
KubeCon
CloudNativeCon
North America 2023


KubeCon | CloudNativeCon

North America 2023

The Future of Interactive Data Science at Scale with Jupyter and Kubeflow

Andrey Velichkevich - Apple
Zachary Sailer - Apple

Who are we?

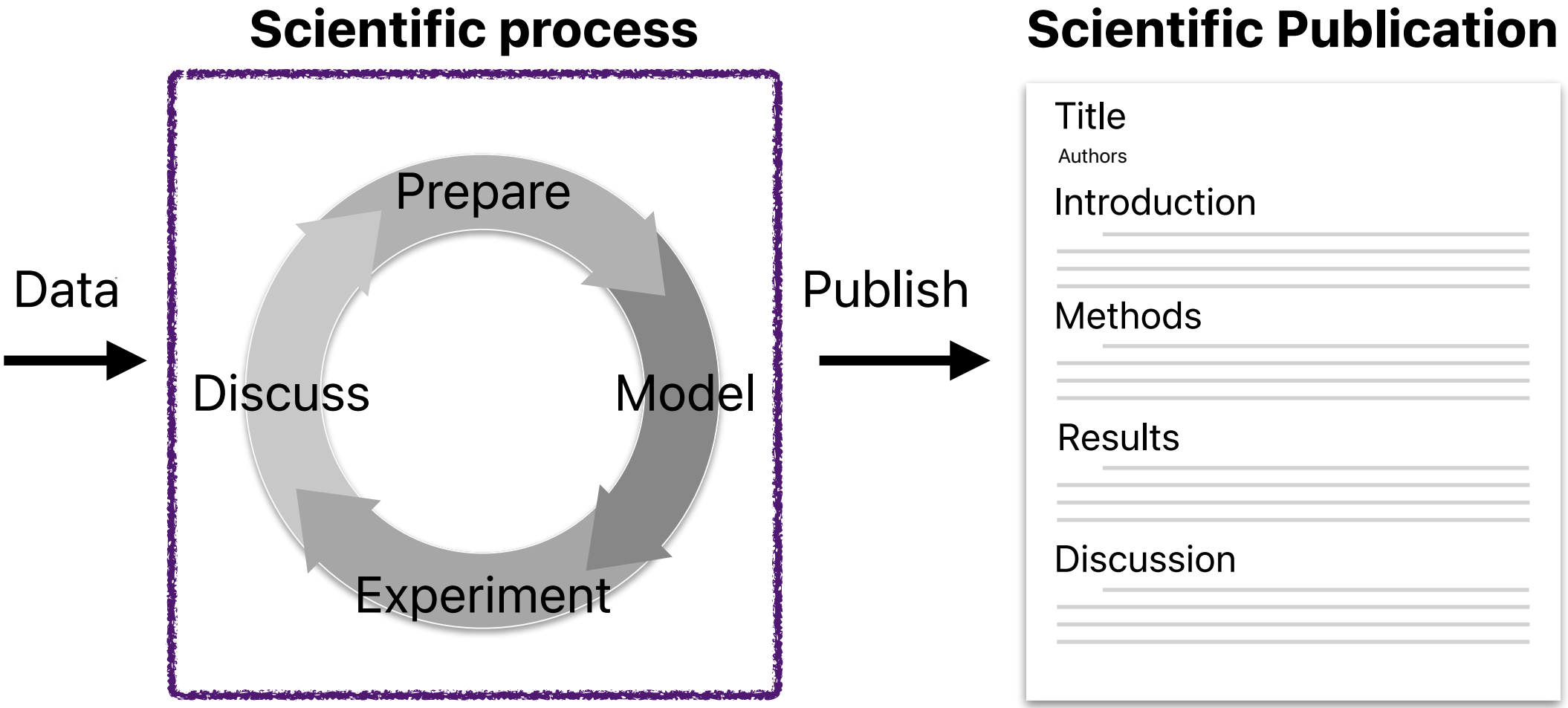
Physics • Jupyter • Software

Science • Kubeflow • MLOps

Let us clearly define what we mean...

> The **future** of **interactive** data **science** at **scale**

How (data) *science* is done in practice

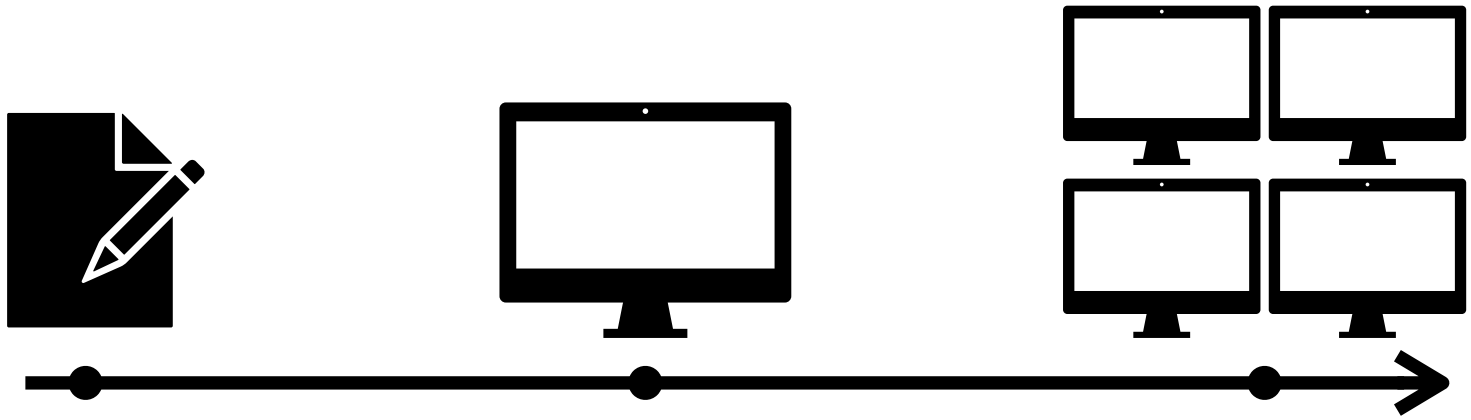


Science was done with **pencil** and **paper**.

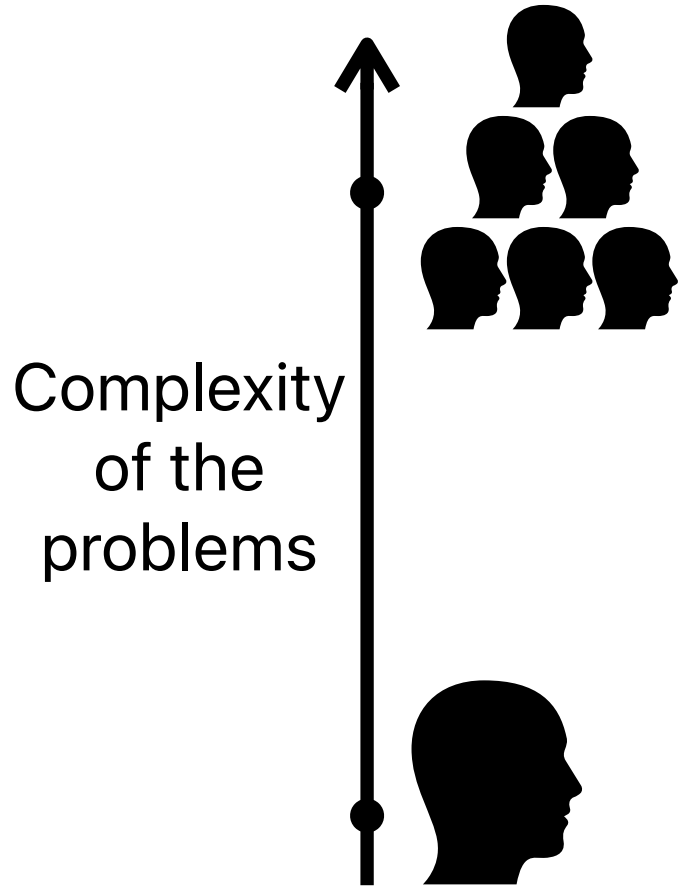
1923

Data is **big** and **complex.**

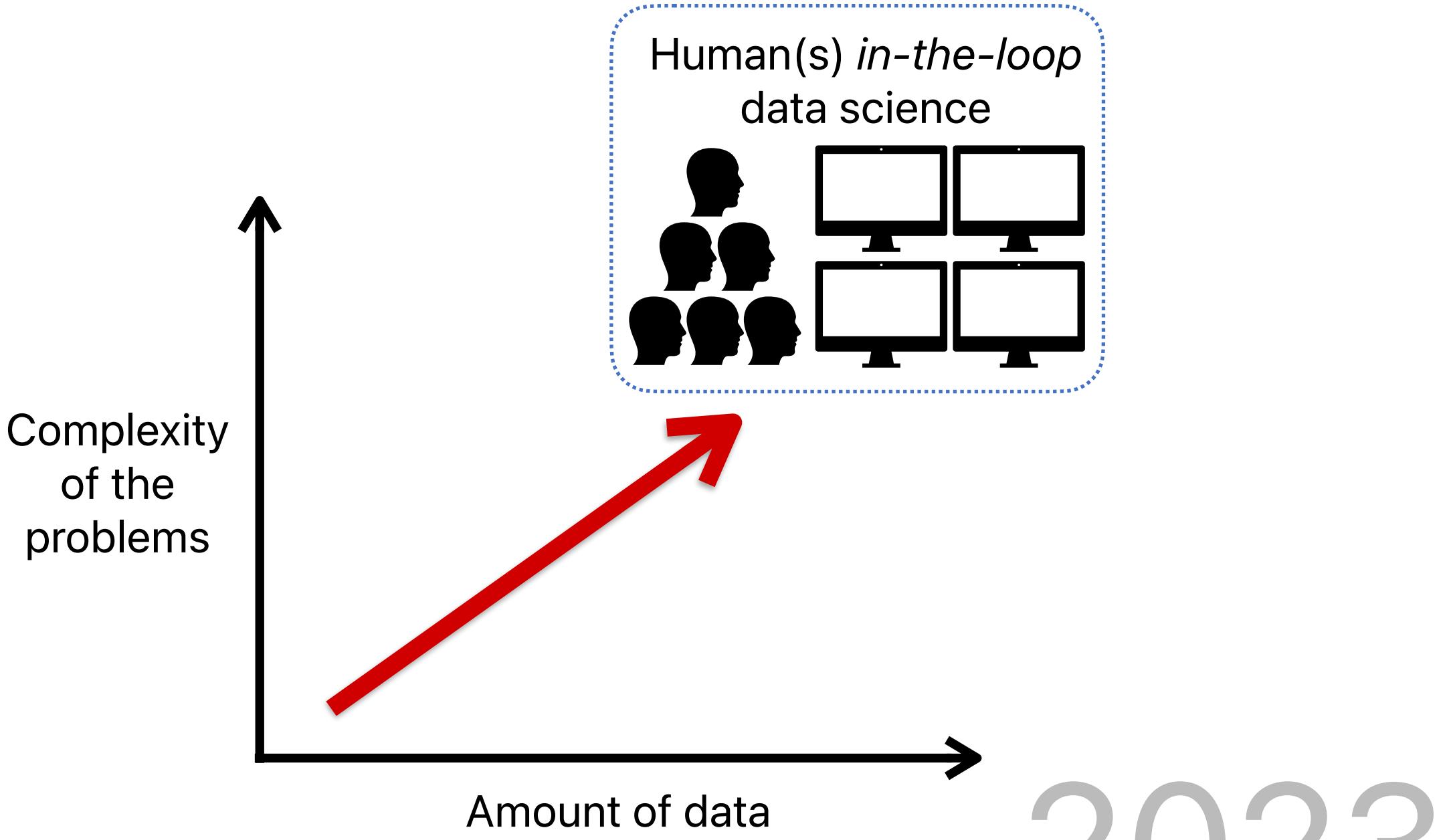
2023



2023

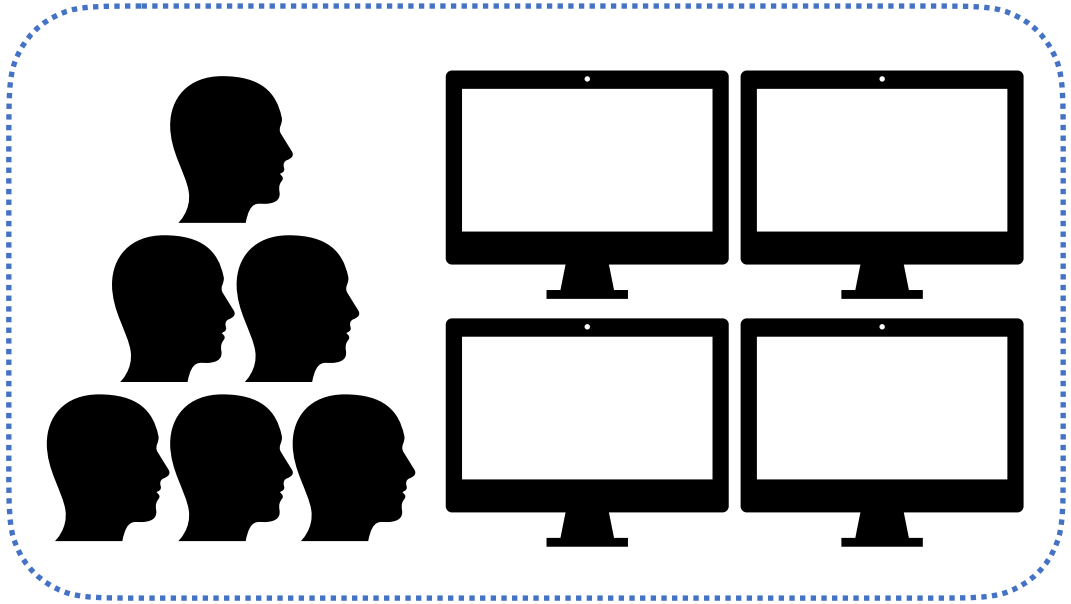


2023


Human(s) *in-the-loop*
data science
Complexity
of the
problems
Amount of data

2023

We need tools that promote
humans-in-the-loop data science



2023

Our requirements *for interactive Data Science*

Computational

Exploratory

Collaborative

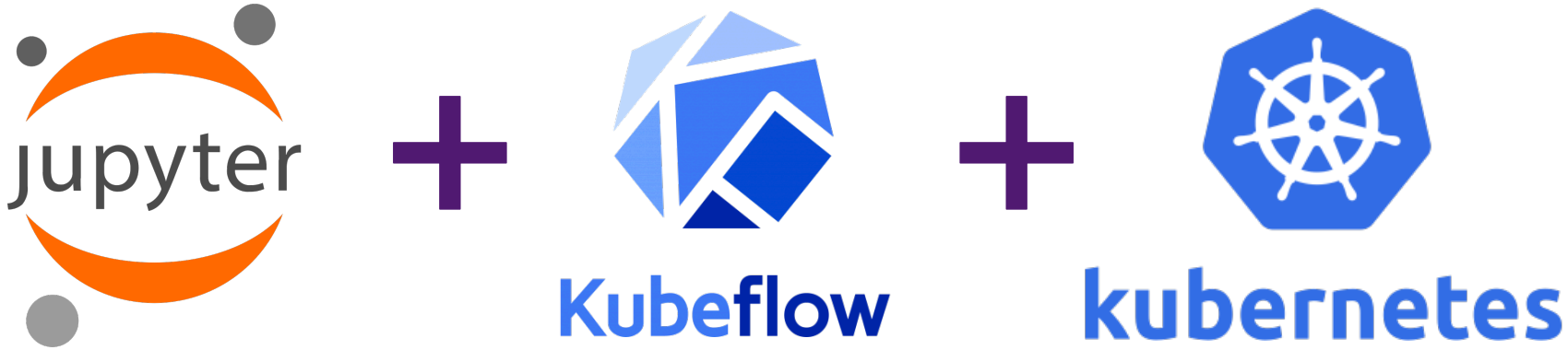
Publication ready

Built for AI/ML

Simple to scale

Demo

What you just saw



Why Jupyter?

File Edit View Run Kernel Git Tabs Settings Help External Links

bayesianStatistics.ipynb

Code git Share Python3.8

Bayesian Statistics

Bayesian statistics is an approach to statistics based on the Bayesian interpretation of probability, where probability expresses a degree of belief in an event.

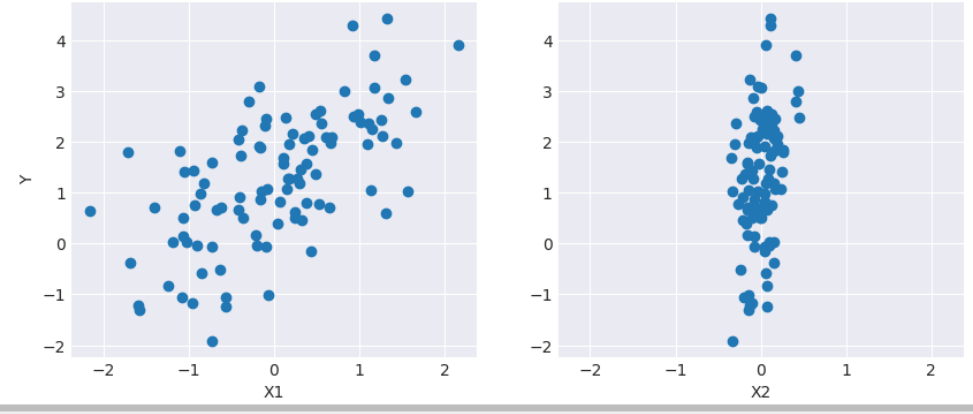
Generate Data

Bayesian statistics begins with model definition and then proceeds to fitting and posterior analysis. We'll consider a simple Bayesian linear regression model with normal priors for the parameters. Let's generate some artificial data for the model using NumPy's random module.

[19]:
```
import numpy as np
import matplotlib.pyplot as plt

# True parameter values
alpha, sigma, beta, size = 1, 1, [1, 2.5], 100

# Predictor variable
X1 = np.random.randn(size)
X2 = np.random.randn(size) * 0.2

# Simulate outcome variable
Y = alpha + beta[0]*X1 + beta[1]*X2 + np.random.randn(size)*sigma
```
Last executed at 2023-10-30 20:29:06 in 4ms

Here is what the simulated data look like in `matplotlib`.

[25]:
```
fig, axes = plt.subplots(1, 2, sharex=True, figsize=(10,4))
axes[0].scatter(X1, Y); axes[1].scatter(X2, Y); axes[0].set_ylabel('Y'); axes[0].set_xlabel('X1'); axes[1].set_xlabel('X2');
```
Last executed at 2023-10-30 20:36:41 in 131ms



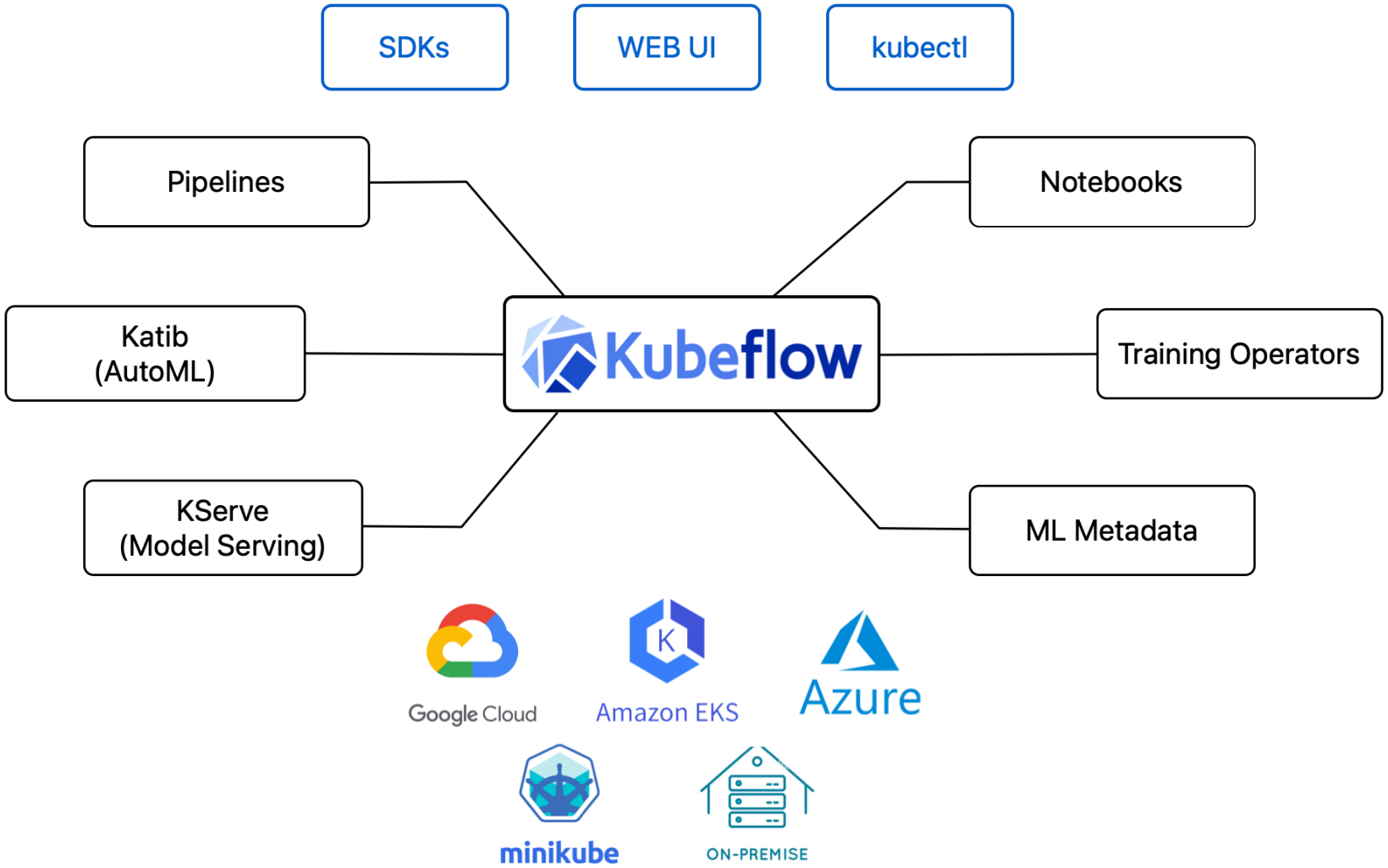
Simple 1 0 Python3.8 | Idle File size: 59.3 KB / 2 MB Mode: Command Ln 2, Col 12 bayesianStatistics.ipynb 0

- ✓ Computational
- ✓ Exploratory
- ✓ Collaborative
- ✓ Publication ready

Built for AI/ML

Simple to scale

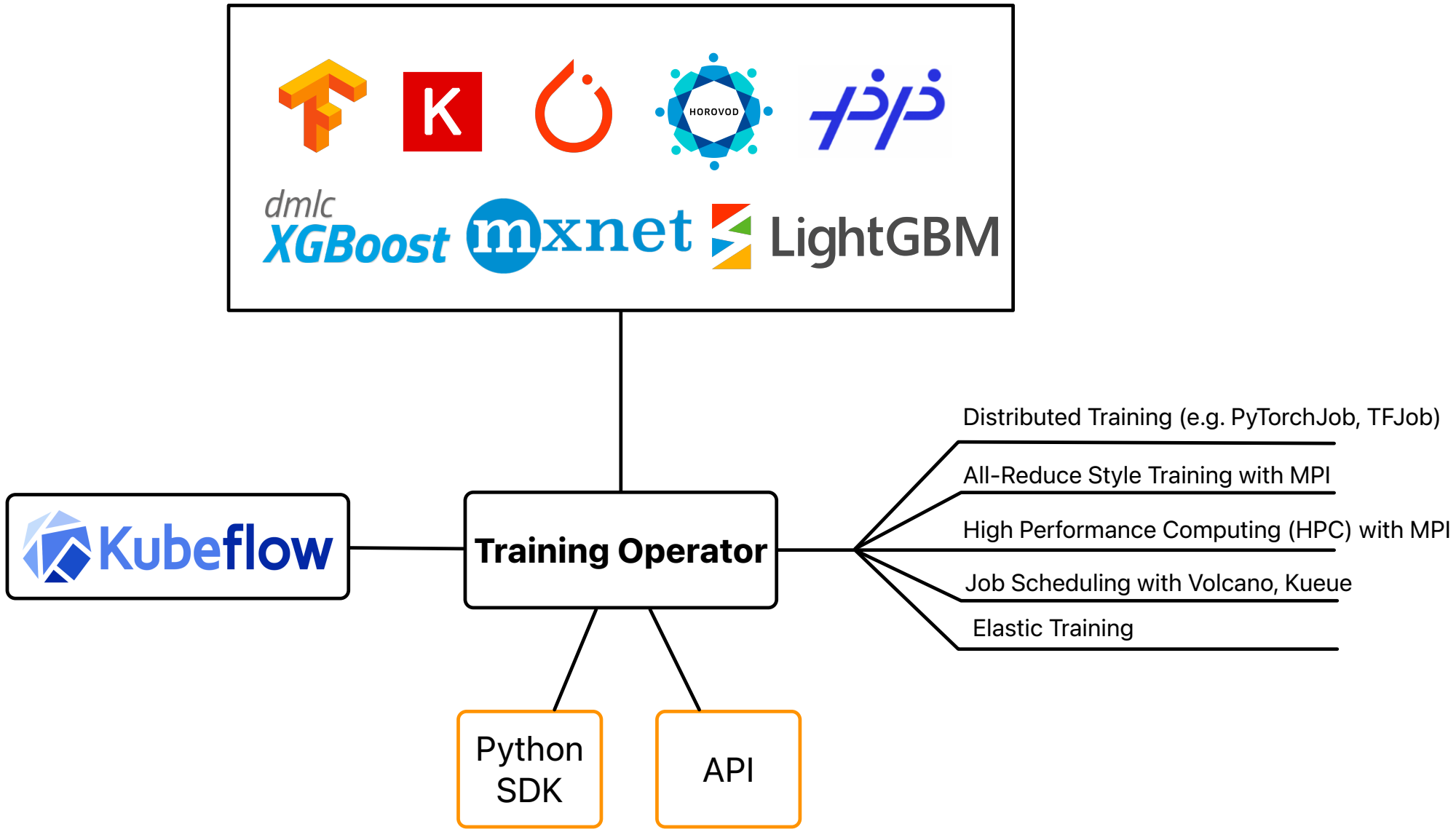
Why Kubeflow ?



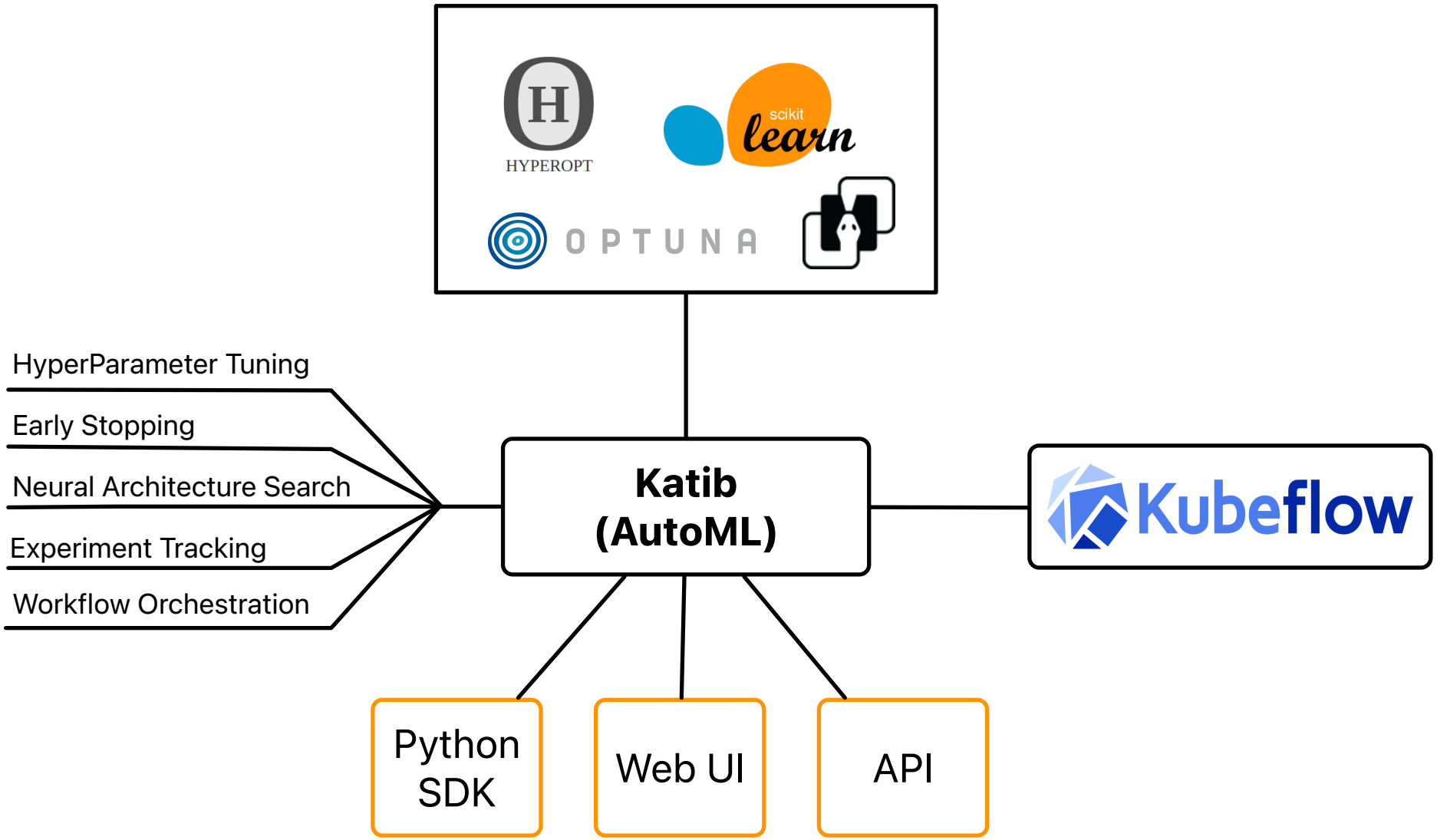
Computational
Exploratory
Collaborative
Publication ready

✓ Built for AI/ML
✓ Simple to scale

Kubeflow Overview - Training Operator



Kubeflow Overview - Katib

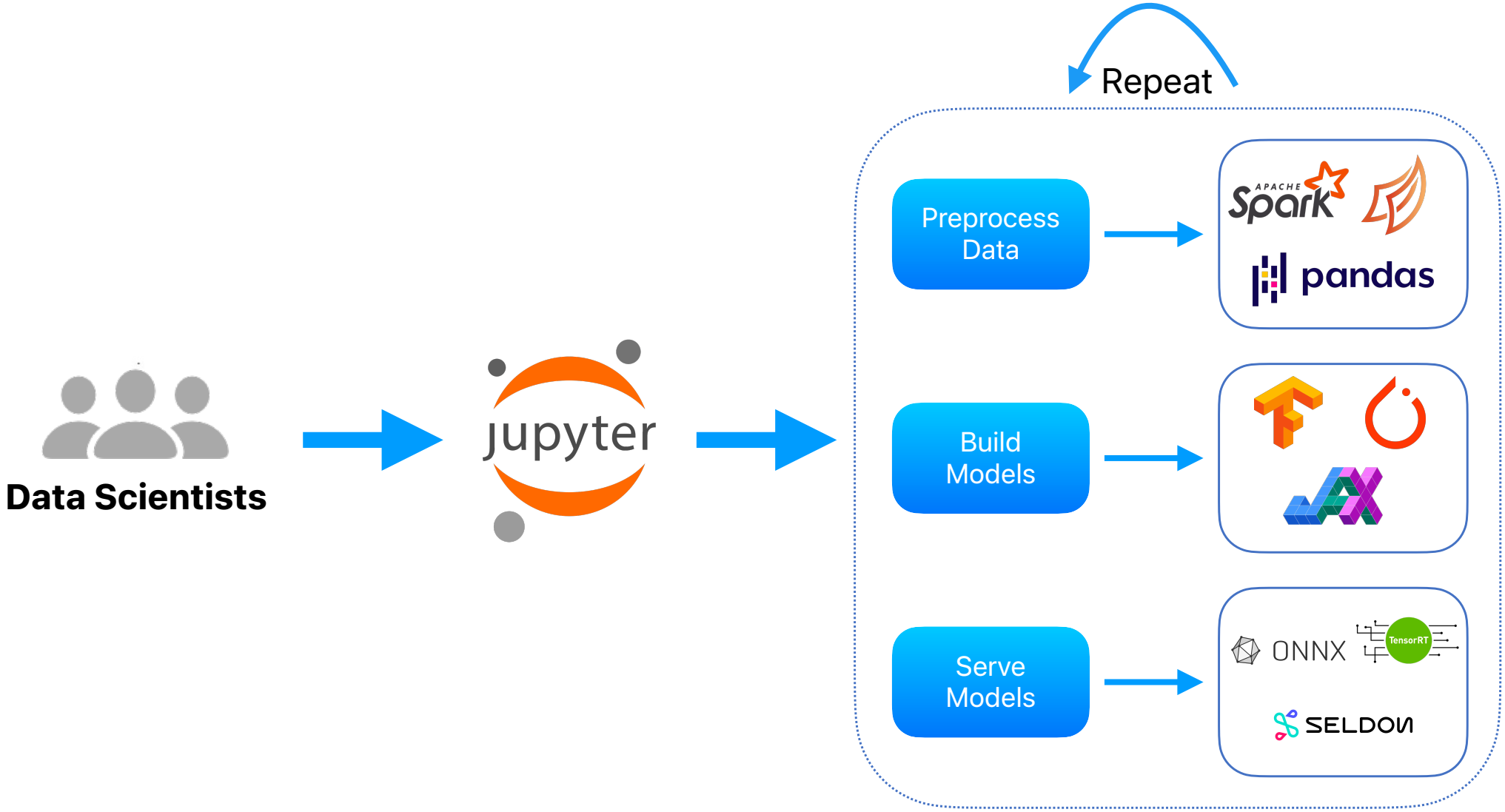


How does Kubeflow simplify scale?

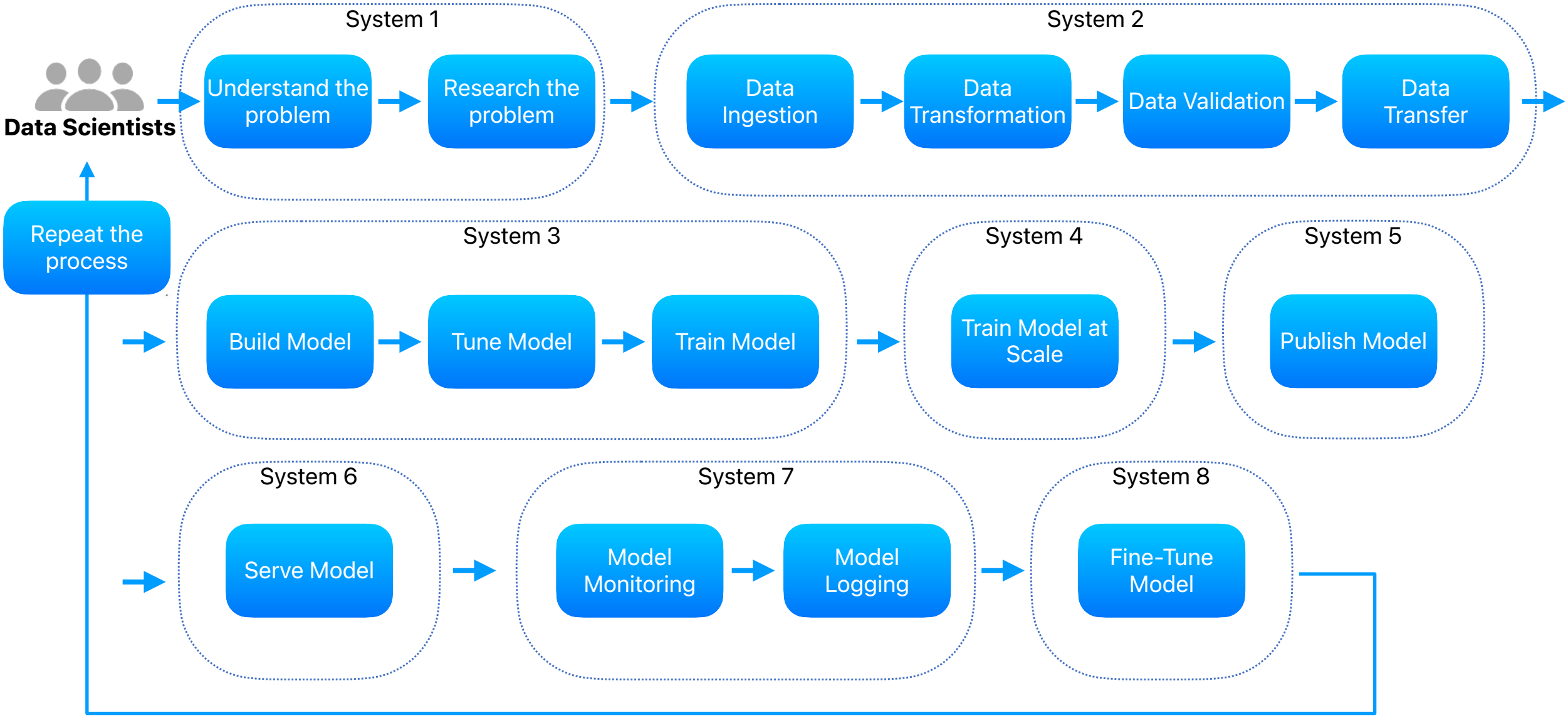
Computational
Exploratory
Collaborative
Publication ready
✓ **Built for AI/ML**
✓ Simple to scale

Kubeflow

Lifecycle of ML - Desired

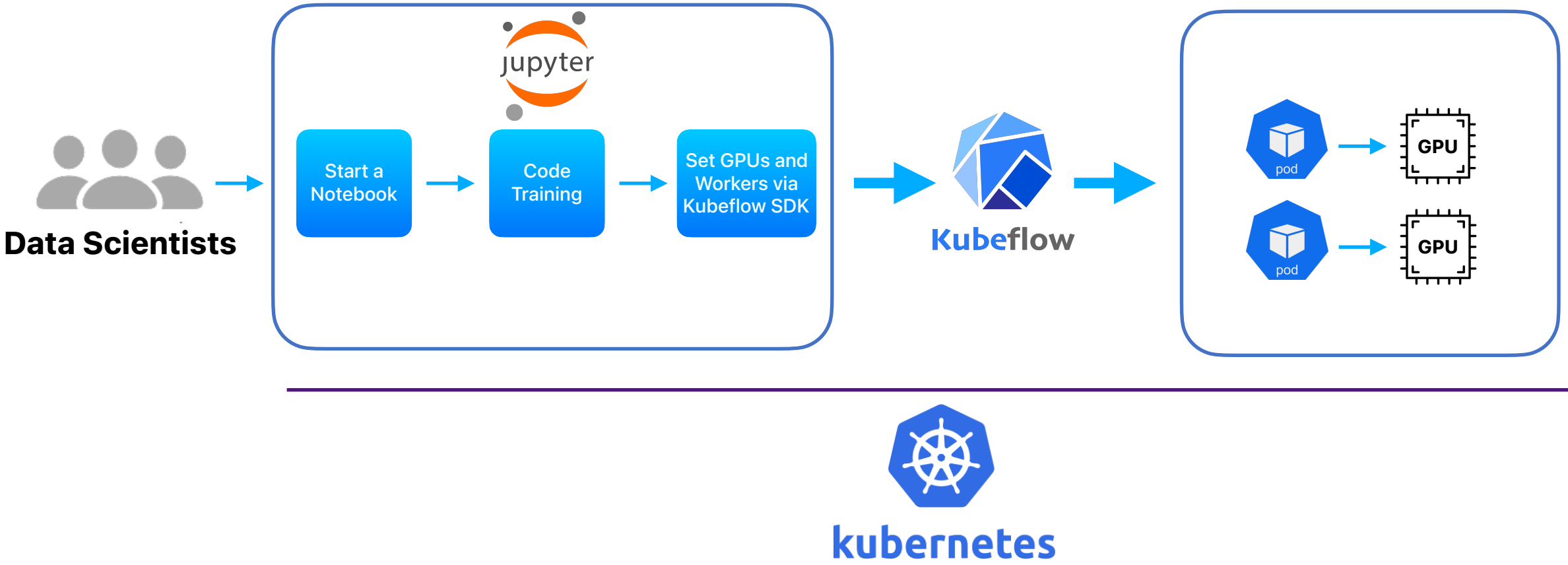


Lifecycle of ML - Reality



Adding machines should be
effortless

Simple to Scale ML Experience



Simple Experiment for Training

[5]:
```python
def train_model():
    import torch
    ...
    # Set dist strategy.
    torch.distributed.init_process_group(
        backend="nccl", rank=RANK, world_size=WORLD_SIZE
    )
    Distributor = nn.parallel.DistributedDataParallel
    model = Distributor(model)

    # Train model
    model.train()
```

[6]:
```python
from kubeflow.training import TrainingClient

job_name = "train-job"

TrainingClient().create_job(
    name=job_name,
    func=train_model,
    num_worker_replicas=100,
    resources_per_worker={"gpu": "1"},
)
```

[7]:
```python
TrainingClient().get_job_logs(name=job_name)
```

Last executed at 2023-01-18 12:52:45 in 63ms

```
2023-01-18T12:44:42Z INFO     Start training for RANK: 0. WORLD_SIZE: 4
2023-01-18T12:44:42Z INFO     Train Epoch: 0 [0/60000 (0%)]     loss=2.3032
2023-01-18T12:44:42Z INFO     Reducer buckets have been rebuilt in this iteration.
2023-01-18T12:44:43Z INFO     Train Epoch: 0 [320/60000 (1%)]   loss=2.2963
2023-01-18T12:44:44Z INFO     Train Epoch: 0 [640/60000 (1%)]   loss=2.2893
2023-01-18T12:44:45Z INFO     Train Epoch: 0 [960/60000 (2%)]   loss=2.2707
2023-01-18T12:44:46Z INFO     Train Epoch: 0 [1280/60000 (2%)]  loss=2.2700
2023-01-18T12:44:47Z INFO     Train Epoch: 0 [1600/60000 (3%)]  loss=2.2697
2023-01-18T12:44:48Z INFO     Train Epoch: 0 [1920/60000 (3%)]  loss=2.2424
2023-01-18T12:44:49Z INFO     Train Epoch: 0 [2240/60000 (4%)]  loss=2.1990
2023-01-18T12:44:50Z INFO     Train Epoch: 0 [2560/60000 (4%)]  loss=2.1941
2023-01-18T12:44:51Z INFO     Train Epoch: 0 [2880/60000 (5%)]  loss=2.1618
```

Simple Experiment for Tuning

KubeCon CloudNativeCon North America 2023

[1]:
```python
def train_model(parameters):
    import tensorflow

    ...
    model.fit(
        dataset,
        epochs=int(parameters["num_epochs"]),
        steps_per_epoch=10
    )
```

[2]:
```python
import kubeflow.katib as katib

job_name = "tune-job"

katib.KatibClient().tune(
    name=job_name,
    objective=train_model,
    parameters={
        "num_epochs": katib.search.int(min=6, max=7)
    },
    objective_metric_name="Accuracy",
    max_trial_count=100,
    parallel_trial_count=10,
    resources_per_trial={"gpu": "1"},
)
```

[3]:
```python
BEST_TRIAL = katib.KatibClient().get_optimal_hyperparameters(
    name=job_name
)

BEST_TRIAL.parameter_assignments[0]
```

Last executed at 2023-09-28 12:53:21 in 29ms

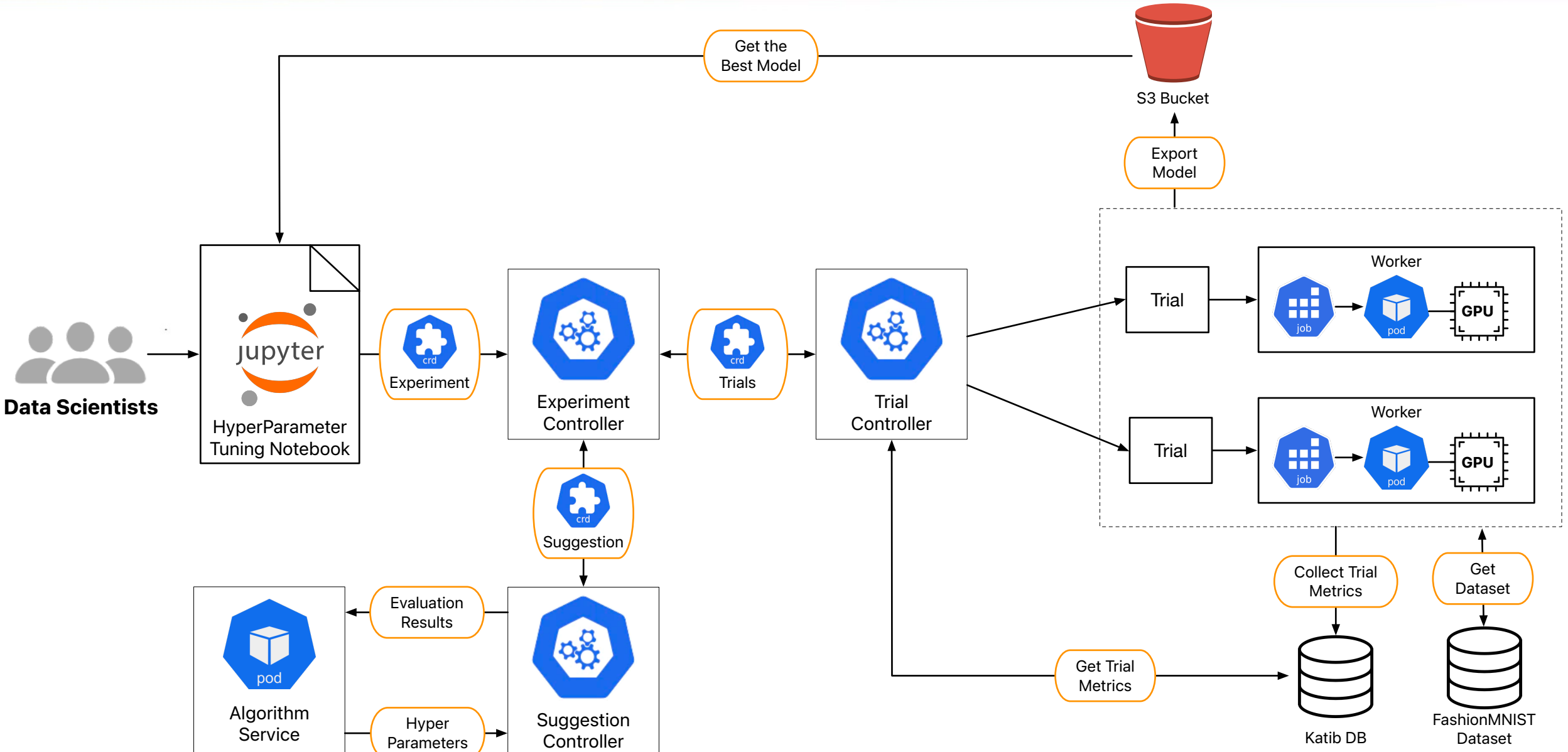
```
{'name': 'num_epochs', 'value': '7'}
```

[4]:
```python
BEST_TRIAL.observation.metrics[0]
```

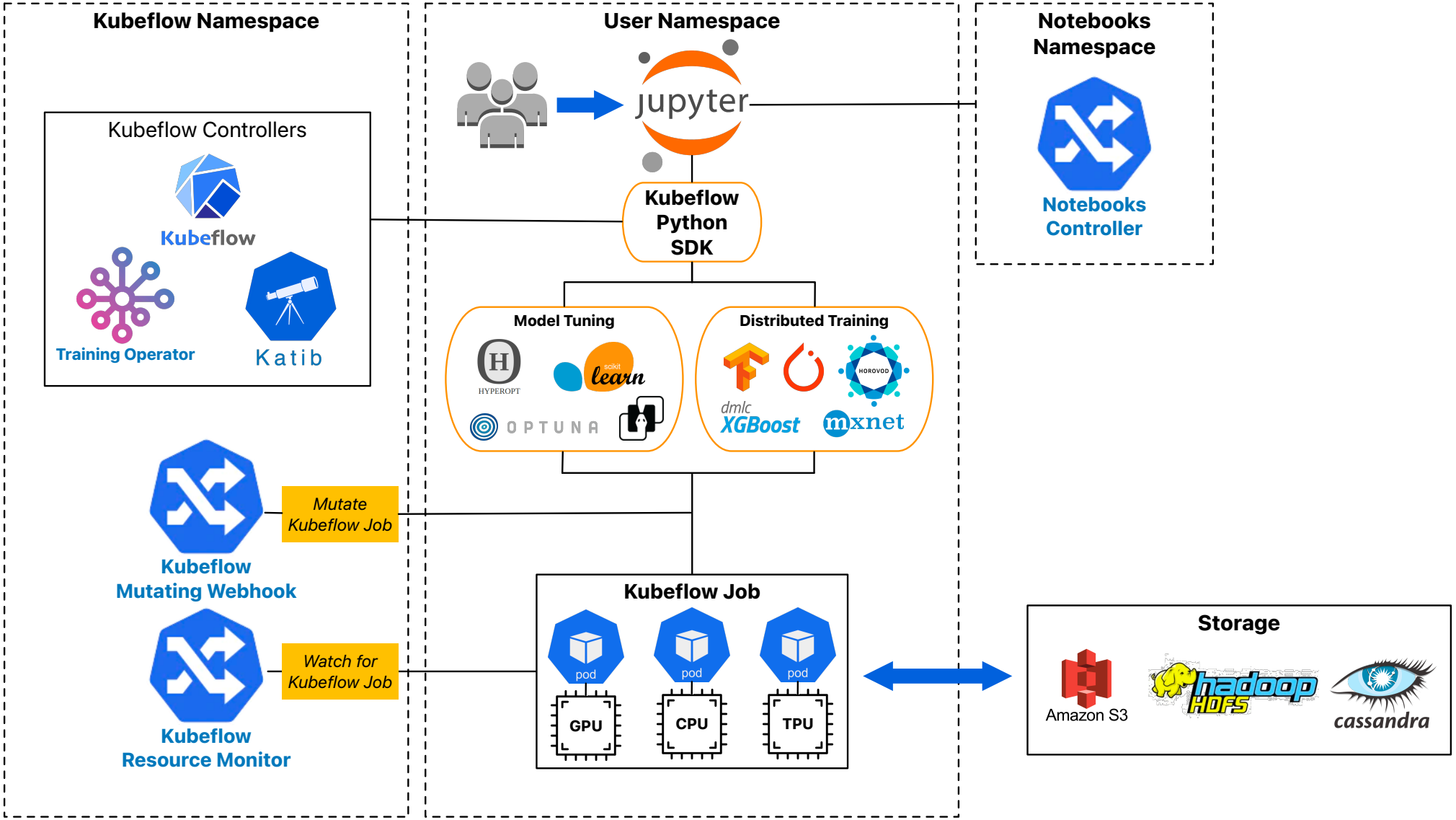
Last executed at 2023-09-28 12:53:21 in 3ms

```
{'latest': '0.8734',
 'max': '0.8734',
 'min': '0.6583333333333333',
 'name': 'Accuracy'}
```

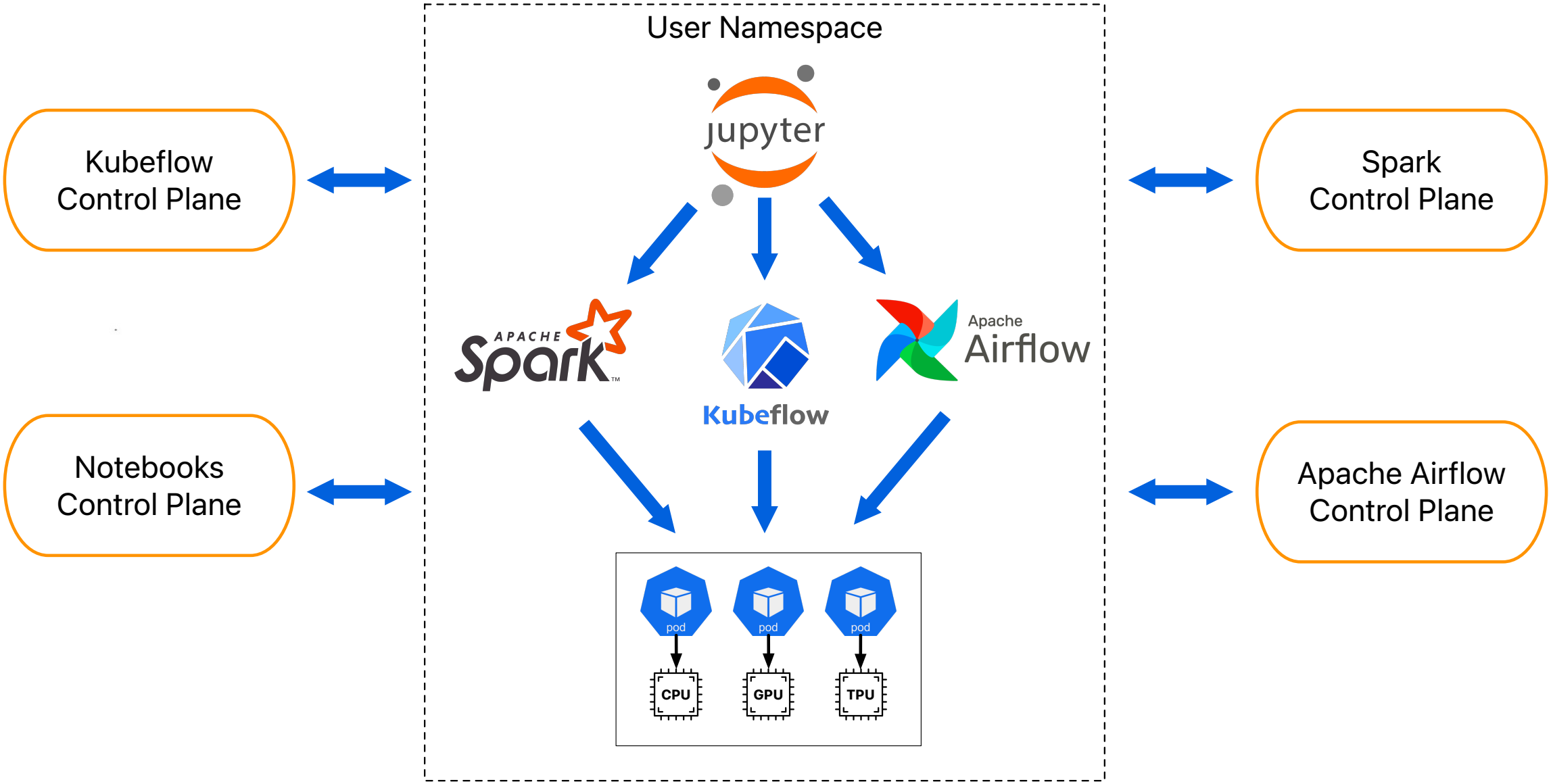
The complexity you will never know about



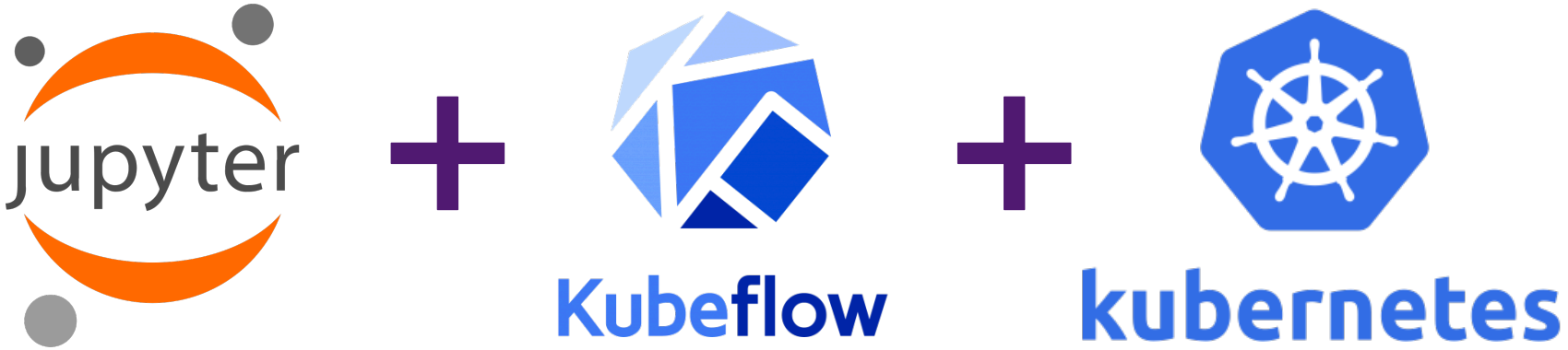
Notebooks & Kubeflow Architecture



Isolated User Environment



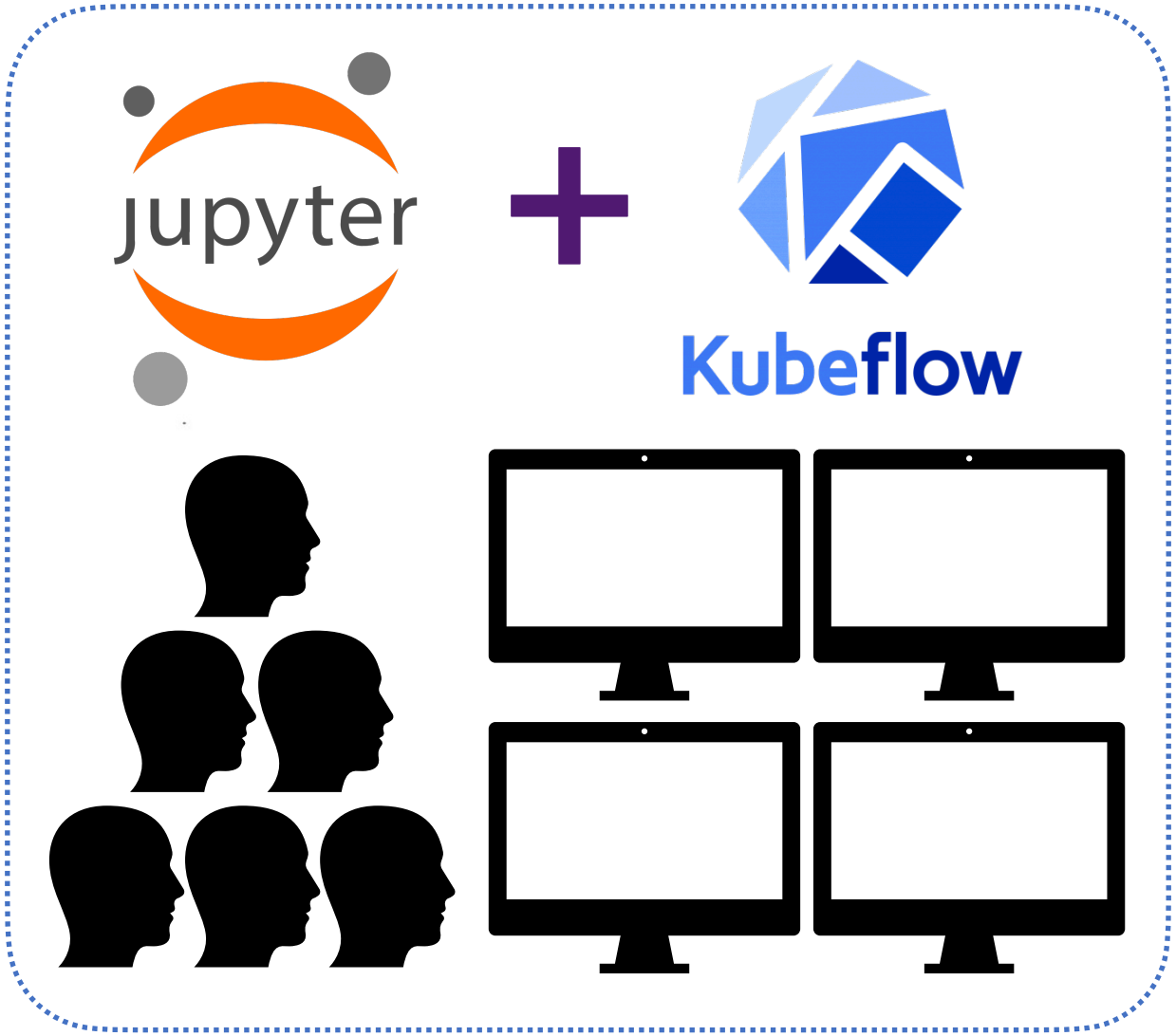
Jupyter + Kubeflow



Interactive • Simple • Portable • Scalable

ML Platform on Kubernetes

Interactive Data Science at Scale



- ✓ Computational
- ✓ Exploratory
- ✓ Collaborative
- ✓ Publication ready
- ✓ Built for AI/ML
- ✓ Simple to scale

Jupyter and Kubeflow Communities

Kubeflow

- Attend the Kubeflow AutoML and Training community meetings
- Join Kubeflow Slack Channels
- How to contribute ?
    - Check the Kubeflow Training Operator and Katib GitHub
    - Issues with the "Help Wanter" Label
    - Submit a new Enhancement Proposal to Kubeflow

Jupyter

- Attend *any* public Jupyter meeting
- How to contribute ?
    - Join a "Contributing Hour"
    - JupyterLab and Jupyter Collaboration Projects

Kubeflow & CNCF

Source: https://www.cncf.io/blog/2023/07/25/kubeflow-brings-mlops-to-the-cncf-incubator

Please scan the QR Code above
to leave feedback on this session