KubeCon | CloudNativeCon

Europe 2024

Production-Ready AI Platform on Kubernetes

Yuan Tang @TerryTangYuan

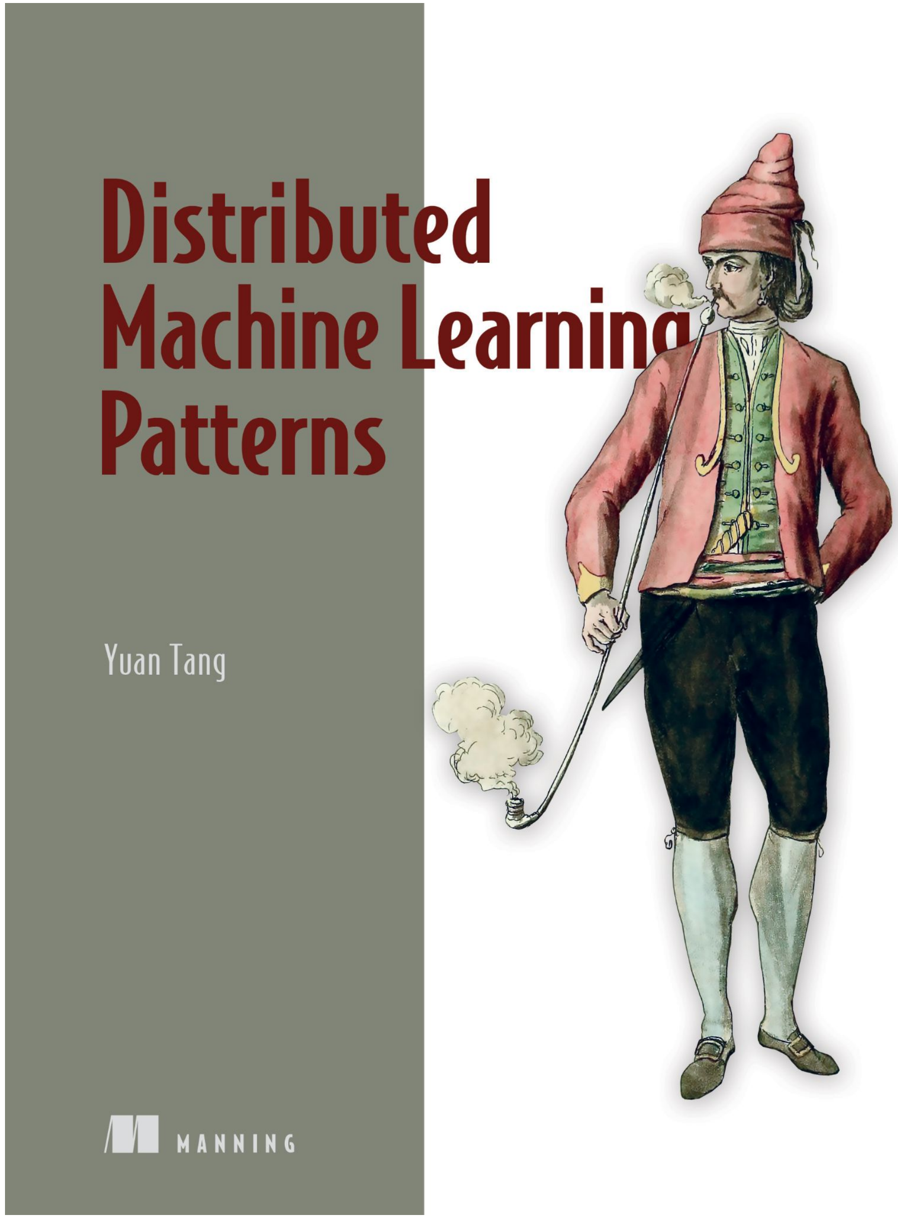
Principal Software Engineer, Red Hat OpenShift AI
Project Lead, Argo & Kubeflow

Agenda

- AI Landscape & Ecosystem
- Elements of Production Readiness
  - Scalability
  - Reliability
  - Observability
  - Flexibility
- Cloud Native Production-ready AI Platform
  - Data Processing
  - Model Training
  - Model Tuning
  - Model Serving
  - Workflow

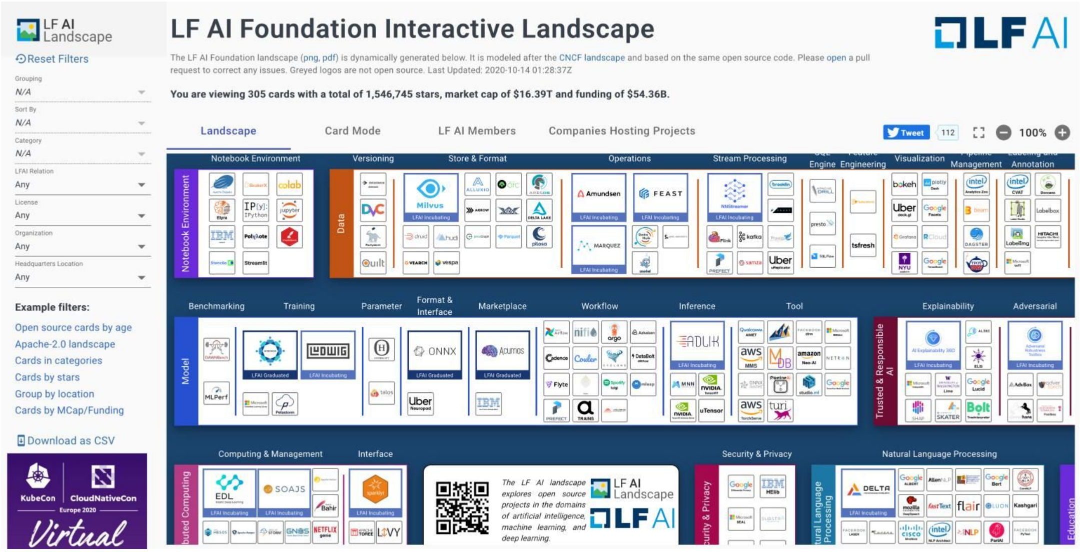
Distributed Machine Learning Patterns

KubeCon | CloudNativeCon Europe 2024



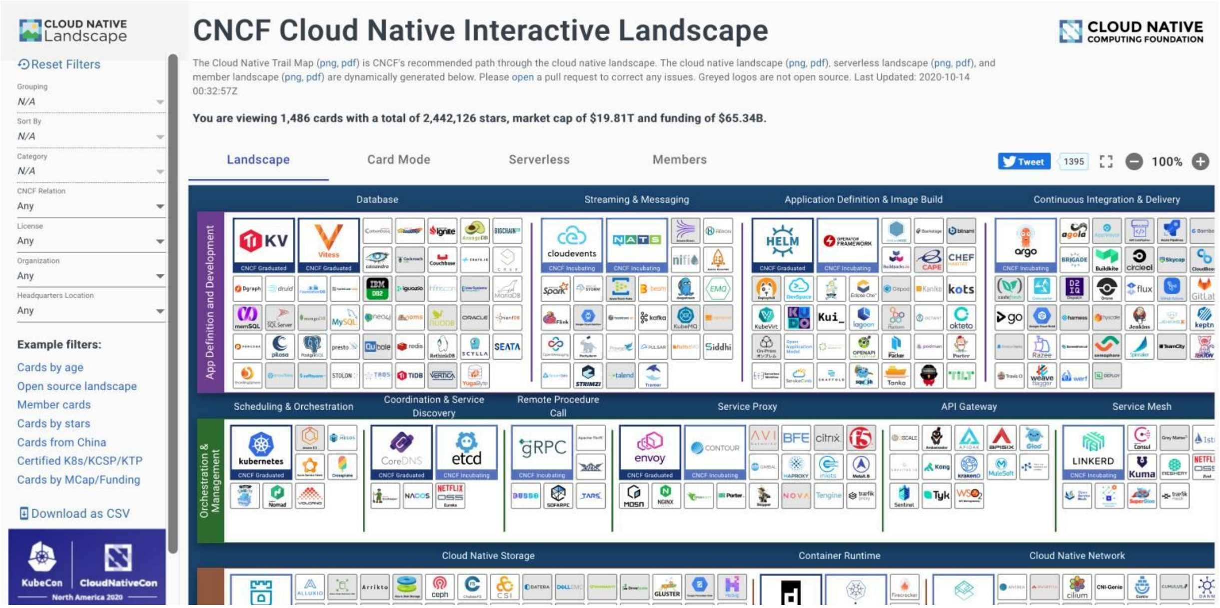
http://mng.bz/QZgv

AI Landscape & Ecosystem



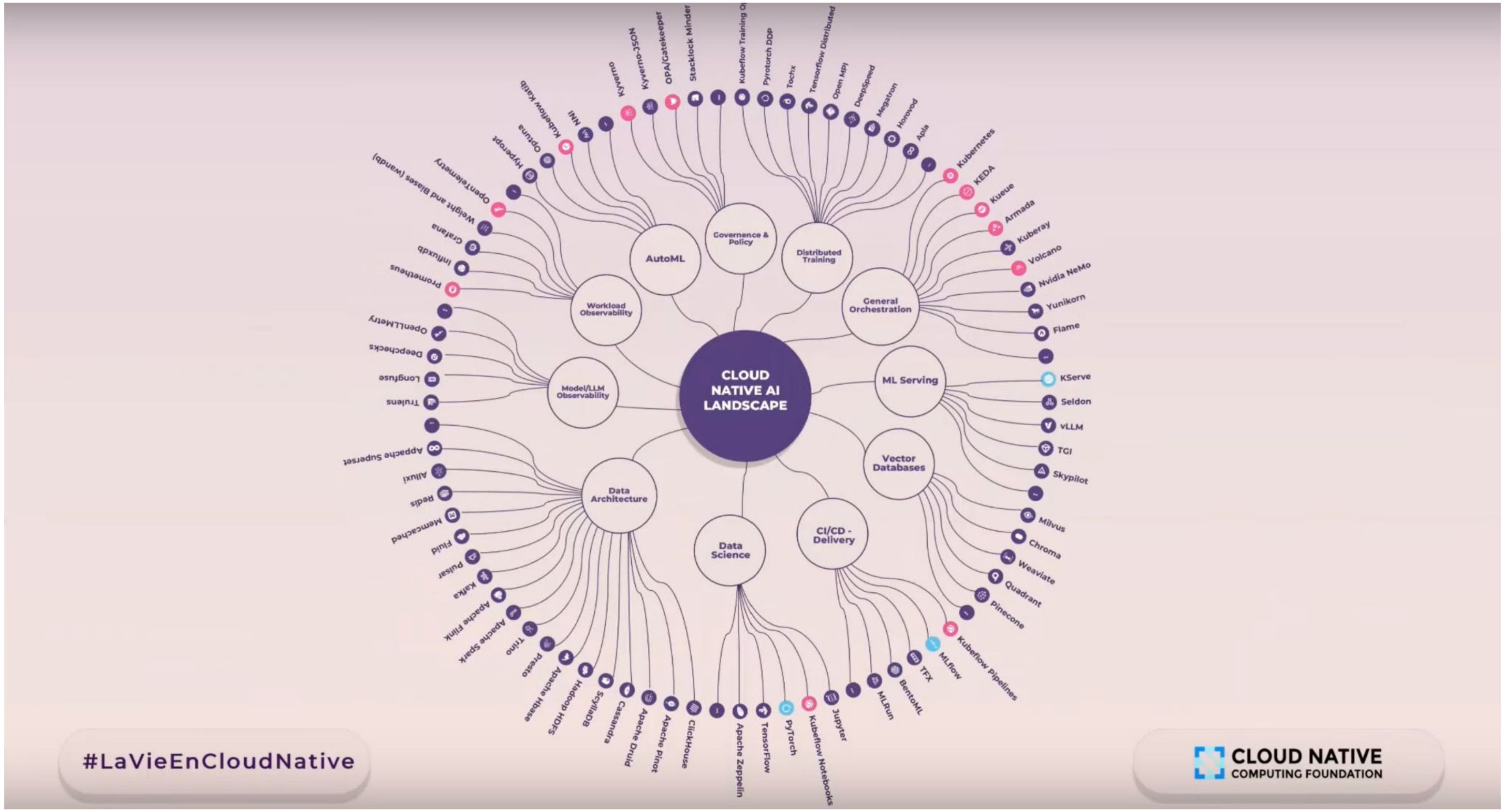
[LF AI & Data Landscape]

[CNCF Cloud Native Landscape]



AI Landscape & Ecosystem

KubeCon CloudNativeCon Europe 2024



[Opening Remarks by Priyanka Sharma at KubeCon EU 2024]

AI Landscape & Ecosystem

Kubeflow

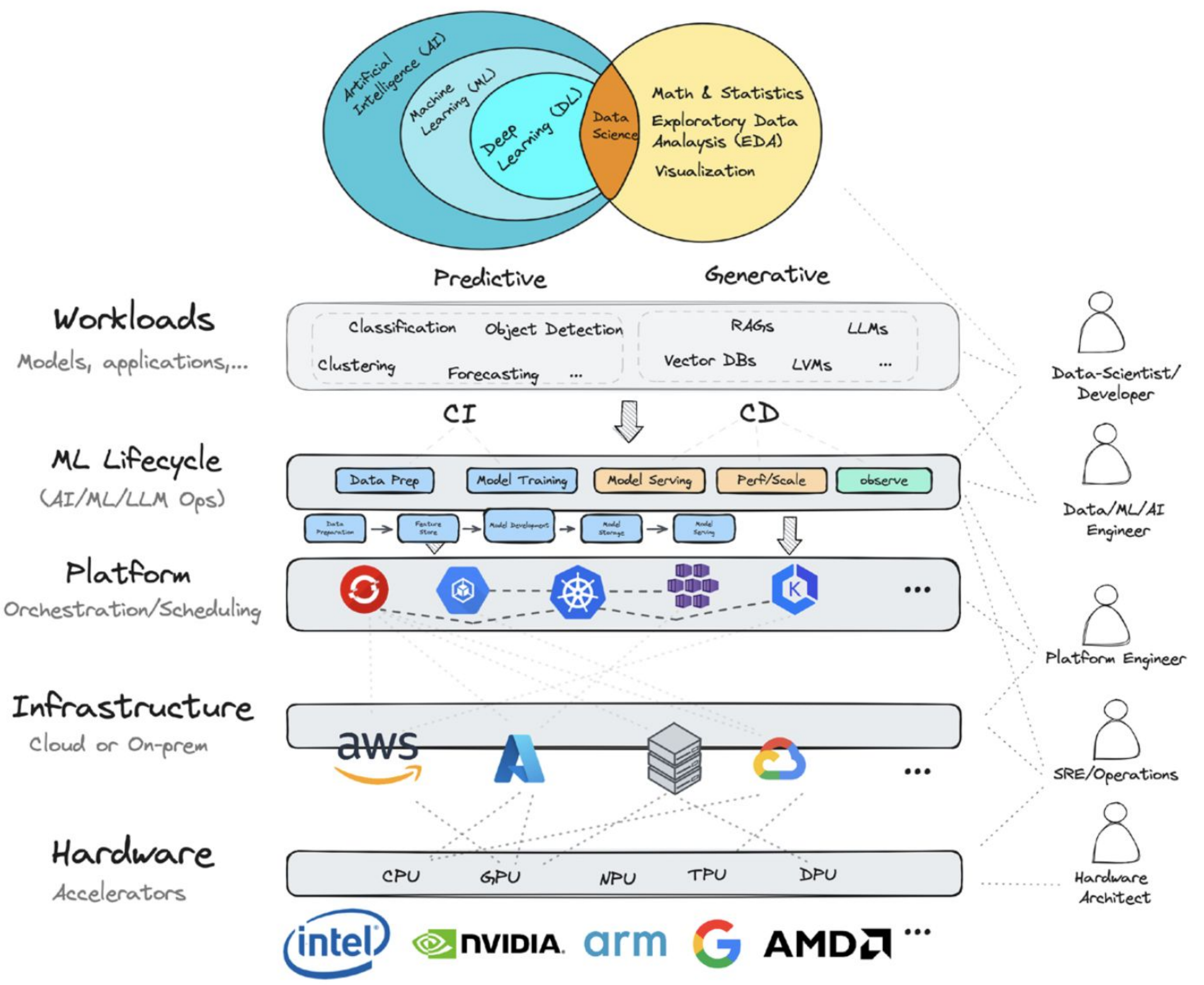
TensorFlow

argo

KServe

PyTorch

AI Landscape & Ecosystem

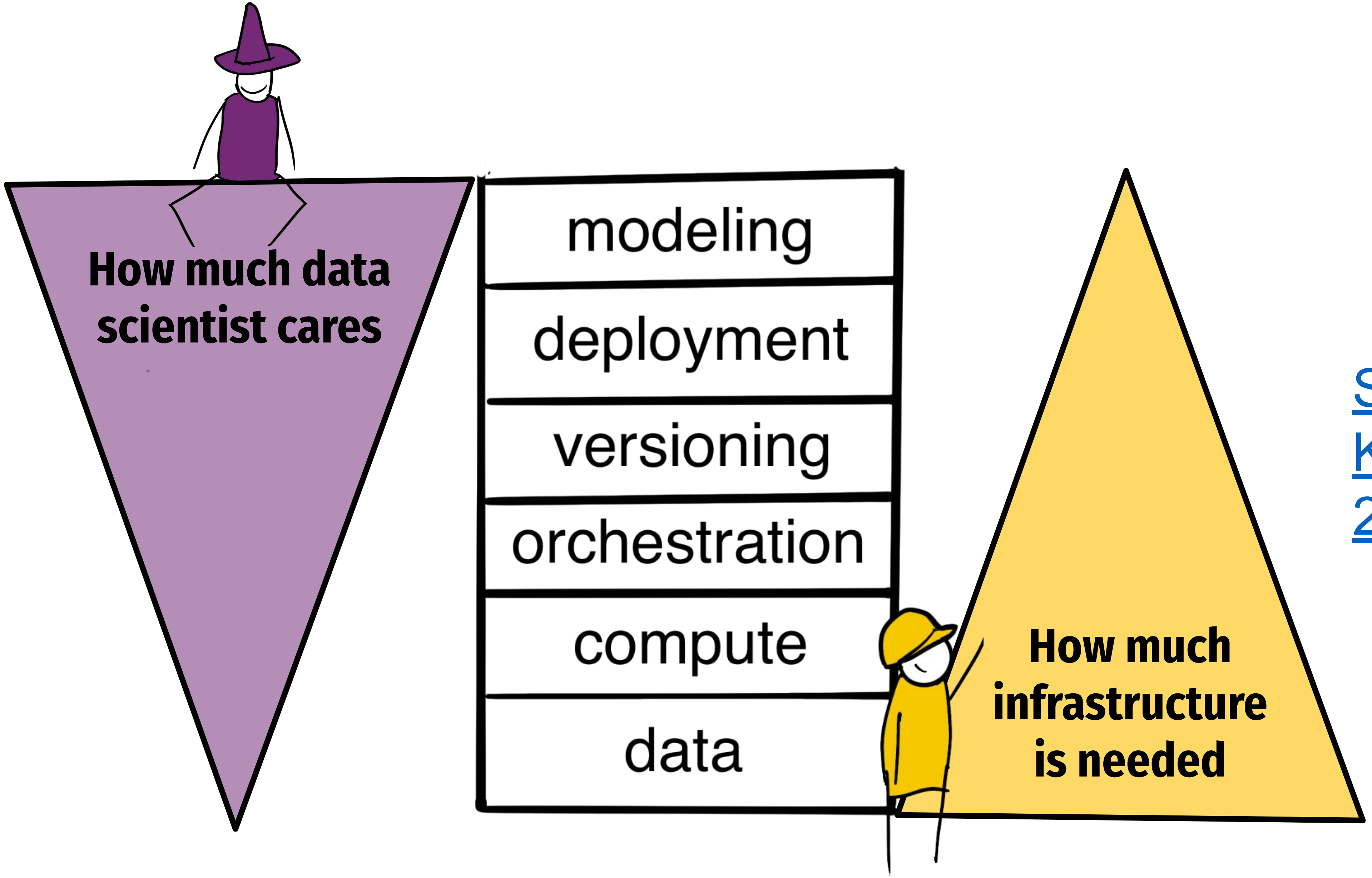


By Cloud Native AI WG

AI Landscape & Ecosystem

Cloud Native AI WG

AI Landscape & Ecosystem



[Savin & Yuan, KubeCon NA 2023]

Production Readiness - Scalability

- Horizontal scaling - more pods
  - K8s horizontal pod autoscaler
  - Knative pod autoscaler: event-driven
- Vertical scaling - more resources for existing pods
  - K8s vertical pod autoscaler
  - Resizer: adjust resources based on cluster nodes
- Cluster autoscaler - automatically adjusts the size of a Kubernetes Cluster
- Algorithm scalability
- Hardware acceleration and resource sharing
- Batch scheduling

Production Readiness - Reliability

- High availability and disaster recovery
    - K8s controller: leader election
- Elasticity and fault-tolerant
- Versioning: GitOps
- Vendor lock-in/hybrid cloud
- Support/SLAs

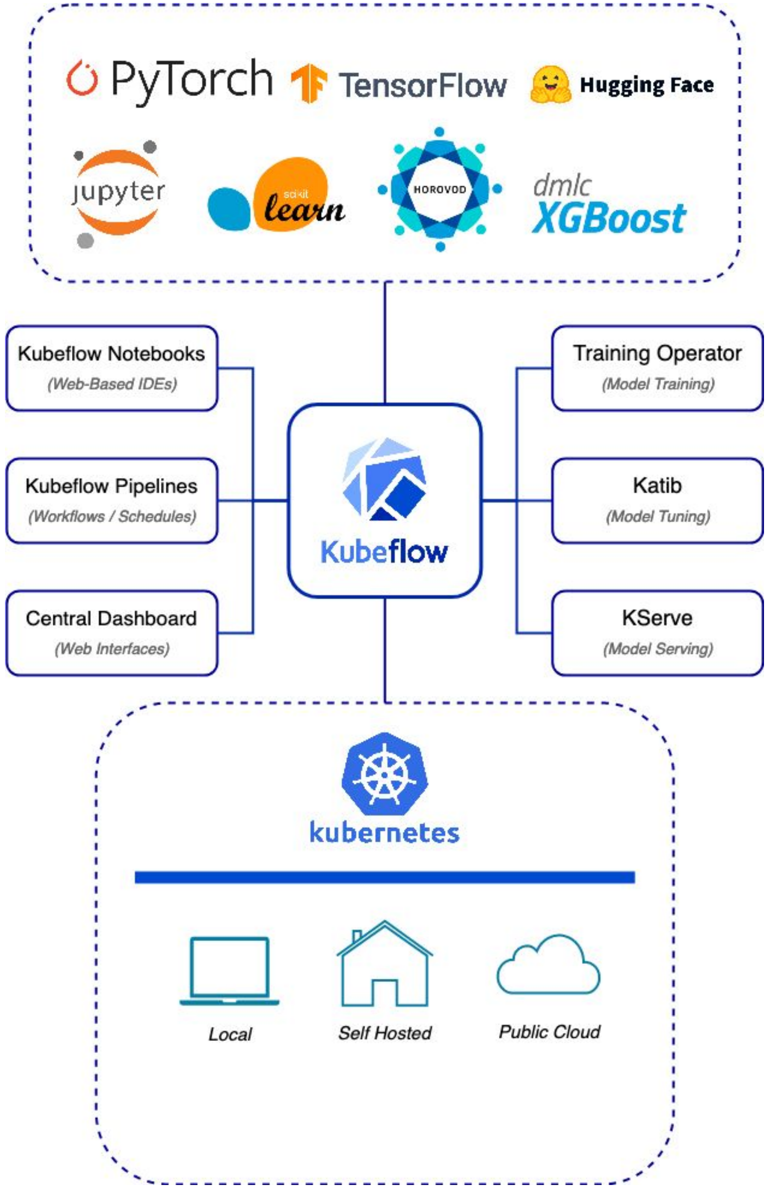
Production Readiness - Observability

- Performance metrics
  - Statistical (HP tuning, experiment tracking)
  - Operational (system, resources)
- Explainability & visualization
- Pipeline tracing
- Audit log

Production Readiness - Flexibility

- Various ML frameworks
- Language-specific SDKs
- Standardized APIs
- Data: size, streaming/batching
- Model: size, framework, performance
- Integration with various hardware accelerators
- Cloud/on-prem/edge
- Vendor lock-in

Kubeflow: The ML Toolkit for Kubernetes



https://www.kubeflow.org/

Cloud Native Production-ready AI Platform
1. Data Processing

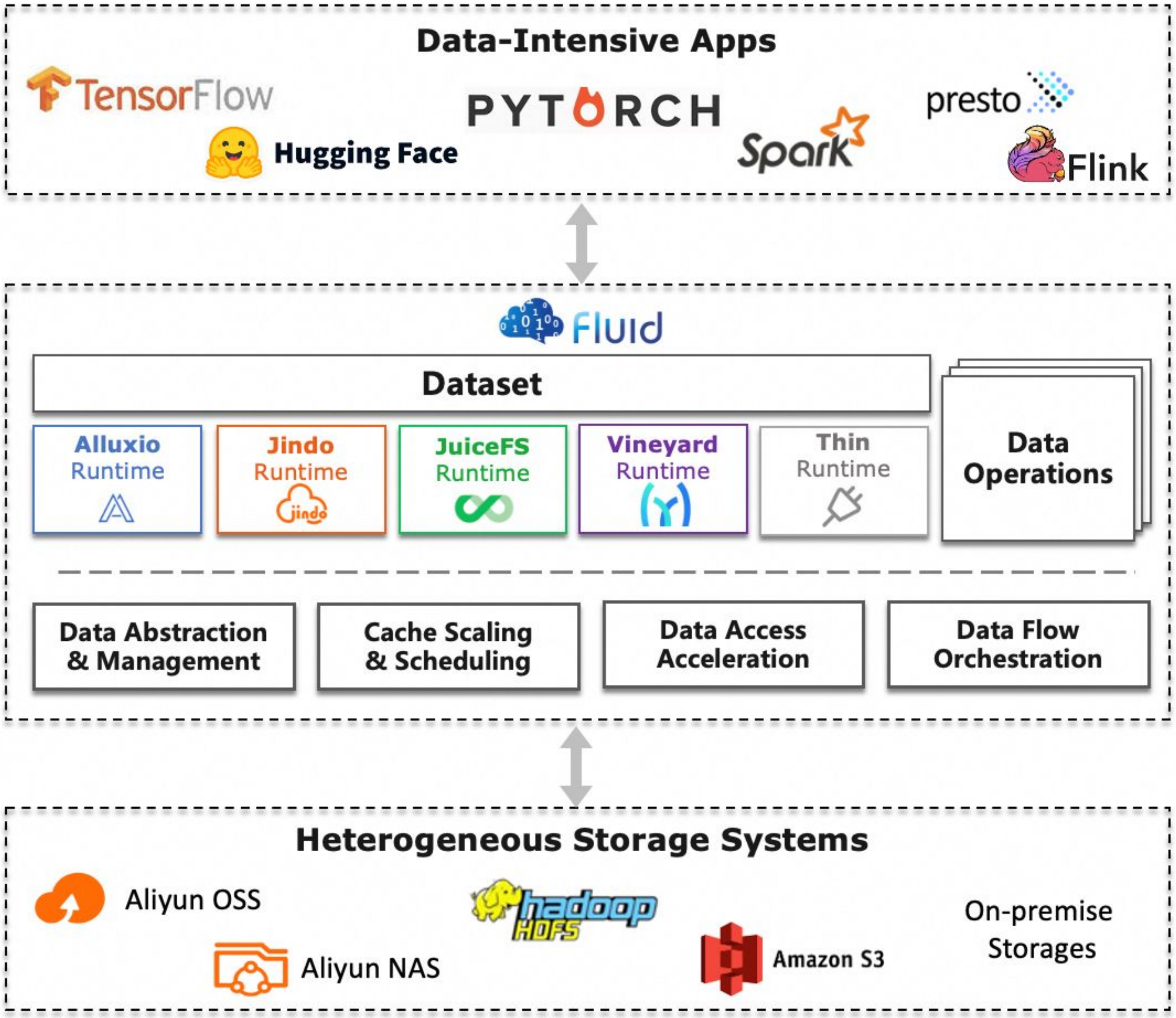
KubeCon CloudNativeCon Europe 2024

- Big data - Apache Spark
  - Batch & Streaming
  - Time series
- Welcome kubeflow/spark-operator to Kubeflow project!

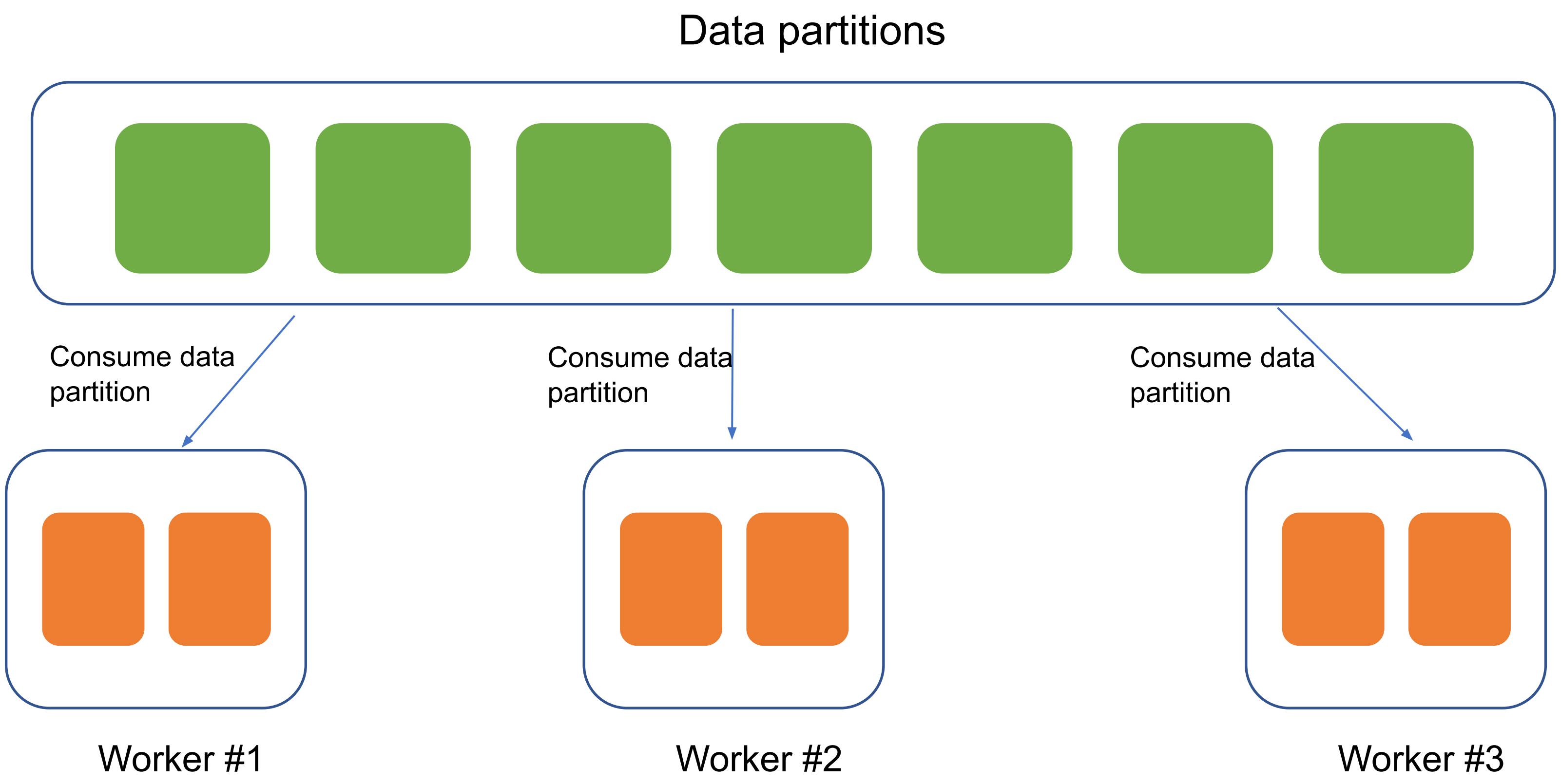

```yaml
apiVersion: "sparkoperator.k8s.io/v1beta2"
kind: SparkApplication
metadata:
  name: pyspark-pi
spec:
  type: Python
  pythonVersion: "3"
  mode: cluster
  image: "gcr.io/spark-operator/spark-py:v3.1.1"
  imagePullPolicy: Always
  mainApplicationFile: local:///opt/spark/examples/src/main/python/pi.py
  sparkVersion: "3.1.1"
  restartPolicy:
    type: OnFailure
    onFailureRetries: 3
    onFailureRetryInterval: 10
  driver:
    cores: 1
    coreLimit: "1200m"
    memory: "512m"
  executor:
    cores: 1
    instances: 1
    memory: "512m"
```

Cloud Native Production-ready AI Platform
1. Data Processing

- **Fluid (fluid-cloudnative/fluid)**
  - Enable dataset warmup and acceleration for data-intensive applications by using distributed cache in Kubernetes
  - Dataset abstractions for heterogeneous data source management
  - Data-aware scheduling



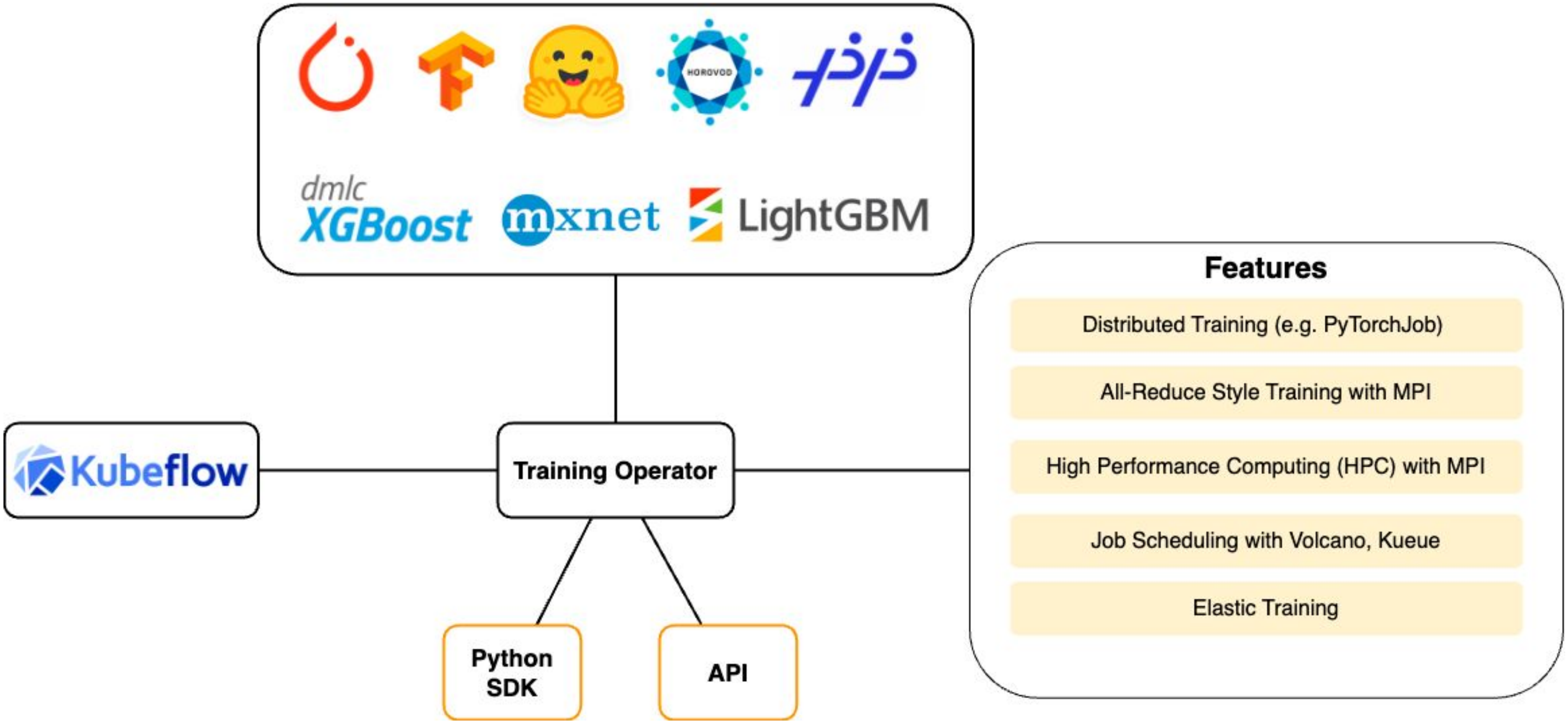
Cloud Native Production-ready AI Platform
2. Model Training



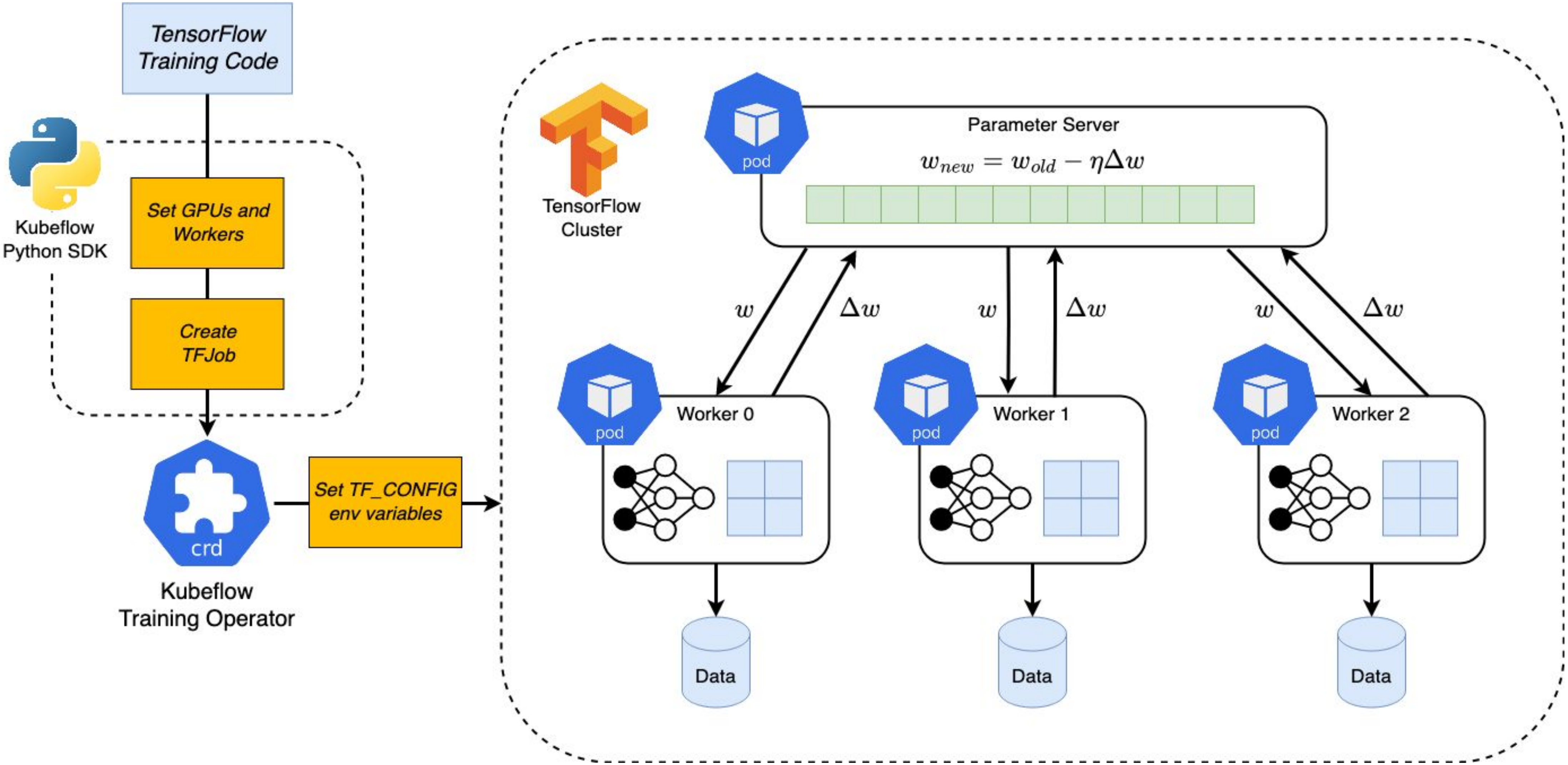
Distributed all-reduce model training with multiple workers and data partitions

Source: [Distributed Machine Learning Patterns book]

Kubeflow Training Operator Architecture



Distributed training with TensorFlow



Distributed large model fine-tuning

Details here by Andrey Velichkevich

```
TrainingClient().train(
    name=job_name_train_api,
    num_workers=1,
    num_procs_per_worker=1,
    model_provider_parameters=HuggingFaceModelParams(
        model_uri="hf://google-bert/bert-base-cased",
        transformer_type=transformers.AutoModelForSequenceClassification,
    ),
    storage_config={
        "access_modes": ["ReadWriteOnce"]
    },
    dataset_provider_parameters=HfDatasetParams(
        repo_id="yelp_review_full",
        split="train[:3000]",
    ),
    train_parameters=HuggingFaceTrainParams(
        training_parameters=transformers.TrainingArguments(
            output_dir="test_trainer",
            save_strategy="no",
            evaluation_strategy="no",
            do_eval=False,
            disable_tqdm=True,
            log_level="info",
        ),
        lora_config=LoraConfig(
            r=8,
            lora_alpha=8,
            lora_dropout=0.1,
            bias="none",
        ),
    ),
    resources_per_worker={
        "gpu": 1,
        "cpu": 5,
        "memory": "10G",
    },
)
```

Cloud Native Production-ready AI Platform
3. Model Tuning

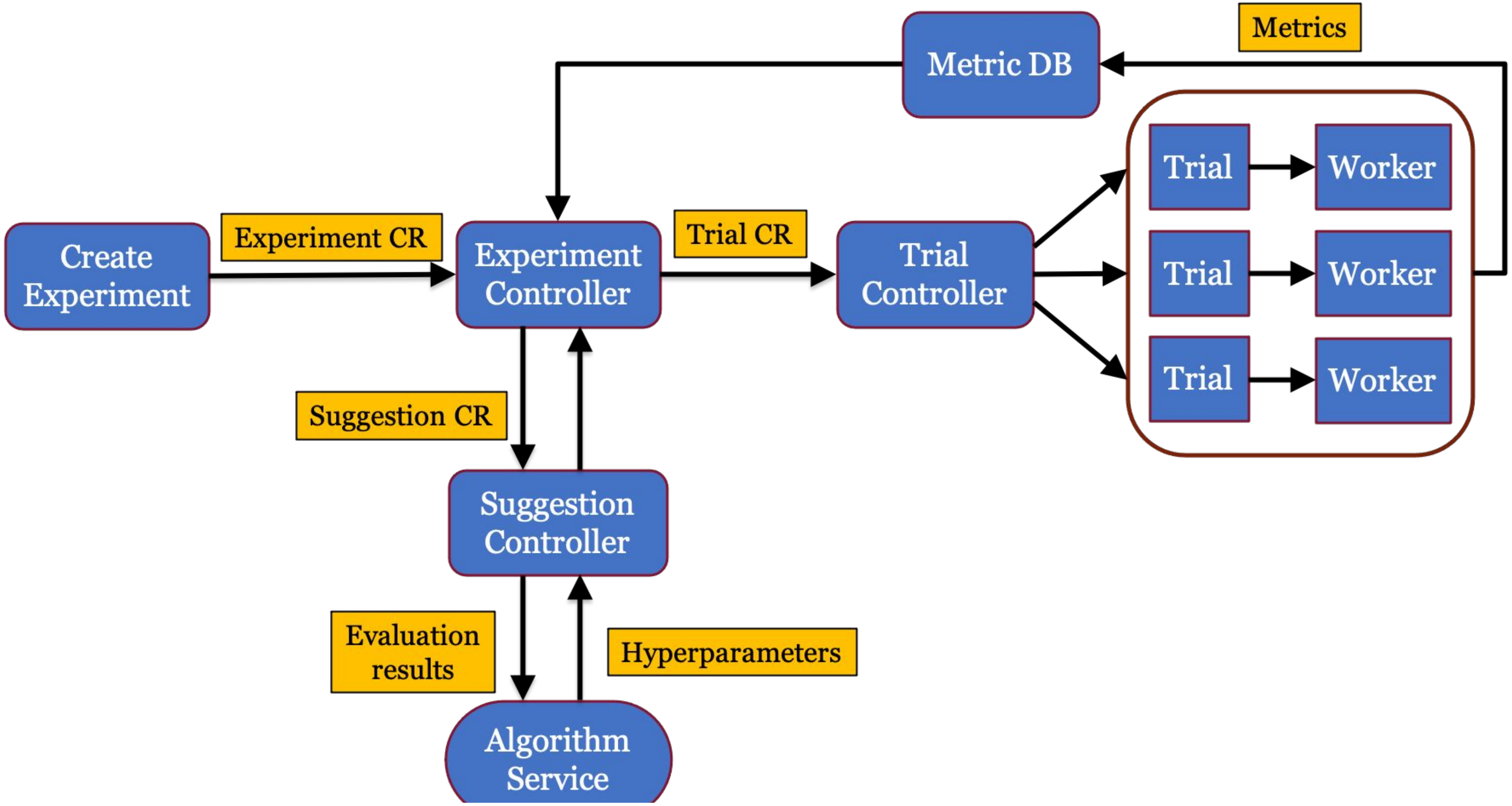
Katib: Kubernetes-native AutoML in Kubeflow

- Supports HP tuning, NAS and Early Stopping
- Agnostic to ML framework and programming languages
- Can be deployed on local machines or on private/public clouds
- Can orchestrate any Kubernetes workloads and custom resources
- Natively integrated with Kubeflow components (Notebooks, Pipelines, Training Operators)

Cloud Native Production-ready AI Platform
3. Model Tuning

Katib Architecture

Reference paper https://arxiv.org/abs/2006.02085



Example

Experiment Budget

Objective

Algorithm

Search Space

```yaml
parallelTrialCount: 2
maxTrialCount: 15
maxFailedTrialCount: 3
objective:
  type: maximize
  goal: 0.99
  objectiveMetricName: Validation-accuracy
algorithm:
  algorithmName: random
parameters:
  - name: lr
    parameterType: double
    feasibleSpace:
      min: "0.01"
      max: "0.05"
  - name: num-epochs
    parameterType: categorical
    feasibleSpace:
      list:
        - 5
        - 10
```

Trial Template

```yaml
trialTemplate:
  retain: true
  primaryContainerName: training-container
  trialParameters:
    - name: learningRate
      description: Learning rate for the training model
      reference: lr
    - name: numberEpochs
      description: Number of epochs to train the model
      reference: num-epochs
  trialSpec:
    apiVersion: batch/v1
    kind: Job
    spec:
      template:
        spec:
          containers:
            - name: training-container
              image: docker.io/kubeflowkatib/mxnet-mnist:v1beta1-45c5727
              command:
                - "python3"
                - "/opt/mxnet-mnist/mnist.py"
                - "--batch-size=64"
                - "--lr=${trialParameters.learningRate}"
                - "--num-epochs=${trialParameters.numberEpochs}"
          restartPolicy: Never
```

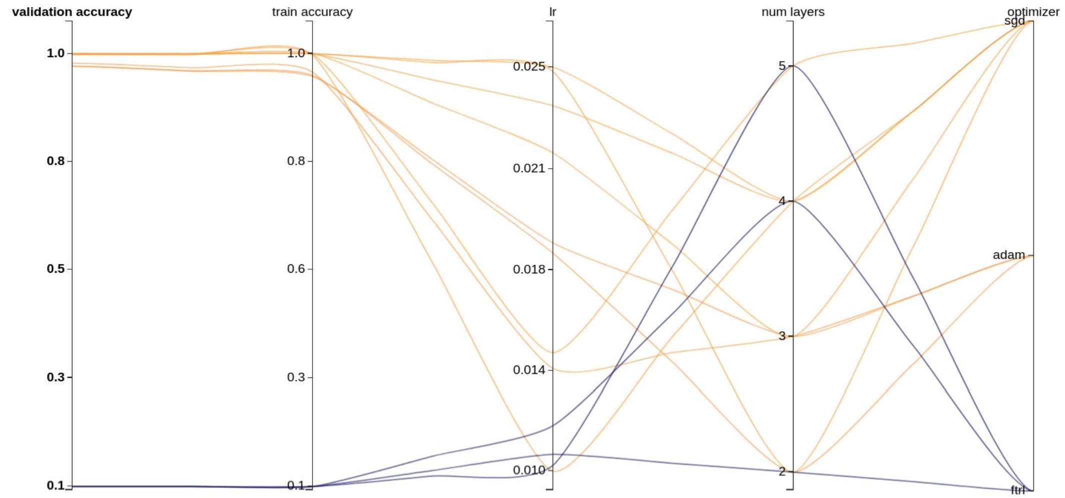
Cloud Native Production-ready AI Platform
3. Model Tuning

KubeCon | CloudNativeCon Europe 2024

Example

← Experiment details

DELETE



OVERVIEW | TRIALS | DETAILS | YAML

Name	bayesianoptimization-example
Status	Experiment has succeeded because max trial count has reached
Best trial	bayesianoptimization-example-6935d45b
Best trial's params	lr: 1.44659e-2 num-layers: 5 optimizer: sgd
Best trial performance	Validation-accuracy: 0.97939 Train-accuracy: 0.99287
User defined goal	Validation-accuracy > 0.99
Running trials	0
Failed trials	0
Succeeded trials	12

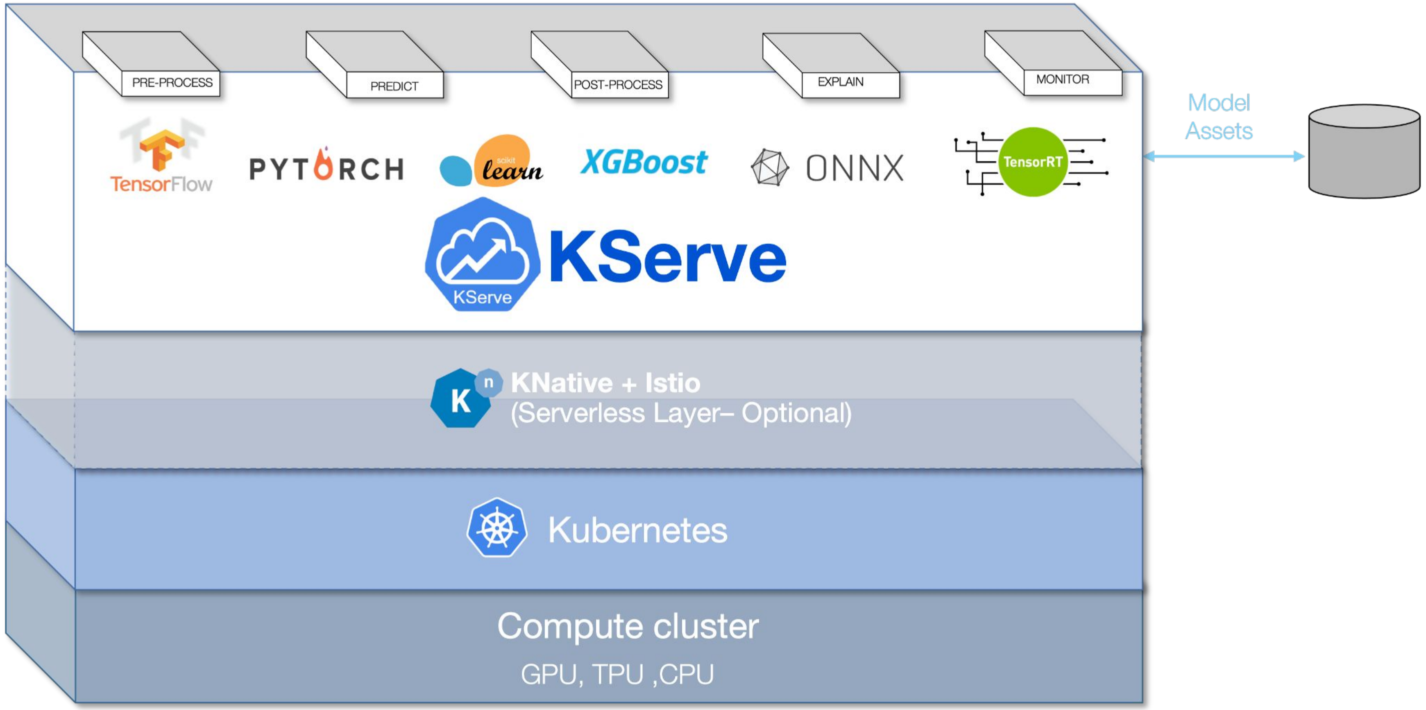
Experiment Conditions

Status	Type	Last Transition Time	Reason	Message
✅	Created	1 month ago	ExperimentCreated	Experiment is created
⚠️	Running	1 month ago	ExperimentRunning	Experiment is running
✅	Succeeded	1 month ago	ExperimentMaxTrialsReached	Experiment has succeeded because max trial count has reached

Cloud Native Production-ready AI Platform
4. Model Serving

KServe: Highly scalable, standard, cloud agnostic model inference platform on Kubernetes

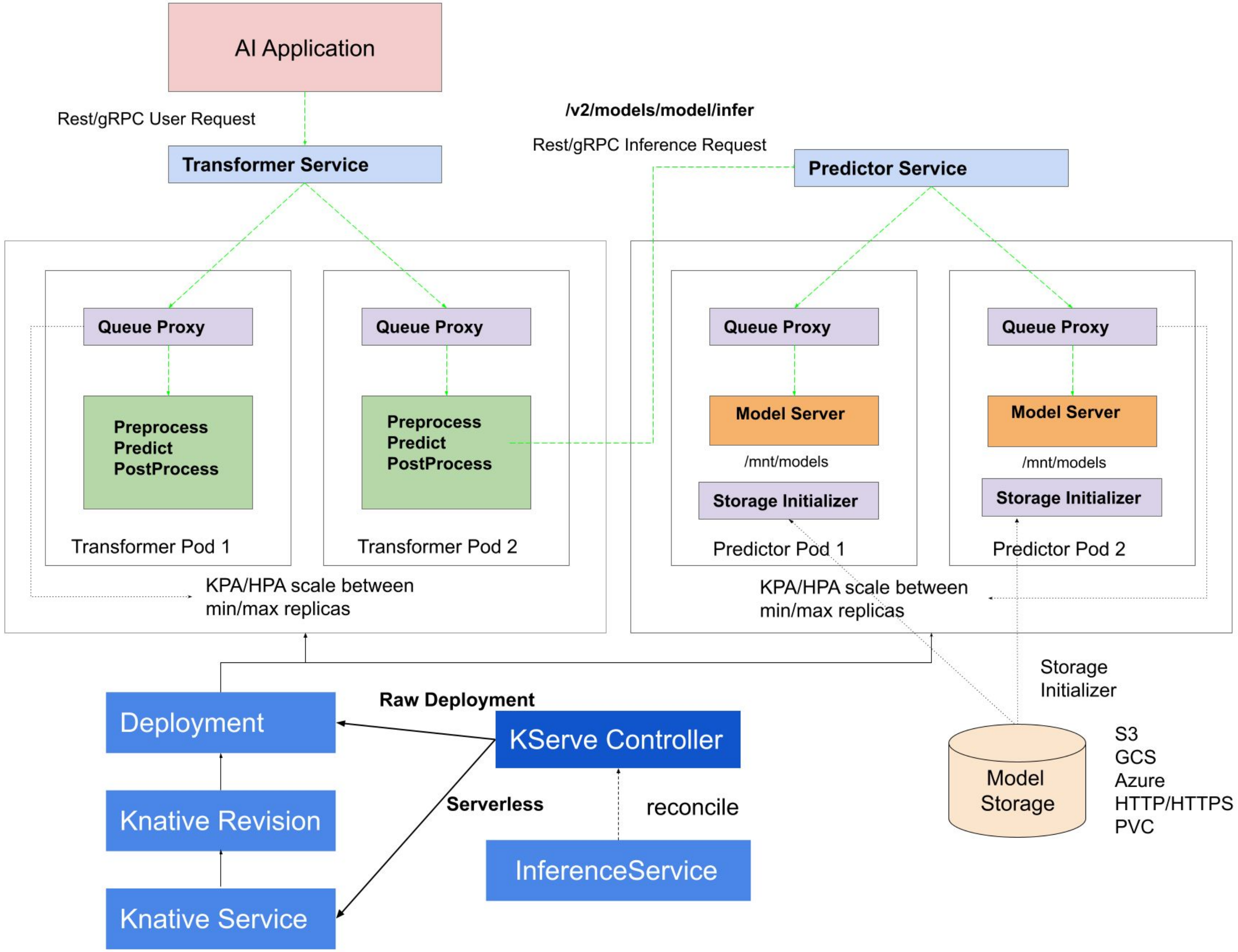
- Performant, standardized inference protocol across ML frameworks.
- Serverless inference workload with request based auto scaling including scale-to-zero on CPU and GPU.
- High scalability, density packing and intelligent routing using ModelMesh.
- Simple and pluggable production serving for inference, pre/post processing, monitoring and explainability.
- Advanced deployments for canary rollout, pipeline, ensembles with InferenceGraph.



Cloud Native Production-ready AI Platform
4. Model Serving

KubeCon | CloudNativeCon Europe 2024

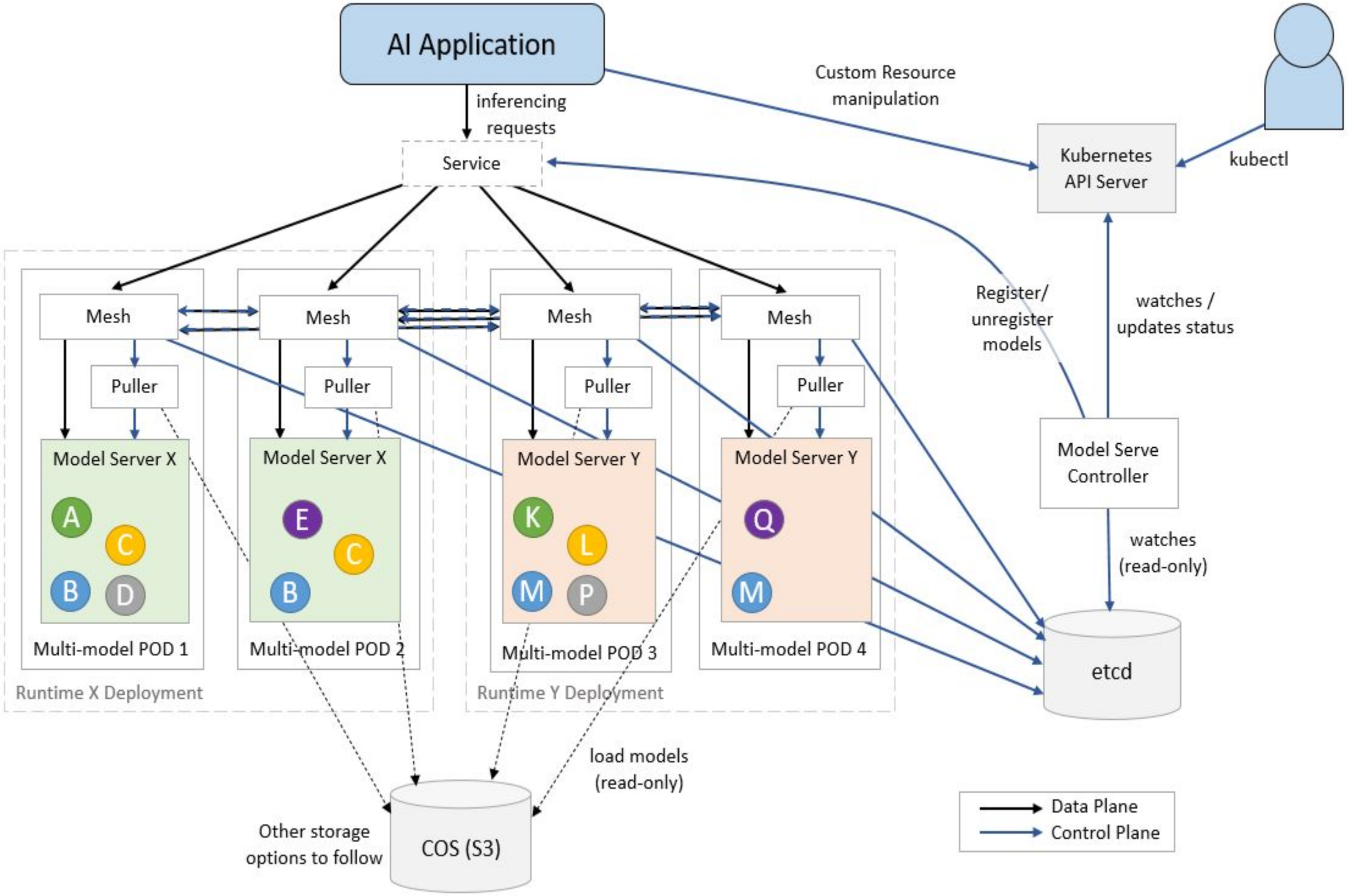
Single model serving



Cloud Native Production-ready AI Platform
4. Model Serving

Multi model serving: ModelMesh

- Designed for high-scale, high-density and frequently-changing model use cases.
- Intelligently loads and unloads models to and from memory to strike an intelligent trade-off between responsiveness to users and computational footprint.



LLMs

```
> curl -H "content-type:application/json" -H
"Host: ${SERVICE_HOSTNAME}" -v
http://${INGRESS_HOST}:${INGRESS_PORT}/v2/model
s/${MODEL_NAME}/infer -d '{"id": "42","inputs":
[{"name": "input0","shape": [-1],"datatype":
"BYTES","data": [""Where is Eiffel Tower?"]}]}'
```


{"text_output":"The Eiffel Tower is located in the 7th arrondissement of Paris, France. It stands on the Champ de Mars, a large public park next to the Seine River. The tower's exact address is:\n\n2 Rue du Champ de Mars, 75007 Paris, France.","model_name":"llama2"}


```yaml
apiVersion: serving.kserve.io/v1beta1
kind: InferenceService
metadata:
  name: huggingface-llama2
spec:
  predictor:
    model:
      modelFormat:
        name: huggingface
      args:
      - --model_name=llama2
      - --model_id=meta-llama/Llama-2-7b-chat-hf
      resources:
        limits:
          cpu: "6"
          memory: 24Gi
          nvidia.com/gpu: "1"
        requests:
          cpu: "6"
          memory: 24Gi
          nvidia.com/gpu: "1"
```

Problem: model initialization takes a long time

Solution: Modelcars feature (model is in OCI image) in KServe brings:

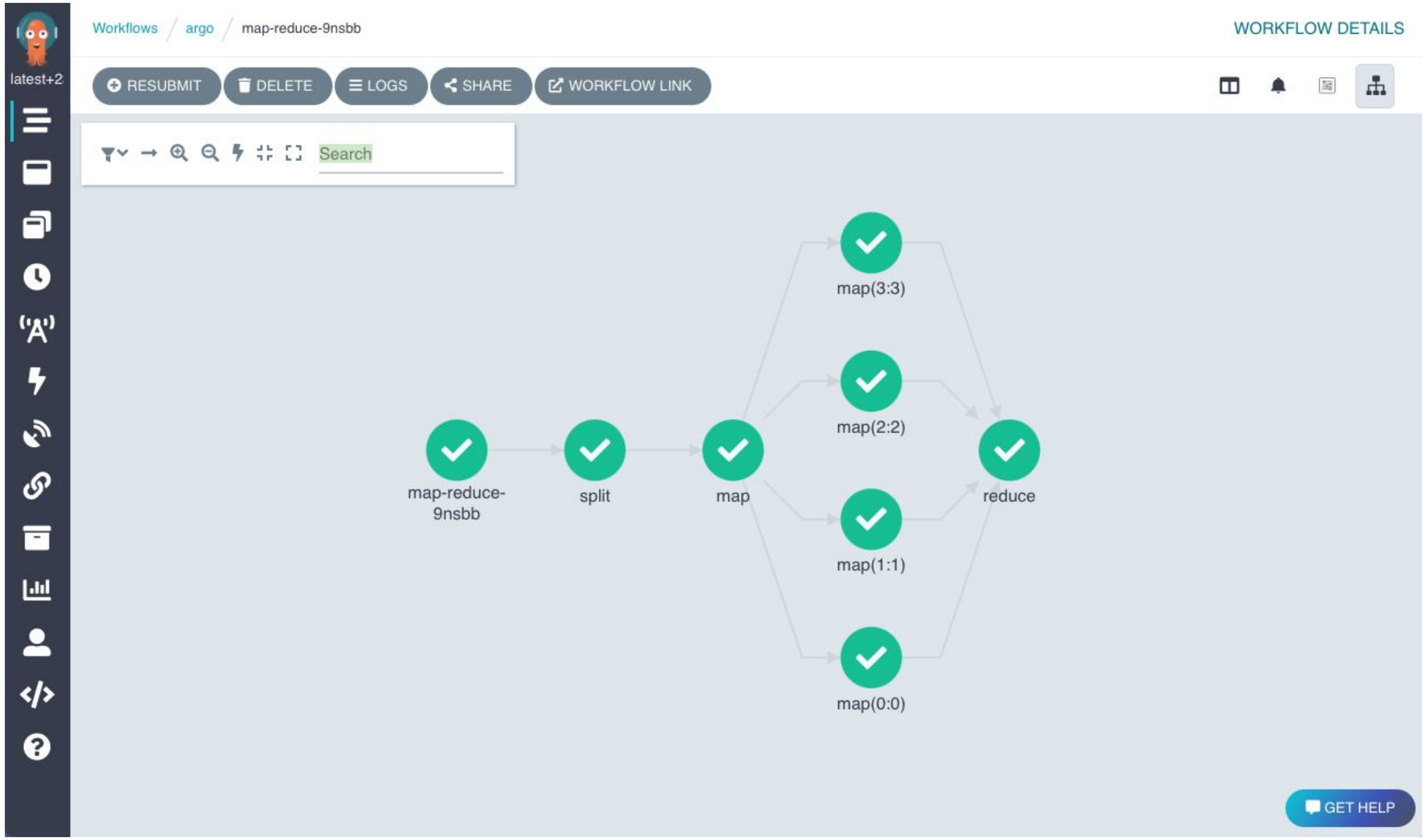
- Reduced Startup Times: By avoiding repetitive downloads of large models, startup delays are significantly minimized.
- Lower Disk Space Usage: The feature decreases the need for duplicated local storage, conserving disk space.
- Enhanced Performance: Modelcars allows for advanced techniques like pre-fetching images and lazy-loading, improving efficiency.

KubeCon CloudNativeCon Europe 2024

Argo Workflows

The container-native workflow engine for Kubernetes

- Machine learning pipelines
- Data processing/ETL
- Infrastructure automation
- Continuous delivery/integration



Cloud Native Production-ready AI Platform
5. Workflow

CRDs and Controllers

- Kubernetes custom resources that natively integrates with other K8s resources (volumes, secrets, etc.)

Interfaces

- CLI: manage workflows and perform operations (submit, suspend, delete/etc.)
- Server: REST & gRPC interfaces
- SDKs: Python, Go, and Java SDKs
- UI: manage and visualize workflows, artifacts, logs, resource usages analytics, etc.

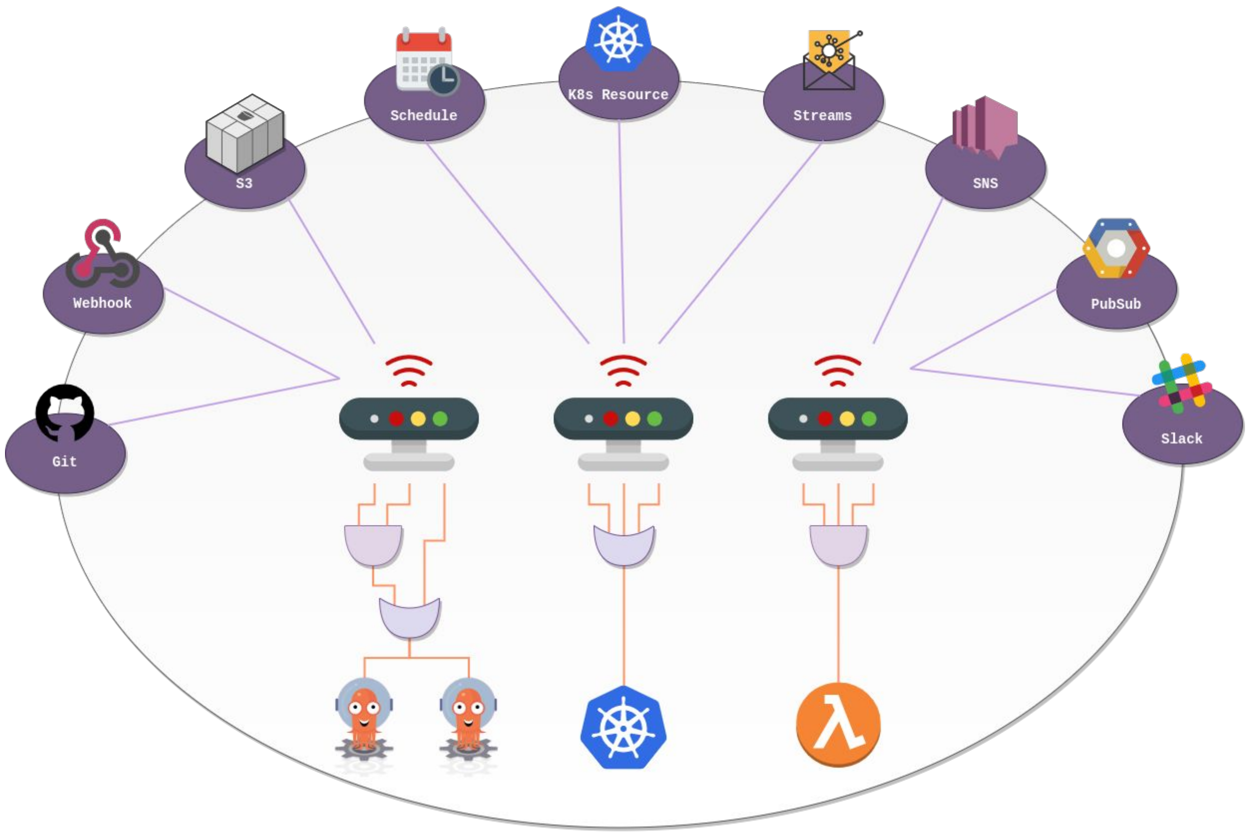
Example

```python
@script()
def echo(message: str):
    print(message)


with Workflow(
    generate_name="dag-diamond-",
    entrypoint="diamond",
) as w:
    with DAG(name="diamond"):
        A = echo(name="A", arguments={"message": "A"})
        B = echo(name="B", arguments={"message": "B"})
        C = echo(name="C", arguments={"message": "C"})
        D = echo(name="D", arguments={"message": "D"})
        A >> [B, C] >> D

w.create()
```

Argo Events

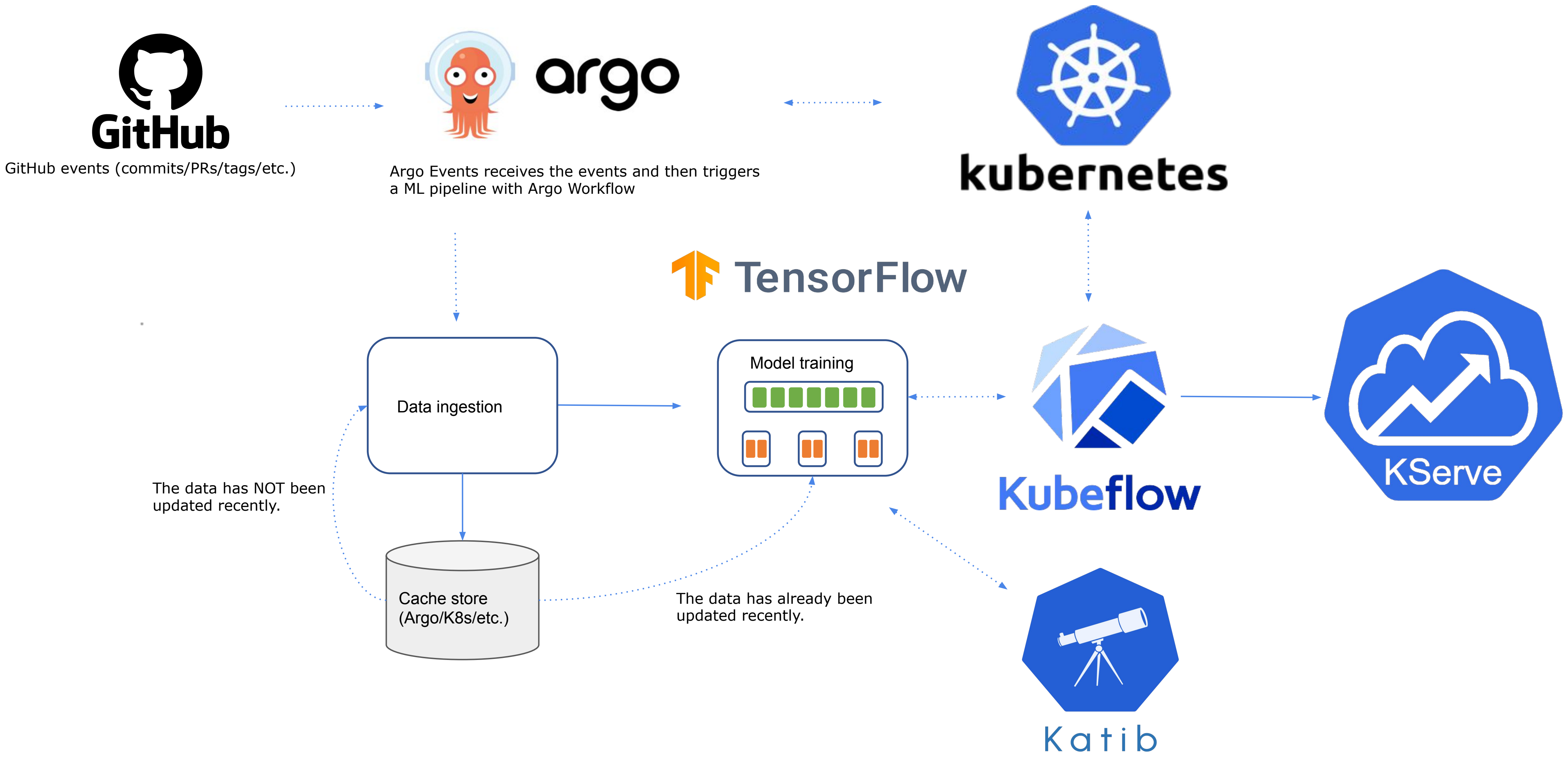
Event-driven workflow automation

- Supports events from 20+ event sources
  - Webhooks, S3, GCP PubSub, Git, Slack, etc.
- Supports 10+ triggers
  - Kubernetes Objects, Argo Workflow, AWS Lambda, Kafka, Slack, etc.
- Manage everything from simple, linear, real-time to complex, multi-source events
- CloudEvents specification compliant

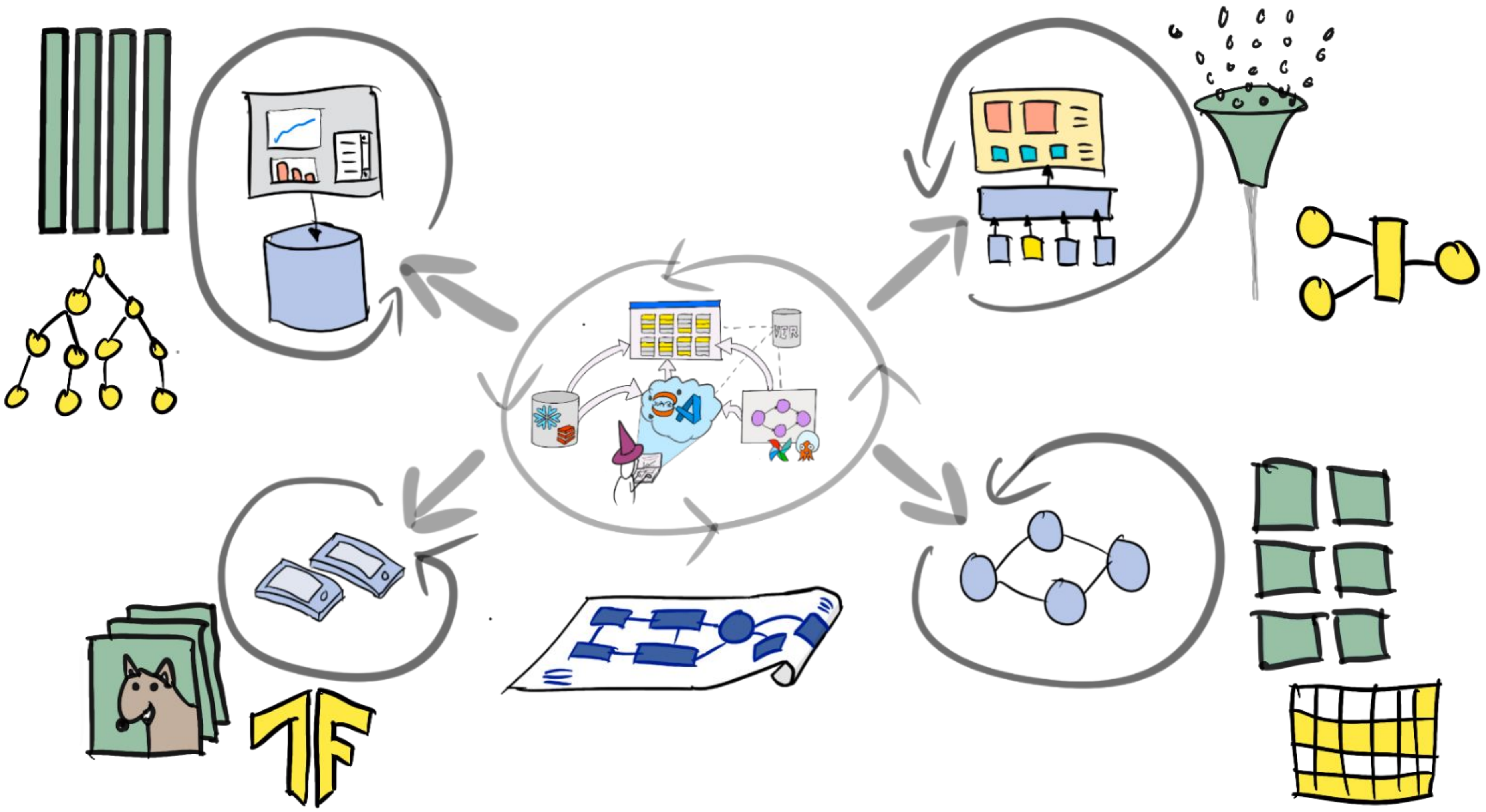


Cloud Native Production-ready AI Platform
5. Workflow

KubeCon CloudNativeCon Europe 2024



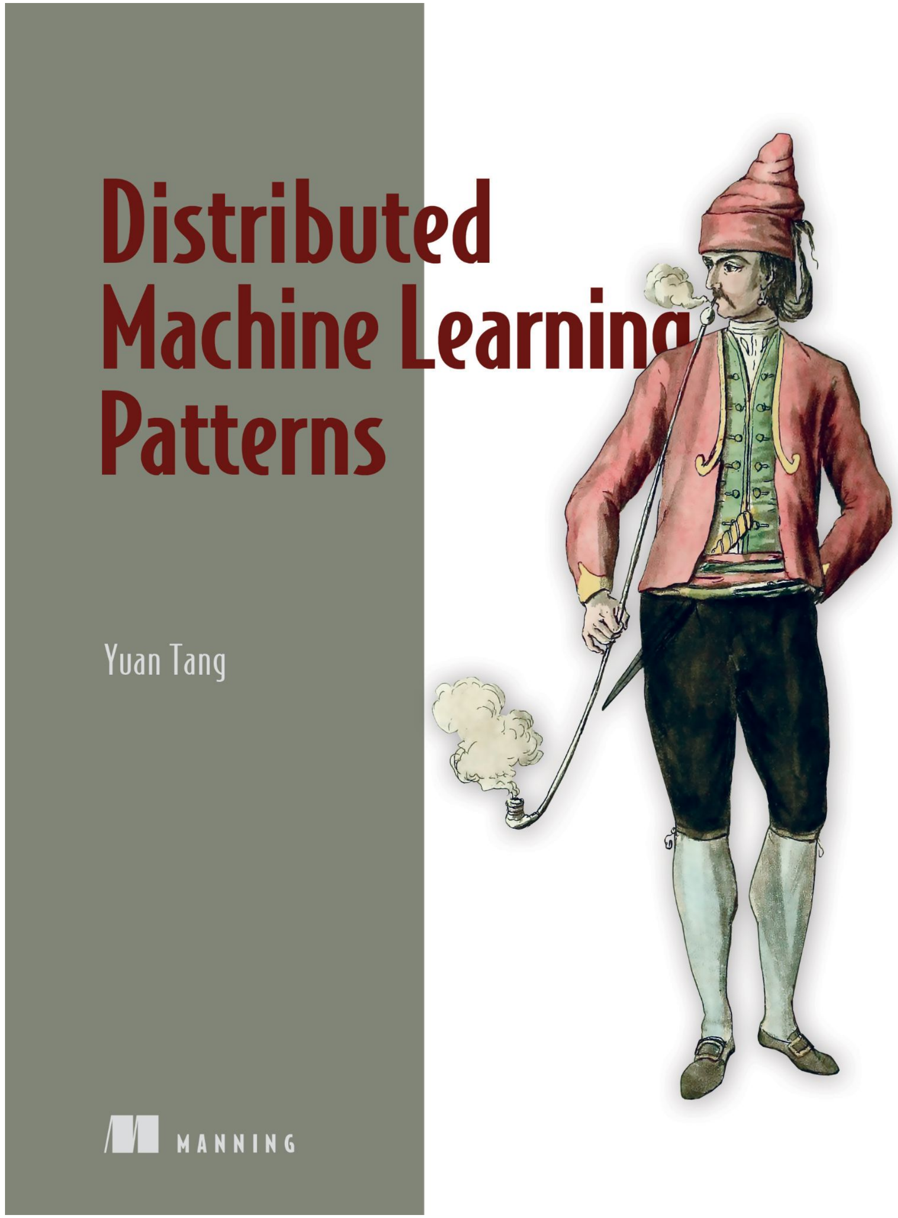
Cloud Native Production-ready AI Platform
6. Iterations



Savin & Yuan, KubeCon 2023

- modeling
- deployment
- versioning
- orchestration
- compute
- data

Distributed Machine Learning Patterns



http://mng.bz/QZgv