Microsoft

DevOps for ML & AI

What is AI Operationalization?

FOR THE BUSINESS ITS:

- It's about streamlining all the required capabilities in an AI practice, including tools and processes for successful delivery of **AI driven business outcomes**. .
- It's about successfully **adopting change** in order to drive organizational success and outcome AI proved to offer

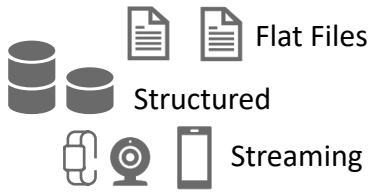
FOR THE IT ITS:

- It's about addressing **technical complexities** while integrating the AI/ML models in real-world solutions
- It's about integrating a non deterministic "learners" associated with AI with traditional **Solution Lifecycle and Service Management** principles

AI Operationalization Framework (Suggested)

Business Understanding

Collection, Generation, Ingestion



- Discovery & Data Catalog
- Ingestion Transforms
- Credential Management
- Privacy & Compliance (GDPR)
- Data Labelling & Simulation

Explore, Transform, Schematize

- Sharing & Collaboration
- Data Pipeline Management
- Versioning Data Transforms
- Batch Transform Scheduling

Model Building & Tuning

- DS IP/ Model Catalog
- Model Build Versioning
  - Featurization, Hyperparameters, Algorithms
  - Training Validation Datasets
  - Accuracy (P/R/F1, Scoring)
  - Auto ML Configs
- Collab. workflow & Reviews

Deployment & Monitoring

- Resiliency (SLAs)
- CI/CD, Deployment WF
- Performance (Inference)
- Monitoring Drift
- KPI Visualization
- Notifications / Triggers

Actionable Insights, Value Realization

Governance (AI Strategy, Policies, Security, Blueprints)

ACM (AI Readiness & Maturity Assessment, AI Adoption & Change Planning)

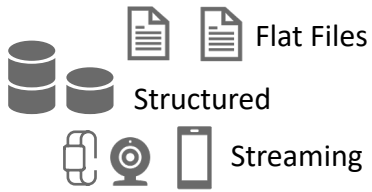
Service Management Strategy (Ops Processes Definitions, Roles & Responsibilities)

- AI Operationalization Resource Kit

ML Workflow

Business Understanding

Collection, Generation, Ingestion



- Discovery & Data Catalog
- Ingestion Transforms
- Credential Management
- Privacy & Compliance (GDPR)
- Data Labelling & Simulation

Explore, Transform, Schematize

- Sharing & Collaboration
- Data Pipeline Management
- Versioning Data Transforms
- Batch Transform Scheduling

Model Building & Tuning

- DS IP/ Model Catalog
- Model Build Versioning
  - Featurization, Hyperparameters, Algorithms
  - Training Validation Datasets
  - Accuracy (P/R/F1, Scoring)
  - Auto ML Configs
- Collab. workflow & Reviews

Deployment & Monitoring

- Resiliency (SLAs)
- CI/CD, Deployment WF
- Performance (Inference)
- Monitoring Drift
- KPI Visualization
- Notifications / Triggers

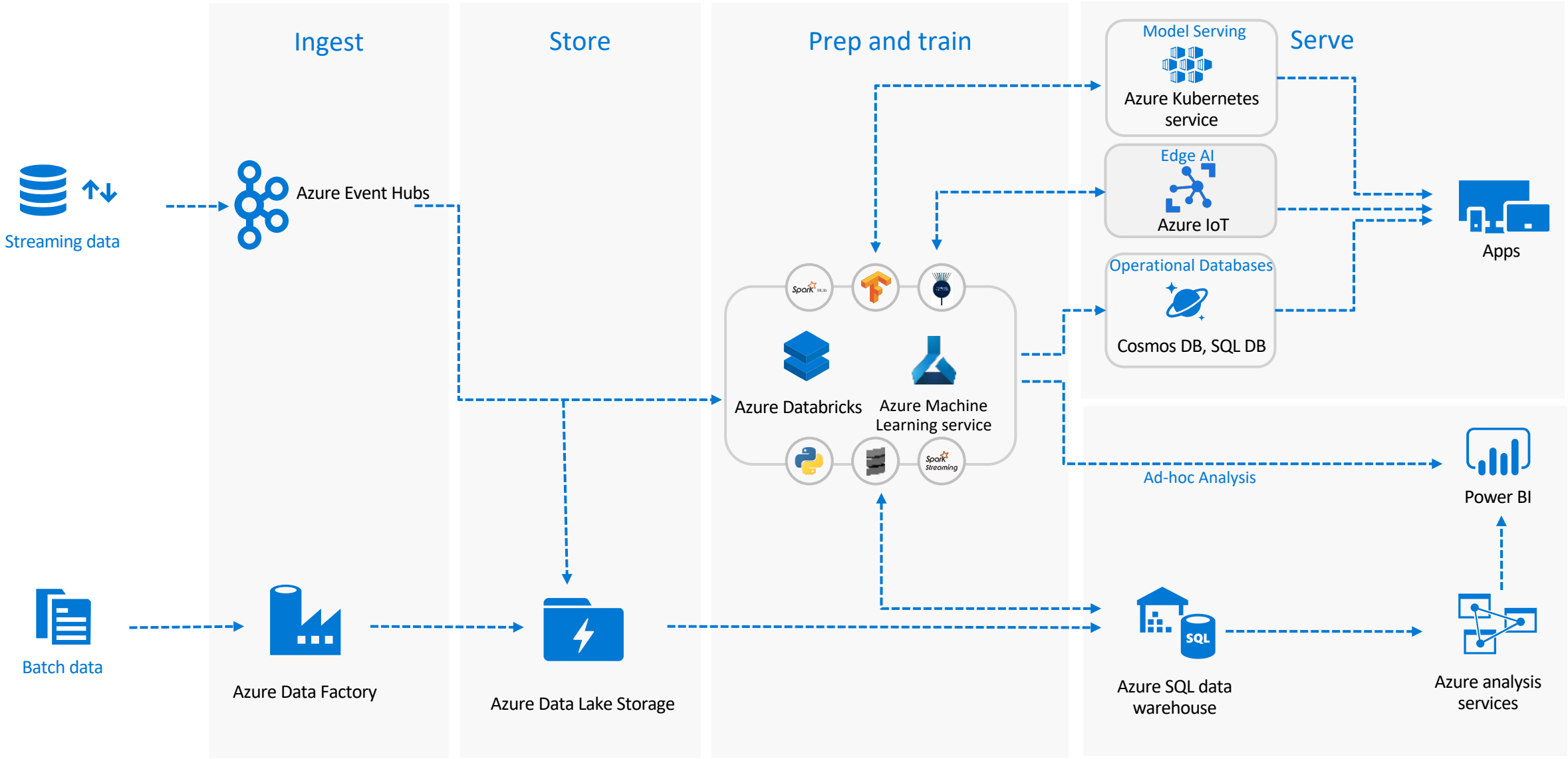
Actionable Insights, Value Realization

Governance (AI Strategy, Policies, Security, Blueprints)

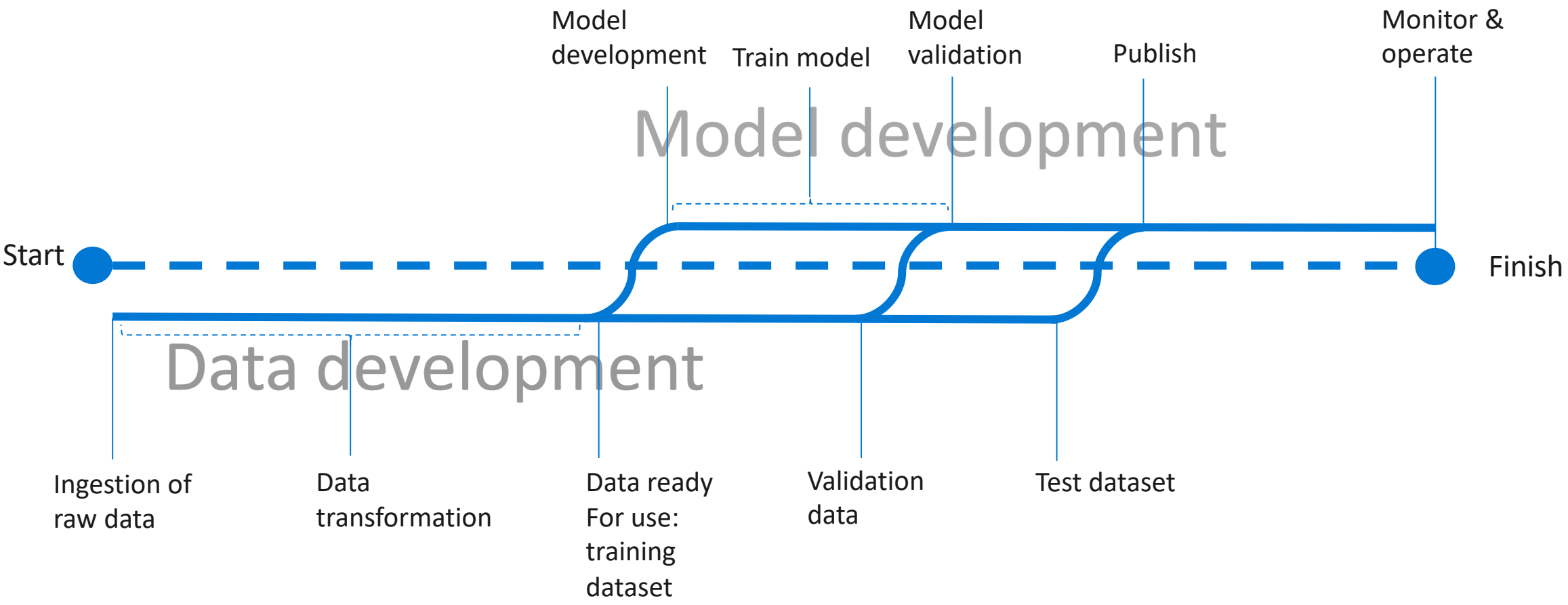
ACM (AI Readiness & Maturity Assessment, AI Adoption & Change Planning)

Service Management Strategy (Ops Processes Definitions, Roles & Responsibilities)

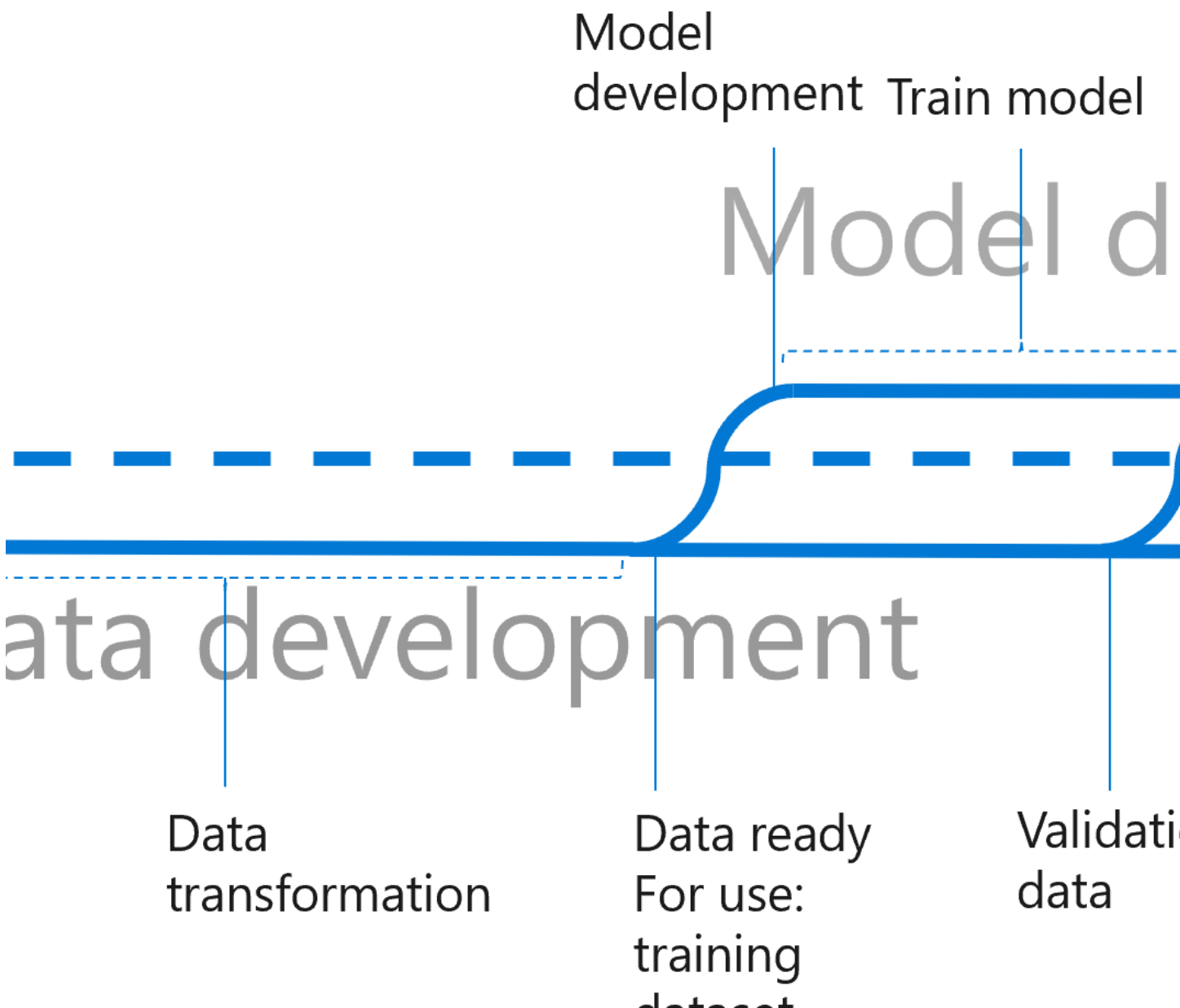
ML Workflow with Azure services



How to avoid the pitfalls: digital audit trail



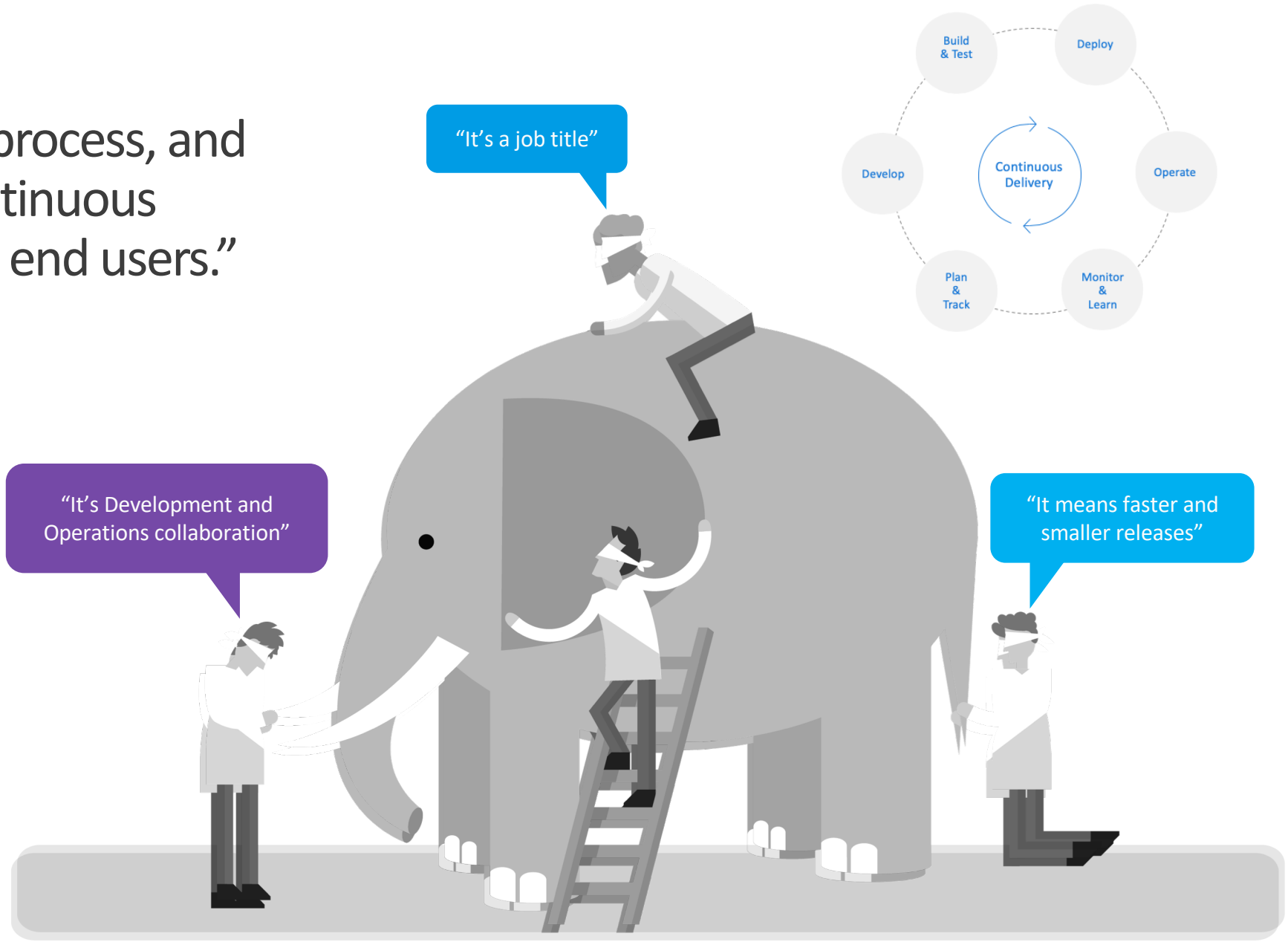
Introduction: DevOps for A.I.



What is DevOps?

"The union of people, process, and products to enable continuous delivery of value to our end users."

Donovan Brown, MSFT Sr PM



Introducing DevOps for A.I.

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

Infusing AI into this lifecycle brings new challenges and changes.

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)

DevOps

Code reproducibility

Code testing

App deployment

MLOps

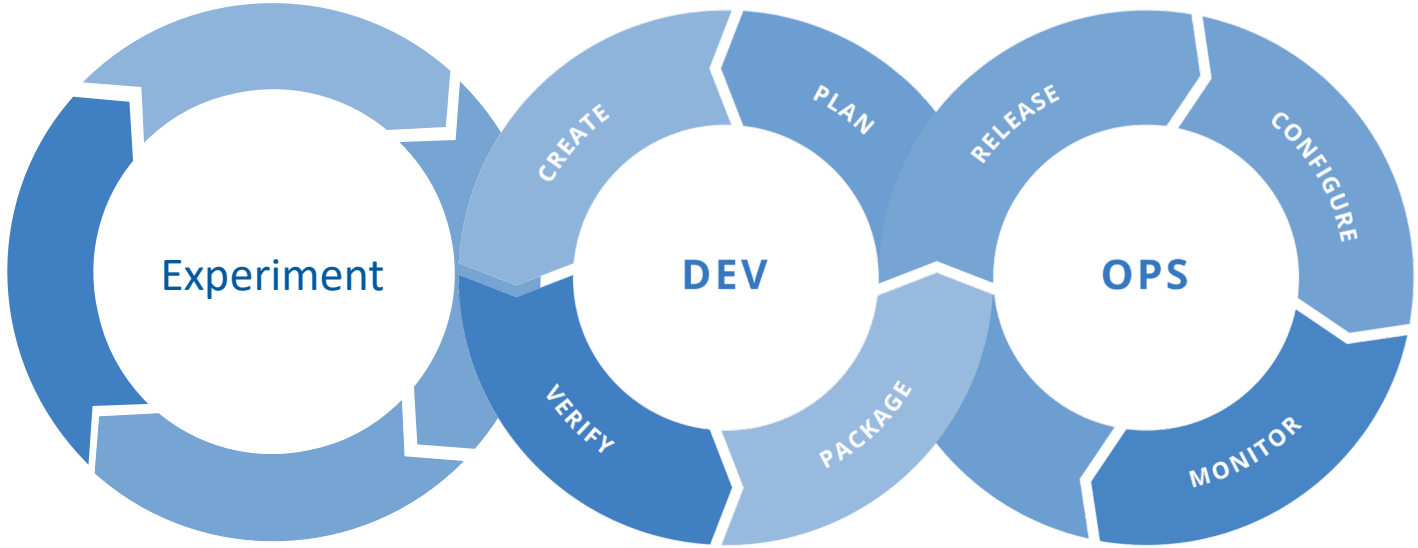
Model reproducibility

Model validation

Model deployment

Model retraining

AI DevOps lifecycle



Experiment

Data Acquisition
Business Understanding
Initial Modeling

Develop

Modeling + Testing
Continuous Integration
Continuous Deployment

Operate

Continuous Delivery
Data Feedback Loop
System + Model Monitoring

Why adopt DevOps for AI?

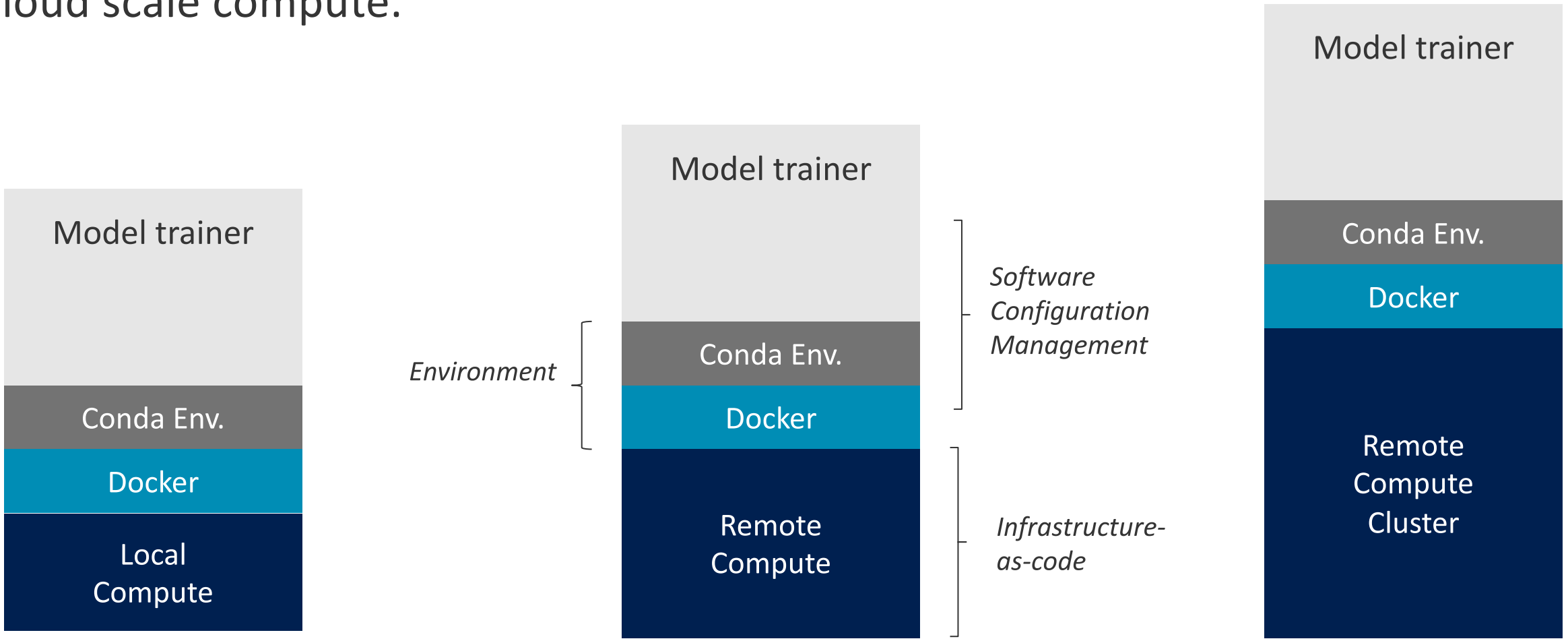
Overcome that data science teams only own experiments, instead of being responsible for the end-to-end flow from experiment to production to operational support on AI.

Benefits:

- Continuous delivery of value (data insights, models) to end users.
- End-to-end ownership of the Analytics Lifecycle by DS teams
- Enforcing a consistent approach to building and deploying AI
- Extending data science with SDE practices to increase delivery quality and cadence.
- Providing a framework for continuous learning, lineage, auditability and regulatory compliance.
- Improving team collaboration through standardization in delivery practices.

Infrastructure as Code in Data Science

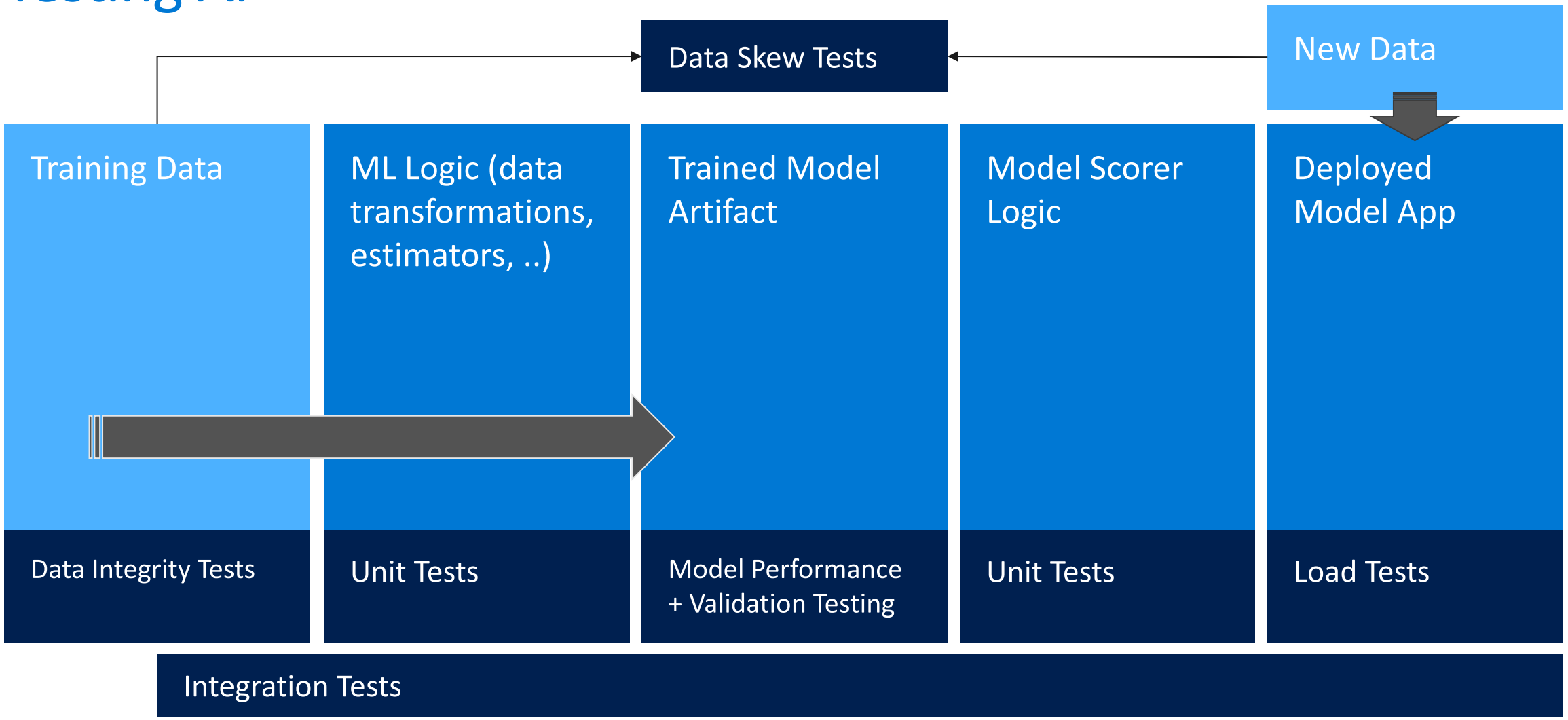
Elevating SCM and IaC practices to scale a local modeling experiment to cloud scale compute.



AI reported issue types

Edge cases	• The model response on a given record is not the expected one. Normally we need to investigate the trainset and detect potential bias. We also need to ensure that the preprocessing is not clipping any values etc. It's a good practice to document these corner cases and even add them in the daily testing to ensure future models don't have the same issue
Null values / unknown categories	• This type of bugs refers to the resiliency of the model in case of missing values and how well can it handle unseen categorical values.
Input issues	• An input stream may stop producing data causing unexpected responses by the model.

Testing AI


Data Skew Tests
New Data
Training Data
ML Logic (data transformations, estimators, ..)
Trained Model Artifact
Model Scorer Logic
Deployed Model App
Data Integrity Tests
Unit Tests
Model Performance + Validation Testing
Unit Tests
Load Tests
Integration Tests

Test Types

Unit tests - Individual components of the codebase are tested separately. In a data science project imaging each transformation on the data set has its own unit test.

Data tests - Tests to validate that the input schema complies with the data dictionary that was created during the data acquisition and data understanding phase. Imagine if you start the modeling phase and all of a sudden the new incoming data change the type of a column from numeric to string. Having the proper data tests will allow you to understand faster that it's not the model code that is failing but rather a change in the schema of the input data.

Model validation tests - Ensures that the responses of the model are in an expected range. For example if the model is returning probabilities, you should ensure that the output is between 0 and 1 as 1.2 would not make any sense.

Model performance tests – Checks the performance of a trained model using set statistical KPIs (e.g. Root Mean Squared Error). One could assert a minimal required level of performance, or compare multiple trained model versions as part of an A/B test.

Test Types

Integration tests - All individual software modules are combined and tested as a group. In the data science context this includes the ingestion process, the scoring modules and perhaps even the integration with the overall digital experience we are aiming for.

Data skew tests - Tests that the distributions of each feature in the incoming data set as similar to what was documented in the data acquisition and understanding phase. If the distribution changes over time, then perhaps this is an indication that the model may need to retrain.

Load tests - Ensures that the model can respond in parallel to multiple requests and within time. Usually load testing is used to do capacity planning and scale up/out the solution to meet the customer's target non functional requirements (e.g. response within 500ms for 300 concurrent users).

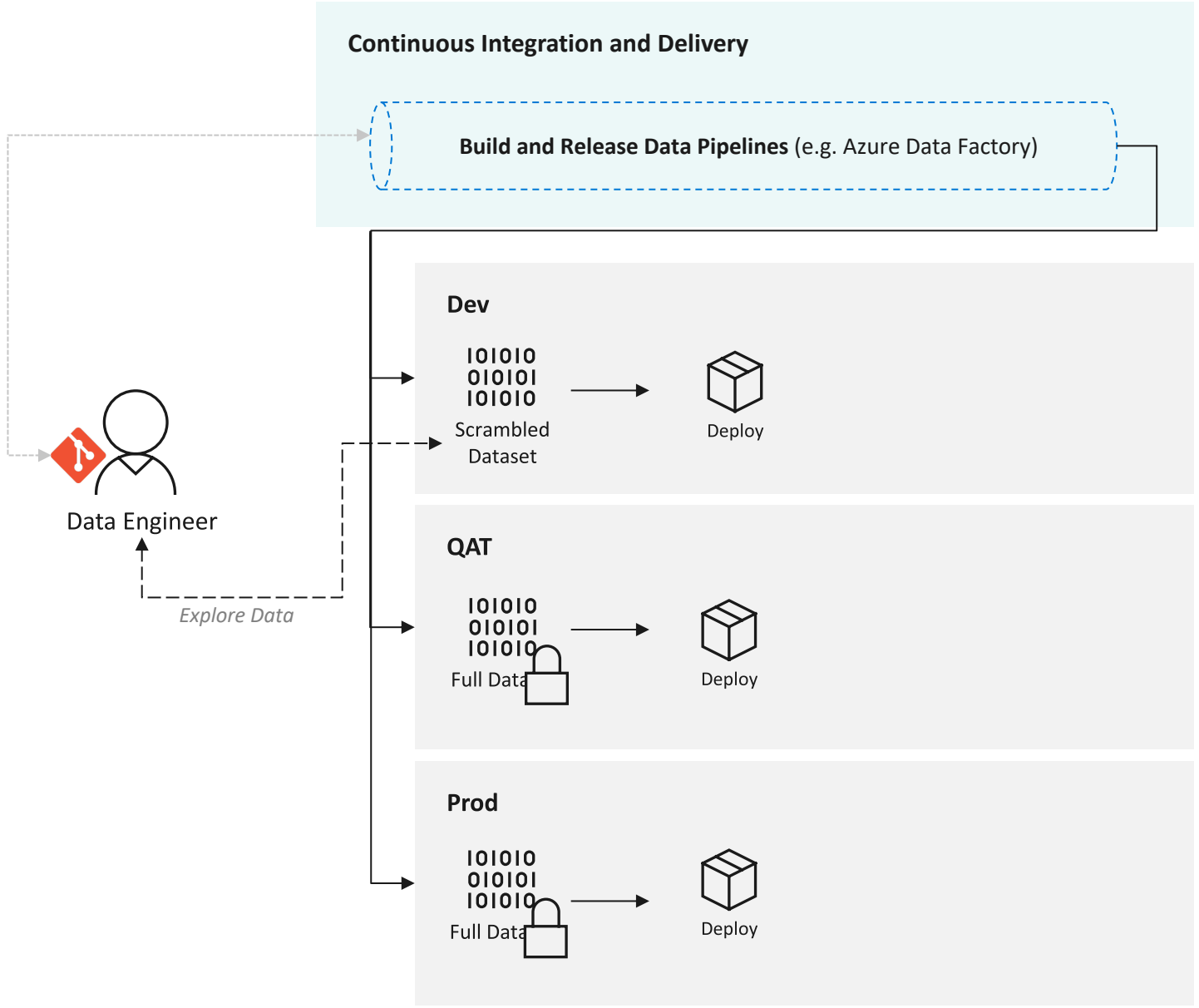
Test Types

Reponsibilities across roles in the context of *model development*.

Test Type	TDSP Phase	Data Scientist	Data Engineer*
Unit Tests	Modeling	X	
Data Integrity Tests	Data Acquisition / Modeling / Deployment	X	
Model Performance	Modeling	X	
Model Validation	Modeling	X	
Integration Tests	Modeling / Deployment	X	X
Load Tests	Deployment		X
Data Monitoring	Deployment		X
Skew Monitoring	Deployment		X
Model Monitoring	Deployment	X	X

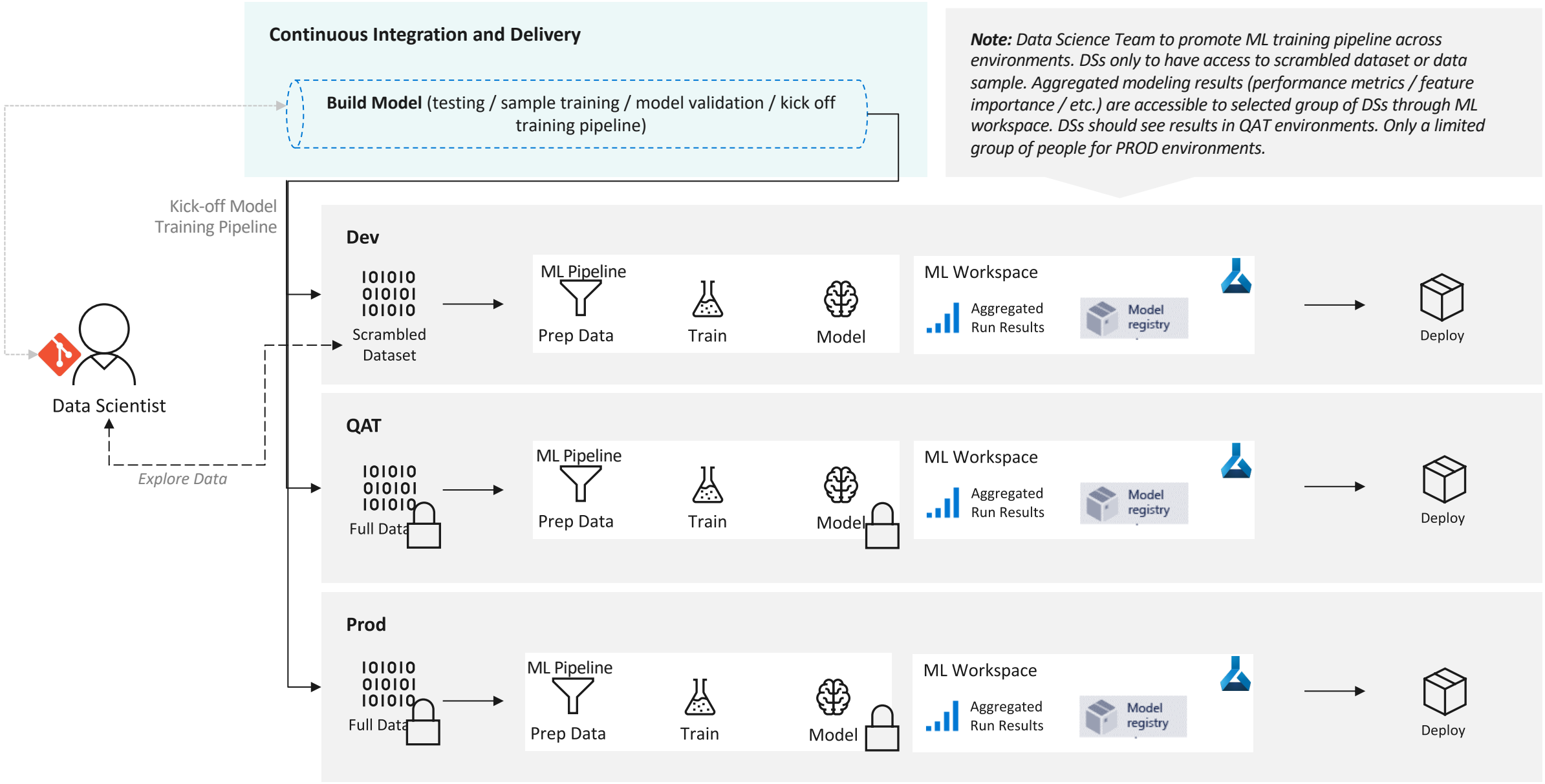
*Note the additional testing reponsibilities for the DE in the context of data pipeline development.

Promoting Data Pipelines across Environments



Note: *Data Engineers to work with sample data to develop data pipelines. A process of continouous integration and delivery will help to build and release, thus promote, pipelines accross environments to transform production data. A very selected group of engineers are assumed to have access to production monitoring systems.*

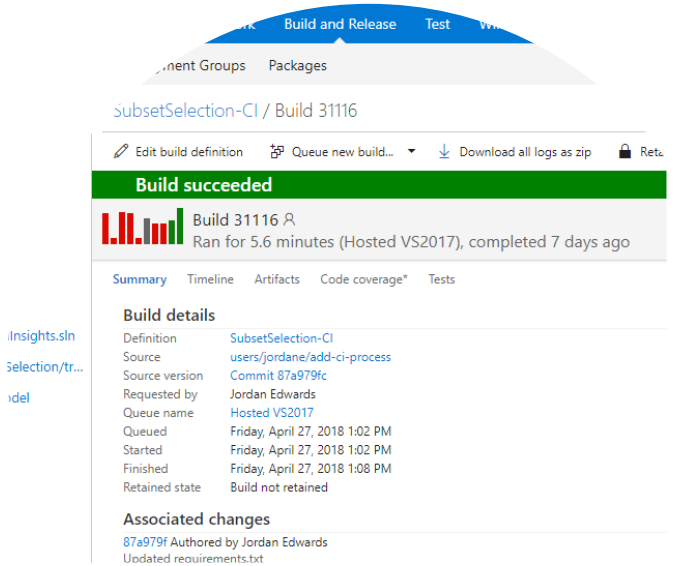
Promoting Model Training across Environments

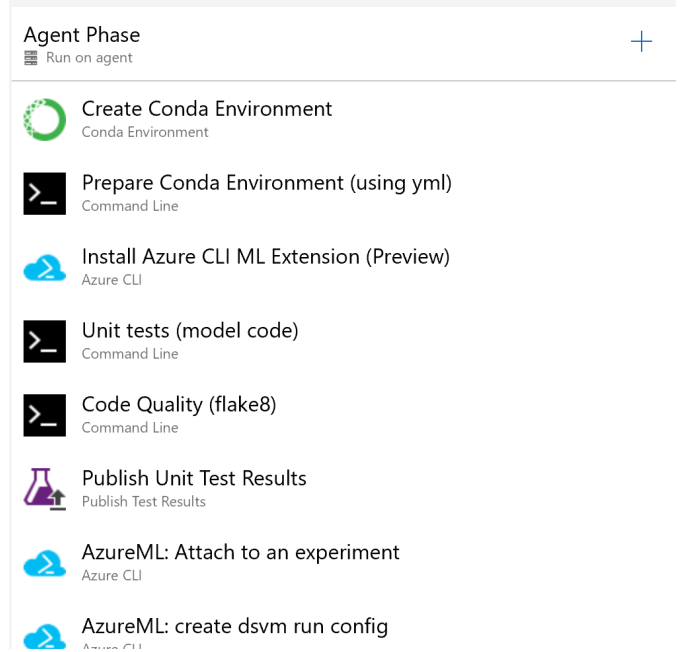


Model CI/CD

The ability to continuously integrate, automatically test, build and deploy Machine Learning artifacts such as data pipelines and models.

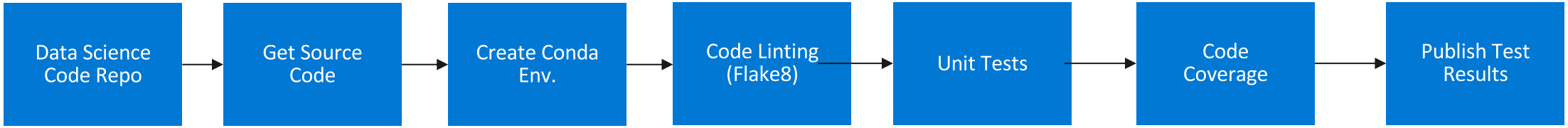
- Machine Learning model regarded as build artifact.
- New code to trigger model build (training).
- Build to include automated model quality tests.
- New succesful build (i.e. new model version) to trigger model release.





Continuous Integration for DS

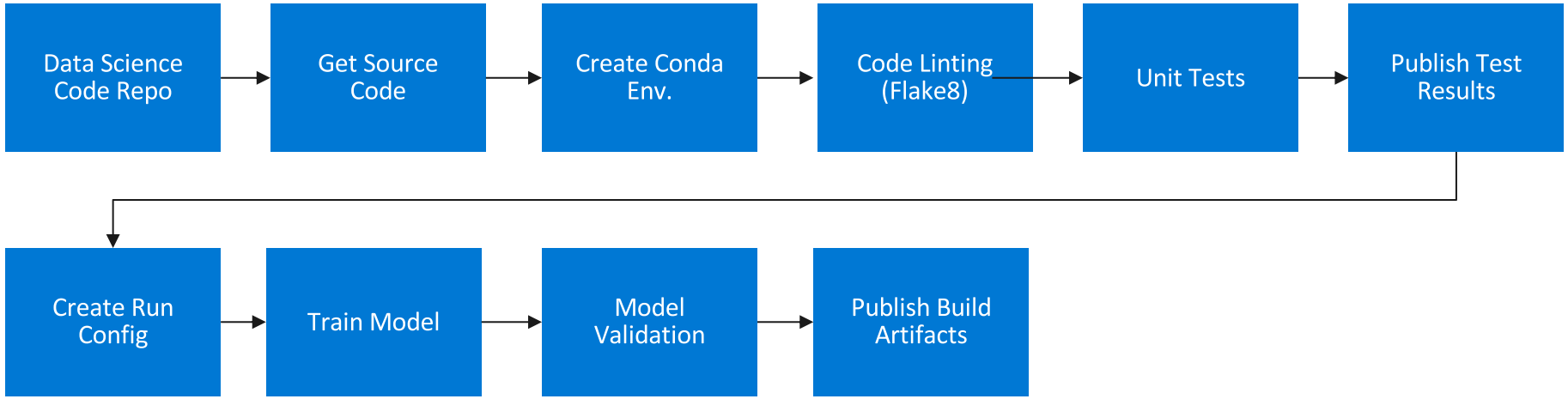
CI - Code Quality



- Trigger on Pull Request
- To keep build duration short, one may train a model for CI purposes on only a subset of the data.

Continuous Integration for DS

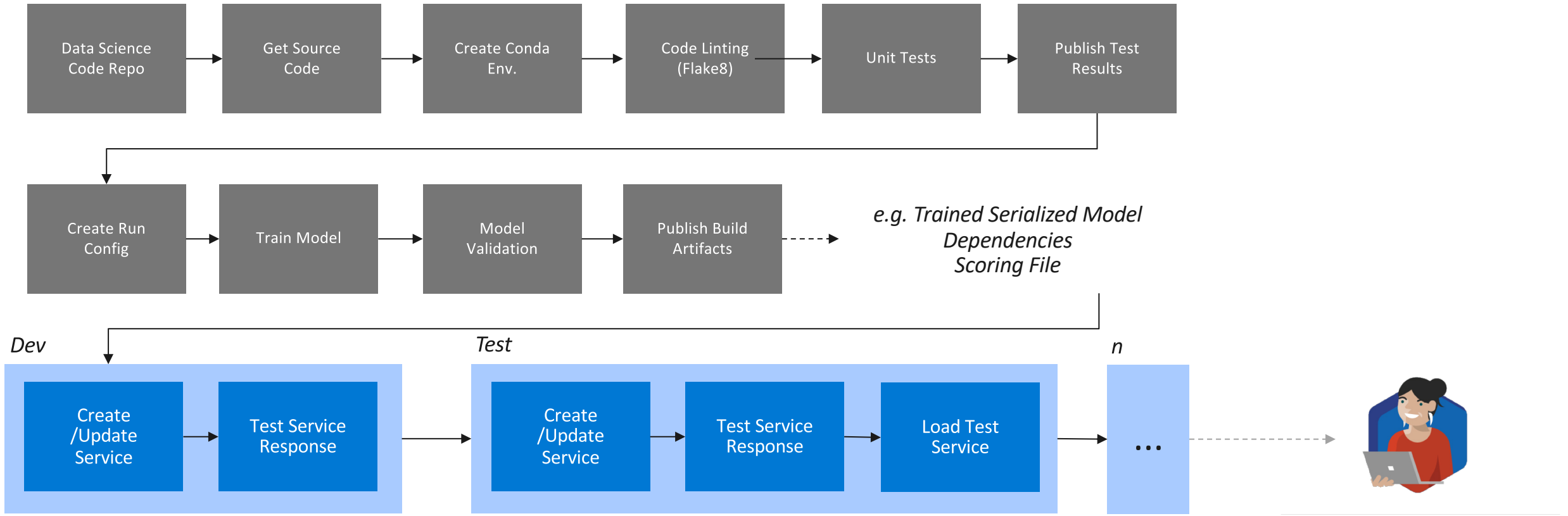
CI – Train Model



- Trained ML model regarded as software build artifact.
- Trigger on Pull Request as model training may take long.
- To keep build duration short, one may train a model for CI purposes, only on a subset of the data.

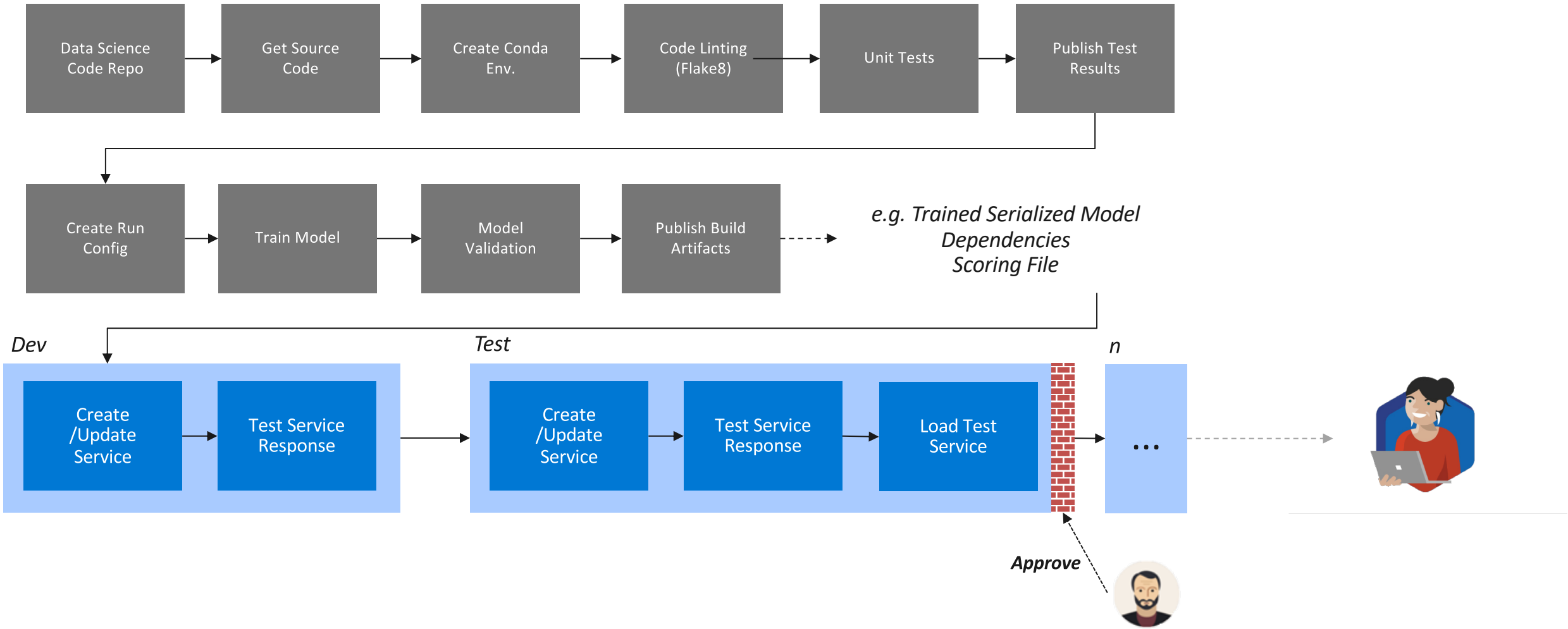
Continuous Delivery for DS

Release Model



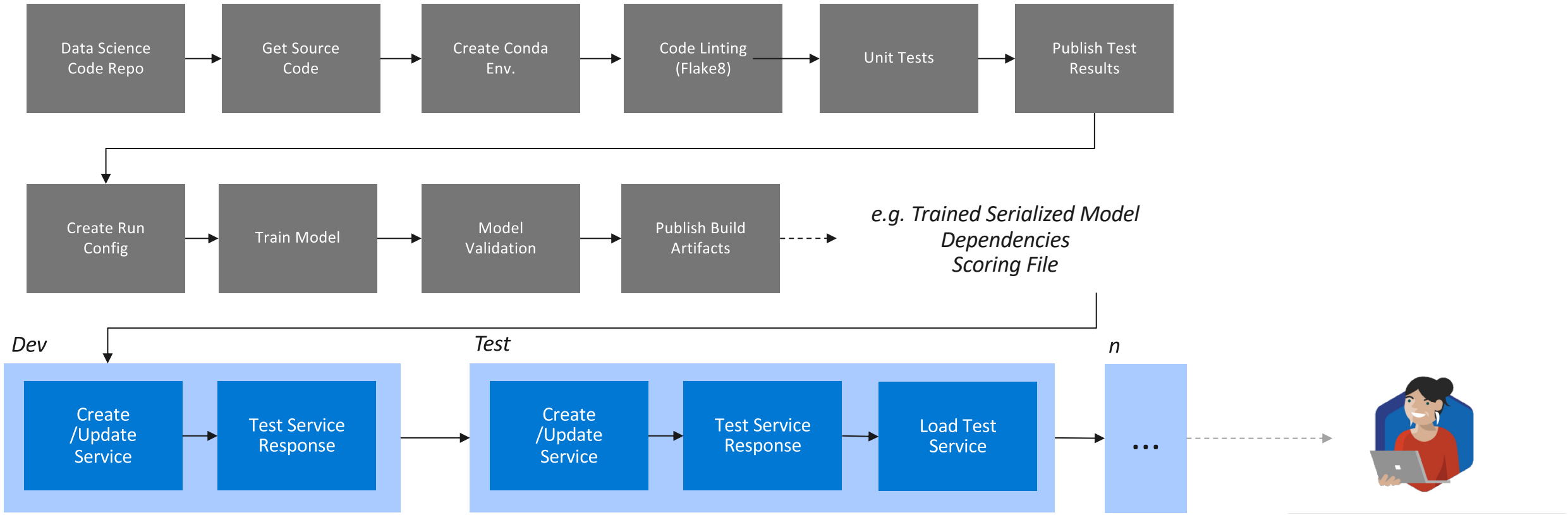
Continuous Delivery for DS

Release Model



Continuous ~~Delivery~~ Deployment

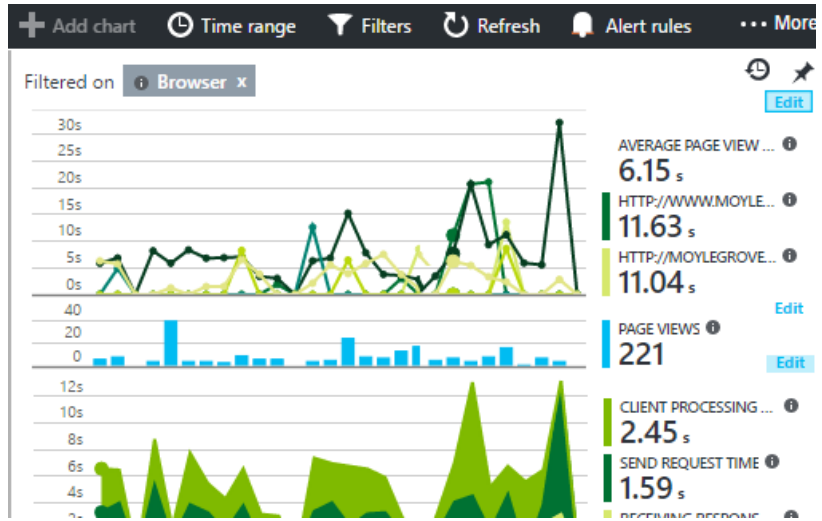
Release Model



Application Performance Monitoring

Concept drift

- Monitor Data / Model Quality
- Establish data feedback loop.
- Trigger retraining flow on set KPIs.



Model Serving Application

- Monitor request load and activities on real-time model scoring webservices.

Logging for reproducability

- Log model in- and outputs, model version
- Version dataset and codebase that led to model artifact.
- Version and log data pipeline execution that led to model input

Service Monitoring

Monitoring model serving applications allows for insights into service consumption, diagnose issues, diagnose user behavior and can allow you to timely detect performance anomalies.

Monitoring opportunities include:

- Request rates, response times, and failure rates.
- Dependency rates, response times, and failure rates.
- Exceptions.

Model Monitoring

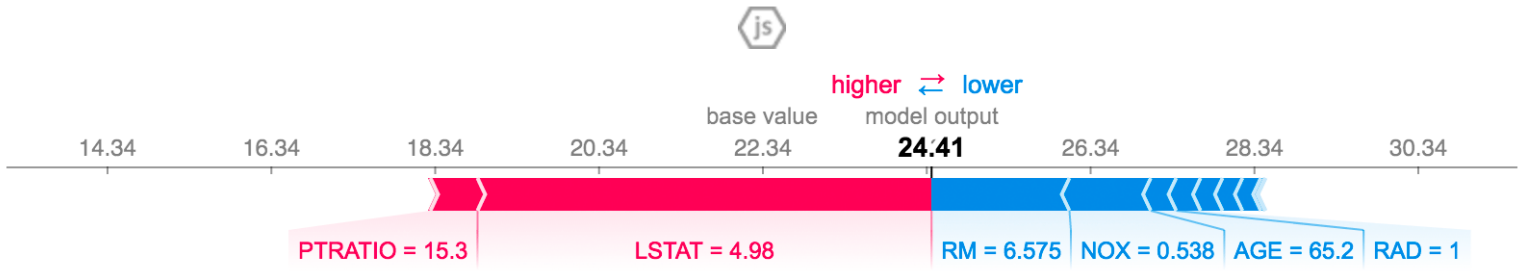
For sliding windows of collected data:

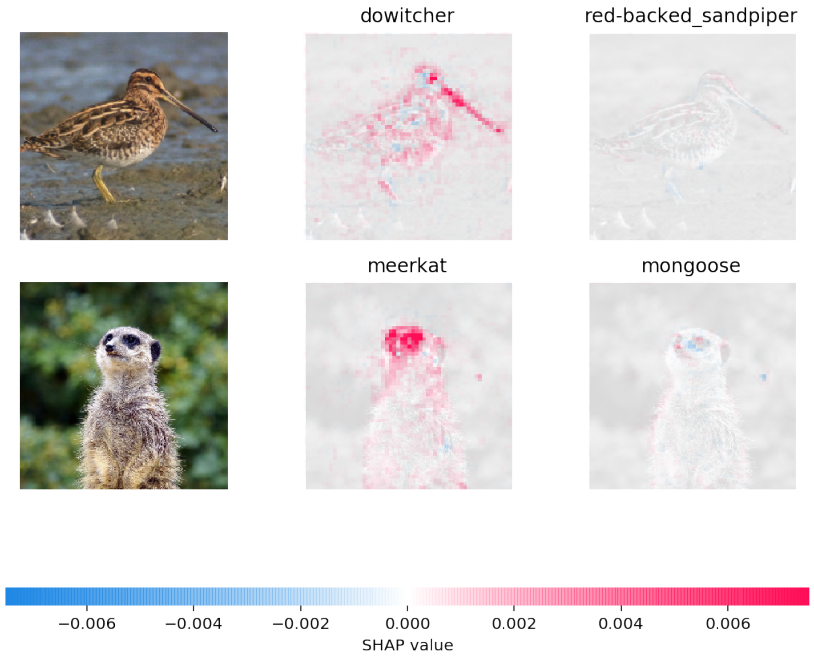
- Timely compare baseline dataset (model training data) with new data as collected during inferencing.
- For each window, apply tests on model risk parameters e.g. chi-squared test, model performance indicators, data quality, etc.
- Include logging for request-related entities such as user-, company- id’s to allow for comparison.
- Store results in ‘monitoring results’ data store for querying, comparative analysis over past periods and reporting.
- Use A/B testing to find right window size for comparison.

Operability & Transparancy

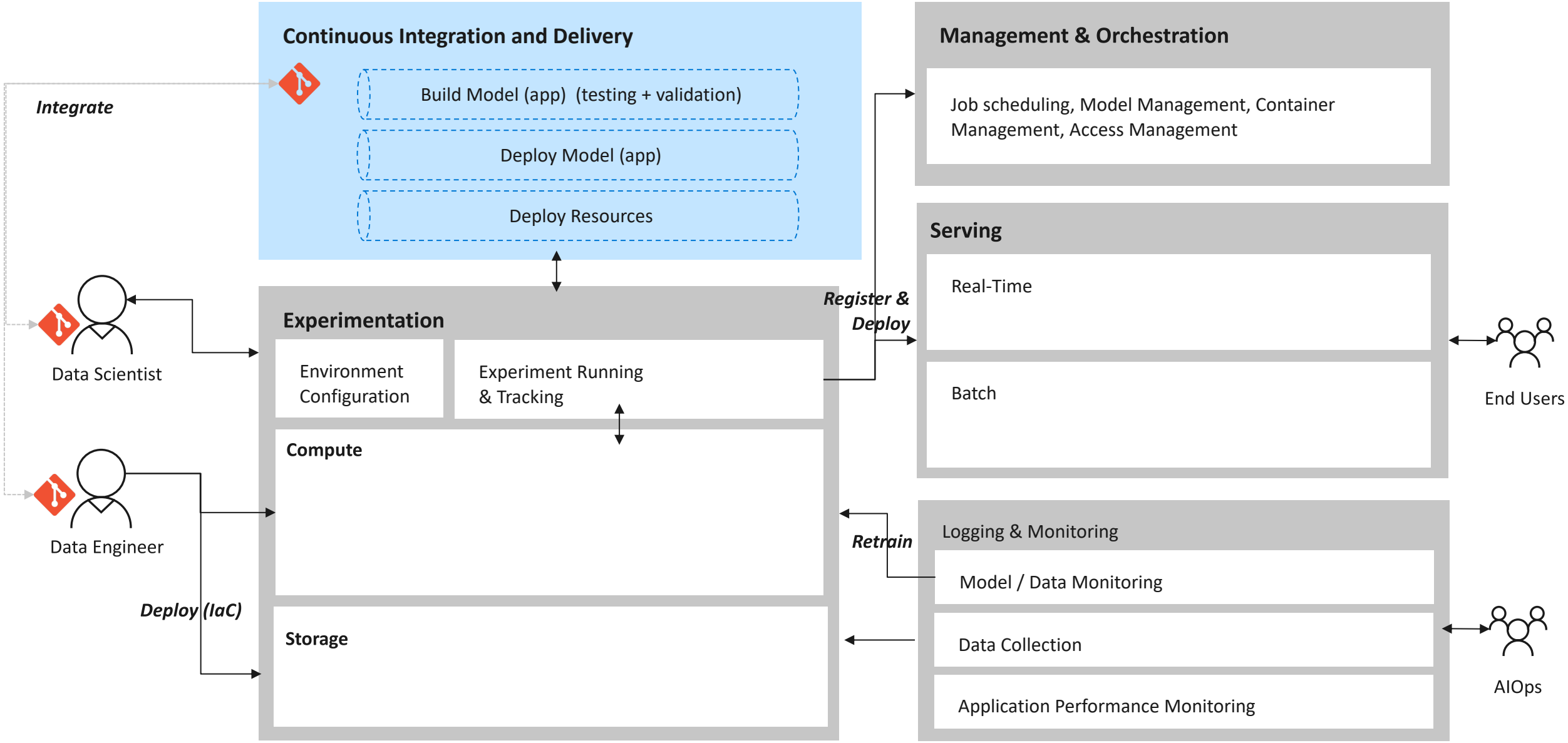
Being capable of providing operational support on deployed models, e.g.

- Understanding what model version was used for a certain prediction.
- Knowing the dataset/code that was used to train a specific model version.
- Use of explainability frameworks (SHAP, Lime) to explain main factors contributing to a specific model's prediction.

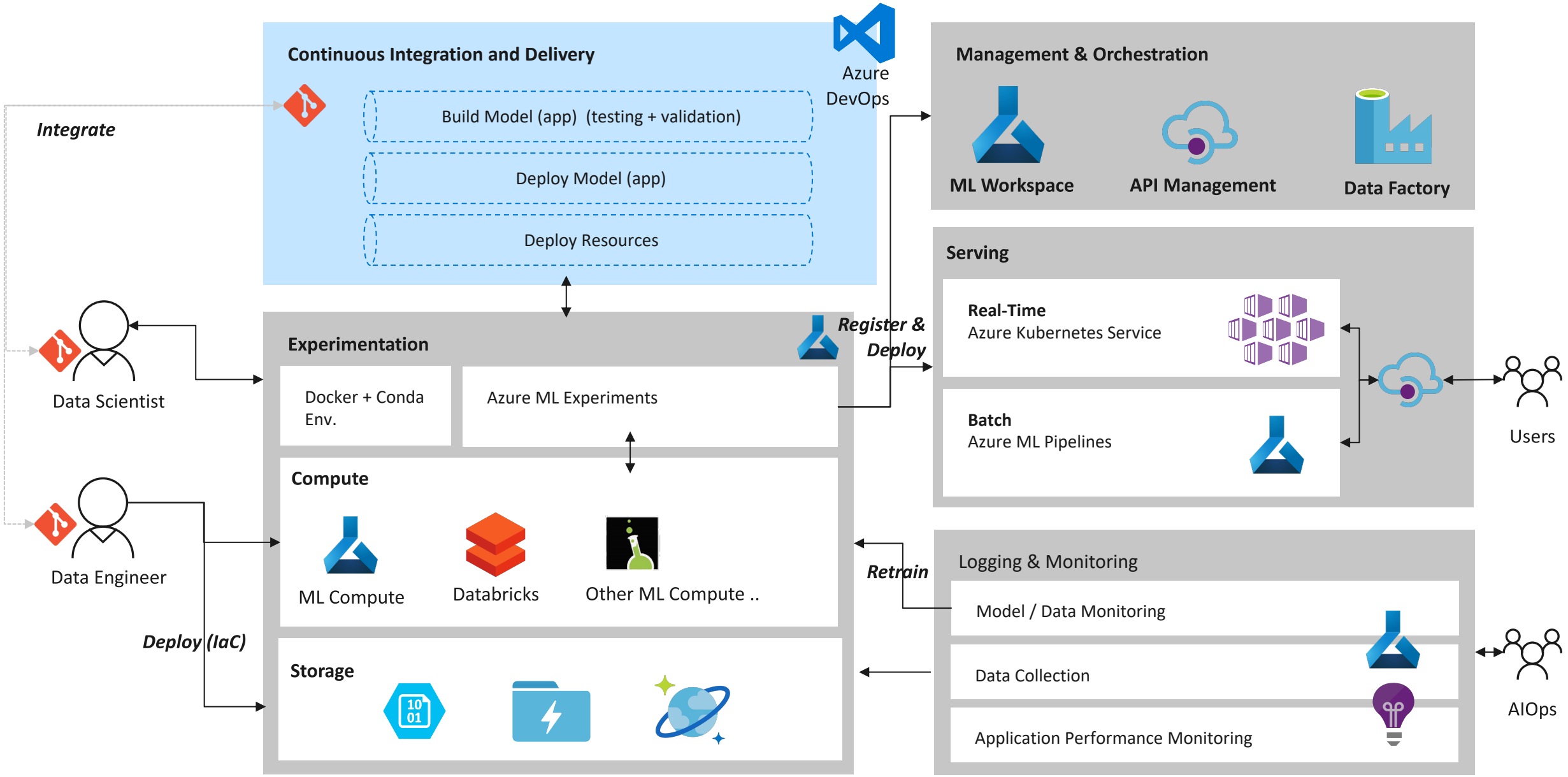




Technology Agnostic AI DevOps Solution



Azure


Continuous Integration and Delivery
Azure DevOps
Integrate
Build Model (app) (testing + validation)
Deploy Model (app)
Deploy Resources
Management & Orchestration
ML Workspace
API Management
Data Factory
Serving
Real-Time
Azure Kubernetes Service
Batch
Azure ML Pipelines
Users
Register & Deploy
Experimentation
Docker + Conda Env.
Azure ML Experiments
Data Scientist
Data Engineer
Compute
ML Compute
Databricks
Other ML Compute ..
Deploy (IaC)
Retrain
Storage
Logging & Monitoring
Model / Data Monitoring
Data Collection
Application Performance Monitoring
AIOps

Azure DevOps

Azure Boards

Deliver value to your users faster using proven agile tools to plan, track, and discuss work across your teams.

Azure Pipelines

Build, test, and deploy with CI/CD that works with any language, platform, and cloud. Connect to GitHub or any other Git provider and deploy continuously.

Azure Repos

Get unlimited, cloud-hosted private Git repos and collaborate to build better code with pull requests and advanced file management.

Azure Test Plans

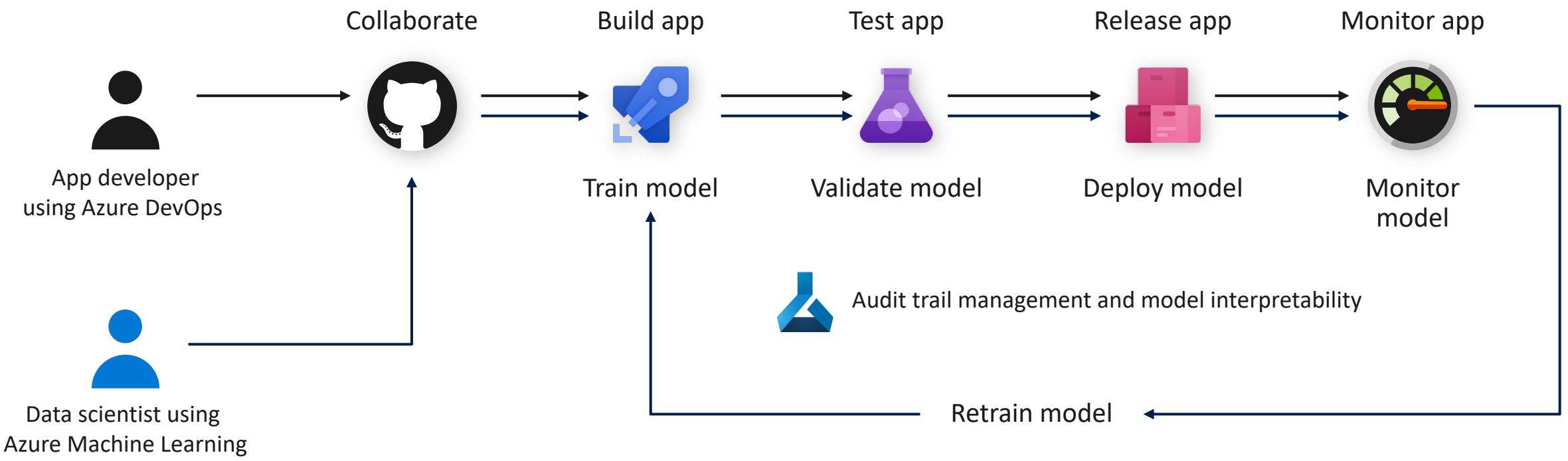
Test and ship with confidence using manual and exploratory testing tools.

Azure Artifacts

Create, host, and share packages with your team, and add artifacts to your CI/CD pipelines with a single click.

https://azure.com/devops

MLOps with Azure Machine Learning



- Model reproducibility
- Model validation
- Model deployment
- Model retraining

Azure MLOps

https://github.com/Microsoft/MLOps

https://github.com/mozamani/ai_iot