



# Building Reusable and Trustworthy pipelines

# Outline

1. Context
2. Design Requirements
3. Proposed Solution
4. Example Code

# Context

```
33
34     self.logduplicates = True
35     self.debug = debug
36     self.logger = logging.getLogger(__name__)
37     if path:
38         self.file = open(st.path, 'w+')
39         self.file.seek(0)
40         self.fingerprints.update(fingerprint)
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getboolean('LOGGING', 'debug')
45         return cls(settings['objdir'], debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         fingerprint = self.request_fingerprint(request)
```

# Hello 🙌!

- ▶ Data engineer @ SnapTravel
- ▶ Data infrastructure, Data engineering, Analytics engineering
- ▶  +  +  +  stack

# Purpose

Share  BI pipelines

 Community with lessons learnt

 feedback

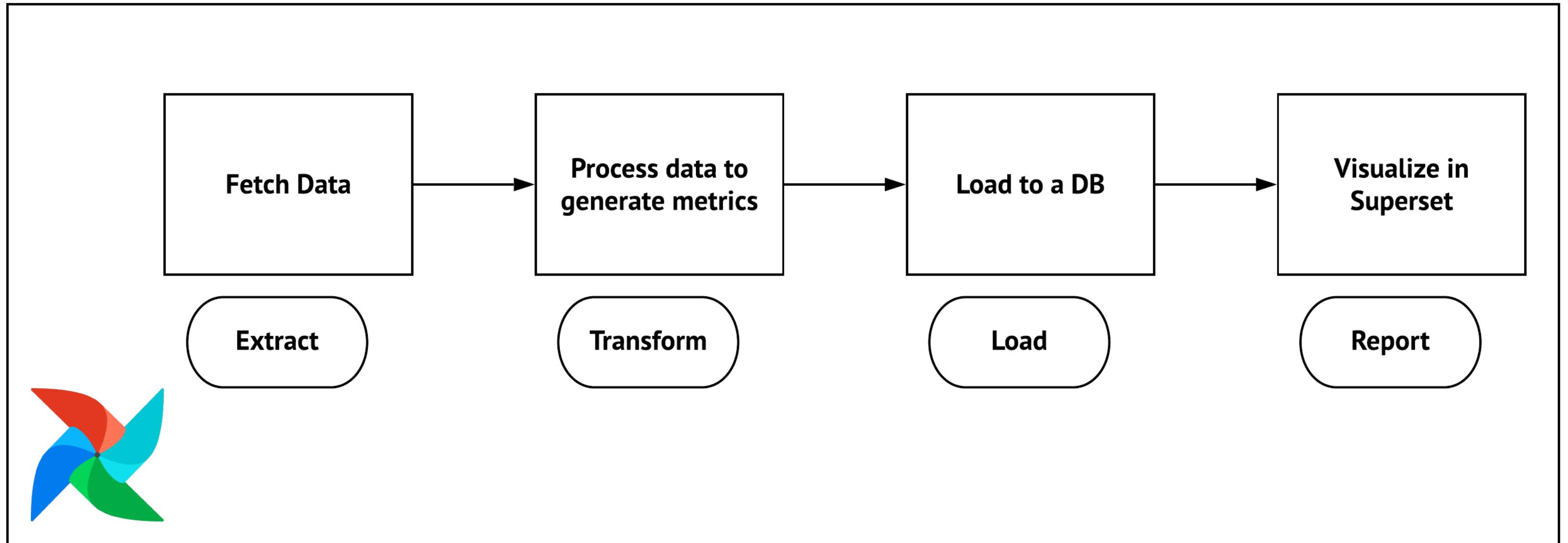
# How are my company ?

- ▶ gross\_revenue
- ▶ contribution\_margin
- ▶ number\_of\_active\_users
- ▶ retention\_rate
- ▶ conversion\_rate

# Hows my airflow repo ?



- ▶ number\_prs\_merged
- ▶ number\_prs\_closed\_without\_merge
- ▶ number\_prs\_opened
- ▶ number\_of\_commit



# Let us consider

- ▶ The pipeline failed after a few days of productionization
- ▶ Now I want to focus on issues
- ▶ Gitlab released a new version of API
- ▶ I want to analyze other apache projects too

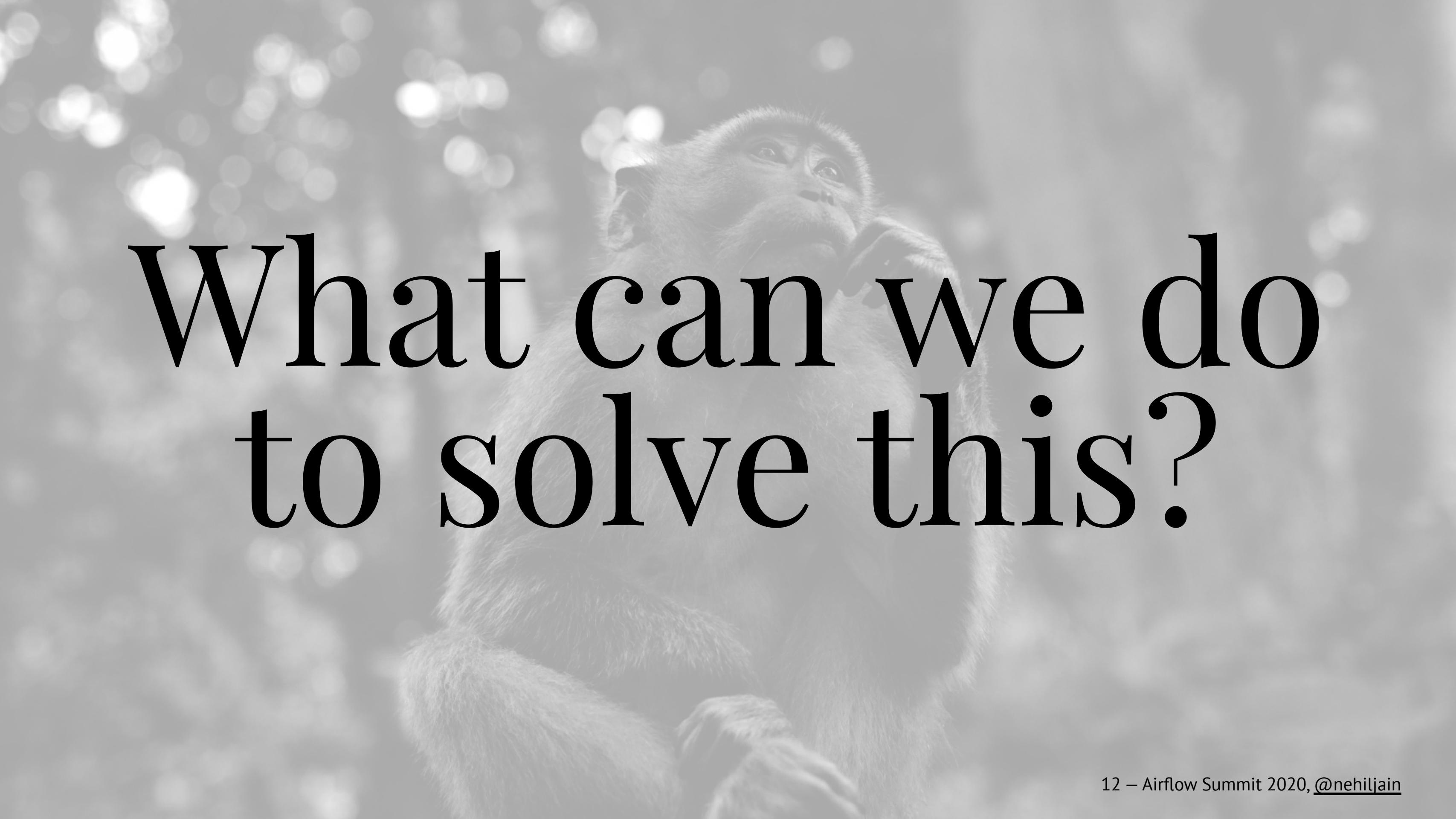


Been there  
done that?



# Classify the problems

- ▶ Toil
- ▶ Data Discovery
- ▶ Data Trust
- ▶ Throw over the boundary
- ▶ Ambiguous ownership
- ▶ Cannot scale Data Analytics



What can we do  
to solve this?

# ..build tools, infrastructure, frameworks and services

## – Maxime Beauchemin



# Design Requirements



**NO PATIENCE  
REQUIRED**

**FREE SAME-DAY DELIVERY**

~~NEW IN~~  
**\*NEW IN\***  
**YOUR AREA**  
~~FROM~~

# Single Source of Truth

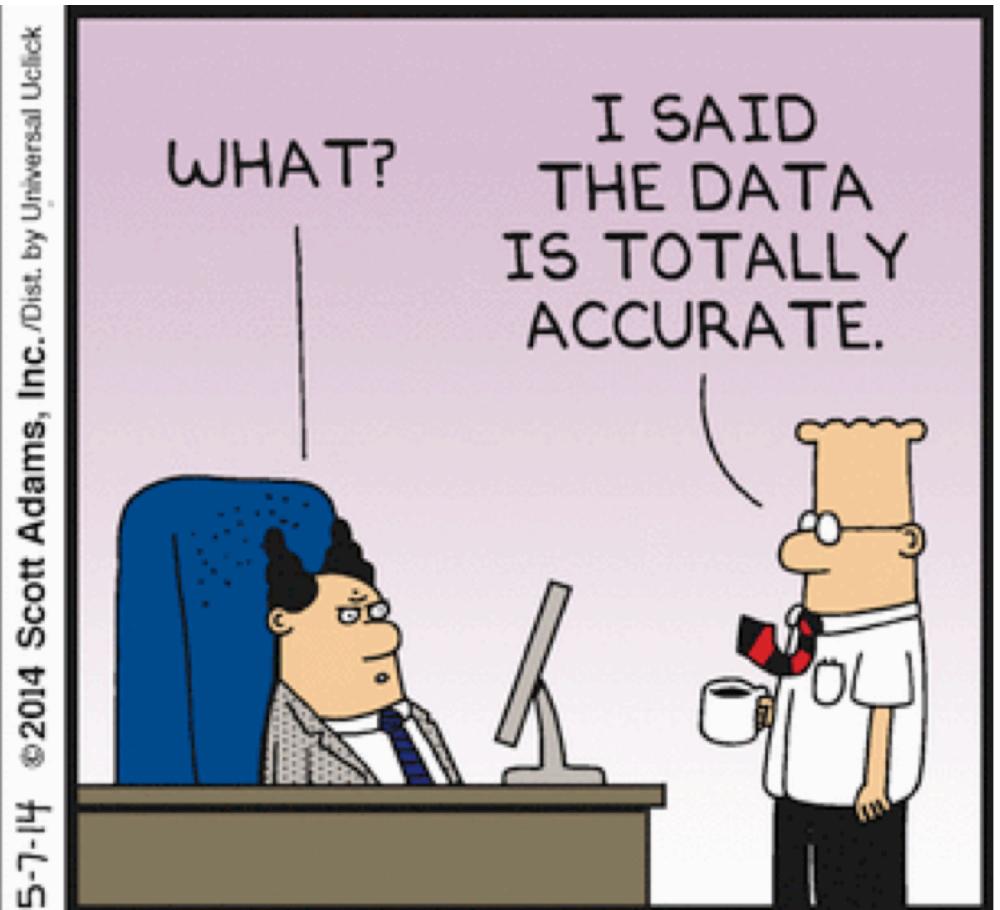
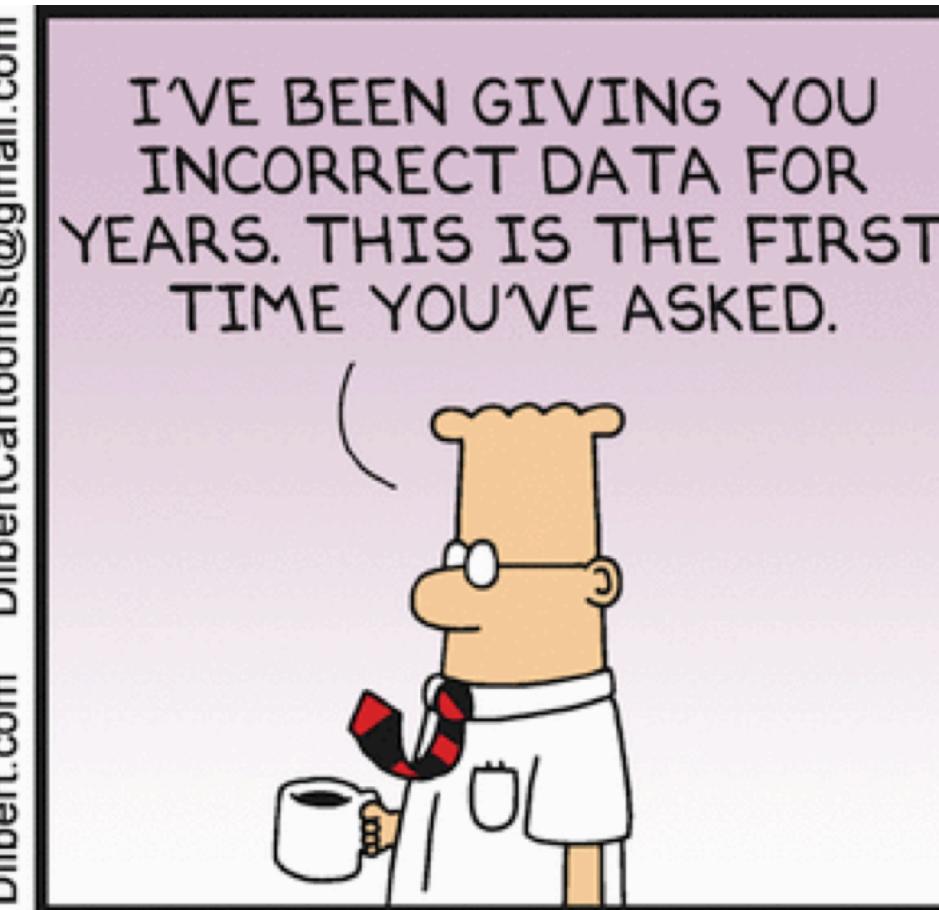
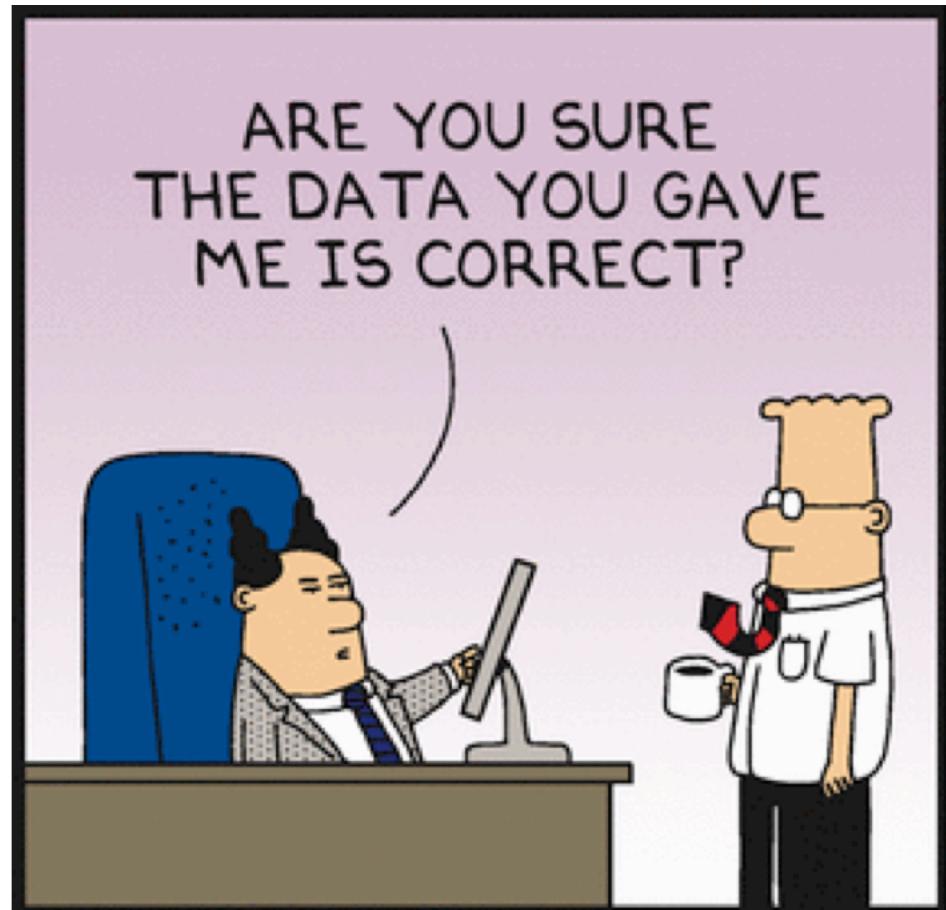
- ▶ Standardization
- ▶ Data Lineage
- ▶ Empower non-technical folks

# Easy to consume

- ▶ Airflow + Other OSS
- ▶ Ideally pip install awesome-elt-tool
- ▶ Low barrier to entry for data analytics
- ▶ Operational creep

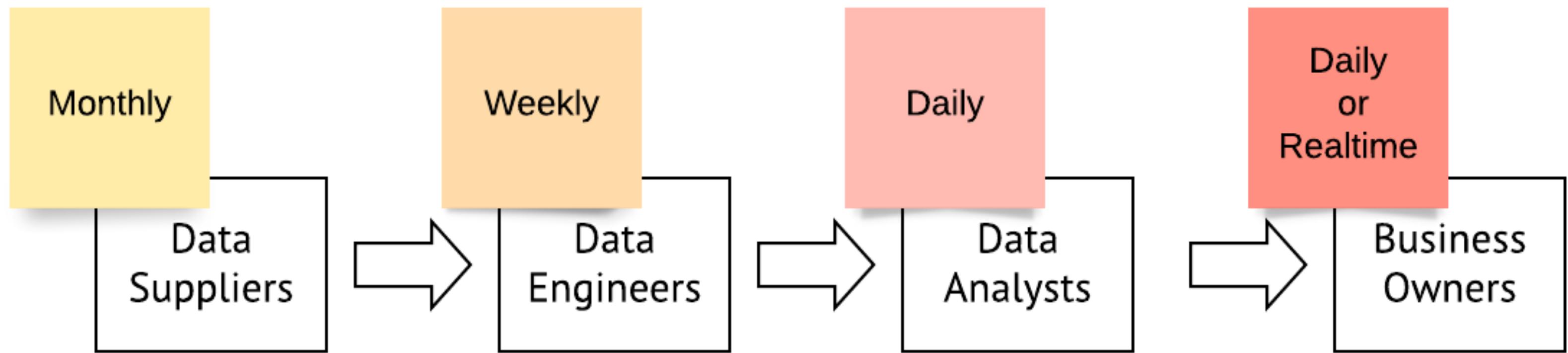
# Promote data integrity

- ▶ Test the raw data supply
- ▶ Automated analytics testing



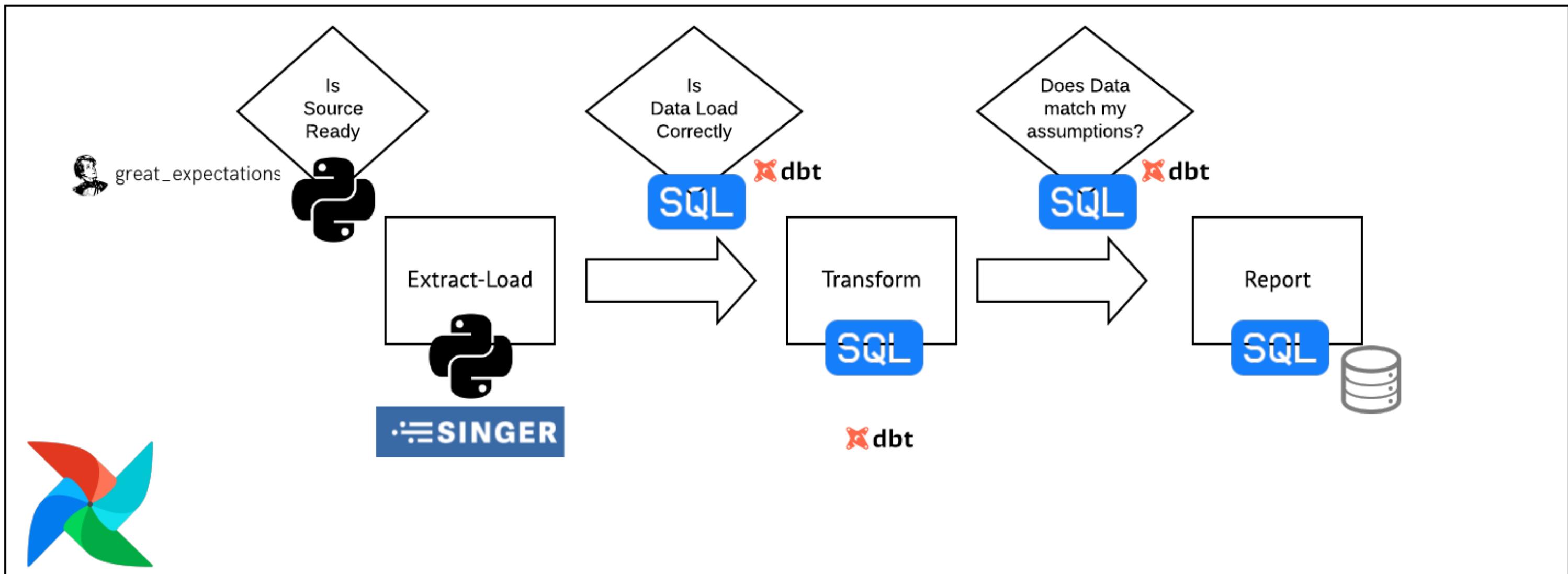
# Meta Data Engineering





# Proposed Solution

# Conceptually



# ETL vs ELT

- ▶ Load once and transform
- ▶ Reduced complexity
- ▶ Reduce cost
- ▶ Speed of delivery



validate your  
source data



# great\_expectations

Always know what to expect from your data

- ▶ `expect_column_to_exist`
- ▶ `expect_table_row_count_to_be_between`
- ▶ `expect_table_row_count_to_equal`
- ▶ `expect_multicolumn_values_to_be_unique`
- ▶ `expect_column_values_to_not_be_null`
- ▶ `expect_column_values_to_be_null`
- ▶ `expect_column_fancy_statistic_to_be`



# Why?

- ▶ Profiling
- ▶ Data Docs <-> Tests
- ▶ Send notifications automatically

A black and white photograph of a juicer. A large orange is being pressed into the juicer's compartment, which is filled with smaller pieces of fruit and pulp. The juicer has a clear plastic container attached to the bottom.

# Extract – Load

# Singer - What?



```
tap-github --config tap_config.json | target-postgres --config target_config.json >> state.json
```

# Singer - Why?

- ▶ Standardized communication
- ▶ Incremental out of the box
- ▶ Documentation
- ▶ See your data in under 10 mins

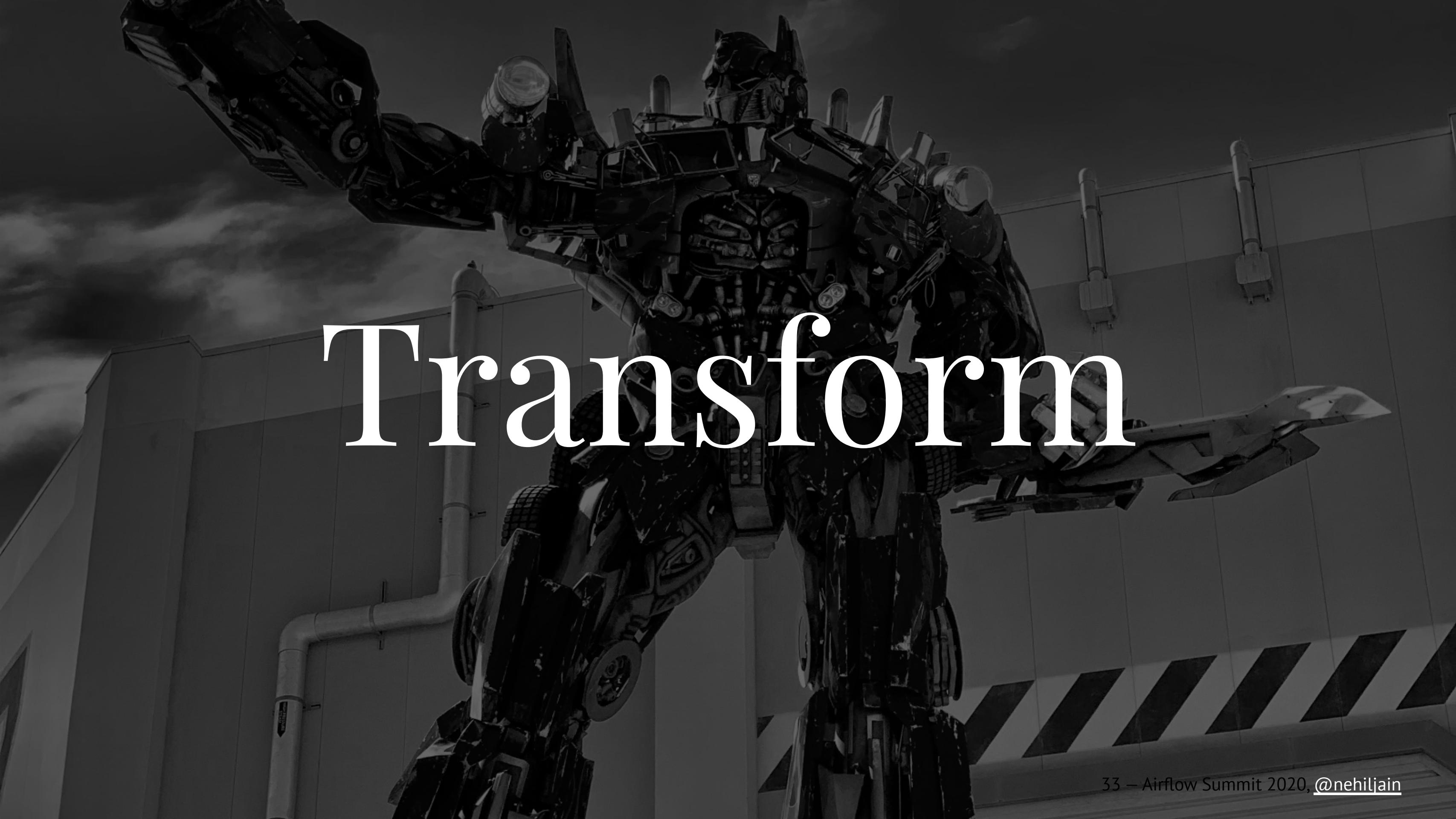
### Singer Taps

Taps extract data from any source and write it to a standard stream in a JSON-based format.

 3PL Central	 Amazon S3 CSV	 Amplitude
 Appsflyer	 Autopilot	 BigCommerce
 Bing Ads	 Braintree	 Bronto
 COVID-19 Public Data	 Campaign Manager	 Campaign Monitor
 Chargebee	 Close	 Club Speed
 Codat	 Dark Sky	 Deputy
 Dynamo DB	Eloqua	 Exchange Rates API
 Facebook Ads	 Facebook Reviews	 Freshdesk
 Front	 FullStory	 GitHub
 GitLab	Google Ads	Google Analytics
 Google Analytics 360	 Google Sheets	 Harvest
 Harvest Forecast	 Heap	 HubSpot
 IBM Db2	 Impact	 Invoiced
 jira	 Klaviyo	 Kustomer
 Lever	 LinkedIn Ads	 Listrak
 LivePerson	 Mailshake	 Mambu
 Marketo	 Mixpanel	 MySQL
 Netsuite Suite Analytics	 Onfleet	 Oracle
 Outbrain	 Outreach	 Pardot
 Pepperjam	 Pipedrive	 Platform Purple
 PostgreSQL	 PureCloud	 Quick Base
 Recharge	 Recurly	 Referral SaaSquatch
 Responsys	 Revinate	 SFTP
 SaaS Optics	 Salesforce	 Salesforce Marketing Cloud
 Seligent	 SendGrid Core	 ShipHero
 Shippo	 Shopify	 Stripe
 SurveyMonkey	 Taboola	 Toggl
 Trello	 Typeform	 Urban Airship
 Uservoice	 WooCommerce	 Wootric
 Workday RaaS	 Xero	 Yotpo
 Zendesk Chat	 Zendesk Support	 Zuora
 Develop a Tap		

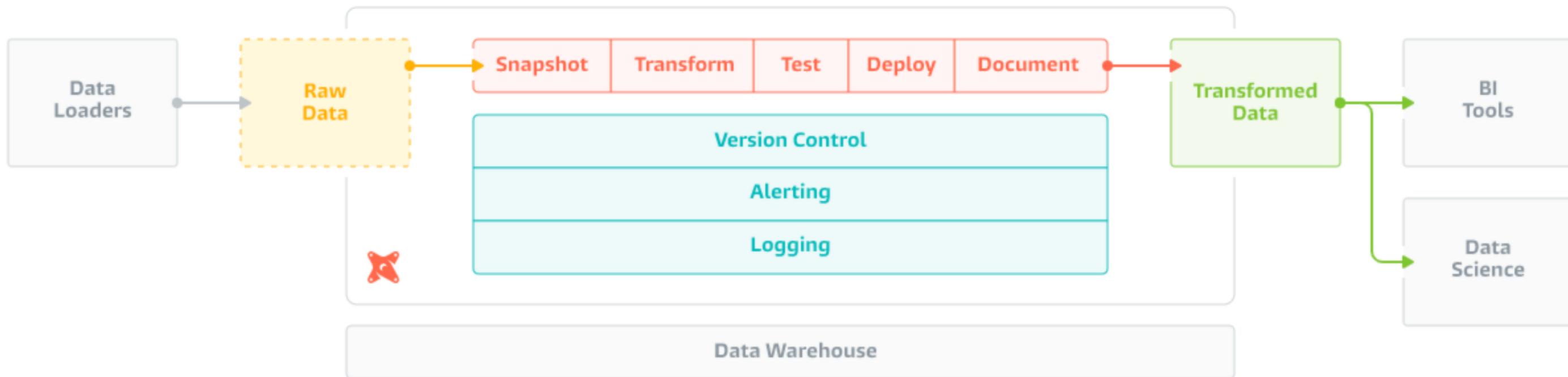
# It's a long list

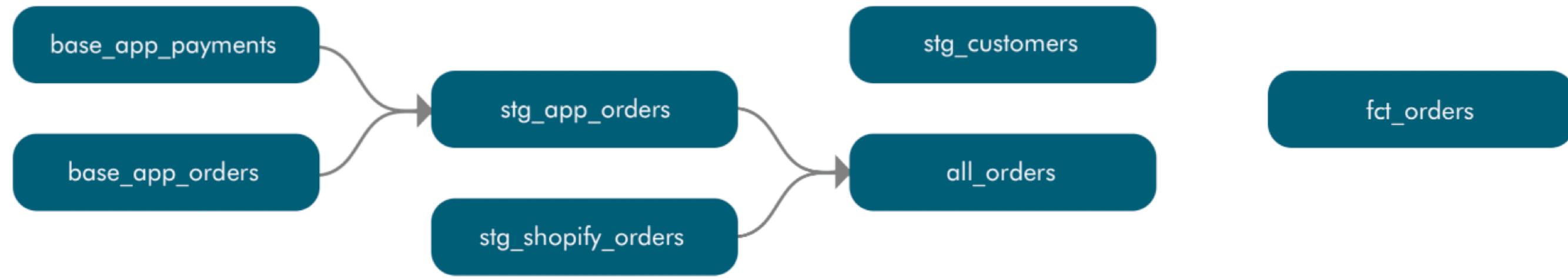




# Transform

# DBT - What?





Building dependencies  
between dbt models

`fct_orders.sql`

```
select
    orders.order_id,
    orders.customer_id,
    customers.first_name as customer_first_name,
    customers.last_name as customer_last_name,
    orders.amount,
    orders.ordered_at
```

dbt Docs

getdbt.com/mrr-playbook/#!model/model.acme.mrr

# dbt

Search for models...

**Overview**

**mrr view**

Details Description Columns SQL

**Details**

TAGS	OWNER	TYPE	PACKAGE	RELATION
untagged	TRANSFORMER	view	acme	analytics.dbt_claire_playbook.mrr

**Description**

This model represents one record per month, per account (months have been filled in to include any periods where no data was available).

This model classifies each month as one of: new, reactivation, upgrade, downgrade, or churn.

**Columns**

COLUMN	TYPE	DESCRIPTION
id	TEXT	
date_month	TIMESTAMP_NTZ	
customer_id	NUMBER	

**Lineage Graph**

```

graph TD
    CRM[customer_revenue_by_month] --> MRR[mrr]
    CCM[customer_churn_month] --> MRR
  
```

<https://www.getdbt.com/mrr-playbook/#!model/model.acme.mrr#description>

# DBT - Why?

## ► Modular code

🔗 /models/order\_payment\_method\_amounts.sql

```
{% set payment_methods = ["bank_transfer", "credit_card", "gift_card"] %}

select
    order_id,
    {% for payment_method in payment_methods %}
    sum(case when payment_method = '{{payment_method}}' then amount end) as {{payment_method}}
    {% endfor %}
    sum(amount) as total_amount
from app_data.payments
group by 1
```

# DBT - Why?

- ▶ Modular code
- ▶ Testing is 1st Class

```
- name: orders
  columns:
    - name: order_id
      tests:
        - unique
        - not_null
    - name: status
      tests:
        - accepted_values:
            values: ['placed', 'shipped', 'completed', 'returned']
    - name: customer_id
      tests:
```

# DBT - Why?

- ▶ Modular code
- ▶ Testing is 1st Class
- ▶ Data documentation is 1st Class

```
description: This table contains clickstream events from the marketing website

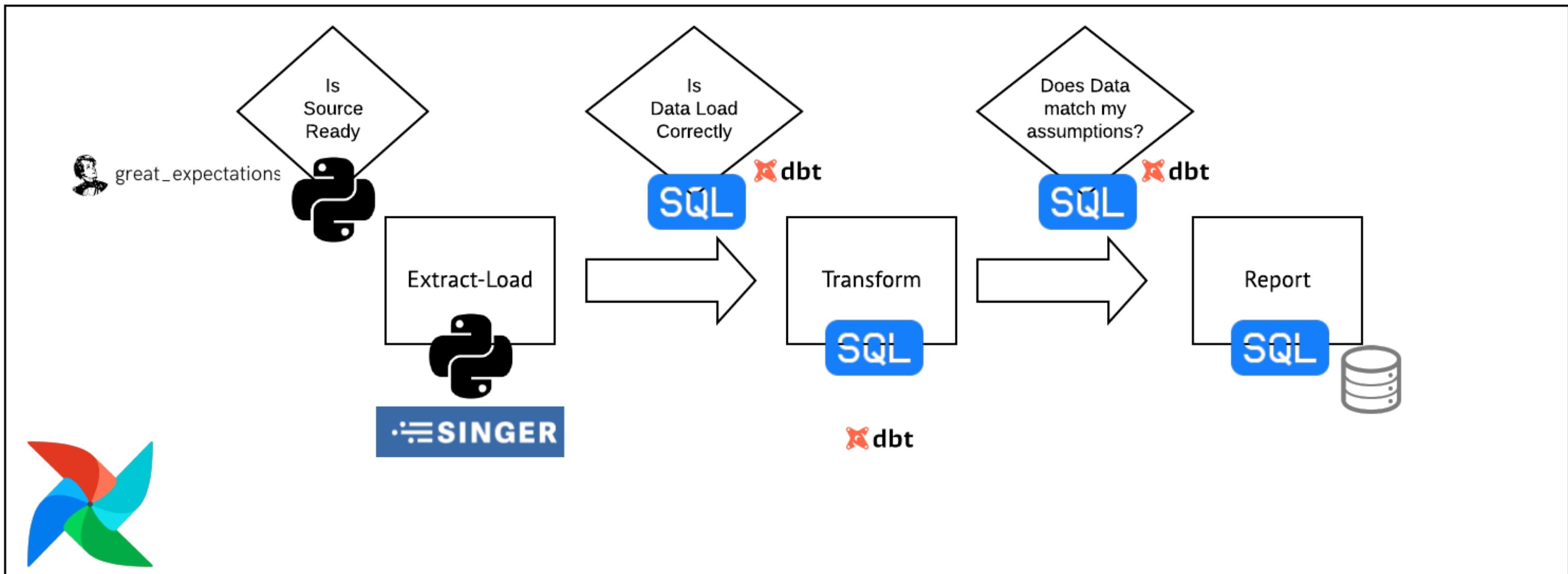
columns:
  - name: event_id
    description: This is a unique identifier for the event
    tests:
      - unique
      - not_null
```

# Great adoption

Weekly Active dbt Projects



# All together



# Meltano

- ▶ Open Source, GitLab
- ▶ Self Hosted

```
pip3 install meltano
meltano init airflow-analytics-project
meltano add extractor tap-github
meltano add loader target-postgres
meltano add transformer dbt
meltano add transform tap-github
# add env variables
meltano elt tap-gitlab target-postgres --transform=run --job_id=gitlab-to-postgres
meltano add orchestrator airflow
```

Let's look at the  
code

```
1 version: 1
2 send_anonymous_usage_stats: true
3 project_id: [REDACTED]
4 plugins:
5   extractors:
6     - name: tap-github
7       namespace: tap_github
8       pip_url: 'git+https://github.com/nehiljain/tap-github.git'
9       executable: tap-github
10      capabilities:
11        - discover
12        - properties
13      settings:
14        - name: access_token
15          env: TAP_GITHUB_ACCESS_TOKEN
16        - name: repository
17          env: TAP_GITHUB_REPOSITORY
18      loaders:
19        - name: target-postgres
20          pip_url: 'git+https://github.com/meltano/target-postgres.git'
21      transforms:
22        - name: tap-github
23          pip_url: 'https://github.com/nehiljain/dbt-tap-github.git'
24      orchestrators:
25        - name: airflow
26          pip_url: wtforms==2.2.1 apache-airflow==1.10.2
27      transformers:
28        - name: dbt
29          pip_url: dbt==0.16.1
30      files:
31        - name: airflow
32          pip_url: 'git+https://gitlab.com/meltano/files-airflow.git'
33      schedules:
34        - name: gitlab-to-postgres
35          extractor: tap-github
36          loader: target-postgres
37          transform: skip
38          interval: '@hourly'
39          start_date: 2020-07-05 18:58:28.155924
```

# A templated approach

```
1 version: 1
2 send_anonymous_usage_stats: true
3 project_id: [REDACTED]
4 plugins:
5   extractors:
6     - name: tap-github
7       namespace: tap_github
8       pip_url: 'git+https://github.com/nehiljain/tap-github.git'
9       executable: tap-github
10      capabilities:
11        - discover
12        - properties
13      settings:
14        - name: access_token
15          env: TAP_GITHUB_ACCESS_TOKEN
16        - name: repository
17          env: TAP_GITHUB_REPOSITORY
18      loaders:
19        - name: target-postgres
20          pip_url: 'git+https://github.com/meltano/target-postgres.git'
21      transforms:
22        - name: tap-github
23          pip_url: 'https://github.com/nehiljain/dbt-tap-github.git'
24      orchestrators:
25        - name: airflow
26          pip_url: 'wtforms==2.2.1 apache-airflow==1.10.2'
27      transformers:
28        - name: dbt
29          pip_url: dbt==0.16.1
30      files:
31        - name: airflow
32          pip_url: 'git+https://gitlab.com/meltano/files-airflow.git'
33      schedules:
34        - name: gitlab-to-postgres
35          extractor: tap-github
36          loader: target-postgres
37          transform: skip
38          interval: '@hourly'
39          start_date: 2020-07-05 18:58:28.155924
```

```
1 version: 1
2 send_anonymous_usage_stats: true
3 project_id: [REDACTED]
4 plugins:
5   extractors:
6     - name: tap-github
7       namespace: tap_github
8       pip_url: 'git+https://github.com/nehiljain/tap-github.git'
9       executable: tap-github
10      capabilities:
11        - discover
12        - properties
13      settings:
14        - name: access_token
15          env: TAP_GITHUB_ACCESS_TOKEN
16        - name: repository
17          env: TAP_GITHUB_REPOSITORY
18      loaders:
19        - name: target-postgres
20          pip_url: 'git+https://github.com/meltano/target-postgres.git'
21      transforms:
22        - name: tap-github
23          pip_url: 'https://github.com/nehiljain/dbt-tap-github.git'
24      orchestrators:
25        - name: airflow
26          pip_url: wtforms==2.2.1 apache-airflow==1.10.2
27      transformers:
28        - name: dbt
29          pip_url: dbt==0.16.1
30      files:
31        - name: airflow
32          pip_url: 'git+https://gitlab.com/meltano/files-airflow.git'
33      schedules:
34        - name: gitlab-to-postgres
35          extractor: tap-github
36          loader: target-postgres
37          transform: skip
38          interval: '@hourly'
39          start_date: 2020-07-05 18:58:28.155924
```

```
1 version: 1
2 send_anonymous_usage_stats: true
3 project_id: [REDACTED]
4 plugins:
5   extractors:
6     - name: tap-github
7       namespace: tap_github
8       pip_url: 'git+https://github.com/nehiljain/tap-github.git'
9       executable: tap-github
10      capabilities:
11        - discover
12        - properties
13      settings:
14        - name: access_token
15          env: TAP_GITHUB_ACCESS_TOKEN
16        - name: repository
17          env: TAP_GITHUB_REPOSITORY
18      loaders:
19        - name: target-postgres
20          pip_url: 'git+https://github.com/meltano/target-postgres.git'
21      transforms:
22        - name: tap-github
23          pip_url: 'https://github.com/nehiljain/dbt-tap-github.git'
24      orchestrators:
25        - name: airflow
26          pip_url: wtforms==2.2.1 apache-airflow==1.10.2
27      transformers:
28        - name: dbt
29          pip_url: dbt==0.16.1
30      files:
31        - name: airflow
32          pip_url: 'git+https://gitlab.com/meltano/files-airflow.git'
33      schedules:
34        - name: gitlab-to-postgres
35          extractor: tap-github
36          loader: target-postgres
37          transform: skip
38          interval: '@hourly'
39          start_date: 2020-07-05 18:58:28.155924
```

```
1 version: 1
2 send_anonymous_usage_stats: true
3 project_id: [REDACTED]
4 plugins:
5   extractors:
6     - name: tap-github
7       namespace: tap_github
8       pip_url: 'git+https://github.com/nehiljain/tap-github.git'
9       executable: tap-github
10      capabilities:
11        - discover
12        - properties
13      settings:
14        - name: access_token
15          env: TAP_GITHUB_ACCESS_TOKEN
16        - name: repository
17          env: TAP_GITHUB_REPOSITORY
18      loaders:
19        - name: target-postgres
20          pip_url: 'git+https://github.com/meltano/target-postgres.git'
21      transforms:
22        - name: tap-github
23          pip_url: 'https://github.com/nehiljain/dbt-tap-github.git'
24    orchestrators:
25      - name: airflow
26        pip_url: wtforms==2.2.1 apache-airflow==1.10.2
27    transformers:
28      - name: dbt
29        pip_url: dbt==0.16.1
30    files:
31      - name: airflow
32        pip_url: 'git+https://gitlab.com/meltano/files-airflow.git'
33    schedules:
34      - name: gitlab-to-postgres
35        extractor: tap-github
36        loader: target-postgres
37        transform: skip
38        interval: '@hourly'
39        start_date: 2020-07-05 18:58:28.155924
```

```
1 version: 1
2 send_anonymous_usage_stats: true
3 project_id: [REDACTED]
4 plugins:
5   extractors:
6     - name: tap-github
7       namespace: tap_github
8       pip_url: 'git+https://github.com/nehiljain/tap-github.git'
9       executable: tap-github
10      capabilities:
11        - discover
12        - properties
13      settings:
14        - name: access_token
15          env: TAP_GITHUB_ACCESS_TOKEN
16        - name: repository
17          env: TAP_GITHUB_REPOSITORY
18      loaders:
19        - name: target-postgres
20          pip_url: 'git+https://github.com/meltano/target-postgres.git'
21      transforms:
22        - name: tap-github
23          pip_url: 'https://github.com/nehiljain/dbt-tap-github.git'
24      orchestrators:
25        - name: airflow
26          pip_url: wtforms==2.2.1 apache-airflow==1.10.2
27      transformers:
28        - name: dbt
29          pip_url: dbt==0.16.1
30      files:
31        - name: airflow
32          pip_url: 'git+https://gitlab.com/meltano/files-airflow.git'
33      schedules:
34        - name: gitlab-to-postgres
35          extractor: tap-github
36          loader: target-postgres
37          transform: skip
38          interval: '@hourly'
39          start_date: 2020-07-05 18:58:28.155924
```

# Sit back & Relax

# Some challenges out there

- ▶ Visualisation/BI layer
- ▶ Analytics code coverage
- ▶ Singer community

# Key Takeaways

- ▶ Standardized tooling
- ▶ ELT >> ETL
- ▶ GE + Singer + DBT orchestrated by Airflow

# Thanks

# Q & A

# Resources

- ▶ Meltano Project
- ▶ Advanced Data Engineering Patterns with Apache Airflow by Maxime Beauchemin
- ▶ The Rise of the Data Engineer
- ▶ The Future of Data Engineering
- ▶ Downfall of the data engineer

# Resources

- ▶ [Singer | Open Source ETL](#)
- ▶ [Why we are building an open-source platform for ELT pipelines - Meltano](#)
- ▶ [Dbt Docs](#)
- ▶ []