Khronos in the World of Open Source and Machine Learning

Markus Tavenrath, Khronos ML Council Chair

7th April 2025

GOSIM

Khronos Connects Software to Silicon

Founded in 2000
~ 160 Members | ~ 40% US, 30% Europe, 30% Asia
ISO/IEC JTC 1 PAS Submitter

KHRONOS GROUP

Non-profit Standards Consortium creating open, royalty-free standards

Focused on runtime APIs and file formats for 3D, XR, AI, vision, parallel compute acceleration

Member-driven, open to any company

Open Standards and Open Source

Open Standards

INTEROPERABILITY via precisely specified (and conformance tested) COMMUNICATION

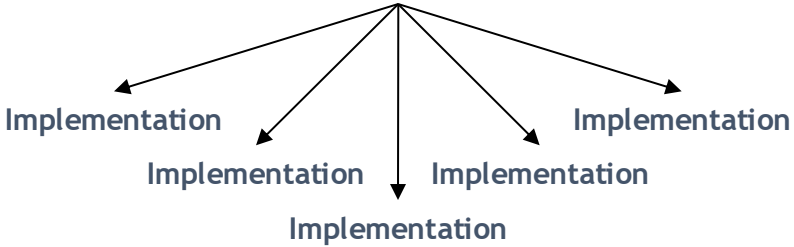
E.g., software to hardware, client to server

Open Source

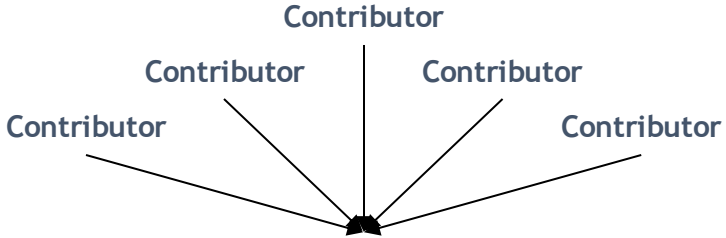
Collective collaboration to share engineering effort

Under a well-defined Contributor License Agreement

Open Standard = Shared Specification



Open Source = Shared Implementation



A technology can be widely deployed to meet multiple market needs without fragmentation

Often used for HARDWARE APIs to enable healthy competition between diverse implementations

Consistency and transparency through a single openly available implementation

Often used for SOFTWARE libraries & compilers when no advantage to competing over multiple implementations

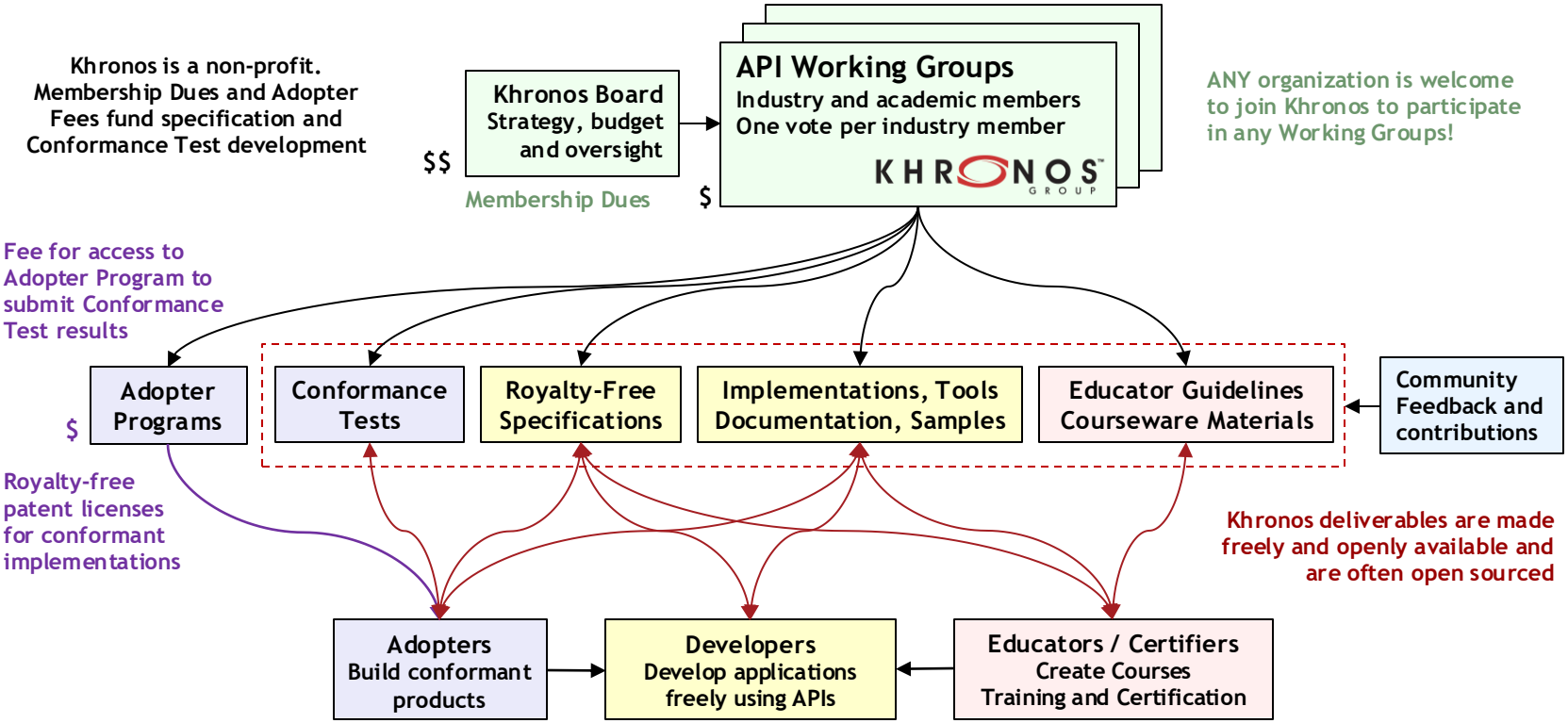
Open standards often use open source for tooling & sample implementations

Benefits of Open Interoperability Standards

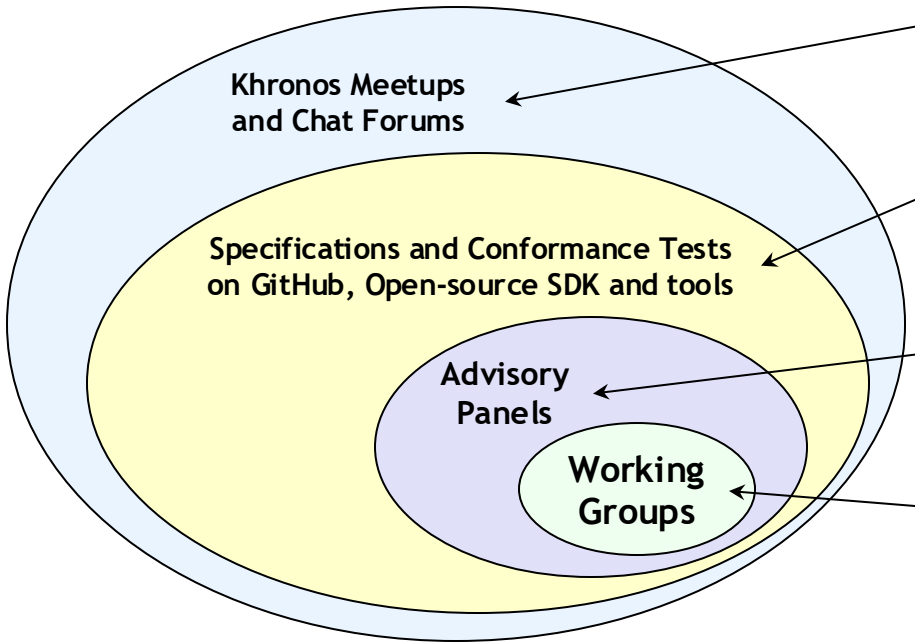
- **Proven solutions – often available royalty free**
  - Leveraging significant industry effort and industry expertise
- **Benefits for hardware and software developers**
  - Cross-platform application portability and reusability
  - Industry-wide ecosystem of tools and libraries
- **Benefits for embedded markets**
  - Decoupled software and hardware development, integration, and safety certification
  - Cross-generation reusability and field upgradability

Why Open Standards?	**Expand Commercial Opportunity** Network effect of compatible products & services	**Reduce Costs** Share design effort and drive increased volume
	Avoid Market Friction Reduce fragmentation and confusion	**Speed Time to Market** Leverage proven functionality and testing
When?	**When Technologies are Proven** Avoid R&D by standards committee	**Consensus Need** Downsides of no available standard widely obvious
How?	**Multi-company Governance to Build Trust** Avoid single-company control or dependency	**Well-defined IP Rights Policy** Royalty-free standards drive wide adoption
	Innovation through Flexible Extensibility Extensions meet timely customer & market needs	**Innovation through Careful Abstraction** Freedom to innovate implementation details

Khronos Cooperative Framework



Khronos Ecosystem Engagement



$0 Open to all!
https://community.khronos.org/
www.khr.io/slack

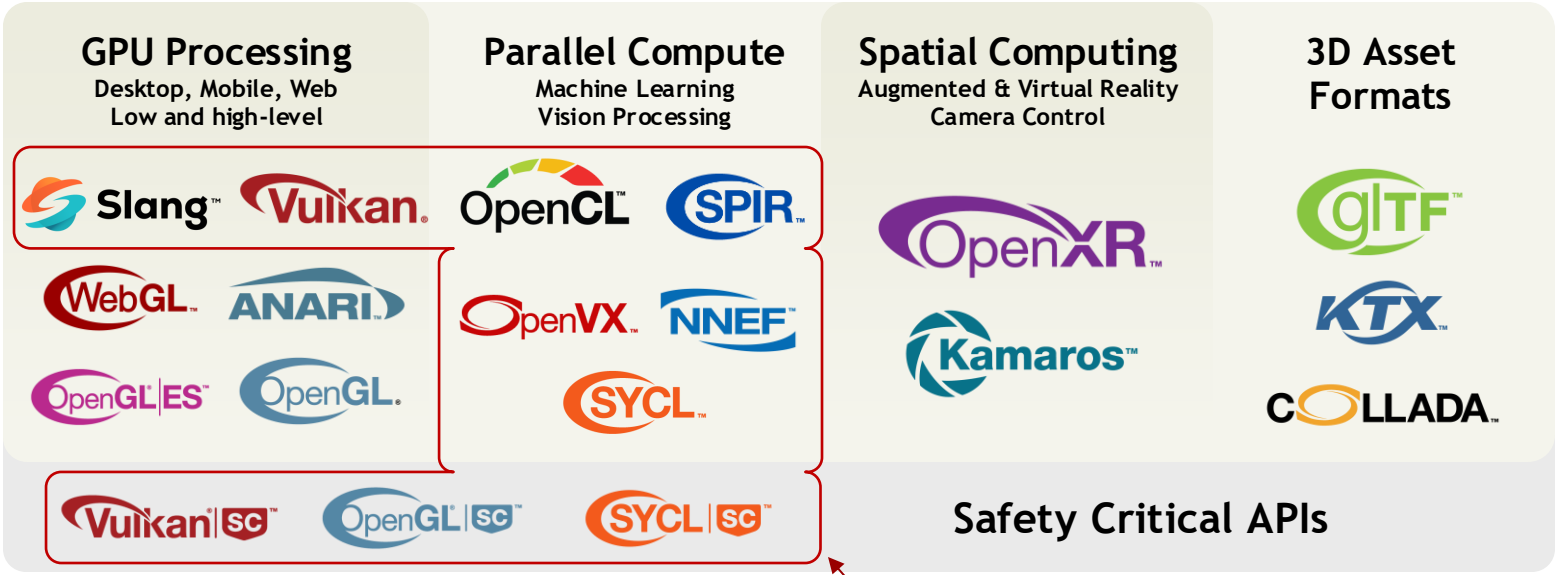
$0 Spec fixes and suggestions. Open source contributions under repo's CLA, typically Apache 2.0
https://github.com/KhronosGroup

$0 Advisors under the Khronos NDA and IPR Framework can comment and contribute to requirements and draft specifications
https://www.khronos.org/advisors/

$ Khronos members under Khronos NDA and IPR Framework participate and vote in working group meetings
https://www.khronos.org/members/

Khronos creates specifications and tools without an NDA as far as possible
BUT *hardware APIs* often need discussion of confidential technology roadmaps
This makes an NDA and IPR framework essential

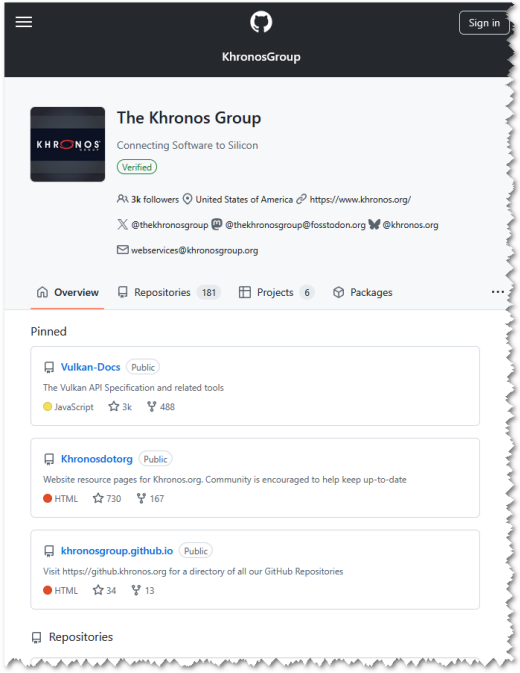
Khronos Active Standards



Khronos standards providing native compute acceleration

Khronos Open-Source Tooling Projects

- **Khronos currently hosts 181 GitHub projects - and counting**
  - https://github.com/KhronosGroup/
- **Specificatations**
- **Conformance Test Suites**
  - To test and confirm API functionality
  - Open source enables early, pre-conformant testing
- **API Ecosystem Support**
  - Educational samples
  - Loaders and layers
  - Utility libraries
  - Language compilers
- **All are invited to file issues**
  - Khronos Apache 2.0 CLA
  - https://www.khronos.org/legal/Khronos_Apache_2.0_CLA



Khronos Slang Open-Source Shading Language

- **Modern, responsive, domain-specific shading language for 3D developers**
  - Boosts productivity and rapidly exposes new technologies and techniques e.g., Neural Graphics
- **Leveraging 15 years of R&D and deployment experience**
  - Originally hosted at NVIDIA from 2017
- **Now hosted at Khronos hosting to foster industry-wide collaboration and innovation**
  - **Open governance** provides all an equal chance to influence and decide Slang's evolution
- **Slang Initiative organized to preserve and enhance open-source project responsiveness**

Technical work 100% under streamlined open-source project

Welcome contributors from any engaged company

Community-driven project structure and best practices suited to a shading language

Working Group bit only for logistical and funding support

Enable the open-source project to focus on technical forward progress

Explore and leverage synergy between Slang, Vulkan and SPIR-V

Slang is the first Khronos Initiative where the *primary* deliverable is open source and *not* an open standard interoperability specification

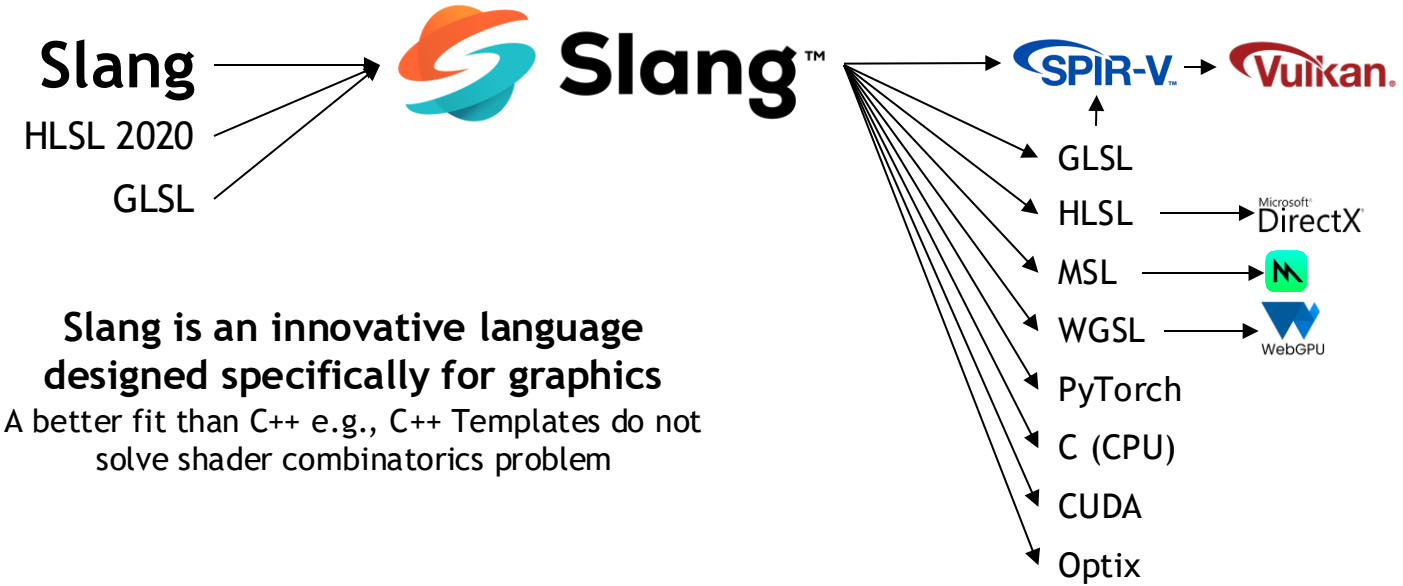
https://github.com/slang-lang/

Slang Open-Source, Cross-Platform Compiler

On-ramp existing code bases and incrementally transition to a modern language

Improved maintainability of large-scale code bases, language expressivity, machine learning

Write Once - Run 'Everywhere' with multiple, diverse compiler backends



Slang is an innovative language designed specifically for graphics

A better fit than C++ e.g., C++ Templates do not solve shader combinatorics problem

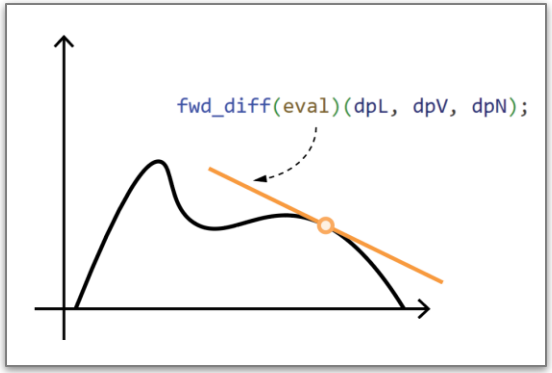
Maintainable Differentiable Code with Autodiff

- **Differentiable functions power gradient descent solution approaches**
  - Slang brings automatic differentiation to languages optimized for GPU usage
  - Developers can optionally provide custom derivatives for just the portions of a shader where it's necessary – flexibility & control
  - Autodiff support includes arbitrary control flow & dynamic dispatch
- **Vulkanised 2025 Presentation**
  - Slang is for Neural Graphics

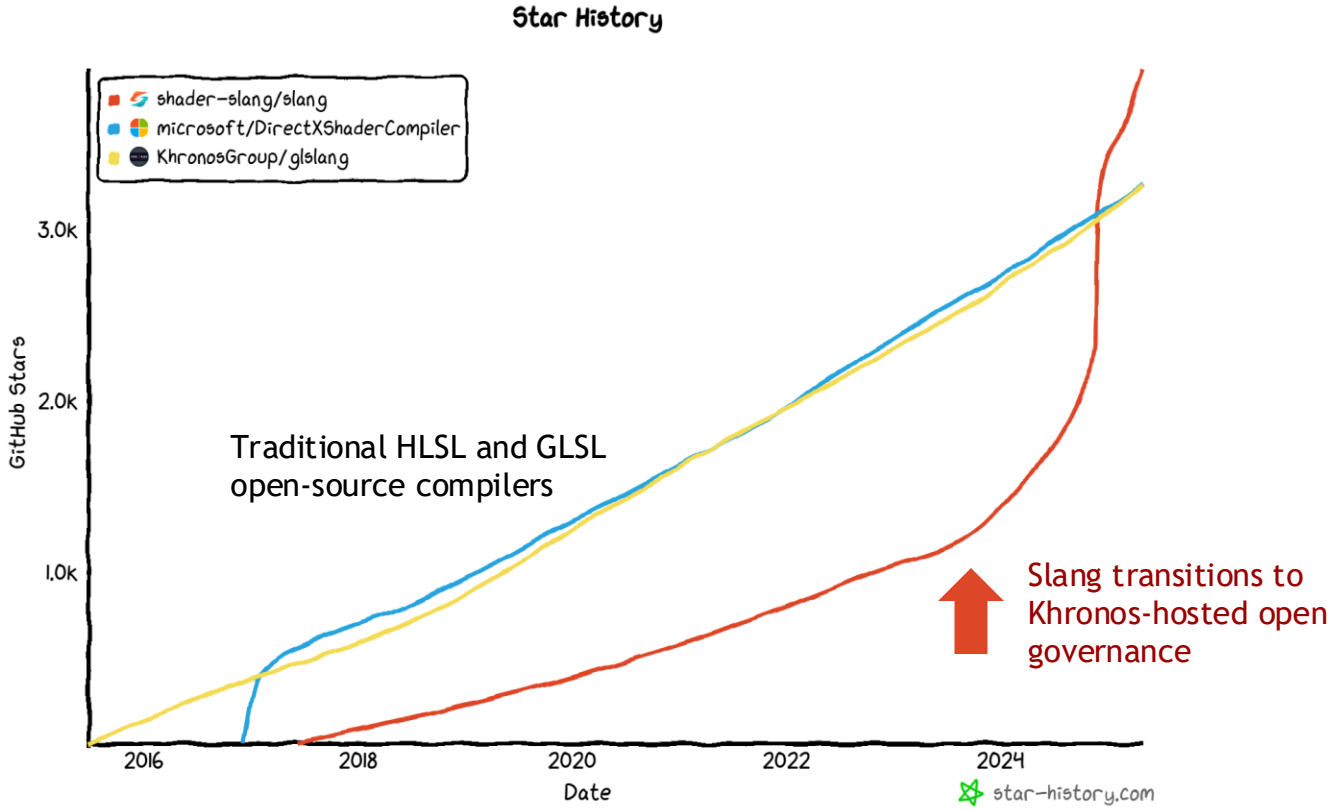
```
interface IBRDF : IDifferentiable
{
    [Differentiable] float3 eval(float3 L, float3 V, float3 N);
}

struct GGXBRDF : IBRDF
{
    float3 baseColor;
    float roughness;
    float metallic;
    float specular;

    [Differentiable] float3 eval(float3 L, float3 V, float3 N)
    {
        float NdotL = dot(N, L);
        float NdotV = dot(N, V);
        if (NdotL < 0 || NdotV < 0)
```



Slang Industry Interest



Khronos Compute Acceleration

Choice of programming models to meet the needs of diverse developers

Higher-level applications, libraries, and languages and APIs often use lower-level standards for hardware access

Higher-level Languages /APIs
Streamlined development and performance portability

- **NNEF** — Neural Network Exchange Format Trained Networks
- **OpenVX** — Graph-based vision and inferencing acceleration
- **Slang** — Open-Source, Cross-Platform Shading Language
- **SYCL** — Single source C++ programming with compute acceleration

Lower-level IRs /APIs
Explicit hardware control

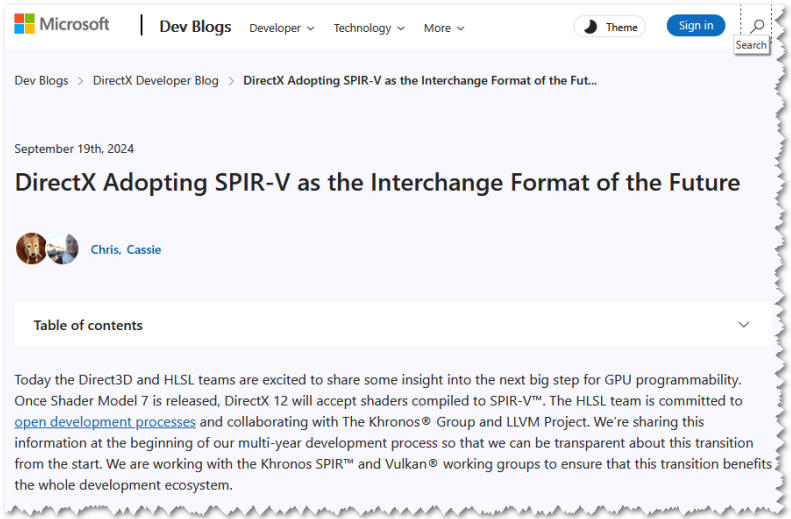
- **Vulkan** — GPU rendering + compute acceleration → GPU
- **SPIR-V** — Intermediate Representation (IR) language compiler target with parallel execution and graphics
  - Shaders ←
  - Kernels →
- **OpenCL** — Heterogeneous compute acceleration → FPGA, CPU, GPU, DSP, Custom HW, AI/Tensor HW

SPIR-V Intermediate Representation

- **SPIR-V is a binary intermediate representation interchange format**
  - Compiler target for graphics shader and compute kernel programs
  - Includes expression of parallel execution and graphics functionality
- **Encourages a rich compiler ecosystem**
  - Decouples APIs from compiler implementation and eliminates need for in-driver-compilers
  - Drivers use open-source compilers for consistency, reduced effort to support diverse languages
- **Why not use LLVM-IR?**
  - ‘SPIR’ precursor to ‘SPIR-V’ expanded on LLVM-IR for graphics and compute
  - BUT LLVM-IR is not version stable - ongoing binary compatibility is critical for hardware drivers
  - New LLVM versions can read older LLVM-IR, but cannot write older LLVM-IR
  - Reading and writing LLVM bitcode requires LLVM tools
- **SPIR-V is a simple & stable binary format suitable for ingestion into hardware drivers**
  - SPIR-V tools contain API & commands for SPIR-V processing
  - SPIRV-Cross tool provides reflection and human readable disassembly to GLSL, HLSL and MSL

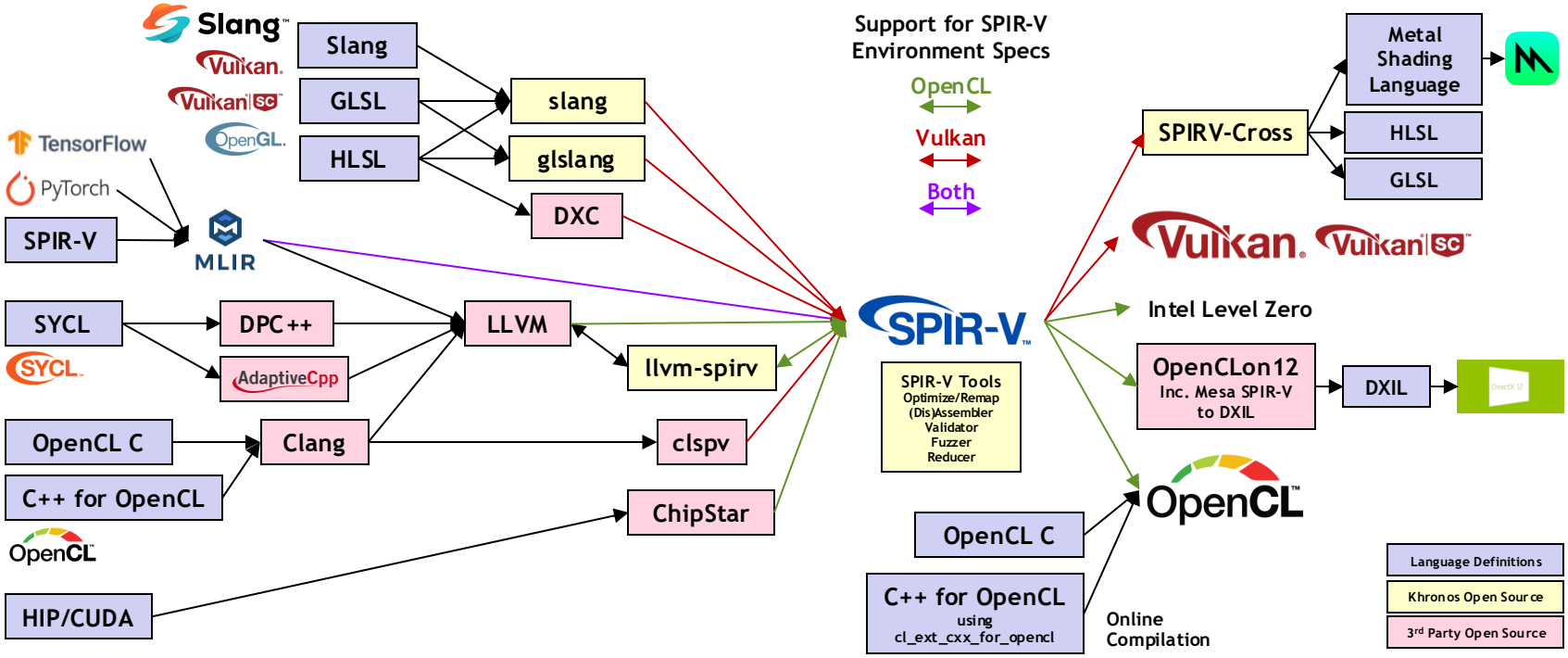
Growing SPIR-V Industry Support

- **Growing number of open-source compilers target Slang**
  - Khronos glslang: GLSL to SPIR-V compiler & validator
  - Microsoft DXC compiles HLSL to SPIR-V
  - Khronos Slang compiles Slang, GLSL & HLSL 2020 to SPIR-V
  - LLVM 20.X promotes SPIR-V to an official backend
- **Microsoft is adopting SPIR-V**
  - For HLSL Shader Model 7 onwards
  - Expanding support for LLVM's SPIR-V backend



SPIR-V Ecosystem

SPIR-V is central to the compute and graphics compiler ecosystem



MLIR is part of the LLVM compiler infrastructure

API Layering Increasingly USed

Language layering often leverages SPIR-V

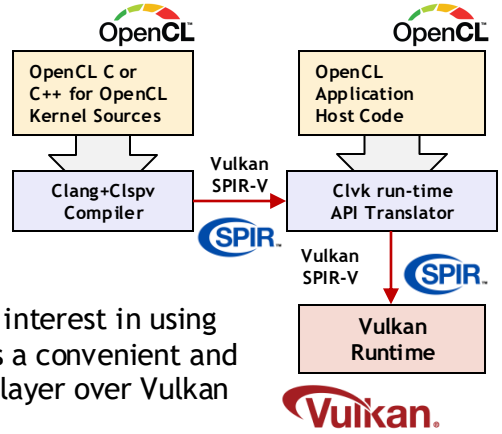
Layers Over	Vulkan	OpenGL	OpenCL	OpenGL ES	DX12	DX8-11
Vulkan		Zink	Rusticl + Zink clspv + clvk/Angle	Angle GLOVE	vkd3d-Proton vkd3d	DXVK WineD3D
OpenGL	Ashes			Angle		WineD3D
DX12	Dozen	Microsoft 'GLOn12'	Microsoft 'CLOn12'			Microsoft D3D11On12
DX9-11	Ashes			Angle		
Metal	MoltenVK			Angle MoltenGL		

ROWS Benefit Platforms by adding APIs
Enable content without additional kernel level drivers

COLUMNS Benefit ISVs by making an API available everywhere
Application deployment flexibility by fighting platform fragmentation
Making an API available across multiple platforms even if no native drivers available

Community Open Source for Khronos APIs

- **Khronos is supportive of community open-source efforts implementing its APIs**
  - Khronos grants free access to Khronos Adopter Program to efforts such as MESA
  - Encouraging full conformance
- **Vulkan**
  - Microsoft ‘Dozen’ layered over DX12
  - MoltenVK layered over Metal
  - MESA RADV for AMD, ANV for Intel, NVK for NVIDIA, PanVK for Arm Mali, Lavapipe for CPUs
  - Google Swiftshader for CPUs using advanced vectorization
  - Igalia Vulkan driver for Raspberry Pi
- **OpenCL**
  - Microsoft ‘CLon12’ layered over DX12 leverages MESA and LLVM
  - POCL uses Clang/LLVM with SPIR-V import
  - MESA Rusticl uses MESA Gallium infrastructure, with SPIR-V import
  - Arm clvk (API translation), clspv (compiler using Clang) over Vulkan
  - Samsung Angle/Ancle (API layer) and clspv layered over Vulkan



Growing interest in using OpenCL as a convenient and compute layer over Vulkan

OpenCL – Low-level Parallel Programing

Programming and Runtime Framework for Application Acceleration

Offload compute-intensive kernels onto parallel heterogeneous processors
CPUs, GPUs, DSPs, FPGAs, Tensor Processors
OpenCL C or C++ kernel languages

Platform Layer API

Query, select and initialize compute devices

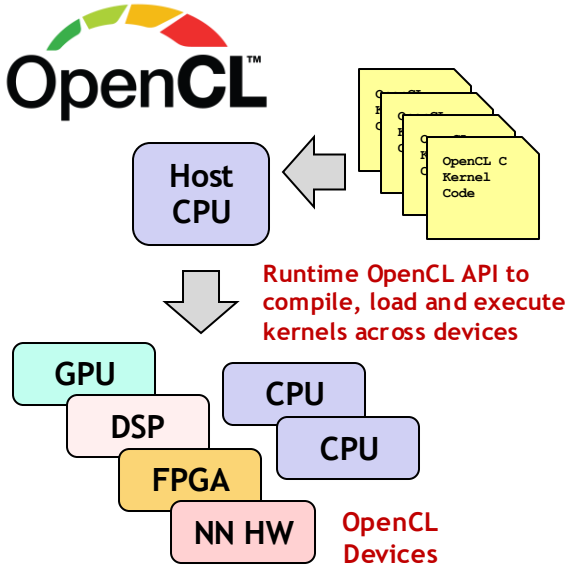
Runtime API

Build and execute kernels programs on multiple devices

Explicit Application Control

Which programs execute on what device
Where data is stored in memories in the system
When programs are run, and what operations are dependent on earlier operations

OpenCL is under very active development



Complements GPU-only APIs

Simpler programming model
Relatively lightweight run-time
More language flexibility, e.g., pointers
Rigorously defined numeric precision

OpenCL and Machine Learning

- **Programming and runtime framework for parallel heterogeneous processors**
  - Offload compute-intensive kernels onto diver hardware: CPUs, GPUs, DSPs etc.
- **OpenCL is often used as a backend for ML compilers and inference engines**
  - Especially in the embedded and mobile markets
- **OpenCL has a robust future pipeline of AI-related extensions**
  - Recordable command buffers, including mutable command buffers
  - 'Cooperative matrix' for standard access to dedicated matrix hardware
  - New AI data types, such as bfloat16 and fp8

Machine Learning frameworks, libraries and compilers using OpenCL for offload acceleration

LiteRT Google

OpenVINO Intel

Arm Compute Library

Acuity VeriSilicon

飞桨 PaddlePaddle Baidu

GLOW Meta

SYCL

MACE Xiaomi

Qualcomm Neural Processing SDK for AI

TI DL Library (TIDL)

tvm Apache

MetaWare EV Synopsis

MNN Alibaba

NNEF - Neural Network Exchange Format

- **Open, royalty-free standard for representing neural network models**
  - Enables model exchange between training frameworks and inference engines
  - Stable open standard specification for direct import to hardware drivers
- **Describes the network's architecture, operations, and trained parameters**
  - Designed with a human-readable syntax
  - Built for future extensibility to support new advancements
- **Backed by open-source tools**
  - Parsing, validation, conversion to and from the NNEF format
- **NNEF does not define a runtime API**
  - OpenVX has extension to import NNEF networks into its runtime

Adopters	Usage	URL
OpenVX	Direct network import to OpenVX import	https://www.khronos.org/openvx/
aiMotive	Import to NN Inference Engine	https://aimotive.com
CoreAVI	Import to ComputeCore inference Engine	https://coreavi.com
TVM	Work ongoing to add NNEF import	https://tvm.apache.org/

SYCL - Single Source C++ Heterogeneous Compute

- **Single-source C++ for parallel heterogeneous compute**
  - Enables parallel code to be written in standard C++
  - Executed across diverse hardware such as CPUs, GPUs, FPGAs, and other accelerators
- **Builds on OpenCL concepts**
  - Developer-friendly C++ interface using core concepts of OpenCL
- **Task-based parallelism**
  - Runtime manages execution of application defined tasks and dependencies
- **Focus on developer productivity, aim to simplify parallel programming:**
  - Unified memory management, Exception handling, Template-based programming

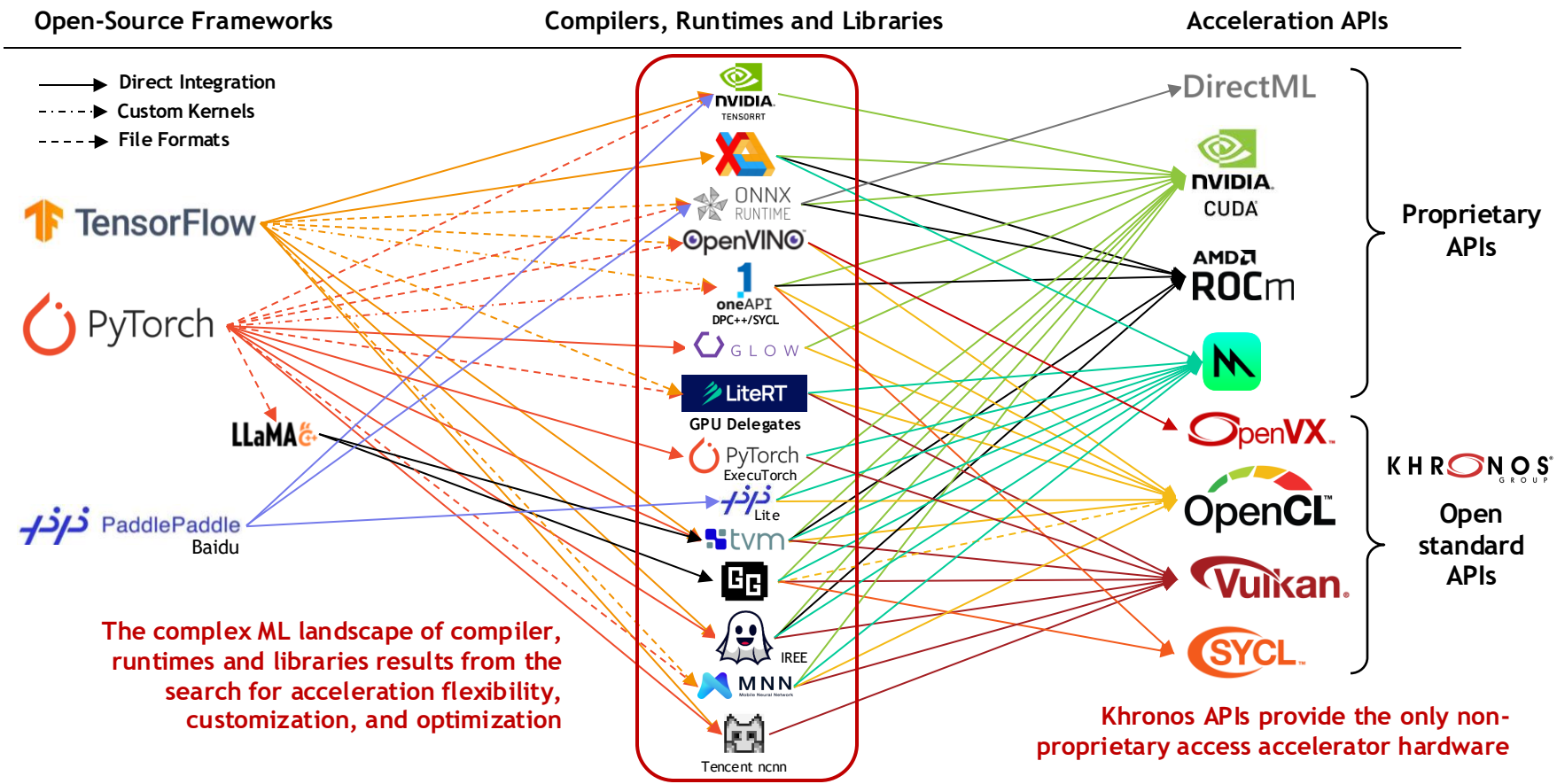
Project	Description	URL
Syclops	Using SYCL to target novel RISC-V-based AI accelerators	https://www.syclops.org/
Pytorch	Sycl backend since version 2.4	https://pytorch.org
LLama.cpp	SYCL backend for acceleration on Intel, AMD, NVIDIA GPUs	https://github.com/ggml-org/llama.cpp
UXL Foundation	Builds an open heterogeneous compute ecosystem built in SYCL Providing the oneDNN library of primitives for deep learning frameworks	https://uxlfoundation.org/

Vulkan for ML Inferencing

- **'Explicit' low-level graphics and compute API providing detailed GPU control**
  - Widely adopted and deployed in mobile, embedded desktop and cloud platforms
  - Native drivers for Windows, Linux, Android and Nintendo Switch
- **Extensions provide an increasing range of high-performance ML-focused primitives**
  - VK_KHR_cooperative_matrix - matrix-matrix multiply in a subgroup
  - VK_NV_cooperative_matrix2 - matrix-matrix multiply in a workgroup scope
  - VK_NV_cooperative_vector - matrix-vector multiplies with fp8/fp16 & int8 support
  - Precisions: VK_KHR_shader_bfloat16 and VK_KHR_shader_float16_int8

Vulkan is being increasingly used to provide backend acceleration in inference frameworks

Machine Learning Acceleration Complexity



Khronos ML Market Research

- **Can open standard acceleration API reduce this market friction and complexity?**
  - Faster performance, less development, porting and support costs
- **What are the real-world industry needs that would have to be met?**
  - Do we need a common ML hardware acceleration stack for CPUs, NPUs and GPUs?
  - Can we achieve performance portability across diverse hardware architectures?
  - Can avoid need for deep knowledge of details like memory, cache and workgroup/subgroup size?
- **What are the most promising potential solutions to explore?**
  - Is a graph abstraction optimal for performance optimization?
  - Which operator sets to support? NNEF / TOSA / ONNX / Aten?
  - How to handle custom operators?
  - How to solve generic quantization?

Khronos is initiating funded market research to try to answer these questions and more!

Seeking first round of input in this online survey

Please take few minutes - your input is appreciated!

Khronos Group AI Opportunity Survey

Call for Input and Participation

- **Help us ensure Khronos standards meets your needs and requirements!**
  - You participation and feedback is very welcome
- **Please participate in the ML Survey!**
  - Let us know if you would like to provide more detailed requirements and feedback
- **All are welcome to join Khronos to directly influence API design and evolution**
  - www.khronos.org/members/
  - Email memberservices@khronosgroup.org
- **More information on all Khronos APIs**
  - https://www.khronos.org/
- **Contact me directly**
  - matavenrath@nvidia.com