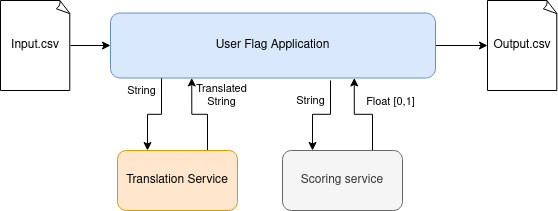
# Content Moderation System

A social network startup has hired you to participate in building its content moderation system. This system will identify users who post offensive or abusive messages in the comments section by assigning a score to each message and generating a report with the offensive score per user.

The Lead Architect has provided you with the following specifications for the project:



1. **Input** data will be provided as a CSV file whose columns are:
   * user\_id: string. The identifier of the user
   * message: string. The message written by the user
2. There will be a **Scoring service**, a REST API developed by another team, that will receive a string as an input and will return a float value between 0 and 1, assigning an offensiveness score to the message.
3. Since the scoring service will only work with English text, there will be another REST API for a **Translation service** that receives the message string as an input and returns the same message translated to English, or the same message if the message was already in English.
4. The **output** data will be a CSV file with these columns:
   * user\_id: string
   * total\_messages: integer. The total number of messages written by the user.
   * avg\_score: float. The average offensiveness score for all the messages of the same user.

Your job is to develop the **User Flag Application** that reads the input file and generates the output file according to the specifications interacting with the external services indicated. **You don’t need to build the translation and the scoring services** but you can simulate their behavior.

## Additional requirements and considerations

* The Translation and Scoring services will have some **network latencies** when interacting with them. These are expected to be between 50ms and 200ms but they can process multiple concurrent requests.
* Your application needs to perform well even for large input data files with millions of entries. It should be able to process files like this as fast as possible.
* The Translation and Scoring services are **idempotent**. Consider the scenario of bots spamming the social network with the same message multiple times.
* You can write your application in **Python** and you have freedom to use popular frameworks (FastAPI, Flask, Airflow..).
* You may spend as much time as you like, but a few hours should be sufficient.
* Please submit the source code with a README file with instructions on how to build and run it
* We will evaluate the exercise considering the correctness, performance, style, readability and test coverage.