Data Pipelines Observability OpenLineage & Marquez

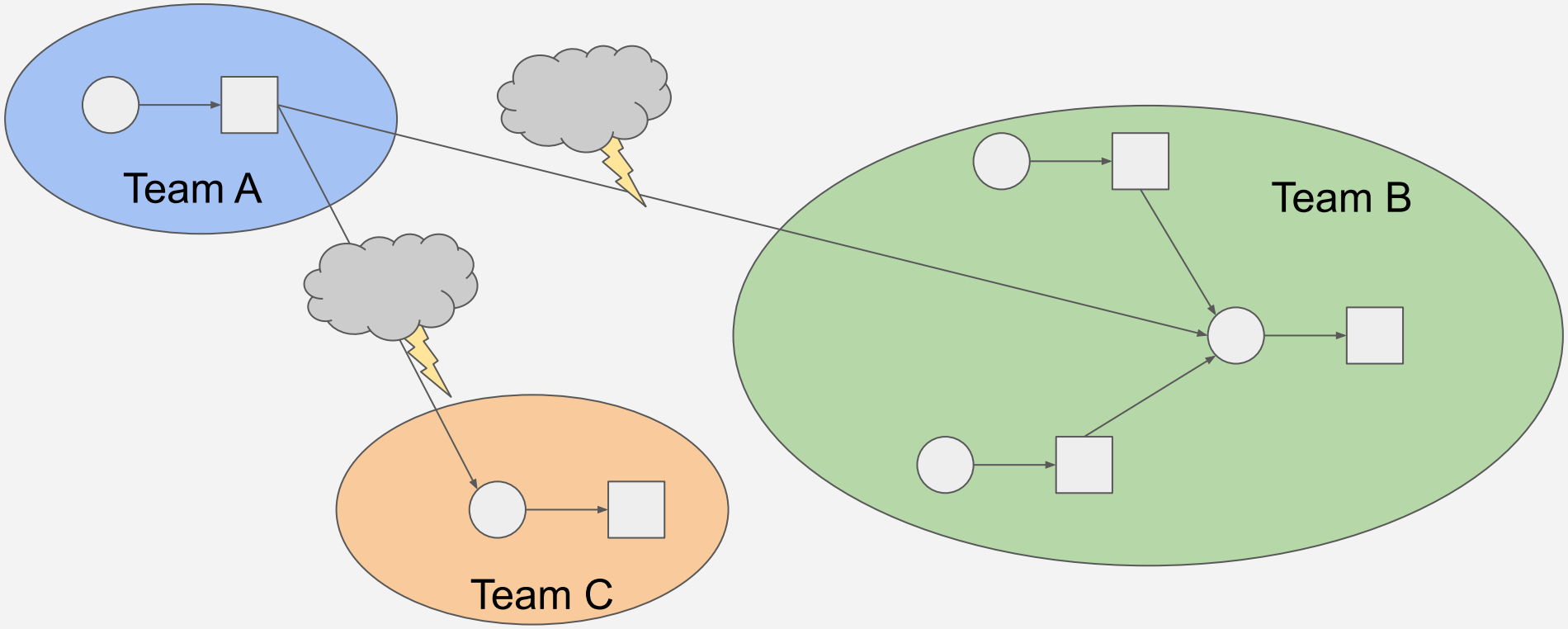
Julien Le Dem
CTO & Co-Founder Datakin
@J_

AGENDA

- The need for metadata
- OpenLineage: open standard for metadata and lineage collection
- Marquez: a reference implementation

The need for Metadata

Building a healthy data ecosystem



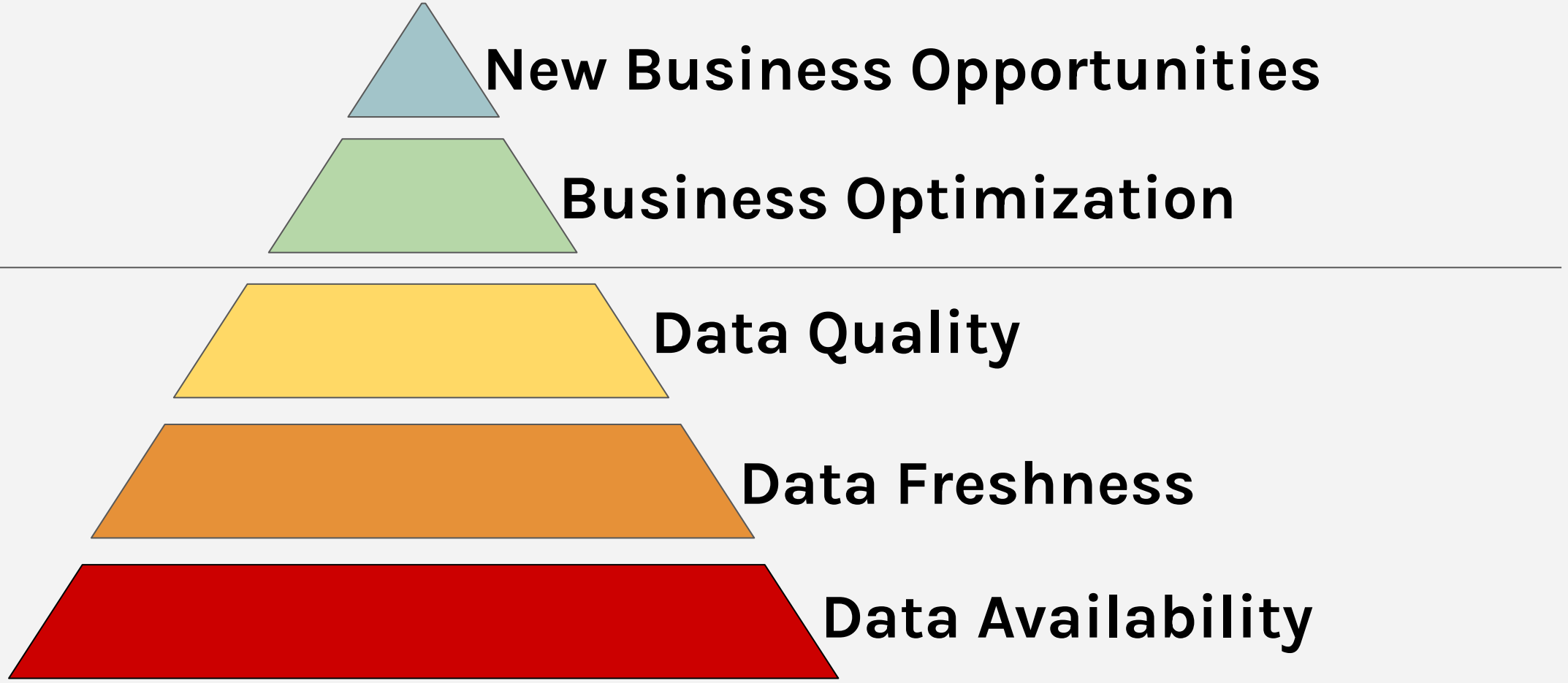
datakin

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



datakin

OpenLineage

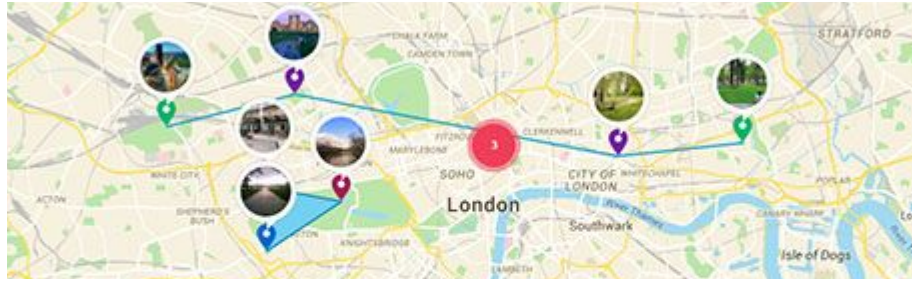
OpenLineage contributors

Creators and contributors from major open source projects involved

Purpose

Define an Open standard for metadata and lineage collection by instrumenting data pipelines as they are running.

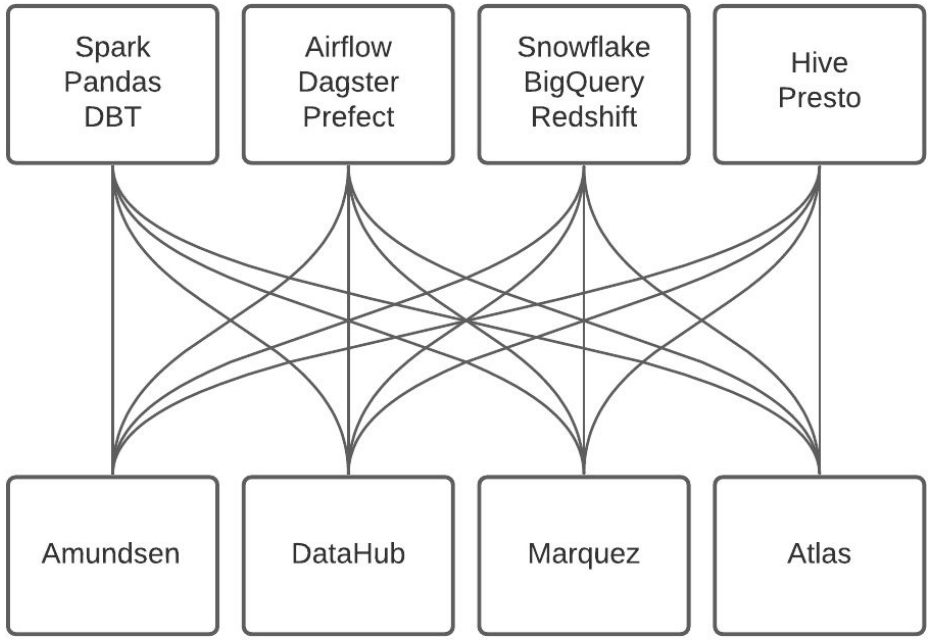
Purpose: EXIF for data pipelines



datakin

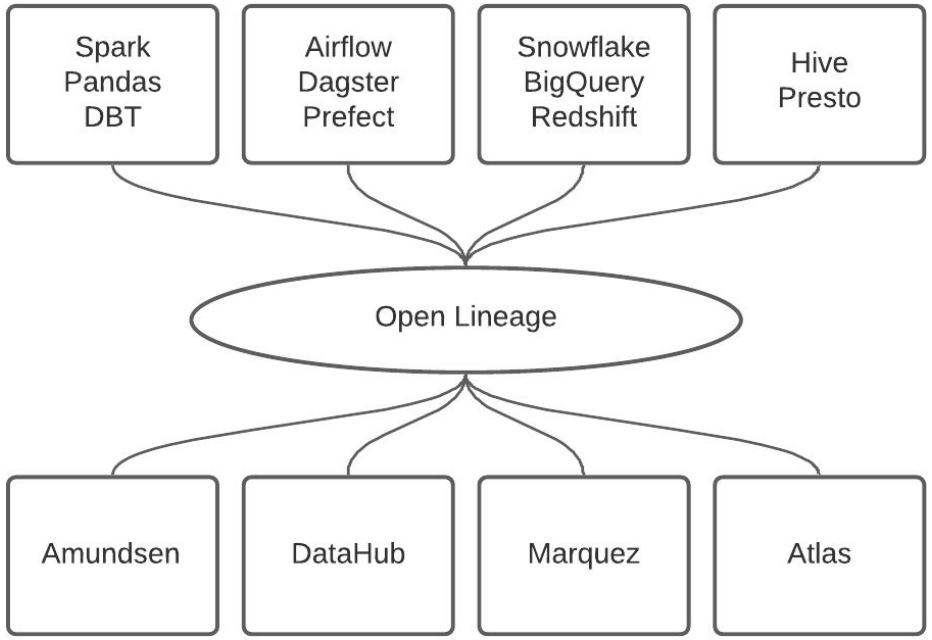
Problem

Before:



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

With Open Lineage

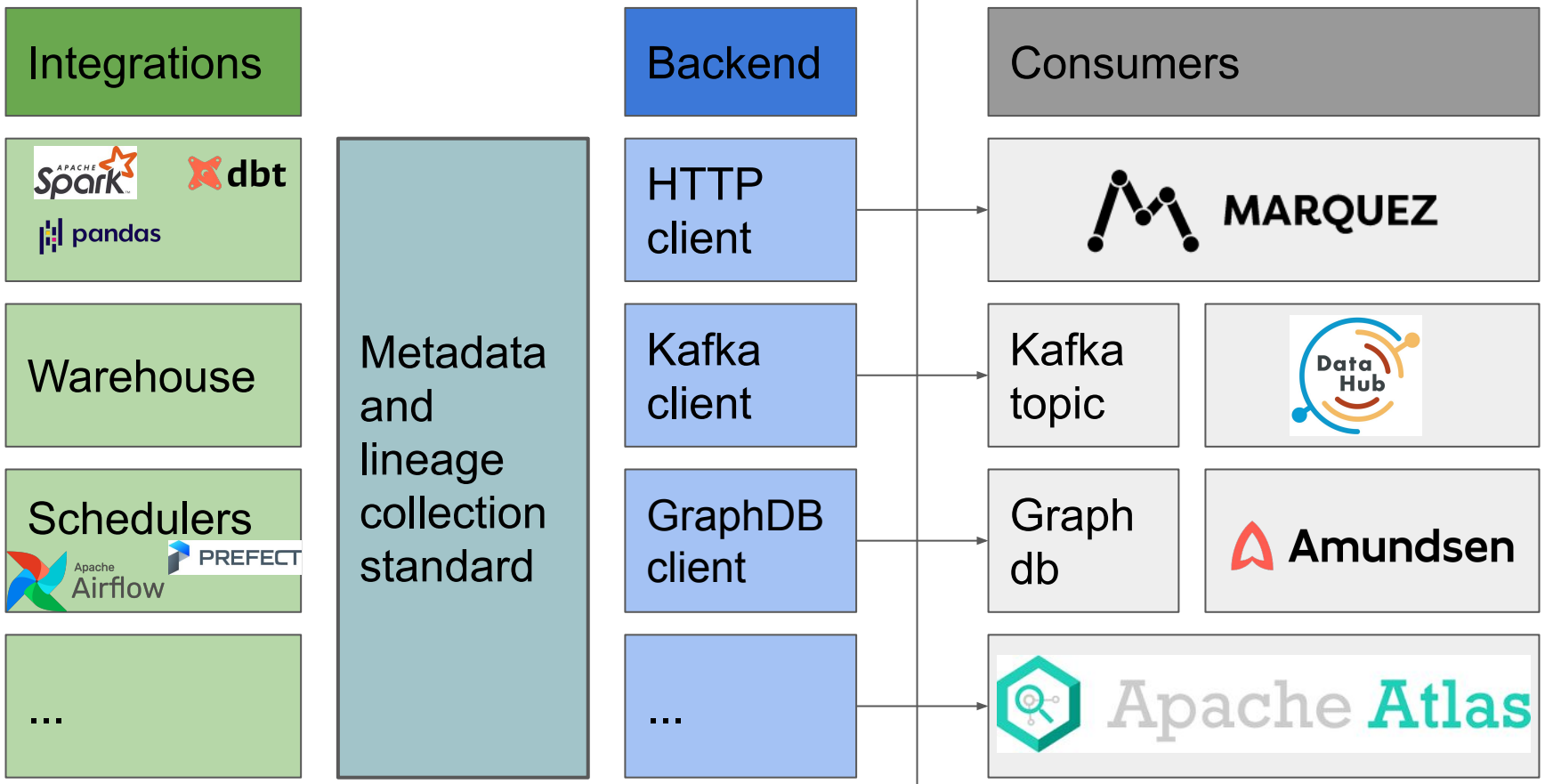


- Effort of integration is shared
- Integration can be pushed in each project: no need to play catch up

datakin

Open Lineage scope

Not in scope



datakin

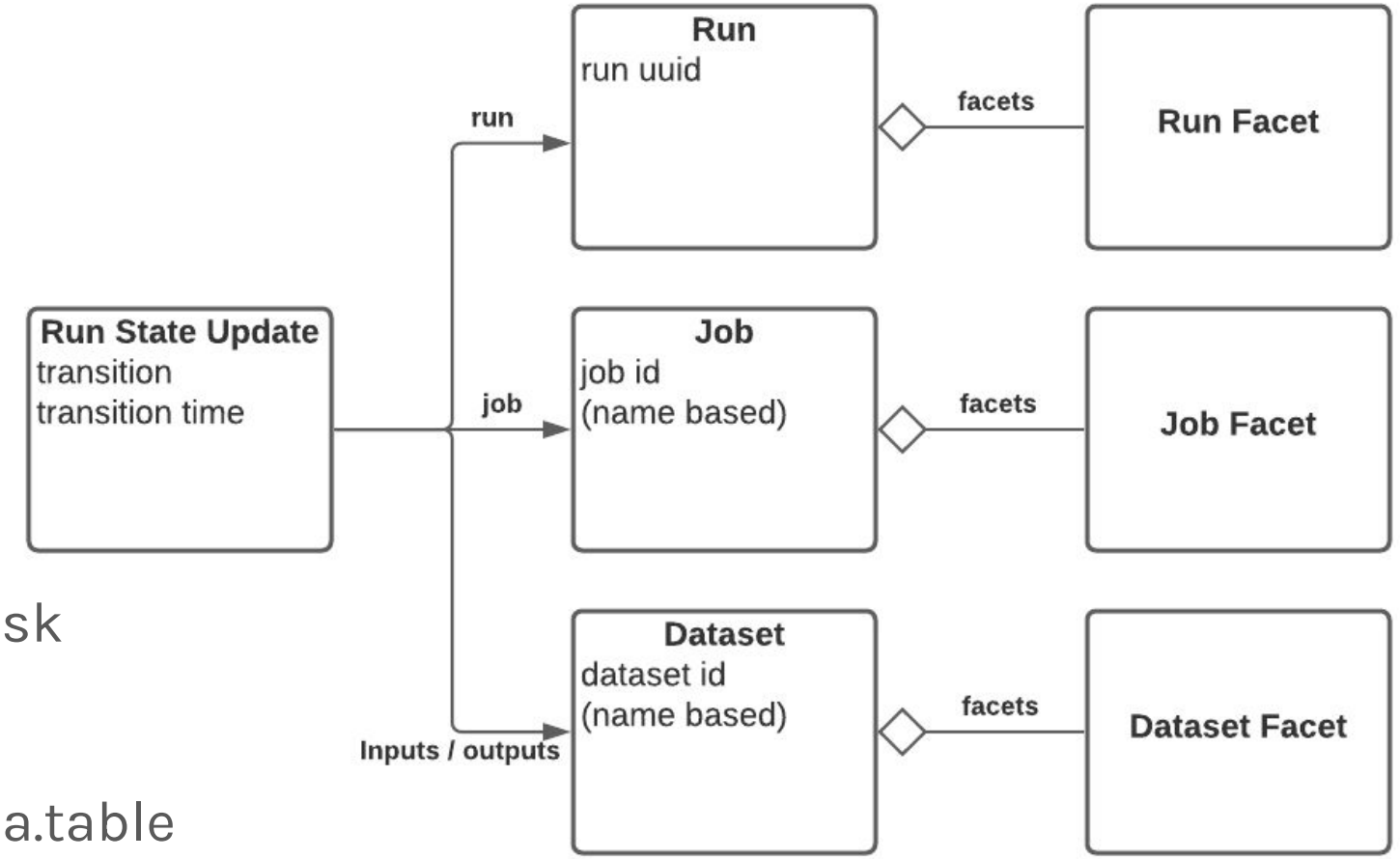
Core Model

- JSONSchema spec
- Consistent naming:
  - Jobs:

    Example: scheduler.job.task

  - Datasets:

    Example: instance.schema.table



datakin

Protocol

- Asynchronous events: unique run id for identifying a run and correlate events
  - Run Start event
    - source code version
    - run parameters
  - Run Complete event
    - input dataset
    - output dataset version and schema
- Configurable backend
  - Kafka
  - Http
  - …

Facets

- Extensible:

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

- Decentralized:

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

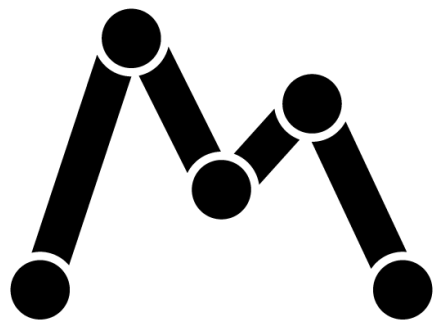
Facet examples

Dataset:
- Stats
- Schema
- Version
- Column level lineage

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

Run:
- Schedule time
- Batch id

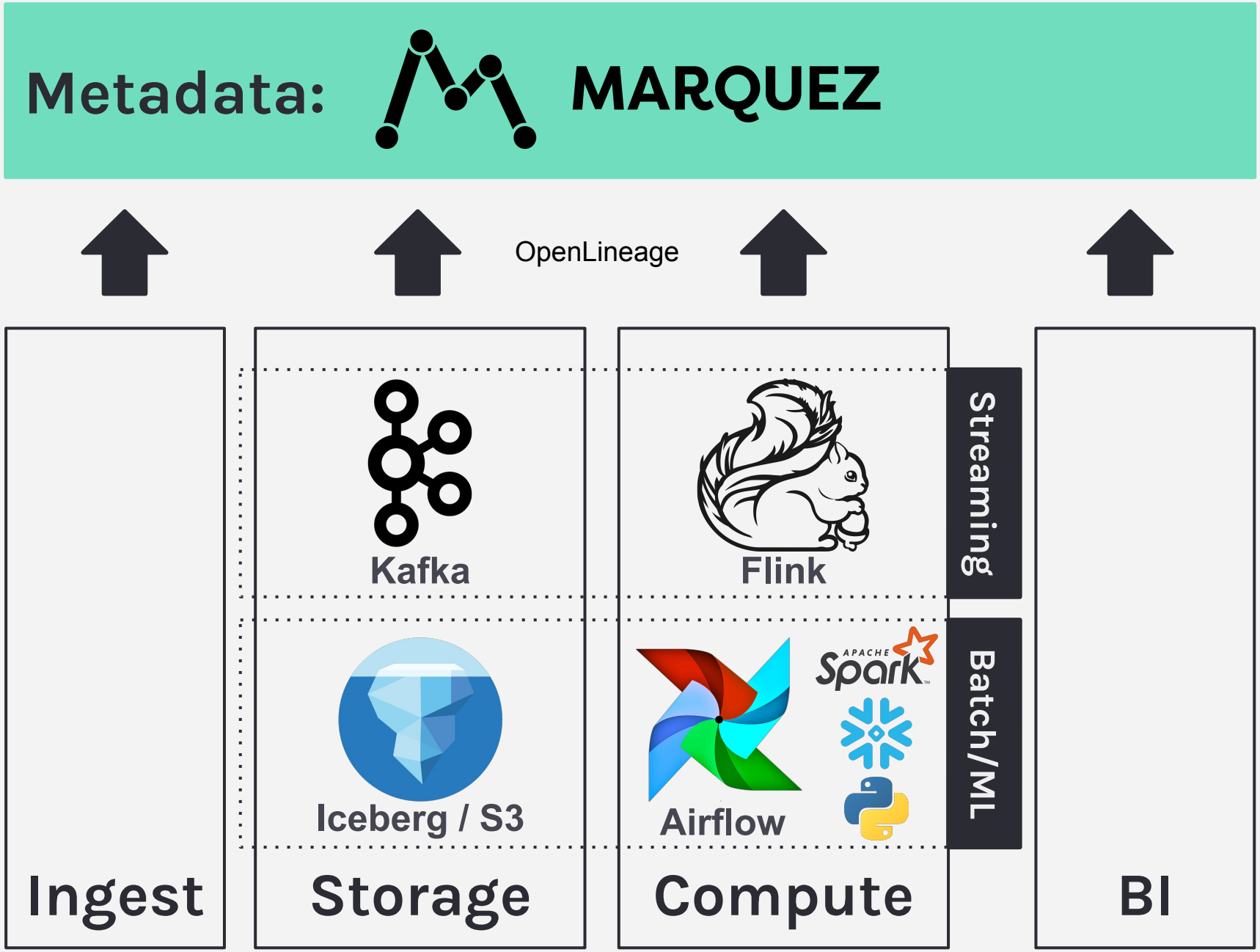
datakin



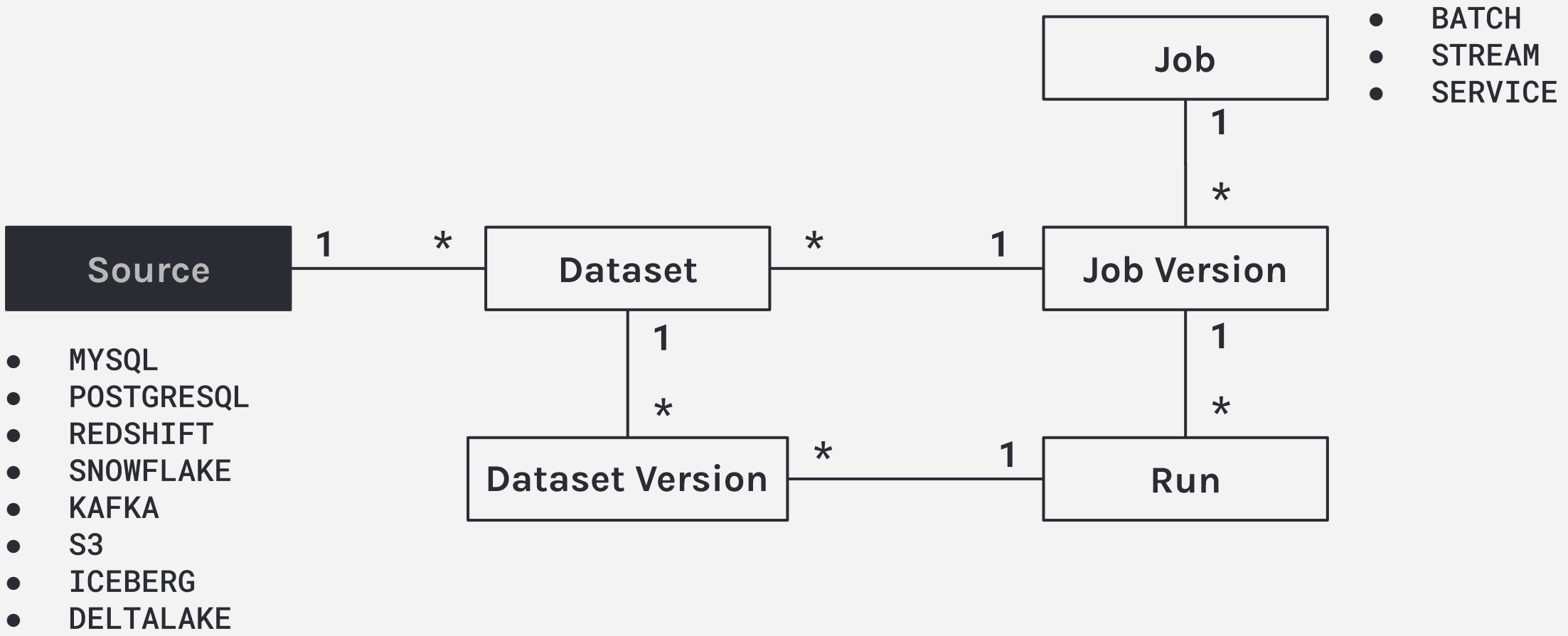
MARQUEZ

datakin

- **Data Platform** built around **Marquez**
- **Integrations**
  - Ingest
  - Storage
  - Compute

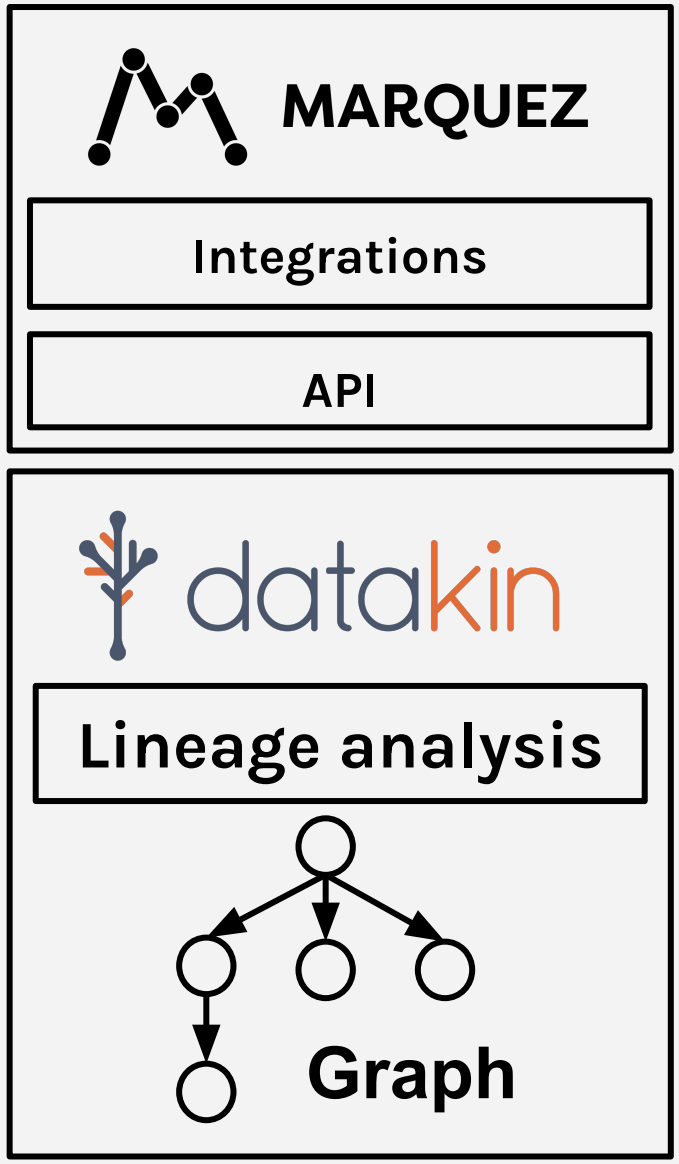


Marquez: Data model



MARQUEZ

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?

Join the conversation

OpenLineage:

- **Github:** github.com/OpenLineage ★
- **Slack:** OpenLineage.slack.com
- **Twitter:** @OpenLineage
- **Email:** groups.google.com/g/openlineage

Marquez:

- **Github:** github.com/MarquezProject/marquez ★
- **Slack:** MarquezProject.slack.com
- **Twitter:** @MarquezProject

datakin

Thank You

*we're hiring! jobs@datakin.com