OpenCL and Ecosystem

State of the Nation

Neil Trevett | Khronos President
NVIDIA Vice President Developer Ecosystem
OpenCL Working Group Chair
ntrevett@nvidia.com | @neilt3d
Oxford, May 2018

State of the OpenCL Nation

OpenCL is needed by the industry and widely used
Unique framework for portable heterogeneous programming
Significant work in OpenCL 2.2 maintenance release - here at IWOCL!
Growing interest in SYCL, SPIR-V and related tools

BUT OpenCL Faces Deployment Friction
OpenCL 1.2 remains the widely adopted baseline – slow adoption of 2.X
Vital platforms such as Android do not have official OpenCL
Many embedded processors are locked out from OpenCL conformance

Focus on Increasing Deployment Flexibility
Enable OpenCL implementations on diverse processors and platforms
Streamline deployment of safety critical systems
Enable OpenCL applications to run on additional run-times

© Copyright Khronos™ Group 2018 - Page 2

OpenCL Evolution

OpenCL 2.2 Maintenance Release here at IWOCL

2011

OpenCL 1.2

Becomes industry baseline for heterogeneous parallel computing

2013

OpenCL 2.0

Enables new class of hardware
SVM
Generic Addresses
On-device dispatch

2015

OpenCL 2.1
SPIR-V 1.0

SPIR-V 1.1 in Core
Kernel Language
Flexibility

2017

OpenCL 2.2
SPIR-V 1.2

OpenCL C++ Kernel Language
Static subset of C++14
Templates and Lambdas

SPIR-V 1.2 in Core
OpenCL C++ support

Pipes
Efficient device-scope communication between kernels

https://www.khronos.org/opencl/

© Copyright Khronos™ Group 2018 - Page 3

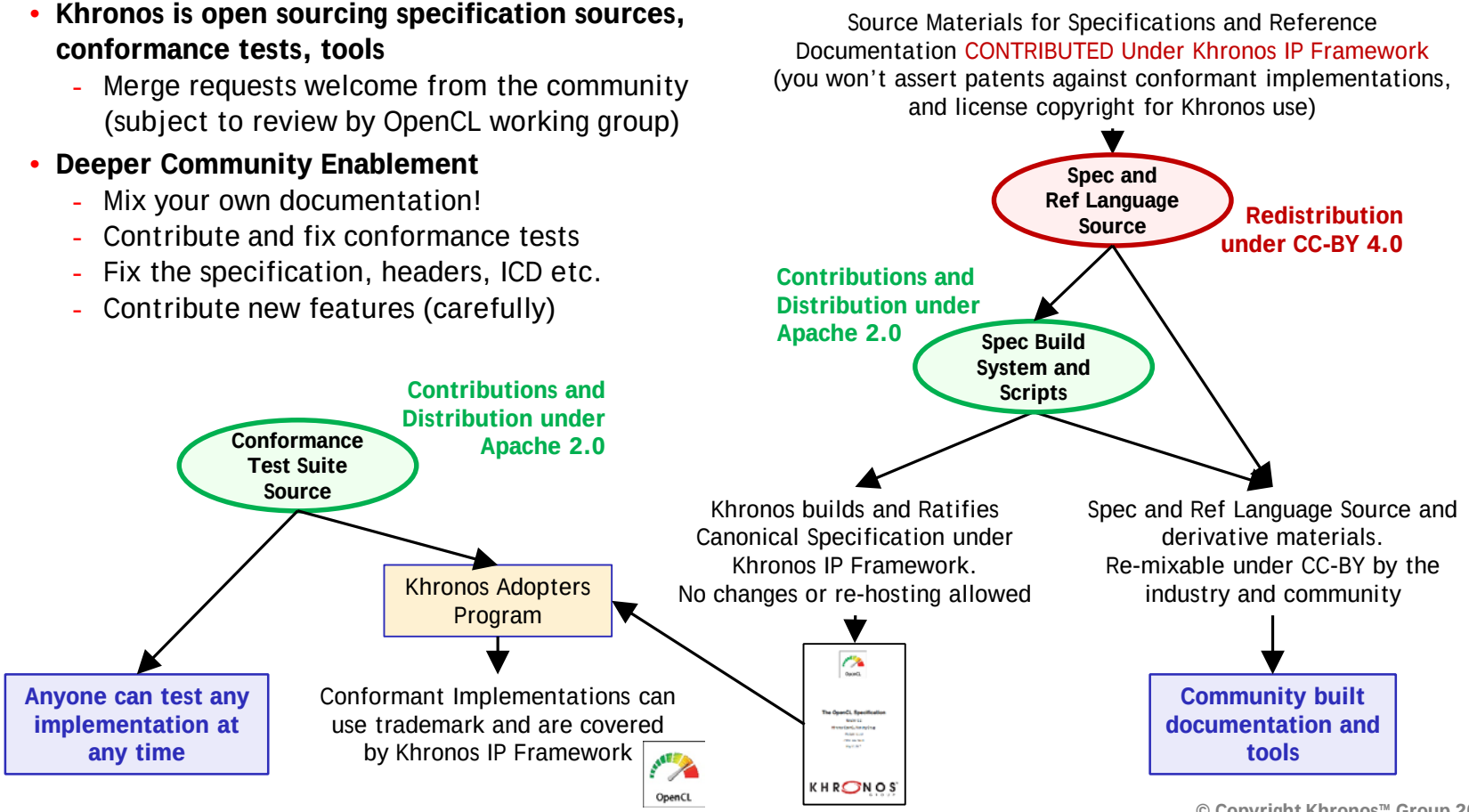
OpenCL 2.2 Maintenance Release

- **Fully backwards compatible**
  - 30+ bug fixes and clarifications
  - Including public GitHub issue fixes – thank you to those who logged bugs!
- **Updated and open-sourced the OpenCL C programming language spec**
  - Now possible to make pull requests for it – just like OpenCL API and C++ specs
  - Same look-and-feel as the other specs
- **Converted the spec toolchain from AsciiDoc to AsciiDoctor**
  - Same toolchain that is used by many other Khronos specs
  - Updated specs should be easier to read and to navigate
- **OpenCL SPIR-V environment specification has been improved**
  - Much easier for SPIR-V generators to know what is legal SPIR-V for OpenCL
- **Unified headers**
  - Use same headers to target any OpenCL version or to use any OpenCL extension

KHRONOS GROUP

New Open Source Engagement Model

- **Khronos is open sourcing specification sources, conformance tests, tools**
  - Merge requests welcome from the community (subject to review by OpenCL working group)
- **Deeper Community Enablement**
  - Mix your own documentation!
  - Contribute and fix conformance tests
  - Fix the specification, headers, ICD etc.
  - Contribute new features (carefully)

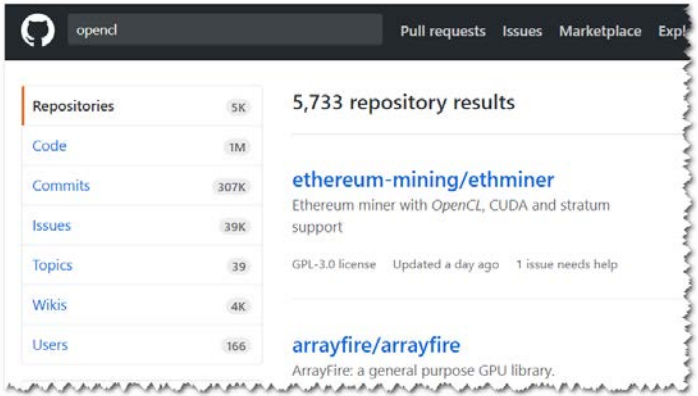


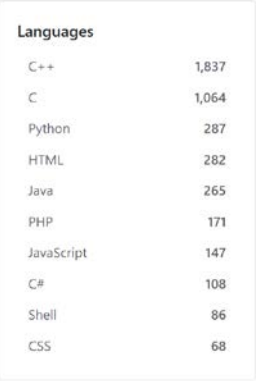
© Copyright Khronos™ Group 2018 - Page 5

Growing OpenCL Adoption

- **100s of applications using OpenCL acceleration**
  - Rendering, visualization, video editing, simulation, image processing
- **Almost 6,000 GitHub repositories using OpenCL**
  - Tools, applications, libraries, languages
  - Up from 4310 one year ago
- **Khronos Resource Hub**

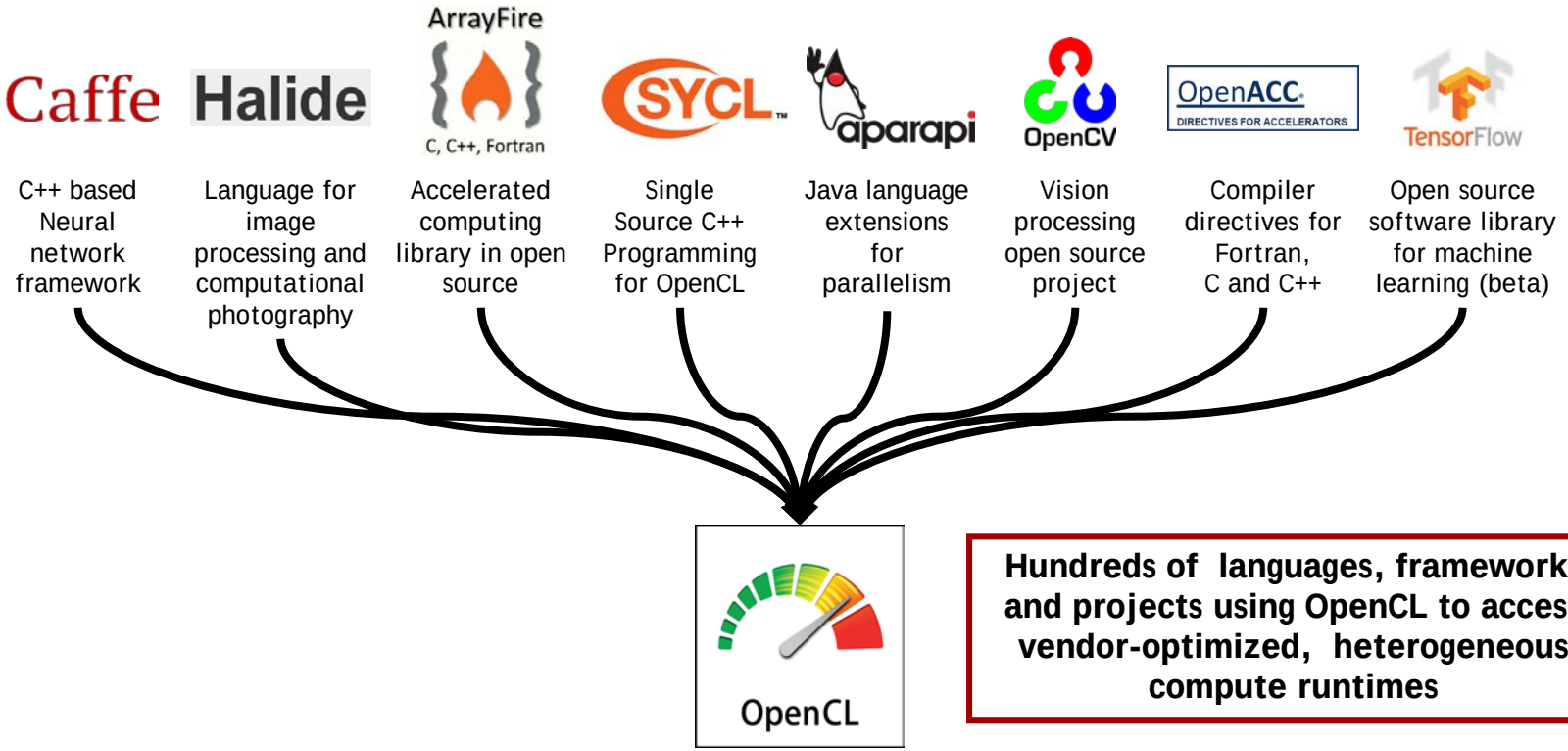
  https://www.khronos.org/opencl/resources/opencl-applications-using-opencl





© Copyright Khronos™ Group 2018 - Page 6

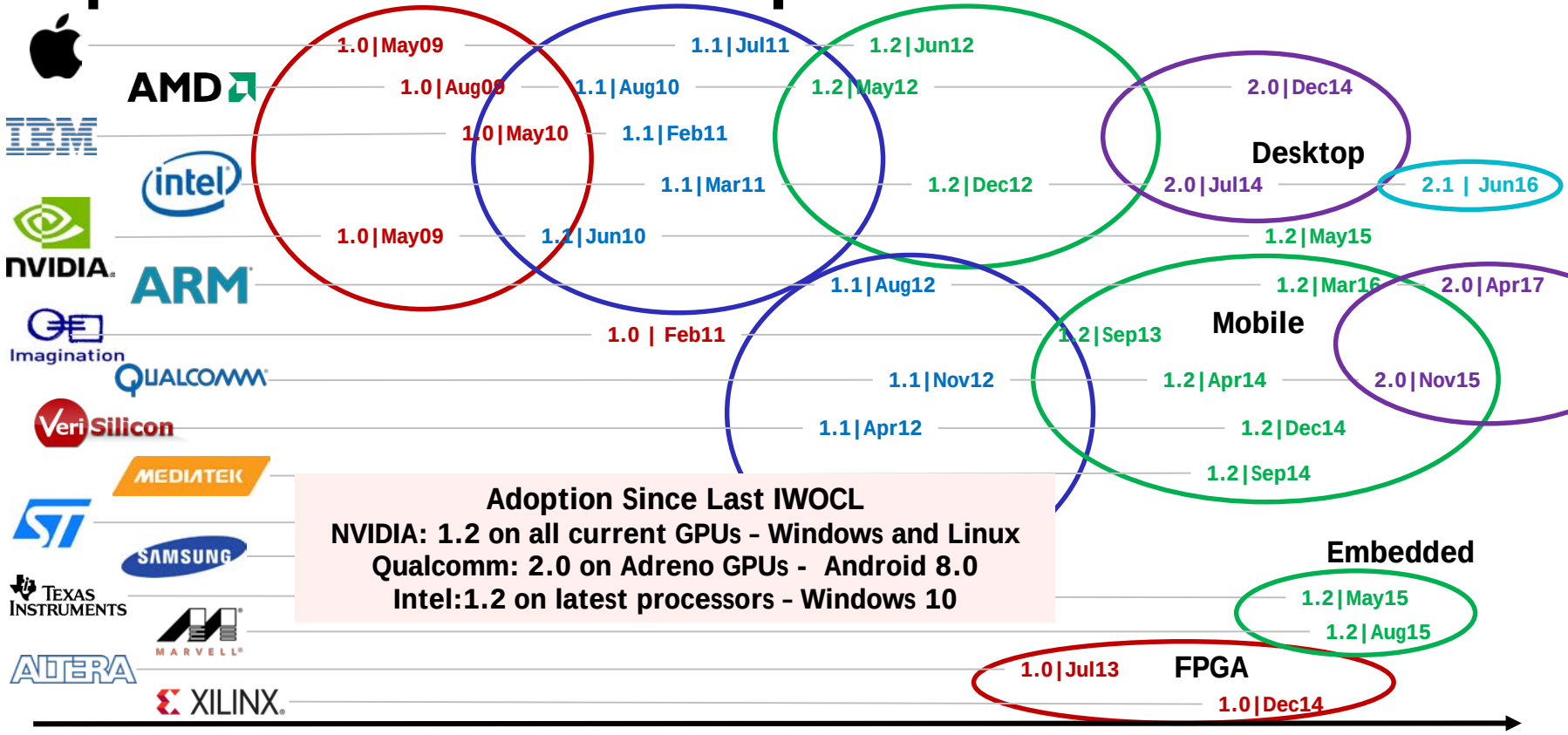
OpenCL as Language/Library Backend



Hundreds of languages, frameworks and projects using OpenCL to access vendor-optimized, heterogeneous compute runtimes

KHRONOS GROUP

OpenCL Conformant Implementations

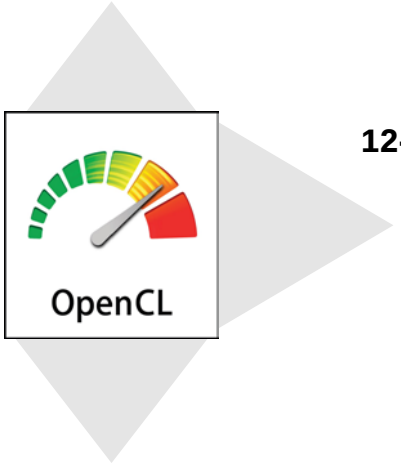


© Copyright Khronos™ Group 2018 - Page 8

Understanding OpenCL Adoption Patterns

OpenCL 1.2 remains the widely-supported industry baseline

SVM in 2.0 is problematic for non-unified memory – e.g. discrete GPUs
SVM in 2.0 is easier on mobile with shared memory
Some 2.0 features are less ‘controversial’ and shipping more widely



No OpenCL 2.2 Yet?

12-18 months between spec and first implementations are common
Don’t panic - OpenCL 2.1 implementations are not late yet
SPIR-V front-ends and tools maturing
C++ comes ‘for free’ with SPIR-V 1.2 ingestion
OR
Is C++ interesting to kernel developers?
Or is single source file, SYCL-style, where C++ interest is?

Only High-end DSPs

Smaller DSPs do not have 32-bit FP – mandated for conformance
Optimized vision and inferencing engines are ‘locked out’

© Copyright Khronos™ Group 2018 - Page 9

SPIR-V Transforms the Language Ecosystem

- **First multi-API, intermediate language for parallel compute and graphics**
  - Natively represents structures in shader and kernel languages
  - https://www.khronos.org/registry/spir-v/papers/WhitePaper.pdf
- **Compiler IR for OpenCL, Vulkan and OpenGL**
  - Easy to parse - just a stream of words
  - Easy to transform - designed to be easy to convert to and from LLVM IR
  - Easy to manipulate and optimize - Static Single Assignment form

Multiple Developer Advantages

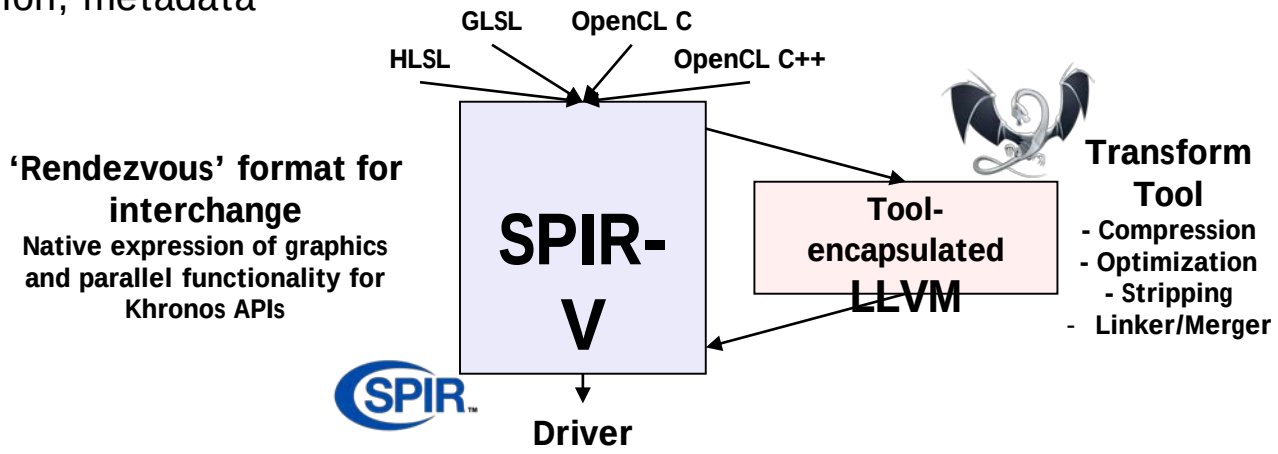
Use same front-end compiler for all platforms
Ship SPIR-V – not shader source code
Simpler and more reliable drivers
Reduces runtime kernel compilation time

Standard
Portable
Intermediate
Representation

© Copyright Khronos™ Group 2018

Support for Both SPIR-V and LLVM

- **LLVM is an SDK, not a formally defined standard**
  - Khronos moved away from trying to use LLVM IR as a standard
  - Issues with versioning, metadata, etc.
- **But LLVM is a treasure chest of useful transforms**
  - SPIR-V tools can use encapsulation and use LLVM to do useful SPIR-V transforms
- **SPIR-V tools can all use different rules – and there will be lots of these**
  - May be lossy and only support SPIR-V subsets
  - Internal form is not standardized
  - May hide LLVM version, metadata



© Copyright Khronos™ Group 2018

Evolution of SPIR Family

SPIR	SPIR 1.2	SPIR 2.0	SPIR-V 1.X
LLVM Interaction	Uses LLVM 3.2	Uses LLVM 3.4	100% Khronos defined Round-trip lossless conversion
Compute Constructs	Metadata/Intrinsics	Metadata/Intrinsics	Native
Graphics Constructs	No	No	Native
Supported Language Feature Sets	OpenCL C 1.2	OpenCL C 1.2 OpenCL C 2.0	OpenCL C 1.2 / 2.X OpenCL C++ GLSL HLSL
OpenCL Ingestion	OpenCL 1.2 Extension	OpenCL 2.0 Extension	OpenCL 2.1/2.2 Core
Graphics API Ingestion	-	-	Vulkan and OpenGL 4.6 Core

SPIR-V defines supported subsets for each 'host' API through 'environment specs'

© Copyright Khronos™ Group 2018 - Page 12

SPIR-V Ecosystem

Open source tools and translators

https://github.com/KhronosGroup/SPIRV-Tools

SPIR-V
- Khronos defined cross-API IR
- Native graphics and parallel compute
- Easily parsed/extended 32-bit stream
- Data object/control flow retained for effective code generation/translation

GLSL → glslang
HLSL → glslang, DXC

Third party kernel and shader languages → SPIR-V

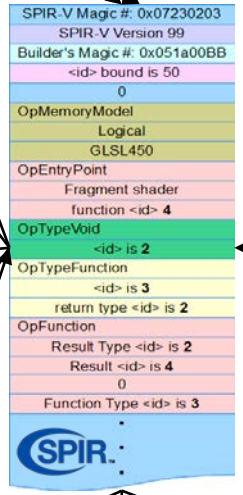
MSL, HLSL, GLSL → SPIRV-Cross → SPIR-V

SPIR-V (Dis)Assembler ↔ SPIR-V

SPIR-V Validator ↔ SPIR-V

SPIRV-opt | SPIRV-remap ↔ SPIR-V

Additional Intermediate Forms ↔ SPIR-V



SPIR-V → OpenCL, Vulkan, OpenGL

OpenCL C Front-end, OpenCL C++ Front-end, SYCL Front-end → LLVM

LLVM ↔ SPIR-V: LLVM to SPIR-V Bi-directional Translator

Khronos liaising with Clang/LLVM Community
E.g. discussing SPIR-V as supported Clang target

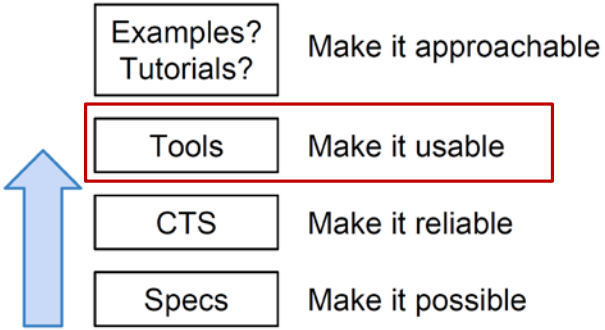
SPIR-V Optimizations
- Inlining (exhaustive)
- Store/Load Elimination
- Dead Code Elimination
- Dead Branch Elimination
- Common Uniform Elimination
- Loop Unrolling and Constant Folding
- Common Subexpression Elimination

SPIR-V 1.3 released with Vulkan 1.1 inc. subgroups

© Copyright Khronos™ Group 2018

OpenCL Tooling Ecosystem Subgroup

- **Coordinating SPIR-V and LLVM ecosystems**
  - Encouraging joint development of new features and tool integration
- **New common SPIRV<->LLVM translator repo w/o using LLVM tree**
  - Extending SPIRV<->LLVM translation, including for Vulkan over time
  - https://github.com/KhronosGroup/SPIRV-LLVM-Translator
- **Support SPIR-V as Clang Backend**
  - Upstream SPIR-V translation to Clang/LLVM & adding target triple
  - Define set of use cases for OpenCL in Clang (build, link, create libs)
  - Leverage and re-use SPIR-V linker/opt/validator Tools
- **Improving documentation**
  - SPIR-V friendly format of LLVM IR



© Copyright Khronos™ Group 2018 - Page 14

OpenCL Ecosystem Roadmap

C++ AMP
Accelerated Massive Parallelism with Microsoft Visual C++

Single source C++ programming.
Great for supporting C++ apps, libraries and frameworks

NVIDIA CUDA

THE C++ PROGRAMMING LANGUAGE

Work with industry to bring Heterogeneous compute to standard ISO C++

SYCL™

SYCL 1.2
C++11 Single source programming

SYCL™

SYCL 1.2.1
C++11 Single source programming

OpenCL

OpenCL 'Next'
Flexible and efficient deployment of parallel computation across diverse processor architectures

OpenCL

2011

OpenCL 1.2
OpenCL C Kernel Language

OpenCL

2015

OpenCL 2.1
SPIR-V in Core

SPIR™

OpenCL

2017

OpenCL 2.2
C++ Kernel Language

SPIR™

Vulkan®

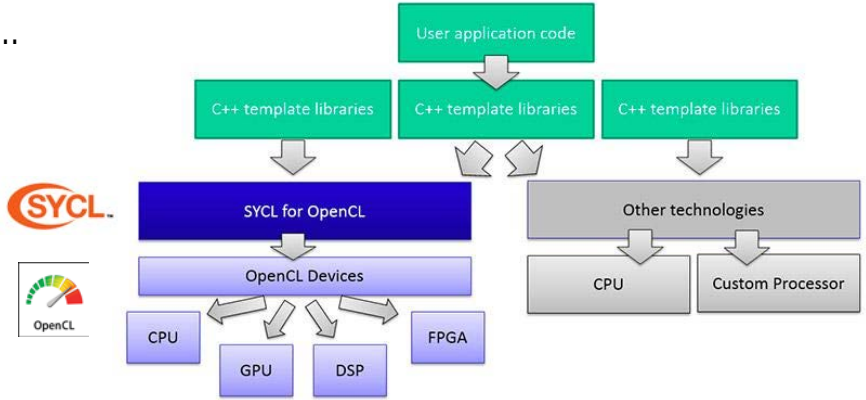
More deployment options:
Enabling dispatch of OpenCL C kernels from Vulkan runtimes

KHRONOS™ GROUP

© Copyright Khronos™ Group 2018

SYCL Ecosystem

- **Single-source heterogeneous programming using STANDARD C++**
  - Use C++ templates and lambda functions for host & device code
  - Layered over OpenCL
- **Fast and powerful path for bring C++ apps and libraries to OpenCL**
  - C++ Kernel Fusion - better performance on complex software than hand-coding
  - SYCLBLAS, SYCL Eigen, SYCL TensorFlow, SYCL DNN, SYCL GTX, VisionCpp,
  - C++17 Parallel STL hosted by Khronos
  - C++20 Parallel STL with Ranges
- **Implementations**
  - triSYCL, ComputeCpp, ComputeCpp SDK …
- **More information at http://sycl.tech**



SYCL Roadmap

- **SYCL 1.2.1 Ratified**
  - CTS and Adopters package in progress
- **SYCL 2.2 Provisional Released**
  - Launched in parallel with OpenCL 2.2 to enables device capabilities from a single source file
  - Shared virtual memory, generic pointers and device-side enqueue etc.
  - Vehicle to align with C++20 and beyond
- **Roadmap**
  - Tighter ISO C++ alignment in parallel - injecting our heterogeneous knowledge into ISO and adapting C++ features
  - More regular releases ~aiming at 1.5 years per release
  - Naming convention adapted to SYCLxxxx where xxxx=year of ratification
- **Focus on Machine learning and Vision processing**
  - For self-driving cars, SYCL TensorFlow, SYCL DNN
- **SYCL Safety Critical**
  - Demanded by Embedded Market customers
  - Especially Advanced Driver Assist Systems (ADAS)

Developer Choice

The development of the two specifications are aligned so code can be easily shared between the two approaches

C++ Kernel Language
Low Level Control
'GPGPU'-style separation of device-side kernel source code and host code

OpenCL

Single-source C++
Programmer Familiarity
Approach also taken by C++ AMP and OpenMP

SYCL™

KHRONOS™ GROUP

Vulkan and New Generation GPU APIs

Non-proprietary, royalty-free open standard 'By the industry for the industry'
Portable across multiple platforms - desktop and mobile
Modern architecture | Low overhead| Multi-thread friendly
EXPLICIT GPU access for EFFICIENT, LOW-LATENCY, PREDICTABLE performance

Vulkan is available on Android 7.0+

KHRONOS GROUP

Pervasive Vulkan 1.0

Vulkan

Major GPU Companies supporting Vulkan for Desktop and Mobile Platforms

AMD arm BROADCOM Imagination intel NVIDIA QUALCOMM VeriSilicon

http://vulkan.gpuinfo.org/

Platforms

Desktop

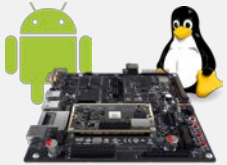
Mobile
(Android 7.0+)

Media Players

Consoles

Virtual Reality

Cloud Services



Embedded

Game Engines

VALVE id CRYENGINE unity EPIC GAMES Croteam Serious Engine XENKO

KHRONOS GROUP

Vulkan 1.1 Launch and Ongoing Momentum

Strengthening the Ecosystem

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

February 2016
Vulkan 1.0

Explicit Access to
GPU Acceleration

Vulkan 1.0 Extensions

Maintenance updates plus additional functionality

Explicit Building Blocks for VR
Explicit Building Blocks for Homogeneous Multi-GPU
Enhanced Windows System Integration
Increased Shading Language Flexibility
Enhanced Cross-Process and Cross-API Sharing

March 2018
Vulkan 1.1

Integration of Proven and
New Technology into Core

Building Vulkan's Future

Deliver complete ecosystem – not just specs
Listen and prioritize developer needs
Drive GPU technology

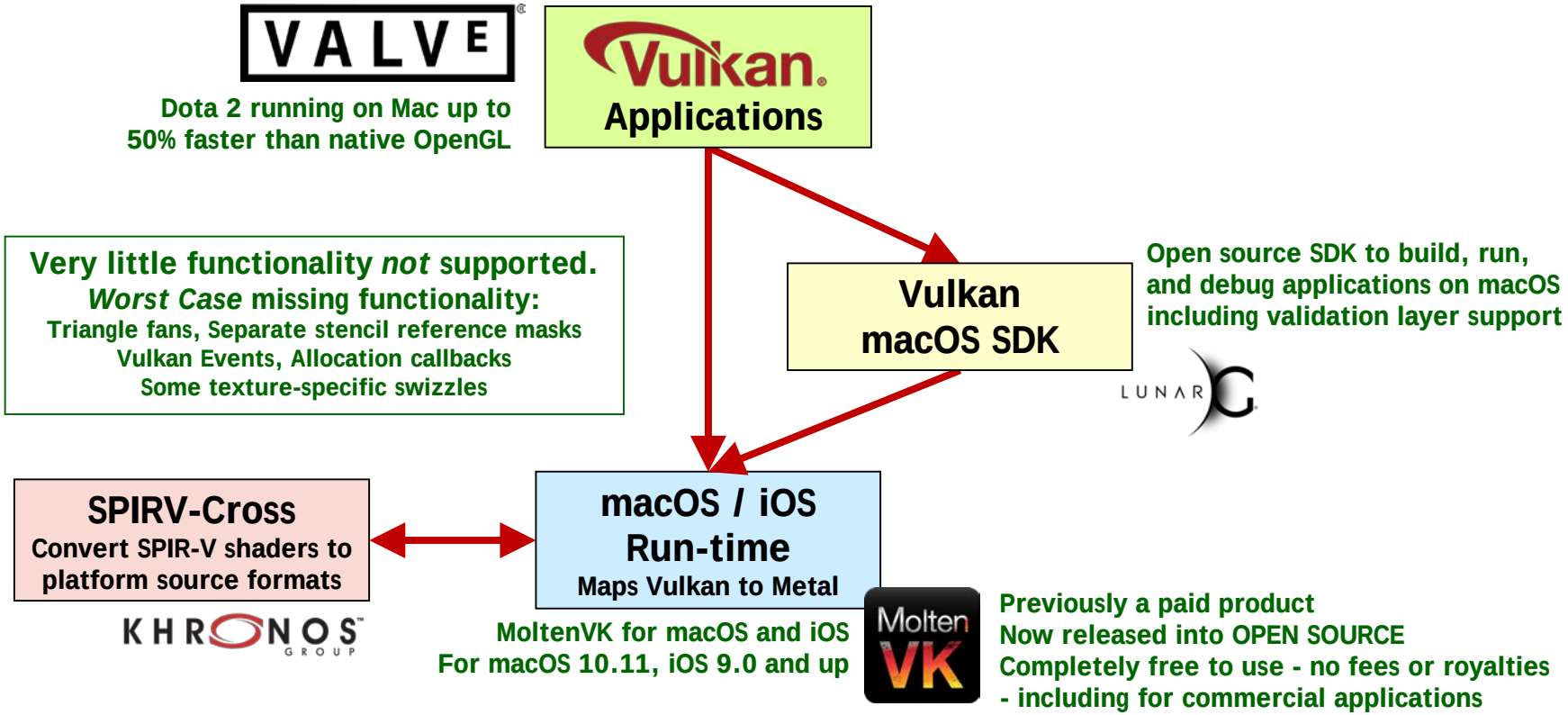
Widening Platform Support

Pervasive GPU vendor driver availability
Port Vulkan apps to macOS/iOS and DX12
Open source drivers

Vulkan 1.1 specification launched March 7th with open source conformance tests and tools, and multiple vendor implementations!

© Copyright Khronos™ Group 2018 - Page 20

Bringing Vulkan 1.0 Apps to Apple Platforms



© Copyright Khronos™ Group 2018 - Page 21

Vulkan Portability Initiative

Widened Platform Support

Open source run-times over additional backends
e.g. consoles and Windows UWP
E.g. Mozilla helping to drive gfx-rs for Vulkan over DX12, OpenGL and Metal
https://github.com/gfx-rs/gfx
https://github.com/gfx-rs/portability

TODAY

Beta release to bring Vulkan 1.0 applications to macOS and iOS running over Metal

Layers

Simulation and validation

Free open source layers and SDKs for apps to be ported to Vulkan-subset layered libraries

JSON Schema defines missing functionality per implementation

Conformance Testing

Subsets cannot be conformant but functionality that is present must work! Aiming to run all applicable tests by GDC 2019

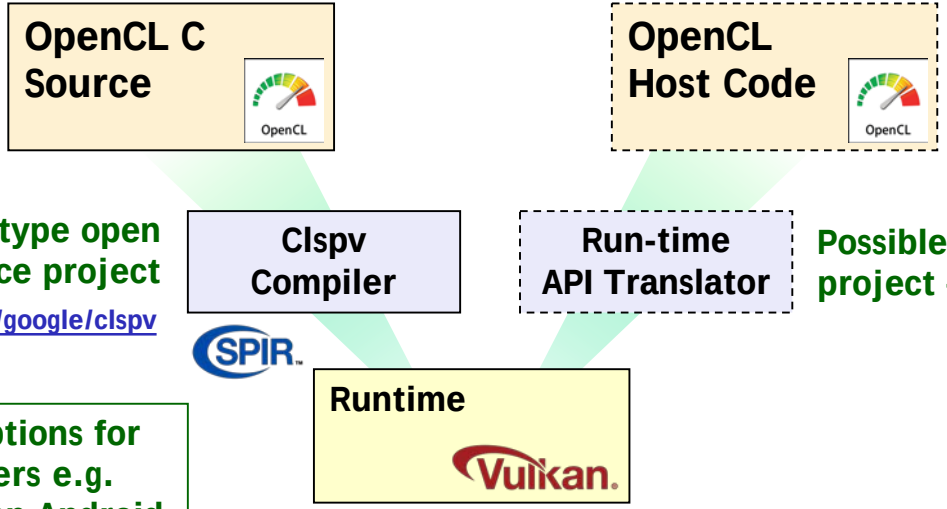
Portability Extension

Query target capabilities

© Copyright Khronos™ Group 2018 - Page 22

Clspv OpenCL C to Vulkan Compiler

- **Experimental collaboration between Google, Codeplay, and Adobe**
  - Successfully tested on over 200K lines of Adobe OpenCL C production code
  - Open source - tracks top-of-tree LLVM and clang, not a fork
- **Compiles OpenCL C to Vulkan's SPIR-V execution environment**
  - Proof-of-concept that OpenCL kernels can be brought seamlessly to Vulkan
  - Significant parts OpenCL C 1.2 so far – shaped by submitted workloads



Prototype open source project

https://github.com/google/clspv

Possible future project – if interest?

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

© Copyright Khronos™ Group 2018

Clspv Project Next Steps

- **The Clspv Process**
  - Try porting apps from OpenCL-native domains to Vulkan
  - Use Clspv to port OpenCL kernels to Vulkan compute shaders
  - Where compiler can't cover the difference, propose or support updates to the underlying Vulkan programming model e.g. 16-bit storage, Variable Pointers, Subgroups
- **Clspv is being shaped and exercised by the workloads attempted**
  - Try yours kernels!
- **Do we need OpenCL to Vulkan API shim?**
  - Khronos can host an open source project
- **Possible domains to explore:**
  - Existing OpenCL applications and libraries
  - Vision processing pipelines
  - Power-efficient machine learning and inferencing
  - Even gaming can benefit from better compute
    - e.g. HPG/SIGGRAPH 2016/17 talks
    - Andrew Lauritzen's talk @ Open Problems in Real-Time Rendering, SIGGRAPH'17

Compact memory types and operations

KHRONOS GROUP

Embedded Processors & OpenCL Conformance

- **The embedded market is a new frontier needing advanced compute**
  - E.g. Vision and inferencing using a wide range of processor architectures
- **BUT OpenCL is currently monolithic – and arguably desktop/HPC-centric**
  - E.g. a processor without 32-bit IEEE floating point cannot realistically be conformant
  - Vendors and developers do not want software emulation of higher precisions
- **Many functionality requirements change between different markets and processors**

OpenCL is disenfranchising one of its most important emerging market opportunities

Supported Precisions	DSP A	DSP B	DSP C
8-bit int	✓	✓	✓
16-bit int	✓	✓	✓
32-bit int	✓	✓	✓
64-bit int	✗	✓	✗
16-bit float	✗	✓	✓
32-bit float	✗	✗	✓
64-bit float	✗	✗	✗
Possible to be OpenCL Compliant?	No	No	Yes

© Copyright Khronos™ Group 2018

OpenCL Next Goals and Philosophy

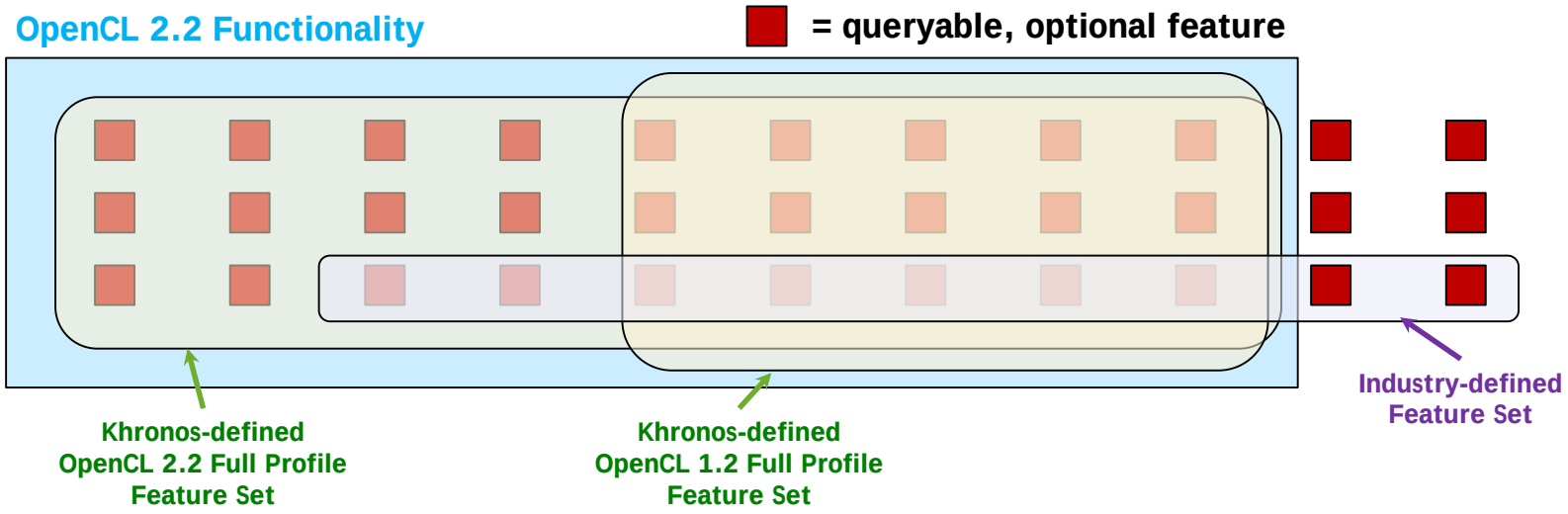
- **Enable *Conformant* OpenCL implementations on diverse processors and platforms**
  - Enable vendors to ship functionality targeted for their customers/markets
- **More implementation flexibility - more OpenCL features become optional**
  - Features can become optional in both API and languages
  - E.g. floating point precisions
- **Enable incremental feature adoption**
  - A conformant OpenCL can expose *precisely* what is available in the hardware
- **Enhanced query mechanisms**
  - So that application can query precisely which features are supported by a device
  - In addition to existing profiles, no changes for existing applications

Enable OpenCL to be a flexible run-time framework that can be cost-effectively deployed across a wide range of heterogeneous devices

KHRONOS GROUP

OpenCL Next Feature Sets

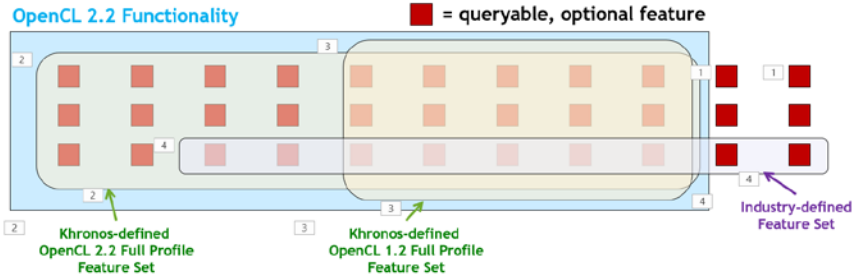
- **Vendor can support ANY combination of features to suit their hardware/market**
  - If all exposed features are conformant – the implementation is conformant
- **Existing profiles not going away! Khronos defined feature set alternatives**
  - No reason for vendors to remove functionality - as would break applications
- **Opportunity to coalesce industry support around market-focused feature sets**
  - Khronos aiming to provide the infrastructure for the industry



© Copyright Khronos™ Group 2018

OpenCL Next Feature Set *Discussion*

- **We need your input!**
  - Brainstorm discussions below!
- **Industry-defined sets to reduce market fragmentation**
  - Who should define these – how reach consensus? Not Khronos?
  - Vertical market focused – e.g. inferencing, vision processing?
  - Opportunity to move past the current 1.2 logjam - OpenCL 1.2++ Desktop Feature Set?
- **Feature Set Conformance - providing an incentive to reduce fragmentation**
  - If 100% of features pass all tests – vendor can claim conformance to that Feature Set
  - Supporting popular Feature Sets may help drive sales
  - An implementation may support multiple Feature Sets



© Copyright Khronos™ Group 2018

Safety Critical APIs – Khronos Experience

OpenGL SC

OpenGL SC 1.0 - 2005
Fixed function graphics subset

OpenGL ES

OpenGL ES 1.0 - 2003
Fixed function graphics

OpenGL SC

OpenGL SC 2.0 - April 2016
Shader programmable pipeline subset

OpenGL ES

OpenGL ES 2.0 - 2007
Shader programmable pipeline

Need for new-generation APIs for safety certifiable vision, graphics and compute
e.g. ISO 26262 and DO-178B/C

Lack of industry understanding and consensus on how APIs should be designed to streamline safety certification

OpenVX

OpenVX SC 1.1 - May 2017
Restricted "deployment" implementation only executes pre-compiled binary format

Generate guidelines for designing safety critical APIs to ease system certification.
Open to Khronos member AND industry experts

KHRONOS SAFETY CRITICAL ADVISORY FORUM SC™

© Copyright Khronos™ Group 2018 - Page 29

Khronos Safety Critical Advisory Forum

Industry outreach and cooperation

AESIN
Automotive ADAS & AV + security
https://aesin.org.uk

MISRA C++
C++ WG23 Programming Vulnerabilities
ISO C Safe and Secure SG
ISO C++ Vulnerabilities Safety Critical SG

KHRONOS SAFETY CRITICAL ADVISORY FORUM SC™

Generate guidelines for designing safety critical APIs to ease system certification.
Open to Khronos member AND industry experts
https://www.khronos.org/advisors/kscaf

We are inviting safety critical experts to join KSCAF!
No cost or work commitment

Khronos SC Activities

OpenCL

OpenCL SC TSG
Working on OpenCL SC
Gathering requirements

SYCL™

SYCL SC
Guidelines to augment Industry
First Safe and Secure Parallel and Heterogeneous C++
Safe AI for Automotive

OpenVX™

OpenVX SC 1.1 - May 2017
Restricted "deployment" implementation only executes pre-compiled binary format

KHRONOS GROUP

Khronos Advisory Panels

The Working Group invites input and shares draft specifications and other materials

Members

Companies pay membership Fee
Sign NDA and IP Framework + Membership
Directly participate in working groups

Advisors

Individuals Pay $0
Sign NDA and IP Framework
Provide requirements and feedback on spec drafts

Advisory Panel membership is 'By Invitation' and renewed annually.
No 'minimum workload' commitment – but we love input and feedback!
Please reach out if you wish to participate!

© Copyright Khronos™ Group 2018

Get Involved!

- **OpenCL is driving to new levels of deployment flexibility**
  - We need to know what you need from OpenCL
  - IWOCL is the perfect opportunity to find out!
- **In particular we need input and direction on OpenCL Next and Feature Sets**
  - Let us know what you think!
- **Any company or organization is welcome to join Khronos**
  - For a voice and a vote in any of these standards www.khronos.org
  - Or ask about joining the OpenCL Advisory Panel as an individual
  - Or ask about joining KSCAF if you are involved in Safety Critical development
- **Neil Trevett**
  - ntrevett@nvidia.com
  - @neilt3d