M3 with Prometheus

August 9, 2018 | Promcon 2018

Nikunj Aggarwal

Agenda

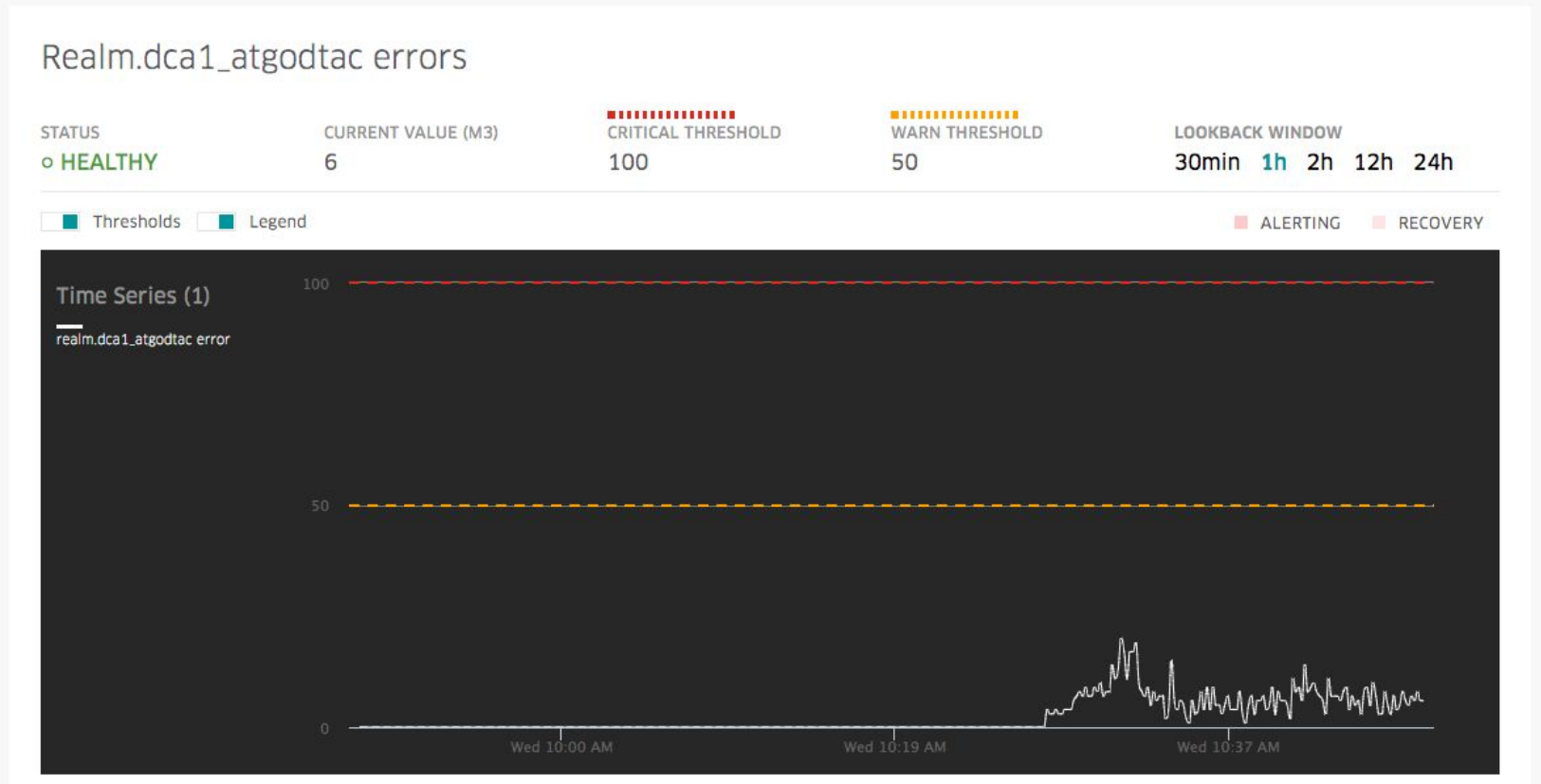
- 30,000 foot view
- Why is serving metrics complex
- Long term strategy

30,000 foot view

30,000 foot view

UBER

30,000 foot view

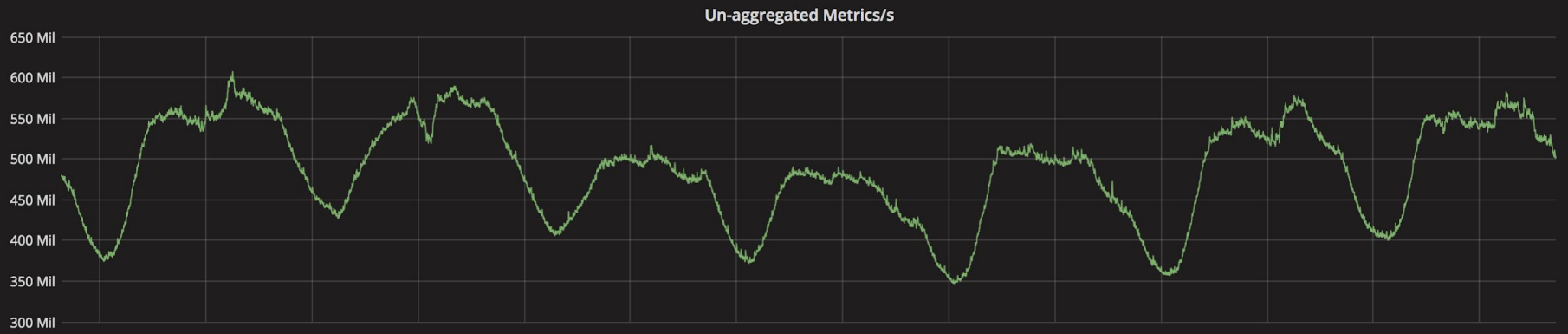


UBER

Why is serving metrics complex

UBER

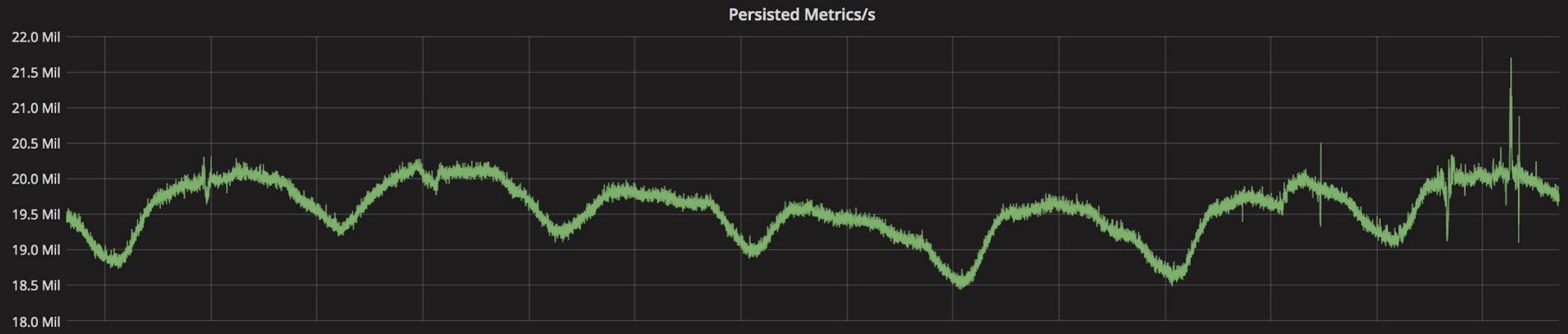
Scale - Ingress



400-600M Pre-aggregated Metrics/s

(~130Gbits/sec)

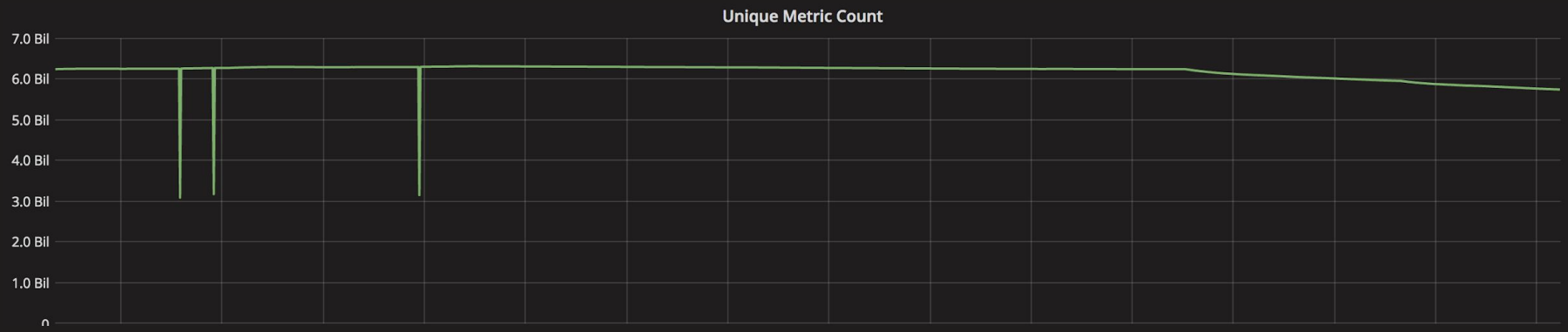
Scale - Ingress



~20M Metrics Stored/s

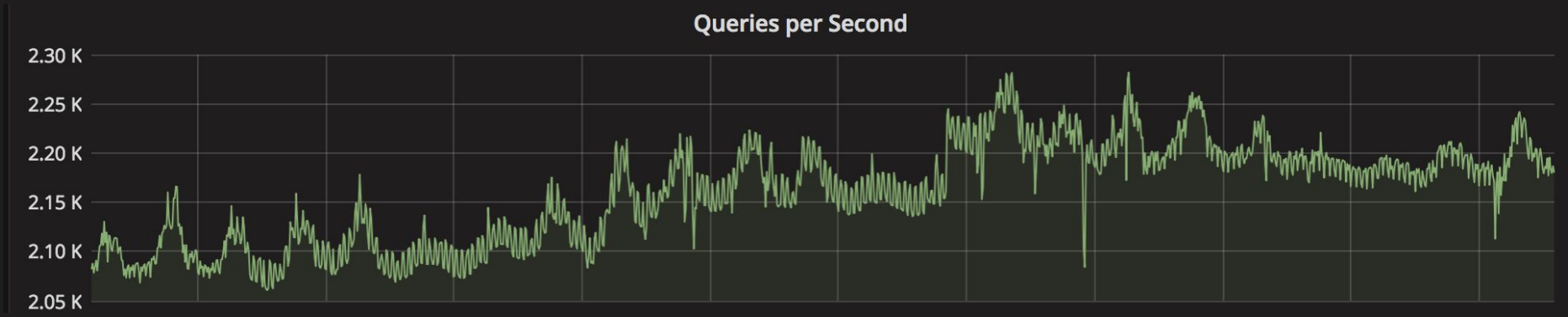
(~50Gbits/sec)

Scale - Ingress



~ **6B** Unique Metric IDs

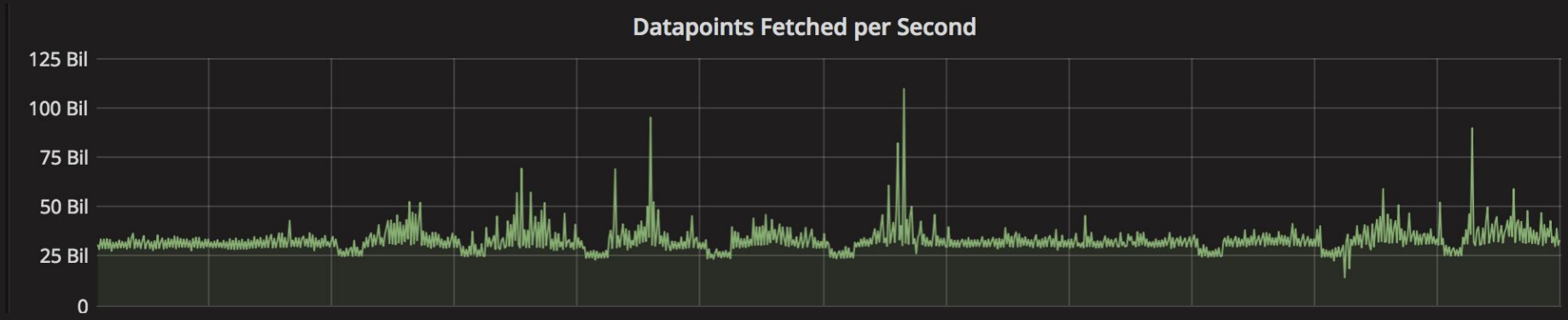
Scale - Egress



~ **2.2K** Queries per second

(9K Grafana Dashboards, 150K Realtime Alerts)

Scale - Egress



~ 30B Datapoints per second

(~20Gbits/sec)

Constantly growing

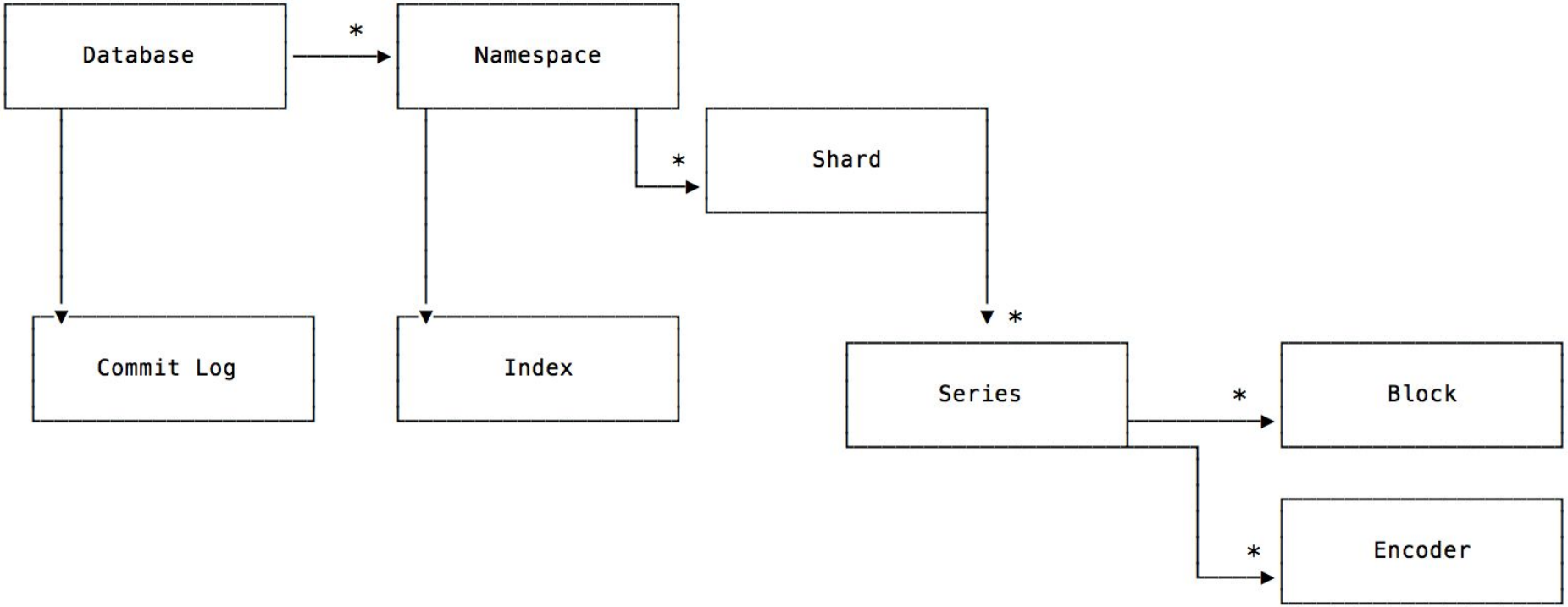
- Persisted Metrics: 20% uptick in the last quarter
- Unique IDs: 50% uptick in the last half year
- QPS: 100% uptick in the last year
- Ingress Traffic: 90,000% in the last 3 years

M3DB

A open source distributed time series database

- Store arbitrary timestamp precision datapoints at any **resolution** for any **retention**
- Optimized file-system storage with **no need for compactions**
- **Replicated** with zone/rack aware layout and configurable replication factor
- **Strongly consistent** cluster membership backed by etcd
- Fast streaming for node add/replace/remove by selecting best peer for a series while also repairing any mismatching series at time of streaming

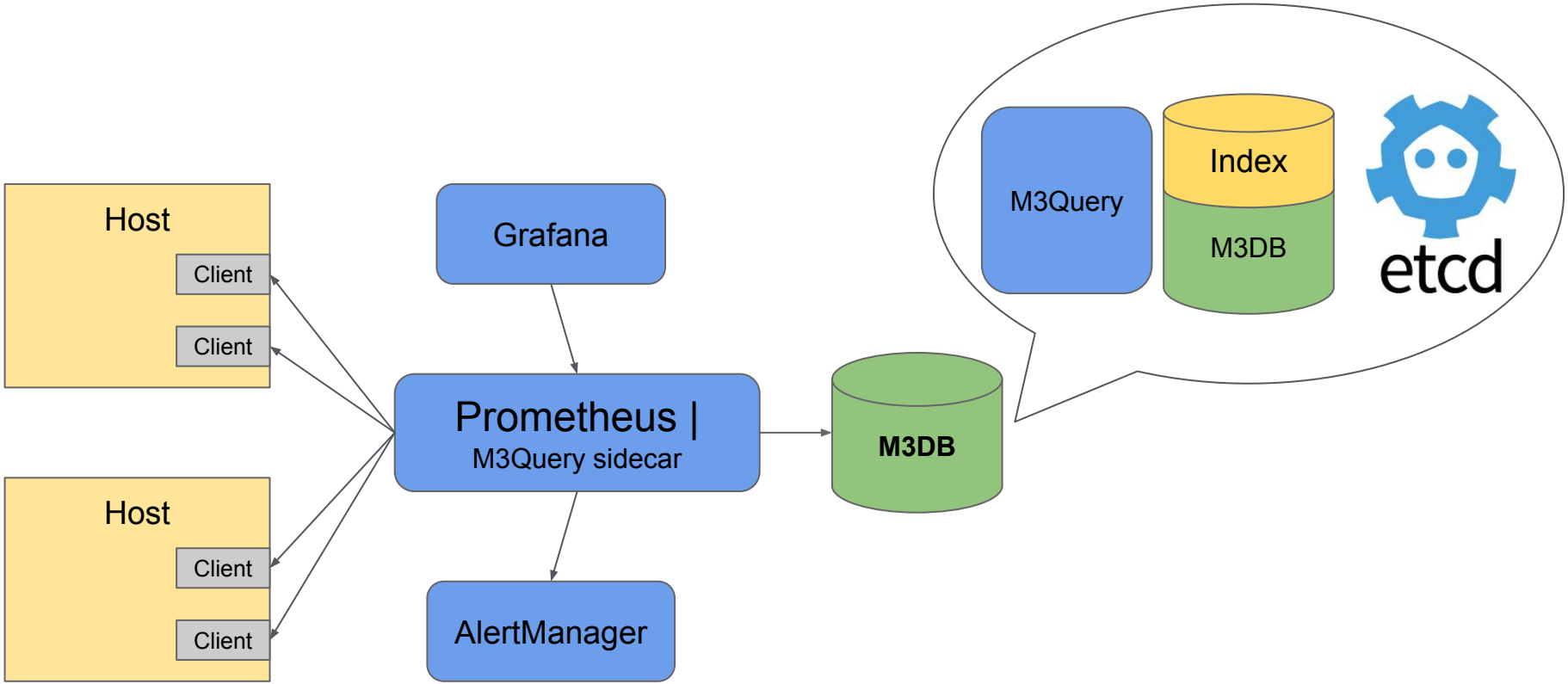
M3DB Logical Constructs



Open source strategy

UBER

OSS H1 2018


Host
Client
Client
Host
Client
Client
Grafana
Prometheus |
M3Query sidecar
AlertManager
M3DB
M3Query
Index
M3DB
etcd

M3 Query Engine

- Remove time series limit
  - Improving memory efficiency
  - Enabling streaming back to client

- Reduce latency
  - Keep data compressed until function application step
  - Concurrently operate on blocks of data

Questions ?

UBER

Appendix

- Eng blog
- M3 repo