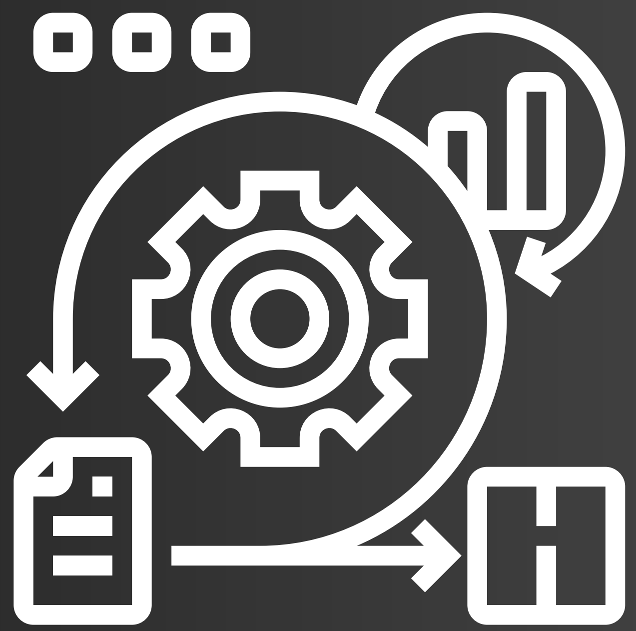
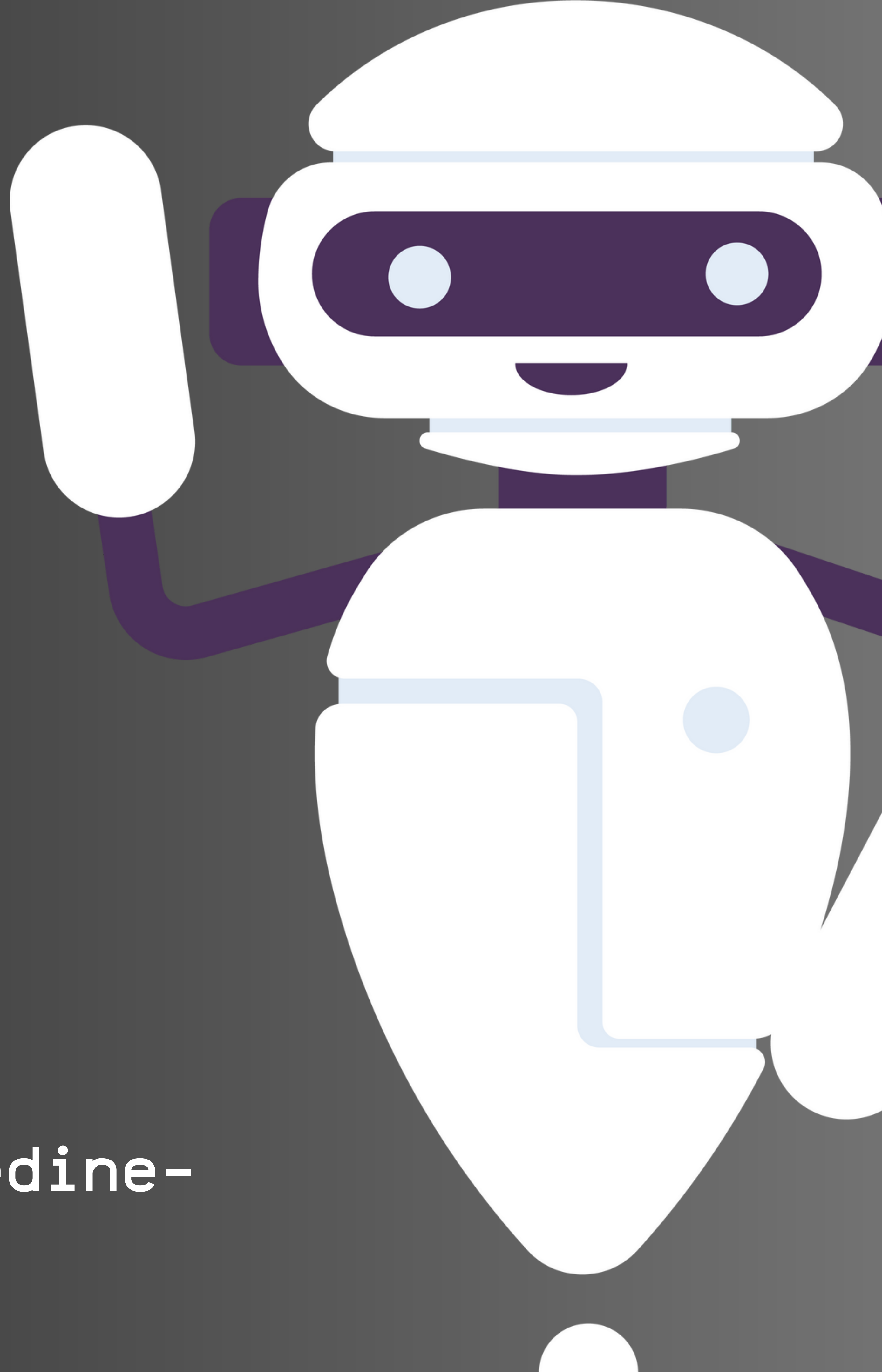


The Next Step After ML: MLOps



-With: AIT SAID Azzedine-

About Me

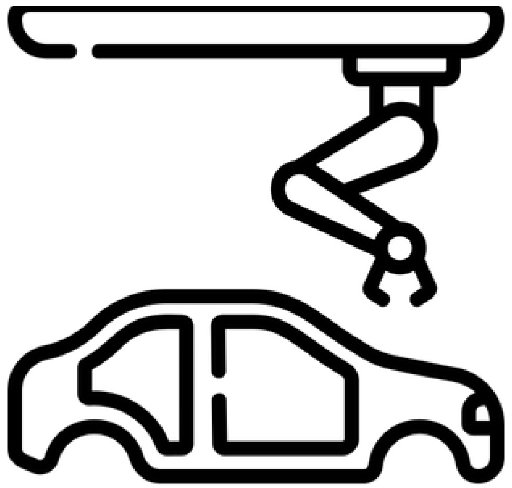
- Final Year Student at ESI Algiers Computer Systems Option.
- Technical Department Manager at School of AI Algiers.
- MLOps intern at Namla Networks
- I Love Data Science <3 !

Objectives

- **Know about the different levels of MLOps.**
- **Deploy a Machine Learning Model into Production.**
- **Monitor a ML Model.**
- **Create a CI/CD solution for a ML System.**

Kaizen !

KAI		ZEN		
改	+	善	=	"good change" aka "continuous improvement"
"change"		"good"		



Automobile Manufacturing

Adopt Final Solutions

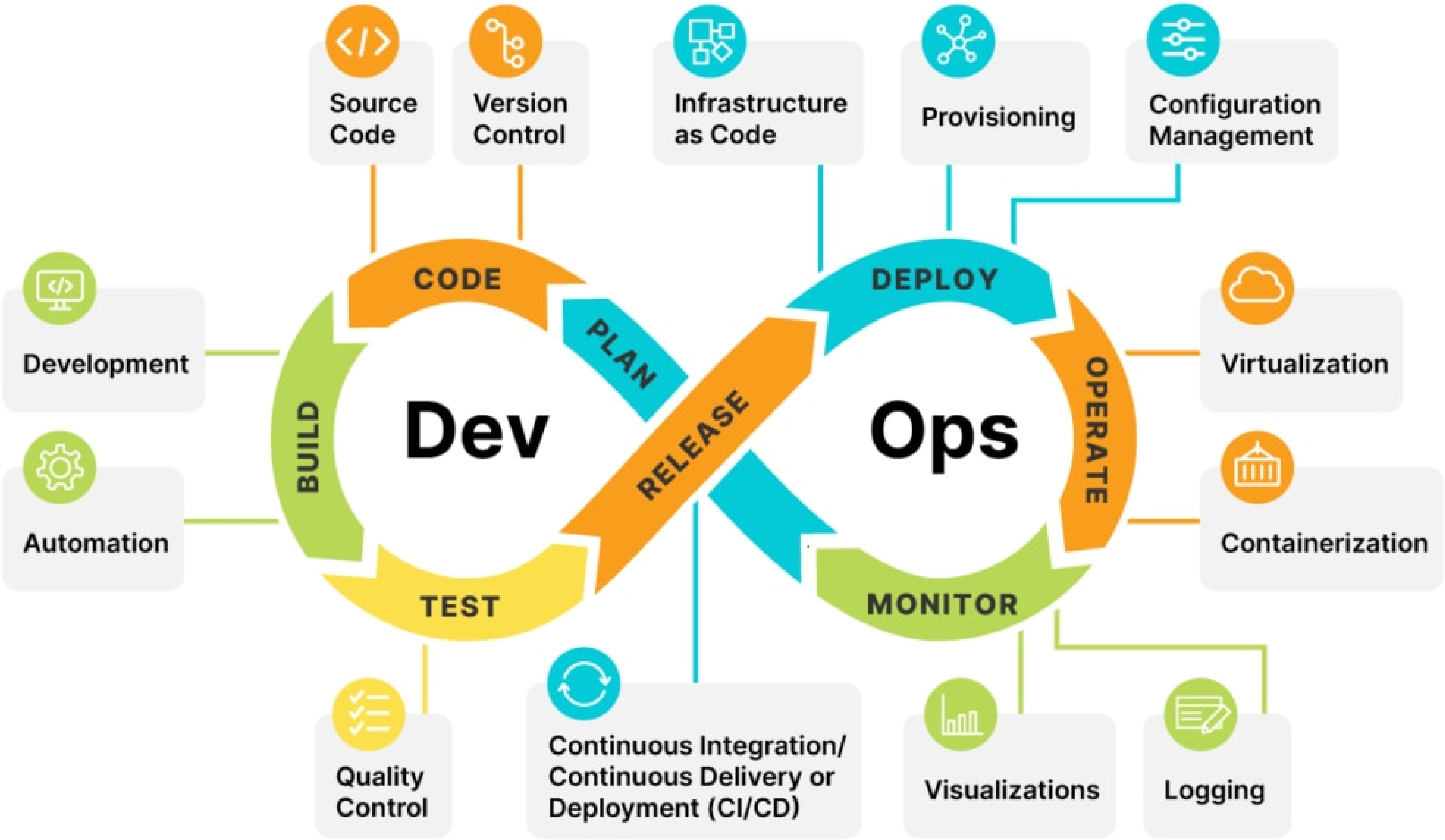
Gather a list of problems

Design Solutions ASAP

Test Solutions

Analyze and measure solutions

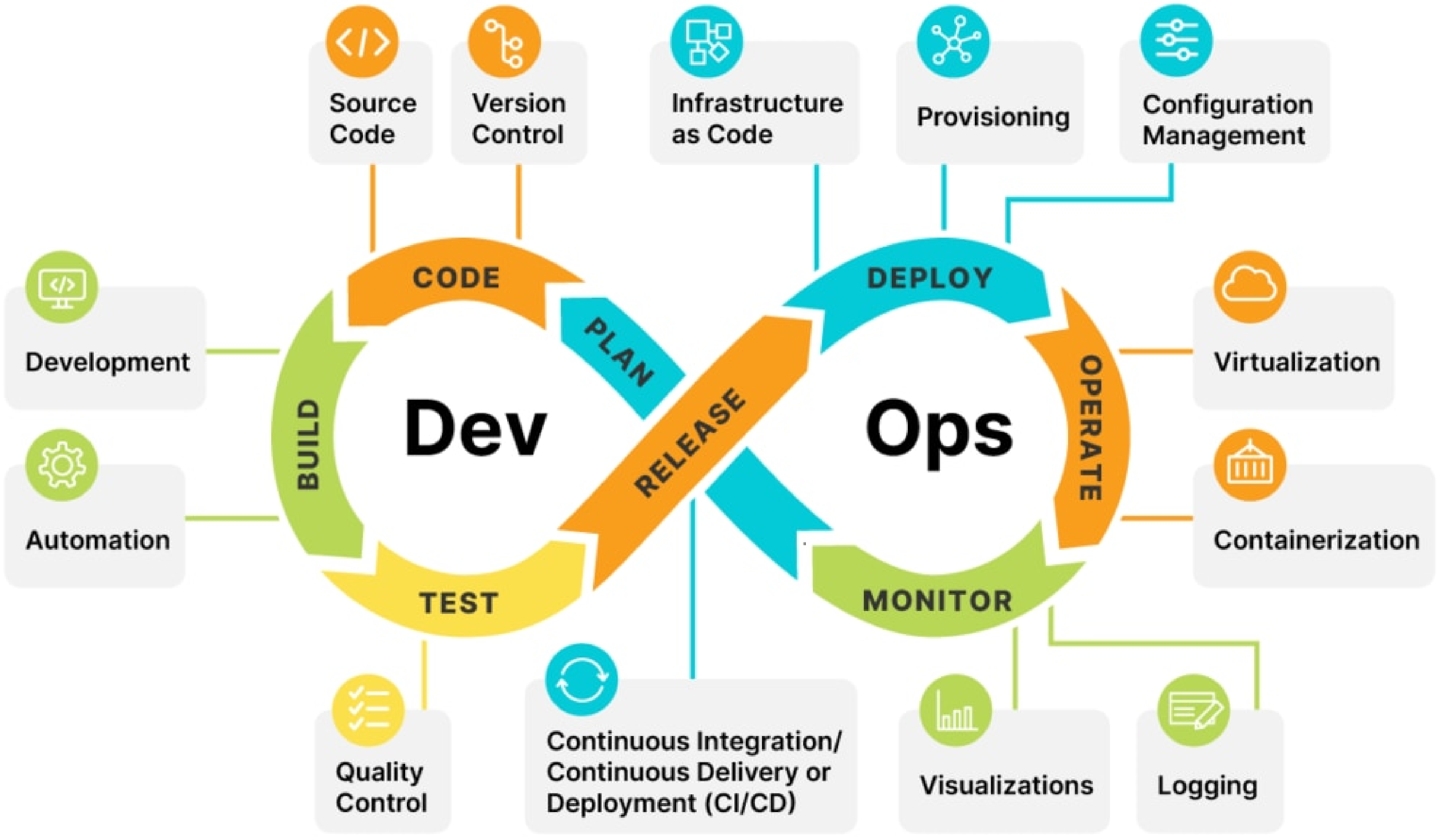
DevOps is the Kaizen of CS



[What Is DevOps? Complete Guide to Best Practices - Orange Matter]

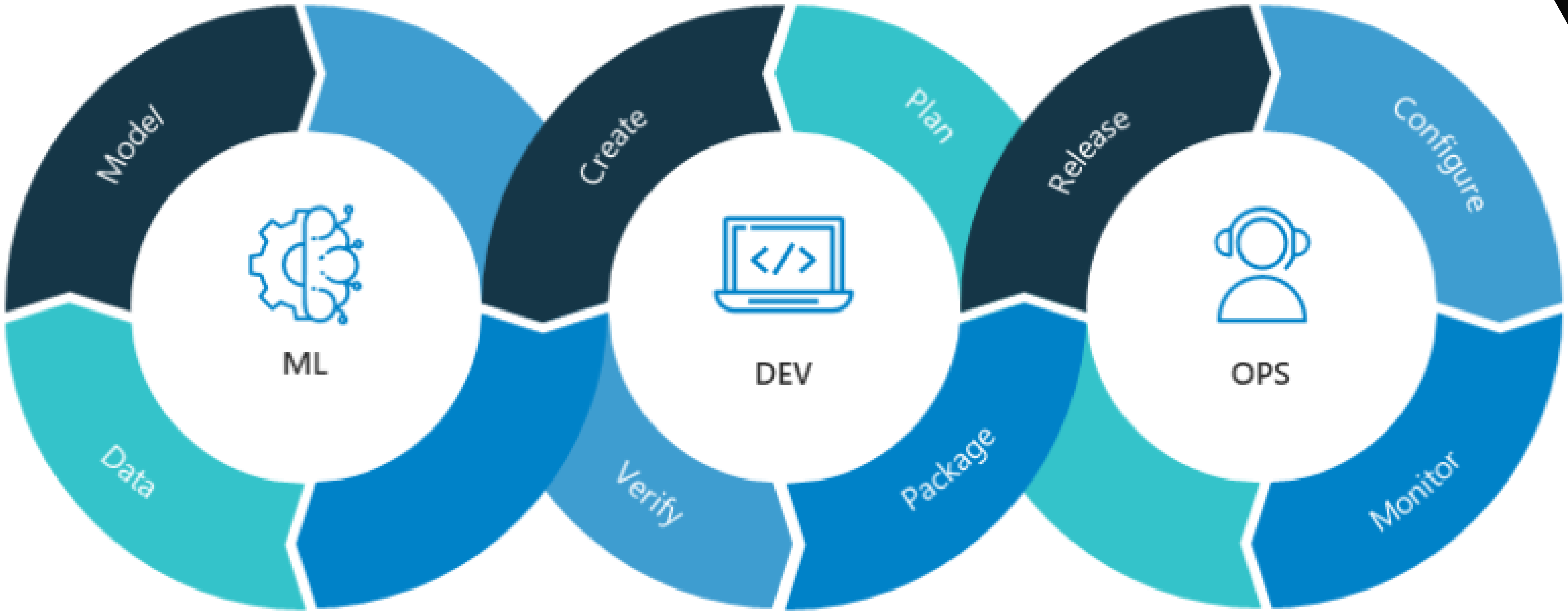
DevOps is the Kaizen of CS

- Big Techs deploy code 208 times more frequently because of good DevOps practices.
- And can recover from incidents 2,604 times faster than their low-performing counterparts.



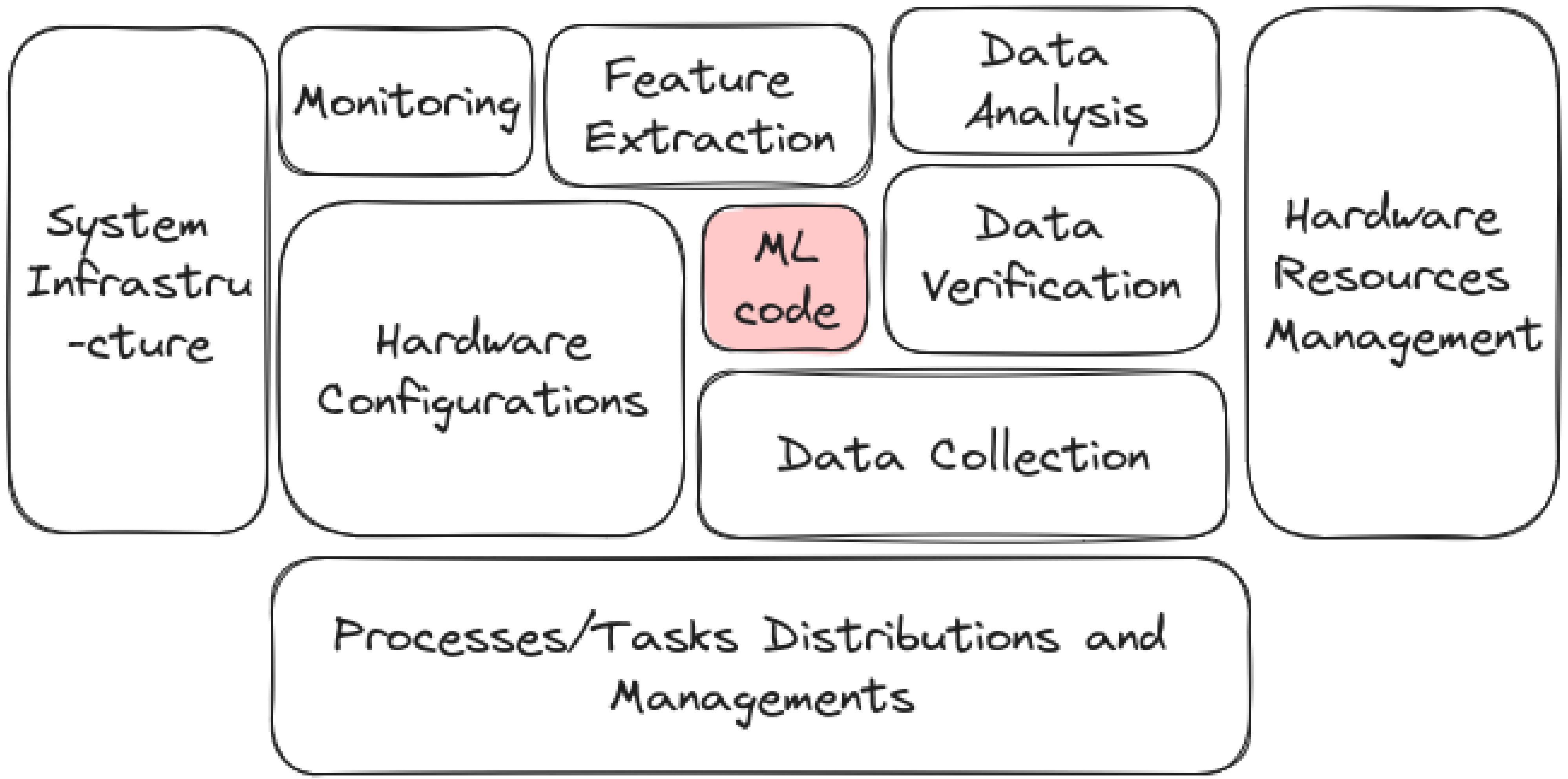
[What Is DevOps? Complete Guide to Best Practices - Orange Matter]

MLOps = ML + DevOps



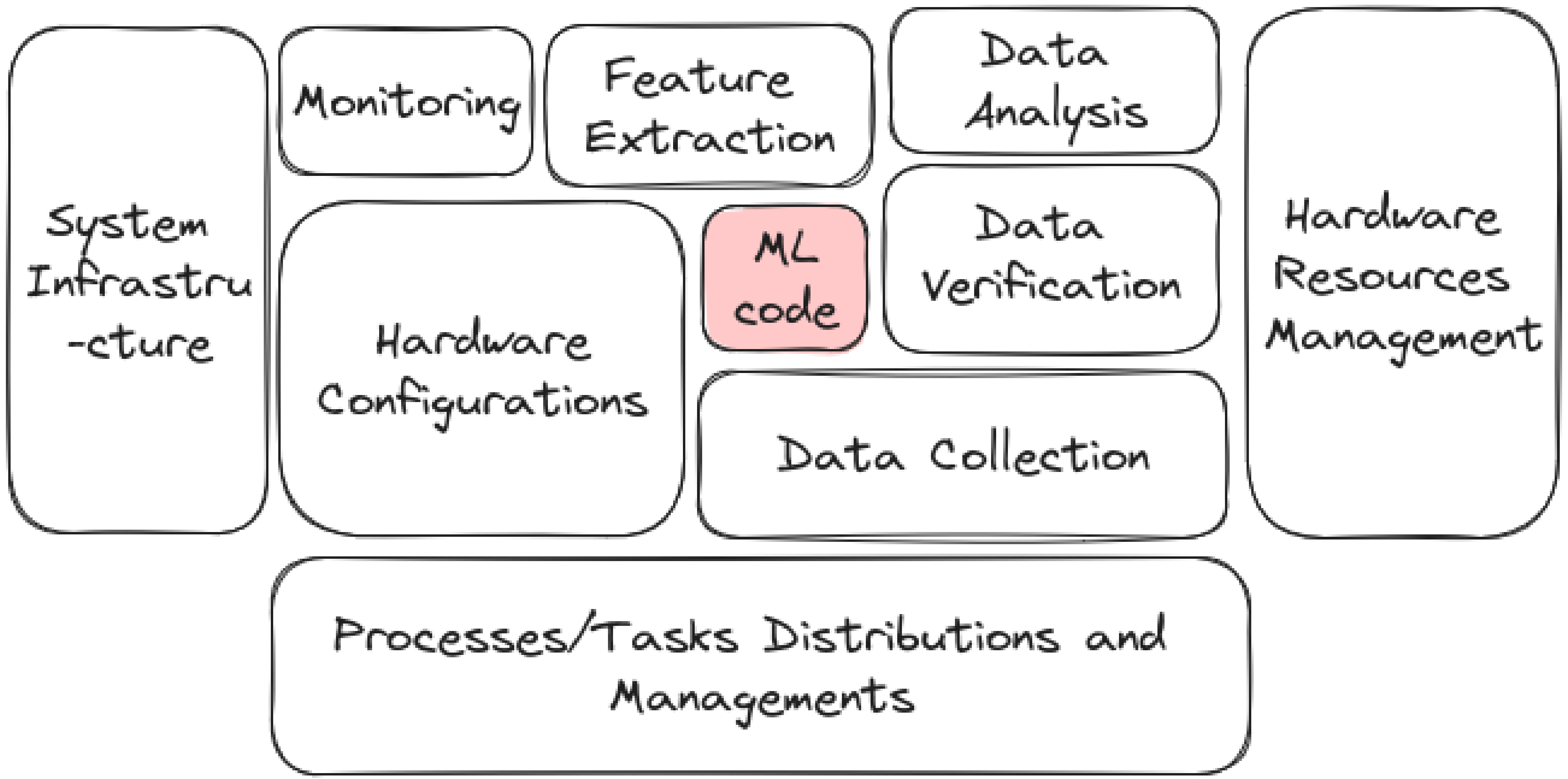
[What is MLOps? | NVIDIA Blog]

Creating a performant model is not enough !

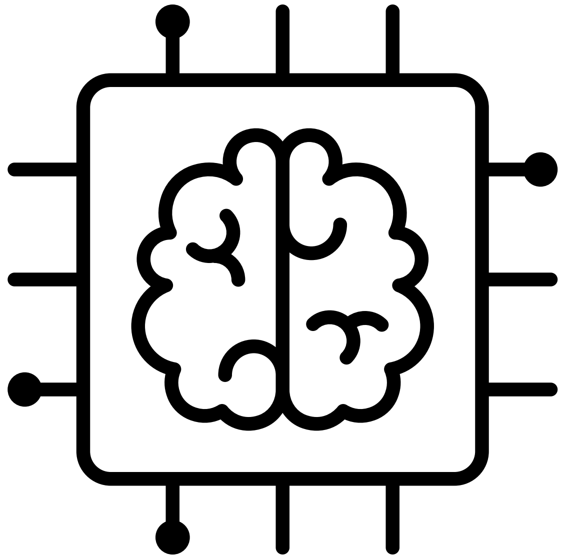


Energy spent in a Machine Learning Project

Creating a performant model is not enough !

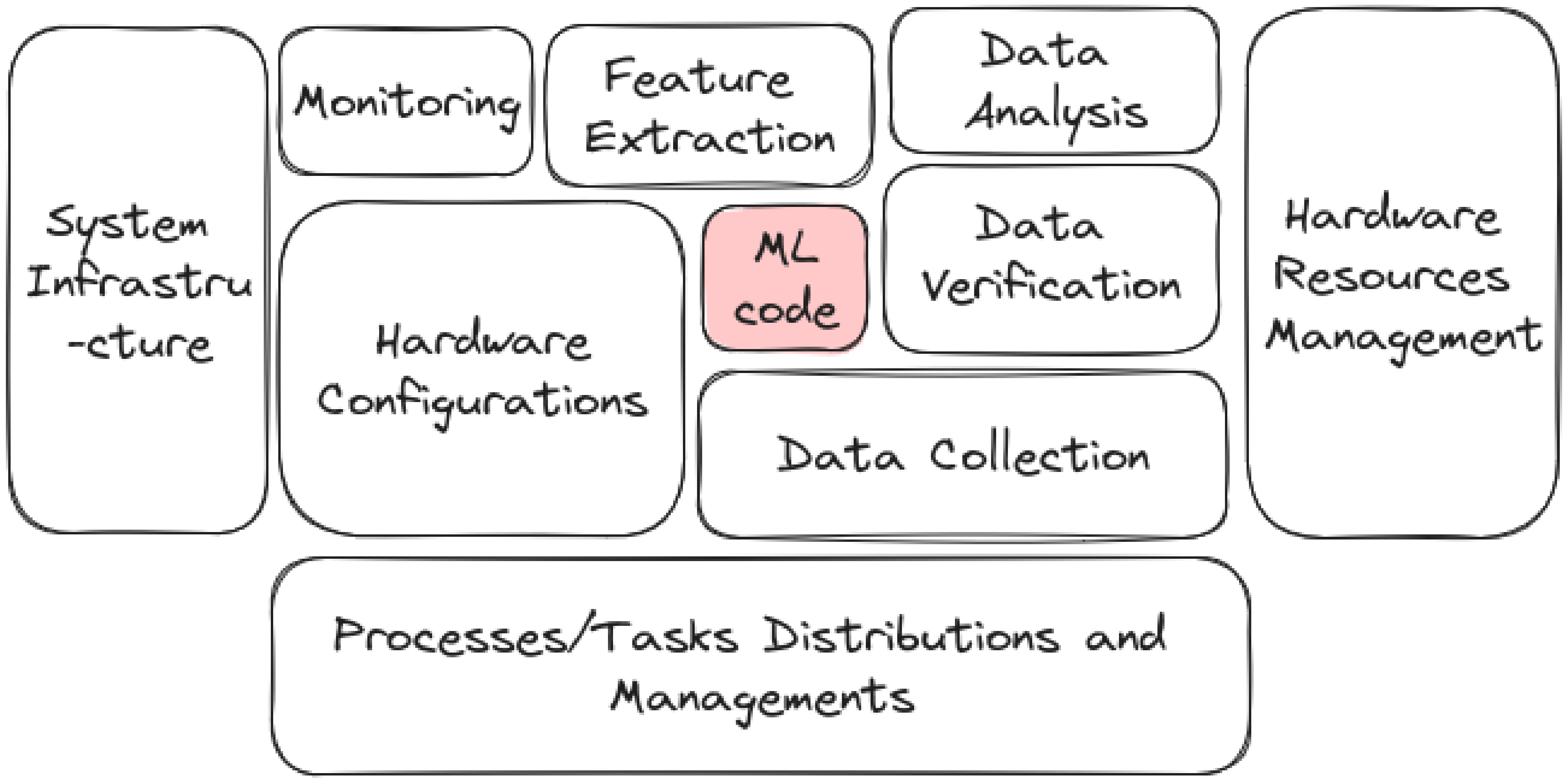


Energy spent in a Machine Learning Project



Business problem: distinguish between ripe and unripe Fruits

Creating a performant model is not enough !



Energy spent in a Machine Learning Project

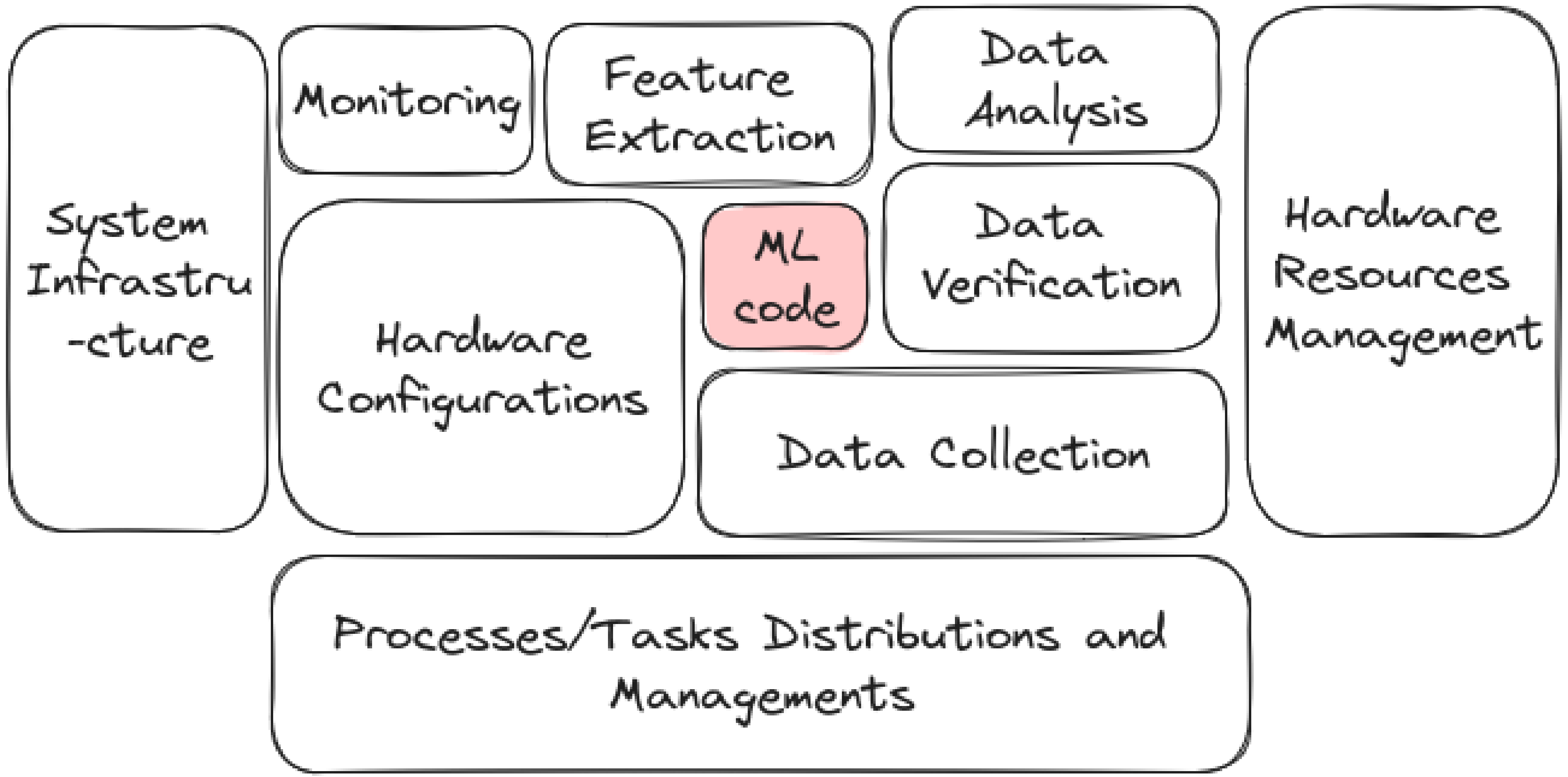
Business problem: distinguish between ripe and unripe Fruits

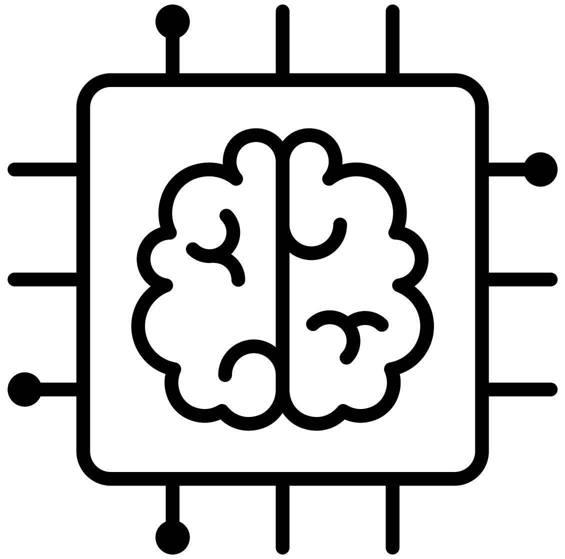
Collect Data

Choose an ML algorithm/Build an NN -> Train, Test and Validate

Performant Model

Creating a performant model is not enough !





Energy spent in a Machine Learning Project

Business problem: distinguish between ripe and unripe Fruits

Collect Data

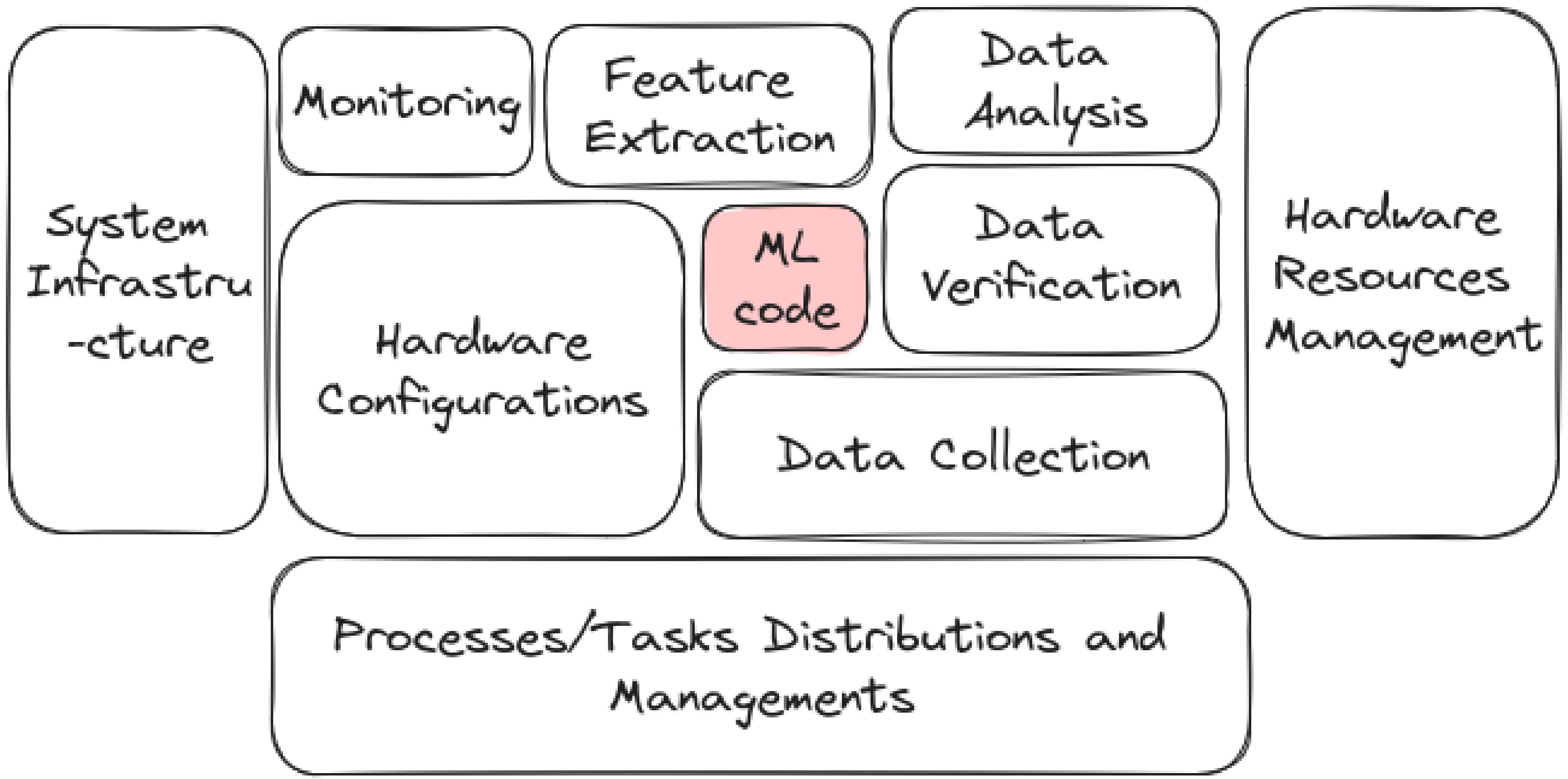
Choose an ML algorithm/Build an NN -> Train, Test and Validate

jupyter

Performant Model

Where is the result ?

Creating a performant model is not enough !



Energy spent in a Machine Learning Project

Access and use

API calls

Codespaces

Deployment Services

Deploy your Model in the cloud... or locally

Business problem: distinguish between ripe and unripe Fruits

Collect Data

Choose an ML algorithm/Build an NN -> Train, Test and Validate

Performant Model

You After Deploying models:

Cash

$$$

£££

Money

Money

Machine Learning Engineers Salary

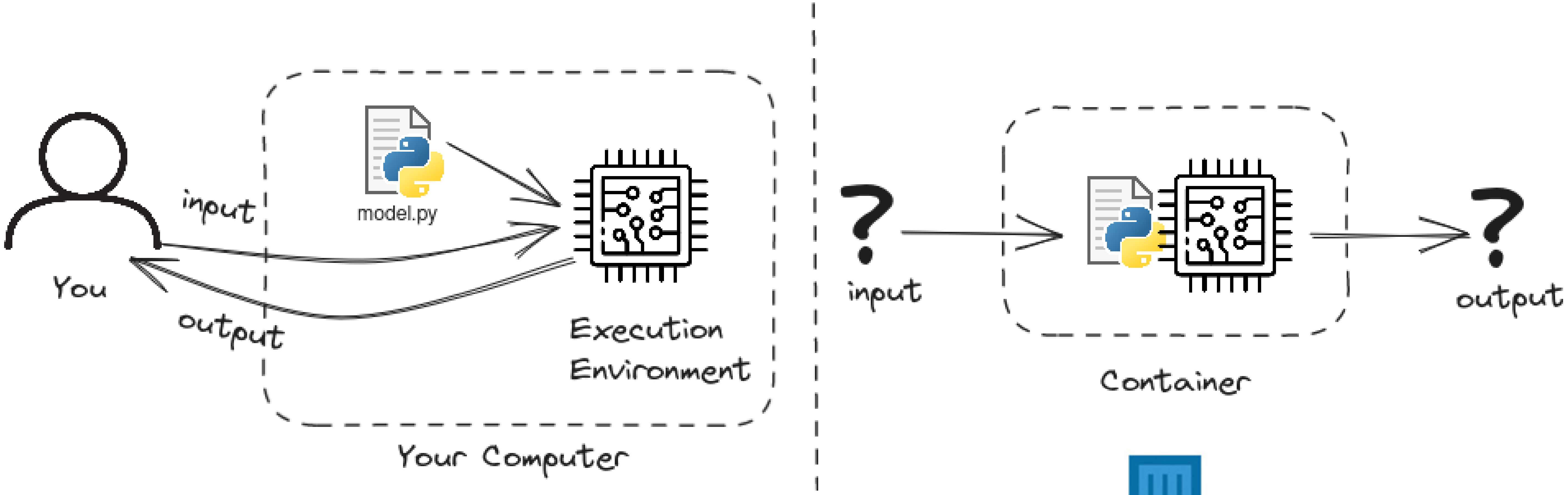
- Wall Street Journal: certified MLOps Engineers can expect salaries exceeding **$200,000**
- Glassdoor.com: The estimated total pay for a MLOps Engineer is **$124,698** per year in the United States area.

Machine Learning Engineer Salary in USA, from salary.com

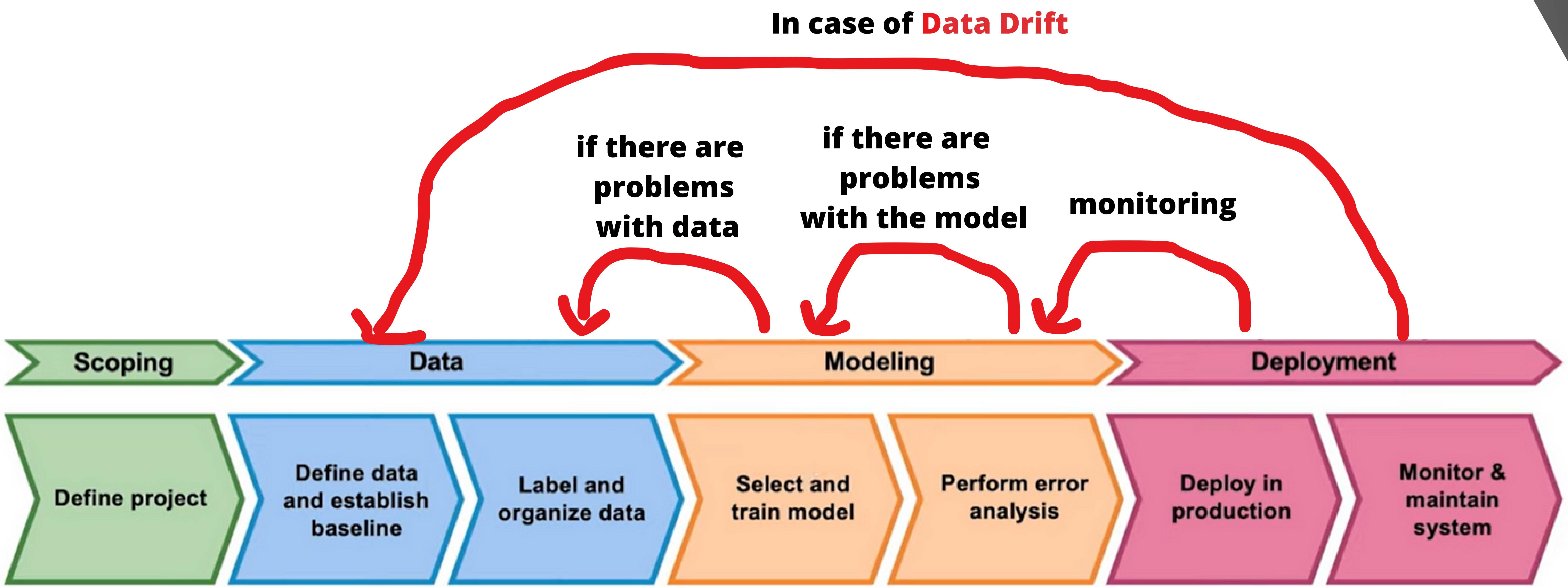
- Indeed.com: certified Data Scientist actually are paid $8,856 per month and $100,160 per year on average

Practice Time !

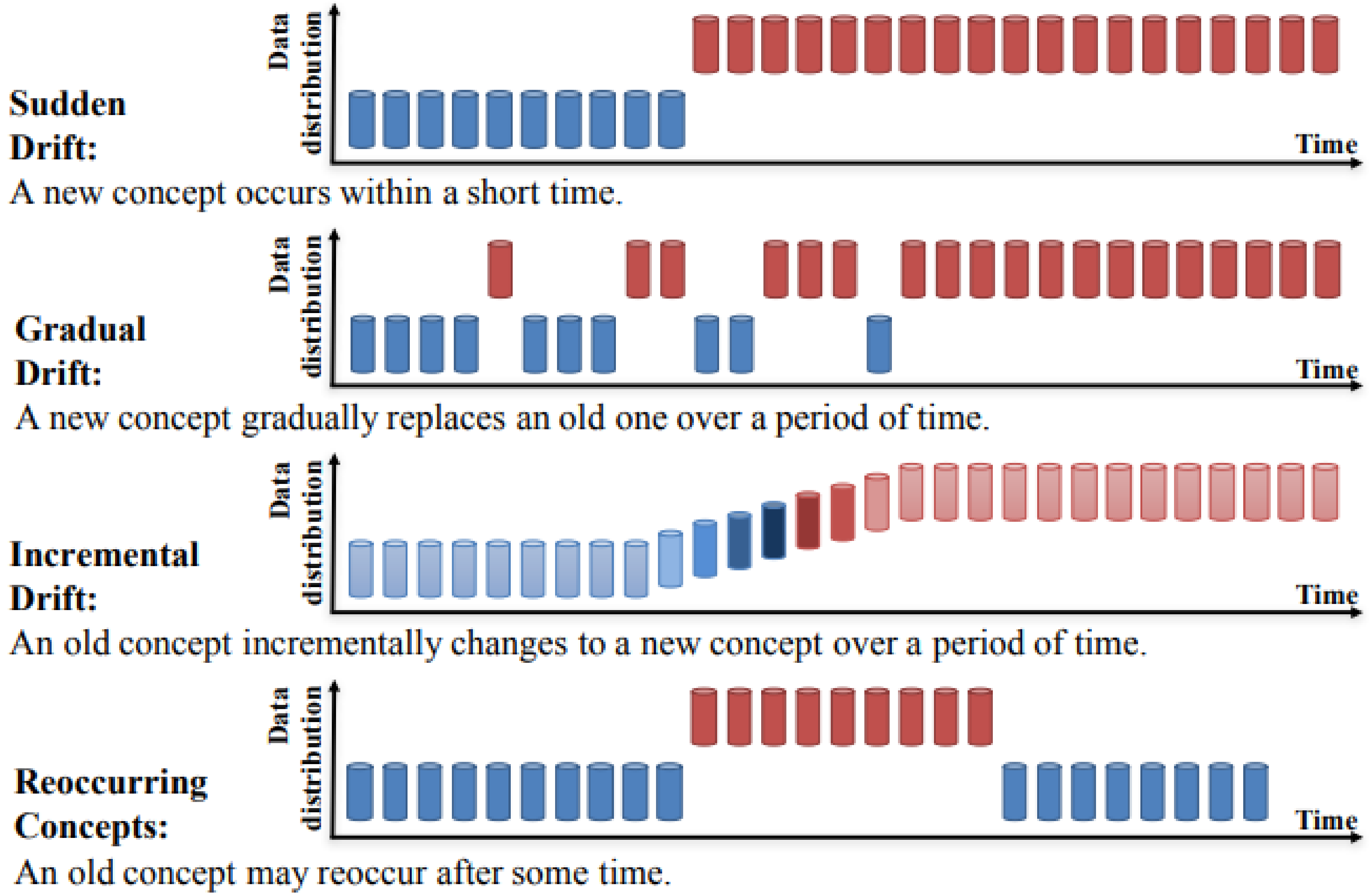
DEPLOYING ML MODELS USING DOCKER AND FLASK

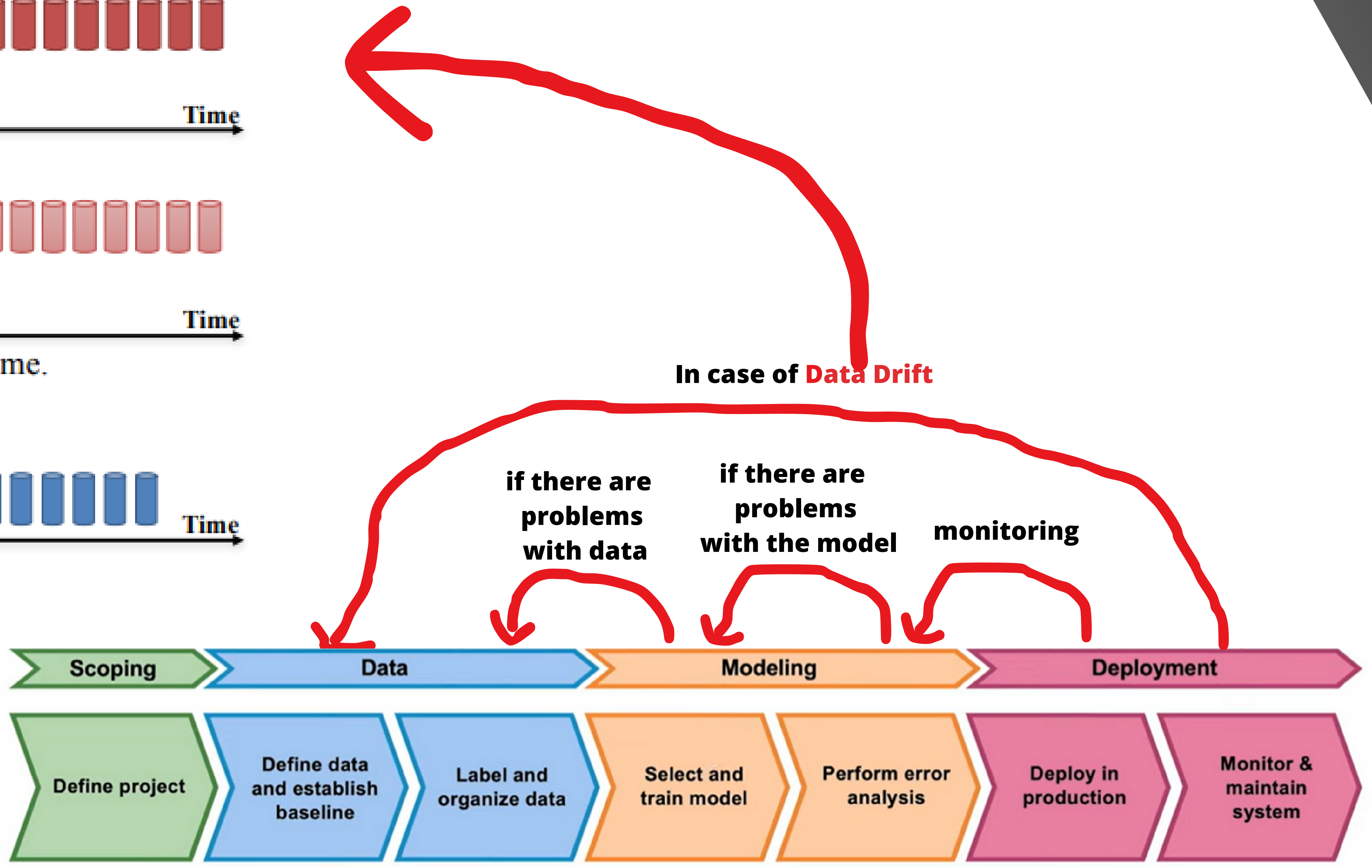


Summary of the ML Pipeline

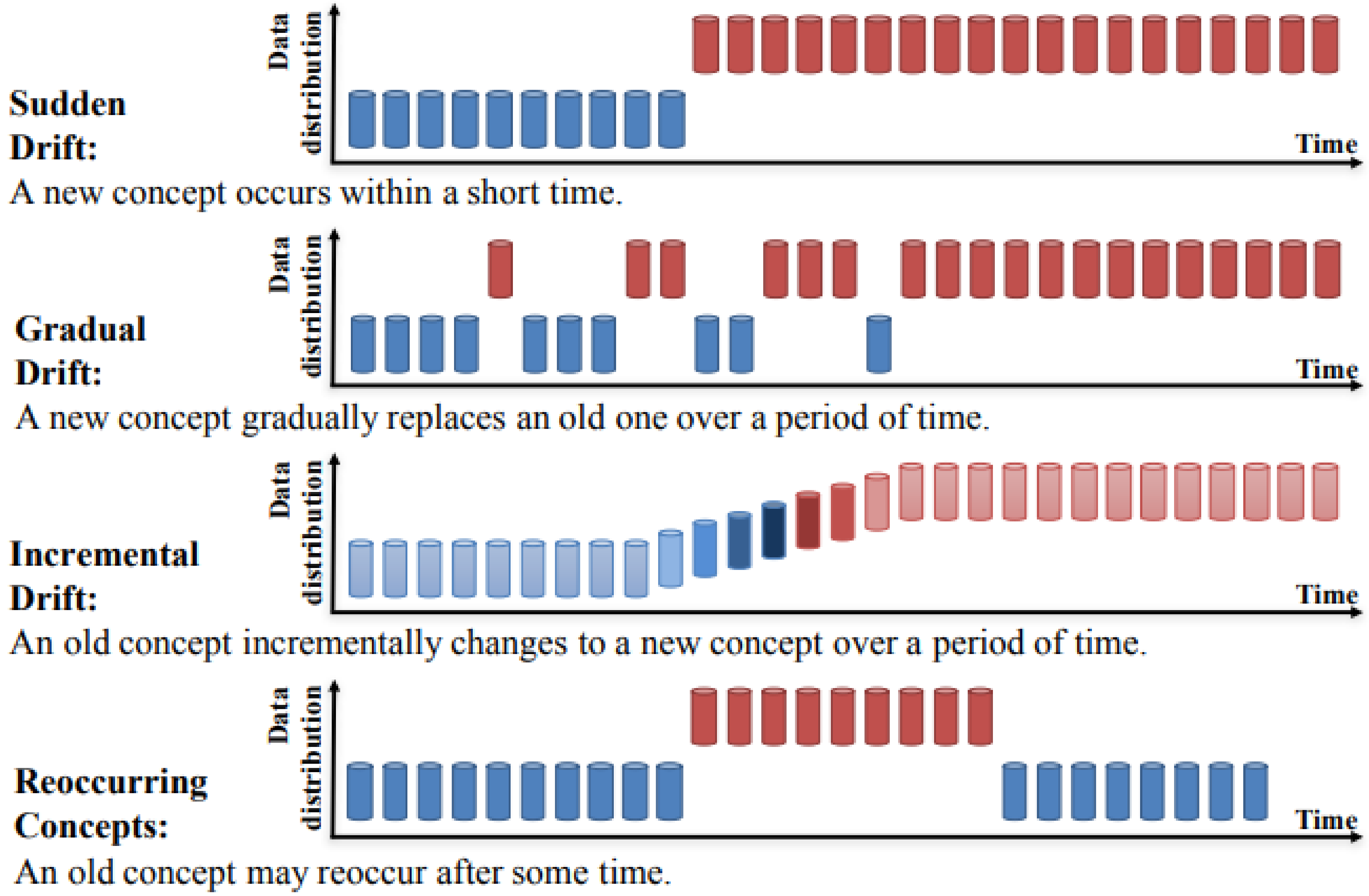


Data Drift is a big Problem in Production

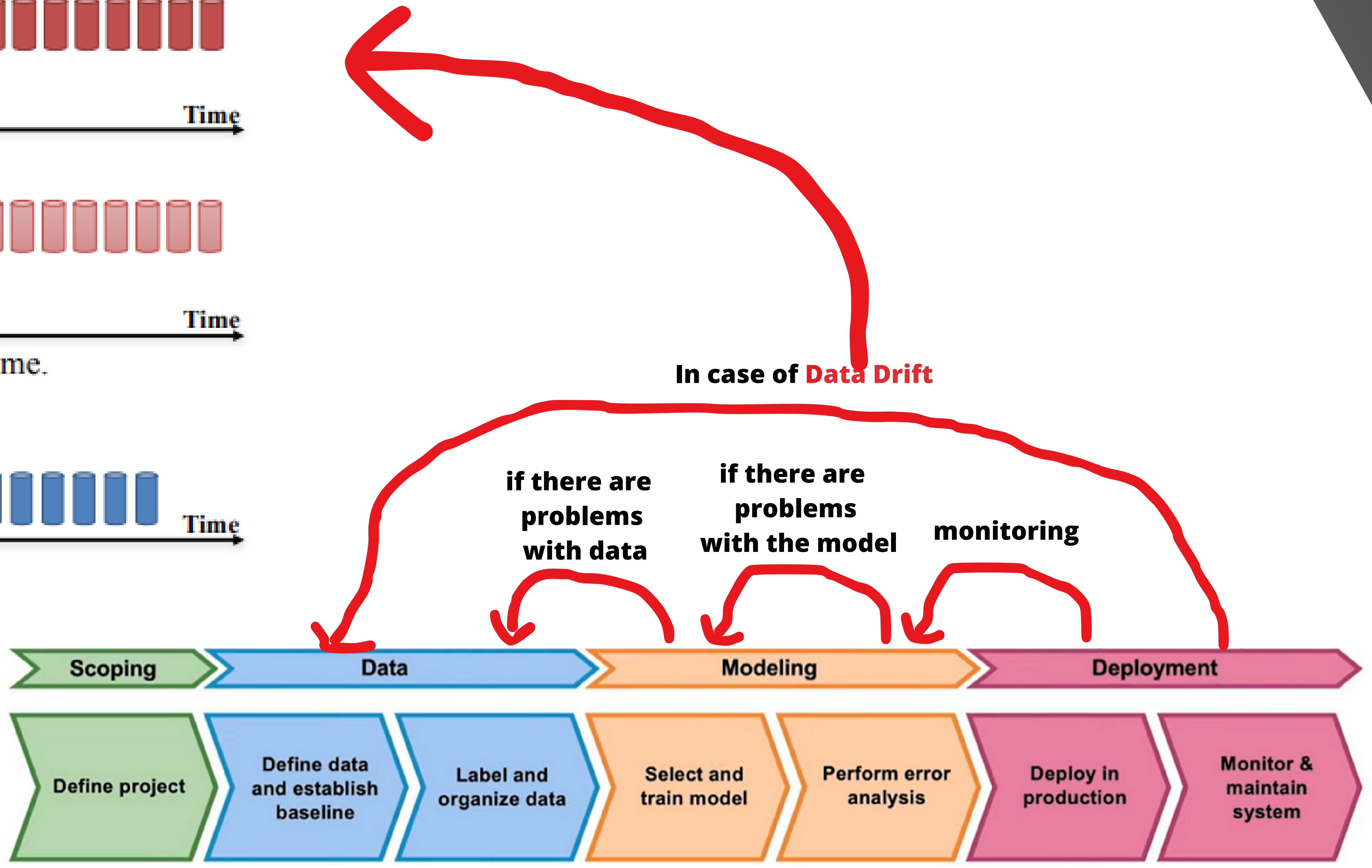




Data Drift is a big Problem in Production



Example: Fraudulent People always come up with new strategies !

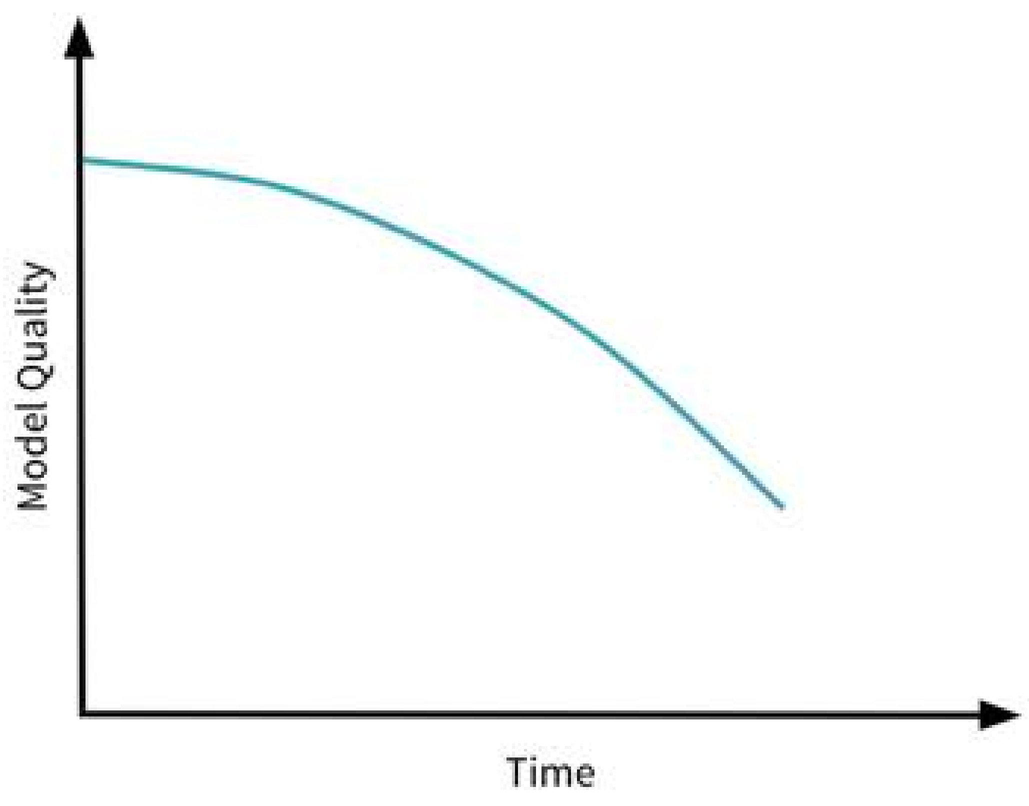


Example: COVID19 made people stay at home and their preferences changed !

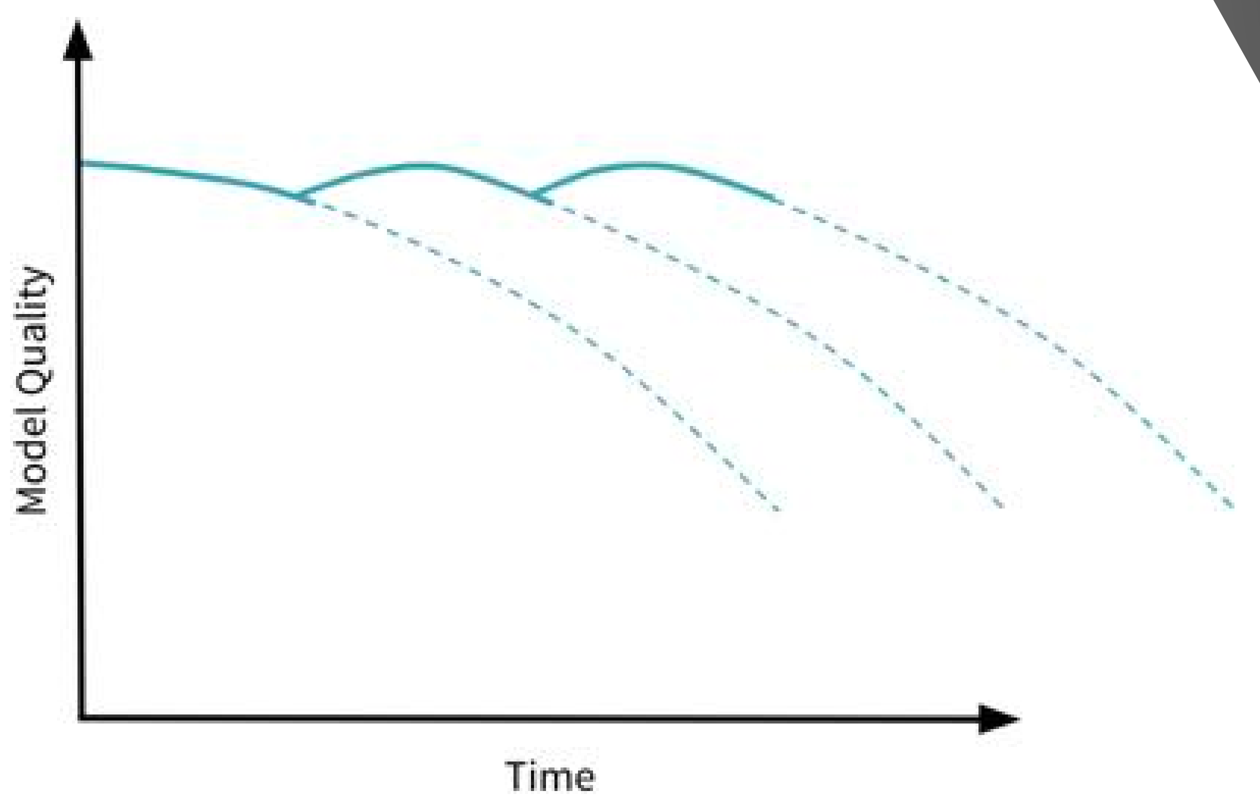
ML Models needs to be refreshed !

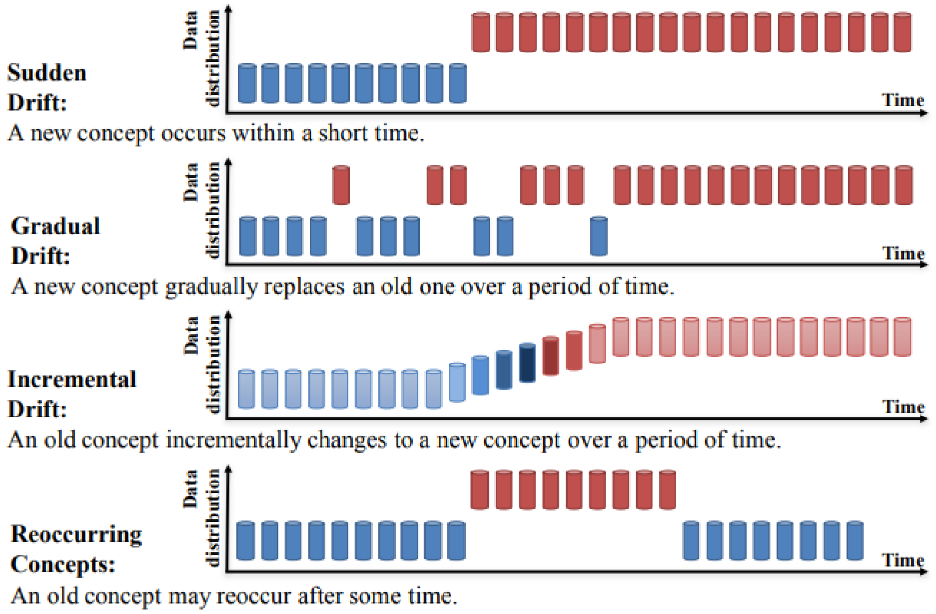
Models are in constant decay over time !

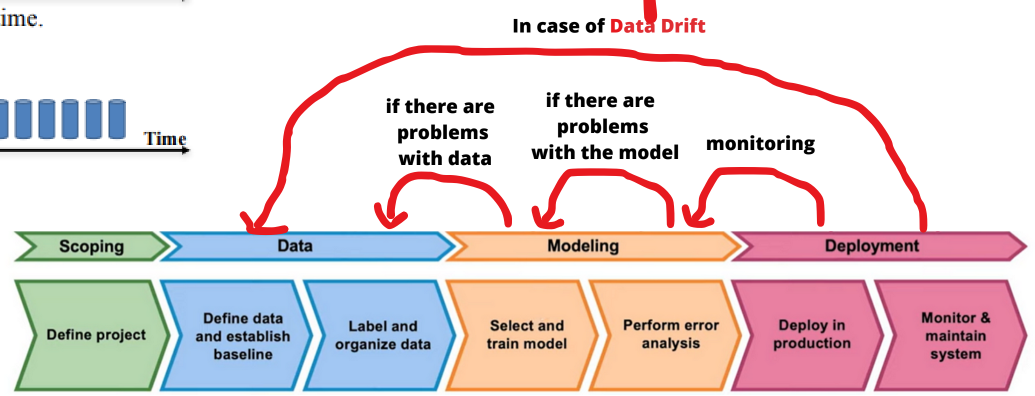
Static models



Refreshed models





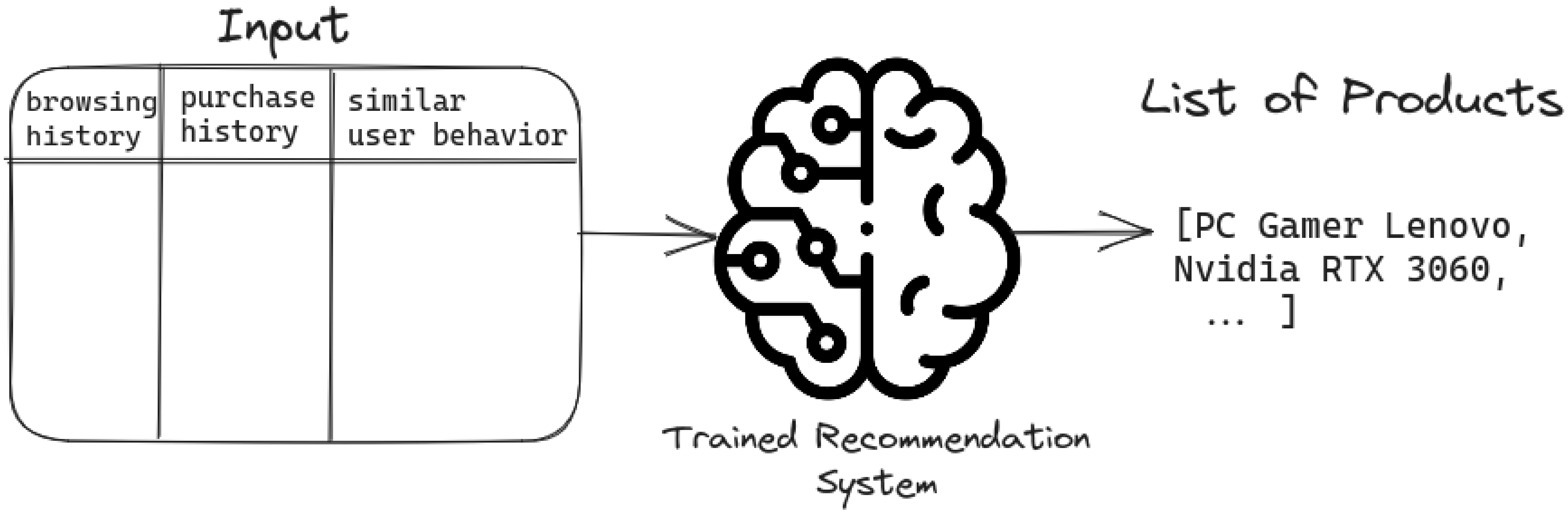


Solution: Refresh the data and the model design !

Monitoring an E-commerce Platform Recommendation System !

Objective: know when the model becomes bad !

Monitoring Metric = CTR



- **Problem: clicking on a product doesn't mean buying it -> Solution: change the metric to the ratio of clicks that lead to a purchase**
- **Problem: in winter, recommending only cold clothes is a good sign -> Solution: We should also add time as a monitoring metric**
- **Problem: in some cases accuracy is a misleading metric ! -> Solution: We should add recall and precision also as monitoring metric**

How to know my model is becoming bad ?

Bad = not accurate !

(but also maybe slow)

How to know my model is becoming bad ?

Maybe it's bad because the input data is bad also !

Bad = not accurate !
(but also maybe slow)

How to know my model is becoming bad ?

Bad = not accurate !
(but also maybe slow)

Maybe it's bad because the input data is bad also !

So what to monitor exactly ? inputs, features, time ... etc

How to know my model is becoming bad ?

Bad = not accurate !
(but also maybe slow)

Maybe it's bad because the input data is bad also !

So what to monitor exactly ? inputs, features, time ... etc

Brainstorming sessions are needed, to choose the right metrics to monitor !

How to know my model is becoming bad ?

Bad = not accurate !
(but also maybe slow)

Maybe it's bad because the input data is bad also !

So what to monitor exactly ? inputs, features, time ... etc

Brainstorming sessions are needed, to choose the right metrics to monitor !

Now I have them, how to monitor ?

How to know my model is becoming bad ?

Bad = not accurate !
(but also maybe slow)

Maybe it's bad because the input data is bad also !

So what to monitor exactly ? inputs, features, time ... etc

Brainstorming sessions are needed, to choose the right metrics to monitor !

Now I have them, how to monitor ?

Use the correct tools and interpret correctly !

How to know my model is becoming bad ?

Bad = not accurate !
(but also maybe slow)

Maybe it's bad because the input data is bad also !

So what to monitor exactly ? inputs, features, time ... etc

Brainstorming sessions are needed, to choose the right metrics to monitor !

Now I have them, how to monitor ?

Use the correct tools and interpret correctly !

And then ?

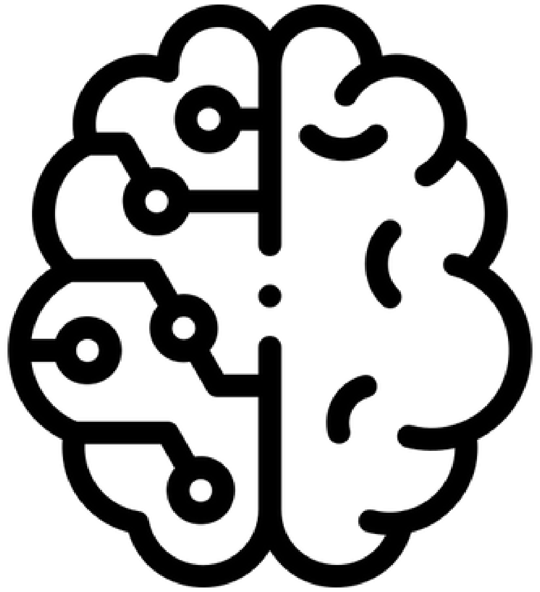
Set Alarms and Triggers !

Adressing the Problem after Detecting it:

Data Engineering Enhancement
- enhance data cleaning
- enhance feature engineeri-ng

Model Retraining
- using new data
- using augmented datasets

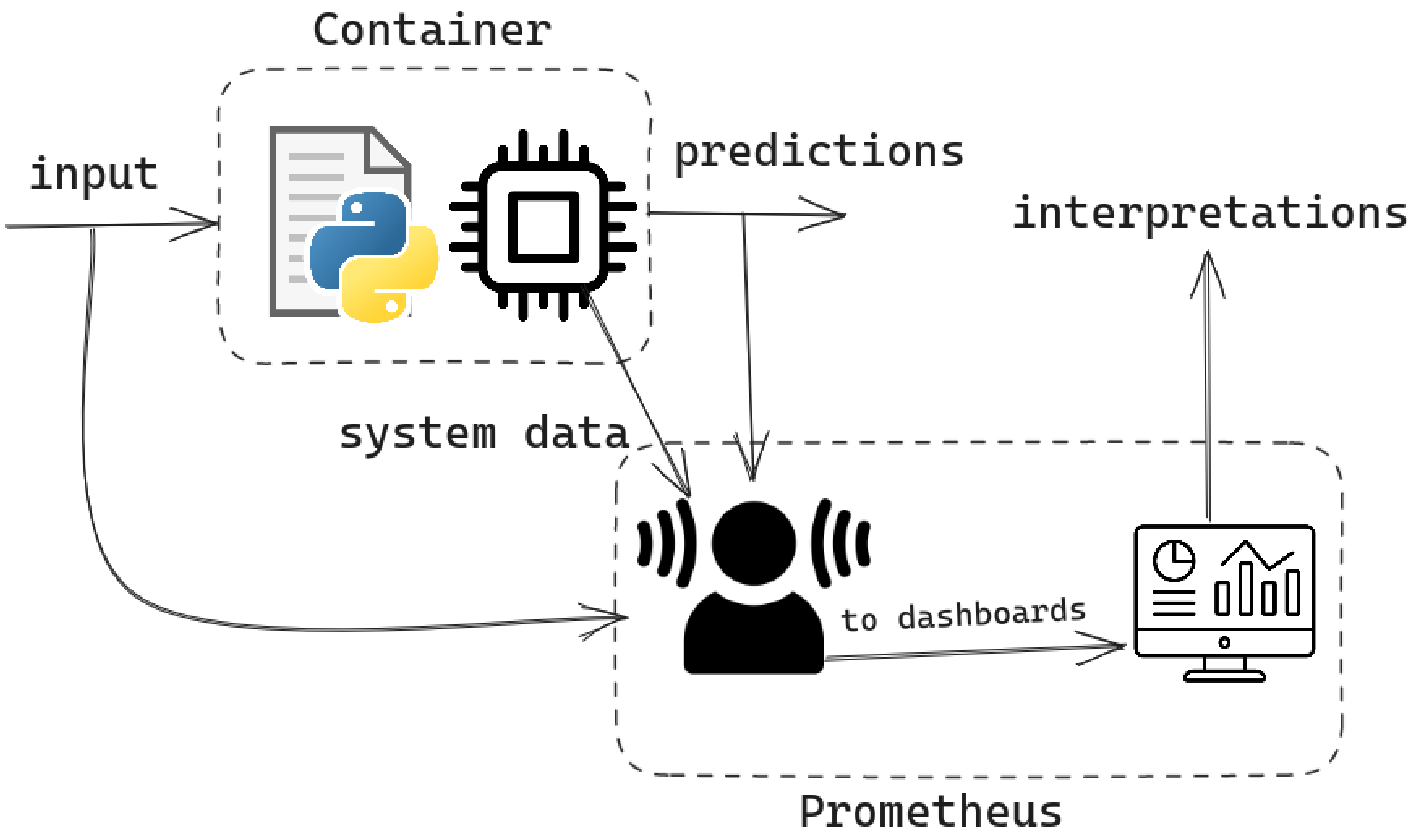
Model Refactoring
- tweaking the architect-ure
- using a new architect-ure



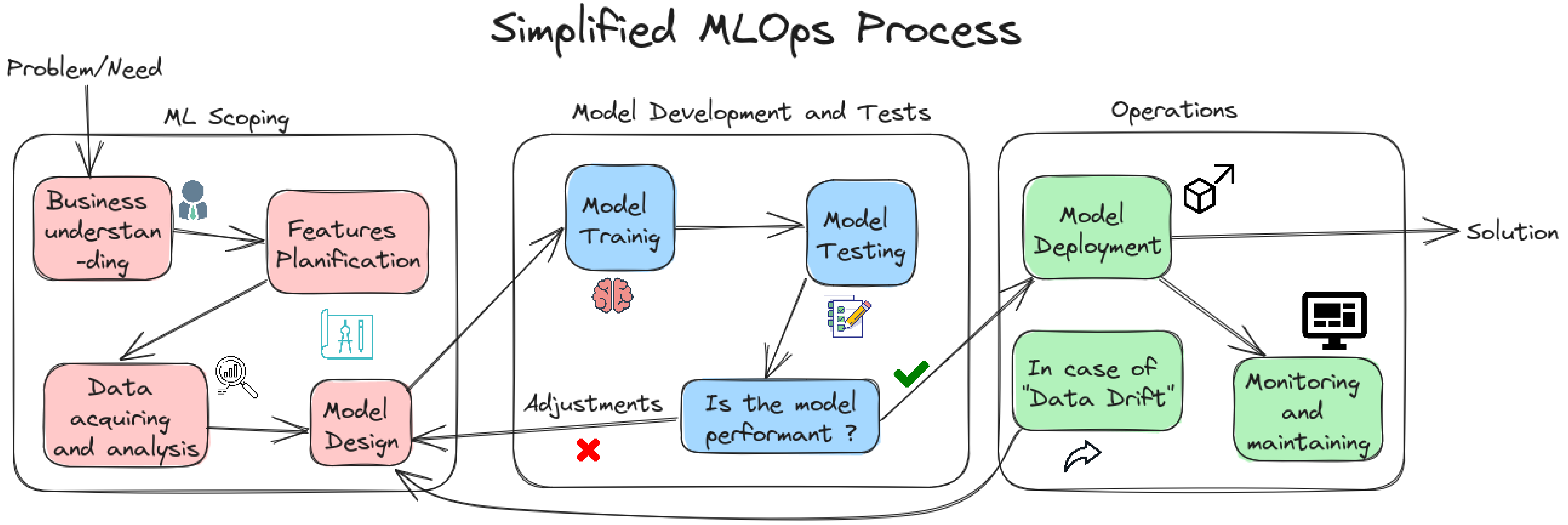
Continuous Feedback Loop
- human continuous feedback
- model continuous feedback

Practice Time !

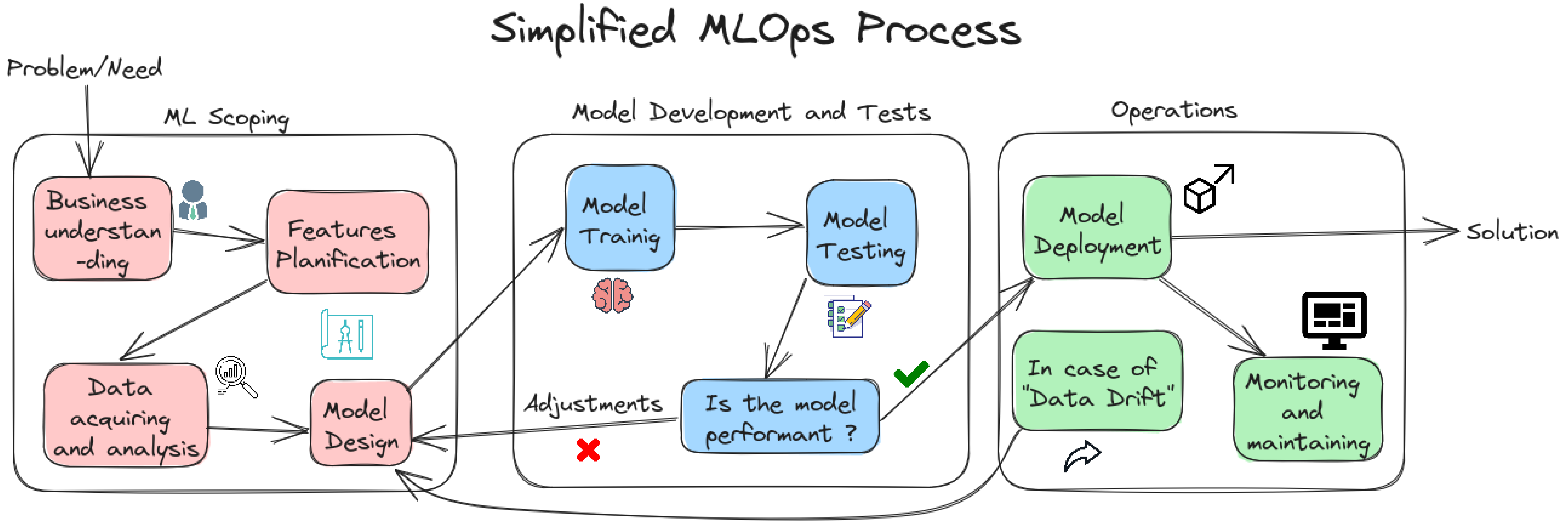
MONITORING A DEPLOYED ML MODEL USING PROMETHEUS



What if we automate things ? Manual (Level 0 of MLOps) is boring and takes a lot of time + resources !

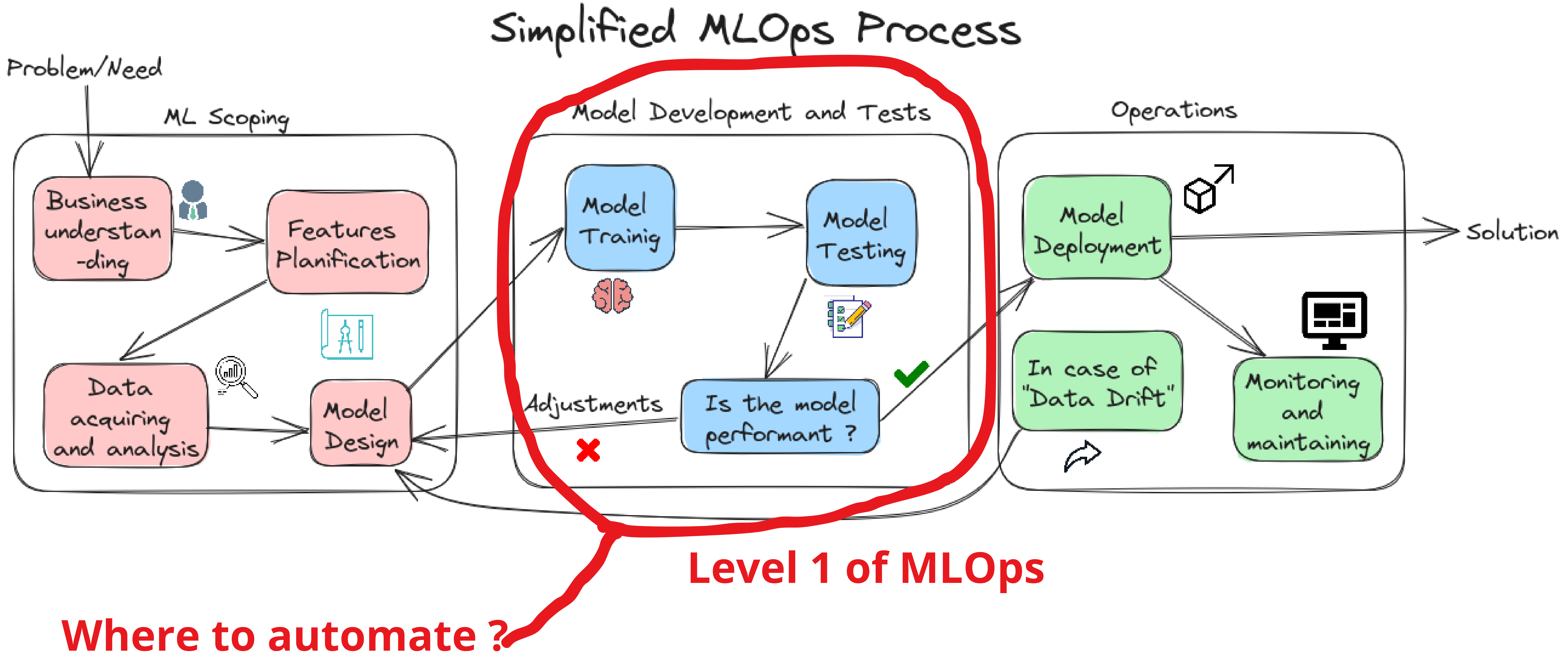


What if we automate things ? Manual (Level 0 of MLOps) is boring and takes a lot of time + resources !

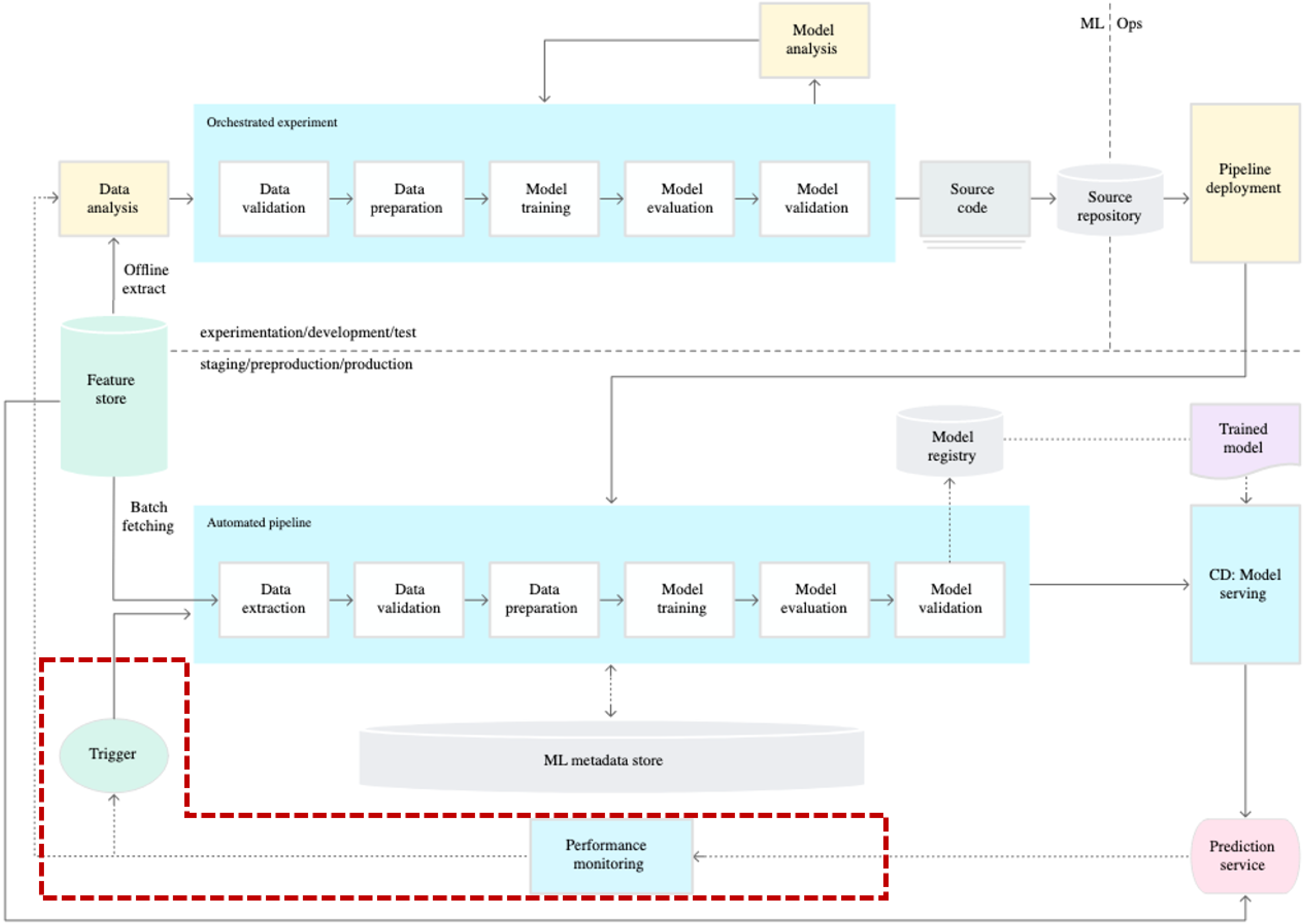


Where to automate ?

What if we automate things ? Manual (Level 0 of MLOps) is boring and takes a lot of time + resources !

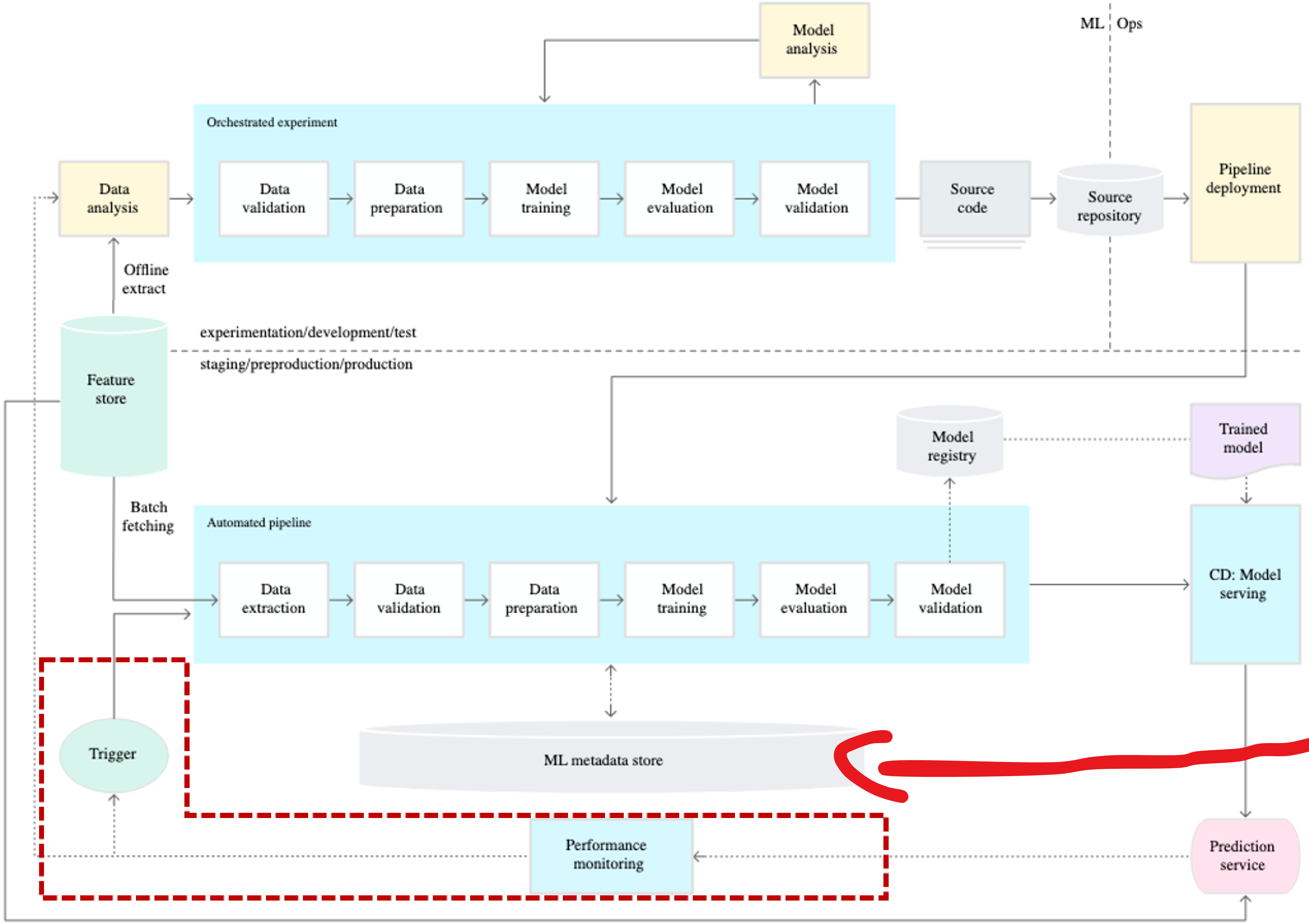


Automating the Learning (Continuous Learning)



mlops-for-all.github.io Level 2 of MLOps (Automating ML)

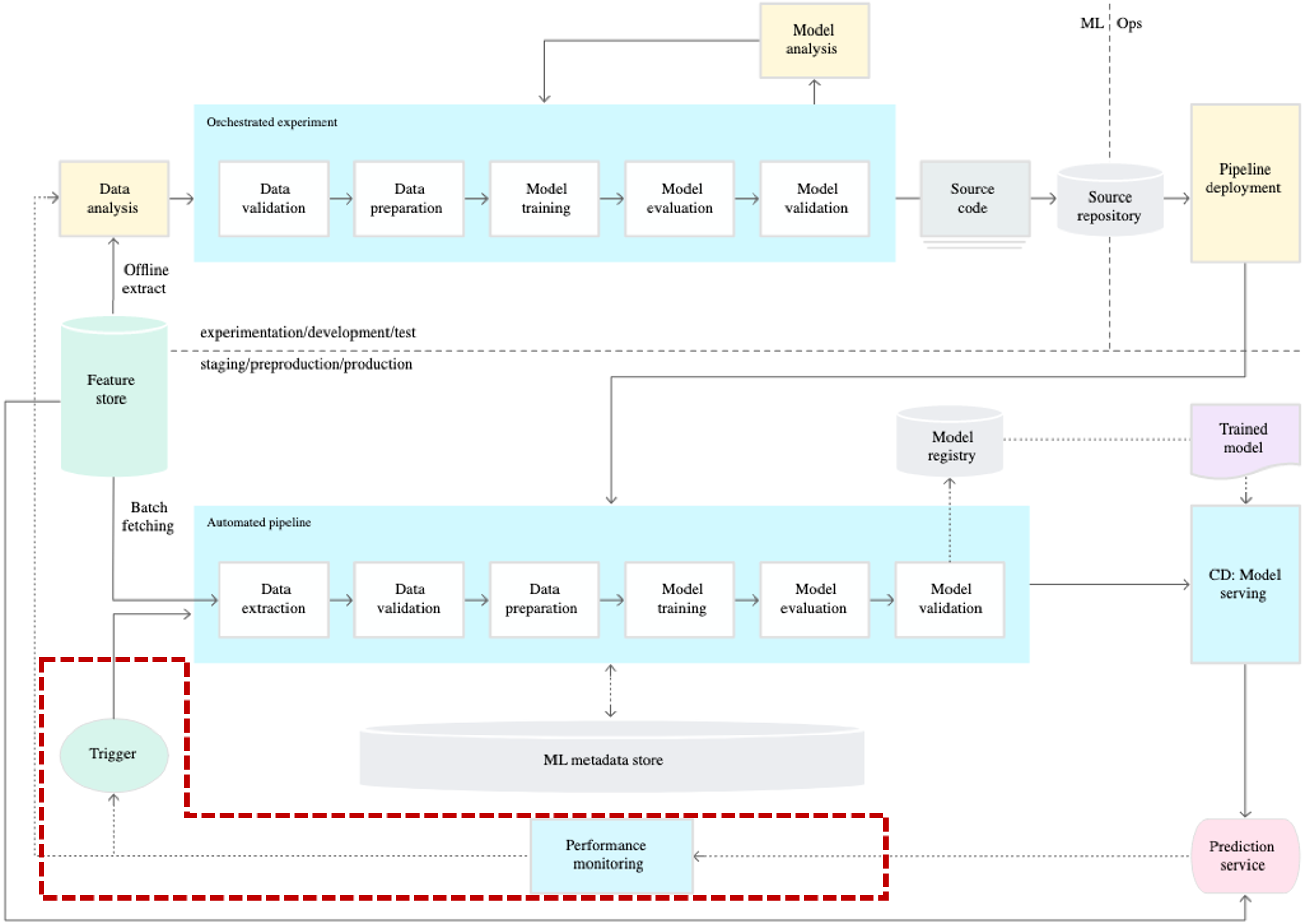
Automating the Learning (Continuous Learning)



What if the Data shifts, and then returns back to normal ??!

mlops-for-all.github.io Level 2 of MLOps (Automating ML)

Automating the Learning (Continuous Learning)



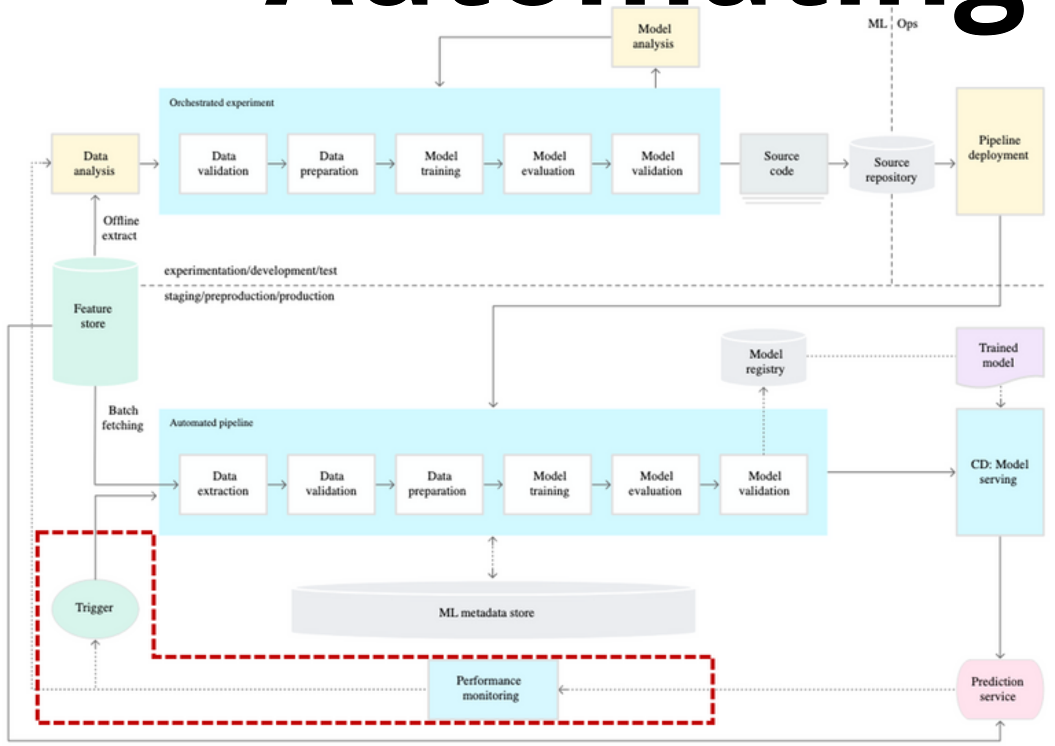
In an e-commerce scenario:

Issues:

- Mismatch in product recommendations across platform and emails.
- Teams duplicating feature engineering efforts.
- Delayed updates lead to irrelevant promotional emails.
- Difficulties in recreating a model's specific features set at a given time.

mlops-for-all.github.io Level 2 of MLOps (Automating ML)

Automating the Learning (Continuous Learning)



mlops-for-all.github.io Level 2 of MLOps (Automating ML)

In an e-commerce scenario:
Issues:

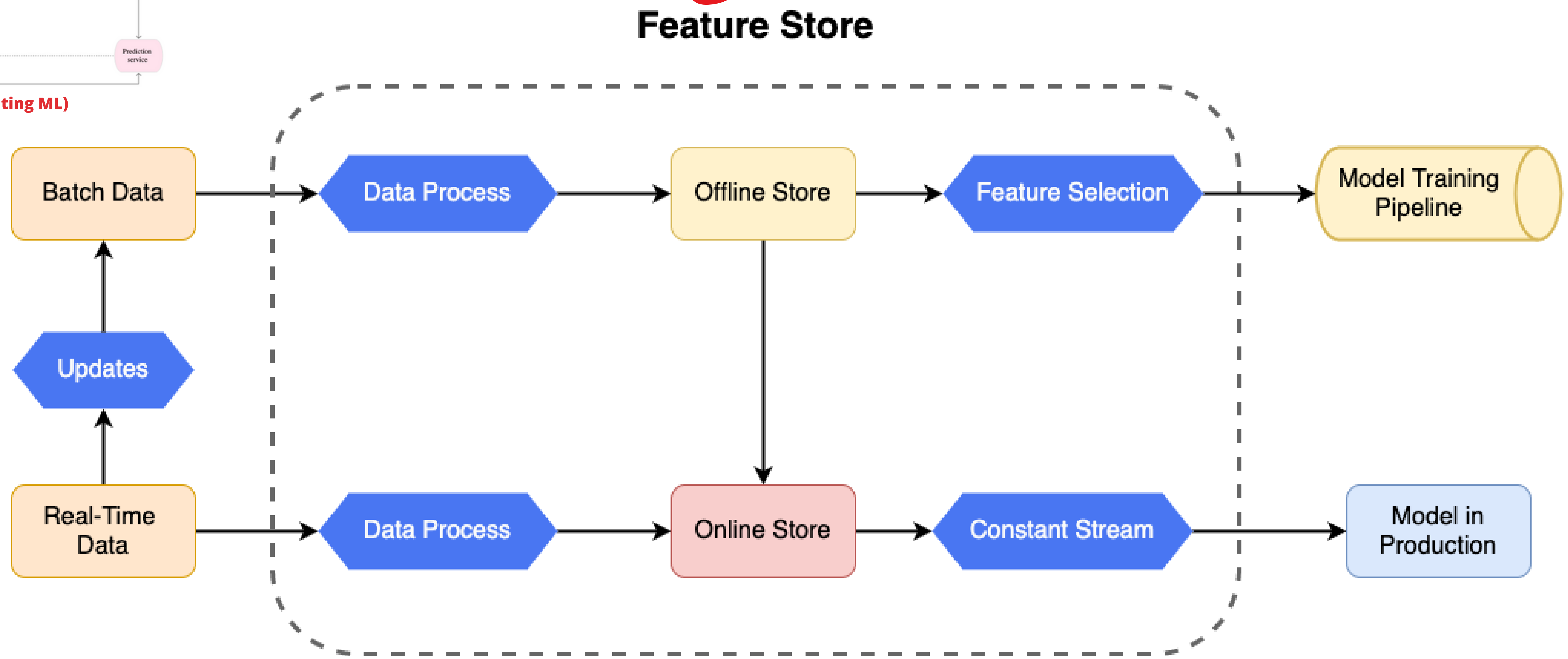
- Mismatch in product recommendations across platform and emails.
- Teams duplicating feature engineering efforts.
- Delayed updates lead to irrelevant promotional emails.
- Difficulties in recreating a model's specific features set at a given time.

Solution for:

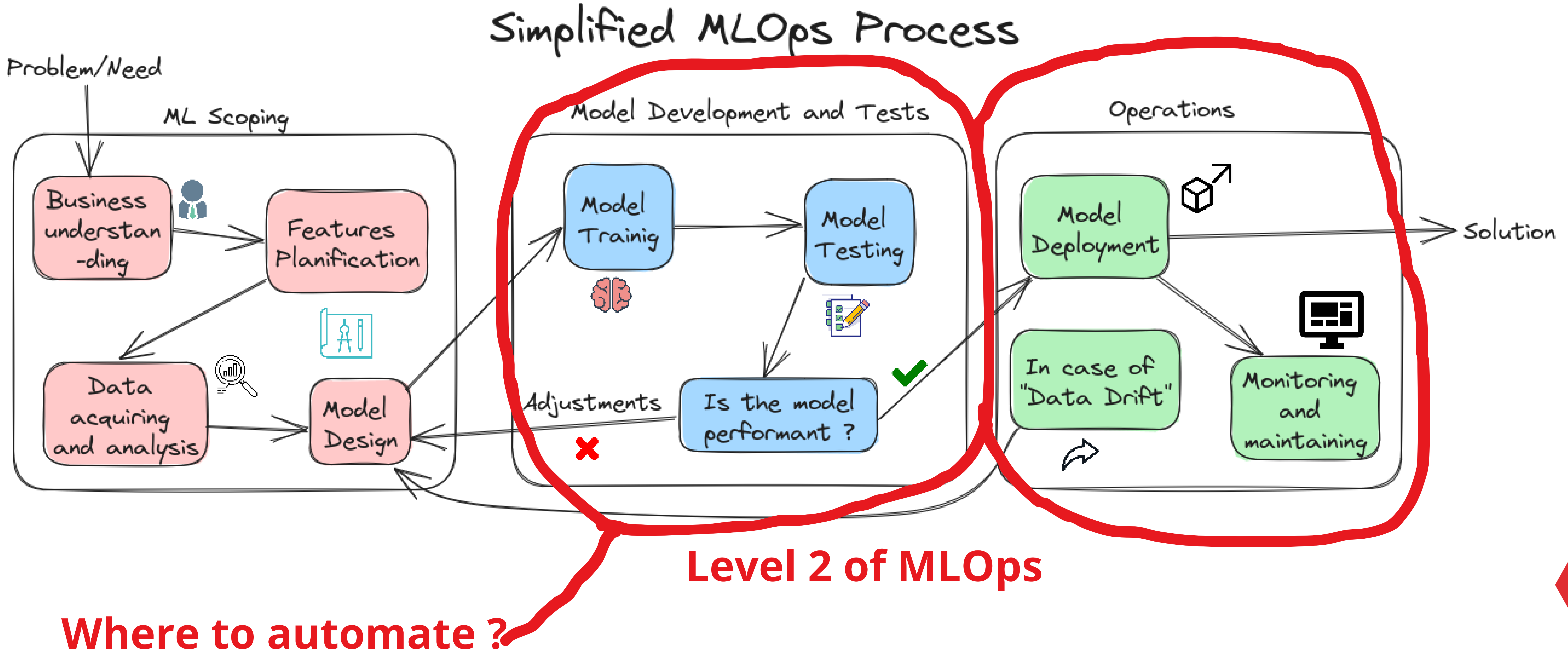
- **Feature Consistency**
- **Feature Reusability**
- **Feature Versioning**

From data lakes or warehouses

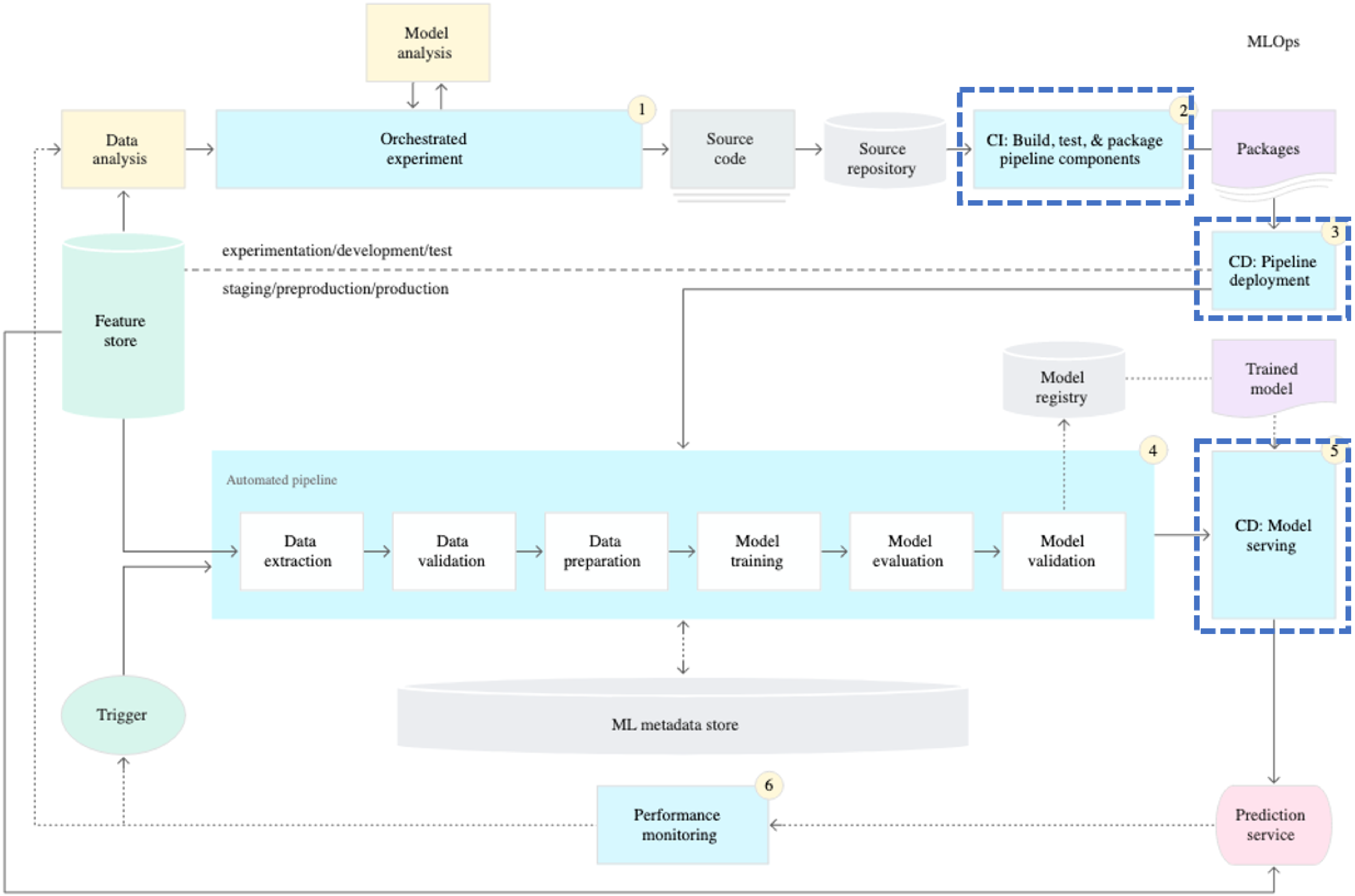
From Monitoring System



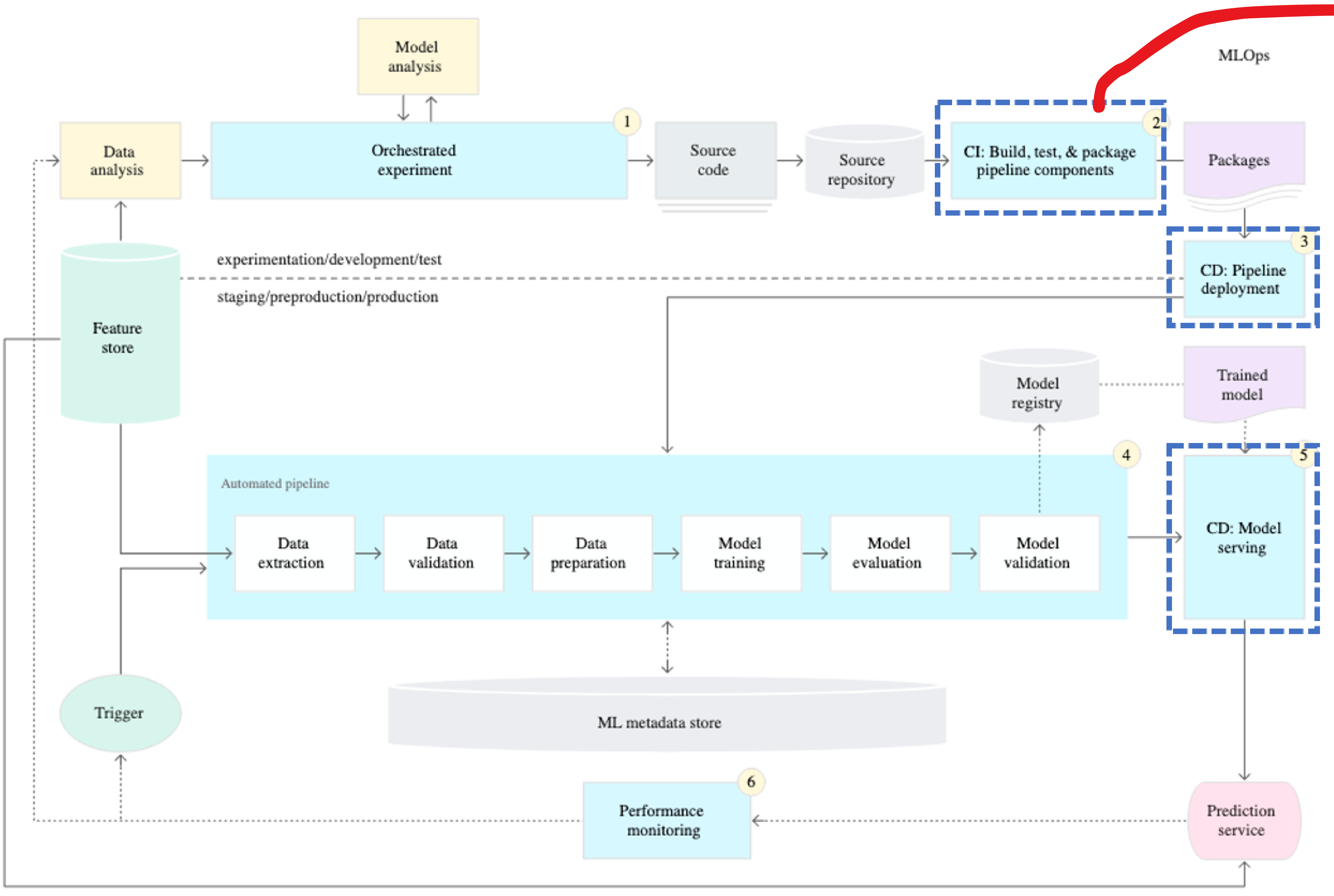
What if we automate things ? Manual (Level 0 of MLOps) is boring and takes a lot of time + resources !



Automating Integration and Deployment (CI/CD)

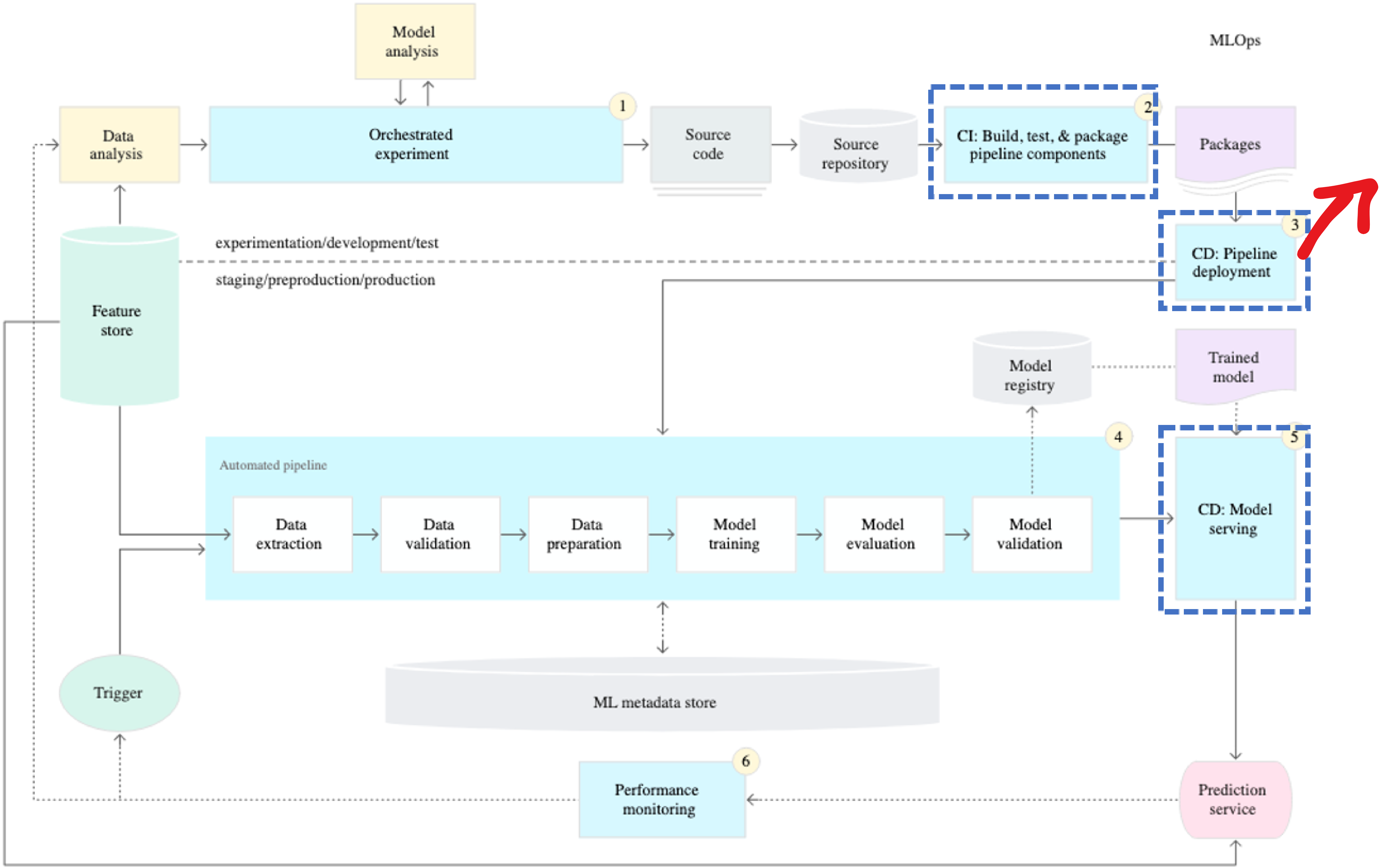


Automating Integration and Deployment (CI/CD)



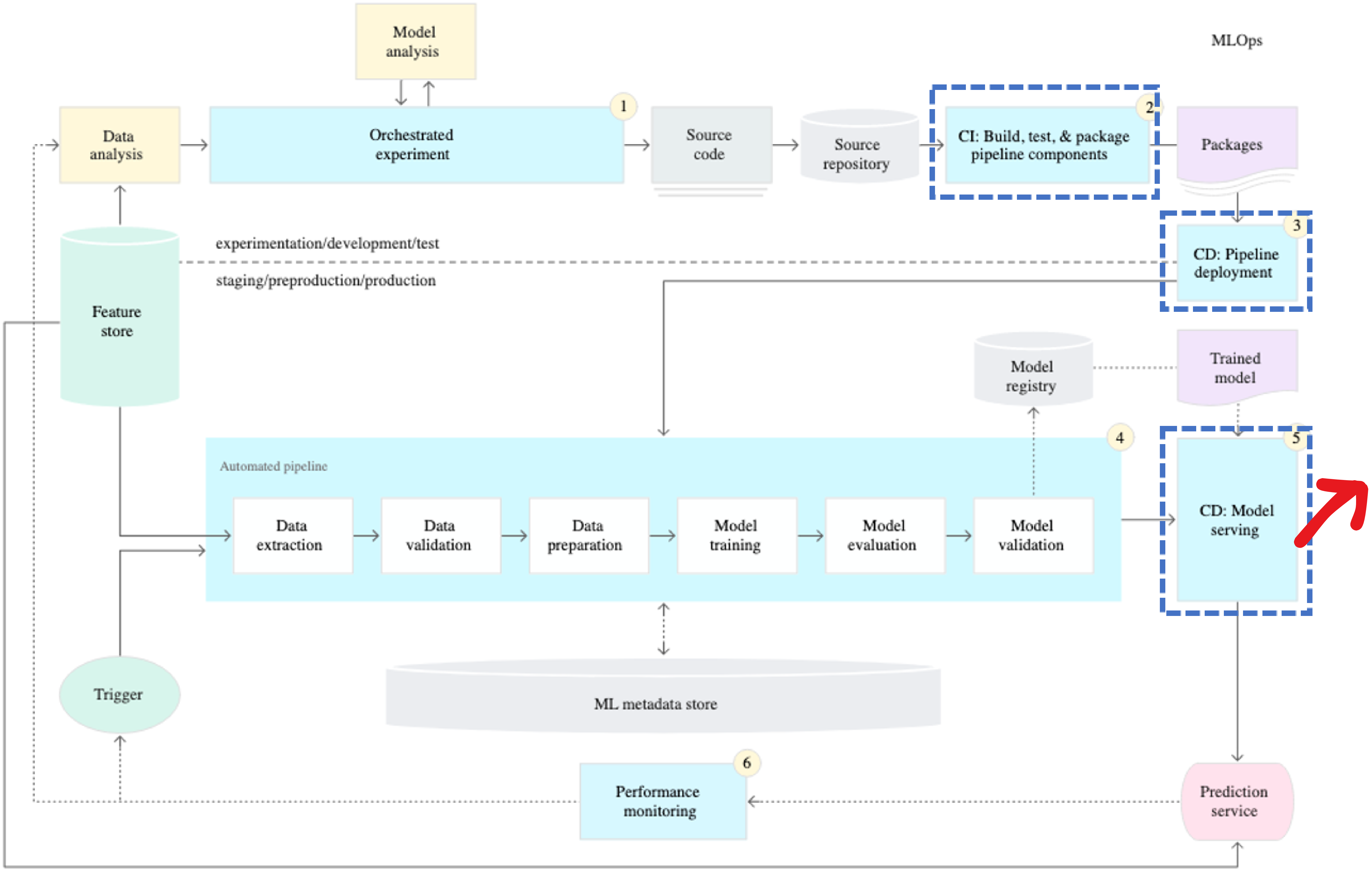
1. **Developers change in the source code**
2. **CI is triggered by changes in the repo**
3. **The code is built, tested and dependencies are integrated**
4. **If everything passes, the application is packaged into a deployable artifact.**

Automating Integration and Deployment (CI/CD)



1. Deployable artifact = Pacakges
2. Those are automatically deployed in their working environment
3. Then the deployed model enters "Level 1 of MLOps"

Automating Integration and Deployment (CI/CD)



1. We deploy the model in a production-ready environment.
2. For that we use TensorFlow Serving, TorchServe, NVIDIA Triton
3. We use a prediction service in order to manage request to the model

“Practice” Time !

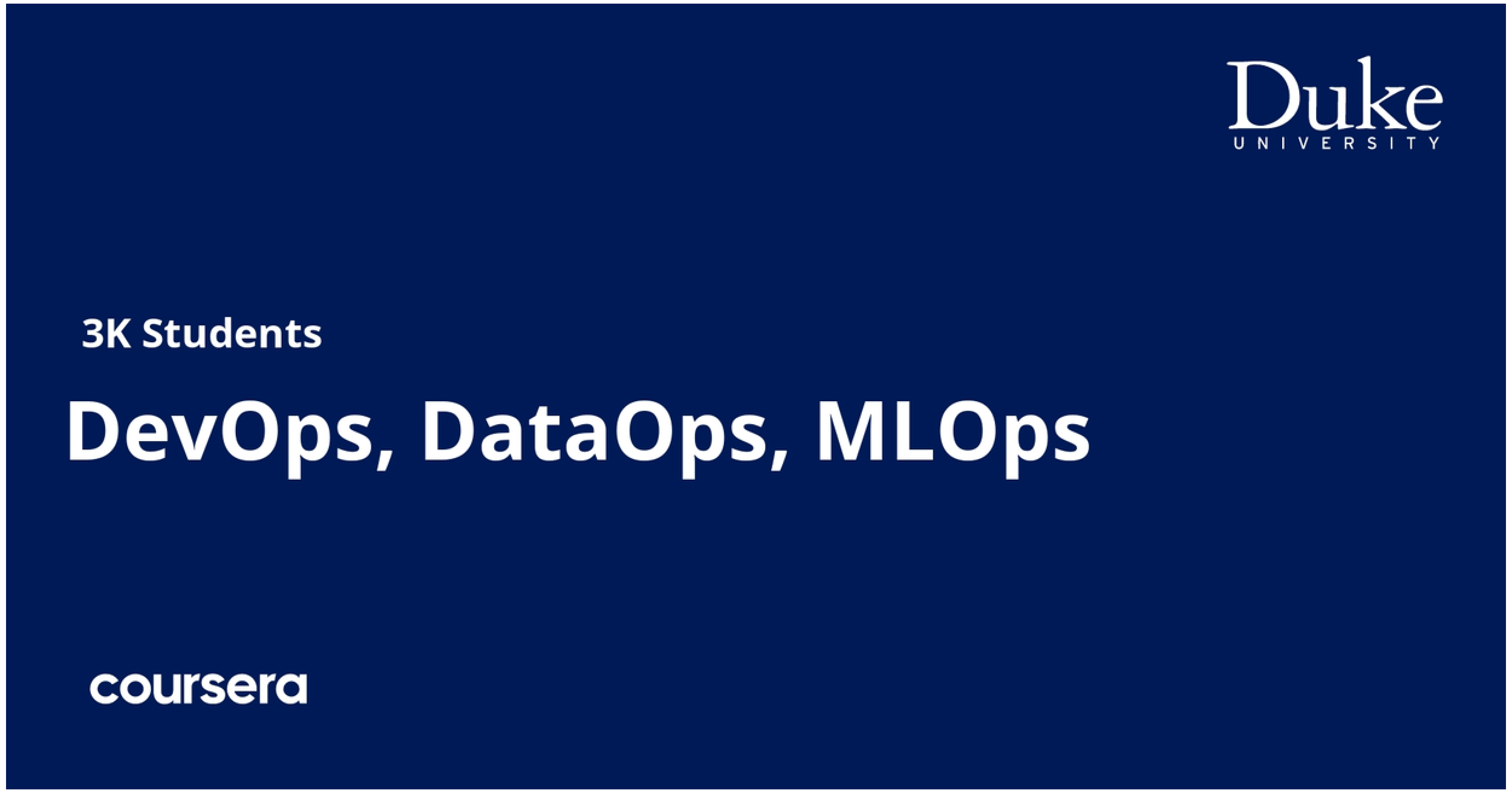
CREATE A CI/CD SOLUTION USING MAKEFILES AND GITHUB ACTIONS

Makefile

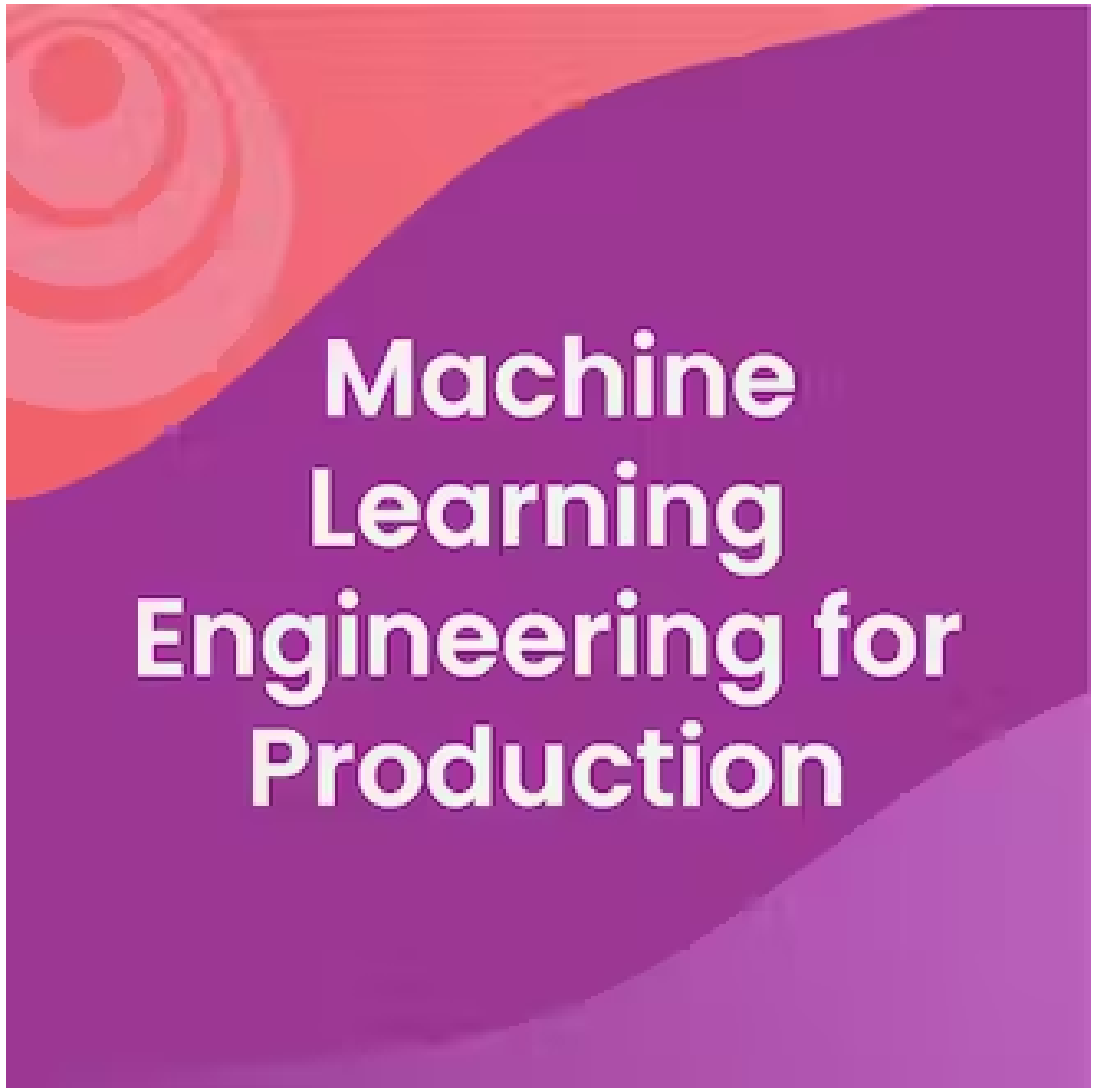
The original build tool

GitHub Actions

Where to go from now ?

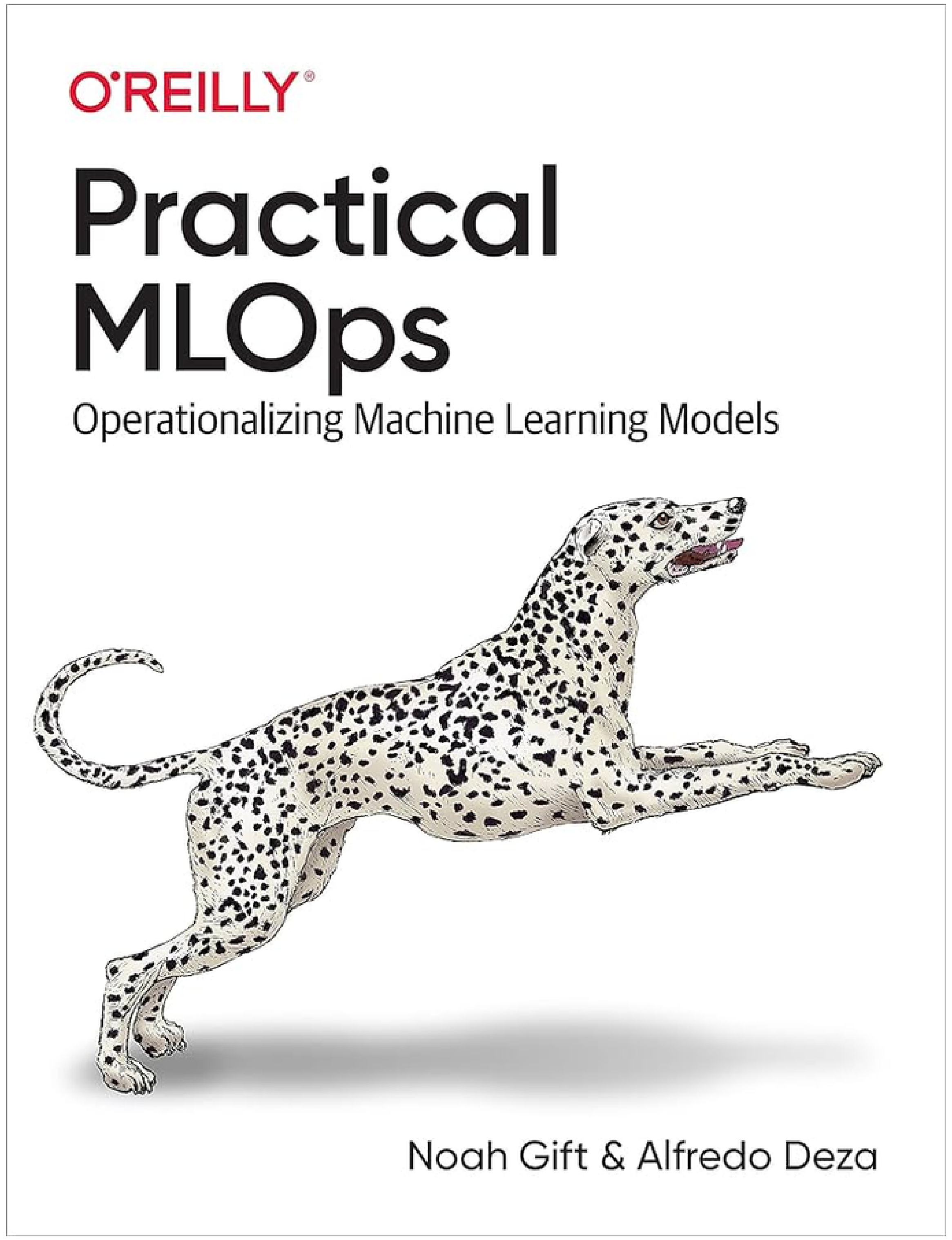


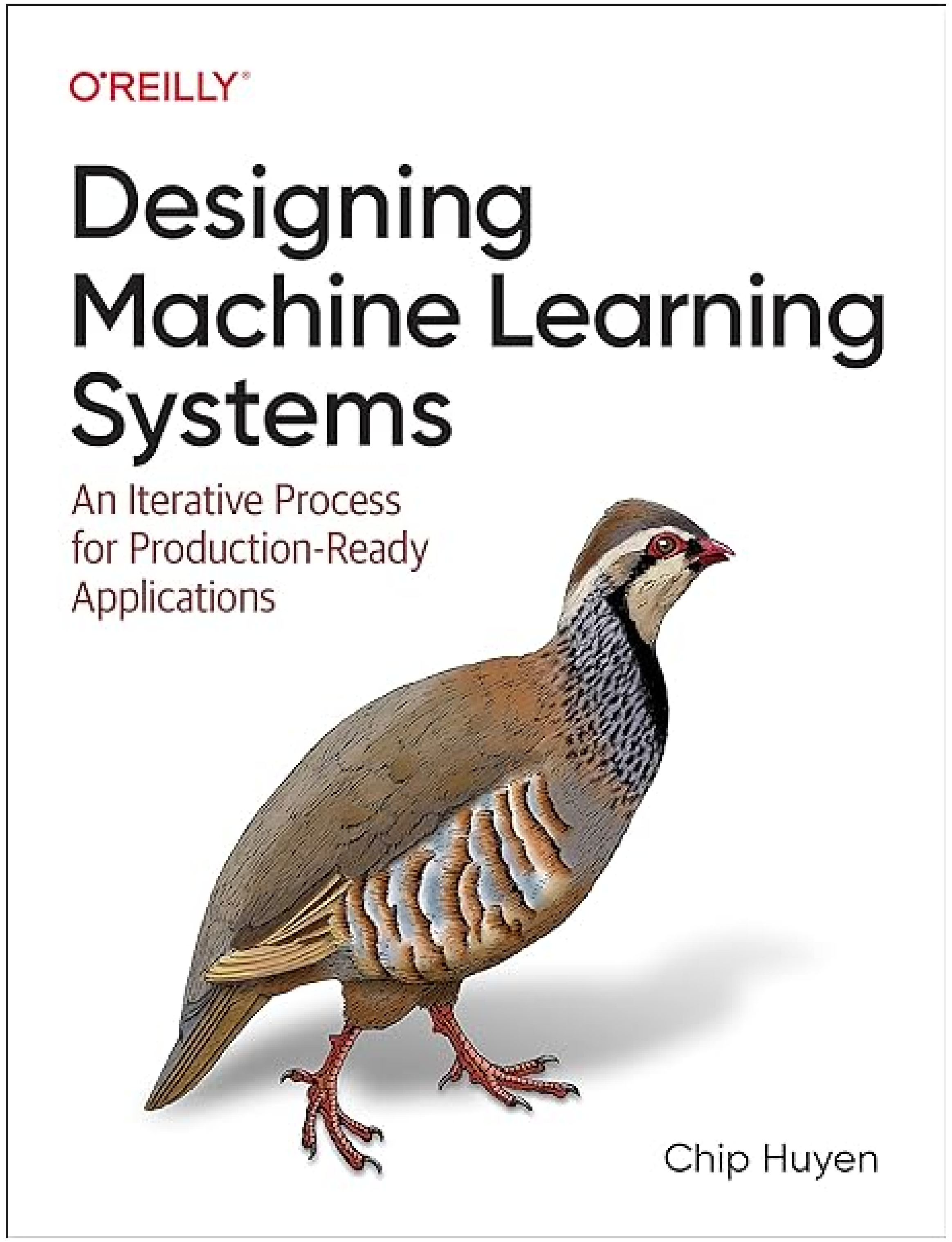
<u>Course</u>1



<u>Course2</u>

Where to go from now ?





Where to go from now ?

MLOps: Continuous delivery and automation pipelines in machine learning | Cloud Architecture Center

This document discusses techniques for implementing and automating continuous integration (CI), continuous delivery (CD), and continuous training (CT) for machine learning (ML) systems.

Google Cloud / May 18

Blog

Blog for ML practicioners with articles about MLOps, ML tools, and other ML-related topics. You'll find here guides, tutorials, case studies, tools reviews, and more.

neptune.ai

Google Cloud MLOps articles

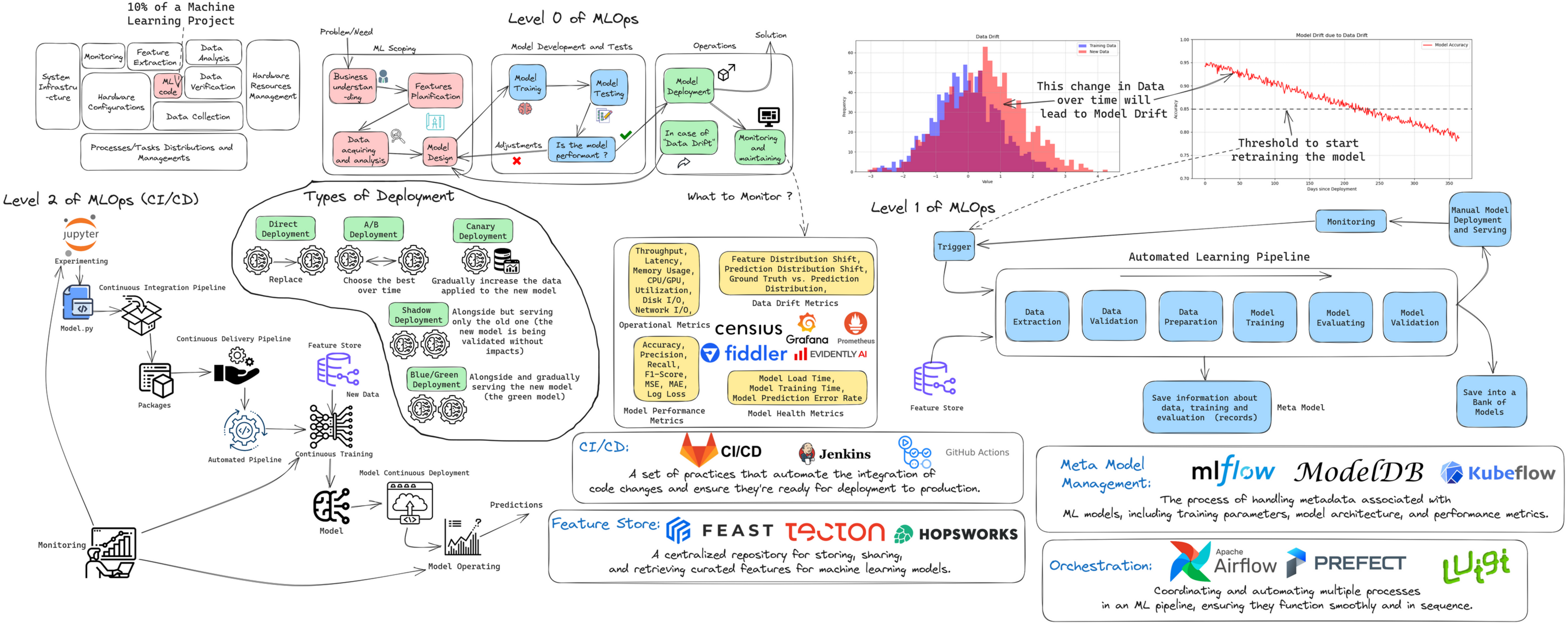
neptune.ai MLOps articles

Feedback, questions or anything you want:

LinkedIn

<u>**Azzedine Idir Aitsaid**</u>

MLOps Overview



Thank You !

Kaizen !