dagster

Rethinking the DAG

Escaping the workflow ↔ data platform impedance mismatch

Pete Hunt
CEO - Dagster Labs

Agenda

- About me
- The mess we’re in
  - Developer velocity
  - Stack complexity
  - Data decentralization
- The impedance mismatch
- Rethinking the DAG and its implications

Who am I?

CEO at Dagster Labs

- Dagster Core (OSS)
- Dagster+ (coming April 17)

Ran data teams at Twitter and co-founded a streaming data company for detecting online abuse

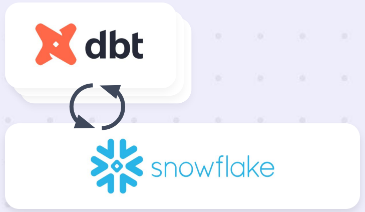
Founding member of the React project at Facebook

The mess we’re in

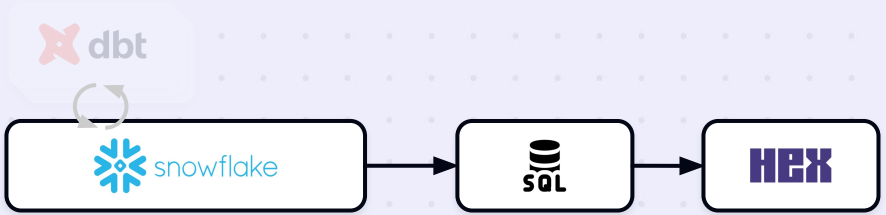
Imagine yourself as a new Head of Data at a Series B company building a Spotify competitor

dagstereo

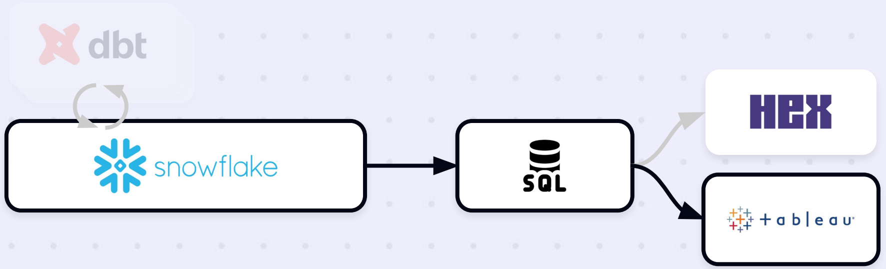
Lydia / Analytics Engineer



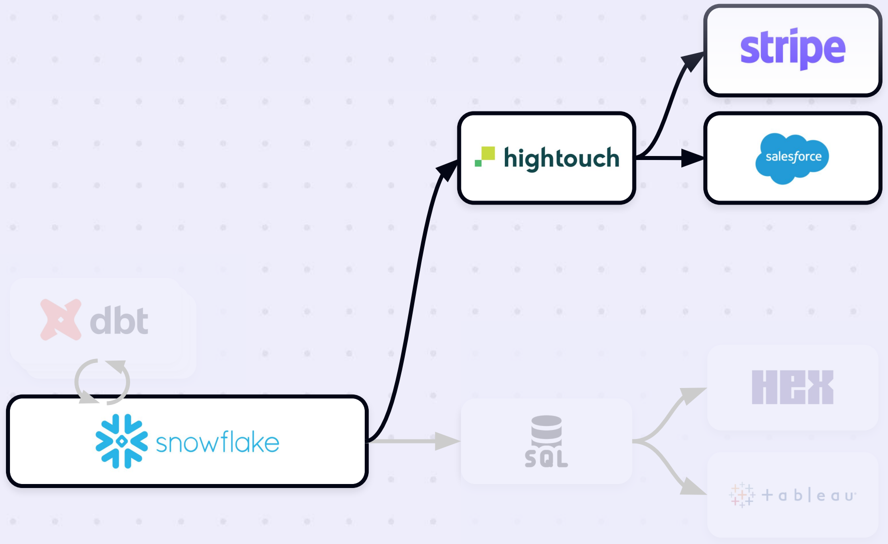
Olivia / CEO



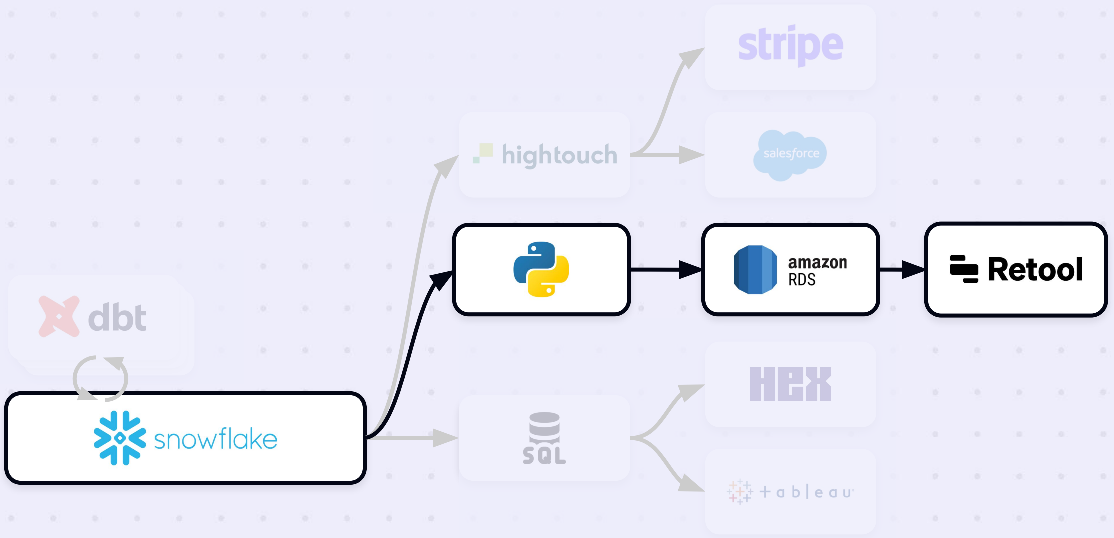
Ed / Head of Marketing



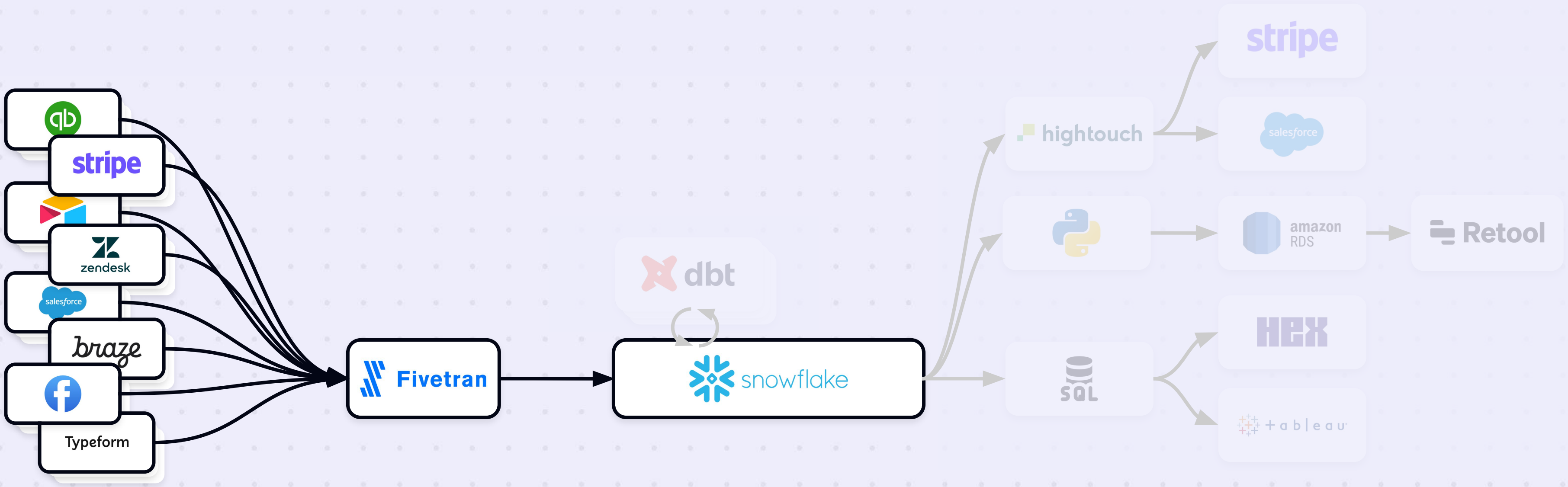
Wade / Head of Sales



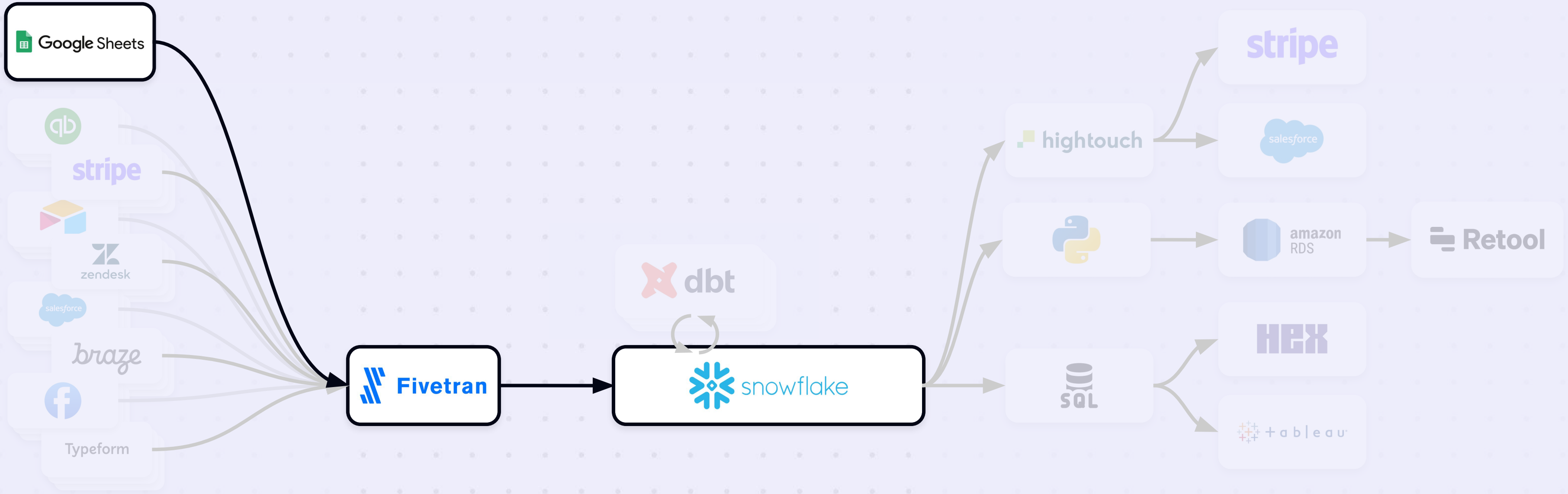
Joel / Customer Success Manager



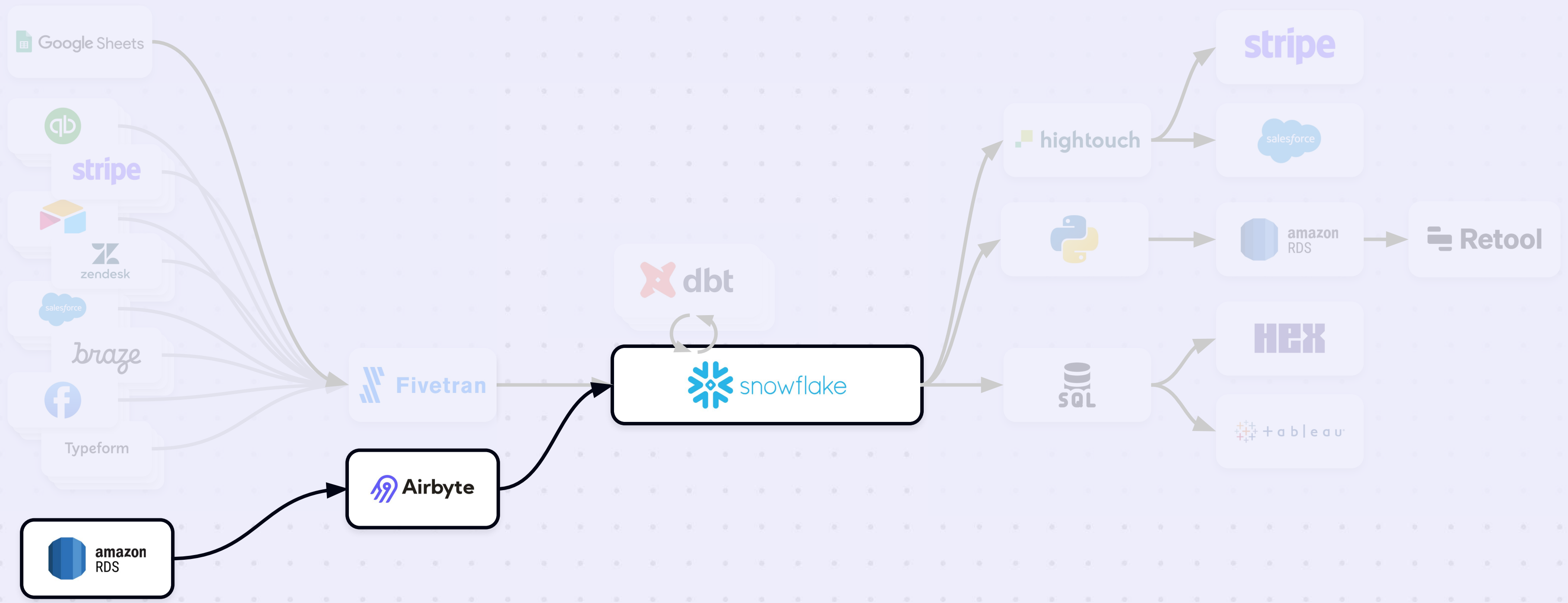
Joel / Data Contractor at ACME Analytics



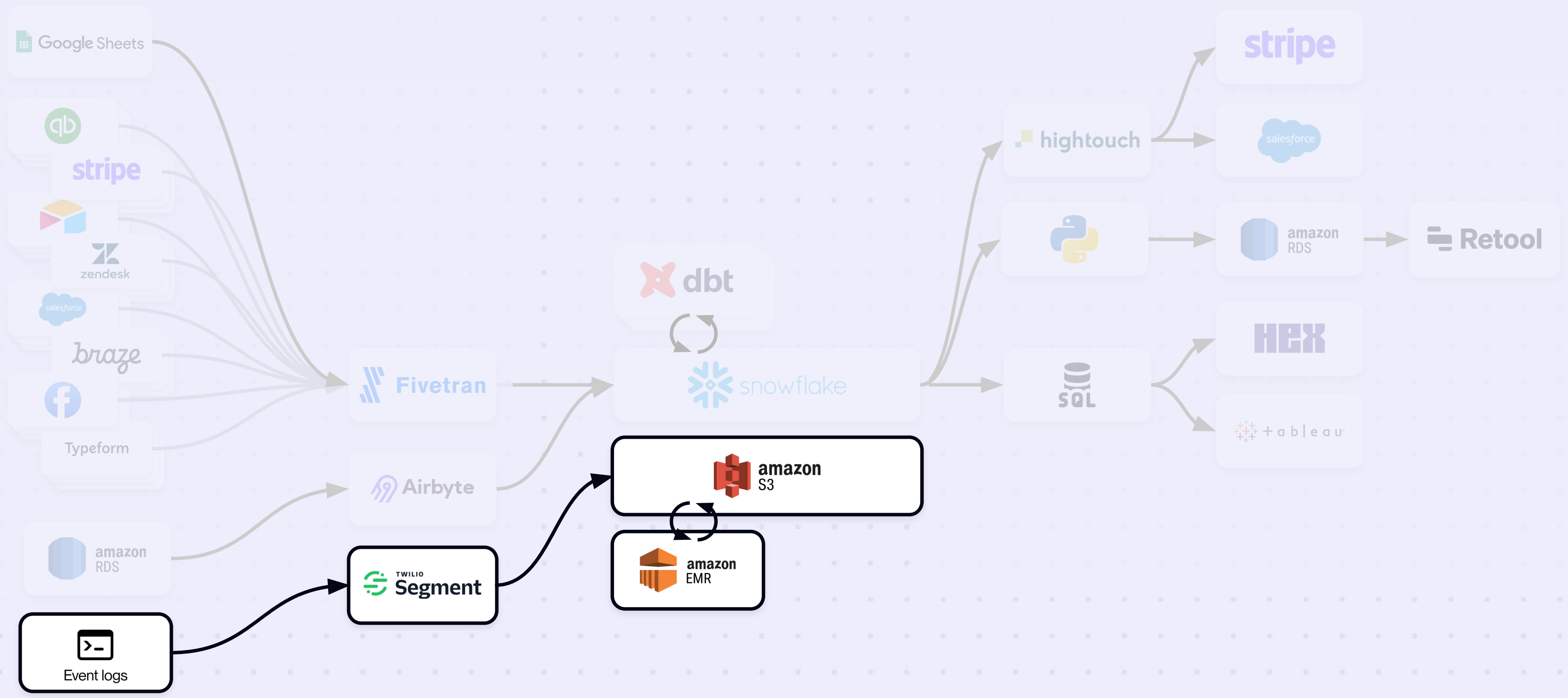
Grace / Product Manager



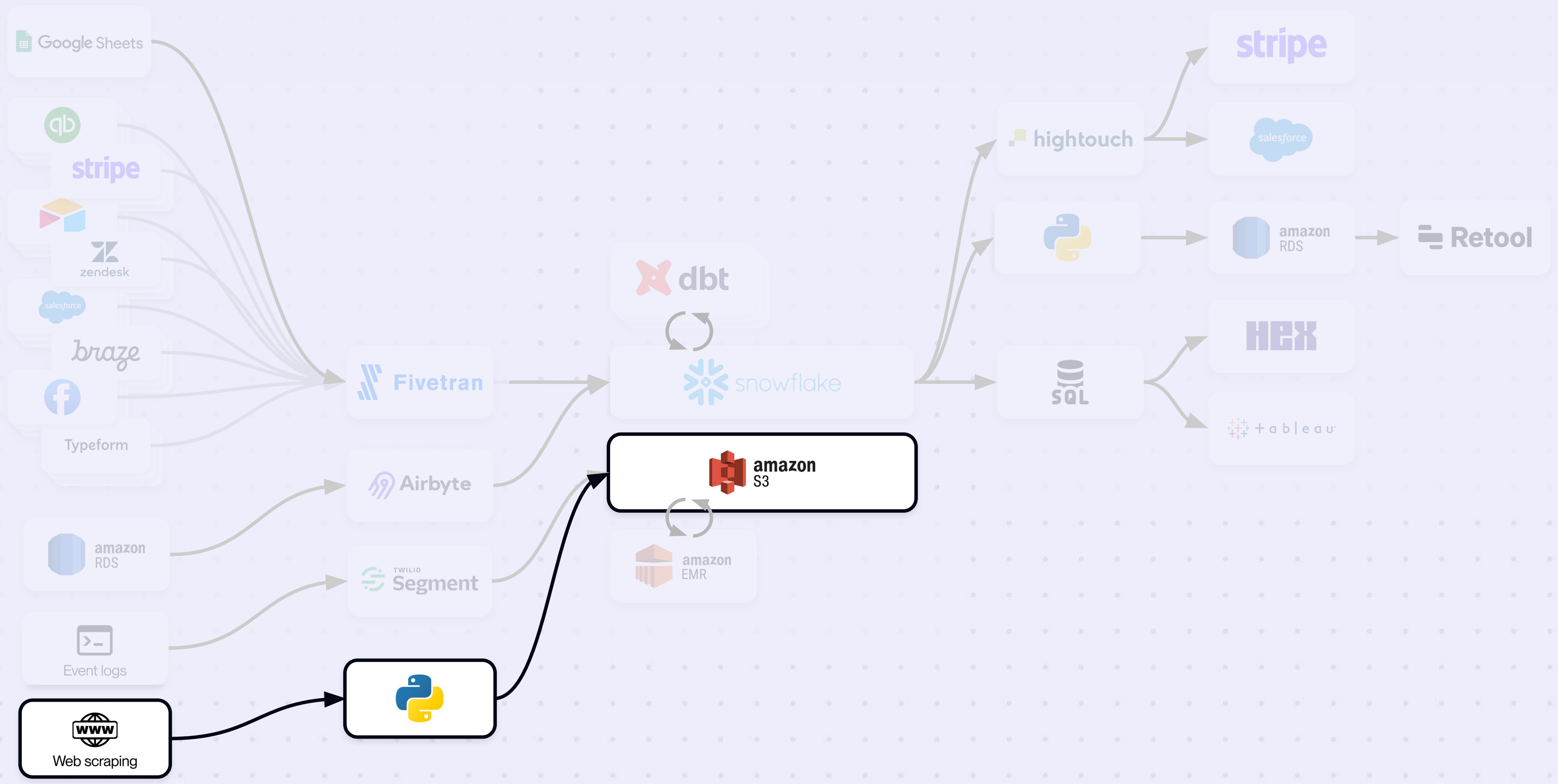
Peyton / Data Engineer



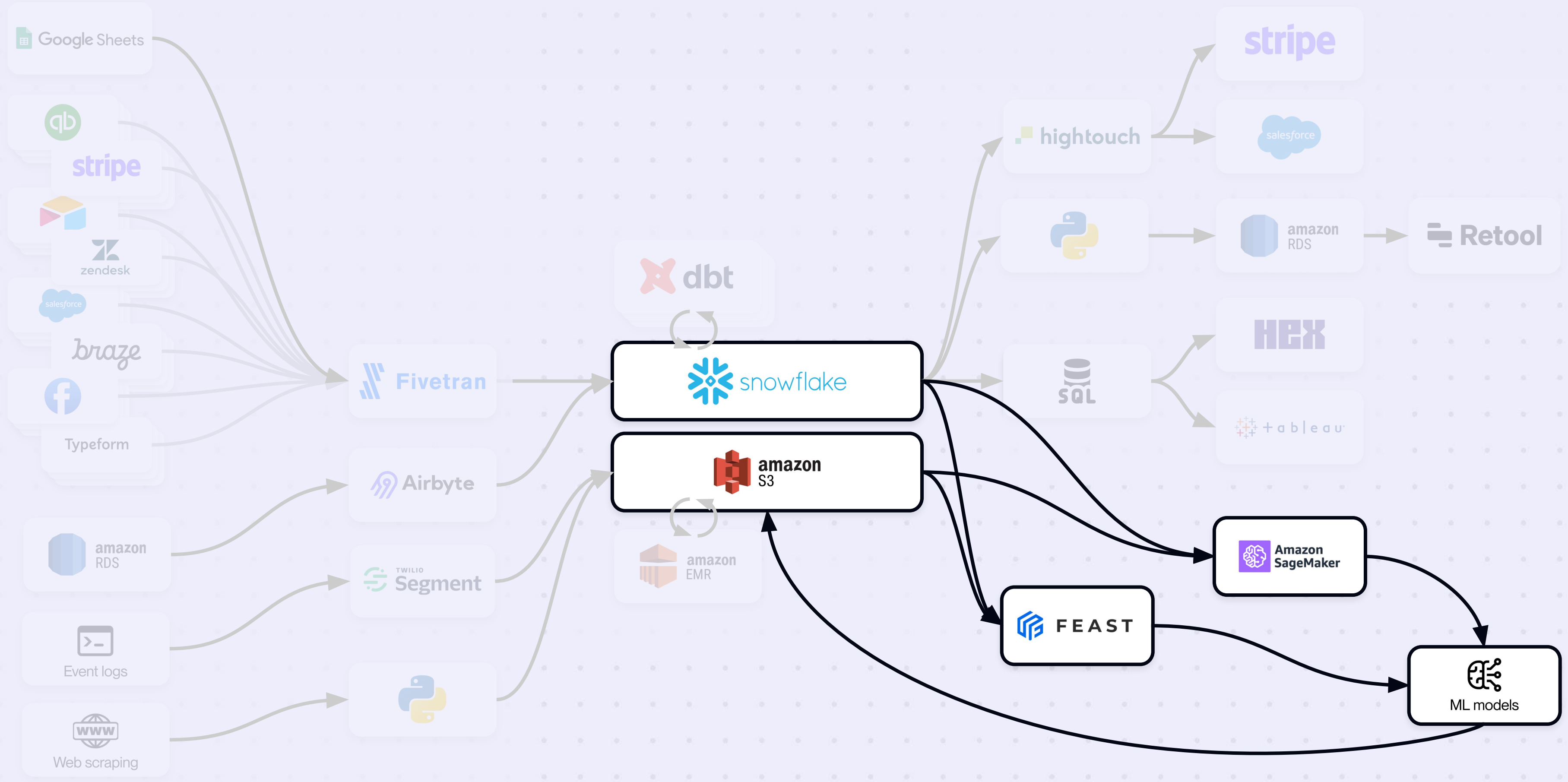
Carlie / Product Engineer



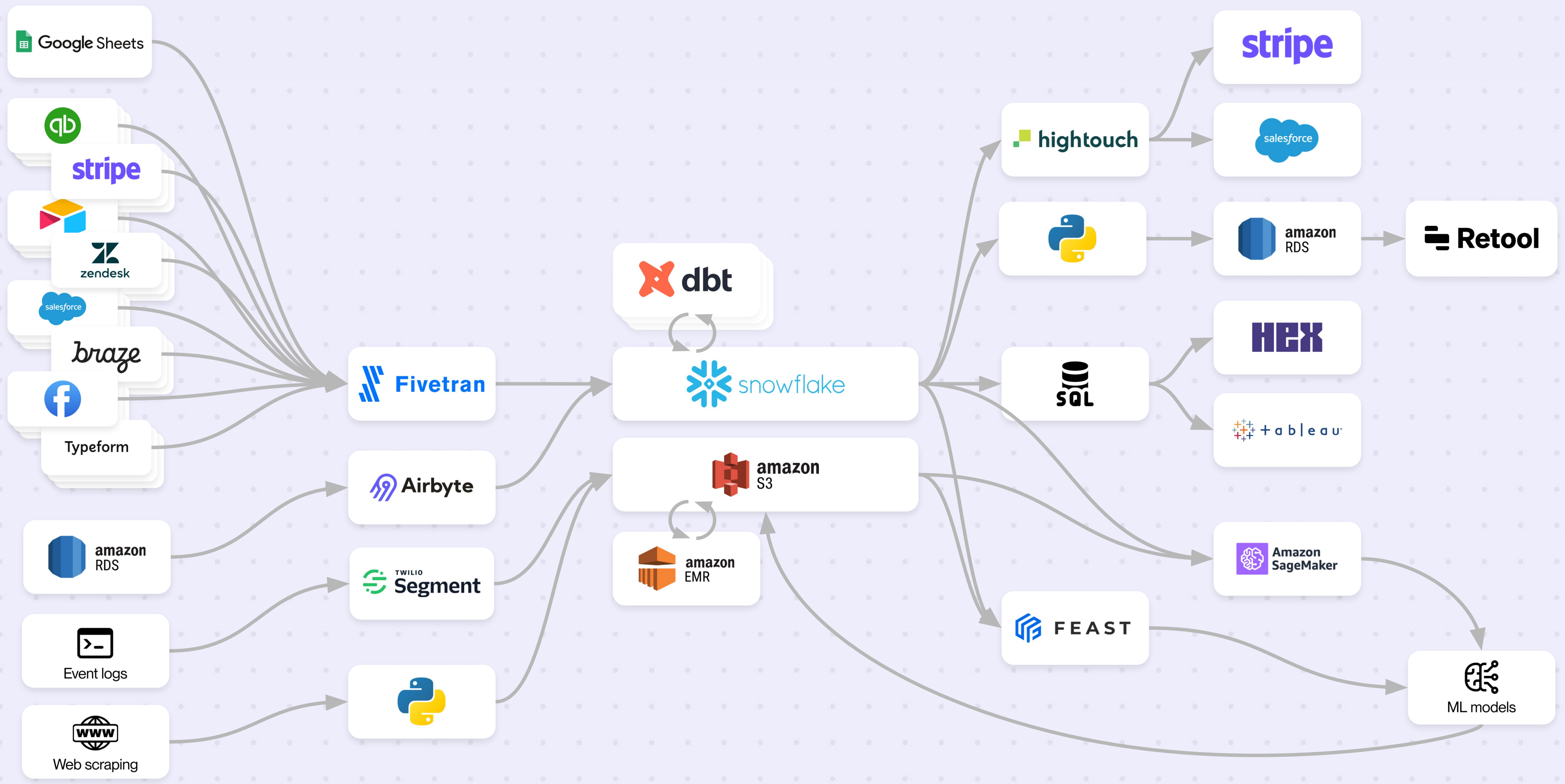
Oscar / ML Engineer



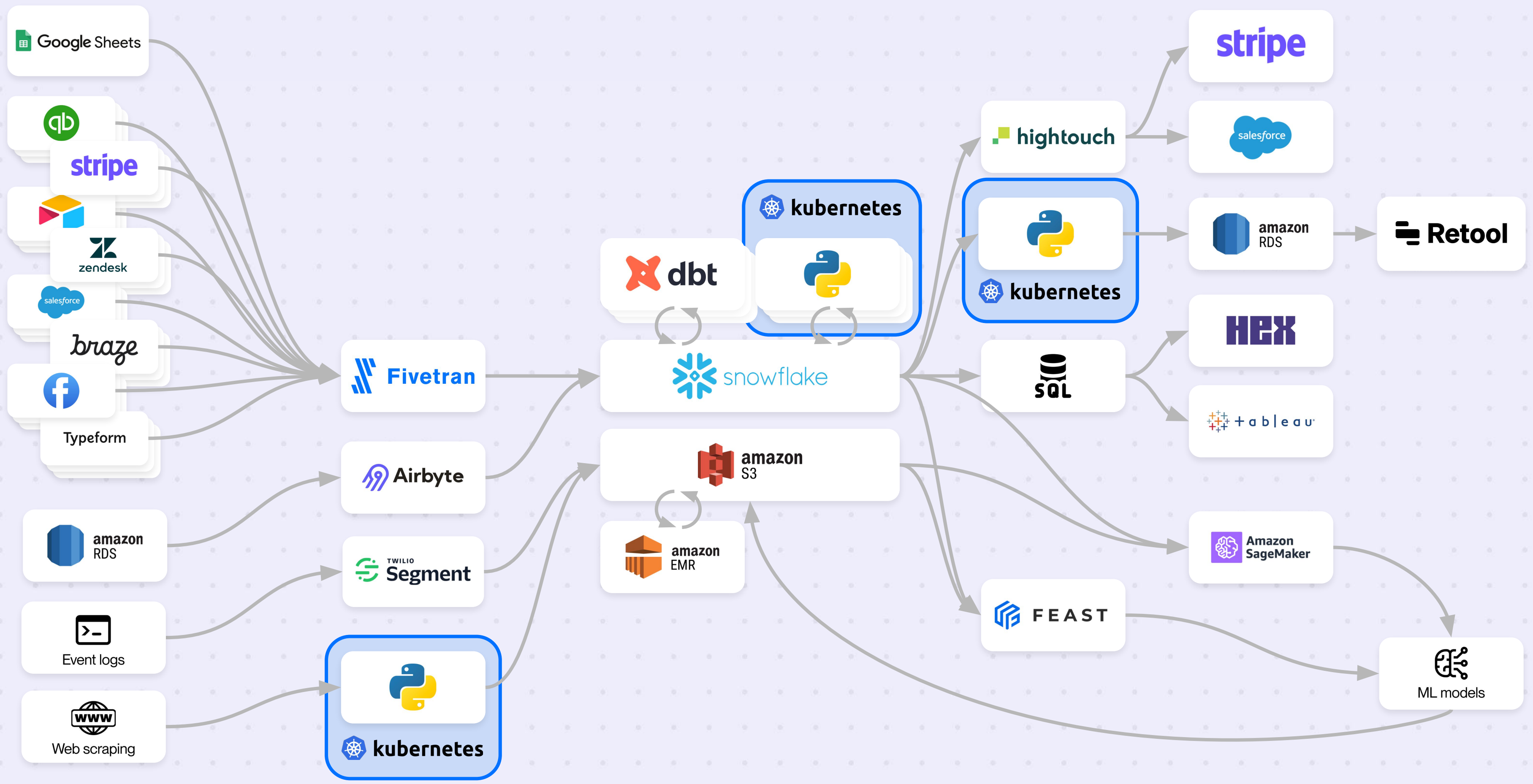
Hector / Data Scientist



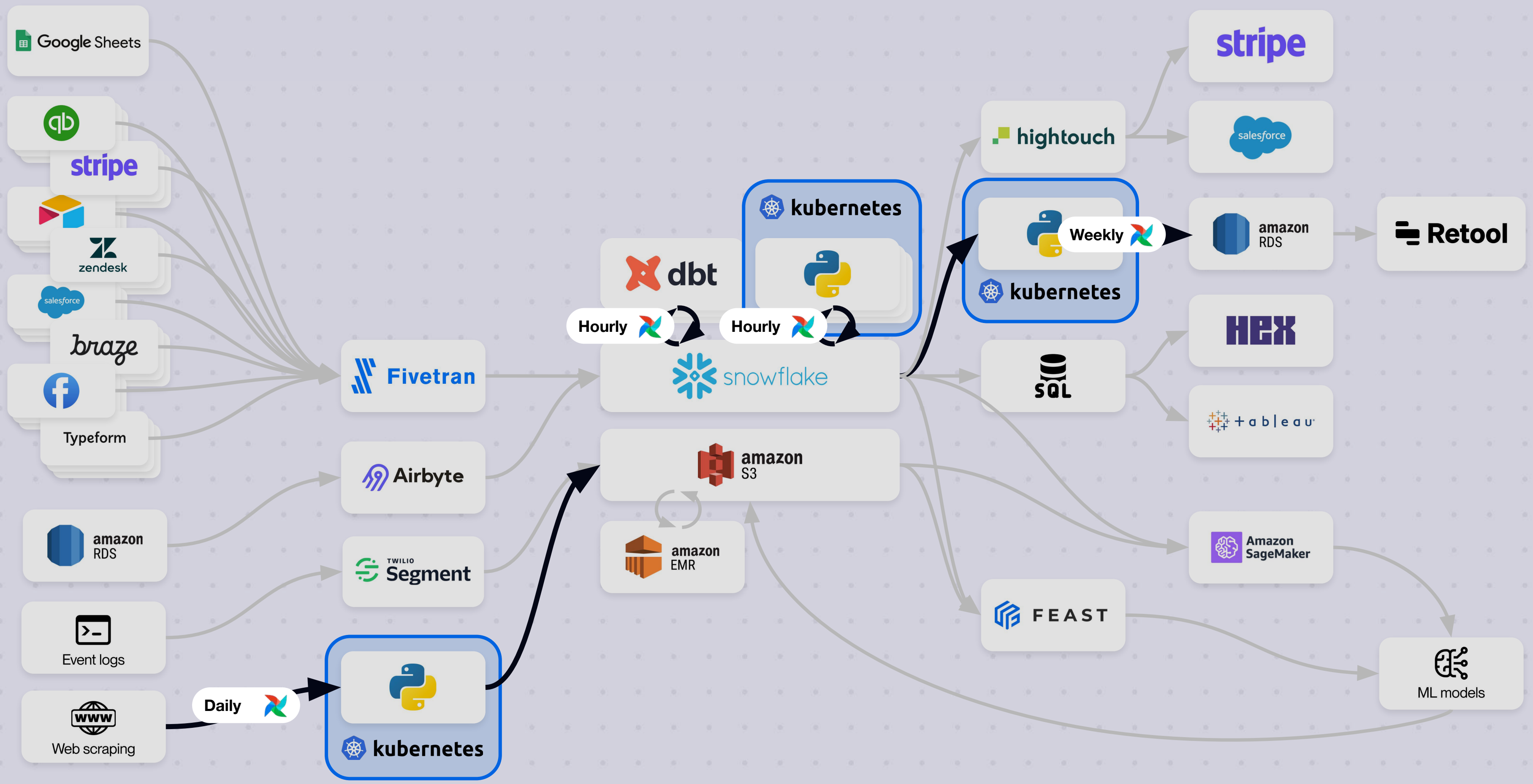
Luke / Data Engineering Lead



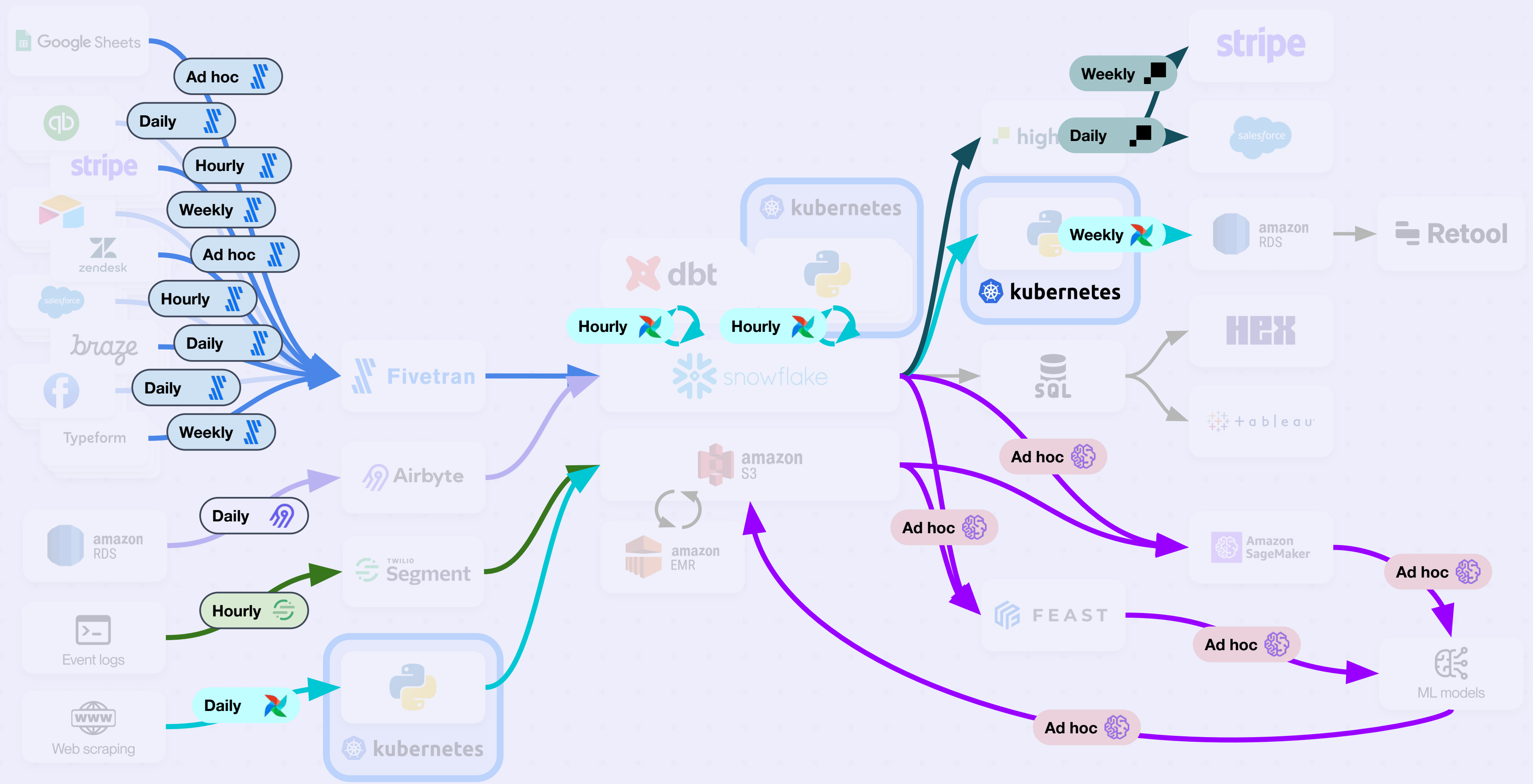
Luke / Data Engineering Lead



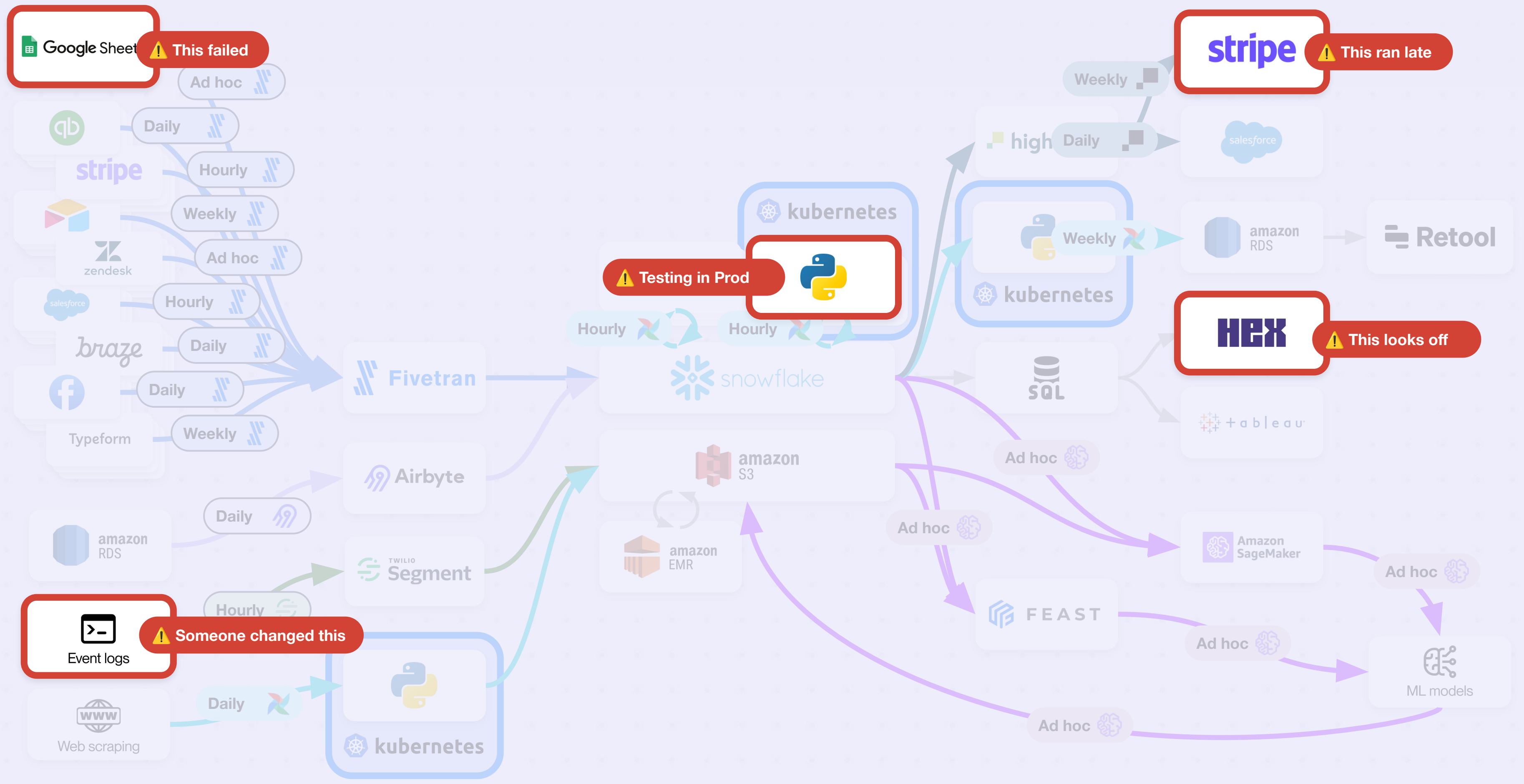
Luke / Data Engineering Lead



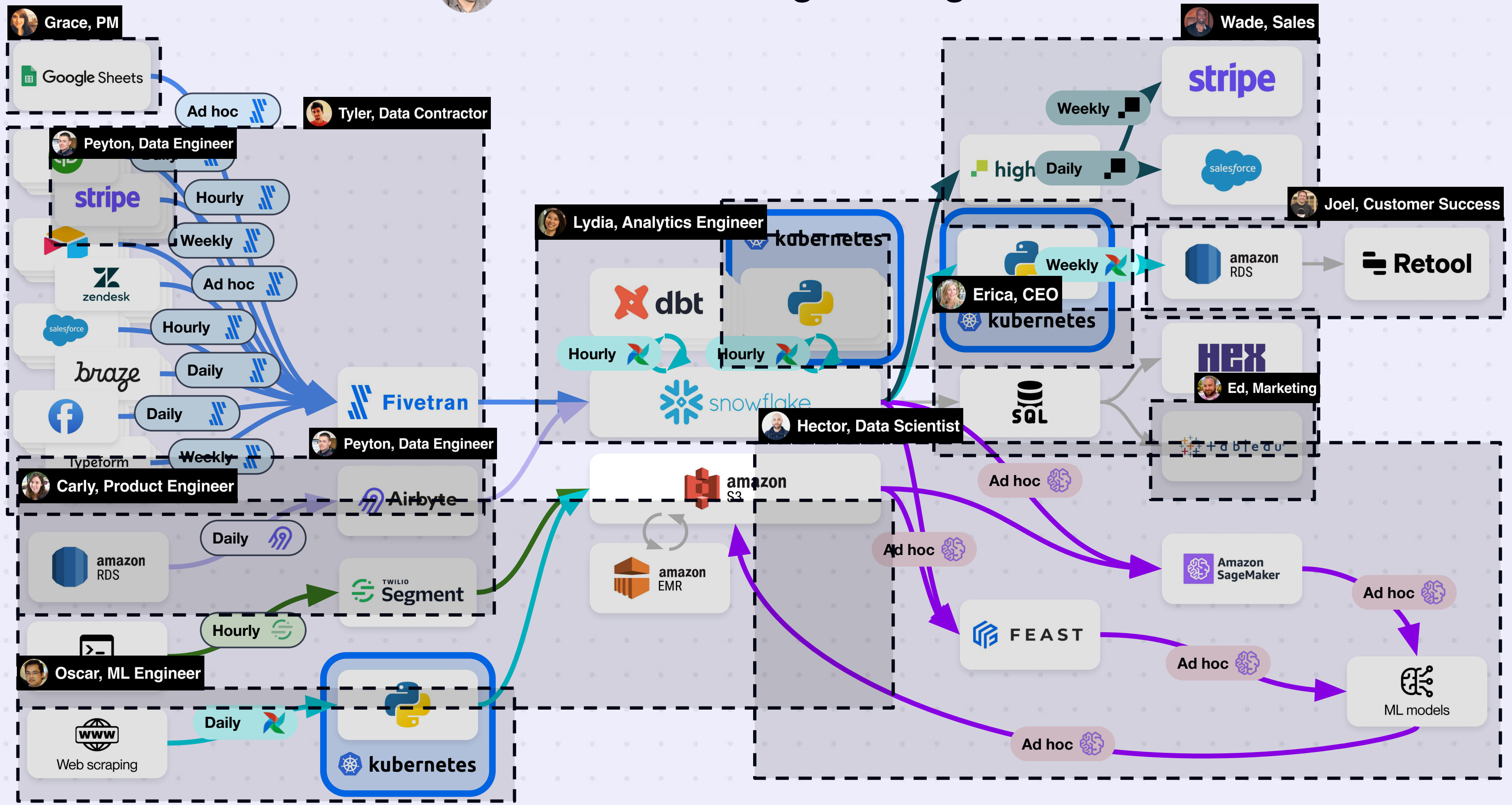
Luke / Data Engineering Lead

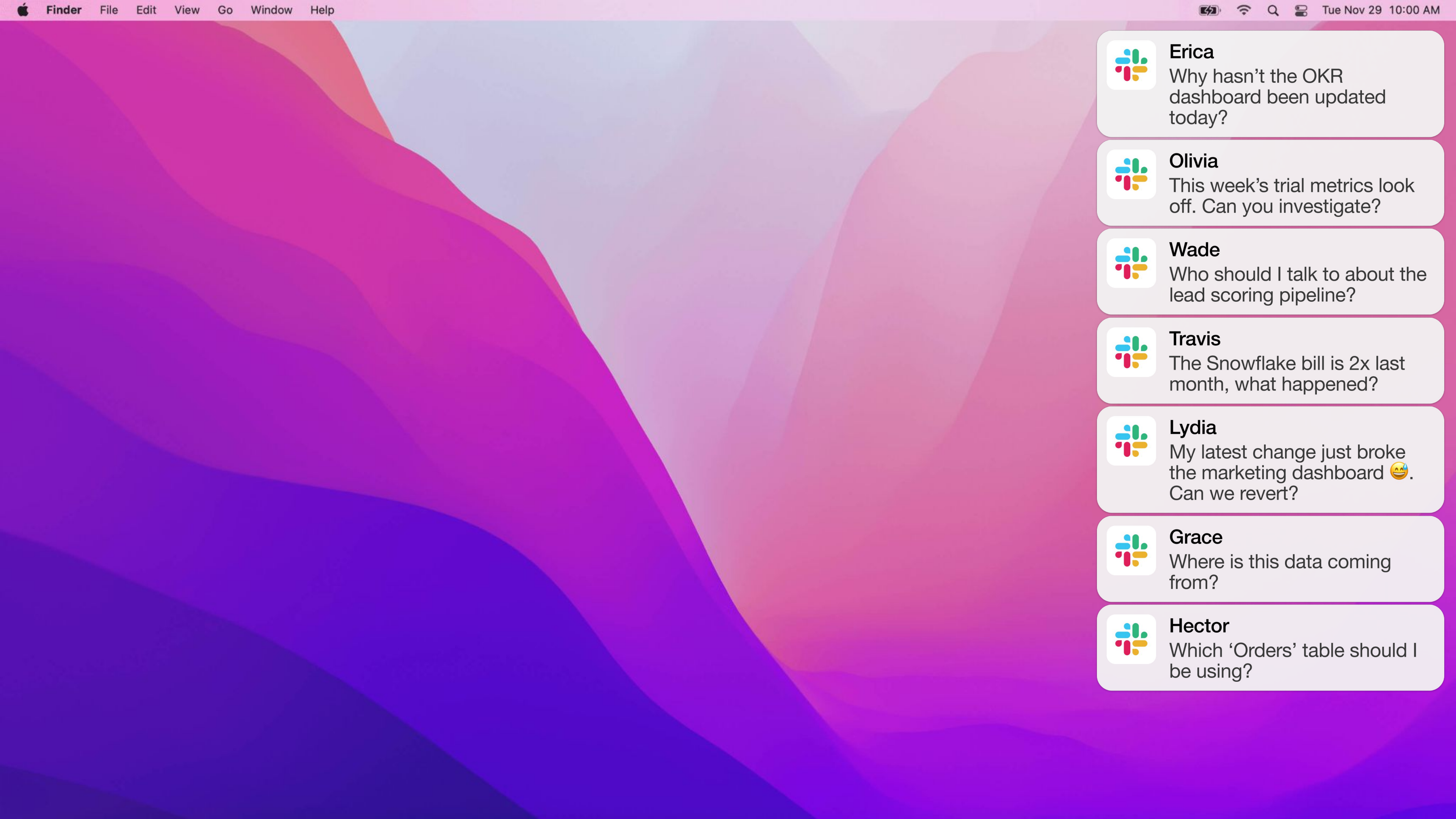


Luke / Data Engineering Lead



Luke / Data Engineering Lead





Erica
Why hasn't the OKR dashboard been updated today?

Olivia
This week's trial metrics look off. Can you investigate?

Wade
Who should I talk to about the lead scoring pipeline?

Travis
The Snowflake bill is 2x last month, what happened?

Lydia
My latest change just broke the marketing dashboard 😅. Can we revert?

Grace
Where is this data coming from?

Hector
Which 'Orders' table should I be using?

Luke’s life is chaos

Slow & painful dev experience

Hard to identify, debug and fix problems across a large codebase that cuts across multiple tools

Complex & costly tech stack

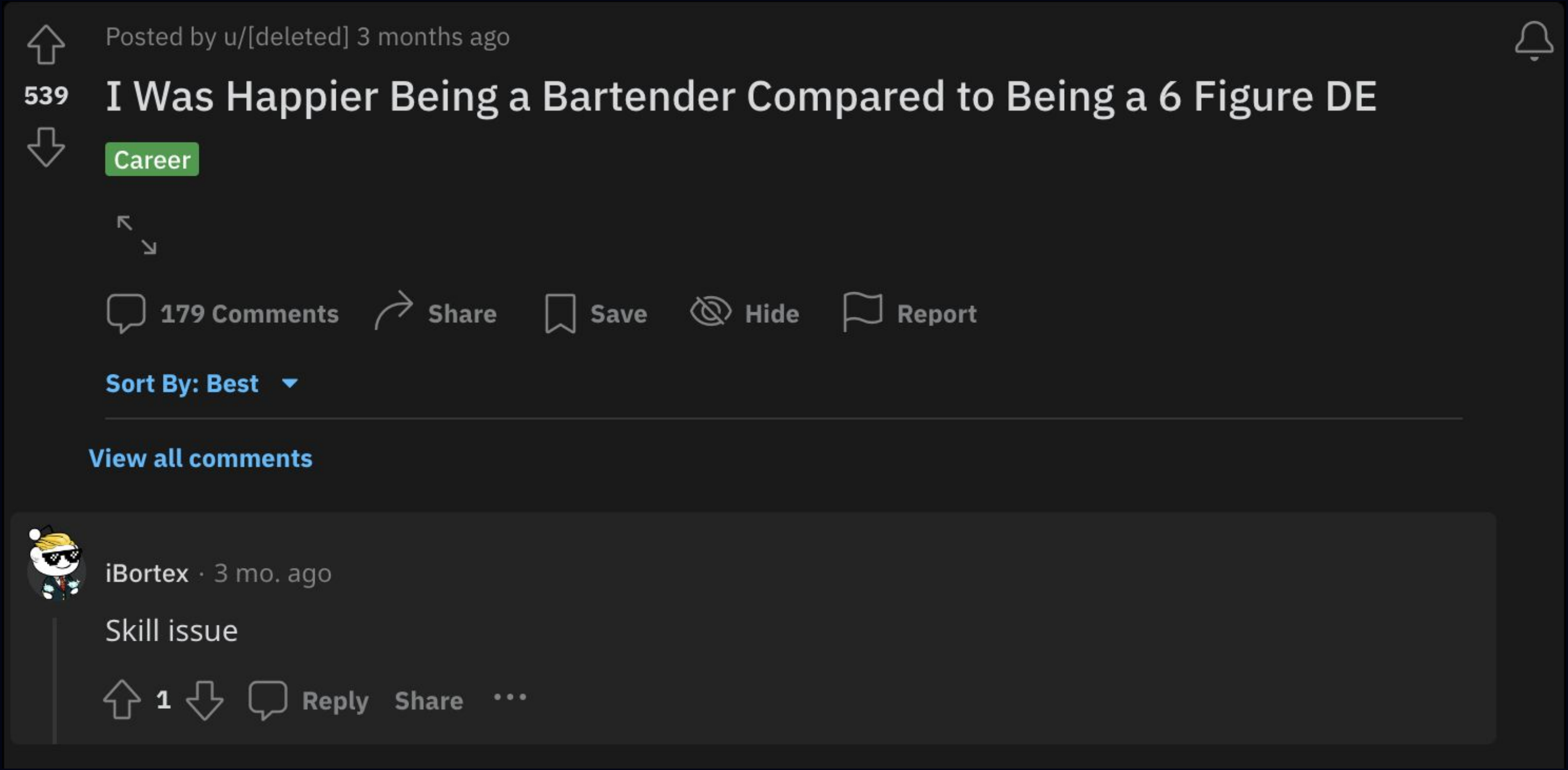
Dozens of point solutions to integrate and maintain with no “single pane of glass” for observability.

Chaotic & intricate ownership

Difficult to strike a balance between centralization and decentralization while maximizing productivity.

Why?

Skill issue?



Posted by u/[deleted] 3 months ago

539

I Was Happier Being a Bartender Compared to Being a 6 Figure DE

Career

179 Comments Share Save Hide Report

Sort By: Best

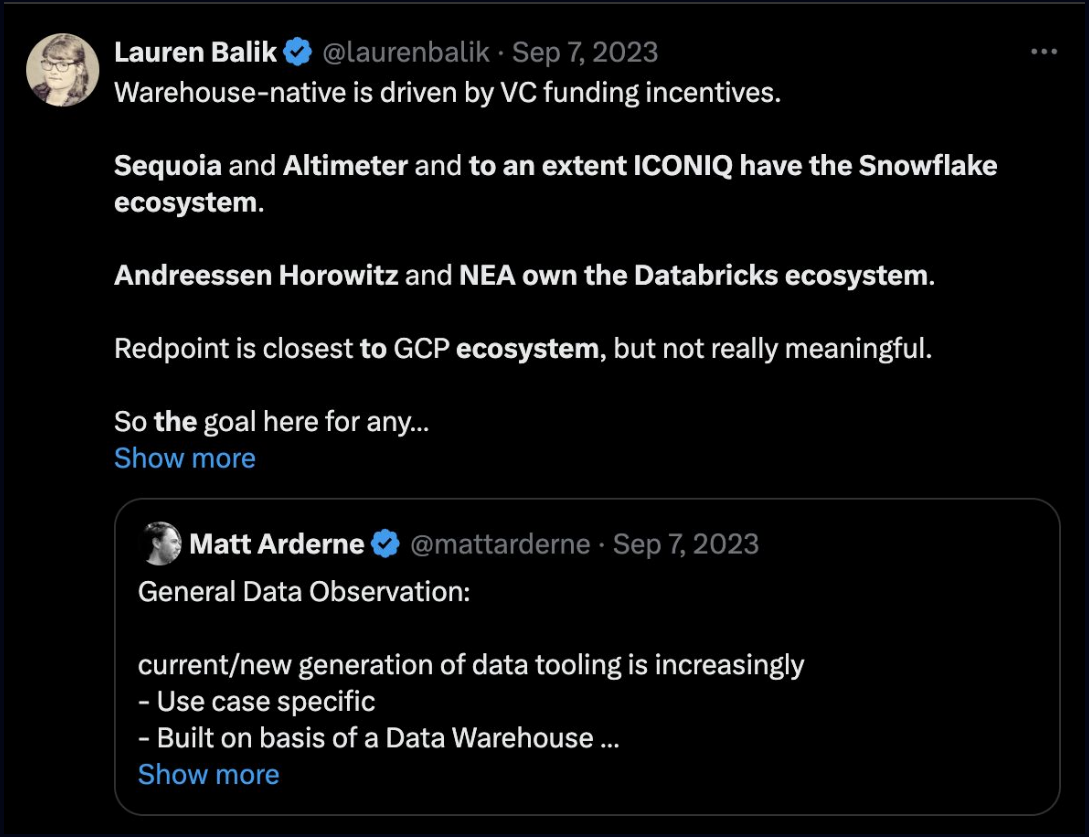
View all comments

iBortex · 3 mo. ago

Skill issue

1 Reply Share ···

ZIRPy venture-backed data companies?



Lauren Balik @laurenbalik · Sep 7, 2023

Warehouse-native is driven by VC funding incentives.

Sequoia and **Altimeter** and **to an extent ICONIQ have the Snowflake ecosystem**.

Andreessen Horowitz and **NEA own the Databricks ecosystem**.

Redpoint is closest **to** GCP **ecosystem**, but not really meaningful.

So **the** goal here for any...
Show more

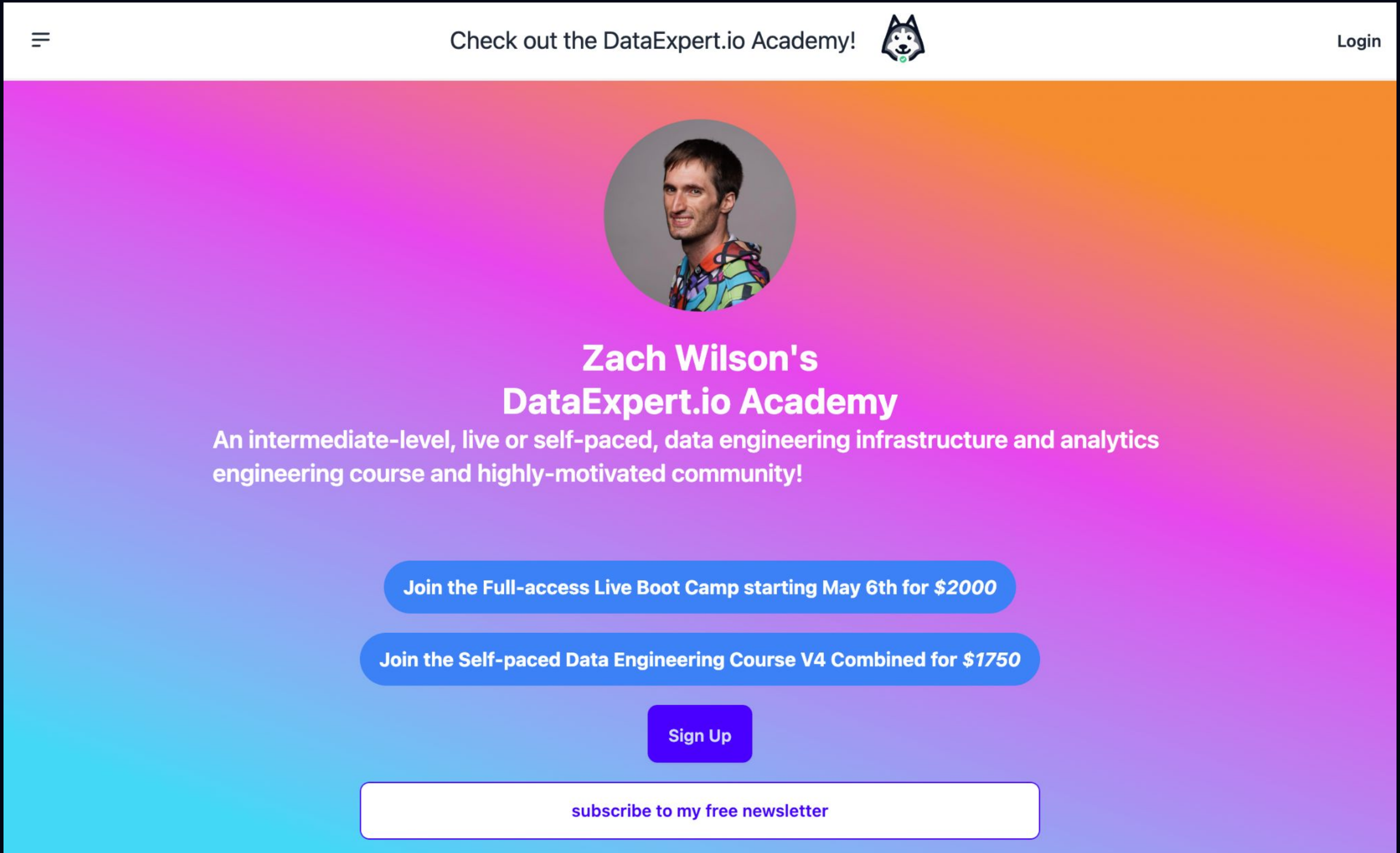
Matt Arderne @mattarderne · Sep 7, 2023

General Data Observation:

current/new generation of data tooling is increasingly
- Use case specific
- Built on basis of a Data Warehouse ...

Show more

Is data just *that* hard?



Check out the DataExpert.io Academy!

Login

Zach Wilson's
DataExpert.io Academy

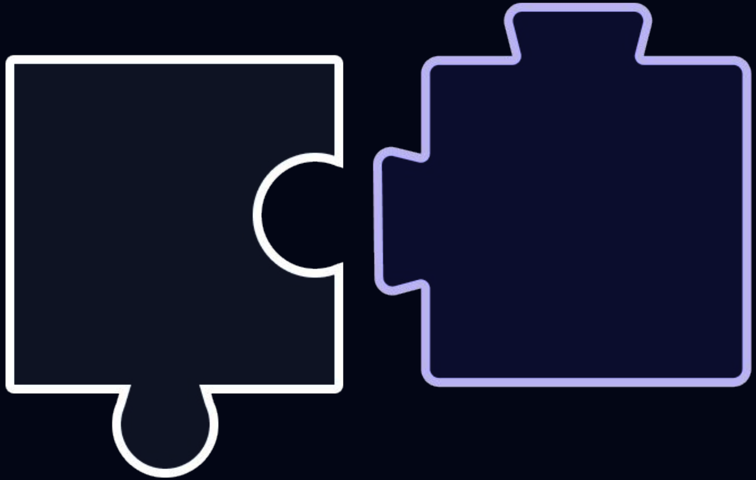
An intermediate-level, live or self-paced, data engineering infrastructure and analytics engineering course and highly-motivated community!

Join the Full-access Live Boot Camp starting May 6th for *$2000*
Join the Self-paced Data Engineering Course V4 Combined for *$1750*
Sign Up
subscribe to my free newsletter

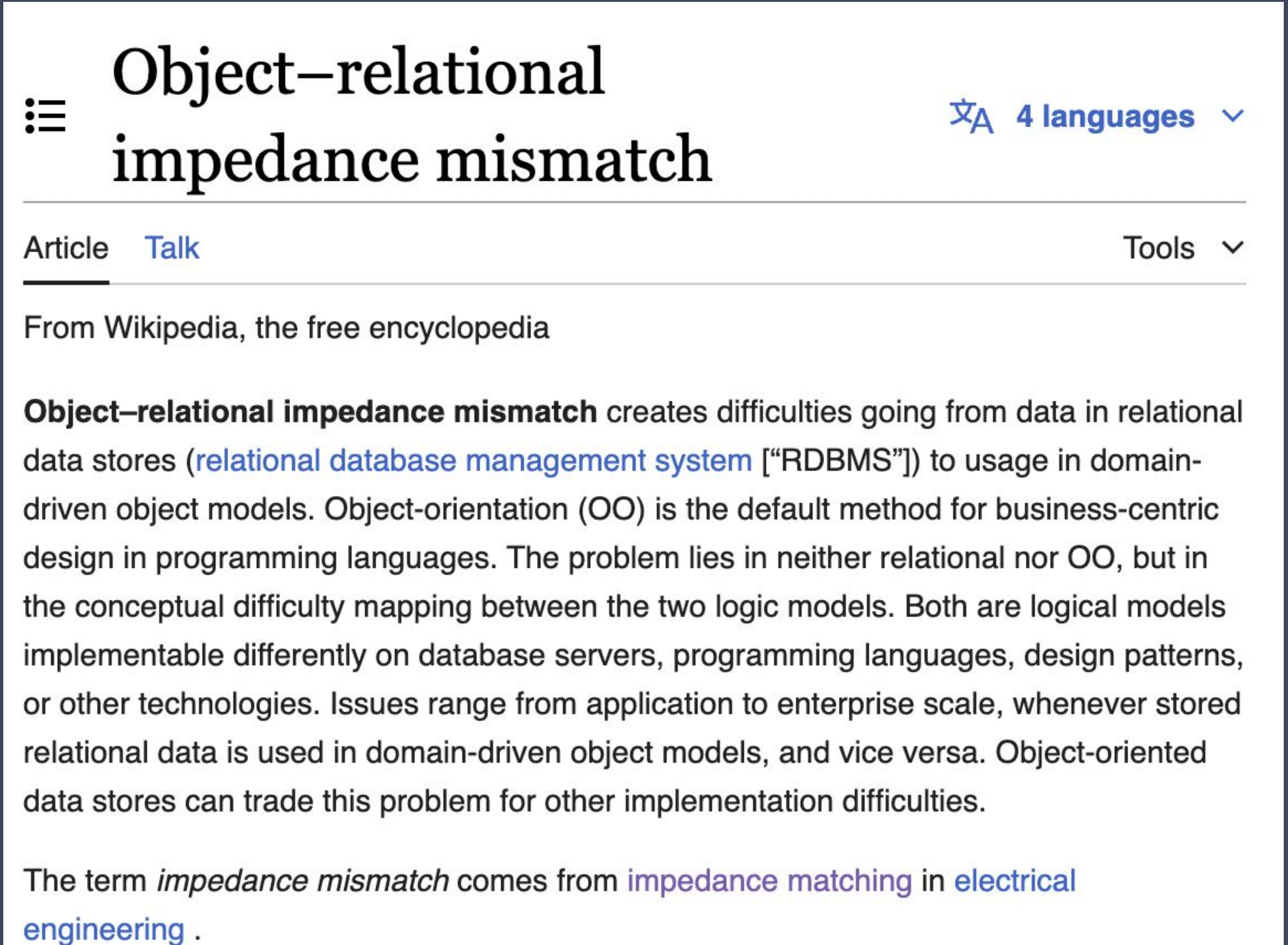
The root technical cause

An **impedance mismatch** between workflow engines and the rest of the data platform

Impedance mismatch?



When two layers of a system use fundamentally incompatible domain models



Object–relational impedance mismatch

4 languages

Article Talk Tools

From Wikipedia, the free encyclopedia

Object–relational impedance mismatch creates difficulties going from data in relational data stores (relational database management system ["RDBMS"]) to usage in domain-driven object models. Object-orientation (OO) is the default method for business-centric design in programming languages. The problem lies in neither relational nor OO, but in the conceptual difficulty mapping between the two logic models. Both are logical models implementable differently on database servers, programming languages, design patterns, or other technologies. Issues range from application to enterprise scale, whenever stored relational data is used in domain-driven object models, and vice versa. Object-oriented data stores can trade this problem for other implementation difficulties.

The term *impedance mismatch* comes from impedance matching in electrical engineering .

The impedance mismatch in data

Workflow Engines

Workflow-oriented tools

Focused on the **task**: a function that performs some work and can depend on other tasks.

Examples: bash scripts, Python functions, K8s jobs

- Airflow
- Prefect
- Github Actions

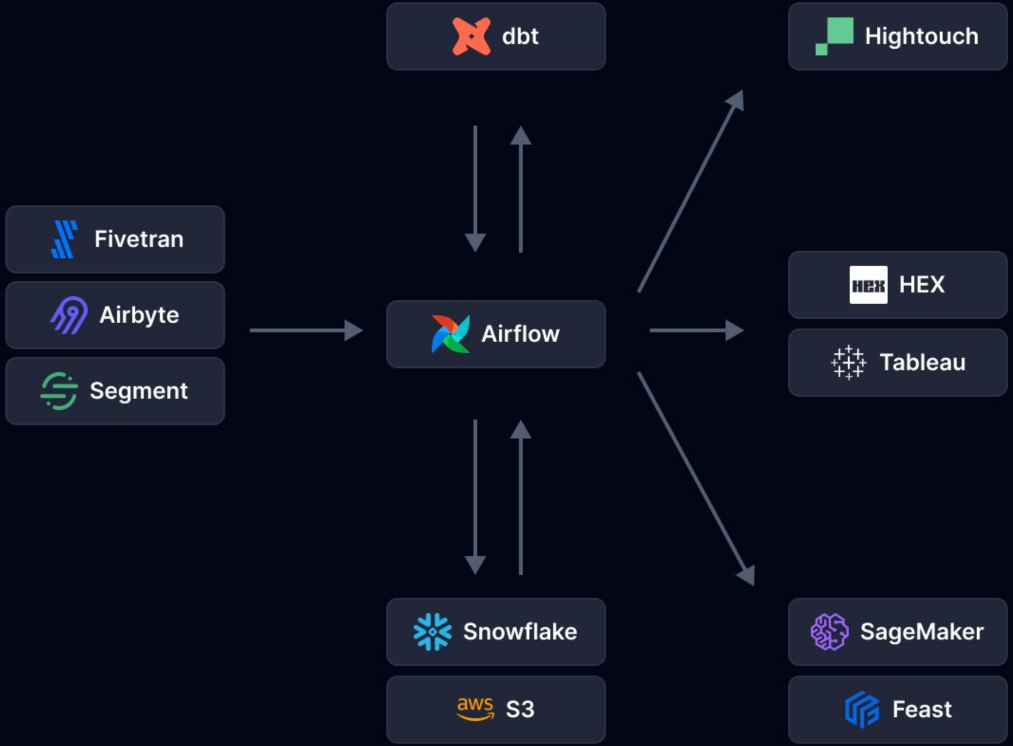
The rest of the platform

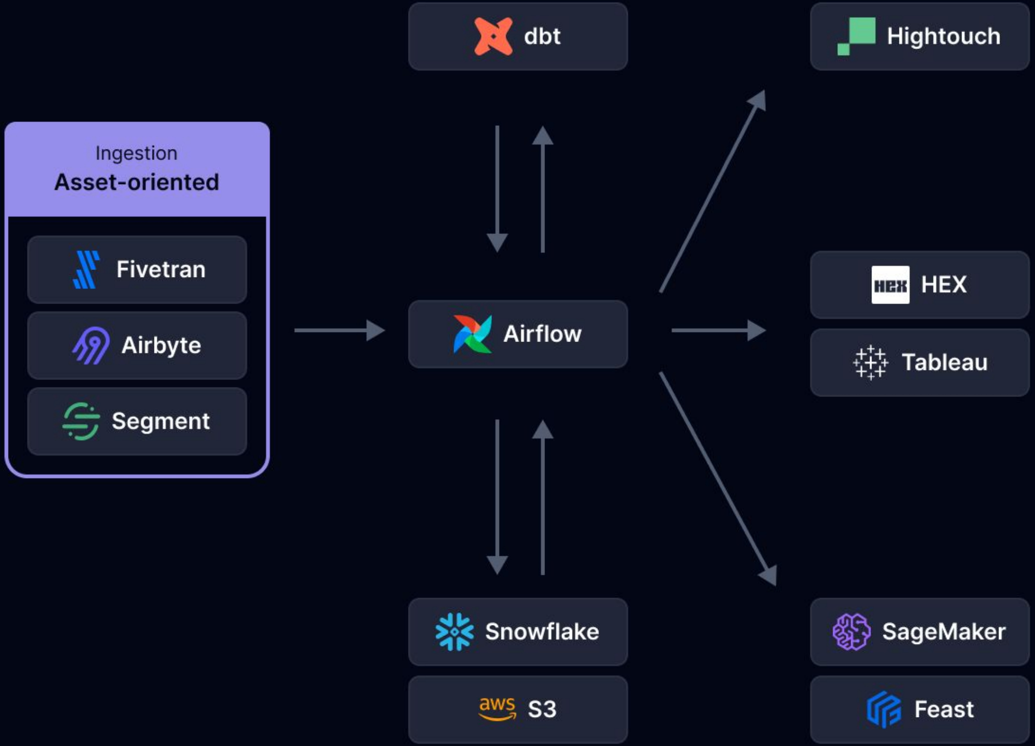
Asset-oriented tools

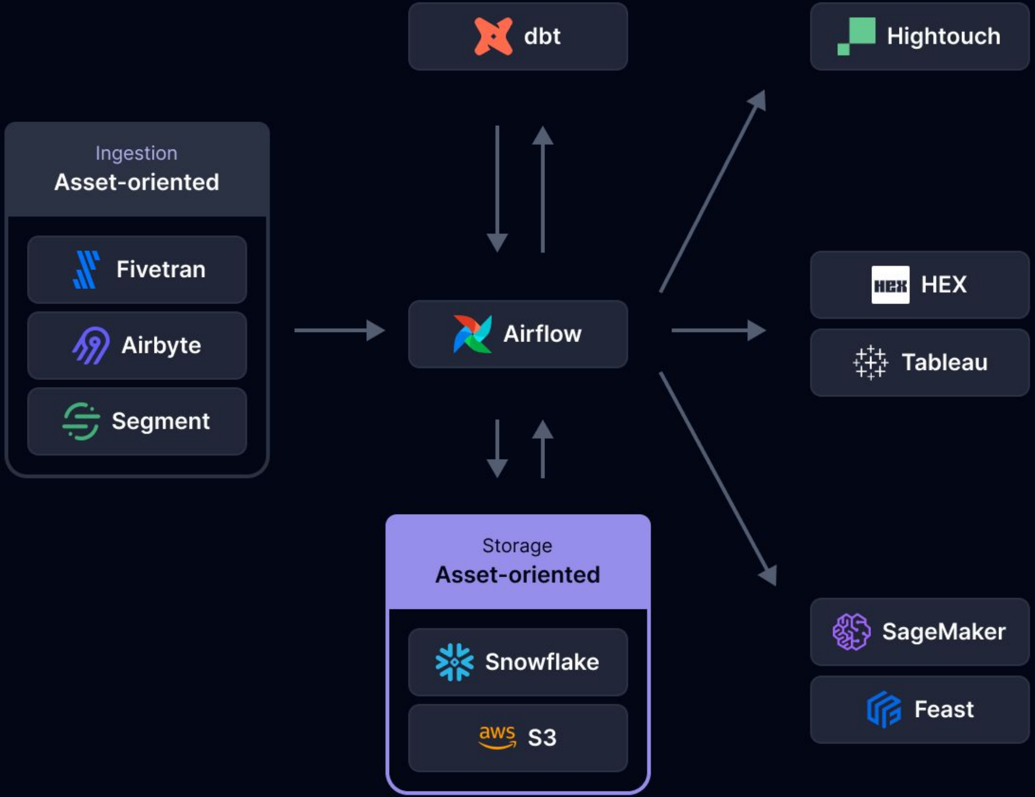
Focused on the **data asset**: an object in persistent storage that captures some understanding of the world

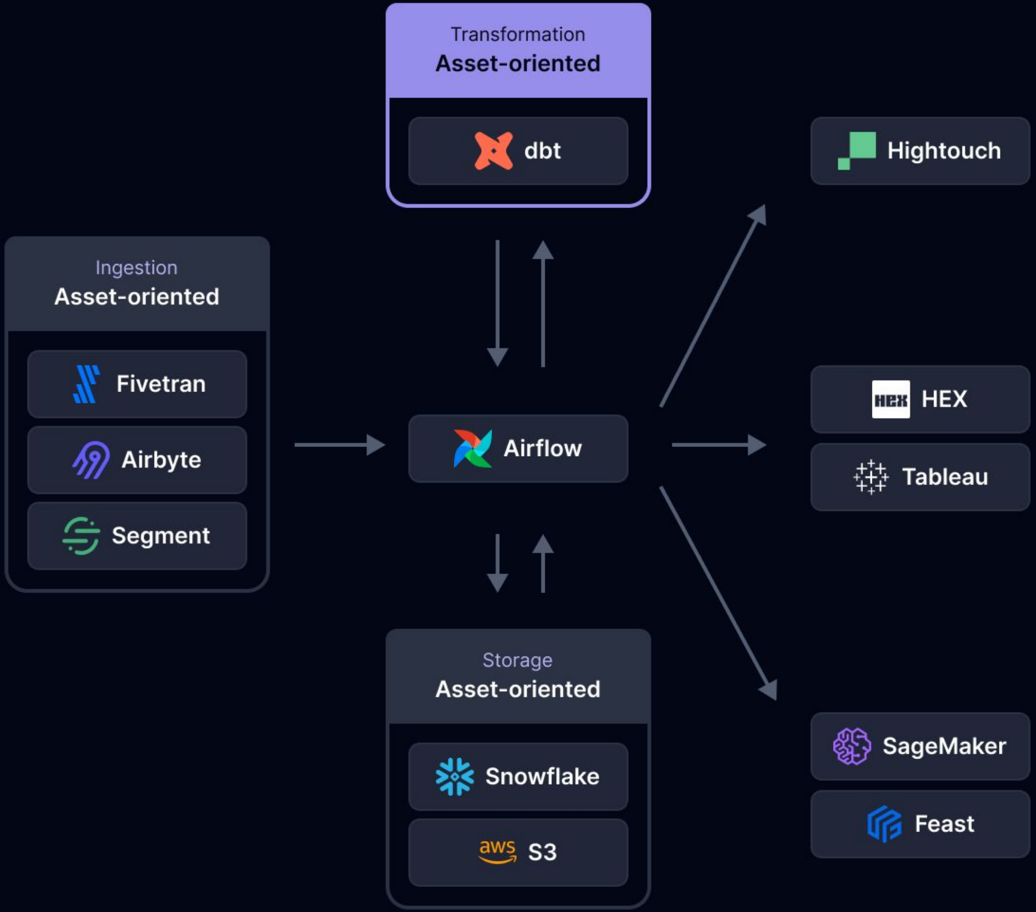
Examples: database tables, ML model, dashboards

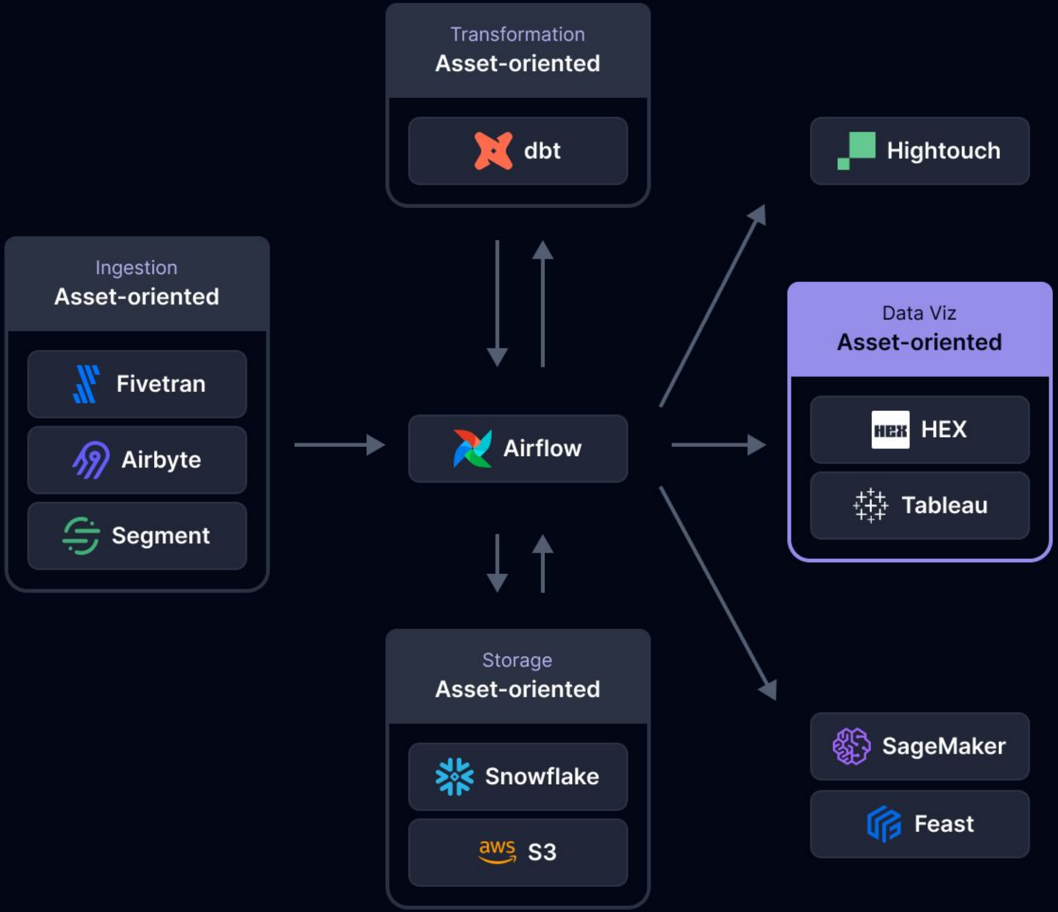
- dbt
- Dagster
- Fivetran
- Snowflake
- Airbyte
- Monte Carlo

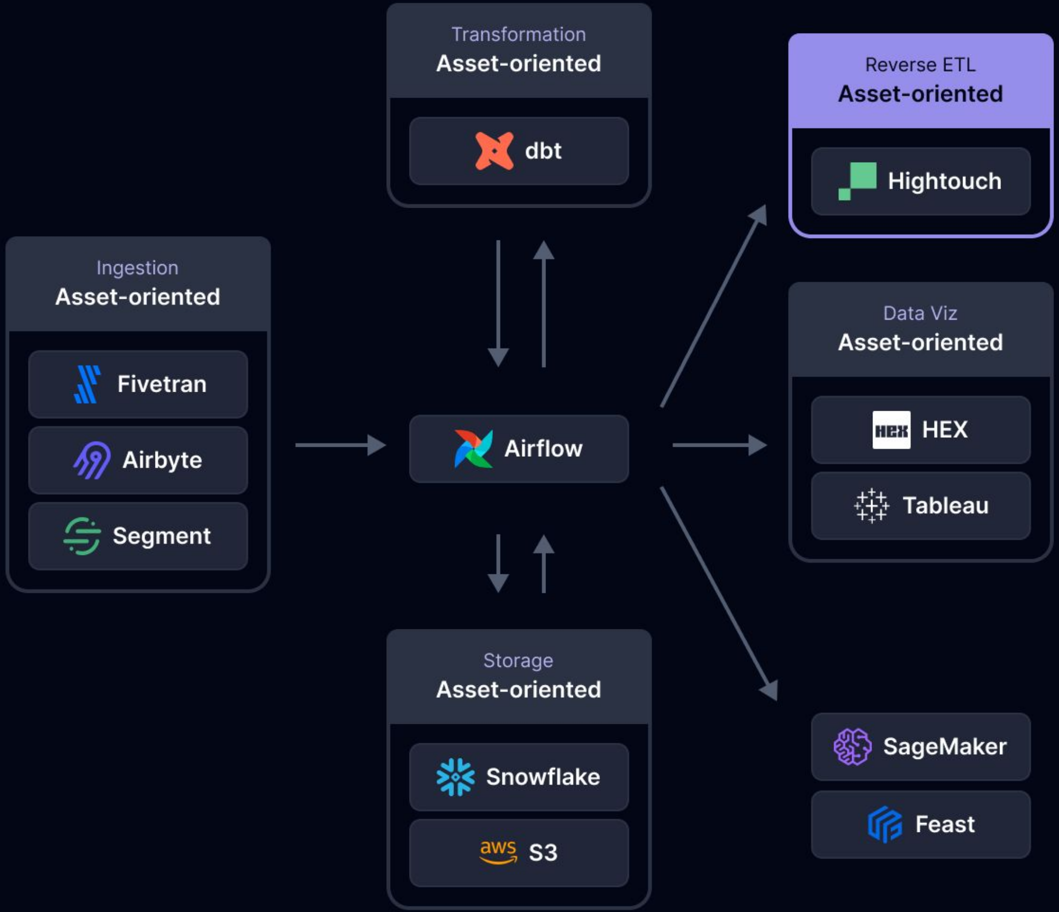

dbt
Hightouch
Fivetran
Airbyte
Segment
Airflow
HEX
Tableau
Snowflake
aws S3
SageMaker
Feast

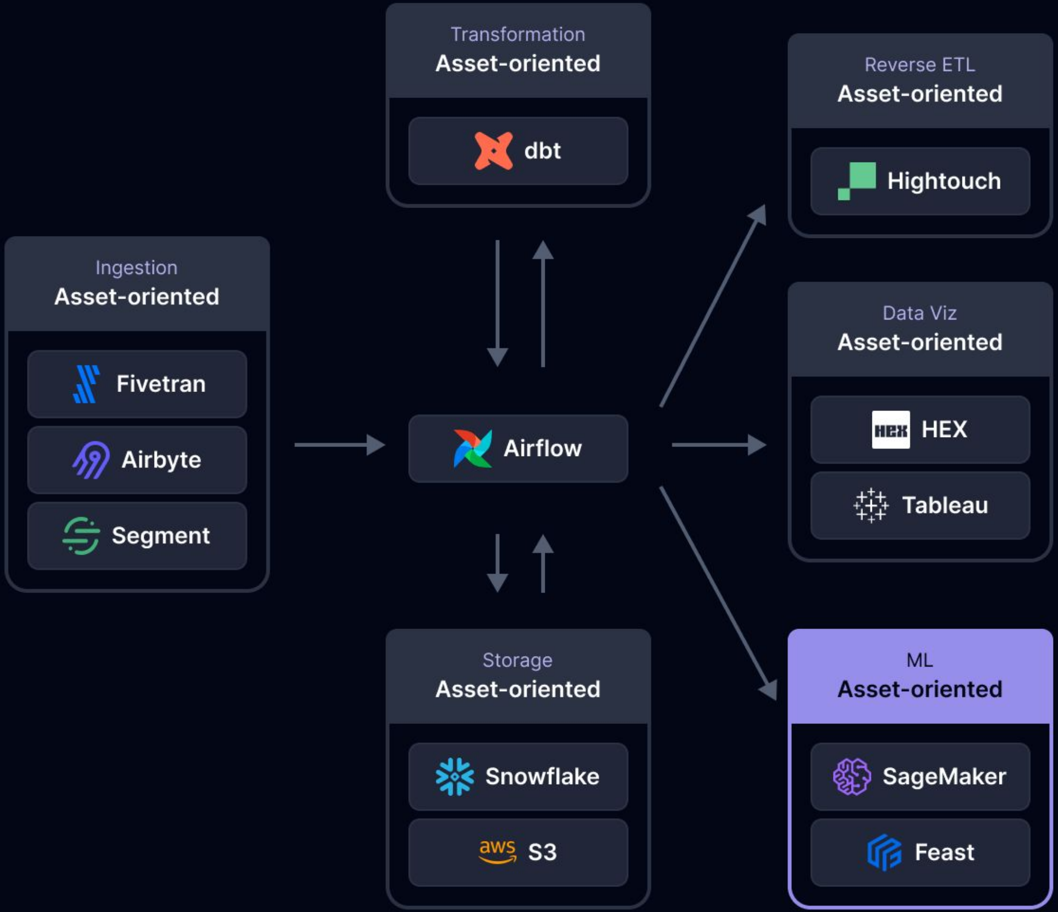

Ingestion
Asset-oriented
Fivetran
Airbyte
Segment
dbt
Airflow
Snowflake
S3
Hightouch
HEX
Tableau
SageMaker
Feast

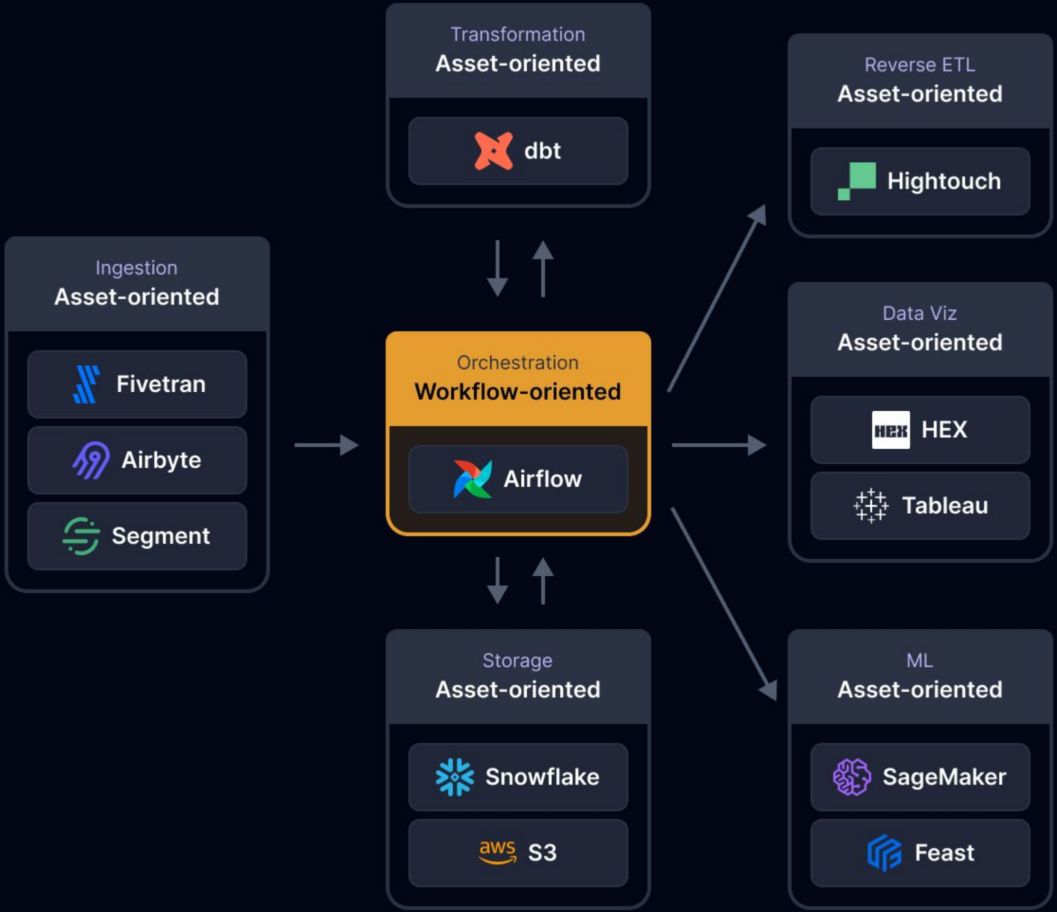

dbt
Hightouch
Ingestion
Asset-oriented
Fivetran
Airbyte
Segment
Airflow
HEX
Tableau
Storage
Asset-oriented
Snowflake
S3
SageMaker
Feast


Transformation
Asset-oriented
dbt
Ingestion
Asset-oriented
Fivetran
Airbyte
Segment
Airflow
Storage
Asset-oriented
Snowflake
aws S3
Hightouch
HEX
Tableau
SageMaker
Feast


Transformation
Asset-oriented
dbt
Hightouch
Ingestion
Asset-oriented
Fivetran
Airbyte
Segment
Airflow
Data Viz
Asset-oriented
HEX
Tableau
Storage
Asset-oriented
Snowflake
aws S3
SageMaker
Feast


Transformation
Asset-oriented
dbt
Reverse ETL
Asset-oriented
Hightouch
Ingestion
Asset-oriented
Fivetran
Airbyte
Segment
Airflow
Data Viz
Asset-oriented
HEX
Tableau
Storage
Asset-oriented
Snowflake
aws S3
SageMaker
Feast


Transformation
Asset-oriented
dbt
Reverse ETL
Asset-oriented
Hightouch
Ingestion
Asset-oriented
Fivetran
Airbyte
Segment
Airflow
Data Viz
Asset-oriented
HEX
Tableau
Storage
Asset-oriented
Snowflake
S3
ML
Asset-oriented
SageMaker
Feast


Transformation
Asset-oriented
dbt
Reverse ETL
Asset-oriented
Hightouch
Ingestion
Asset-oriented
Fivetran
Airbyte
Segment
Orchestration
Workflow-oriented
Airflow
Data Viz
Asset-oriented
HEX
Tableau
Storage
Asset-oriented
Snowflake
S3
ML
Asset-oriented
SageMaker
Feast

What does this mean?

Mismatched programming model

Engineers are more productive when they can think declaratively in terms of desired outputs (assets).

Scheduling inflexibility

Workflow tasks do not have a notion of “freshness” leading to unnecessary spend by excessively rematerializing assets..

Many-to-one relationship between assets and workflow steps

Single workflow tasks like `dbt run` may produce many data assets.

Metadata and observability

The workflow engine has the execution logs, some other system has the schema. A separate data catalog needs to be integrated to join the data together and make it useful for stakeholders

How does the world change when we move from workflow-orientation to asset-orientation?

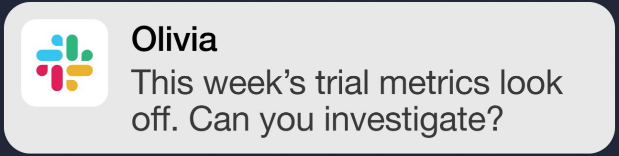
Dev experience

Stack complexity

Ownership

The dev experience
is slow & painful

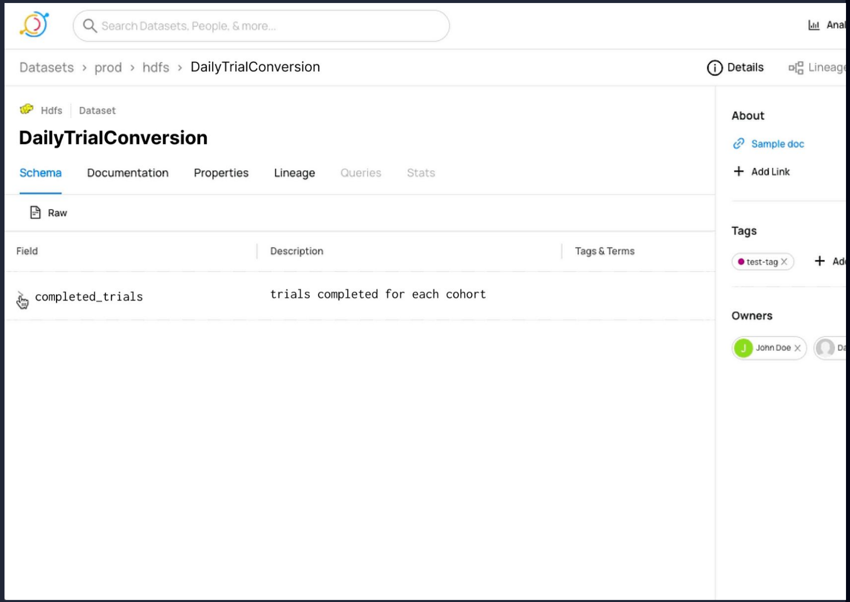
Our hero Luke gets a bug report that the weekly report is missing data.



Olivia

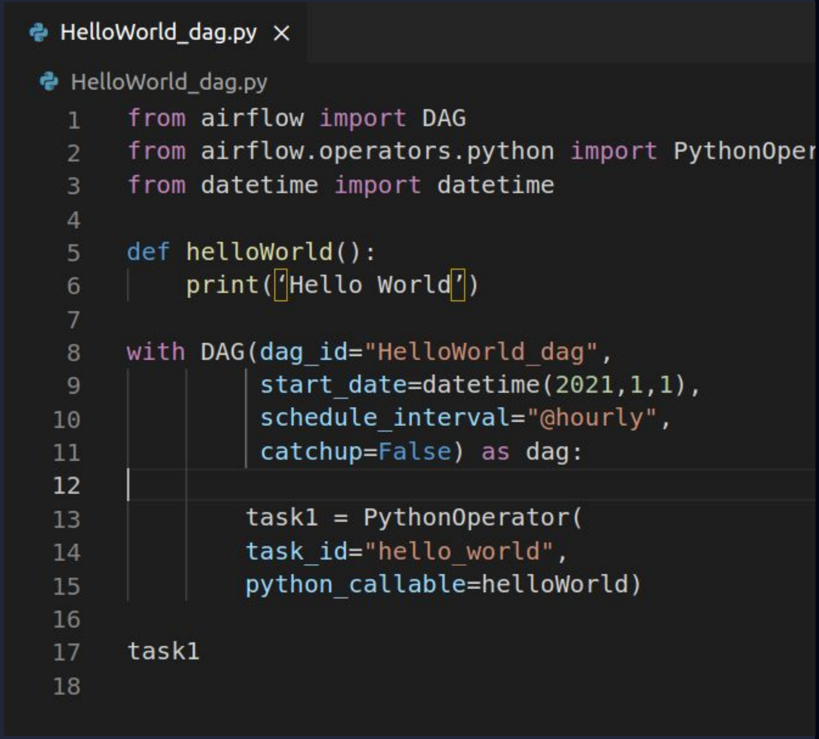
This week’s trial metrics look off. Can you investigate?

He searches around his data catalog, but it was last updated 2 weeks ago because the sync job broke.



Field	Description	Tags & Terms
completed_trials	trials completed for each cohort	

He has to spelunk through the codebase, reading git blame and grepping for every string he can think of to find the code that is related to the problematic data asset.



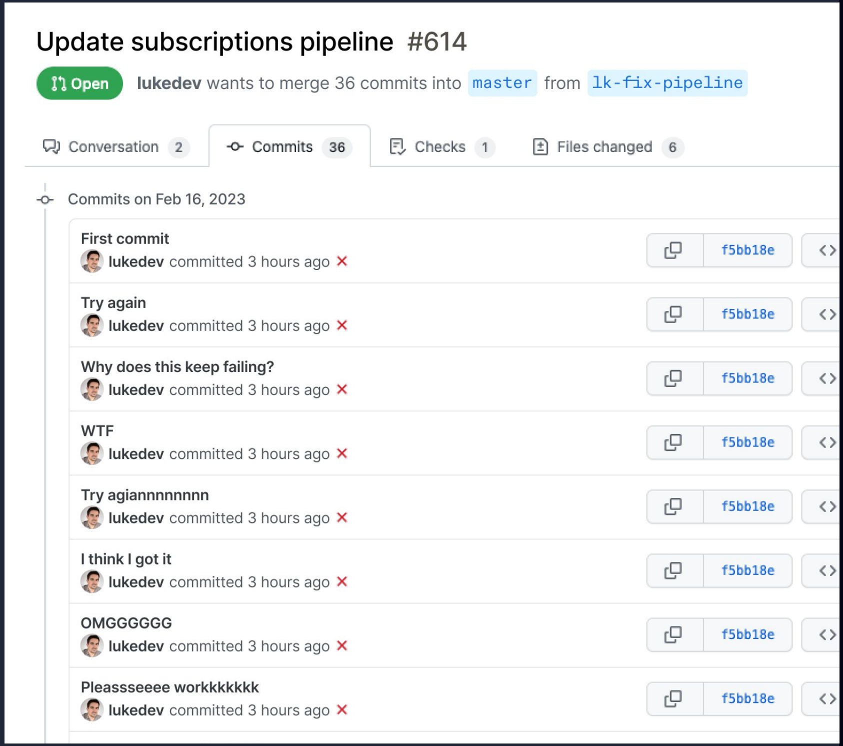
```python
from airflow import DAG
from airflow.operators.python import PythonOper
from datetime import datetime

def helloWorld():
    print(‘Hello World’)

with DAG(dag_id="HelloWorld_dag",
         start_date=datetime(2021,1,1),
         schedule_interval="@hourly",
         catchup=False) as dag:

        task1 = PythonOperator(
        task_id="hello_world",
        python_callable=helloWorld)

task1
```

The dev experience is slow & painful

He pushes it to the staging environment, which takes 15 minutes, to see if it works.

The dev experience is slow & painful

Oops, he made a typo, time to wait another 15 minutes while we push again...



Update subscriptions pipeline #614

Open lukedev wants to merge 36 commits into master from lk-fix-pipeline

Commits on Feb 16, 2023

First commit
lukedev committed 3 hours ago ✗ f5bb18e

Try again
lukedev committed 3 hours ago ✗ f5bb18e

Why does this keep failing?
lukedev committed 3 hours ago ✗ f5bb18e

WTF
lukedev committed 3 hours ago ✗ f5bb18e

Try agiannnnnnnn
lukedev committed 3 hours ago ✗ f5bb18e

I think I got it
lukedev committed 3 hours ago ✗ f5bb18e

OMGGGGGG
lukedev committed 3 hours ago ✗ f5bb18e

Pleassseeee workkkkkkk
lukedev committed 3 hours ago ✗ f5bb18e

How the impedance mismatch caused Luke’s problems

No single source of truth

- The data catalog had to assemble a view of the world using the exhaust of several tools
- Workflow-oriented orchestrator required manual integration to associate metadata with the entry in the catalog
- The integration between the two tools broke

Opaque relationship between data assets and pipeline code

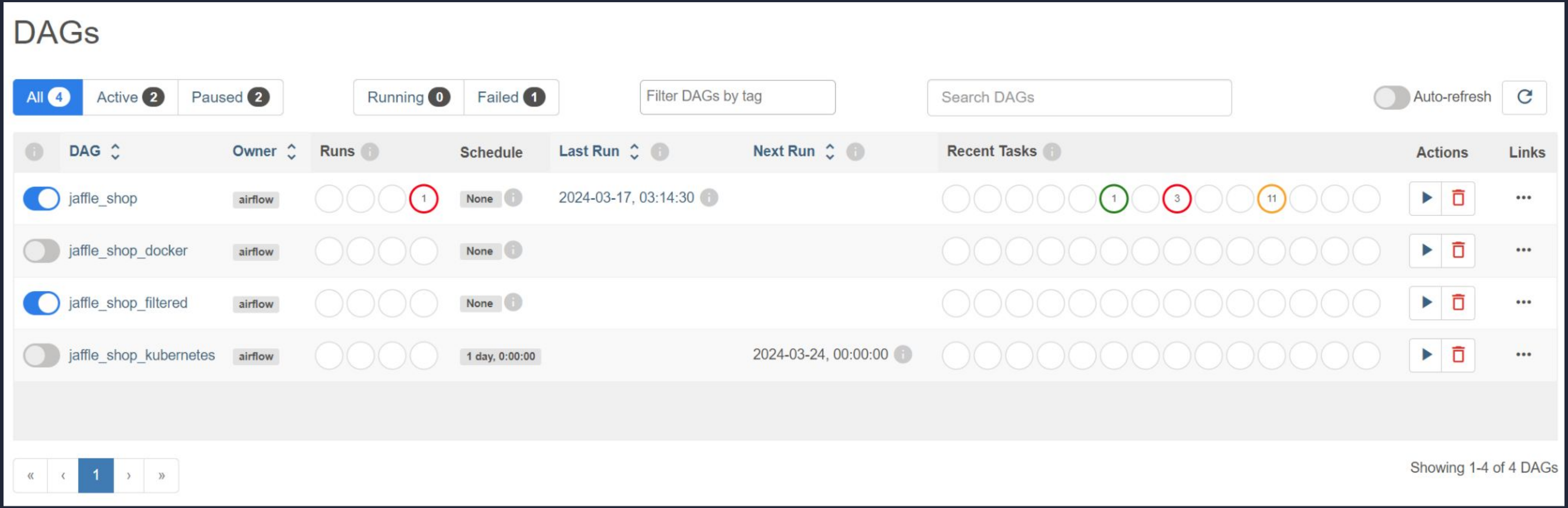
- Data pipelines were written in a workflow-oriented style.
- There was no clear correspondence between the data asset and the workflow task that produced it

Workflow-oriented vs Asset-oriented code

```python
extract_task = PythonOperator
  task_id='extract_data',
  python_callable=extract_data,
  dag=dag,
)

transform_task = PythonOperator(
  task_id='transform_data',
  python_callable=transform_data,
  provide_context=True,
  dag=dag,
)

save_task = PythonOperator(
  task_id='save_to_s3',
  python_callable=save_to_s3,
  op_args=['{{ ti.xcom_pull(task_ids="transform_data") }}'],
  op_kwargs={'execution_date': '{{ ts }}'},
  dag=dag,
)

extract_task > transform_task >> save_task
```

```python
@asset
def clickstream_data():
    extract_data()
    transform_data()
    save_to_s3()
```

Workflow-oriented UI



DAGs

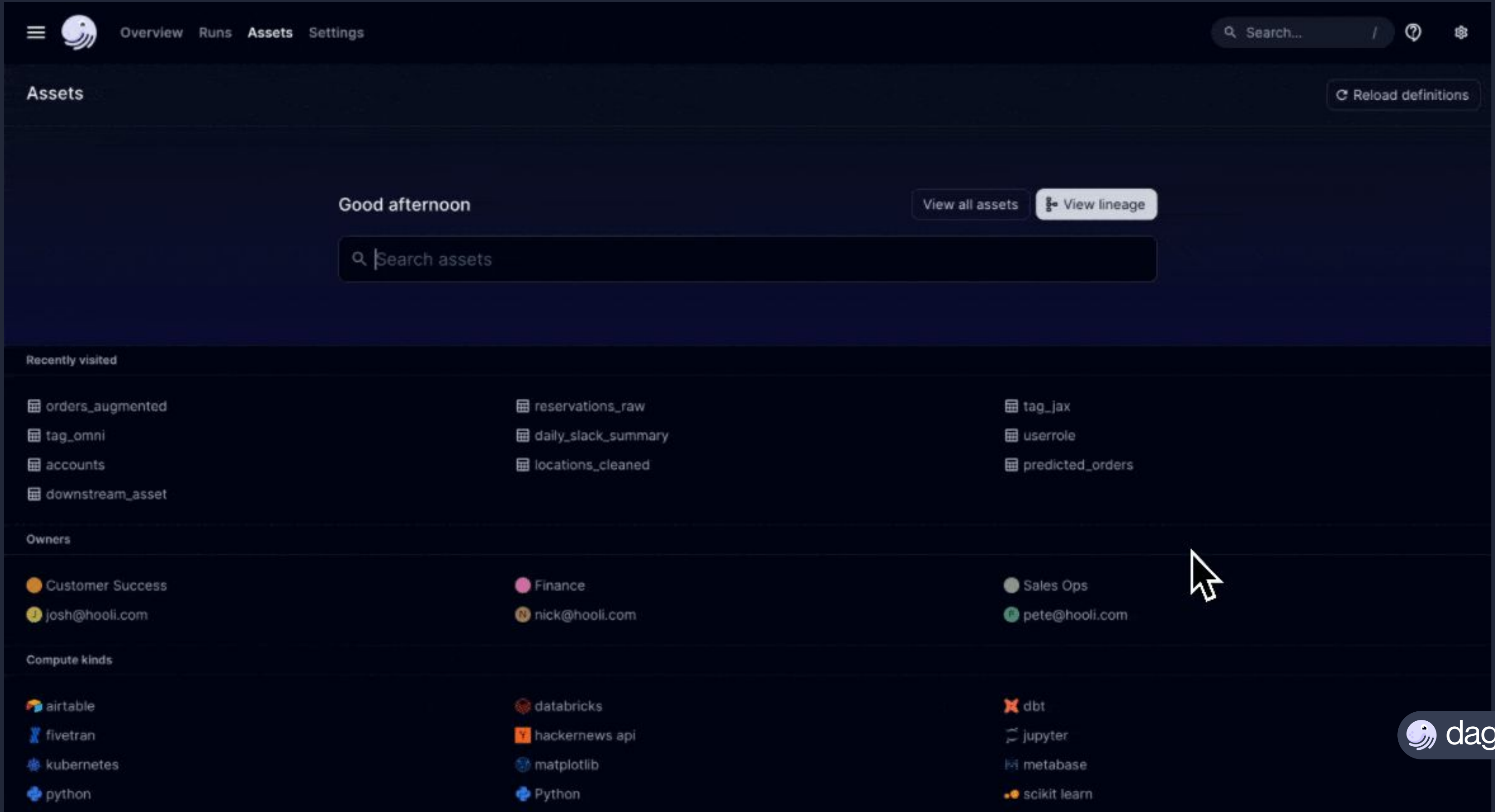
All 4 | Active 2 | Paused 2 | Running 0 | Failed 1 | Filter DAGs by tag | Search DAGs | Auto-refresh

DAG	Owner	Runs	Schedule	Last Run	Next Run	Recent Tasks	Actions	Links
jaffle_shop	airflow	1	None	2024-03-17, 03:14:30		1 3 11		
jaffle_shop_docker	airflow		None					
jaffle_shop_filtered	airflow		None					
jaffle_shop_kubernetes	airflow		1 day, 0:00:00		2024-03-24, 00:00:00			

« ‹ 1 › »

Showing 1-4 of 4 DAGs

Asset-oriented UI - Catalog



Overview Runs Assets Settings

Search... /

Assets

Reload definitions

Good afternoon

View all assets View lineage

Search assets

Recently visited

orders_augmented	reservations_raw	tag_jax
tag_omni	daily_slack_summary	userrole
accounts	locations_cleaned	predicted_orders
downstream_asset		

Owners

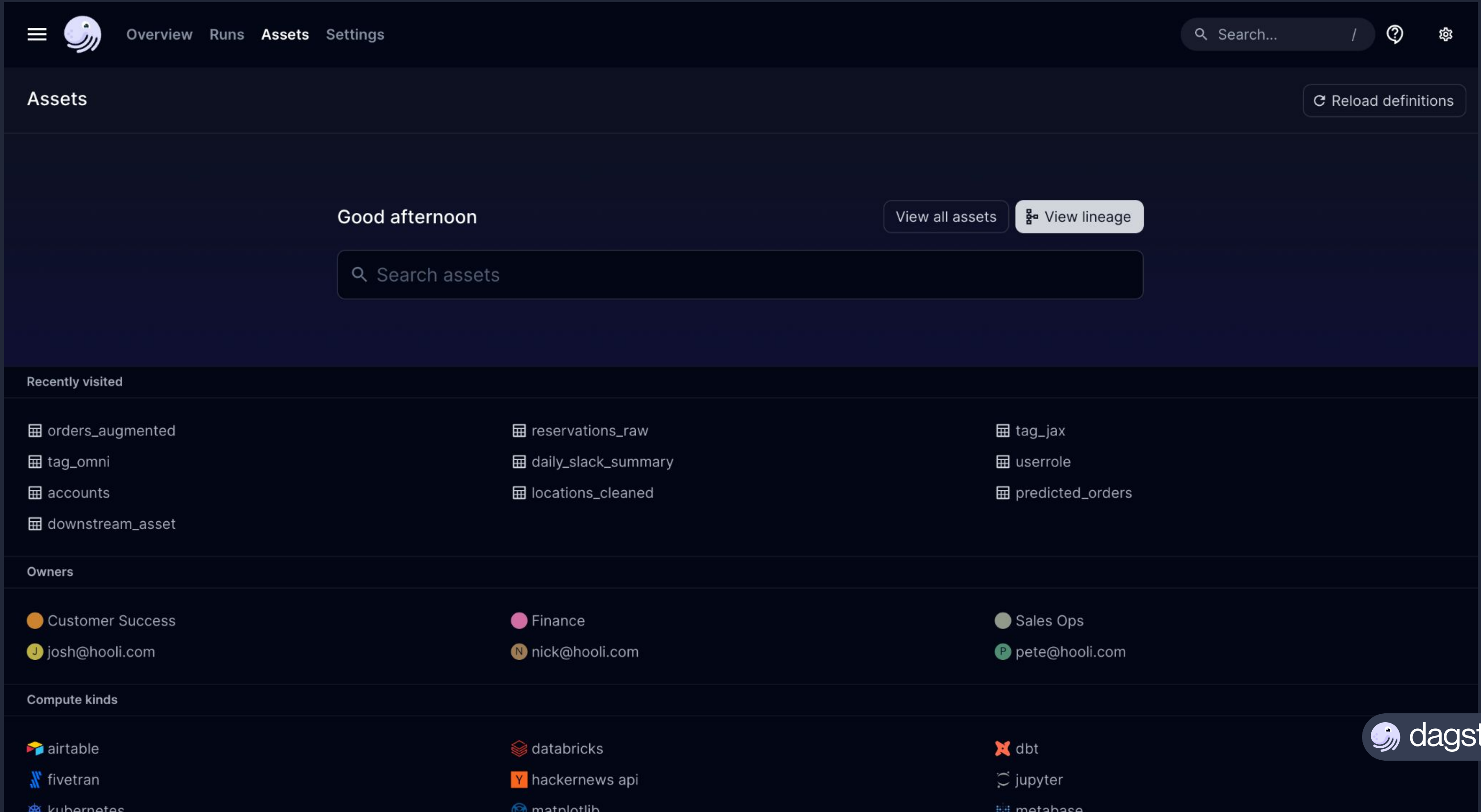
Customer Success	Finance	Sales Ops
josh@hooli.com	nick@hooli.com	pete@hooli.com

Compute kinds

airtable	databricks	dbt
fivetran	hackernews api	jupyter
kubernetes	matplotlib	metabase
python	Python	scikit learn

dagster+

Asset-oriented UI - Catalog



Assets

Good afternoon

Recently visited

orders_augmented	reservations_raw	tag_jax
tag_omni	daily_slack_summary	userrole
accounts	locations_cleaned	predicted_orders
downstream_asset		

Owners

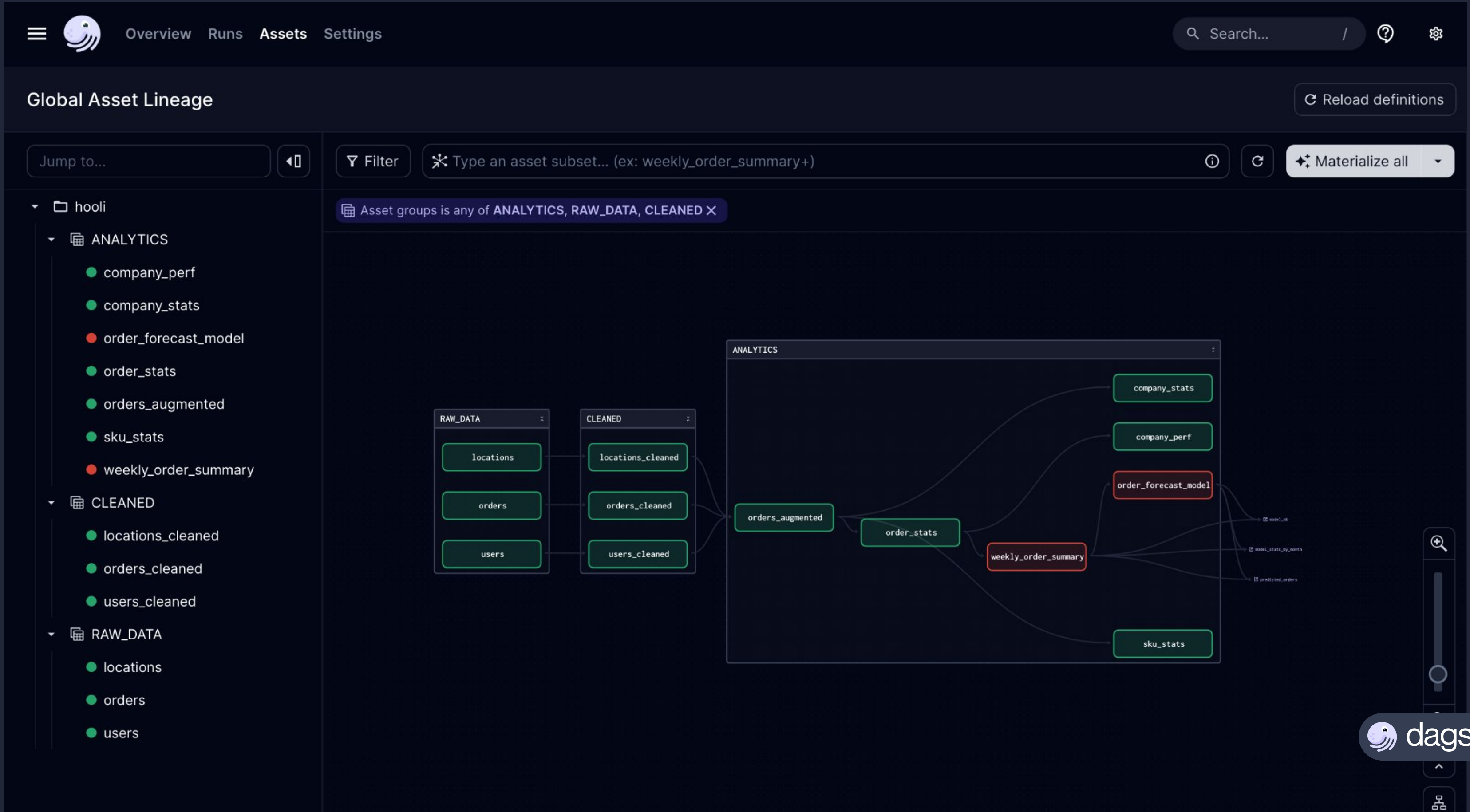
Customer Success	Finance	Sales Ops
josh@hooli.com	nick@hooli.com	pete@hooli.com

Compute kinds

airtable	databricks	dbt
fivetran	hackernews api	jupyter
kubernetes	matplotlib	metabase

dagster+

Asset-oriented UI - lineage

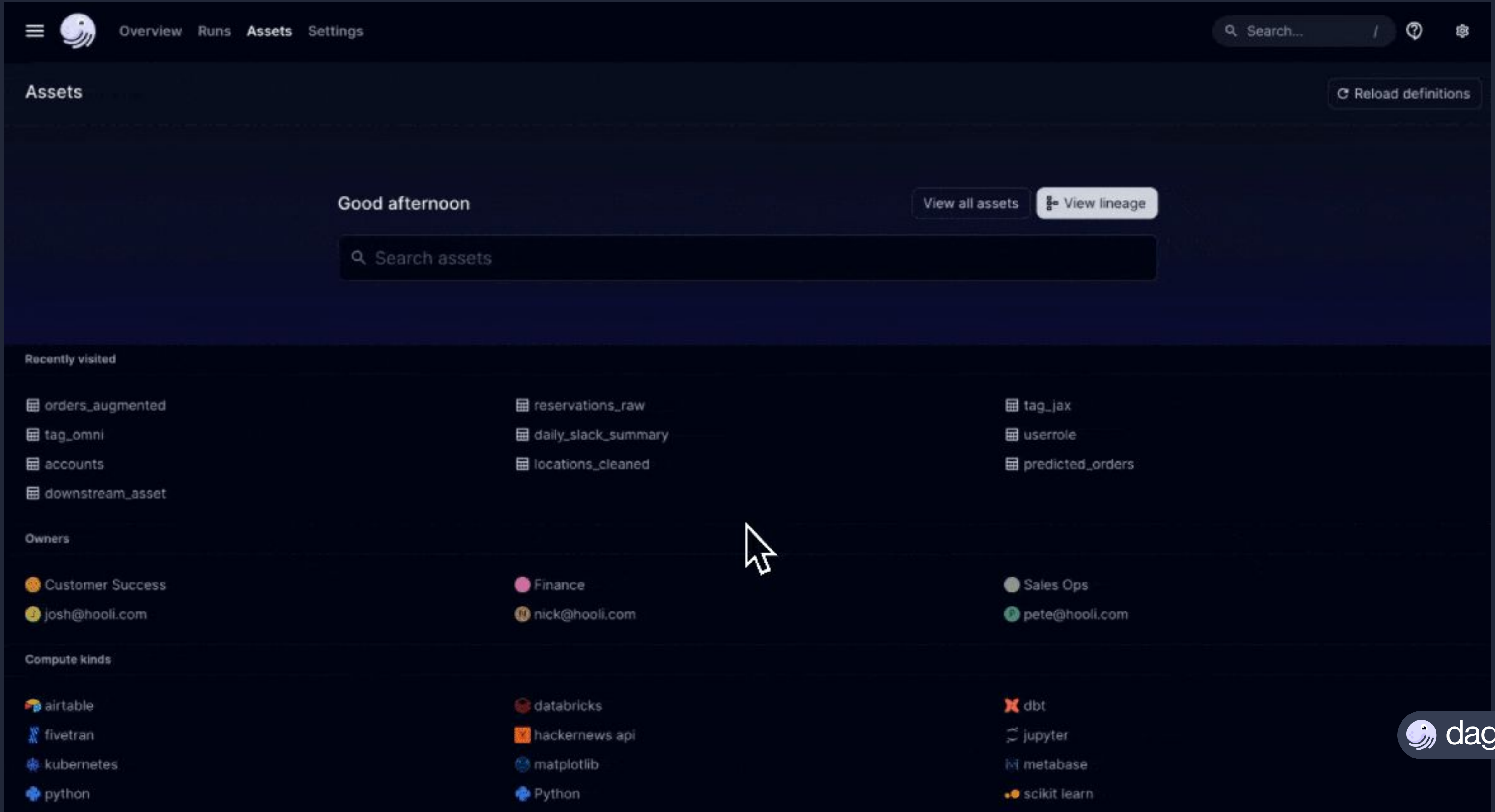


Global Asset Lineage

Asset groups is any of **ANALYTICS**, **RAW_DATA**, **CLEANED**

- hooli
  - ANALYTICS
    - company_perf
    - company_stats
    - order_forecast_model
    - order_stats
    - orders_augmented
    - sku_stats
    - weekly_order_summary
  - CLEANED
    - locations_cleaned
    - orders_cleaned
    - users_cleaned
  - RAW_DATA
    - locations
    - orders
    - users

Asset-oriented UI - lineage



Assets

Good afternoon

Recently visited

orders_augmented	reservations_raw	tag_jax
tag_omni	daily_slack_summary	userrole
accounts	locations_cleaned	predicted_orders
downstream_asset		

Owners

Customer Success	Finance	Sales Ops
josh@hooli.com	nick@hooli.com	pete@hooli.com

Compute kinds

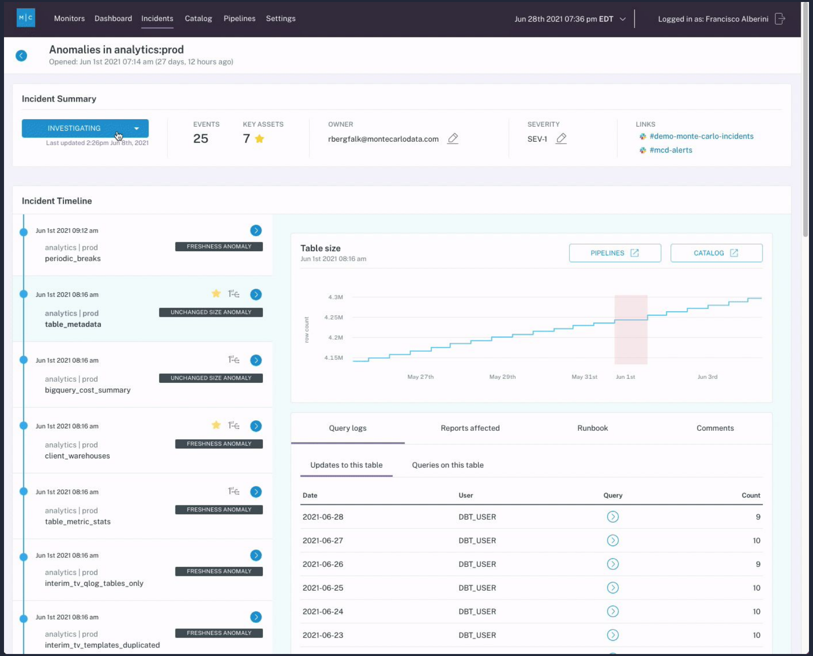
airtable	databricks	dbt
fivetran	hackernews api	jupyter
kubernetes	matplotlib	metabase
python	Python	scikit learn

dagster+

The tech stack is complex & expensive

Luke's customers, data pipeline authors, want to be alerted if their data fails quality checks.

- He negotiates a deal with a vendor
- Asks his stakeholders to tag all of their queries
- They say: "Q1 2025"



Anomalies in analytics:prod

Opened: Jun 1st 2021 07:14 am (27 days, 12 hours ago)

Incident Summary

	EVENTS	KEY ASSETS	OWNER	SEVERITY	LINKS
INVESTIGATING Last updated 2:26pm Jun 8th, 2021	25	7	rbergfalk@montecarlodata.com	SEV-1	#demo-monte-carlo-incidents #mcd-alerts

Incident Timeline

- Jun 1st 2021 09:12 am — analytics | prod — periodic_breaks — FRESHNESS ANOMALY
- Jun 1st 2021 08:16 am — analytics | prod — table_metadata — UNCHANGED SIZE ANOMALY
- Jun 1st 2021 08:16 am — analytics | prod — bigquery_cost_summary — UNCHANGED SIZE ANOMALY
- Jun 1st 2021 08:16 am — analytics | prod — client_warehouses — FRESHNESS ANOMALY
- Jun 1st 2021 08:16 am — analytics | prod — table_metric_stats — FRESHNESS ANOMALY
- Jun 1st 2021 08:16 am — analytics | prod — interim_tv_qlog_tables_only — FRESHNESS ANOMALY
- Jun 1st 2021 08:16 am — analytics | prod — interim_tv_templates_duplicated — FRESHNESS ANOMALY

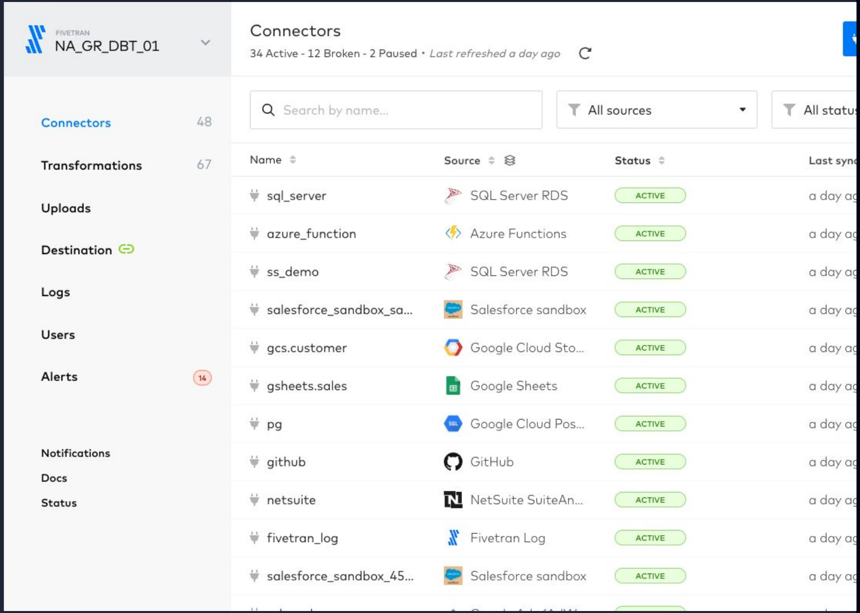
Table size

Jun 1st 2021 08:16 am

Date	User	Query	Count
2021-06-28	DBT_USER		9
2021-06-27	DBT_USER		10
2021-06-26	DBT_USER		9
2021-06-25	DBT_USER		10
2021-06-24	DBT_USER		10
2021-06-23	DBT_USER		10

Luke’s customers also want to move some data around.

- He negotiates a deal with an ELT vendor
- Writes a custom operator for his stakeholders to use
- Needs to remember to wire it up to every other tool in the stack


FIVETRAN
NA_GR_DBT_01

Connectors 48
Transformations 67
Uploads
Destination
Logs
Users
Alerts 14

Notifications
Docs
Status

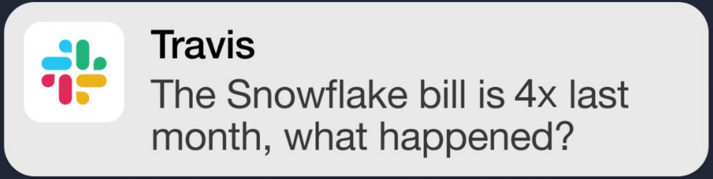
Connectors
34 Active - 12 Broken - 2 Paused • *Last refreshed a day ago*

Search by name...
All sources
All statu

Name	Source	Status	Last sync
sql_server	SQL Server RDS	ACTIVE	a day ag
azure_function	Azure Functions	ACTIVE	a day ag
ss_demo	SQL Server RDS	ACTIVE	a day ag
salesforce_sandbox_sa...	Salesforce sandbox	ACTIVE	a day ag
gcs.customer	Google Cloud Sto...	ACTIVE	a day ag
gsheets.sales	Google Sheets	ACTIVE	a day ag
pg	Google Cloud Pos...	ACTIVE	a day ag
github	GitHub	ACTIVE	a day ag
netsuite	NetSuite SuiteAn...	ACTIVE	a day ag
fivetran_log	Fivetran Log	ACTIVE	a day ag
salesforce_sandbox_45...	Salesforce sandbox	ACTIVE	a day ag

Luke gets the Snowflake bill for the month and it's up 400%

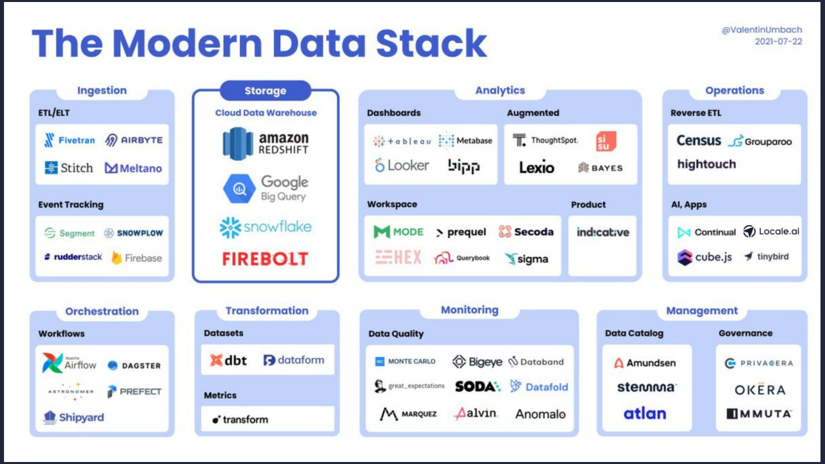
- He looks at the query_history table and sees its driven by a query that has no attribution information
- He greps through the codebase and eventually finds the code that issues the query
- He fixes the bug, but asks his customers to tag their queries for next time.
- They say: "Q1 2025"



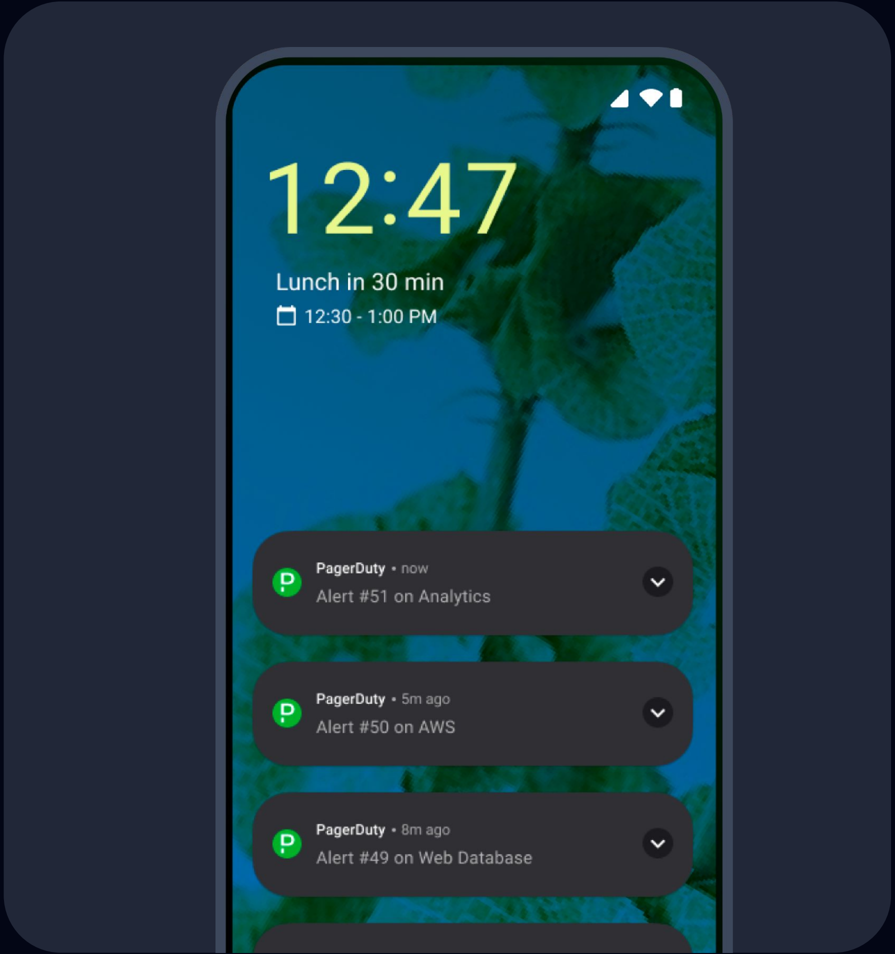
Travis

The Snowflake bill is 4x last month, what happened?

By the end of the year, he's bought 20 different data tools and runs 10 new OSS services.



Oh, and his platform eng counterparts just got laid off, so he has to carry the pager every week now.



12:47

Lunch in 30 min
12:30 - 1:00 PM

PagerDuty • now
Alert #51 on Analytics

PagerDuty • 5m ago
Alert #50 on AWS

PagerDuty • 8m ago
Alert #49 on Web Database

Why the impedance mismatch made Luke buy more tools

Disconnected systems

Because Fivetran and Monte Carlo are asset-oriented tools and Airflow is not, Luke had to prod his stakeholders to do an expensive, manual integration phase for each data asset.

Manual integration

Similarly, he was unable to attribute Snowflake spend to specific teams or assets without hours or days of sifting through the codebase, or asking his stakeholders to, again, manually tag each one of their queries.

Poor observability

Because Airflow is a workflow engine, it does not have any built-in features that support asset-oriented capabilities like data observability, cost management and data discovery, necessitating the purchase, integration and maintenance of numerous point solutions.

Why the impedance mismatch made Luke buy more tools

Manual tagging of queries

- Airflow is workflow-oriented, and has no knowledge of which tasks correspond to which data assets

- For this reason, Airflow can’t help Luke’s customers with this problem, so they must tag their queries with asset attribution manually, which is expensive and fragile.

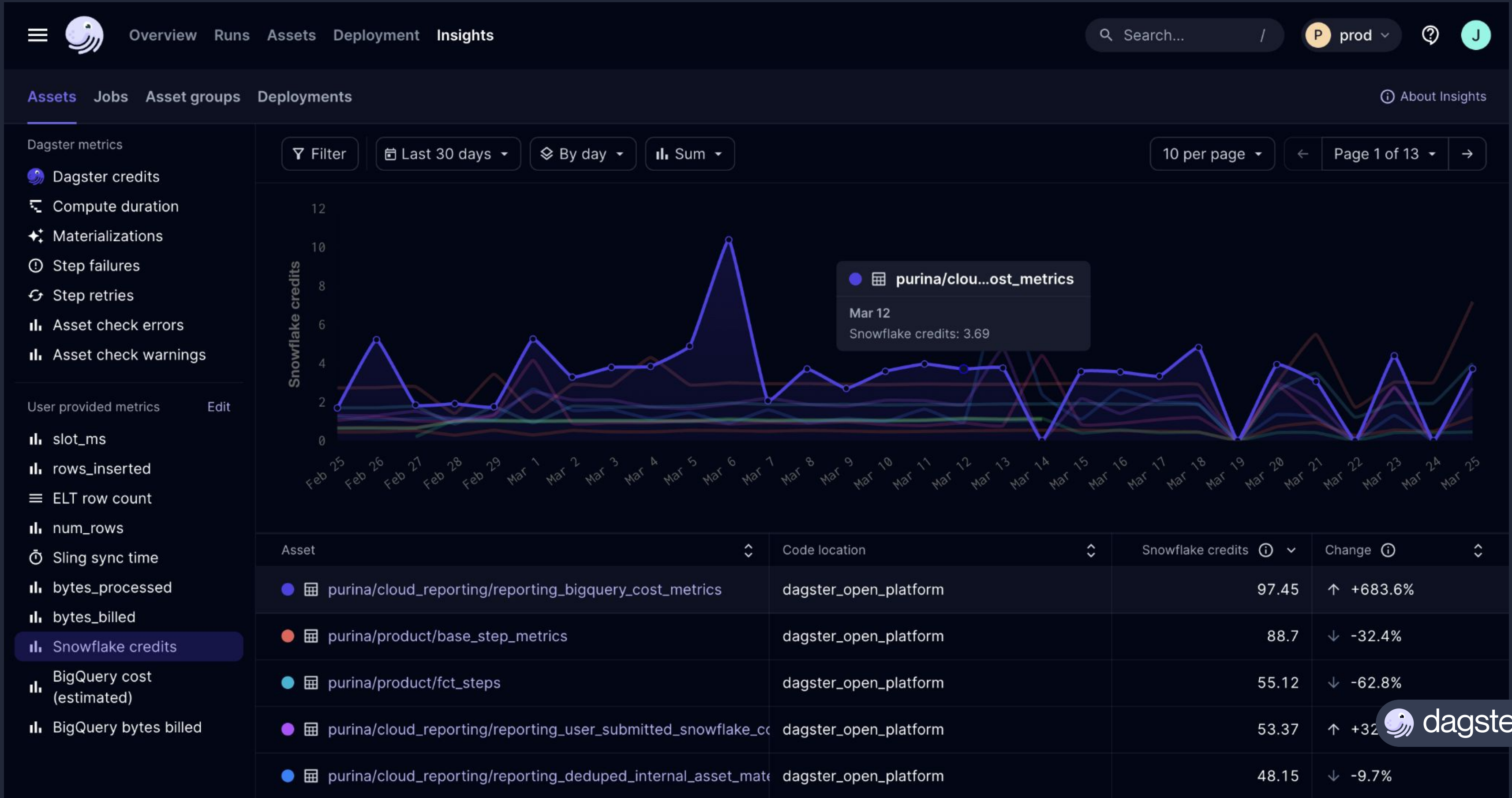
No data observability

- As tasks in workflow engines are black boxes, Airflow has no knowledge of the data they are operating on.

- Thus, Airflow’s observability is limited to high-level cluster and workflow health.

- There is no visibility into the data itself, necessitating manual integration of point solutions.

Cost control with Dagster Insights



Dagster metrics

- Dagster credits
- Compute duration
- Materializations
- Step failures
- Step retries
- Asset check errors
- Asset check warnings

User provided metrics Edit

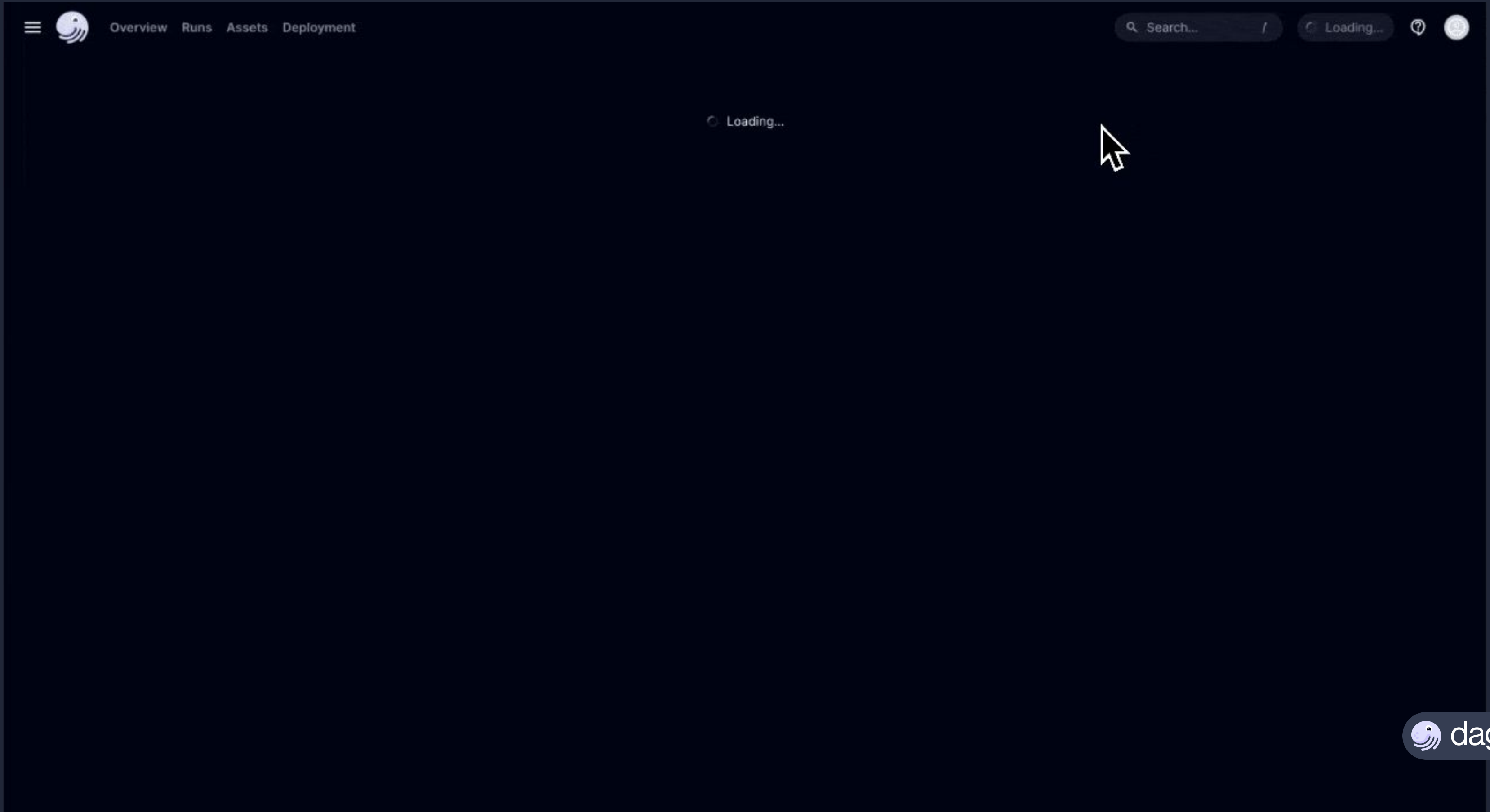
- slot_ms
- rows_inserted
- ELT row count
- num_rows
- Sling sync time
- bytes_processed
- bytes_billed
- Snowflake credits
- BigQuery cost (estimated)
- BigQuery bytes billed

purina/clou...ost_metrics
Mar 12
Snowflake credits: 3.69

Asset	Code location	Snowflake credits	Change
purina/cloud_reporting/reporting_bigquery_cost_metrics	dagster_open_platform	97.45	↑ +683.6%
purina/product/base_step_metrics	dagster_open_platform	88.7	↓ -32.4%
purina/product/fct_steps	dagster_open_platform	55.12	↓ -62.8%
purina/cloud_reporting/reporting_user_submitted_snowflake_co	dagster_open_platform	53.37	↑ +32
purina/cloud_reporting/reporting_deduped_internal_asset_mate	dagster_open_platform	48.15	↓ -9.7%

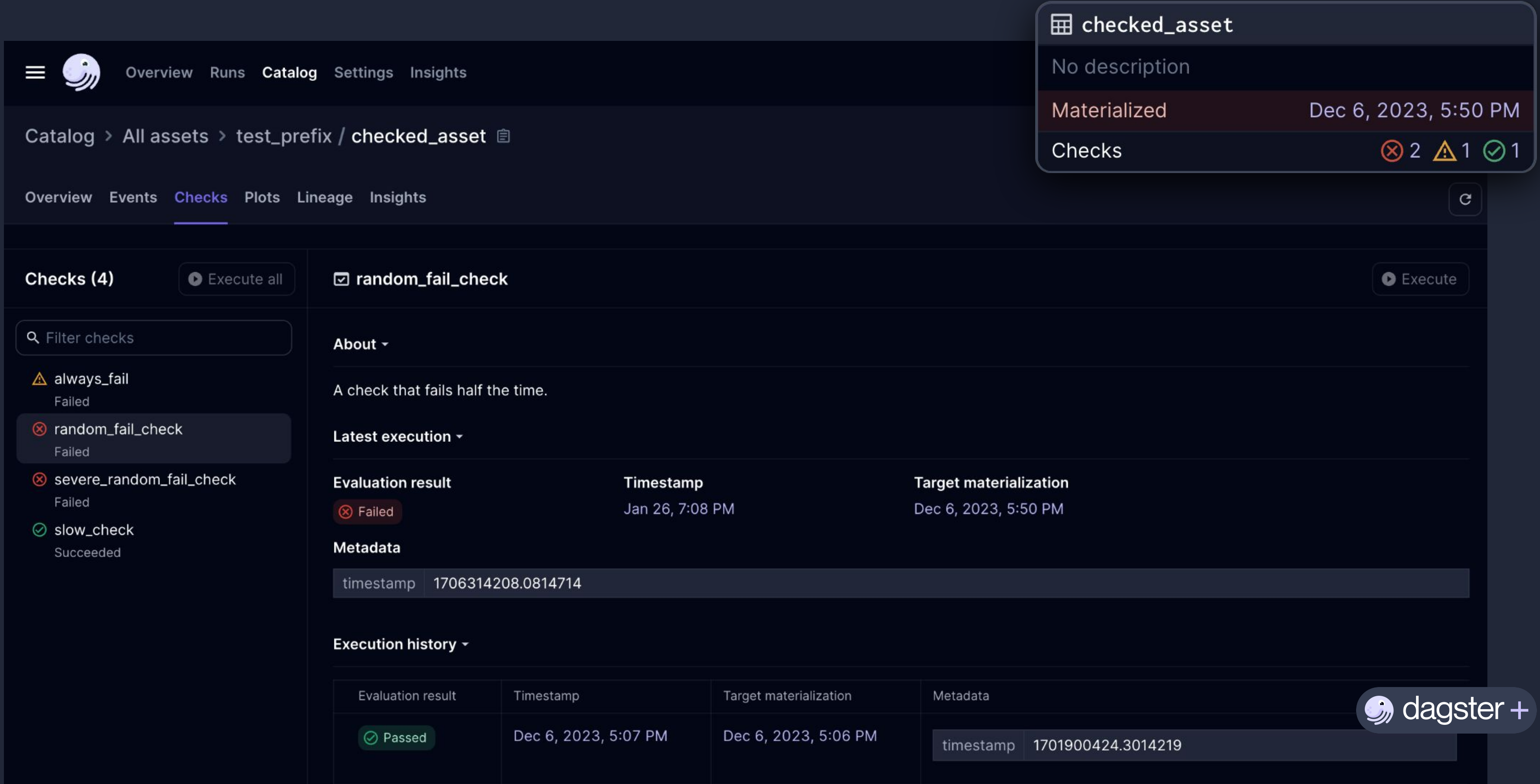
dagster+

Cost control with Dagster Insights



dagster+

Data Quality checks with Dagster



checked_asset

No description

Materialized Dec 6, 2023, 5:50 PM

Checks 2 1 1

Overview Runs Catalog Settings Insights

Catalog > All assets > test_prefix / checked_asset

Overview Events Checks Plots Lineage Insights

Checks (4) Execute all

Filter checks

always_fail
Failed

random_fail_check
Failed

severe_random_fail_check
Failed

slow_check
Succeeded

random_fail_check Execute

About

A check that fails half the time.

Latest execution

Evaluation result	Timestamp	Target materialization
Failed	Jan 26, 7:08 PM	Dec 6, 2023, 5:50 PM

Metadata

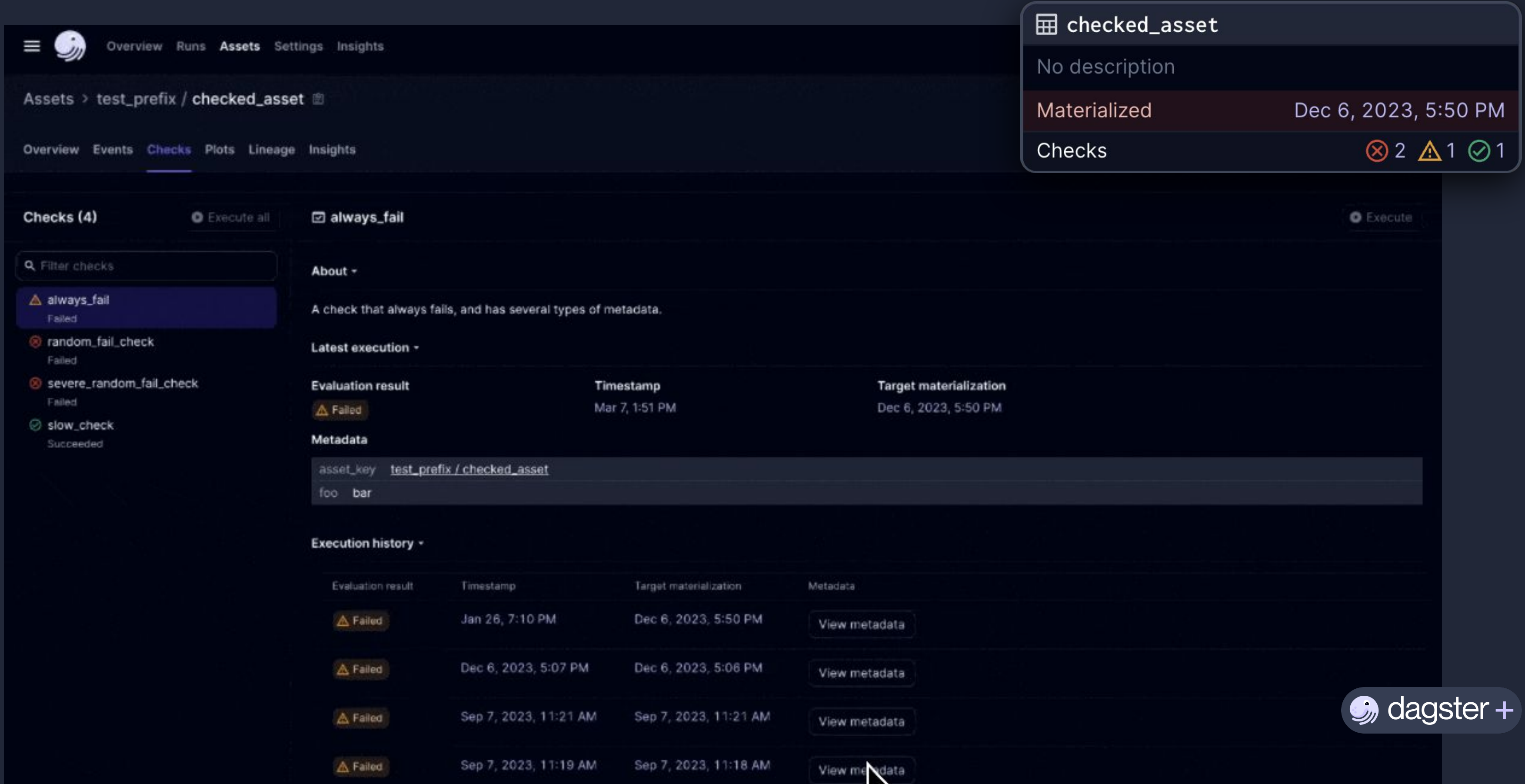
timestamp	1706314208.0814714

Execution history

Evaluation result	Timestamp	Target materialization	Metadata
Passed	Dec 6, 2023, 5:07 PM	Dec 6, 2023, 5:06 PM	timestamp 1701900424.3014219

dagster+

Data Quality checks with Dagster



Overview Runs Assets Settings Insights

Assets > test_prefix / checked_asset

Overview Events Checks Plots Lineage Insights

Checks (4) Execute all

Filter checks

- always_fail — Failed
- random_fail_check — Failed
- severe_random_fail_check — Failed
- slow_check — Succeeded

always_fail Execute

About

A check that always fails, and has several types of metadata.

Latest execution

Evaluation result	Timestamp	Target materialization
Failed	Mar 7, 1:51 PM	Dec 6, 2023, 5:50 PM

Metadata

asset_key	test_prefix / checked_asset
foo	bar

Execution history

Evaluation result	Timestamp	Target materialization	Metadata
Failed	Jan 26, 7:10 PM	Dec 6, 2023, 5:50 PM	View metadata
Failed	Dec 6, 2023, 5:07 PM	Dec 6, 2023, 5:06 PM	View metadata
Failed	Sep 7, 2023, 11:21 AM	Sep 7, 2023, 11:21 AM	View metadata
Failed	Sep 7, 2023, 11:19 AM	Sep 7, 2023, 11:18 AM	View metadata

checked_asset

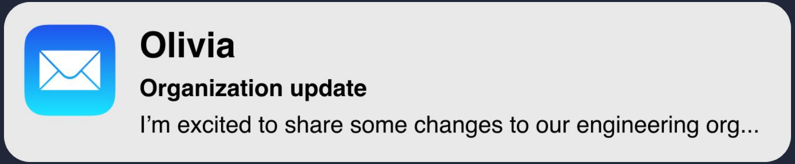
No description

Materialized	Dec 6, 2023, 5:50 PM
Checks	⊗ 2 ⚠ 1 ✓ 1

dagster+

Ownership is chaotic & complicated

All of the data teams got reorged, and now workflows are shared between multiple teams.

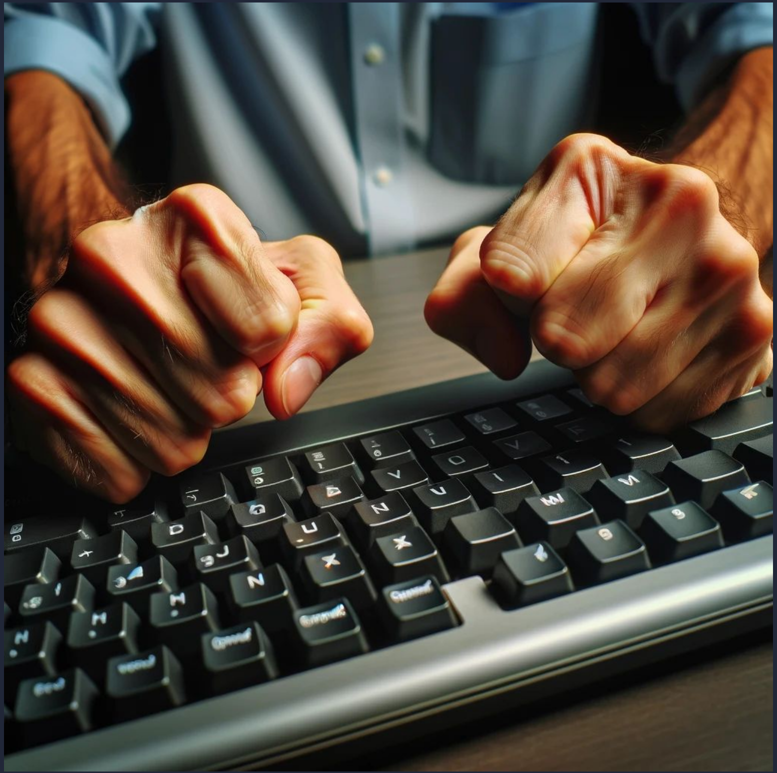


Olivia

Organization update

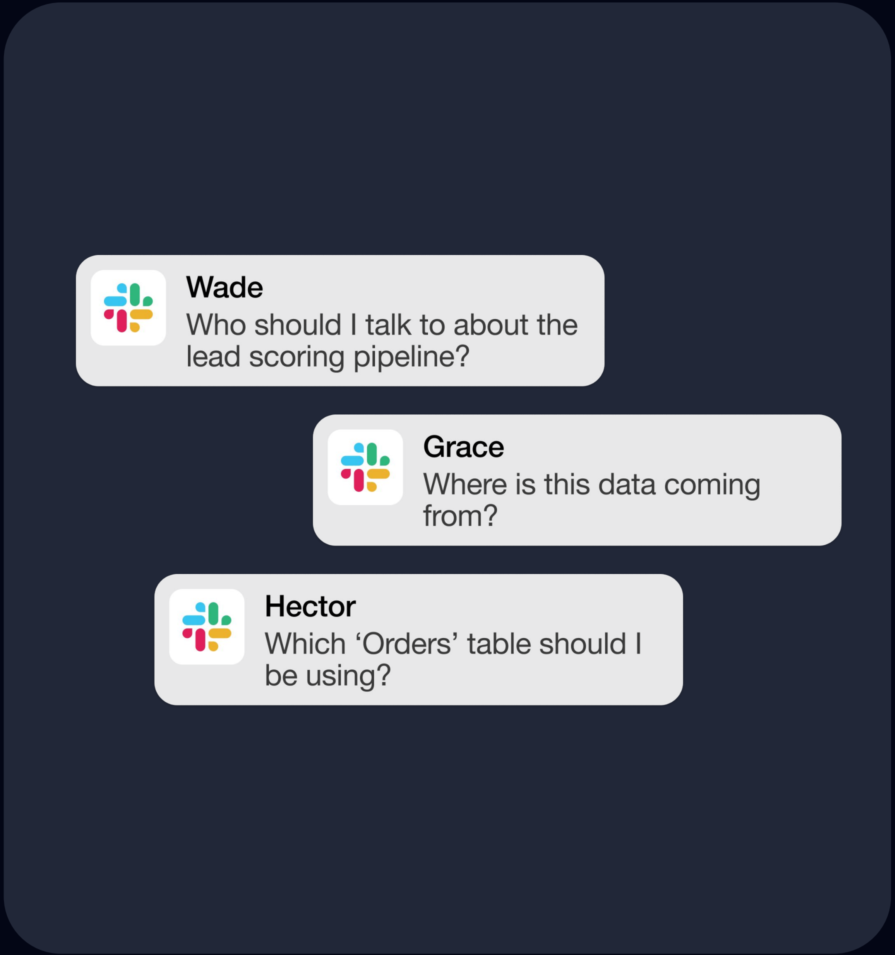
I’m excited to share some changes to our engineering org...

Teams start to step on each others' toes technically and socially.



They eventually conclude that it's easier to spin up separate infrastructure for their workflows than collaborate on a single instance.

Because there are multiple owners and Airflow instances, stakeholders are more confused than ever. The #help-data Slack channel is chaos.

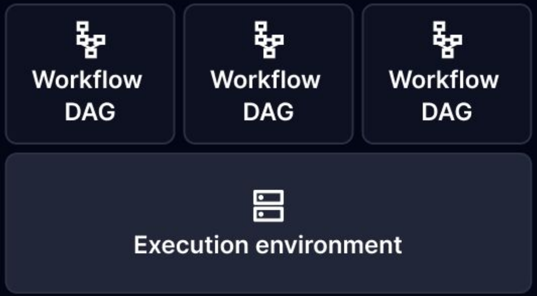


Wade
Who should I talk to about the lead scoring pipeline?

Grace
Where is this data coming from?

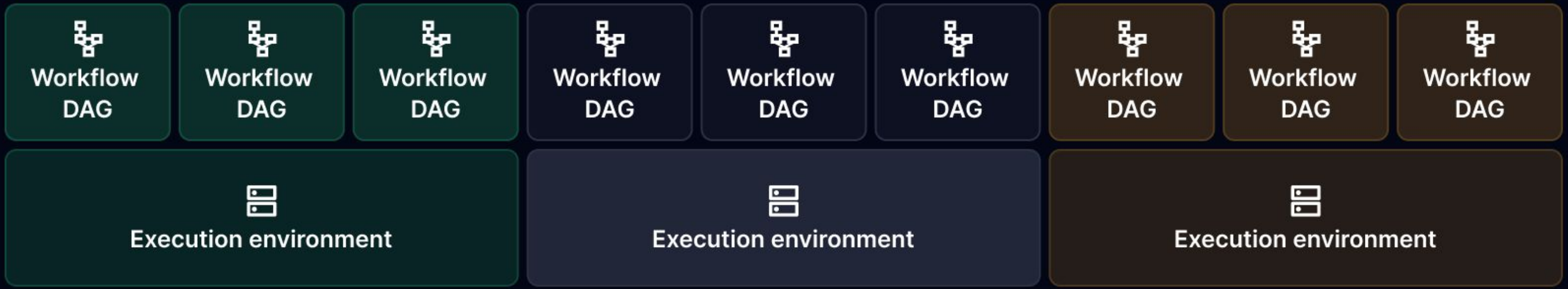
Hector
Which ‘Orders’ table should I be using?

Why the impedance mismatch made ownership complicated



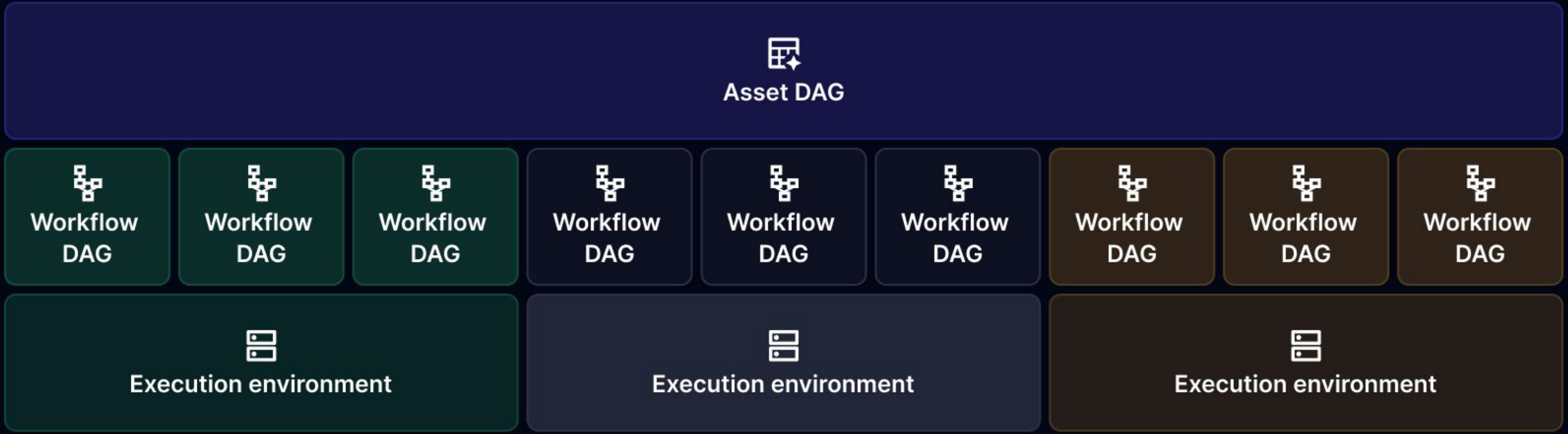
Workflow-oriented orchestration (Airflow, others)

Why the impedance mismatch made ownership complicated



Workflow-oriented orchestration (Airflow, others)

Why the impedance mismatch made ownership complicated



Asset-oriented orchestration adds an overlay layer

Wrapping up

- There is an impedance mismatch between **workflow-oriented** orchestrators and the rest of the (**asset-oriented**) data platform.
- This impedance mismatch causes problems with:
  - Developer experience
  - Stack complexity
  - Collaboration
- There is a way out

Thanks!

- @floydophone on Twitter
- linkedin.com/in/pwhunt on LinkedIn
- dagster.io for Dagster
- dagster+

Notes from Schrockn

- Introduce that the orchestrator needs to be the central tool in the stack
    - Orchestrator is asset oriented
    - Data engineers need to live in it / uber DAG
    - Needs to be designed for the SDLC
    - Uber dag
        - “Everyone’s saying it”
- Dev workflow doesn’t connect to asset orientation, or does not land
    - Cut it
    - Pivot to just about going from data asset to code and back again.
    - Intuitive, no centralized uberdag, basis for making the system of record for data assets
- Simply slide #59 re: monte carlo, make more concrete
- Cut some vendor names
- Create an “after” picture of the system diagram. “It changes the game”