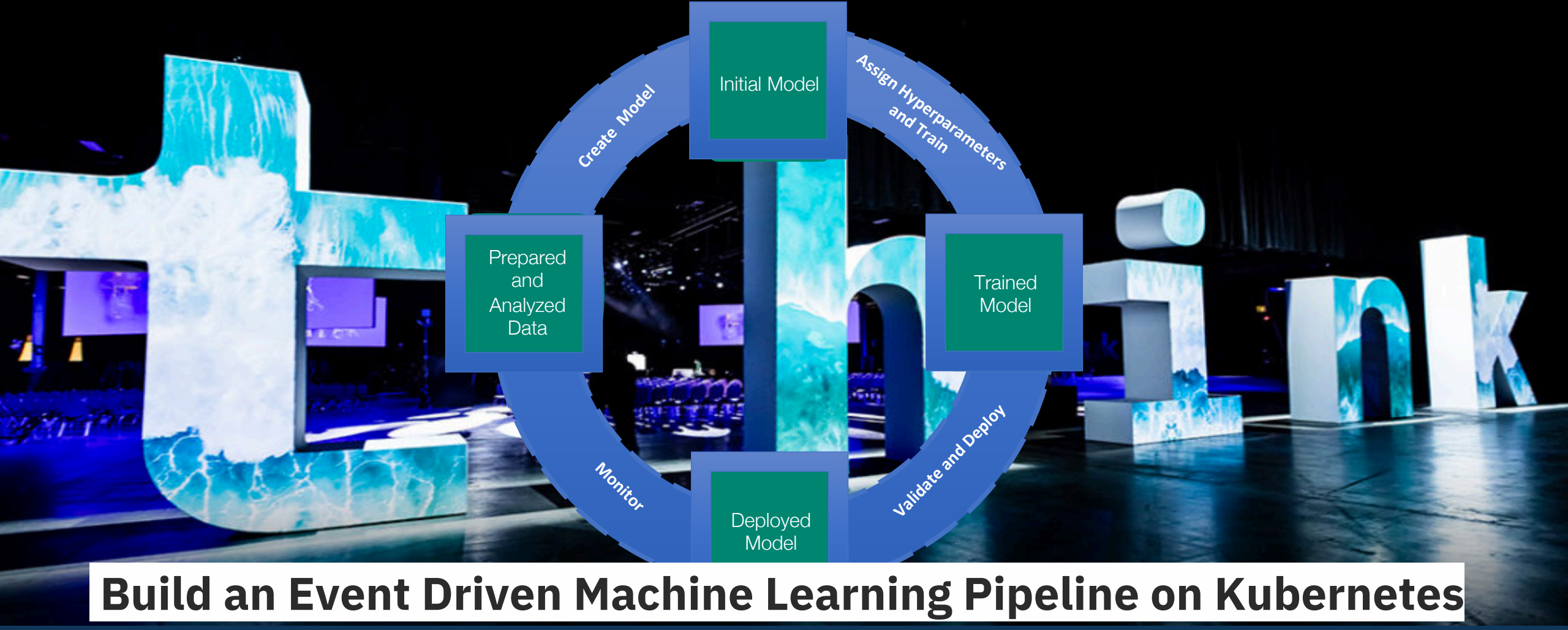


Build an Event Driven Machine Learning Pipeline on Kubernetes

Hou Gang Liu
Advisory Software Developer IBM
kubeflow katib/manifest maintainer

houganglliu

Animesh Singh
STSM and Program Director IBM
kubeflow kfserving maintainer

animeshsingh

Center for Open Source Data and AI Technologies (CODAIT)

Code – Build and improve practical frameworks to enable more developers to realize immediate value.

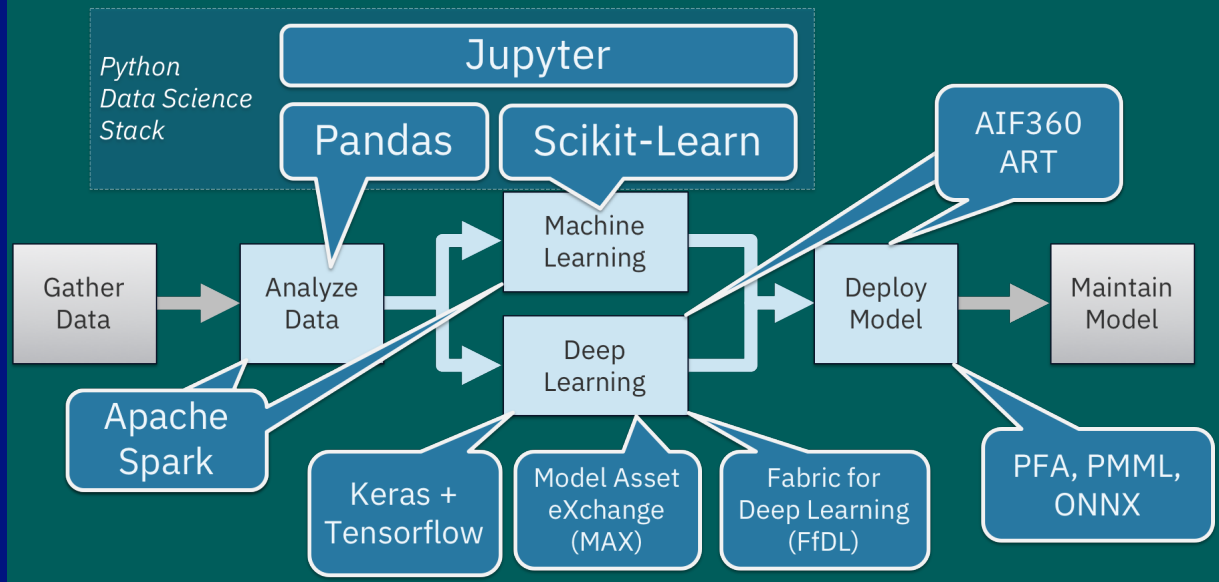
Content – Showcase solutions for complex and real-world AI problems.

Community – Bring developers and data scientists to engage with IBM

- Team contributes to over **10 open source projects**
- **17 committers** and many contributors in Apache projects
- Over **1100 JIRAs** and **66,000 lines of code** committed to Apache Spark itself; over **65,000 LoC into SystemML**
- Over **25 product lines** within IBM leveraging Apache Spark
- Speakers at over 100 conferences, meetups, unconferences and more

IBM CODE

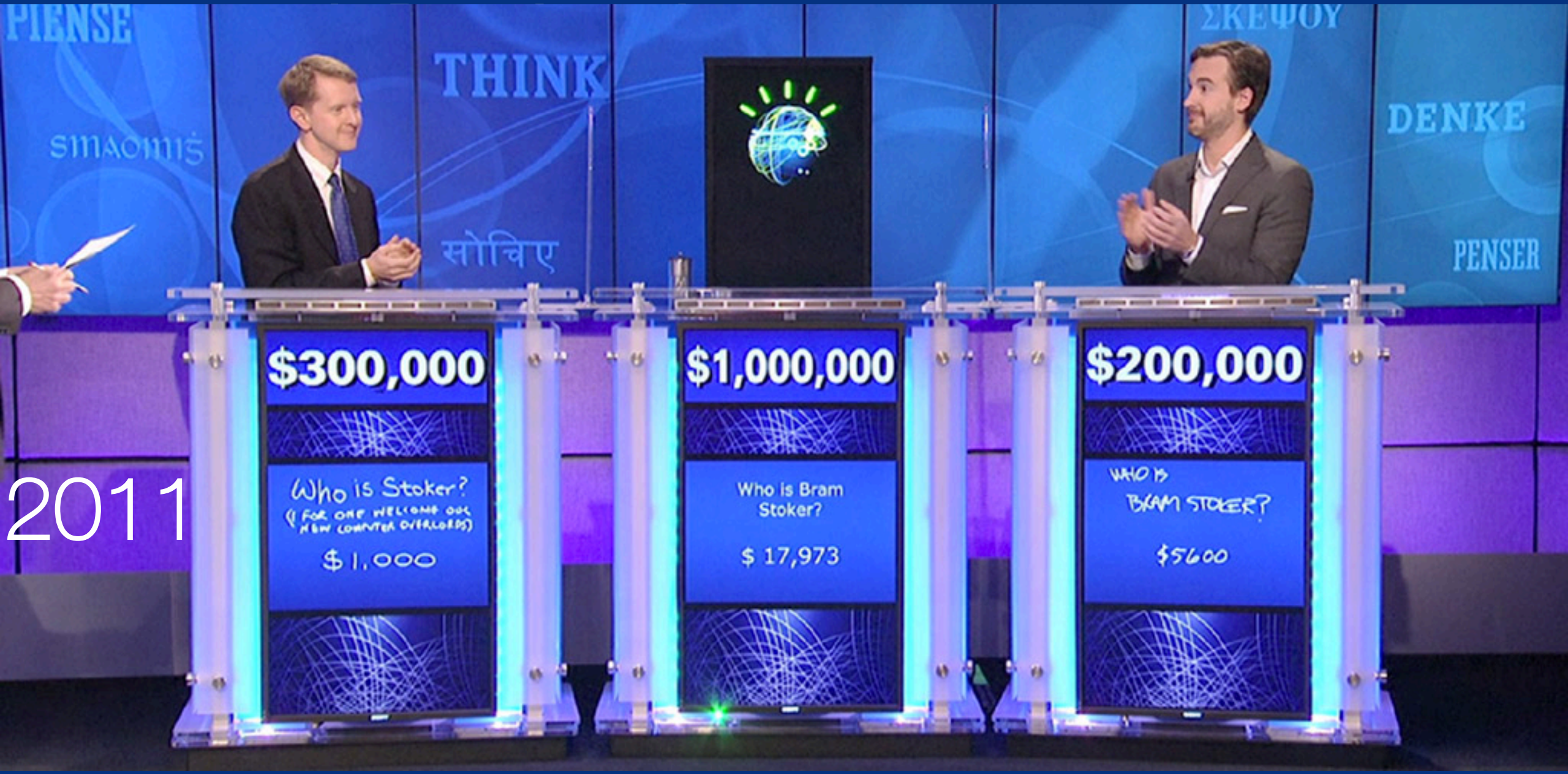
Improving Enterprise AI lifecycle in Open Source

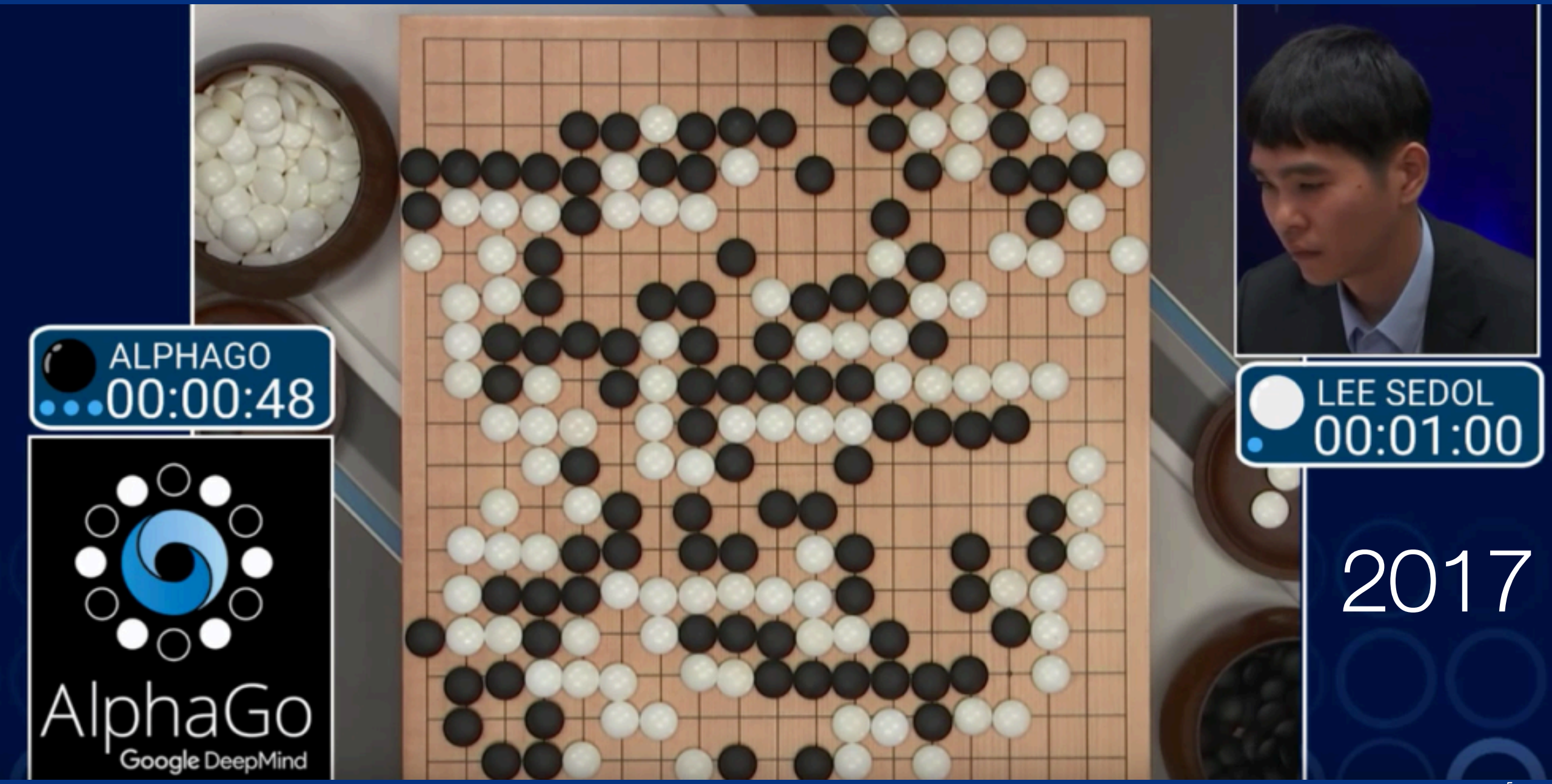


CODAIT

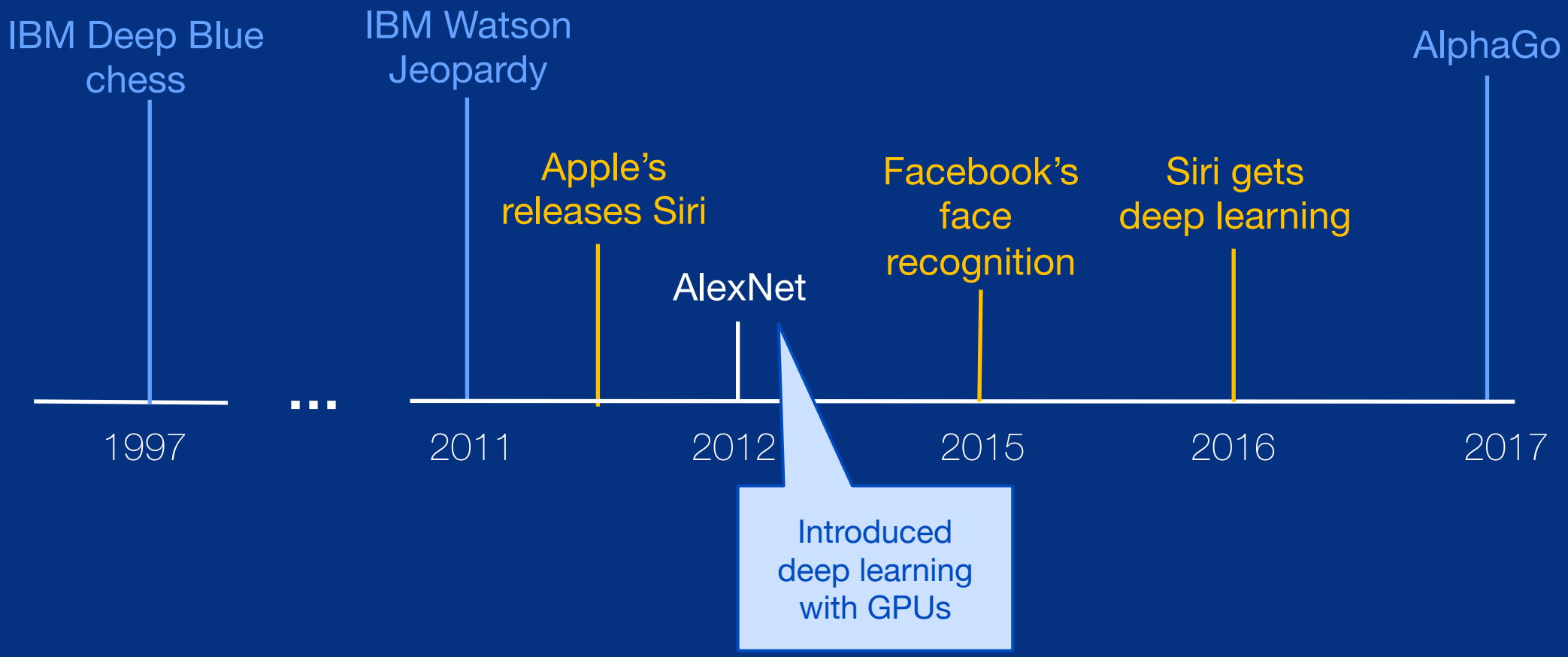
codait.org

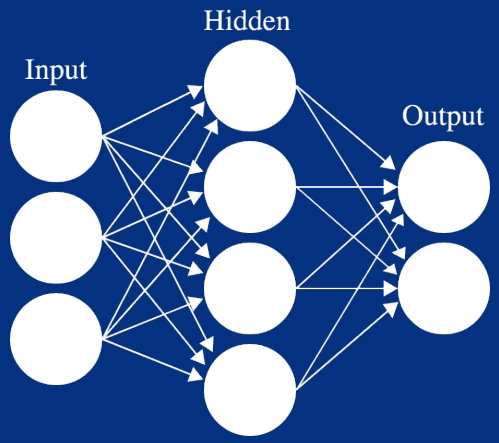

1997
2:00
2:00
GARRY
KASPAROV
DEEP
BLUE


PIENSE
THINK
σκεφου
DENKE
सोचिए
PENSER
$300,000
Who is Stoker?
((FOR ONE WELCOME OUR NEW COMPUTER OVERLORDS))
$1,000
$1,000,000
Who is Bram Stoker?
$ 17,973
$200,000
WHO IS BRAM STOKER?
$5600
2011


ALPHAGO
00:00:48
AlphaGo
Google DeepMind
LEE SEDOL
00:01:00
2017

Progress in Deep Learning





Deep Learning = Training Artificial Neural Networks

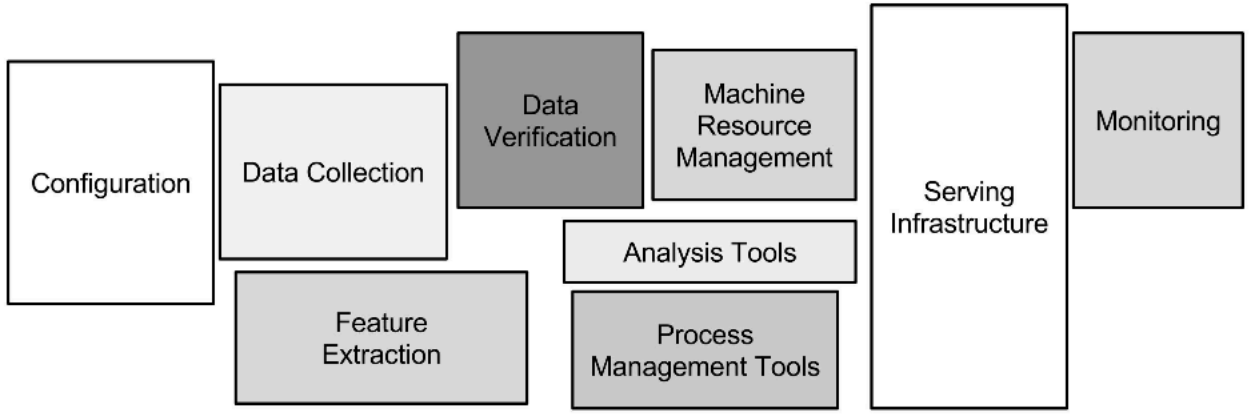
- 25 million “neurons”
- 100 million connections (parameters)

A human brain has:

- 200 billion neurons
- 32 trillion connections between them

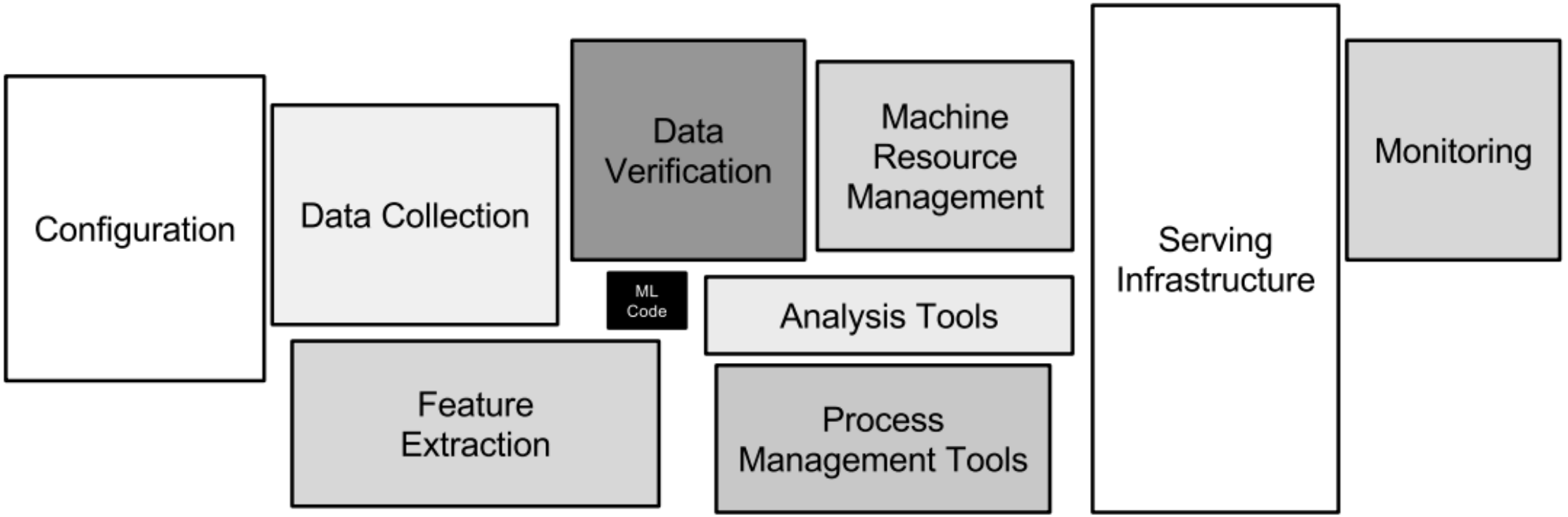
Perception

ML Code



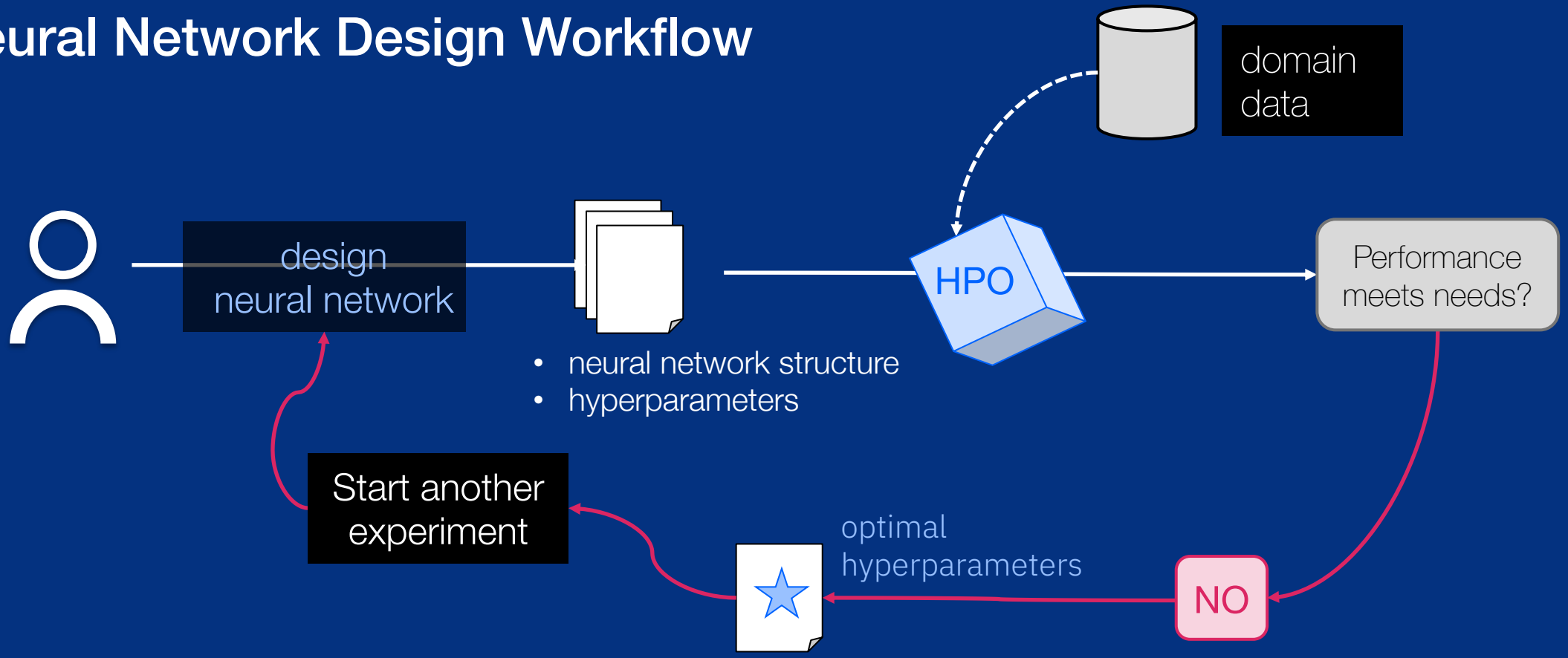
*Source: Hidden Technical Debt in Machine Learning Systems

In reality …

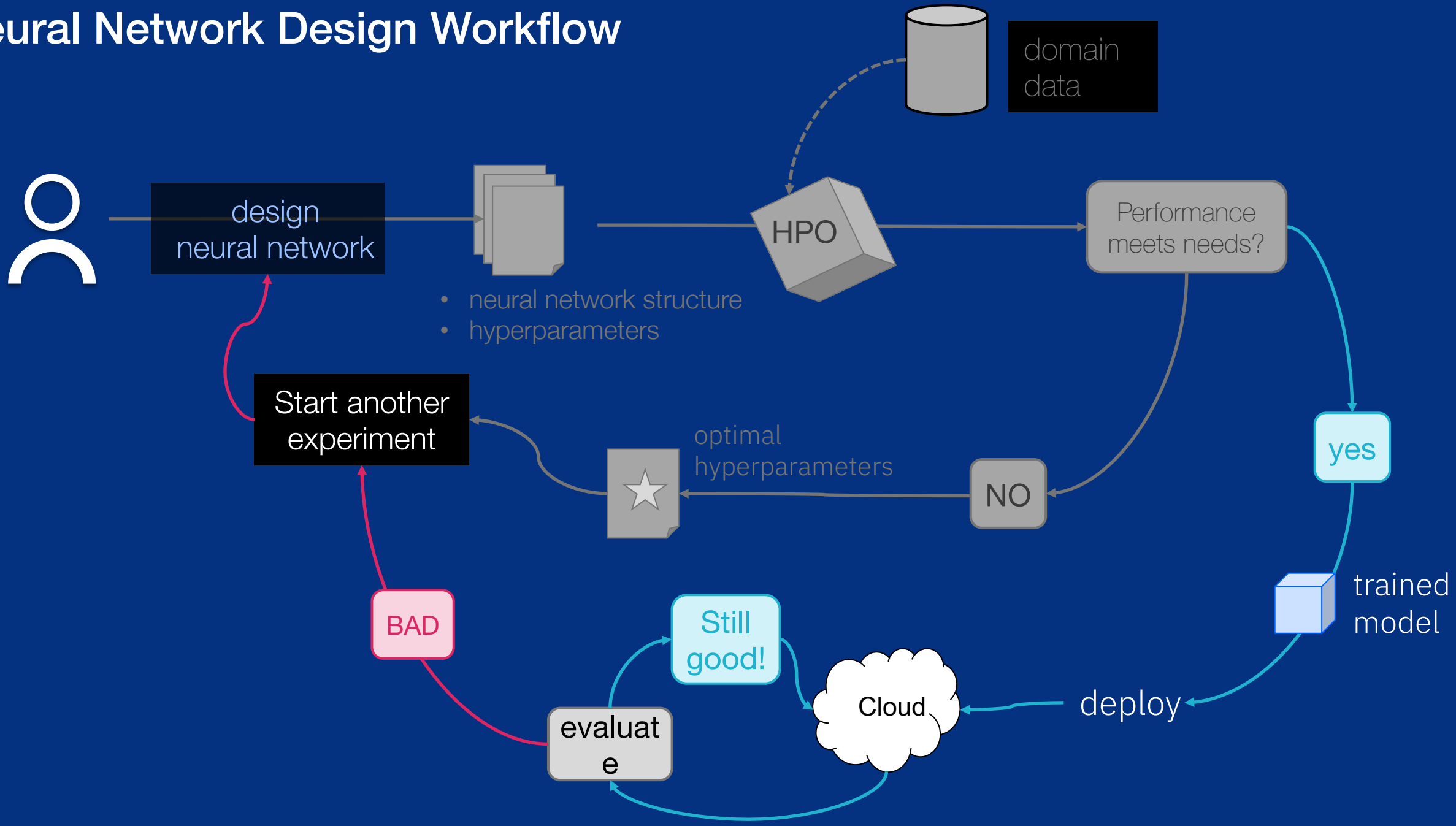


*Source: Hidden Technical Debt in Machine Learning Systems

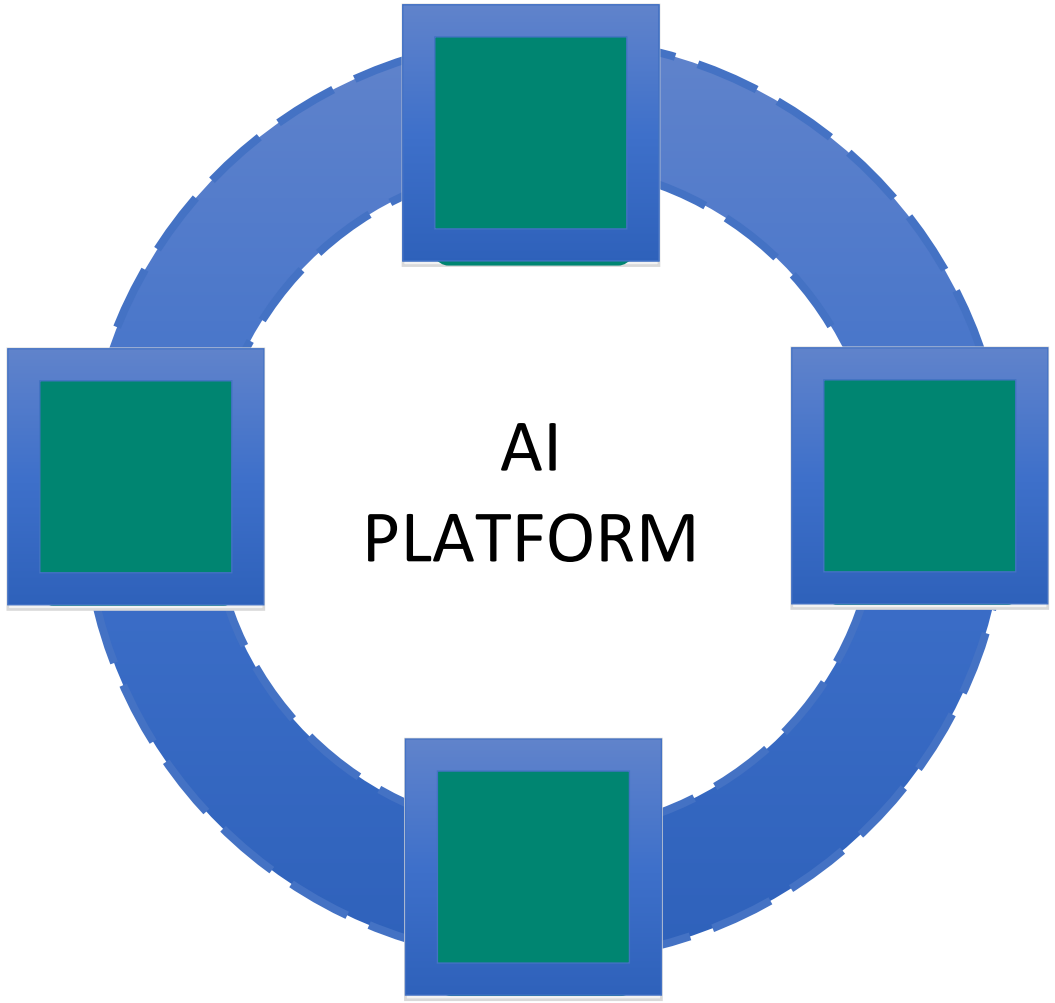
Neural Network Design Workflow



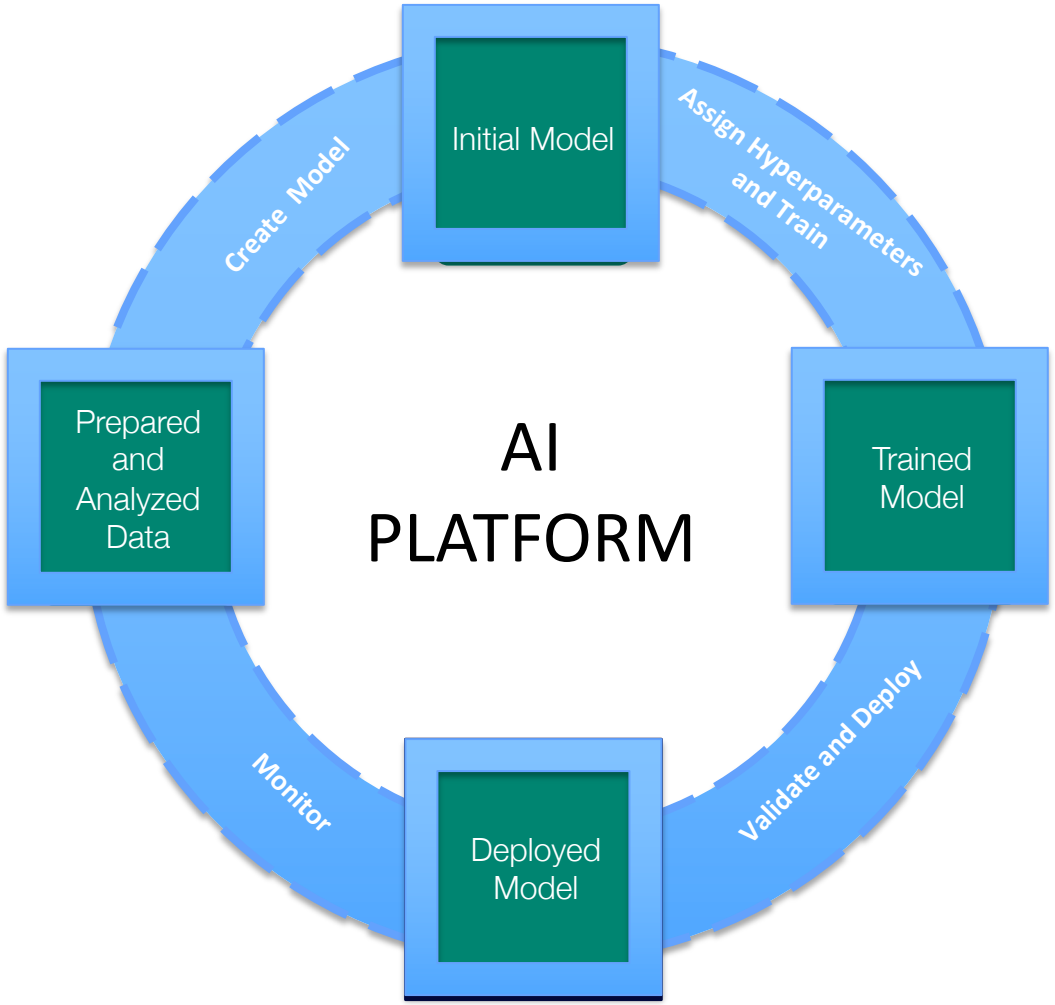
Neural Network Design Workflow



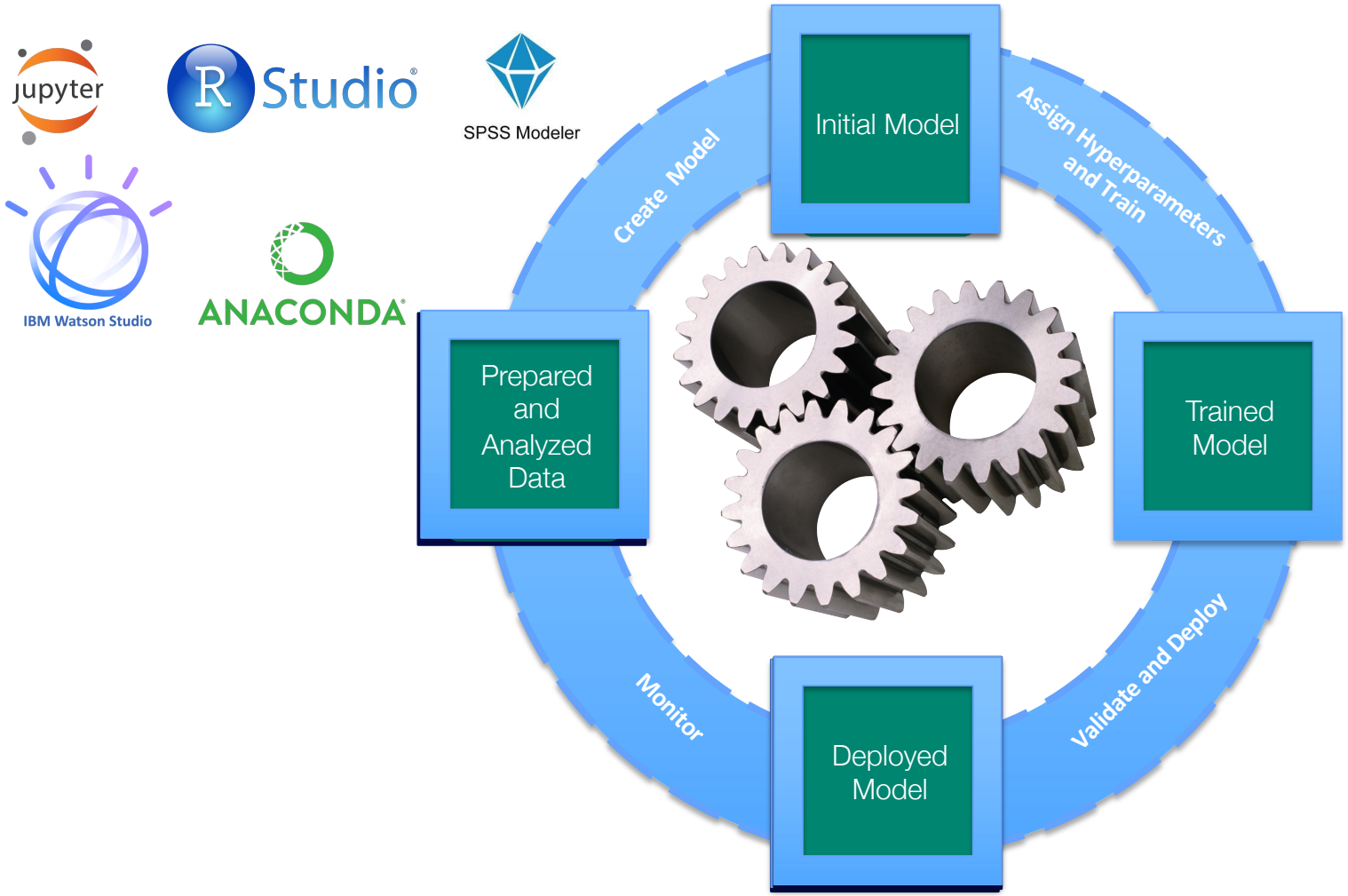
Let`s understand it from the context of an AI Lifecycle



We need a Cloud native AI Platform to build, train, deploy and monitor Models



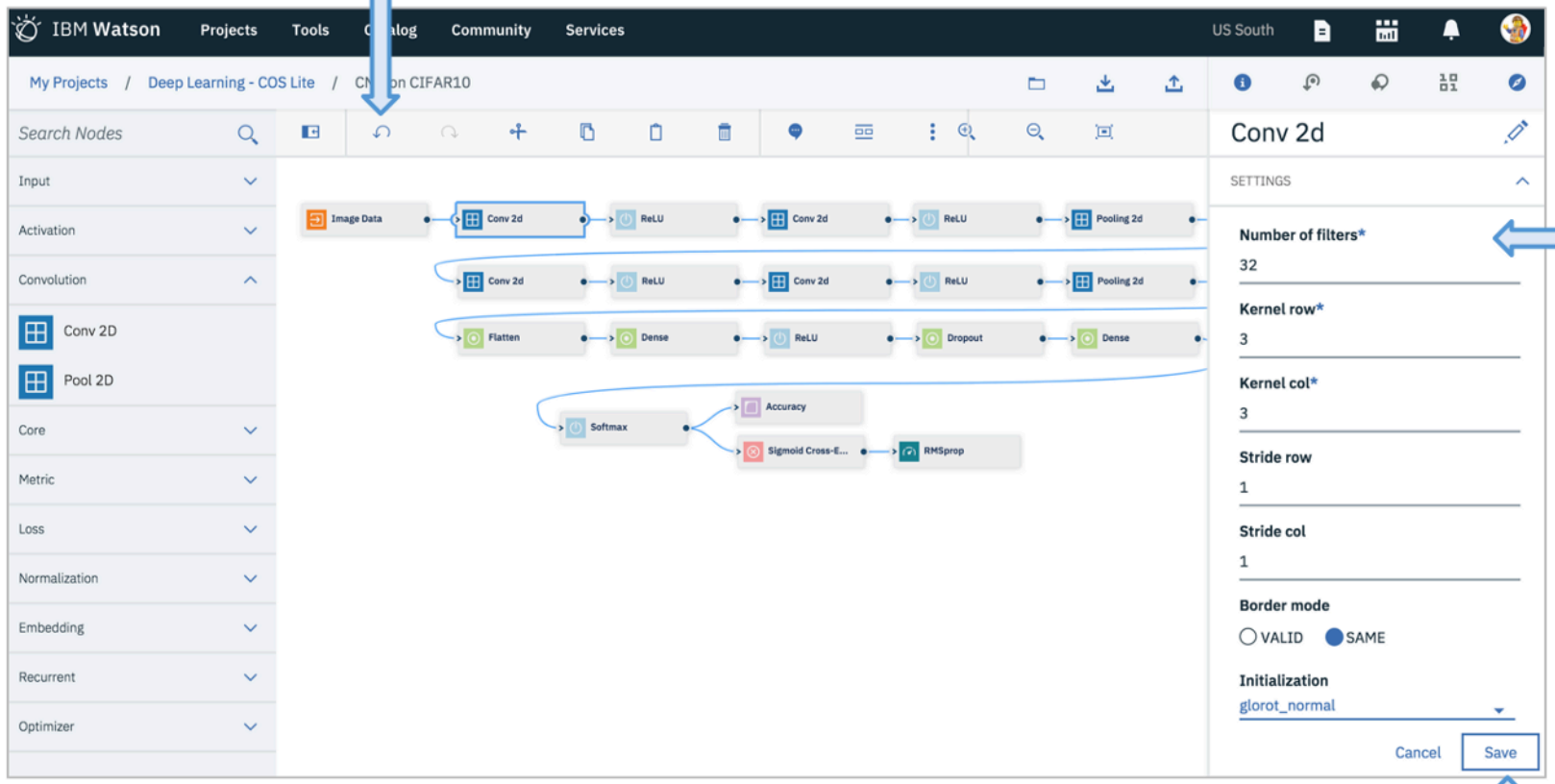
Many tools available to build initial models



Neural Network Modeller within Watson Studio

An intuitive drag-and-drop, no-code interface for designing neural network structure

Real-time validation of network flow



Drag-and-drop network layers

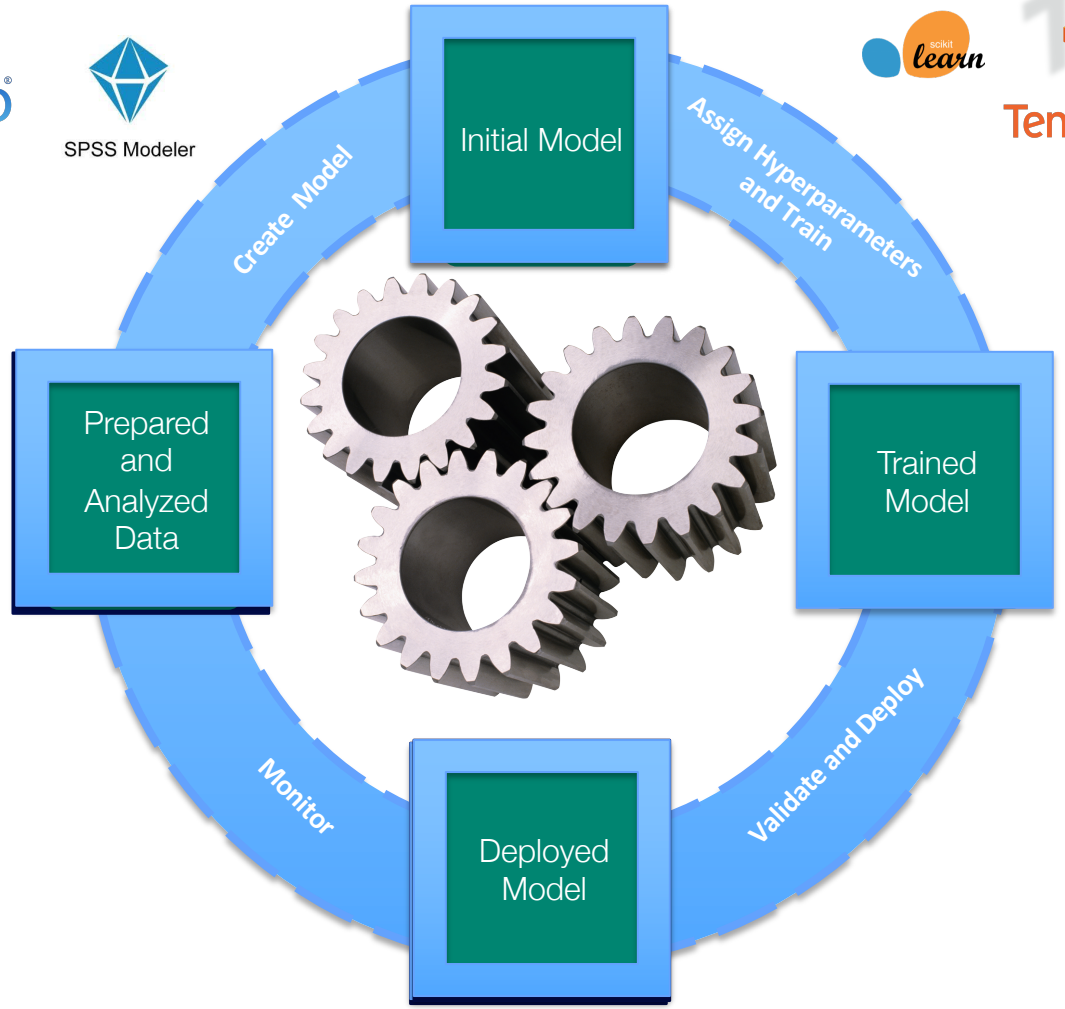
- Customize layer by setting hyperparameters

- Generate CPU or GPU compatible code

- Save as popular framework code
- Export as a python notebook
- Execute as batch experiment

IBM CODE

Many tools to train machine learning and deep learning models



jupyter

RStudio

SPSS Modeler

IBM Watson Studio

ANACONDA

scikit learn

TensorFlow

PaddlePaddle

PyTorch

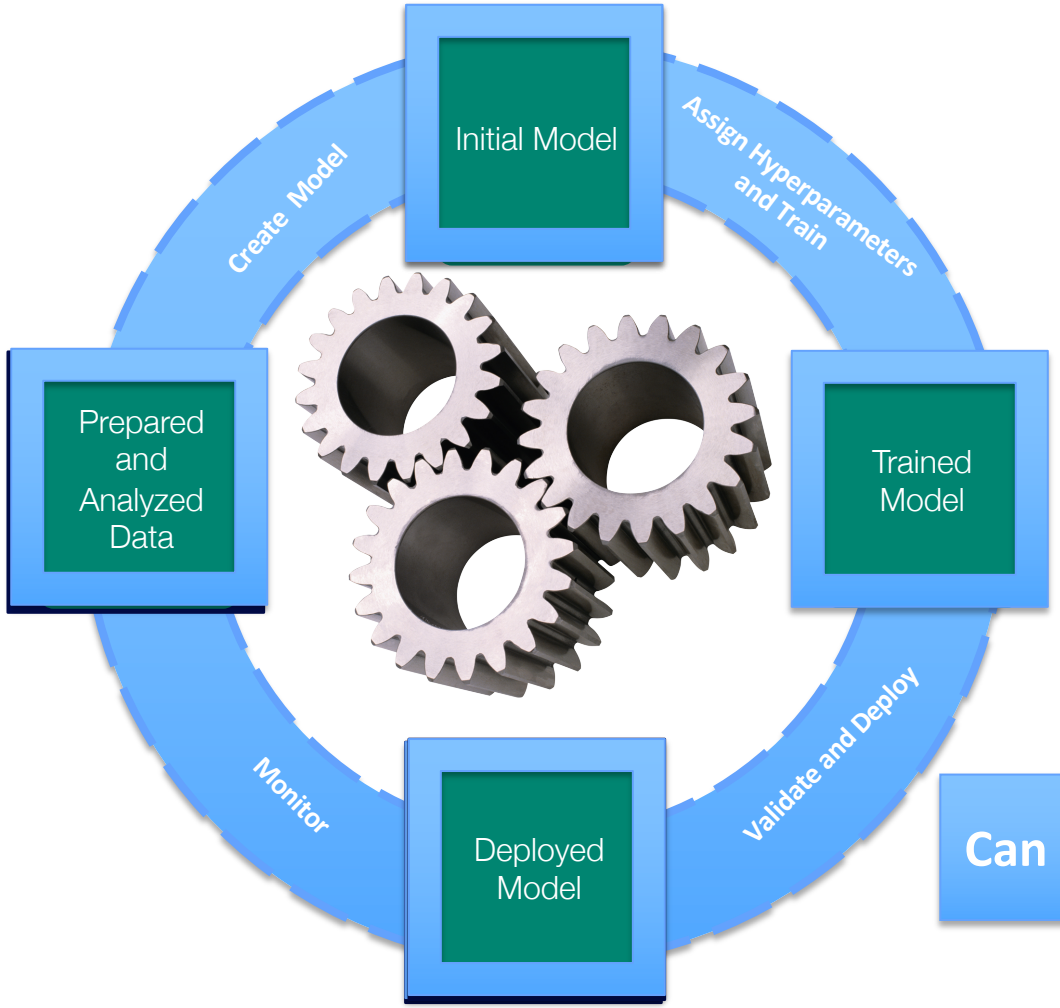
Keras

Caffe

mxnet

dmlc XGBoost

Training is accomplished. Model is ready – Can we trust it?



Can the model be trusted?

What does it take to trust a decision made by a machine?

(Other than that it is 99% accurate)?

Is it fair?

Is it easy to understand?

Did anyone tamper with it?

Is it accountable?

Our vision for Trusted AI

Pillars of trust, woven into the lifecycle of an AI application

FAIRNESS

EXPLAINABILITY

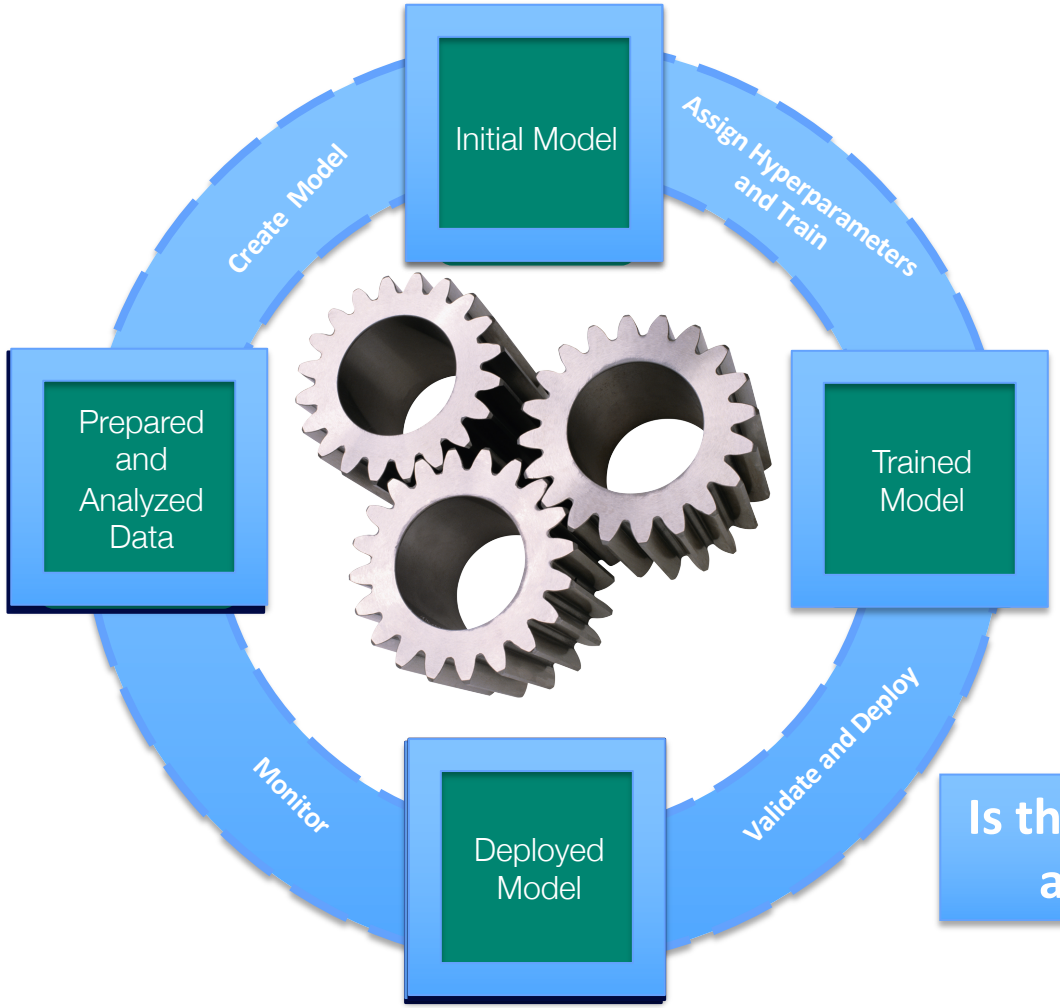
ROBUSTNESS

ASSURANCE

supported by an instrumented platform

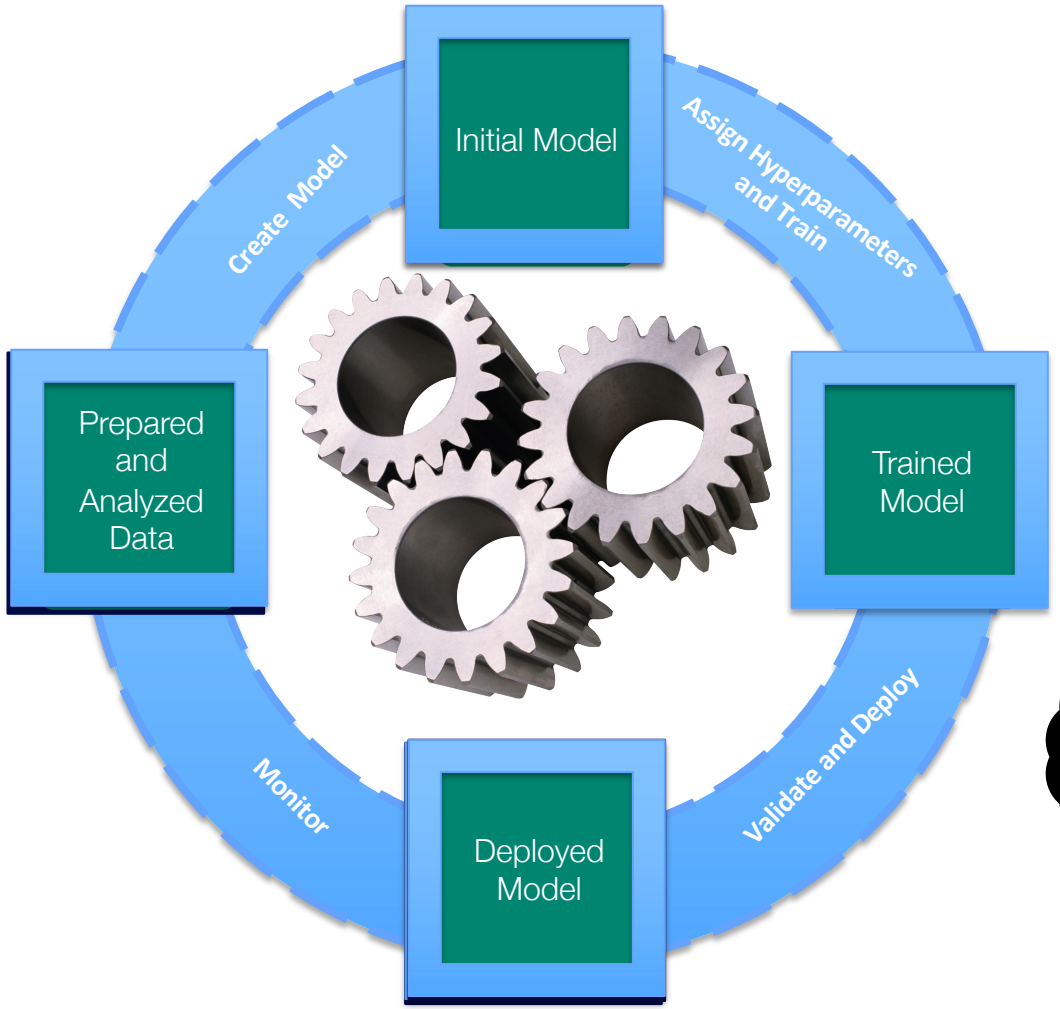
AIOpenScale

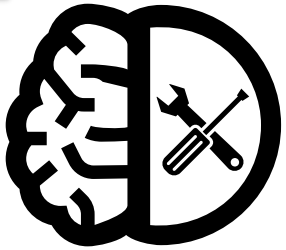
So let`s start with vulnerability detection of Models?



Is the model vulnerable to adversarial attacks?

Enter: Adversarial Robustness Toolbox





ART

IBM Adversarial Robustness Toolbox

https://github.com/IBM/adversarial-robustness-toolbox

ART is a library dedicated to adversarial machine learning. Its purpose is to allow rapid crafting and analysis of attack and defense methods for machine learning models. The Adversarial Robustness Toolbox provides an implementation for many state-of-the-art methods for attacking and defending classifiers.

ART

The Adversarial Robustness Toolbox contains implementations of the following attacks:

Deep Fool (Moosavi-Dezfooli et al., 2015)
Fast Gradient Method (Goodfellow et al., 2014)
Jacobian Saliency Map (Papernot et al., 2016)
Universal Perturbation (Moosavi-Dezfooli et al., 2016)
Virtual Adversarial Method (Moosavi-Dezfooli et al., 2015)
C&W Attack (Carlini and Wagner, 2016)
NewtonFool (Jang et al., 2017)

The following defense methods are also supported:

Feature squeezing (Xu et al., 2017)
Spatial smoothing (Xu et al., 2017)
Label smoothing (Warde-Farley and Goodfellow, 2016)
Adversarial training (Szegedy et al., 2013)

Virtual adversarial training (Miyato et al., 2017)

IBM CODE

Implementation for state-of-the-art methods for attacking and defending classifiers.

Evasion attacks

- FGSM
- JSMA
- BIM
- PGD
- Carlini & Wagner
- DeepFool
- NewtonFool
- Universal perturbation

Evasion defenses

- Feature squeezing
- Spatial smoothing
- Label smoothing
- Adversarial training
- Virtual adversarial training
- Thermometer encoding
- Gaussian data augmentation

Poisoning detection

- Detection based on clustering activations
- Proof of attack strategy

Evasion detection

- Detector based on inputs
- Detector based on activations

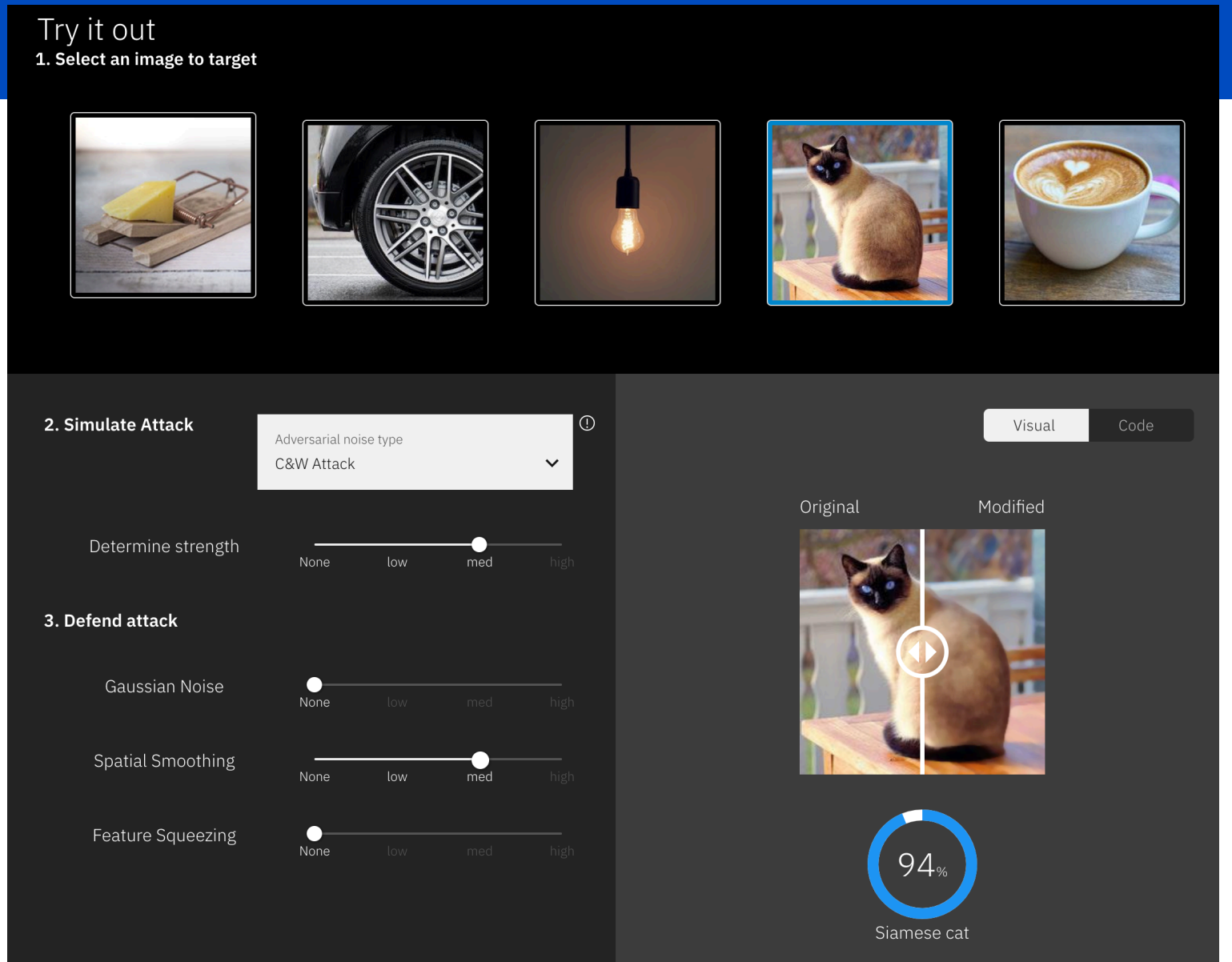
Robustness metrics

- CLEVER
- Empirical robustness
- Loss sensitivity

Unified model API

- Training
- Prediction
- Access to loss and prediction gradients

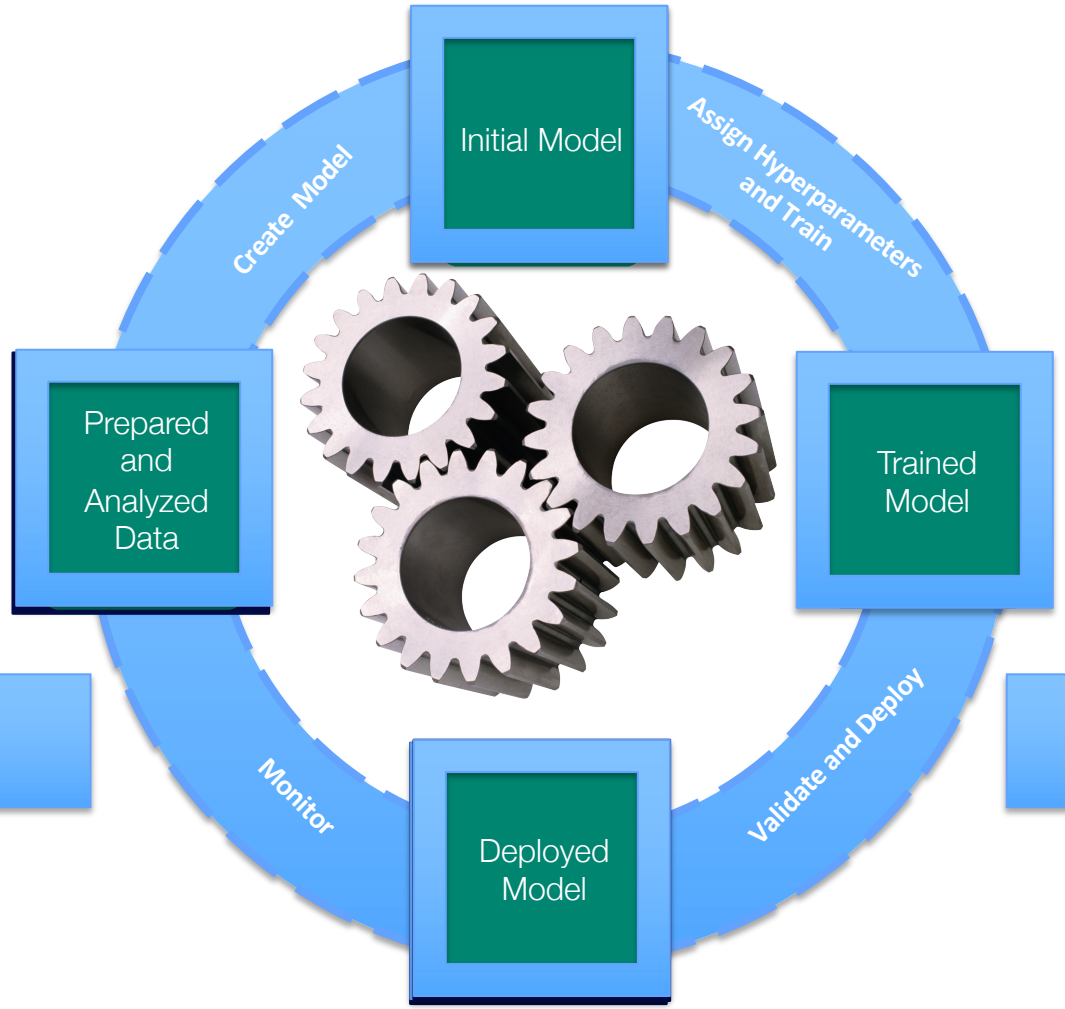
ART Demo: https://art-demo.mybluemix.net/



IBM CODE

Robustness check accomplished. How do we check for bias throughout lifecycle?

Is the dataset biased?



Are predictions biased?

Are model weights biased?

Unwanted bias and algorithmic fairness

Machine learning, by its very nature, is always a form of statistical discrimination

Discrimination becomes objectionable when it places certain privileged groups at systematic advantage and certain unprivileged groups at systematic disadvantage

Illegal in certain contexts

Unwanted bias and algorithmic fairness

Machine learning, by its very nature, is always a form of statistical discrimination

Unwanted bias in training data yields models with unwanted bias that scale out

Prejudice in labels

Undersampling or oversampling

AI Fairness 360

https://github.com/IBM/AIF360

AIF360 toolkit is an open-source library to help detect and remove bias in machine learning models.

The AI Fairness 360 Python package includes a comprehensive set of metrics for datasets and models to test for biases, explanations for these metrics, and algorithms to mitigate bias in datasets and models.

Toolbox

Fairness metrics (30+)
Fairness metric explanations
Bias mitigation algorithms (9+)

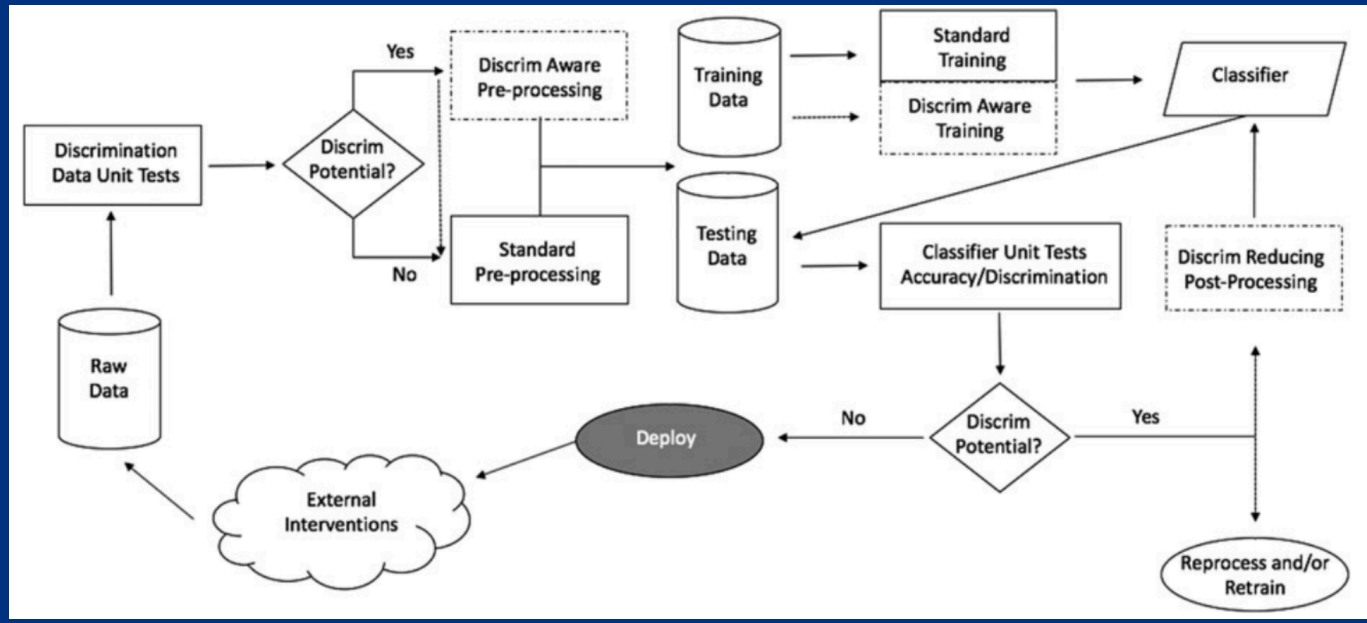
IBM CODE

Supported bias mitigation algorithms

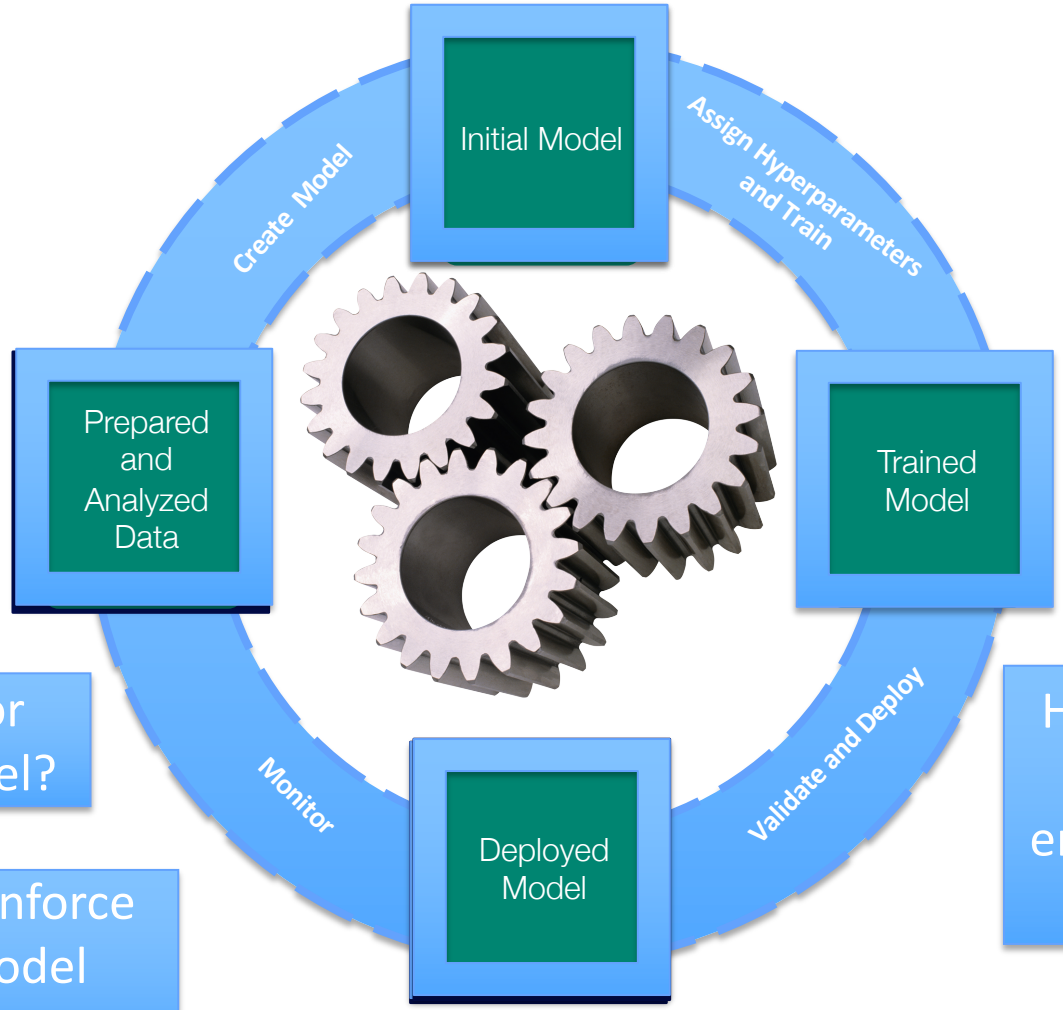
Optimized Preprocessing (Calmon et al., 2017)
Disparate Impact Remover (Feldman et al., 2015)
Equalized Odds Postprocessing (Hardt et al., 2016)
Reweighing (Kamiran and Calders, 2012)
Reject Option Classification (Kamiran et al., 2012)
Prejudice Remover Regularizer (Kamishima et al., 2012)
Calibrated Equalized Odds Postprocessing (Pleiss et al., 2017)
Learning Fair Representations (Zemel et al., 2013)
Adversarial Debiasing (Zhang et al., 2018)

Supported fairness metrics

Comprehensive set of group fairness metrics derived from selection rates and error rates
Comprehensive set of sample distortion metrics
Generalized Entropy Index (Speicher et al., 2018)



Model is trained, tested and validated. Then we can deploy it. Do we need anything else?



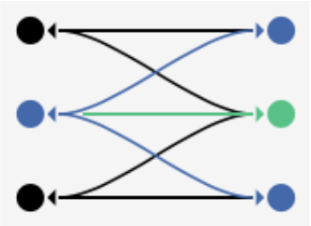
How can I monitor and trace the model?

How can add and enforce policies to the model easily?

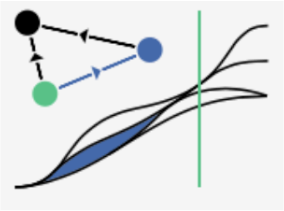
How can I manage multiple version of my model to enable dark launch, A/B test and traffic shift easily

Istio

An open service mesh platform to connect, observe, secure, and control microservices.

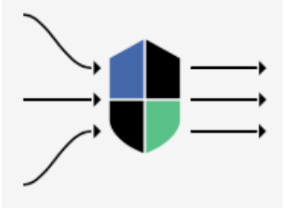


Connect: Traffic Control, Discovery, Load Balancing, Resiliency



Observe: Metrics, Logging, Tracing

Secure: Encryption (TLS), Authentication, and Authorization of service-to-service communication

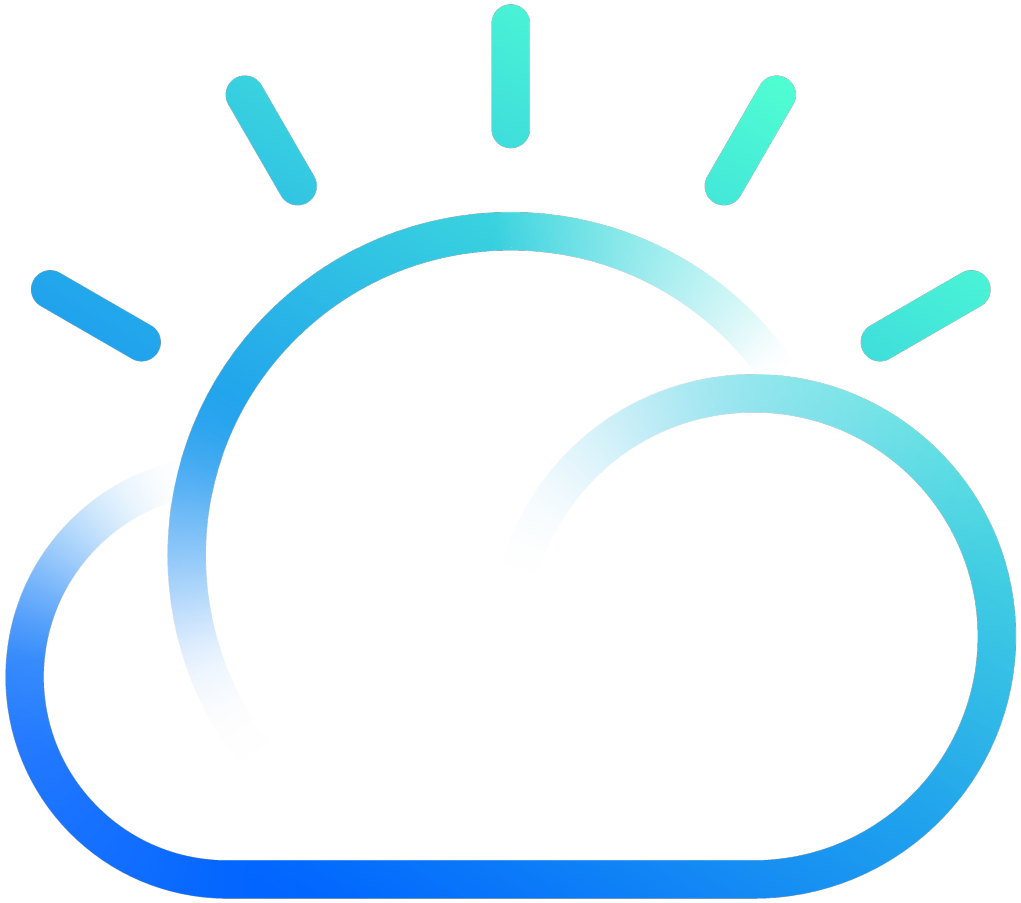


Control: Policy Enforcement

So AI in general and Deep Learning in particular are very iterative and repetitive.

And they need Cloud.

Why?

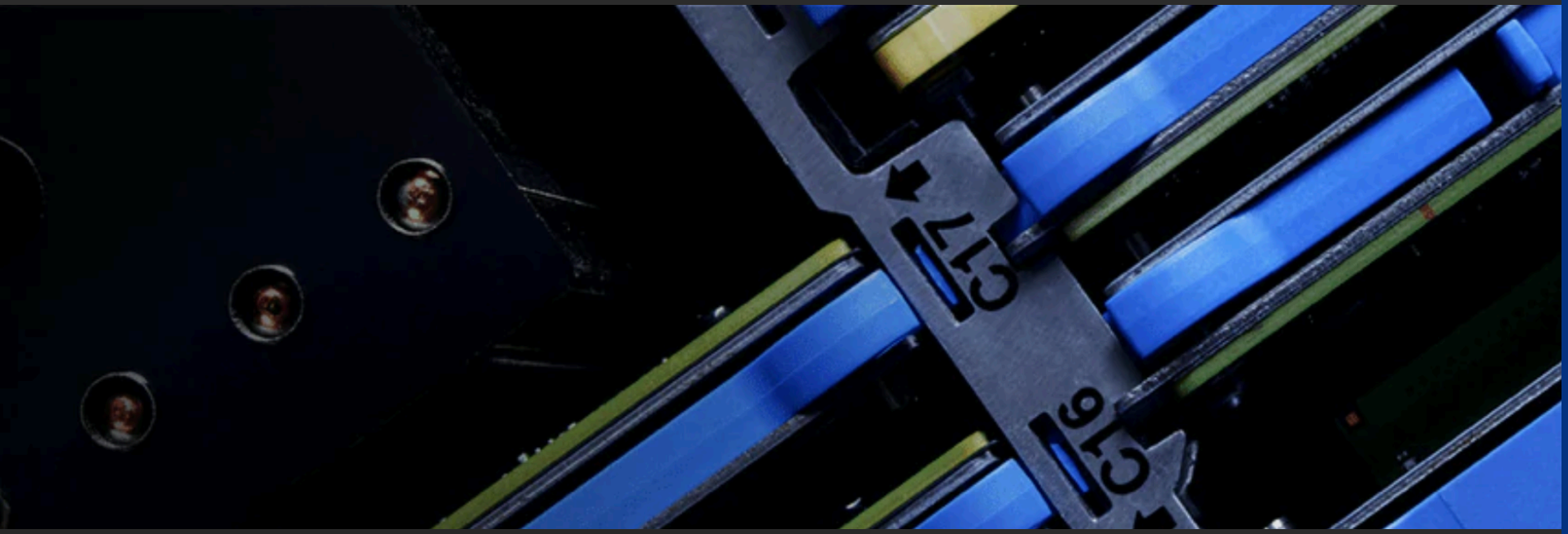


IBM Cloud

AI requires the strength of HPC & GPUs

Ability to scale AI workloads on demand

Ability to utilize various technologies and achieve high performance computing.



1. Model/Data Parallelism
2. MPI/NCCL

NCCL (pronounced "Nickel") is a stand-alone library of standard collective communication routines for GPUs, implementing all-reduce, all-gather, reduce, broadcast, and reduce-scatter. It has been optimized to achieve high bandwidth on platforms using PCIe, NVLink, NVswitch, as well as networking using InfiniBand Verbs or TCP/IP sockets.

NCCL supports an arbitrary number of GPUs installed in a single node or across multiple nodes, and can be used in either single- or multi-process (e.g., MPI) applications.

To scale we need to go Cloud native for AI

Microservices

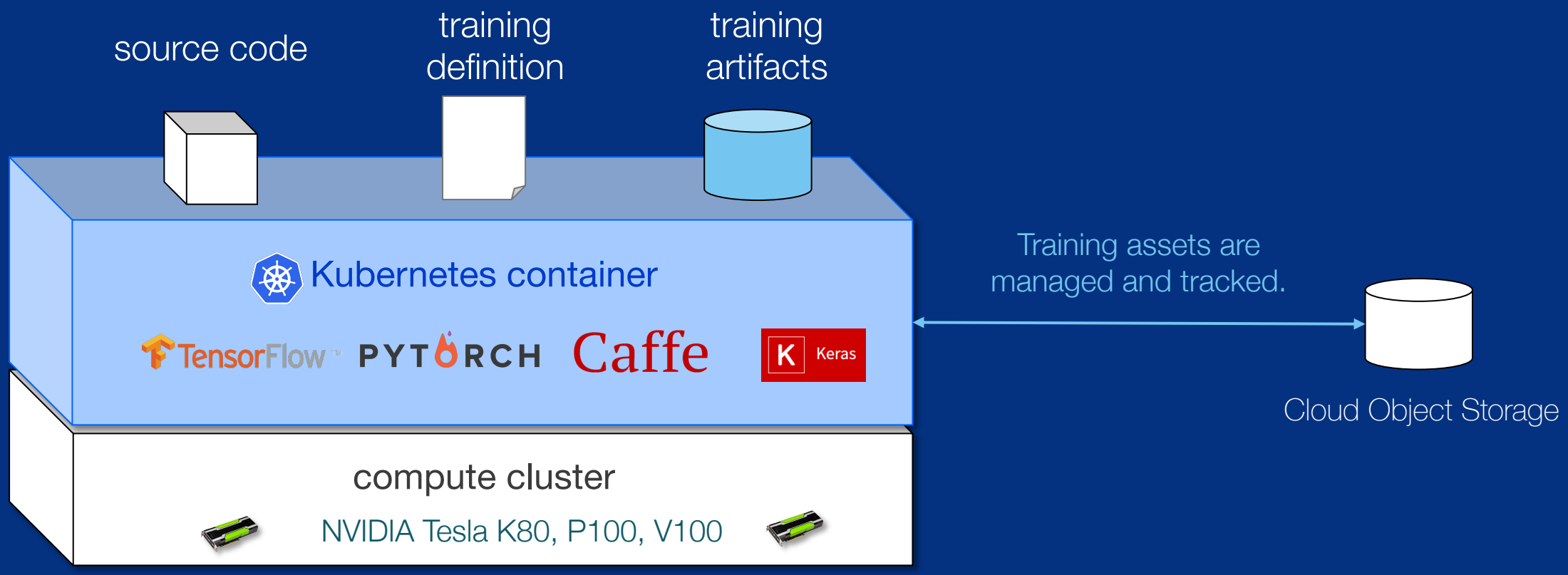
Containers

DevOps automation

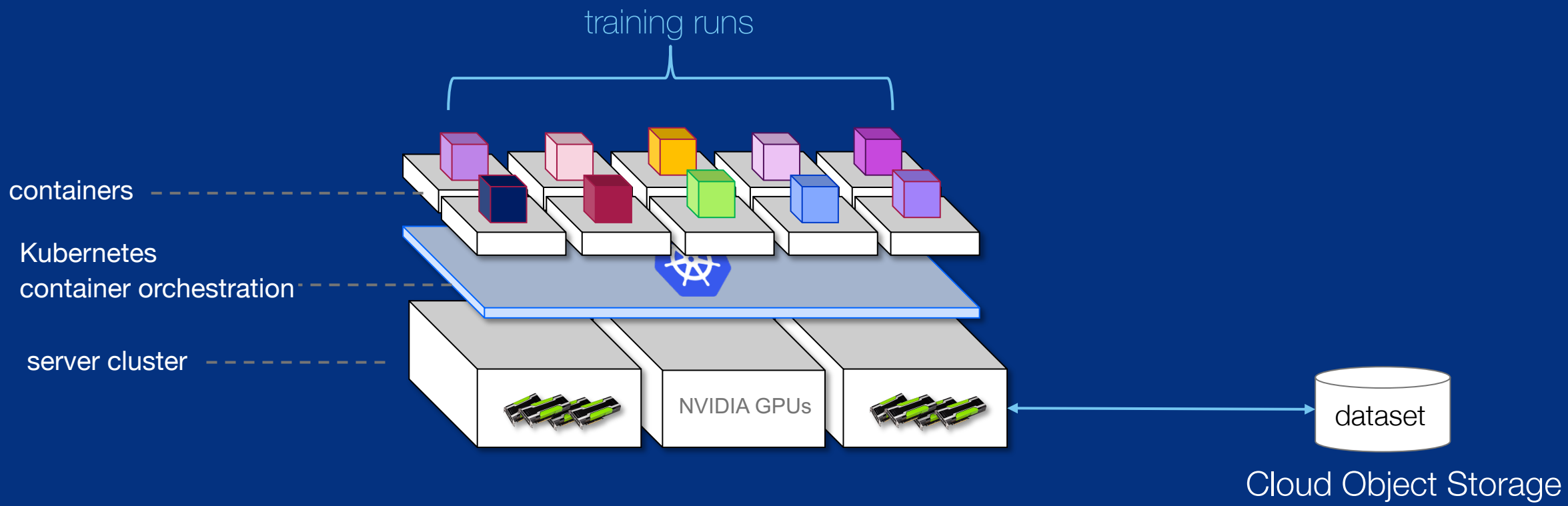
IBM CODE

Access to elastic compute leveraging Kubernetes

Auto-allocation means infrastructure is used only when needed



Model training distributed across containers



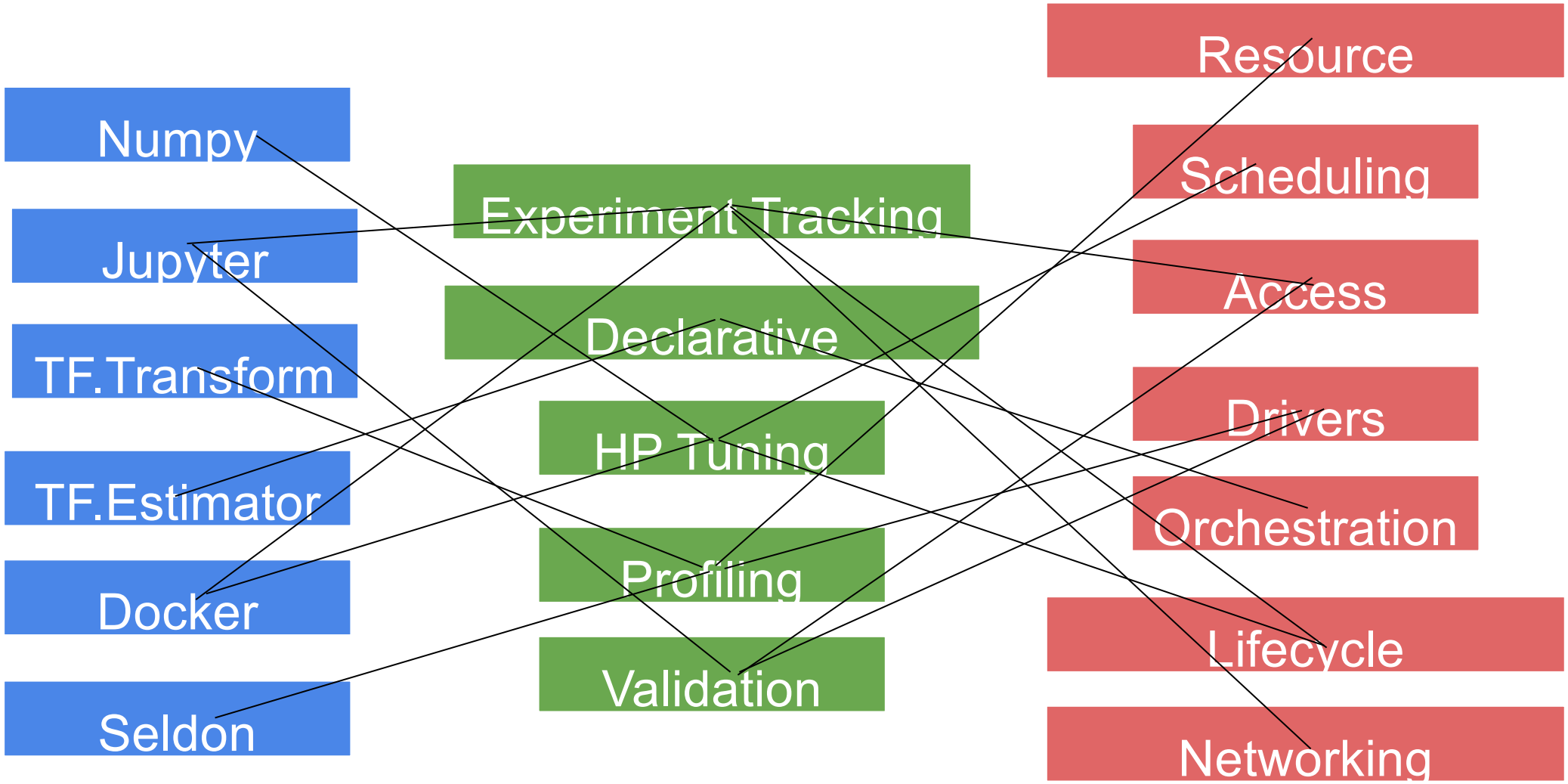
Kubernetes is not the end game

Kelsey Hightower 

@kelseyhightower

Kubernetes is a platform for building platforms. It's a better place to start; not the endgame.

 556 1:04 PM - Nov 27, 2017



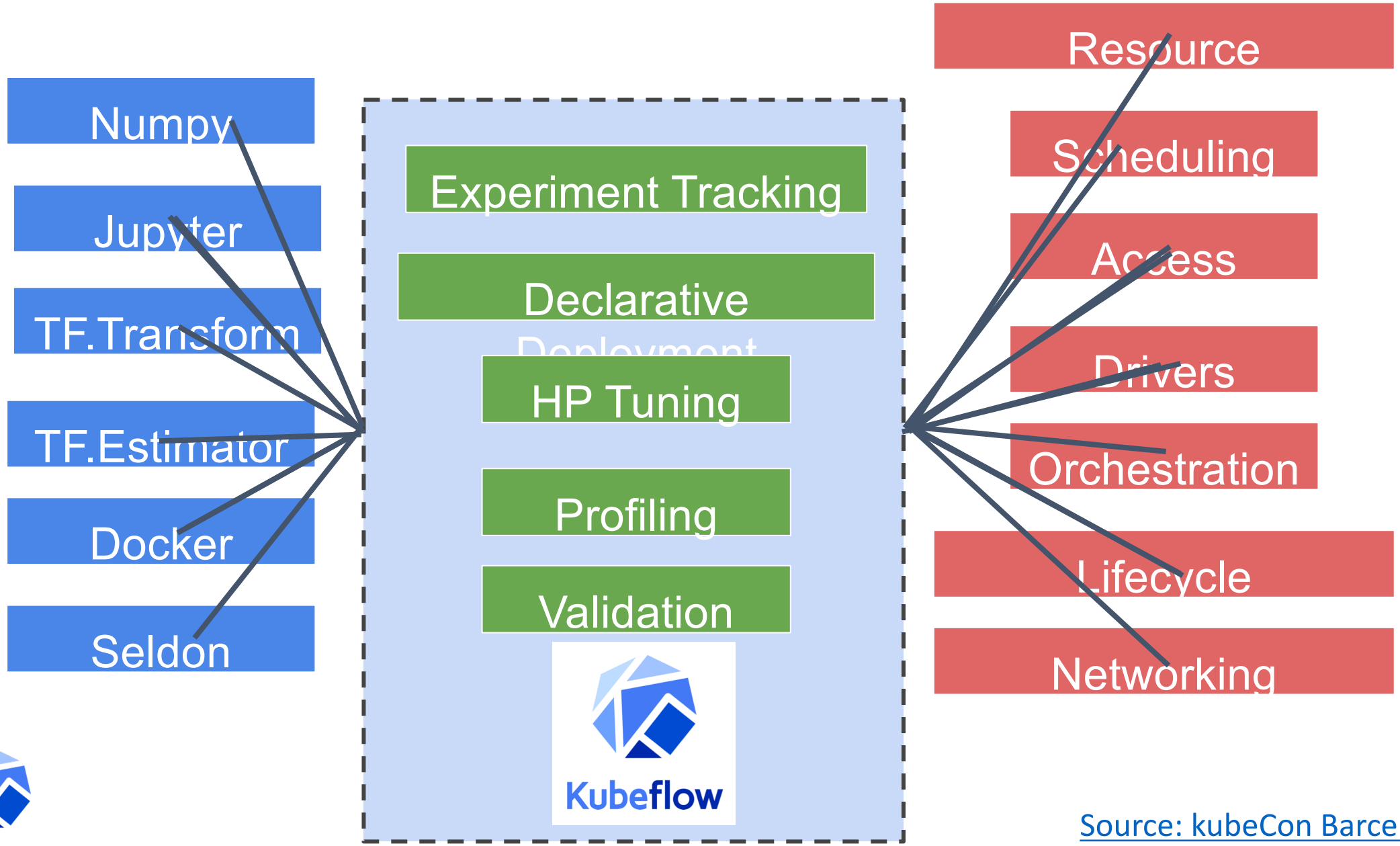
Source: kubeCon Barcelona 2019

Oh, you want to use ML on K8s?

First, can you become an expert in ...

- Containers
- Packaging
- Kubernetes service endpoints
- Persistent volumes
- Scaling
- Immutable deployments
- GPUs, Drivers & the GPL
- Cloud APIs
- DevOps
- ...

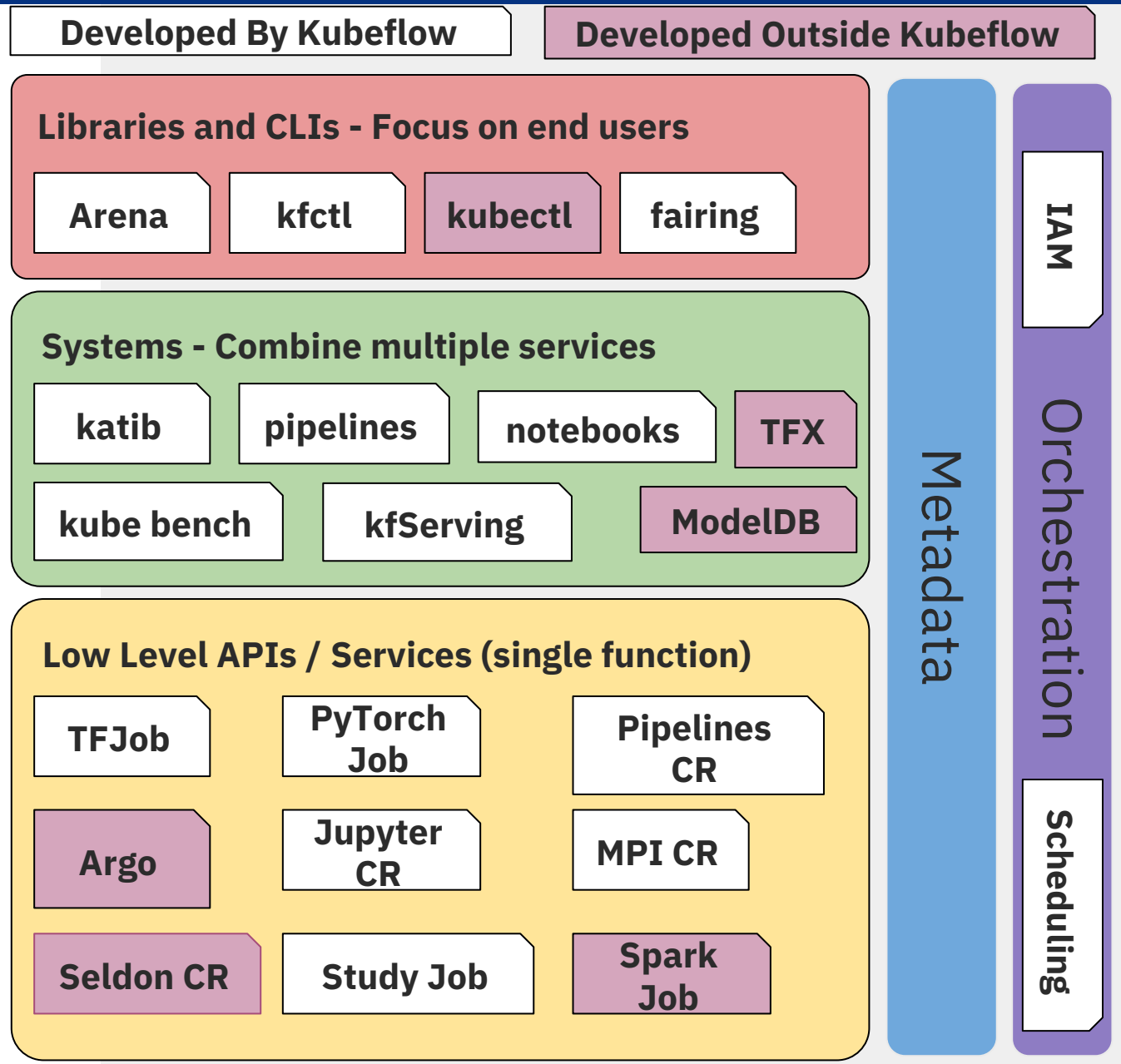
[Source: kubeCon Barcelona 2019]



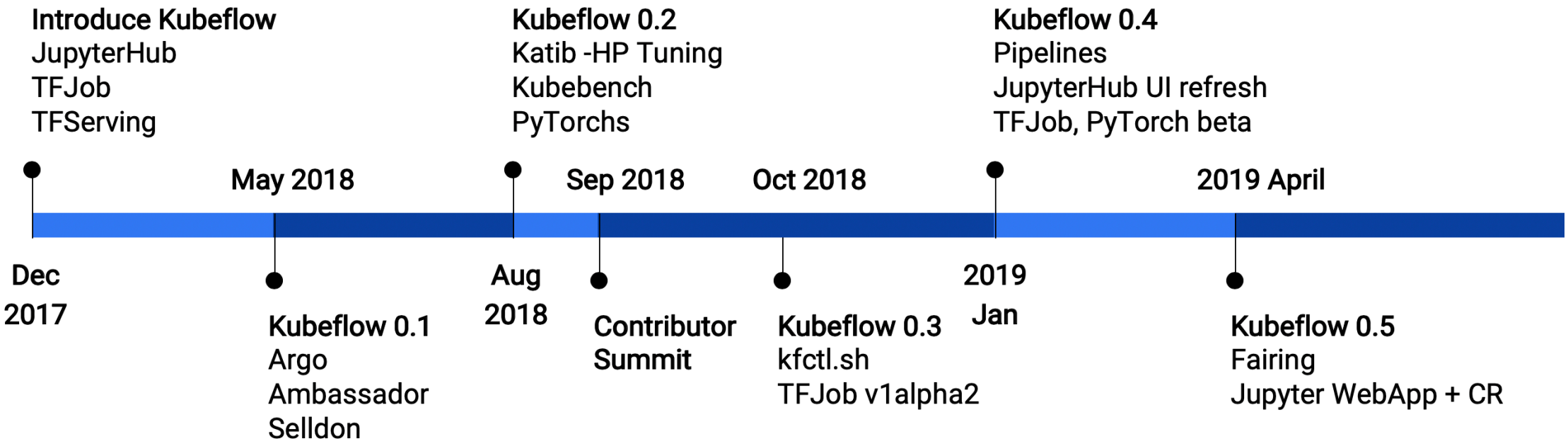
Source: kubeCon Barcelona 2019

Kubeflow architecture

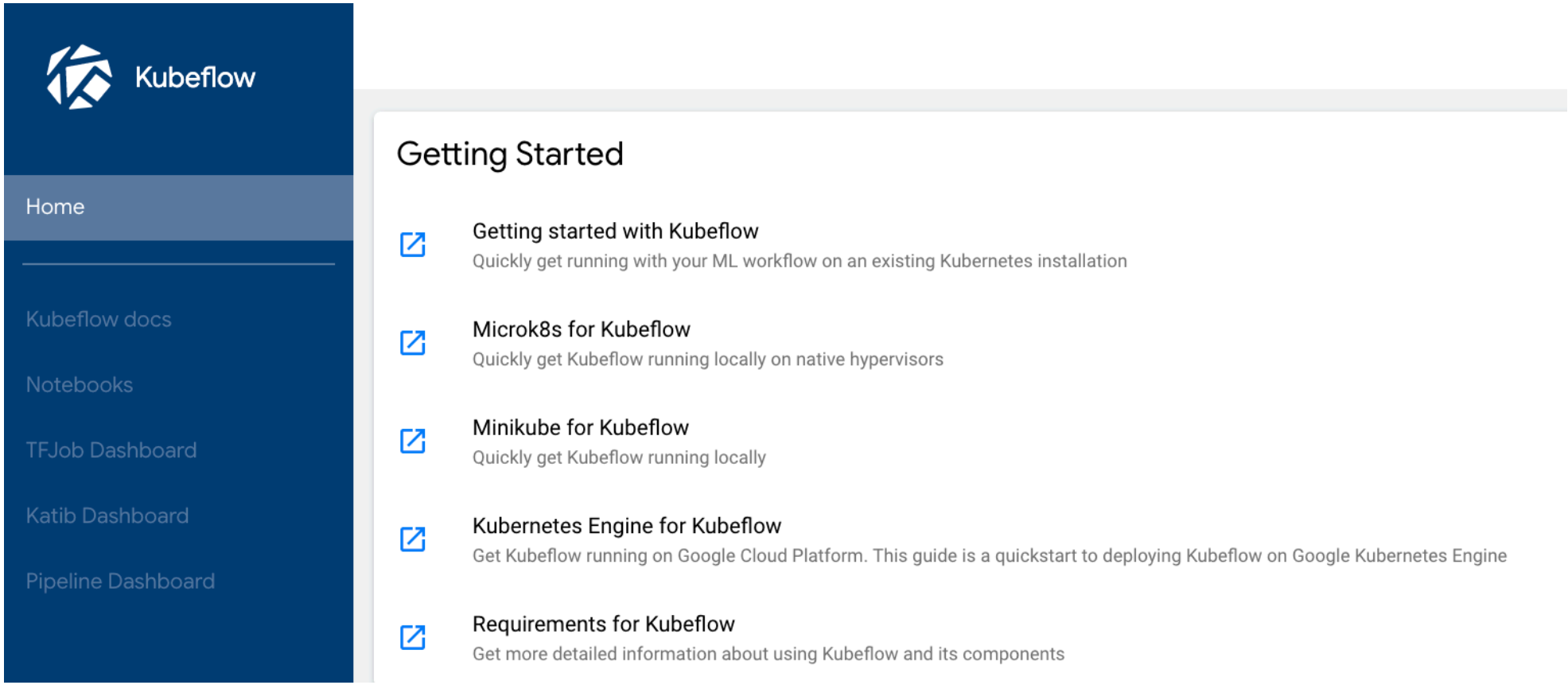
- Make it super easy to deploy and administer a platform
  - Leverage KF & non KF components
- Tie it together using
  - Orchestration
    - Combine components into complex workflows
  - Metadata
    - Collect data from multiple components



kubeflow


Introduce Kubeflow
JupyterHub
TFJob
TFServing
Dec 2017
May 2018
Kubeflow 0.1
Argo
Ambassador
Selldon
Aug 2018
Kubeflow 0.2
Katib -HP Tuning
Kubebench
PyTorchs
Sep 2018
Contributor Summit
Oct 2018
Kubeflow 0.3
kfctl.sh
TFJob v1alpha2
2019 Jan
Kubeflow 0.4
Pipelines
JupyterHub UI refresh
TFJob, PyTorch beta
2019 April
Kubeflow 0.5
Fairing
Jupyter WebApp + CR

kubeflow

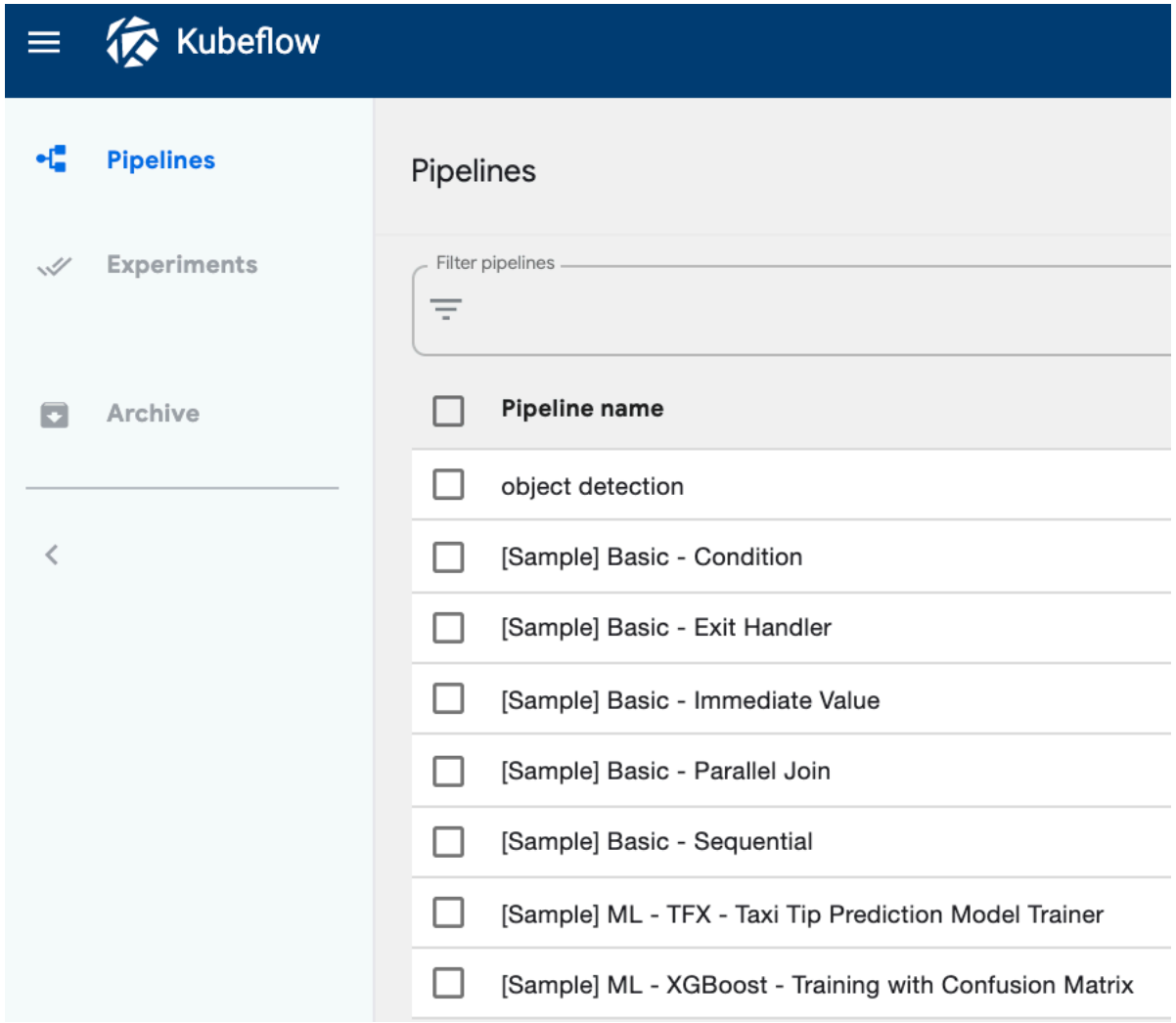

Kubeflow
Home
Kubeflow docs
Notebooks
TFJob Dashboard
Katib Dashboard
Pipeline Dashboard
Getting Started
Getting started with Kubeflow
Quickly get running with your ML workflow on an existing Kubernetes installation
Microk8s for Kubeflow
Quickly get Kubeflow running locally on native hypervisors
Minikube for Kubeflow
Quickly get Kubeflow running locally
Kubernetes Engine for Kubeflow
Get Kubeflow running on Google Cloud Platform. This guide is a quickstart to deploying Kubeflow on Google Kubernetes Engine
Requirements for Kubeflow
Get more detailed information about using Kubeflow and its components

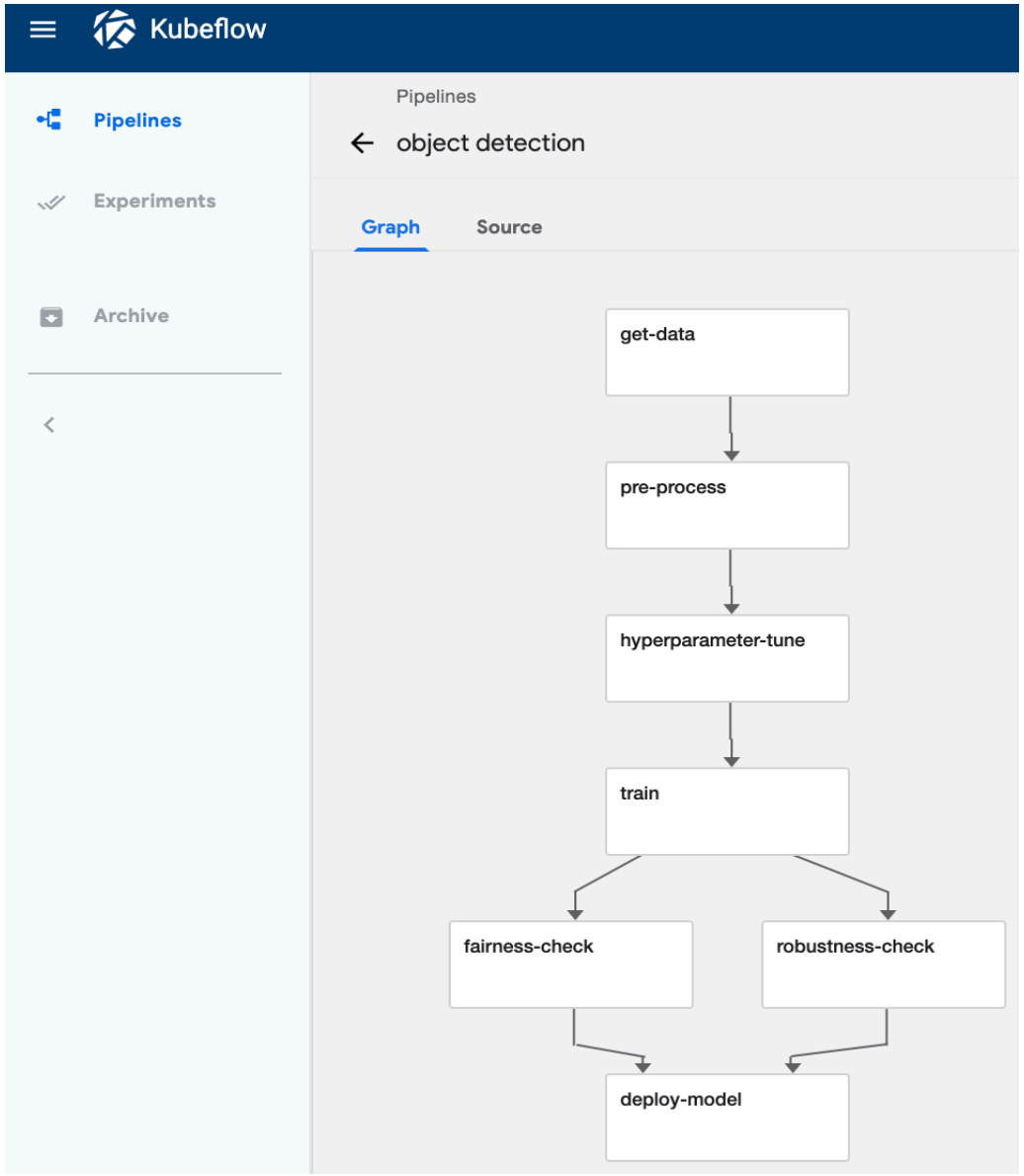
Kubeflow pipeline

```
@dsl.pipeline(
  name='Object detection',
  description='Object detection'
)
def object_detection(worker=3):
  getData = get_data()
  pre_process = pre_process(getData.output)
  hpo = hyperparameter_tune(pre_process.output)
  train = start_train(hpo.output, worker)
  r_check = robustness_check(train.output)
  f_check = fairness_check(train.output)
  deploy = deploy_model(r_check.output, f_check.output)
```

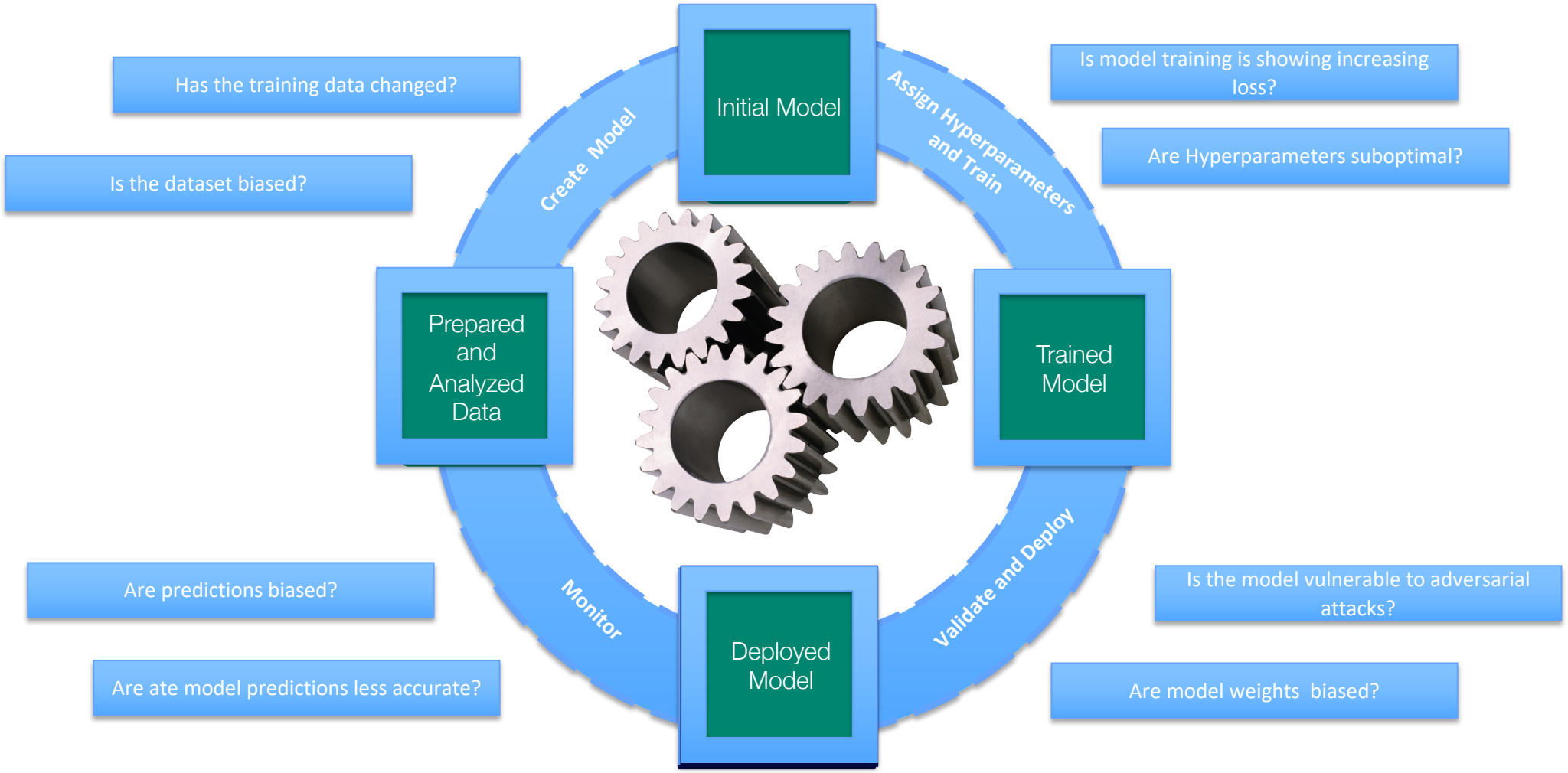
```
# dsl-compile --py object_detection.py --output object_detection.tgz
```

Kubeflow pipeline

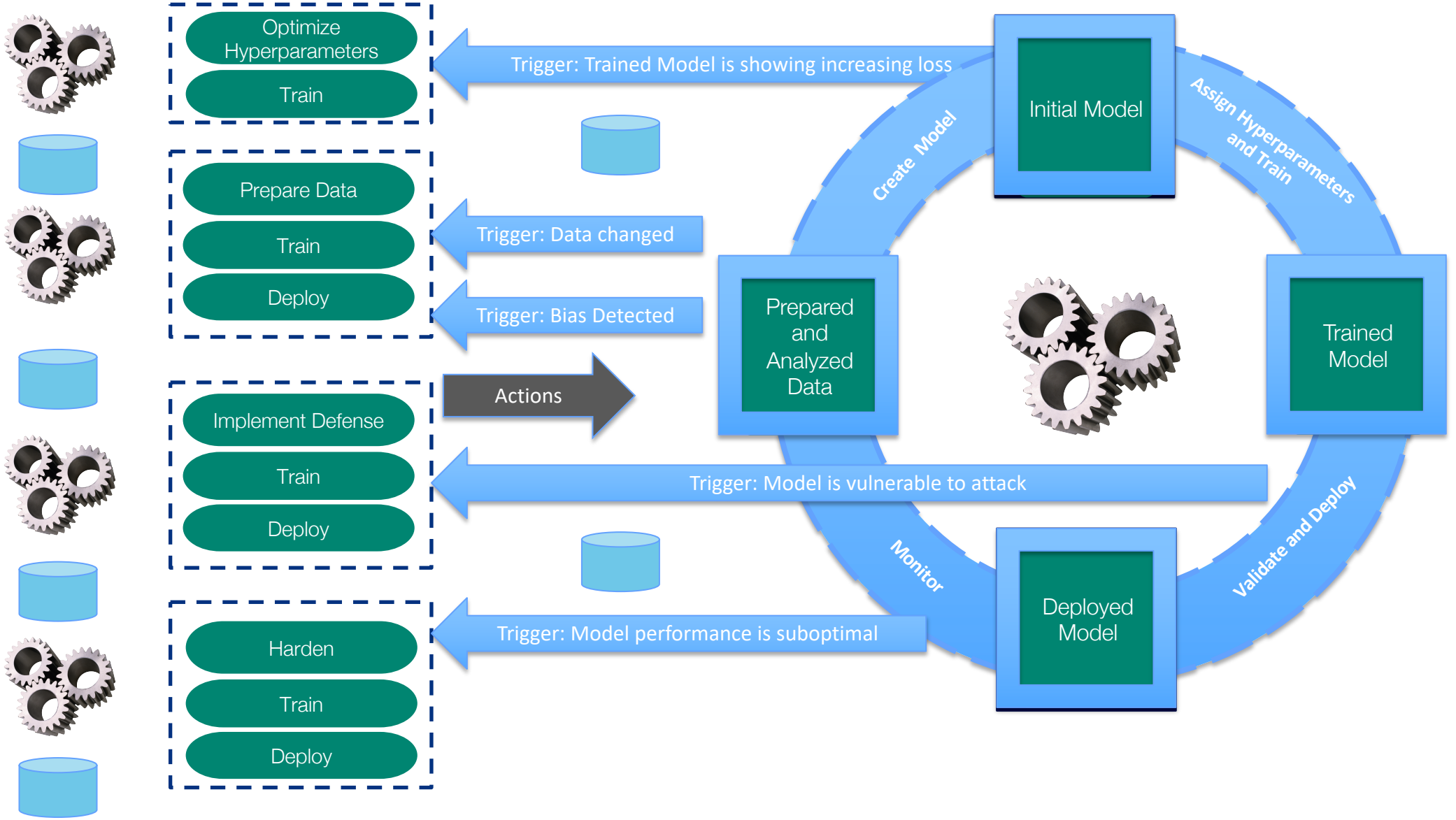




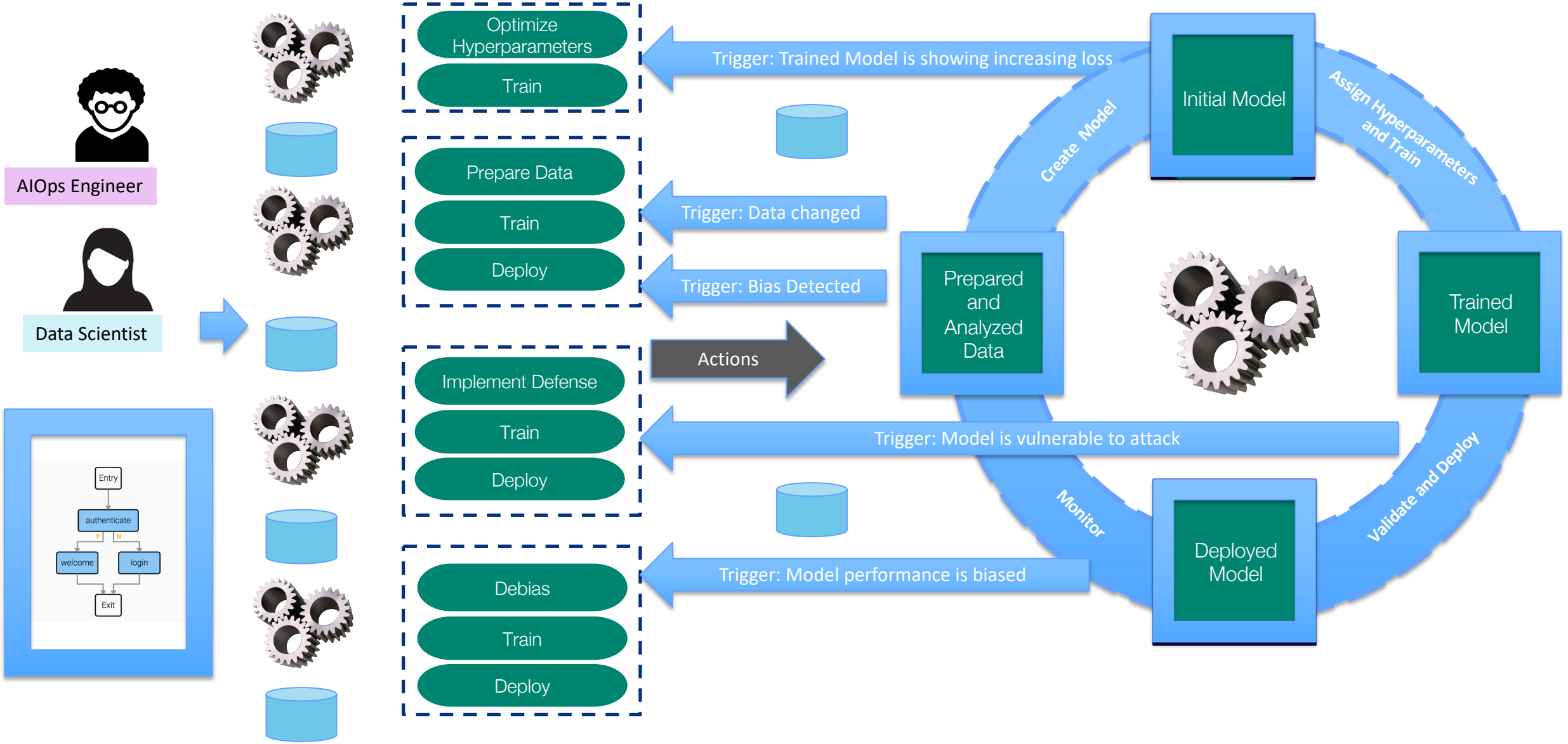
Infact, we need a transparent, trusted and automated AI Pipeline



Transparent, trusted, automated, event driven and auditable AI Pipeline



Transparent, trusted, automated, event driven, auditable AI Pipeline as a Service



Knative

Kubernetes-based platform to build, deploy, and manage modern serverless workloads.

Build

Provides easy-to-use, simple source-to-container builds, so you can focus on writing code and know how to build it. Knative solves for the common challenges of building containers and runs it on cluster.

Serving

Run serverless containers on Kubernetes with ease, Knative takes care of the details of networking, autoscaling (even to zero), and revision tracking. You just have to focus on your core logic.

Eventing

Universal subscription, delivery, and management of events. Build modern apps by attaching compute to a data stream with declarative event connectivity and developer-friendly object model.

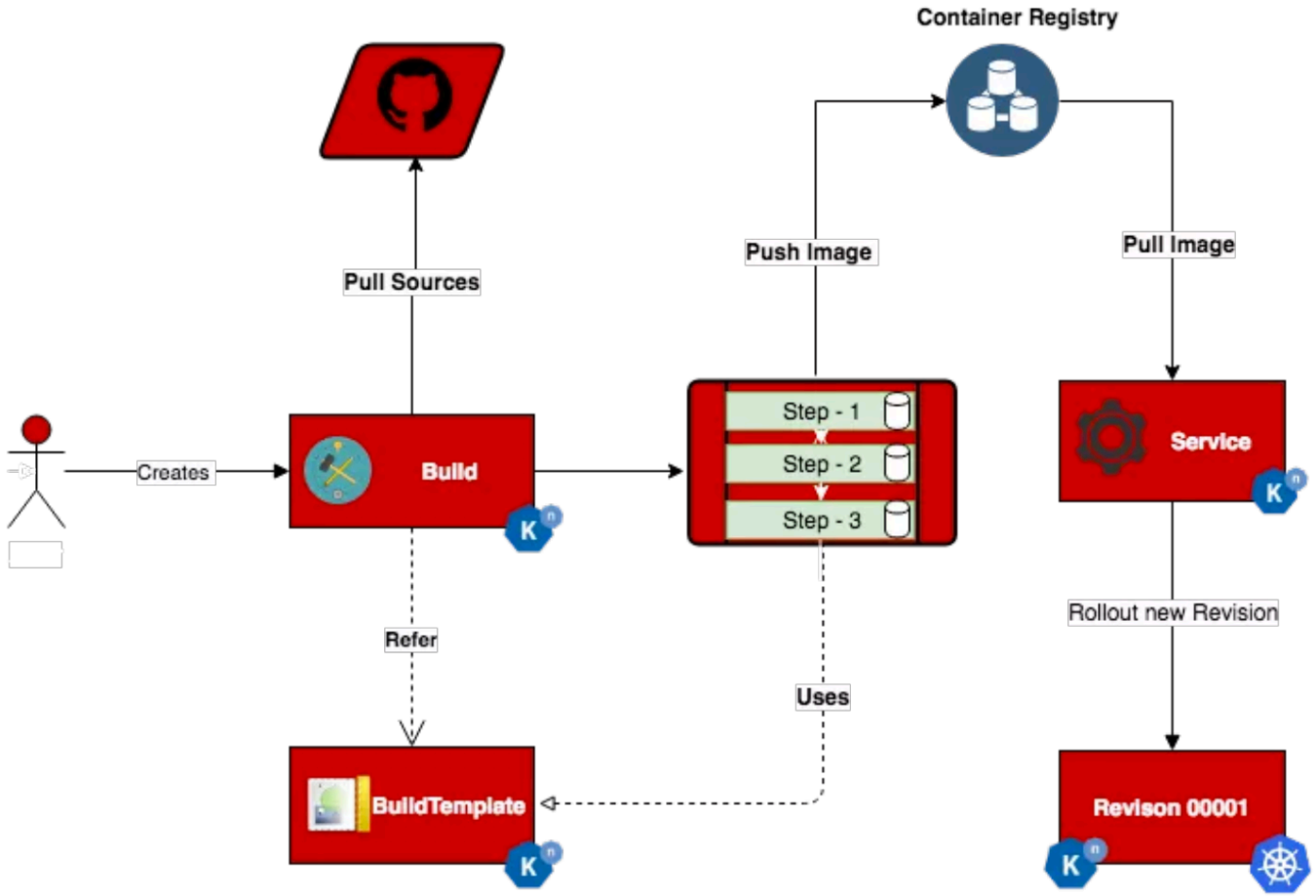
Knative build

Build — Source-to-container build orchestration

Knative Build Components

- Build
- Builder
- BuildTemplate

For example, you can write a build that uses Kubernetes-native resources to obtain your source code from a repository, build a container image, then run that image.



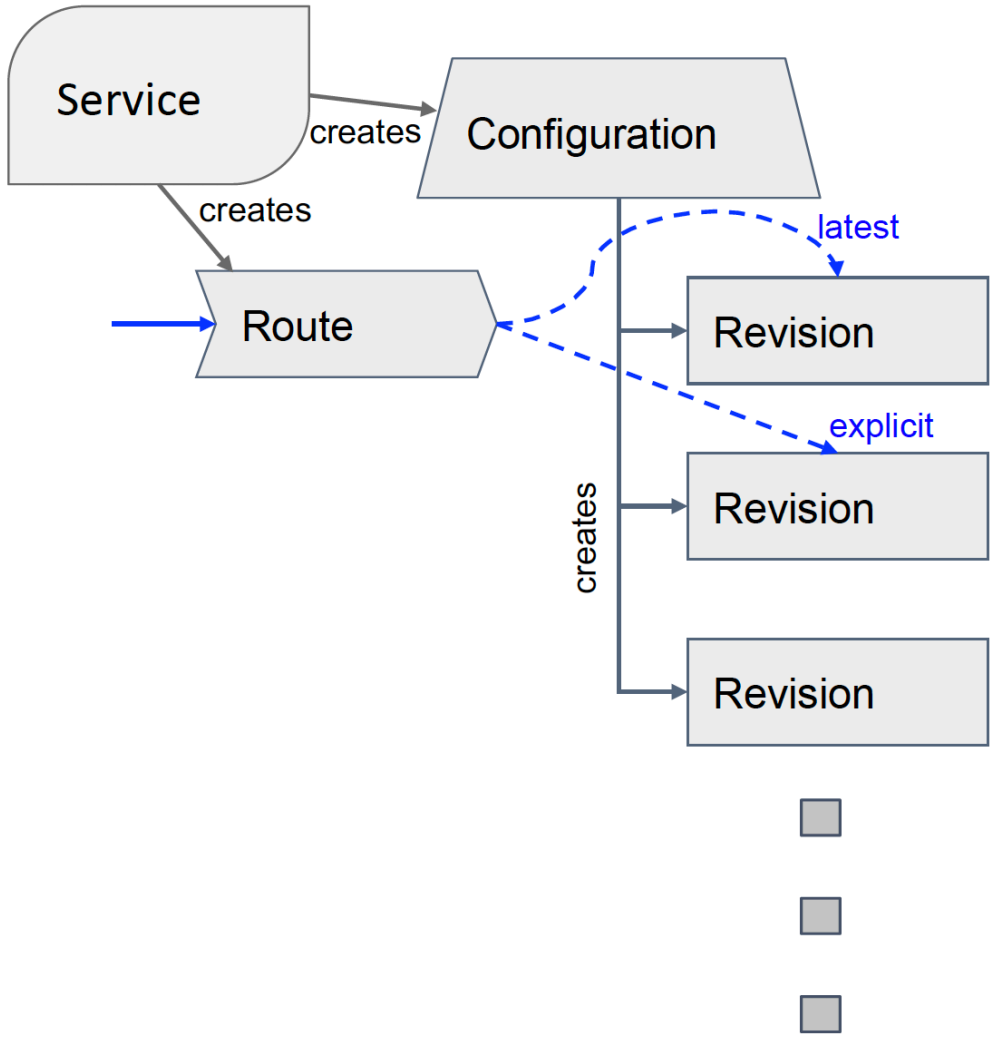
- A Build can include multiple steps where each step specifies a Builder.
- A Builder is a type of container image that you create to accomplish any task, whether that's a single step in a process, or the whole process itself.
- The steps in a Build can push to a repository.
- A BuildTemplate can be used to defined reusable parameterized templates.

Knative serving

Serving — Request-driven compute model, scale to zero, autoscaling, routing and managing traffic

Knative Serving components

- Configuration
  - Desired current state of deployment (#HEAD)
  - Records both code and configuration (separated, ala 12 factor)
  - Stamps out builds / revisions as it is updated
- Revision
  - Code and configuration snapshot
  - k8s infra: Deployment, ReplicaSet, Pods, etc
- Route
  - Traffic assignment to Revisions (fractional scaling or by name)
  - Built using Istio
- Service
  - Provides a simple entry point for UI and CLI tooling to achieve common behavior
  - Acts as a top-level controller to orchestrate Route and Configuration.



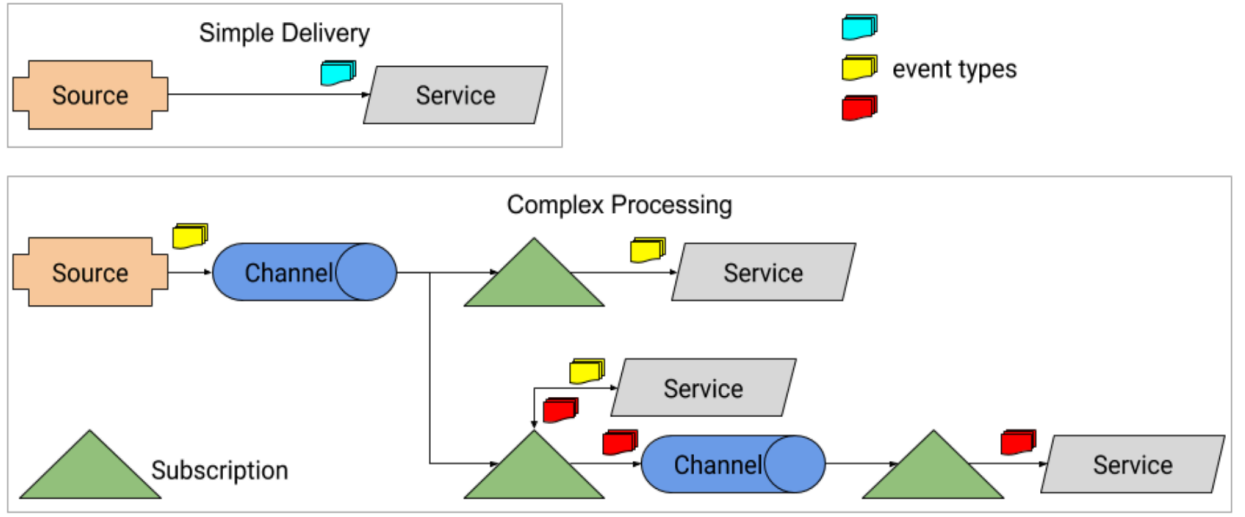
Knative eventing

Broker and **Trigger** are CRDs providing an event delivery mechanism that hides the details of event routing from the event producer and event consumer.

The **Event Registry** maintains a catalog of the event types that can be consumed from the different Brokers

Event Sources are Kubernetes Custom Resources which provide a mechanism for registering interest in a class of events from a particular software system.

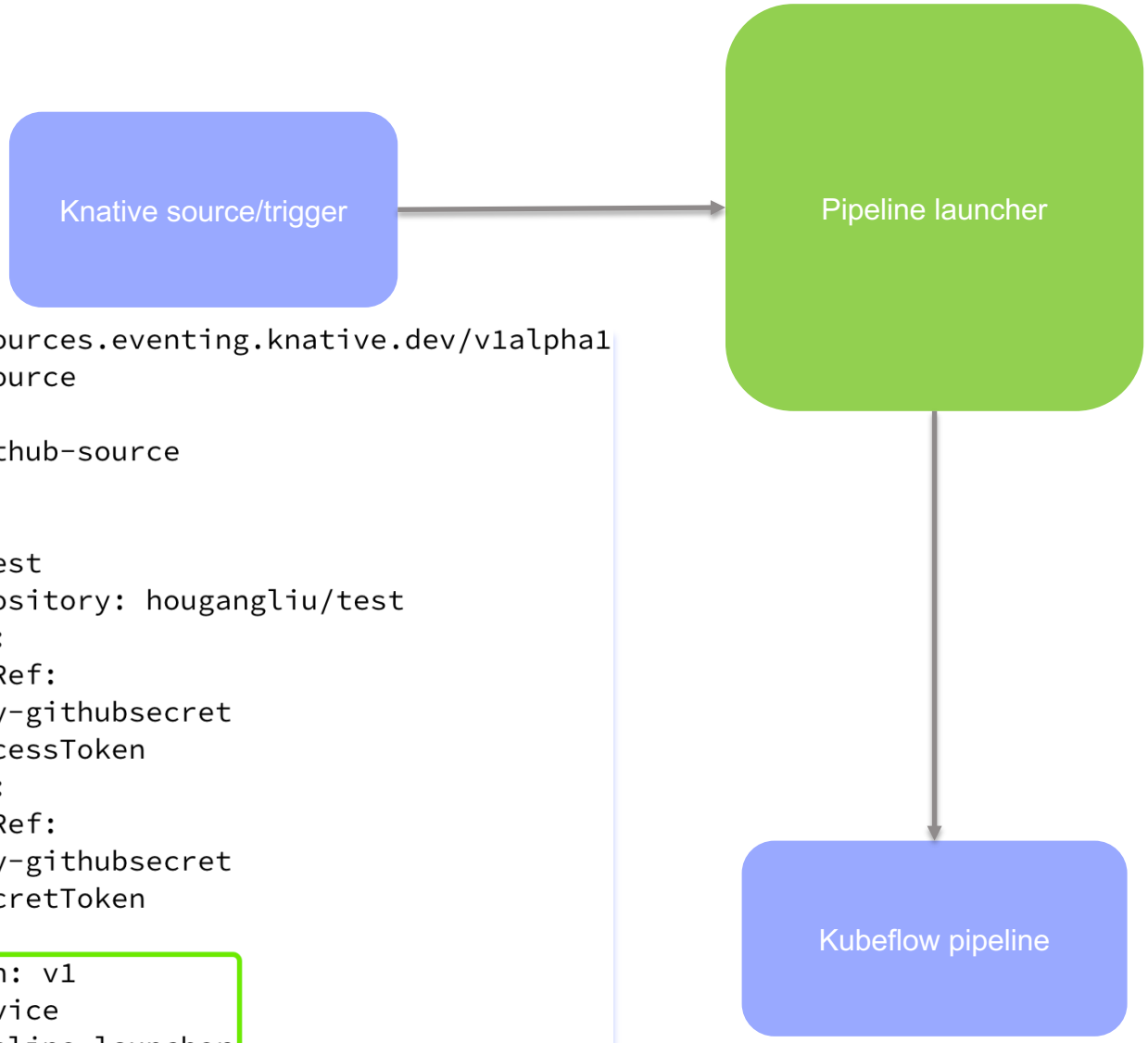
Channels are Kubernetes Custom Resources which define a single event forwarding and persistence layer.



Knative eventing

Name	Status	Support	Description
AWS SQS	Proof of Concept	None	Brings AWS Simple Queue Service messages into Knative.
Apache Camel	Proof of Concept	None	Allows to use Apache Camel components for pushing events into Knative.
Apache Kafka	Proof of Concept	None	Brings Apache Kafka messages into Knative.
BitBucket	Proof of Concept	None	Registers for events of the specified types on the specified BitBucket organization/repository. Brings those events into Knative.
Cron Job	Proof of Concept	None	Uses an in-memory timer to produce events on the specified Cron schedule.
GCP PubSub	Proof of Concept	None	Brings GCP PubSub messages into Knative.
GitHub	Proof of Concept	None	Registers for events of the specified types on the specified GitHub organization/repository. Brings those events into Knative.
GitLab	Proof of Concept	None	Registers for events of the specified types on the specified GitLab repository. Brings those events into Knative.
Google Cloud Scheduler	Active Development	None	Create, update, and delete Google Cloud Scheduler Jobs. When those jobs are triggered, receive the event inside Knative.
Google Cloud Storage	Active Development	None	Registers for events of the specified types on the specified Google Cloud Storage bucket and optional object prefix. Brings those events into Knative.
Kubernetes Api Server	Active Development	Knative	Brings Kubernetes resource changes into Knative as references or as full resources.

Event Driven ML pipeline



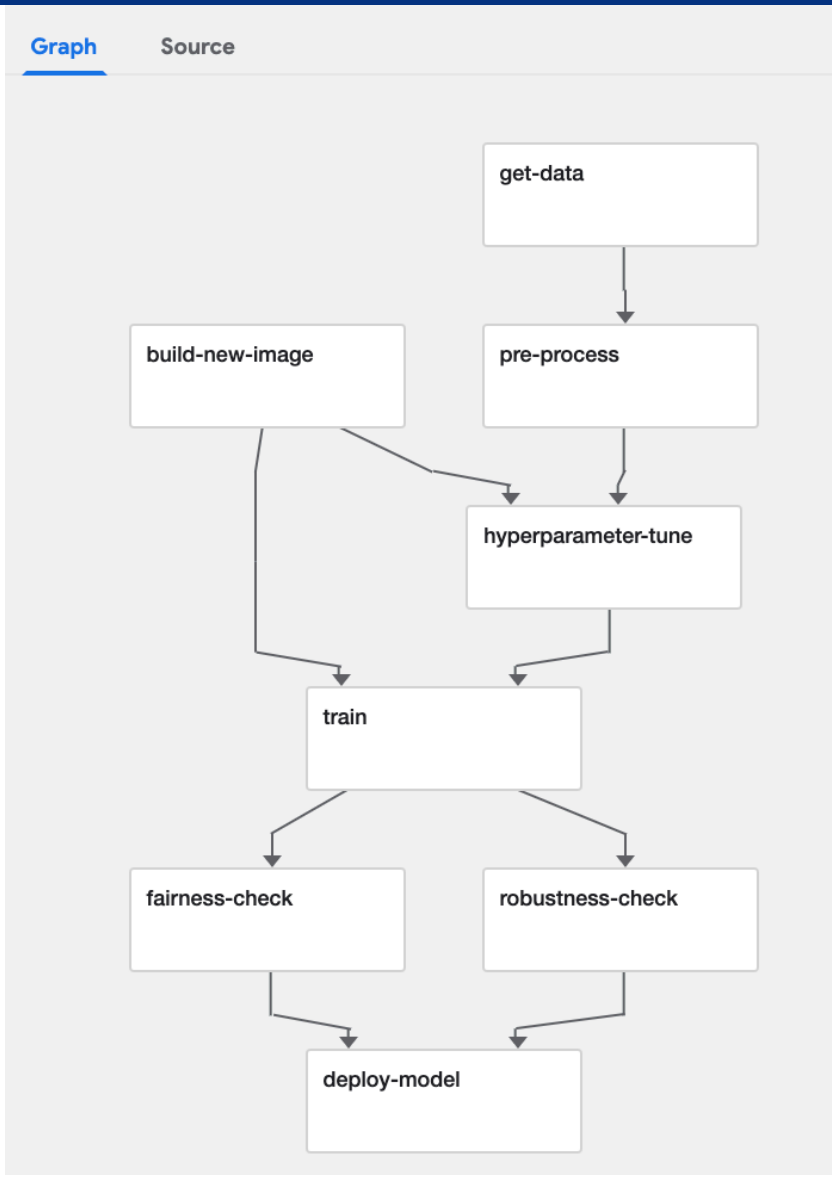

```yaml
apiVersion: sources.eventing.knative.dev/v1alpha1
kind: GitHubSource
metadata:
  name: my-github-source
spec:
  eventTypes:
  - pull_request
  ownerAndRepository: hougangliu/test
  accessToken:
    secretKeyRef:
      name: my-githubsecret
      key: accessToken
  secretToken:
    secretKeyRef:
      name: my-githubsecret
      key: secretToken
  sink:
    apiVersion: v1
    kind: Service
    name: pipeline-launcher
    namespace: kube-system
```

```yaml
apiVersion: v1
kind: ConfigMap
metadata:
  name: pipeline-launcher
  namespace: kube-system
data:
  configfile: |
    version: v1alpha1
    event_map:
      github.com/hougangliu/object_detection:
        pull:
        - "object_detection"
        push:
        - "object_detection"
      github.ibm.com/hougangliu/test:
        pull:
        - "test1"
        - "test2"
        push:
        - "test"
      ...
```

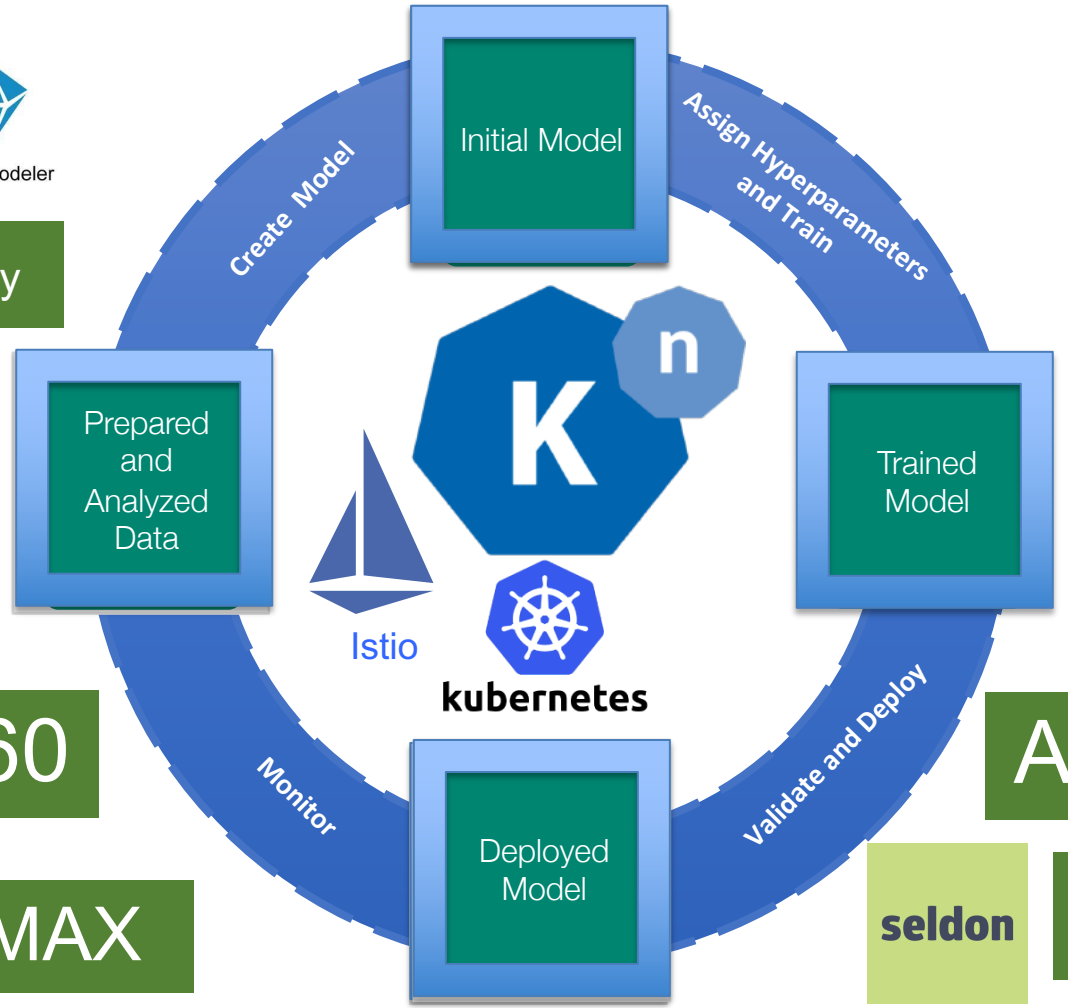
Event Driven ML pipeline

```
@dsl.pipeline(
  name='Object detection',
  description='Object detection'
)
def object_detection(
        worker=3,
        new_image_name="hougangliu/object_detection:latest"):
  getData = get_data()
  pre_process = pre_process(getData.output)
  new_image = build_image(new_image_name)
  hpo = hyperparameter_tune(pre_process.output, new_image.output)
  train = start_train(hpo.output, new_image.output, worker)
  r_check = robustness_check(train.output)
  f_check = fairness_check(train.output)
  deploy = deploy_model(r_check.output, f_check.output)
```

```
apiVersion: build.knative.dev/v1alpha1
kind: Build
metadata:
  name: build-objective-detection
spec:
  serviceAccountName: build-auth
  source:
    git:
      url: https://github.com/hougangliu/object_detection.git
      revision: master
  steps:
    - image: hougangliu/image-build:latest
      args: ["make", "build"]
      name: build-image
    - name: push-image
      image: hougangliu/image-push:latest
      args: ["make", "push"]
      volumeMounts:
        - name: docker-socket-example
          mountPath: /var/run/docker.sock
```



Together: A Transparent, and trusted event driven Open Source AI Pipeline



ANACONDA

RStudio

SPSS Modeler

jupyter

Jupyter Enterprise Gateway

Kubeflow

kube-batch

TensorFlow

PYTORCH

Keras

PaddlePaddle

mxnet

AIF360

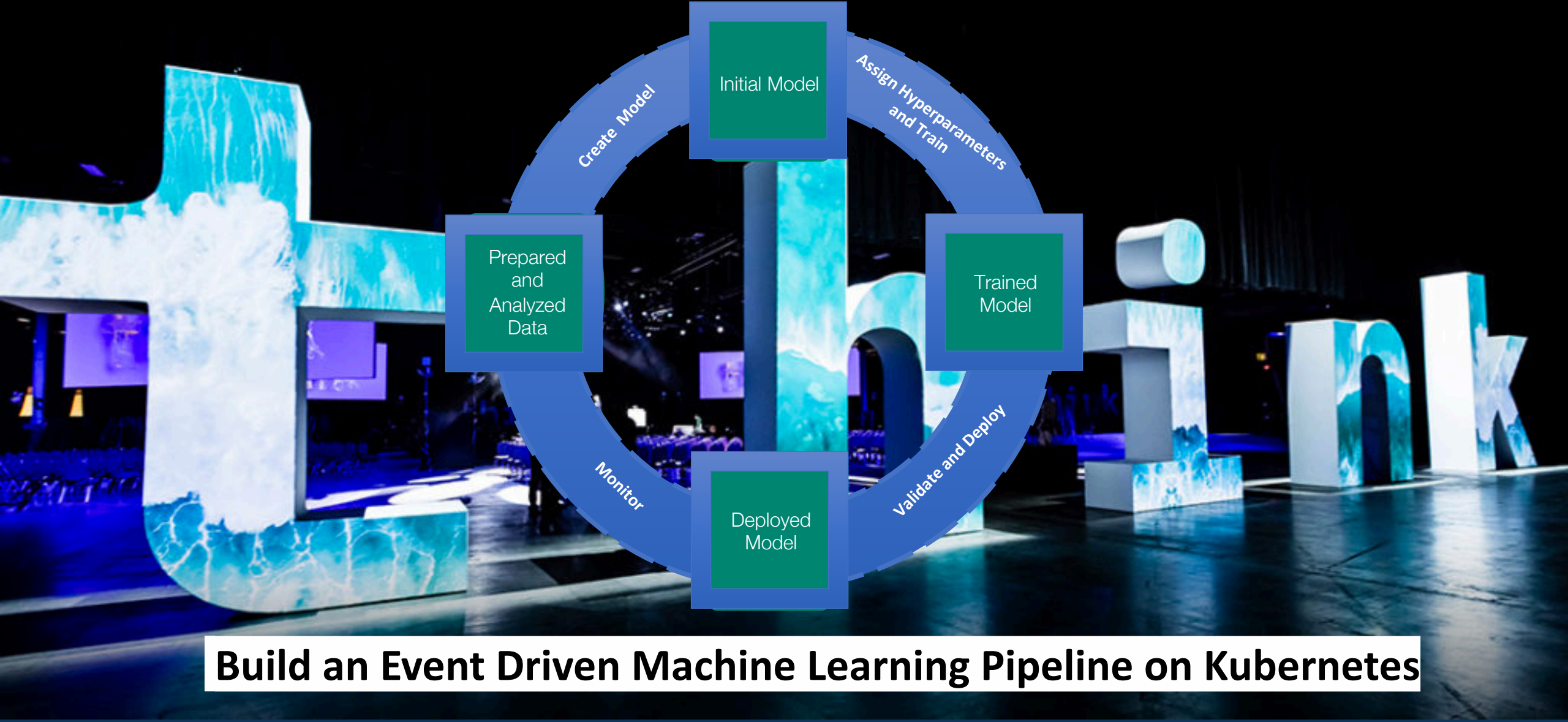
AIF360

MAX

seldon

kfservie

tensorrt-inference-server



Build an Event Driven Machine Learning Pipeline on Kubernetes

THANKS