Meeting of the LF AI & Data Technical Advisory Council (TAC)

November 18, 2021

LF AI & DATA

Antitrust Policy

- Linux Foundation meetings involve participation by industry competitors, and it is the intention of the Linux Foundation to conduct all of its activities in accordance with applicable antitrust and competition laws. It is therefore extremely important that attendees adhere to meeting agendas, and be aware of, and not participate in, any activities that are prohibited under applicable US state, federal or foreign antitrust and competition laws.

- Examples of types of actions that are prohibited at Linux Foundation meetings and in connection with Linux Foundation activities are described in the Linux Foundation Antitrust Policy available at http://www.linuxfoundation.org/antitrust-policy. If you have questions about these matters, please contact your company counsel, or if you are a member of the Linux Foundation, feel free to contact Andrew Updegrove of the firm of Gesmer Undergone LLP, which provides legal counsel to the Linux Foundation.

Recording of Calls

Reminder:

TAC calls are recorded and available for viewing on the TAC Wiki

Reminder: LF AI & Data Useful Links

- Web site: lfaidata.foundation
- Wiki: wiki.lfaidata.foundation
- GitHub: github.com/lfaidata
- Landscape: https://landscape.lfaidata.foundation or https://l.lfaidata.foundation
- Mail Lists: https://lists.lfaidata.foundation
- Slack: https://slack.lfaidata.foundation
- Youtube: https://www.youtube.com/channel/UCfasaeqXJBCAJMNO9HcHfbA
- LF AI Logos: https://github.com/lfaidata/artwork/tree/master/lfaidata
- LF AI Presentation Template: https://drive.google.com/file/d/1eiDNJvXCqSZHT4Zk_-czASlz2GTBRZk2/view?usp=sharing

- Events Page on LF AI Website: https://lfaidata.foundation/events/
- Events Calendar on LF AI Wiki (subscribe available): https://wiki.lfaidata.foundation/pages/viewpage.action?pageId=12091544
- Event Wiki Pages: https://wiki.lfaidata.foundation/display/DL/LF+AI+Data+Foundation+Events

Agenda

- Roll Call (2 mins)
- Approval of Minutes from previous 2 meetings (2 mins)
- Kserve new project in incubation (40 minutes)
- LF AI General Updates (2 min)
- Open Discussion (2 min)

TAC Voting Members

* = still need backup specified on wiki

Board Member	Contact Person	Email
AT&T	Anwar Atfab*	anwar@research.att.com
Baidu	Ti Zhou	zhouti@baidu.com
Ericsson	Rani Yadav-Ranjan*	rani.yadav-ranjan@ericsson.com
Huawei	Huang Zhipeng	huangzhipeng@huawei.com
IBM	Susan Malaika	malaika@us.ibm.com
Nokia	Jonne Soininen	jonne.soininen@nokia.com
OPPO	Jimin Jia*	jiajimin@oppo.com
SAS	Nancy Rausch	nancy.rausch@sas.com
Tech Mahindra	Amit Kumar	Kumar_Amit@techmahindra.com
Tencent	Bruce Tao	brucetao@tencent.com
Zilliz	Jun Gu	jun.gu@zilliz.com
ZTE	Wei Meng	meng.wei2@zte.com.cn
Graduate Project	**Contact Person**	**Email**
Acumos	Nat Subramanian	natarajan.subramanian@techmahindra.com
Angel	Bruce Tao	brucetao@tencent.com
Egeria	Mandy Chessell	mandy_chessell@uk.ibm.com
Horovod	Travis Addair*	taddair@uber.com
Milvus	Xiaofan Luan	xiaofan.luan@zilliz.com
ONNX	Jim Spohrer (Chair of TAC)	spohrer@us.ibm.com
Pyro	Fritz Obermeyer*	fritz.obermeyer@gmail.com

Minutes approval

LF AI & DATA

Approval of October 21st, 2021 Minutes

Draft minutes from the October 21th TAC call were previously distributed to the TAC members via the mailing list

Proposed Resolution:

- That the minutes of the October 21th meeting of the Technical Advisory Council of the LF AI & Data Foundation are hereby approved.

Approval of November 4th, 2021 Minutes

Draft minutes from the November 4th TAC call were previously distributed to the TAC members via the mailing list

Proposed Resolution:

- That the minutes of the November 4th meeting of the Technical Advisory Council of the LF AI & Data Foundation are hereby approved.

KServe Proposal to Incubate in LF AI & Data

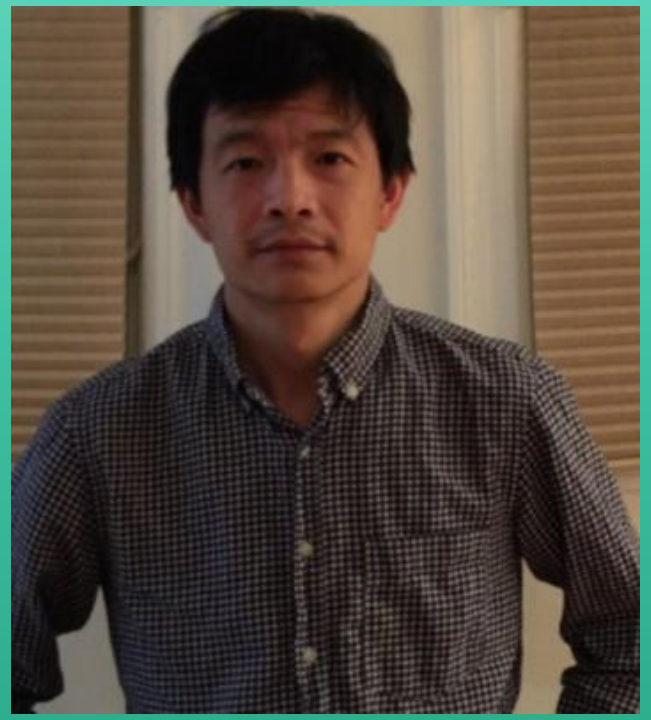
https://github.com/lfai/proposing-projects/blob/master/proposals/kserve.md

11/18/2021

LF AI & DATA

Serving Machine Learning models Serverlessly at Scale Using KServe

Animesh Singh
IBM



Dan Sun
Bloomberg

Enterprises are still struggling to scale AI beyond experimentation

88% of corporate AI initiatives are struggling to move beyond test stages

Source: Artificial Intelligence, The Next Digital Frontier. McKinsey Global Institute, 2017

"I have no quantification of the business impact of my AI solutions"

"My data scientists have developed some models, but I do not know if they always achieve the best possible solution"

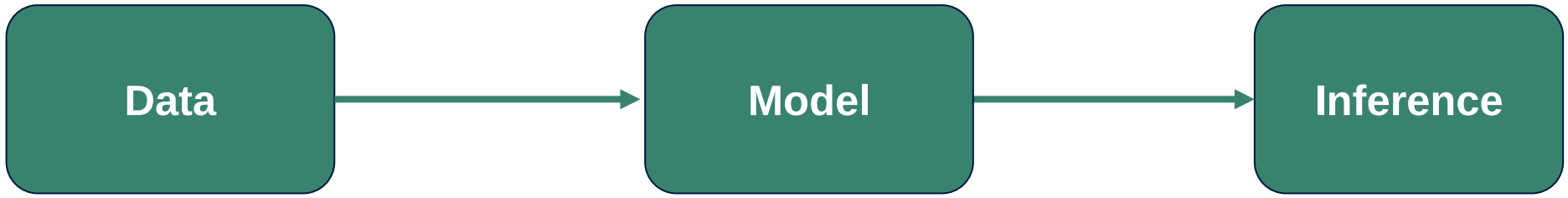
"I have an analytics team that has executed multiple PoCs, but none of that has made it into production"

"We've deployed multiple algorithms, but we have not seen any improvement in our business KPIs"

"We find it difficult finding and hiring the right AI talent"

"My business users **do not trust** the results of my AI applications, and they do not get used"

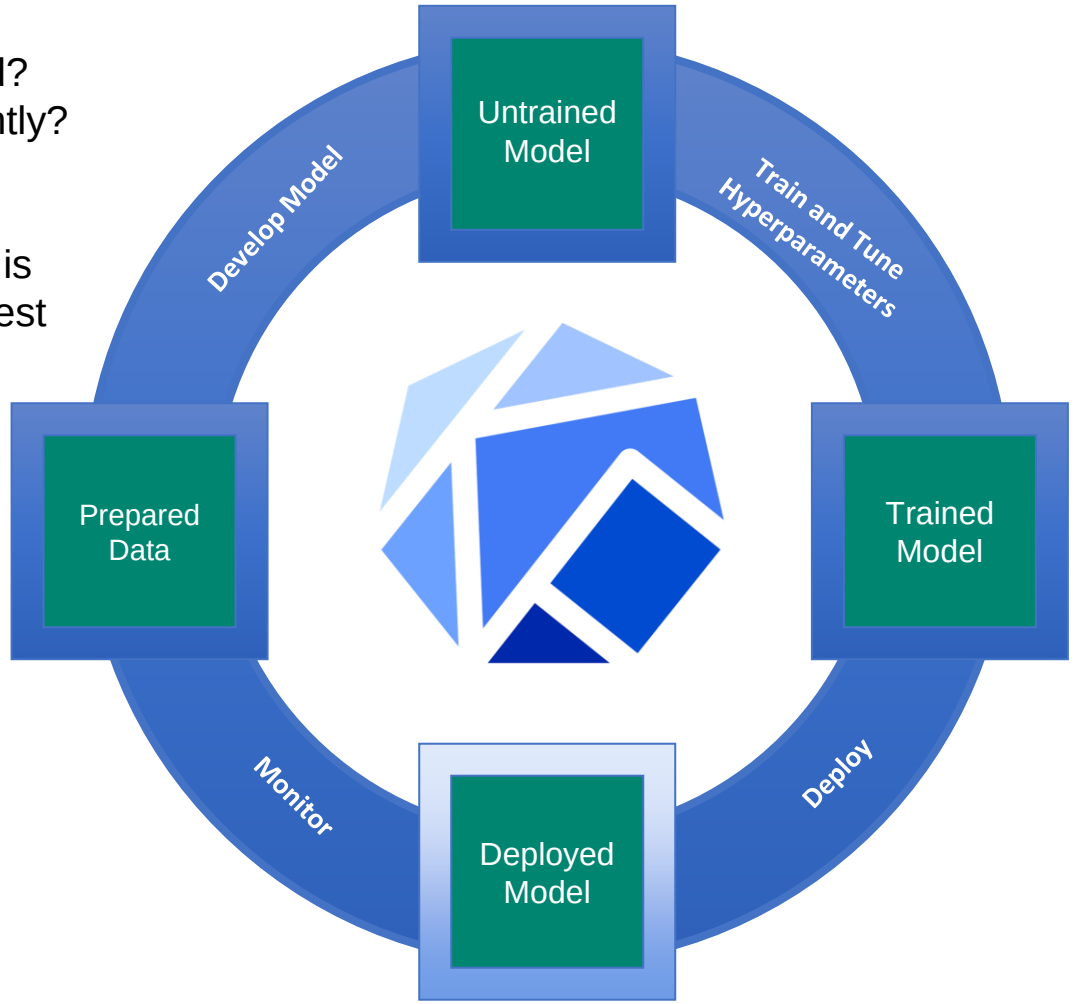
Problem: production grade inference



- Hi my magic model, given these two sentences please tell me their similarities!
- Hi my magic model, given this news article please tell me it’s topics!
- Hi my magic model, given last few songs I heard please play me the next songs!
- Hi my magic model, route my resume to the prospective employer!
- …..

Production grade inference: How hard could it be?

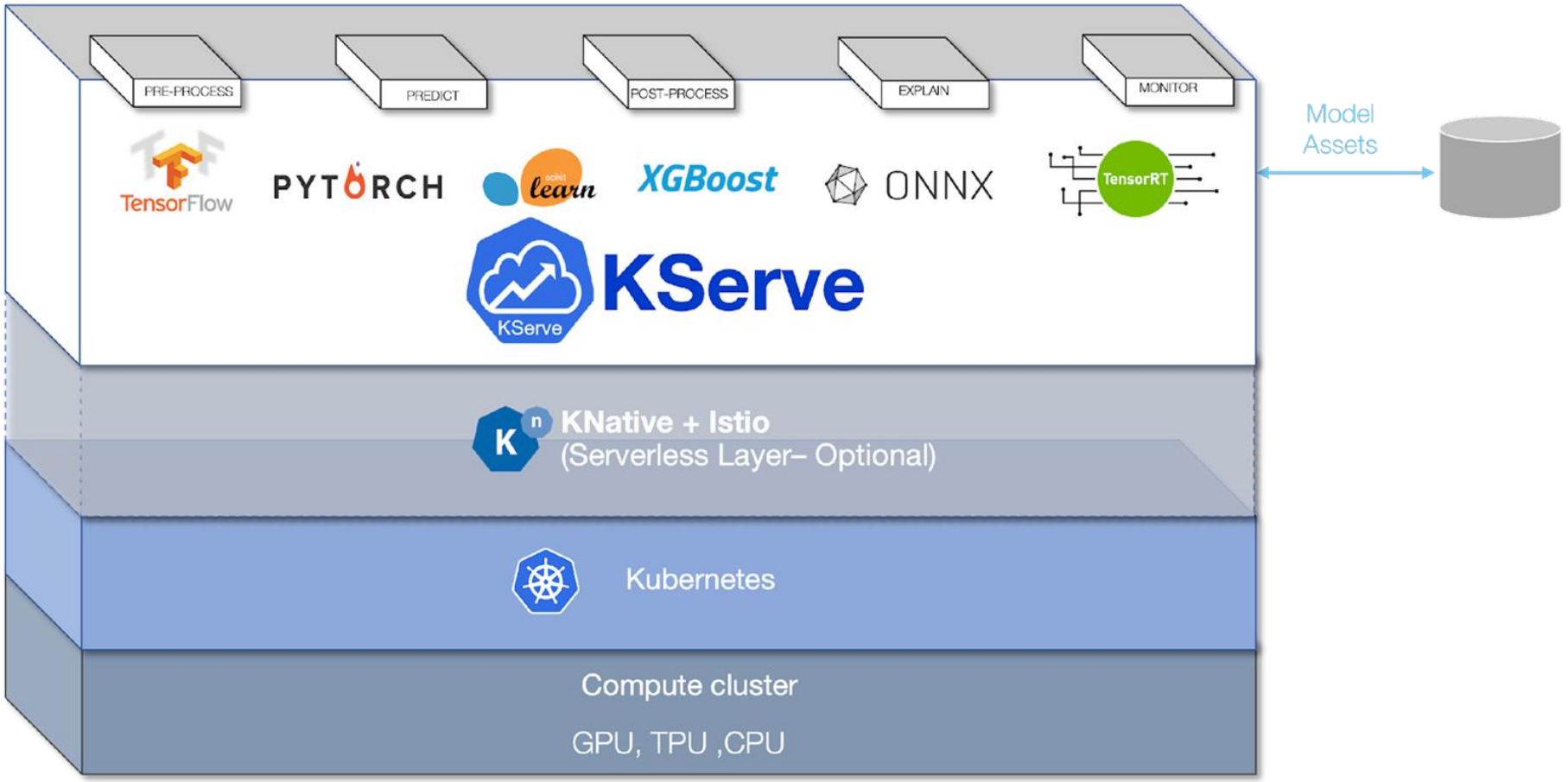
- ❑ Cost:
  Is the model over or under scaled?
  Are resources being used efficiently?
- ❑ Monitoring:
  Are the endpoints healthy? What is the performance profile and request trace?
- ❑ Rollouts:
  Is this rollout safe? How do I roll back? Can I test a change without swapping traffic?
- ❑ Protocol Standards: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



- ❑ How do I handle batch predictions?
- ❑ How do I leverage standardized Data Plane protocol so that I can move my model across MLServing platforms?
- ❑ 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

Here comes KServe!

Highly scalable and standards based Model Inferencing Platform on Kubernetes for Trusted AI



KServe Overview

- KServe is a **Model Inferencing Platform** on **Kubernetes.** Run anywhere **Kubernetes** runs, never worry about **vendor lock-in**.

- Provides **performant, standardized inference protocol** across **ML frameworks**.

- Support modern **serverless inference workload** with **Autoscaling** including **Scale to Zero on GPU**.

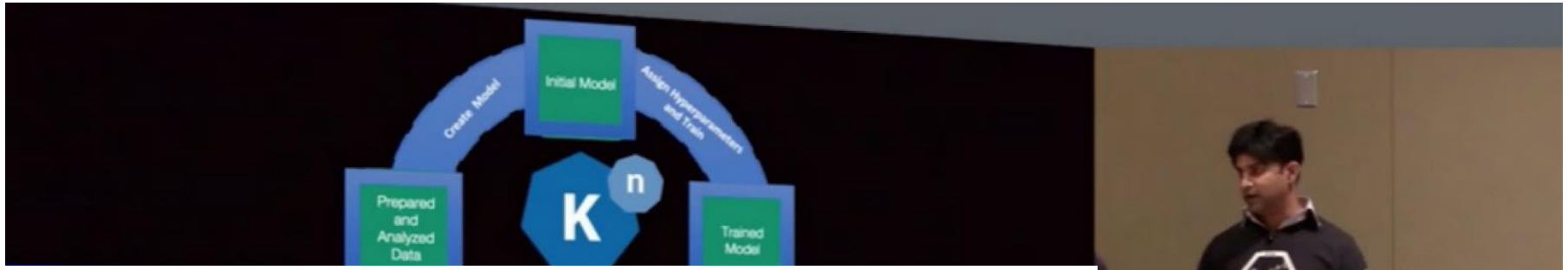
- **Simple** and **Pluggable production serving** for **production ML serving** including **prediction, pre/post processing, monitoring and explainability**..

- **Advanced deployments** with **canary rollout, experiments, ensembles and transformers**.

The story of KServe

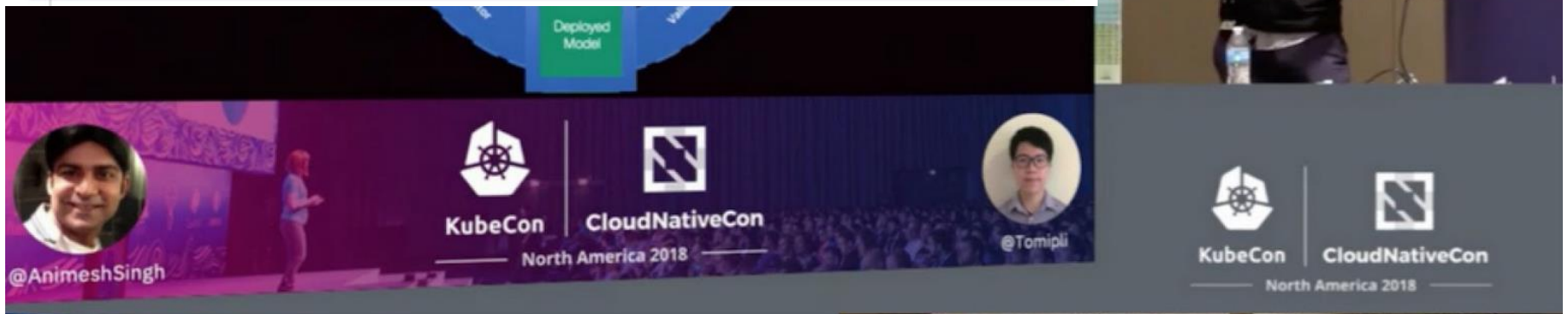
Previously KFServing

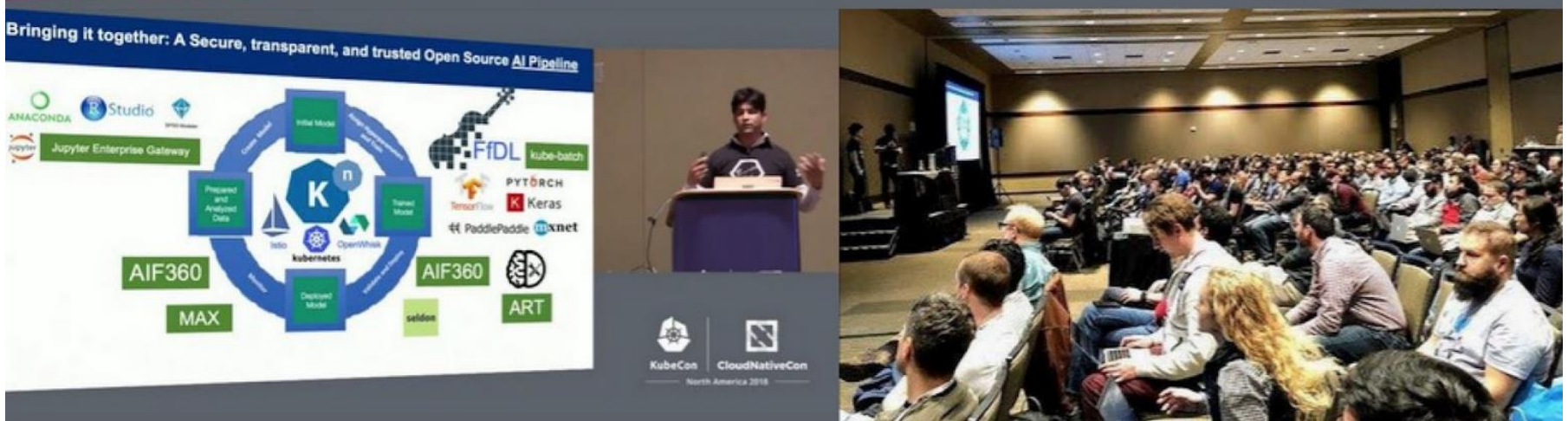
KubeCon Dec 2018



Tuesday, December 11 • 2:35pm - 3:10pm

Machine Learning Model Serving and Pipeline Using KNative - Animesh Singh & Tommy Li, IBM





July 2019

Animesh Singh
@AnimeshSingh

Great meeting with @KubeFlow team at the #KFServing Summit! Bloomberg, Nvidia, Google, IBM, Microsoft and Seldon cooking something awesome! Thanks @ellisbigelow for getting us all together, organizing it impeccably and driving deep technical discussions around Model Serving!

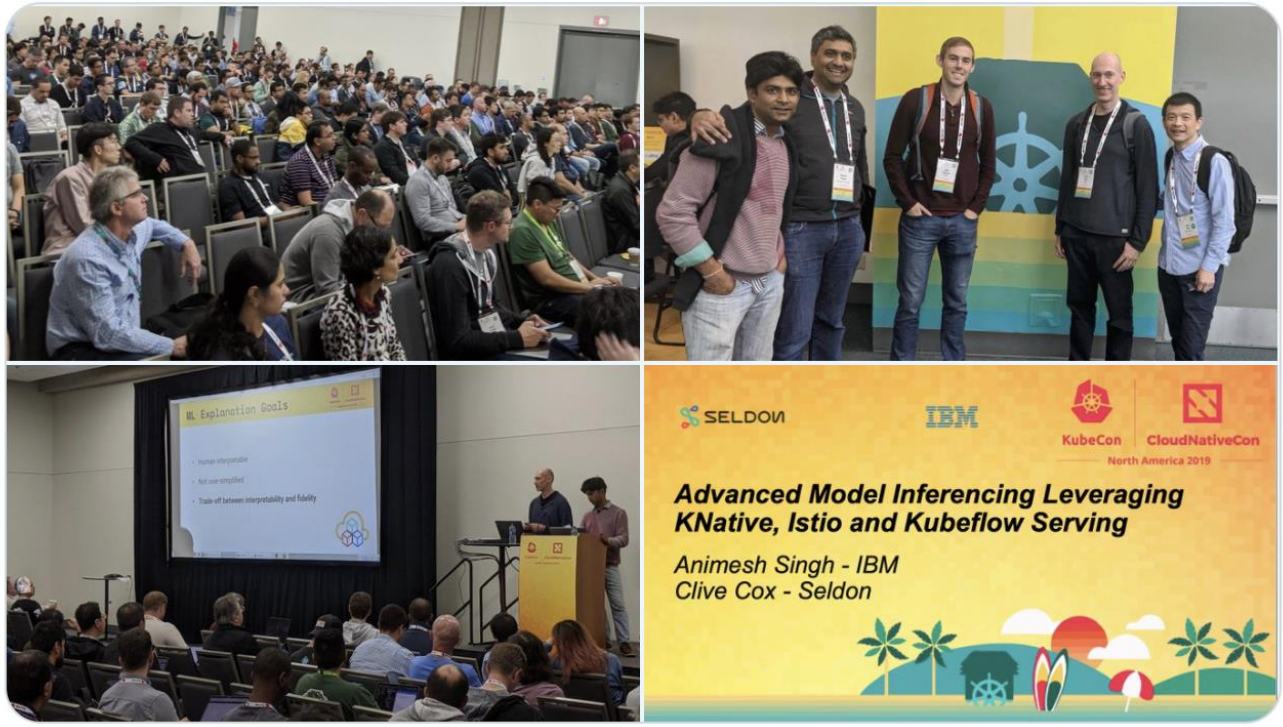
Nov 2019

Animesh Singh
@AnimeshSingh

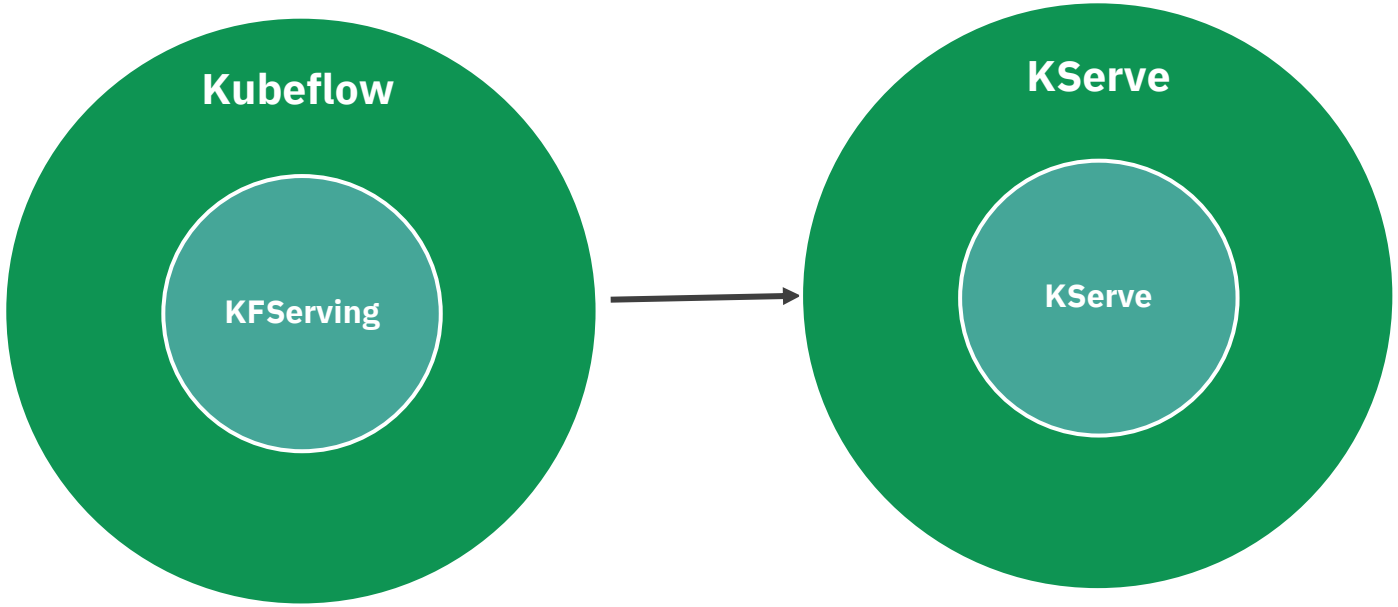
Thanks for coming and making the talk standing room only , and great interaction afterwards. Slides from our talk are available here.

slideshare.net/AnimeshSingh/a...

It was great to meet the team! Looking forward for more feedback on @Kubeflow #KubeCon

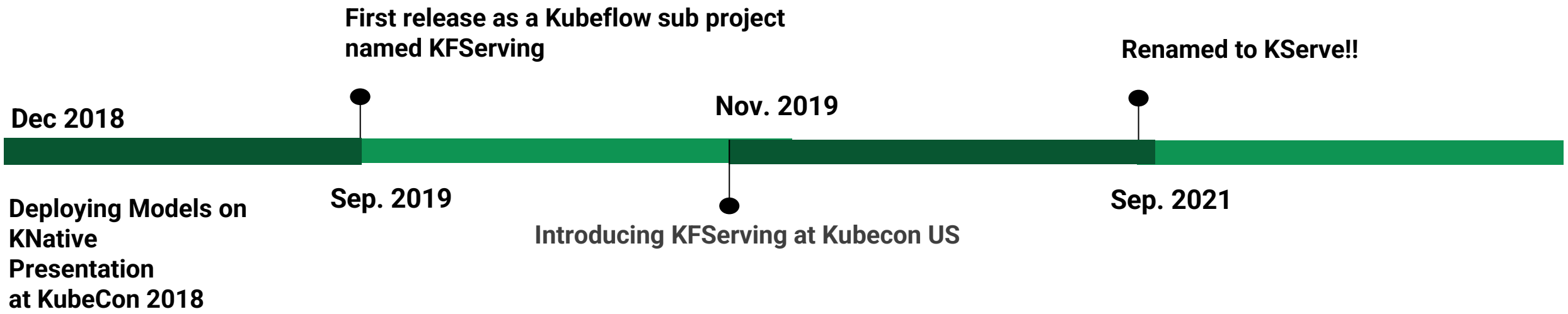


Sep 2021



https://blog.kubeflow.org/release/official/2021/09/27/kfserving-transition.html

The story of KServe




AI/ML community
Products with the word "Quantum"
Products with the word "Flow"
made with mematic

KServe Contributors

KServe Standardized Inference Protocol

- HTTP/GRPC
- Standard Inference protocol that supports multiple model server
  - Triton
  - TorchServe
  - MLServer

- https://kserve.github.io/website/modelserving/data_plane/

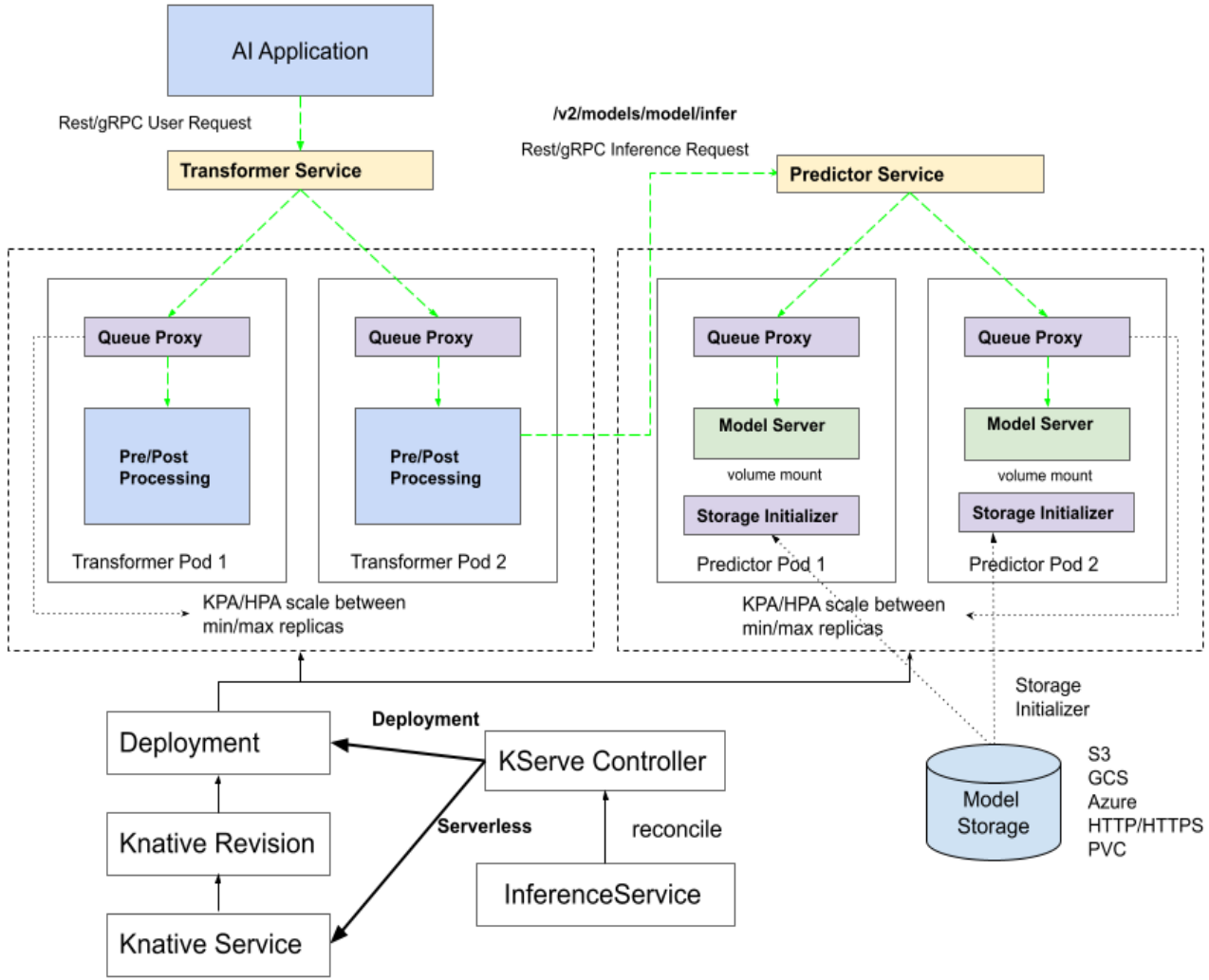
HTTP Protocol

- Health:
  - GET v2/health/live
  - GET v2/health/ready
  - GET v2/models/${MODEL_NAME}[/versions/${MODEL_VERSION}]/ready
- Server Metadata:
  - GET v2
- Model Metadata:
  - GET v2/models/${MODEL_NAME}[/versions/${MODEL_VERSION}]
- Inference:
  - POST v2/models/${MODEL_NAME}[/versions/${MODEL_VERSION}]/infer

gRPC Protocol

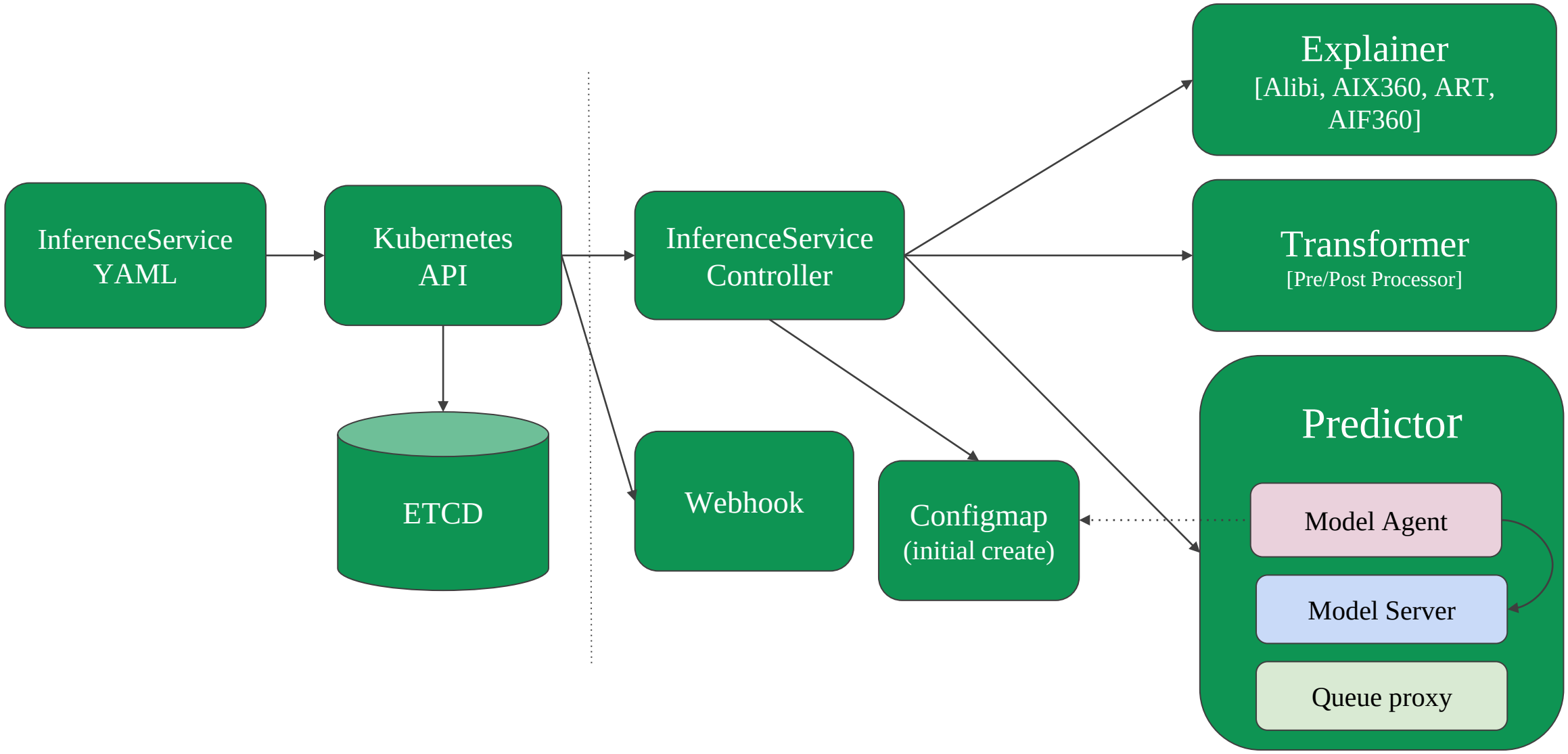
- Health:
  - rpc ServerLive(ServerLiveRequest) returns (ServerLiveResponse) {}
  - rpc ServerReady(ServerReadyRequest) returns (ServerReadyResponse) {}
  - rpc ModelReady(ModelReadyRequest) returns (ModelReadyResponse) {}
- Server Metadata:
  - rpc ServerMetadata(ServerMetadataRequest) returns (ServerMetadataResponse) {}
- Model Metadata:
  - rpc ModelMetadata(ModelMetadataRequest) returns (ModelMetadataResponse) {}
- Inference:
  - rpc ModelInfer(ModelInferRequest) returns (ModelInferResponse) {}

Single Model Serving



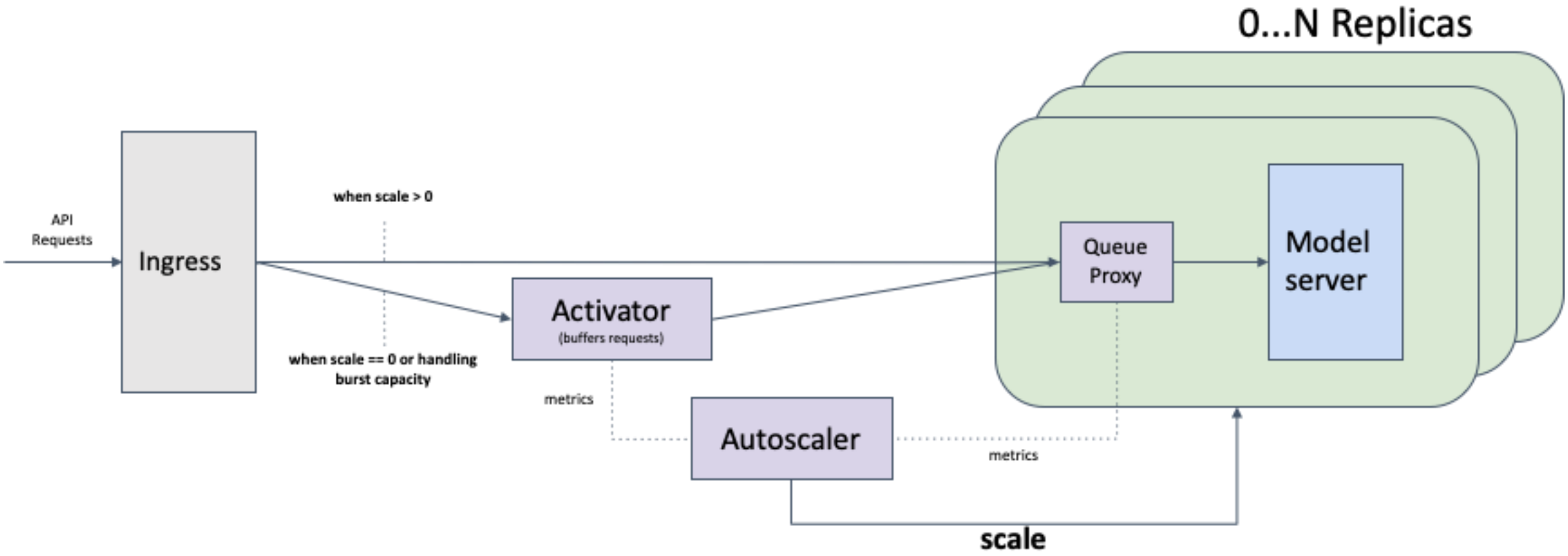

```
apiVersion: "serving.kserve.io/v1beta1"
kind: "InferenceService"
metadata:
 name: "sklearn-feast-transformer"
spec:
 transformer:
  containers:
  - image: kserve/driver-transformer:latest
    name: driver-container
    command:
    - "python -m driver_transformer"
    args:
    - --entity_ids
    - driver_id
    - --feature_refs
    - driver_hourly_stats:acc_rate
    - driver_hourly_stats:avg_daily_trips
    - driver_hourly_stats:conv_rate
 predictor:
  sklearn:
    storageUri: "gs://pv-kfserving/driver"
```

KServe Control Plane



Serverless Inference

- Scale based on # in-flight requests against expected concurrency
- Simple solution for heterogeneous ML inference autoscaling

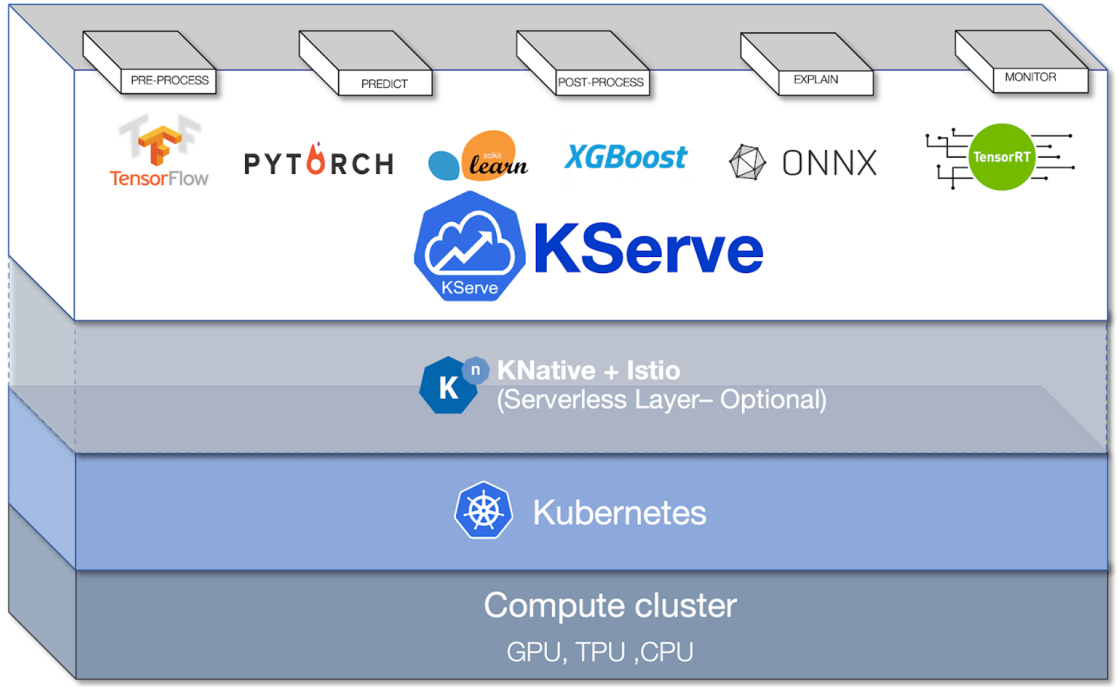


LFAI Trusted AI OSS projects available in KServe

AI Fairness 360

AI Explainability 360

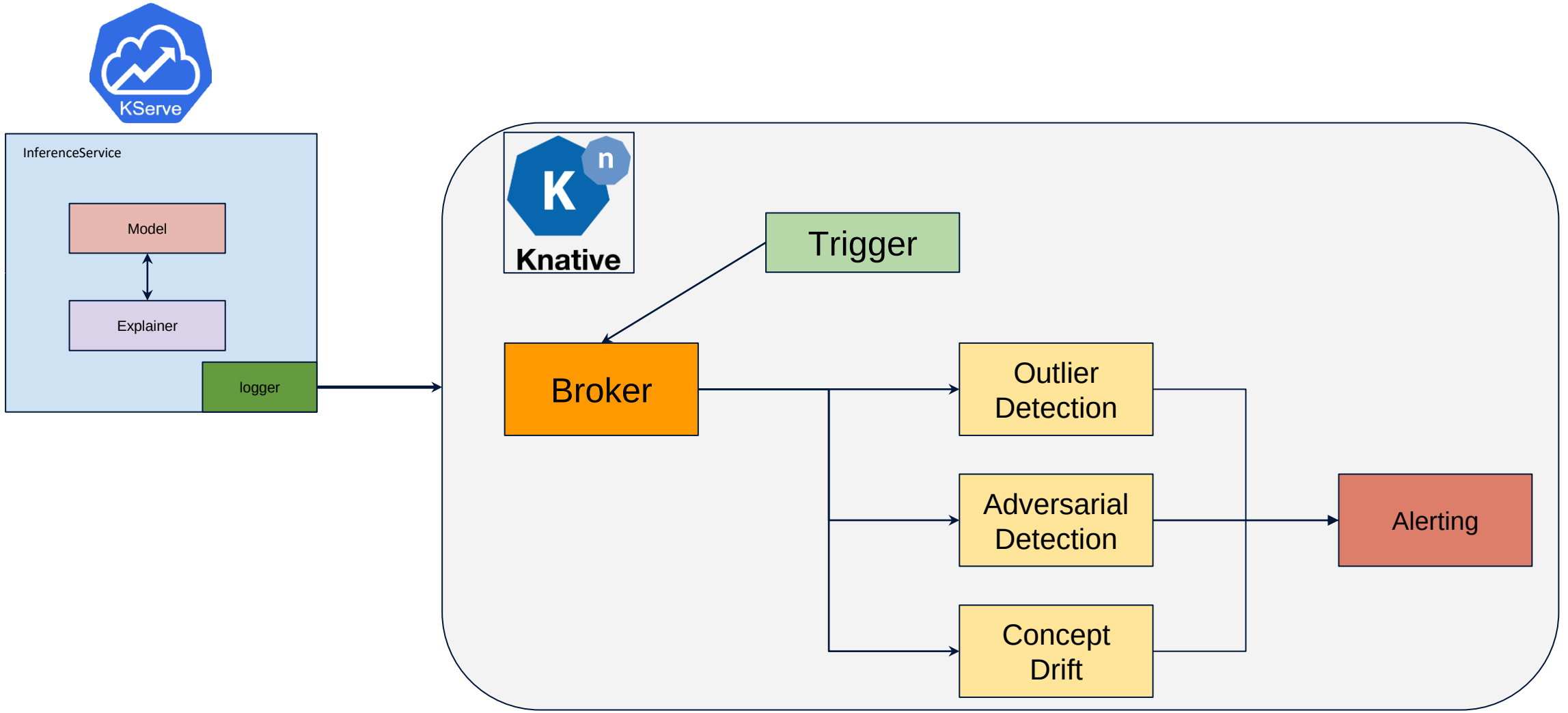
Adversarial Robustness Toolbox



LFAI

https://ai-fairness-360.org/
https://ai-explainability-360.org/
https://adversarial-robustness-toolbox.org/

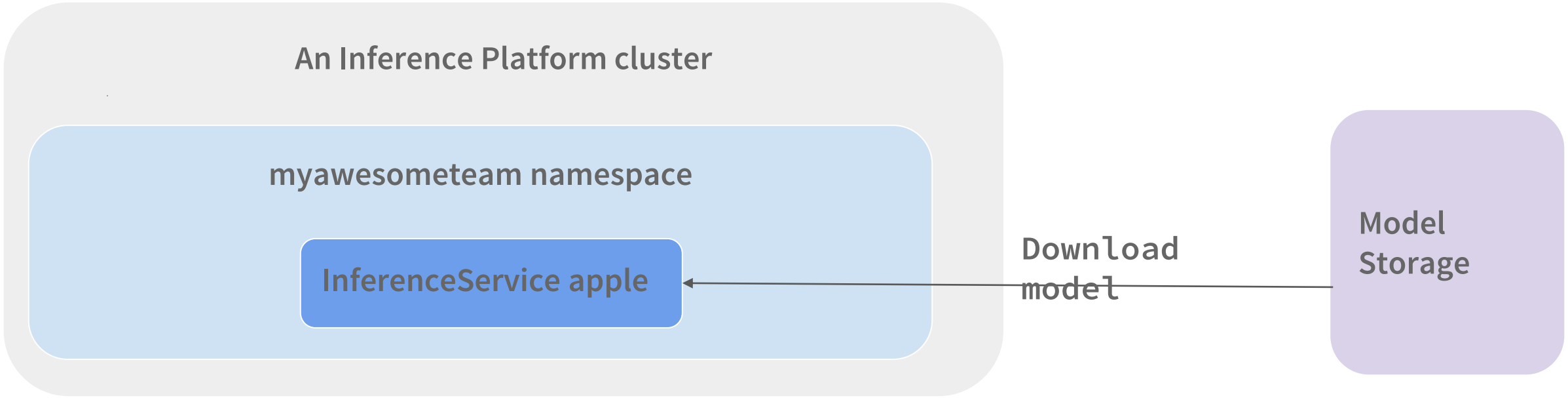
And more advanced Metrics



New scalability problem

Deploy large number of models in production

Current KServe model deployment



https://apple-myawesometeam.bloomberg.com/v1/models/apple:predict

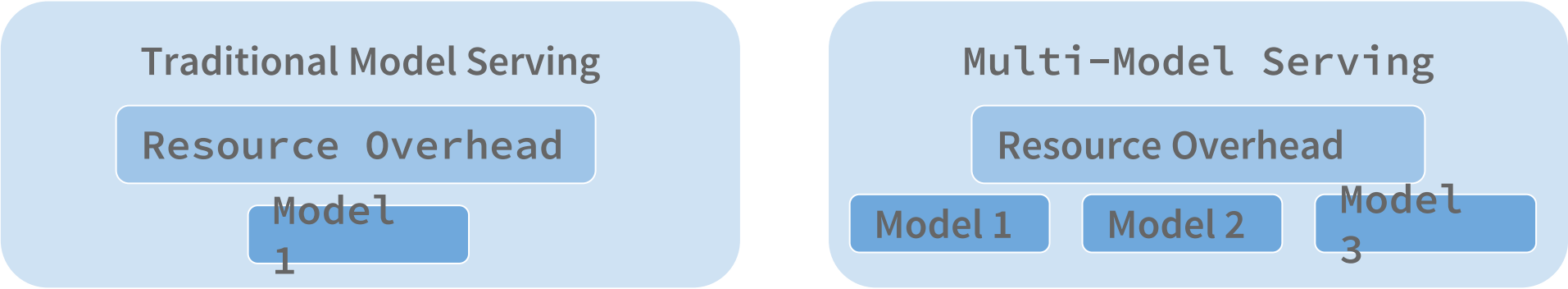
Sending request data: {"instances": [[6.8, 2.8, 4.8, 1.4]]}
Got response code 200, content: '{"predictions": [1]}'

Scalability limitations

- Compute Resource limitation
- Maximum pod limitation
- Maximum IP address limitation
- ...

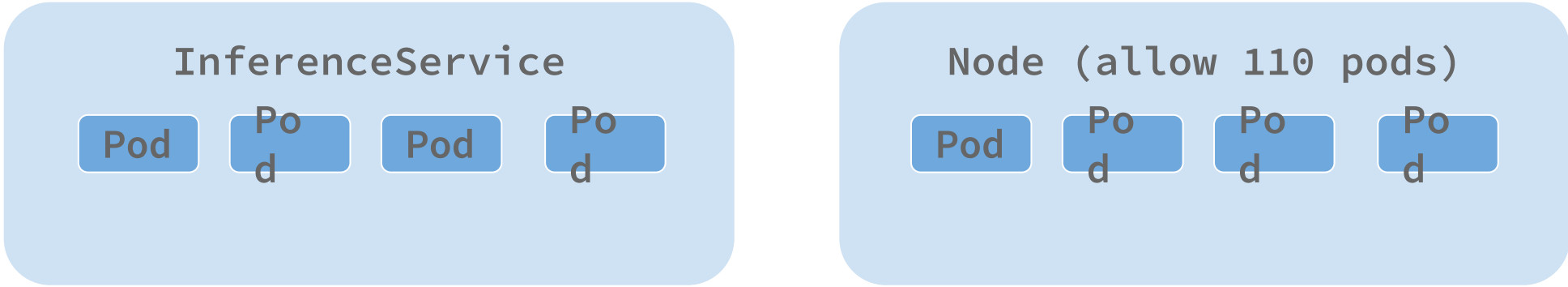
Compute Resource limitations

- Each InferenceService has about 0.5 CPU and 0.5 GB memory overhead
- Deploy 10 models each with 2 pods -> 1 CPU and 1 GB overhead per model
- Load 10 models into one InferenceService -> 0.1 CPU and 0.1 GB overhead per model



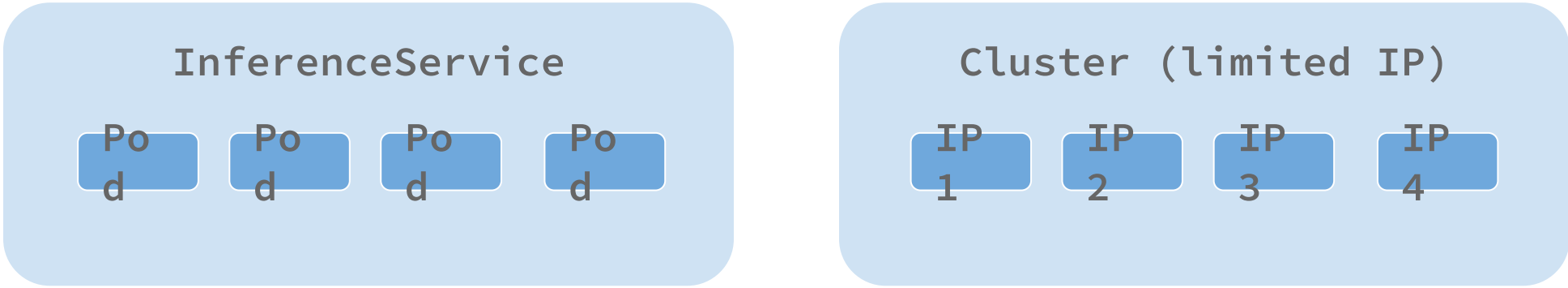
Maximum pod limitations

- Kubernetes default pod limit: 110 per node
- Kubernetes scalability best practice: at most 100 pods per node
- A 50 nodes cluster can deploy about 1000 to 4000 m



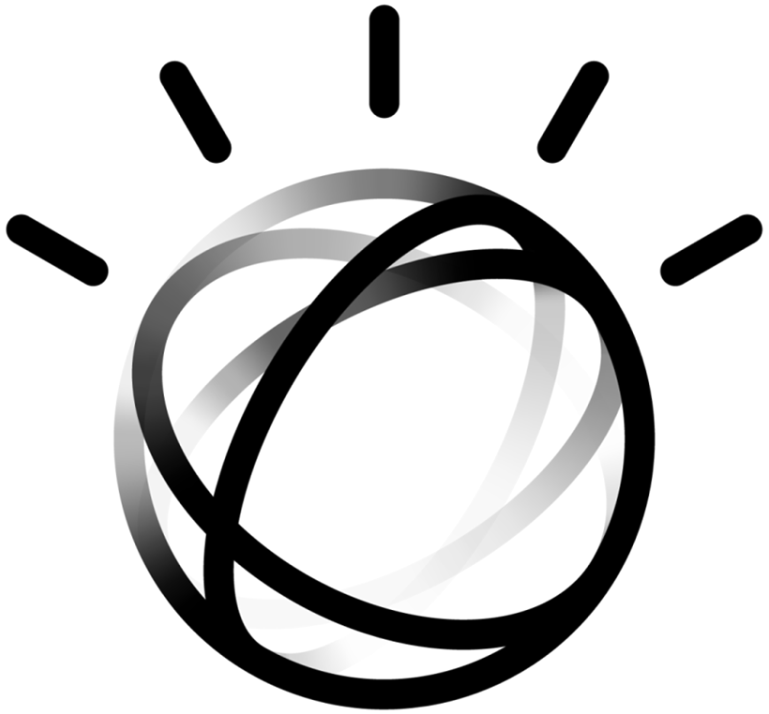
Maximum IP address limitations

- Each pod has an independent IP address
- IPs are assigned to new models, replicas of models, transformers, explainers and other controller plane pods in the cluster.



Enter ModelMesh
by
IBM Watson

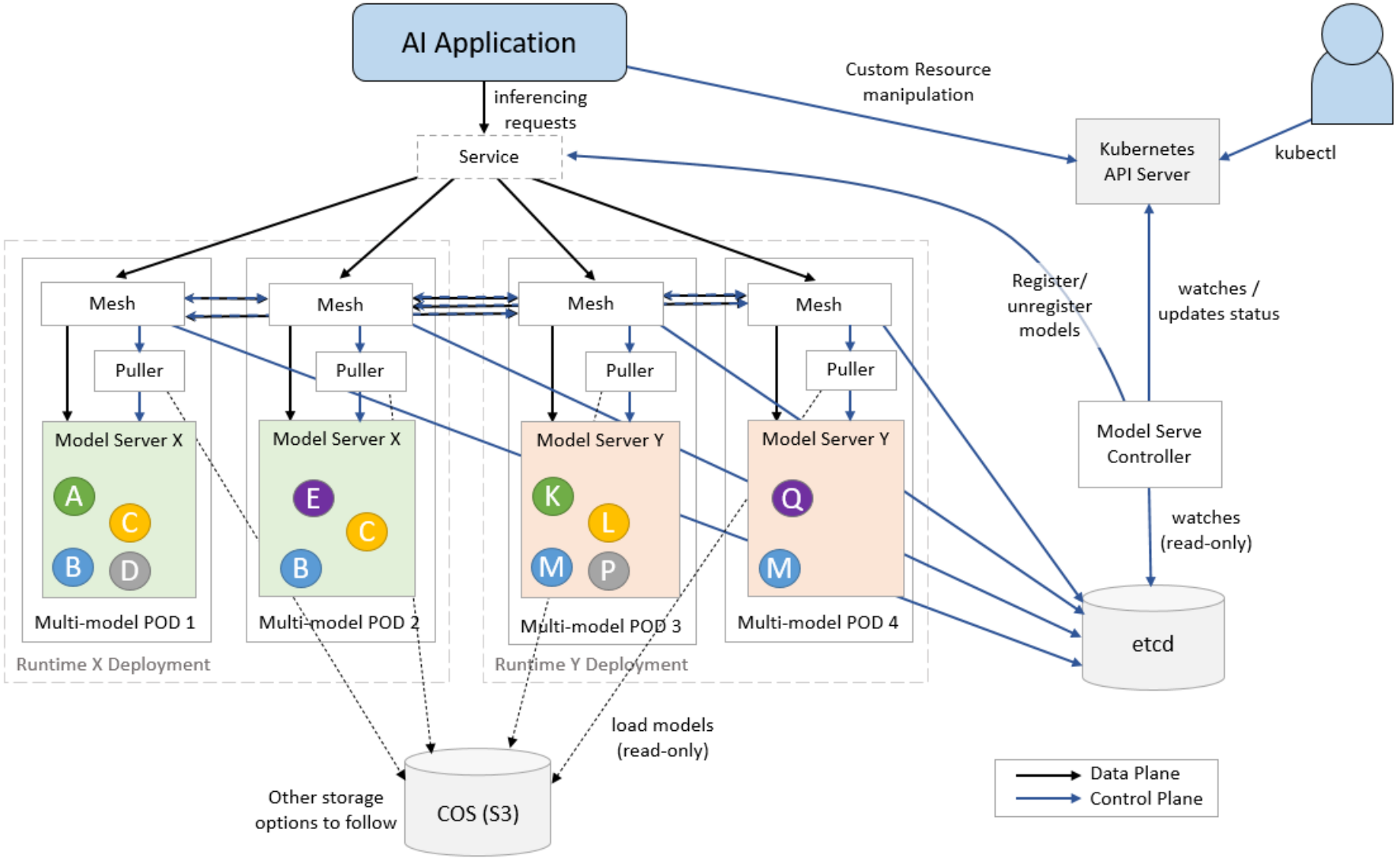
ModelMesh for Multi-Model Serving



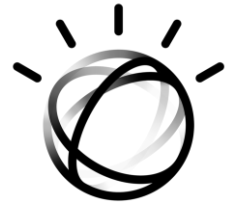
- ModelMesh, model serving management layer for IBM Watson products.
- Running successfully in production for several years, ModelMesh underpins most of the Watson cloud services, including Watson Assistant, Watson Natural Language Understanding, and Watson Discovery.
- Designed for high-scale, high-density, and frequently-changing model use cases.
- ModelMesh intelligently loads and unloads AI models to and from memory to strike an intelligent trade-off between responsiveness to users and their computational footprint.

ModelMesh Architecture

Framework for high-scale, high-density and frequently-changing model use cases.



ModelMesh Components



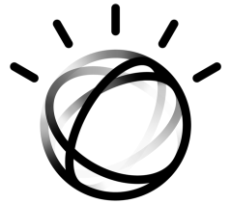
Core Components

- github.com/kserve/modelmesh-serving - Model serving controller.
- github.com/kserve/modelmesh - ModelMesh containers used for orchestrating model placement and routing.

Runtime Adapters

- github.com/kserve/modelmesh-runtime-adapter - the containers which run in each model serving pod and act as an intermediary between ModelMesh and third-party model-server containers. Incorporates the "puller" logic which is responsible for retrieving models from storage.

Serving Runtimes



Out-of-the-box integration with the following model servers:

Triton Inference Server

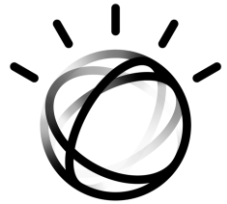
NVIDIA's server for frameworks like TensorFlow, PyTorch, TensorRT, or ONNX.

MLServer

Seldon's Python-based server for frameworks like SKLearn, XGBoost, or LightGBM.

ServingRuntime custom resources can be used to add support for other existing or custom-built model servers.

ModelMesh Features



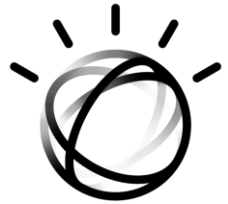
Cache Management and HA

- The clusters of multi-model server pods are managed as a distributed LRU cache, with available capacity automatically filled with registered models.

- ModelMesh decides when and where to load and unload copies of models based on usage recency and current request volumes - if a particular model is under heavy load it will be scaled across more pods.

- ModelMesh also acts as a router, balancing inference requests between all copies of the target model, coordinating just-in-time loads of models that aren't currently in memory, and retrying/re-routing failed requests.

ModelMesh Features



Intelligent Placement and Queuing

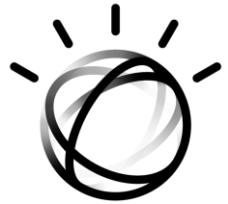
- Placement of models into the existing model-server pods is done in such a way to balance both the “cache age” across the pods as well as the request load. Heavily used models are placed on less-utilized pods and vice versa.

- Concurrent model loads are constrained/queued to minimize impact to runtime traffic, and priority queues are used to allow urgent requests to jump the line (i.e. cache misses where an end-user request is waiting).

Resiliency

- Failed model loads are automatically retried in different pods and after longer intervals, to facilitate automatic recovery, for example after a temporary storage outage.

ModelMesh Features



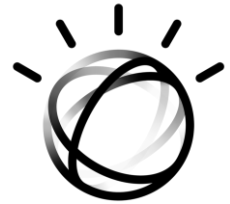
Operational Simplicity

- ModelMesh deployments can be upgraded as if they were homogeneous - it manages propagation of models to new pods during a rolling update automatically without any external orchestration required and without any impact to inference requests.

- There is no central controller involved in model management decisions - the logic is decentralized with lightweight coordination that makes use of etcd.

- Stable “v-model” endpoints are used to provide seamless transition between concrete model versions. ModelMesh ensures that the new model has loaded successfully before switching the pointer to route requests to the new version.

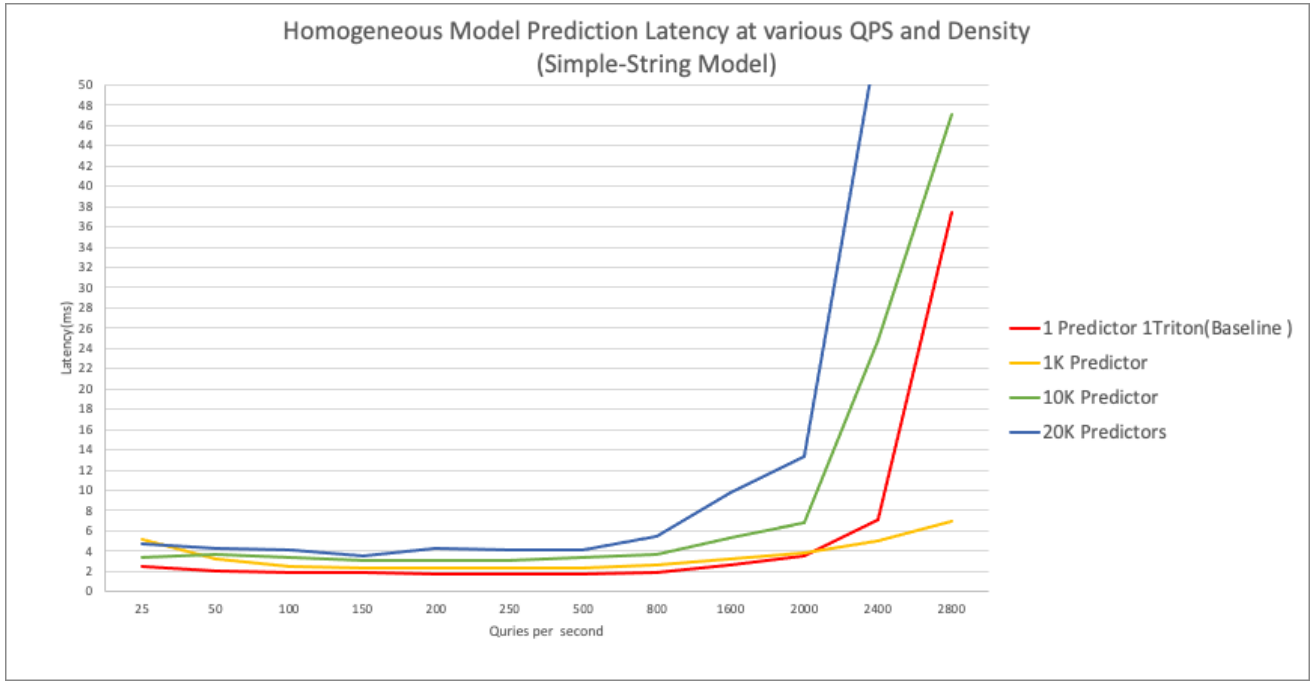
ModelMesh Features



Scalability

ModelMesh supports hundreds of thousands of models in a single production deployment of 8 pods, by over-committing the aggregate available resources and intelligently keeping a most-recently-used set of models loaded across them in a heterogeneous manner.

- We did some sample tests to determine the density and scalability for ModelMesh on an instance deployed on a single worker node (8vCPU x 64GB) Kubernetes cluster.
- The tests were able to pack 20K simple-string (700Bytes) models into only two serving runtime pods, which were load tested by sending thousands of concurrent inference requests to simulate a high traffic scenario.
- All loaded models responded with single digit millisecond latency.



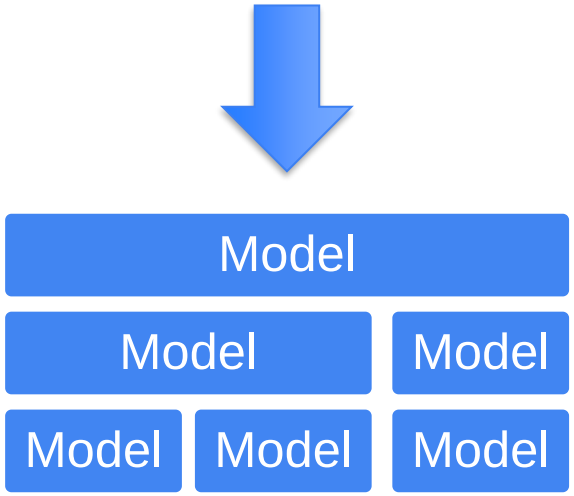
ModelMesh and KServe: Better Together!

Announcing

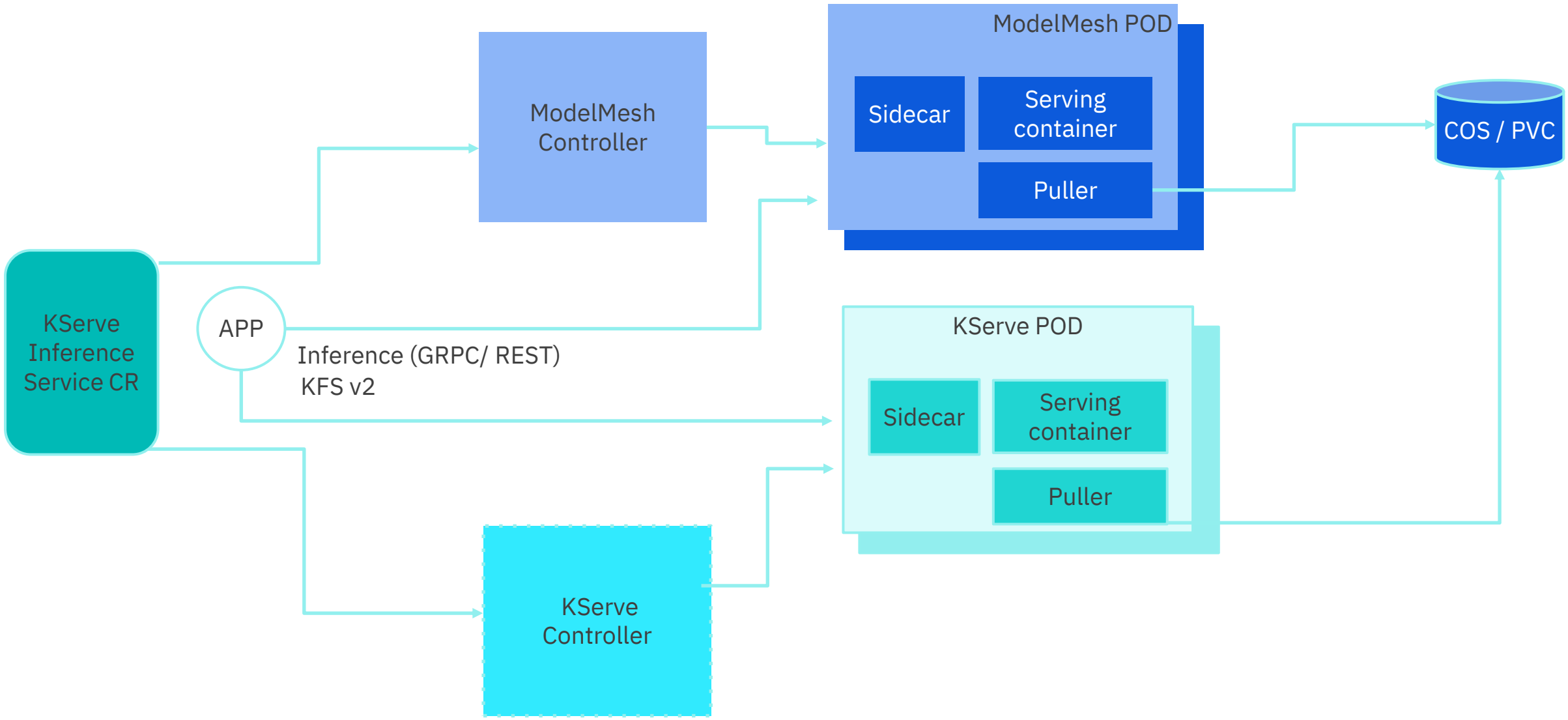
ModelMesh is being contributed to Open Source, and joining KServe!

ModelMesh
from
IBM Watson®

KServe



KServe v0.7 delivered with ModelMesh!



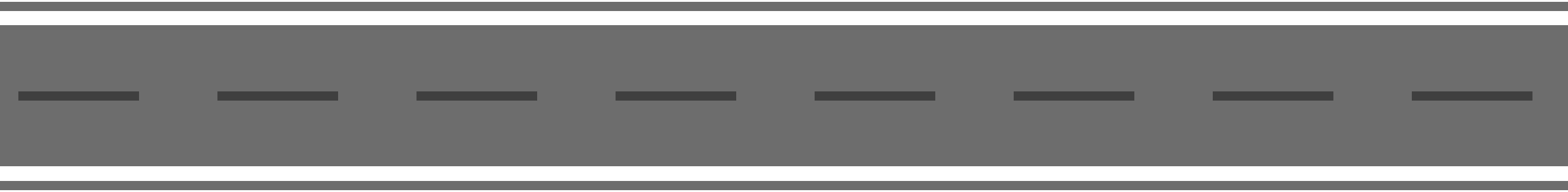
Preliminary Road Map

Q4 2021

1. KServe InferenceService Serving Runtime integration.
2. ModelMesh controller multi-namespace support.
3. Increased storage support.

Q1 2022

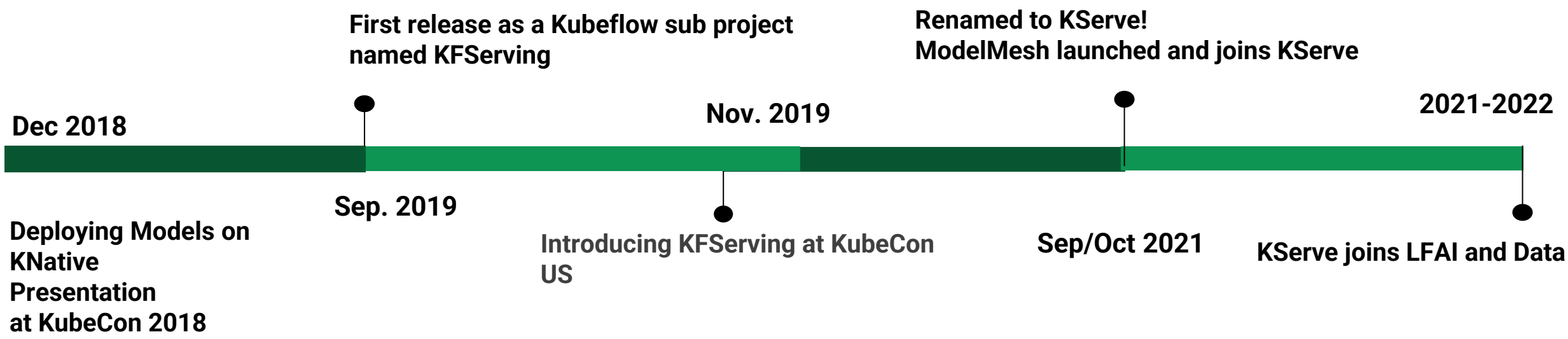
1. Inference Graph
2. Transformer
3. Canary Rollout
4. Consolidate MM controller with KServe controller.



ModelMesh


General / Modelmesh monitoring dashboard
Last 5 minutes
5s
81.8
63.5
modelmesh-serving-triton-2.x...
modelmesh-serving-triton-2.x...
5 ms
4 ms
4 ms
4 ms
4 ms
4 ms
22:14:00
22:15:00
22:16:00
22:17:00
22:18:00
modelmesh-serving-triton-2.x-54b747f6b7-qqlwz
modelmesh-serving-triton-2.x-54b747f6b7-5245b
Number of processed request per second
350
300
250
200
150
100
50
0
All ModelMesh Model Status
Managed
18030
Loaded
15746
Serving Runtime Cache Usage
2.50 GB
2 GB
1.50 GB
1 GB
500 MB
0 kB
Value
Total Cache Capacity
Cache Miss Rate
Number of missed cache models
100
80
60
40
20
0
Models on Serving Runtime
modelmesh-serving-
modelmesh-serving-
Serving Runtime Model Loading Rate
1.25
1
0.750
Model Unloading Rate

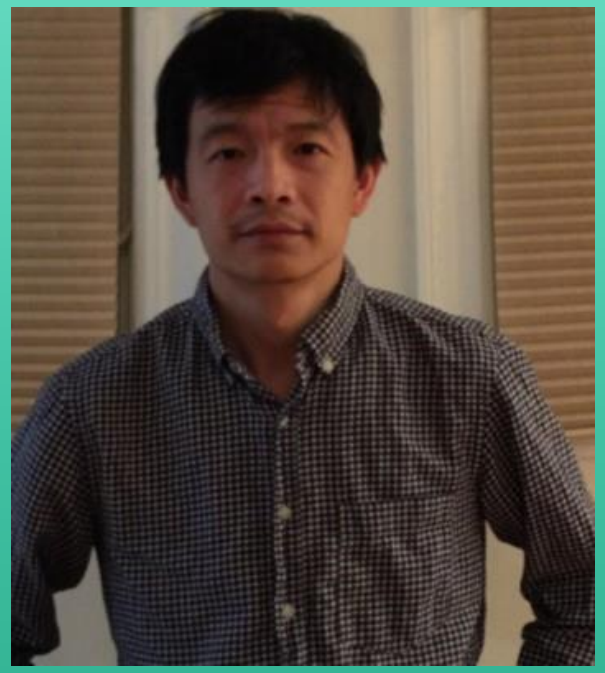
Moving Forward: Host KServe in LFAI and Data




Hey everyone, this one built the thing because of "problems he faced productionalizing ML at ..."
https://clear.ml
See? Nobody cares!

Contact Us

Animesh Singh
IBM
@AnimeshSingh



Dan Sun
Bloomberg

- KServe Website: https://kserve.github.io/website/
- KServe Github: https://github.com/kserve/kserve
- ModelMesh: https://github.com/kserve/modelmesh-serving
- https://github.com/kserve/kserve/blob/master/CONTRIBUTING.md#get-involved

Q&A - KServe Incubation in LF AI & Data

LF AI & DATA

TAC Vote on KServe Project Incubation Proposal

Proposed Resolution:

The TAC approves the KServe project as an Incubation project of the LF AI & Data Foundation

Next Steps

LF AI & Data staff will work with the KServe community to:

1. Onboard the project leading to the announcement of the project joining LF AI & Data
2. Explore potential integrations between the project and other hoted projects
3. Integrate the project with LF AI & Data operations

Upcoming TAC Meetings

LF AI & DATA

Upcoming TAC Meetings (Tentative)

- December 2, 2021: Flyte graduation, Soajs, Delta
- December 16, 2021: Janusgraph, DataPractices.org
- December 30, 2021: Canceled for the holiday

Please send agenda topic requests to tac-general@lists.lfaidata.foundation

Open Discussion

LF AI & DATA

TAC Meeting Details

- To subscribe to the TAC Group Calendar, visit the wiki: https://wiki.lfaidata.foundation/x/cQB2
- Join from PC, Mac, Linux, iOS or Android: https://zoom.us/j/430697670
- Or iPhone one-tap:
  - US: +16465588656,,430697670# or +16699006833,,430697670#
- Or Telephone:
  - Dial(for higher quality, dial a number based on your current location):
  - US: +1 646 558 8656 or +1 669 900 6833 or +1 855 880 1246 (Toll Free) or +1 877 369 0926 (Toll Free)
- Meeting ID: 430 697 670
- International numbers available: https://zoom.us/u/achYtcw7uN

Legal Notice

- The Linux Foundation, The Linux Foundation logos, and other marks that may be used herein are owned by The Linux Foundation or its affiliated entities, and are subject to The Linux Foundation's Trademark Usage Policy at https://www.linuxfoundation.org/trademark-usage, as may be modified from time to time.
- Linux is a registered trademark of Linus Torvalds. Please see the Linux Mark Institute's trademark usage page at https://lmi.linuxfoundation.org for details regarding use of this trademark.
- Some marks that may be used herein are owned by projects operating as separately incorporated entities managed by The Linux Foundation, and have their own trademarks, policies and usage guidelines.
- TWITTER, TWEET, RETWEET and the Twitter logo are trademarks of Twitter, Inc. or its affiliates.
- Facebook and the "f" logo are trademarks of Facebook or its affiliates.
- LinkedIn, the LinkedIn logo, the IN logo and InMail are registered trademarks or trademarks of LinkedIn Corporation and its affiliates in the United States and/or other countries.
- YouTube and the YouTube icon are trademarks of YouTube or its affiliates.
- All other trademarks are the property of their respective owners. Use of such marks herein does not represent affiliation with or authorization, sponsorship or approval by such owners unless otherwise expressly specified.
- The Linux Foundation is subject to other policies, including without limitation its Privacy Policy at https://www.linuxfoundation.org/privacy and its Antitrust Policy at https://www.linuxfoundation.org/antitrust-policy. each as may be modified from time to time. More information about The Linux Foundation's policies is available at https://www.linuxfoundation.org.
- Please email legal@linuxfoundation.org with any questions about The Linux Foundation's policies or the notices set forth on this slide.

LF AI & DATA