WebGPU API

Lecture 1: Introduction

Benjamin Kenwright

Brief Overview

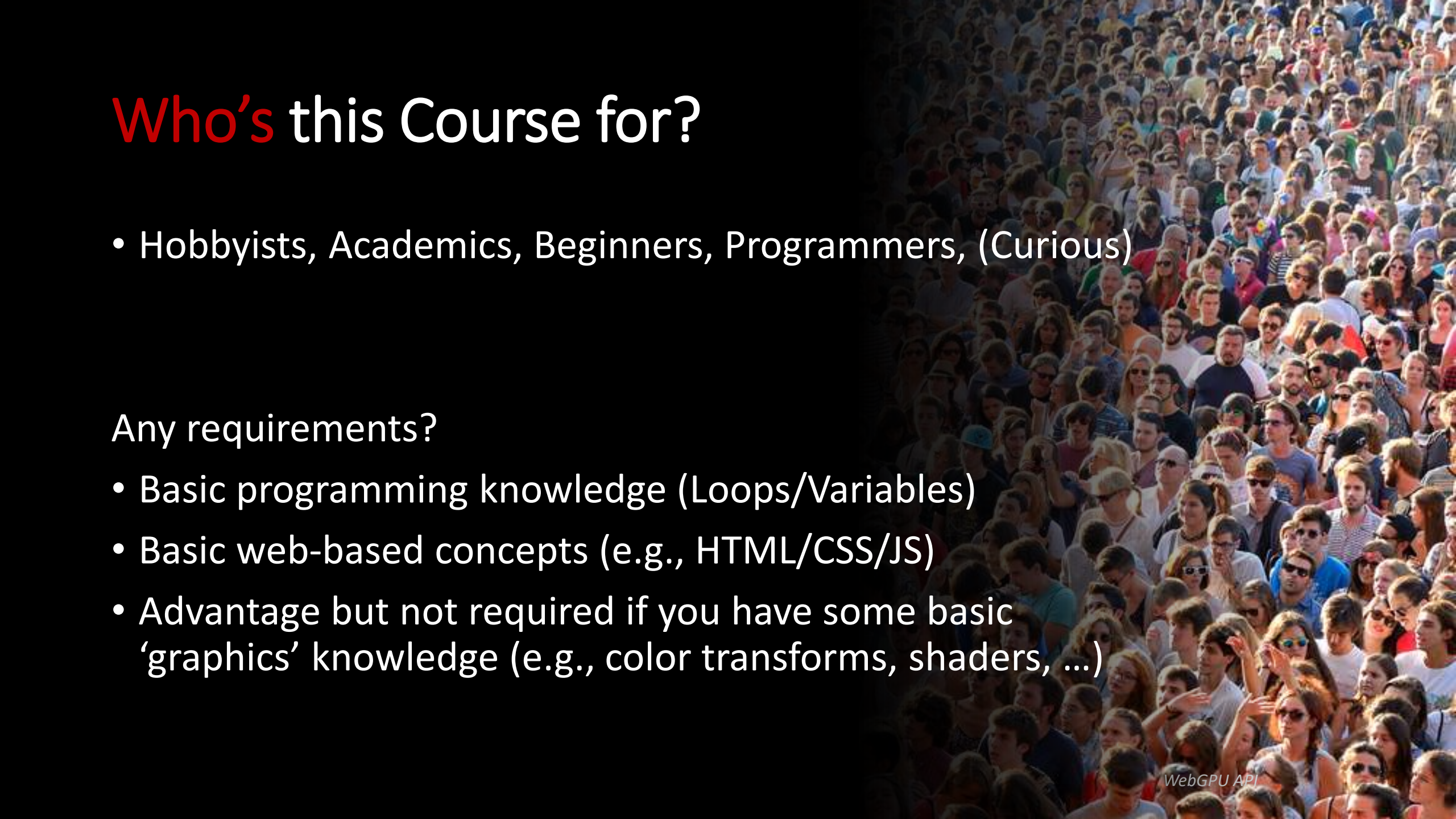
- Course Structure
- Background History/Trends
- Basic Concepts/WebGPU API
  - Whats/Whys/Where
- WebGL vs WebGPU
- Shader Languages
  - GLSL and new WGSL (WebGPU Shader Language)
- Discussion & Questions

Who's this Course for?

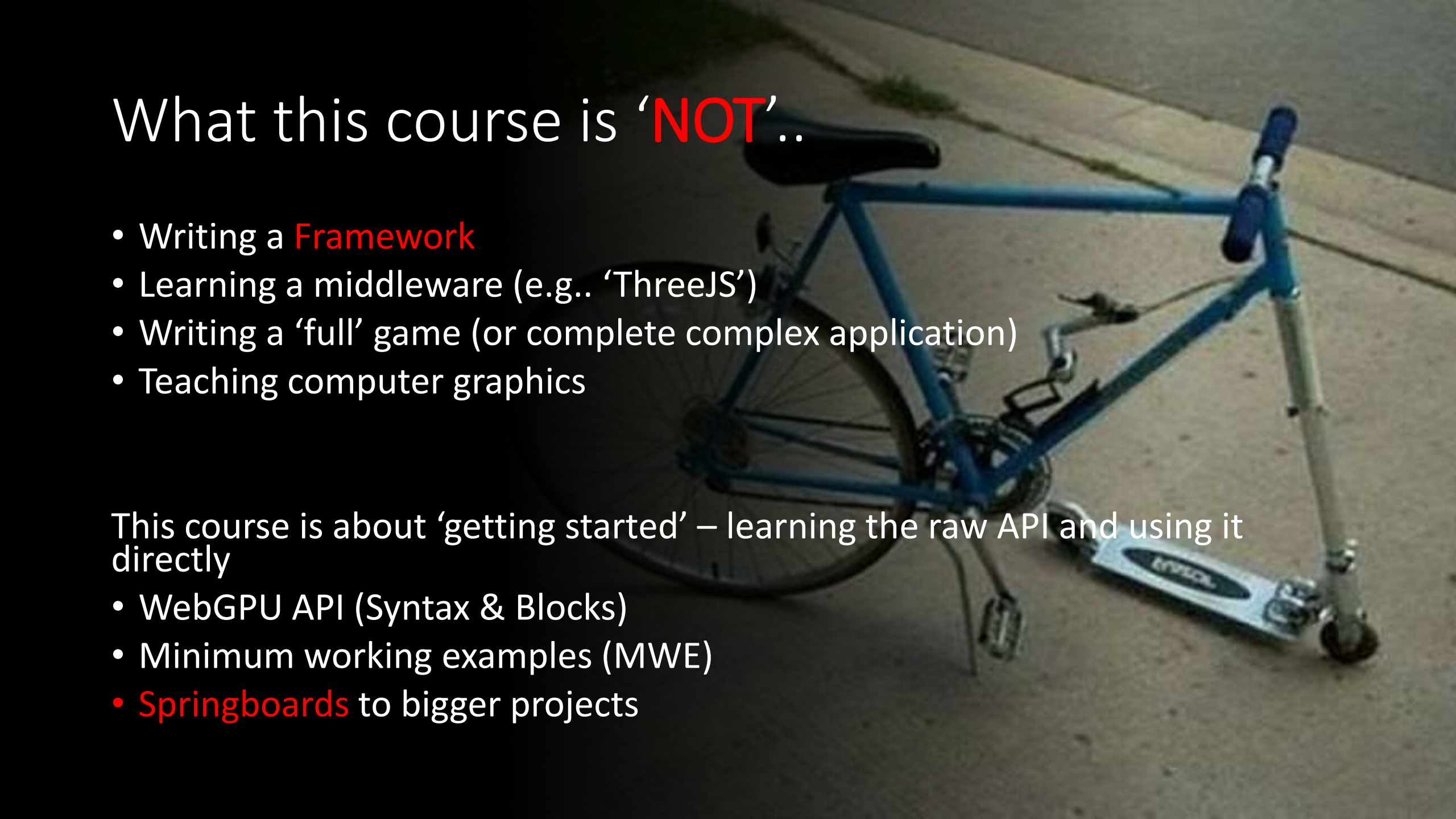
- Hobbyists, Academics, Beginners, Programmers, (Curious)

Any requirements?

- Basic programming knowledge (Loops/Variables)
- Basic web-based concepts (e.g., HTML/CSS/JS)
- Advantage but not required if you have some basic 'graphics' knowledge (e.g., color transforms, shaders, …)



What this course is ‘NOT’..



- Writing a Framework
- Learning a middleware (e.g.. ‘ThreeJS’)
- Writing a ‘full’ game (or complete complex application)
- Teaching computer graphics

This course is about ‘getting started’ – learning the raw API and using it directly

- WebGPU API (Syntax & Blocks)
- Minimum working examples (MWE)
- Springboards to bigger projects

Notes

- Warning…
  - Supported Browsers (install)
  - Involve coding! (Not ‘theoretical’)
  - …
- Simple (MWE) Examples
  - Clearing screen, flashing, …
  - Buffers, geometry…
  - Extending examples (step by step)
- Graphics then Compute
- ..

Stop Points
(Try and Test Yourself)

Activities after each Example

Question Slides (Think)


A new API …
A new beginning…
WebGPU

What is the WebGPU API?

What is the WebGPU API?

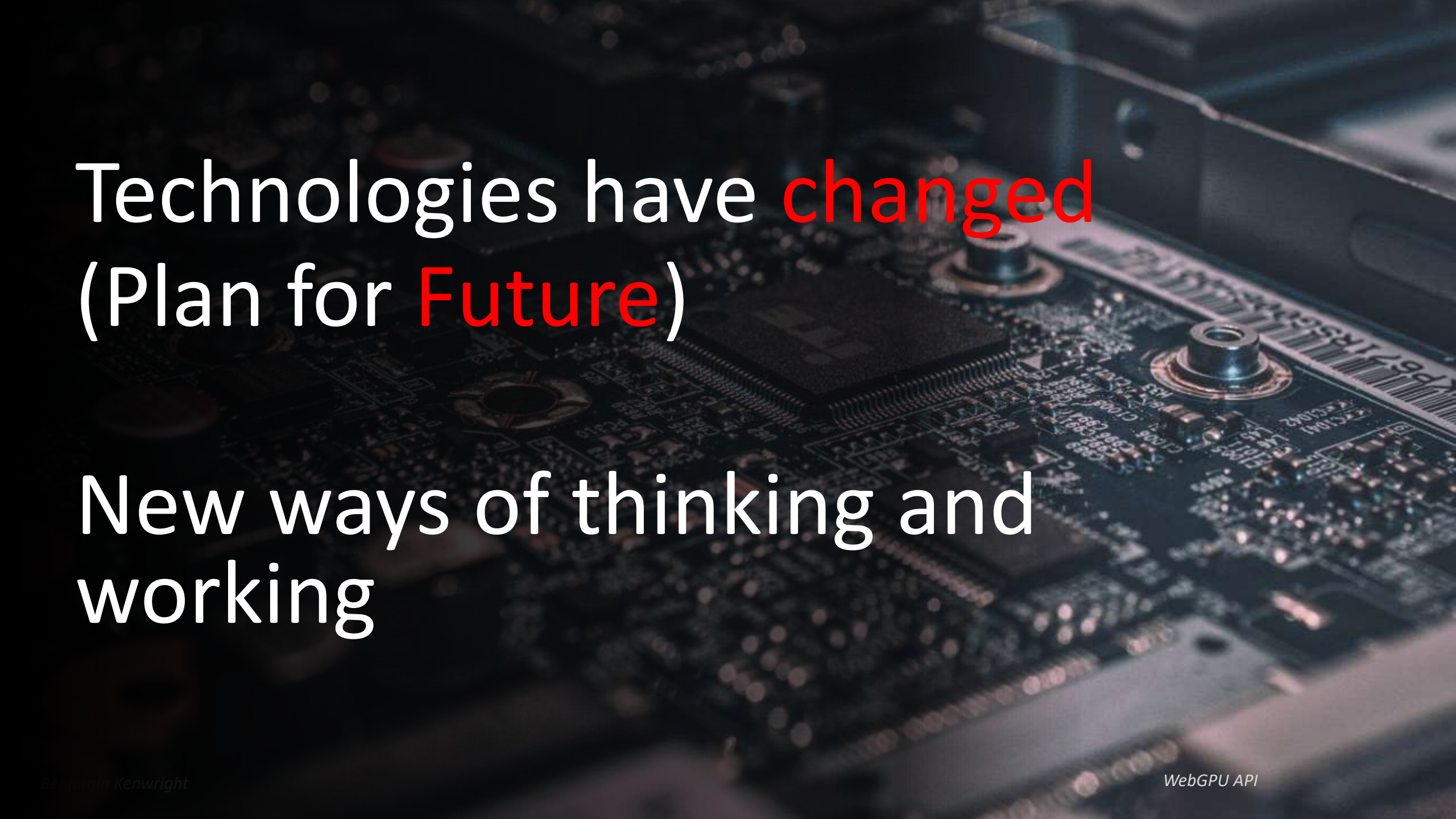
WebGPU API is a *new* Cross-Platform 2/3D Compute/Graphics API (for the web)

High-level graphics/compute API that taps “openness” of the Native GPU API (Vulkan/DX/Metal)

Why a new GPU API?

Do we need the WebGPU API?

Why not create just create WebGL 3?


Technologies have changed
(Plan for Future)
New ways of thinking and working

Problems with Backward Compatibility

Time for a tidy up!

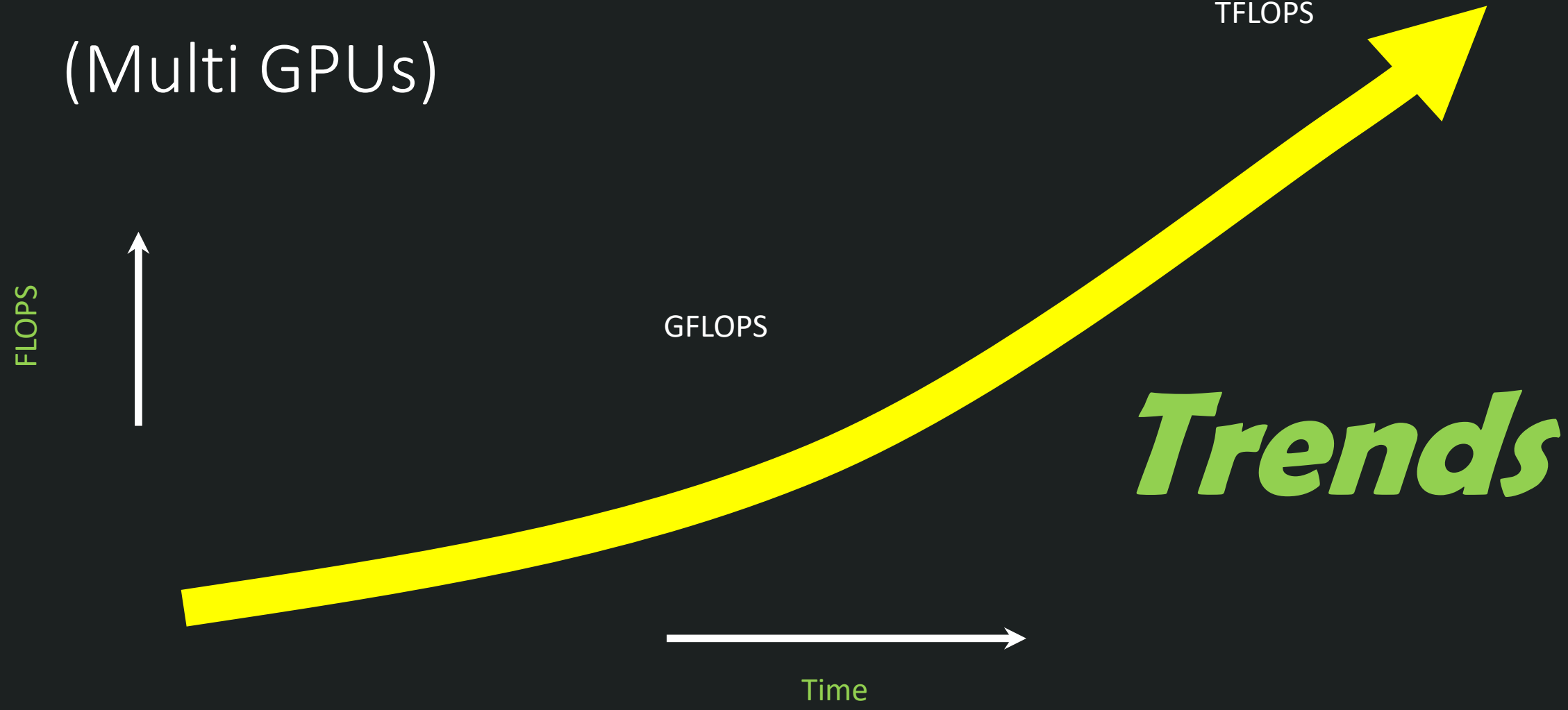
New API (Fresh Start)

Future Web-Based Technologies

Change how we approach the problem…
Tools, resources, technologies, …

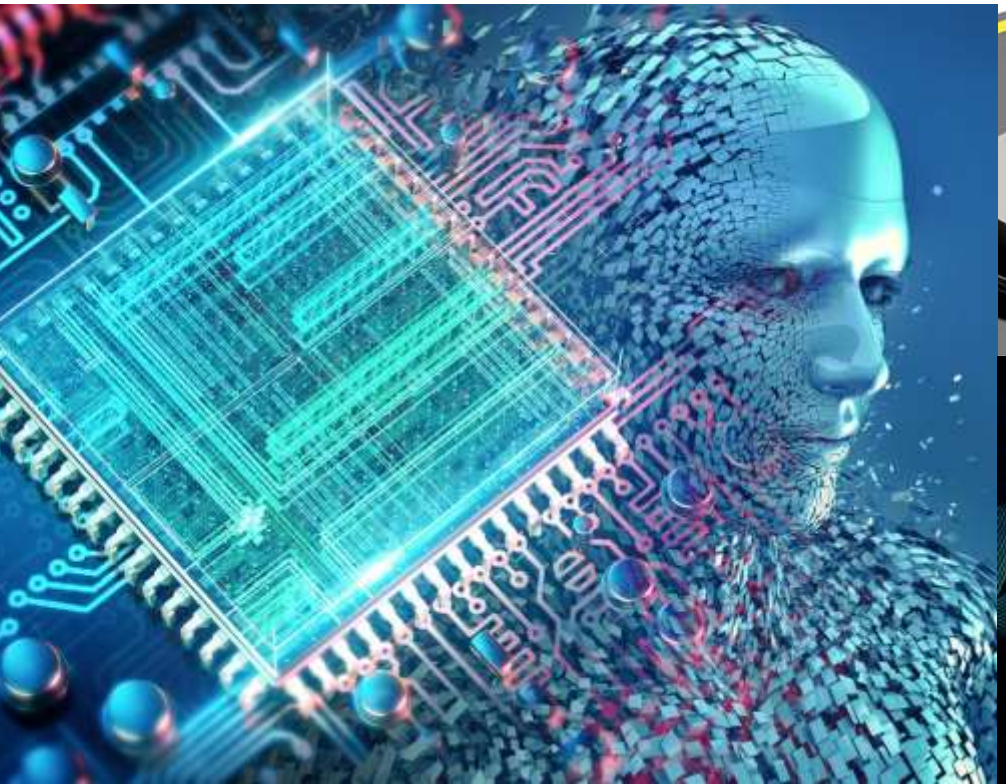
GPU 'Performance'

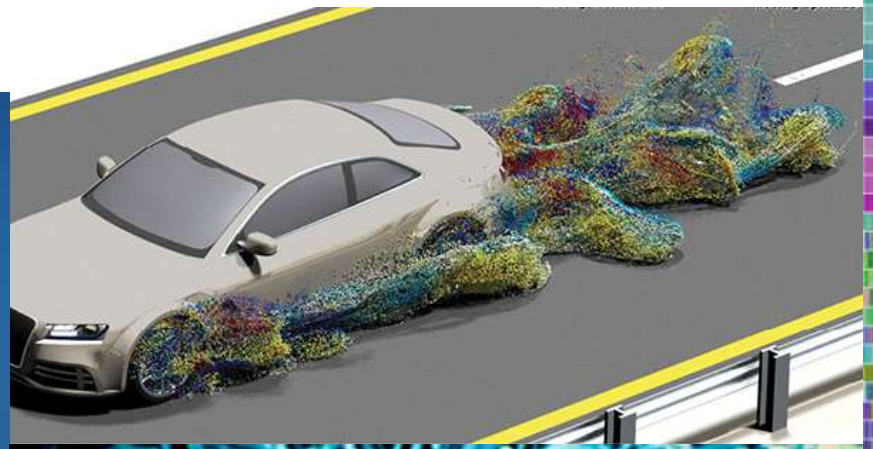
(Multi GPUs)

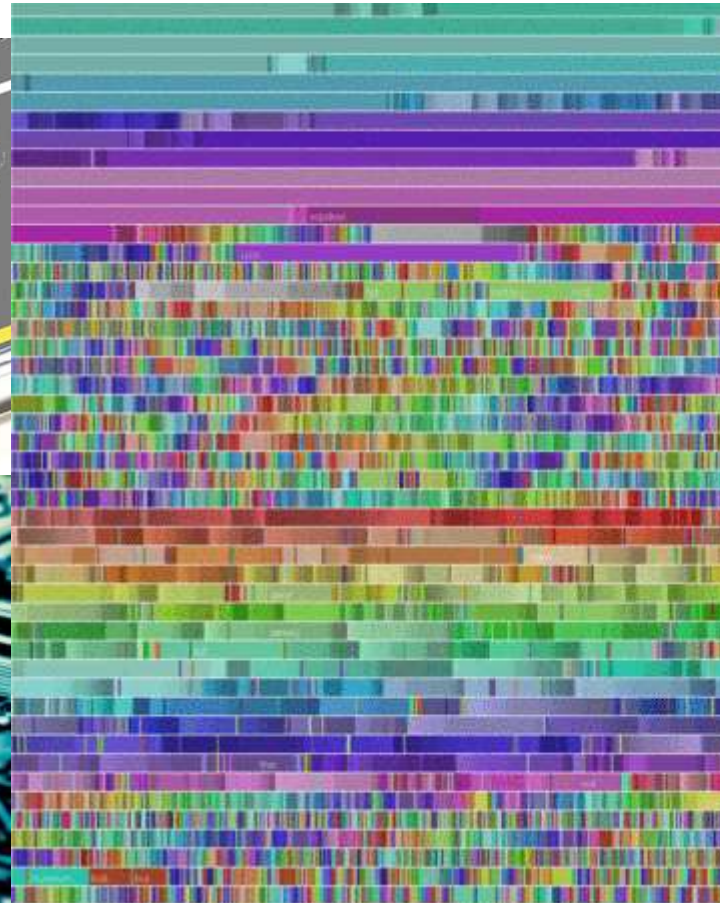


GPU (Not just for ‘Graphics’)

AI
Physics (Simulations)
Security/Crypt
Analytics







Is WebGL dead?

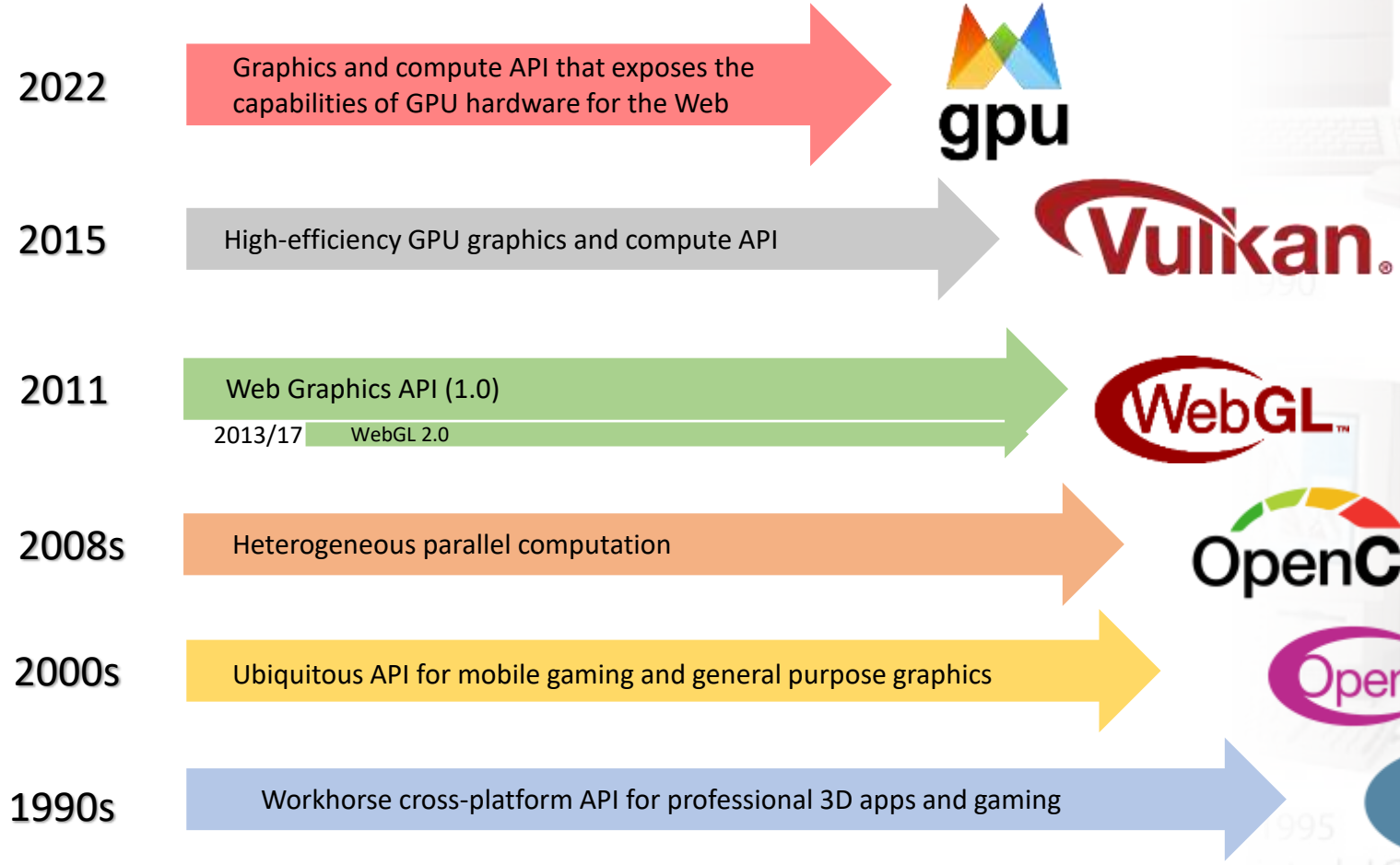
Does this mean WebGL is dead?
Won't be supported anymore? RIP WebGL?

NO

New Tool

WebGL is part of the W3c standards
Be around for a long time

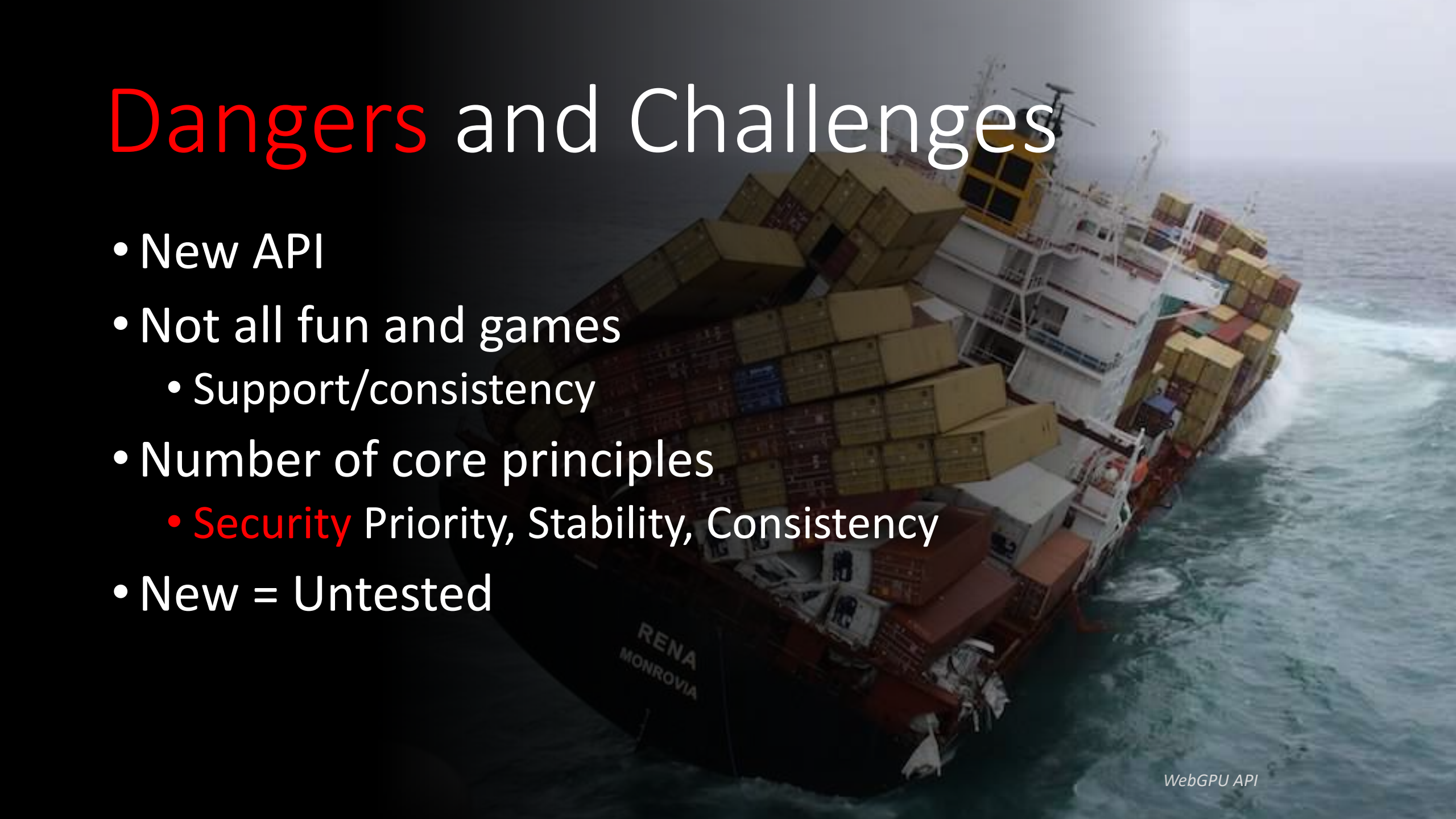
All APIs continue to evolve and be maintained to meet industry standards.
Provides a rich mix of open technologies for future innovation



Simplified Overview of the Open Graphics and Compute Standards

Dangers and Challenges

- New API
- Not all fun and games
  - Support/consistency
- Number of core principles
  - Security Priority, Stability, Consistency
- New = Untested



API is ‘NEW’ (Final Changes)

- Final phases development tests/checks before release
- Check official specification (final updates)
  - WebGPU API: https://www.w3.org/TR/webgpu/
  - WGSL: https://www.w3.org/TR/WGSL/

Open standard

Important WebGPU Features

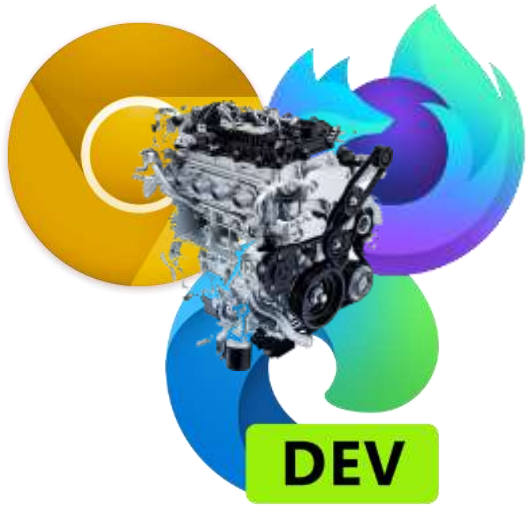
WebGPU multi-worker/tasks

1. Asynchronous texture & buffer uploads
2. Asynchronous shader compilation
3. Asynchronous pipeline state creation
4. Parallel rendering/encoding
5. Multiple threads/pool own command buffer

Textures (bindless)
GPU more fine-grained Visibility
No need to pass data back and forth between CPU and GPU

Opens the door…

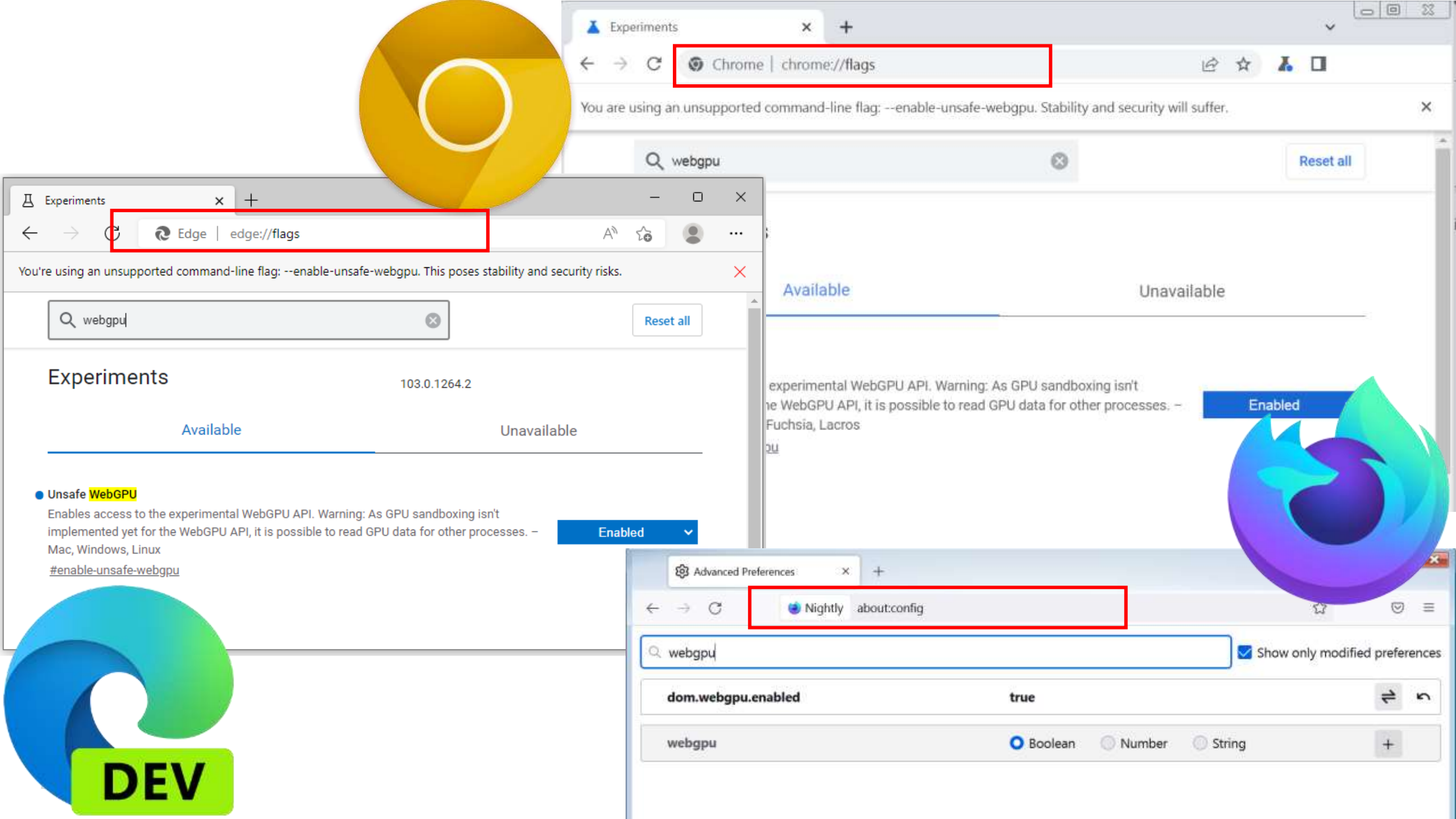
Do you have the WebGPU API?



Enabling API (Chrome Canary/Firefox Nightly/Edge Dev)

WebGPU is available (for Chrome Canary/Firefox Nightly/MS Edge Dev) behind an experimental flag.

You can enable on:
Chrome -> chrome://flags
Firefox -> about:config
Edge -> edge://flags


Experiments
Chrome | chrome://flags
You are using an unsupported command-line flag: --enable-unsafe-webgpu. Stability and security will suffer.
webgpu
Reset all
Available
Unavailable
experimental WebGPU API. Warning: As GPU sandboxing isn't
he WebGPU API, it is possible to read GPU data for other processes. –
Fuchsia, Lacros
Enabled
Experiments
Edge | edge://flags
You're using an unsupported command-line flag: --enable-unsafe-webgpu. This poses stability and security risks.
webgpu
Reset all
Experiments
103.0.1264.2
Available
Unavailable
Unsafe WebGPU
Enables access to the experimental WebGPU API. Warning: As GPU sandboxing isn't implemented yet for the WebGPU API, it is possible to read GPU data for other processes. – Mac, Windows, Linux
#enable-unsafe-webgpu
Enabled
DEV
Advanced Preferences
Nightly about:config
webgpu
Show only modified preferences
dom.webgpu.enabled
true
webgpu
Boolean
Number
String

Stats: https://caniuse.com/webgpu

WebGPU - WD

An API for complex rendering and compute, using hardware acceleration. Use cases include demanding 3D games and acceleration of scientific calculations. Meant to supersede WebGL.

Usage % of all users ?

Global 0%

Current aligned | Usage relative | Date relative | Filtered | All

IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
	12-79		4-79													
	3 80-93	2-62	3 80-93	3.1-11	10-72											
6-10	3 4 94-101	1 63-100	3 4 94-101	2 11.1-15.4	3 73-85	3.2-15.4		2.1-4.4.4	12-12.1				4-15.0			
11	3 4 102	1 101	3 4 102	2 15.5	3 86	15.5	all	101	64	102	1 101	12.12	16.0	10.4	7.12	2.5
		1 102-103	3 4 103-105	2 16.0-TP	3 87	16.0										

(Red – Disabled by Default)

Notes | Test on a real browser | Known issues (0) | Resources (6) | Feedback

All major browser engines are working on implementing this spec.

Chrome status: Origin trial

WebKit status: In Development

1 Can be enabled in Firefox with the `dom.webgpu.enabled` flag.

2 Can be enabled in Safari on MacOS with the `WebGPU` experimental feature.

3 Can be enabled in some Chromium browsers (on Windows, MacOS, Linux) with the `enable-unsafe-webgpu` flag.

4 Part of an origin trial.

Small check – see if your browser has the WebGPU API

Small JS Program

```
const gpu = navigator.gpu;
console.log( gpu );
if ( gpu === undefined )
{
  console.log("Oh no! You don't have WebGPU or it isn't enabled");
}
else
{
  console.log('Yahhh! You have the API');
}
```

1

2

Note, as you'll see later, just because you 'have' the API available, doesn't necessarily mean you can use it (also needs to have a driver/device)

https://webgpulab.xbdev.net/index.php?page=editor&id=test&

navigator object contains information about the browser (e.g., apis, such as, WebXR and WebGPU API)

You can always ‘list’ all the available entries in the navigator like this:

```
for (let n in navigator){ console.log( n ); }
```

2 **navigator.gpu** is the ‘webgpu’ api – if this is not ‘undefined’ you have the api

Question

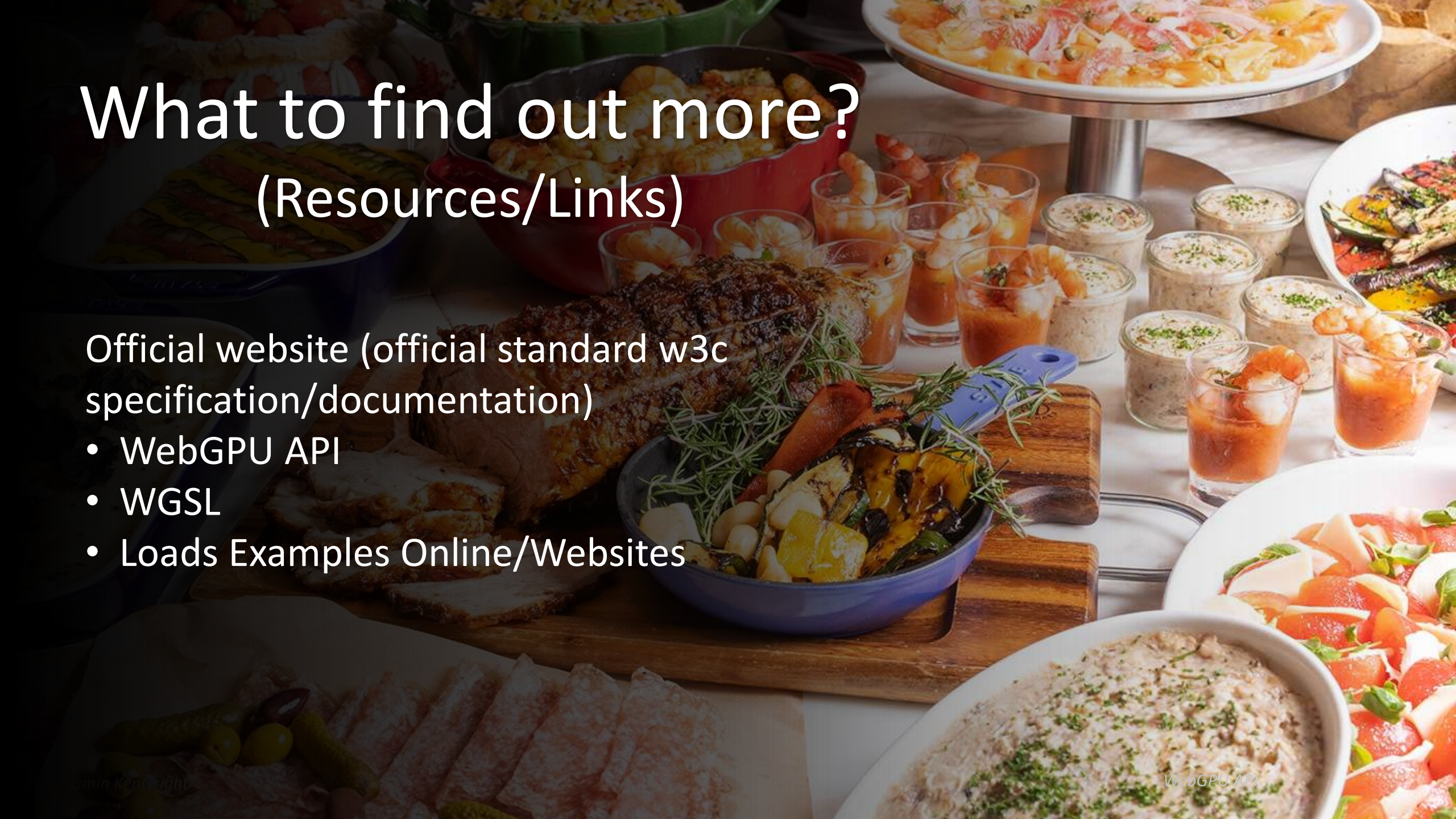
Who manages the WebGPU API?

(Organisation)

Answer

Who manages the WebGPU API?

WebGPU is managed by **Khronos Group** (working group). The group includes Apple, Google, Mozilla and Opera

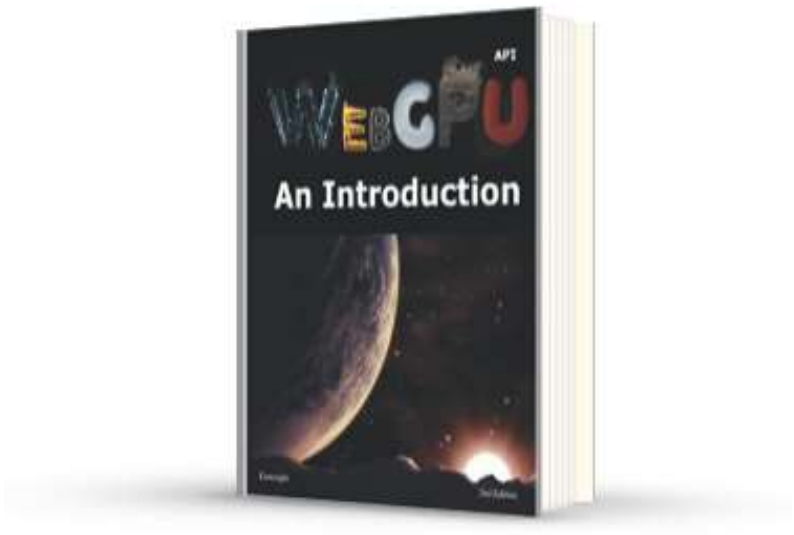


What to find out more?

(Resources/Links)

Official website (official standard w3c specification/documentation)

- WebGPU API
- WGSL
- Loads Examples Online/Websites



WebGPU API: An Introduction

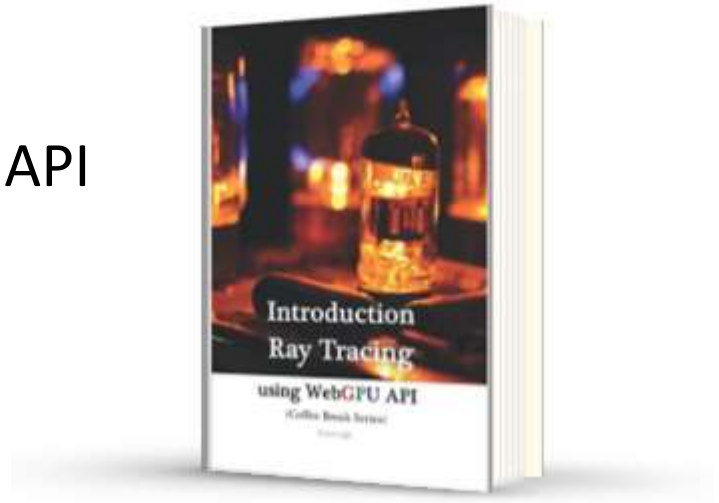
ISBN-13 : 979-8836435554

Kenwright

Introduction to Ray-Tracing using WebGPU API in 20 Minutes (Coffee Book Series)

ISBN-13 : 979-8842178605

Kenwright



Summary/Conclusion

Stay Tuned Lecture 2

Summary/Conclusion

- New era
- Final stages (lots of opportunities/potential)
- Easy to get started (no excuse)
- Challenges/updates/fixes
- Must not forget about ‘security’, ‘stability’, …
- WebGPU driving future technologies to think outside the box
- WebGPU API is ‘NEW’ (untested)
- Example, try WebGPULab.com range examples
- …

Lots of examples and resources available….
Learn the basics first (makes it easy when debugging/fixing problems in more complex projects)