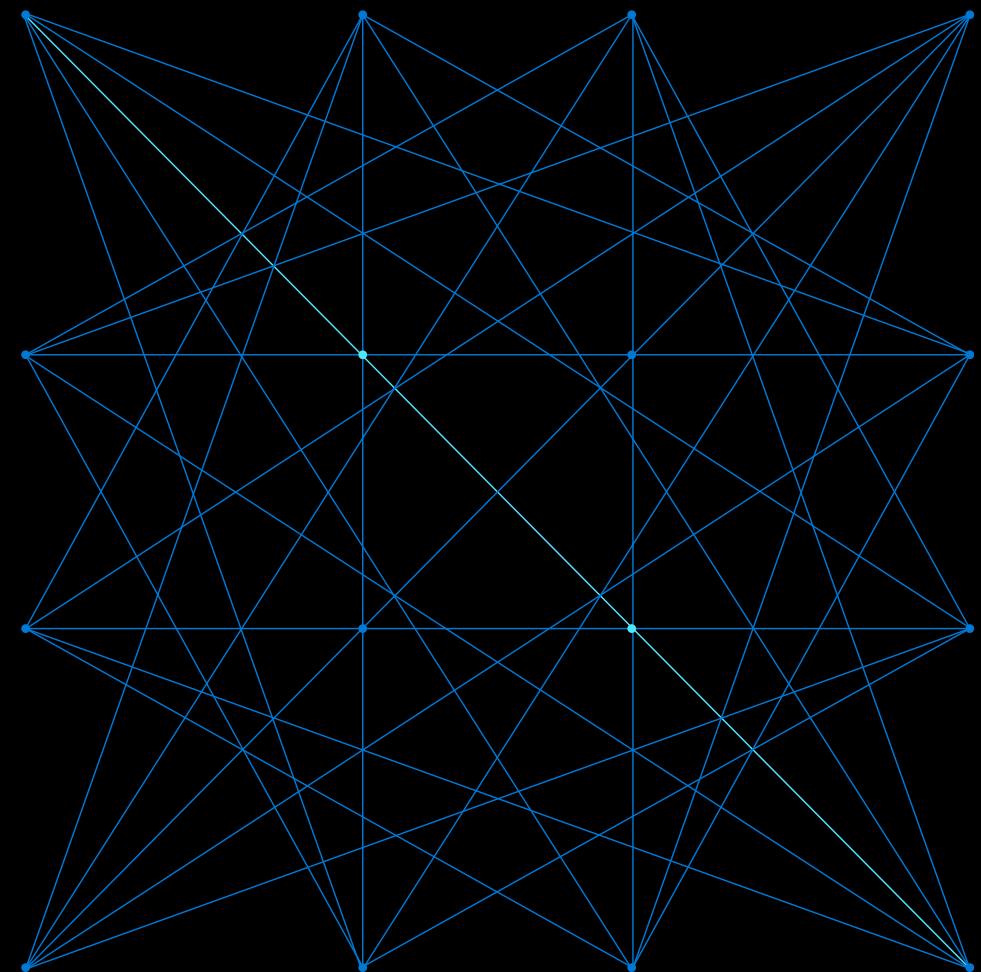


Azure webinar series

# Get Time to Value with MLOps Best Practices



# Welcome

## How do I ask a question?

If you have a technical or content-related question, please use the Q&A window

We will address the questions as they come in

## Can I view this presentation after the webinar?

Yes, this presentation is being recorded

A link to the recorded presentation will be sent to the email address you used to register

# Meet our speaker



**Jordan Edwards**

Principal PM Manager

# Buzzword Bingo!

MLOps

DevOps for ML

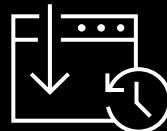
"Software Engineering 2.0"

GitOps for ML

Git for Data

# MLOps == How to bring ML to production

Bring together **people**, **process**, and **platform** to automate ML-infused software delivery & provide continuous value to our users.



## People

- Blend together the work of individual engineers in a repository.
- Each time you commit, your work is automatically built and tested, and bugs are detected faster.
- Code, data, models and training pipelines are shared to accelerate innovation.

101010  
010101  
101010

## Process

- Provide templates to bootstrap your infrastructure and model development environment, expressed as code.
- Automate the entire process from code commit to production.

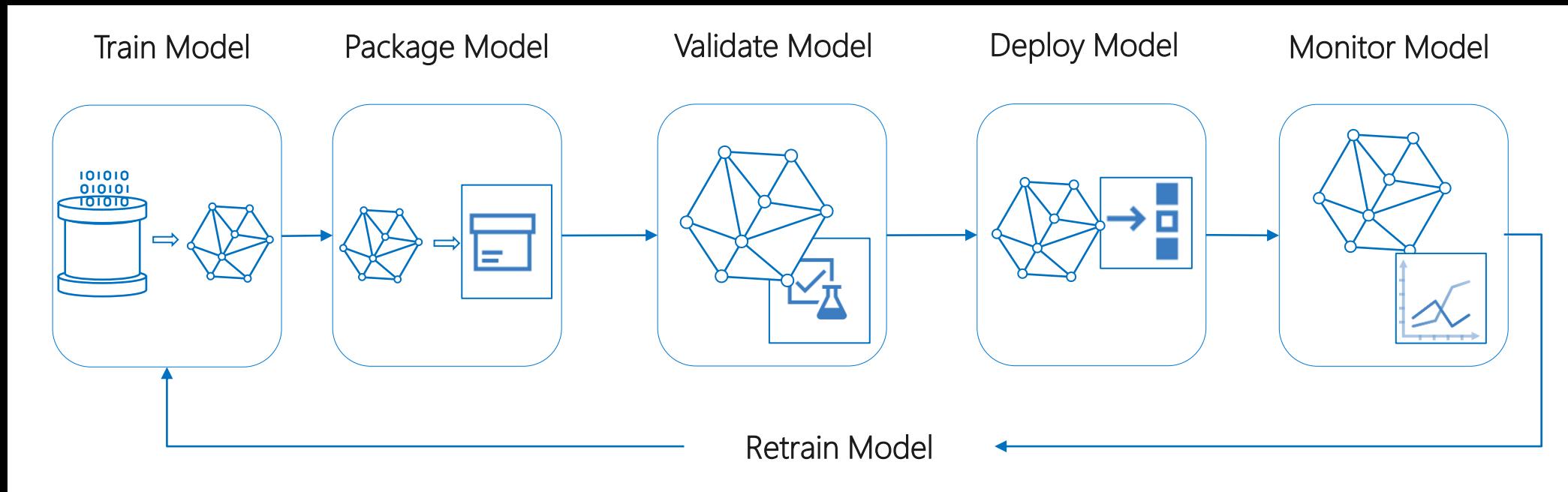


## Platform

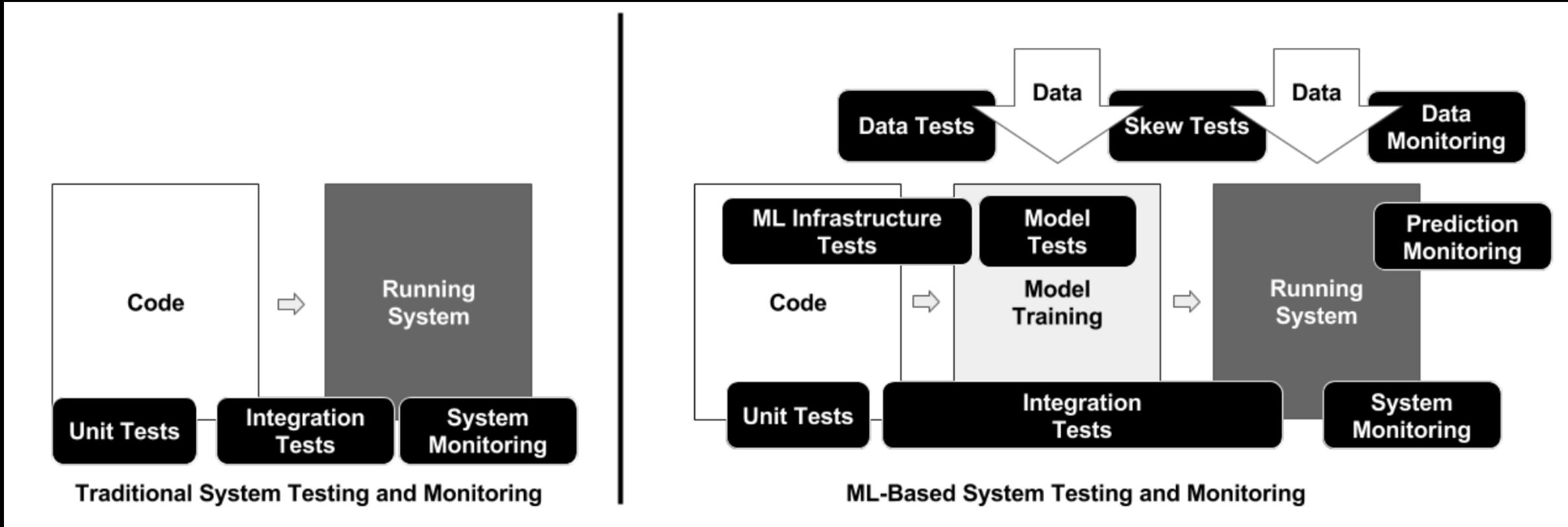
- Safely deliver features to your customers as soon as they're ready.
- Monitor your pipelines, infrastructure and products in production and know when they aren't behaving as expected.

# What does MLOps look like?

- **Develop & train model** that solves a real business problem
- **Package model** so you can use it somewhere else
- **Validate model behavior** – functionally, in terms of responsiveness & regulatory compliance
- **Deploy model** – use the model to make predictions
- **Monitor model** behavior & business value, know **when to replace / deprecate a stale model**

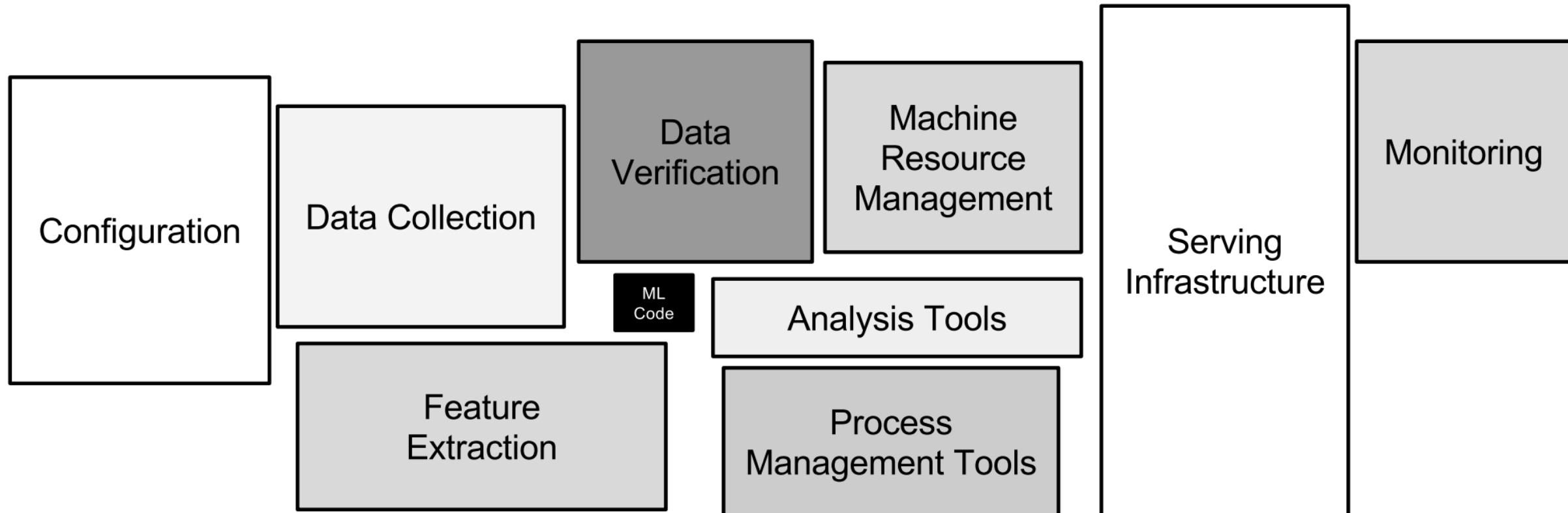


# Traditional vs. ML infused systems



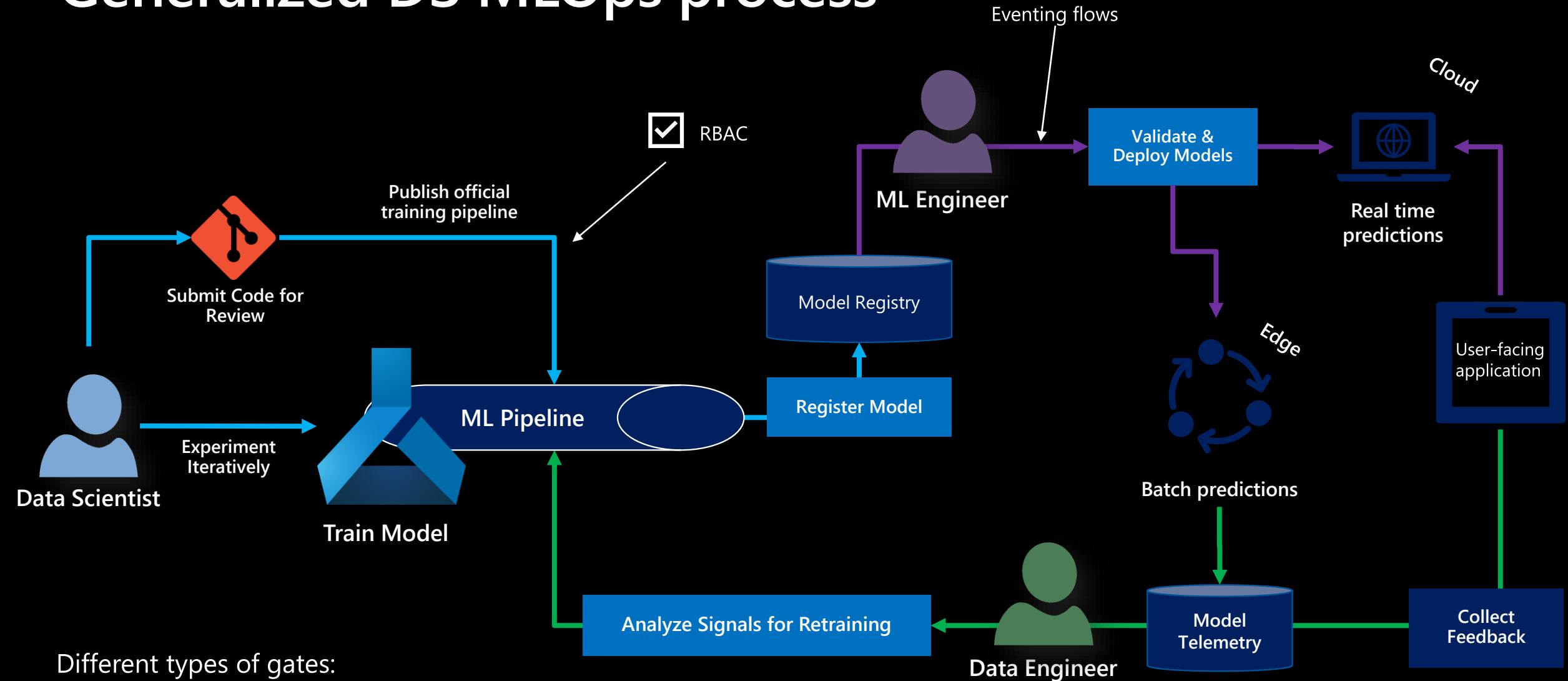
ML introduces two new assets into the software development lifecycle – **data** and **models**. Traditional software systems do not need to be retrained

# More assets & process to manage



Sculley, D.; Holt, Gary; Golovin, Daniel; Davydov, Eugene; Phillips, Todd; Ebner, Dietmar; Chaudhary, Vinay; Young, Michael; Crespo, Jean-Francois; Dennison, Dan (7 December 2015). ["Hidden Technical Debt in Machine Learning Systems"](#)

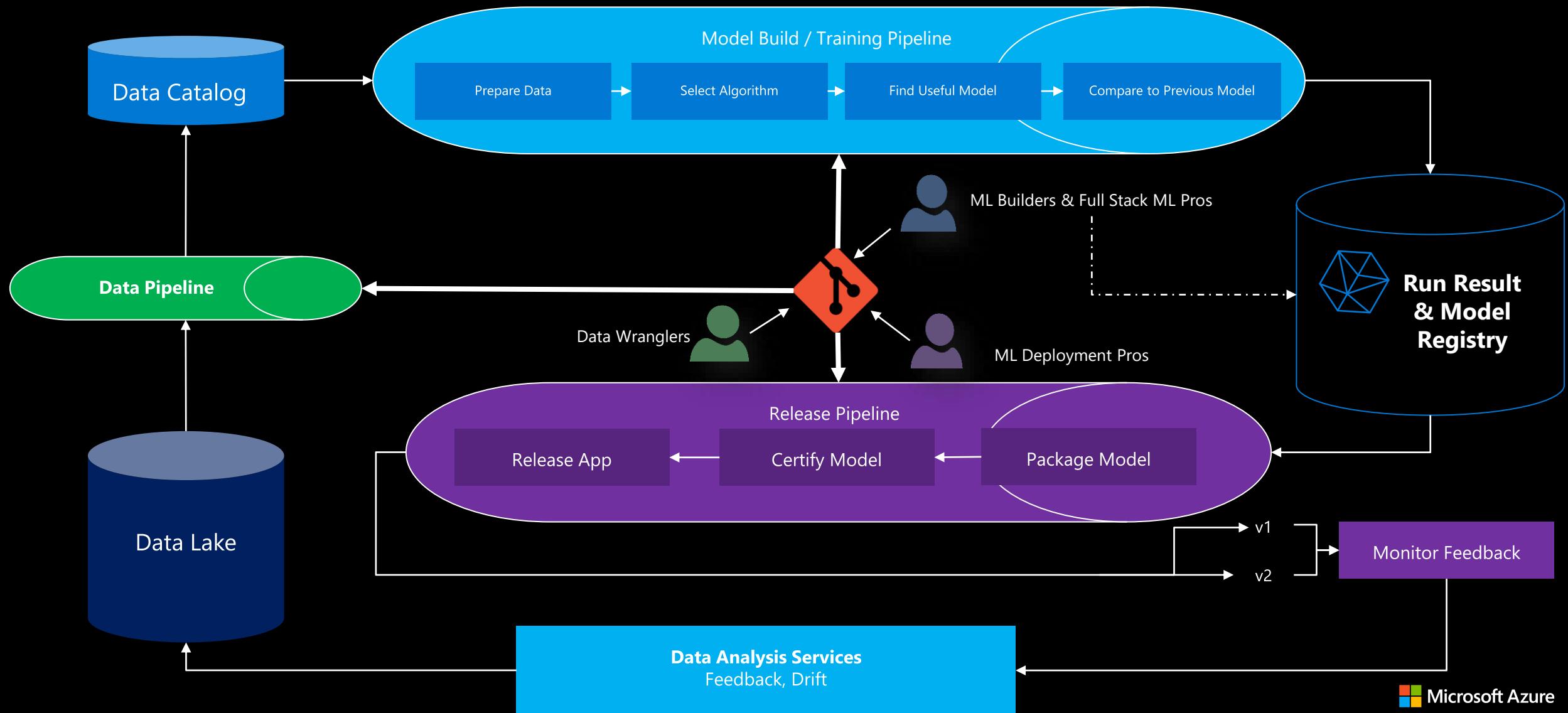
# Generalized DS MLOps process



Different types of gates:

- Cost
- Compliance
- Quota

# MLOps e2e



# There are many jobs & tools involved in production ML



Data Scientist

Azure Machine Learning  
GitHub  
TensorFlow, PyTorch, sklearn  
Azure Compute



IT / Ops



Data Analyst



Business Owner



& many more...



Data Engineer

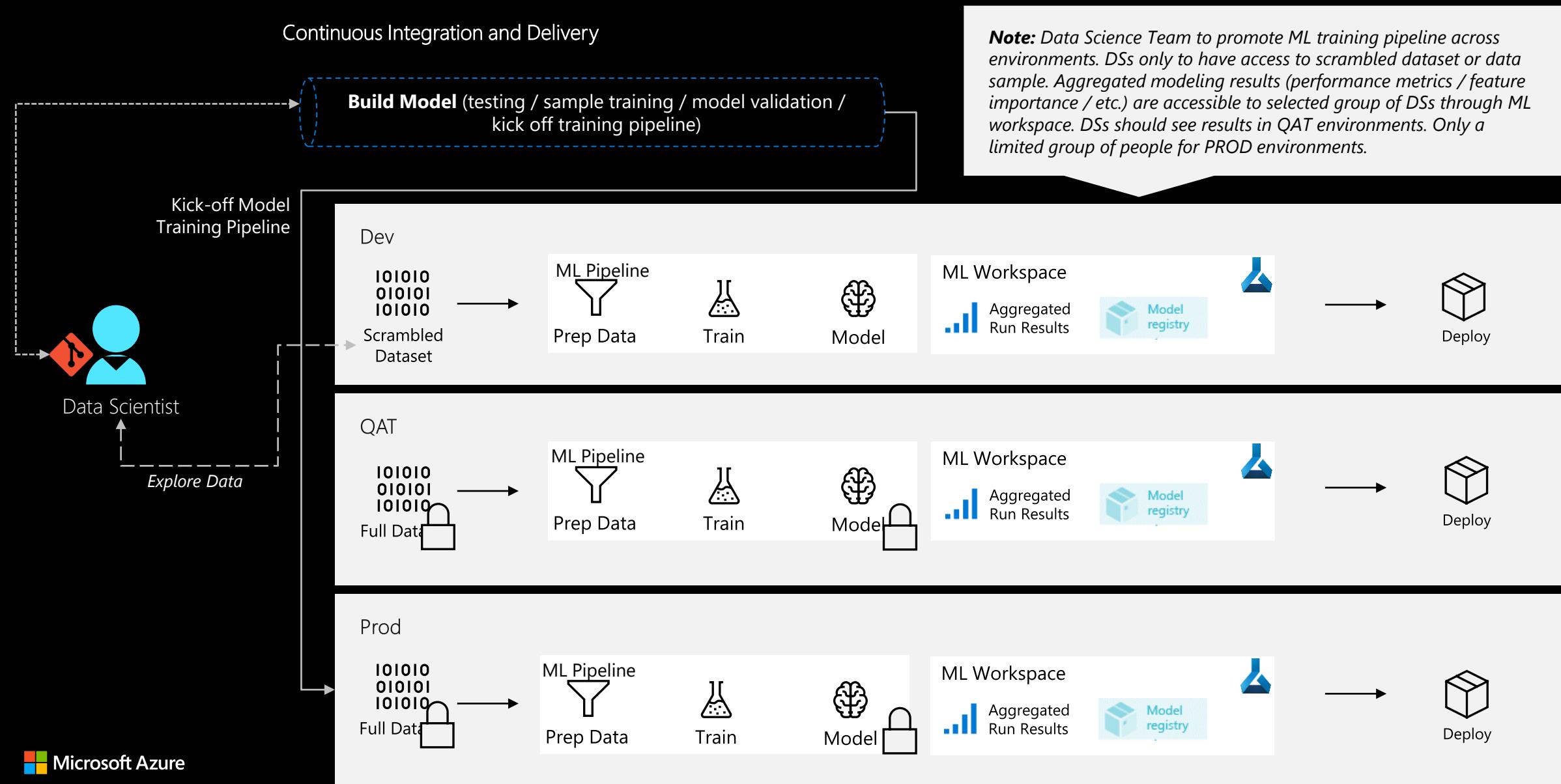
Azure Data Lake  
Azure Data Factory  
Azure DataBricks  
Azure SQL



ML  
Engineer

Azure DevOps  
GitHub  
Azure Kubernetes Service  
Azure IoT Edge  
Azure Monitor

# Model Training – gitops flow



# Promoting Model Release

```
name: akstestgh
type: online
infrastructure: azureml:devplatv2aks
auth_mode: Key
traffic:
  etblue: 50
  etgreen: 50
deployments:
  etblue:
    model: azureml:my-model-1:1
    scale_settings:
      scale_type: manual
      instance_count: 1
    request_settings:
      request_timeout_ms: 3000
    resource_requirements:
      cpu: 1.5
      memory_in_gb: 1.0
  etgreen:
    model: azureml:my-model-1:2
    scale_settings:
      scale_type: manual
      instance_count: 1
    request_settings:
      request_timeout_ms: 3000
    resource_requirements:
      cpu: 1.5
      memory_in_gb: 1.0
```

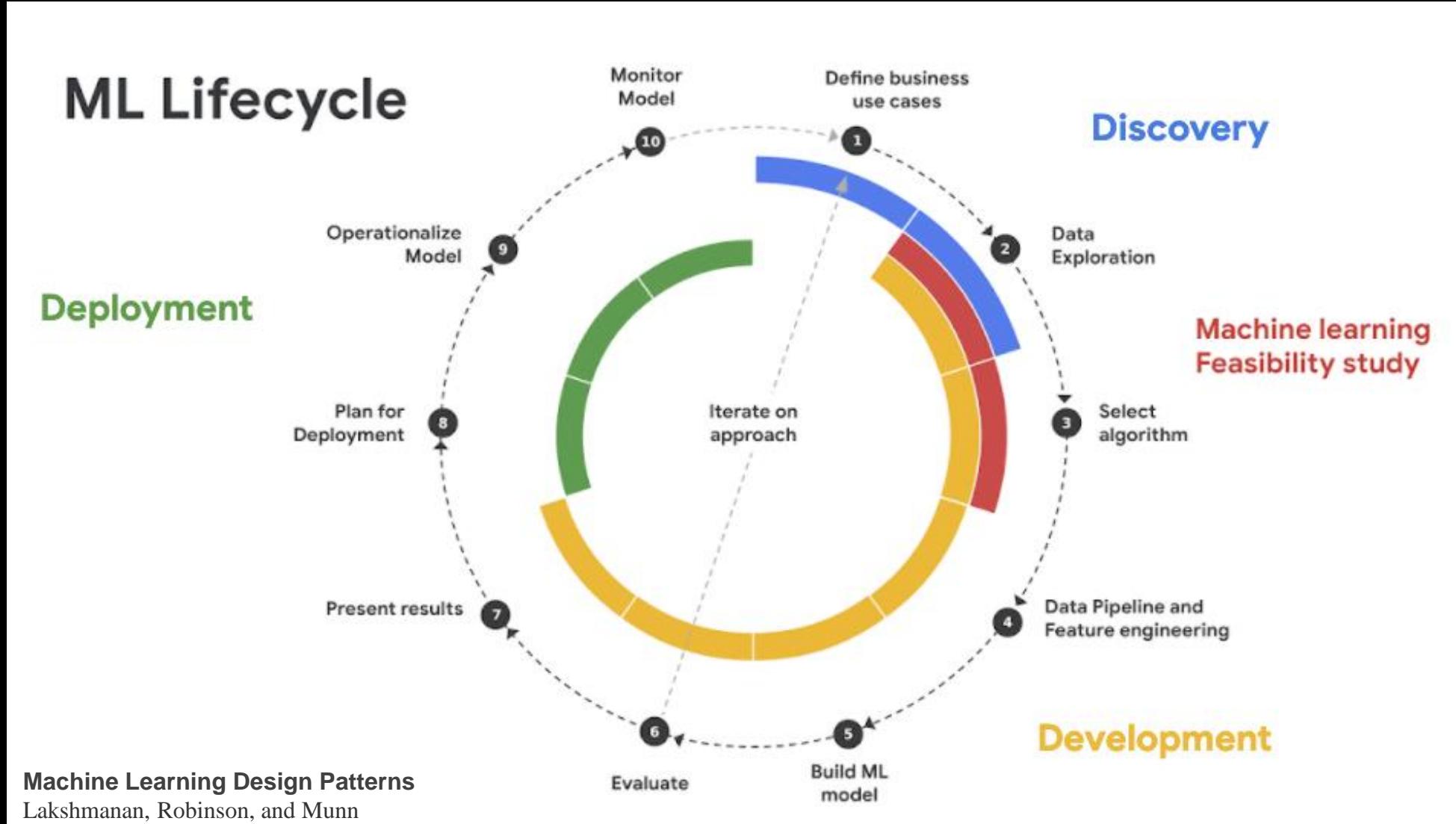


GitOps

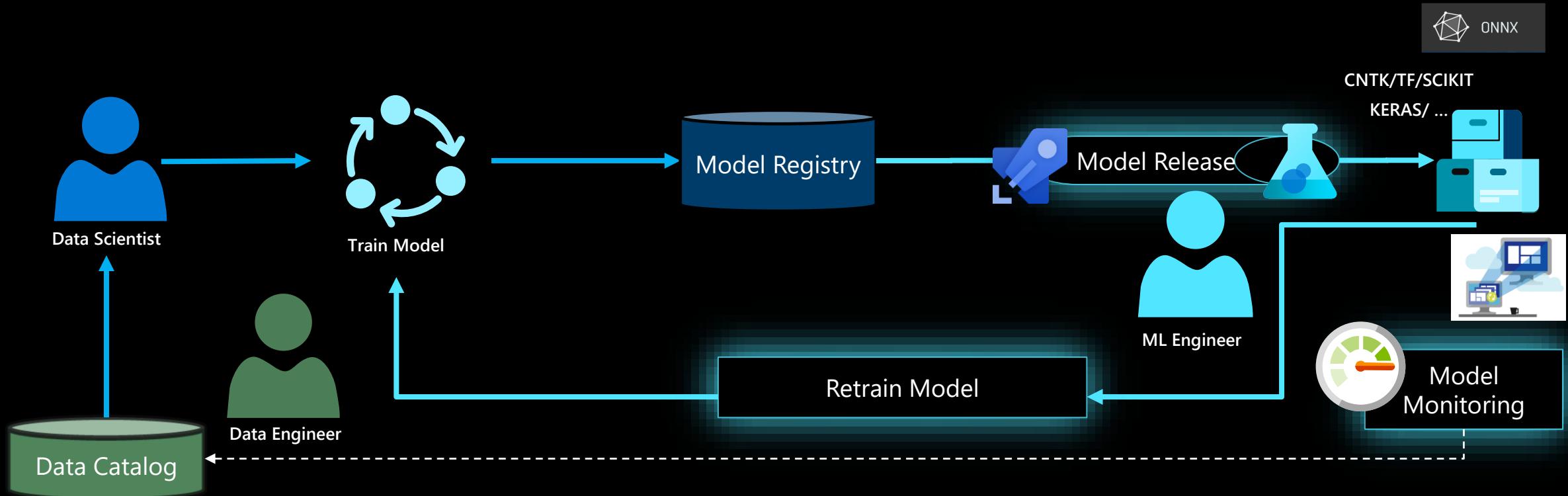
```
name: akstestgh
type: online
infrastructure: azureml:devplatv2aks
auth_mode: Key
traffic:
  etblue: 80
  etgreen: 20
deployments:
  etblue:
    model: azureml:my-model-1:1
    scale_settings:
      scale_type: manual
      instance_count: 1
    request_settings:
      request_timeout_ms: 3000
    resource_requirements:
      cpu: 1.5
      memory_in_gb: 1.0
  etgreen:
    model: azureml:my-model-1:2
    scale_settings:
      scale_type: manual
      instance_count: 1
    request_settings:
      request_timeout_ms: 3000
    resource_requirements:
      cpu: 1.5
      memory_in_gb: 1.0
```

So... how do we implement  
MLOps in the real world?

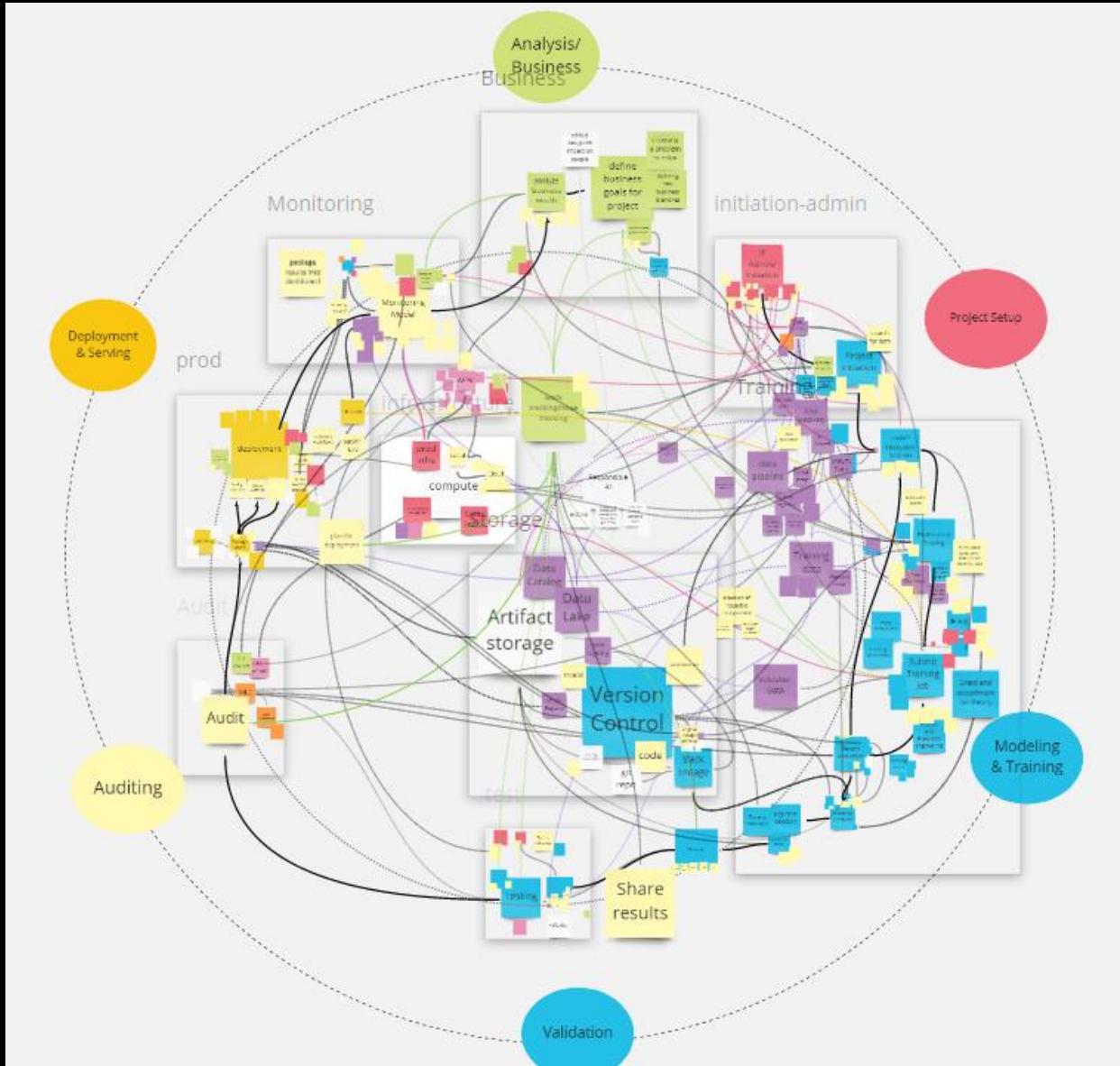
# Machine Learning Lifecycle



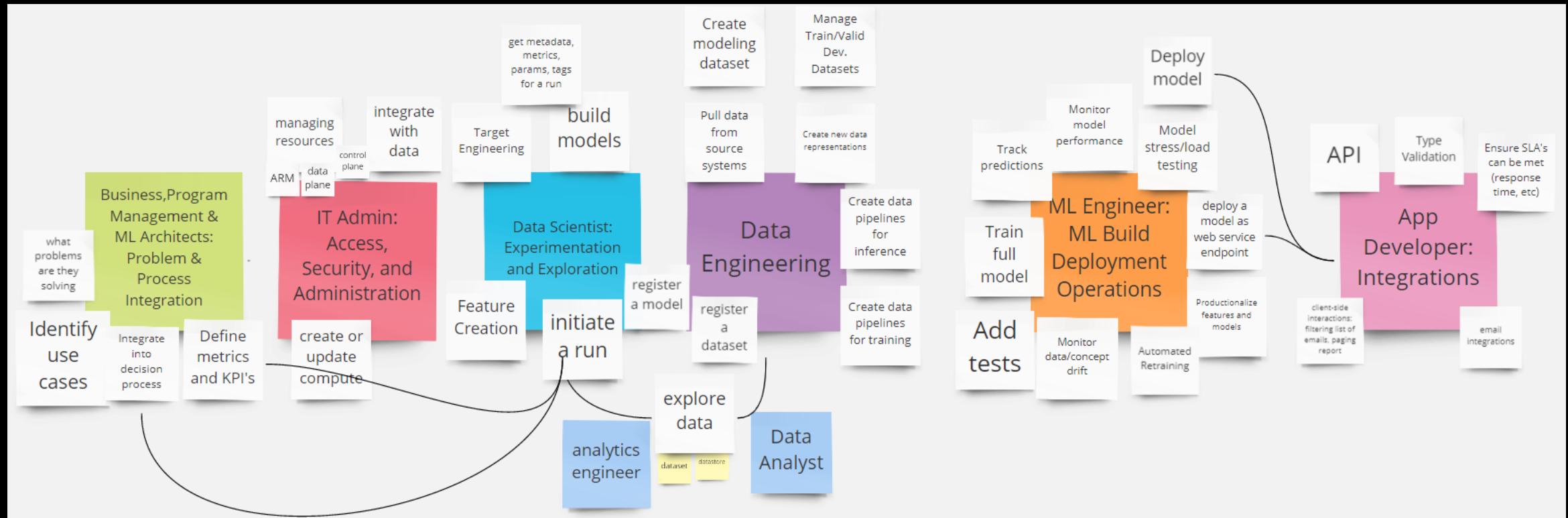
# The ML Lifecycle (Ideal State)



# Machine Learning Lifecycle In Reality: Sociotechnical implications , nonlinearities, dependencies



# The Complexities Of Production ML Roles

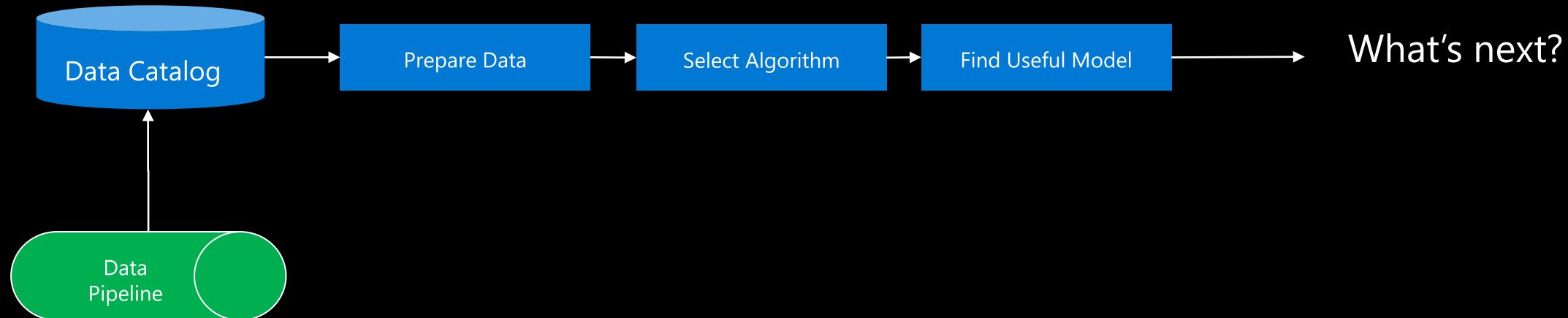


# MLOps – Process Maturity Model

Maturity Level	Training Process	Release Process	Integration into app
Level 1 – No MLOps	Untracked, file is provided for handoff	Manual, hand-off	Manual, heavily DS driven
Level 2- Training Operationalized	Tracked, run results and model artifacts are captured in a repeatable way	Manual release,clean handoff process, managed by SWE team	Manual, heavily DS driven, basic integration tests added
Level 3 – Release Operationalized	Tracked, run results and model artifacts are captured in a repeatable way	Automated, CI/CD pipeline set up, everything is version controlled	Semi-automated, unit and integration tests added, still needs human signoff
Level 4 – Training & Release Operationalized Together	Tracked, run results and model artifacts are captured in a repeatable way, <b>retraining set up</b> based on metrics from app	Automated, CI/CD pipeline set up, everything is version controlled, A/B testing has been added	Semi-automated, unit and integration tests added, <b>may</b> need human signoff

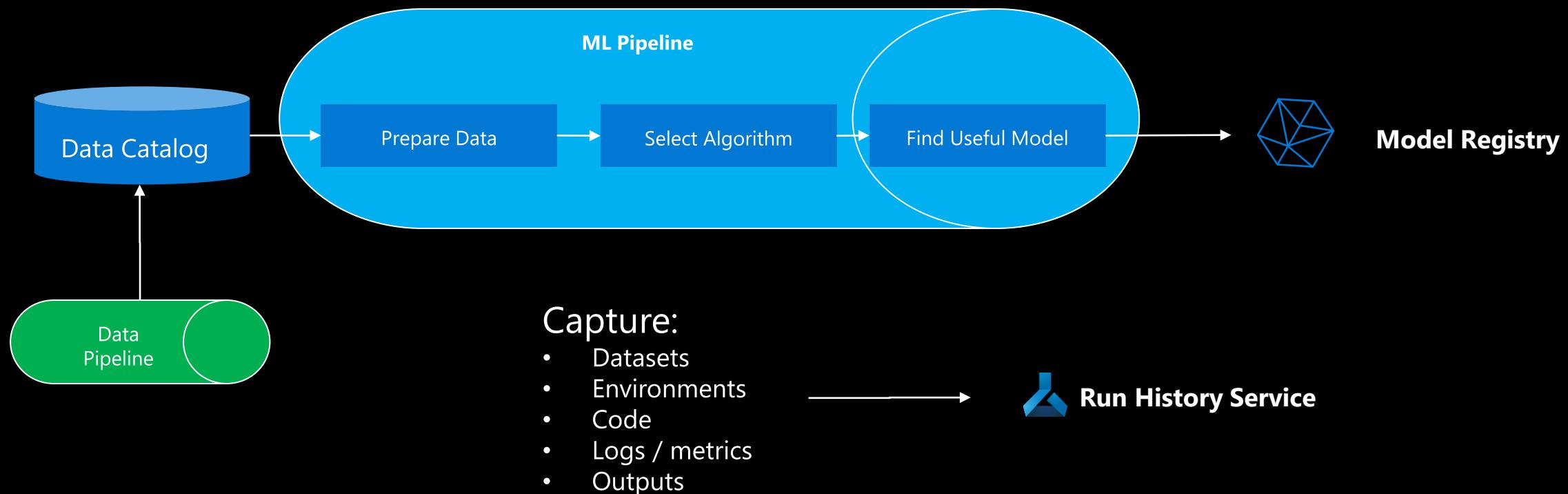
# Level 1 – No MLOps

Interactive, exploratory, get to something useful.



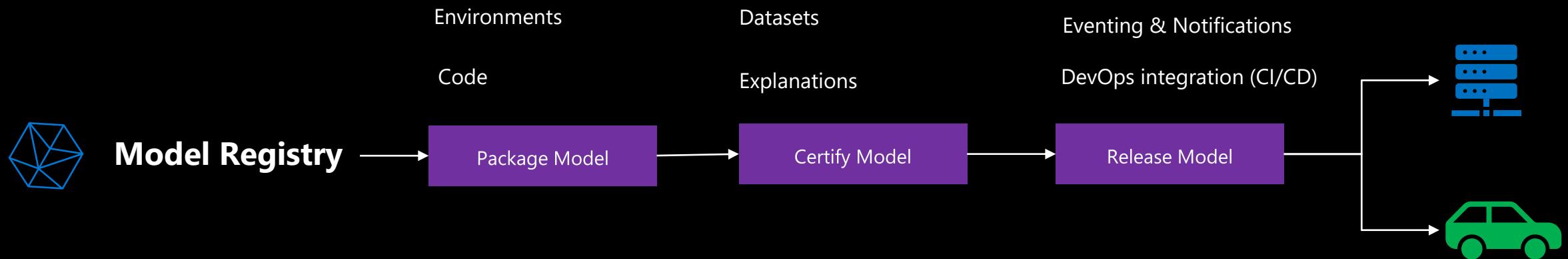
# Level 2 – Reproducible Model Training

Version code, data, ensure model can be recreated.

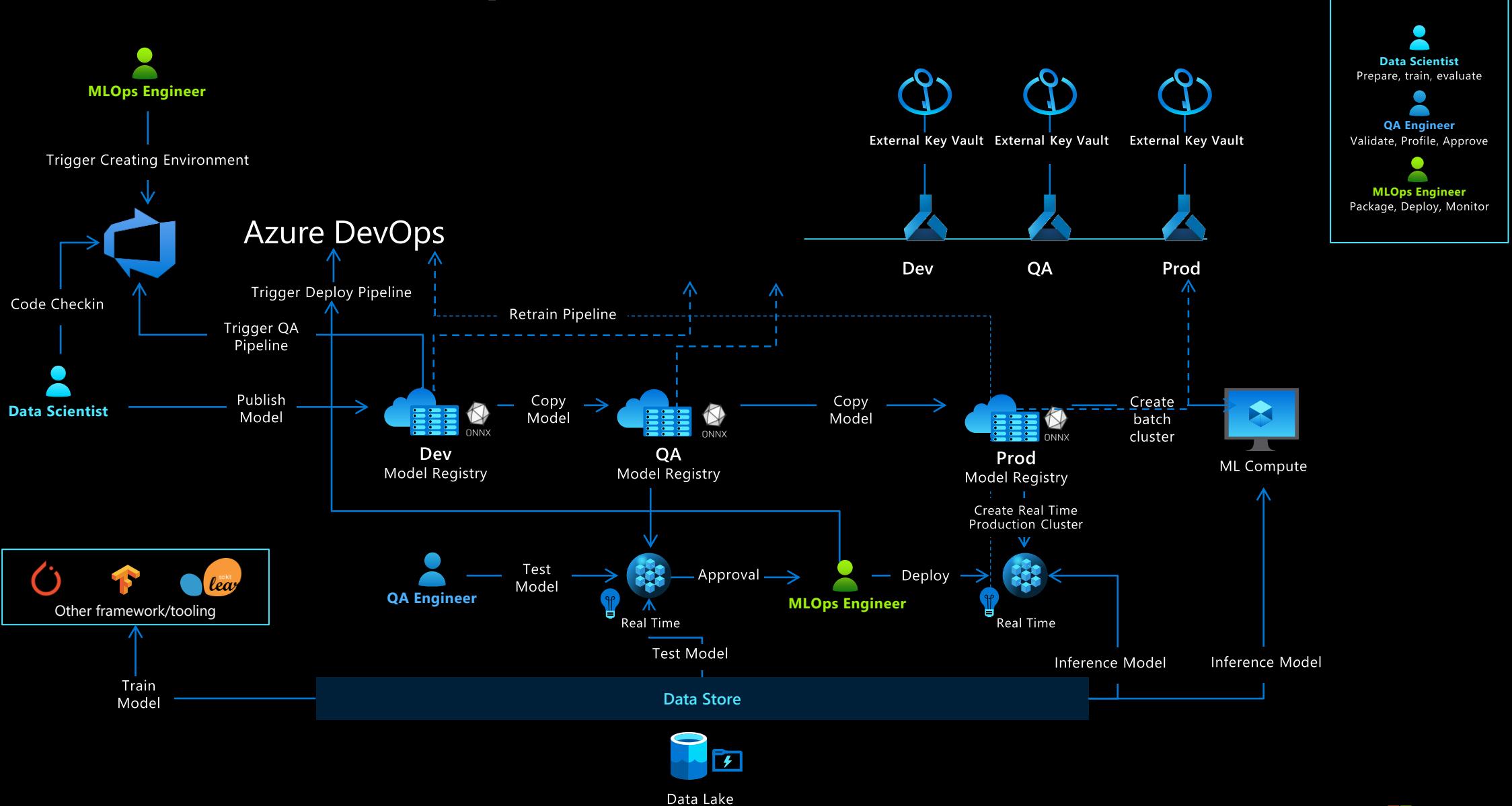


# Level 3 – Managed Model Operationalization

Package, certify, deploy



# Level 4 – E2E MLOps



# How does Azure ML help with MLOps?



# Azure Machine Learning

Tools and services for the end-to-end ML lifecycle

E2E Responsible AI



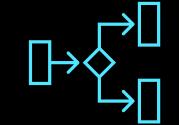
## ML Applications

Automated algo selection, Automated hyperparameter tuning



## Lifecycle Management

Eventing / Orchestration (ML Pipelines, integration w/ other CI/CD tools)



## Managed Training Jobs

Distributed, Sweeps, Workflows



## Managed Scoring Endpoints

Online, Batch



## Asset Management

Code, Data, Models, Environments, Metrics



SDK / CLI / API Interfaces

# Track everything

Track runs, code, data,  
configuration for reproducibility  
of the models

Run 2 ✓ Completed

Refresh Resubmit Cancel

Details Metrics Images Child runs Outputs + logs Snapshot Raw JSON Explanations (preview)

### Properties

Status  
Completed

Created  
Dec 16, 2019 7:35 PM

Duration  
5m 37.96s

Compute target  
[cpu-cluster](#)

Run ID  
sklearn-mnist\_1576553195\_cdcf9fb

Run number  
2

Script name  
train.py

Created by  
Jordan Edwards

Input datasets  
Input name: mnist, ID: [6ee8f4b0-7af3-4842-8950-11ef7ba08e61](#)

Arguments  
--data-folder DatasetConsumptionConfig:mnist --regularization 0.5

Registered models  
[sklearn\\_mnist](#)

mlflow.source.git.repoURL  
<https://github.com/Azure/MachineLearningNotebooks.git>

mlflow.source.git.branch  
master

mlflow.source.git.commit  
3d6caa10a378d77bf8c3b954b733424081b82e04

# Automation & Eventing

Fully managed event routing for all activities in the ML lifecycle

Let's look at some examples...

The screenshot shows the 'Events' blade in the Azure Machine Learning studio. The left sidebar lists various resources: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, and Events (which is selected). Below these are sections for Authoring (Preview) and Assets. The Authoring section includes Automated machine learning, Notebook VMs, and a Visual interface. The Assets section includes Experiments, Pipelines, Compute, Models, Images, Deployments, and Activities. The main content area is titled 'mlopsign-AML-WS - Events' and shows the 'Event Subscription' tab selected. It features a heading 'Events, automated.' and a description about building reactive, event-driven apps with Event Grid. It includes sections for 'TOPIC DETAILS' (Topic Type: Machine Learning, Topic Resource: mlopsign-AML-WS), 'EVENT TYPES' (4 selected: Model registered, Model deployed, Run completed, Dataset drift detected), and 'ENDPOINT DETAILS' (Endpoint Type). The background of the blade is a scenic view of a mountainous landscape.

# Package & deploy anywhere

Capture framework / version / resource requirements

Supports no-code deployment

Open and interoperable

## Supported frameworks:

- scikit-learn
- TensorFlow (SavedModel)
- ONNX (all models)
- Lightgbm
- Pytorch
- ...

The screenshot shows the Microsoft Azure Model Registry interface. On the left, there's a sidebar with a tree icon and a list of registered models: "DiabetesRegressionModel" (selected), "TensorFlowModel", "ONNXModel", "LightGBMModel", "PyTorchModel", and "CustomModel".  
  
The main area has two tabs: "Register a model" and "Deploy a model".  
  
**Register a model (Top Left):**

- Name \***: DiabetesRegressionModel
- Description**: A scikit-learn model designed to detect if someone may have diabetes
- Model Framework**: ScikitLearn
- Model Version**: 0.19.1
- Model file \***: sklearn\_regr...

  
**Deploy a model (Bottom Right):**

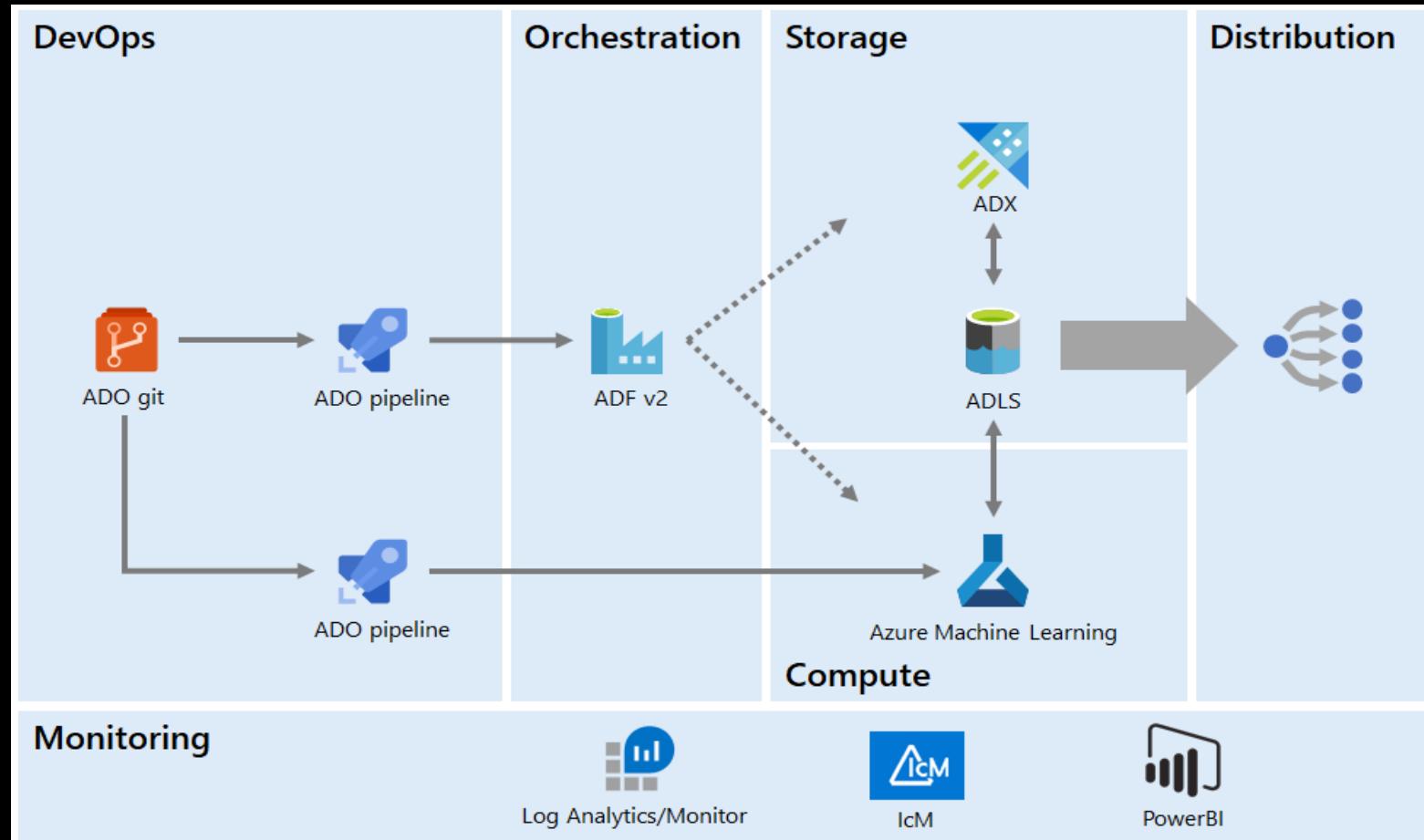
- Name \***: diabetesmodelapi
- Description**: (empty)
- Compute type \***: AKS
- Name \***: ignite-test
- Models**: DiabetesRegressionModel:1
- Enable authentication**: (switch is on)
- Type**: Token-based Authentication

Below these fields, a note says: "This model supports no-code deployment. You may optionally override the default environment and driver file." There is also a checkbox for "Use custom deployment assets".  
  
In the bottom right corner, there is a small modal window showing deployment details:

Last updated on	11/2/2019 9:55:3 AM
Compute target	ignite-test
REST endpoint	http://52.224.223.47:80/api/v1/service/diabetesmodelapi/score
Key-based authentication enabled	false
Token-based authentication enabled	true
CPU	0.1

# Common Deployment & Orchestration Patterns

Data Factory  
DataLake  
DevOps Pipeline



# Data Factory integration

Now every time data drift is detected, your data factory pipeline will automatically be triggered. View details on your data drift run and machine learning pipeline on the [new workspace portal](#).

Endpoints						
Real-time endpoints		Pipeline endpoints				
<input type="button" value="Refresh"/>	<input type="button" value="Disable"/>	<input type="button" value="Enable"/>	<input type="checkbox"/> View disabled	<input type="text"/>	<input type="button" value="Search to filter items..."/>	
Name	Description	Modified on	Modified by	Last run submit time	Last run status	Status
My_New_Pipeline	My Published Pipeline D...	Oct 31st, 2019 12:51 AM		Oct 31st, 2019 2:19 AM	● Running	Active
DataDriftPipeline-68ddef0f	Pipeline for run_invoker.py	Oct 30th, 2019 11:03 PM		Oct 31st, 2019 1:24 AM	● Finished	Active

< Prev Next >

When a resource event occurs

\* Subscription

\* Resource Type: Microsoft.MachineLearningServices.Workspaces

\* Resource Name

Event Type Item - 1

+ Add new item

Add new parameter

Connected to shipatel@microsoft.com

When a resource event occurs

Choose an action

Search: azure data factory

For You All Built-in Standard Enterprise Custom

Azure Data Factory

Triggers Actions

Create a pipeline run Azure Data Factory

Create a pipeline run Azure Data Factory

Data Factory Publish all Validate all Refresh Discard all Data flow debug ARM template

Factory Resources Pipelines 1 Training Pipeline Datasets 1 new\_dataset Data flows 0

Activities Search activities Move & transform Copy data Data flow Azure Data Explorer Azure Function Batch Service Data Lake Analytics

Copy data Copy data1 ML Execute Pipeline ML Execute Pipeline1

# DevOps Integration

Automate training & deployment  
into existing release management  
processes

The screenshot shows the Microsoft Azure DevOps interface. At the top, a modal window titled "Add an Azure Resource Manager service connection" is open, showing fields for "Connection name" (mlops-ignite), "Scope level" (AzureMLWorkspace), "Subscription" (AzureML Nursery (15ae9cb6-95c1-483d-a0e3-b1a1a3b0632)), "Resource Group" (ignite), and "Machine Learning Workspace" (ignite). Below the modal, the main pipeline interface is visible. The pipeline title is "Deploy employee attrition model and ... > Release-16". The pipeline has two stages: "Release" and "Stages". The "Release" stage contains a "Continuous deployment" card for "Microsoft.VisualStudio..." dated 10/17/2019, 1:17 PM. The "Artifacts" section lists three items: "\_IBM\_attrition\_explainer" (3 files), "\_IBM\_attrition\_mo..." (8 files), and "\_azureml-workshop-20..." (1 file, master branch). The "Stages" stage contains a "Deploy to Test" card with a green checkmark indicating "Succeeded" and 3 warnings from 10/17/2019, 1:38 PM. A second "Deploy to Production" card shows a person icon and a blue dot indicating it is "Queued", with the note "Waiting in the deployment queue for 15 days". At the bottom right, the Microsoft Azure logo is visible.

# Azure Pipeline technology comparison

## Which Azure pipeline technology should I use?

The Azure cloud provides several types of pipeline, each with a different purpose. The following table lists the different pipelines and what they are used for:

Scenario	Primary persona	Azure offering	OSS offering	Canonical pipe	Strengths
Model orchestration (Machine learning)	Data scientist	Azure Machine Learning Pipelines	Kubeflow Pipelines	Data -> Model	Distribution, caching, code-first, reuse
Data orchestration (Data prep)	Data engineer	Azure Data Factory pipelines	Apache Airflow	Data -> Data	Strongly typed movement, data-centric activities
Code & app orchestration (CI/CD)	App Developer / Ops	Azure Pipelines ↗	Jenkins	Code + Model -> App/Service	Most open and flexible activity support, approval queues, phases with gating



# GitHub Actions Integration

Create data / environments / jobs / endpoints directly from GitHub Actions

Azure ML will write output directly to GHA and pull requests as desired.

```
- name: azure login
  uses: azure/login@v1
  with:
    creds: ${{secrets.AZURE_TOKEN}}
- run: az config set defaults.workspace=devplatv2
- run: az config set defaults.group=demorg
- run: az account set -s 92c76a2f-0e1c-4216-b65e-abf7a3f34c1e
- run: az ml data create --file examples/blob_dataset.yml
- run: az ml environment create --file examples/fastai-vision-env.yml
- run: az ml job create --file examples/commandjob.yml --name helloworld_${GITHUB_RUN_ID}
- run: az ml job create --file examples/fastai_mnist_job.yml --name fastai_mnist_${GITHUB_RUN_ID}
- run: az ml job create --file examples/fastai_pets_job.yml --name fastai_pets_${GITHUB_RUN_ID}
```

The screenshot shows two GitHub comments from the 'github-actions' bot. The top comment, dated Oct 14, 2019, is titled 'ML Workflow For SHA 5287272 has been instantiated.' It lists built Docker images: 'hamelsmu/ml-cicd' and 'hamelsmu/ml-cicd-gpu'. The bottom comment, also dated Oct 14, 2019, is titled 'Model Evaluation Results' and displays a table of evaluation metrics for 'candidate' and 'baseline' models.

Category	Run ID	SHA	Train Loss	Val Loss	Acc	Val Acc	Runtime
candidate	ddscgocn	5287272	0.366	0.534	0.862	0.796	542.478
baseline	d2mg9r7l	0fbe4ae	0.392	0.527	0.851	0.798	577.173

# Send emails / hit a webhook whenever a run completes

Streamline model reuse & consumption

The screenshot shows the Logic Apps Designer interface in Microsoft Azure. The top navigation bar includes 'Save As', 'Discard', 'Designer', 'Code view', 'Templates', 'Connectors', 'Help', and a 'Dashboard' link. The main area displays a workflow titled 'When a resource event occurs'. This step is connected to a 'Create a new release' step, which is highlighted with a blue border. The 'Create a new release' step has the following configuration:

- Account Name:** aidemos
- Project Name:** MLops
- Release Definition Id:** deploy workshop model
- Description:** Deployment triggered because a new model meeting prefix constraints is in place.
- Is Draft:** Specifies whether the release is a draft.
- Reason:** ContinuousIntegration
- Add new parameter:** (dropdown menu)

Below this step, there is a note: "Connected to jordan@bing.com. Change connection." At the bottom of the workflow, there is a '+ New step' button and a 'Update a work item' step with an 'Azure DevOps' connector and a 'Create file' action.

**Azure Event Grid Tenant:** Microsoft

**Sign in to create a connection to Azure Event Grid.**

**When a resource event occurs**

**Create a new release**

**Subscription:** (topicSubscriptionId)

**Resource Type:** Microsoft.MachineLearningServices.Workspaces

**Resource Name:** /subscriptions/92c76a2f-0e1c-4216-b65e-abf7a3f34c1e/resourceGroups/ml-RG/providers/Microsoft.MachineLearningServices/workspaces/mllopsign-AN

**Event Type Item - 1:**

- Microsoft.MachineLearningServices.DatasetDriftDetected
- Microsoft.MachineLearningServices.ModelDeployed
- Microsoft.MachineLearningServices.ModelRegistered
- Microsoft.MachineLearningServices.RunCompleted

**Enter custom value**

**+ New step**

**Update a work item**

**Azure DevOps**

**Create file**

**Microsoft Azure**

# Set up a data drift monitor...

Compare datasets over time  
Determine when to take a closer look

Monitor settings

Settings for the data drift scheduled pipeline that will monitor the target dataset and send an email alert if the data drift percentage is above the set threshold.

Enable  Monitor enabled

Latency (hrs)

Email addresses

Threshold

Azure ML Workspace (Preview) <https://ml.azure.com/data/monitor/107bugbash4?wsid=/subscriptions/60582a10-b9fd-49f1-a546-c4194134bba8/resourcegroups/copeters...>

driftDemoWS > Datasets > Dataset monitors > 107bugbash4

107bugbash4

Settings Backfill Refresh

Start date: Tue Jan 01 2019 End date: Sun Sep 01 2019

Drift overview

Data drift magnitude

Drift contribution by feature

Feature details

Select feature: Select metrics:

Percentage

100%  
80%  
60%  
40%  
20%  
0%

19-01-01 19-02-01 19-03-01 19-04-01 19-05-01 19-06-01 19-07-01 19-08-01 19-09-01

Date

100%  
80%  
60%  
40%  
20%  
0%

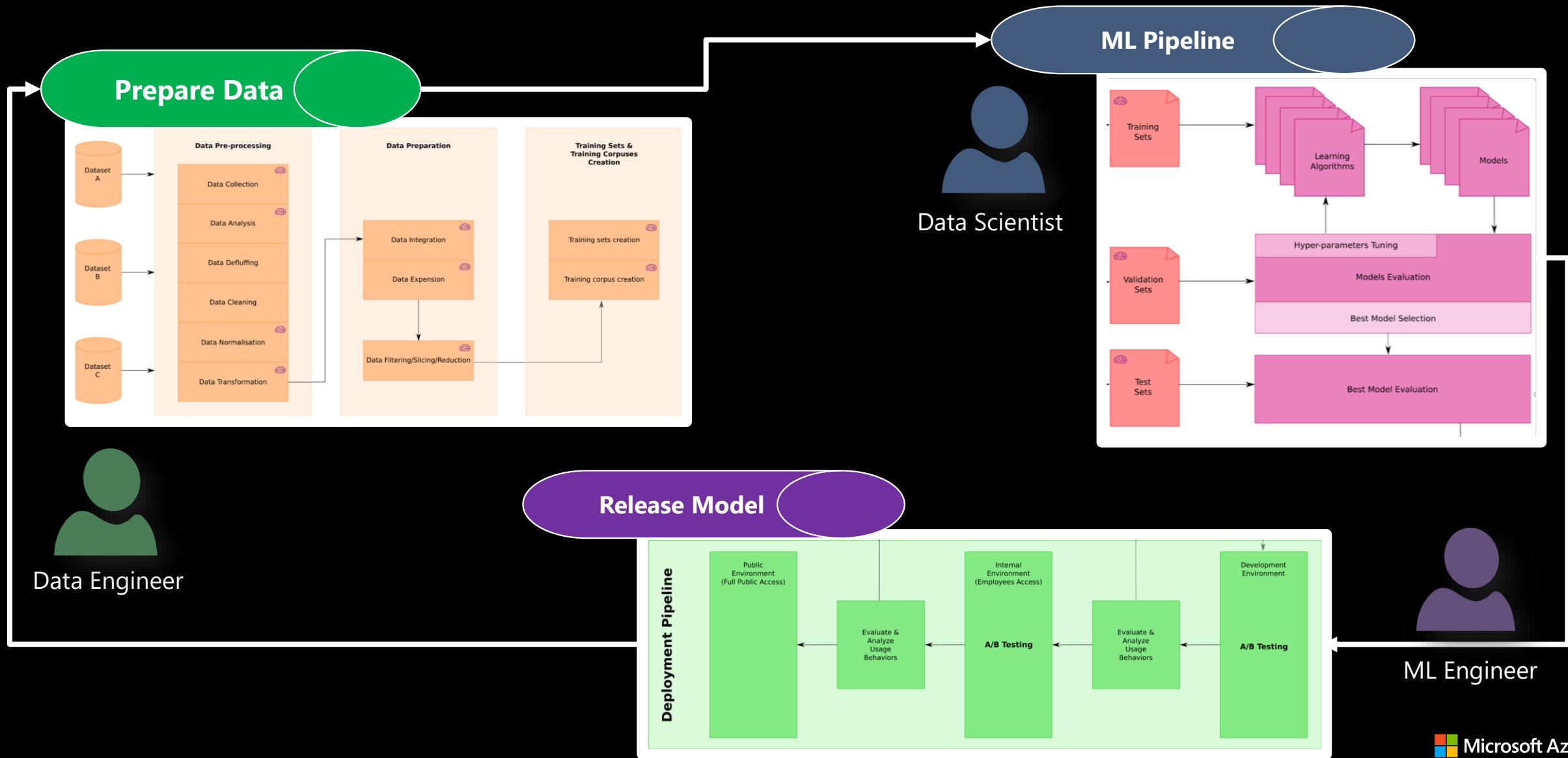
19-01-01 19-02-01 19-03-01 19-04-01 19-05-01 19-06-01 19-07-01 19-08-01 19-09-01

temperature countryOrRegion windSpeed  
windAngle precipTime snowDepth  
latitude longitude elevation  
stationName

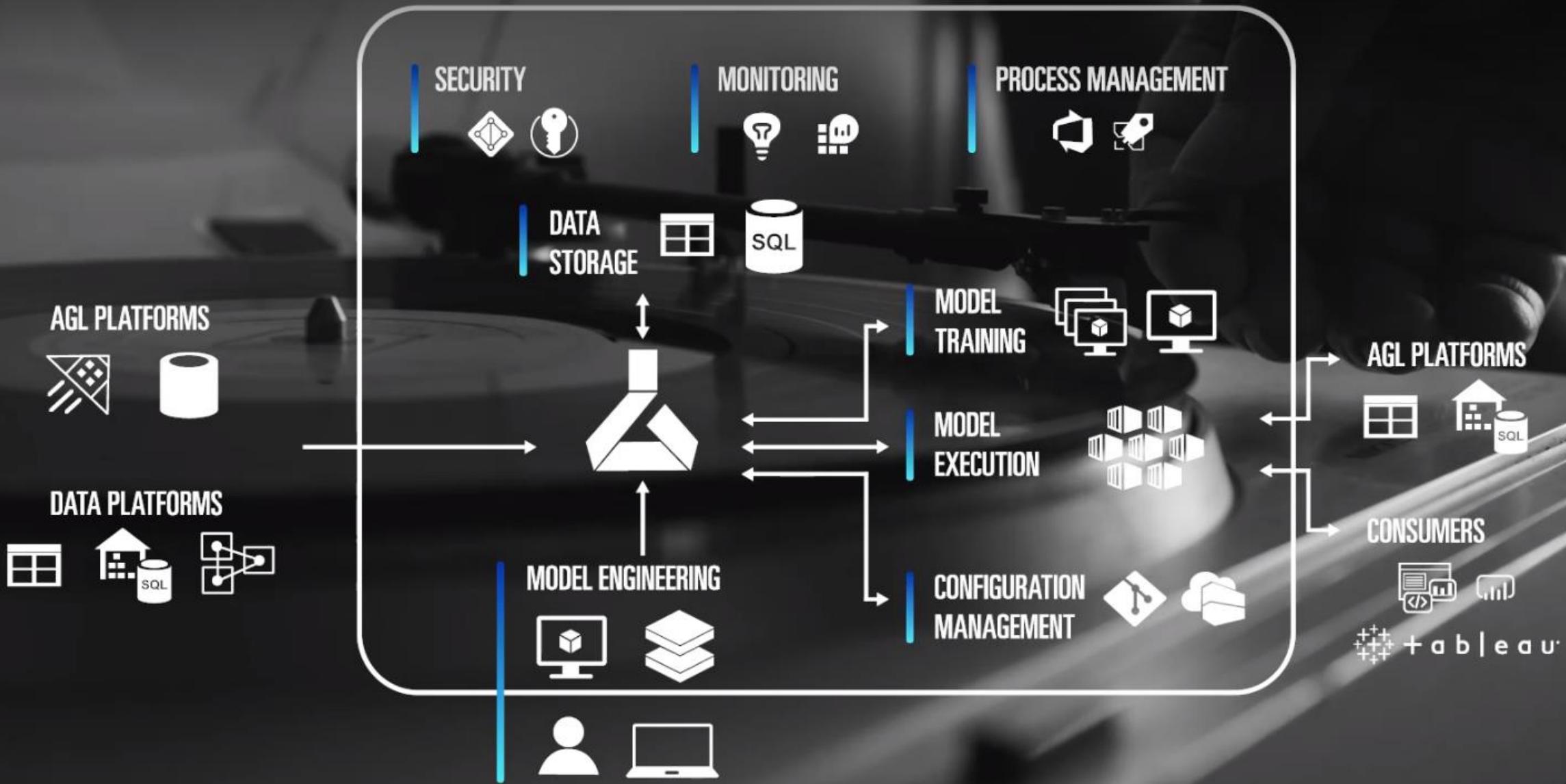
# Demo: Azure ML for MLOps

# Customer Examples

# Video Call Prediction for Microsoft Teams



[Microsoft Stream](#) : boiler plate code that research build and share with all



# Carhartt connects customers to all-American clothing with Azure Machine Learning

7 views · 0 likes · 0 comments



## APPLICATION ARCHITECTURE

MARKET ATTACK, HOW IT WAS DEVELOPED



FRONT-END  
GIS SERVER



BACKEND  
GIS SERVER



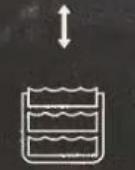
SQL DW



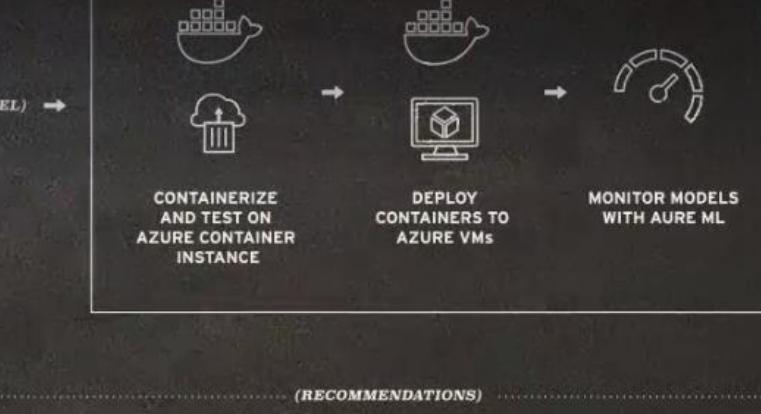
AZURE  
DATABRICKS  
FOR ETL



AZURE  
DATABRICKS  
FOR BAYESIAN  
DECISION NETWORK



ADLSv2



(HISTORICAL DATA)

### CARHARTT DEMAND SYSTEM ON AZURE MACHINE LEARNING

BUILD MODELS



NOTEBOOKS

TRAIN MODELS



AZURE ML COMPUTE

VALIDATE MODELS



CONTAINERIZE AND  
TEST ON AZURE  
CONTAINER INSTANCE

DEPLOY MODELS



DEPLOY CONTAINERS  
TO AZURE  
KUBERNETES CLUSTER

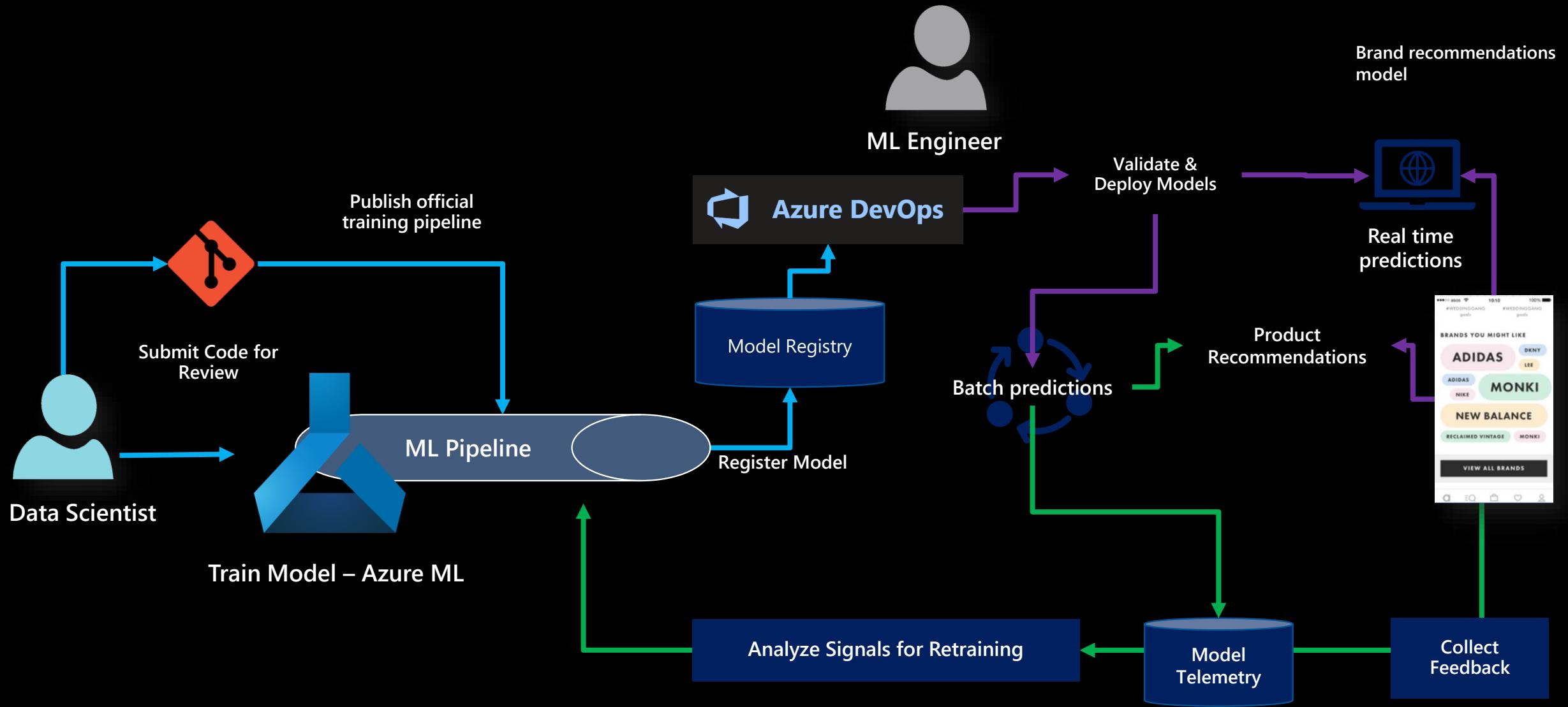
MONITOR MODELS



MONITOR MODELS  
WITH AZURE M

[Microsoft Stream](#)  
or MLOps pipeline and that is actually doing

# Leveraging MLOps to ship recommender systems.



# Wrapping up

# Summary – MLOps process maturity model & best practices

1. Track how the model was created
2. Ensure model can be reproduced by automation
3. Provide a process to notify ML engineers / model consumers when a new model has been produced.
4. Provide a release process appropriate for your business to package, certify and vet the model for use in production.
5. Ensure your monitoring solution captures the model's behavior to bridge the "retraining gap"

Maturity Level	Training Process	Release Process	Integration into app
Level 1 – No MLOps	Untracked, file is provided for handoff	Manual, hand-off	Manual, heavily DS driven
Level 2- Training Operationalized	Tracked, run results and model artifacts are captured in a repeatable way	Manual release,clean handoff process, managed by SWE team	Manual, heavily DS driven, basic integration tests added
Level 3 – Release Operationalized	Tracked, run results and model artifacts are captured in a repeatable way	Automated, CI/CD pipeline set up, everything is version controlled	Semi-automated, unit and integration tests added, still needs human signoff
Level 4 – Training & Release Operationalized Together	Tracked, run results and model artifacts are captured in a repeatable way, <b>retraining set up</b> based on metrics from app	Automated, CI/CD pipeline set up, everything is version controlled, A/B testing has been added	Semi-automated, unit and integration tests added, <b>may</b> need human signoff

# Key Takeaways

## Better together: ML + DevOps mindset

MLOps provides structure for building, deploying and managing an enterprise-ready AI application lifecycle

## MLOps enhances delivery

Adoption will increase the agility, quality and delivery of AI project teams.

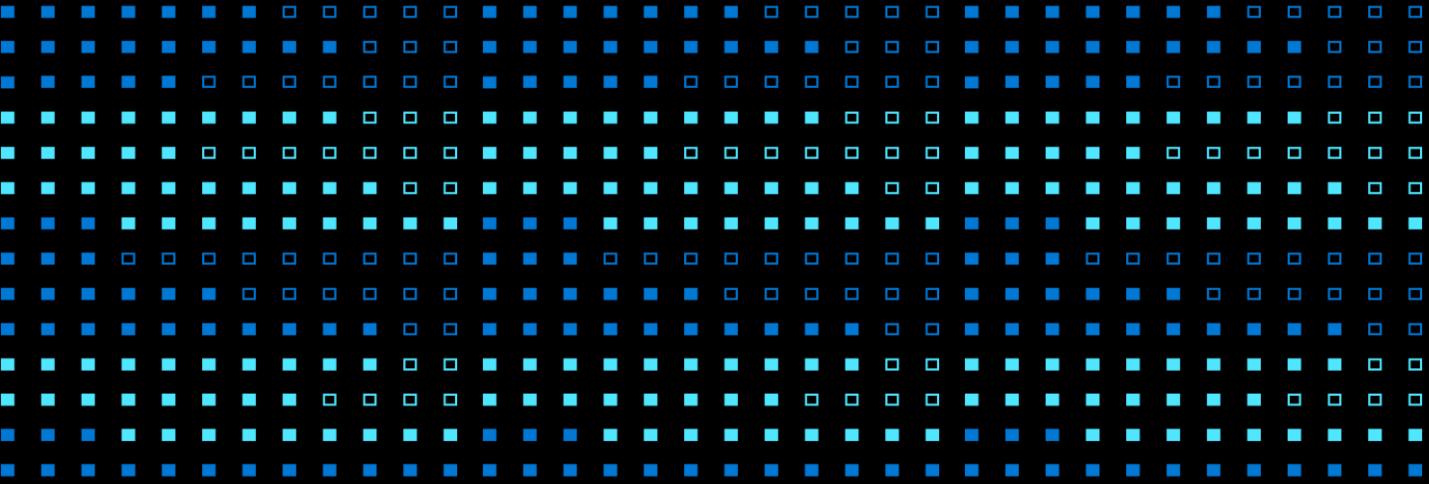
## More than technology

MLOps is a conversation about people, process and technology  
AI principles and practices need to be understood by all roles



# Q&A

Please submit your questions into the Q&A window. We have Subject Matter Experts ready to answer your questions.





Thank you for joining us.