OpenCL and Compute Acceleration Standards

Neil Trevett
Khronos President
OpenCL Working Group Chair
NVIDIA VP Developer Ecosystems

November 2021

Topics

Brief overview of Khronos compute acceleration standards

And why they might be of interest to the RISC-V Community

Deeper dive into OpenCL

Including roadmap developments

Discussion on how Khronos and RISC-V could collaborate

Khronos is open to any organization – please get directly involved if you wish!

We welcome feedback and cooperation between organizations

These slides will be available online

www.khronos.org

KHRONOS GROUP

Khronos Connects Software to Silicon

Founded in 2000
>180 Members ~ 40% US, 30% Europe, 30% Asia

KHRONOS GROUP

Open, royalty-free interoperability standards to harness the power of GPU, XR and multiprocessor hardware

3D graphics, augmented and virtual reality, parallel programming, inferencing and vision acceleration

Non-profit, member-driven standards organization, open to any company

Proven multi-company governance and Intellectual Property Framework

KHRONOS GROUP

Open and Royalty Free Standards

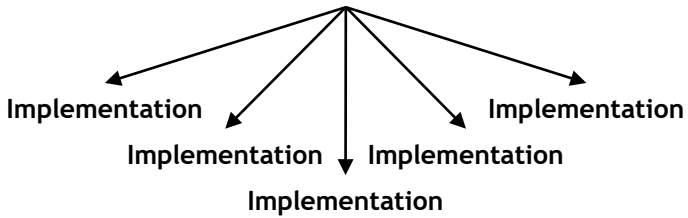
‘Open’ means…

Open to all who wish to participate in their creation
Created under transparent, well-defined multi-company governance
No company has superior voting or ownership rights
Designs based on technical merit
No restrictions on who can implement and adopt

‘Free’ means…

No charge for access to specification documents
No charge to users of specifications
Royalty-free patent license for implementors from members
More Member Patents == More Protection

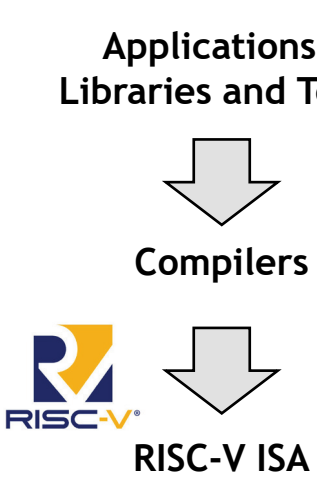
Effective Open Standard
=
Precise Specification
+
Conformance Tests



APIs and RISC-V

CPU ISAs are relatively stable

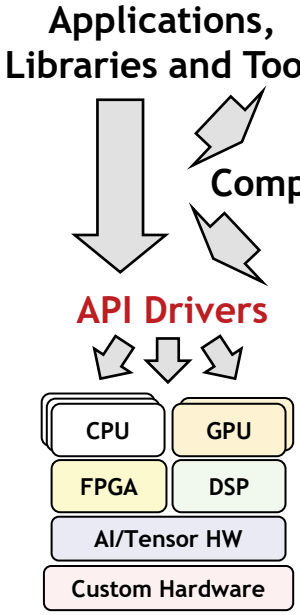
Can be targeted directly by compilers and tools



Accelerated applications on RISC-V need both compilers and APIs working together

Accelerator architectures are constantly evolving

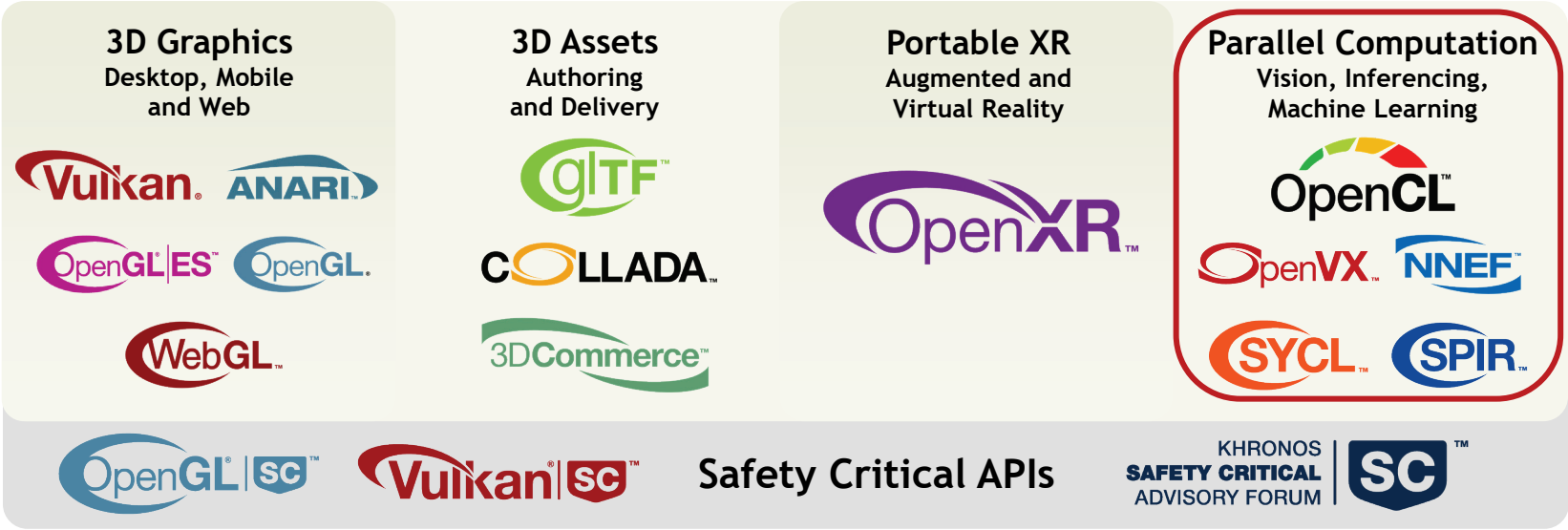
APIs needed for portability across diverse processors



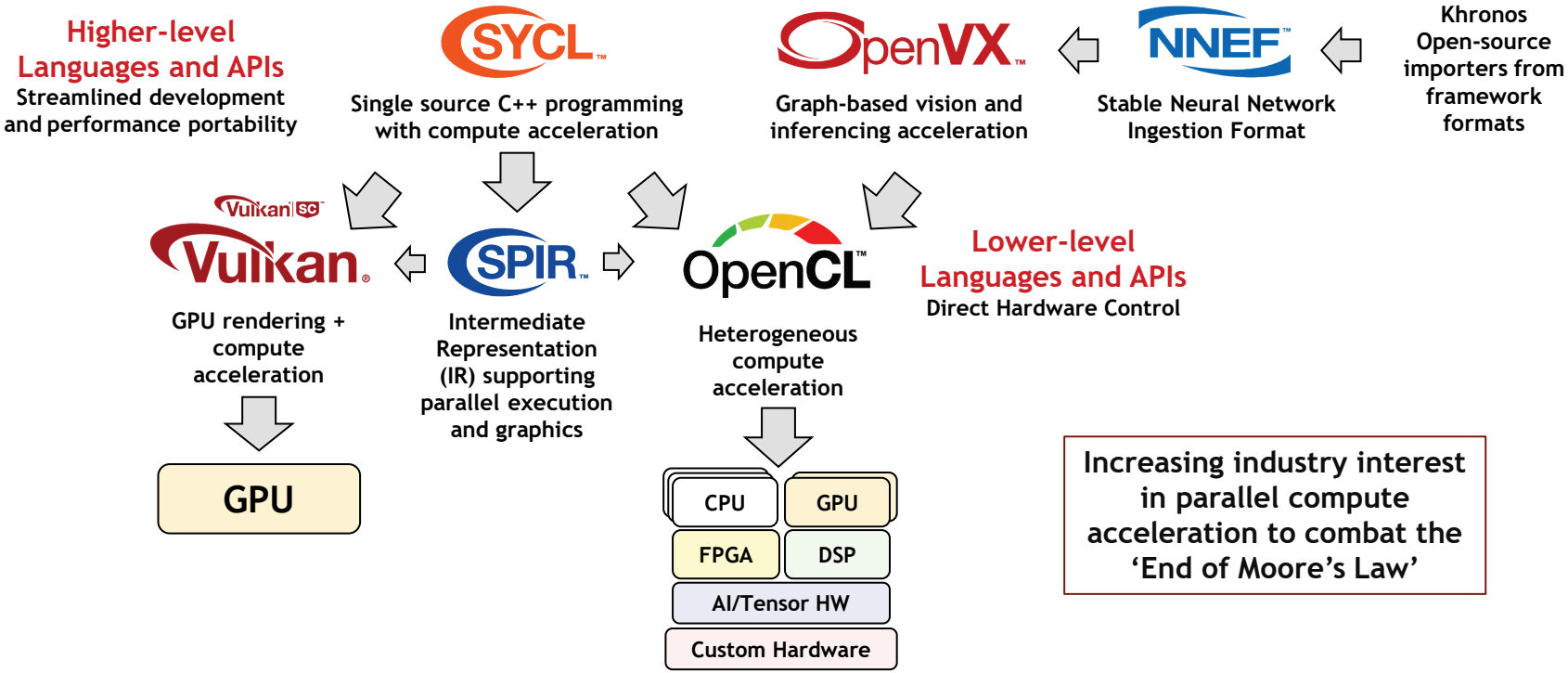
Open Standard APIs

Share development costs
Ensure software portability
Speed time to market
Do not limit innovation

Khronos Active Standards



Khronos Compute Acceleration Standards



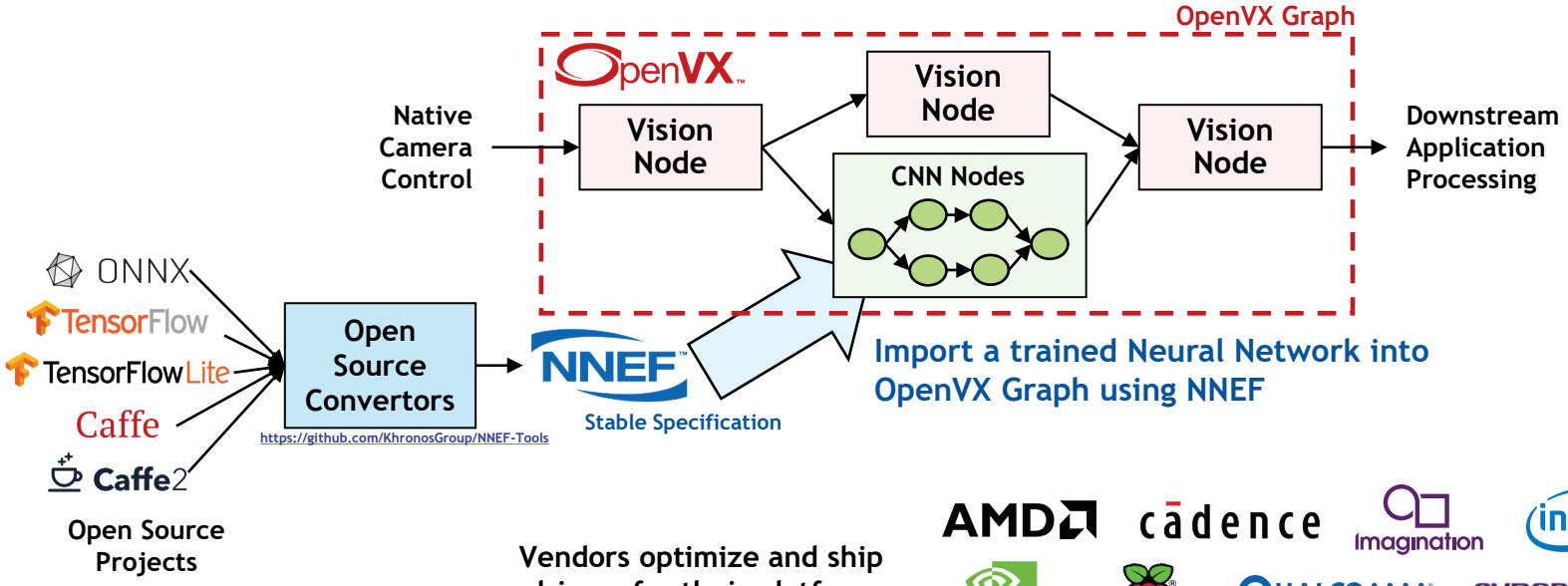
Increasing industry interest in parallel compute acceleration to combat the 'End of Moore's Law'

OpenVX Cross-Vendor Vision and Inferencing

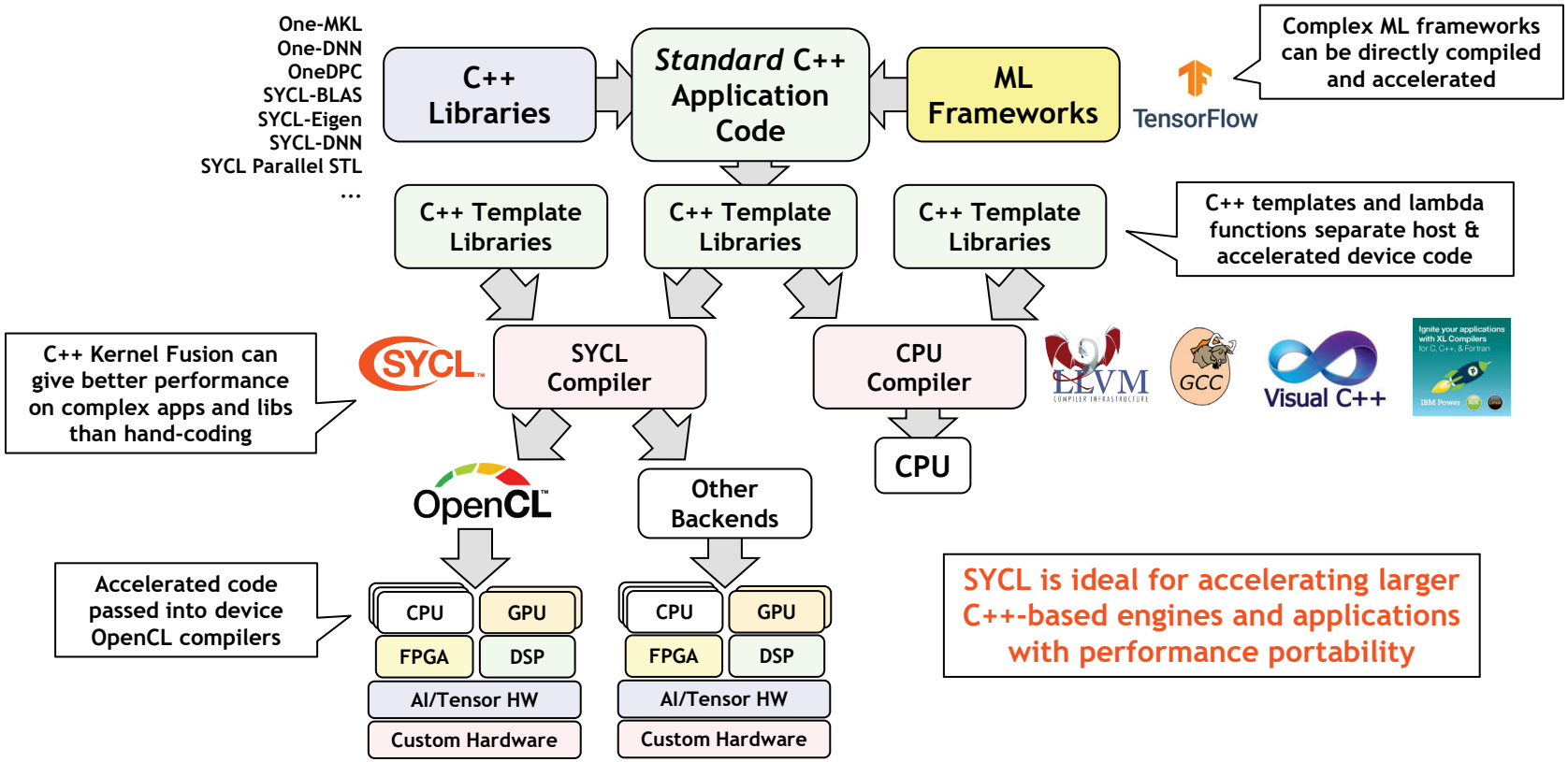
High-level graph-based abstraction for portable, efficient vision processing

Implementable on almost any hardware or processor with no performance portability

Graphs contain vision processing and NN nodes for global optimization



SYCL Single Source C++ Parallel Programming



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

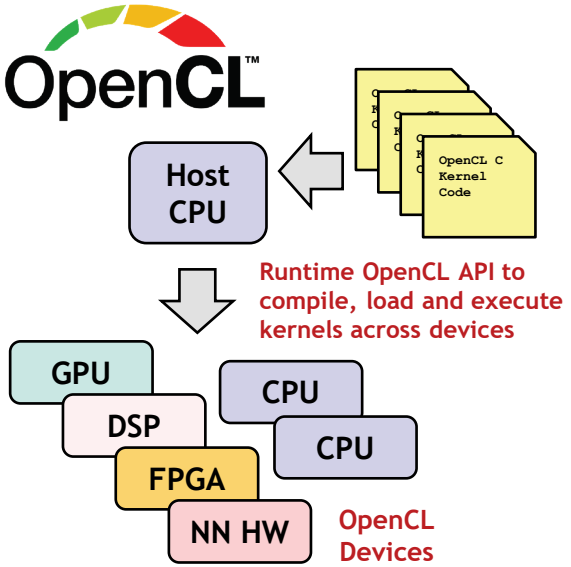
Query, select and initialize compute devices

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

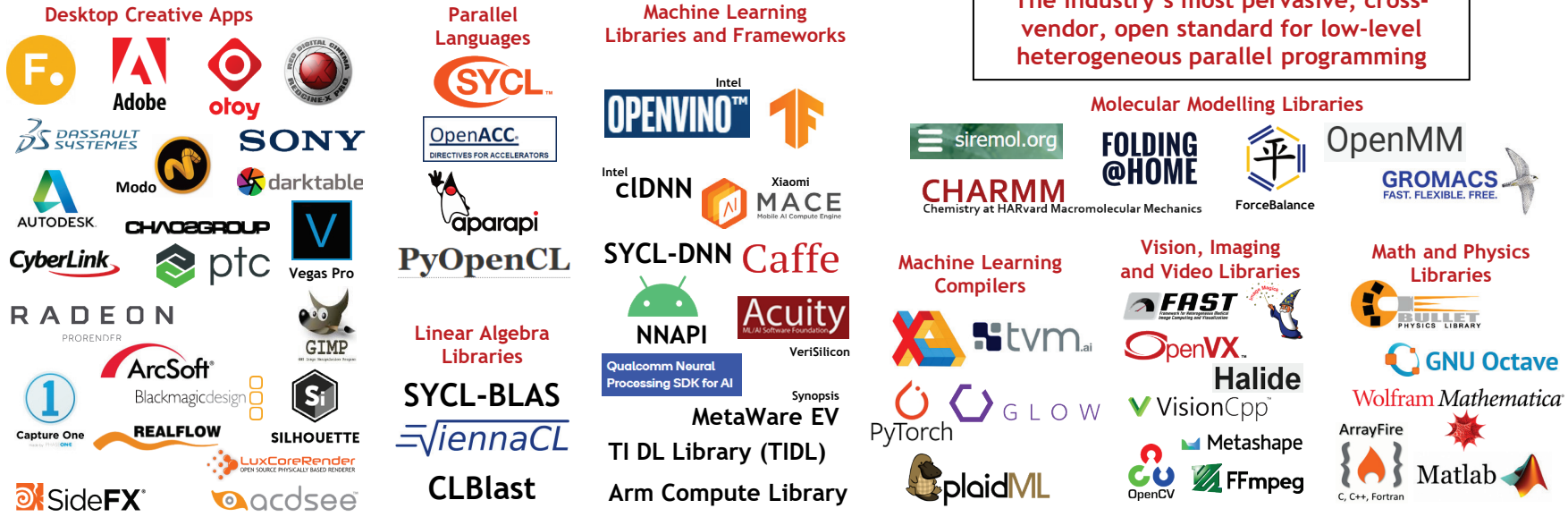


Complements GPU-only APIs

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

OpenCL is Widely Deployed and Used

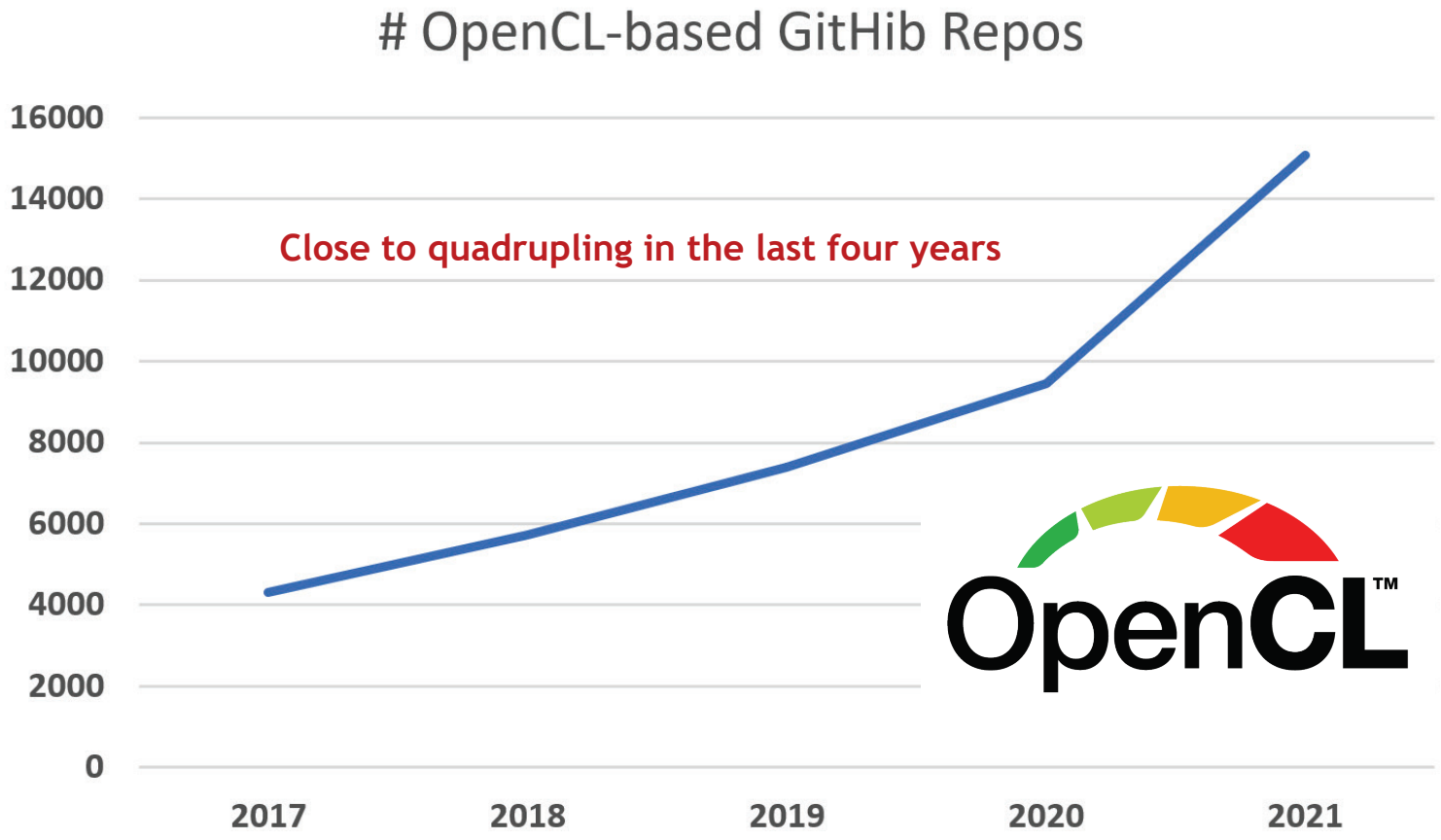
The industry's most pervasive, cross-vendor, open standard for low-level heterogeneous parallel programming



Conformant Implementations

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

OpenCL Open-Source Project Momentum



OpenCL 3.0

Increased Ecosystem Flexibility

All functionality beyond OpenCL 1.2 queryable
Macros for optional OpenCL C language features
Widely adopted extensions to be integrated into core

OpenCL C++ for OpenCL

Open-source C++ for OpenCL front end compiler combines OpenCL C and C++17 replacing OpenCL C++ language spec

Unified Specification

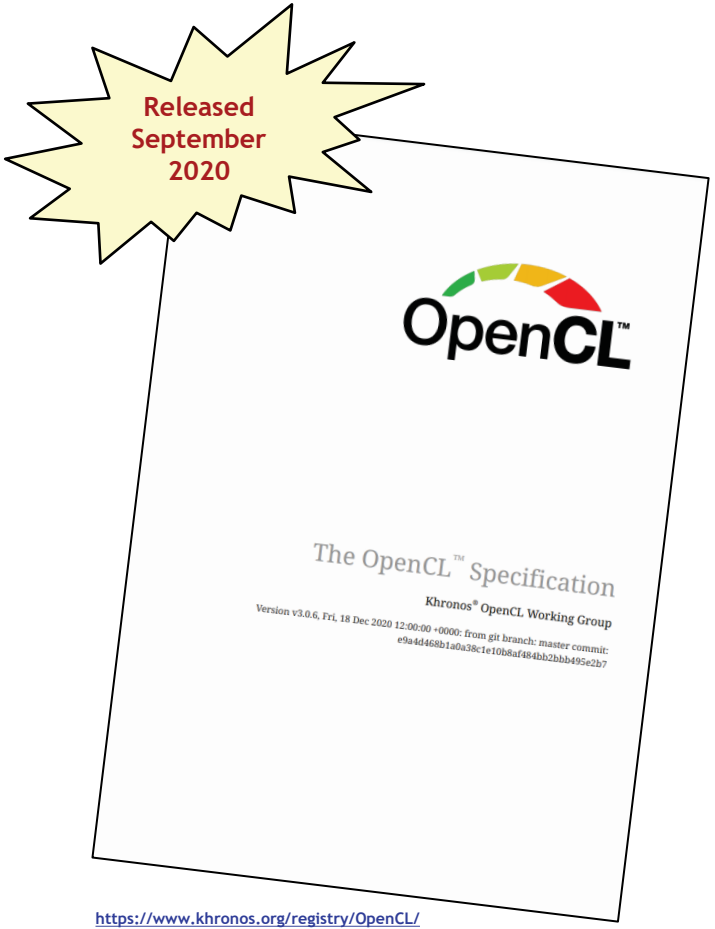
All versions of OpenCL in one specification for easier maintenance, evolution and accessibility
Source on Khronos GitHub for community feedback, functionality requests and bug fixes

New Functionality

Subgroups with SPIR-V 1.3 in core (optional)
Asynchronous DMA extension for embedded processors

Easy OpenCL 3.0 migration for applications

OpenCL 1.2 applications - no change
OpenCL 2.X applications - no code changes if all used functionality present
Queries recommended for future portability

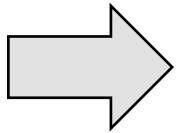


https://www.khronos.org/registry/OpenCL/

OpenCL 3.0 Adoption

OpenCL 3.0 Adopters

codeplay
Microsoft
QNX
QUALCOMM



arm
Google
Imagination
intel
NVIDIA
VeriSilicon

OpenCL 3.0 Adopters Shipping Conformant Implementations

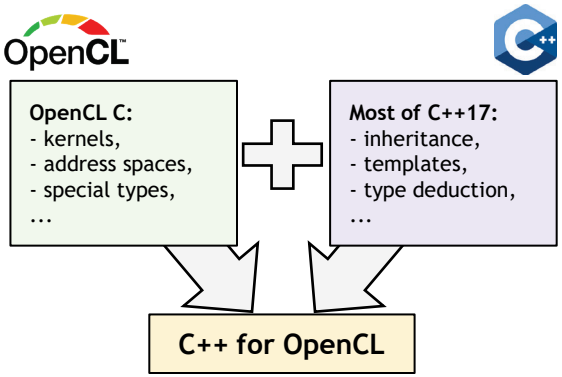
OpenCL™

Product Conformance Status

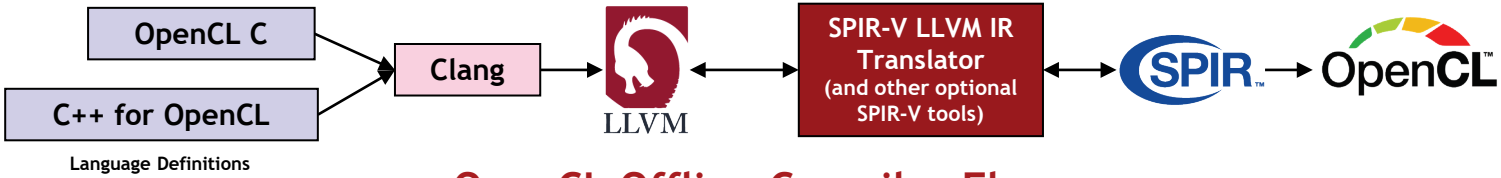
https://www.khronos.org/conformance/adopters/conformant-products/opencl

C++ for OpenCL

- **Open-Source Compiler Front-end**
  - Replaces the OpenCL C++ kernel language spec
  - Official releases published in OpenCL-Docs repo
- **Enables full OpenCL C and most C++17 capabilities**
  - OpenCL C code is valid and fully compatible
  - Enables gradual transition to C++ for existing apps
  - Language documentation
- **Supported in Clang since release 9.0**
  - Generates SPIR-V 1.0 plus SPIR-V 1.2 where necessary
  - Full details are provided in OpenCL-Guide
- **Online compilation via cl_ext_cxx_for_opencl**



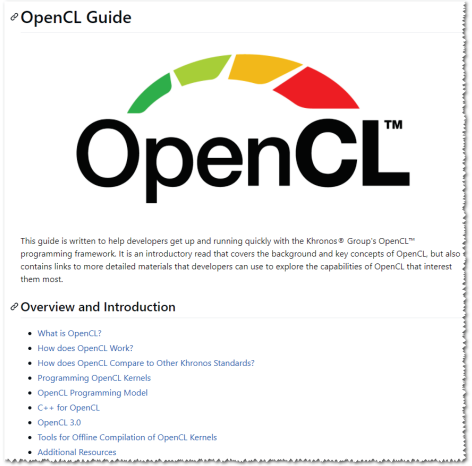
Check it out in
Compiler Explorer



OpenCL Offline Compiler Flow

OpenCL SDK – In Development

- **Bringing together all the components needed to develop OpenCL applications**
  - OpenCL Headers (include/api)
  - OpenCL C++ bindings (include/cpp)
  - OpenCL Utility Libraries (include/utils)
  - Build system and CI
- **Other resources useful to OpenCL developers**
  - OpenCL Guide
  - Code samples (samples/)
  - Documentation (docs/)
- **Loader and Layers**
  - Initial layers implemented
  - SDK and Layers Tutorial
- **Watch GitHub Repo for updates**
  - Community contributions welcome!



https://github.com/KhronosGroup/OpenCL-Guide

More Information at
https://github.com/KhronosGroup/OpenCL-SDK

New OpenCL Extensions Shipped in 2021

Enhanced subgroup functionality
Extended bit-level operations
Queries for a device universally unique identifier
Enhanced queries for platform and device versions
Queries for PCI bus information
SPIR-V support for C++ linkage types
Queries for a suggested local work size
Integer Dot Product for Faster Neural Network Inferencing

All OpenCL core and extension specifications can be found on the OpenCL Registry
https://www.khronos.org/registry/OpenCL/

Asynchronous DMA Extensions

OpenCL embraces a new class of Embedded Processors

Many DSP-like devices have Direct Memory Access hardware

Transfer data between global and local memories via DMA transactions

Transactions run asynchronously in parallel to device compute enabling wait for transactions to complete

Multiple transactions can be queued to run concurrently or in order via fences

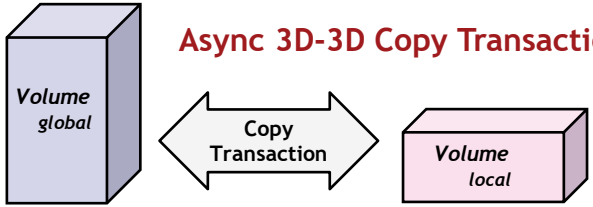
OpenCL abstracts DMA capabilities via extended asynchronous workgroup copy built-ins

(New!) 2- and 3-dimensional async workgroup copy extensions support complex memory transfers

(New!) async workgroup fence built-in controls execution order of dependent transactions

New extensions complement the existing 1-dimensional async workgroup copy built-ins

Async 3D-3D Copy Transaction



Reshaping possible

$V_{global} = V_{local}$

Async Fence controls order of dependent transactions

async_copy$_1$
async_copy$_2$
async_fence
async_copy$_3$

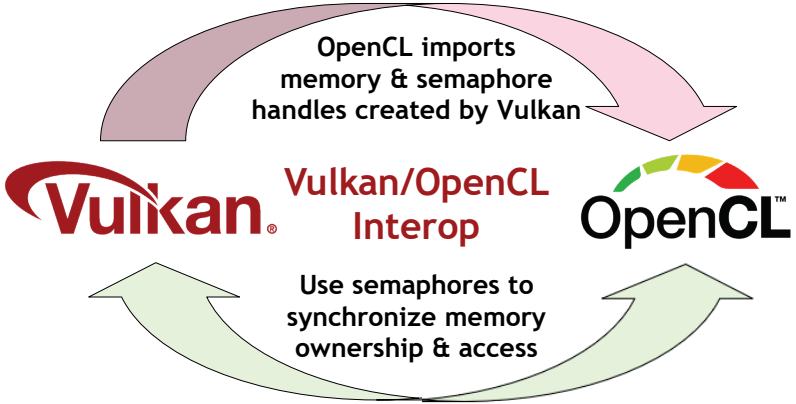
All transactions prior to async_fence must complete before any new transaction starts, without a synchronous wait

The first of significant upcoming advances in OpenCL to enhance support for embedded processors

Semaphore and Memory Sharing with Vulkan

Set of External Sharing Extensions

Generic extensions to import external memory and semaphores exported by other APIs
API-specific interop extensions e.g., Vulkan
More flexible than previous interop APIs using implicit resources



Provisional Extensions for developer feedback before finalization

https://www.khronos.org/blog/khronos-releases-opencl-3.0-extensions-for-neural-network-inferencing-and-opencl-vulkan-interop

OpenCL Roadmap Discussions

Command Buffer Recording and Replay

Expect and Assume Optimization Hints

Unified Shared Memory

Floating-point Atomics

Explicit Cache Control

Indirect Dispatch

Global Barriers

YUV Multi-planar Images

Generalized Image from Buffer

Tensor Objects

Dataflow compute

2D and 3D Prefetch Built In Functions

Developer Feedback Welcome!

What is your highest priority?
What is missing?
Requirements and use cases

Feedback welcome on Specification GitHub
https://github.com/KhronosGroup/OpenCL-Docs

New functionality is proven as extensions before being added to core

API Layering

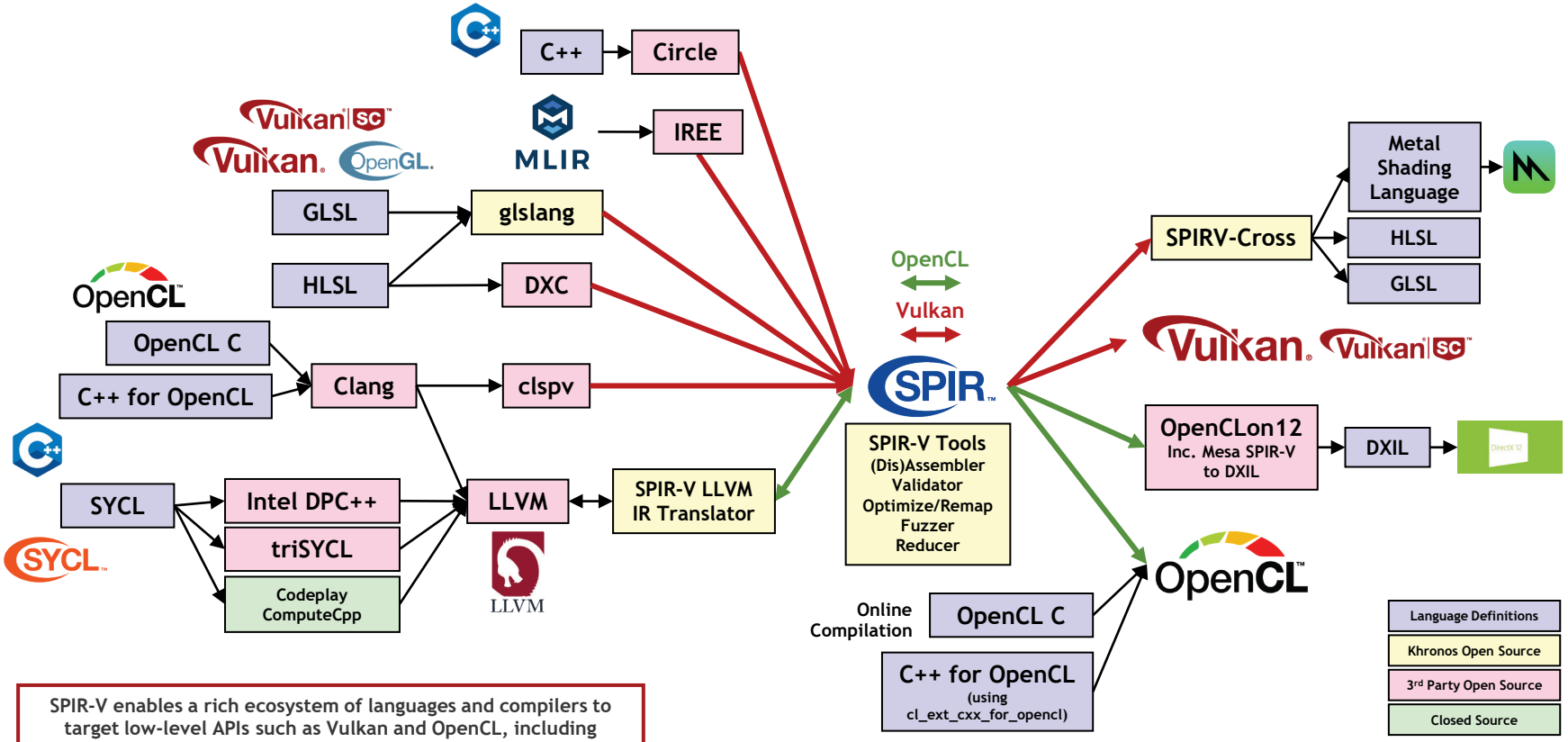
Enabled by growing robustness of open-source compiler ecosystem

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

ROWS Benefit Platforms by adding APIs

COLUMNS Benefit ISVs by making an API available everywhere

SPIR-V Language Ecosystem



SPIR-V enables a rich ecosystem of languages and compilers to target low-level APIs such as Vulkan and OpenCL, including deployment flexibility: e.g., running OpenCL kernels on Vulkan

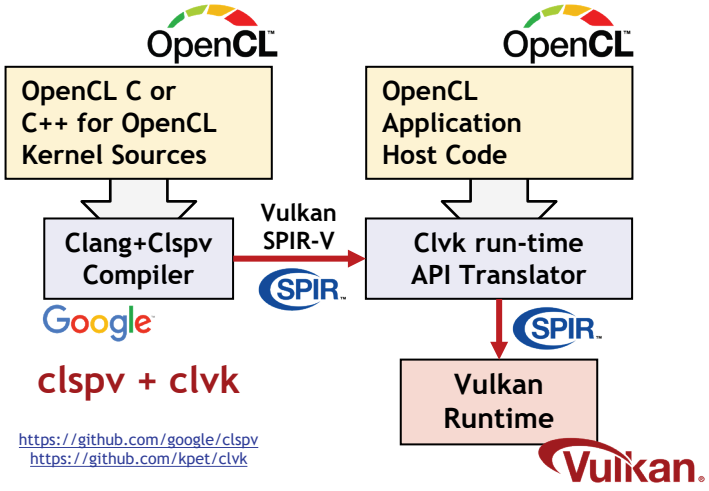
Layered OpenCL Implementations

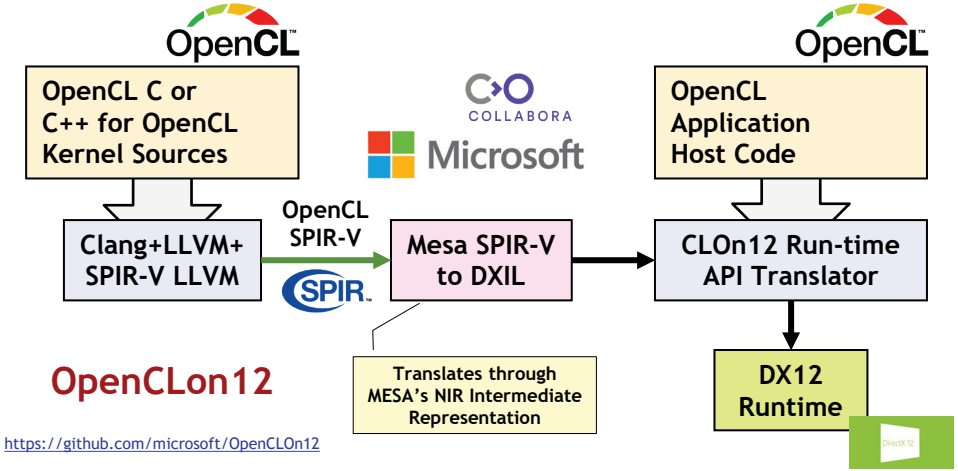
clspv + clvk

clspv - Google's open-source OpenCL kernel to Vulkan SPIR-V compiler
Tracks top-of-tree LLVM and Clang - not a fork
Clvk - prototype open-source OpenCL to Vulkan run-time API translator
Used by shipping apps and engines on Android
e.g., Adobe Premiere Rush video editor - 200K lines of OpenCL C kernel code

OpenCLOn12

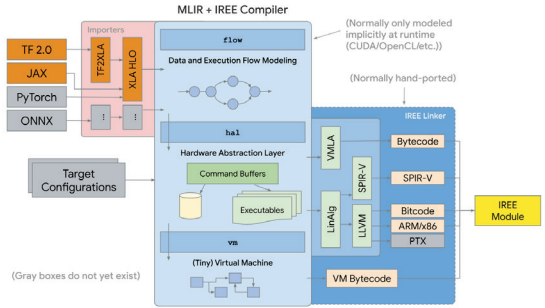
Microsoft and COLLABORA
GPU-accelerated OpenCL on any DX12 PC and Cloud instance (x86 or Arm)
Leverages Clang/LLVM AND MESA
OpenGLOn12 - OpenGL 3.3 over DX12 is already conformant



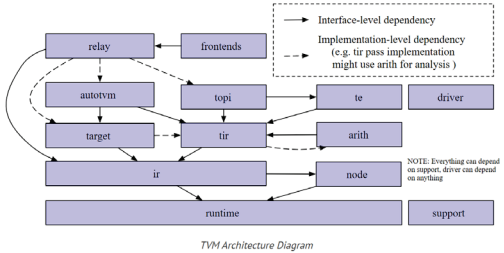


ML Compiler Acceleration Backends

	OctoML	Google	facebook
Import Formats	Caffe, Keras, MXNet, ONNX	TensorFlow Graph, PyTorch, ONNX	ONNX PyTorch
Intermediate IR	NNVM / Relay IR	XLA HLO MLIR	Glow Core/Glow IR
Output	LLVM, OpenCL, Metal, CUDA, SPIR-V	LLVM, OpenCL, Vulkan, SPIR-V	LLVM, OpenCL SPIR-V



Logical Architecture Components

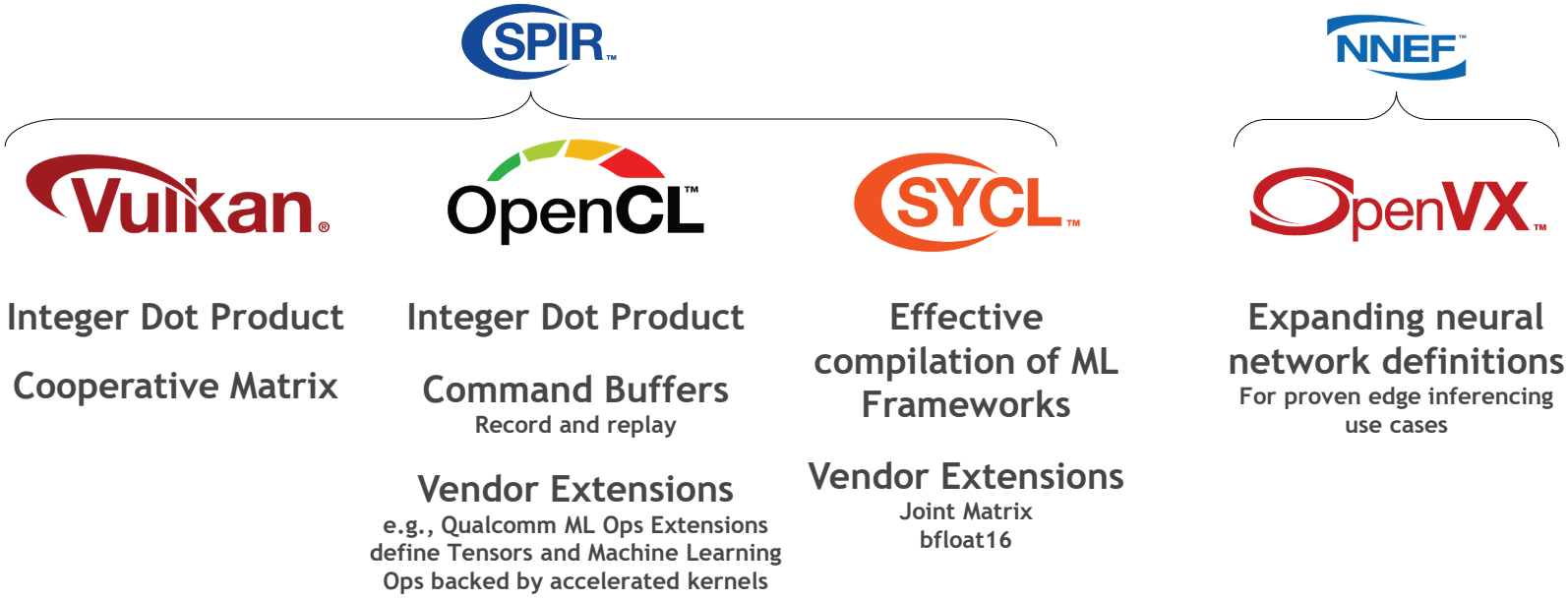


TVM Architecture Diagram

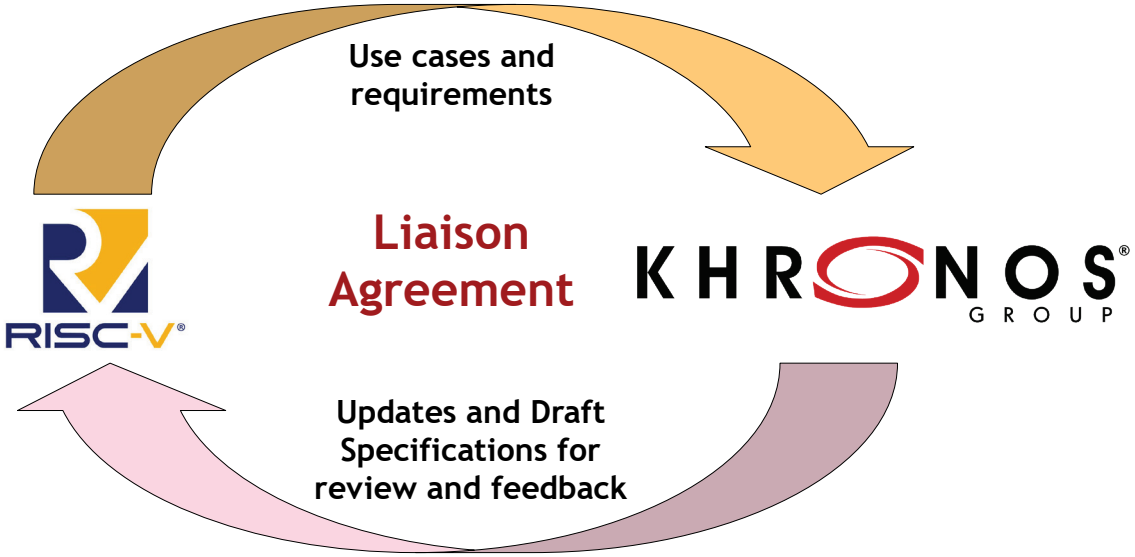
Ongoing Open Standard API Evolution

Khronos working to provide increasingly effective parallel compution acceleration

For example, extension being developed to enhance machine learning are below
RISC-V requirements and use cases can influence API directions and focus!



Possible Khronos and RISC-V Liaison



Khronos welcomes liaison agreements with other industry organizations for effective communication and cooperation

KHRONOS® GROUP

Thank You!

Neil Trevett
@neilt3d

KHRONOS® GROUP