

# KHRONOS CROSS-PLATFORM STANDARDS UPDATE

Vulkan, SPIR-V, OpenXR, glTF, and OpenCL

Neil Trevett, GTC, 20<sup>th</sup> March 2019



# Khronos Mission



Khronos members are **industry leaders** from around the world that join to **safely** cooperate - to advance their own businesses and the industry as a whole



Khronos is an **open**, member-driven industry consortium developing **royalty-free standards**, and vibrant ecosystems, to harness the power of **silicon acceleration** for demanding **graphics** rendering and **computationally intensive** applications

# Active Khronos Open Standards

COLLADA



3D Asset Authoring  
and Delivery

Parallel Computation, Vision,  
Machine Learning and Inferencing



**KHRONOS**  
GROUP



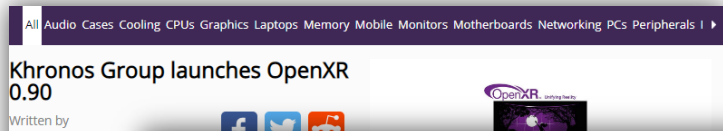
High-performance  
3D Graphics

High-performance access to AR  
and VR platforms and devices





# OpenXR 0.90 Provisional Spec Released This Week!



## Khronos Group releases early OpenXR spec for AR and VR hardware standards

DEAN TAKAHASHI @DEANTAK MARCH 18, 2019 6:02 AM

### Companies P



Above: Khronos Group has all the...  
Image Credit: Khronos Group

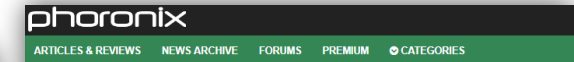
## Khronos' OpenXR 0.90 Provisional Spec For VR & AR Devices Launched At GDC 2019

OpenXR aims to solve the issue of XR fragmentation.

By Peter Graham Last updated Mar 18, 2019 0 Comments

Game Developers Conference 2019 GDC 2019 OpenXR The Khronos Group

Ever since the Khronos Group announced back in 2016 plans to have an open standard for virtual reality (VR) and augmented reality (AR) devices called OpenXR, the initiative has garnered more and more support. With the first public demos showcased at SIGGRAPH last year, for the start of the Game Developer Conference (GDC) 2019 the group has publicly launched the OpenXR 0.90



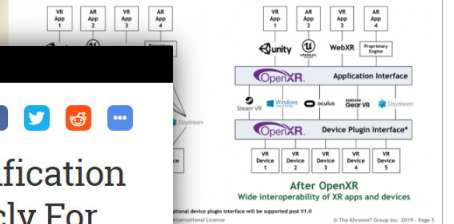
## OpenXR 0.90 Released For AR/VR Standard - MonoD Is An Open-Source Implementation

Written by Michael Larabel in Standards on 18 March 2019 at 09:00 AM EDT. 1 Comment

Last year we were expecting The Khronos Group to introduce OpenXR 1.0 for this standard to address fragmentation and provide interoperability in the VR space followed by AR. That debut last year didn't happen although they did show off the first demonstration at SIGGRAPH. This week though at GDC they are announcing the OpenXR 0.90 provisional specification release.

The OpenXR 0.90 provisional specification is now available today. Yes, v0.90 and not 1.0... This caught me by surprise too when being briefed last week. This provisional specification ended up incorporating not only VR support but also AR (augmented reality) into the design. They are hoping for more feedback from AR/VR developers before officially declaring 1.0 especially with the AR support squeezing in when originally they only anticipated to get that in post-1.0.

### OpenXR - Solving XR Fragmentation



## GDC 2019: OpenXR Specification And API Released Publicly For AR And VR Devices



by IAN HAMILTON • MARCH 18TH, 2019



<https://uploadvr.com/openxr-provisional-release/>  
<https://bit-tech.net/news/tech/software/khronos-group-launches-openxr-090-1/>  
[https://www.phoronix.com/scan.php?page=news\\_item&px=Khronos-OpenXR-0.90](https://www.phoronix.com/scan.php?page=news_item&px=Khronos-OpenXR-0.90)  
<https://www.roadtovr.com/openxr-0-9-provisional-release-microsoft-oculus-collabora-implementations/>  
<https://venturebeat.com/2019/03/18/khronos-group-releases-early-openxr-spec-for-ar-and-vr-hardware-standards/>  
<https://www.vrfocus.com/2019/03/khronos-openxr-0-90-provisional-spec-for-vr-ar-devices-launched-at-gdc-2019/>

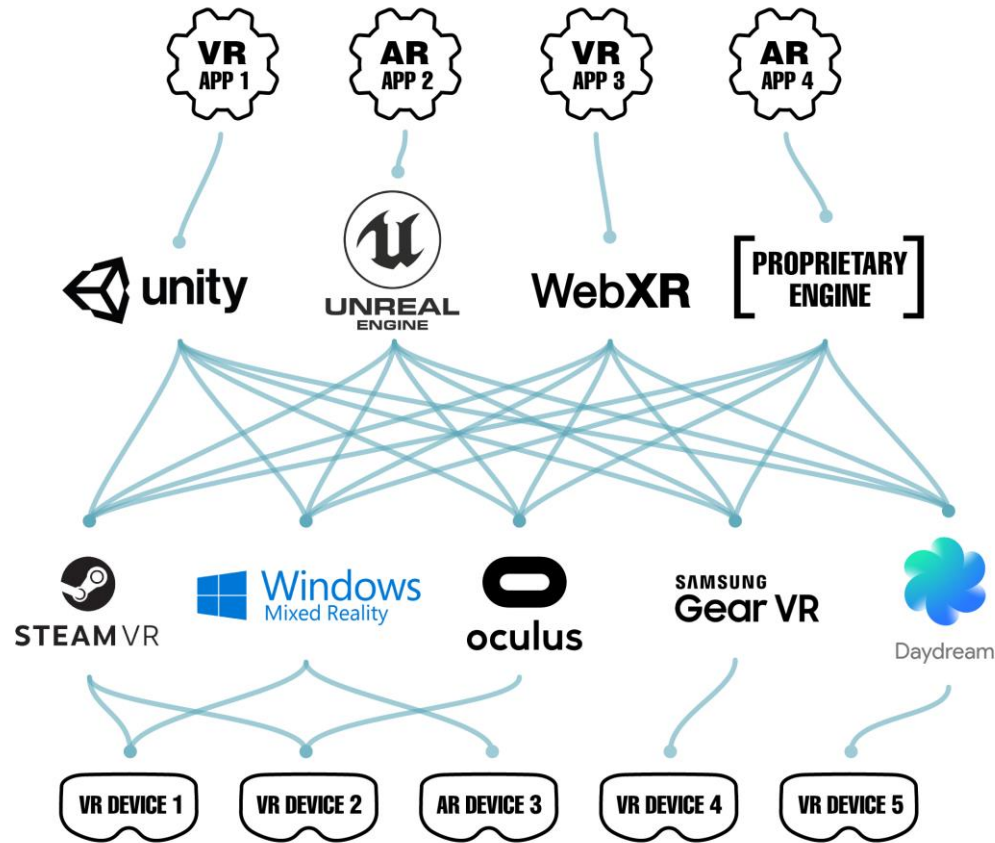
$$XR = AR + VR$$

Virtual Reality



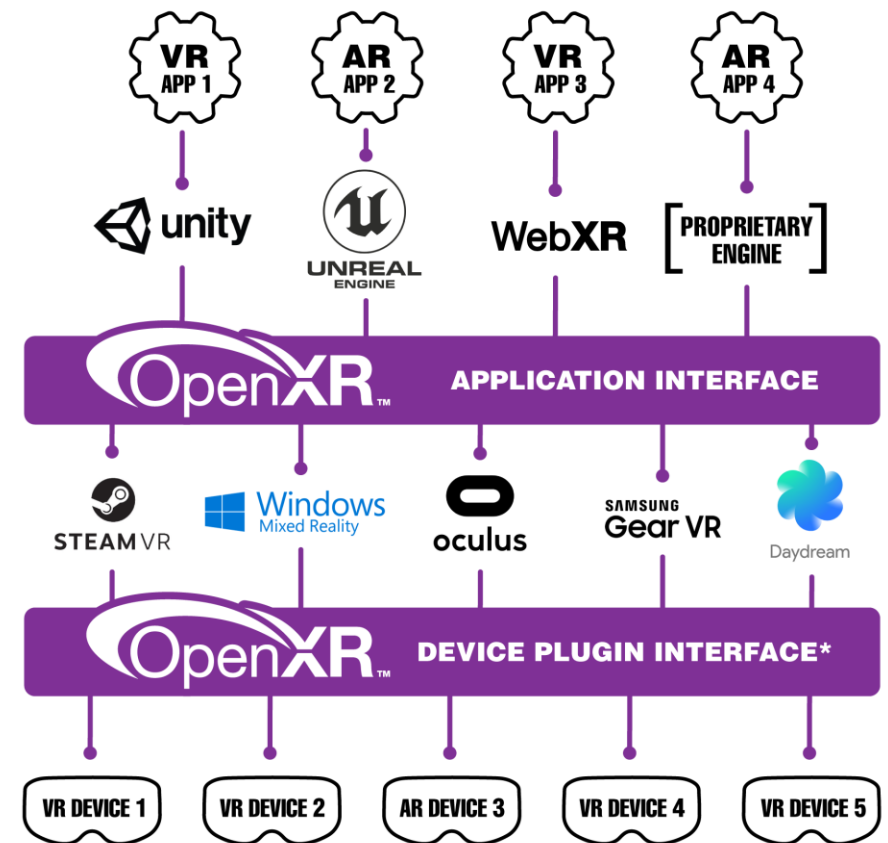
Augmented Reality

# OpenXR - Solving XR Fragmentation



**Before OpenXR**

XR Market Fragmentation



**After OpenXR**

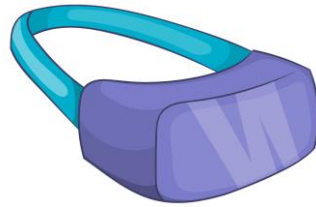
Wide interoperability of XR apps and devices

\* OpenXR 1.0 is focused on enabling cross-platform applications. Optional device plugin interface will be supported post V1.0

# OpenXR Win-Win-Win

## XR Vendors

Can bring more applications onto their platform by leveraging the OpenXR content ecosystem



## XR End-Users

Can run the apps they want on their system  
- reducing market confusion and increasing consumer confidence



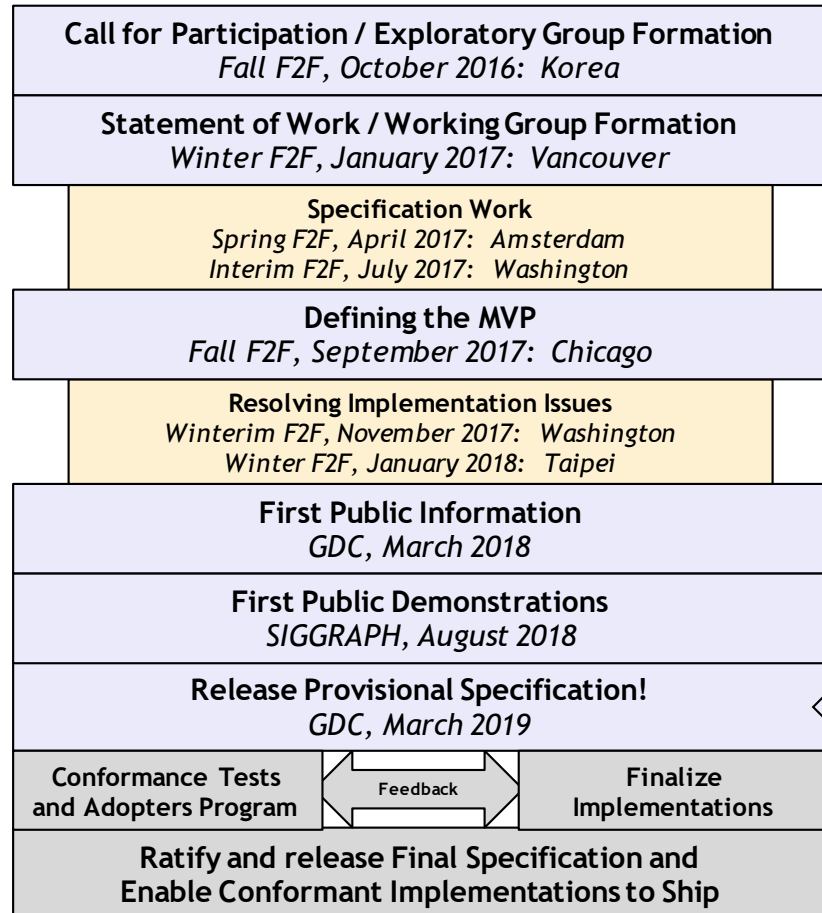
## XR ISVs

Can easily ship on more platforms for increased market reach



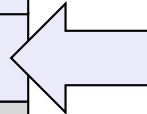
OpenXR does not replace XR Runtime Systems!  
It enables any XR Runtime to expose CROSS-PLATFORM APIs to access their functionality

# OpenXR Specification Released!



High-performance access to AR  
and VR platforms and devices

**OpenXR 0.90 Provisional Specification Released**  
**Enables industry review and feedback**  
**First prototype implementations shipping**





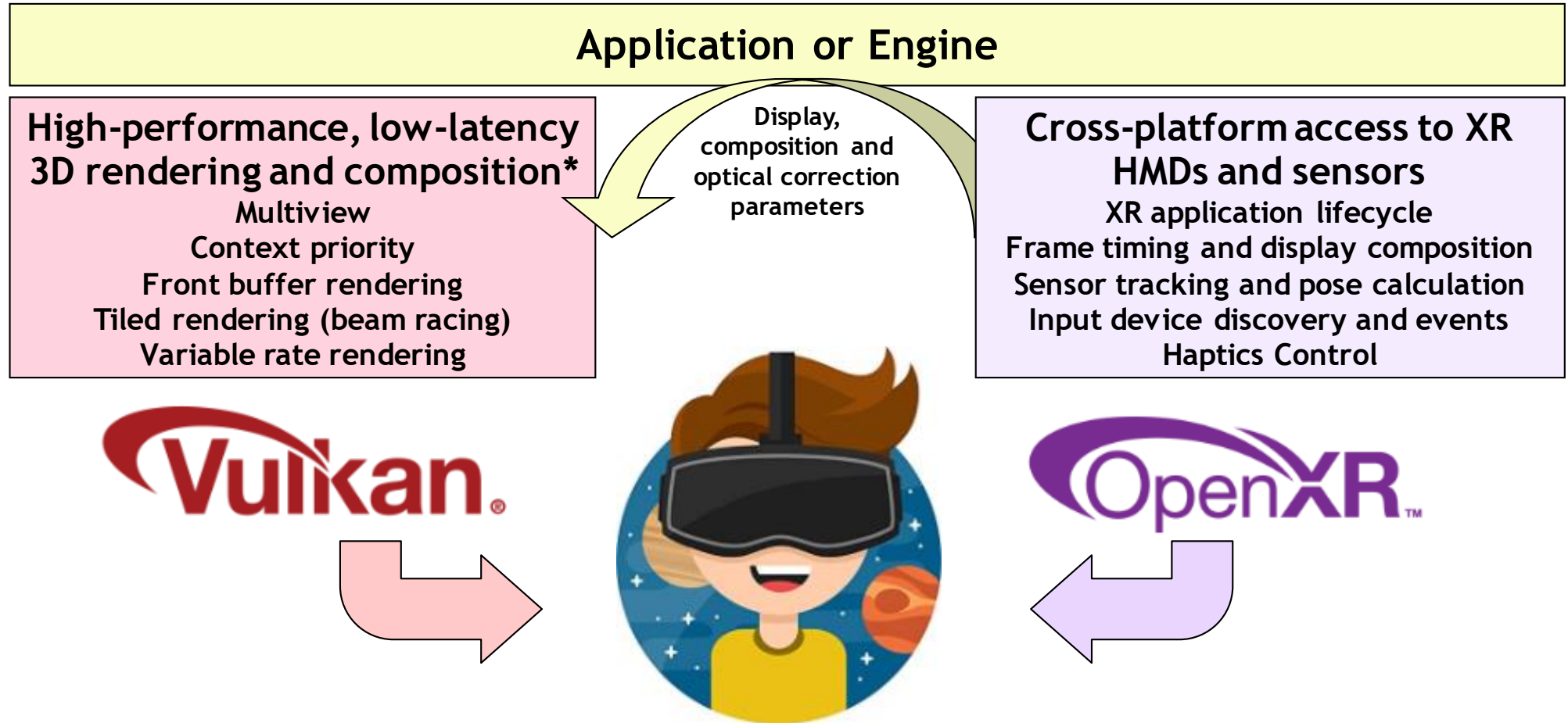
# Companies Publicly Supporting OpenXR



OpenXR is a collaborative design

Integrating many lessons from proprietary 'first-generation' XR API designs

# Khronos APIs for XR



\* OpenXR can be used with other 3D APIs such as Direct3D, OpenGL and OpenGL ES

# Vulkan and New Generation GPU APIs

Modern architecture | Low overhead | Multi-thread friendly  
EXPLICIT GPU access for EFFICIENT, LOW-LATENCY,  
PREDICTABLE performance



Non-proprietary, royalty-free open standard 'By the industry for the industry'  
Portable across multiple platforms - desktop and mobile

\* Vulkan 1.1 runs on Android API level 24 and higher and is a requirement in all 64-bit devices running Android Q and higher

# Pervasive Vulkan



Major GPU Companies supporting Vulkan for Desktop and Mobile Platforms



<http://vulkan.gpuinfo.org/>

## Platforms



Desktop



Mobile  
(Android 7.0+)



Media Players



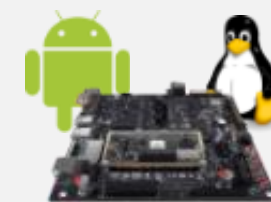
Consoles



Virtual Reality



Cloud Services



Embedded

## Game Engines







# Vulkan 1.1 Ecosystem Evolution

## Strengthening Tools and Compilers

Improved developer tools (SDK, validation/debug layers)  
Shader toolchain improvements (size, speed, robustness)  
Shading language flexibility - HLSL and OpenCL C support  
More rigorous conformance testing

## Building Vulkan's Future

Listen and prioritize developer needs  
Drive GPU technology

### Released Vulkan 1.1 Extensions

Memory Model  
Reduced precision arithmetic types in shaders  
Detailed driver property queries  
New synchronization primitives  
Memory residency management

<https://www.khronos.org/registry/vulkan/specs/1.1-khr-extensions/html/vkspec.html#extension-appendices-list>

### Roadmap Discussions

Cross-vendor performance counter queries  
Variable-resolution rendering  
Video processing  
Machine Learning  
Ray tracing

### Vulkan 1.0 Extensions

Maintenance updates plus additional functionality

Multiview  
Multi-GPU

Enhanced Windows System Integration  
Increased Shader Flexibility:  
16 bit storage, Variable Pointers  
Enhanced Cross-Process and  
Cross-API Sharing



March 2018  
Vulkan 1.1

Integration of 1.0 Extensions  
plus new functionality  
e.g. Subgroup Operations

## Widening Platform Support

Pervasive GPU vendor native driver availability  
Open source drivers - ANV (Intel), AMDVLK/RADV (AMD)  
Vulkan Portability to macOS/iOS and DX12



February 2016  
Vulkan 1.0

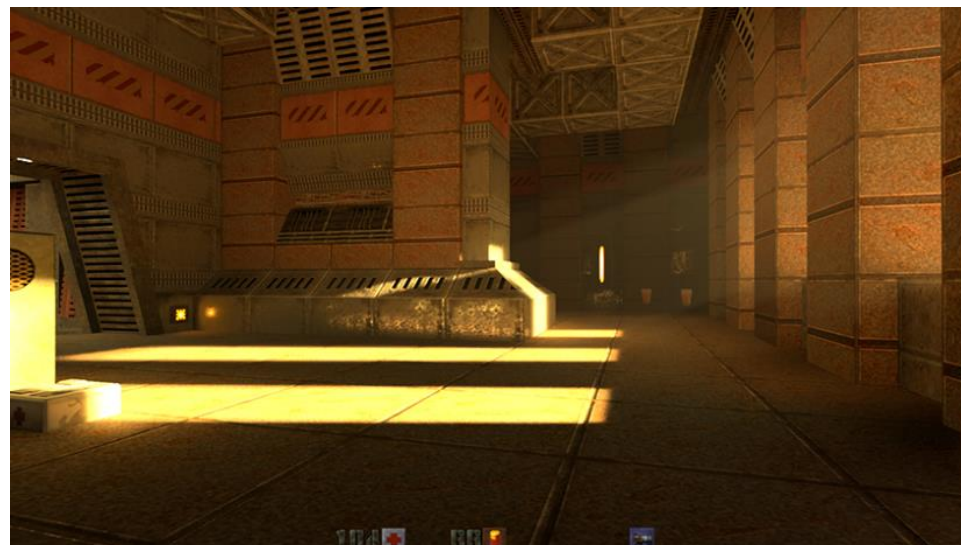
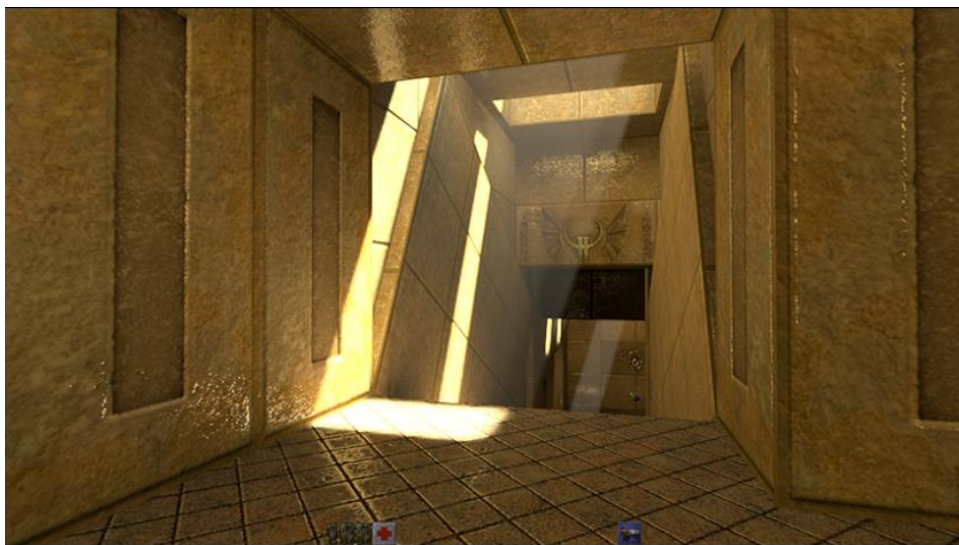
# NVIDIA VKRAY

## Bringing RTX Raytracing to Vulkan

NVIDIA Vulkan Vendor extension - VK\_NV\_ray\_tracing

Similar programming interface to DXR

<https://devblogs.nvidia.com/vulkan-raytracing/>



<https://www.nvidia.com/en-us/geforce/news/quake-ii-rtx-ray-tracing-vulkan-vkcray-geforce-rtx/>

# NSIGHT TOOLS

## Increasing Vulkan Support

Nsight Graphics 2019.2 adds Vulkan Range Profiling

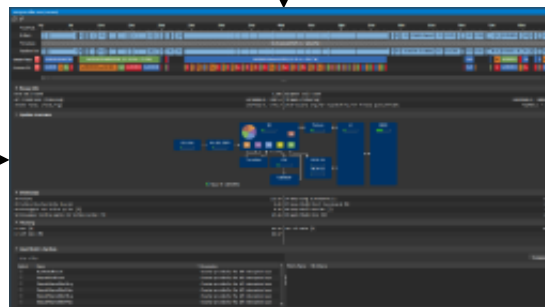
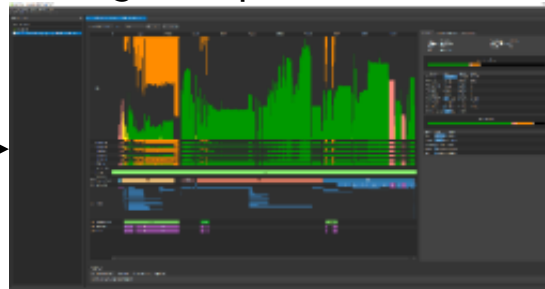
NVIDIA VKRay debugging & profiling on Turing

Available on Linux and Windows



Nsight Systems

Nsight Graphics GPU Trace



Nsight Graphics Range Profiler



Nsight Graphics Shader Profiler

# Vulkan Portability Initiative

Enabling Vulkan applications on platforms without native drivers by layering cleanly queryable subsets of Vulkan over DX12, Metal and other APIs

## Multiple Layered Vulkan Implementations

Additional open source run-times over additional backends  
E.g. Mozilla's gfx-rs for Vulkan over Metal and DX12 is in beta.  
Useful for Vulkan on UWP platforms such as Windows 10 S, Polaris, Xbox One.  
Secondary backends include OpenGL/D3D11

<https://github.com/gfx-rs/gfx>  
<https://github.com/gfx-rs/portability>



## Vulkan Portability Extension

Layered implementations can portably expose to apps what Vulkan functionality is not supported

## Extend Vulkan Conformance Test Suite

To handle layered implementations - test only what is present - but what is present must work!

## Enhanced Vulkan Layers

Extend DevSim/Validation Layers to flag or simulate queries for features not present in a targeted layered implementation



**TODAY**

Open source tools, SDKs and libraries to bring Vulkan 1.0 applications to Apple using Metal





# Vulkan Portability Initiative on Apple

Almost all mandatory Vulkan 1.0 functionality is supported:

- No Triangle Fans
- No separate stencil reference masks
- Events are not supported

Selected Optional Features and Extensions are added as required - driven by industry input and feedback

- Robust buffer access
- BC texture compressed formats
- Fragment shader atomics

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

**Vulkan.**  
Applications

Open source SDK to build, run, and debug applications on macOS - including validation layer support  
<https://vulkan.lunarg.com/>

**Vulkan  
macOS SDK**



**SPIRV-Cross**  
Convert SPIR-V shaders to Metal Shaders



**macOS / iOS  
Run-time**  
Maps Vulkan to Metal

MoltenVK supports macOS 10.11 / iOS 9.0 and up



Open source beta release for macOS



**OPEN SOURCE.**  
Free to use - no fees or royalties - including commercial applications

# Apps Shipping On Apple with Vulkan Backend

**Forsaken Remastered** was just updated with **Vulkan** support! If you're on Linux, you're probably hitting 60fps with the existing OpenGL renderer, but it's good to be future proof. If you're on a Mac, though, you *definitely* want to switch. On my MacBook, the framerate goes from around 15 to a solid 60!

## Initial Vulkan Performance On macOS With Dota 2 Is Looking Very Good

Written by Michael Lambel in Valve on 1 June 2018 at 05:37 PM EDT, 34 Comments



Yesterday Valve released Vulkan support for Dota 2 on macOS. Indeed, this first major game relying upon MoltenVK for mapping Vulkan over the Apple Metal drivers is delivering performance gains.

## Valve Releases Artifact As Its Cross-Platform, Vulkan-Powered Digital Card Game

Written by Michael Lambel in Valve on 28 November 2018 at 04:16 PM EST, 29 Comments



Valve managed to ship their latest game today as planned and without any major delays.

Artifact is now available with launch-day support for Linux, macOS, and Windows. Artifact is a competitive digital card game, a targeted Dota 2-esque intellectual card game experience. Valve still plans to evolve Artifact and its ecosystem.

Multiple iOS and macOS apps organically ported - support through MoltenVK website e.g. Forsaken Remastered on Mac



Production Dota 2 on Mac Ships - up to 50% more perf than Apple's OpenGL



First iOS Apps using MoltenVK ship through app store



Qt Running on Mac through MoltenVK



WINE

Google Filament PBR Renderer on Mac



Initial ports of Wine games in progress using Vulkan on Mac



ARTIFACT

Artifact from Steam ships on MoltenVK on macOS - first Vulkan-only Valve app on Mac

RPCS3

RPCS3 PlayStation 3 Emulator on Mac

Dolphin

GameCube and Wii Emulator working on MacOS



Diligent Engine runs on MacOS

June 2018

September 2018

November 2018

January 2019

# Vulkan Portability Milestones



MoltenVK  
released in open  
source for macOS  
and iOS



macOS SDK  
released by  
LunarG



gfx-rs DX11  
backend in alpha  
- joins DX12 and  
Metal backends



RENDERDOC™  
RenderDoc  
integrated  
with MoltenVK



Early prototype of  
Portability Extension  
released on public  
GitHub and shipped  
in MoltenVK



Features added to MoltenVK  
since GDC 2018

- 24 new Vulkan extensions
  - Tessellation
- Per-texture swizzling
- Pipeline cache load/store
  - GPU switching
- 3 swapchain images
- Performance improvements

## Future Milestones

CTS 100% Pass for Reported Functionality +  
Completed DevSim/Validation Layers  
-> Portability Extension 1.0 Release  
Follow progress and give feedback at  
<https://github.com/KhronosGroup/Vulkan-Portability>

GDC  
2018

August  
2018

September  
2018

January  
2019

GDC  
2019

# Vulkan AAA Content Shipping on Desktop...

## Vulkan AAA Titles on PC



## AAA titles on Linux

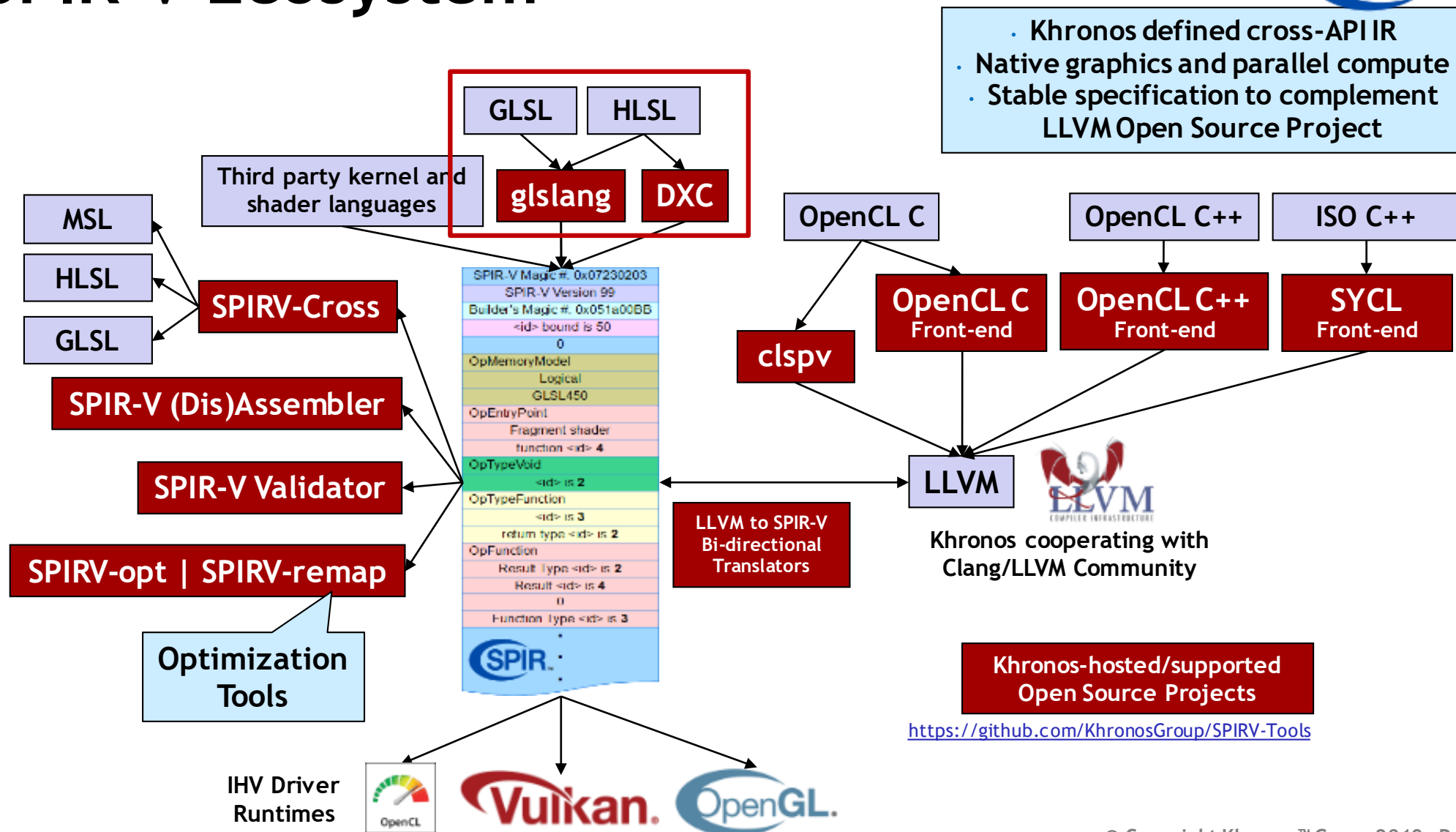


## macOS, SteamOS and PC





# SPIR-V Ecosystem

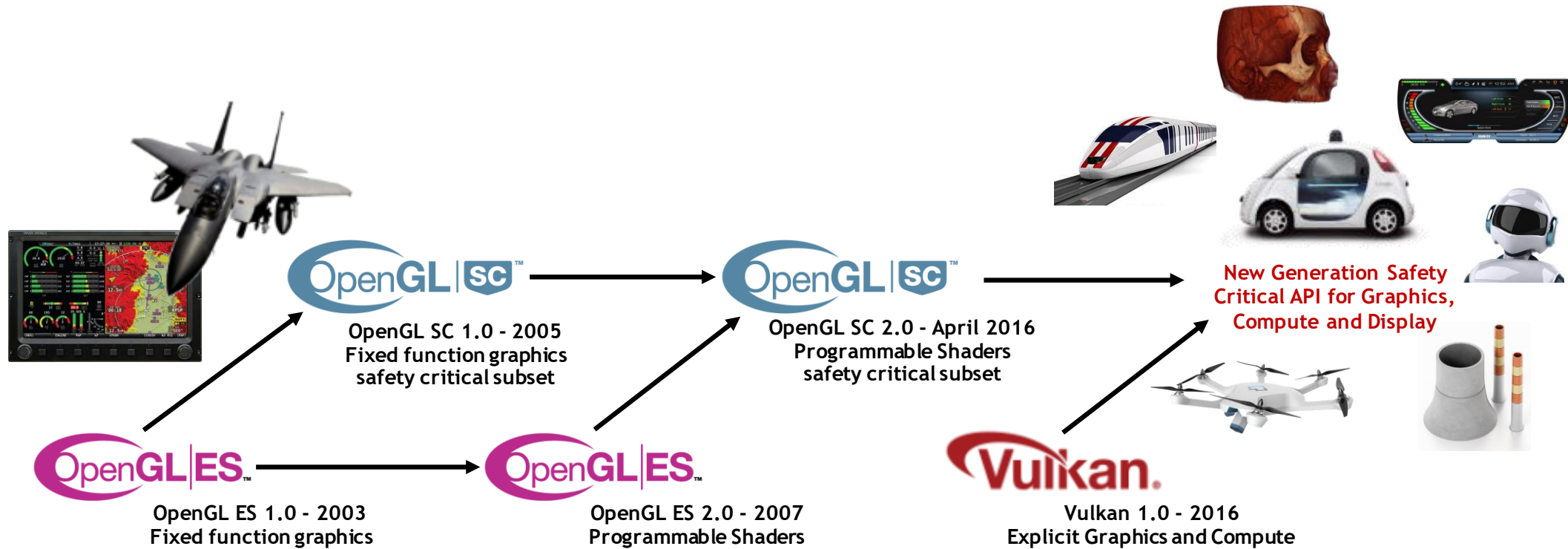


# HLSL and Vulkan

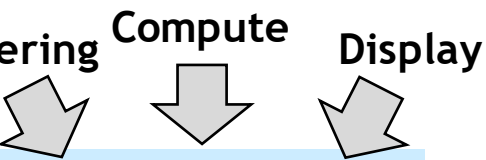
- **Microsoft DXC Shader Compiler**
  - Microsoft's open source shader compiler for HLSL - first released Spring 2017
  - "Spiregg" backend adds SPIR-V code generation - initiated and supported by Google
  - Uses Microsoft's parser, tracks undocumented HLSL, including C++ features
  - <https://github.com/Microsoft/DirectXShaderCompiler>
- **New functionality increasingly being added to both GLSL *and* HLSL through DXC**
  - e.g. NVIDIA has added VKRay raytracing vendor extension to DXC
  - Vulkan Memory Model will be added soon

	glslang w/HLSL Frontend	DXC / Spiregg
Overview	Simple, custom HLSL parser	Uses Microsoft's HLSL parser
Shader Model Support	4, 5.1	5.1, 6
Open Source	Yes	Yes
Generates SPIR-V	Yes	Yes

# Safety Critical GPU API Evolution



**KHRONOS<sup>®</sup> GROUP**




- Widely adopted, royalty-free open standard
- Low-level explicit API - smaller surface area than OpenGL
- Not burdened by debug functionality
  - Very little internal state
  - Well-defined thread behavior



**Vulkan SC Working Group announced February 2019**  
**Any company welcome to join Khronos and participate**

# Clearly Definable Design Goals to Adapt Vulkan for SC

- Reduce driver size and complexity
  - > Offline pipeline creation, no dynamic display resolutions
- Deterministic Behavior
  - > No ignored parameters, static memory management, eliminate undefined behaviors
- Robust Error Handling
  - > Error callbacks so app can respond, Fatal error callbacks for fast recovery initiation
- C API - MISRA C Compliance



- > Offline pipeline creation, no dynamic display resolutions

**Deterministic Behavior**  
-> No ignored parameters, static memory management, eliminate undefined behaviors

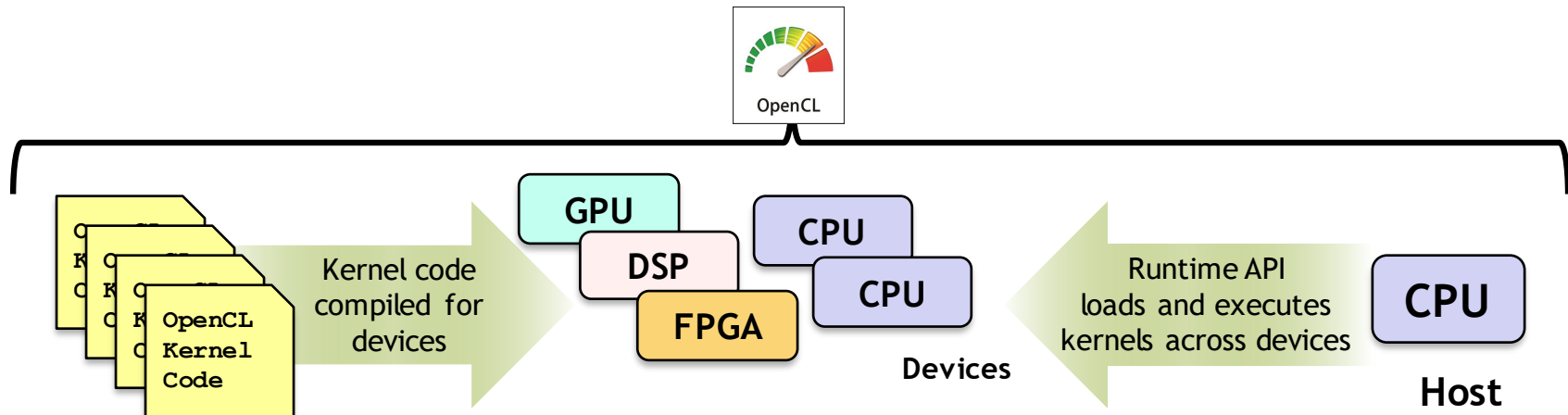
**Robust Error Handling**  
-> Error callbacks so app can respond, Fatal error callbacks for fast recovery initiation

## C API - MISRA C Compliance



# OpenCL - Low-level Parallel Programming

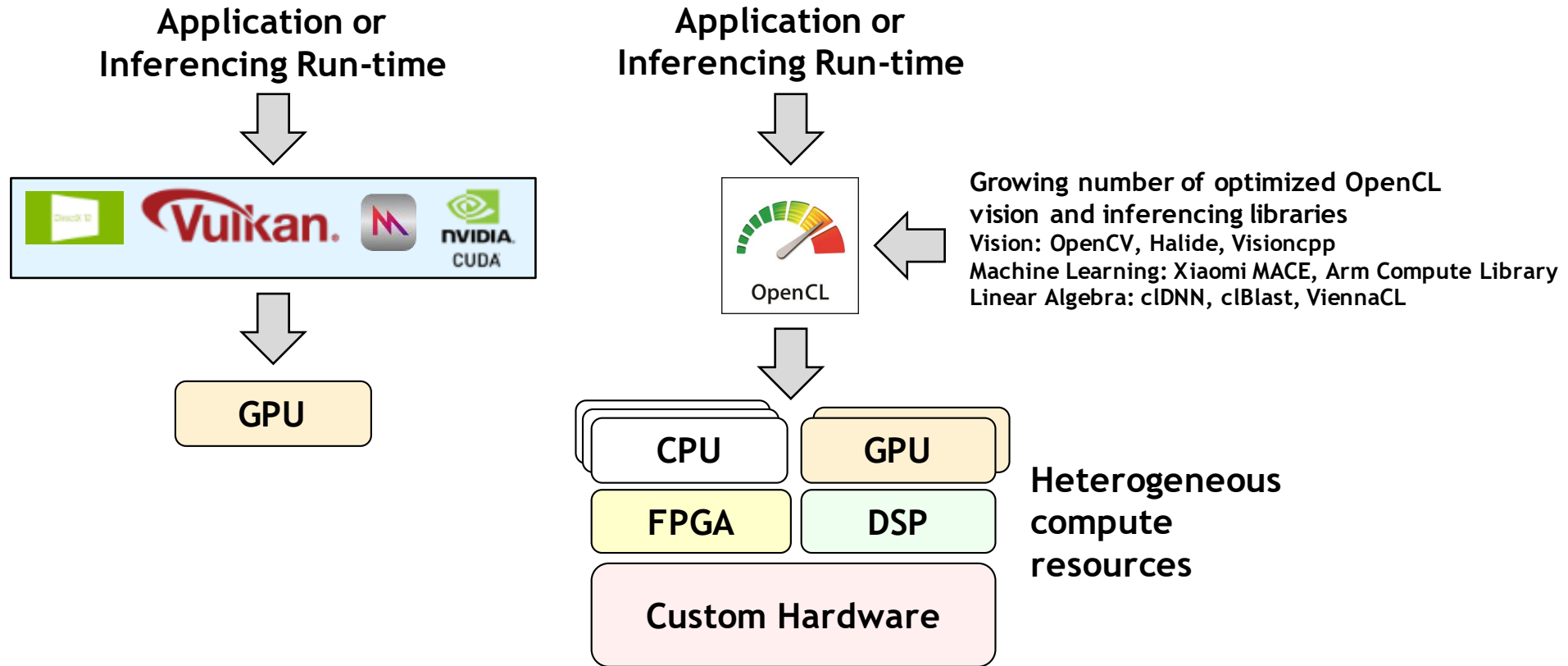
- **Low-level programming of heterogeneous parallel compute resources**
  - One code tree can be executed on CPUs, GPUs, DSPs and FPGA ...
- **OpenCL C or C++ language to write kernel programs to execute on any compute device**
  - Platform Layer API - to query, select and initialize compute devices
  - Runtime API - to build and execute kernels programs on multiple devices
- **The programmer gets to control:**
  - What programs execute on what device
  - Where data is stored in various speed and size memories in the system
  - When programs are run, and what operations are dependent on earlier operations





# OpenCL - Unique Heterogeneous Runtime

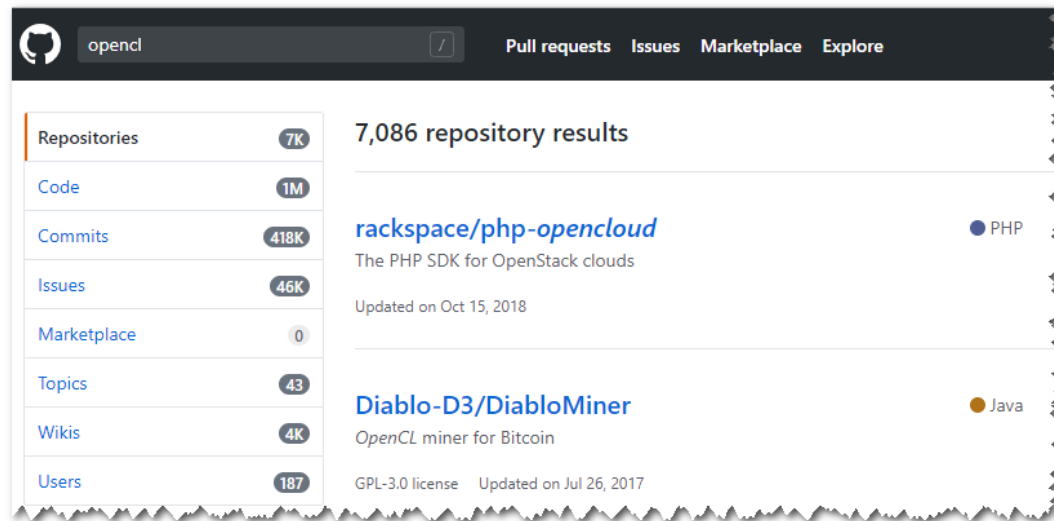
OpenCL is the only industry standard for low-level heterogeneous compute  
Portable control over memory and parallel task execution  
“The closest you can be to diverse accelerator hardware and still be portable”



# OpenCL Industry Adoption

- 100s of applications and libraries using OpenCL
  - Rendering, visualization, video editing, simulation, image processing
- Over 7,000 GitHub repositories using OpenCL
  - Tools, applications, libraries, languages
  - Up from 4,310 18 months ago
- Khronos Resource Hub

<https://www.khronos.org/opencl/resources/opencl-applications-using-opencl>



## Hardware Implementors

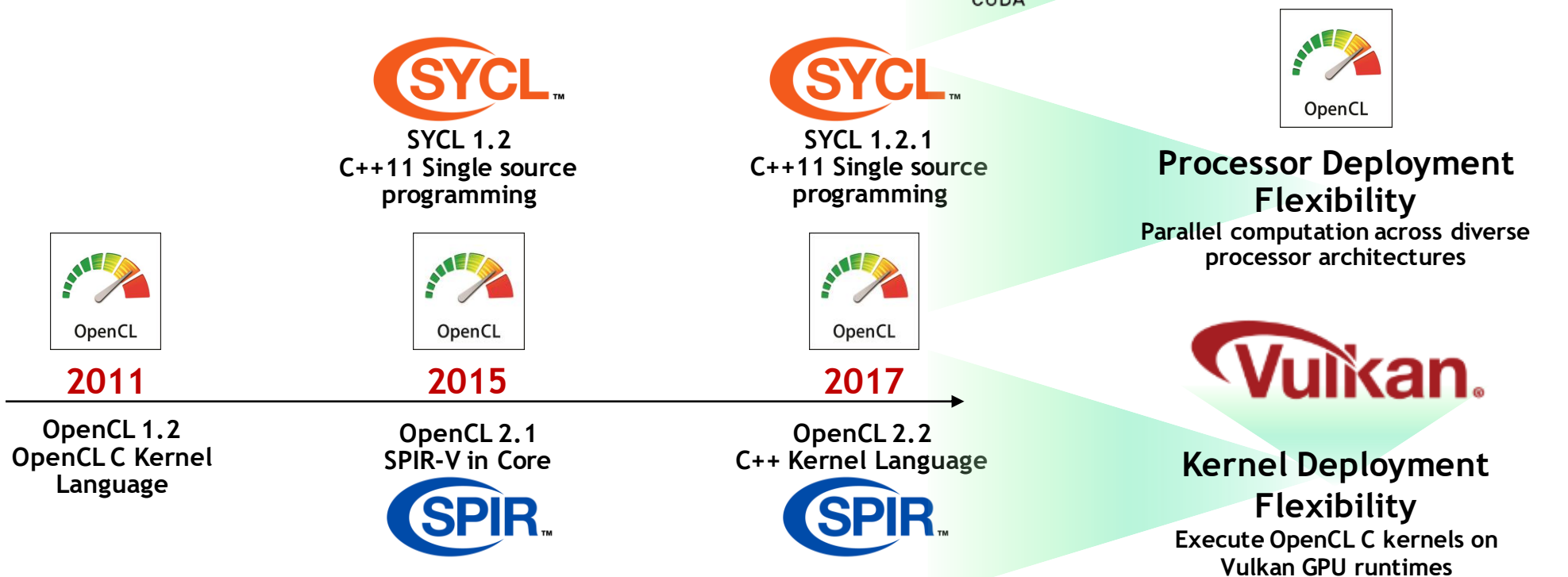


## Sample OpenCL Apps and Libraries



# OpenCL Ecosystem Roadmap

OpenCL has an active  
three track roadmap

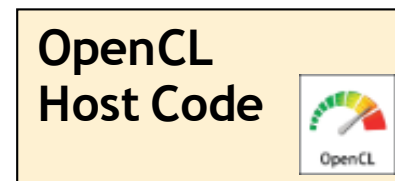
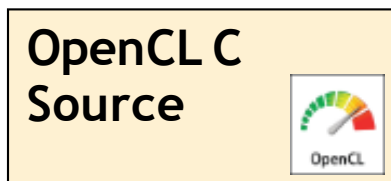


# OpenCL Deployment Flexibility



- Experimental Clspv Compiler from Google, Adobe and Codeplay
  - Compiles OpenCL C to Vulkan's SPIR-V execution environment
  - Tested on over 200K lines of Adobe OpenCL C production code
  - Open source - tracks top-of-tree LLVM and clang, not a fork

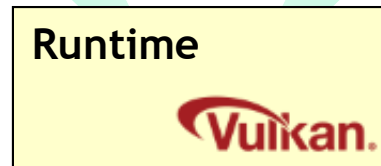
Increasing deployment options  
for OpenCL developers  
e.g. Vulkan is a supported API  
on Android



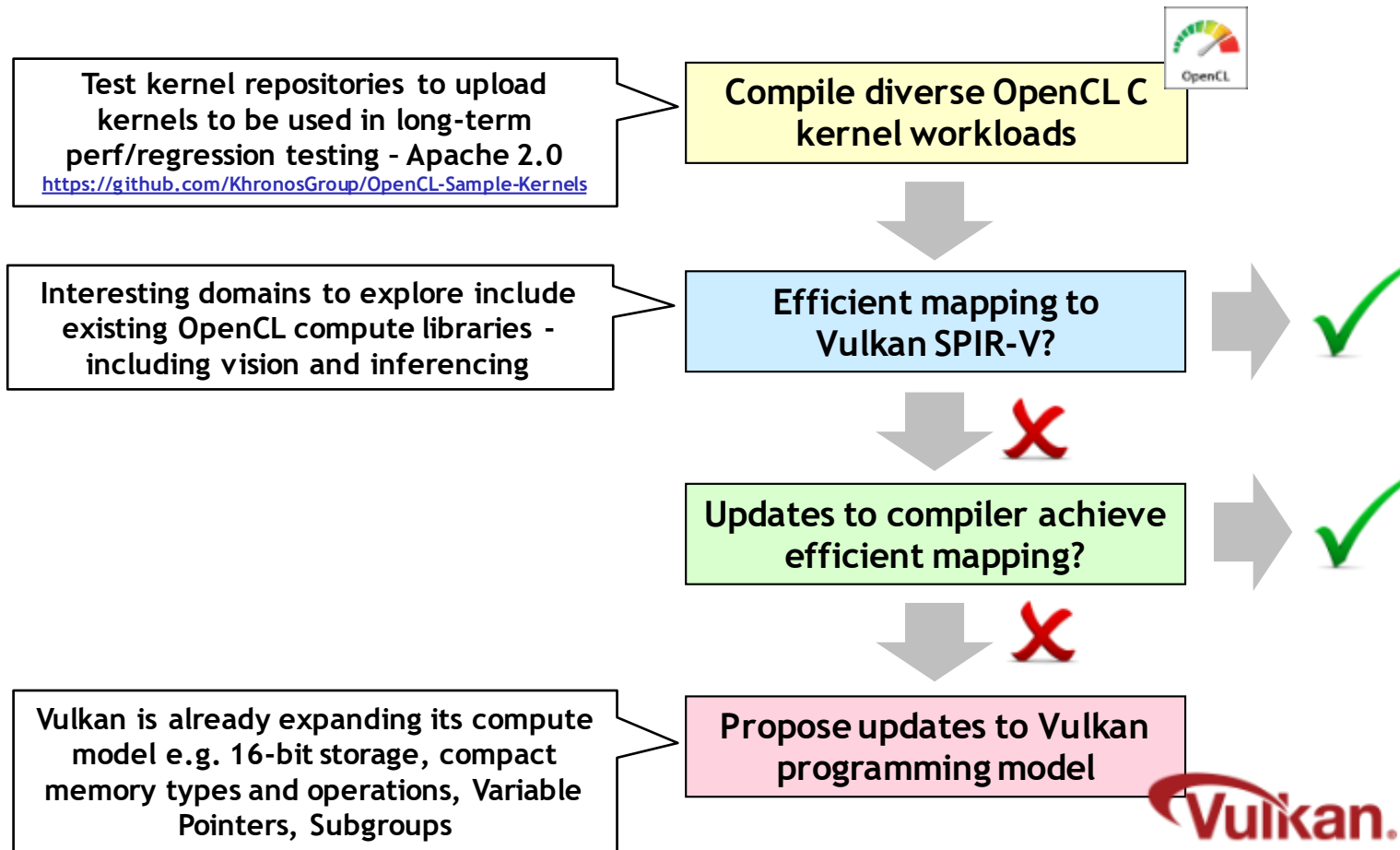
Prototype open  
source project  
<https://github.com/google/clspv>



Prototype open  
source project  
<https://github.com/kpet/clvk>



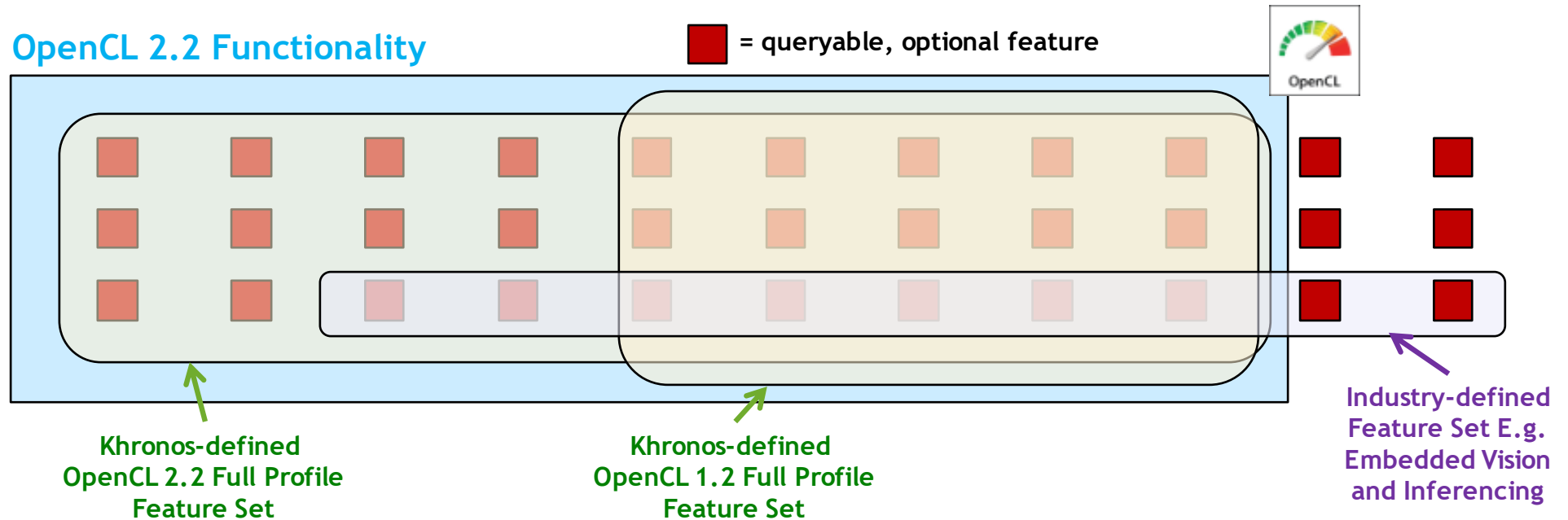
# Refining clspv with Diverse Workloads





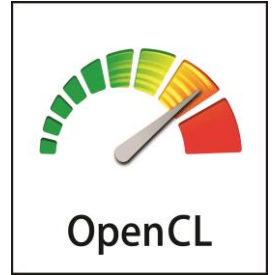
# OpenCL Next - Feature Set Flexibility

- Defining OpenCL features that become optional for enhanced deployment flexibility
  - API and language features e.g. floating point precisions
- Feature Sets avoid fragmentation
  - Defined to suit specific markets - e.g. desktop, embedded vision and inferencing
- Implementations are conformant if fully support feature set functionality



# NVIDIA AND OPENCL

## Active Investment and Support



### Production-class OpenCL 1.2 on Linux and Windows

Active, ongoing improvements in power efficiency and performance

Optimized data transfers and GPU memory allocation (CL\_NV\_CREATE\_BUFFER)

Multi-GPU optimizations, multi-command-queue use case tuning

Turing optimizations: workgroup size and pro-rendering use cases

### Expanding OpenCL Interop options

Today, OpenCL interops with OpenGL and D3D9/10/11

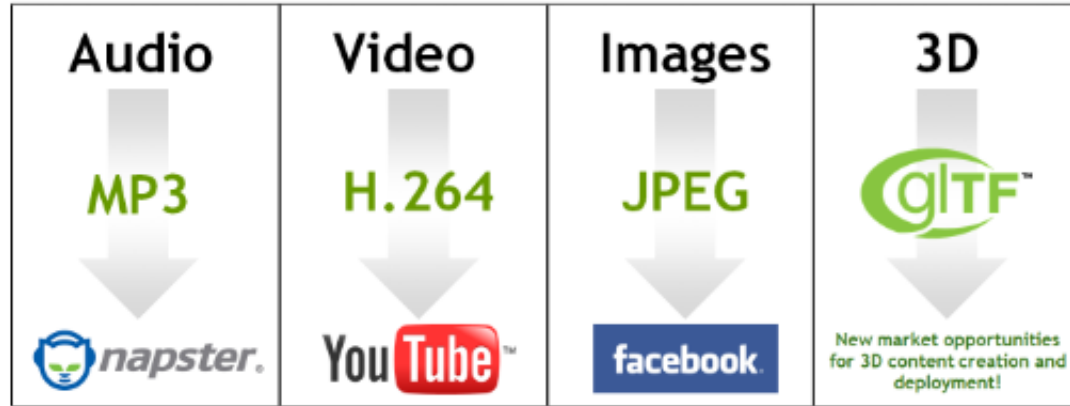
NVIDIA driving Vulkan/OpenCL interop extension at Khronos

Uses standard Vulkan external memory interop hooks

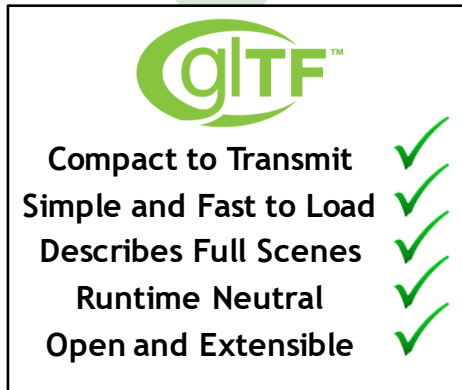
### Beta support for selected OpenCL 2.0 features

Shared Virtual Memory, Device-side-enqueue, Generic address space

# glTF - The JPEG of 3D!



glTF spec development  
on open GitHub - get involved!  
<https://github.com/KhronosGroup/glTF>



Efficient, reliable transmission  
Bring 3D assets into 1000s of  
apps and engines



glTF 1.0 - December 2015  
Primarily for WebGL  
Uses GLSL for materials



glTF 2.0 - June 2017  
Native AND Web APIs  
Physically Based Rendering  
Metallic-Roughness and Specular-Glossiness



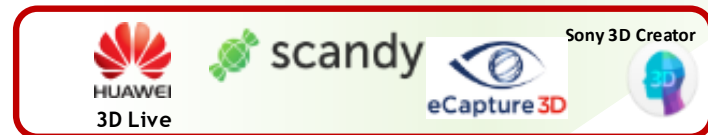
Dedicated 3D Authoring Tools



Authoring Tools that Export 3D



VR / AR Authoring Tools



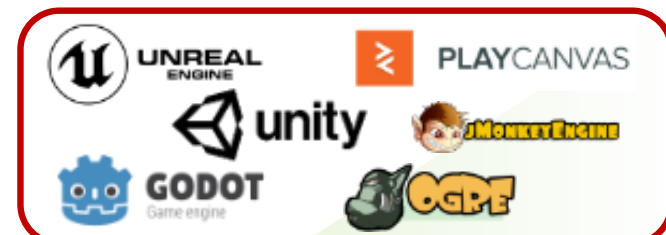
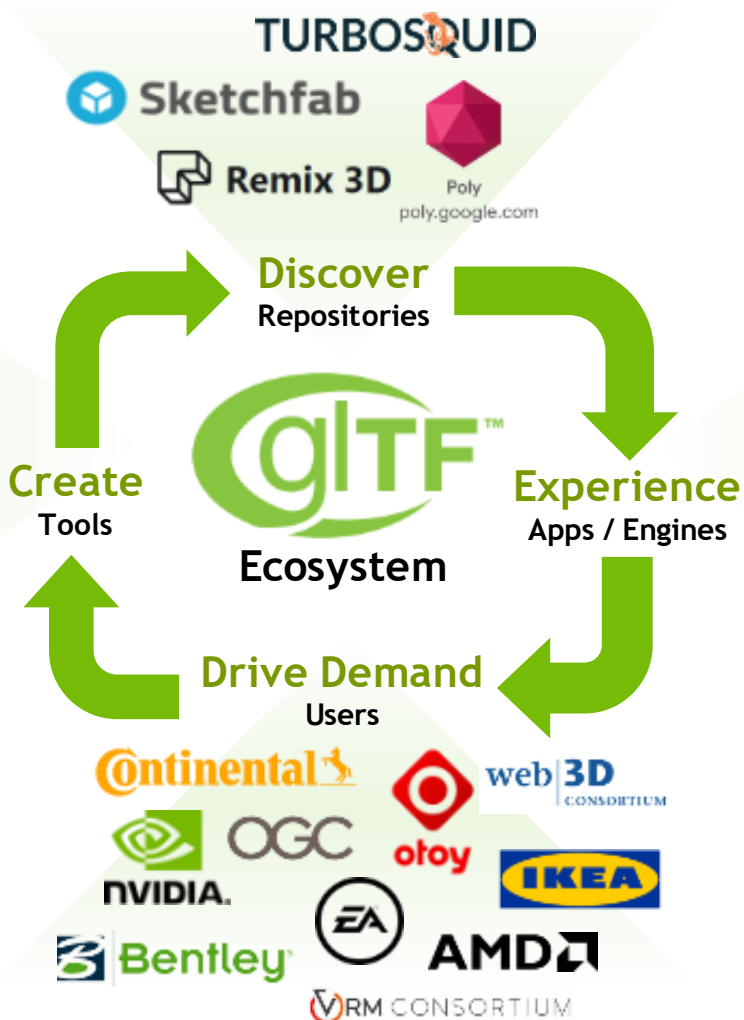
3D Scanning Tools



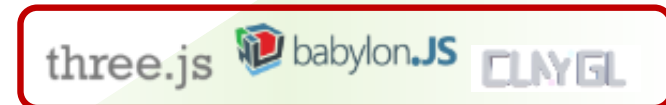
Convertors and Optimizers



Validation and Reference Tools



Game Engines



Web Engines



3D Apps and Engines



VR / AR Apps and Engines



Productivity and Social Apps

# Official Khronos glTF Sample Renderer



- Ecosystem robustness & consistency is key to glTF's mission
  - Promote consistency across engines: shaders, color spaces, ...
- Iron out fine points in the PBR spec
  - Easy to follow reference for PBR formulas
- Open-source JavaScript & WebGL - available for reviewing and testing!
  - <https://github.com/KhronosGroup/glTF-WebGL-PBR/tree/reference-viewer>



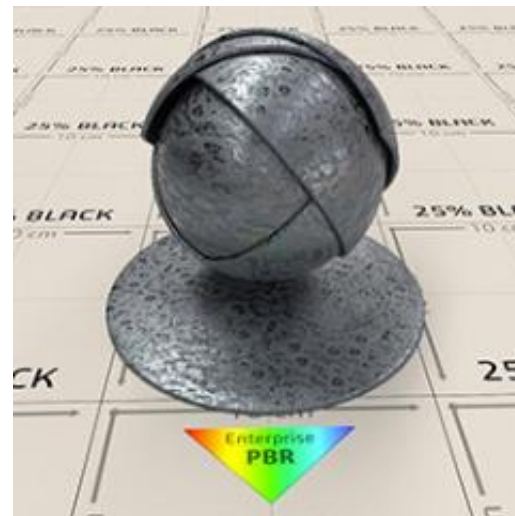


# PBR Next

- **Next-generation Physically-Based Rendering (PBR) materials**
  - Absorption/attenuation, clear coat, subsurface scattering, anisotropy
- **Extend existing Metal Roughness glTF 2.0 PBR parameters**
  - Consistency and fallbacks for performance are key topics
- **Inspiration from Dassault Systèmes Enterprise PBR Shading Model (DSPBR)**
  - [https://github.com/DassaultSystemes-Technology/EnterprisePBRShadingModel/tree/master/glTF\\_ext](https://github.com/DassaultSystemes-Technology/EnterprisePBRShadingModel/tree/master/glTF_ext)
  - 3DS\_materials\_enterprise\_pbr (draft)
- **Collaborating with engine developers:**
  - Dassault Systèmes
  - Google Filament
  - Microsoft BabylonJS
  - OTOY Octane

**Join the GitHub Discussion!**

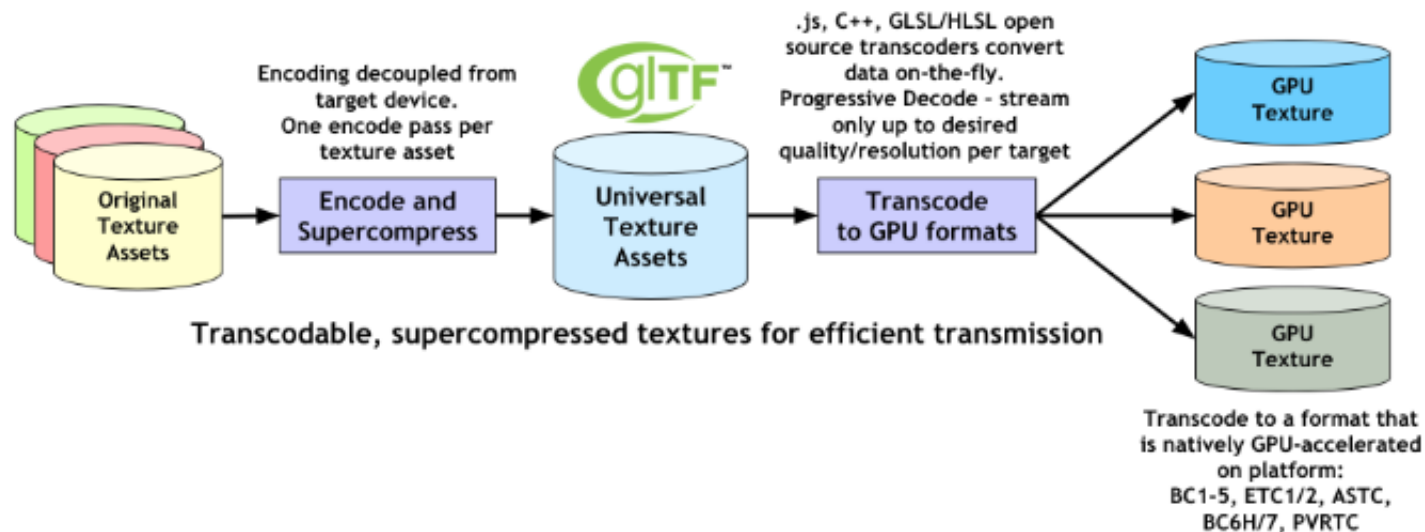
<https://github.com/KhronosGroup/glTF/issues/1442>



Images from <https://dassaultsystemes-technology.github.io/EnterprisePBRShadingModel/>

# Universal Textures: KTX2 & CTF

- Khronos Texture Format 2 (KTX2) is a container format for textures
  - Specification ready for community feedback: [github.com/KhronosGroup/KTX-Specification](https://github.com/KhronosGroup/KTX-Specification)
  - Upcoming open source tools: [github.com/KhronosGroup/KTX-Software](https://github.com/KhronosGroup/KTX-Software)
- Compressed Texture Transmission Format (CTTF)
  - Universal texture compression via runtime transcoding
  - Prototype codec: [github.com/KhronosGroup/glTF-Texture-Transmission-Tools](https://github.com/KhronosGroup/glTF-Texture-Transmission-Tools)
  - Upcoming spec: [github.com/KhronosGroup/CTTF-Specification](https://github.com/KhronosGroup/CTTF-Specification)
  - Call for industry collaboration to support open source tools



# NVIDIA AND KHRONOS API STANDARDS

**Vulkan**



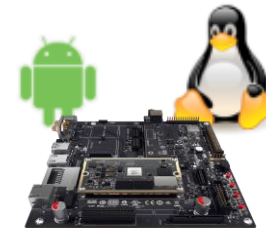
Desktop - Kepler and later



Nintendo Switch



Shield Android TV



Embedded Tegra / Jetson

NVIDIA shipped Vulkan 1.1 across our product range on day of specification release.  
NVIDIA chairing multiple Vulkan initiatives at Khronos: Ray Tracing, Machine-Learning etc.  
Added Vulkan support to NSIGHT Graphics: Frame Debugger Range Profiler, GPU Trace

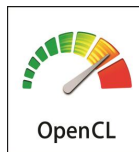
<https://developer.nvidia.com/Vulkan>

**OpenXR**

GRAPHICS	HEADSET	TOUCH & PHYSICS	AUDIO
LENS MATCHED SHADING	CONTEXT PRIORITY	PHYSX	VRWORKS AUDIO
SINGLE PASS STEREO	DIRECT MODE	PROFESSIONAL	VIDEO
MULTIRES SHADING	FRONT BUFFER RENDERING	WARP & BLEND	VRWORKS 360 VIDEO
VR SLI		SYNCHRONIZATION	GPUDIRECT FOR VIDEO
		GPU AFFINITY	

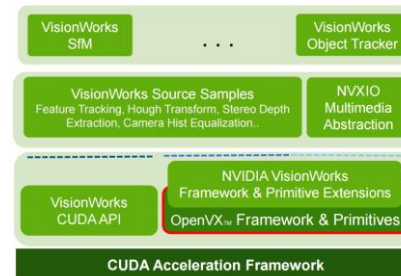
Actively working to define and drive OpenXR to enable and complement VRWorks

<https://developer.nvidia.com/vrworks>



Production-class OpenCL 1.2 on Linux and Windows.  
Active improvements in efficiency and performance.  
Selected OpenCL 2.0 features in beta and Vulkan Interop in development  
<https://developer.nvidia.com/opengl>

**OpenVX**



First to ship production OpenVX 1.0.  
Extended OpenVX used in VisionWorks for Tegra/Jetson

<https://developer.nvidia.com/embedded/visionworks>

# How To Get Involved!

- Any company or organization is welcome to join Khronos!
  - For a voice and a vote in any of these standards - membership starts at \$3,500
- OR request an invite to Vulkan, OpenCL, OpenXR Advisory Panels
  - No fee, execute Khronos NDA and IP Framework, provide requirements and spec feedback
- We welcome your feedback at NVIDIA or Khronos
  - Khronos Forums: <https://forums.khronos.org/>
  - Khronos Slack Channels: <https://khronosdevs.slack.com/messages>
  - Khronos open source GitHub repositories: <https://github.khronos.org/>
- Contact Neil Trevett
  - ntrevett@nvidia.com | @neilt3d | [www.khronos.org](http://www.khronos.org)

