SAVA

SAVA Technologies Ltd.

Data Engineer

Case Study

CONFIDENTIAL

The case study is divided in 2 sections.

Section 1 – ETL pipeline development (Estimated time 60 minutes)
The first section consists of a coding task in which you are asked to build a simple ETL pipeline that can be run on a local machine. The pipeline extracts data from a set of files generated from experimental tests, perform a set of simple transformations and stores the results.

Section 2 – System level design (Estimated time 60 minutes)
The second section consists of designing a cloud based data solution able to meet a set of requirements which include your previously developed pipeline along with data storage, automatic processing and data visualization.

In the follow up video call (duration approximately 1 hour), you will be asked to present your solution including the answers to the listed questions. We kindly ask you to keep the presentation format inside 20-30 minutes to make room for questions.

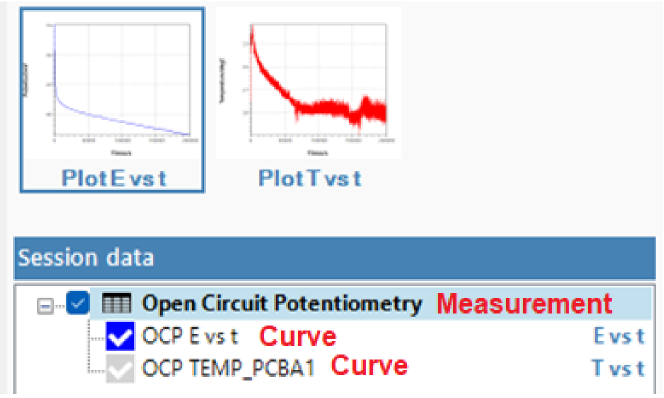
Section 1 – ETL pipeline development

A) CONTEXT

An Open Circuit Potentiometry (https://www.palmsens.com/knowledgebase-topic/open-circuit-potential) experiment is a standard electrochemistry experiment to test the stability of a system, measuring the open circuit potential over time.

The Chemistry team collects experimental data using the PalmSens software (https://palmsens.com/software/). The software does not have a free to use version but you are given access to a python SDK that is able to read the content of a .pssession file.

During an experimental session several measurements can be recorded, from which a series of curves can be generated.



In this example, the Chemistry team performed 10 experiments on 2 batches of 5 different sensors (1 sensor per experiment) and collected OCP data. From each measurement they obtained 2 curves: E vs t (potential vs time) and T vs t (temperature vs time). The goal of the Data team is to extract a set of metrics from each experiment and report the results.

B) FOLDER CONTENT

You will find in the:

- **10 .pssession files** (along with screenshots displaying the content of each file) obtained from the PalmSens software, containing experimental data from **2 different batches** of sensors, every filename is the sensor ID.
- **pspython/,** an SDK library to extract data from a PalmSens file
- A python script in the **./example.py** file from which you should be able to understand how to use the SDK to extract the data from the experiment.
- A basic class describing OCP data is implemented in **ocp_data.py** and can be used to calculate some relevant metrics, using the function calculate_drift().

C) GOAL

Your goal is to write an ETL pipeline consisting of these operations:

1. **Extract**
   a. Use the SDK to load the data from the experiment and obtain an OCP object
2. **Transform**
   a. Modify the OCP class in ocp_data.py implementing some functions to:
      i. Calculate the experiment duration in hours
      ii. Allow to discard two portions of data at the start and the end of the experiment (to remove bad data)
      iii. Convert the data to a pandas dataframe containing the voltage values at each timestamp
   b. Export the results (after discarding the first hour and the last 2 hours from the experiment) in 3 formats:
      i. A file containing the metrics extracted using calculate_drift()
      ii. A file containing the voltage values at each timestamp
      iii. A .png displaying the voltage values over time
3. **Load/Save**
   a. Store locally the results obtained, along with experiment metadata (any piece of information that you deem necessary to classify and organize the results)

D) NOTE

This section aims to test your ability to develop (locally) a data pipeline that is scalable and flexible. This includes the possibility of anticipating different experiments type and ability to handle errors in data format. Keep these requirements in mind when developing your code. You are not asked to fully implement all the functions you deem necessary if it’s too time consuming but you can replace them with fixed/blank outputs to display their intended purpose.

E) FAQ/Troubleshooting

- Please use python 3.8 (compatibility requirement for using the pythonnet library)
- Make sure the .dll files are enabled in code/pspython
- We kindly ask you not to share your solution publicly

Section 2 – System level design

A) CONTEXT

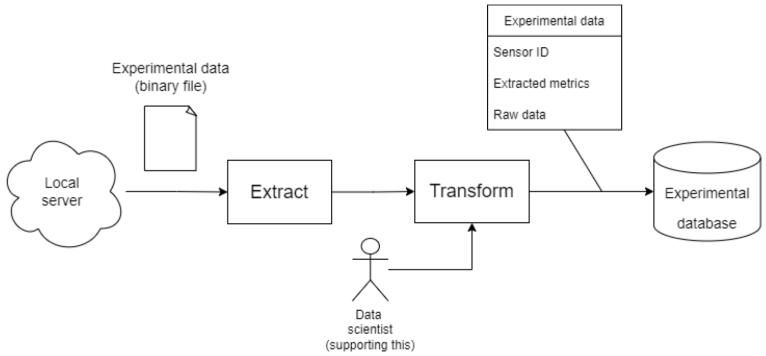
The **Chemistry team** performs several experiments per day. In each experiment, a different **sensor** is tested and the results are stored in a file (binary) and uploaded to the local laboratory computer.

The **Manufacturing team** logs information about sensor characteristics during production stages. These information are input using web forms that store the information directly in the cloud company server in an SQL database.

B) GOAL

Your goal is to design a cloud based data architecture solution able to meet the following requirements:

1) An ETL pipeline **(see task 1 for reference)** that automatically processes experimental data extracting relevant information and uploads the results to a database. Assume you are supported by a Data Scientist whose role is to define the transformation module that extract relevant metrics from each experiment.



2) A data warehouse featuring

- A clean organized database containing information from both manufacturing and experimental tests, which would facilitate all common Data Scientists operations: data analysis, merging tables, etc…
- Easy access to image files showing experimental data to allow Chemistry team members to visualize processed results

3) A visualization tool that is used to display weekly statistics about experimental results and manufacturing performance to Managers.

C) SOLUTION FORMAT

Please provide a short presentation containing

1) A simple sketch of your proposed architecture listing the mandatory elements to be included in the system, paying attention to the interfaces and the framework required for the software to work robustly;

2) The pros and cons of the elements and the topology (how the elements are arranged) you proposed, describing what aspects make them the ideal candidate to meet the requirements in your view.

3) An answer to this list of questions:

A. How would you handle incompatible formats during files acquisition? (e.g. missing information or incorrect naming of files)
B. How would you set up the ETL tasks so that new data are automatically updated once a new experiment is submitted by an user?
C. How would you set up unit and system level testing?
D. How would you ensure that data can be recovered in the event of a complete failure?
E. How would your architecture solution change if company data size changed from Pentabytes to a few Gigabytes?