

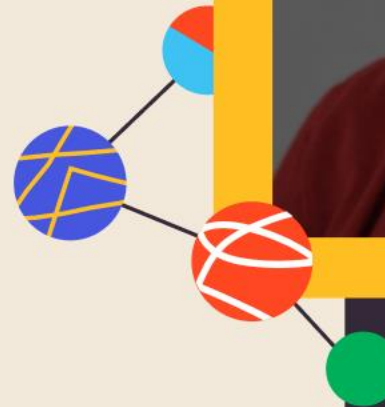
W> JavaScript
Congress 2021

ONLINE
NOVEMBER
23-25

Ron Dagdag

Lead Software Engineer at Spacee and Microsoft MVP

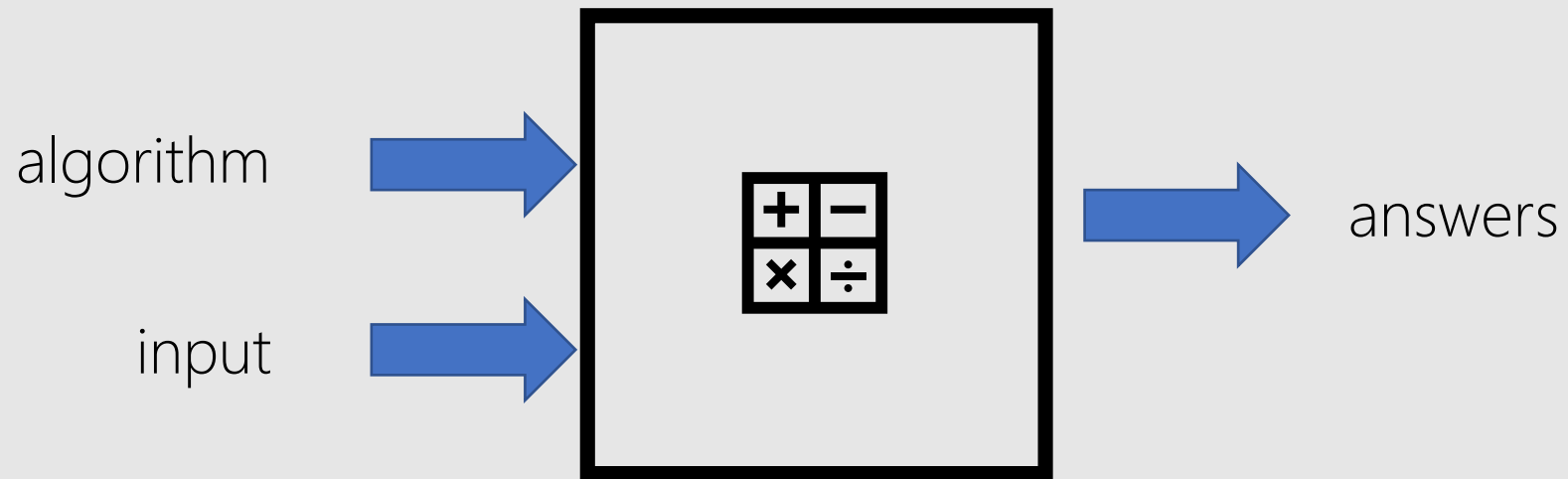
jscongress.com



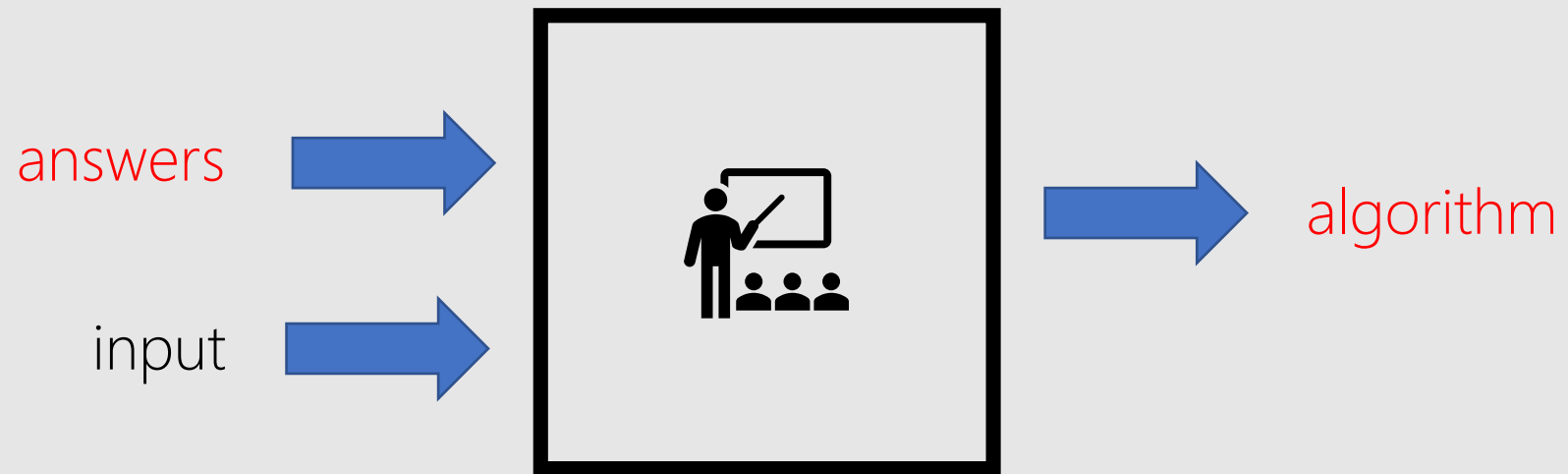
ONNX
Not ONIX
Not ONYX



programming



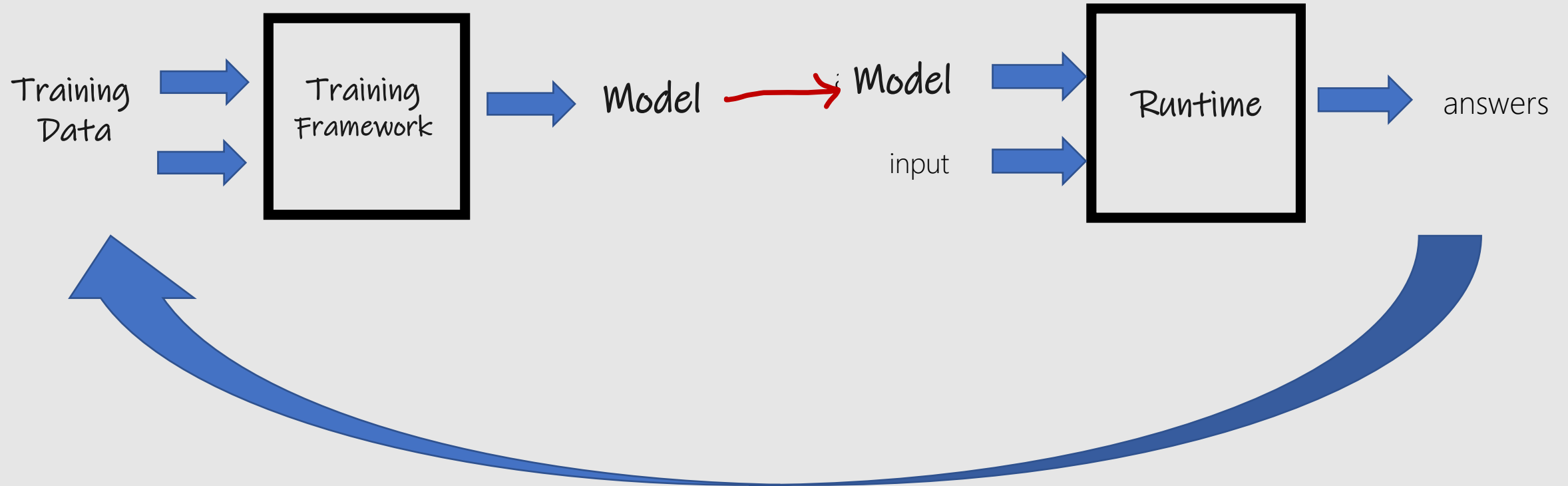
machine learning



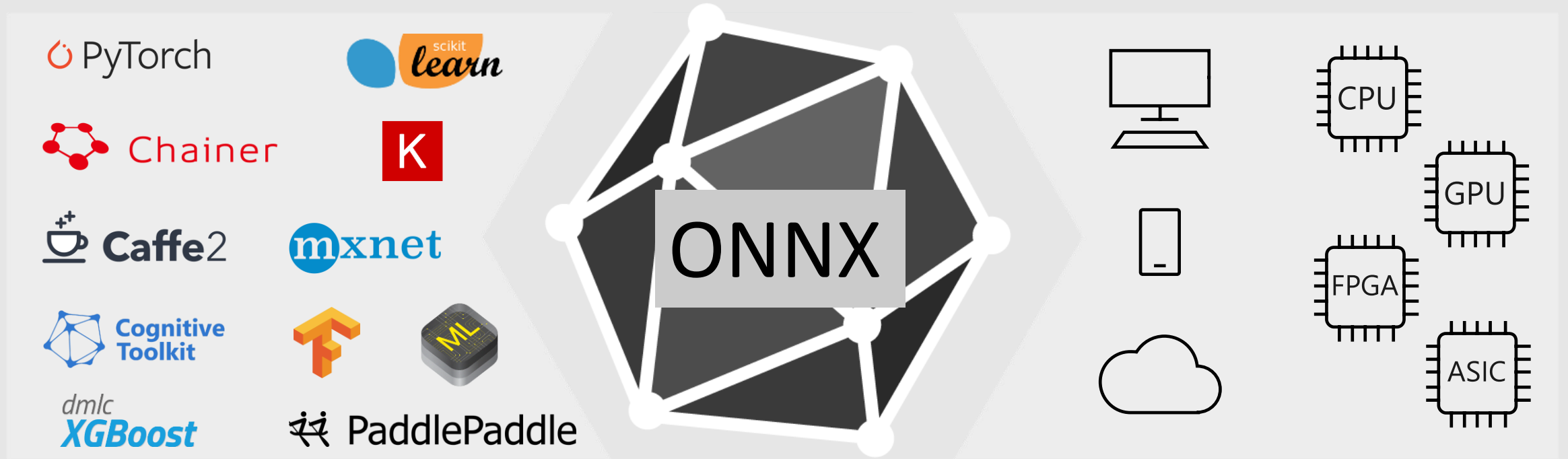
ML Primer

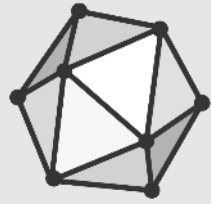
machine Learning

Inferencing



Open and Interoperable AI





ONNX

Open Neural Network Exchange

Open format for ML models

github.com/onnx

onnx.ai/



ONNX Partners

ABBYY®

Alibaba Group
阿里巴巴集团

AMD

arm

aws

Baidu 百度

BECKHOFF

BITMAIN

cadence®

CEVA®

Facebook
Open Source

GRAPHCORE

habana

HAILO

Hewlett Packard
Enterprise

HUAWEI

IBM®

Idein Inc

intel AI

MathWorks®

MAXAR

MEDIATEK

mi

Microsoft

NVIDIA

NXP

OctoML

OPEN AI LAB
开放智能

Preferred
Networks

SIEMENS

SONY

Qualcomm

sas

商汤
sensetime

skymizer

SYNOPSYS®

Tencent

unity

verizon
media

vmware®

WOLFRAM

Yandex

ZETANE



OLFAI
GRADUATE
PROJECT

Agenda

- ✓ What is ONNX, When to use ONNX
- ☐ How to create ONNX models
- ☐ How to deploy ONNX models

When to use ONNX?

- Trained in Python - deploy into a C#/Java/Javascript app
- High Inferencing latency for production use
- Model to run resource on IoT/edge devices
- Model to run on different OS or Hardware
- Combine running models created from different frameworks
- Training takes too long (transformer models)

Create

Frameworks



Native support

Converters

Services



Native support



Deploy

Cloud Services

Azure Machine Learning services

Ubuntu VM

Windows VM

Windows Devices

IoT/Edge Devices

Other Devices
(iOS, Android, etc)

Native support

Converters

Frameworks



**Step 1:
Create**

Services



Azure Custom
Vision Service

Native
support

Converters

Native
support



ONNX Model

Azure

Azure Machine Learning services

Ubuntu VM

Windows Server 2019 VM



**Step 2:
Deploy**

Other Devices
(iOS, etc)

Native
support

Converters

A still life composition featuring several brown eggs in a cardboard carton, a pair of wire-rimmed glasses, and an open book. The text "Secret Recipe" is overlaid in white, with a vertical line separating the words. The background is dark and moody, with the objects resting on a reflective surface.

Secret Recipe

4 ways to get an ONNX model



ONNX Model Zoo



Azure Custom Vision Service



Convert existing models



Train models in Azure Machine Learning

Automated Machine Learning

ONNX Model Zoo: github.com/onnx/models

Image Classification

This collection of models take images as input, then classifies the major objects in the images into a set of predefined classes.

Model Class	Reference	Description
MobileNet	Sandler et al.	Efficient CNN model for mobile and embedded vision applications. Top-5 error from paper - ~10%
ResNet	He et al., He et al.	Very deep CNN model (up to 152 layers), won the ImageNet Challenge in 2015. Top-5 error from paper - ~3.6%
SqueezeNet	Iandola et al.	A lightweight CNN model with fewer layers and parameters. Top-5 error from paper - ~4.8%
VGG	Simonyan et al.	Deep CNN model, won the ImageNet Challenge in 2014. Top-5 error from paper - ~7.4%

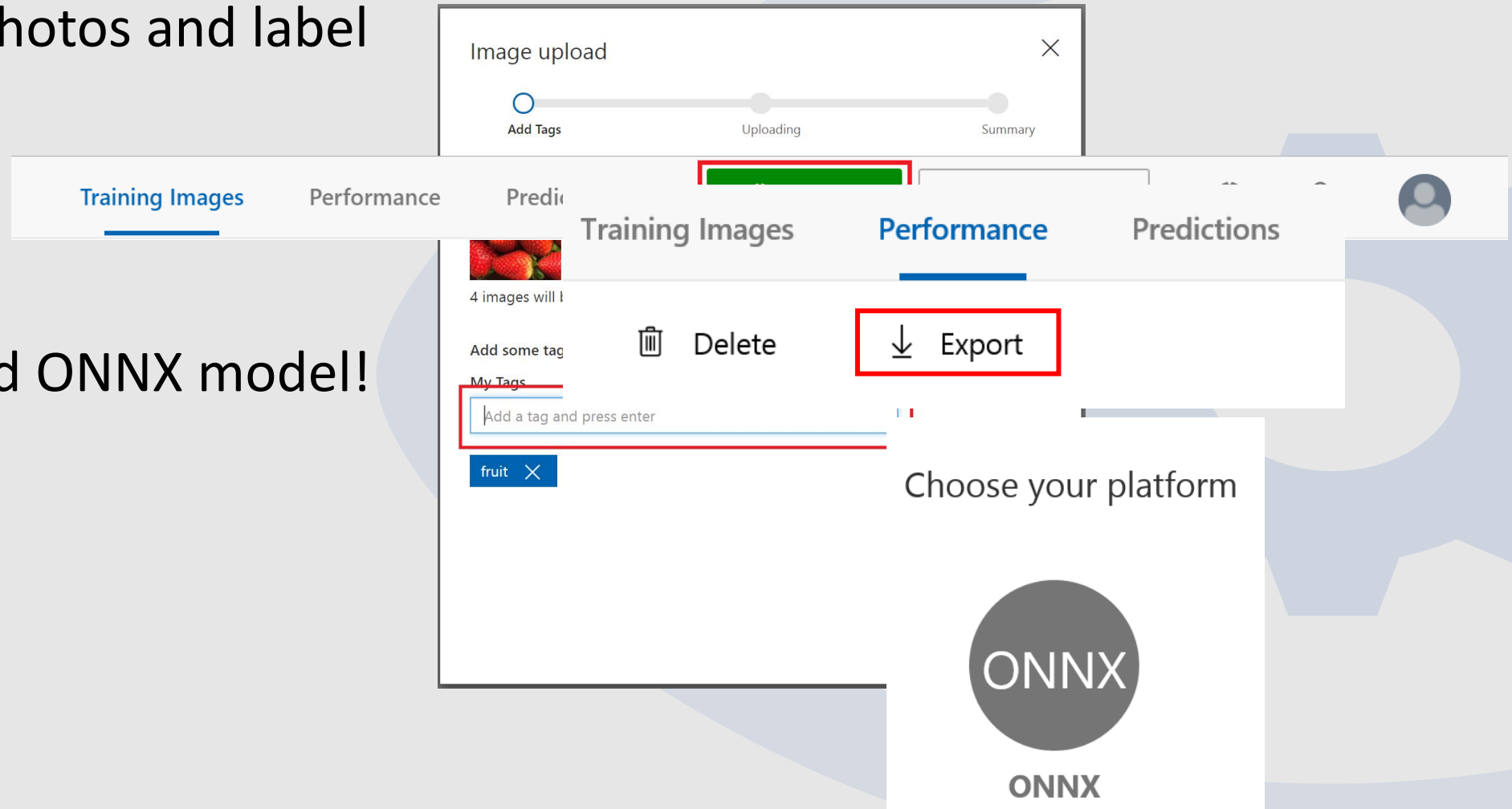
Model	Download	Checksum	Download (with sample test data)	ONNX version	Opset version	Top-1 accuracy (%)	Top-5 accuracy (%)
ResNet-18	44.6 MB	MD5	42.9 MB	1.2.1	7	69.70	89.49
ResNet-34	83.2 MB	MD5	78.6 MB	1.2.1	7	73.36	91.43
ResNet-50	97.7 MB	MD5	92.0 MB	1.2.1	7	75.81	92.82
ResNet-101	170.4 MB	MD5	159.4 MB	1.2.1	7	77.42	93.61
ResNet-152	230.3 MB	MD5	216.0 MB	1.2.1	7	78.20	94.21

Custom Vision Service: customvision.ai

1. Upload photos and label

2. Train

3. Download ONNX model!



Convert
models



LibSVM

MATLAB®



SIEMENS



Convert models

1. Load existing model
2. (Convert to ONNX)
3. Save ONNX model



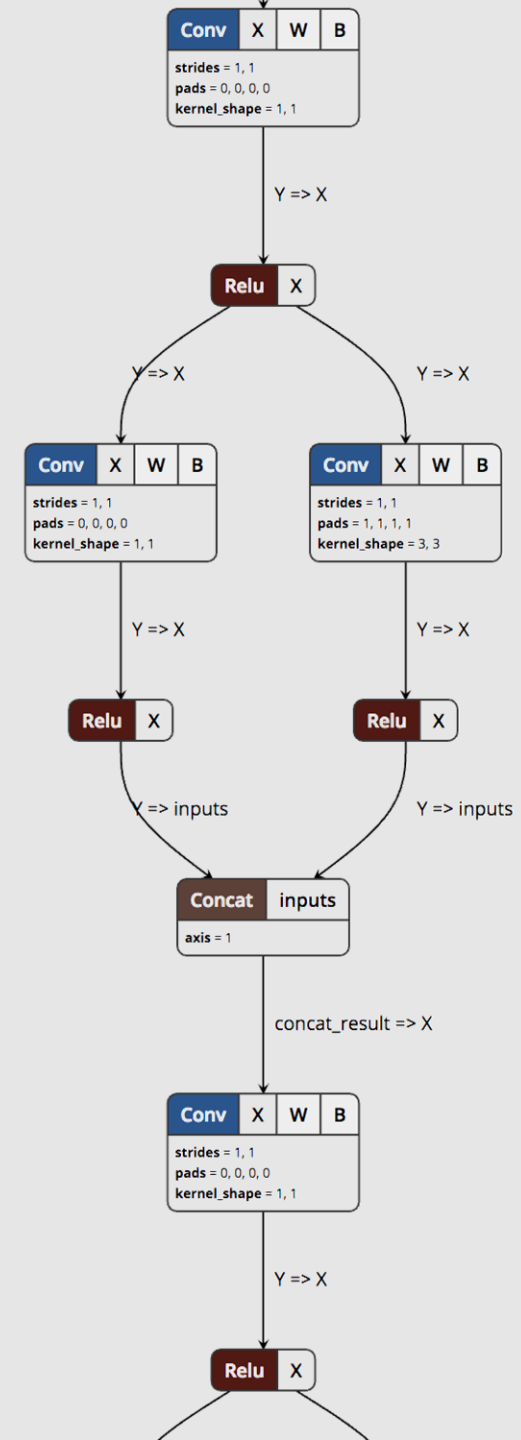
ONNX Models

Graph of operations

Netron

<https://netron.app/>

<https://lutzroeder.github.io/netron/>



Convert models: PyTorch

```
import torch  
import torch.onnx
```

```
model = torch.load("model.pt")
```

```
sample_input = torch.randn(1, 3, 224, 224)
```

```
torch.onnx.export(model, sample_input, "model.onnx")
```



Convert models: Keras

```
In [ ]: import onnxmltools
        from keras.models import load_model
```

```
In [ ]: # Update the input name and path for your Keras model
        input_keras_model = 'model.h5'

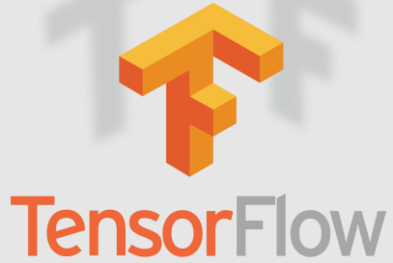
        # Change this path to the output name and path for the ONNX model
        output_onnx_model = 'model.onnx'
```

```
In [ ]: # Load your Keras model
        keras_model = load_model(input_keras_model)

        # Convert the Keras model into ONNX
        onnx_model = onnxmltools.convert_keras(keras_model)

        # Save as protobuf
        onnxmltools.utils.save_model(onnx_model, output_onnx_model)
```

Convert models:



```
> python -m tf2onnx.convert  
    --saved-model tensorflow-model-path  
    --output model.onnx
```

<https://github.com/onnx/tensorflow-onnx>



Convert models:



```
# Train a model.
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
iris = load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y)
clr = RandomForestClassifier()
clr.fit(X_train, y_train)

# Convert into ONNX format
from skl2onnx import convert_sklearn
from skl2onnx.common.data_types import FloatTensorType
initial_type = [('float_input', FloatTensorType([None, 4]))]
onx = convert_sklearn(clr, initial_types=initial_type)
with open("rf_iris.onnx", "wb") as f:
    f.write(onx.SerializeToString())
```

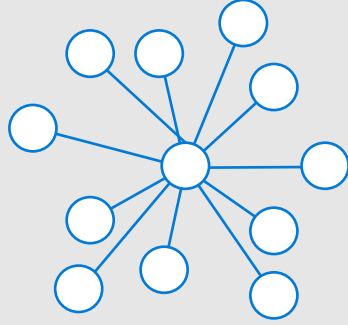


ONNX as an intermediary format

- **Convert to Tensorflow for Android**
 - [Convert a PyTorch model to Tensorflow using ONNX](#)
- **Convert to CoreML for iOS**
 - <https://github.com/onnx/onnx-coreml>
- **Fine-tuning an ONNX model with MXNet/Gluon**
 - https://mxnet.apache.org/versions/1.3.1/tutorials/onnx/fine_tuning_gluon.html

<https://github.com/onnx/tutorials>

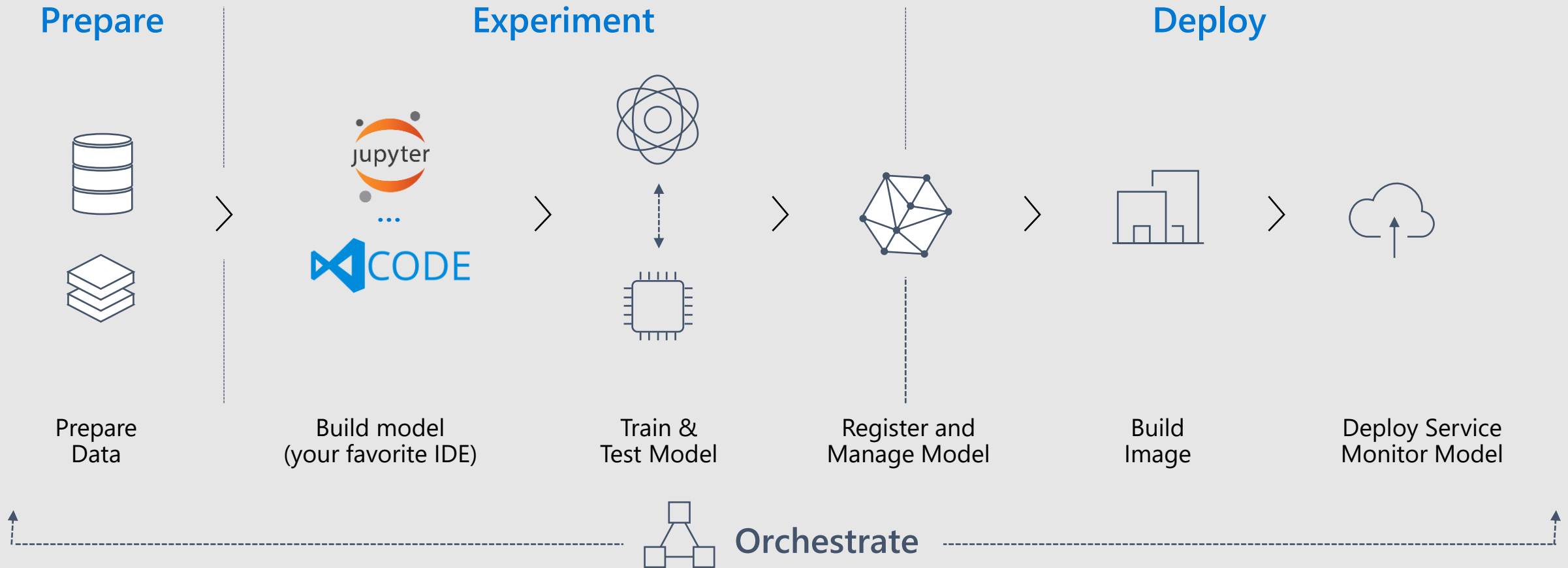
Train models in the Cloud



- Experiment locally then quickly scale with GPU clusters in the cloud
- Use automated machine learning and hyper-parameter tuning.
- Keeping Track of experiments, manage models, and easily deploy with integrated CI/CD tooling

Machine Learning

Typical E2E Process



Frameworks



**Step 1:
Create**

Services



Azure Custom
Vision Service

Native
support

Converters

Native
support



ONNX Model

Azure

Azure Machine Learning services

Ubuntu VM

Windows Server 2019 VM




**Step 2:
Deploy**

Other Devices
(iOS, etc)

Native
support

Converters

A baker in a white shirt and apron is shown from the waist down, working on a wooden table. The table is covered with a layer of white flour. A large, round loaf of bread is being shaped on the table. The background is a plain, light-colored wall.

Baker vs Starting a Bakery

Create

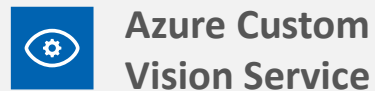
Frameworks



Native support

Converters

Services



Native support

ONNX Model

Deploy

Azure

Azure Machine Learning services

Ubuntu VM

Windows Server 2019 VM

Windows/Linux Devices

IoT Edge Devices

Other Devices
(iOS, etc)

Native support

Converters

A person's hands are visible, holding a large, round, rustic loaf of bread. The bread has a thick, golden-brown crust with some darker, caramelized spots. It is wrapped in a blue and white striped cloth. The background is a blurred wooden surface.

Cloud or Edge

Deploy in Cloud

- Model management services
- Deploy as web service
- Capture model telemetry



Azure
Machine Learning

Machine Learning

Typical E2E Process

Prepare



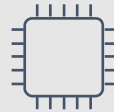
Prepare
Data



Experiment



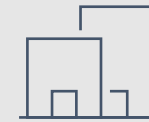
Build model
(your favorite IDE)



Train &
Test Model



Register and
Manage Model



Build
Image



Deploy Service
Monitor Model

Deploy





ONNX

ONNX Docker Image

[onnx-base](#): Use published ONNX package from PyPi with minimal dependencies.

[onnx-dev](#): Build ONNX from source with minimal dependencies.

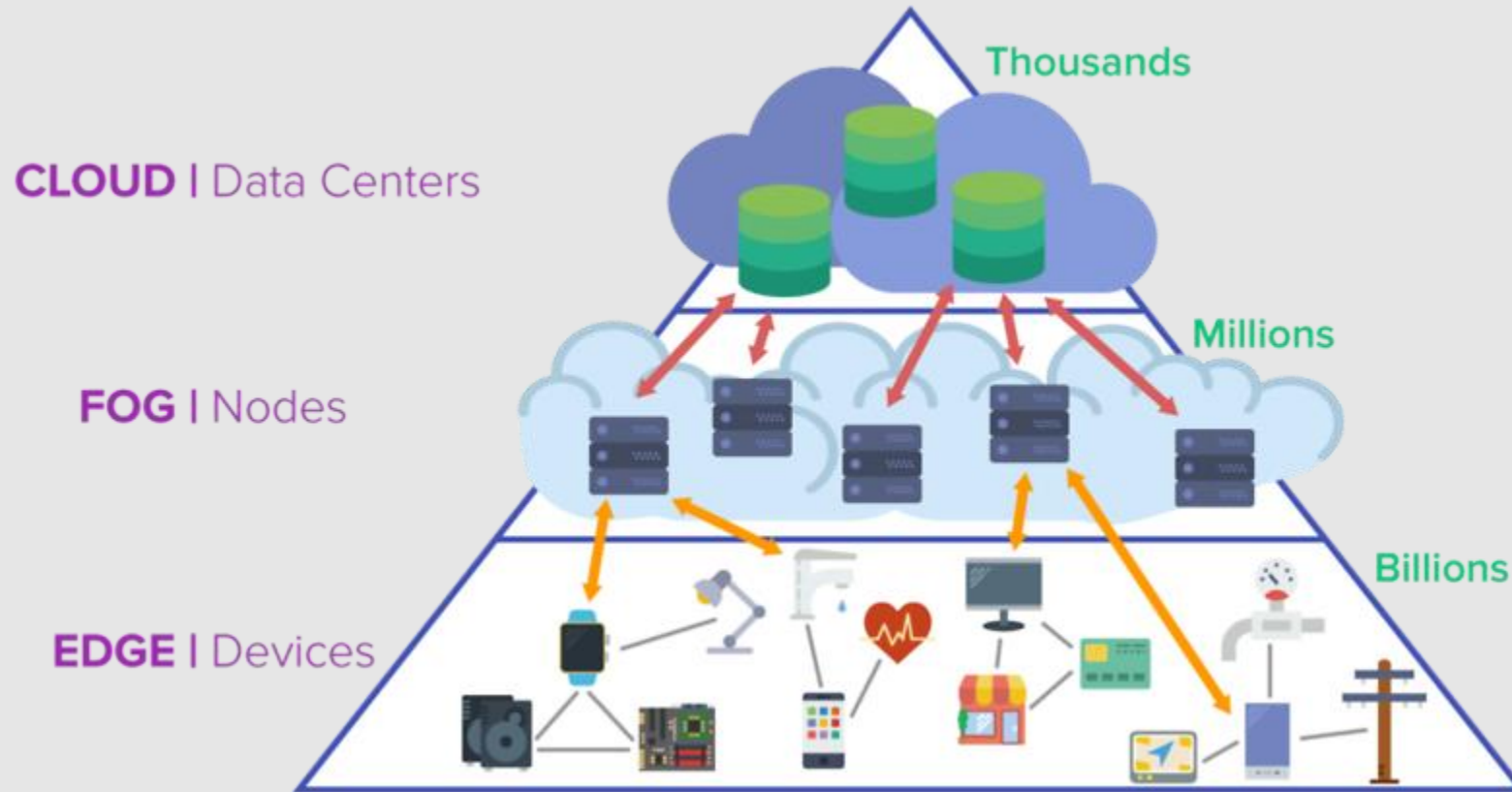
[onnx-ecosystem](#): Jupyter notebook environment

- getting started quickly with ONNX models
- ONNX converters
- inference using ONNX Runtime.

Caffe2/PyTorch Docker

```
docker run -it --rm onnx/onnx-docker:cpu /bin/bash
```

What is the Edge?

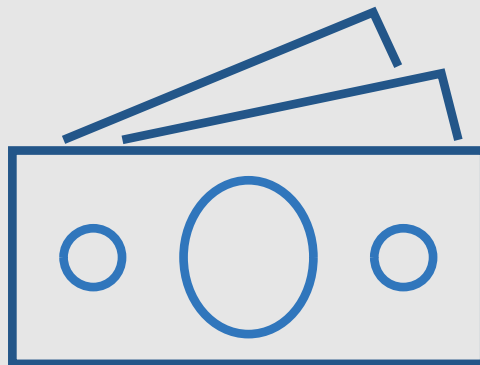


[Imagimob AB](#)

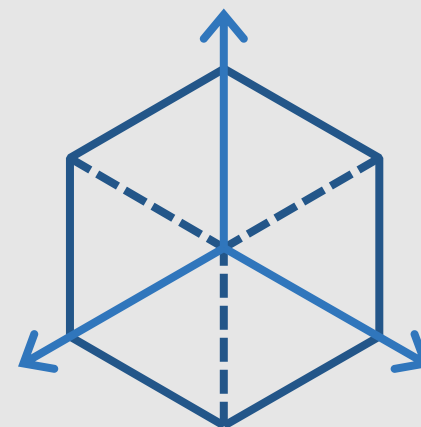
AI on the edge



Low latency



Scalability



Flexibility

ONNX Runtime

- High performance inference engine for ONNX models
- Founded and Open Sourced by Microsoft under MIT License
- Supports full ONNX-ML spec
- Extensible architecture to plug-in hardware accelerators
- Ships with Windows 10 as WinML
- onnxruntime.ai



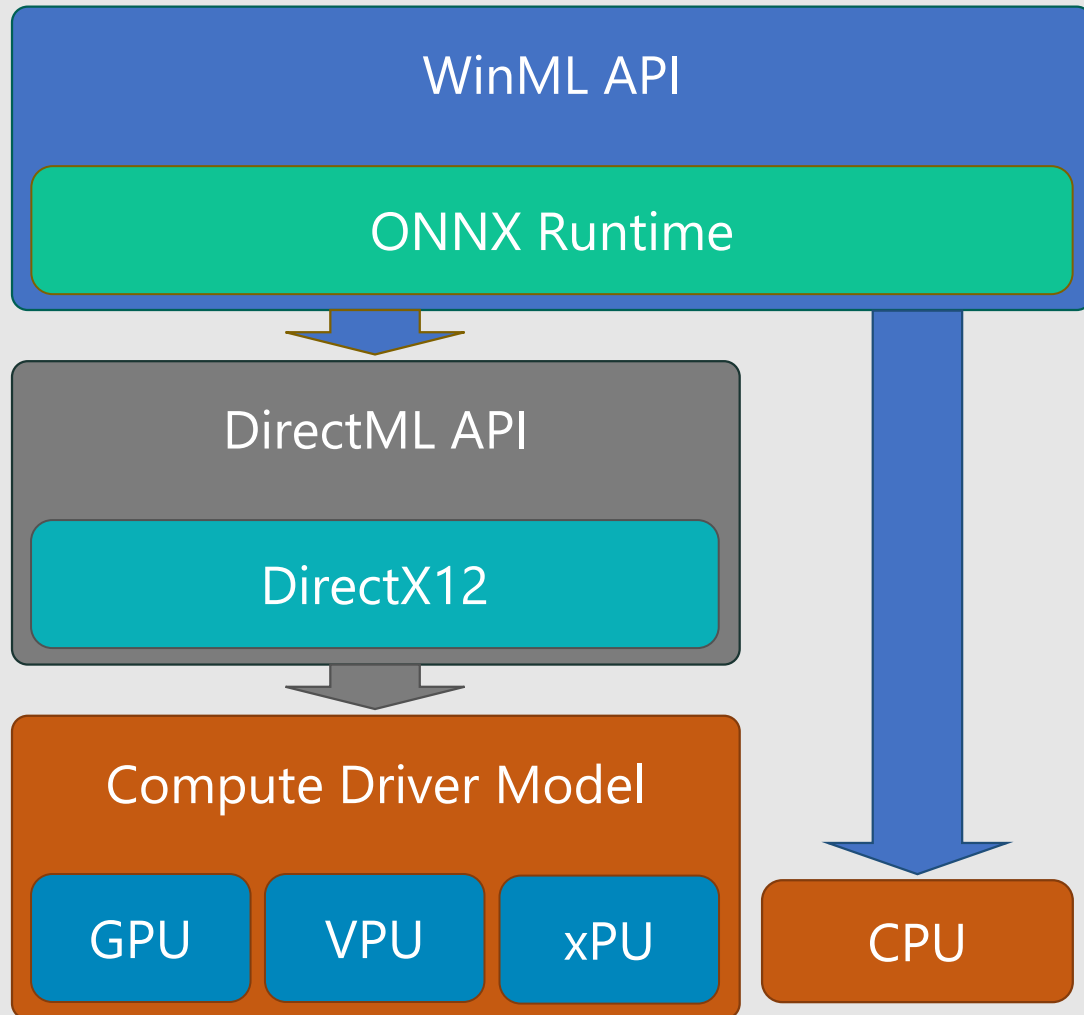
ONNX

ONNX Runtime

Get Started Easily

Optimize Inferencing	Optimize Training							
Platform	Windows	Linux	Mac	Android	iOS	Web Browser (Preview)		
API	Python	C++	C#	C	Java	JS	Obj-C	WinRT
Architecture	X64	X86	ARM64	ARM32	IBM Power			
Hardware Acceleration	Default CPU	CUDA	DirectML	oneDNN	OpenVINO			
	TensorRT	NNAPI	ACL (Preview)	ArmNN (Preview)	CoreML (Preview)			
	MIGraphX (Preview)	NUPHAR (Preview)	Rockchip NPU (Preview)	Vitis AI (Preview)				
Installation Instructions	Install Nuget package Microsoft.ML.OnnxRuntime.Gpu Refer to docs for requirements.							

Windows AI platform



- WinML
 - **Practical**, simple model-based API for ML inferencing on Windows
- DirectML
 - **Realtime, high control** ML operator API; part of DirectX family
- Compute Driver Model
 - Robust **hardware reach**/abstraction layer for compute and graphics silicon

ONNX
Runtime
JavaScript

Node.js binding

Web

React Native

ONNX Runtime Node.js

- Node.js binding
- ONNX model inferencing
- Electron
- Uses web assembly

Install

```
# install latest release version  
npm install onnxruntime-node
```

Import

```
// use ES6 style import syntax (recommended)  
import * as ort from 'onnxruntime-node';
```

```
// or use CommonJS style import syntax  
const ort = require('onnxruntime-node');
```

ONNX Runtime Web (ORT-Web)

- JavaScript library for running ONNX models on browsers
- adopted Web Assembly and WebGL technologies
- optimized ONNX model inference runtime for both CPUs and GPUs.

Install

```
# install latest release version  
npm install onnxruntime-web  
  
# install nightly build dev version  
npm install onnxruntime-web@dev
```

Import

```
// use ES6 style import syntax (recommended)  
import * as ort from 'onnxruntime-web';
```

```
// or use CommonJS style import syntax  
const ort = require('onnxruntime-web');
```

Why inference in the browser



It's faster



It's safer and helps with
privacy



It works offline



It's cheaper

Why not in the browser?



**THE MODEL IS TOO LARGE AND
REQUIRES HIGHER HARDWARE SPECS.**



DOWNLOADED ONTO THE DEVICE

React Native

- score pre-trained ONNX models
- ONNX Runtime Mobile
- light-weight inference solution
- Android and iOS

Install

```
# install latest release version  
npm install onnxruntime-react-native
```

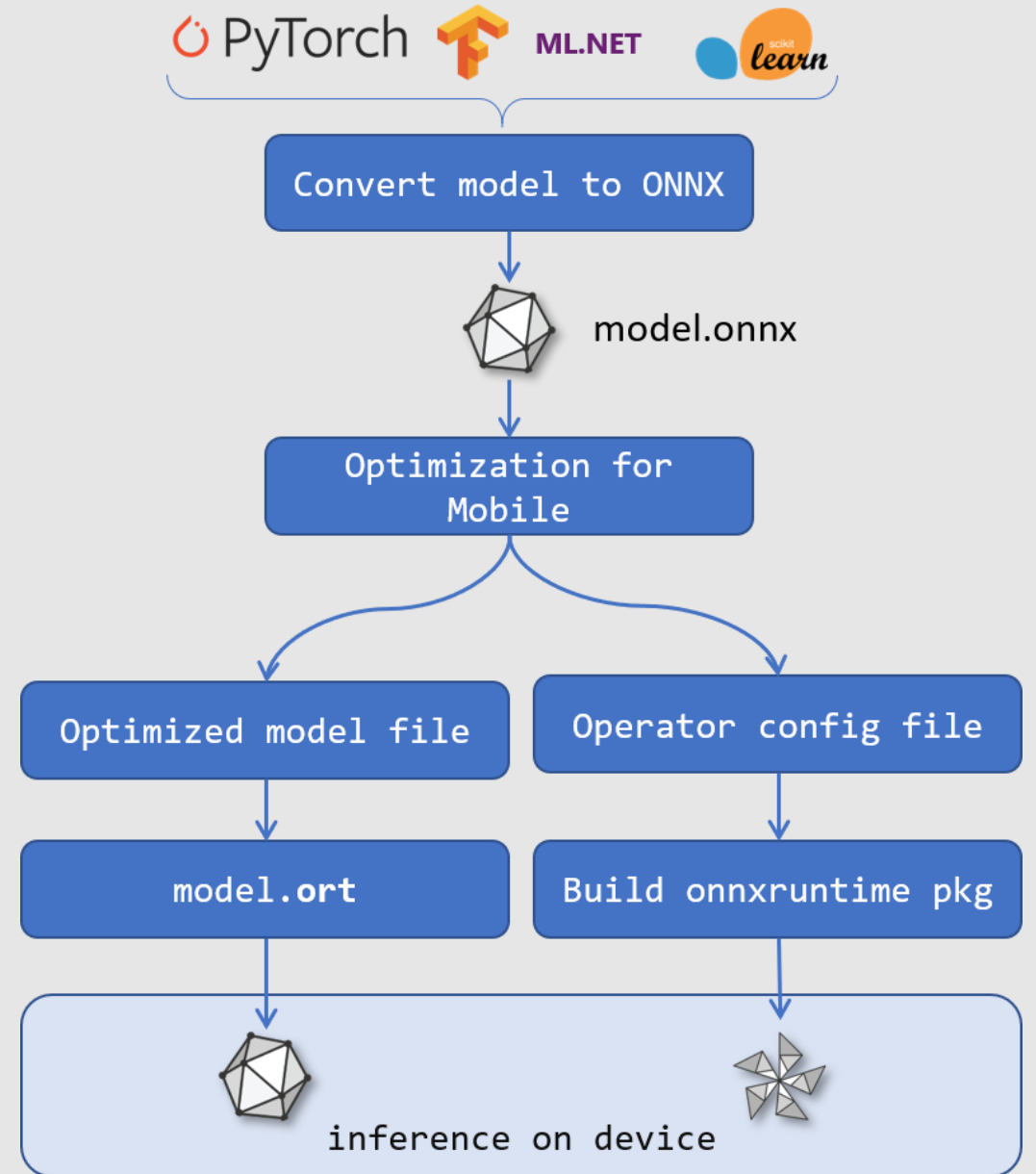
Import

```
// use ES6 style import syntax (recommended)  
import * as ort from 'onnxruntime-react-native';
```

```
// or use CommonJS style import syntax  
const ort = require('onnxruntime-react-native');
```

ONNX Runtime Mobile

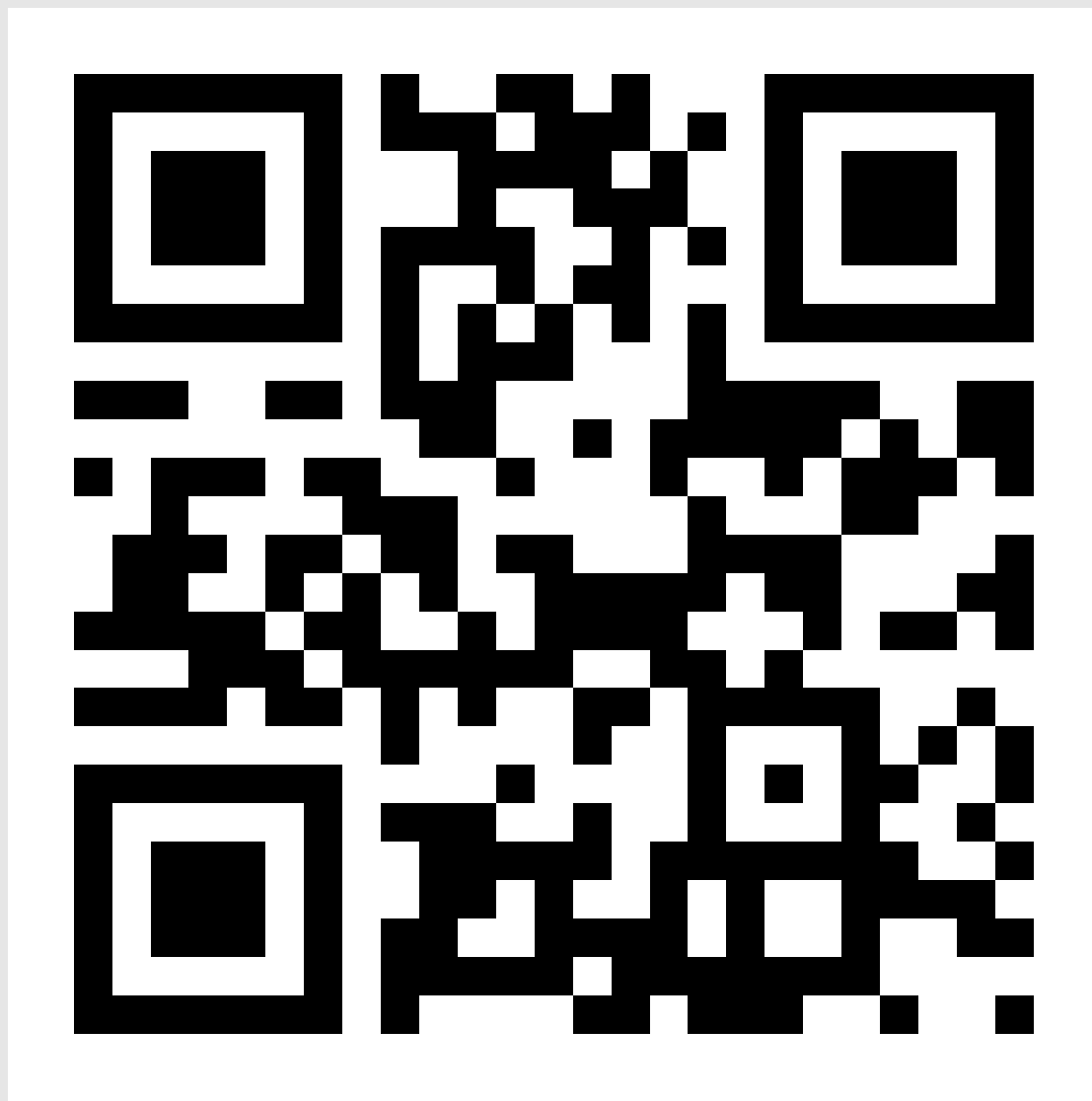
- minimizes the binary size
- pre-optimized ONNX model to an internal format ('ORT format model')



Compatibility Chart

Compatibility

OS/Browser	Chrome	Edge	Safari	Electron	Node.js
Windows 10	wasm, webgl	wasm, webgl	-	wasm, webgl	wasm
macOS	wasm, webgl	wasm, webgl	wasm, webgl	wasm, webgl	wasm
Ubuntu LTS 18.04	wasm, webgl	wasm, webgl	-	wasm, webgl	wasm
iOS	wasm, webgl	wasm, webgl	wasm, webgl	-	-
Android	wasm, webgl	wasm, webgl	-	-	-



<http://bit.ly/ml-onnx>



Recap

- ✓ What is ONNX

ONNX is an open standard so you can use the right tools for the job and be confident your models will run efficiently on your target platforms

- ✓ How to create ONNX models

ONNX models can be created from many frameworks

- ✓ How to deploy ONNX models

ONNX models can be deployed with Windows ML, .NET/Javascript/Python and to the cloud with Azure ML and the high performance ONNX Runtime

About Me

Ron Dagdag



Ron Lyle Dagdag

Immersive Experience Developer

Cell: 682-560-3988

ron@dagdag.net



Experience AR

www.dagdag.net

@rondagdag

http://ron.dagdag.net

Lead Software Engineer at Spacee

5th year Microsoft MVP awardee

Personal Projects
www.dagdag.net

Email: ron@dagdag.net
Twitter [@rondagdag](https://twitter.com/rondagdag)

Connect me via Linked In
www.linkedin.com/in/rondagdag/


Thanks for geeking out with me about ONNX

<https://linktr.ee/rondagdag>

Hackster Portfolio

www.dagdag.net

@rondagdag



Ron Dagdag
Dad / Lead Software Engineer / 3D Developer / Tax Return Preparer.
Passionate to learn about Robotics, VR, AR, Artificial Intelligence, IOT
@rondagdag
FORT WORTH, United States
Team [Augmented Reality](#)
Team [Virtual Reality](#)

INTERNET OF "KINECT" 2,443 16 44
IoT Gateway Azure Event Hub Stream Analytics Production Base
Posture Recognition using K...
Ron Dagdag

Easy 60 0
Littlebits Arduino Keyboard ...
Ron Dagdag

Intermediate 701 9
Alexa, tell Echobot to fly
Ron Dagdag

Advanced 1,345 12
Control your "Earth Rover" i...
Ron Dagdag

Advanced 2,256 30
ConstructAR - The Holograp...
TEAM ConstructAR

Intermediate 449 4
Color Changing Fireworks in...
Ron Dagdag