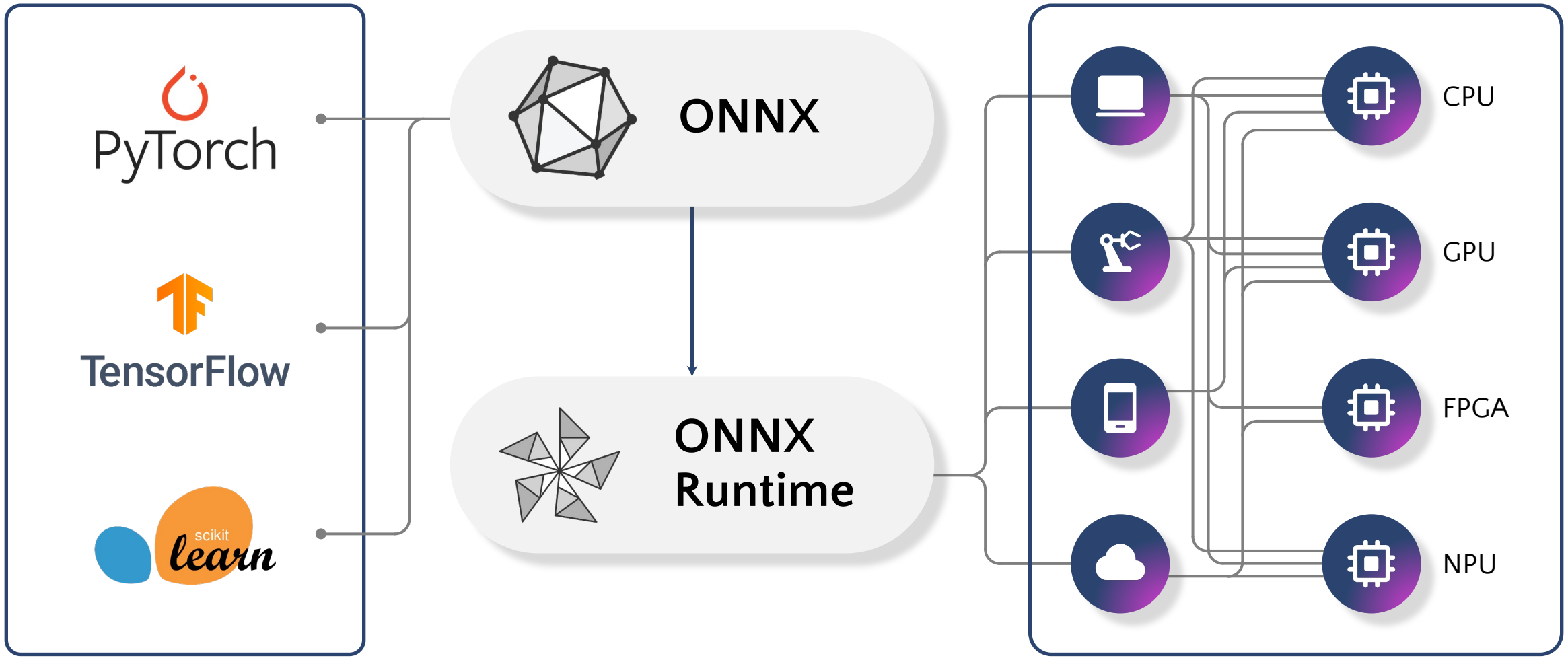
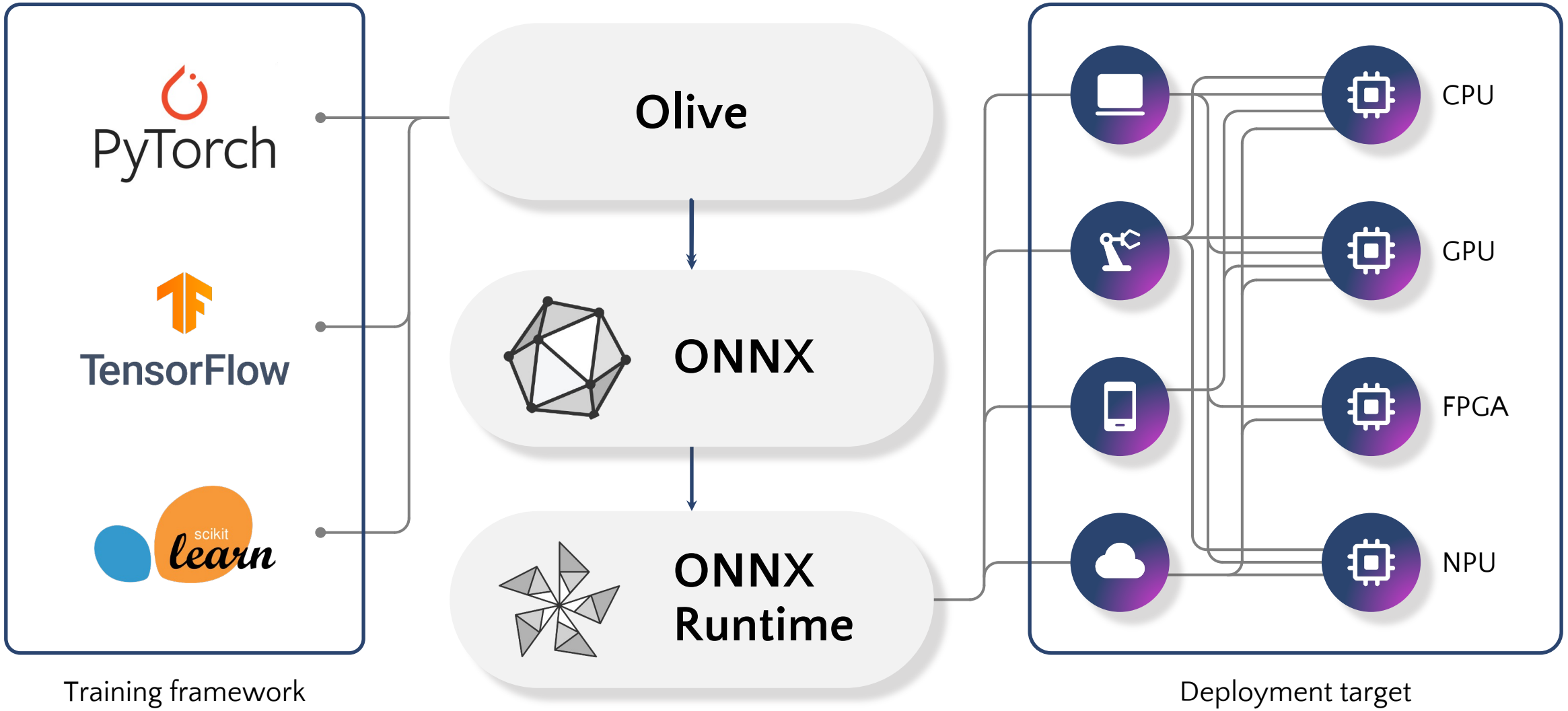
OLIVE

Hardware-aware model optimization solution

Emma Ning, Microsoft PRINCIPAL PM


PyTorch
TensorFlow
scikit learn
ONNX
ONNX Runtime
CPU
GPU
FPGA
NPU
Training framework
Deployment target



Simplify model optimization process
Ease the burden on developer for deep optimization toolchain knowledge

Olive Toolchain

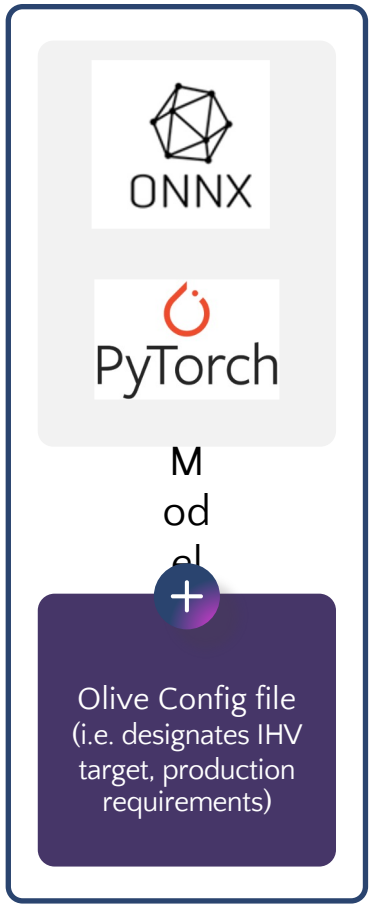
Olive optimization framework

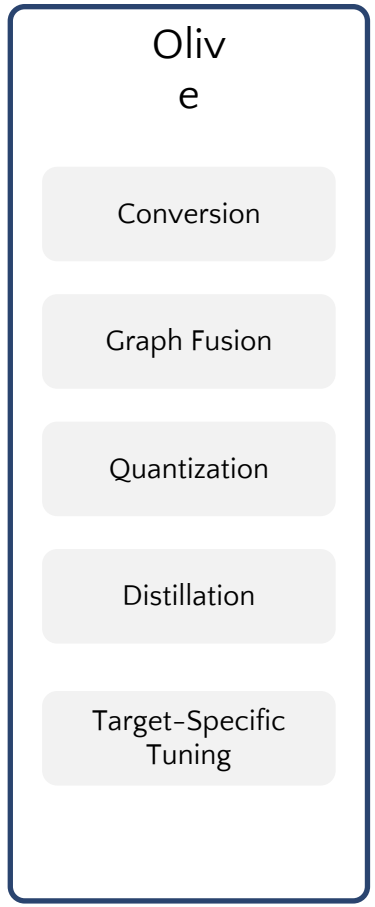
Olive (GitHub link)

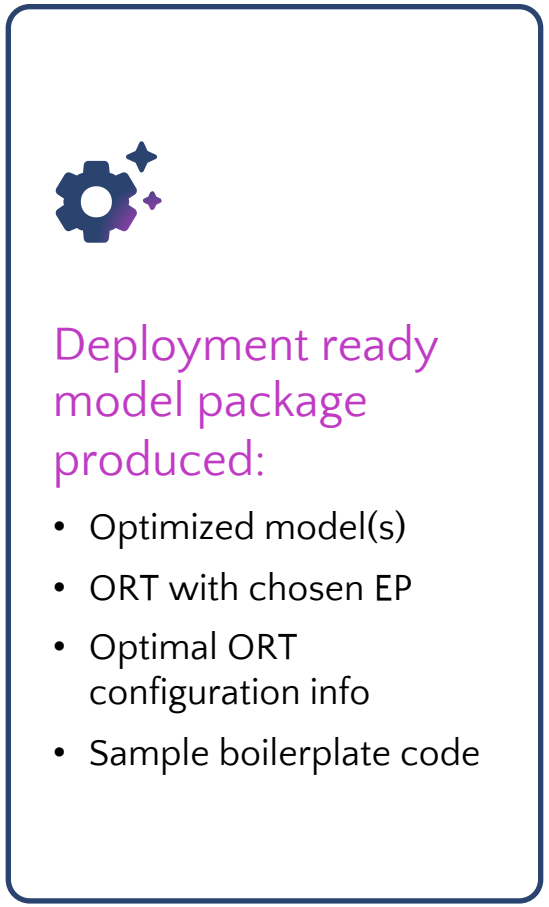
composes model conversion, compression, optimization techniques targeting a variety of hardware accelerators

Enables model optimization fitting to ONNX Runtime EPs across HW(CPUs, GPUs, NPUs)

Open Sourced in March 2023





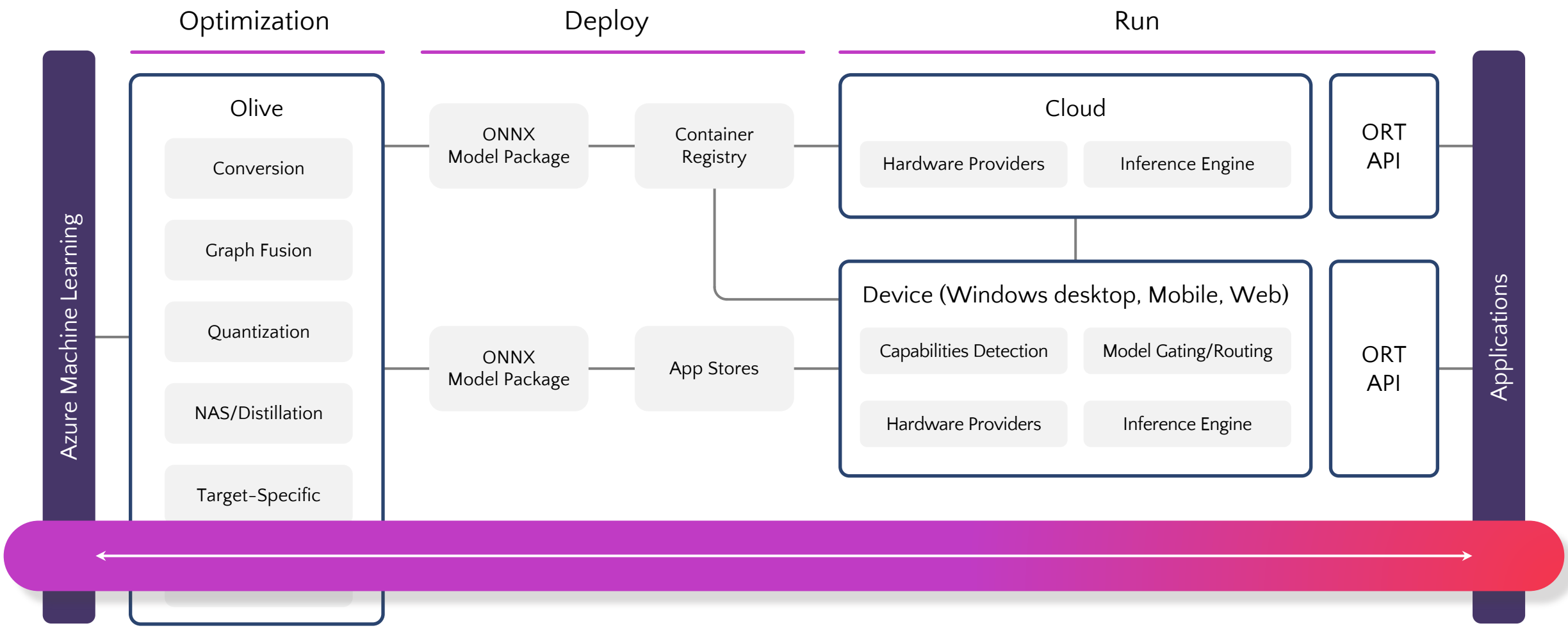


Deployment ready model package produced:

- Optimized model(s)
- ORT with chosen EP
- Optimal ORT configuration info
- Sample boilerplate code

Olive + ONNX Runtime in Hybrid Loop

Cloud+Edge ML platform built on heterogenous hardware



Olive + ONNX Runtime

Open source, E2E inference optimization solution

Olive – hardware-aware model optimization

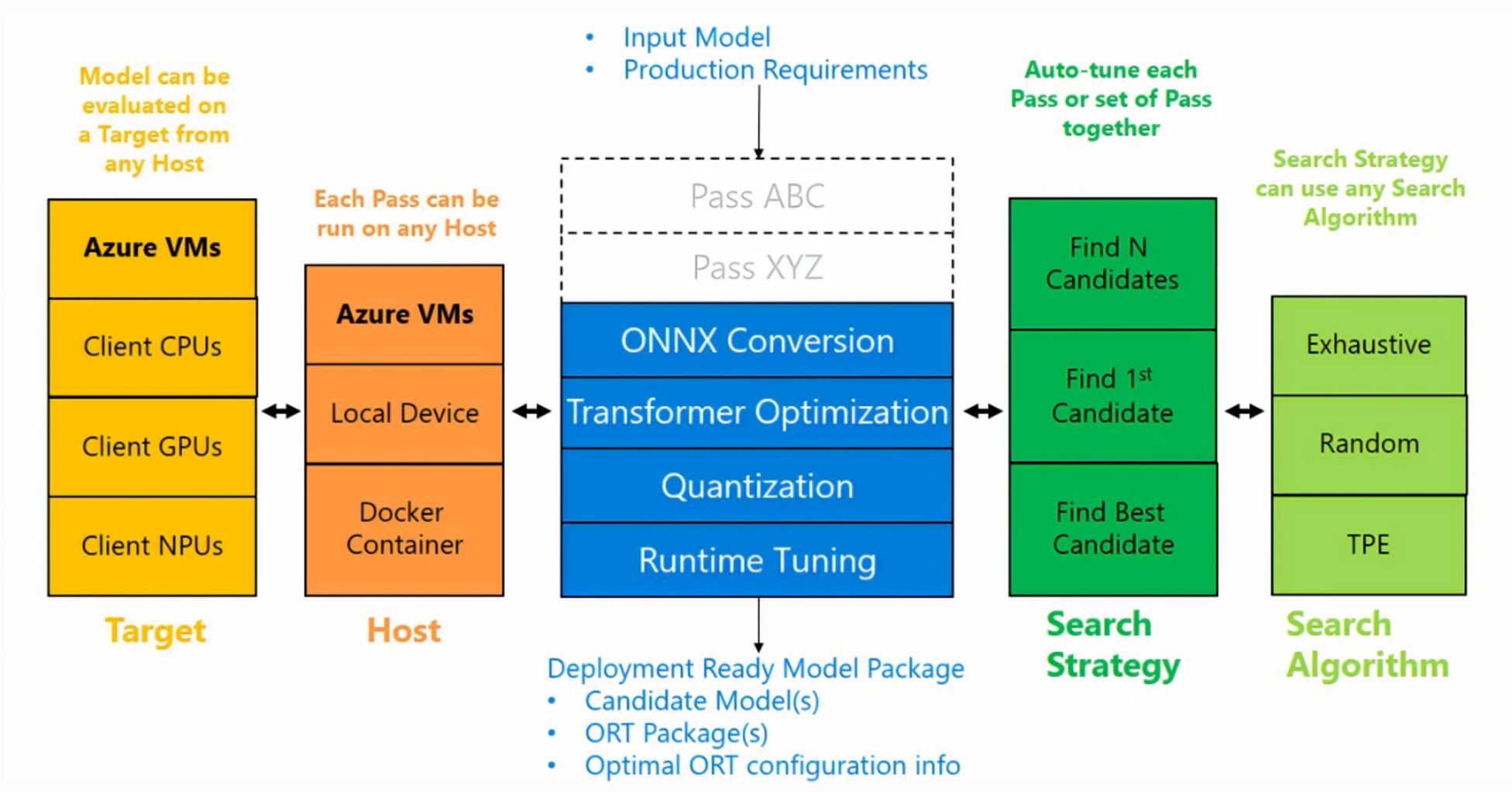
- Prepares model for the production use
- Ahead of Time hardware specific optimization
- Unified optimization framework for optimization toolkits integration

ONNX Runtime – high performance inference engine across hardware

- Runs the model on the edge and in the cloud
- JIT graph optimizations
- Unified runtime framework for hardware accelerators integration

Olive Architecture

Olive optimization framework



3 steps using Olive

1.Install Olive and necessary packages.

```
pip install olive-ai
```

1.Describe your model and your needs in a json configuration file.

1.Accelerate the model using Olive via a command line.

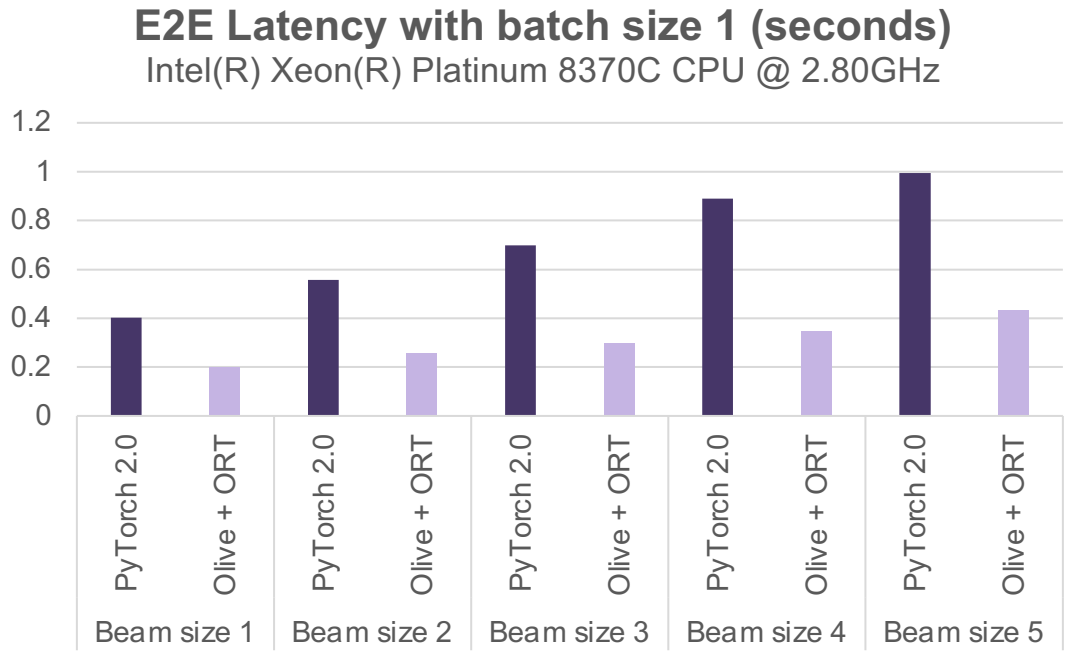
```
python -m olive.workflows.run --config my_model_acceleration_description.json
```

```
{
    "description" : "Complete my_model_acceleration_description.json used in this quick tour",
    "input_model":{
        "type": "PyTorchModel",
        "config": {
            "model_path": "resnet.pt",
            "io_config": {
                "input_names": ["input"],
                "input_shapes": [[1, 3, 32, 32]],
                "output_names": ["output"],
            }
        }
    },
    "evaluators": {
        "my_evaluator":{
            "metrics":[
                {
                    "name": "my_latency_metric",
                    "type": "latency",
                    "sub_types": [{"name": "avg"}]
                }
            ]
        }
    },
    "passes": {
        "onnx_conversion": {
            "type": "OnnxConversion",
            "config": {
                "target_opset": 13
            }
        },
        "quantization": {
            "type": "OnnxDynamicQuantization"
        },
    },
    "engine": {
        "log_severity_level": 0,
        "evaluator": "common_evaluator"
    }
}
```

Whisper with Olive+ONNX Runtime

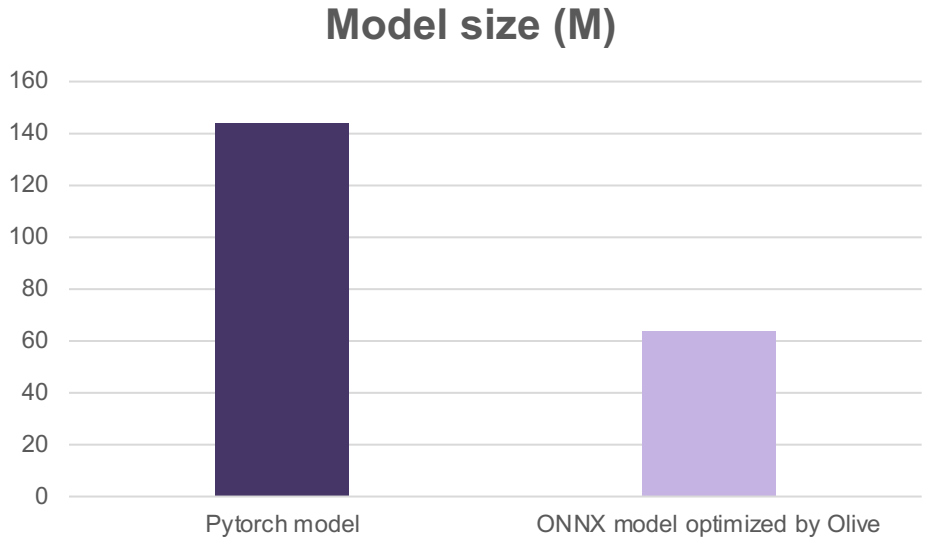
>2x E2E latency reduction

E2E latency: from loading audio to output result



2.25x model size reduction

ONNX model includes core graph and pre/post processing



Olive + IHVs

- Intel Neural compressor in Olive
  - Intel has contributed INC Quantization into the Olive
  - Supports both dynamic and static quantization

- AMD Vitis-AI Quantizer in Olive
  - AMD has contributed itis-AI Quantizer into the Olive
  - supports power-of-2 scale quantization methods
  - supports Vitis AI Execution Provider

We encourage and warmly welcome community contributions!

DISCUSSION