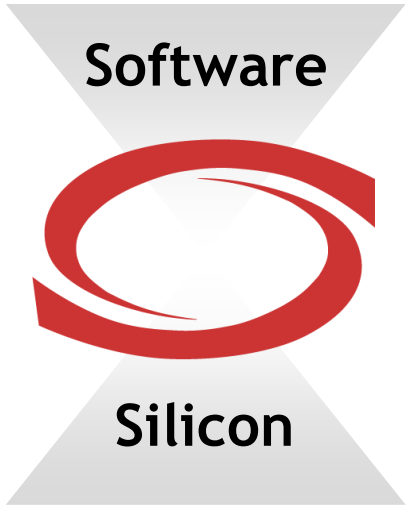
The OpenVX Standard for Portable, Efficient Vision Processing

24 May 2018

Khronos Connects Software to Silicon

Industry Consortium creating OPEN STANDARD APIs for hardware acceleration
Any company is welcome—one company, one vote

ROYALTY-FREE specifications
State-of-the art IP framework protects members AND the standards



Conformance Tests and Adopters
Programs for specification integrity and cross-vendor portability

Low-level silicon APIs needed on every platform: graphics, parallel compute, vision, neural networks, augmented and virtual reality…

International, non-profit organization
Membership and Adopters fees cover operating and engineering expenses

Strong industry momentum
100s of man years invested by industry experts

Well over a *BILLION* people use Khronos APIs *every day*…

Khronos Open Standards

3D for the Web
- VR/AR and games in-browser
- Efficiently delivering runtime 3D assets

COLLADA
WebGL
glTF

Real-Time 2D/3D
- Virtual and augmented reality displays
- Cross-platform gaming and UI
- CAD and product design

Vulkan
OpenGL ES
OpenGL

NNEF
OpenVX

Vision and Neural Networks
- Tracking and odometry
- Scene analysis/understanding
- Neural network inferencing

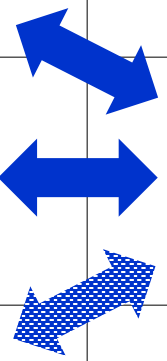
Parallel Computation
- Machine learning acceleration
- Embedded vision processing
- High-Performance Computing (HPC)

OpenCL
SPIR
SYCL

KHRONOS GROUP

Some Popular Khronos Standards

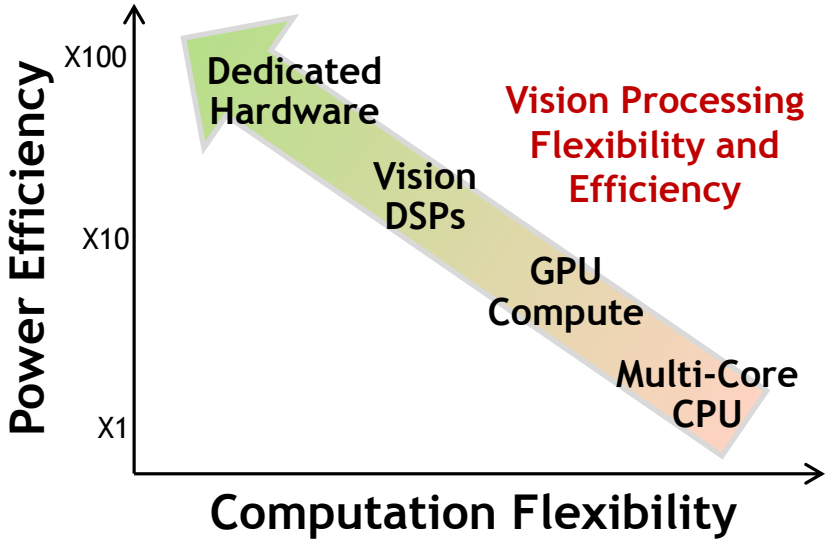
	Run-Time APIs	File Formats
GPU 3D and Compute	Vulkan, OpenGL, WebGL	glTF
Heterogenous Compute (Parallel Processing)	OpenCL, SYCL	SPIR
Vision and Inferencing	OpenVX	NNEF
VR and AR	OpenXR	

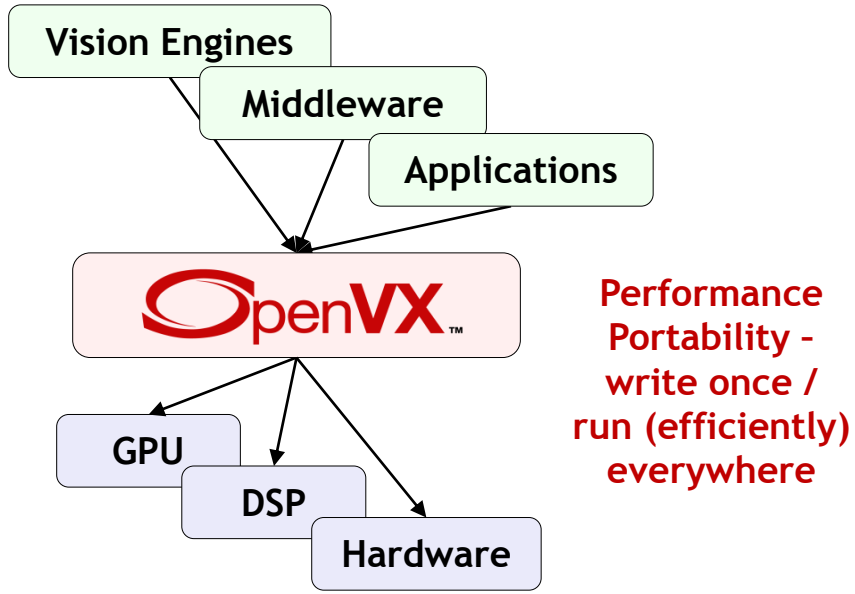


KHRONOS GROUP

OpenVX – Low Power Vision Acceleration

- **Targeted at vision acceleration in real-time, mobile, and embedded platforms**
  - High performance AND low power consumption are key
- **Higher abstraction than OpenCL for performance portability across diverse architectures**
  - Multi-core CPUs, GPUs, DSPs and DSP arrays, ISPs, FPGAs, dedicated hardware…
- **Extends portable vision acceleration to very low power domains**
  - Doesn't require high-power CPU/GPU complex or OpenCL precision





OpenVX Evolution

OpenVX 1.0
Spec released October 2014

Initial Functionality
Established graph framework
40 function nodes

Implementations
24 compliant products
from 8 companies

OpenVX 1.1
Spec released May 2016

New Functionality
Expanded nodes functionality
Enhanced graph framework

Implementations
18 compliant products
from 5 companies

OpenVX 1.2
Spec released May 2017

New Functionality
Conditional node execution
Feature detection
Classification operators
Expanded imaging operations

Extensions
Neural network acceleration
Graph save and restore
16-bit image operation

Safety Critical
OpenVX 1.1 SC for
safety-certifiable systems

New Extensions
Released 2017-2018

Pipelining/streaming
OpenCL interop
Kernel import: NNEF

OpenVX
Roadmap

Programmable user
kernels on accelerator

Enhanced NN support
NN-only profile

Other profiles/subsets

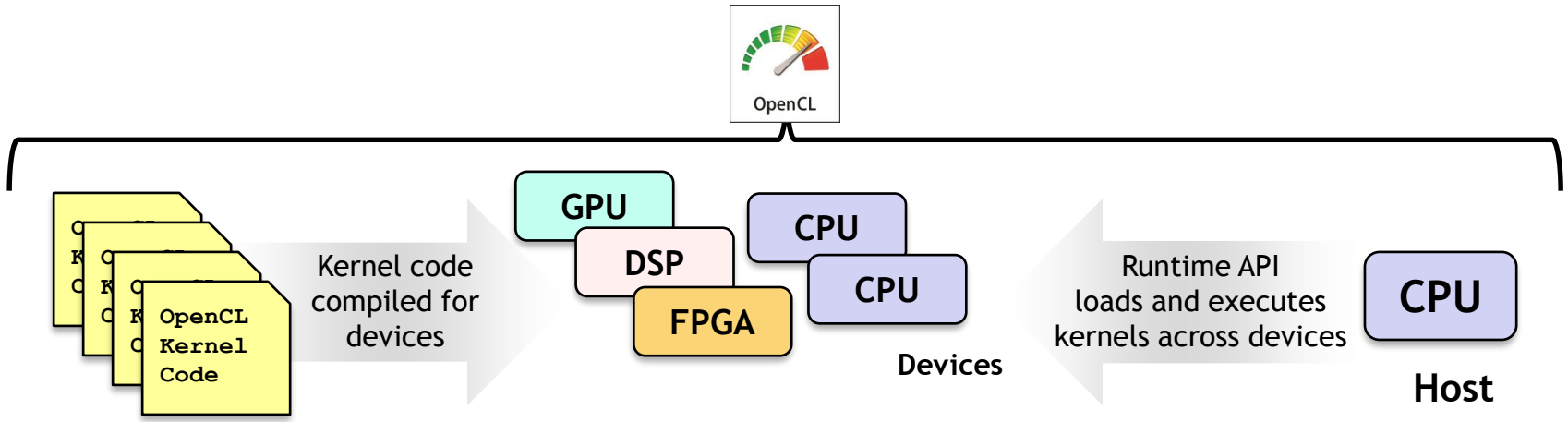
Safety-critical merge

More feature detectors

More flow control

OpenCL - Low-level Parallel Programing

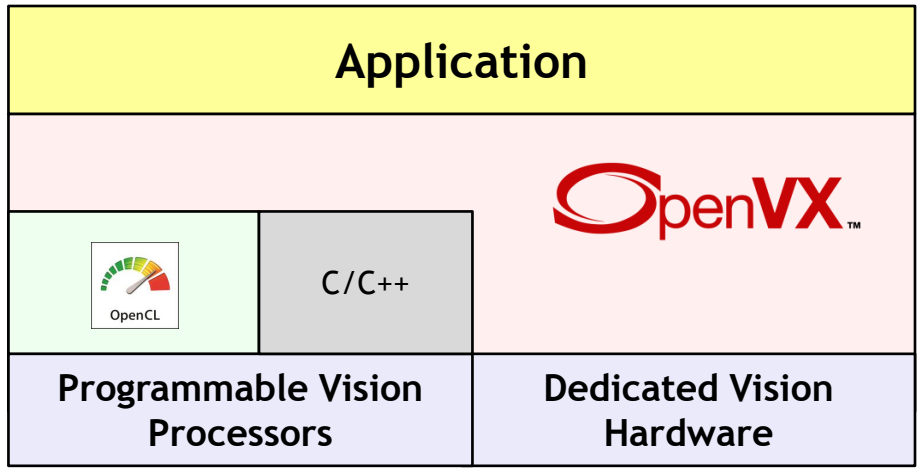
- **Low-level programming of heterogeneous parallel compute resources**
  - One code tree can be executed on CPUs, GPUs, DSPs, and FPGA
- **OpenCL C language to write kernel programs to execute on any compute device**
  - Platform-layer API—to query, select, and initialize compute devices
  - Runtime API—to build and execute kernels programs on multiple devices
- **New in OpenCL 2.2—OpenCL C++ kernel language, a static subset of C++14**
  - Adaptable and elegant sharable code, great for building libraries
  - Templates enable meta-programming for highly adaptive software
  - Lambdas used to implement nested/dynamic parallelism



Layered Vision Processing Ecosystem

Implementers may use OpenCL to *implement* OpenVX nodes on programmable processors

And then developers can use OpenVX to enable a developer to easily *connect* those nodes into a graph



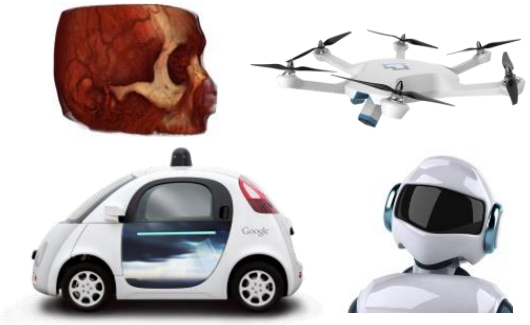
OpenVX enables the graph to be *extended* to include hardware architectures that don't support programmable APIs

The OpenVX graph enables implementers to *optimize* execution across diverse hardware architectures and drive to lower power implementations

How OpenVX Compares to Alternatives

	OpenVX	OpenCV	OpenCL
Governance	Open-standard API designed to be implemented and shipped by IHVs	Community-driven, open source library	Open-standard API designed to be implemented and shipped by IHVs
Programming Model	Graph defined with C API and then compiled for run-time execution	Immediate run-time function calls, reading to and from memory	Explicit kernels are compiled and executed via run-time API
Built-In Vision Functionality	Small but growing set of popular functions	Vast, mainly on PC/CPU	None, user programs their own or call vision library over OpenCL
Target Hardware	Any combination of processors or non-programmable hardware	Mainly PCs and GPUs	Any heterogeneous combination of IEEE FP-capable processors
Optimization Opportunities	Pre-declared graph enables significant optimizations	Each function reads/writes memory, power performance inefficient	Any execution topology can be explicitly programmed
Conformance	Implementations must pass conformance to use trademark	Extensive Test Suite but no formal Adopters program	Implementations must pass conformance to use trademark
Consistency	All core functions must be available in conformant implementations	Available functions vary depending on implementation/platform	All core functions must be available in all conformant implementations

Safety Critical APIs – Khronos Experience



OpenGL SC 1.0 - 2005
Fixed function graphics subset

OpenGL SC 2.0 - April 2016
Shader programmable pipeline subset

OpenGL ES 1.0 - 2003
Fixed function graphics

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

General lack of industry consensus on how APIs should be designed to streamline safety certification

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 members AND industry experts

KHRONOS SAFETY CRITICAL ADVISORY FORUM SC