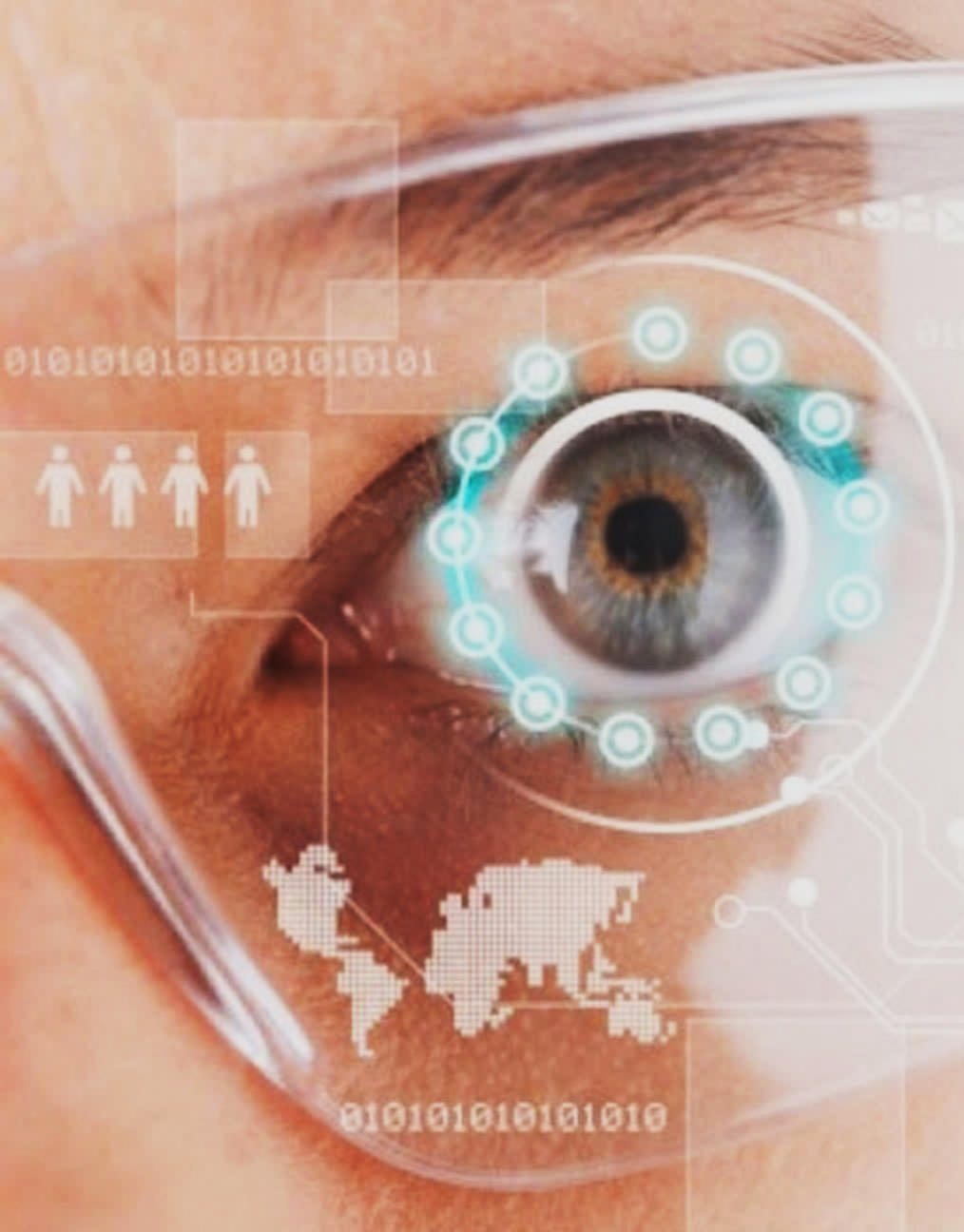


KHRONOS GROUP

Current and Planned Standards for Computer Vision and Machine Learning

Neil Trevett
Khronos President
NVIDIA VP Developer Ecosystems
ntrevett@nvidia.com | @neilt3d

embedded VISION ALLIANCE

KHRONOS GROUP

>150 Members ~ 40% US, 30% Europe, 30% Asia

Khronos is an open, non-profit, member-driven industry consortium developing royalty-free standards to harness the power of silicon acceleration for demanding graphics rendering and computationally intensive applications such as 3D Graphics, Virtual Reality, Augmented Reality, Vision Processing and Machine Learning

Khronos Active Initiatives

3D Graphics
Desktop, Mobile, Web
Embedded and Safety Critical

Vulkan
OpenGL ES
OpenGL
WebGL
EGL
OpenGL SC
Vulkan SC

3D Assets
Authoring
and Delivery

glTF
COLLADA
3DCommerce

Portable XR
Augmented and
Virtual Reality

OpenXR

Parallel Computation
Vision, Inferencing,
Machine Learning

SPIR
SYCL
OpenCL
NNEF
OpenVX

KHRONOS SAFETY CRITICAL ADVISORY FORUM | SC

Guidelines for creating APIs to streamline system safety certification

Exploratory Groups

Heterogeneous Communications between offload compute devices

Making High-Level Languages more effective at acceleration offload

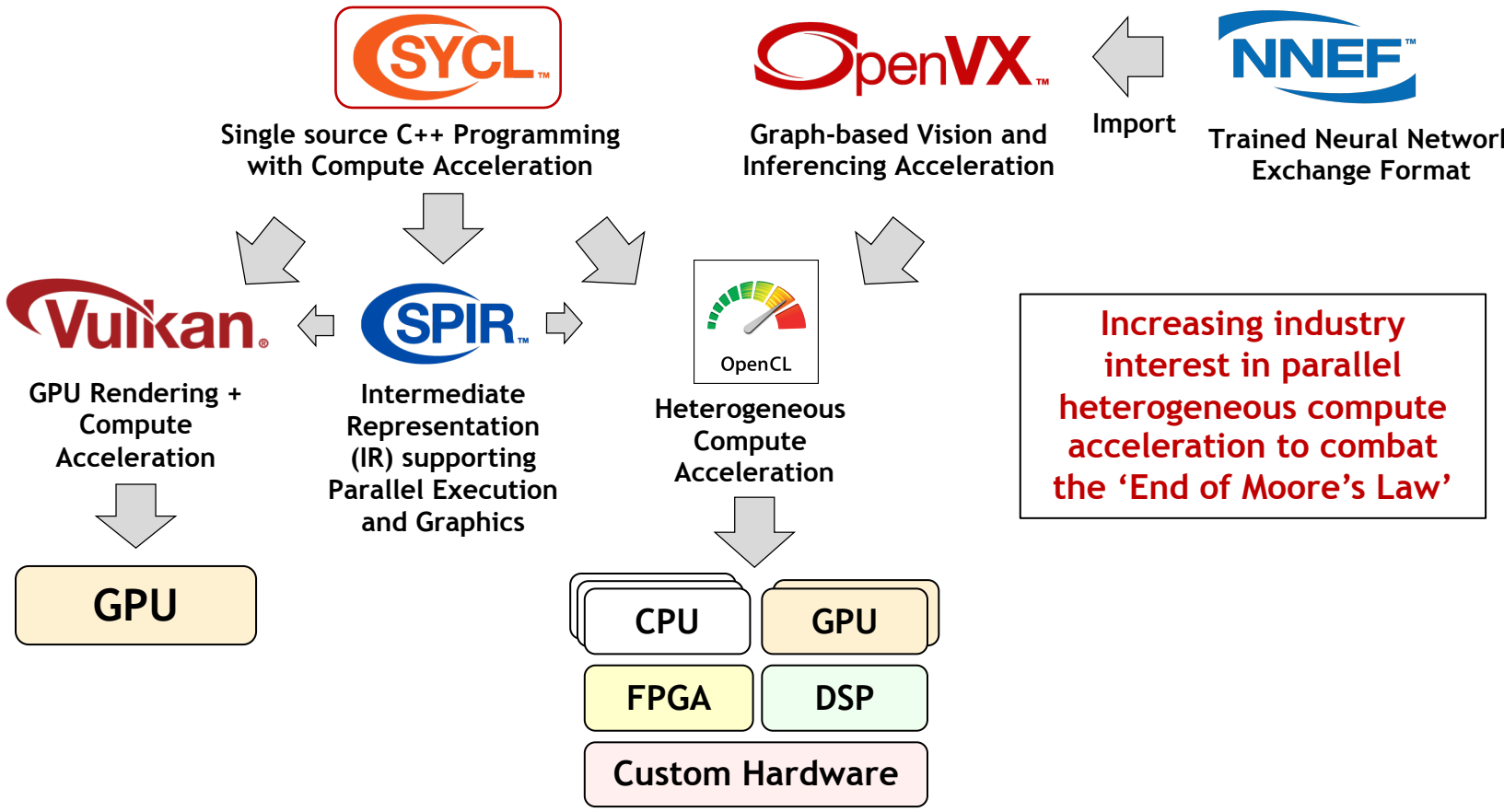
Rendering for scientific visualization and data analytics

KHRONOS GROUP

Khronos Compute Acceleration Standards

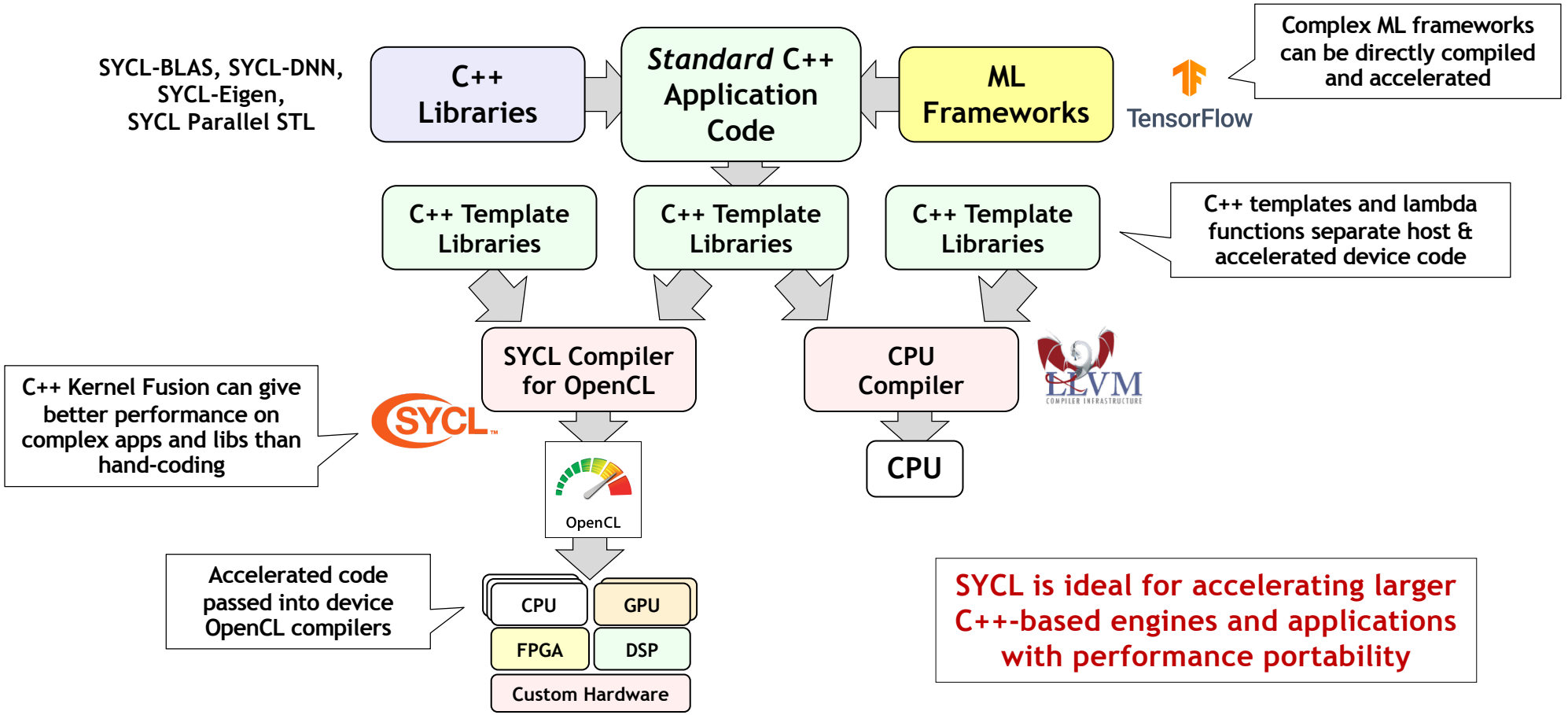
Higher-level APIs
Streamlined programming
and performance portability

Lower-level APIs
Direct Hardware Control



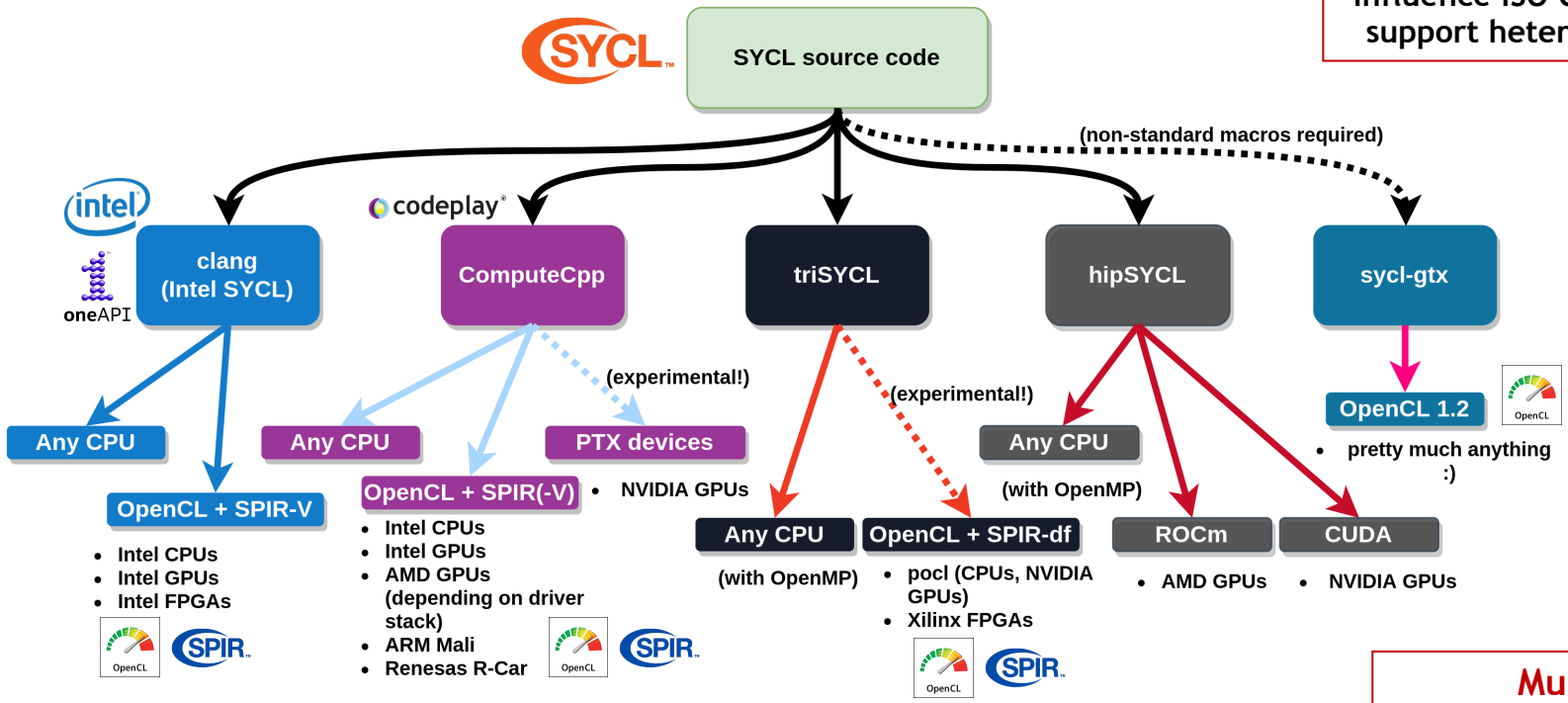
Increasing industry interest in parallel heterogeneous compute acceleration to combat the 'End of Moore's Law'

SYCL Single Source C++ Parallel Programming



SYCL Implementations

SYCL enables Khronos to influence ISO C++ to (eventually) support heterogenous compute



Intel's 'One API'
Based on SYCL using LLVM/clang
https://newsroom.intel.com/news/intels-one-api-project-delivers-unified-programming-model-across-diverse-architectures/#gs.bvdi6z

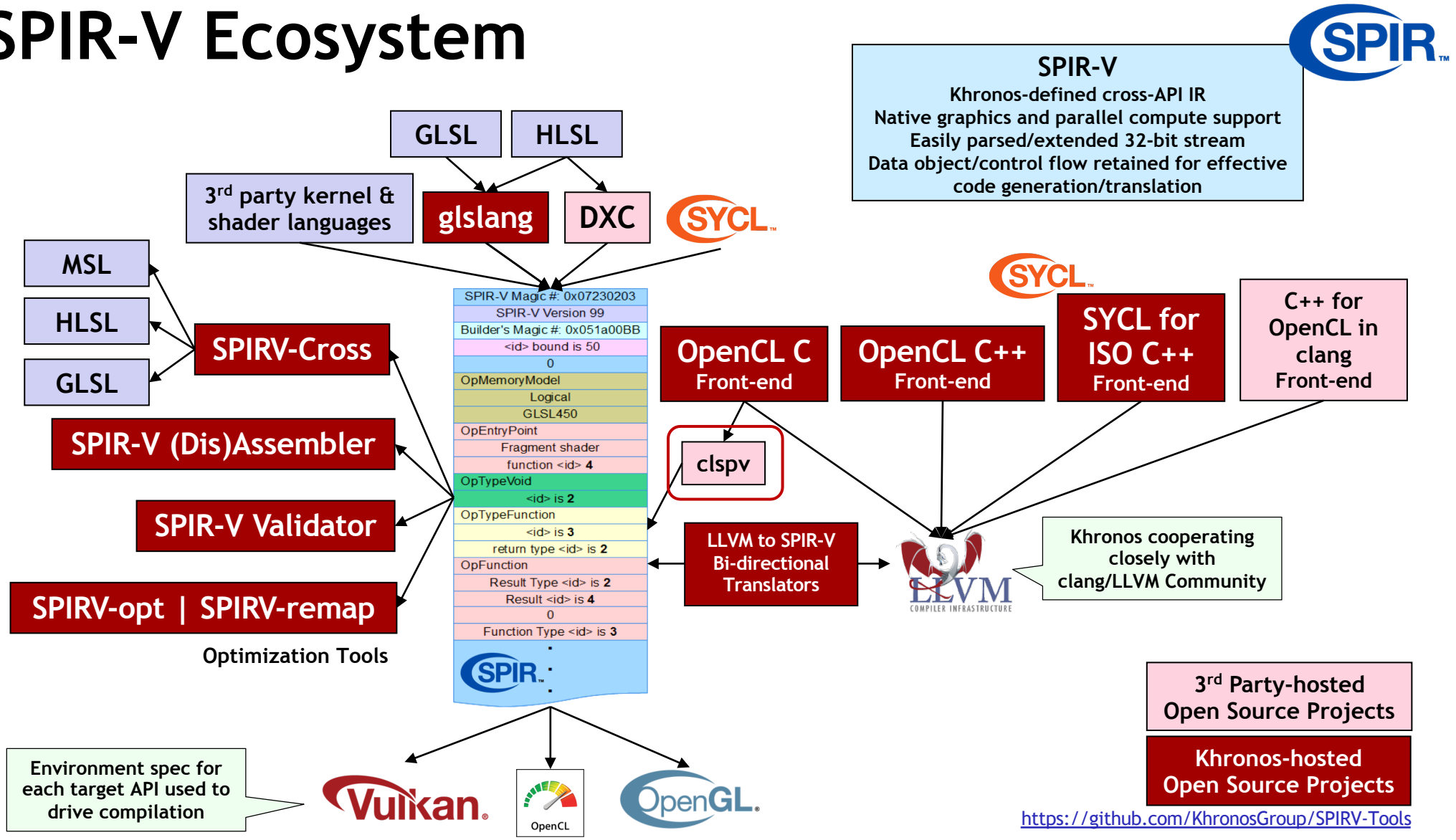
ComputeCpp
Codeplay's SYCL 1.2.1 on multiple hardware

triSYCL
Open-source test-bed

HipSYCL
SYCL 1.2.1 implementation over NVIDIA CUDA and AMD HIP/ROCm

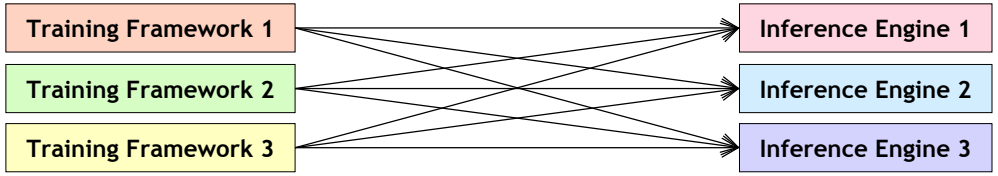
Multiple Backend Support in Development
SYCL beginning to be supported on multiple low-level APIs in addition to OpenCL
e.g. Vulkan, ROCm and CUDA
http://sycl.tech

SPIR-V Ecosystem



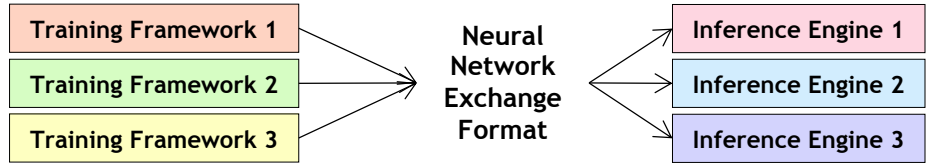
Neural Network Exchange Formats

Before - Training and Inferencing Fragmentation



Every Inferencing Engine needs a custom importer from every Framework

After - NN Training and Inferencing Interoperability



Common Optimization and processing tools

Two Neural Network Exchange Format Initiatives

NNEF	ONNX
Embedded Inferencing Import	Training Interchange
Defined Specification	Open Source Project
Multi-company Governance at Khronos	Initiated by Facebook & Microsoft
Stability for hardware deployment	Software stack flexibility

ONNX and NNEF are Complementary

ONNX moves quickly to track authoring framework updates
NNEF provides a stable bridge from training into edge inferencing engines

NNEF and ONNX Industry Support

NNEF V1.0 released in August 2018

After positive industry feedback on Provisional Specification.
Maintenance update issued in September 2019
Extensions to V1.0 released for expanded functionality

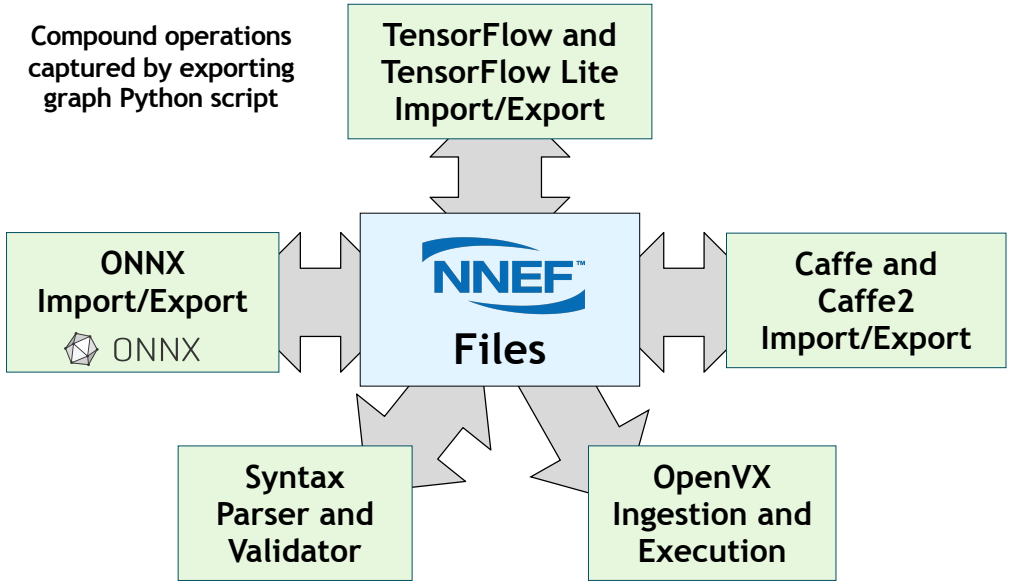
NNEF Working Group Participants

ONNX 1.6 Released in September 2019

Introduced support for Quantization
ONNX Runtime being integrated with GPU inferencing engines such as NVIDIA TensorRT

ONNX Supporters

NNEF Tools Ecosystem



NNEF open source projects hosted on Khronos
NNEF GitHub repository under Apache 2.0
https://github.com/KhronosGroup/NNEF-Tools

NNEF Model Zoo
Now available on GitHub. Useful for checking that ingested NNEF produces acceptable results on target system

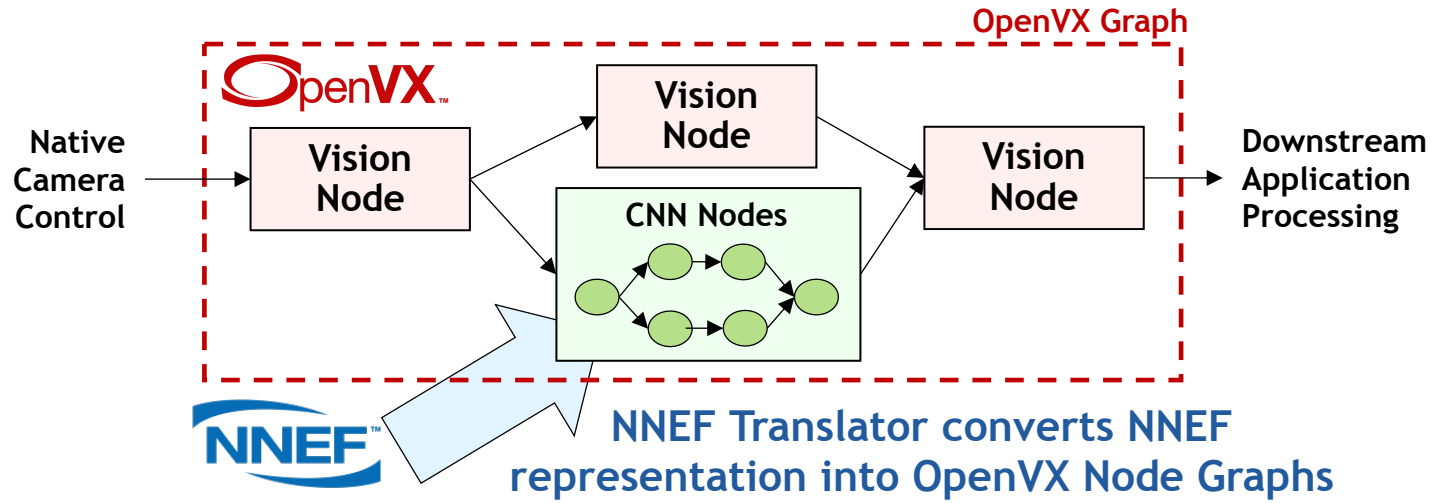
NNEF adopts a rigorous approach to design lifecycle
Especially important for safety-critical or mission-critical applications in automotive, industrial and infrastructure markets

OpenVX Cross-Vendor Inferencing

OpenVX

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

Graph can contain vision processing and NN nodes - enables global optimizations
Optimized OpenVX drivers created and shipped by processor vendors
Implementable on almost any hardware or processor with performance portability
Run-time graph execution need very little host CPU interaction



Performance comparable to hand-optimized, non-portable code

Real, complex applications on real, complex hardware
Much lower development effort than hand-optimized code

AMD cadence ELVEES intel Imagination QUALCOMM NVIDIA Texas Instruments SYNOPSYS socionext VeriSilicon

Hardware Implementations

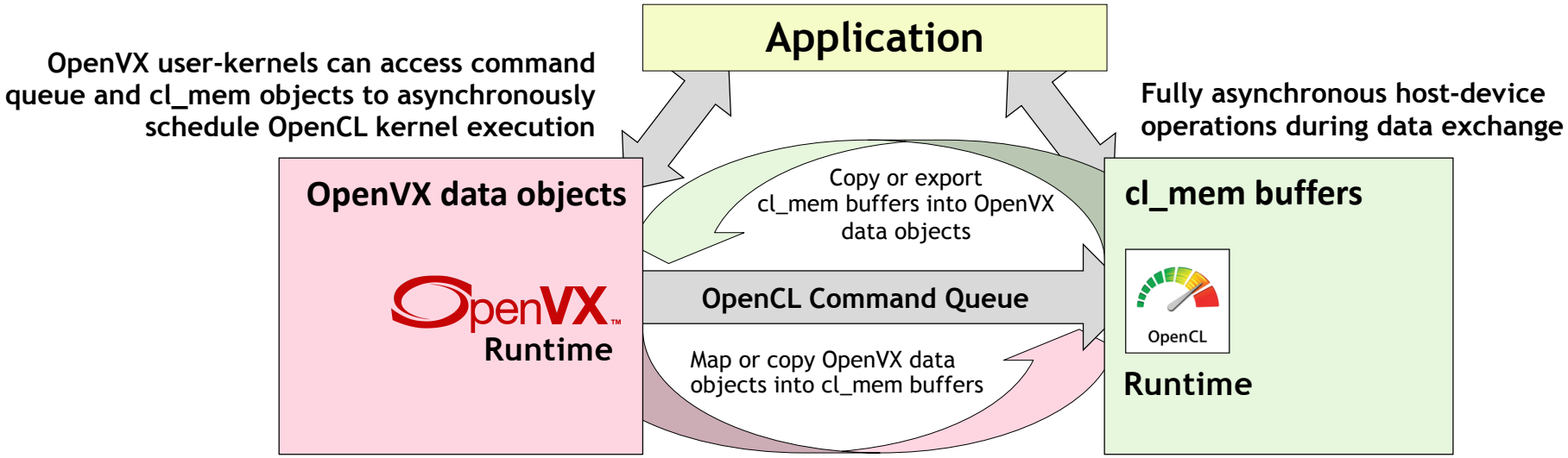
OpenVX Accelerated Custom Nodes

OpenVX/OpenCL Interop

- Provisional Extension
- Enables custom OpenCL acceleration to be invoked from OpenVX User Kernels
- Memory objects can be mapped or copied

Kernel/Graph Import

- Provisional Extension
- Defines container for executable or IR code
- Enables arbitrary code to be inserted as an OpenVX Node in a graph



OpenVX/OpenCL Interop

OpenVX 1.3 Released October 2019

Open Source Conformance Test Suite

https://github.com/KhronosGroup/OpenVX-cts/tree/openvx_1.3

Open Source OpenVX Tutorial and Code Samples

https://github.com/rgiduthuri/openvx_tutorial

Functionality Consolidation into Core 1.3

Neural Net Extension, NNEF Kernel Import, Safety Critical etc.

Deployment Flexibility through Feature Sets

Conformant Implementations ship one or more complete feature sets
Enables market-focused Implementations
- Baseline Graph Infrastructure (enables other Feature Sets)
- Default Vision Functions
- Enhanced Vision Functions (introduced in OpenVX 1.2)
- Neural Network Inferencing (including tensor objects)
- NNEF Kernel import (including tensor objects)
- Binary Images
- Safety Critical (reduced features for easier safety certification)

https://www.khronos.org/registry/OpenVX/specs/1.3/html/OpenVX_Specification_1_3.html

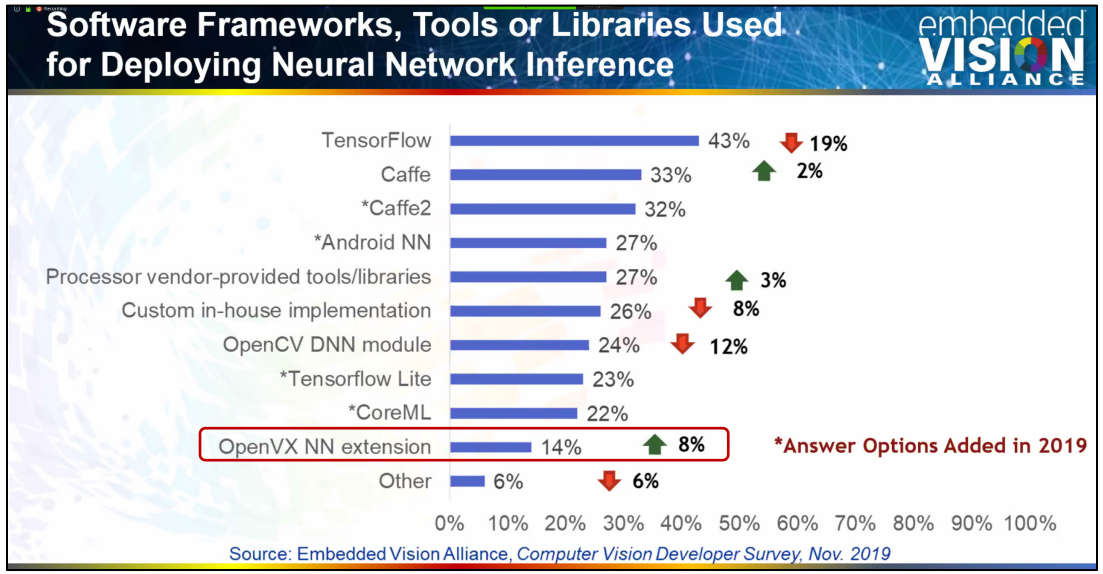
Open Source OpenVX 1.3 for Raspberry Pi

Raspberry Pi 3 Model B with Raspbian OS
Automatic optimization of memory access patterns via tiling and chaining
Highly optimized kernels leveraging multimedia instruction set
Automatic parallelization for multicore CPUs and GPUs
Automatic merging of common kernel sequences
https://github.com/KhronosGroup/OpenVX-sample-impl/tree/openvx_1.3

OpenVX and OpenCV are Complementary

	OpenCV	OpenVX
Implementation	Community driven open source library	Callable API implemented, optimized and shipped by hardware vendors
Scope	100s of imaging and vision functions Multiple camera APIs/interfaces	Tight focus on dozens of core hardware accelerated functions plus extensions and accelerated custom nodes. Uses external camera drivers
Conformance	Extensive OpenCV Test Suite but no formal Adopters program	Implementations must pass Khronos Conformance Test Suite to use trademark
IP Protection	None. Source code licensed under BSD. Some modules require royalties/licensing	Protected under Khronos IP Framework - Khronos members agree not to assert patents against API when used in Conformant implementations
Acceleration	OpenCV 3.0 Transparent API (or T-API) enables function offload to OpenCL devices	Implementation free to use any underlying API such as OpenCL. Can use OpenCL for Custom Nodes
Efficiency	OpenCV 4.0 G-API graph model for some filters, arithmetic/binary operations, and well-defined geometrical transformations	Graph-based execution of all Nodes. Optimizable computation and data transfer
Inferencing	Deep Neural Network module to construct networks from layers for forward pass computations only. Import from ONNX, TensorFlow, Torch, Caffe	Neural Network layers and operations represented directly in the OpenVX Graph. NNEF direct import, ONNX through NNEF convertor

OpenVX Adoption for Inferencing



Source: Embedded Vision Alliance, Computer Vision Developer Survey, November 2019. © 2019 Embedded Vision Alliance

OpenVX™

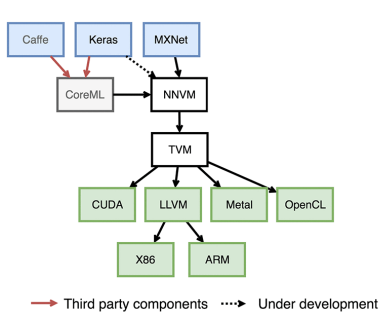
Rise in use of OpenVX for Inferencing

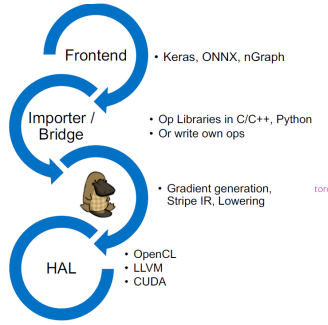
- Availability of NNEF Import and NN Extension
- Opportunity for silicon vendors to optimize inferencing solutions under a standard API
- Increasing deployment on embedded systems (in addition to HPC, Cloud, PCs, Mobile)

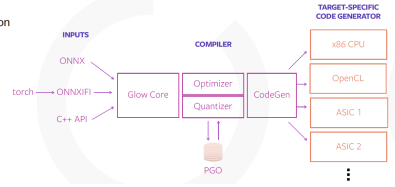
Trend will be accelerated by OpenVX 1.3's deployment flexibility

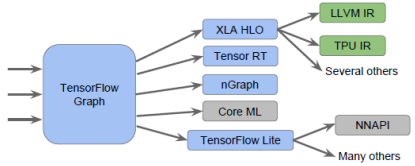
Primary Machine Learning Compilers

	tvm.ai	plaidML	Glow	XLA
	amazon	intel	facebook	Google
Import Formats	Caffe, Keras, MXNet, ONNX	TensorFlow Graph, MXNet, PaddlePaddle, Keras, ONNX	PyTorch, ONNX	TensorFlow Graph, PyTorch
Front-end / IR	NNVM / Relay IR	nGraph / Stripe IR	Glow Core / Glow IR	XLA HLO, MLIR
Output	OpenCL, LLVM, CUDA, Metal	OpenCL, LLVM, CUDA	OpenCL LLVM	LLVM, TPU IR, XLA IR TensorFlow Lite / NNAPI (inc. HW accel)









ML Compiler Steps

Embedded NN Compilers
CEVA Deep Neural Network (CDNN)
Cadence Xtensa Neural Network Compiler (XNNC)

	tvm.ai / amazon	plaidML / intel	Glow / facebook	XLA / Google
Import Formats	Caffe, Keras, MXNet, ONNX	TensorFlow Graph, MXNet, PaddlePaddle, Keras, ONNX	PyTorch, ONNX	TensorFlow Graph, PyTorch
Front-end / IR	NNVM / Relay IR	nGraph / Stripe IR	Glow Core / Glow IR	XLA HLO, MLIR
Output	OpenCL, LLVM, CUDA, Metal	OpenCL, LLVM, CUDA	OpenCL, LLVM	LLVM, TPU IR, XLA IR TensorFlow Lite / NNAPI (inc. HW accel)

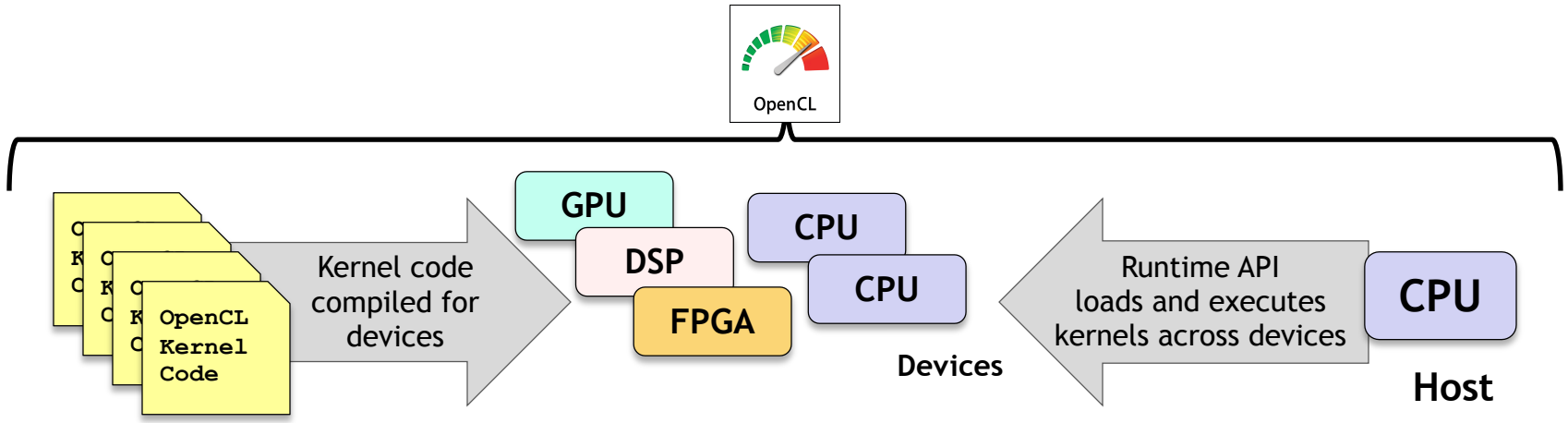
Consistent Steps

1. Import Trained Network Description
2. Apply graph-level optimizations e.g. node fusion, node lowering and memory tiling
3. Decompose to primitive instructions and emit programs for accelerated run-times

Fast progress but still area of intense research
If compiler optimizations are effective - hardware accelerator APIs can stay ‘simple’ and won’t need complex metacommands (combined primitive commands) like DirectML

OpenCL - Low-level Parallel Programing

- **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 is Widely Deployed and Used

Hardware Implementations

Desktop Creative Apps

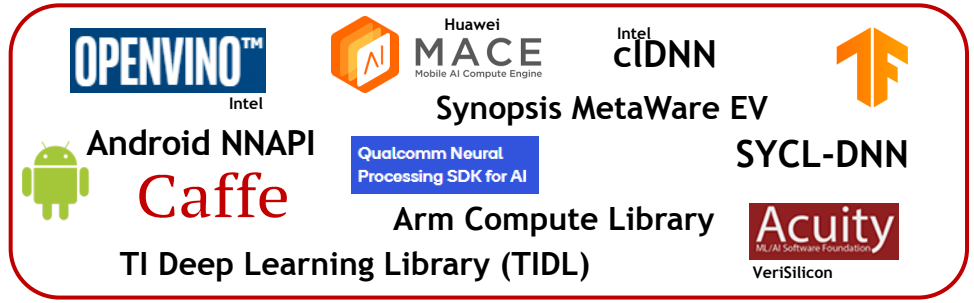
Linear Algebra Libraries

Parallel Computation Languages

Math and Physics Libraries

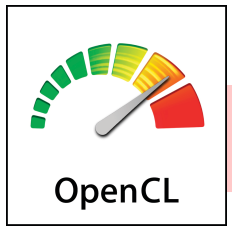
Vision and Imaging Libraries

Machine Learning Inferencing Compilers



Machine Learning Libraries

OpenCL Evolution



May 2017
OpenCL 2.2

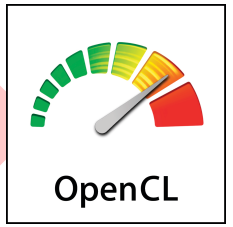
OpenCL Extensions
Scratch-Pad Memory Management for DSPs
Vulkan / OpenCL Interop
Extended Subgroups etc.

Expanding Language Ecosystem
Tighter LLVM integration and cooperation
Open source C++ for OpenCL Kernel Language
SPIR-V 1.4 ingestion for compiler efficiency
SPIR-V Extended debug info

Improving Software Ecosystem
Tool, libraries, ICD Loader

Regular Maintenance Updates
Spec clarifications, formatting and bug fixes
https://www.khronos.org/registry/OpenCL/

Integration of Extensions plus New Core functionality



Target 2020
'OpenCL Next'

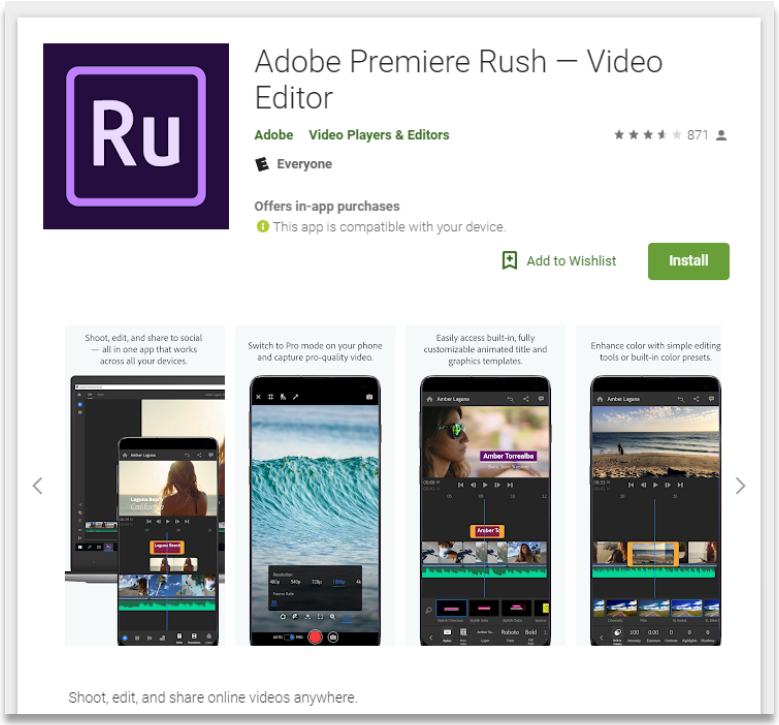
Repeat Cycle for next Core Specification

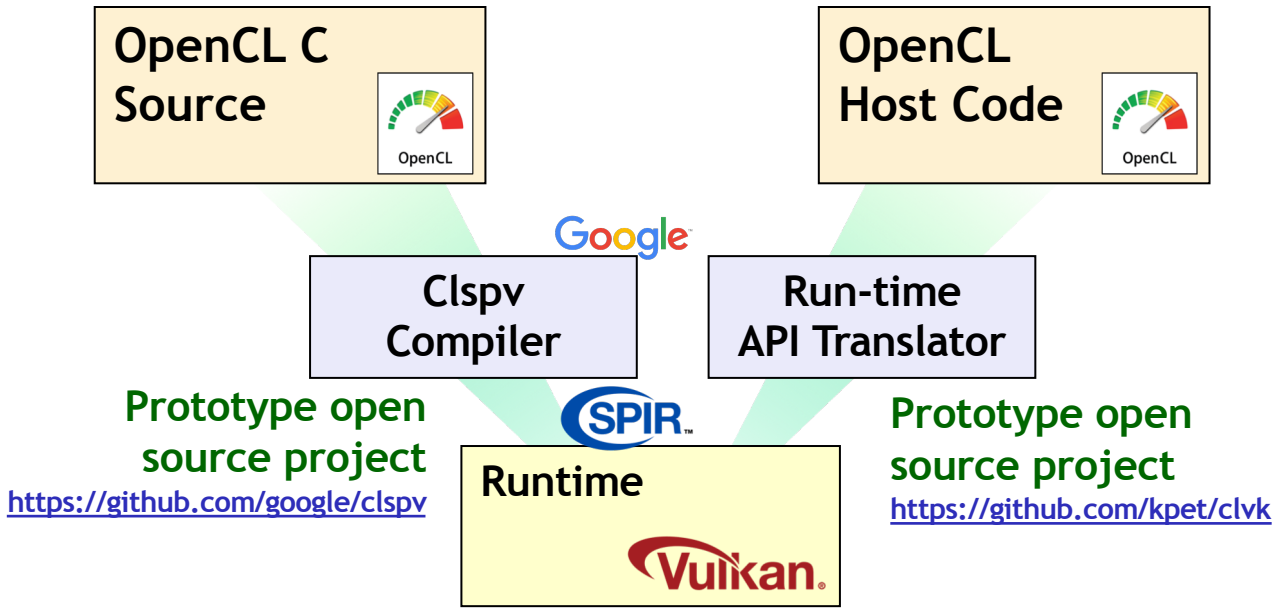
Focus for OpenCL Next is 'Deployment Flexibility'

- Flexible Profile so mobile and embedded vendors can ship customer-targeted functionality and be officially conformant
- Raise the bar for cross-vendor functionality on desktop and in cloud

Deploying OpenCL C Over Vulkan

- **Clspv - Google's experimental compiler for OpenCL C to Vulkan SPIR-V**
  - Open source - tracks top-of-tree LLVM and clang, not a fork
- **Adobe Premiere Rush has 200K lines of OpenCL C kernel code**
  - Professional-quality, cross-platform video capture and editing system
  - Now shipping on Android on Vulkan



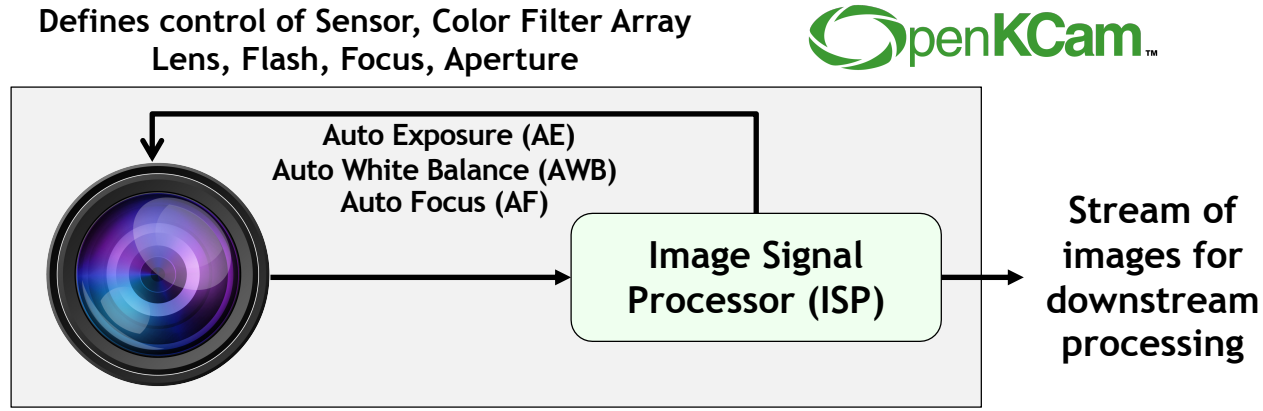


Need for New Camera Control API Standard?

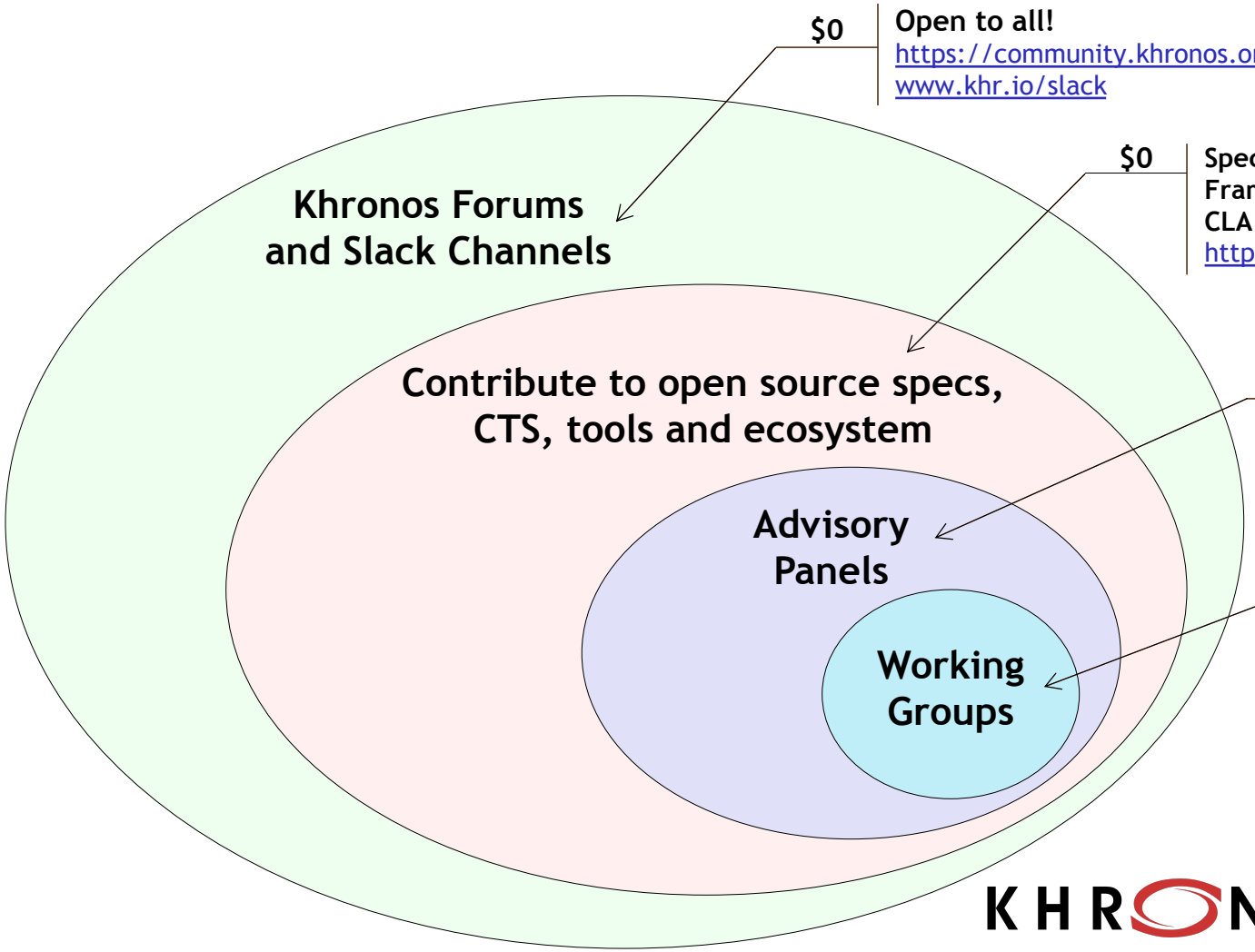
- **Khronos suspended work on OpenKCam standard several years ago**
  - Mobile market went proprietary - but embedded market has different needs
- **OpenKCAM was aiming at advanced control of ISP and camera with cross-platform portability**
  - Generate sophisticated image stream for advanced imaging & vision apps
  - Portable access to growing sensor diversity: e.g. depth sensors and sensor arrays
  - Cross sensor synch: e.g. synch of multiple camera and MEMS sensors
  - Advanced, high-frequency per-frame burst control of camera/sensor: e.g. ROI
  - Multiple input, output re-circulating streams with RAW, Bayer or YUV Processing

Emerging potential liaison opportunity for cooperation – maybe over OpenKCam and GeniCam?
Work together to integrate next generation camera control and acceleration APIs?

Khronos
European Machine Vision Association
MESA / X.org
Embedded Vision Alliance



Khronos Ecosystem Engagement



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

$0 — **Spec fixes and suggestions made under the Khronos IP Framework. Open source contributions under repo's CLA - typically Apache 2.0**
https://github.com/KhronosGroup

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

$ — **Khronos members under Khronos NDA and IP Framework participate and vote in working group meetings. Starts at $3.5K/yr**
https://www.khronos.org/members/

KHRONOS GROUP

Thank You and Resources

- **Khronos is creating cutting-edge royalty-free open standards**
  - For 3D, compute, vision, inferencing acceleration
- **These slides and information on Khronos Standards**
  - www.khronos.org
- **Any company is welcome to join Khronos**
  - https://www.khronos.org/members/
- **Dedicated developer resources**
  - Khronos Developer Forum: https://community.khronos.org/
  - Khronos Developer Slack Channel: www.khr.io/slack
- **We are happy to help answer any questions!**
  - Neil Trevett, Khronos President: ntrevett@nvidia.com, @neilt3d
  - Khronos Developer Relations, Kris Rose: kris@khronos.org, @kristoferrose

KHRONOS GROUP