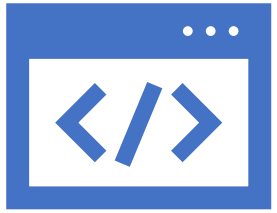
Modern PWA

A winning combo for the best client experience

Kenneth Christiansen, Anssi Kostiainen

intel Vision



The web is an unquestioned Key Application Platform

Offer the core building blocks for application development

Achieve great, near native-like performance

Offer capabilities and common features available to native apps

The unique qualities of the web, strengthened

Remain cross platform to work across all desktop operation systems

More responsive - load instantly, be responsive from the get-go

Meet and exceed the user safe environment needs

The web is on the right path

- The Web is becoming an unrivaled Application Development Platform

- Solid application building blocks
  - Web App Manifest and Service Workers
  - CSS Grid, Flexbox, Container Queries
  - Design Systems powered by Web Components
  - URLPattern, Navigation API and Shared Element Transitions

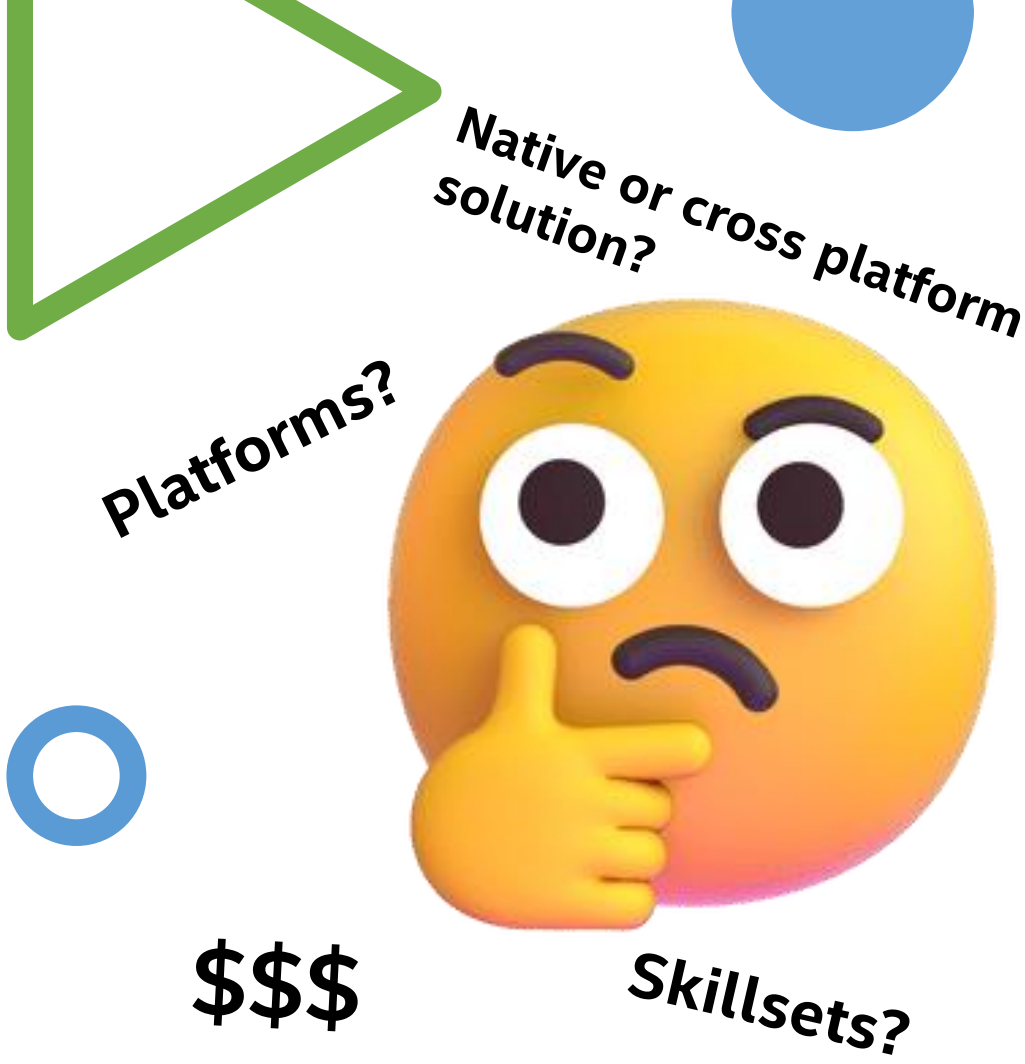
- Relentless focus on performance
  - Web Assembly, SIMD optimizations
  - MediaPipe with native support for framing, background blurring etc.
  - Native support for machine learning

- Innovating on capabilities via Project Fugu, e.g., File System Access

Why do we care?

Platforms/ecosystems are only as important as the experience they bring

The cost of creating high quality apps for the target audience often a balancing act



Today it is often mobile first with a web based (or hybrid solution) for the desktop

A winning strategy

JS Developers – 17.4 M[1]
Web app & front end
Web framework & runtime

60% of
Dev community

60-65%[2] **of the time spent on PC is spent browsing the web**

~65%
Time spent on Web

Rise of modern web fueled by PWA[3]

Growth of available PWAs in 2021 — 270%

CAGR 2021-27 — 31.9%

Google & MSFT leading the way

- Cross platform with great support on desktop OSes like Windows and ChromeOS
- Most browsers are built around open-source projects which we can contribute to
- Web features are being standardized in the open by interested parties
- Interoperability and backwards compatibility ensure the platform keeps evolving

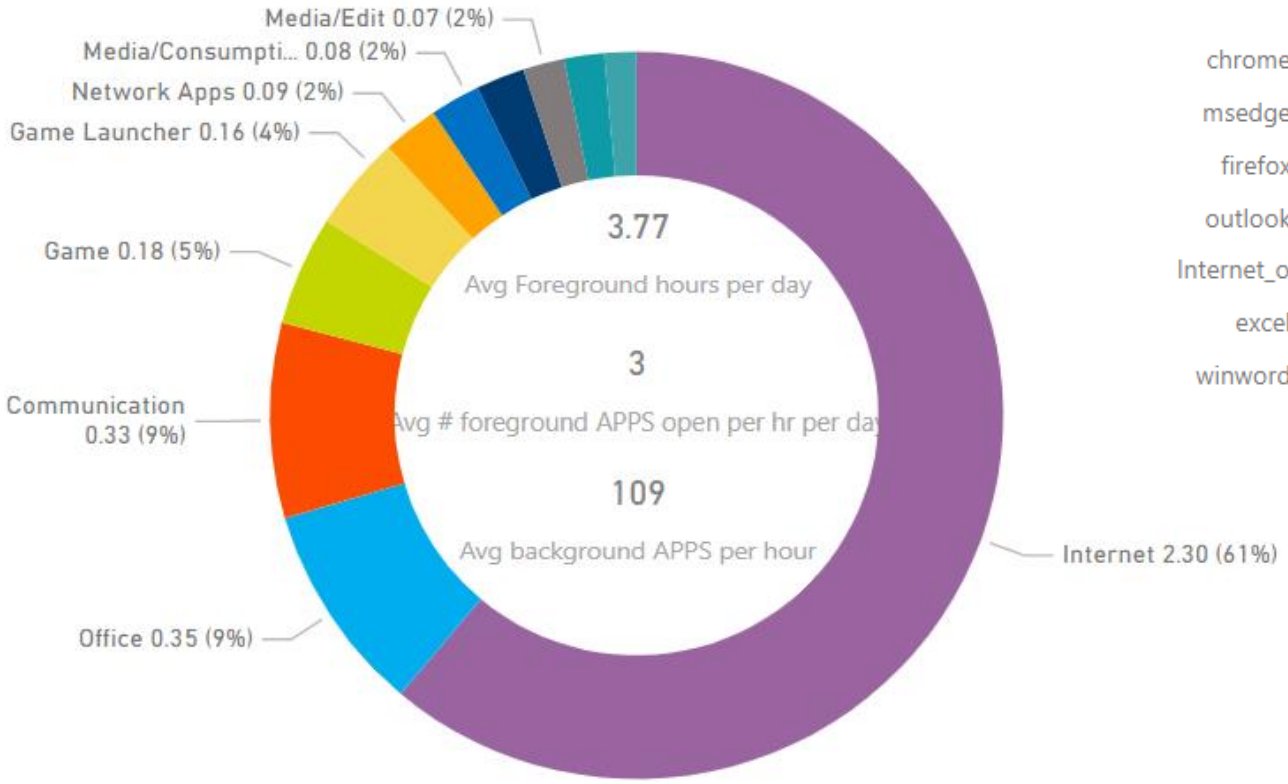
PWA

[1] Source: Slashdata – Dev Nation '22,
[2] Source: Intel DCA, Microsoft
[3] https://firt.dev/pwa-2021, https://www.emergenresearch.com/industry-report/progressive-web-application-market

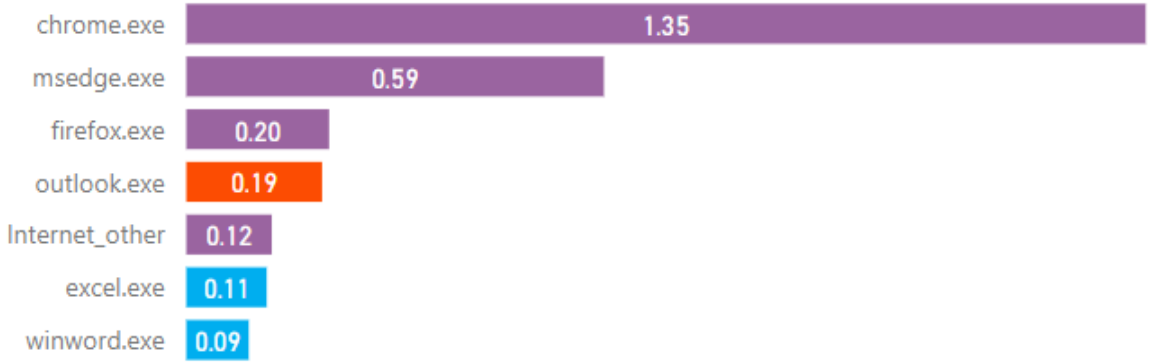
intel

Web & Progressive Web Apps dominate PC users' time†

Application Usage by Category



Browsers are top apps on PC*



Web

Source: Intel DCA
July 2022
~8M PCs WW

Weekday: **61%**

Weekend: **66%**

† Source: Intel DCA, July 2022

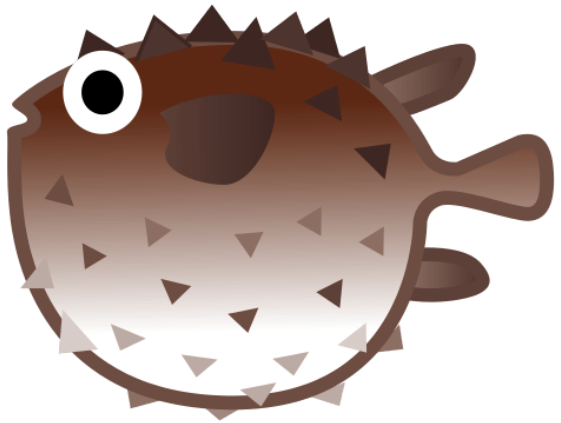
Almost 2/3rd of PC cycles worldwide are spent on the web!

What makes up a delightful UX?

Performance and capabilities + an enjoyable user experience

- Core capabilities for desktop apps are **now** built in
  - **Seamless** copy and paste
  - **Frictionless** access to local files
  - **Safe access** to external hardware for education, hobbyists, and enterprise usage

Project Fugu effort

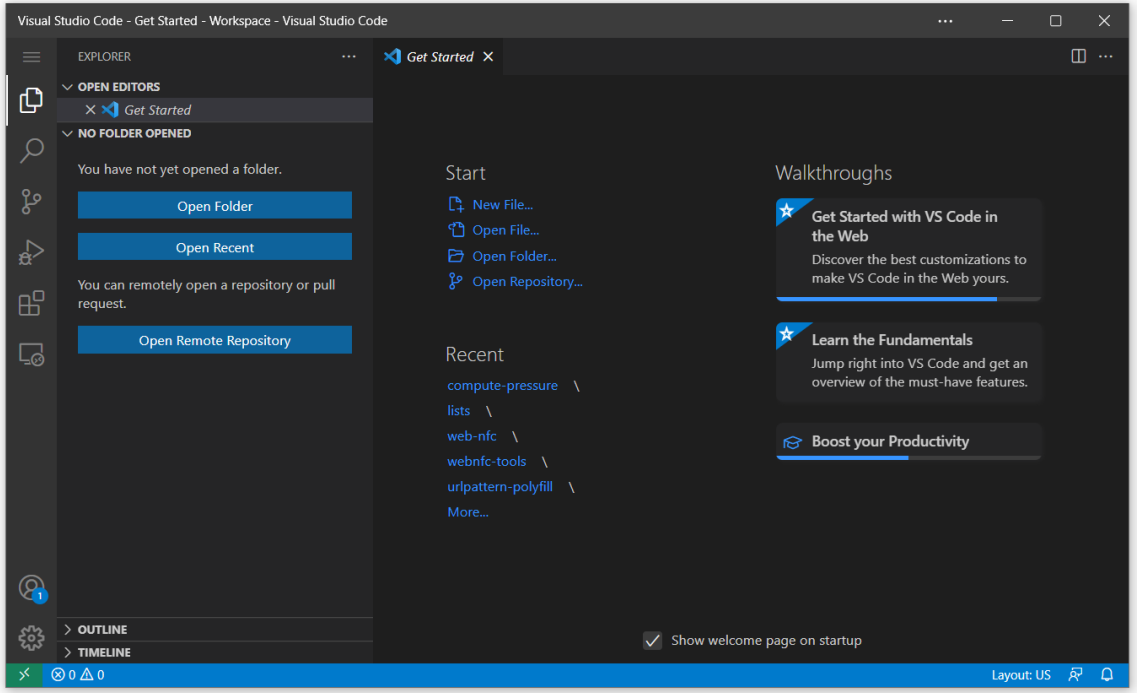


Enabling new developer experiences

Visual Studio Code

Visual Studio Code for the Web

Visual Studio Code for the Web provides a free, zero-install Microsoft Visual Studio Code experience running entirely in your browser, allowing you to quickly and safely browse source code repositories and make lightweight code changes. To get started, go to https://vscode.dev in your browser.



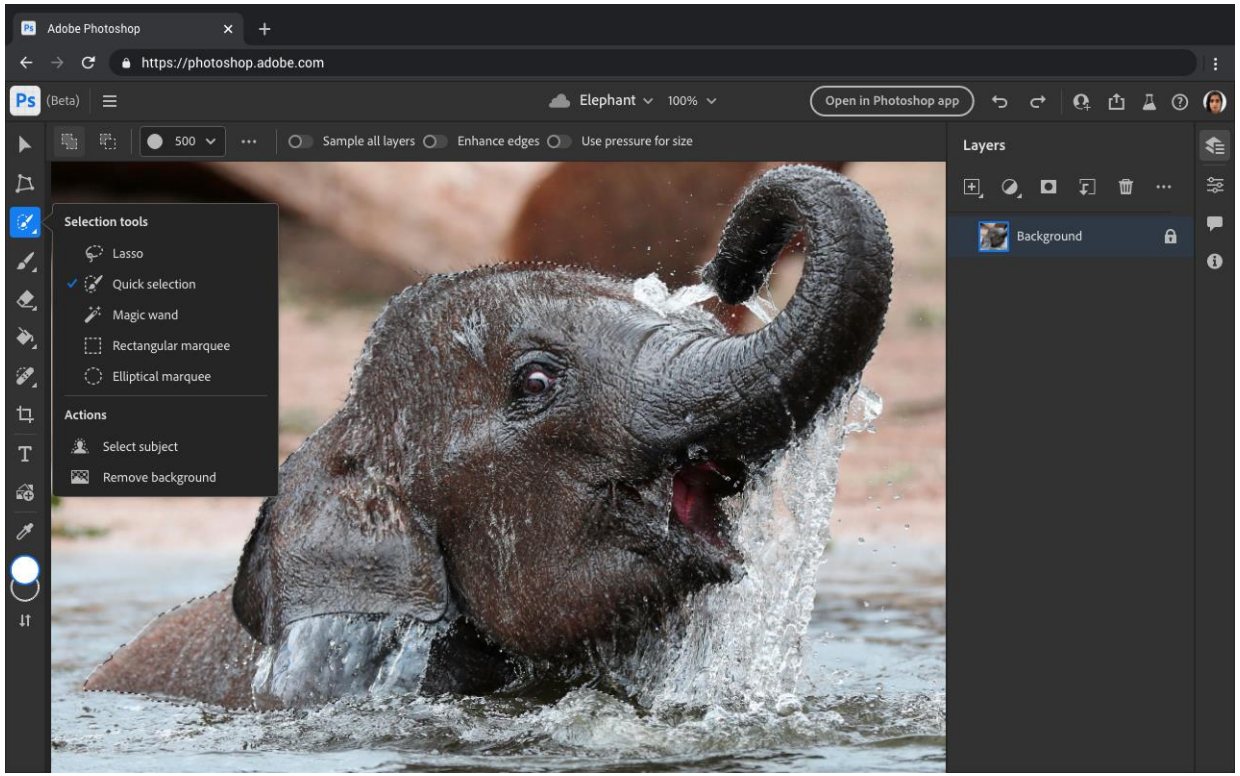
Made possible with File System Access API

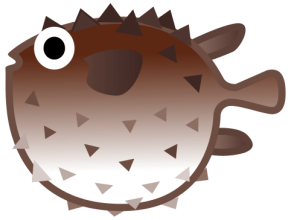
intel

Give new life to legacy apps

Made possible with a set of new APIs

- High performance storage - Origin Private File Systems provide highly optimized in-place and exclusive write access
- Web Assembly to bring existing C++ code to the web
- Dynamic Multithreading support for Web Assembly
- SIMD - Halide is essential to Adobe's performance and it provides a 3-4× speedup on average and in some cases an 80-160× speedup.
- P3 colorspace support for canvas
- Web Components



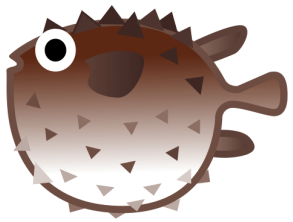


Connect with external devices

iRobot

Introducing the Root® robots for coding, discovery, and play.

Creative and seriously fun, a love for coding is one of the best gifts you can give a child. The iRobot® Root® coding robots make learning to code easy and natural in any environment, at home or at school.

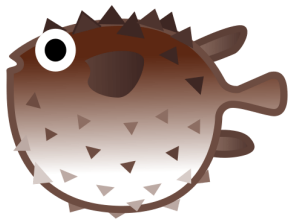


intel

Safety and privacy

- Native apps offers “fake safety”, but users use just a few apps
  - Signing, app store approval, yet often own installers (Windows)
  - Very powerful direct access to many native APIs without user approval
  - Users install a limited set of apps, but browse many web sites/apps per month

- Web APIs are designed with safety and privacy in mind
- Hard problems, but we are constantly innovating and refining our approaches



intel Current Focus

Capabilities

Bridge the gap between the web and native

No silicon left behind as the desktop moves to the web

Performance

Achieve near-native performance

Optimize the use of Intel silicon

New and upcoming APIs to improve performance

WebAssembly

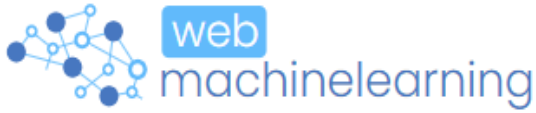
Fast CPU execution,
SIMD + MT support

WebGL

Legacy GPU execution,
97.6% device support

WebGPU

Modern GPU execution
3.7x faster than WebGL*



Web Neural Network

CPU/GPU/VPU execution
Near-native performance

*Geomean of TensorFlow.js Model Benchmark (https://tensorflow.github.io/tfjs/e2e/benchmarks/local-benchmark/index.html), Machine config

intel.

Accelerating Wasm performance

128bit SIMD

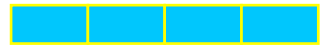
128bit SIMD

256bit SIMD

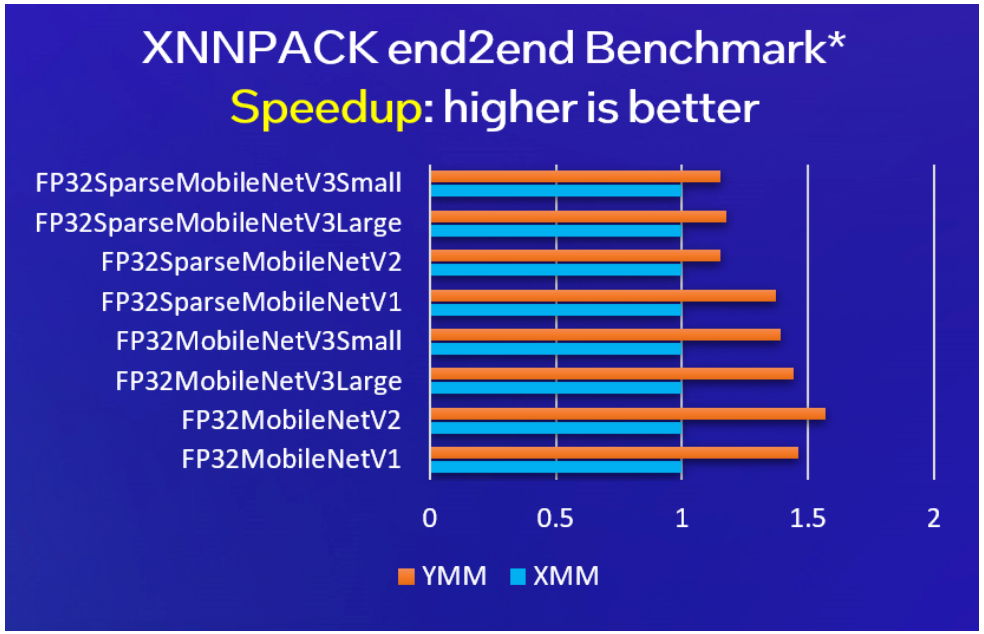
```
 8  vmulps xmm9,xmm8,xmm9
 9  vaddps xmm3,xmm9,xmm3
10  vmulps xmm9,xmm8,xmm10
11  vaddps xmm2,xmm9,xmm2
12  vmulps xmm9,xmm8,xmm11
13  vaddps xmm0,xmm9,xmm0
14  vmulps xmm8,xmm8,xmm12
15  vaddps xmm1,xmm8,xmm1
..  ...  ...
31  vmulps xmm14,xmm13,xmm9
32  vaddps xmm7,xmm14,xmm7
33  vmulps xmm14,xmm13,xmm10
34  vaddps xmm6,xmm14,xmm6
35  vmulps xmm14,xmm13,xmm11
36  vaddps xmm5,xmm14,xmm5
37  vmulps xmm13,xmm13,xmm12
38  vaddps xmm4,xmm13,xmm4
```

```
64  vmulps ymm7,ymm8,ymm7
65  vaddps ymm3,ymm7,ymm3
66  vmulps ymm6,ymm8,ymm6
67  vaddps ymm2,ymm6,ymm2
68  vmulps ymm5,ymm8,ymm5
69  vaddps ymm1,ymm5,ymm1
70  vmulps ymm4,ymm8,ymm4
71  vaddps ymm0,ymm4,ymm0
```

SSE: 128 bit



AVX: 256 bit

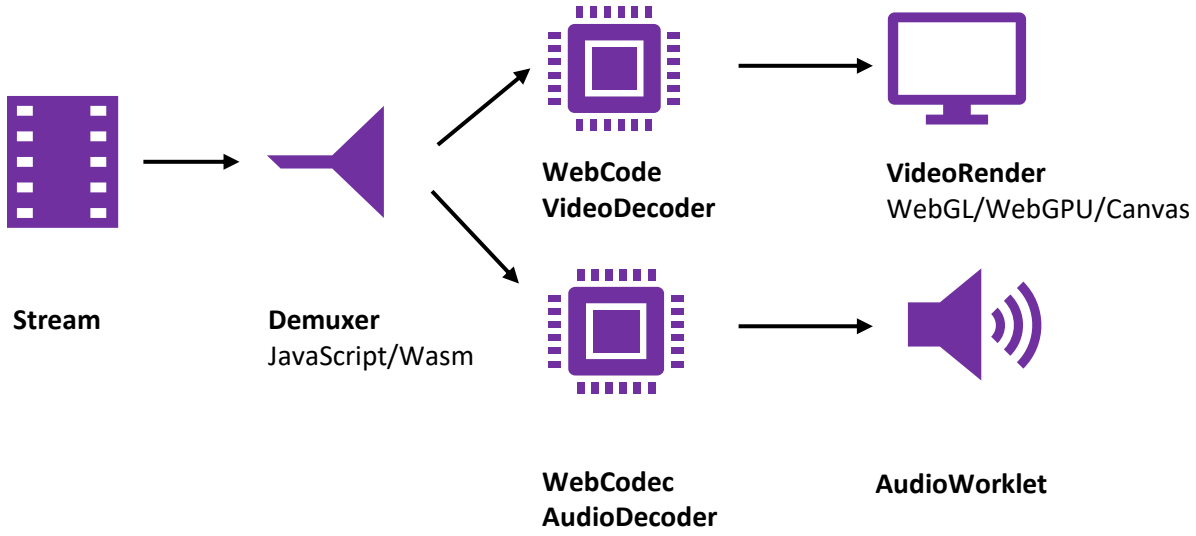


Convert 128-bit Wasm SIMD instructions into 256-bit IA instructions dynamically

* XNNPACK end2end Benchmark (https://github.com/google/XNNPACK/blob/master/bench/end2end.cc) Machine Config: TGL-RVP, Ubuntu 20.04.1 LTS

intel

Accelerating media processing



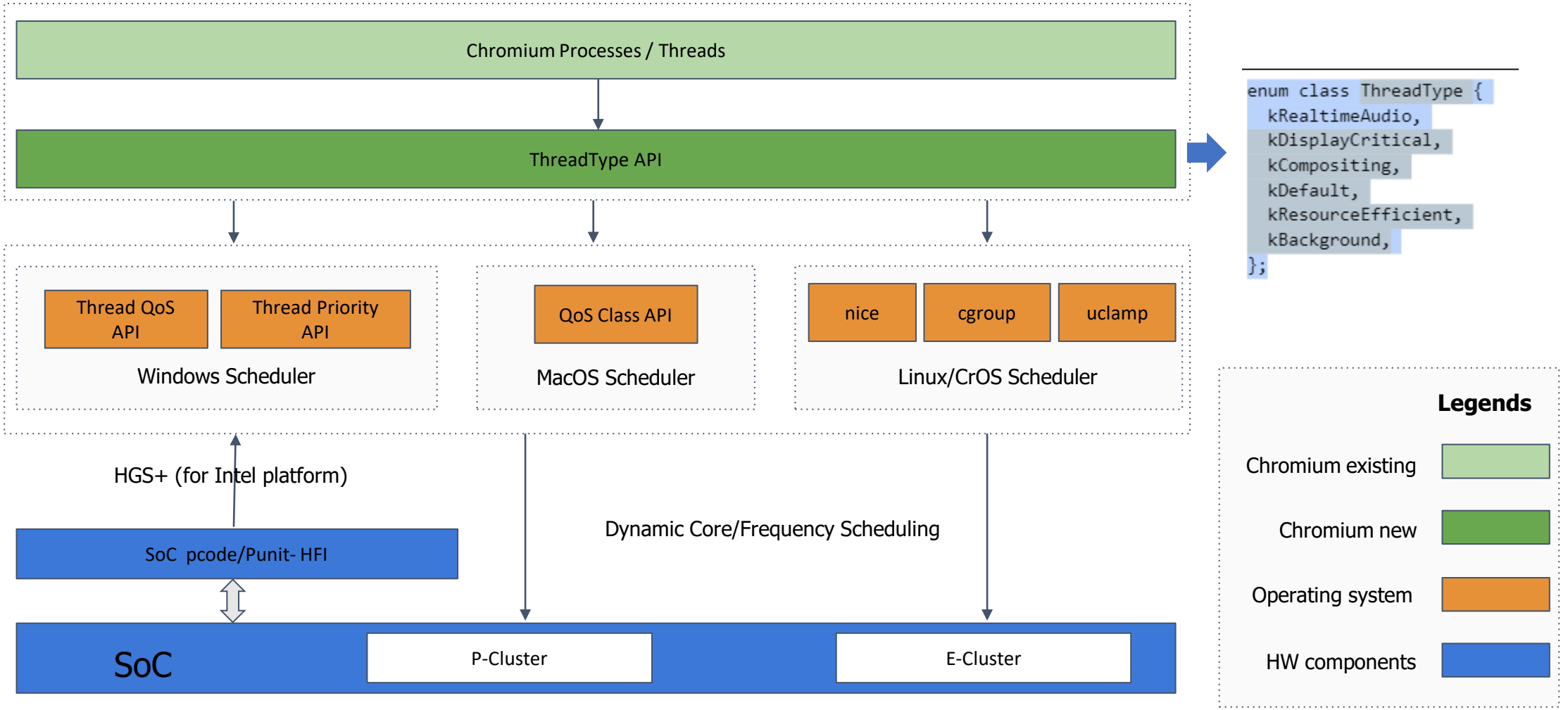

```
// WebCodecs encoder example: vp9
const init = {
  output: output_cb,
  error: error_cb,
};

const config = {
  codec: "vp09.00.10.8",
  width: 1280,
  height: 720,
  bitrate: 10e6,
  framerate: 30,
  acceleration: "prefer-hardware",
  scalabilityMode: "L1T3",
};

let encoder = new VideoEncoder(init);
encoder.configure(config);
encoder.encode(video_frame);
```

Enables hardware encoders/decoders via WebCodec API

- Video formats: AV1, AVC1, VP8, VP9, HEVC
- Audio formats: MP3, MP4a, Opus, Vorbis, ULAW, ALAW
- Supports HDR (High Dynamic Range) on Intel

intel.

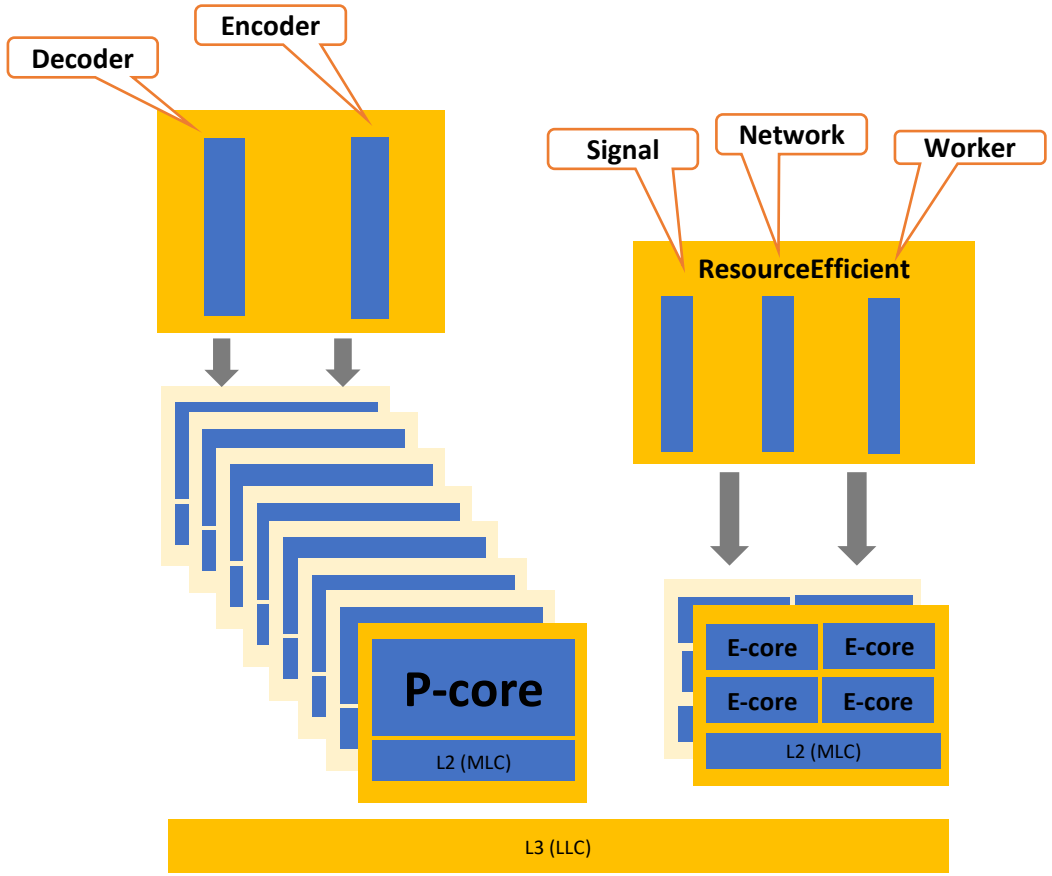
Improving power efficiency w/ Intel hybrid-core



Reduce PWA power by tagging threads/tasks with their roles instead of priority

intel.

Improving power efficiency for video conferencing

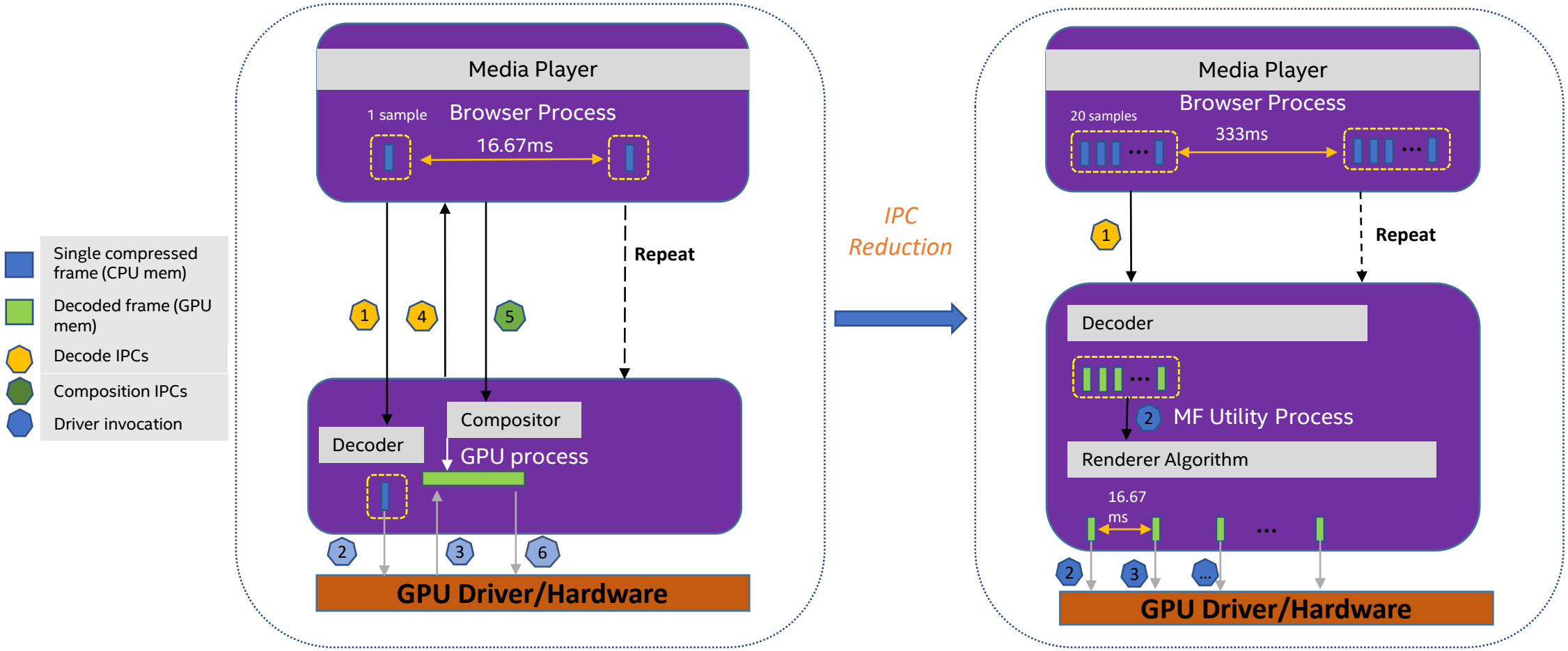


- Assign threads with different ThreadTypes
  - TheadPriority: desired task starting time
  - ThreadQoS: desired task completion time
- Assign WebRTC signal/network/worker threads with **`ResourceEfficient`** Type
  - Scheduled frequently
  - Less sensitive to latency
  - Less computation heavy
- **`ResourceEfficient`** threads will be scheduled to **E-Cores** whenever possible

100+mW power savings achieved for video call on Windows (w/ 12th Gen Intel Core)

intel.

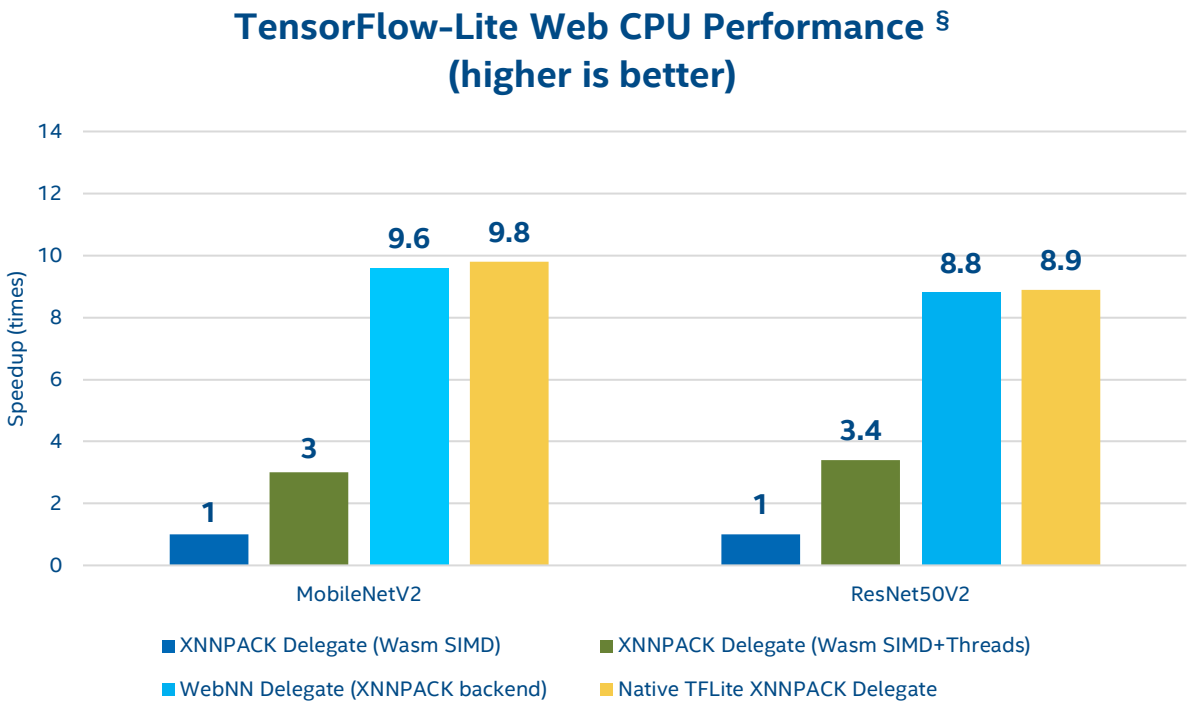
Video playback power-efficiency improvements

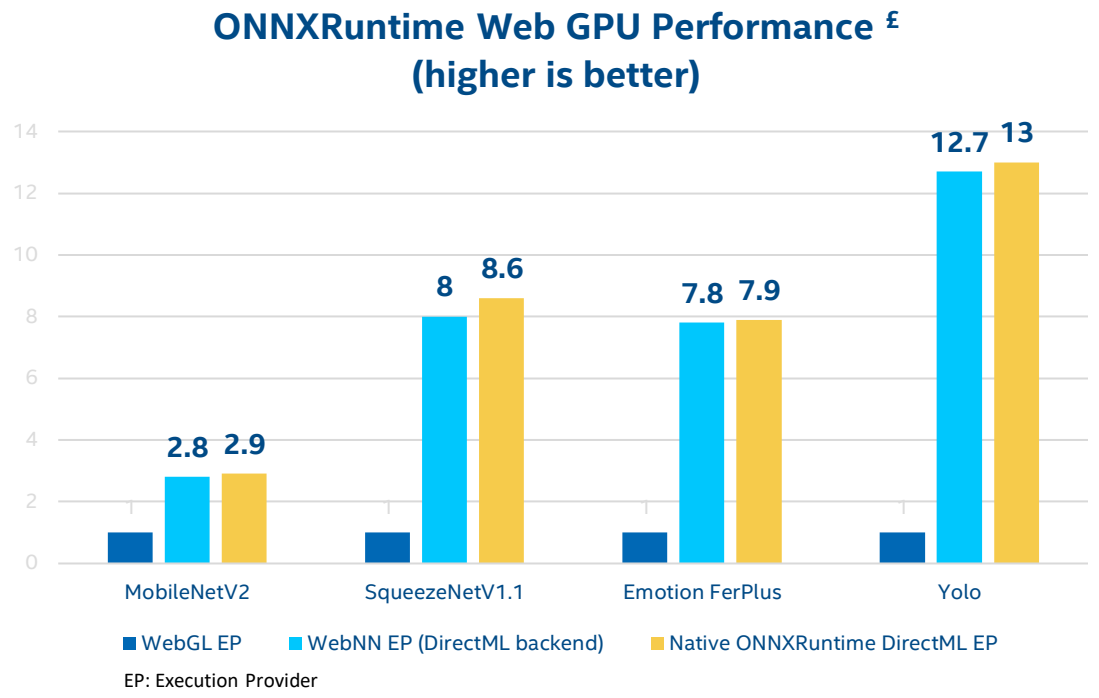


10% SoC power saving for video playback on Windows (w/ 12th Gen Intel Core)

intel.

WebNN – JavaScript* ML Framework





WebNN delivers near-native Power and Perf characteristics thanks to efficient paths to the HW capabilities & features

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

† All models are FP32, batch size 1, Tested on ADL Laptop DELL Vostro 5620, CPU: 12th Gen Intel(R) Core(TM) i7-1260P 4 p-cores / 4.70 GHz, 8 e-cores / 3.40 GHz, OS: Windows 11 Professional latest (21H2) .

§ Performance data was tested on Chromium WebNN/XNNPACK Prototype by running TensorFlow.js end-2-end model benchmark.

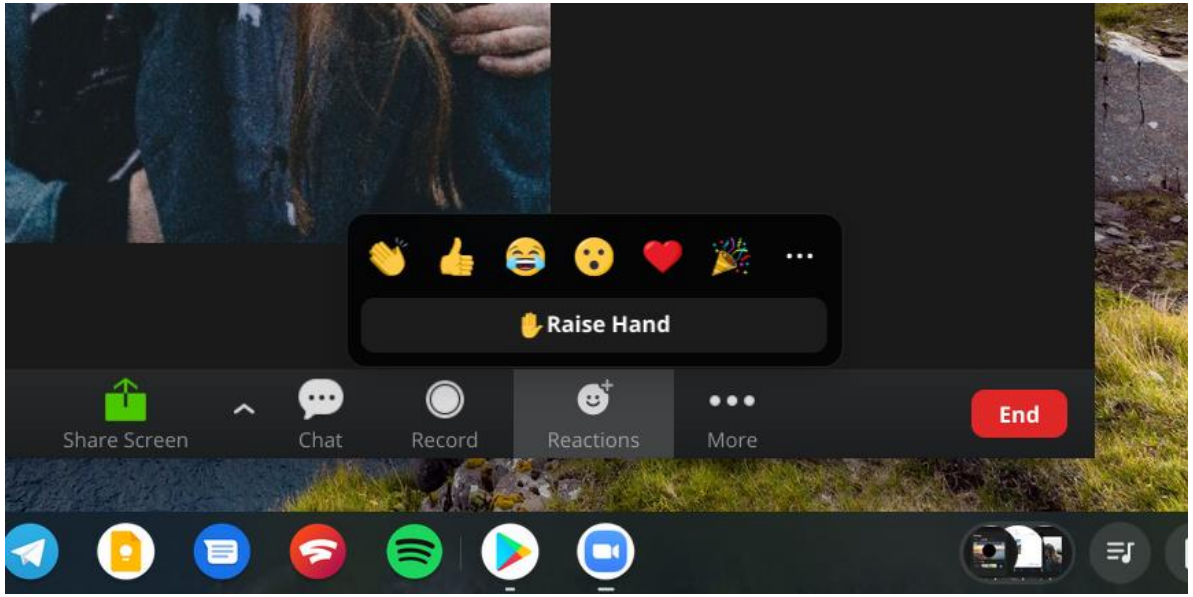
£Performance data was tested on Electron.js with WebNN-native node.js add-on by running ONNXRuntime Web Demo.

Zoom PWA

PWA technology enables modern Zoom experiences on ChromeOS and on any browser

- Leverages Web Codecs for full control over media processing, incl. hardware acceleration
- New web capabilities led by Intel to enhance these experiences:
  - Background blurring
  - Face detection / auto-framing
  - Eye gaze correction
  - Lighting correction
  - Noise suppression
  - Compute Pressure

https://pwa.zoom.us/wc



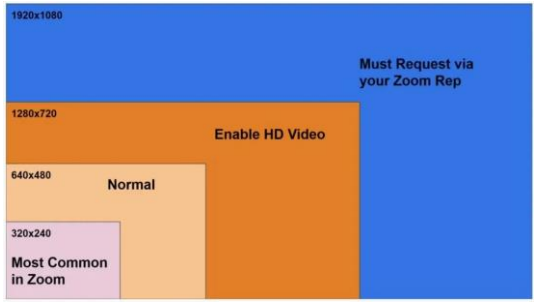
Compute Pressure API

Check if there's enough juice for the frills

Is the client device being throttled?

Gain insights into different kinds of system pressure, starting with CPU

Adjust number of video feeds



Adjust video resolution and frame-per-second

Skip feed filters and non-essentials like WebRTC noise suppression

Turn quality-vs-speed and size-vs-speed towards ”speed” in WebCodecs

High-level state changes

Abstract CPU stalls, temperature, other factors

Pressure states represents the minimal set of useful states that allows websites to react to changes in compute and system pressure with minimal degration in quality or service, or user experience.

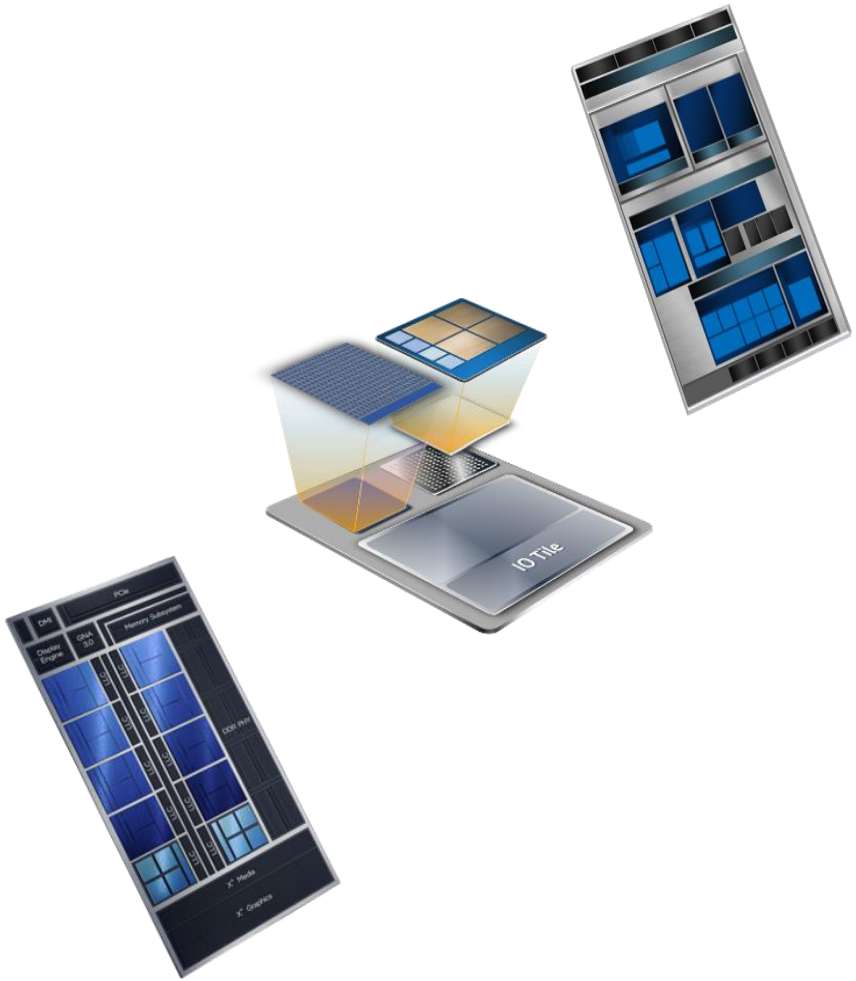
WebIDL

```
enum PressureState { "nominal", "fair", "serious", "critical" };
```

The ***PressureState*** enum represents the pressure state with the following states:

- **Nominal**: Current workloads are causing minimal pressure, allowing the system to run at a lower clock frequency to preserve power.
- **Fair**: The system is doing fine, everything is smooth and it can take on additional work without issues.
- **Serious**: There is some serious pressure on the system, but it is sustainable and the system is doing well, but it is getting close to its limits:
  - Clock speed (depending on AC or DC power) is consistently high
  - Thermals are high but system can handle it

  At this point, if you add more work the system may move into critical.
- **Critical**: The system is now about to reach its limits, but it hasn't reached *the* limit yet. Critical doesn't mean that the system is being actively throttled, but this state is not sustainable for the long run and might result in throttling if the workload remains the same. This signal is the last call for the web application to lighten its workload.



intel.

Example

```
function pressureChange(records, observer) {
  // For simplicity only look at last sample.
  const record = records.at(-1);
  switch (record.state) {
    case "critical":
      // Stop all unnecessary work.
      break;
    case "serious":
      // We are OK for now, but don't do additional work.
    default:
      // We are fine!
  }
}

const observer = new PressureObserver(pressureChange);
observer.observe("cpu");
```

web neural network

WA SIMD + Threads

WebGPU

Intel collaborates with the web ecosystem. Help us build the APIs *you* need:

- Tell us how we can make *your* web experience better, what are *your* pain points, we listen
- Adopt the new APIs to benefit from Intel hardware* capabilities and to leverage hybrid-core architecture efficiently

Let's win together!

*CPU, GPU and other accelerators

intel.

Notices and Disclaimers

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

intel