







Building Reusable and Trustworthy pipelines

Outline

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

Context

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

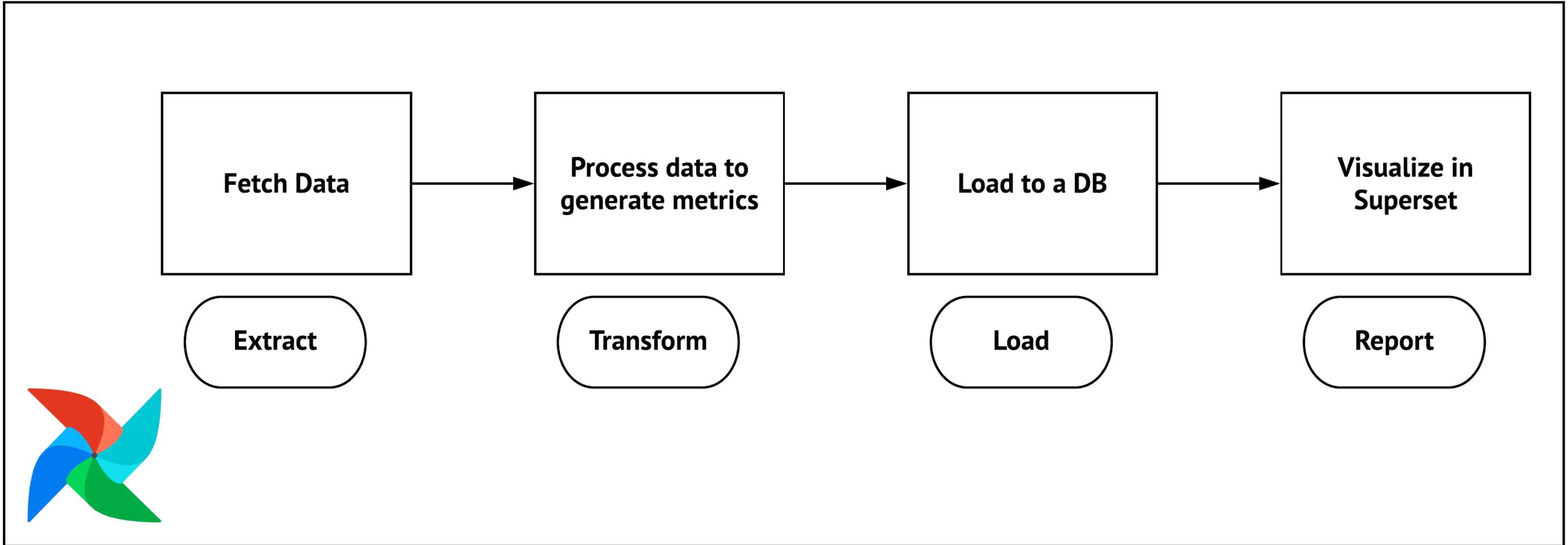
 feedback

How are my company ?

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

How's my airflow repo ?

- ▶ `number_prs_merged`
- ▶ `number_prs_closed_without_merge`
- ▶ `number_prs_opened`
- ▶ `number_of_commits`



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

A grayscale photograph of a monkey holding a smartphone in its mouth, looking at the screen. The monkey is positioned in the center of the frame, with its body and arms visible. The background is a blurred, bokeh-style image of light spots, possibly representing a forest or a bright outdoor setting. The text "What can we do to solve this?" is overlaid on the image in a large, black, serif font.

What can we do to solve this?

..build tools, infrastructure, frameworks and services

– Maxime Beauchemin





Design Requirements

amazon Prime
amazon.com/sameday
FREE SAME-DAY DELIVERY

amazon Prime
amazon.com/sameday
FREE SAME-DAY DELIVERY

amazon Prime
amazon.com/sameday
FREE SAME-DAY DELIVERY

amazon Prime
amazon.com/sameday
FREE SAME-DAY DELIVERY

amazon Prime
amazon.com/sameday
FREE SAME-DAY DELIVERY

amazon Prime
amazon.com/sameday
FREE SAME-DAY DELIVERY

amazon Prime
amazon.com/sameday
FREE SAME-DAY DELIVERY

NO PATIENCE REQUIRED

FREE SAME-DAY DELIVERY

**★ NEW IN ★
YOUR AREA**

Single Source of Truth

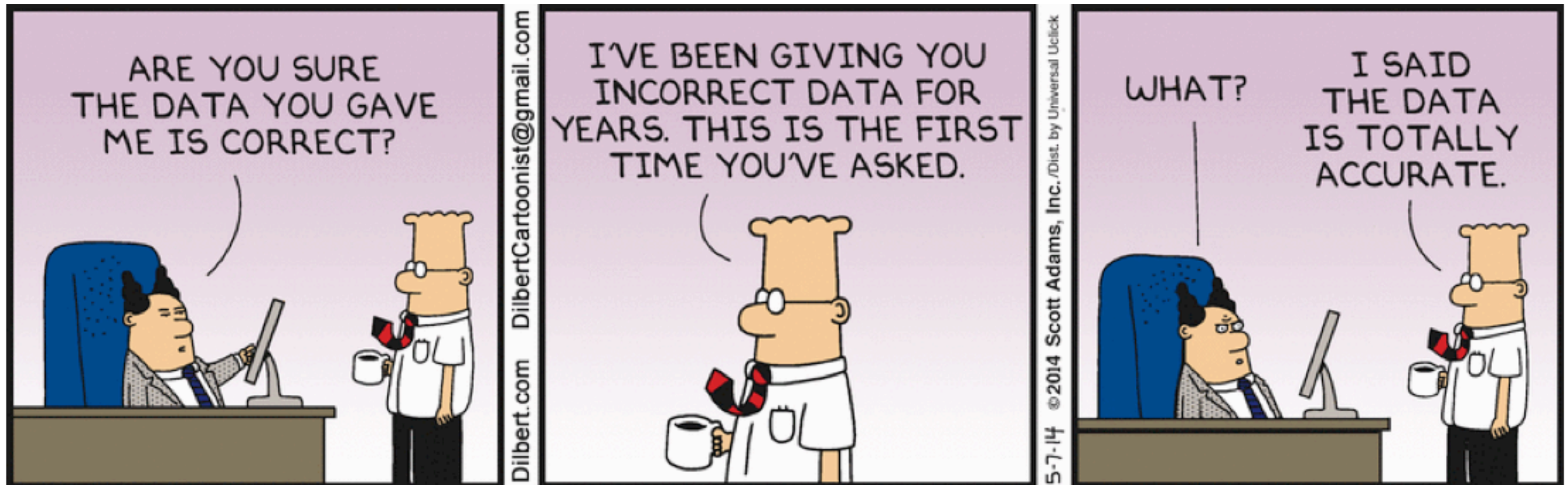
- ▶ Standardization
- ▶ Data Lineage
- ▶ Empower non-technical folks

Easy to consume

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

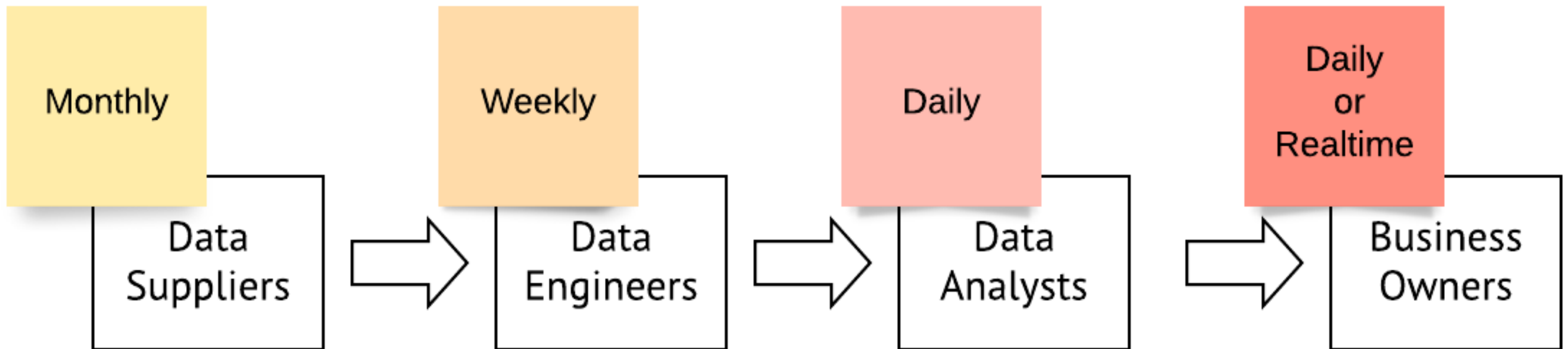
Promote data integrity


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



Meta Data Engineering

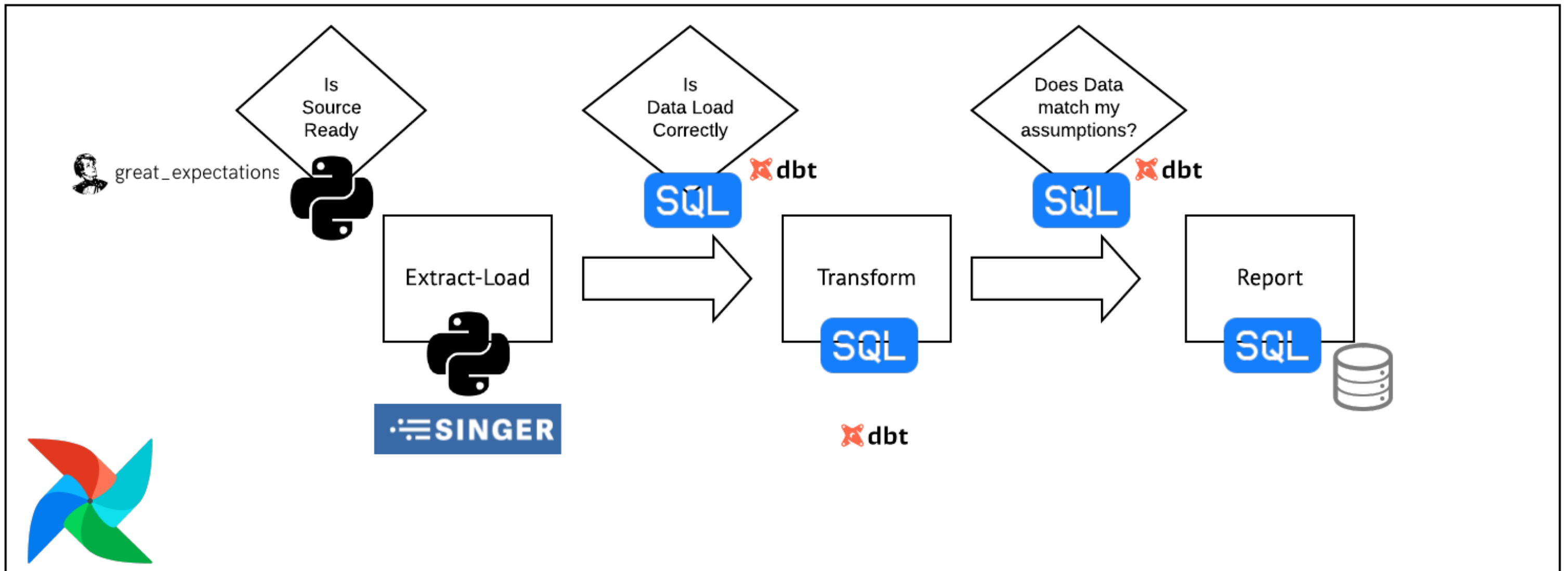






Proposed Solution

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



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	 Reviniate	 SFTP
 SaaSOptics	 Salesforce	 Salesforce Marketing Cloud
 Settigent	 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		

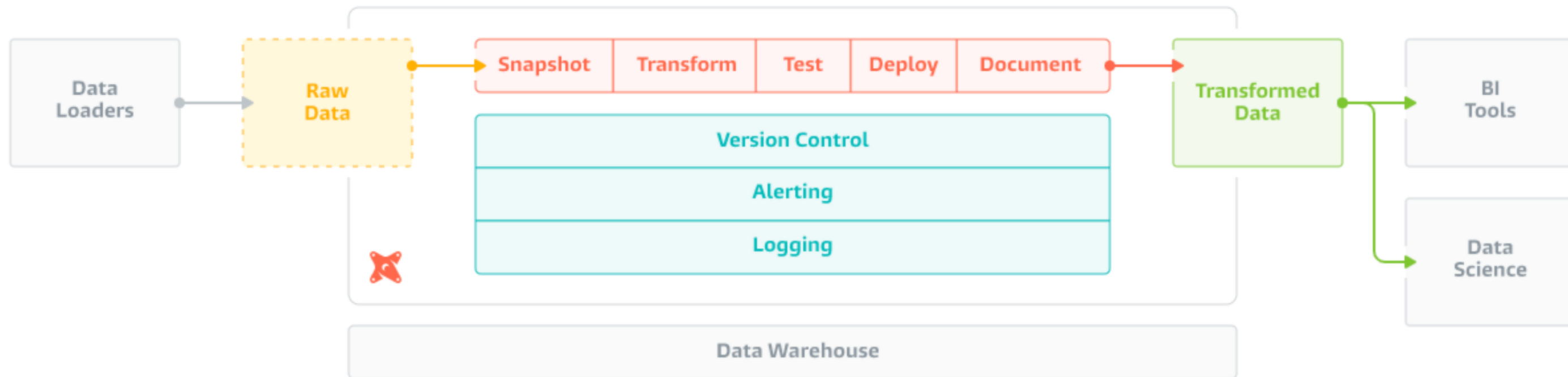
It's a long list

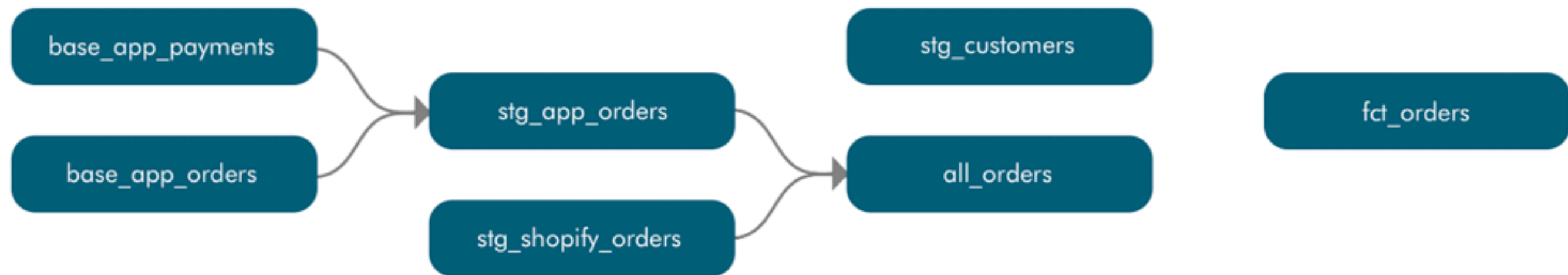
Singer Taps		
Taps extract data from any source and write it to a standard stream in a JSON-based format.		
 3PL Central	 Amazon S3 CSV	 Amplitude
 AppsFlyer	 Autopilot	 BigCommerce
 Bing Ads	 Braintree	 Bronto
 COVID-19 Public Data	 Campaign Manager	 Campaign Monitor
 Chargebee	 Close	 Club Speed
 Codat	 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 Db2	 Impact	 Invoiced
 Jira	 LinkedIn Ads	 Kustomer
 Lever	 Mailshake	 Klaviyo
 LivePerson	 Mixpanel	 MySQL
 Marketo	 Onfleet	 Oracle
 Netsuite Suite Analytics	 Outreach	 Pardot
 Outbrain	 Pipedrive	 Platform Purple
 Pepperjam	 PureCloud	 Quick Base
 PostgreSQL	 Recurly	 Referral SaaSquatch
 Recharge	 Revinat	 SFTP
 Responsys	 Salesforce	 Salesforce Marketing Cloud
 SaaSOptics	 SendGrid Core	 ShipHero
 Settelligent	 Shopify	 Stripe
 Shippo	 Taboola	 Toggle
 SurveyMonkey	 Typeform	 Urban Airship
 Trello	 WooCommerce	 Wootric
 UserVoice	 Xero	 Yotpo
 Workday RaaS	 Zendesk Chat	 Zuora
 Zendesk Support	Develop a Tap	

A large, detailed Transformer robot, Optimus Prime, is shown in a low-angle shot, towering over a building. The robot's body is composed of various mechanical parts, including what appears to be a truck cab in its head and a jet engine in its chest. The background is a cloudy sky. The building behind the robot has a striped hazard pattern at the bottom. The word "Transform" is written in a large, white, serif font across the center of the image.

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

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 periods with no data)

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

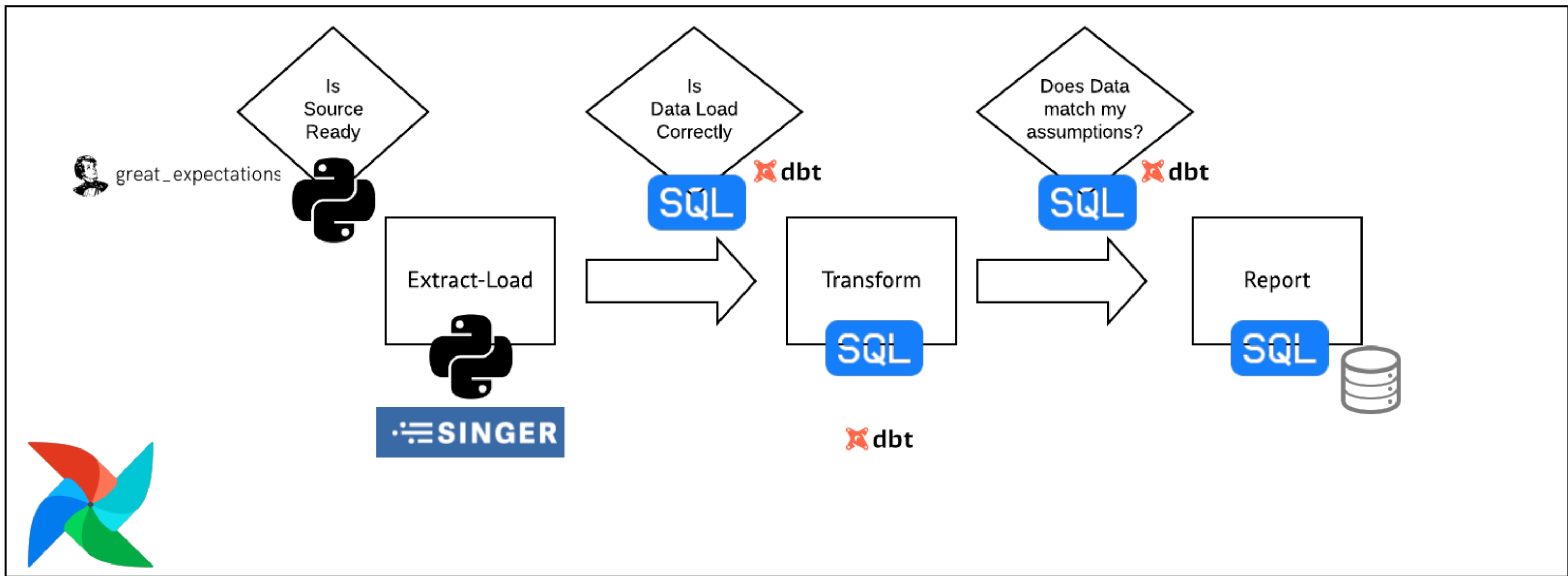
columns:
  - name: event_id
    description: This is a unique identifier for the event
    tests:
      - unique
      - not_null
```

Great adoption

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: ████████████████████
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: ██████████
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_urls: 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: ████████████████████
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: ████████████████████
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: ████████████████████
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: ████████████████████
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](#)