



IMPROBABLE



Thanos

Prometheus at Scale

Bartek Płotka



@bwplotka

github.com/improbable-eng/thanos

Edinburgh, 22th October 2018



Bartek Płotka
Software Engineer

bartek@improbable.io

[!\[\]\(99f58673407353e96a019fbca558fd72_img.jpg\)](#) [!\[\]\(2113e5cba4d11862fa536c379e9b61cd_img.jpg\)](#) [@bwplotka](#)

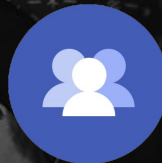


*"Improbable's platform, SpatialOS, is designed to let anyone build massive simulations, running in the cloud: imagine Minecraft with thousands of players in the same space or researchers creating simulated cities to model the behaviour of millions. **Its ultimate goal: to create totally immersive, persistent virtual worlds.**"*

- WIRED, May 2017



Founded:
2012



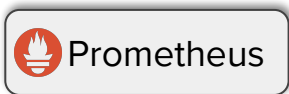
Employees:
310



Games in
Development:
+19

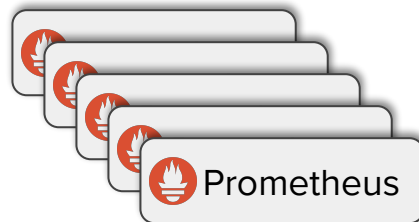
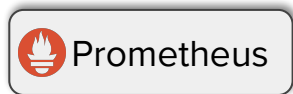


Agenda



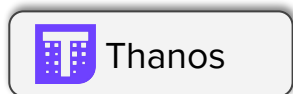
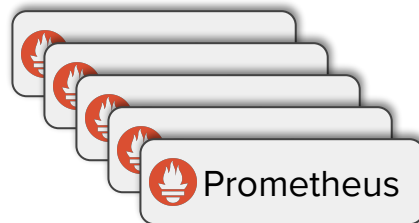
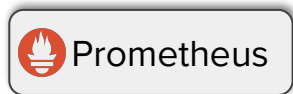


Agenda

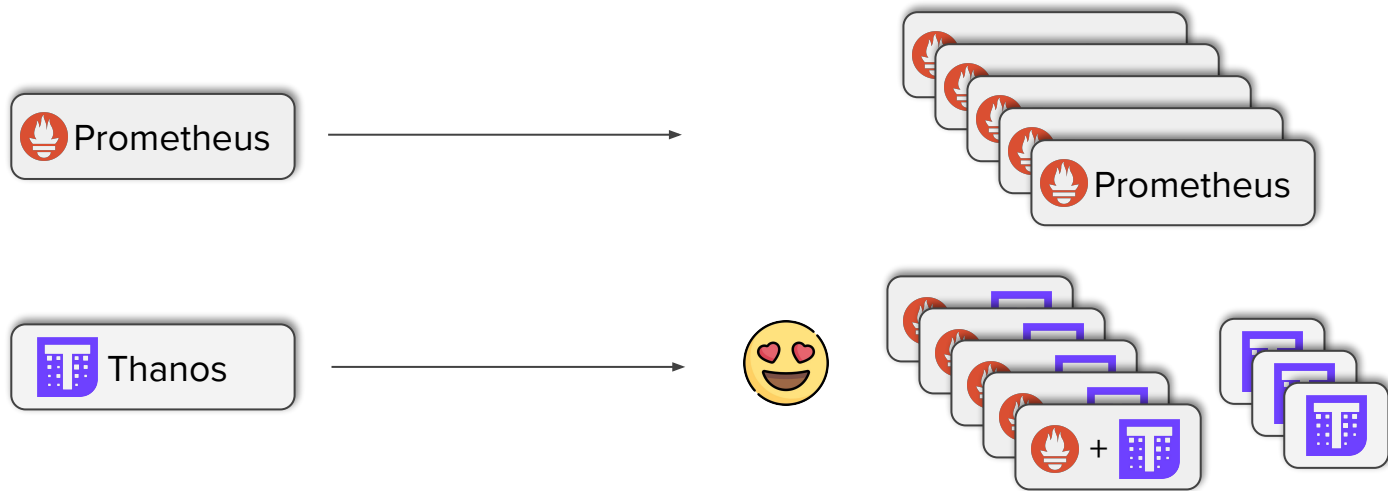




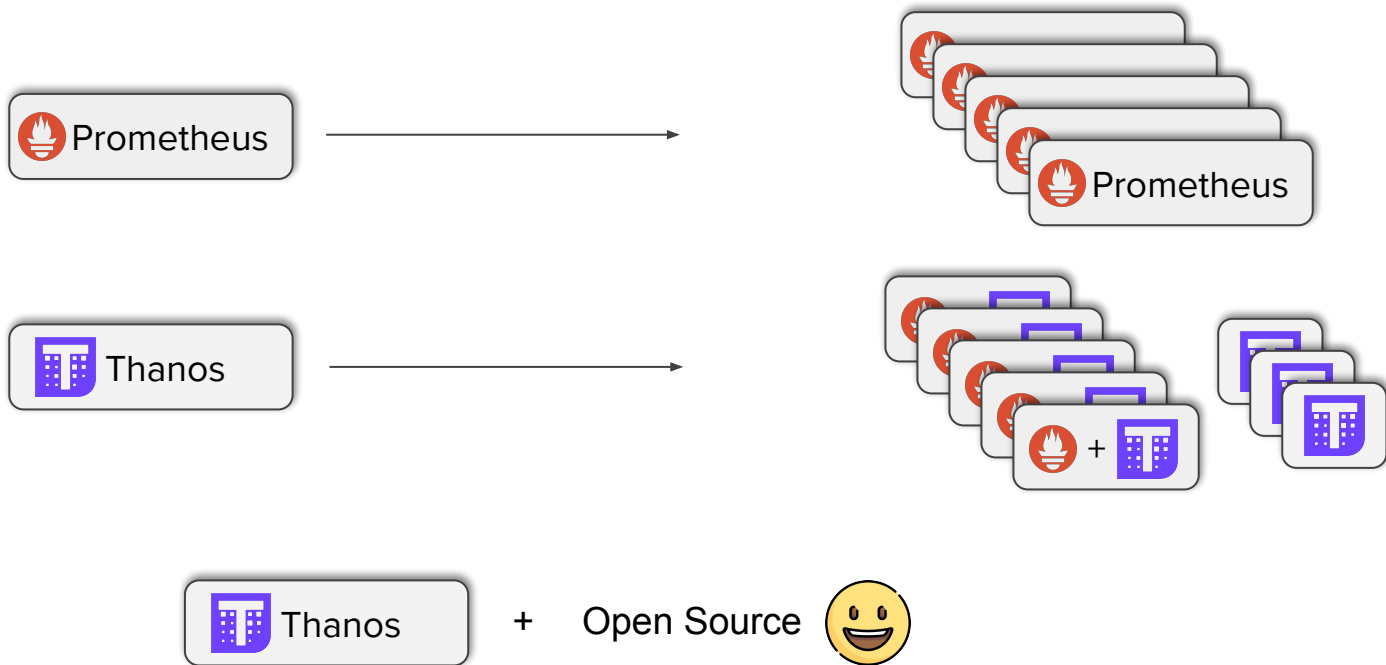
Agenda



Agenda



Agenda



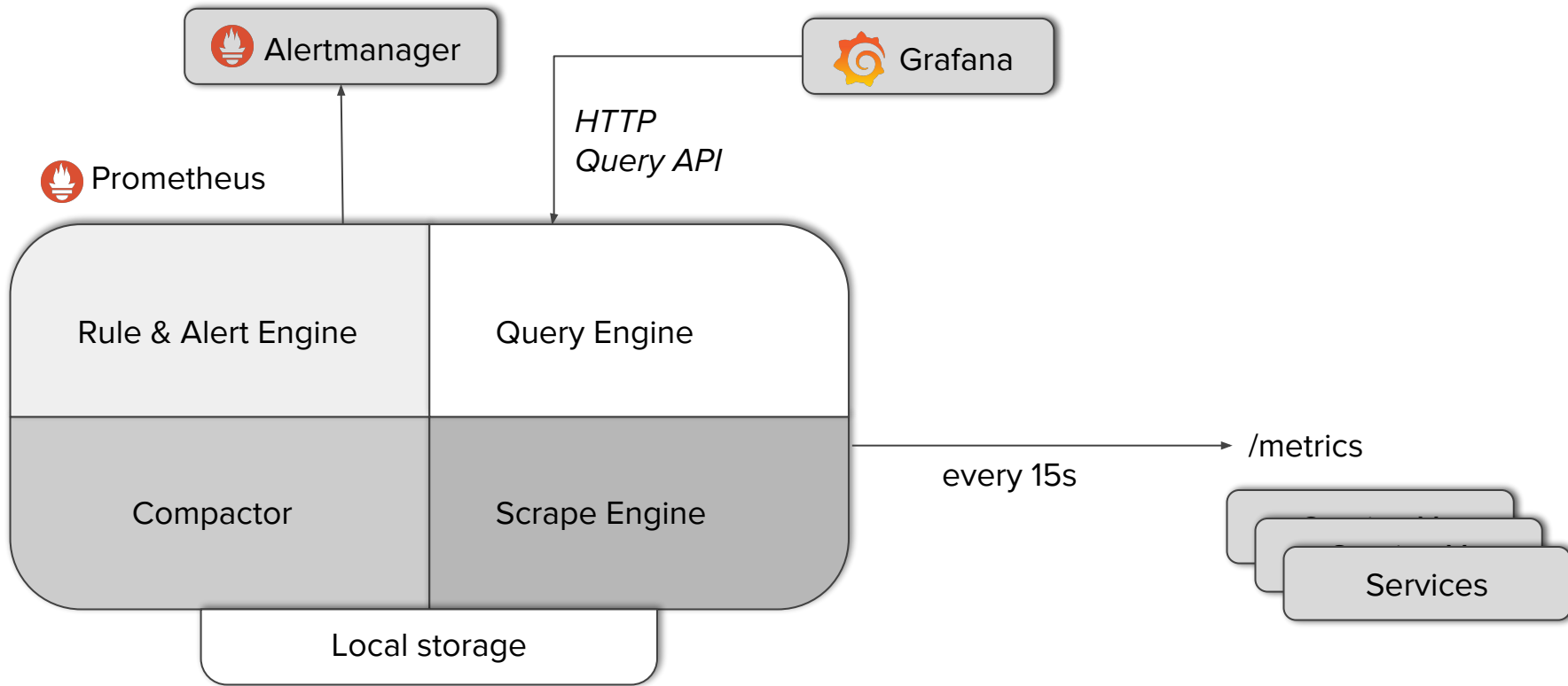


Prometheus

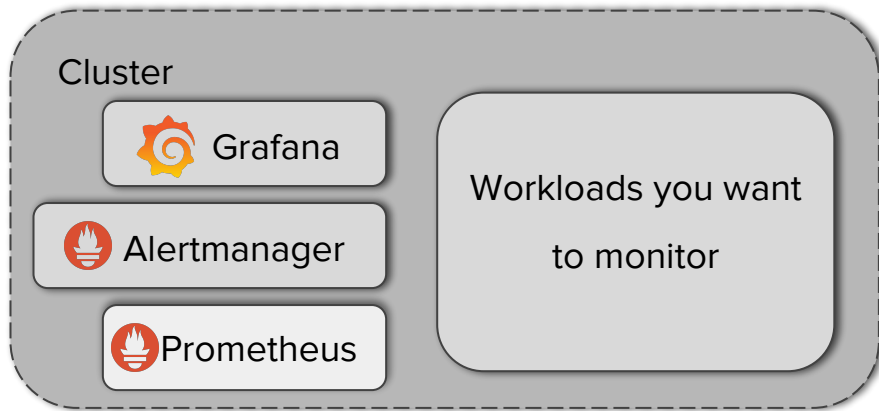
<https://github.com/prometheus/prometheus>



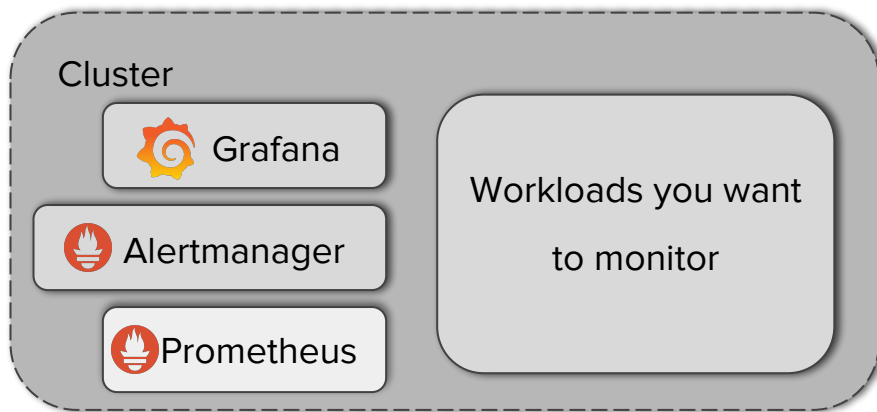
Prometheus



Single Prometheus

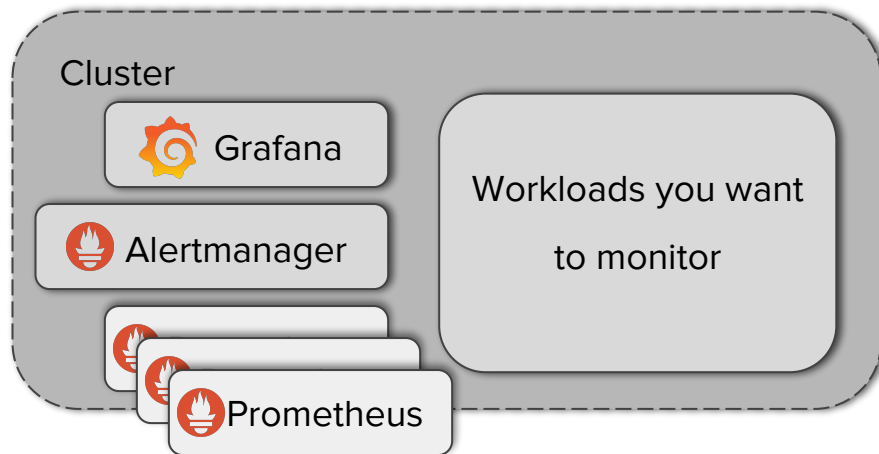


Single Prometheus

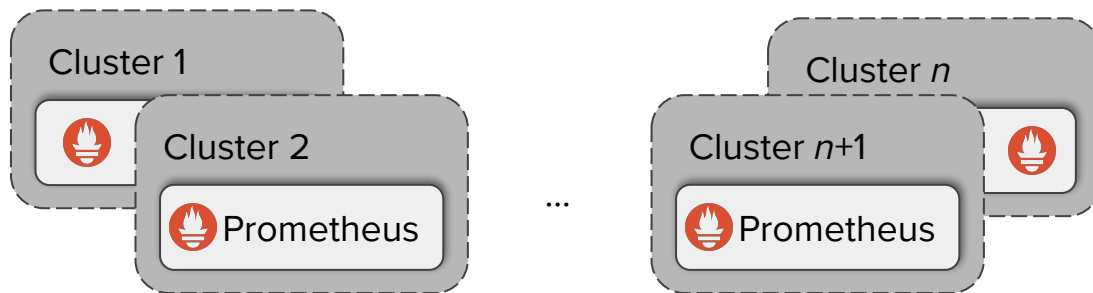


[200 000 samples / sec = 1 CPU, 13 GB RAM](#) = **3 mln series with 15s scrape interval.**

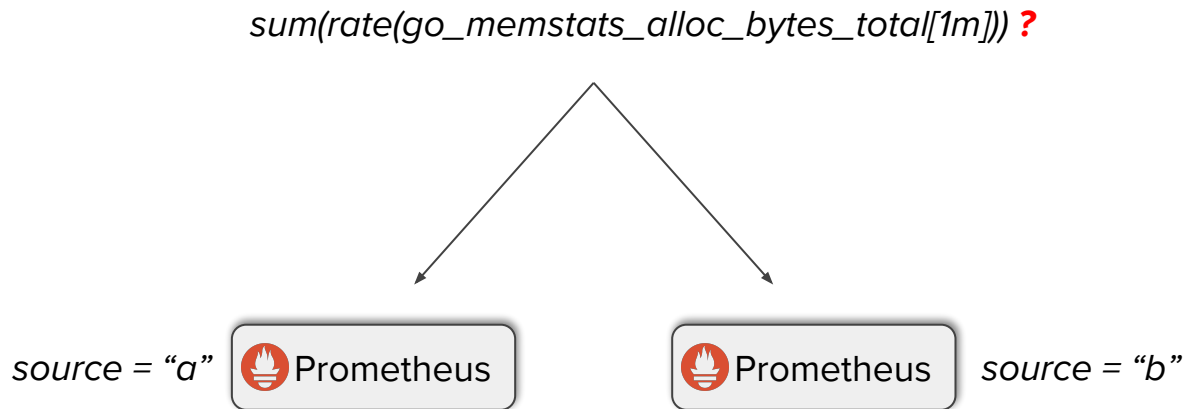
High Availability



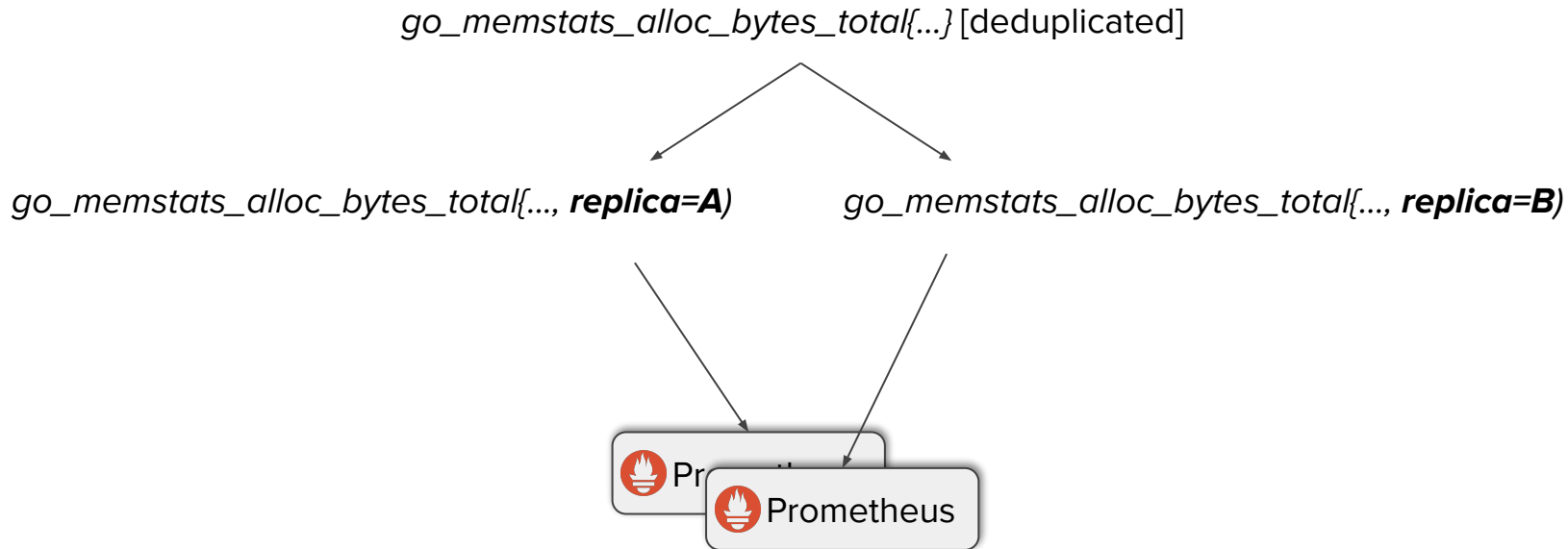
Globally distributed clusters



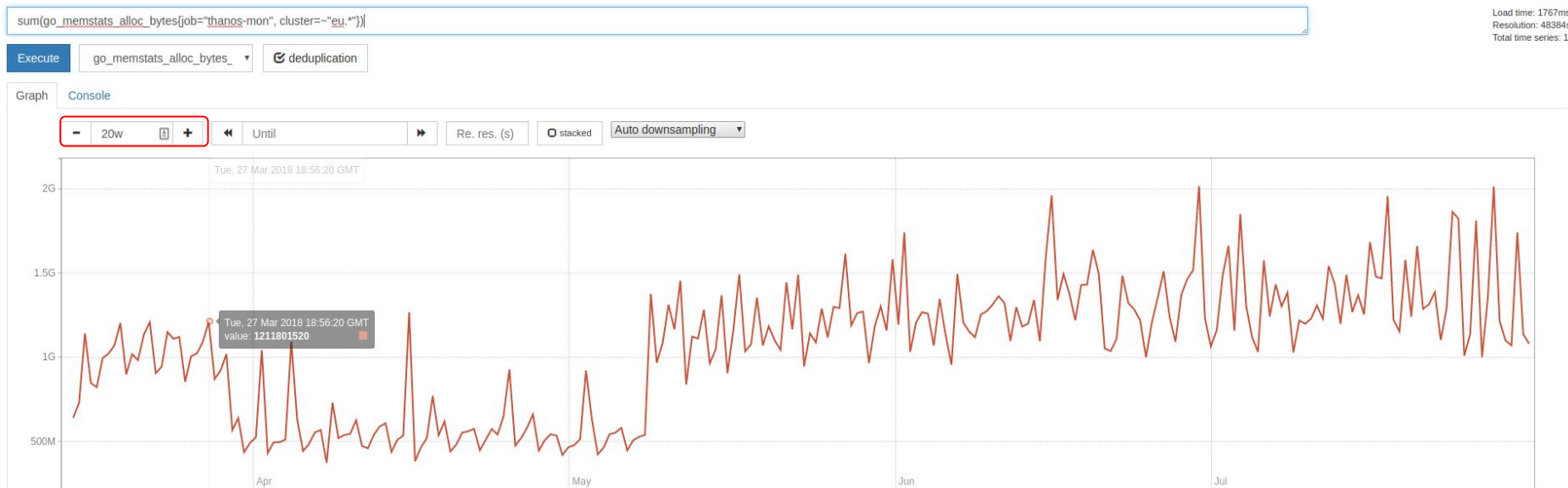
Problem: Global View



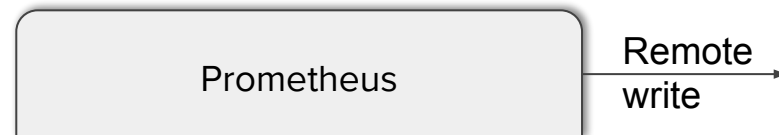
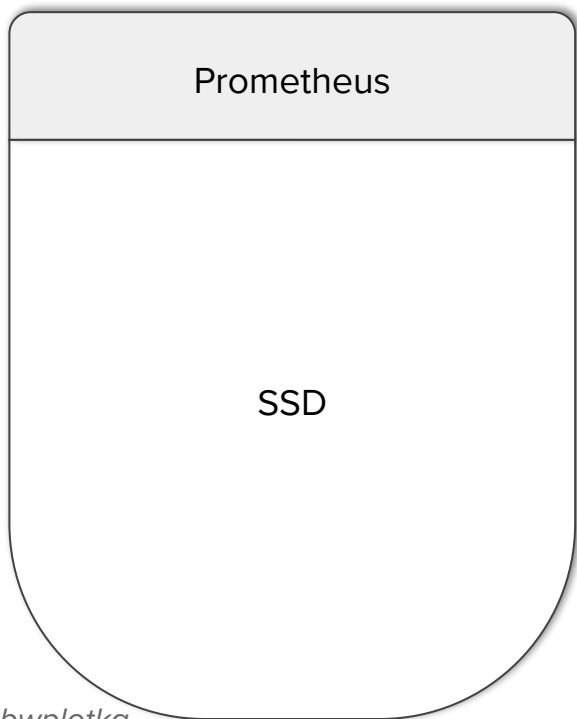
❏ Problem: Global View + HA



Problem: Metric retention



Problem: Metric retention



Thanos

Goals

- Have a global view
- Have a HA in place
- Increase retention

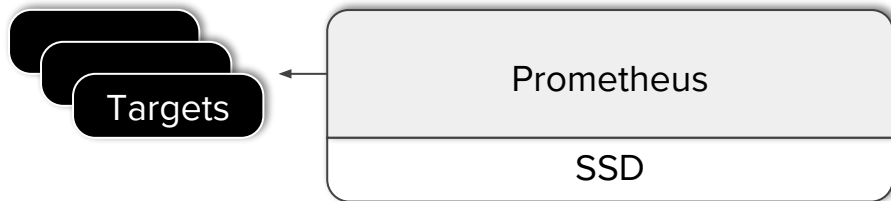




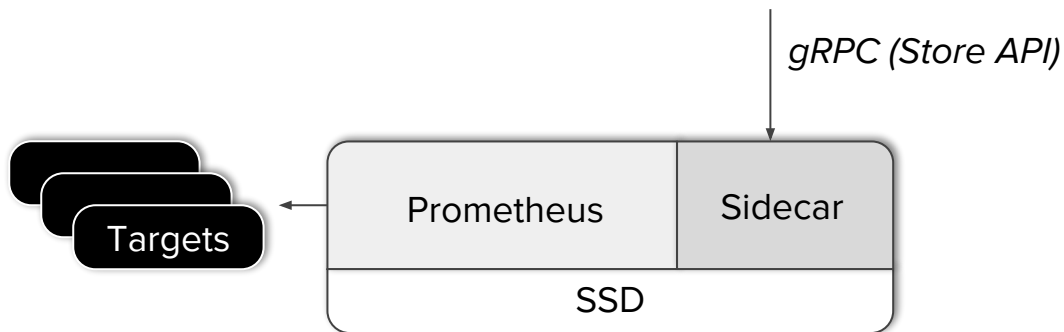
Global View

*See everything from a single
place!*

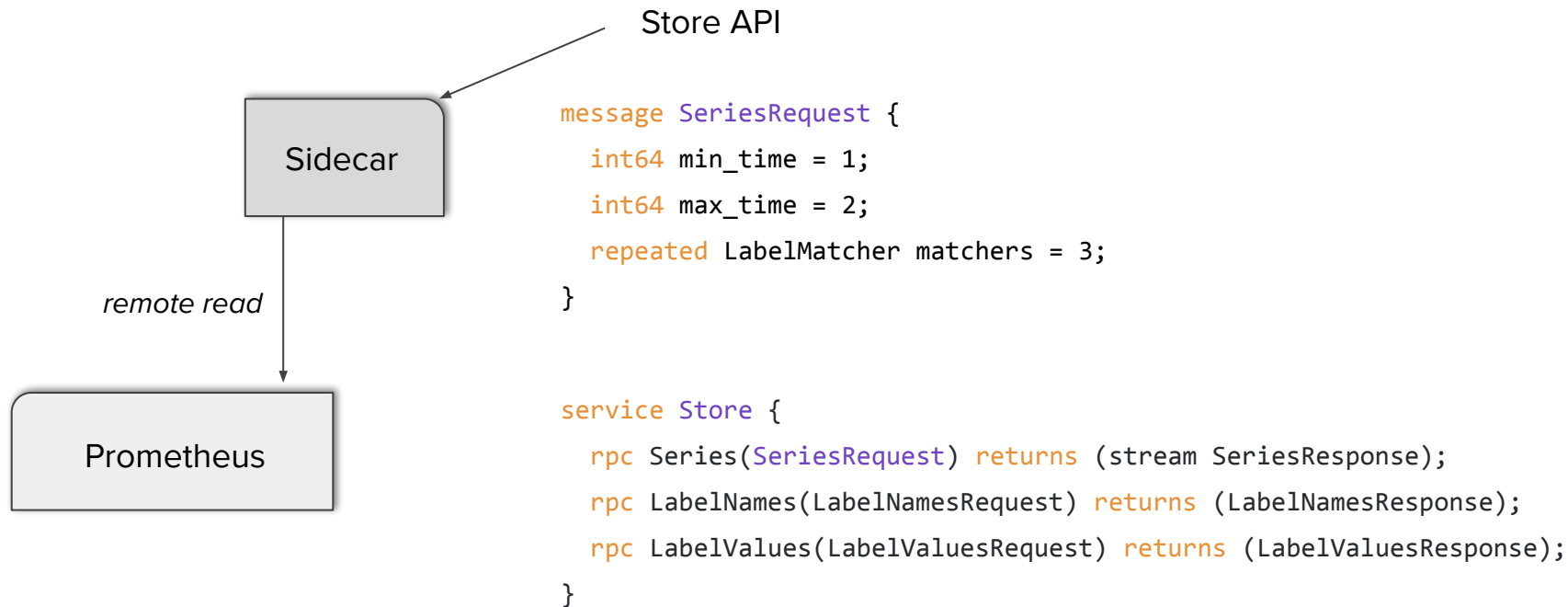
Prometheus



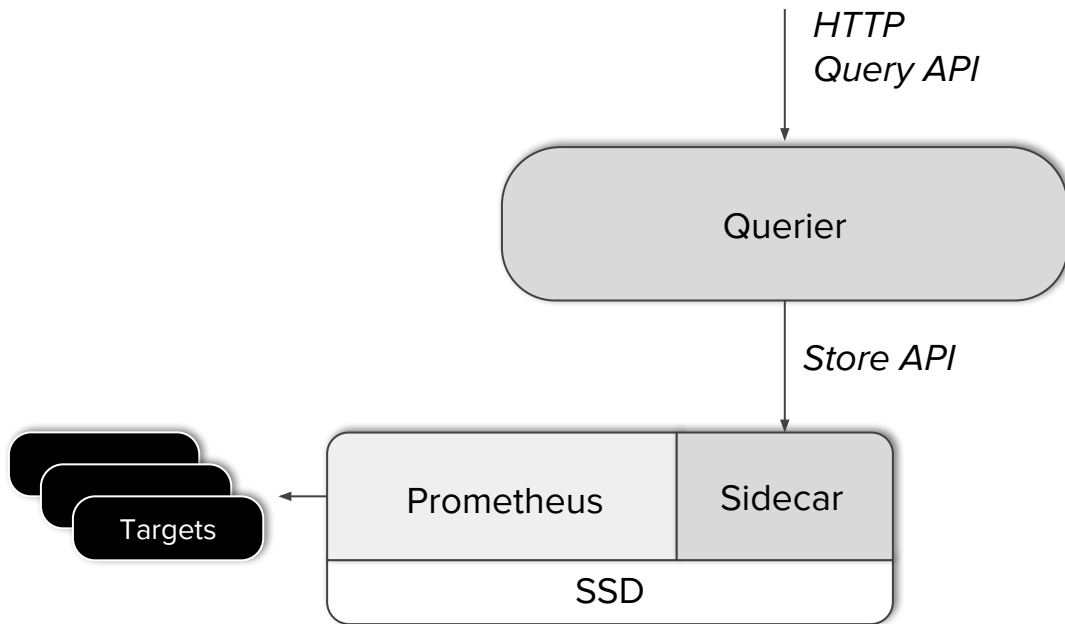
Sidecar



Store API

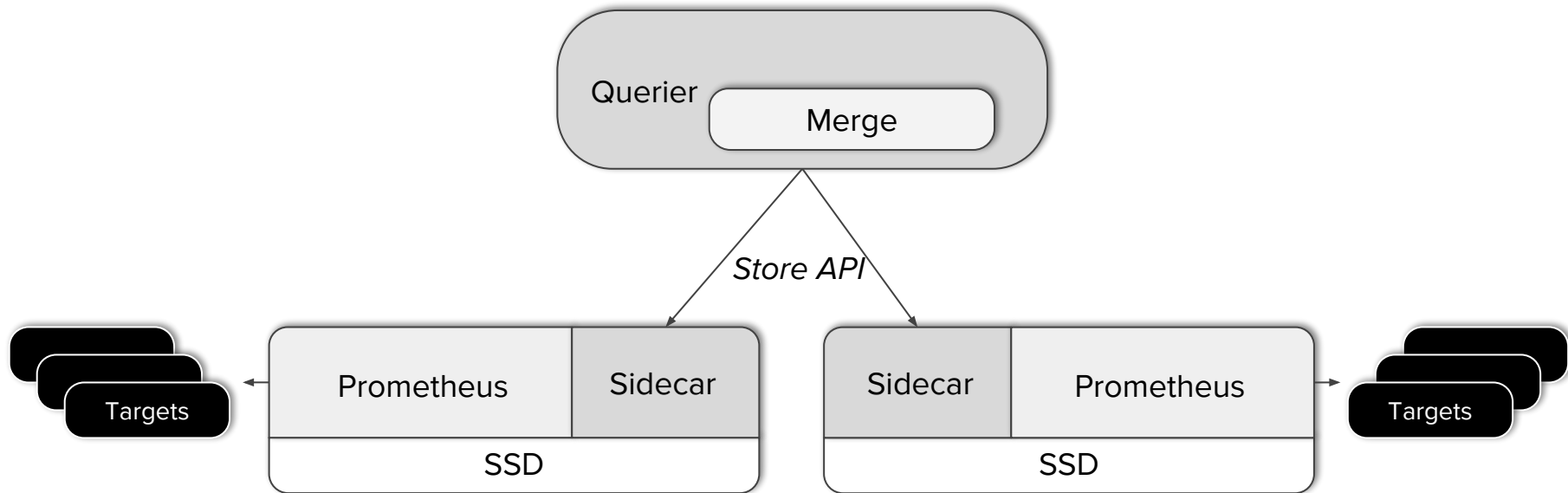


Querier

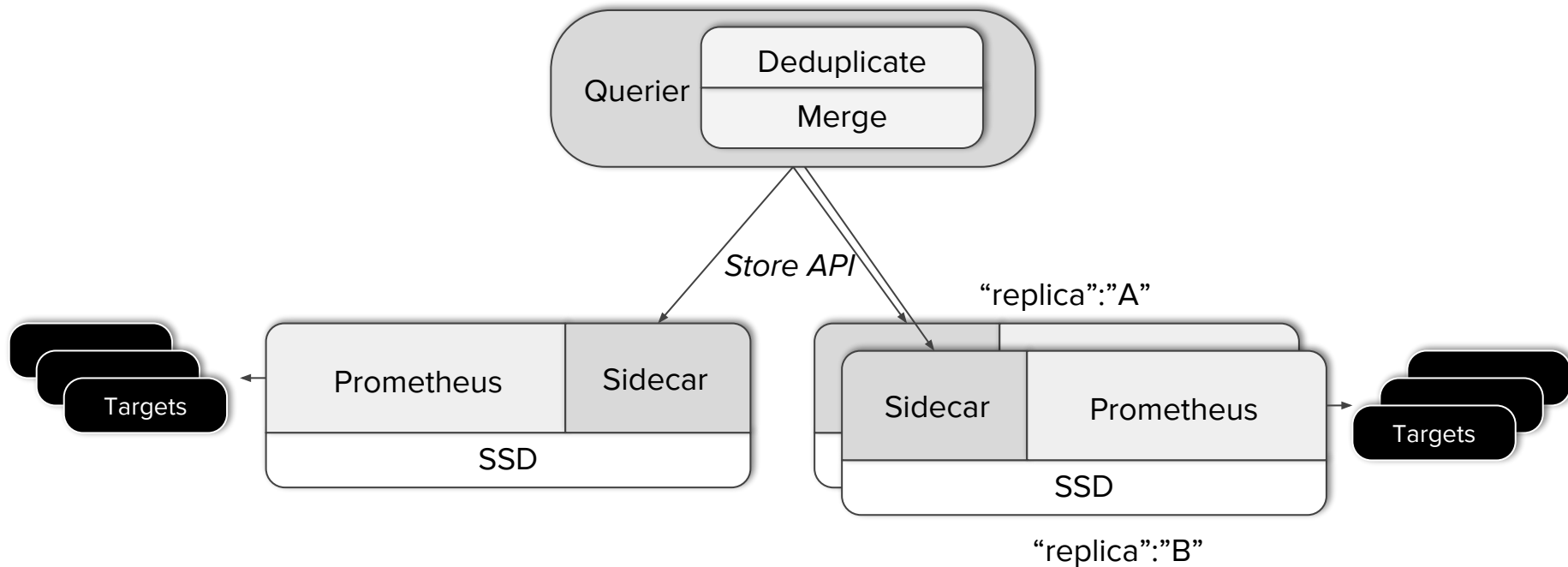




Global View



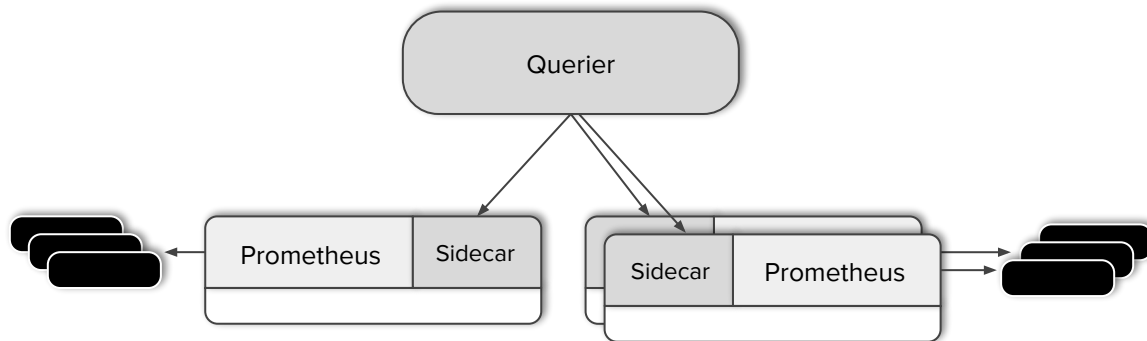
Global View + Availability



Thanos

Goals

- Have a global view ✓
- Have a HA in place ✓





Global View + Availability

Thanos Graph Status Help

go_memstats_alloc_bytes(cluster="us1-testing", job="thanos-mon", pod="thanos-mon-1")

Execute

http_request_duration_millise

☐ deduplication

Load time: 6078ms
Resolution: 19353s
Total time series: 2

Graph Console

- 8w [i] + << 2018-08-22 11:51 >> Re. res. (s) ☐ stacked Only raw data



go_memstats_alloc_bytes(cluster="us1-testing", environment="testing", host="us1-testing-worker-zq95", instance="10.122.16.56:9090", job="thanos-mon", monitor_namespace="default", namespace="default", pod="thanos-mon-1", replica="thanos-mon-1")
go_memstats_alloc_bytes(cluster="us1-testing", environment="testing", host="us1-testing-worker-zq95", instance="10.122.16.56:9090", job="thanos-mon", monitor_namespace="default", namespace="default", pod="thanos-mon-1", replica="thanos-mon-0")

Remove Graph

Add Graph



Global View + Availability

Thanos Graph Status ▾ Help

go_memstats_alloc_bytes(cluster="us1-testing", job="thanos-mon", pod="thanos-mon-1")

Execute

http_request_duration_millise ▾

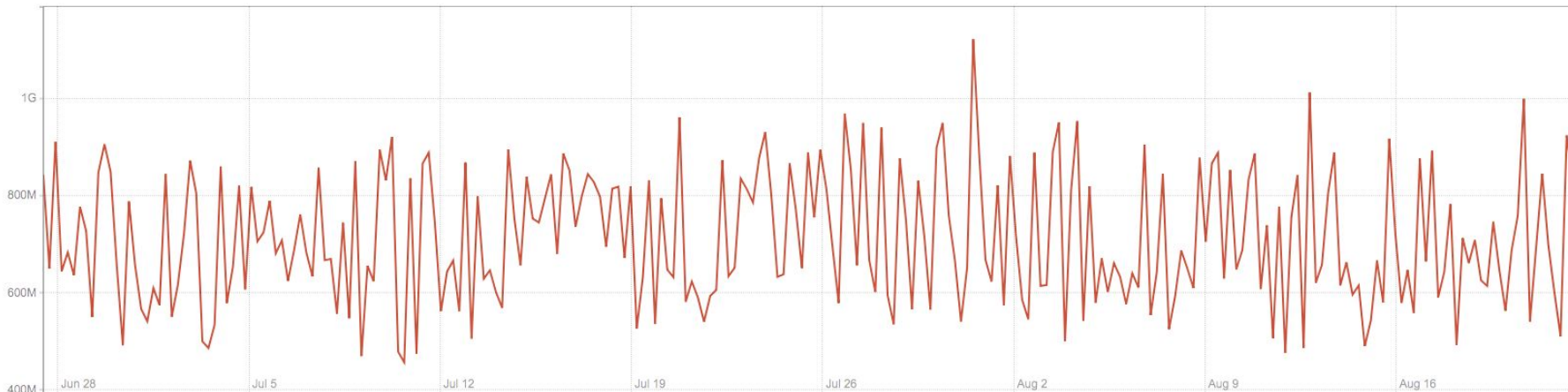
deduplication

Load time: 1980ms
Resolution: 19353s
Total time series: 1

Graph

Console

8w 2018-08-22 11:51 Re. res. (s) stacked Only raw data ▾



go_memstats_alloc_bytes(cluster="us1-testing", environment="testing", host="us1-testing-worker-zq95", instance="10.122.16.56:9090", job="thanos-mon", monitor_namespace="default", namespace="default", pod="thanos-mon-1")

Add Graph

Remove Graph

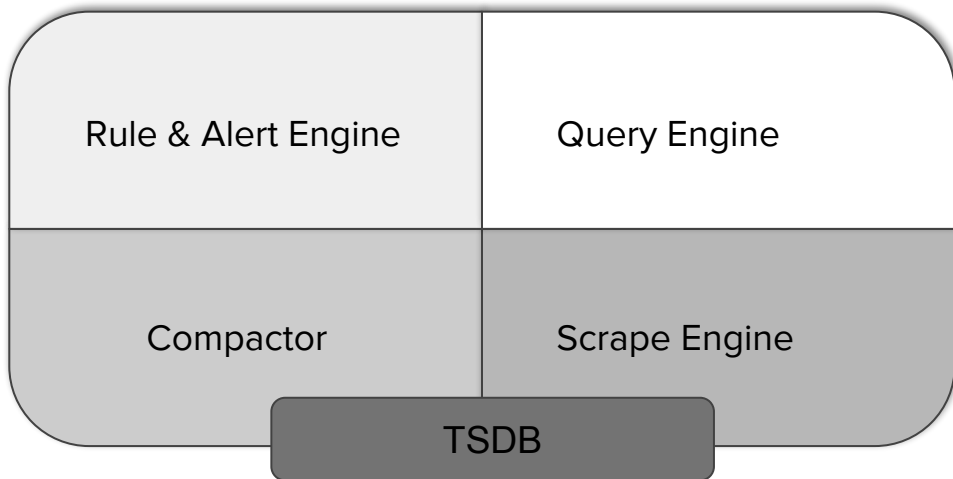
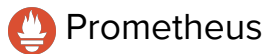


Historical Metrics

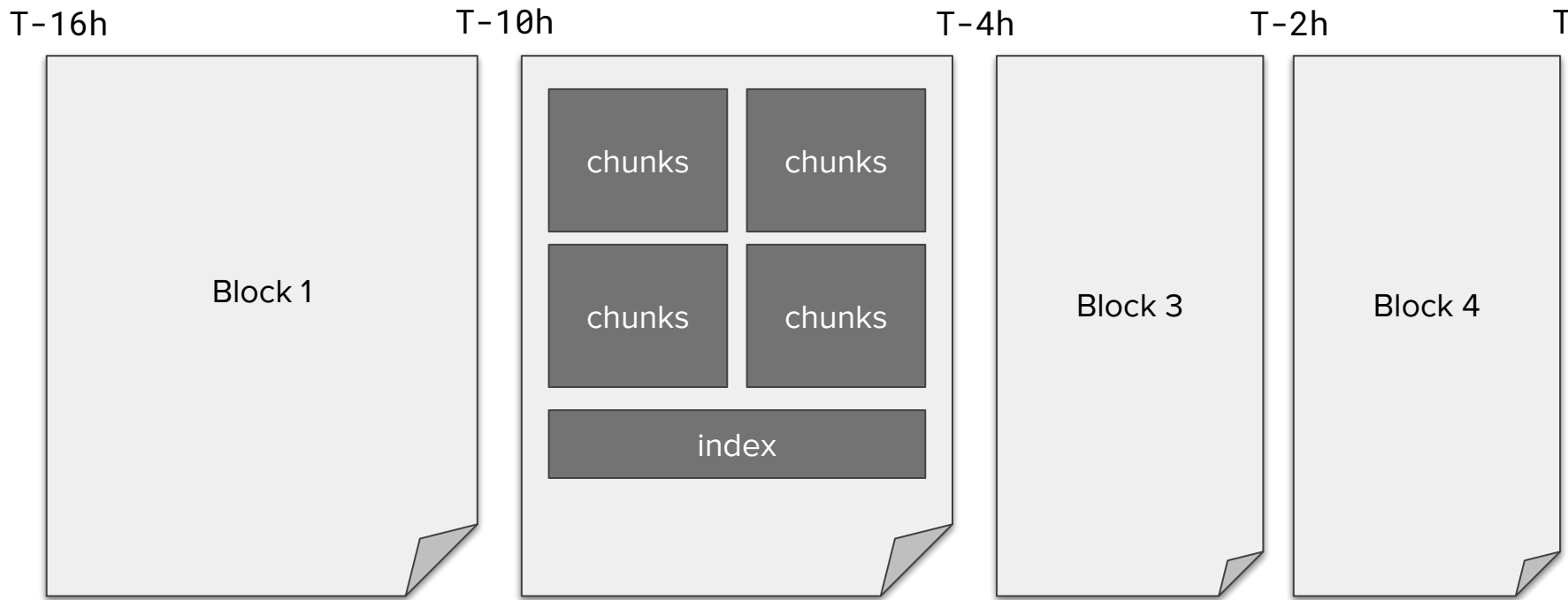
*What exactly happened
X months ago?*



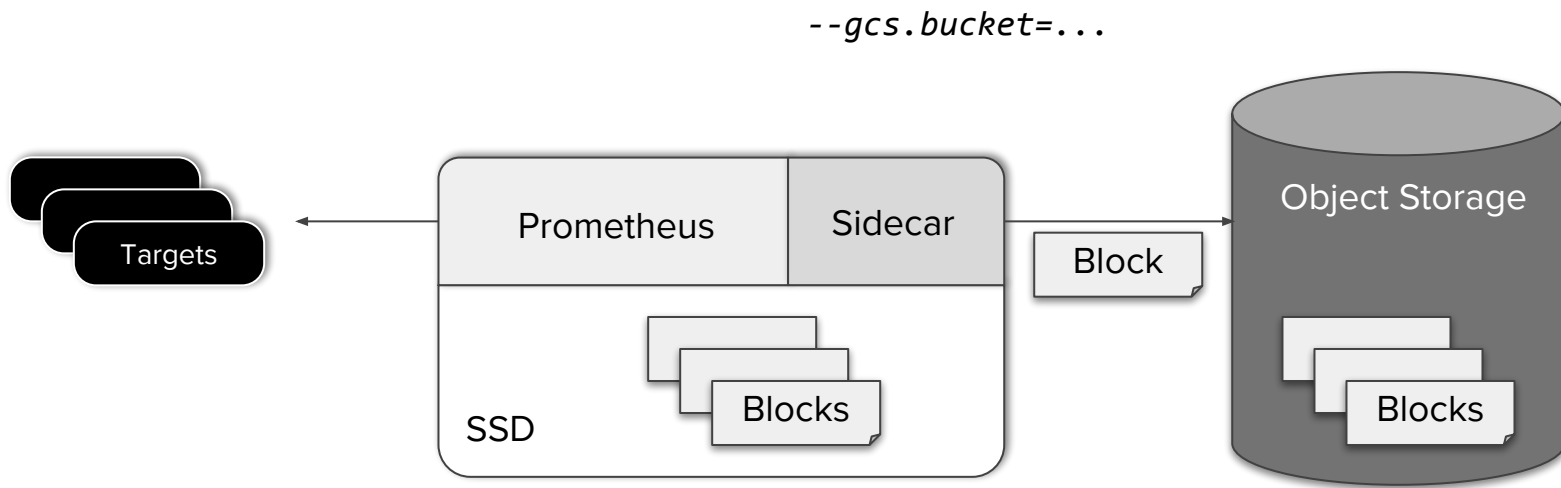
Prometheus



TSDB Layout

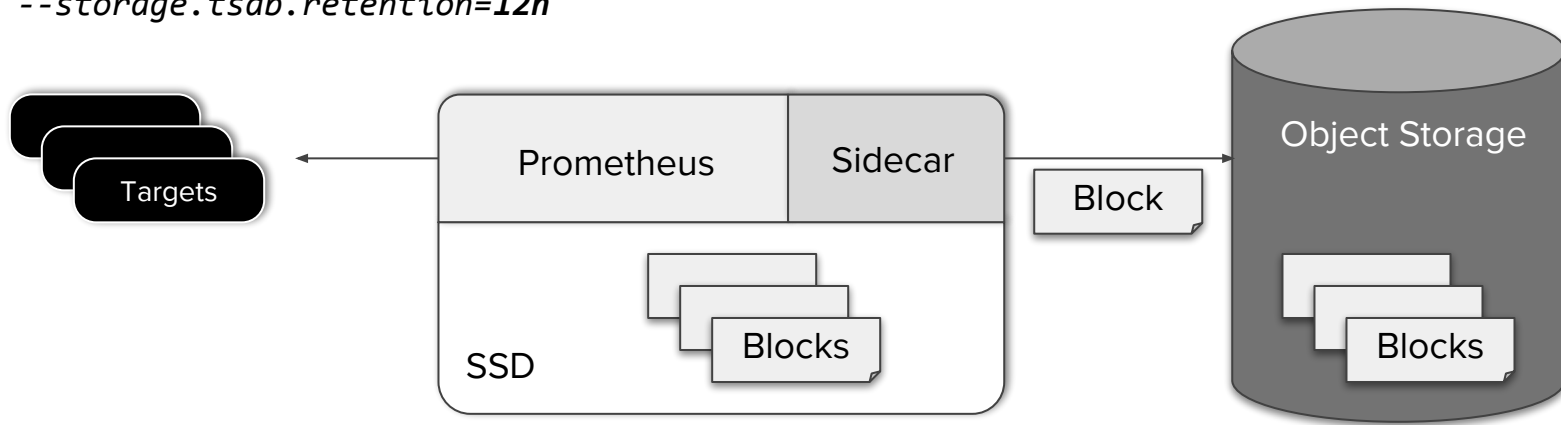


Data saving

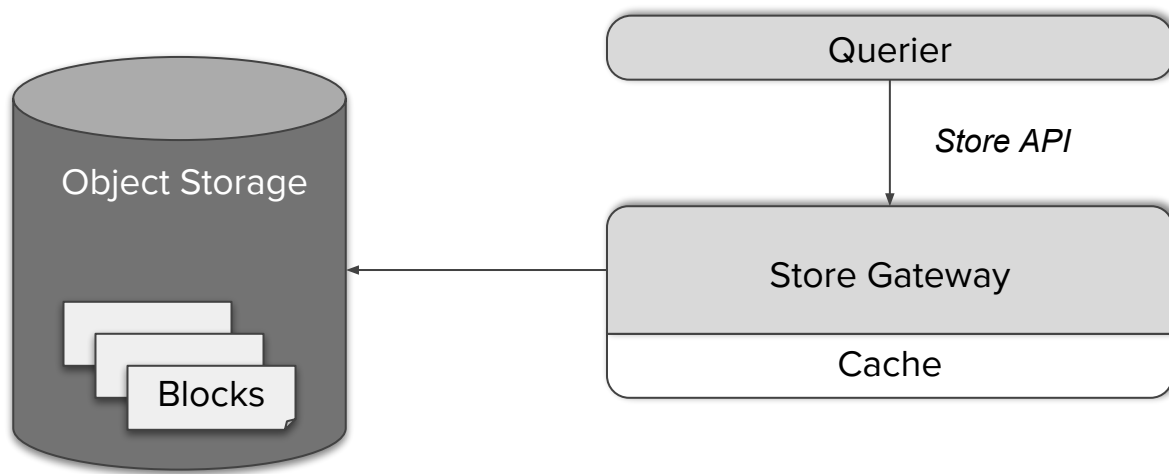


Data saving

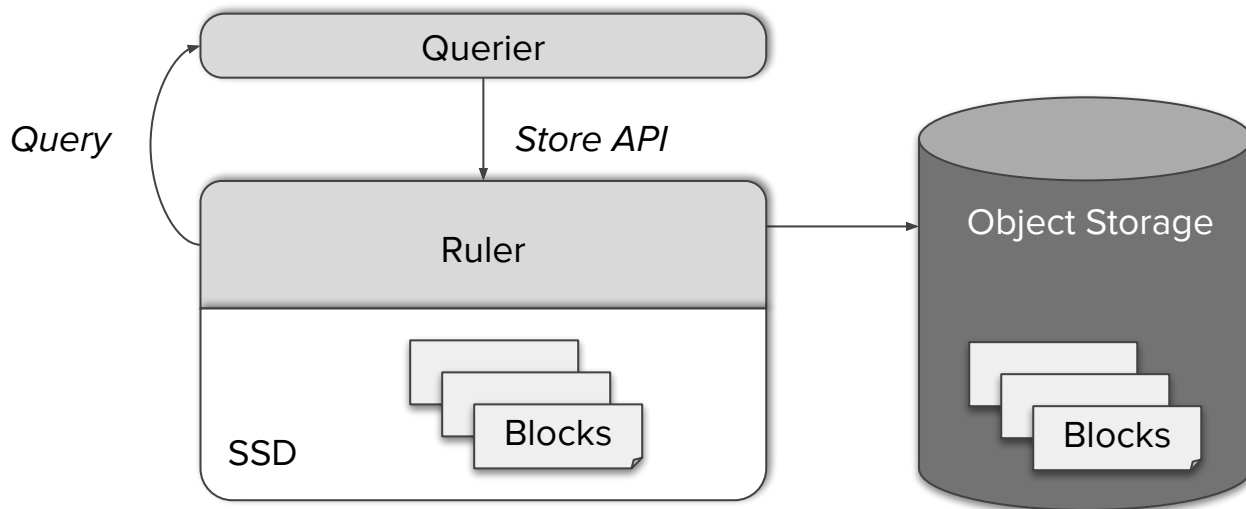
```
--storage.tsdb.max-block-duration=2h  
--storage.tsdb.retention=12h
```



Store Gateway



Ruler



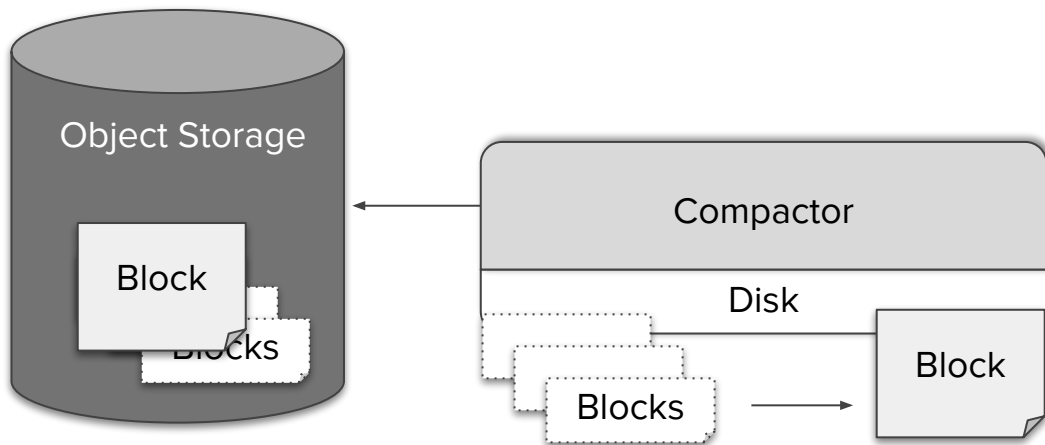


Compaction

Density matters



Compaction





Downsampling



Downsampling

Decimal	Double Representation	XOR with previous
12	0x4028000000000000	
24	0x4038000000000000	0x0010000000000000
15	0x402e000000000000	0x0016000000000000
12	0x4028000000000000	0x0006000000000000
35	0x4041800000000000	0x0069800000000000

Decimal	Double Representation	XOR with previous
15.5	0x402f000000000000	
14.0625	0x402c200000000000	0x0003200000000000
3.25	0x400a000000000000	0x0026200000000000
8.625	0x4021400000000000	0x002b400000000000
13.1	0x402a333333333333	0x000b733333333333

Raw: 16 bytes/sample

**Compressed: 1.07
bytes/sample**

Downsampling

Decompressing **one** sample takes **10-40** nanoseconds.



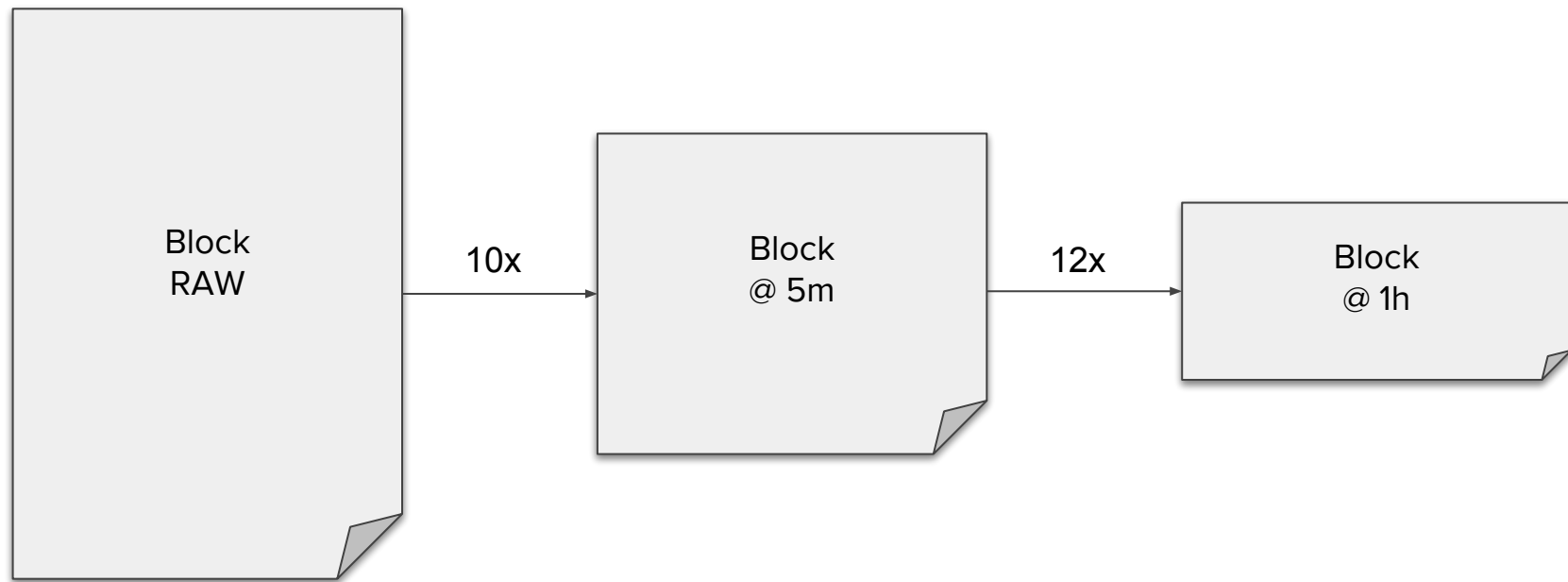
Downsampling

Decompressing **one** sample takes **10-40** nanoseconds.

Decompressing **1000 series** @ 30s scrape interval for 1 year data takes **10-40** seconds alone.

Plus your actual computation over all those samples, e.g. `rate()`

Downsampling



Thanos

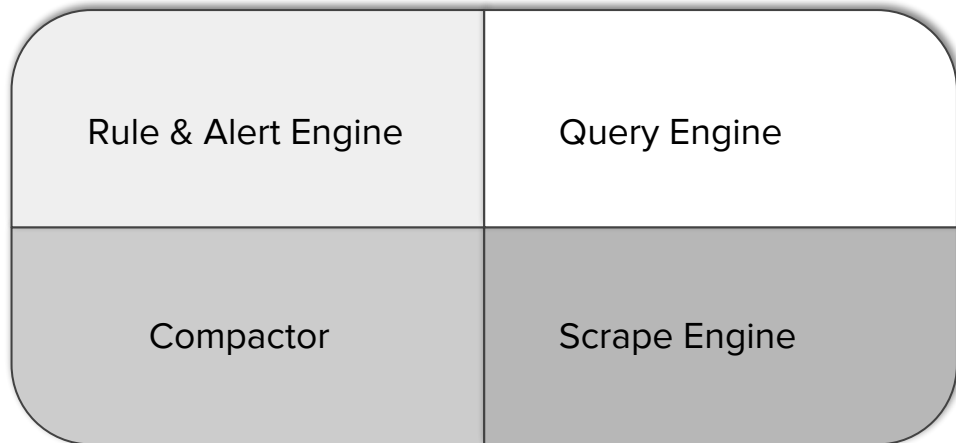
Goals

- Have a global view ✓
- Have a HA in place ✓
- Increase retention ✓

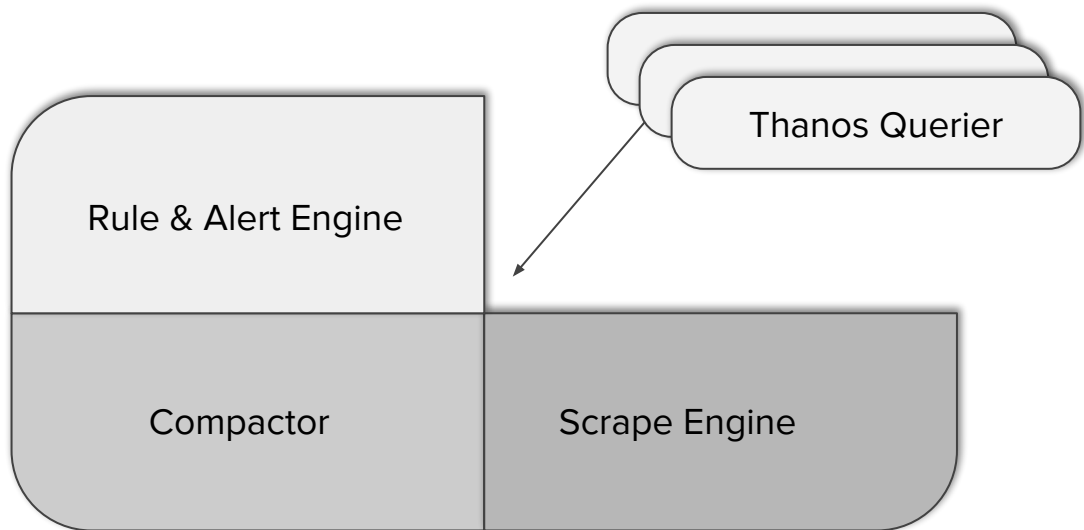


Prometheus

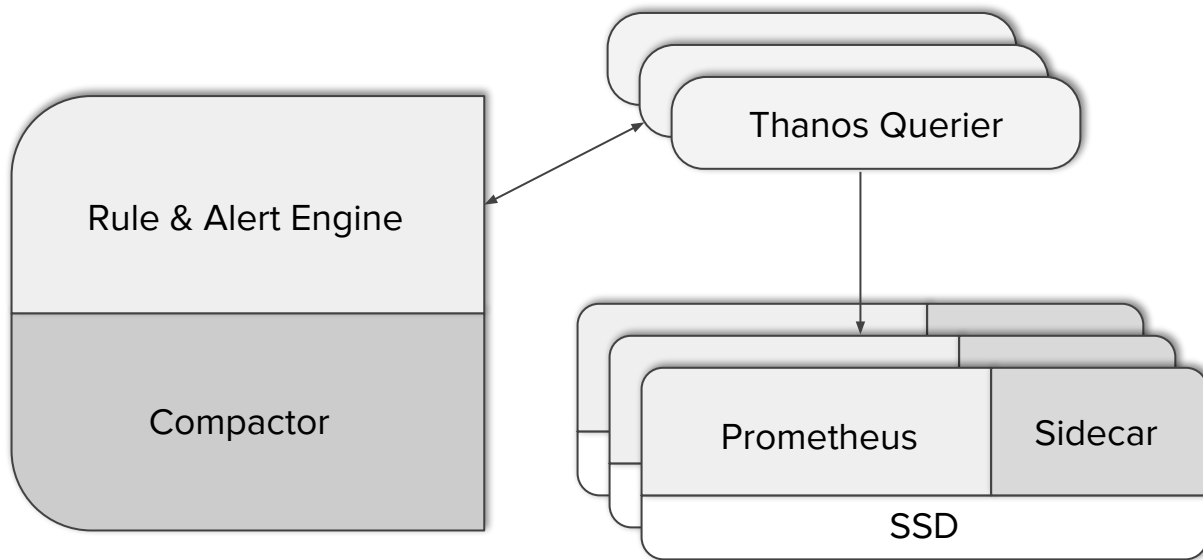
 Prometheus



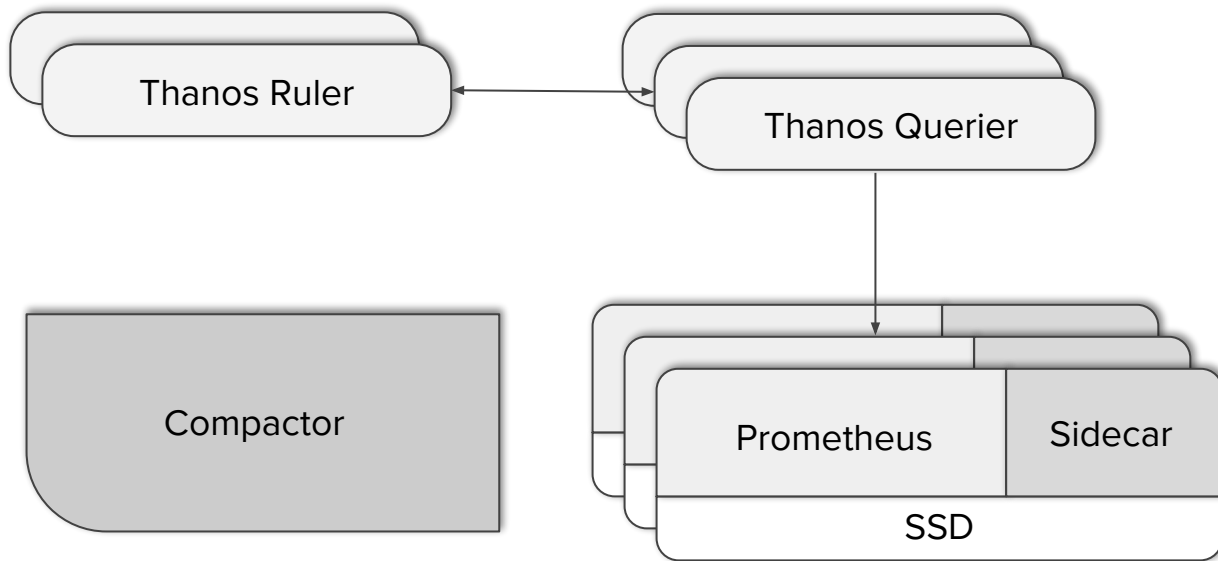
Thanos



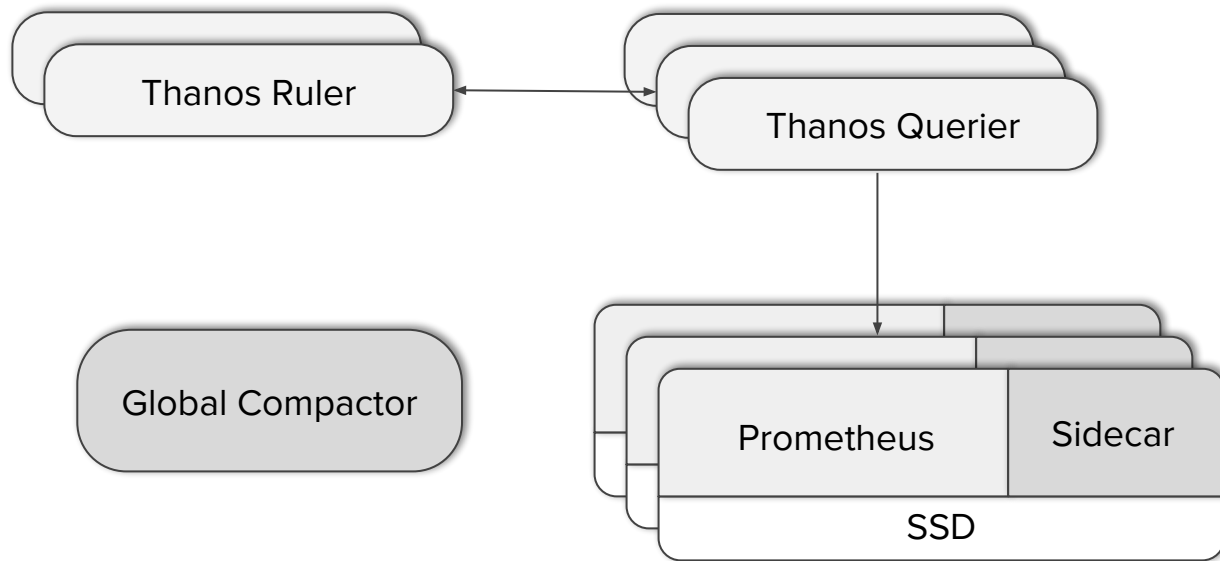
Thanos



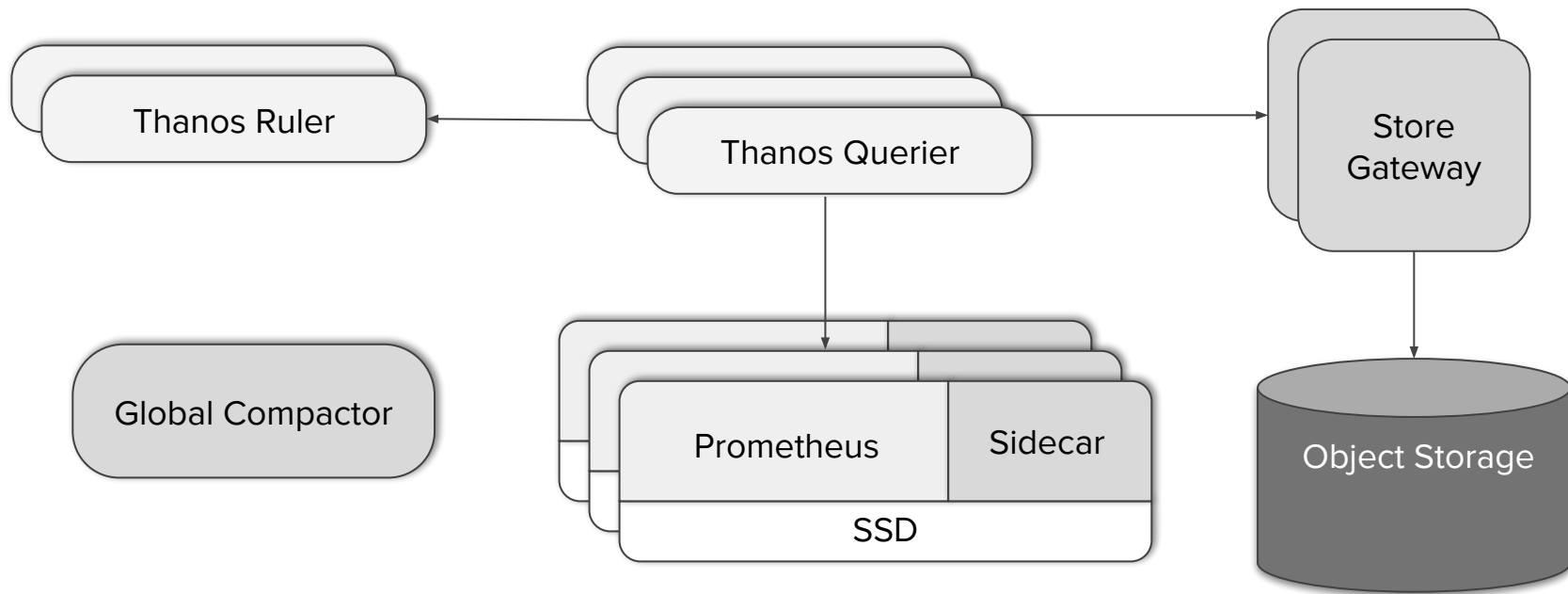
Thanos



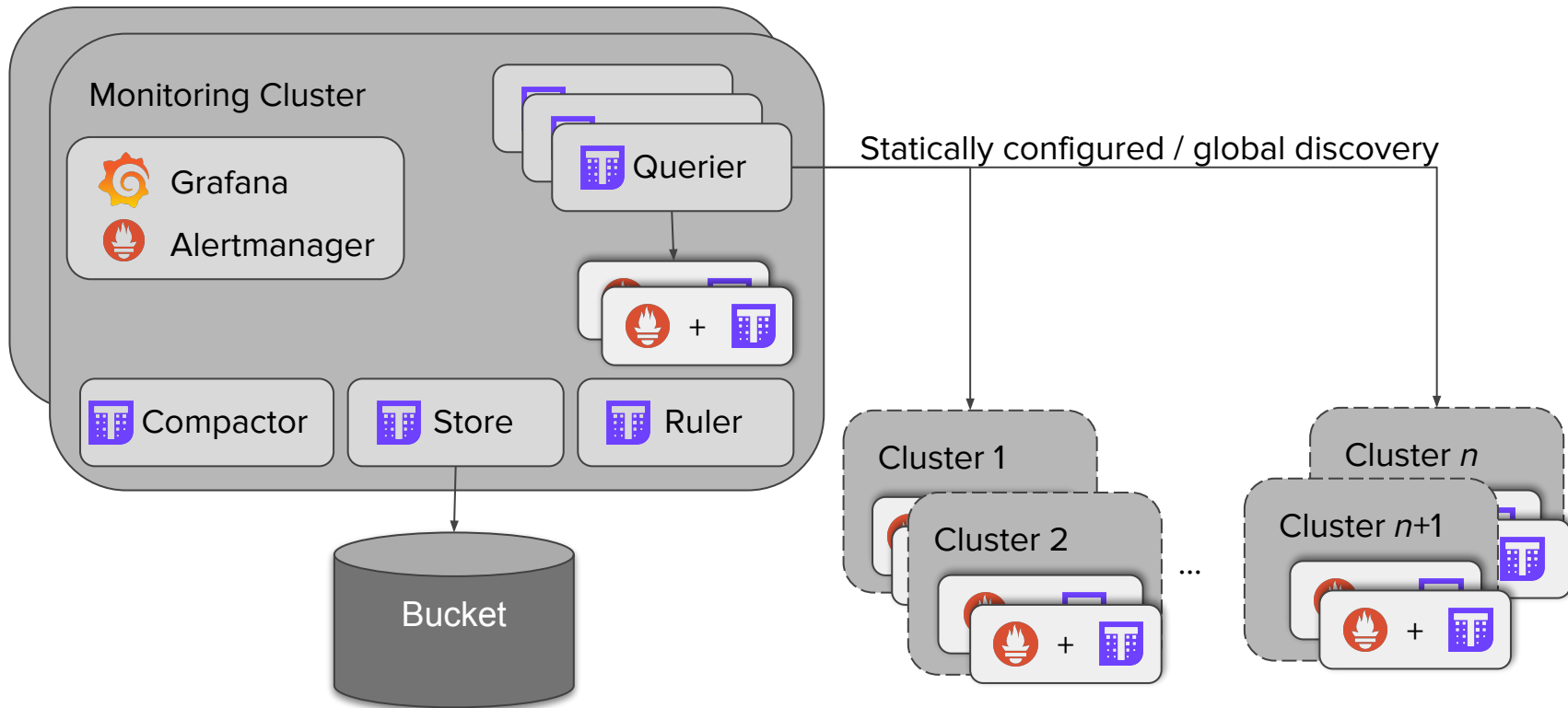
Thanos



Thanos



Example Deployment





Open Source

&



Thanos



Open Source:

It is worth it!

OPEN





**Open source from the
very beginning**



**Be flexible
while staying focused**





Stay focused: Injection API

`<file_sd_config>`

File-based service discovery provides a more generic way to configure static targets and serves as an interface to plug in custom service discovery mechanisms.

It reads a set of files containing a list of zero or more `<static_config>`s. Changes to all defined files are detected via disk watches and applied immediately. Files may be provided in YAML or JSON format. Only changes resulting in well-formed target groups are applied.

The JSON file must contain a list of static configs, using this format:

```
[
  {
    "targets": [ "<host>", ... ],
    "labels": {
      "<labelname>": "<labelvalue>", ...
    }
  },
  ...
]
```

Source: https://prometheus.io/docs/prometheus/latest/configuration/configuration/#<file_sd_config>



Less magic == better

Avoid magic: Gossip example

- Easy to misconfigure
- Hard to debug
- Difficult cross cluster setup
- Over-complicated for Thanos needs

Thanos

Goals

- Have a global view ✓
- Have a HA in place ✓
- Increase retention ✓
- Join effort with community ✓





Summary

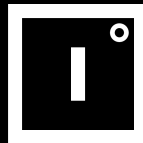
- Start small with just single Prometheus.
- Extend your setup gradually.
 - Just global querier.
 - Federated global querier.
 - Object storage.
- Model your Thanos deployment.

Generally: Share early, keep project focused and simple.



...psst, join our slack workspace for more info!

github.com/improbable-eng/thanos



improbable.io

Credits:

- [Percona blog post about Prometheus 2 perf](#)
- [Emojis designed by Freepik from Flaticon](#)

Bartek Płotka



@bwplotka