



# Beyond Billions: Operating Thanos, Prometheus & OpenTelemetry at Trillion-scale

**NARENDRA SANIKOMMU**, Senior Software Engineer, NVIDIA

CNCF OBSERVABILITY SUMMIT NA / MINNEAPOLIS / MAY 2026



# What "trillion-scale" actually looks like

**~3.5T** Samples ingested /  
day

(~40M Samples ingested / sec)



**~2B** Active timeseries



***"At this scale, defaults break. Every knob matters."***

# Five lessons I wish I had on day one

**01** Paved roads beat wikis *Config is a contract, not a wiki.*

**02** Start multi-tenant, or pay later *Isolation is logical, not physical.*

**03** Queries will eat your ingestion *Dual-duty components hide outages.*

**04** Cardinality costs more on read than write *Budget at write, enforce at read.*

**05** Your OTel Collector is a distributed system *Draw it, or debug it.*

# Paved roads beat wikis

*"Recommended configs nobody reads are not recommendations — they're evidence."*

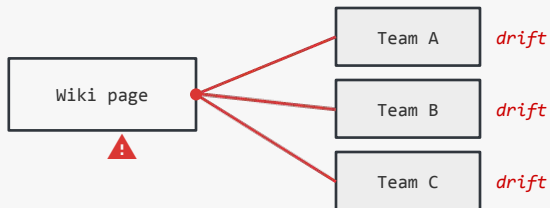
## THE TAKEAWAY

- Ship configs as versioned artifacts
- Make the golden path the easy path
- Telemetry on who runs which version

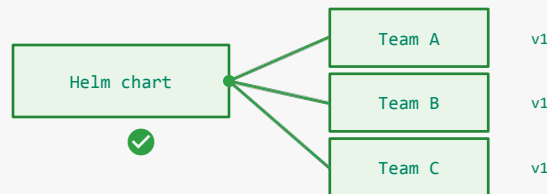
## HARD-WON LESSON

We published the right way to configure Prometheus clients. Nobody read the docs. Many incidents ended with "they weren't using the recommended settings." The fix: remove the option to drift.

### BEFORE



### AFTER



# Start multi-tenant, or pay later

*"One cluster per team optimizes for day one and punishes every day after."*

## THE TAKEAWAY

- Shared cluster, tenant boundary at Receiver
- Per-tenant quotas, dashboards, alerts
- Blast radius via limits, not hardware
- Upgrades happen once, not N times

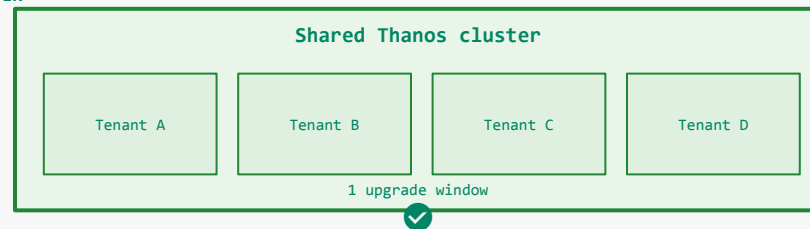
## HARD-WON LESSON

We gave each team its own Thanos cluster. Ten clusters meant ten upgrade windows and ten drift profiles. Consolidating was a year of migration work.

### BEFORE



### AFTER



# Queries will eat your ingestion

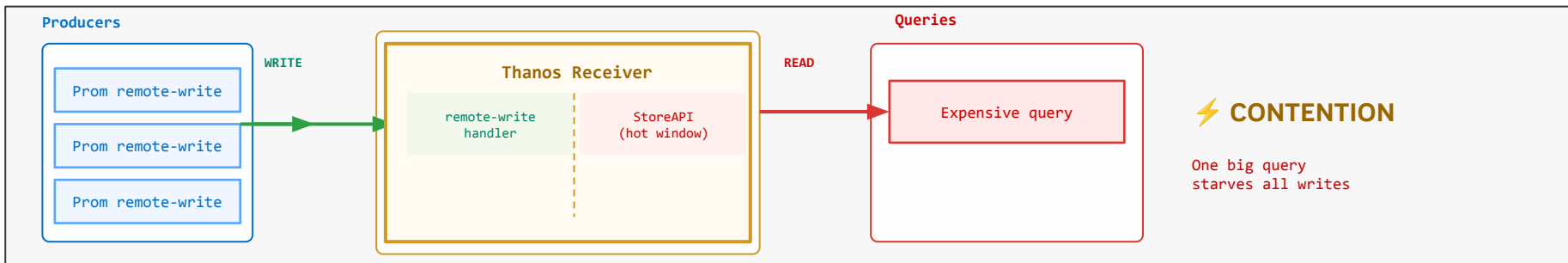
*"Thanos Receivers are dual-role. Rate limits belong there."*

## THE TAKEAWAY

- Protect at the point of contention
- Separate SLIs for ingest vs. live-query
- Circuit-break expensive queries at frontend

## HARD-WON LESSON

Remote-write lag spiked with no increase in write volume. The culprit: expensive queries hitting the Receivers' StoreAPI — one big query starved every producer.



# Cardinality costs more on read than write

*"Writes pay for cardinality once. Queries pay every single time."*

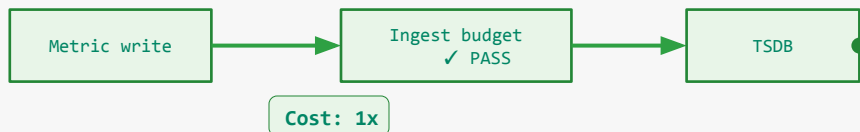
## THE TAKEAWAY

- One budget, two enforcement points
- Surface query cost in the response
- Cardinality leaderboards — social pressure works

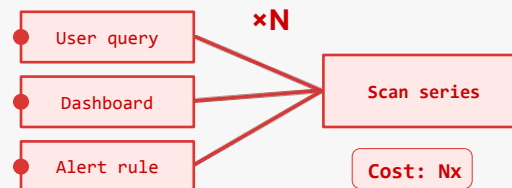
## HARD-WON LESSON

We treated cardinality as an ingest problem. Then queries couldn't handle it. Writes pay once; queries pay on every request.

### WRITE PATH



### QUERY PATH



# Your OTEL Collector is a distributed system

*"Out-of-order samples = your pipeline has more than one brain."*

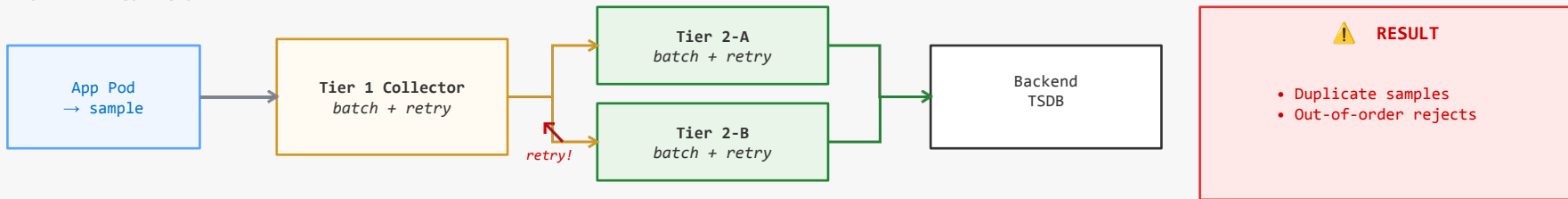
## THE TAKEAWAY

- One place, one decision for batching/retries
- Draw your topology or you don't operate it
- SLO the pipeline end-to-end

## HARD-WON LESSON

We saw out-of-order rejections and duplicated series. Multiple Collector tiers were each batching and retrying independently. Same sample, two labels, clock skew everywhere.

## MULTI-TIER COLLECTOR PIPELINE



# Takeaways

- 01** Config is a contract. Contracts live in code, not wikis.
- 02** Multi-tenancy is a day-zero decision. Retrofitting costs 10x.
- 03** If a query can hurt ingestion, your ingestion SLO is fiction.
- 04** Cardinality is a lifecycle, not a gate — budget at write, enforce at read.
- 05** Your OTel Collector is a distributed system. Draw it, or debug it.

# Thank You

## ENGINEERING

Anisha Nagarajan

Baig Awan

Charlie Cantoni

Deepak Choudhary

Dheeraj Kapur

Satish Kumar Gandham

Swetha Banagiri

## MANAGEMENT

Chris Galtenberg

Krishna Beck

Satish Dandu

Teja Thotapalli

Zachary Taylor

# Questions?

## Narendra Sanikommu

*Senior Software Engineer, NVIDIA*



[linkedin.com/in/narendrareddysanikommu](https://www.linkedin.com/in/narendrareddysanikommu)

