



Bringing AI Everywhere

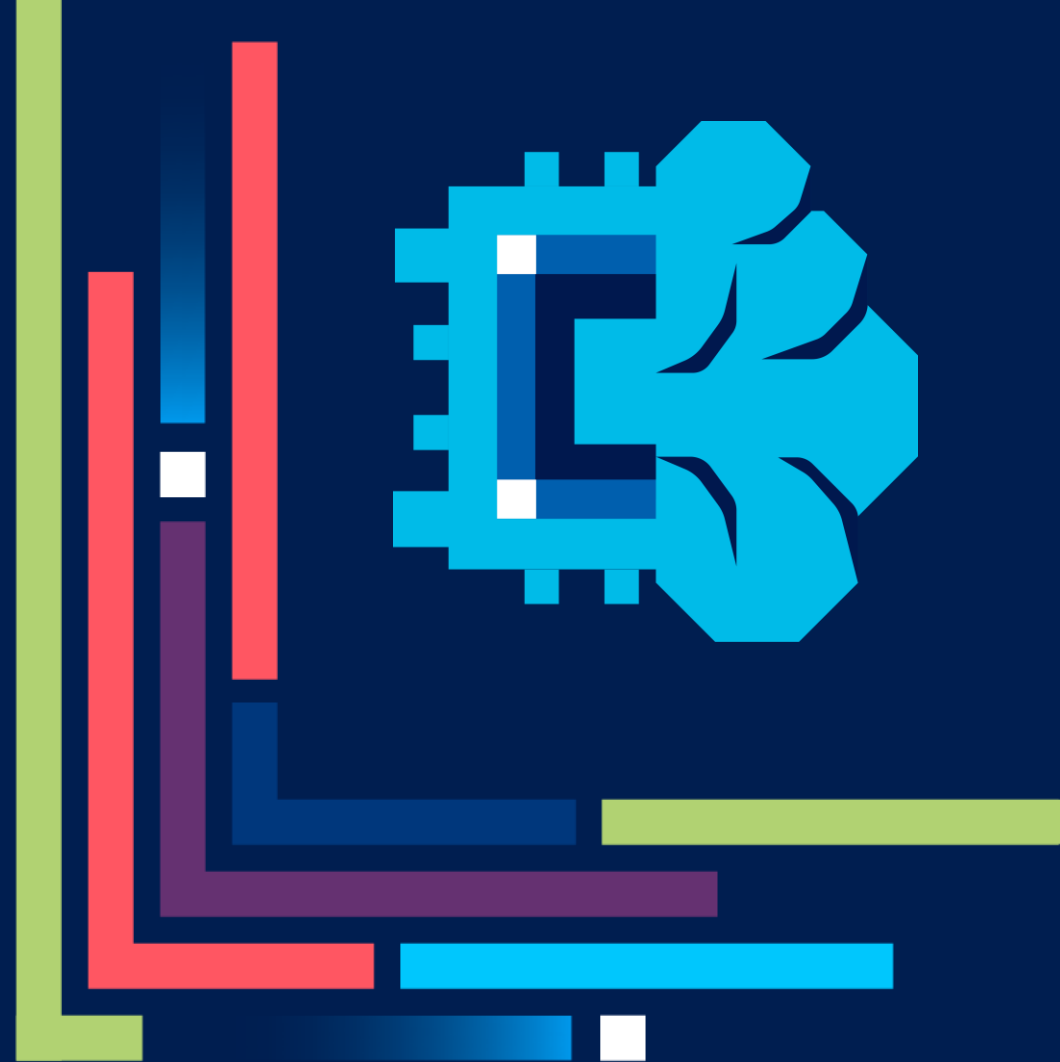
Web Neural Network API

Accelerating Deep Neural
Networks on the Web

Jamie Chang, CCE/WED

Ningxin Hu, SSE/WPE

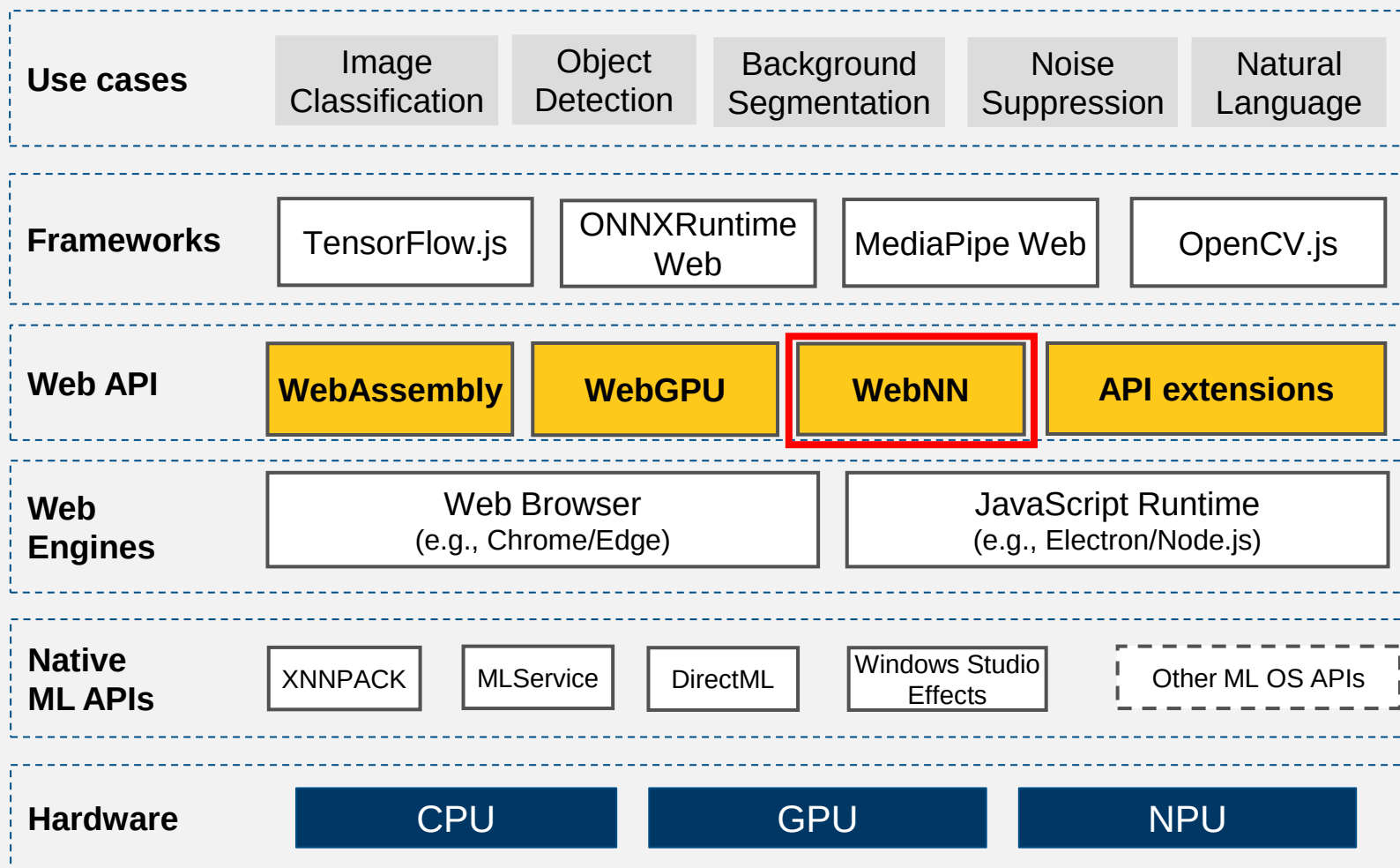
March 27th, 2024



Web Neural Network (WebNN)

- A new web standard that allows web apps and frameworks to accelerate deep neural networks with on-device hardware such as GPUs, CPUs, or purpose-built AI accelerators.
- An emerging W3C Web standard API for AI acceleration, Intel and Microsoft spec co-editors.
- Brings a unified abstraction of neural networks to the web
- Provides AI hardware (XPU) access through native machine learning API of the operating system
- Delivers near-native performance and reliability of results while delivering the next gen use cases
- Status:
 - Spec:
 - CNN/RNN - Candidate Recommendation published
 - Transformers – Q4'23
 - Implementation:
 - [XNNPACK/CPU for Win/Linux]: Dev Trial is available in Chrome and Edge Canary
 - [DirectML/GPU for Win]: Initial support landed upstream, Dev Trial in Q4'23
 - [DirectML/NPU for Win]: work in progress
 - [MLService/NPU for CrOS]: POC done, patches upstreaming

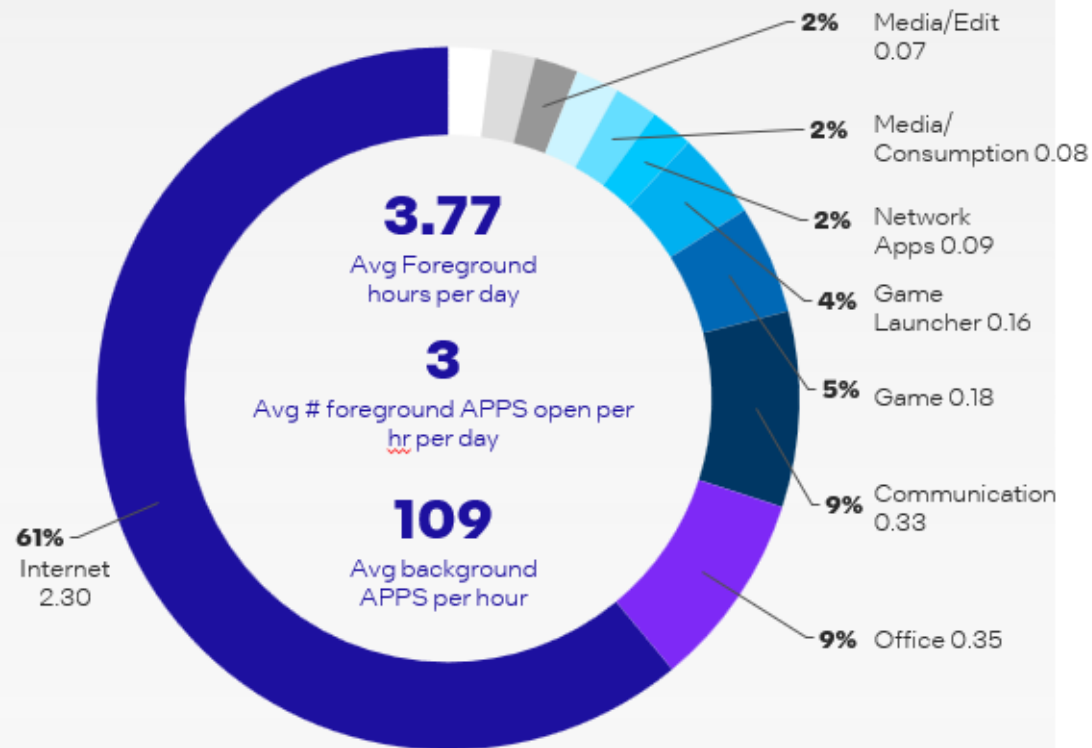
XPU-Accelerated Web AI Overview



Almost 2/3rd of PC cycles worldwide are spent on the web

Emerging Web APIs aim to bring AI to the masses across all platforms!

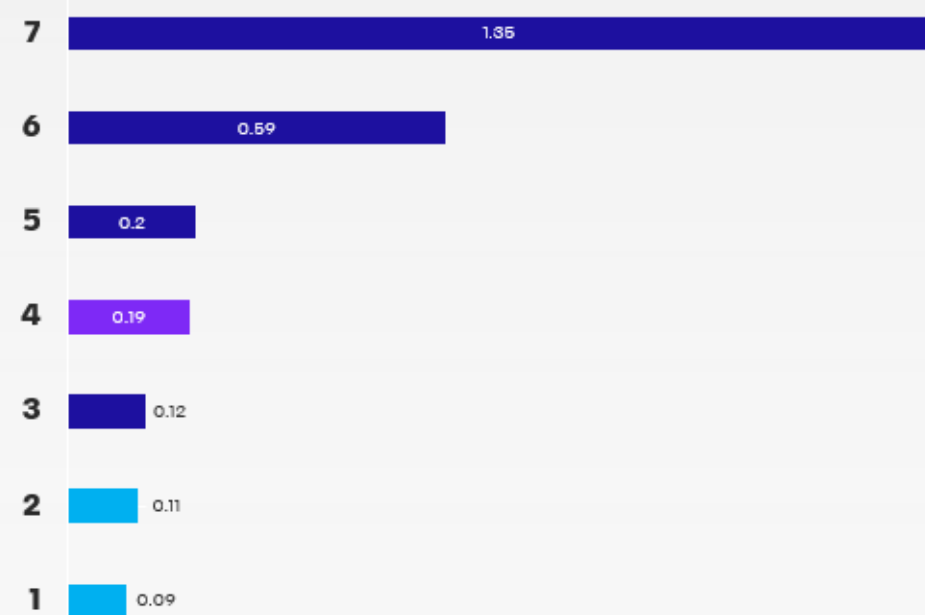
Application Usage by Category



† Source: Intel DCA, July 2022

* Other names and brands may be claimed as the property of others.

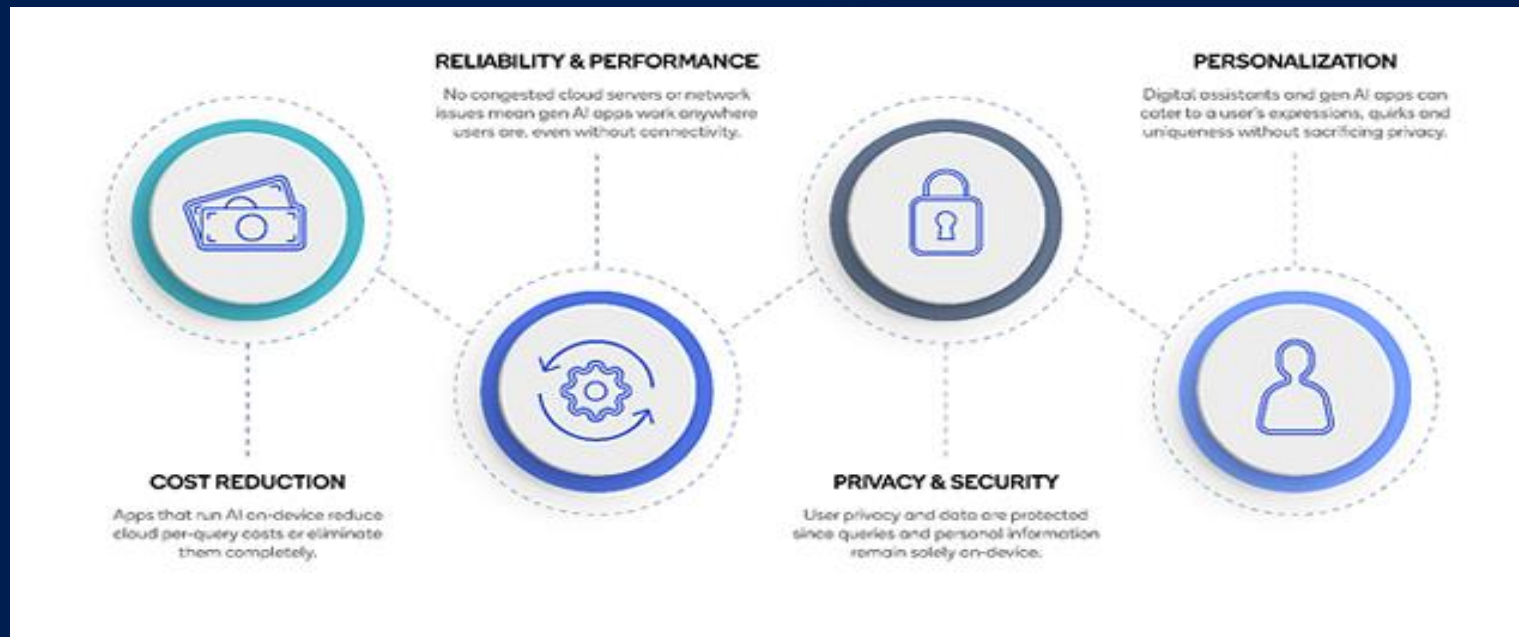
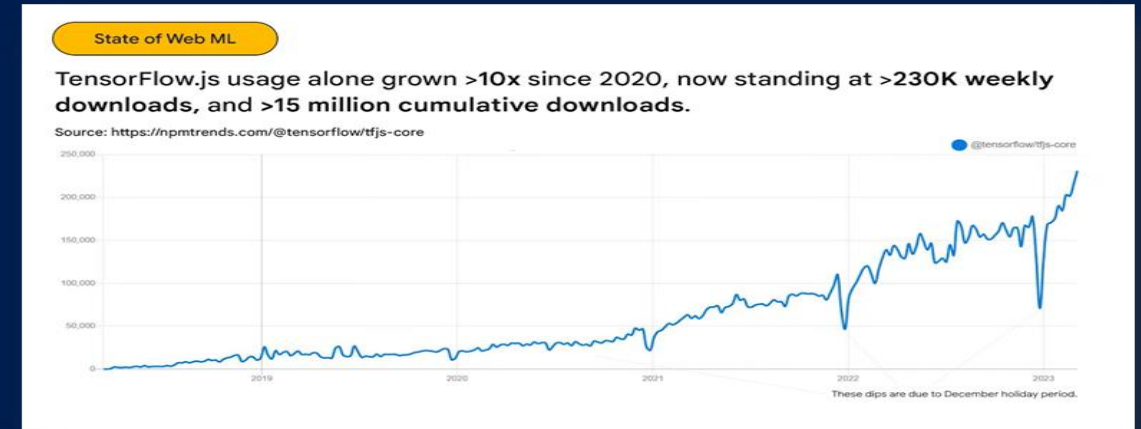
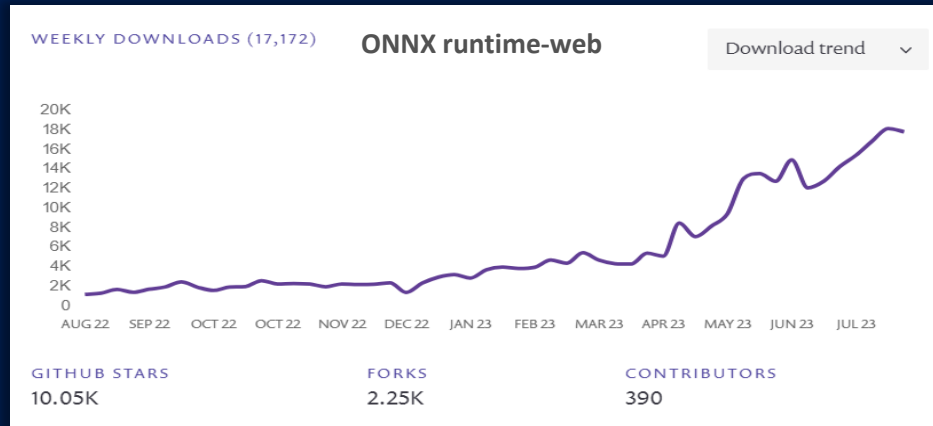
Browsers are top apps on PC*



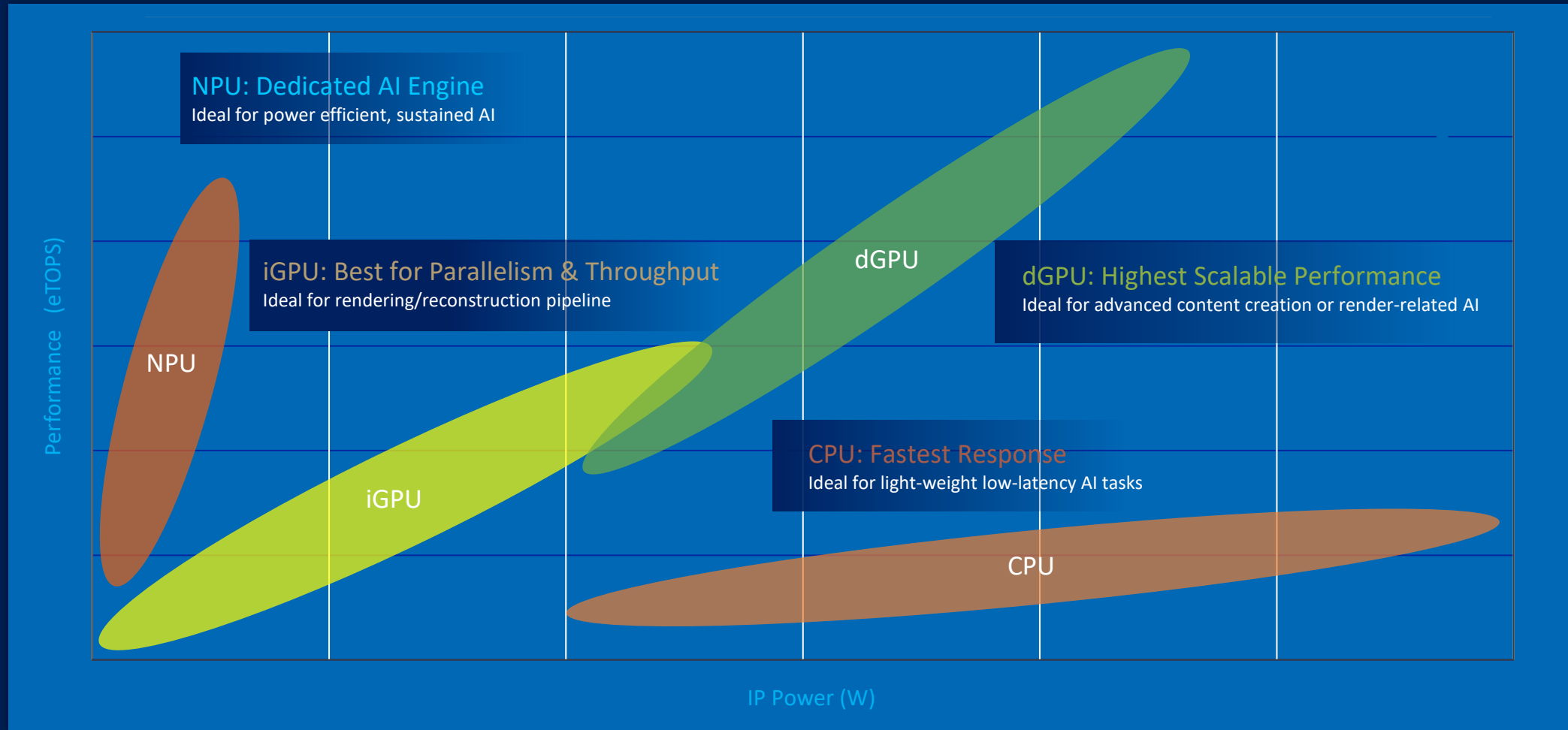
Web Weekday: **61%**
Weekend: **66%**

Source: Intel DCA, July 2022
~8M PCs Worldwide

Web is the best platform to scale AI delivery



Heterogenous Hardware for AI



Source: Intel illustration of generally accepted principles

Agenda

WebNN

Delivering AI experiences across CPU, GPU, NPU

WebNN API & Spec

Browser Implementations

Performance & Demo

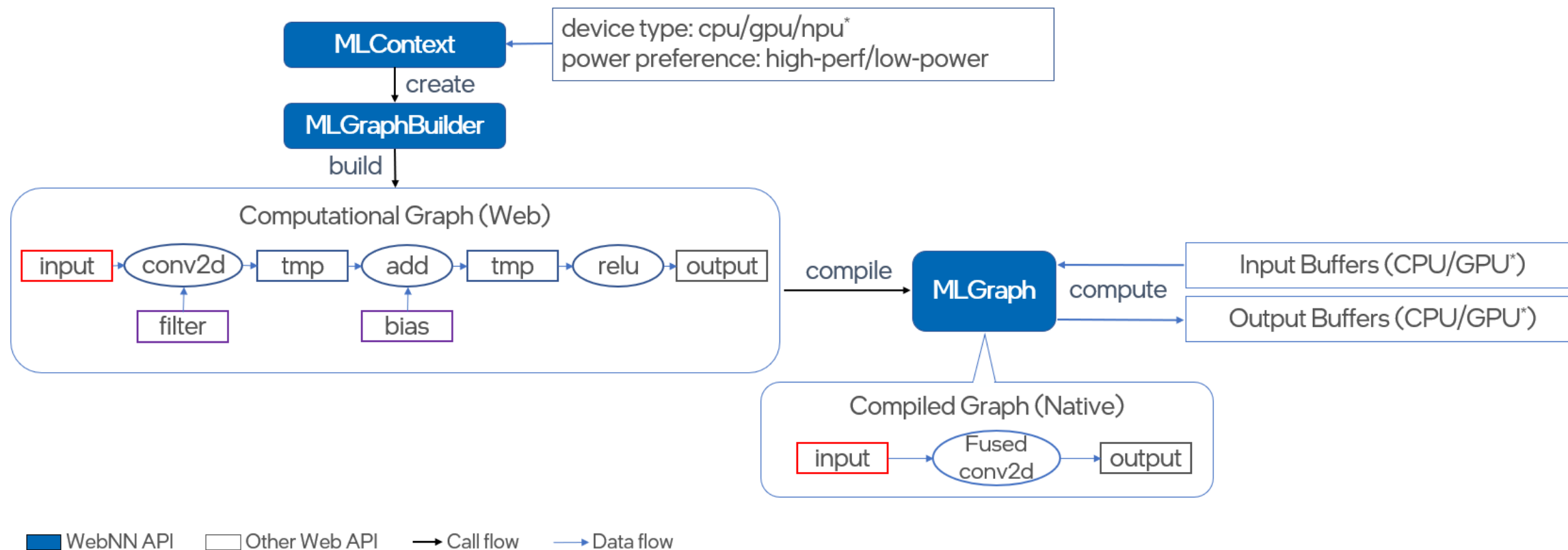
Call to Action

References

WebNN Specification

W3C Candidate Recommendation Draft	TABLE OF CONTENTS		Web Neural Network API W3C Candidate Recommendation Draft, 22 March 2024  ▼ More details about this document This version: https://www.w3.org/TR/2024/CRD-webnn-20240322/ Latest published version: https://www.w3.org/TR/webnn/ Editor's Draft: https://webmachinelearning.github.io/webnn/ Previous Versions: https://www.w3.org/TR/2024/CRD-webnn-20240321/ History: https://www.w3.org/standards/history/webnn/ Implementation Report: https://wpt.fyi/results/webnn?label=master&label=experimental&aligned&q=webnn Test Suite: https://github.com/web-platform-tests/wpt/tree/master/webnn Feedback: GitHub Inline In Spec Editors: Ningxin Hu (Intel Corporation) Dwayne Robinson (Microsoft Corporation) Former Editor: Chai Chaoweerasit (Microsoft Corporation) Explainer: explainer.md Polyfill: webnn-polyfill / webnn-samples
	1	Introduction	
	2	Use cases	
	2.1	Application Use Cases	
	2.1.1	Person Detection	
	2.1.2	Semantic Segmentation	
	2.1.3	Skeleton Detection	
	2.1.4	Face Recognition	
	2.1.5	Facial Landmark Detection	
	2.1.6	Style Transfer	
	2.1.7	Super Resolution	
	2.1.8	Image Captioning	
	2.1.9	Text-to-image	
	2.1.10	Machine Translation	
	2.1.11	Emotion Analysis	
	2.1.12	Video Summarization	
	2.1.13	Noise Suppression	
	2.1.14	Speech Recognition	
	2.1.15	Text Generation	
	2.1.16	Detecting fake video	
	2.2	Framework Use Cases	
	2.2.1	Custom Layer	
	2.2.2	Network Concatenation	
	2.2.3	Performance Adaptation	
	2.2.4	Operation Level Execution	
	2.2.5	Integration with real-time video processing	
←	3	Security Considerations	
	3.1	Guidelines for new operations	
←	4	Privacy Considerations	
	5	Ethical Considerations	

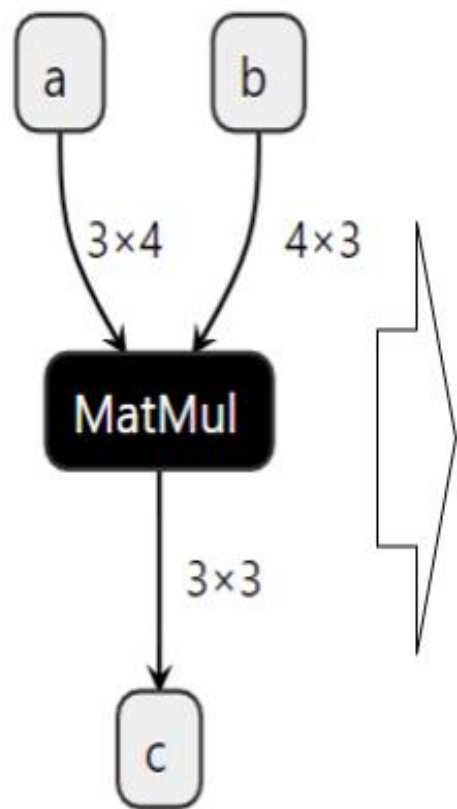
WebNN Programming Model



*WebNN next features

WebNN brings a unified abstraction of neural networks to Web

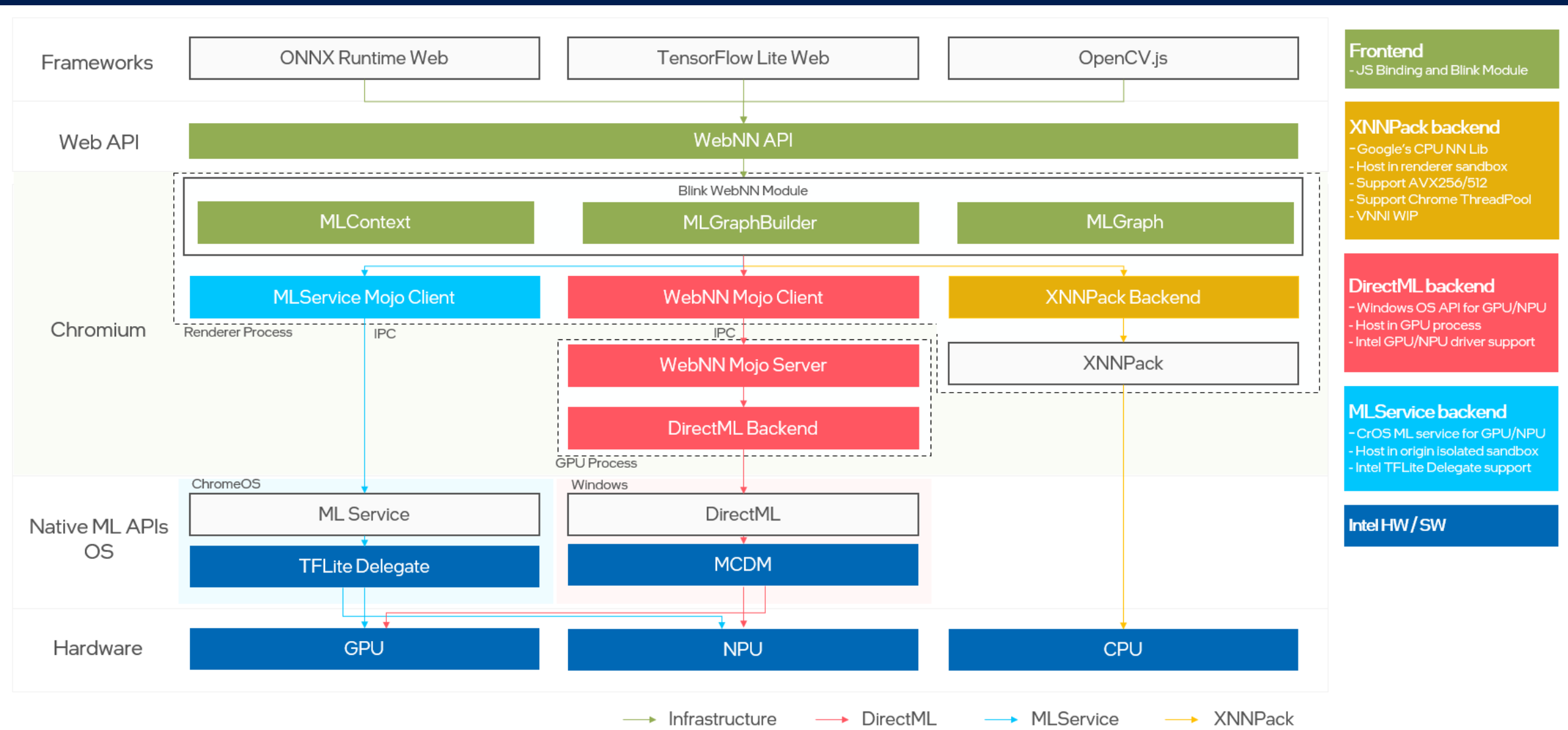
Hello Tensors



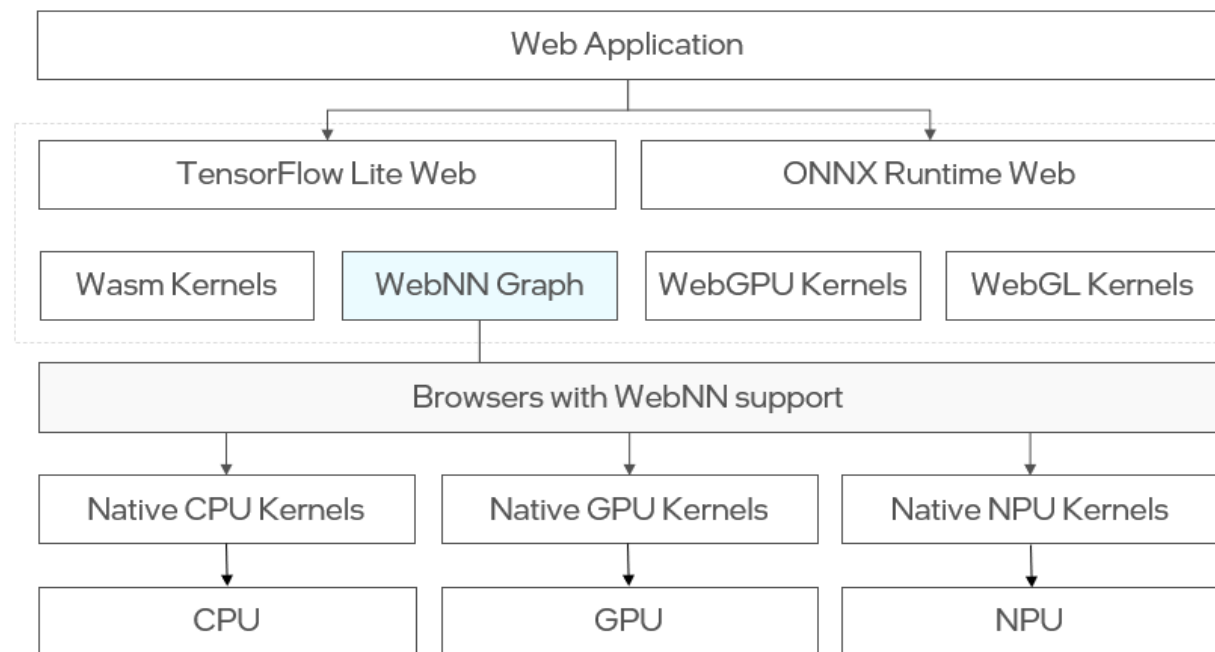
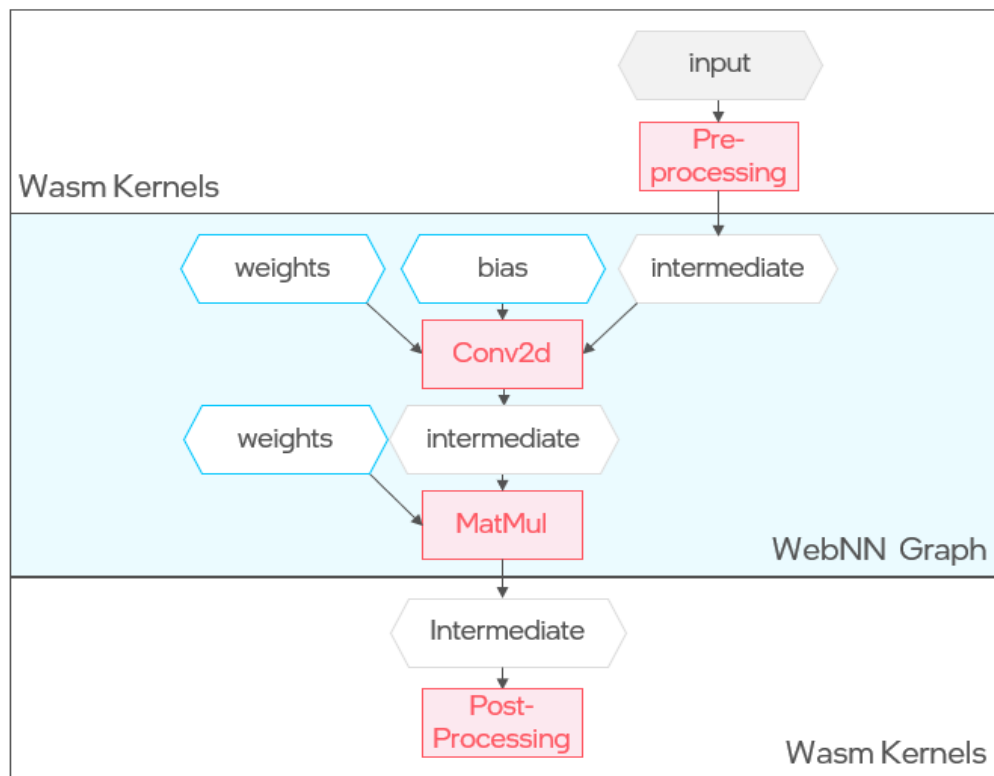
$$a \times b = c$$

```
1  const context = await navigator.ml.createContext();
2  const builder = new MLGraphBuilder(context);
3  // 1. Create a computational graph 'c = a * b'.
4  const a = builder.input('a', {type: 'float32', dimensions: [3, 4]});
5  const b = builder.input('b', {type: 'float32', dimensions: [4, 3]});
6  const c = builder.gemm(a, b);
7  // 2. Compile it into an executable.
8  const graph = await builder.build({c});
9  // 3. Bind input and output buffers to the graph and execute.
10 const bufferA = new Float32Array(3*4).fill(1.0);
11 const bufferB = new Float32Array(4*3).fill(0.8);
12 const bufferC = new Float32Array(3*3);
13 const results = await context.compute(graph, {'a': bufferA, 'b': bufferB}, {'c': bufferC});
14 // 4. Print the results.
15 console.log(`values: ${results.outputs.c}`);
```

WebNN Browser Implementations



WebNN JS ML Frameworks Integration

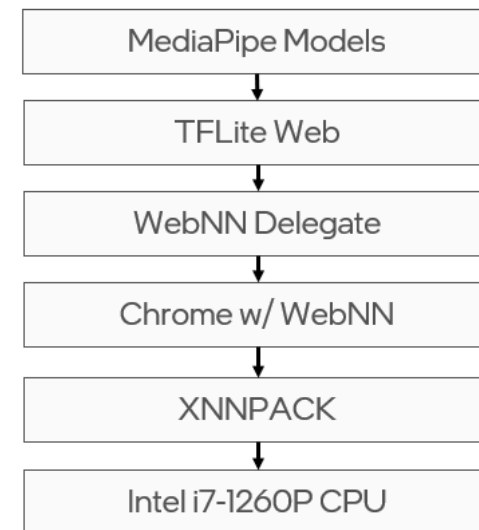
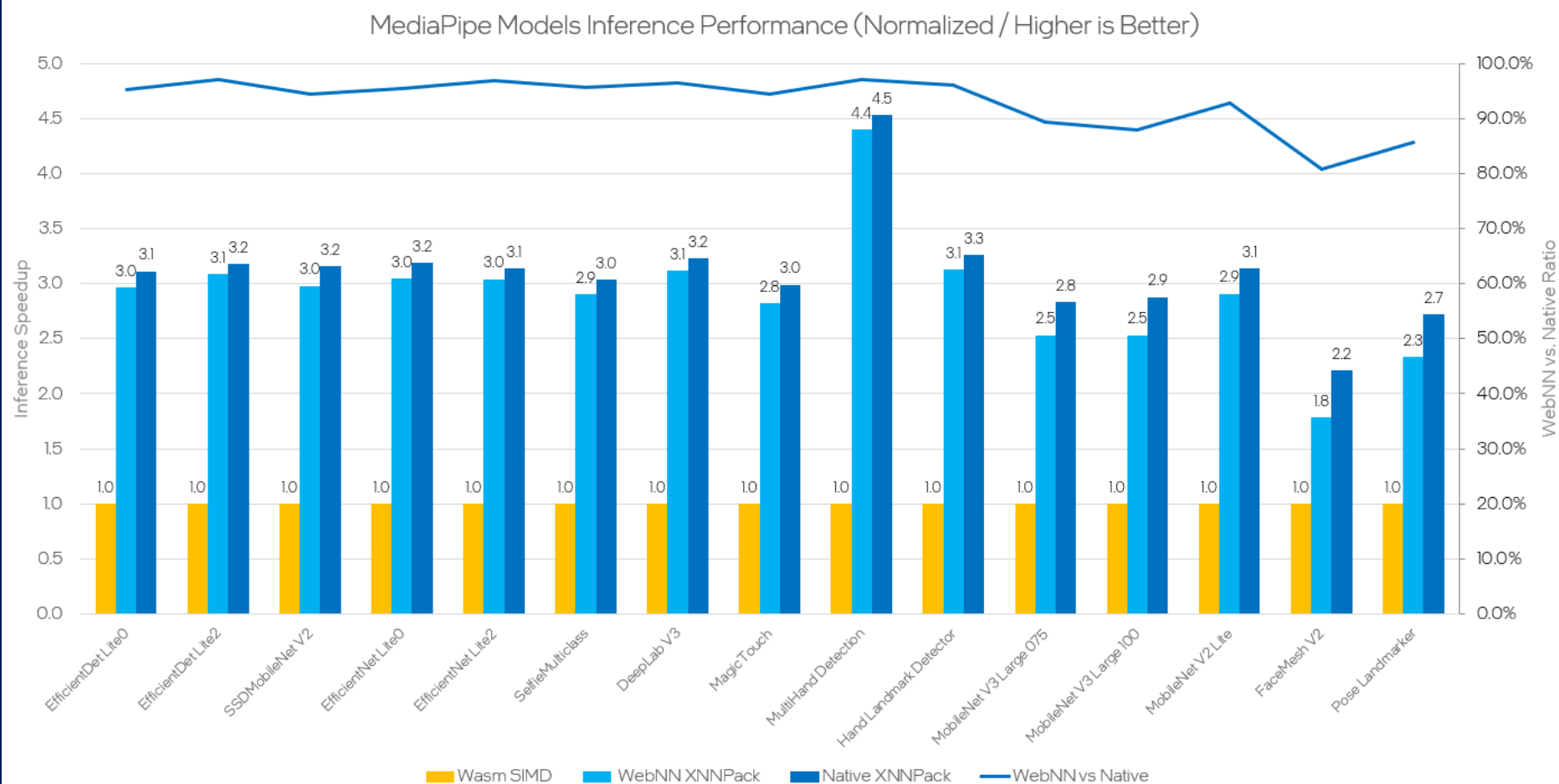


Integration Status

 ONNX	Landed
 TensorFlow Lite	Prototype Available

Mainstream ML frameworks are integrating WebNN

MediaPipe Models Performance using WebNN CPU

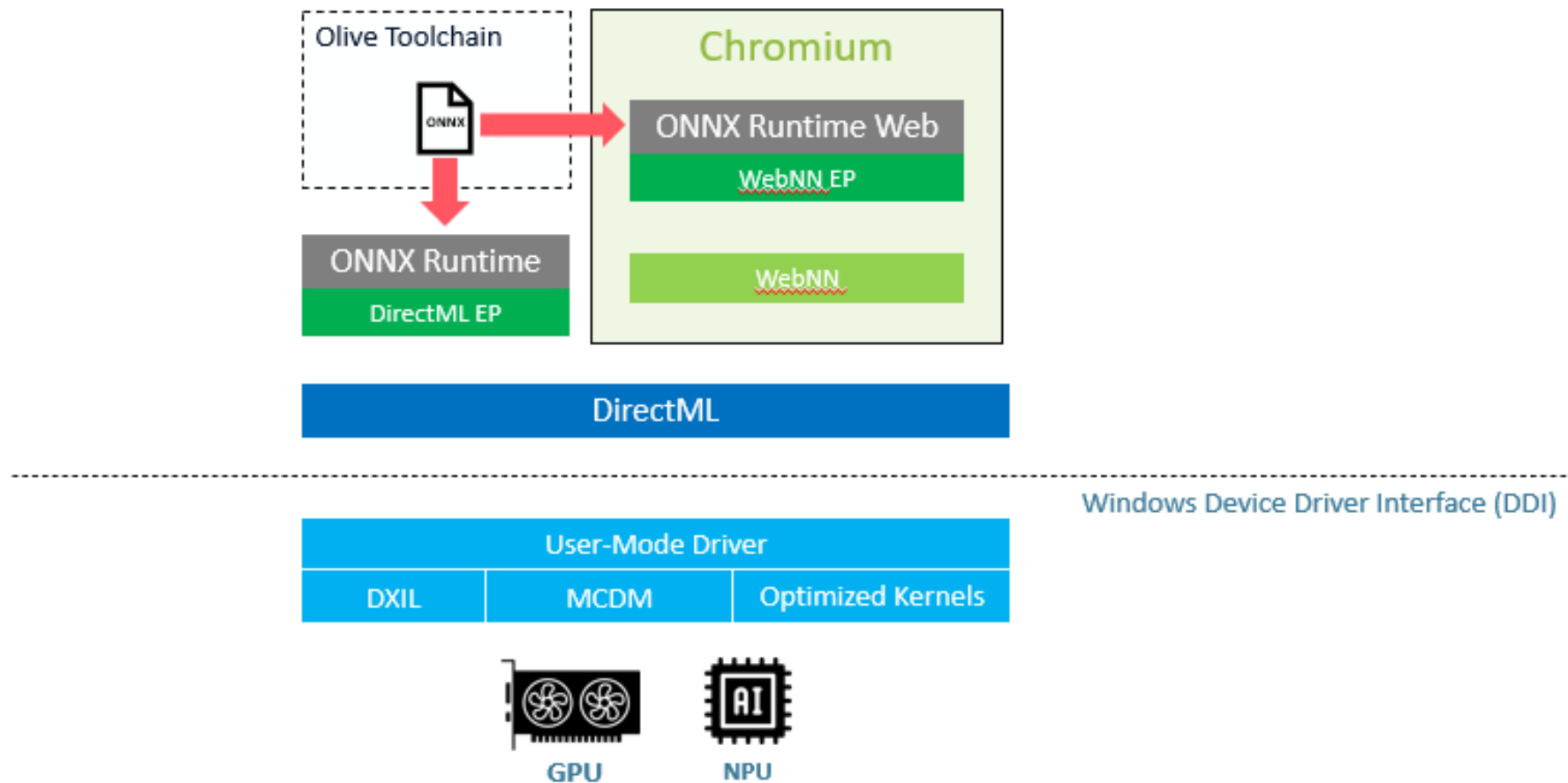


The benchmark stack

- Browser: Chrome Canary 118.0.5943.0
- DUT: Dell/Linux/i7-1260P, single p-core
- Workloads: [MediaPipe solution models](#) (FP32, batch=1)

WebNN delivers near-native inference performance on CPU

ONNX Runtime DirectML Architecture



- DXIL: [DirectX Intermediate Language](#)
- MCDM: [Microsoft Compute Driver Model](#)
- Optimized Kernels: Driver implementation of specific custom ML operations

DirectML Use Cases

USER: Can you describe the city of Seattle in the summertime in the style of Comedian Trevor Noah?

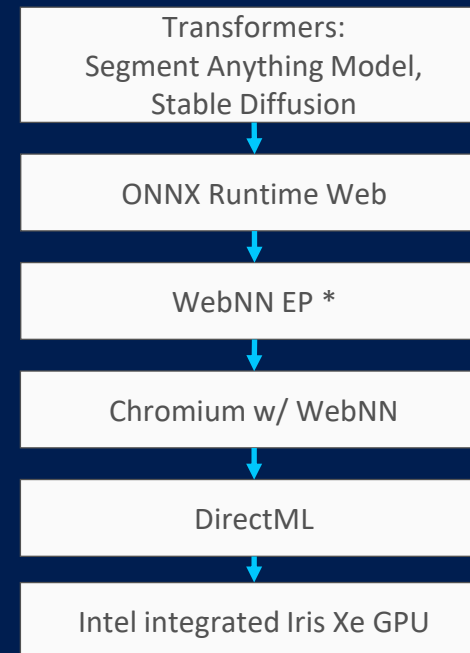
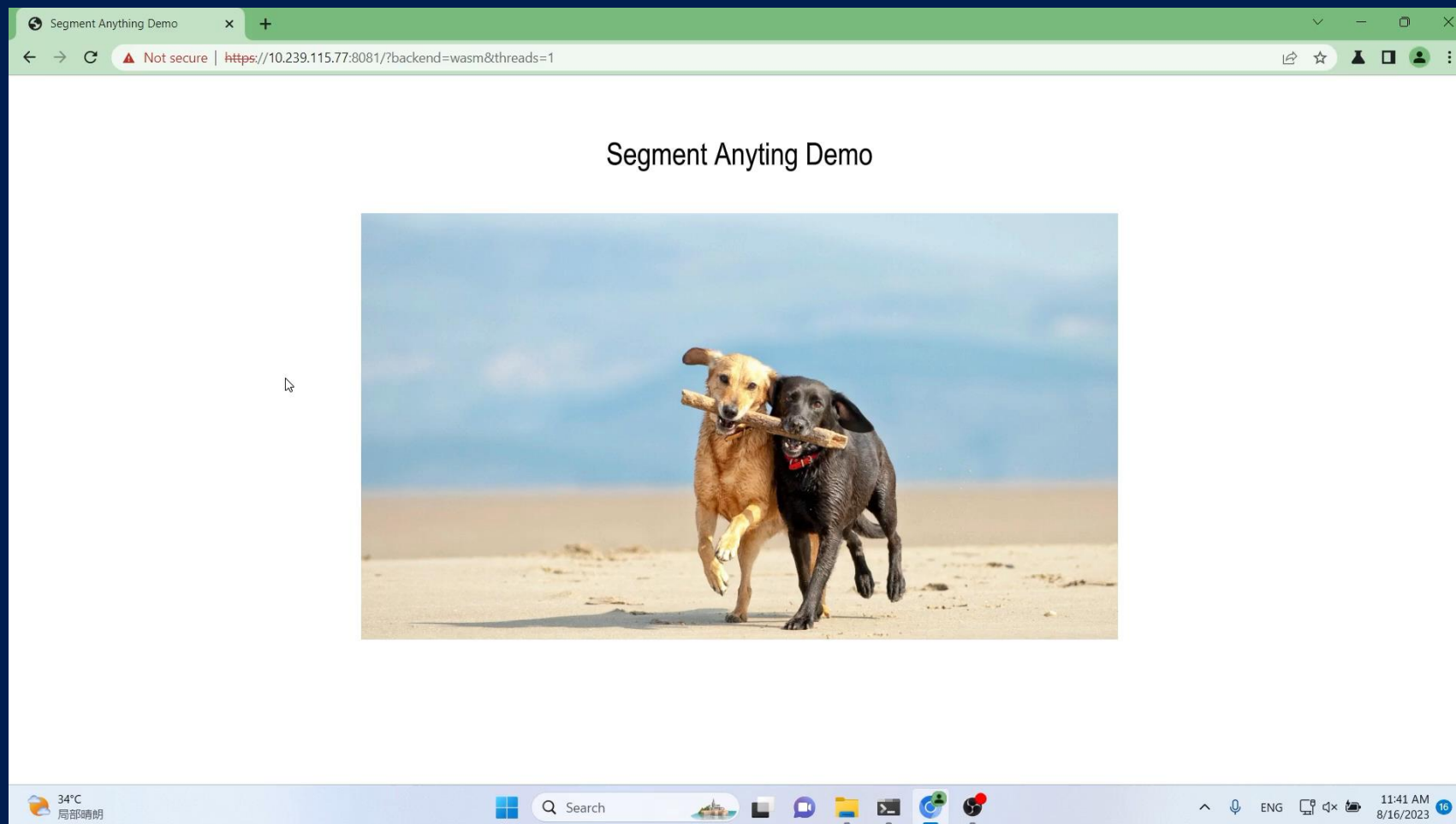
Dolly:

Sure! Seattle in the summertime is all about the outdoors. Every morning when you wake up the weather report says it will be in the 60's. And for the first half of the day you just feel grateful that it is not raining. Then you go outside and it is 106 degrees. Seattleites all have a number they call "the Noah Index". 1000 is freezing, 2000 is tolerable, 3000 is pleasantly warm, 4000 is a blistering sun burn, 5500 is only slightly less pleasurable than 4000, and we're talking about thousands of people there, so the Noah Index gets really big really fast. Seattleites all enjoy the outdoors. We're surrounded by mountains and lakes so we appreciate the great outdoors. We love to go camping and hiking. And when it is hot we love to go to the unnecessary and un-needed Dose of Mountains concert. We've all been to the Dose of Mountains concert, don't get me started on how much we love Dose of Mountains, they are a Olympian institution. And they go on every Tuesday through the end of July. We love our mountains and lakes, and we love being outside. Even in the rain.

End



Transformers on WebNN DirectML Demo



The demo stack

* WebNN Execution Provider of ONNX Runtime Web
Segment Anything (ONNX/FP32) model:
Size: 19.5MB
Number of parameters: 5.11M

WebNN transformers demo guides the WebNN next version work

ONNX Runtime Web Code Samples with WebNN

```
import { InferenceSession } from "onnxruntime-web";

// Initialize the ONNX model
const initModel = async () => {
  env.wasm.numThreads = 1; // 4
  env.wasm.simd = true;
  env.wasm.proxy = true;
  const options: InferenceSession.SessionOptions = {
    // provider name: wasm, webnn
    // deviceType: cpu, gpu, npu
    // powerPreference: default, high-performance
    executionProviders:
      [{ name: "wasm" }], // WebAssembly CPU
  }
  // ...
};
```

WebAssembly backend

```
import { InferenceSession } from "onnxruntime-web";

// Initialize the ONNX model
const initModel = async () => {
  env.wasm.numThreads = 1; // 4
  env.wasm.simd = true;
  env.wasm.proxy = true;
  const options: InferenceSession.SessionOptions = {
    // provider name: wasm, webnn
    // deviceType: cpu, gpu, npu
    // powerPreference: default, high-performance
    executionProviders:
      [{ name: "webnn", deviceType: "gpu", powerPreference: 'default' }],
  }
  // ...
};
```

WebNN backend

Switching to WebNN can be done by modifying a single line of code

WebNN ChromeOS MLService on NPU

WebNN Image Classification Demo

Backend: Wasm (CPU) | WebGL (GPU) | **WebNN (NPU)** | WebNN (GPU)

Model: MobileNet V2 | SqueezeNet | **ResNet V2 50**

IMAGE | LIVE CAMERA

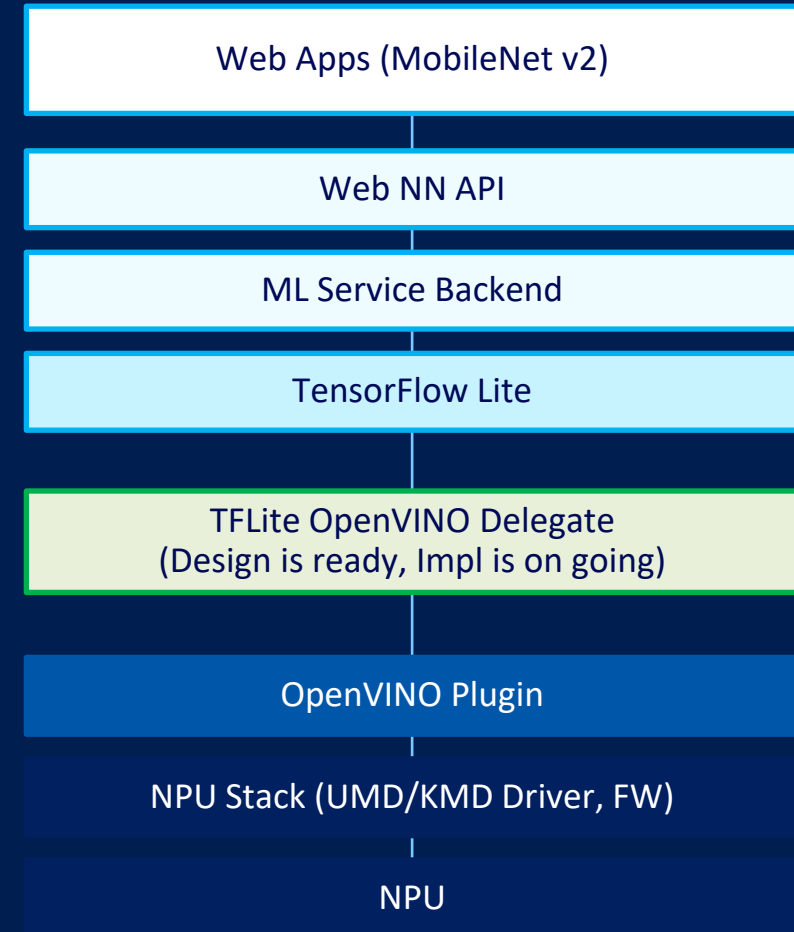
Load time: 791.30 ms | Build time: 1769.80 ms | Median inference time: **4.50 ms**

#	Label	Probability
1	Labrador retriever	93.70%
2	golden retriever	2.85%
3	basset	0.43%

Pick Image

WEBNN API | SUPPORTED | W3C SPEC | USE CASES

©2023 WebNN API



Call to Action

- **Try WebNN CPU/GPU acceleration on Google Chrome and Microsoft Edge Canary today**
 - Navigate to **about:flags** in the URL bar and turn on “Enables WebNN API”
 - Visit https://webmachinelearning.github.io/webnn-samples/image_classification/
 - Visit https://webmachinelearning.github.io/webnn-samples/semantic_segmentation/index.html
- **Join Intel and other industry leaders, including Microsoft and Google, participate in WebNN standardization and open-source implementation**

References

- WebNN Spec: <https://www.w3.org/TR/webnn/>
- WebNN Explainer: <https://github.com/webmachinelearning/webnn/blob/main/explainer.md>
- WebNN Implementation Status: <https://webmachinelearning.github.io/webnn-status/>
- WebNN Samples: <https://webmachinelearning.github.io/webnn-samples/>
- WebNN Image Classification: https://webmachinelearning.github.io/webnn-samples/image_classification/
- WebNN Semantic Segmentation: https://webmachinelearning.github.io/webnn-samples/semantic_segmentation/index.html
- ONNX Runtime WebNN Execution Provider: <https://github.com/microsoft/onnxruntime/tree/main/onnxruntime/core/providers/webnn>
- Developing Web-Based AI Apps for the Client: https://static.rainfocus.com/intel/innv2023/sess/1689805601141001Xyiy/supmat/DevelopingWebBasedAIApps_Final_1694887505835001jjpL.pdf

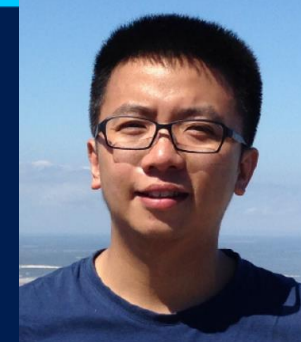
Web Machine Learning Working Group Launch

Intel

Moh Haghghat, Ningxin Hu

Microsoft

Chai Chaoweeraprasit



Notices and Disclaimers

For notices, disclaimers, and details about performance claims, visit www.intel.com/PerformanceIndex or scan the QR code:



© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries. Other names and brands may be claimed as the property of others.

intel[®] Ai
summit

Thank You!

