KHRONOS GROUP

APIs for Accelerating Vision and Inferencing

An Overview of Options and Trade-offs

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

eVe embedded VISION europe

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

KHRONOS® GROUP

Khronos is an open, non-profit, member-driven industry consortium developing royalty-free standards, and vibrant ecosystems, to harness the power of silicon acceleration for demanding graphics rendering and computationally intensive applications such as inferencing and vision processing

Some Khronos Standards Relevant to Embedded Vision and Inferencing

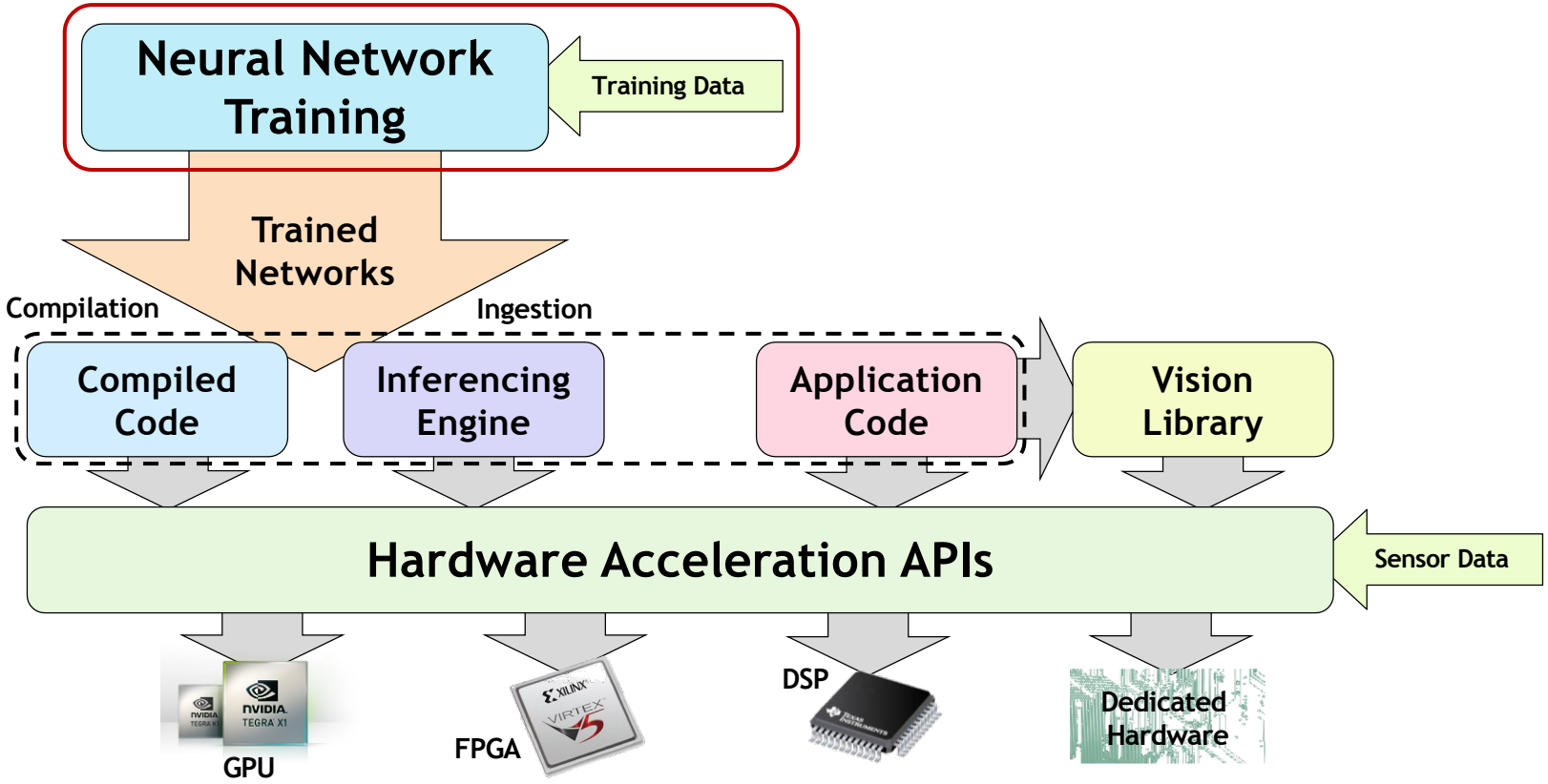
OpenVX	NNEF	OpenCL	SPIR	SYCL	Vulkan
Vision and Inferencing Runtime	Neural Network Exchange Format	Heterogeneous Parallel Compute	Parallel Compute IR	C++ Parallel Compute	GPU Acceleration

Embedded Vision and Inferencing Acceleration

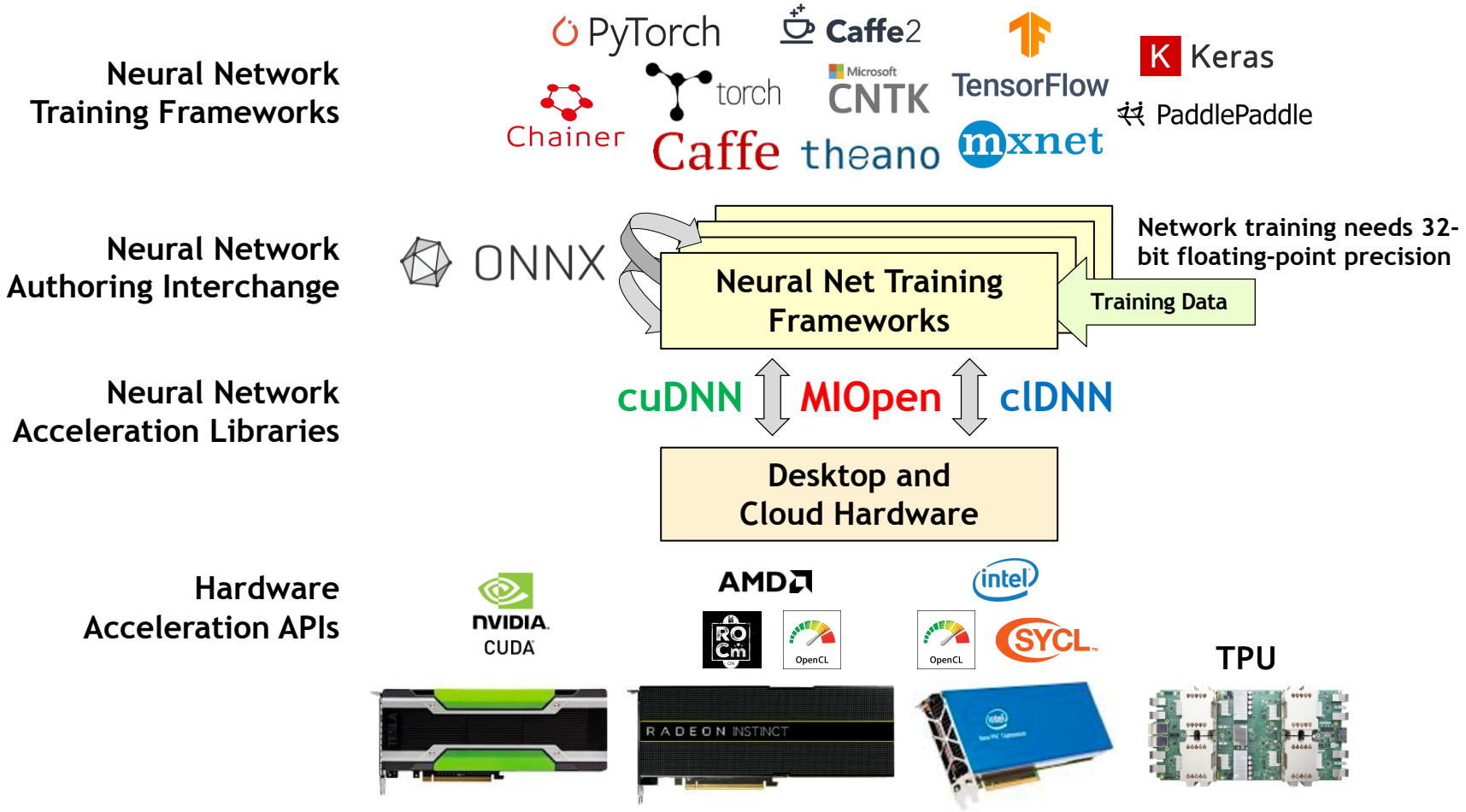
Networks trained on high-end desktop and cloud systems

Apps link to compiled code or inferencing library

Diverse Embedded Hardware (GPUs, DSPs, FPGAs)



Neural Network Training Acceleration

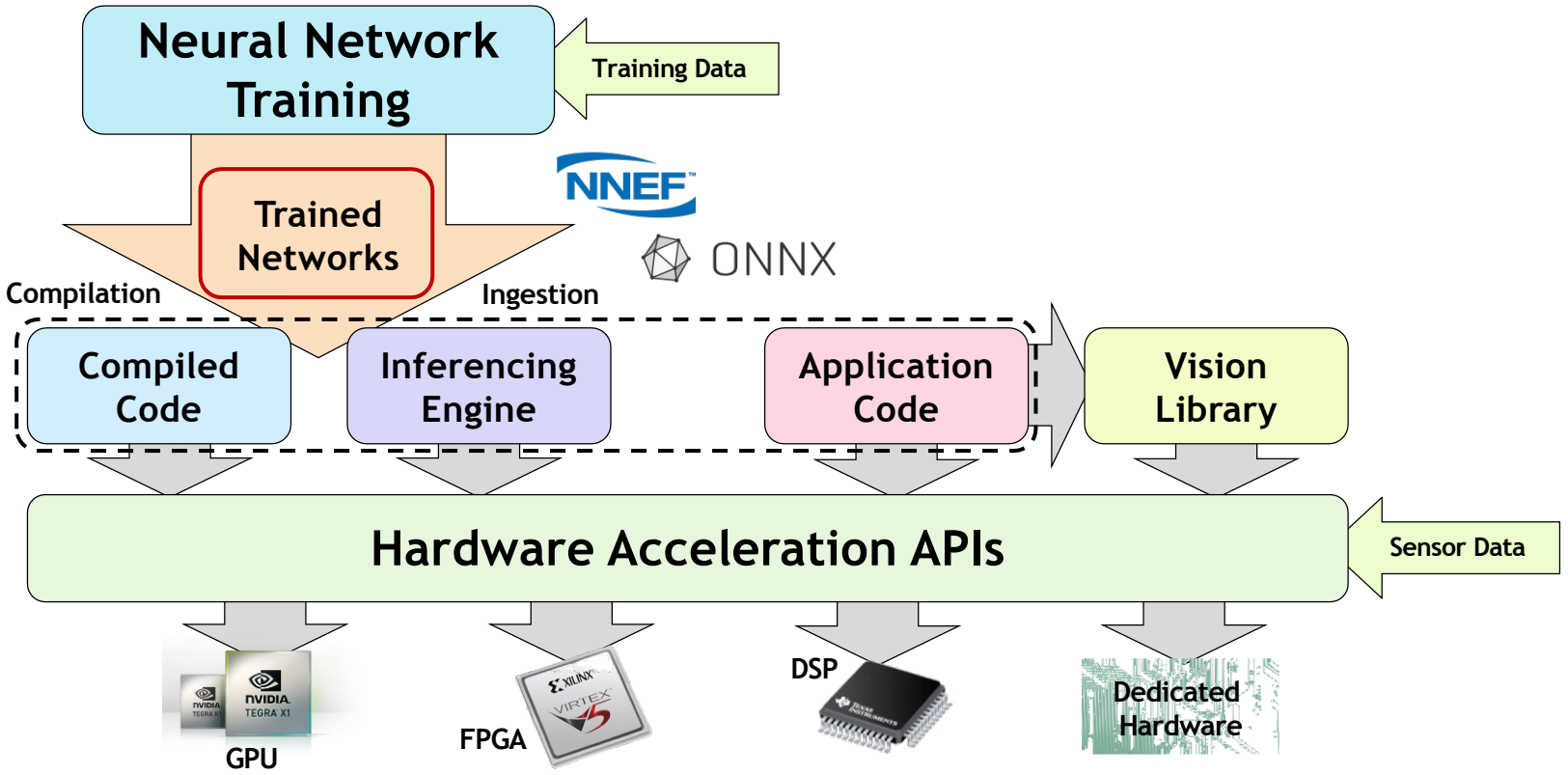


Embedded Vision and Inferencing Acceleration

Networks trained on high-end desktop and cloud systems

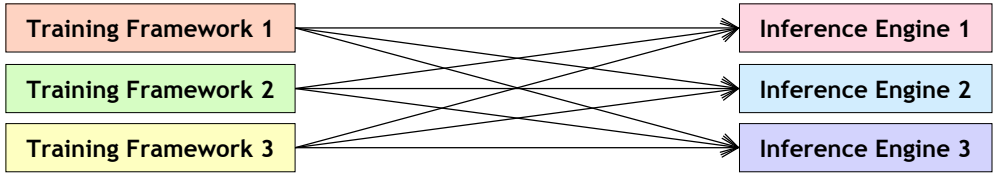
Apps link to compiled code or inferencing library

Diverse Embedded Hardware (GPUs, DSPs, FPGAs)



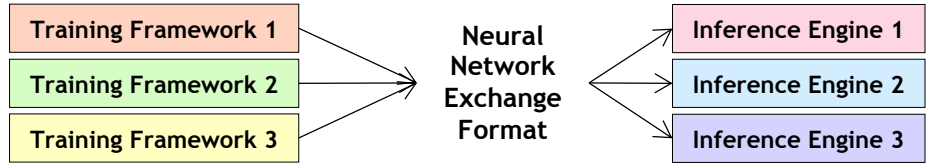
Neural Network Exchange Formats

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

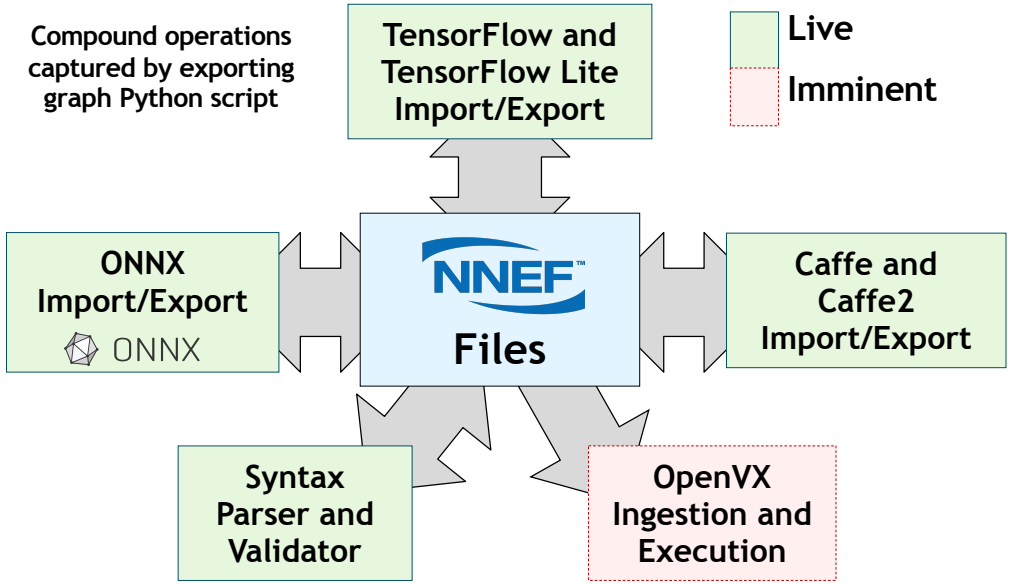
NNEF Working Group Participants

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™

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

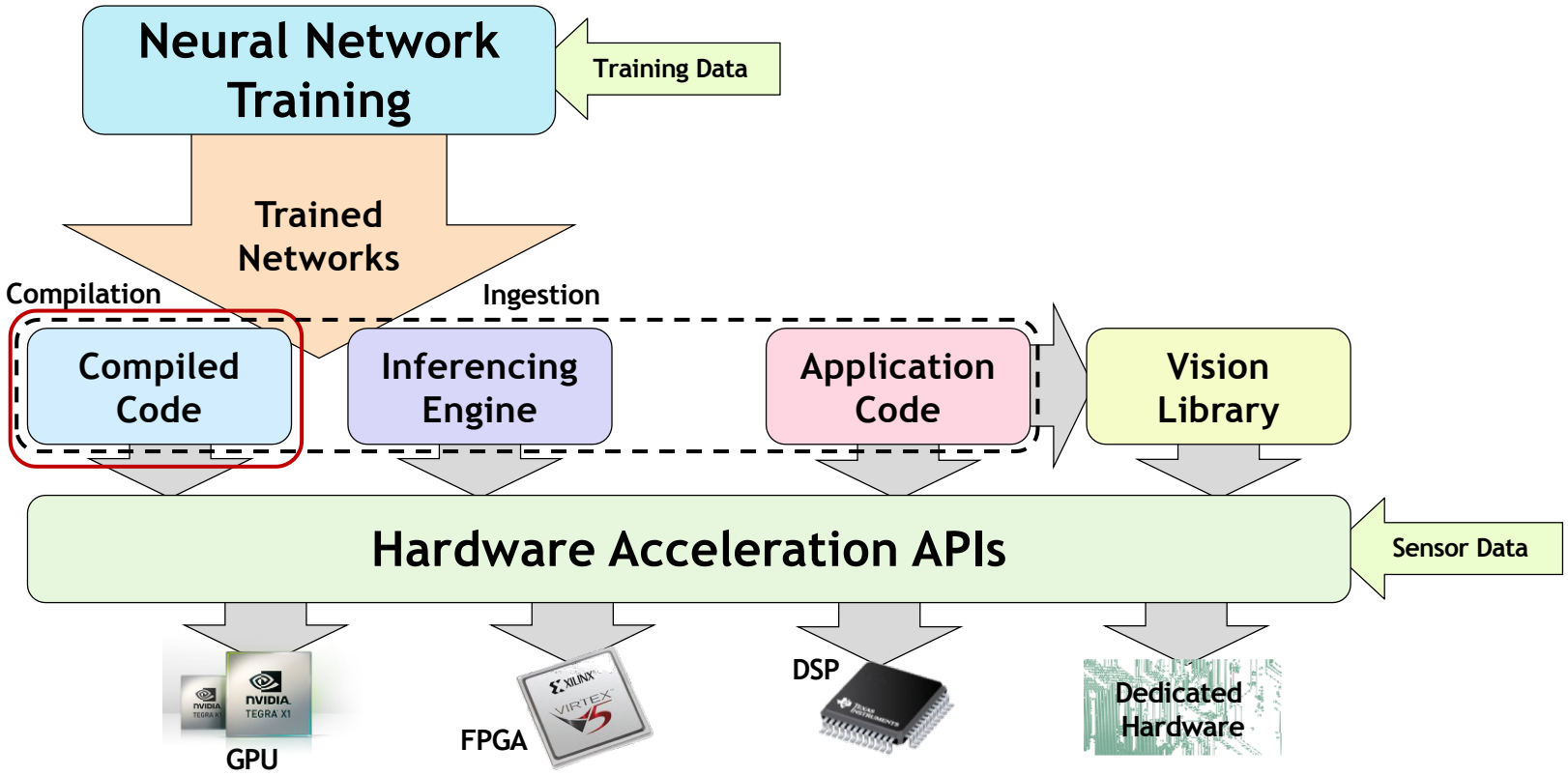
Especially important for safety-critical or mission-critical applications in automotive, industrial and infrastructure markets

Embedded Vision and Inferencing Acceleration

Networks trained on high-end desktop and cloud systems

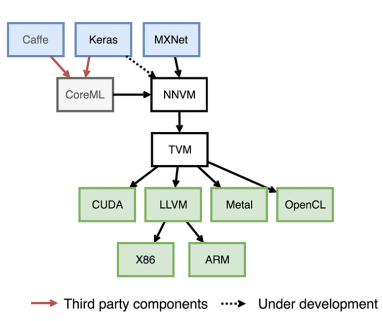
Apps link to compiled code or inferencing library

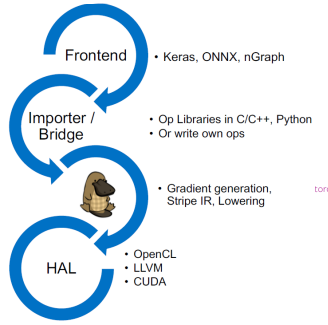
Diverse Embedded Hardware (GPUs, DSPs, FPGAs)

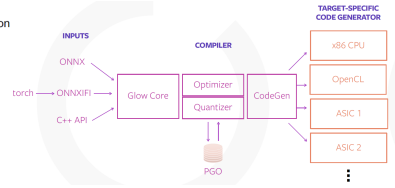


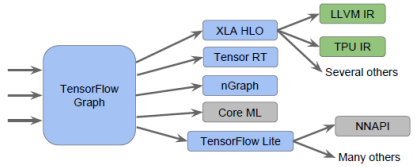
Primary Machine Learning Compilers

	tvm.ai	plaidML	Glow	XLA
	amazon	intel		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

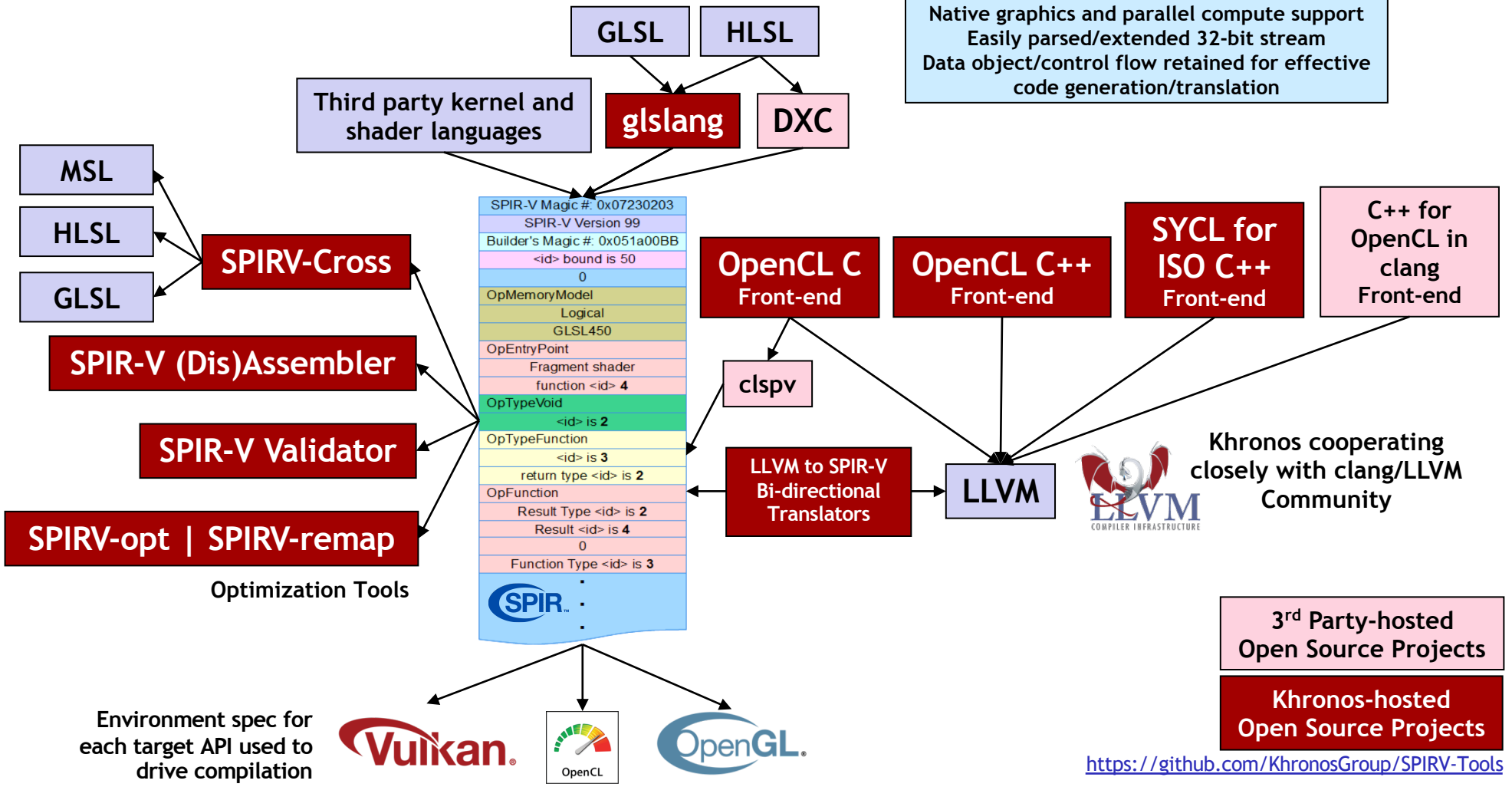
MLIR

Google's Multi-Level IR

Enables multiple domain-specific dialects within a common framework
Experimenting with emitting to Vulkan/SPIR-V

SPIR-V Ecosystem

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



3rd Party-hosted Open Source Projects

Khronos-hosted Open Source Projects

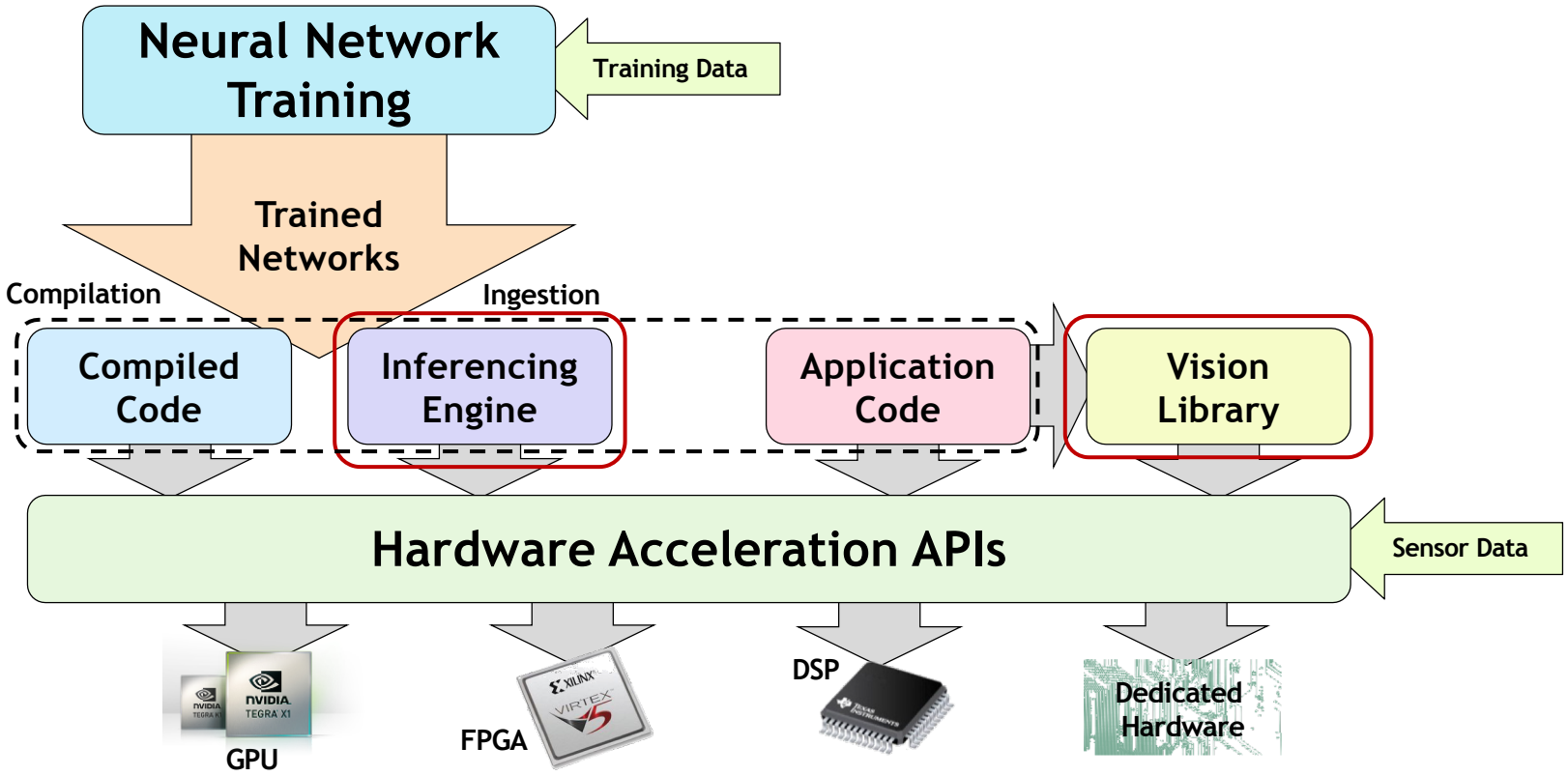
https://github.com/KhronosGroup/SPIRV-Tools

Embedded Vision and Inferencing Acceleration

Networks trained on high-end desktop and cloud systems

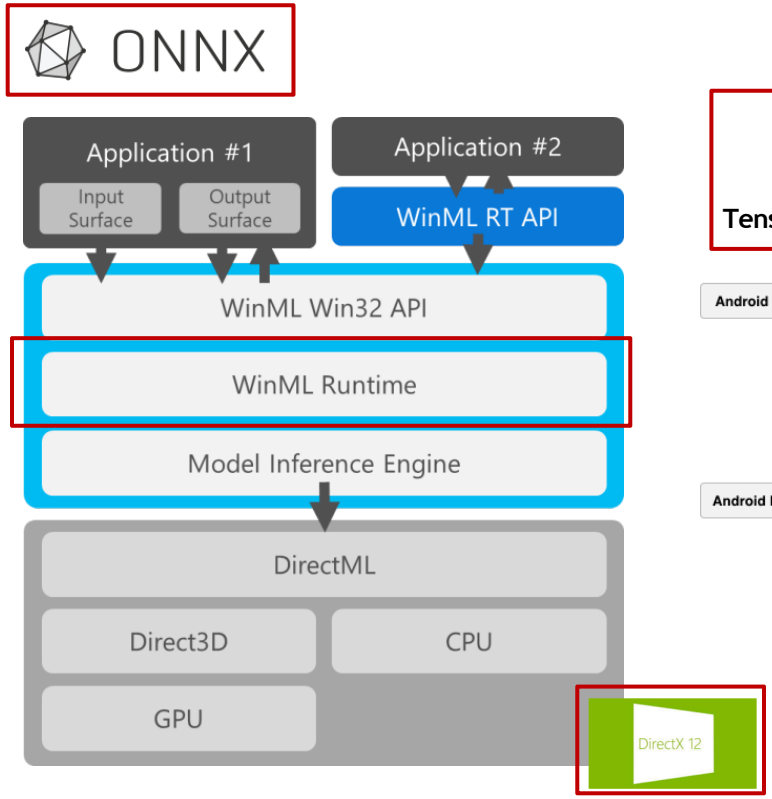
Apps link to compiled code or inferencing library

Diverse Embedded Hardware (GPUs, DSPs, FPGAs)

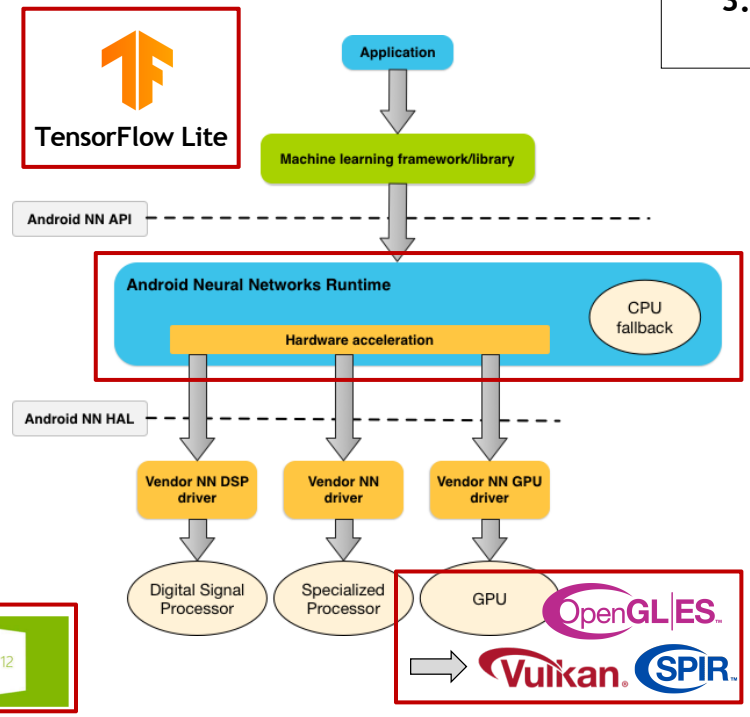


PC/Mobile Platform Inferencing Stacks

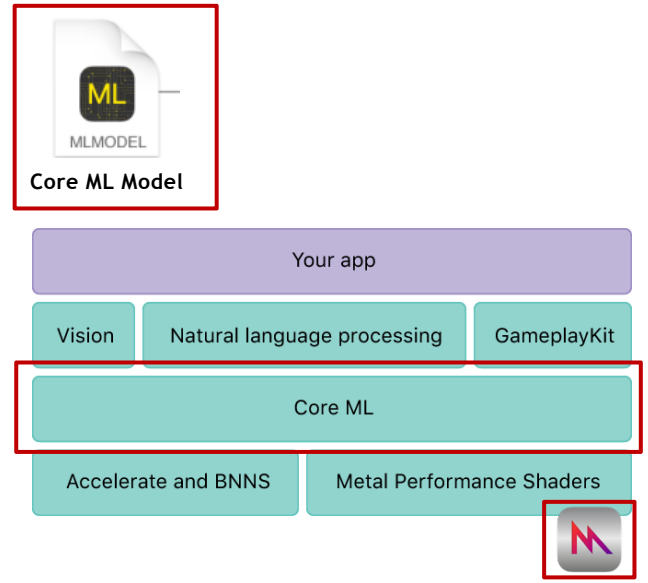
Consistent Steps
1. Import and optimize trained NN model file
2. Ingest graph into ML runtime
3. Accelerate inferencing operations using underlying low-level API



Microsoft Windows
Windows Machine Learning (WinML)
https://docs.microsoft.com/en-us/windows/uwp/machine-learning/



Google Android
Neural Network API (NNAPI)
https://developer.android.com/ndk/guides/neuralnetworks/



Apple MacOS and iOS
CoreML
https://developer.apple.com/documentation/coreml

Inferencing Engines

Platform Engines

Android NNAPI
Microsoft WinML
Apple CoreML

Desktop IHVs

AMD MIVisionX over MIOpen
Intel OpenVINO over clDNN
NVIDIA TensorRT over cuDNN

Acceleration APIs

Vulkan
OpenGL|ES
DirectX 12
ROCm
OpenCL
SYCL
NVIDIA CUDA

Cross-platform Inferencing Engines

OpenCV
OpenVX

Both provide Inferencing AND Vision acceleration

Mobile / Embedded

MACE Mobile AI Compute Engine (Huawei)
Qualcomm Neural Processing SDK for AI
Acuity ML/AI Software Foundation (VeriSilicon)

Arm Compute Library
Huawei MACE
Qualcomm Neural Processing SDK
Synopsis MetaWare EV Dev Toolkit
TI Deep Learning Library (TIDL)
VeriSilicon Acuity

OpenCL

Almost all Embedded Inferencing Engines use OpenCL to access accelerator silicon

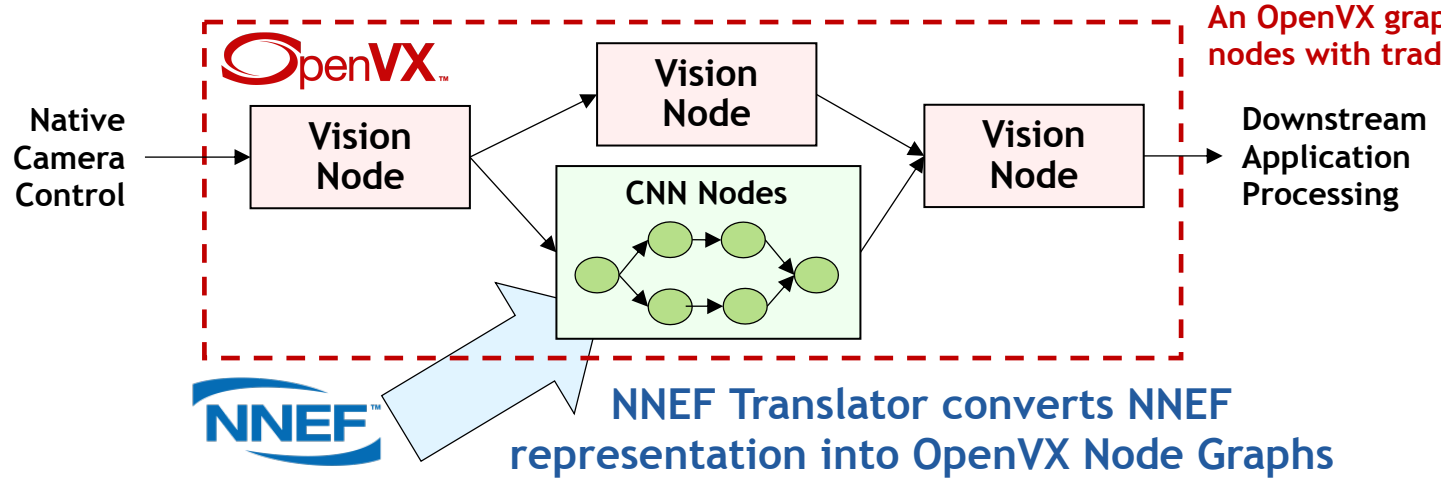
OpenVX and OpenCV are Complementary

		
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 Cross-Vendor Inferencing

OpenVX

A high-level graph-based abstraction for portable, efficient vision processing
Optimized OpenVX drivers created and shipped by processor vendors
Can be implemented on almost any hardware or processor
Graph can contain vision processing and NN nodes - enables global optimizations
Run-time graph execution can be almost completely autonomously from the host CPU



An OpenVX graph mixing CNN nodes with traditional vision nodes

Performance comparable to hand-optimized, non-portable code

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

Hardware Implementations

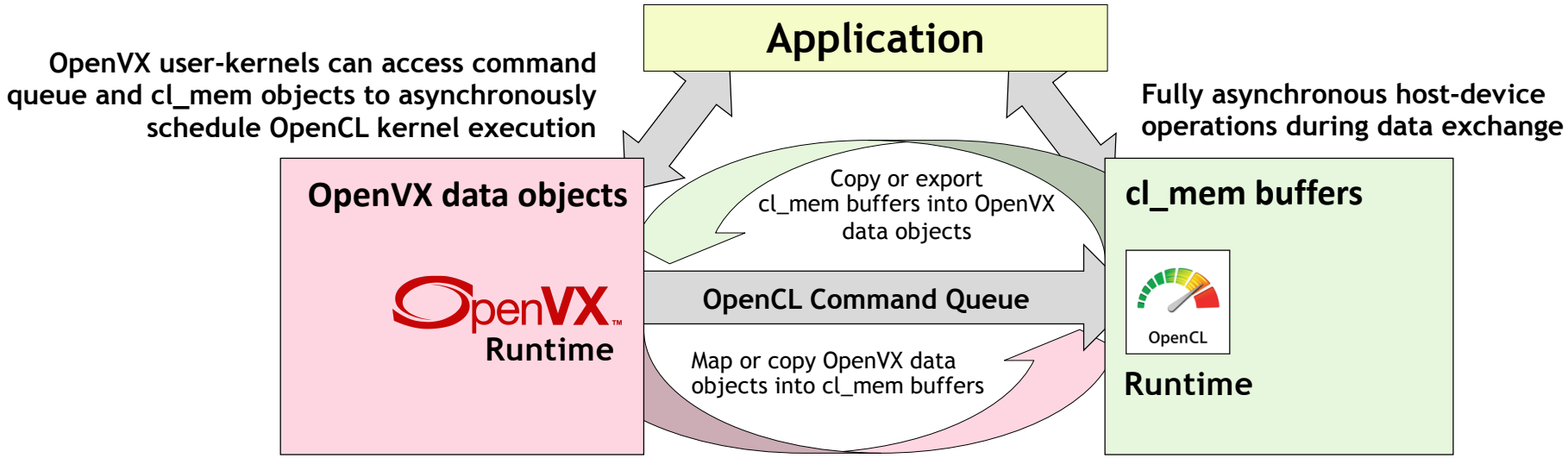
Extending OpenVX with Custom Nodes

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 a OpenVX Node in a graph



OpenVX/OpenCL Interop

OpenVX 1.3 Announced This Week!

OpenVX 1.3 Feature Sets

Enables deployment flexibility while avoiding fragmentation
Implementations with one or more complete feature sets are conformant
- 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 Prototype OpenVX 1.3 Conformance Test Suite

Finalization expected before the end of 2019
https://github.com/KhronosGroup/OpenVX-cts/tree/openvx_1.3

Open Source OpenVX Tutorial and Code Samples

https://github.com/rgiduthuri/openvx_tutorial

Open source OpenVX 1.3 for Raspberry Pi

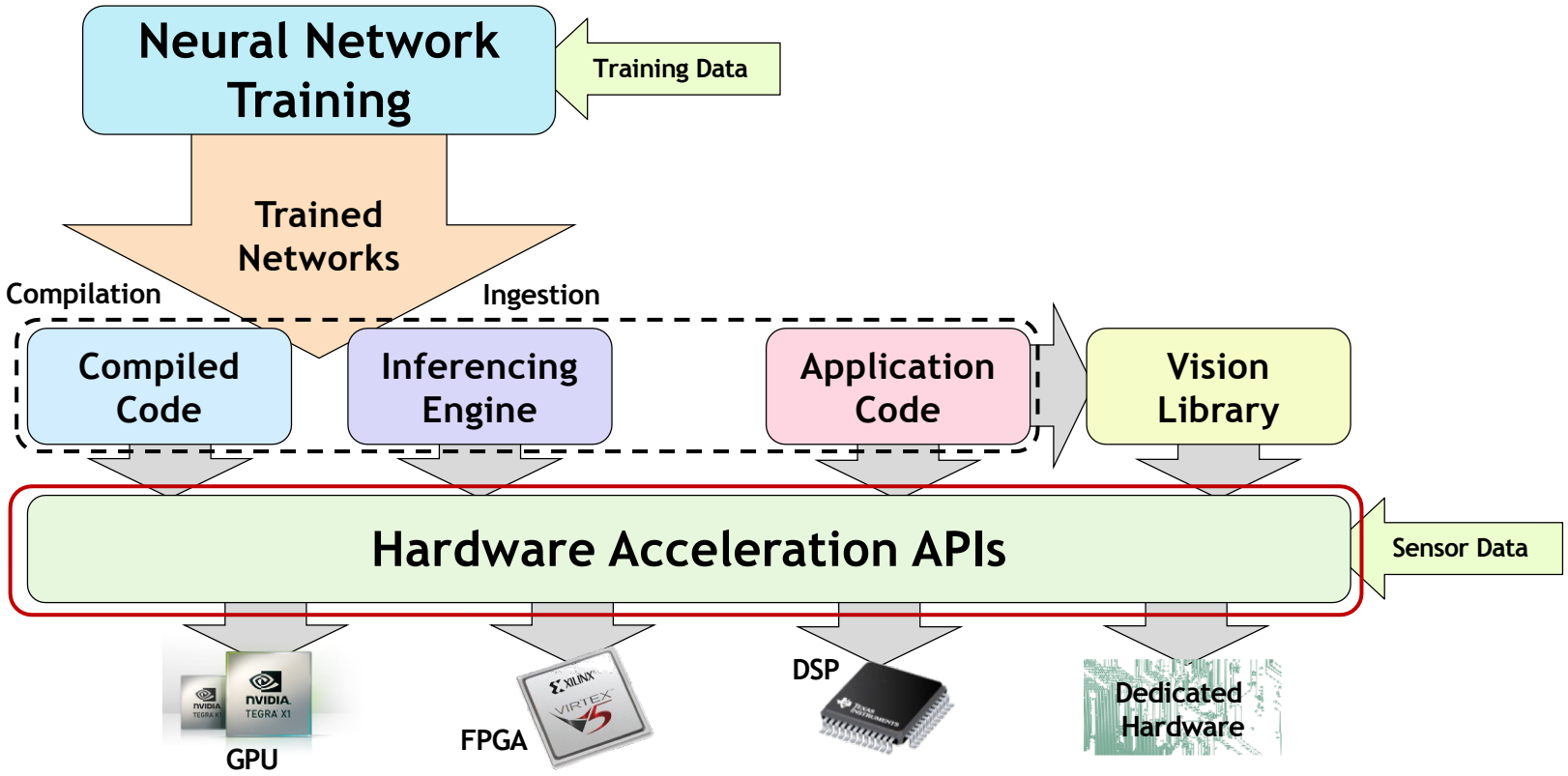
Raspberry Pi 3 Model B with Raspbian OS
Automatic optimization of memory access patterns via tiling and chaining
Highly optimized kernels leveraging multimedia instruction set
Automatic parallelization for multicore CPUs and GPUs
Automatic merging of common kernel sequences
https://github.com/KhronosGroup/OpenVX-sample-impl/tree/openvx_1.3

Embedded Vision and Inferencing Acceleration

Networks trained on high-end desktop and cloud systems

Apps link to compiled code or inferencing library

Diverse Embedded Hardware (GPUs, DSPs, FPGAs)



GPU and Heterogeneous Acceleration APIs

Only Vulkan is cross platform API with rendering - including Embedded Linux and Android

GPU APIs

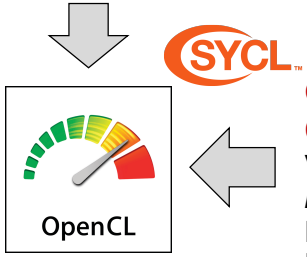
Application

GPU Rendering APIs gradually expanding compute capabilities e.g. short data types

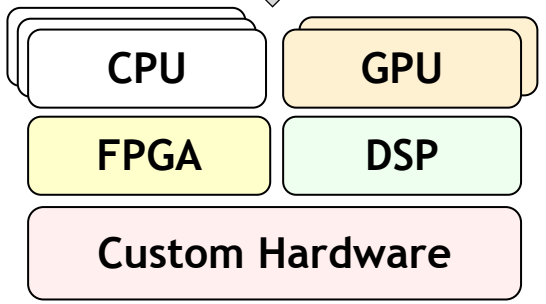
GPU

Heterogenous APIs

Application



Growing number of optimized OpenCL libraries
Vision and Imaging
Machine Learning and Inferencing
Linear Algebra and Mathematics
Physics



Heterogeneous Compute Resources

Many mobile SOCs and Embedded Systems becoming increasingly heterogeneous
Autonomous vehicles
Vision and inferencing

OpenCL provides a programming and runtime framework
OpenCL C and C++ kernels can be compiled and loaded to any available device
Low-level control over memory allocation and parallel task execution
Simpler, relatively lightweight and more flexible than GPU APIs

OpenCL is Widely Deployed and Used

Hardware Implementations

Desktop Creative Apps

CLBlast ViennaCL SYCL-BLAS

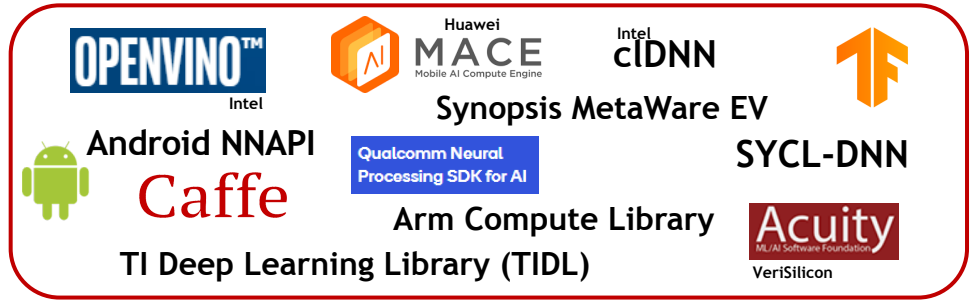
Linear Algebra Libraries

Parallel Computation Languages

Math and Physics Libraries

Vision and Imaging Libraries

Machine Learning Inferencing Compilers

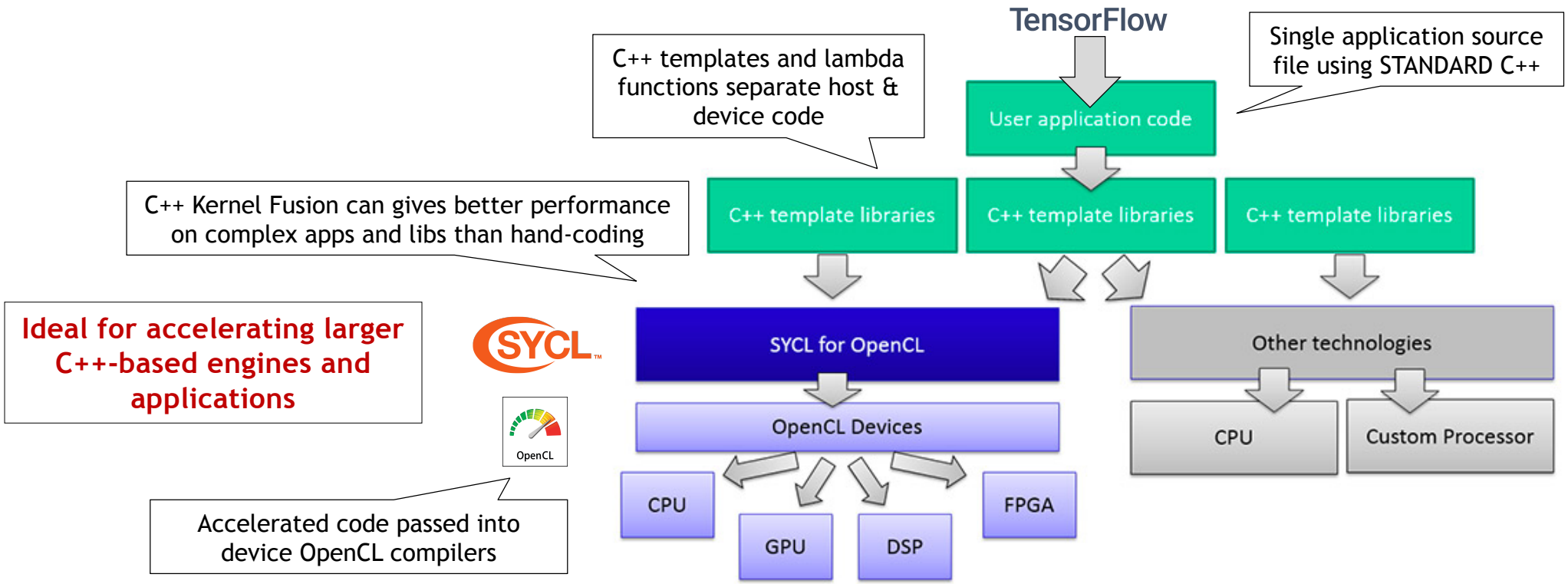


Machine Learning Libraries

SYCL Single Source C++ Parallel Programming

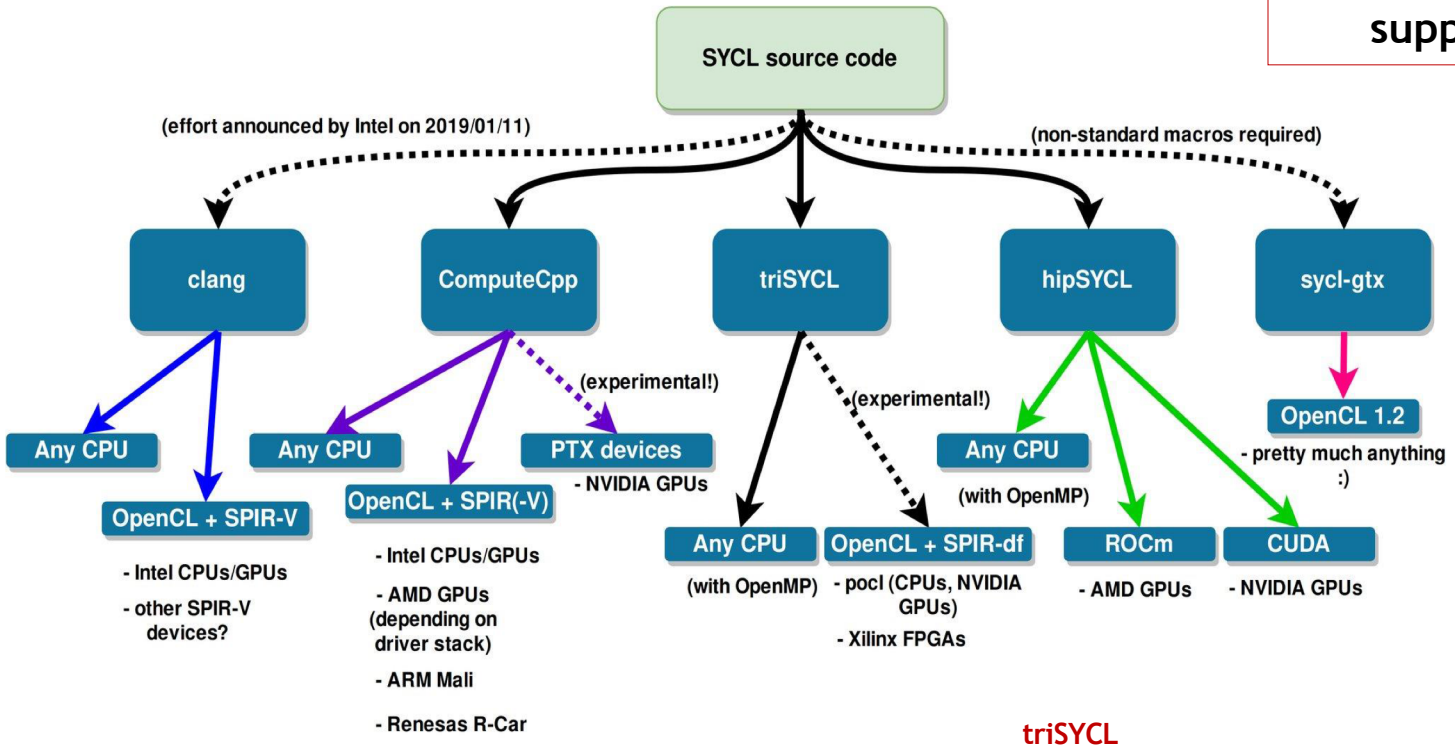
- **SYCL 1.2.1 Adopters Program released in July 2018**
  - https://www.khronos.org/news/press/khronos-releases-conformance-test-suite-for-sycl-1.2.1
- **Multiple SYCL libraries for vision and inferencing**
  - SYCL-BLAS, SYCL-DNN, SYCL-Eigen, SYCL Parallel STL

E.g. complex ML frameworks can be directly compiled and accelerated



SYCL Implementations

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



Multiple Backend Support Coming

SYCL beginning to be supported on low-level APIs in addition to OpenCL e.g. Vulkan and CUDA

http://sycl.tech

Intel Adoption

Intel's 'One API' Initiative uses SYCL

https://newsroom.intel.com/news/intels-one-api-project-delivers-unified-programming-model-across-diverse-architectures/#gs.bydj6z

LLVM/clang SYCL Compiler

Compiles C++-based SYCL source files into code for both CPU and a wide range of compute accelerators

ComputeCpp

Codeplay Software's v1.2.1 conformant implementation available to download today

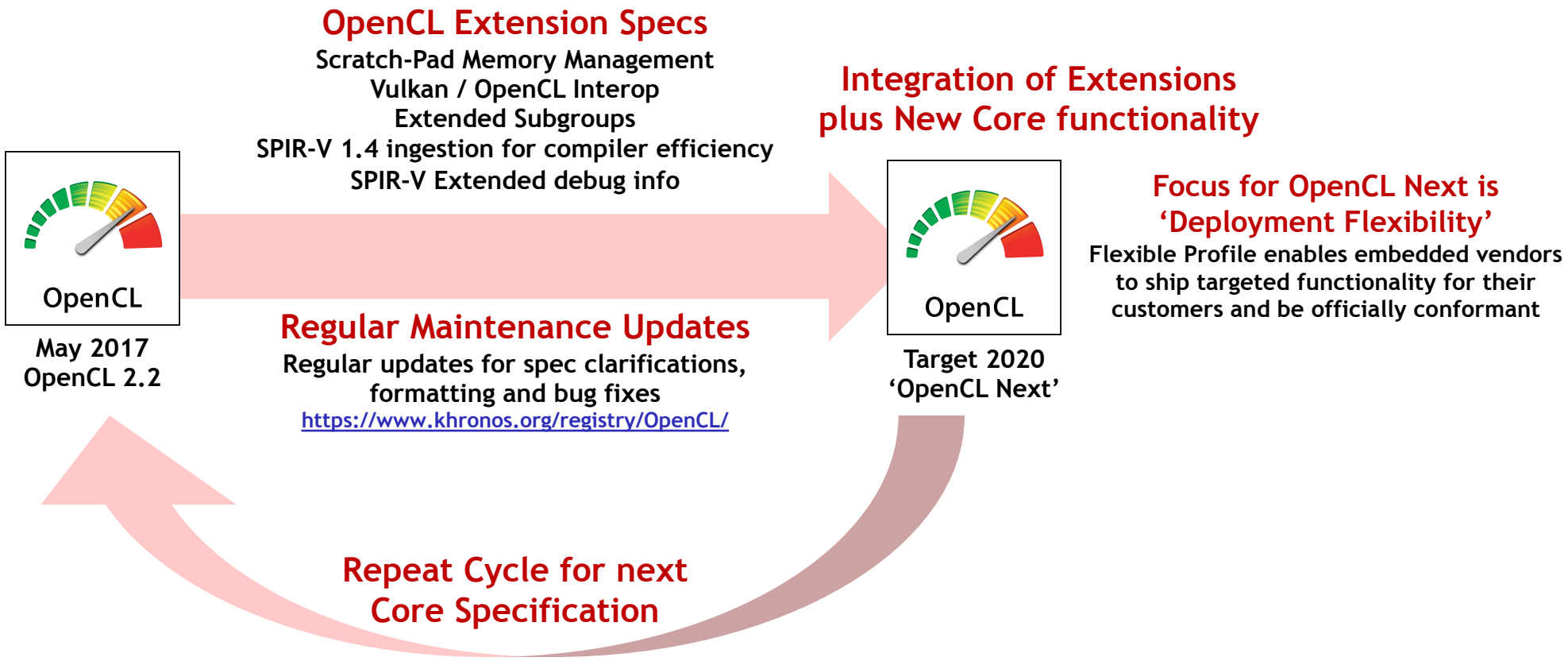
triSYCL

Open-source test-bed to experiment with the specification of the OpenCL SYCL C++ layer and to give feedback to Khronos

HipSYCL

SYCL 1.2.1 implementation that builds upon NVIDIA CUDA/AMD HIP/ROCm

OpenCL Evolution



Pervasive 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

Android
(Android 7.0+)
(Vulkan 1.1 required on Android Q)

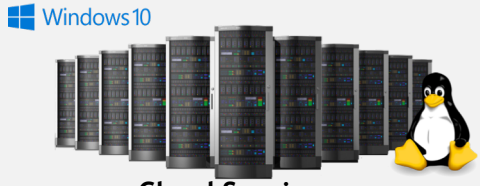
iOS macOS

Apple
(via porting layers)

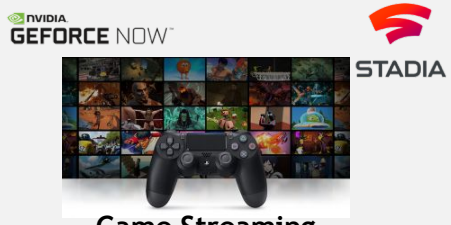
Media Players

Consoles

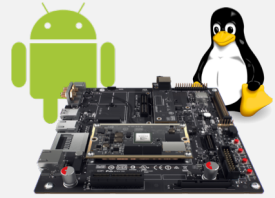
Virtual Reality



Cloud Services



Game Streaming



Embedded

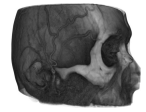
Game Engines

EPIC GAMES id CRYENGINE unity VALVE

Croteam Serious Engine

XENKO NetEase Games Passion of gamers

Safety Critical GPU API Evolution



OpenGL SC 1.0 - 2005
Fixed function graphics safety critical subset

OpenGL SC 2.0 - April 2016
Programmable Shaders safety critical subset

New Generation Safety Critical APIs for Graphics, Compute and Display

OpenGL ES 1.0 - 2003
Fixed function graphics

OpenGL ES 2.0 - 2007
Programmable Shaders

Vulkan 1.0 - 2016
Explicit Graphics and Compute

OpenCL

Potential OpenCL SC work will leverage the deployment flexibility of 'OpenCL Next' to minimize API surface area

Rendering Compute Display

Industry Need
for GPU Acceleration APIs designed to ease system safety certification is increasing
ISO 26262 / ASIL-D

ISO 26262

Vulkan is Compelling Starting Point for SC GPU API Design

- Widely adopted, royalty-free open standard
- Low-level explicit API - smaller surface area than OpenGL
- Not burdened by debug functionality
- Very little internal state
- Well-defined thread behavior

Vulkan

Clearly Definable Design Goals to Adapt Vulkan for SC

Reduce driver size and complexity
-> Offline pipeline creation, no dynamic display resolutions

Deterministic Behavior
-> No ignored parameters, static memory management, eliminate undefined behaviors

Robust Error Handling
-> Error callbacks so app can respond, Fatal error callbacks for fast recovery initiation

C API - MISRA C Compliance

Khronos Vulkan SC Working Group started work in February 2019

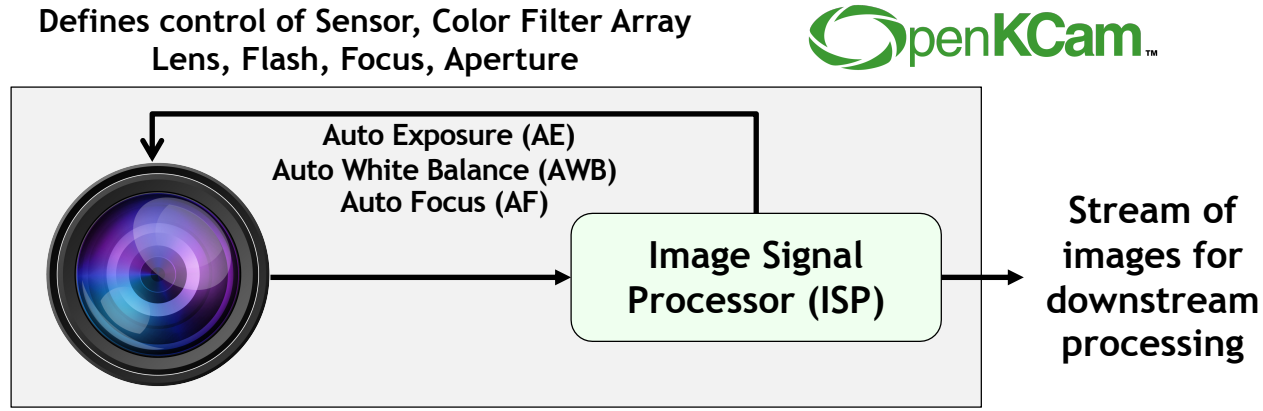
Need for New Camera Control API Standard?

- **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

Liaison opportunity for cooperation - maybe over OpenKCam and GeniCam?
Work together to integrate next generation camera control and acceleration APIs?

emva
european machine vision association

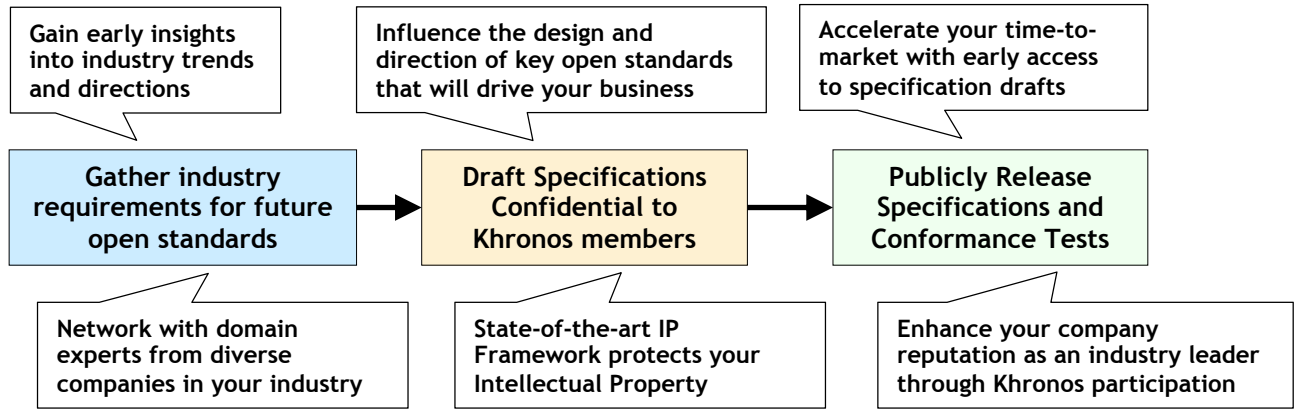
KHRONOS GROUP



Thank You and Resources

- **Khronos Standards: OpenVX, NNEF, OpenCL, SPIR, SYCL, Vulkan and more…**
  - Any company is welcome to join Khronos www.khronos.org
  - OpenVX 1.3: www.khronos.org/openvx/
  - SYCL: http://sycl.tech
- **Compilers for Machine Learning Conference Proceedings:** www.c4ml.org
- **MLIR:** www.blog.google/technology/ai/mlir-accelerating-ai-open-source-infrastructure/
- **Neil Trevett:** ntrevett@nvidia.com | @neilt3d

KHRONOS GROUP



Benefits of Khronos membership

KHRONOS GROUP