Meeting of the LF AI & Data Technical Advisory Council (TAC)

August 25, 2022

LF AI & DATA

Antitrust Policy

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Undergone LLP, which provides legal counsel to the Linux Foundation.

Recording of Calls

Reminder:

TAC calls are recorded and available for viewing on the TAC Wiki

Reminder: LF AI & Data Useful Links

- Web site: lfaidata.foundation
- Wiki: wiki.lfaidata.foundation
- GitHub: github.com/lfaidata
- Landscape: https://landscape.lfaidata.foundation or https://l.lfaidata.foundation
- Mail Lists: https://lists.lfaidata.foundation
- Slack: https://slack.lfaidata.foundation
- Youtube: https://www.youtube.com/channel/UCfasaeqXJBCAJMNO9HcHfbA
- LF AI Logos: https://github.com/lfaidata/artwork/tree/master/lfaidata
- LF AI Presentation Template: https://drive.google.com/file/d/1eiDNJvXCqSZHT4Zk_-czASlz2GTBRZk2/view?usp=sharing

- Events Page on LF AI Website: https://lfaidata.foundation/events/
- Events Calendar on LF AI Wiki (subscribe available): https://wiki.lfaidata.foundation/pages/viewpage.action?pageId=12091544
- Event Wiki Pages: https://wiki.lfaidata.foundation/display/DL/LF+AI+Data+Foundation+Events

Agenda

- Roll Call (2 mins)
- Approval of Minutes from previous meeting (2 mins)
- Feathr – Joining incubation from LinkedIn (50 minutes)
- LF AI General Updates (2 min)
- Open Discussion (2 min)

TAC Voting Members - Please note

Please ensure that you do the following to facilitate smooth procedural quorum and voting processes:

- Change your Zoom display name to include your First/Last Name, Company/Project Represented
  - example: Nancy Rausch, SAS
- State your First/Last Name and Company/Project when submitting a motion
  - example: First motion, Nancy Rausch/SAS

TAC Voting Members

Note: we still need a few designated backups specified on [wiki]

Member Representatives (8 out of 16 required for quorum)

Member Company or Graduated Project	Membership Level or Project Level	Voting Eligibility	Country	TAC Representative	Designated TAC Representative Alternates
4paradigm	Premier	Voting Member	China	Zhongyi Tan	
Baidu	Premier	Voting Member	China	Ti Zhou	Daxiang Dong, Yanjun Ma
Ericsson	Premier	Voting Member	Sweden	Rani Yadav-Ranjan	
Huawei	Premier	Voting Member	China	Howard (Huang Zhipeng)	Charlotte (Xiaoman Hu) , Leon (Hui Wang)
Nokia	Premier	Voting Member	Finland	@Michael Rooke	@Jonne Soininen
OPPO	Premier	Voting Member	China	Jimin Jia	
SAS	Premier	Voting Member	USA	*Nancy Rausch	JP Trawinski
ZTE	Premier	Voting Member	China	Wei Meng	Liya Yuan
Adversarial Robustness Toolbox Project	Graduated Technical Project	Voting Member	USA	Beat Buesser	
Angel Project	Graduated Technical Project	Voting Member	China	Bruce Tao	Huaming Rao
Egeria Project	Graduated Technical Project	Voting Member	UK	Mandy Chessell	Nigel Jones, David Radley, Maryna Strelchuk, Ljupcho Palashevski, Chris Grote
Flyte Project	Graduated Technical Project	Voting Member	USA	Ketan Umare	
Horovod Project	Graduated Technical Project	Voting Member	USA	Travis Addair	
Milvus Project	Graduated Technical Project	Voting Member	China	Xiaofan Luan	Jun Gu
ONNX Project	Graduated Technical Project	Voting Member	USA	Alexandre Eichenberger	Prasanth Pulavarthi, Jim Spohrer
Pyro Project	Graduated Technical Project	Voting Member	USA	Fritz Obermeyer	

*Current TAC Chair

Minutes approval

LF AI & DATA

Approval of August 11, 2022 Minutes

Draft minutes from the August 11 TAC call were previously distributed to the TAC members via the mailing list

Proposed Resolution:

- That the minutes of the August 11 meeting of the Technical Advisory Council of the LF AI & Data Foundation are hereby approved.

Feathr

Hangfei Lin

LF AI & DATA

Feathr

An Enterprise-Grade, High Performance Feature Store

github.com/linkedin/feathr

Hangfei Lin

Aug 25th, 2022

LinkedIn

Battle-tested at LinkedIn for 5+ years, and now **Open Sourced** on Github

github.com/linkedin/feathr

Now natively integrated with Azure and Databricks

Agenda

1 Why Donate

2 Problem statement

3 Solution

4 Impacts and Roadmap

Why Donate Feathr to Linux Foundation

1. Neutral Holding

- Vendor-neutral
- Not for profit

2. open governance model

- Instill trust in contributors and adopters in the management of the project

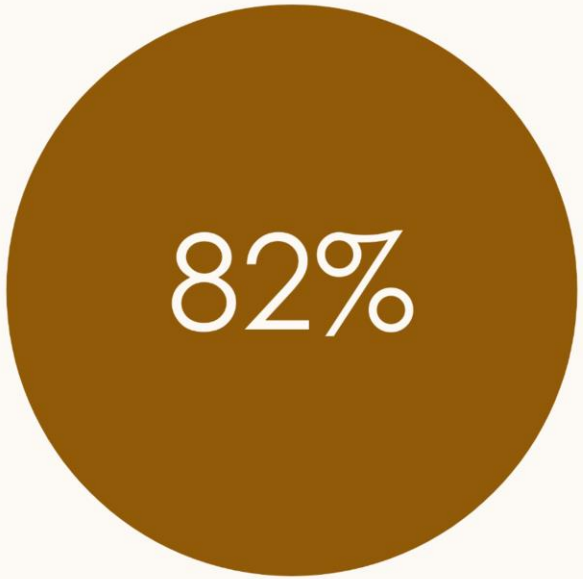
3. Growing community

- become an integral part of the wider AI community
- enable collaboration and the creation of new opportunities with the wider Linux open-source community
- support from a world-wide developer community

Problem Statement

Survey in Forbes: Big data engineering for AI

Re: Building training sets + Cleaning and organizing data + Collecting datasets

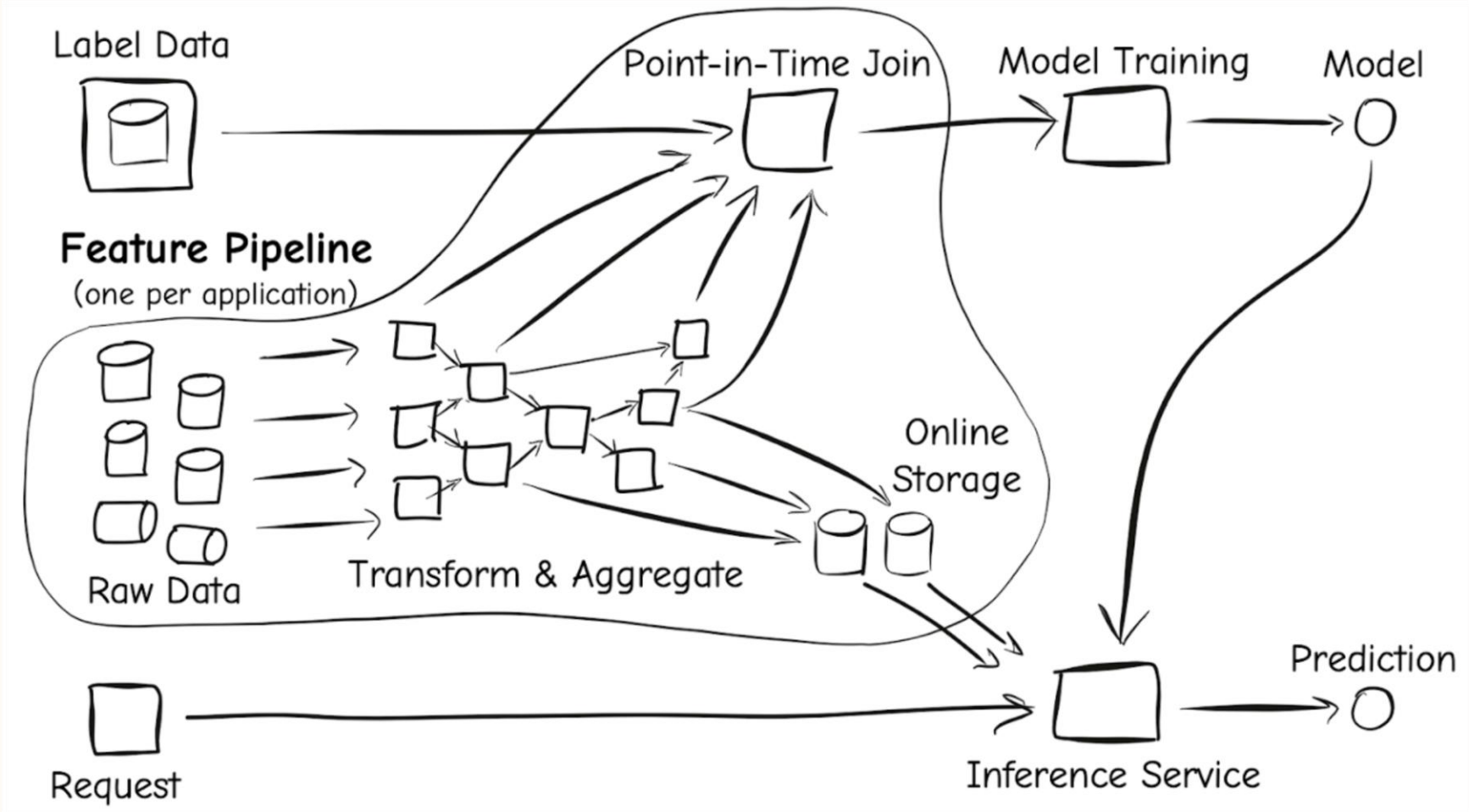


Time spent on data preparation

Respondents said data preparation 'least enjoyable' part of data science

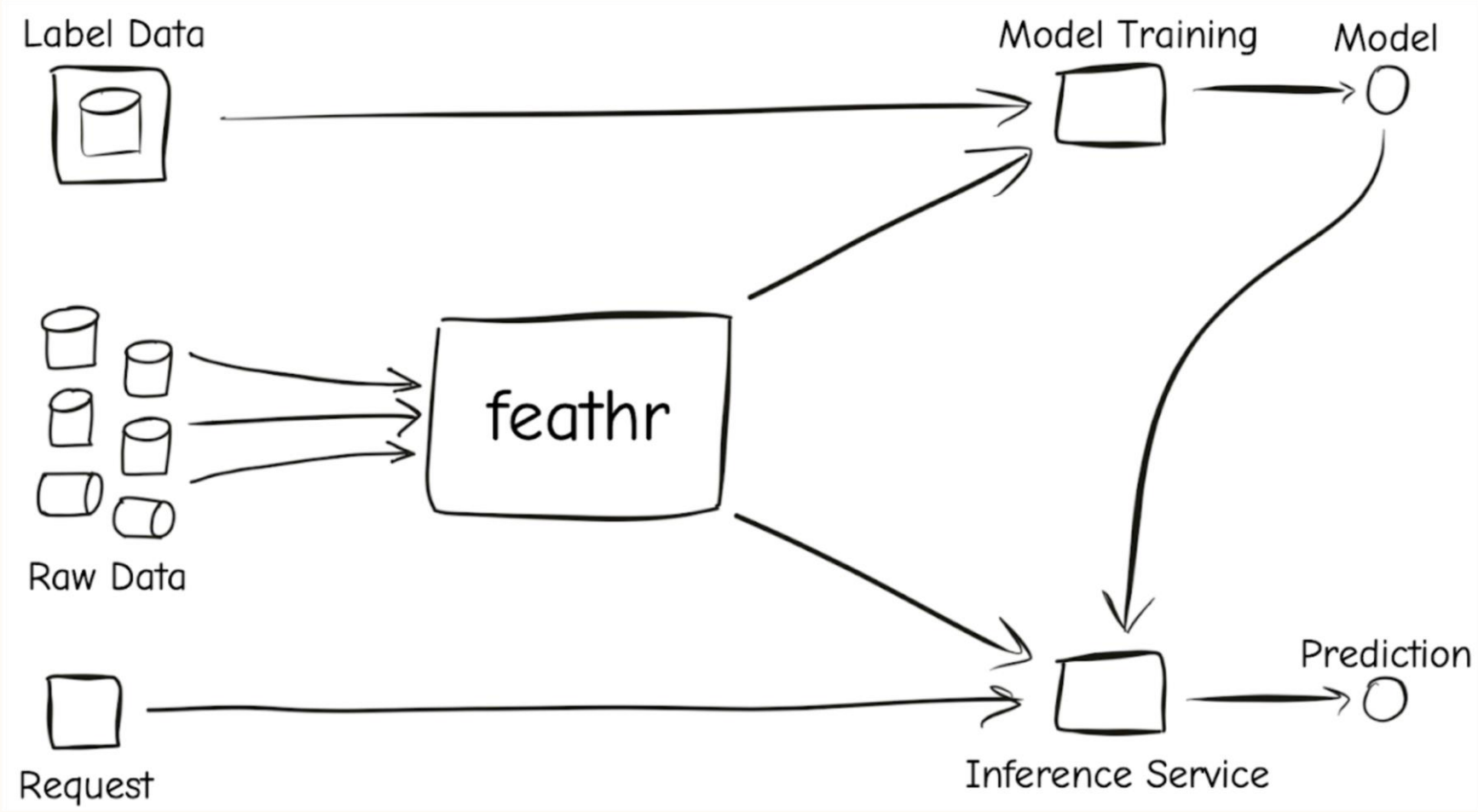
Why Feathr

Handle all the complex weight-lifting and "boring" work automatically



Why Feathr

Handle all the complex weight-lifting and “boring” work automatically



Problem: The complexity of feature preparation pipelines

1. Coupling

- Feature preparation code being coupled with model/application code
- Different programming APIs for different environments, e.g. online, offline, nearline, etc.
- Boilerplate and repetitive work
- Hard to test and debug

2. Train/Inference Skew

- Offline training and online inference usually require different data serving pipelines.
- Ensuring features are generated consistently is time intensive and error prone.
- Teams are deterred from using real time data for inferencing due to the difficulty in serving the right data.

3. Re-use and share

- The cost of building and maintaining feature pipelines was borne redundantly across many teams.
- Team-specific pipelines also made it impractical to reuse features across projects. e.g. no common type system, no common feature namespace

Solution

What is Feathr

An abstraction layer between raw data and model

- **Define** features based on raw data sources using simple APIs.
- **Get** those features by their names during model training and model inferencing.
- **Share** features across your team and organizations.

Highlights

Cloud-native

Native integration achieved with our colleagues at Azure

Rich transformation

Python built-in transformations and PySpark UDF, on-demand evaluation

Scalable &High Performance

Highly optimized feature compute engine

How to Use it(example)

Use case: Create Feature Definition

Load raw source data, and define transformation

```
batch_source = HdfsSource(
    name="nycTaxiBatchSource",                              # Source name to enrich your metadata
    path="abfss://green_tripdata_2020-04.csv",              # Path to your data
    event_timestamp_column="lpep_dropoff_datetime",         # Event timestamp for point-in-time correctness
    timestamp_format="yyyy-MM-dd HH:mm:ss")                 # Supports various fromats inculding epoch

trip_id = TypedKey(key_column="trip_id",
                   key_column_type=ValueType.INT64,
                   description="trip id")

features = [
    Feature(name="f_trip_distance",                         # Ingest feature data as-is
            feature_type=FLOAT,
            key=trip_id),
    Feature(name="f_is_long_trip_distance",
            feature_type=BOOLEAN,
            transform="cast_float(trip_distance)>30",
            key=trip_id)      # SQL-like syntax to transform raw data into feature
]

anchor = FeatureAnchor(name="anchor_features",              # Features anchored on same source
                       source=batch_source,
                       features=features)
```

Use Case - Streaming Feature

Create features from streaming source

```
stream_source = KafKaSource(name="kafkaStreamingSource",
                            kafkaConfig=KafkaConfig(brokers=["feathrazureci.servicebus.windows.net:
                                                    topics=["feathrcieventhub"],
                                                    schema=schema)
                            )

driver_id = TypedKey(key_column="driver_id",
                     key_column_type=ValueType.INT64,
                     description="driver id",
                     full_name="nyc driver id")

kafkaAnchor = FeatureAnchor(name="kafkaAnchor",
                            source=stream_source,
                            features=[Feature(name="f_modified_streaming_count",
                                              feature_type=INT32,
                                              transform="trips_today + 1",
                                              key=driver_id),
                                      Feature(name="f_modified_streaming_count2",
                                              feature_type=INT32,
                                              transform="trips_today + 2",
                                              key=driver_id)]
                            )
```

Use case: Build training dataset

Point-in-time Join Correct semantics

```
# Requested features to be joined
# Define the key for your feature
location_id = TypedKey(key_column="DOLocationID",
                       key_column_type=ValueType.INT32,
                       description="location id in NYC",
                       full_name="nyc_taxi.location_id")
feature_query = FeatureQuery(feature_list=["f_location_avg_fare"], key=[location_id])

# Observation dataset settings
settings = ObservationSettings(
  observation_path="abfss://green_tripdata_2020-04.csv",     # Path to your observation data
  event_timestamp_column="lpep_dropoff_datetime",            # Event timepstamp field for your data,
  timestamp_format="yyyy-MM-dd HH:mm:ss")                    # Event timestamp format,  optional

# Prepare training data by joining features to the input (observation) data.
# feature-join.conf and features.conf are detected and used automatically.
feathr_client.get_offline_features(observation_settings=settings,
                                   output_path="abfss://output.avro",
                                   feature_query=feature_query)
```

Use Case – Feature Materialization

Materialize feature values to online storage for realtime access

```
client = FeathrClient()
redisSink = RedisSink(table_name="nycTaxiDemoFeature")
# Materialize two features into a redis table.
settings = MaterializationSettings("nycTaxiMaterializationJob",
sinks=[redisSink],
feature_names=["f_location_avg_fare", "f_location_max_fare"])
client.materialize_features(settings)
```

Use Case - Feature Sharing and Discovery

Share features and discover features

Use Case - Derived Feature

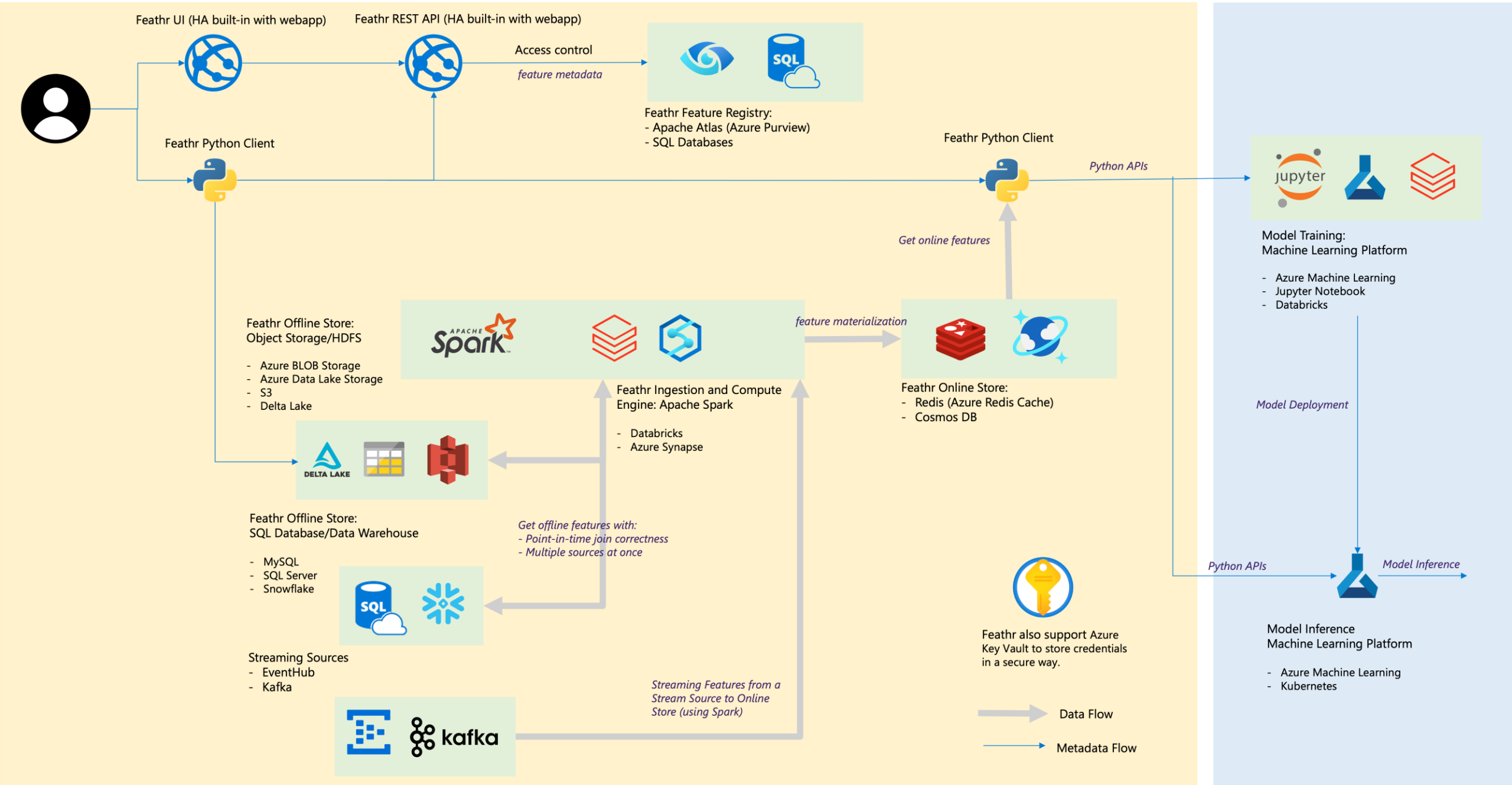
Define features on top of other features

```
# Compute a new feature(a.k.a. derived feature) on top of an existing feature
derived_feature = DerivedFeature(name="f_trip_time_distance",
                                 feature_type=FLOAT,
                                 key=trip_key,
                                 input_features=[f_trip_distance, f_trip_time_duration],
                                 transform="f_trip_distance * f_trip_time_duration")

# Another example to compute embedding similarity
user_embedding = Feature(name="user_embedding", feature_type=DENSE_VECTOR, key=user_key)
item_embedding = Feature(name="item_embedding", feature_type=DENSE_VECTOR, key=item_key)

user_item_similarity = DerivedFeature(name="user_item_similarity",
                                      feature_type=FLOAT,
                                      key=[user_key, item_key],
                                      input_features=[user_embedding, item_embedding],
                                      transform="cosine_similarity(user_embedding, item_embedding)"
```

Architecture


Feathr UI (HA built-in with webapp)
Feathr REST API (HA built-in with webapp)
Access control
feature metadata
Feathr Feature Registry:
- Apache Atlas (Azure Purview)
- SQL Databases
Feathr Python Client
Feathr Python Client
Python APIs
jupyter
Model Training:
Machine Learning Platform
- Azure Machine Learning
- Jupyter Notebook
- Databricks
Get online features
Feathr Offline Store:
Object Storage/HDFS
- Azure BLOB Storage
- Azure Data Lake Storage
- S3
- Delta Lake
feature materialization
Feathr Ingestion and Compute
Engine: Apache Spark
- Databricks
- Azure Synapse
Feathr Online Store:
- Redis (Azure Redis Cache)
- Cosmos DB
Model Deployment
DELTA LAKE
Feathr Offline Store:
SQL Database/Data Warehouse
- MySQL
- SQL Server
- Snowflake
Get offline features with:
- Point-in-time join correctness
- Multiple sources at once
Python APIs
Model Inference
Model Inference
Machine Learning Platform
- Azure Machine Learning
- Kubernetes
Feathr also support Azure
Key Vault to store credentials
in a secure way.
Streaming Sources
- EventHub
- Kafka
Streaming Features from a
Stream Source to Online
Store (using Spark)
kafka
Data Flow
Metadata Flow

Impacts of Feathr

Impacts in LinkedIn

- reduced engineering time required for adding and experimenting with new features from weeks to days
- performed faster than the custom feature processing pipelines that they replaced by as much as 50% amortizing investments in Feathr runtime optimization
- enabled feature sharing between similar applications, leading to significant gains in business metrics

Feathr Community

- Open-sourced
- 876 github stars
- 26 contributors
- 312 commits
- Being adopted by companies from various industries
  - fortune 500 companies building CRM systems,
  - Big Data Analytics consulting company
  - the largest insurance company in the UK
- Presentations and blogs
  - ML Platforms Meetup
  - LinkedIn Eng Blog Post
  - ML Platform Meetup Silicon Valley

Roadmap

What's coming next?

Potential collaboration with LF AI and Data Projects

1 Milvus

- Use Feathr client to produce and push embedding features into Milvus
- Use Feathr client to access the embeddings from Milvus

2 JanusGraph

- Use Feathr client to produce and push Use Feathr client to access the embeddings from Milvus

3 OpenLineage

- Adopt OpenLineage for feature metadata

Feathr Roadmap

- Online Transformation
- Streamlined support for realtime/streaming feature
- Feature versioning
- Feature data deletion and retention
- Community building

Summary

- **Feathr** is an open source feature store which can be seen as an abstraction layer between raw data and model.
- **Feathr** allows users to define features with transformation on top of raw data source, and get feature values by feature name during both training and inferencing.
- **Feathr** simplifies feature preparation workflows and enables feature sharing across teams and company.

Q&A

Thank you

(Check out our github: github.com/linkedin/feathr)

LinkedIn

TAC Vote on Project Proposal: Feathr

Proposed Resolution:

The TAC approves the Feathr Project as an incubation project at the sandbox level of the LF AI Foundation

Upcoming TAC Meetings

LF AI & DATA

Upcoming TAC Meetings

- September 8, 2022 – Possible talk from Databricks on Data Lakes
- September 22, 2022 – Starting project reviews

Please note we will be restarting project reviews in the September timeframe. We are always open to special topics as well.

If you have a topic idea or agenda item, please send agenda topic requests to tac-general@lists.lfaidata.foundation

Open Discussion

LF AI & DATA

TAC Meeting Details

- To subscribe to the TAC Group Calendar, visit the wiki: https://wiki.lfaidata.foundation/x/cQB2 _____
- Join from PC, Mac, Linux, iOS or Android: https://zoom.us/j/430697670
- Or iPhone one-tap:
  - US: +16465588656,,430697670# or +16699006833,,430697670#
- Or Telephone:
  - Dial(for higher quality, dial a number based on your current location):
  - US: +1 646 558 8656 or +1 669 900 6833 or +1 855 880 1246 (Toll Free) or +1 877 369 0926 (Toll Free)
- Meeting ID: 430 697 670
- International numbers available: https://zoom.us/u/achYtcw7uN

Legal Notice

- The Linux Foundation, The Linux Foundation logos, and other marks that may be used herein are owned by The Linux Foundation or its affiliated entities, and are subject to The Linux Foundation's Trademark Usage Policy at https://www.linuxfoundation.org/trademark-usage, as may be modified from time to time.
- Linux is a registered trademark of Linus Torvalds. Please see the Linux Mark Institute's trademark usage page at https://lmi.linuxfoundation.org for details regarding use of this trademark.
- Some marks that may be used herein are owned by projects operating as separately incorporated entities managed by The Linux Foundation, and have their own trademarks, policies and usage guidelines.
- TWITTER, TWEET, RETWEET and the Twitter logo are trademarks of Twitter, Inc. or its affiliates.
- Facebook and the "f" logo are trademarks of Facebook or its affiliates.
- LinkedIn, the LinkedIn logo, the IN logo and InMail are registered trademarks or trademarks of LinkedIn Corporation and its affiliates in the United States and/or other countries.
- YouTube and the YouTube icon are trademarks of YouTube or its affiliates.
- All other trademarks are the property of their respective owners. Use of such marks herein does not represent affiliation with or authorization, sponsorship or approval by such owners unless otherwise expressly specified.
- The Linux Foundation is subject to other policies, including without limitation its Privacy Policy at https://www.linuxfoundation.org/privacy and its Antitrust Policy at https://www.linuxfoundation.org/antitrust-policy. each as may be modified from time to time. More information about The Linux Foundation's policies is available at https://www.linuxfoundation.org.
- Please email legal@linuxfoundation.org with any questions about The Linux Foundation's policies or the notices set forth on this slide.

LF AI & DATA