We build a ML pipeline after we deploy

by Alyona Galyeva
for EuroPython 2021

Agenda

1. Introduction

2. The awesome solution:
   - ML pipeline out of the box in 1 line of code
   - add ML pipeline anytime even in prod env

Agenda

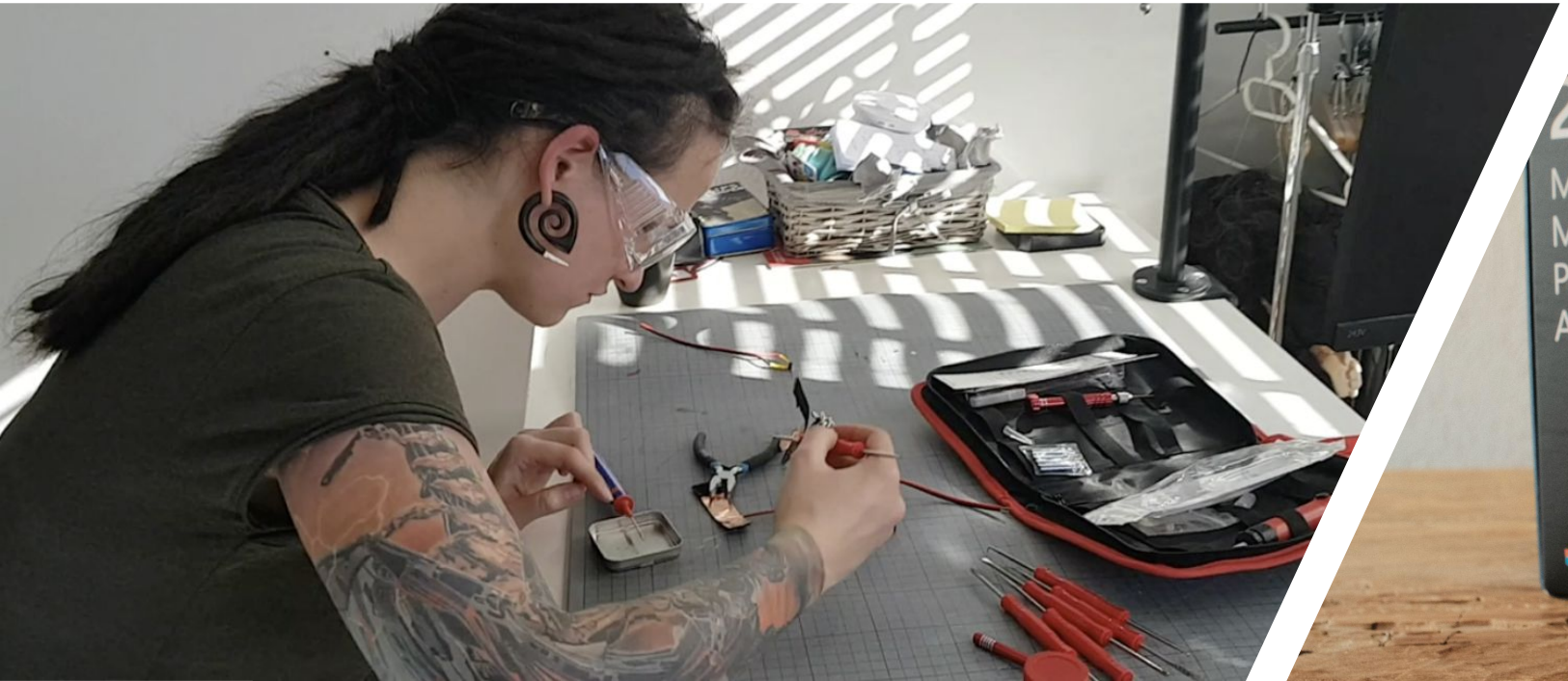
1. Introduction

2. ML Pipeline:
   - why and when
   - building blocks
   - engineering
   - debugging and monitoring
   - open-source Python libraries to save your time

Introduction


Jury
MAARTEN SUKEL
GAURAV AGARWAL
ALYONA GALYEVA
MICHIEL VAN BUTHOVEN
JUDIT SZAFFENAUER
WaiDATATHON
FOR SUSTAINABLE FUTURE
October 12, 2020
WOMEN IN AI
IrenaStollova
JoannaKurkowska
GabySoza
PrawalHR
[New User]
DaphneNdyemanya


WaiDATATHO
FOR SUSTAINABLE FU
CONGRATULATI
EVERY PART
YOU DID




2021
Microsoft
Most Valuable
Professional
Award
We recognize and value your exceptional contributions to technical communities worldwide.
MVP
Microsoft
Most Valuable Professional Award
2021
Alyona Galyeva
2021
MVP
Microsoft®
Most Valuable
Professional

ML Pipeline: why and when

Why?

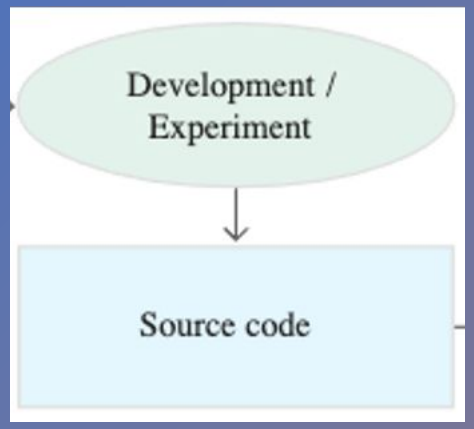
reduce the cost of data science projects:

- focus on new cases/models
- prevent bugs
- audit

When?

- go from PoC to MVP
- time to scale

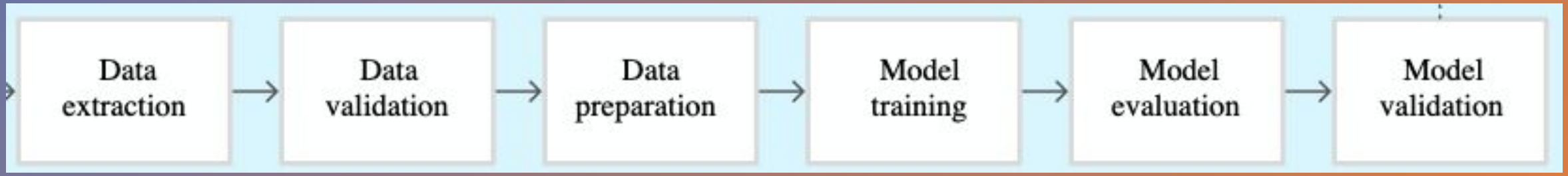
ML Pipeline: building blocks



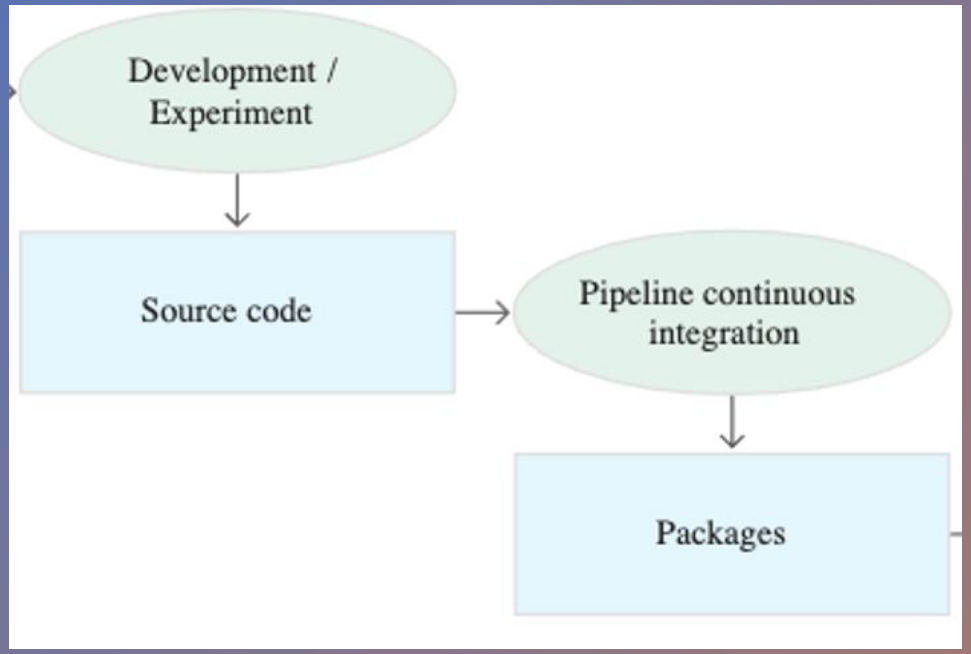
Development and experimentation

Try new ML algorithms with orchestrated experiment steps

Output: ML pipeline steps source code



https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

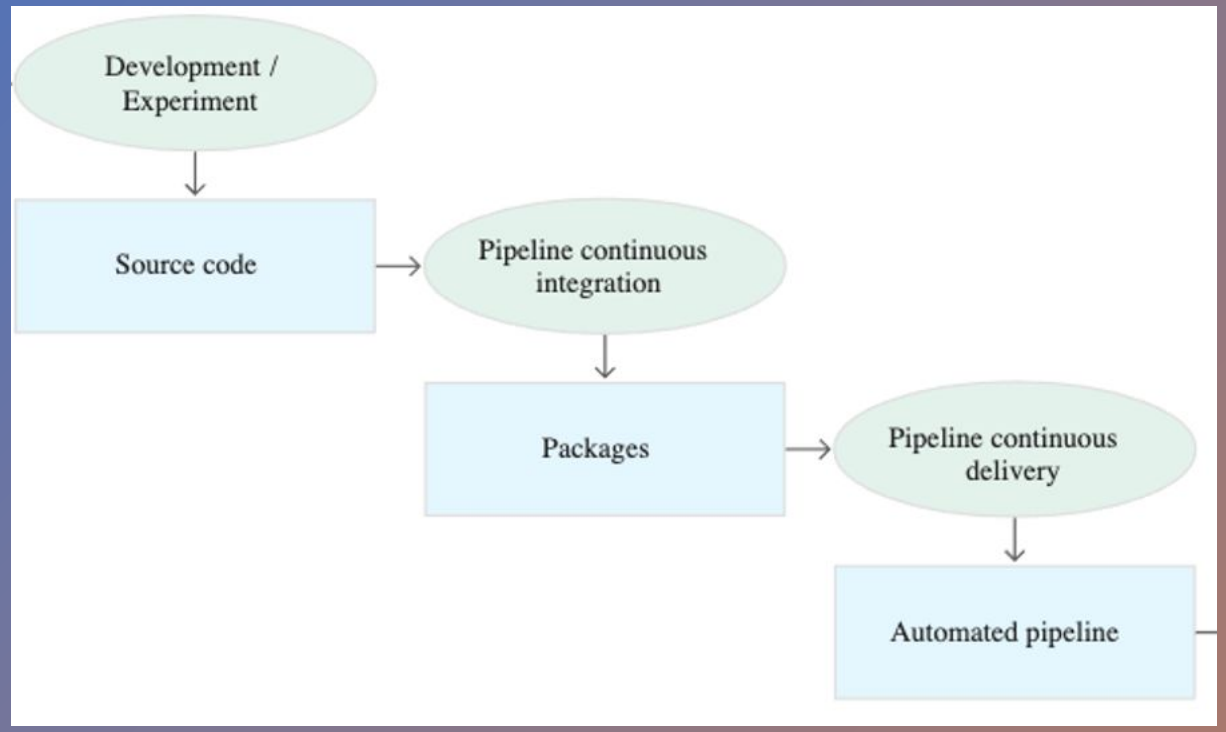


ML pipeline continuous integration

Build source code, run tests

Output: ML pipeline components to deploy

https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

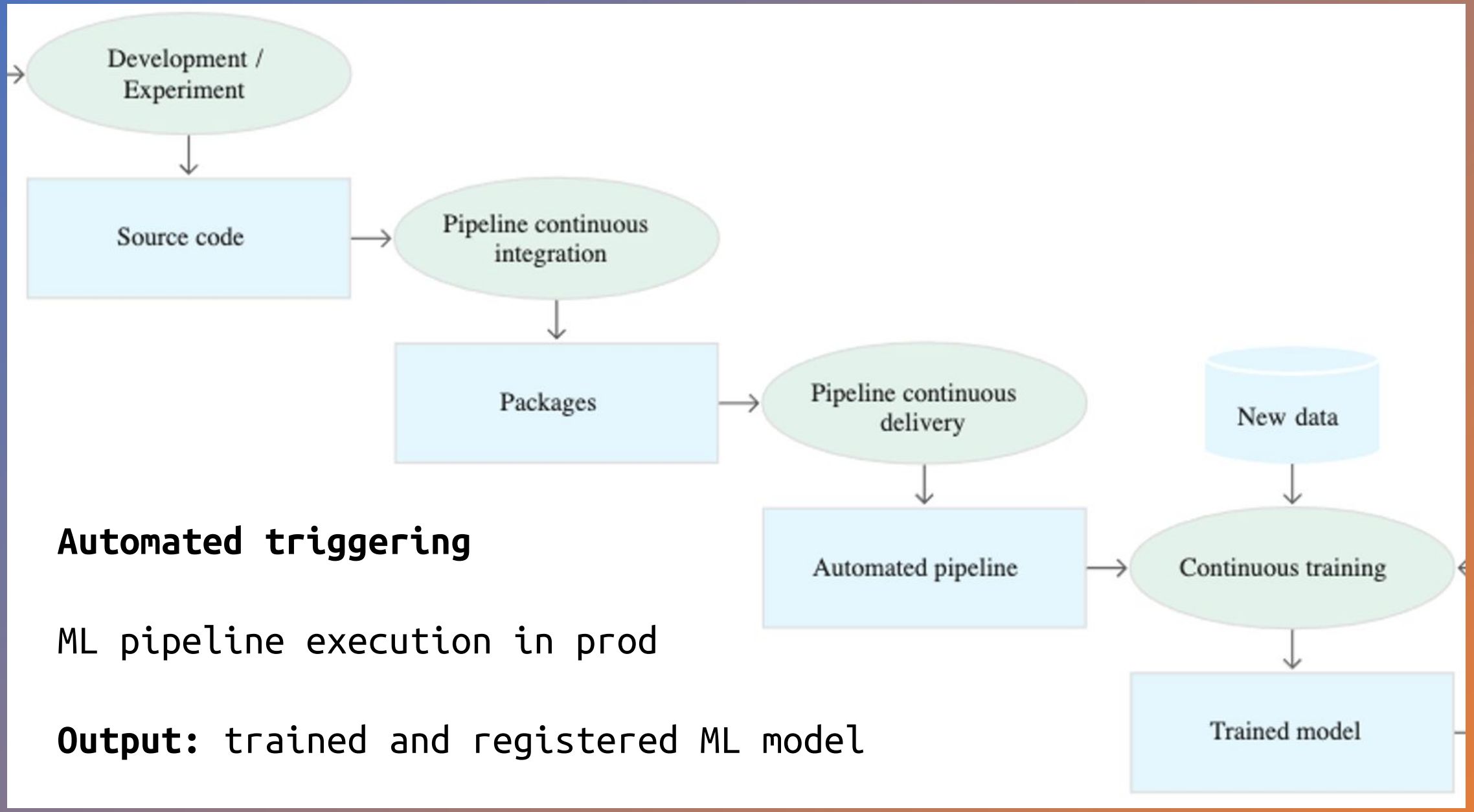


ML pipeline continuous delivery

Deploy artifacts to the target env

Output: deployed ML pipeline

https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

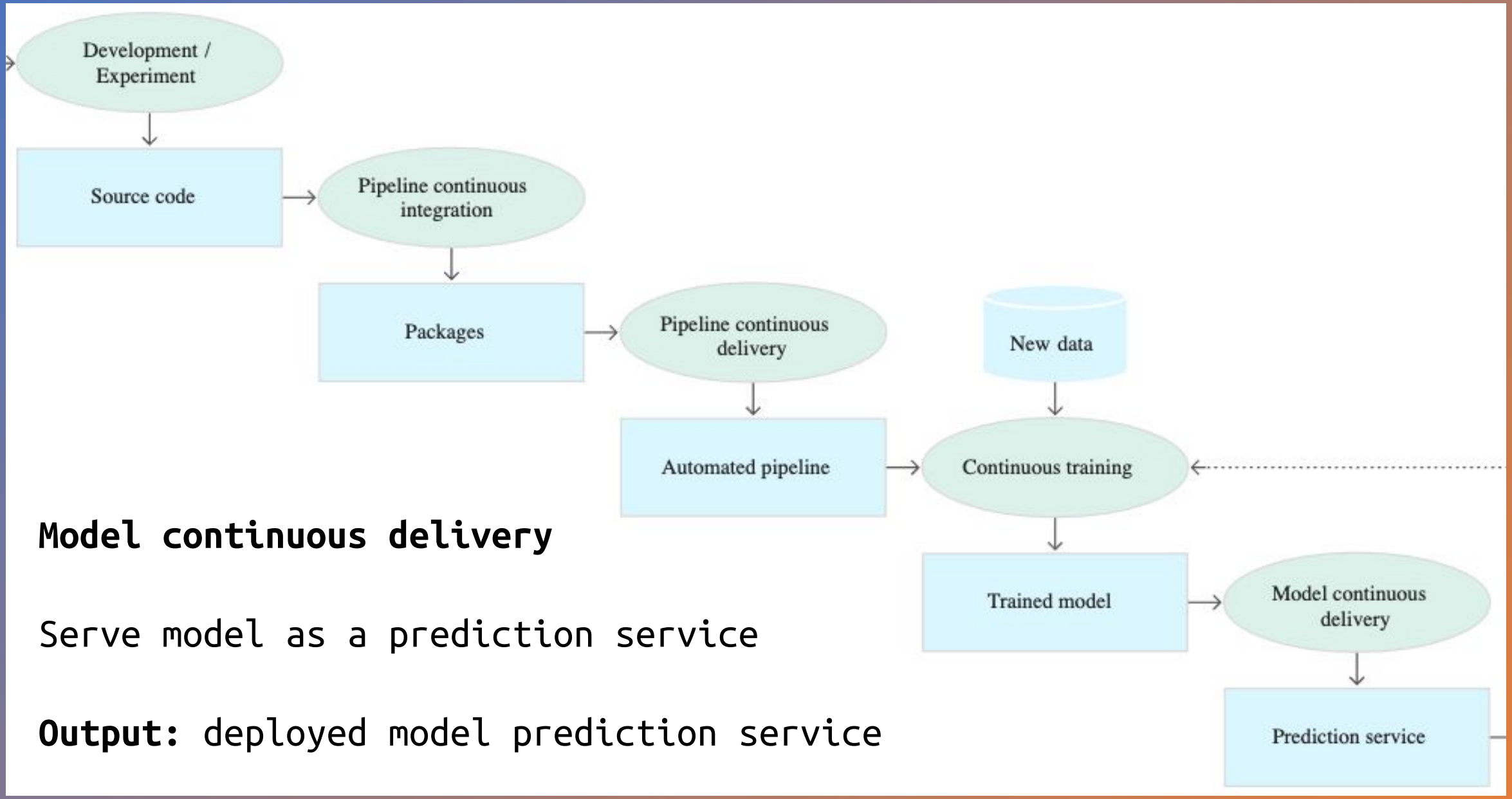


Automated triggering

ML pipeline execution in prod

Output: trained and registered ML model

https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

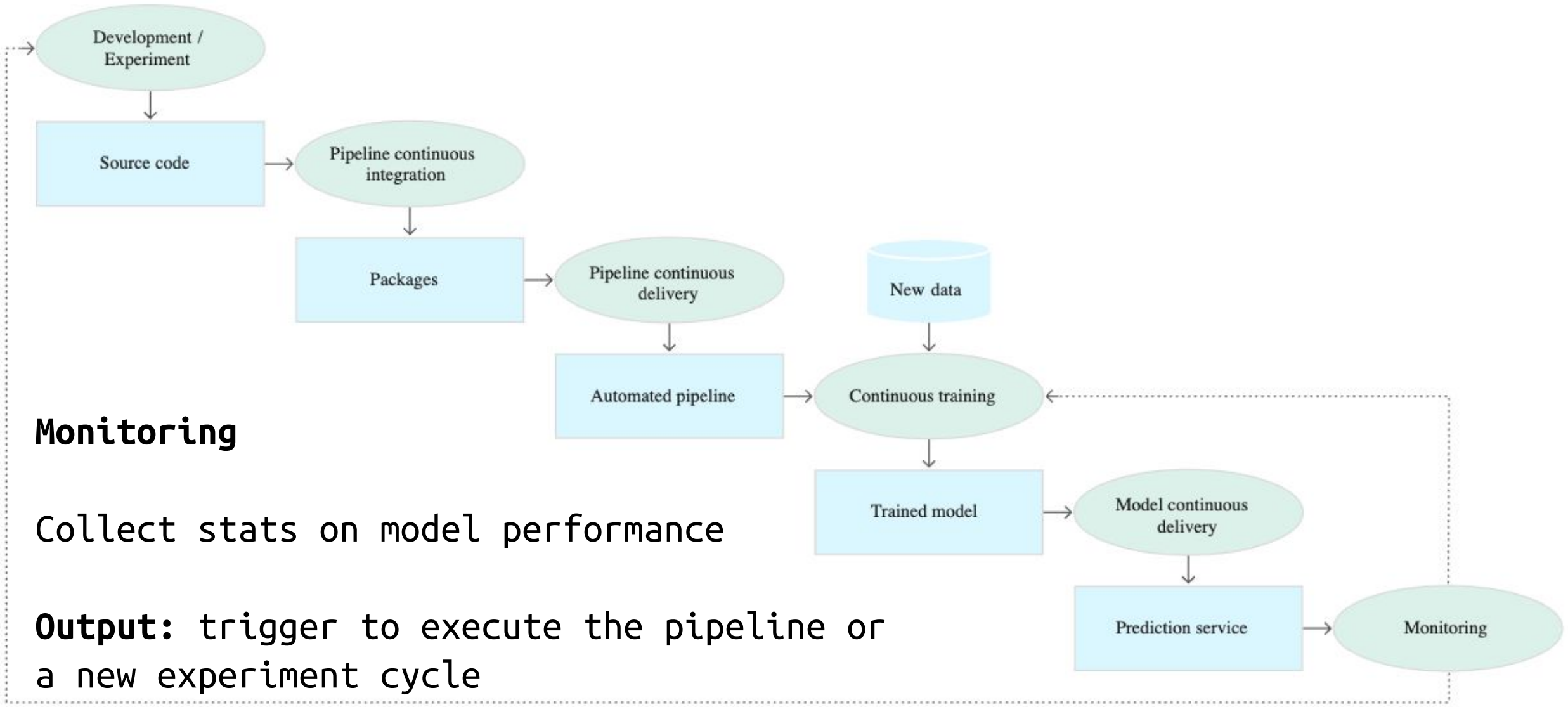


Model continuous delivery

Serve model as a prediction service

Output: deployed model prediction service

https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning



Monitoring

Collect stats on model performance

Output: trigger to execute the pipeline or a new experiment cycle

https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning

ML Pipeline: engineering

Engineer around failures

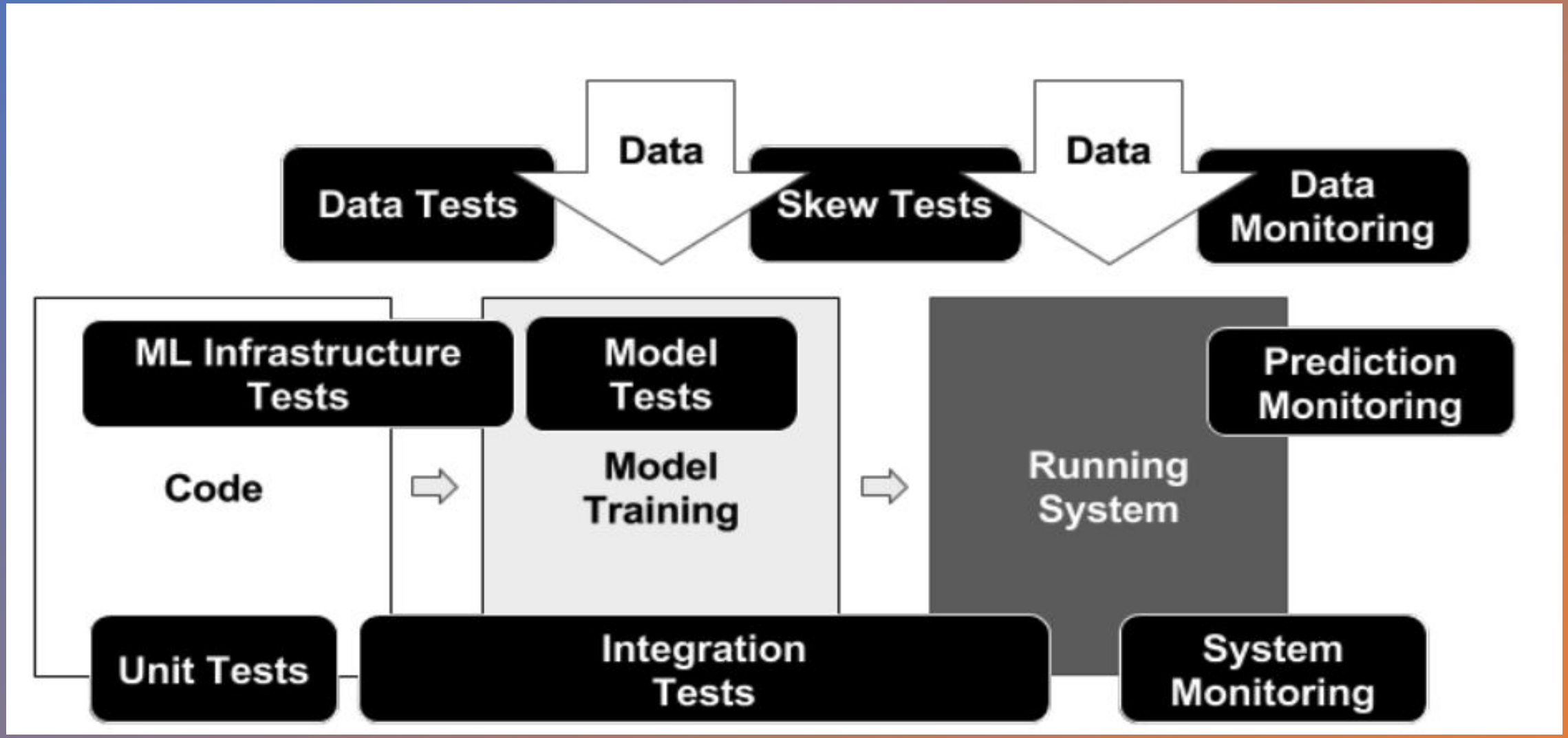
- input checks
- output checks
- model fallback

Engineer for performance

- scale
- caching
- feedback collection

ML Pipeline: debugging and monitoring

ML debugging and monitoring



Source: https://research.google/pubs/pub46555/

Top 3 debugging issues

- unpinned libraries
- data pre-processing
- scattered config for diff envs

Monitoring

- system monitoring
- data monitoring
- model monitoring

ML Pipeline:
Python libraries

Delta lake, DVC - data versioning
Airflow - data processing
Great Expectations - data validation
Feast - feature store
Hyperopt, Katib - tuning
Kubeflow, Pachyderm, TensorFlow Extended- platform
LIME, SHAP, Alibi Explain- model interpretability
Mlflow - lifecycle management
Seldon Core - model serving
FastAPI - restful APIs
Pytest, locust - testing
Evidently, Alibi Detect - debugging and monitoring

Questions?