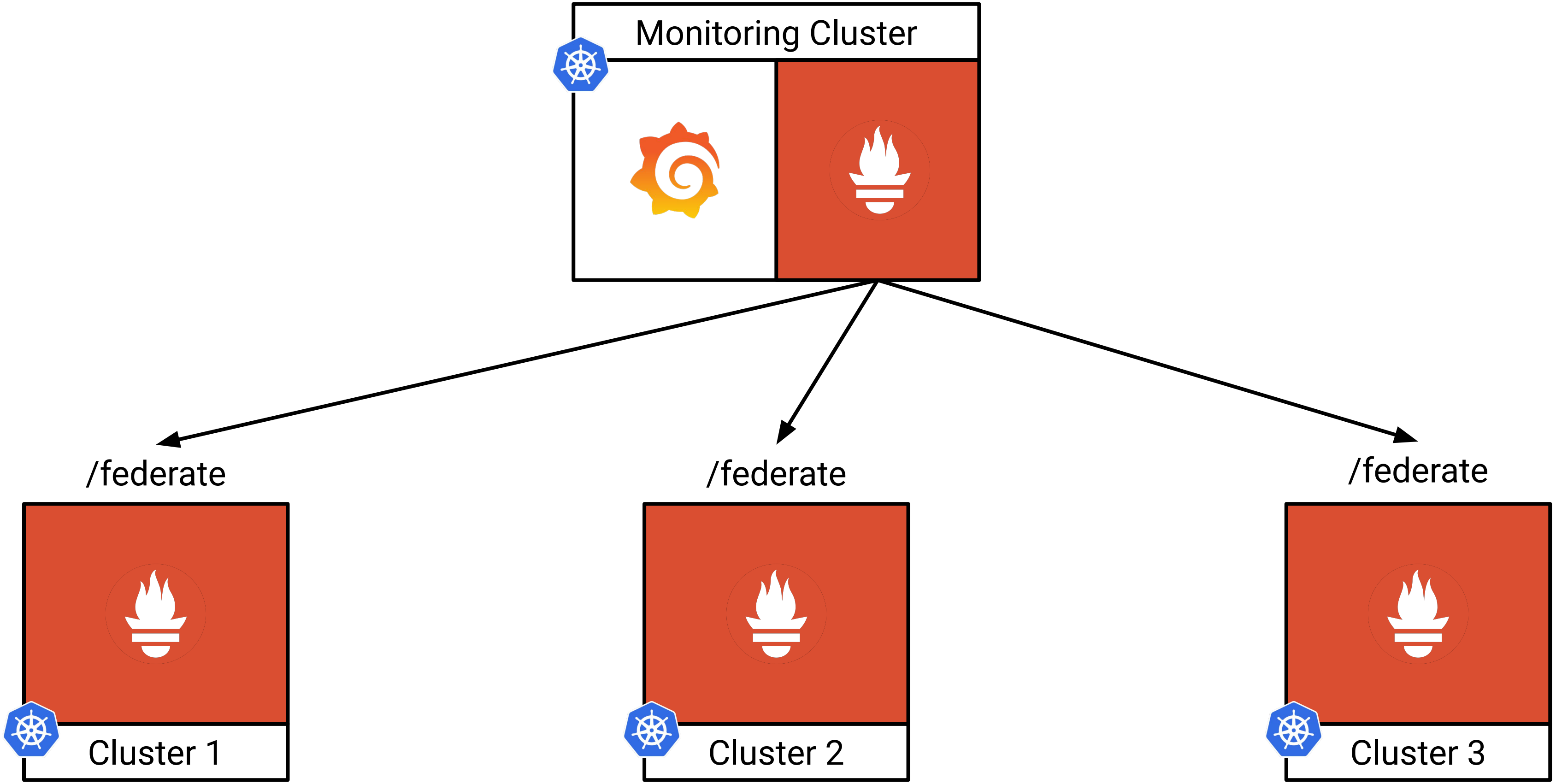
@ThanosMetrics

Thanos

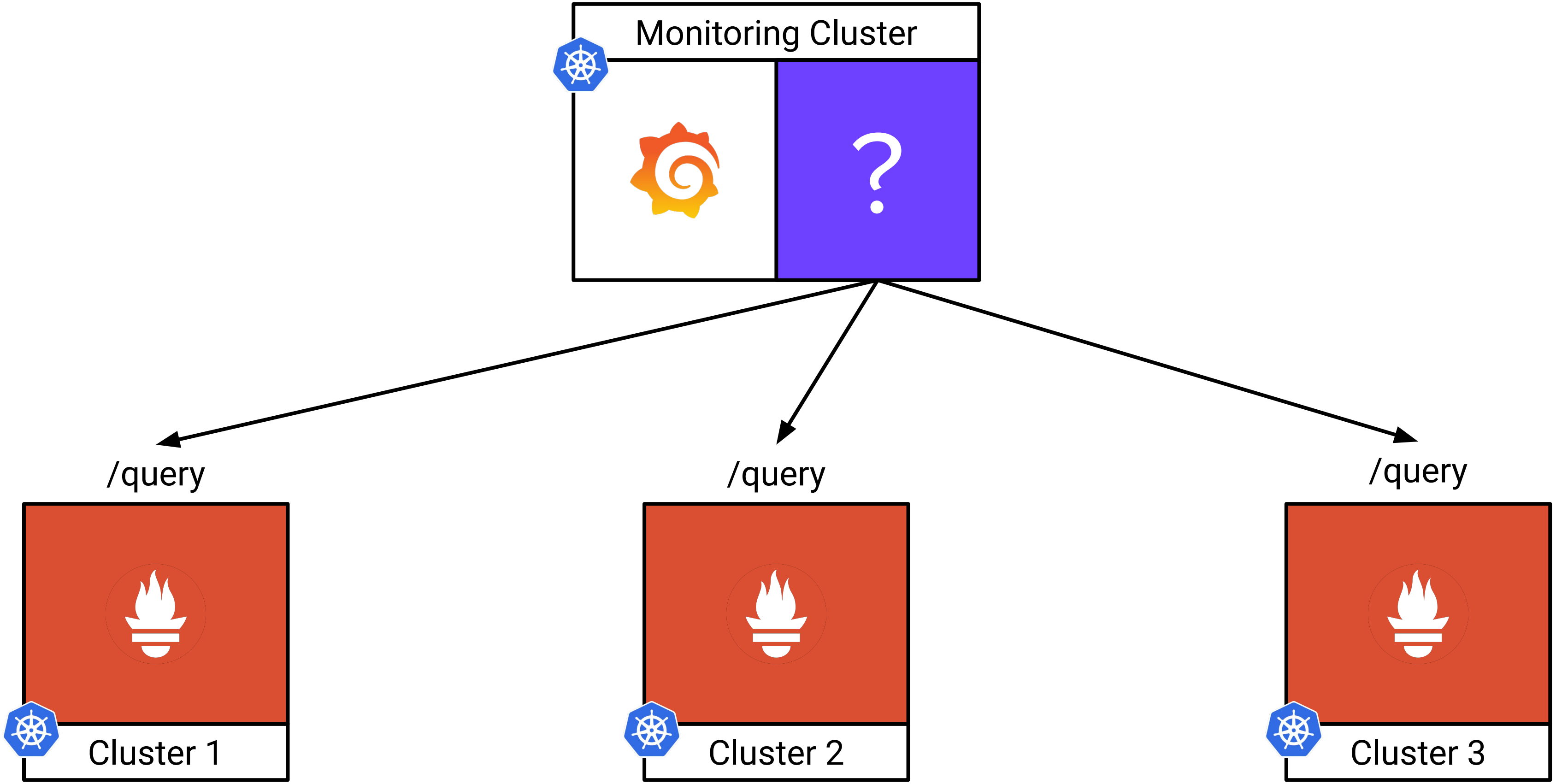
Project Aims

- Prometheus Compatible
- Global Query View
- Unlimited Retention
- Downsampling & Compaction

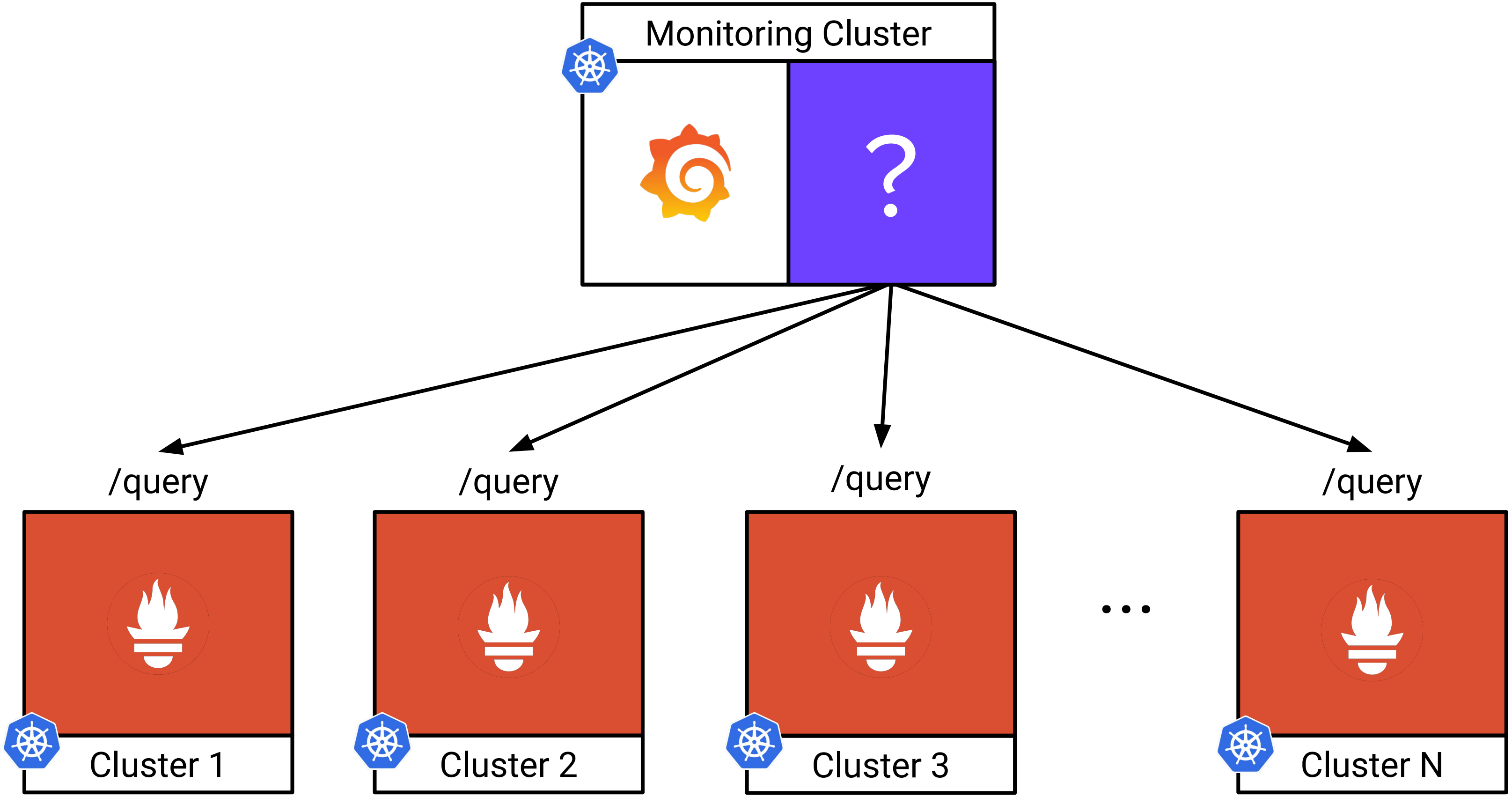
Scalability



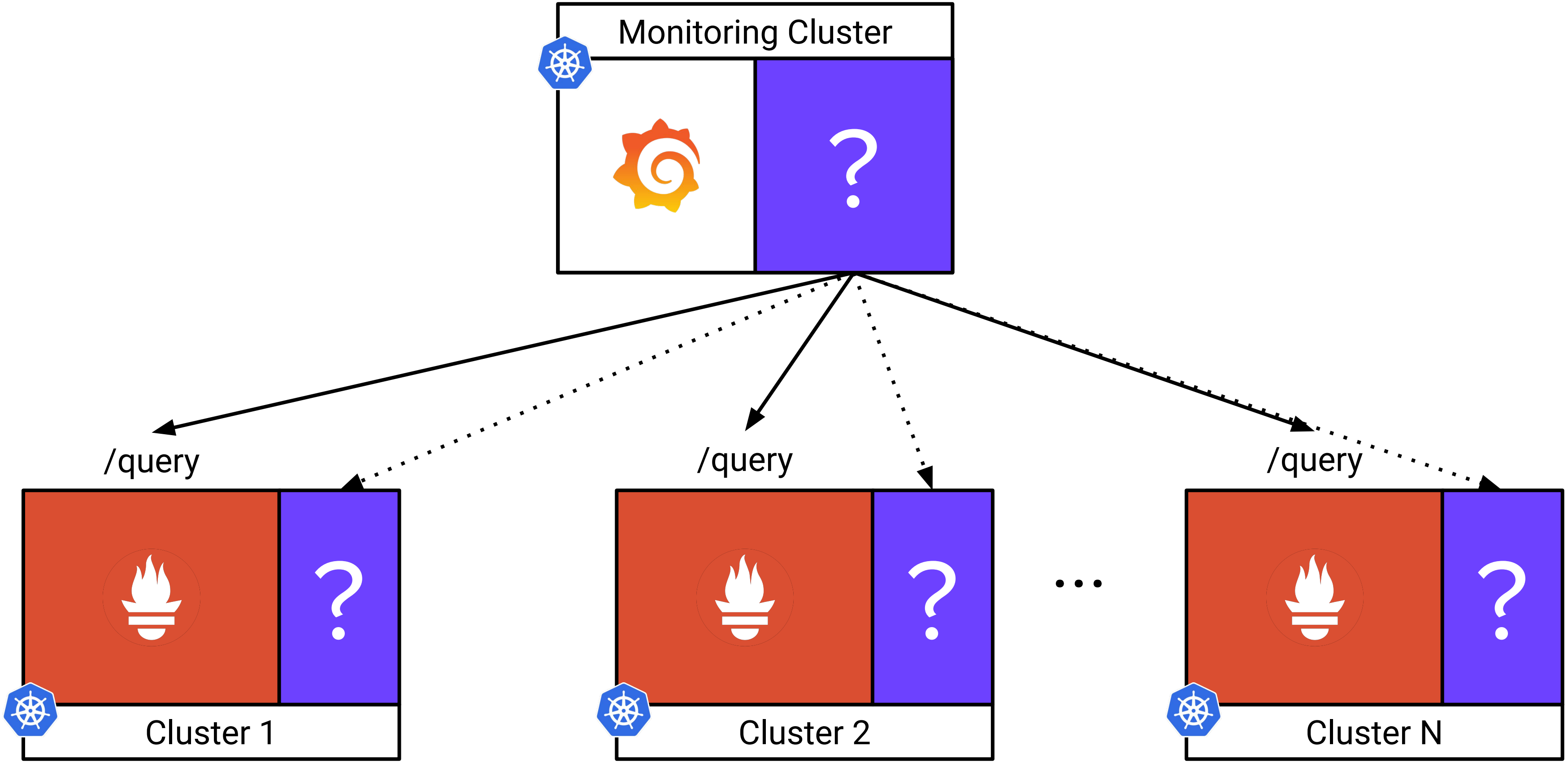
Scalability



Scalability

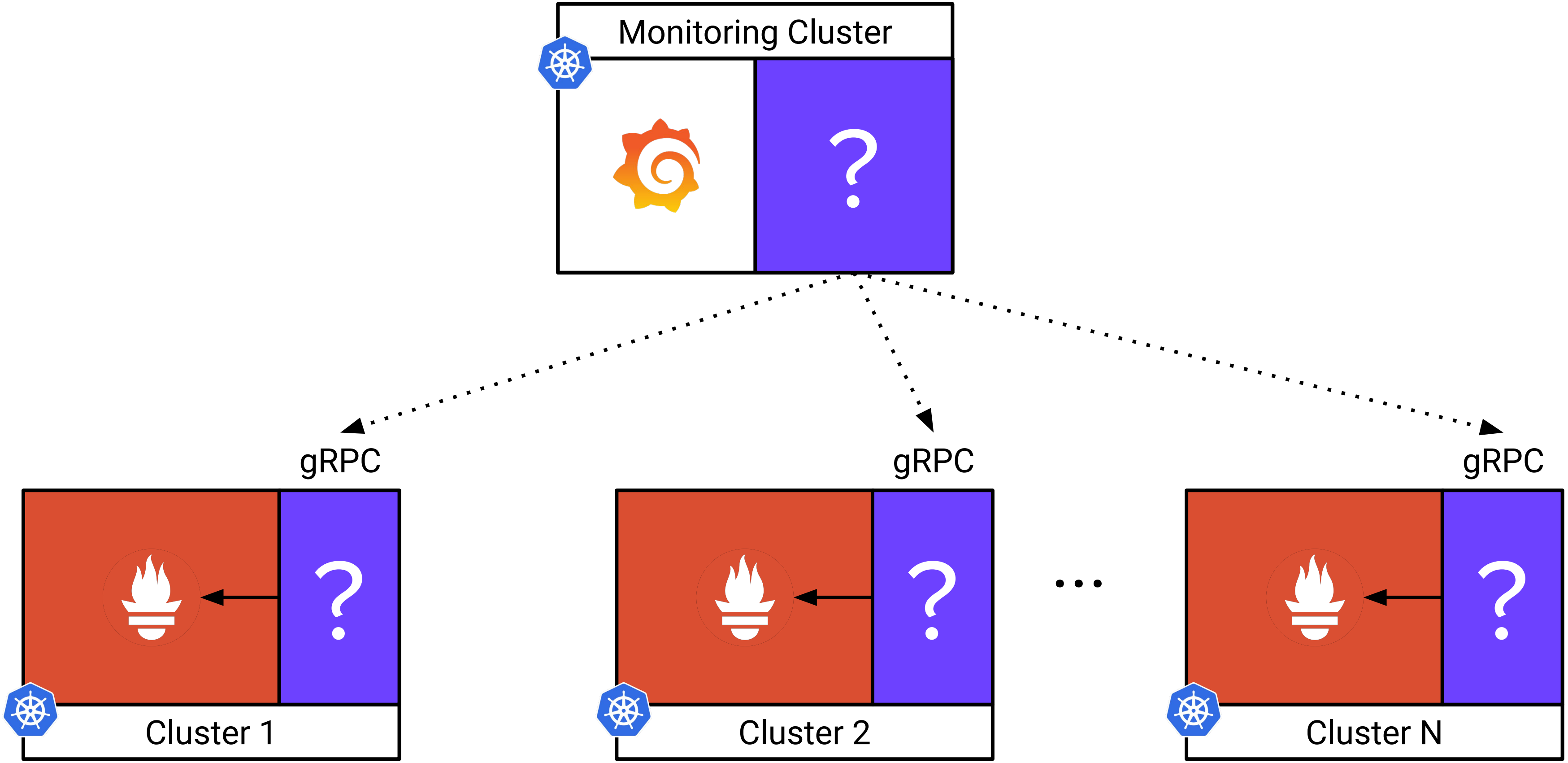


Scalability



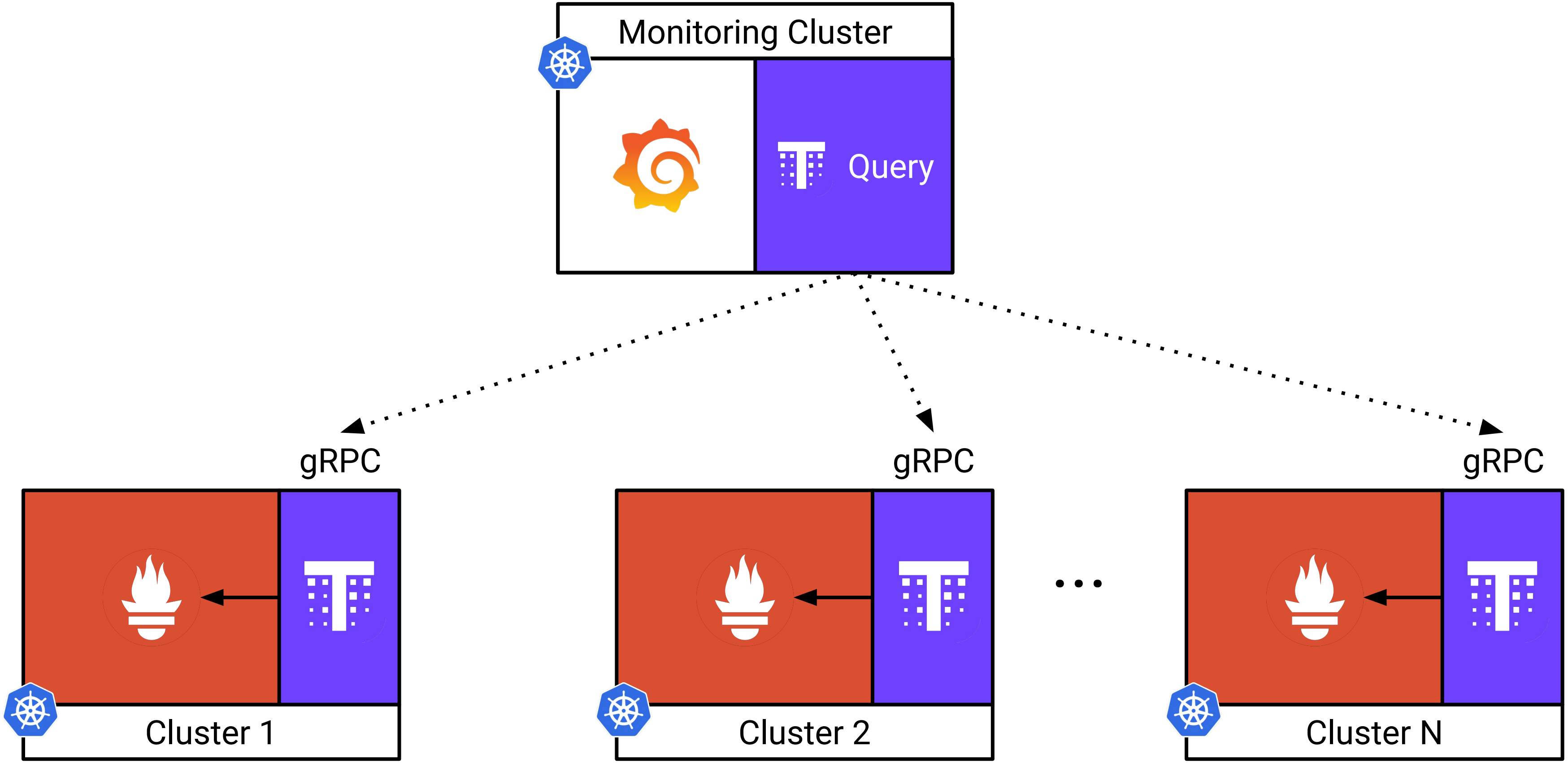
Scalability

@ThanosMetrics



Scalability

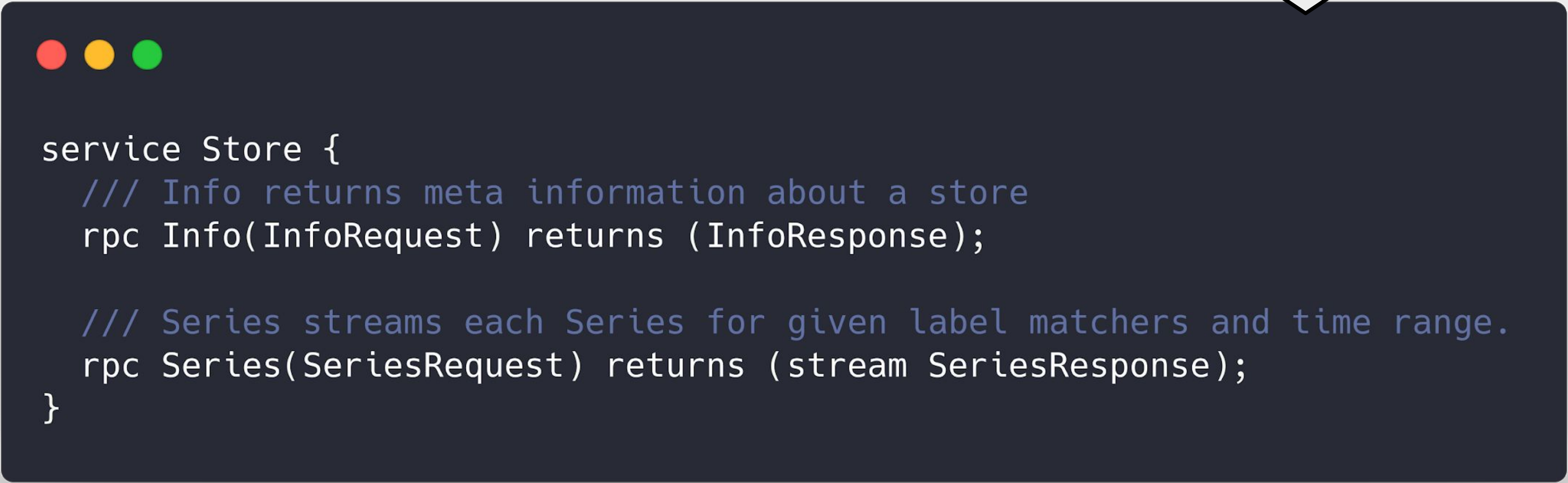
@ThanosMetrics



Store API

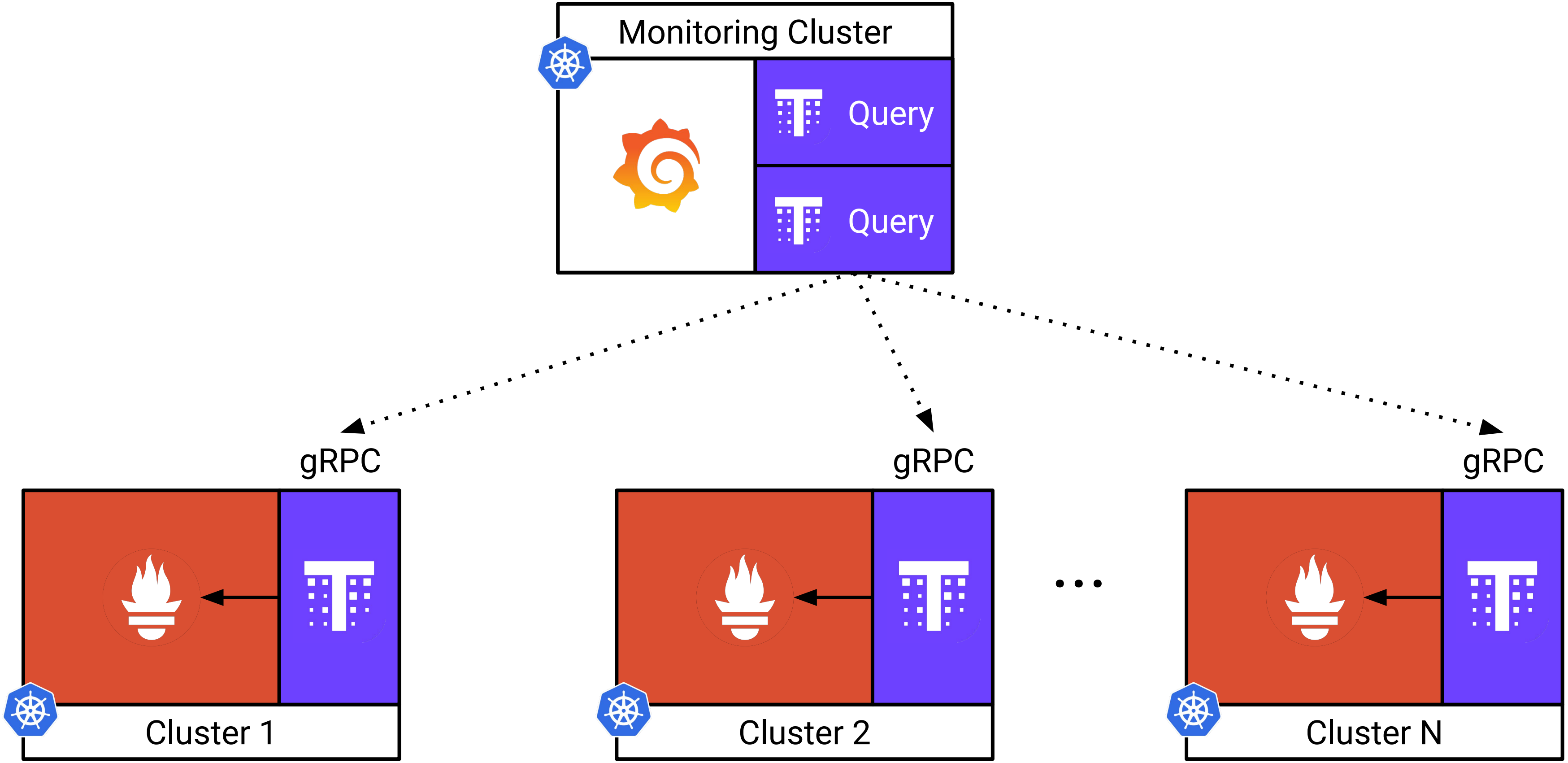
Thanos Deep Dive: Inside a Distributed Monitoring System

Tomorrow (5:20)



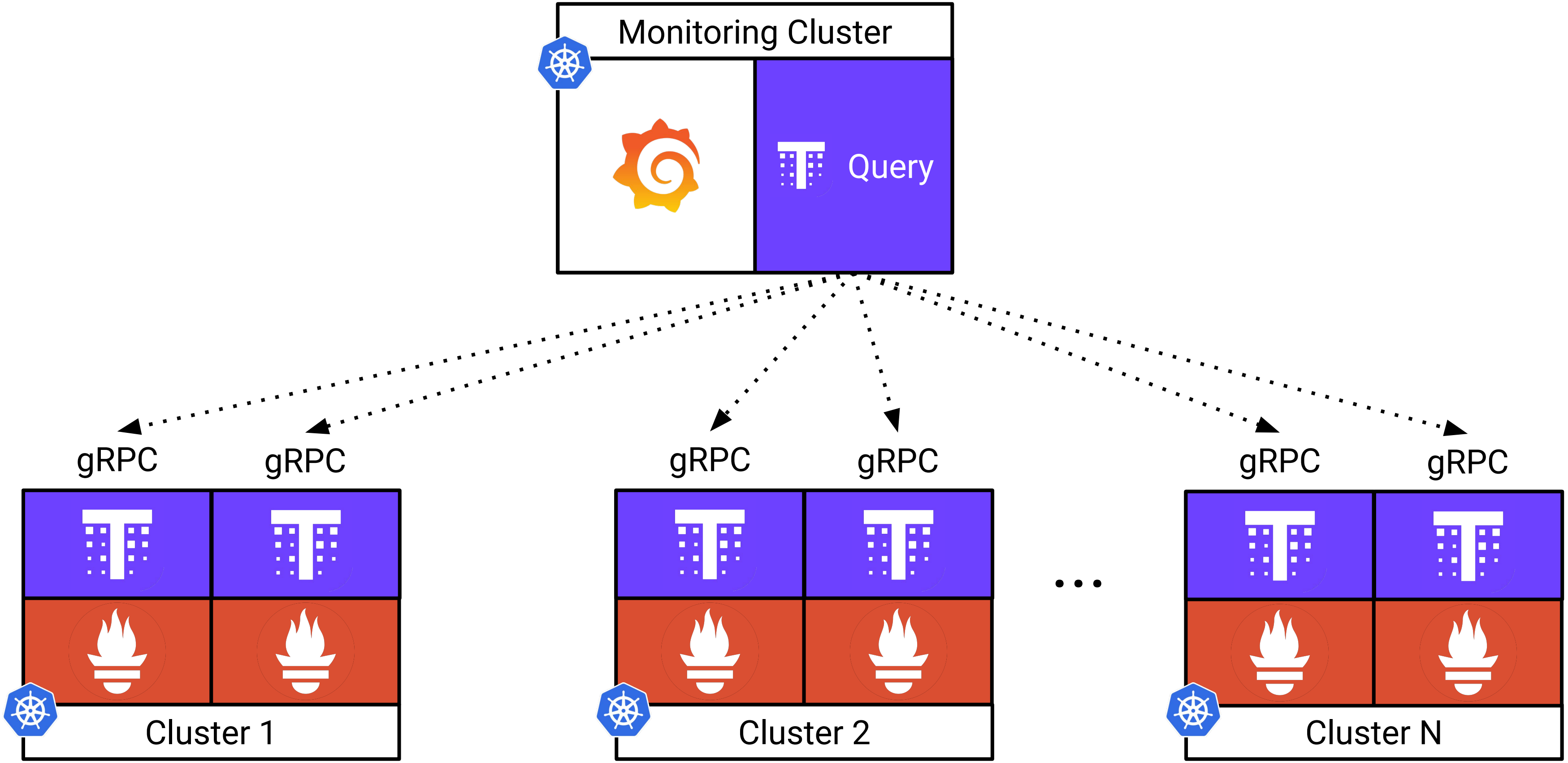
```
service Store {
  /// Info returns meta information about a store
  rpc Info(InfoRequest) returns (InfoResponse);

  /// Series streams each Series for given label matchers and time range.
  rpc Series(SeriesRequest) returns (stream SeriesResponse);
}
```

High Availability

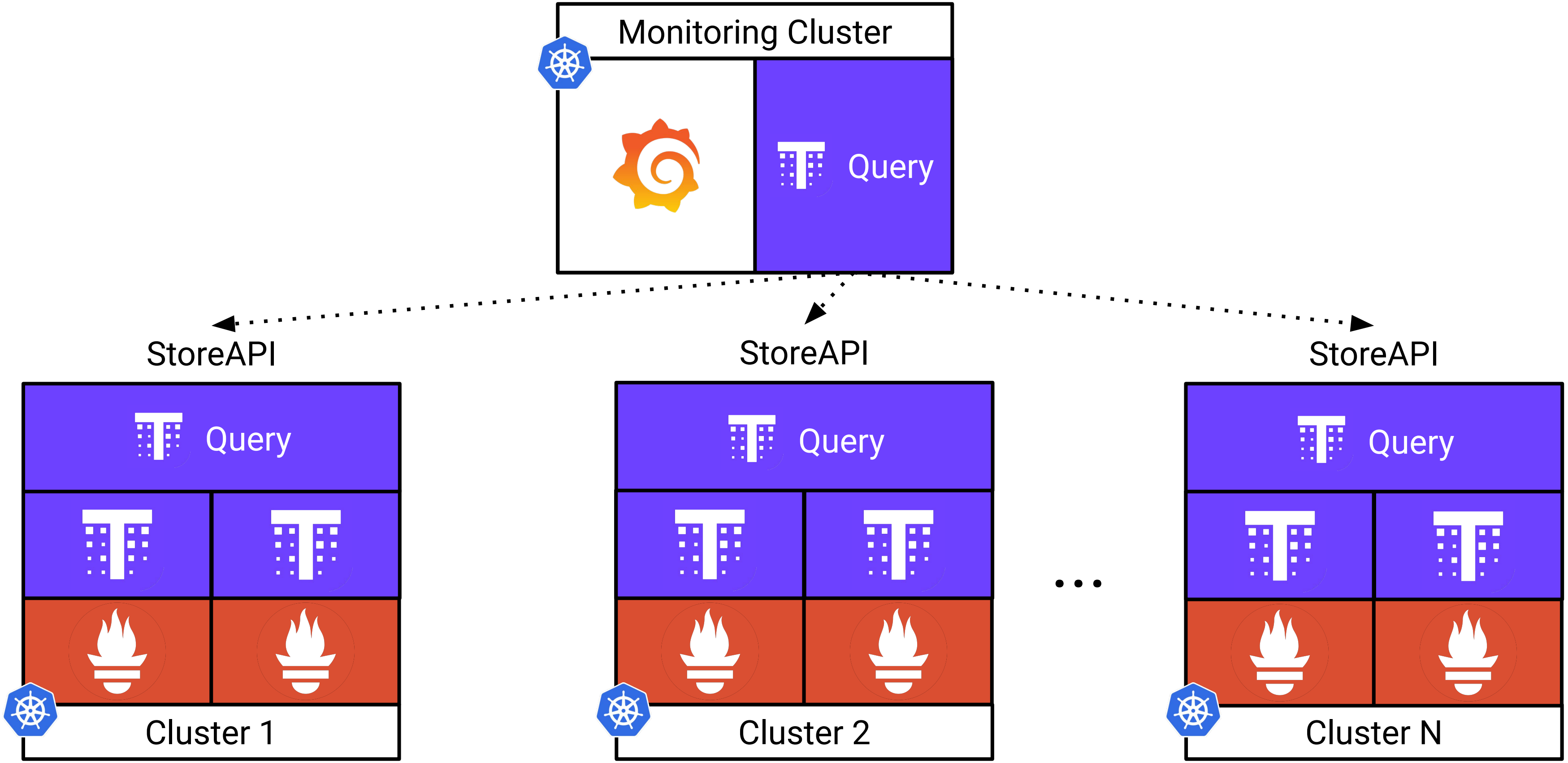


High Availability

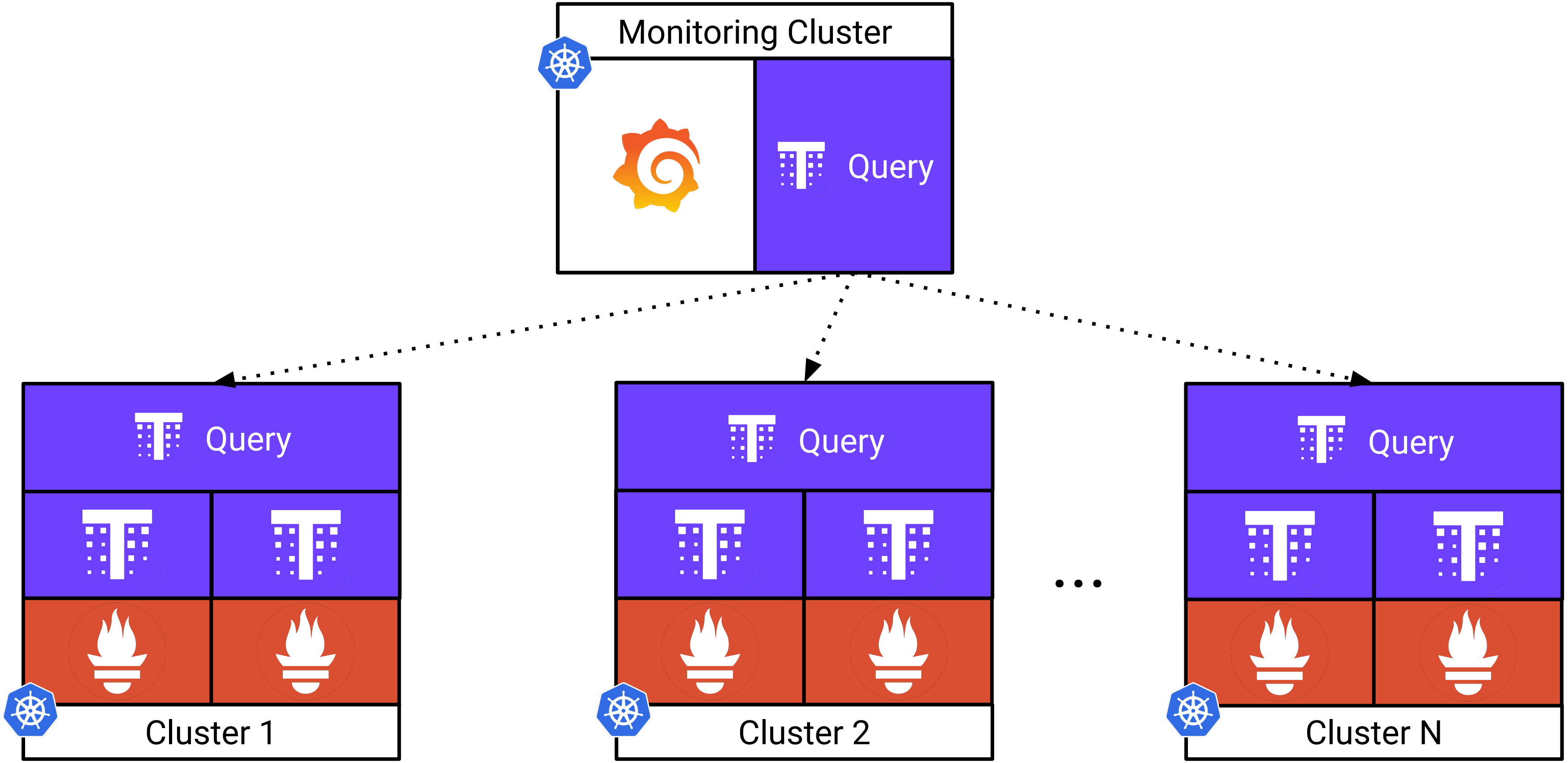
@ThanosMetrics



High Availability

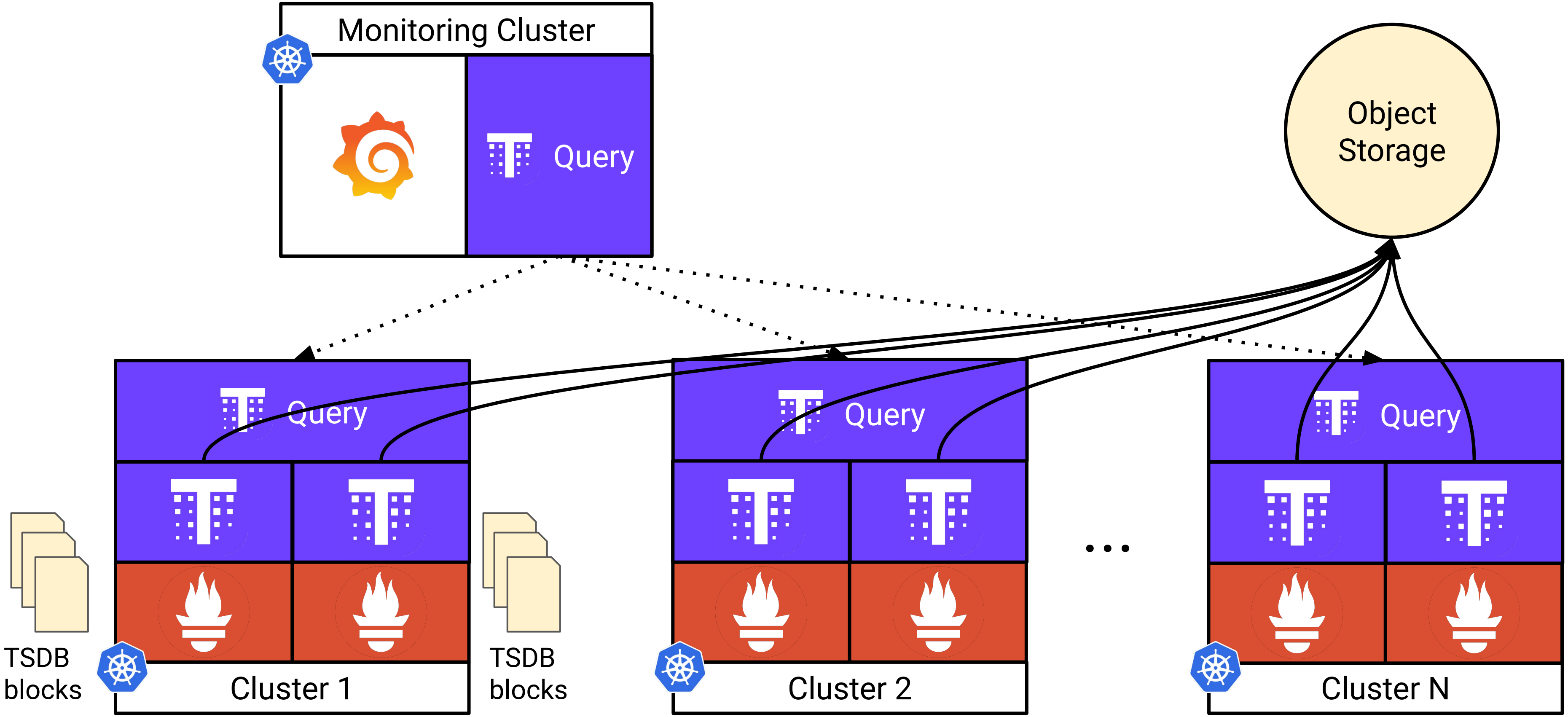


Improved Storage



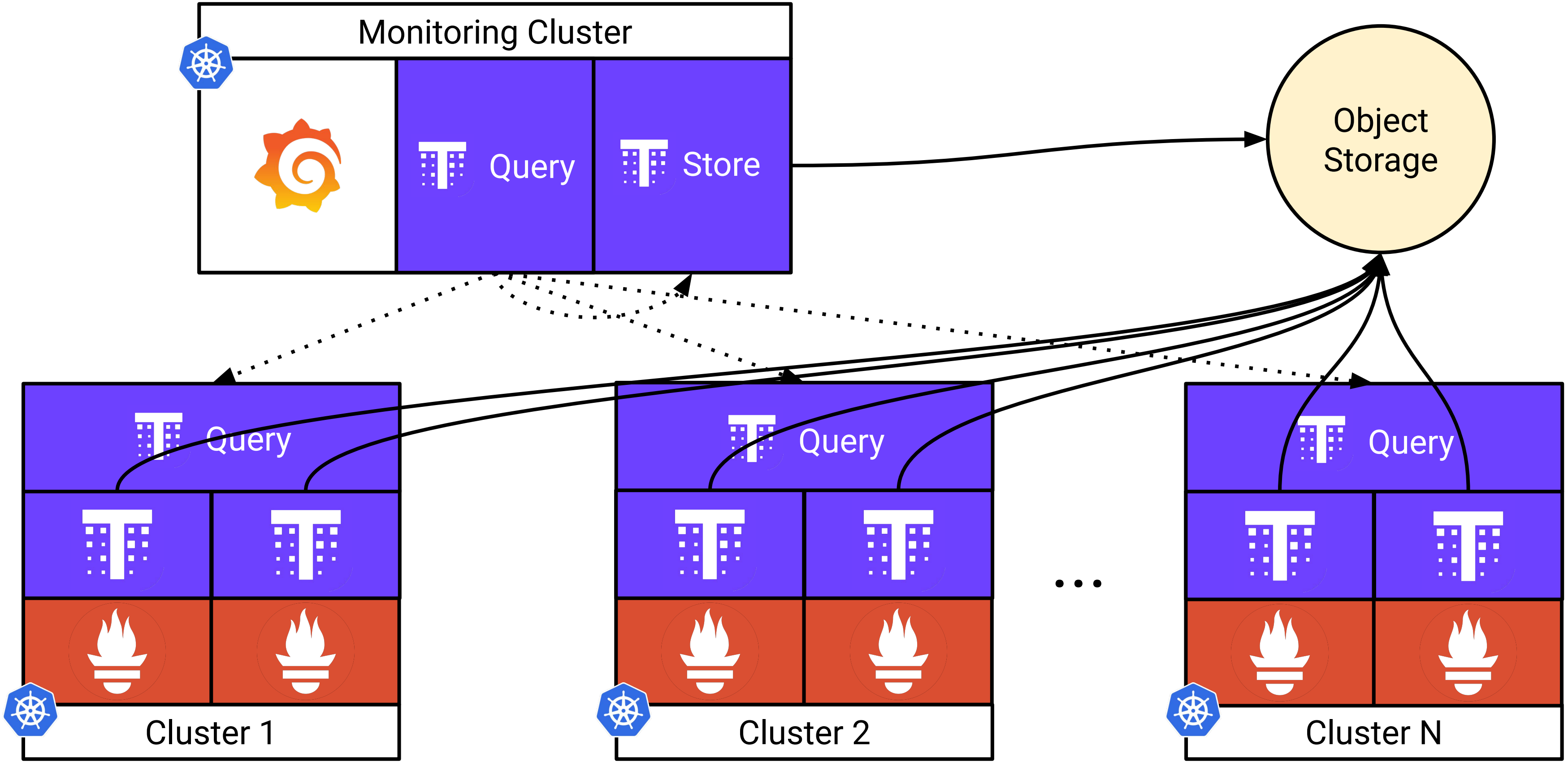
TSDB Uploaded to Object Storage

@ThanosMetrics



Store Gateway

@ThanosMetrics

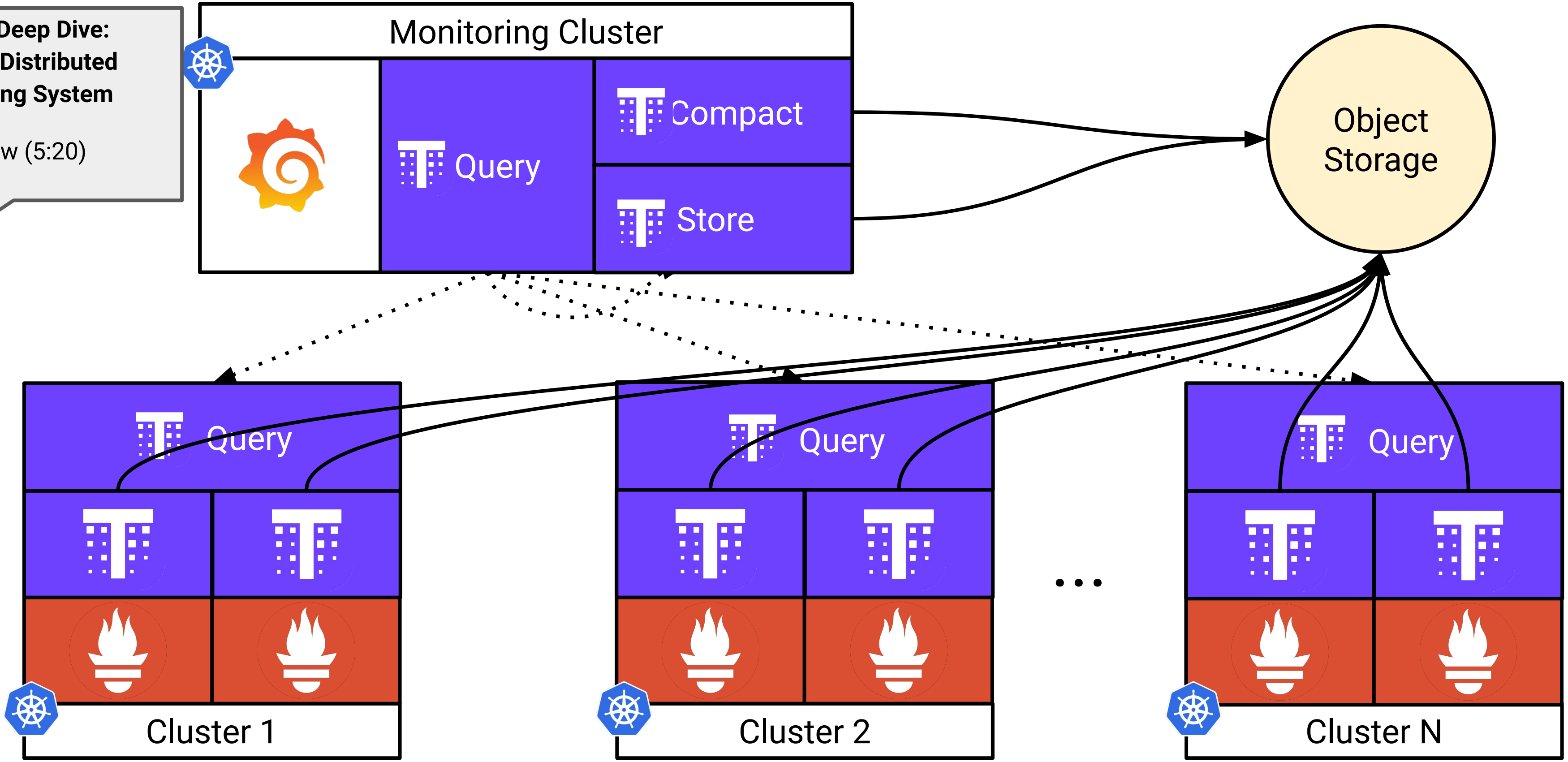


Compactor

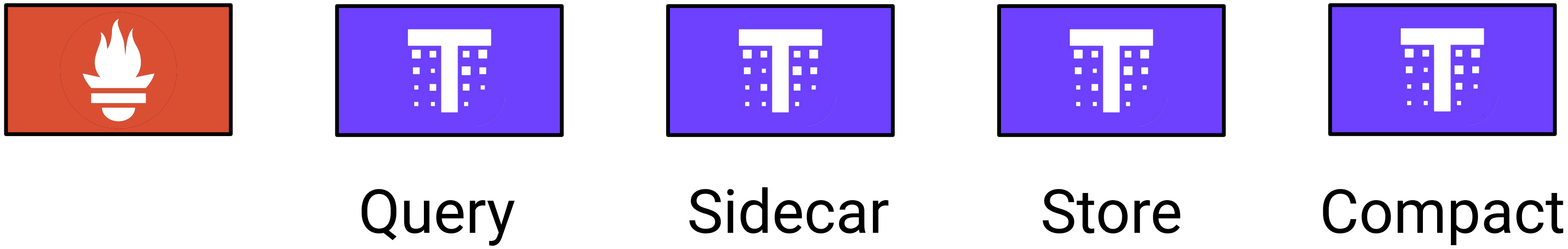
@ThanosMetrics

Thanos Deep Dive:
Inside a Distributed
Monitoring System

Tomorrow (5:20)



Thanos, Prometheus At Scale



Query

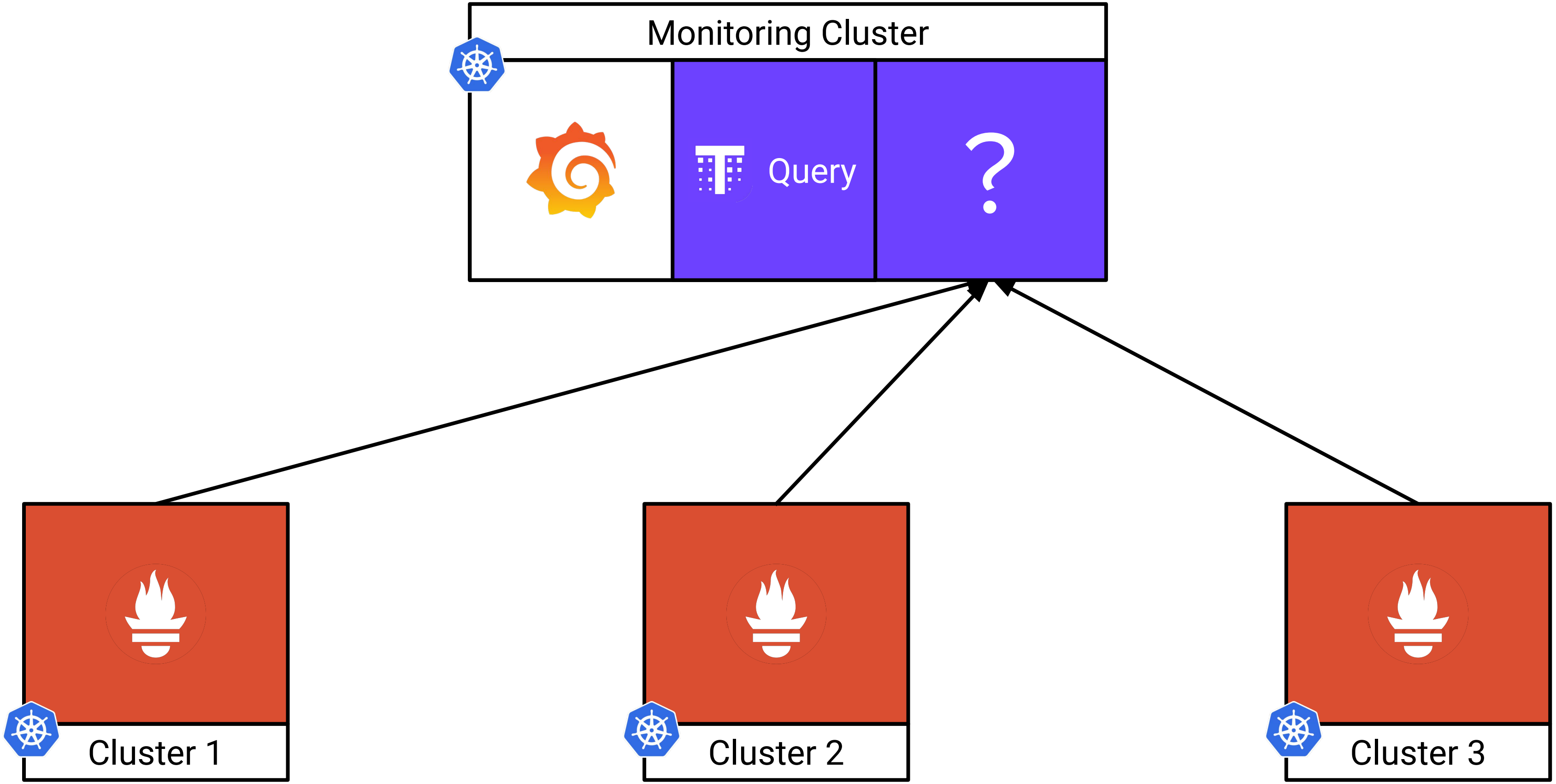
Sidecar

Store

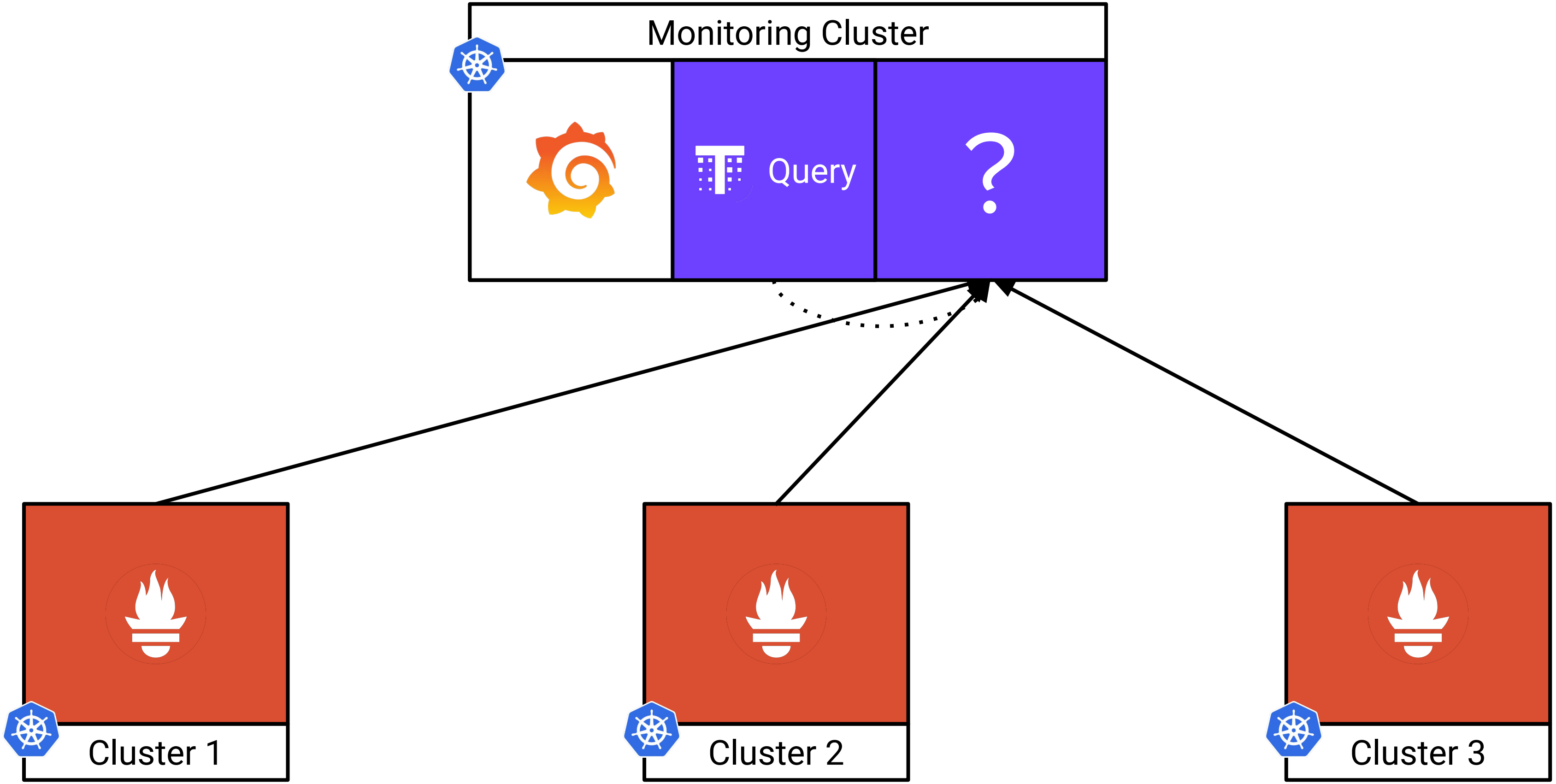
Compact

Scalability

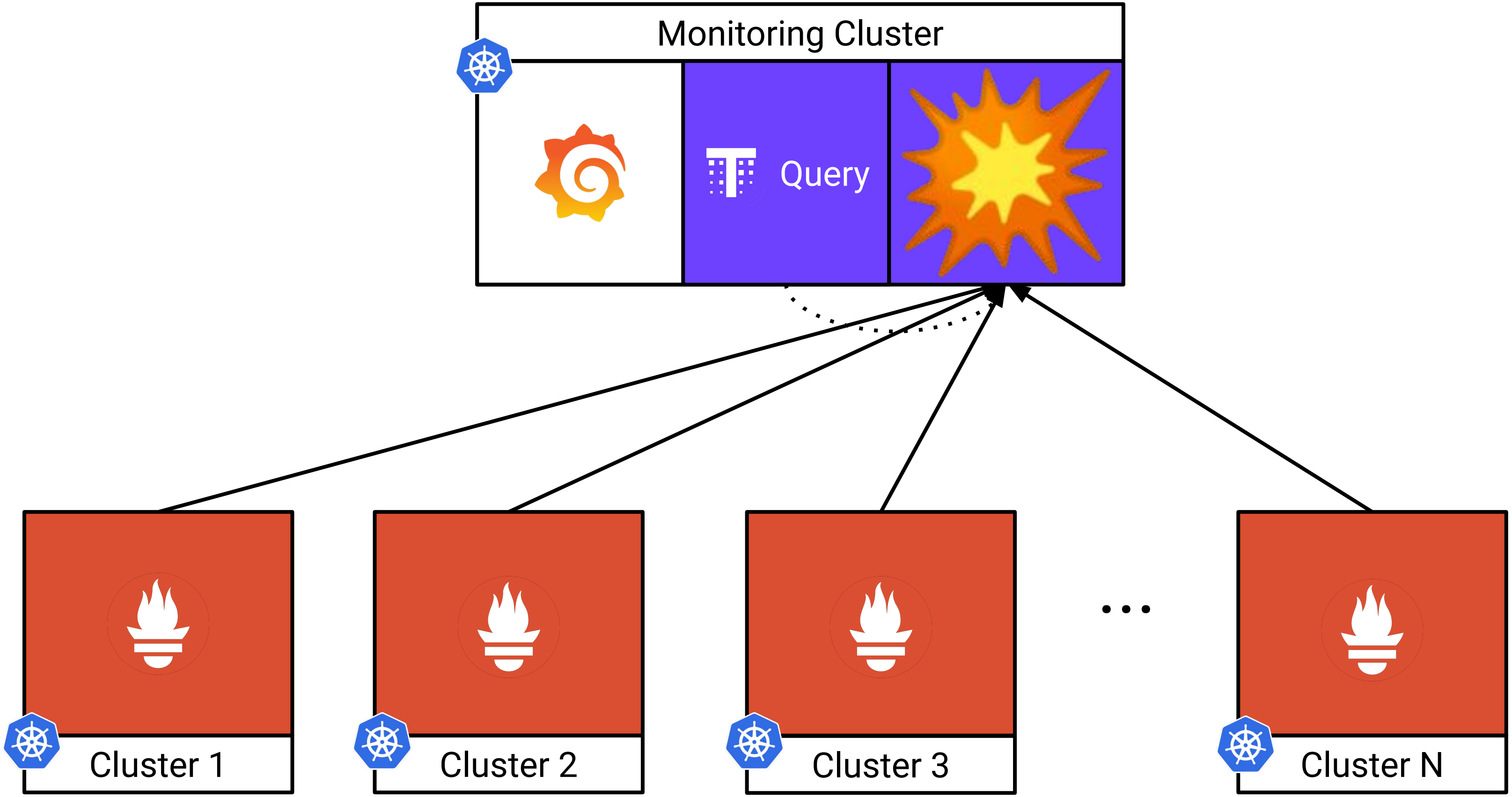
@ThanosMetrics



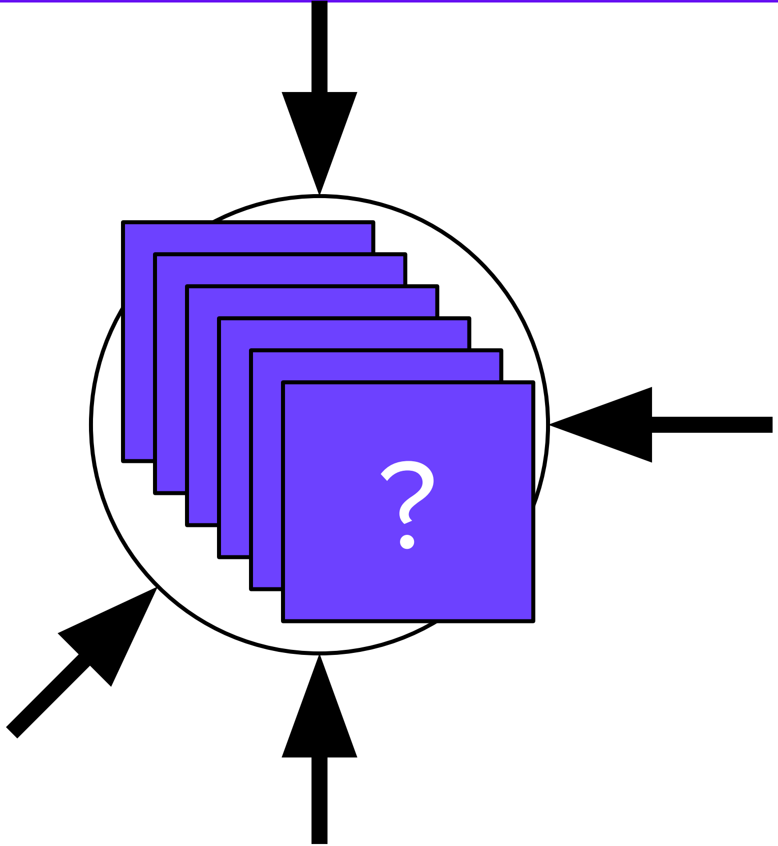
Scalability



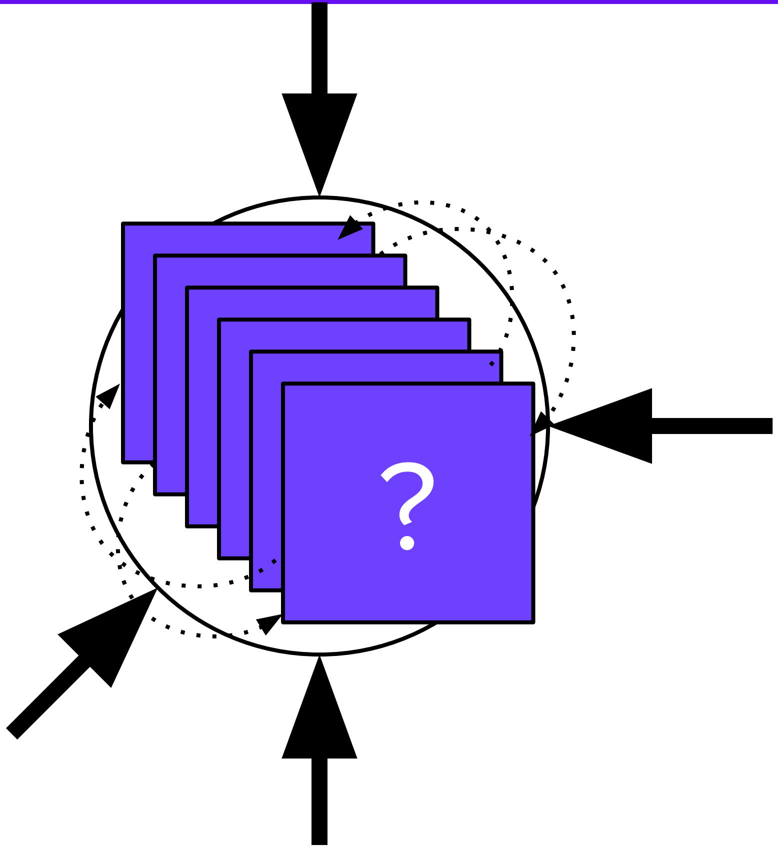
Scalability



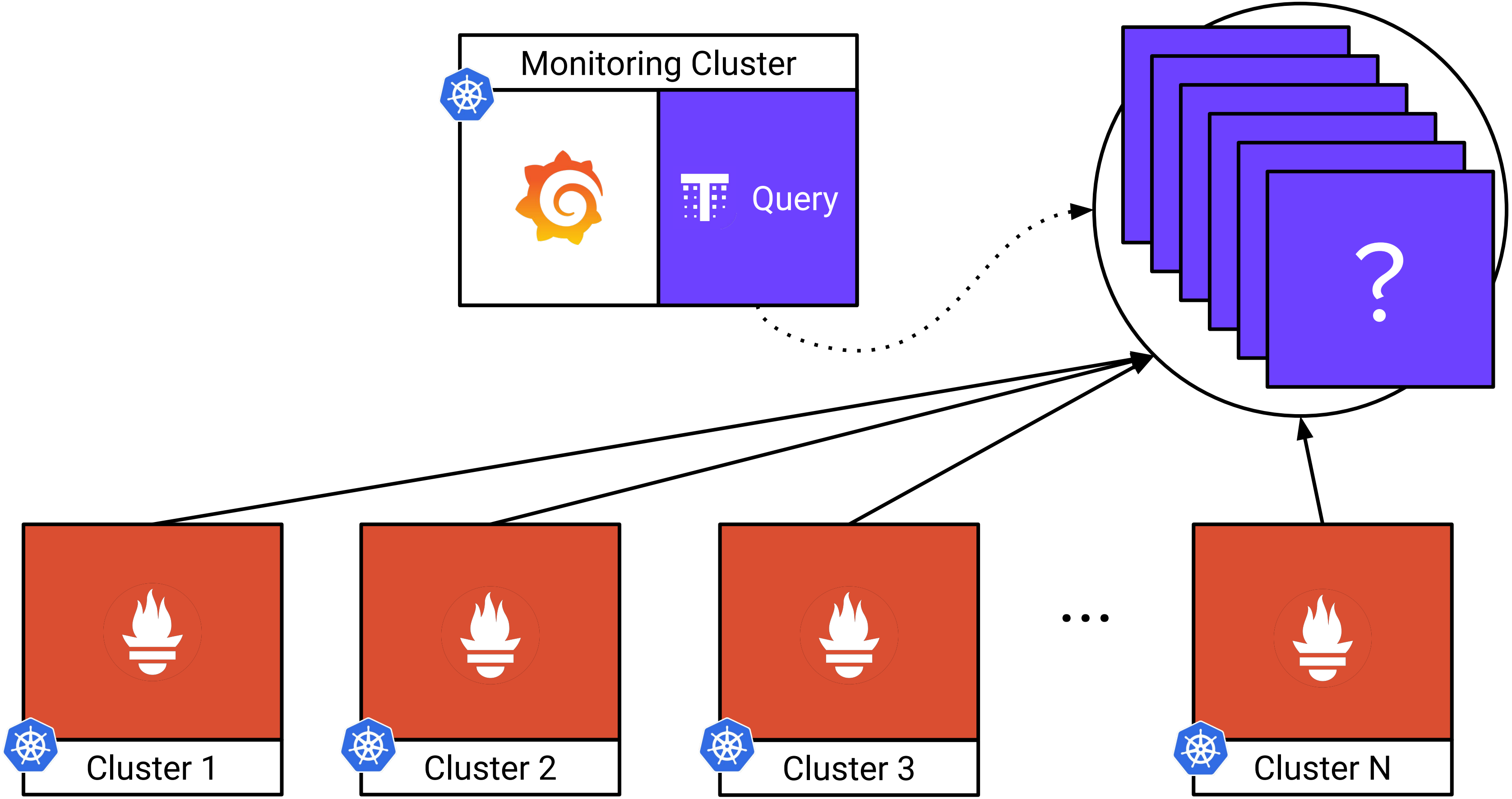
Hash Ring



Hash Ring

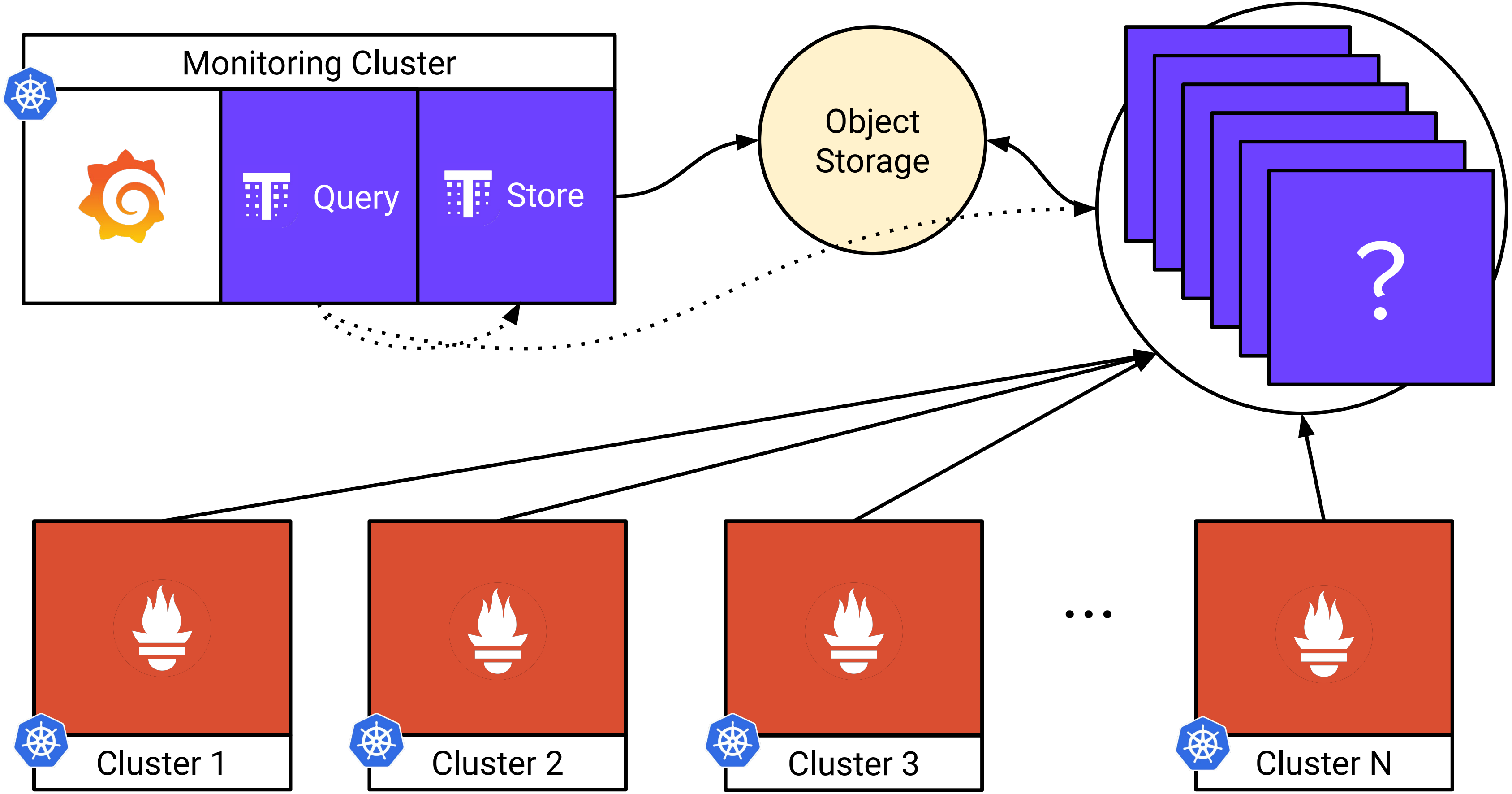


Scalability



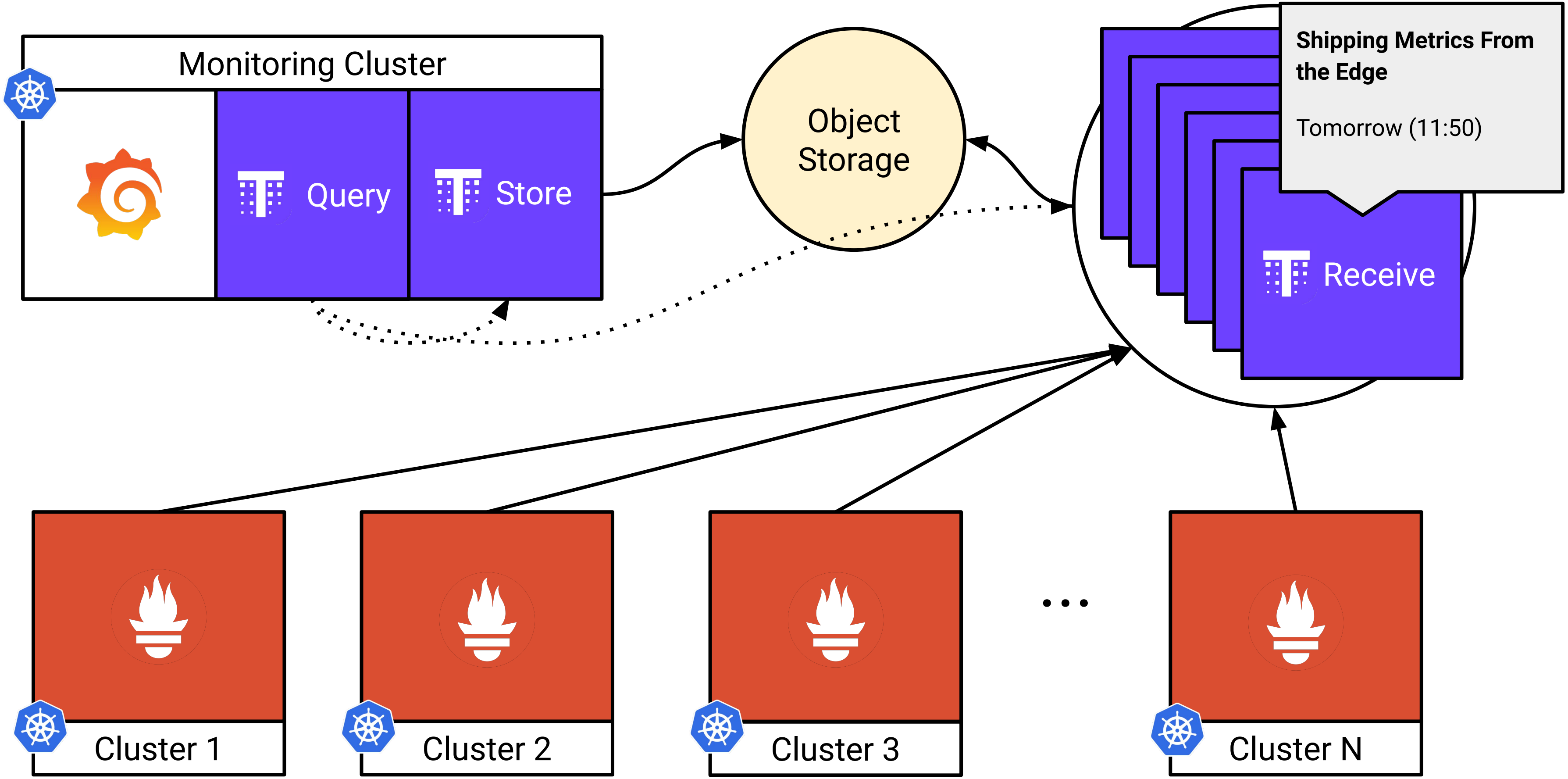
Scalability

@ThanosMetrics



Scalability

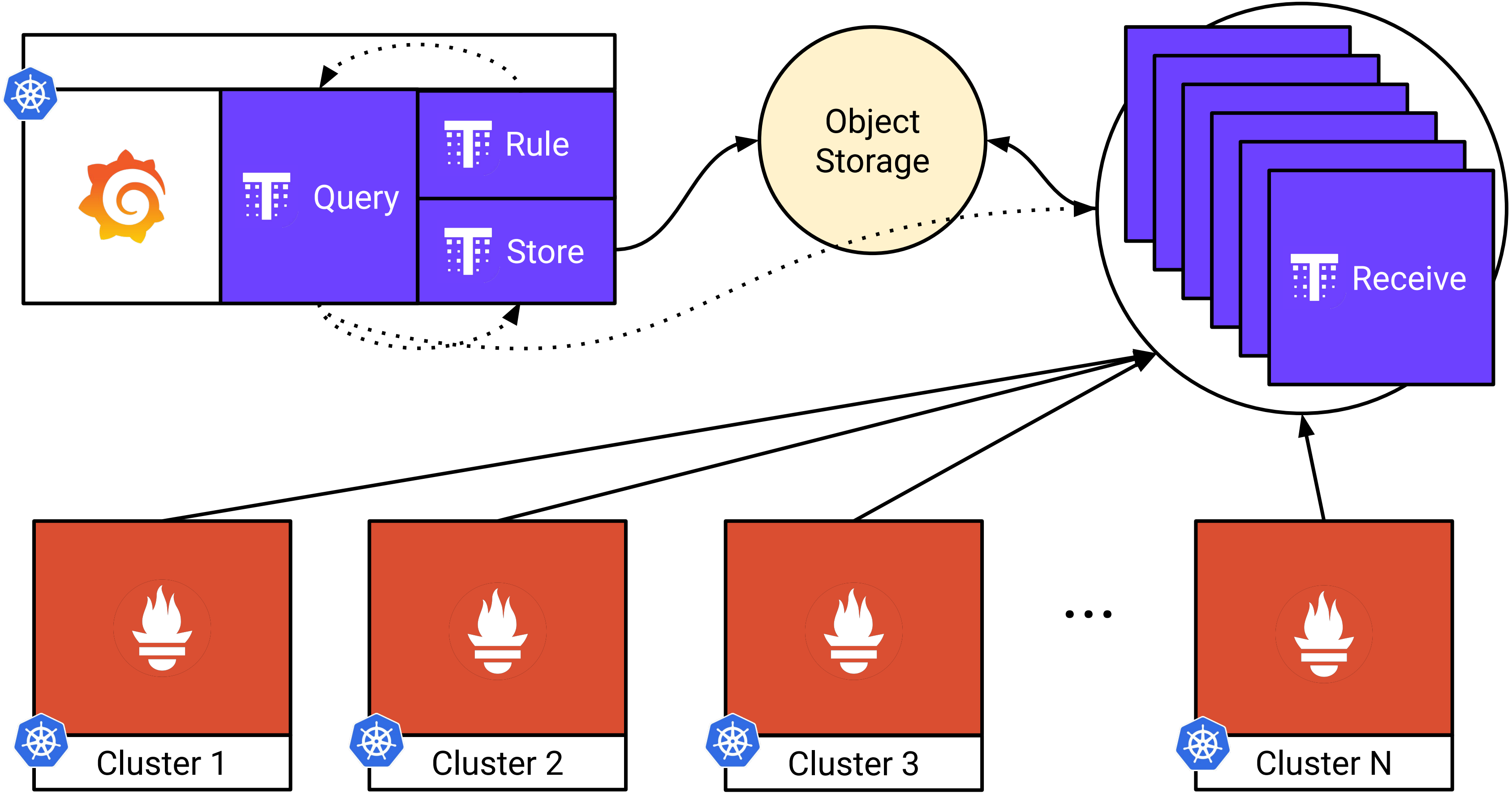
@ThanosMetrics



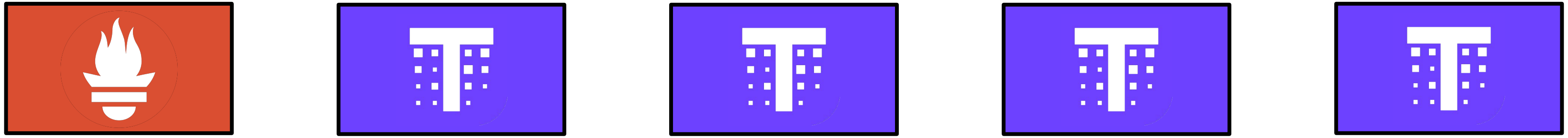
Shipping Metrics From the Edge

Tomorrow (11:50)

Recording & Alerting Rules



Thanos, Prometheus At Scale



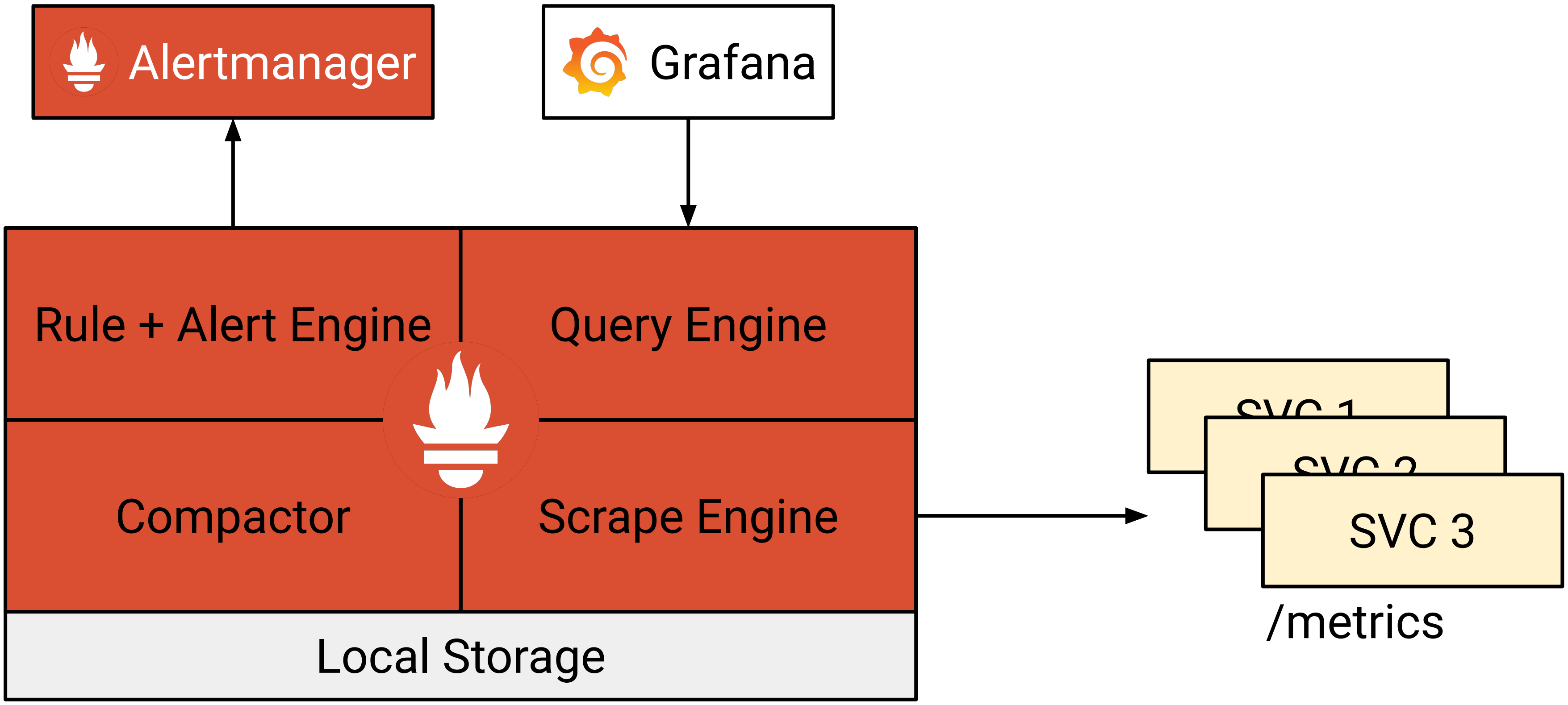
Query

Receive

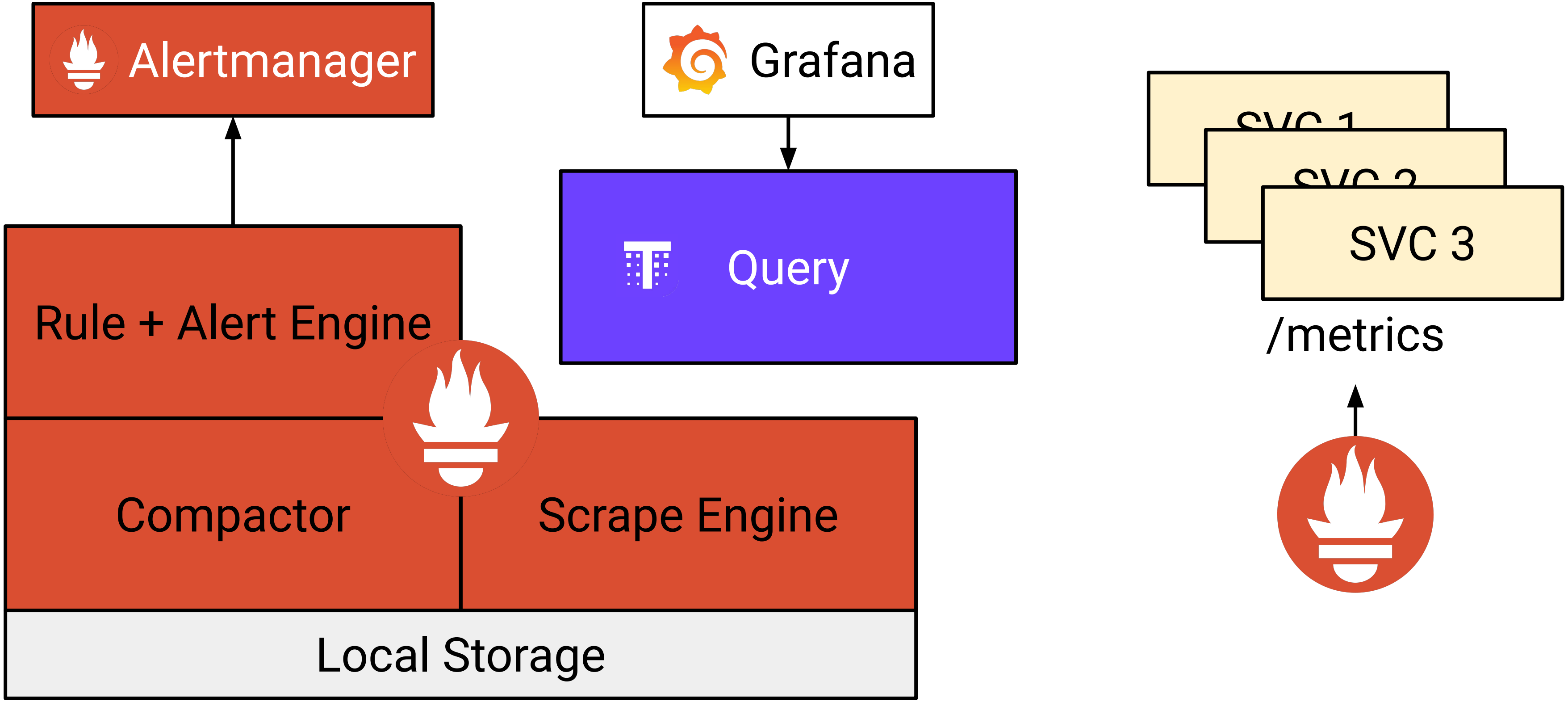
Store

Rule

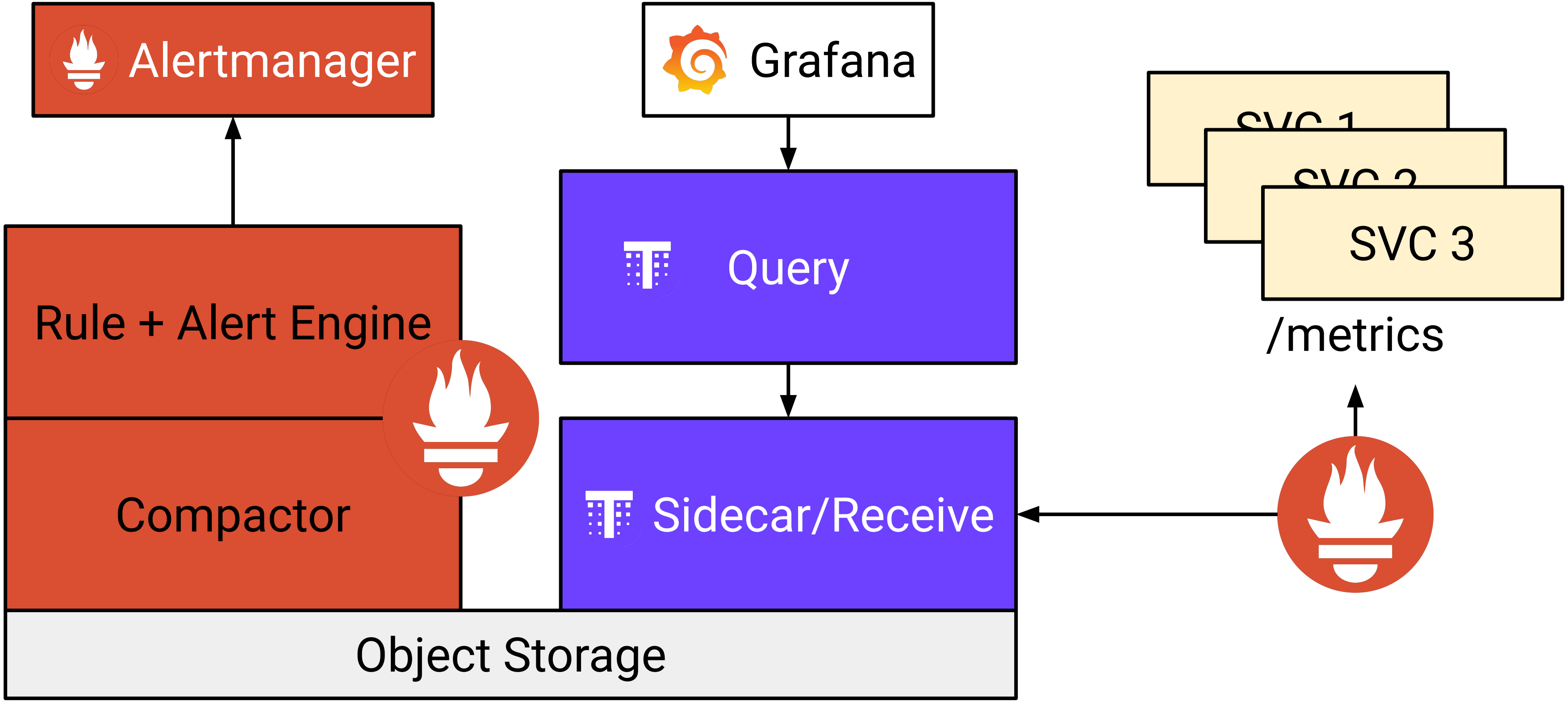
Distributed Prometheus



Distributed Prometheus

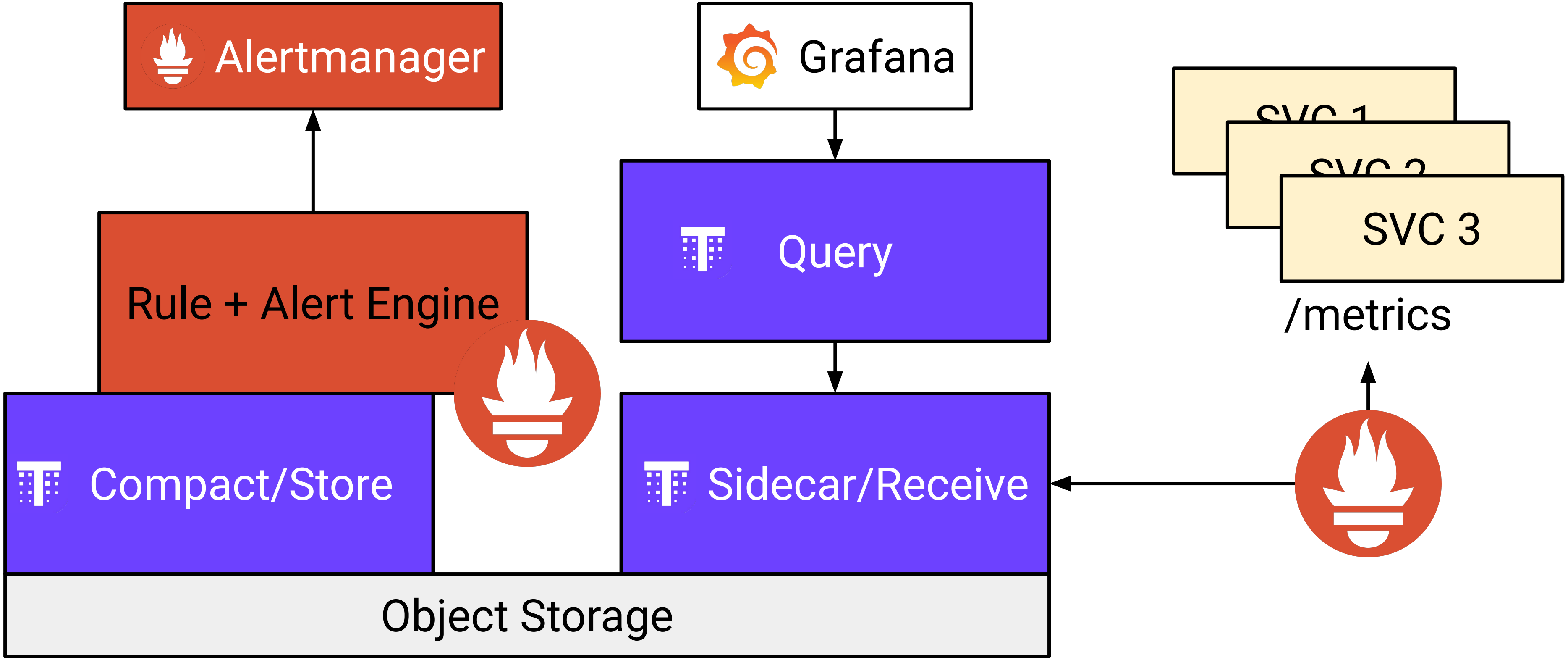


Distributed Prometheus

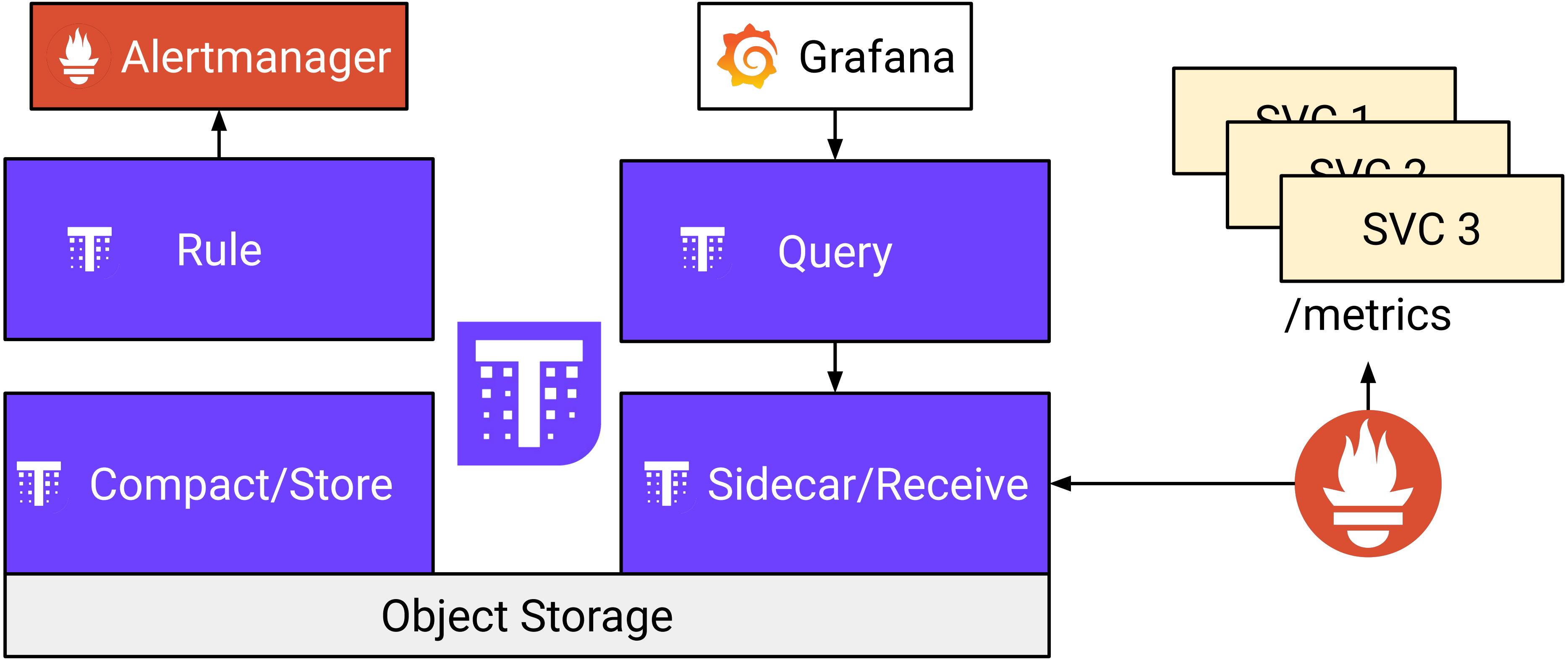


Distributed Prometheus

@ThanosMetrics



Distributed Prometheus



https://www.katacoda.com/thanos

Thank You!

https://thanos.io

KubeCon + CloudNativeCon

@ThanosMetrics

Shipping Metrics From the Edge

- Matthias Loibl, Red Hat
- Wednesday November 20, 2019
- 11:50am (Room 11AB)

Thanos Deep Dive: Inside a Distributed Monitoring System

- Bartłomiej Płotka & Frederic Branczyk, RedHat
- Wednesday November 20, 2019
- 5:20pm (Room 6C)

FAQ

- How does Thanos compare to Cortex/M3DB/X/Y/Z?
- When do I use the sidecar vs the receiver?
- Sounds too good to be true; what are the bottlenecks?
- Can I use Thanos with my favourite object storage provider?
- How do I know if I need Thanos vs a big Prometheus?