Building Reusable and Trustworthy pipelines

1 – Airflow Summit 2020,

Outline

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

Context

Hello 👋!

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

Purpose

Share 🧰 BI pipelines

💡 Community with lessons learnt

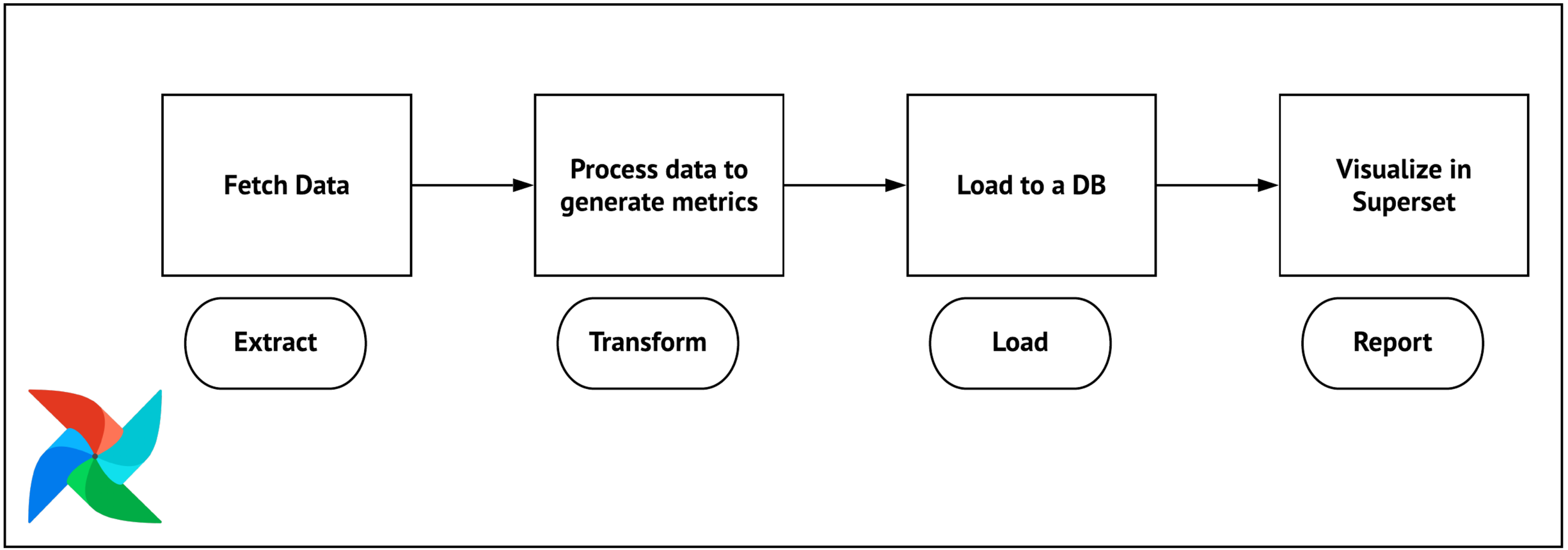
👂 feedback

How are my 📈?

- `gross_revenue`
- `contribution_margin`
- `number_of_active_users`
- `retention_rate`
- `conversion_rate`

What part of Airflow repo needs my attention?

- `number_prs_merged`
- `number_prs_closed_without_merge`
- `number_prs_opened`
- `number_of_commit`


Fetch Data
Process data to generate metrics
Load to a DB
Visualize in Superset
Extract
Transform
Load
Report

Let us consider

- The pipeline failed

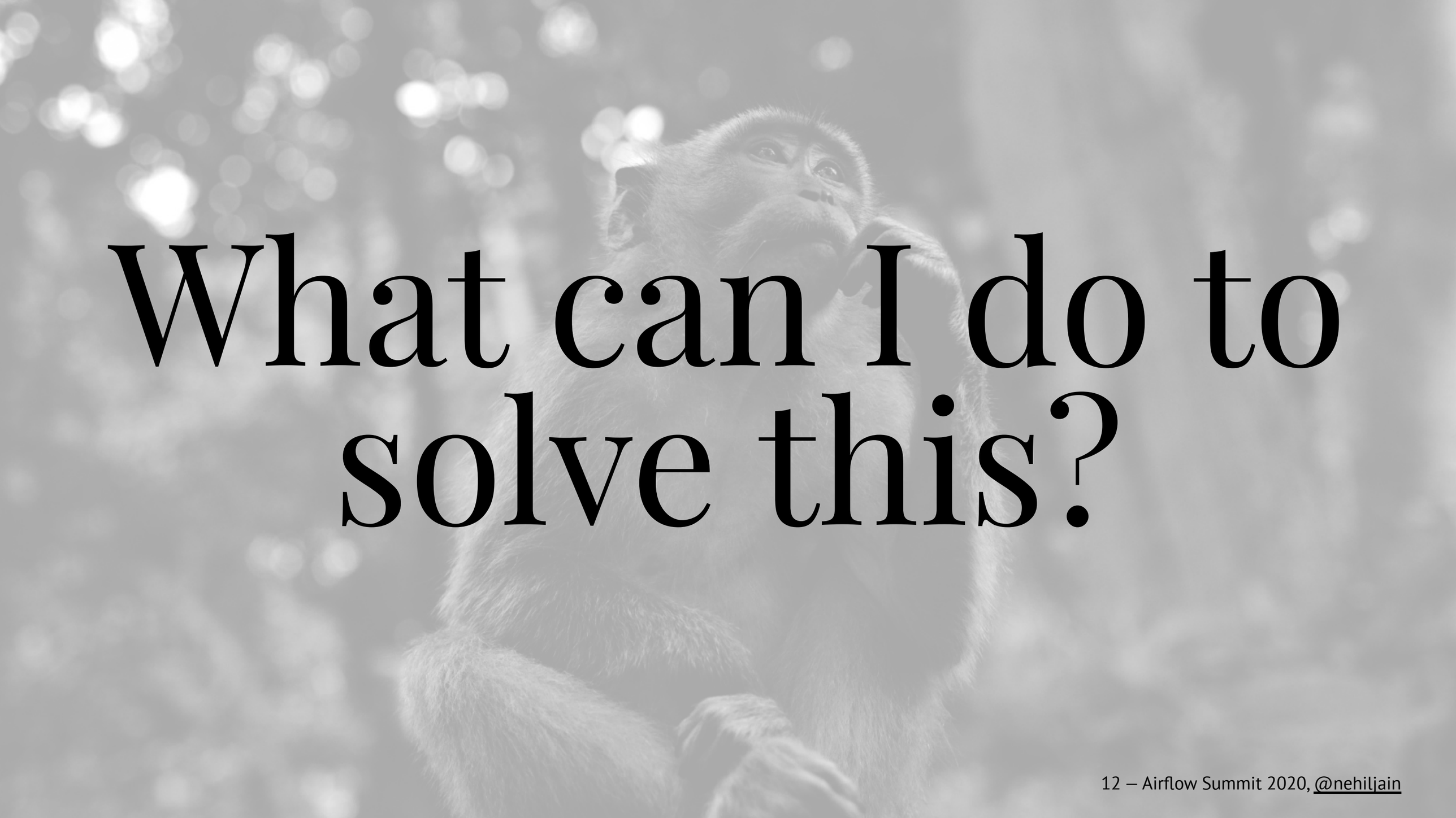
after a few days of productionization

- Now I want to focus on issues
- Gitlab released a new version of API
- I want to analyze other apache projects too

Been there felt that?

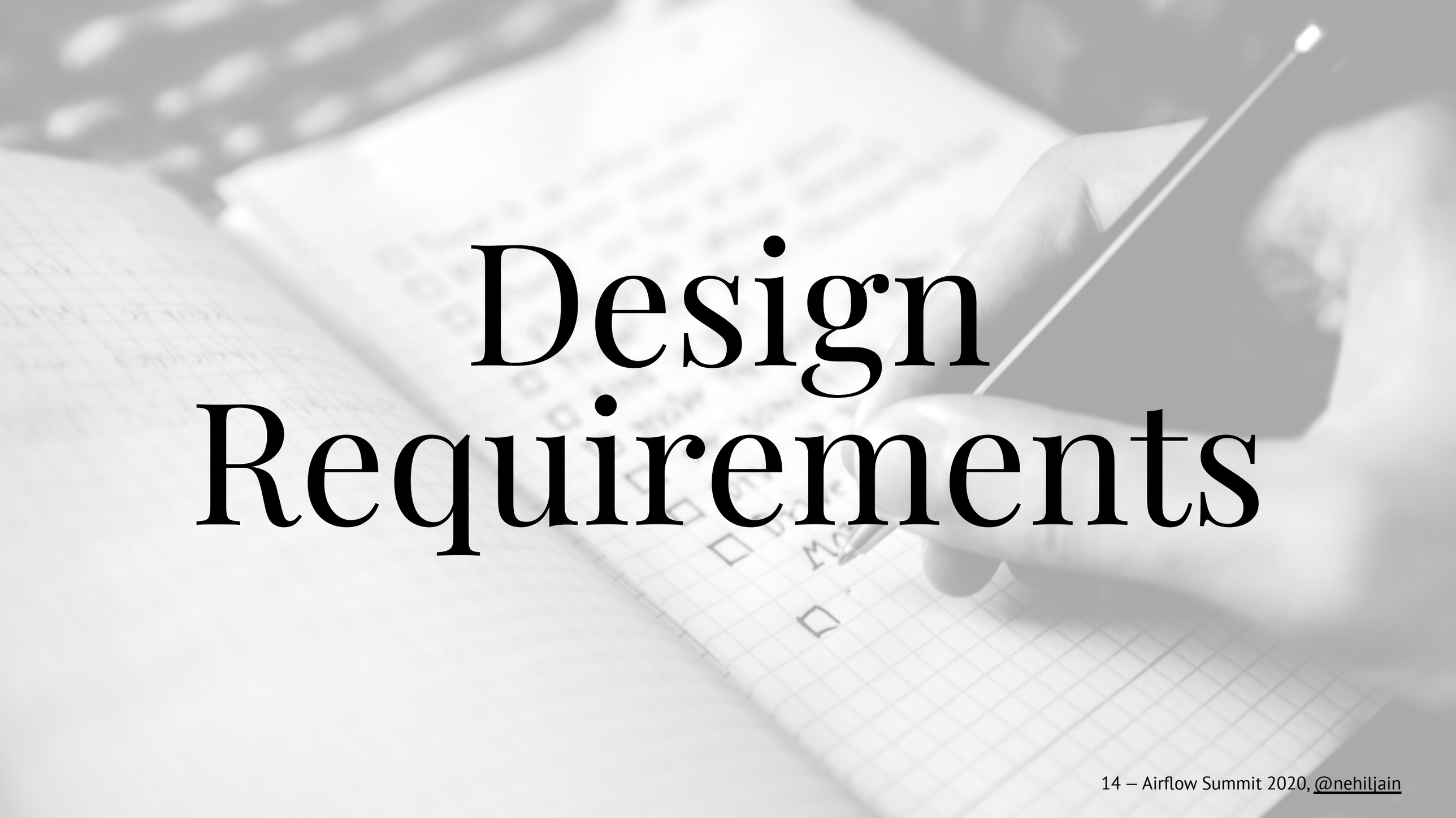
🙋‍♀️ Been there felt that? 🙋‍♂️

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



What can I do to solve this?

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



Design Requirements


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

Single Source of Truth

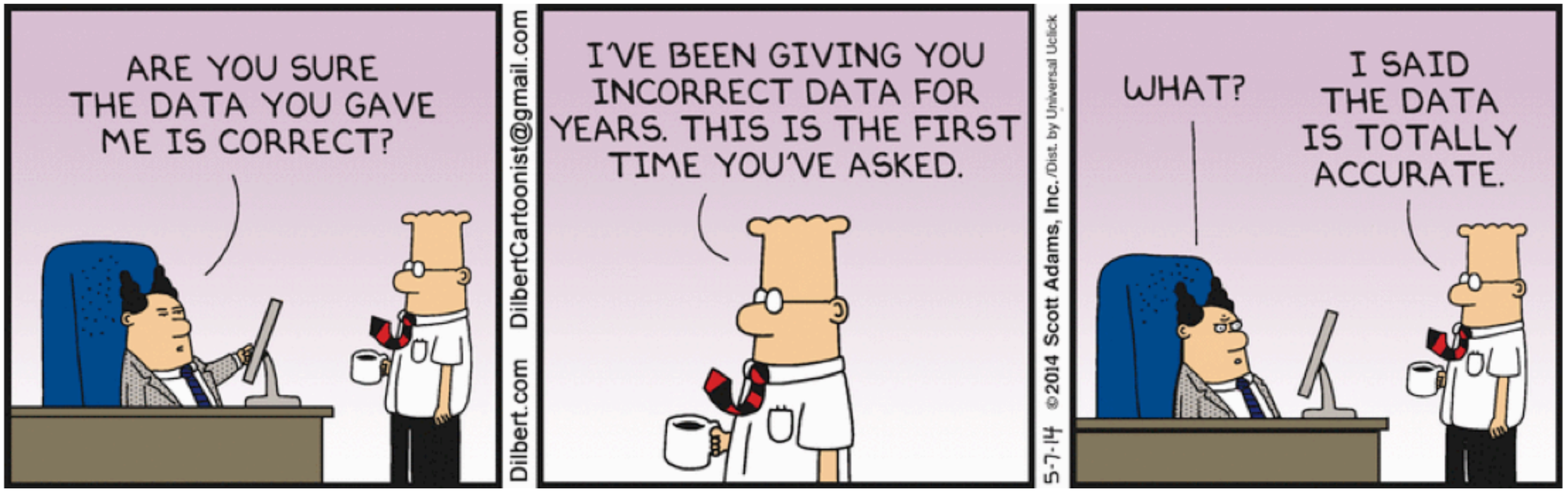
- Standardization
- Data Lineage
- Empower non-technical folks

Easy to consume

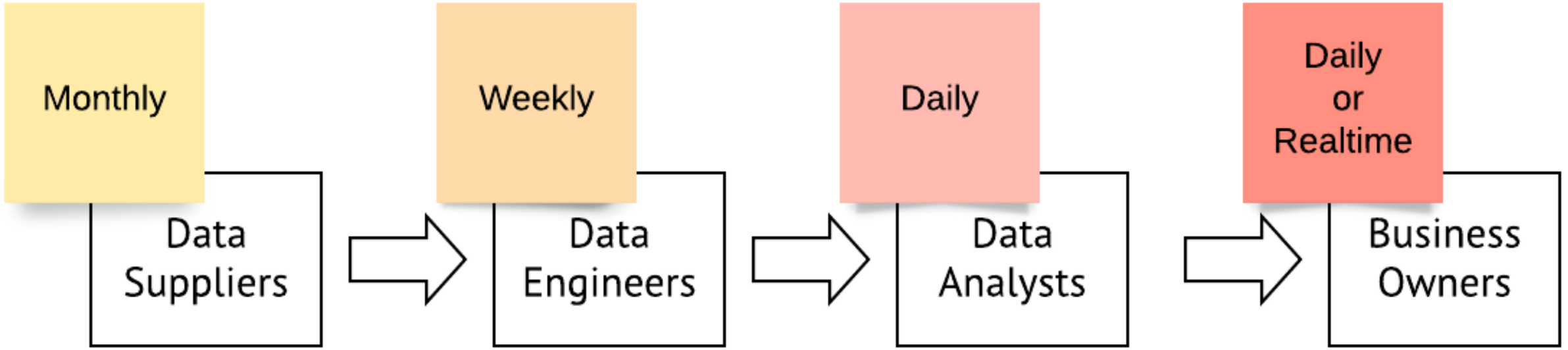
- Airflow + Other OSS
- Ideally `pip install awesome-elt-tool`
- Low barrier to entry for data analytics
- Operational creep

Promote data integrity

- Test the raw data supply
- Automated analytics testing

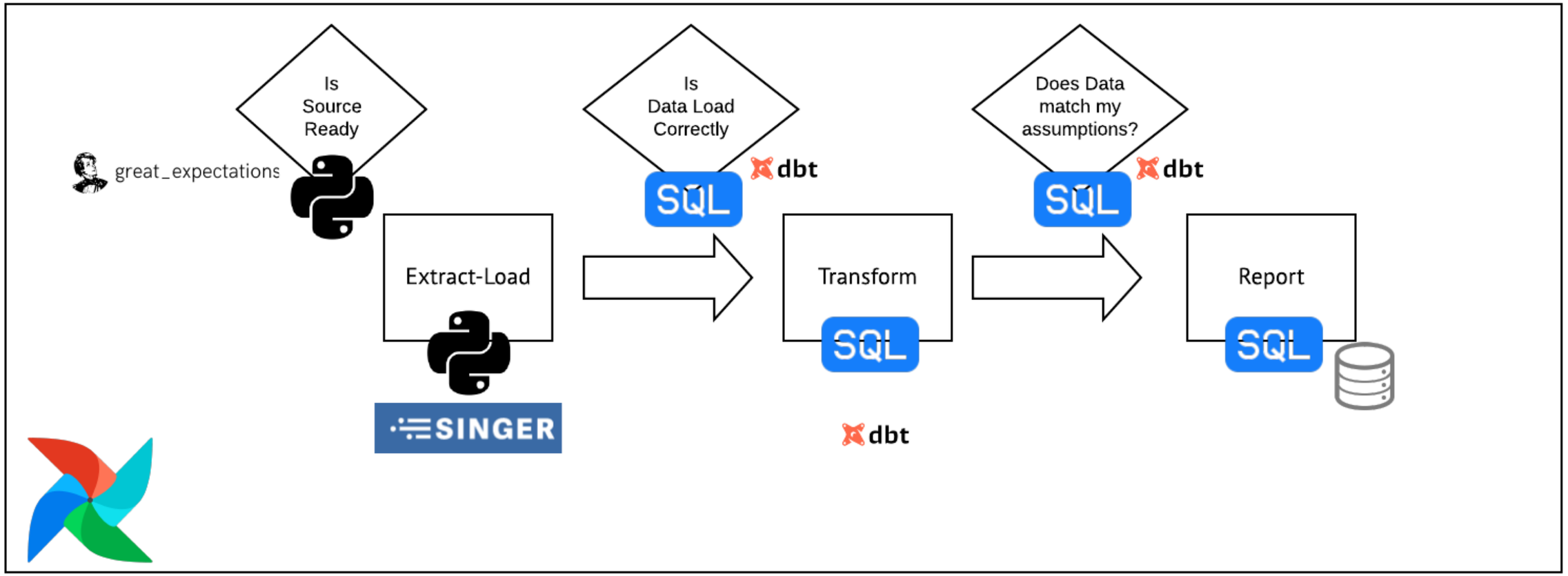


Meta Data Engineering


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

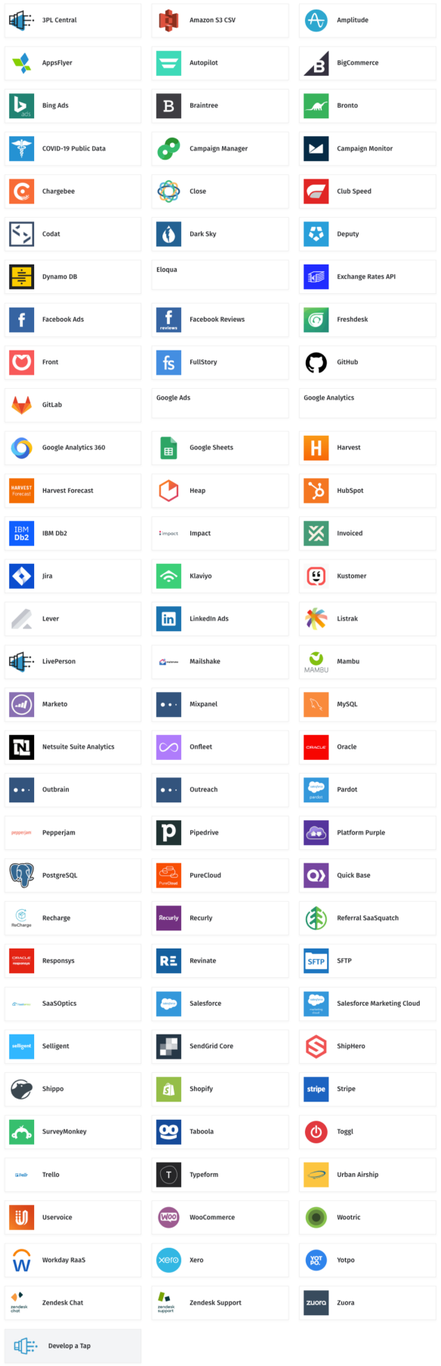
Singer - What?

Singer - Why?

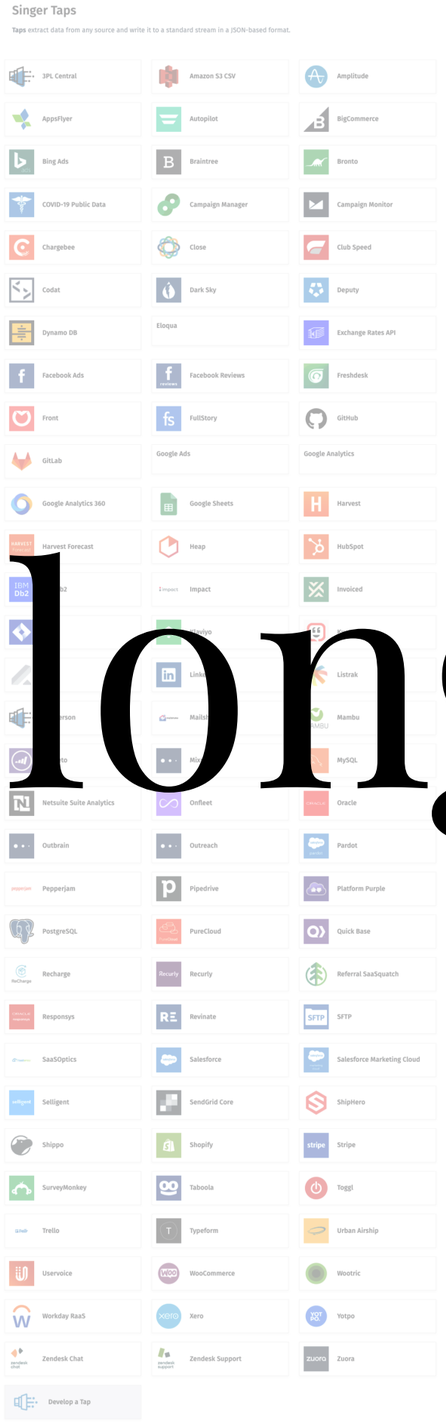
- Standardized communication
- Incremental out of the box
- Documentation
- See your data in under 10 mins

Singer Taps

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

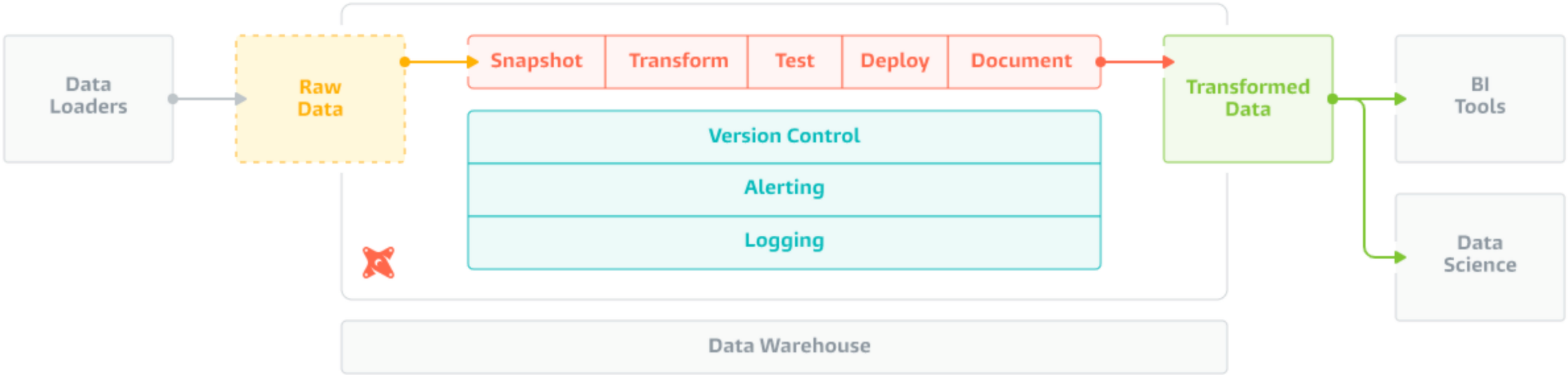


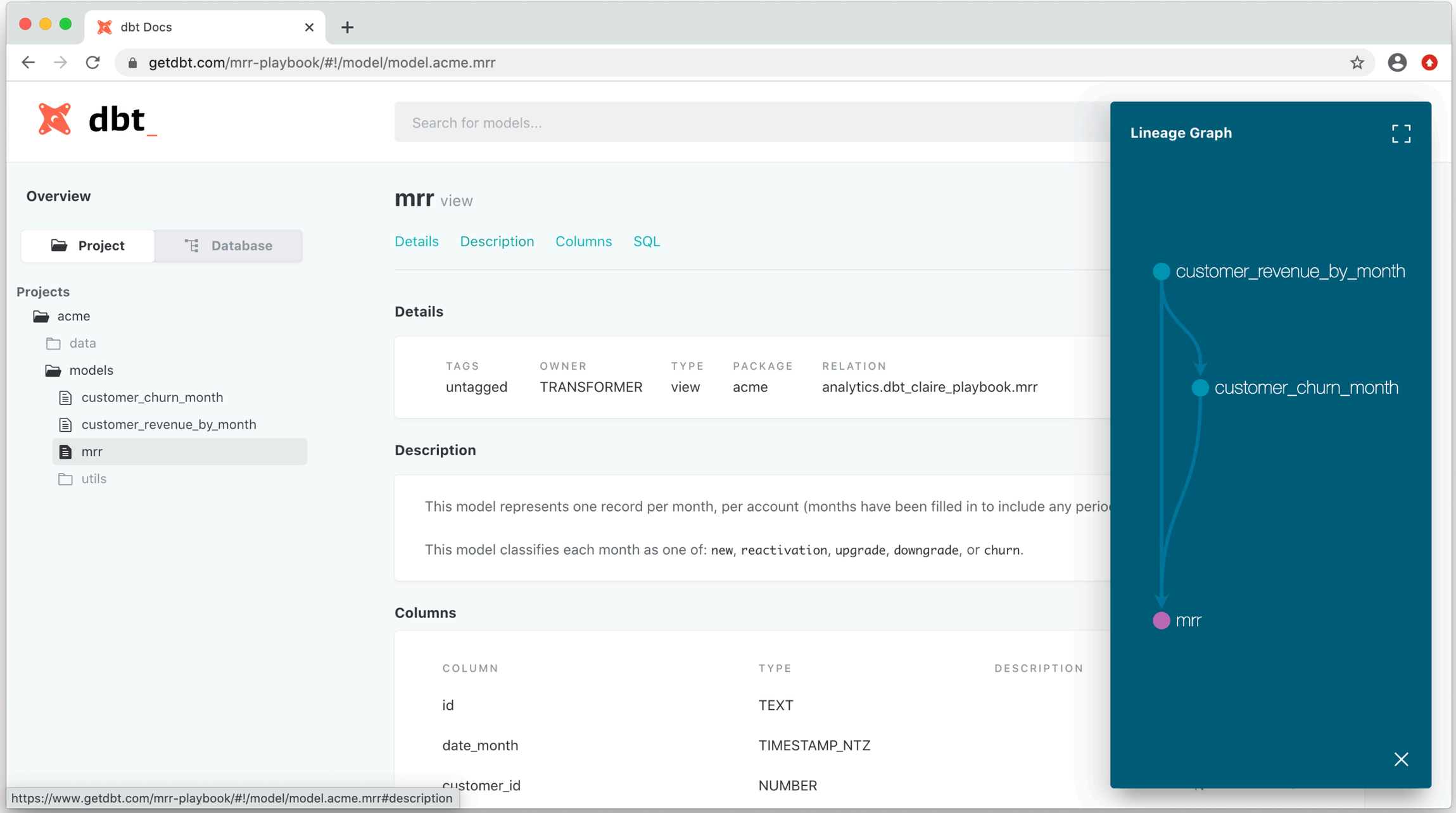
Its a long list



Transform

DBT - What?


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


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
untagged
OWNER
TRANSFORMER
TYPE
view
PACKAGE
acme
RELATION
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?

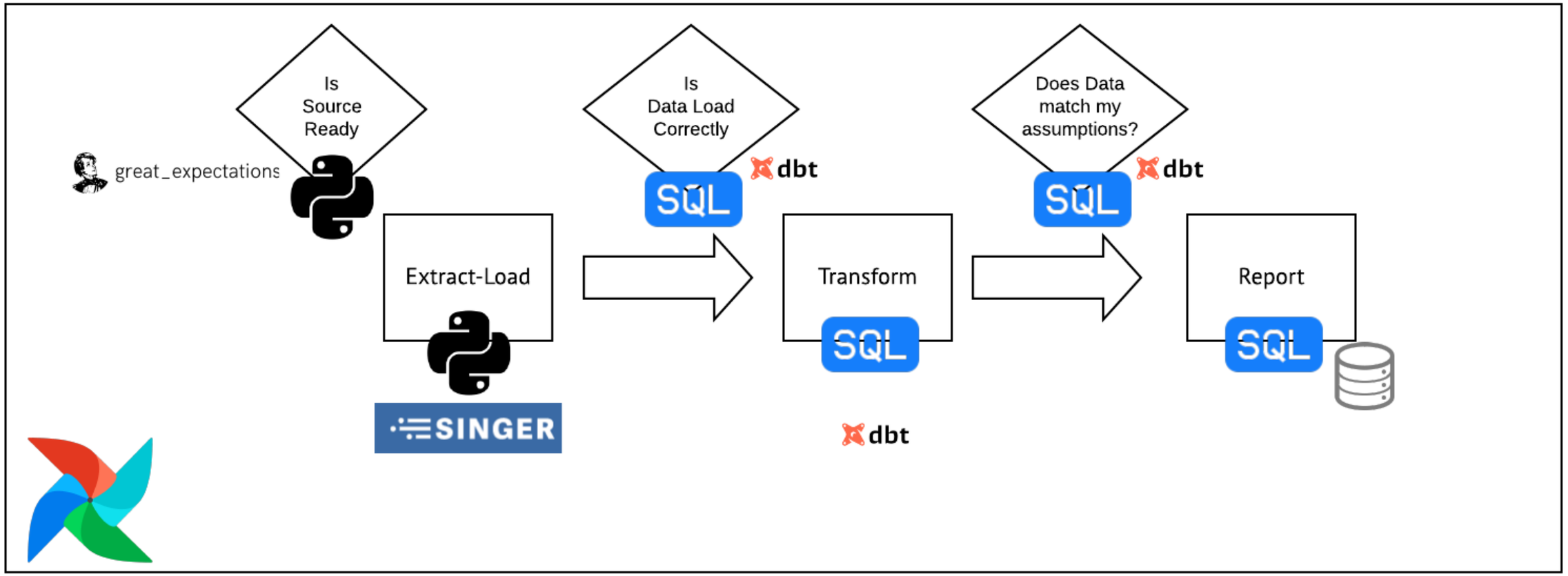
- Modular code
- Testing is 1st Class
- Data documentation is 1st Class

```yaml
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:
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:
      env:
    - name:
      env:
  loaders:
  - name: target-postgres
    pip_url: 'git+https://github.com/meltano/target-postgres.git'
  transforms:
  - name:
    pip_u
  orchestrators:
  - name:
    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
```

```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
```

Some challenges out there

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

Key Takeaways

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

Resources

1. The Rise of the Data Engineer
2. The Future of Data Engineering
3. Downfall of the data engineer
4. Supercharging your ETL with Airflow and Singer
5. Singer | Open Source ETL
6. Why we are building an open-source platform for ELT