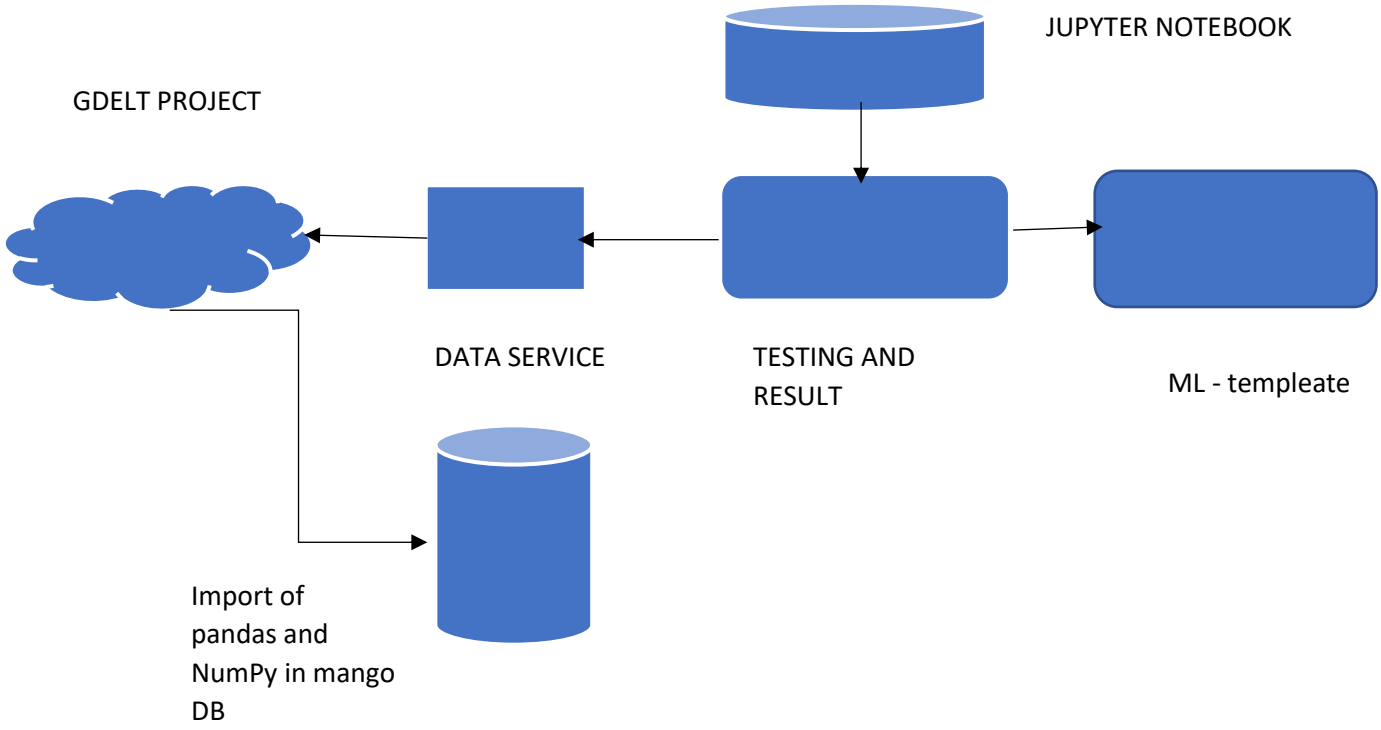
ARCHITECTURE PROPOSAL:



This above design can also perform in python Language. And implemented with jupyter notebook or colab.

Standard workflow is presented on the picture above. The numbers on the diagram have the following meaning:

1. The user accesses Frontend (web application, aka *Midas Eyes*) to obtain information about existing models and previous executions. Also, the user can create and train the new model or apply an existing one as a new input data via the UI.
2. Information about executions and models is requested from Executor Service by Frontend via WebSocket.
3. Metadata of models and executions is stored in the PostgreSQL database.

4. Results and their estimations are requested from Result Evaluation Service by Frontend via WebSocket.
5. Results and estimations are stored in the PostgreSQL database.
6. When “train” execution has been started, Executor Service downloads sources from GitLab. For the “apply” execution they are already stored and can be run immediately. Then Executor Service runs the ml-template extended by a developer with some algorithm.
7. Before starting a job on the Spark cluster, ml-template checks an existence of the input data, specified by the user, in Data Service via REST API.
8. If data, required by the execution, is missing, Data Service downloads it from external sources — in our case, GDELT and Yahoo.
9. Data Service stores downloaded data in the Mongo database.
10. When data is ready, ml-template submits the task to the Spark cluster.
11. During the execution Spark requests necessary data from the Mongo database.
12. When execution is finished, results are submitted to Result Evaluation Service. This service makes necessary estimates including comparison with the ground truth (“label” in terms of Spark).

Also, a few words about the technological stack of microservices. Data, Execution and Result Evaluation services are powered by Java, Spring Boot, Hibernate and Gradle. The ml-template is written using Scala, but can be created also on Python, R or Java. Frontend is based on React.js and Bootstrap.