databricks

00 - Introduction

(https://databricks.com)

Let's start with a business problem:

Building a Customer 360 database and reducing customer churn with the Databricks Lakehouse

In this demo, we'll step in the shoes of a retail company selling goods with a recurring business.

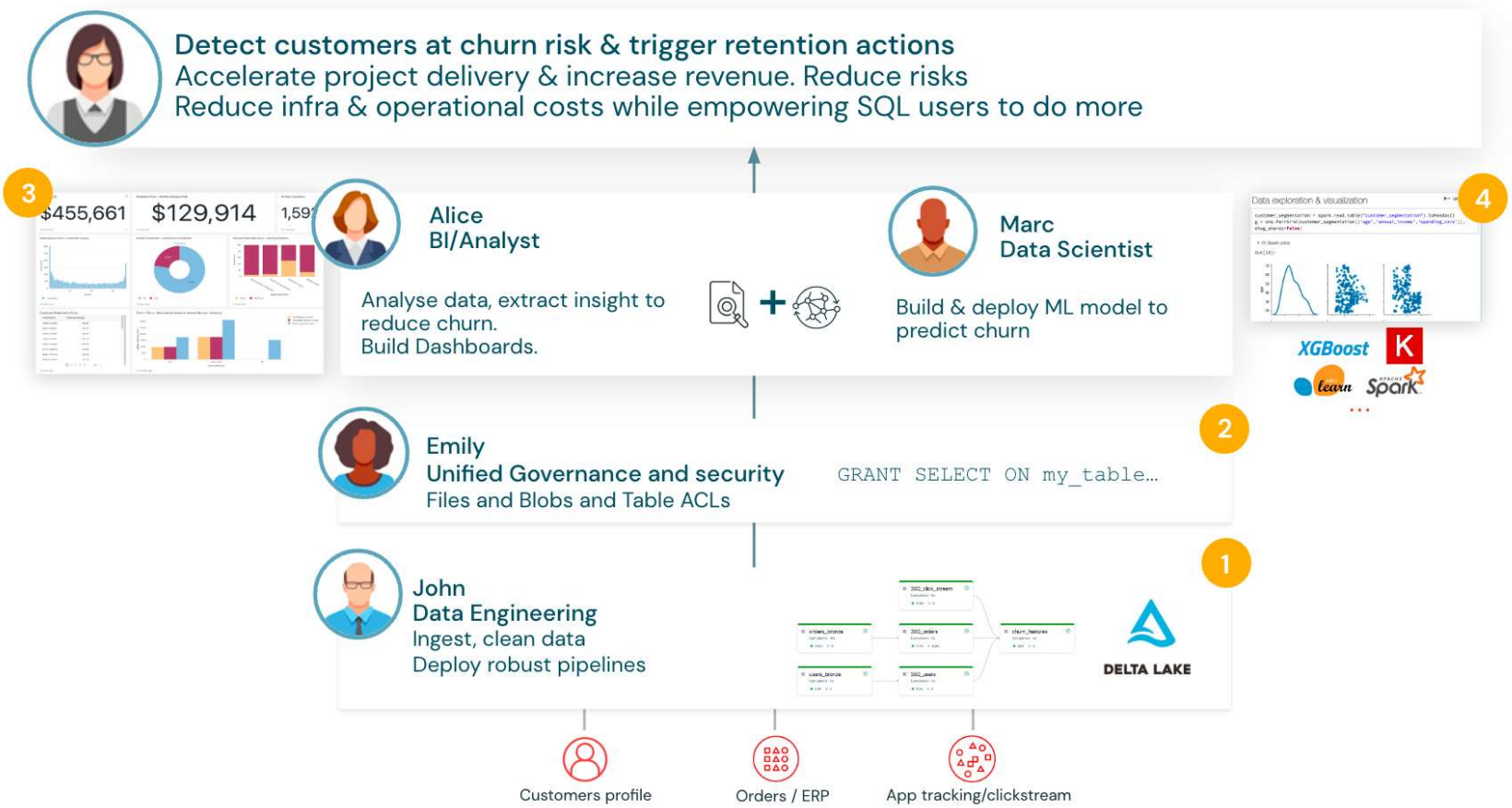
The business has determined that the focus must be placed on churn. We're asked to:

- Analyse and explain current customer churn: quantify churn, trends and the impact for the business
- Build a proactive system to forecast and reduce churn by taking automated action: targeted email, phoning etc.

What we'll build

To do so, we'll build an end-to-end solution with the Lakehouse. To be able to properly analyse and predict our customer churn, we need information coming from different external systems: Customer profiles coming from our website, order details from our ERP system and mobile application clickstream to analyse our customers activity.

At a very high level, this is the flow we'll implement:



1. Ingest and create our Customer 360 database, with tables easy to query in SQL
2. Secure data and grant read access to the Data Analyst and Data Science teams.
3. Run BI queries to analyse existing churn
4. Build ML model to predict which customer is going to churn and why

As a result, we will have all the information required to trigger custom actions to increase retention (email personalized, special offers, phone call...)

Raw data generation

For this demonstration we will not be using real data or an existing dataset, but will rather generate them.

The cell below will execute a notebook that will generate the data and store them on DBFS. If you want to see the actual code click here to open it on a different tab ($./includes/CreateRawData)

```
%run ./includes/CreateRawData
```

```
Python interpreter will be restarted.
Requirement already satisfied: Faker in /local_disk0/.ephemeral_nfs/envs/pythonEnv-6265a51b-ede1-4a6b-a2b5-75e2ffaddd18/lib/python3.9/site-pa
ckages (23.2.1)
Requirement already satisfied: python-dateutil>=2.4 in /databricks/python3/lib/python3.9/site-packages (from Faker) (2.8.2)
Requirement already satisfied: six>=1.5 in /databricks/python3/lib/python3.9/site-packages (from python-dateutil>=2.4->Faker) (1.16.0)
Python interpreter will be restarted.

Raw data already exists
```

The raw data on DBFS

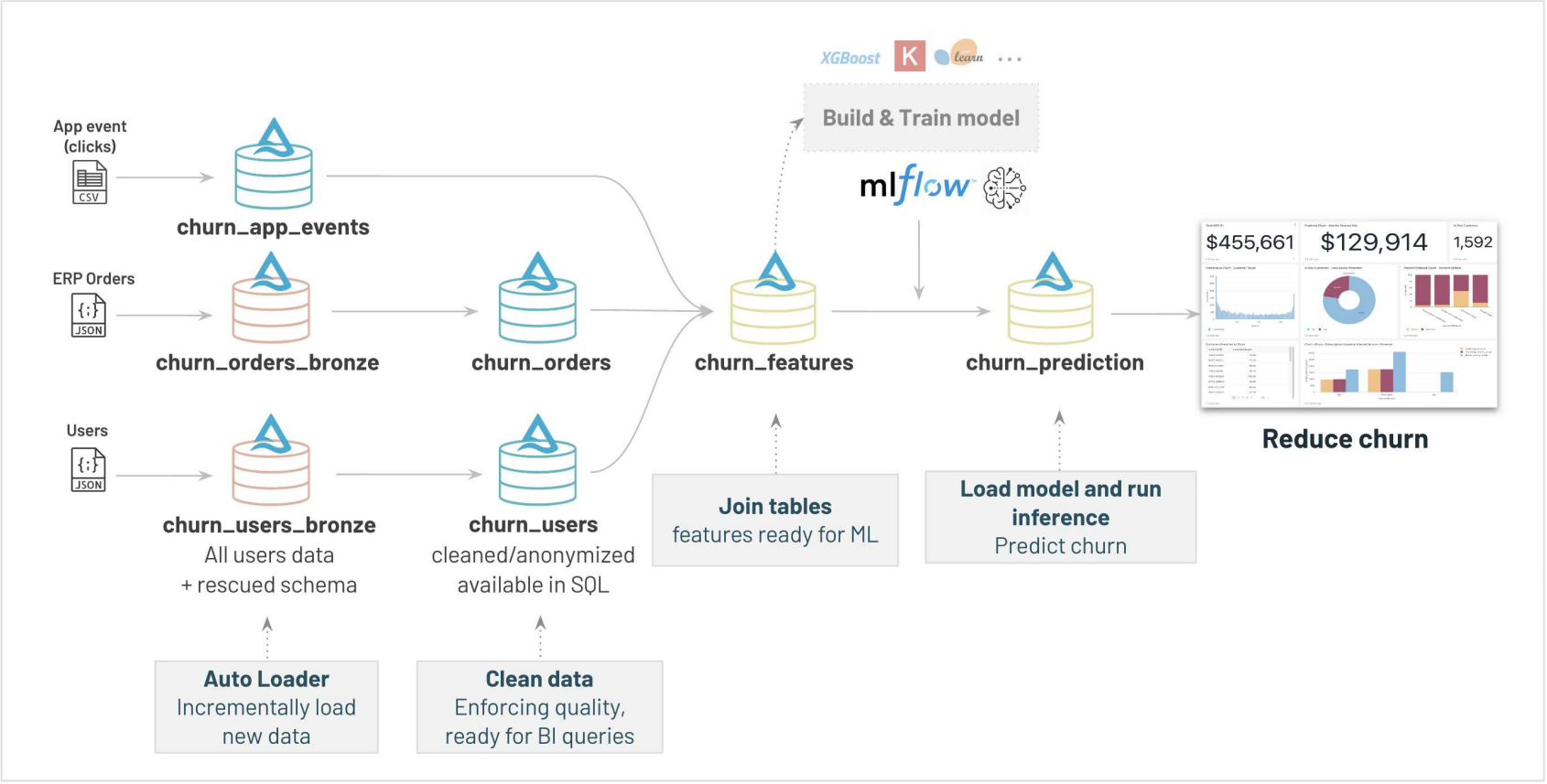
```
ordersFolder = rawDataDirectory + '/orders'
usersFolder = rawDataDirectory + '/users'
eventsFolder = rawDataDirectory + '/events'
print('Order raw data stored under the DBFS folder "' + ordersFolder + '"')
print('User raw data stored under the DBFS folder "' + usersFolder + '"')
print('Website event raw data stored under the DBFS folder "' + eventsFolder + '"')
```

```
Order raw data stored under the DBFS folder "/cloud_lakehouse_labs/retail/raw/orders"
User raw data stored under the DBFS folder "/cloud_lakehouse_labs/retail/raw/users"
Website event raw data stored under the DBFS folder "/cloud_lakehouse_labs/retail/raw/events"
```

What we are going to implement

We will initially load the raw data with the autoloader, perform some cleaning and enrichment operations, develop and load a model from MLFlow to predict our customer churn, and finally use this information to build our DBSQL dashboard to track customer behavior and churn.

Bronze layer: Data close to source. Silver: QA. Gold: Highest grain.



Let's start with

Data Engineering with Delta ($./01%20-%20Data%20Engineering%20with%20Delta)