FEATURE STORE SUMMIT

12-13 OCTOBER | 08:30 AM - 4:00 PM PT

ORGANIZED BY HOPSWORKS

Twitter's Management of ML Features in Dynamic Environments

David Liu
Machine Learning Engineer
Twitter

https://www.linkedin.com/in/mavysavydav/

@mavysavydav on Twitter

What is a Feature Store?

- Set of standards?
- A framework?
- Metadata management?
- Databases?

Feature Stores for ML

Is a feature store just a rebranding of existing tools/systems?

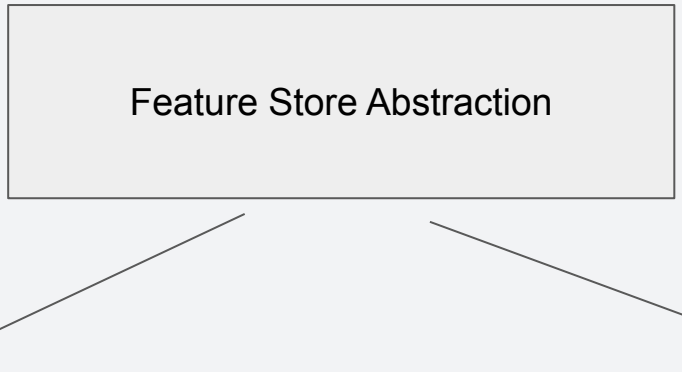
- Datastores / databases / data warehouses
- High load data fetching in online contexts
- SQL Joins
- Map-reduce based joins
- Data processing jobs management
- Workflow management
- Config management

Core Tenets of a Feature Store

In order of priority:

1. Ensure consistency
2. Increase experimental/productionization velocity
3. Shareability of features

What should a feature store be in the user's perspective? (Consuming features)



Online serving

API that takes in references to features and ids, and returns the feature data for those ids.

Experimentation (Offline)

API that takes in references to features and data, and returns the data with the features joined in.

What should a feature store be in the user's perspective? (Producing/Registering features)

Fill out / implement the provided config template for the recurring data job
-Specify data source
-Specify any transform logic
-Specify any needed metadata
-Set schedule

Register Features
-Provide address of where the features are stored
-Specify the features to include in the feature store and which entity they're associated with
- Specify any metadata

Feature Stores for ML

Is a feature store just a rebranding of existing tools/systems?

- Datastores / databases / data warehouses
- High load data fetching in online contexts
- SQL Joins
- Map-reduce based joins
- Data processing jobs management
- Workflow management
- Config management

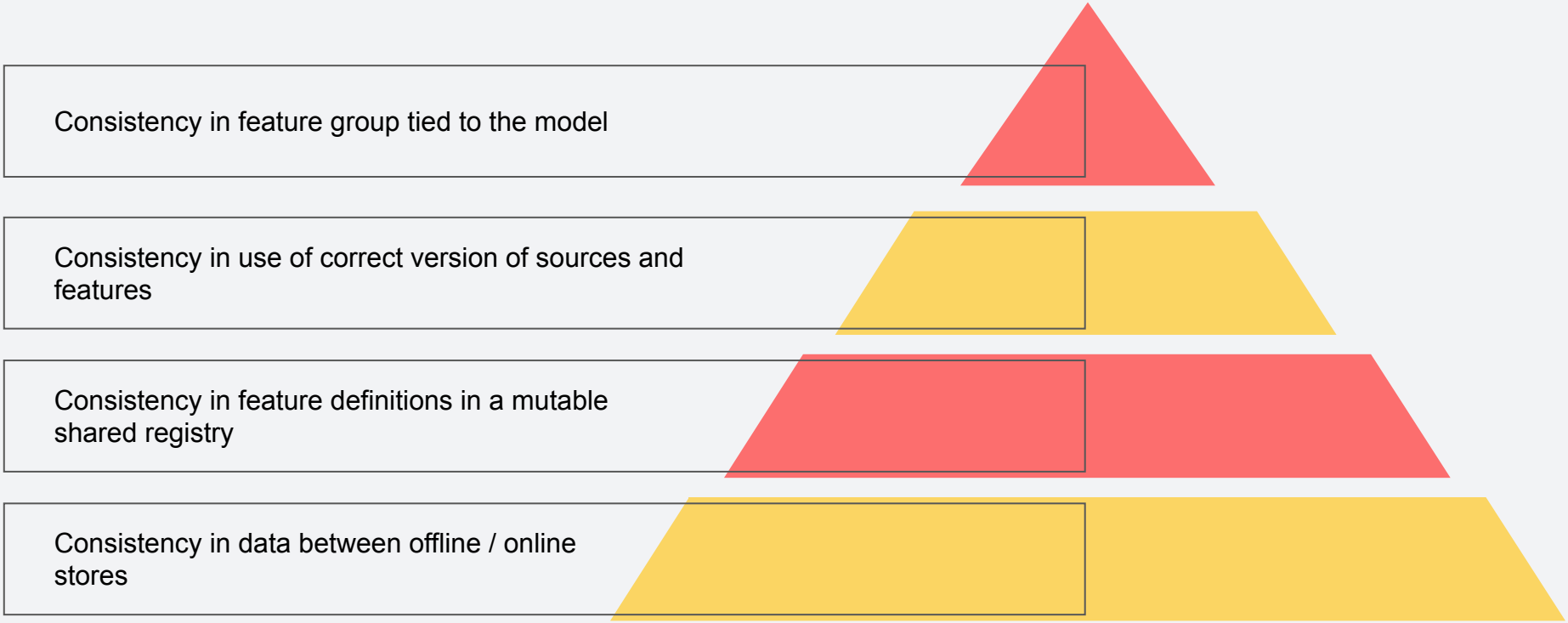
Don't need to worry about these details!!

Core Tenets of a Feature Store

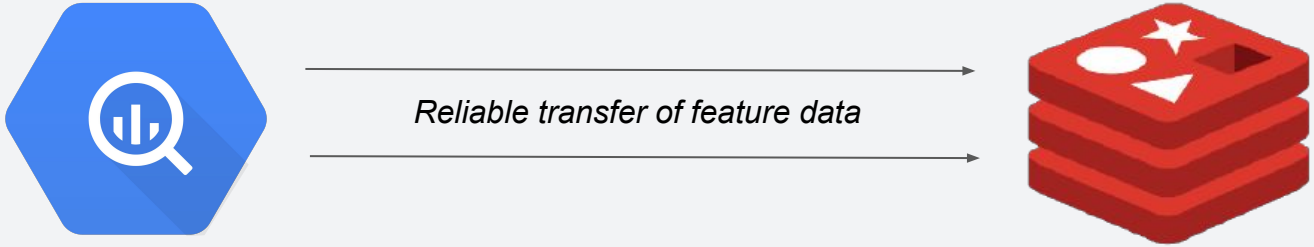
In order of priority:

1. Ensure consistency
2. Increase experimental/productionization velocity
3. Shareability of features

Consistency Pyramid



Consistency in data between offline / online stores



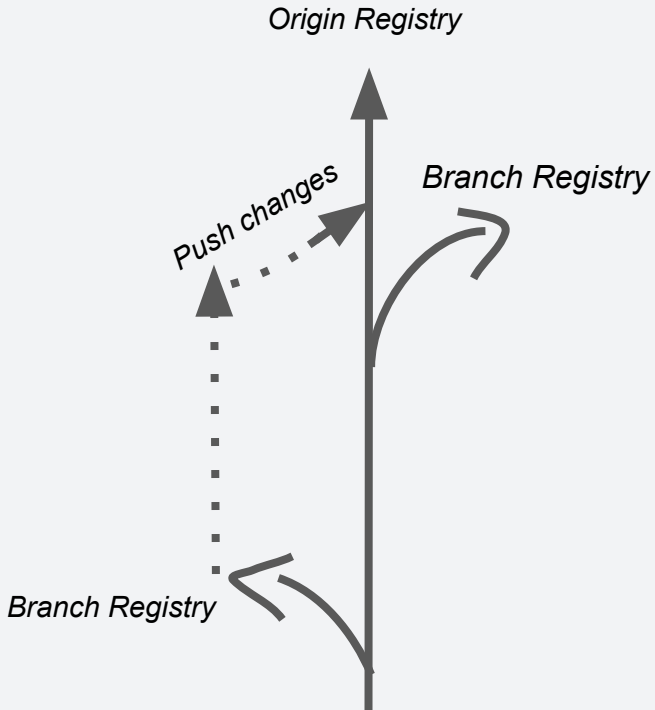
Standardized schemas and structure in offline store

Standardized schemas and structure in online store

Consistency in feature definitions in a mutable shared registry

Using the same model as git:

origin/master + branches



Feature Stores for ML

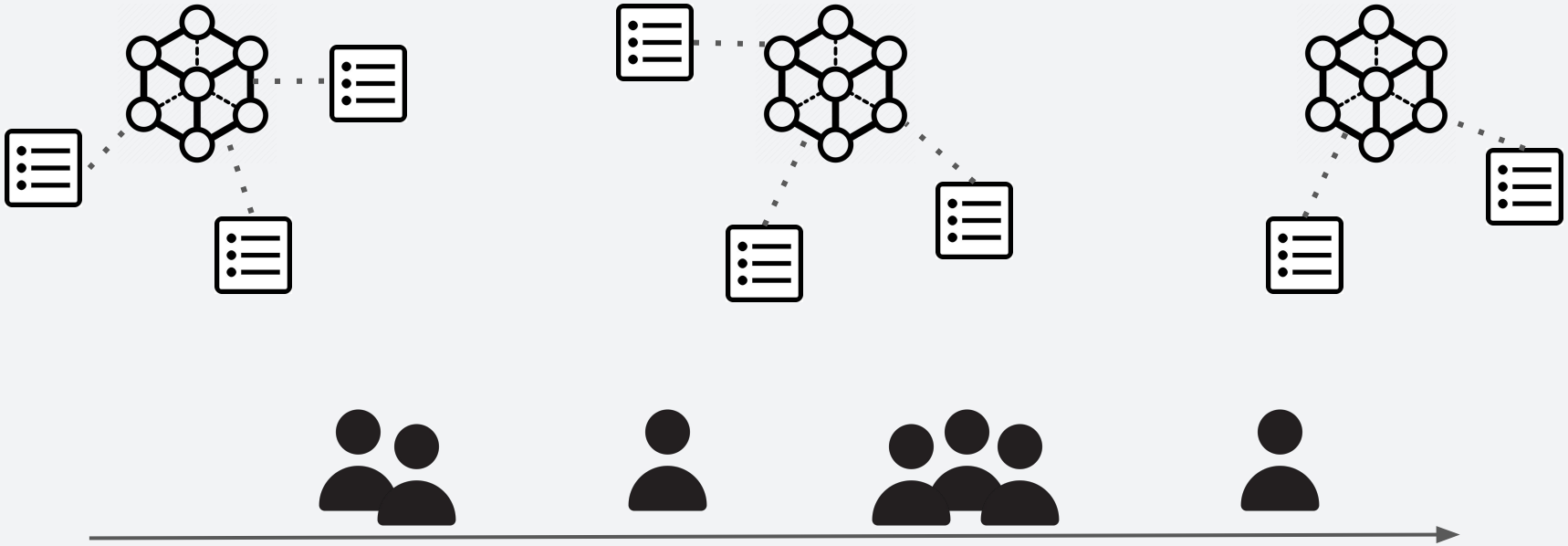
Consistency in use of correct version of sources and features

- First class support or can it just be incorporated as part of the naming scheme or extra metadata fields?

- Which versioning scheme would most reduce the chance of human error?

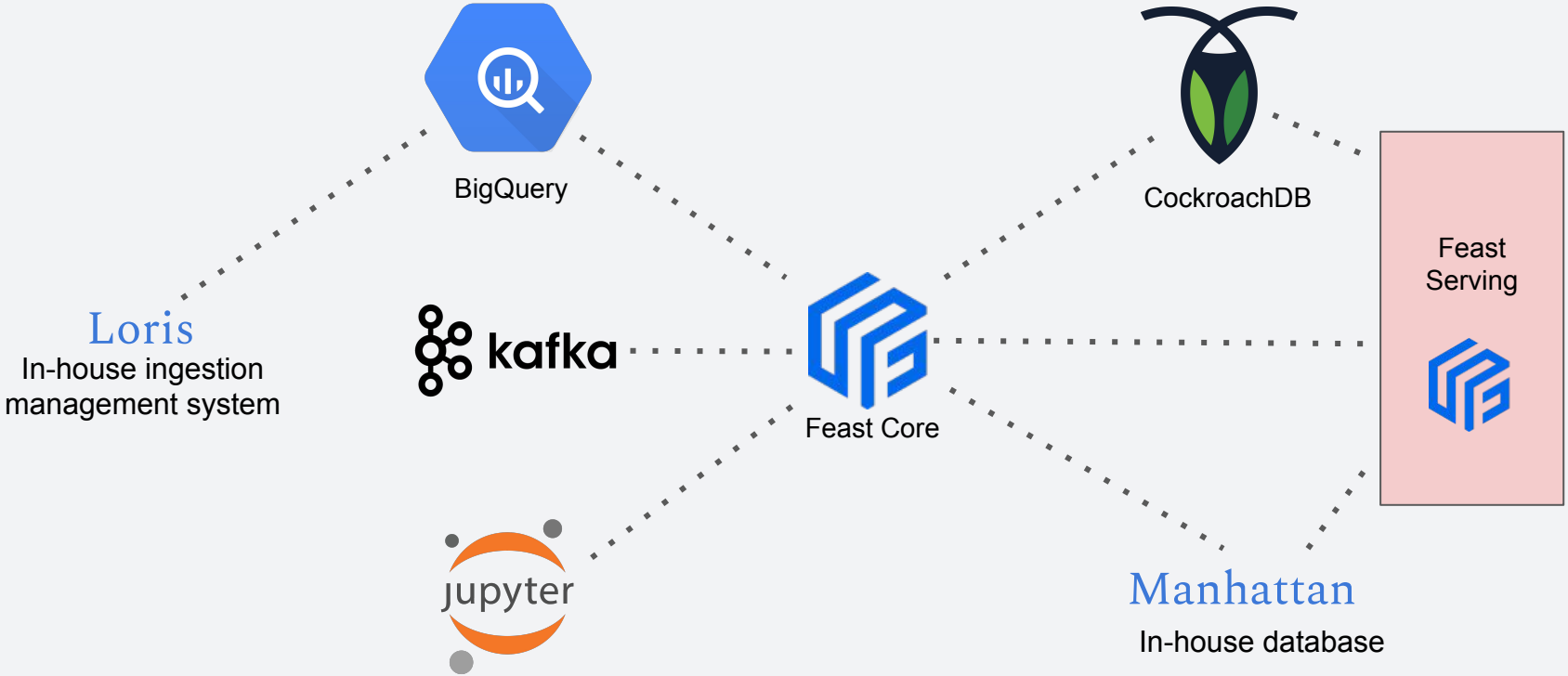
- Intuitive grouping of feature versions vs optimizing for datastore / data warehouse / database performance?

Consistency in feature group tied to the model

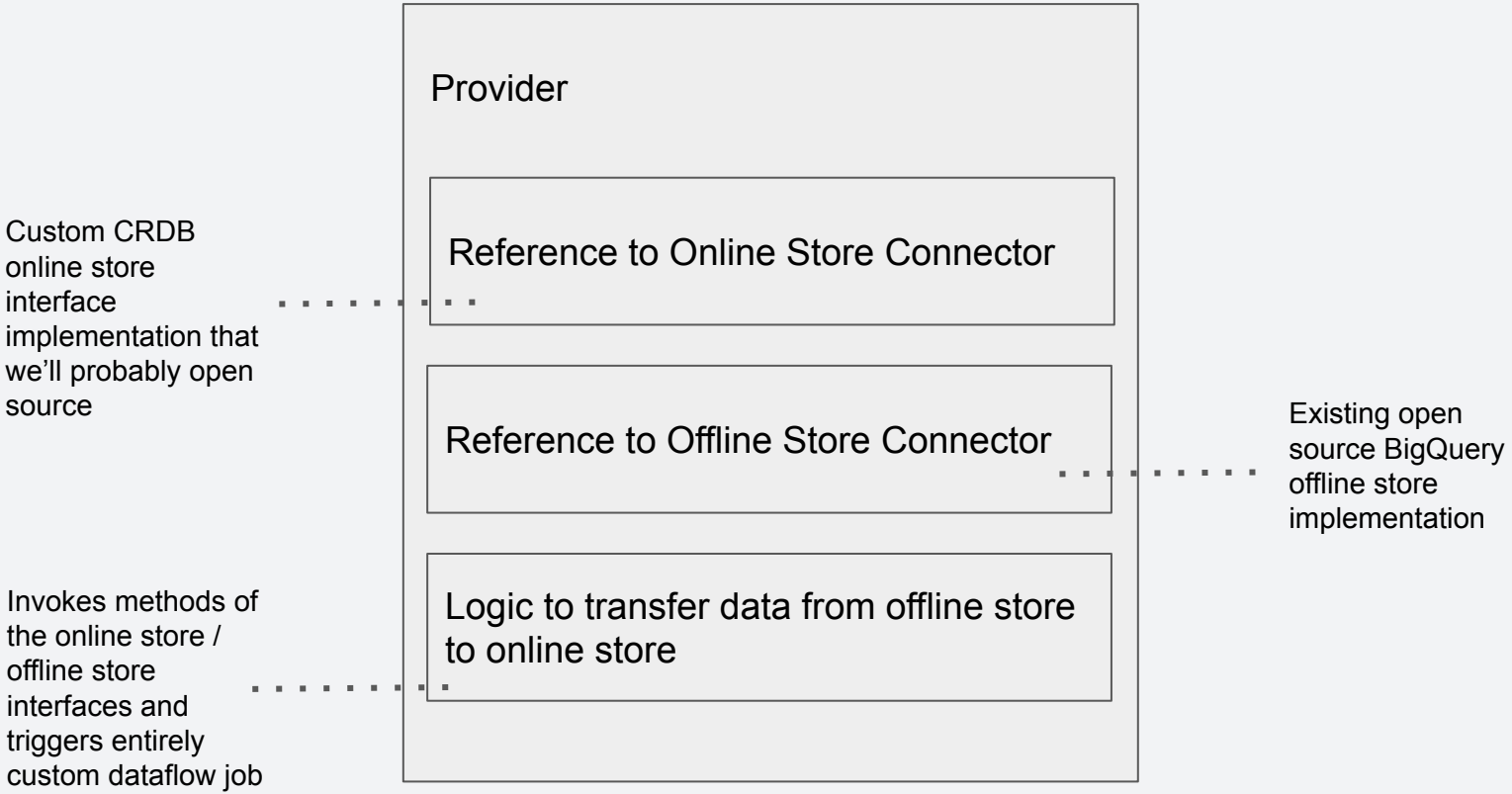


Time elapsing as the model is developed / retrained / reworked in a highly collaborative context

How Twitter solves some of these issues:



The Flexibility of the Feast Provider Abstraction



Feature Stores for ML

Core Tenets of a Feature Store

In order of priority:

1. Ensure consistency
2. Increase experimental/productionization velocity
3. Shareability of features

Increase experimental/productionization velocity

Integrate well with other ML components

- TFX Example Gens
- Data types consistency throughout the ML pipeline
- Kubeflow

Increase experimental/productionization velocity

Hot / live updates of feature sets that services use + on/off toggles

Increase experimental/productionization velocity

Automatically aggregation of experiment details that can be viewed via UI

- Model used
- Feature set used
- Which toggles were on/off

Increase experimental/productionization velocity

Ease of discovering which permissions are needed for certain feature data, ease of requesting it, and turnaround time for approval.

Increase experimental/productionization velocity

Ease of capacity planning / requesting capacity

Core Tenets of a Feature Store

In order of priority:

1. Ensure consistency
2. Increase experimental/productionization velocity
3. Shareability of features

Shareability of Features

- Centralized Catalog
- Discovery
  - Usage tracking
  - Feature importance
- Reliability of using other teams' features. Can the data be trusted?
- Deprecation process
- Get a data sample

The challenges we haven’t figured out yet

- How to minimize the amount of support in this two-sided marketplace of producers and consumers? How do we maximize self-serve?
- Access control that does its job but doesn’t introduce much friction
- What incentivizes sharing? Why would they want to share if they may have to provide maintenance / support for those pipelines?
- Massive datasets: 5000+ features, billions of rows
- How can a scribing system hook onto our new platform?

Thank you!

https://www.linkedin.com/in/mavysavydav/

@mavysavydav