Data Lineage with Apache Airflow using Marquez

DATA COUNCIL ‘20

Hey!

I’m Willy Lulciuc

Software Engineer, Datakin

@wslulciuc

AGENDA

Today: Limited context

- What is the data source?
- What is the schema?
- Who is the owner?
- How often is it updated?
- Where is it coming from?
- Who is using the data?

Data contextualization!

SOURCES

- MYSQL
- POSTGRESQL
- REDSHIFT
- SNOWFLAKE
- KAFKA
- S3

DATA

FORMATS

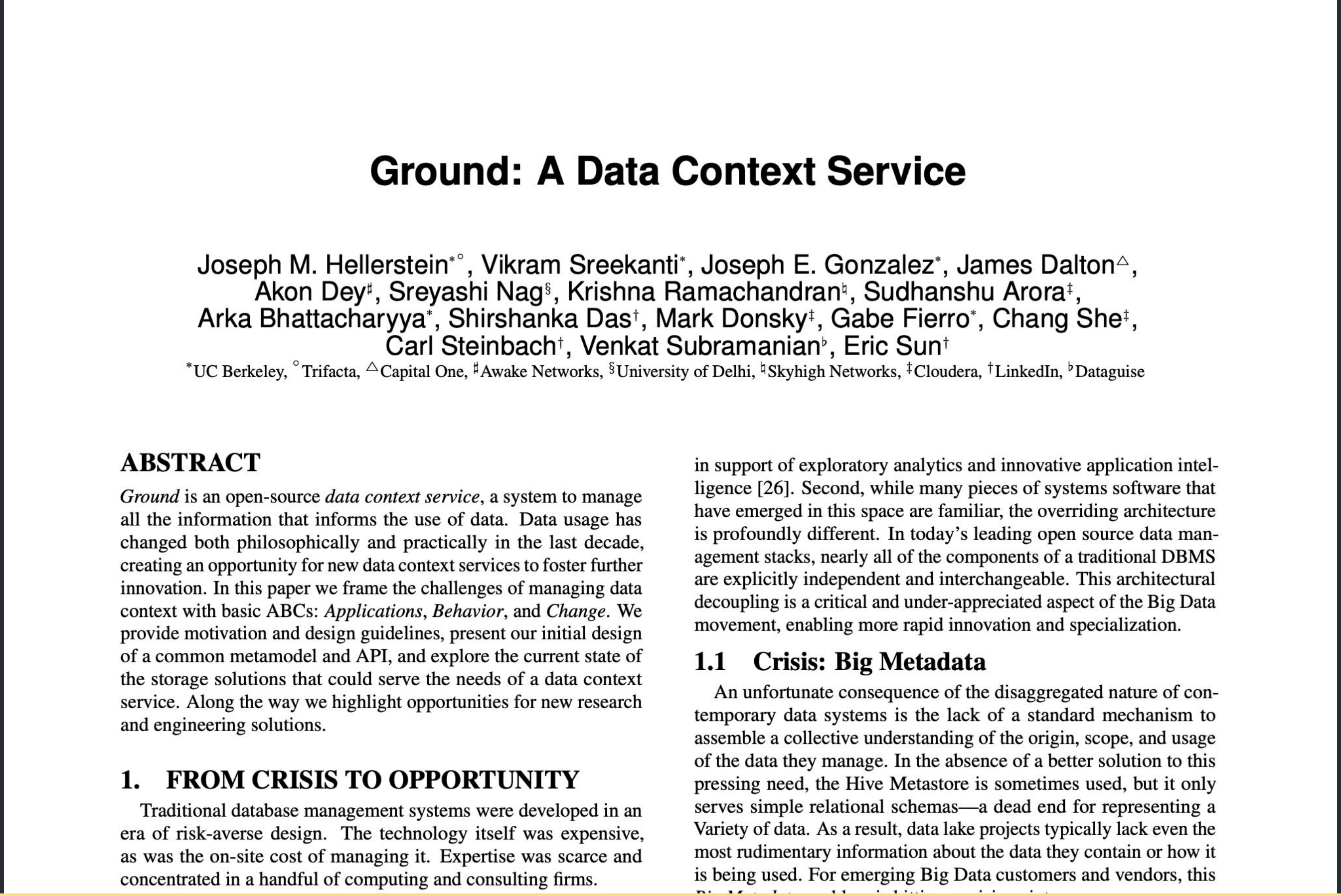
- CSV
- JSON
- AVRO
- PARQUET

01 Why metadata?

Why manage and utilize metadata?

Data quality	Data lineage	Democratize
● Build trust in data	● Understand dependencies	● Self-service and accountable data culture

02 Intro to Marquez



Ground: A Data Context Service

Joseph M. Hellerstein[*°], Vikram Sreekanti[*], Joseph E. Gonzalez[*], James Dalton[△], Akon Dey[♯], Sreyashi Nag[§], Krishna Ramachandran[♮], Sudhanshu Arora[‡], Arka Bhattacharyya[*], Shirshanka Das[†], Mark Donsky[‡], Gabe Fierro[*], Chang She[‡], Carl Steinbach[†], Venkat Subramanian[♭], Eric Sun[†]
[*]UC Berkeley, [°]Trifacta, [△]Capital One, [♯]Awake Networks, [§]University of Delhi, [♮]Skyhigh Networks, [‡]Cloudera, [†]LinkedIn, [♭]Dataguise

ABSTRACT

Ground is an open-source *data context service*, a system to manage all the information that informs the use of data. Data usage has changed both philosophically and practically in the last decade, creating an opportunity for new data context services to foster further innovation. In this paper we frame the challenges of managing data context with basic ABCs: *Applications*, *Behavior*, and *Change*. We provide motivation and design guidelines, present our initial design of a common metamodel and API, and explore the current state of the storage solutions that could serve the needs of a data context service. Along the way we highlight opportunities for new research and engineering solutions.

1. FROM CRISIS TO OPPORTUNITY

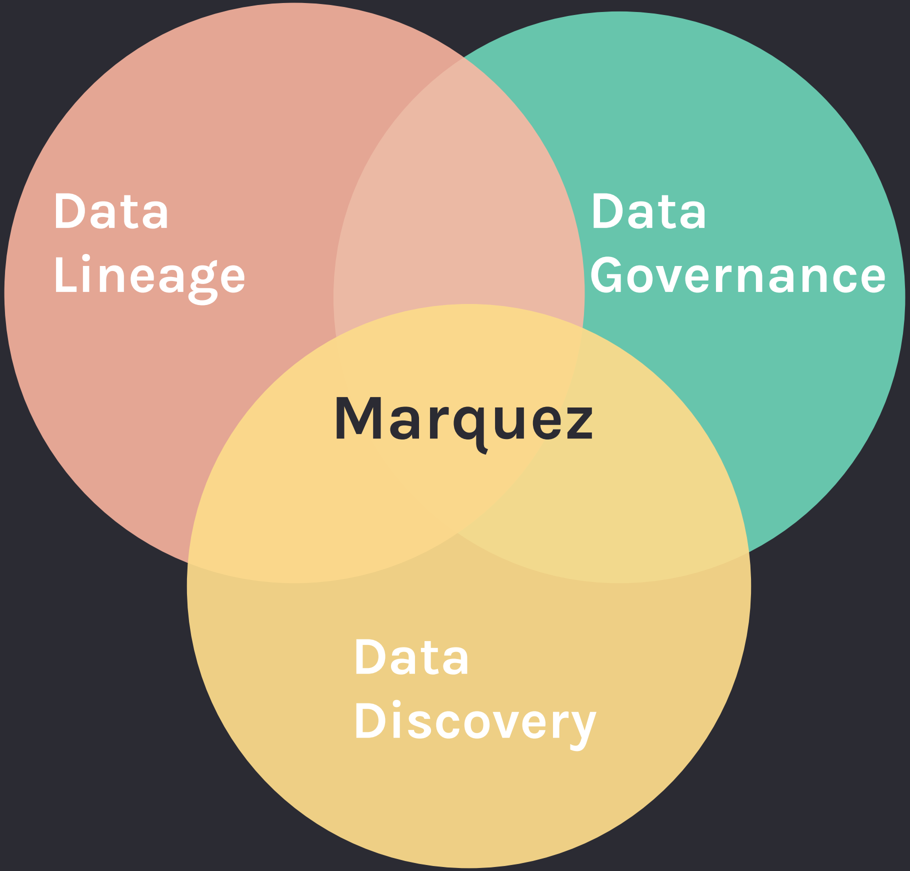
Traditional database management systems were developed in an era of risk-averse design. The technology itself was expensive, as was the on-site cost of managing it. Expertise was scarce and concentrated in a handful of computing and consulting firms.

in support of exploratory analytics and innovative application intelligence [26]. Second, while many pieces of systems software that have emerged in this space are familiar, the overriding architecture is profoundly different. In today's leading open source data management stacks, nearly all of the components of a traditional DBMS are explicitly independent and interchangeable. This architectural decoupling is a critical and under-appreciated aspect of the Big Data movement, enabling more rapid innovation and specialization.

1.1 Crisis: Big Metadata

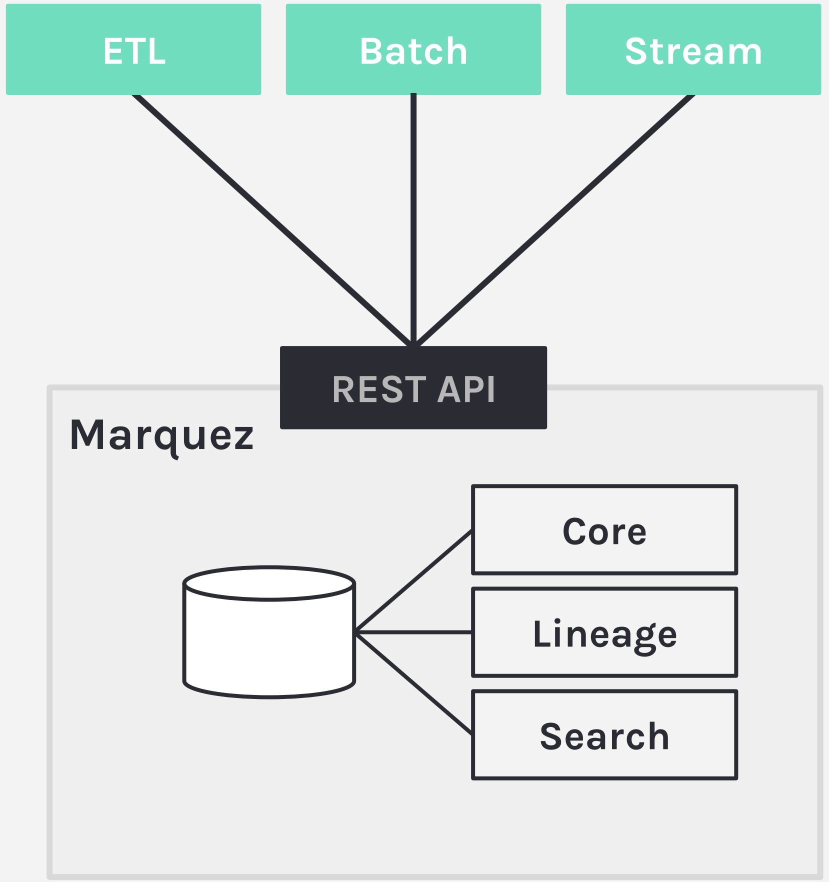
An unfortunate consequence of the disaggregated nature of contemporary data systems is the lack of a standard mechanism to assemble a collective understanding of the origin, scope, and usage of the data they manage. In the absence of a better solution to this pressing need, the Hive Metastore is sometimes used, but it only serves simple relational schemas—a dead end for representing a Variety of data. As a result, data lake projects typically lack even the most rudimentary information about the data they contain or how it is being used. For emerging Big Data customers and vendors, this

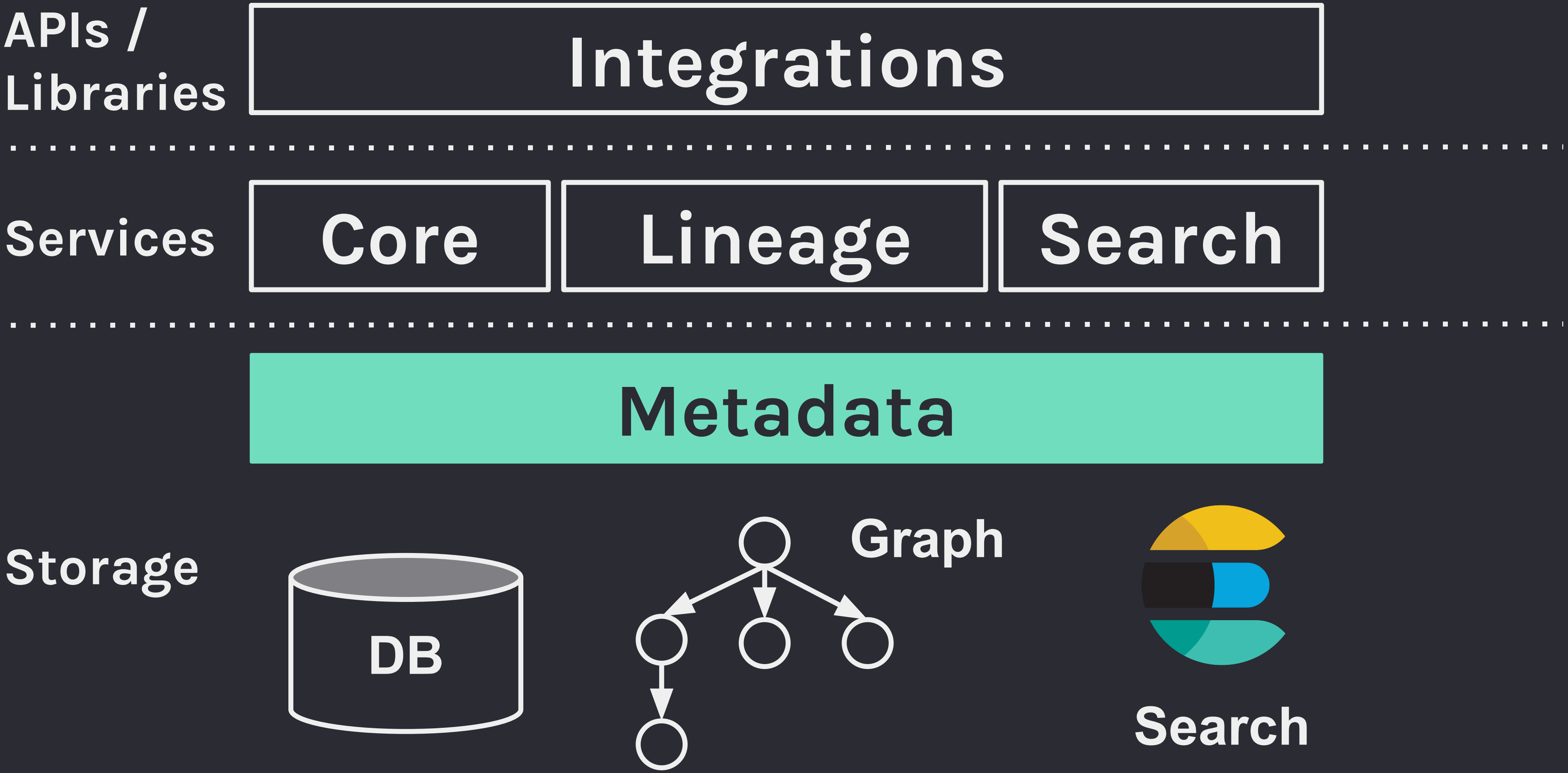
http://cidrdb.org/cidr2017/papers/p111-hellerstein-cidr17.pdf


Data Lineage
Data Governance
Marquez
Data Discovery

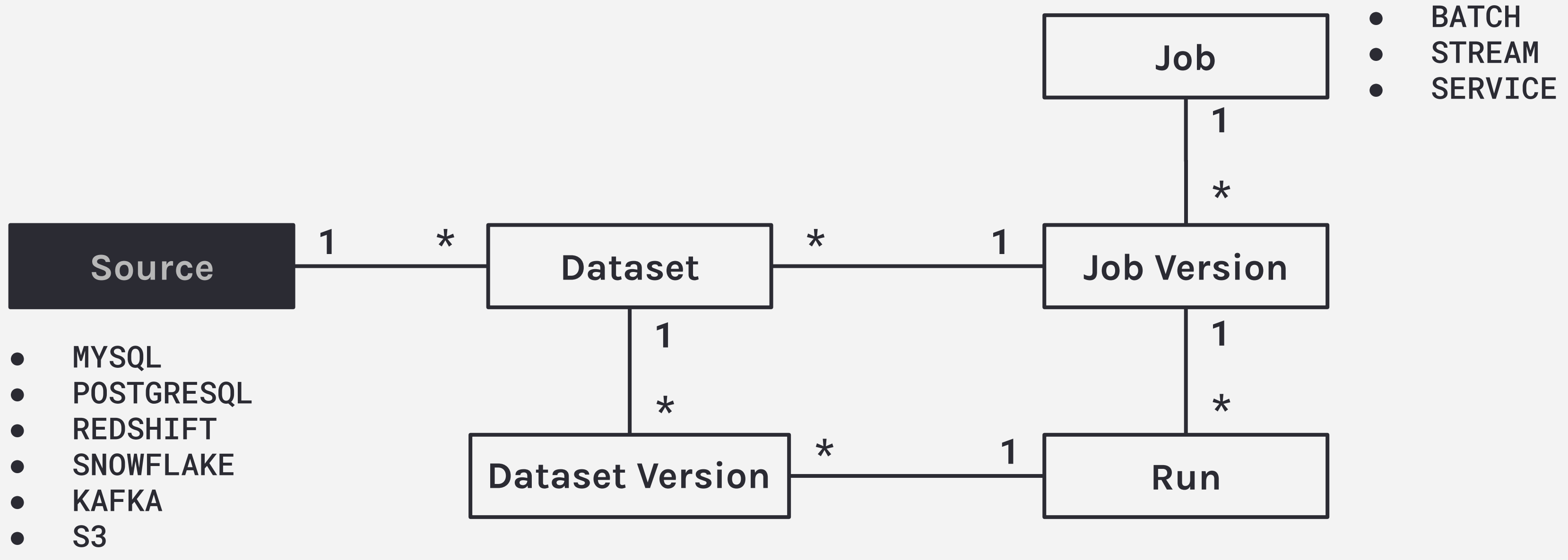
Metadata Service

- **Centralized metadata management**
  - Sources
  - Datasets
  - Jobs
- **Modular framework**
  - Data governance
  - Data lineage
  - Data discovery + exploration

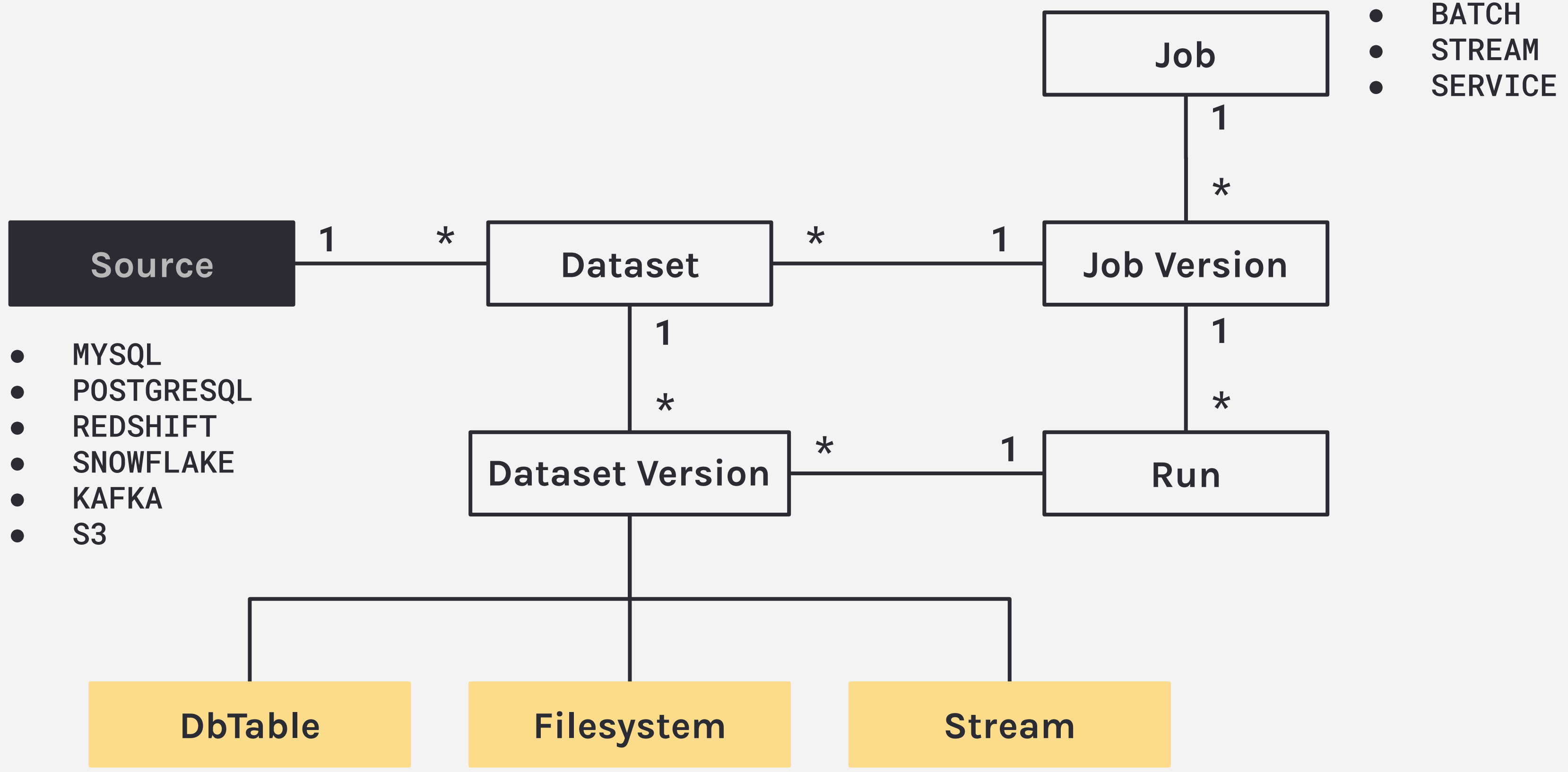



APIs / Libraries
Integrations
Services
Core
Lineage
Search
Metadata
Storage
DB
Graph
Search

Marquez: Data model

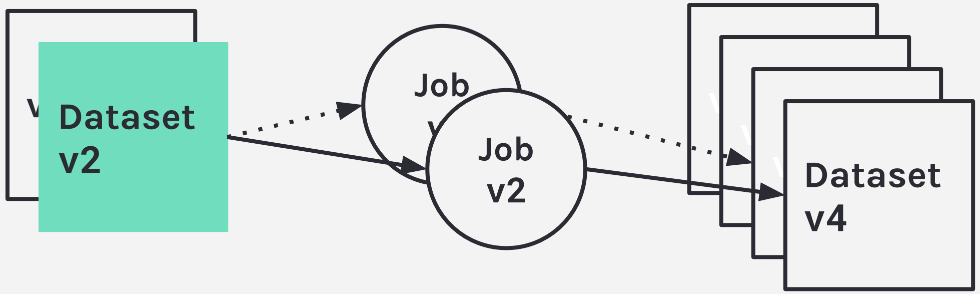


Marquez: Data model



Design benefits

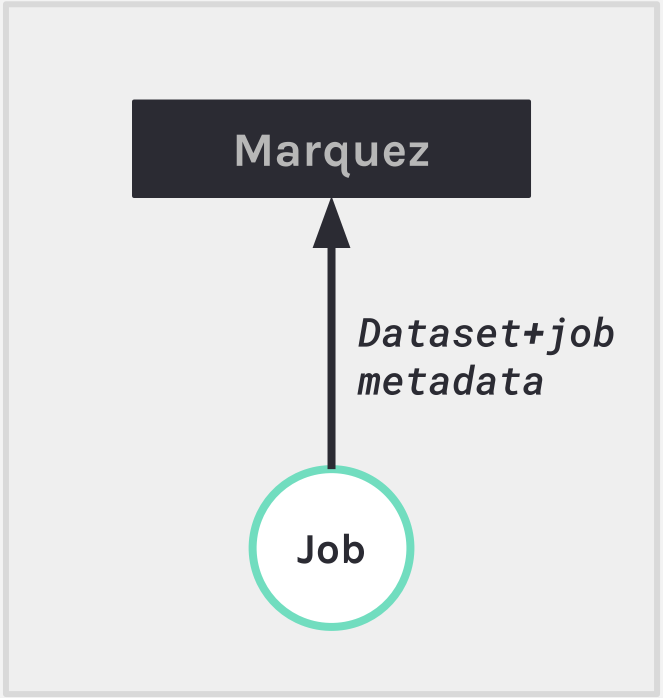
- Debugging
  - What **job version(s)** produced and consumed **dataset version X**?



- Backfilling
  - Full / incremental processing

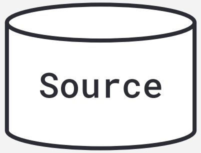
How is metadata collected?

- **Push-based** metadata collection
- REST API
- Language-specific SDKs
    - Java
    - Python



Marquez: Metadata collection

01



```
{
  "type":"POSTGRESQL",
  "name":"analyticsdb”,
  "connectionUrl":"jdbc:postgresql://localhost:5431/analytics”,
  "description":“Contains tables such as office room bookings.”
}
```

Marquez: Metadata collection

01 Source

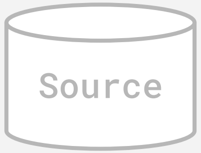
```
{
  "type":"POSTGRESQL",
  "name":"analyticsdb”,
  "connectionUrl":"jdbc:postgresql://localhost:5431/analytics”,
  "description":“Contains tables such as office room bookings.”
}
```

02 Dataset

```
{
  "type":"DB_TABLE",
  "name":"room_bookings”,
  "physicalName":"public.room_bookings”,
  "sourceName":"analyticsdb”,
  "namespace":"datascience",
  "fields": [...],
  "description":“All global room bookings for each office.”
}
```

Marquez: Metadata collection

01



```
{
  "type":"POSTGRESQL",
  "name":"analyticsdb",
  "connectionUrl":"jdbc:postgresql://localhost:5431/analytics",
  "description":"Contains tables such as office room bookings."
}
```

02

```
{
  "type":"DB_TABLE",
  "name":"room_bookings",                          LOGICAL DATASET NAME
  "physicalName":"public.room_bookings",           PHYSICAL DATASET NAME
  "sourceName":"analyticsdb",
  "namespace":"datascience",
  "fields": [...],
  "description":"All global room bookings for each office."
}
```

Marquez: Metadata collection

01 Source

```
{
  "type":"POSTGRESQL",
  "name":"analyticsdb”,
  "connectionUrl":"jdbc:postgresql://localhost:5431/analytics”,
  "description":“Contains tables such as office room bookings.”
}
```

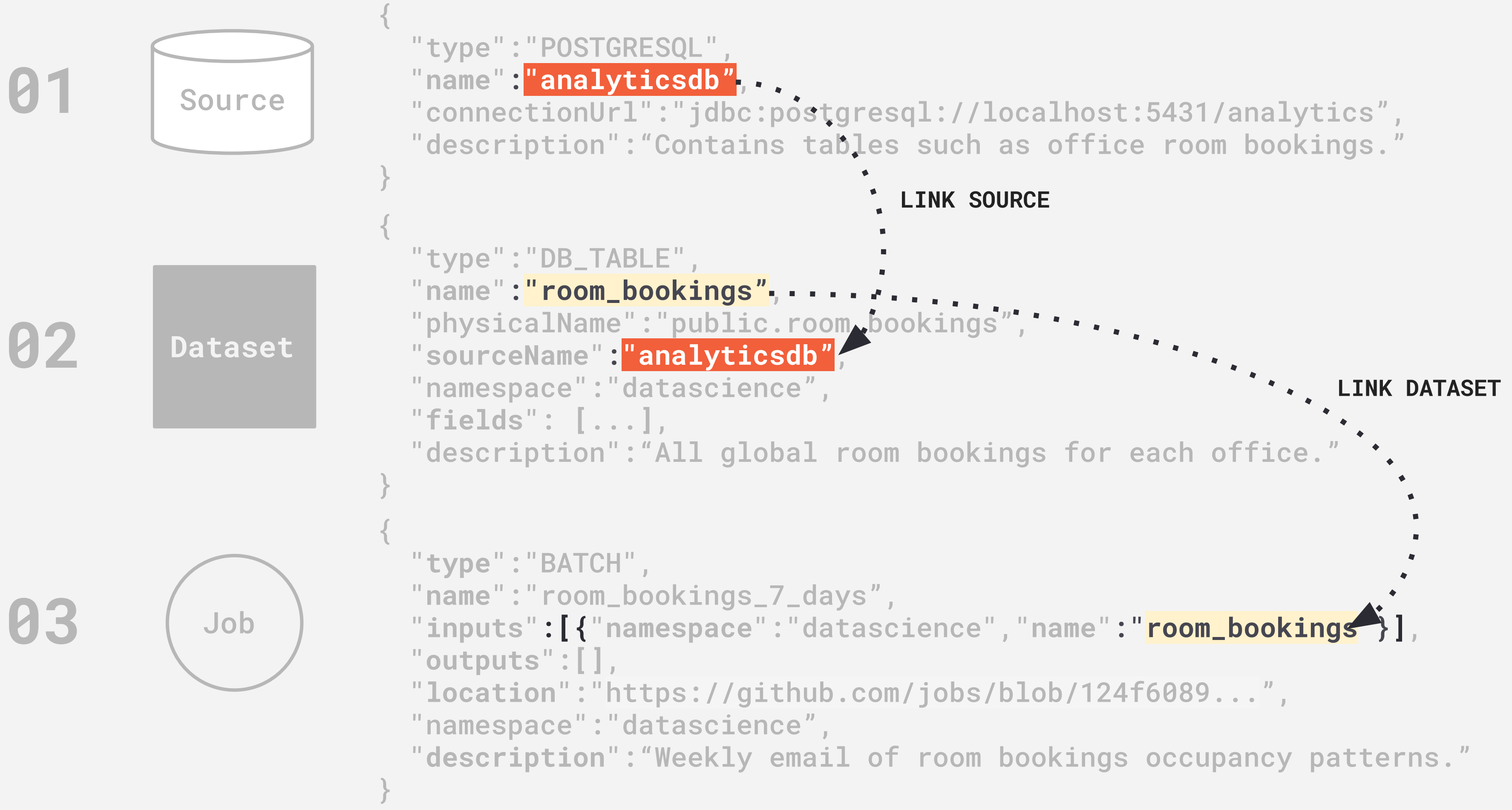
02 Dataset

```
{
  "type":"DB_TABLE",
  "name":"room_bookings”,
  "physicalName":"public.room_bookings”,
  "sourceName":"analyticsdb”,
  "namespace":"datascience”,
  "fields": [...],
  "description":“All global room bookings for each office.”
}
```

03 Job

```
{
  "type":"BATCH",
  "name":"room_bookings_7_days”,
  "inputs":[{"namespace":"datascience","name":"room_bookings”}],
  "outputs":[],
  "location":"https://github.com/jobs/blob/124f6089...”,
  "namespace":"datascience",
  "description":“Weekly email of room bookings occupancy patterns.”
}
```

Marquez: Metadata collection



```
{
  "type":"POSTGRESQL",
  "name":"analyticsdb”,
  "connectionUrl":"jdbc:postgresql://localhost:5431/analytics”,
  "description":“Contains tables such as office room bookings.”
}

{
  "type":"DB_TABLE",
  "name":"room_bookings”,
  "physicalName":"public.room_bookings”,
  "sourceName":"analyticsdb”,
  "namespace":"datascience”,
  "fields": [...],
  "description":“All global room bookings for each office.”
}

{
  "type":"BATCH",
  "name":"room_bookings_7_days”,
  "inputs":[{"namespace":"datascience","name":"room_bookings”}],
  "outputs":[],
  "location":"https://github.com/jobs/blob/124f6089...”,
  "namespace":"datascience”,
  "description":“Weekly email of room bookings occupancy patterns.”
}
```

Marquez: Metadata collection

01 Source

```
{
  "type":"POSTGRESQL",
  "name":"analyticsdb”,
  "connectionUrl":"jdbc:postgresql://localhost:5431/analytics”,
  "description":“Contains tables such as office room bookings.”
}
```

02 Dataset

```
{
  "type":"DB_TABLE",
  "name":"room_bookings”,
  "physicalName":"public.room_bookings”,
  "sourceName":"analyticsdb”,
  "namespace":"datascience”,
  "fields": [...],
  "description":“All global room bookings for each office.”
}
```

OWNERSHIP

03 Job

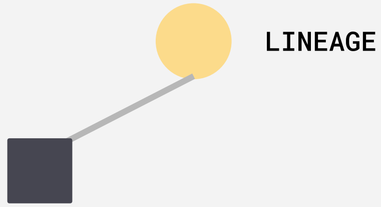
```
{
  "type":"BATCH",
  "name":"room_bookings_7_days”,
  "inputs":[{"namespace":"datascience","name":"room_bookings”}],
  "outputs":[],
  "location":"https://github.com/jobs/blob/124f6089...”,
  "namespace":"datascience”,
  "description":“Weekly email of room bookings occupancy patterns.”
}
```

Marquez: Metadata collection

01

Job
v1

```
{
  "type":"BATCH",
  "name":"room_bookings_7_days"
  "inputs":[{
    "namespace":"datascience",
    "name":"room_bookings"
  }],
  "outputs":[],
  ...
}
```

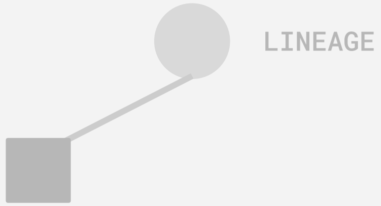


■ DATASET ● JOB

Marquez: Metadata collection

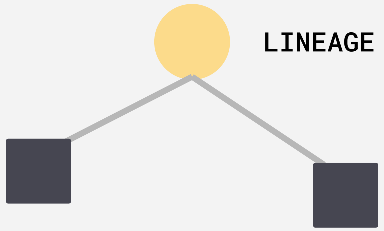
01

```
{
  "type":"BATCH",
  "name":"room_bookings_7_days"
  "inputs":[{
    "namespace":"datascience",
    "name":"room_bookings"
  }],
  "outputs":[],
  ...
}
```



02

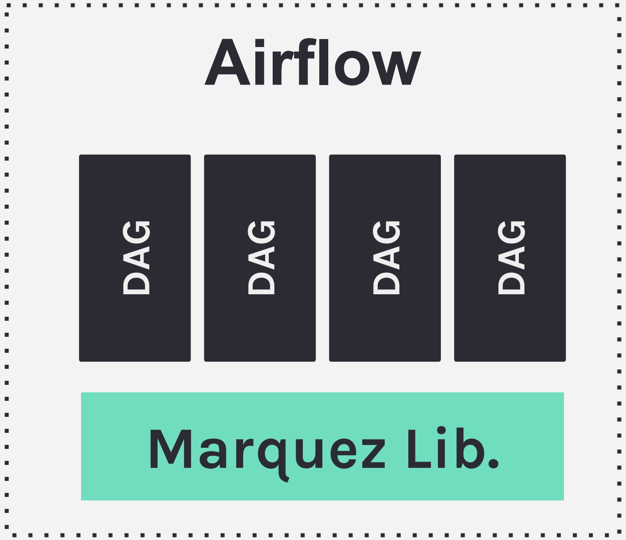
```
{
  "type":"BATCH",
  "name":"room_bookings_7_days"
  "inputs":[{
    "namespace":"datascience",
    "name":"room_bookings"
  }],
  "outputs":[{
    "namespace":"datascience",
    "name":"room_bookings_aggs"
  }],
  ...
}
```



■ DATASET ● JOB

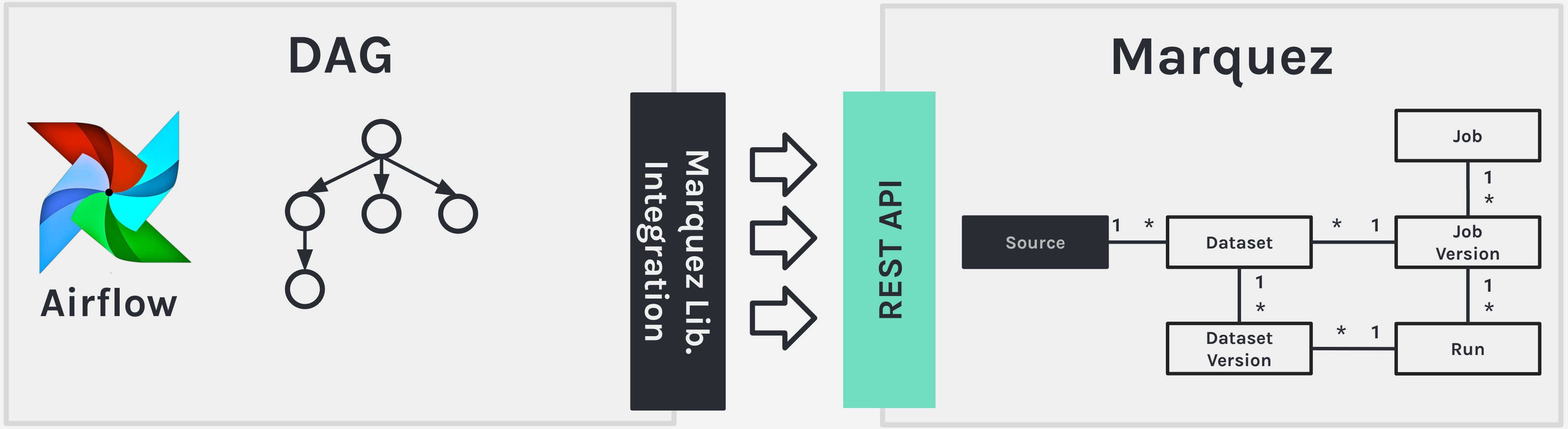
Marquez + Airflow

Airflow support for Marquez



- **Metadata**
  - Task lifecycle
  - Task parameters
  - Task runs linked to **versioned** code
  - Task inputs / outputs
- **Lineage**
  - Track inter-DAG dependencies
- **Built-in**
  - SQL parser
  - Link to code builder (**GitHub**)
  - Metadata extractors

Capturing task-level metadata in a nutshell



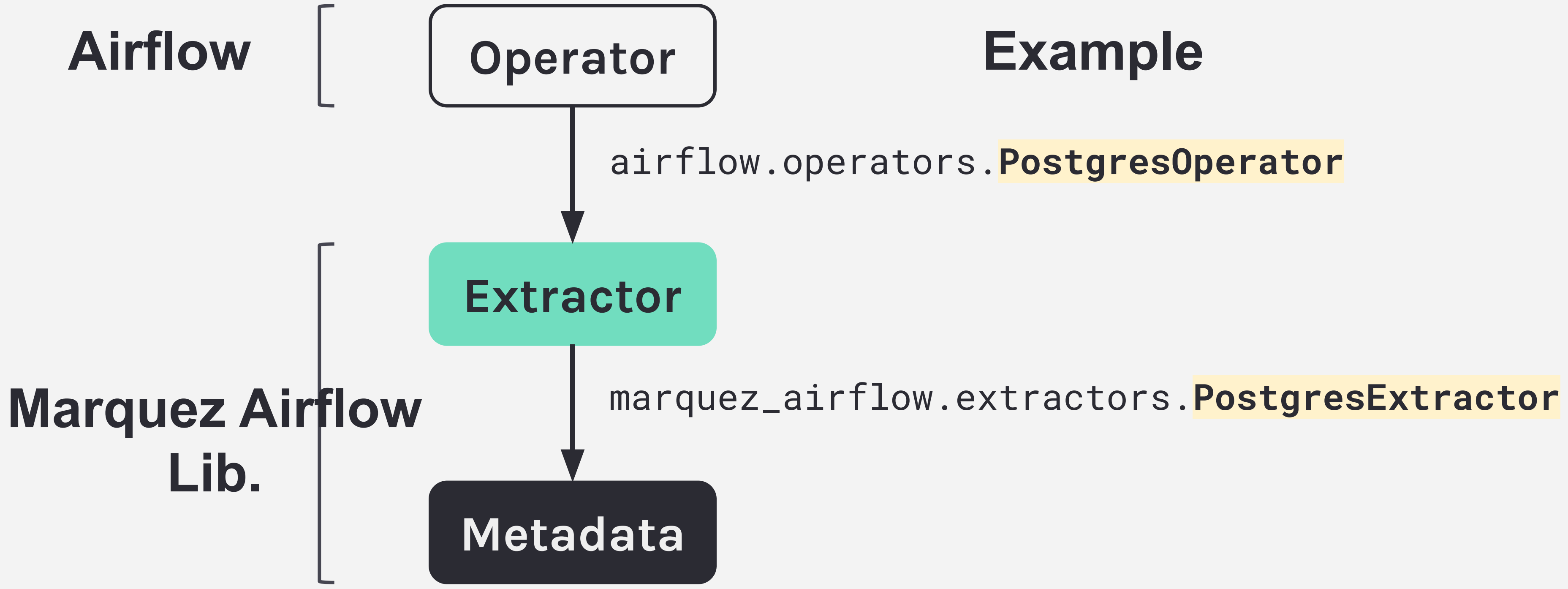
Marquez Airflow Lib.

- Open source! 🥇
- Enables **global** task-level metadata collection
- Extends Airflow’s **DAG** class

room_bookings_7_days_dag.py

```
from marquez_airflow import DAG
from airflow.operators.postgres_operator import PostgresOperator

...
```

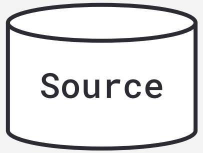

Airflow
Marquez Airflow Lib.
Operator
Extractor
Metadata
Example
airflow.operators.PostgresOperator
marquez_airflow.extractors.PostgresExtractor

Operator Metadata

new_room_booking_dag.py

```
t1=PostgresOperator(
 task_id='new_room_booking',
 postgres_conn_id='analyticsdb',
 sql='''
   INSERT INTO room_bookings VALUES(%s, %s, %s)
 '''
 parameters=... # room booking
)
```

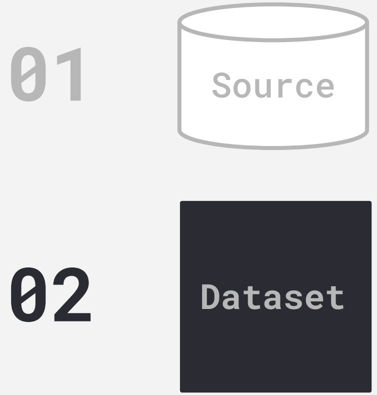
01



Operator Metadata

new_room_booking_dag.py

```
t1=PostgresOperator(
 task_id='new_room_booking',
 postgres_conn_id='analyticsdb',
 sql='''
   INSERT INTO room_bookings VALUES(%s, %s, %s)
 '''
 parameters=... # room booking
)
```

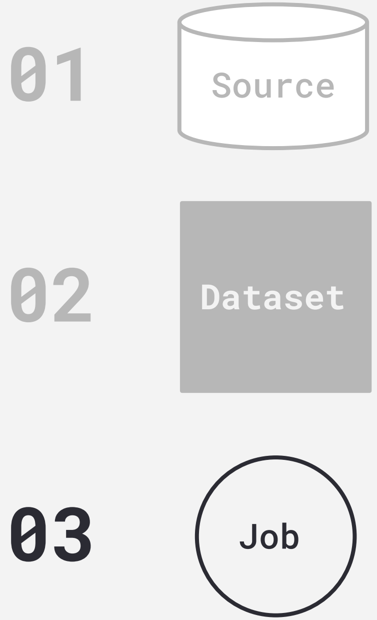


Marquez: Airflow

Operator Metadata

new_room_booking_dag.py

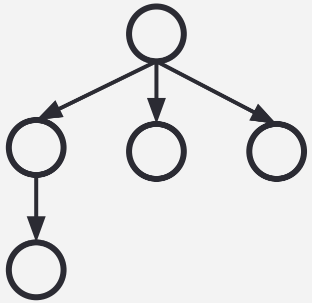
```
t1=PostgresOperator(
 task_id='new_room_booking',
 postgres_conn_id='analyticsdb',
 sql='''
   INSERT INTO room_bookings VALUES(%s, %s, %s)
 '''
 parameters=... # room booking
)
```

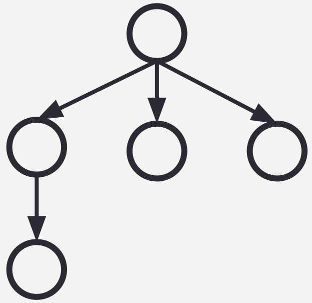


Managing inter-DAG dependencies

`new_room_bookings_dag.py`

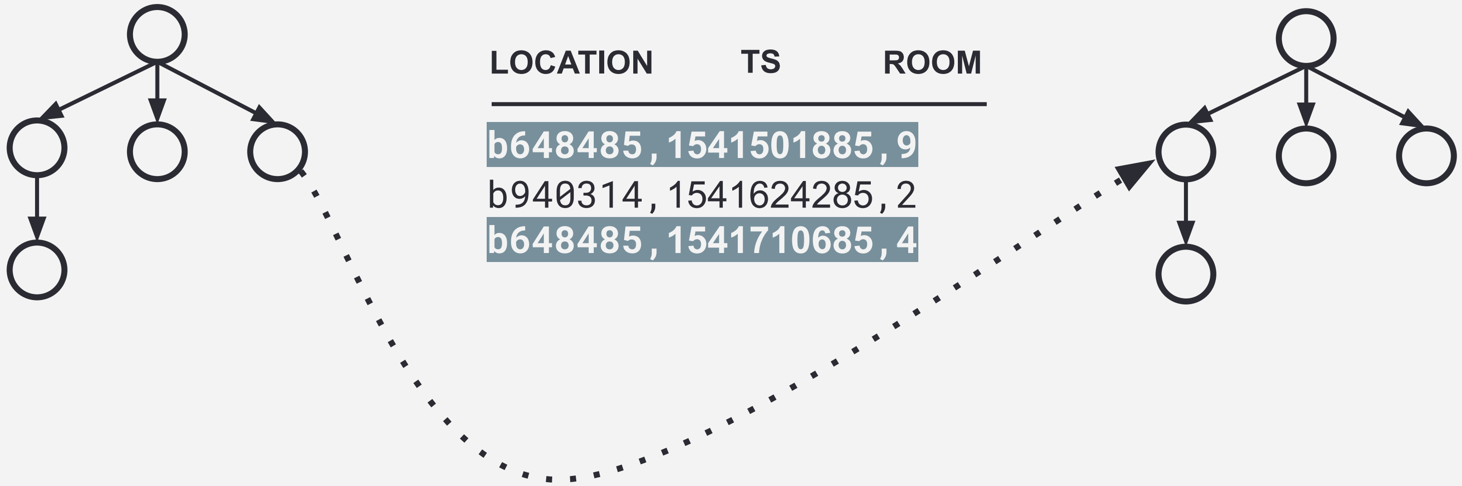
`top_room_bookings_dag.py`





Managing inter-DAG dependencies

`new_room_bookings_dag.py` `public.room_bookings` `top_room_bookings_dag.py`



LOCATION	TS	ROOM
b648485	1541501885	9
b940314	1541624285	2
b648485	1541710685	4

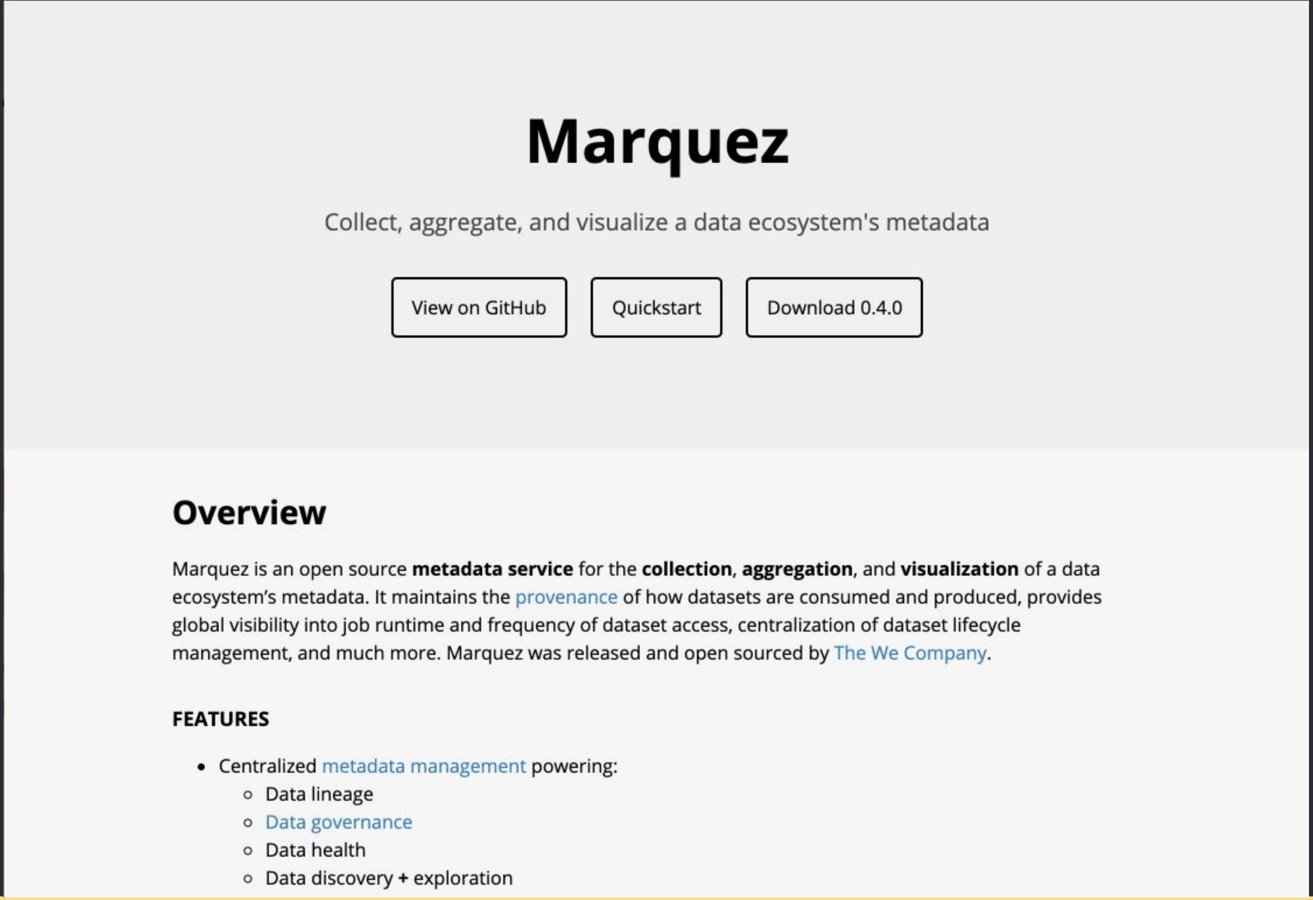
04 Future Work

Roadmap

- **Short-term**
    - Configurable data sources
    - Expand Airflow operator support
    - Docs
- **Medium-term**
    - Spark integration

MARQUEZ + LF AI

Thanks! <o/



Marquez

Collect, aggregate, and visualize a data ecosystem's metadata

Overview

Marquez is an open source **metadata service** for the **collection**, **aggregation**, and **visualization** of a data ecosystem's metadata. It maintains the provenance of how datasets are consumed and produced, provides global visibility into job runtime and frequency of dataset access, centralization of dataset lifecycle management, and much more. Marquez was released and open sourced by The We Company.

FEATURES

- Centralized metadata management powering:
  - Data lineage
  - Data governance
  - Data health
  - Data discovery + exploration

https://marquezproject.github.io/marquez github.com/MarquezProject

@MarquezProject

Questions?