Data (as a) Product

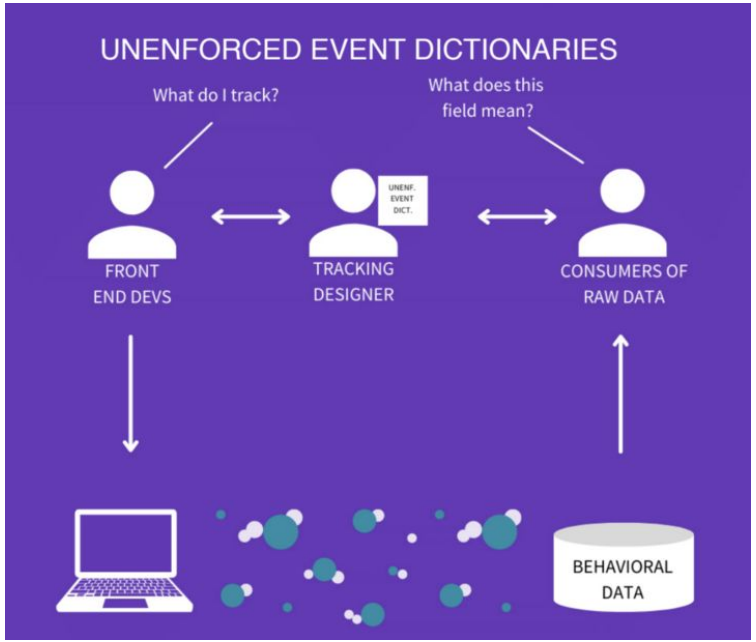
Scaling self-serve analytics with data contracts

Jon Su

Show of hands

Who trusts their digital analytics data?

The Status Quo for Event Tracking Design



- Design intent communicated to devs and consumers by ***"someone"***
- Unclear ownership
- Data treated as a 2nd class citizen - created as a by-product with no **intention after the fact**

Unenforced event dictionaries are at the heart of measurement

And every company wants to self-serve

- Scale beyond a centralised data team
- Develop their own insights and build impactful data apps
- Improve their decision-making and time to decision

And every company wants to self-serve

- Scale beyond a centralised data team
- Develop their own insights and build impactful data apps
- Improve their decision-making and time to decision

- **To self-serve however, downstream users need to have these answers:**
  - Know what data is available
  - How to access the data
  - Understand the semantics of the data
  - Require certain levels of data quality
  - Know how reliable and on-time the data can be delivered

And every company wants to self-serve

- Scale beyond a centralised data team
- Develop their own insights and build impactful data apps
- Improve their decision-making and time to decision

- **To self-serve however, downstream users need to have these answers:**
  - Know what data is available
  - How to access the data
  - Understand the semantics of the data
  - Require certain levels of data quality
  - Know how reliable and on-time the data can be delivered
- **Most importantly someone needs to own the data (not just the data team)**

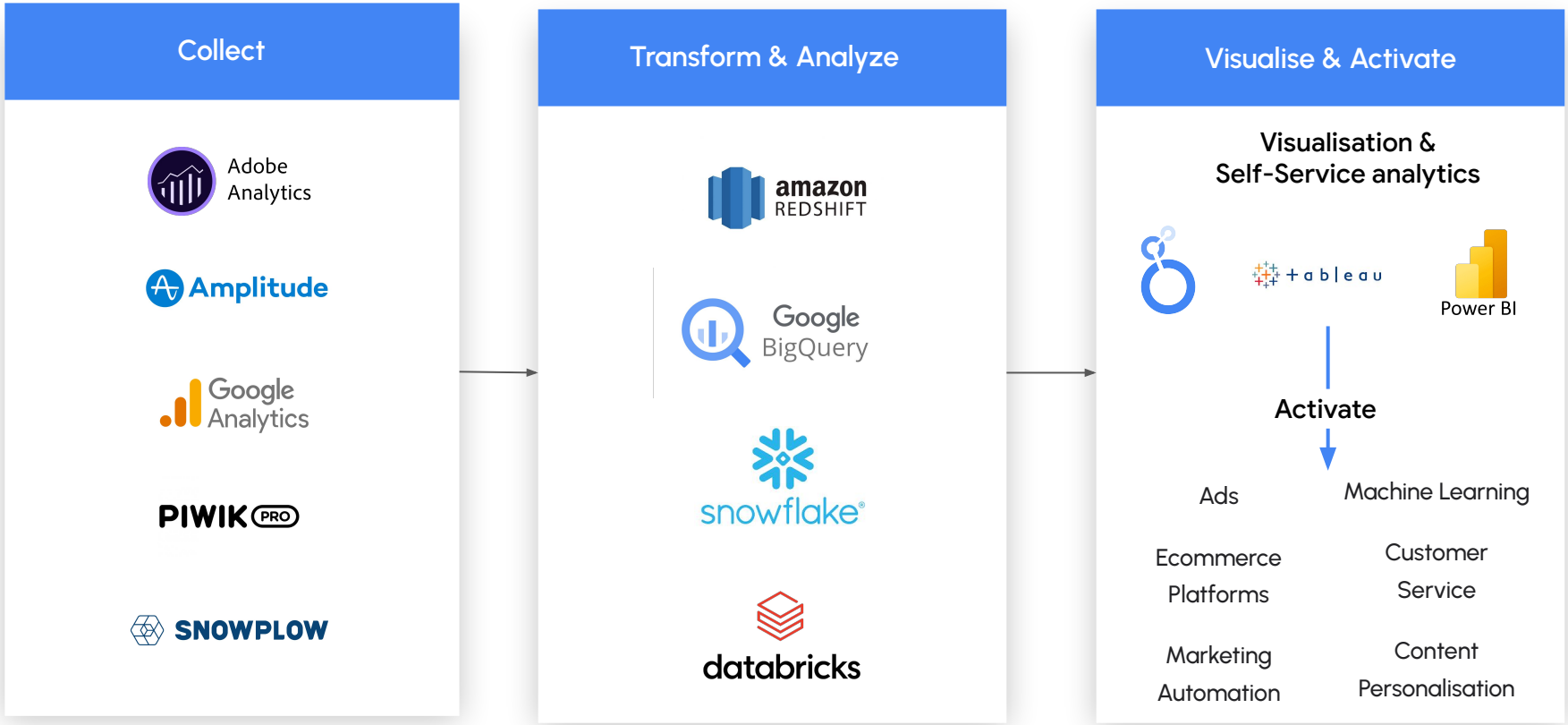
But there
are challenges…

Common data challenges

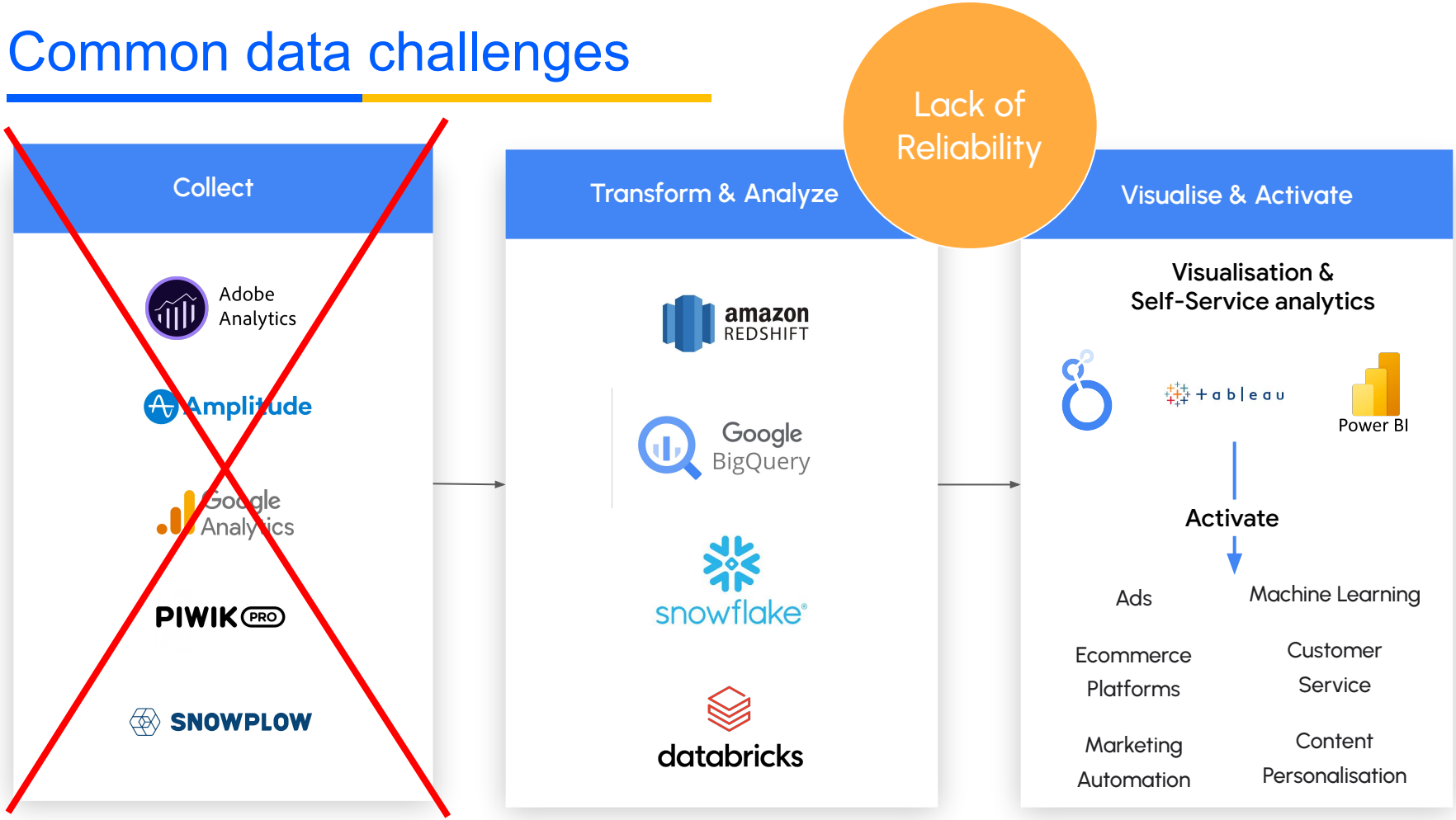
- Data can change anytime, without warning
- No team collaboration
- No documentation about the data
- Using the data requires
  in-depth knowledge of the implementation

Reference: Andrew Jones
https://docs.google.com/presentation/d/1CXoTBlJzUZtYMg7biVupu_d5ClvKR8e2w3xt_zNoqAE/edit#slide=id.g1ca3ea548cb_0_108

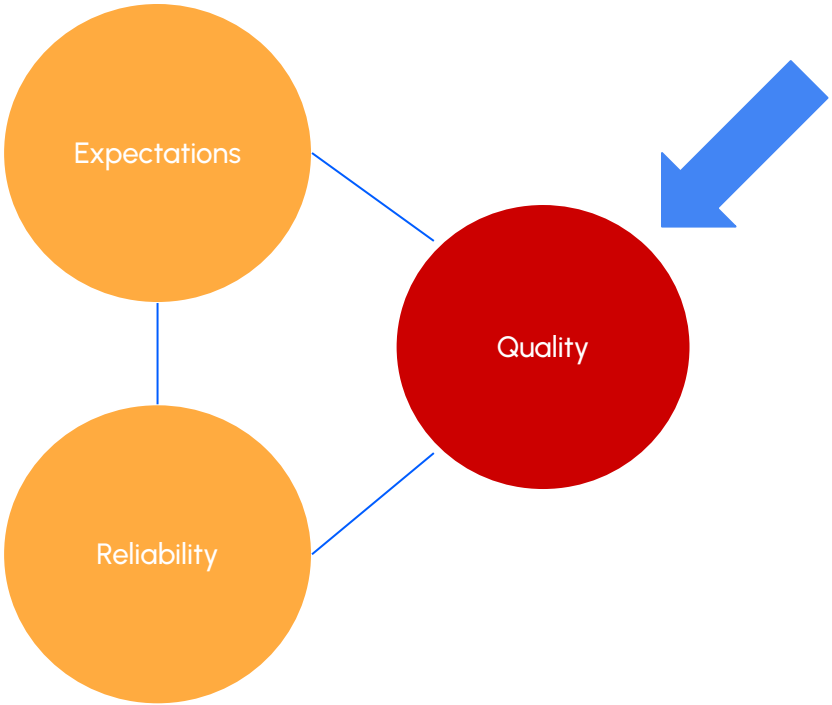
Common data challenges



Common data challenges

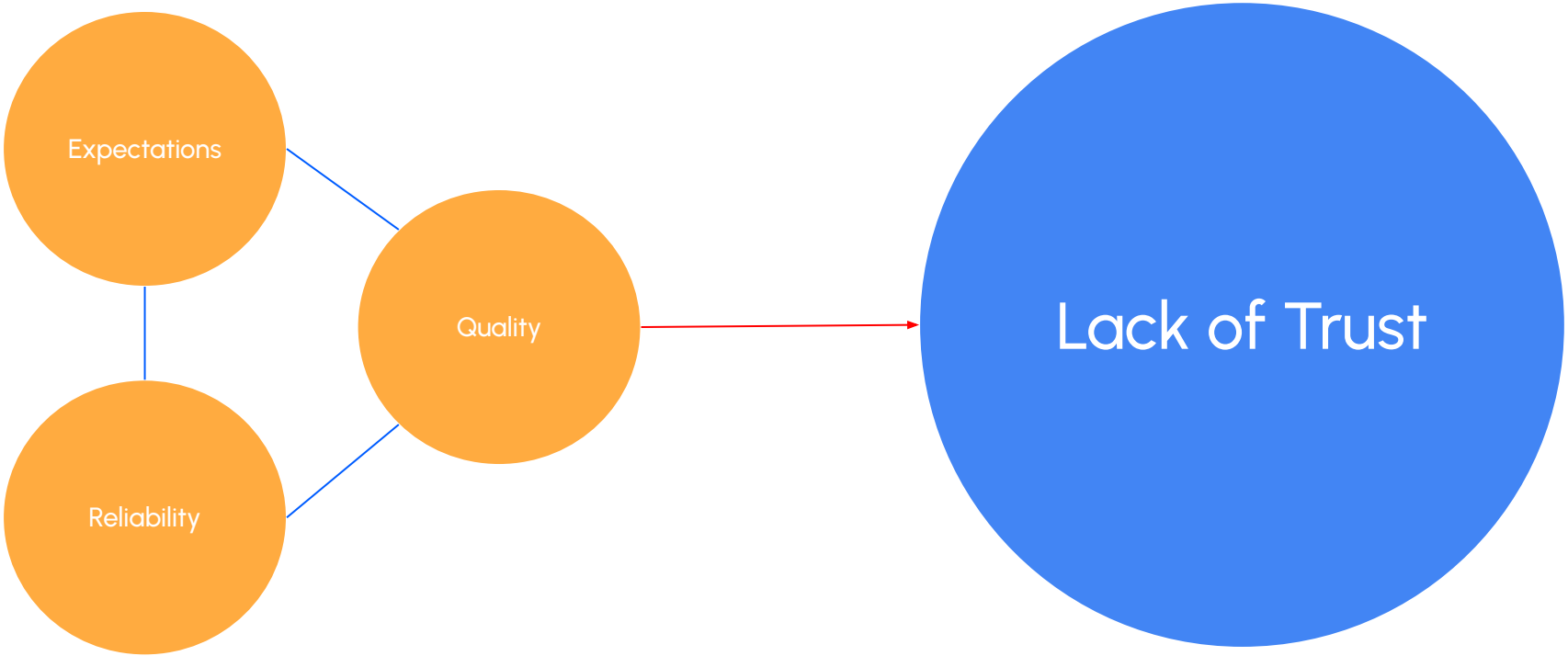


Common data challenges



Reference: Andrew Jones
https://docs.google.com/presentation/d/1CXoTBlJzUZtYMg7biVupu_d5ClvKR8e2w3xt_zNoqAE/edit#slide=id.g1ca3ea548cb_0_108

Common data challenges



Reference: Andrew Jones
https://docs.google.com/presentation/d/1CXoTBlJzUZtYMg7biVupu_d5ClvKR8e2w3xt_zNoqAE/edit#slide=id.g1ca3ea548cb_0_108

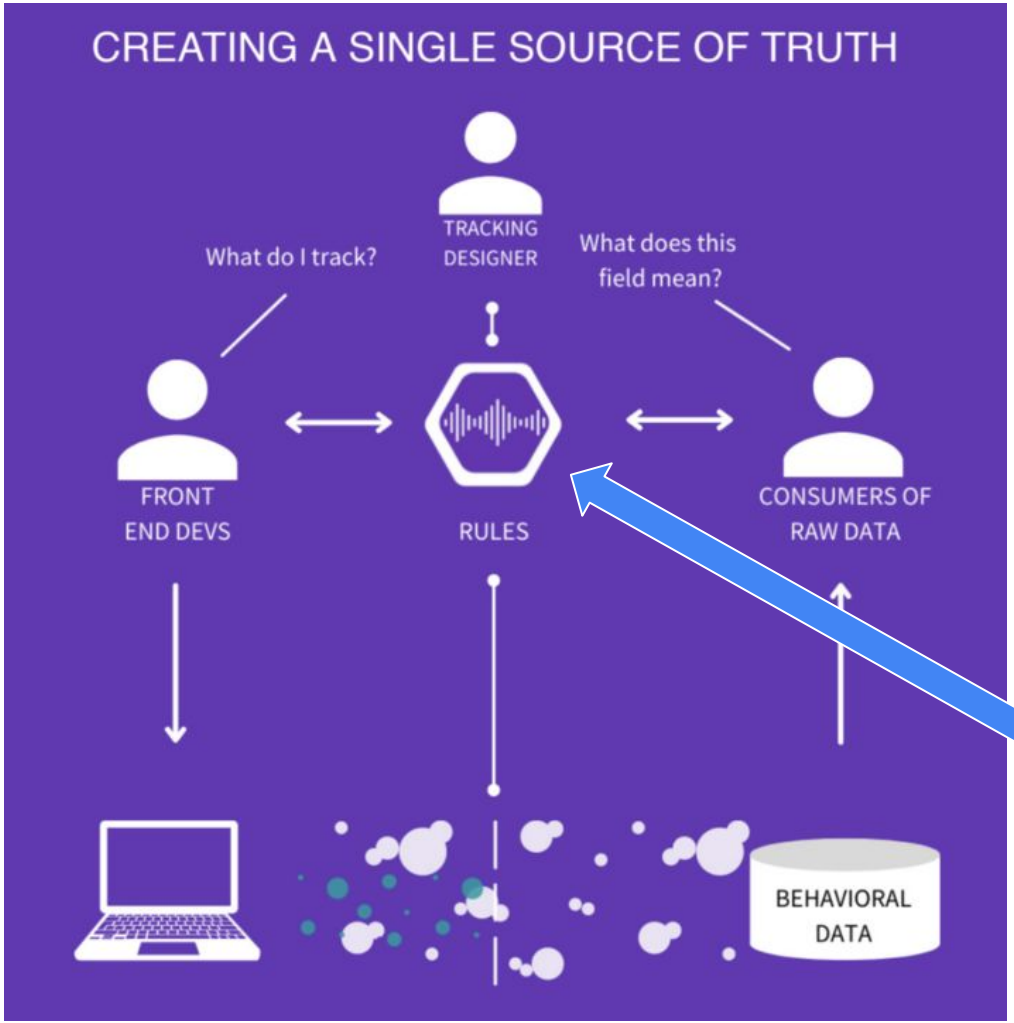
So how do we solve these challenges?

A data (as a) product is a ready-to-use data asset that is intentionally created, managed, maintained and delivered for consumption by authorised data consumers, in serving a particular purpose.

A data (as a) product is a ready-to-use data asset that is intentionally created, managed, maintained and delivered for consumption by authorised data consumers, in serving a particular purpose.

Includes:

- The data itself
- Metadata required to enable to understand the data and self-serve
- Clear ownership
- **And importantly…**

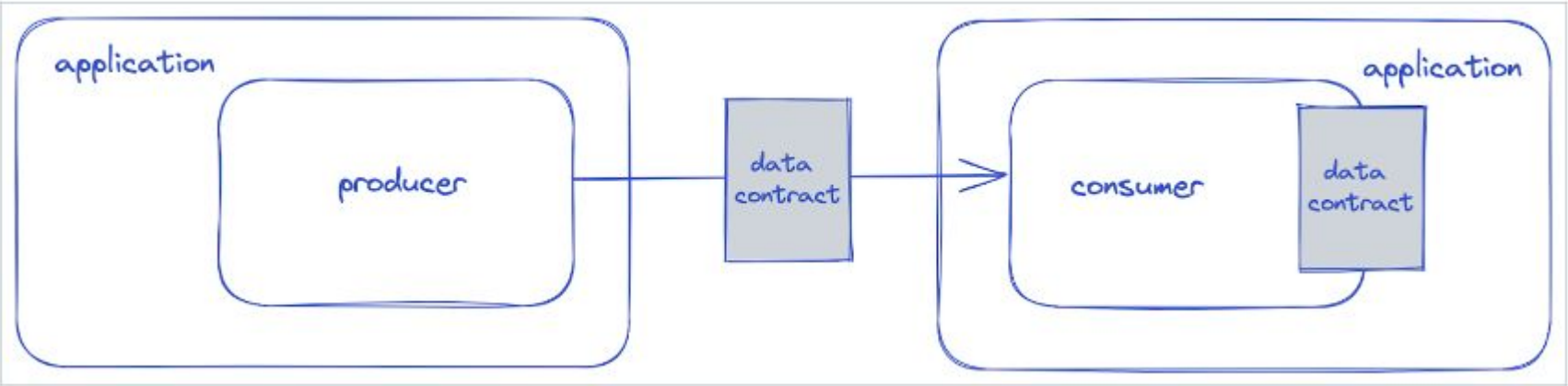


An enforceable set of rules!

A data contract is a formal agreement or set of predefined rules that defines the structure, format and requirements for how data is collected or being exchanged.

How does it work?

- The upstream component implements and enforces a data contract when implementing tracking.
- The downstream component can then assume and **trust** that the data it receives will always conform to the contract.



Reference: https://docs.confluent.io/platform/current/schema-registry/fundamentals/data-contracts.html

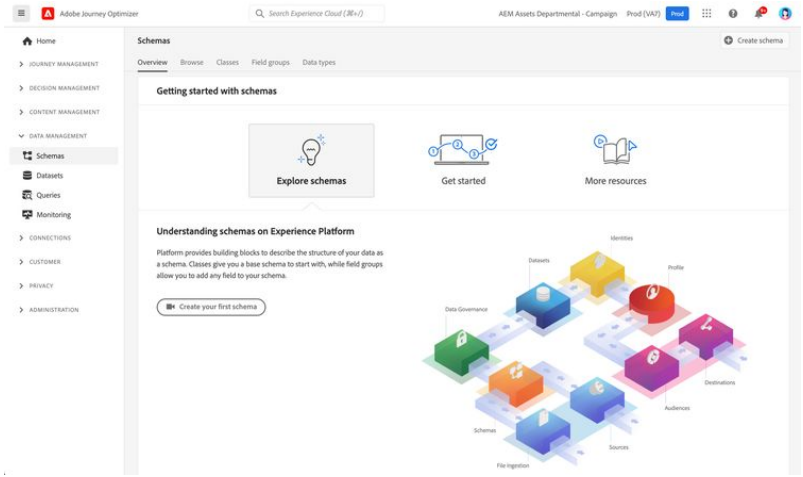
Data contracts are complex and can evolve..

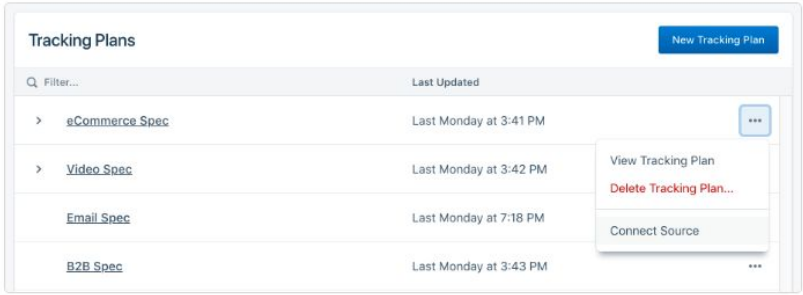
The schema is only one element of a data contract. Data contracts also supports the following:

- **Structure**: fields and their types
- **Integrity constraints**: validation rules (i.e muse be an integer, greater than a value)
- **Metadata**: additional information (i.e sensitive fields) and ownership
- **Rules or policies**: any rule(s) on how to treat a field (i.e hashing) for Enhanced Privacy Compliance, where data can be used
- **SLAs**: Accuracy, completeness and latency
- **Observability**: monitoring and alerts
- **Change or evolution**: how new changes can be accomodated

Reference: https://docs.confluent.io/platform/current/schema-registry/fundamentals/data-contracts.html

Data contracts for digital analytics





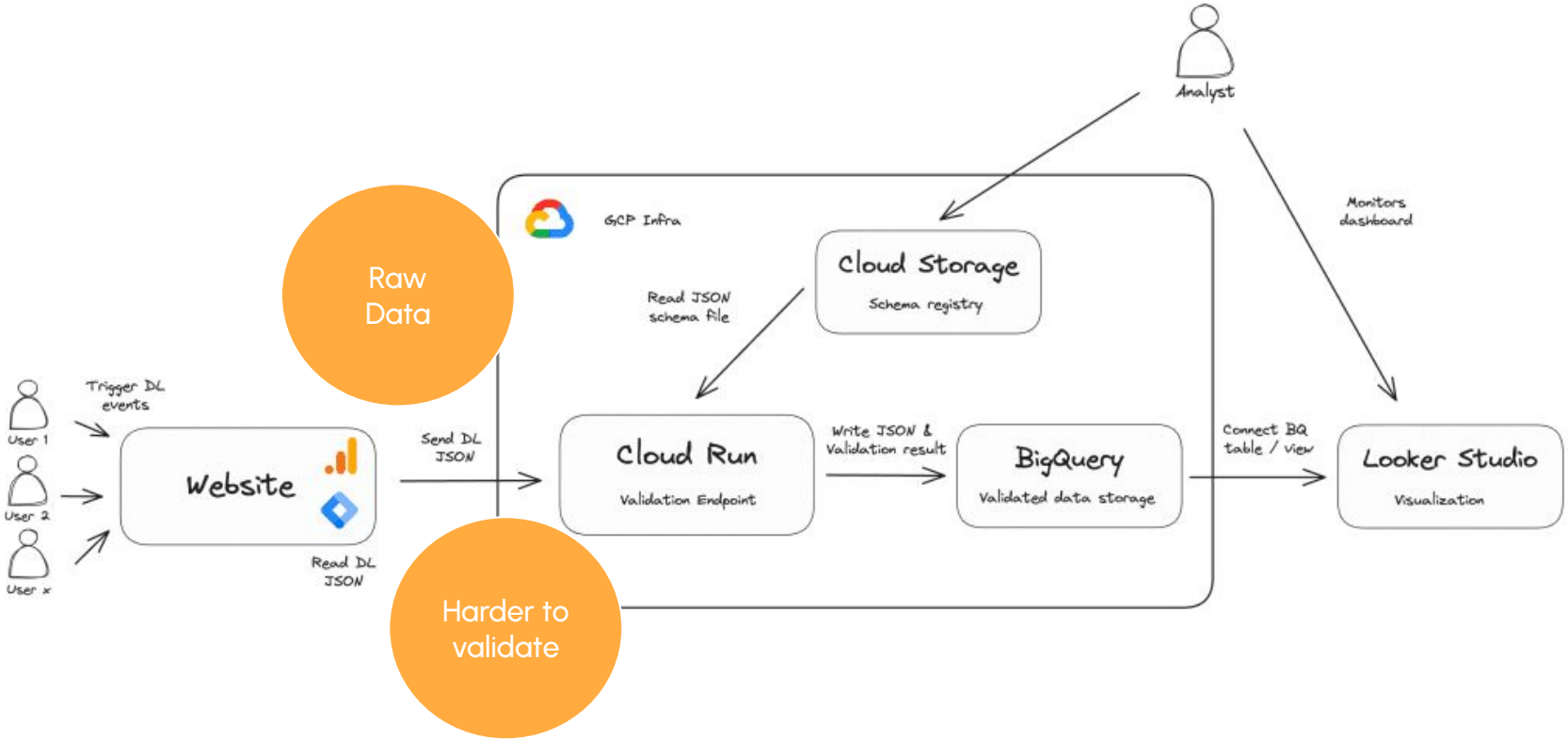

```
{
  "$schema": "http://iglucentral.com/schemas/com.snowplowanalytics.self-desc/schema/jsonschema/
  "description": "Schema for an example event",
  "self": {
    "vendor": "com.snowplowanalytics",
    "name": "example_event",
    "format": "jsonschema",
    "version": "1-0-0"
  },
  "type": "object",
  "properties": {
    "name": {
      "type": "string",
      "maxLength": 128
    },
    "job_role": {
      "description": "",
      "type": [
        "string",
        "null"
      ],
      "maxLength": 128
    },
    "promo_code": {
      "description": "",
      "type": [
        "string",
        "null"
      ],
      "minLength": 8,
      "maxLength": 20
    }
  },
```

What about Google Analytics?

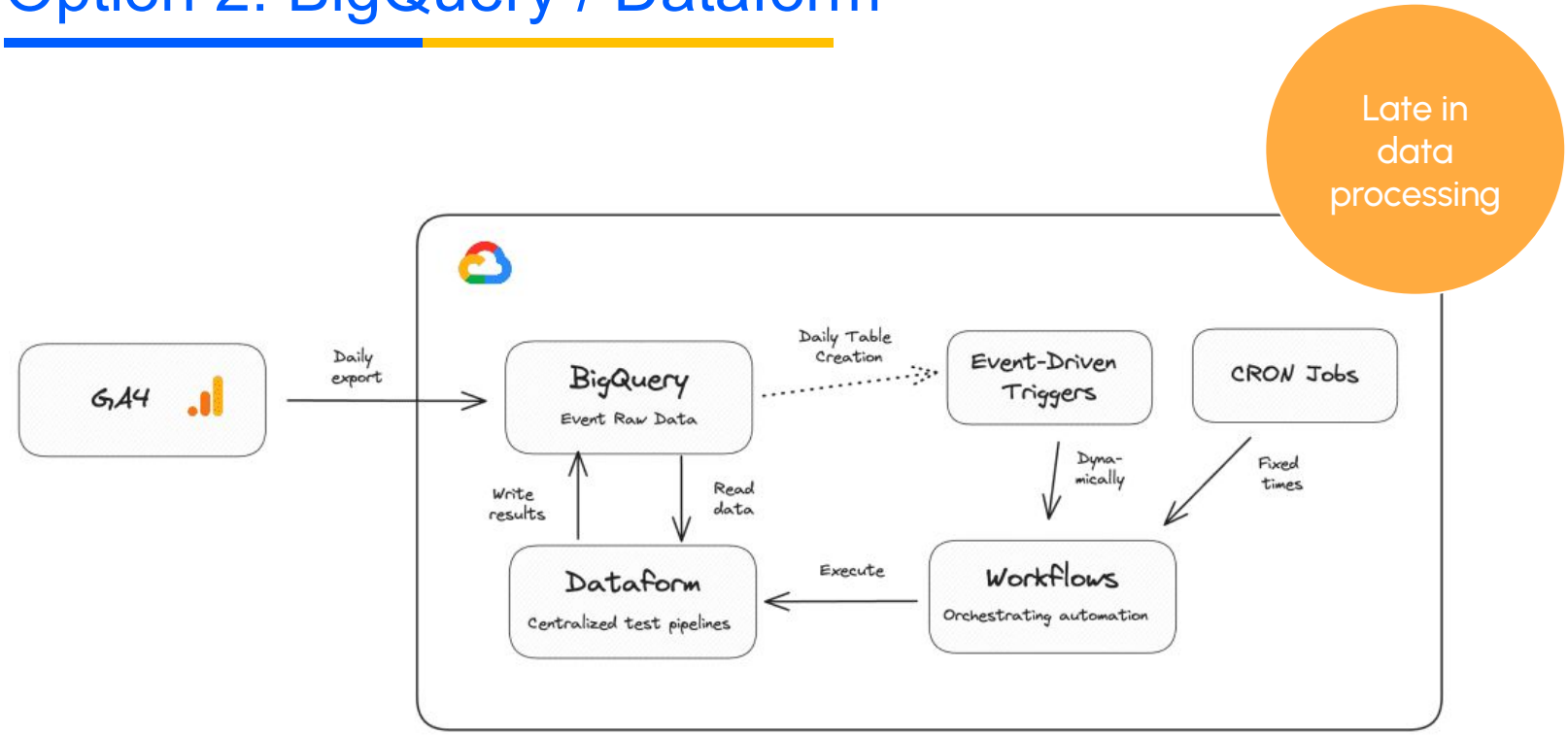
Go check this guy out!

Gunnar Griese
https://gunnargriese.medium.com/ensuring-data-quality-for-ga4-at-scale-with-google-cloud-platform-358c0d015e5c

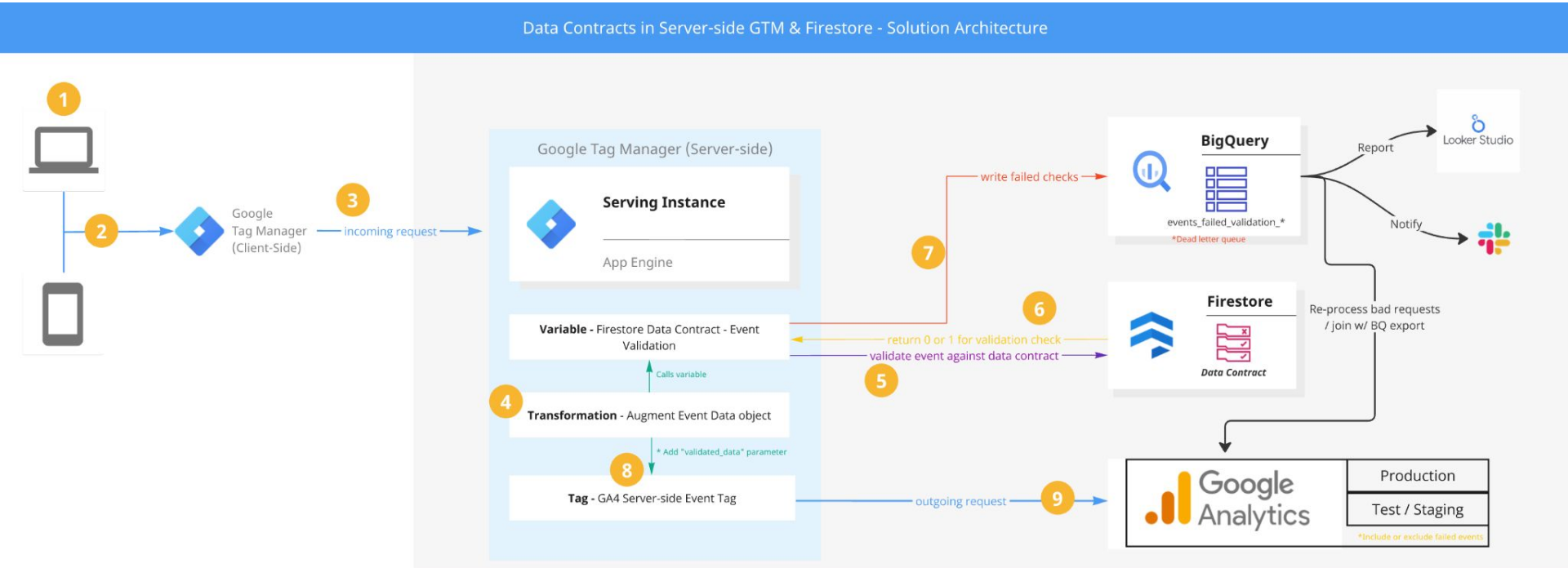
Option 1: DataLayer Schema Validation



Option 2: BigQuery / Dataform



Option 3: GTM-SS Event Validation



Demo

https://github.com/jkersu/ga4-data-contract

README | Apache-2.0 license

GA4 Data Contract Event Validation using GTM Server-Side

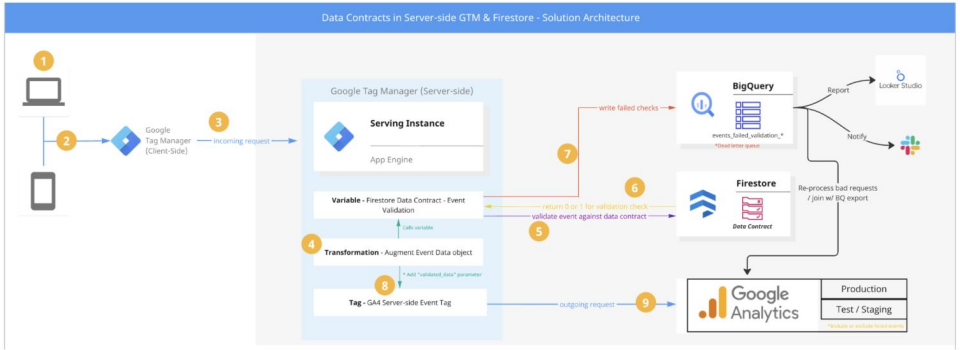
You can use this variable template for Google Tag Manager Server Side container to validate a Google Analytics 4 request against a Firestore document which contains a JSON schema of validation rules for a given event.

Repository Structure

- **datalayer-test:** example datalayer pushes for certain GA4 events that either pass or fails validation against the example schemas in "schemas"
- **gtm-templates:** contains the raw Javascript code for the GTM variable template (template.js). "template.tpl" can be imported directly into a GTM Server Side container
- **images:** contains images for this repo
- **schemas:** some example schemas for certain GA4 events and GA4 ecommerce item (product) to be used as reference and starting point
- **app.py:** a basic Python script that imports JSON file(s) from a Google Cloud Storage bucket into a Firestore collection
- **requirements.txt:** dependencies required to run "app.py"

How it works

The image below outlines in high-level how this GTM-SS variable template works.



One of the biggest benefits…

Encourages cross-functional teams to sit down, collaborate together and critically think about the business value for use cases & by extension define measurement

Encourages cross-functional teams to sit down, collaborate together and critically think about the business value for use cases & by extension define measurement

Encourages cross-functional teams to sit down, collaborate together and critically think about the business value for use cases & by extension define measurement

Encourages cross-functional teams to sit down, collaborate together and critically think about the business value for use cases & by extension define measurement

Encourages cross-functional teams to sit down, collaborate together and critically think about the business value for use cases & by extension define measurement

Any Q's?

Jon Su
https://jonsu.me