



**OpenLineage &  
Airflow - data lineage  
has never been easier**



May 2022

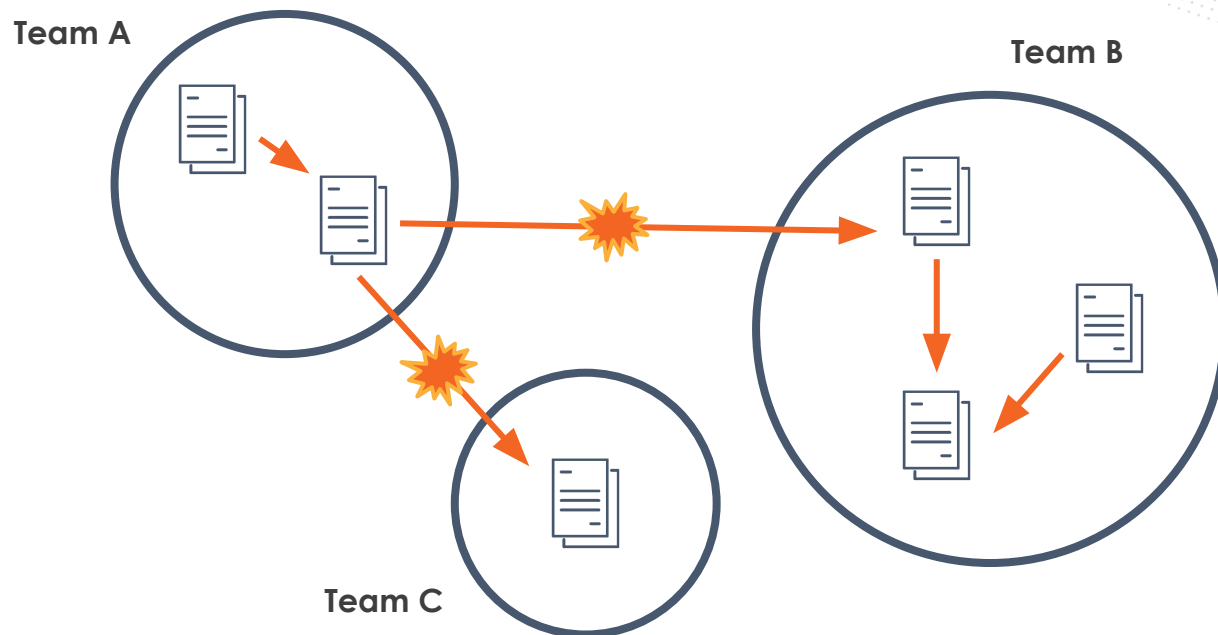
Maciej



Paweł



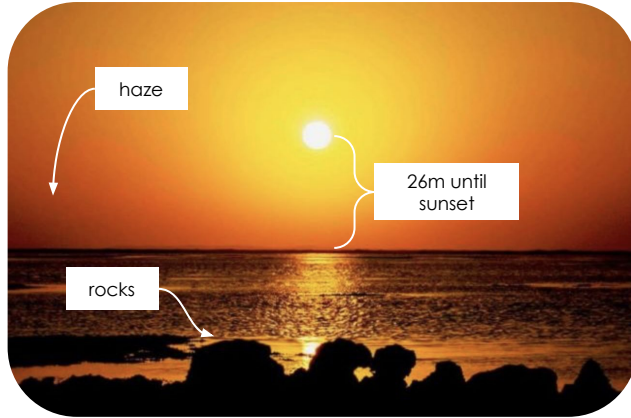
# OpenLineage to build a healthy data ecosystem



## Interesting questions:

- What is the data source?
- What is the schema?
- Who is the owner?
- How often is it updated?
- Where does it come from?
- Who is using it?
- What has changed?

# Infer or observe?



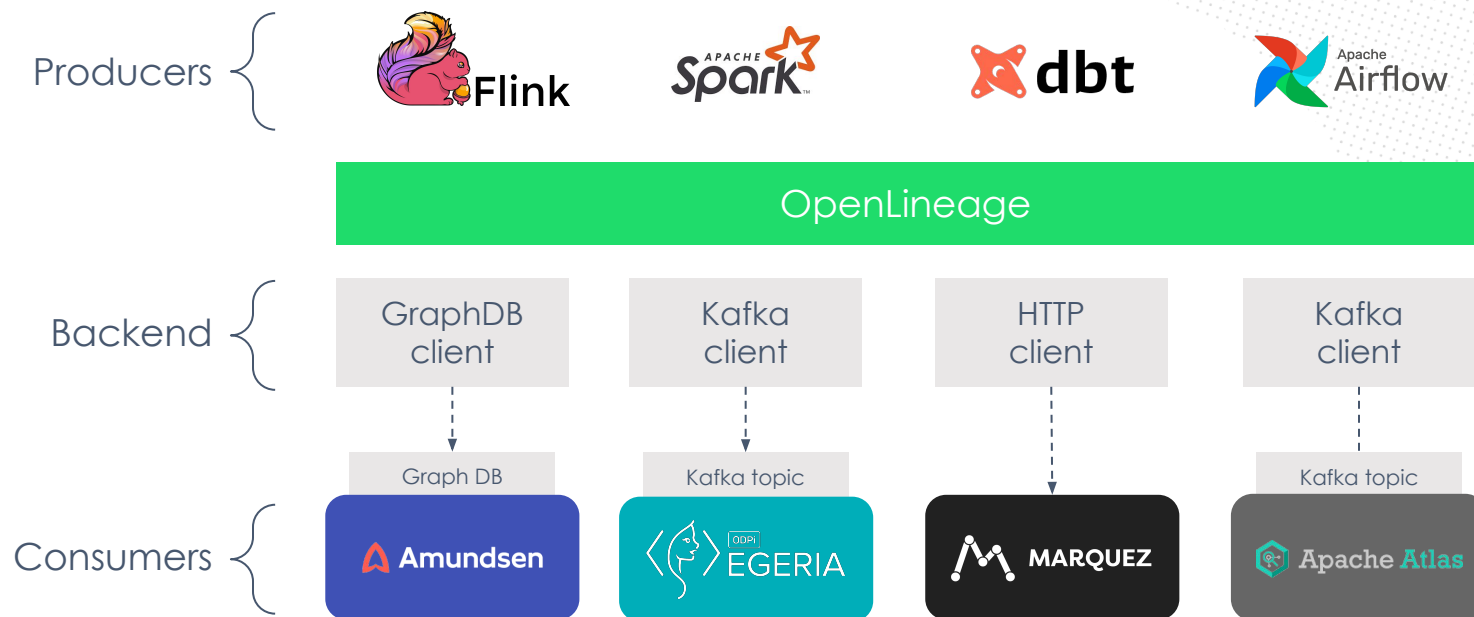
You can try to infer the date and location of an image after the fact...



...or you can capture it when the image is originally created!

# OpenLineage mission

- To define an open standard for the collection of lineage metadata from pipelines as they are running.



# The dark past of Airflow Integration

- We want to achieve real-time notification about TaskInstance start, success, fail
- First way to do it? Subclassing DAG
  - `from airflow import DAG`
  - + `from openlineage.airflow import DAG`
- We can overload DAG methods and get notifications this way.
- Modify all the dags, have to set up openlineage-airflow locally.

# The dark past of Airflow Integration

- We want to achieve real-time notification about TaskInstance start, success, fail
- First way to do it? Subclassing DAG
  - from airflow import DAG
  - + from openlineage.airflow import DAG
- We can overload DAG methods and get notifications this way.
- Modify all the dags, have to set up openlineage-airflow locally.
- Stopped working in Airflow 2

## Fully support running more than one scheduler concurrently #10956

[Code](#)

 Merged ashb merged 70 commits into `apache:master` from `astronomer:scheduler-ha` on 9 Oct 2020

 Conversation 253  Commits 70  Checks 68  Files changed 40

+3,423 -2,306 



ashb commented on 15 Sep 2020 · edited

Member  ...

Reviewers

## The closer, slightly dark past



- LineageBackend - sounds like right tool for the job?
- Both Airflow 1.10 and Airflow 2.1+ supported
- You can choose your LineageBackend in Airflow config
- Does not allow us to emit events on task start or failure
- We need those to reliably report what happened!
- Let's contribute!









# The closer, slightly dark past

- LineageBackend - sounds like right tool for the job?
- Both Airflow 1.10 and Airflow 2.1+ supported
- You can choose your LineageBackend in Airflow config
- Does not allow us to emit events on task start or failure
- We need those to reliably report what happened!
- Let's contribute!
- Turns out it's not so simple.

## LineageBackend is notified on pre\_execute and post fail #18470

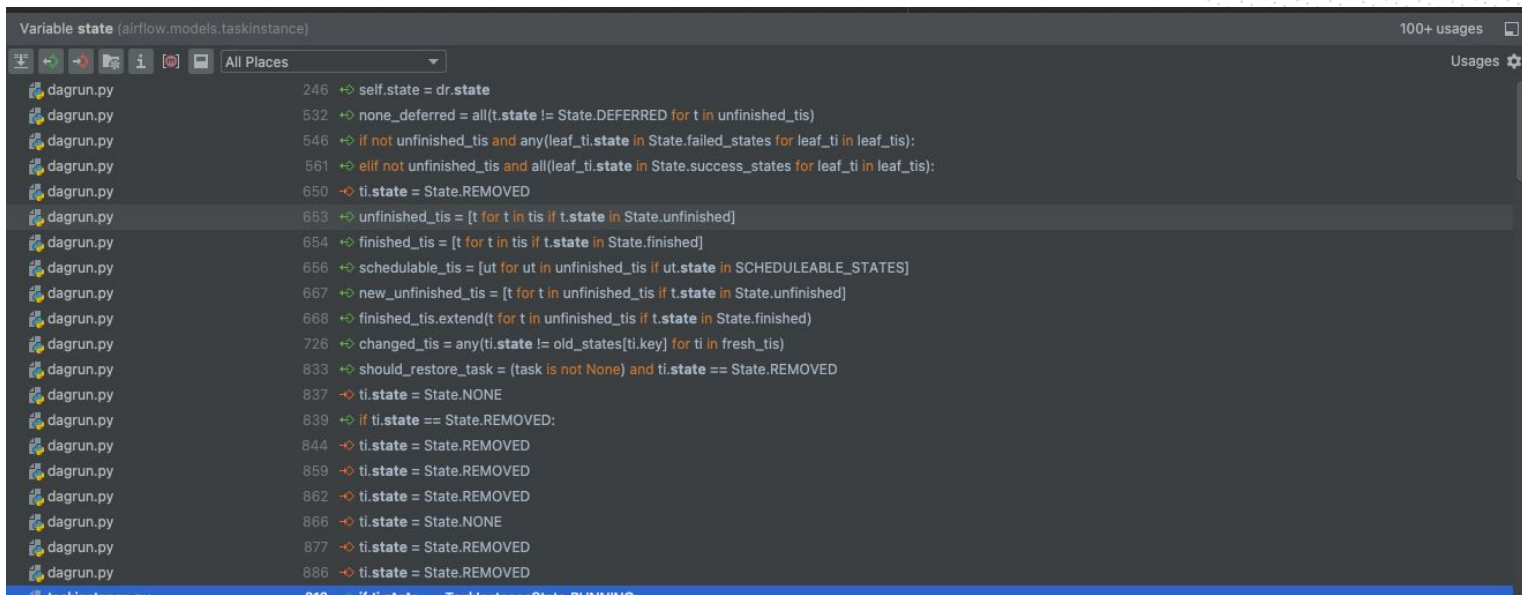
 Closed mobuchowski wants to merge 1 commit into `apache:main` from `mobuchowski:add_methods_to_lineage_backend` 

 Conversation 9  Commits 1  Checks 33  Files changed 5

 mobuchowski commented on 23 Sep 2021 Contributor  ...

# The great present

- Let's add new interface!
- We want our plugin to be notified when TaskInstanceState changes to RUNNING, SUCCESS, FAILED



```
Variable state (airflow.models.taskinstance) 100+ usages
All Places Usages
dagrun.py 246 self.state = dr.state
dagrun.py 532 none_deferred = all(t.state != State.DEFERRED for t in unfinished_tis)
dagrun.py 546 if not unfinished_tis and any(leaf_ti.state in State.failed_states for leaf_ti in leaf_tis):
dagrun.py 561 elif not unfinished_tis and all(leaf_ti.state in State.success_states for leaf_ti in leaf_tis):
dagrun.py 650 ti.state = State.REMOVED
dagrun.py 653 unfinished_tis = [t for t in tis if t.state in State.unfinished]
dagrun.py 654 finished_tis = [t for t in tis if t.state in State.finished]
dagrun.py 656 schedulable_tis = [ut for ut in unfinished_tis if ut.state in SCHEDULEABLE_STATES]
dagrun.py 667 new_unfinished_tis = [t for t in unfinished_tis if t.state in State.unfinished]
dagrun.py 668 finished_tis.extend(t for t in unfinished_tis if t.state in State.finished)
dagrun.py 726 changed_tis = any(ti.state != old_states[ti.key] for ti in fresh_tis)
dagrun.py 833 should_restore_task = (task is not None) and ti.state == State.REMOVED
dagrun.py 837 ti.state = State.NONE
dagrun.py 839 if ti.state == State.REMOVED:
dagrun.py 844 ti.state = State.REMOVED
dagrun.py 859 ti.state = State.REMOVED
dagrun.py 862 ti.state = State.REMOVED
dagrun.py 866 ti.state = State.NONE
dagrun.py 877 ti.state = State.REMOVED
dagrun.py 886 ti.state = State.REMOVED
dagrun.py 889 if ti.state == TaskInstanceState.RUNNING:
```

# The great present

- SQLAlchemy allows us to listen to existing database events
- AirflowPlugin mechanism allows us to automatically load plugin code from external Python packages
- Pluggy allows us to call registered plugins without needing to know what they are

# The great present

- Okay
- Is present in Airflow 2.3!

## Add Listener Plugin API that tracks TaskInstance state changes #2

 Merged

ashb merged 7 commits into `apache:main` from `mobuchowski:tasklistener` on 13 Jan



Conversation 70



Commits 7



Checks 50



Files changed 20



**mobuchowski** commented on 21 Dec 2021 • edited ▼

Contributor



# Features

- Extractors
- Built-in extractors -
  - BigQueryExtractor
  - SnowflakeExtractor
  - PostgresExtractor
  - GreatExpectationsExtractor
  - ...
- Possibility to create custom extractors

# Features

- Additional common library to help with writing extractors
- SQL parser
- Other integrations (dbt...) can use those features as well

# The shiny future

- Prevalence of PythonOperator
- Can we get data directly from Hooks?
- Hooks are very diverse.

# The shiny future

- Prevalence of PythonOperator
- Can we get data directly from Hooks?
- Hooks are very diverse.
- AIP-48 solves a lot of those problems



# Apache Spark Integration

## Top 3 recent features:

- Support for Spark 3.2.1
- Extensibility API
  - Possibility to write custom plugins to enrich existing OpenLineage events.
- Column level lineage
  - Which input columns were used to produce output column X?

## Other:

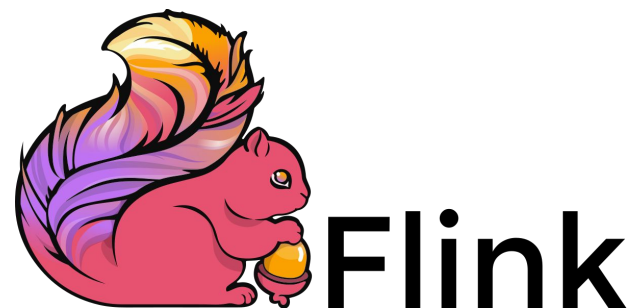
- Spawning Spark from your Airflow DAG? We'll keep track of that.
- Lifecycle state change - understand the meaning of `DROP`, `DELETE`, `ALTER...`
- Dataset versions for Iceberg & Delta



# Apache Flink integration

## Status:

- **Under construction.**
- We're already able to:
  - Identify sources & sinks Kafka topics,
  - Fetch datasets' schemas for Avro,
  - Include checkpoint statistics in OpenLineage events,
  - Retrieve information on Iceberg sources & sinks,
  - ...
- Looking forward to publish first experimental version.



# You can contribute too!

## Status:

- [github.com/OpenLineage/OpenLineage](https://github.com/OpenLineage/OpenLineage)