Unified Reporting of Service Reliability

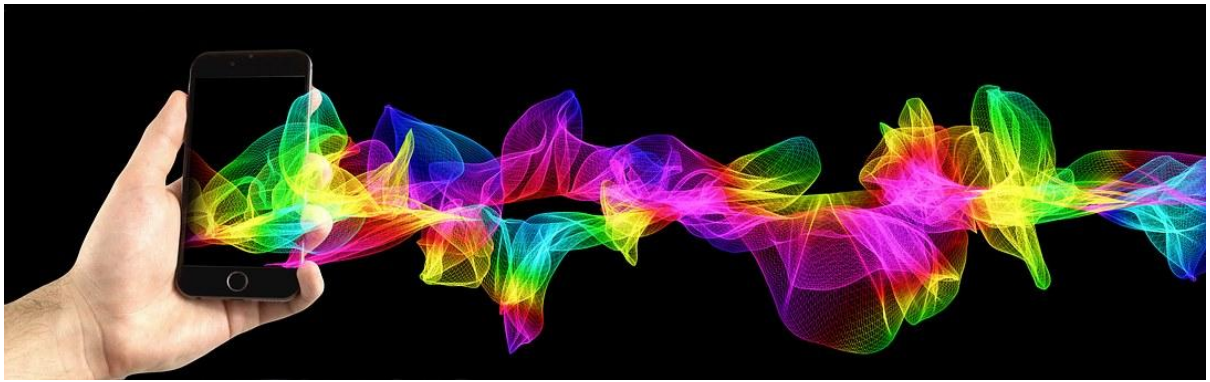
Helen Zhang, Google

SREcon19 Asia/Pacific
2019-06-13

Google Cloud

Image source: pixabay.com

We Have a Dream



Gain actionable insights from a unified view of service reliability

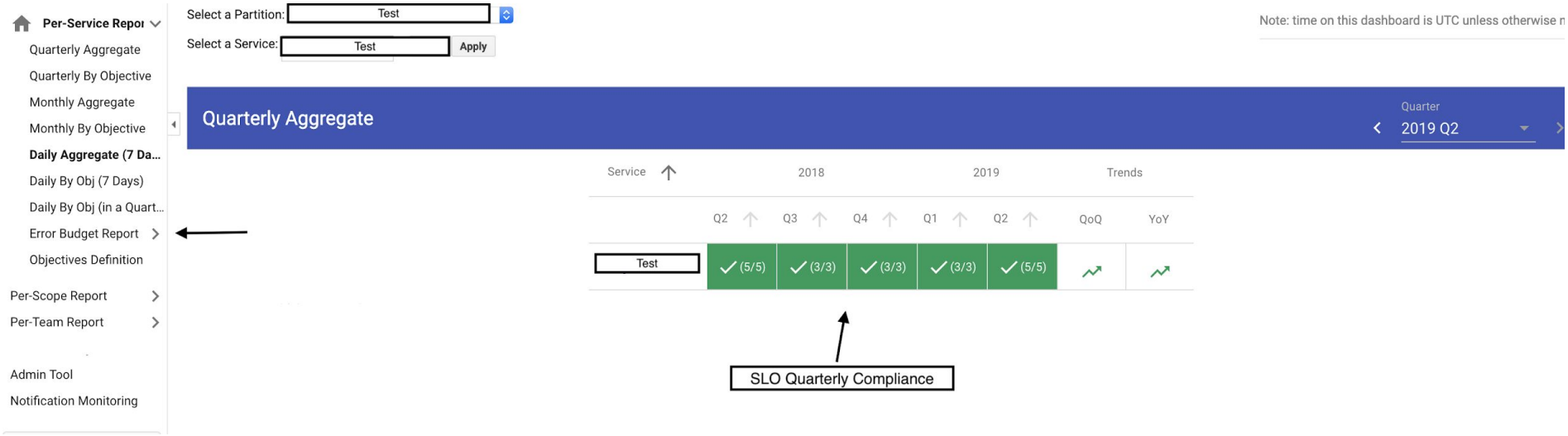
Google Cloud

Image source: pixabay.com

Agenda

- Problem
- Solution
- Success and Challenges
- Takeaways

Problem



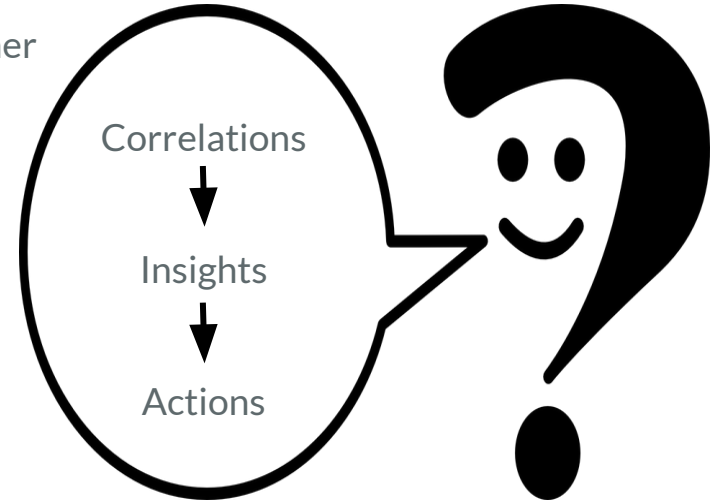
Tools exist to visualize SLO compliance, error budget, but…

Google Cloud

Problem

No “one-stop” tool exists to correlate SLO metrics to other service events to gain actionable insights:

- What **launches** or production rollouts caused a production **outage**, broke **SLO compliance**, and generated a **Cloud support ticket**?
- What **actions** can we take**?**



Google Cloud

Image source: pixabay.com

A Bonus Problem

Can we use ML to predict the probability of a service's SLO violation?

Google Cloud

Image source: pixabay.com

We Have a Plan

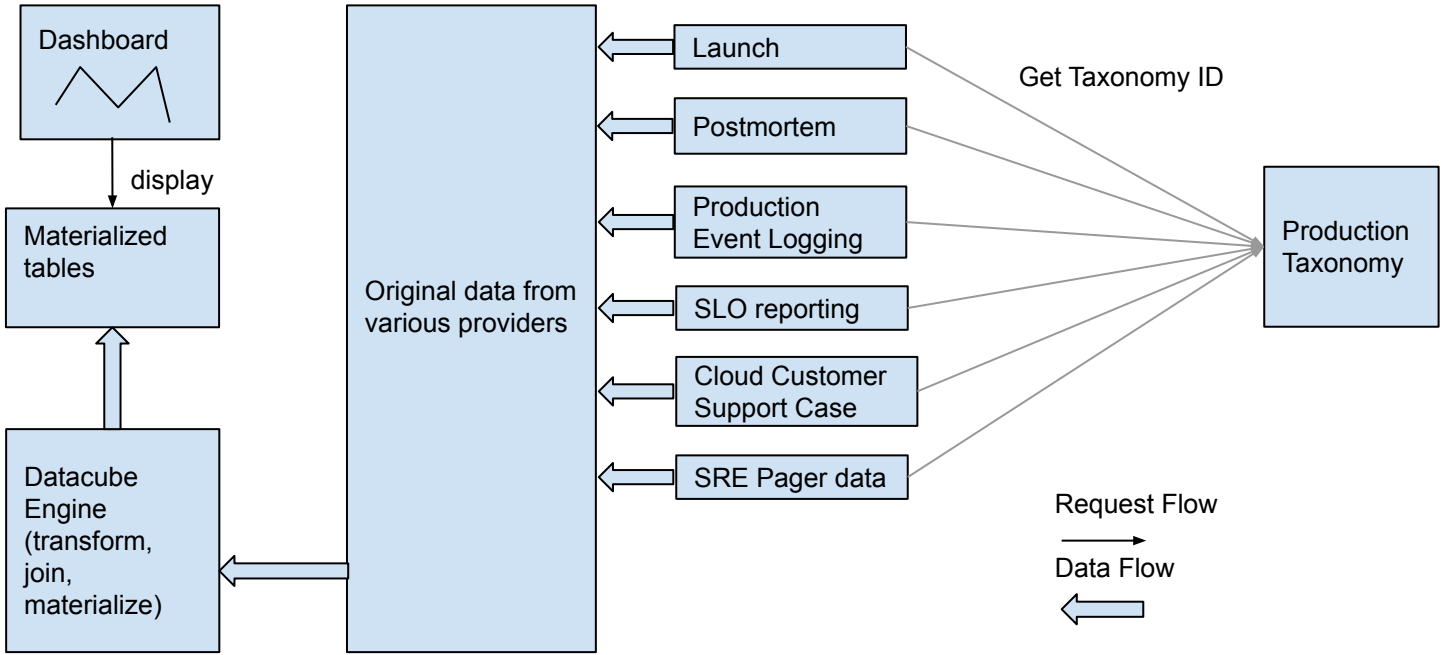
Build a multidimensional “data cube”

- One cube = one entity (service/product/product group)
- Each dimension = one aspect of production data (e.g. SLO compliance, outage count, SRE pager load)
- See the correlated data for one entity? Query one cube!

Google Cloud

Image source: pixabay.com

Unified Reporting Architecture: 10,000 foot view



Google Cloud

ML - Only a Start

- Used ML to predict SLO violations
  - Initial explorations didn't go far
- Challenges
  - Predicting rare events is hard
  - Limited data quantity and quality.
    i.e. need **more high quality** data
- Not actively working on it, but would like to pursue it further in the future

Google Cloud

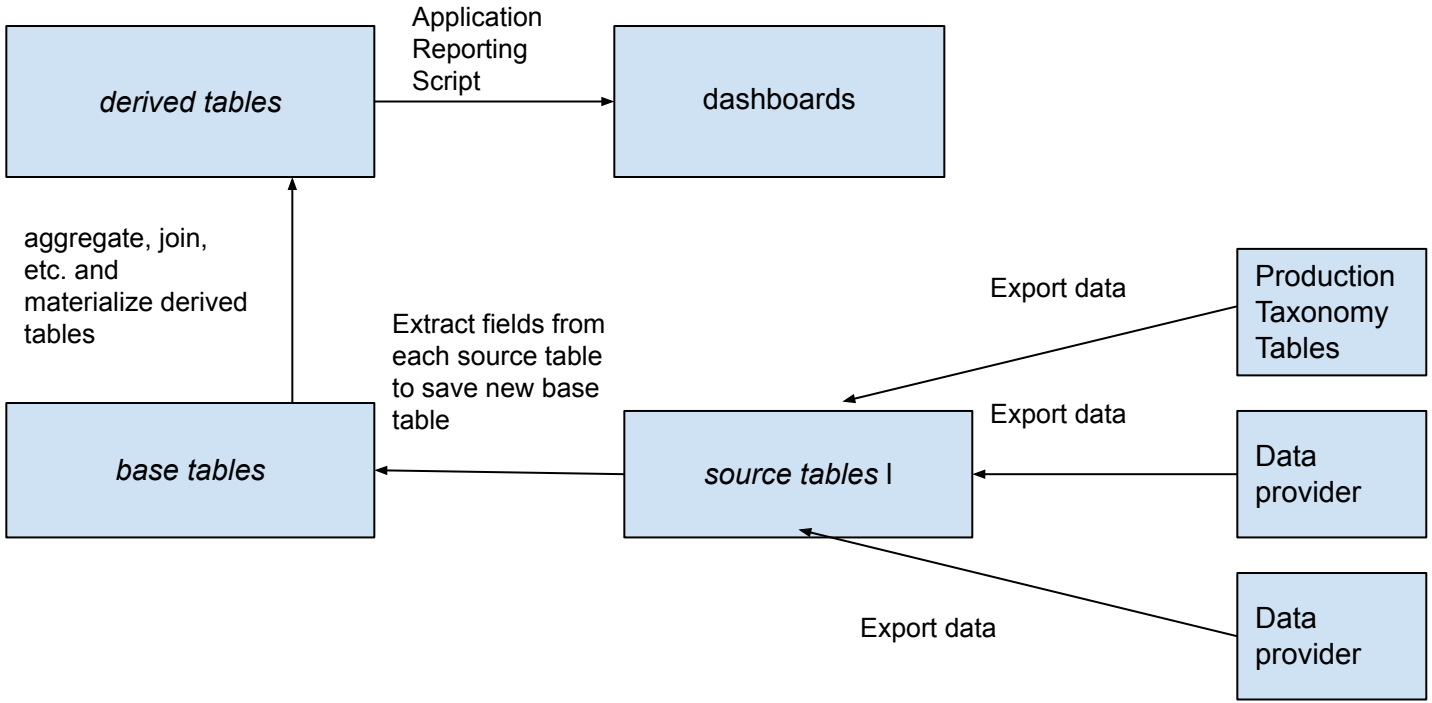
Image source: pixabay.com

Unified Reporting Design Overview

- Step 1: Production Taxonomy
  - A Unique ID for different entities: product, project, service, etc.
  - A different team did this work

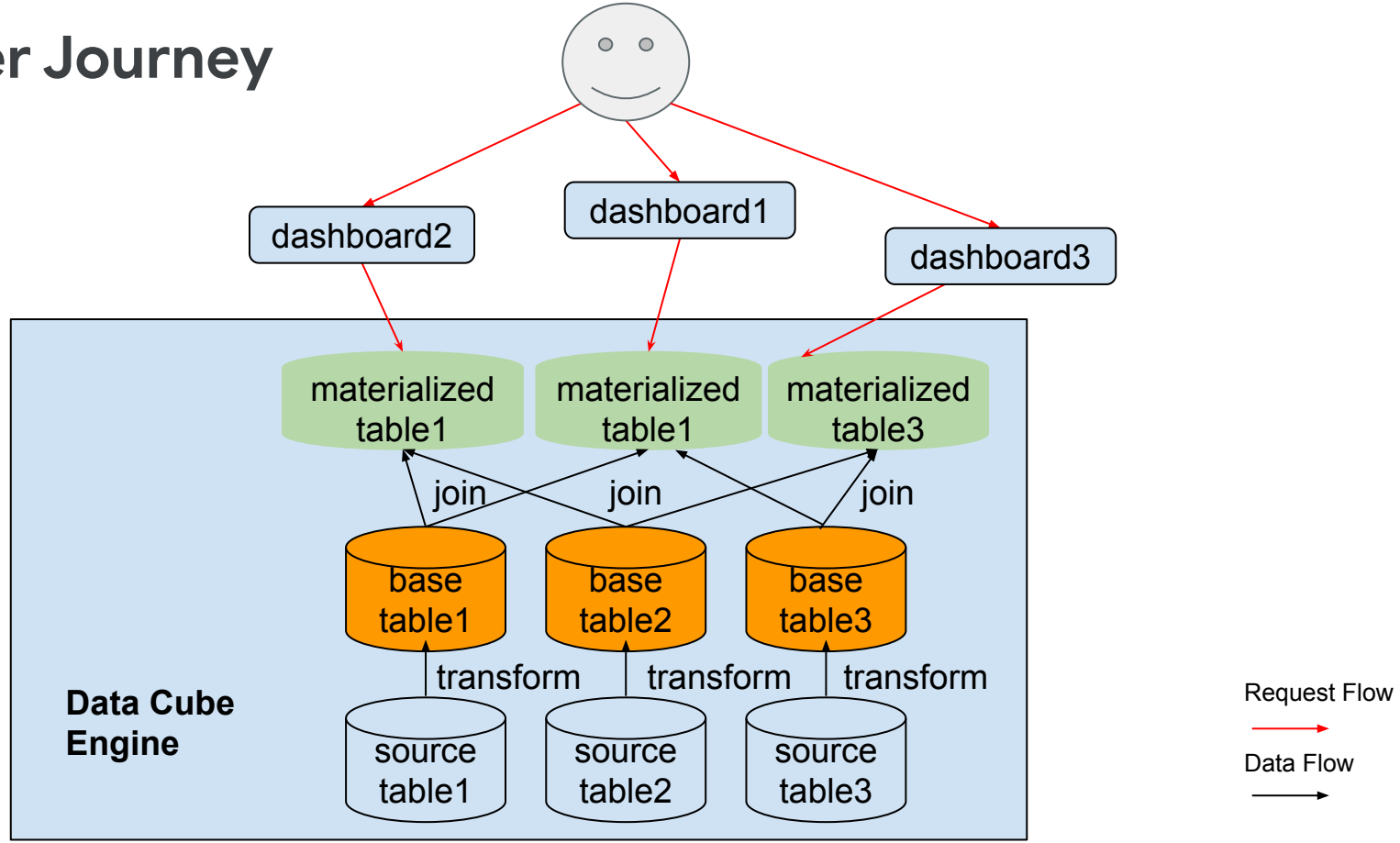
- Step 2: Data Cube
  - Ingest and join different data sources using Production Taxonomy ID
  - I and my team worked on this part

Life of a Dataset



Google Cloud

User Journey



Design Principles

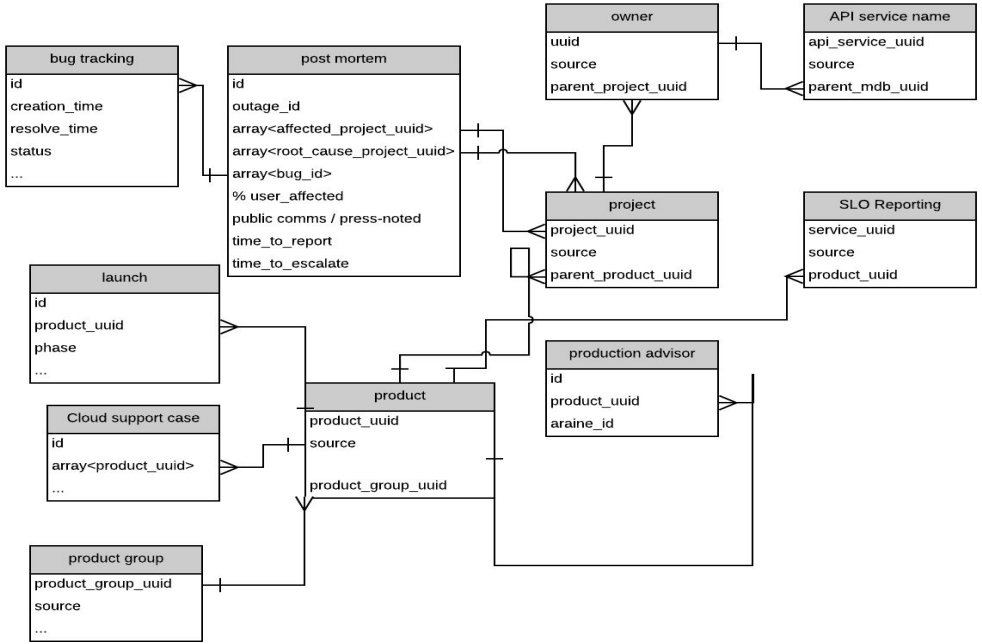
- Use the simplest infrastructure
- Focus on data

Data Modeling

- Entity Relationship Database
- Star Schema

Entity Relationship Database Model

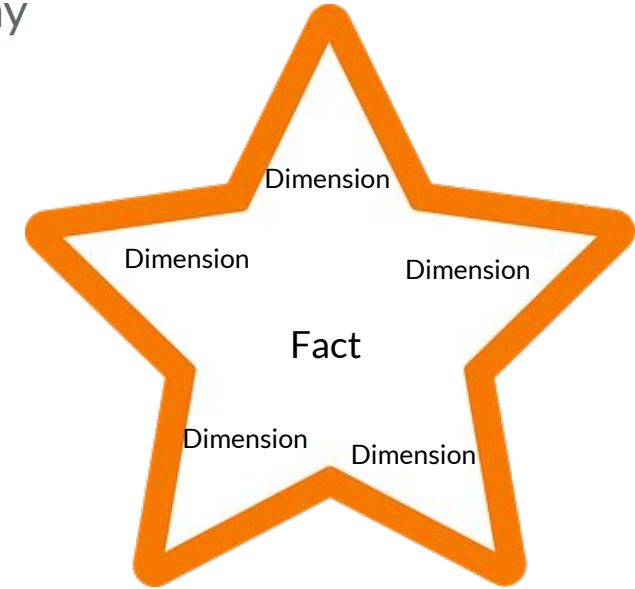
- Model Product Area, Product Group, Product, Project, Owner, API Service name entities
- Model the following relationships among all the entities:
  - Service API ↔ group [n:1]
  - mdb ↔ project [n:1]
  - project ↔ product [n:1]
  - product ↔ product group [n:1]
  - product ↔ product area [n:1]



Google Cloud

Star Schema Model

- Most widely used for data warehouses
- Consists of one or more fact tables referencing any number of dimension tables.



Google Cloud

ERD Model is the Best Option

A natural fit for the existing schema of all data sources

Star Schema doesn't work well for M:M relationships, common in our use cases, e.g.

- 1 outage is associated to SLO violations of multiple services
- 1 service's SLO violation can cause multiple outages

Insights Needed

- Are my service's SLIs/SLOs aligned with customer happiness?

- How often do customers report outages before our monitoring/alerting system detects them?

Insight and Action: Fix ill-defined SLI/SLO

		Correlation		Insight	Action
Service	Aggregation Period	SLO Compliance Met?	Major Outage Happened?	SLO reflect User Happiness?	
A	Quarterly	Yes	No	Yes	Nothing
B	Quarterly	No	Yes	Yes	SLI/SLO is good; Fix the service
C	Quarterly	Yes	Yes	No	Fix SLI/SLO; Fix the service
D	Quarterly	No	No	No	Fix SLI/SLO

Limitations

- Impact is limited due to outstanding data quality issues

- A cross-team technical program (not run by our team) is created to drive making service SLIs/SLOs reflect customer experience

Insight and Action: Fix monitoring/alerting gaps

Production Outage	**Correlation** Customers detect sooner than Google? →	**Insight** Gaps in monitoring/alerting? →	**Action**
Outage 1	No	No	Nothing
Outage 2	Yes	Yes	Fix Monitoring/Alerting
Outage 3	Yes	Yes	Fix Monitoring/Alerting

Challenges

Outstanding quality issues unresolved

- Limited quantity, incomplete, and inaccurate source data
- Correlation inaccuracy due to the lack of a common identifiers across data sources

Google Cloud

Image source: pixabay.com

Takeaways

- Establish a solid process to enforce clean data from the source
- Focus
  - Standardize and automate
  - Have a vision for the future, but don’t be disappointed if the first attempt doesn’t succeed