

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

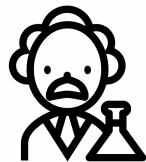
Data platform users



Data Modelers



Analysts



Data Scientists



Product
Managers



General
Managers



Engineers



Experimenters



Data Platform

Lyft Data Team

Lyft Data Team

Core Data Infra

Streaming Infra

Visualization

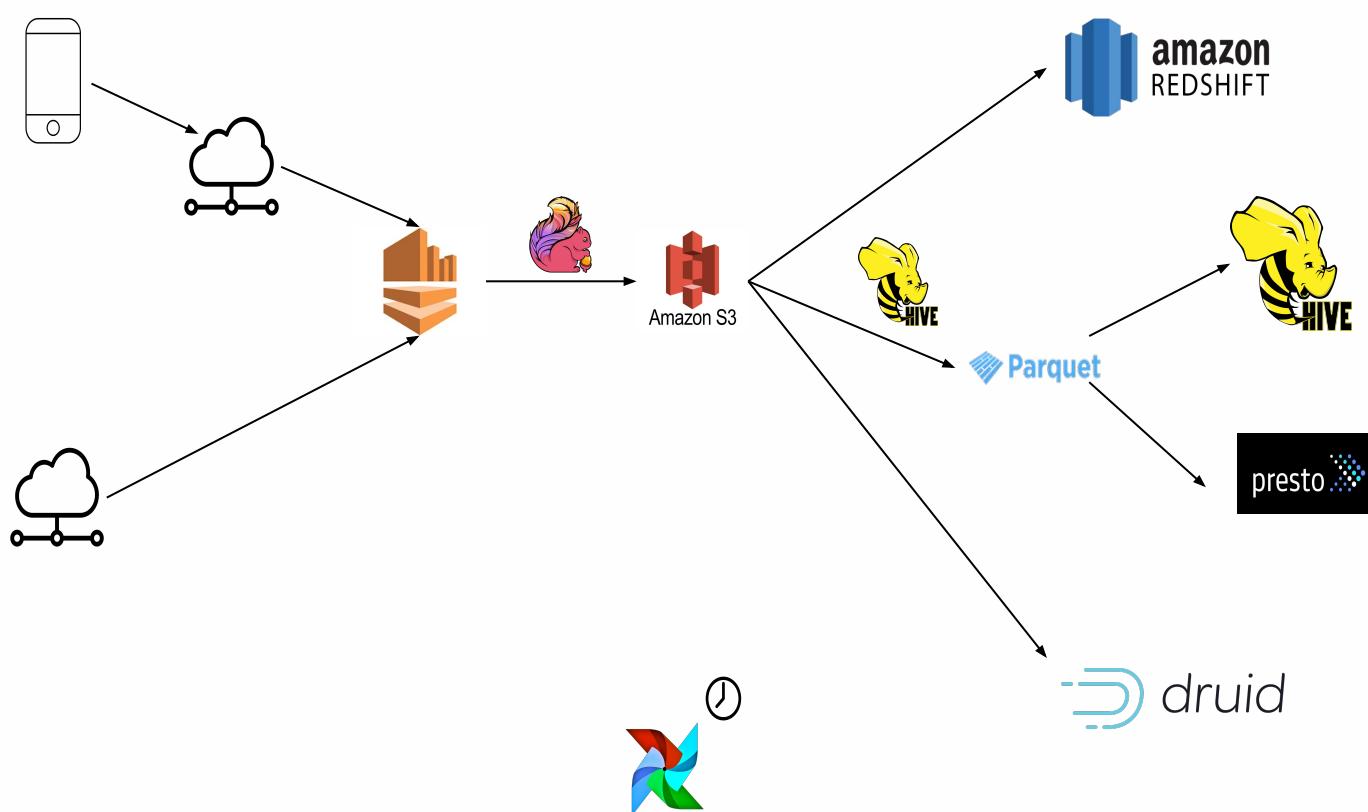
Experimentation

BI and Logging

ML Infra



Core Infra high level architecture



looker

MODE

infinity

+ a b | e a u

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

Code	244
Commits	143
Issues	287
Wikis	

Languages	
SQL	117
Python	41
PLSQL	34
Markdown	2
PLpgsql	2
CSV	1
Jupyter Notebook	1
SQLPL	1
SaltStack	1
Shell	1

[Advanced search](#) [Cheat sheet](#)

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;
```

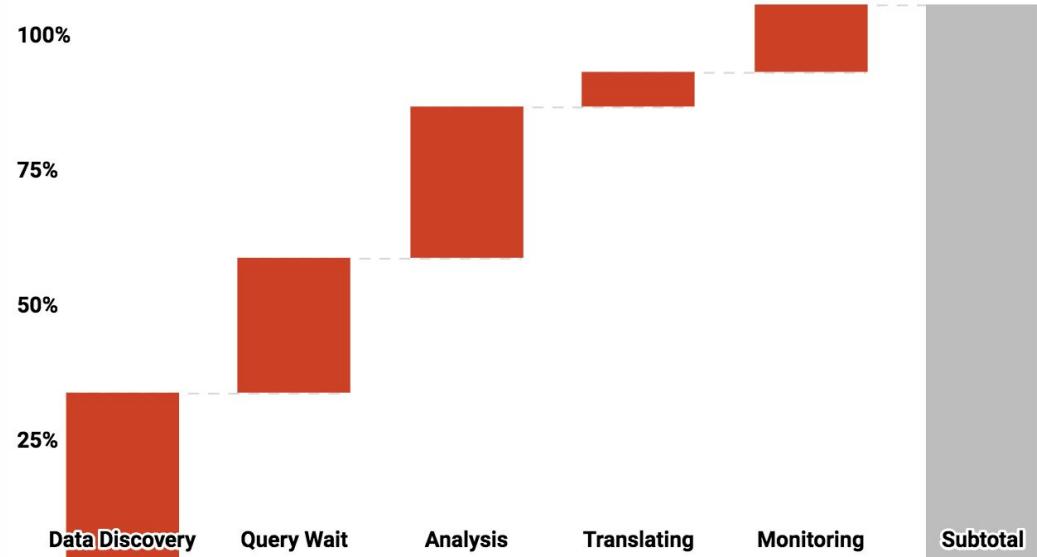
Exploring with **SELECT *** is EVIL

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.

Data Scientists Time Spent



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

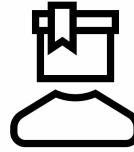
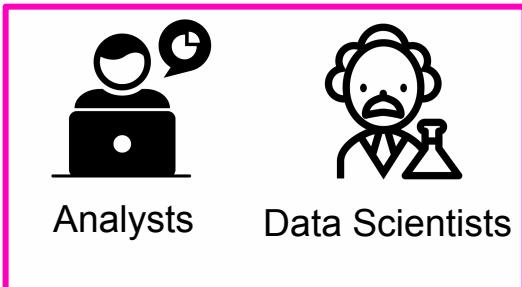
How data is changing over time

Audience for data discovery

Data Discovery - User personas



Data Modelers



Product
Managers



General
Managers



Engineers



Experimenters



Data Platform

3 Data Scientist personas

		
<p>Power user</p> <ul style="list-style-type: none">• All info in their head• Get interrupted a lot due to questions	<p>Noob user</p> <ul style="list-style-type: none">• Lost• Ask “power users” a lot of questions	<p>Manager</p> <ul style="list-style-type: none">• Dependencies• landing on time• Communicating with stakeholders

Data Discovery answers 3 kinds of questions

Search based	Lineage based	Network based
<p>Where is the table/dashboard for X?</p> <p>What does it contain?</p>	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

The screenshot shows a landing page for a data catalog named "AMUNDSEN". At the top left is the Lyft logo. To the right are links for "Announcements", "Browse", "FAQ", and a user profile icon labeled "RA". Below the header is a search bar with the placeholder "Search for data resources...". A note below the search bar explains the search pattern: "Search within a category using the pattern with wildcard support 'category:*searchTerm*', e.g. 'schema:*core*'. Current categories are 'column', 'schema', 'table', and 'tag'." A section titled "Popular Tables" contains four entries, each with a table icon and a title followed by a description and a "View" link:

- rides**
This is the main table for rides. This is a dummy description. >
- passengers**
This is the main table for passengers. This is a dummy description. >
- drivers**
This is the main table for drivers. This is a dummy description. >
- bikes**
This is the main table for bikes. This is a dummy description. >

At the bottom of the page, a footer note states: "Amundsen was last indexed on March 1st 2019 at 5:15:25 am".

Search results ranked on relevance and query activity

 **passenger**

Search within a category using the pattern with wildcard support 'category:*searchTerm*', e.g. 'schema:*core*'. Current categories are 'column', 'schema', 'table', and 'tag'.

1-2 of 2 results 

 **passenger**

This is the main table for passenger . This is a dummy description.



 **passenger_ride_cancellations**

Passenger ride cancels. This is a dummy description.



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
<ul style="list-style-type: none">Names, Descriptions, Tags, [owners, frequent users]	<ul style="list-style-type: none">Querying activityDashboardingDifferent weights for automated vs adhoc querying

Back to mocks...

Search results ranked on relevance and query activity

 **passenger**

Search within a category using the pattern with wildcard support 'category:*searchTerm*', e.g. 'schema:*core*'. Current categories are 'column', 'schema', 'table', and 'tag'.

1-2 of 2 results 

 **passenger**

This is the main table for passenger . This is a dummy description.



 **passenger_ride_cancellations**

Passenger ride cancels. This is a dummy description.



Detailed description and metadata about data resources



AMUNDSEN

Announcements Browse FAQ

RA

Rides

May 25, 2012 – Mar 03, 2019

The source for all ride related data.

Columns

`users string`

Dummy description. You can click here to edit.

`desk_count int`

Dummy description. You can click here to edit.

`passenger string`

Add Description

`ride_id string`

Add Description

`driver_os string`

Add Description

`driver_os_version string`

Dummy description. You can click here to edit.

`driver_app_version string`

Add Description

OWNED BY

- test@lyft.com
- default-user@lyft.com

+ Add

FREQUENT USERS



GENERATED BY



SOURCE CODE



TABLE LINEAGE (BETA)



TABLE PROFILE (BETA)

Preview Data
Explore with SQL

TAGS

driver passenger events

Data Preview within the tool

Computed stats about column metadata

desk_count int

Dummy description. You can click here to edit.

How is this data generated?

These stats are based on data collected for this column on 3/1/2019

passenger string

Add Description



COUNT

123456

COUNT_NULL

321

COUNT_DISTINCT

123456

LEN_MAX

24

LEN_MIN

24

LEN_AVG

24

LEN_SUM

1234567

TABLE I



TABLE I



Disclaimer: these stats are arbitrary.

Built-in user feedback

PRODUCT FEEDBACK

Rating Bug Report Request

How likely are you to recommend this tool to a friend or co-worker?

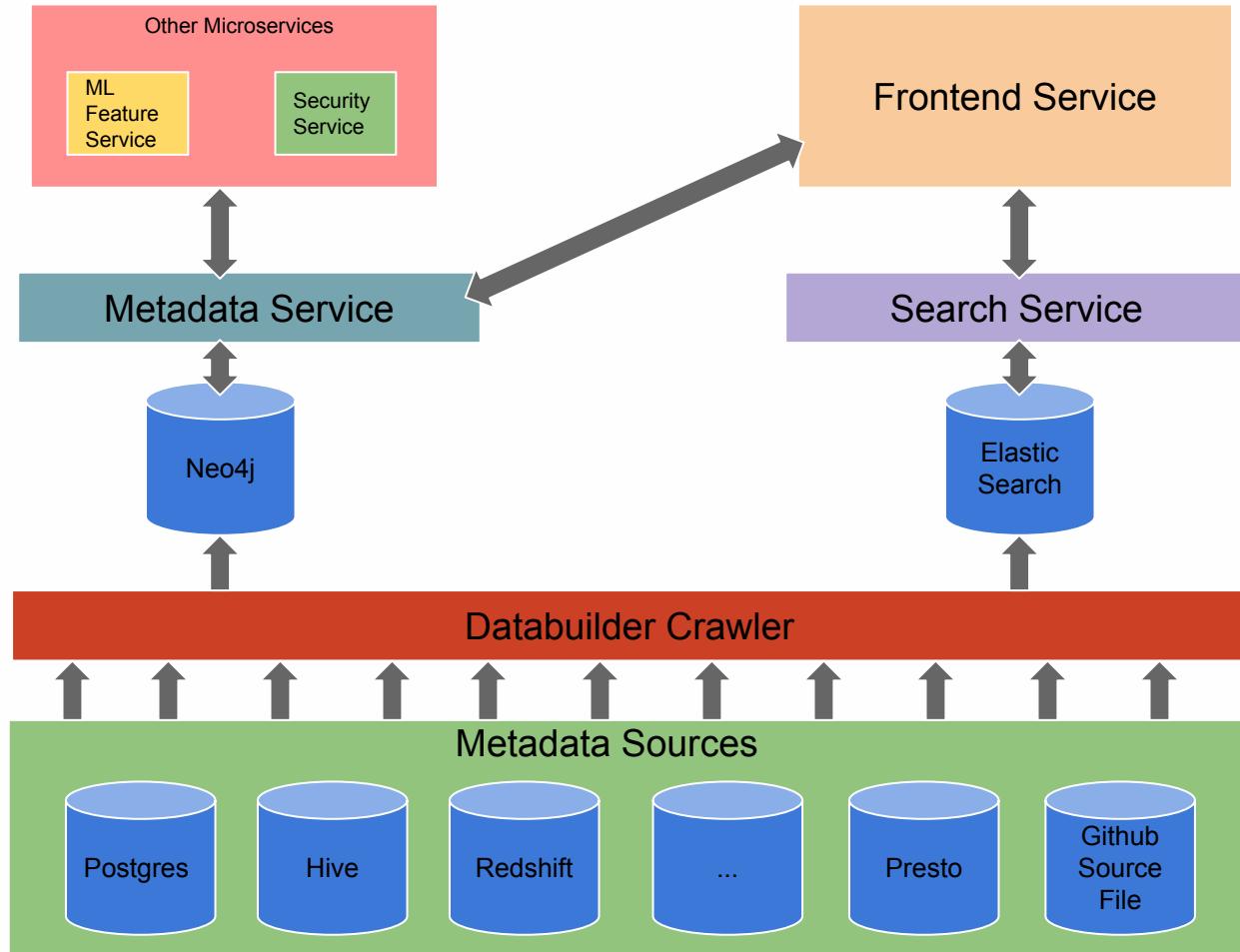
1 2 3 4 5 6 7 8 9 10

Not Very Likely Very Likely

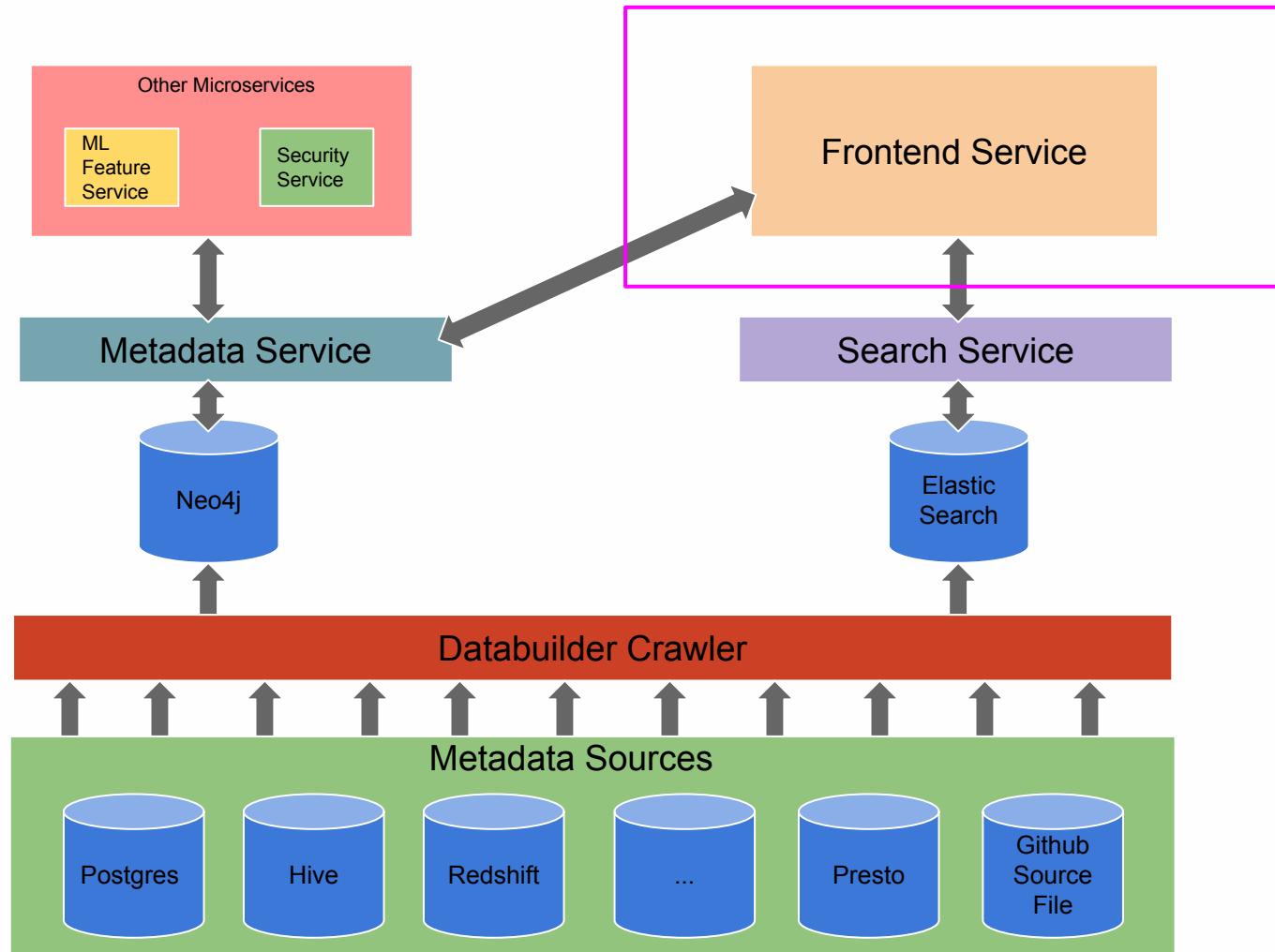
Additional Comments

Submit

Amundsen's architecture



1. Frontend Service



Amundsen table detail page



Announcements Browse FAQ



Rides

May 25, 2012 – Mar 03, 2019

The source for all ride related data.

Columns

users string

Dummy description. You can click here to edit.

desk_count int

Dummy description. You can click here to edit.

passenger string

Add Description

ride_id string

Add Description

driver_os string

Add Description

driver_os_version string

Dummy description. You can click here to edit.

driver_app_version string

Add Description

OWNED BY

- test@lyft.com
- default-user@lyft.com

+ Add

FREQUENT USERS



GENERATED BY



SOURCE CODE



TABLE LINEAGE (BETA)



TABLE PROFILE (BETA)

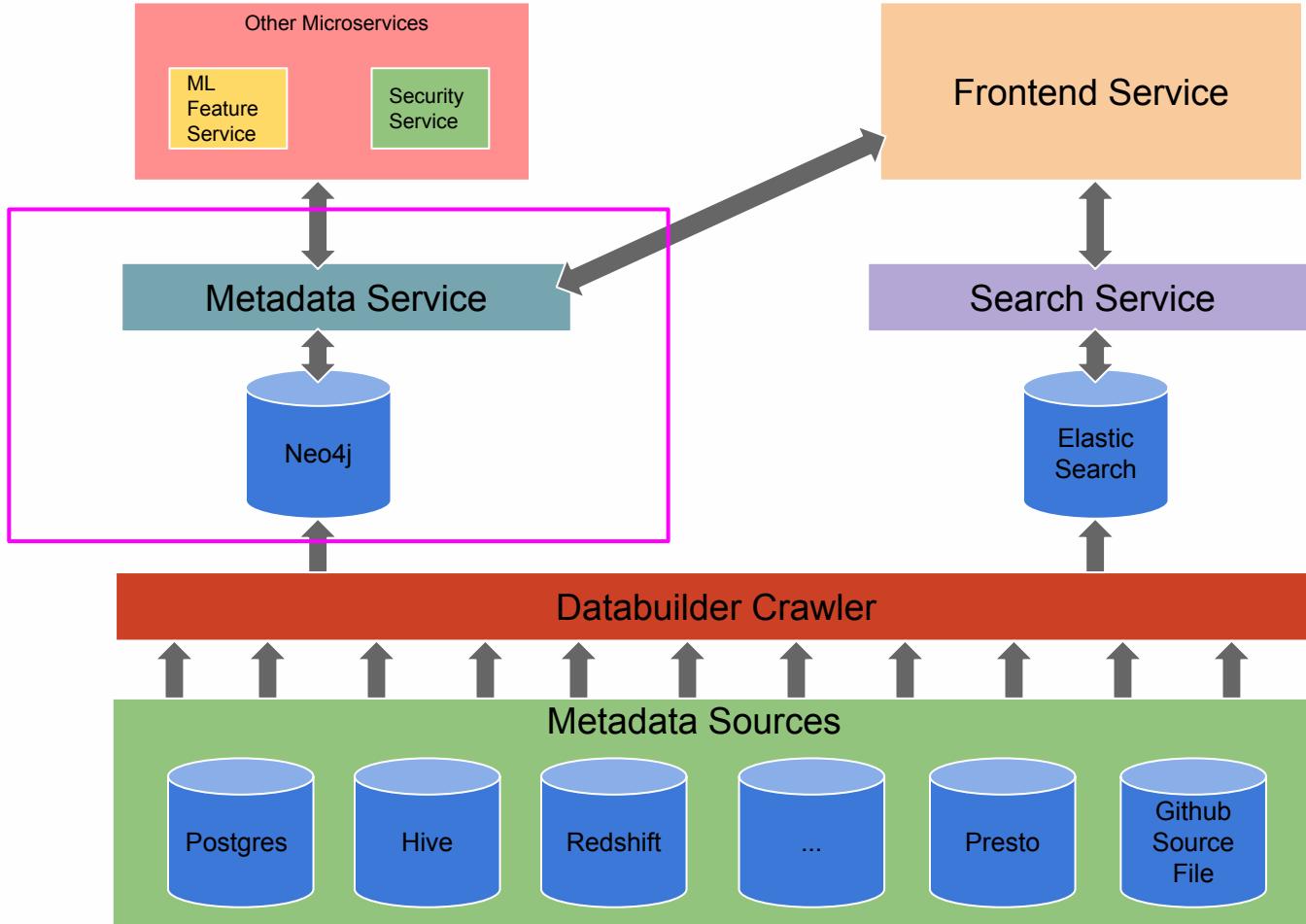
Preview Data
Explore with SQL

TAGS

driver passenger events

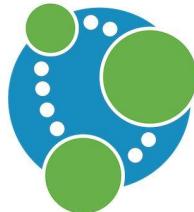


2. Metadata Service



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



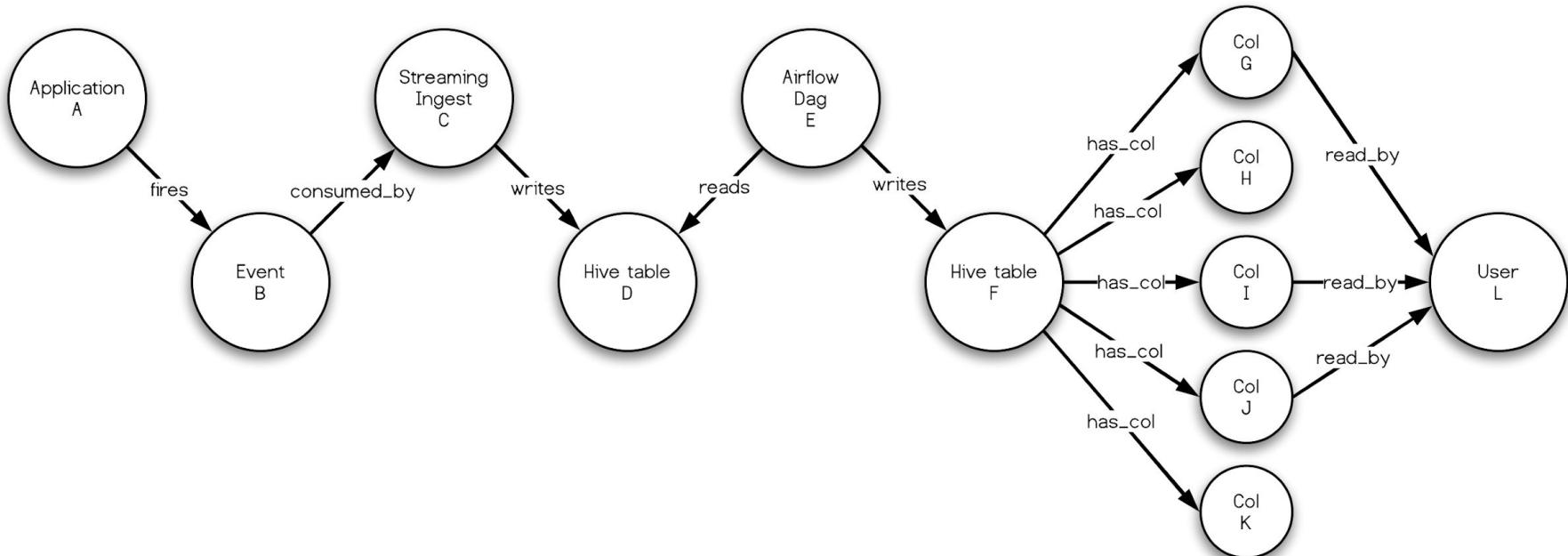
Apache **Atlas**

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

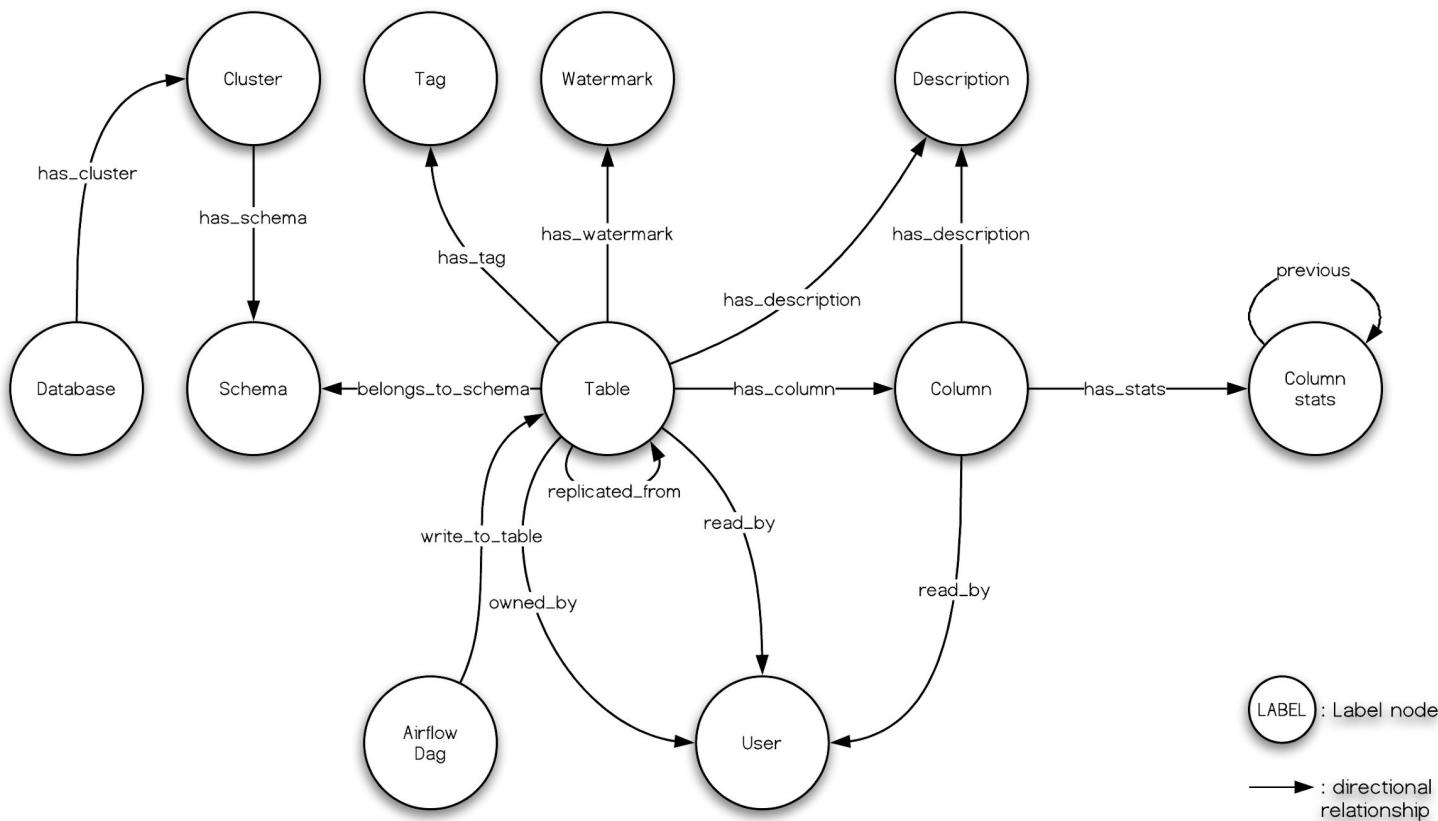
Trade Off #1

Why choose Graph database

Why Graph database?



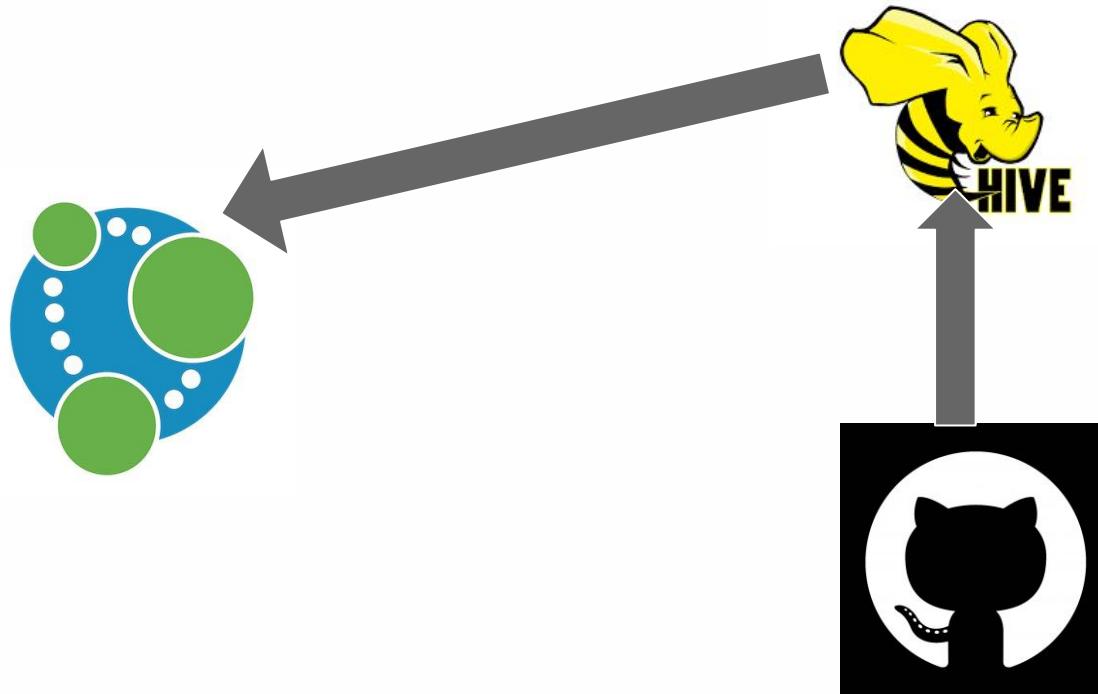
Why Graph database?



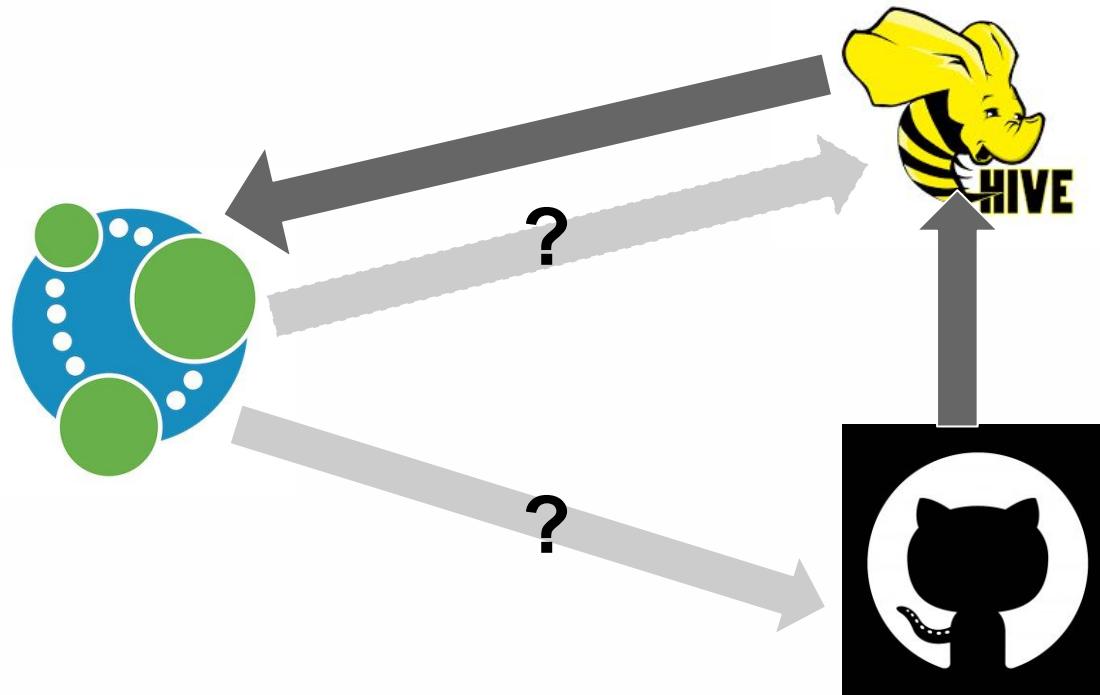
Trade Off #2

Why not propagate the
metadata back to source

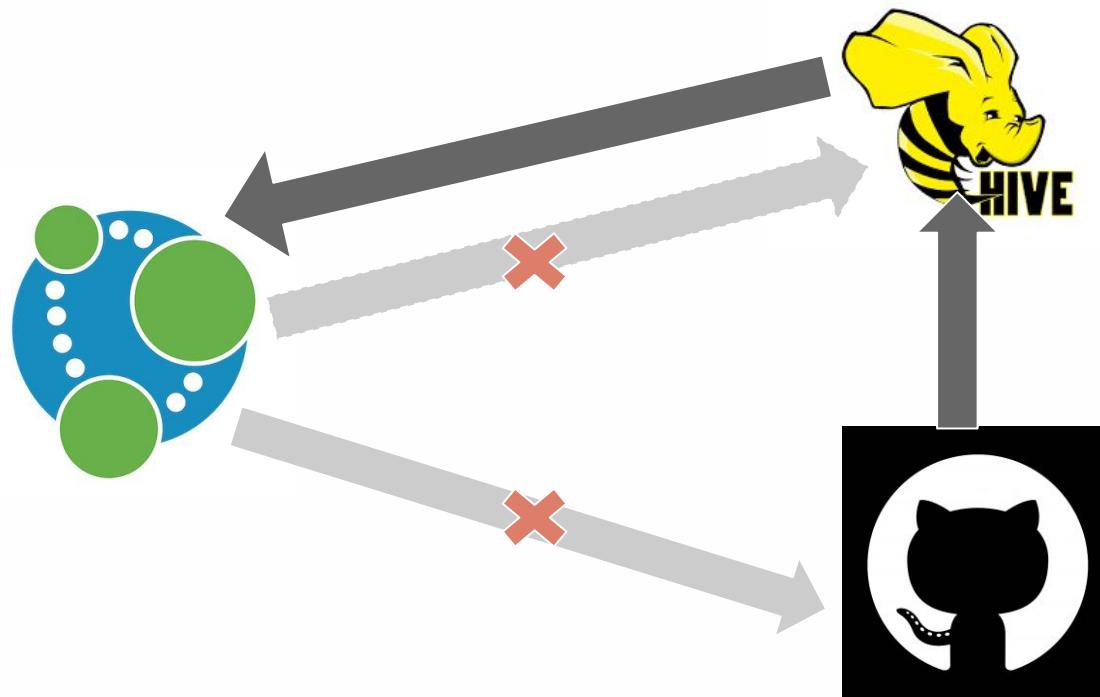
Why not propagate the metadata back to source



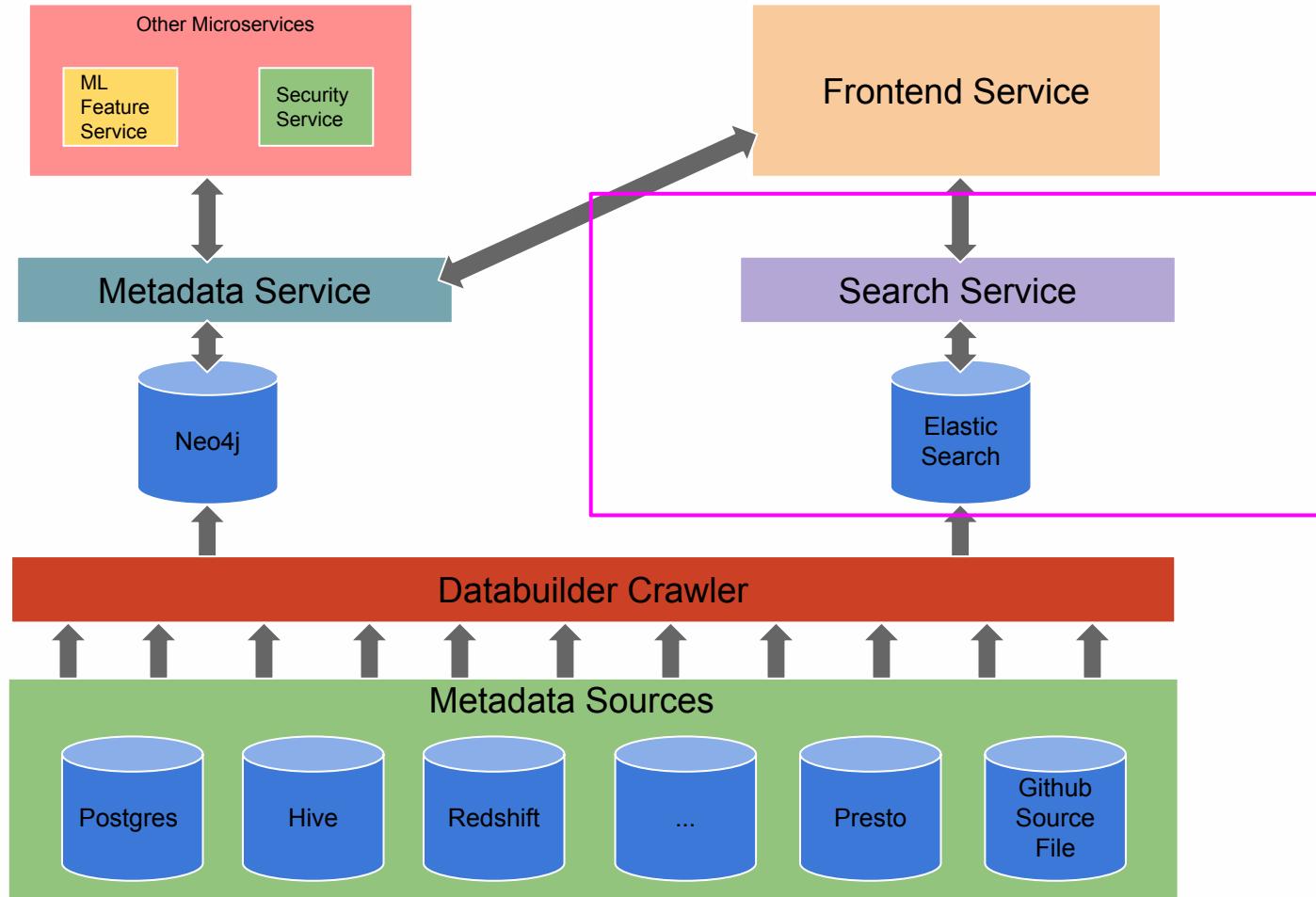
Why not propagate the metadata back to source



Why not propagate the metadata back to source



3. Search Service



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

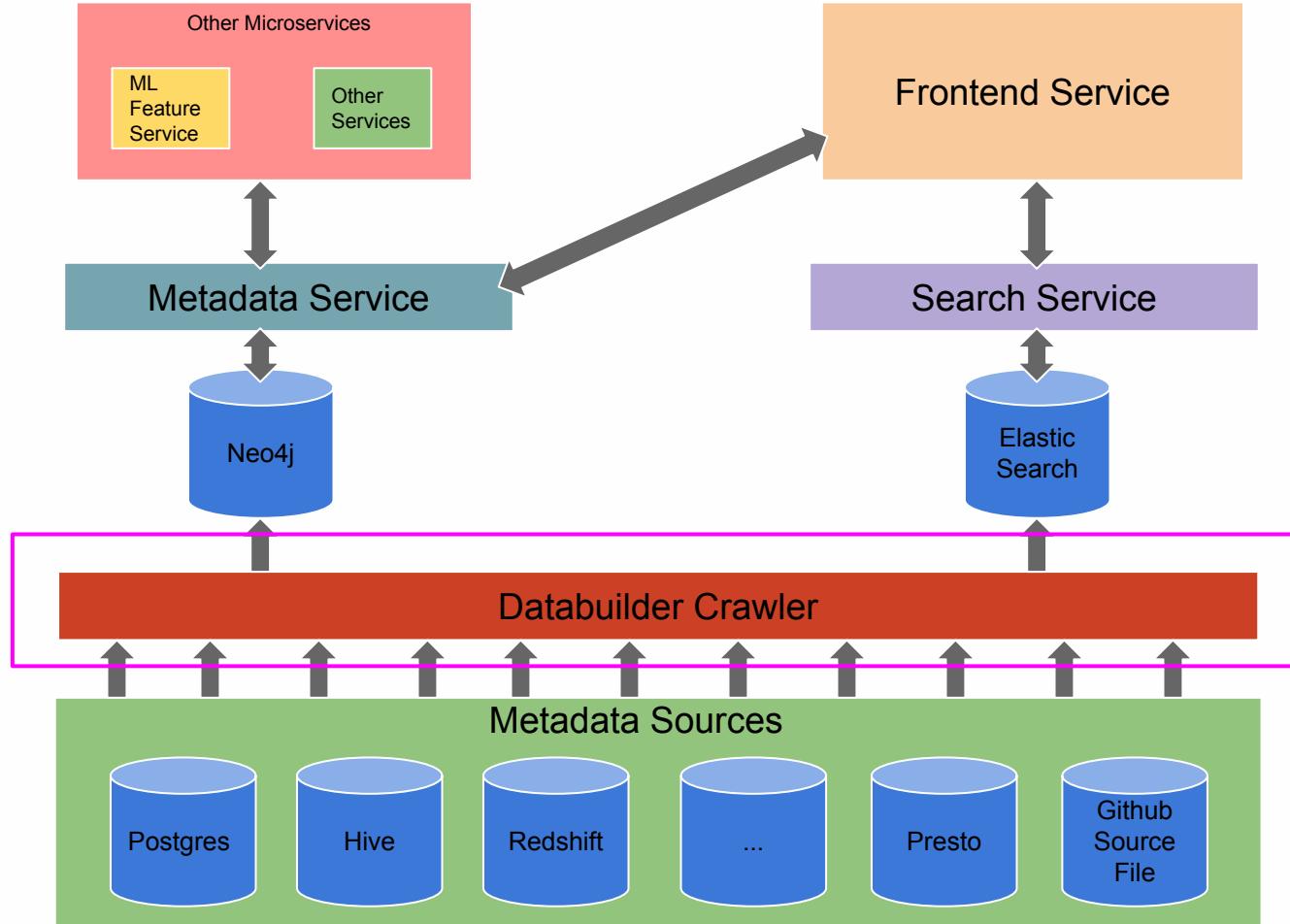
Challenge #1

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



Challenge #1

Various forms of metadata

Metadata Sources @ Lyft



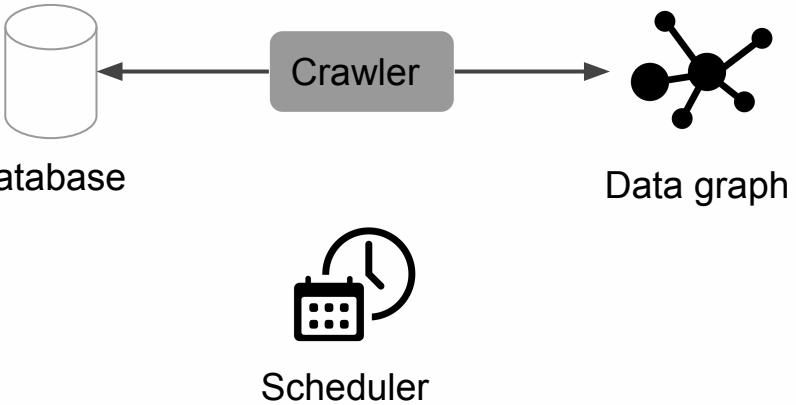
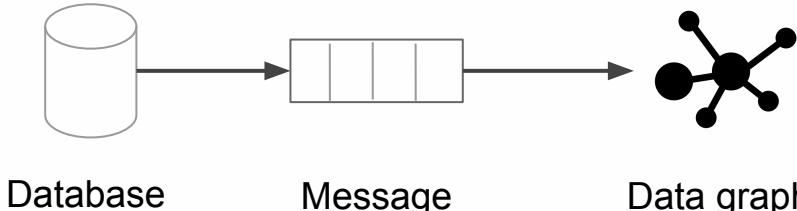
Metadata - Challenges

- **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
 - ...

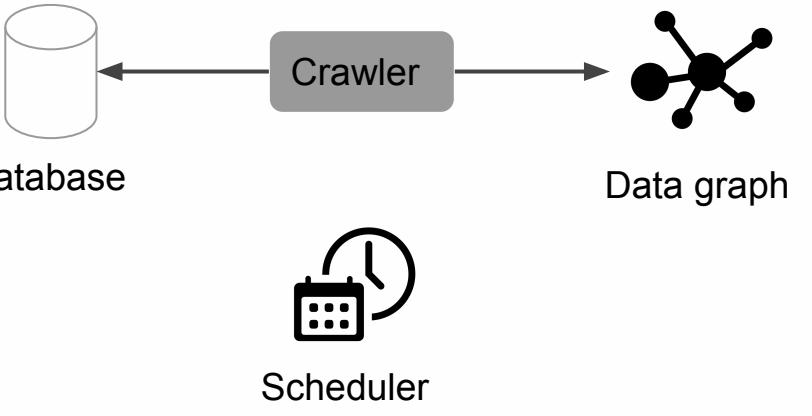
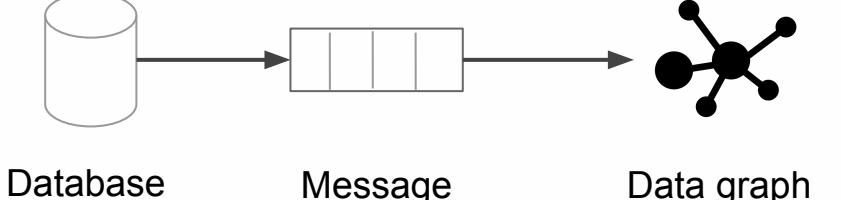
Challenge #2

Pull model vs Push model

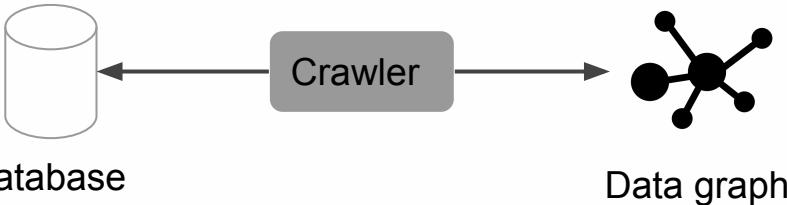
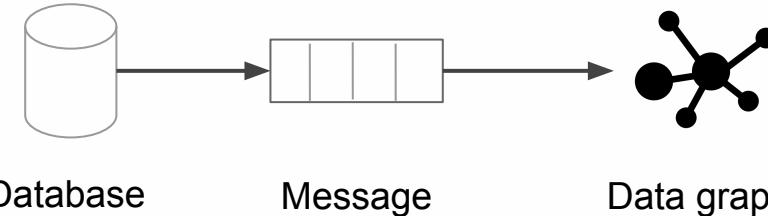
Pull model vs. Push model

Pull Model	Push Model
<ul style="list-style-type: none">Periodically update the index by pulling from the system (e.g. database) via crawlers.  <p>The diagram illustrates the Pull Model. It shows a vertical cylinder labeled "Database" on the left, a rounded rectangle labeled "Crawler" in the center, and a complex network of nodes labeled "Data graph" on the right. Arrows indicate a cyclical flow: one from the Database to the Crawler, another from the Crawler to the Data graph, and a third from the Data graph back to the Database. Below the Crawler is a icon of a calendar with a clock, labeled "Scheduler".</p>	<ul style="list-style-type: none">The system (e.g. database) pushes metadata to a message bus which downstream subscribes to.  <p>The diagram illustrates the Push Model. It shows a vertical cylinder labeled "Database" on the left, a rectangular box divided into four smaller boxes labeled "Message queue" in the middle, and a complex network of nodes labeled "Data graph" on the right. A single horizontal arrow points from the Database to the Message queue, and another arrow points from the Message queue to the Data graph.</p>

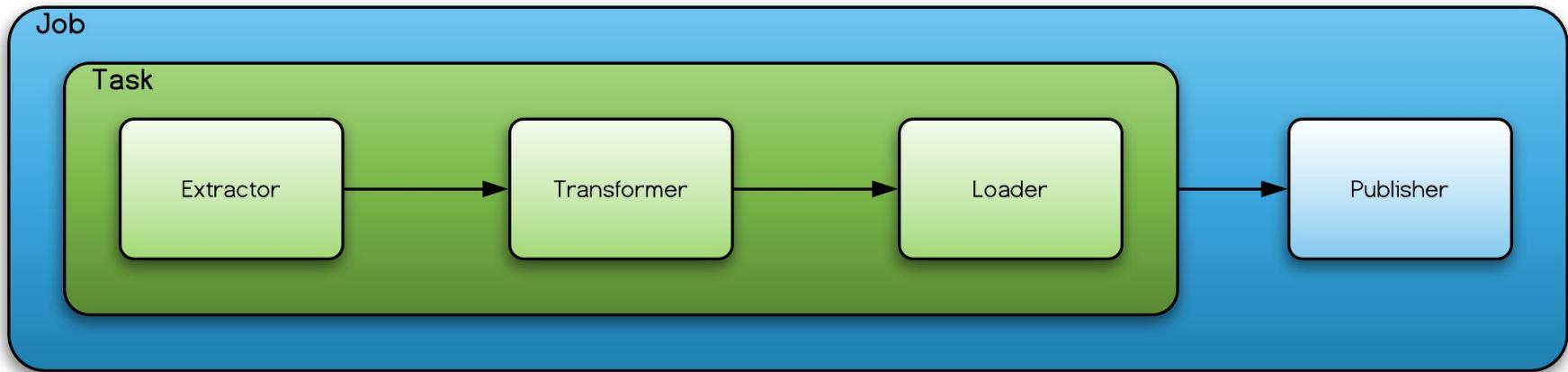
Pull model vs. push model

Pull Model	Push Model
<ul style="list-style-type: none">• Onus of integration lays on data graph• No interface to prescribe, hard to maintain crawlers  <p>The diagram illustrates the Pull Model architecture. It features a central Crawler node (gray rounded rectangle) connected by arrows to both a Database (cylinder icon) and a Data graph (a cluster of nodes connected by lines). Below the crawler is a Scheduler icon (calendar with a clock), which is connected to the crawler by a curved arrow.</p>	<ul style="list-style-type: none">• Onus of integration lies on database• Message format serves as the interface• Allows for near-real time indexing  <p>The diagram illustrates the Push Model architecture. It shows a sequential flow from left to right: a Database (cylinder icon) has an arrow pointing to a Message queue (rectangle divided into four horizontal sections), which in turn has an arrow pointing to a Data graph (cluster of nodes).</p>

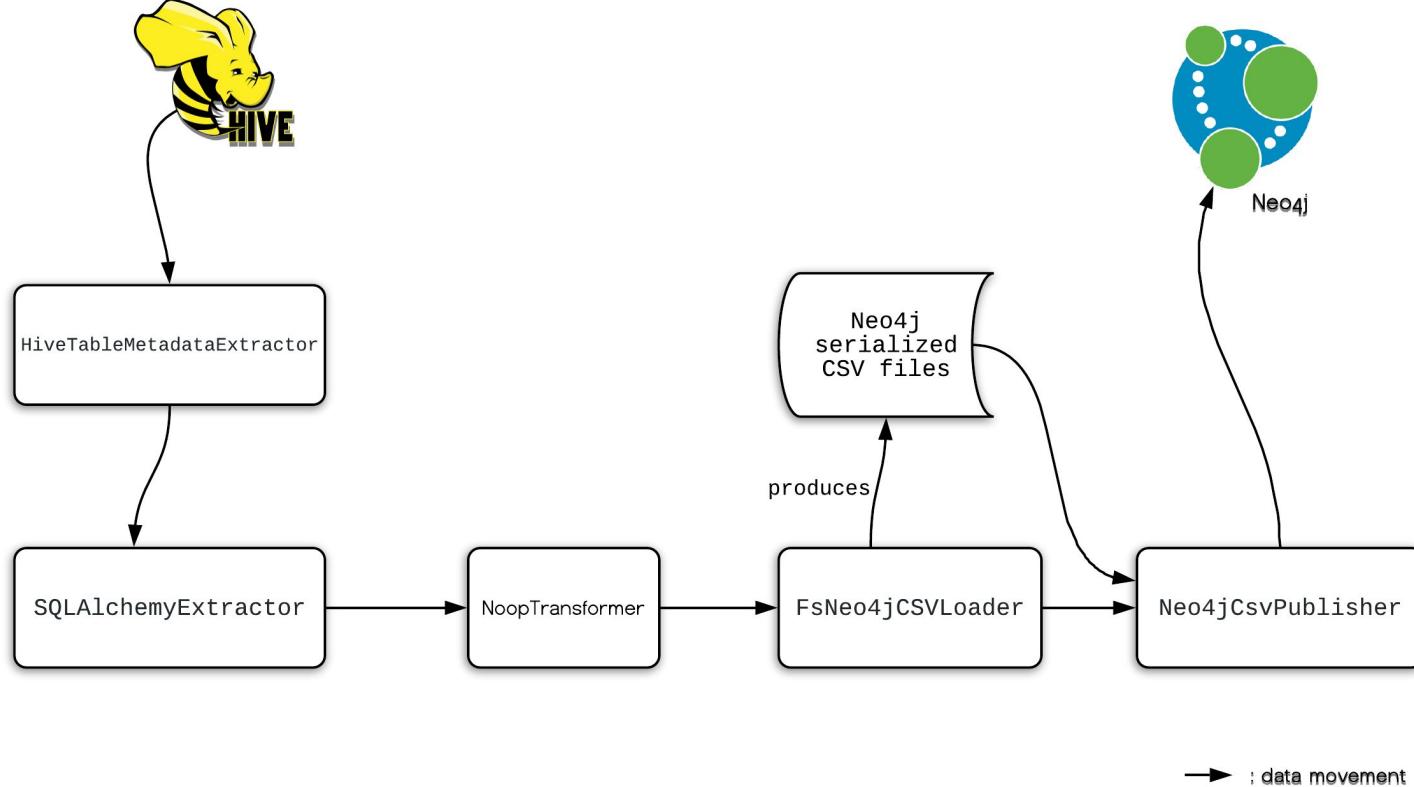
Pull model vs. push model

Pull Model	Push Model
<ul style="list-style-type: none">• Onus of integration lays on data graph• No interface to prescribe, hard to maintain crawlers  <p>Database Data graph</p> <p>Preferred if</p> <ul style="list-style-type: none">• Waiting for indexing is ok• Working with “strapped” teams• There’s already an interface	<ul style="list-style-type: none">• Onus of integration lies on database• Message format serves as the interface• Allows for near-real time indexing  <p>Database Message queue Data graph</p> <p>Preferred if</p> <ul style="list-style-type: none">• Near-real time indexing is important• Clean interface doesn’t exist• Other tools like Wherehows are moving towards Push Model

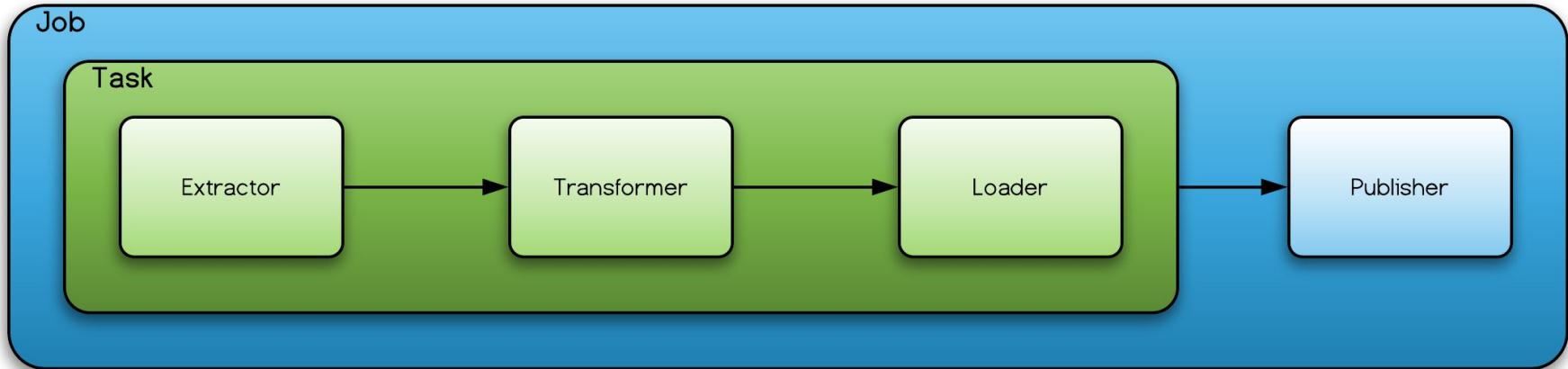
4. Databuilder



Databuilder in action



How are we building data? Databuilder



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

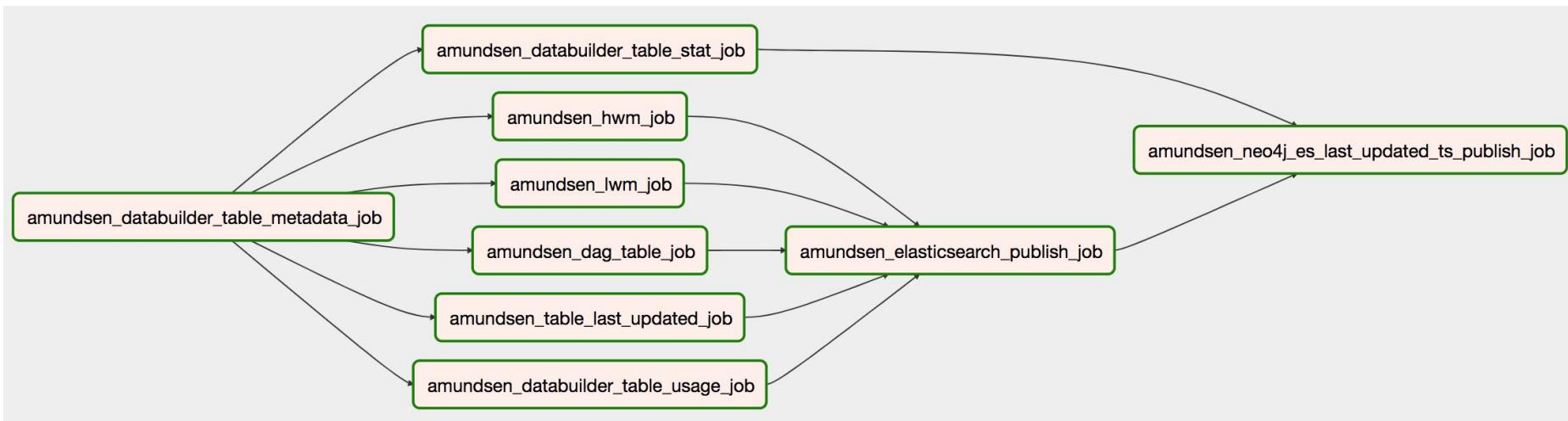
```
job = DefaultJob(conf=job_config,  
                 task=task,  
                 publisher=Neo4jCsvPublisher())
```

```
# run job  
job.launch()
```

How is databuilder orchestrated?



Amundsen uses Apache Airflow to orchestrate Databuilder jobs



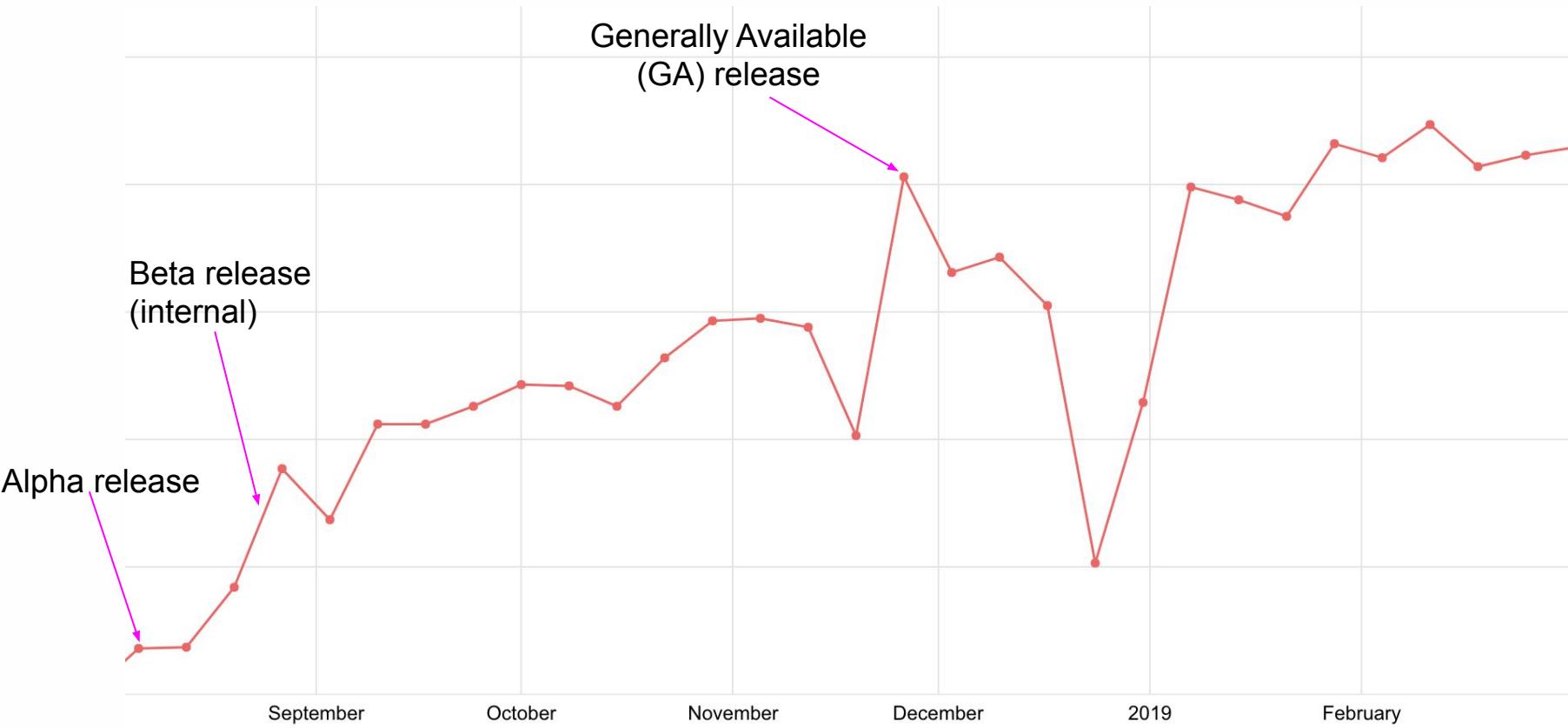
What's next?

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



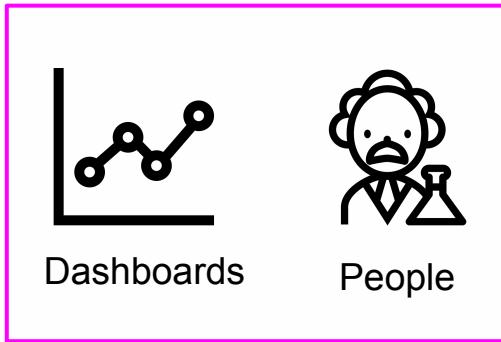
Impact - Amundsen at Lyft



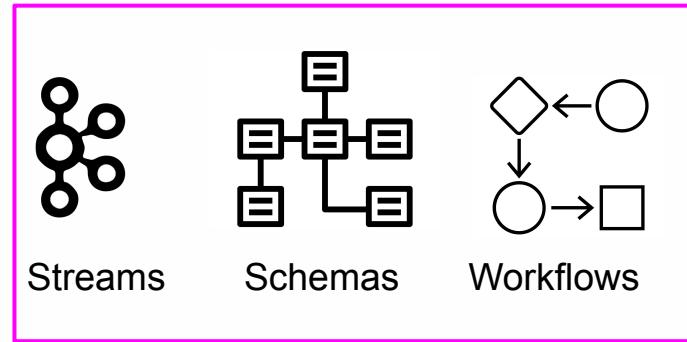
Adding more kinds of data resources



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



Mark Grover | @mark_grover

Tao Feng | @feng-tao

Slides at go.lyft.com/datadiscoveryslides

Blog post at go.lyft.com/datadiscoveryblog

Icons under Creative Commons License from <https://thenounproject.com/>



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

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

[See passes & pricing](#)



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
