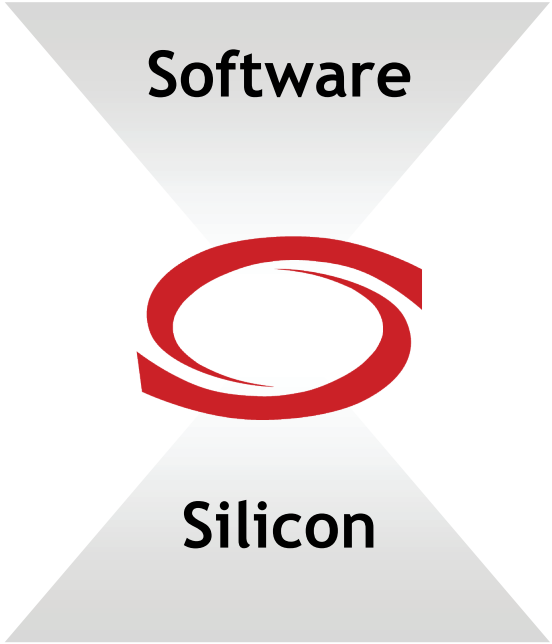
KHRONOS API ECOSYSTEM

Neil Trevett, GTC, 8th May 2017

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

COLLADA

WebGL glTF

3D Portability

Real-time 2D/3D
- Cross-platform gaming and UI
- VR and AR Displays
- CAD and Product Design
- Safety-critical displays

OpenGL|ES

Vulkan

OpenGL|SC

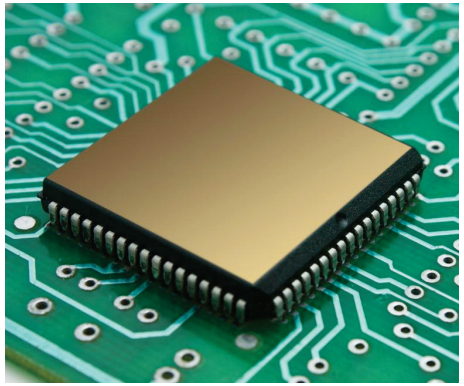
OpenGL

OpenVG

NNEF

OpenVX

OpenXR



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

SPIR

OpenCL

SYCL

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

KHRONOS GROUP

Vulkan and New Generation 3D APIs

Only Windows 10

Only Apple

Cross Platform

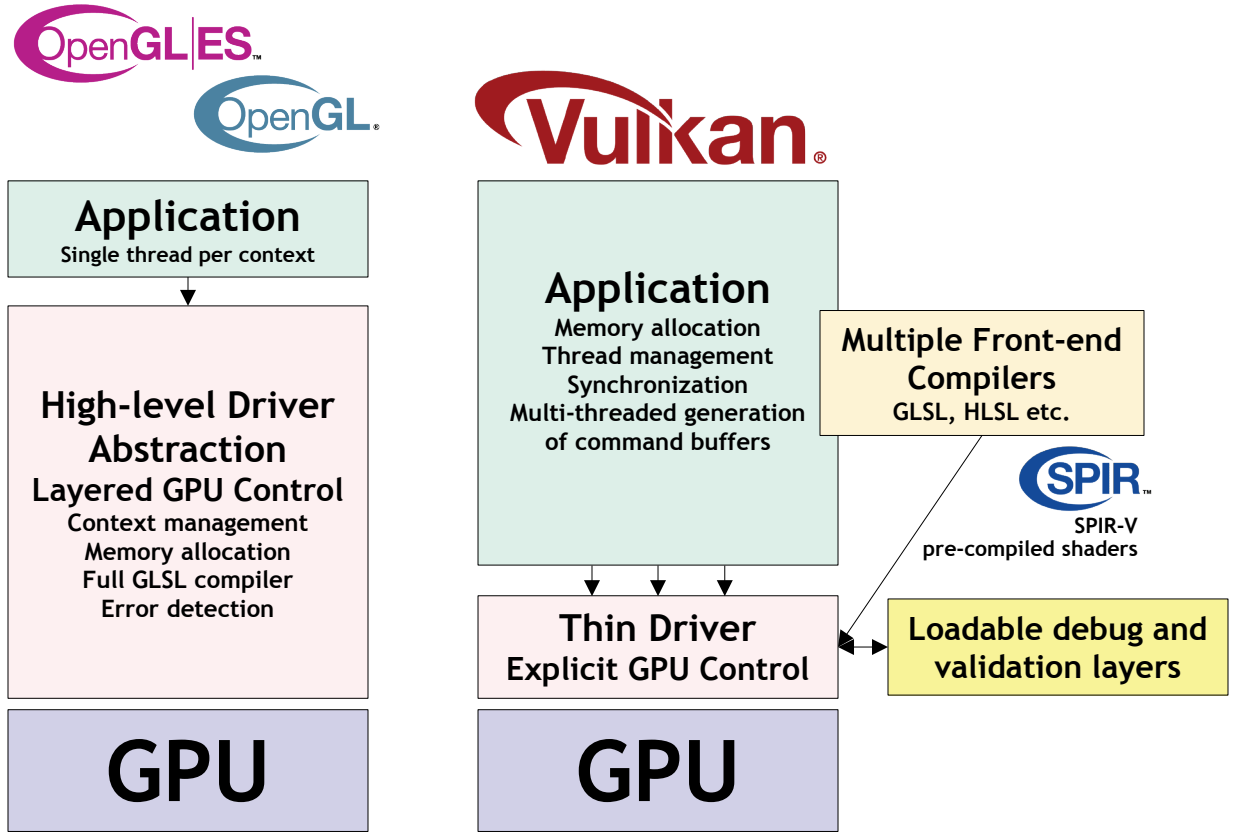
Vulkan Explicit GPU Control

Complex drivers cause overhead and inconsistent behavior across vendors

Always active error handling

Full GLSL preprocessor and compiler in driver

OpenGL vs. OpenGL ES



Vulkan 1.0 provides access to
OpenGL ES 3.1 / OpenGL 4.X-class GPU functionality
but with increased performance and flexibility

Resource management offloaded to app: low-overhead, low-latency driver

Consistent behavior: no 'fighting with driver heuristics'

Validation and debug layers loaded only when needed

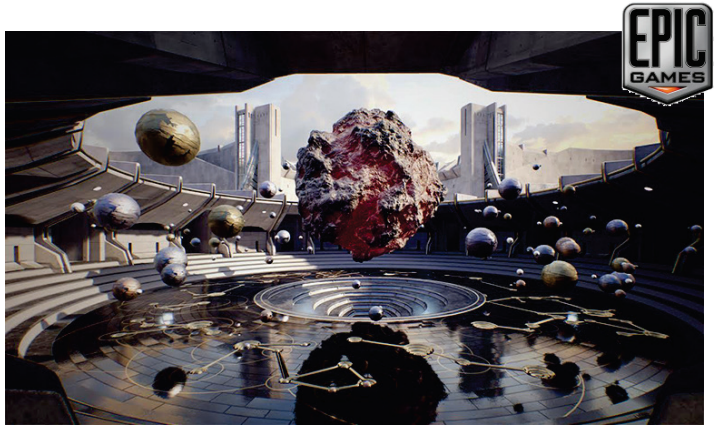
SPIR-V intermediate language: shading language flexibility

Multi-threaded command creation. Multiple graphics, command and DMA queues

Unified API across all platforms with feature set flexibility

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

unity

Vulkan support in Unity 5.6

XENKO Silicon Studio

Vulkan support since V1.8

CRYENGINE CRYTEK

Vulkan support in V5.4 – Late May 2017

Vulkan Momentum

Games Studios publicly confirming that work is ongoing on Vulkan Titles

In first 12months :
#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

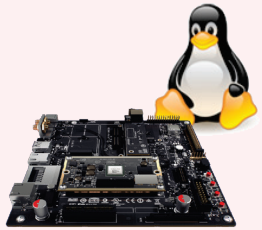
All Major GPU Companies shipping Vulkan Drivers - for Desktop and Mobile Platforms

Mobile, Embedded and Console Platforms Supporting Vulkan

Android 7.0

Nintendo Switch

Android TV



Embedded Linux

Latest Vulkan Functionality

GDC 17

Vulkan 1.0.42 with
new extension sets

Explicit Building Blocks for VR

Multiview
Efficiently Render geometry to multiple surfaces
Each with its own viewing parameters
E.g. render stereo pairs or environment maps

Resource Sharing
Share memory and synchronization primitives
Works across process and instance boundaries

Descriptor Updates
Update resource references between
draw or compute dispatch calls
Efficiently repeatedly update fixed set of resources

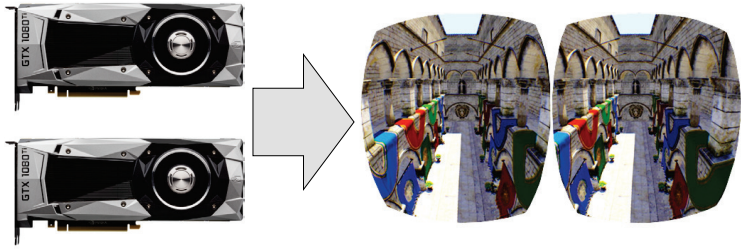
Explicit Building Blocks for Multi-GPU

Works with NVIDIA SLI and AMD Crossfire
Does NOT support dGPU/iGPU

"Device Group" is set of physical devices
Acts as single logical device

Access separate GPUs only for explicit control
Memory allocation and binding resources
Command Buffer Recording/Submission
Synchronization

Enables a variety of operating modes
AFR (alternate frame), SFR (Sequential frame)
VR SLI Stereo view rendering



NVIDIA

Shipped 1.0.42 beta drivers on the day the specifications were released PLUS building block Vulkan extensions for VRWorks on Maxwell and Pascal

Khronos APIs for Virtual Reality

Khronos 3D APIs already powering VR rendering

OpenGL

The standard for desktop VR today

OpenGL|ES

On millions of mobile VR devices

WebGL

Powering WebVR in browsers

Vulkan

The future of explicit, low-latency VR

But What About INPUT Hardware for VR..

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

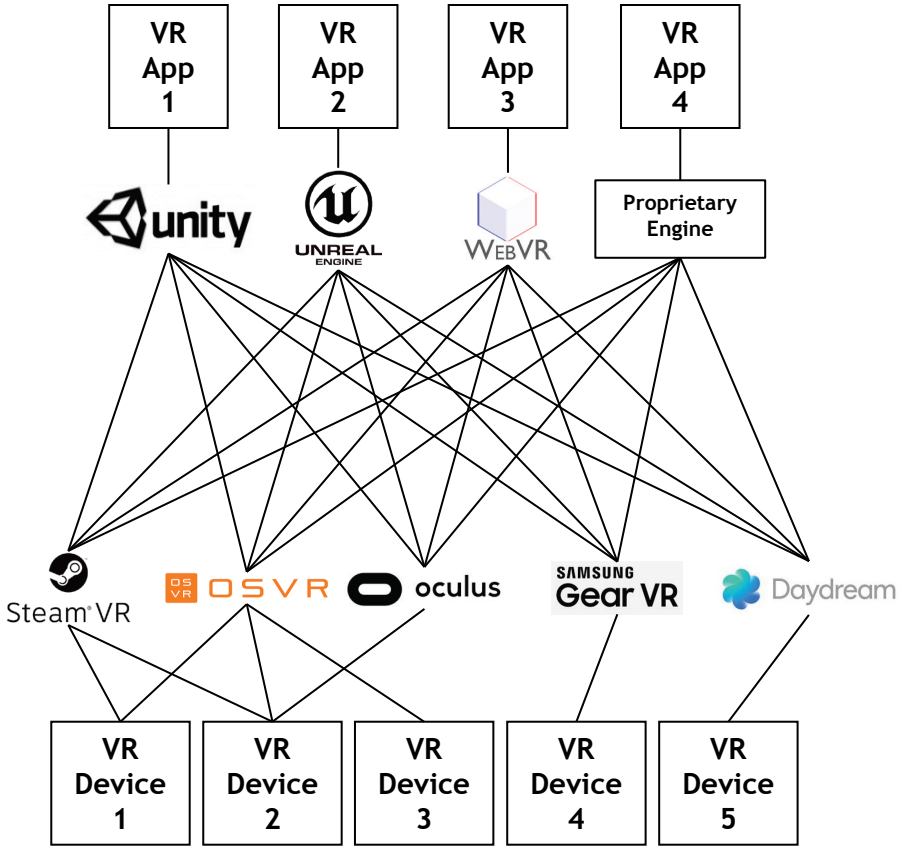
OpenXR

OpenXR!

Cross-Platform, portable Virtual Reality

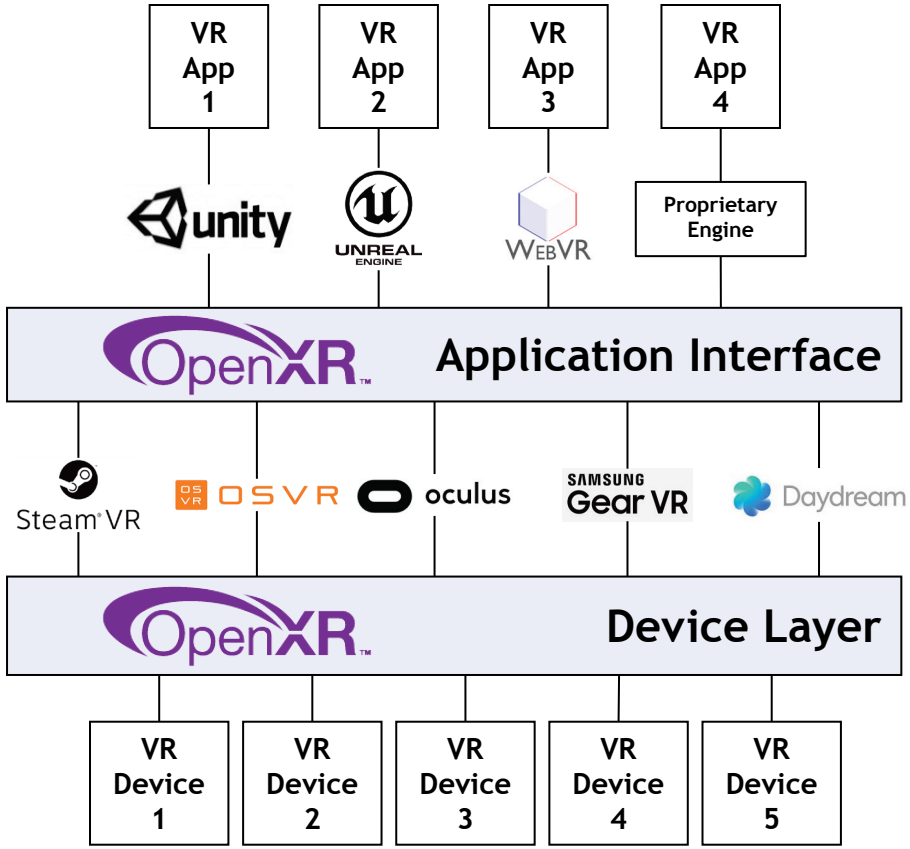
KHRONOS GROUP

OpenXR - Solving VR Fragmentation



Before OpenXR

VR Market Fragmentation



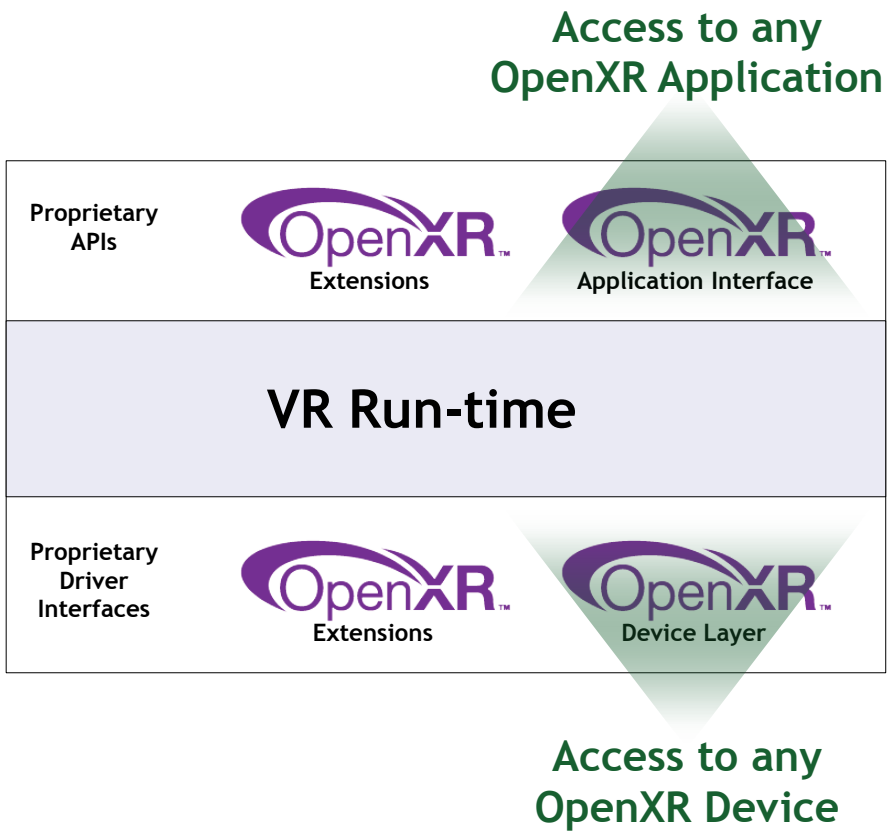
After OpenXR

Wide interoperabilityof VR apps and devices

OpenXR Working Group Members

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

OpenXR and VR Run-times



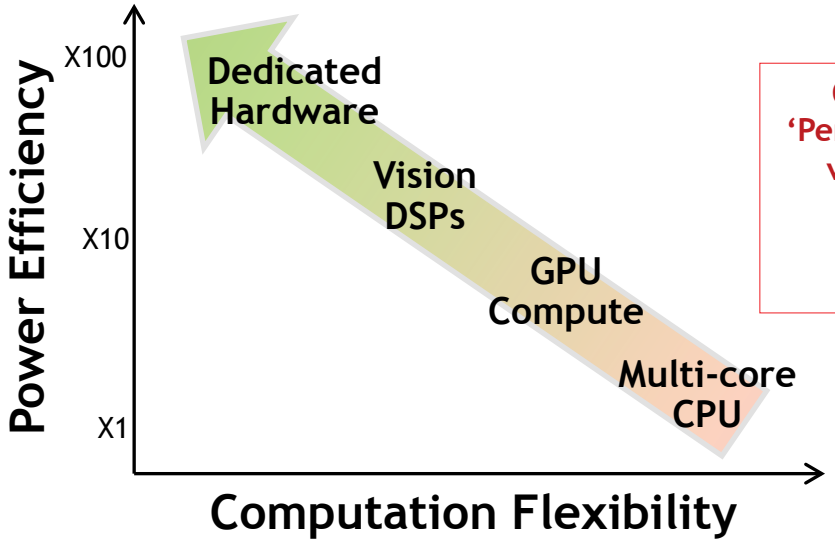
Any successful standard encourages and enables healthy industry competition

OpenXR will not replace VR run-times - or outlaw existing interfaces

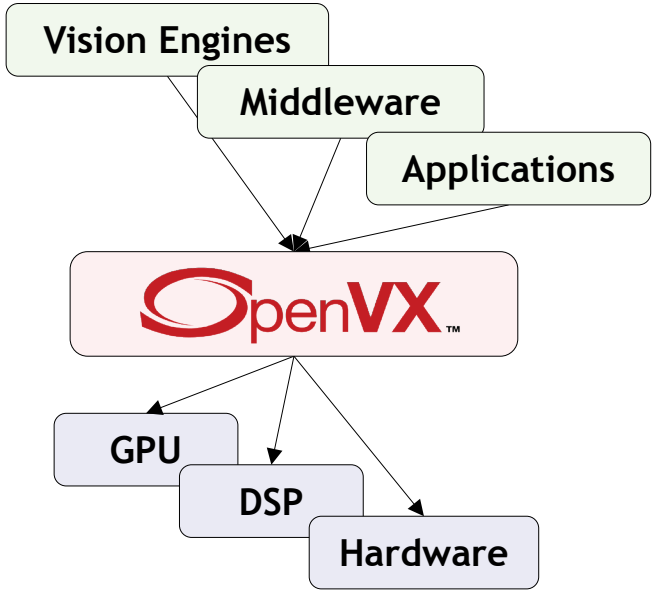
OpenXR will simply provide cross-vendor APIs that can be exposed by a runtime to access more apps and devices

OpenVX - Efficient Vision Acceleration

- **Vision acceleration for real-time, mobile and embedded platforms**
  - High performance AND low power consumption are key
- **Higher abstraction than OpenCL for performance portability across diverse architectures**
  - Multi-core CPUs, GPUs, DSPs and DSP arrays, ISPs, FPGAs, Dedicated hardware…
- **Extends portable vision acceleration to very low power domains**
  - Doesn't require high-power CPU/GPU Complex or CUDA/OpenCL precision

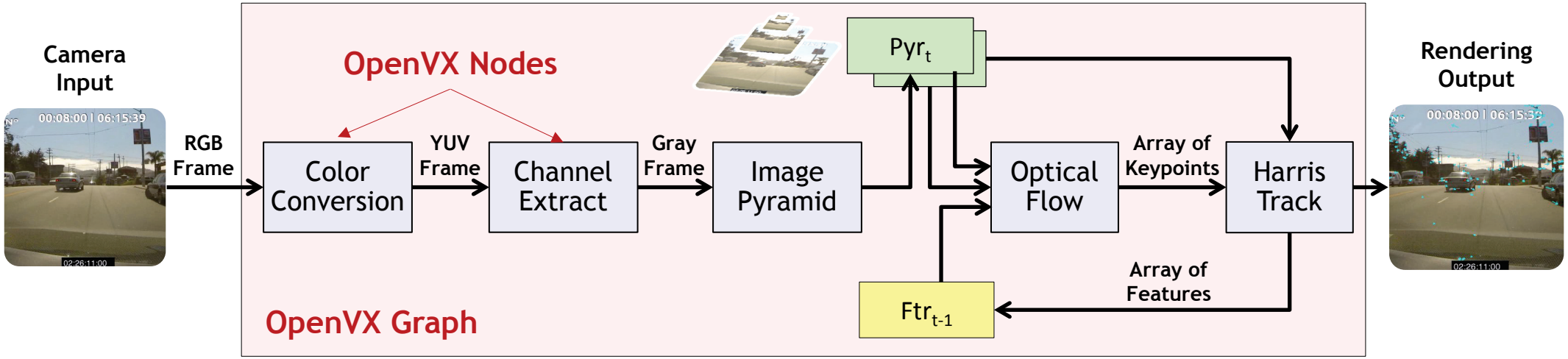


OpenVX provides 'Performance Portable' vision processing. Write once, run (efficiently) everywhere



OpenVX - Graph-Level Abstraction

- **OpenVX developers express a graph of image operations ('Nodes')**
  - Using a C API
- **Nodes can be executed on any hardware or processor coded in any language**
  - Implementers can optimize under the high-level graph abstraction
- **Graphs are the key to run-time power and performance optimizations**



Feature Extraction Example Graph

Simple Edge Detector in OpenVX

```
vx_image input = vxCreateImage(1920, 1080);
vx_image output = vxCreateImage(0, 0);
vx_image horiz = vxCreateVirtualImage();
vx_image vert = vxCreateVirtualImage();
vx_image mag = vxCreateVirtualImage();

vx_graph g = vxCreateGraph();
vxSobel3x3Node(g, input, horiz, vert);
vxMagnitudeNode(g, horiz, vert, mag);
vxThresholdNode(g, mag, THRESH, output);

status = vxVerifyGraph(g);
status = vxProcessGraph(g);
```

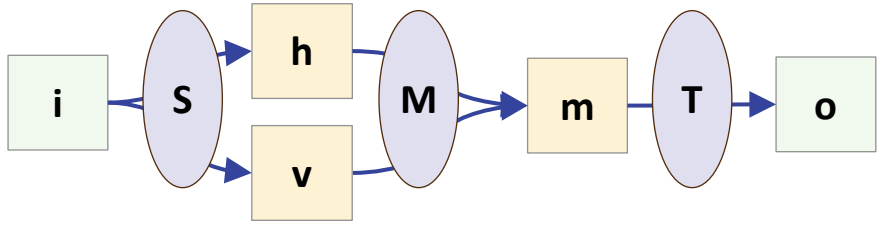
Declare Input and Output Images

Declare Intermediate Images

Construct the Graph topology

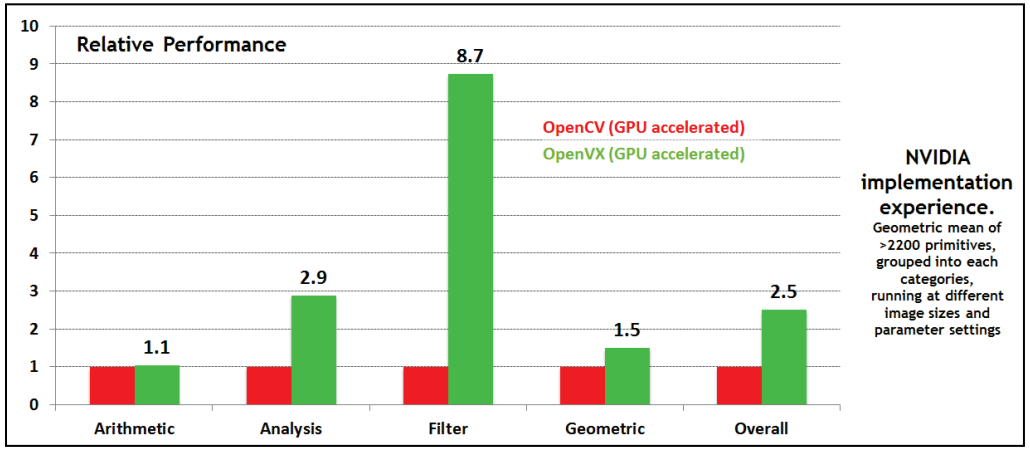
Compile the Graph

Execute the Graph



OpenVX - Efficiency AND Rapid Development

- **Graphs enable automatic optimizations**
  - Especially tiling and kernel fusion
- **Automatic selection of kernels**
  - Can be optimized by data type at graph verification time
- **User does NOT need to handle:**
  - Hardware features: like scatter-gather, custom ISPs, hardware blocks
  - Custom kernels: intrinsics, assembly code
  - Custom data movement: tiling, local memory management, DMA, memory hierarchy
- **The OpenVX framework does all of this for you**



OpenVX Evolution

Conformant Implementations

AMD
cadence
CEVA
intel
Imagination
NVIDIA
socionext
SYNOPSYS
VeriSilicon

OpenVX 1.0
Spec released October 2014

Conformant Implementations

cadence
Imagination
Texas Instruments

New Functionality
Expanded Nodes Functionality
Enhanced Graph Framework

AMD OpenVX Tools
- Open source, highly optimized for x86 CPU and OpenCL for GPU
- "Graph Optimizer" looks at entire processing pipeline and removes, replaces, merges functions to improve performance and bandwidth
- Scripting for rapid prototyping, without re-compiling, at production performance levels

http://gpuopen.com/compute-product/amd-openvx/

OpenVX 1.1
Spec released May 2016

New Functionality
Conditional node execution
Feature detection
Classification operators
Expanded imaging operations

Extensions
Neural Network Acceleration
Graph Save and Restore
16-bit image operation

Safety Critical
OpenVX 1.1 SC for safety-certifiable systems

OpenVX 1.2
Spec released May 2017

New Functionality Under Discussion

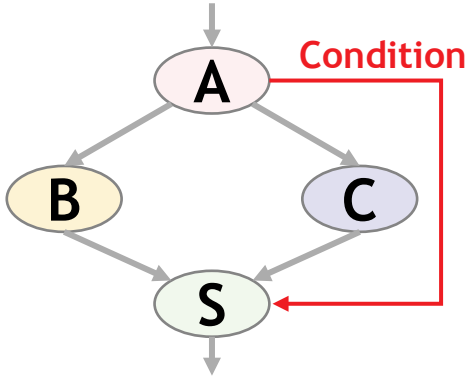
NNEF Import
(Neural Net Exchange Format)

Programmable user kernels with accelerator offload

OpenVX
Roadmap

New OpenVX 1.2 Functions

- **Feature detection: find features useful for object detection and recognition**
  - Histogram of gradients – HOG
  - Local binary patterns – LBP
  - Template matching
  - Line finding
- **Classification: detect and recognize objects in an image based on a set of features**
  - Import a classifier model trained offline
  - Classify objects based on a set of input features
- **Image Processing: transform an image**
  - Generalized nonlinear filter: Dilate, erode, median with arbitrary kernel shapes
  - Non maximum suppression: Find local maximum values in an image
  - Edge-preserving noise reduction
- **Conditional execution & node predication**
  - Selectively execute portions of a graph based on a true/false predicate
- **Many, many minor improvements**
- **New Extensions**
  - **Import/export:** compile a graph; save and run later
  - **16-bit support:** signed 16-bit image data
  - **Neural networks:** Layers are represented as OpenVX nodes



If A then S ← B else S ← C

OpenCL 2.2 - Top to Bottom C++

Single Source C++ Programming
Full support for features in C++14-based Kernel Language

API and Language Specs
Brings C++14-based Kernel Language into core specification

Portable Kernel Intermediate Language
Support for C++14-based kernel language e.g. constructors/destructors

OpenCL C++ Kernel Language
SPIR-V 1.2 with C++ support
SYCL 2.2 for single source C++

Dec08 — OpenCL 1.0 Specification
3-component vectors
Additional image formats
Multiple hosts and devices
Buffer region operations
Enhanced event-driven execution
Additional OpenCL C built-ins
Improved OpenGL data/event interop

18 months

Jun10 — OpenCL 1.1 Specification
Device partitioning
Separate compilation and linking
Enhanced image support
Built-in kernels / custom devices
Enhanced DX and OpenGL Interop

18 months

Nov11 — OpenCL 1.2 Specification
Shared Virtual Memory
On-device dispatch
Generic Address Space
Enhanced Image Support
C11 Atomics
Pipes
Android ICD

24 months

Nov13 — OpenCL 2.0 Specification
SPIR-V in Core
Subgroups into core
Subgroup query operations
clCloneKernel
Low-latency device timer queries

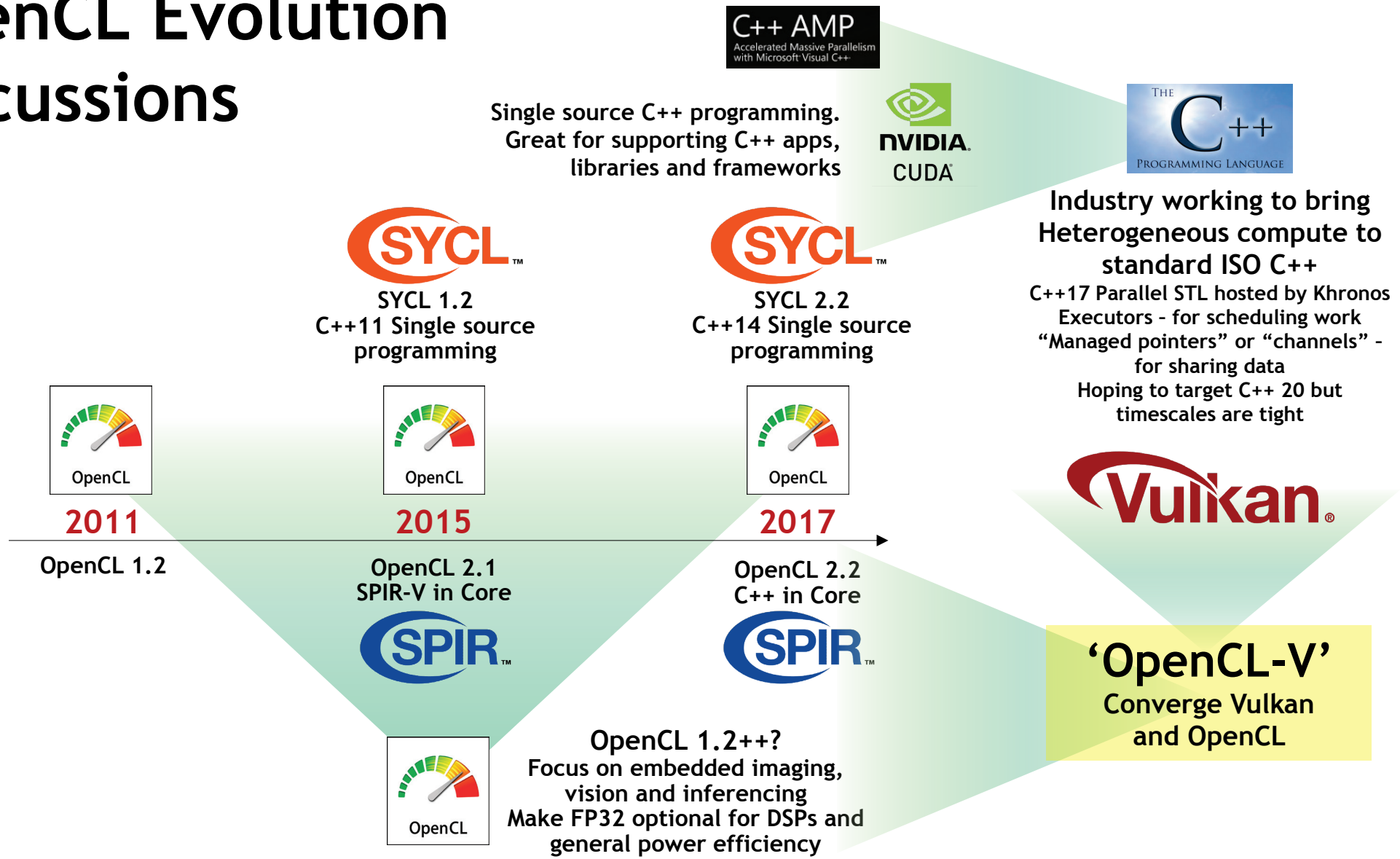
24 months

Nov15 — OpenCL 2.1 Specification

18 months

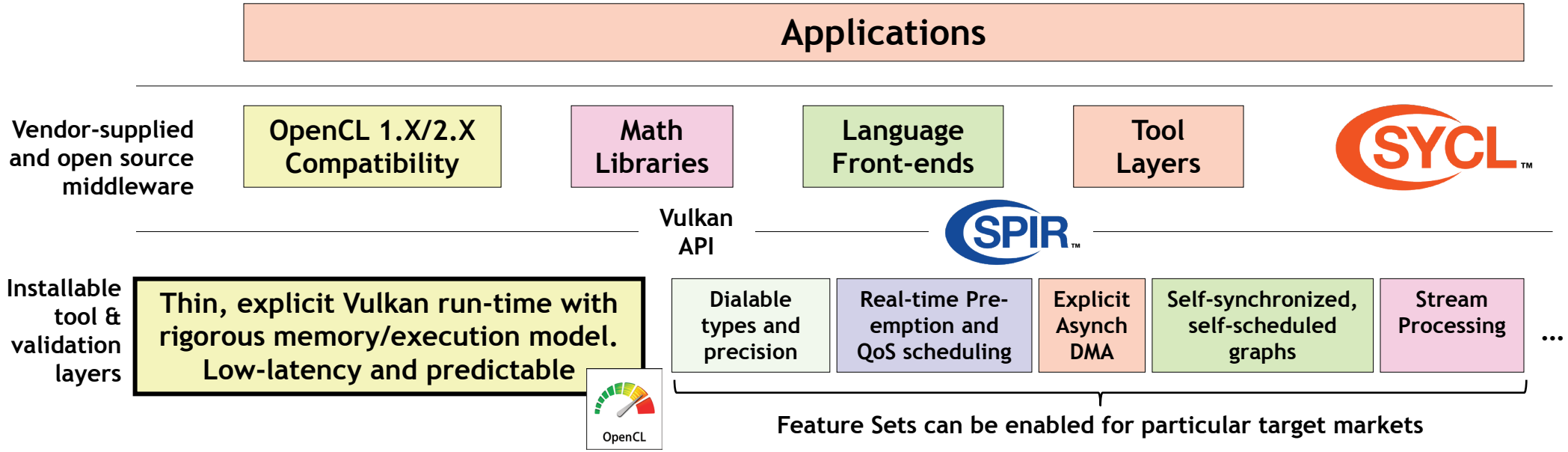
May17 — OpenCL 2.2 Specification

OpenCL Evolution Discussions



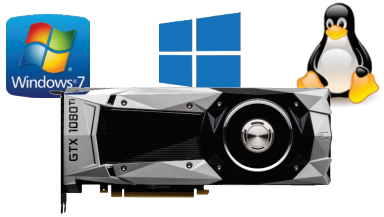
‘OpenCL-V’ - OpenCL and Vulkan Convergence

- **Converge over time with OpenCL roadmap with Vulkan API and run-time**
  - Expand Vulkan to include OpenCL-class compute
  - Support more processor types, e.g. DSPs and FPGAs (graphics optional)
- **Layered ecosystem for backwards-compatibility and market flexibility**
  - Feature sets for target market agility
- **Single runtime stack for graphics *and* compute**
  - Streamline development, adoption and deployment for the entire industry



NVIDIA and Khronos Standards

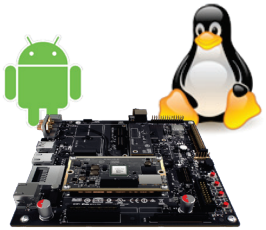
Vulkan



Desktop - Kepler and later

Nintendo Switch

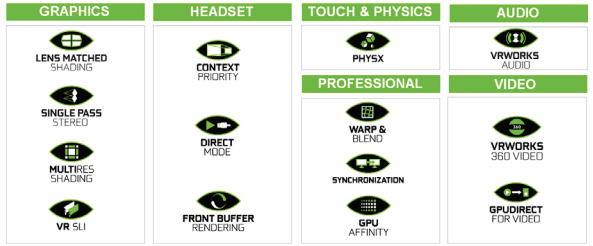
Shield Android TV



Embedded Tegra / Jetson

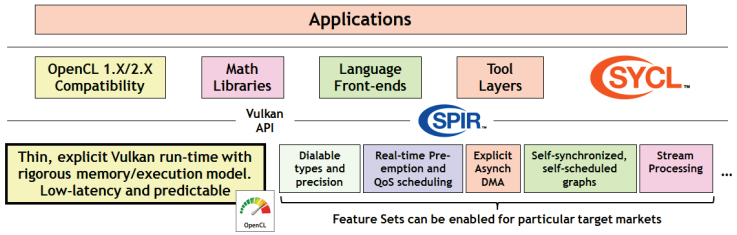
NVIDIA shipping latest specification, driving experimental extensions (e.g. for VR), adding Vulkan support to NSIGHT

OpenXR™



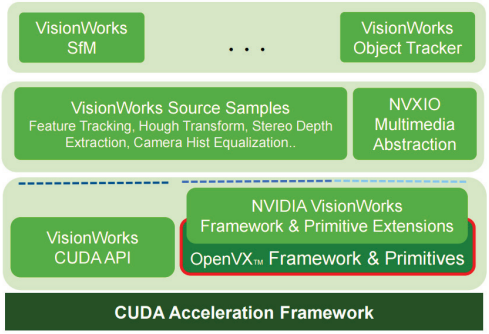
Actively helping to define and drive OpenXR to enable and complement VRWorks

OpenCL



Shipping extended OpenCL 1.2 on Linux and Windows desktop. Actively exploring roadmap convergence with Vulkan

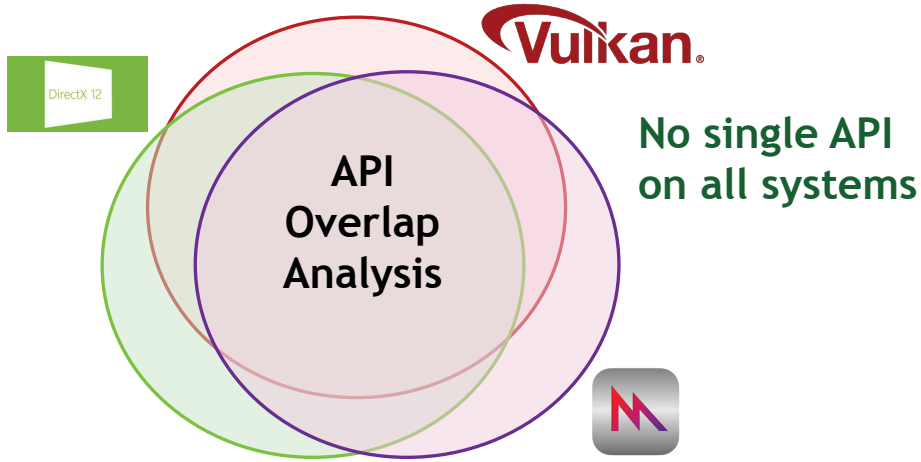
OpenVX™



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

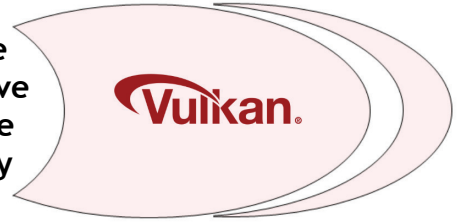
KHRONOS™ GROUP

GPU Portability - Call For Participation



No single API
on all systems

Use Feature
Sets to remove
non-portable
functionality



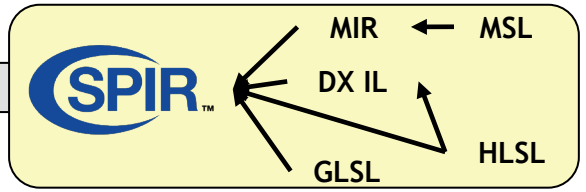
Use Extensions
to add
functionality
e.g. security and
robustness for
the Web

Portable 'Vulkan
Subset' API
Specification

'WebGL Next'

- Lift 'Portability API' to JavaScript *and* use in WebAssembly natve code
- Nexgen graphics and GPU compute for the Web

Vulkan Portability Solution
C/C++ Portability API Library
+ Shading Language tools
All open source



Open source compilers/translators for
shading and intermediate languages

Vulkan is non-proprietary and is
already designed to be portable

A Portability Solution needs to address
APIs *and* shading languages

WebGL, WebVR and glTF Meetup at GTC

Audio	Video	Images	3D
MP3	H.264	JPEG	glTF
napster	YouTube	facebook	New market opportunities for 3D content creation and deployment!

glTF - OpenGL Transmission Format
Efficient transmission of 3D assets

Wednesday 7-9PM Room 231
Fast-paced WebGL, WebVR and glTF presentations and demos. Beer!

Now glTF 2.0 has PBR!
Cool, portable materials
Rendering API independence
Coming to Microsoft Office!

glTF

- Compact to Transmit ✓
- Fast to Load ✓
- Describes Full Scenes ✓
- Runtime Neutral ✓
- Extensible ✓

Laugh Engine running on Vulkan
https://github.com/jian-ru/laugh_engine

Thank You!

Vulkan, OpenCL, OpenVX Advisory Panels
$0, come under Khronos NDA and IP Framework
Provide requirements and spec draft feedback

- **NVIDIA resources**
  - https://developer.nvidia.com/Vulkan
  - https://developer.nvidia.com/vrworks
  - https://developer.nvidia.com/embedded/visionworks
  - https://developer.nvidia.com/opencl
- **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**
  - www.khronos.org
- **Neil Trevett**
  - ntrevett@nvidia.com | @neilt3d