Disrupting Data Discovery

May 1st, 2019

Mark Grover | @mark_grover | Product Management, Lyft

go.lyft.com/datadiscoveryslides

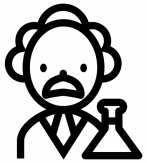
Agenda

- Data at Lyft
- Challenges with Data Discovery
- Data Discovery at Lyft
- Architecture
- Summary

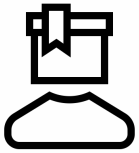
Data platform users

Data Modelers

Analysts



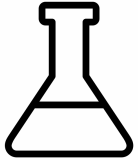
Data Scientists



Product Managers

General Managers

Engineers

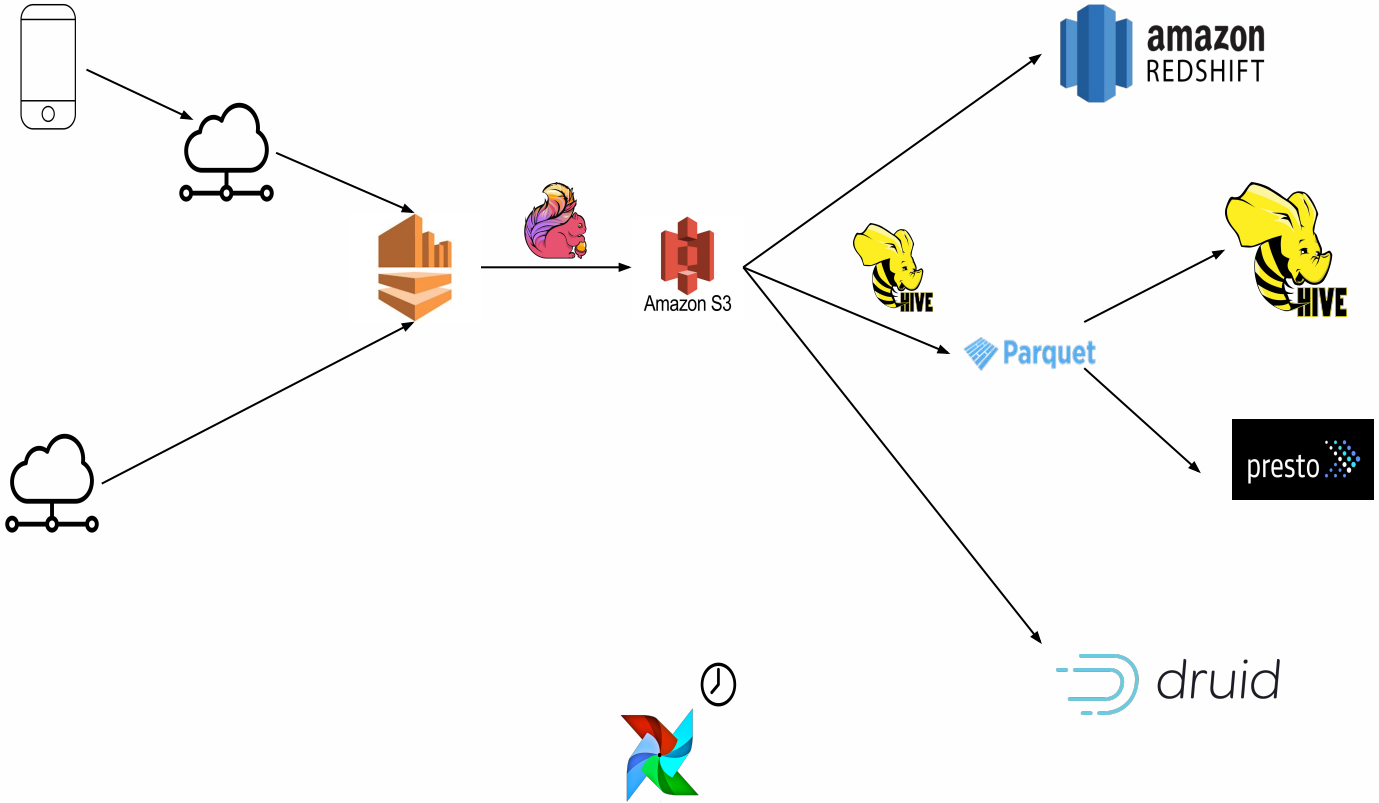


Experimenters

Lyft Data Team

Lyft Data Team					
Core Data Infra	Streaming Infra	Visualization	Experimentation	BI and Logging	ML Infra

Core Infra high level architecture



looker

MODE

+ableau

Custom apps

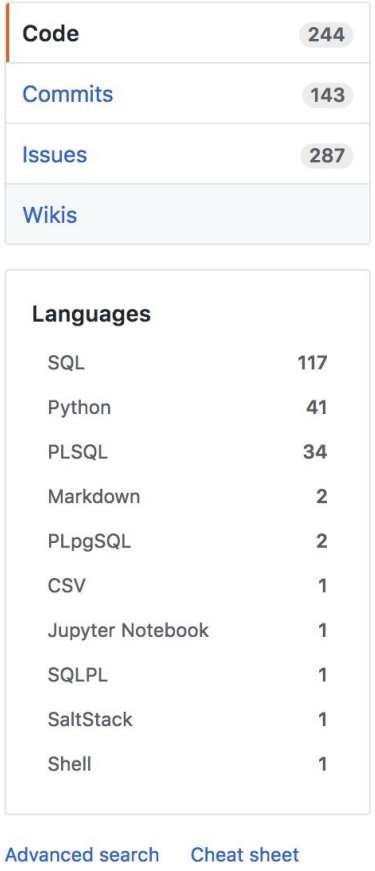
Data Discovery

Hi! I am a n00b Data Scientist!

- My first project is to analyze and predict Strata Attendance
- Where is the data?
- What does it mean?

Status quo

- Option 1: Phone a friend!
- Option 2: Github search



Understand the context

- What does this field mean?
  - Does attendance data include employees?
  - Does it include revenue?
- Let me dig in and understand

Explore

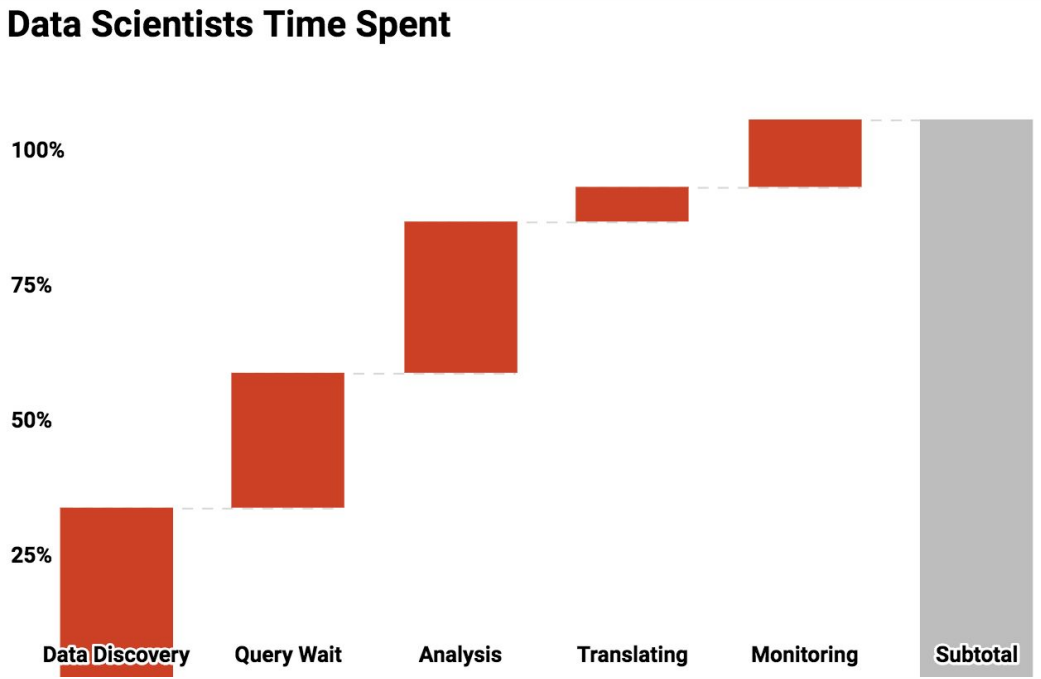
```
SELECT
*
FROM
default.my_table
WHERE ds='2018-01-01'
LIMIT 100;
```

Exploring with SELECT * is EVIL

1. Lack of productivity for data scientists
2. Increased load on the databases

Data Scientists spend upto 1/3rd time in Data Discovery...

- Data discovery
  - Lack of understanding of what data exists, where, who owns it, who uses it, and how to request access.



All about metadata

Serving more metadata about existing resources

Application Context

Existence, description, semantics, etc.

Behavior

How data is created and used over time

Change

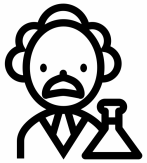
How data is changing over time

Audience for data discovery

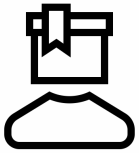
Data Discovery - User personas

Data Modelers

Analysts



Data Scientists



Product Managers

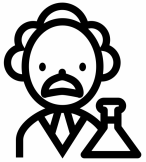
General Managers

Engineers

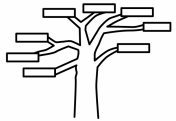
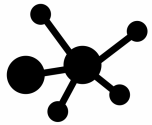
Experimenters

Data Platform

3 Data Scientist personas

 Power user	 Noob user	 Manager
• All info in their head • Get interrupted a lot due to questions	• Lost • Ask “power users” a lot of questions	• Dependencies landing on time • Communicating with stakeholders

Data Discovery answers 3 kinds of questions

Search based 	**Lineage based** 	**Network based** 
Where is the table/dashboard for X? What does it contain?	I am changing a data model, who are the owner and most common users?	I want to follow a power user in my team.
Does this analysis already exist?	This table's delivery was delayed today, I want to notify everyone downstream.	I want to bookmark tables of interest and get a feed of data delay, schema change, incidents.

Buy vs. Build vs. Adopt

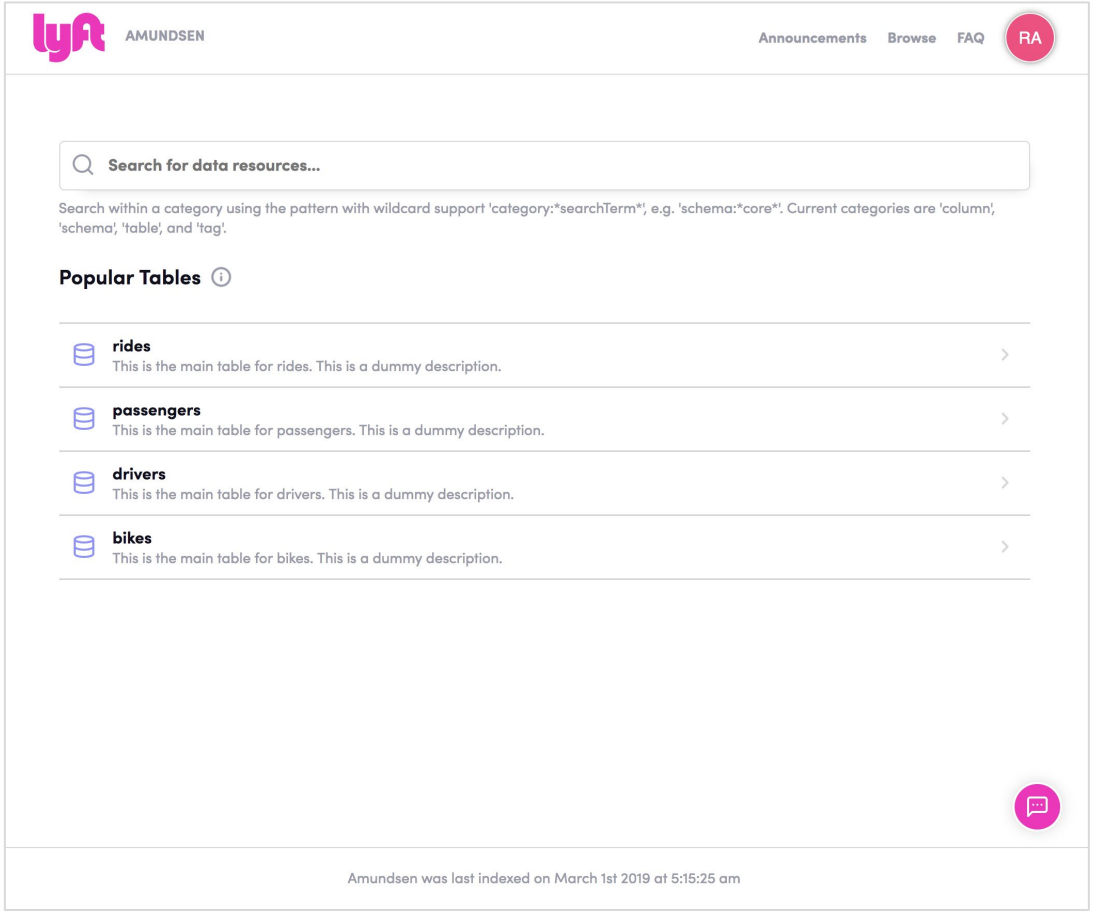
Compared various existing solutions/open source projects

Criteria / Products	Alation	Where Hows	Airbnb Data Portal	Cloudera Navigator	Apache Atlas
Search based	✓	✓	✓	✓	✓
Lineage based	✓	✓	✓	✓	✓
Network based	✗	✗	✓	✗	✗
Hive/Presto support	✓	✓	✓	✓	✓
Redshift support	✗	✗	✗	✗	✗
Open source (pref.)	✗	✓	✗	✗	✓

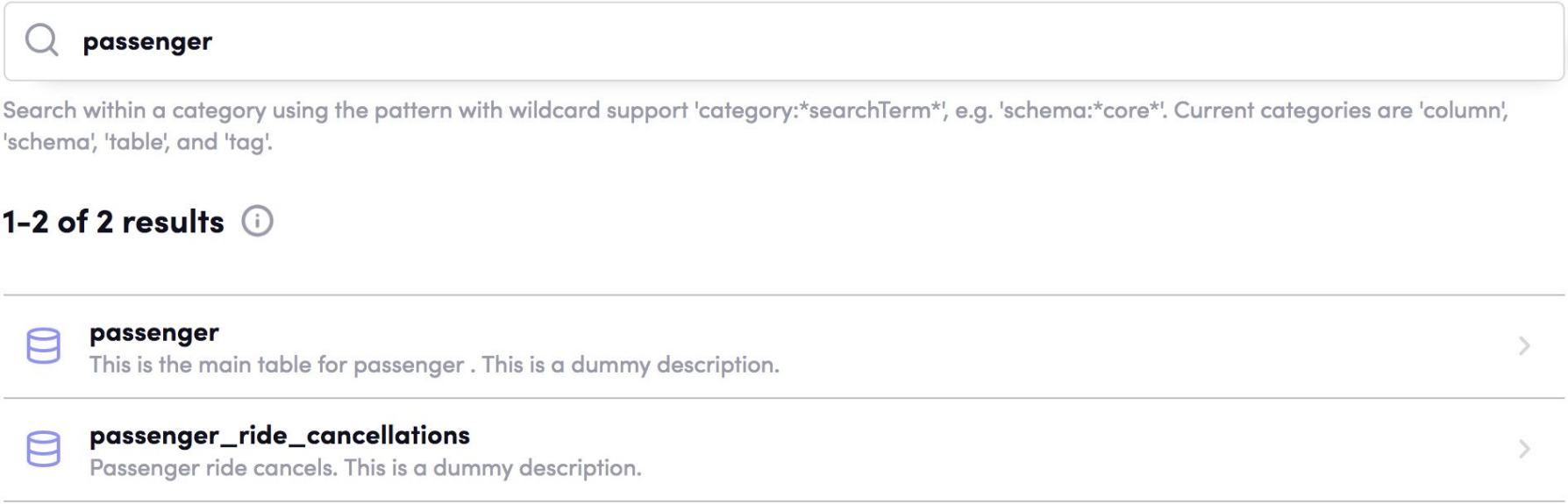
Meet Amundsen

First person to discover the South Pole - Norwegian explorer, Roald Amundsen

Landing page optimized for search



Search results ranked on relevance and query activity



How does search work?

Relevance - search for “apple” on Google

Low relevance

High relevance

Popularity - search for “apple” on Google

Low popularity

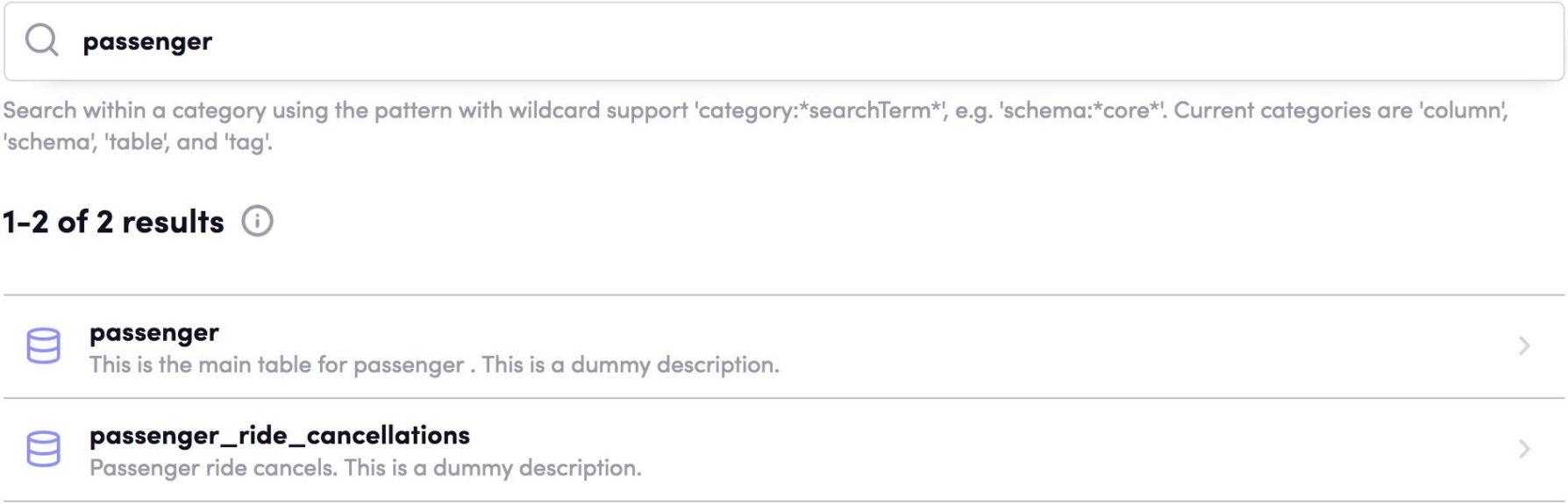
High popularity

Striking the balance

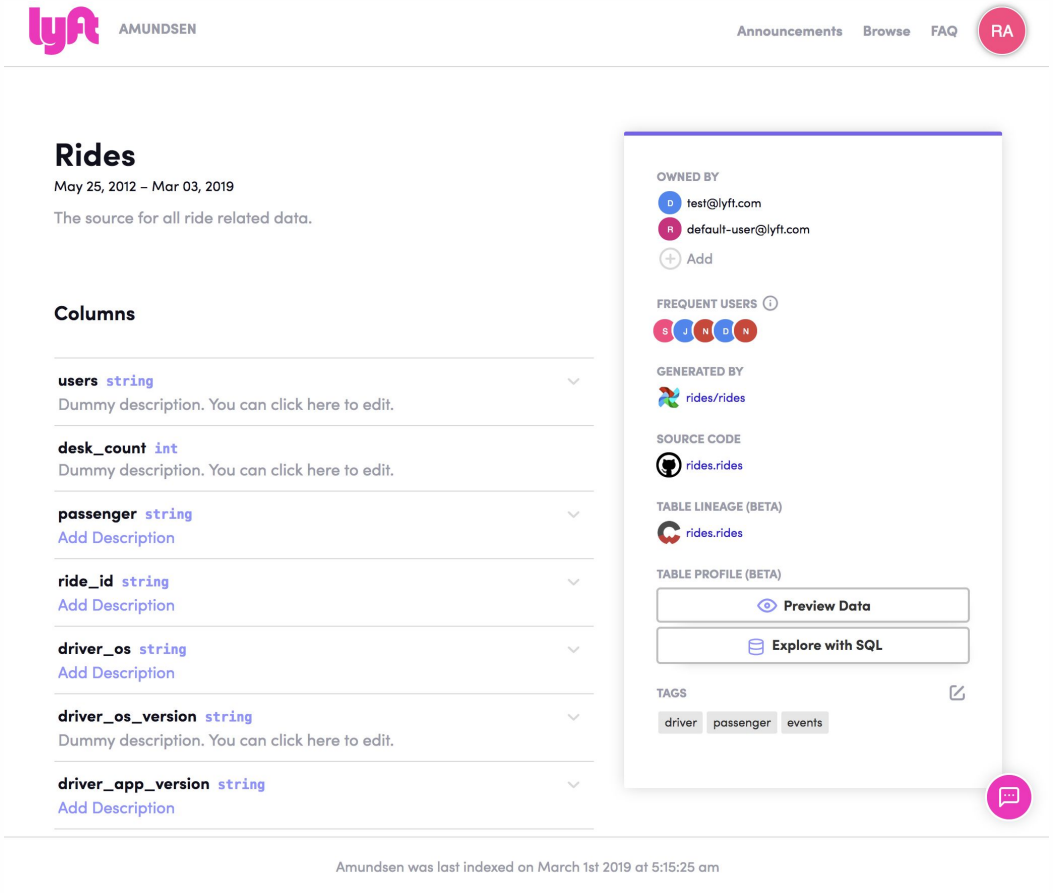
Relevance	Popularity
● Names, Descriptions, Tags, [owners, frequent users]	● Querying activity ● Dashboarding ● Different weights for automated vs adhoc querying

Back to mocks...

Search results ranked on relevance and query activity



Detailed description and metadata about data resources

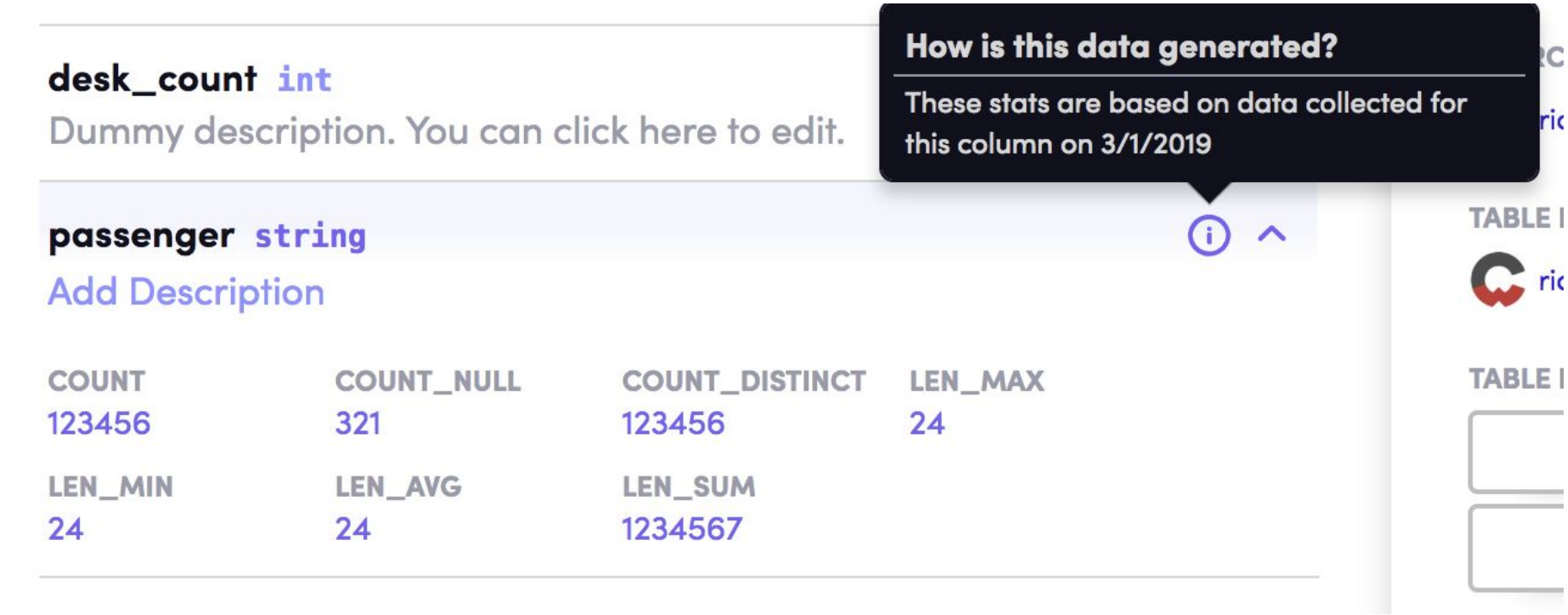


Data Preview within the tool

Rides

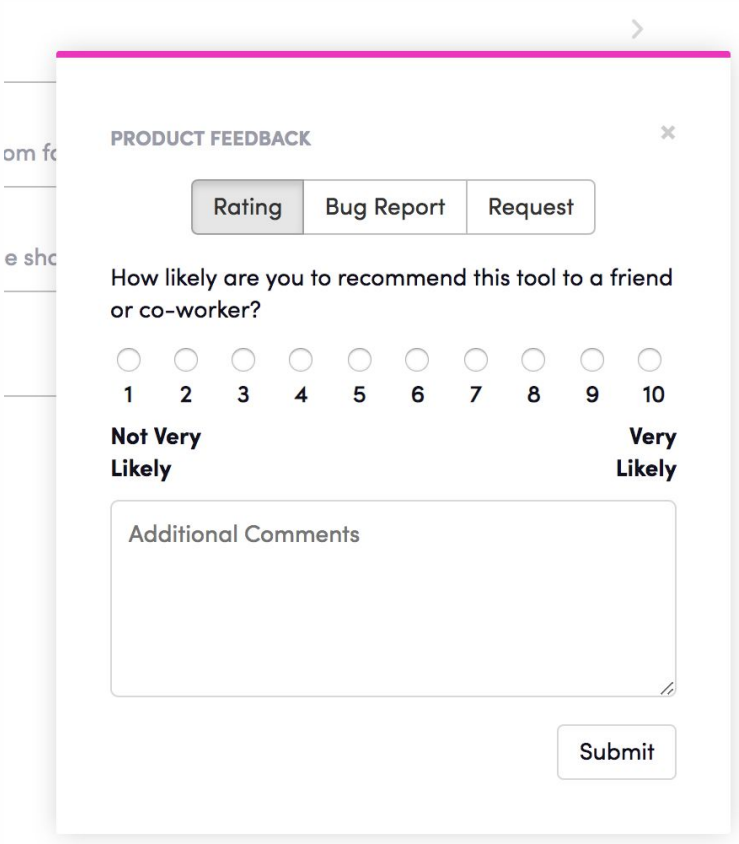
USERS	DESK_COUNT	PASSENGER	RIDE_ID	DRIVER_OS	DRIVER_OS_VERSION	DRIVER_APP_VERSION
REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED
REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED
REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED
REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED
REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED
REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED
REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED
REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED
REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED
REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED
REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED	REDACTED

Computed stats about column metadata

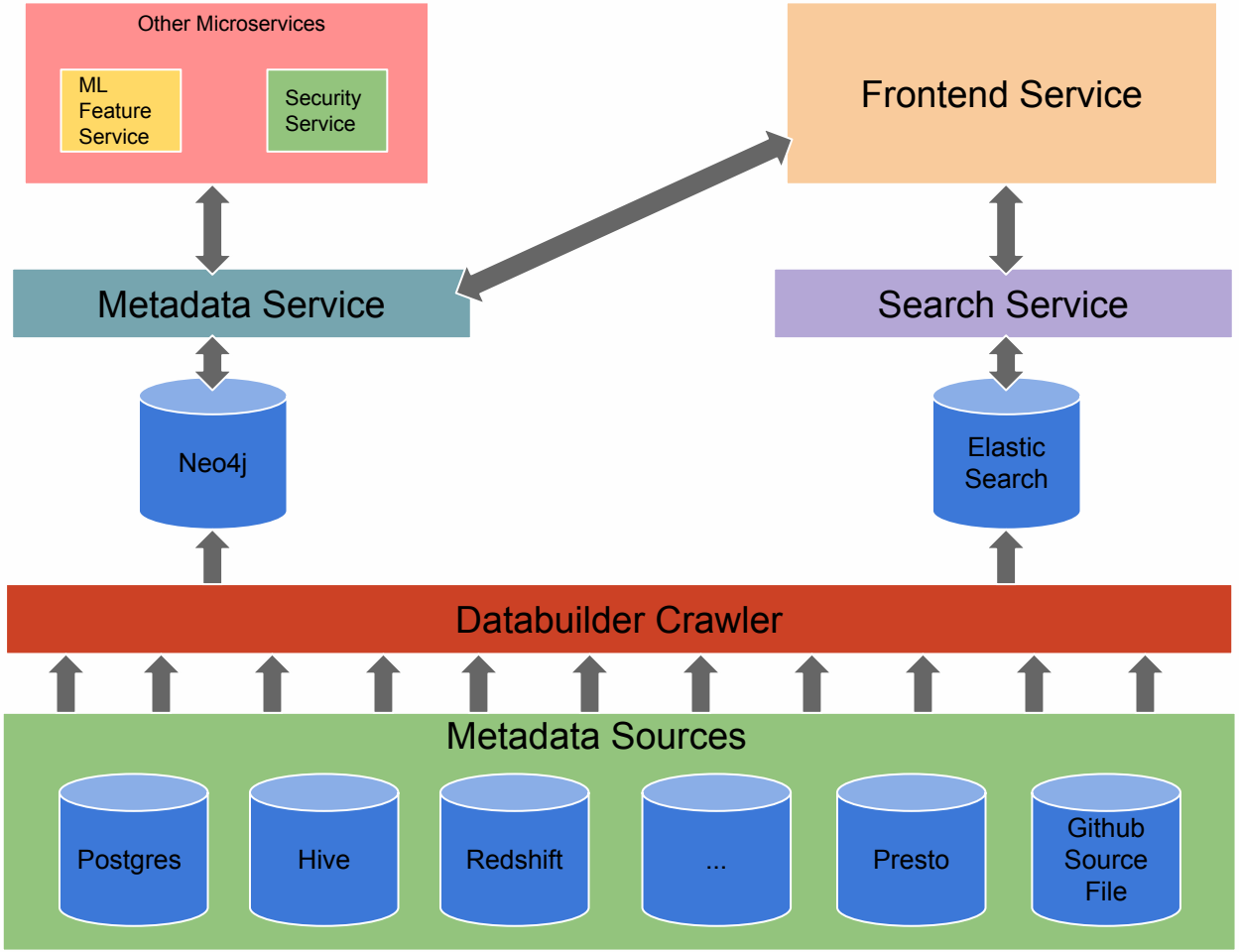


Disclaimer: these stats are arbitrary.

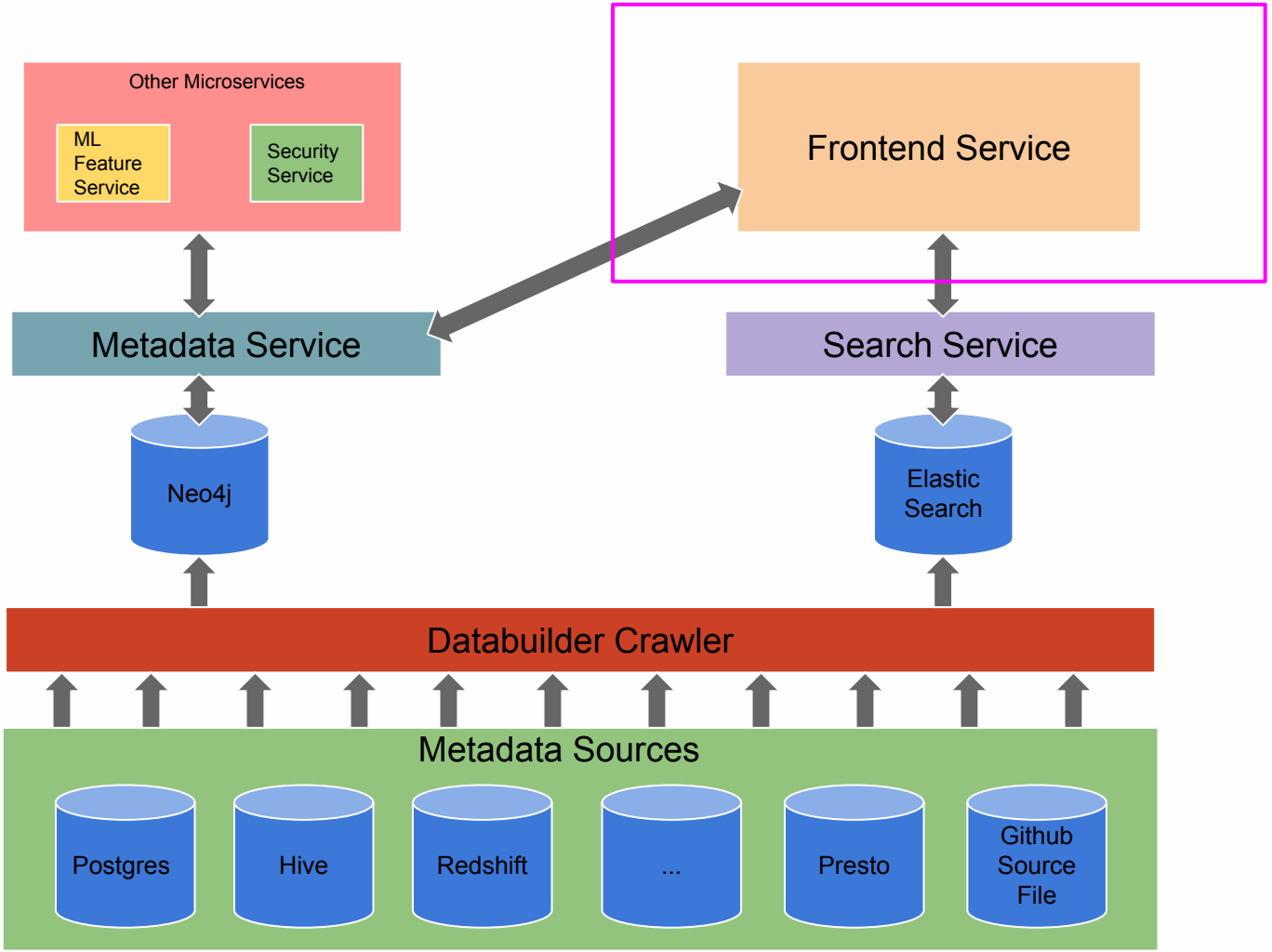
Built-in user feedback



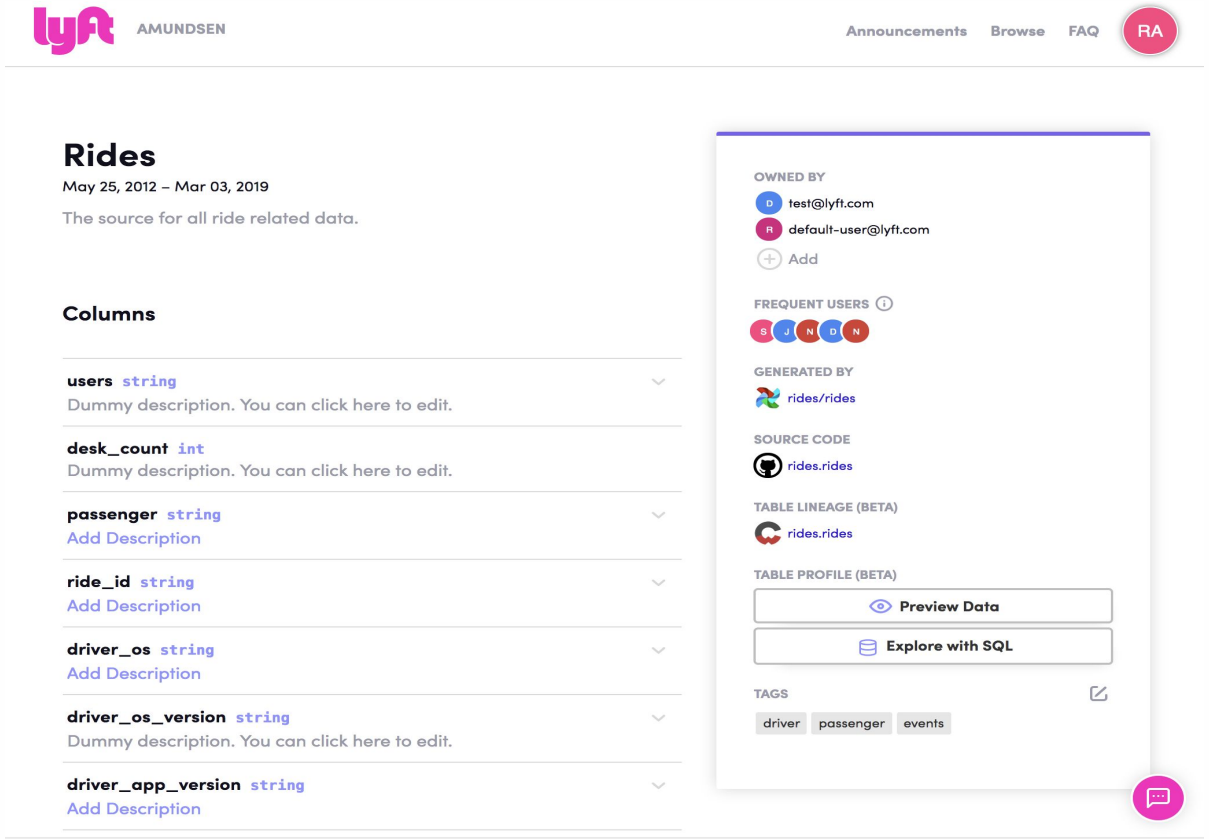
Amundsen's architecture


Other Microservices
ML Feature Service
Security Service
Frontend Service
Metadata Service
Search Service
Neo4j
Elastic Search
Databuilder Crawler
Metadata Sources
Postgres
Hive
Redshift
...
Presto
Github Source File

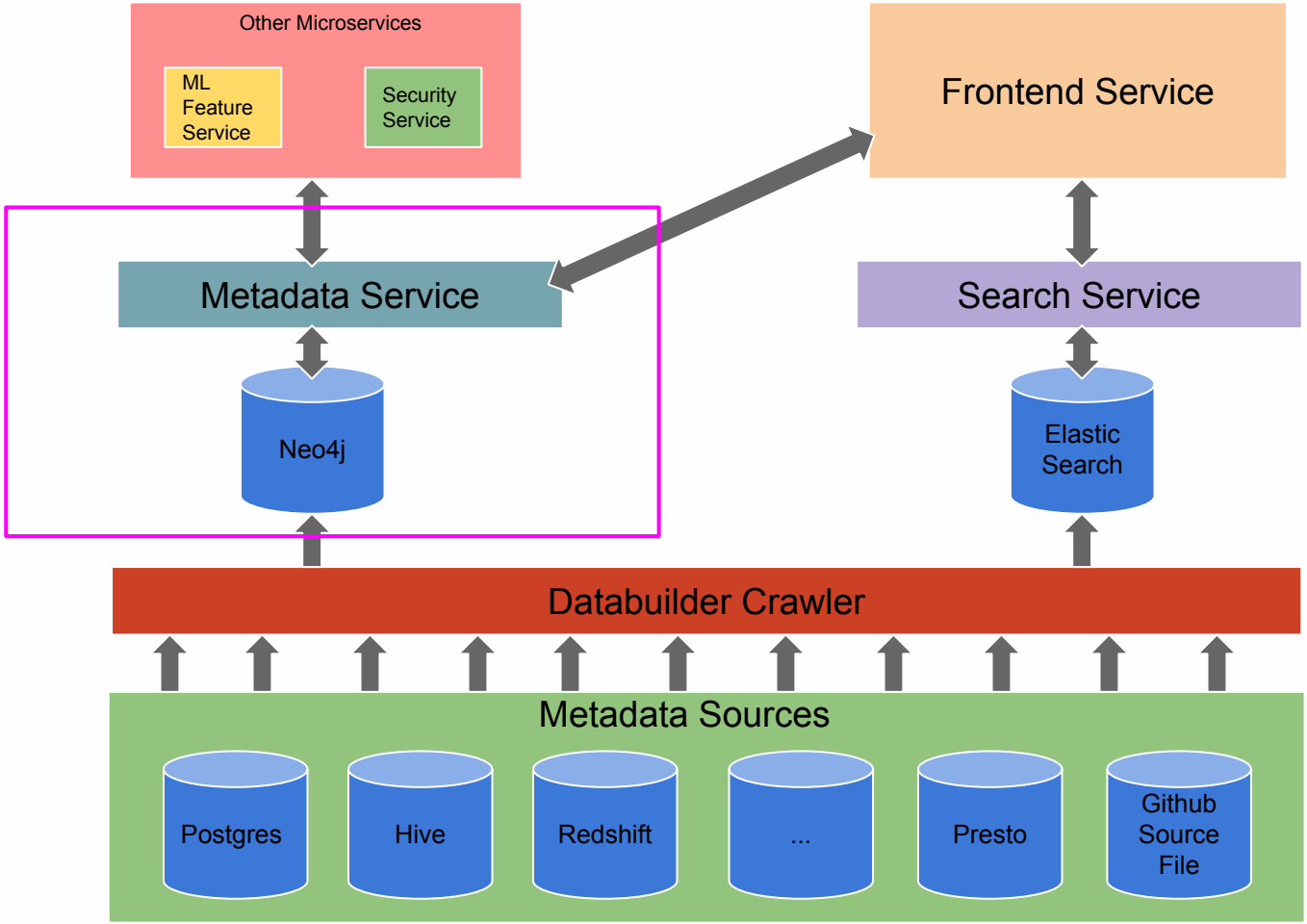
1. Frontend Service


Other Microservices
ML Feature Service
Security Service
Frontend Service
Metadata Service
Search Service
Neo4j
Elastic Search
Databuilder Crawler
Metadata Sources
Postgres
Hive
Redshift
...
Presto
Github Source File

Amundsen table detail page

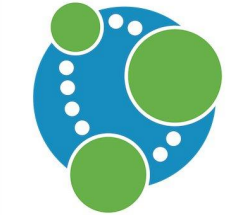


2. Metadata Service


Other Microservices
ML Feature Service
Security Service
Frontend Service
Metadata Service
Search Service
Neo4j
Elastic Search
Databuilder Crawler
Metadata Sources
Postgres
Hive
Redshift
...
Presto
Github Source File

2. Metadata Service

- A thin proxy layer to interact with graph database
  - Currently Neo4j is the default option for graph backend engine
  - Work with the community to support Apache Atlas

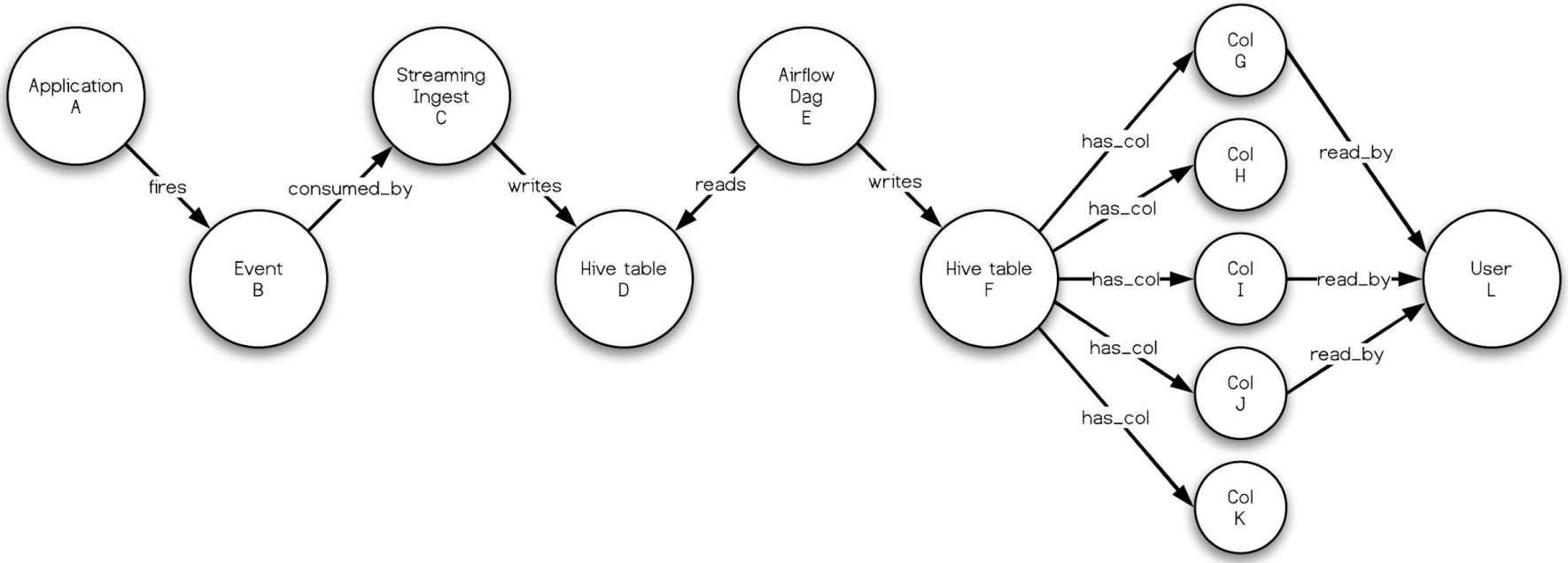


- Support Rest API for other services pushing / pulling metadata directly

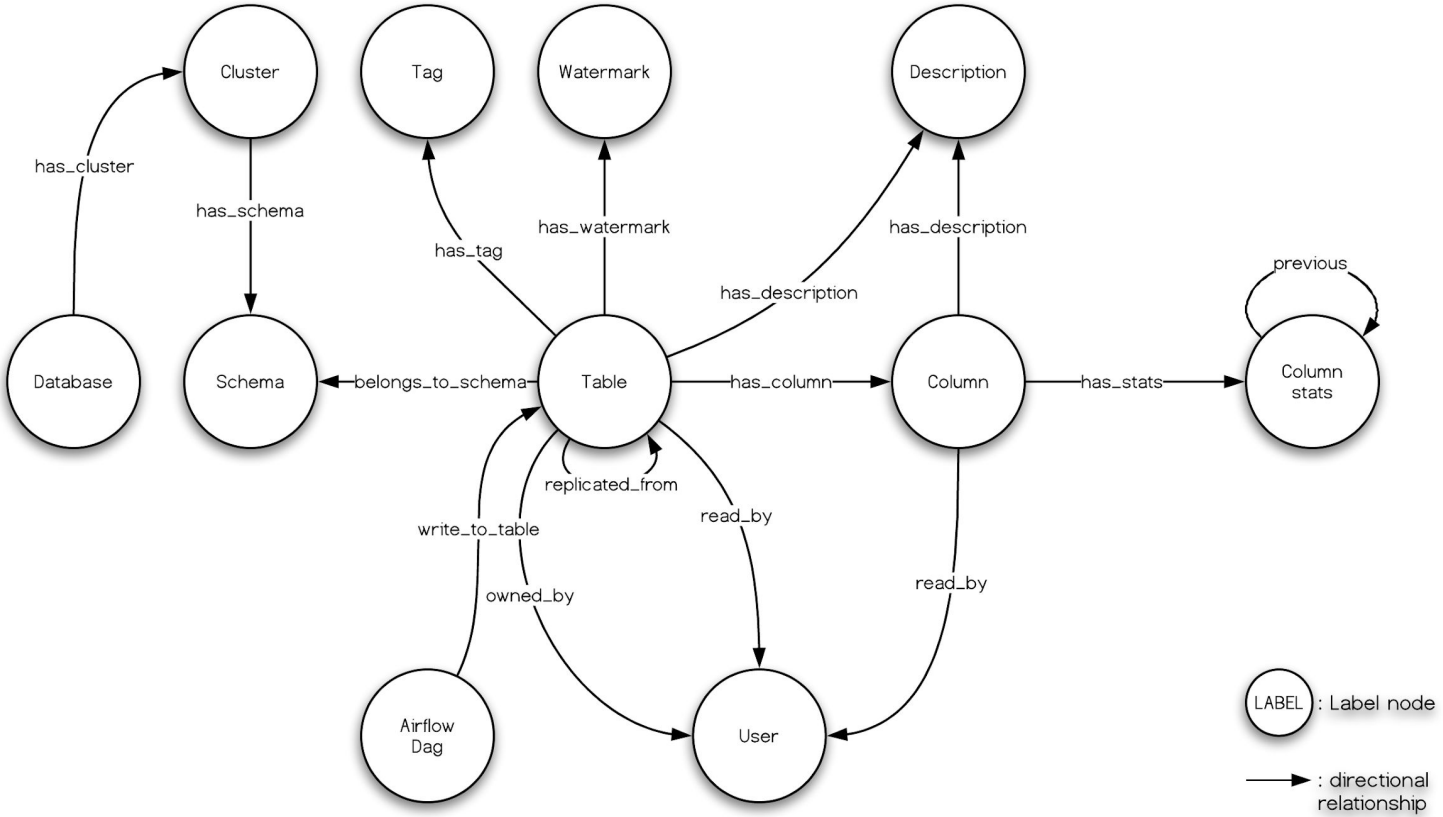
Trade Off #1

Why choose Graph database

Why Graph database?



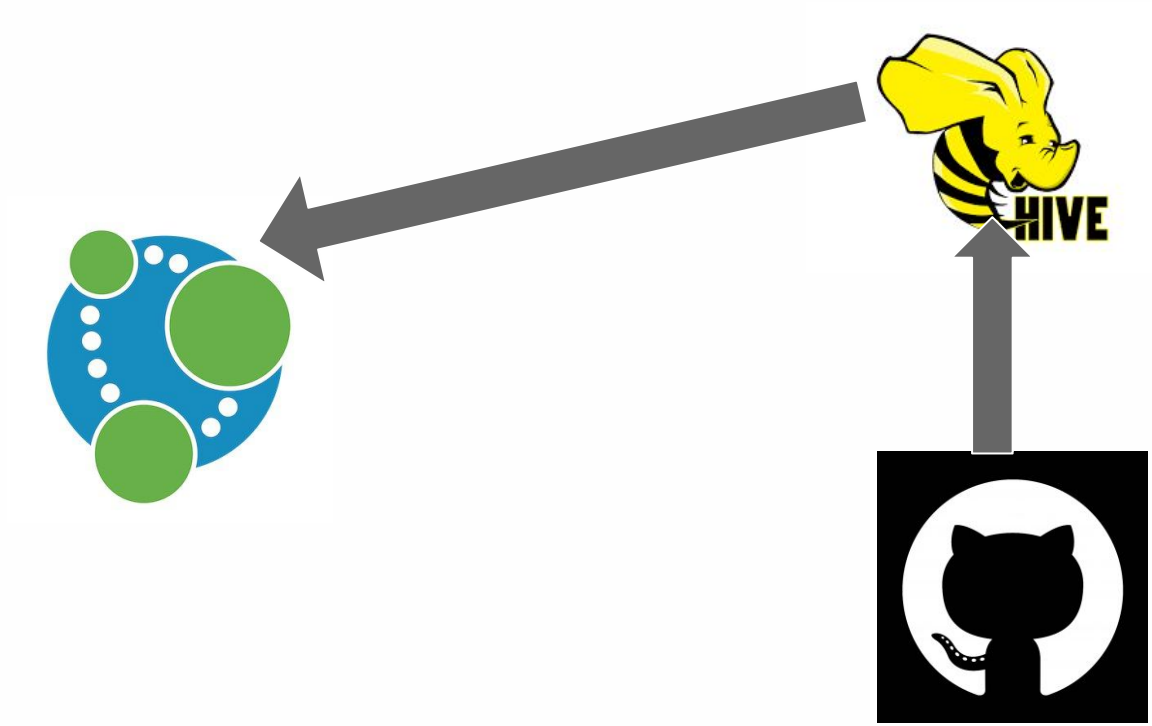
Why Graph database?



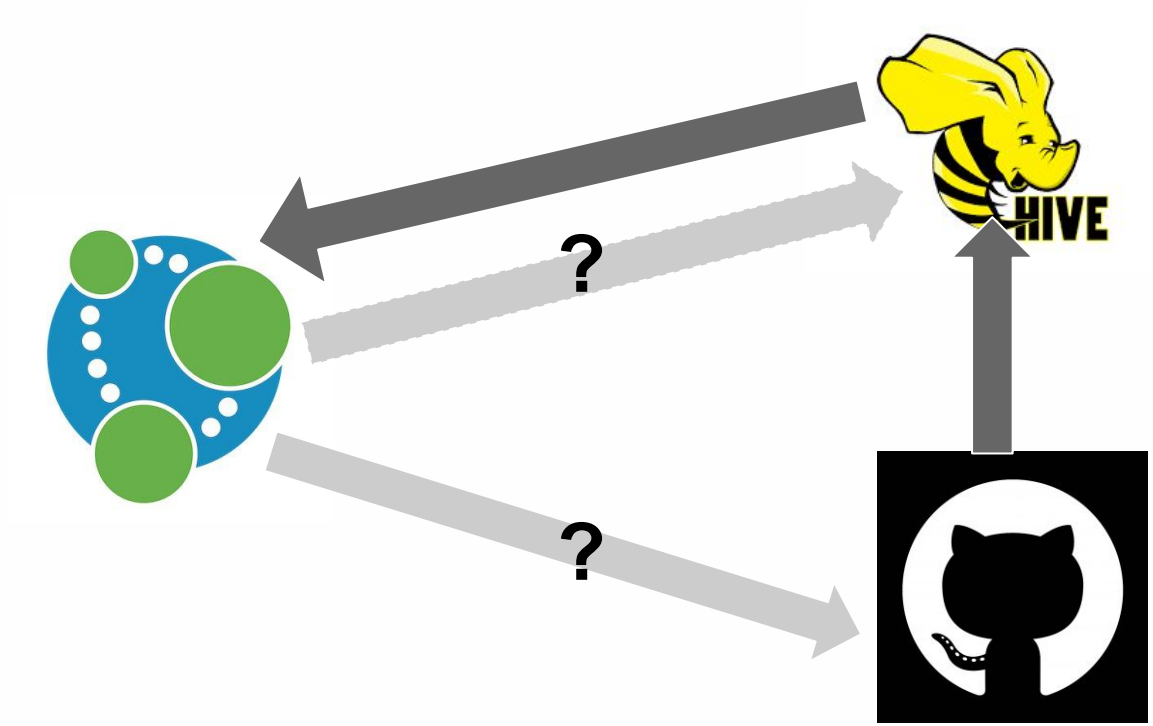
Trade Off #2

Why not propagate the metadata back to source

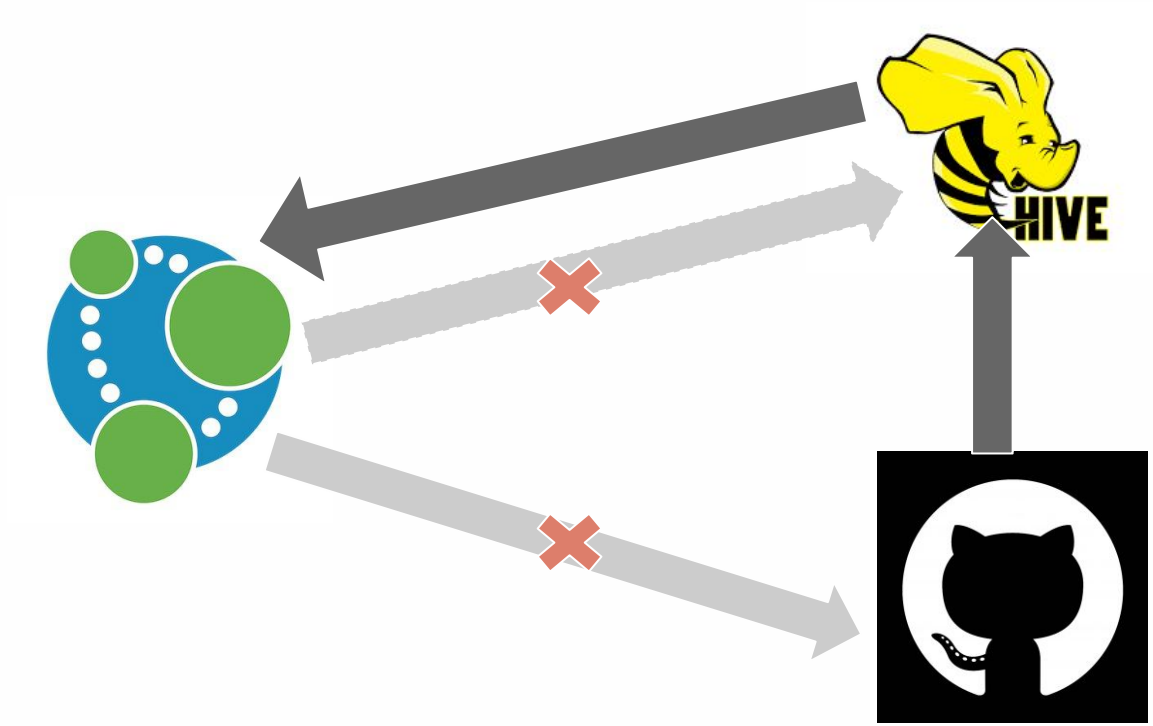
Why not propagate the metadata back to source



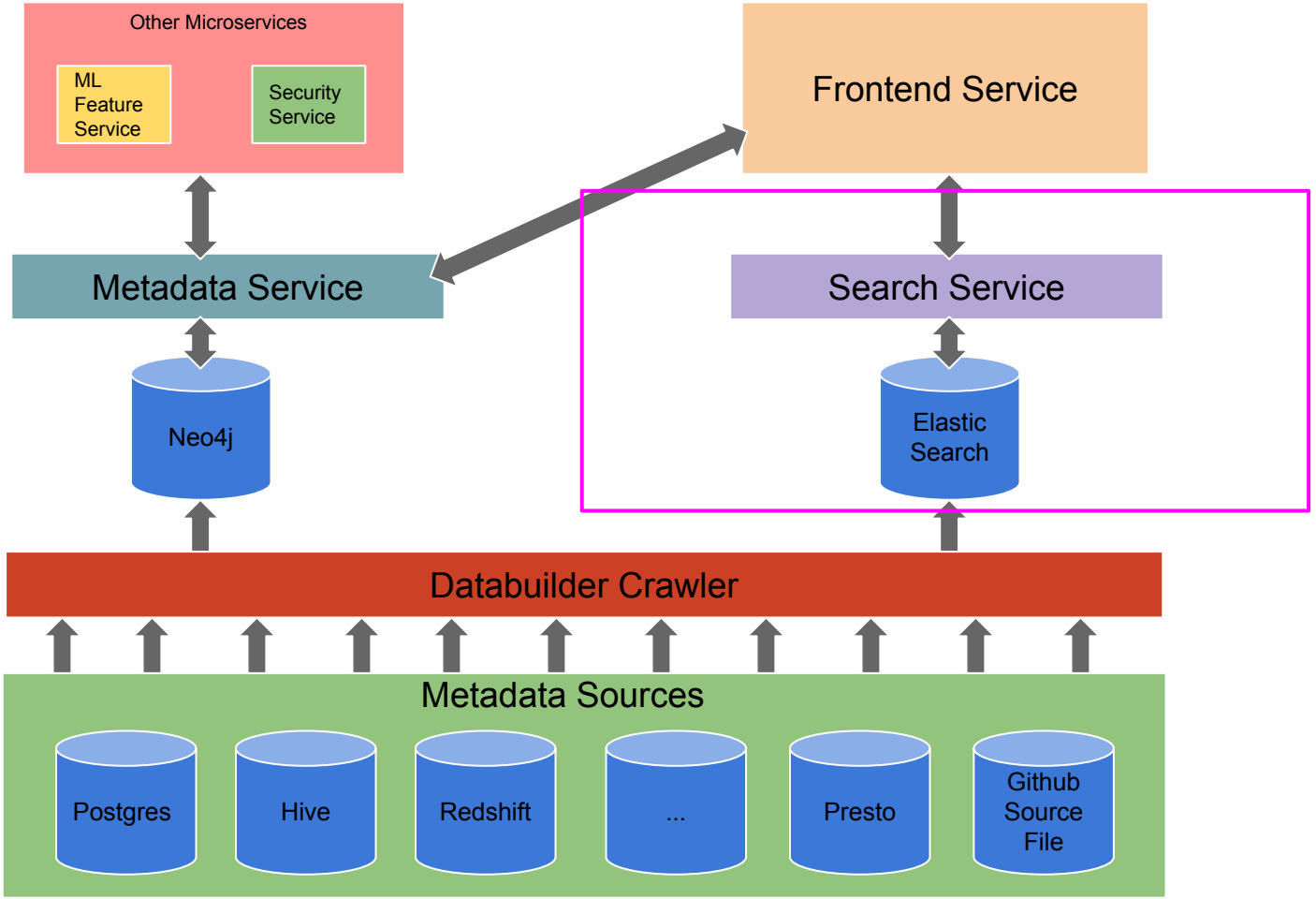
Why not propagate the metadata back to source



Why not propagate the metadata back to source



3. Search Service


Other Microservices
ML Feature Service
Security Service
Frontend Service
Metadata Service
Search Service
Neo4j
Elastic Search
Databuilder Crawler
Metadata Sources
Postgres
Hive
Redshift
...
Presto
Github Source File

3. Search Service

- A thin proxy layer to interact with the search backend
  - Currently it supports Elasticsearch as the search backend.

- Support different search patterns
  - **Normal** Search: match records based on relevancy
  - **Category** Search: match records first based on data type, then relevancy
  - **Wildcard** Search

Challenge #1

How to make the search result more relevant?

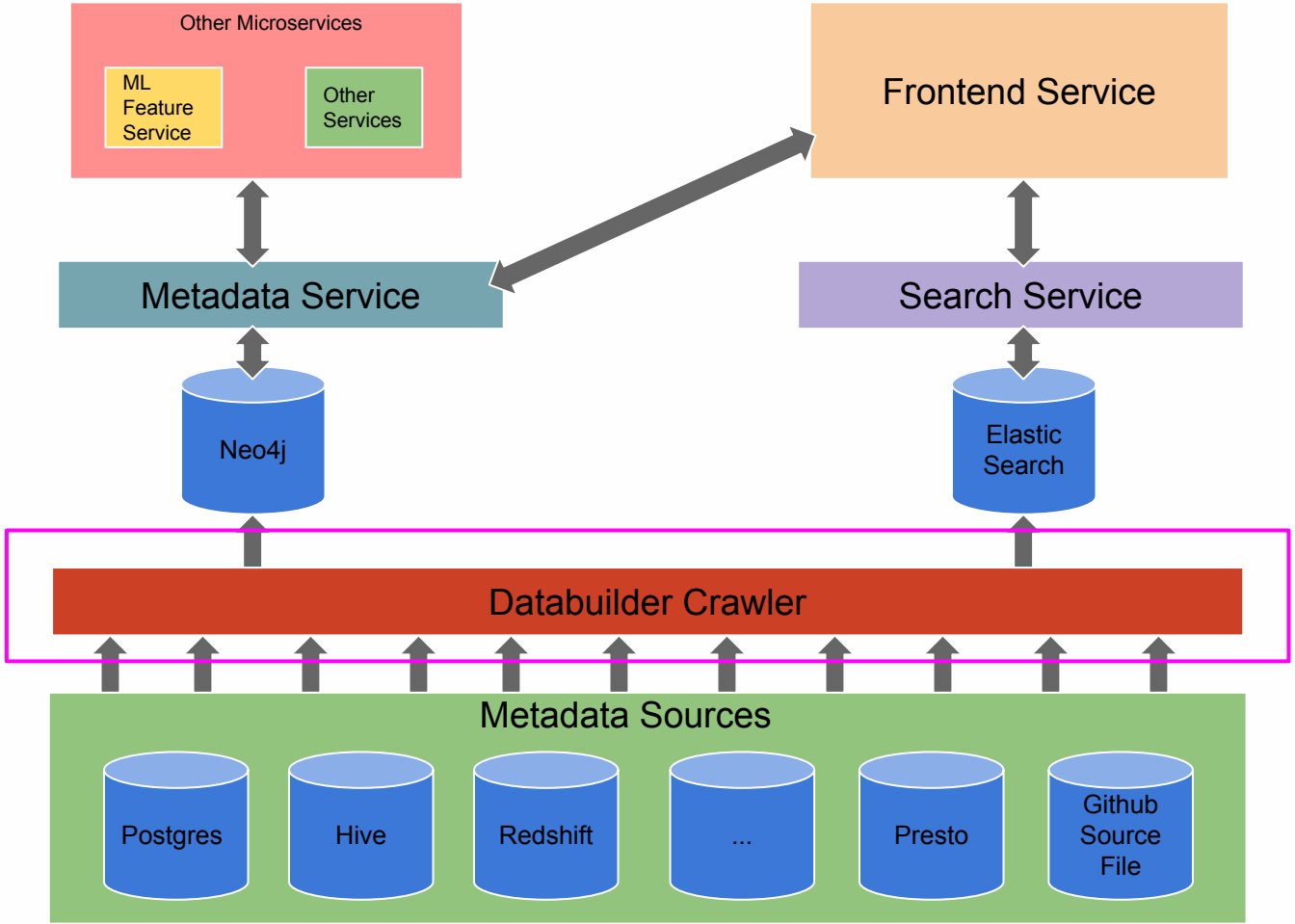
How to make the search result more relevant?

- Define a search quality metric
  - Click-Through-Rate (CTR) over top 5 results

- Search behaviour instrumentation is key

- Couple of improvements:
  - Boost the **exact table** ranking
  - Support **wildcard** search (e.g. `event_*`)
  - Support **category** search (e.g. `column: is_line_ride`)

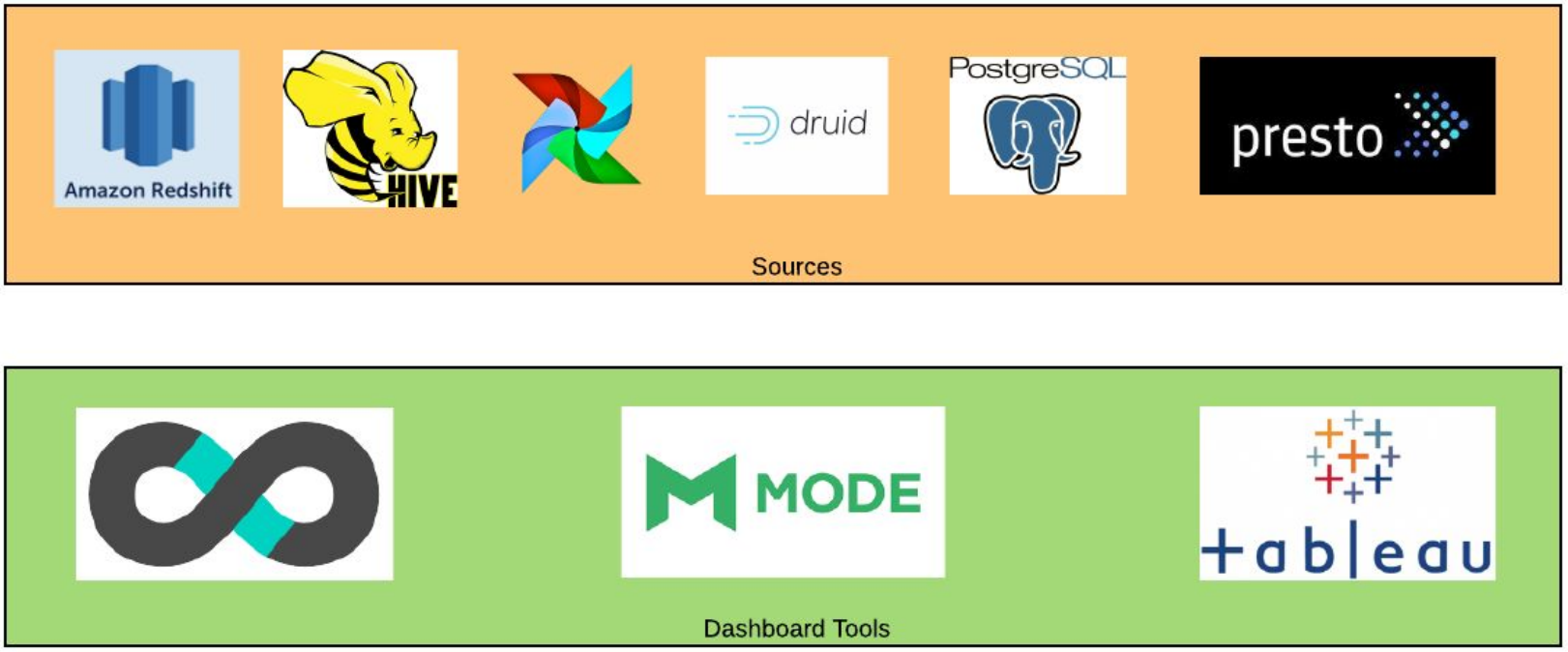
4. Data Builder


Other Microservices
ML Feature Service
Other Services
Frontend Service
Metadata Service
Search Service
Neo4j
Elastic Search
Databuilder Crawler
Metadata Sources
Postgres
Hive
Redshift
...
Presto
Github Source File

Challenge #1

Various forms of metadata

Metadata Sources @ Lyft



Metadata - Challenges

- **No Standardization**: No single data model that fits for all data resources
  - A data resource could be a table, an Airflow DAG or a dashboard

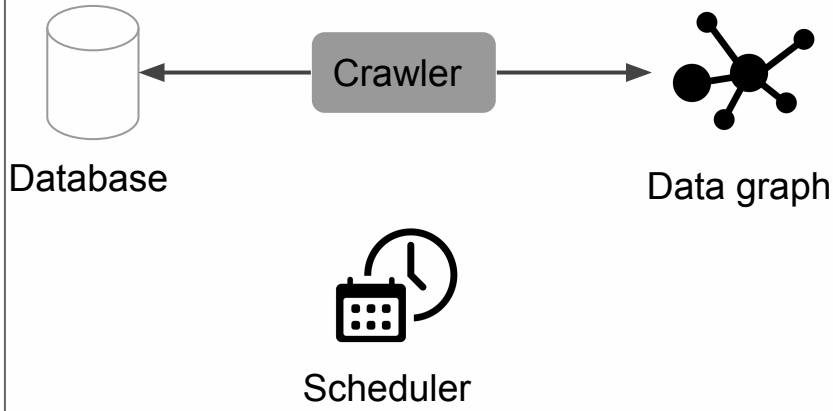
- **Different Extraction**: Each data set metadata is stored and fetched differently
  - Hive Table: Stored in Hive metastore
  - RDBMS(postgres etc): Fetched through DBAPI interface
  - Github source code: Fetched through git hook
  - Mode dashboard: Fetched through Mode API
  - ...

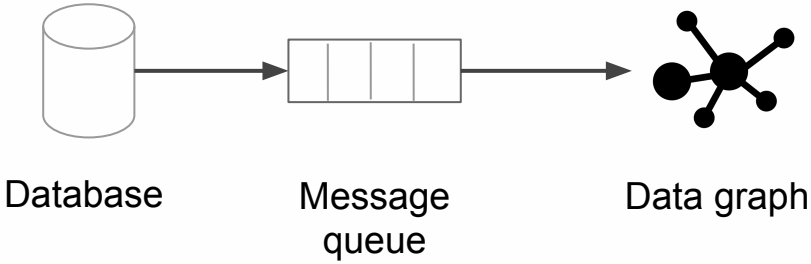
Challenge #2

Pull model vs Push model

Pull model vs. Push model

Pull Model	Push Model
• Periodically update the index by pulling from the system (e.g. database) via crawlers.	• The system (e.g. database) pushes metadata to a message bus which downstream subscribes to.

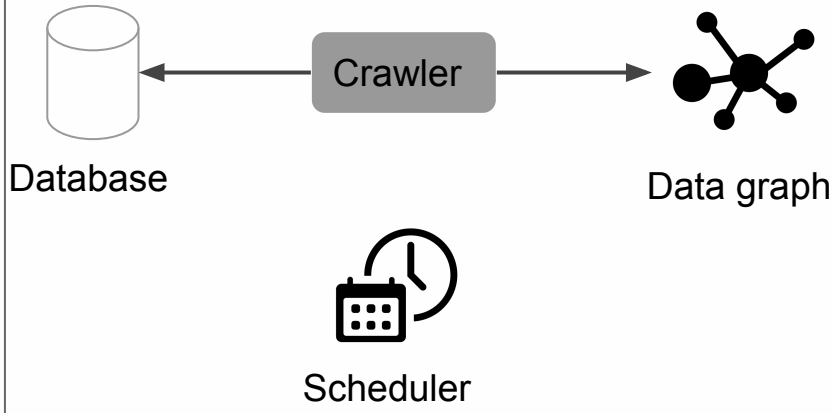




Pull model vs. push model

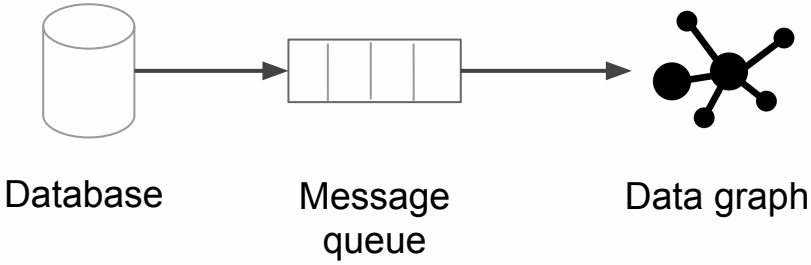
Pull Model

- Onus of integration lays on data graph
- No interface to prescribe, hard to maintain crawlers



Push Model

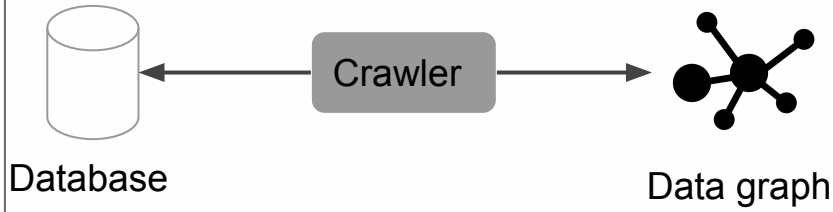
- Onus of integration lies on database
- Message format serves as the interface
- Allows for near-real time indexing



Pull model vs. push model

Pull Model

- Onus of integration lays on data graph
- No interface to prescribe, hard to maintain crawlers

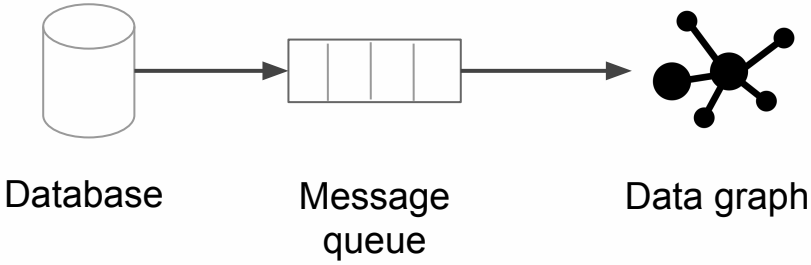


Preferred if

- Waiting for indexing is ok
- Working with “strapped” teams
- There’s already an interface

Push Model

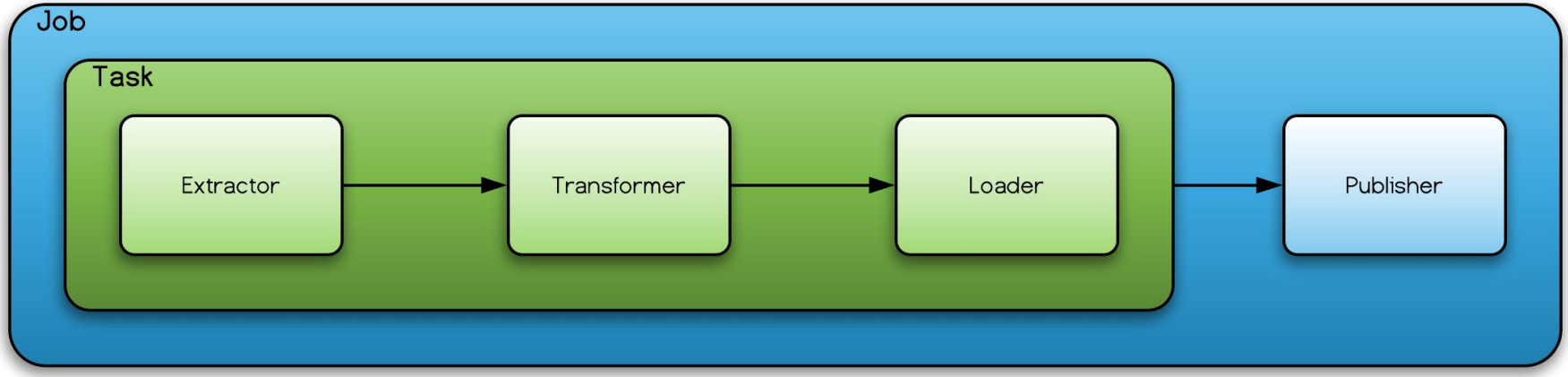
- Onus of integration lies on database
- Message format serves as the interface
- Allows for near-real time indexing



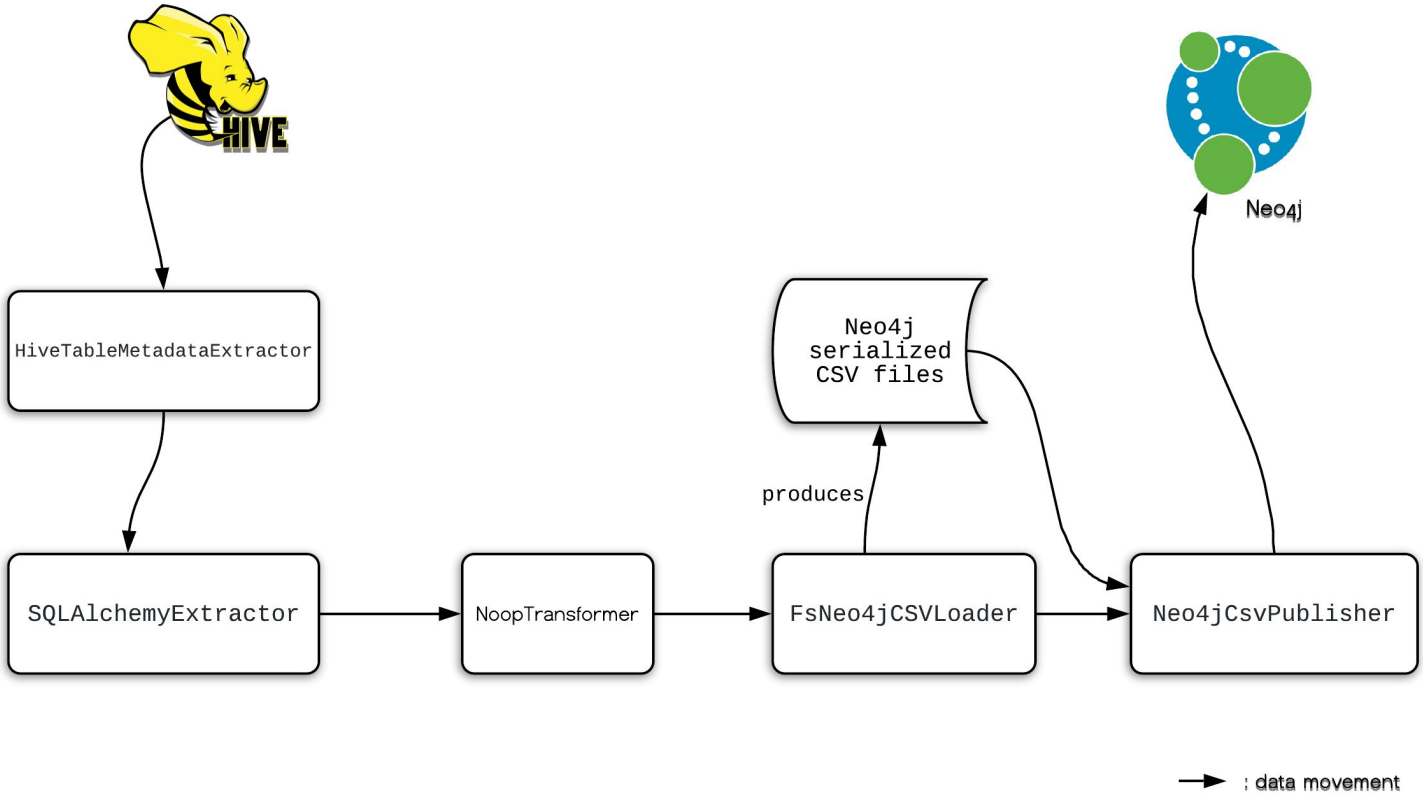
Preferred if

- Near-real time indexing is important
- Clean interface doesn’t exist
- Other tools like Wherehows are moving towards Push Model

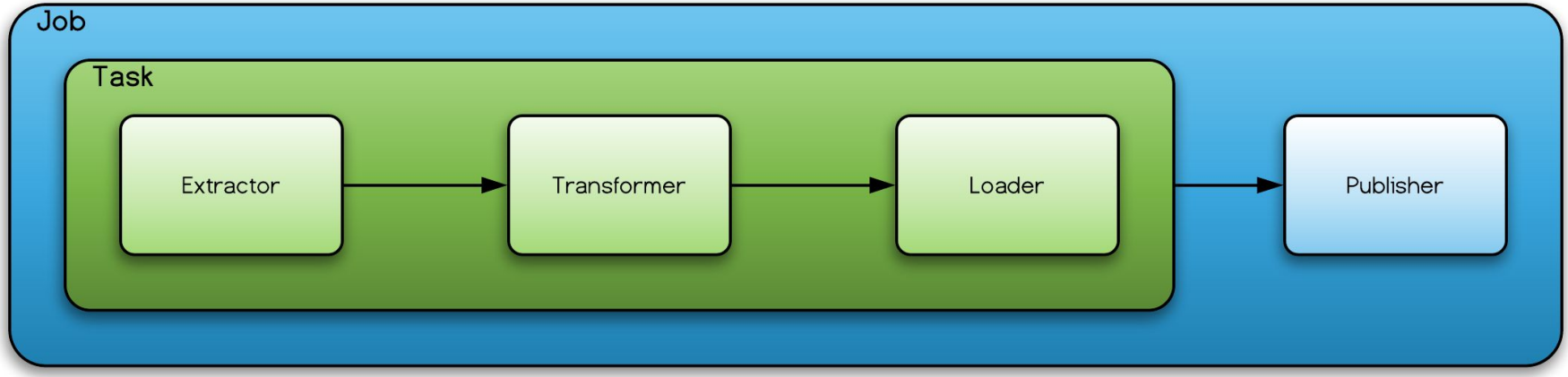
4. Databuilder



Databuilder in action



How are we building data? Databuilder



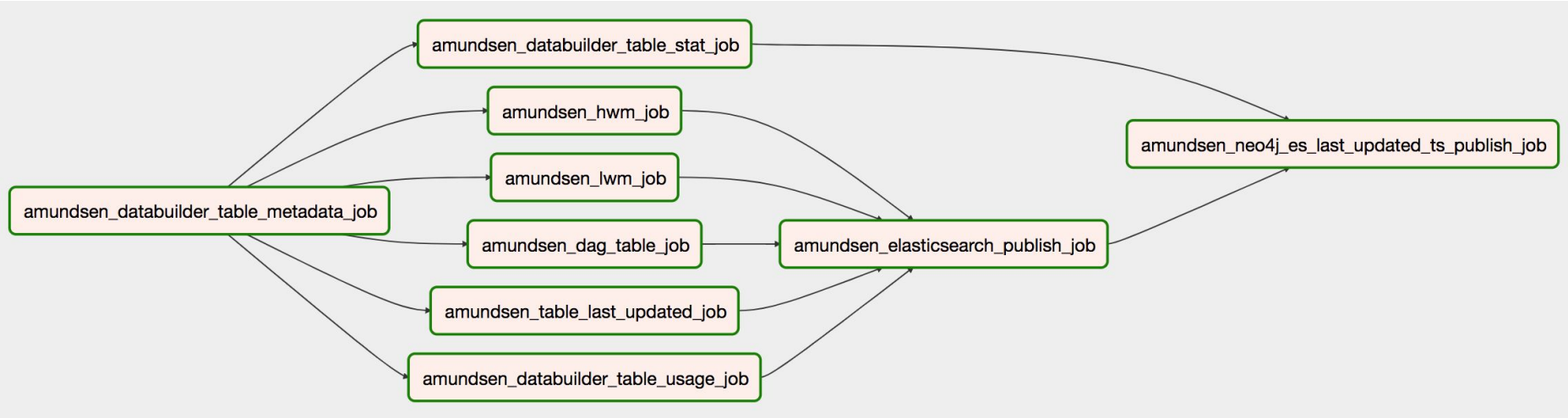

```
task = DefaultTask(extractor=SQLAlchemyExtractor(),
                   loader=FsNeo4jCSVLoader())

job = DefaultJob(conf=job_config,
                 task=task,
                 publisher=Neo4jCsvPublisher())

# run job
job.launch()
```

How is databuilder orchestrated?

Amundsen uses Apache Airflow to orchestrate Databuilder jobs

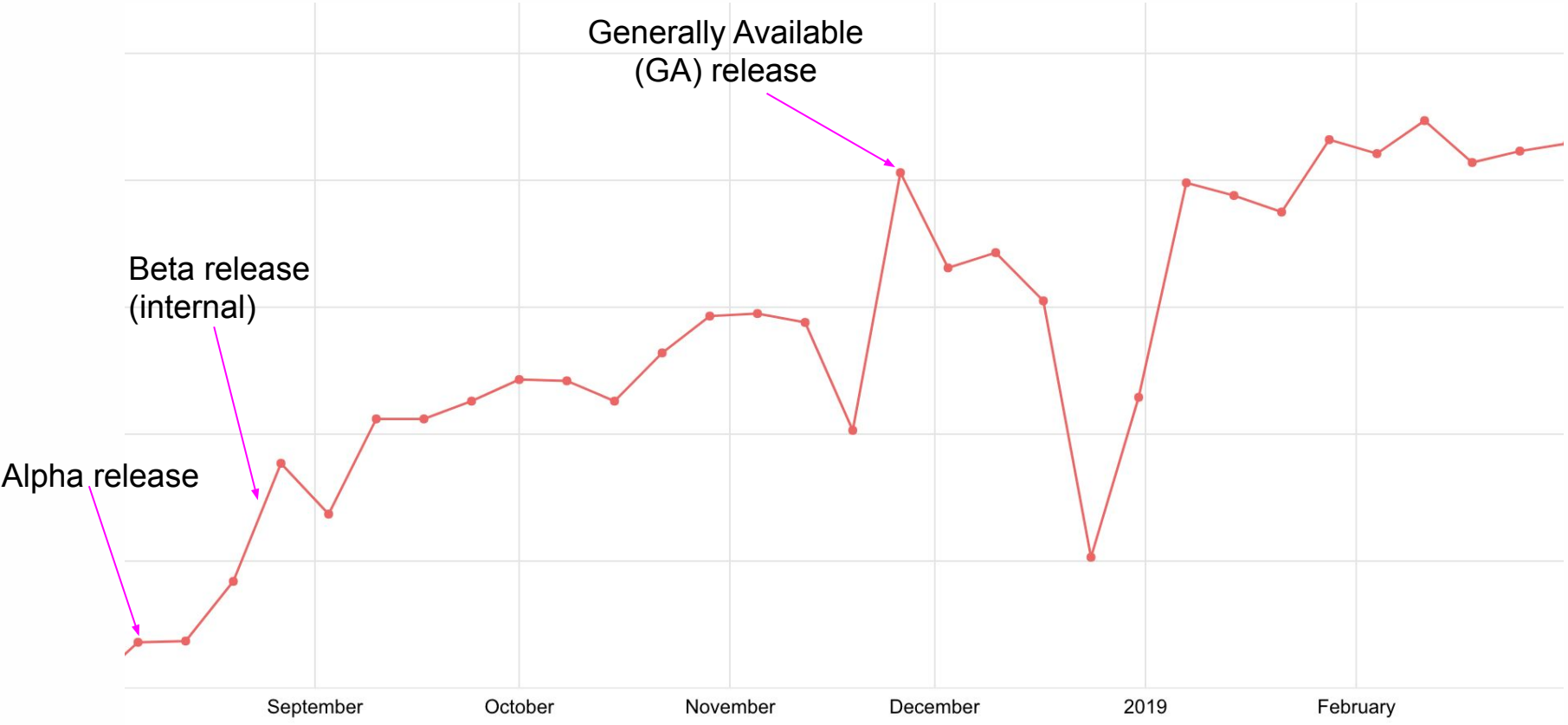


What's next?

Amundsen seems to be more useful than what we thought

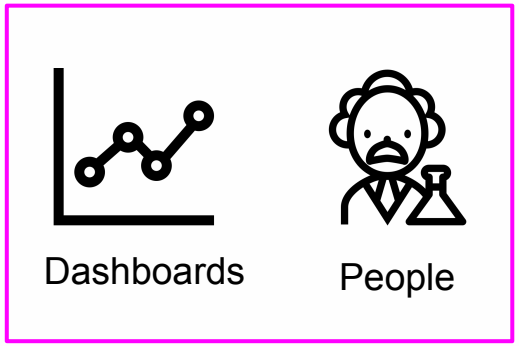
- Tremendous success at Lyft
  - Used by Data Scientists, Engineers, PMs, Ops, even Cust. Service!
- Many organizations have similar problems
  - Collaborating with ING, WeWork and more
  - We plan to announce open source soon

Impact - Amundsen at Lyft

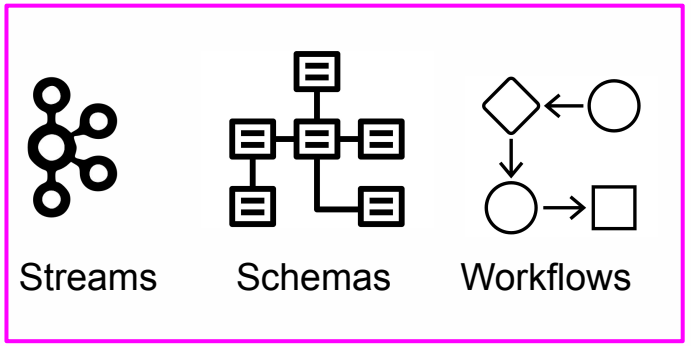


Adding more kinds of data resources

Phase 1
(Complete)



Phase 2
(In development)



Phase 3
(In Scoping)

Summary

Summary

- Data Discovery adds 30+% more productivity to Data Scientists
- Metadata is key to the next wave of big data applications
- Amundsen - Lyft's metadata and data discovery platform
- Blog post with more details: go.lyft.com/datadiscoveryblog

Data @ Lyft

Mark Grover | @mark_grover

Tao Feng | @feng-tao

Slides at go.lyft.com/datadiscoveryslides

Blog post at go.lyft.com/datadiscoveryblog

We're Hiring! Apply at www.lyft.com/careers or email data-recruiting@lyft.com

Data Engineering

Engineering Manager
San Francisco

Software Engineer
San Francisco, Seattle, & New York City

Data Infrastructure

Engineering Manager
San Francisco

Software Engineer
San Francisco & Seattle

Experimentation

Software Engineer
San Francisco

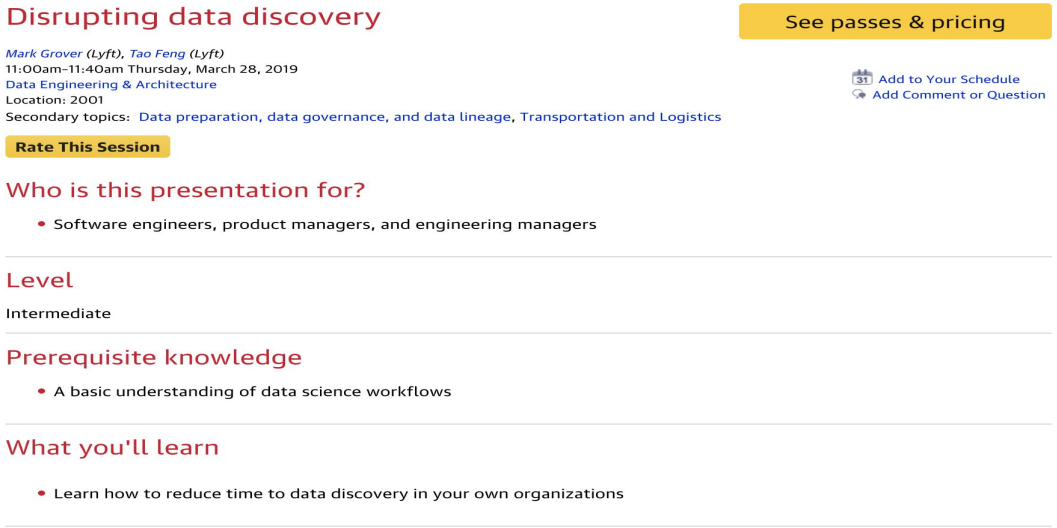
Observability

Software Engineer
San Francisco

Streaming

Software Engineer
San Francisco

Rate this session



Disrupting data discovery

See passes & pricing

Mark Grover (Lyft), Tao Feng (Lyft)
11:00am–11:40am Thursday, March 28, 2019
Data Engineering & Architecture
Location: 2001
Secondary topics: Data preparation, data governance, and data lineage, Transportation and Logistics

Add to Your Schedule
Add Comment or Question

Rate This Session

Who is this presentation for?

- Software engineers, product managers, and engineering managers

Level

Intermediate

Prerequisite knowledge

- A basic understanding of data science workflows

What you'll learn

- Learn how to reduce time to data discovery in your own organizations

session page on conference website

Strata SF 2019

O'Reilly Events App

Backup