

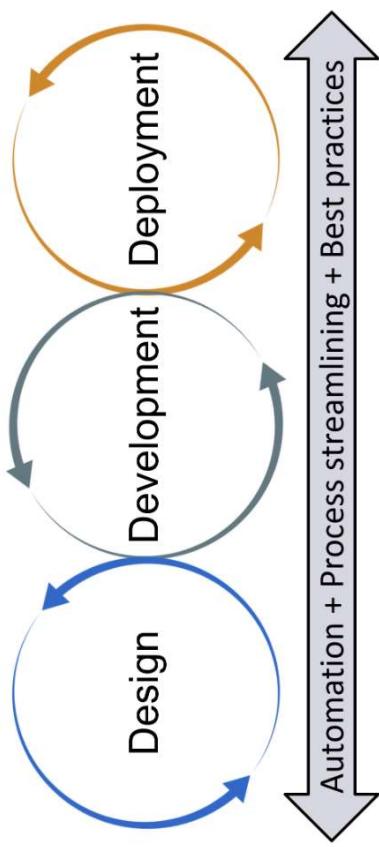
# Introducing full automation and best practices to MLOps

FULLY AUTOMATED MLOPS



Arturo Opsetmoen Amador  
Senior Consultant - Machine Learning

# The MLOps lifecycle

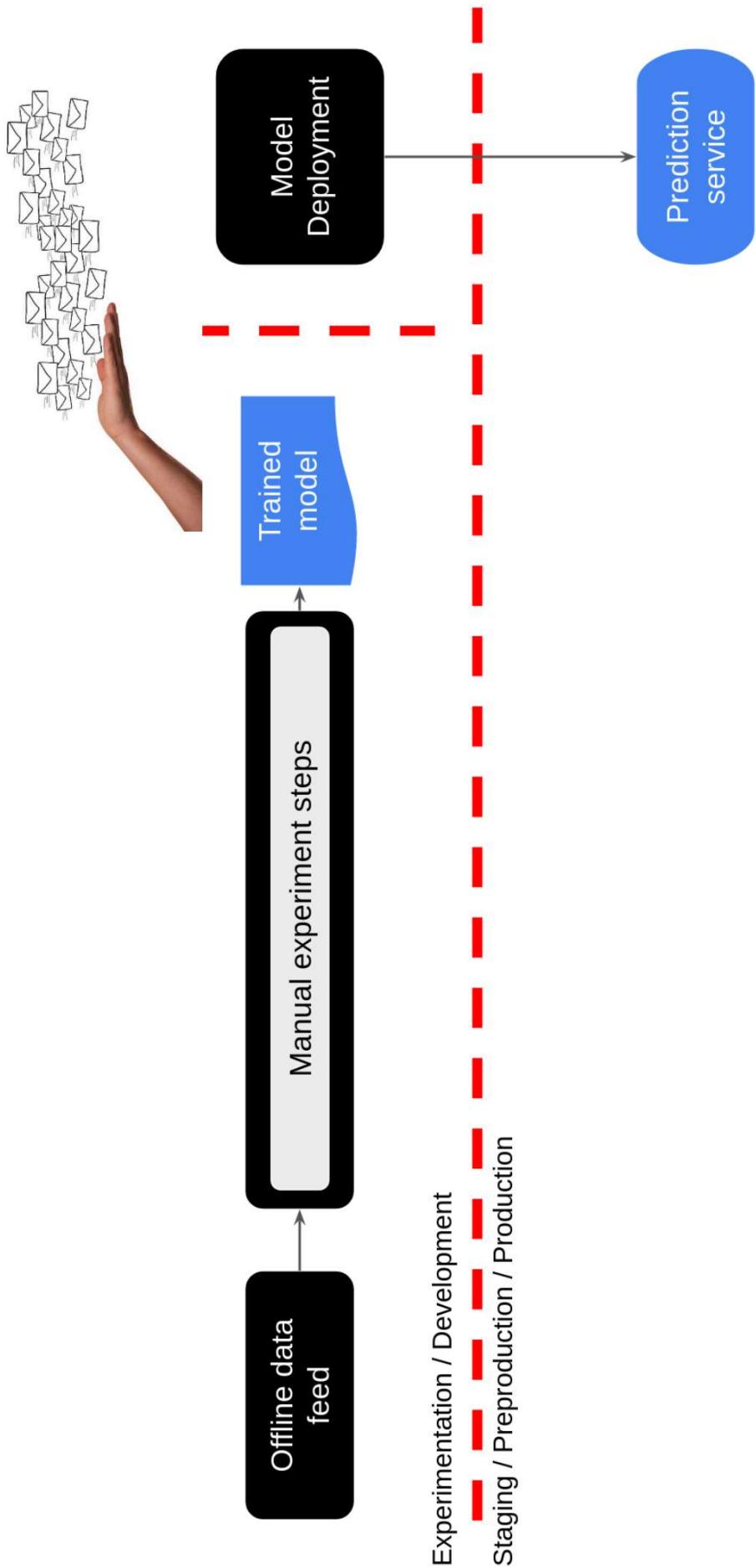


# Maturity levels in MLOps

The maturity levels include:

- Manual ML workflow
- Semi-automated ML workflow
- Fully automated ML workflow

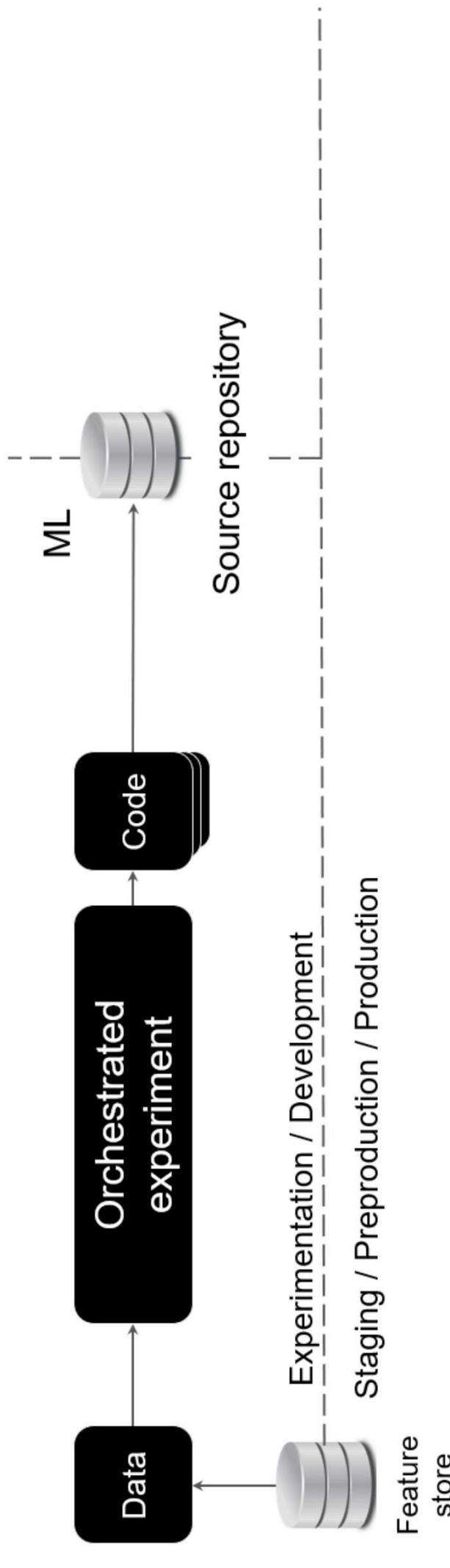
# Manual ML workflow - Ad hoc experimentation



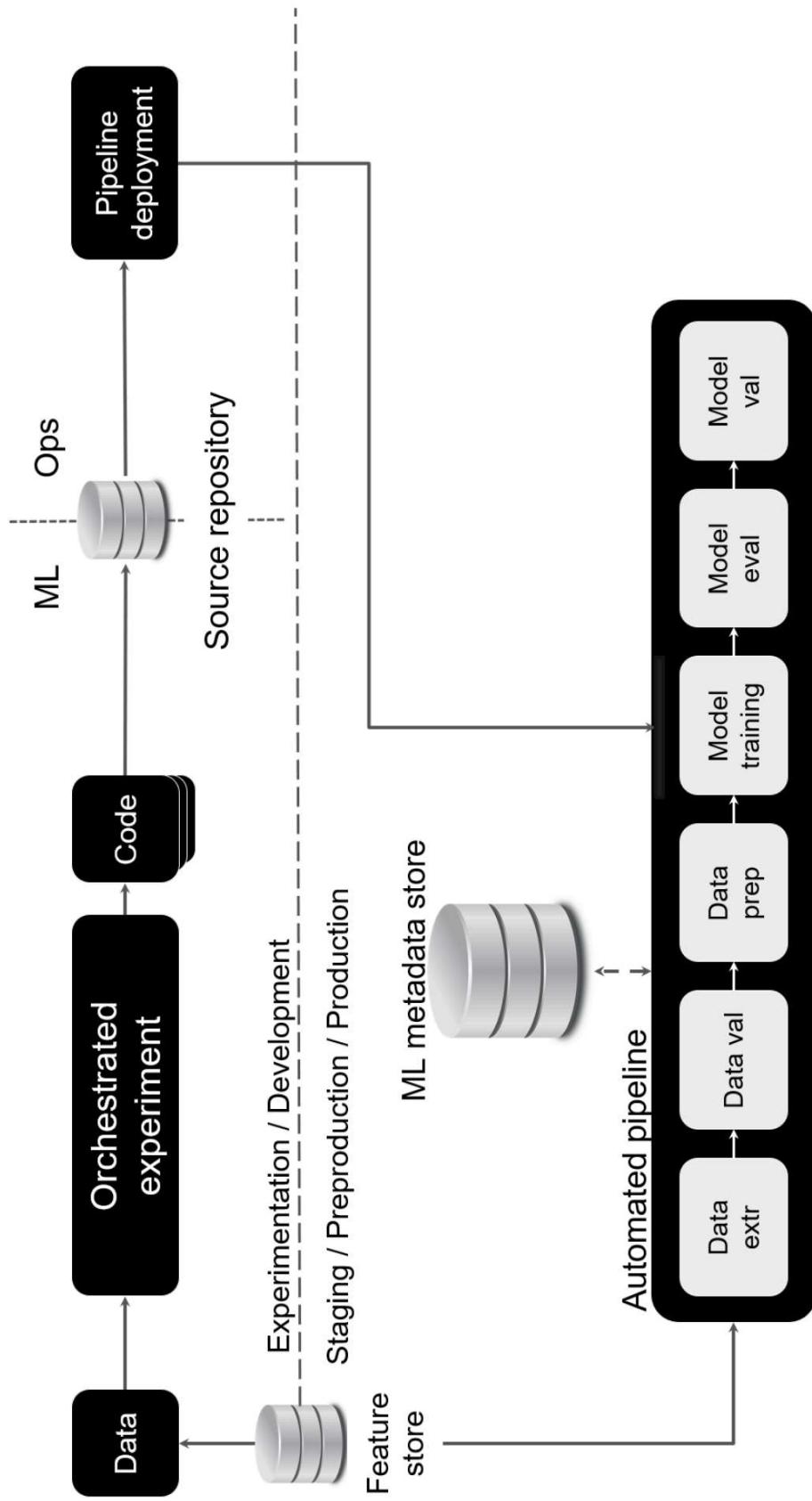
# Semi-automated ML workflow



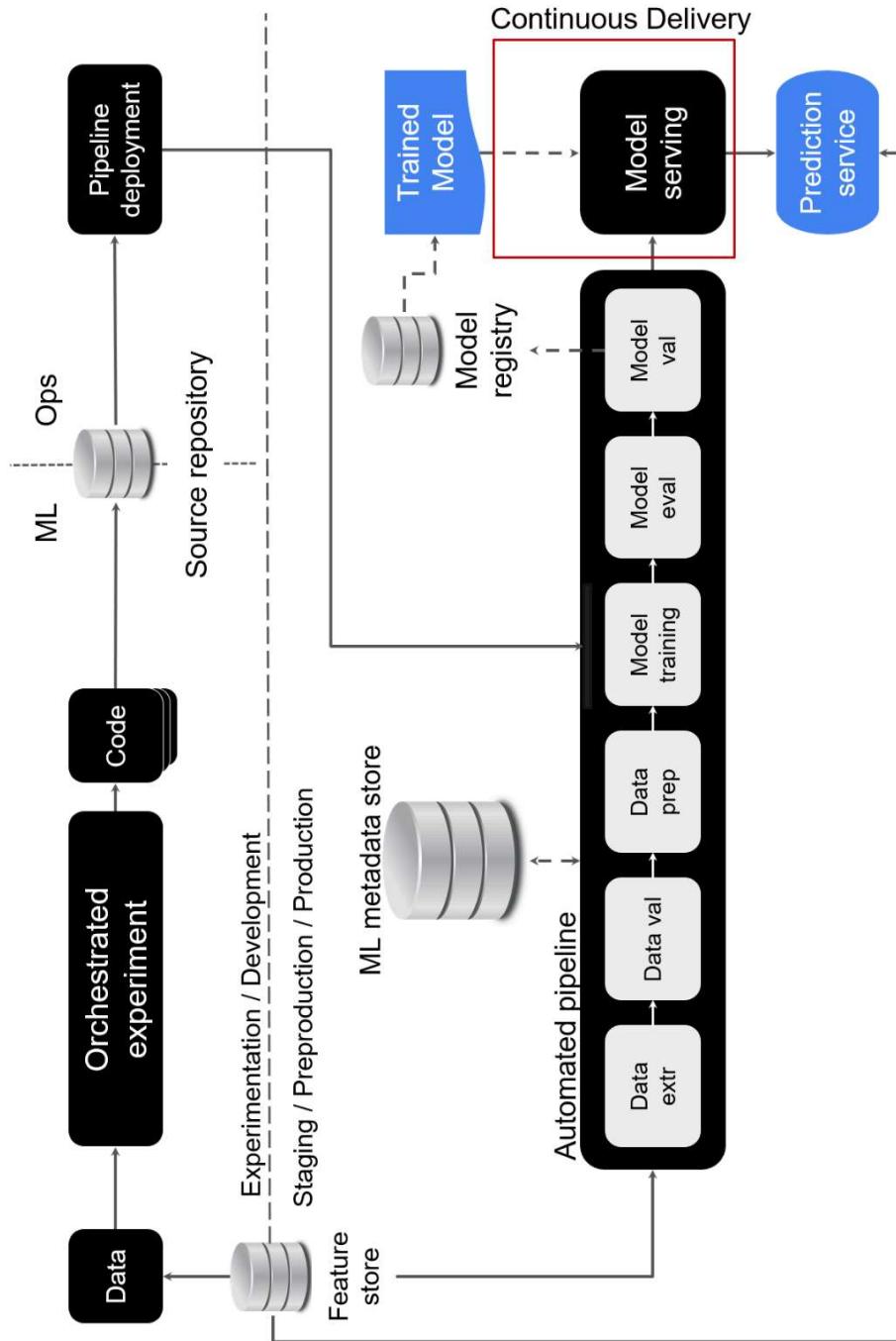
# Semi-automated ML workflow



# Semi-automated ML workflow

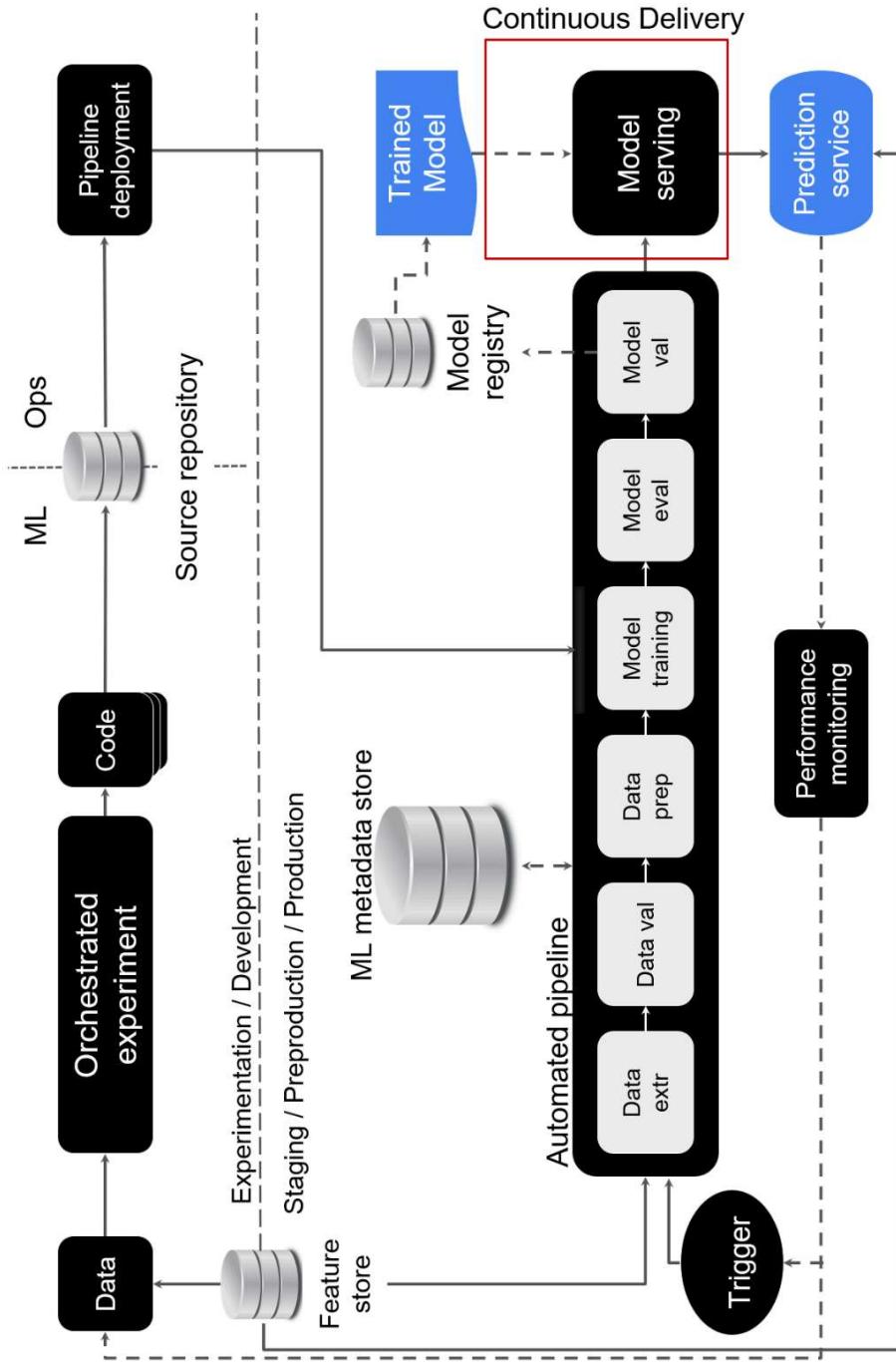


# Semi-automated ML workflow

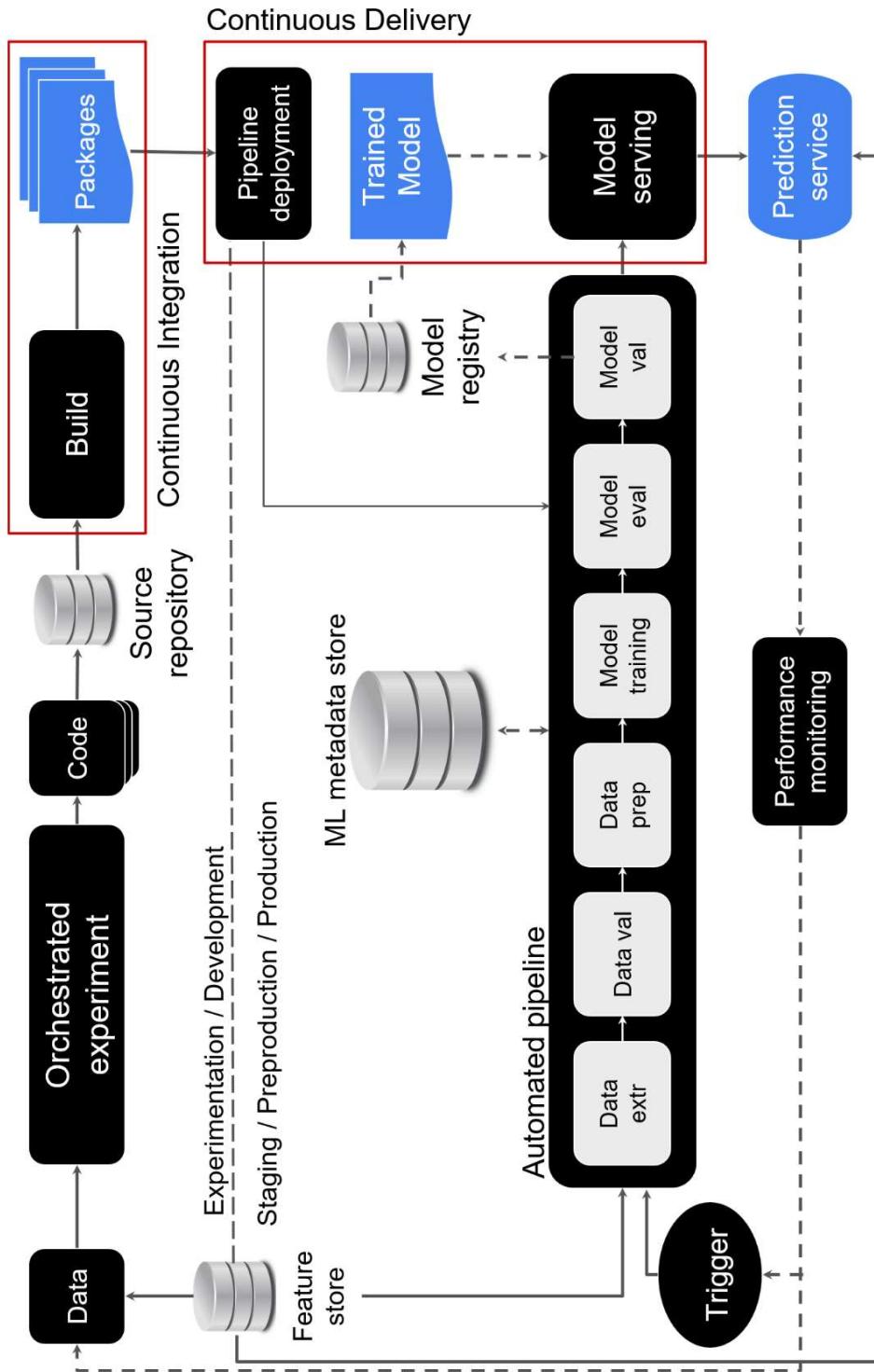


FULLY AUTOMATED MLOPS

# Semi-automated ML workflow



# Fully automated ML workflow



datacamp

FULLY AUTOMATED MLOPS

# Automation in the ML life cycle - Design

Design & Planning		Activity in Workflow	Fully Automated
Business understanding			●
Data understanding			●
System design			●

Good practices:

- Apply reproducible processes
- Write detailed documentation

# Automation in the ML life cycle - Development

Development & Experimentation		Activity in Workflow	Fully Automated
		Developing PoCs	●
		Data engineering	●
		Model development	●
		...	

Good practices:

- Remember we are developing software
- Using version control
- Use orchestration tools

# Automation in the ML life cycle - Operations

Deployment & Operations	Activity in Workflow	Fully Automated
	Testing	✓
	Versioning	✓
	Monitoring	✓

Use:

- Automated testing
- CI/CD/CT/CM

# Let's practice!

FULLY AUTOMATED MLOPS

# The automation, monitoring, incident response pattern

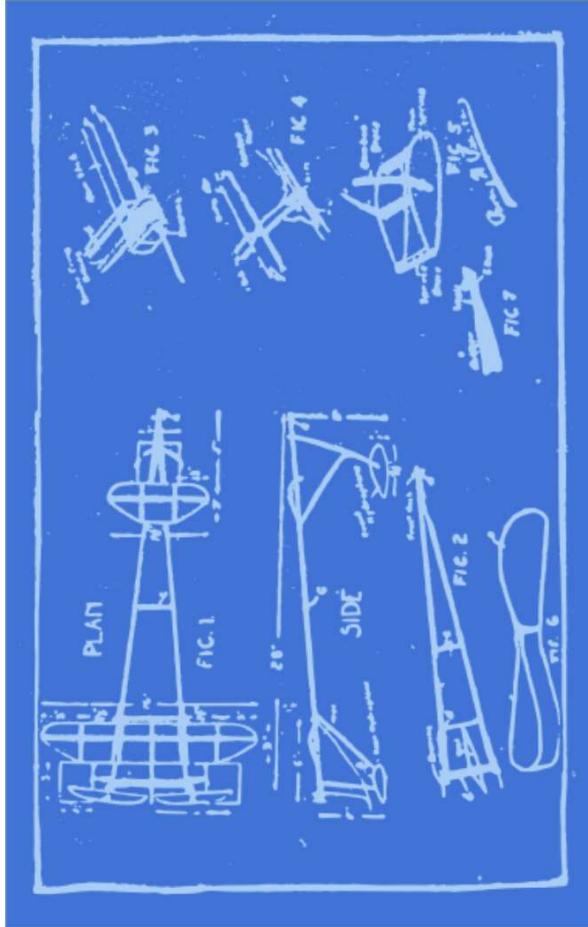
FULLY AUTOMATED MLOPS



Arturo Opsetmoen Amador  
Senior Consultant - Machine Learning

# What is a software design pattern?

A general, reusable solution to a commonly occurring problem...



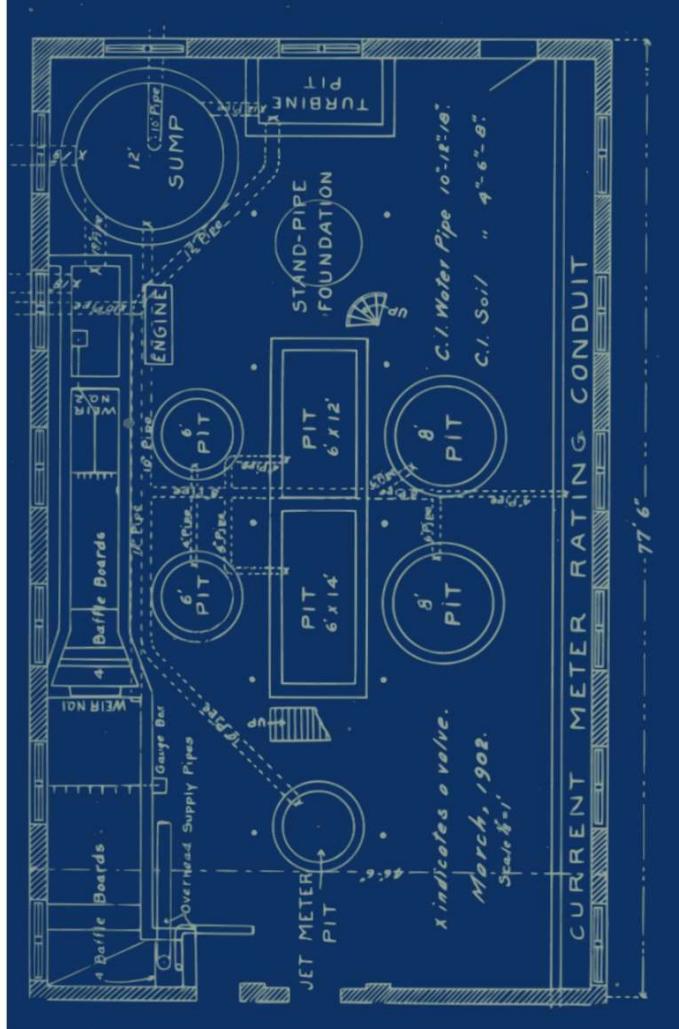
# Automate, monitor, respond



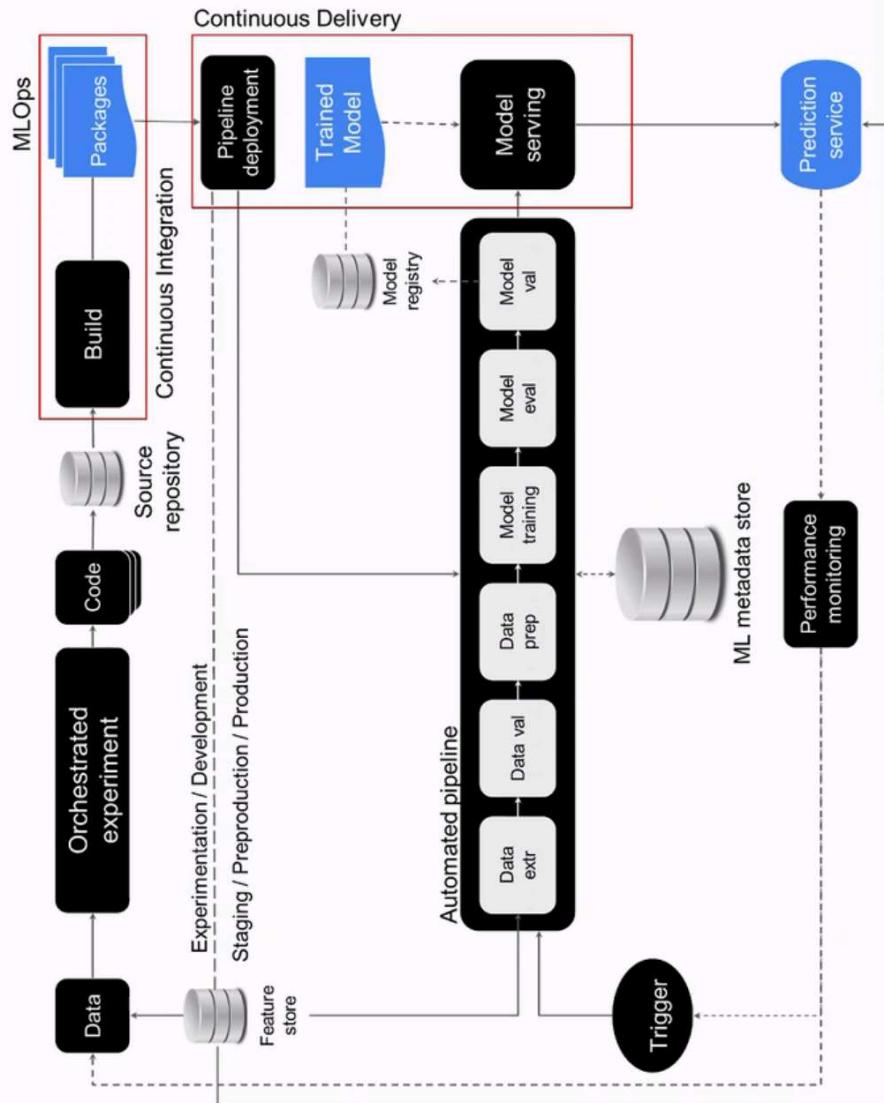
- Improves the reliability of the ML systems we design

# Three examples of a design pattern in MLOps

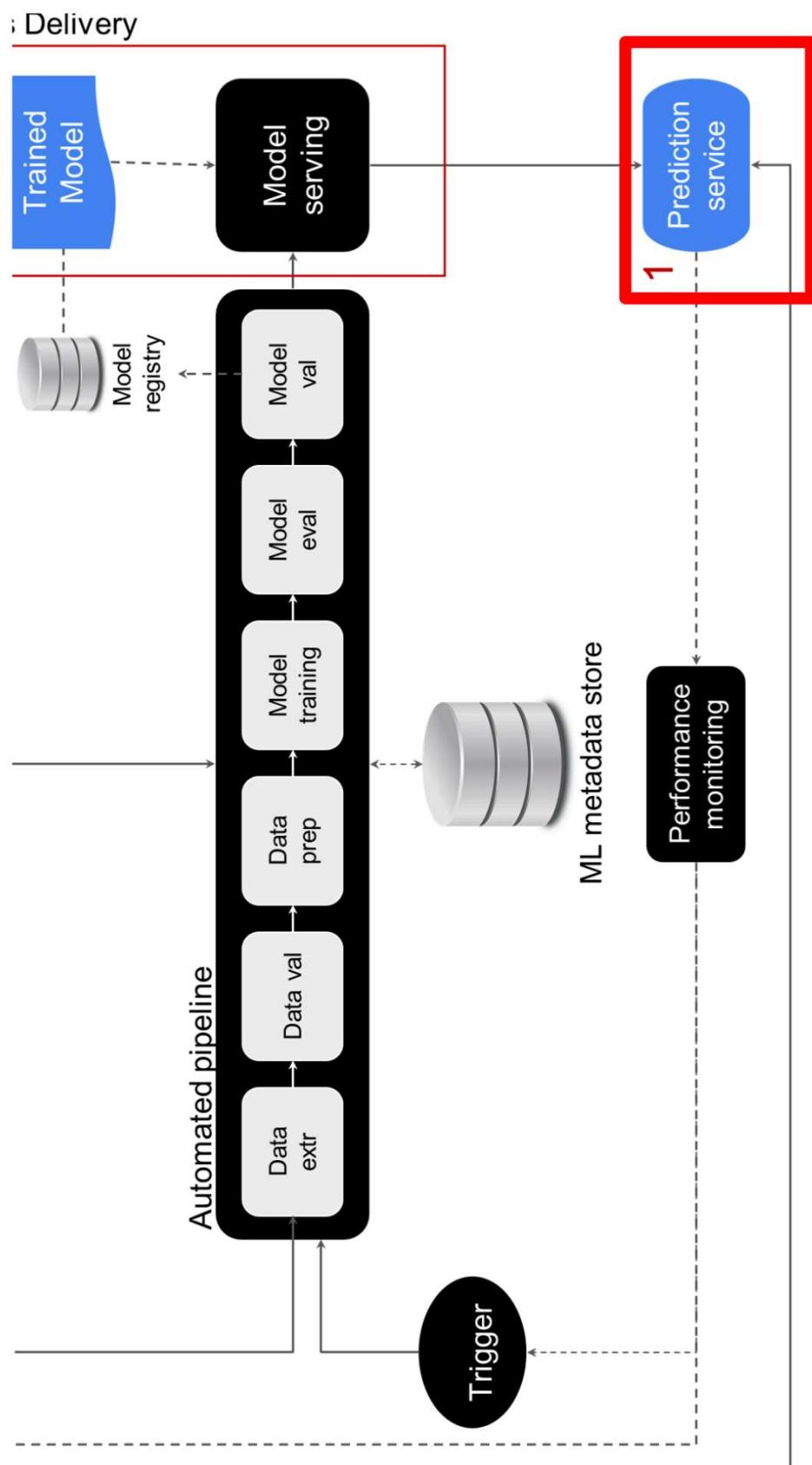
1. Automated Model Retraining
2. Model Rollback
3. Feature Imputation



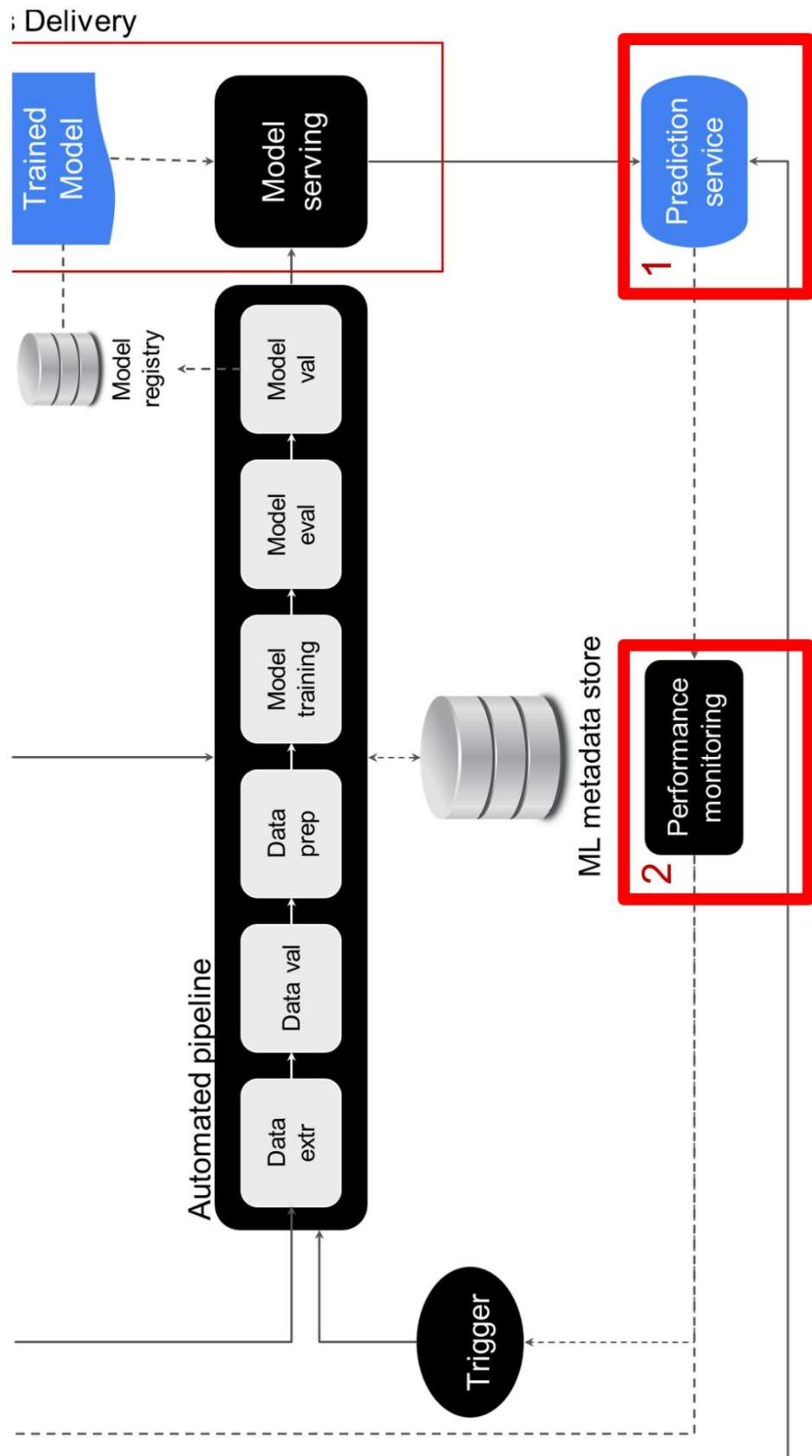
# 1. Automated model retraining



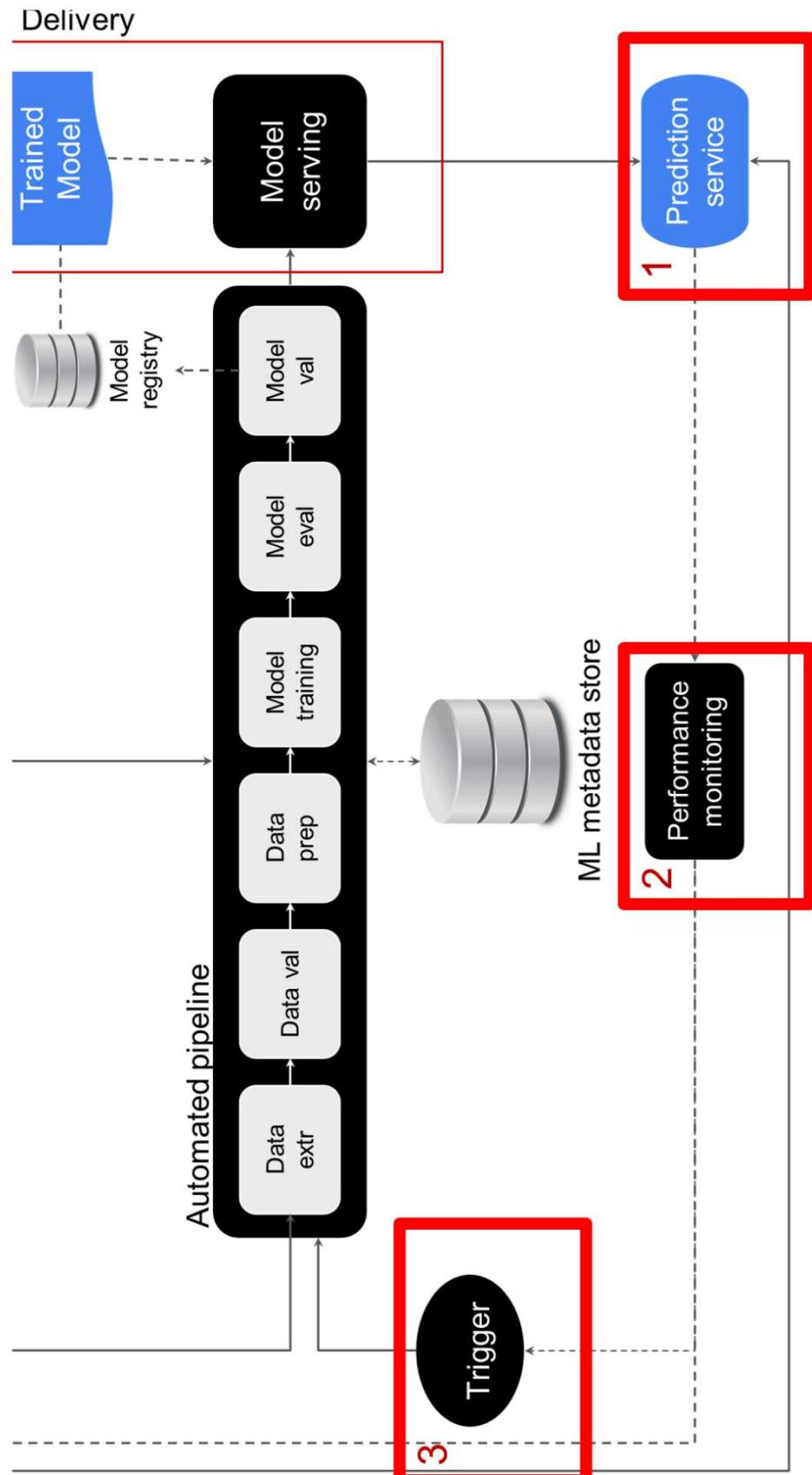
# 1. Automated model retraining - running predictions



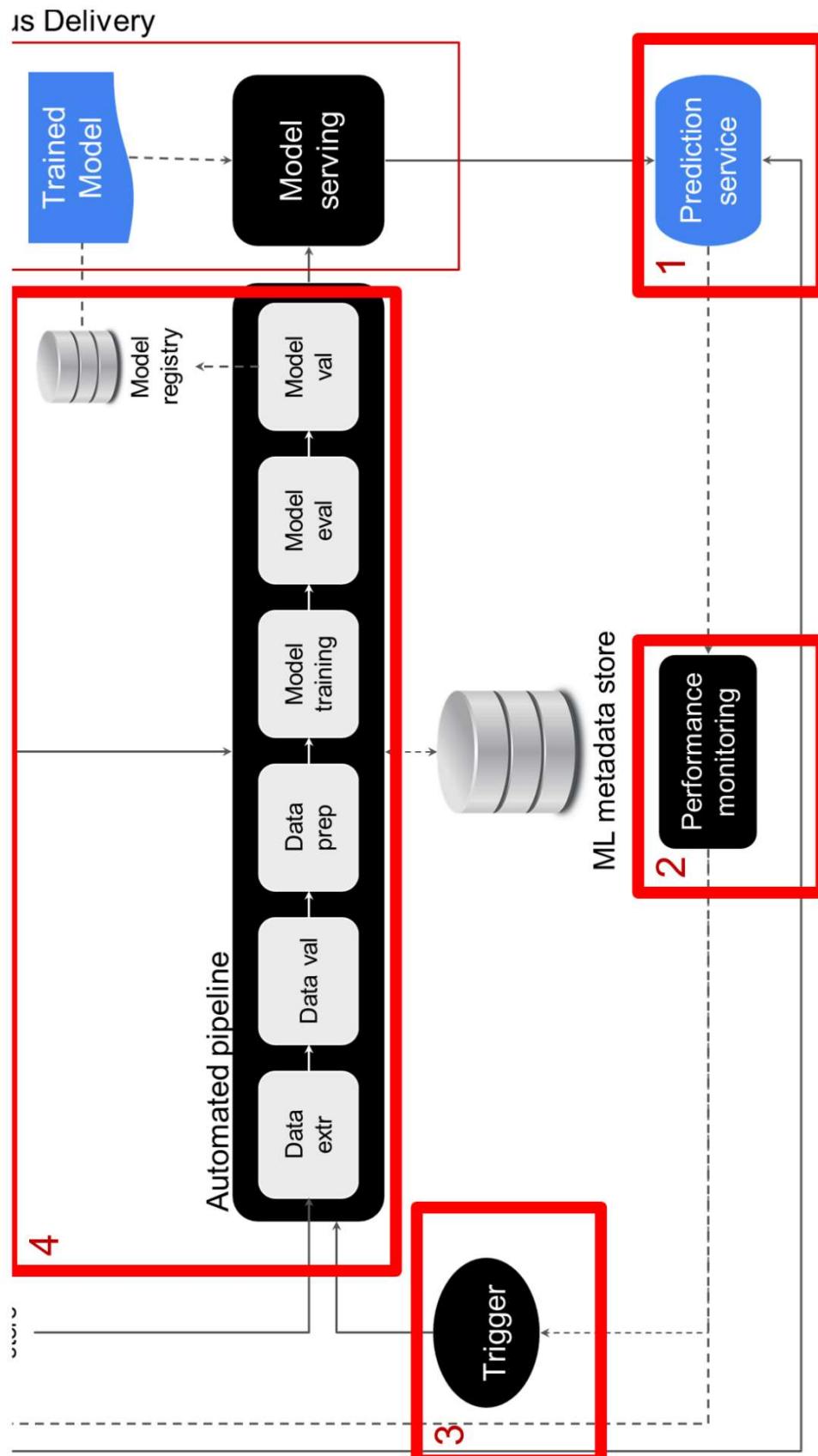
# 1. Automated model retraining - Monitoring



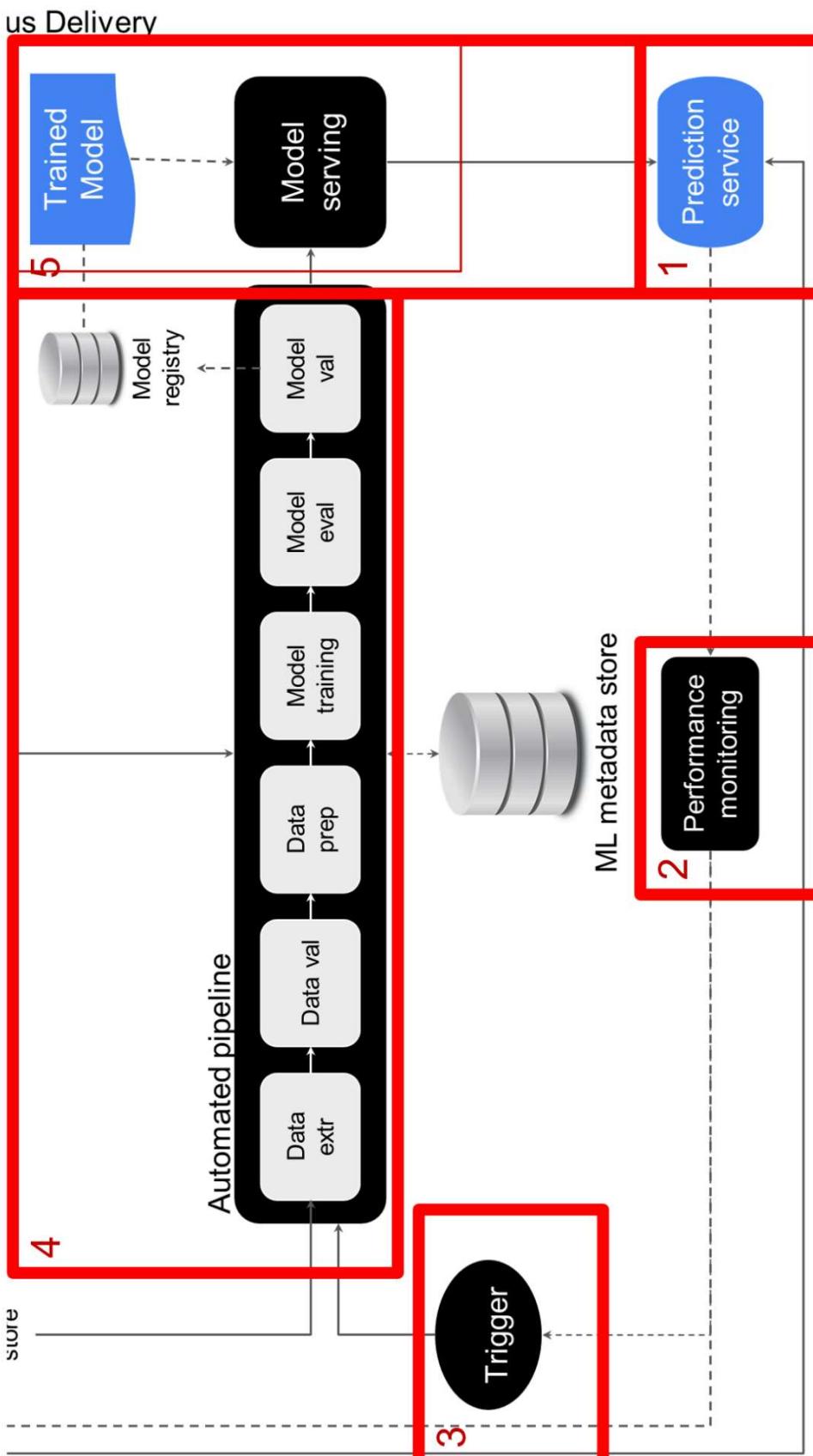
# 1. Automated model retraining - Trigger



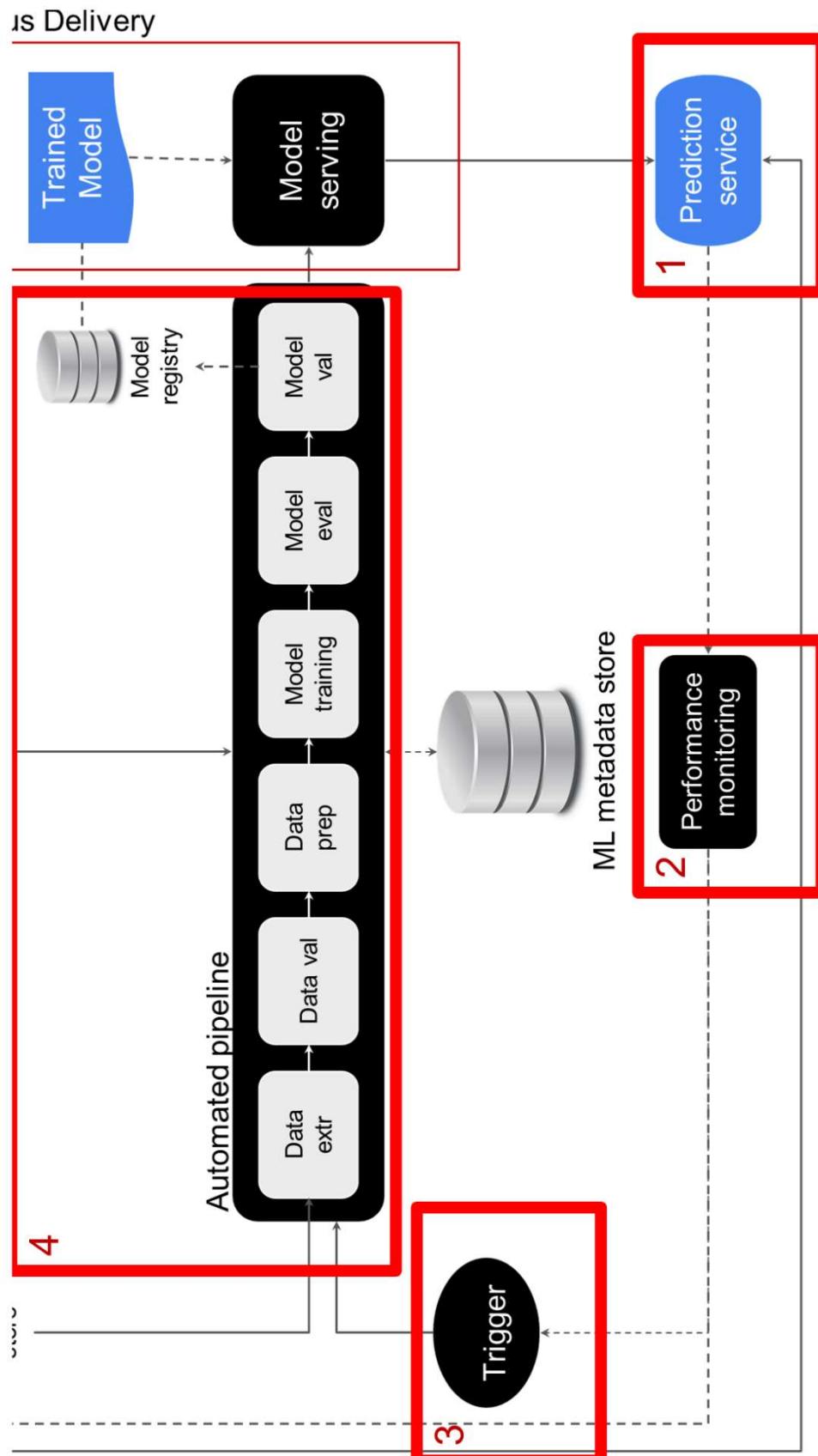
# 1. Automated model retraining - Automated pipeline



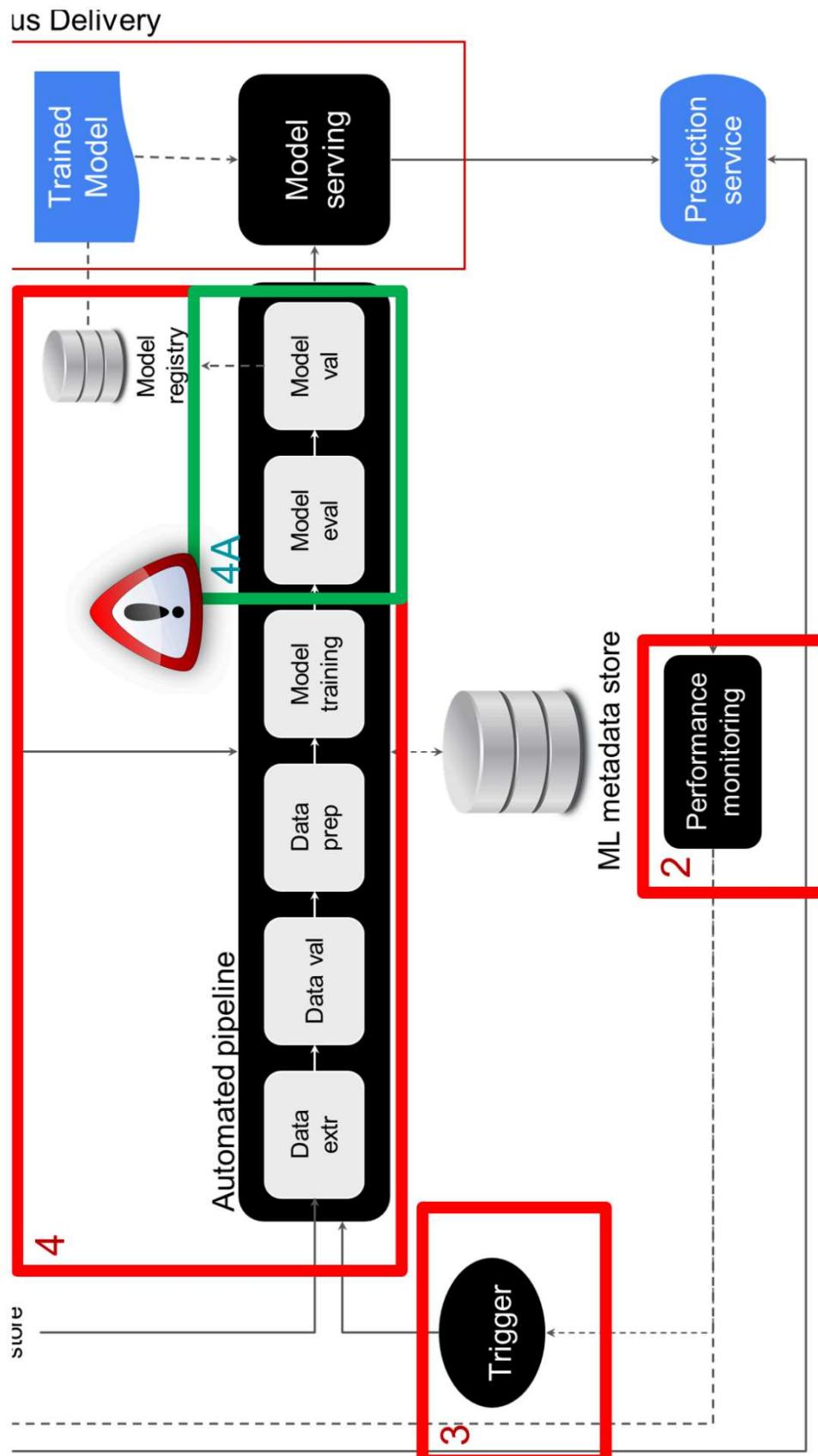
# 1. Automated model retraining - Deployment



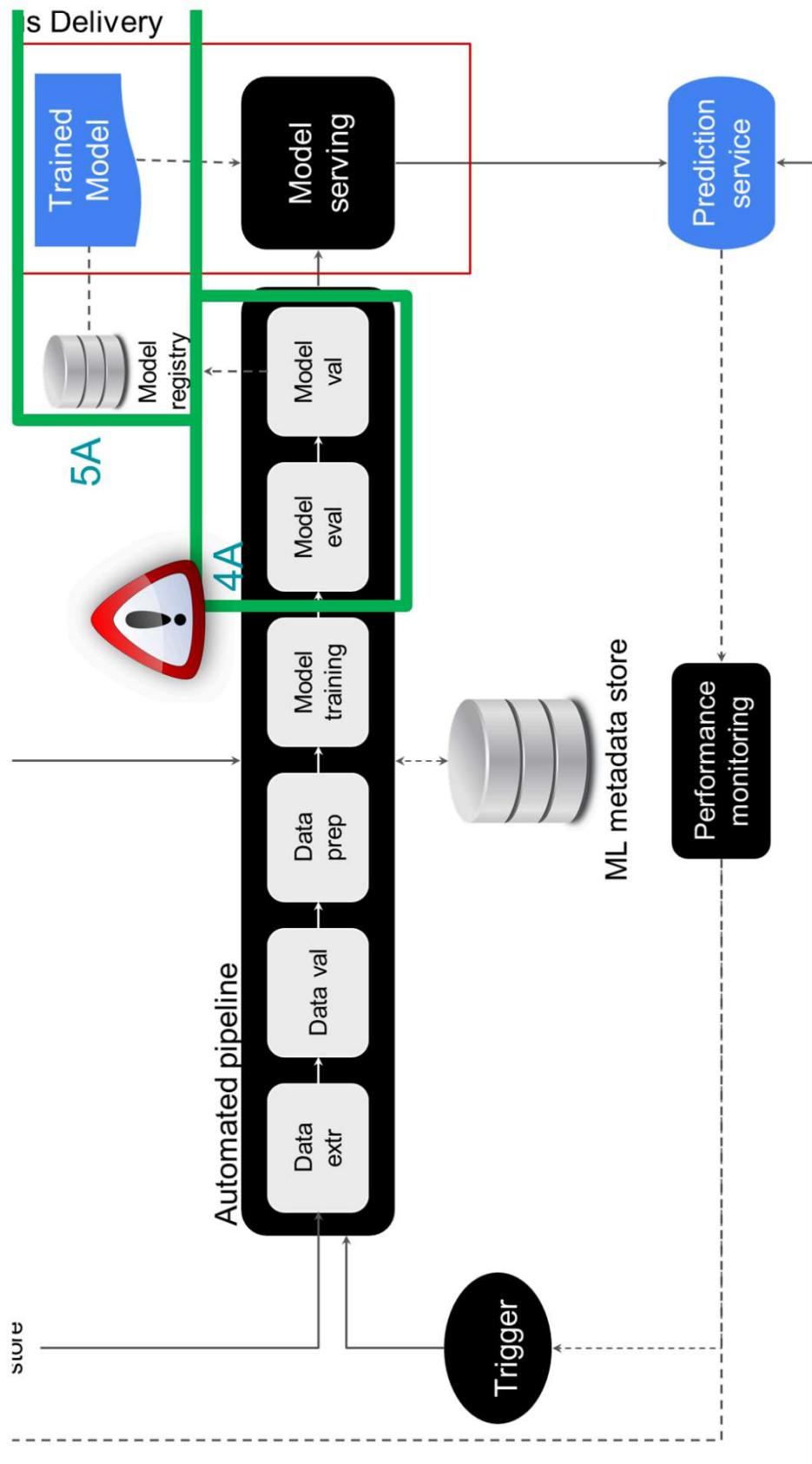
## 2. Model rollback



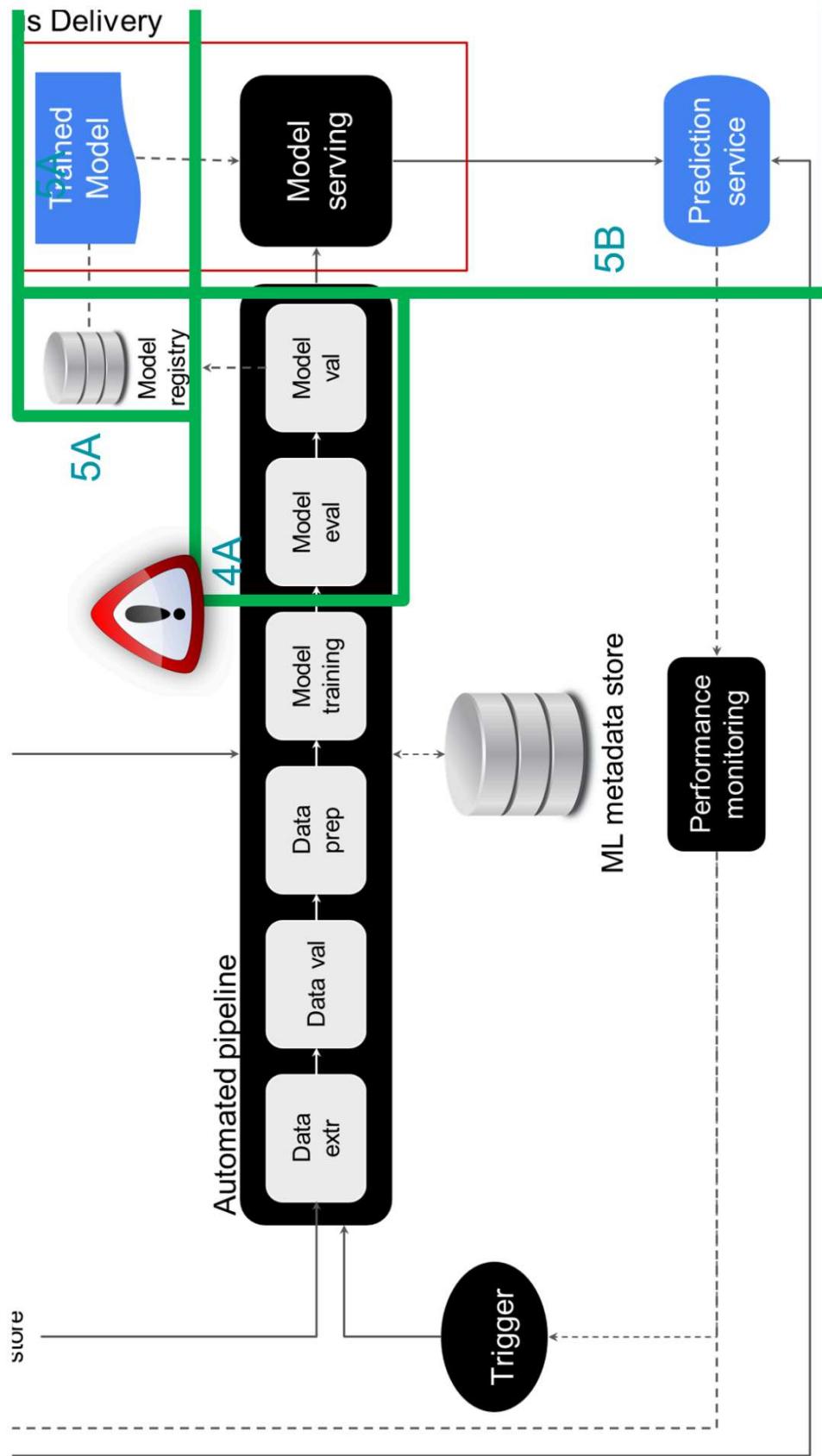
## 2. Model rollback - Validation fail



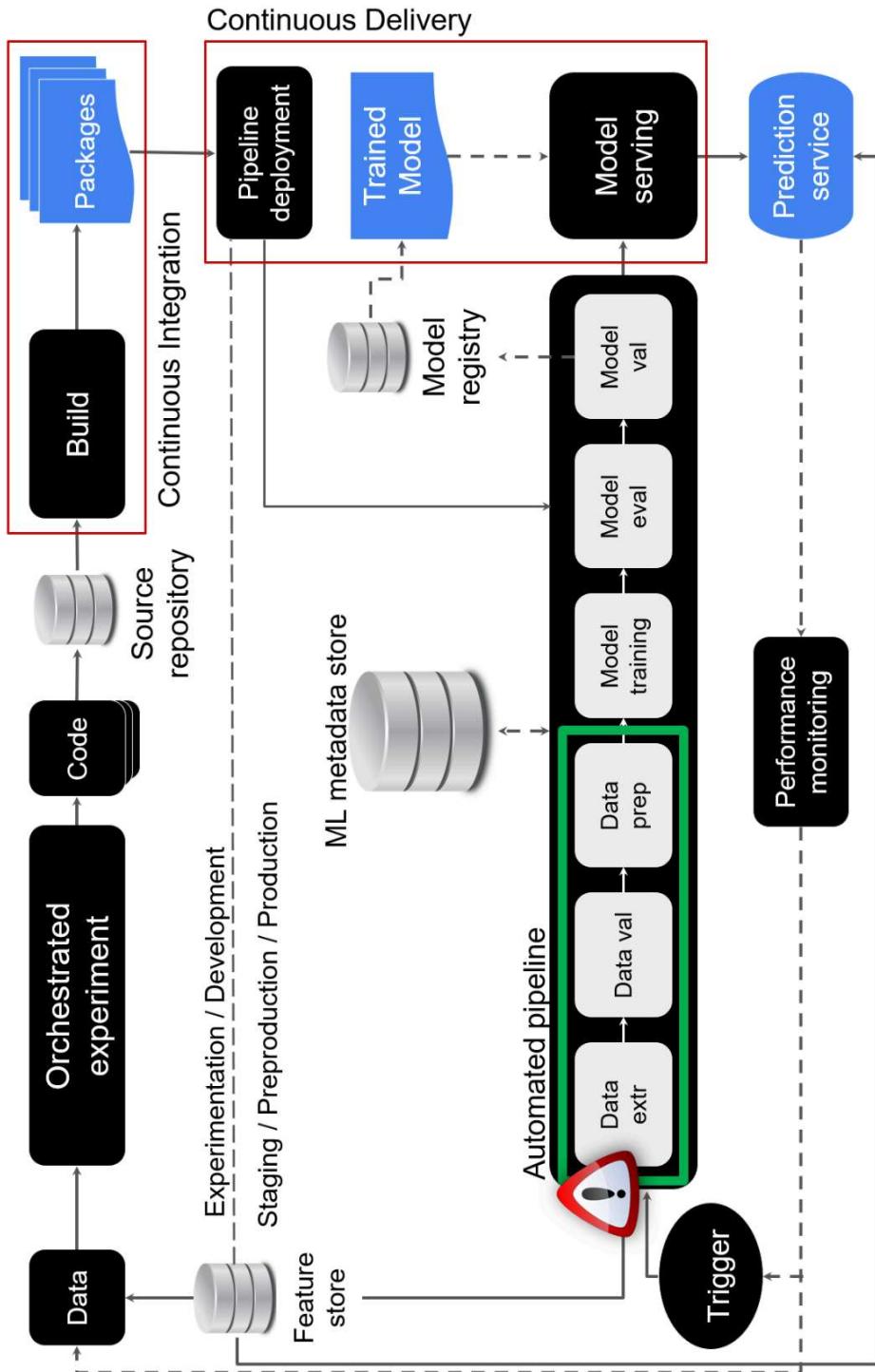
## 2. Model rollback - Last functional model



## 2. Model rollback - Redeployment



### 3. Feature imputation - Data intensive pipeline



## 3. Feature imputation - Data quality

Feature A	Feature B
213	Blue
234	NAN
132	NAN
182	NAN
190	Red

• • •

- • •
- 
- 
- 

- Varying levels of data quality
- Some features might fall below a QA threshold

Feature Y	Feature Z
908	Low
NAN	High
731	Medium
NAN	Very Low
NAN	Very High

• • •

## 3. Feature imputation - Defective features

Feature A	Feature B	Feature C
213	NAN	Low
234	NAN	High
132	NAN	Medium
182	NAN	Very Low
190	Red	Very High
• • •	• • •	• • •

- Detect failing features
- Apply feature imputation

NaN	Low
NaN	Very High
NaN	Very High
NaN	Medium

## 3. Feature imputation - Potential fixes

- Numerical Values

- Mean/Median Imputation
- KNN Imputation

- Categorical Values

- Frequent Category Imputation
- Adding a "Missing" Category

# Let's practice!

FULLY AUTOMATED MLOPS

# Automated testing in MLOps

FULLY AUTOMATED MLOPS

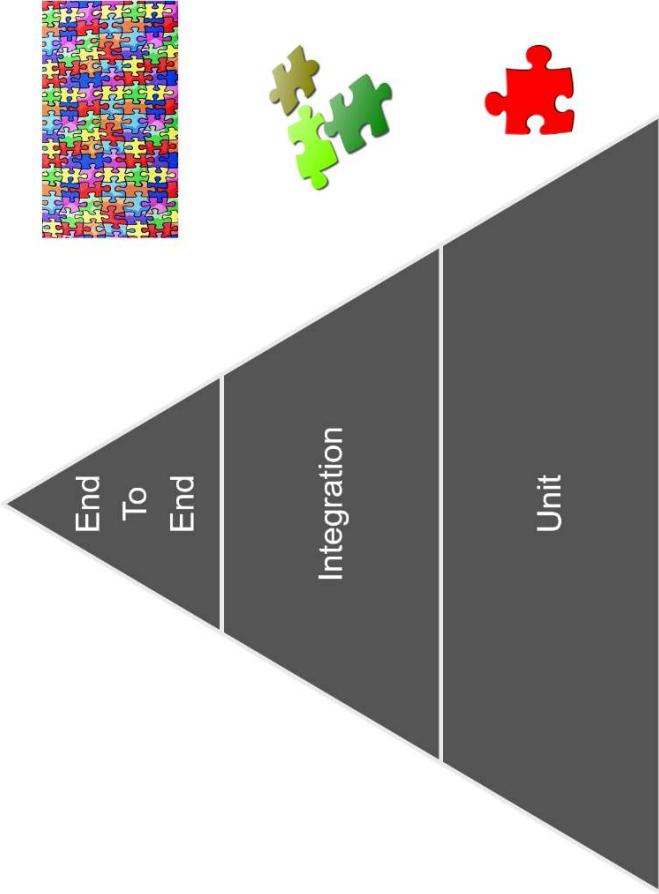


**Arturo Opsetmoen Amador**  
Senior Consultant - Machine Learning



# What is software testing?

The process of evaluating and verifying that a software product or application works as expected

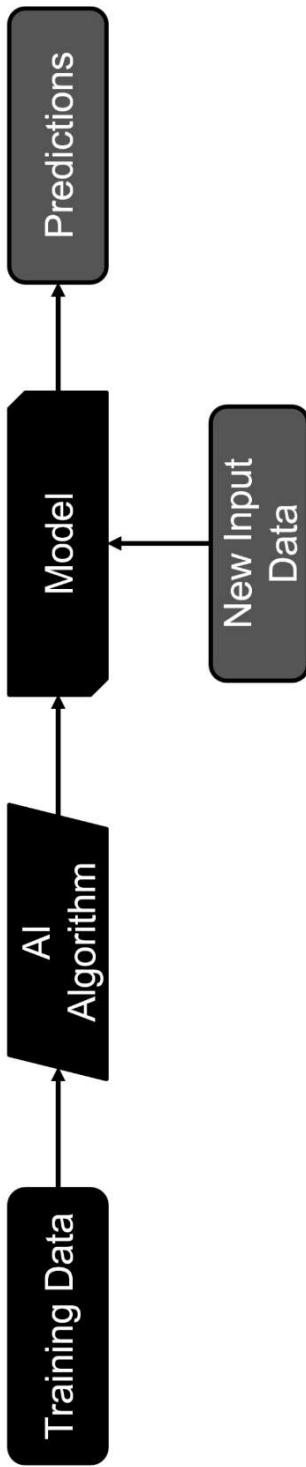


Three most common types of software testing:

- Unit tests
- Integration tests
- End-to-end tests

# ML software has a different nature

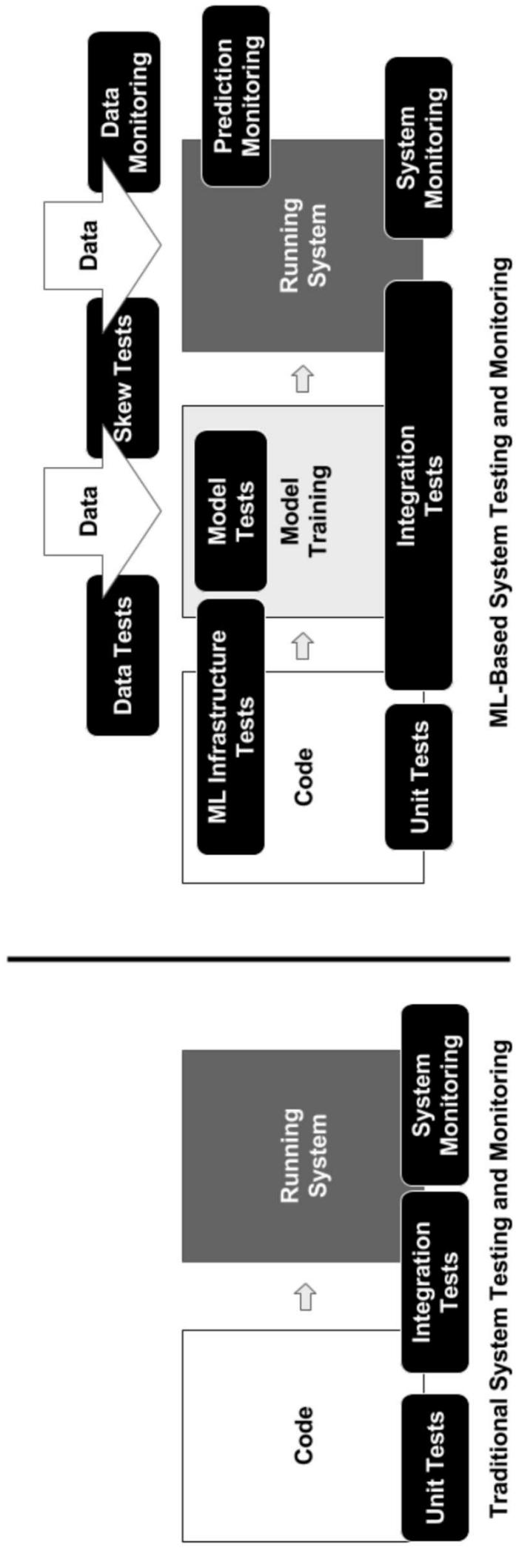
ML APPLICATIONS ARE NOT EXPLICITLY PROGRAMMED



They depend on:

- Data
- Models

# Testing in an MLOps system is different



<sup>1</sup> <https://research.google/pubs/pub46555/>

# Testing ML systems

In addition to traditional software testing:

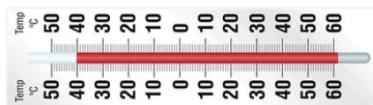
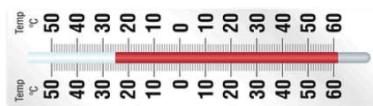
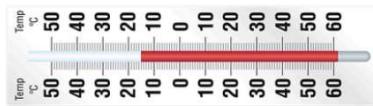
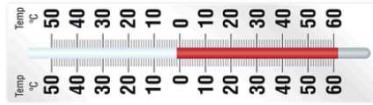
- Data tests
- Model tests
- Pipeline tests



# Testing the data

Tests for features and data include:

- Feature expectations
- Value from data feature justify its costs
- Privacy control
- Avoid used of unlawful data



# Testing the models

Tests for models include:

- Business & ML metrics correlate
- All hyperparameters have been tuned
- Model overfitting
- Model staleness
- Baseline comparison

# Testing ML pipelines

Tests for ML pipelines include:

- Training is reproducible
- Integration testing of the ML pipeline
- Model debuggability

# Let's practice!

FULLY AUTOMATED MLOPS

# Automated hyperparameter tuning

FULLY AUTOMATED MLOPS



Arturo Opsetmoen Amador  
Senior Consultant - Machine Learning



# What is a hyperparameter?

Hyperparameters are tunable values that control the learning process

- Not learned during the training process
- Set before training an ML model

Examples:

- Model architecture in a Neural Network
- Number of branches in a decision tree
- Learning rate

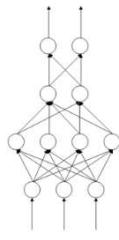
# What is hyperparameter tuning?



## Hyperparameters



Layers = 5  
Neurons = 512  
Learning rate = 0.1



## Model Parameters



Weights  
optimization



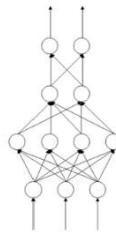
## Scoring

89%



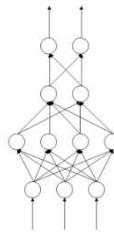
91%

Weights  
optimization



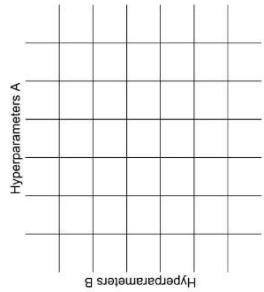
90%

Weights  
optimization

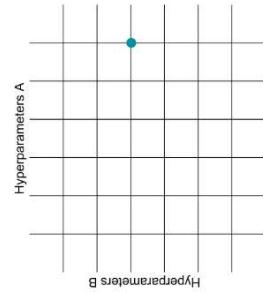


# Hyperparameter tuning methods

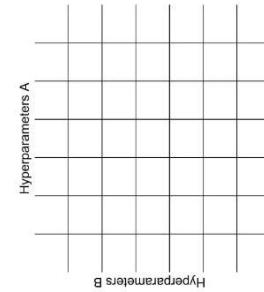
- Grid Search



- Random Search



- Bayesian Optimization



# Automate hyperparameter tuning

AUTOMATE

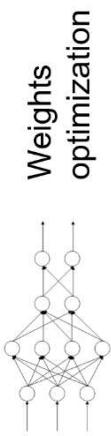


Hyperparameters

Layers = 5  
Neurons = 512  
Learning rate = 0.1



Model Parameters



Weights optimization

Layers = 5  
Neurons = 1024  
Learning rate = 0.01



Weights optimization

Layers = 6  
Neurons = 2048  
Learning rate = 0.01



Weights optimization



Scoring



# Automated hyperparameter tuning steps

- Need to define:
  - Set of hyperparameters to optimize
  - Search space for each parameter
  - A performance metrics to optimize
  - Stopping Criteria

Define hyperparam. to optimize

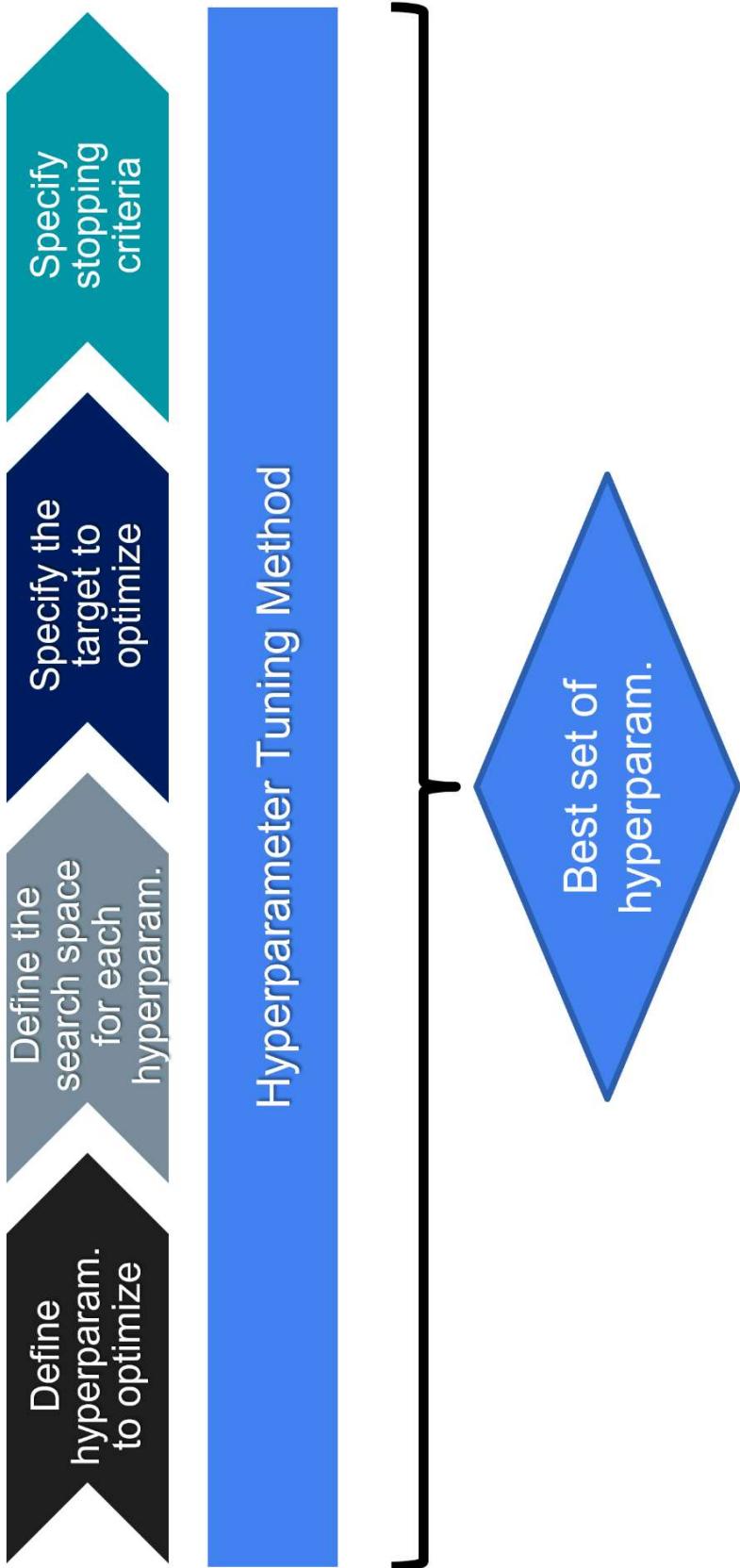
Define the search space for each hyperparam.

Specify the target to optimize

Specify stopping criteria

## Hyperparameter Tuning Method

# Automatically finding the best set of hyperparameters



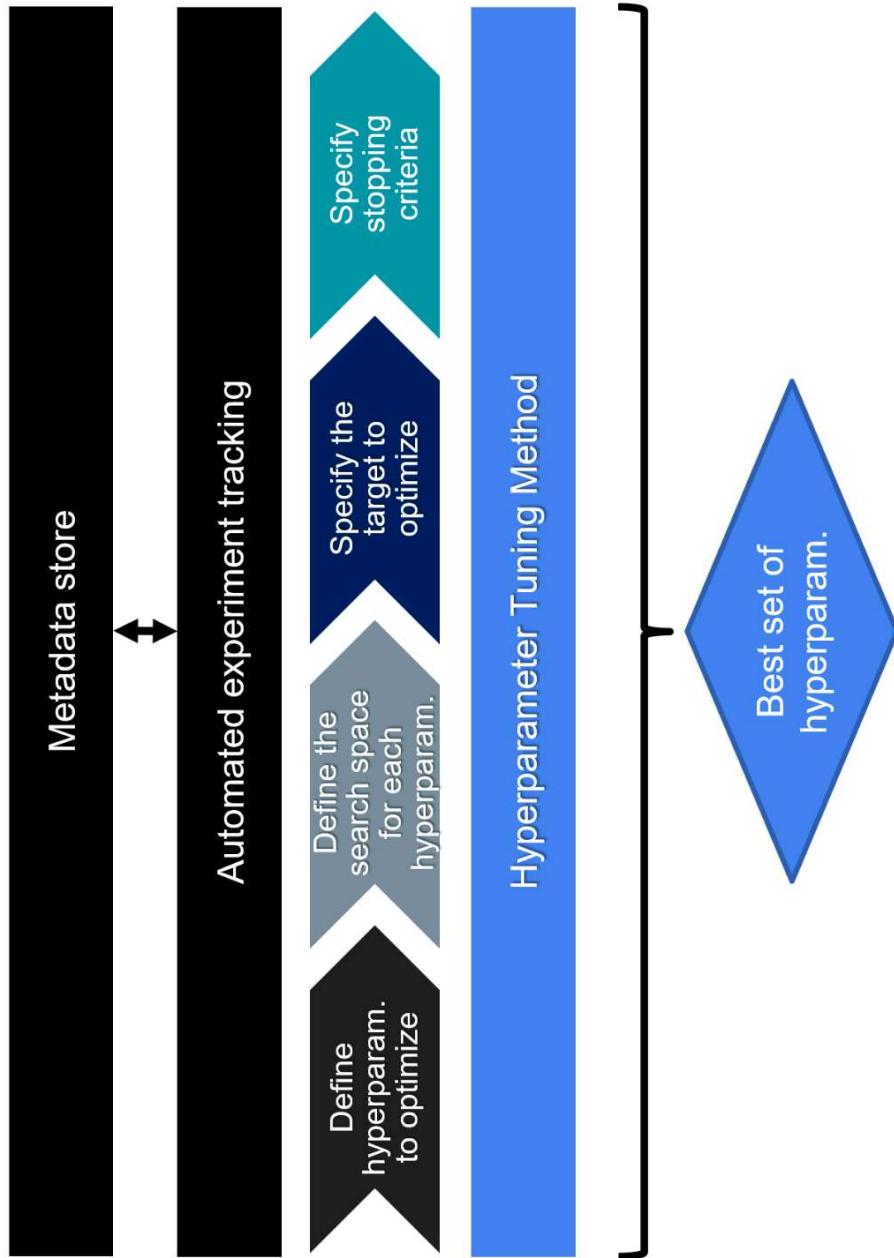
# Hyperparameters and environment symmetry

Development &  
Experimentation

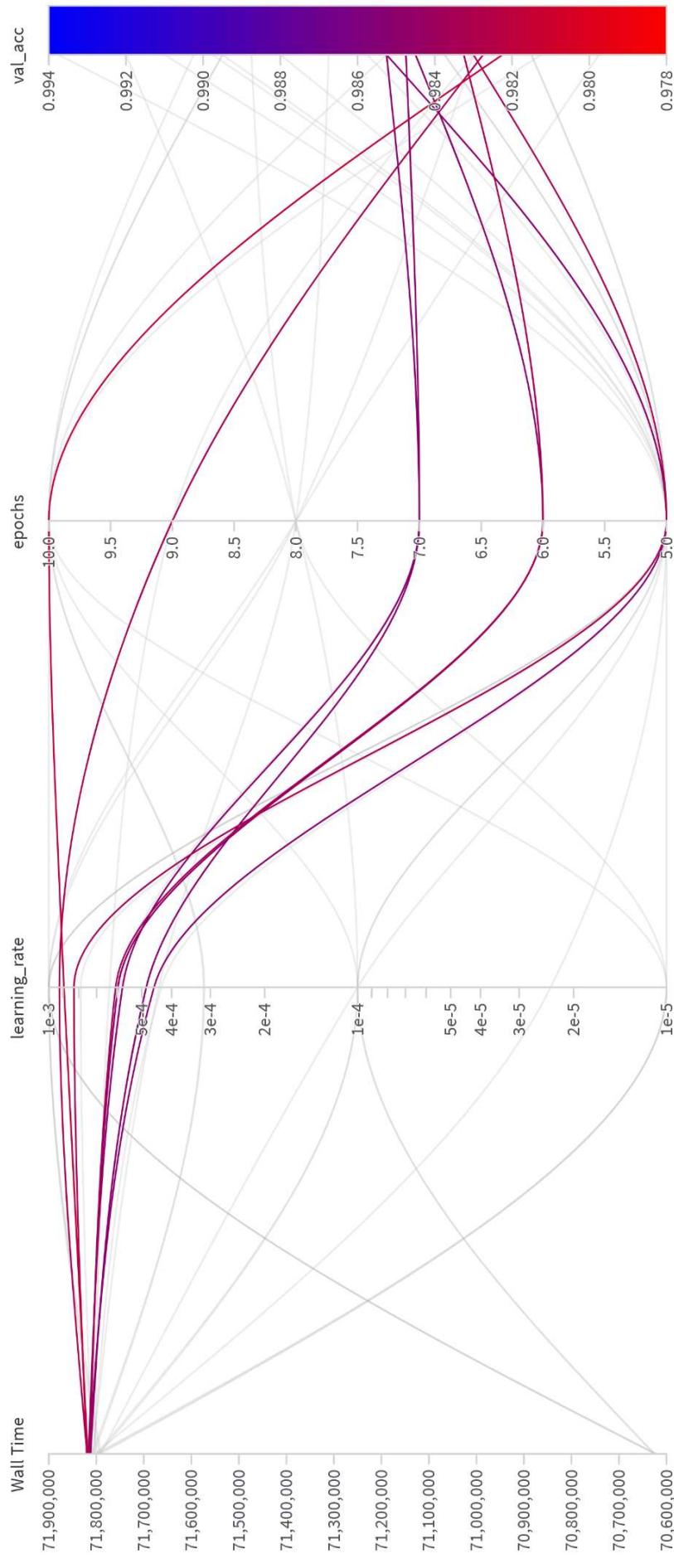
Production

Hyperparameter Tuning

# Hyperparameter tuning - Experiment tracking



# Example - Hyperparameter visualization



# Let's practice!

FULLY AUTOMATED MLOPS