



Prometheus, the future!

Goutham Veeramachaneni

Grafana Labs

About Me

- Prometheus maintainer
- Cortex maintainer
- Loki maintainer
- BIKING





Prometheus: The monitoring system

- Made by frustrated Xogglers at Soundcloud
- Was really good
- People loved it



Prometheus: The monitoring system

- Made by frustrated Xogglers at Soundcloud
- Was really good
- People loved it



Prometheus: The monitoring system

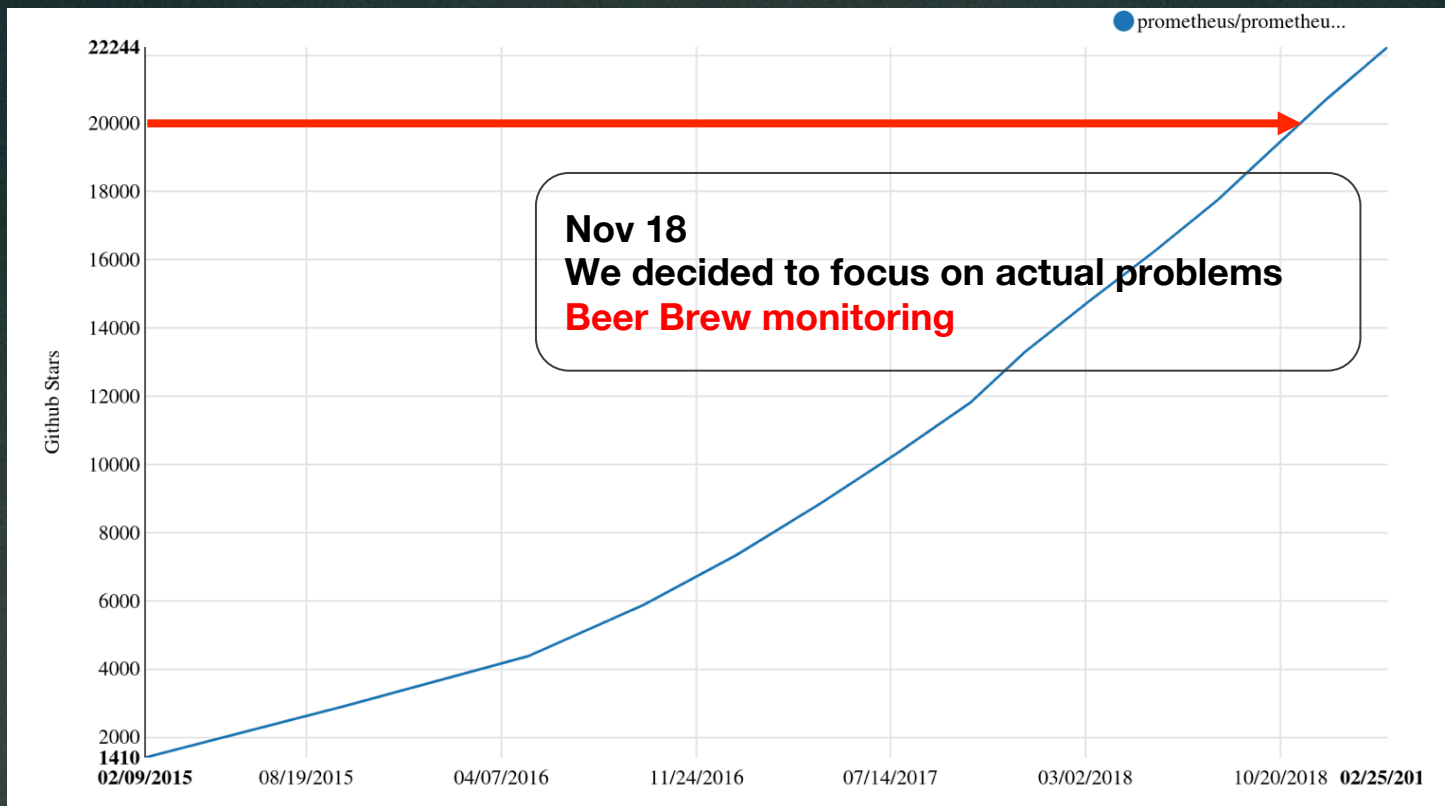
- Made by frustrated Xoogle engineers
- Was really good
- People loved it



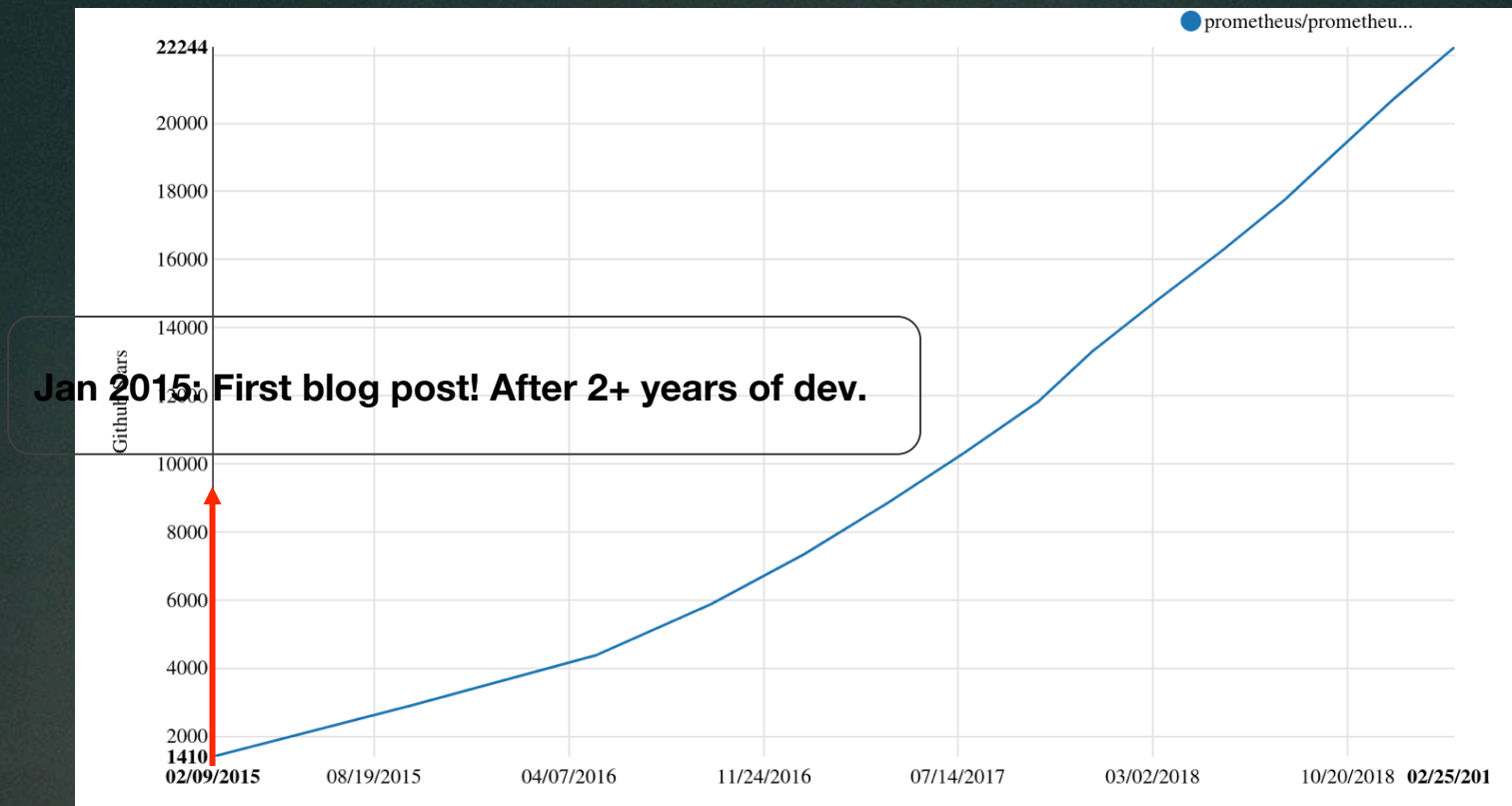
Prometheus: Engineering

- Made by frust
- Was really good
- People loved it

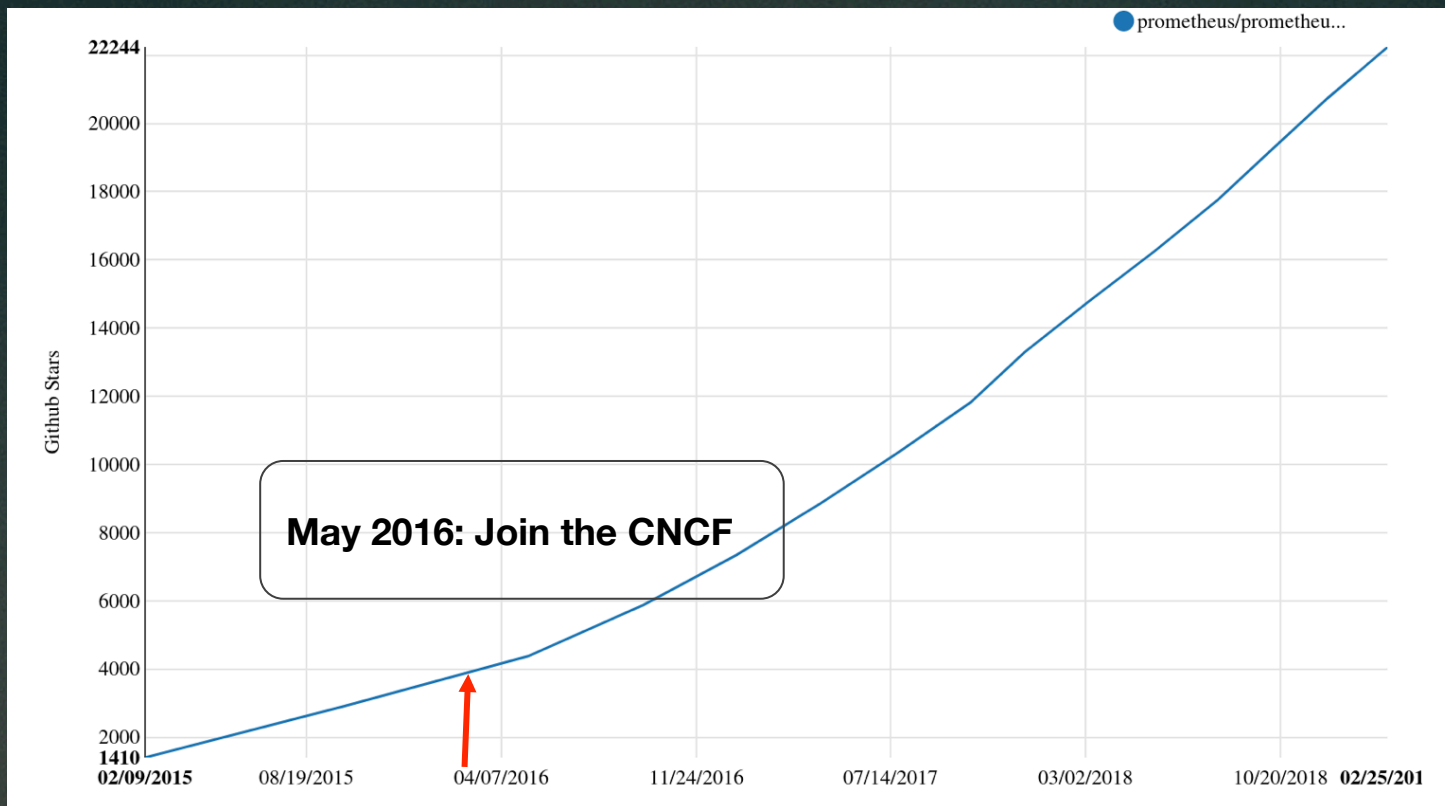
Prometheus: The monitoring system



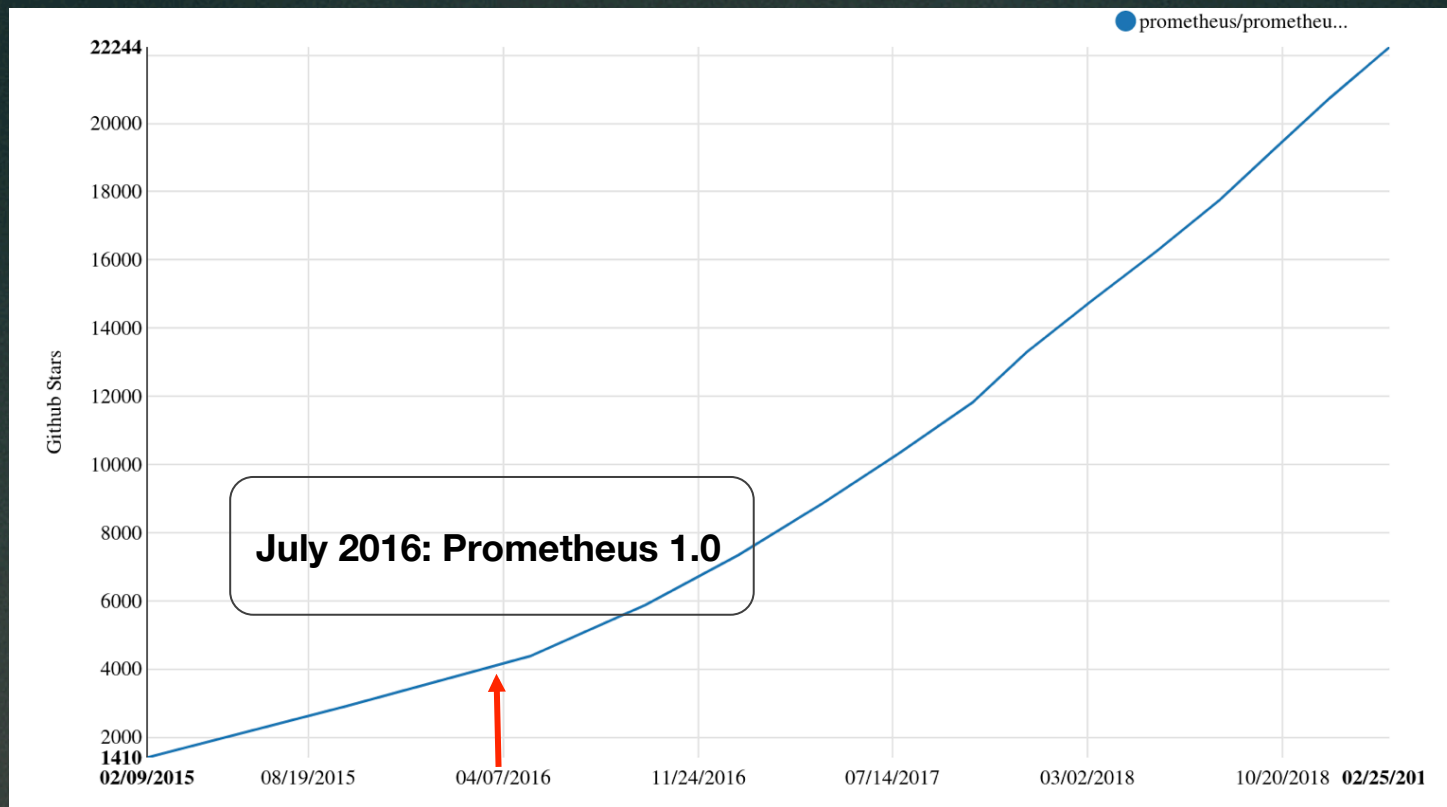
Prometheus: The monitoring system



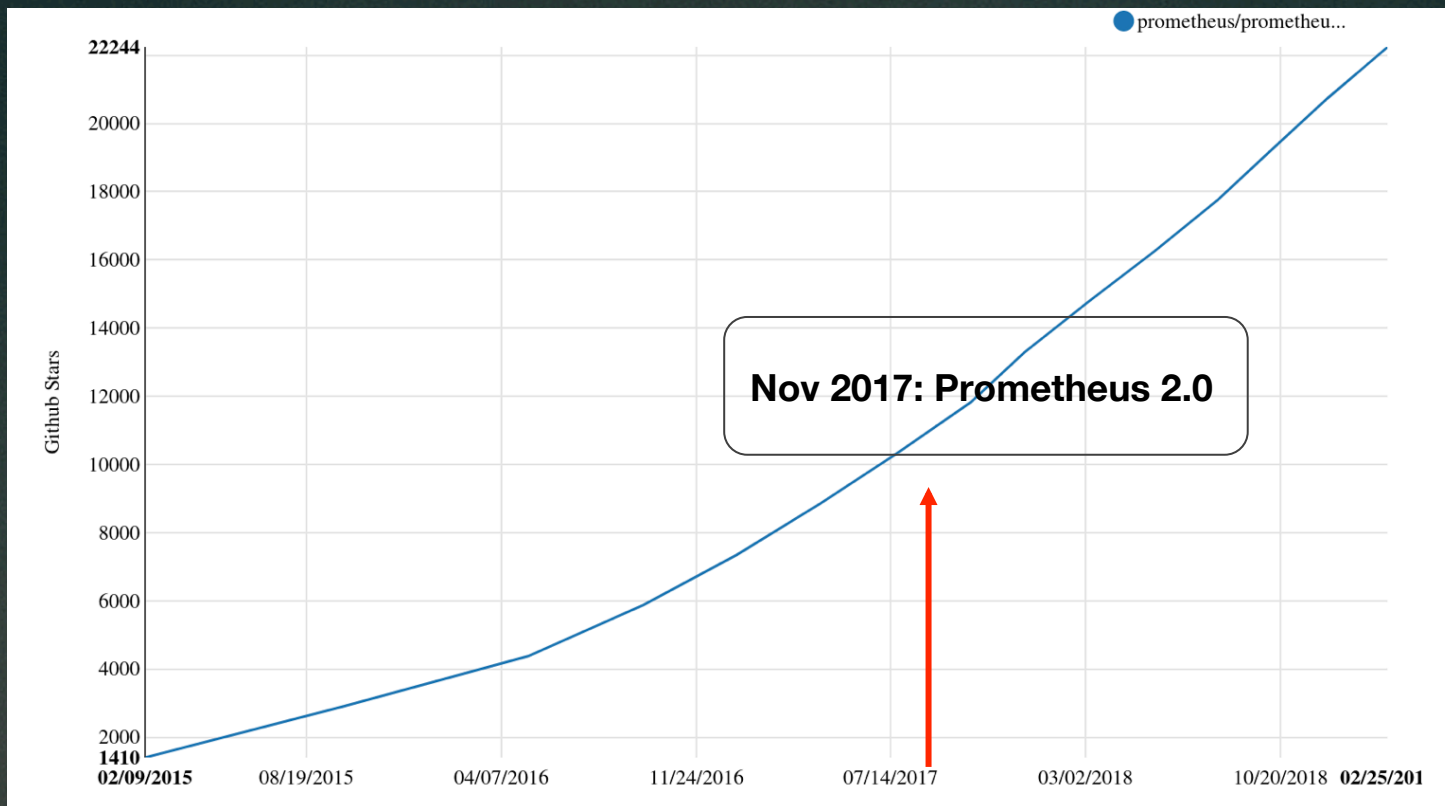
Prometheus: The monitoring system



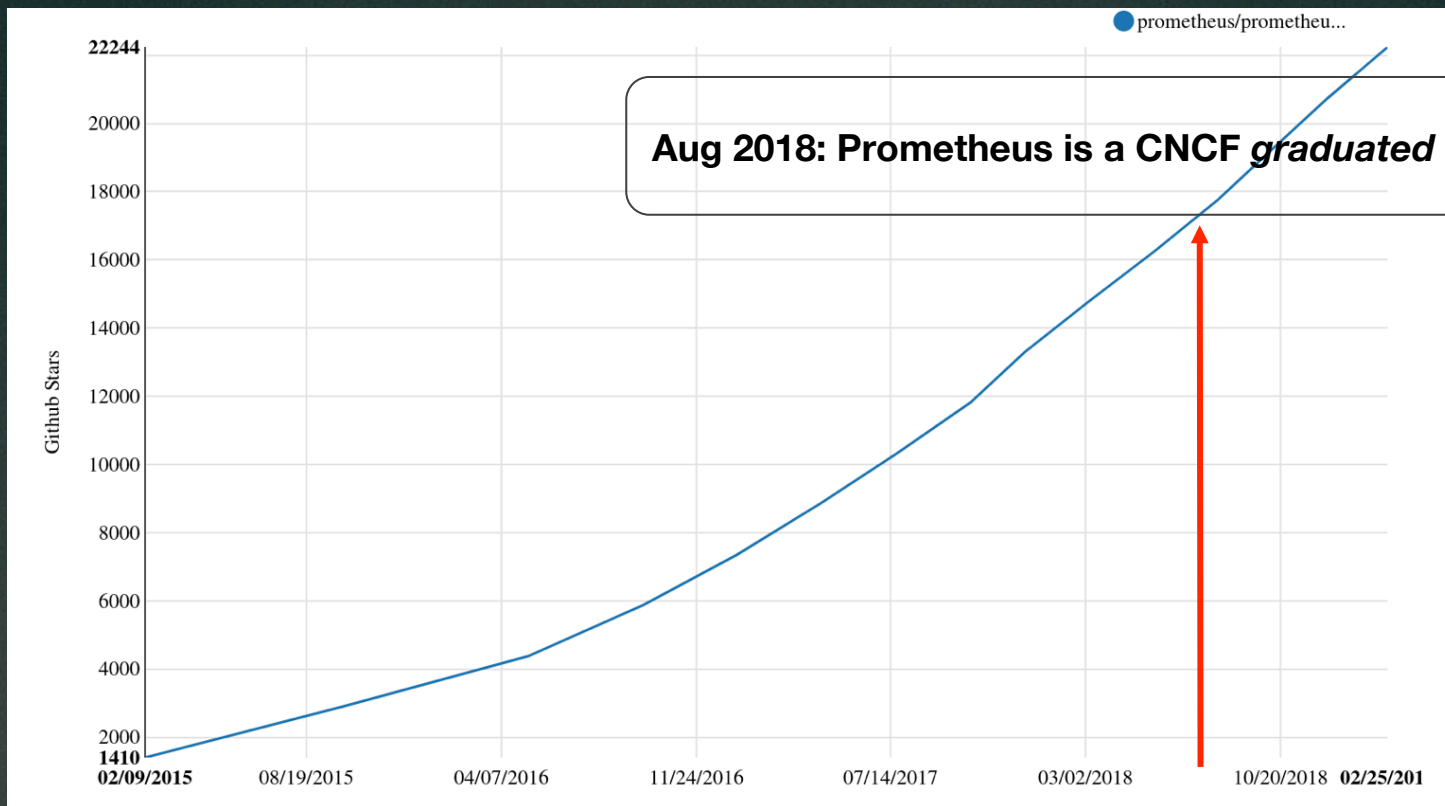
Prometheus: The monitoring system

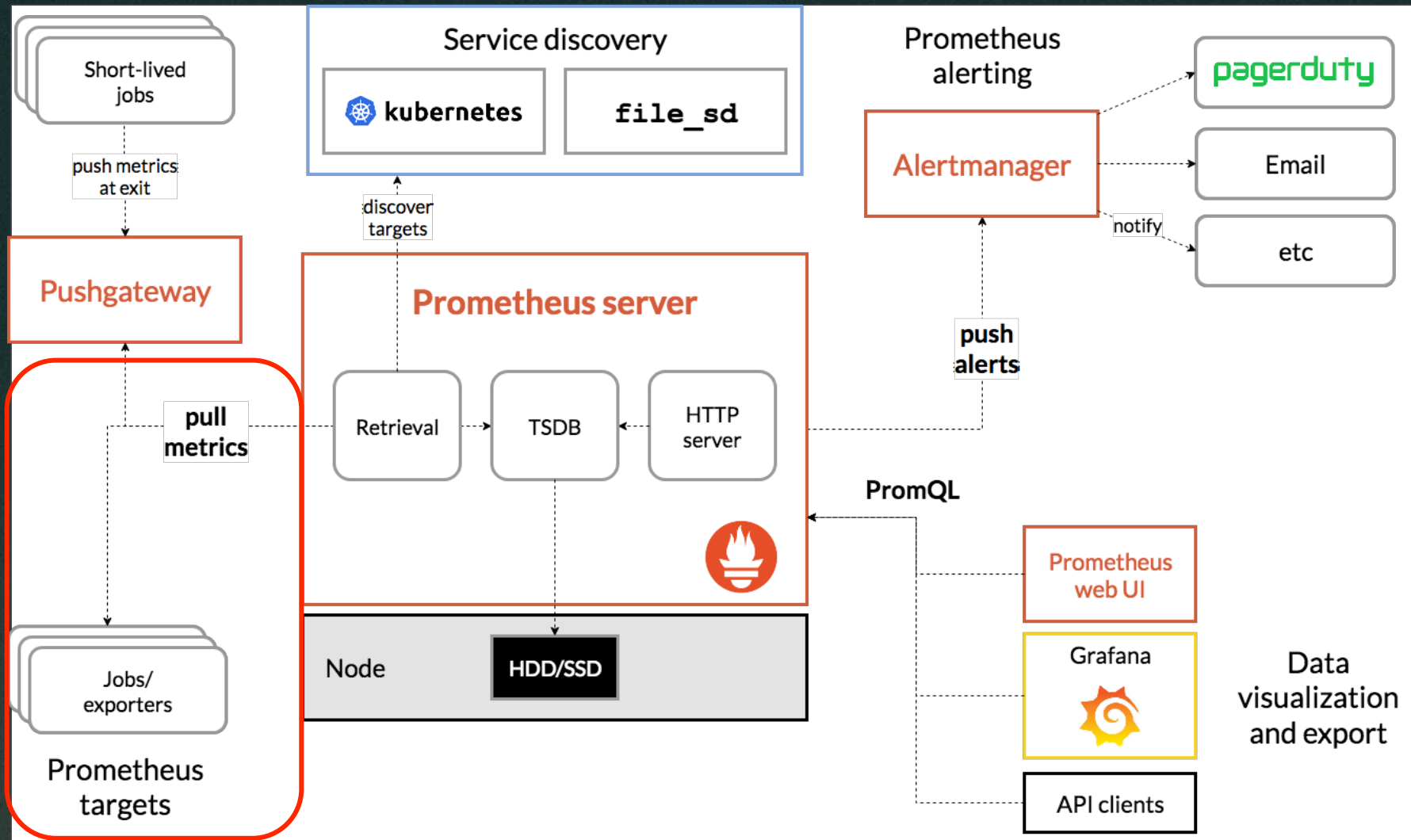


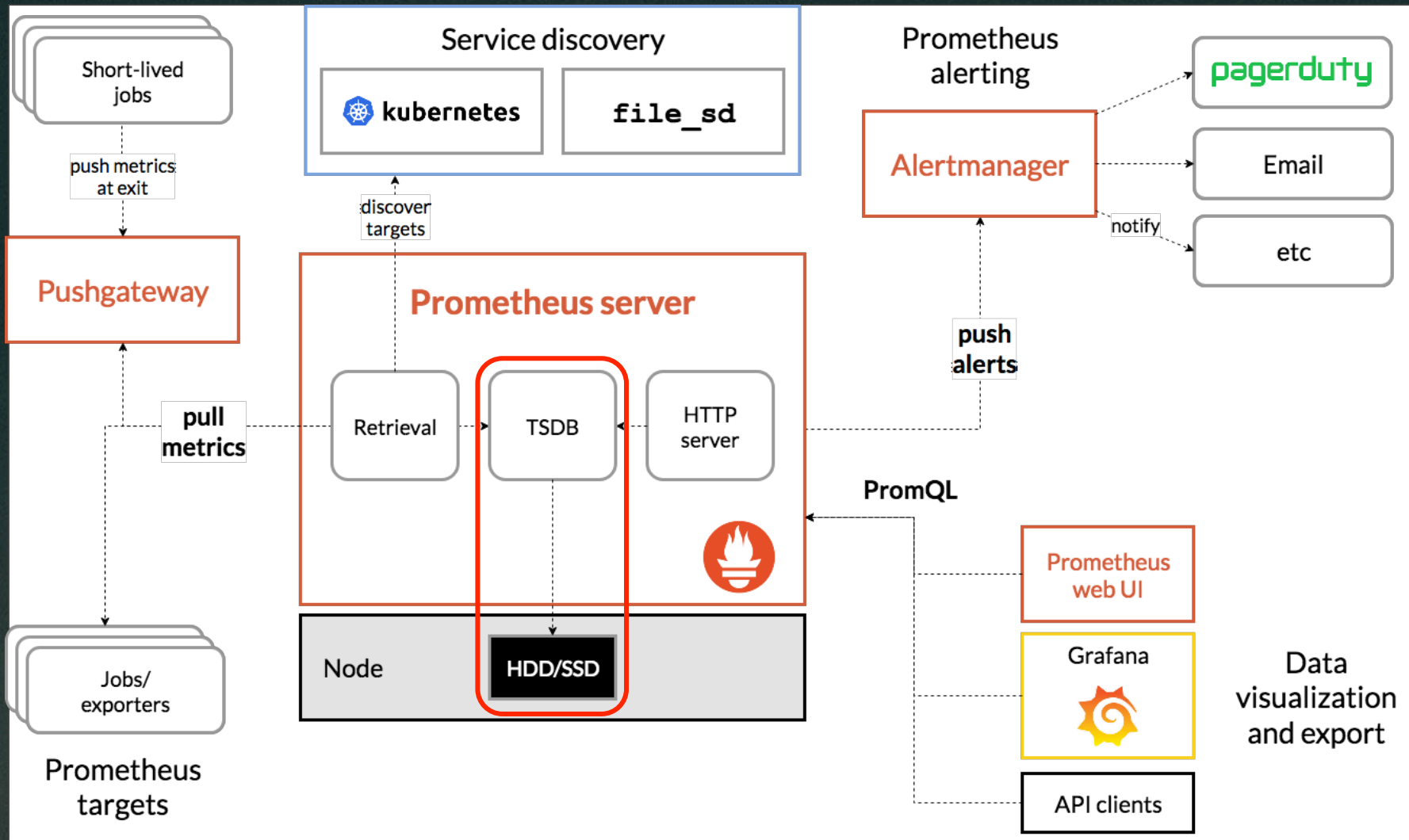
Prometheus: The monitoring system

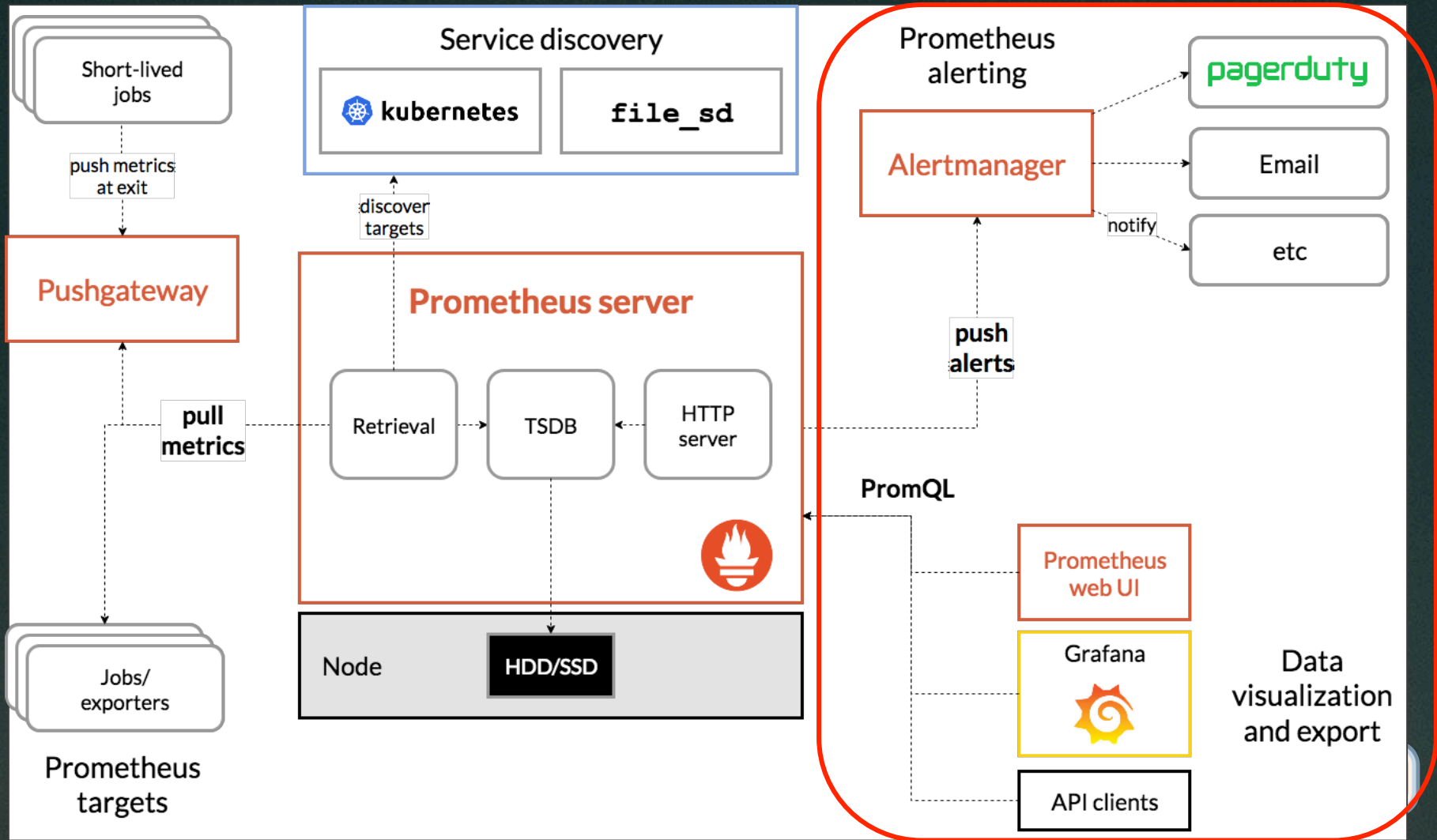


Prometheus: The monitoring system









Major Milestone: Prometheus 2.0

- Completely re-thought a lot of things, re-wrote the storage engine.
- 2-10x improvement across the board.
- Fix-*ing* EVERYTHING.
- 2.0 was **Nov 2017**



Expanding the team

3 more team-members past year!



Release Process

- 1.0:
 - We did it when someone wanted to do it.
 - Worked really well until it didn't.
- No testing, huge set of changes
- 2.4+: Every 6 weeks
- We fixed it, maybe a little too well ;)



Release Process

- 1.0:
 - We did it when someone wanted to do it.
 - Worked really well until it didn't.
- No testing, huge set of changes
- 2.4+: Every 6 weeks
- We fixed it, maybe a little too well ;)



Github: @sipian

Data Integrity

- 1.0: We made no promises about your data
- 1.0: Still *really, really* good!



Data Integrity

- 1.0: We made no promises about your data
- 1.0: Still *really, really* good!
- 2.0: Backups and Restore!



Data Integrity

- 1.0: We made no promises about your data
- 1.0: Still *really, really* good!
- 2.0: Backups and Restore!
- Coming soon to a Prometheus near you: **Backfilling**

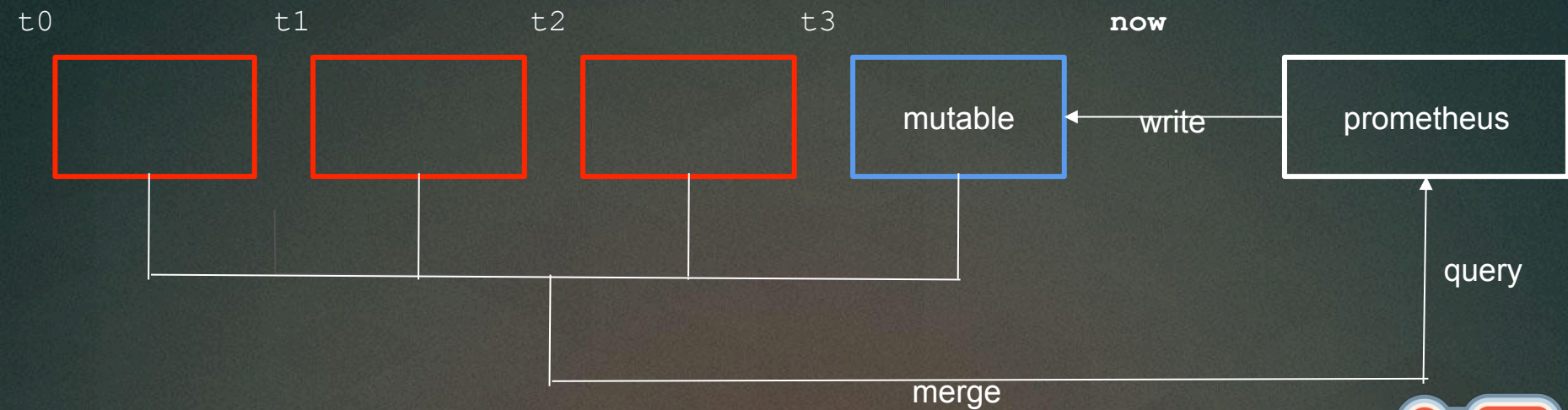


Backfilling

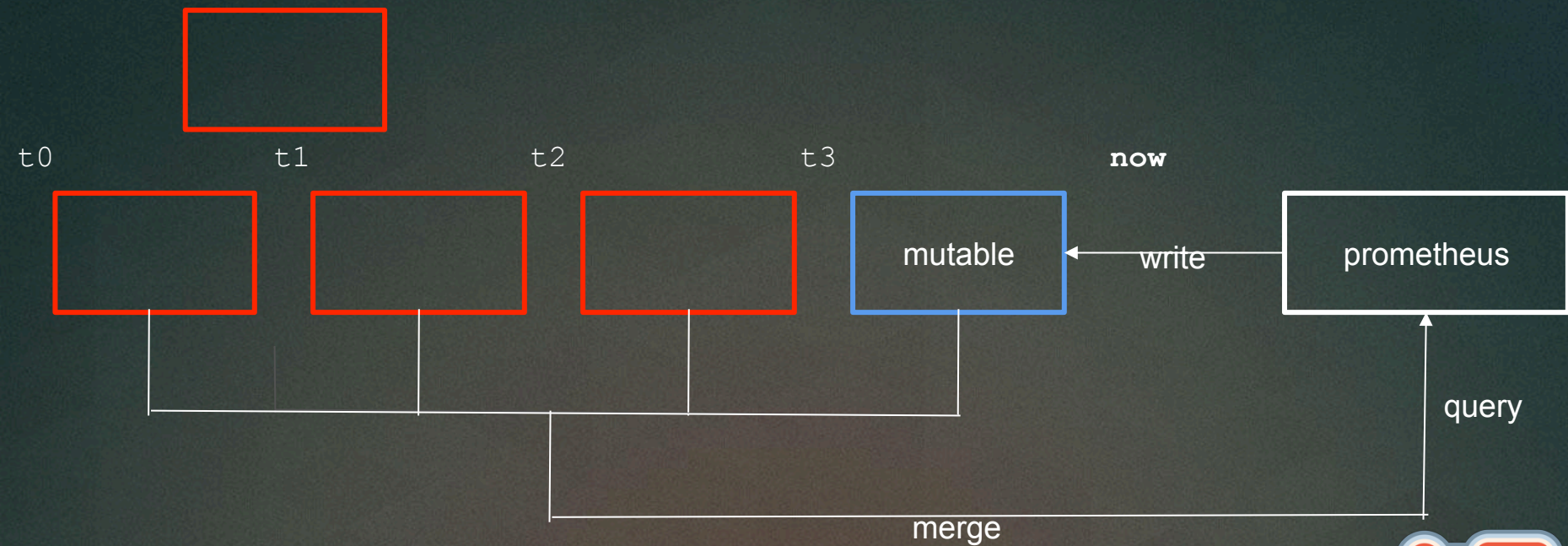
- TONS OF REQUESTS! No good answer.
- In *master*, will be in 2.8



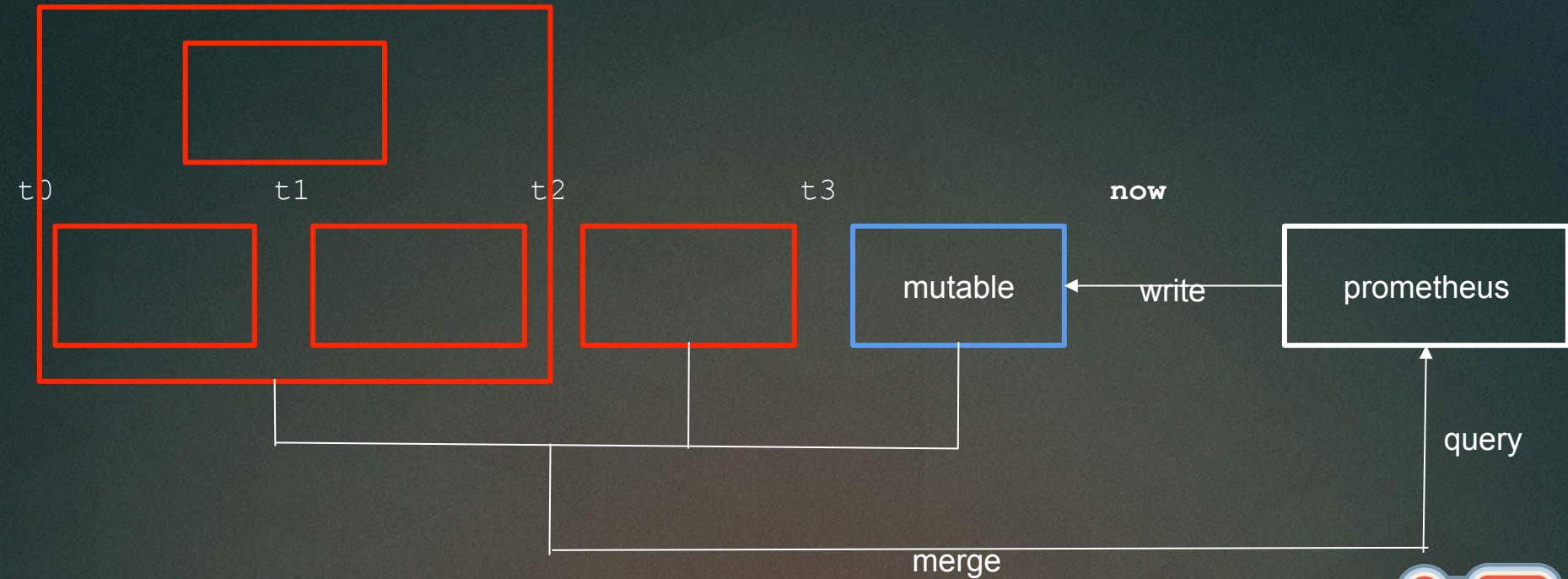
Backfilling



Backfilling



Backfilling



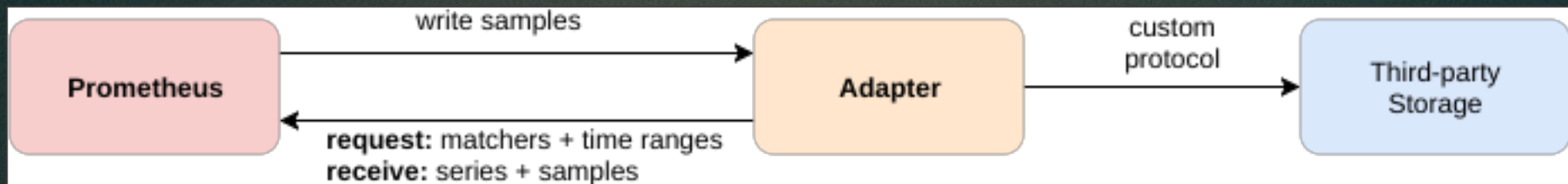
Data Integrity

- Still a single node system



Long Term Storage

Via the *remote_write* and *remote_read*



Remote Write

- snappy-compressed protocol buffer encoding over HTTP
- Uses an in-memory sample buffer in-case remote is down
- OOMs when remote is unreachable
- **Dataloss** when remote is down for more than few mins



Remote Write

Use WAL to gather samples for remote write #4588

Merged tomwilkie merged 3 commits into `prometheus:master` from `cstyan:callum-tail-wal` 14 days ago

 Conversation **204**

 Commits **3**

 Checks **1**

 Files changed **13**



cstyan commented on Sep 7, 2018

Contributor



Example of WAL tailing. Still need to do something with what we read out of the WAL.

cc @csmarchbanks @tomwilkie



Aaaaand more!

- Alerts unit testing
- Metadata API
- New shiny UI (almost there)
- Even MOAR faster
- Subqueries!



WHAT'S NEXT?

Security

- 1.x: Secure it yourself
- Security Audit: Everything is ~great, but give a default way to secure things



Security

- 1.x: Secure it yourself
- Security Audit: Everything is ~great, but give a default way to secure things

TLS and authentication in HTTP serving endpoints

The HTTP serving endpoints in Prometheus, Alertmanager, and the official exporters do not have built-in support for TLS and authentication yet. Adding this support will make it easier for people to deploy Prometheus components securely without requiring a reverse proxy to add those features externally.



BIGGEST THING



BIGGEST THING

Community



Community

- We **changed** the world
- Now, we need to **CONQUER** it!



OpenMetrics

- The Prometheus exposition format is pretty f***** good!
- Let's make it a standard!



OpenMetrics

- The Prometheus exposition format is pretty f***** good!
- Let's make it a standard!



OPEN METRICS



Why? Cuz it's simple and naaice!

```
# HELP go_memstats_stack_inuse_bytes Number of bytes in use by the stack allocator.
# TYPE go_memstats_stack_inuse_bytes gauge
go_memstats_stack_inuse_bytes 2.0086784e+07
# HELP go_memstats_stack_sys_bytes Number of bytes obtained from system for stack allocator.
# TYPE go_memstats_stack_sys_bytes gauge
go_memstats_stack_sys_bytes 2.0086784e+07
# HELP go_memstats_sys_bytes Number of bytes obtained from system.
# TYPE go_memstats_sys_bytes gauge
go_memstats_sys_bytes 6.729791576e+09
# HELP go_threads Number of OS threads created.
# TYPE go_threads gauge
go_threads 54
# HELP net_contrack_dialer_conn_attempted_total Total number of connections attempted by the given dialer a given name.
# TYPE net_contrack_dialer_conn_attempted_total counter
net_contrack_dialer_conn_attempted_total{dialer_name="alertmanager"} 65088
net_contrack_dialer_conn_attempted_total{dialer_name="default"} 0
net_contrack_dialer_conn_attempted_total{dialer_name="default/kube-state-metrics"} 392
net_contrack_dialer_conn_attempted_total{dialer_name="default/kubernetes"} 12
net_contrack_dialer_conn_attempted_total{dialer_name="default/node-exporter"} 456
net_contrack_dialer_conn_attempted_total{dialer_name="kube-system/cadvisor"} 260
net_contrack_dialer_conn_attempted_total{dialer_name="kube-system/kubelet"} 431
net_contrack_dialer_conn_attempted_total{dialer_name="kubernetes-pods"} 30846
net_contrack_dialer_conn_attempted_total{dialer_name="remote_storage"} 203174
```

Why? Cuz it's simple and naaice!

```
# HELP prometheus_sd_kubernetes_events_total The number of Kubernetes events handled.  
# TYPE prometheus_sd_kubernetes_events_total counter  
prometheus_sd_kubernetes_events_total{event="add",role="endpoints"} 3220  
prometheus_sd_kubernetes_events_total{event="add",role="node"} 258  
prometheus_sd_kubernetes_events_total{event="add",role="pod"} 27838  
prometheus_sd_kubernetes_events_total{event="add",role="service"} 3198  
prometheus_sd_kubernetes_events_total{event="delete",role="endpoints"} 8
```



Why?

- Explosion of compatible metric endpoints
- Avoid lock-in
- Kill SNMP ;)

Status

- CNCF Project
- Spec shelled out, now being written
- Prometheus 2.5 has support for it
- The prometheus python client is first client that can expose OM
- Google, Uber will write another reference parser



First commitments

Cloudflare

CNCF at large

GitLab

Google

Grafana

InfluxData

Prometheus!

RobustPerception

SpaceNet

Uber

..... Too many to list ;)



Community: Enable it

- MOAR exporters
- MOAR language support
- <https://github.com/prometheus-community>



THANKS!

Have any questions about our conquest of the world?

