

# Introducing full automation and best practices to MLOps

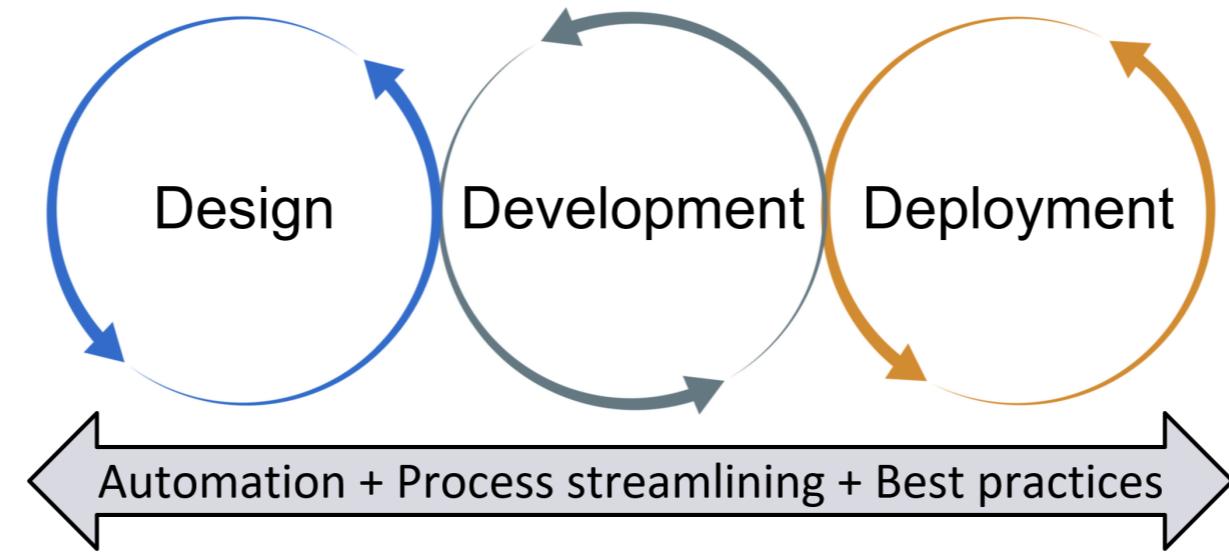
FULLY AUTOMATED MLOPS



**Arturo Opsetmoen Amador**

Senior Consultant - Machine Learning

# The MLOps lifecycle

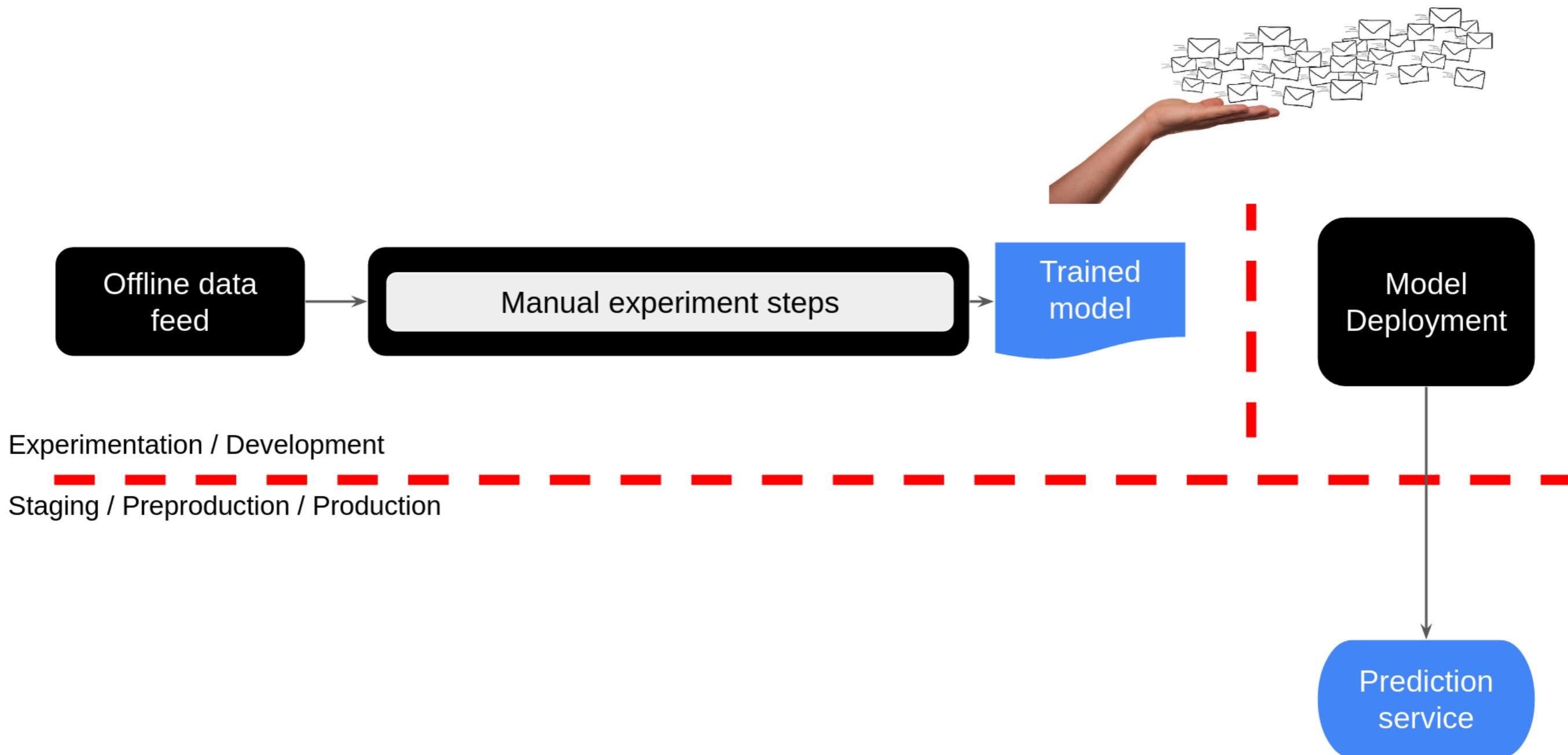


# Maturity levels in MLOps

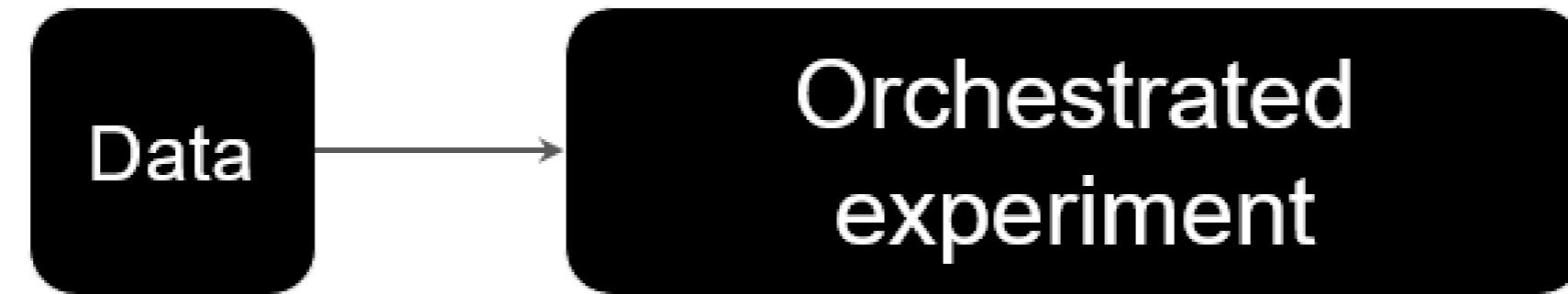
The maturity levels include:

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

# Manual ML workflow - Ad hoc experimentation

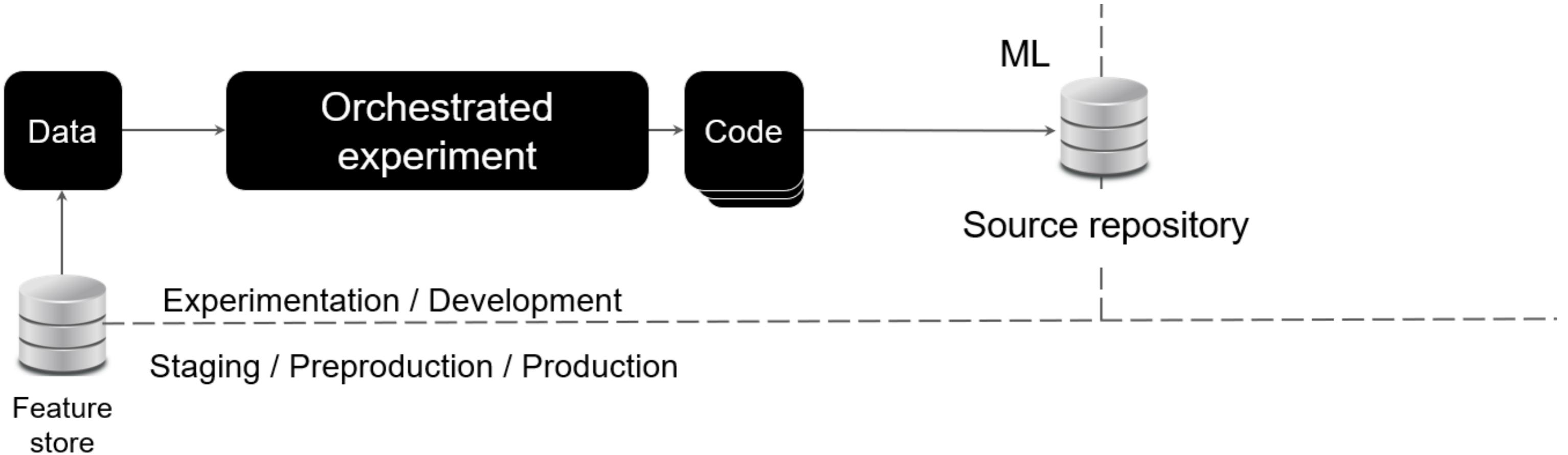


# Semi-automated ML workflow

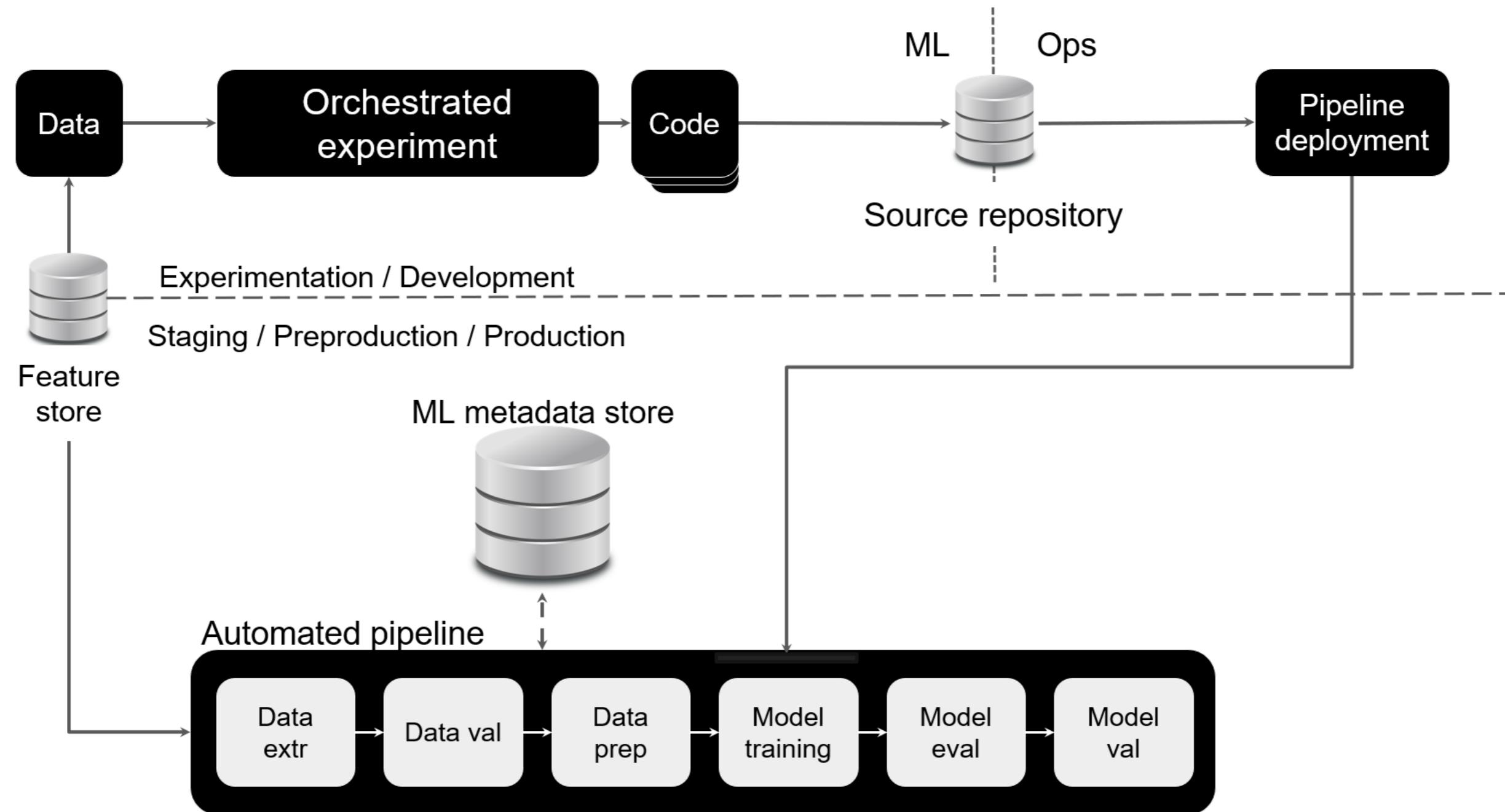


Experimentation / Development

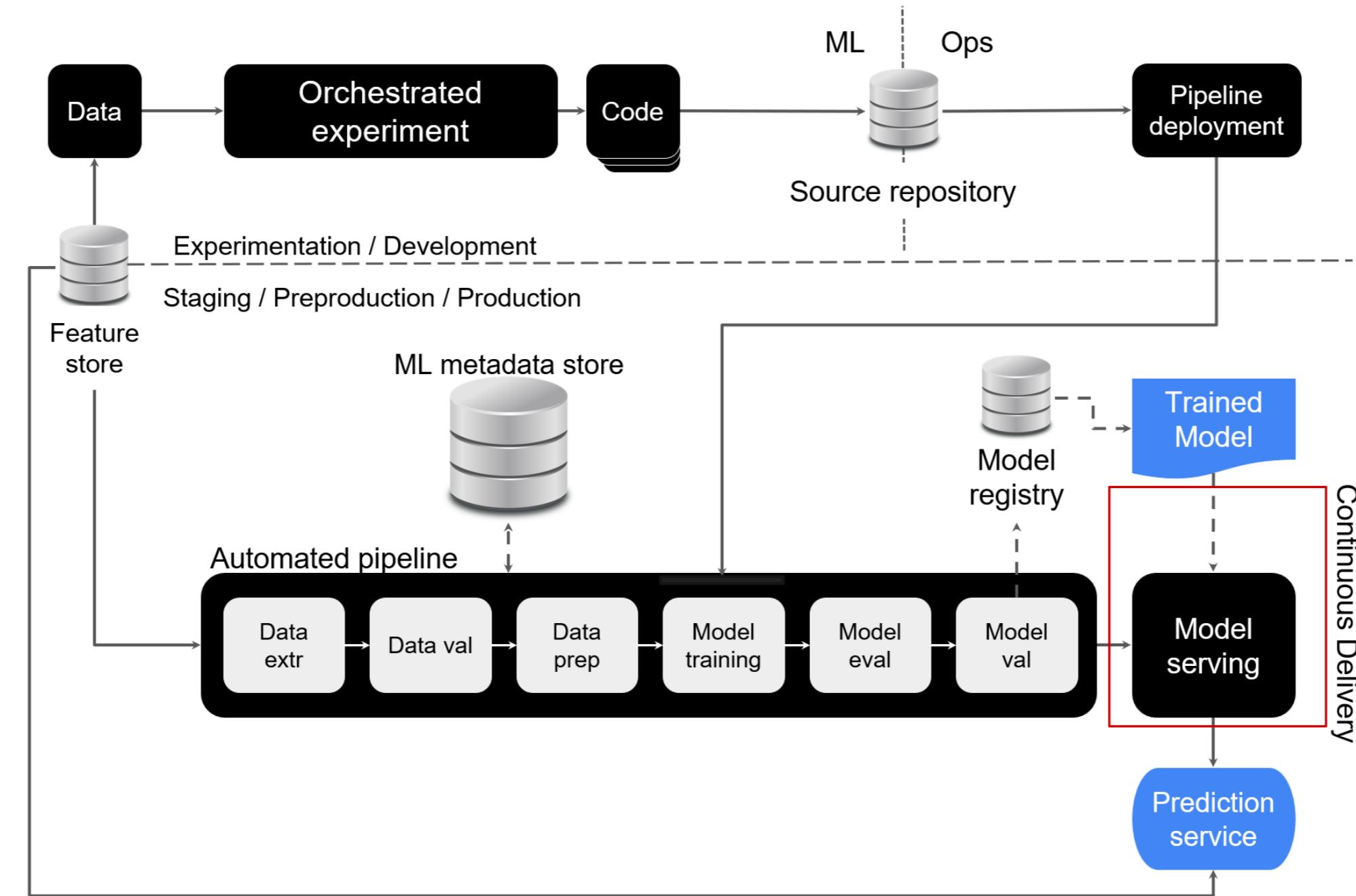
# Semi-automated ML workflow



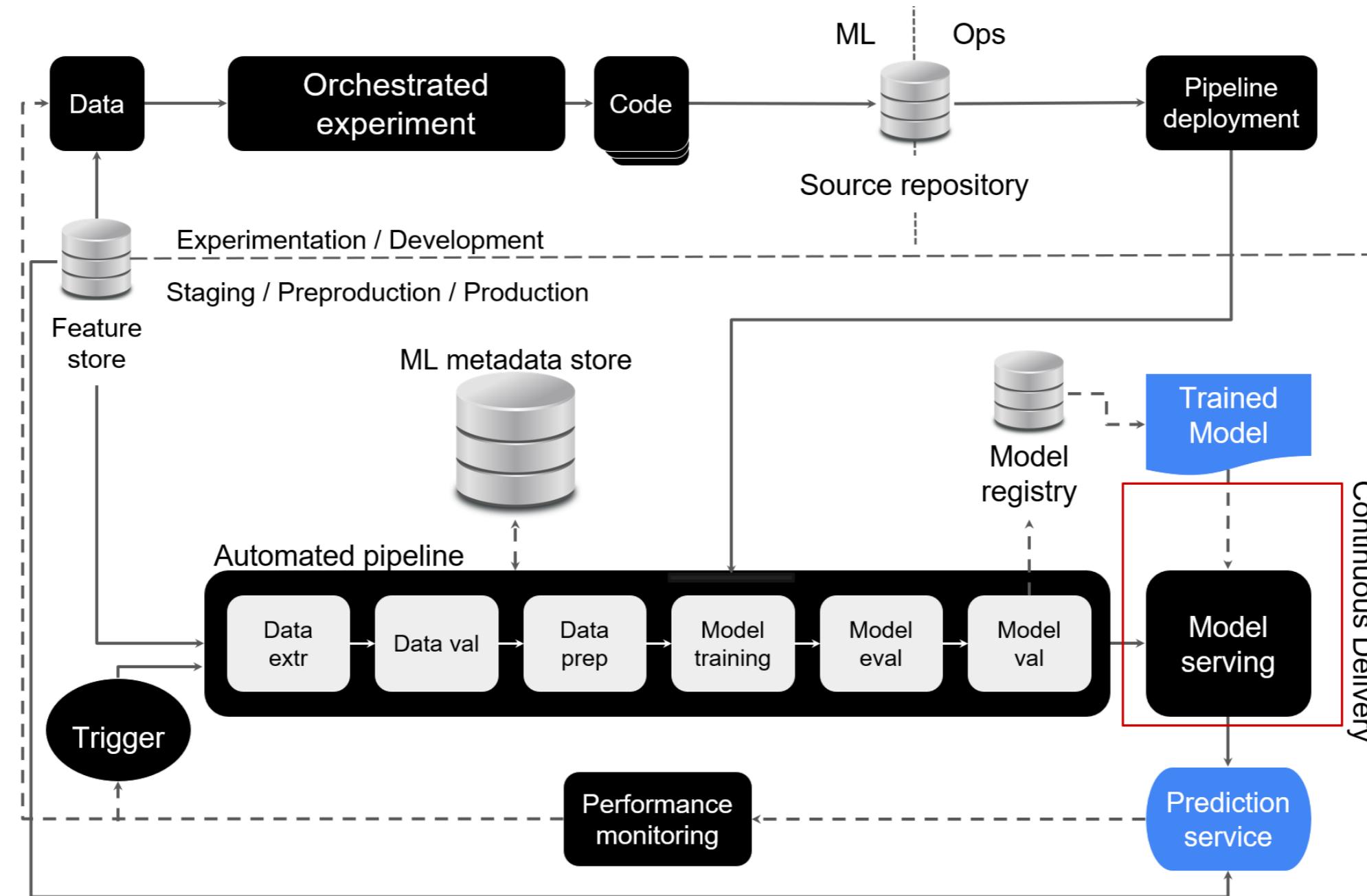
# Semi-automated ML workflow



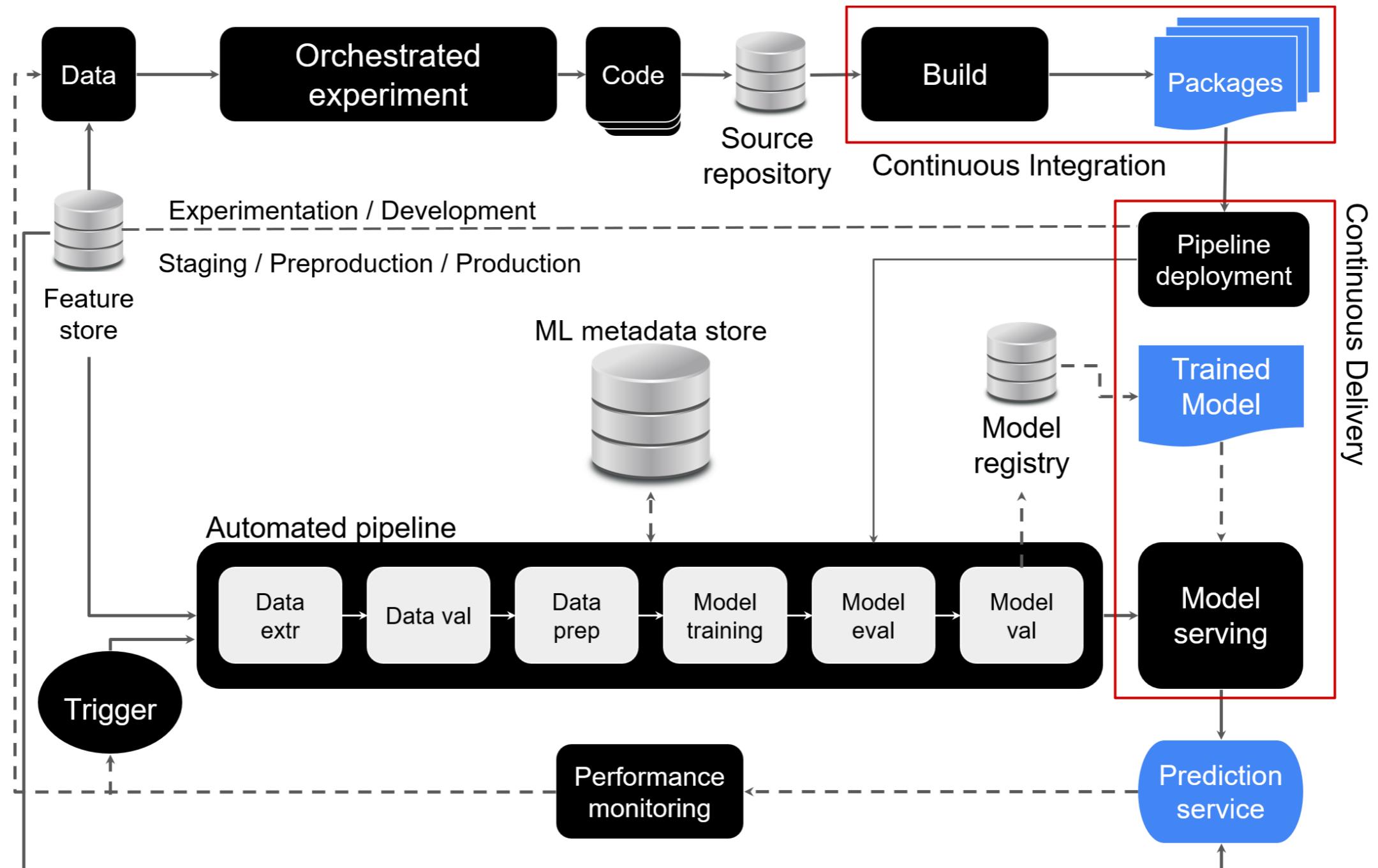
# Semi-automated ML workflow



# Semi-automated ML workflow



# Fully automated ML workflow



# Automation in the ML life cycle - Design

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

Good practices:

- Apply reproducible processes
- Write detailed documentation

# Automation in the ML life cycle - Development

	Activity in Workflow	Fully Automated
Development & Experimentation	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

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

Use:

- Automated testing
- CI/CD/CT/CM

# **Let's practice!**

**FULLY AUTOMATED MLOPS**

# The automation, monitoring, incident response pattern

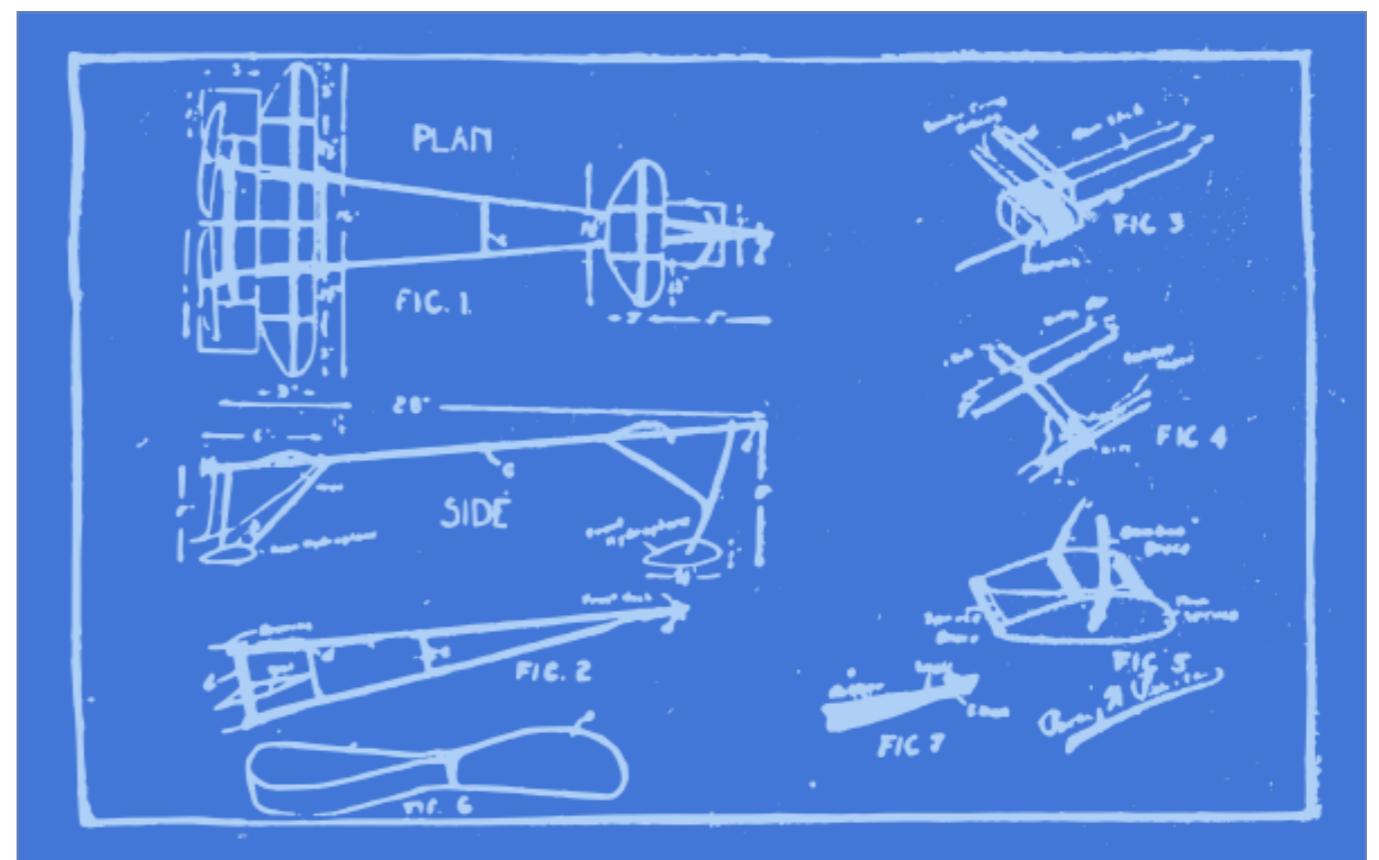
FULLY AUTOMATED MLOPS



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

# What is a software design pattern?

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



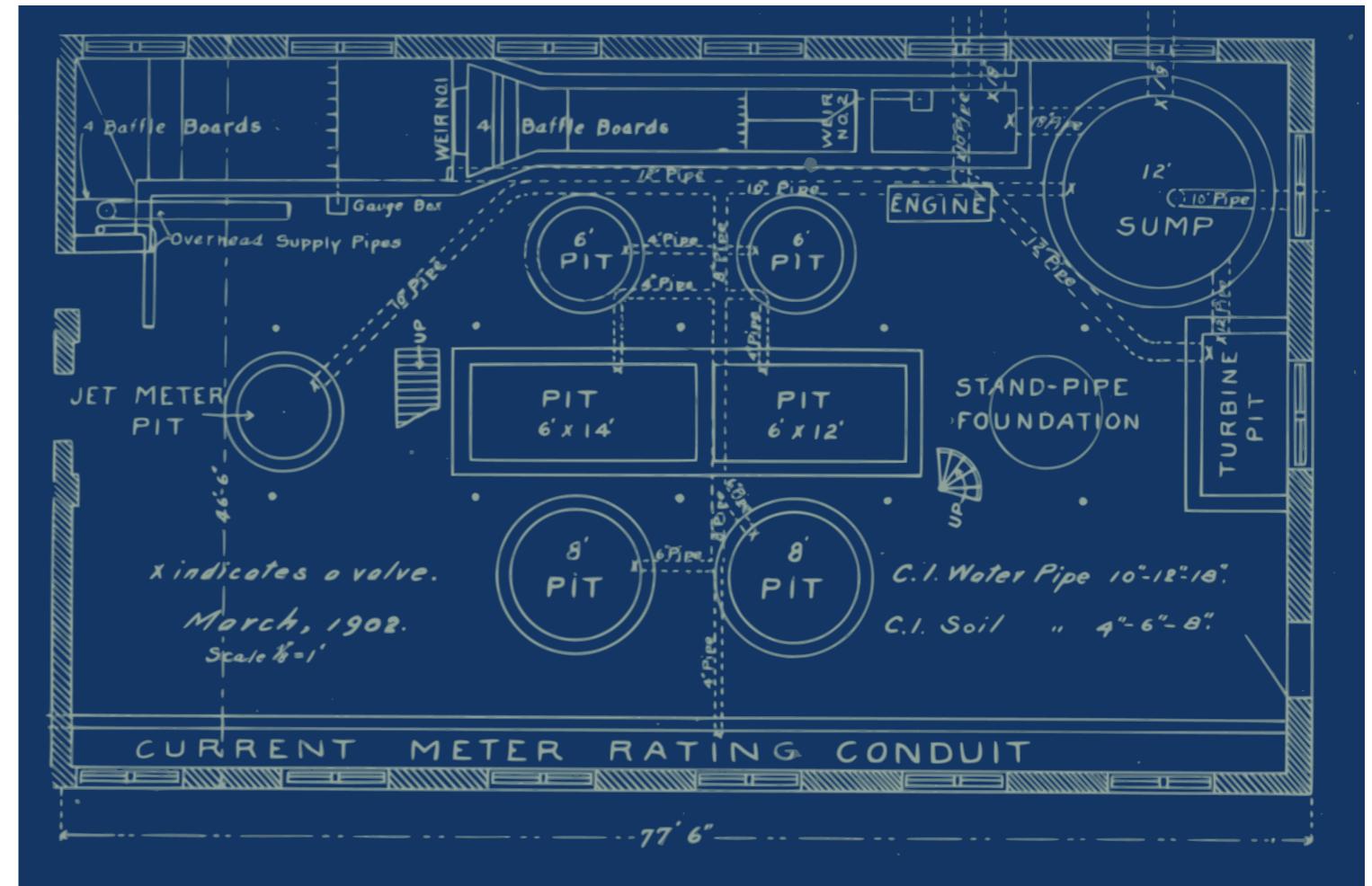
# Automate, monitor, respond



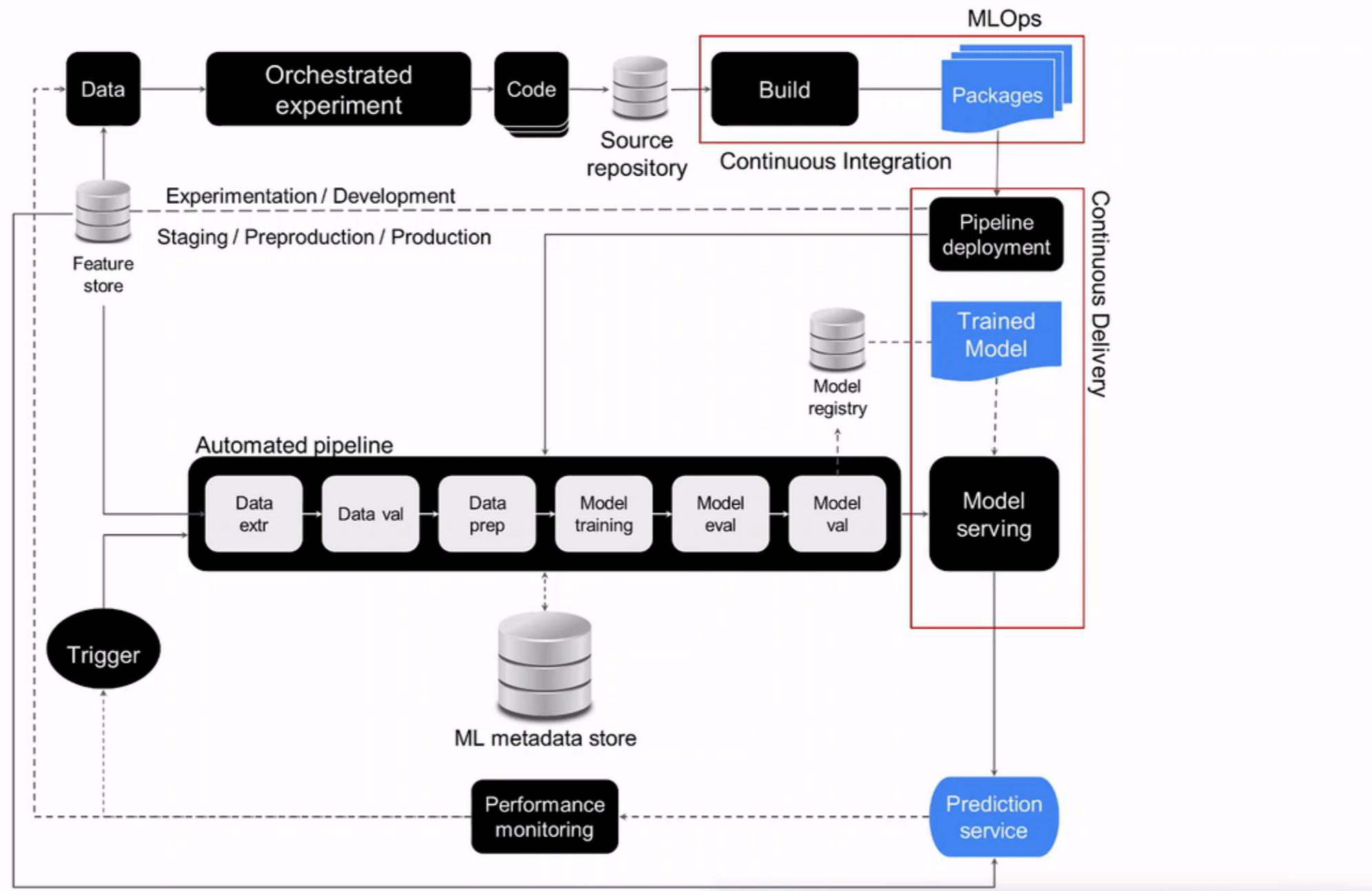
- Improves the reliability of the ML systems we design

# Three examples of a design pattern in MLOps

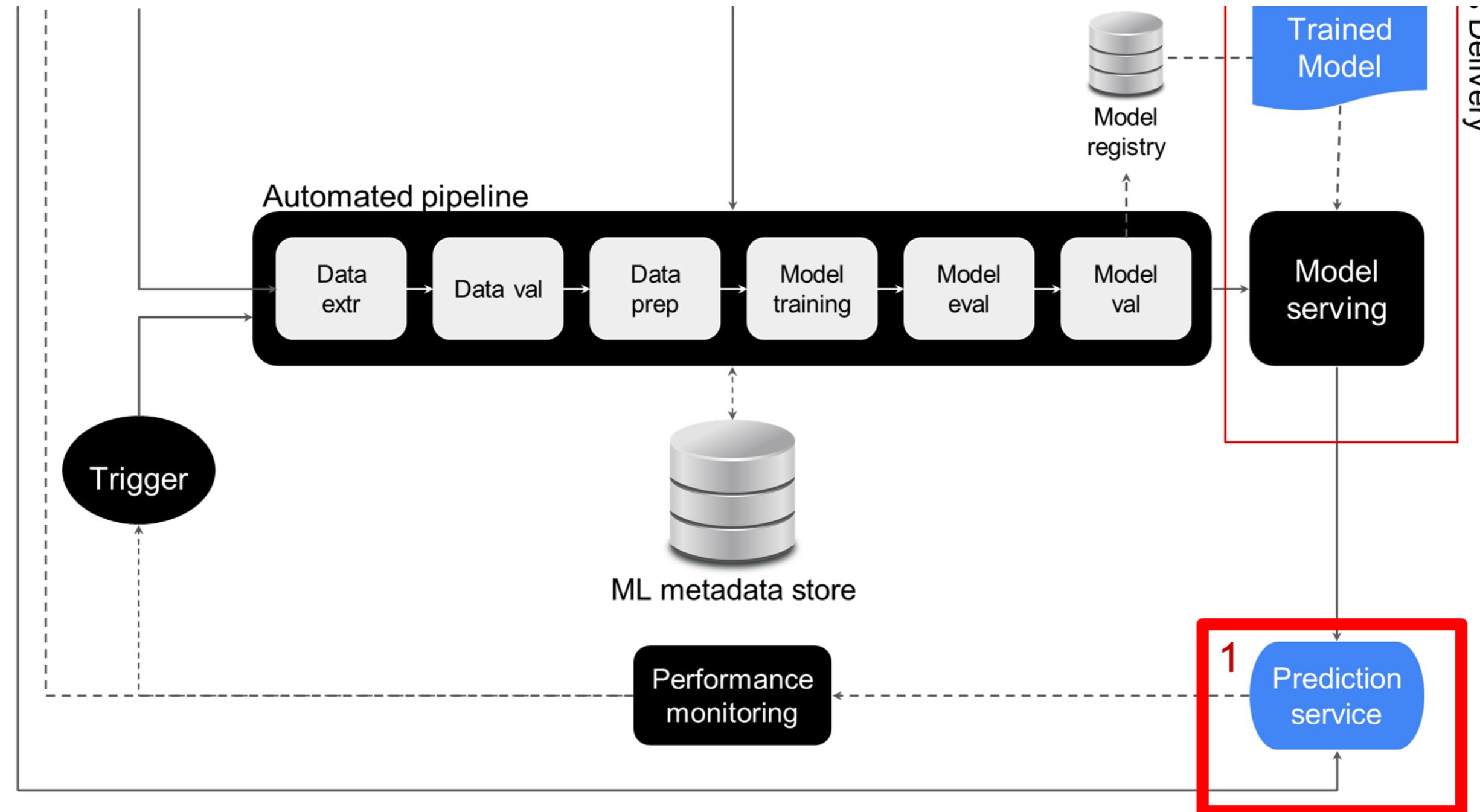
1. Automated Model Retraining
2. Model Rollback
3. Feature Imputation



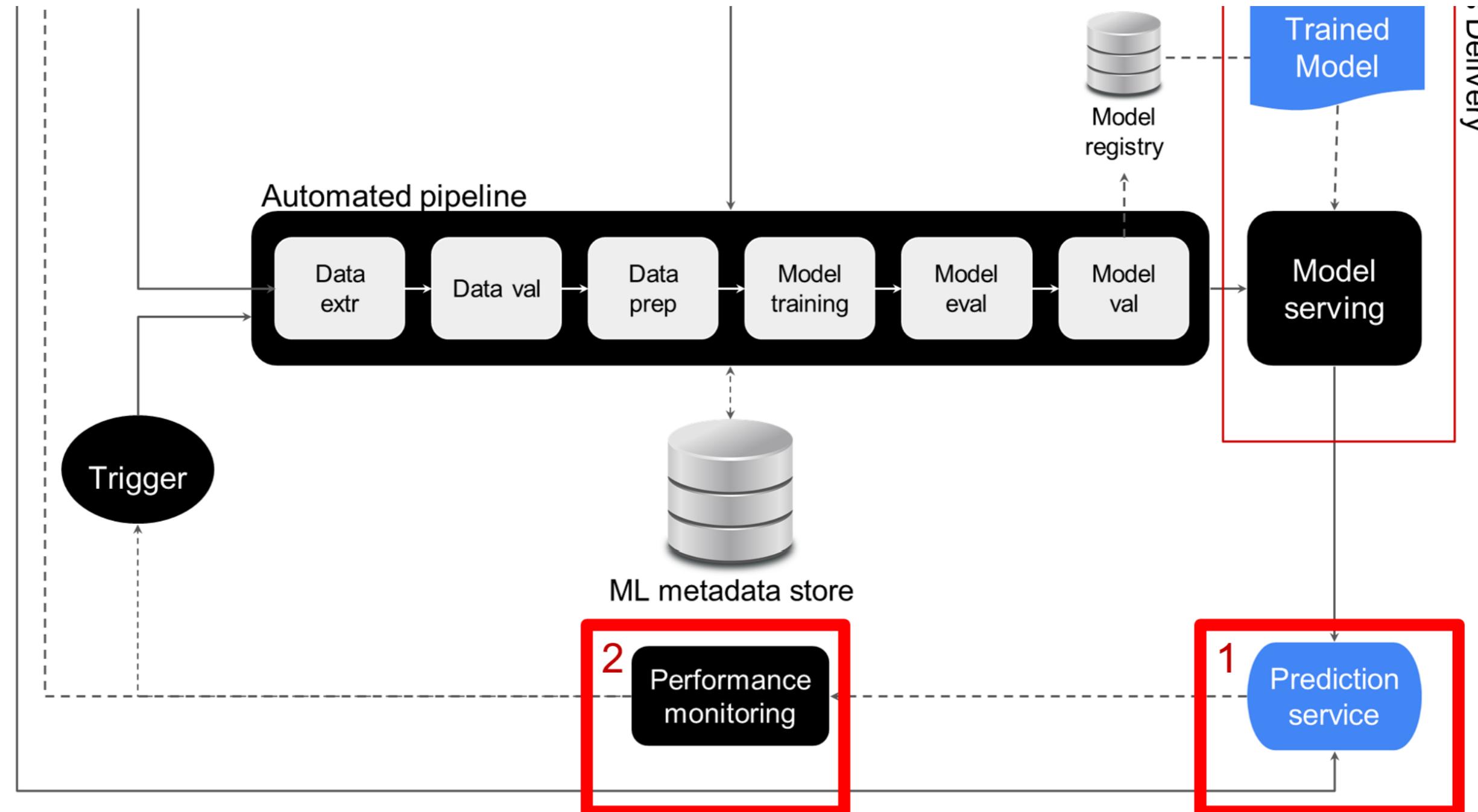
# 1. Automated model retraining



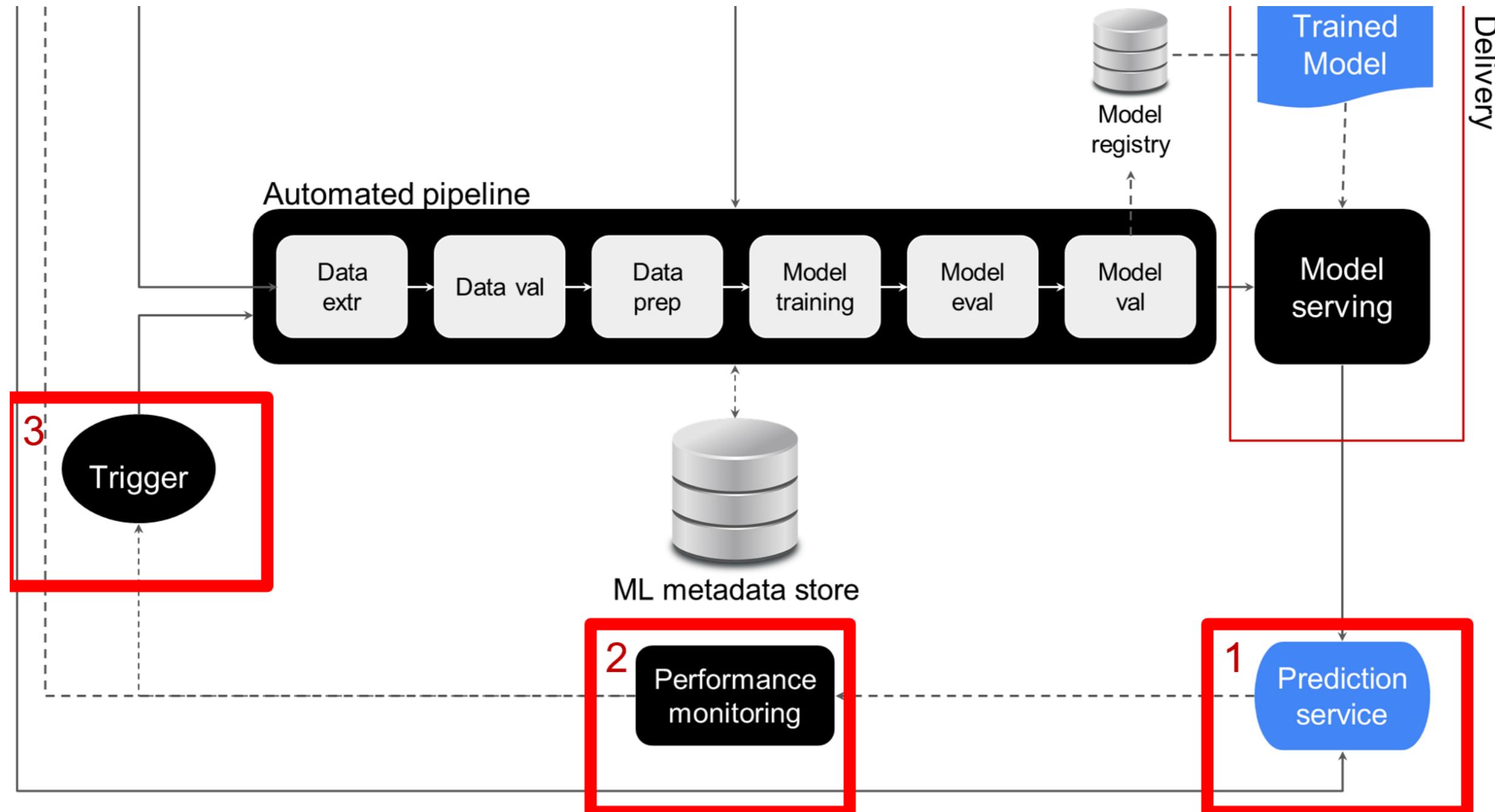
# 1. Automated model retraining - running predictions



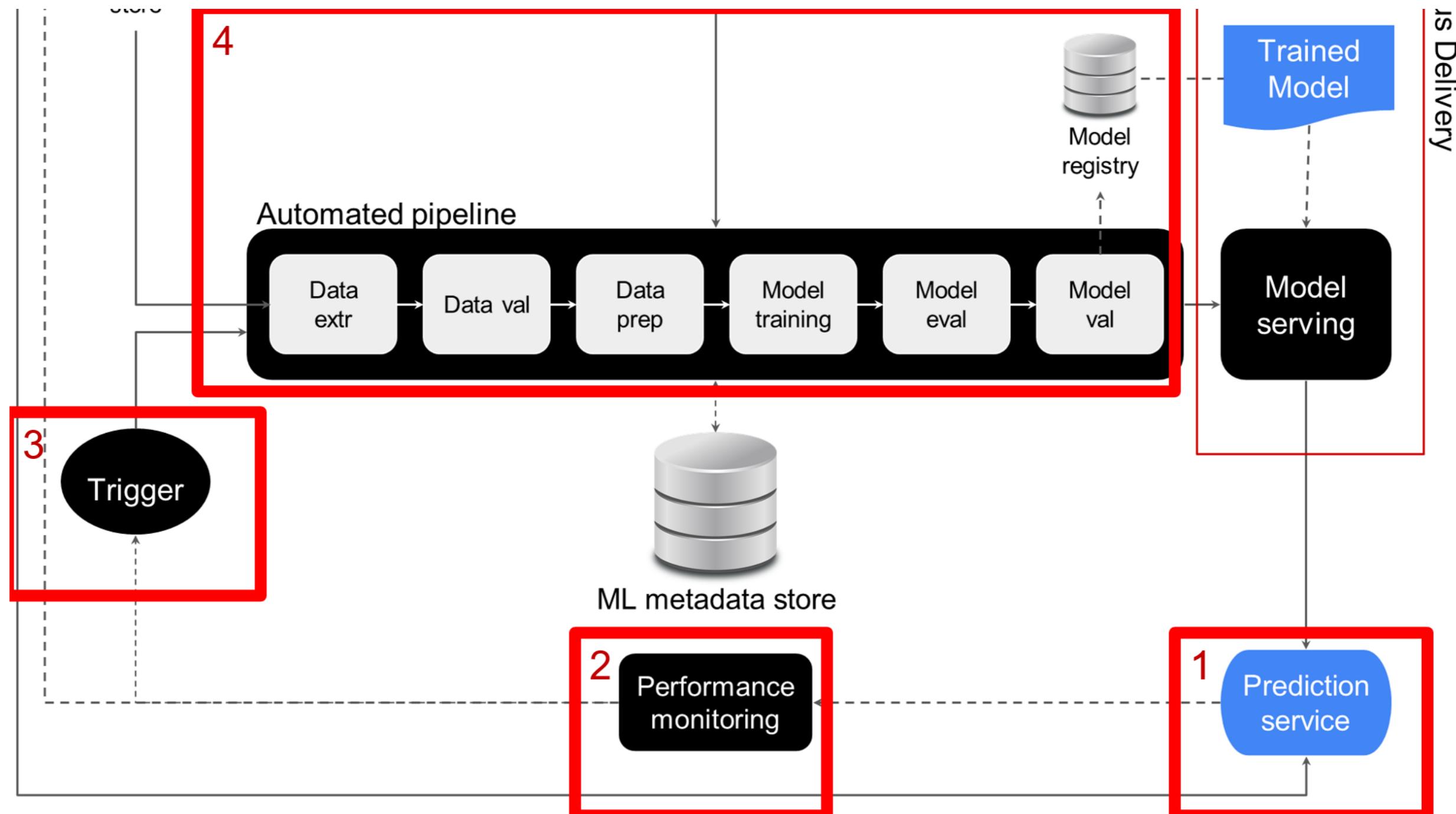
# 1. Automated model retraining - Monitoring



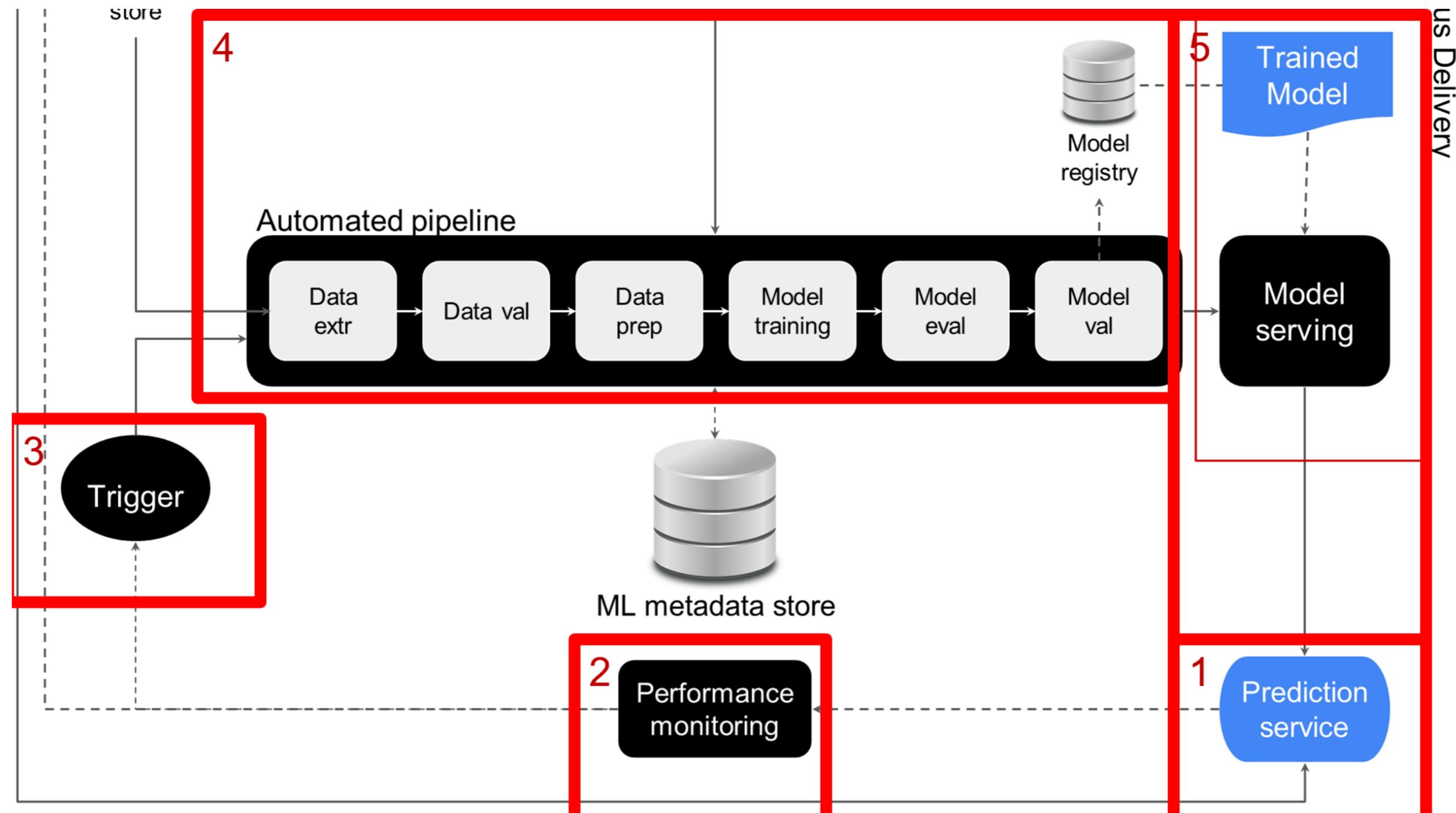
# 1. Automated model retraining - Trigger



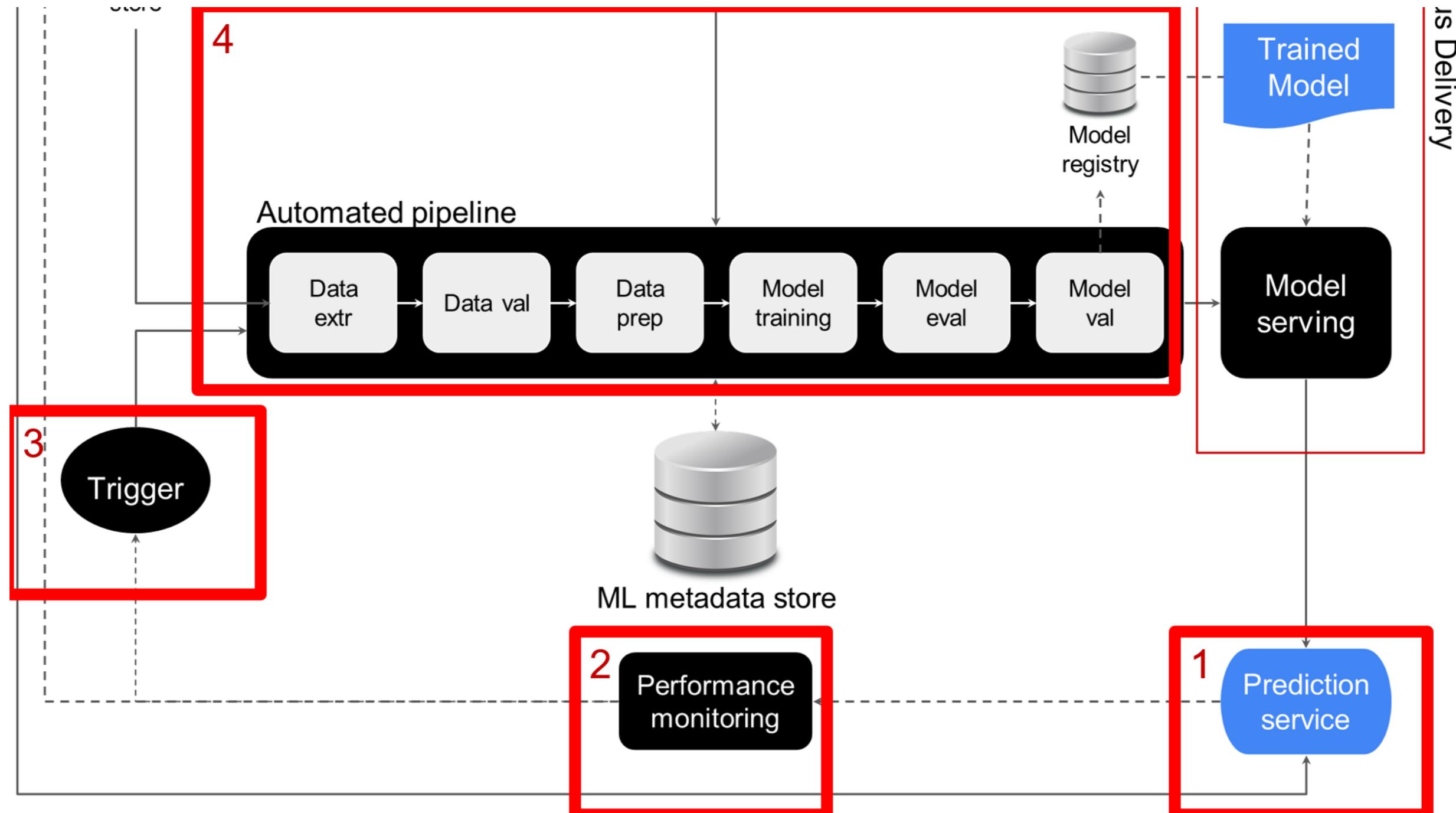
# 1. Automated model retraining - Automated pipeline



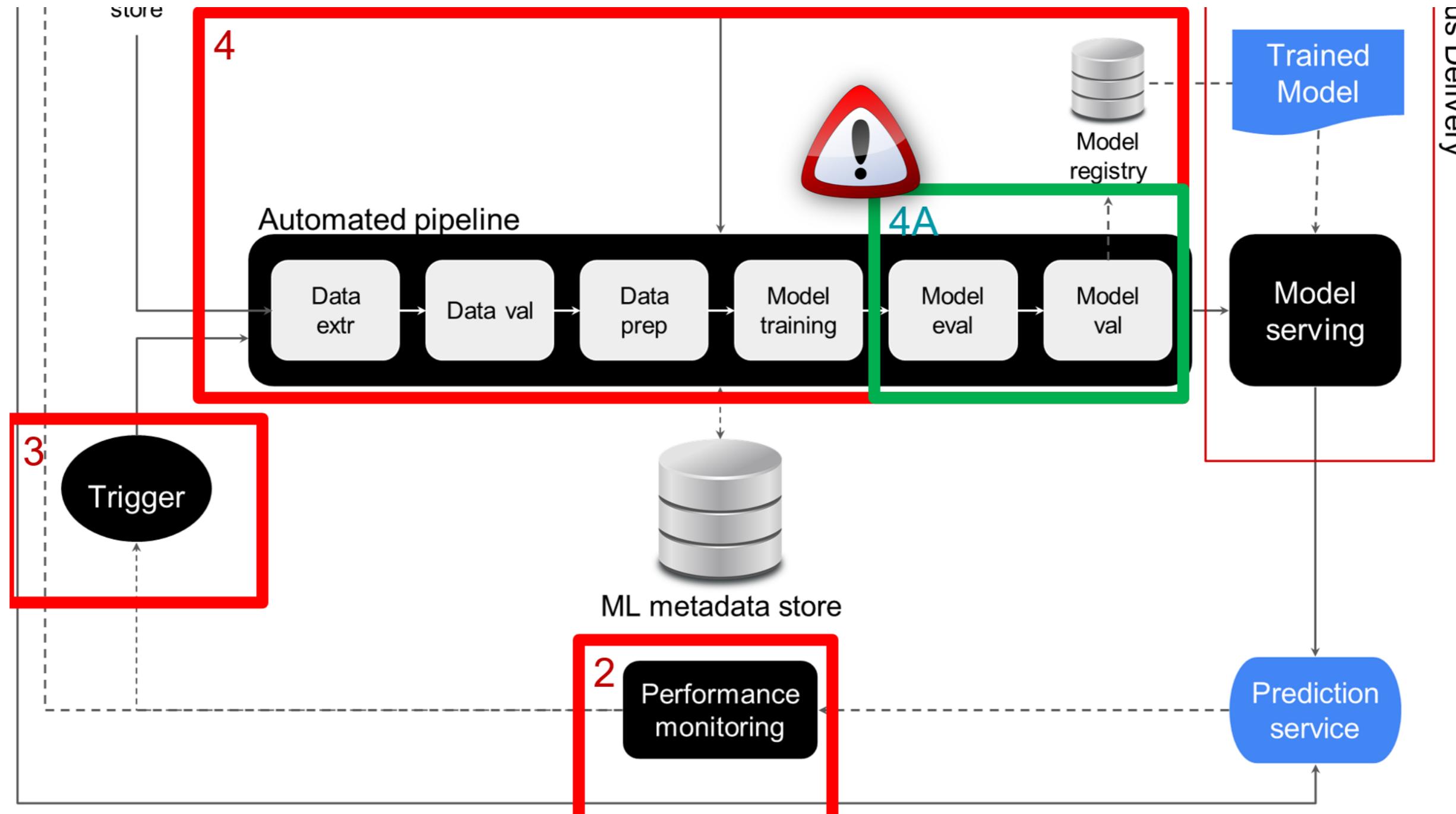
# 1. Automated model retraining - Deployment



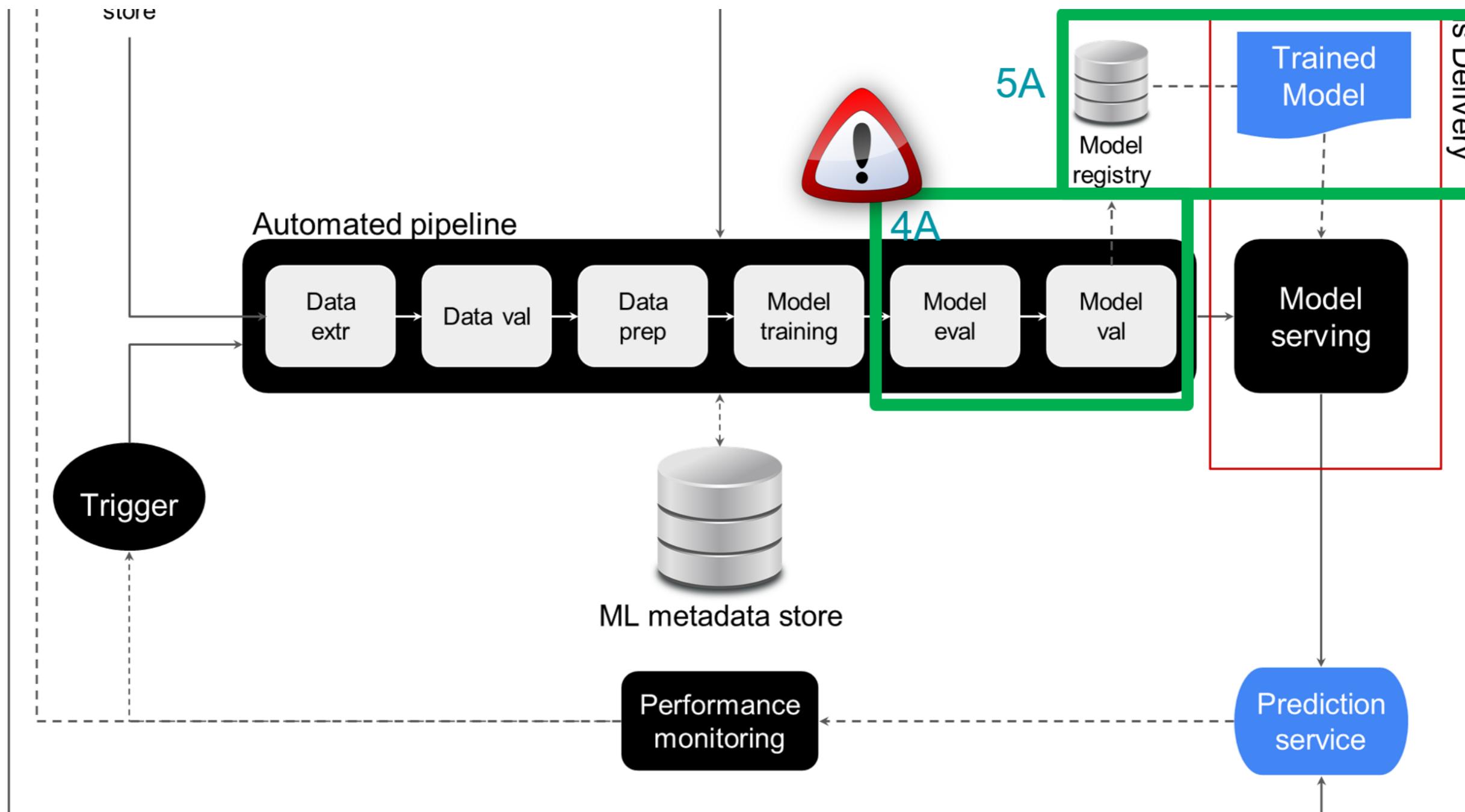
## 2. Model rollback



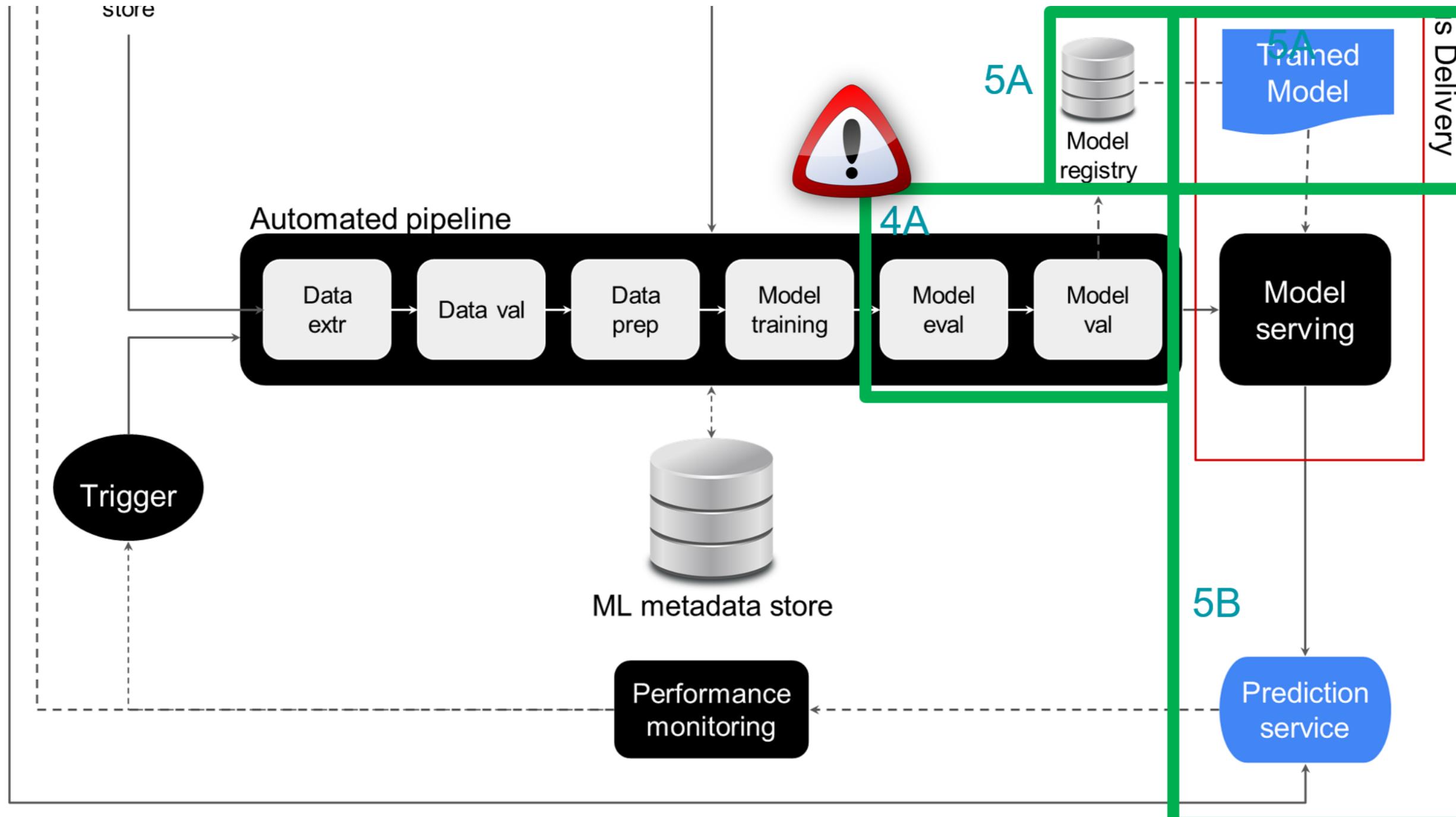
## 2. Model rollback - Validation fail



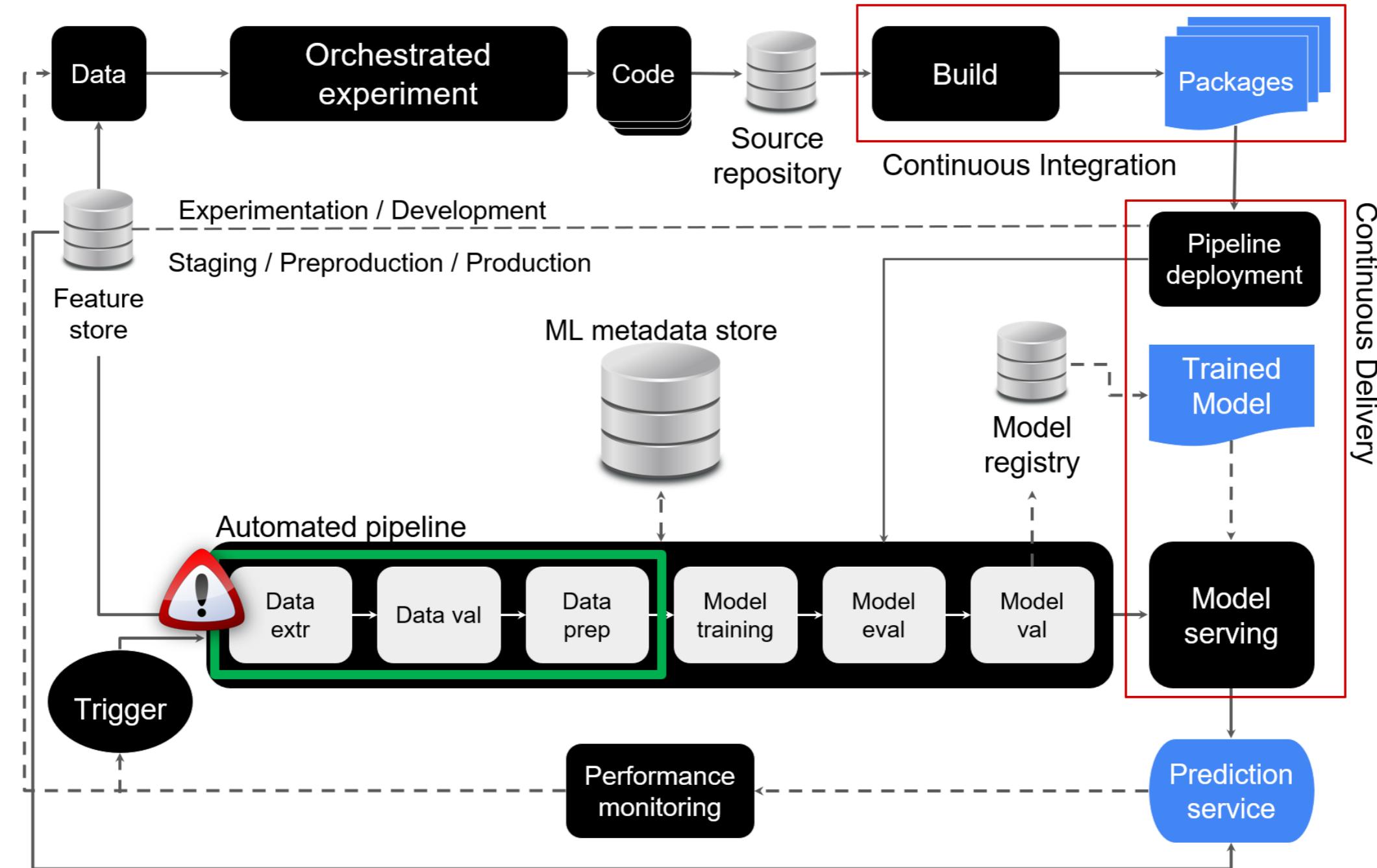
## 2. Model rollback - Last functional model



## 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

• • •

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

•  
•  
•

179	NaN
139	NaN
285	NaN
401	NaN

NaN	Low
NaN	Very High
NaN	Very High
NaN	Medium

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

### 3. Feature imputation - Defective features

Feature A	Feature B
213	
234	NaN
132	NaN
182	NaN
190	Red

• • •

179	NaN
139	NaN
285	NaN
401	NaN

Feature Y	Feature Z
	Low
NaN	High
731	Medium
NaN	Very Low
NaN	Very High

•

•

•

NaN	Low
NaN	Very High
NaN	Very High
NaN	Medium

- Detect failing features
- Apply feature imputation

### **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**

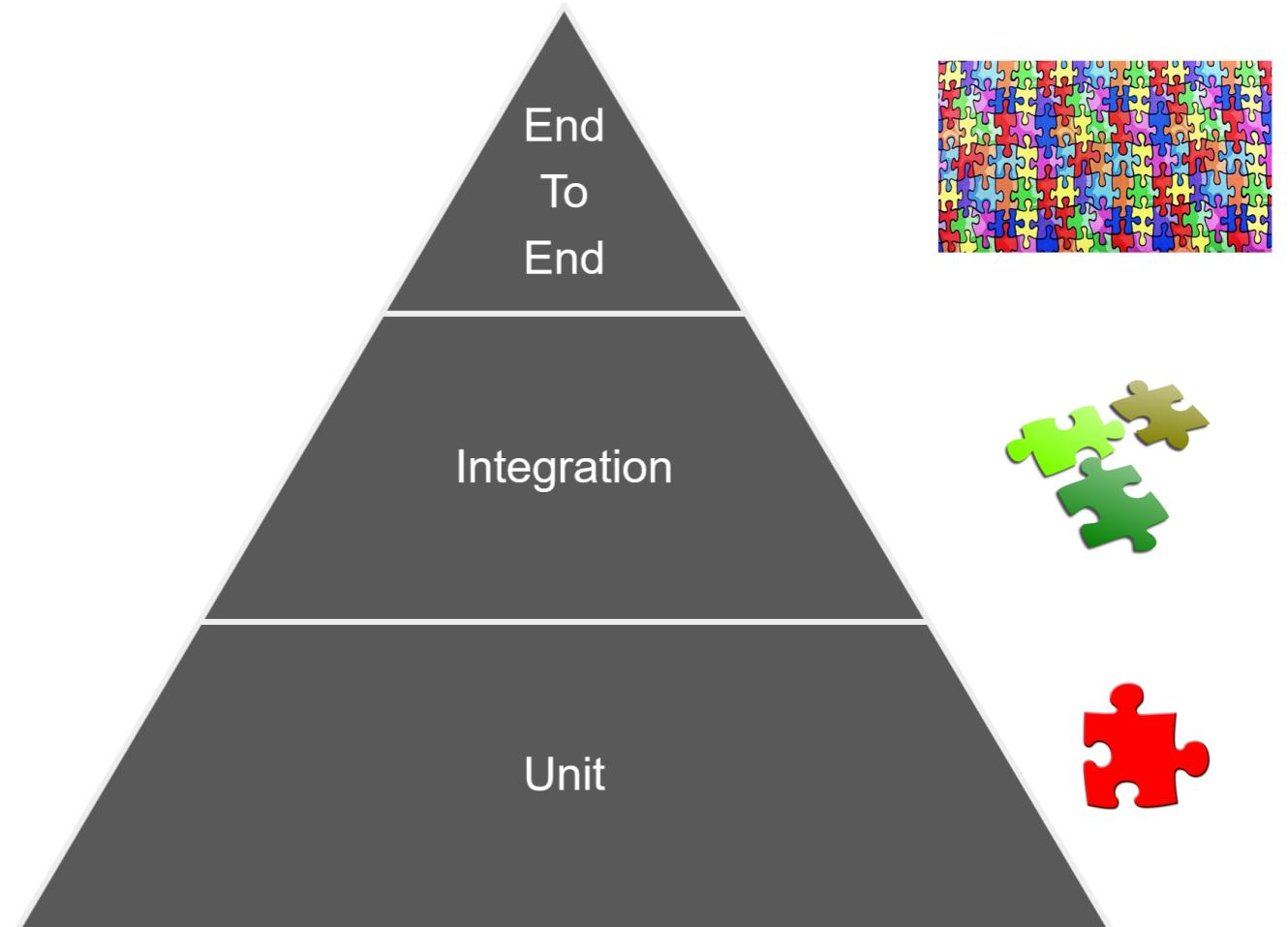
Senior Consultant - Machine Learning

# What is software testing?

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

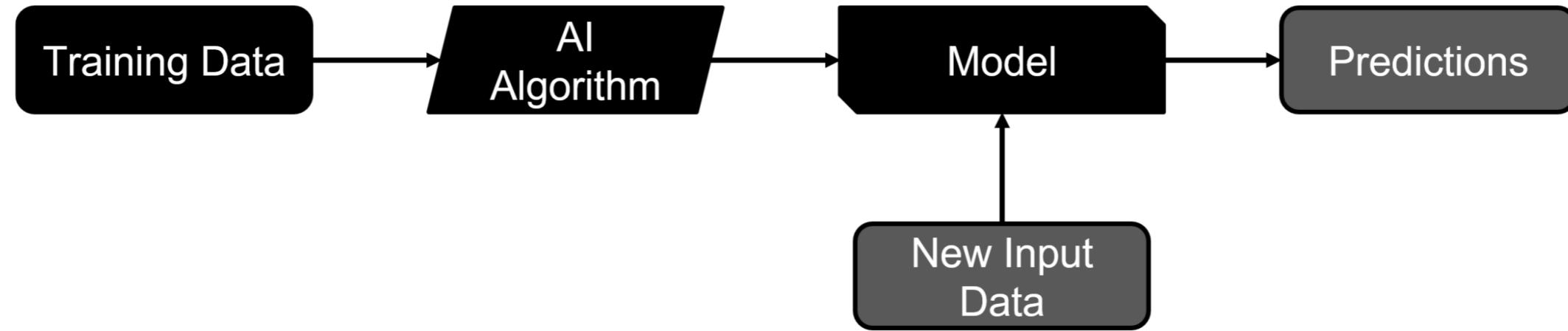
Three most common types of software testing:

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



# ML software has a different nature

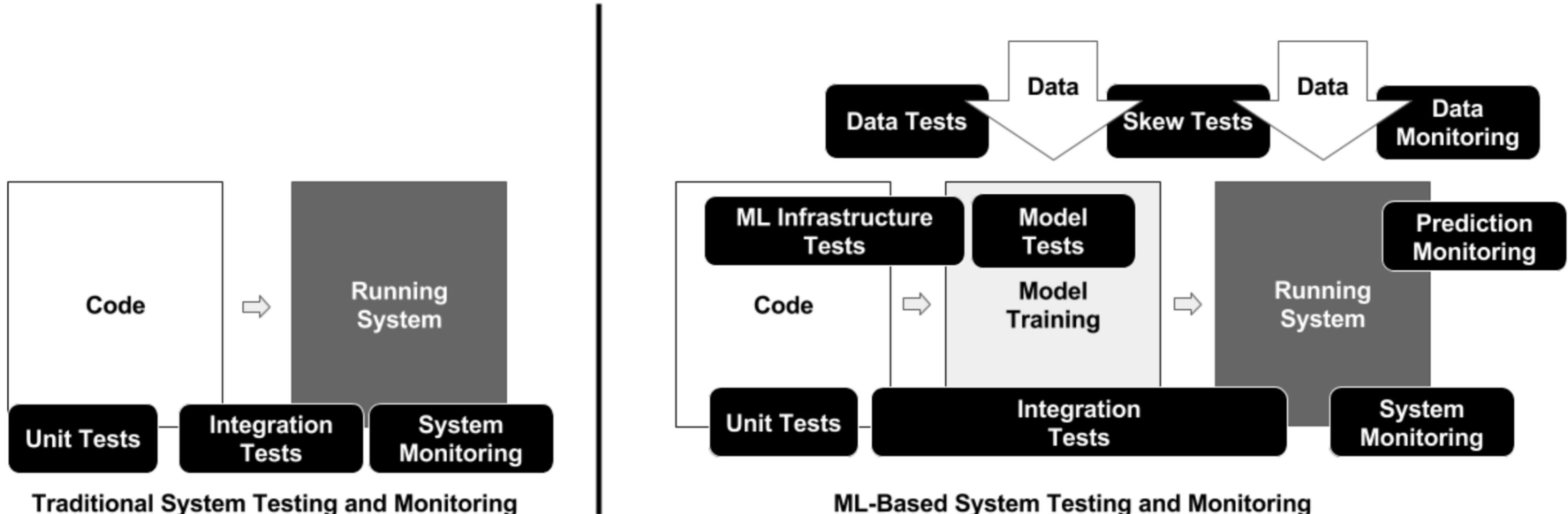
ML APPLICATIONS ARE NOT EXPLICITLY PROGRAMMED



They depend on:

- Data
- Models

# Testing in an MLOps system is different



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

# Testing ML systems

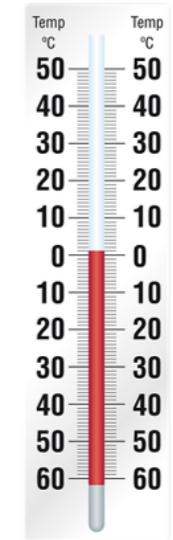
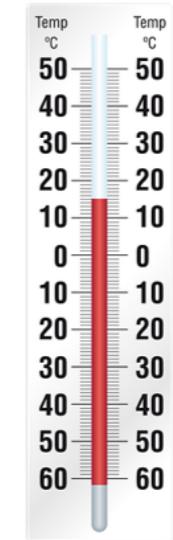
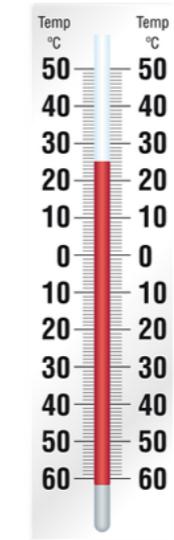
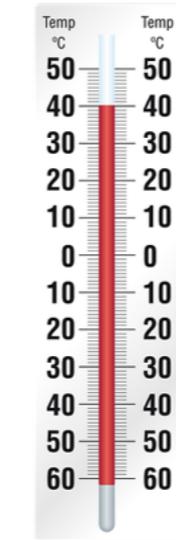
In addition to traditional software testing:

- 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



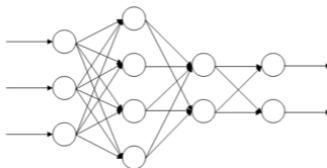
Layers = 5  
Neurons = 1024  
Learning rate = 0.01



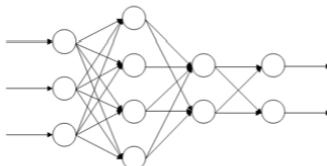
Layers = 6  
Neurons = 2048  
Learning rate = 0.01



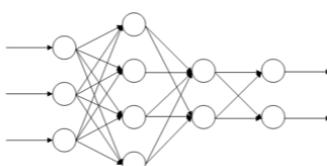
Model Parameters



Weights optimization



Weights optimization



Weights optimization



Scoring

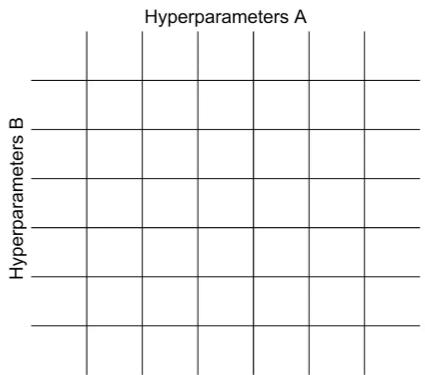
89%

91%

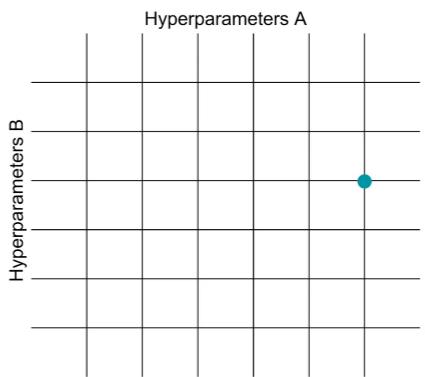
90%

# Hyperparameter tuning methods

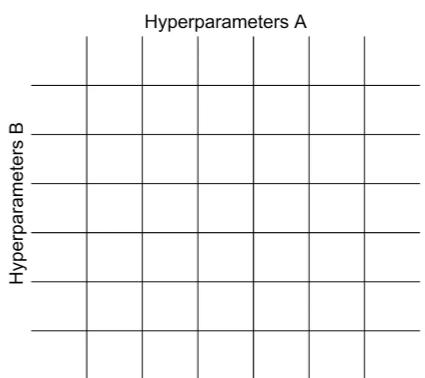
- Grid Search



- Random Search



- Bayesian Optimization



# Automate hyperparameter tuning

AUTOMATE



Hyperparameters



Layers = 5  
Neurons = 512  
Learning rate = 0.1



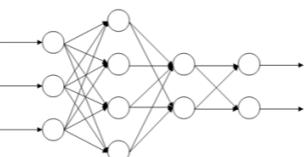
Layers = 5  
Neurons = 1024  
Learning rate = 0.01



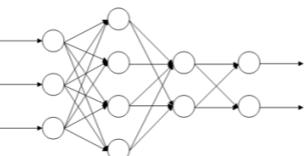
Layers = 6  
Neurons = 2048  
Learning rate = 0.01



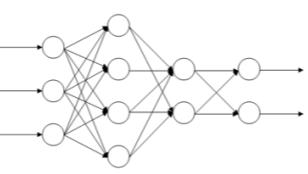
Model Parameters



Weights optimization



Weights optimization



Weights optimization



Scoring

89%

91%

90%

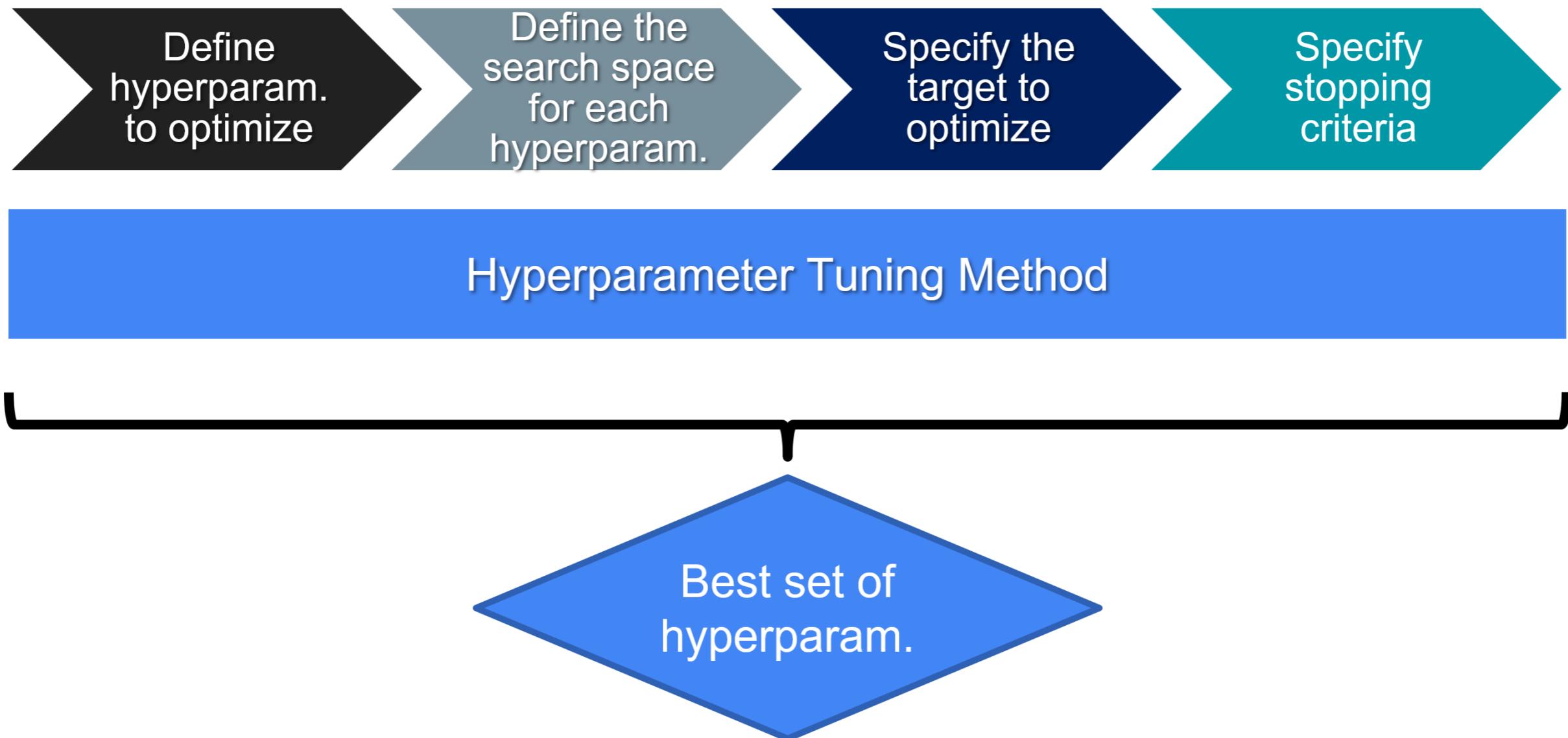
# Automated hyperparameter tuning steps

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



Hyperparameter Tuning Method

# Automatically finding the best set of hyperparameters



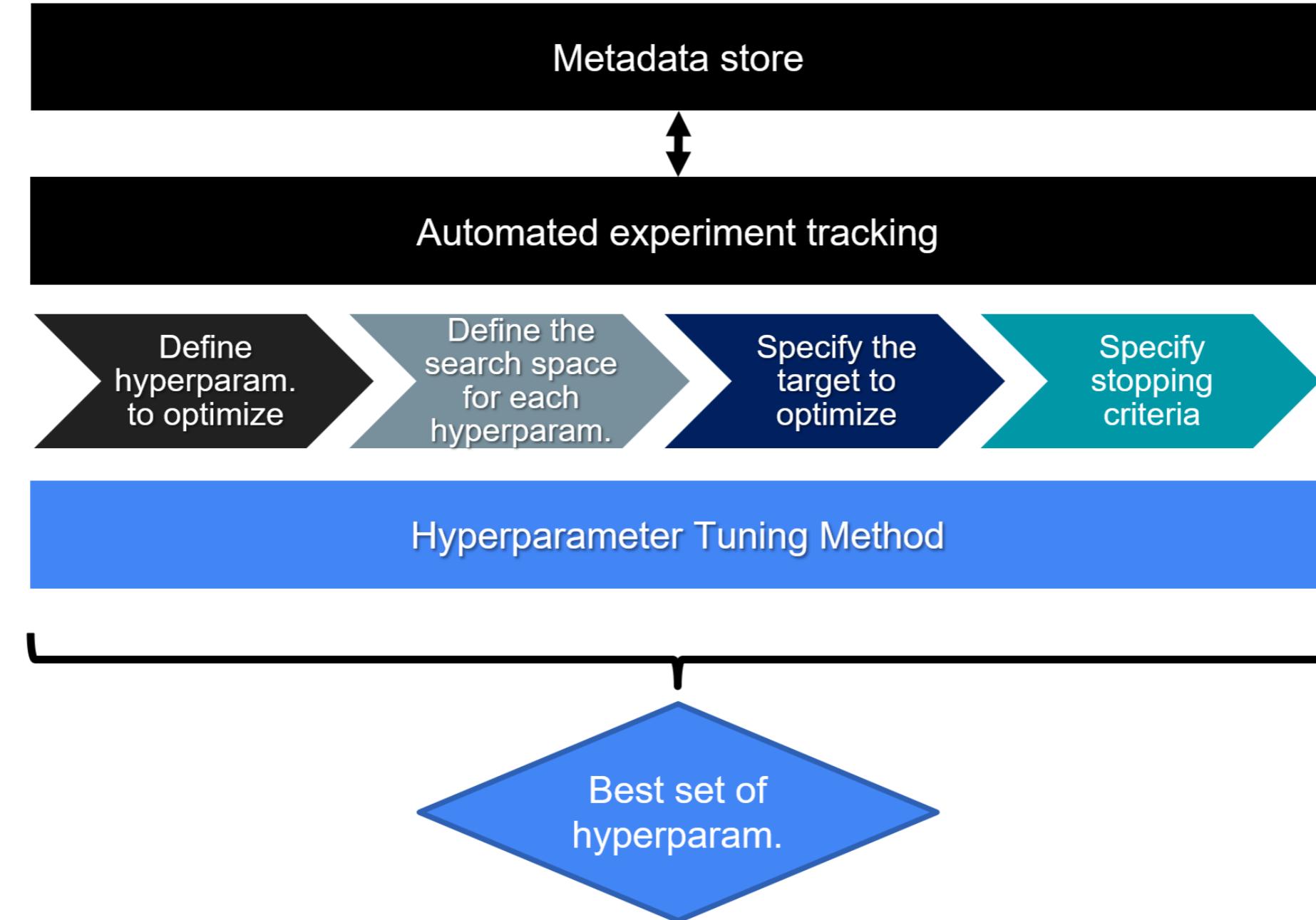
# Hyperparameters and environment symmetry

Development &  
Experimentation

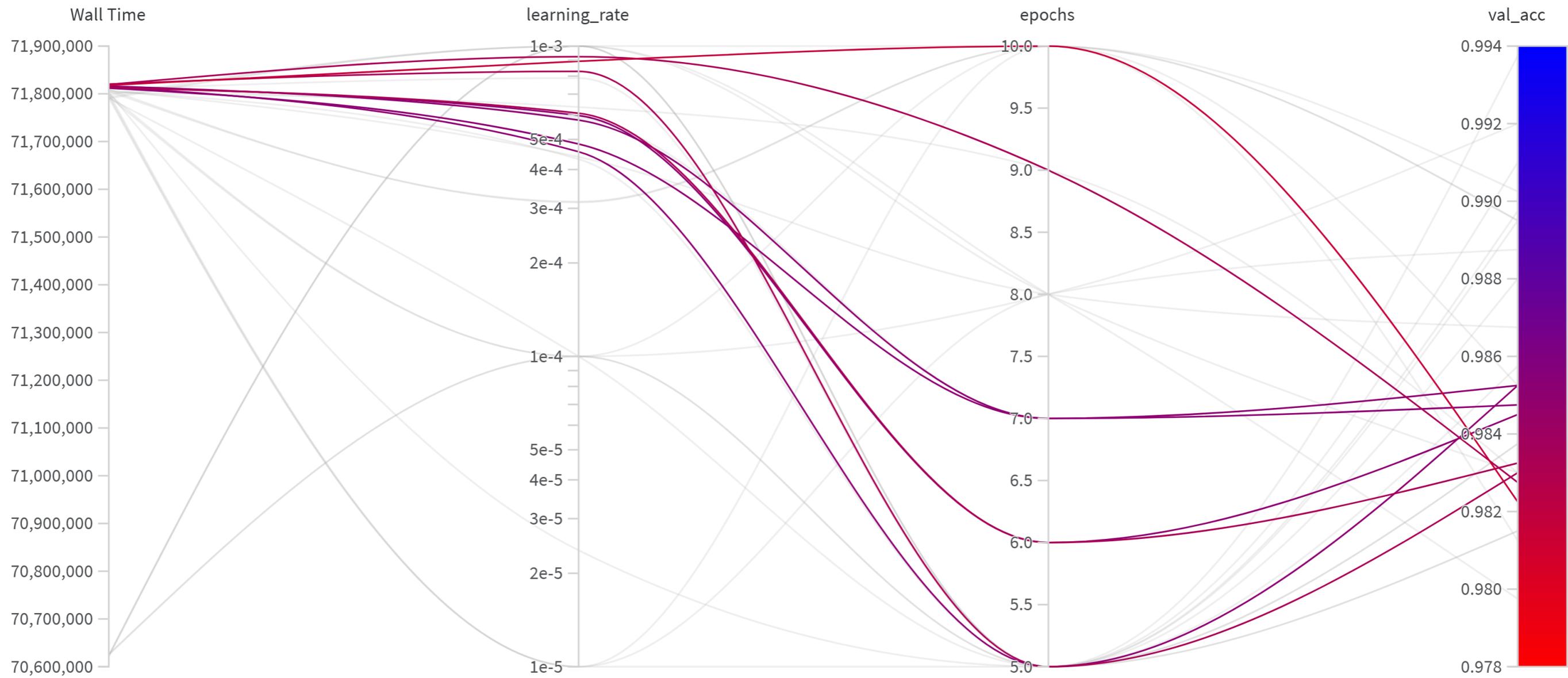
Production

Hyperparameter Tuning

# Hyperparameter tuning - Experiment tracking



# Example - Hyperparameter visualization



# **Let's practice!**

**FULLY AUTOMATED MLOPS**