Breaking the wall between DevOps and A.I.

Presented by Gabrielle Davelaar & Jordan Edwards

Next

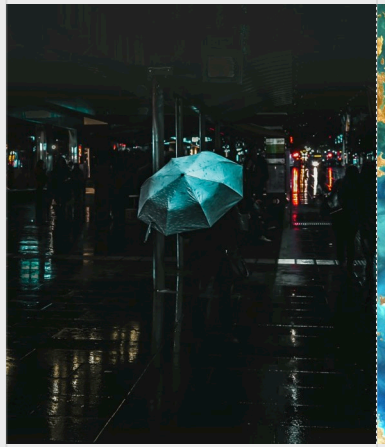
Today’s talk

After this talk you will know:

- What the pitfalls are on the horizon
- How to avoid them
- How DevOps can help

Who are we?

Introduction



01

Where it all starts

The valley

02

How to get out

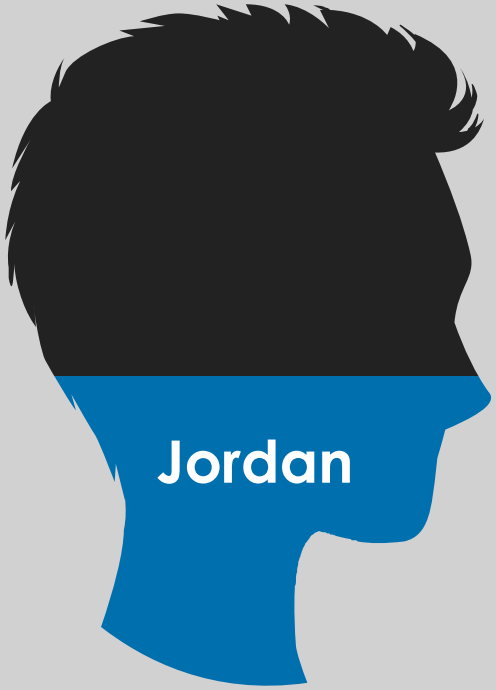
The wall

03

Finding Utopia

Demo

04



Engineer

DevOps

A.I.

Data solution architect

DevOps

A.I.

What does “DevOps for AI” mean?

DevOps is the standard way to manage application lifecycles through a pipeline of code, test, build, deploy (CI/CD cycle).

Infusing AI into this lifecycle brings new challenges and changes to the DevOps pipeline.

A **CI/CD solution for AI** requires supporting:

- **Reproducibility** of data → model
- **Validation** of model (does it meet quality bar, A/B comparison)
- **Storage, versioning** (track lineage and evolution of model over time)
- **Deployment, tracking, data collection** (across intelligent cloud + edge)

Breaking today's wall in DevOps & A.I.

Disappointment valley

The inability to move from a proof of concept to actual production with no clue what is actually happening

From **art** to **science**

Democratize A.I.: Transparent, agile built for long term success

Hybrid DevOps

Not reinventing the wheel all over again just adjusting

Bridging the valley.

"By far the greatest danger of Artificial Intelligence is that people conclude too early that they understand it." -Eliezer Yudkowsky

TRENDS of DESTRUCTION

Letting you fall even deeper into the valley of disappointment

SUBOPTIMAL KNOWLEDGE

knowing what can be solved with A.I. and what not, how long at takes with what resources

GARBAGE IN GARBAGE OUT

Digital audit trail: The ability to replicate an outcome no matter the timeframe and who used it

EVIL BLACKBOX

Avoiding algorithmic bias, understanding how algorithms make decisions based on our own (un)intentional bias

I'VE ALWAYS DONE IT THIS WAY

Academia & the corporate sector are completely different, so are the data scientists in both industries

Suboptimal knowledge ≠ Project transparency

Project transparency: *Knowing how long a project will take, what costs are involved and how much resources are necessary*

a

Overview

The ability to have early insights when a project is not going to work out

b

Resources

The ability to know if more resources are necessary to reach a deadline

c

Performance

The ability to have a good estimation on the costs per project

The A to Z story in one story

Digital audit trail

Garbage in garbage out

Tracing back the steps

Versioning

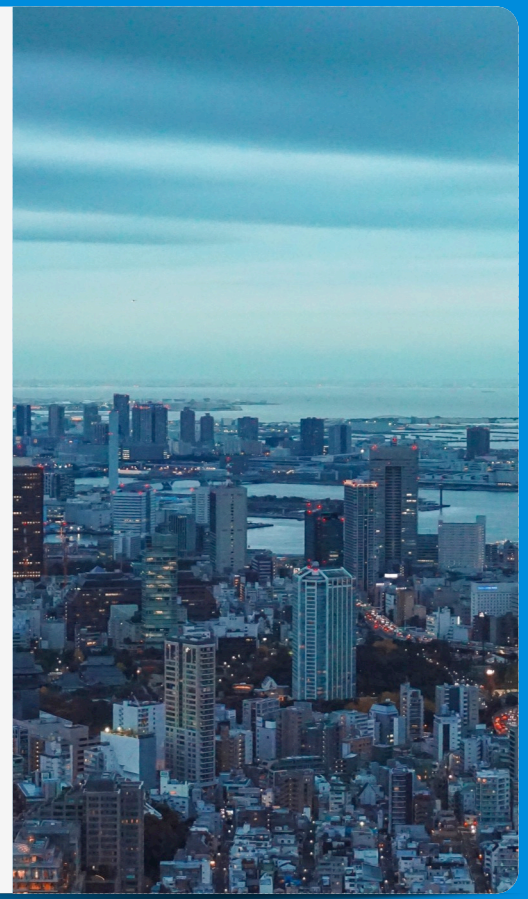
Knowing which dataset was used with which model

Initial modeling

Predefining and setting the variables

Security

Knowing how the data came in, who worked on it and when

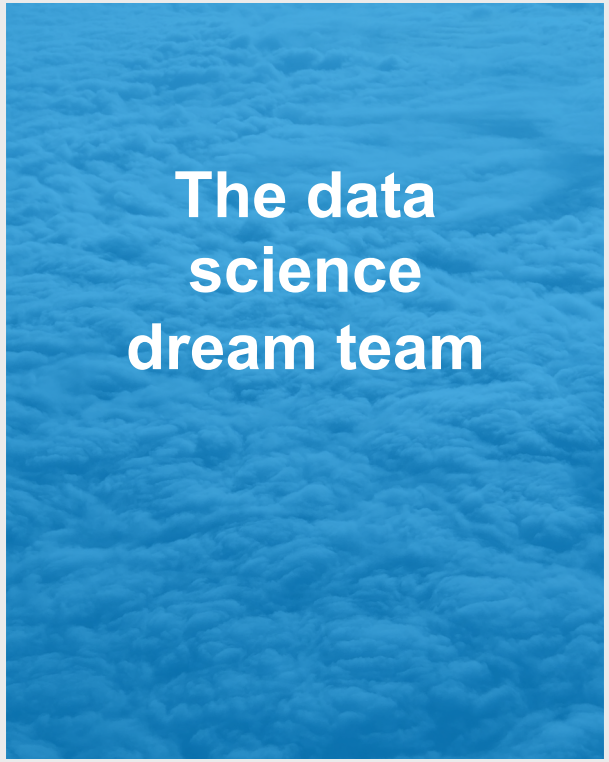


Building transparent models

Reducing algorithmic bias through application & model monitoring together with algorithms that are capable of self explaining features

Data science unicorns

The majority of the average data scientist has been trained as a mathematician or a statistician not a developer. Engaging an engineer early in the process together with automating will create strong processes ready for production



Integration into a simple,
easily-repeatable everyday development task

CI/CD

Data prep
Feature engineering logic modular
Unit testing
Dataset invariants

Automated testing

Interactive data exploriation

Data quality assessment

Same configurations no matter the infrastructure

Configuration – as - code

A.I. lifecycle focus areas

Model control

Processes and procedures to make models reproducible (from source control to data retention policies)

Model validation

Unit testing, functional testing and performance testing - both in isolation and when embedded in an application

Model versioning

Provide a consistent way to store & share models

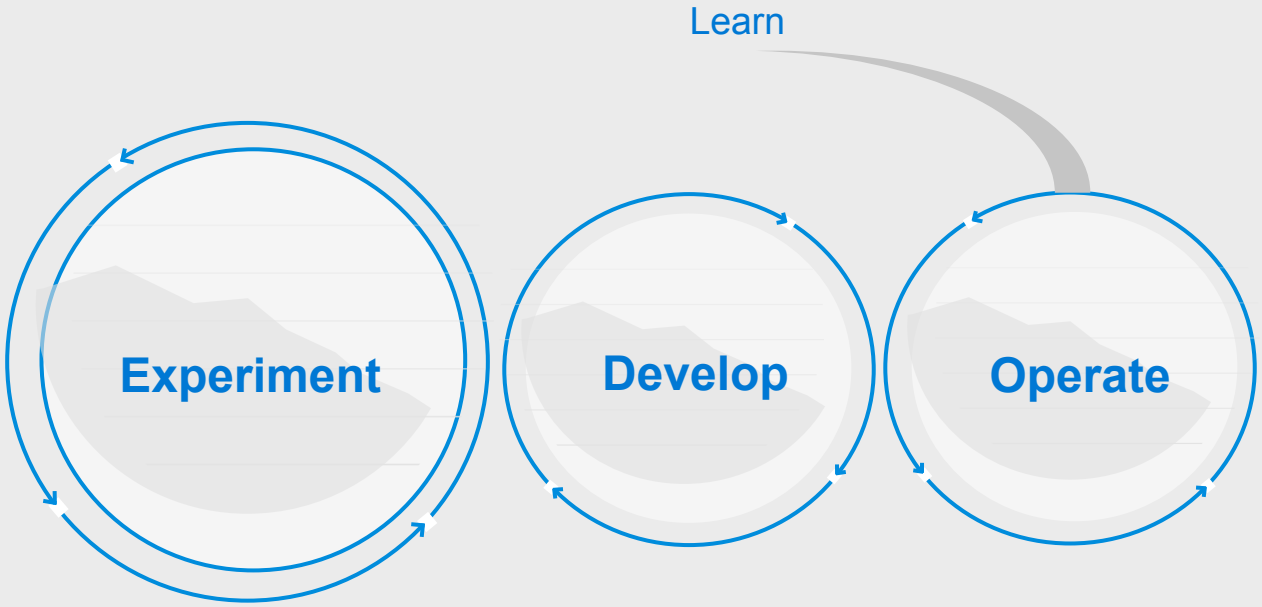
Model storage

A way to track where models are embedded / running

Model deployment

Describing an efficient process to get a model build into an application or service and leveraged to light up an end-user scenario.

Breaking the wall between DevOps and AI



Prove Feasibility

Data Acquisition

Business Understanding

Data Exploration

Initial Modeling

Build

Data Ingestion

Modeling

Deployment

Operate

Monitoring

Compliance (GDPR)

Governance

Security

Previous
Next

The journey
MODEL LIFECYCLE

Let’s put the theory into practice

Previous
Next

The journey

Three Steps in building AI models.

Prepare

Prepare Data

Experiment

Build model (your favorite IDE)

Train & Test Model

Deploy

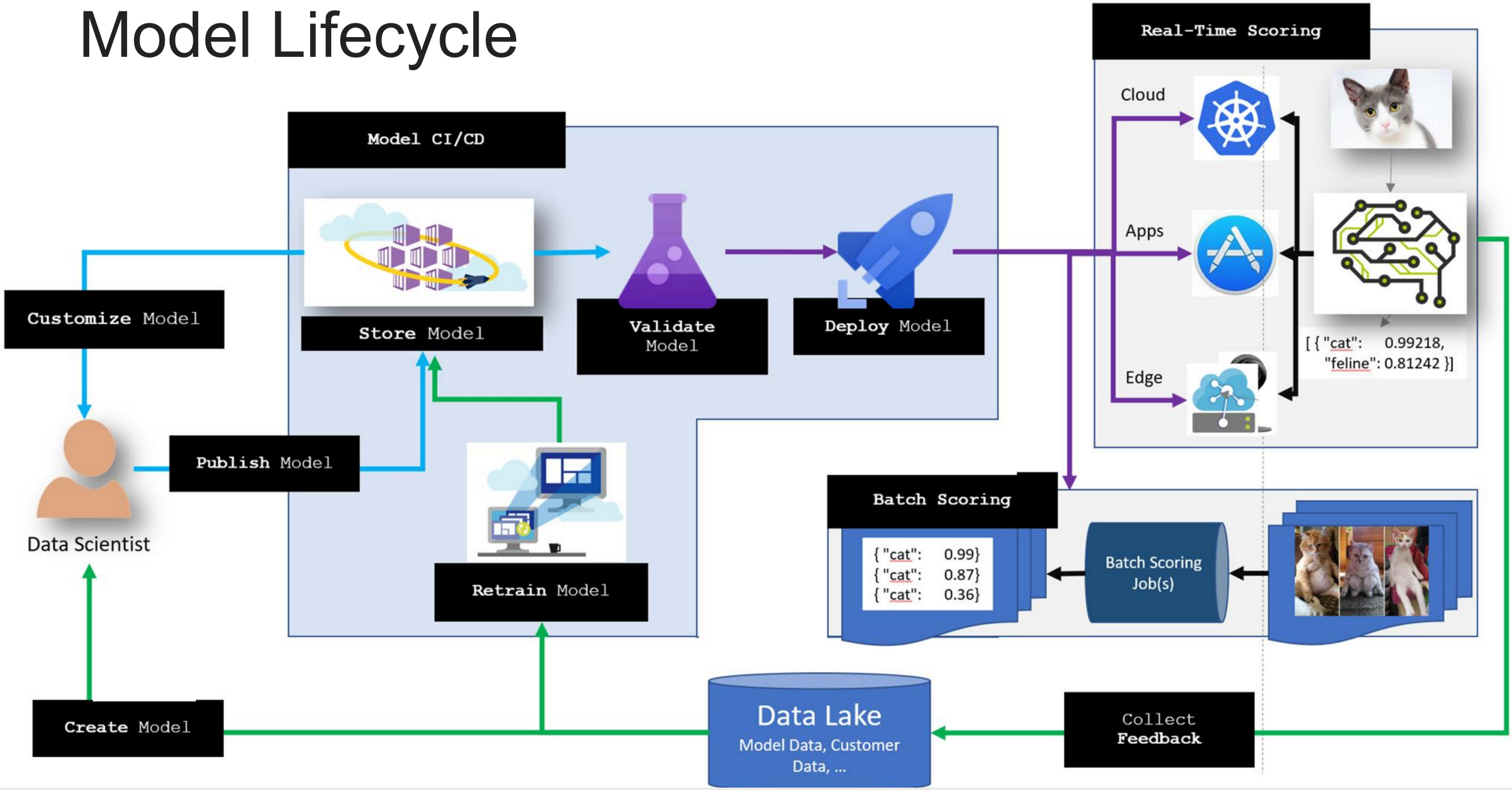
Register and Manage Model

Build Image

Deploy Service Monitor Model

Previous
Next

Model Lifecycle

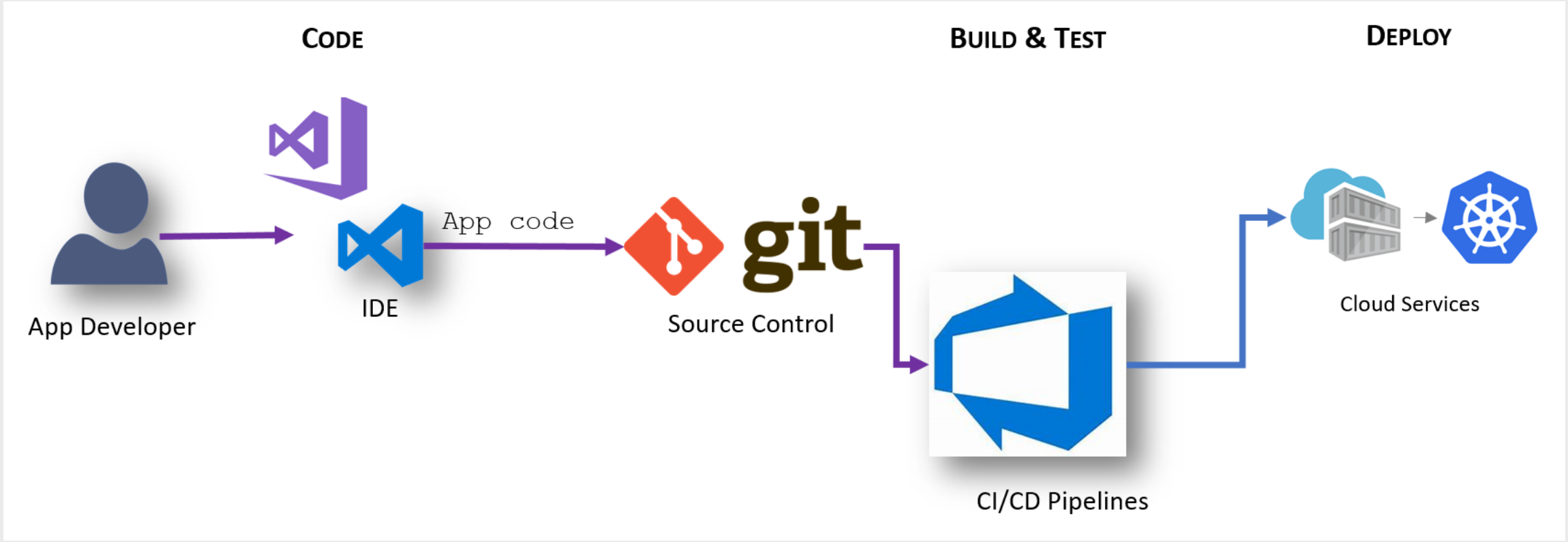


DevOps A.I.

Breaking the wall

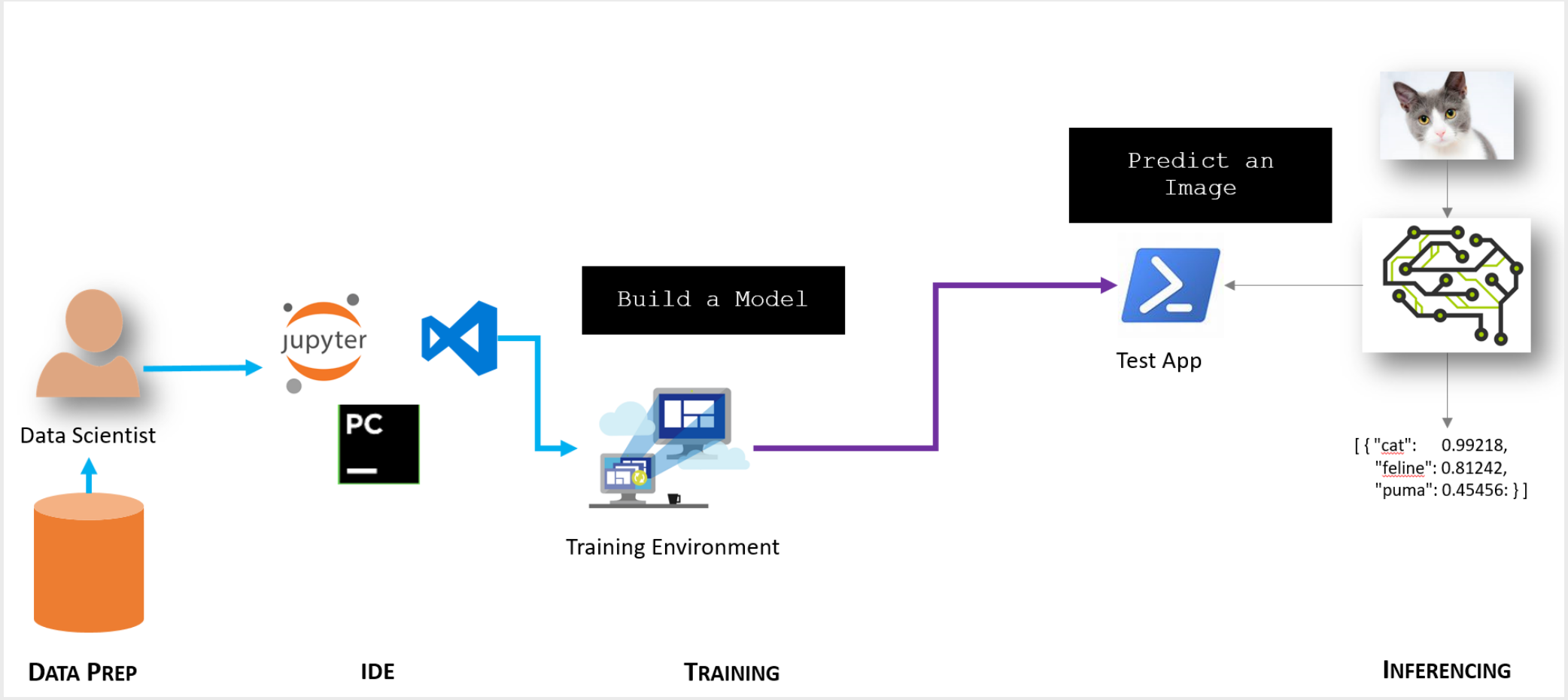
Lifecycle convergence

Basic Workflow: App Developer



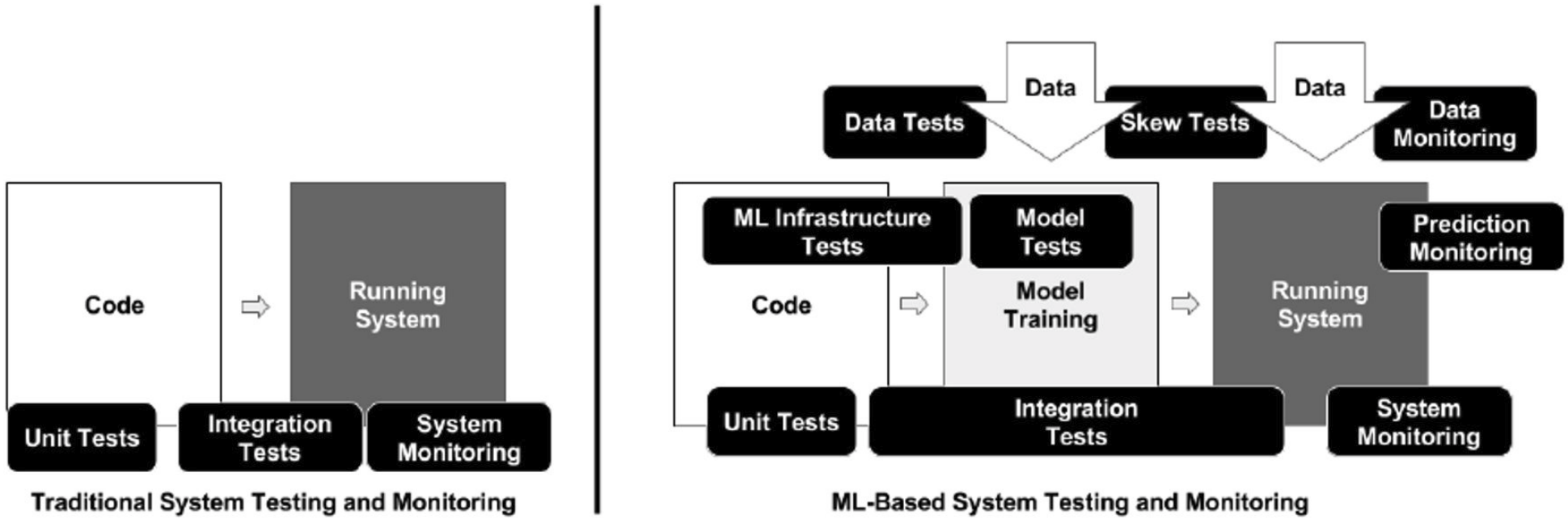
Previous
Next

Basic Workflow: Data scientist



Previous
Next

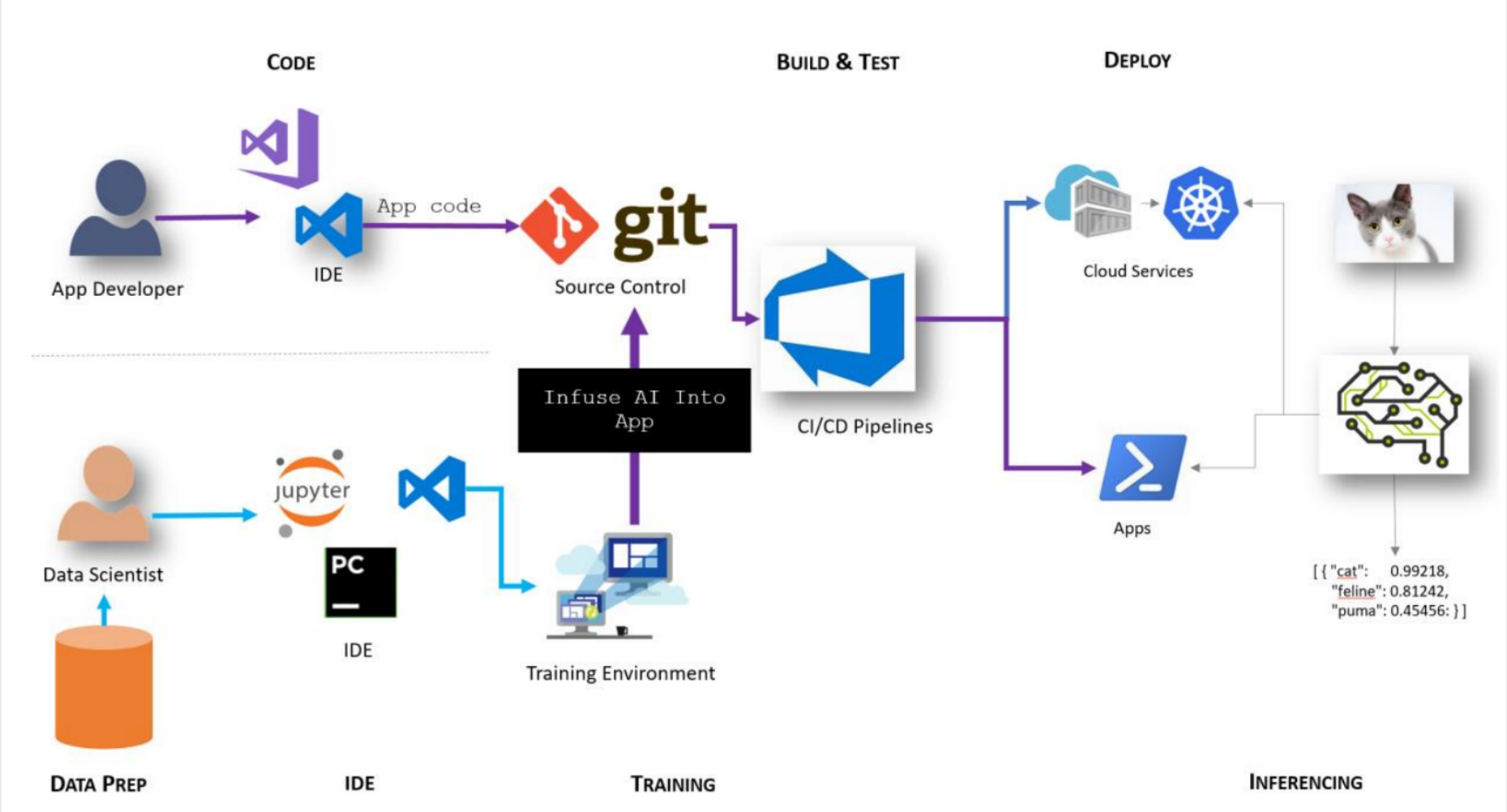
Traditional vs. A.I. Applications



Source: Google AI Paper "What's your ML test score? A rubric for ML production systems'

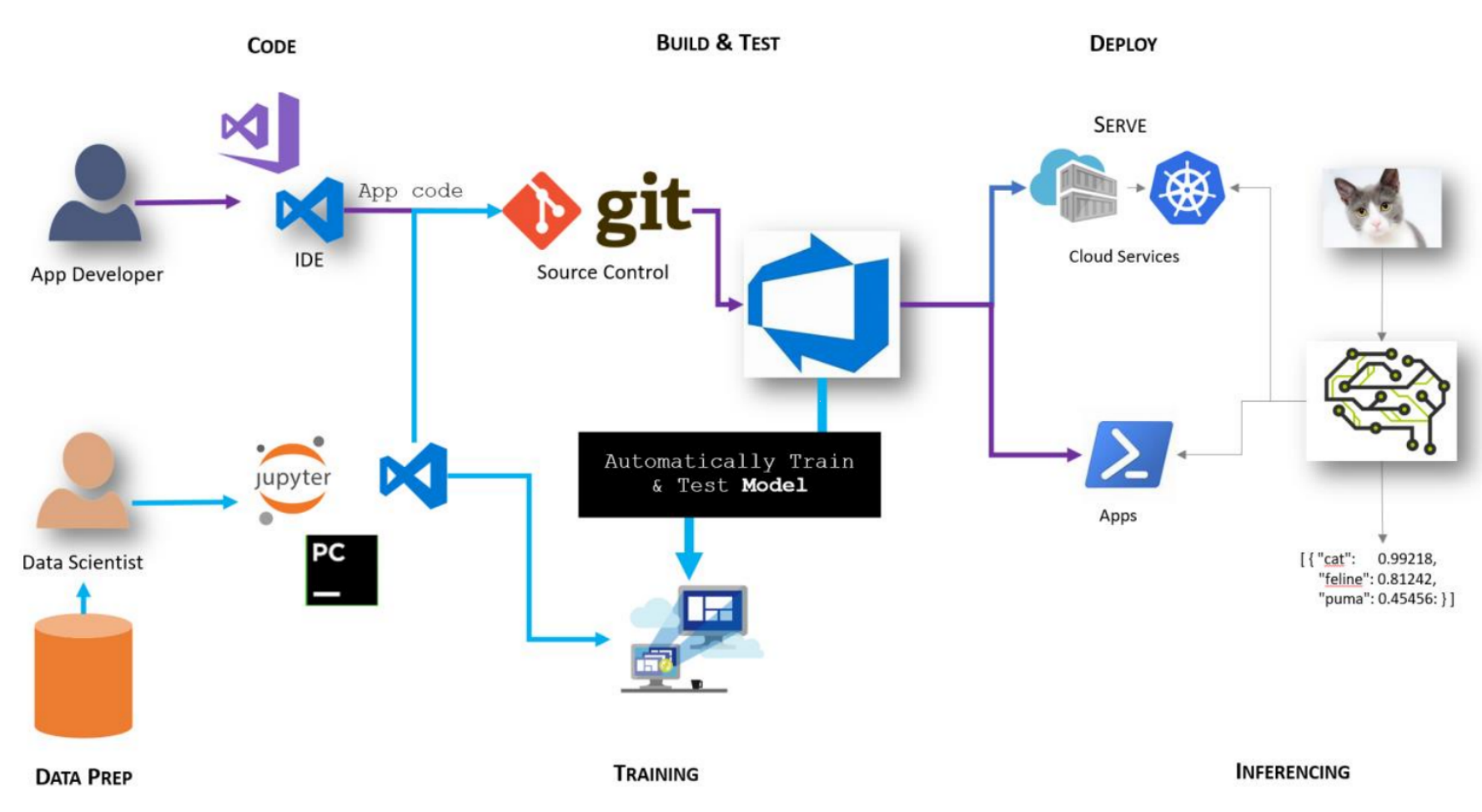
Previous
Next

Step 1 – Infuse A.I. into app



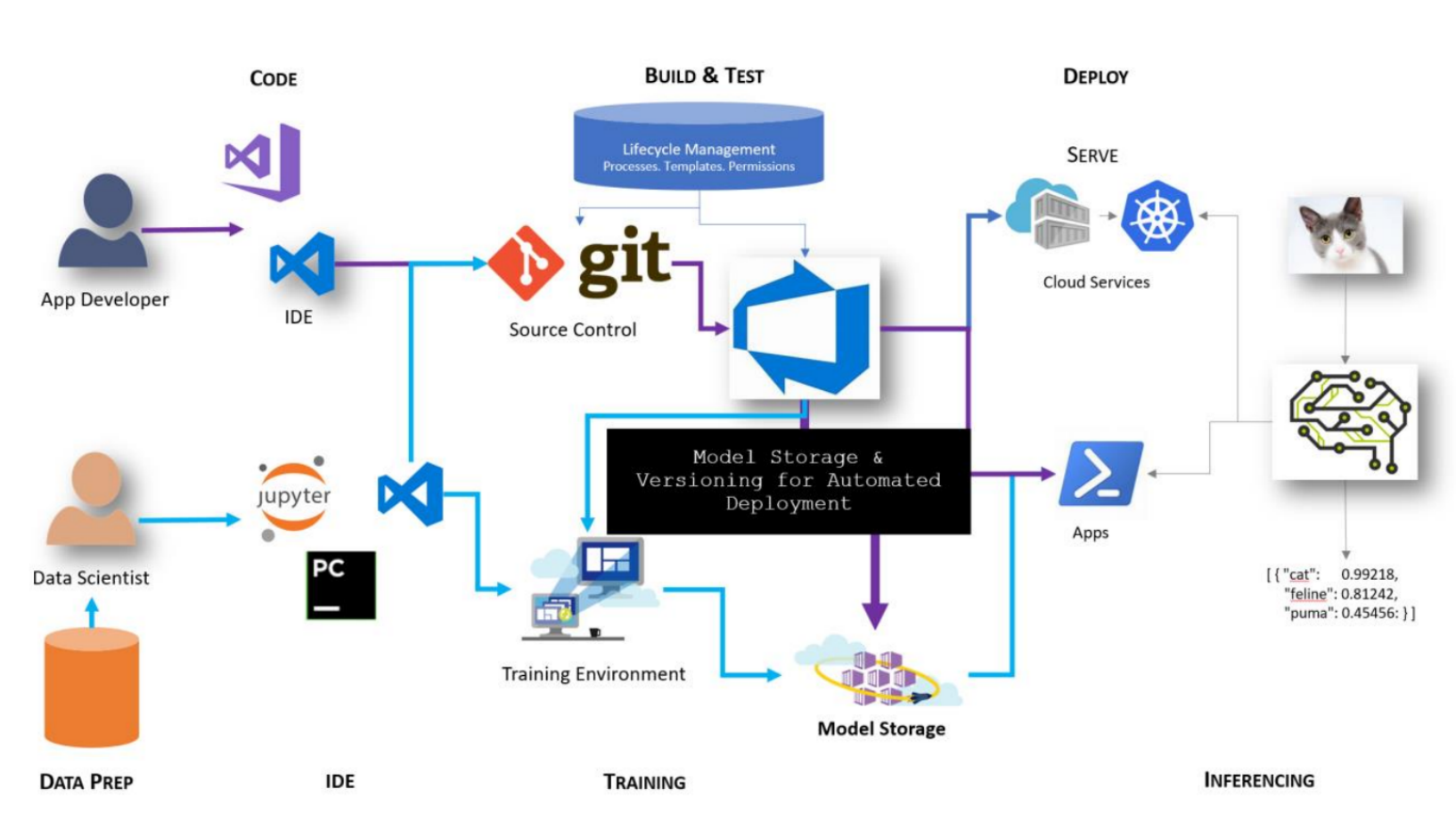
Previous
Next

Step 2 – Automate model training



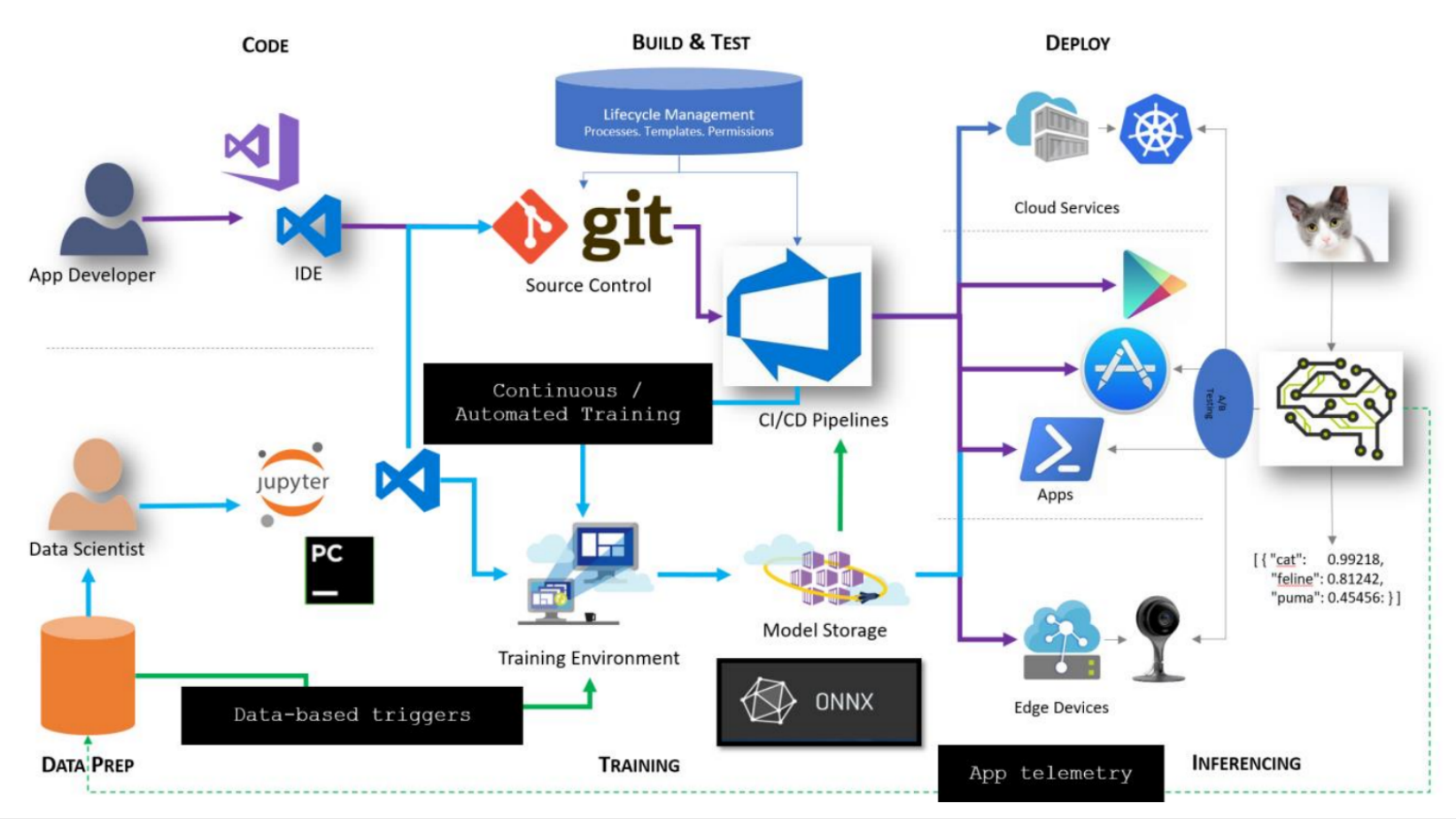
Previous
Next

Step 3 – Store, version and validate



Previous
Next

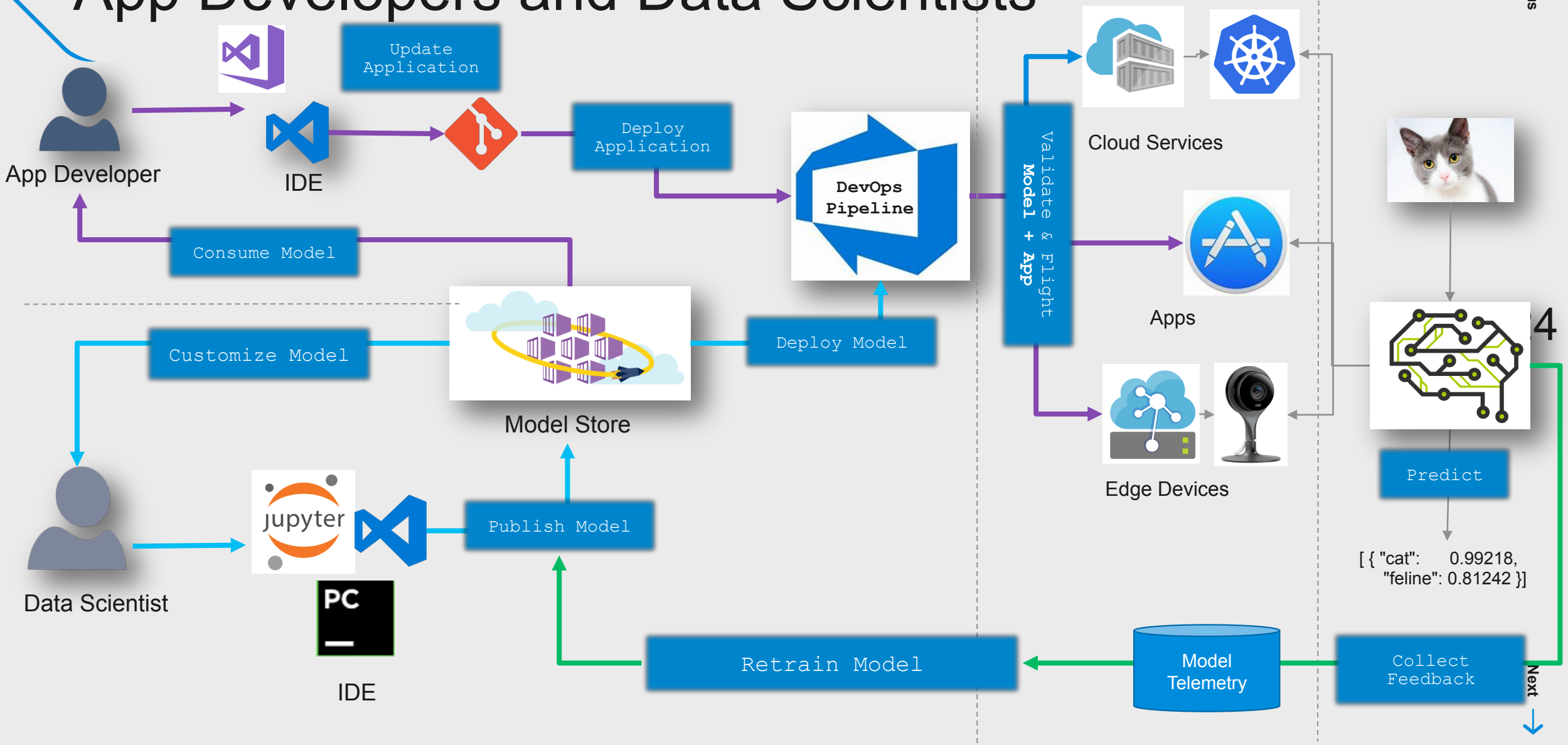
Step 4 – Automate model release



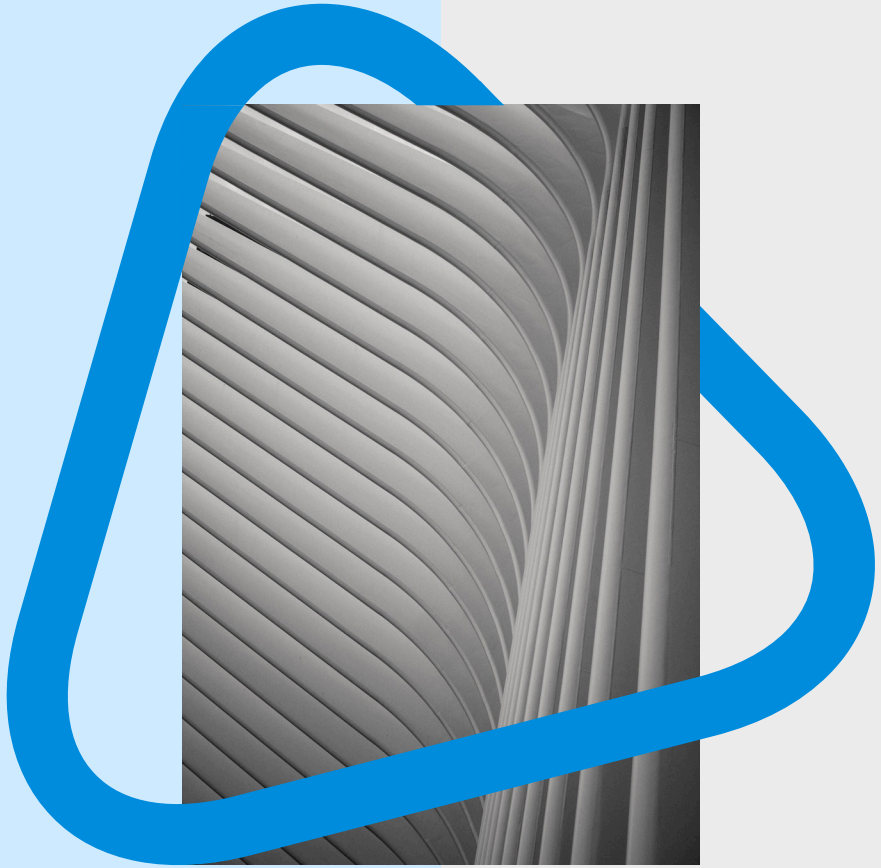
Previous
Next

The journey

App Developers and Data Scientists



Pain points for data scientists & DevOps



1. Software wise:

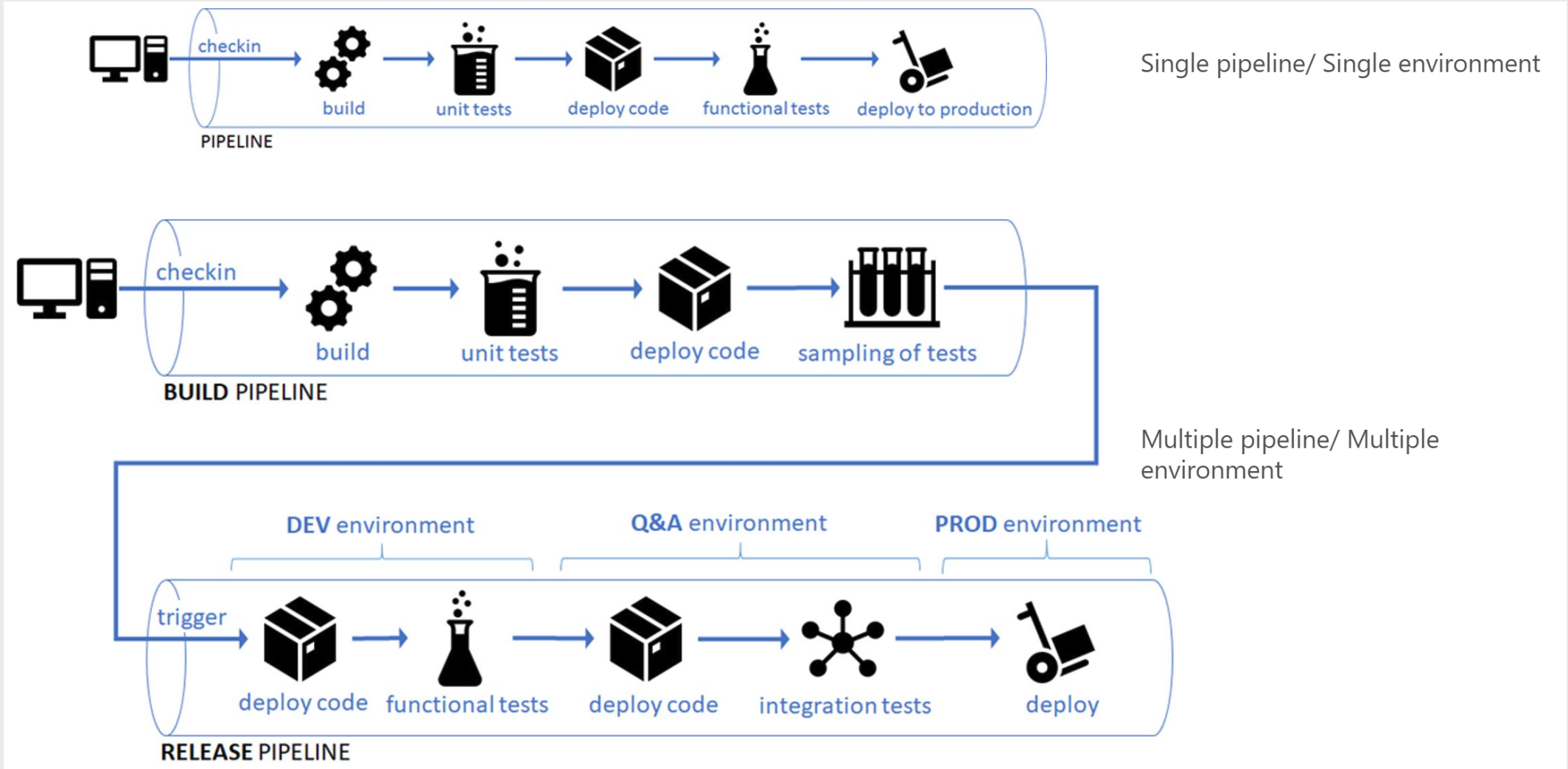
- ML stack might be different from rest of the application stack
- Glue coding
- Need to re-write featurizing and scoring code multiple times (in different languages)
- Hard to track breaking changes

02. Modelwise:

- Testing accuracy of ML model
- ML code is not always version controlled
- Hard to reproduce models
- Integrating model into application can take weeks
- Want to start using customer data to build models

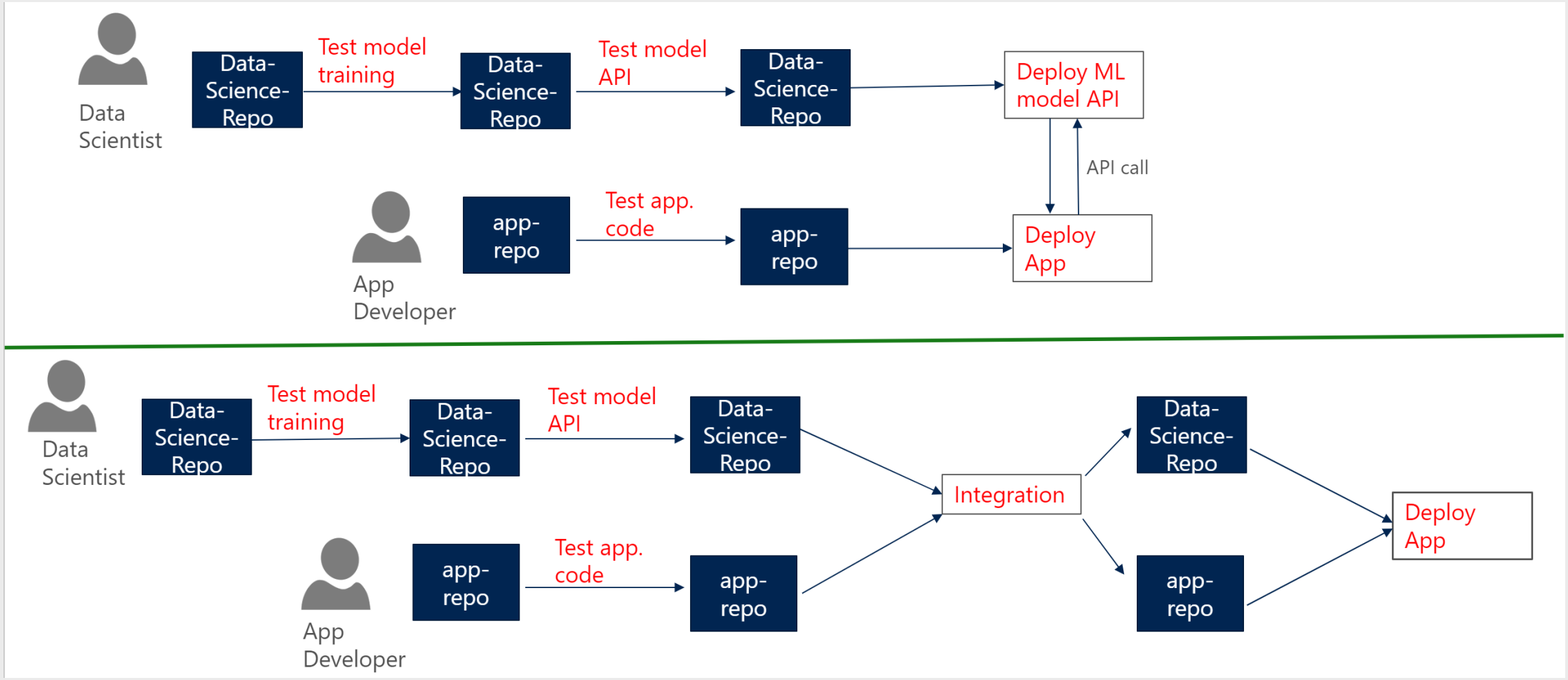
Previous
Next

Traditional application



Previous
Next

A.I. application



Previous
Next

DevOps A.I.

Proposed process

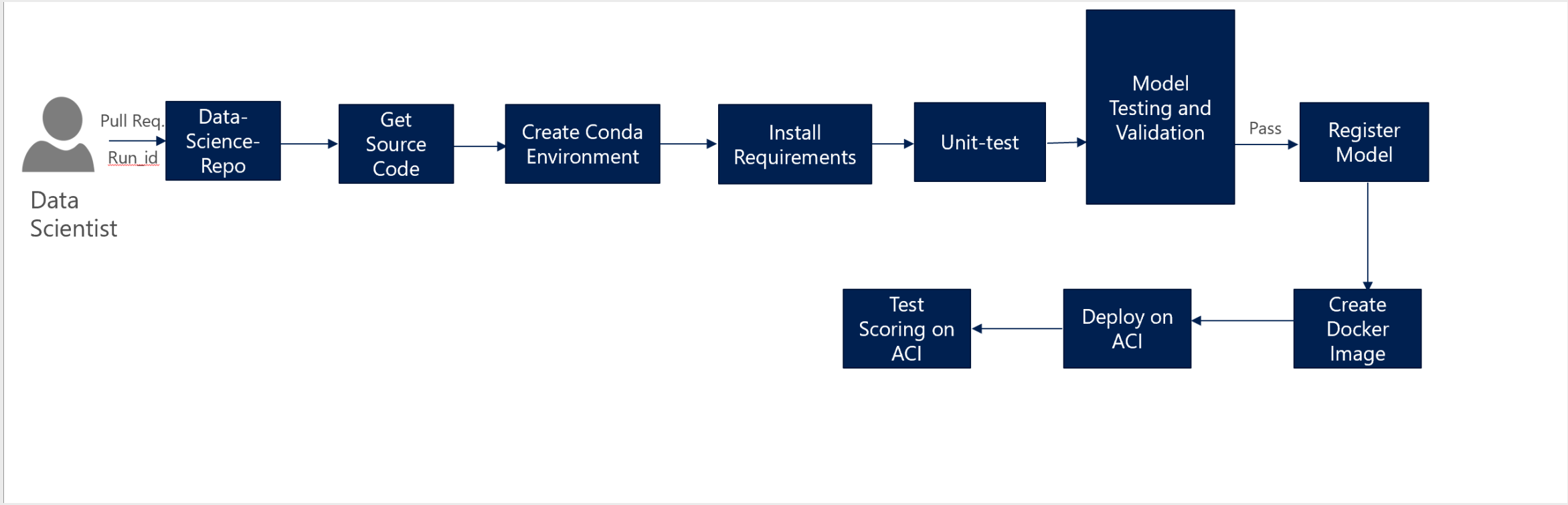
CI/CD for models

Previous
Next

Feature branch: triggered on each commit

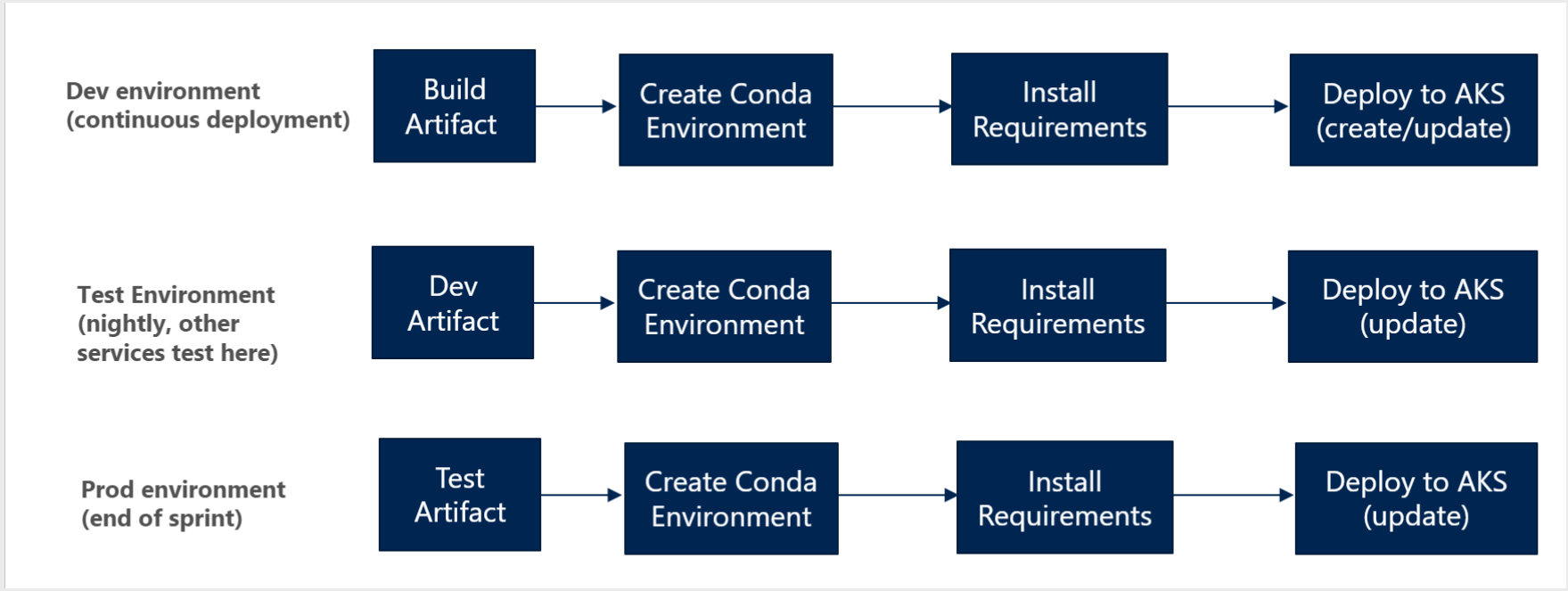
Previous
Next

Master branch: triggered on each pull request



Previous
Next

Dev / test / prod

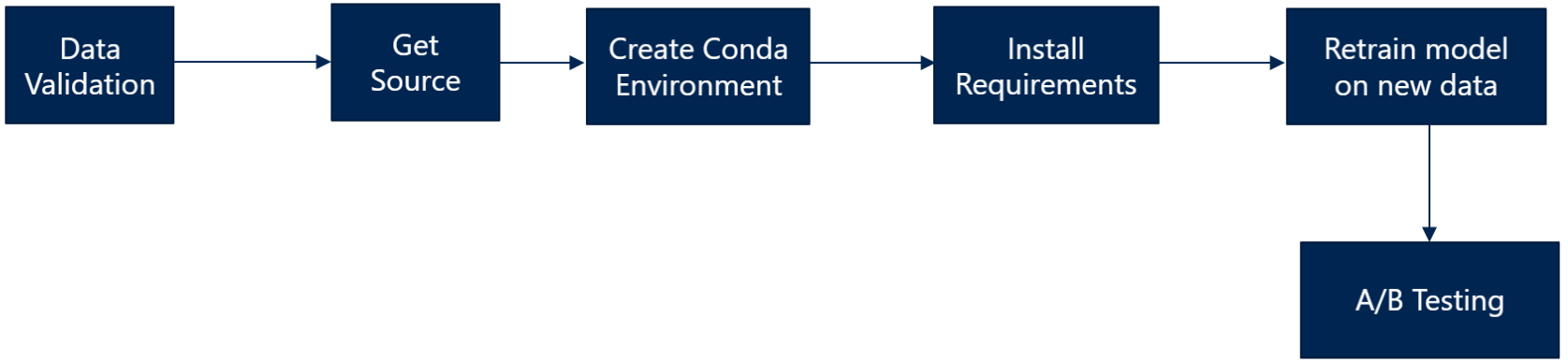


Previous
Next

Other pipelines

ONNX, CoreML, OneML
(end of sprint, everytime there is a new model)

Retraining Pipeline
(every night, or triggered on new data uploading to blob)
It will run in a pre-prod environment so it has access to production data, and wouldn't be promoted unless it passes A/B tests against prod data.



Previous
Next

Demos

- **Automatic retraining and deployment** to mobile (from Cognitive Services)
  - https://www.youtube.com/watch?v=ReG2QCbd-cE
- CI/CD pipeline for models in Azure DevOps
  - https://aidemos.visualstudio.com/DevOps%20for%20AI%20-%20Demo/_build