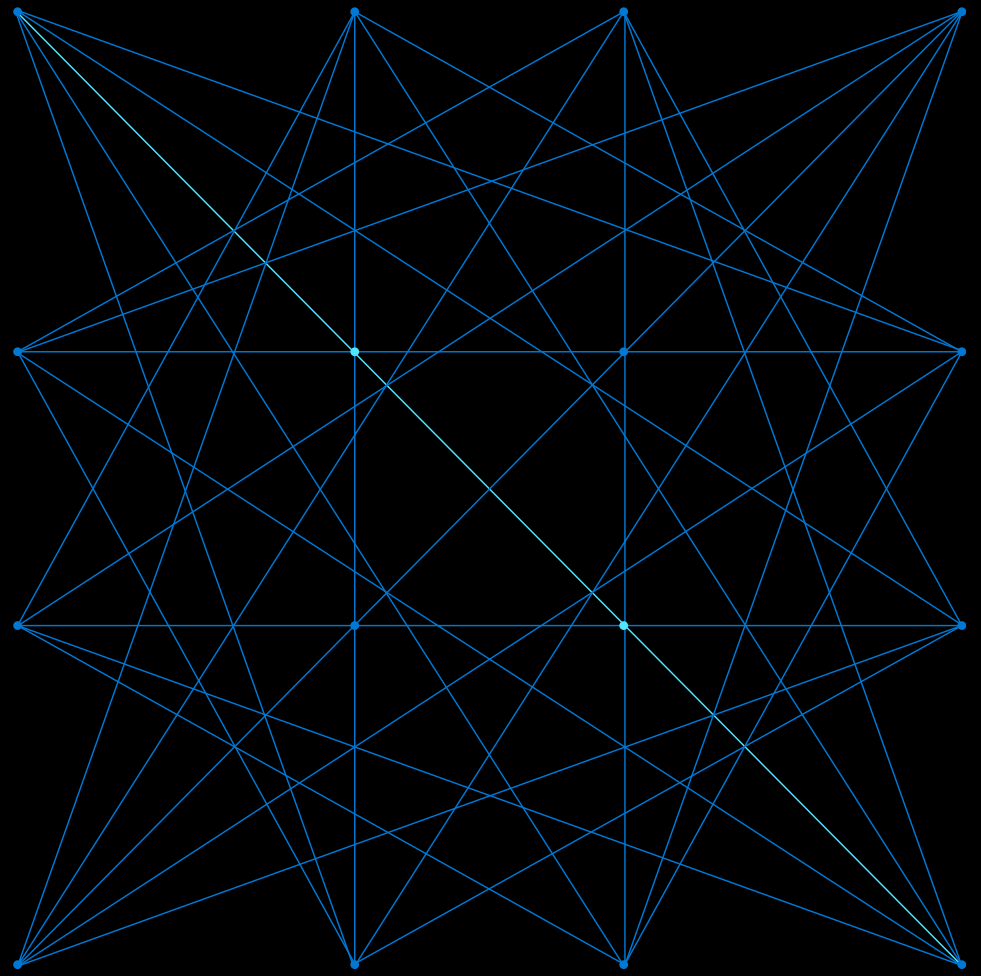
Microsoft Azure

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

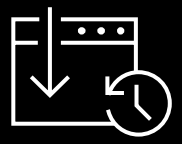
Principal PM Manager

Buzzword Bingo!

MLOps
DevOps for ML
"Software Engineering 2.0"
GitOps for ML
Git for Data

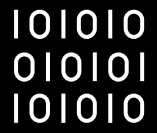
MLOps == How to bring ML to production

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.



Process

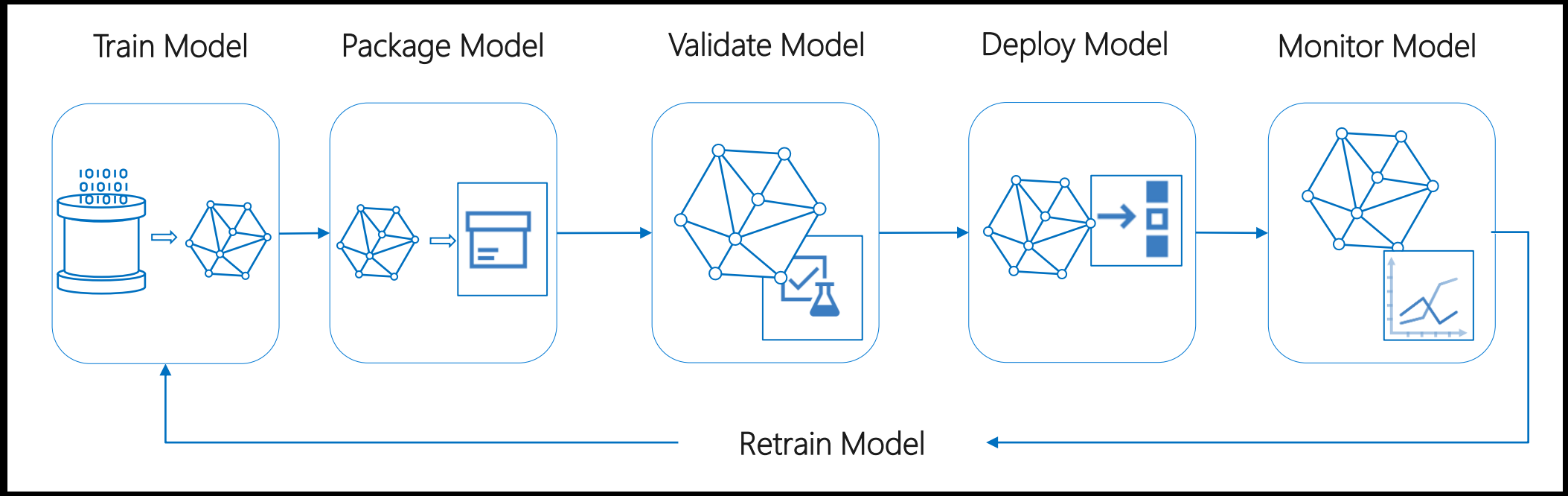
- Provide templates to bootstrap your infrastructure and model development environment, expressed as code.
- Automate the entire process from code commit to production.

Platform

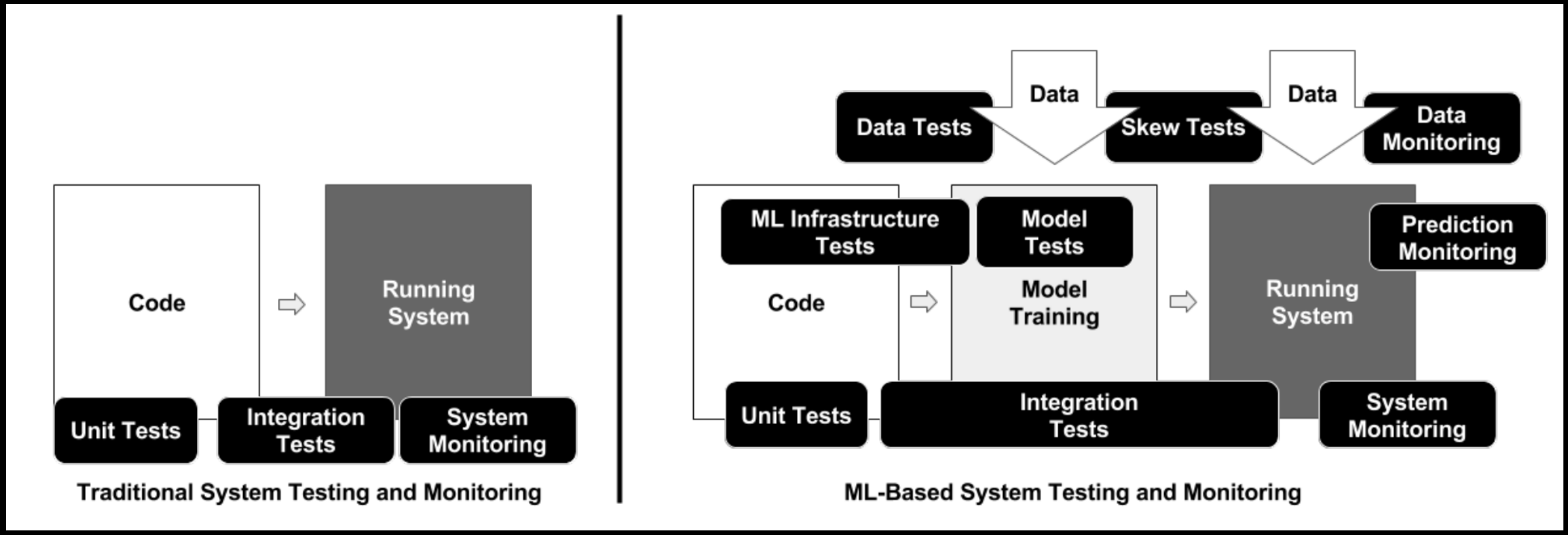
- Safely deliver features to your customers as soon as they're ready.
- Monitor your pipelines, infrastructure and products in production and know when they aren't behaving as expected.

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**

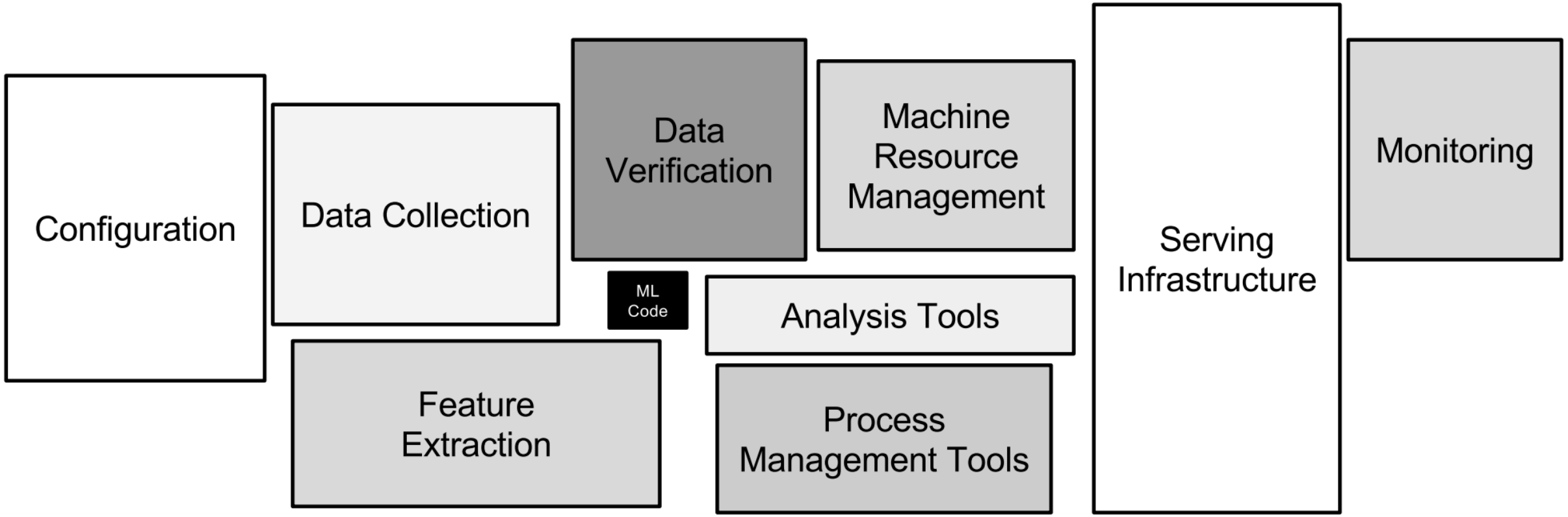


Traditional vs. ML infused systems



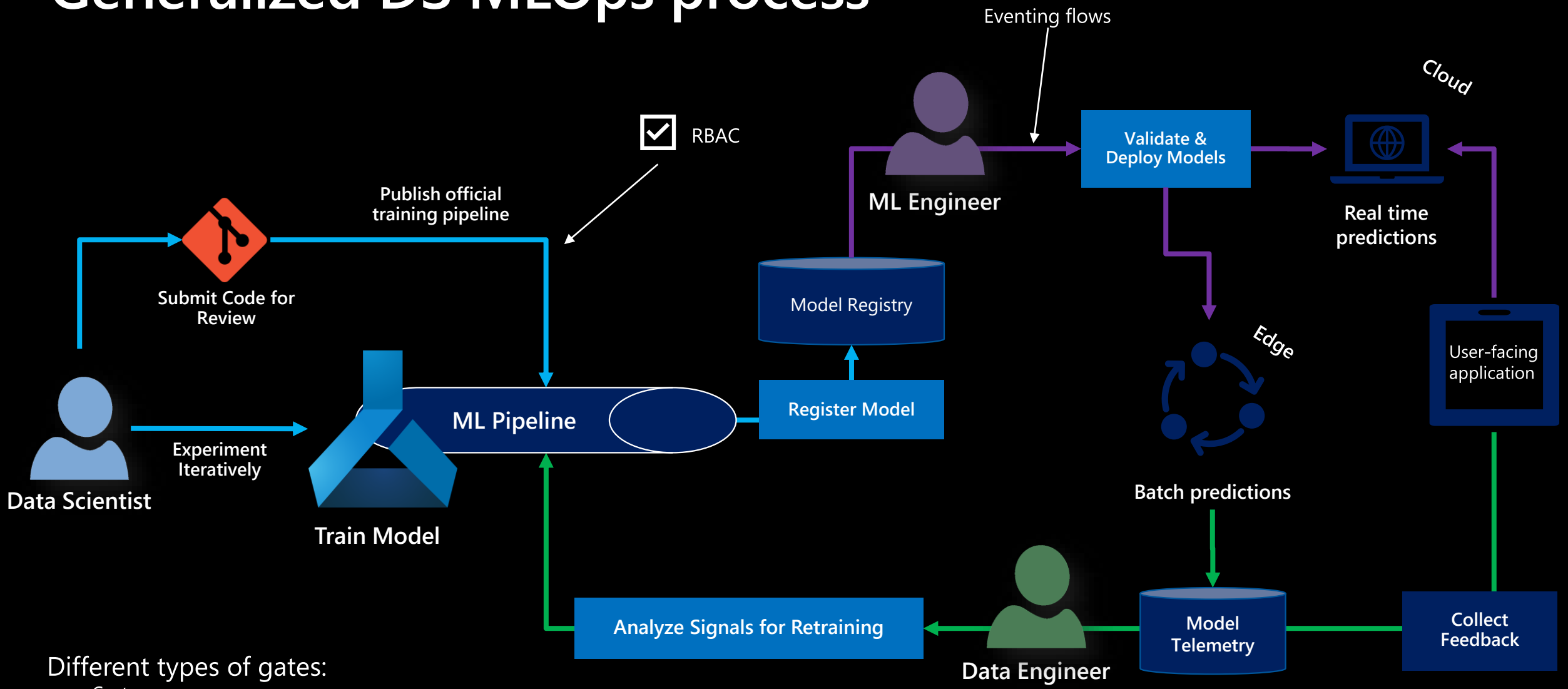
ML introduces two new assets into the software development lifecycle – **data** and **models**. Traditional software systems do not need to be retrained

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

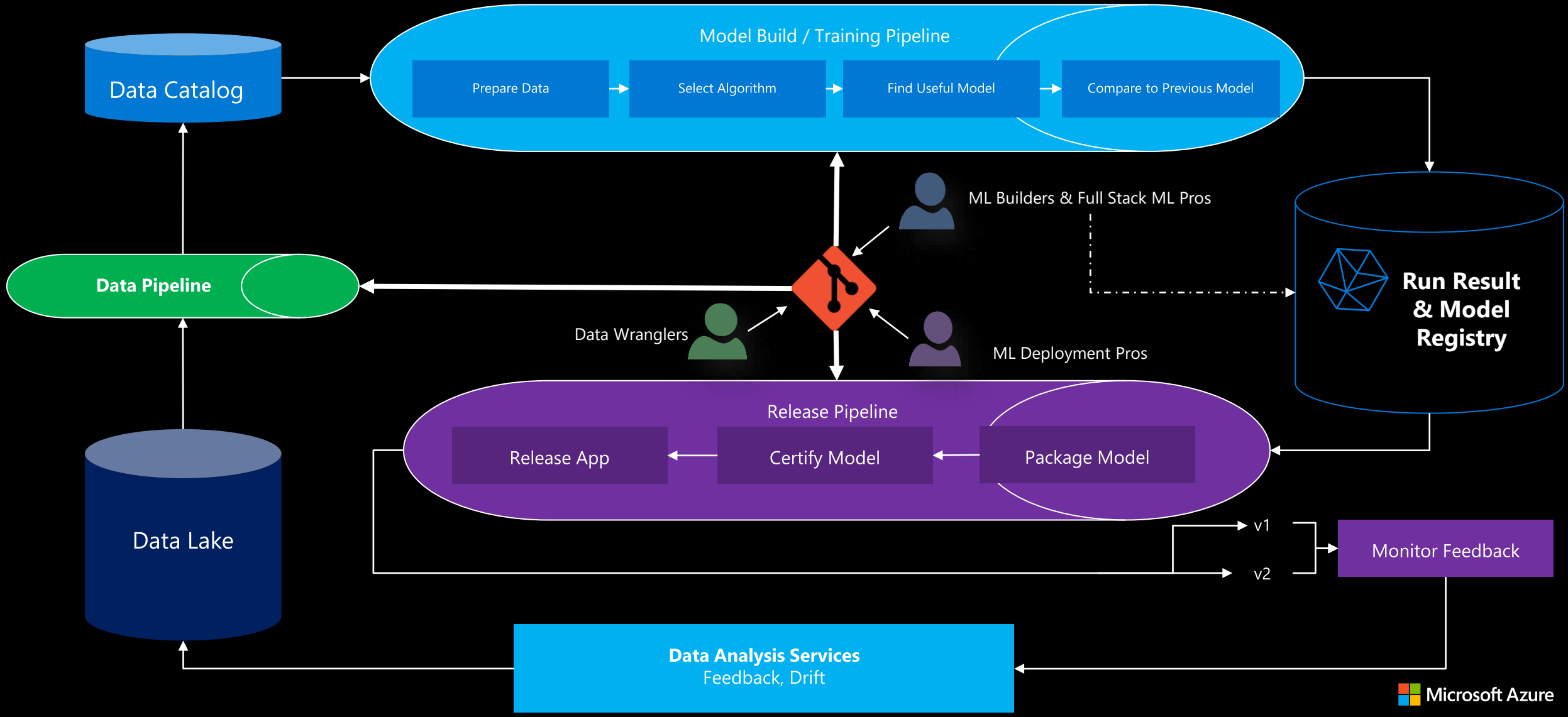


Different types of gates:

- Cost
- Compliance
- Quota

Microsoft Azure

MLOps e2e


Model Build / Training Pipeline
Prepare Data
Select Algorithm
Find Useful Model
Compare to Previous Model
Data Catalog
ML Builders & Full Stack ML Pros
Run Result & Model Registry
Data Pipeline
Data Wranglers
ML Deployment Pros
Release Pipeline
Release App
Certify Model
Package Model
Data Lake
v1
v2
Monitor Feedback
Data Analysis Services
Feedback, Drift
Microsoft Azure

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

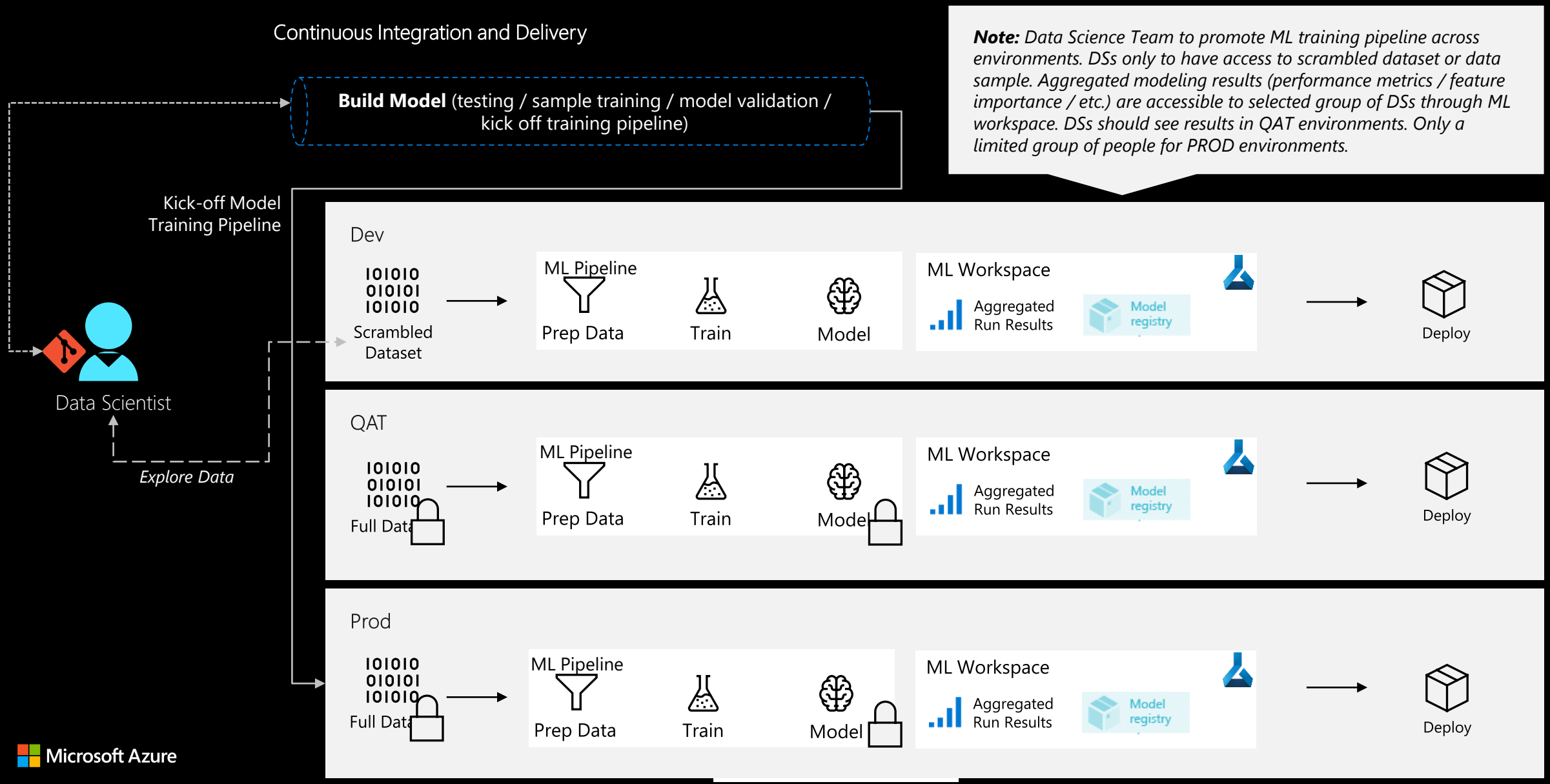
Azure Data Lake
Azure Data Factory
Azure DataBricks
Azure SQL

Azure DevOps
GitHub
Azure Kubernetes Service
Azure IoT Edge
Azure Monitor

ML
Engineer

Microsoft Azure

Model Training – gitops flow



Note: *Data Science Team to promote ML training pipeline across environments. DSs only to have access to scrambled dataset or data sample. Aggregated modeling results (performance metrics / feature importance / etc.) are accessible to selected group of DSs through ML workspace. DSs should see results in QAT environments. Only a limited group of people for PROD environments.*

Promoting Model Release

```yaml
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
```

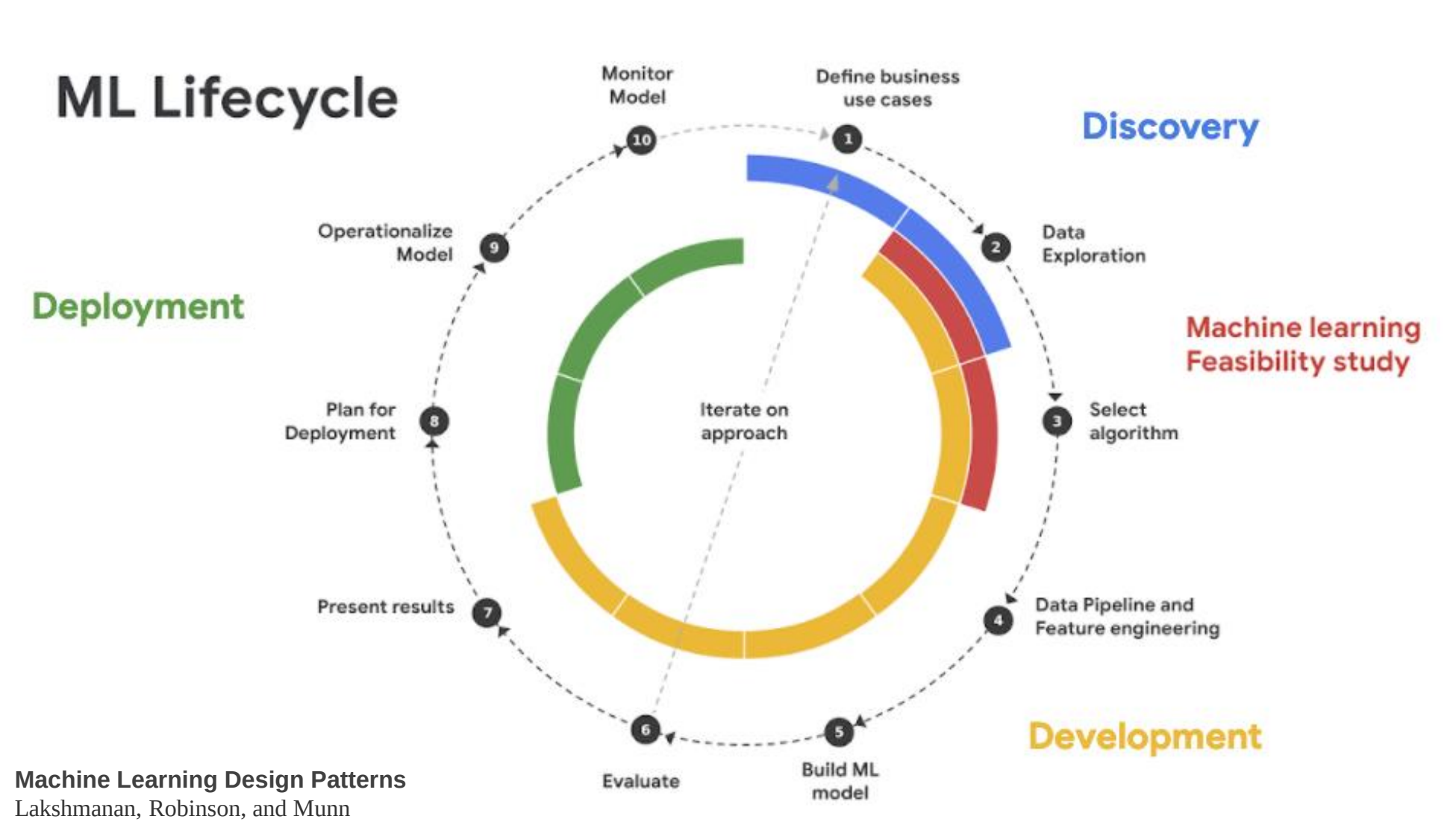
Ramp traffic up in a controlled fashion

GitOps

```yaml
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
```

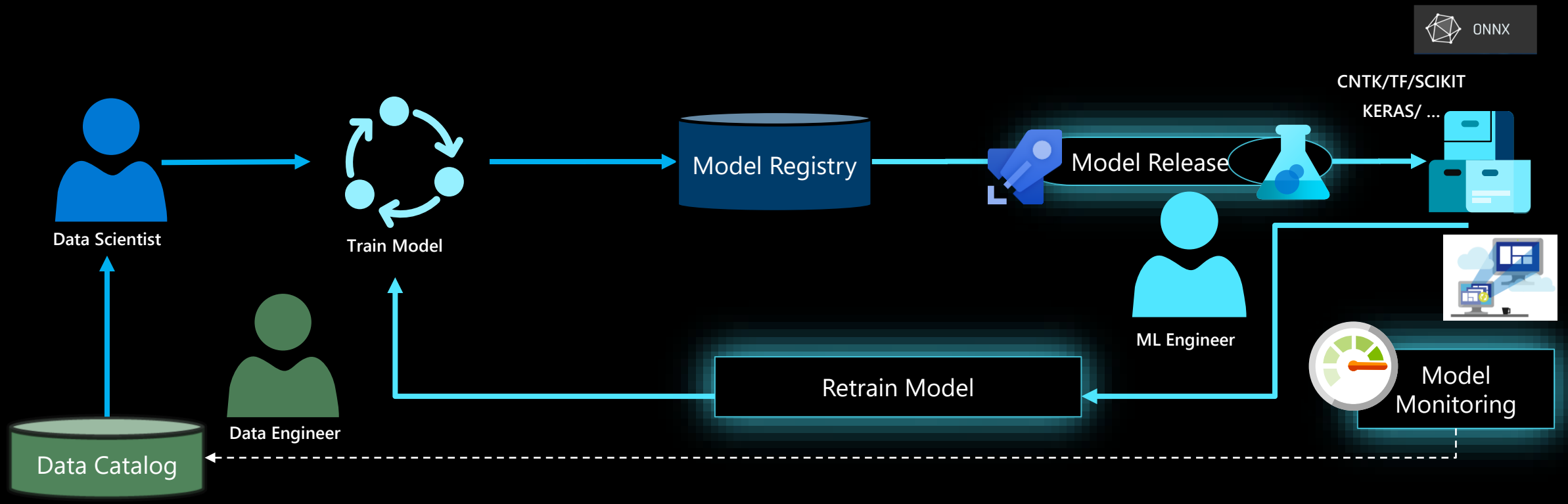
So… how do we implement MLOps in the real world?

Machine Learning Lifecycle



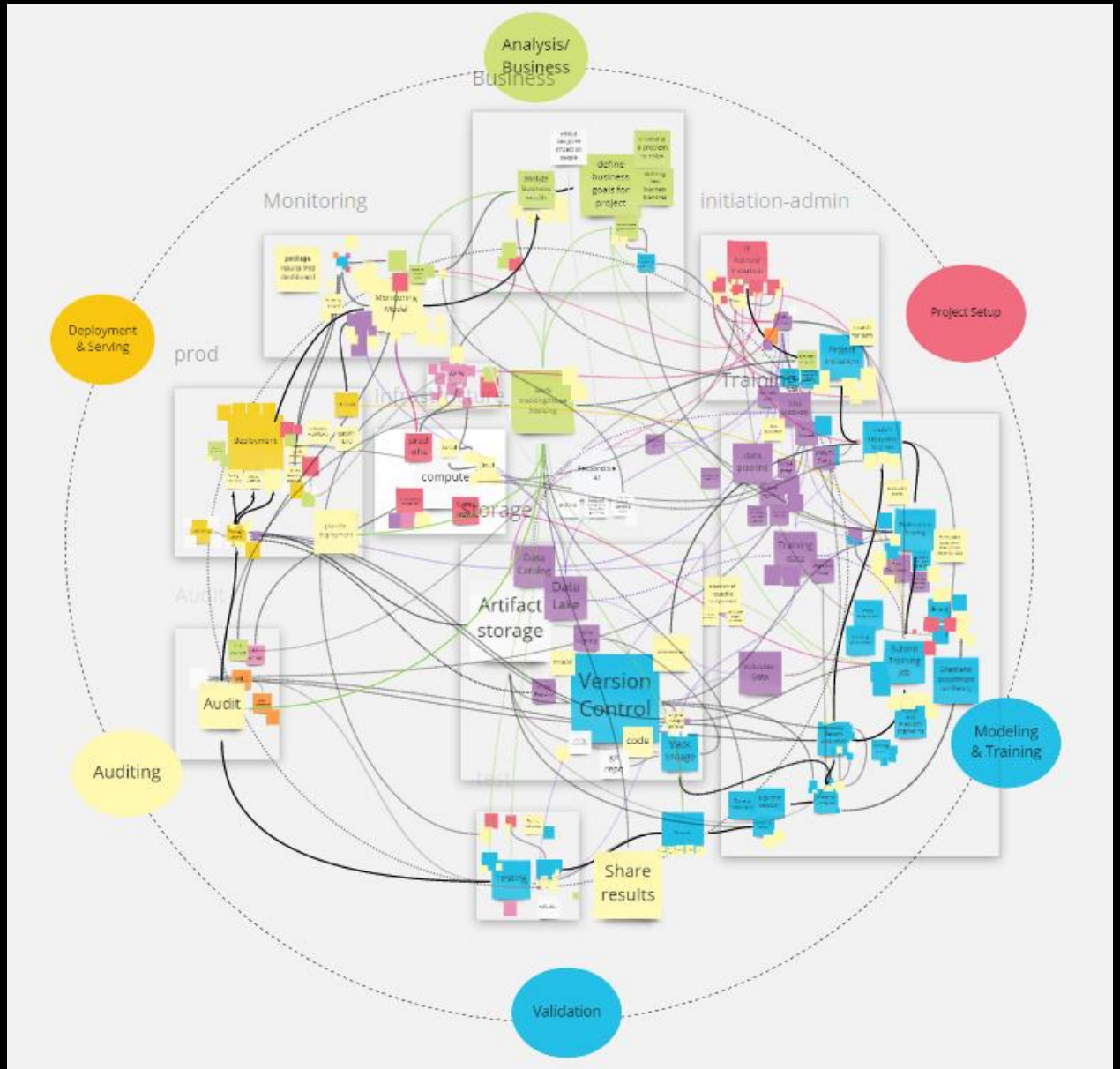
Machine Learning Design Patterns
Lakshmanan, Robinson, and Munn

The ML Lifecycle (Ideal State)



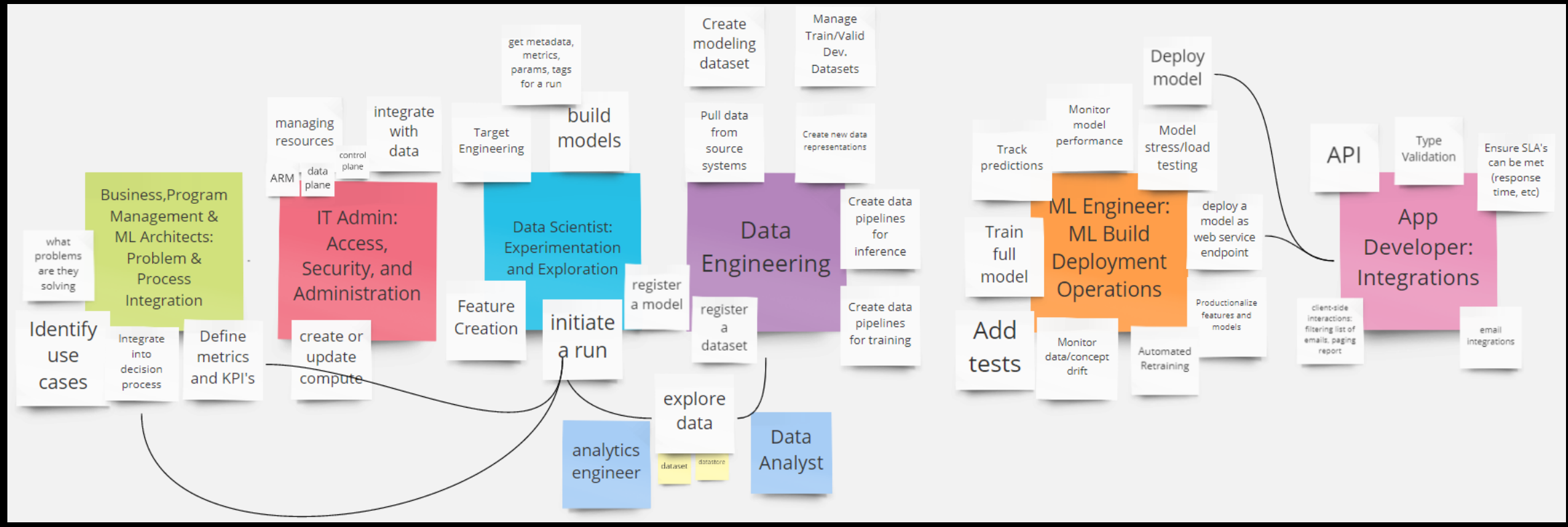
Machine Learning Lifecycle In Reality:

Sociotechnical implications , nonlinearities, dependencies



Microsoft Azure

The Complexities Of Production ML Roles



Microsoft Azure

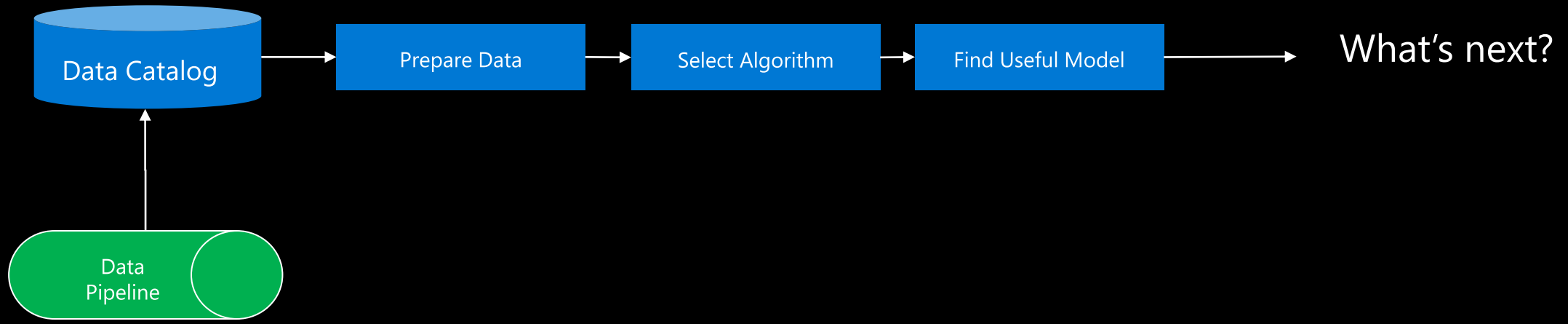
MLOps – Process Maturity Model

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, **retraining set up** 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, **may** need human signoff

Level 1 – No MLOps

Interactive, exploratory, get to something useful.

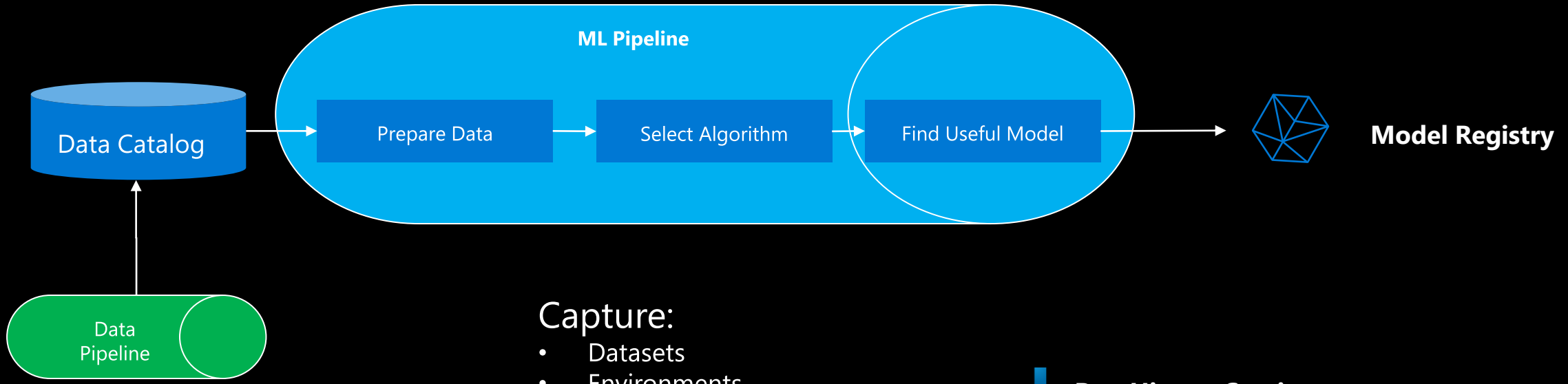
Data Scientist



Level 2 – Reproducible Model Training

Version code, data, ensure model can be recreated.

Data Scientist



Capture:

- Datasets
- Environments
- Code
- Logs / metrics
- Outputs

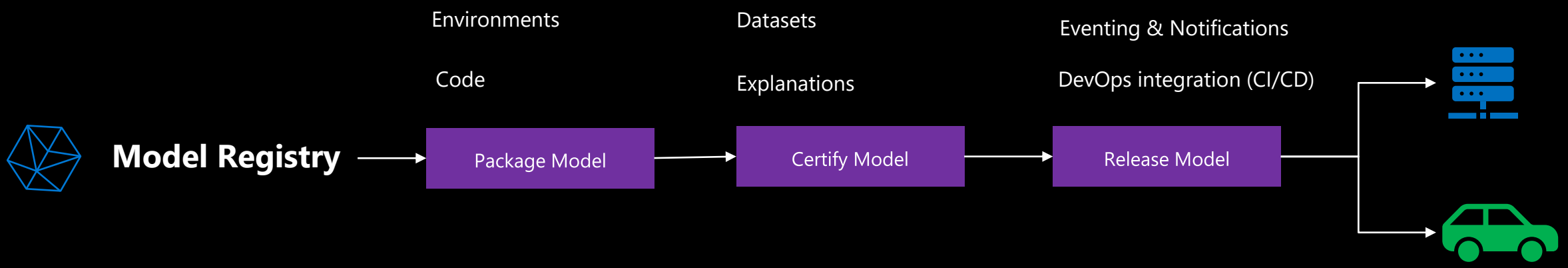
Run History Service

Microsoft Azure

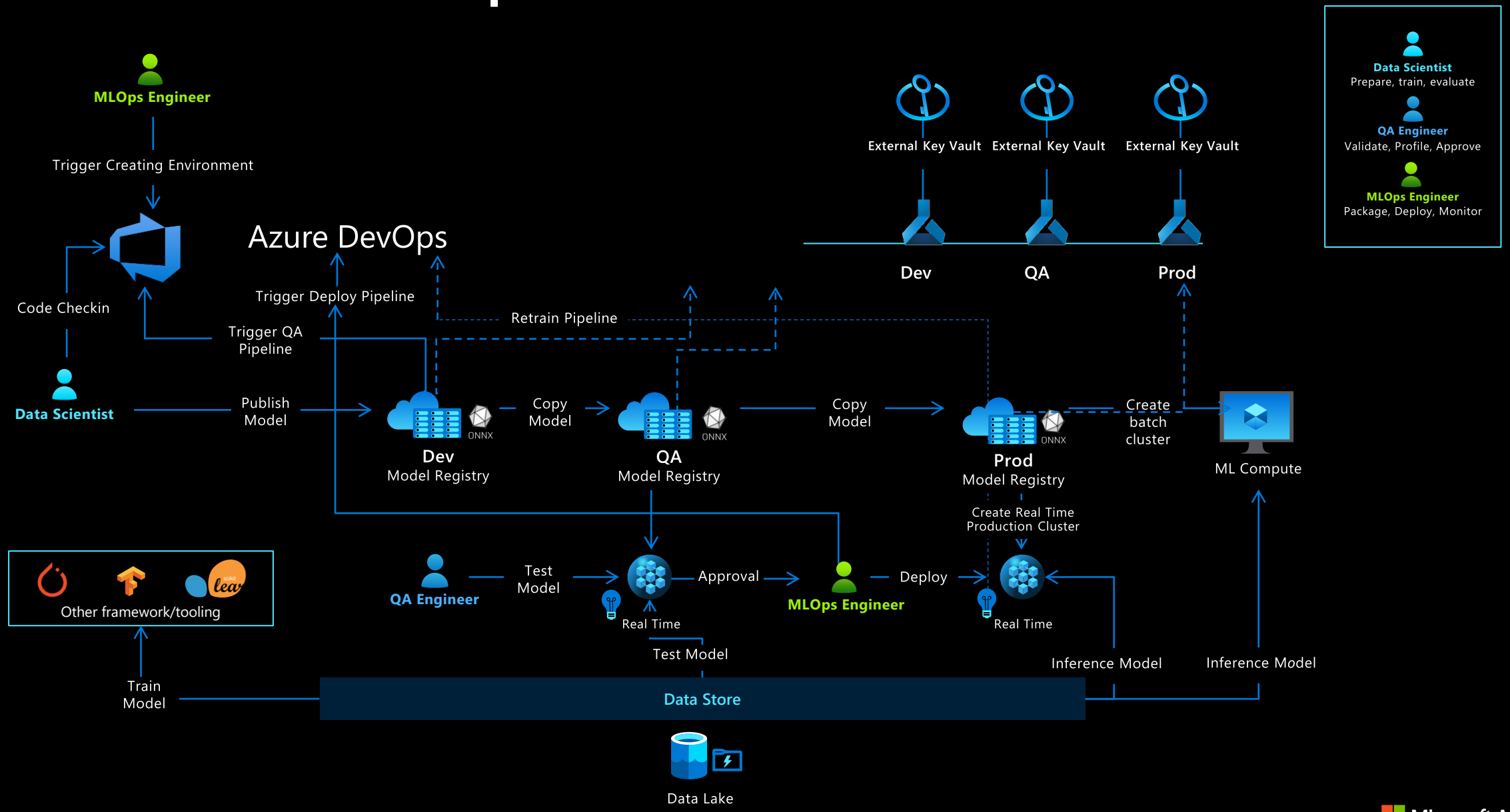
Level 3 – Managed Model Operationalization

Package, certify, deploy

ML Engineer



Level 4 – E2E MLOps



Microsoft Azure

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

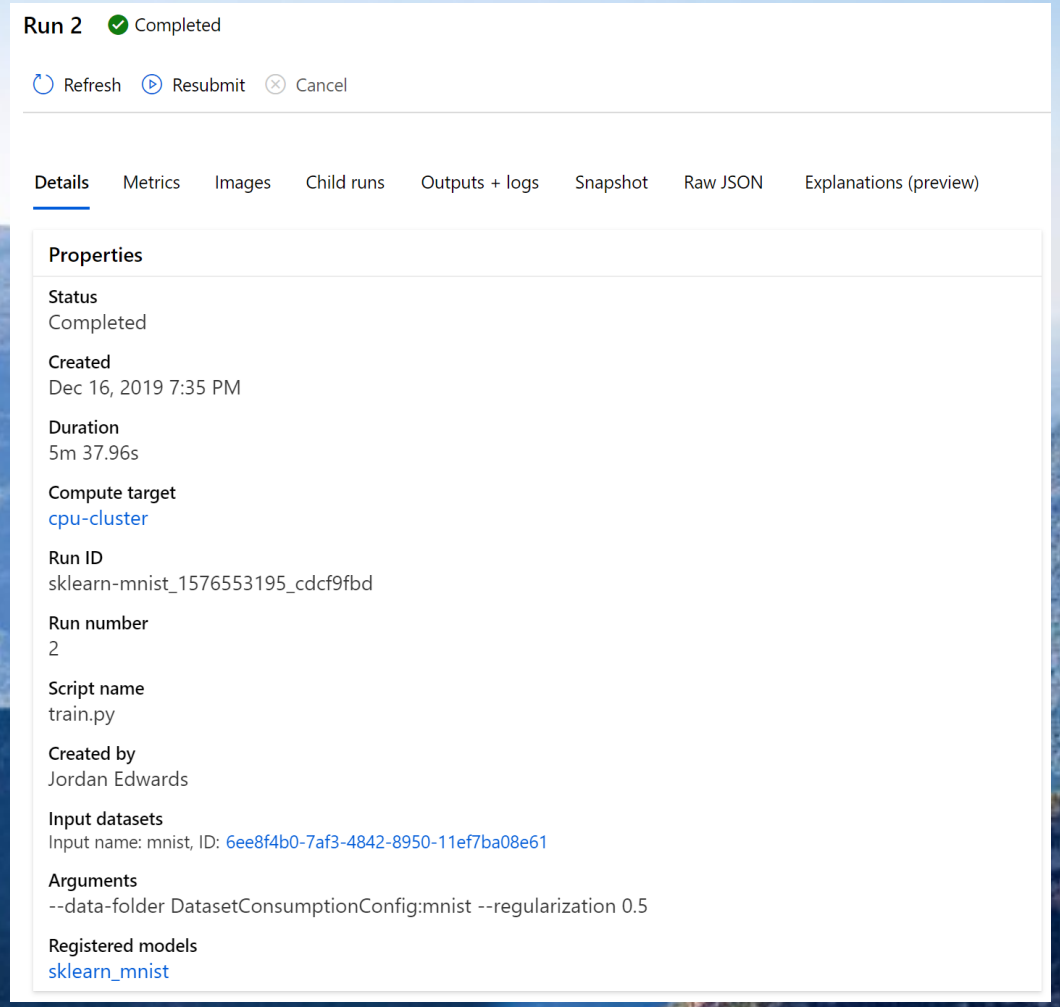
Managed Scoring Endpoints
Online, Batch

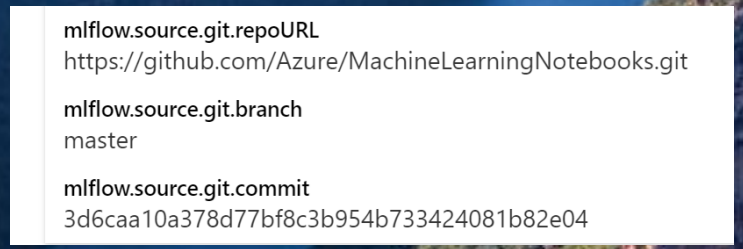
Asset Management
Code, Data, Models, Environments, Metrics

SDK / CLI / API Interfaces

Track everything

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



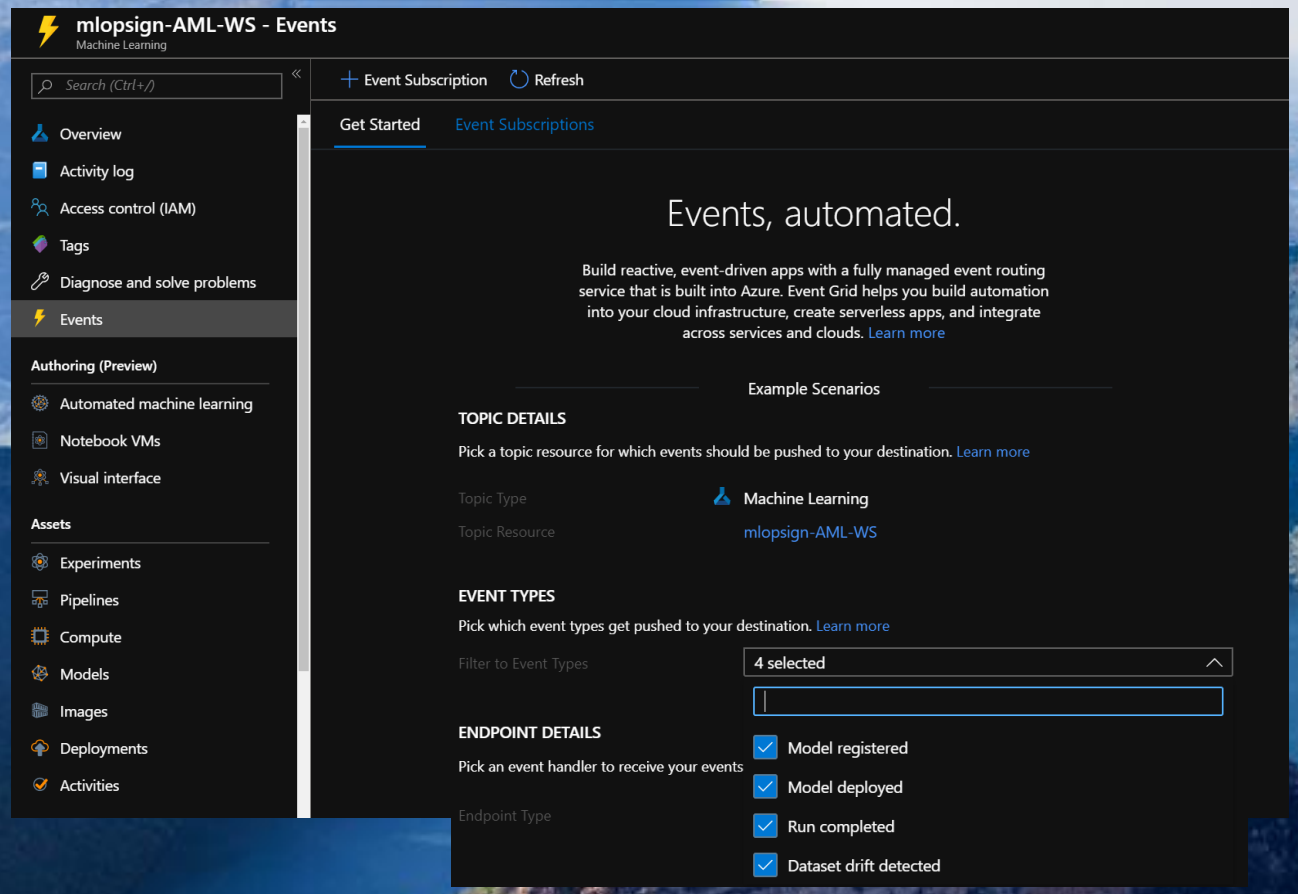


Microsoft Azure

Automation & Eventing

Fully managed event routing for all activities in the ML lifecycle

Let's look at some examples...



Microsoft Azure

Package & deploy anywhere

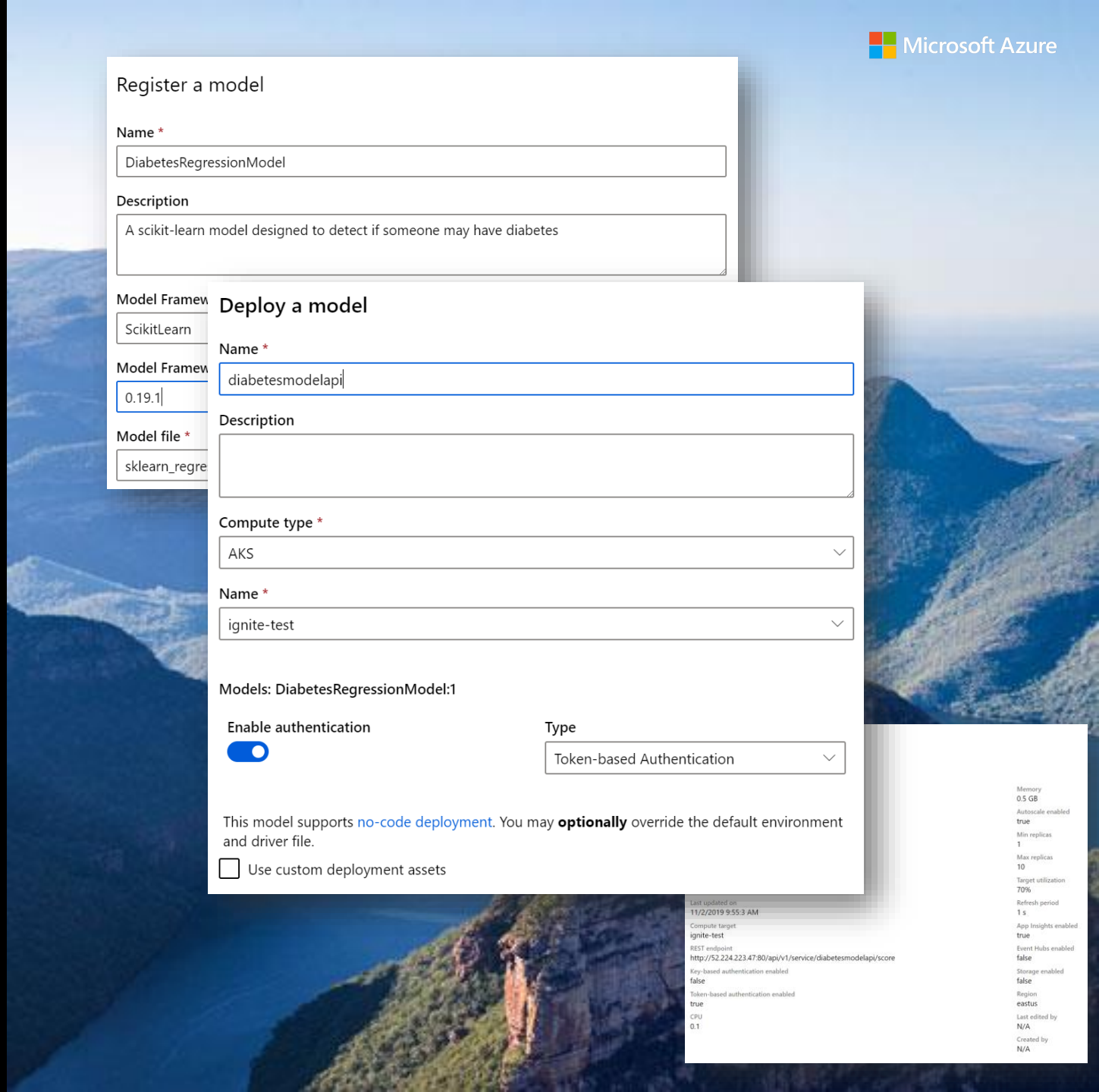
Capture framework / version / resource requirements

Supports no-code deployment

Open and interoperable

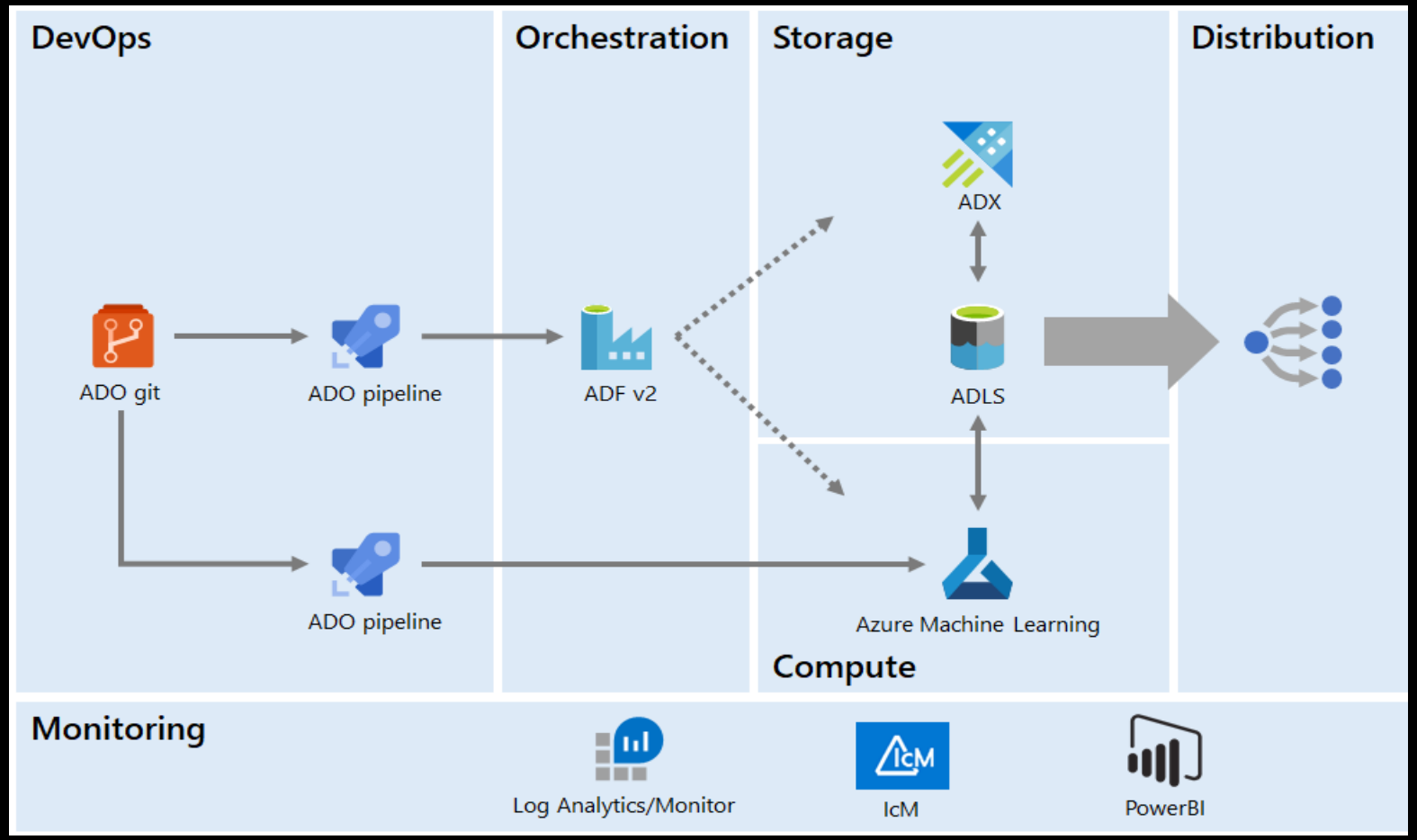
Supported frameworks:

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

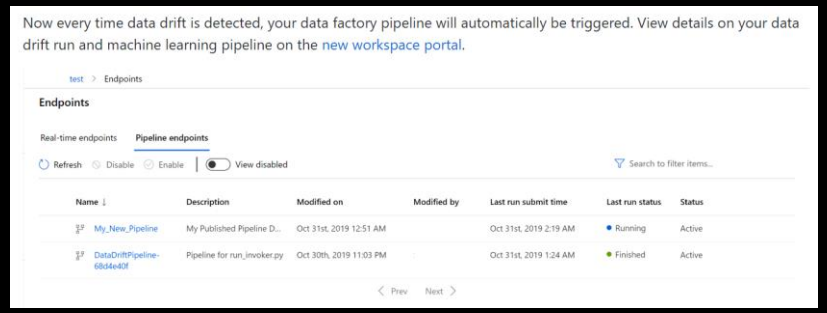


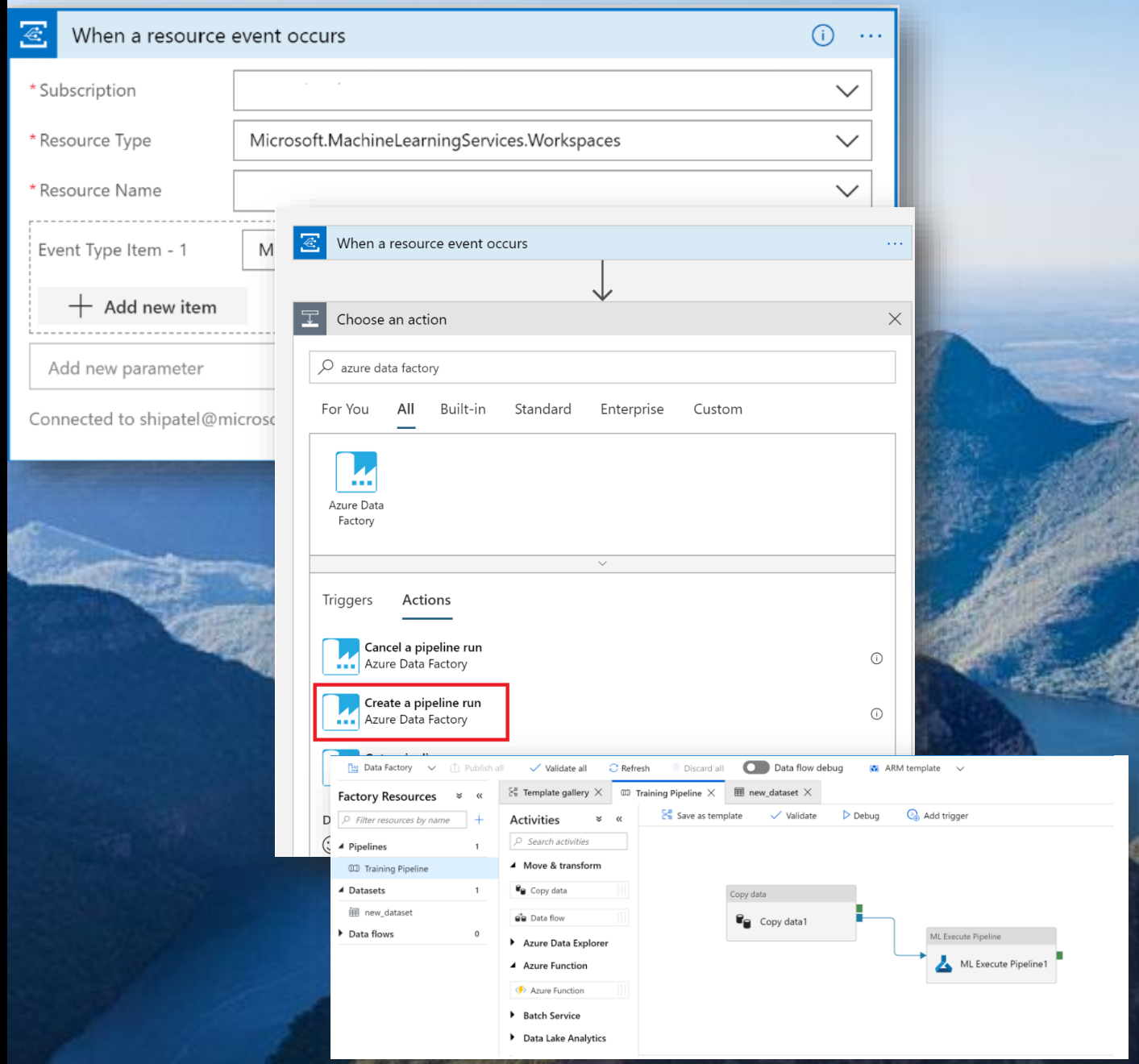
Common Deployment & Orchestration Patterns

Data Factory
DataLake
DevOps Pipeline



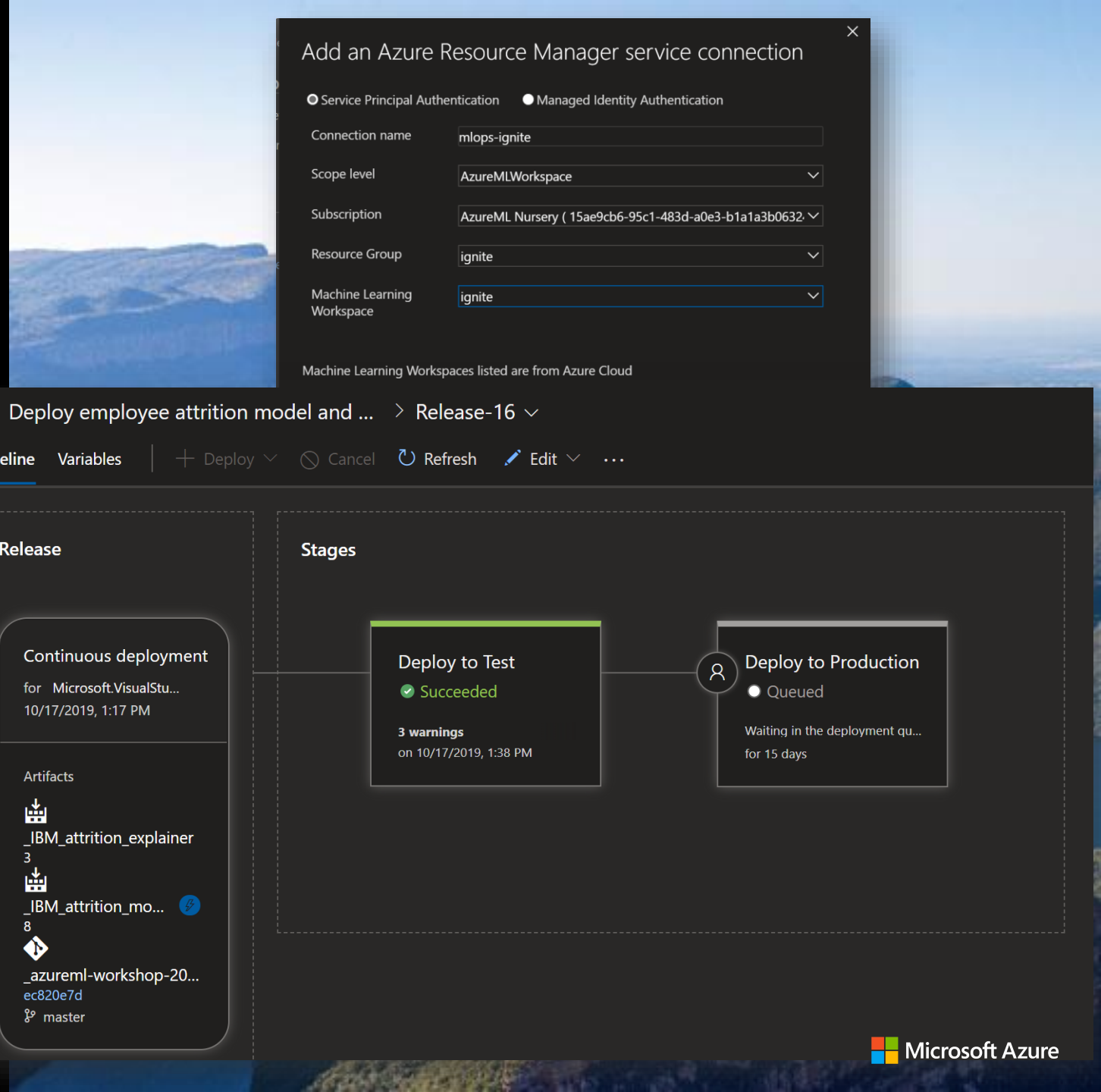
Data Factory integration





DevOps Integration

Automate training & deployment into existing release management processes



Microsoft Azure

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:

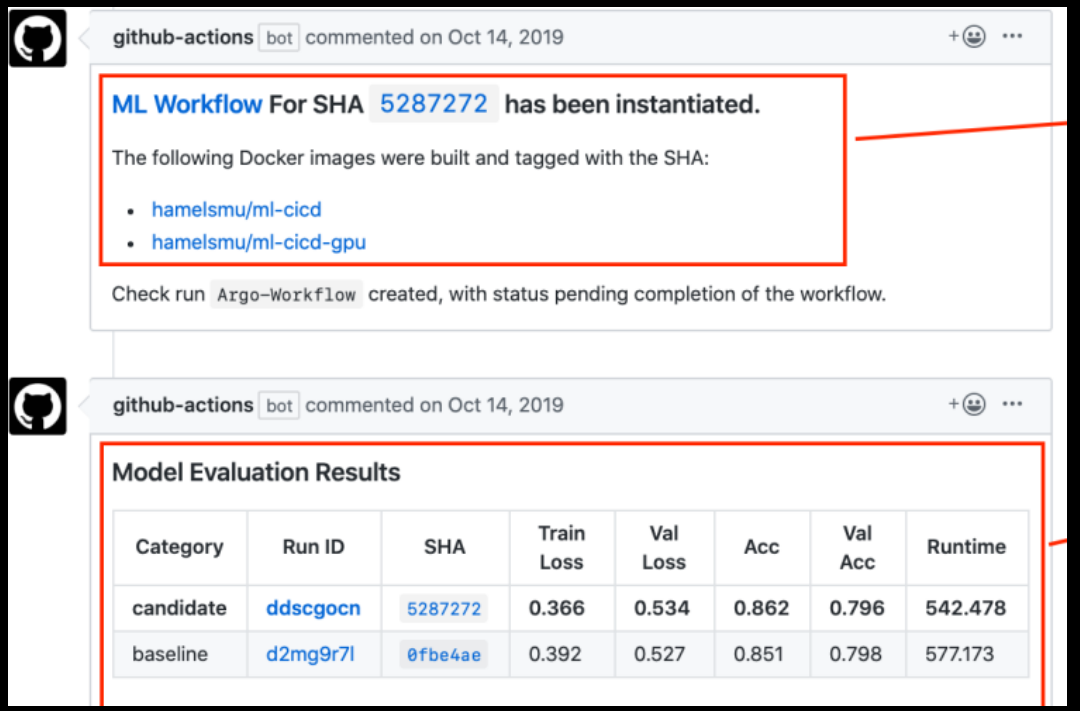
Scenario	Primary persona	Azure offering	OSS offering	Canonical pipe	Strengths
Model orchestration (Machine learning)	Data scientist	Azure Machine Learning Pipelines	Kubeflow Pipelines	Data -> Model	Distribution, caching, code-first, reuse
Data orchestration (Data prep)	Data engineer	Azure Data Factory 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

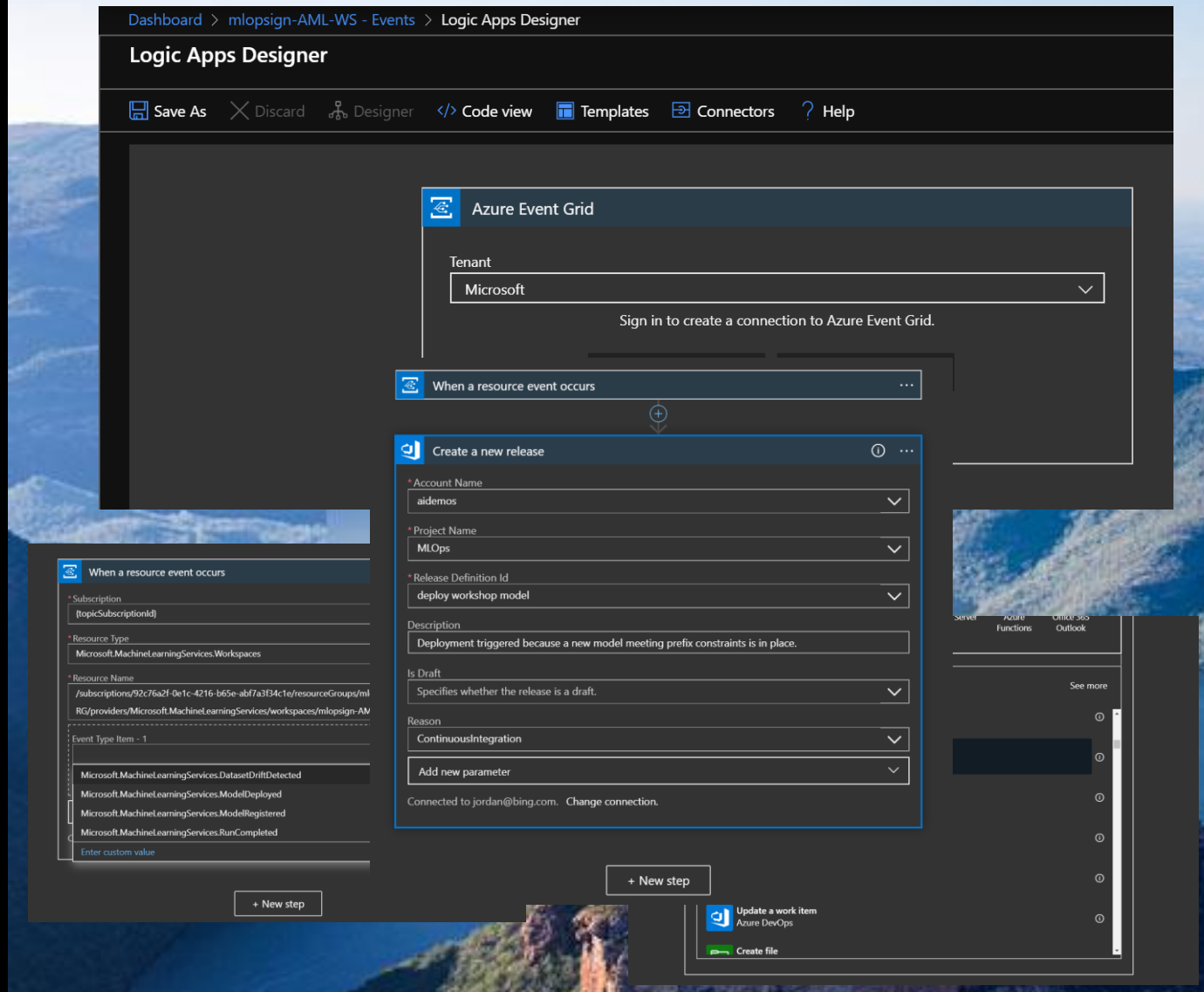
Azure ML will write output directly to GHA and pull requests as desired.

```yaml
- 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
```



Send emails / hit a webhook whenever a run completes

Streamline model reuse & consumption

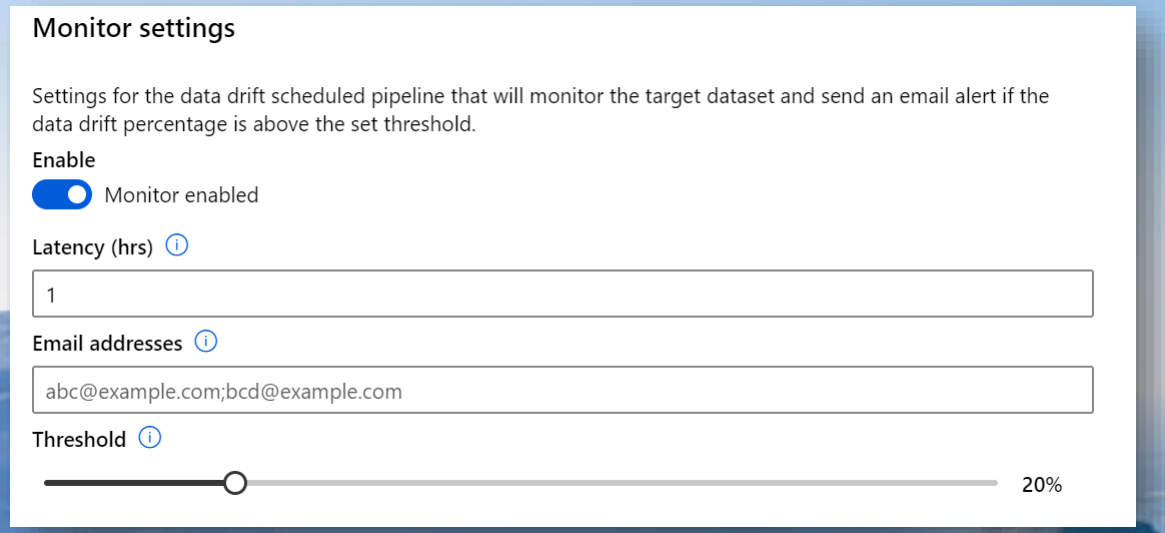


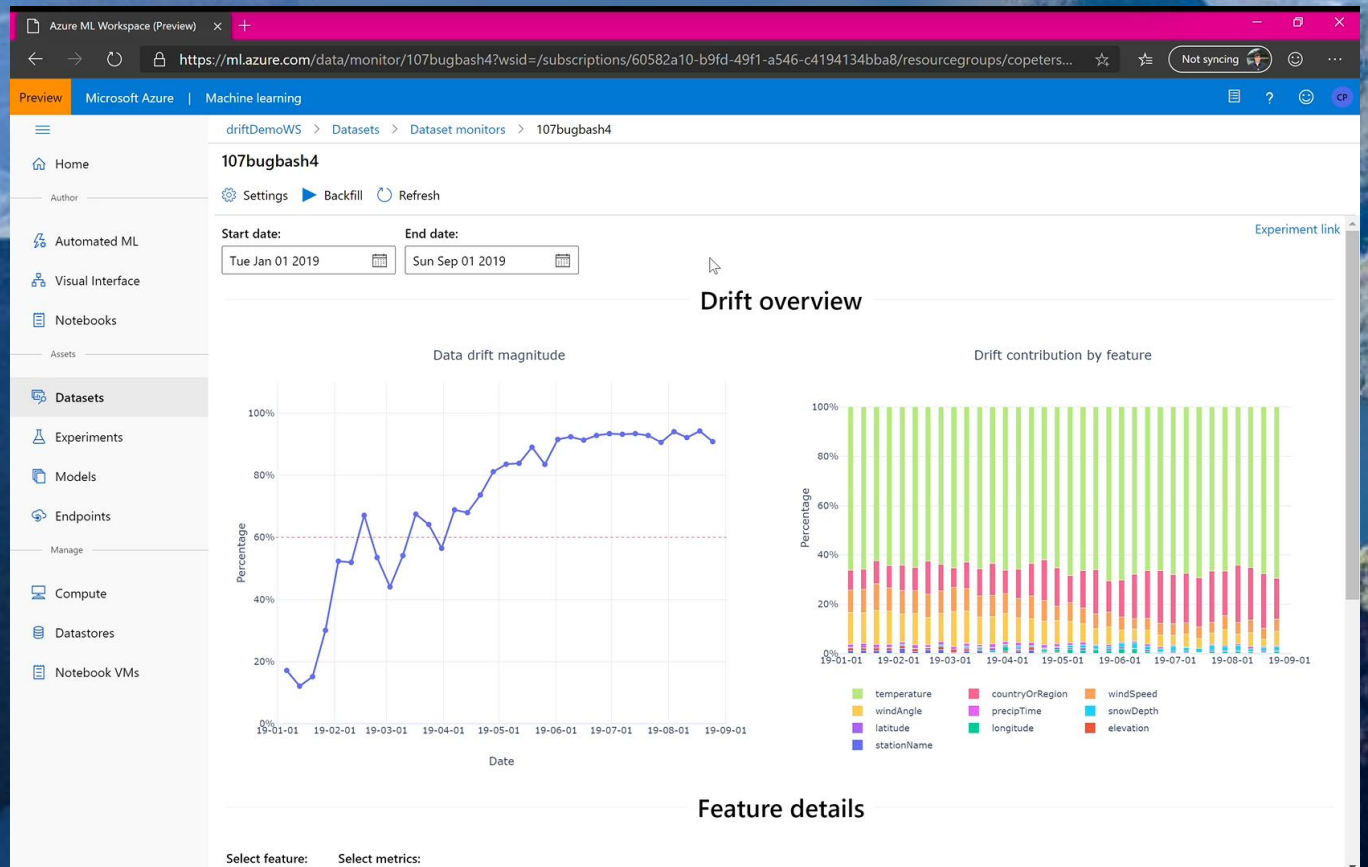
Microsoft Azure

Set up a data drift monitor…

Compare datasets over time

Determine when to take a closer look





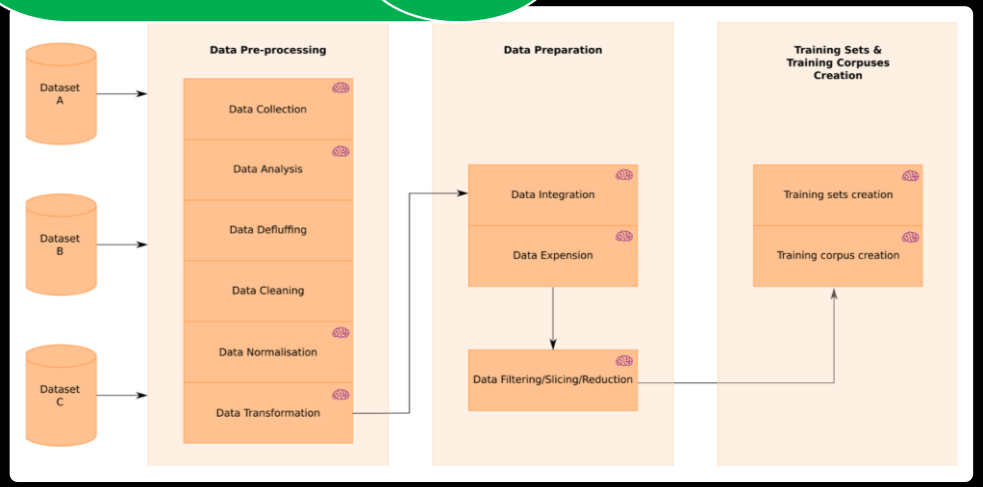
Microsoft Azure

Demo: Azure ML for MLOps

Customer Examples

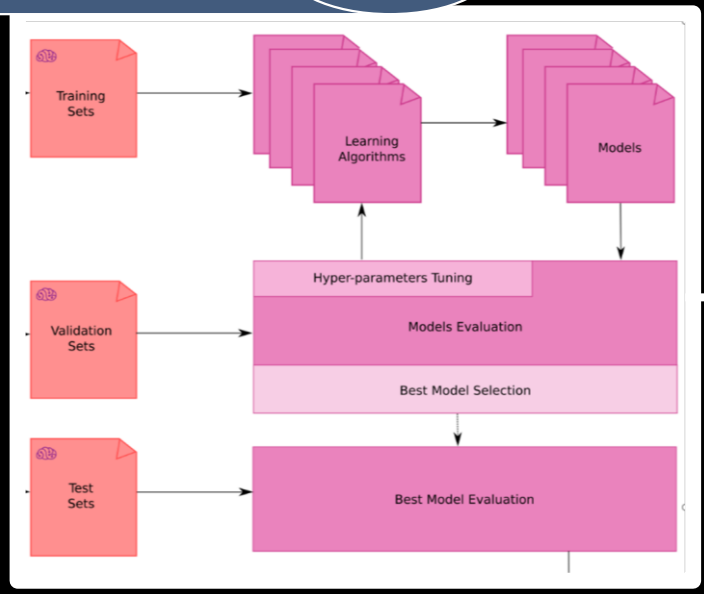
Video Call Prediction for Microsoft Teams

Prepare Data



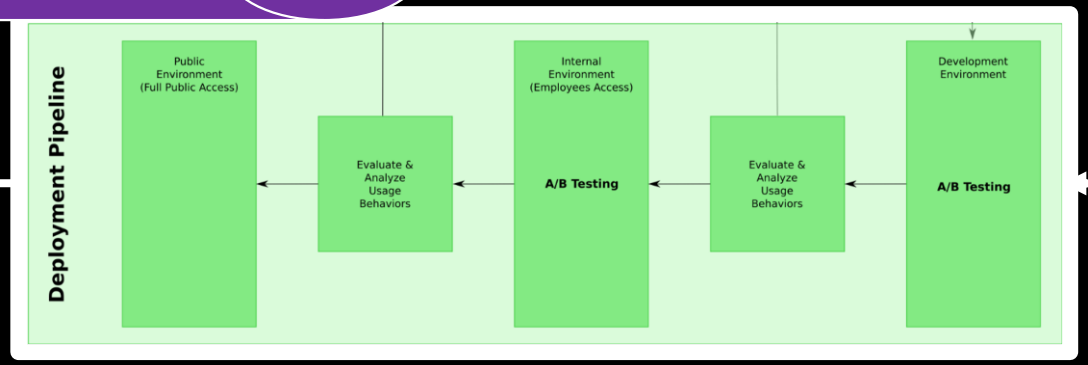
Data Engineer

ML Pipeline



Data Scientist

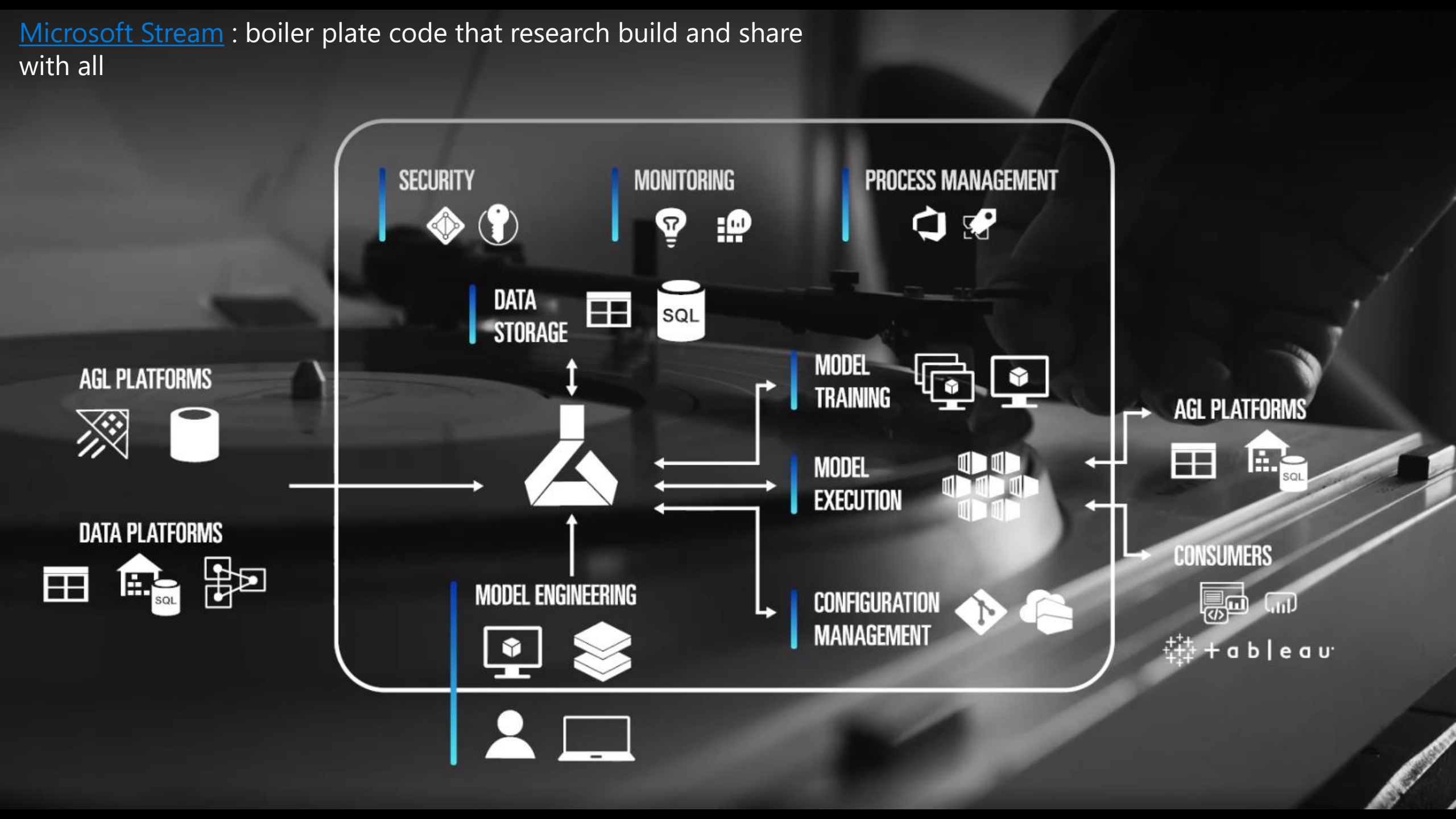
Release Model



ML Engineer

Microsoft Azure

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

ADLSv2

AZURE DATABRICKS FOR BAYESIAN DECISION NETWORK

(TRAINED MODEL)

BAYESIAN DECISION NETWORK ON AZURE MACHINE LEARNING

VALIDATE MODELS

CONTAINERIZE AND TEST ON AZURE CONTAINER INSTANCE

DEPLOY MODELS

DEPLOY CONTAINERS TO AZURE VMs

MONITOR MODELS

MONITOR MODELS WITH AURE ML

(RECOMMENDATIONS)

(PREDIC

(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

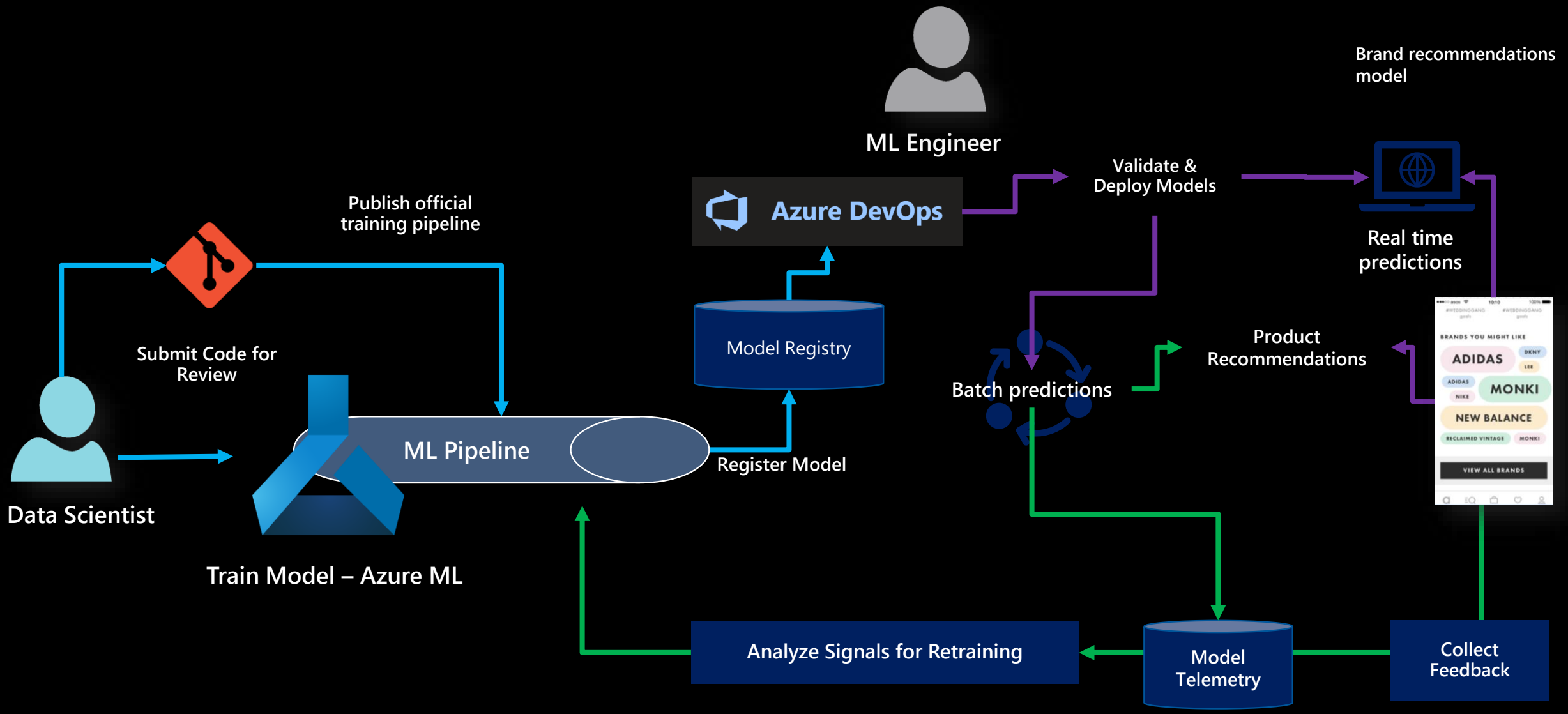
MONITOR MODELS WITH AZURE M

Microsoft Stream

or MLOps pipeline and that is actually doing

Microsoft Azure

Leveraging MLOps to ship recommender systems.



https://github.com/Microsoft/Recommenders

Microsoft Azure

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, **retraining set up** 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, **may** need human signoff

Key Takeaways

Better together: ML + DevOps mindset

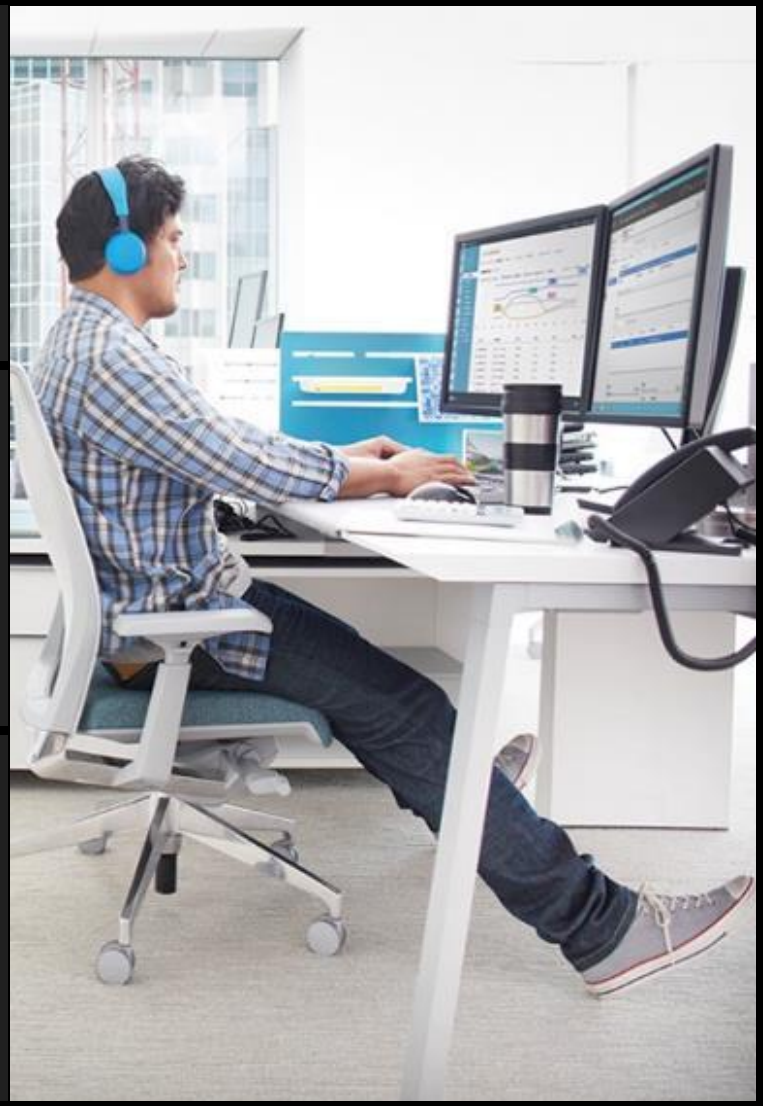
MLOps provides structure for building, deploying and managing and an enterprise-ready AI application lifecycle

MLOps enhances delivery

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

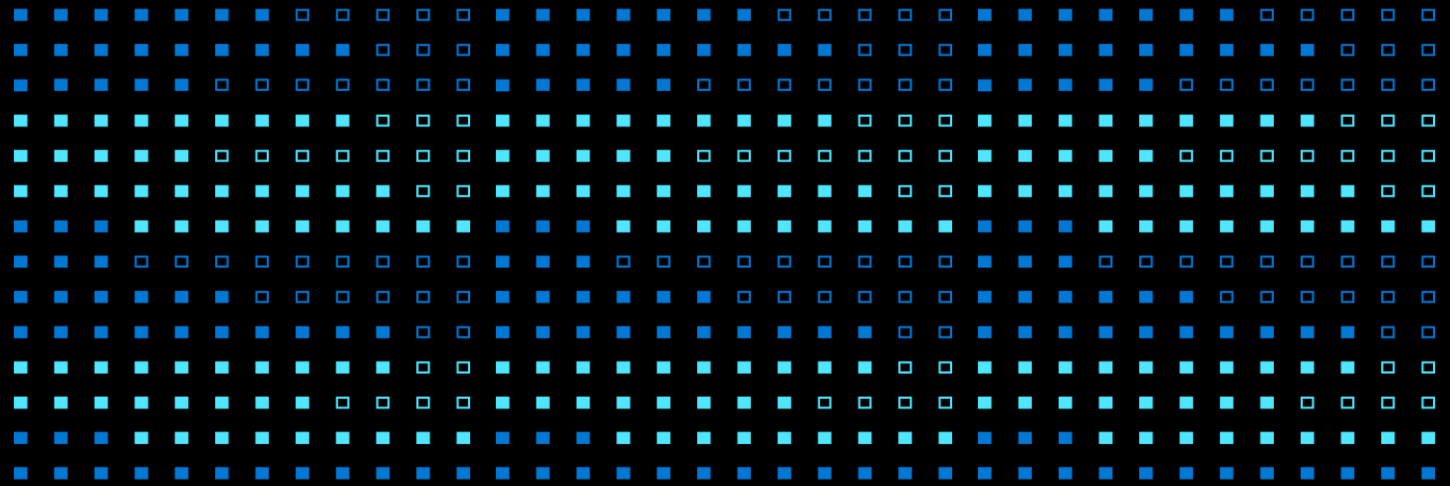
More than technology

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.



Microsoft Azure

Microsoft Azure

Thank you for joining us.