KubeCon | CloudNativeCon

North America 2019

KubeCon | CloudNativeCon North America 2019

Inferencing Leveraging KNative, Istio and Kubeflow Serving

Animesh Singh - IBM
Clive Cox - Seldon

Agenda

- Introduction to Machine Learning Serving and its challenges
- Kubeflow Serving Introduction
- Monitoring ML Models
- Summary and Roadmap

Enterprise Machine Learning

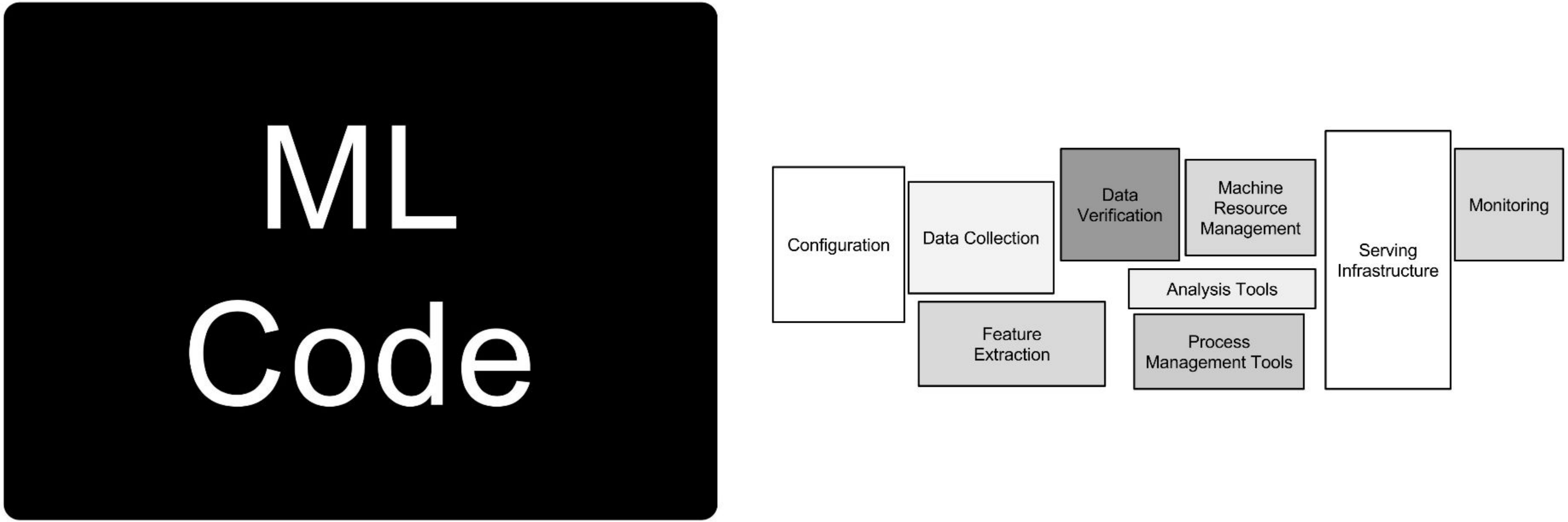
ginablaber
@ginablaber

Follow

The story of enterprise Machine Learning: “It took me 3 weeks to develop the model. It’s been >11 months, and it’s still not deployed.” @DineshNirmalIBM #StrataData #strataconf

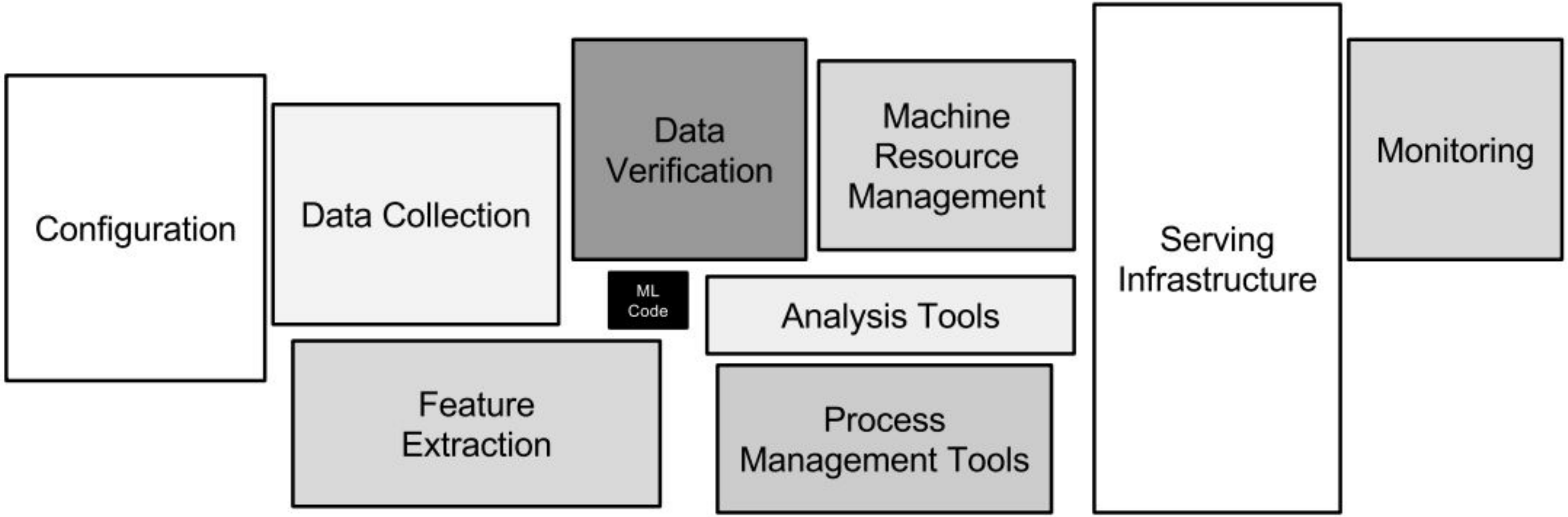
10:19 AM - 7 Mar 2018

Perception



Source: Hidden Technical Debt in Machine Learning Systems

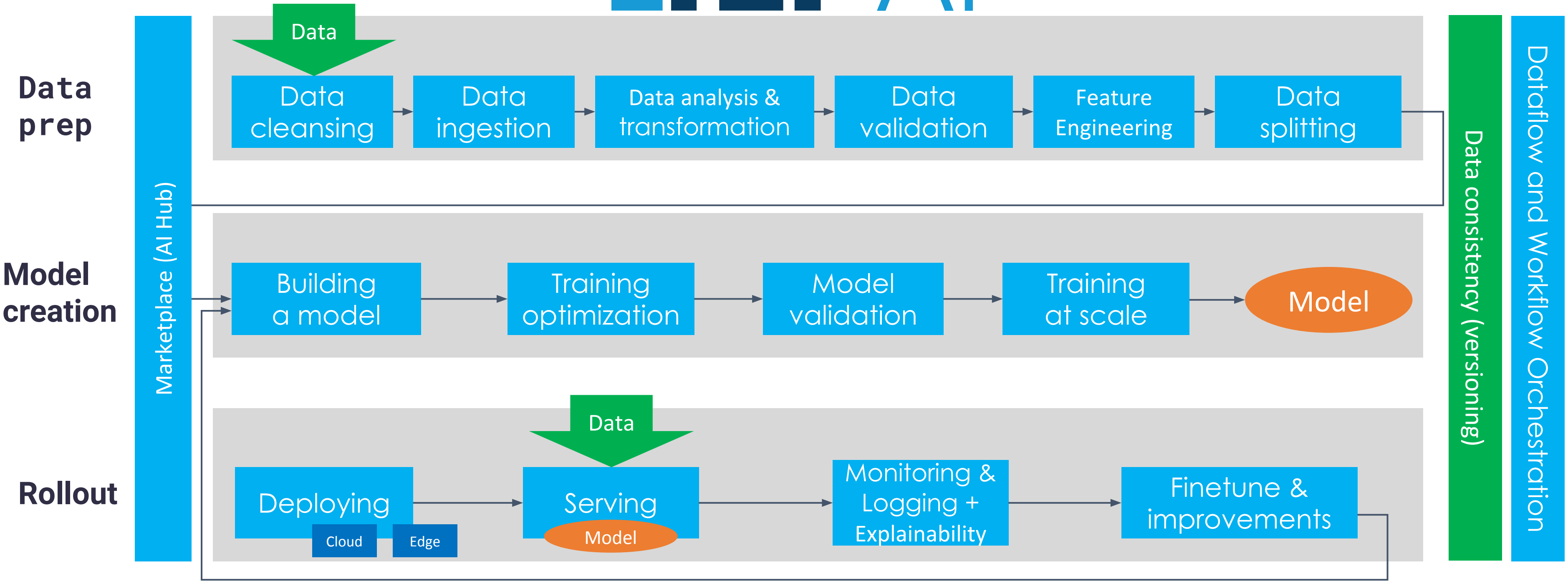
In reality…ML Code is tiny part in this overall platform



*Source: Hidden Technical Debt in Machine Learning Systems

ML Workflow

LF AI



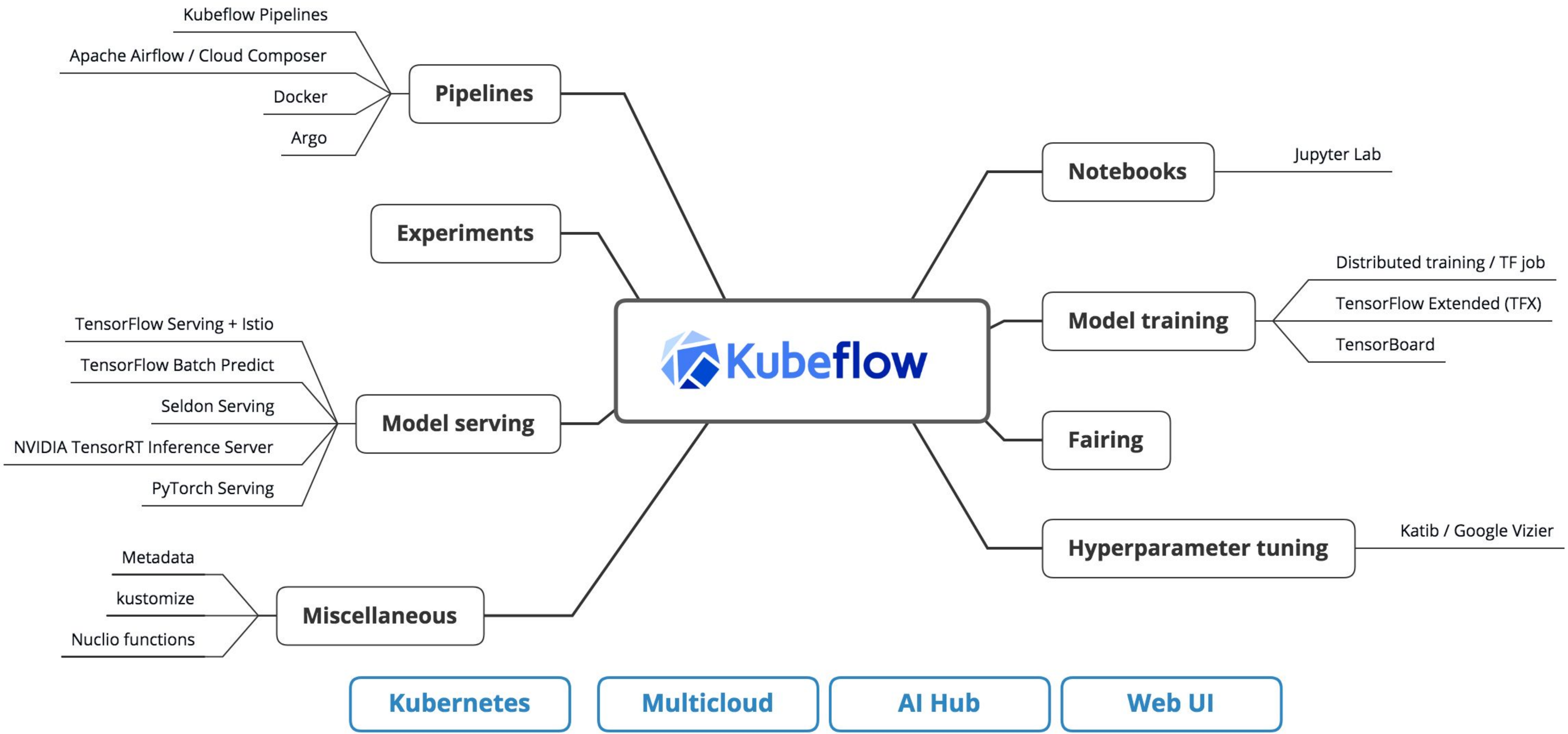
End to end ML on Kubernetes?

KubeCon | CloudNativeCon North America 2019

First, can you become an expert in ...

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

Introducing: Kubeflow



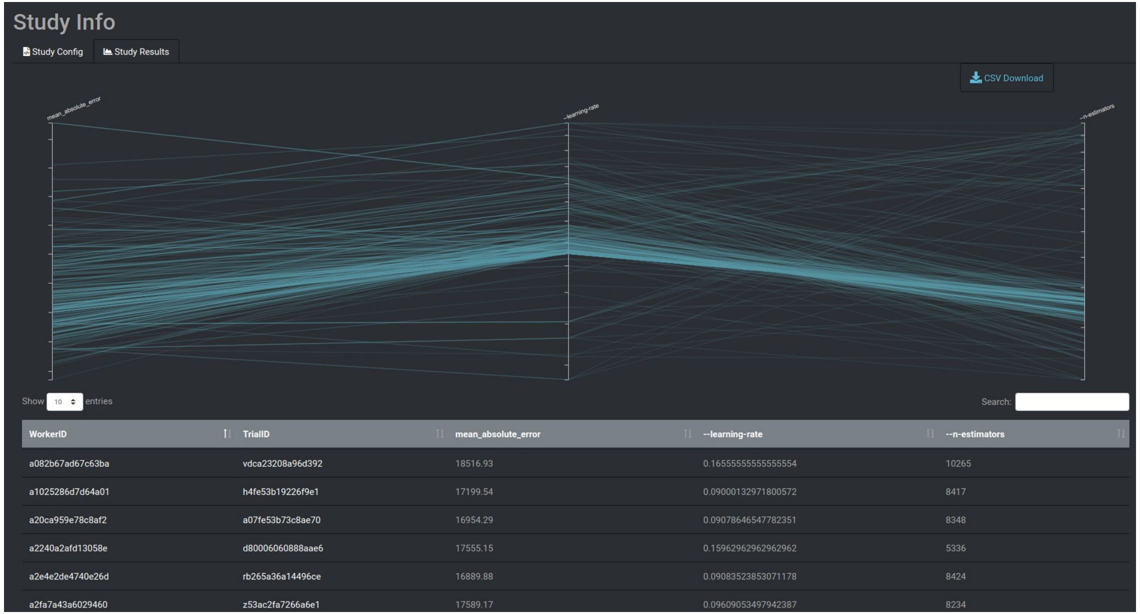
Distributed Model Training and HPO (TFJob, PyTorch Job, Katib, …)

- Addresses One of the key goals for model builder persona:

  Distributed Model Training and Hyper parameter optimization for **Tensorflow, PyTorch etc.**

- Common problems in HP optimization
  - Overfitting
  - Wrong metrics
  - Too few hyperparameters

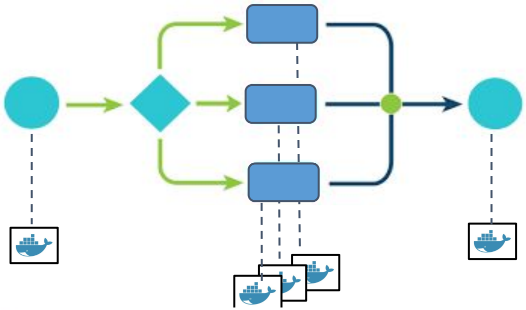
- Katib: a fully open source, Kubernetes-native hyperparameter tuning service
  - Inspired by Google Vizier
  - Framework agnostic
  - Extensible algorithms

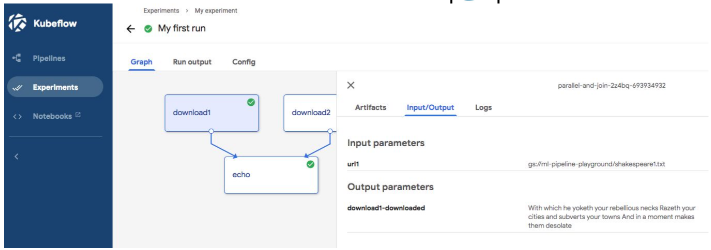


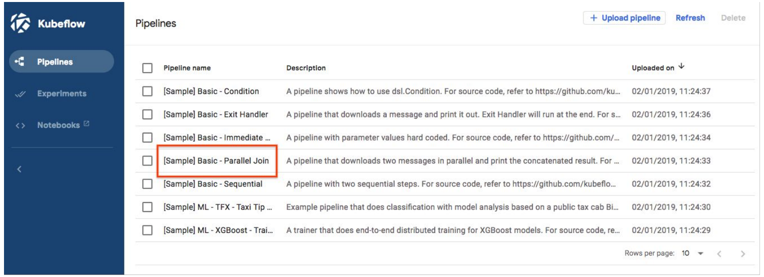
TensorFlow PYTORCH XGBoost HOROVOD

Kubeflow Pipelines

- Containerized implementations of ML Tasks
  - Pre-built components: Just provide params or code snippets (e.g. training code)
  - Create your own components from code or libraries
  - Use any runtime, framework, data types
  - Attach k8s objects - volumes, secrets
- Specification of the sequence of steps
  - Specified via Python DSL
  - Inferred from data dependencies on input/output
- Input Parameters
  - A “Run” = Pipeline invoked w/ specific parameters
  - Can be cloned with different parameters
- Schedules
  - Invoke a single run or create a recurring scheduled pipeline







IBM and Seldon Major Contributors

Source devstats.org

Companies summary

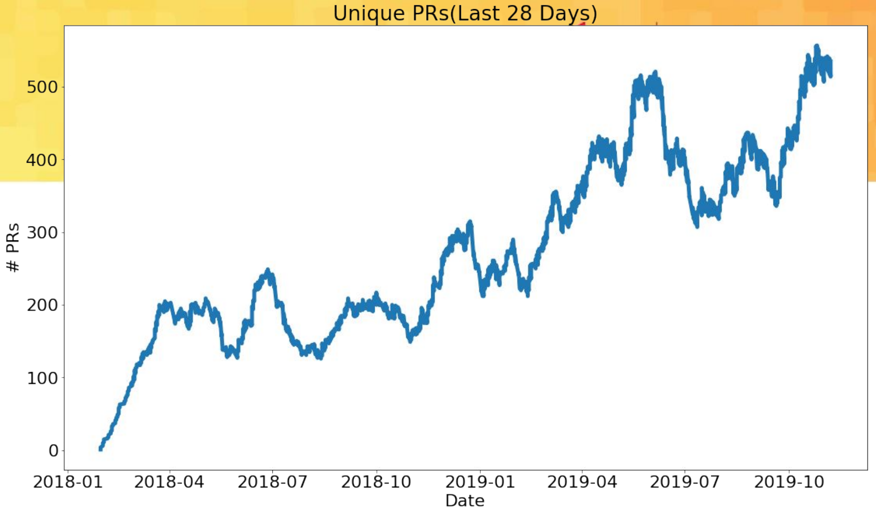
Range: Last year | Metric: Contributions

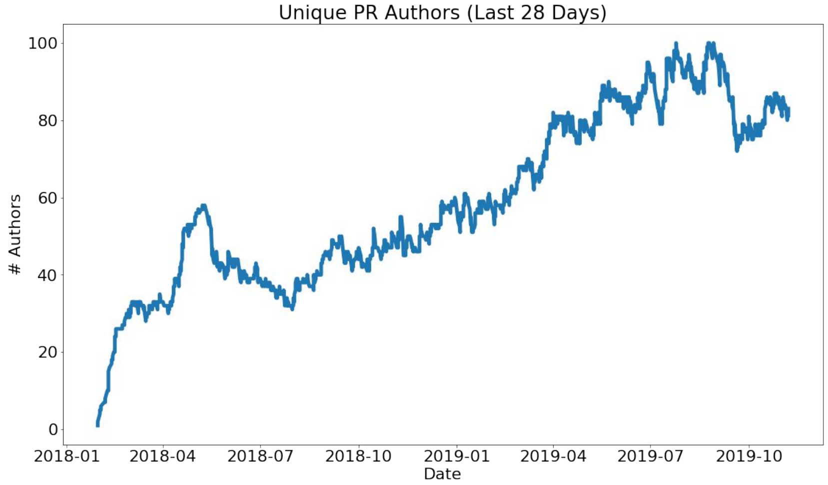
Kubeflow Companies statistics (Contributions, Range: Last year), bots excluded

Company	Number
All	68330
Google	28445
IBM	4318
Cisco	4197
Caicloud	1688
Amazon	693
Microsoft	681
Seldon	474
Net EASE	444
NetEase	398
NTT	315
Intel	214

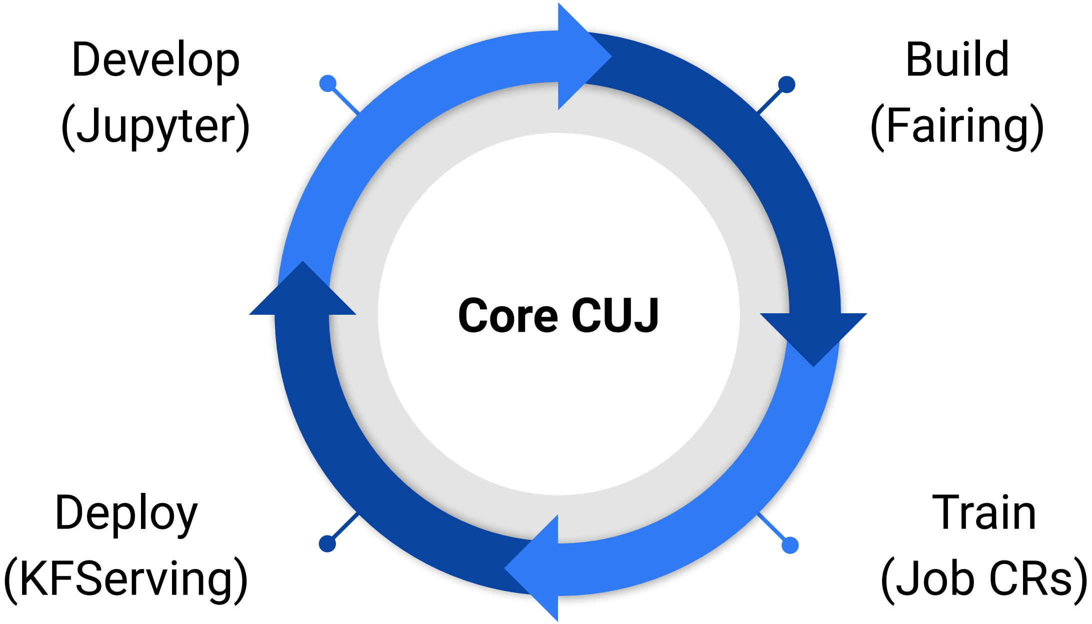
Community is growing!

Alibaba Cloud
aws
蚂蚁金服 ANT FINANCIAL
argo
Arrikto
Bloomberg
Caicloud
CANONICAL
CISCO
DATAWIRE
DELL
GitHub
Google
GOJEK
heptio
IBM
iguazio
intel
in
Microsoft
MOMENTA
One Convergence
Pachyderm
PRIMER
jupyter
redhat
SELDON
shopify
tusimple
UBER
云知声 Unisound
weaveworks





Kubeflow 1.0 Arriving January 2020



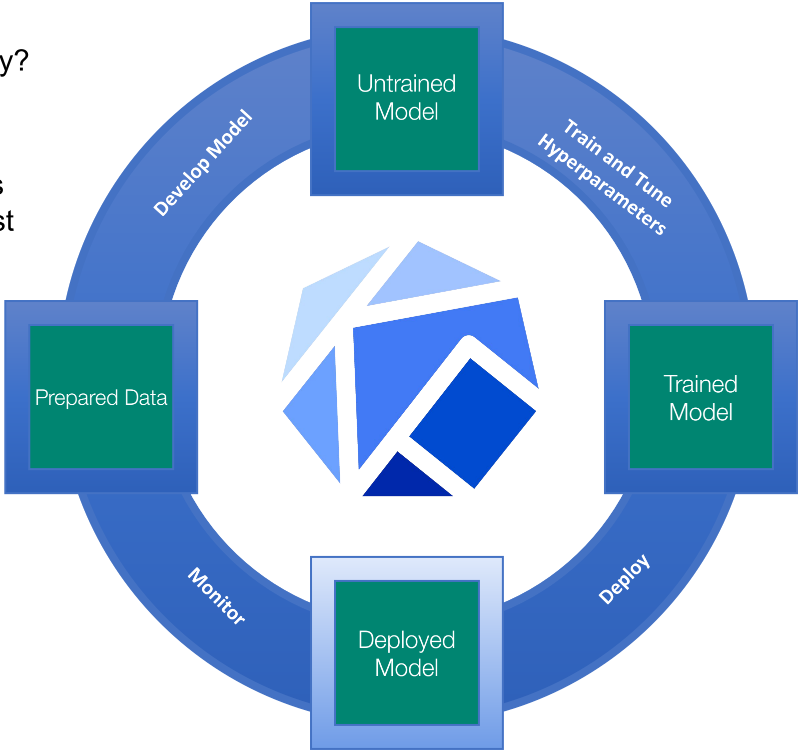
http://bit.ly/kf_roadmap

Production Model Serving

Production Model Serving? How hard could it be?

KubeCon | CloudNativeCon North America 2019

- ❑ Cost:
  Is the model over or under scaled?
  Are resources being used efficiently?
- ❑ Monitoring:
  Are the endpoints healthy? What is the performance profile and request trace?
- ❑ Rollouts:
  Is this rollout safe? How do I roll back? Can I test a change without swapping traffic?
- ❑ Protocol Standards:
  How do I make a prediction?
  GRPC? HTTP? Kafka?



- ❑ Frameworks:
  How do I serve on Tensorflow? XGBoost? Scikit Learn? Pytorch? Custom Code?
- ❑ Features:
  How do I explain the predictions? What about detecting outliers and skew? Bias detection? Adversarial Detection?
- ❑ How do I wire up custom pre and post processing

Experts fragmented across industry

- Seldon Core was pioneering Graph Inferencing.
- IBM and Bloomberg were exploring serverless ML lambdas. IBM gave a talk on the ML Serving with Knative at last KubeCon in Seattle
- Google had built a common Tensorflow HTTP API for models.
- Microsoft Kubernetizing their Azure ML Stack

Putting the pieces together

- Kubeflow created the conditions for collaboration.
- A promise of open code and open community.
- Shared responsibilities and expertise across multiple companies.
- Diverse requirements from different customer segments

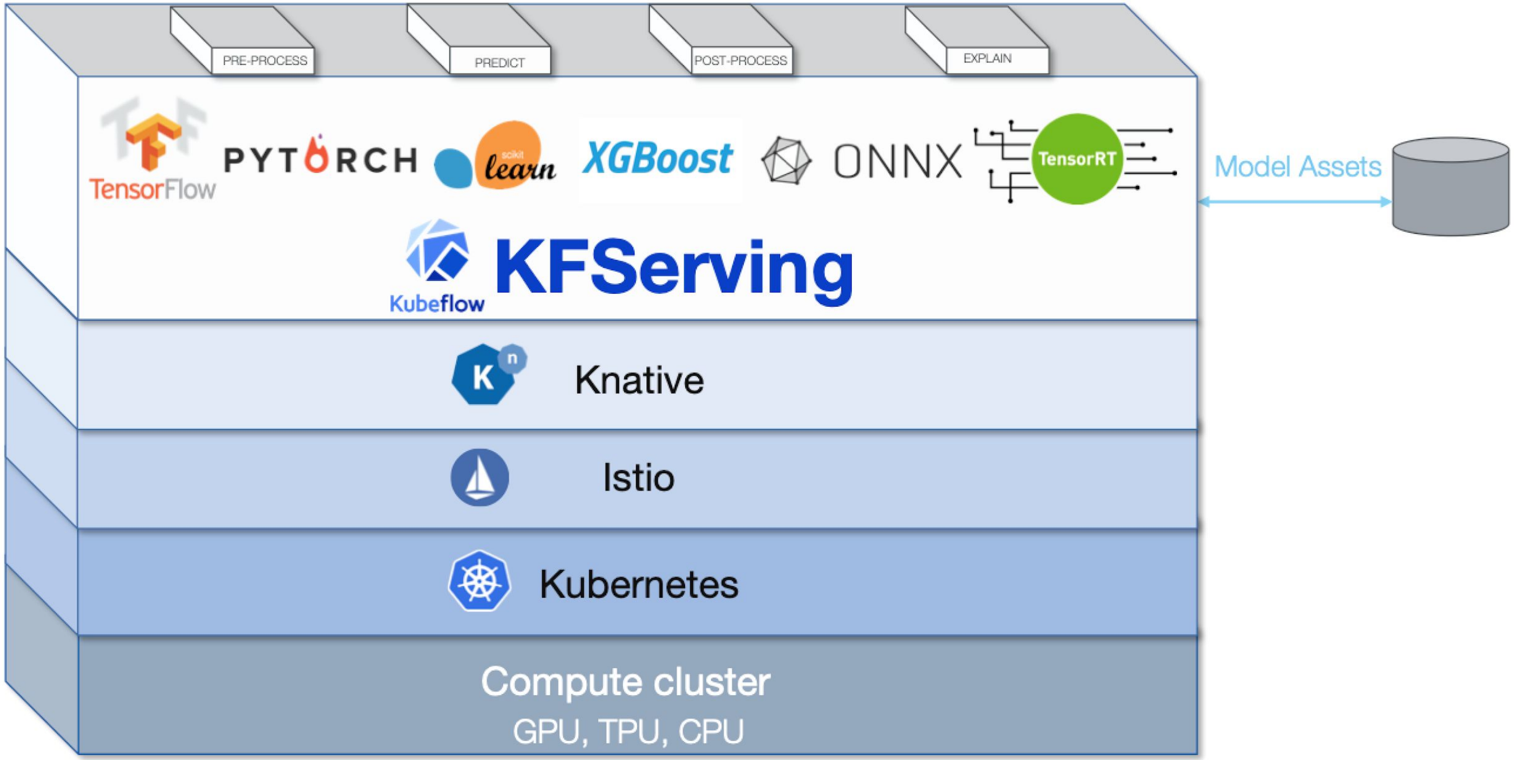
Google SELDON IBM Bloomberg Microsoft

Introducing KFServing

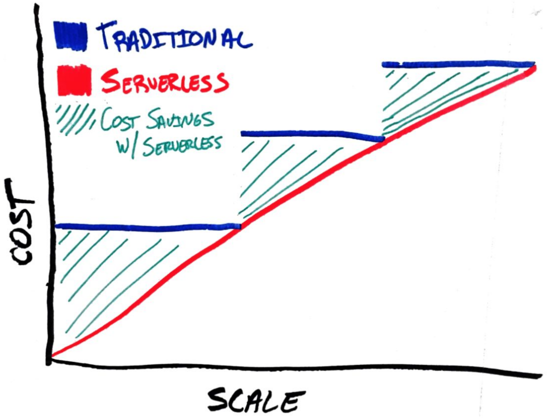
KFServing

- Founded by Google, Seldon, IBM, Bloomberg and Microsoft
- Part of the Kubeflow project
- Focus on 80% use cases - single model rollout and update
- Kfserving 1.0 goals:
  - Serverless ML Inference
  - Canary rollouts
  - Model Explanations
  - Optional Pre/Post processing

KFServing Stack



KNative



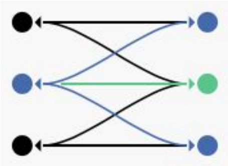
IBM is
2nd largest contributor

Knative provides a set of building blocks that enable declarative, container-based, serverless workloads on Kubernetes. Knative Serving provides primitives for serving platforms such as:

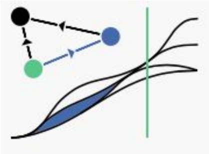
- Event triggered functions on Kubernetes
- Scale to and from zero
- Queue based autoscaling for GPUs and TPUs. KNative autoscaling by default provides inflight requests per pod
- Traditional CPU autoscaling if desired. Traditional scaling hard for disparate devices (GPU, CPU, TPU)

Istio

An open service mesh platform to **connect**, **observe**, **secure**, and **control** microservices.
Founded by Google, IBM and Lyft. IBM is the 2nd largest contributor

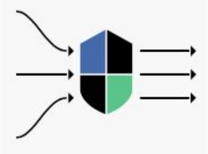


Connect: Traffic Control, Discovery, Load Balancing, Resiliency



Observe: Metrics, Logging, Tracing

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

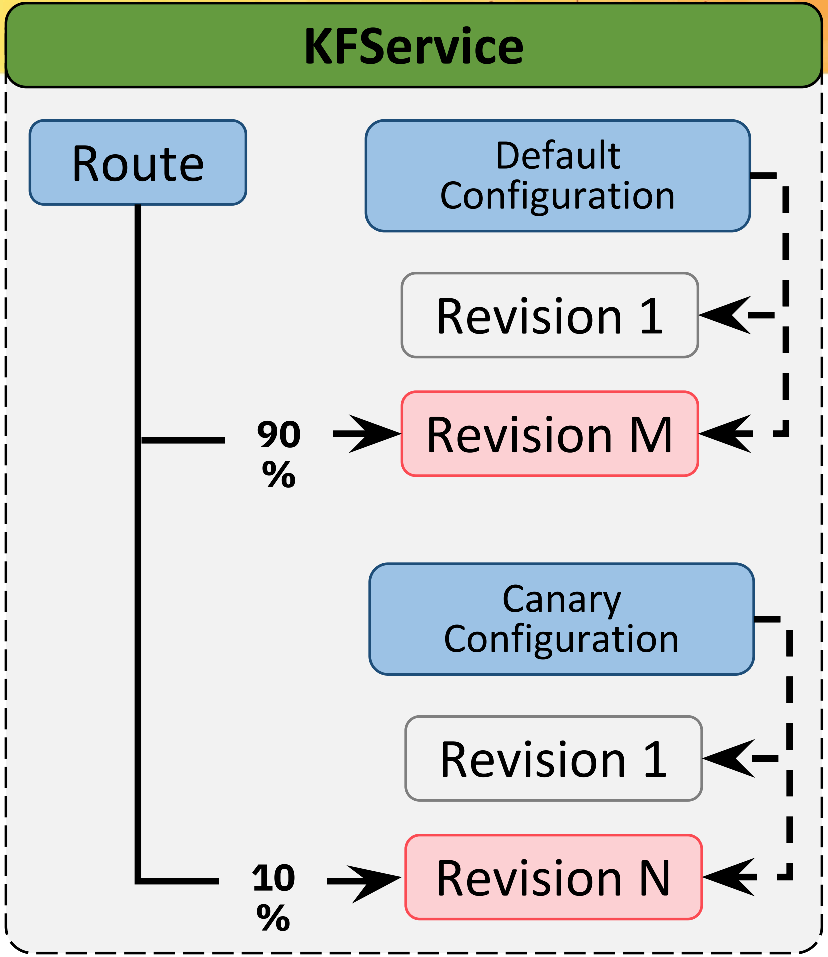


Control: Policy Enforcement

KFServing: Default and Canary Configurations

Manages the hosting aspects of your models

- **InferenceService** - manages the lifecycle of models
- **Configuration** - manages history of model deployments. Two configurations for default and canary.
- **Revision** - A snapshot of your model version
  - Config and image
- **Route** - Endpoint and network traffic management



Supported Frameworks, Components and Storage

Model Servers

- TensorFlow
- Nvidia TRTIS
- PyTorch
- XGBoost
- SKLearn
- ONNX

Components:

- Predictor, Explainer, Transformer

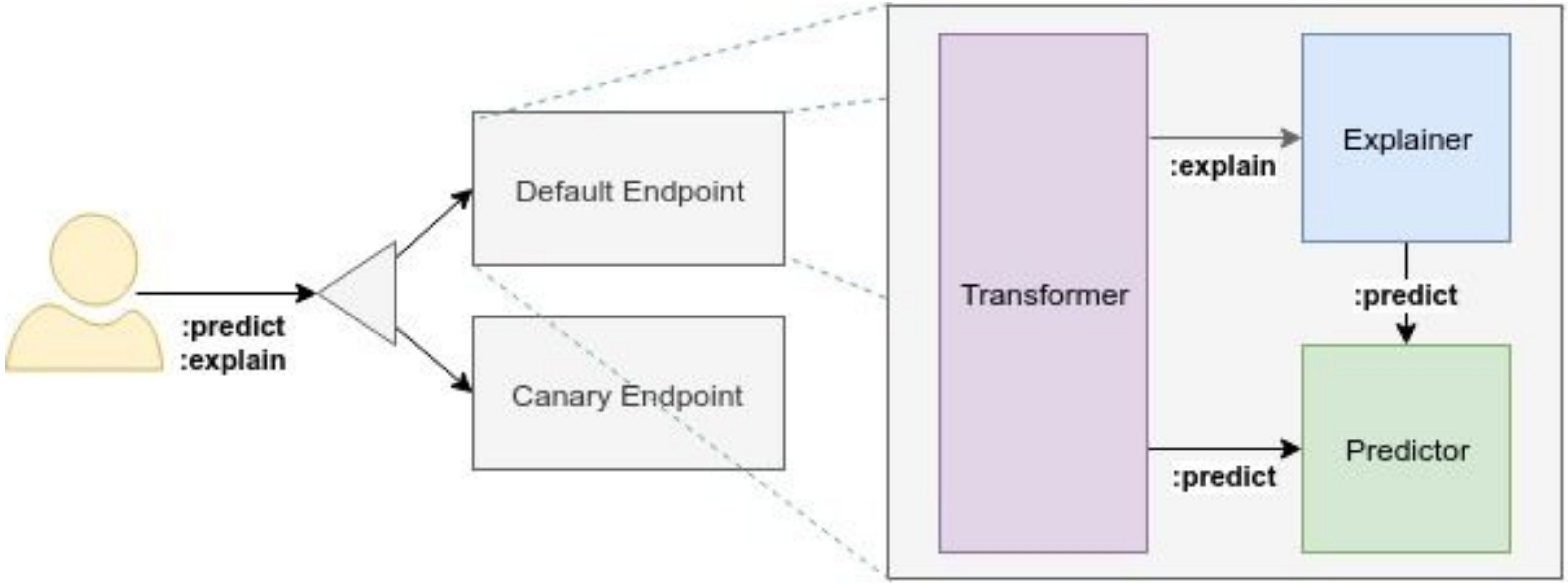
Storages

- AWS/S3
- GCS
- Azure Blob
- PVC

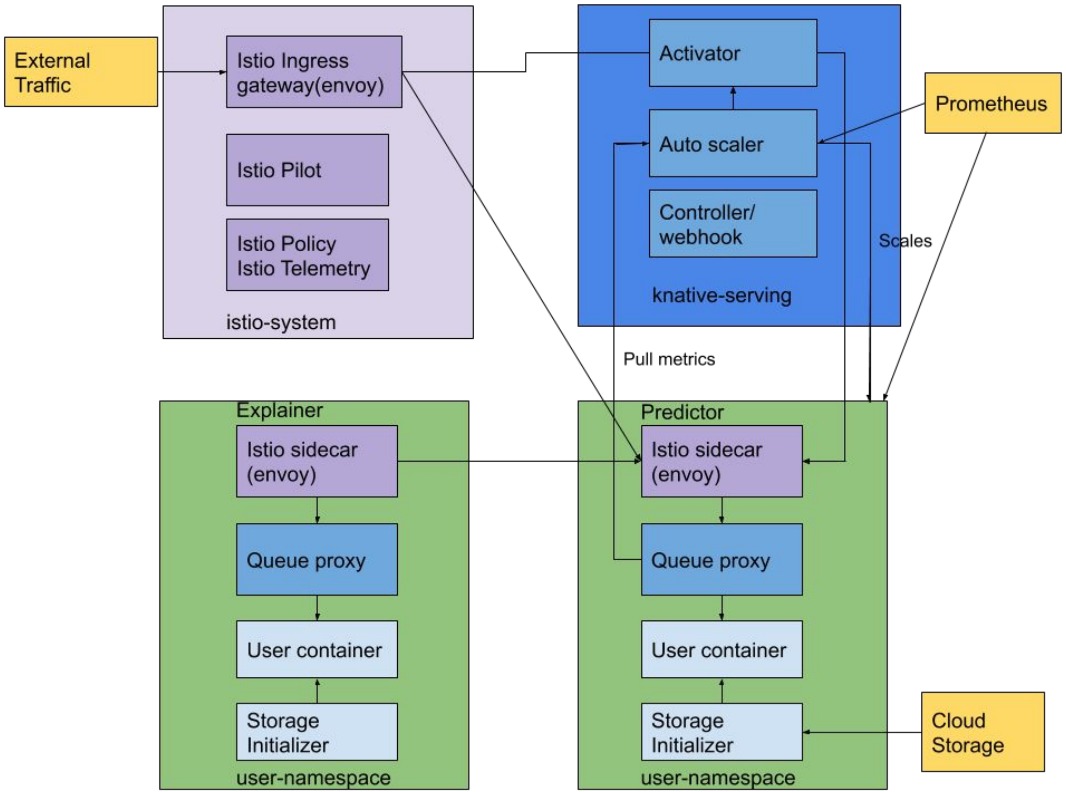
Kubeflow

Inference Service Control Plane

The InferenceService architectureconsists of a static graph of components which coordinate requests for a single model. Advanced features such as Ensembling, A/B testing, and Multi-Arm-Bandits should compose InferenceServices together.



KFServing Deployment View



KFServing Data Plane Unification

- Today's popular model servers, such as TFServing, ONNX, Seldon, TRTIS, all communicate using similar but non-interoperable HTTP/gRPC protocol

- KFServing v1 data plane protocol uses TFServing compatible HTTP API and introduces explain verb to standardize between model servers, punt on v2 for gRPC and performance optimization.

KFServing Data Plane v1 protocol

API	Verb	Path	Payload
List Models	GET	/v1/models	[model_names]
Readiness	GET	/v1/models/<model_name>	
Predict	POST	/v1/models/<model_name>:predict	Request: {instances:[]} Response: {predictions:[]}
Explain	POST	/v1/models<model_name>:explain	Request: {instances:[]} Response: {predictions:[], explanations:[]}

KFServing Examples


```
apiVersion: "serving.kubeflow.org/v1alpha1"
kind: "InferenceService"
metadata:
  name: "sklearn-iris"
spec:
  default:
        sklearn:
        modelUri: "gs://kfserving-samples/models/sklearn/iris"
```

scikit learn

```
apiVersion: "serving.kubeflow.org/v1alpha1"
kind: "InferenceService"
metadata:
  name: "flowers-sample"
spec:
  default:
        tensorflow:
        modelUri: "gs://kfserving-samples/models/tensorflow/flowers"
```

TensorFlow

```
apiVersion: "serving.kubeflow.org/v1alpha1"
kind: "InferenceService"
metadata:
  name: "pytorch-iris"
spec:
  default:
        pytorch:
        modelUri: "gs://kfserving-samples/models/pytorch/iris"
```

PYTORCH

Canary/Pinned Examples

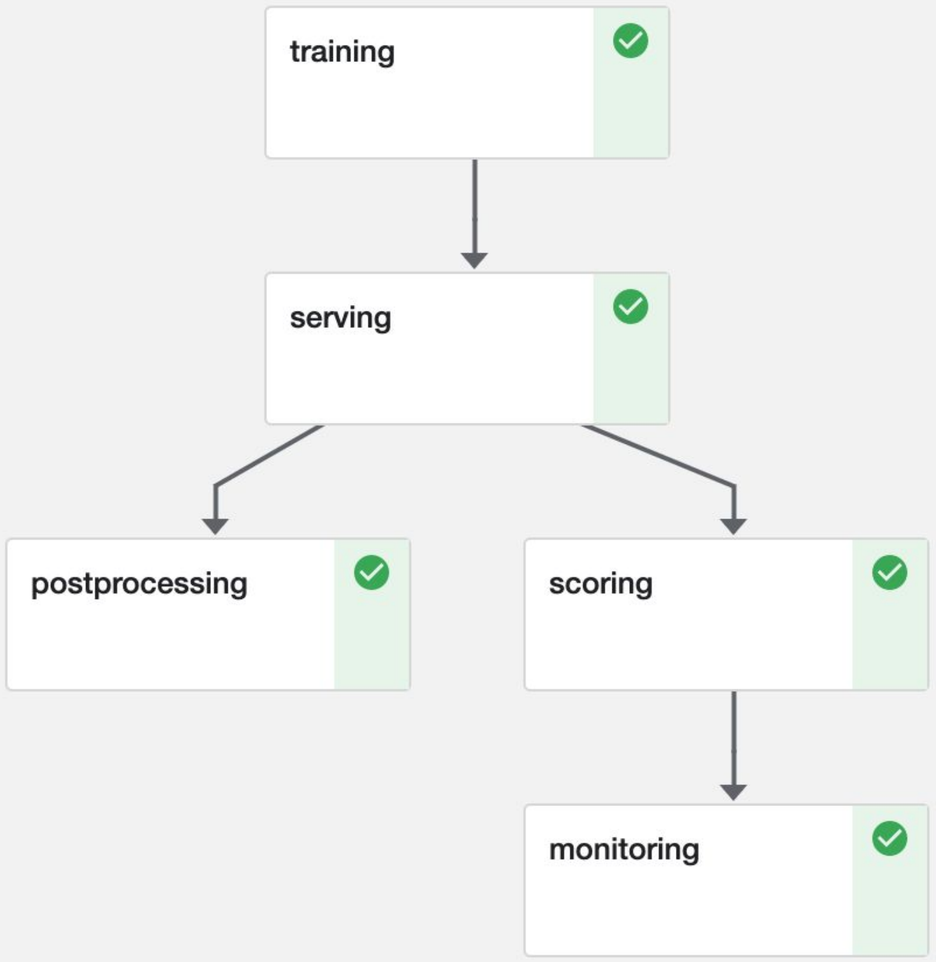

```yaml
apiVersion: "serving.kubeflow.org/v1alpha1"
kind: "KFService"
metadata:
  name: "my-model"
spec:
  default:
        # 90% of traffic is sent to this model
        tensorflow:
        modelUri: "gs://mybucket/mymodel-2"
  canaryTrafficPercent: 10
  canary:
        # 10% of traffic is sent to this model
        tensorflow:
        modelUri: "gs://mybucket/mymodel-3"
```

Canary

```yaml
apiVersion: "serving.kubeflow.org/v1alpha1"
kind: "KFService"
metadata:
  name: "my-model"
spec:
  default:
        tensorflow:
        modelUri: "gs://mybucket/mymodel-2"
  # Defaults to zero, so can also be omitted or explicitly set to zero.
  canaryTrafficPercent: 0
  canary:
        # Canary is created but no traffic is directly forwarded.
        tensorflow:
        modelUri: "gs://mybucket/mymodel-3"
```

Pinned

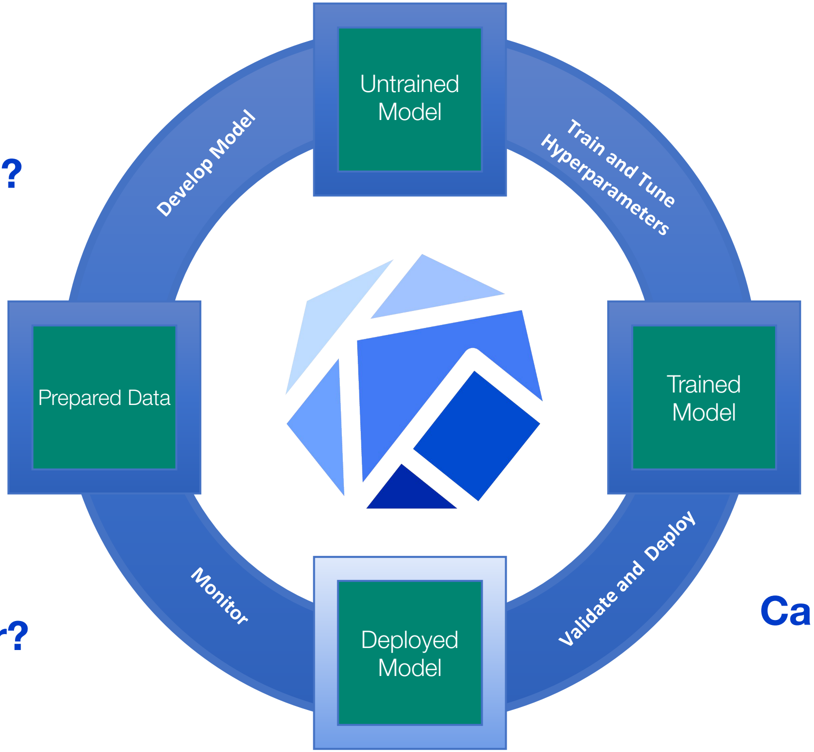
Demo



Model Serving is accomplished. Can the predictions be trusted?

Are there concept drifts?

Is the model vulnerable to adversarial attacks?

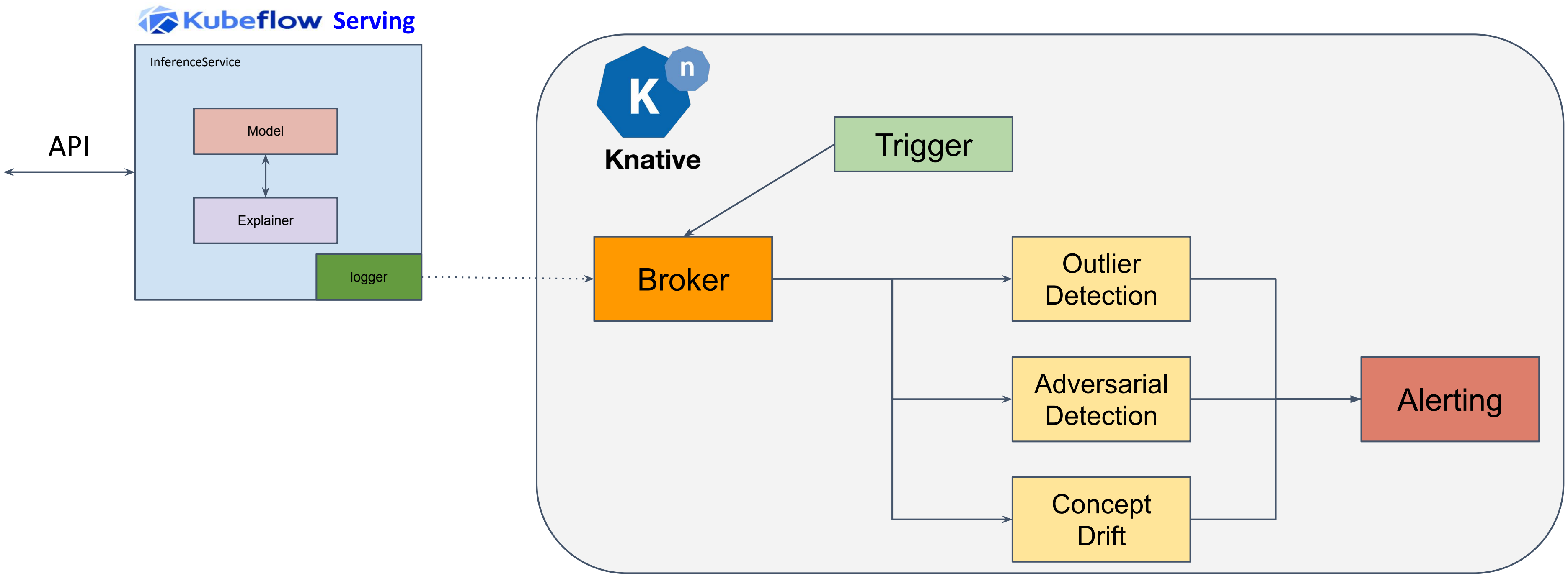


Is there an outlier?

Can the model explain its predictions?

Production Machine Learning Serving

Production ML Architecture



Machine Learning Explanations

Why Explain ML Models?

Regulation (GDPR):

[the data subject possesses the right to access] *“meaningful information about the logic involved, as well as the significance and the envisaged consequences of such processing for the data subject."*

Insight:

- Is my model doing what I think it’s doing?
- Investigate model behaviour, e.g. on outliers

ML Explanation Goals

- Human interpretable
- Not over-simplified
- **Trade-off between interpretability and fidelity**

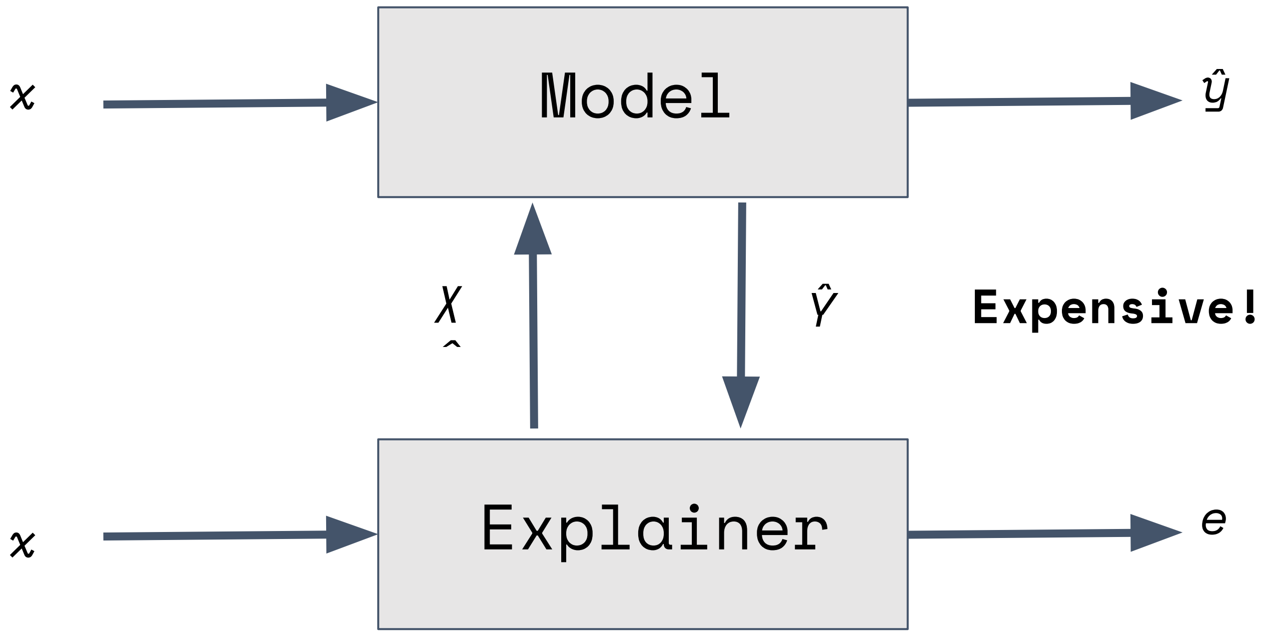
Local Black Box Explanations

Explain this:

Age:
23
Occupation:
Bar staff
Postcode:
IV3 5SN
Owns house:
No

Deny:
p=0.95
Accept:
p=0.05

Architecture



Seldon Alibi:Explain

https://github.com/SeldonIO/alibi


Giovanni Vacanti


Janis Klaise


Arnaud Van Looveren

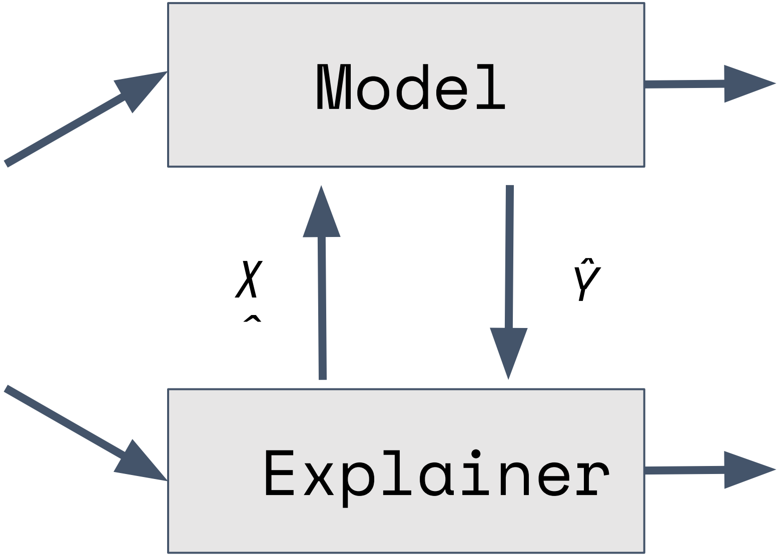

Alexandru Coca

State of the art implementations:

- Anchors
- Counterfactuals
- Contrastive explanations
- Trust scores

ALIBI EXPLAIN

Anchors



Persian cat: p=0.90

Dishwasher: p=0.003

Notebook:

Precision: 0.95

KfServing Explanations

```yaml
apiVersion: "serving.kubeflow.org/v1alpha2"
kind: "InferenceService"
metadata:
 name: "income"
spec:
 default:
   predictor:
     sklearn:
       storageUri: "gs://seldon-models/sklearn/income/model"
   explainer:
     alibi:
       type: AnchorTabular
       storageUri: "gs://seldon-models/sklearn/income/explainer"
```

```yaml
apiVersion: "serving.kubeflow.org/v1alpha2"
kind: "InferenceService"
metadata:
 name: "moviesentiment"
spec:
 default:
   predictor:
     sklearn:
       storageUri: "gs://seldon-models/sklearn/moviesentiment"
   explainer:
     alibi:
       type: AnchorText
```

Explanation Demos

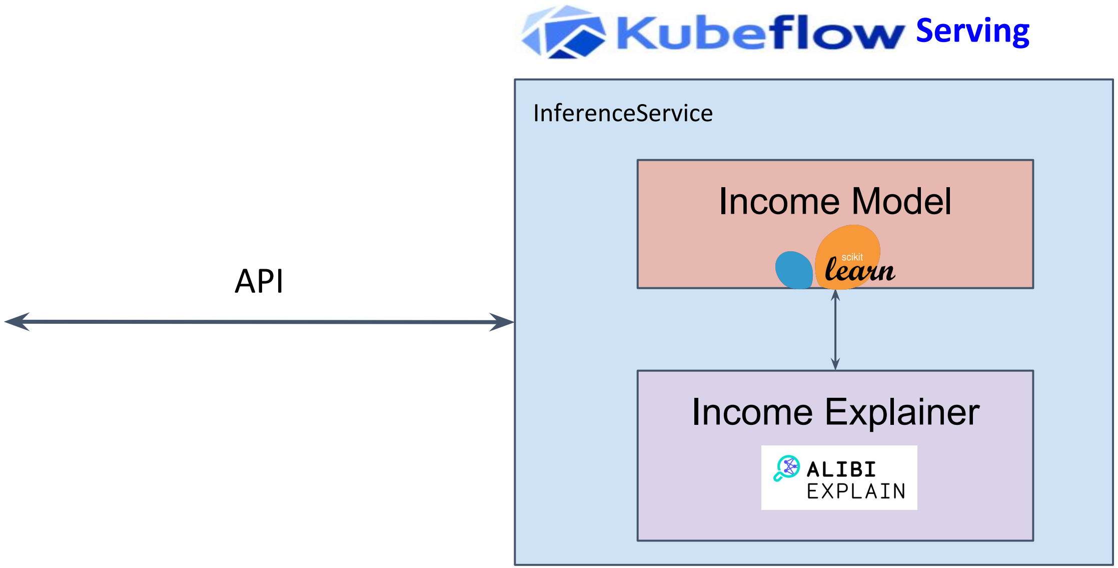
Income Prediction SKLearn Classifier and Alibi:Explain AnchorTabular Explainer

https://github.com/kubeflow/kfserving/blob/master/docs/samples/explanation/alibi/income/income_explanations.ipynb

Movie Review RoBERTa Classifier and Alibi:Explain AnchorText Explainer

https://github.com/SeldonIO/seldon-models/blob/master/pytorch/moviesentiment_roberta/inference/kfserving/movie_review_explanations.ipynb

Income Model and Explainer



Explanations: Resources

AI Explainability 360
↳ (AIX360)

https://github.com/IBM/AIX360

AIX360

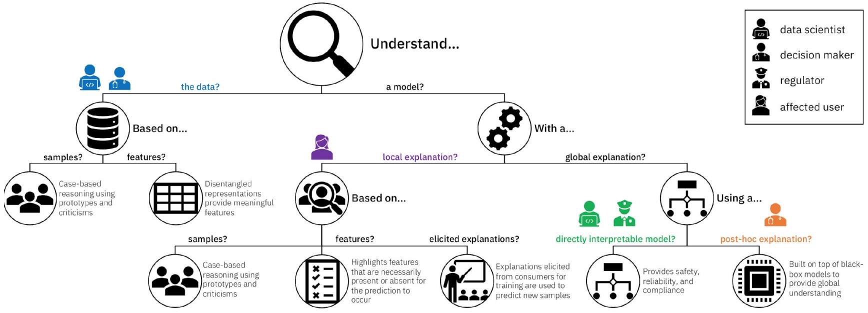
AIX360 toolkit is an open-source library to help explain AI and machine learning models and their predictions. This includes three classes of algorithms: local post-hoc, global post-hoc, and directly interpretable explainers for models that use image, text, and structured/tabular data.

The AI Explainability360 Python package includes a comprehensive set of explainers, both at global and local level.

Toolbox

Local post-hoc
Global post-hoc
Directly interpretable

http://aix360.mybluemix.net



KubeCon | CloudNativeCon

North America 2019

Payload Logging

Payload Logging

Why:

- Capture payloads for analysis and future retraining of the model
- Perform offline processing of the requests and responses

KfServing Implementation (alpha):

- Add to any InferenceService Endpoint: Predictor, Explainer, Transformer
- Log Requests, Responses or Both from the Endpoint
- Simple specify a URL to send the payloads
- URL will receive CloudEvents

cloudevents

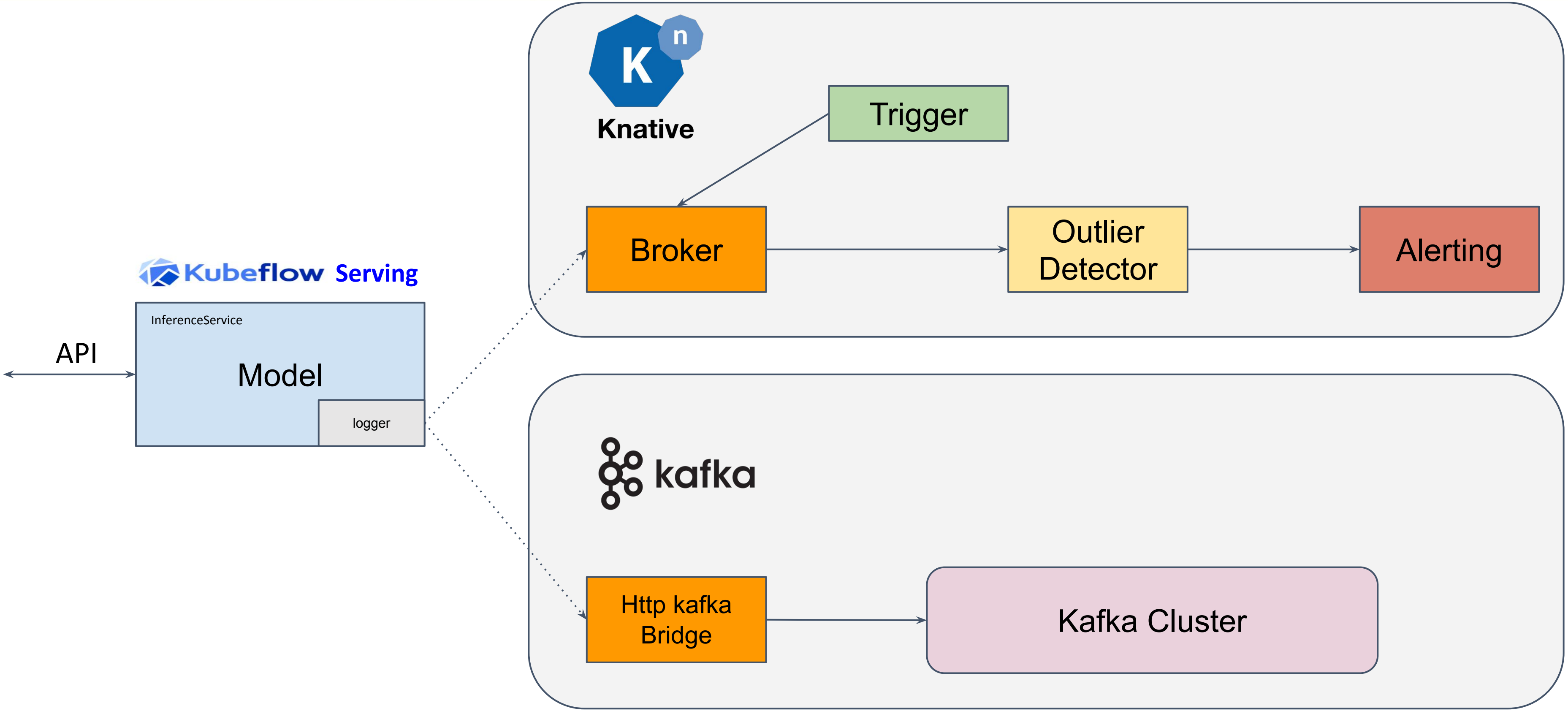
```
POST /event HTTP/1.0
Host: example.com
Content-Type: application/json
ce-specversion: 1.0
ce-type: repo.newItem
ce-source: http://bigco.com/repo
ce-id: 610b6dd4-c85d-417b-b58f-3771e532

<payload>
```

Payload Logging


```yaml
apiVersion: "serving.kubeflow.org/v1alpha2"
kind: "InferenceService"
metadata:
 name: "sklearn-iris"
spec:
 default:
   predictor:
     minReplicas: 1
     logger:
       url: http://message-dumper.default/
       mode: all
     sklearn:
       storageUri: "gs://kfserving-samples/models/sklearn/iris"
       resources:
         requests:
           cpu: 0.1
```

Payload Logging Architecture Examples



KubeCon | CloudNativeCon

North America 2019

ML Inference Analysis

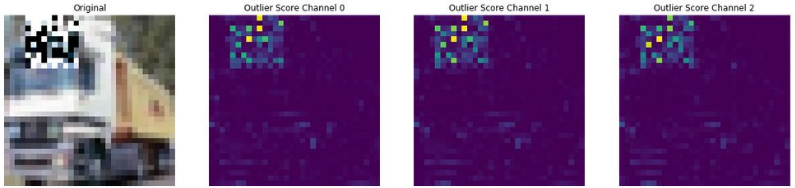
ML Inference Analysis

<u>Don't trust predictions on instances outside of training distribution!</u>

- Outlier Detection
- Adversarial Detection
- Concept Drift

Outlier Detection

Don't trust predictions on instances outside of training distribution!



→ ***Outlier Detection***

Detector types:

- stateful online vs. pretrained offline
- feature vs. instance level detectors

Data types:

- tabular, images & time series

Outlier types:

- global, contextual & collective outliers

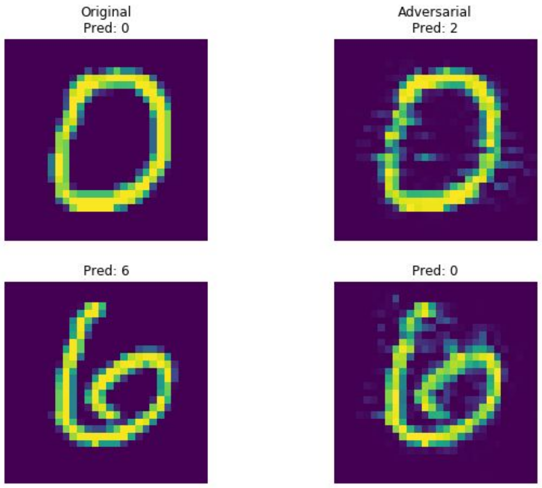
Adversarial Detection

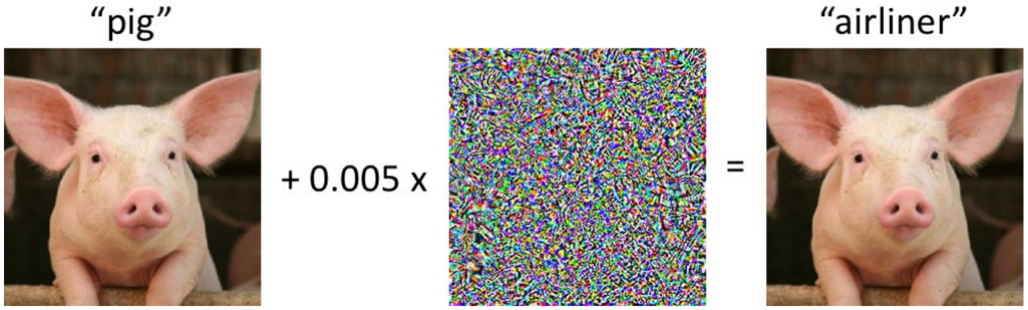
Don't trust predictions on instances outside of training distribution!

→ ***Adversarial Detection***

- Outliers w.r.t. the model prediction

- Detect small input changes with a big impact on predictions!





Concept Drift

<u>Production data distribution != training distribution?</u>

→ ***Concept Drift! Retrain!***

Need to track the right distributions:

- feature vs. instance level
- continuous vs. discrete
- online vs. offline training data
- track streaming number of outliers

Seldon Alibi:Detect

https://github.com/SeldonIO/alibi-detect

Giovanni Vacanti

Janis Klaise

Arnaud Van Looveren

Alexandru Coca

State of the art implementations:

- Outlier Detection
- Adversarial Detection
- Concept Drift (roadmap)

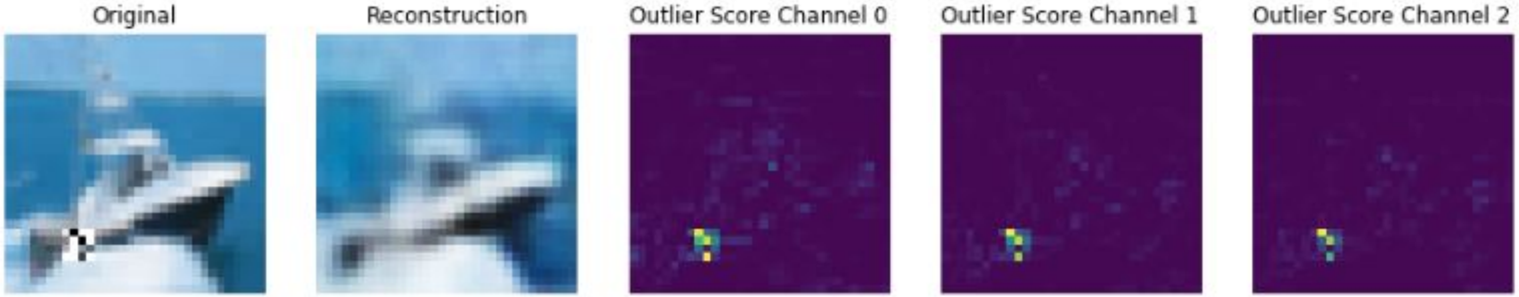
Outlier Detection Demo

KubeCon | CloudNativeCon North America 2019

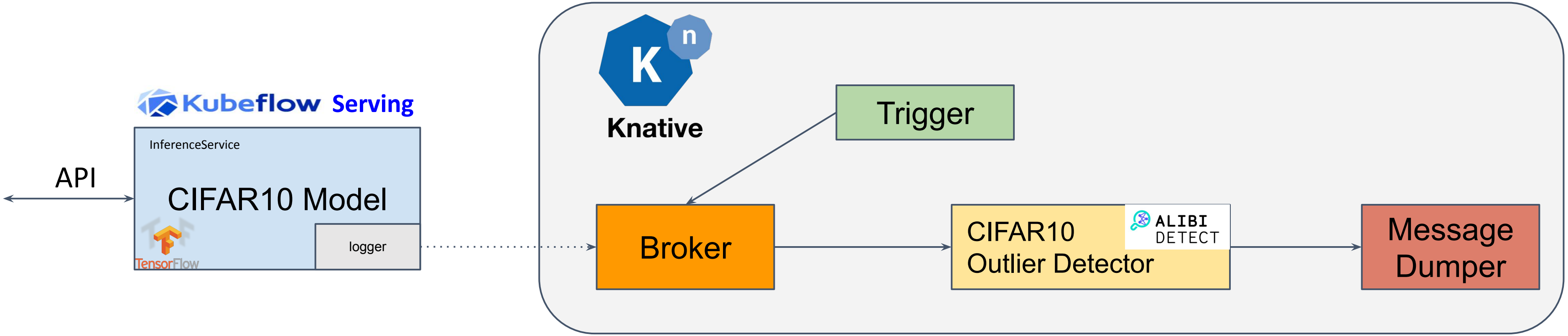
KFServing CIFAR10 Model with Alibi:Detect VAE Outlier Detector

https://github.com/SeldonIO/alibi-detect/tree/master/integrations/samples/kfserving/od-cifae10

Outlier image and heatmap of VAE outlier score per RGB channel



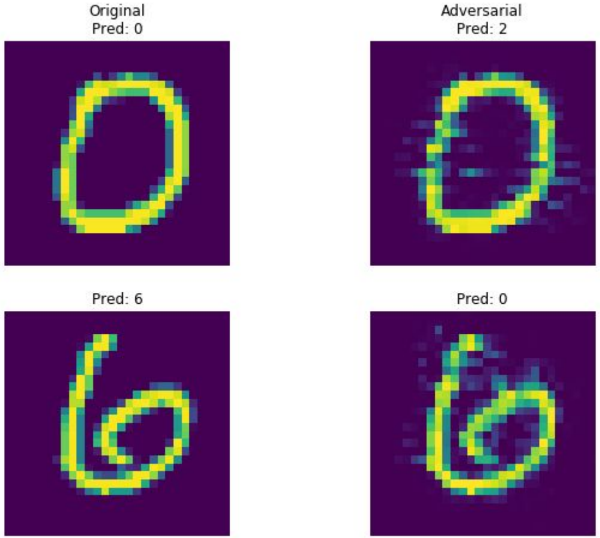
Outlier Detection on CIFAR10



Adversarial Detection Demos

KFServing MNIST Model with Alibi:Detect VAE Adversarial Detector

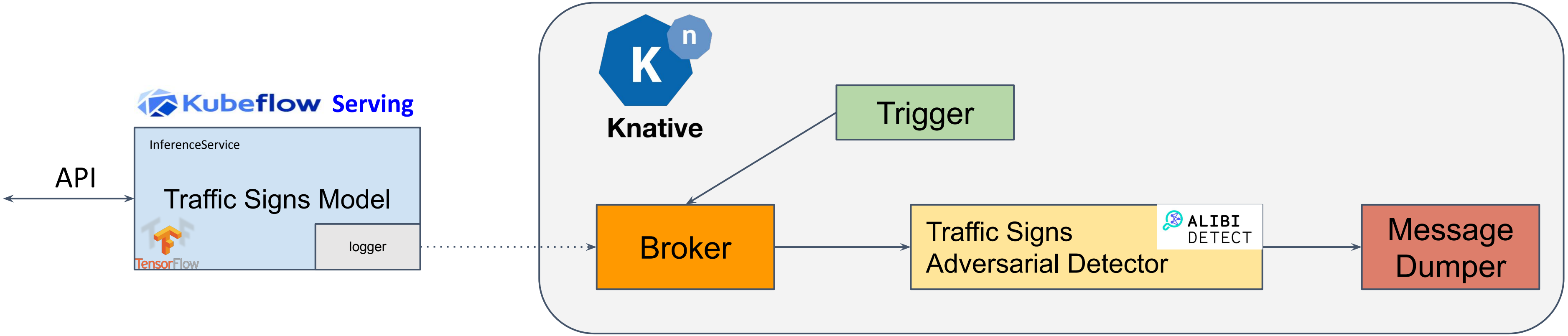
https://github.com/SeldonIO/alibi-detect/tree/master/integrations/samples/kfserving/ad-mnist



KFServing Traffic Signs Model with Alibi:Detect VAE Adversarial Detector

https://github.com/SeldonIO/alibi-detect/tree/master/integrations/samples/kfserving/ad-signs

Adversarial Detection on Traffic Signs



Adversarial Attack, Detection and Defense Mechanisms: Resources

Adversarial Robustness 360 ↳ (ART)

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

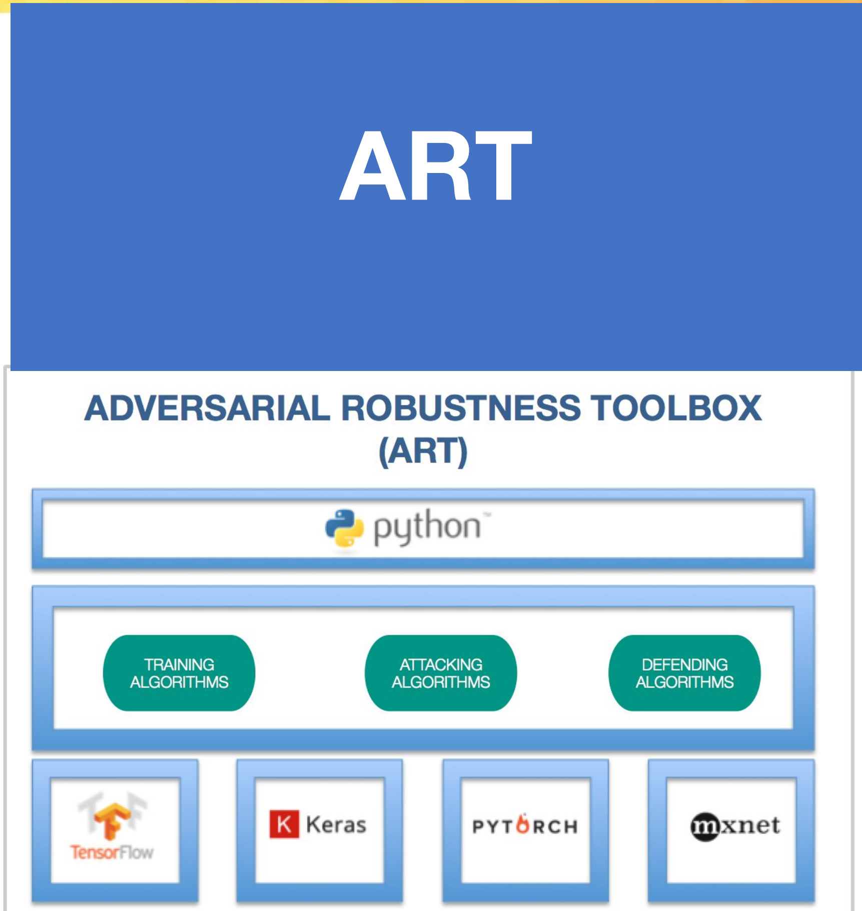
ART is a library dedicated to adversarial machine learning. Its purpose is to allow rapid crafting and analysis of **attack, defense and detection methods** for machine learning models. Applicable domains include finance, self driving vehicles etc.

The Adversarial Robustness Toolbox provides an implementation for many state-of-the-art methods for attacking and defending classifiers.

Toolbox: Attacks, defenses, and metrics

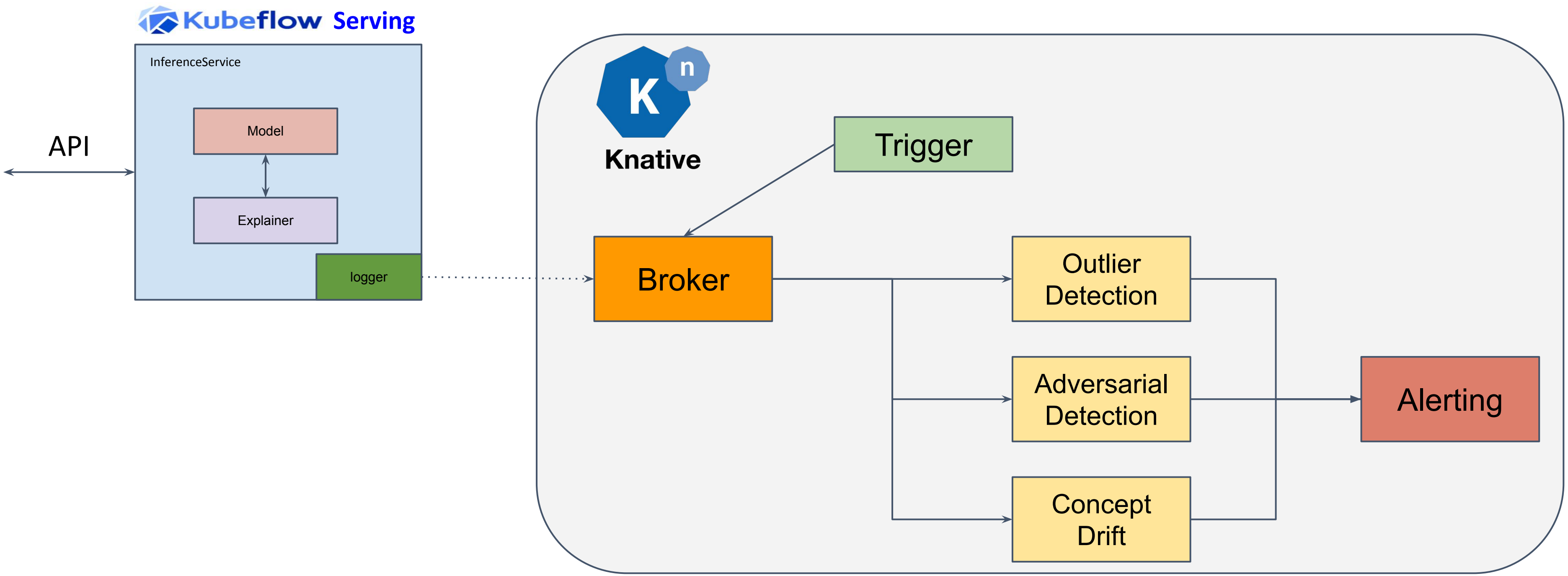
Evasion attacks
Defense methods
Detection methods
Robustness metrics

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



Summary and Roadmap

Production ML Architecture



Open Source Projects

- ML Inference - KFServing - Seldon Core	https://github.com/kubeflow/kfserving https://github.com/SeldonIO/seldon-core
- Model Explanations - Seldon Alibi - IBM AI Explainability 360	https://github.com/seldonio/alibi https://github.com/IBM/AIX360
- Outlier and Adversarial Detection and Concept Drift - Seldon Alibi-detect	https://github.com/seldonio/alibi-detect
- Adversarial Attack, Detection and Defense - IBM Adversarial Robustness 360	https://github.com/IBM/adversarial-robustness-toolbox

Related Tech Kubecon Talks

Tuesday, November 19 • 2:25pm - 3:00pm

Introducing KFServing: Serverless Model Serving on Kubernetes - Ellis Bigelow, Google & Dan Sun, Bloomberg

Wednesday, November 20 • 5:20pm - 5:55pm

Serverless Platform for Large Scale Mini-Apps: From Knative to Production - Yitao Dong & Ke Wang, Ant Financial

Wednesday, November 20 • 11:50am - 12:25pm

From Brownfield to Greenfield: Istio Service Mesh Journey at Freddie Mac - Shriram Rajagopalan, Tetrate & Lixun Qi, Freddie Mac

Thursday, November 21 • 10:55am - 12:25pm

CloudEvents - Intro, Deep-Dive and More! - Doug Davis, IBM