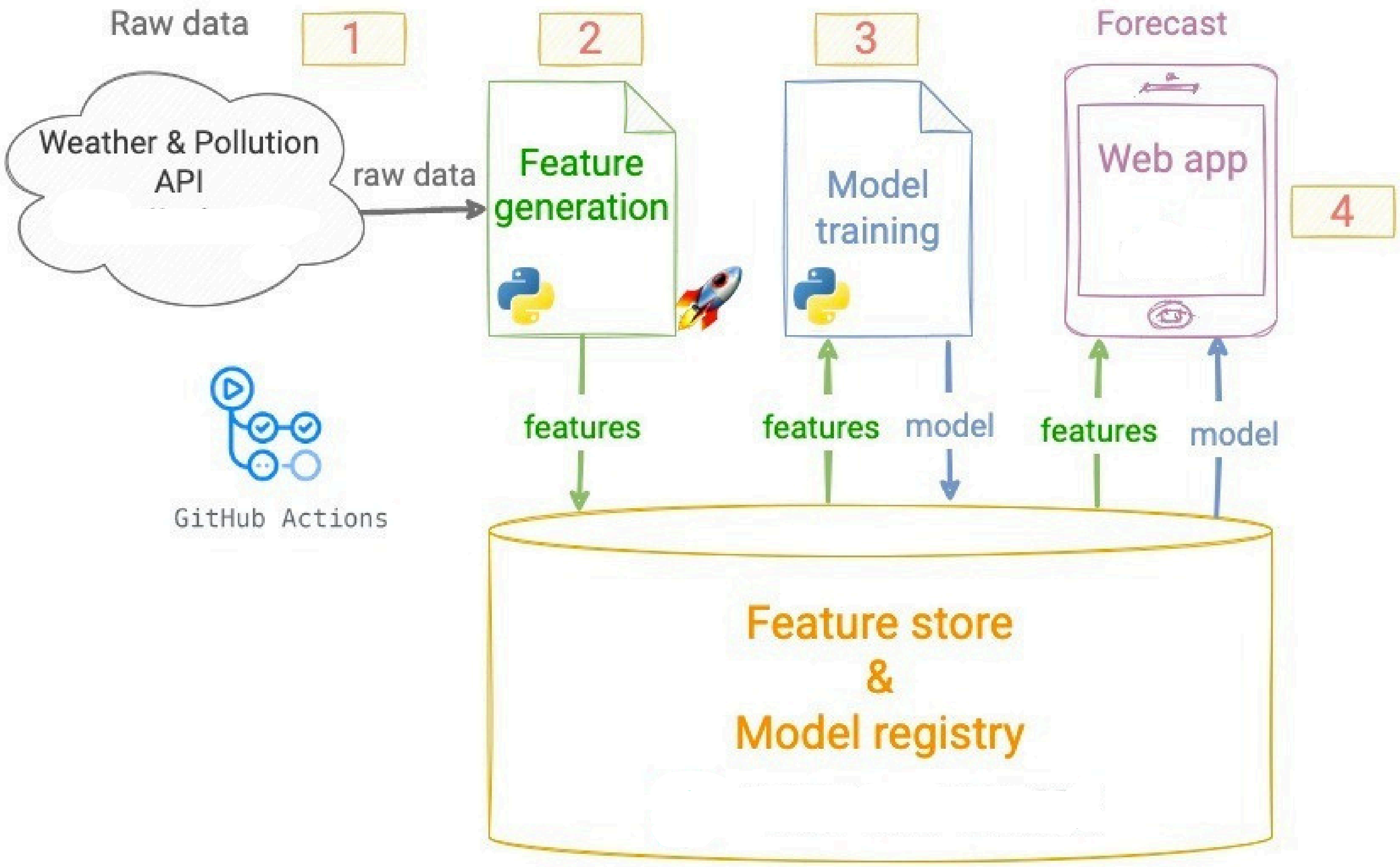
Pearls AQI Predictor

Let’s predict the Air Quality Index (AQI) in your city in the next 3 days, using a 100% serverless stack.

PROJECT

The following is a high level overview for you to achieve this

Air Quality Index (AQI) prediction service



Feature pipeline

Write a Python script that:

1 → Fetches raw weather and pollutant data from an external API like AQICN or OpenWeather

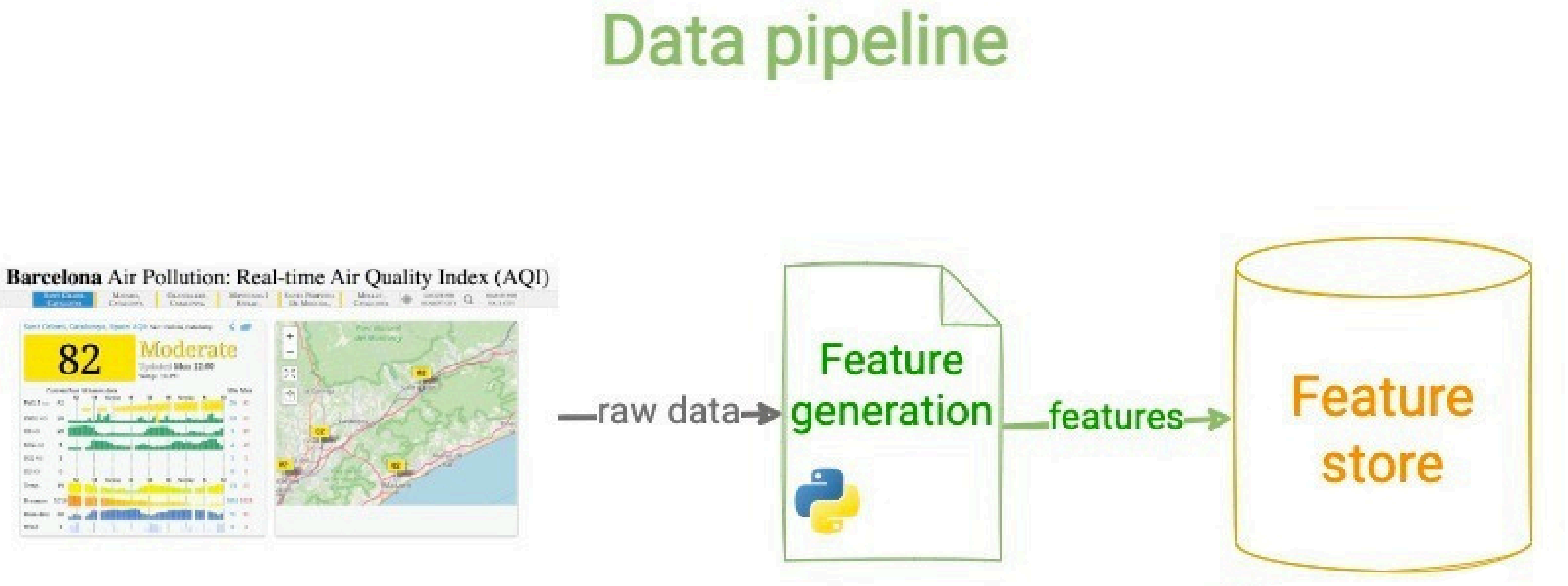
(The above api is just an example, you may need to explore other options too)

2 → Computes features from this raw data (aka model inputs), and targets (aka model outputs)

- Include time-based features (hour, day, month) and derived features like AQI change rate.

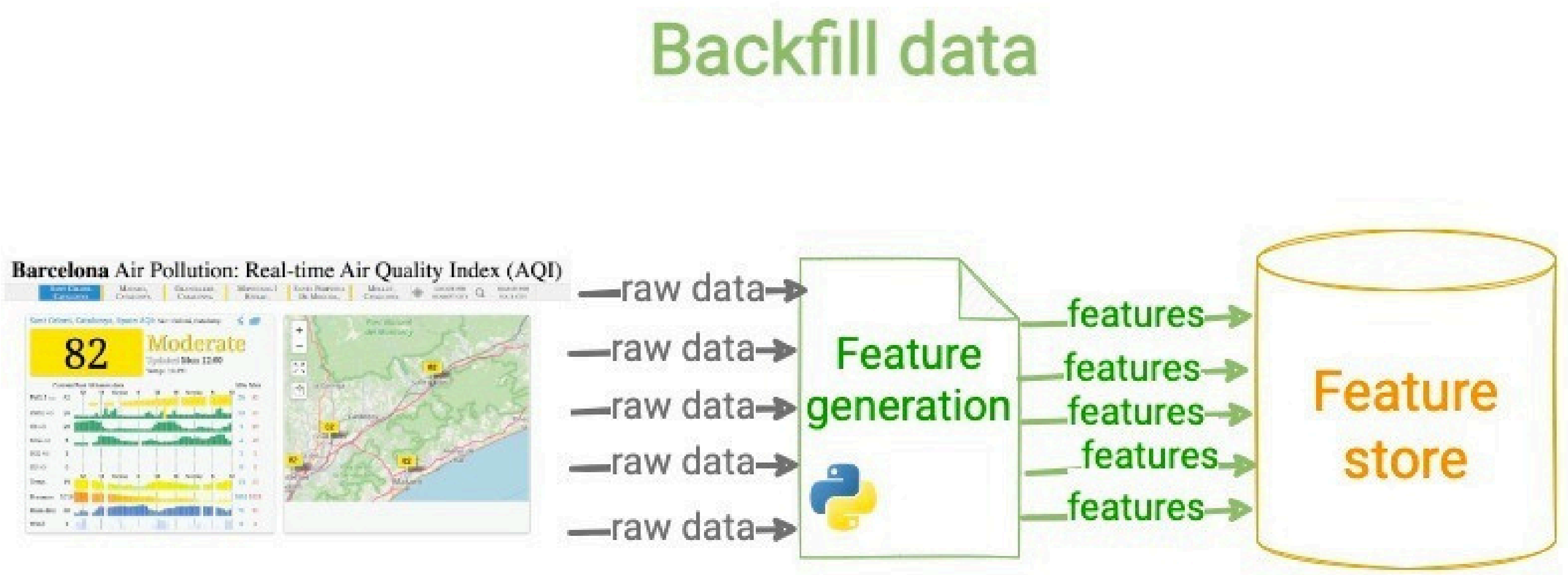
3 →Stores these features in the Feature store

- You may want to explore Hopsworks or Vertex AI (Free tiers)



Backfill historical (features, targets)

Run the feature script from step 1 for a range of past dates, to generate training data for your ML models.



Training pipeline

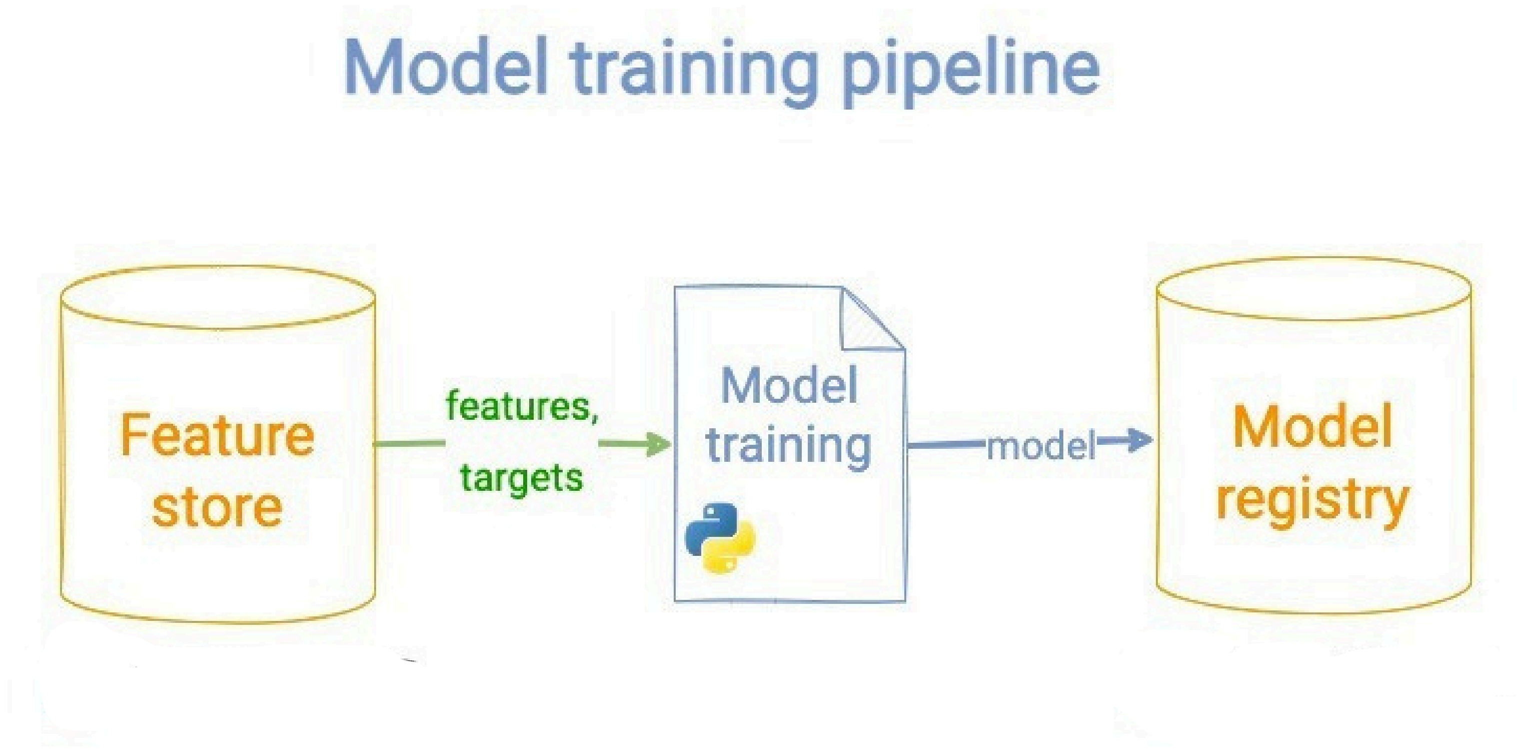
Write a Python script that:

1 → Fetches historical (features, targets)from the Feature Store.

2 → Trains and evaluate the best ML model possible for this data

- *Experiment with Scikit-learn models (Random Forest, Ridge Regression) and TensorFlow/PyTorch for advanced models.*
- *Evaluate performance using RMSE, MAE, and R^2.*

3 → Stores the trained model in the Model Registry.



Automate pipeline runs

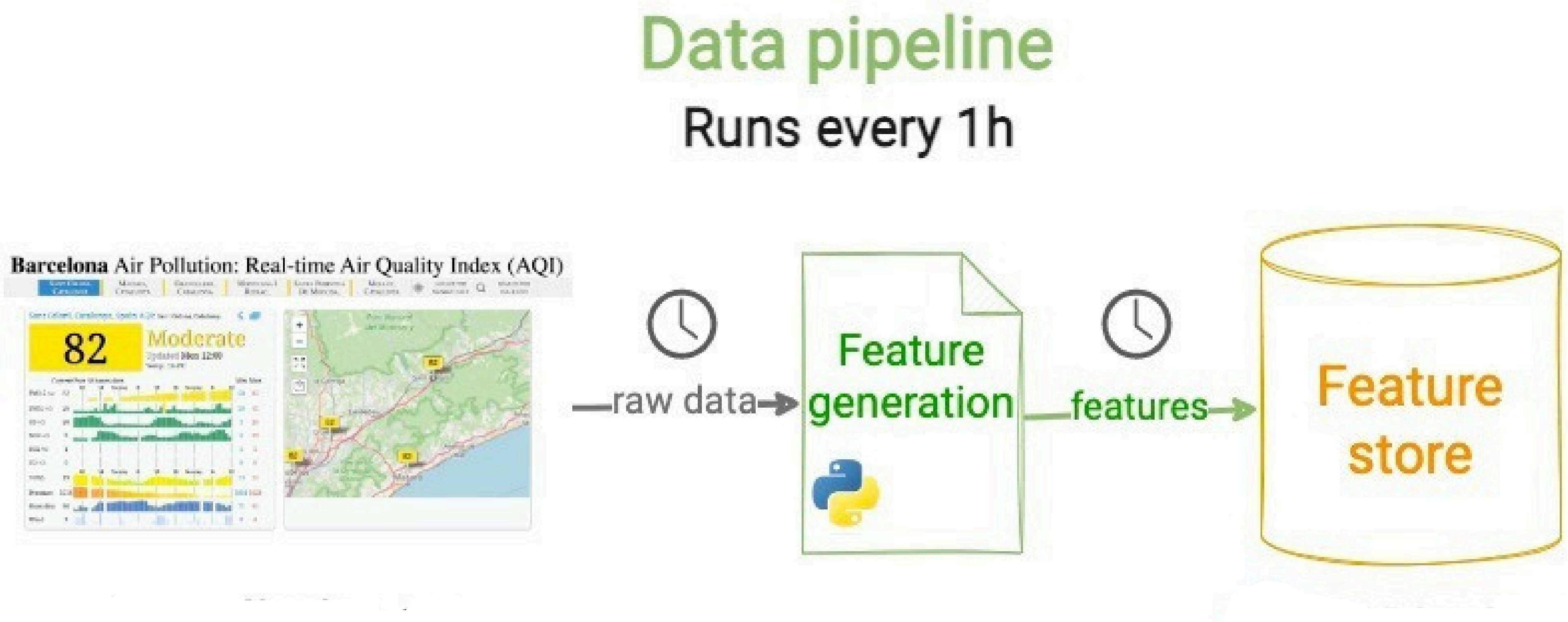
Create a CI/CD pipeline that automatically runs

→ the **feature script every hour**, and
→ the **training script every** day.

Some popular and free CI/CD tools that you can use are Apache Airflow and Github Actions but you are encouraged to explore other tools too

Data pipeline

Runs every 1h



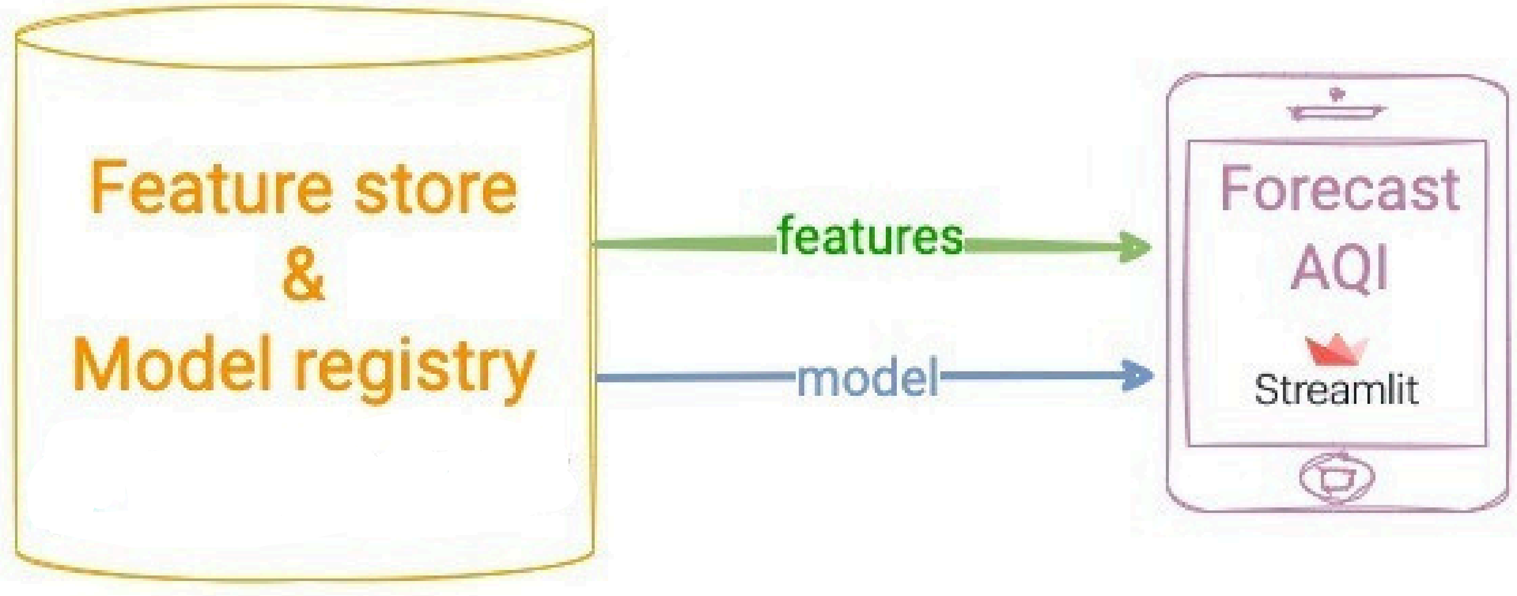
Web app to show model predictions

Your app

1 → Loads the model and features from the *Feature Store*,

2 → Computes model predictions and shows them on a simple and descriptive dashboard.

3 → Use Streamlit/Gradio and Flask/FastApi for the web app



Some Guidelines

- Perform EDA to identify trends.
- You should use a variety of forecasting models, from statistical modelling to deep learning models
- Use SHAP or LIME for feature importance explanations.
- Add alerts for hazardous AQI levels

Final Submissions:

- 1. End-to-end AQI prediction system.
- 2. A scalable, automated pipeline.
- 3. An interactive dashboard showcasing real-time and forecasted AQI data.
- 4. A detailed report documenting everything you managed to achieve