DATA+AI SUMMIT 2022

Building and Scaling Machine Learning-Based Products in the World's Largest Brewery

Dr Renata Castanha
Technical Product Manager
Anheuser-Busch InBev (Brazil)

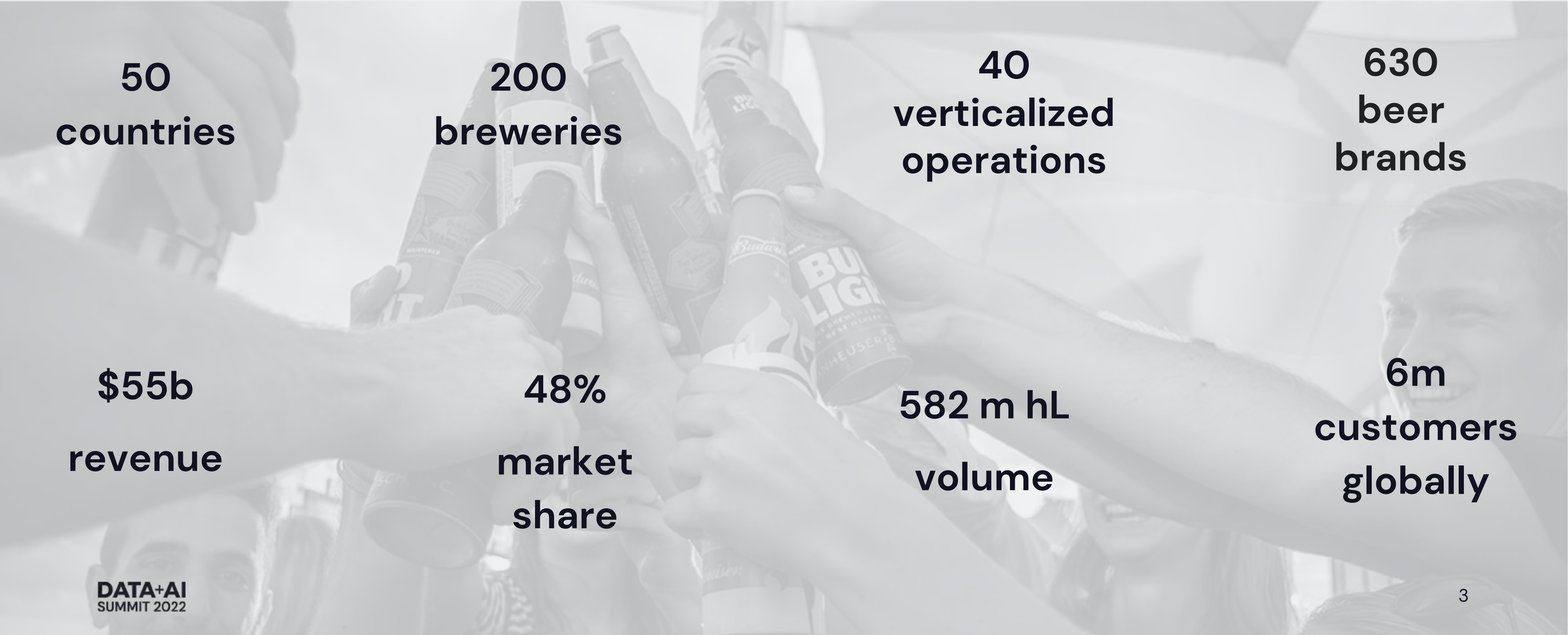
ORGANIZED BY databricks

AGENDA

- ABI
- Previous state + paradigm shift
- Data Platform Products
- How to build a model
- Next steps and Lessons learned

Anheuser-Busch InBev

World's largest brewery



50 countries

200 breweries

40 verticalized operations

630 beer brands

$55b revenue

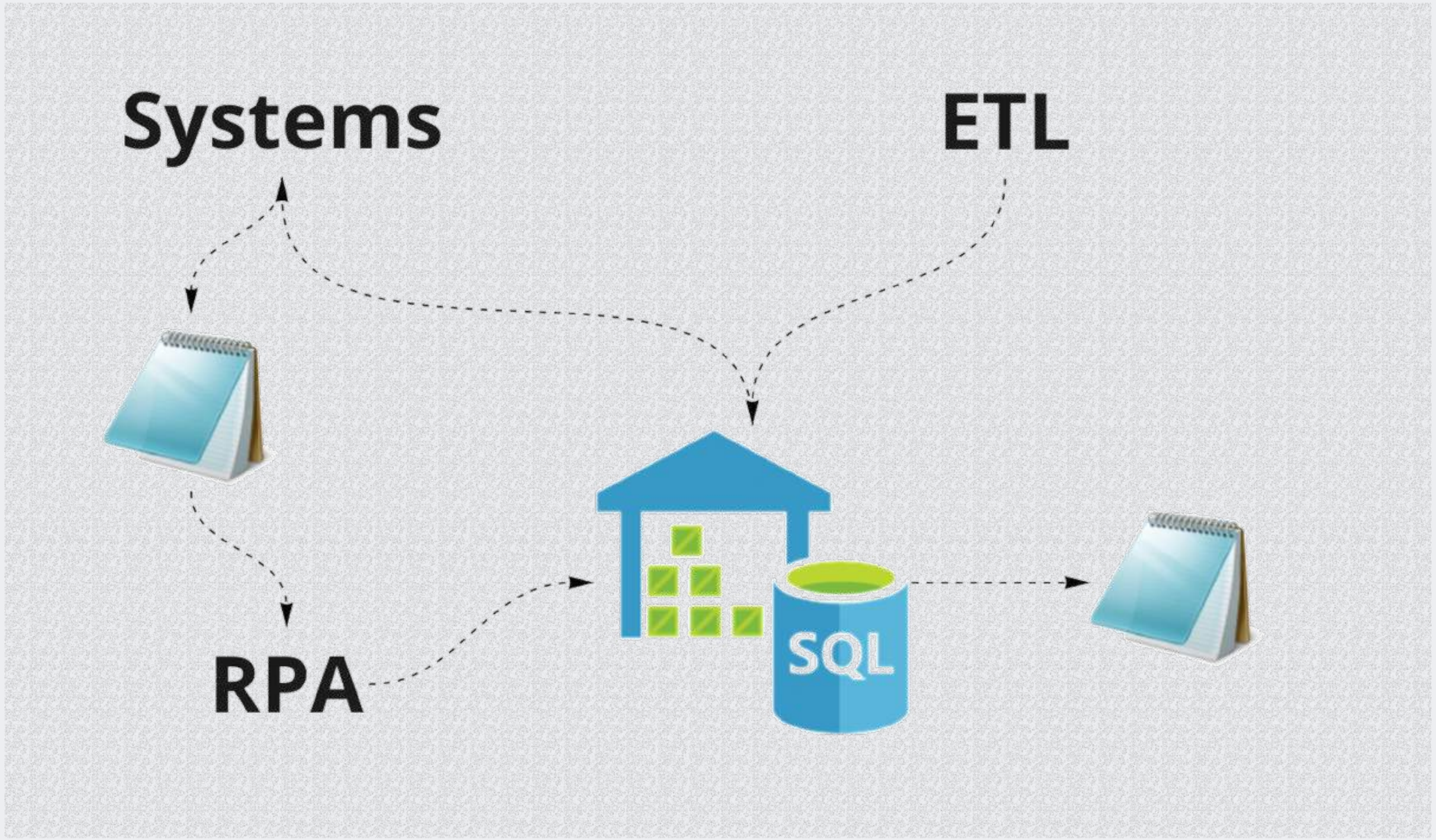
48% market share

582 m hL volume

6m customers globally

Previous State

Legacy Architecture



Problems to be solved

Technical gaps

- Governance
- Sustainability
- Centralization attempt

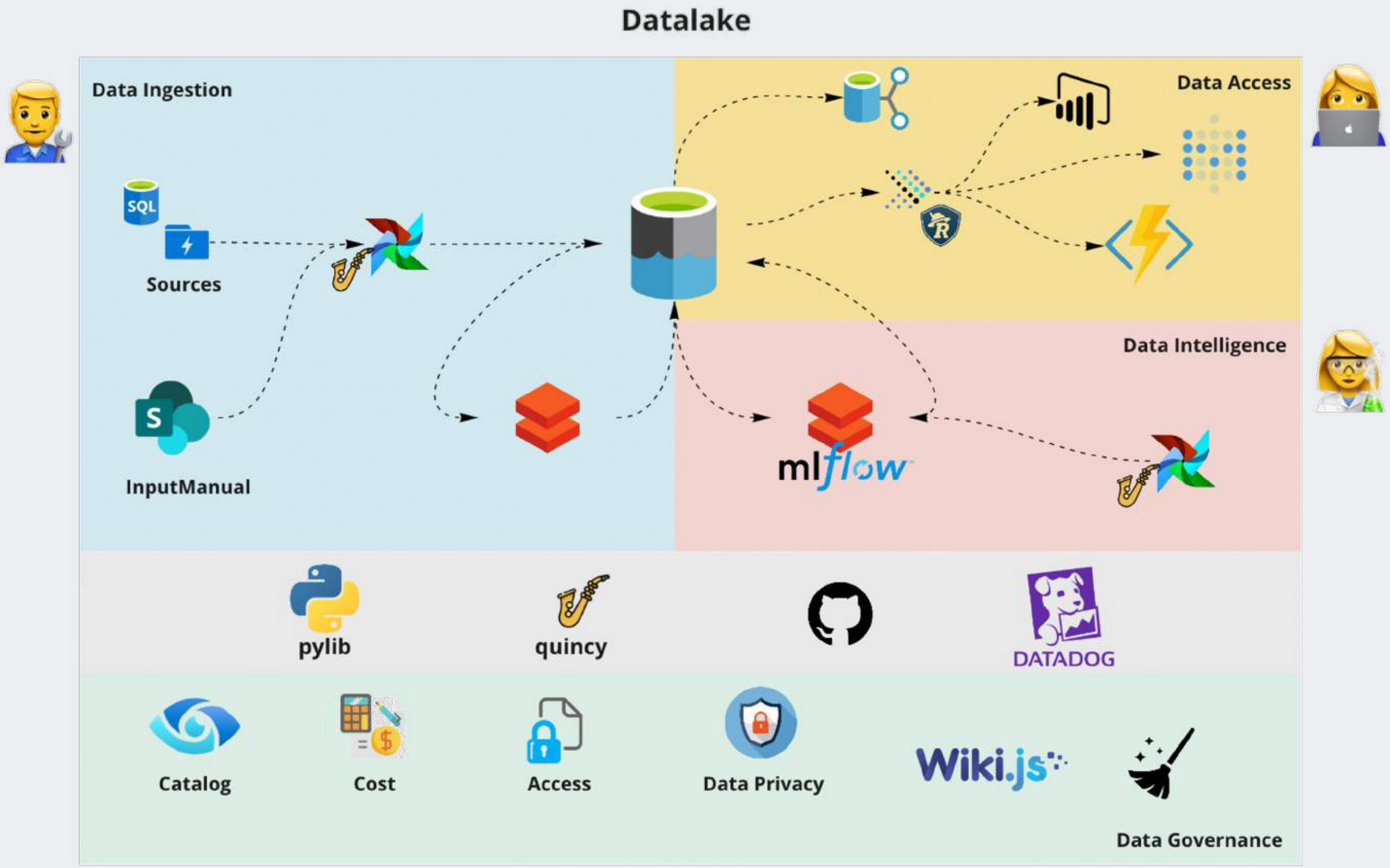
PARADIGM SHIFT

Data Platform

Cutting-edge architecture designed with the following principles:

- Deliver **value to the users** in a consistent and automated manner
- **Reproducibility**, so algorithms are easy to maintain, in a single, collaborative ecosystem
- **Reduce technical debt**, so data scientists are more concerned with solving the business problem than with deploying and maintaining infrastructure
- **Tech product vision**

Data Platform



Standing out

Collaborative and centralized library

Code duplication

Read, write

Quincy

Airflow abstraction

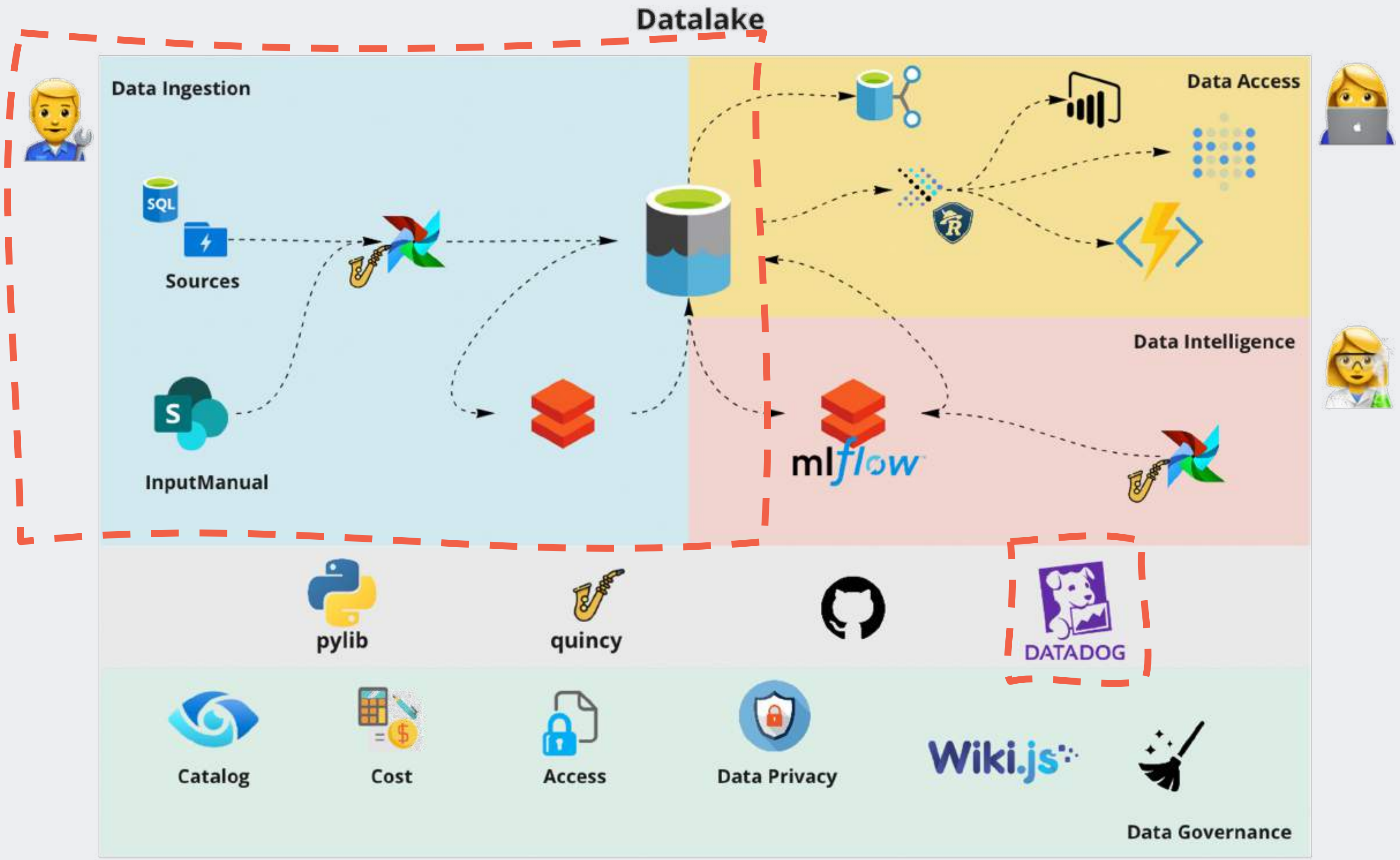
YAML files to DAGs

ETL and batch models

Fast, accessible and reliable architecture

DATA INGESTION PLATFORM

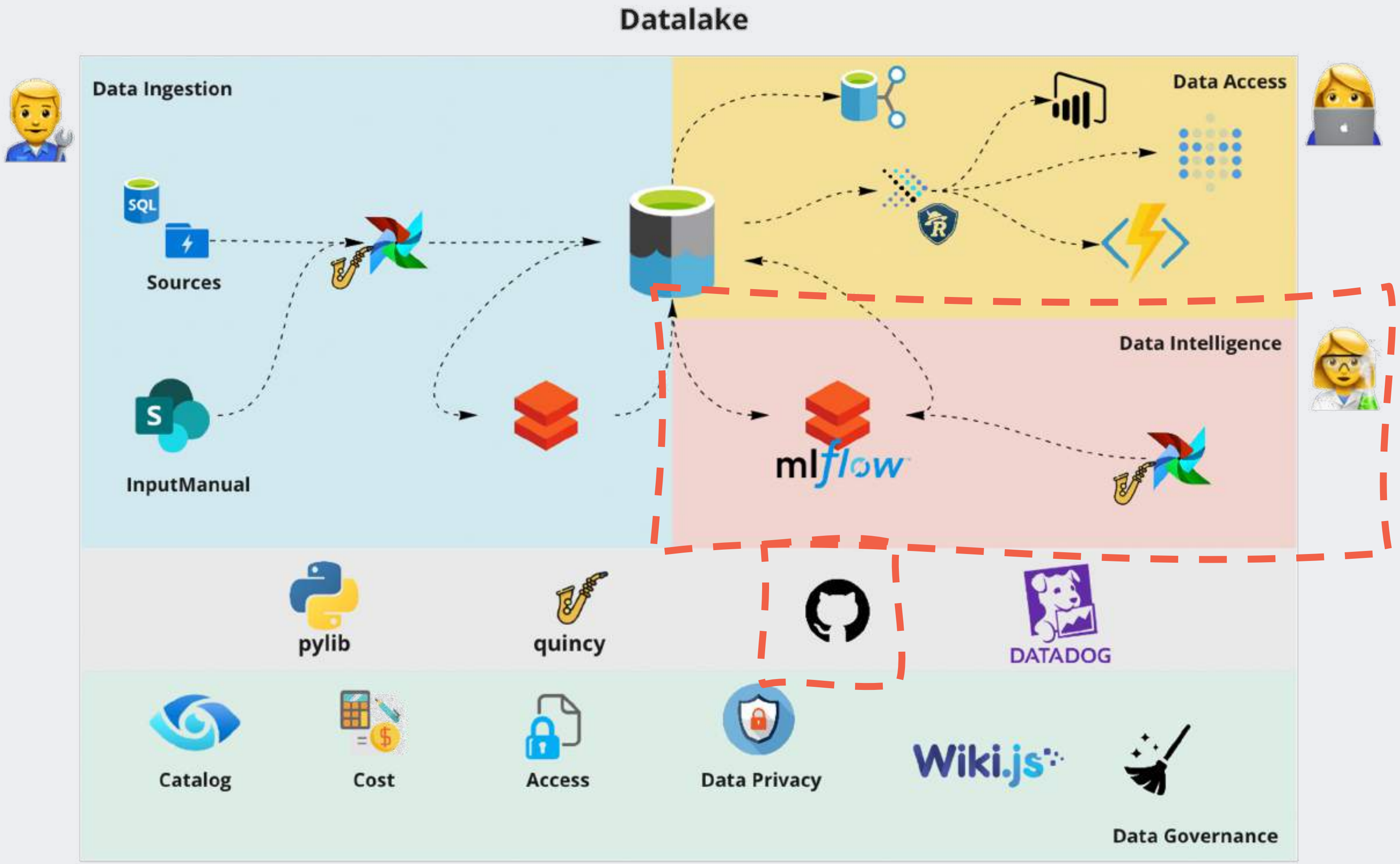
Easy data for all



- ~ 400 users
- ~ 5k deploys
- ~ 300 Tb

DATA INTELLIGENCE PLATFORM

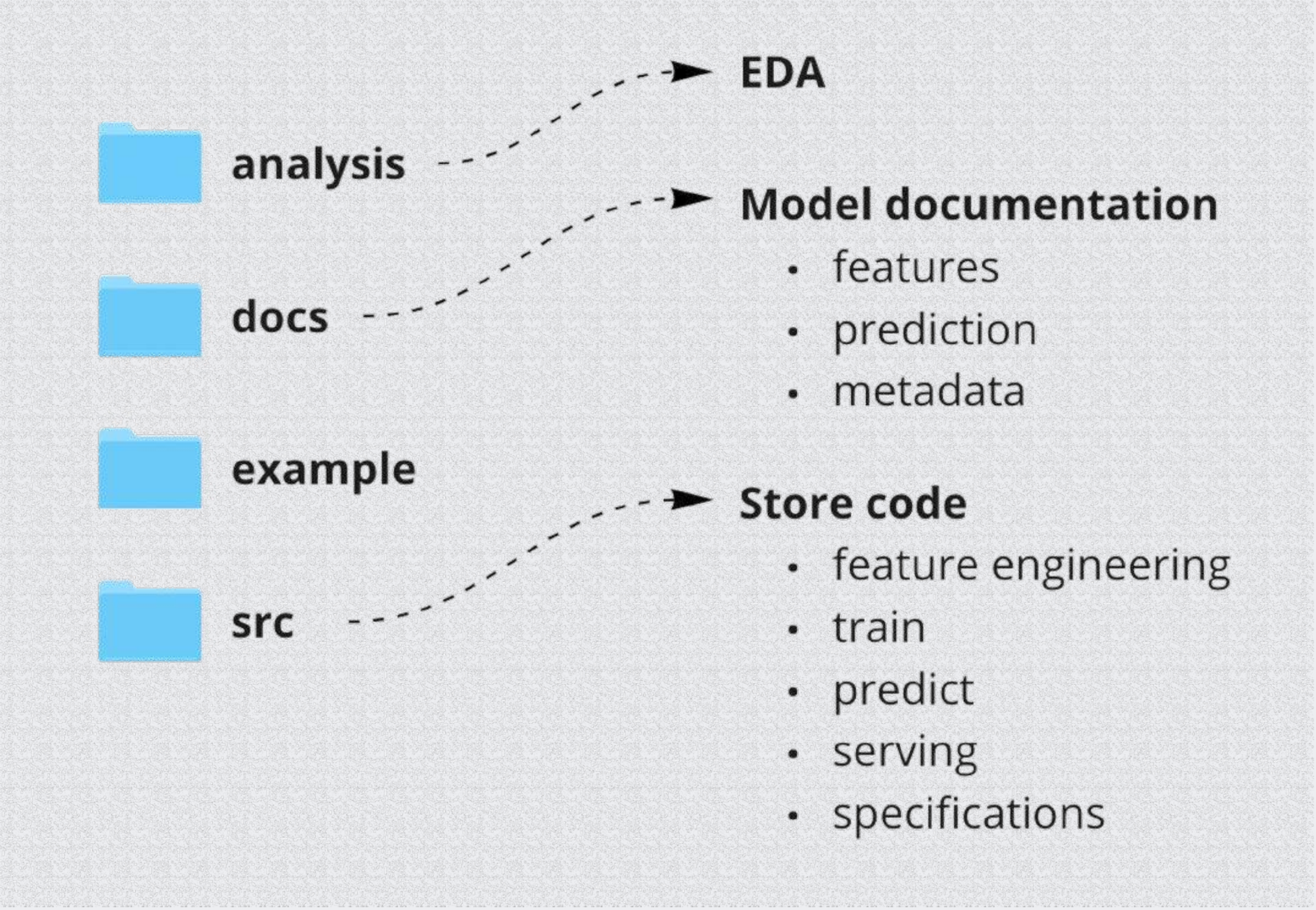
Empower users on DS/ML tools and techniques



- > 150 direct users
- ~ 10 ML prod
- ~ 50 ML dev

DATA INTELLIGENCE PLATFORM

Data Science Template



analysis → EDA

docs → Model documentation

- features
- prediction
- metadata

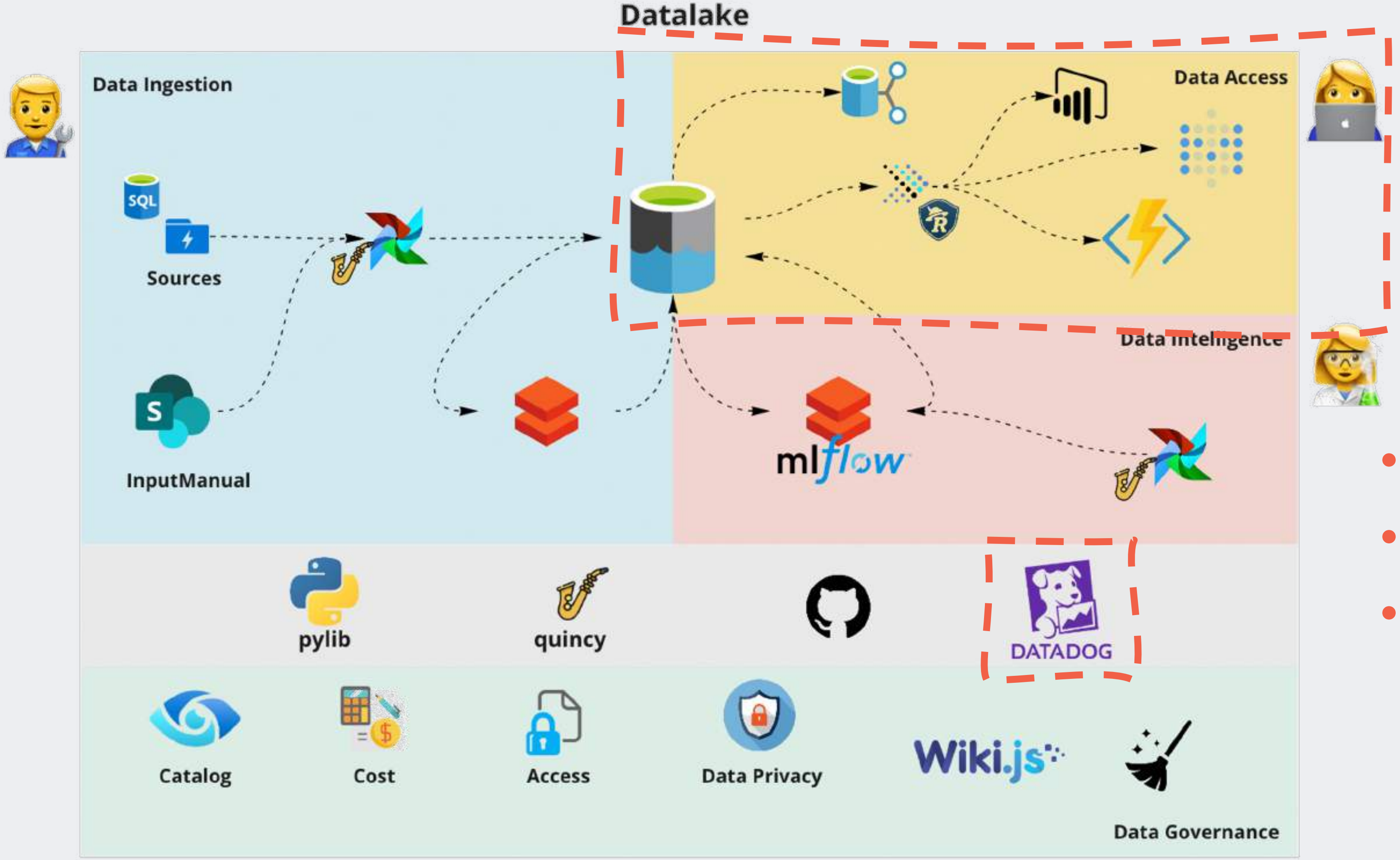
example

src → Store code

- feature engineering
- train
- predict
- serving
- specifications

DATA ACCESS PLATFORM

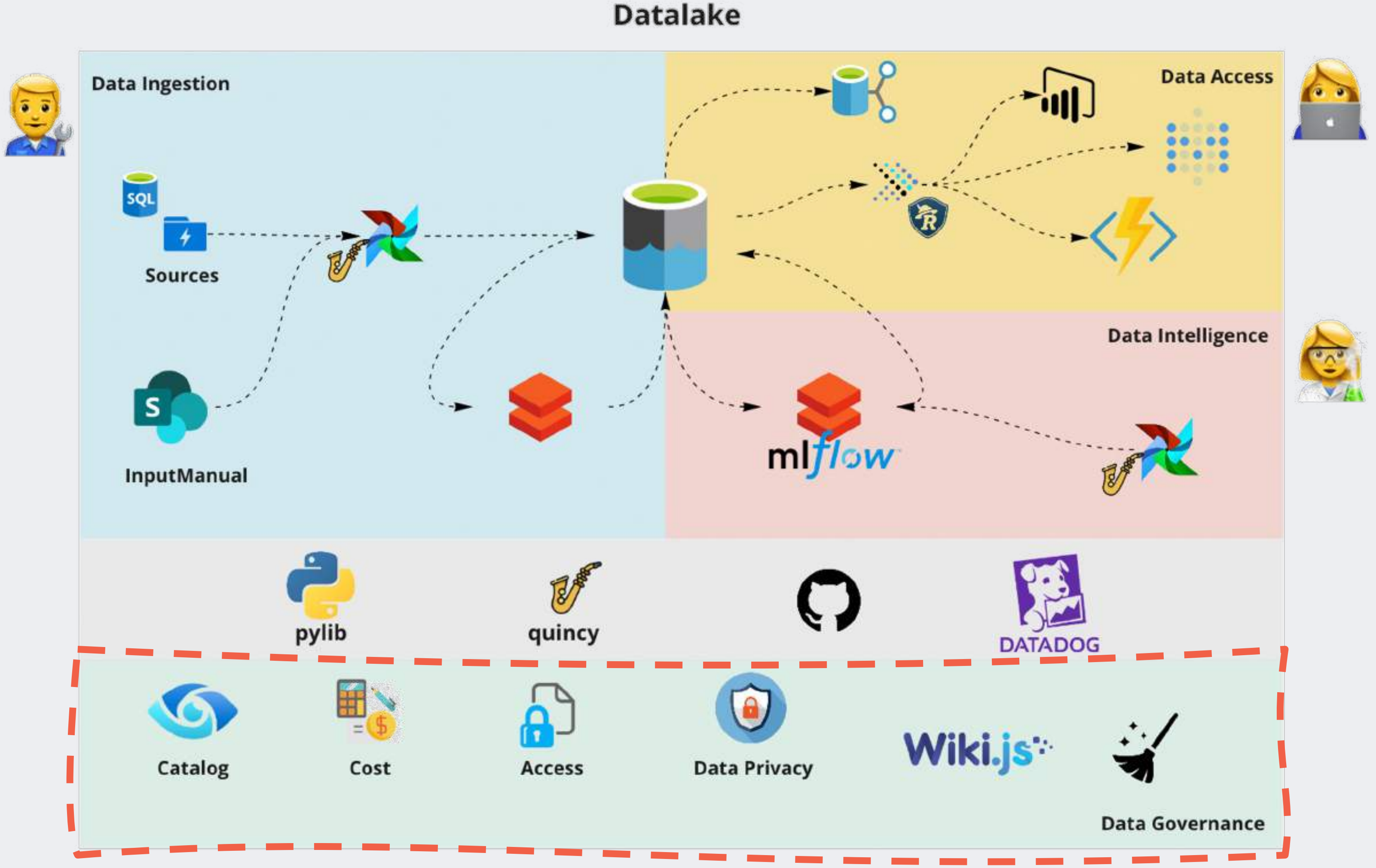
Democratize access to information



- ~ 170 users
- > 10k queries
- ~ 1 mi API requests/week

DATA GOVERNANCE LAYER

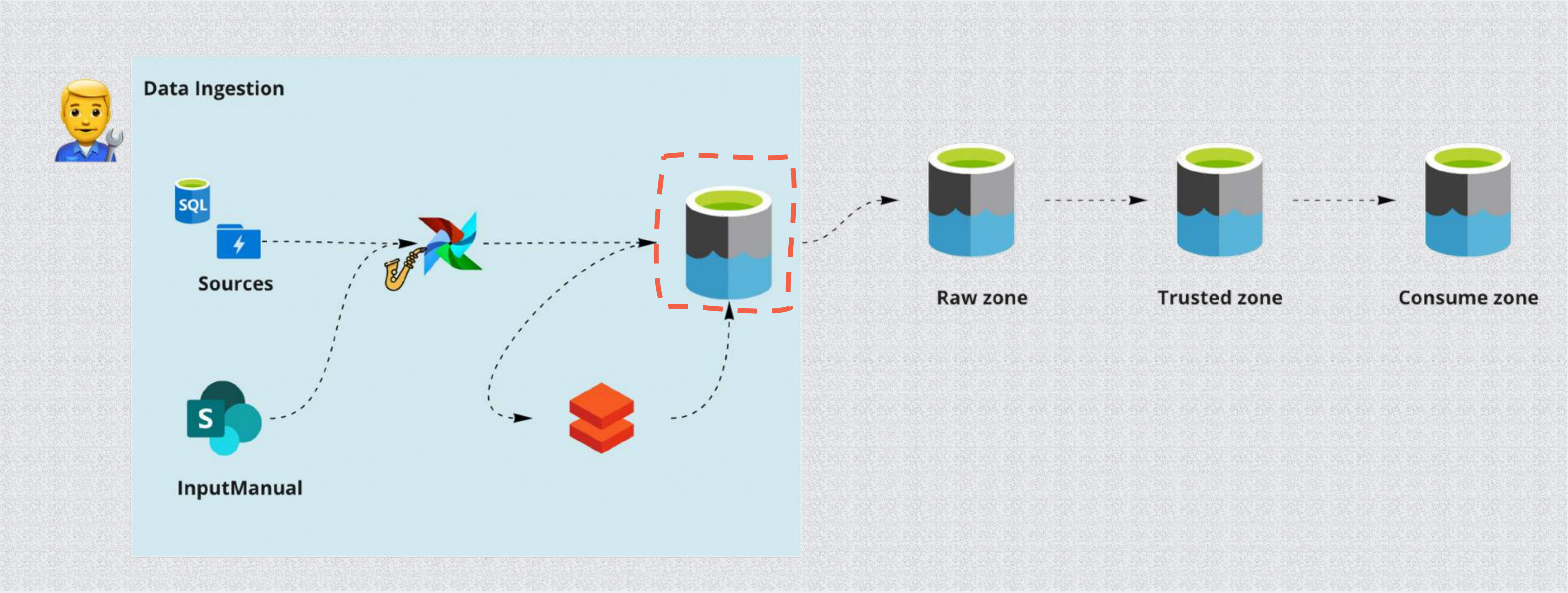
Making sure we are sustainable and compliant



How to build a model (e2e) using the platform?

Data Ingestion

Worry about the data, not the process



Creating your DAG

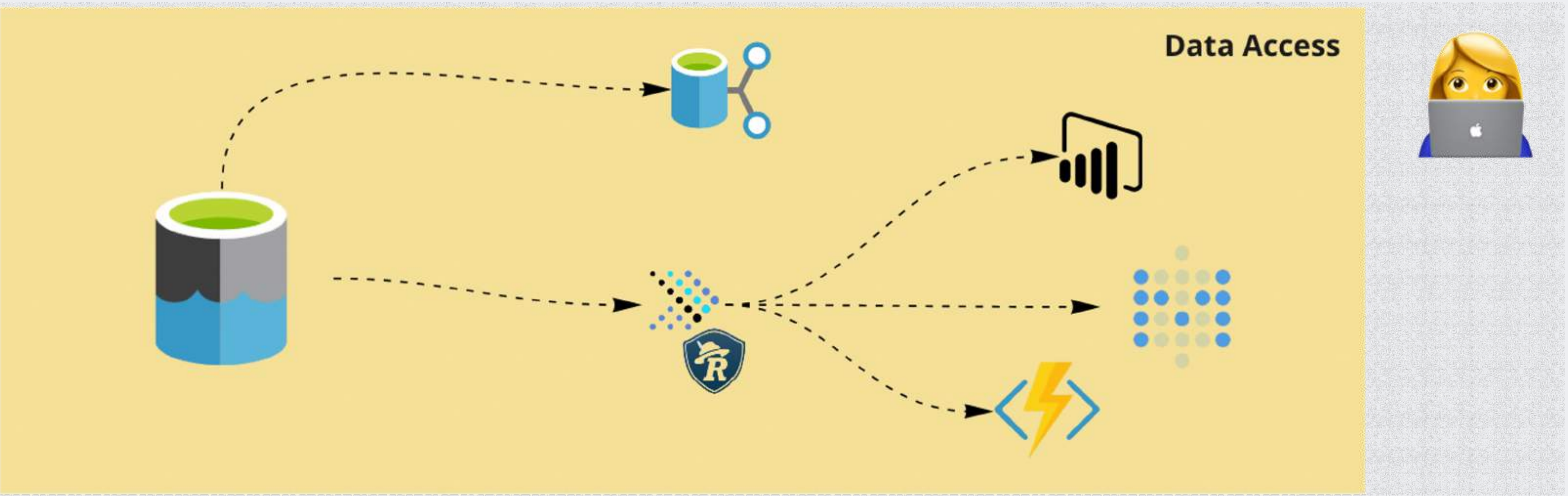
Worry about the data, not the process

```
dag:
    dag_id: 123456
    dag_class: “source”
    dag_type: “connector”
    schedule_interval: “@hourly”
    system: “payments”
    country: “Brazil”

datasets:
  - name: “client_payment”
    active: True
    domain: “clients”
    entity: “entity”
    task_owner: “Renata C”
    start_date: # datetime(YYYY,MM,DD)
    connection_id: “client_payment_id”
    metadata: metadata.json
```

+ Data pre-processing

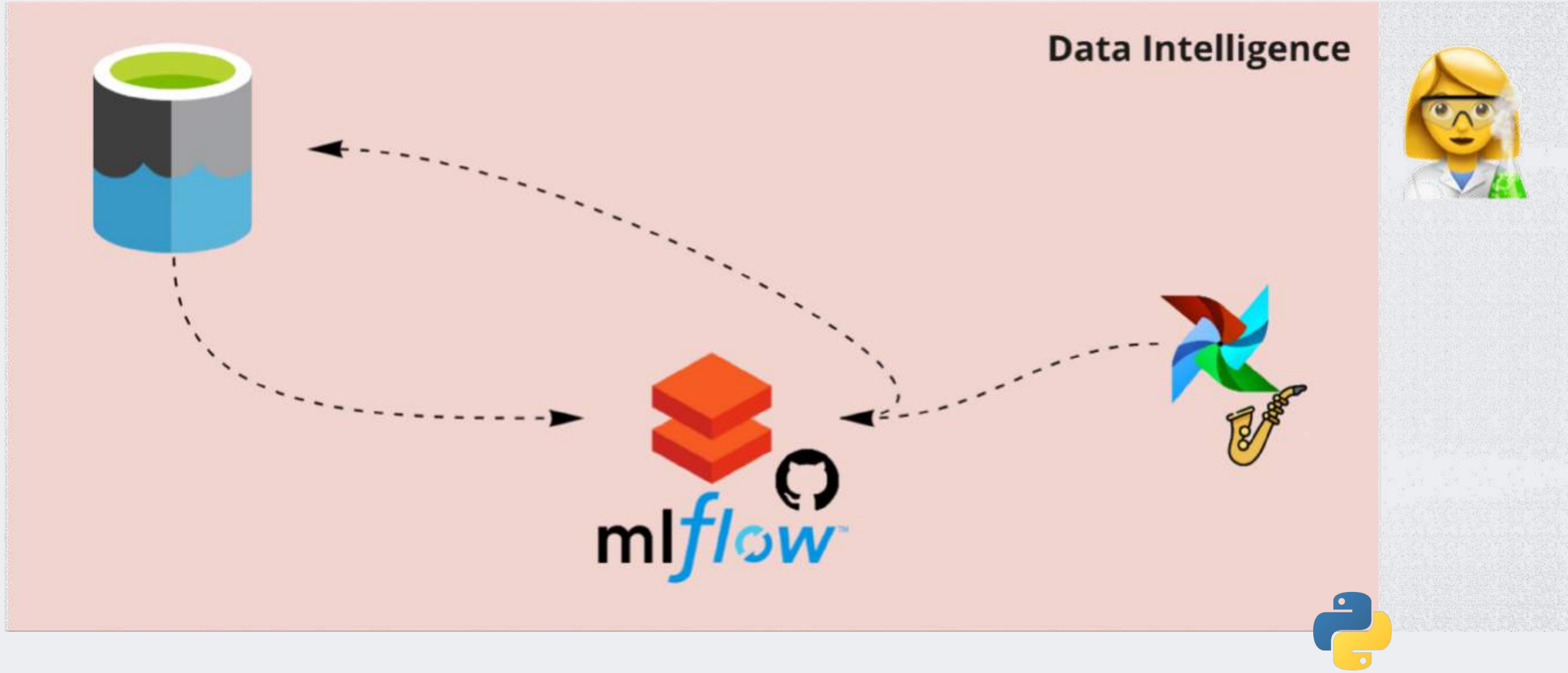
Good to go!

Show your results or build your model



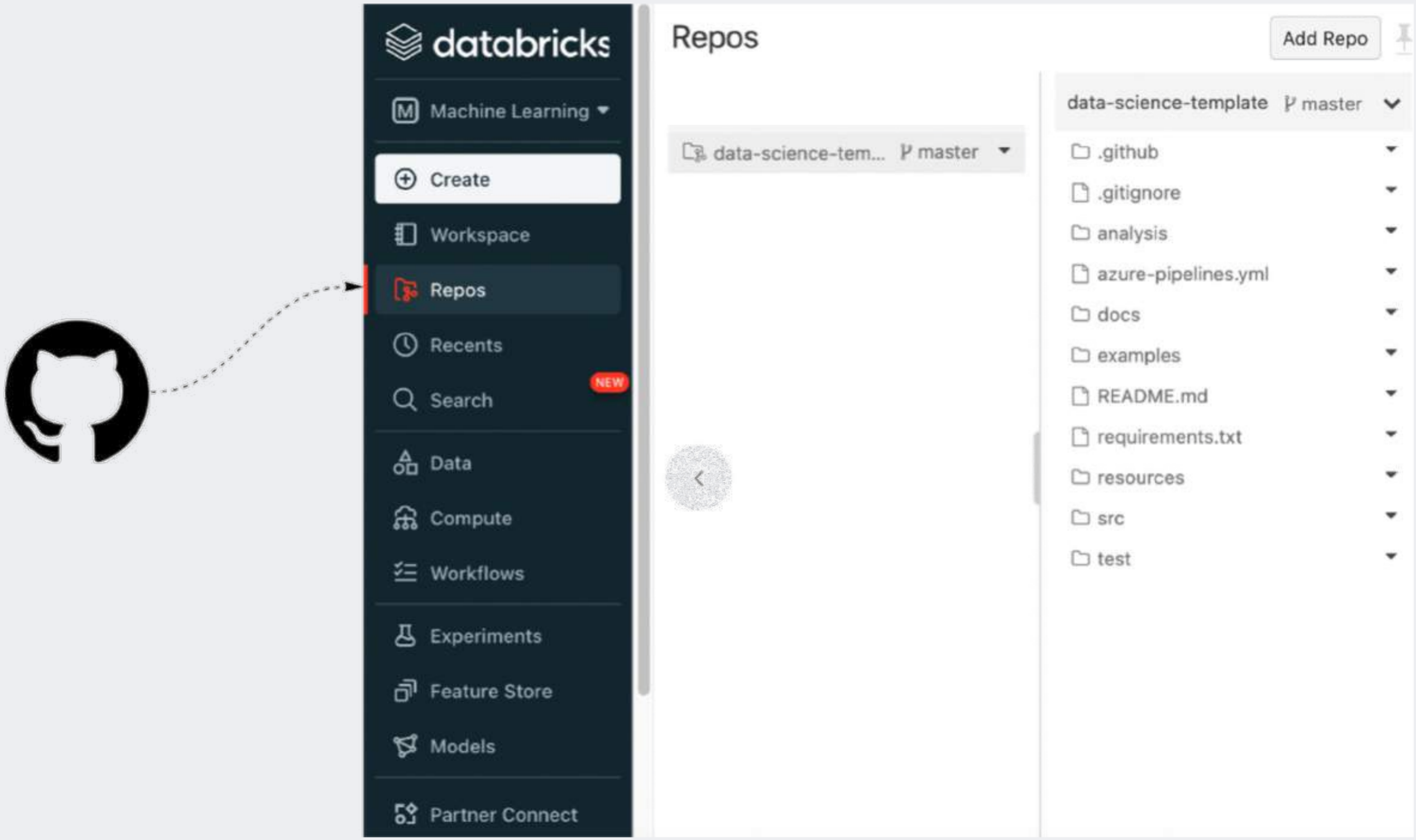
Good to go!

Worry about the model, not the infra



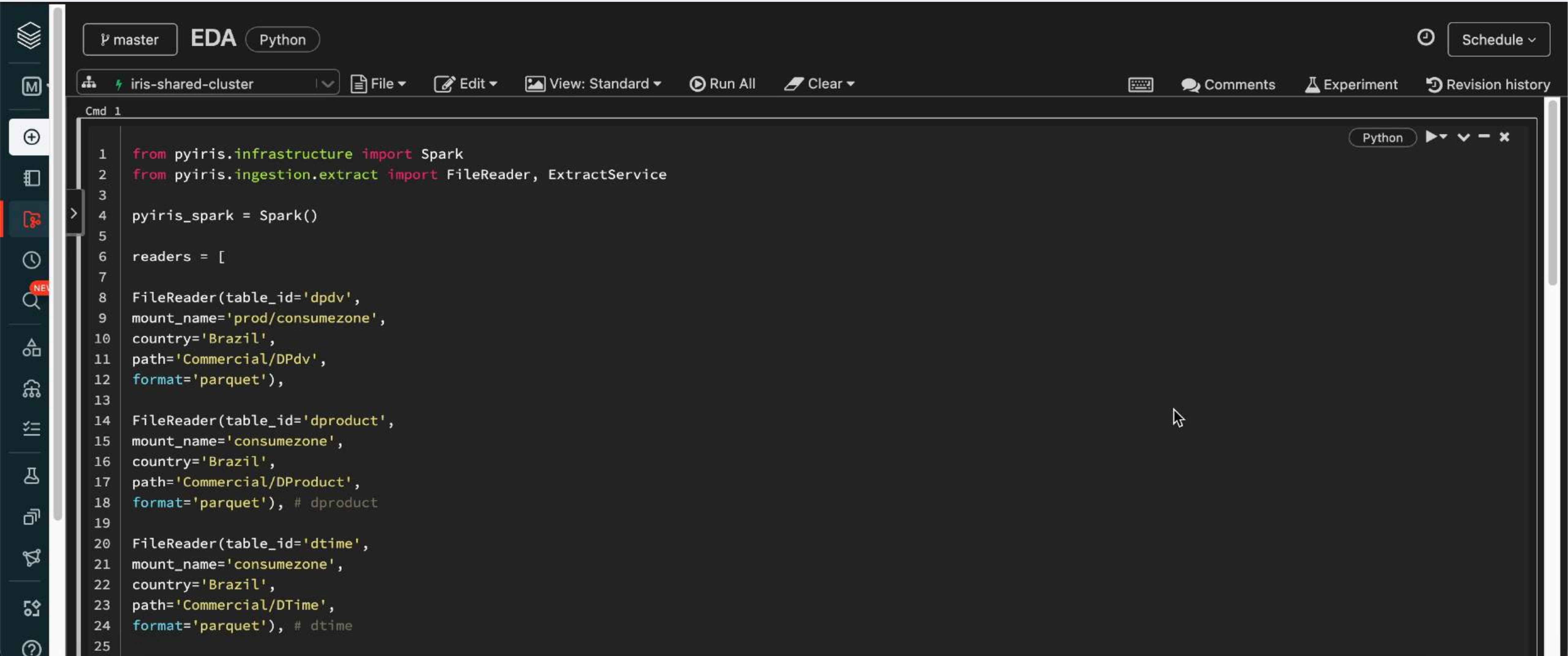
Accessing the DS template

Worry about the model, not the infra



Data & business understanding

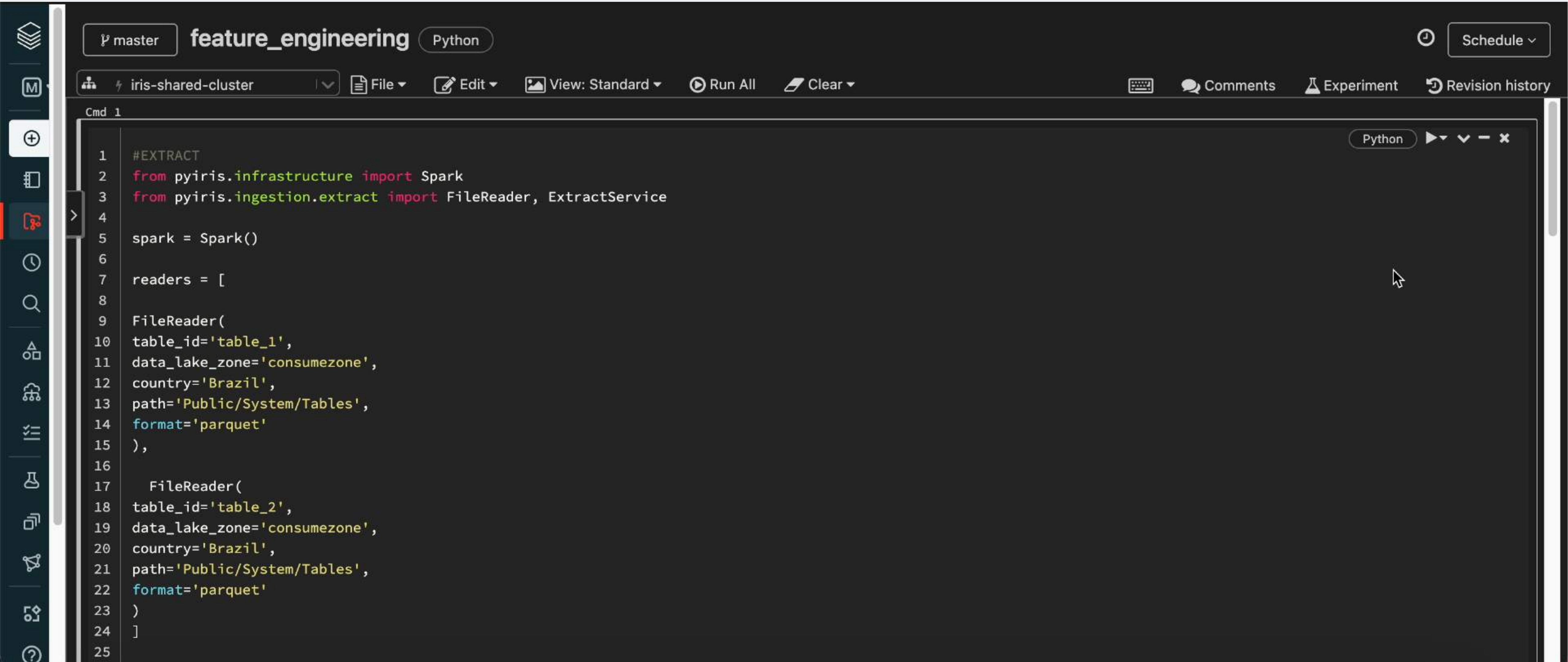
How to read and write data with our library



```python
from pyiris.infrastructure import Spark
from pyiris.ingestion.extract import FileReader, ExtractService

pyiris_spark = Spark()

readers = [

FileReader(table_id='dpdv',
mount_name='prod/consumezone',
country='Brazil',
path='Commercial/DPdv',
format='parquet'),

FileReader(table_id='dproduct',
mount_name='consumezone',
country='Brazil',
path='Commercial/DProduct',
format='parquet'), # dproduct

FileReader(table_id='dtime',
mount_name='consumezone',
country='Brazil',
path='Commercial/DTime',
format='parquet'), # dtime

```

Going ML

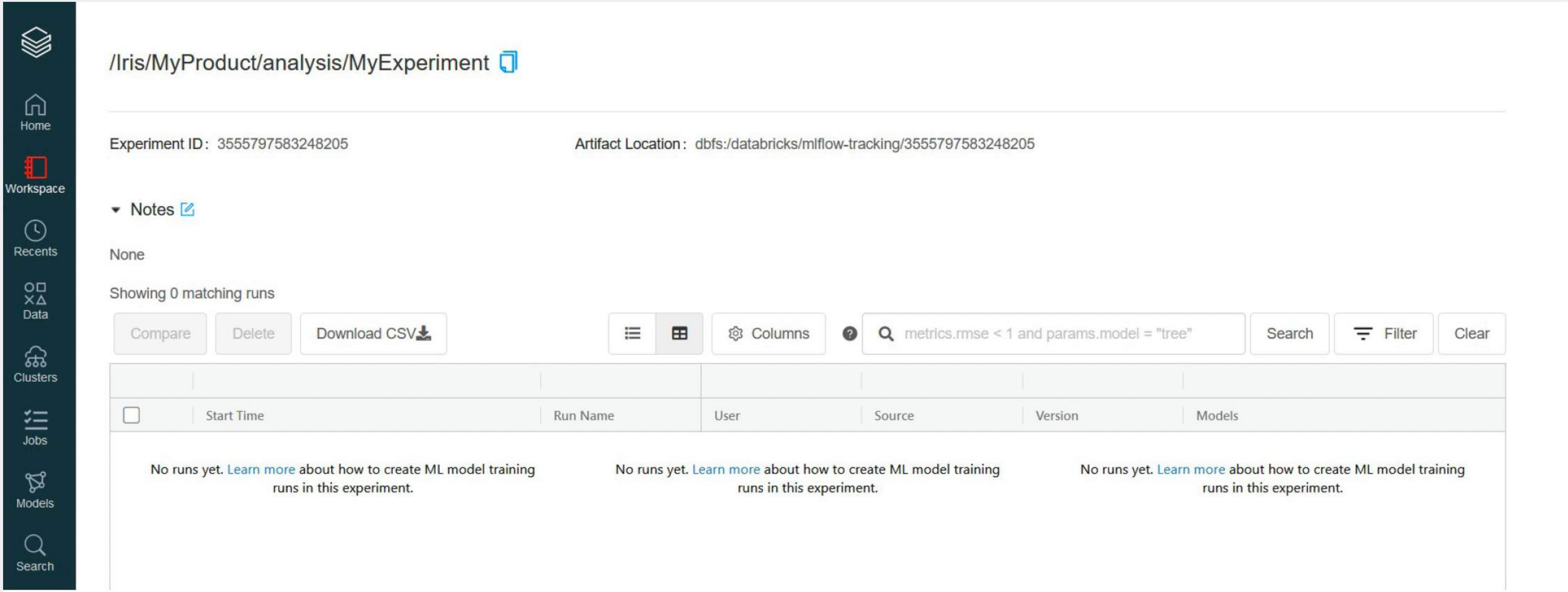
Feature Engineering



```python
#EXTRACT
from pyiris.infrastructure import Spark
from pyiris.ingestion.extract import FileReader, ExtractService

spark = Spark()

readers = [

FileReader(
table_id='table_1',
data_lake_zone='consumezone',
country='Brazil',
path='Public/System/Tables',
format='parquet'
),

  FileReader(
table_id='table_2',
data_lake_zone='consumezone',
country='Brazil',
path='Public/System/Tables',
format='parquet'
)
]

```

Going ML

Creating experiment



/Iris/MyProduct/analysis/MyExperiment

Experiment ID: 3555797583248205

Artifact Location: dbfs:/databricks/mlflow-tracking/3555797583248205

Notes

None

Showing 0 matching runs

□	Start Time	Run Name	User	Source	Version	Models

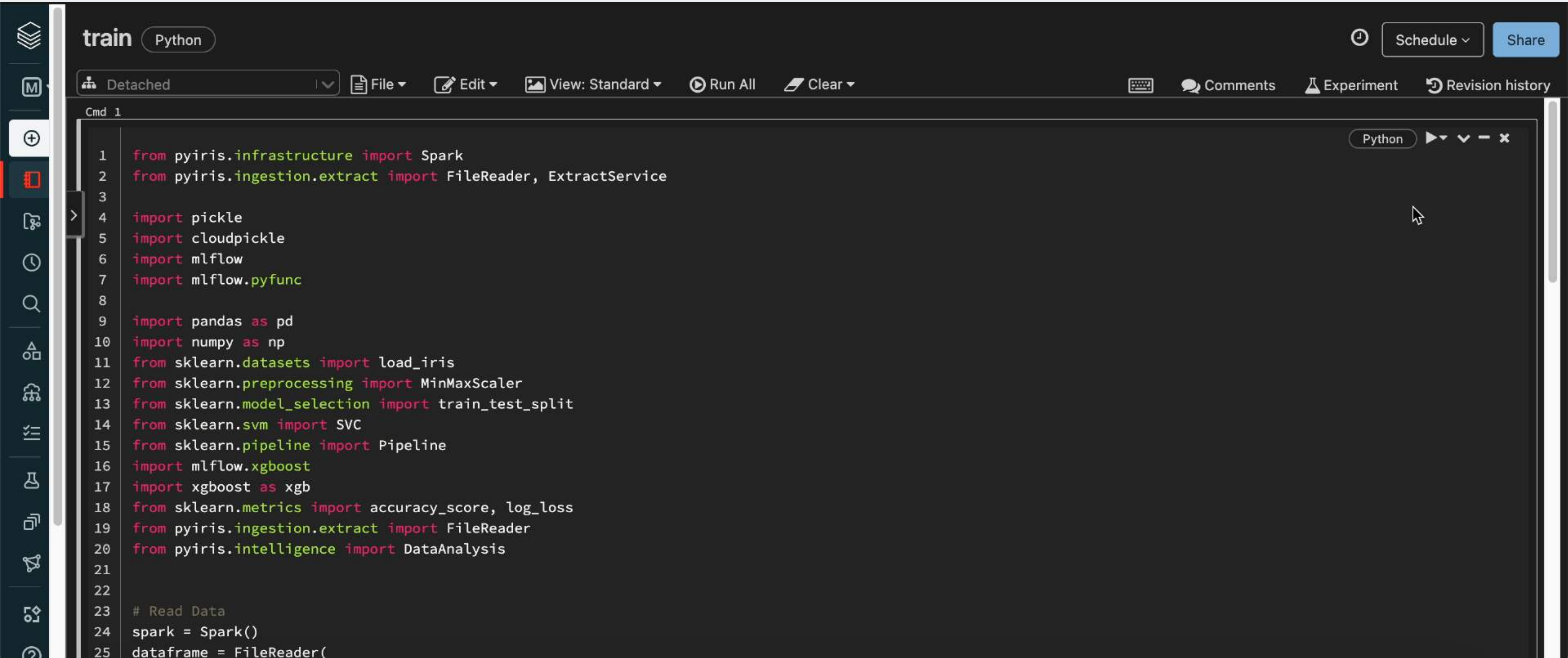
No runs yet. Learn more about how to create ML model training runs in this experiment.

No runs yet. Learn more about how to create ML model training runs in this experiment.

No runs yet. Learn more about how to create ML model training runs in this experiment.

Going ML

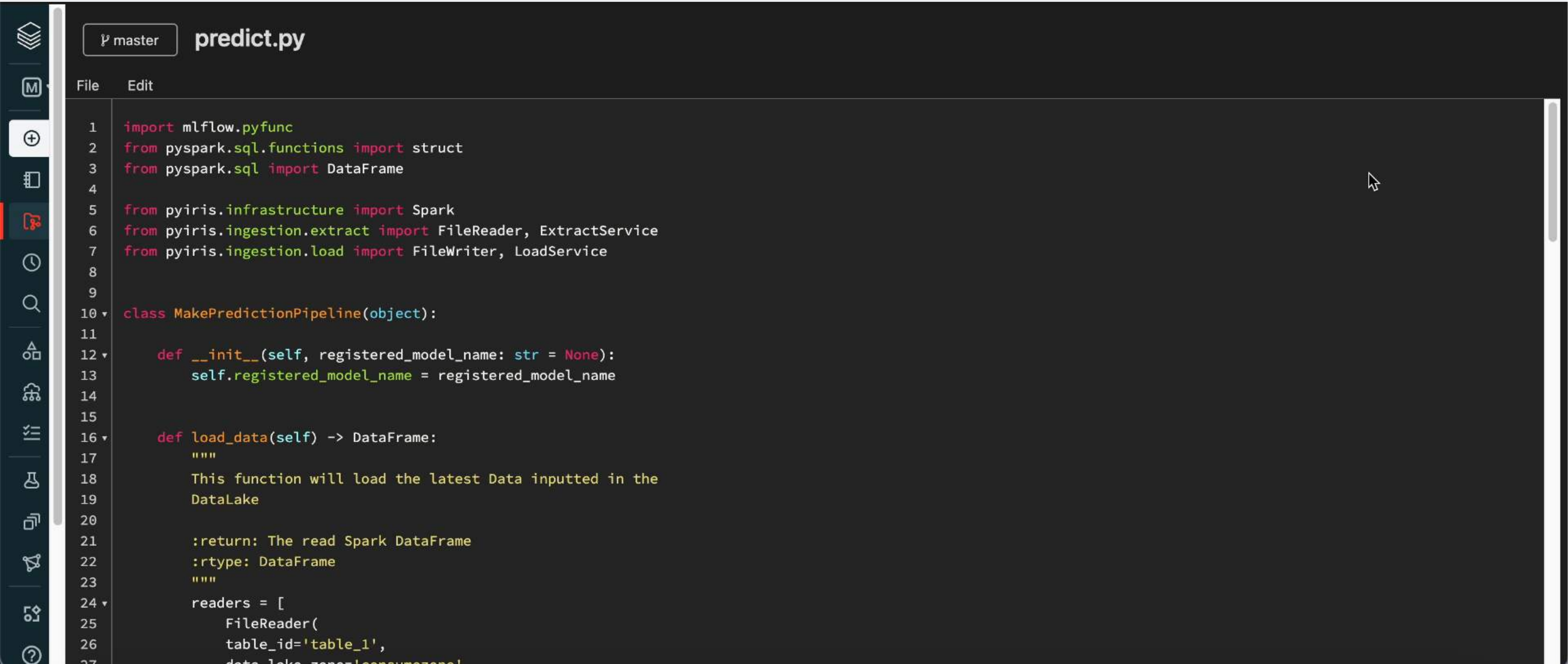
Model training



```python
from pyiris.infrastructure import Spark
from pyiris.ingestion.extract import FileReader, ExtractService

import pickle
import cloudpickle
import mlflow
import mlflow.pyfunc

import pandas as pd
import numpy as np
from sklearn.datasets import load_iris
from sklearn.preprocessing import MinMaxScaler
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.pipeline import Pipeline
import mlflow.xgboost
import xgboost as xgb
from sklearn.metrics import accuracy_score, log_loss
from pyiris.ingestion.extract import FileReader
from pyiris.intelligence import DataAnalysis


# Read Data
spark = Spark()
dataframe = FileReader(
```

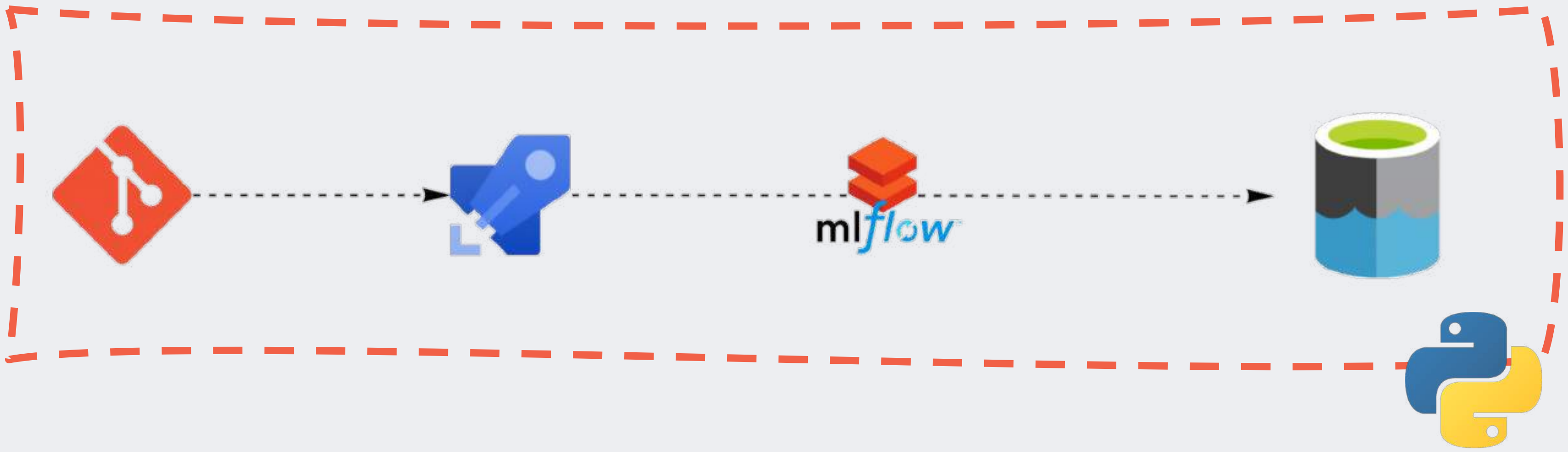
Going ML

Predict



```python
import mlflow.pyfunc
from pyspark.sql.functions import struct
from pyspark.sql import DataFrame

from pyiris.infrastructure import Spark
from pyiris.ingestion.extract import FileReader, ExtractService
from pyiris.ingestion.load import FileWriter, LoadService


class MakePredictionPipeline(object):

    def __init__(self, registered_model_name: str = None):
        self.registered_model_name = registered_model_name


    def load_data(self) -> DataFrame:
        """
        This function will load the latest Data inputted in the
        DataLake

        :return: The read Spark DataFrame
        :rtype: DataFrame
        """
        readers = [
            FileReader(
            table_id='table_1',
```

Deploying – DS Template (core)



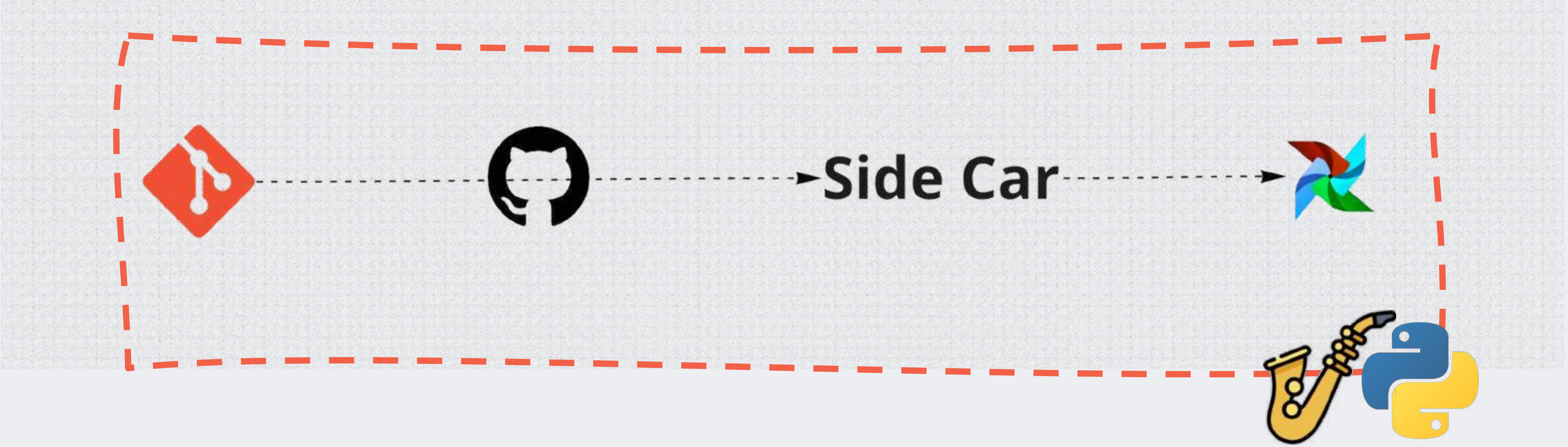
Scheduling your task

Creating your DAG

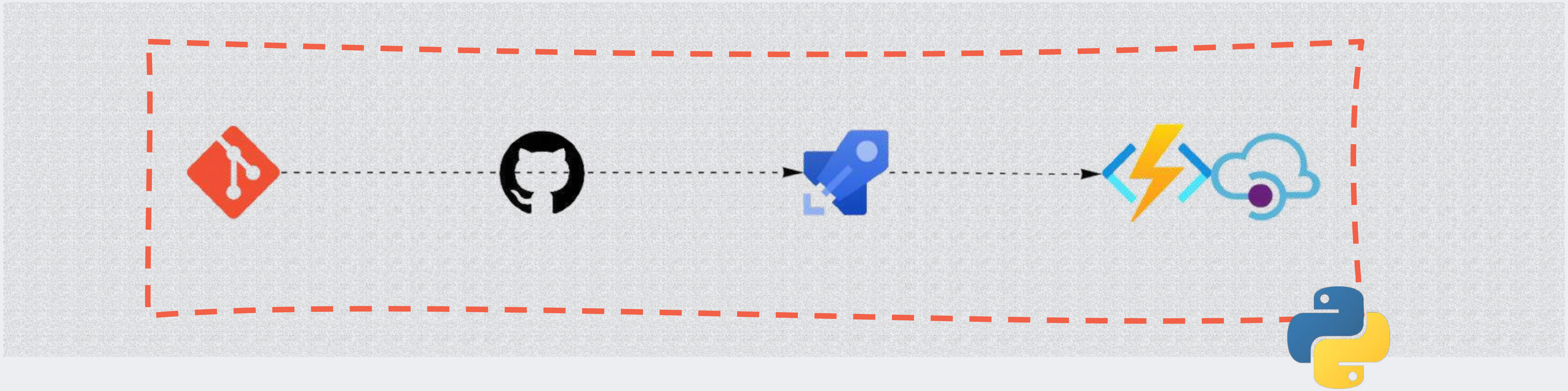
```
dag:
dag_id: ”clients_churn"
dag_type: “predict”
dag_class: "analytical"
country: "Brazil"
context: ”Commercial"
domain: ”Clients"
owner: ”Renata C"
schedule_interval: # @daily, @hourly, @weekly or cron syntax
start_date: # datetime(YYYY,MM,DD)
product_location: ”Commercial/Clients/Products"
```

```
tasks:
- module: "main"
num_workers: "1"
cluster: "Standard_DS3_v2"
libraries:
   - cloudpickle==1.3.0.
   - pyarrow==4.0.1
```

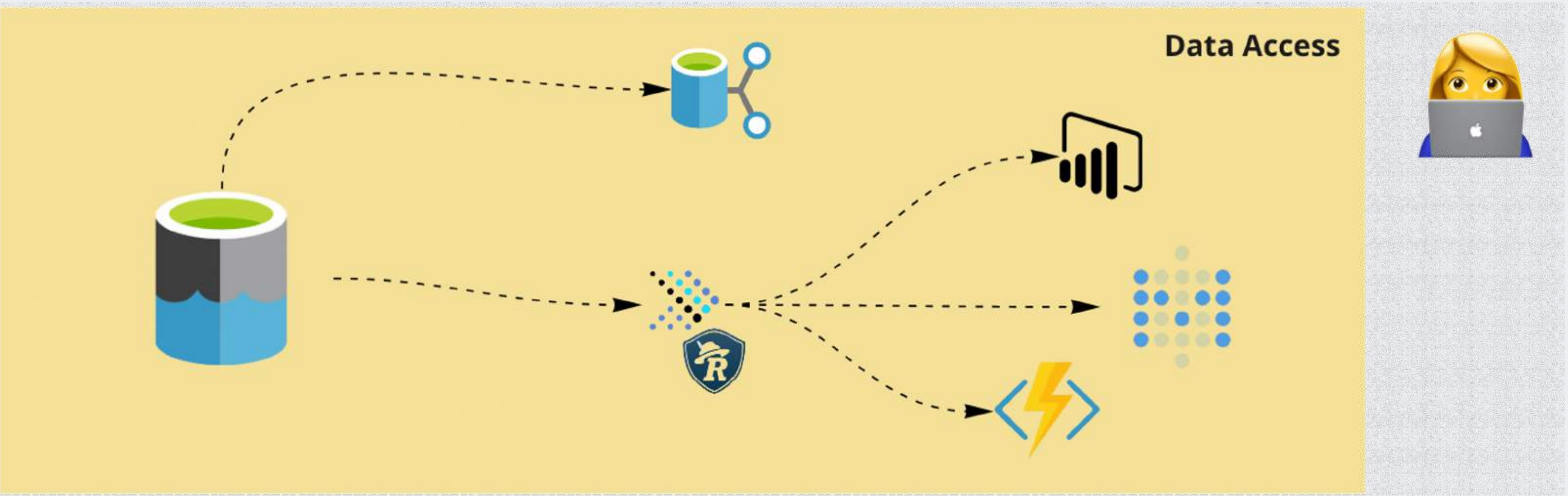
Deploying – Batch (Quincy)



Deploying – API



Enjoy the actionable insights



Next steps

- Delta implementation

- API abstracting

- Metrics platform

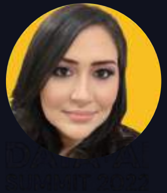
Lessons learned

- Governance since day 1
- Don’t productionize kludge – specially without documentation (data swamp)
- Support tools for scalable growth

The amazing team!

DATA+AI
SUMMIT 2022

Thank you



renata.castanha@ambev.com.br

in renatacgcastanha