#RonDagdag

Making neural networks run in browser with ONNX

RON DAGDAG

Director of Software Engineering and Microsoft MVP

@RonDagdag

Neural Network?

Machine Learning?

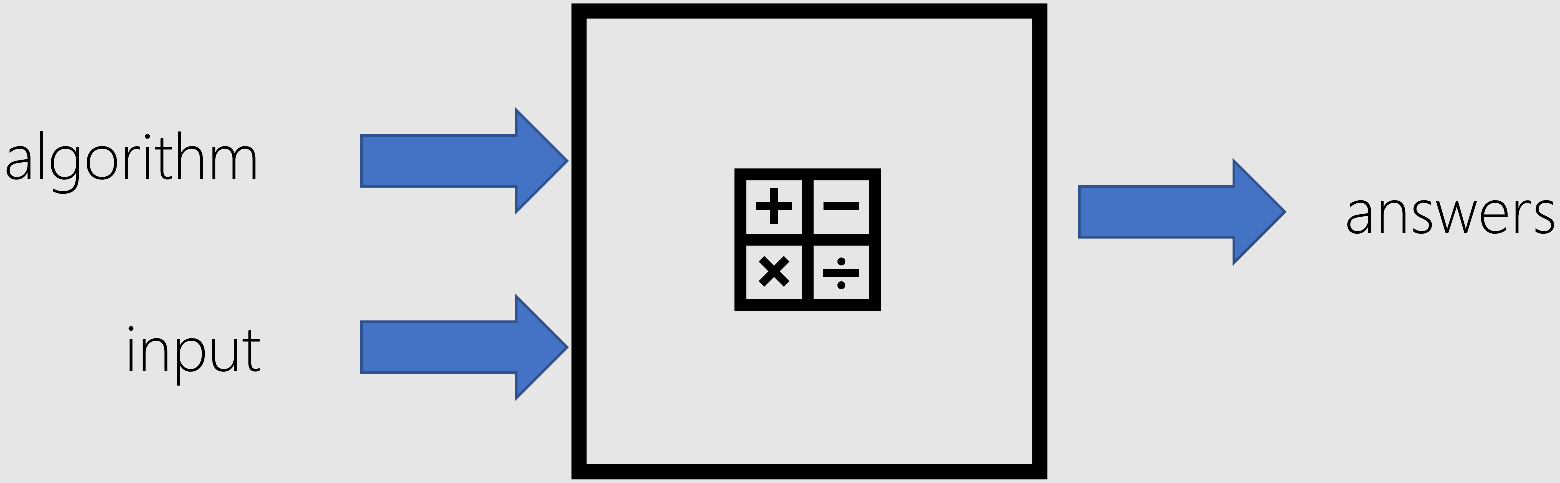
Web Development?

Pokemon Fans?

r CC BY-

ONNX
Not ONIX
Not ONYX

programming

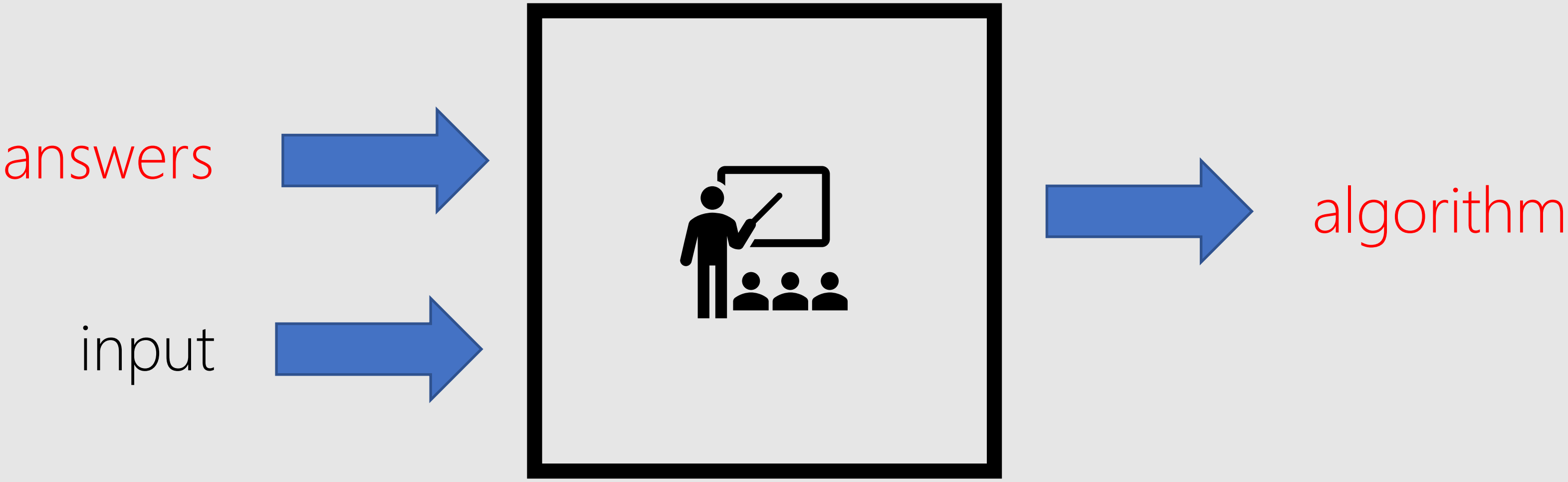


algorithm

input

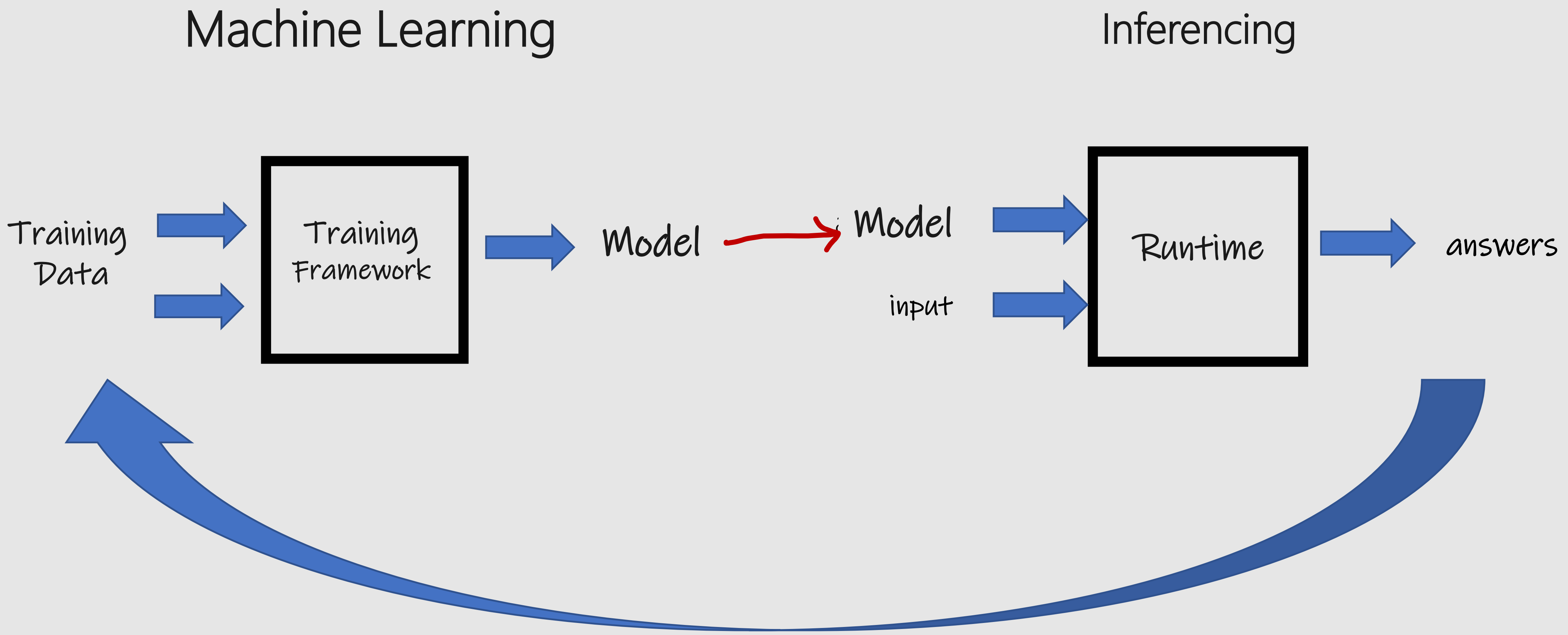
answers

machine learning

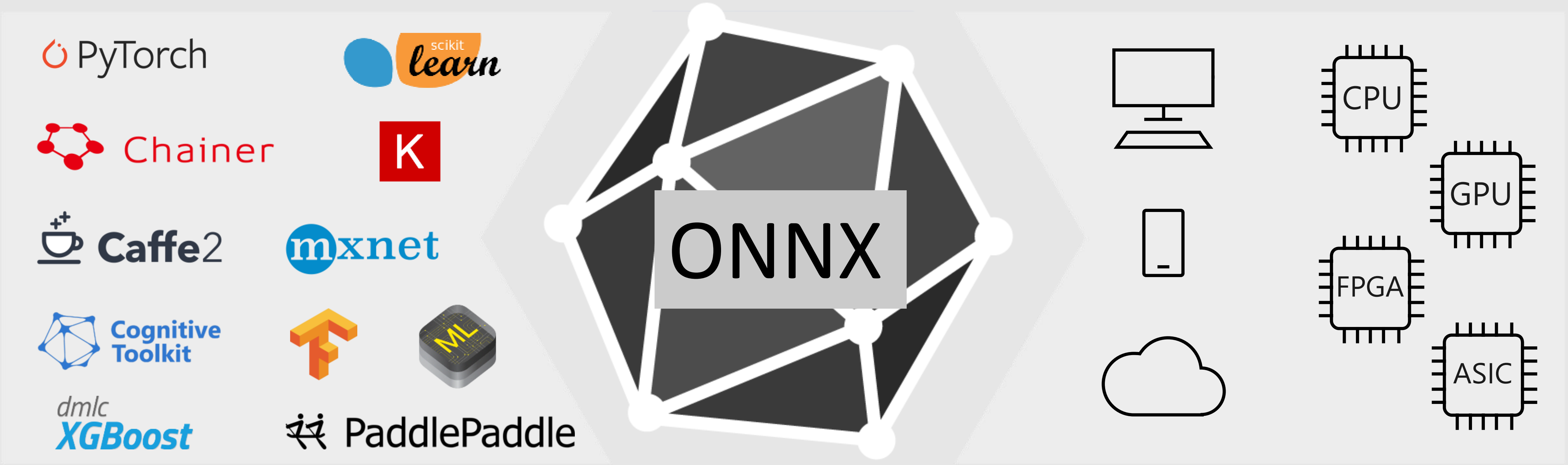


@rondagdag

ML Primer


Machine Learning
Inferencing
Training Data
Training Framework
Model
Model
input
Runtime
answers

Open and Interoperable AI



ONNX

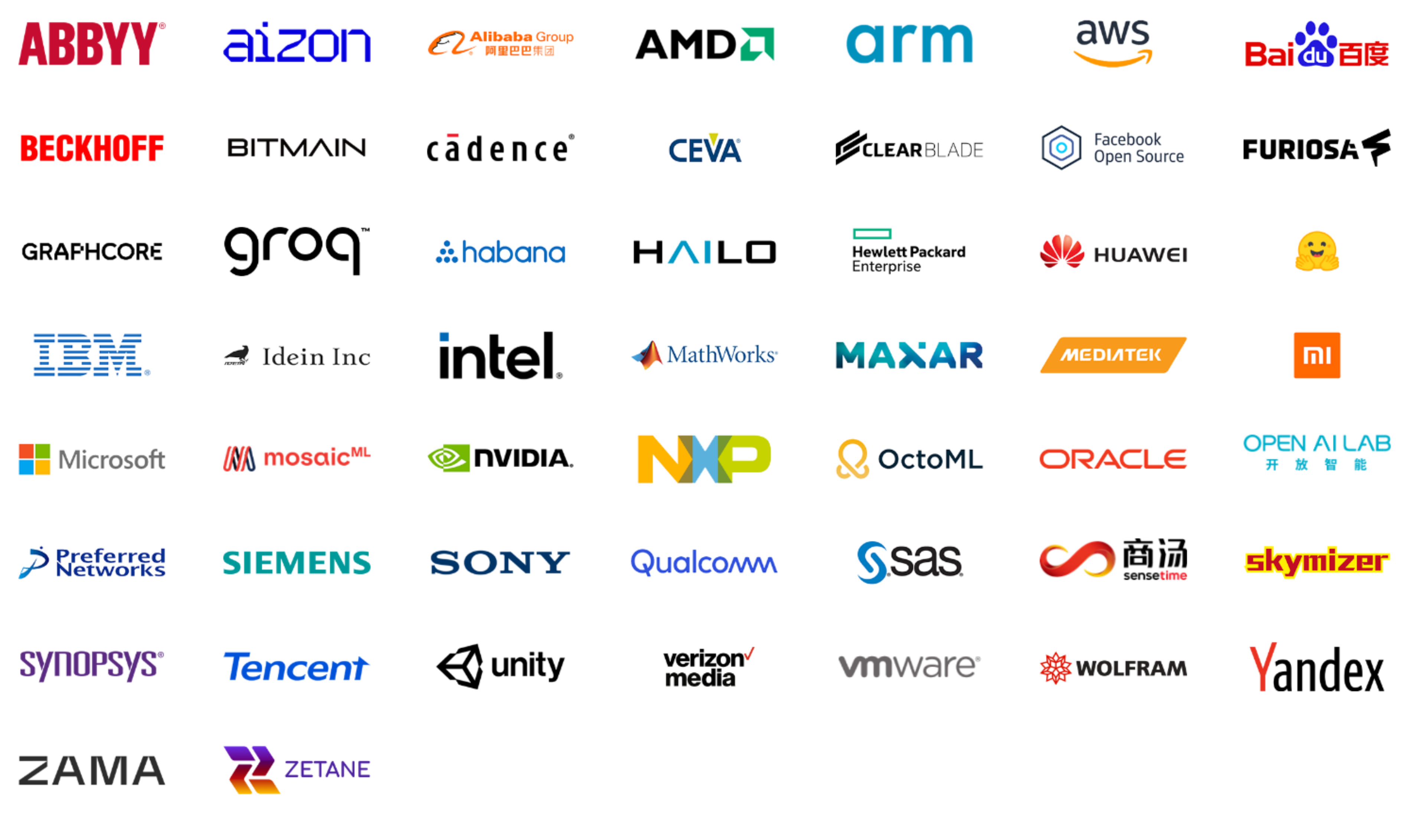
Open Neural Network Exchange

Open format for ML models

github.com/onnx

onnx.ai/

ONNX Partners

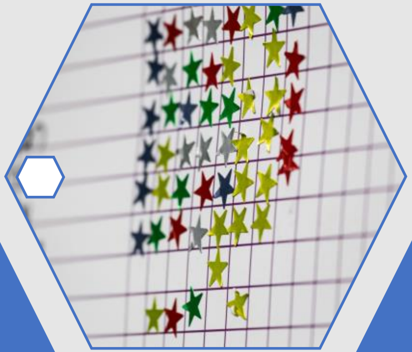


@rondagdag

13.4k
Github Stars

2300+

Pull
Requests



240+
Contributors

3.1k Github
Forks

40+ Model
Zoo

LF AI
GRADUATE
PROJECT

@rondagdag

When to use ONNX?

- Trained in Python or ML.NET - deploy into a C#/Java/Javascript app
- PyTorch docker image > 3.6 GB – need something lighter
- 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)

Agenda

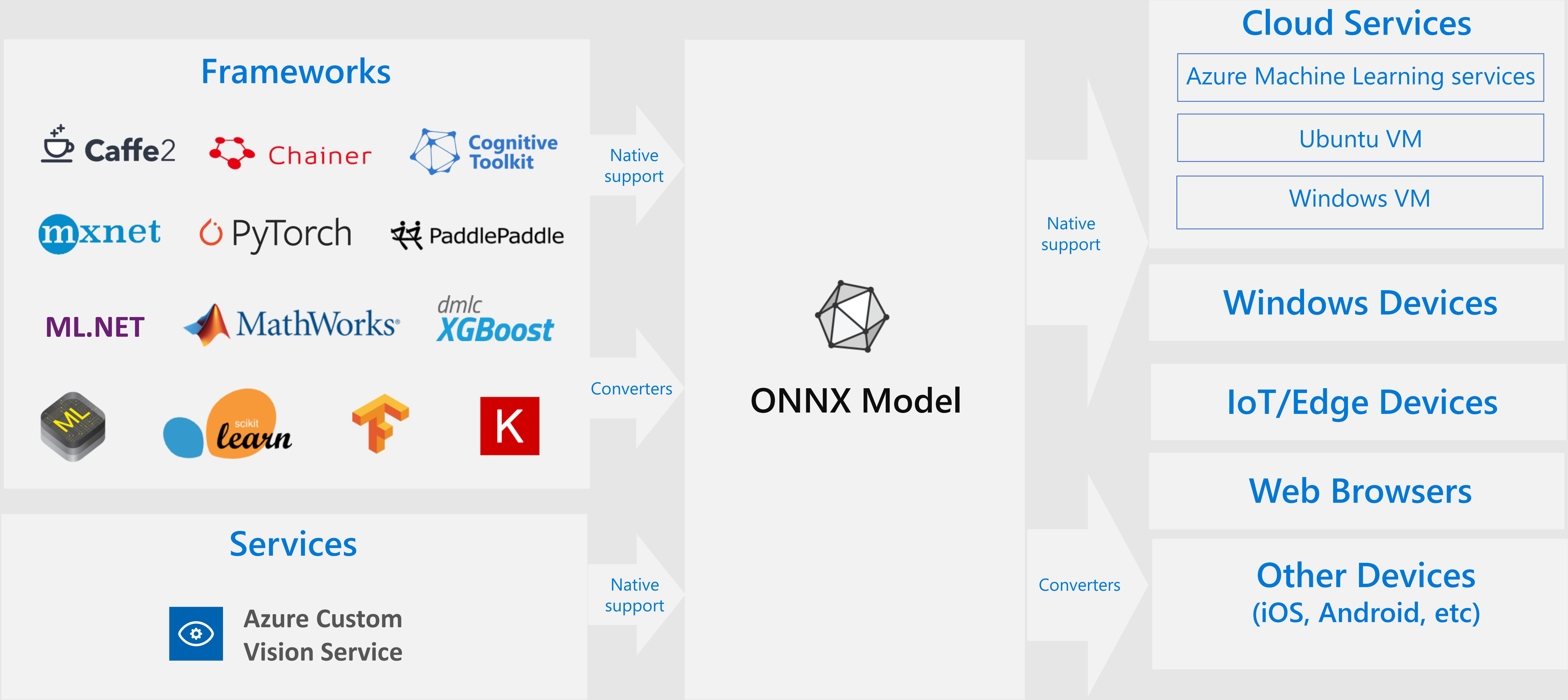
✓ What is ONNX, When to use ONNX

❑ How to create ONNX models

❑ How to deploy ONNX models

Create

Deploy



Frameworks

Caffe2

Chainer

Cognitive Toolkit

mxnet

PyTorch

PaddlePaddle

ML.NET

MathWorks

dmlc XGBoost

ML

scikit learn

K

Services

Azure Custom Vision Service

Cloud Services

Azure Machine Learning services

Ubuntu VM

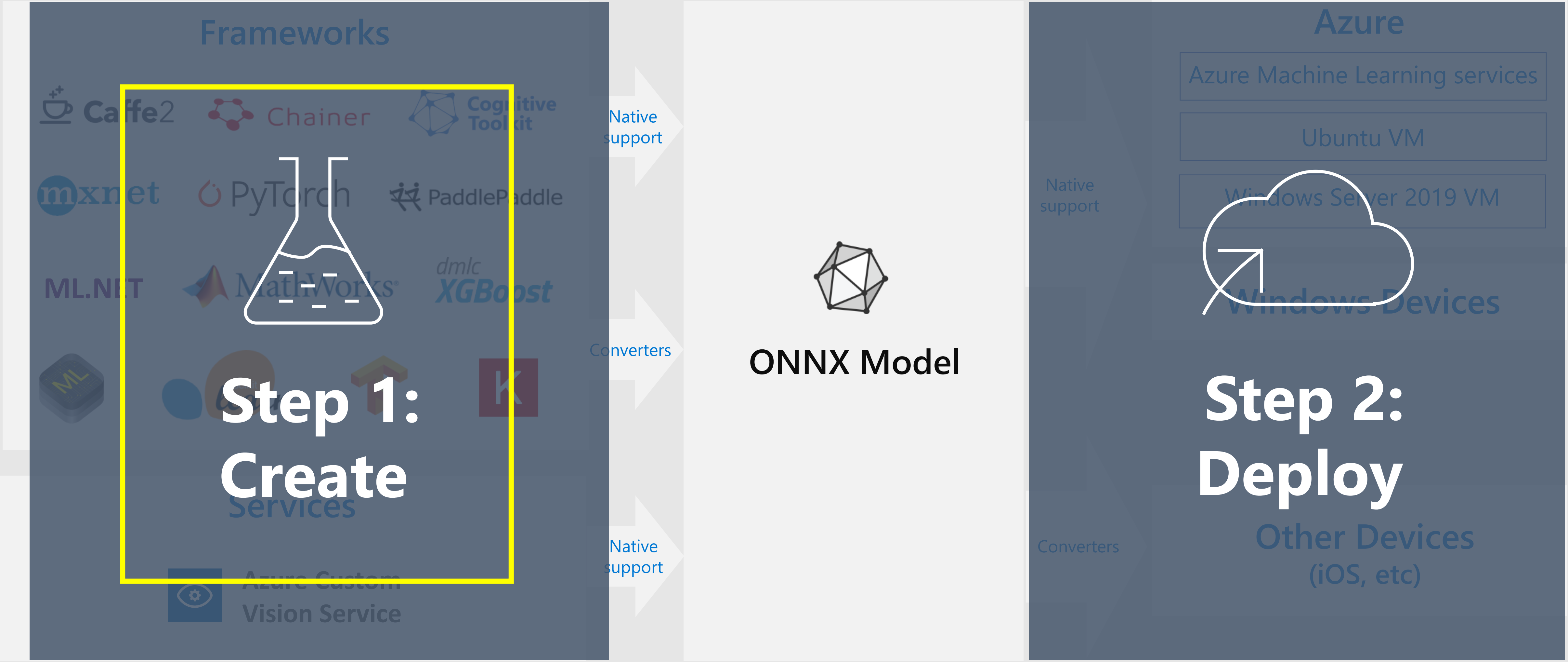
Windows VM

Windows Devices

IoT/Edge Devices

Web Browsers

Other Devices (iOS, Android, etc)



Step 1: Create

ONNX Model

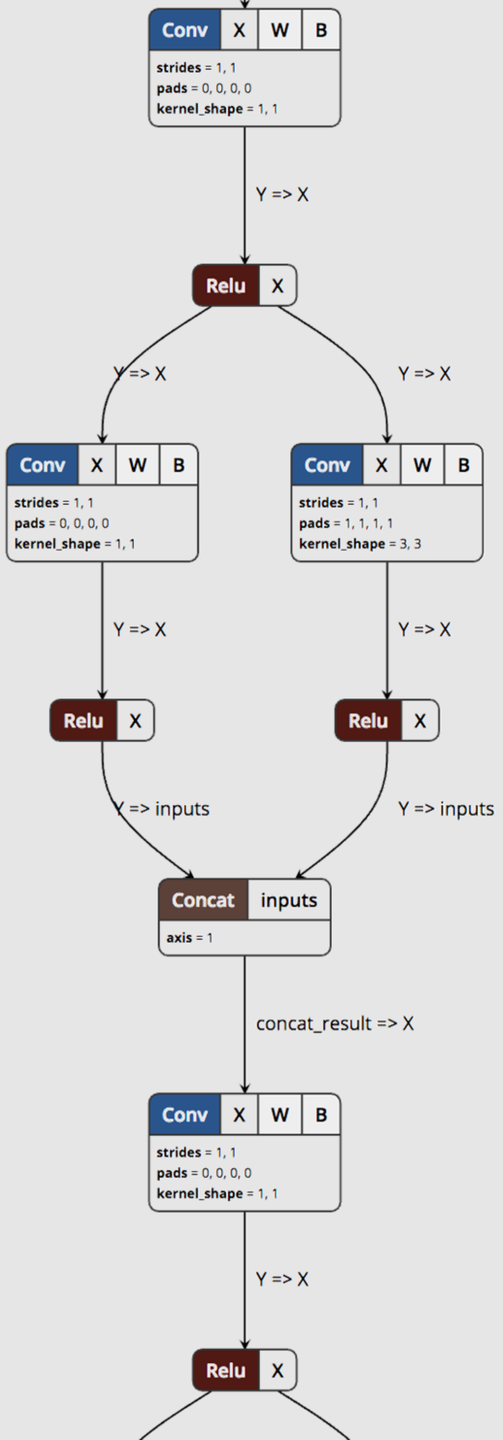
Step 2: Deploy

@rondagdag

Secret Recipe

ONNX Models

Graph of operations



Netron

https://netron.app/

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

3 ways to get an ONNX model

- ONNX Model Zoo
- Azure Custom Vision Service
- Convert existing models

ONNX Model Zoo: github.com/onnx/models

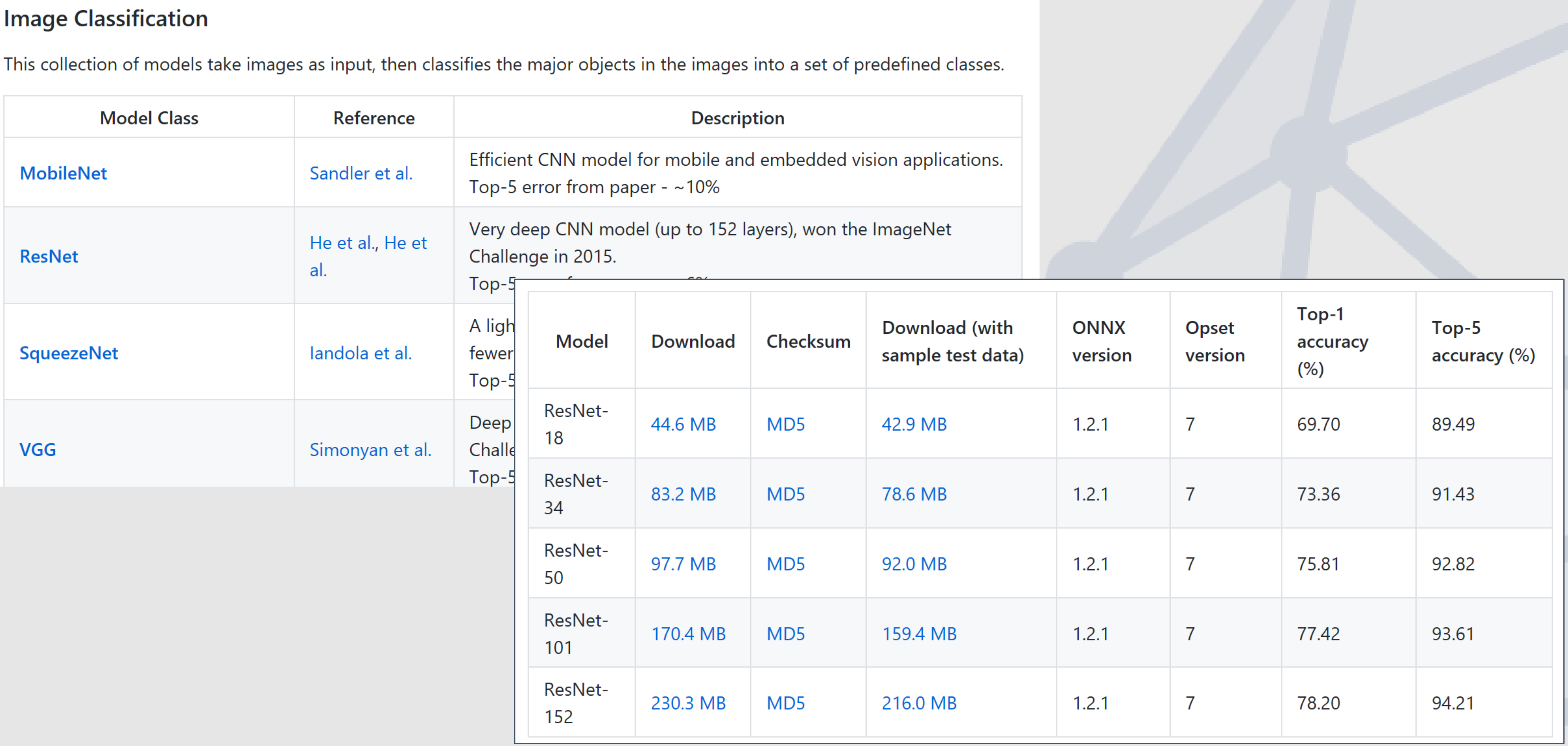


Image Classification

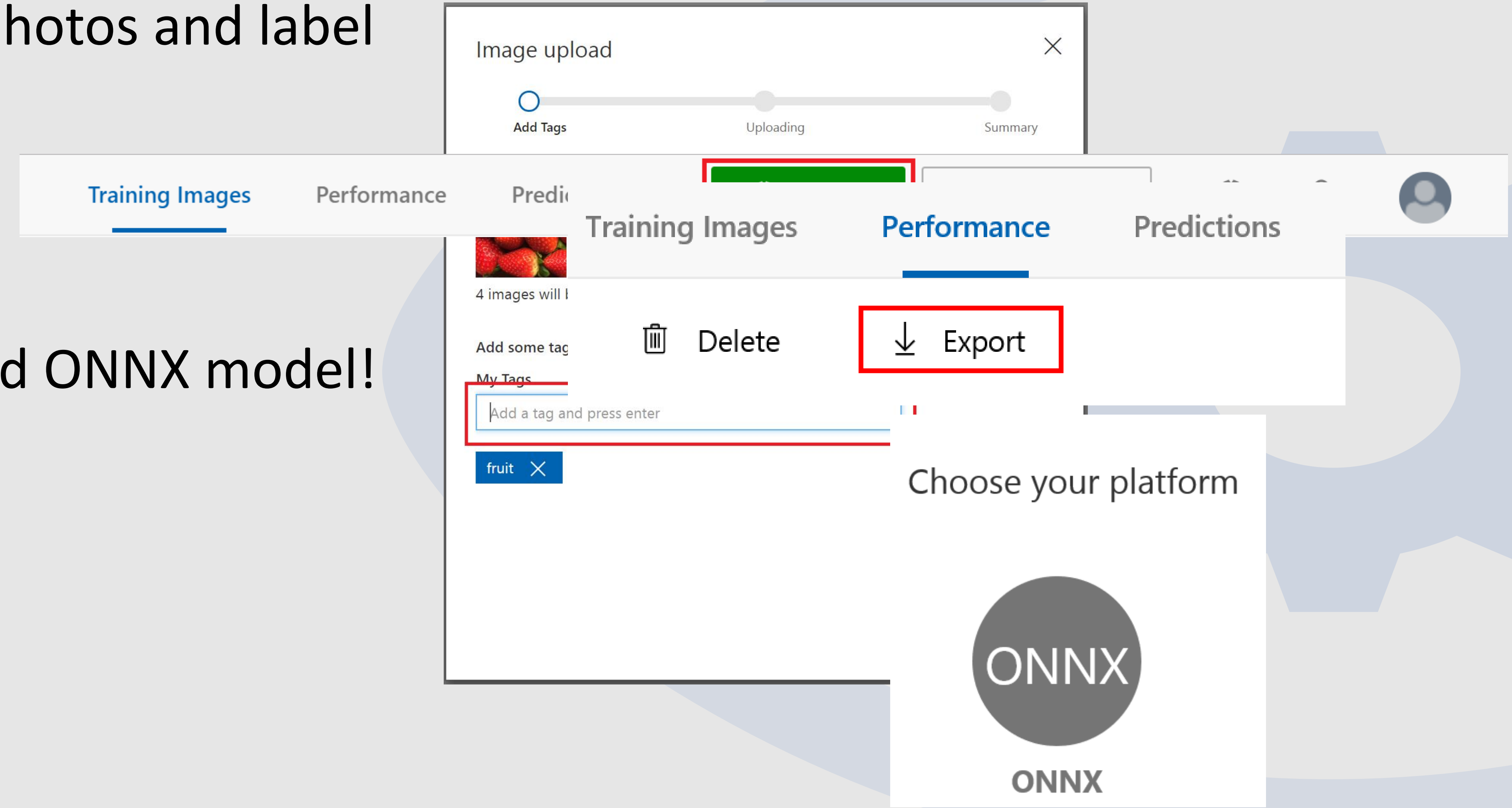
This collection of models take images as input, then classifies the major objects in the images into a set of predefined classes.

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
SqueezeNet	Iandola et al.	A ligh fewer Top-5
VGG	Simonyan et al.	Deep Challe Top-5

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

Caffe2

Yandex CatBoost

Chainer

Cognitive Toolkit

CoreML

Keras

LibSVM

MATLAB

[M]^{S}MindSpore

mxnet

MyCaffe™

NCNN

NeoML

Neural Network Libraries

飞桨 PaddlePaddle

PaddlePaddle

PyTorch

sas

SIEMENS

Apache SINGA

scikit learn

SciKit Learn

Tengine

TensorFlow

dmlc XGBoost

WOLFRAM

Convert models

1. Load existing model

2. (Convert to ONNX)

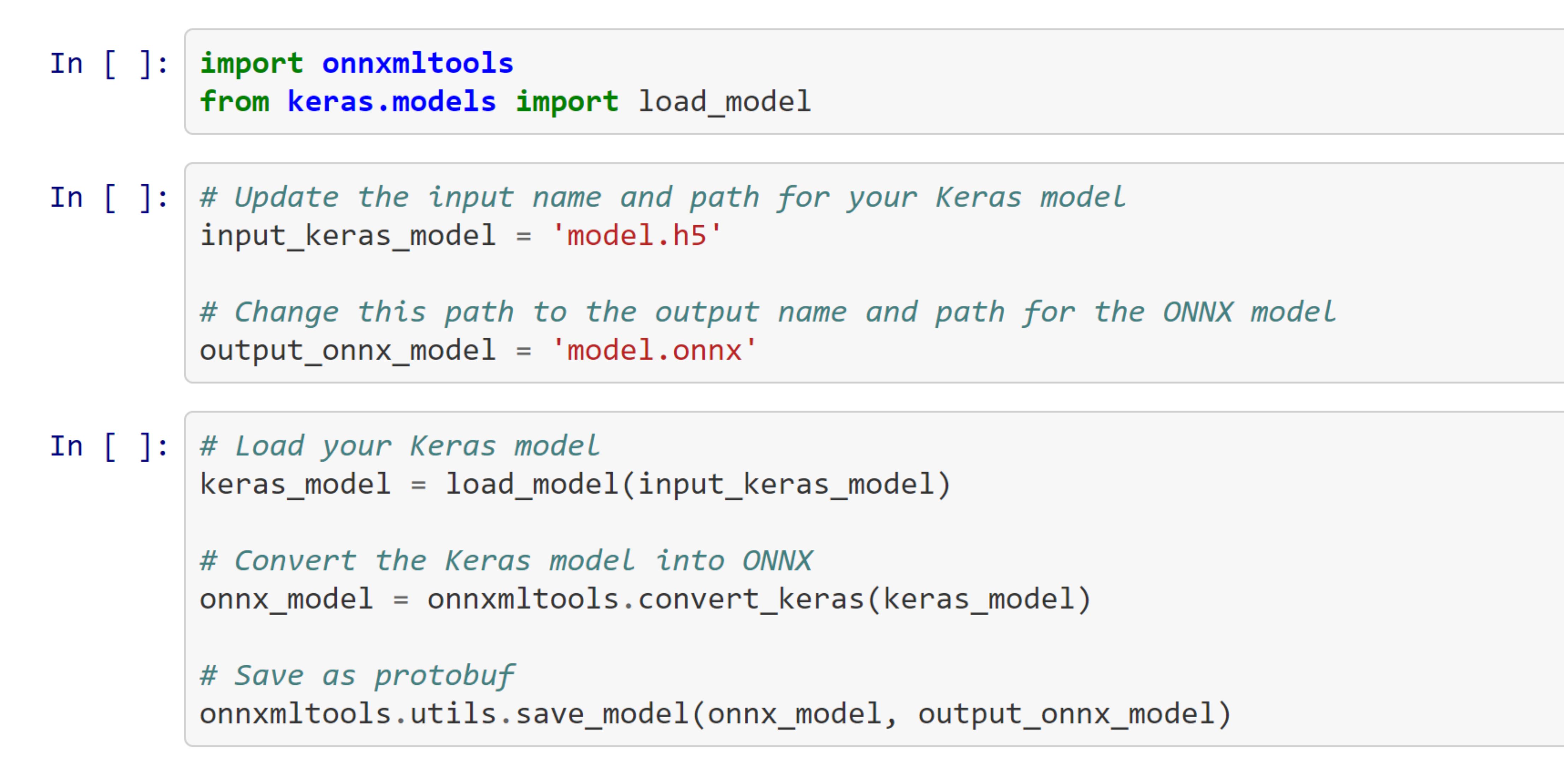
3. Save ONNX model

https://github.com/onnx/tutorials

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 [ ]:

```python
import onnxmltools
from keras.models import load_model
```

In [ ]:

```python
# 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 [ ]:

```python
# 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:

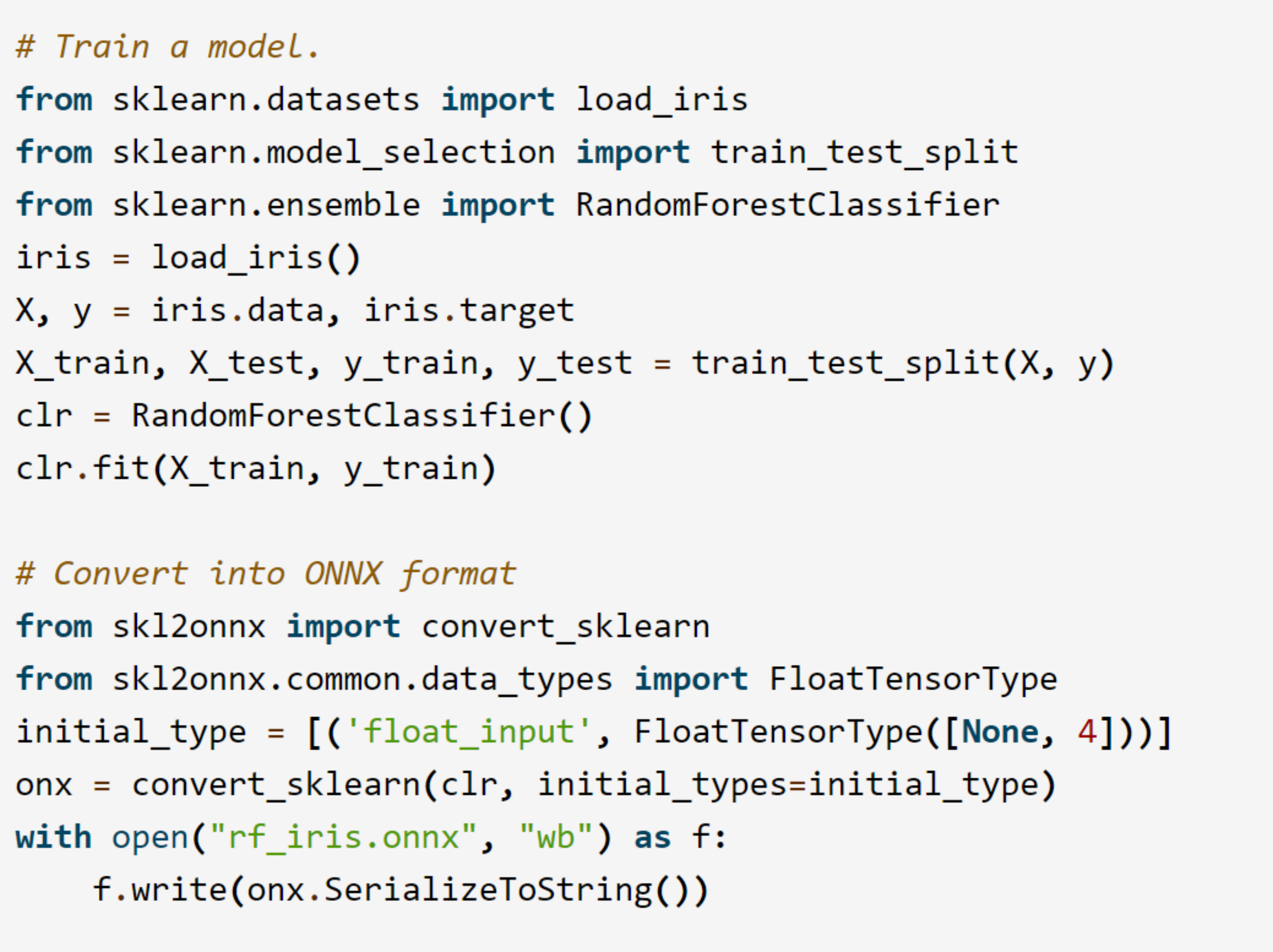
TensorFlow

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

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

Convert models:

scikit learn



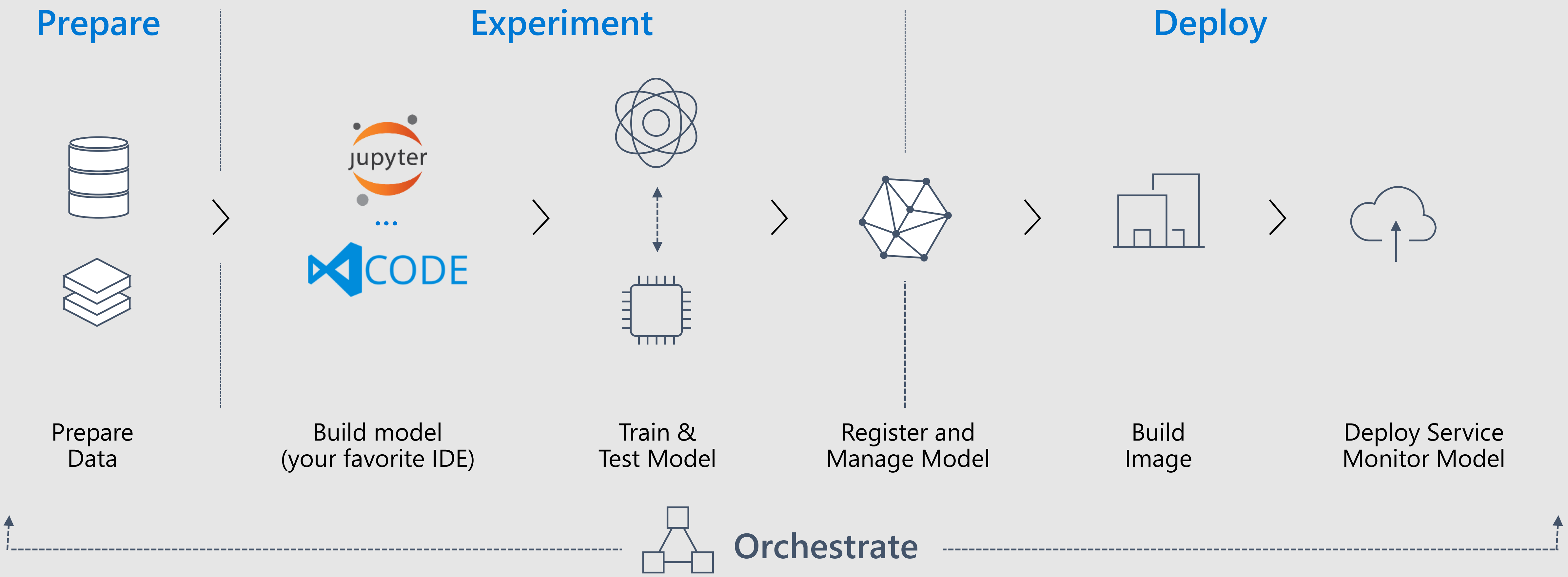
```
# Train a model.
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
iris = load_iris()
X, y = iris.data, iris.target
X_train, X_test, y_train, y_test = train_test_split(X, y)
clr = RandomForestClassifier()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

https://github.com/onnx/tutorials

Baker
vs
Starting a Bakery

Machine Learning

Typical E2E Process



Prepare

Experiment

Deploy

Prepare
Data

Build model
(your favorite IDE)

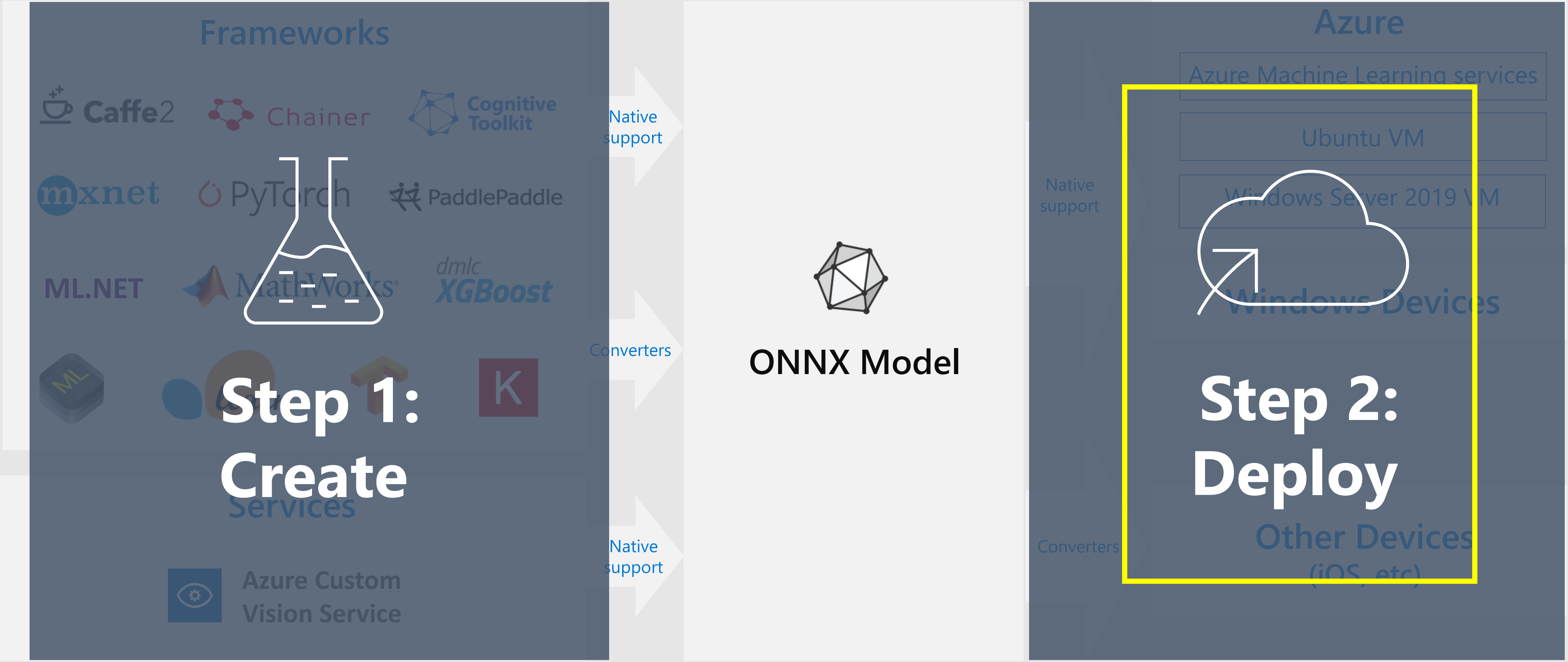
Train &
Test Model

Register and
Manage Model

Build
Image

Deploy Service
Monitor Model

Orchestrate



Frameworks

Caffe2 Chainer Cognitive Toolkit

mxnet PyTorch PaddlePaddle

ML.NET MathWorks dmlc XGBoost

K

Services

Azure Custom Vision Service

Native support

Converters

Native support

ONNX Model

Azure

Azure Machine Learning services

Ubuntu VM

Windows Server 2019 VM

Native support

Windows Devices

Converters

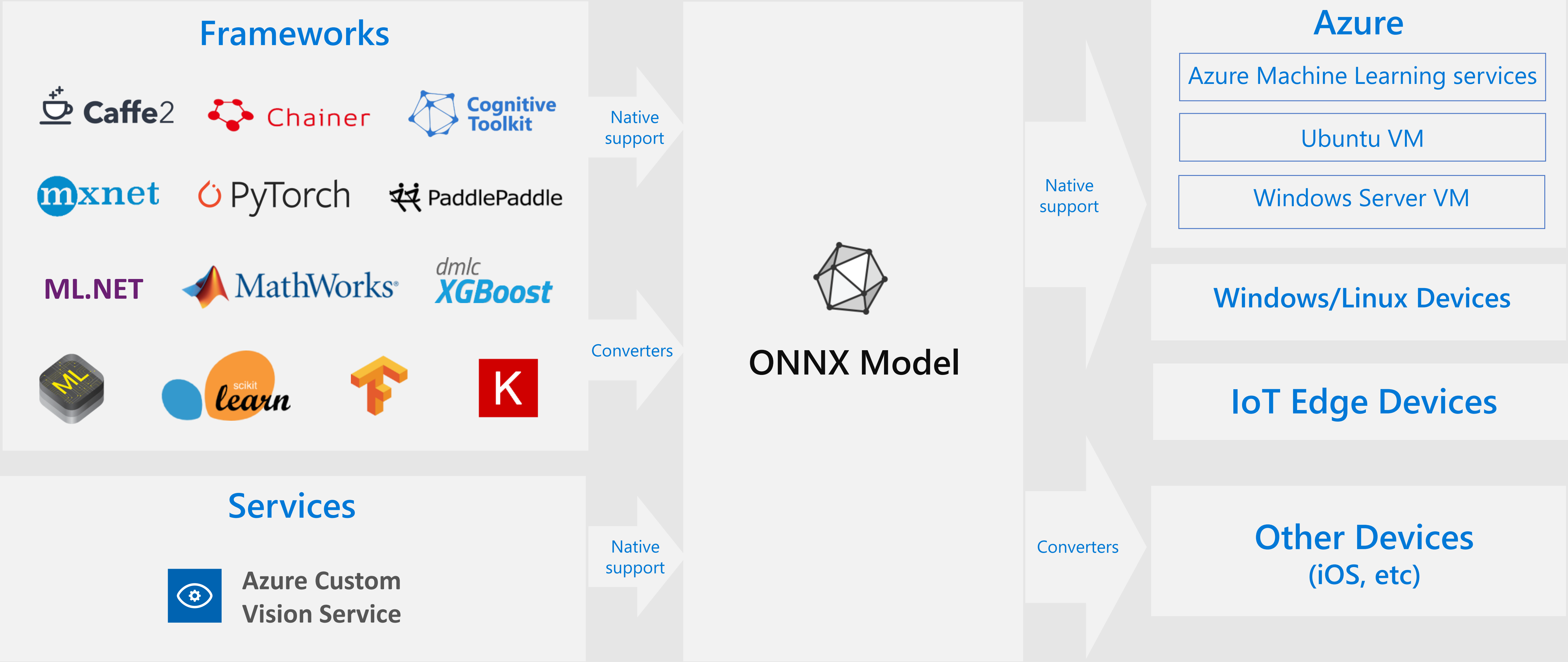
Other Devices (iOS, etc)

Step 1: Create

Step 2: Deploy

Create

Deploy



Frameworks

Services

Azure Custom Vision Service

Native support

Converters

Native support

ONNX Model

Native support

Converters

Azure

Azure Machine Learning services

Ubuntu VM

Windows Server VM

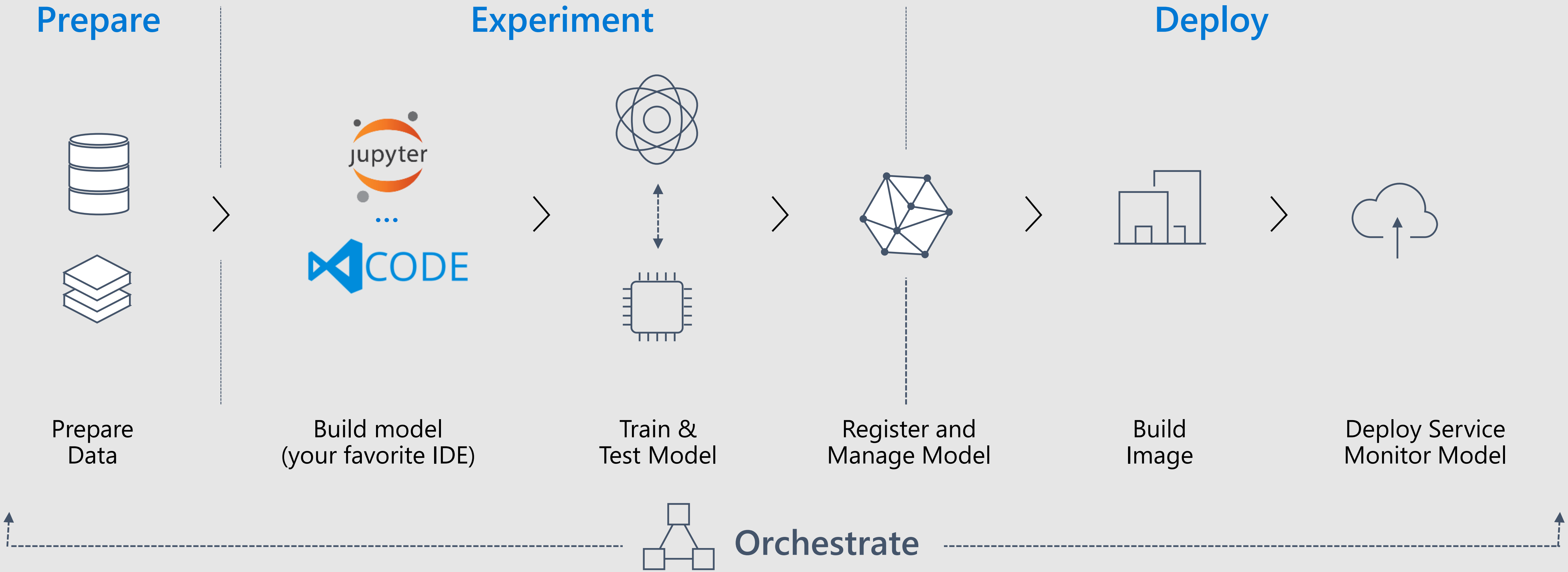
Windows/Linux Devices

IoT Edge Devices

Other Devices (iOS, etc)

Machine Learning

Typical E2E Process



Prepare

Experiment

Deploy

Prepare
Data

Build model
(your favorite IDE)

Train &
Test Model

Register and
Manage Model

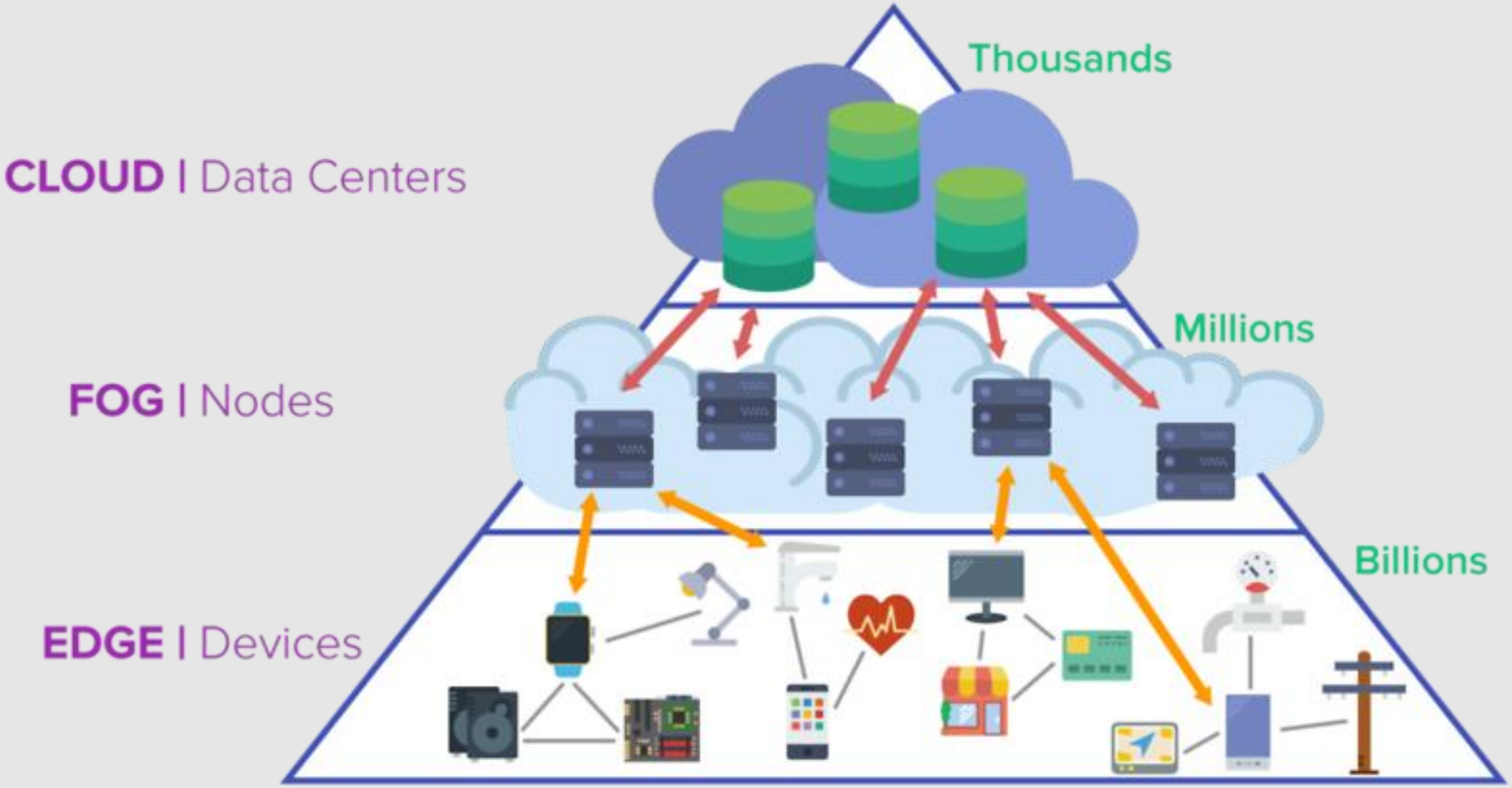
Build
Image

Deploy Service
Monitor Model

Orchestrate

Cloud
or
Edge

What is the Edge?



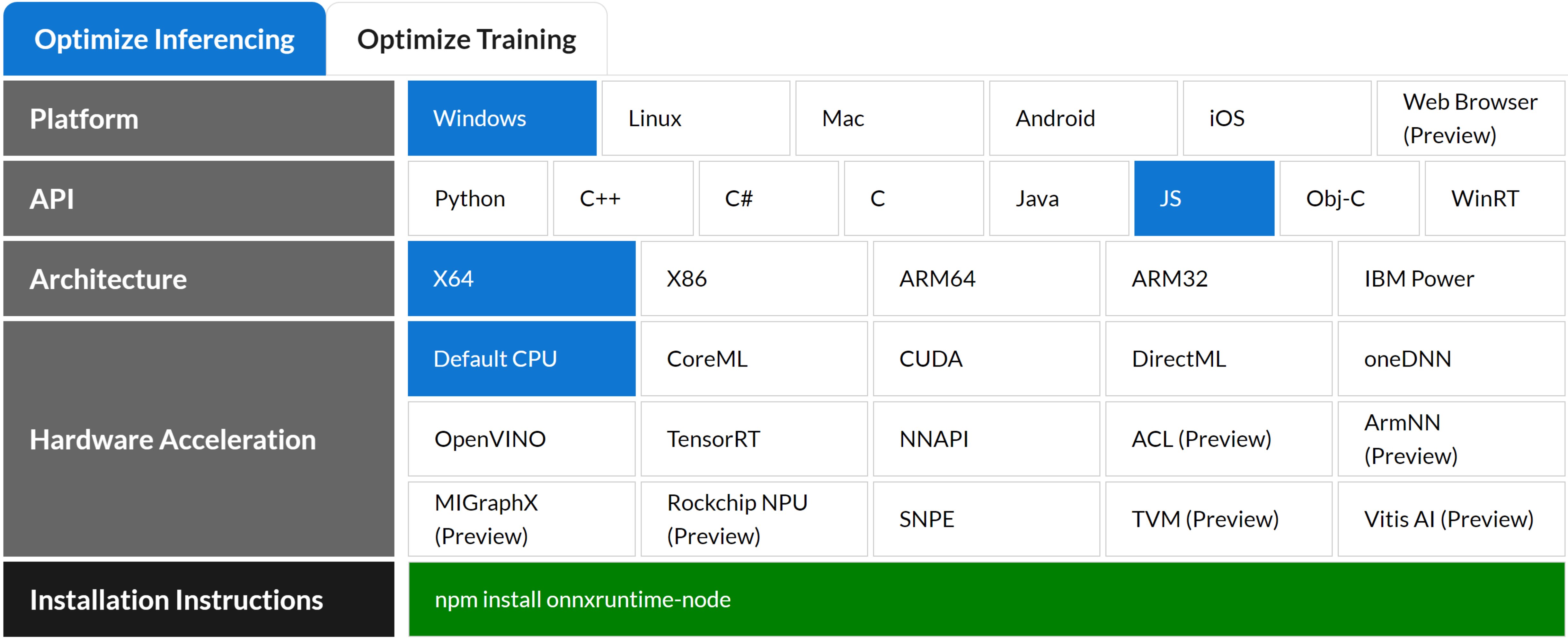
Imagimob AB

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 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	CoreML	CUDA	DirectML	oneDNN			
	OpenVINO	TensorRT	NNAPI	ACL (Preview)	ArmNN (Preview)			
	MIGraphX (Preview)	Rockchip NPU (Preview)	SNPE	TVM (Preview)	Vitis AI (Preview)			
Installation Instructions	npm install onnxruntime-node							

ONNX Runtime JavaScript

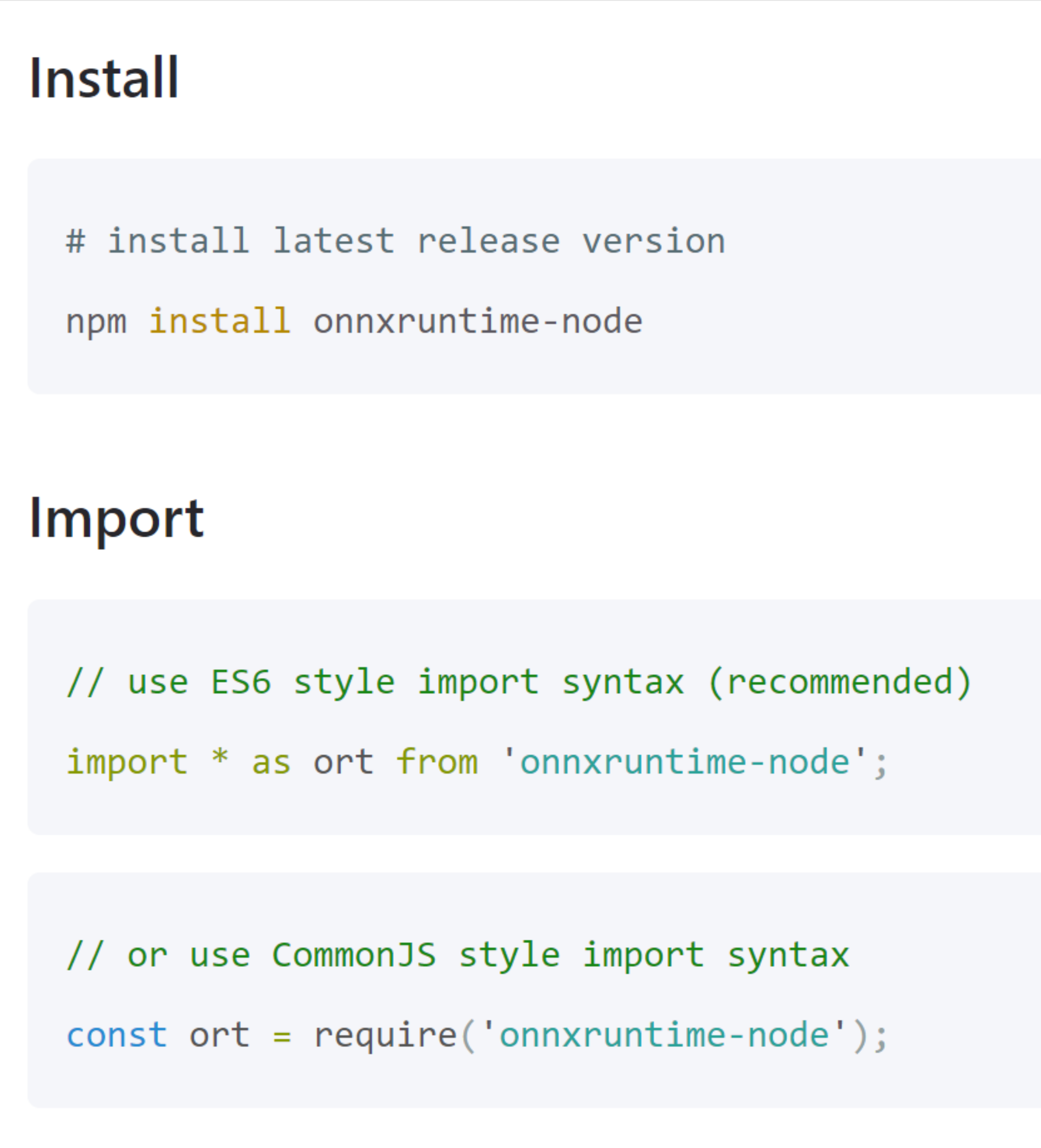
Node.js binding

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';
```

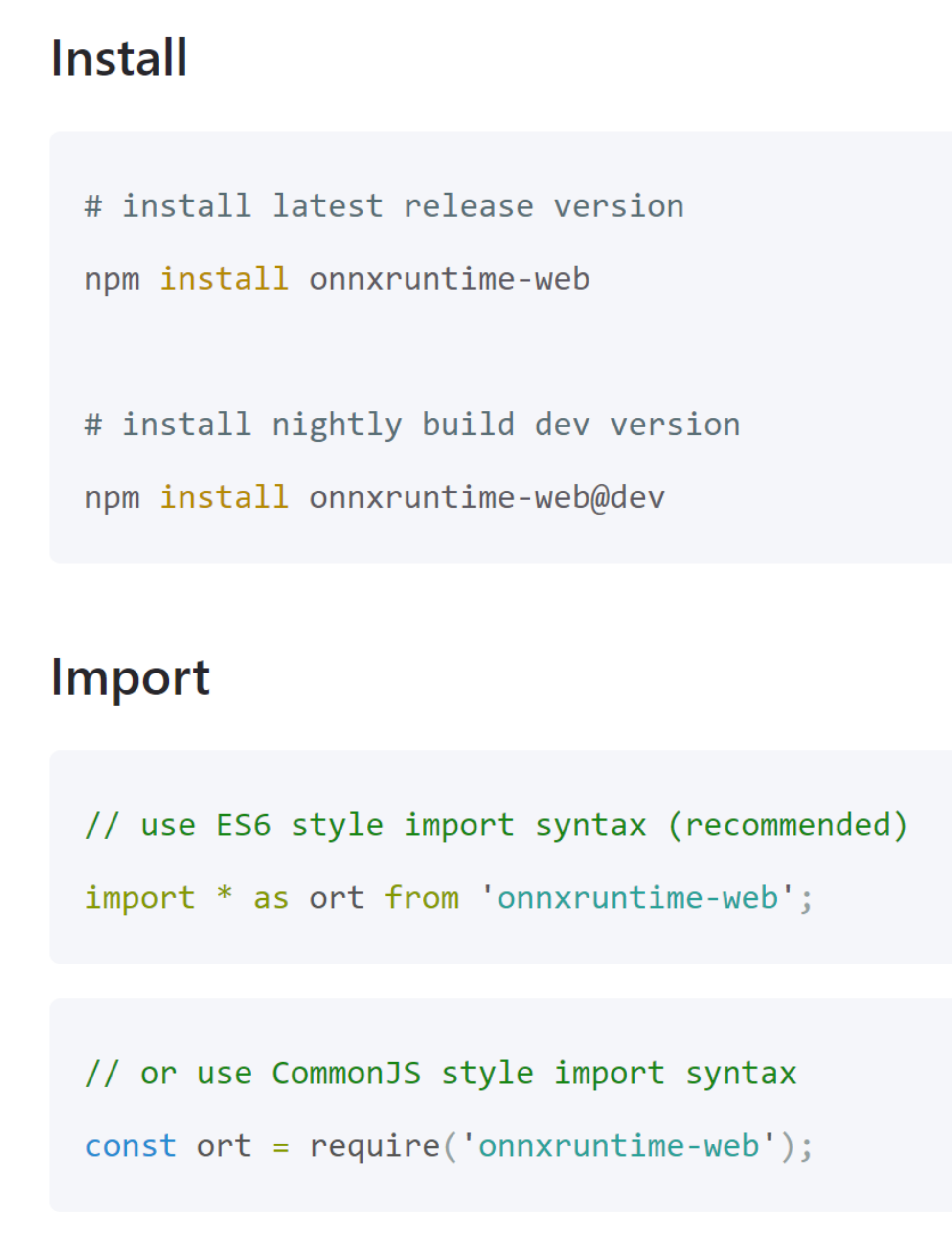
```
// or use CommonJS style import syntax
const ort = require('onnxruntime-node');
```

Node.js Demo

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.

- React Template
  https://github.com/microsoft/onnxruntime-nextjs-template



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

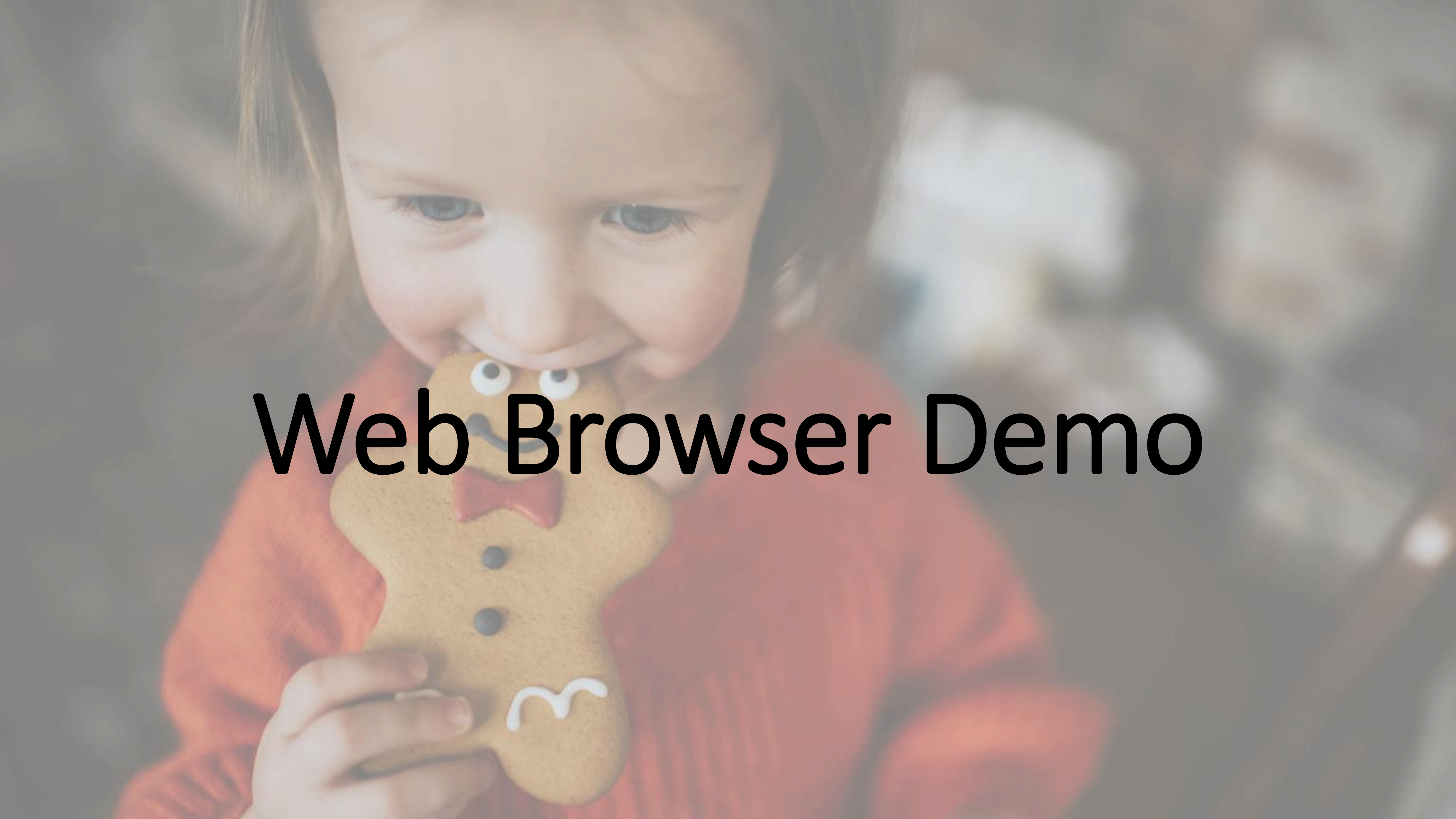
It works offline

It’s cheaper

Why not in the browser?

THE MODEL IS TOO LARGE AND REQUIRES HIGHER HARDWARE SPECS.

DOWNLOADED ONTO THE DEVICE



Web Browser Demo

Resources

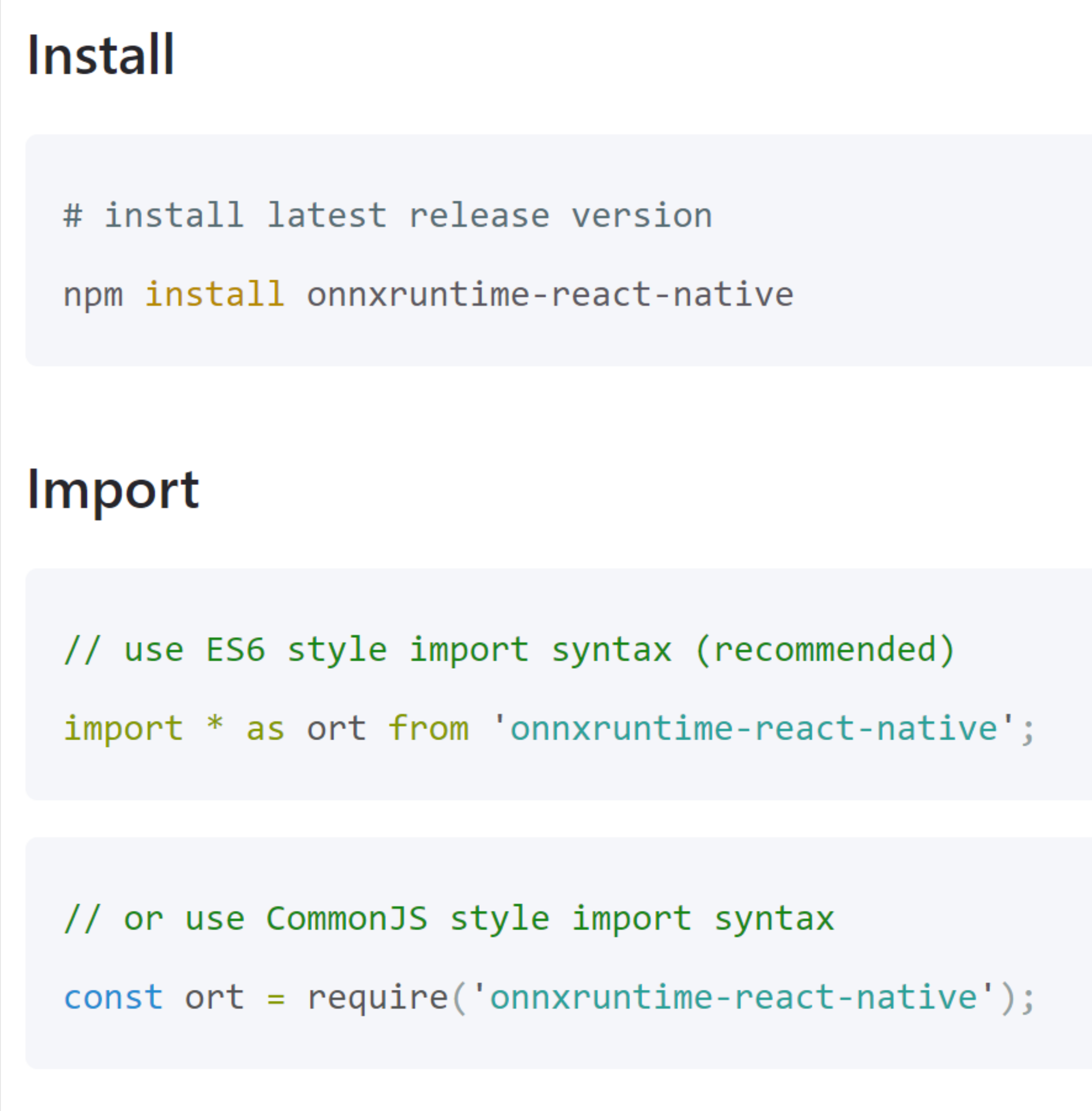
https://github.com/microsoft/onnxruntime-nextjs-template

https://github.com/microsoft/onnxruntime-web-demo

https://microsoft.github.io/onnxruntime-web-demo/#/

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
```

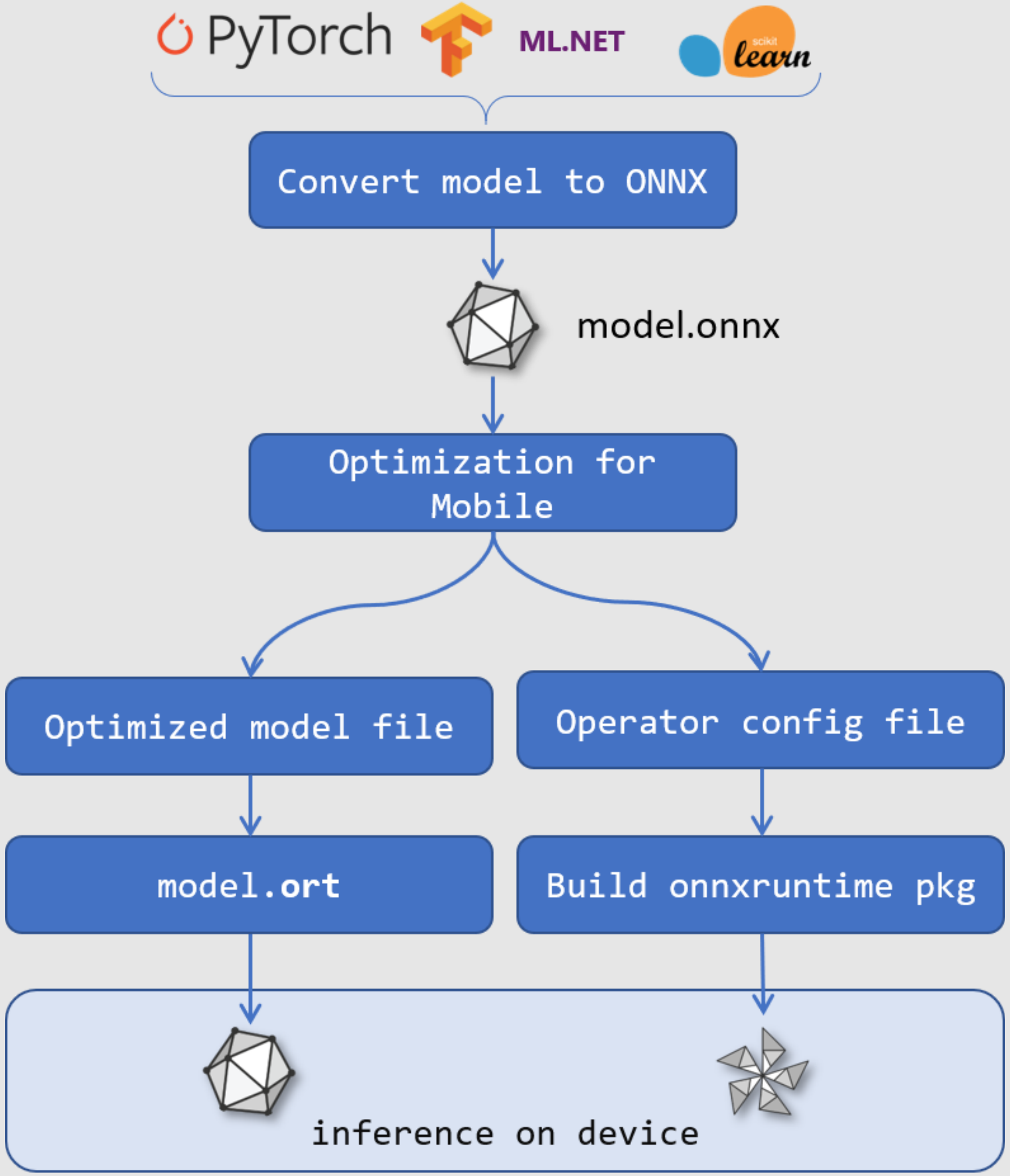
Import

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

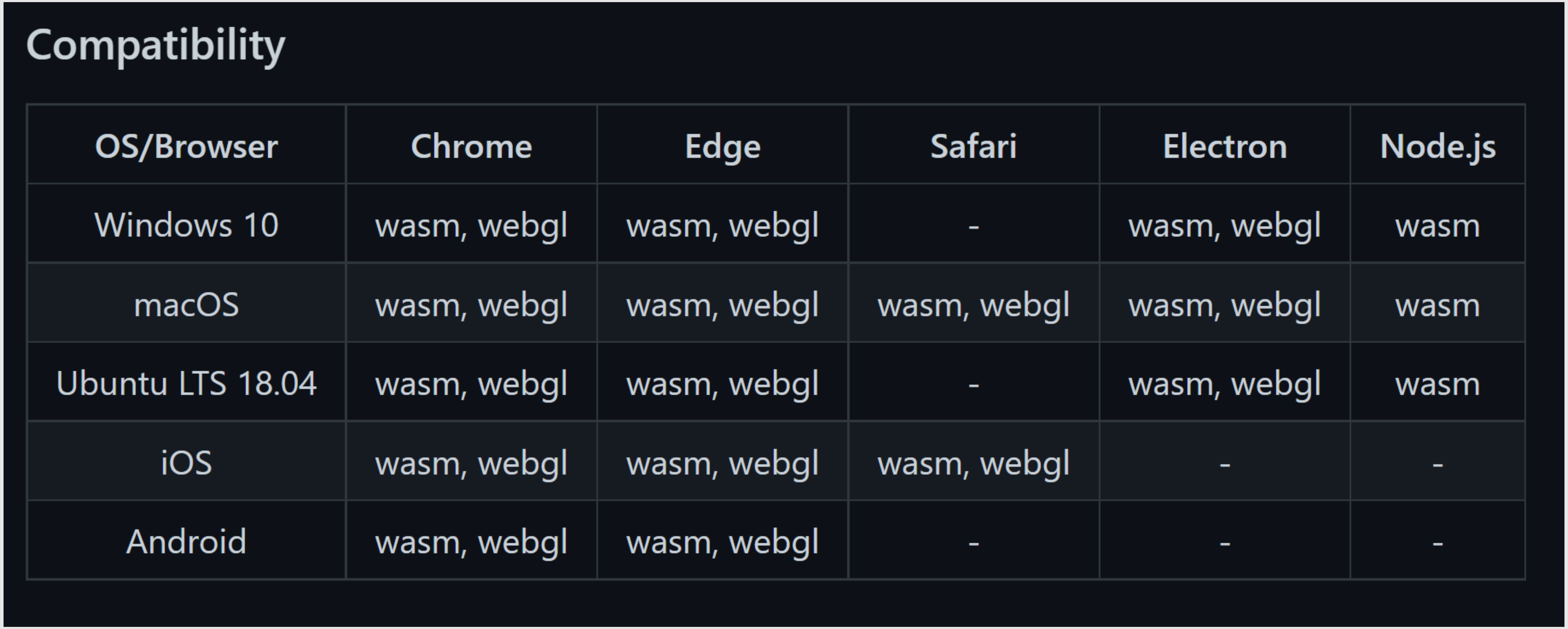
```
// or use CommonJS style import syntax
const ort = require('onnxruntime-react-native');
```

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	-	-	-

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

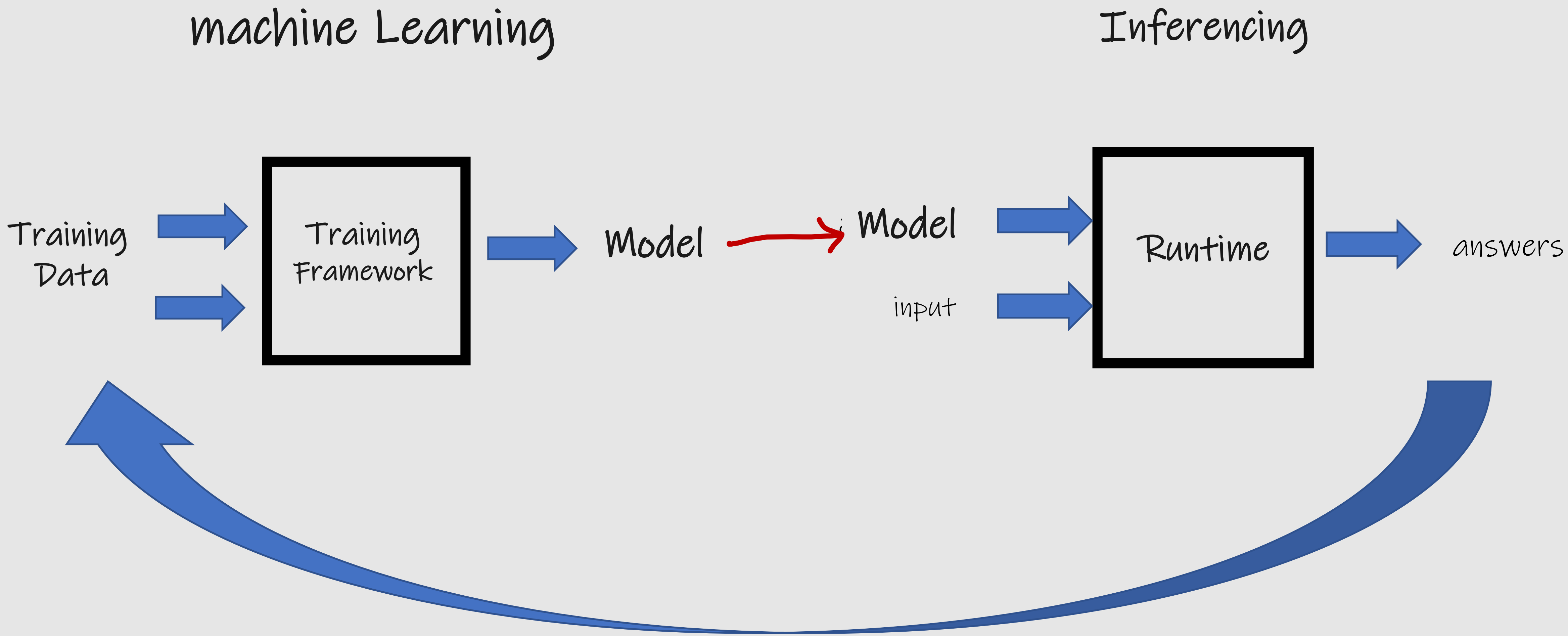
- ✓ How to create ONNX models

  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

ML Primer


machine Learning
Inferencing
Training Data
Training Framework
Model
Model
input
Runtime
answers

https://github.com/rondagdag/onnx-web-presentation

About Me

Ron Dagdag

MVP Microsoft® Most Valuable Professional

https://linktr.ee/rondagdag

- Director of Software Engineering at Spacee
- 5th year Microsoft MVP awardee
- Personal Projects
  www.dagdag.net
- Email: ron@dagdag.net
  Twitter @rondagdag
- Connect me via Linked In
  www.linkedin.com/in/rondagdag/
- Thanks for geeking out with me about ONNX