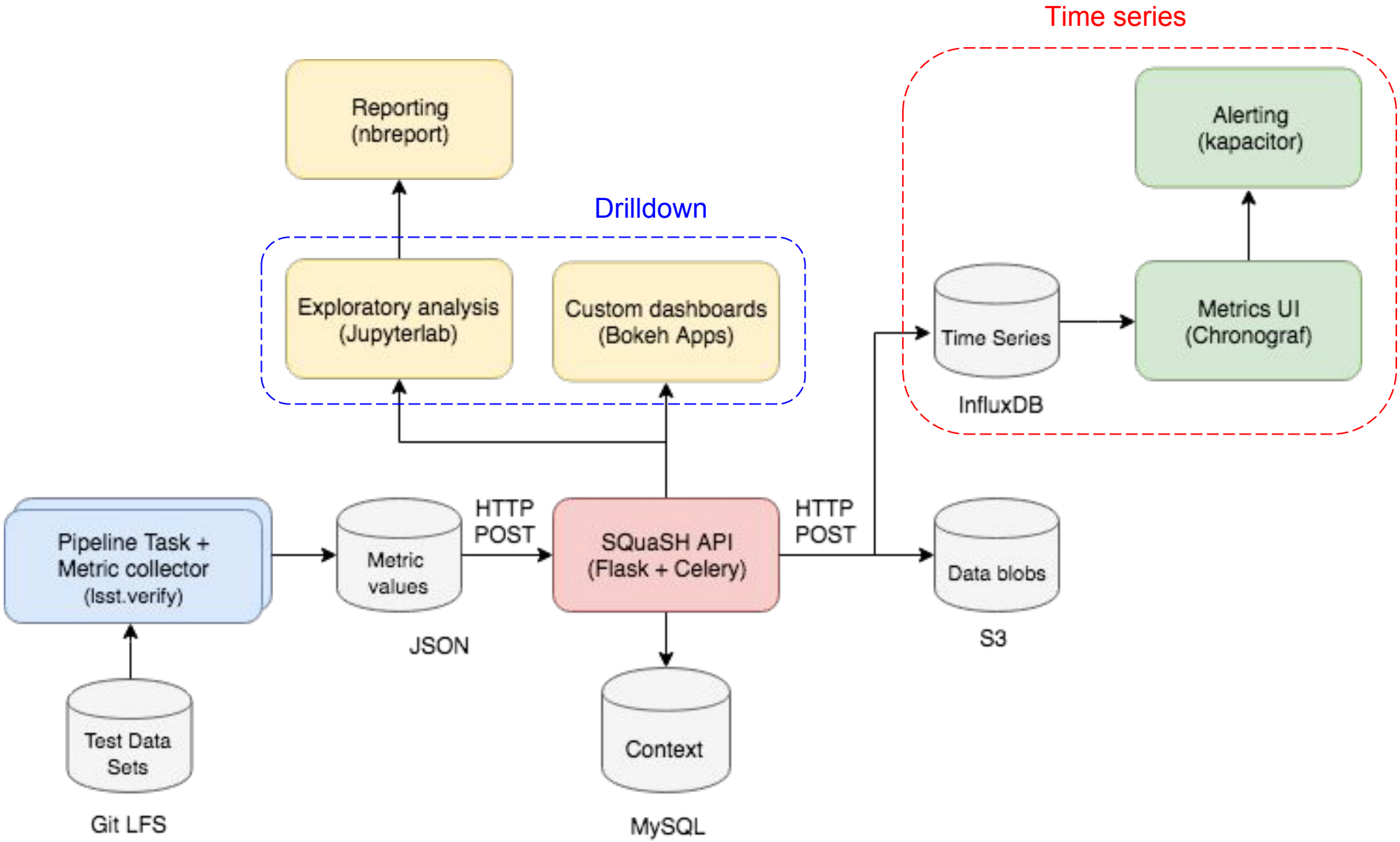
# Adopting InfluxDB + Chronograf + Kapacitor for SQuaSH

SQuaRE team

# (Current) Pipeline metrics system



# Why adopting a Time Series Database?

**The most popular database management systems**

| December 2018 | | Score |
|---|---|---|
| 1. | Oracle | 1283 |
| 2. | MySQL | 1161 |
| 3. | Microsoft SQL Server | 1040 |
| 4. | PostgreSQL | 461 |
| 5. | MongoDB | 379 |

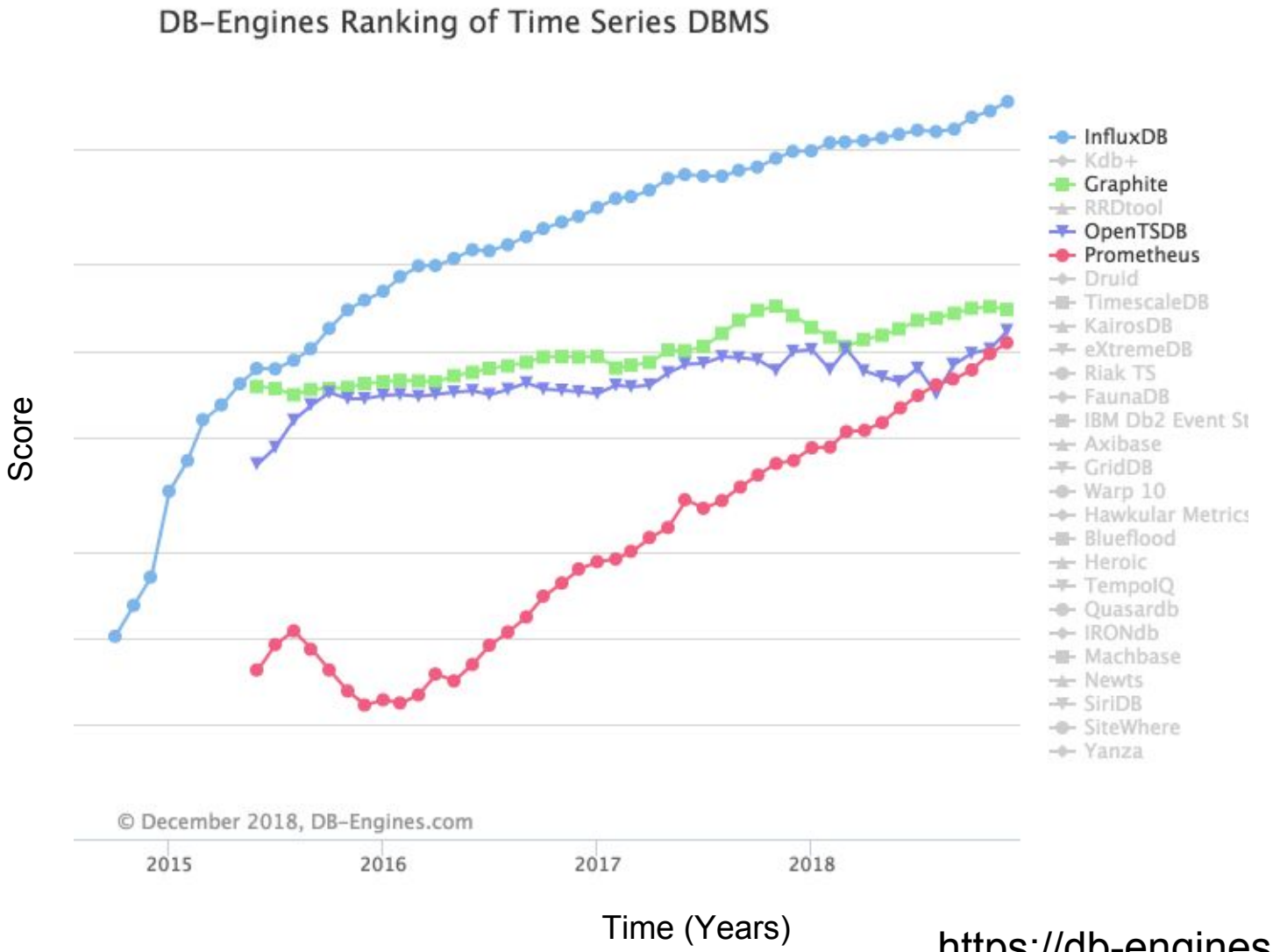
» more

https://db-engines.com/en/ranking

Can I use a relational database instead?

- Yes, but it's painful
- It's not optimized for time series data
- You are going to create yet another solution for a solved problem

# Why InfluxDB?




25+

https://db-engines.com/en/ranking

# Why InfluxDB?

- InfluxDB open-source time series database, has an SQL-like query language and provides an HTTP API
- Chronograf open-source web application for time series visualization written in Go and React.js
- Kapacitor open-source framework written in Go for processing, monitoring, and alerting on time series data.

https://www.influxdata.com

# InfluxDB Concepts

**Measurement**: ap_association

**time** | **tag-set** | **field-set**

| Time | ci_id | ci_dataset | instrument | filter | visit | ccdnum | AssociationTime | numNewDiaObjects | numUnassociatedDiaObjects |
|---|---|---|---|---|---|---|---|---|---|
| 2018-12-07T14:25:59Z | 67 | CI-HiTS2015 | DECAM | g | 411371 | 56 | 0.84 | 116 | 22 |
| 2018-12-07T14:33:21Z | 67 | CI-HiTS2015 | DECAM | g | 411371 | 60 | 0.82 | 68 | 285 |
| 2018-12-07T14:41:00Z | 67 | CI-HiTS2015 | DECAM | g | 411420 | 10 | 0.7 | 131 | 0 |
| 2018-12-07T14:48:32Z | 67 | CI-HiTS2015 | DECAM | g | 411420 | 5 | 0.65 | 175 | 0 |
| 2018-12-07T14:55:55Z | 67 | CI-HiTS2015 | DECAM | g | 419802 | 10 | 1.19 | 267 | 94 |
| 2018-12-07T15:03:16Z | 67 | CI-HiTS2015 | DECAM | g | 419802 | 5 | 1.24 | 315 | 136 |
| 2018-12-08T14:27:42Z | 68 | CI-HiTS2015 | DECAM | g | 411371 | 56 | 0.84 | 116 | 22 |
| 2018-12-08T14:35:09Z | 68 | CI-HiTS2015 | DECAM | g | 411371 | 60 | 0.81 | 68 | 285 |
| 2018-12-08T14:42:32Z | 68 | CI-HiTS2015 | DECAM | g | 411420 | 10 | 0.67 | 131 | 0 |
| 2018-12-08T14:49:53Z | 68 | CI-HiTS2015 | DECAM | g | 411420 | 5 | 0.64 | 175 | 0 |
| 2018-12-08T14:57:22Z | 68 | CI-HiTS2015 | DECAM | g | 419802 | 10 | 1.17 | 267 | 94 |
| 2018-12-08T15:04:58Z | 68 | CI-HiTS2015 | DECAM | g | 419802 | 5 | 1.21 | 315 | 136 |

- A **Measurement** is conceptually similar to an SQL table
- The measurement name describes the data stored in the associated **Fields**
- **Fields** correspond to the actual data, while **Tags** are used to annotate your data (metadata). An important difference is that tags are indexed in InfluxDB while fields are not.

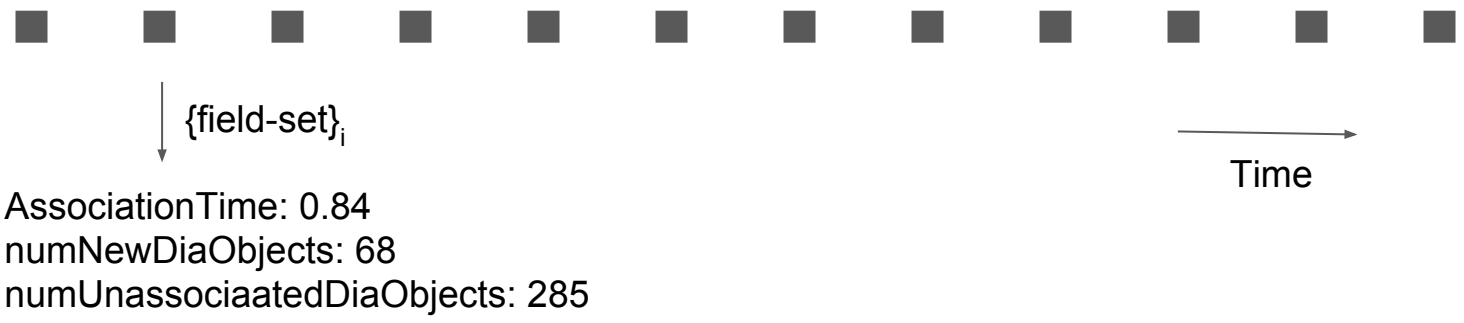
https://docs.influxdata.com/influxdb/v1.7/concepts/key_concepts/

# InfluxDB Concepts

A **Point** contains the field-set of a serie for a given tag-set and timestamp.

A **Series** contains **Points** and is defined by a measurement and tag-set, e.g:

- ap_association,ci_dataset=CI_HiTS2015,filter=g,visit=411371, ccdnum=60




- ap_association,ci_dataset=CI_HiTS2015,filter=g,visit=411371
- ap_association,ci_dataset=CI_HiTS2015,filter=g

InfluxDB creates and stores all the possible series for efficiency!

# InfluxDB Concepts

**Series cardinality** depends essentially on how you design your **tag-set**

```
> show series cardinality
cardinality estimation
----------------------
2319
```

Rule of thumb: have fewer series with more points than more series with fewer points to improve performance.

A **Database** store one or more series.

# Mapping lsst.verify concepts to InfluxDB

- Verification Package ⇨ InfluxDB Measurement
- Verification Job metadata (but not all, e.g. “ci_id”) ⇨ InfluxDB Tag
- Verification Job measurement (metric value) ⇨ InfluxDB Field
- CI pipeline run time
- Verification Job time
- InfluxDB creation time

⇨ InfluxDB Time

**Result**: store verification metrics as time series data in InfluxDB.

# Resources

- InfluxData stack
- InfluxDB Concepts
- InfluxDB Query Language (InfluxQL)
- Other resources