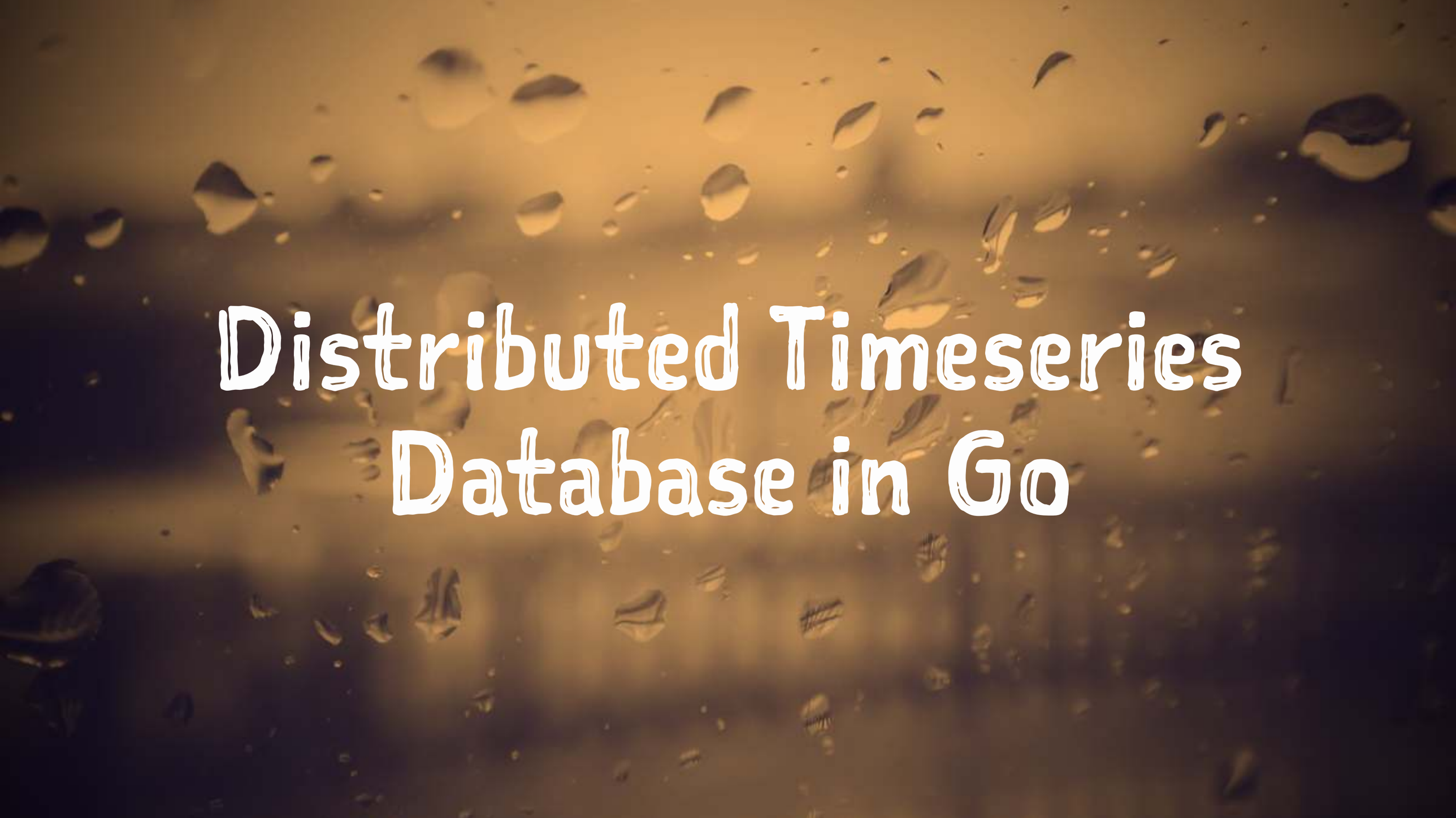

Distributed Timeseries
Database in Go


Go is the future of NoSQL/NewSQL

Databases written in Go

- Prometheus
- CockroachDb
- InfluxDb
- Dgraph
- EtcD
- Consuld


Talk about architecture

What is a timeseries

Use cases for timeseries

- Stocks
- Monitoring
- IOT

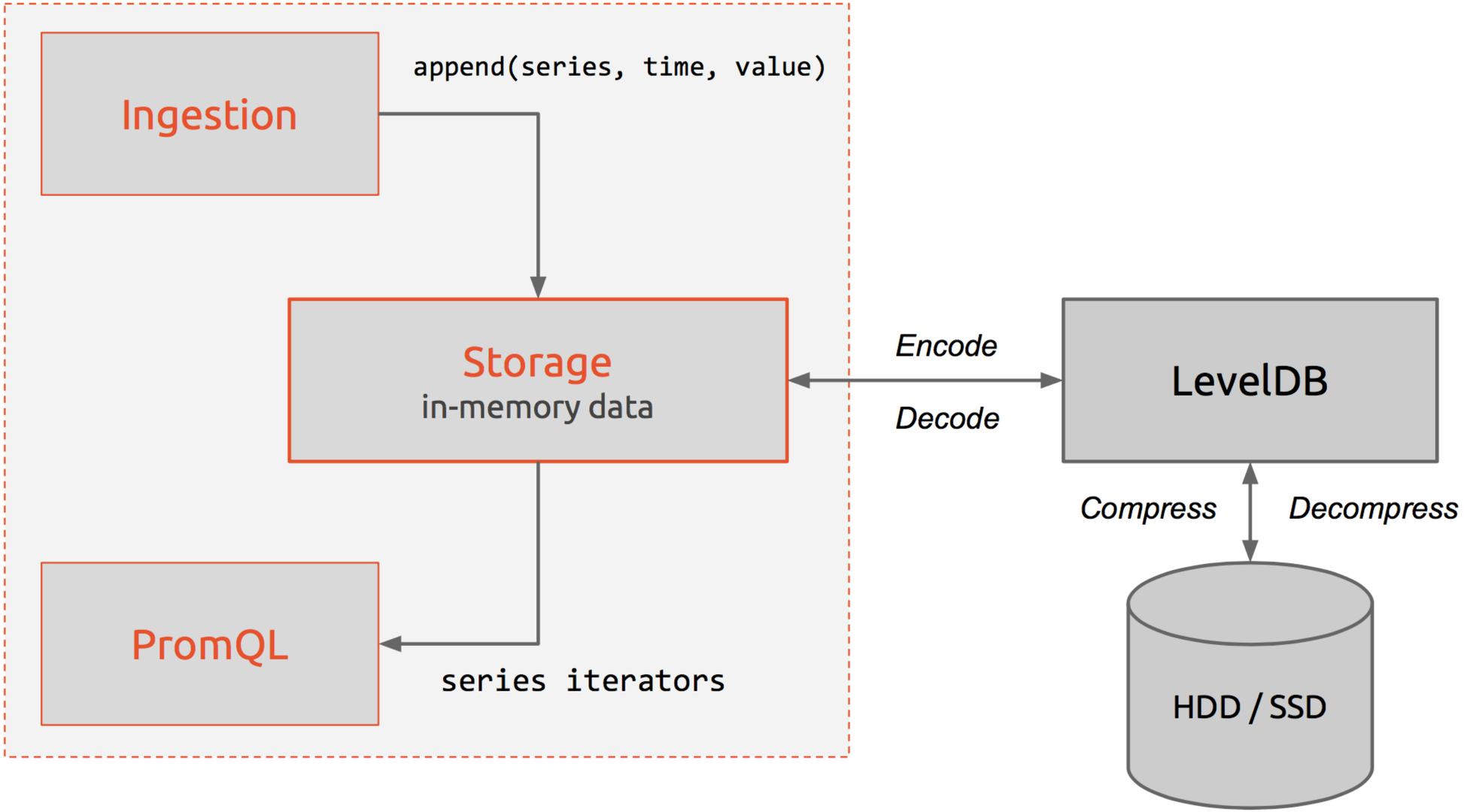
About timeseries

- Timeseries can be lossy
- Timeseries compress uniquely on data sets
- Write heavy
- Key, Time, DataPoint
  - CNX:IND, June 15 12:23, $23.40

Dark days

- Graphite
- InfluxDb
- Mysql storing metrics
- OpenTSDB (UGGHHHHHH)


Prometheus


Ingestion
append(series, time, value)
Storage
in-memory data
PromQL
series iterators
Encode
Decode
LevelDB
Compress
Decompress
HDD / SSD

```
rate(http_request_latency[1m])
```

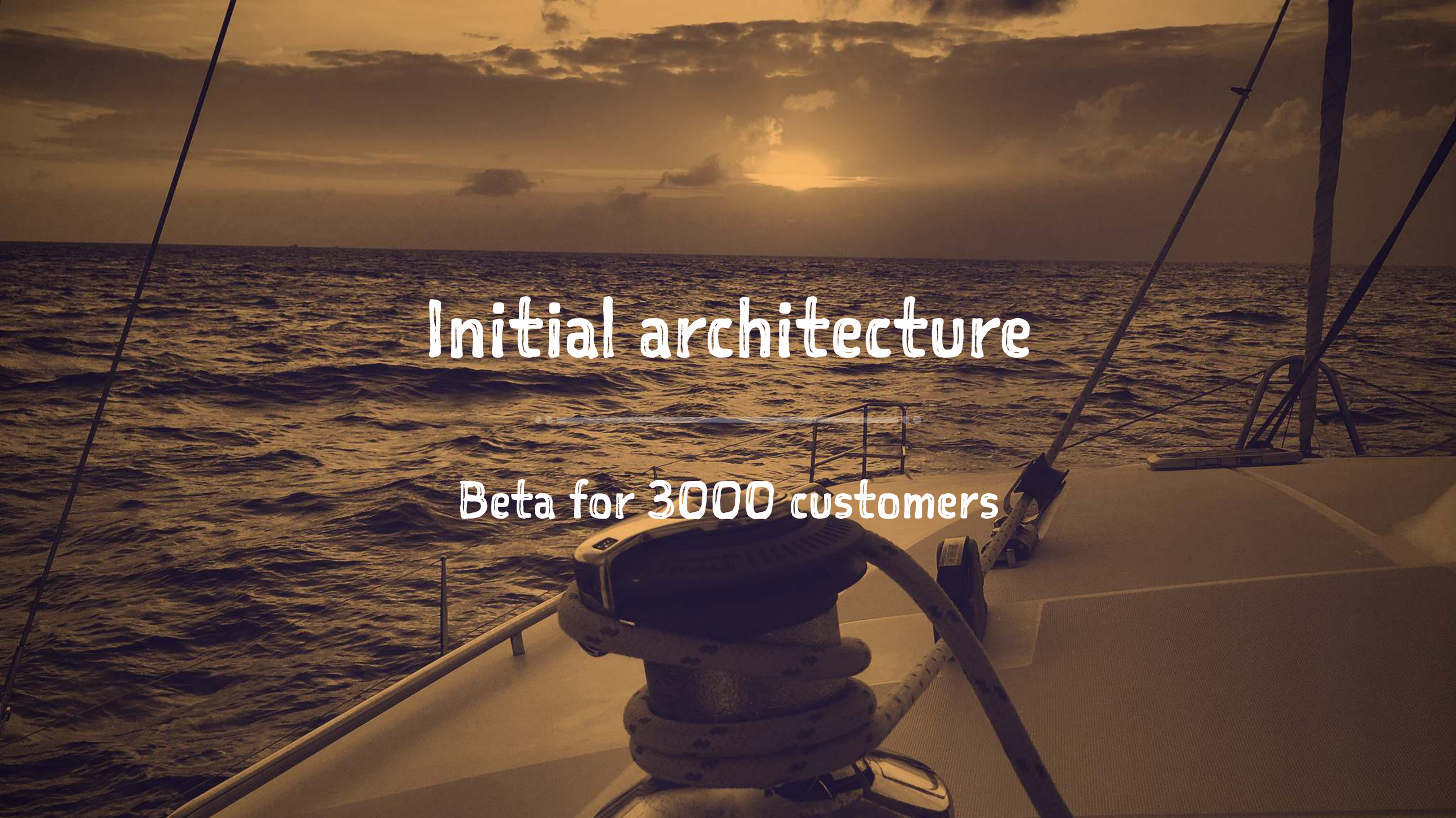
Julius Volz
@juliusvolz

You ask why @PrometheusIO has PromQL and not some SQL-like dialect for time series computations? Behold:

PromQL: `errors{job="foo"} / total{job="foo"}`

SQL:

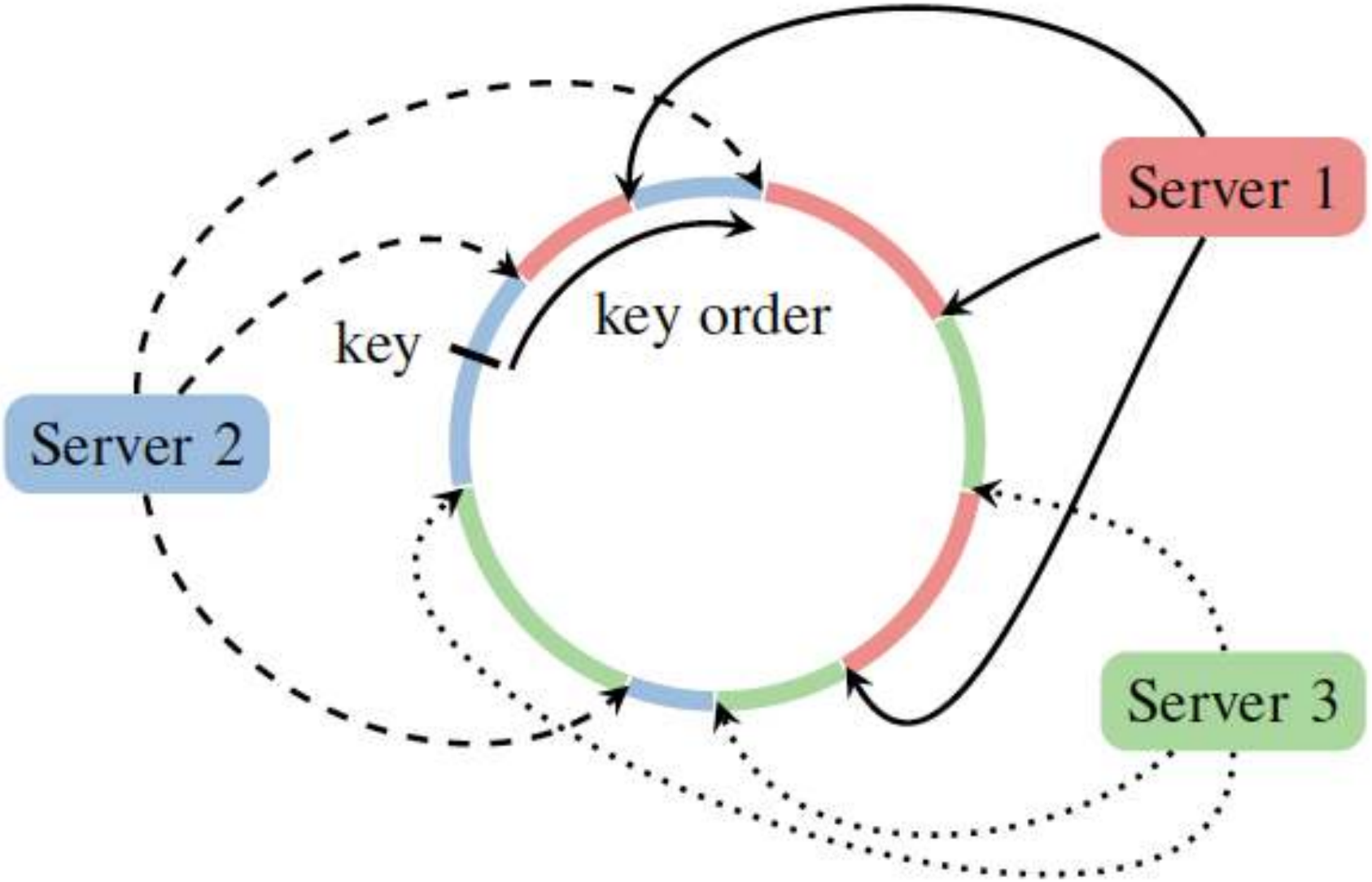
```
SELECT errors.job, errors.instance, [...more labels...], errors.value / total.value FROM errors,
total WHERE errors.job="foo" AND total.job="foo" JOIN [...some more complicated stuff here...]
```

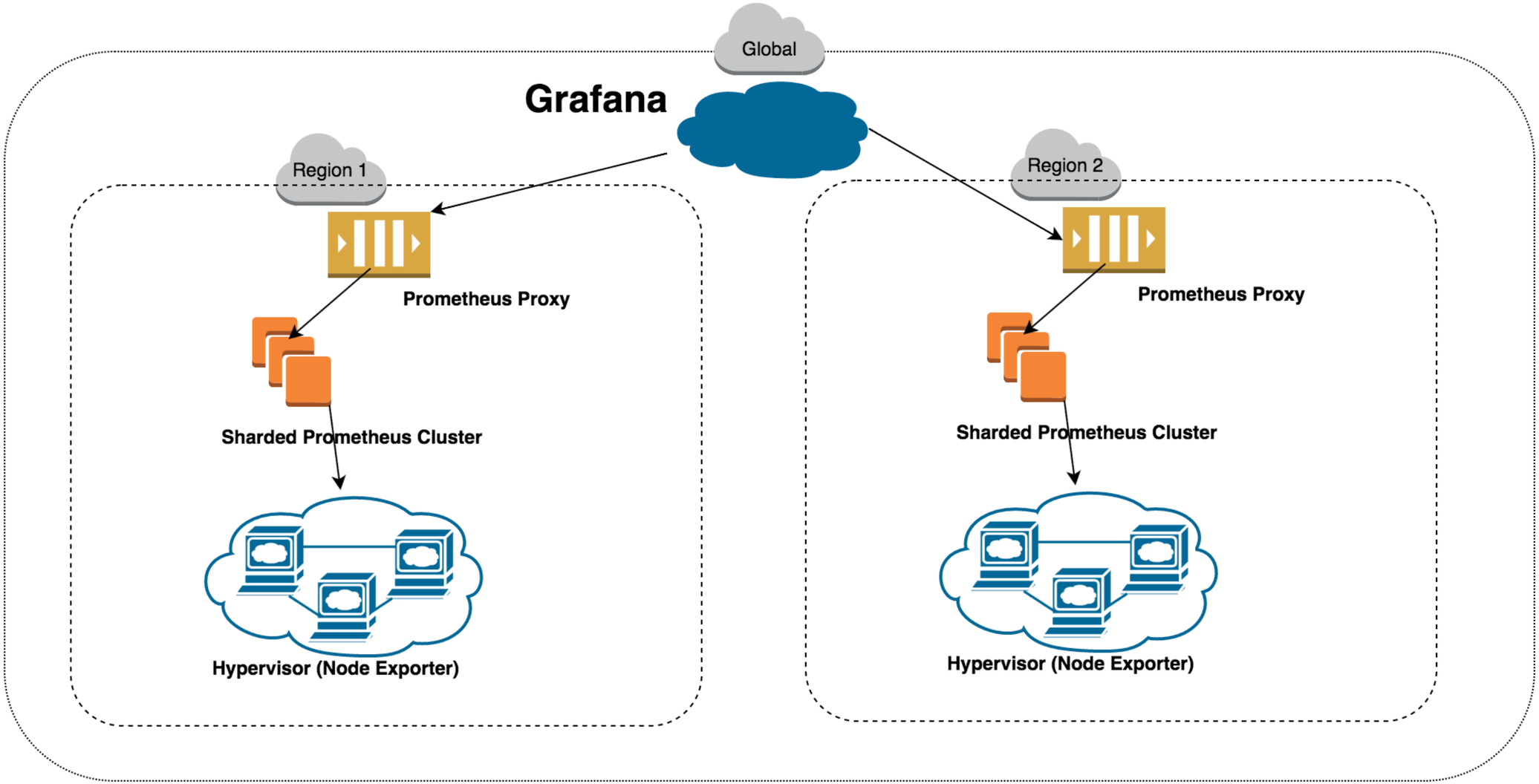


Initial architecture

Beta for 3000 customers


Hash sharded Prometheus
3-4 per datacenter


Server 1
key order
key
Server 2
Server 3


Global
Grafana
Region 1
Prometheus Proxy
Sharded Prometheus Cluster
Hypervisor (Node Exporter)
Region 2
Prometheus Proxy
Sharded Prometheus Cluster
Hypervisor (Node Exporter)

Performance requirements

- 3 Gbits/sec of traffic
- 100k Writes a second
- 50ms Reads
- 100,000 customers to start
- 20 TB of storage

Introducing Vulcan

https://github.com/digitalocean/vulcan

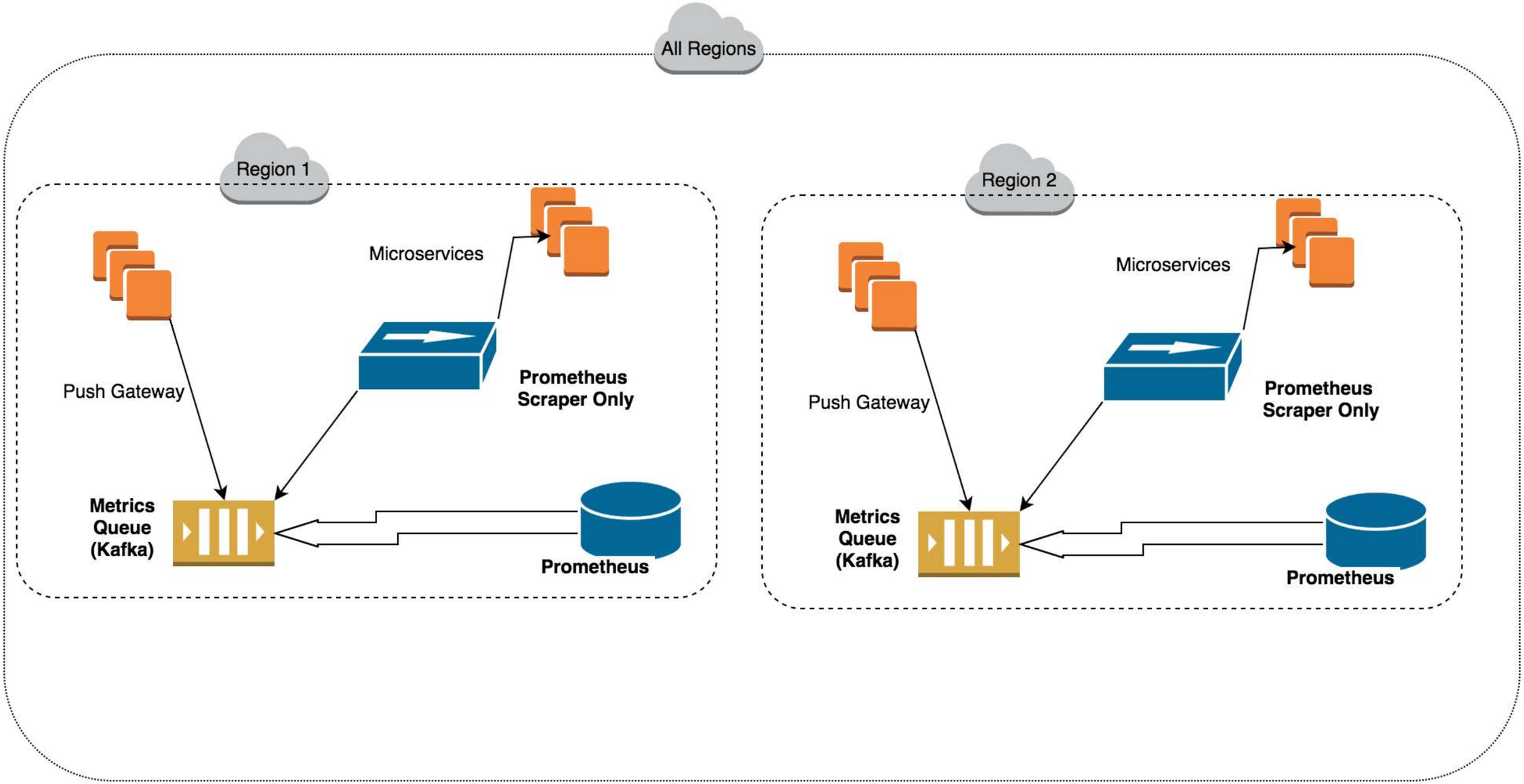

Strange PRs

A fateful meeting at Soundcloud...

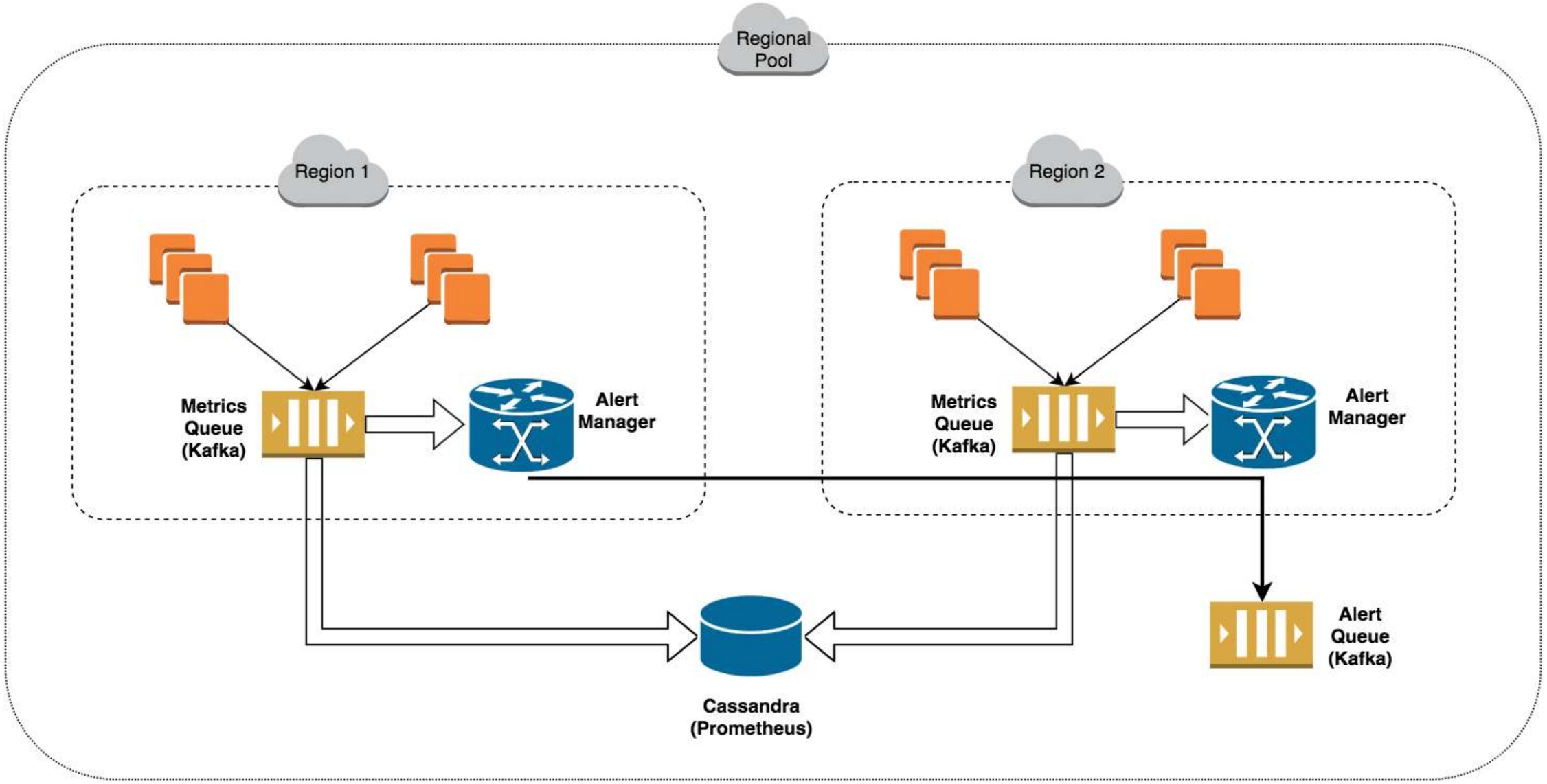
Architecture changes

- Split to microservices
- Containerization
- Message Queues

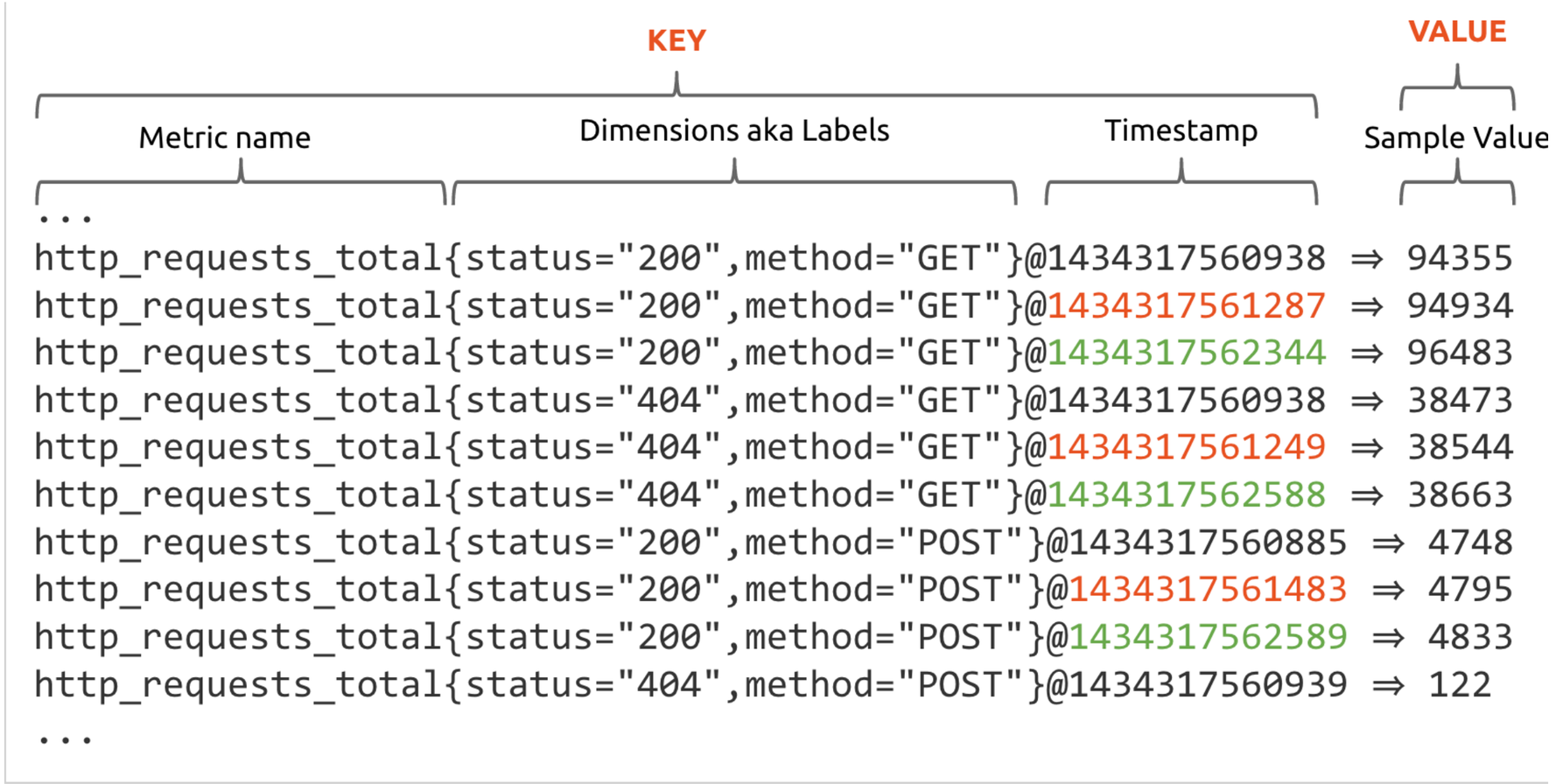
Pipelining data


All Regions
Region 1
Microservices
Push Gateway
Prometheus
Scraper Only
Metrics
Queue
(Kafka)
Prometheus
Region 2
Microservices
Push Gateway
Prometheus
Scraper Only
Metrics
Queue
(Kafka)
Prometheus


Scaling storage


Regional Pool
Region 1
Region 2
Metrics Queue (Kafka)
Alert Manager
Metrics Queue (Kafka)
Alert Manager
Alert Queue (Kafka)
Cassandra (Prometheus)

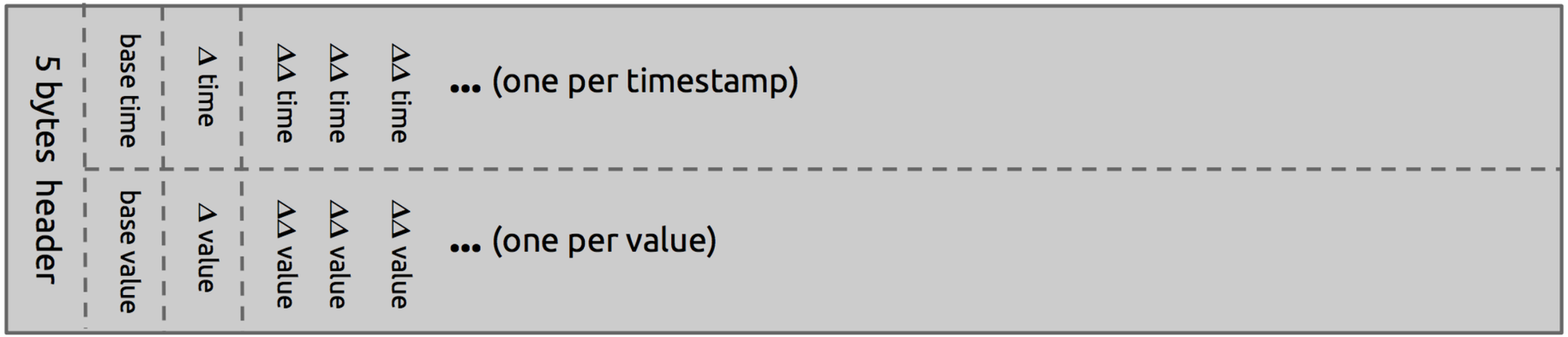
Metrics format

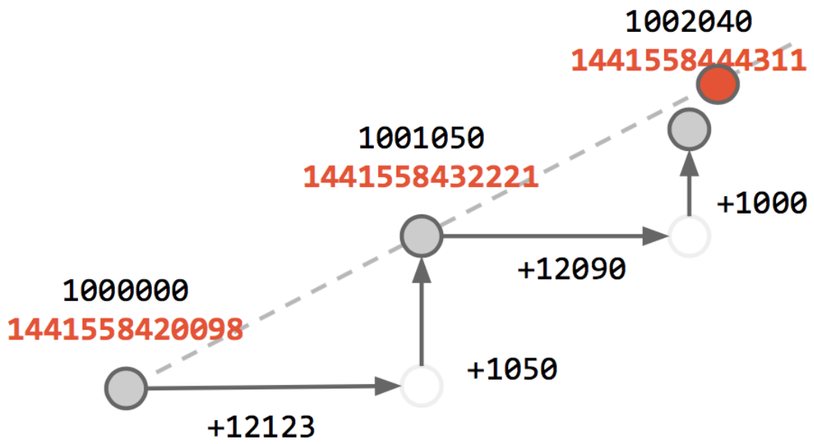


Timeseries Schema

- VI Timeseries Table
  - key (Combined Key)
  - timestamp (Combined Key)
  - datapoint (float64)

V2 Chunks (1 KB)

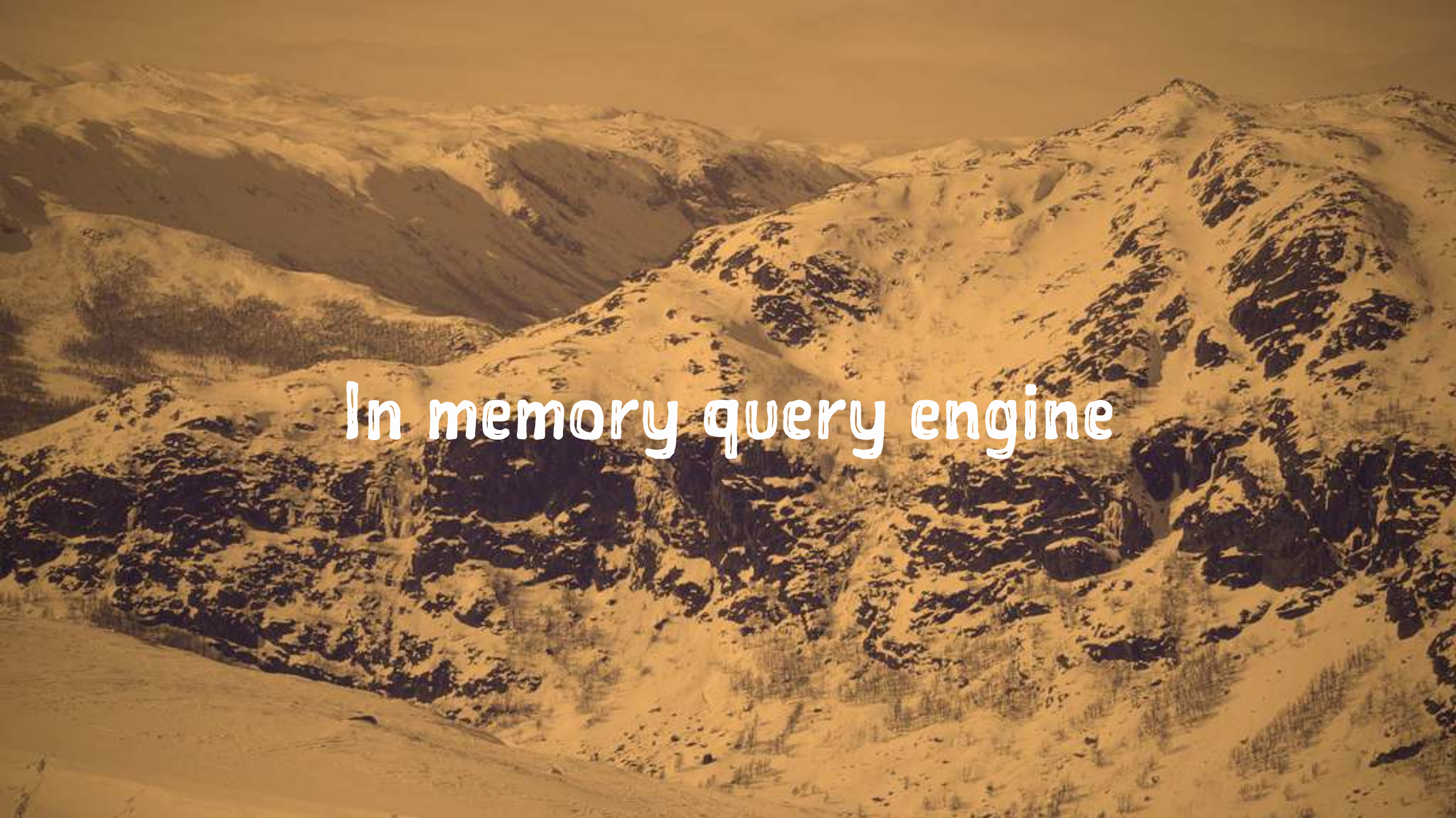




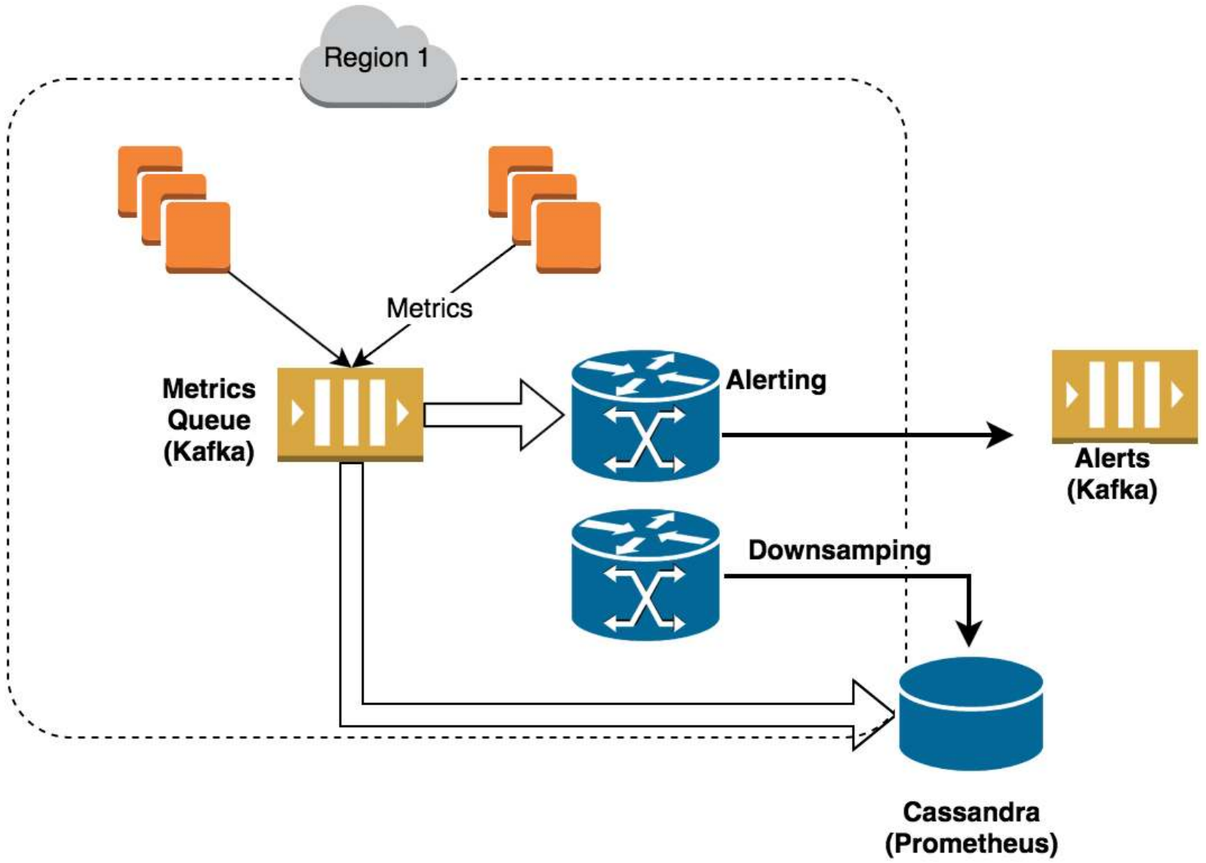
1000000	1441558420098
1050	12123
-60	-33
-50	-56
+50	-8

- V2 Timeseries Table
  - key (Combined Key)
  - timestamp range (2hours) (Combined Key)
  - raw data (1kb blob)

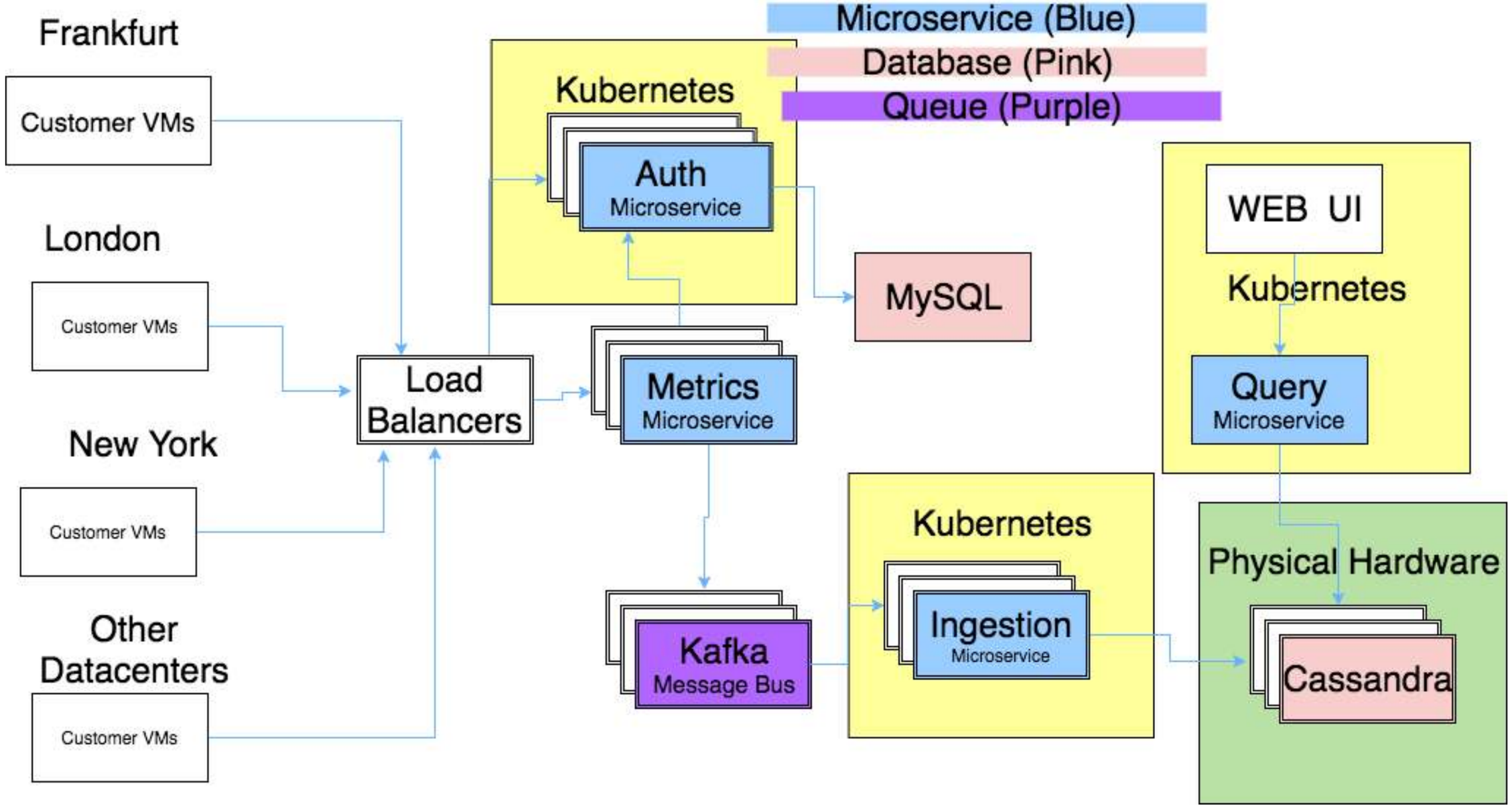
- Index Table
  - Customer (Combined key)
  - keyPrefix (Combined key)
  - time
  - key


In memory query engine

Downsampling




Final Architecture


Microservice (Blue)
Database (Pink)
Queue (Purple)
Frankfurt
Customer VMs
London
Customer VMs
New York
Customer VMs
Other Datacenters
Customer VMs
Load Balancers
Kubernetes
Auth
Microservice
MySQL
Metrics
Microservice
WEB UI
Kubernetes
Query
Microservice
Kafka
Message Bus
Kubernetes
Ingestion
Microservice
Physical Hardware
Cassandra


Questions?
NOOCYNY