



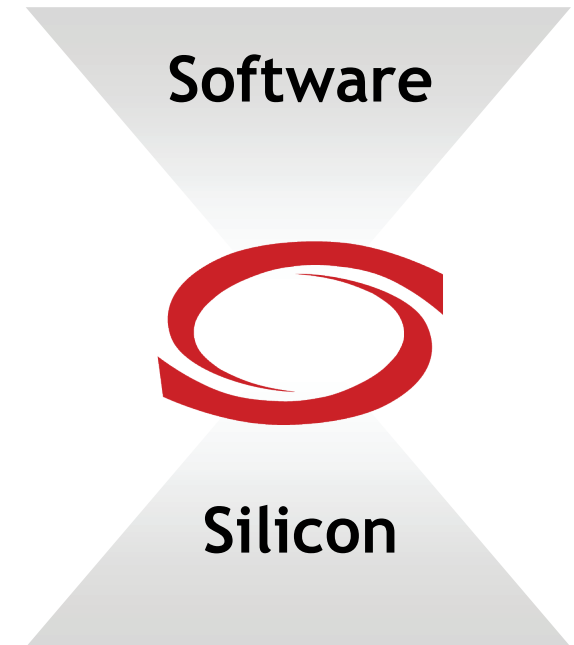
Khronos Updates

GDC 2017

Neil Trevett

Vice President Developer Ecosystem, NVIDIA | President, Khronos
ntrevett@nvidia.com | [@neilt3d](https://twitter.com/neilt3d)

Khronos Mission



Khronos is an International Industry Consortium of over 100 companies creating royalty-free, **open standard APIs** to enable software to access hardware acceleration for **3D graphics, Virtual and Augmented Reality, Parallel Computing, Neural Networks and Vision Processing**

Khronos Standards Ecosystem

3D for the Web

- Real-time apps and games in-browser
- Efficiently delivering runtime 3D assets



Real-time 2D/3D

- Cross-platform gaming and UI
 - VR and AR Displays
- CAD and Product Design
 - Safety-critical displays

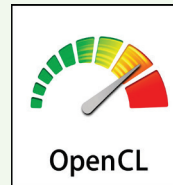
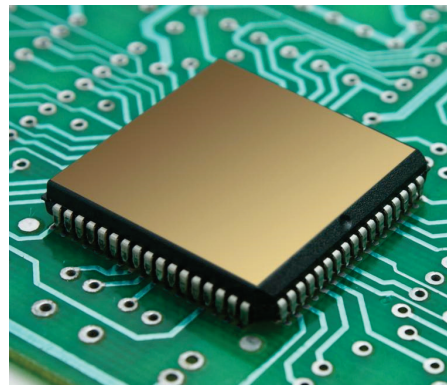


NNEF



VR, Vision, Neural Networks

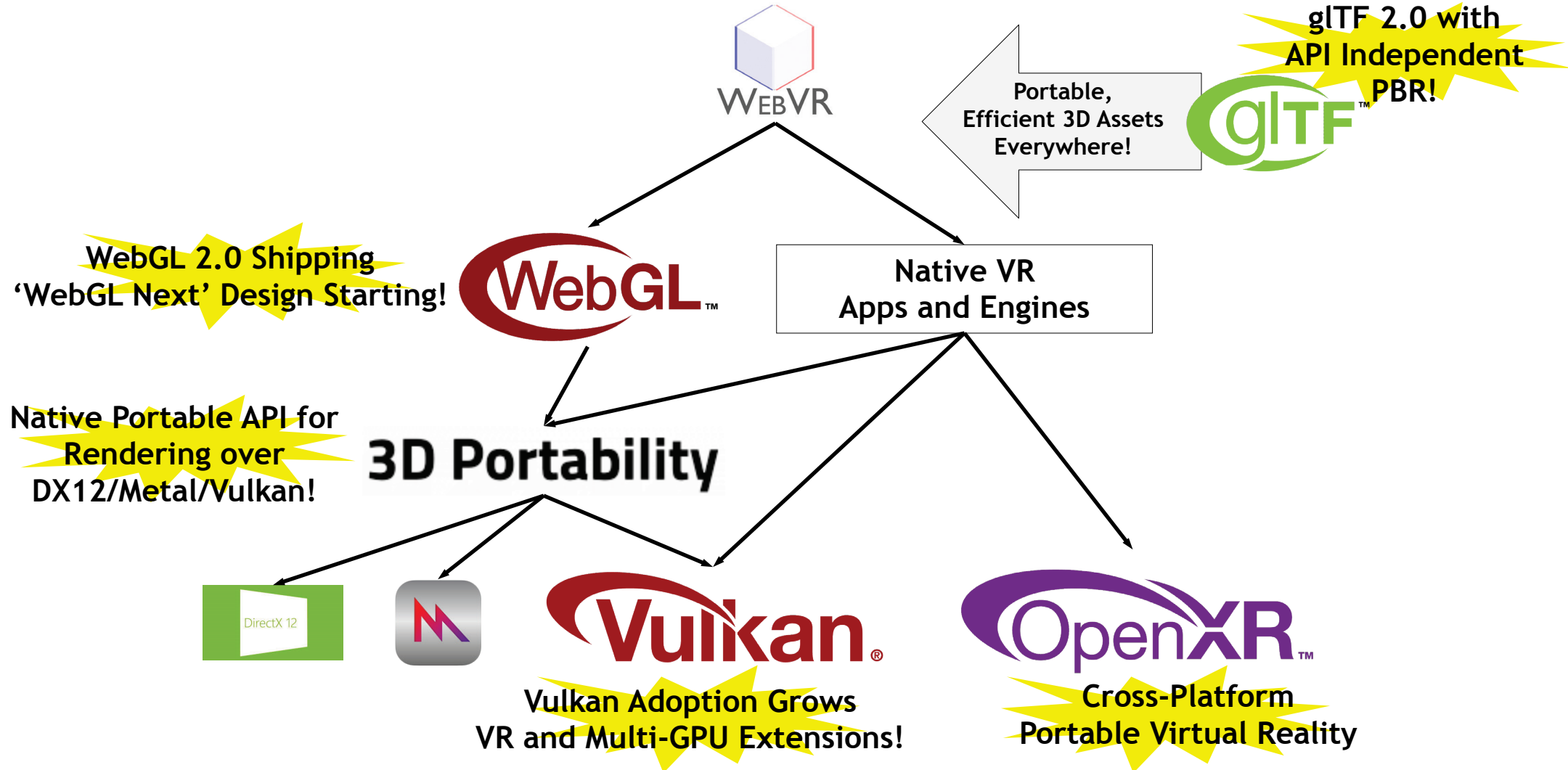
- VR/AR system portability
- Tracking and odometry
- Scene analysis/understanding
- Neural Network inferencing



Parallel Computation

- Machine Learning acceleration
- Embedded vision processing
- High Performance Computing (HPC)

Khronos News at GDC 2017



Khronos News Here at GDC 2017

- Adoption Grows for Vulkan®; New Features Released
 - [Details here](#)
- Announcing OpenXR™ for Portable Virtual Reality
 - <https://www.khronos.org/blog/the-openxr-working-group-is-here>
- WebGL™ 2.0 Specification Finalized and Shipping
 - <https://www.khronos.org/blog/webgl-2.0-arrives>
- Call for Participation in 3D Portability Exploratory Group
 - A native API for rendering code that can run efficiently over Vulkan, DX12 and Metal [khronos.org/3dportability](https://www.khronos.org/3dportability)
- Developer preview on glTF™ 2.0
 - <https://www.khronos.org/blog/call-for-feedback-on-gltf-2.0>



Vulkan and New Generation 3D APIs



Only Windows 10



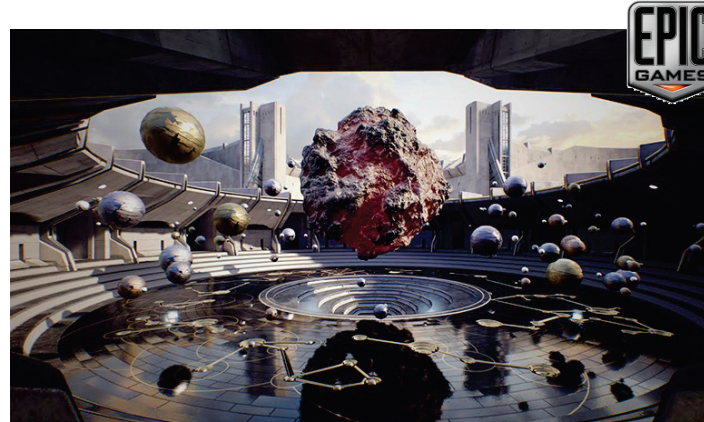
Only Apple



Vulkan Games and Game Engines



Dota 2 on Vulkan port of Source 2



'ProtoStar' demo on Vulkan port of Unreal Engine 4



Talos Principle on Vulkan port of Serious Engine



Doom's Vulkan patch is a PC performance game-changer

DOOM on Vulkan port of id Tech 6



Vulkan support coming



Vulkan support in V1.8



Developer Preview

Vulkan Momentum Continues to Build



Games Studios publicly confirming that work is ongoing on Vulkan Titles



In first 12 months :
#Vulkan Games on PC = 11

In first 18 months
#DX12 Games on PC = 19

https://en.wikipedia.org/wiki/List_of_games_with_Vulkan_support

All Major GPU Companies shipping Vulkan Drivers - for Desktop and Mobile Platforms



Mobile, Embedded and Console Platforms Supporting Vulkan



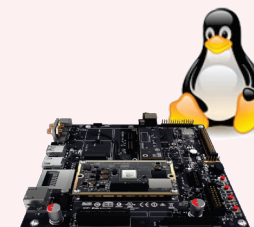
Android 7.0



Nintendo Switch



Android TV



Embedded Linux

New Vulkan Functionality at GDC 2017



- Vulkan 1.0.42 released with new extension sets for VR and multi-GPU
 - <https://www.khronos.org/registry/vulkan/#apispecs>
- Enables developers with key functionality today
 - AND gathers experience and feedback for future Vulkan core spec releases
- First use of KHX extensions
 - Developed by the working group - and ratified - like traditional KHR extensions
 - But will have TEMPORARY lifetime - should NOT be built into production code
 - Enables developer feedback without polluting long-term extension space
- New LunarG SDK for Vulkan Header 1.0.42.0 released today!
 - Includes support for all newly Released Functionality!
- NVIDIA has published [their new Vulkan beta drivers](#) on day of spec release
 - With full support for all the new v1.0.42 extensions
 - Plus building block Vulkan extensions for VRWorks on Maxwell and Pascal



Vulkan Multi-GPU Support

- Native multi-GPU support for NVIDIA SLI and AMD Crossfire platforms
 - WDDM must be in “linked display adapter” mode
 - The most common use case - does NOT support dGPU/iGPU
- Explicit control of how GPUs cooperate to enable a variety of operating modes
 - AFR (alternate frame), SFR (Sequential frame) and VR SLI Stereo view rendering
- A “device group” is a set of physical devices that support multi-GPU rendering
 - Acts as single logical device - makes adding device group support as easy as possible
 - Only access each physical GPU in a device group when need explicit control:
 - Memory allocation and binding resources
 - Command Buffer Recording/Submission
 - Synchronization



Vulkan Extension Sets for VR

- The most requested functionality by developers
 - Building Block approach provides explicit level of control to build powerful VR systems
- **Multiview extension set**
 - Render geometry to multiple surfaces, each with its own viewing parameters
 - Can efficiently render stereo pairs or environment maps
- **Sharing extension set**
 - Share memory and synchronization primitives across process and instance boundaries
 - Useful for implementing real-time rendering systems such as VR runtimes
- **Descriptor Update extension set**
 - Alternate ways to update resource references between draw or compute dispatch calls
 - More efficient when a fixed set of resources must be updated repeatedly
 - More convenient for legacy applications



Khronos APIs for VR

Khronos APIs Powering VR Rendering Today

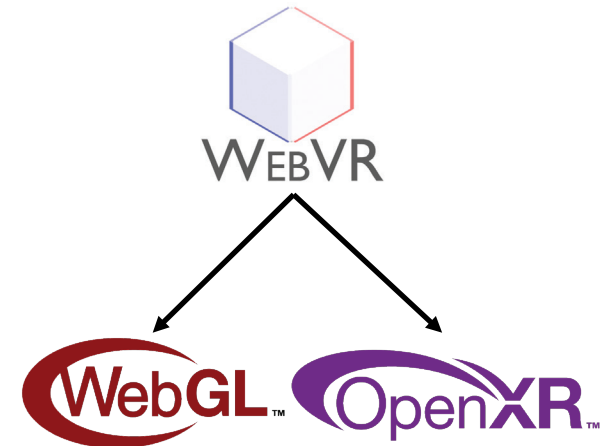
OpenGL ES and OpenGL on millions
of mobile VR devices
WebVR in browsers power by WebGL
SteamVR Beta using Vulkan

3D API features used by
VR compositors
Context priority
Front buffer rendering
Tiled rendering (beam racing)
Multiview



But What About Everything Else for VR..

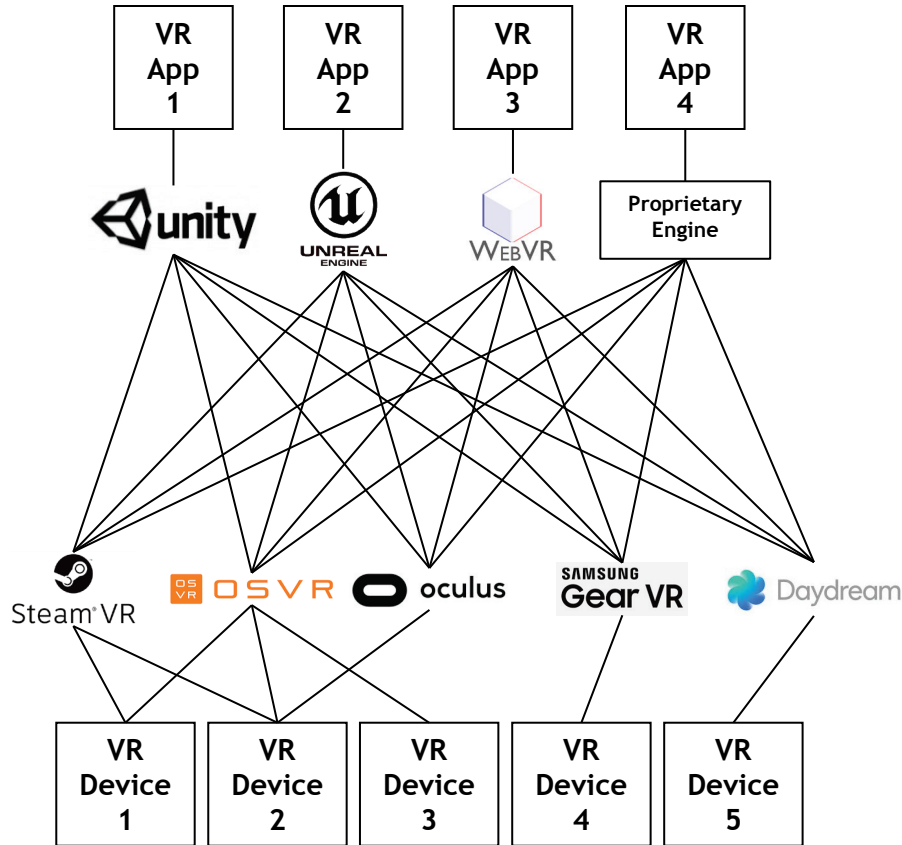
Device discovery
Multiple sensor tracking
Device Events
Haptics
Parameters for optics corrections etc. etc...



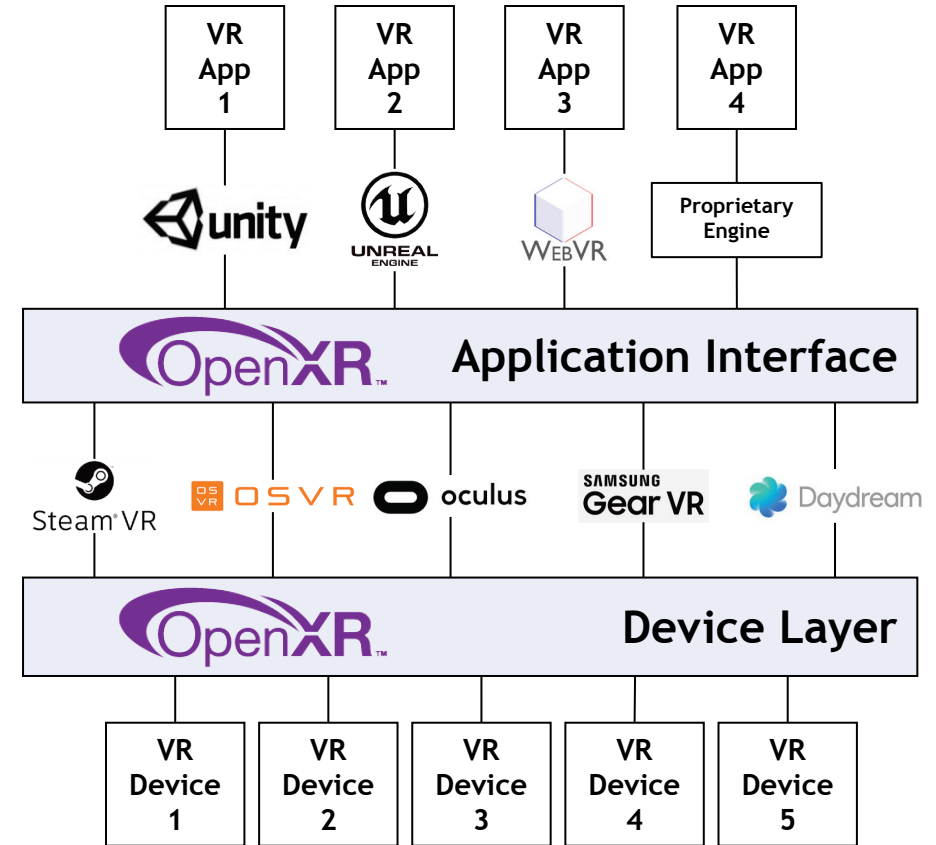
OpenXR!

Cross-Platform, Portable, Virtual Reality
Complements Rendering APIs

OpenXR - Solving VR Fragmentation



Before OpenXR
VR Market
Fragmentation



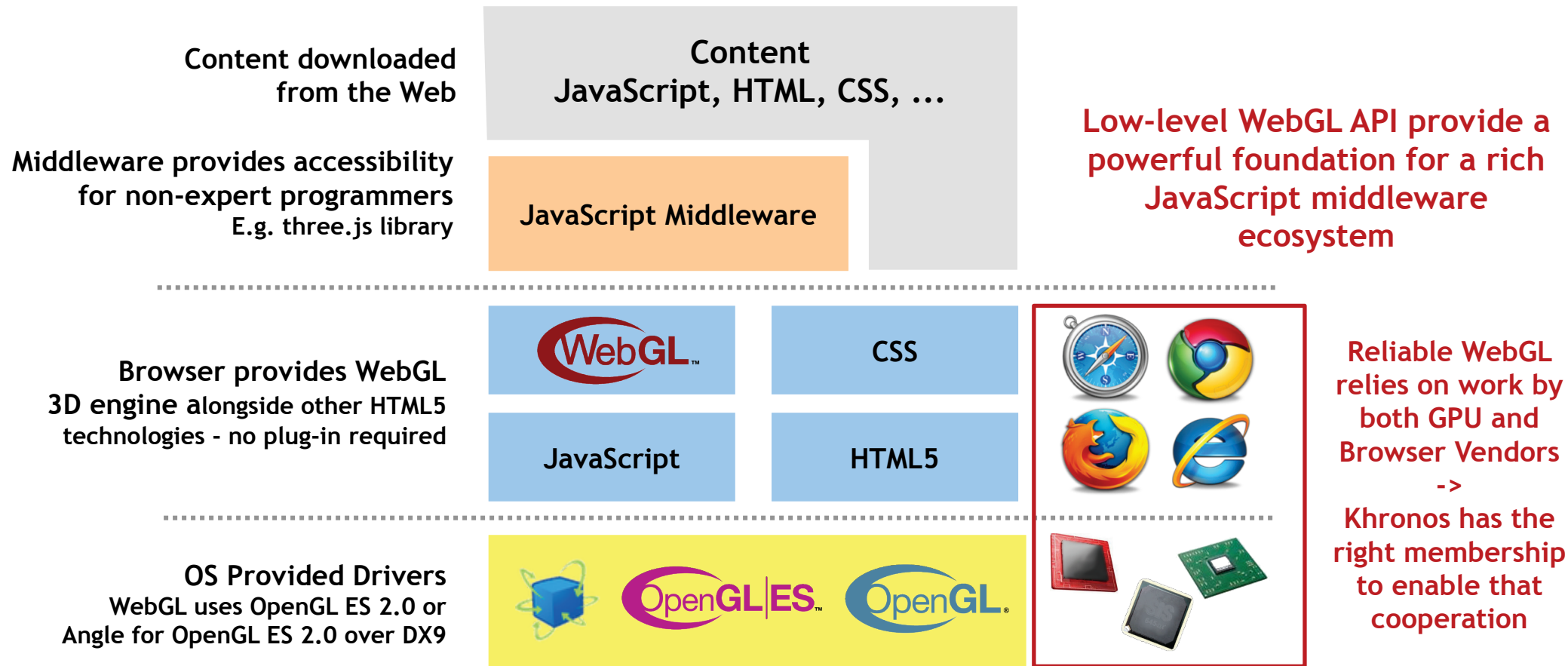
After OpenXR
Wide interoperability of
VR apps and devices

OpenXR Working Group Members



Design work has started in December 2016
Typically 12-18 months to develop a V1.0 specification

WebGL - 3D for the Web



Pervasive WebGL 1.0

- WebGL on EVERY major desktop and mobile browser

WebGL - 3D Canvas graphics - OTHER

Method of generating dynamic 3D graphics using JavaScript, accelerated through hardware

Global 57.62% + 33.11% = 90.73%
U.S.A. 70.95% + 25.07% = 96.02%

Current aligned	Usage relative	Date relative	Show all	IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
							49					4.3	
							54					4.4	
						50	55			9.3		4.4.4	
1	1			11	14	51	56	10	43	10.2	all	53	56
	1				15	52	57	10.1	44				
						53	58	TP	45				
						54	59						

<http://caniuse.com/#feat=webgl>

WebGL 2.0 Timeline

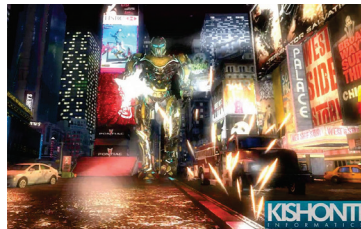
Fixed function
Pipeline



Programmable
Vertex and
fragment shaders



32-bit integers and floats
NPOT, 3D/depth textures
Texture arrays
Multiple Render Targets



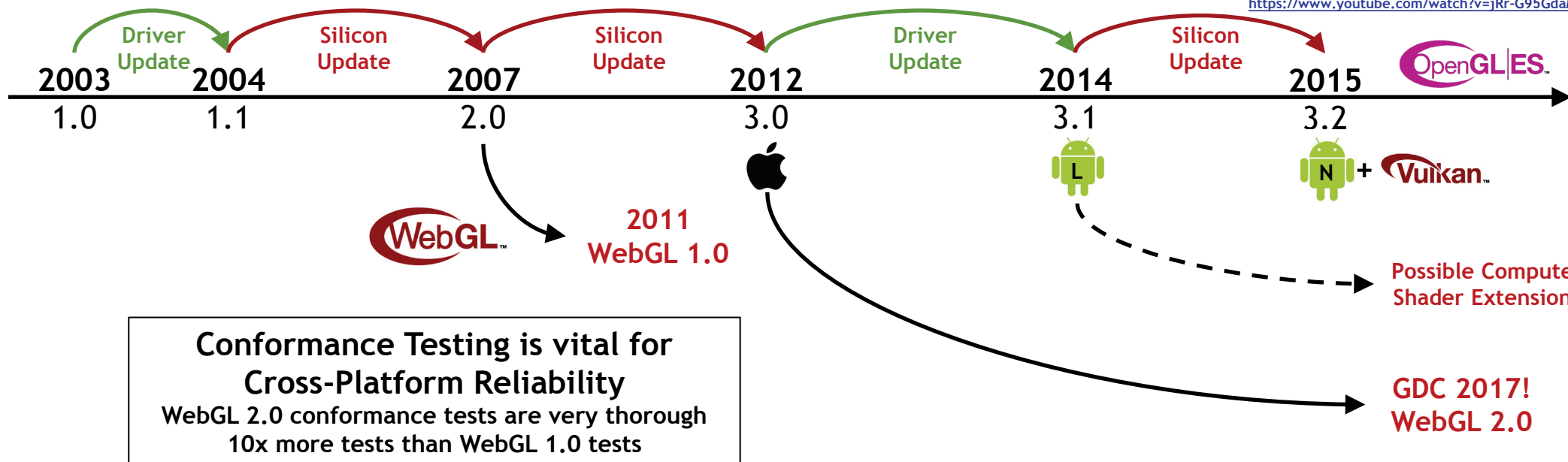
Compute Shaders



Tessellation and geometry shaders
ASTC Texture Compression
Floating point render targets
Debug and robustness for security



Epic's Rivalry demo using full Unreal Engine 4 on mobile
<https://www.youtube.com/watch?v=jRr-G95GdaM>



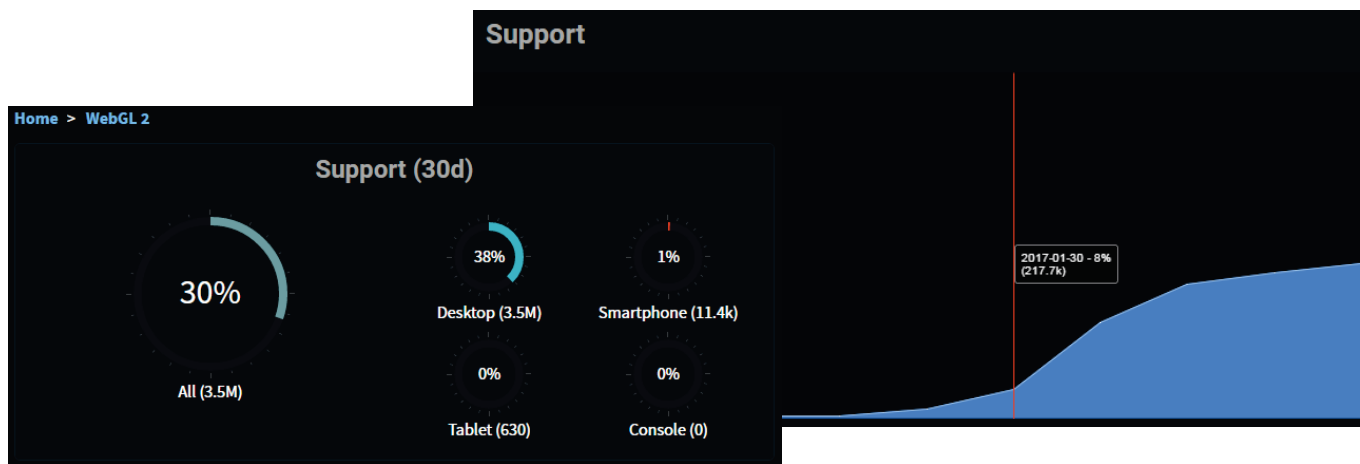
WebGL 2.0 - WebGL 2.0 Spec

- Enhanced visual quality, performance, features
 - Instancing Multiple render targets | Uniform buffers | Transform feedback
 - Multisampled Renderbuffers | 3D textures | NPOT textures
 - More texture formats | Occlusion queries | Vertex array objects
 - Sampler objects | Sync objects | Fragment depth | Primitive restart | ...
- WebGL 2.0 now available in Chrome/Firefox!
 - Chrome: Released to desktop platforms - soon on Android
 - Firefox: Released on all platforms
 - Edge/Safari: plan to ship WebGL 2.0
 - Desktop support at 38% - will continue rising

<http://webglstats.com/webgl2>



WebVR Running Over WebGL 2.0





WebGL 2.0

PlayCanvas: *After the Flood* [Video]

The 3D Portability Problem



Only Windows 10



Only Apple



For developers wishing to use the new generation of explicit 3D APIs there is no single API that runs on all desktop and mobile systems!
Problem for native apps that don't use game engines AND nexgen WebGL!

Need for Portability

“The golden age of application portability through OpenGL and OpenGL ES being available on all desktop and mobile systems is passing.

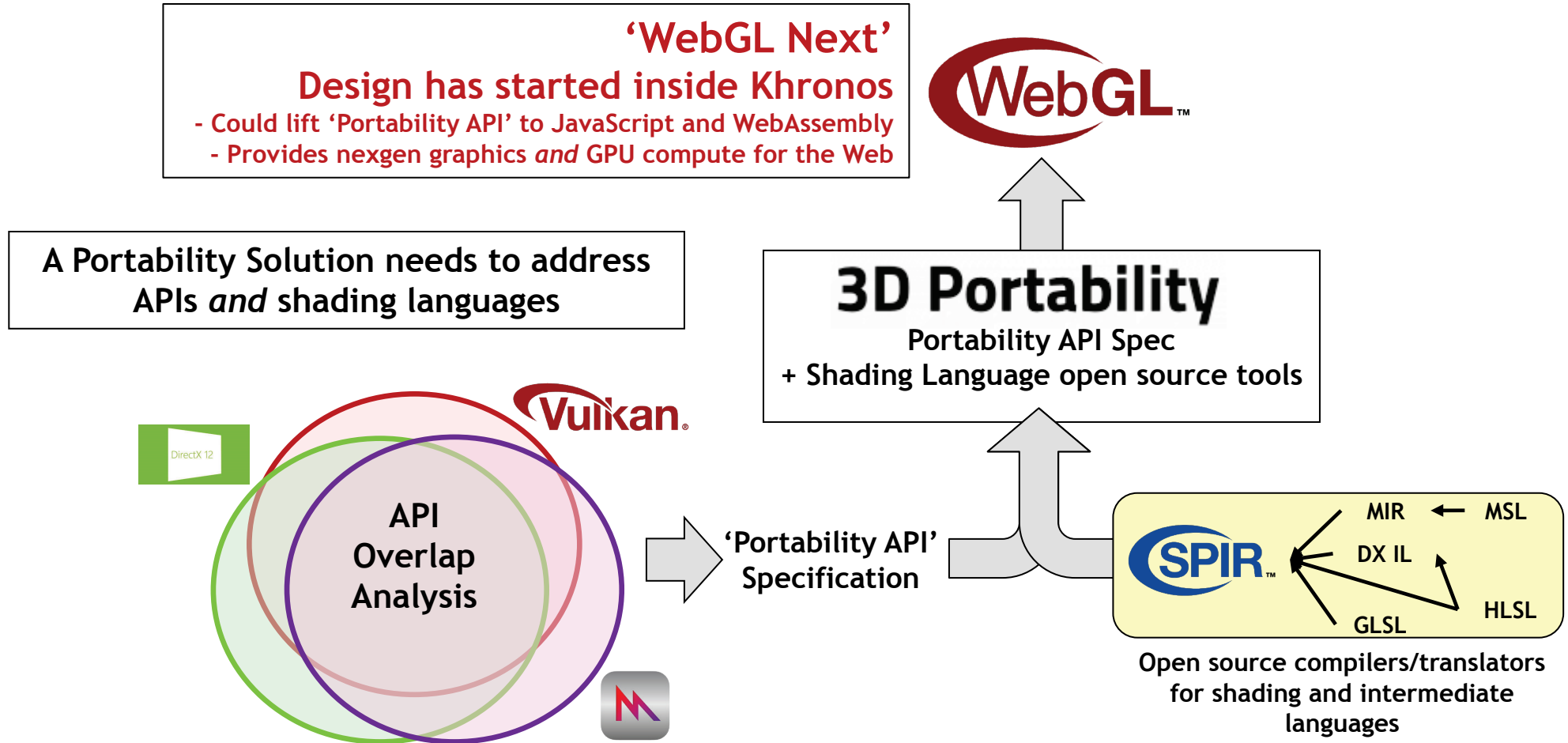
Developers now want to tap into the enhanced performance of the new generation explicit APIs: Vulkan, DX12 and Metal.

Every cross-platform developer, as well as WebGL, is facing the challenge of rendering portably and efficiently across systems using all three of these APIs.

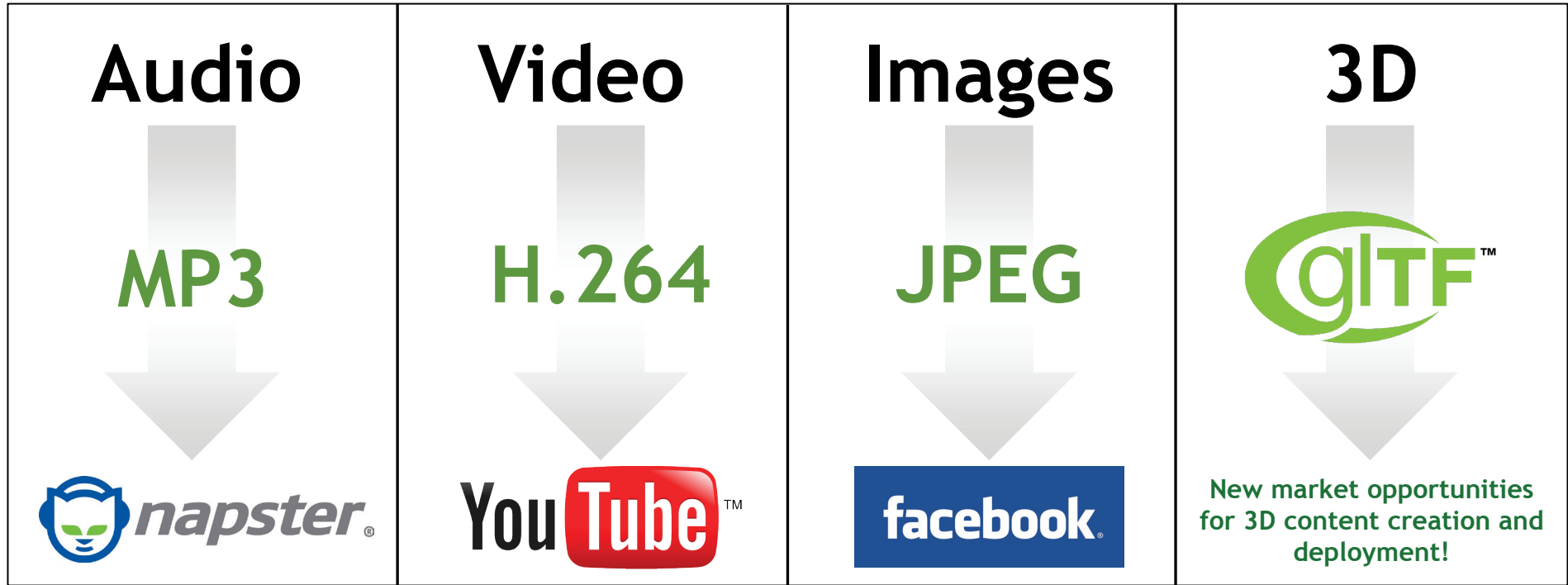
Khronos has been leading the development of cutting-edge native and Web 3D APIs for many years, and is uniquely qualified to solve this urgent issue confronting the industry.”

Jon Peddie, President of [Jon Peddie Research](#)

3D Portability API - Call For Participation



glTF - Runtime 3D Asset Delivery



model/gltf+json MIME type Approved by IANA



Compact to Transmit



Fast to Load



Describes Full Scenes



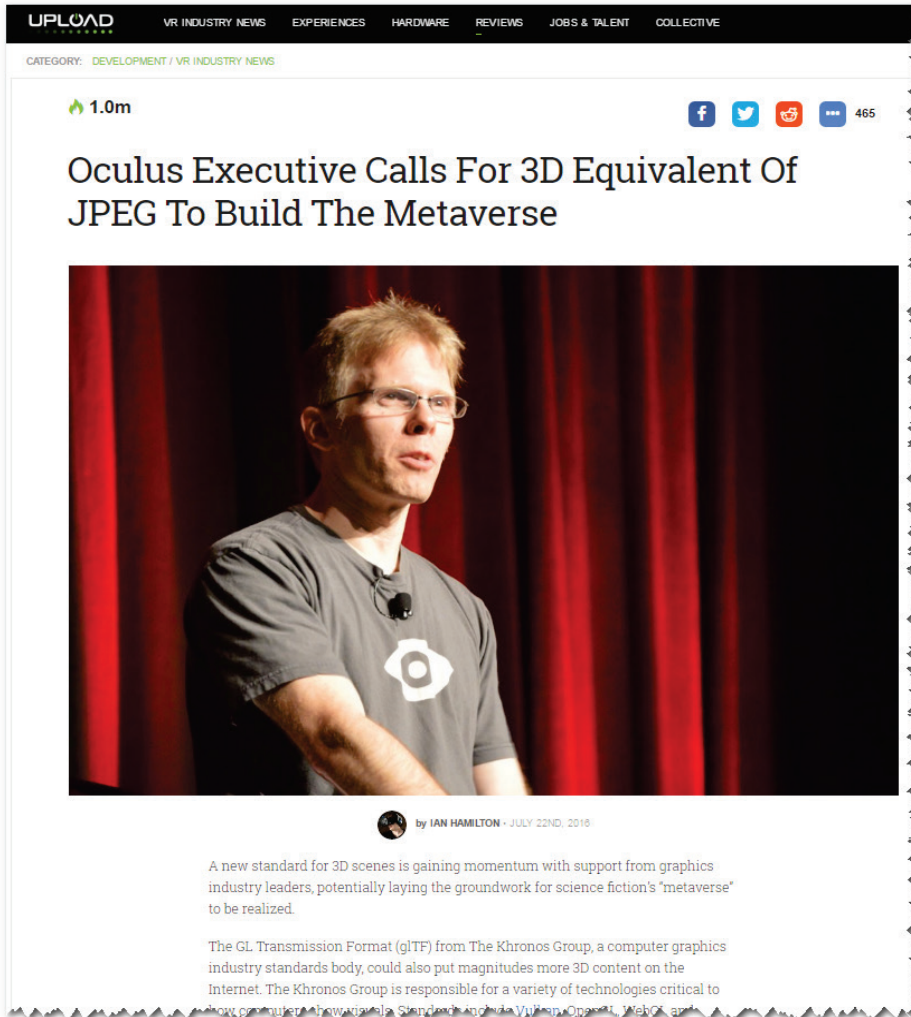
Runtime Neutral



Extensible



Strong glTF Momentum



Publicly Stated Support for glTF

glTF Milestones



All glTF spec development on open
GitHub:

<https://github.com/KhronosGroup/glTF>

2012 thru
2014

Dec
2015

Oct
2016

We are here!
Seeking feedback on
developer preview glTF 2.0
before finalization

Spring
2017

Design Iteration and
Multiple
Implementations

Original motivation:
standardized way to
deliver 3D into
WebGL applications

glTF 1.0
Spec
Ratified and
Released

Significant Industry
Adoption

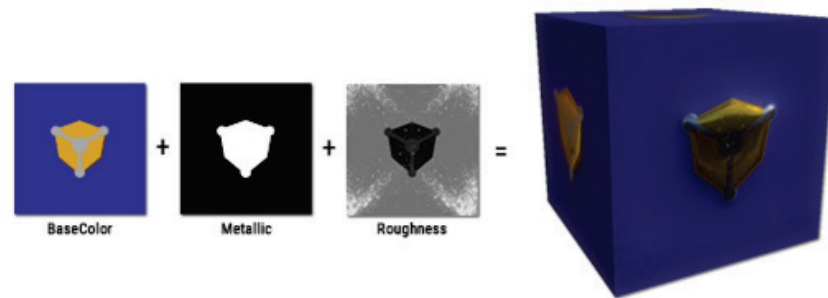
Validator
Project

glTF 2.0 Target
Spec Finalization

glTF 2.0 adds Physically
Based Rendering for
higher-quality materials
and rendering API
independence

glTF 2.0 Physically Based Rendering

- **In Core: Metallic-Roughness Material model**
 - baseColor — base color
 - metallic — metalness
 - roughness — roughness
- **Simple to implement with small resources**
 - Can be everywhere
- **Extension: Specular-Glossiness Material model**
 - diffuse — reflected diffuse color
 - specular — specular color
 - glossiness — glossiness
- **A little more resource heavy**
 - Optional extension (e.g. on low-power devices)
- **The two models can be combined**



glTF 2.0 PBR materials in various engines



WebGL reference implementation

<http://www.seas.upenn.edu/~moneimne/WebGL-PBR/>



Laugh Engine running on Vulkan

https://github.com/jian-ru/laugh_engine



takahiro(John Smith)
@superhoge

Following

Three.js glTF 2.0 loader loads 2.0 BoomBox!
[#threejs](#) [#gltf](#)





Backup Materials

What's new in glTF 2.0

- **Physically Based Rendering (PBR) material definitions**
 - Material information stored in textures
- **Graphics API neutral**
 - GLSL materials moved to extension
 - Proven by implementations using WebGL, Vulkan and Direct3D
- **Improvements**
 - Binary glTF in core
 - Enhanced Performance

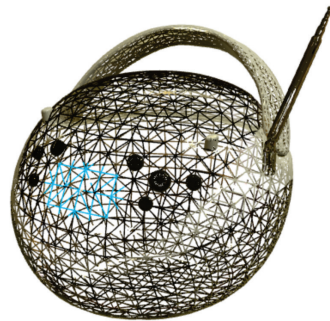


glTF 2.0 Scene Description Structure

.gltf (JSON)
Node hierarchy, PBR material textures, cameras

.bin
Geometry: vertices and indices
Animation: key-frames
Skins: inverse-bind matrices

**.png
.jpg
...**
Textures



Geometry



Texture based PBR materials

Physically Based Rendering

- Standardize the BRDF inputs for common PBR workflows
 - Metallic-Roughness and Specular-Glossiness
- Incredible industry effort
 - Started by Fraunhofer and supported by Microsoft, Sketchfab, NVIDIA, Autodesk, Marmoset, University of Pennsylvania, and others

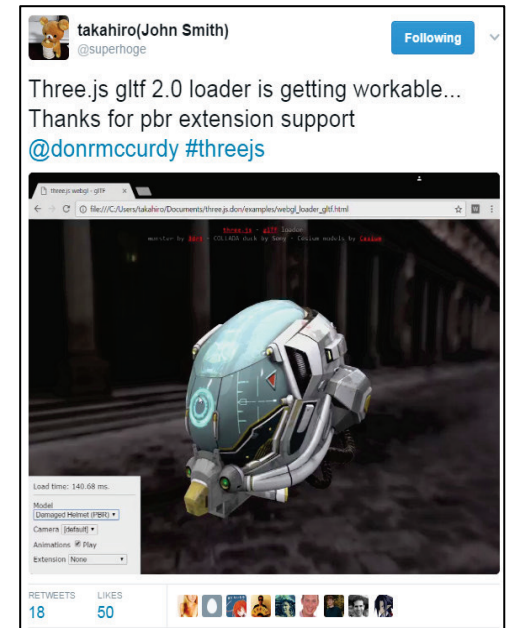


Sketchfab User: theblueturtle
<https://sketchfab.com/models/b81008d513954189a063ff901f7abfe4>

Rapid Transition to glTF 2.0

- There are breaking changes from 1.0 to 2.0 - but processing is streamlined and simplified
 - Overwhelming community feedback to take the pain now - NOT significant work to upgrade
- Industry moving quickly to glTF 2.0 — lots of early adopters
 - BabylonJS, three.js, Cesium, xeogl, instant3Dhub
- [gltf-pipeline](#) includes glTF 2.0 updates — including glTF 1.0 to glTF 2.0 translator
 - Open source - use this to support both glTF 1.0 and 2.0 or move your users to 2.0
- Converters/Translators/Validators glTF 2.0 updates nearly ready
 - [COLLADA2GLTF](#) and [obj2gltf](#) translators
 - Khronos [Validator](#) and [Gltf-test](#)
- Samples and Tutorials
 - glTF [2.0 sample models](#) with PBR are emerging
 - Extensive [glTF tutorial series](#) in draft

**Consider moving your
pipeline to glTF 2.0 ASAP**



Blender glTF 2.0 Exporter RFQ

- Khronos-funded project to bring glTF 2.0 export to Blender
 - Blender, has some early work on glTF export: <https://github.com/Kupoman/blendergltf>
 - Project is to build out that work to glTF 2.0
 - Resultant code is to be contributed, royalty-free to the Blender open source project
- RFQ Milestones
 - 1. February 27th — Khronos Releases RFQ
 - 2. March 17th — RFQ responses received by Khronos
 - 3. March 24th — Contractor selected and notified
 - 4. March 29th — Contract executed and start of work

Details here:

<https://www.khronos.org/rfq/>

Please Consider Making a Bid!



glTF 2.0 PBR Rendering - Image courtesy Fraunhofer

glTF Roadmap Discussions

- **Mesh Compression**
 - Google Draco team
- **Progressive Geometry Streaming**
 - Fraunhofer SRC
- **Unified Compressed Texture Format for Transmission**
 - Basis format from Binomial
 - Optimized transmission format with efficient local expansion to any GPU format
- **Lighting Extension**
 - Enhanced lighting control
- **Extensions for API and language specifics**
 - Optional hooks for enhanced perf/functionality
 - Vulkan, DX12, Metal, GLSL, HLSL, SPIR-V, Metal C++

Share your roadmap priorities with us!

<https://github.com/KhronosGroup/glTF>



glTF 2.0 PBR Rendering - Image courtesy instant3Dhub / instantUV - Max Limper