

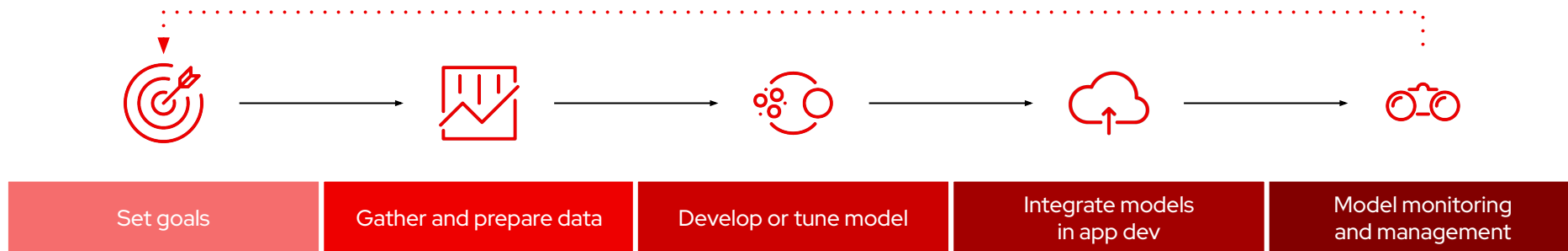
Un puente entre MLOps y DevOps con OpenShift AI

Juan Vicente Herrera
@jvicenteherrera



Move models from experimentation to production faster

Operationalize AI is the catalyst for incorporating AI into practical applications

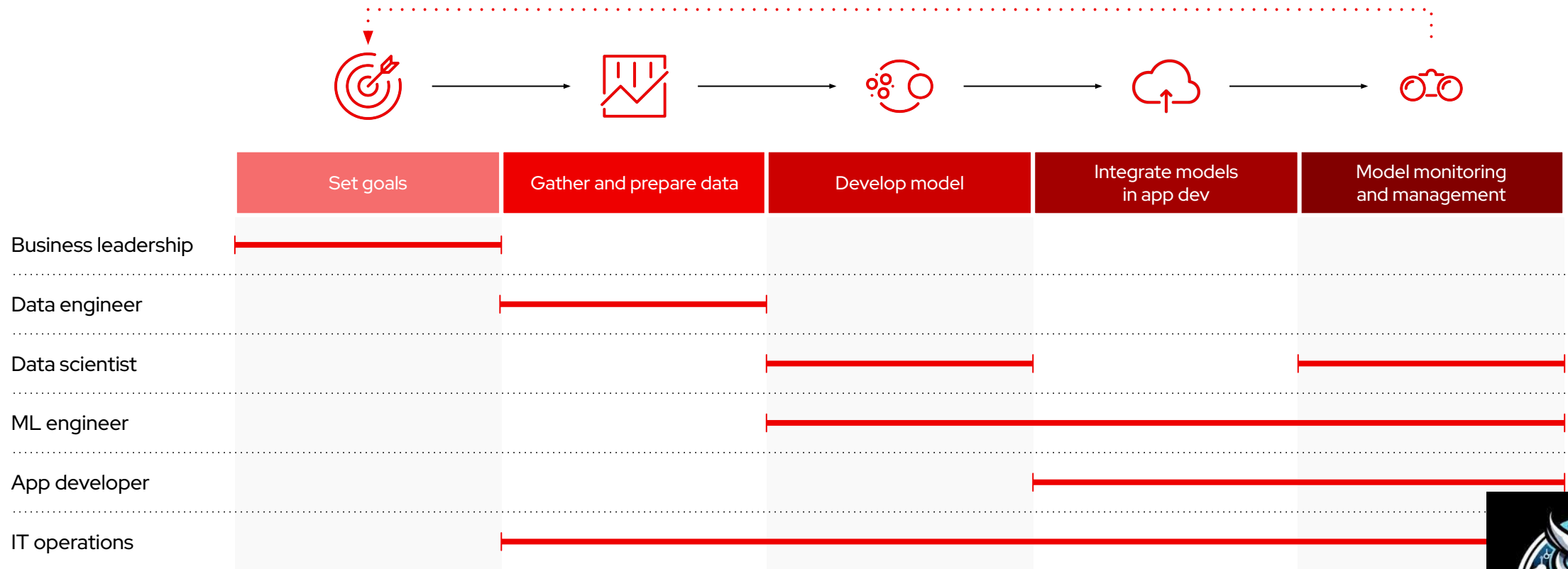


Operationalize AI is the process of integrating AI capabilities into the day-to-day operations of an organization. It involves taking your models from experimentation to production while contributing to the overall goals of the organization.

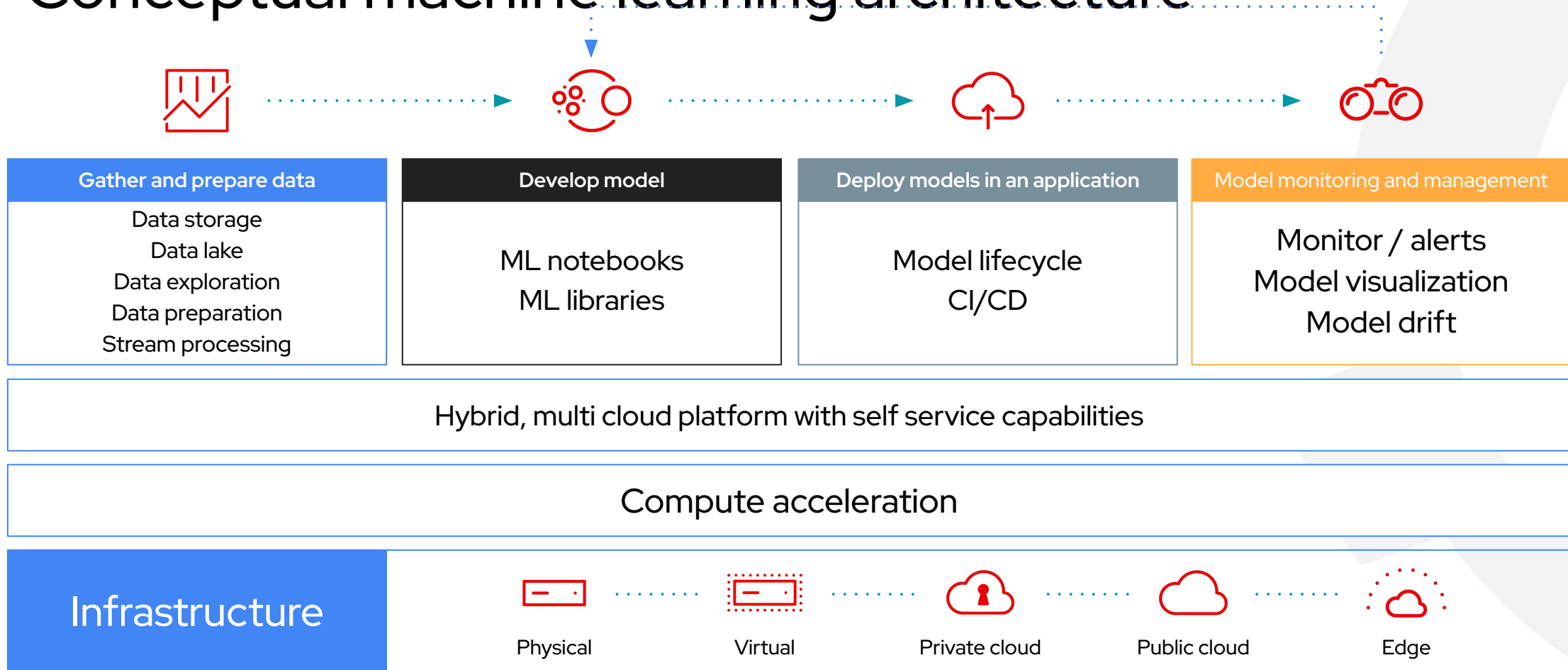


Operationalizing AI/ML requires collaboration

Every member of your team plays a critical role in a complex process

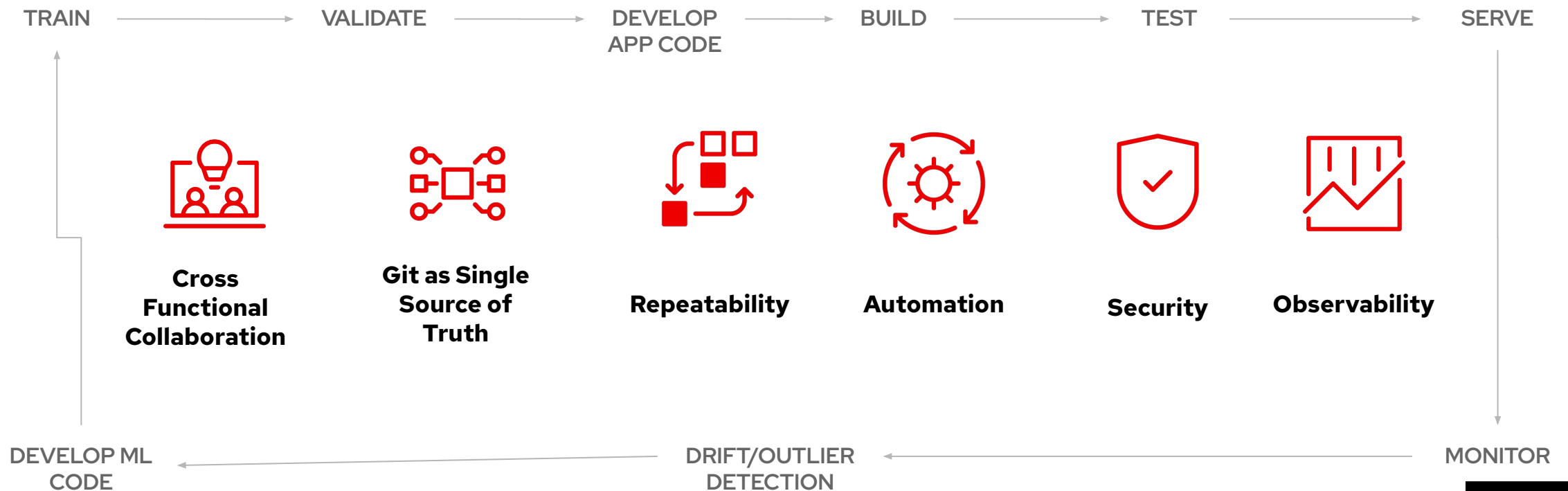


Conceptual machine learning architecture

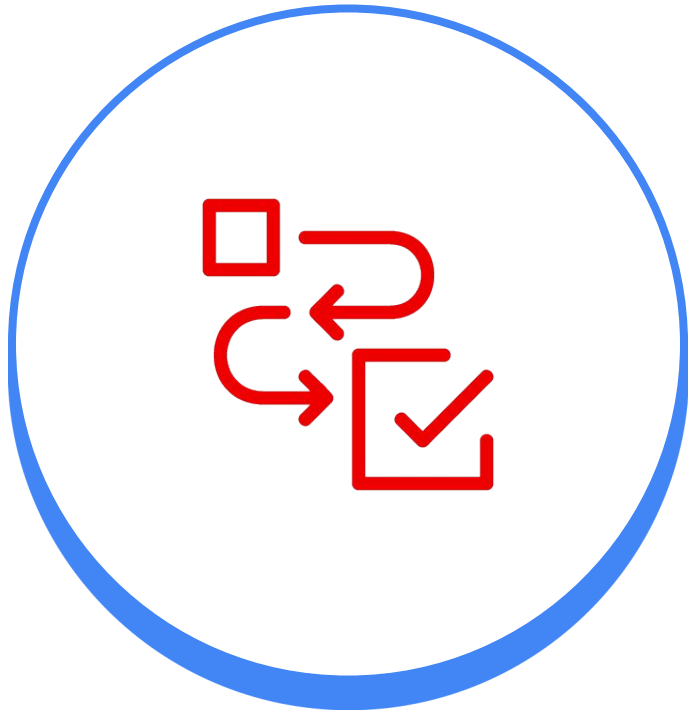


Enter MLOps

MLOps incorporates DevOps and GitOps to improve the lifecycle management of the ML application



Just like DevOps, MLOps requires changes.



Multi-disciplinary teams

Cross-train on the basics.

Automation

Automate everything that can be automated.

Patience

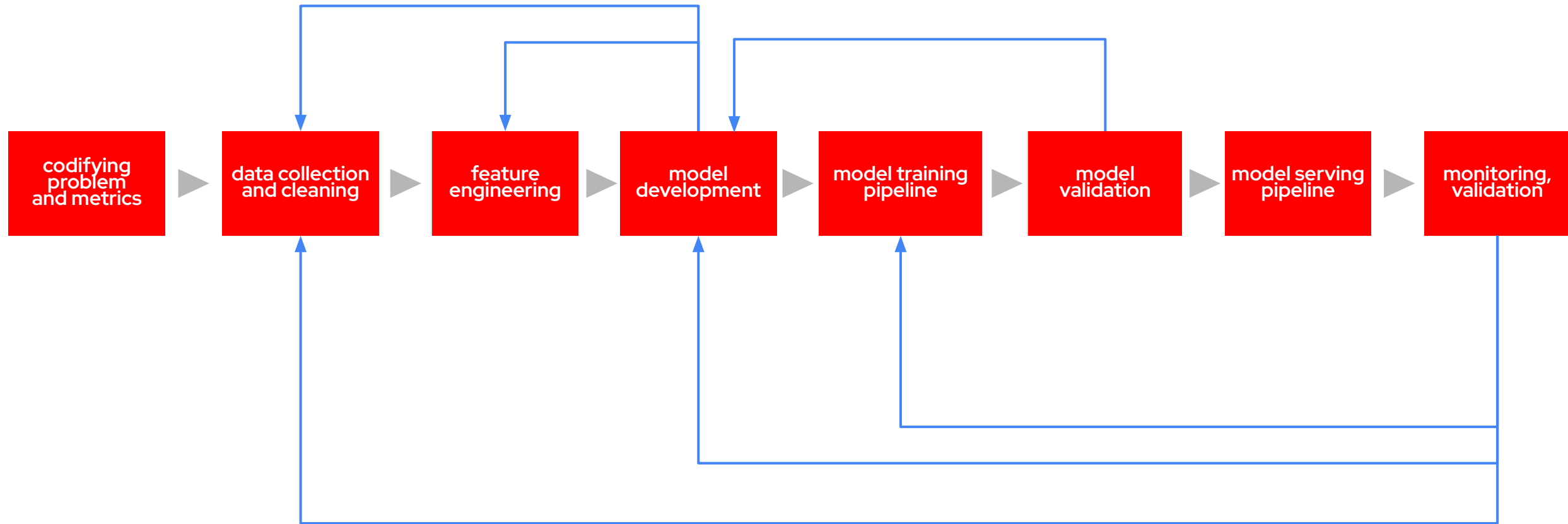
It's a gradual process, so it won't happen overnight.

Metrics

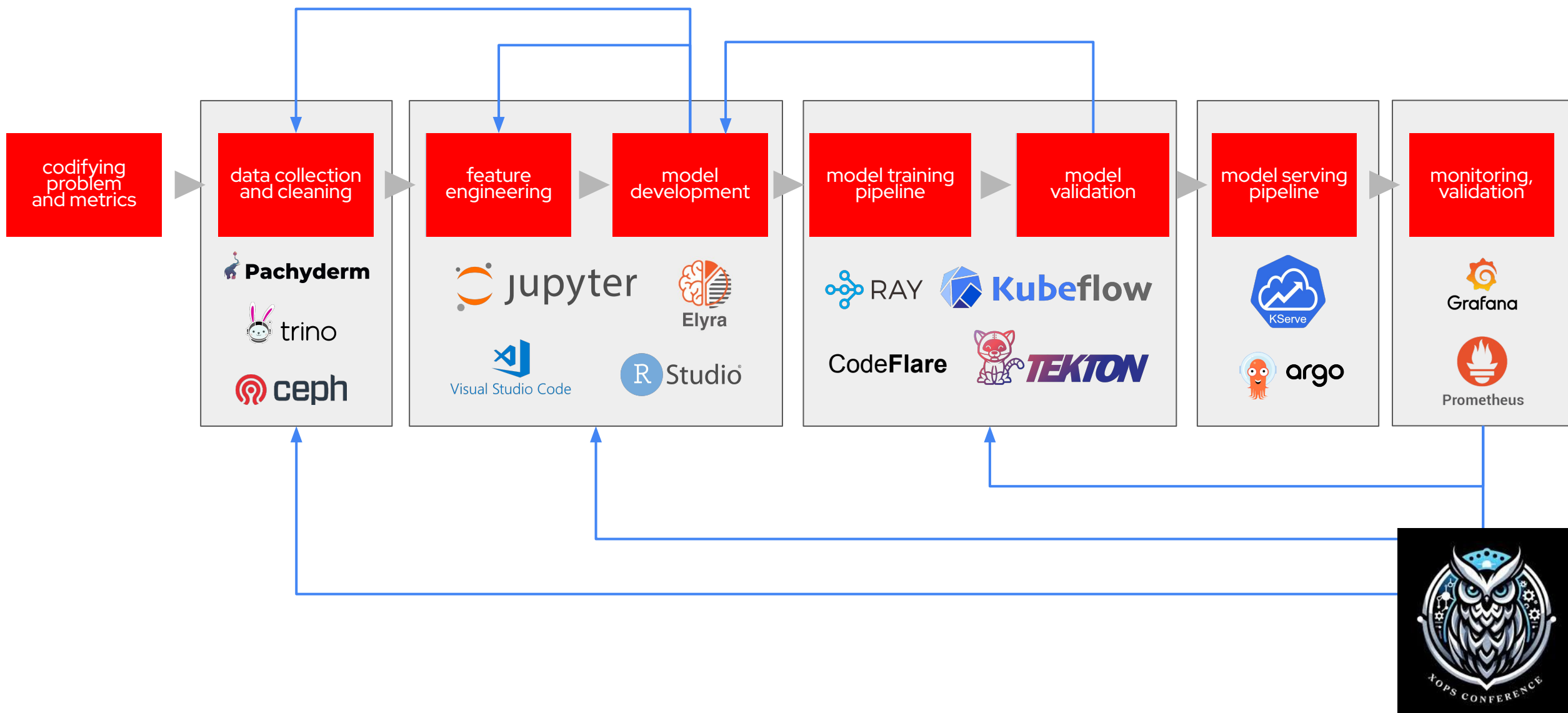
Pair measuring and tracking with transparency.



The MLOps workflow



MLOps with open source



Open Data Hub

An open source MLOps suite



- **Multi-tenant data science platform**
- **Self-service workbenches**



- **Preinstalled machine learning libraries**
- **Custom stack can be integrated**



- **Distributed model training**
- **Parallelize workloads across nodes and GPUs**



Elyra

- **AI pipeline editor**
- **Define workflows through Jupyter**



Kubeflow Pipelines

- **Machine learning workflow orchestration**
- **Experiment tracking**



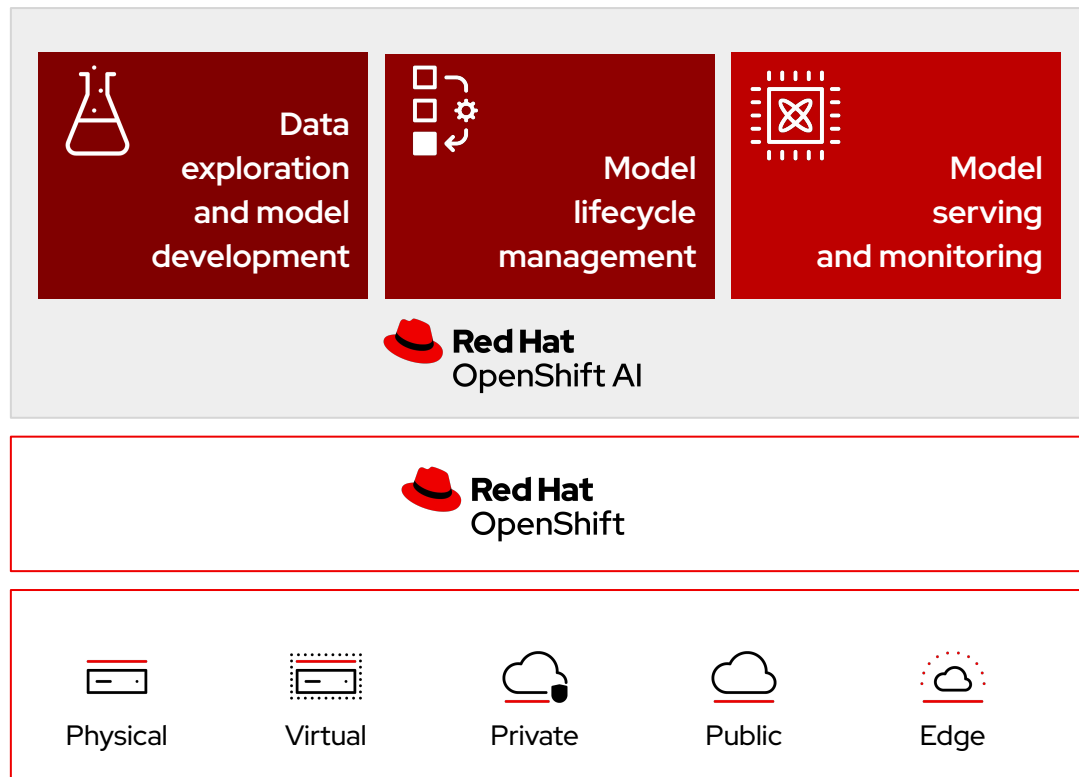
Kserve ModelMesh

- **Deploying machine learning models as micro-services**
- **Pre-built inference servers**



Understanding the building blocks

A common platform to bring IT, data science, and app dev teams together



Model development

Conduct exploratory data science in JupyterLab with access to core AI / ML libraries and frameworks including TensorFlow and PyTorch using our notebook images or your own.



Lifecycle Management

Create repeatable data science pipelines for model training and validation and integrate them with devops pipelines for delivery of models across your enterprise.

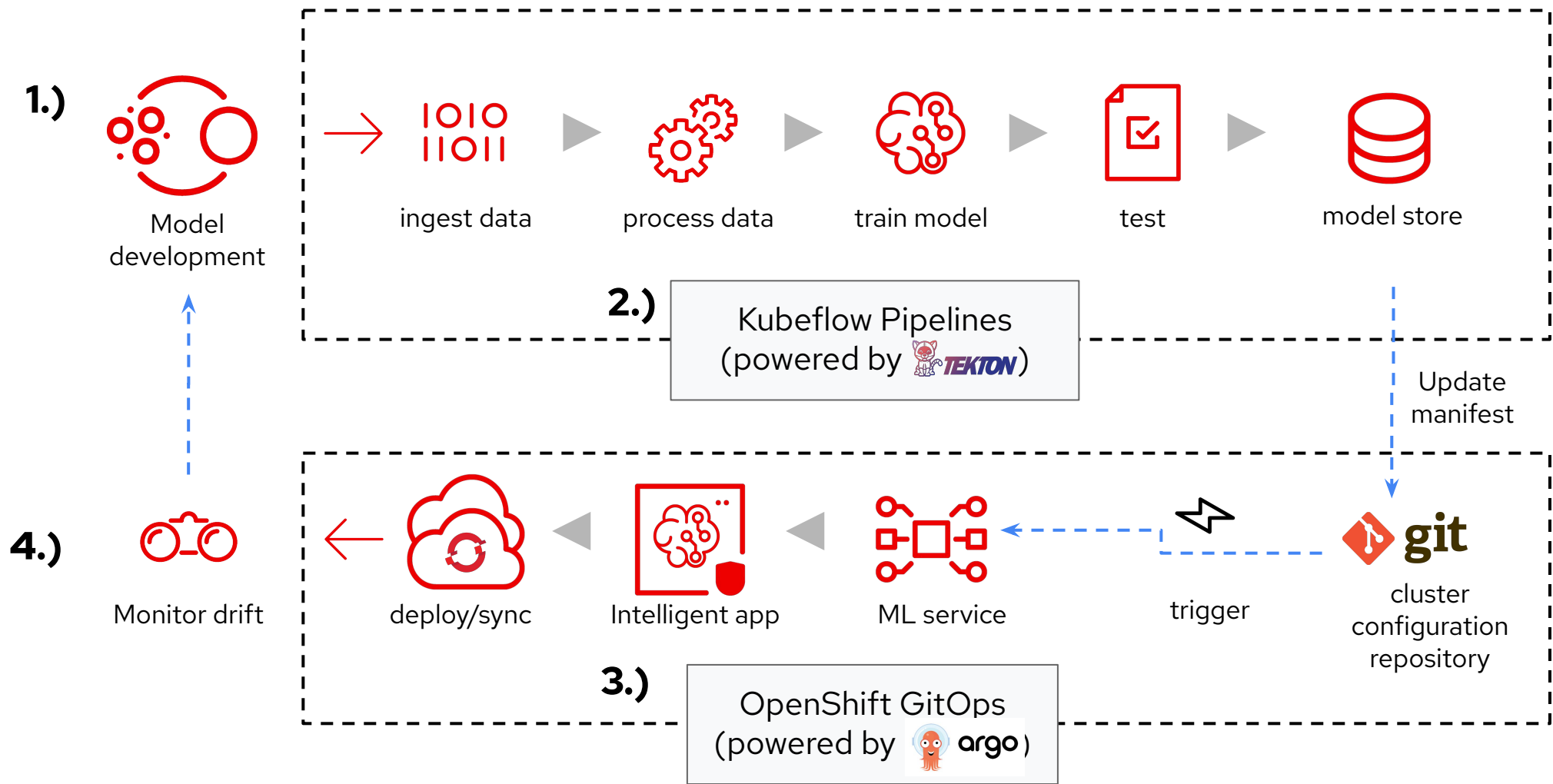


Model serving & monitoring

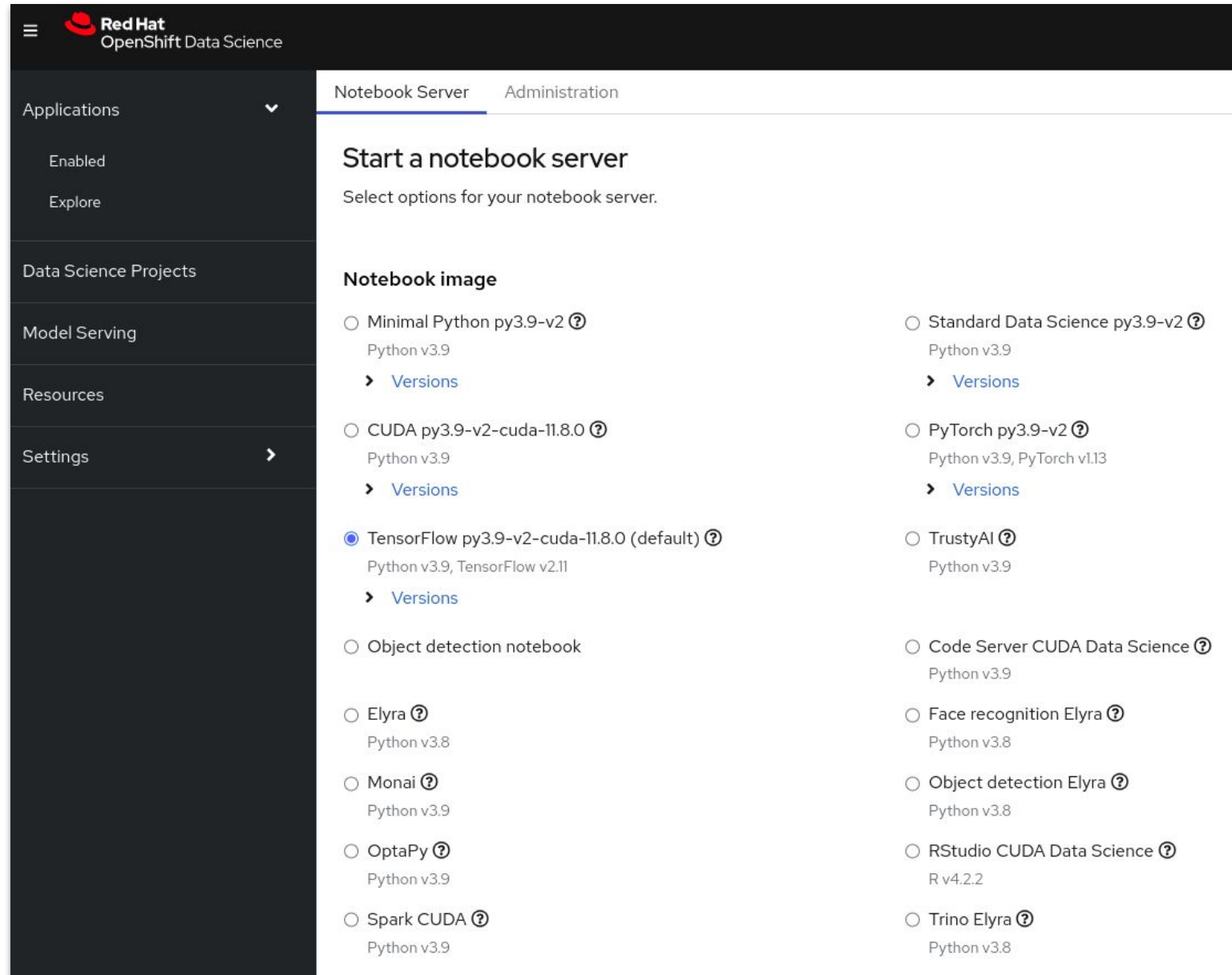
Deploy models across any cloud, fully managed, and self-managed OpenShift footprint and centrally monitor their performance.



MLOps with Red Hat OpenShift



Workbench images



Red Hat OpenShift Data Science

Applications

Enabled

Explore

Data Science Projects

Model Serving

Resources

Settings

Notebook Server Administration

Start a notebook server


Select options for your notebook server.

Notebook image

- ☐ Minimal Python py3.9-v2 ⓘ
Python v3.9
› Versions
- ☐ CUDA py3.9-v2-cuda-11.8.0 ⓘ
Python v3.9
› Versions
- ☒ TensorFlow py3.9-v2-cuda-11.8.0 (default) ⓘ
Python v3.9, TensorFlow v2.11
› Versions
- ☐ Object detection notebook
- ☐ Elyra ⓘ
Python v3.8
- ☐ Monai ⓘ
Python v3.9
- ☐ OptaPy ⓘ
Python v3.9
- ☐ Spark CUDA ⓘ
Python v3.9
- ☐ Standard Data Science py3.9-v2 ⓘ
Python v3.9
› Versions
- ☐ PyTorch py3.9-v2 ⓘ
Python v3.9, PyTorch v1.13
› Versions
- ☐ TrustyAI ⓘ
Python v3.9
- ☐ Code Server CUDA Data Science ⓘ
Python v3.9
- ☐ Face recognition Elyra ⓘ
Python v3.8
- ☐ Object detection Elyra ⓘ
Python v3.8
- ☐ RStudio CUDA Data Science ⓘ
R v4.2.2
- ☐ Trino Elyra ⓘ
Python v3.8



Data Science Projects

 **Red Hat**
OpenShift Data Science

Applications

Enabled

Explore

Data Science Projects

Model Serving

Resources

Settings

Data science projects

View your existing projects or create new projects.

Name

Find by name

Create data science project

Launch Jupyter

Name	Workbench	Status	Created
face detection ? admin	JupyterLab R Studio VS Code	<input checked="" type="checkbox"/> Running <input checked="" type="checkbox"/> Running <input checked="" type="checkbox"/> Running	5/7/2023, 2:01:57 PM
fraud detection ? admin	development	<input checked="" type="checkbox"/> Running	5/7/2023, 2:05:15 PM
myproject ? admin	myworkbench	<input checked="" type="checkbox"/> Running	5/8/2023, 11:11:59 AM
object detection ? admin	development	<input checked="" type="checkbox"/> Running	5/7/2023, 2:03:...



Project Resources

Red Hat

OpenShift Data Science

Applications

Enabled

Explore

Data Science Projects

Model Serving

Resources

Settings

Data science projects > object detection

object detection

Jump to section

Workbenches

Cluster storage

Data connections

Models and model servers

Workbenches

Create workbench

Name	Notebook image	Container size	Status	
> development	Object detection notebook	Small	Running	Open

Cluster storage

Add cluster storage

Name	Type	Connected workbenches	
> development	Persistent storage	development	

Data connections

Add data connection

Name	Type	Connected workbenches	Provider	
models	Object storage	No connections	AWS S3	

Models and model servers

Type	Deployed models	Tokens	
ovms	1	Tokens disabled	

Model name	Inference endpoint	Status
object-detection	https://object-detection-object-detection.apps.cluster-rb29x.rb29x.sandbox2896.opentlc.com/v2/model...	



Serve, scale, and monitor your models

Select the required resources and scale model serving as needed

Make your model public and secure

Configure model server

Model server replicas

Number of model server replicas to deploy

- 1 +

Compute resources per replica

Model server size

Small

Model route

☒ Make deployed available via an external route

Token authorization

☐ Require token authentication

Configure Cancel

Deploy model

Configure properties for deploying your model

Project

modelserving-test

Name *

myModel

Model framework

onnx - 1

Model location

☒ Existing data connection

Name

storage-config

Folder path

onnx/road_conditions.onnx

☐ New data connection

Deploy Cancel

Select your model framework

Models and model servers

Deploy model

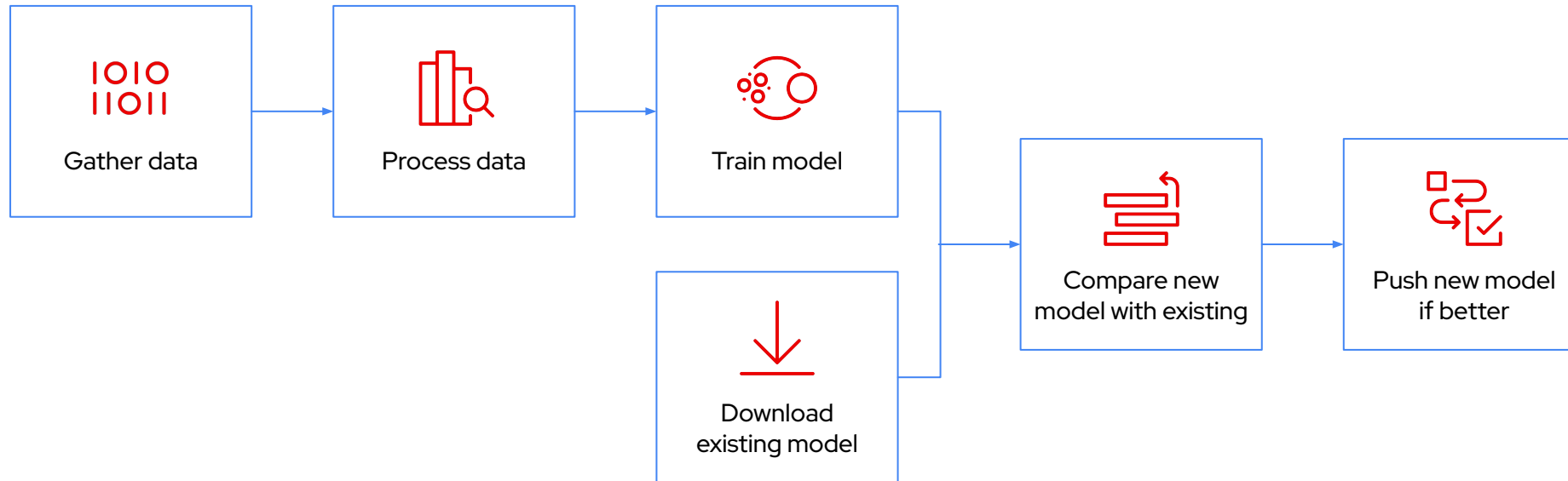
Type	Deployed models	Tokens
ovms	1	Tokens disabled

Model name	Inference endpoint	Status
myModel	https://mymodel-modelserving-test.apps.pilot.j61u.p1.openshiftapps.com/v2/models/mymodel/infer	

View your deployed model fleet endpoint



Data science pipelines component



- ▶ Continuously deliver and test models in production
- ▶ Schedule, track, and manage pipeline runs
- ▶ Easily build pipelines using graphical front end
- ▶ Orchestrate data science tasks into pipelines
- ▶ Chain together processes like data prep, build models, and serve models



Creating Pipelines with Kubeflow Pipelines

- Data Science Pipelines (DSP) allows data scientists **to track progress as they iterate over development** of ML models. With DSP, a data scientist
- They can create and **track experiments to arrive at the best version of of training data, model hyperparameters, model code**, etc., and repeatably rerun these experiments.



Creating Pipelines with Kubeflow Pipelines

Flip coin example

1. Define a pipeline in Python.
2. Compile the Python file into a Tekton resource definition.
3. Import the pipeline.
4. Execute the pipeline.

The screenshot displays the Kubeflow Pipelines interface for a specific pipeline run. At the top, the breadcrumb navigation shows 'Runs - elyra-pipeline' followed by the run ID 'issues_prediction-01-0528130920'. Below this, the pipeline name 'issues_prediction-01-0528130920' is shown alongside its execution mode 'One-off' and status 'Completed'. An 'Actions' button is located in the top right corner. The central part of the interface features a visual representation of the pipeline as a sequence of three steps: 'data_ingestion', 'data_p...ssing', and 'data_t...sting', each marked with a green checkmark to indicate successful completion. Below the pipeline diagram are icons for zooming in, zooming out, and full-screen viewing. At the bottom, there are three tabs: 'Details', 'Input parameters', and 'Run output'. The 'Details' tab is currently selected, showing a table with the following information:

Name	issues_prediction-01-0528130920
Pipeline version	issues_prediction-01
Project	elyra-pipeline
Run ID	309d89a3-b28b-4e1c-b95a-ad7b4c1c87aa
Workflow name	issues_prediction-01-200-d8

Creating and Using Model Servers

1. Create a custom Scikit-learn model server by creating a Python container and a RHOAI serving runtime.
2. Deploy a version of the diabetes model trained with Scikit-learn by using the RHOAI console.



ENJOY
XOPS
CONFERENCE

www.xops.co.uk

XOPS

XOPS

XOPS

XOPS