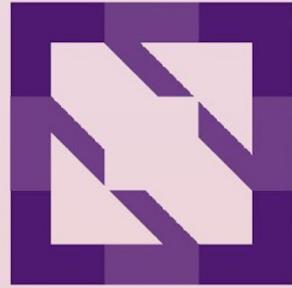


KubeCon



CloudNativeCon

North America 2023



KubeCon



CloudNativeCon

North America 2023

Thanos Unleashed: Mastering the Challenges of Production-scale metrics

Joel Verezhak – Open Systems

open  systems

KubeCon = Kubernetes Conference



open systems





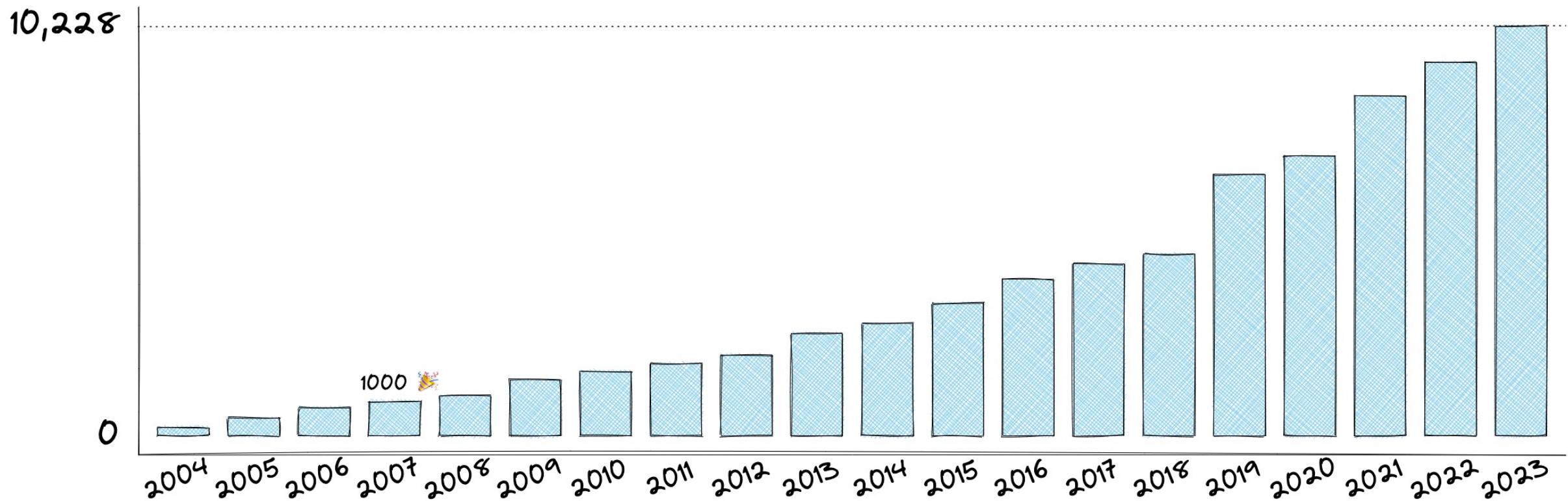
KubeCon

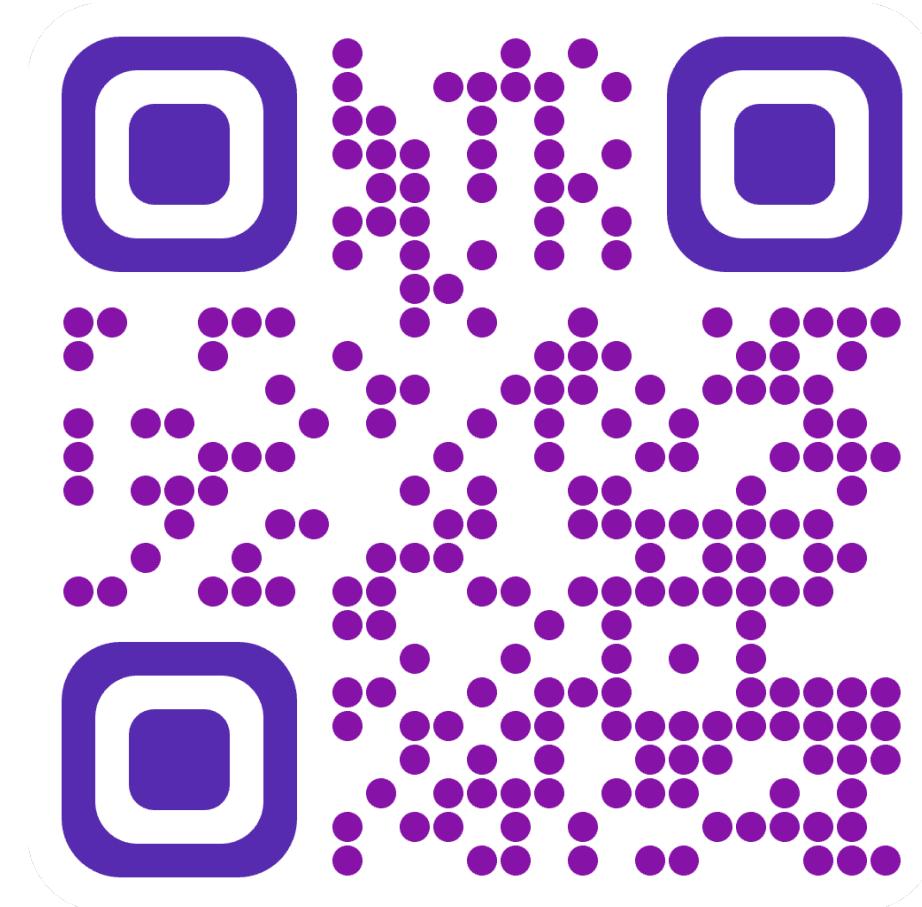


CloudNativeCon

North America 2023

Total Active Hosts





Link to KubeCon EU talk with details about how we do monitoring at Open Systems!

Today we'll talk about



KubeCon



CloudNativeCon

North America 2023



Scalability, resilience, and performance of the write-path
One weird trick that reduced storage costs by 86% (*cloud providers hate this*)



Thanos

Open source, highly available Prometheus setup with long term storage capabilities.



Global Query View

Scale your Prometheus setup by enabling querying of your Prometheus metrics across multiple Prometheus servers and clusters.



Unlimited Retention

Extend the system with the object storage of your choice to store your metrics for unlimited time. Supports GCP, S3, Azure, Swift and Tencent COS.



Prometheus Compatible

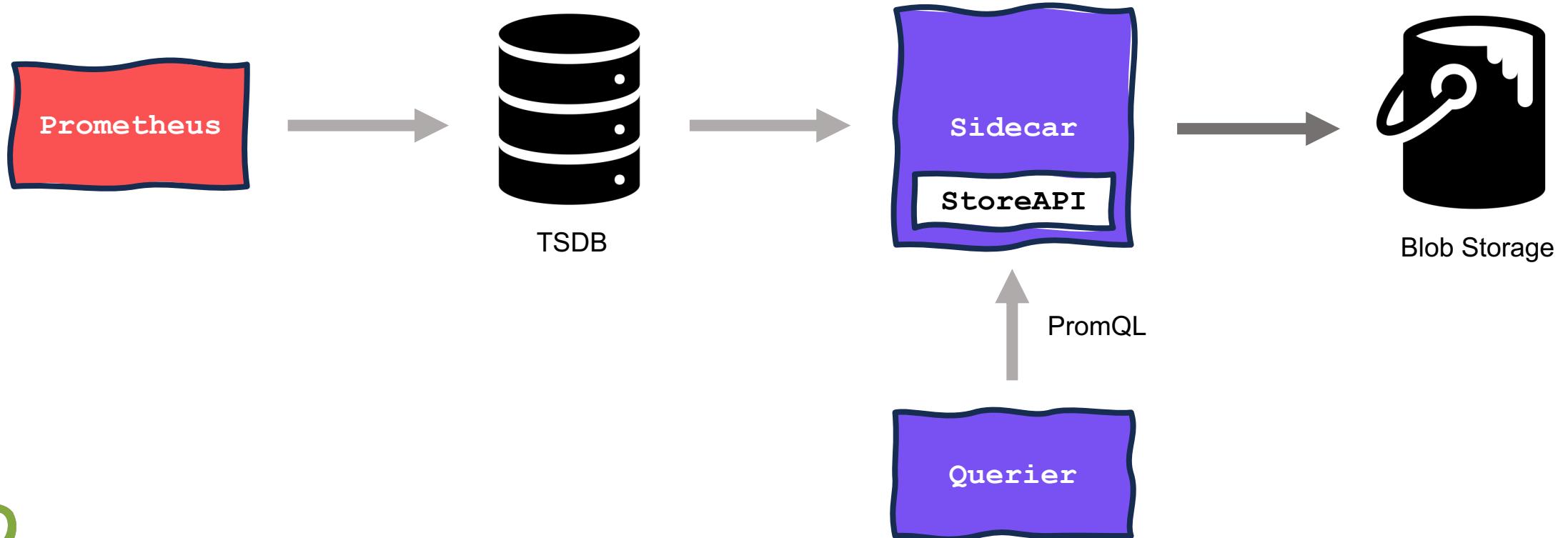
Use the same tools you love, such as Grafana and others, that support the Prometheus Query API.



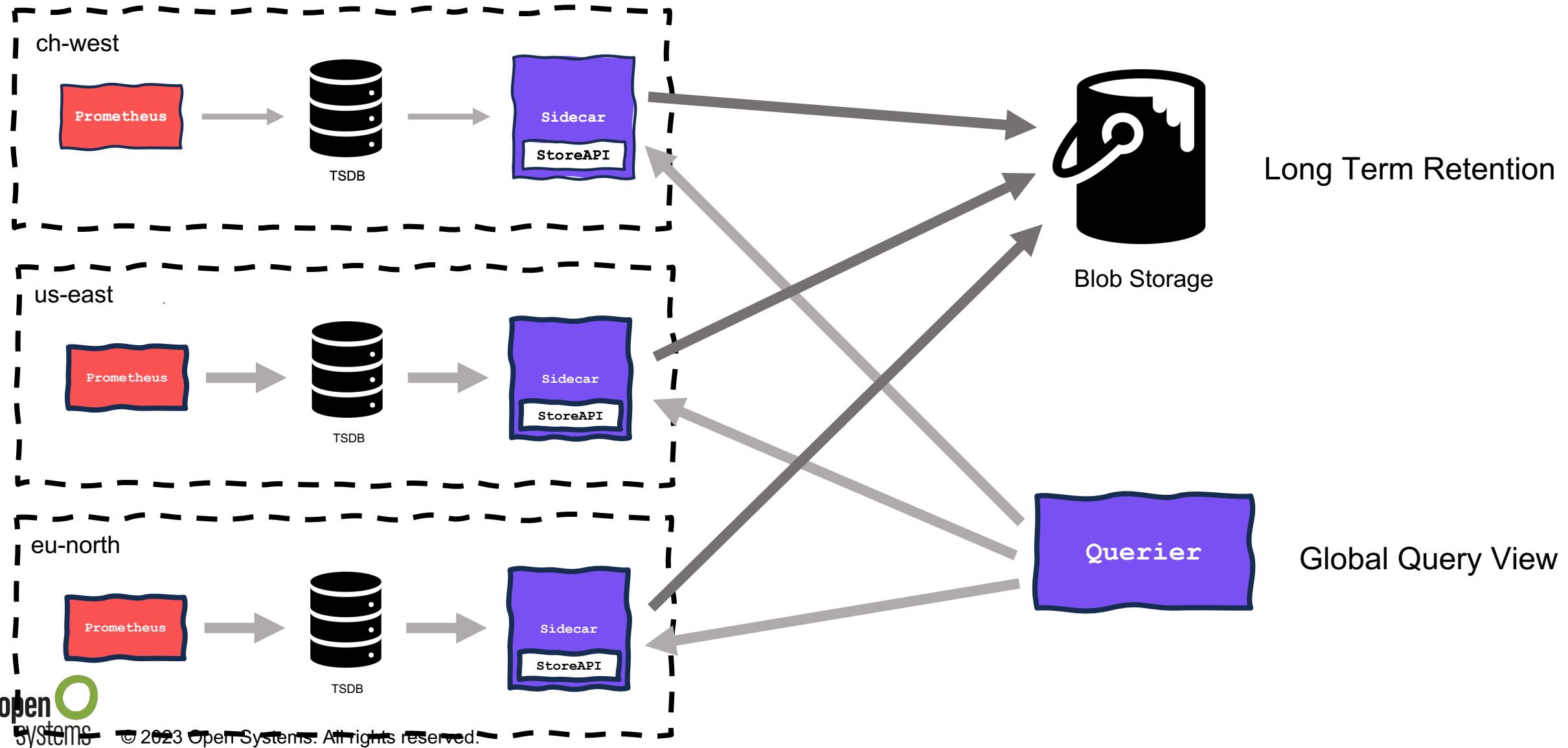
Downsampling & Compaction

Downsample historical data for massive query speedup when querying large time ranges or configure complex retention policies.

Thanos Sidecar



Thanos Sidecar





KubeCon



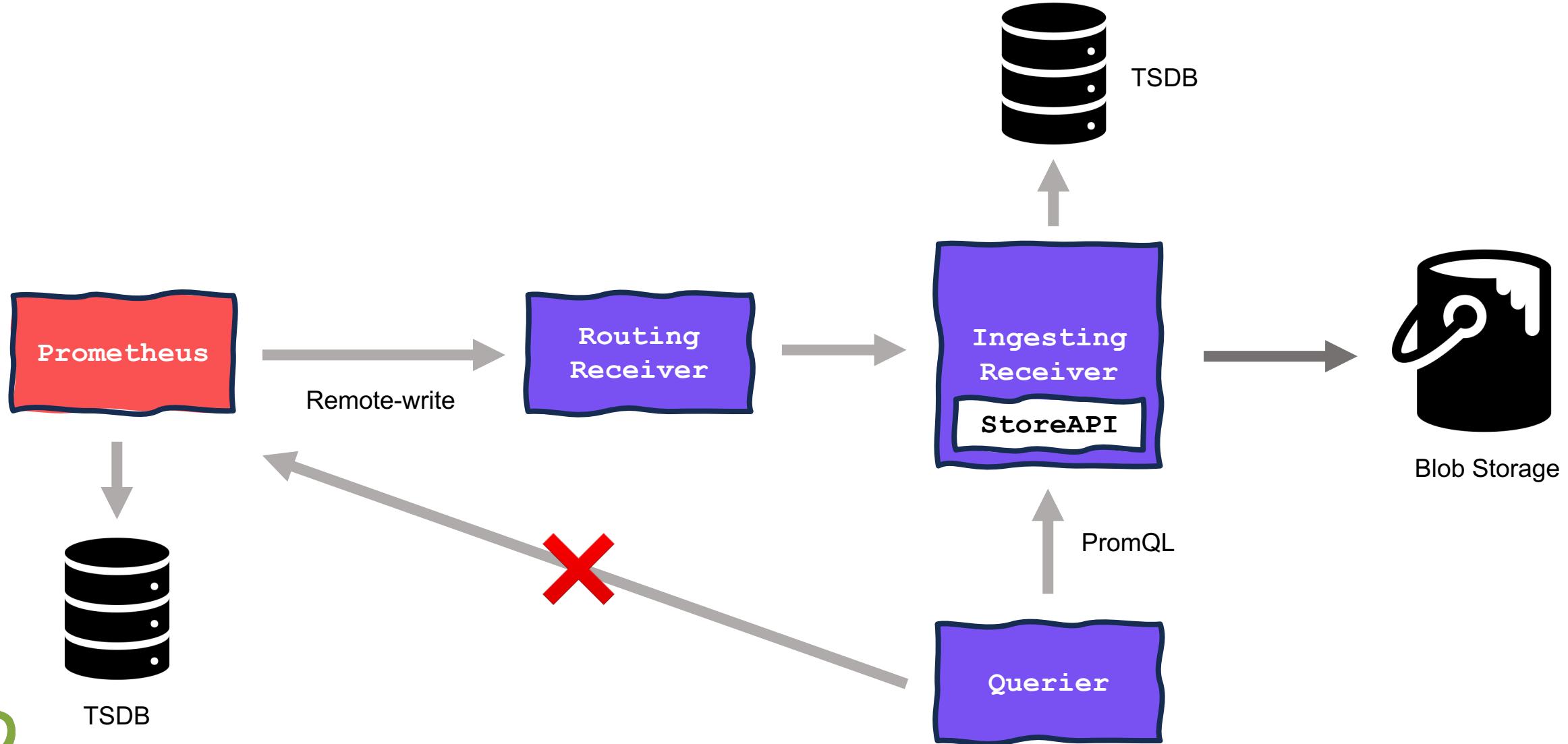
CloudNativeCon

North America 2023

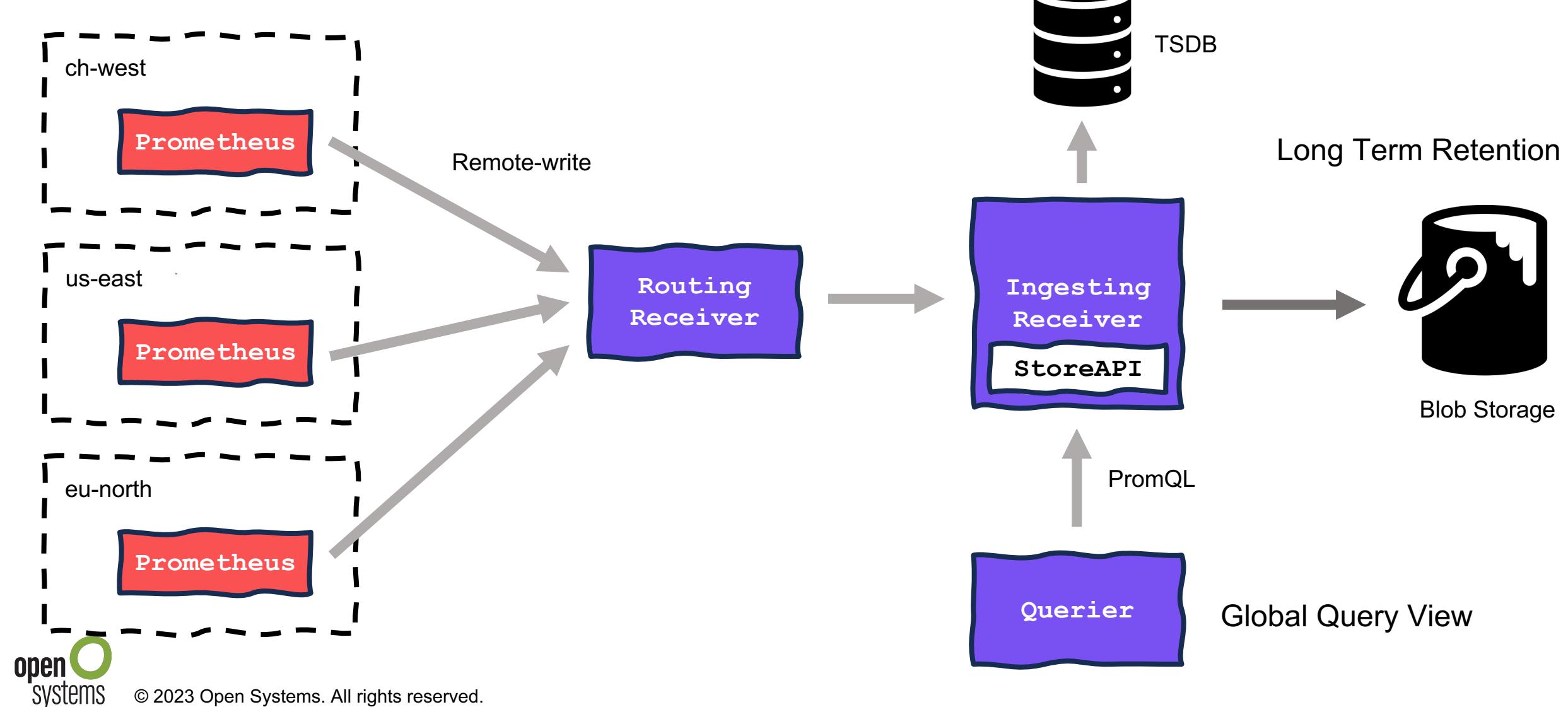


But what if we can't use the Sidecar approach?

Thanos Receiver



Thanos Receiver



Our Thanos Cluster



KubeCon

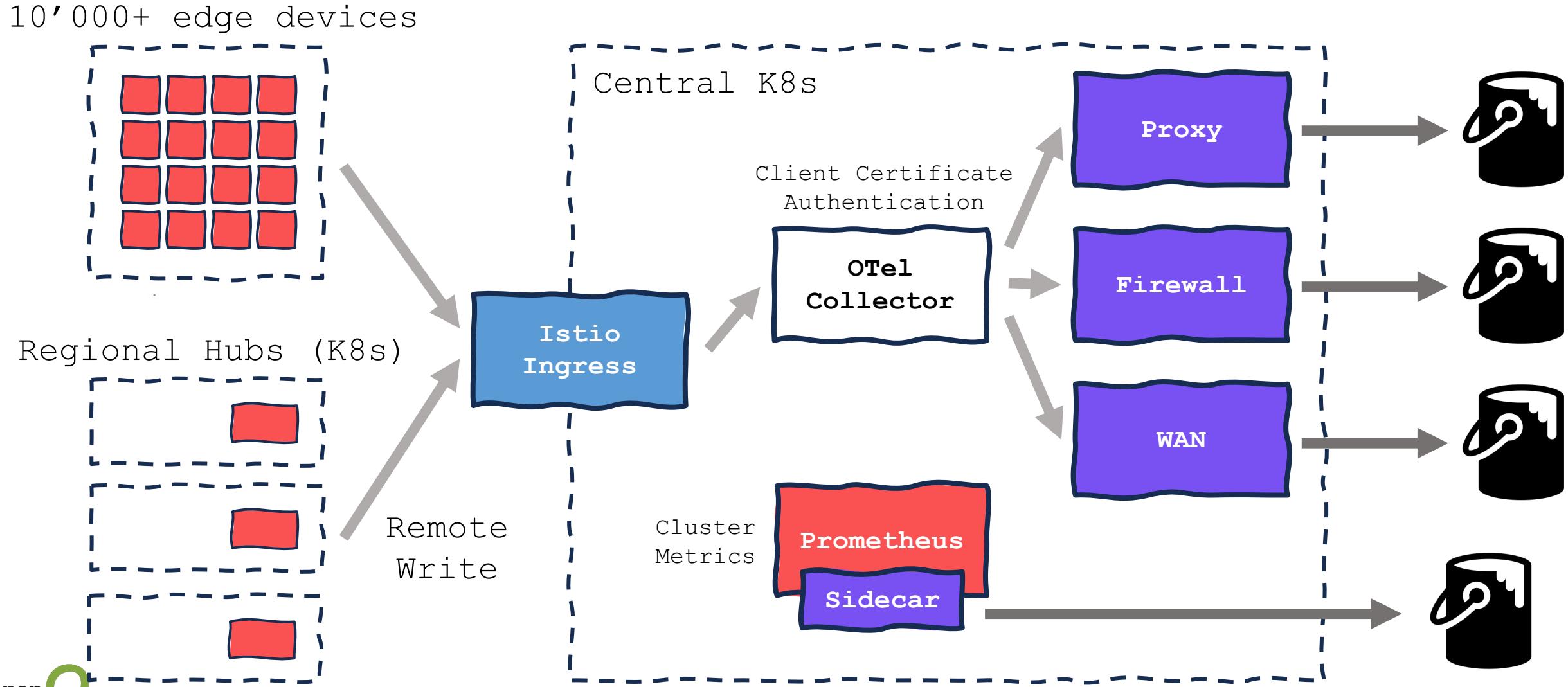


CloudNativeCon

North America 2023

- ~200 million active time series
- ~90 CPUs
- ~1 TiB Memory
- ~450/250 MB/s receive/transmit bandwidth

Our Setup



Write-path: Main Concerns



KubeCon

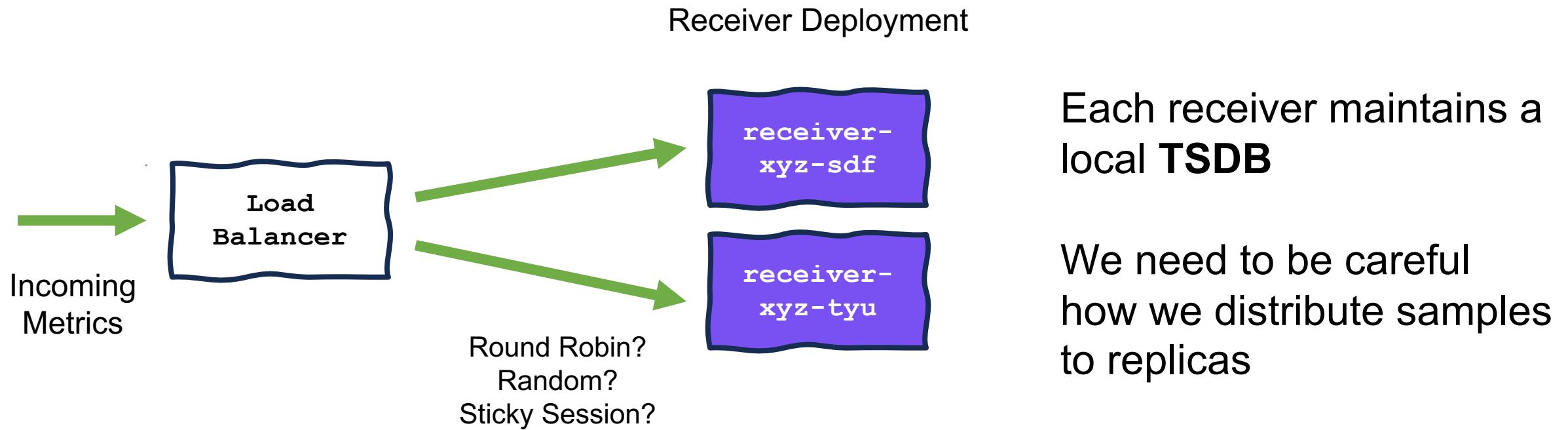


CloudNativeCon

North America 2023

Scalability and Resilience
Tenant Isolation and QoS
Data Availability

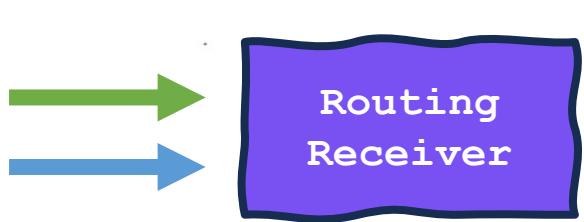
Thanos Write-path: naïve approach



Thanos Write-path: stateful approach

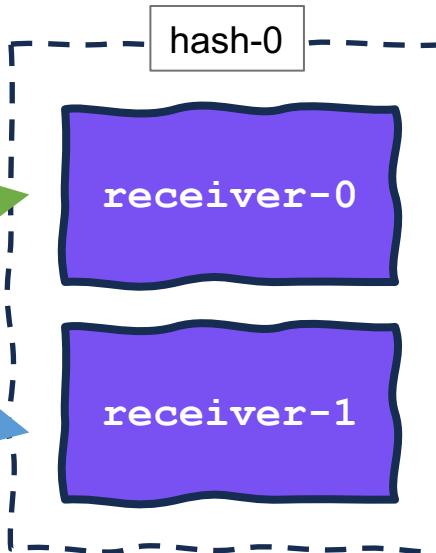
Labelset **hashed** and mapped to a given receive replica

`cpu_usage{tenant="proxy", host_id="1234"}`



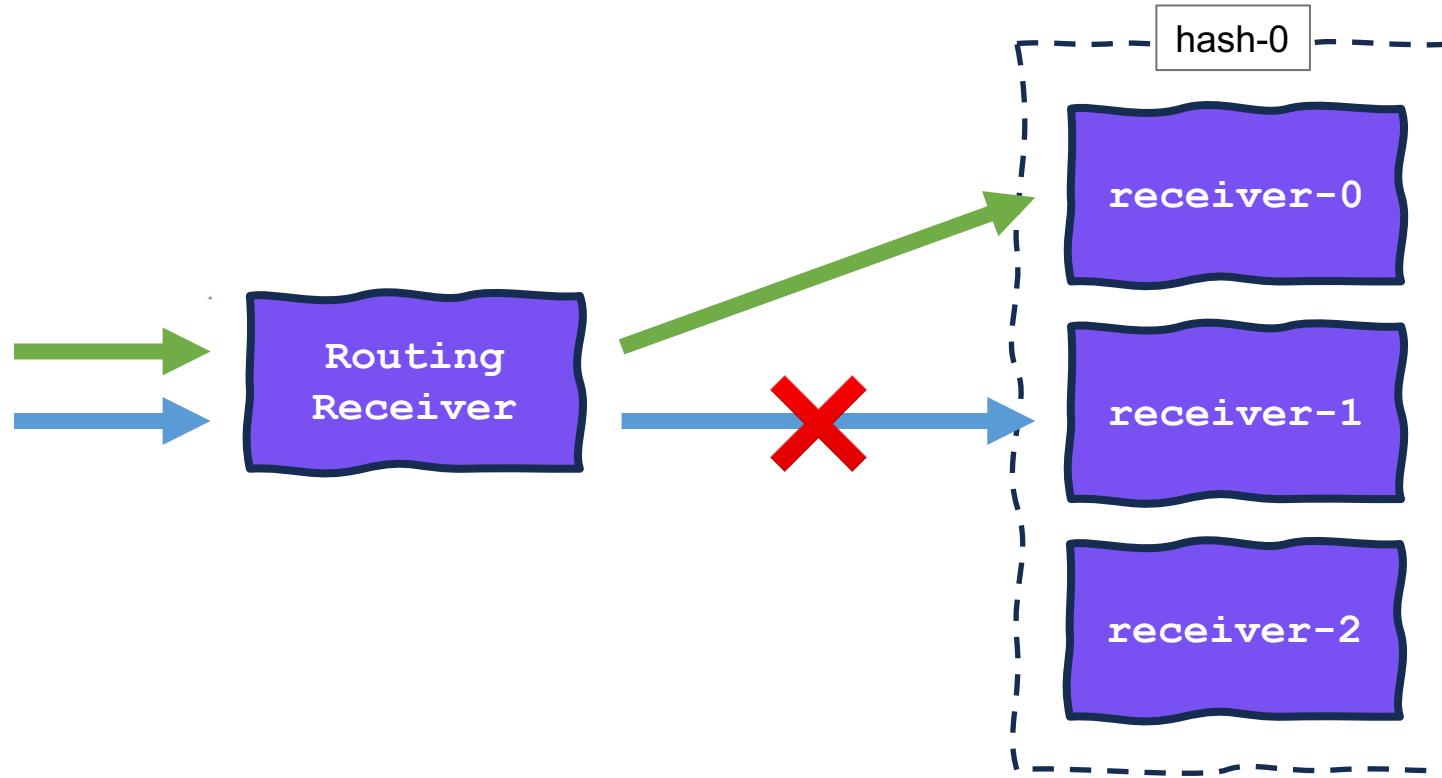
`cpu_usage{tenant="proxy", host_id="5678"}`

Receiver StatefulSet



Metrics from the same labelset always end up in the same **TSDB**

Diving in: the (remote) Write-path

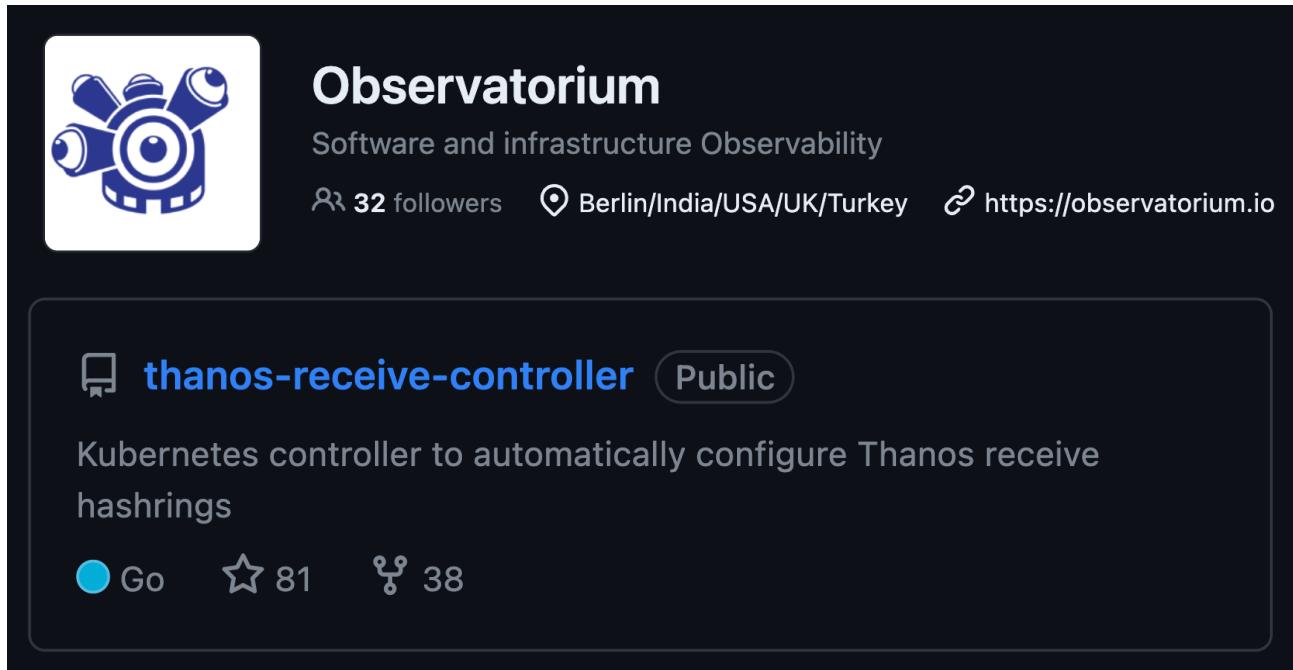


How to handle **scaling**?

What if a receiver becomes **unhealthy**?

Hash Ring Management

- Options for hash ring management:
- Static definitions
 - Use a **controller**



Observatorium
Software and infrastructure Observability

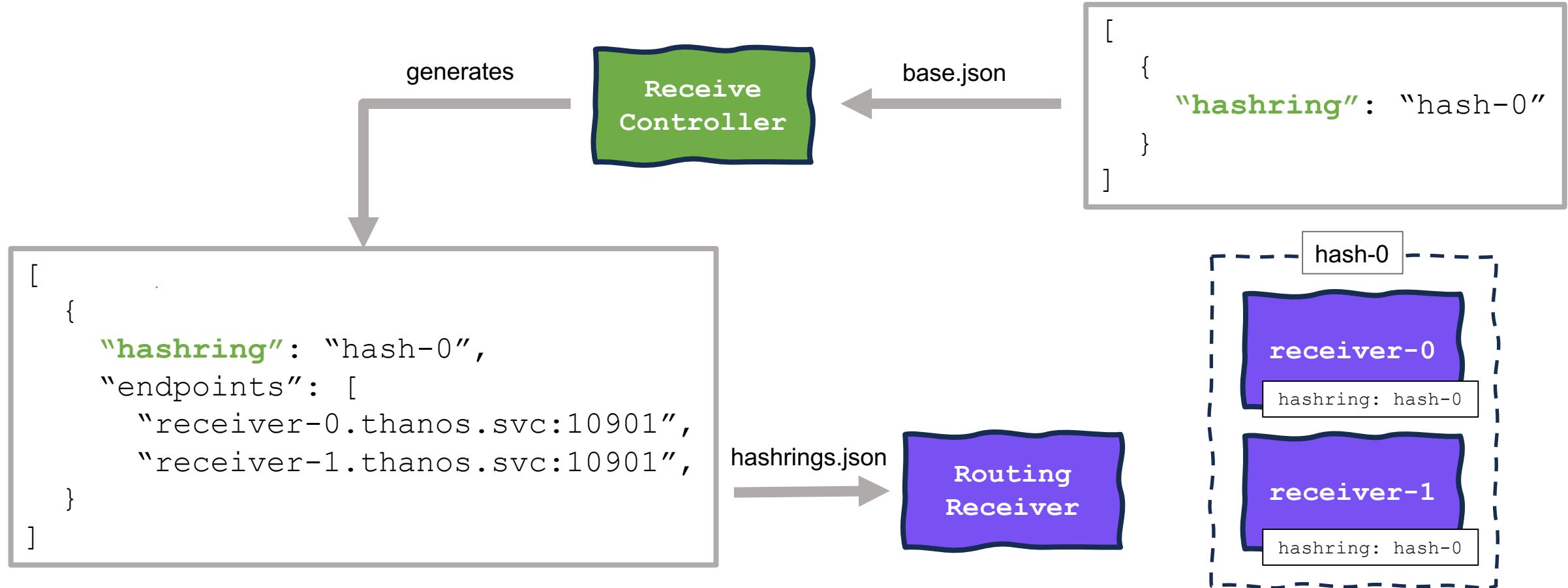
32 followers Berlin/India/USA/UK/Turkey <https://observatorium.io>

thanos-receive-controller Public

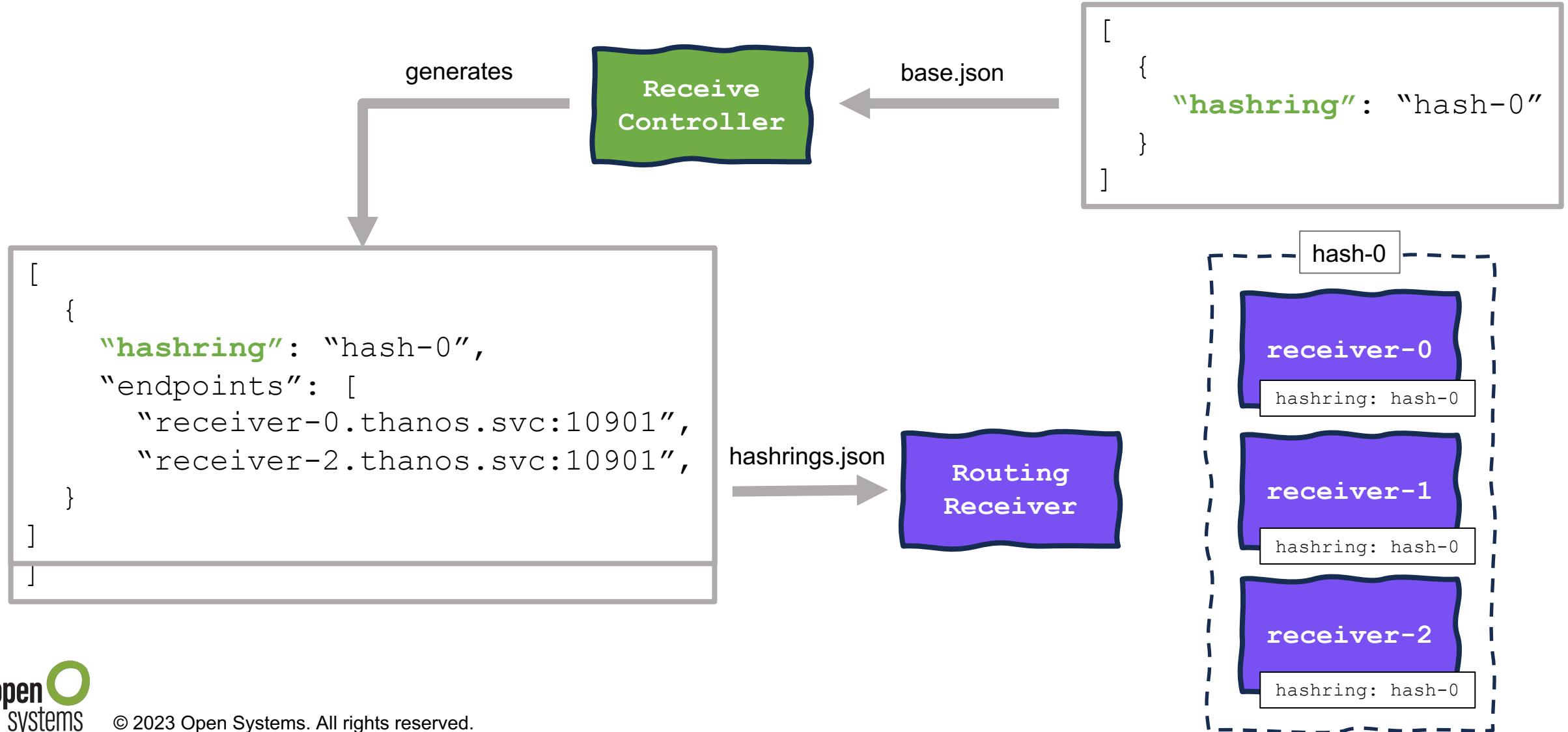
Kubernetes controller to automatically configure Thanos receive hashrings

Go 81 38

Receiver Hash Rings



Receiver Hash Rings



Scalability: Key Config



KubeCon



CloudNativeCon

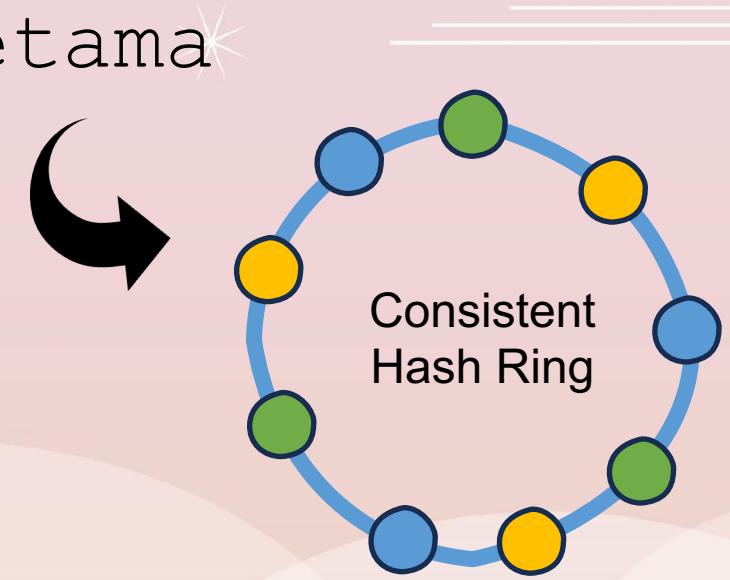
North America 2023

Routing Receiver

```
--receive.hashrings-algorithm=ketama
```

Receive Controller

```
--allow-only-ready-replicas  
--allow-dynamic-scaling
```

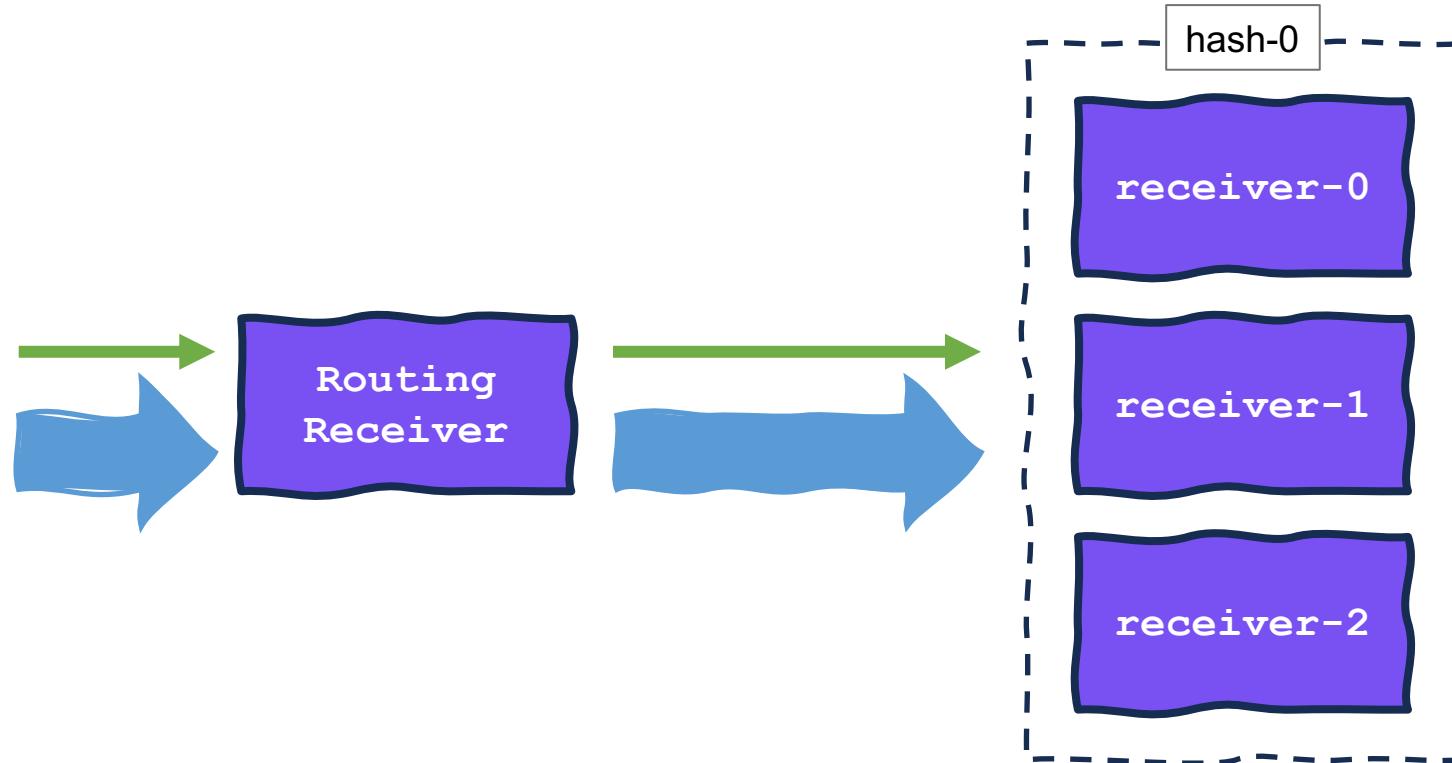


The Perl Hash incident

```
15 #!/usr/bin/perl
14
13 my %labels = (
12     company_id => "6",
11     host_id   => "1234",
10 );
9
8 my $key = "company_id";  # Correctly setting the key we want to access
7 my $value = $labels{$key}; # Correct way to access the hash value
6
5 # Bug: Accidentally setting the variable to the hash reference instead of the value
4 my $buggy_value = \%labels;
3
2 print "Value: $value\n";  # Output the correct value (6)
1 print "Buggy Value: $buggy_value\n"; # Bug: Outputs the hash reference
16
```

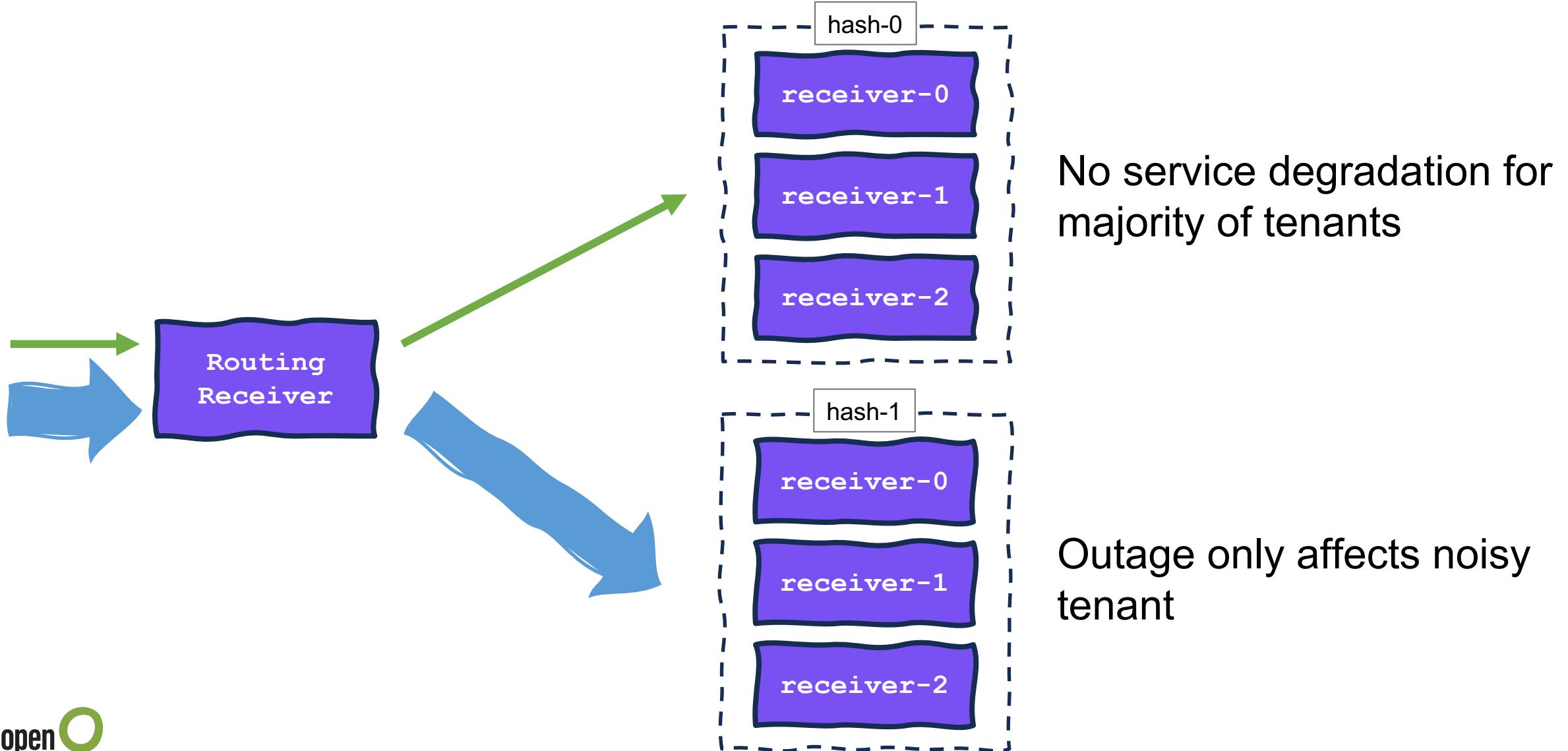
```
└> perl metrics.pl
Value: 6
Buggy Value: HASH(0x15a829c60)
```

Noisy neighbours

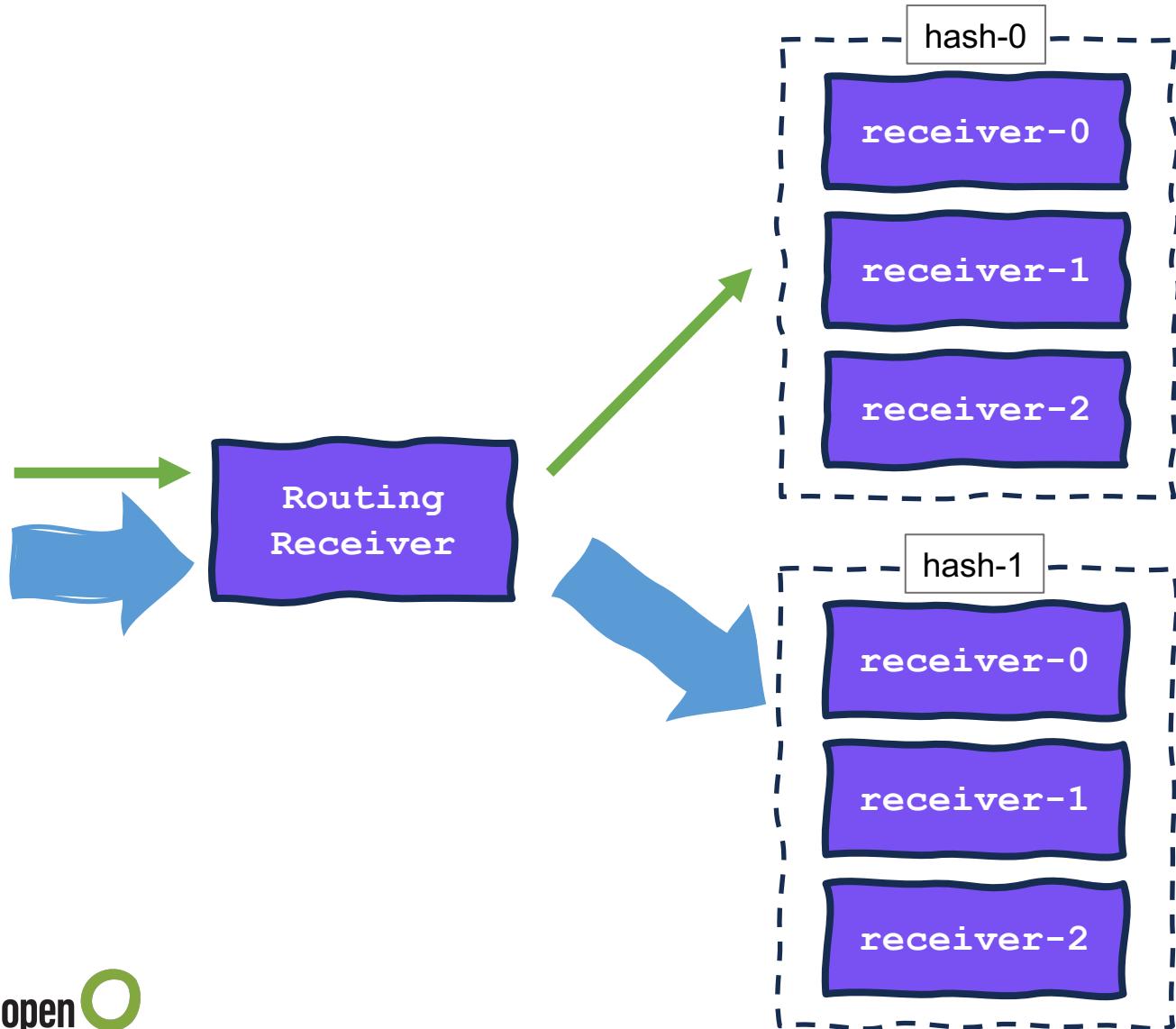


One troublesome tenant
leads to a full service outage

Hard tenancy



Creating a hard tenant



Need to map tenants to specific hashrings
in receive controller

```
[  
  {  
    # noisy tenant has dedicated  
    # hashring  
    "hashring": "hash-1",  
    "tenants": ["noisy"]  
  },  
  {  
    # default hashring for all  
    # other tenants  
    "hashring": "hash-0"  
  }  
]
```



KubeCon



CloudNativeCon

North America 2023

What if we don't want to juggle multiple hash rings?



Use Active Series Limiting

Active Series Limiting

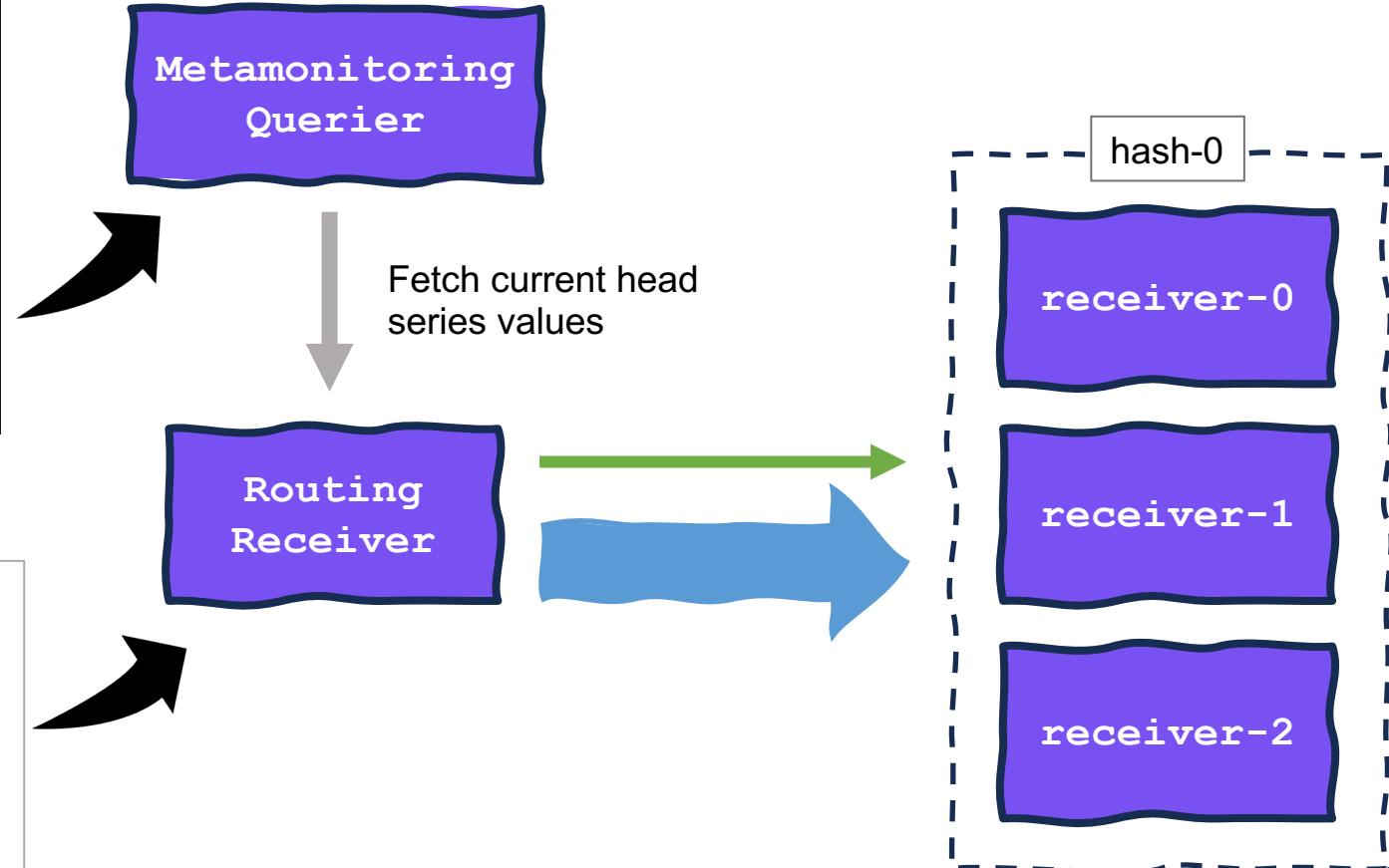
Basic idea:

1. Set reasonable limits for how many active series tenants can ship
2. Query current metric values and limit requests before they reach ingestors

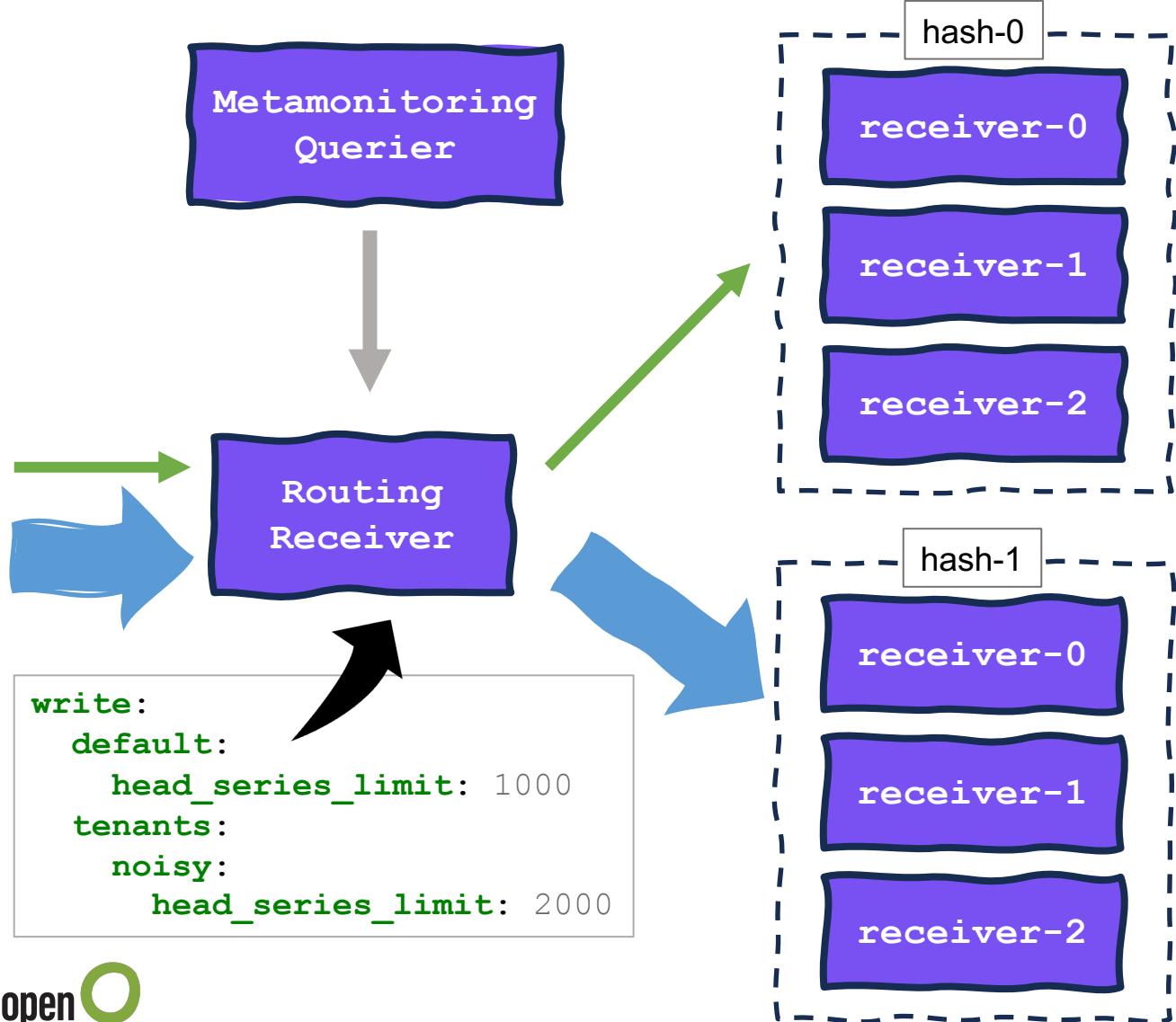
Active Series Limiting

sum(prometheus_tsdb_head_series) by (tenant)	
tenant	Value ↓
sdp	28508533
firewall	25738176
wan	23582130
bandwidthcontrol	9032692

```
write:  
  default:  
    head_series_limit: 1000  
  tenants:  
    sdp:  
      head_series_limit: 2000
```



Hybrid approach



```
[  
  {  
    # noisy tenant has dedicated  
    # hashring  
    "hashring": "hash-0",  
    "tenants": ["noisy"]  
  },  
  {  
    # default hashring for all  
    # other tenants  
    "hashring": "hash-0"  
  }  
]
```

The Connectivity incident

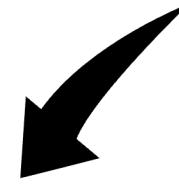
Availability

Outage?

Service availability



SUNDAY, 29.10.2023



00:00 02:00 03:00 05:00 07:00 09:00 11:00 13:00 15:00 17:00 19:00 21:00

ISP Outage ● Service Down ● Maintenance ● Inactive ● Service Up ●

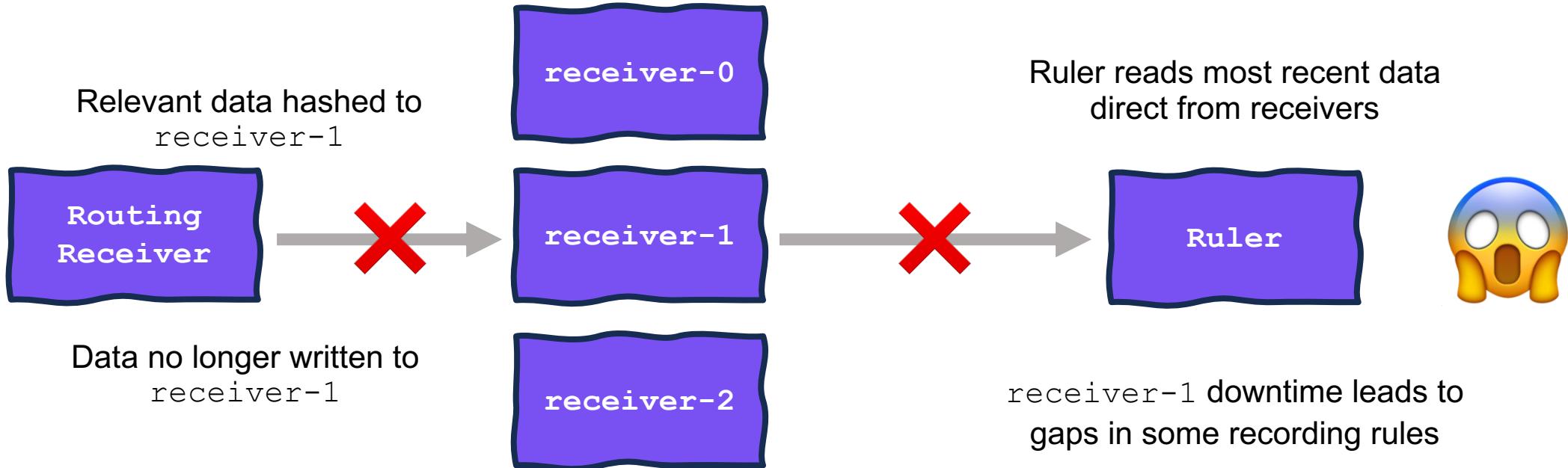
SLA Statistics

Tier 1	SLA OK	99.98 %	0.00 %
SLA CLASS	SLA STATUS	SLA TARGET	SLA REACH

The Connectivity incident

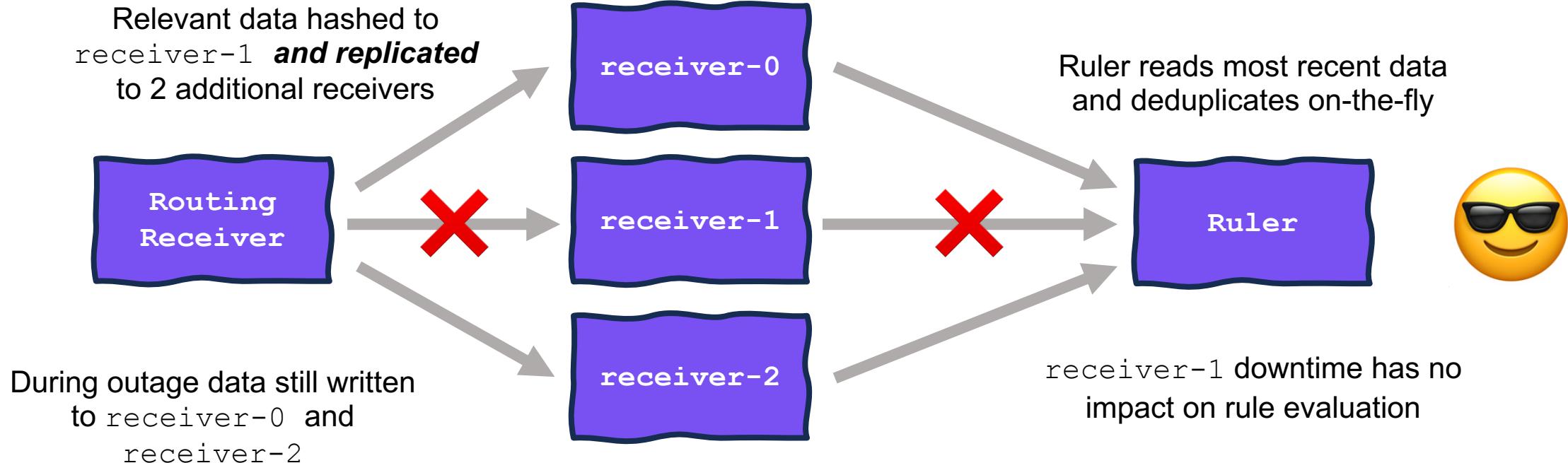


Oh, what's occurring?



An outage with Replication

--receive.replication-factor=3



Replication: Quorum

Replication Factor	Quorum	Max Unavailable
1	1	0
2	2	0
3	2	1
4	3	1
5	3	2
6	4	2
7	4	3
8	5	3
9	5	4
10	6	4

thanos / pkg / receive / handler.go

Code Blame 1158 lines (1020 loc) · 34 KB

```
617 func (h *Handler) writeQuorum() int {
618     return int((h.options.ReplicationFactor / 2) + 1)
619 }
```

$$Q = \frac{R}{2} + 1$$

Ensure that minumum number of replicas \geq replication factor

Data Availability: Key Config



KubeCon



CloudNativeCon

North America 2023

Routing Receiver

--receive.replication-factor=3

```
apiVersion: policy/v1
kind: PodDisruptionBudget
metadata:
  name: thanos-receive
spec:
  maxUnavailable: 1
  selector:
    matchLabels:
      app: thanos-receive
```

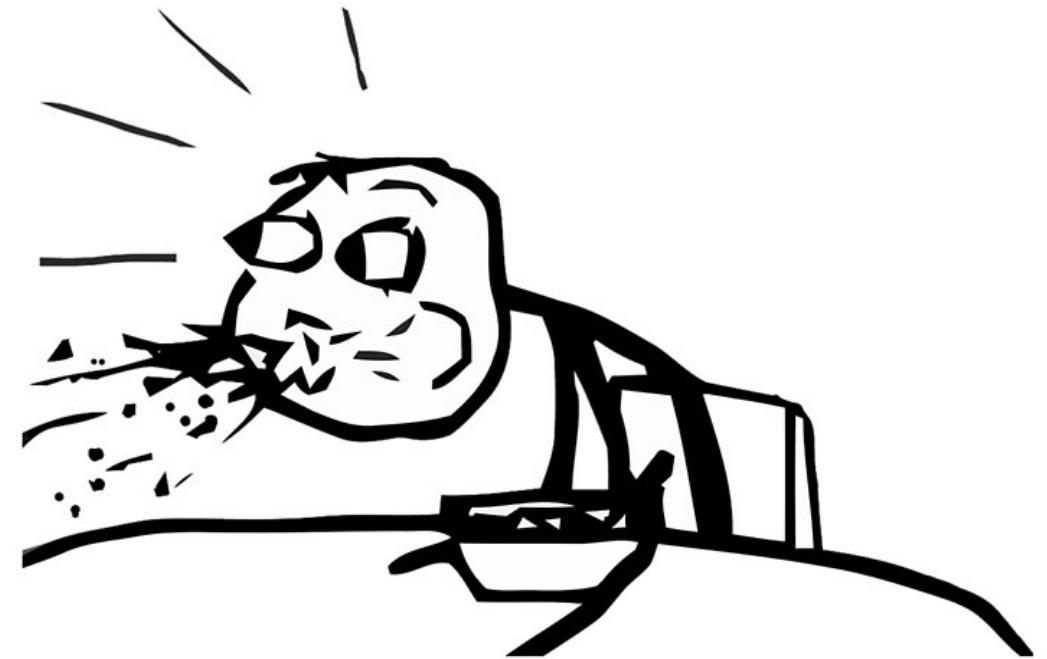
Use a Pod Disruption Budget to ensure that a Quorum of receives will always be available for writes

The increasing cloud costs incident

How it feels using the Cloud:

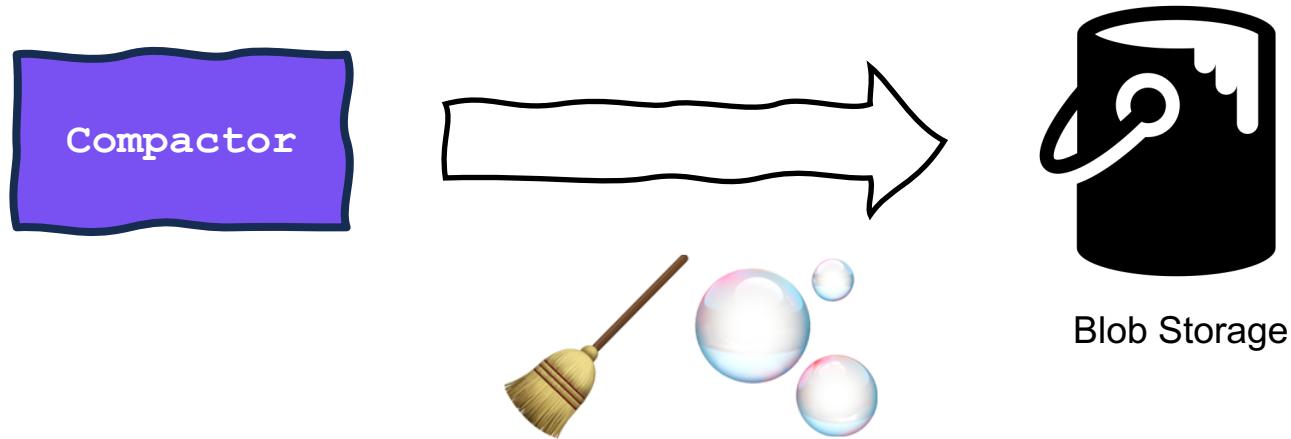


My boss (checking the Azure budget forecast over breakfast):



Block Storage Cleanup

Source of increased costs tracked down to a component called the **Compactor**



Compactor performs:

- Retention
- Downsampling
- **Deletion**

Compactor Cleanup

Compactor synchronizes block metadata every 5 minutes

This involves:

- Deleting blocks marked for deletion
- Deleting **partially uploaded blocks**

Can happen if a receiver dies during block upload

The happy path

```
ts=2023-10-31T17:35:12.240705889Z caller=fetcher.go:487 level=info
component=block.BaseFetcher msg="successfully synchronized block metadata"
duration=97.061446ms duration_ms=97 cached=157 returned=48 partial=1
```

```
ts=2023-10-31T17:35:12.240743623Z caller=clean.go:34 level=info
msg="started cleaning of aborted partial uploads"
```

```
ts=2023-10-31T17:35:12.240758082Z caller=clean.go:49 level=info msg="found
partially uploaded block; marking for deletion"
block=01HDY61KHDC82D9P6B44EBRQC3
```

```
ts=2023-10-31T17:35:12.250571421Z caller=clean.go:59 level=info
msg="deleted aborted partial upload" block=01HDY61KHDC82D9P6B44EBRQC3
thresholdAge=48h0m0s
```

```
ts=2023-10-31T17:35:12.271540804Z caller=clean.go:61 level=info
msg="cleaning of aborted partial uploads done"
```

A puzzle

```
ts=2023-04-01T12:30:32.987868066Z caller=fetcher.go:487
level=info component=block.BaseFetcher msg="successfully synchronized block metadata" duration=99.576361ms duration_ms=99
cached=153 returned=51 partial=1289
```

```
ts=2023-05-02T04:17:03.080862067Z caller=fetcher.go:487
level=info component=block.BaseFetcher msg="successfully synchronized block metadata" duration=99.576361ms duration_ms=99
cached=153 returned=51 partial=1567
```

```
ts=2023-05-29T22:19:57.387263016Z caller=fetcher.go:487
level=info component=block.BaseFetcher msg="successfully synchronized block metadata" duration=99.576361ms duration_ms=99
cached=153 returned=51 partial=2293
```

Marked for deletion, but not deleted?

```
ts=2023-10-31T15:35:12.623129229Z caller=clean.go:49 level=info
msg="found partially uploaded block; marking for deletion"
block=01HDXZ5WS4KJGNAKW6B0Z57HM1
```

```
ts=2023-10-31T15:35:12.634064598Z caller=clean.go:49 level=info
msg="found partially uploaded block; marking for deletion"
block=01HDXZ5W9FKDWXTPBW6RN7MEAK
```

```
ts=2023-10-31T15:35:12.645055937Z caller=clean.go:49 level=info
msg="found partially uploaded block; marking for deletion"
block=01HDXZ5WSPJFZVKTT92Z93C86G
```

Blocks in Storage



KubeCon



CloudNativeCon

North America 2023

Location: [thanos-blocks](#) / [01HA821RE2GW64MSH5628ACKTY](#) / chunks

Name
<input type="checkbox"/> 01HA821QZ81A9JBGDR130748DY
<input type="checkbox"/> 01HA821RE2GW64MSH5628ACKTY
<input type="checkbox"/> 01HA822MHAPH4K4VJKWB58P6BN
<input type="checkbox"/> 01HA823ATRHPEJFZB779G7ZBNN
<input type="checkbox"/> 01HA823T2JM7KCCR3DM3JS3N0Z
<input type="checkbox"/> 01HA823TZJC9KFB7GH304A5XEH
<input type="checkbox"/> 01HA823WQWVHF8RS9YX97RE0KR
<input type="checkbox"/> 01HA8242WGCH5TZ967XX9T5Y9Y
<input type="checkbox"/> 01HA8244ZVMZCQJA3WKW1GKWN2
<input type="checkbox"/> 01HA8249PB194VEZZ4P1FRGS4J
<input type="checkbox"/> 01HA824F13PG3RMMTP7C6DC5GE

Name
<input type="checkbox"/> [..]
<input type="checkbox"/> chunks
<input type="checkbox"/> index
<input type="checkbox"/> meta.json

Name
<input type="checkbox"/> [..]
<input type="checkbox"/> 000001
<input type="checkbox"/> 000002
<input type="checkbox"/> 000003



Flat vs Hierarchical Namespace

Blobs live in a flat namespace

Slashes in names can be used to *simulate* directory structure

But it's really an illusion

```
└─ ~/blob/flat
    └─ tree

    .
    └─ 01HA821QZ81A9JBGDR130748DY/chunks/000001
    └─ 01HA821QZ81A9JBGDR130748DY/chunks/000002
    └─ 01HA821QZ81A9JBGDR130748DY/chunks/000003
    └─ 01HA821QZ81A9JBGDR130748DY/index
    └─ 01HA821QZ81A9JBGDR130748DY/meta.json

    0 directories, 5 files
```

Flat vs Hierarchical Namespace

Azure has an optional feature for Blob storage: ***hierarchical namespace***

Data Lake Storage

Hierarchical namespace

Enabled

Blob service

Hierarchical namespace

Disabled

```
~/blob/hierarchical
> tree
.
└── 01HA821QZ81A9JBGDR130748DY
    ├── chunks
    │   ├── 000001
    │   ├── 000002
    │   └── 000003
    └── index
    └── meta.json

2 directories, 5 files
```

Flat vs Hierarchical Namespace

Name
<input type="checkbox"/> 01HAA39J6KGZX5YBYX0WZ4PJ1
<input type="checkbox"/> 01HAA39J6MZA6MZEH4FH7BKCPT
<input type="checkbox"/> 01HAA39J6NJQE945CDRQR67FG
<input type="checkbox"/> 01HAAA59EH18BARNFSRBXSYR3V
<input type="checkbox"/> 01HAAA59EJR7H2WD1743TX54P7
<input type="checkbox"/> 01HAAA59EK6SQQFDC62Y0DG1RR
<input type="checkbox"/> 01HAAH10PZ31WR0CK1V78ED4ZN
<input type="checkbox"/> 01HAAH10Q0CJKWR33Q5F5YK0G2

Name
<input type="checkbox"/> [..]
<input type="checkbox"/> chunks

Name
<input type="checkbox"/> [..]

```
~/blob/hierarchical
> tree
.
└── 01HA821QZ81A9JBGDR130748DY
    └── chunks

2 directories, 0 files
```

The Answer

Deletion of blobs with HNS enabled leaves behind empty directories

Thanos interpreted these empty directories as partial uploads

=> thousands of delete requests for non-existent blocks every 5 minutes

...but we still got charged for them!



Blob storage: take-home message



 Data Lake Storage

86% reduction in storage account costs

 Blob service

Hierarchical namespace

Enabled

Disabled

Summing Up



KubeCon



CloudNativeCon

North America 2023

Use **Ketama** hash ring: no excuses!



Enable **Replication** if you run a lot of recording rules

Leverage **Hard Tenancy** and **Active Series Limiting** to mitigate effect of cardinality explosions

Be wary of overly helpful automation and cloud features!

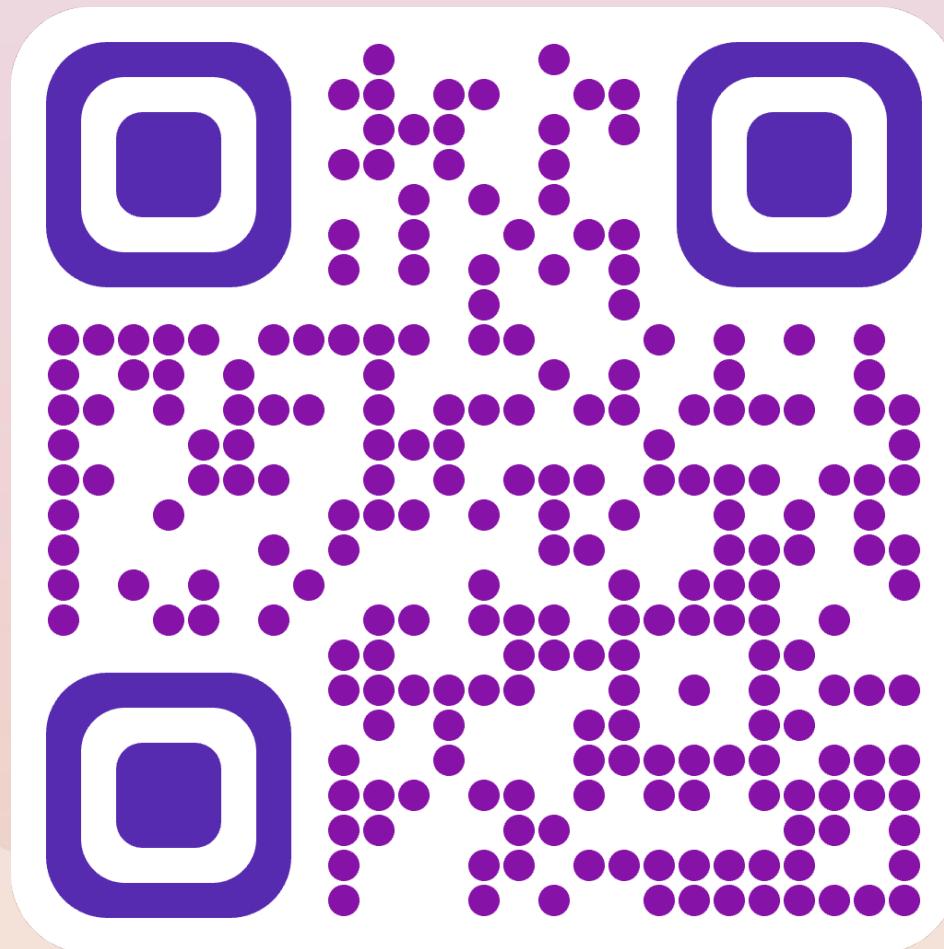


KubeCon



CloudNativeCon

North America 2023



Please scan the QR Code above
to leave feedback on this session