M3 and Prometheus

Monitoring at Planet Scale for Everyone

Berlin, 2019-11-06

Rob Skillington & Łukasz Szczęsny

Who are we?

Rob Skillington
CTO at Chronosphere
@robskillington
M3DB Creator
OpenMetrics Contributor

Łukasz Szczęsny
Snr SRE at Chronosphere
@wybczu
M3 Contributor

Let’s talk

Monitoring an increasing number of things...

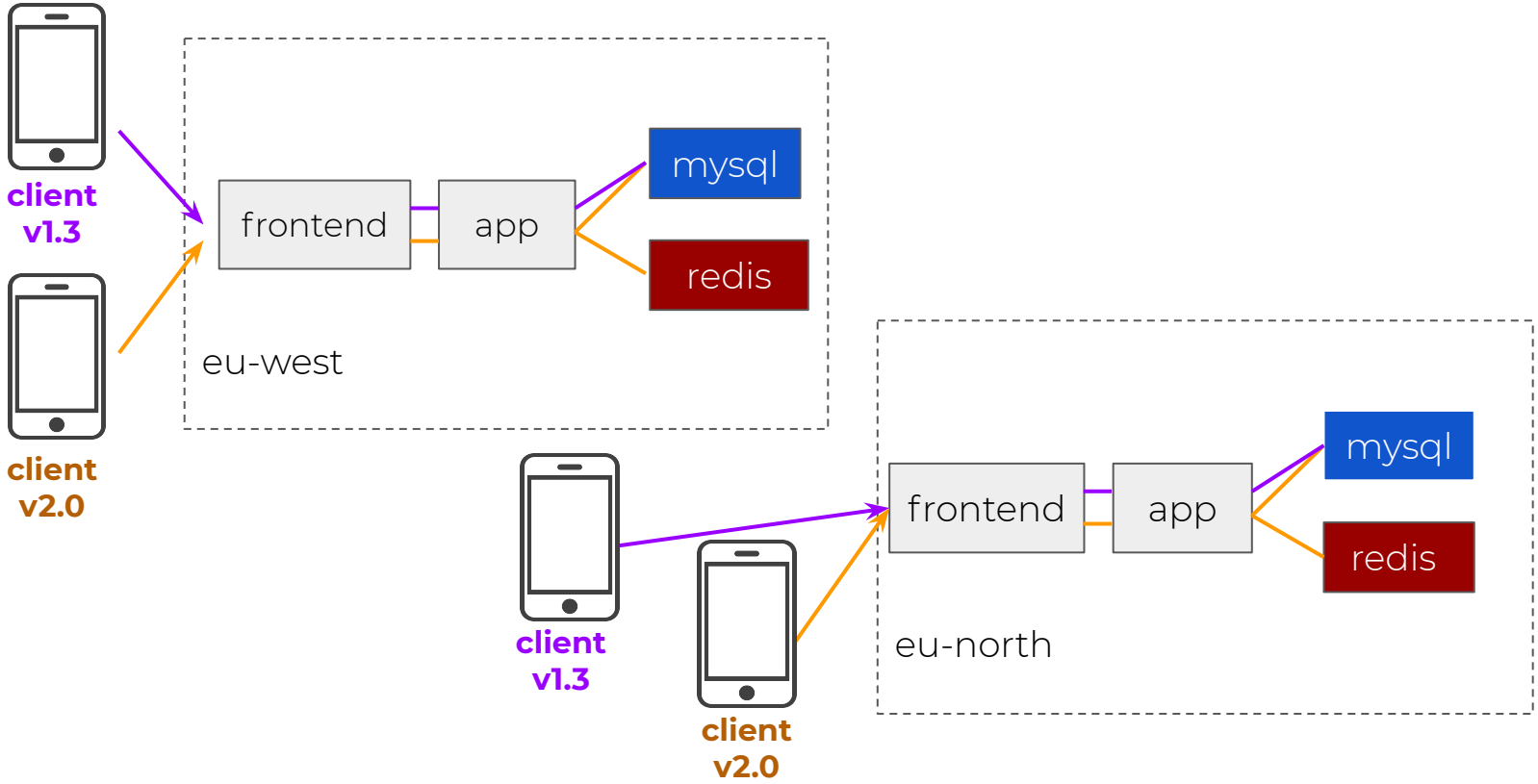
Metrics being used as a platform more than ever...

Operating in many regions or environments...

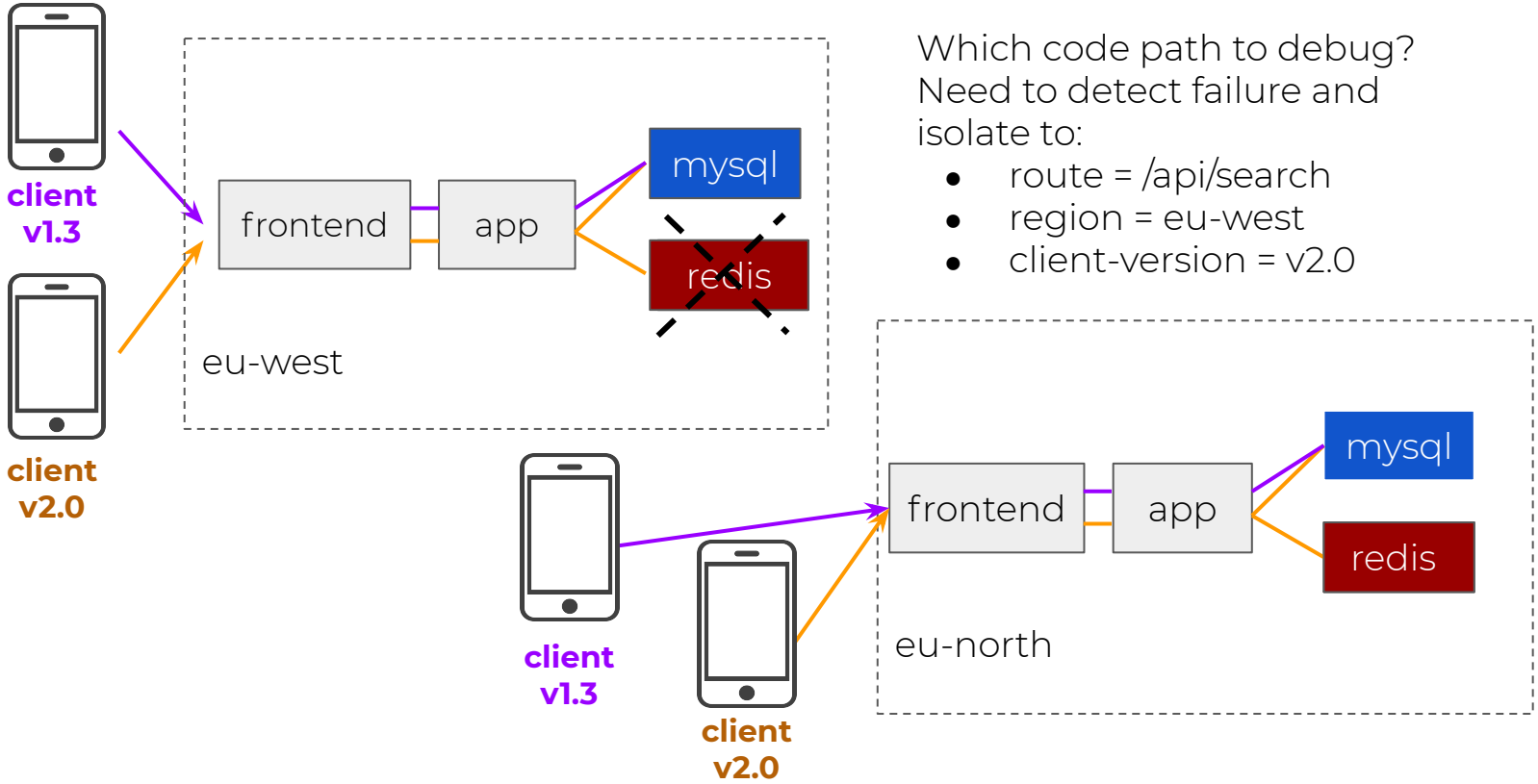
M3 and Prometheus/Graphite...

High dimensionality metrics?

Example system being monitored



Example system being monitored



Which code path to debug?
Need to detect failure and isolate to:

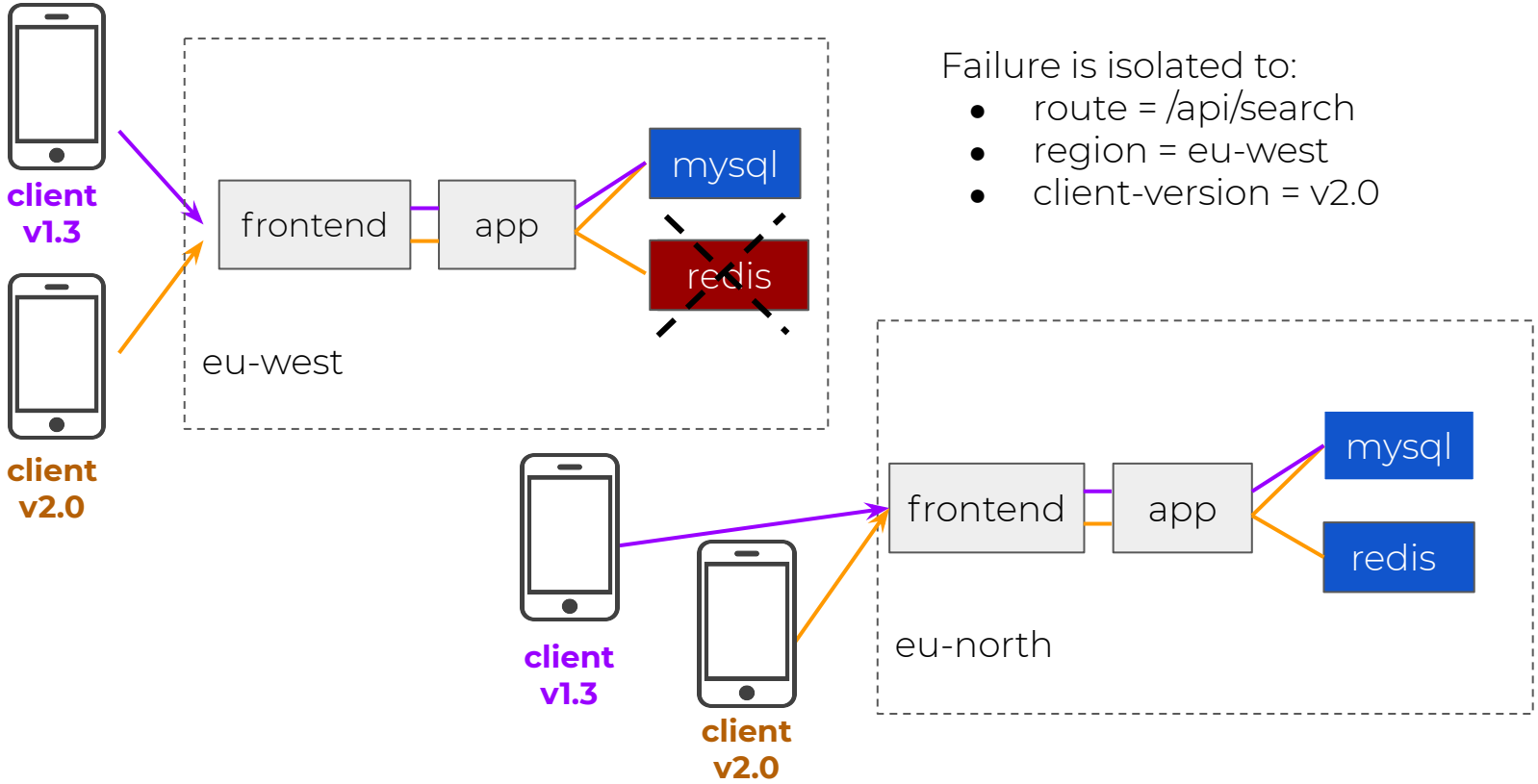
- route = /api/search
- region = eu-west
- client-version = v2.0

Let’s use high dimensionality metrics

- Let’s debug this using HTTP status code delivered by frontends:
  - http_status_code

Route (100?)	**Status Code (5?)**	**Region (12?)**	**Client App Version (40?)**
/api/search	2xx	eu-east	1.3
/api/order	4xx	eu-west	2.0
....	...		

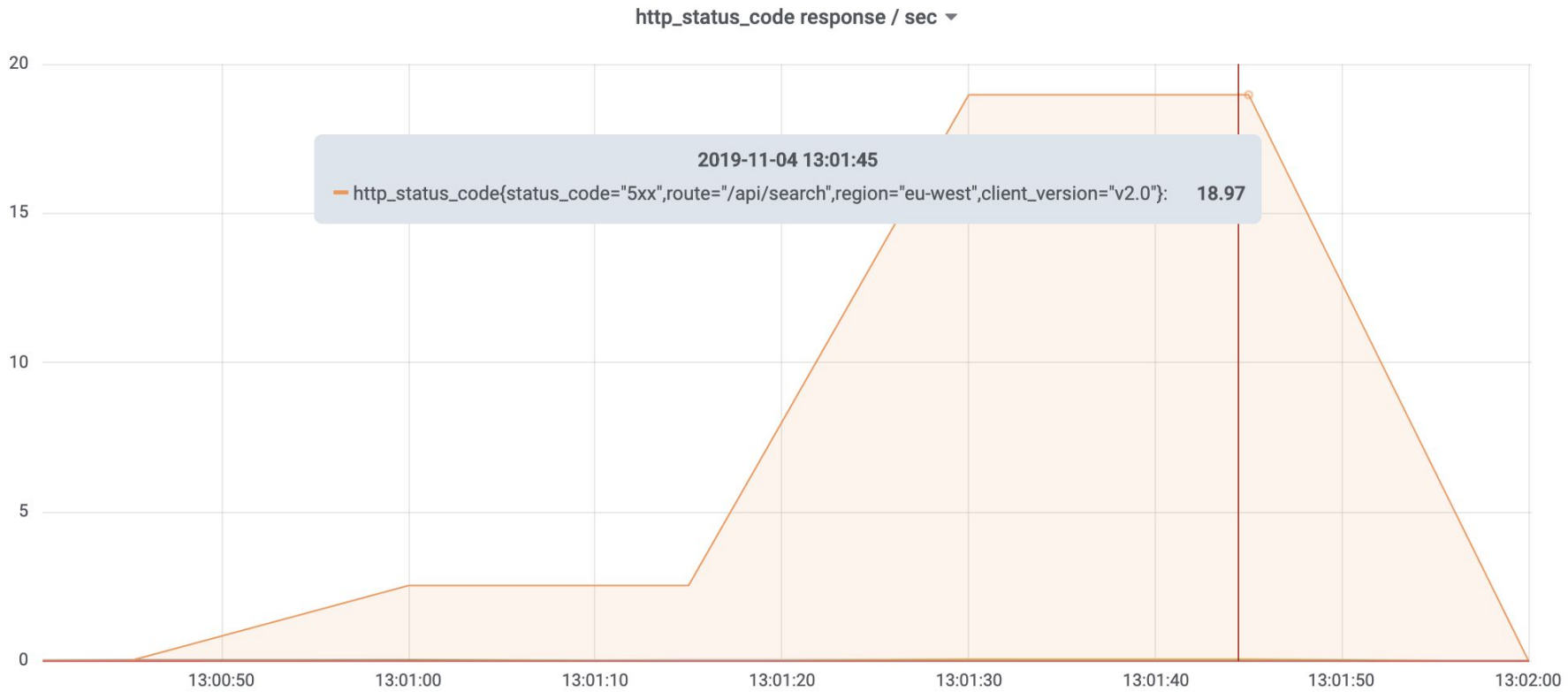
Revisiting this example...



Failure is isolated to:

- route = /api/search
- region = eu-west
- client-version = v2.0

Ideally we would see...



How many time series is that?

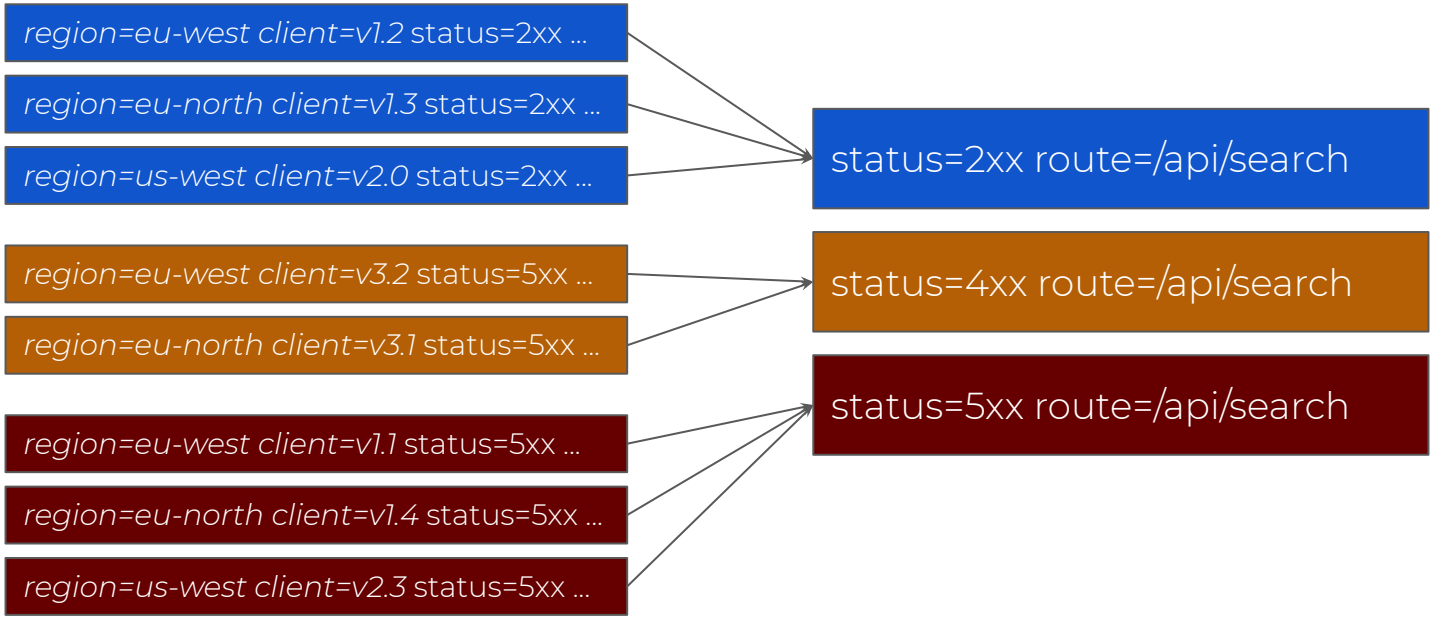
Route (100?)	**Status Code (5?)**	**Region (12?)**	**Client App Version (40?)**
/api/search	2xx	eu-east	1.3
/api/order	4xx	eu-west	2.0
....	...		

100 routes * 5 status codes * 12 regions * 40 client versions

= 240,000 unique time series

Partial-solution #1

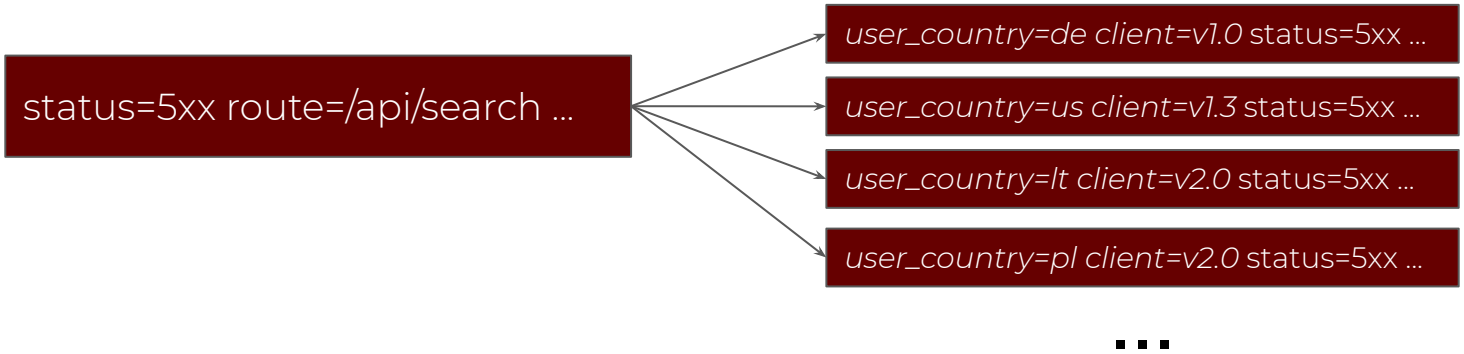
You can roll up metrics to make viewing fast



For drill down and high granular alerting

240k time series, expensive but not too bad..? However add any other dimensions and it gets out of control (any multiplier on 240k explodes to millions quickly)

e.g. Unique country code user = 249



What is Prometheus? What is M3?

First built at SoundCloud (began 2012, open source in 2014)

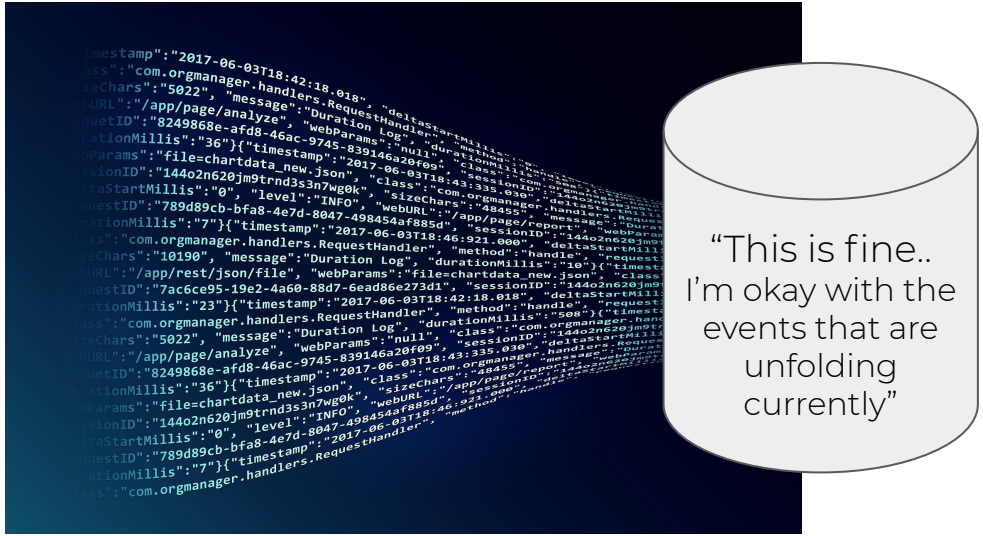
- An open source monitoring system and time series database.
- All-in-one single node monitoring solution using metrics.

Built at Uber to scale monitoring horizontally and cost effective (began 2015, open source in 2018)

- Distributed monitoring system and time series database, compatible as remote storage for Prometheus.

Ok great, but what do I need?

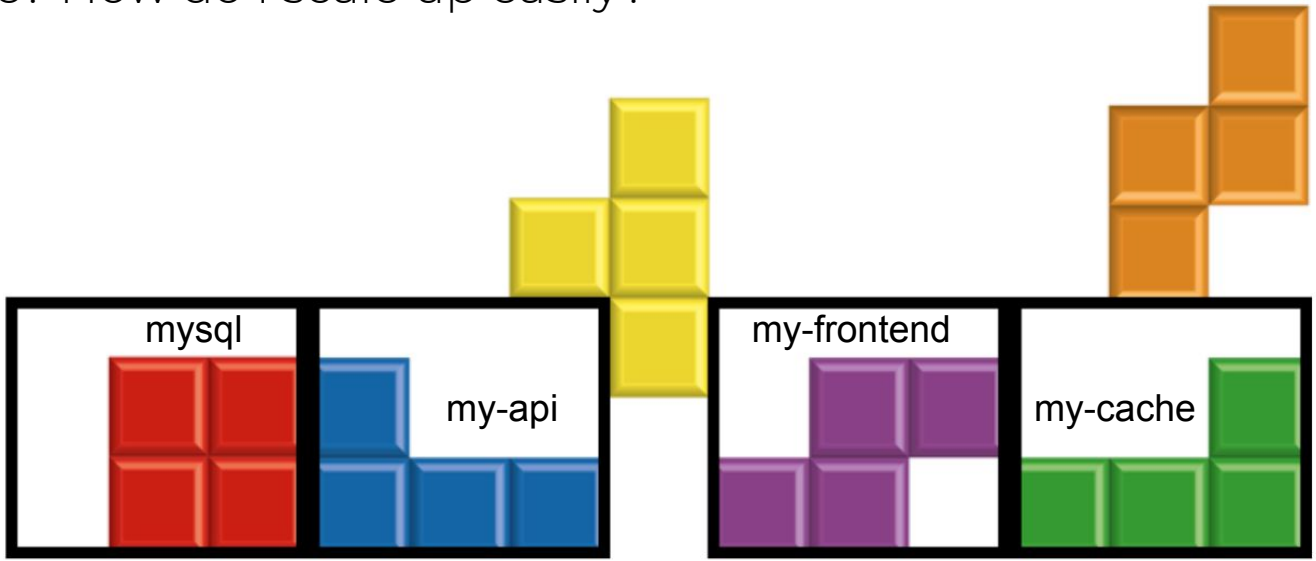
A single Prometheus instance can hold a reasonable amount of data (and you should always get started using Prometheus)



Ok great, but what do I need?

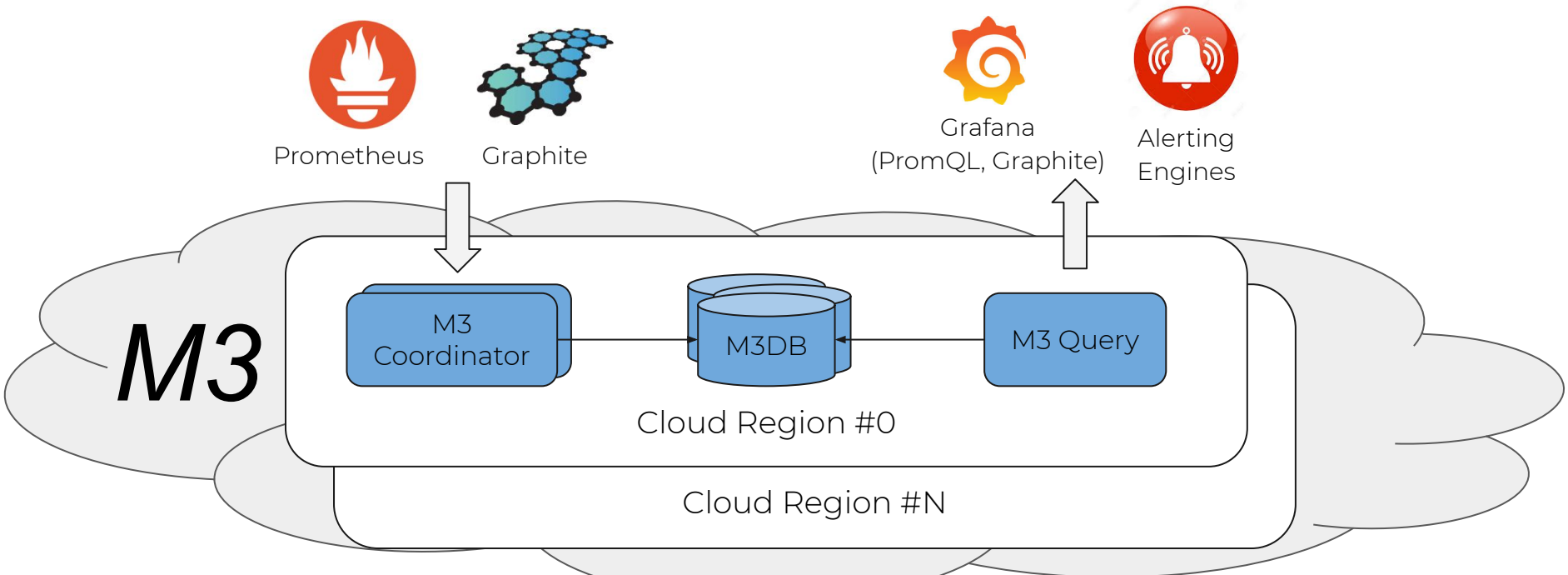
Ok great, but what do I need?

Can I fit a service's high cardinality metrics into an existing Prometheus instance? How do I scale up easily?



So what is M3 and how does it help?

Horizontally scalable platform that supports multiple metric formats



Why M3

1. Suitable for many scenarios

2. Scalable to billions of metrics

3. Focus on simple operation

1. Suitable for many scenarios

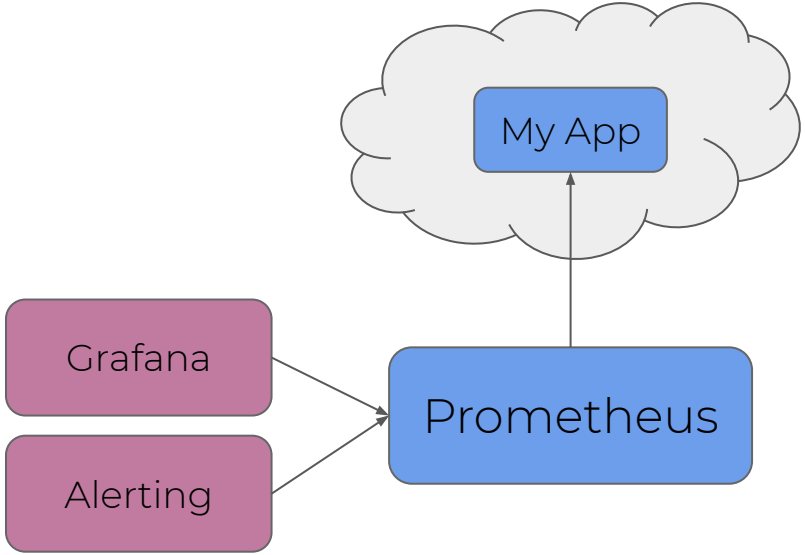
Cloud Native, Kubernetes or On Prem, Multi-Region, Prometheus and Graphite compatible

1. Suitable for many scenarios

M3 and Prometheus

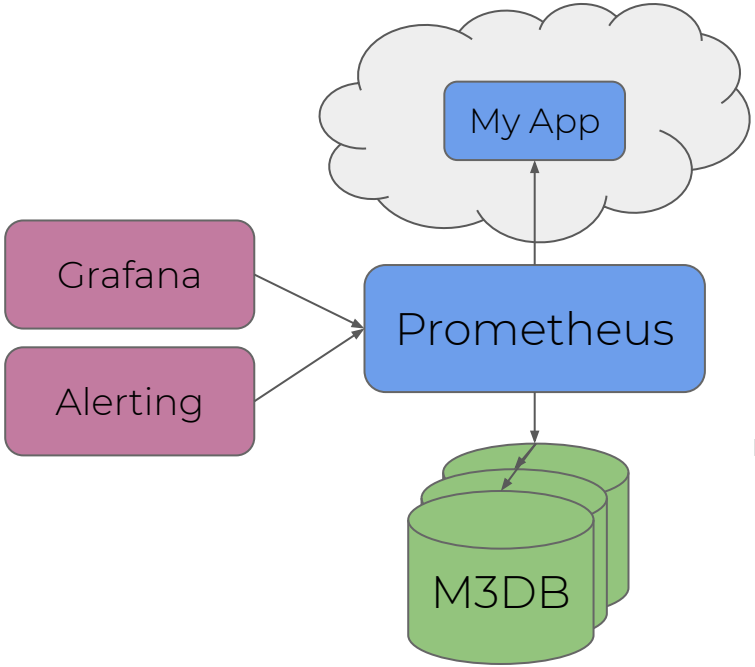
- Store metrics for weeks, months or years
- Store metrics at different retention based on mapping rules (e.g. app:nginx endpoints:/api*)
- Scale up storage just by adding more nodes

Prometheus



DEMO

Prometheus



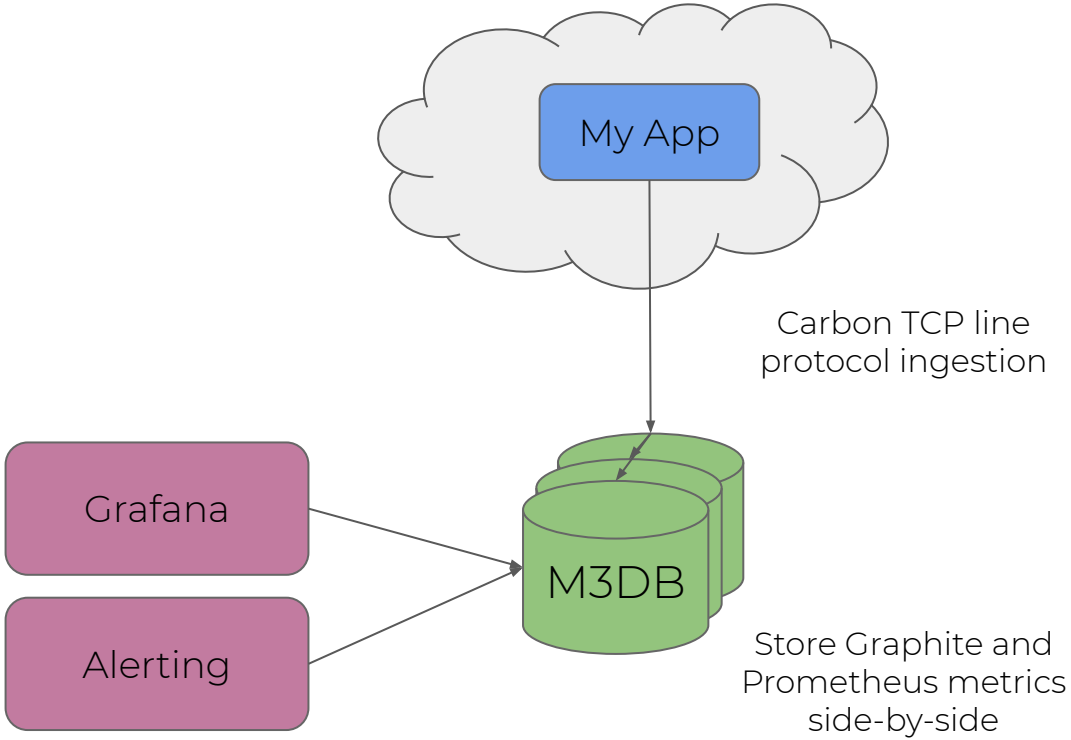
Prometheus remote read and write to M3DB

1. Suitable for many scenarios

M3 and Graphite

- Ingest Carbon TCP protocol
- Support for Graphite query API

Graphite


My App
Carbon TCP line protocol ingestion
Grafana
Alerting
M3DB
Store Graphite and Prometheus metrics side-by-side

2. Scalable to billions of metrics

2. Scalable to billions of metrics

M3 at scale

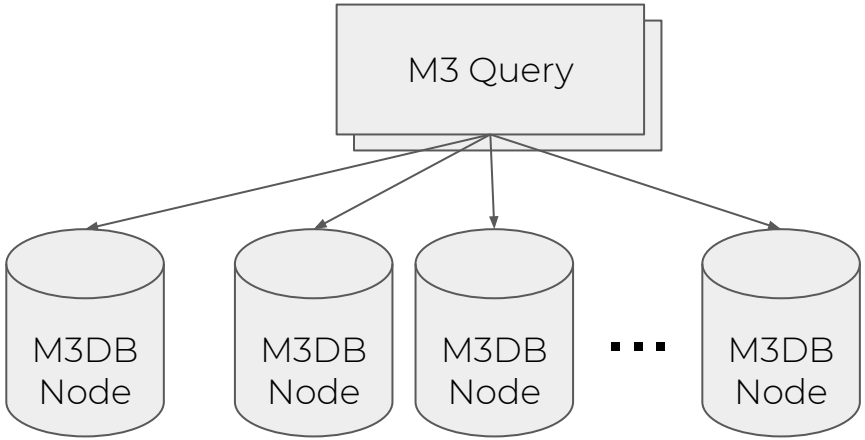
- Collects metrics for 1000s of applications
- No onboarding to monitoring or provisioning of servers (just add storage nodes as required)

2. Scalable to billions of metrics

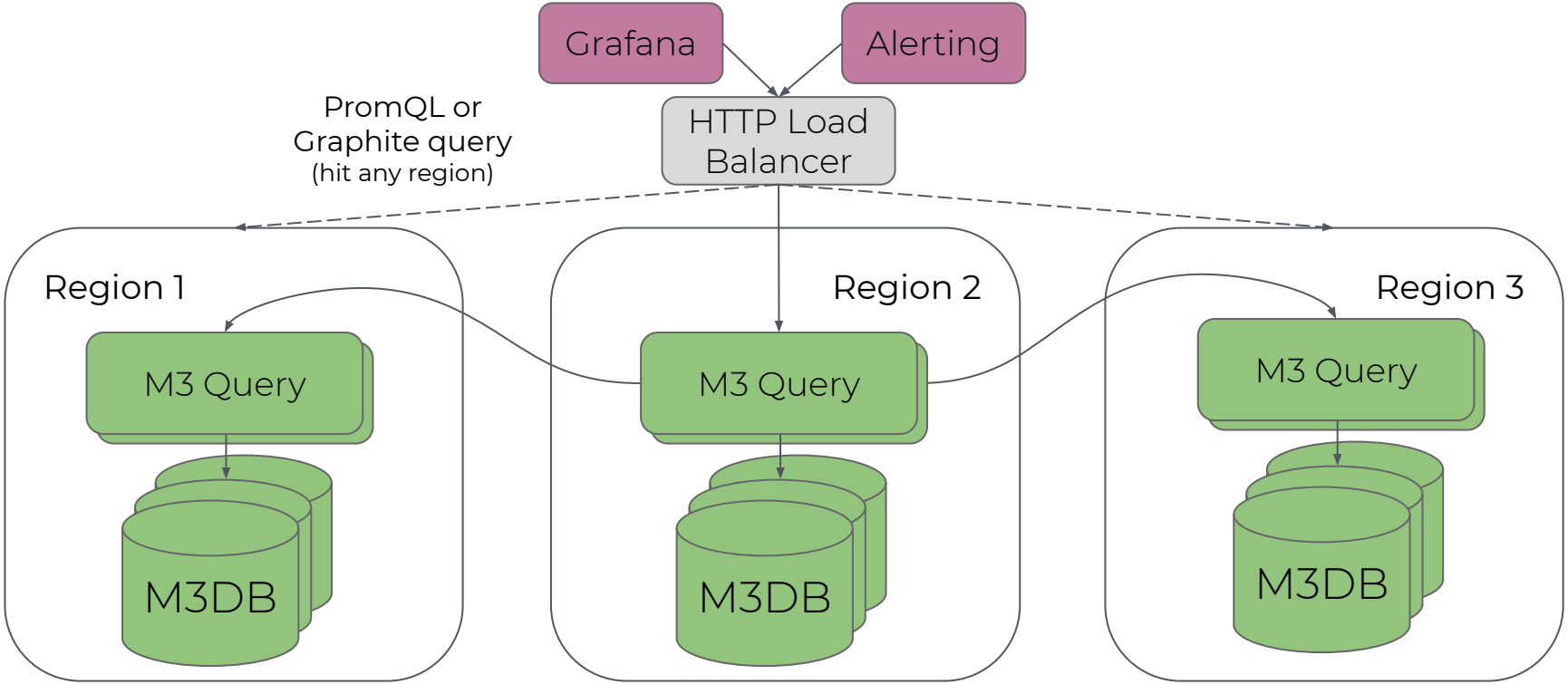
Reverse index uses FST segments, like ElasticSearch with Apache Lucene. It can regexp over billions of metric names and dimensions, unlike other solutions out there.

Each storage node

Find metrics matching query and return in parallel knowing exactly where to extract series data from local store.



Global view with region-local storage



Multi-Region

2. Scalable to billions of metrics

Architected for Reliability and Scale

- Global metrics collection and query
- Low inter-region network bandwidth, data always kept in region
- Replication across Availability Zones within a region as soon as metric collected

3. Focus on simple operation

3. Focus on simple operation

- M3 can be deployed on premise without any dependencies - it’s easy to get started.
  - One binary and a YAML configuration file
  - Can be easily deployed using your favourite config management tool
- Clustered version is open source
  - HA setup is pretty straightforward
  - Scaling a cluster used to require a lot of manual work

3. Focus on simple operation

- M3 runs on Kubernetes and the M3DB k8s operator can manage the cluster for you!

See more at https://github.com/m3db/m3db-operator

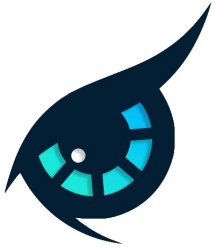
Why M3

1. Suitable for many scenarios ✓

2. Scalable to billions of metrics ✓

3. Focus on simple operation ✓

Come say hi!



chronosphere

Thank you and Q&A

M3 GitHub Monorepo (Apache 2 licensed):
https://github.com/m3db/m3

M3 Slack:
https://bit.ly/m3slack

Chronosphere:
https://chronosphere.io

Twitter:
https://twitter.com/chronosphereio

chronosphere

M3 Links and References

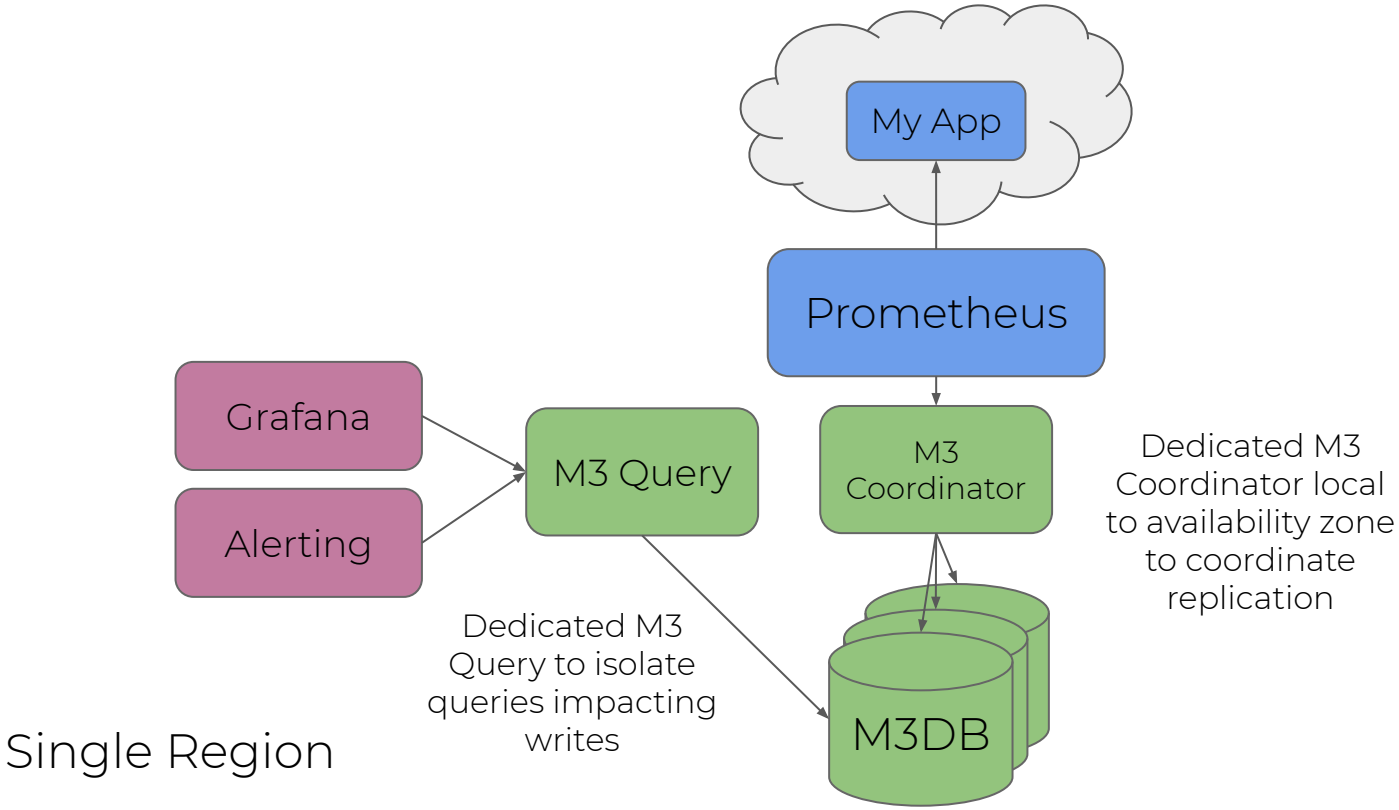
License:
Apache 2

Website:
https://www.m3db.io

Docs:
https://docs.m3db.io

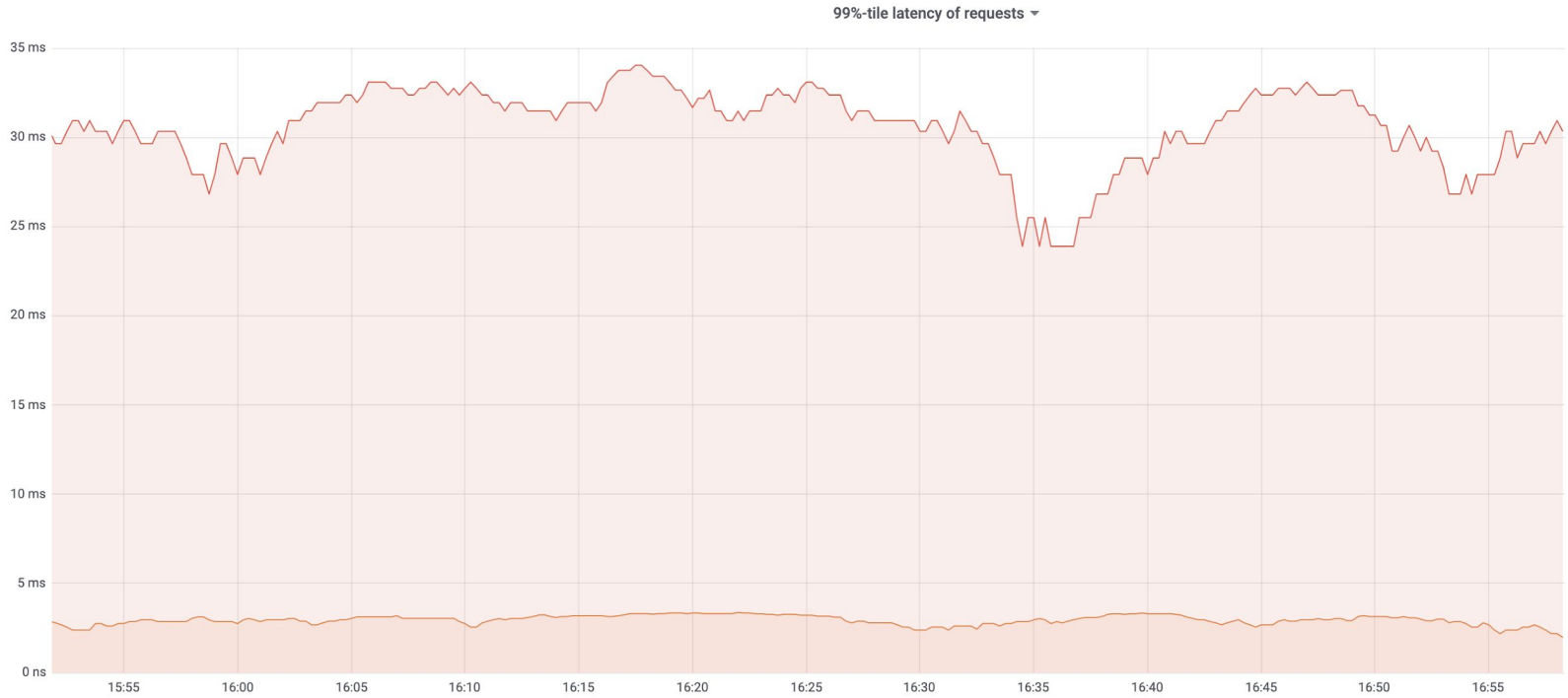
Mailing list:
https://groups.google.com/forum/#!forum/m3db

M3 and Prometheus with read/write isolation



What is Prometheus and M3 used for?

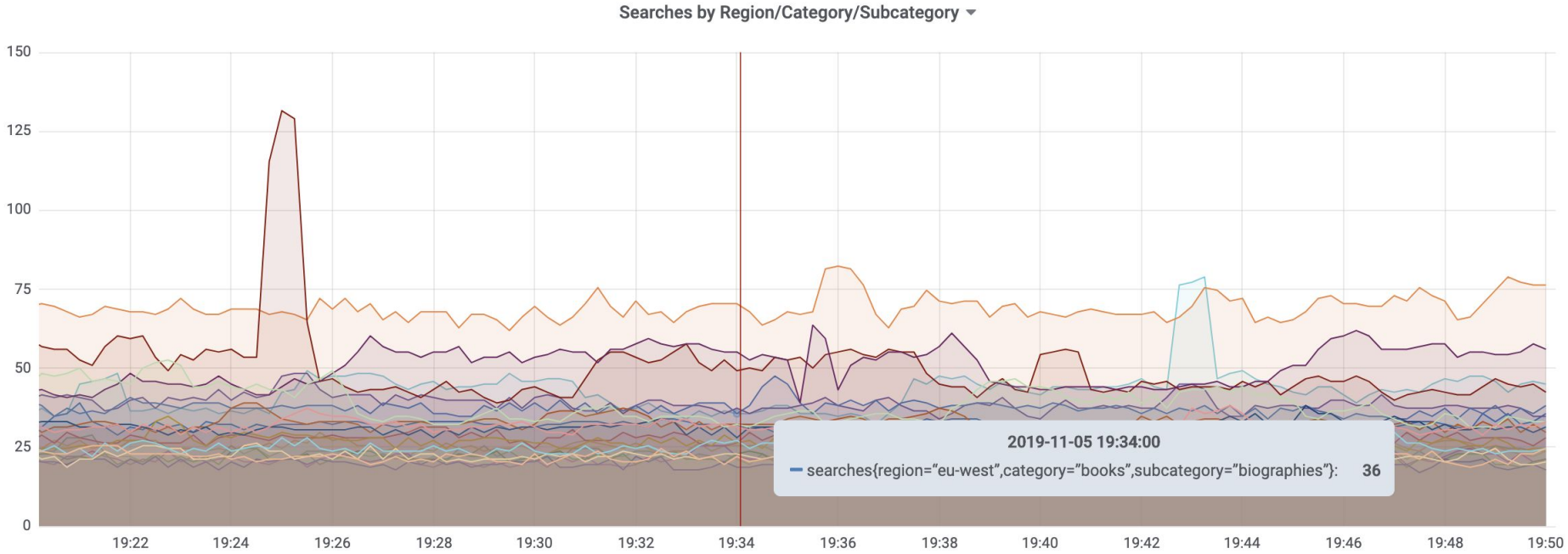
Real time alerting of application metrics



What is Prometheus and M3 used for?

Tracking business metrics (e.g., searches for “books” with category “biographies” in a region):

```
m.Tagged(Tags{region="eu-west",category="books",subcategory="biographies"}).Counter("searches").Inc(1)
```



What is Prometheus and M3 used for?

Infrastructure metrics such as network routing and datacenter health