Building
Reusable and Trustworthy
pipelines

Outline

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

Context

```
        self.logdupes = True
        self.debug = debug
        self.logger = logging.getLogger(__name__)
        if path:
            self.file = open(os.path.join(path, 'requests.seen'), 'a+')
            self.file.seek(0)
            self.fingerprints.update(x.rstrip() for x in self.file)

    @classmethod
    def from_settings(cls, settings):
        debug = settings.getbool('DUPEFILTER_DEBUG')
        return cls(job_dir(settings), debug)

    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
```

Hello 👋!

- Data engineer @ SnapTravel
- SnapTravel
  - M-commerce startup
  - Data team: 8, Data Sources: 86
- Data infrastructure, Data engineering, Analytics engineering
- ☰ + ✤ + ❄ + ✖ stack

Purpose

Share 🧰 BI pipelines

💡 Community with lessons learnt

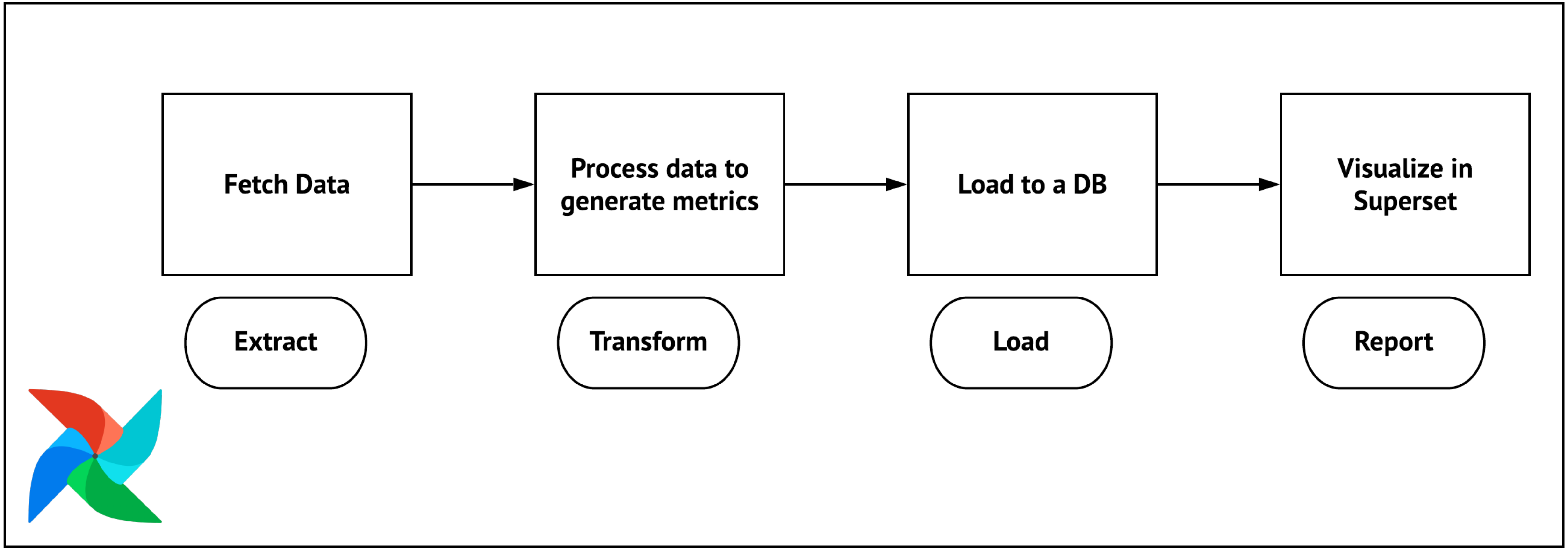
👂 feedback

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_commits`



Fetch Data

Process data to generate metrics

Load to a DB

Visualize in Superset

Extract

Transform

Load

Report

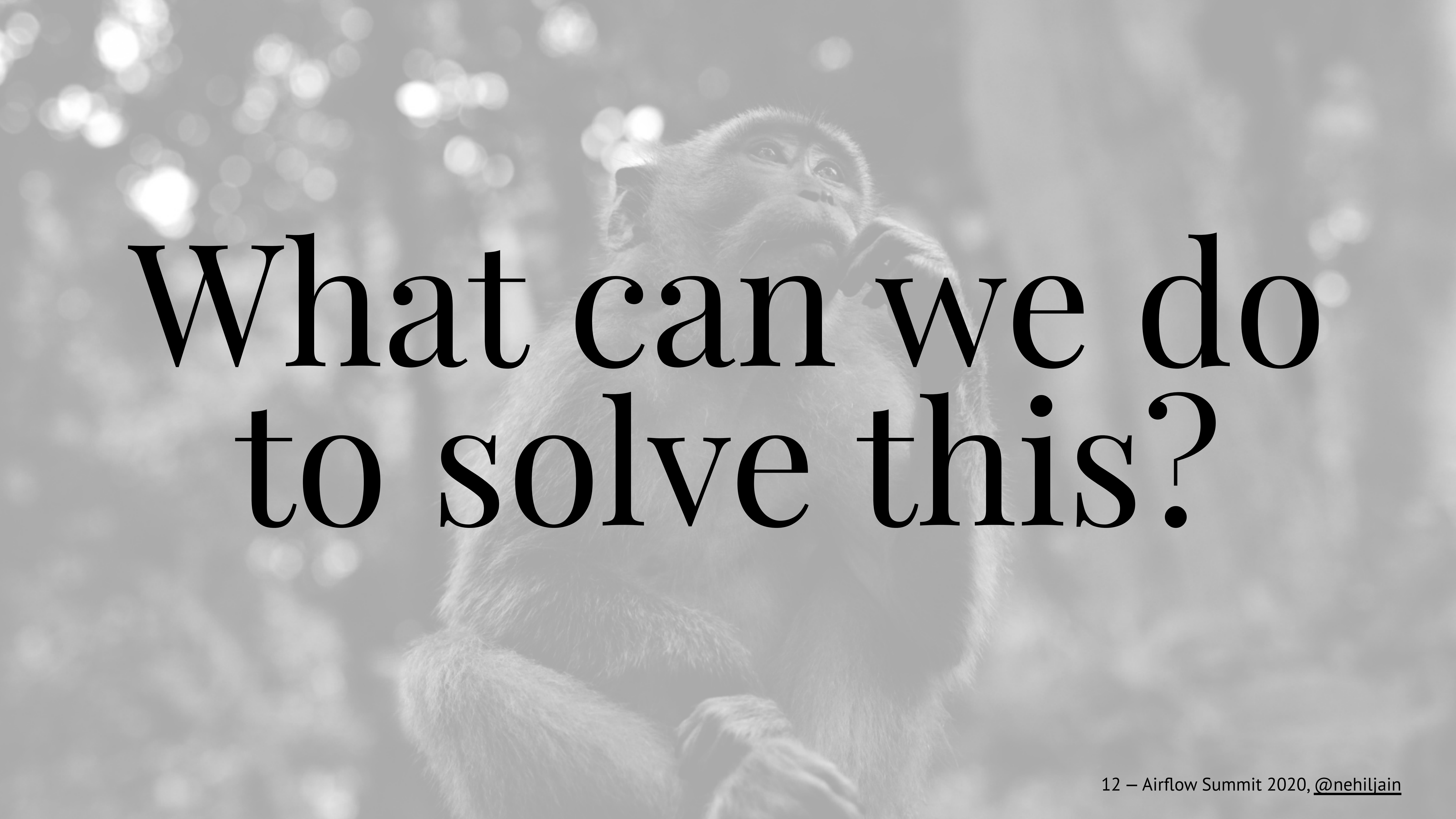
Let us consider

- The pipeline failed in production
- Shift focus on to issues, comments
- Gitlab released a new version of API
- I want to analyze other apache projects too
- Github produced similar insights and their numbers didn't match mine

🙋‍♀️ Been there done that? 🙋‍♂️

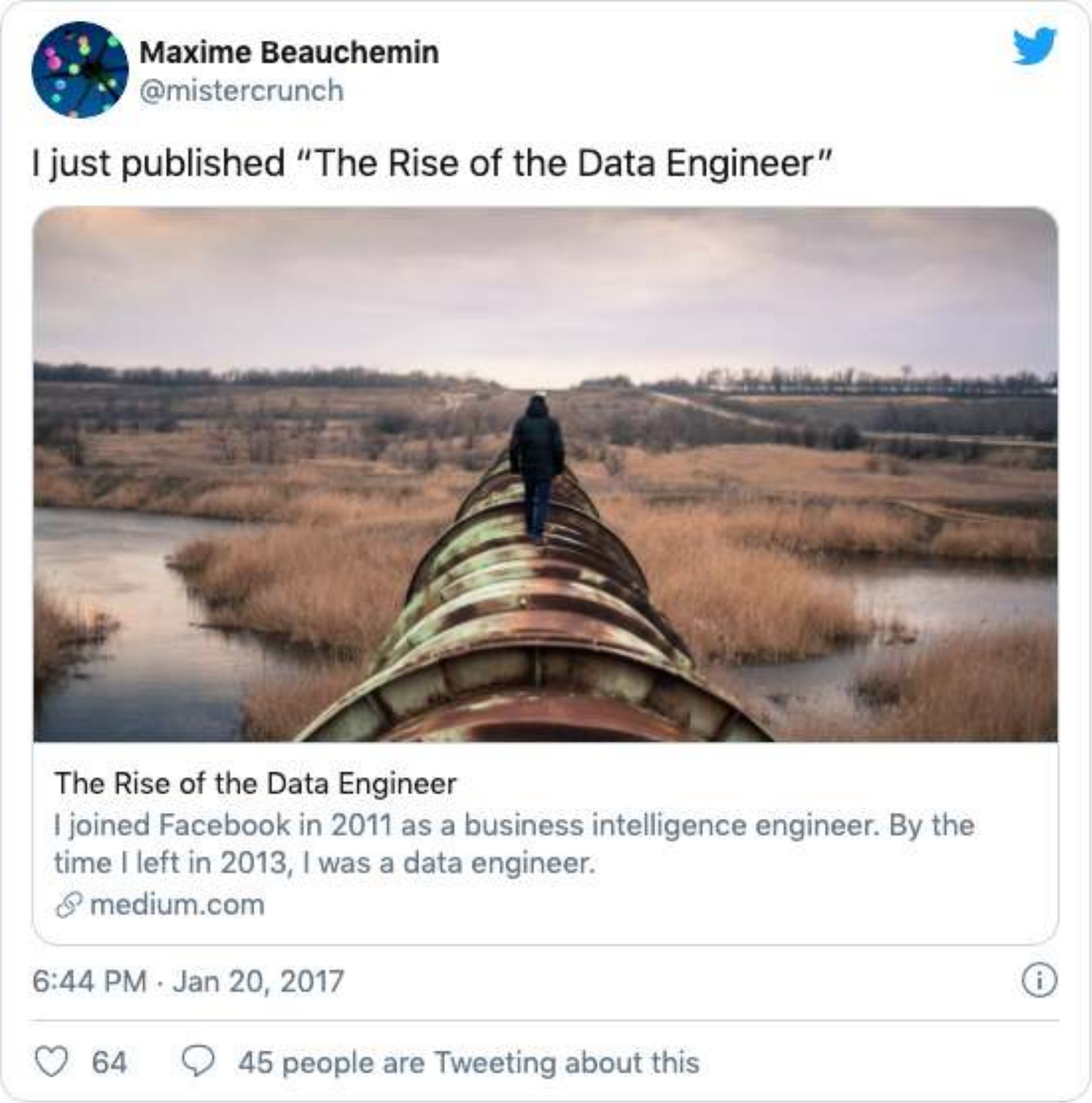
Classify the problems

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



What can we do to solve this?

..build tools, infrastructure, frameworks and services — Maxime Beauchemin



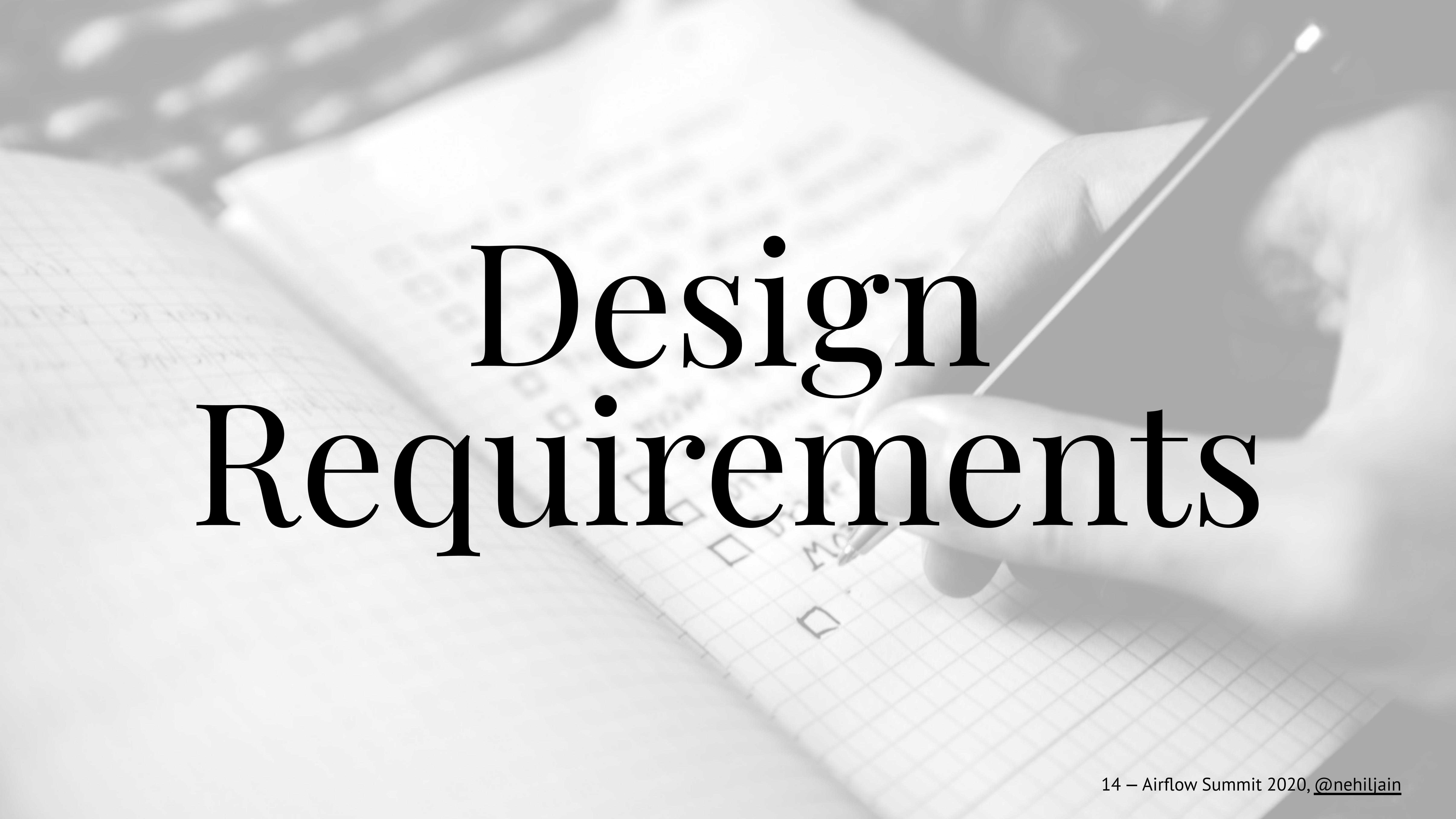
Maxime Beauchemin
@mistercrunch

I just published "The Rise of the Data Engineer"

The Rise of the Data Engineer
I joined Facebook in 2011 as a business intelligence engineer. By the time I left in 2013, I was a data engineer.
medium.com

6:44 PM · Jan 20, 2017

♡ 64 45 people are Tweeting about this



Design Requirements

amazon Prime
amazon.com/sameday
FREE SAME-DAY DELIVERY
NO PATIENCE REQUIRED
FREE SAME-DAY DELIVERY
NEW IN YOUR AREA

Single Source of Truth

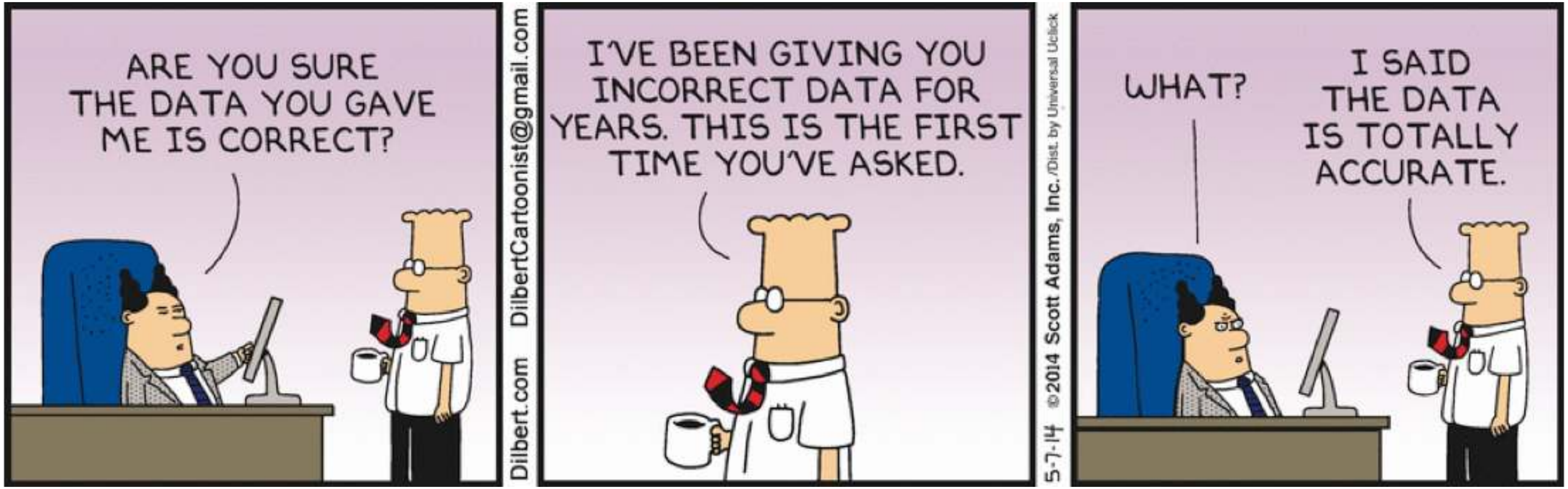
- Standardization
- Data Lineage
- Empower non-technical folks

Easy to consume

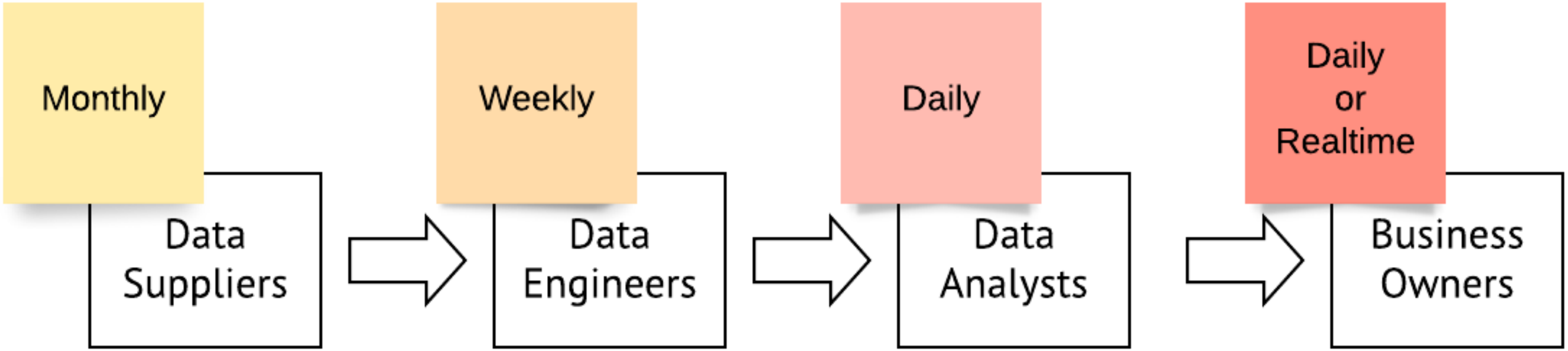
- Airflow + Other OSS
- Ideally `pip install awesome-elt-tool`
- Low barrier to entry for data analytics
- Operational creep

Promote data integrity

- Test the raw data supply
- Automated analytics testing


ARE YOU SURE THE DATA YOU GAVE ME IS CORRECT?
Dilbert.com DilbertCartoonist@gmail.com
I'VE BEEN GIVING YOU INCORRECT DATA FOR YEARS. THIS IS THE FIRST TIME YOU'VE ASKED.
5-7-14 ©2014 Scott Adams, Inc. /Dist. by Universal Uclick
WHAT?
I SAID THE DATA IS TOTALLY ACCURATE.

Meta Data Engineering



Monthly

Data Suppliers

Weekly

Data Engineers

Daily

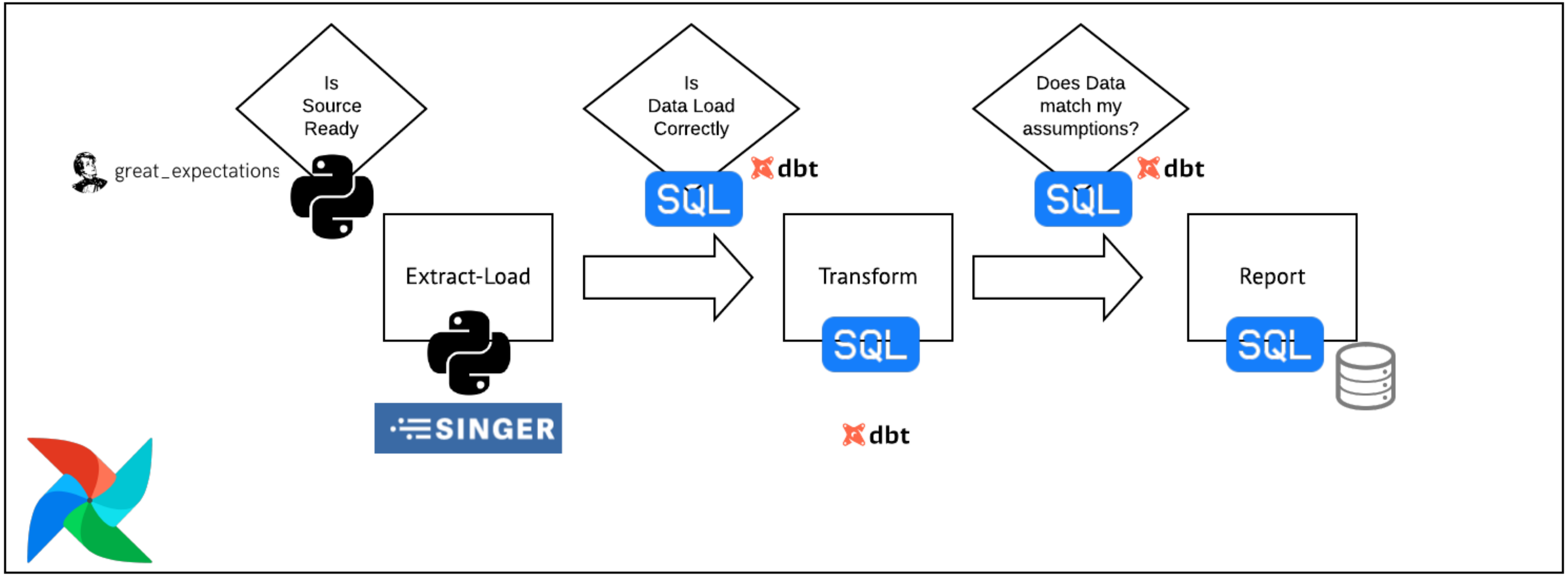
Data Analysts

Daily or Realtime

Business Owners

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

pandas

SQLAlchemy

snowflake

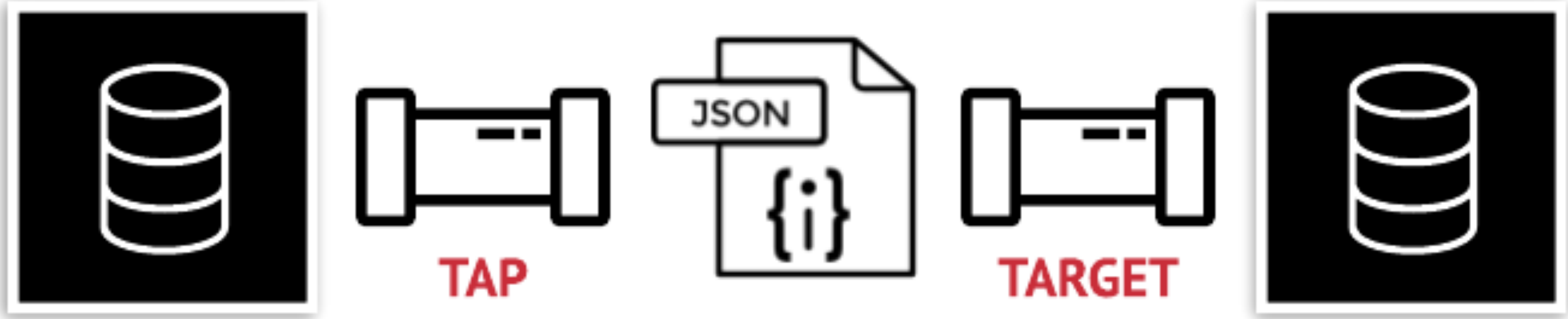
APACHE Spark

Why?

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

Extract - Load

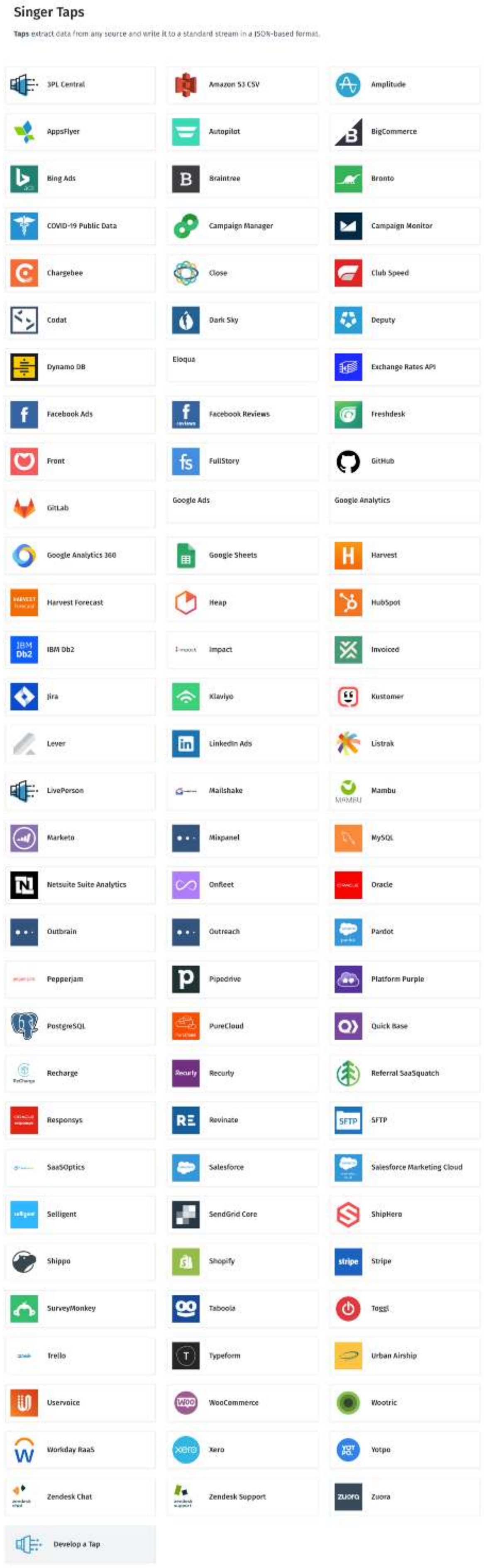
Singer - What?



```
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

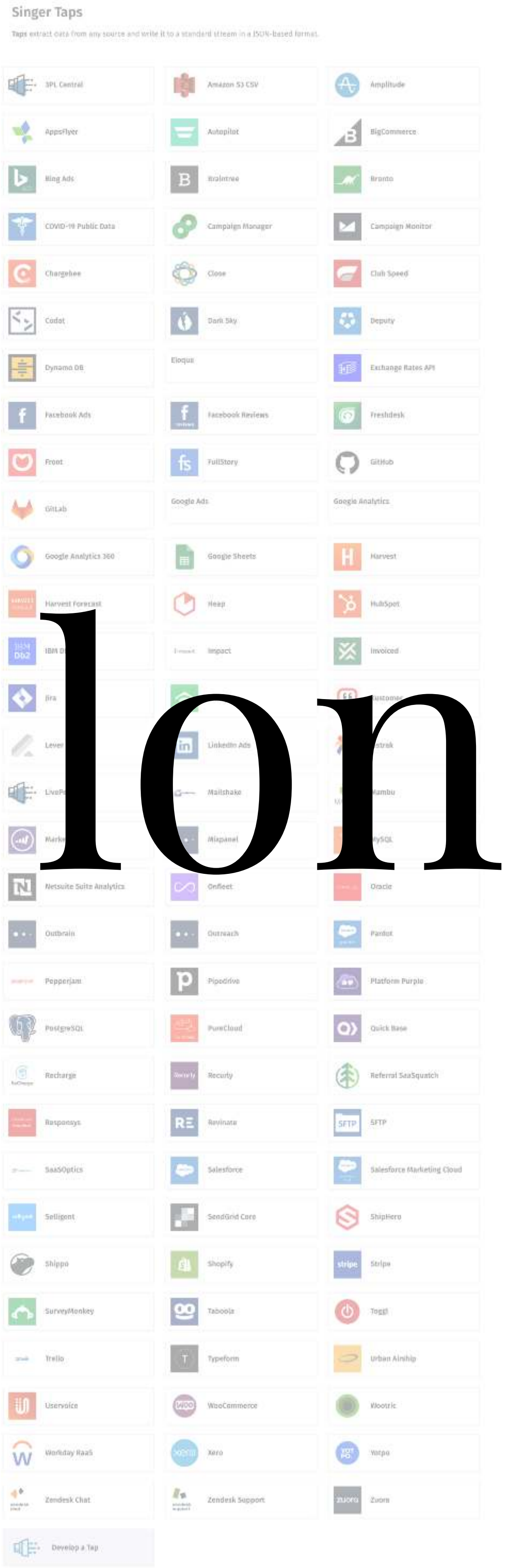


Singer Taps

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
- SaaSOptics
- Salesforce
- Salesforce Marketing Cloud
- Selligent
- 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

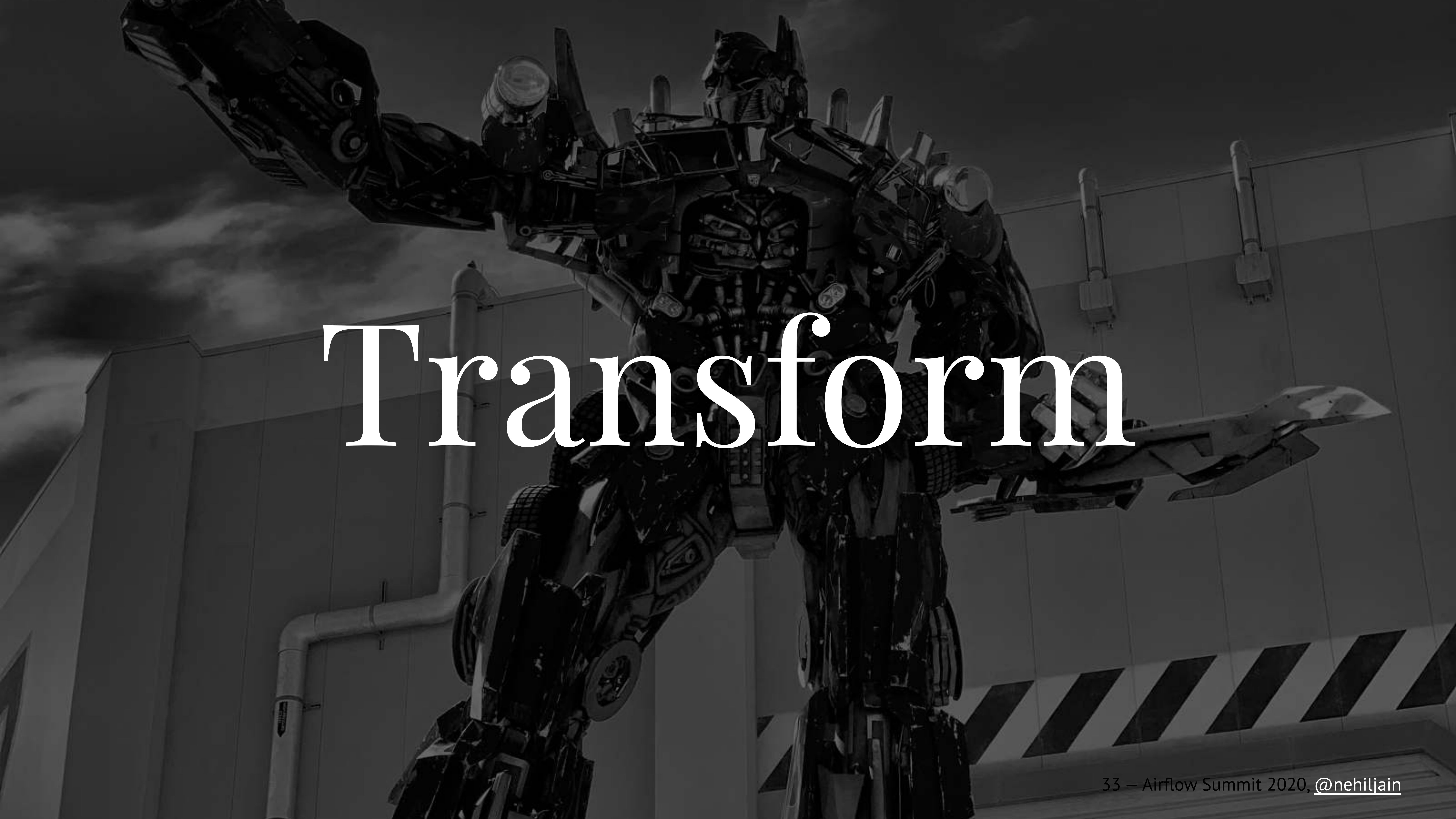
It's a long list



Singer Taps

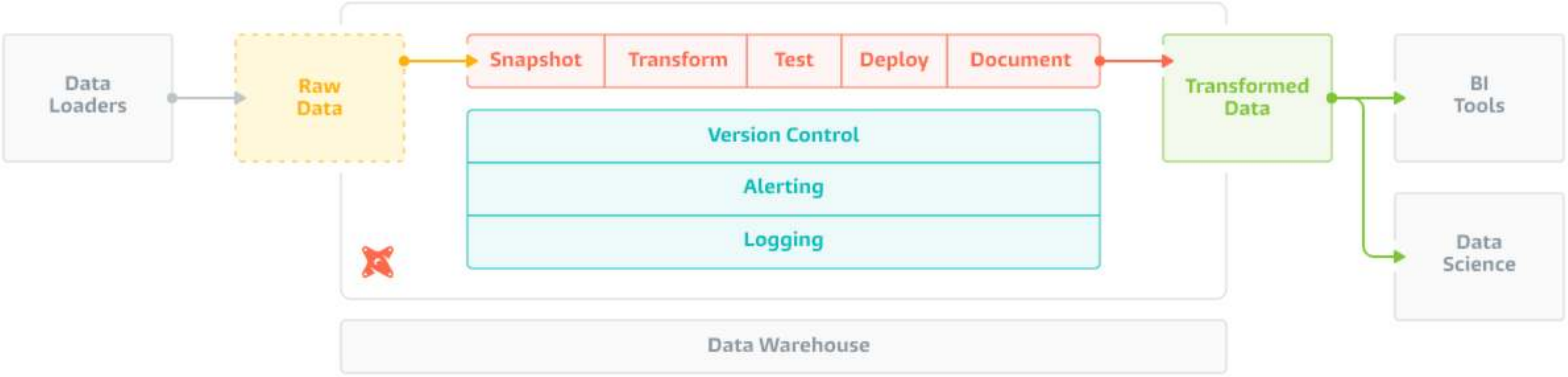
Taps extract data from any source and write it to a standard stream in a JSON-based format.

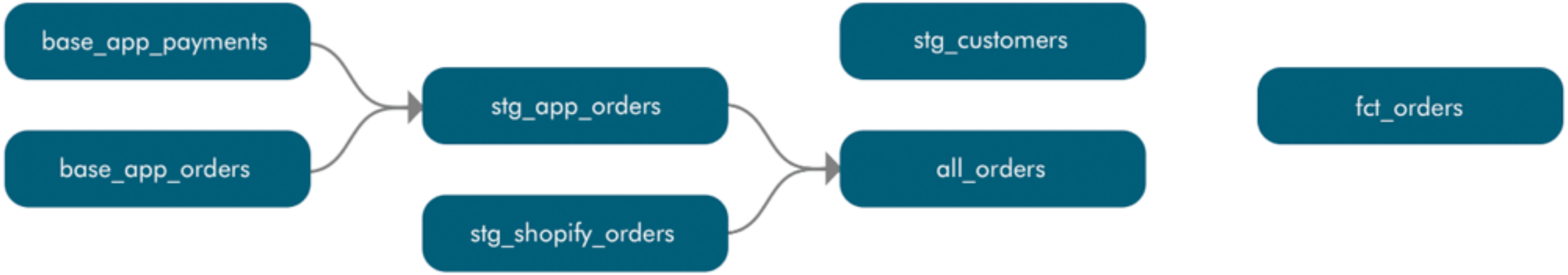
- 3PL Central
- Amazon S3 CSV
- Amplitude
- AppsFlyer
- Autopilot
- BigCommerce
- Bing Ads
- Braintree
- Bronto
- COVID-19 Public Data
- Campaign Manager
- Campaign Monitor
- Chargebee
- Close
- Club Speed
- Codat
- Dark Sky
- Deputy
- Dynamo DB
- Eloqua
- Exchange Rates API
- Facebook Ads
- Facebook Reviews
- Freshdesk
- Front
- FullStory
- GitHub
- GitLab
- Google Ads
- Google Analytics
- Google Analytics 360
- Google Sheets
- Harvest
- Harvest Forecast
- Heap
- HubSpot
- IBM D
- Impact
- Invoiced
- Jira
- ustomer
- Lever
- LinkedIn Ads
- strak
- LiveP
- Mailshake
- ambu
- Marke
- Mixpanel
- ySQL
- Netsuite Suite Analytics
- Onfleet
- Oracle
- Outbrain
- Outreach
- Pardot
- Pepperjam
- Pipedrive
- Platform Purple
- PostgreSQL
- PureCloud
- Quick Base
- Recharge
- Recurly
- Referral SaaSquatch
- Responsys
- Revinate
- SFTP
- SaaSOptics
- Salesforce
- Salesforce Marketing Cloud
- Selligent
- 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

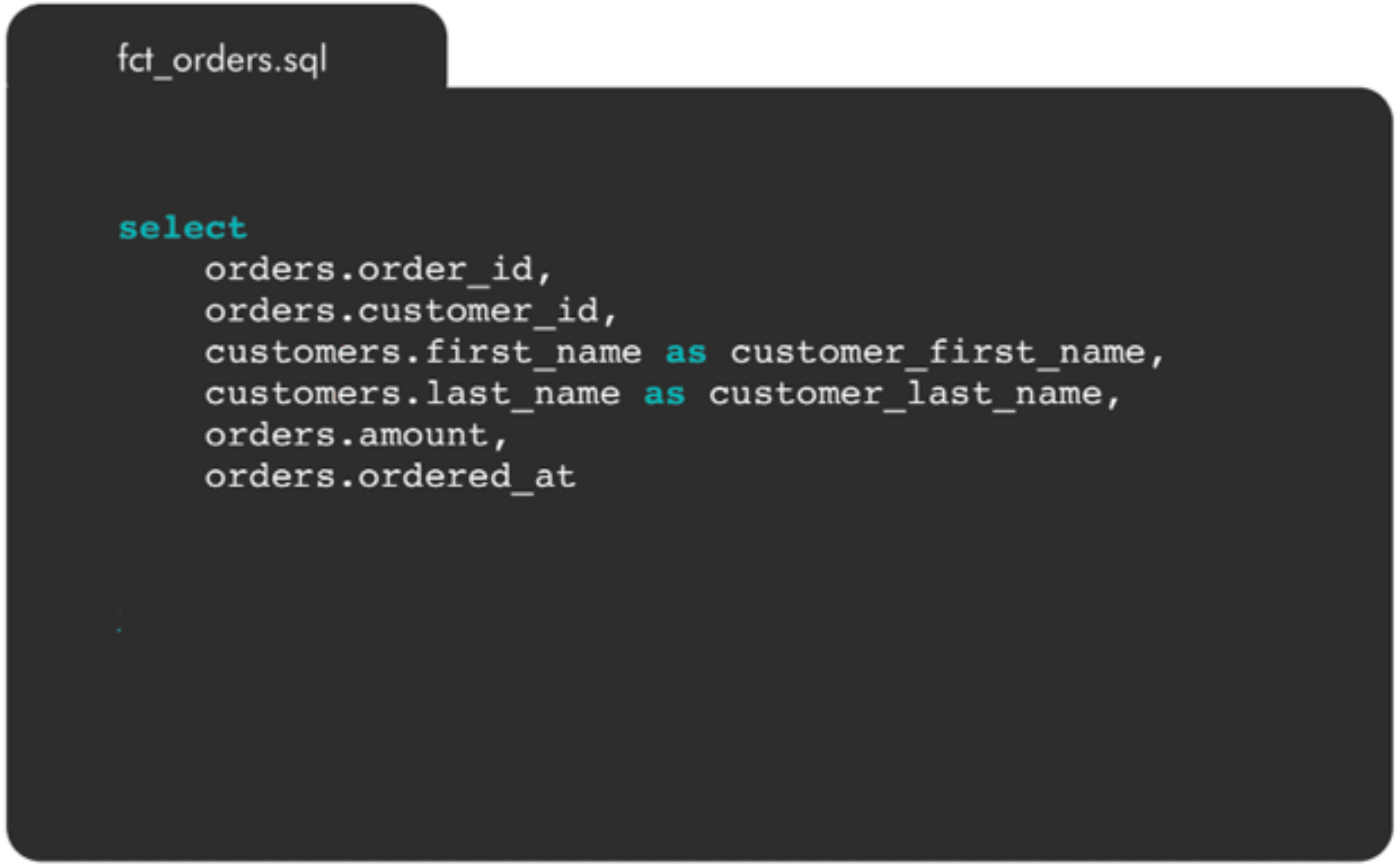


Transform

DBT - What?

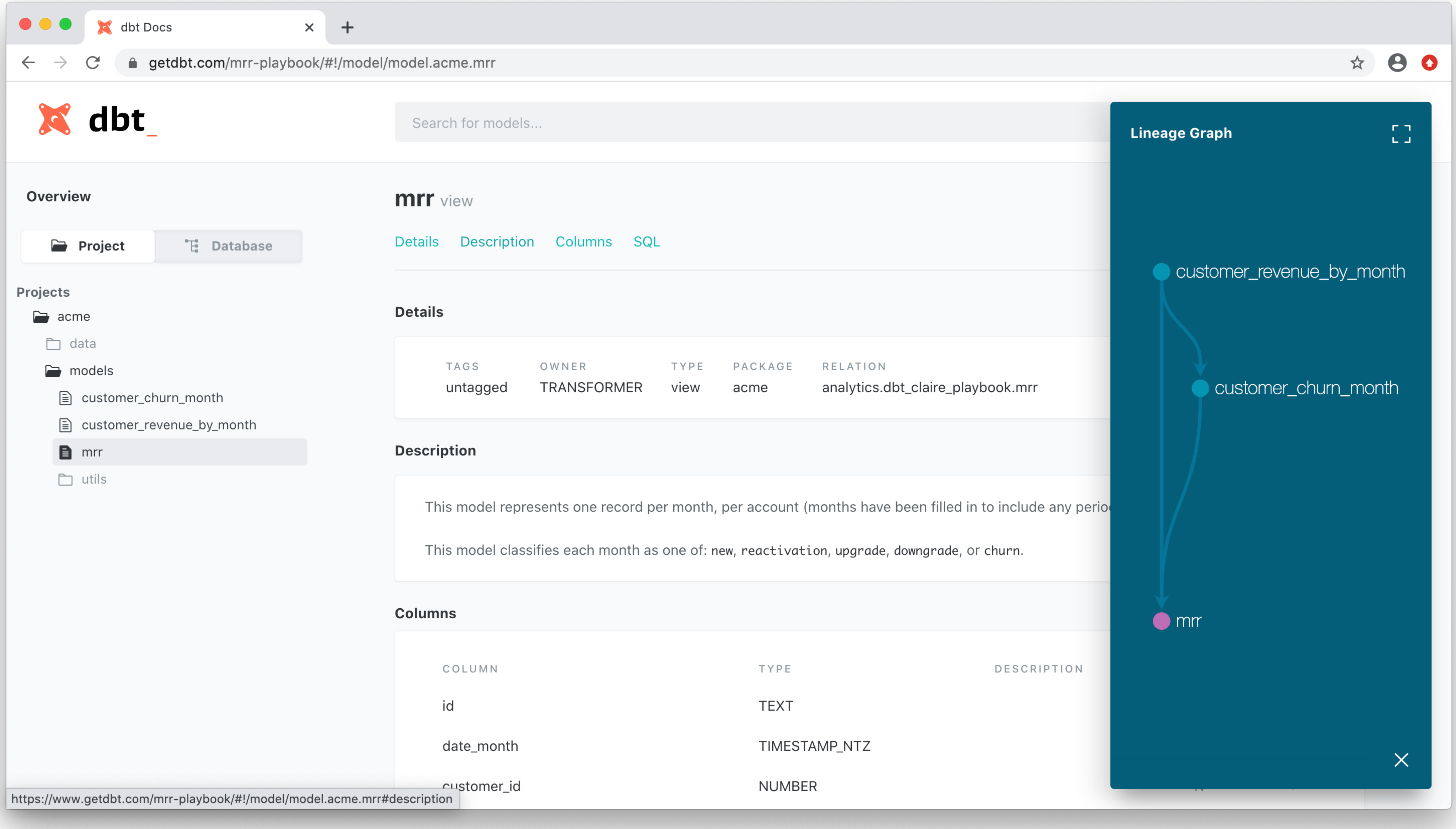

Data Loaders
Raw Data
Snapshot
Transform
Test
Deploy
Document
Version Control
Alerting
Logging
Transformed Data
BI Tools
Data Science
Data Warehouse





```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
```

Building dependencies between dbt models



dbt Docs

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

dbt_

Search for models...

Overview

Project Database

Projects

- acme
  - data
  - models
    - customer_churn_month
    - customer_revenue_by_month
    - mrr
  - utils

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 perio

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

customer_revenue_by_month

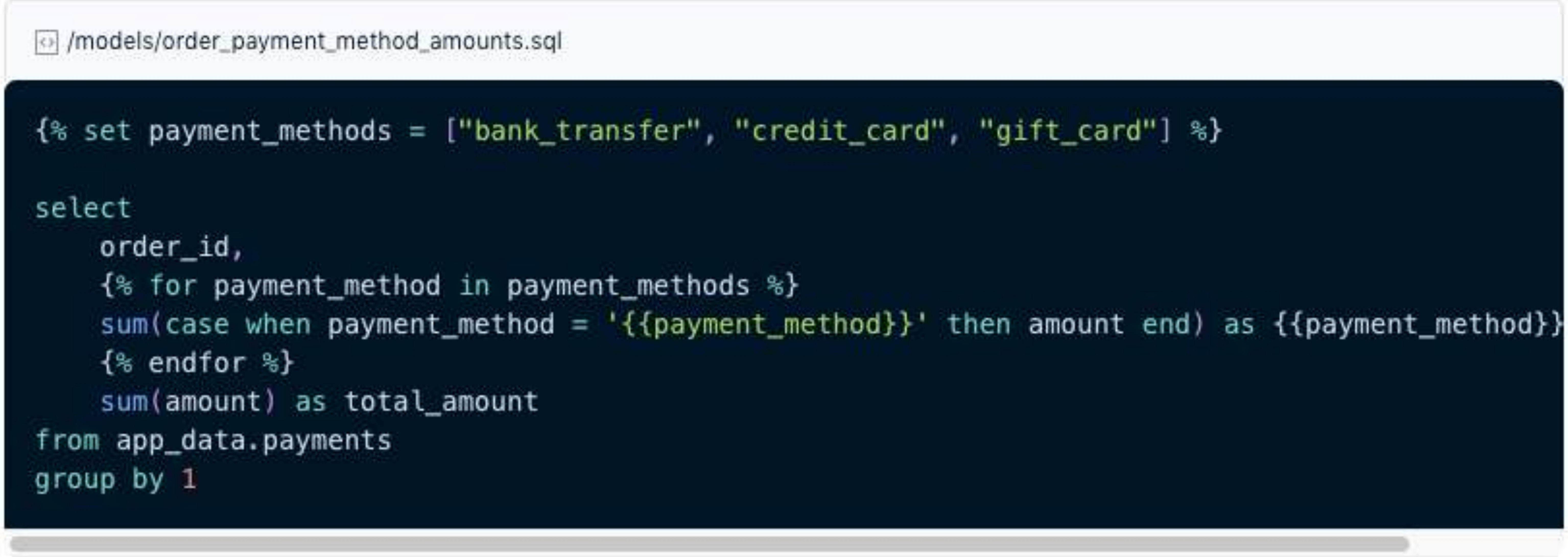
customer_churn_month

mrr

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

DBT - Why?

- Modular code



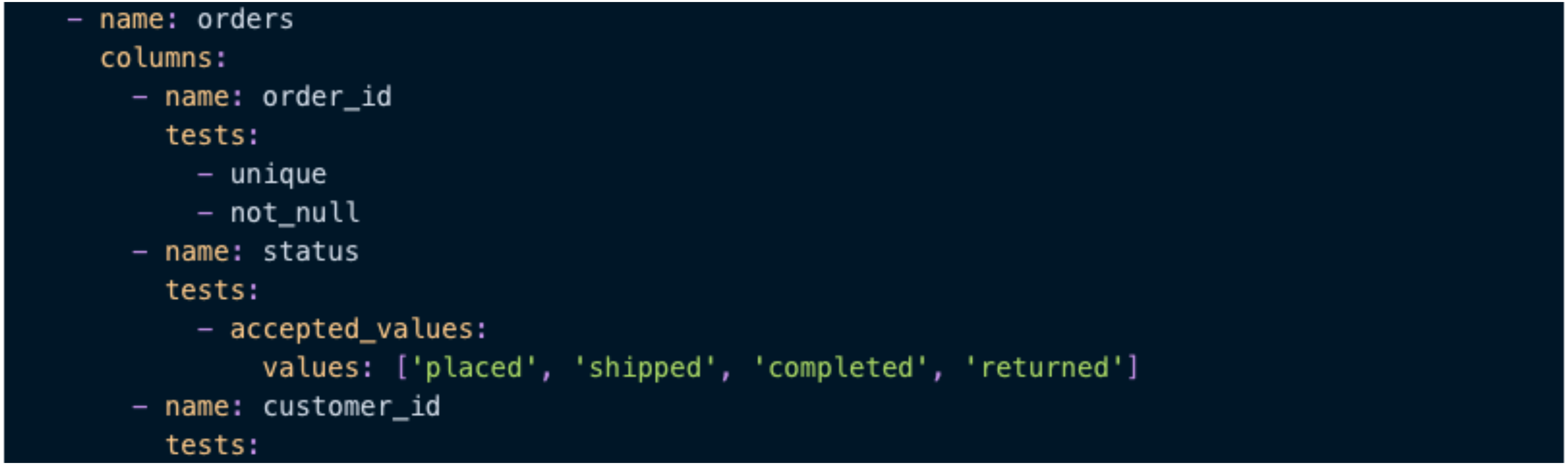
/models/order_payment_method_amounts.sql

```
{% 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?

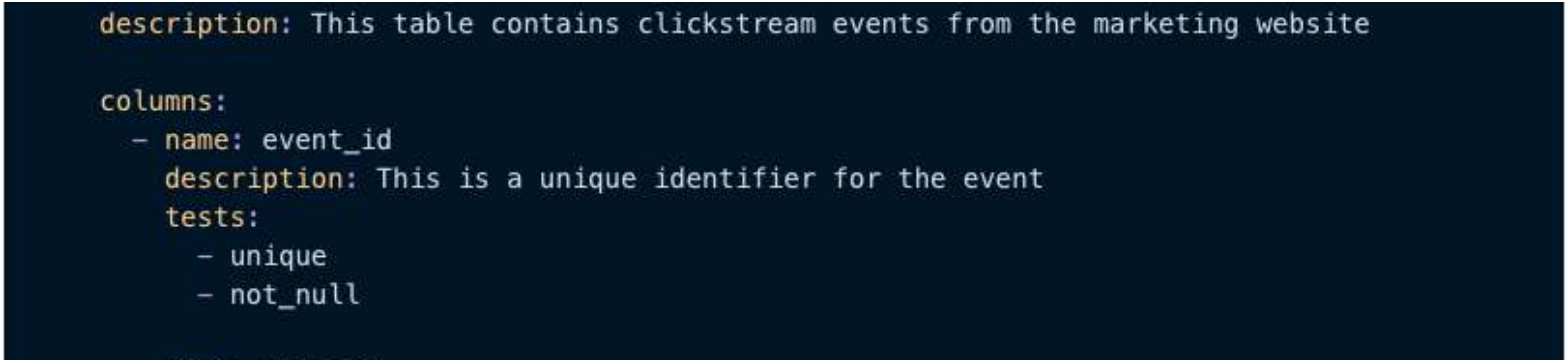
- Modular code
- Testing is 1st Class



```
  - 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?

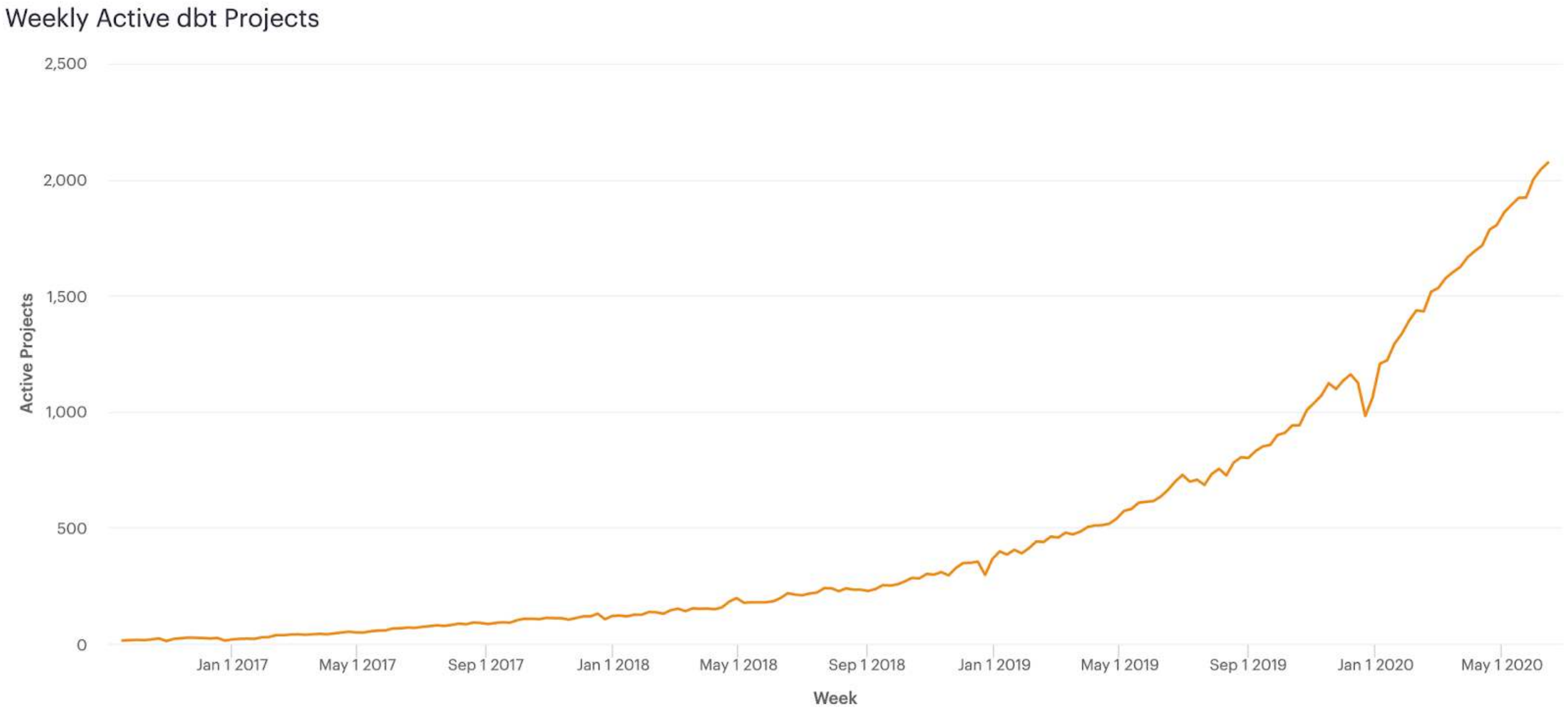
- Modular code
- Testing is 1st Class
- Data documentation is 1st Class



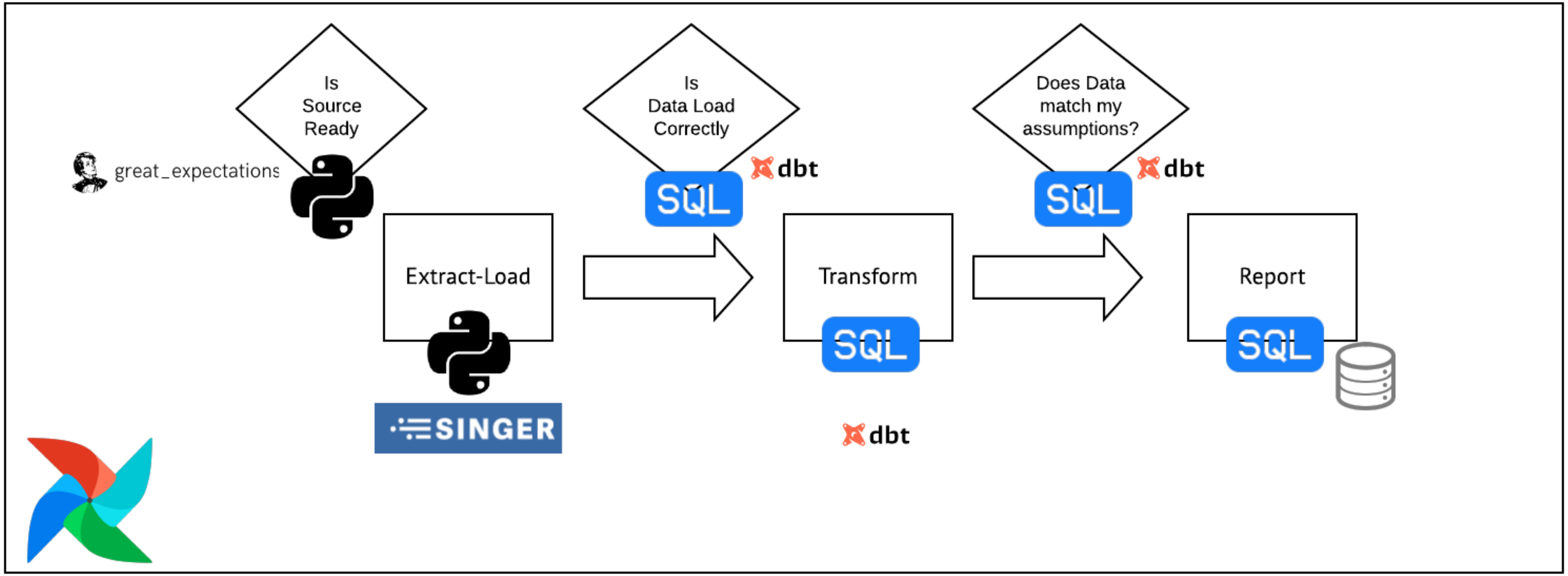
```
description: This table contains clickstream events from the marketing website

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

Great adoption



All together

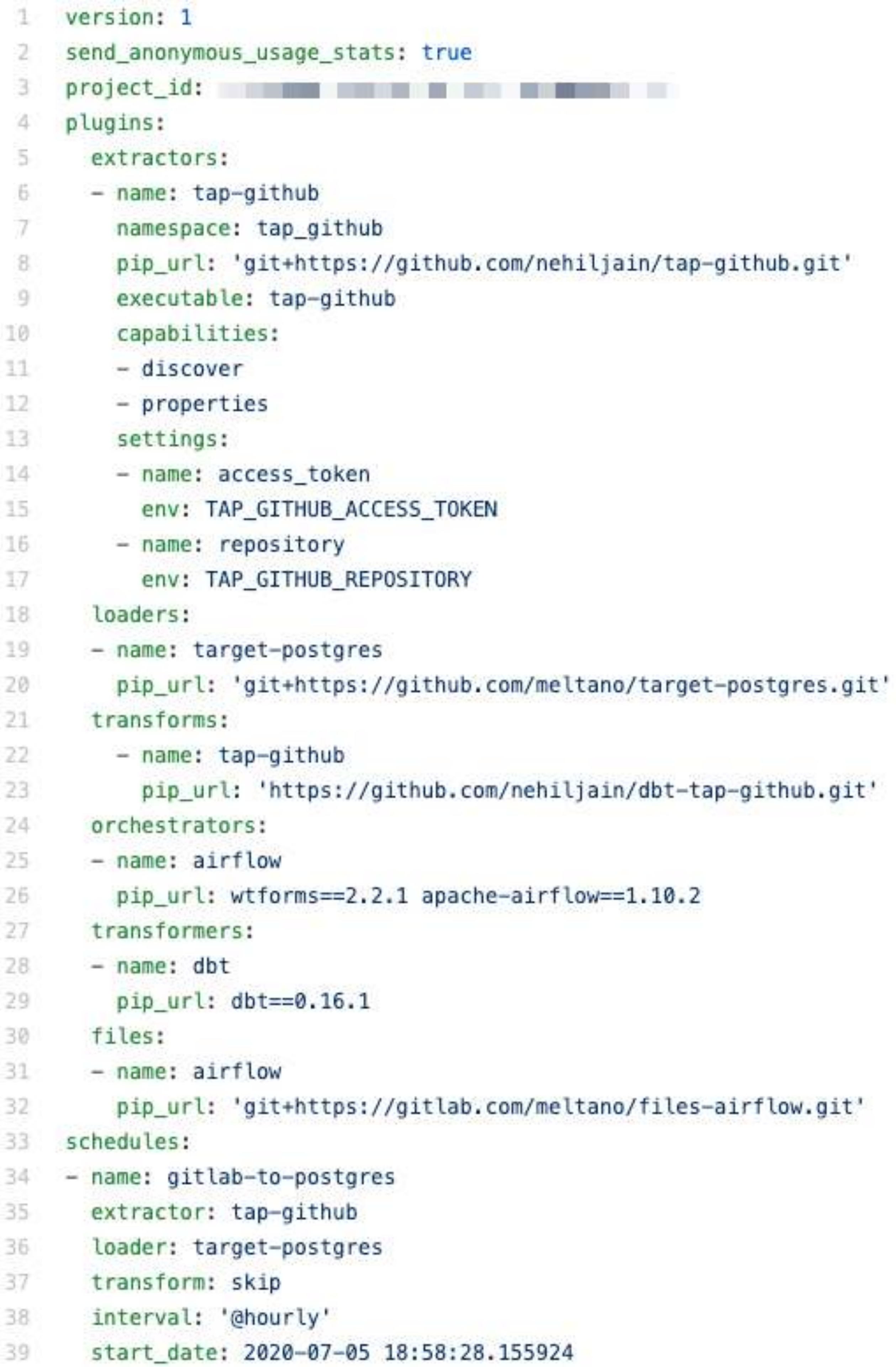


Meltano

- Open Source, GitLab
- Self Hosted

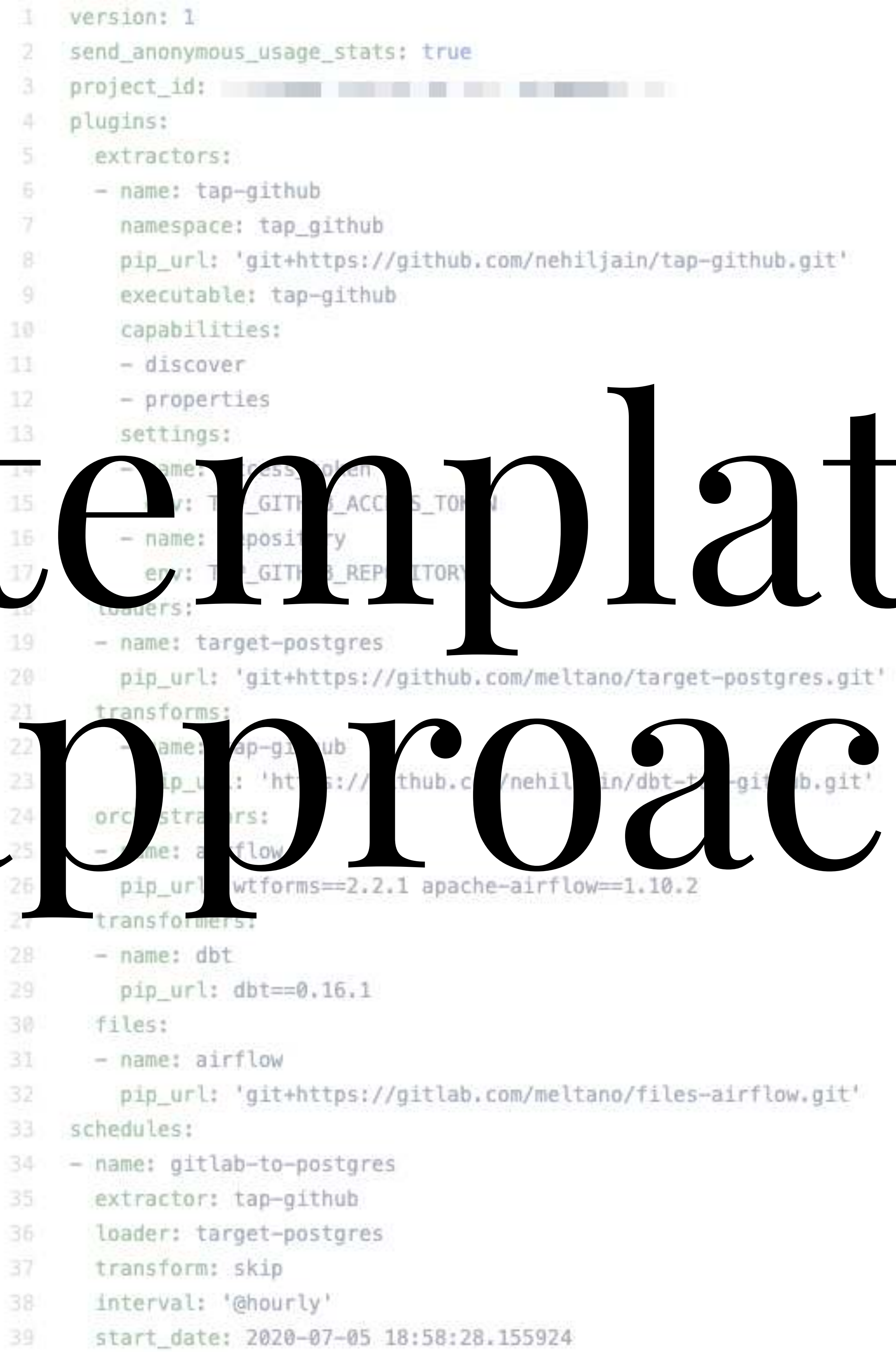
```
pip3 install meltano
meltano init airflow-analytics-project
meltano add extractor tap-github
meltano add loader target-postgres
meltano add transformer dbt
meltano add transform tap-github
# add env variables
meltano elt tap-gitlab target-postgres --transform=run --job_id=gitlab-to-postgres
meltano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

Let's look at the code

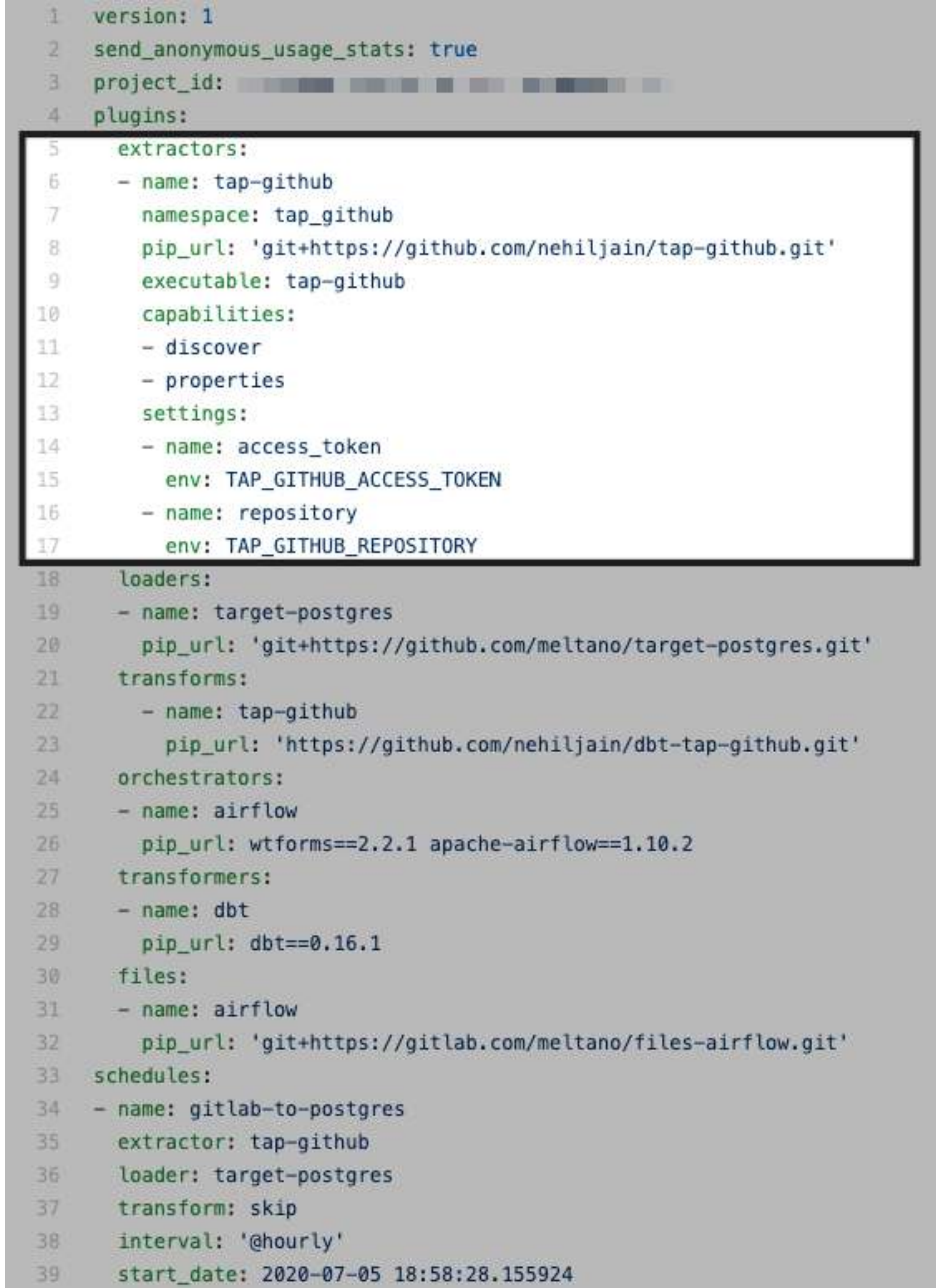


```yaml
version: 1
send_anonymous_usage_stats: true
project_id: [redacted]
plugins:
  extractors:
  - name: tap-github
    namespace: tap_github
    pip_url: 'git+https://github.com/nehiljain/tap-github.git'
    executable: tap-github
    capabilities:
    - discover
    - properties
    settings:
    - name: access_token
      env: TAP_GITHUB_ACCESS_TOKEN
    - name: repository
      env: TAP_GITHUB_REPOSITORY
  loaders:
  - name: target-postgres
    pip_url: 'git+https://github.com/meltano/target-postgres.git'
  transforms:
    - name: tap-github
      pip_url: 'https://github.com/nehiljain/dbt-tap-github.git'
  orchestrators:
  - name: airflow
    pip_url: wtforms==2.2.1 apache-airflow==1.10.2
  transformers:
  - name: dbt
    pip_url: dbt==0.16.1
  files:
  - name: airflow
    pip_url: 'git+https://gitlab.com/meltano/files-airflow.git'
schedules:
- name: gitlab-to-postgres
  extractor: tap-github
  loader: target-postgres
  transform: skip
  interval: '@hourly'
  start_date: 2020-07-05 18:58:28.155924
```

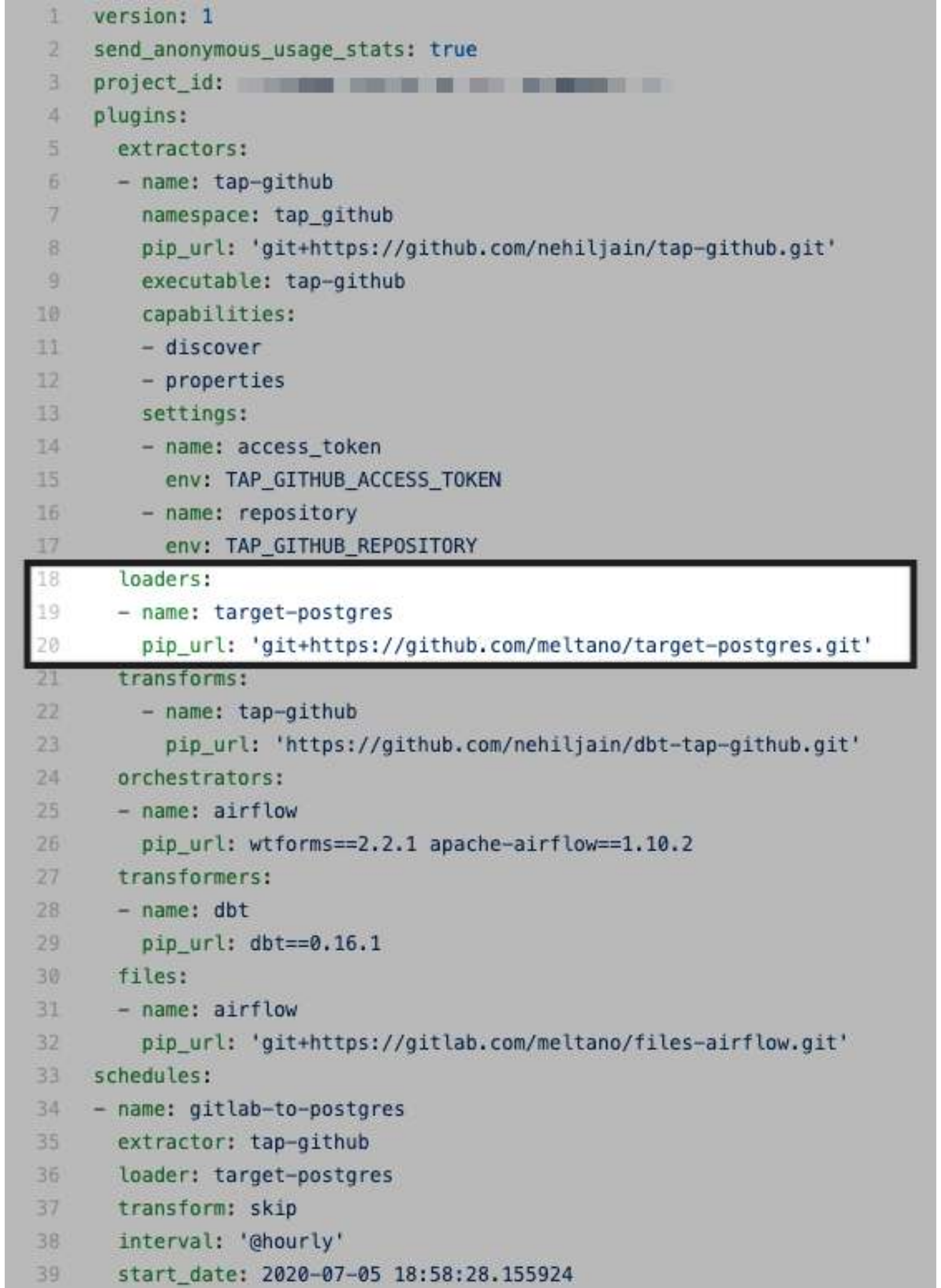
A templated approach



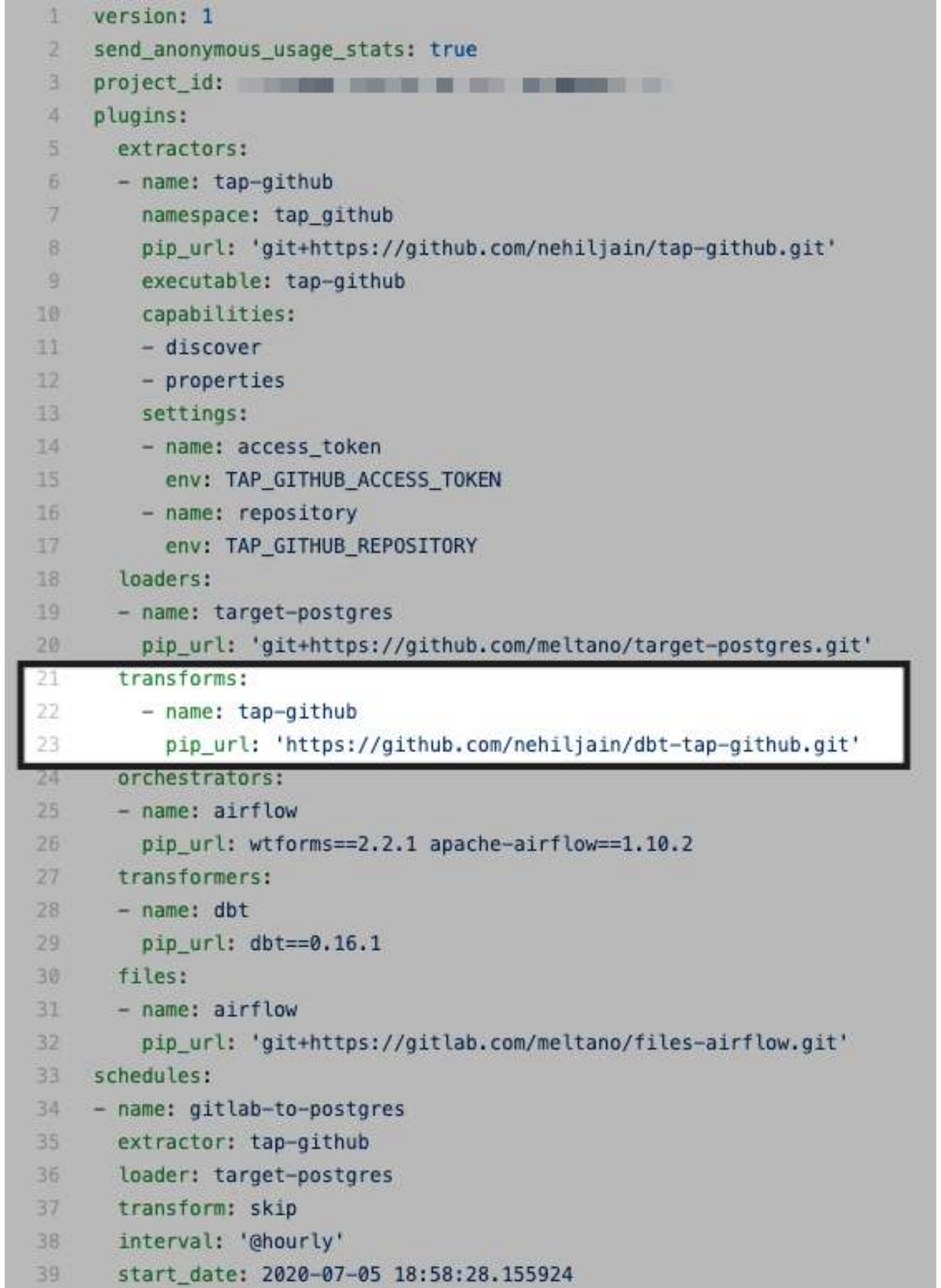
```
version: 1
send_anonymous_usage_stats: true
project_id: [redacted]
plugins:
  extractors:
  - name: tap-github
    namespace: tap_github
    pip_url: 'git+https://github.com/nehiljain/tap-github.git'
    executable: tap-github
    capabilities:
    - discover
    - properties
    settings:
    - name: access_token
      env: TAP_GITHUB_ACCESS_TOKEN
    - name: repository
      env: TAP_GITHUB_REPOSITORY
  loaders:
  - name: target-postgres
    pip_url: 'git+https://github.com/meltano/target-postgres.git'
  transforms:
  - name: tap-github
    pip_url: 'https://github.com/nehiljain/dbt-tap-github.git'
  orchestrators:
  - name: airflow
    pip_url: wtforms==2.2.1 apache-airflow==1.10.2
  transformers:
  - name: dbt
    pip_url: dbt==0.16.1
  files:
  - name: airflow
    pip_url: 'git+https://gitlab.com/meltano/files-airflow.git'
schedules:
- name: gitlab-to-postgres
  extractor: tap-github
  loader: target-postgres
  transform: skip
  interval: '@hourly'
  start_date: 2020-07-05 18:58:28.155924
```



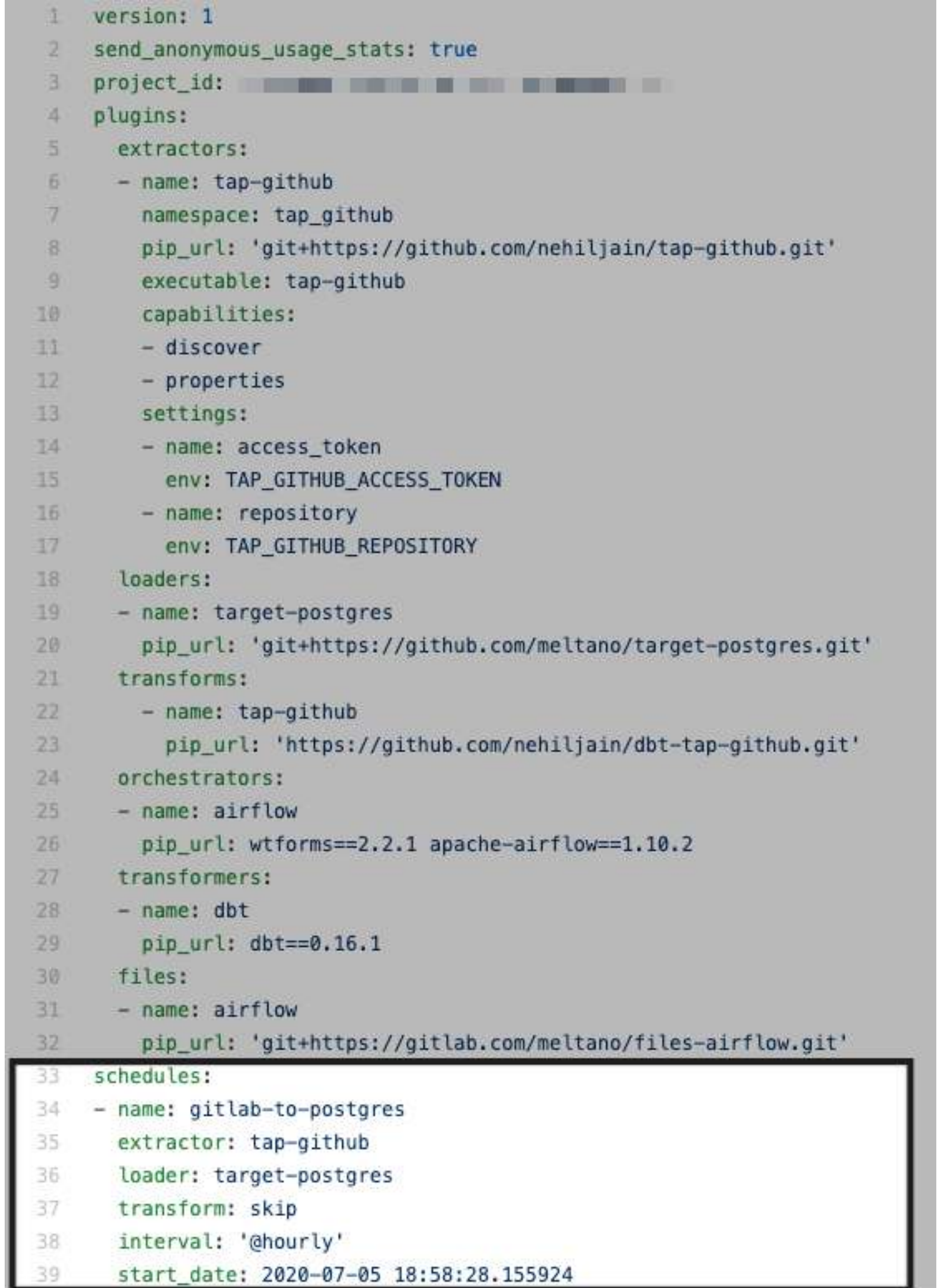
```
version: 1
send_anonymous_usage_stats: true
project_id: [redacted]
plugins:
  extractors:
  - name: tap-github
    namespace: tap_github
    pip_url: 'git+https://github.com/nehiljain/tap-github.git'
    executable: tap-github
    capabilities:
    - discover
    - properties
    settings:
    - name: access_token
      env: TAP_GITHUB_ACCESS_TOKEN
    - name: repository
      env: TAP_GITHUB_REPOSITORY
  loaders:
  - name: target-postgres
    pip_url: 'git+https://github.com/meltano/target-postgres.git'
  transforms:
    - name: tap-github
      pip_url: 'https://github.com/nehiljain/dbt-tap-github.git'
  orchestrators:
  - name: airflow
    pip_url: wtforms==2.2.1 apache-airflow==1.10.2
  transformers:
  - name: dbt
    pip_url: dbt==0.16.1
  files:
  - name: airflow
    pip_url: 'git+https://gitlab.com/meltano/files-airflow.git'
schedules:
- name: gitlab-to-postgres
  extractor: tap-github
  loader: target-postgres
  transform: skip
  interval: '@hourly'
  start_date: 2020-07-05 18:58:28.155924
```



```yaml
version: 1
send_anonymous_usage_stats: true
project_id: [redacted]
plugins:
  extractors:
  - name: tap-github
    namespace: tap_github
    pip_url: 'git+https://github.com/nehiljain/tap-github.git'
    executable: tap-github
    capabilities:
    - discover
    - properties
    settings:
    - name: access_token
      env: TAP_GITHUB_ACCESS_TOKEN
    - name: repository
      env: TAP_GITHUB_REPOSITORY
  loaders:
  - name: target-postgres
    pip_url: 'git+https://github.com/meltano/target-postgres.git'
  transforms:
    - name: tap-github
      pip_url: 'https://github.com/nehiljain/dbt-tap-github.git'
  orchestrators:
  - name: airflow
    pip_url: wtforms==2.2.1 apache-airflow==1.10.2
  transformers:
  - name: dbt
    pip_url: dbt==0.16.1
  files:
  - name: airflow
    pip_url: 'git+https://gitlab.com/meltano/files-airflow.git'
schedules:
- name: gitlab-to-postgres
  extractor: tap-github
  loader: target-postgres
  transform: skip
  interval: '@hourly'
  start_date: 2020-07-05 18:58:28.155924
```



```yaml
version: 1
send_anonymous_usage_stats: true
project_id: [redacted]
plugins:
  extractors:
  - name: tap-github
    namespace: tap_github
    pip_url: 'git+https://github.com/nehiljain/tap-github.git'
    executable: tap-github
    capabilities:
    - discover
    - properties
    settings:
    - name: access_token
      env: TAP_GITHUB_ACCESS_TOKEN
    - name: repository
      env: TAP_GITHUB_REPOSITORY
  loaders:
  - name: target-postgres
    pip_url: 'git+https://github.com/meltano/target-postgres.git'
  transforms:
    - name: tap-github
      pip_url: 'https://github.com/nehiljain/dbt-tap-github.git'
  orchestrators:
  - name: airflow
    pip_url: wtforms==2.2.1 apache-airflow==1.10.2
  transformers:
  - name: dbt
    pip_url: dbt==0.16.1
  files:
  - name: airflow
    pip_url: 'git+https://gitlab.com/meltano/files-airflow.git'
schedules:
- name: gitlab-to-postgres
  extractor: tap-github
  loader: target-postgres
  transform: skip
  interval: '@hourly'
  start_date: 2020-07-05 18:58:28.155924
```



```yaml
version: 1
send_anonymous_usage_stats: true
project_id: [redacted]
plugins:
  extractors:
  - name: tap-github
    namespace: tap_github
    pip_url: 'git+https://github.com/nehiljain/tap-github.git'
    executable: tap-github
    capabilities:
    - discover
    - properties
    settings:
    - name: access_token
      env: TAP_GITHUB_ACCESS_TOKEN
    - name: repository
      env: TAP_GITHUB_REPOSITORY
  loaders:
  - name: target-postgres
    pip_url: 'git+https://github.com/meltano/target-postgres.git'
  transforms:
    - name: tap-github
      pip_url: 'https://github.com/nehiljain/dbt-tap-github.git'
  orchestrators:
  - name: airflow
    pip_url: wtforms==2.2.1 apache-airflow==1.10.2
  transformers:
  - name: dbt
    pip_url: dbt==0.16.1
  files:
  - name: airflow
    pip_url: 'git+https://gitlab.com/meltano/files-airflow.git'
schedules:
- name: gitlab-to-postgres
  extractor: tap-github
  loader: target-postgres
  transform: skip
  interval: '@hourly'
  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