

Project

MLOps experimental

Giorgia Bertacchini

2022

Intro

The project is a practical implementation of **MLOps practices**.

For all life cycle of ML are create two project that represent the phases: develop and production.

Project

Develop phase

- Data Ingestion
- Data Preparation
- Model Building & Training
- Experimentation
- Model Deploy
- Model Serving

Production phase

- Prediction Service
- Monitoring
- Alerting
- Application & Trigger

Tools & libraries

Develop phase

Workflow
orchestration



Data analysis
and
manipulation



Model
training



Experimentation
management



Model packaging
and serving



Data
versioning



Code
versioning



Deploying
pipeline kedro

kedro-docker

Tools & libraries

Production phase

Drift and model
monitoring



Systems
monitoring
and alerting



Alert
management

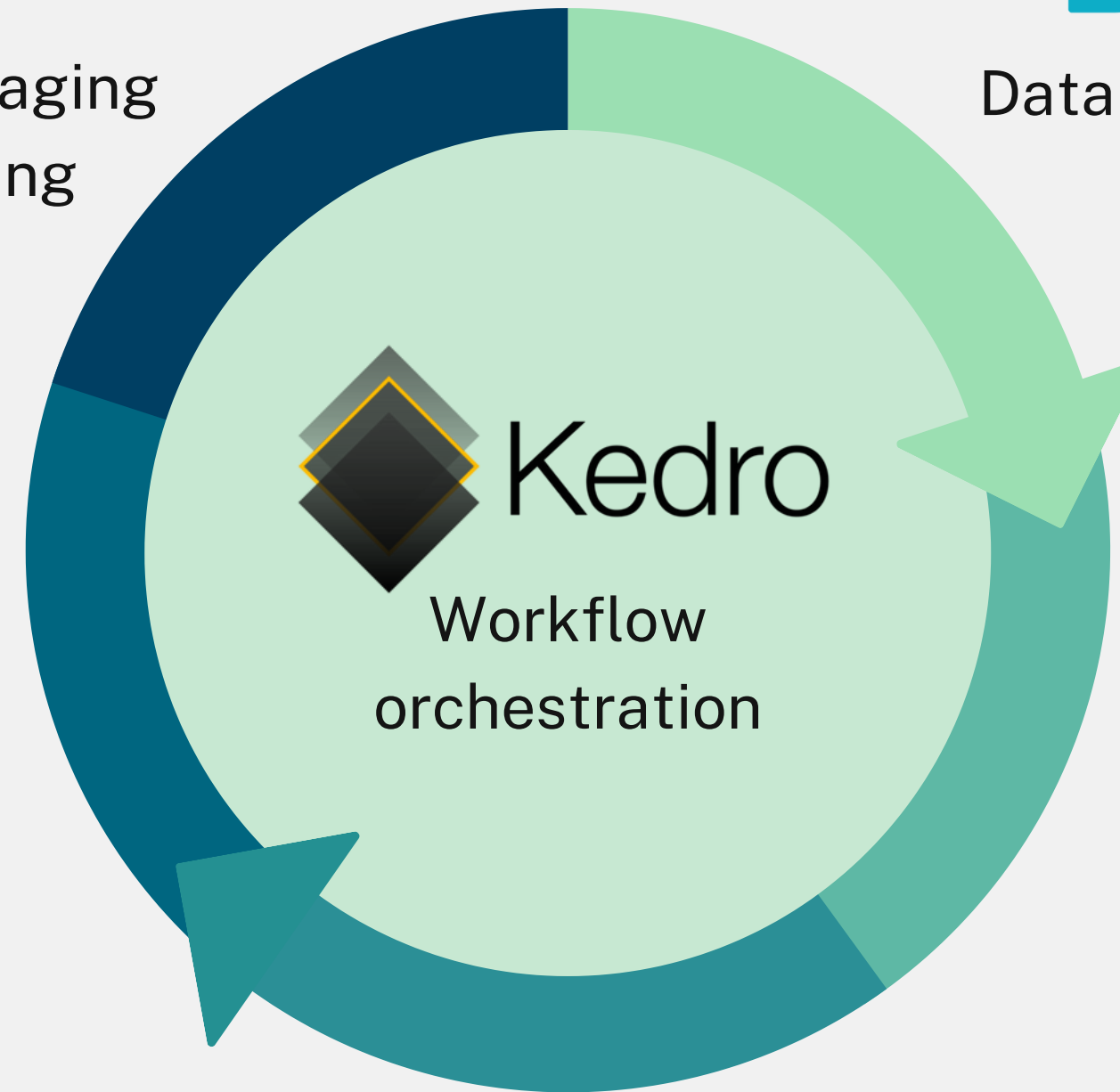
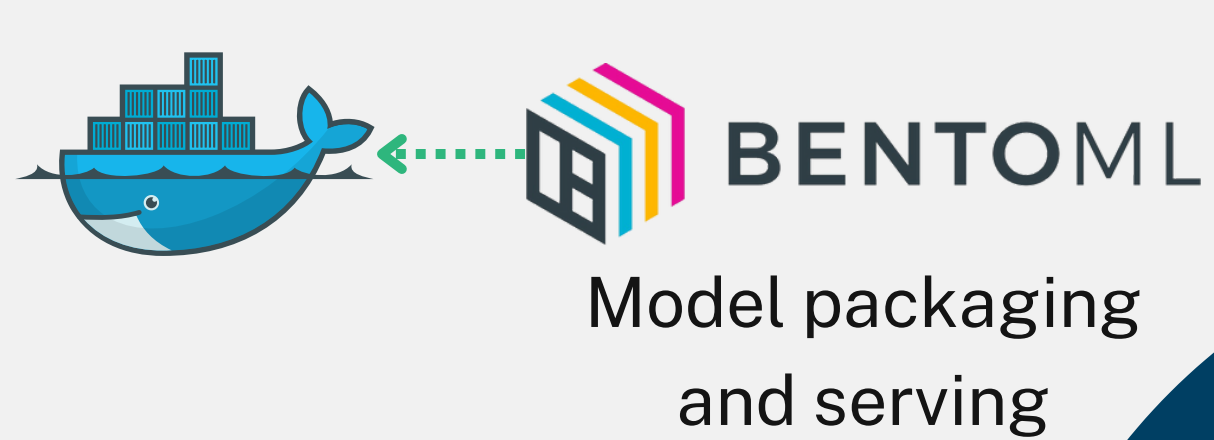


Managed
observability
platform

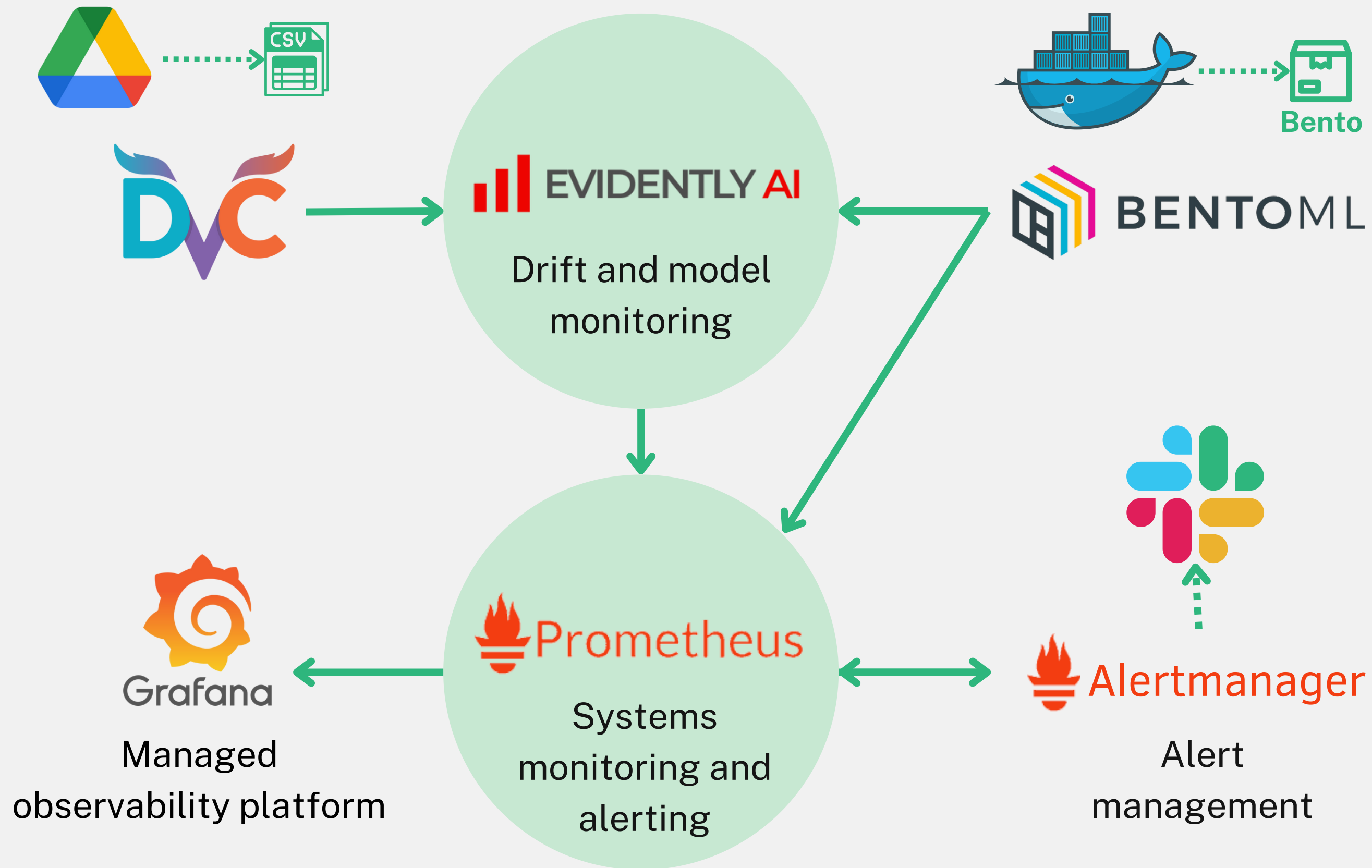


Building
application





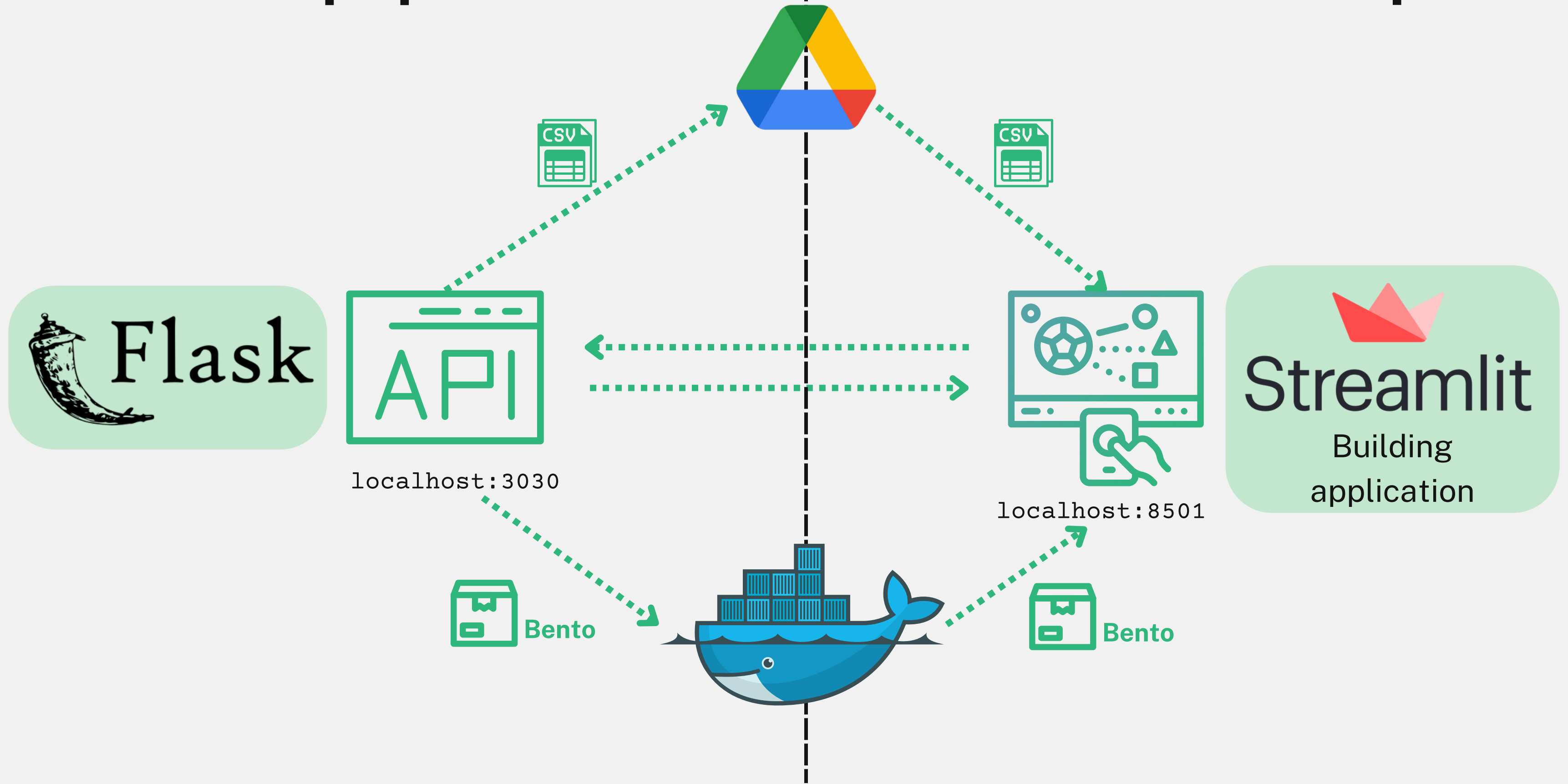
Develop phase



Production
phase

Develop phase

Production phase



Commands line

```
python run.py pipeline
```

```
python run.py exp
```

```
python run.py pipeline-docker
```

```
python run.py run
```

```
python run.py new-dataset <global_url>
```

Develop phase file: run.py

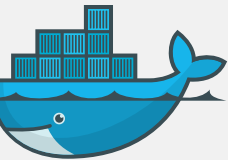
open kedro ui to localhost:4141



open mlflow ui to localhost:5000



create docker image about pipeline to create ml model



run training model and building bento service



upload dataset from file csv with <global_url>



Flask app

Develop phase file: app.py

```
@app.get("/dvc_file")
```

return the code for take reference dataset from Google Drive.

```
@app.route("/load_new_data", methods=[ "POST" ])
```

receive a dataframe to load as new reference dataset.

```
@app.get("/retrain")
```

load in Google drive new dataset, run mlflow to training model, build and dockerize bento.

```
@app.get("/bento")
```

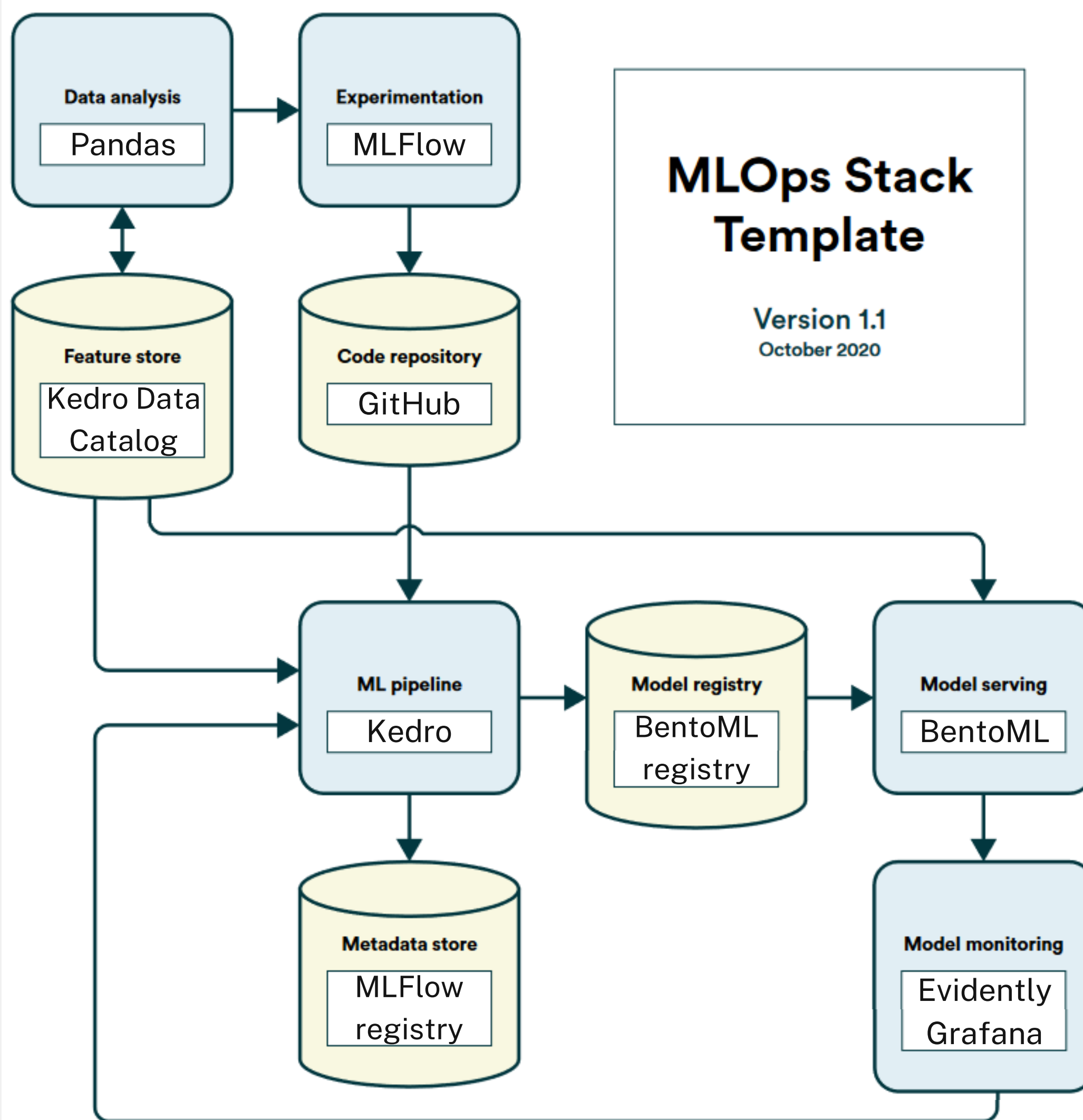
build and dockerize bento.

```
@app.get("/header")
```

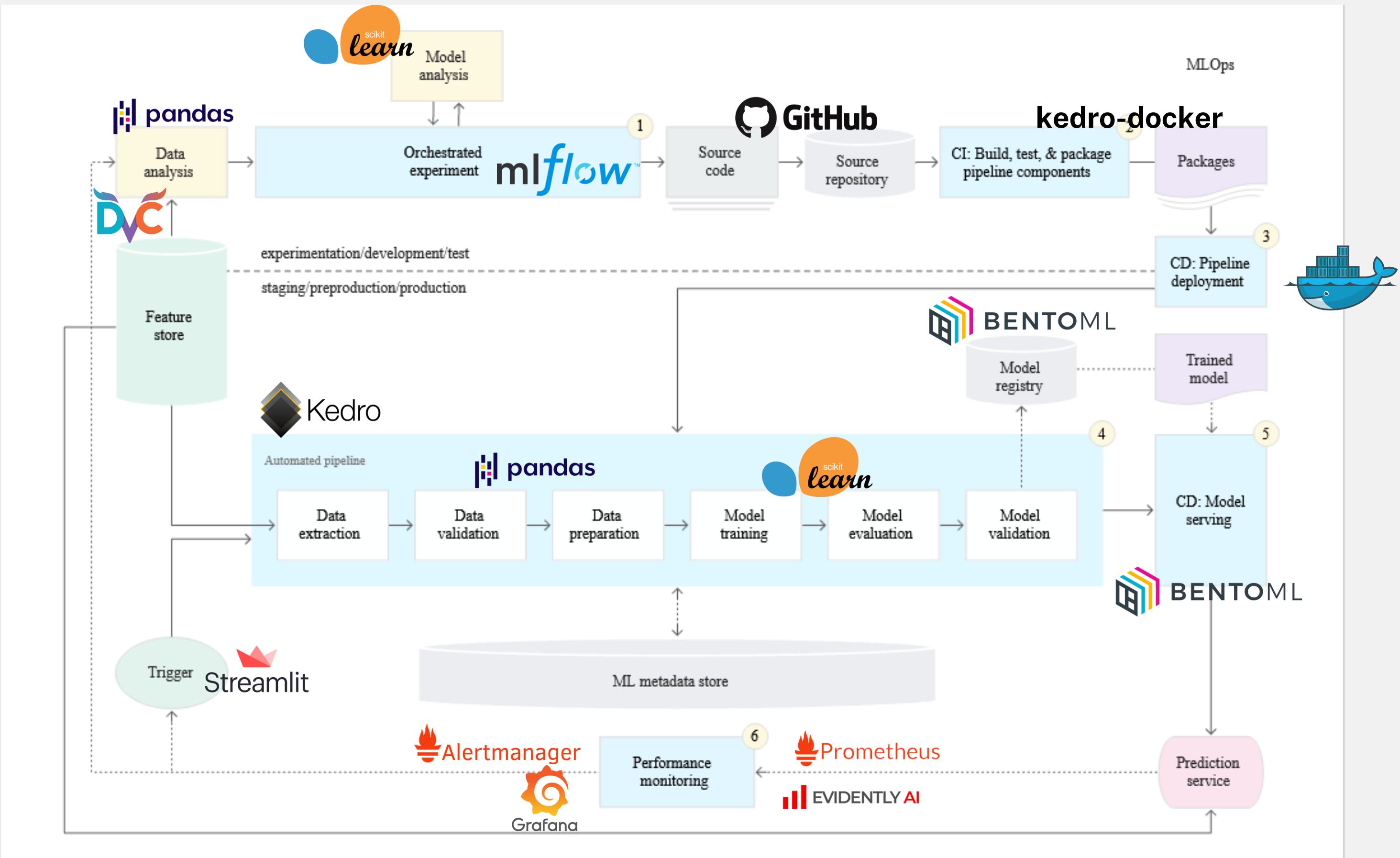
return name columns of dataset header.

MLOps Stack Canvas

Project name: MLOps tirocinio			Team members: Giorgia Bertacchini		
Data analysis & experiment management For data analysis uses Pandas, a Python library. For experiment management uses MLFlow, an open source platform.	Data Sources & Data Versioning Data sources are static file in table format, as csv. Use specialized data versioning systems as DVC to save in a Google Drive folder the raw dataset. Another system use is Kedro that save each versions of intermediate and output data and of ml model.	Value Proposition This project would predict the quality of each case receive from csv file.	ML pipeline orchestration The DAG (directed acyclic graph) is create by Kedro tool, an open-source Python framework.	Model registry & Model versioning All model are collect in MLFlow ui and in BentoML platform.	
		MLOps Dilemmas Tooling: are all open-source. Platforms: using a hybrid solution, with more platforms.		Model deployment Ml model and its environment are deployment through BentoML, that create deployable artifacts (Bentos).	
		Foundations For source version control system uses GitHub.	Model & Data & Application Monitoring To observing the model there are Evidently and Prometheus tools, which can alerts user and Grafana provide a dashboard for users.	Prediction serving BentoML provide a service, to require predictions to ml model in Bentos.	



MLOps stack



MLOps stack: Specifications



Development and experimentation: you iteratively try out new ML algorithms and new modeling where the experiment steps are orchestrated. The output of this stage is the source code of the ML pipeline steps, which are then pushed to a source repository.



Pipeline continuous integration: The outputs of this stage are pipeline components (packages, executables, and artifacts) to be deployed in a later stage. Model validation via cross-validation for choose hyperparameters.

kedro-docker

Pipeline continuous delivery: you deploy the artifacts produced by the CI stage to the target environment. The output of this stage is a deployed pipeline with the new implementation of the model.



Automated triggering: the pipeline is automatically executed in production based on a schedule or in response to a trigger. The output of this stage is a newly trained model that is pushed to the model registry.

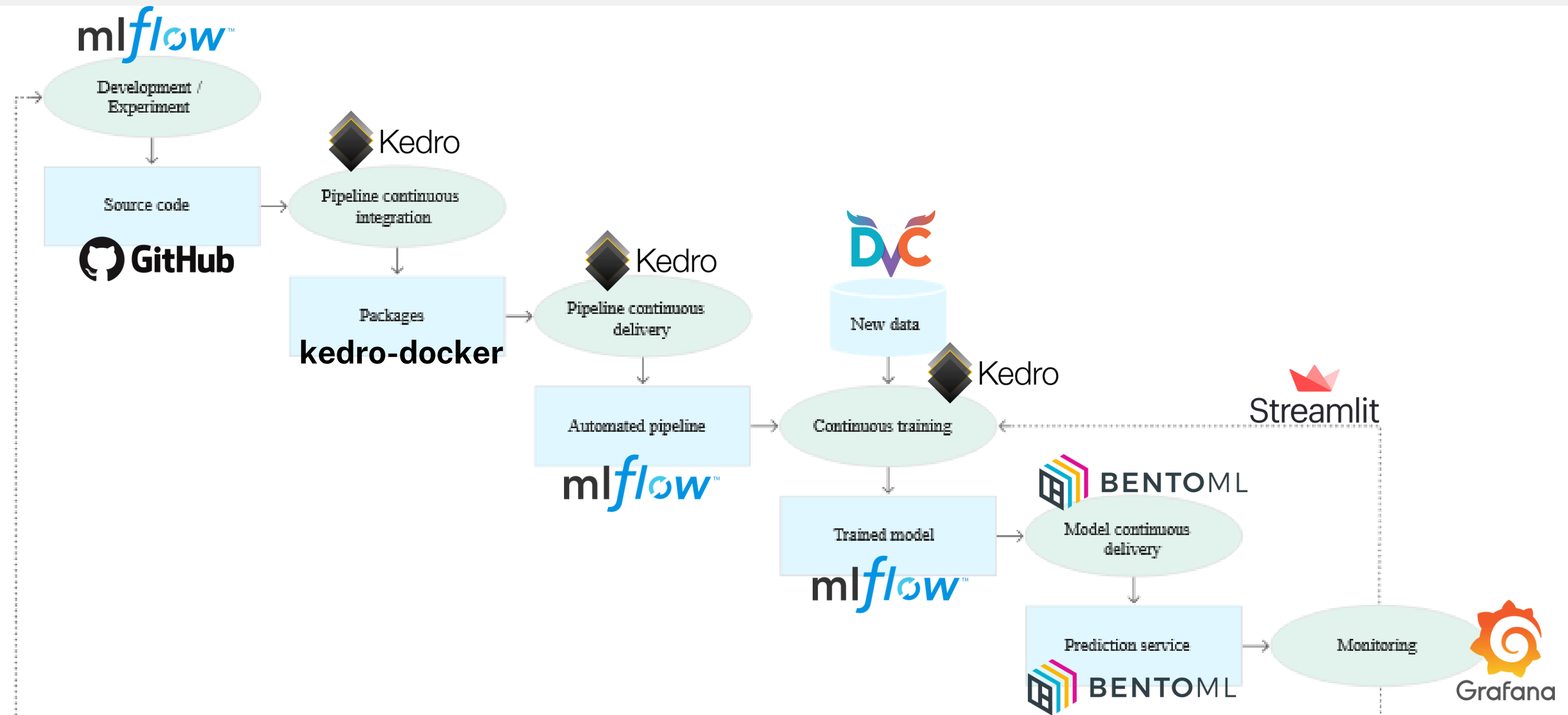


Model continuous delivery: you serve the trained model as a prediction service for the predictions. The output of this stage is a deployed model prediction service.



Monitoring: you collect statistics on model performance based on live data. The output of this stage is a trigger to execute the pipeline or to execute a new experiment cycle.

MLOps stack



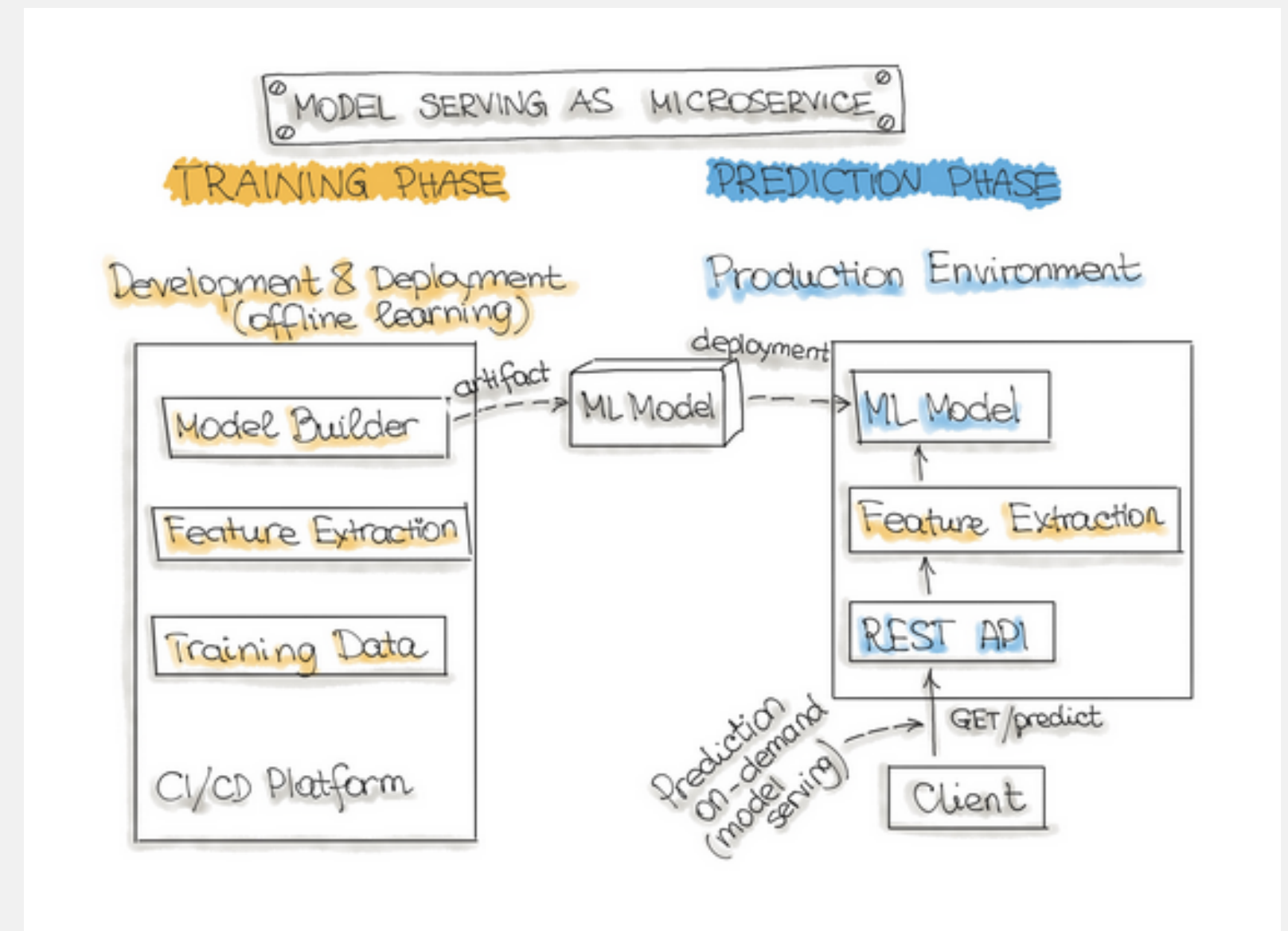
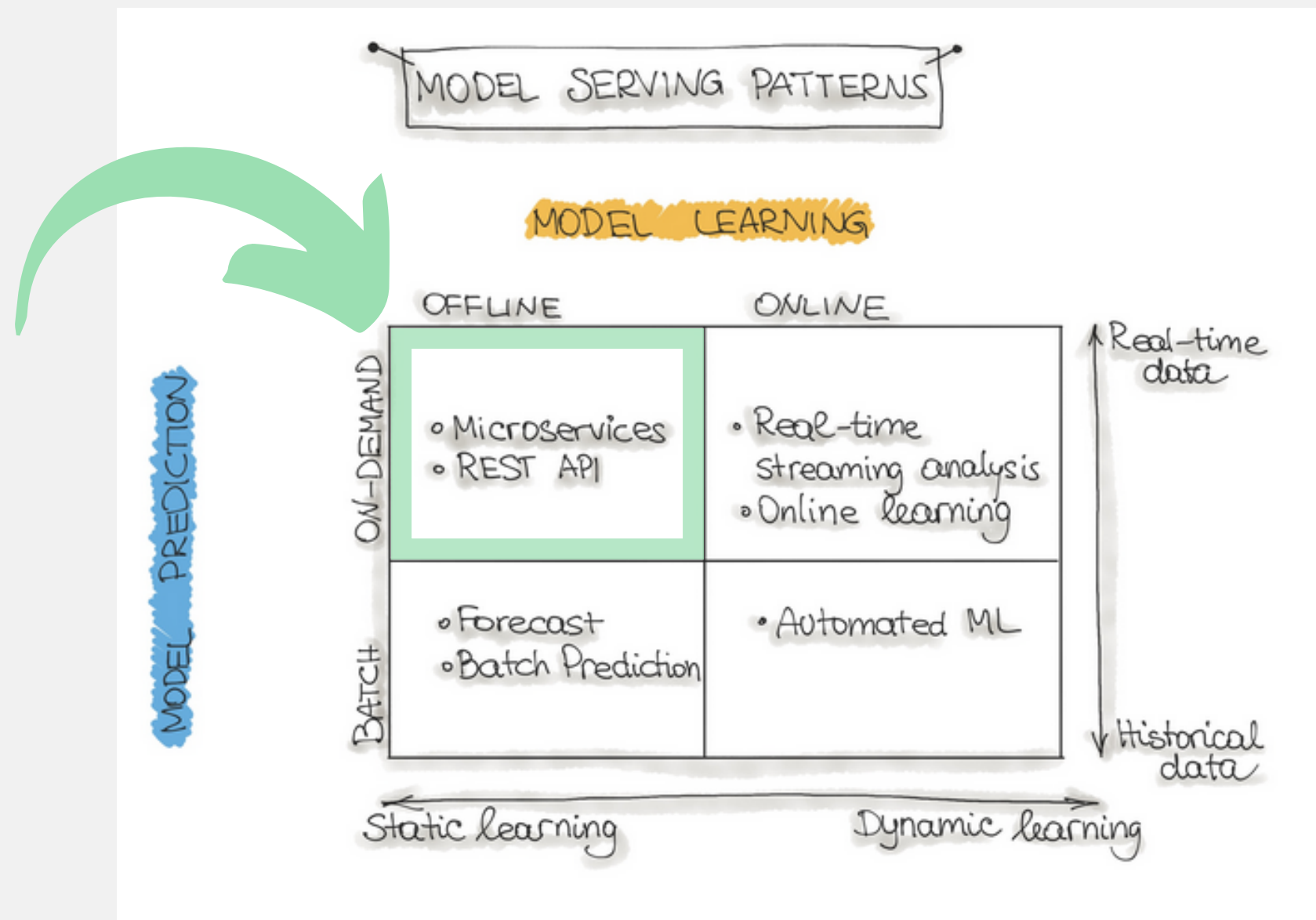
To classified the project

This project have:

- Model learning Offline
- Model predictions On-Demand

So use Microservices and Rest API application.

As we can see the next schema is very similar to previous schema, where the two phase are separate but connected.



GitHub url

Develop phase:

<https://github.com/giorgiaBertacchini/MLOps-DevelopPhase>

Production phase:

<https://github.com/giorgiaBertacchini/MLOps-ProductionPhase-Slide>