

MARQUEZ

Data Lineage and observability

Julien Le Dem

CTO and co-founder Datakin

@J_

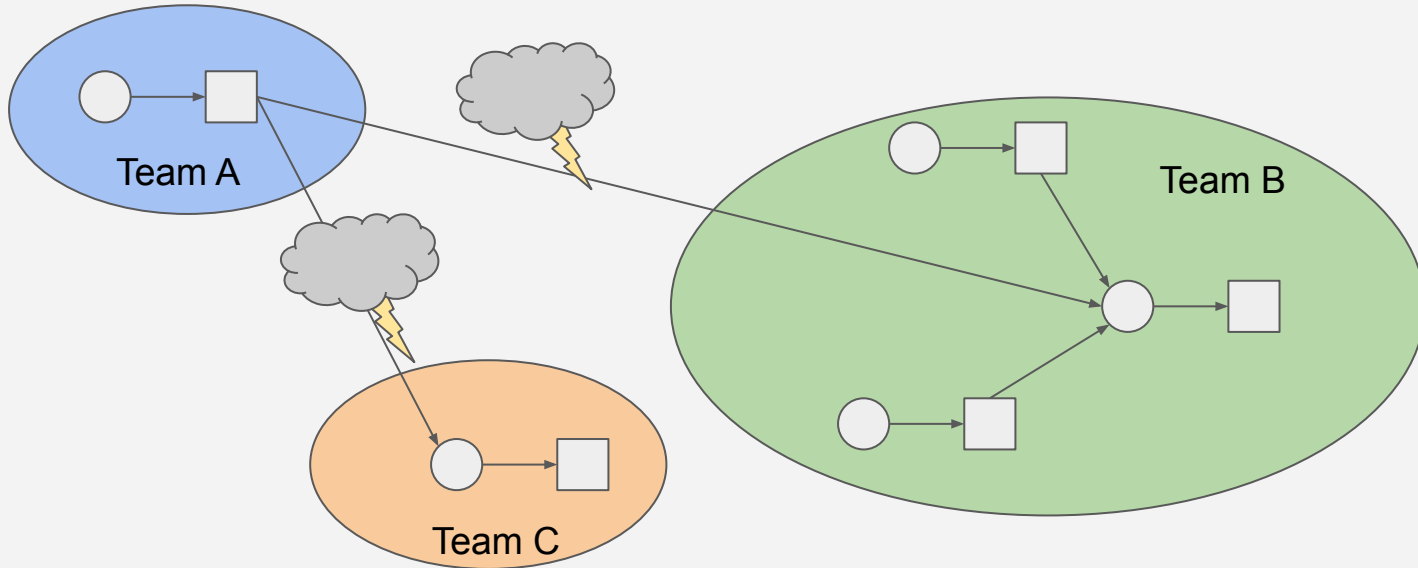
AGENDA

- 01 Why metadata?
- 02 Intro to Marquez
- 03 Airflow integration
- 04 Marquez community

01 Why metadata?

**Need to create a healthy
data ecosystem**

Team interdependencies



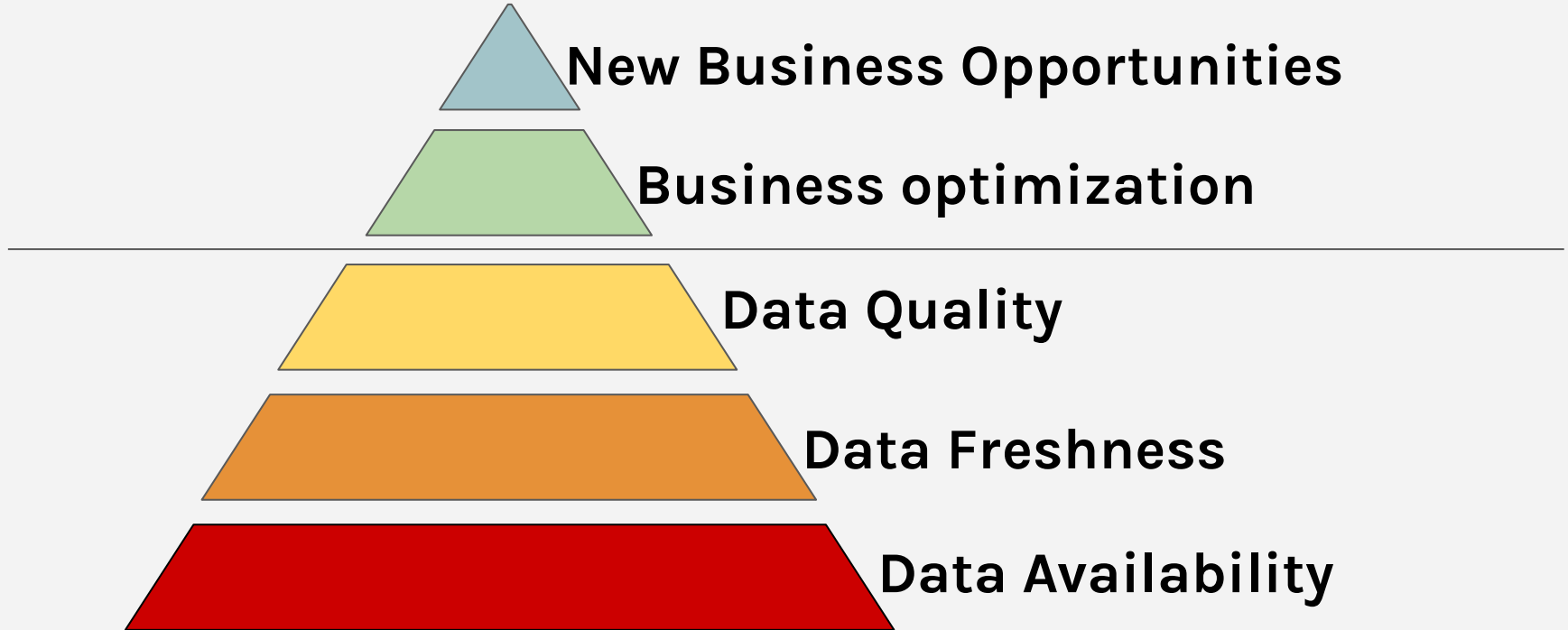
Today: Limited context



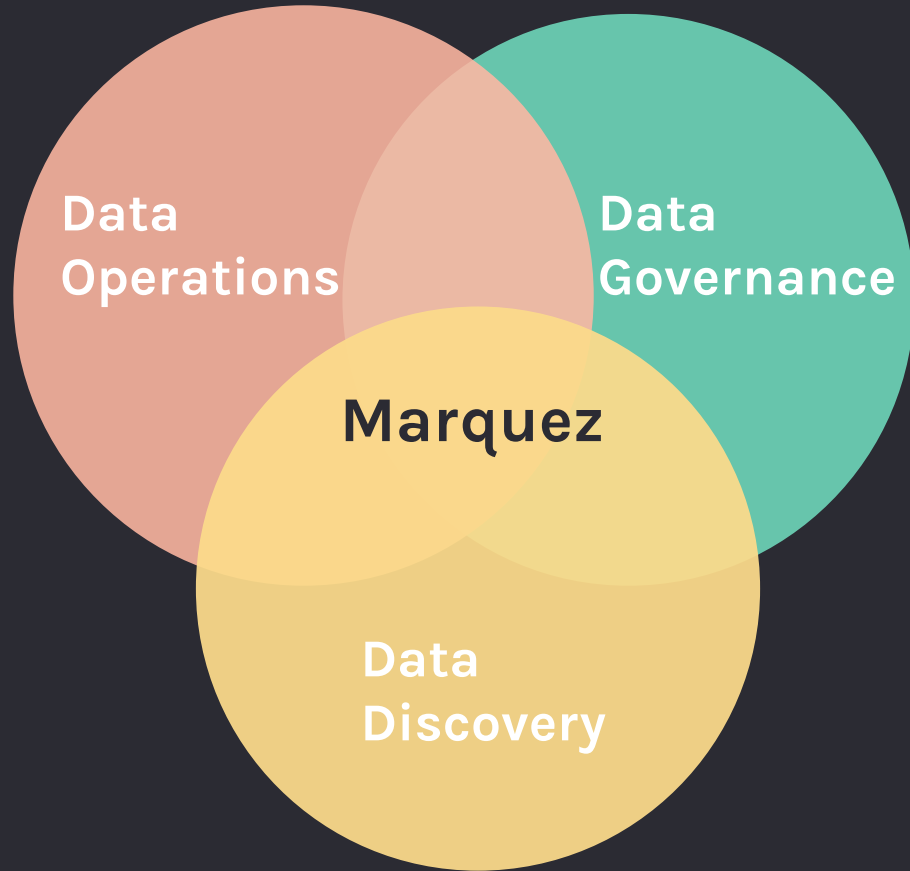
DATA

- 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?
- What has changed?

~~Maslow's~~ Data hierarchy of needs



02 Intro to Marquez



Data
Operations

Data
Governance

Marquez

Data
Discovery

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, [‡]Dataguisse

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

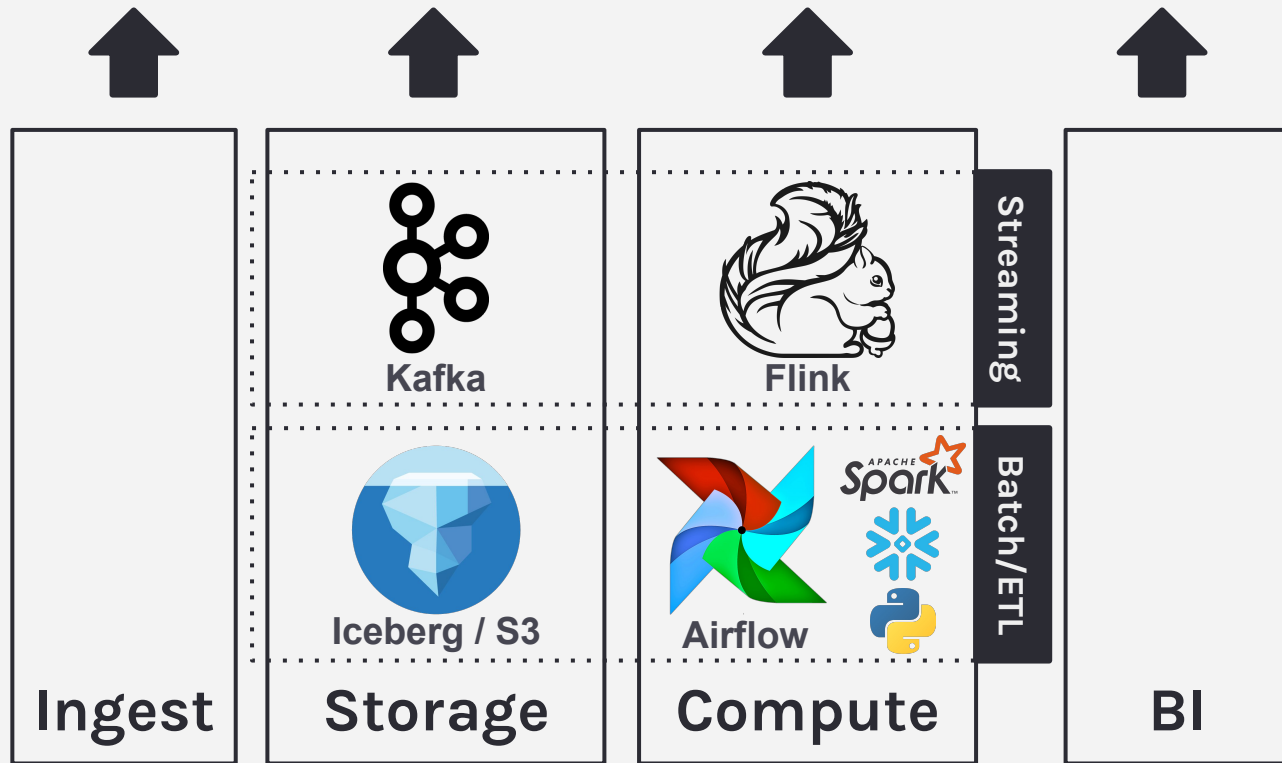
An unfortunate consequence of the disaggregated nature of contemporary data systems is the lack of a standard mechanism to assemble a collective understanding of the origin, scope, and usage of the data they manage. In the absence of a better solution to this pressing need, the Hive Metastore is sometimes 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



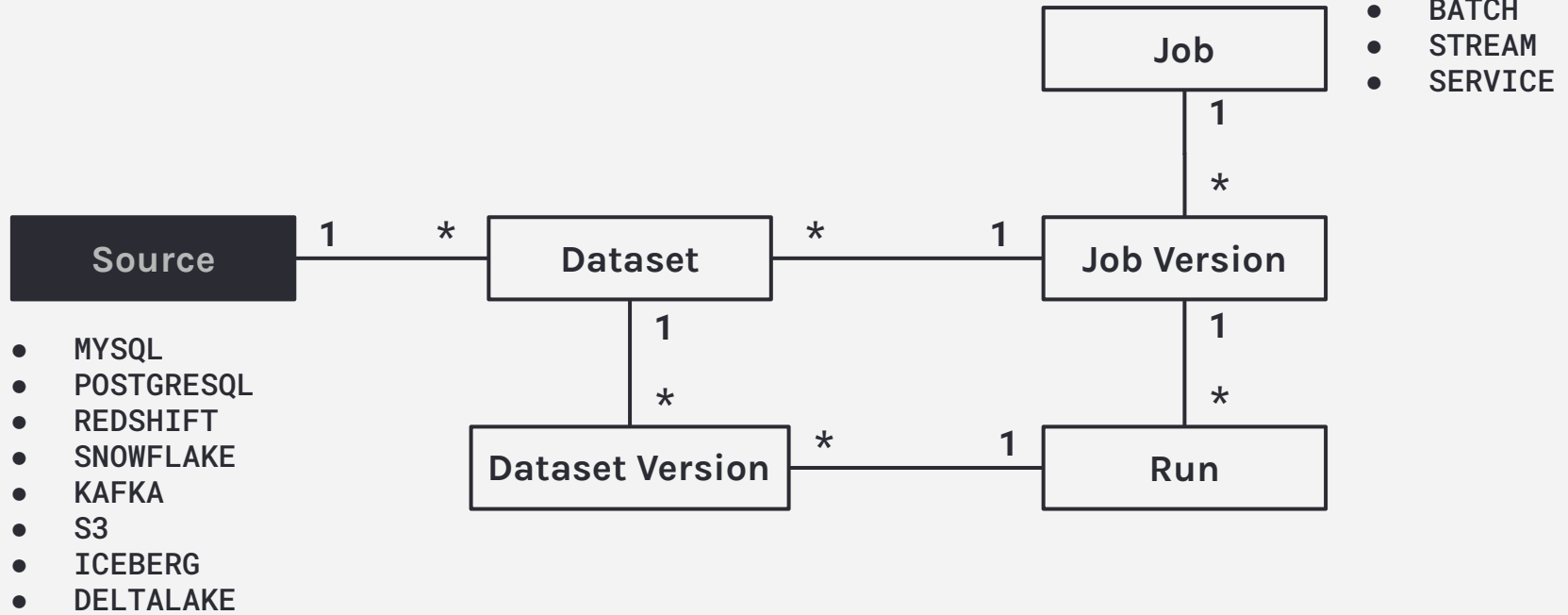
Metadata (Marquez)

- **Data Platform** built around **Marquez**

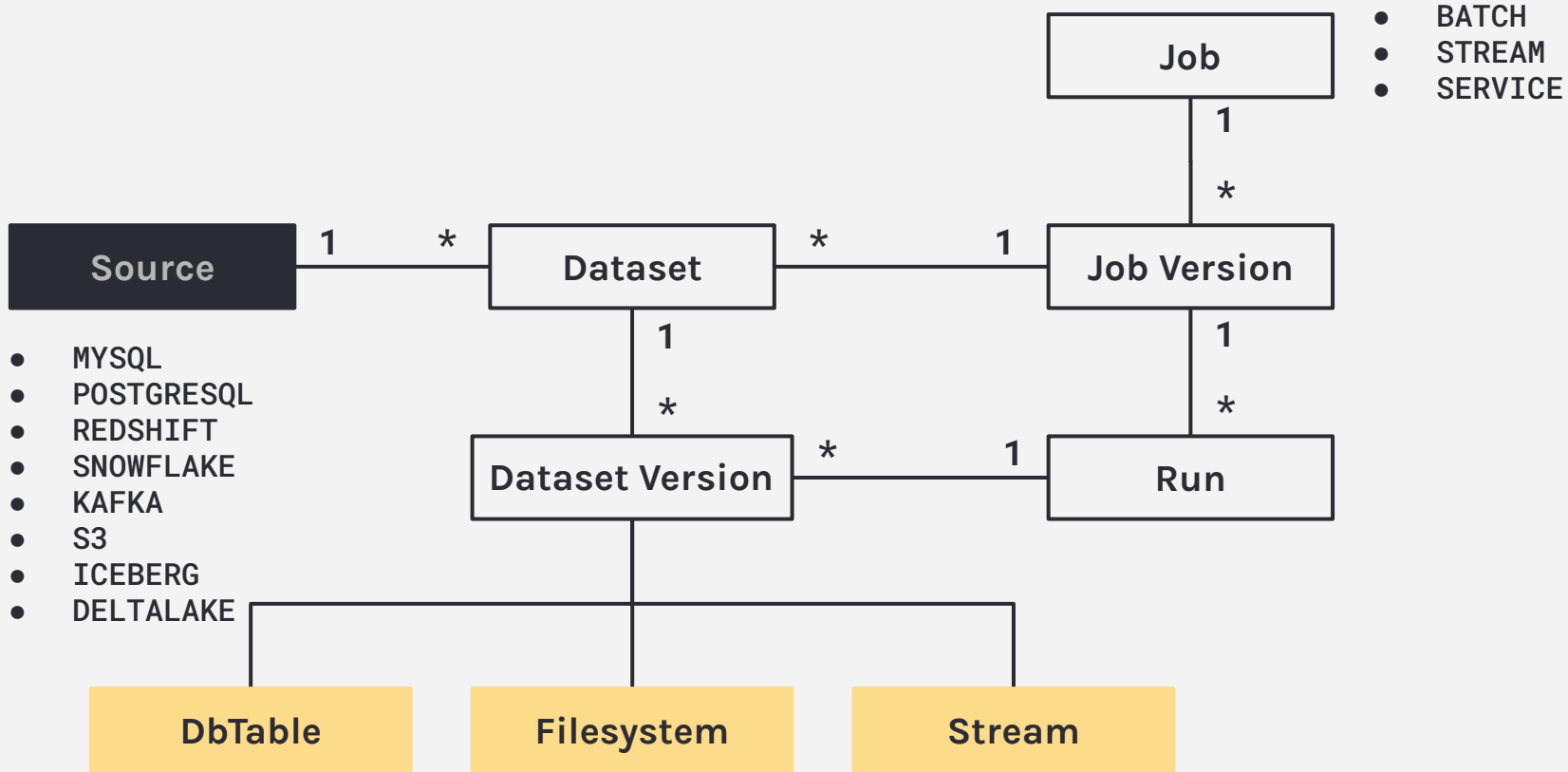
- **Integrations**
 - Ingest
 - Storage
 - Compute



Marquez: Data model

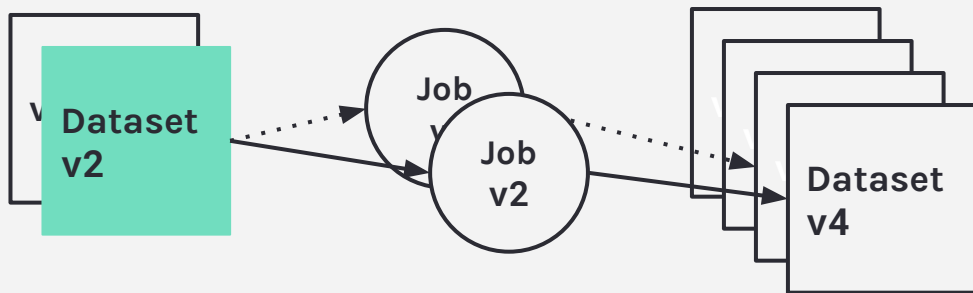


Marquez: Data model



Design benefits

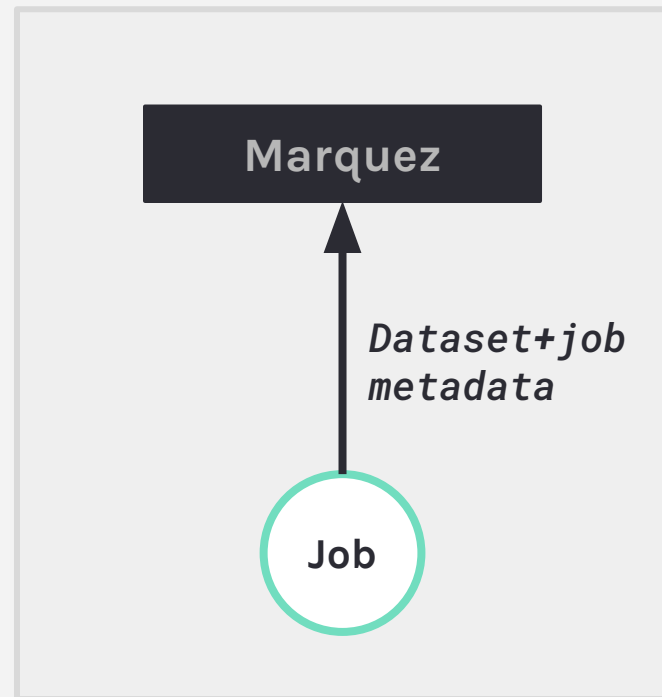
- Debugging
 - What **job version(s)** produced and consumed **dataset version X**?



- Backfilling
 - Full / incremental processing

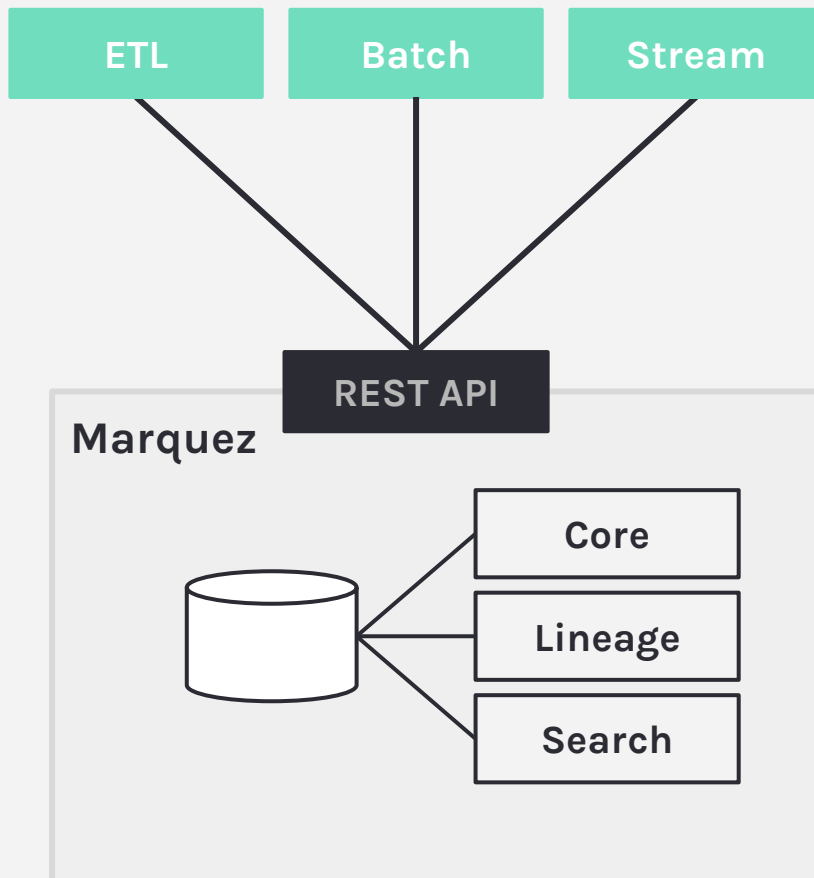
How is metadata collected?

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



Metadata Service

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



Client -
side

Integrations

Marquez UI

Extensions

APIs

Lineage collection

Core API

datakin

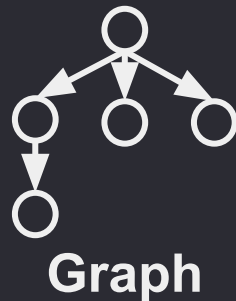
Metadata

Lineage
analysis

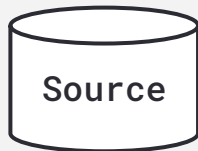
Storage

Core
DB

Listener



01



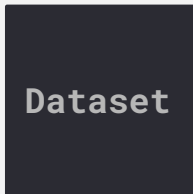
```
{  
  "type": "POSTGRESQL",  
  "name": "analyticsdb",  
  "connectionUrl": "jdbc:postgresql://localhost:5431/analytics",  
  "description": "Contains tables such as office room bookings."  
}
```

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",  
  "physicalName": "public.room_bookings",  
  "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

01



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

LINK SOURCE

02



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

LINK DATASET

03

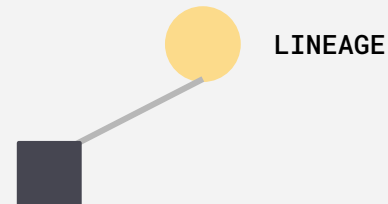


```
{  
  "type": "BATCH",  
  "name": "room_bookings_7_days",  
  "inputs": [{"namespace": "datascience", "name": "room_bookings"}],  
  "outputs": [],  
  "location": "https://github.com/jobs/blob/124f6089...",  
  "namespace": "datascience",  
  "description": "Weekly email of room bookings occupancy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

01



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

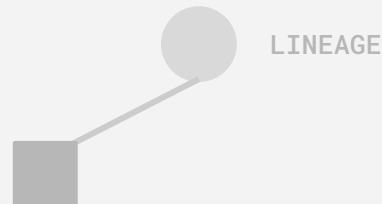


Marquez: Metadata collection

01



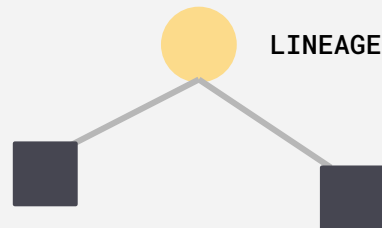
```
{  
  "type": "BATCH",  
  "name": "room_bookings_7_days"  
  "inputs": [{  
    "namespace": "datascience",  
    "name": "room_bookings"  
  }],  
  "outputs": [],  
  ...  
}
```



02



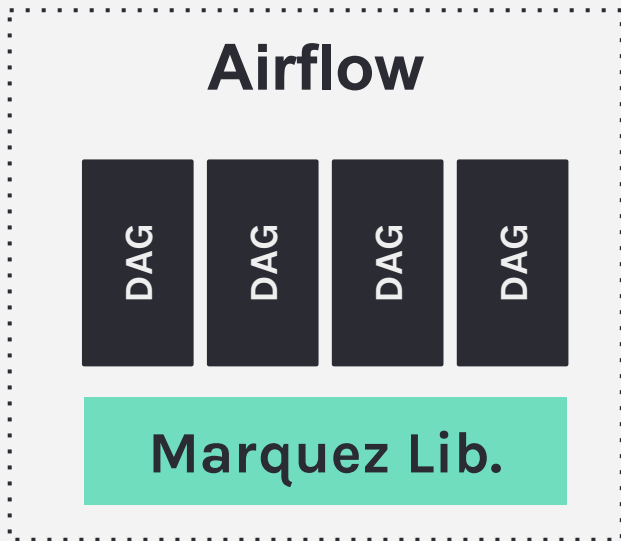
```
{  
  "type": "BATCH",  
  "name": "room_bookings_7_days"  
  "inputs": [{  
    "namespace": "datascience",  
    "name": "room_bookings"  
  }],  
  "outputs": [{  
    "namespace": "datascience",  
    "name": "room_bookings_aggs"  
  }],  
  ...  
}
```



03 Airflow integration

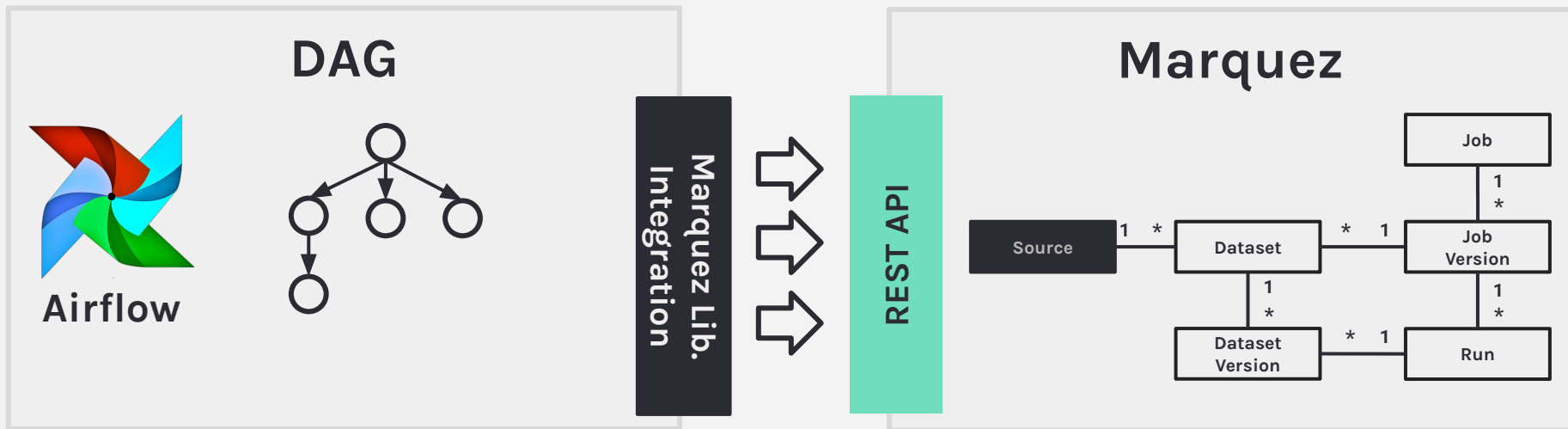


Airflow support for Marquez



- **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: marquez-airflow
- Enables **global** task-level metadata collection
- Extends Airflow's DAG class

room_bookings_7_days_dag.py

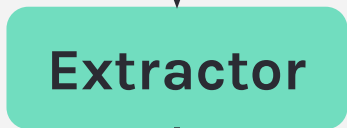
```
from marquez_airflow import DAG
from airflow.operators.postgres_operator import PostgresOperator
...
```

Airflow



Example

`airflow.operators.PostgresOperator`



Marquez Airflow Lib.



`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

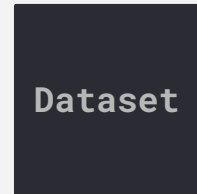
new_room_booking_dag.py

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

01



02



Operator Metadata

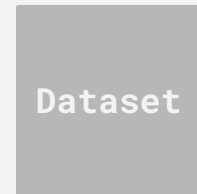
new_room_booking_dag.py

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

01



02

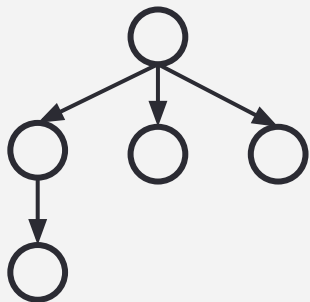


03

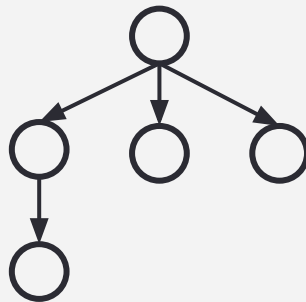


Managing inter-DAG dependencies

`new_room_bookings_dag.py`



`top_room_bookings_dag.py`

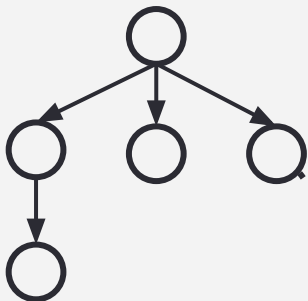


Managing inter-DAG dependencies

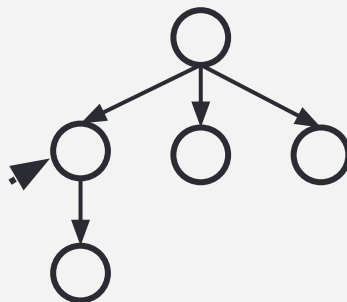
`new_room_bookings_dag.py`

`public.room_bookings`

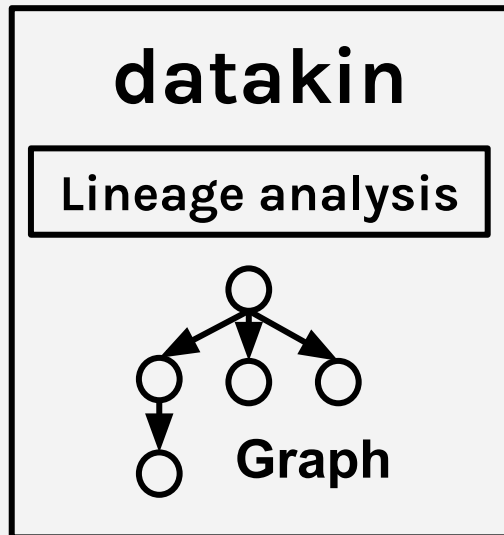
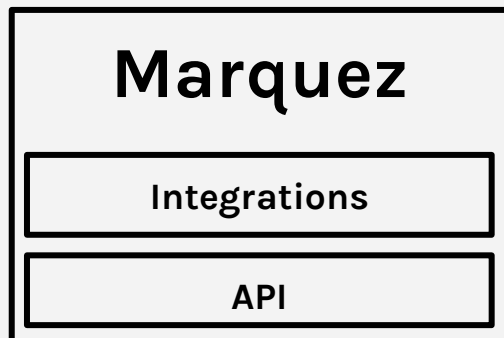
`top_room_bookings_dag.py`



LOCATION	TS	ROOM
<code>b648485</code>	<code>, 1541501885</code>	<code>, 9</code>
<code>b940314</code>	<code>, 1541624285</code>	<code>, 2</code>
<code>b648485</code>	<code>, 1541710685</code>	<code>, 4</code>



Datakin leverages Marquez metadata



- **Marquez standardizes metadata collection**
 - Job runs
 - parameters
 - version
 - inputs / outputs
- **Datakin enables**
 - Understanding operational dependencies
 - Impact analysis
 - Troubleshooting: What has changed since the last time it worked?

04 Community

Marquez

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

[View on GitHub](#)

[Quickstart](#)

[Download 0.4.0](#)

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>

Now part of the LF AI foundation

Governance

- Decision mechanisms
- Becoming a maintainer
- Code of Conduct

Neutral

- Not controlled by a company
- Community driven


Community

- Build trust
- Grow adoption
- Everybody is on an equal footing

github.com/MarquezProject



@MarquezProject



Project Marquez

Collect, aggregate, and visualize a data ecosystem's metadata
<https://marquezproject.ai>

Repositories 8 Packages People 6 Teams 1 Projects Settings

Find a repository... Type: All Language: All Customize pins New

marquez-java

Java client for Marquez
java

Java Apache-2.0 5 5 2 (1 issue needs help) 1 Updated 2 days ago

marquez-web

Marquez Web UI
react

TypeScript 4 16 20 (5 issues need help) 2 Updated 5 days ago

marquez

Collect, aggregate, and visualize a data ecosystem's metadata
data-catalog data-discovery data-dictionary data-governance
data-lineage data-provenance metadata-service

Java Apache-2.0 41 328 34 0 Updated 16 days ago

marquez-python

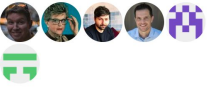
Python client for Marquez
python3

Top languages

- Python
- Java
- Shell
- TypeScript

People

6 >



Invite someone

Thanks! <o/

Questions?