Accelerating GPU inferencing with DirectML and DirectX 12

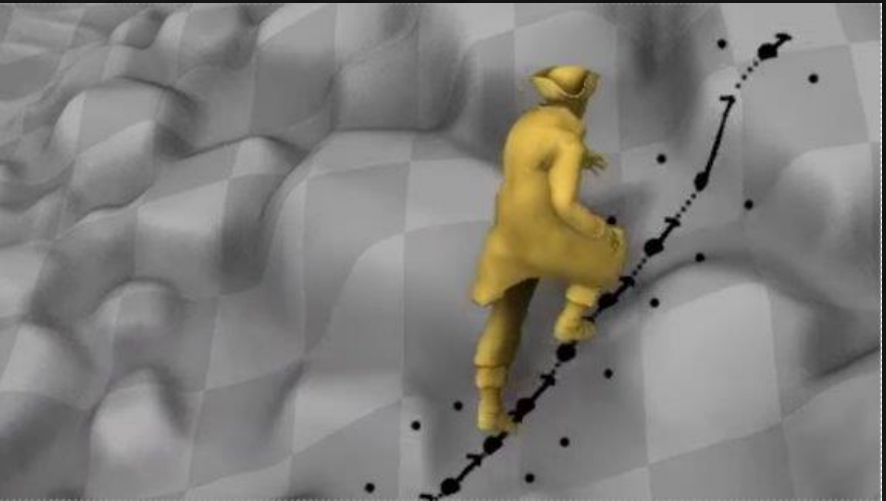
Shrinath Shanbhag

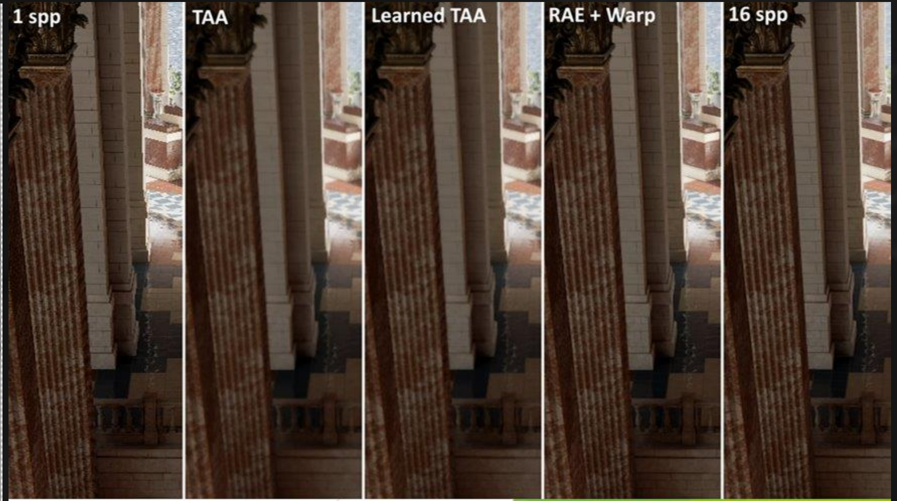
Senior Software Engineer

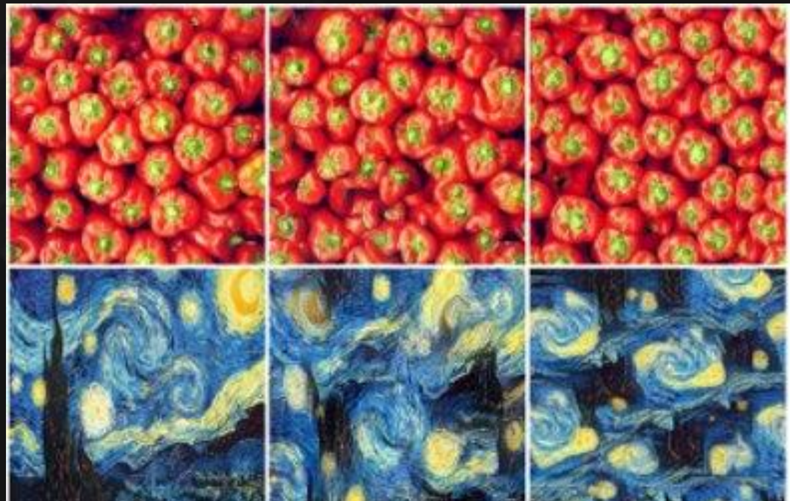
Microsoft Corporation

Machine Learning

- Machine learning has become immensely popular over the last decade
  - Traditionally used for linear regression and logistic regression (classification)
    - Example – Prediction of housing value, classification of samples into different classes
  - Is useful today in many novel applications scenarios such as - Super resolution, antialiasing, character motion synthesis, texture synthesis, human-like player AI and more







Machine Learning at Microsoft

- Microsoft has made huge investments in AI and Machine Learning across the company.

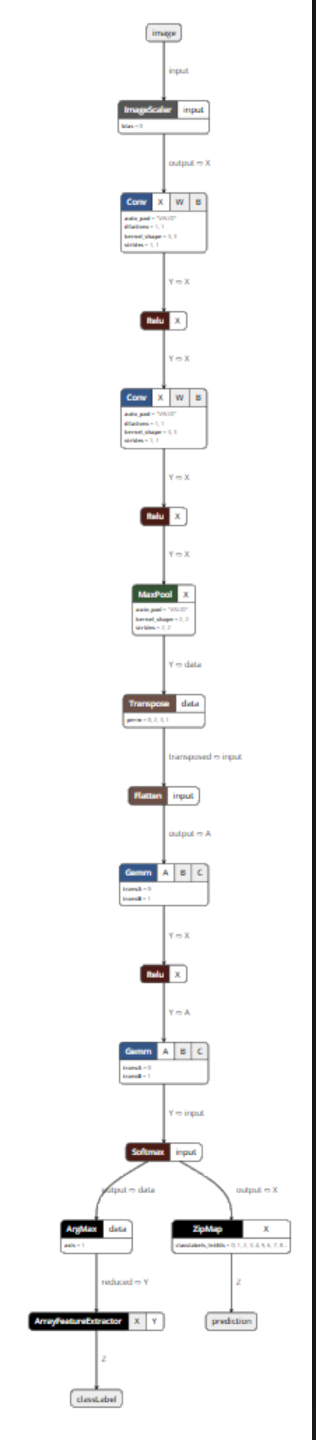
- AI capabilities are embedded in products such as
  - Office 365 uses ML for productivity enhancement features like flood fill.
  - Windows 10 uses ML for Windows Hello, intelligent video creation in the Photos App.
  - Bing and Cortana use AI to search and answer questions etc.

- Microsoft Cognitive Toolkit, Azure Machine Learning Services, Windows Machine Learning, are part of Microsoft's Machine Learning API's and Services
  - Microsoft Cognitive Toolkit is a free, easy to use, open-source, commercial-grade toolkit that trains deep learning algorithms.
  - Azure ML services provide machine learning at big data scale and supports a number of frameworks such as Caffe, Cognitive Toolkit, TensorFlow and others.
  - Windows Machine Learning allows you to use trained ML models in you applications, to evaluate locally on Windows 10 devices leveraging the device's CPU and GPU.

Windows Machine Learning

- Previewed with Windows Spring Creators Update
  - Applications use the WinML API for inferencing
  - Enables a variety of machine learning scenarios in your PC apps and games
  - Consumes the Open Neural Network Exchange (ONNX) model format

- Simple to use
  - Train your model in framework of choice and/or with cloud services
  - Convert model to Open Neural Network Exchange (ONNX)
  - Use WinML to load, bind, and evaluate in your application



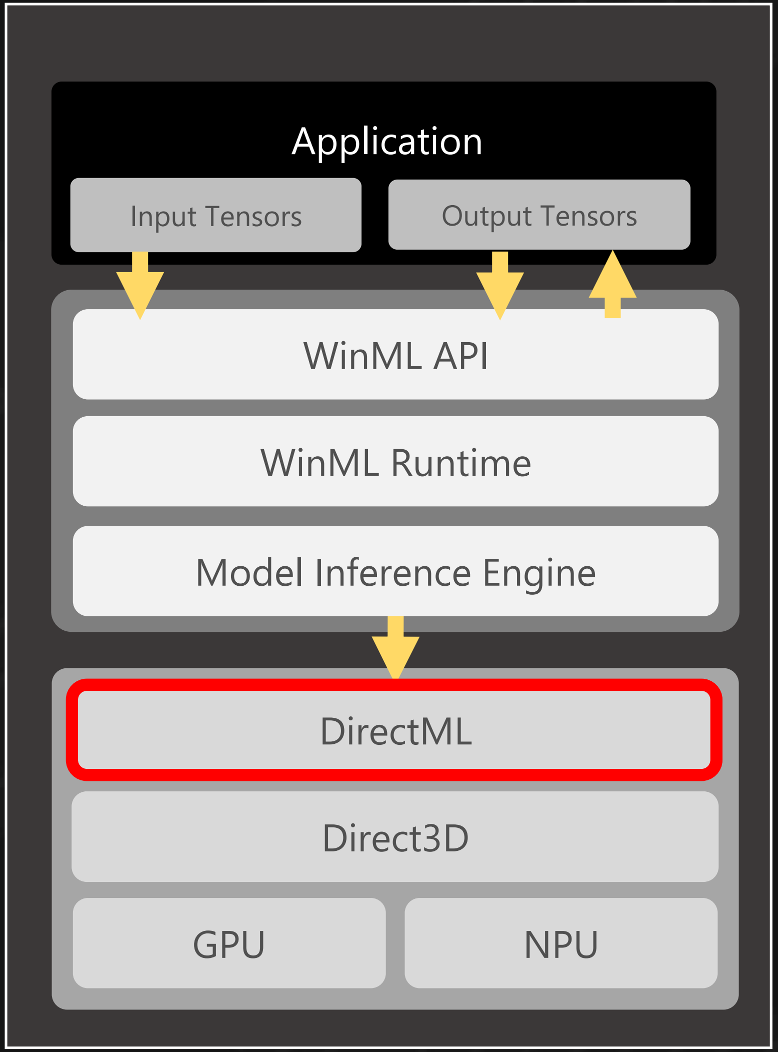
WinML today

- Graduated out of preview namespace
  - Windows.AI.MachineLearning available today in Windows Insider Program (WIP) builds
- First release targets ONNX 1.2.2
- Additional feature support
  - Models trained with FP16 weights reduce memory footprint and increase performance
  - Custom operators give flexibility to expand functionality beyond ONNX
  - Metacommands enable better performance and hardware utilization

Windows Machine Learning Architecture

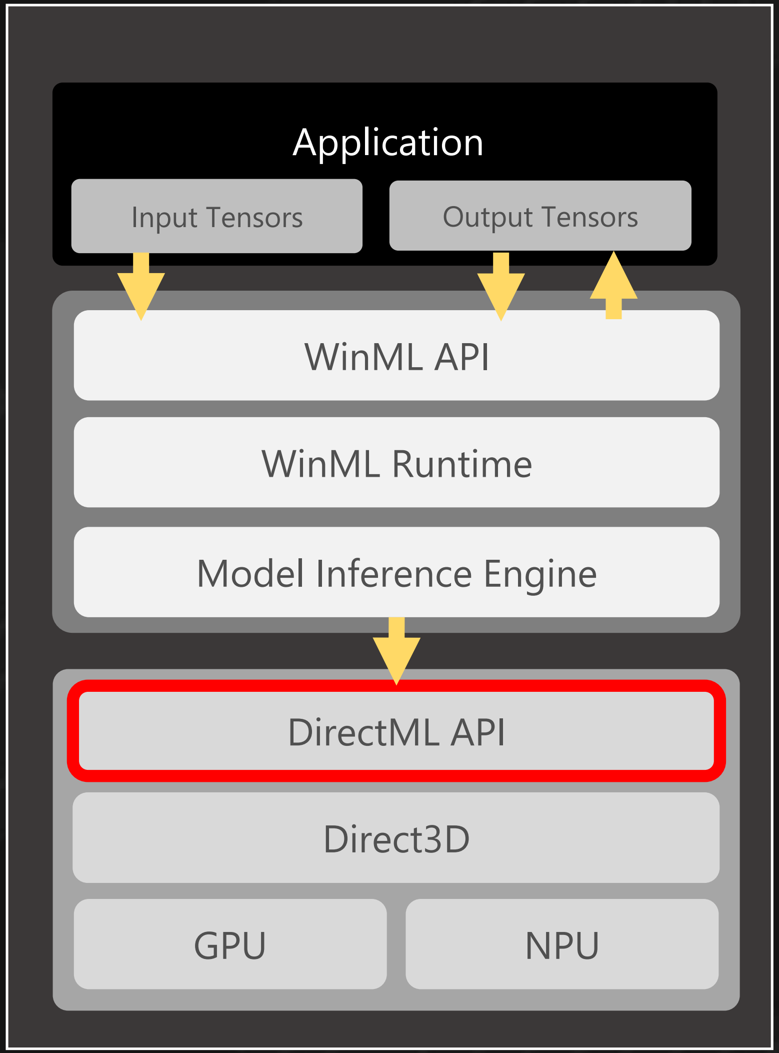
- Windows Machine Learning is
  - Hardware accelerated
  - Supported on all DX12-capable hardware
  - Delivered to all Windows customers in the OS

- Uses DirectML for GPU hardware acceleration



Hello DirectML

- Part of the Microsoft DirectX® family of APIs
- Low-level API for performing ML inferencing
- DirectX 12 style interface
  - Very low overhead, thin abstractions over silicon
  - Broad hardware support
  - Conformant, compatible, consistent
- Puts control into developer's hands



Why DirectML ?

- Winml API is primarily model focused: Load, Bind , Eval

- Domains like games need a different level of abstraction
    - Developer control
    - High performance
    - Low latency
    - Fine-grained resource management
    - Suitable for integration into existing engines or rendering pipelines

- ML frameworks and libraries out there with similar requirements
    - Cognitive Toolkit, PyTorch, MXNet, TensorFlow etc.

What does DirectML do?

- Provides hardware-accelerated ML operators for inferencing.
  - Support from hardware partners enables architecture-specific optimizations

- Provides developer flexibility and control
  - Resource management
  - Schedule ML work as they see fit
  - Interleave work with other DX12 workloads

- Supported on all DX12-compatible hardware
  - Examples:
    - NVIDIA Kepler and above
    - AMD Radeon 7000-series and above
    - Intel Haswell (4th-gen core) and above
  - If no GPU is available, fall back to CPU

Which operators does DirectML provide ?

- Elementwise
- MatMul
- Activation
- FC
- Convolution
- Pooling
- Normalization
- Random
- RNN
- GRU
- LSTM
- And more…

DirectML Programming Model

- DirectML is a low level programming API and so the workflow is more involved.
  - You manage most things yourself
  - Parse the graph or create it programmatically on the fly
  - Create and manage buffers
  - Upload and download data to and from GPU
  - Create and dispatch each operator

What is the DirectML workflow ?

- Similar workflow to D3D12
  - Create DirectML device
  - Create resources, operators
  - Bind resources and PSO
  - Execute command list on your D3D12 command queue

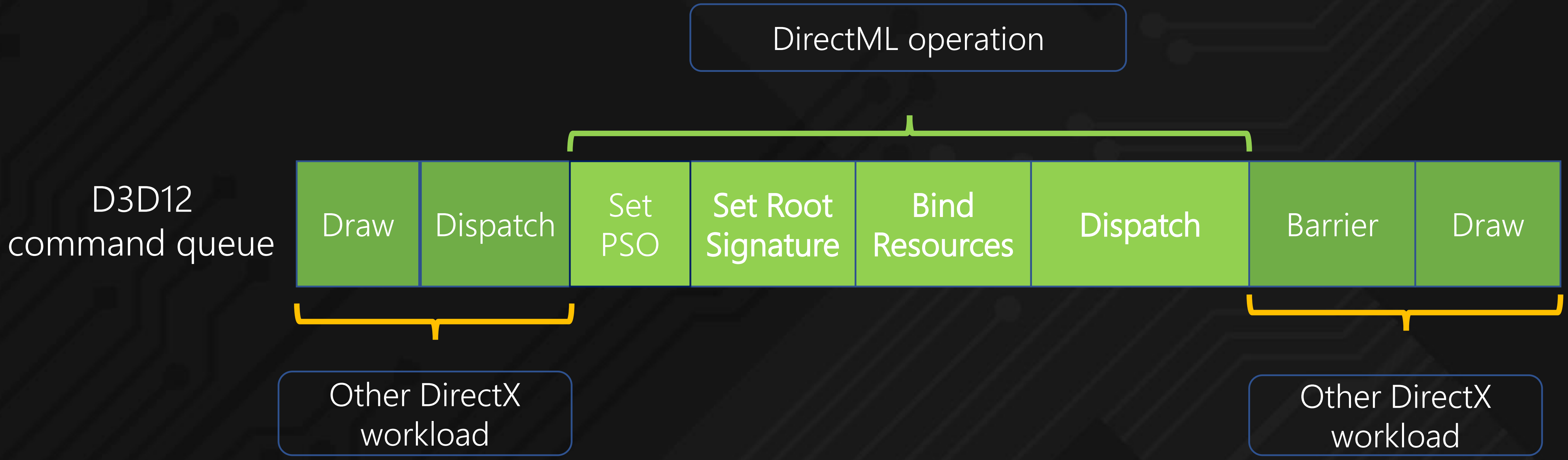
- `CreateDmlDevice(ID3D12Device)`
- `IDmlDevice::CreateDMLDeviceContext`
- `IDmlDevice::CreateDMLResource`
- `IDmlDevice::Create*Operation`
- `IDmlDeviceContext::AddOperation`
- `ID3D12CommandQueue::ExecuteCommandLists`

- Resource lifetime and synchronization are caller’s responsibility

Demo

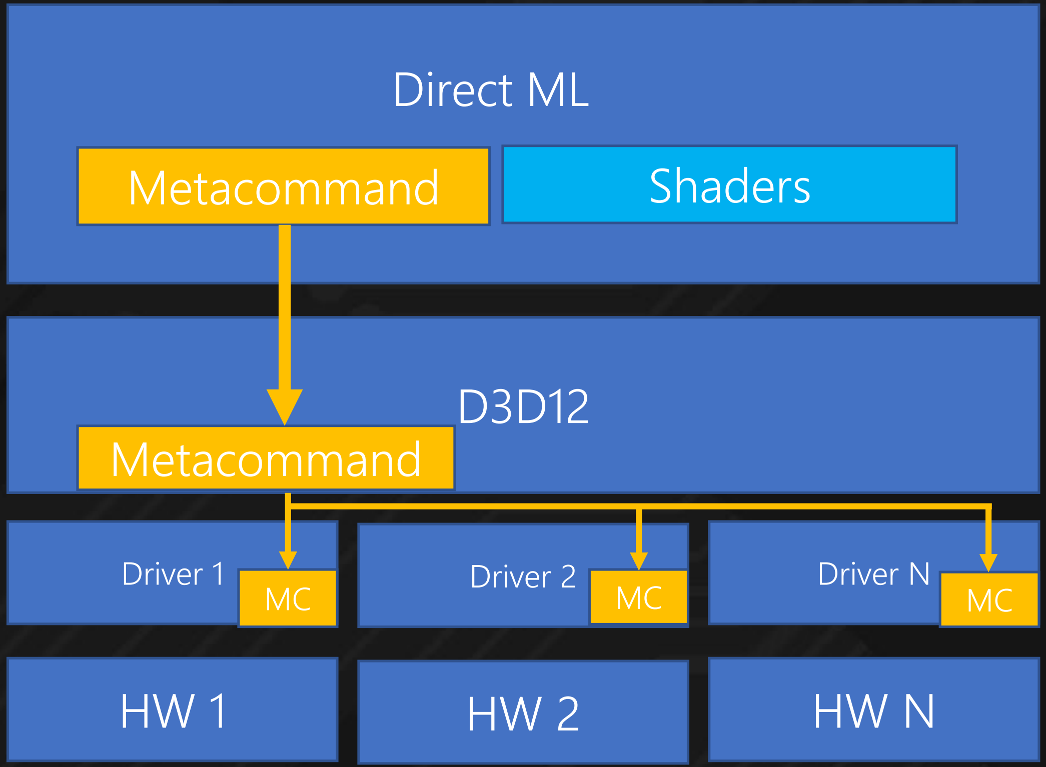
Can I see some DML code ?

What do DirectML operations look like ?



How does DirectML perform?

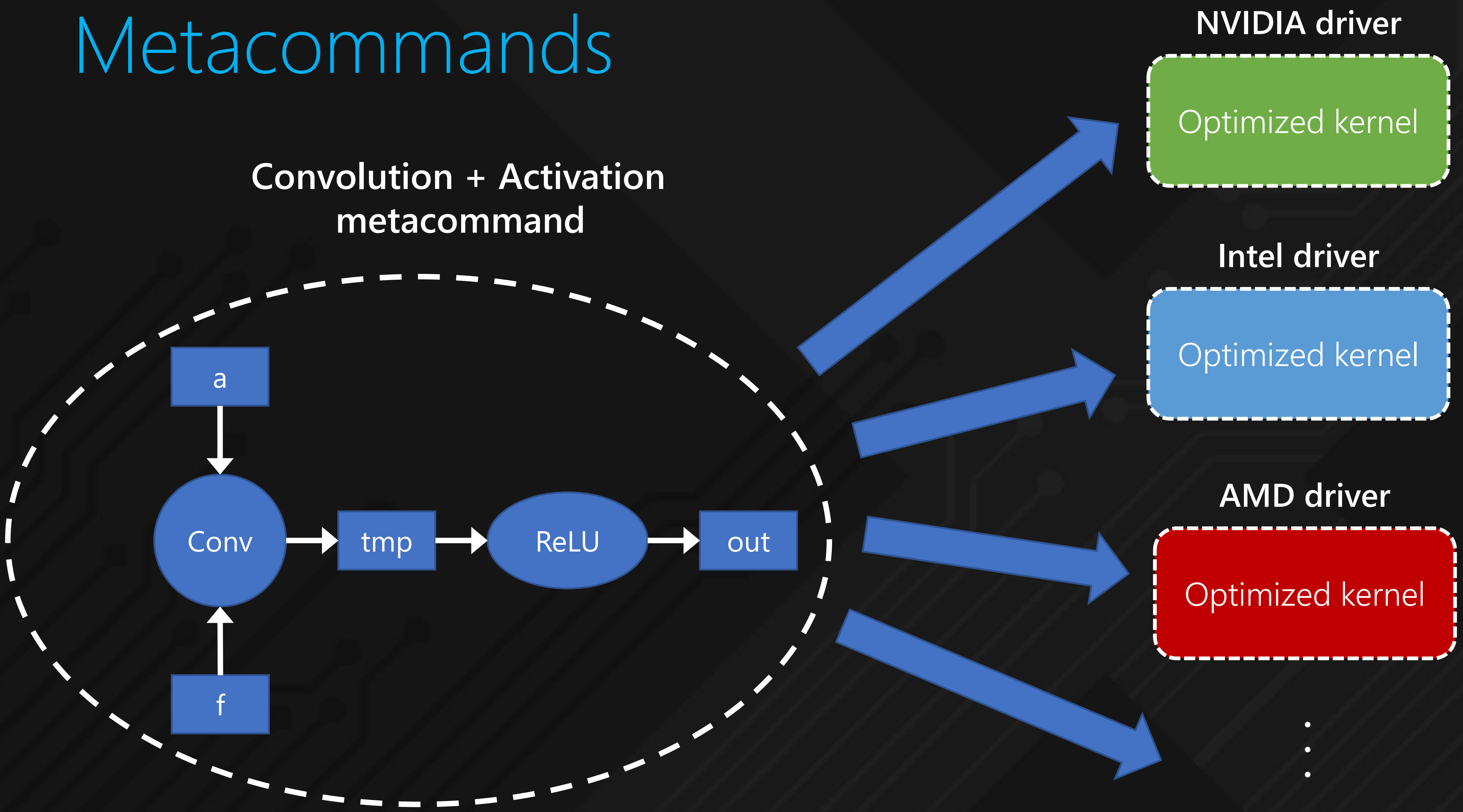
- DirectML aims to achieve HW native performance
- DirectML uses new DirectX 12 feature called Metacommands
- Metacommands allow vendors to expose hardware-specific optimizations



What are Metacommands ?

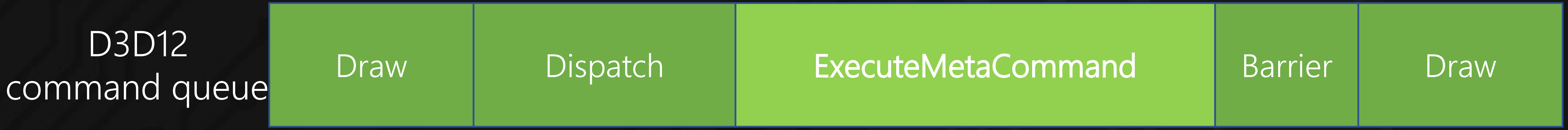
- DirectML defines a set of machine learning metacommands
  - Enables hardware-specific optimizations even though DirectML is a hardware-agnostic API
  - Efficient compute shader fallbacks for hardware/drivers without support
- Allows DirectML to perform better than generic hand-written compute shaders

Metacommands



Metacommand operations

- Execution of metacommands performed on D3D12 command lists
  - Just like Draws, Dispatches, etc.



D3D12 command queue

Draw

Dispatch

ExecuteMetaCommand

Barrier

Draw

When should I use DirectML?

- You have a trained ML Model which is ready to go.
- You have an app that needs to deal with demanding real-time, high-performance, or resource-constrained scenarios
  - Examples: Games can use ML models for upscaling, denoising, anti-aliasing, style transfer etc.
- You are writing custom ML frameworks and need a high performance backend on Windows

What is the DirectML Roadmap ?

- DirectML still under active development
- First preview version in Spring 2019
- Private preview available for early adopters - contact us at:

askwindowsml@microsoft.com

- Stay tuned to the DirectX blog – slides will be posted along with links and information on how to get started with Windows ML.

https://blogs.msdn.microsoft.com/directx/

Questions?