Processing 10M samples/second to drive smart maintenance in complex IIoT systems

Geir Engdahl - CTO, Cognite
Daniel Berqvist - Developer Advocate, Google

Strata
DATA CONFERENCE

cognite iot core

Alle | Bilder | Videoer | Nyheter | Shopping | Mer | Innstillinger | Verktøy

Alle språk ▾ | Når som helst ▾ | Alle resultater ▾

Tips: Bare søk etter **norske** resultater. Du kan spesifisere søkespråket ditt i Innstillinger

Videoer

From Edge to Cloud: How Cloud IoT Core Is Supporting Industrial IoT ...

Google Cloud Platform
YouTube - 12. apr. 2019

Cognitive IoT: Where digital meets physical

IBM Watson Internet of
YouTube - 18. des. 2015

Configure a Raspberry Pi and connect it to Watson IoT Platform

IBM Watson Internet of
YouTube - 14. mar. 2016


IVAR AASEN
16/1-D
AKER BP
21.2 vol %
18.2 °C
98.7 barg
16.9 °C
20.2 barg
1.23 vol %

Cognite

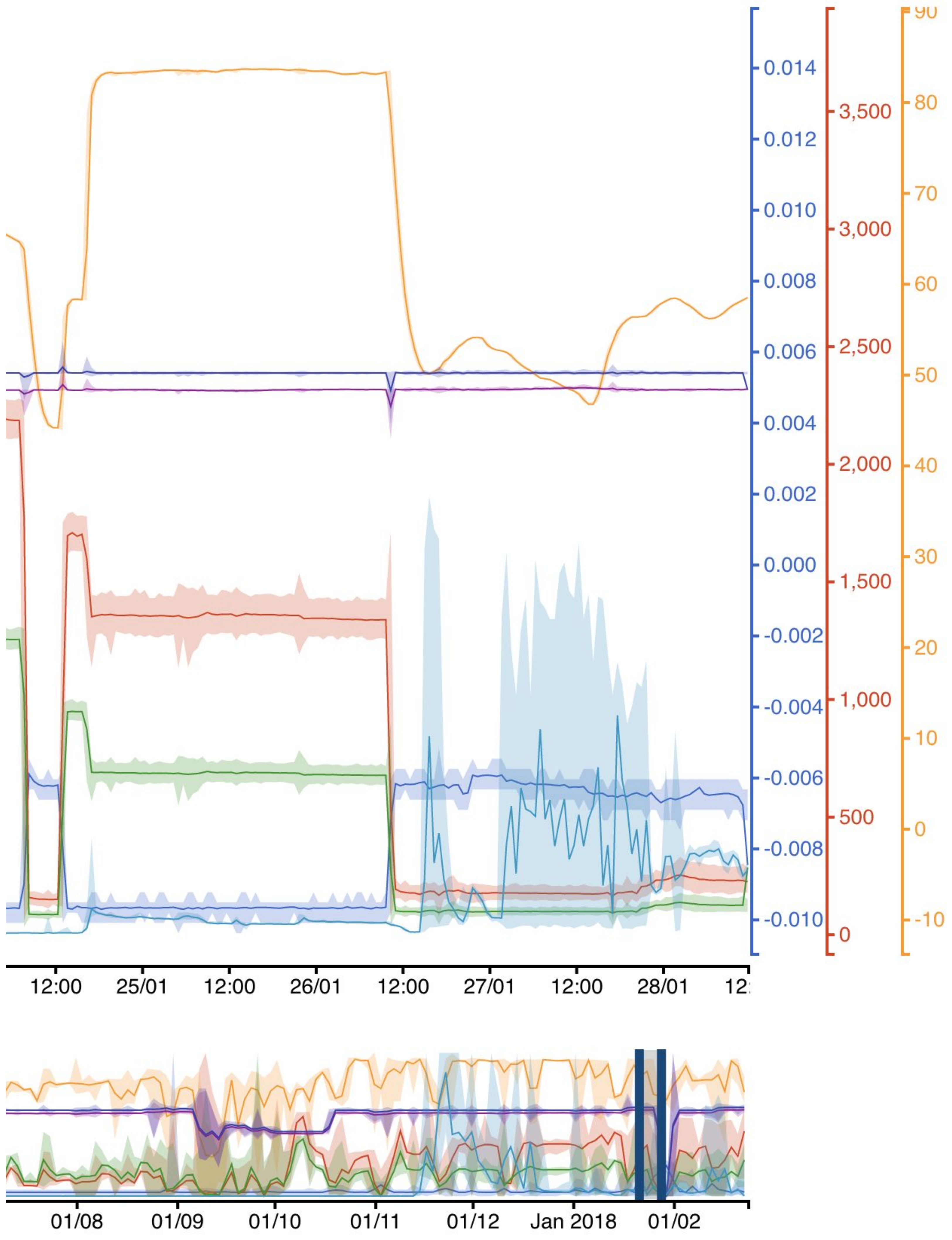
DEMO

Charting library you just saw is open-sourced

https://github.com/cognitedata/griff-react

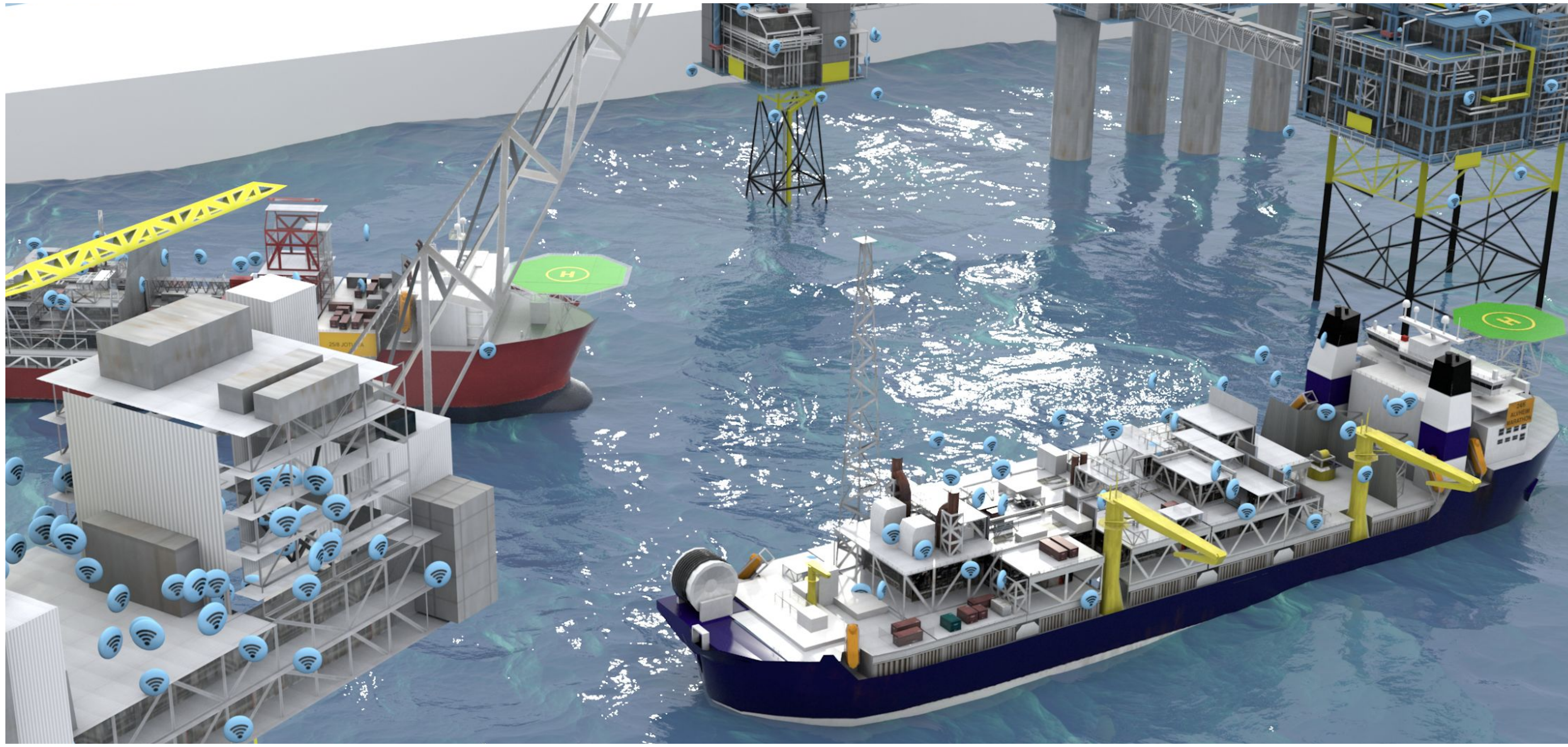
- High performance charting of large time series
- Dynamic data loading
- No tight coupling to Cognite TSDB
- Uses React and d3

```
yarn add @cognite/griff-react

Or

npm i @cognite/griff-react
```



IoT & the data explosion

50 billion devices connected to internet by 2023 according to Statista (2018) [1]. Cognite currently covers 500 000 sensors, each producing one GB every two years



[1] https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ (2018)

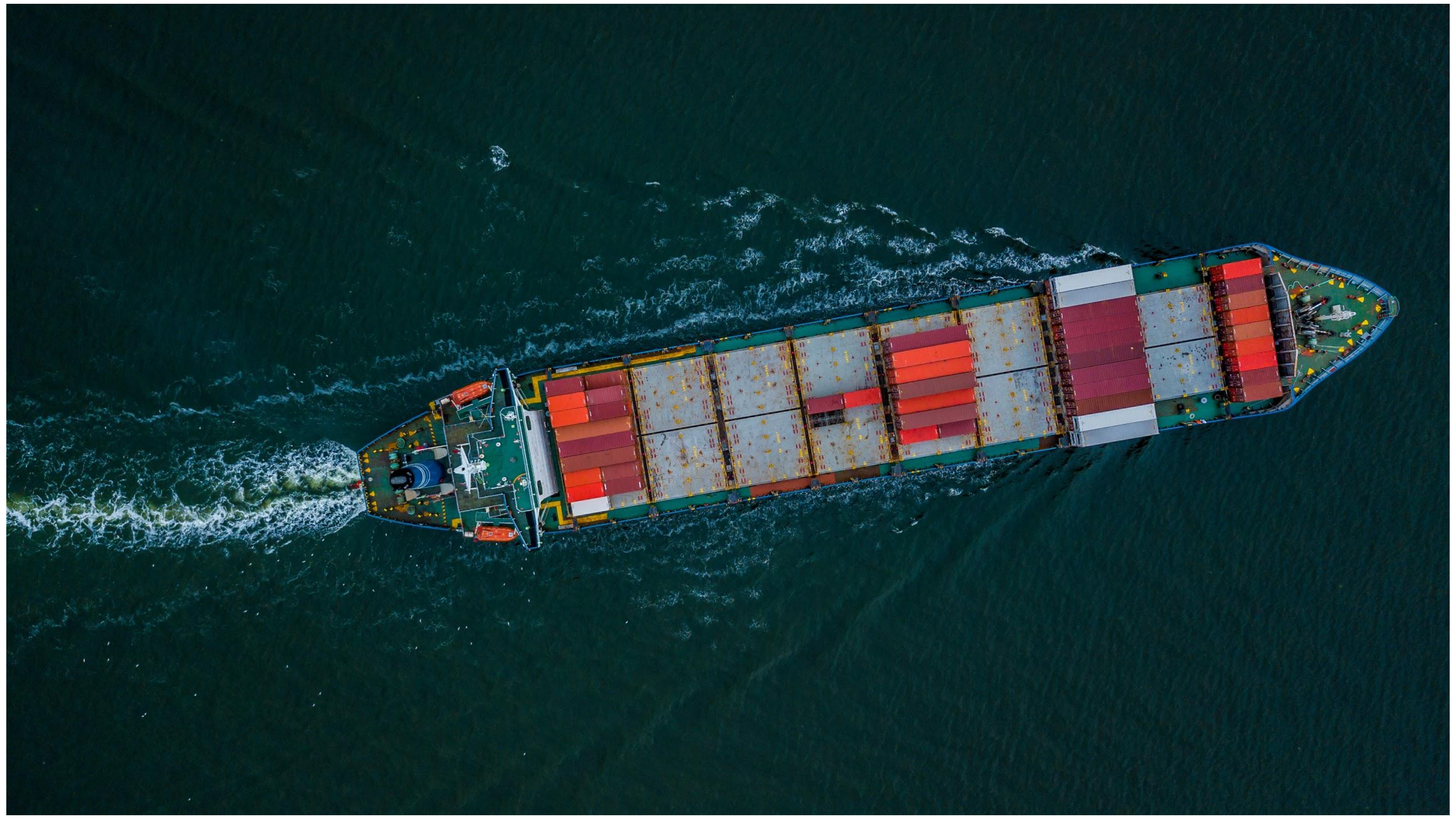
Time series requirements

- Robustness
- High volume of reads and writes
- Low latency
- Arbitrary granularity aggregates
- Efficient backfill
- Efficient sequential reads

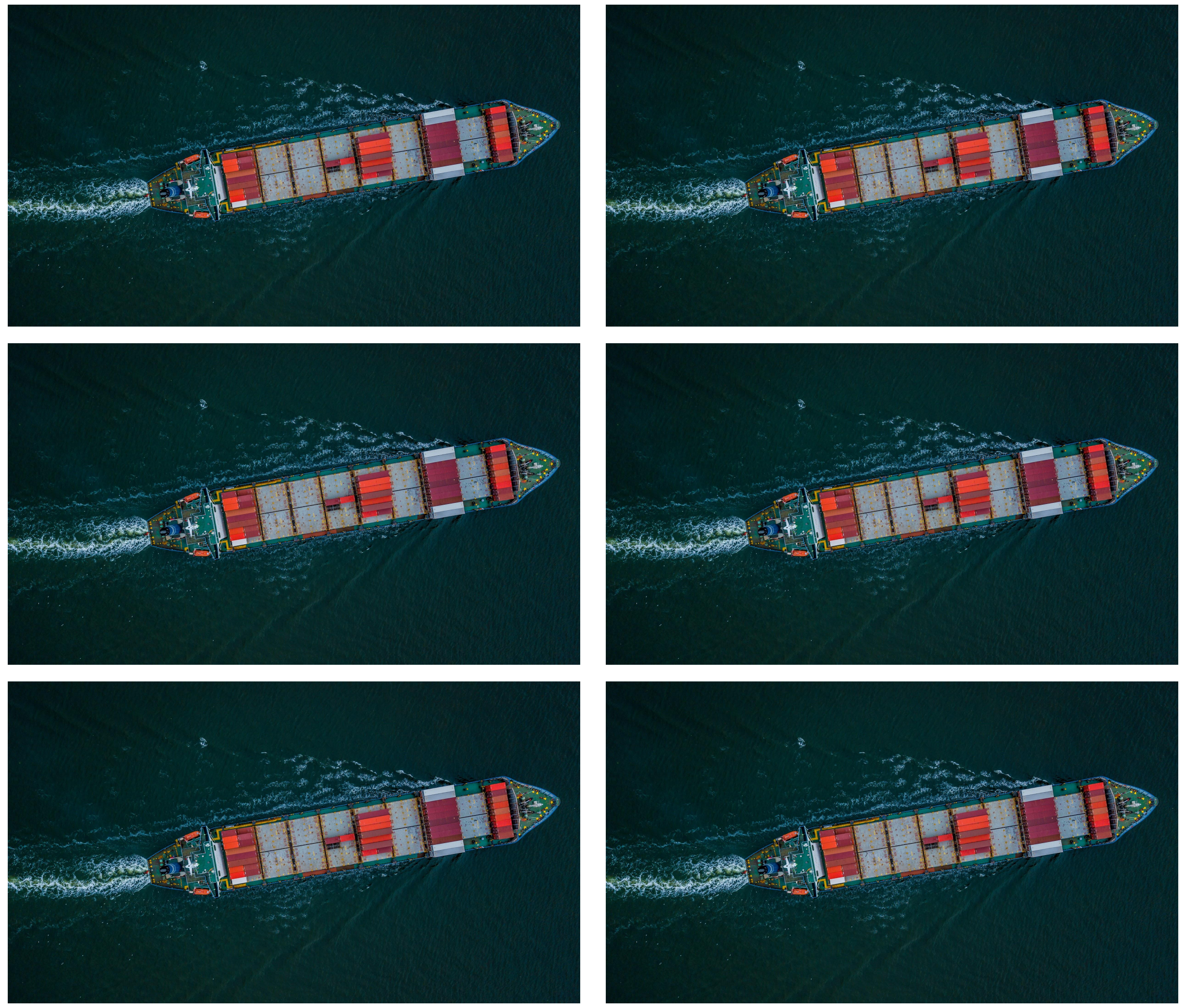
Surely there must be an off-the-shelf solution that satisfies this!

Databases for IoT - two approaches

Single node*



Horizontally scaling



* Often does master - slave, or other read-only replication, but not partitioning

OpenTSDB experiments

- No limit parameter on queries
- No batch inserts, so slow backfills
- Can lose incoming data points
- Aggregates not pre-computed on write

Disclaimer: OpenTSDB experiments from summer 2017 on version 2.3.0

The case for Cloud Bigtable

- Fully managed
- 10k writes/s per node (SSD)
- Scalable to 100s of PBs
- Can scan forward efficiently
- Column families and versioning

A brief introduction to Google Cloud Bigtable

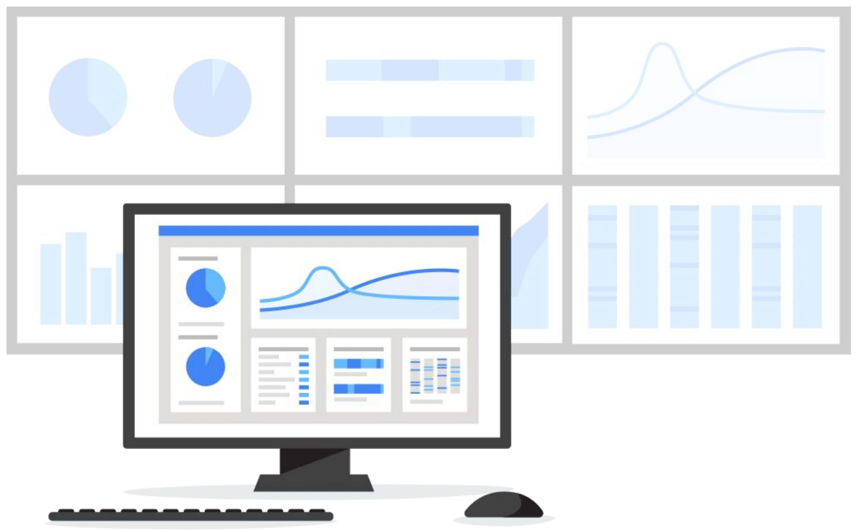
Achieve your performance goals

Single digit ms
write latency for
performance-critical apps

Serve global audiences

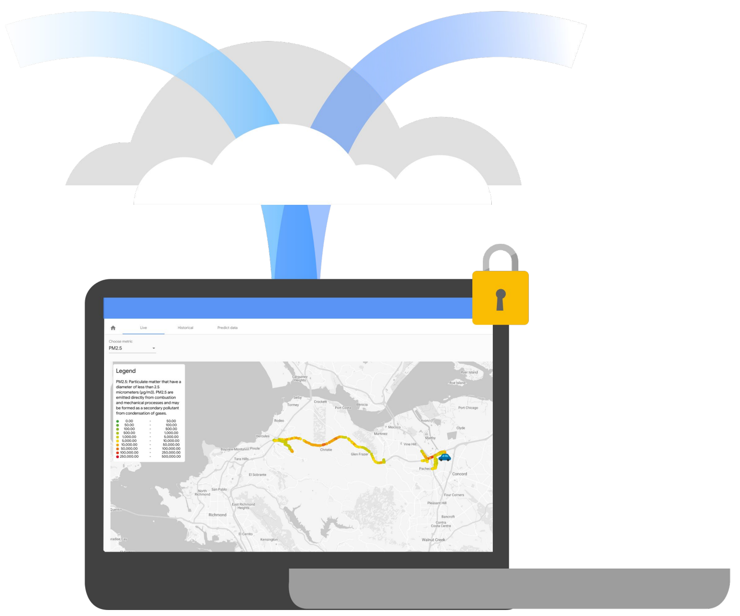
99.99% availability
across Google's
dedicated network

From DevOps to NoOps



Reduce management
effort from weeks
to minutes

Supercharge your applications



Stream, secure,
analyze and drive ML/AI

Wide-columnar data model

NoSQL (no-join) distributed key-value store, designed to scale-out

Has only **one index** (the row-key)

Supports atomic **single-row transactions**

Sparse: Unwritten cells do not take up any space

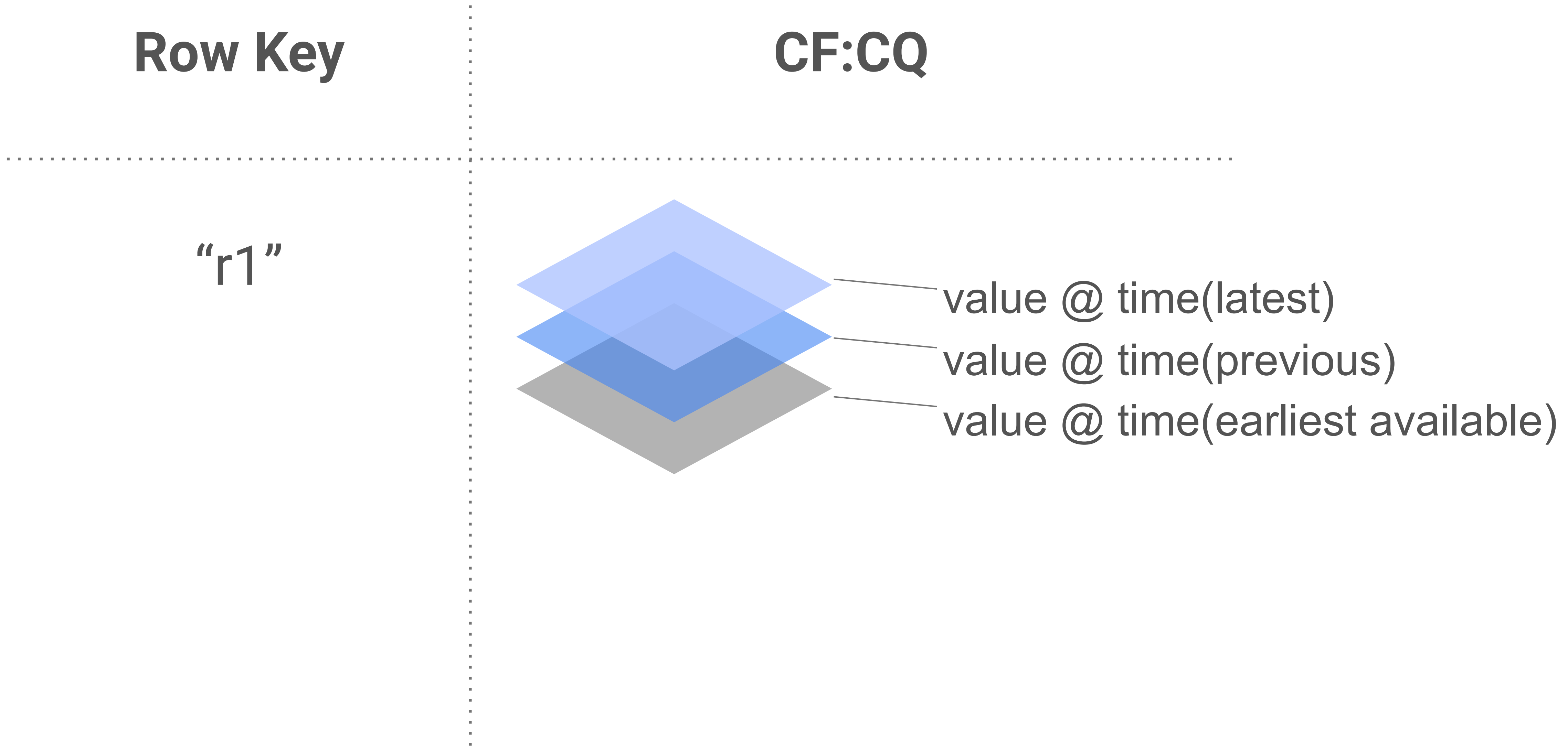
	Column-Family-1		Column-Family-2	
Row Key	*Column-Qualifier-1*	*Column-Qualifier-2*	*Column-Qualifier-1*	*Column-Qualifier-2*
r1	r1, cf1:cq1	r1, cf1:cq2	r1, cf2:cq1	r1, cf2:cq2
r2	r2, cf1:cq1	r2, cf1:cq2	r2, cf2:cq1	r2, cf2:cq2

Three-dimensional data space

Every cell is **versioned** (default is timestamp on server)

Configurable **garbage collection** retains latest N versions (or after TTL)

Expiration can be set at **column-family level**

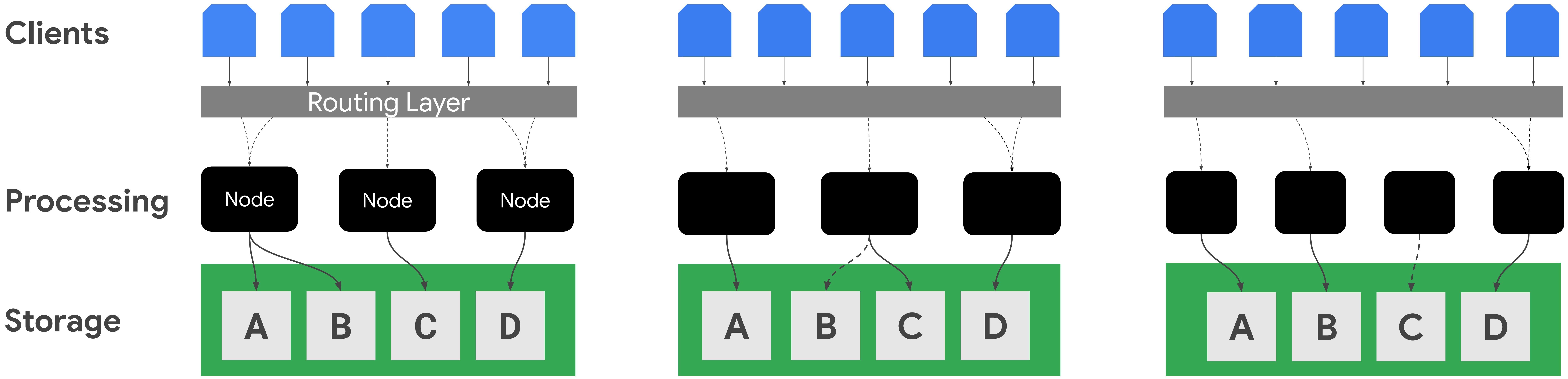


Cloud Bigtable - Optimizing throughput

Cloud Bigtable separates processing from storage through use of nodes, each of which provides access to a group of database rows

Rebalancing automatically reduces the load on highly active nodes (in this case there is a lot of activity for data group A)

User-driven resizing as needed to match data throughput targets, with no downtime



Original Setup

Rebalancing

Resizing

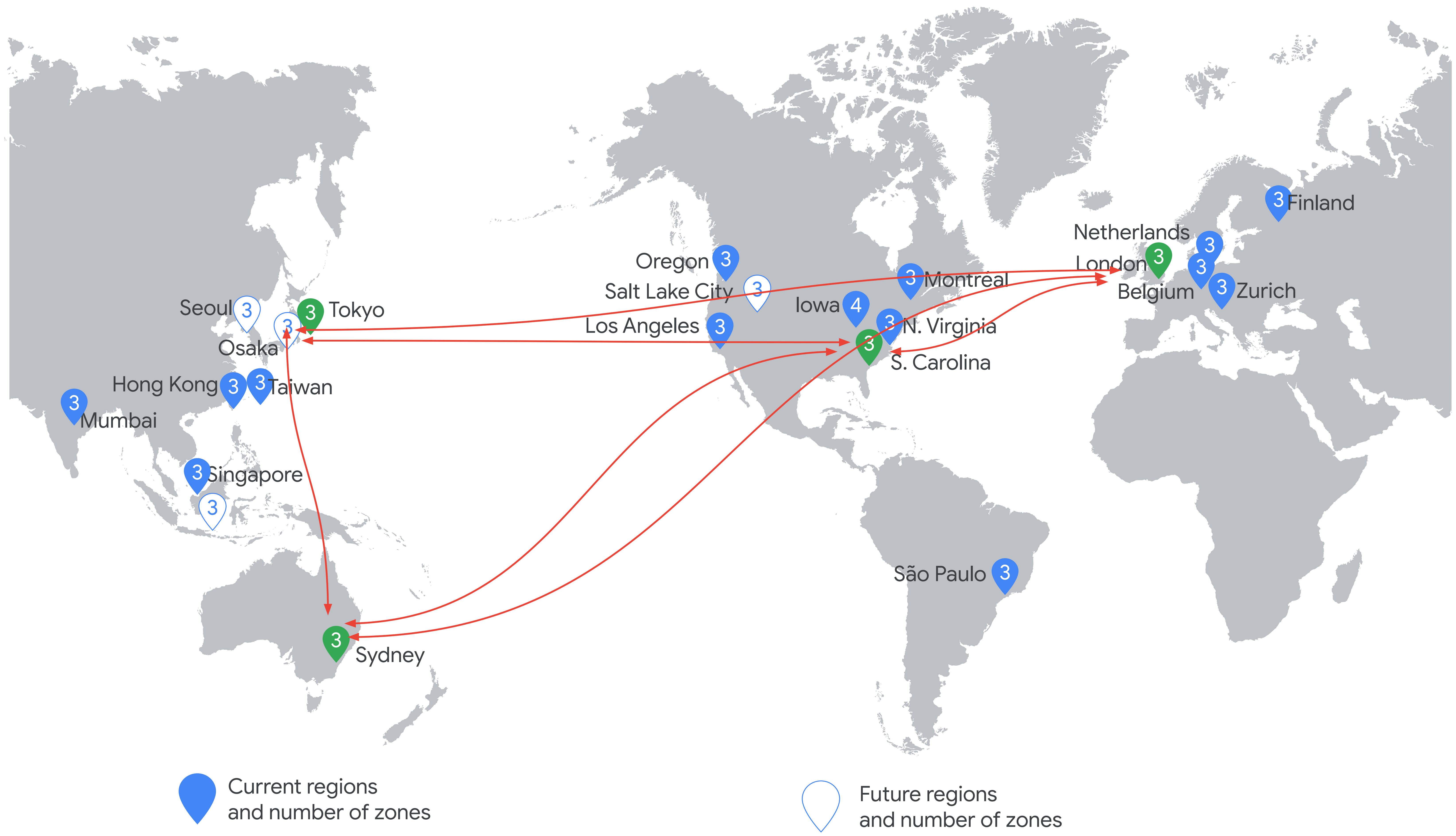
Cloud Bigtable replication

Regional replication

- SLA increased to 99.99%
- Isolate serving and analytics
- Independently scale clusters
- Automatic failover in case of a zonal failure

Global replication

- Increases durability/availability beyond one region
- Fastest region-specific access
- Option for DR replica for regulated customers



Cloud Bigtable for IoT - best practices

Recommendations for row key design	Recommendations for data column design
Use tall and narrow tables	Rows can be big but are not infinite (1000 timestamp/value pairs per row is a good rule of thumb)
Prefer rows to column versions	Keep related data in the same table; keep unrelated data in different tables
Design your row key with your queries in mind	Store data you will access in a single query in a single column family
Ensure that your row key avoids hotspotting	Don’t exploit atomicity of single rows
Reverse timestamps only when necessary	

How Cognite stores data in Cloud Bigtable

Row key ← This is the only thing you can lookup, but can also scan **forward**

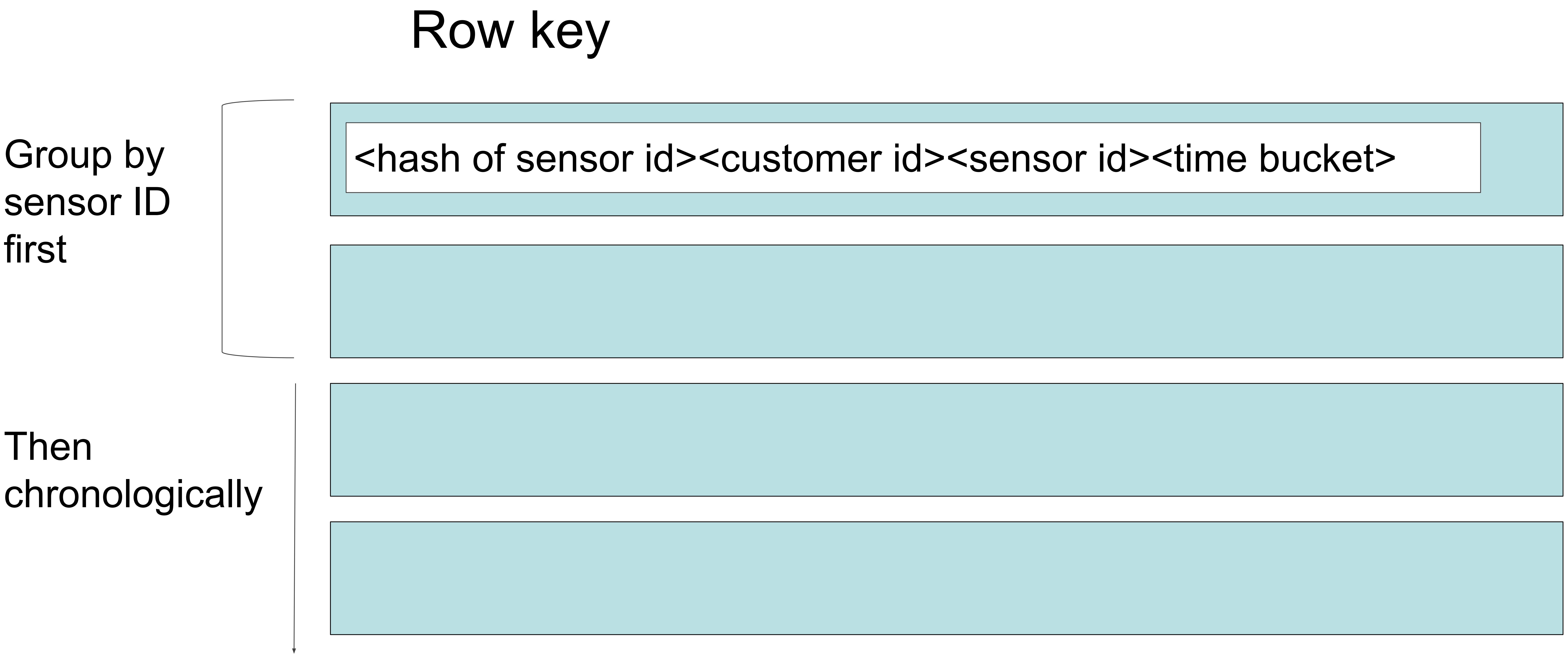
Group by customer ID, sensor ID first

Then chronologically

- “Customer1-Sensor1-2018-07-24-01”
- “Customer1-Sensor1-2018-07-24-02”
- “Customer1-Sensor2-2018-01-01-01”
- “Customer1-Sensor2-2018-01-01-02”

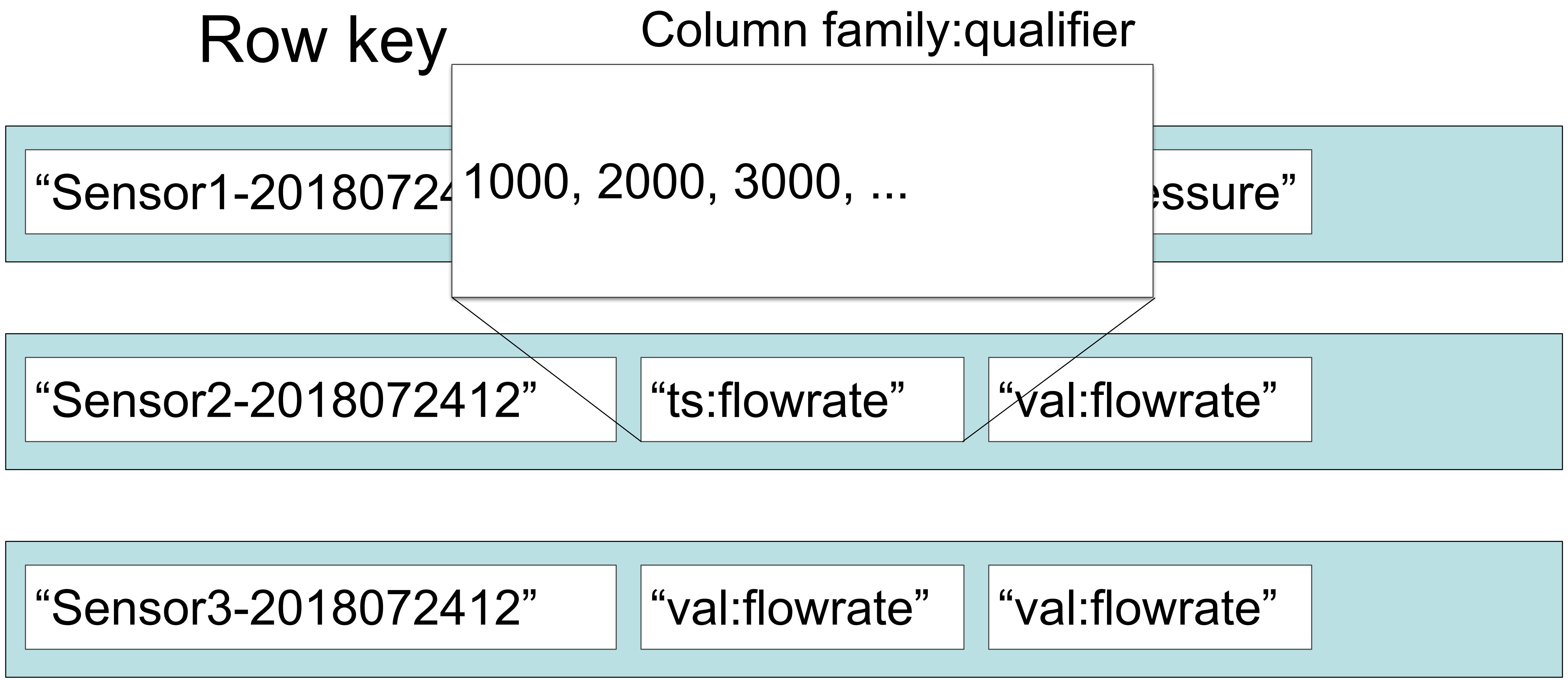

0
2.95
5.9
8.85
11.8
59
118
µOps/row/min
Average: 10.77 µOps/row/min
Total (avg over time): 10.1 KOps
04/19/2019
04/20/2019
04/21/2019
04/22/2019
04/23/2019
04/24/2019
04/25/2019

Improved key schema

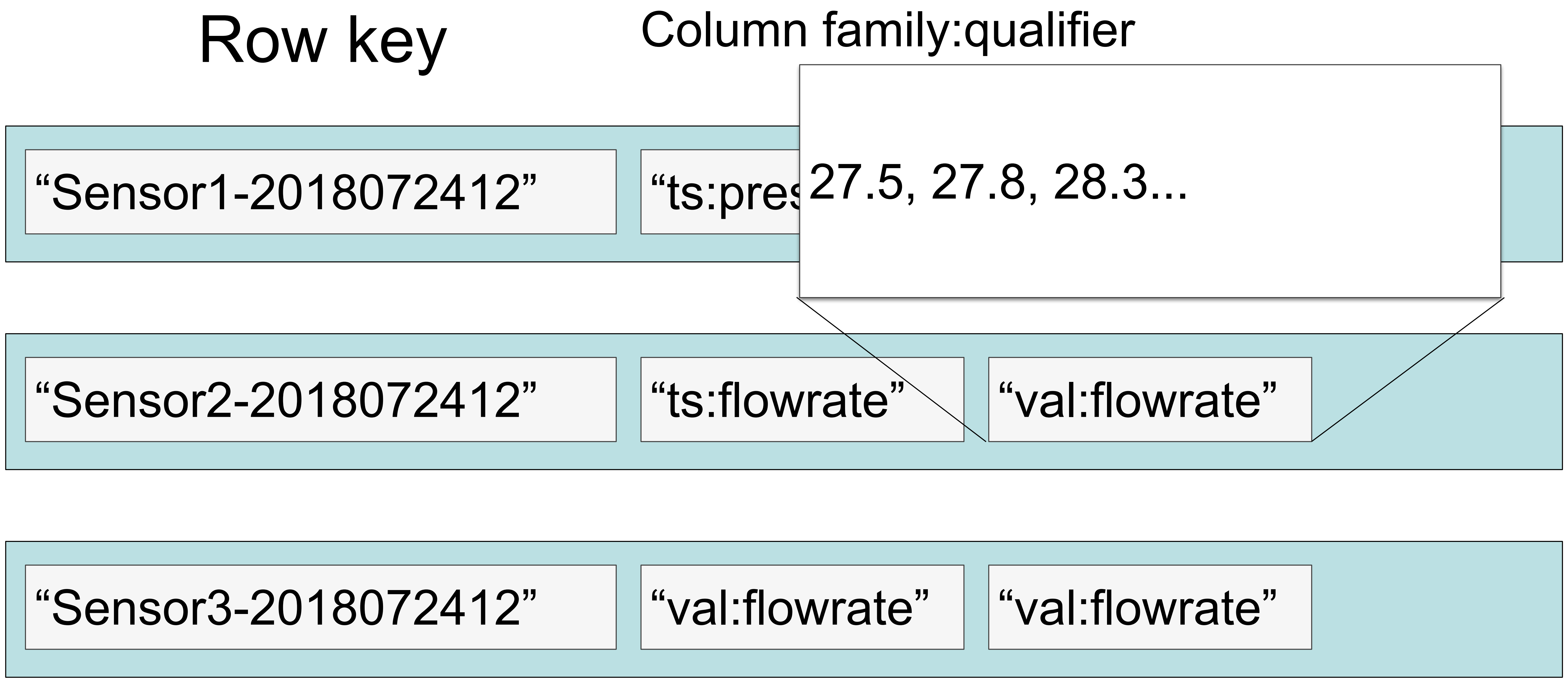



0
40.5
81
121
162
810
1620
µOps/row/min
Average: 106.95 µOps/row/min
Total (avg over time): 549.03 KOps
04/20/2019
04/21/2019
04/22/2019
04/23/2019
04/24/2019
04/25/2019
04/26/2019

How Cognite stores data in Cloud Bigtable



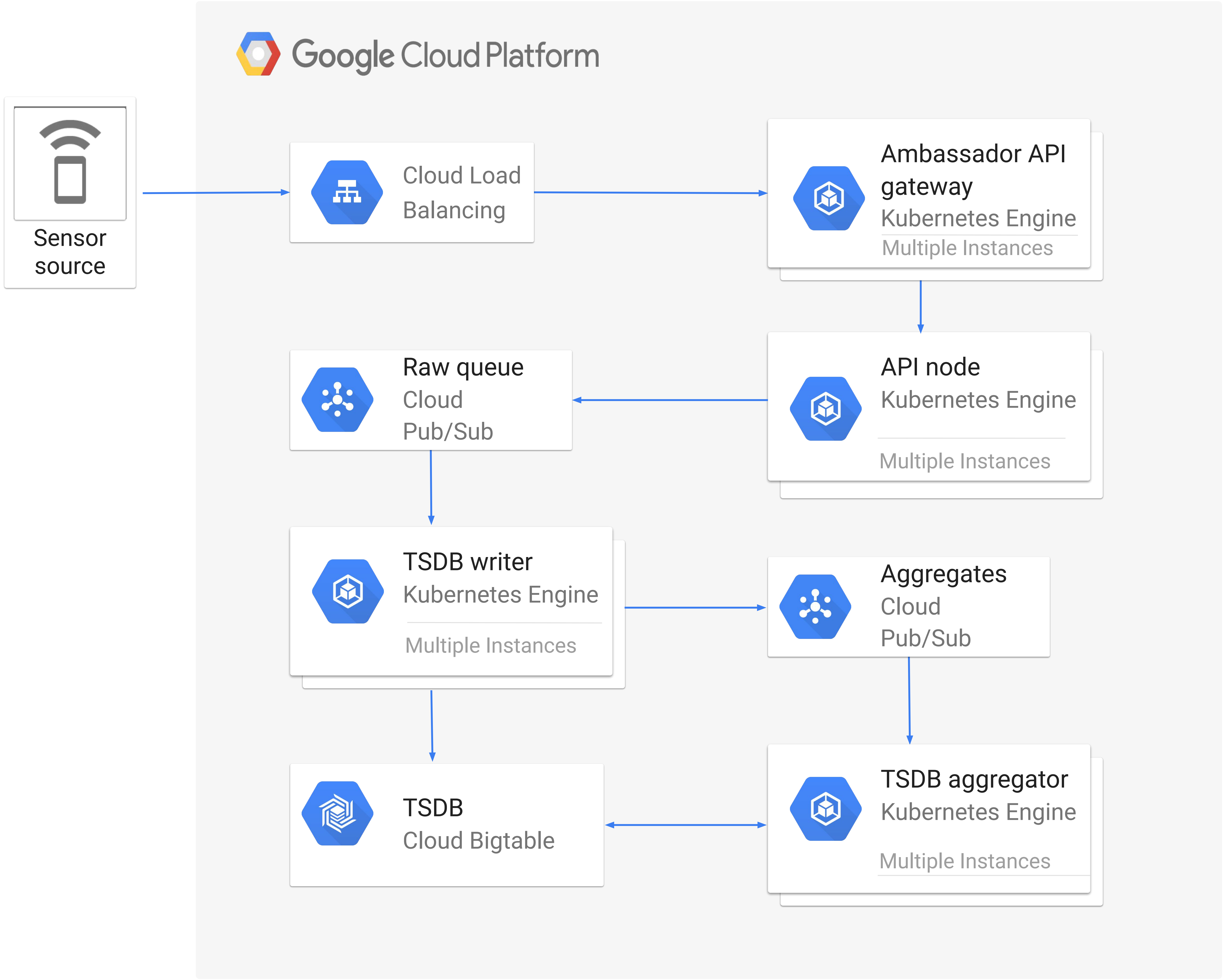
How Cognite stores data in Cloud Bigtable



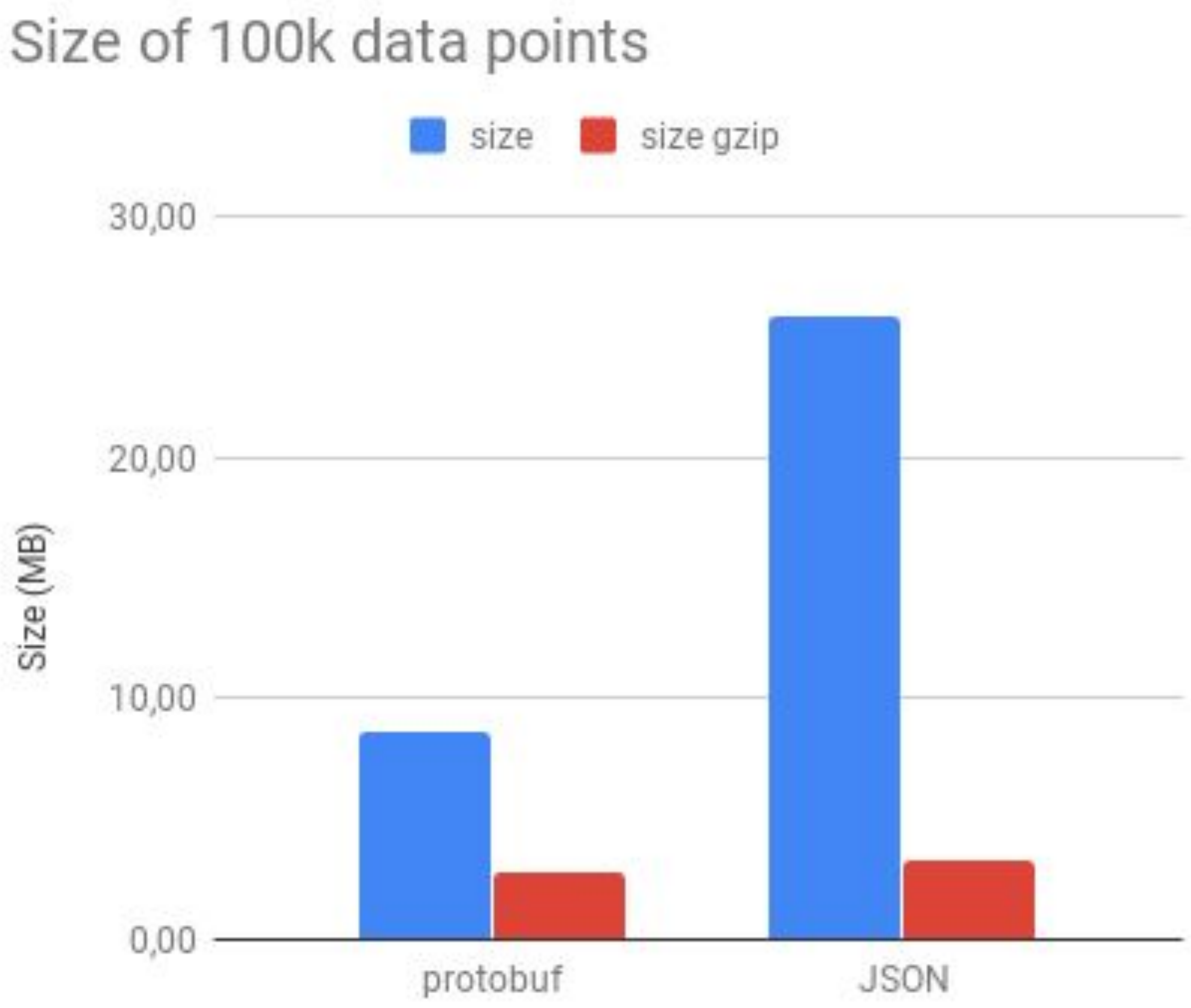
System performance

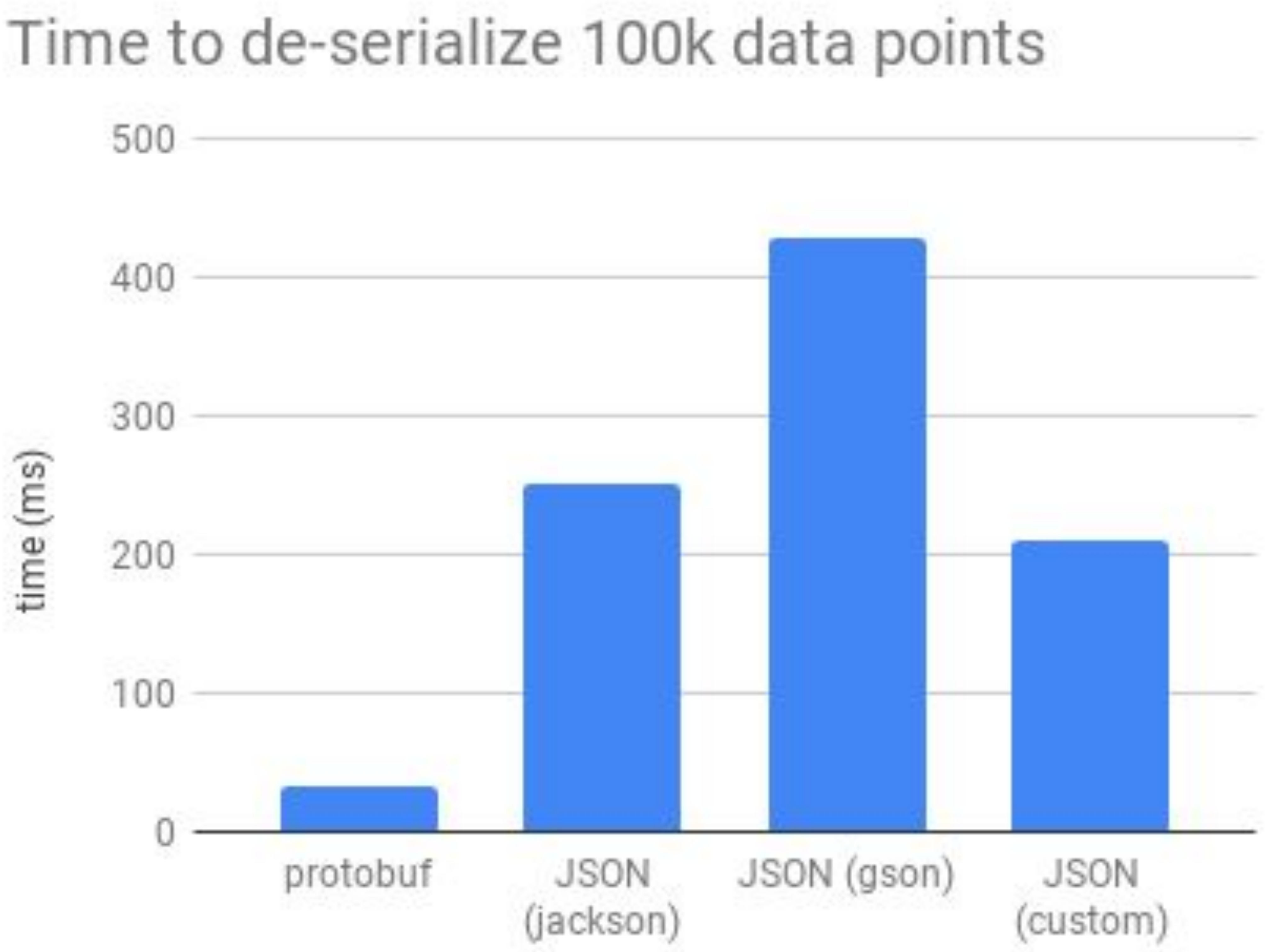
Performance:

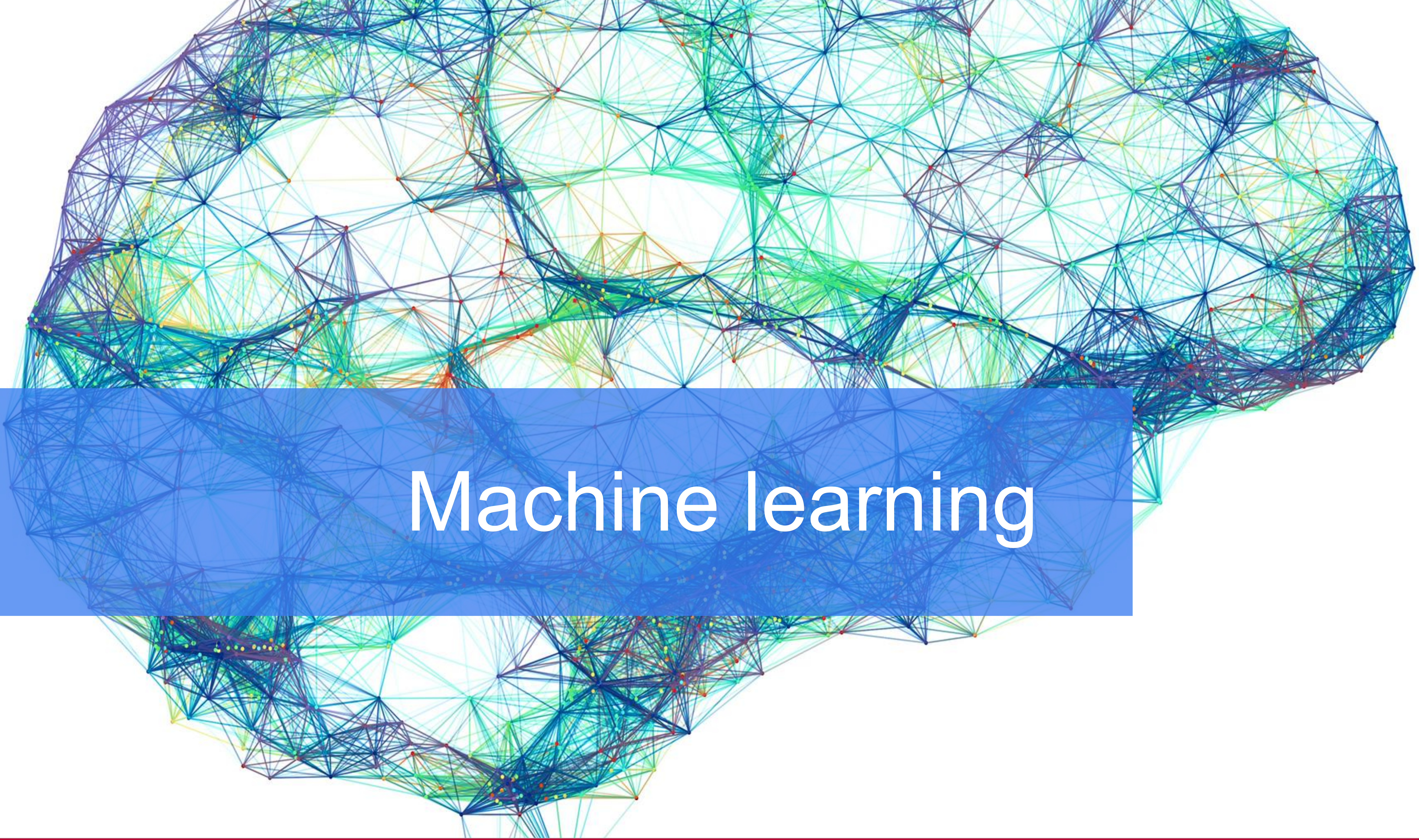
- **Throughput**: Handles up to 10M data points per second
- **Latency**: Data queryable after 200ms (99th percentile)



Protobuf vs JSON



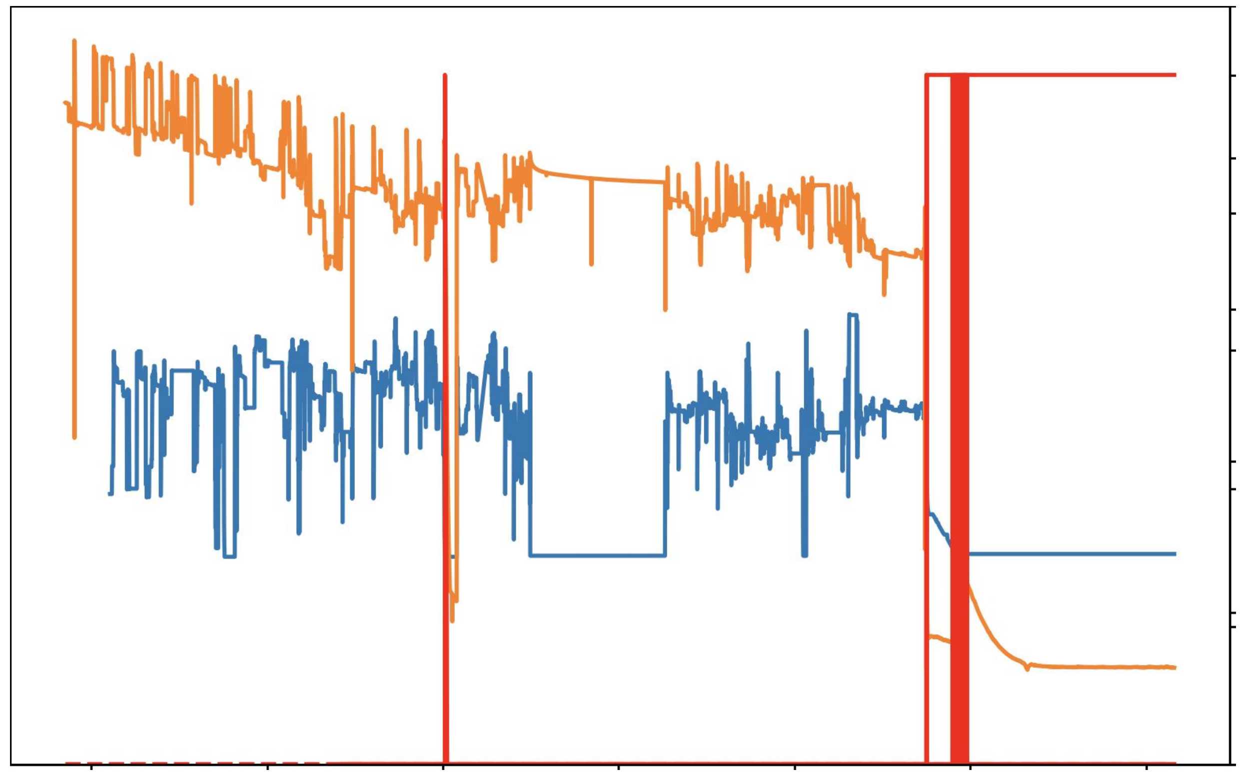




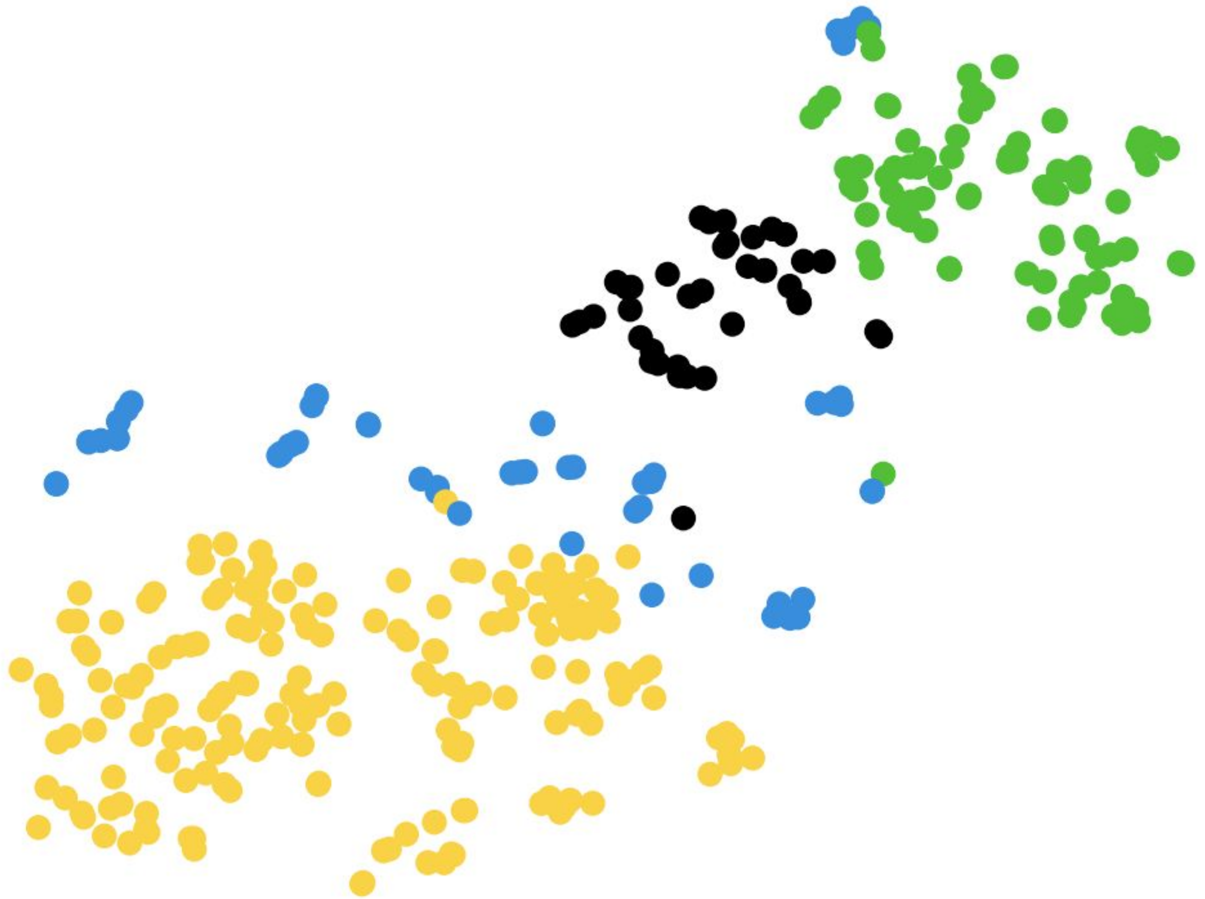
Machine learning

Unsupervised anomaly detection

Forecasting

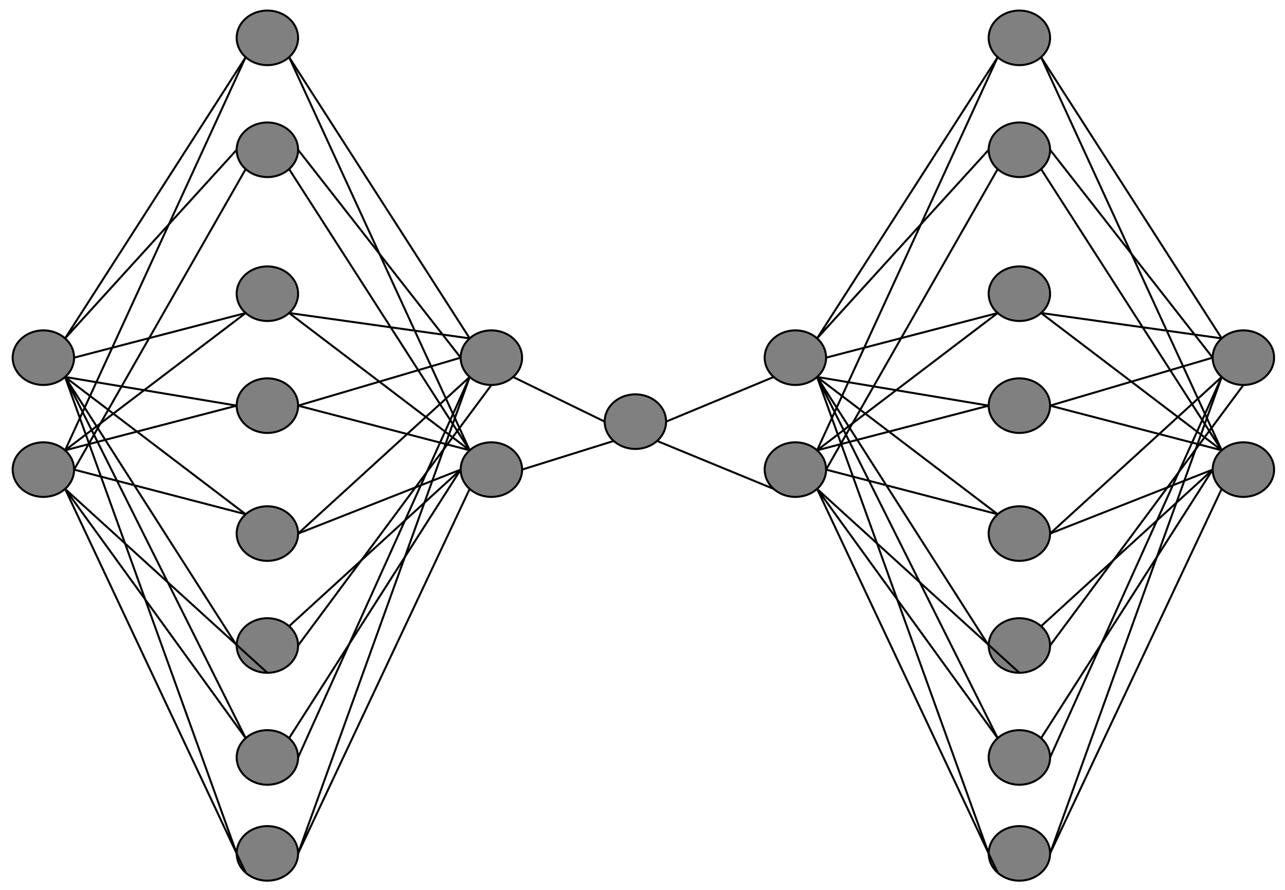


Clustering

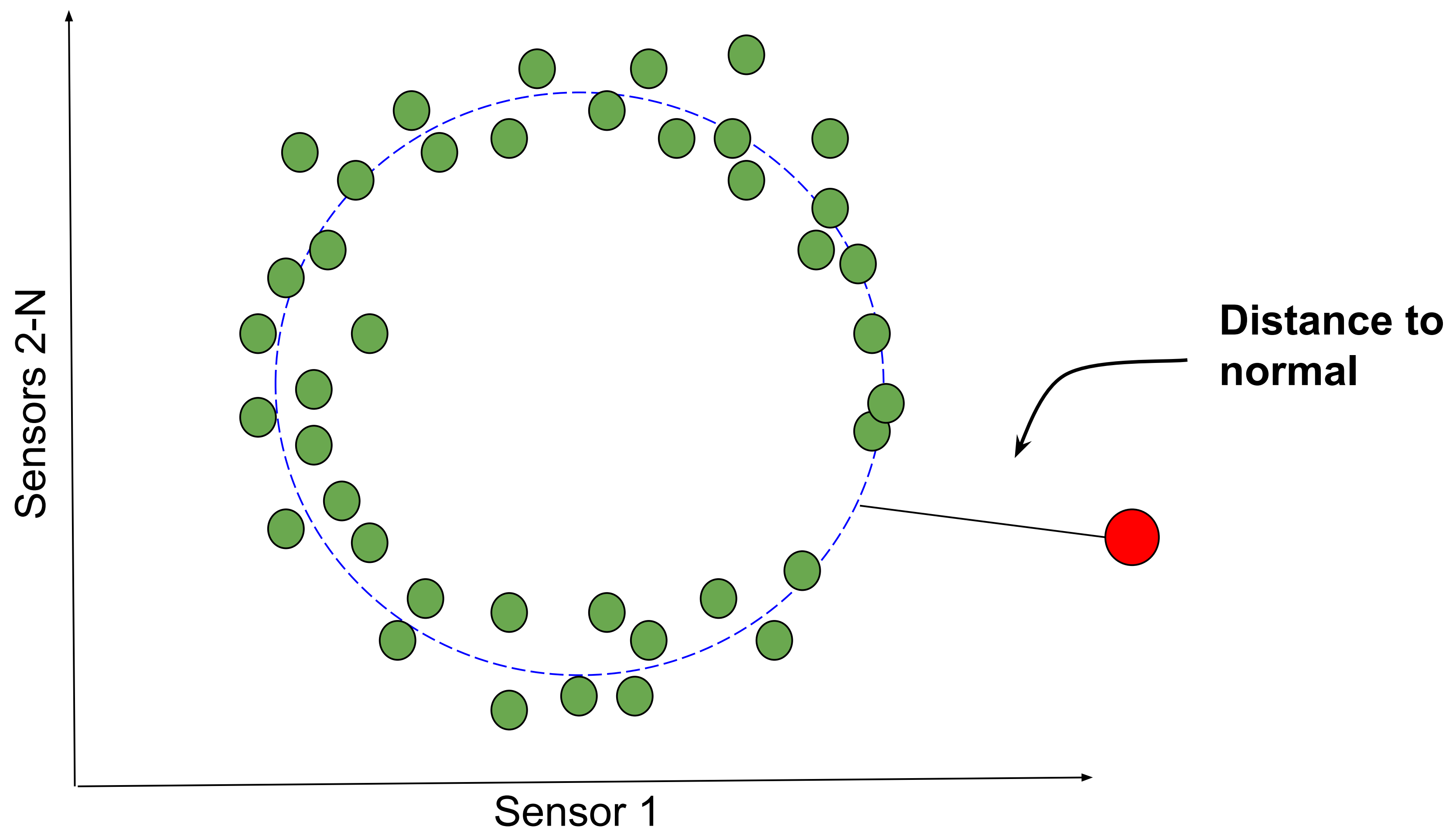


Unsupervised detection with AutoEncoders

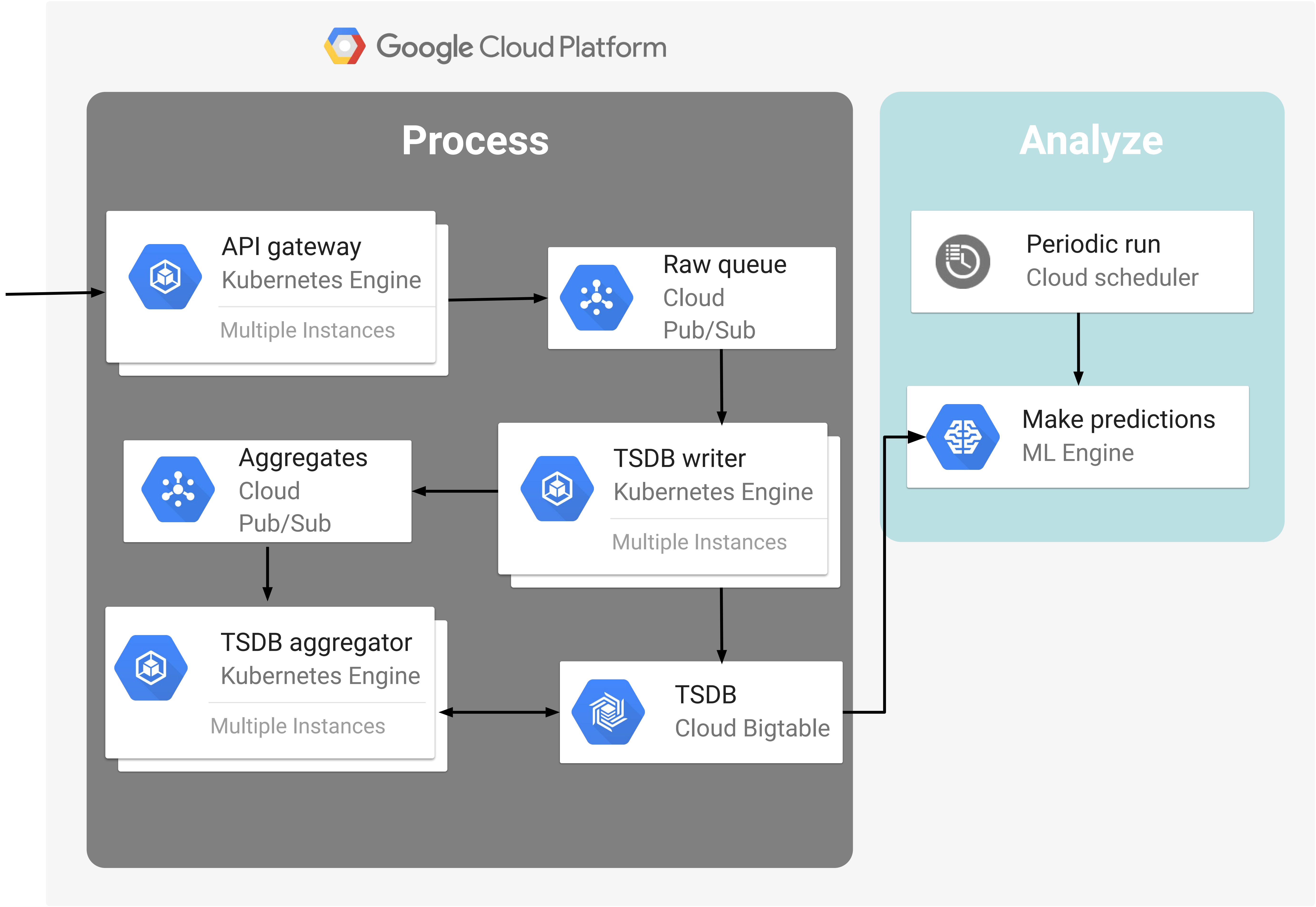
Architecture search....



… to learn a parameterization of normality



Machine learning architecture



Future improvements

- Ability to query consistent snapshots back in time
- High frequency time series
- Efficient latest data point query

Next steps

Cloud Bigtable

- cloud.google.com/bigtable
- cloud.google.com/bigtable/docs/schema-design-time-series

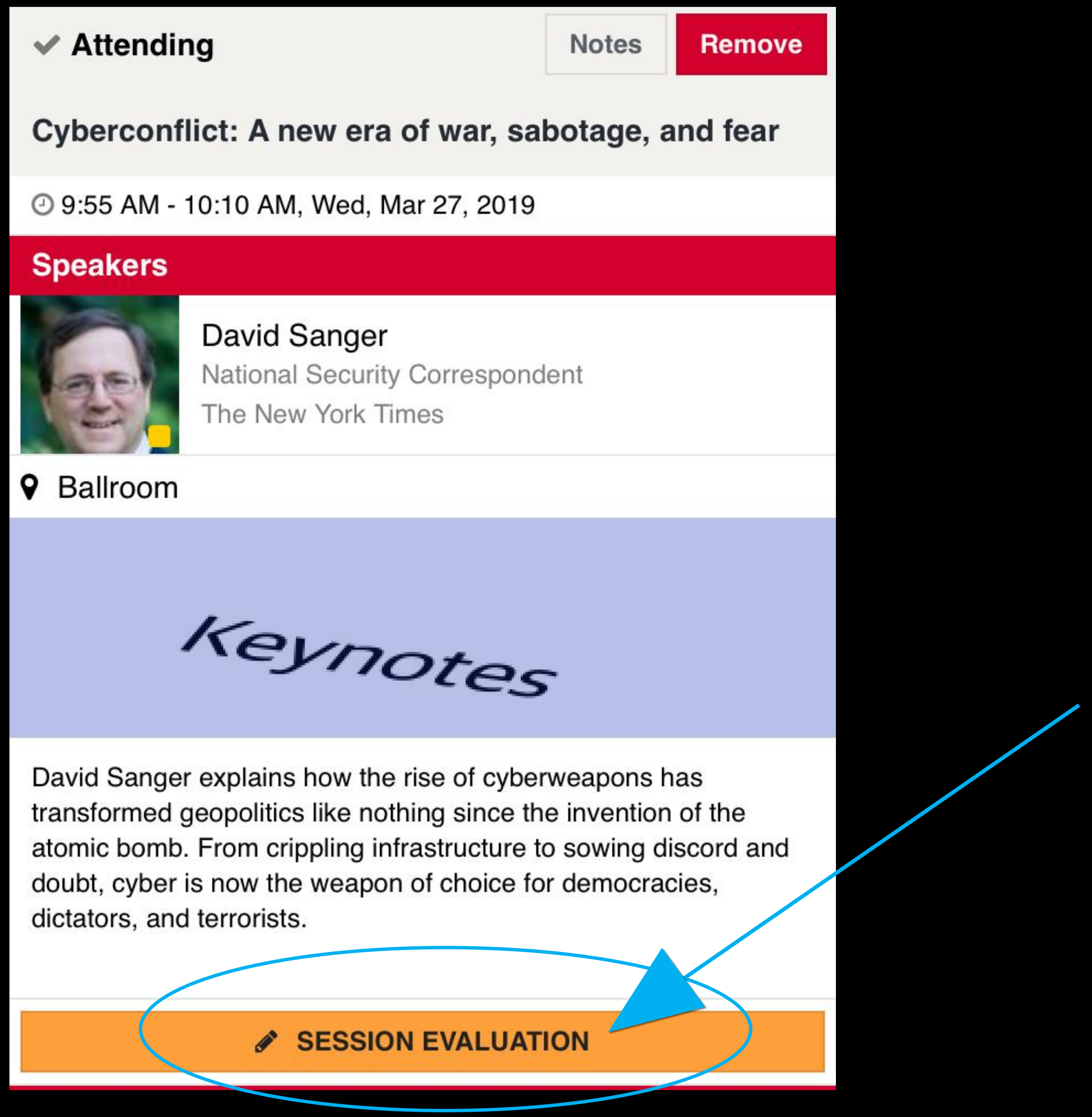
Machine learning

- cloud.google.com/products/ai

Q&A

Rate today's session

Session page on conference website



O'Reilly Events App