



Observability for data pipelines with Open Lineage

Julien Le Dem
CTO & Co-Founder Datakin
@J_



AGENDA

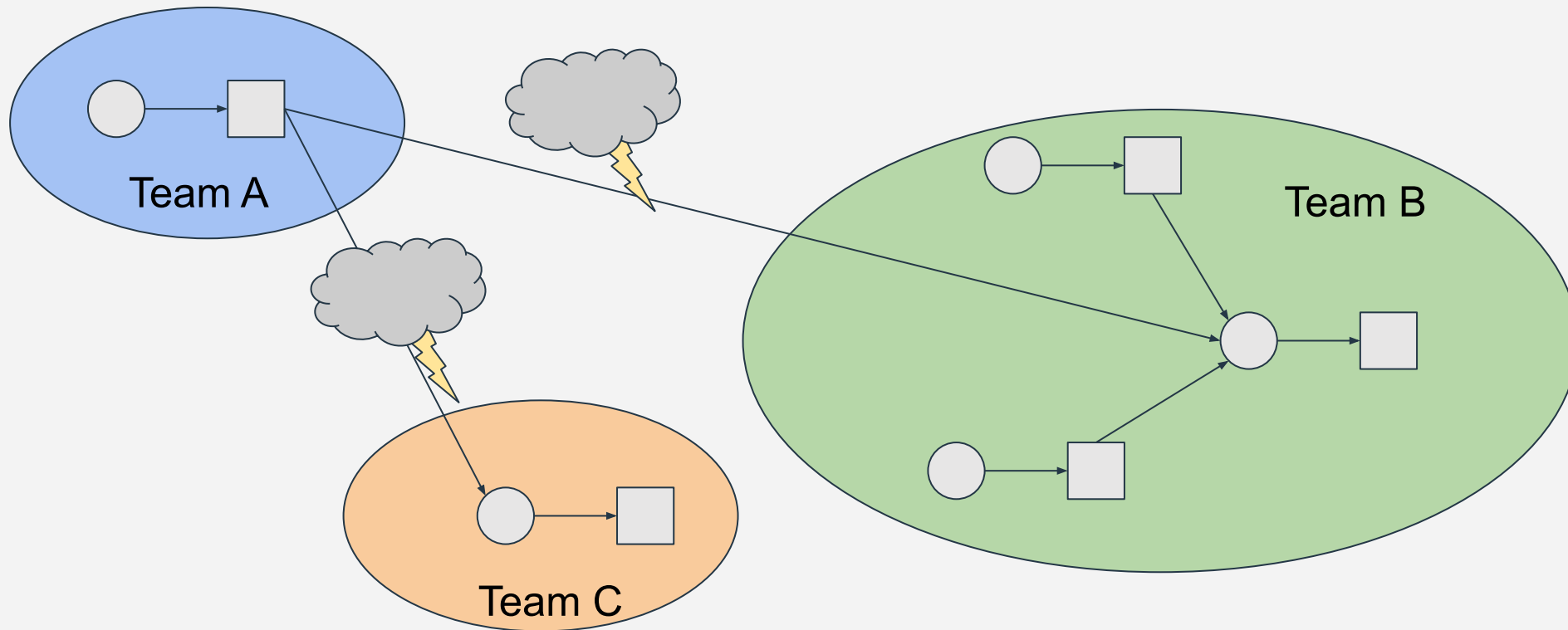
Why metadata?

Open Lineage and Marquez
Community

Why Metadata?

Need to create a **healthy**
data ecosystem

Team interdependencies



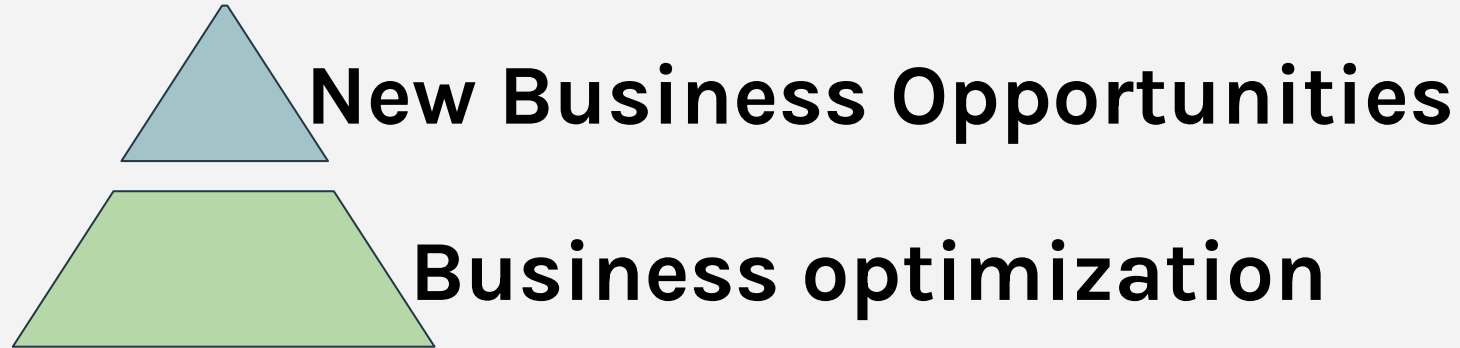
Today: Limited context



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?

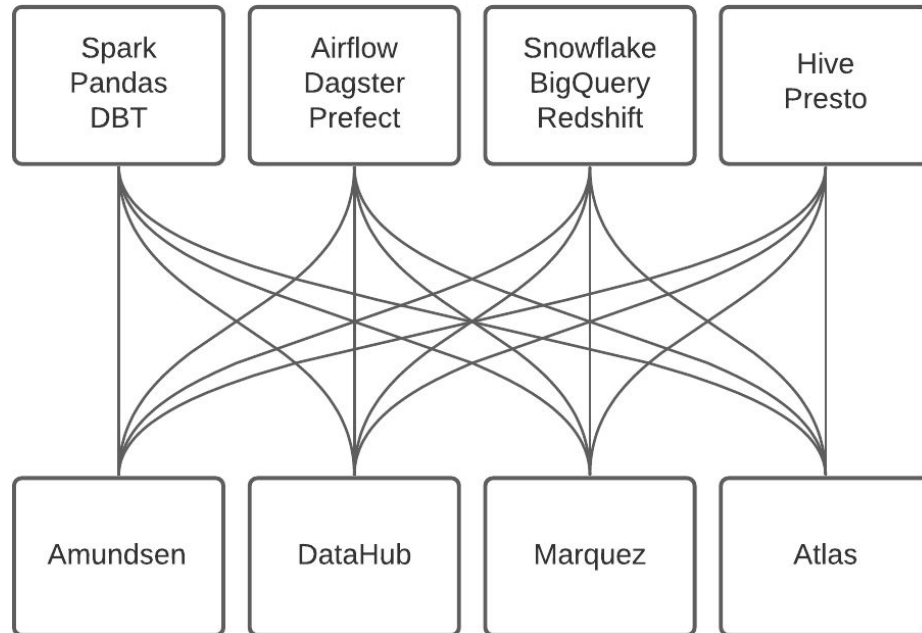
~~Maslow's~~ Data hierarchy of needs



Open Lineage

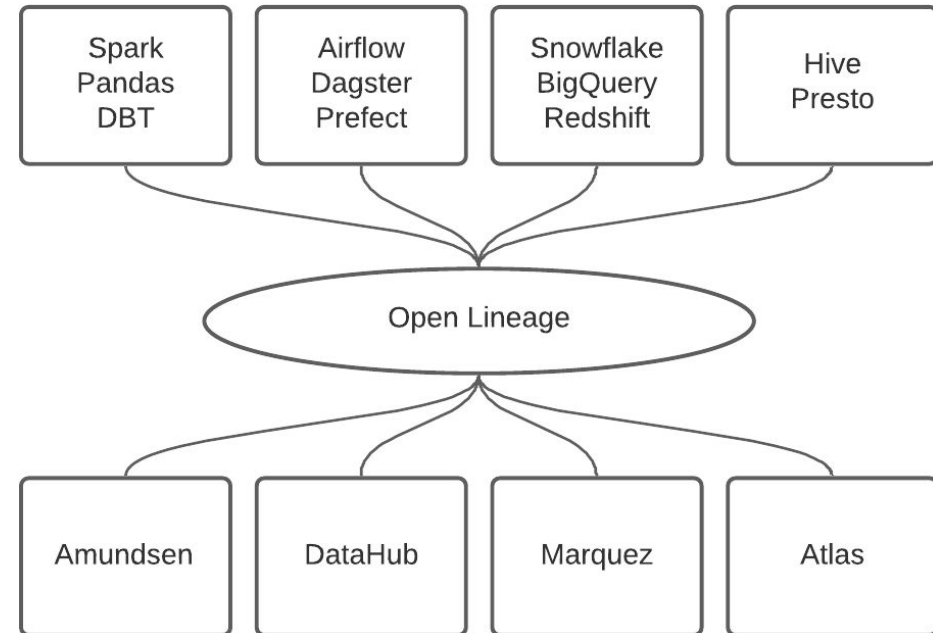
Problem

Today:



- Duplication of effort: Each project has to instrument all jobs
- Integrations are external and can break with new versions

With Open Lineage



- Effort of integration is shared
- Integration can be pushed in each project: no need to play catch up

Purpose

- Open standard for metadata and lineage collection
- Instrument jobs as they are running
- Define a generic model of job/dataset/runs entities
- Consistent naming strategies for jobs and datasets
- Define specific facets that can enrich those entities

Projects involved in Open Lineage (so far)



great_expectations

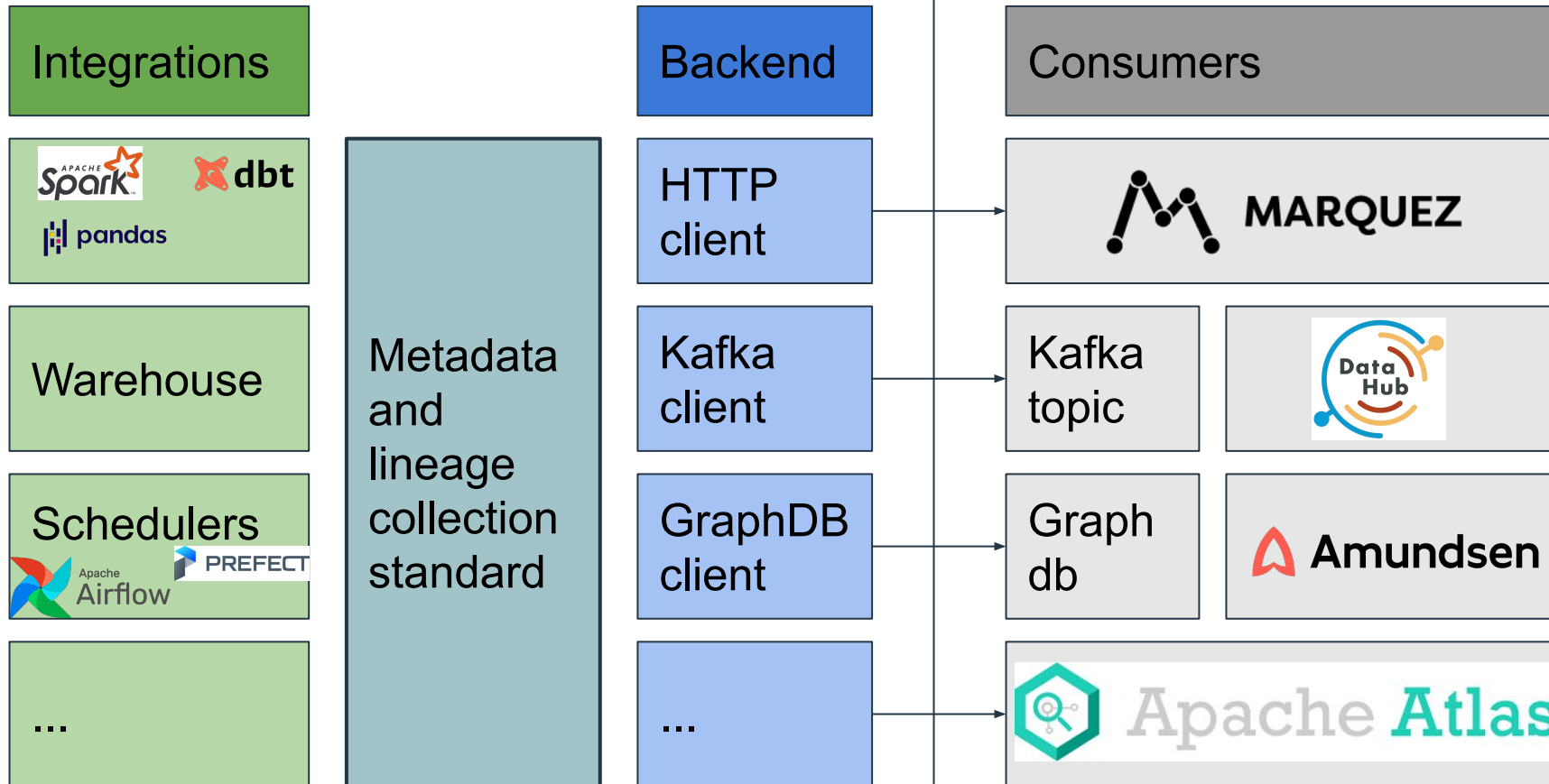


DAGSTER

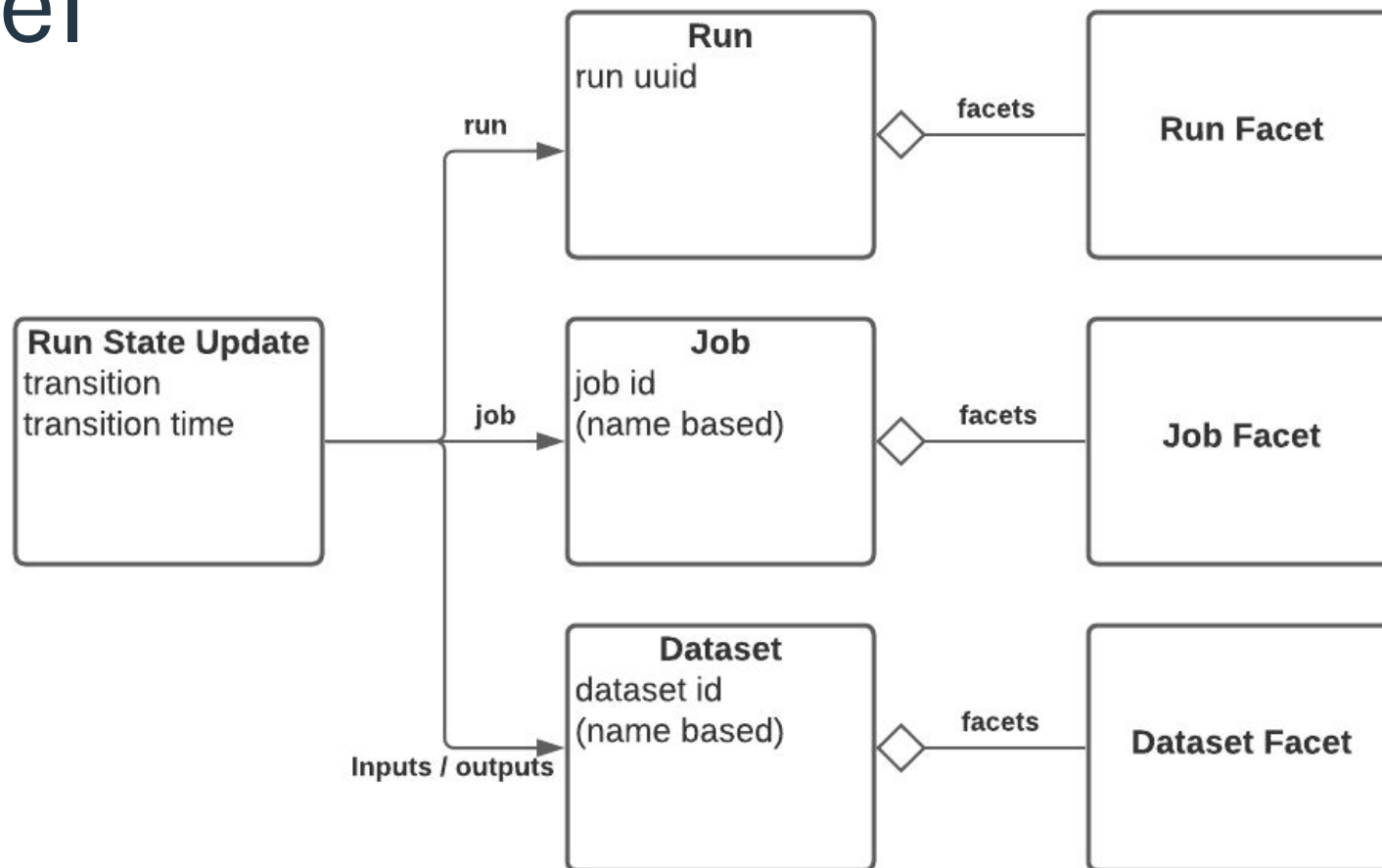


Open Lineage scope

Not in scope



Core Model



Core Model

Consistent naming:

- Jobs:

Example: scheduler.job.task

- Datasets:

Example: instance.schema.table

Facets

Facets are atomic pieces of metadata identified by a unique name that can be attached to the core entities.

Prefixes in facet names allow the definition of Custom facets that can be promoted to the spec at a later point.

Facet examples

Dataset:

- Stats
- Schema
- Version
- Column level lineage

Job:

- Source code
- Dependencies
- params
- Source control
- Query plan
- Query profile

Run:

- Schedule time
- Batch id

Protocol

- Asynchronous events
 - unique id for identifying a run and correlate events
- Configurable backend
 - Kafka
 - Http
 - ...

Lifecycle

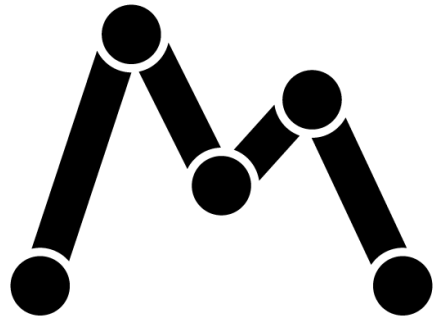
- Create unique run id
- Run start event
 - Send plan/profile info
- Run complete event
 - Send output Dataset version updates

Join the conversation

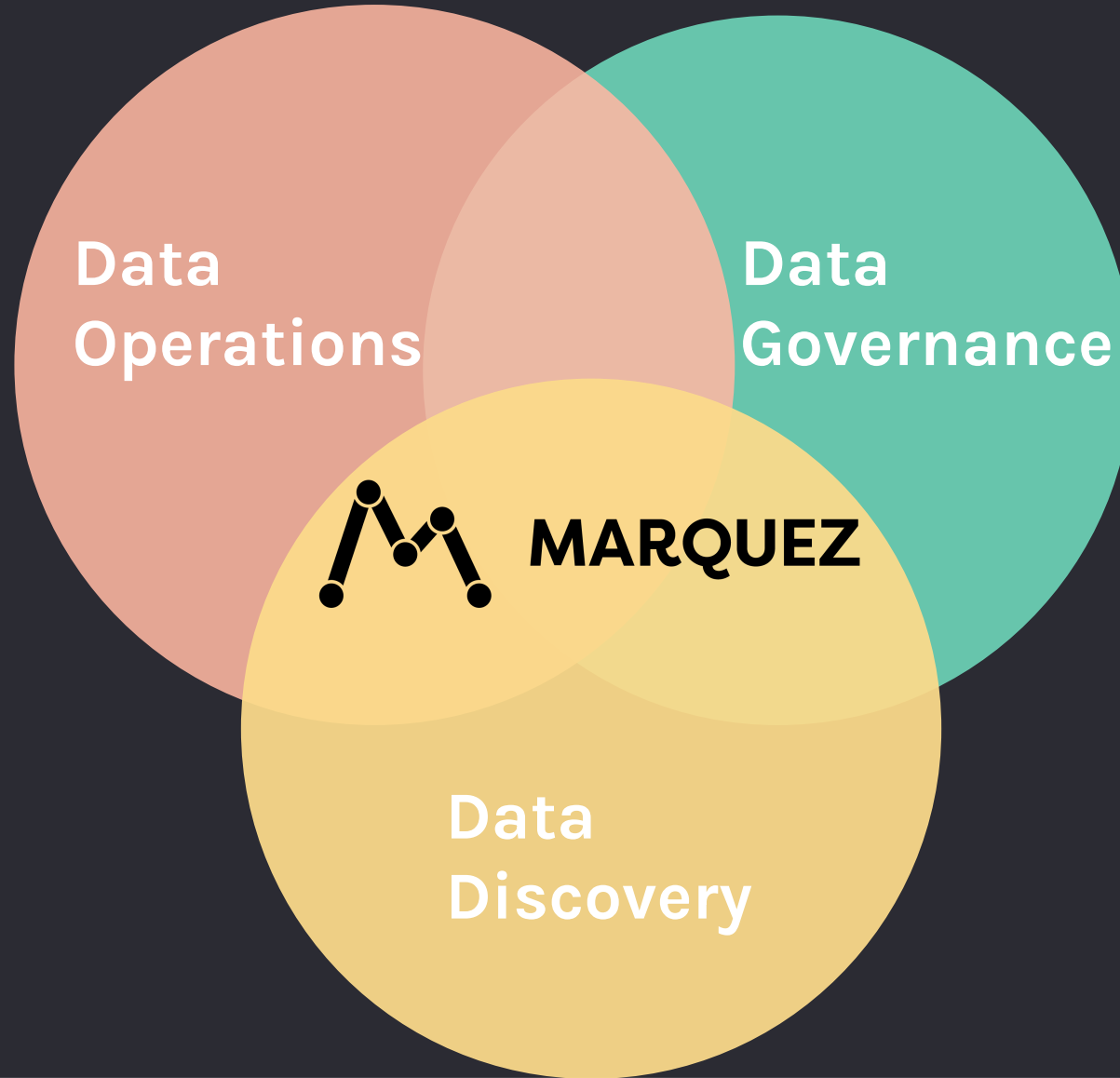
Github: <https://github.com/OpenLineage>

Slack: [OpenLineage.slack.com](https://openlineage.slack.com)

Email: <https://groups.google.com/g/openlineage>



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

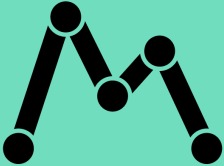
Traditional database management systems were developed in an era of risk-averse design. The technology itself was expensive, as was the on-site cost of managing it. Expertise was scarce and concentrated in a handful of computing and consulting firms.

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

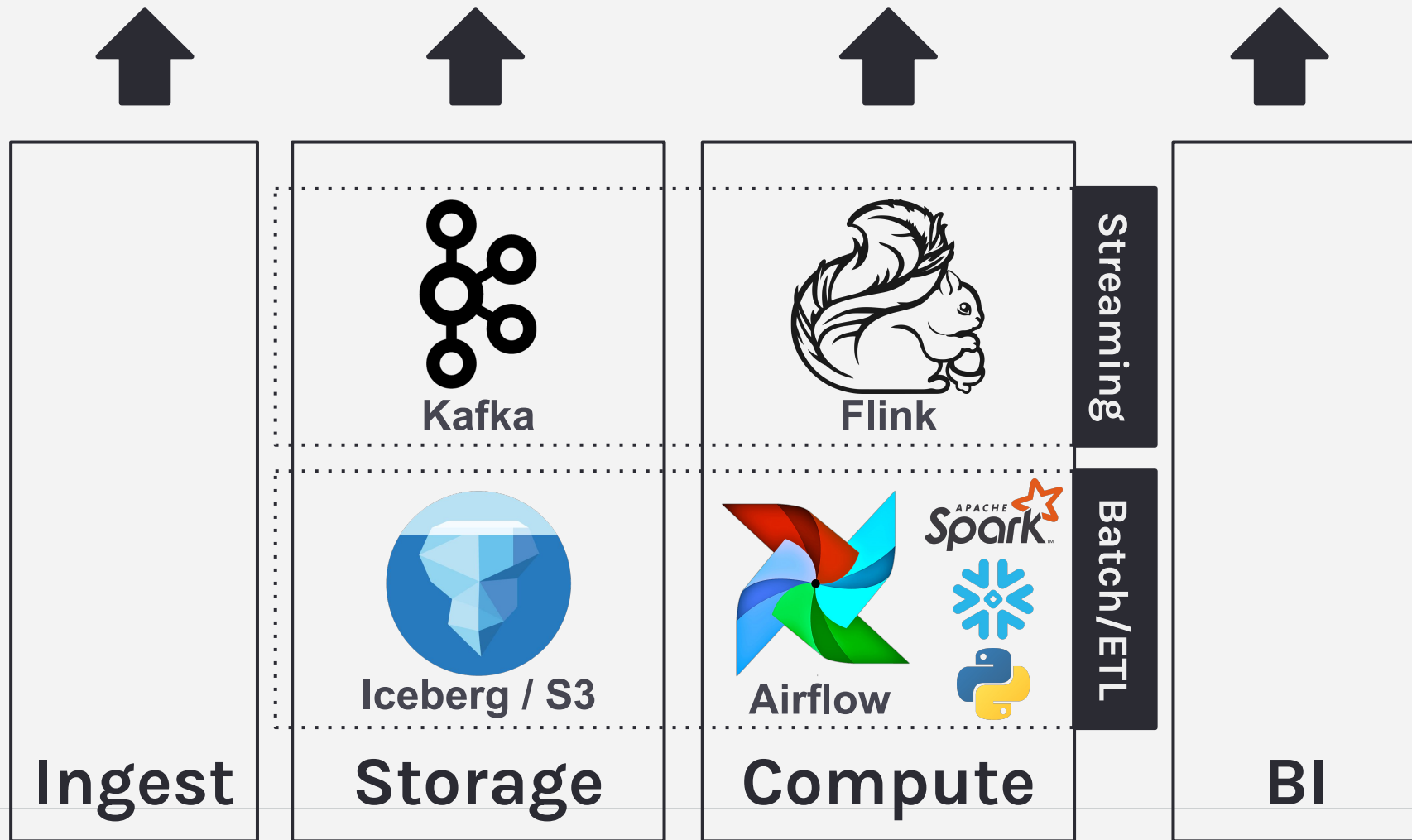
An unfortunate consequence of the disaggregated nature of contemporary data systems is the lack of a standard mechanism to assemble a collective understanding of the origin, scope, and usage of the data they manage. In the absence of a better solution to this pressing need, the Hive Metastore is sometimes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

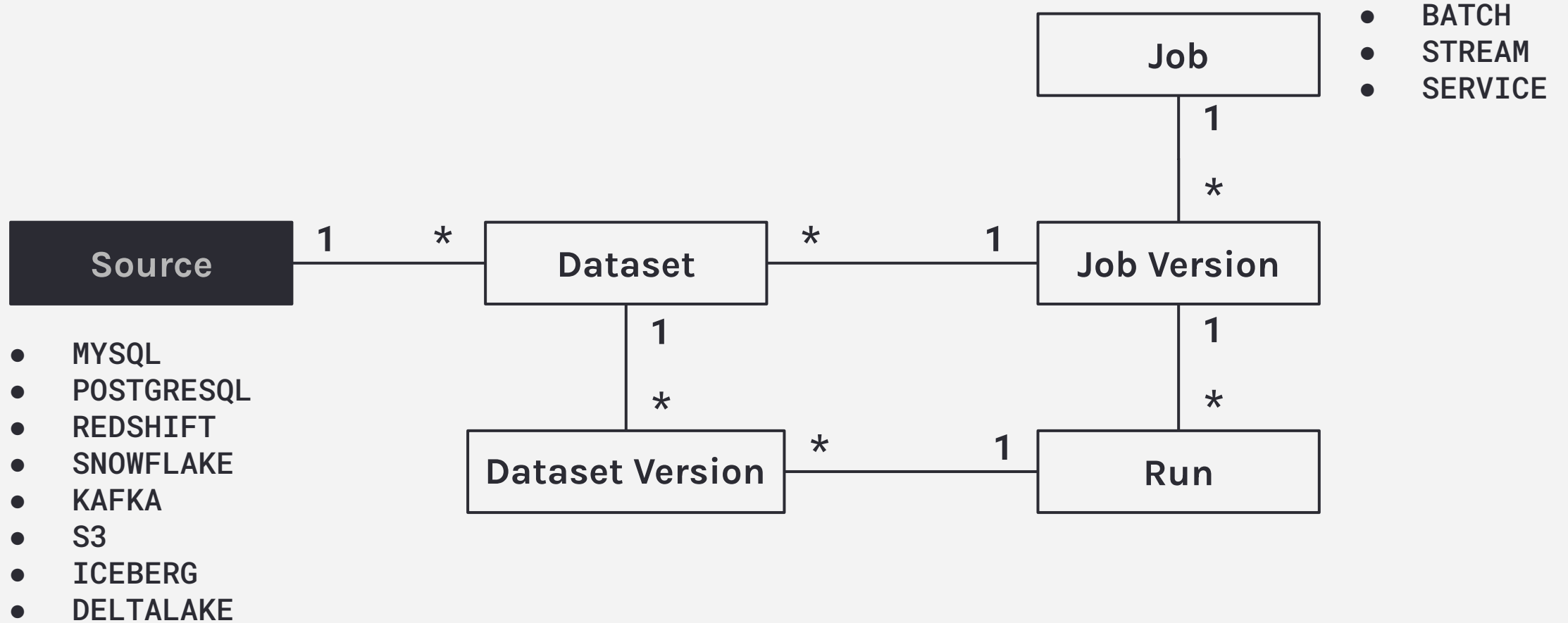


Metadata:  **MARQUEZ**

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

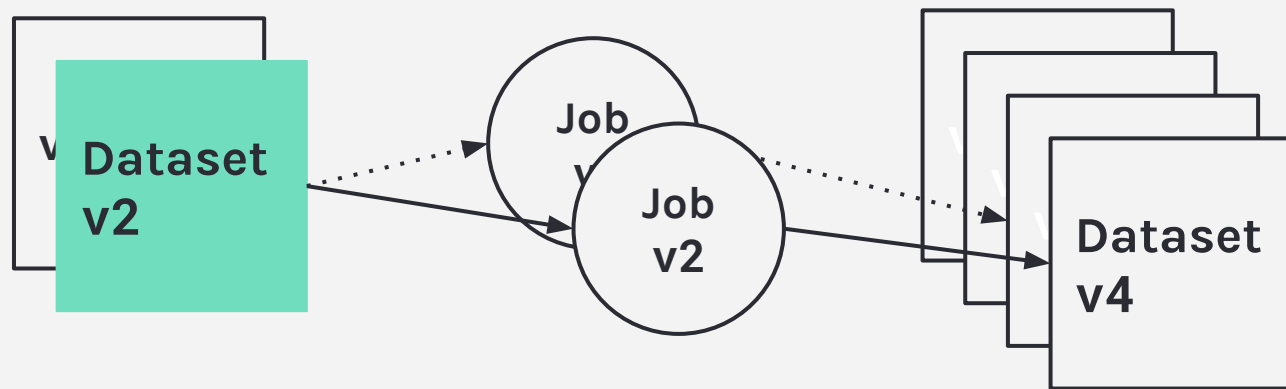
- **Integrations**
 - Ingest
 - Storage
 - Compute





Design benefits

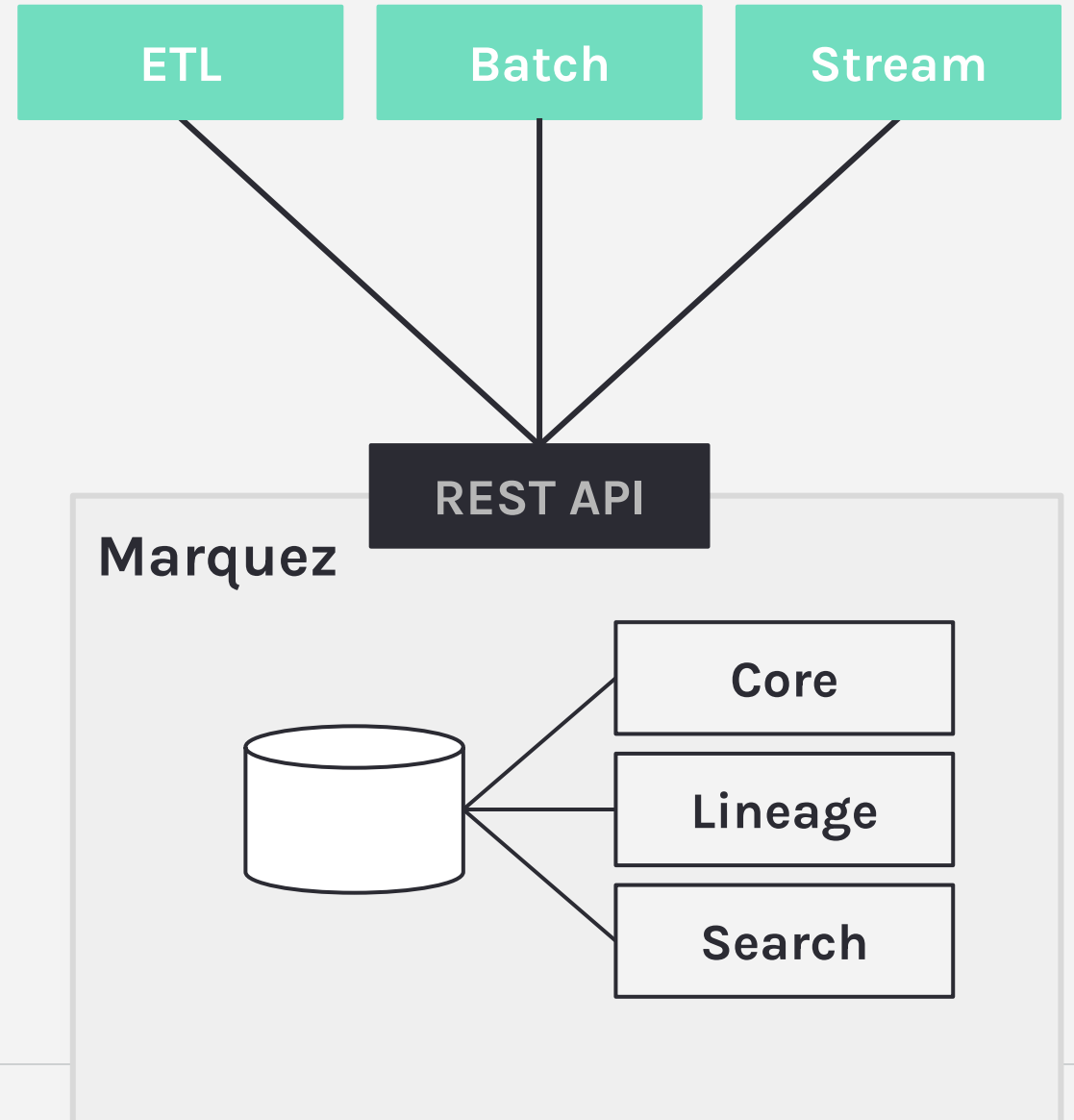
- Debugging
 - What **job version(s)** produced and consumed **dataset version X**?



- Backfilling
 - Full / incremental processing

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

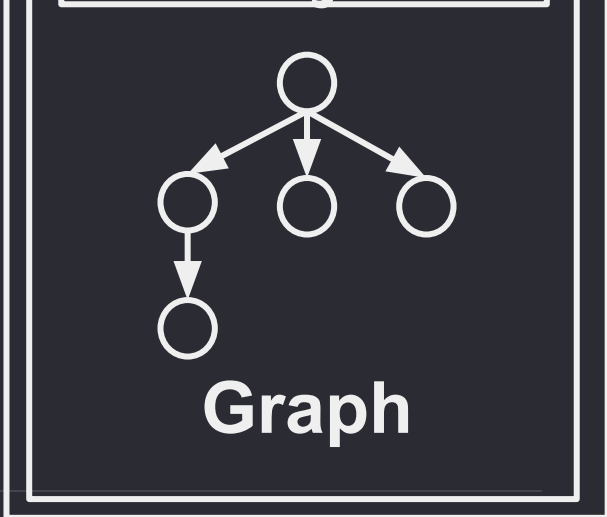
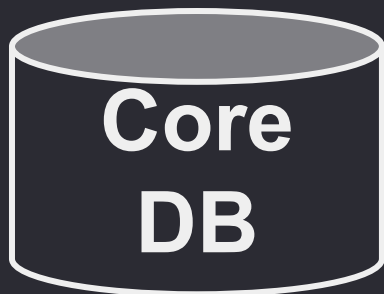
Core API

datakin

Metadata

Lineage analysis

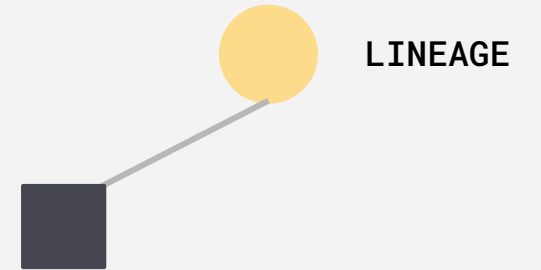
Storage



01



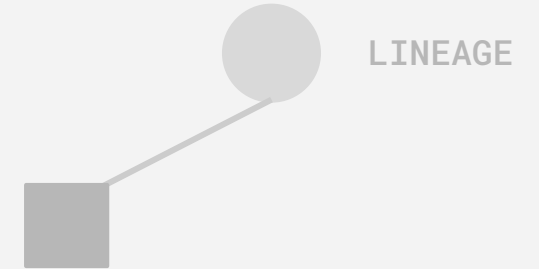
```
{  
  "type": "BATCH",  
  "name": "room_bookings_7_days"  
  "inputs": [{  
    "namespace": "datascience",  
    "name": "room_bookings"  
  }],  
  "outputs": [],  
  ...  
}
```



01



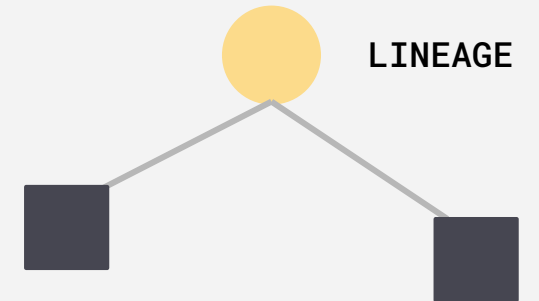
```
{  
  "type": "BATCH",  
  "name": "room_bookings_7_days",  
  "inputs": [{  
    "namespace": "datascience",  
    "name": "room_bookings"  
  }],  
  "outputs": [],  
  ...  
}
```



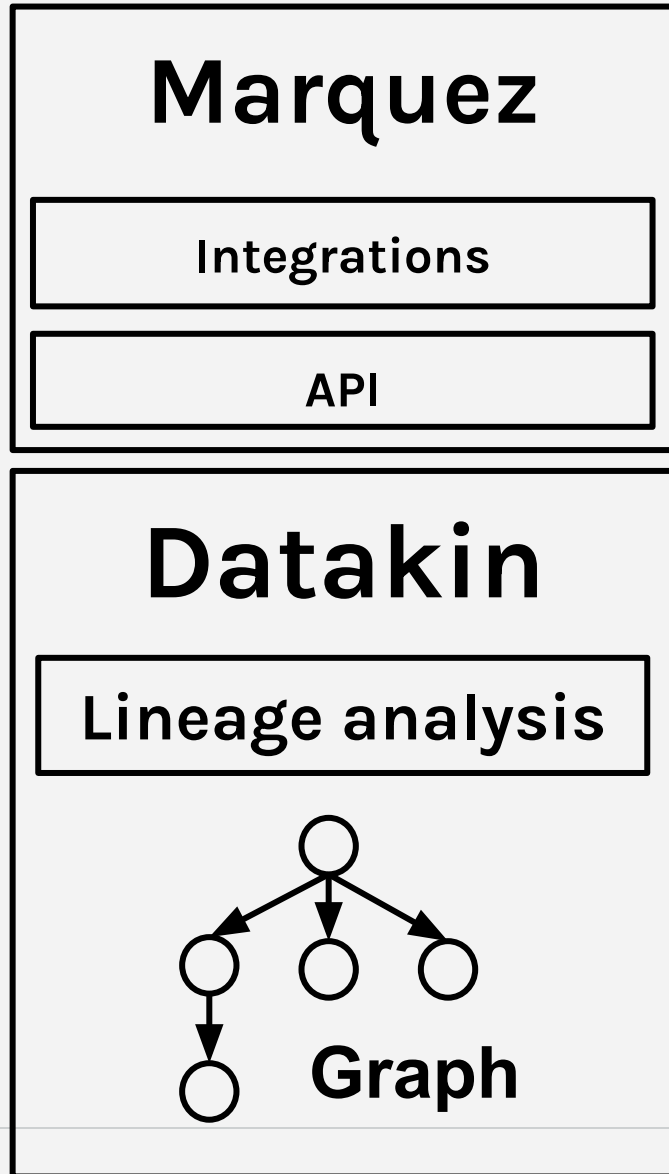
02



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



Datakin leverages Marquez metadata



- **Open Lineage and Marquez standardize 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?

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>

Part of the LF AI & Data 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/marquez ★



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

Repositories 8 Packages People 14 Teams 2 Projects Settings

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

marquez-airflow
Airflow support for Marquez
airflow python3
Python Apache-2.0 12 32 11 (1 issue needs help) 2 Updated 5 hours ago

marquez
Collect, aggregate, and visualize a data ecosystem's metadata
metadata data-catalog data-discovery data-dictionary
data-governance data-lineage data-provenance
Java Apache-2.0 55 413 40 3 Updated 9 hours ago

marquez-python Archived
Python client for Marquez
deprecated
Python Apache-2.0 11 13 0 0 Updated 5 days ago

Top languages
Python Java Shell TypeScript HTML

Most used topics Manage
deprecated marquez metadata

People 14 >
Invite someone



Thank You

