Let's design a real-time ML system ⚡

Step by step

The problem

Let's build a **real-time** API that serves short-term predictions on crypto prices.

For example, to predict the price of Ethereum (ETH) in the next 10 seconds.

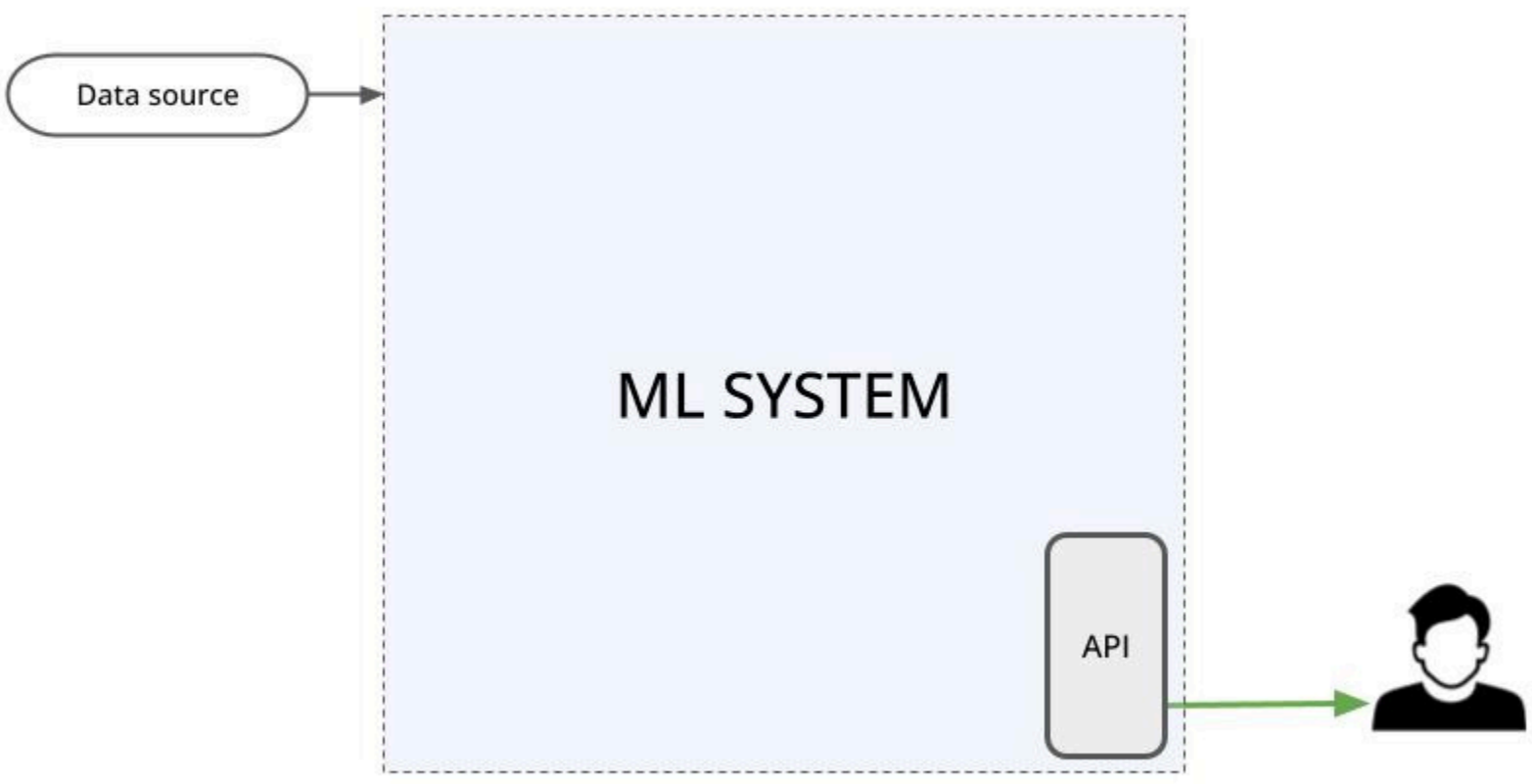
This API can be consumed by
> a **human trader**, or
> an **algorithmic trading bot**

to improve the return of a trading strategy.

ML System

Our API is just one component of an ML system that continuously

> **Ingests** raw market data in real-time and
> **Generates** new predictions using a Machine Learning model.

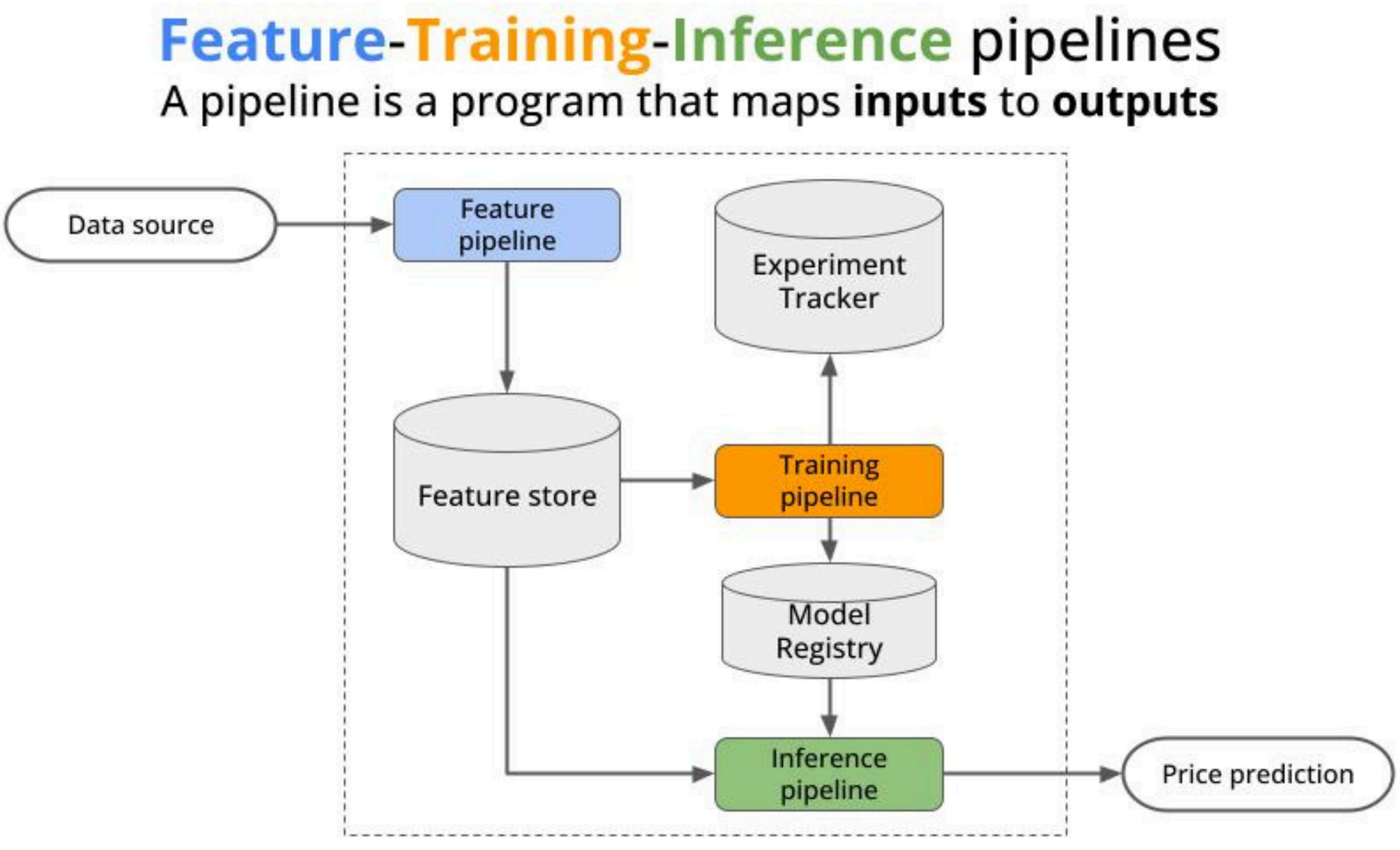


This system is made of **3 types of pipelines** (aka programs that map inputs to outputs)

> **Feature pipelines** -> map raw data to ML features.

> **Training pipeline** -> maps training data to a model artifact.

> **Inference pipeline** -> maps model artifact and recent data to predictions



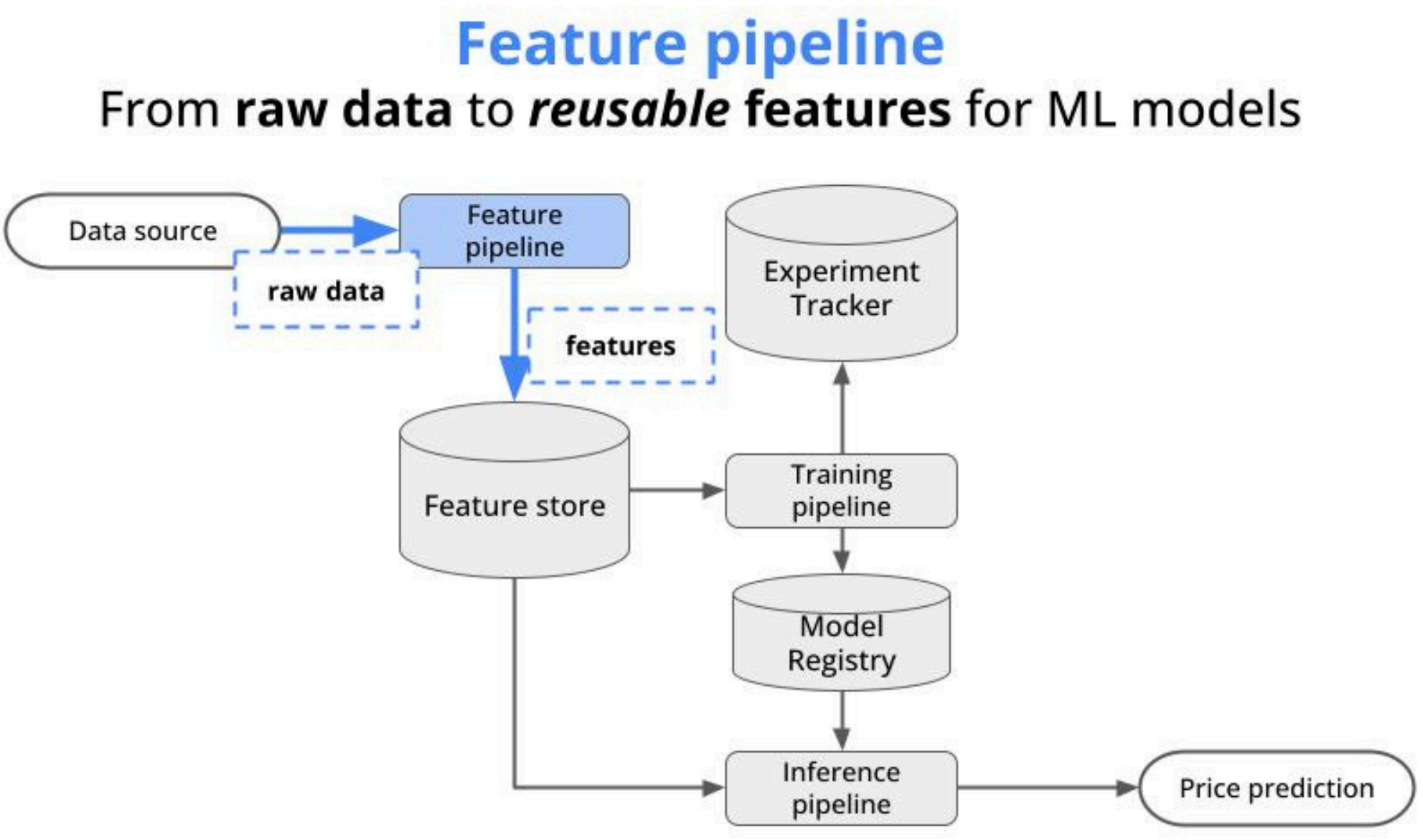
Feature pipeline

A Feature pipeline is a program that

> **ingests** real-time price data from an third party API.

> **transforms** price data into 10-second Open-High-Low-Close prices,

> **pushes** it to the Feature Store.



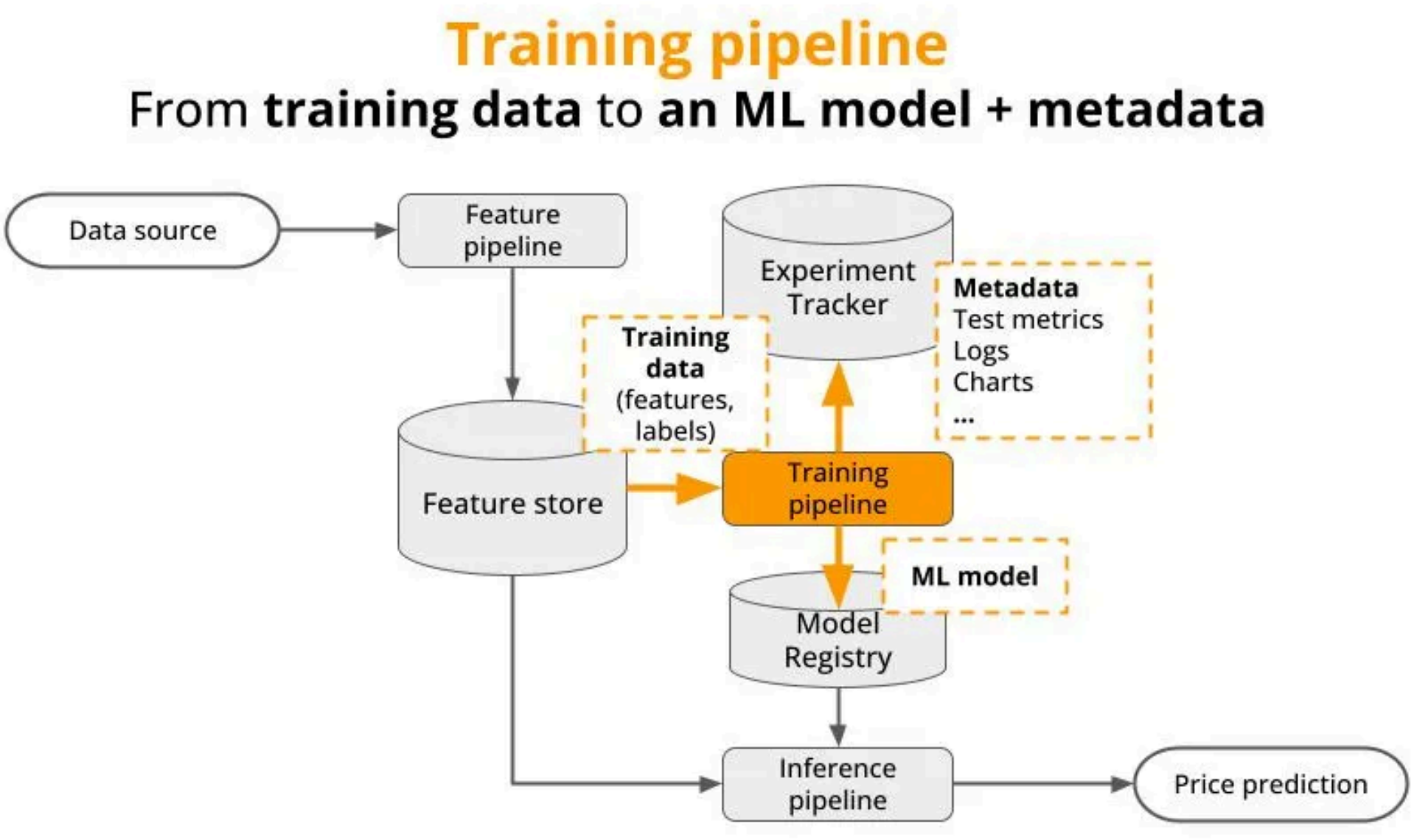
Training pipeline

The Training pipeline is a program that

> **ingests** training data from the feature store,

> **trains** a predictive ML model, and

> **pushes** the model to the model registry

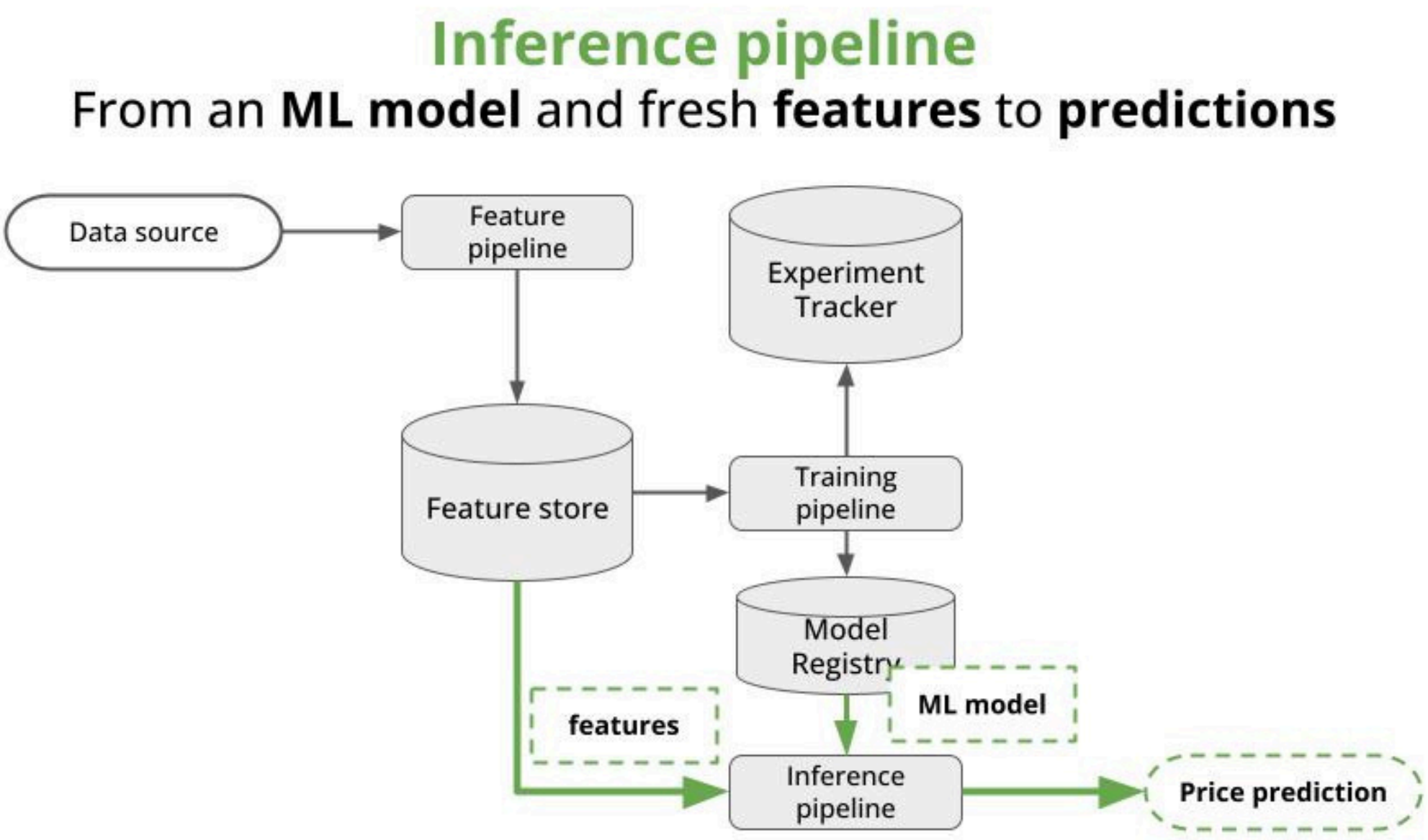


Inference pipeline

The Inference pipeline is a program that

> **loads** the model from the registry and recent features, and

> **serves** fresh predictions through our API.

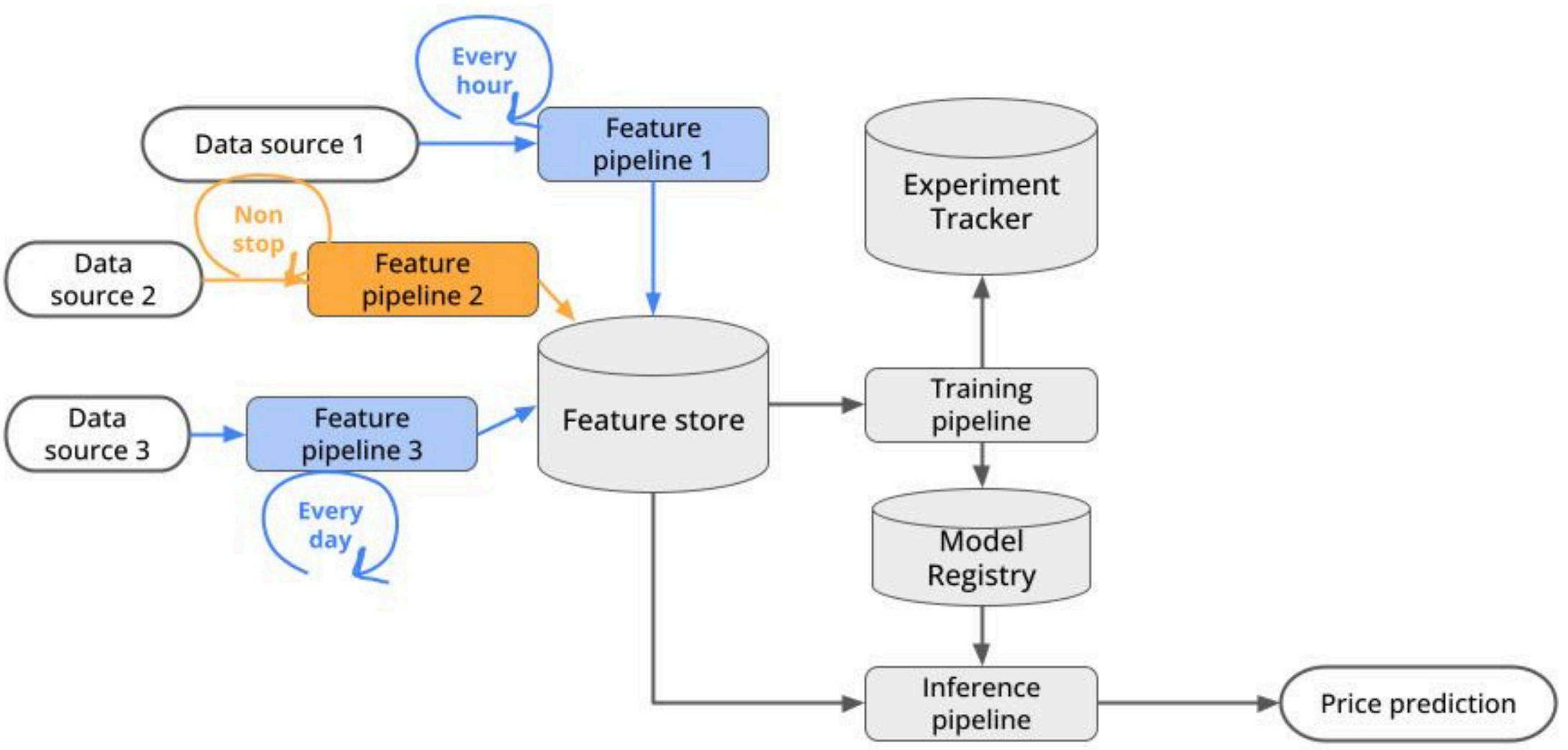


Now what?

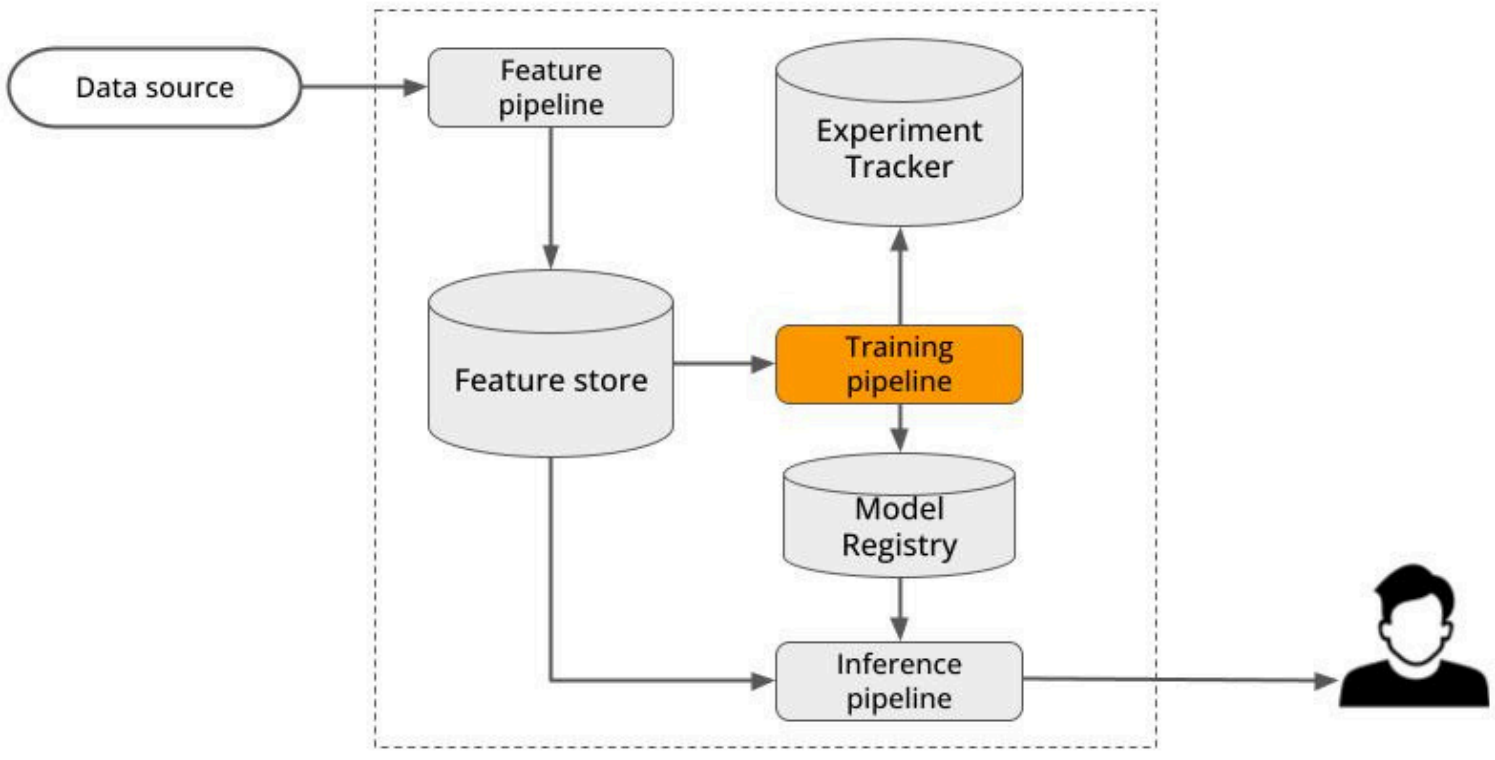
Once you have the 3 pipelines up and running, it is time to start improving the system.

For example

Add more **feature pipelines**, to provide more signals to your ML model.



For example
Improve the underlying ML model



For example
Optimizing the inference pipeline to produce **faster** predictions.