APIs for Accelerating Vision and Inferencing: Options and Trade-offs

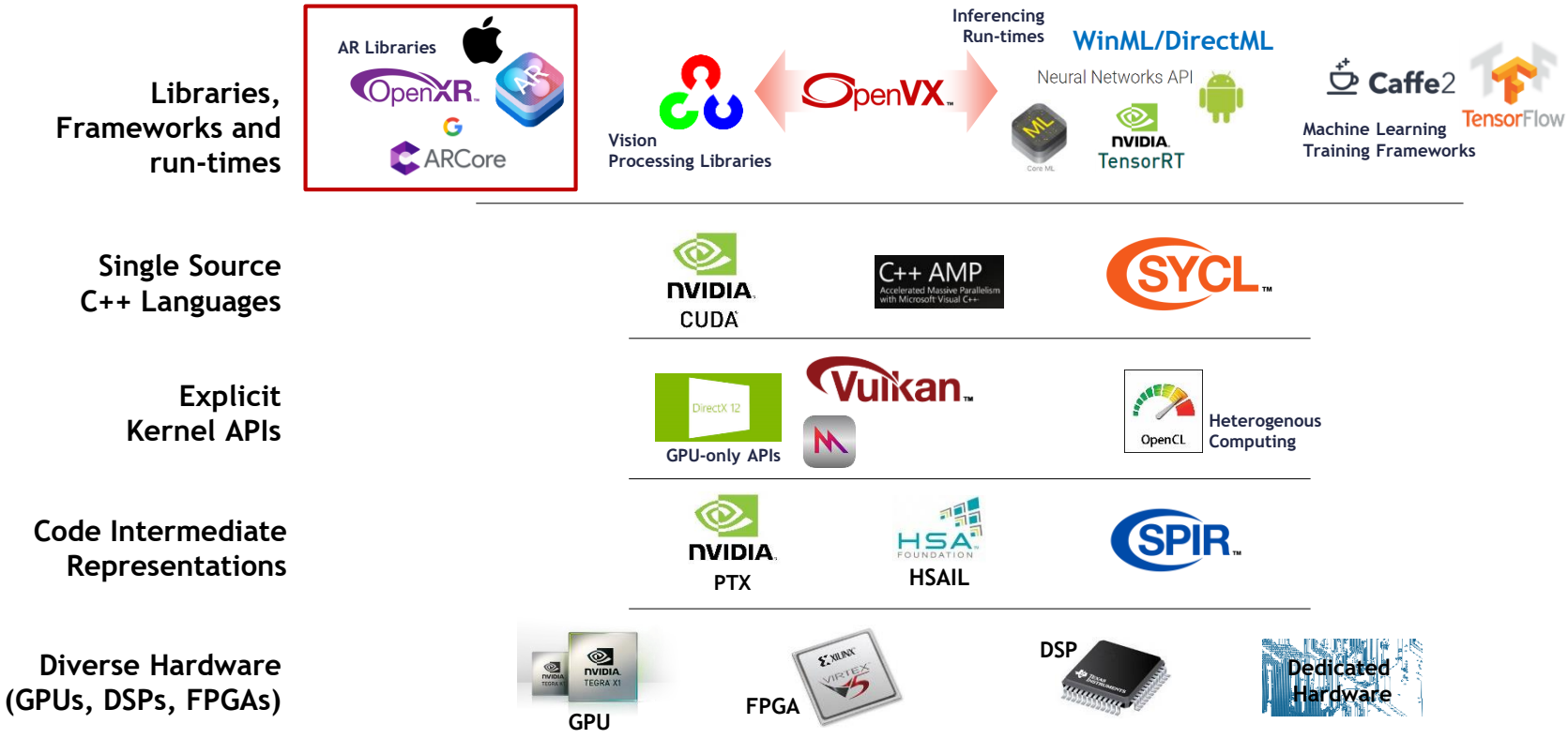
Neil Trevett, Khronos President

May 2018

KHRONOS GROUP

The Search for Vision Performance and Portability

Leads to a Layered Acceleration Ecosystem



Mobile Augmented Reality Libraries

Encapsulated Vision-based Functionality Also leveraging motion sensors	ARKit	ARCore
Pose Tracking	Yes	Yes
Plane Detection and Tracking	Yes	Yes
Image Recognition and Tracking	Yes	Yes
Ambient light level and temperature	Yes	Yes
Link to Neural Net-based Object Detection	Yes	Yes
Access to Point Cloud	Yes	Yes
Face tracking (iPhone X)	Yes	No
Light probe (using a tracked face)	Yes	No
Multi-user cloud-based anchors	No	Yes
OS Availability	iOS	Android and iOS

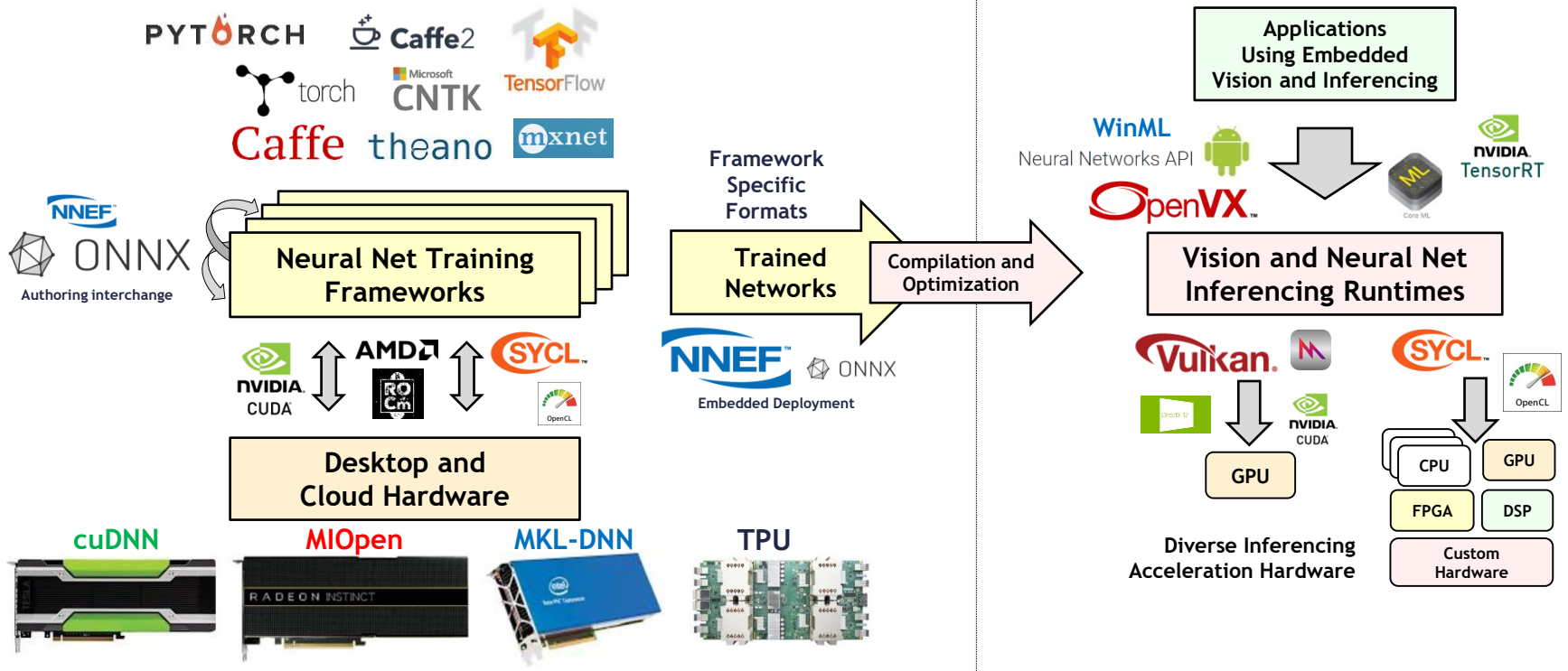
8th Wall vuforia

AR Libraries typically use ARKit/ARCore if available or implement own tracking if not

OpenXR

V1.0 is VR focused. Subsequent versions will extend to cross-platform AR

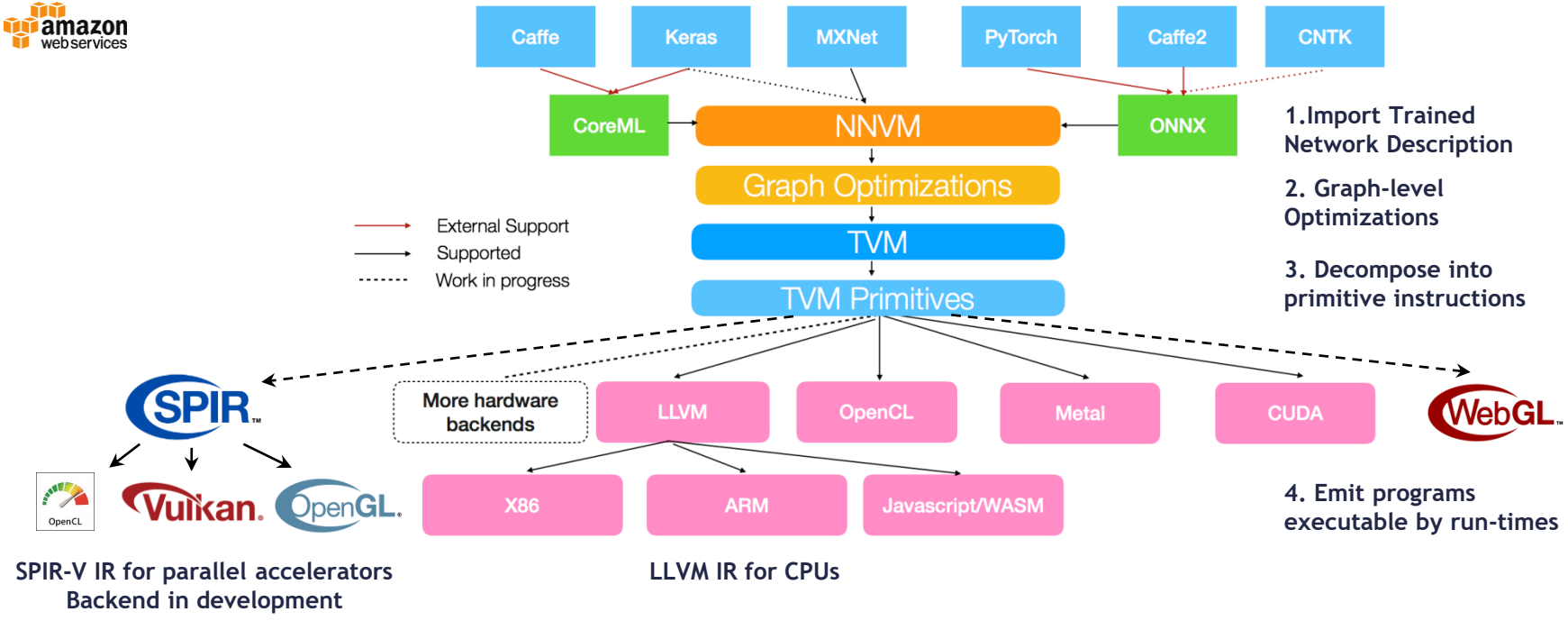
Neural Network Workflow



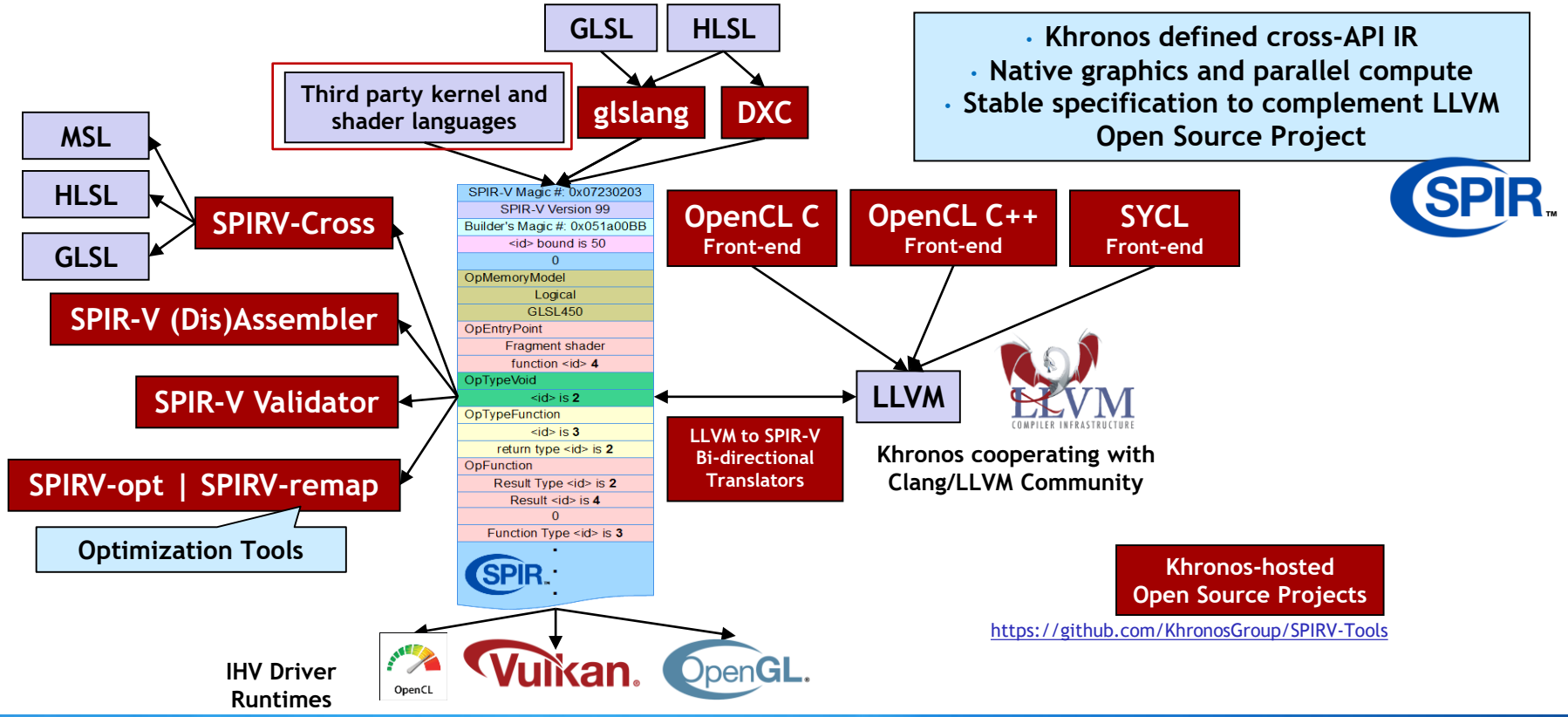
NNVM - Open Compiler for AI Inferencing

Paul G. Allen School of Computer Science & Engineering, University of Washington
http://www.tvmlang.org/2017/08/17/tvm-release-announcement.html

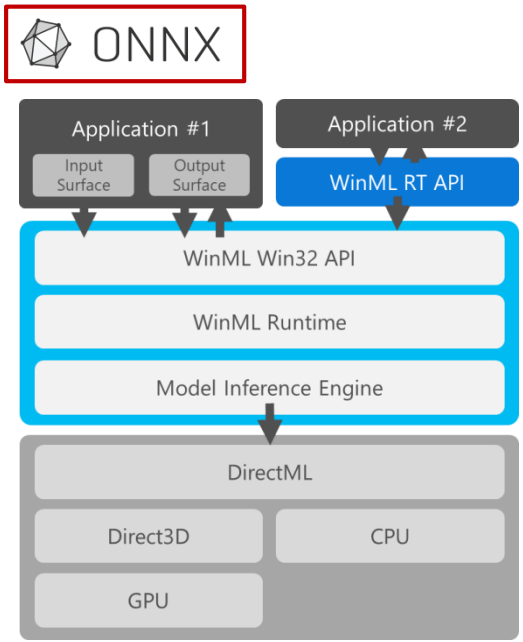
amazon web services



SPIR-V Ecosystem



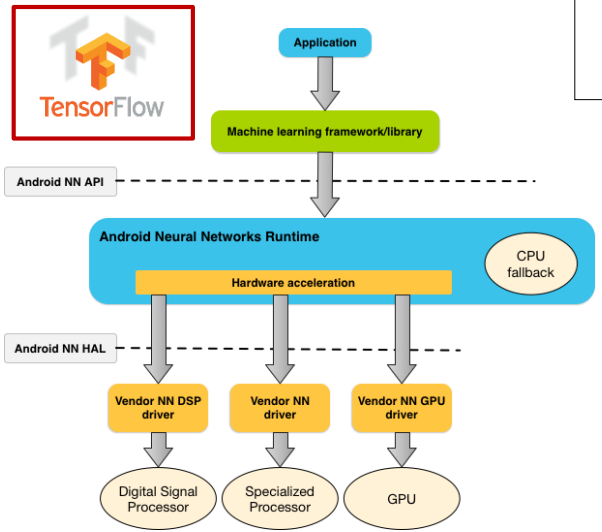
Platform Neural Network Stacks



Microsoft Windows

Windows Machine Learning (WinML)

https://docs.microsoft.com/en-us/windows/uwp/machine-learning/



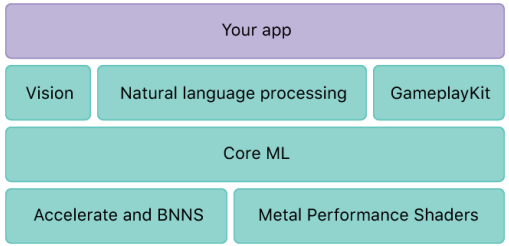
Google Android

Neural Network API (NNAPI)

https://developer.android.com/ndk/guides/neuralnetworks/

Common Fundamental Steps

1. Import trained NN model file
2. Build optimized version of graph
3. Accelerate on GPU or other processor using available low-level API

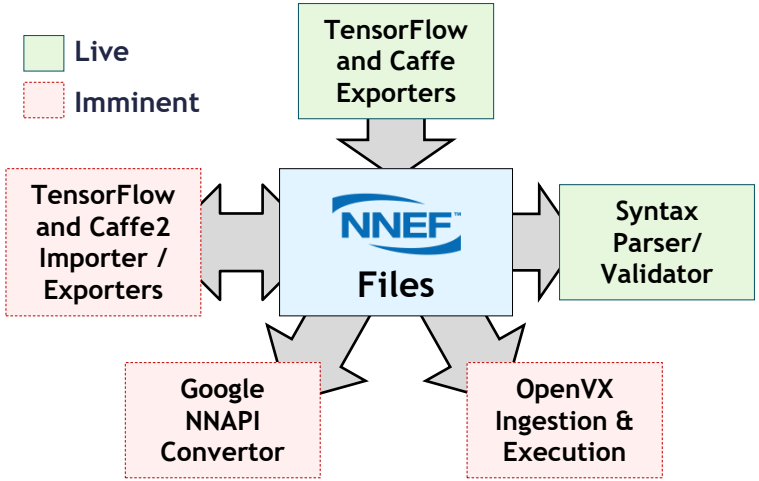


Apple MacOS and iOS

CoreML

https://developer.apple.com/documentation/coreml

NNEF Ecosystem



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

NNEF 1.0 Provisional Released for industry feedback before finalization

Comparing Neural Network Exchange Industry Initiatives

NNEF	ONNX
Defined Specification	Open Source Project
Stability for hardware deployment	Software stack flexibility
Multi-company Governance	Initiated by Facebook
Flat and Compound Ops	Flat Ops Only

NNEF File Structure

Network Structure File

Distilled, platform independent network description
Human readable, syntactical elements from Python

Standardized Operations

Rigorously defined semantics
Linear, convolution, pooling, normalization, activation, unary/binary
Supports fully connected, convolutional, recurrent architectures

Two Levels of Expressiveness

FLAT: Basic transfer of computation graphs with standardized operations
Simple to parse and translate to vendor specific formats

COMPOSITIONAL: Define custom compound operations
Higher-level graph descriptions
More complex to parse but offers more optimization hints

NNEF™

Can associate multiple Data Files with one Network Structure File e.g. the same data in multiple formats

Network Data File

Binary format contains parameter tensors
Supports float and quantized (integer) data
Flexible bit widths and quantization algorithms
Quantization algorithms expressed as extensible compound operations
Quantization info provided as hints for execution

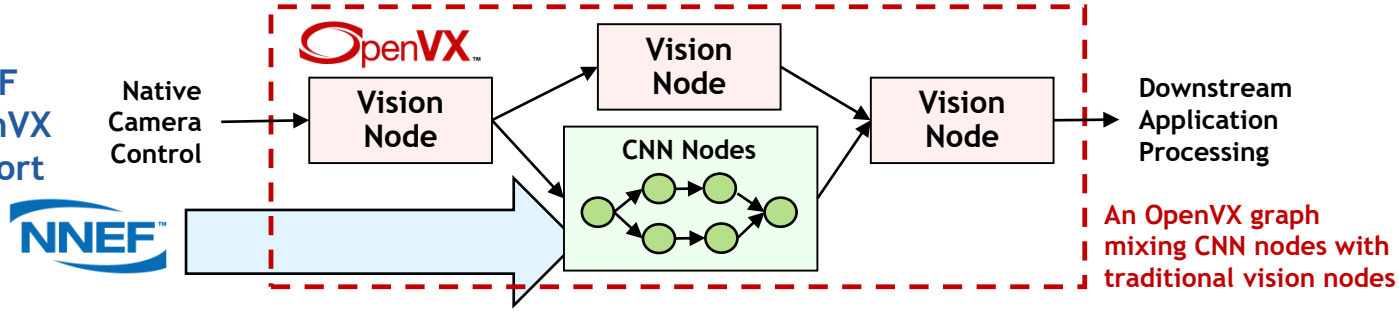
Split Structure and Data files

Easy independent access to network structure or individual parameter data
Set of files can use a container such as tar or zip with optional compression and encryption

NNEF Execution within OpenVX

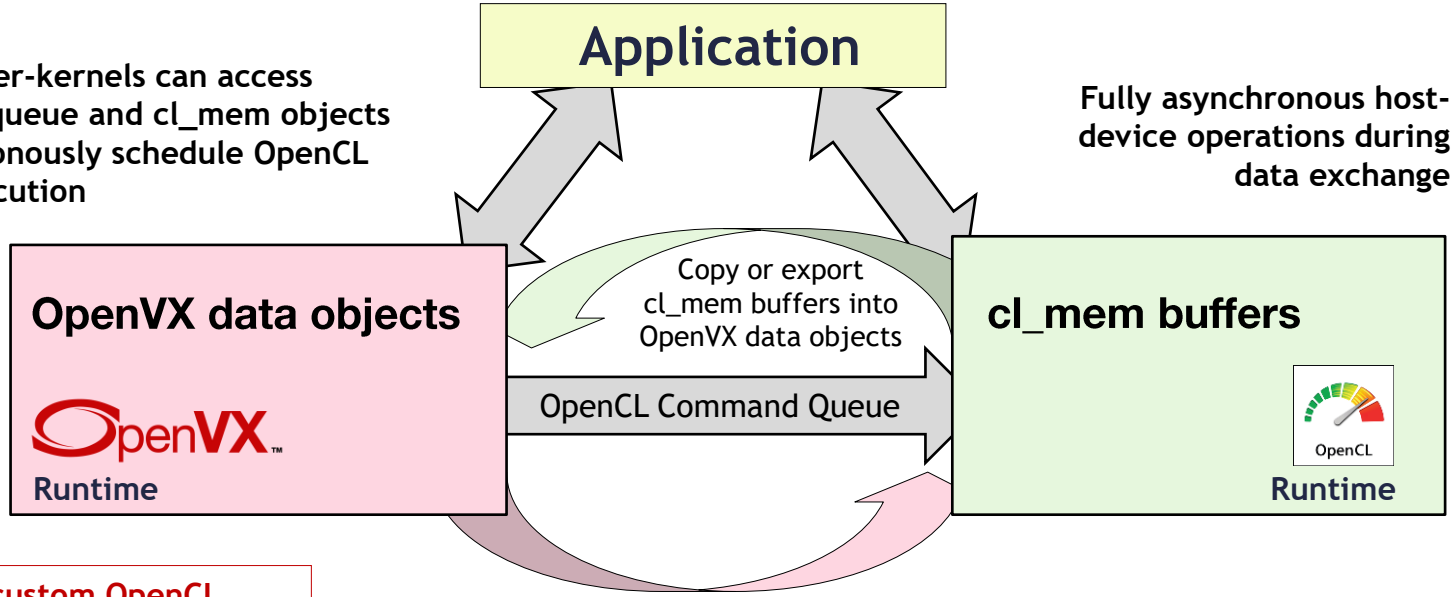
- Convolution Neural Network topologies can be represented as OpenVX graphs
  - Can also combine traditional vision and neural network operations
- OpenVX Neural Network Extension
  - Defines OpenVX nodes to represent many common NN layer types
  - Layer types include convolution, activation, pooling, fully-connected, soft-max
  - Defines multi-dimensional tensors objects to connect layers
- Kernel Import Extension
  - Enables loading of external program representations into OpenVX graphs

Importer converts NNEF representation into OpenVX Graphs using Kernel Import Extension



OpenVX / OpenCL Interop Extension

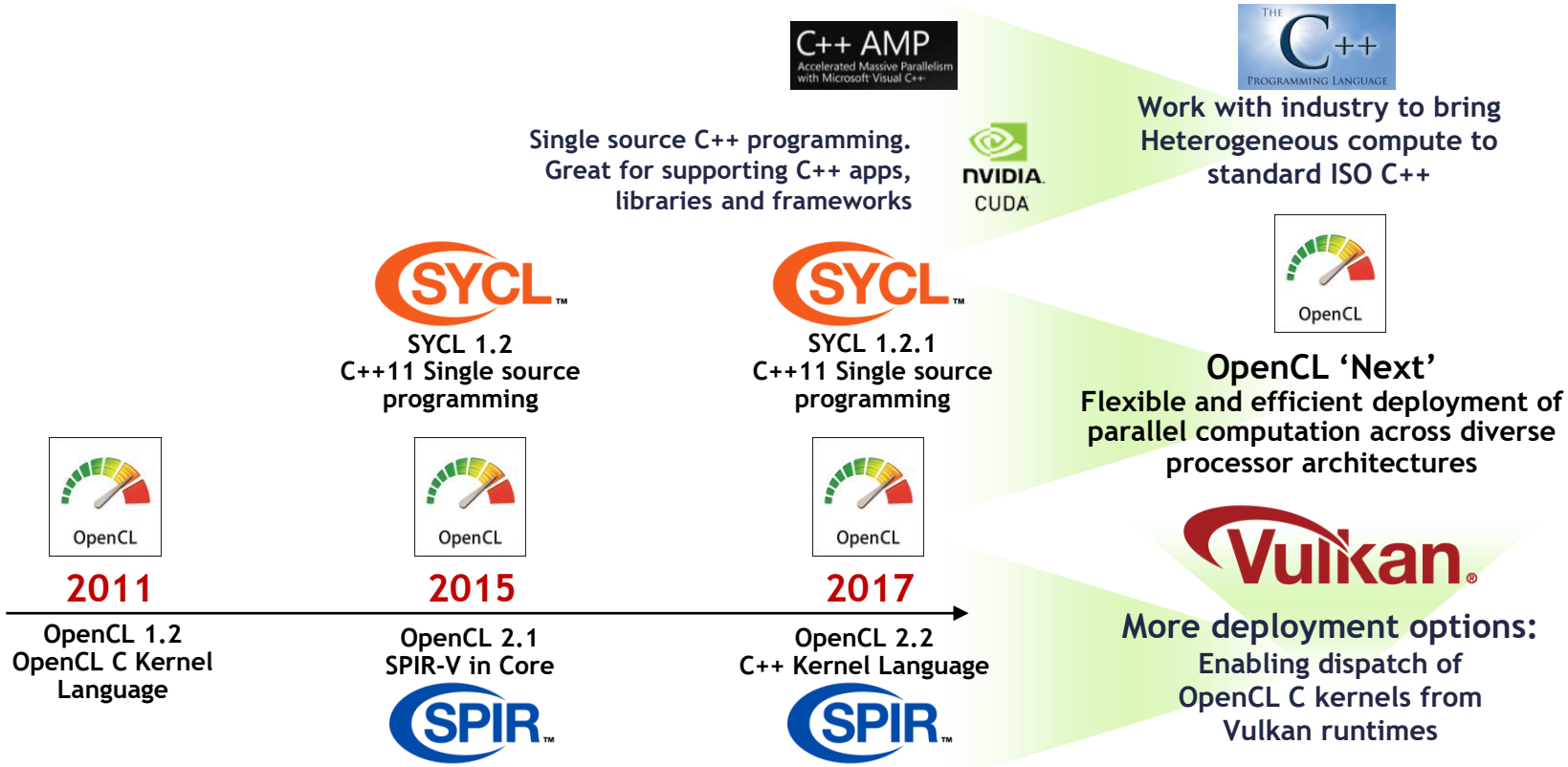
OpenVX user-kernels can access command queue and cl_mem objects to asynchronously schedule OpenCL kernel execution



Fully asynchronous host-device operations during data exchange

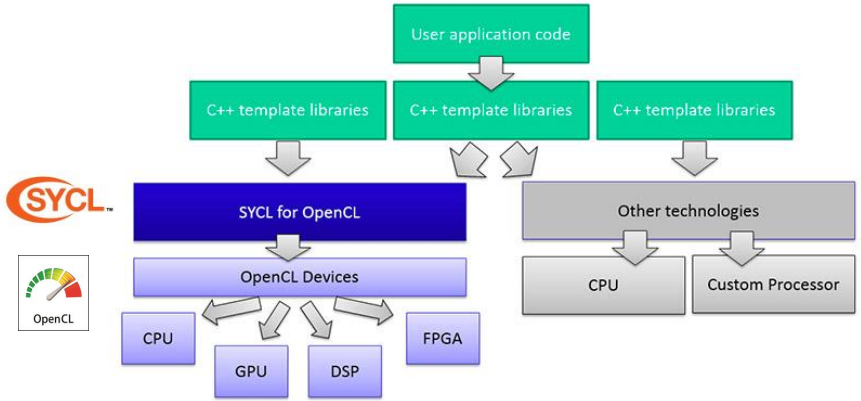
Enables custom OpenCL acceleration to be used within OpenVX User Kernels

OpenCL Ecosystem Roadmap



SYCL Ecosystem

- Single-source heterogeneous programming using STANDARD C++
  - Use C++ templates and lambda functions for host & device code
  - Layered over OpenCL
- Fast and powerful path for bring C++ apps and libraries to OpenCL
  - C++ Kernel Fusion - better performance on complex software than hand-coding
  - SYCLBLAS, SYCL Eigen, SYCL TensorFlow, SYCL DNN, SYCL GTX, VisionCpp
- Close cooperation with ISO C++
  - C++17 Parallel STL hosted by Khronos
  - C++20 Parallel STL with Ranges
- Implementations
  - triSYCL, ComputeCpp, ComputeCpp SDK …
- More information at http://sycl.tech



Pervasive Vulkan 1.0

Major GPU Companies supporting Vulkan for Desktop and Mobile Platforms

AMD arm BROADCOM Imagination intel NVIDIA QUALCOMM VeriSilicon

http://vulkan.gpuinfo.org/

Platforms

Desktop

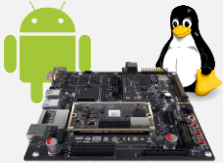
Mobile
(Android 7.0+)

Media Players

Consoles

Virtual Reality

Cloud Services



Embedded

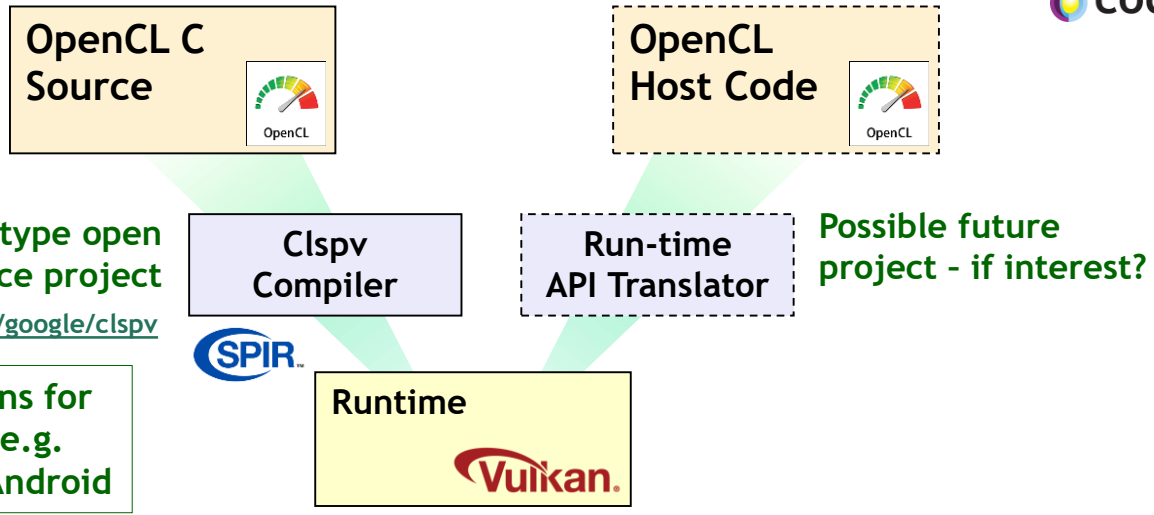
Clspv OpenCL C to Vulkan Compiler

- Experimental collaboration between Google, Codeplay, and Adobe
  - Successfully tested on over 200K lines of Adobe OpenCL C production code
- Compiles OpenCL C to Vulkan's SPIR-V execution environment
  - Proof-of-concept that OpenCL kernels can be brought seamlessly to Vulkan
  - Significant parts OpenCL C 1.2 so far - shaped by submitted workloads

Google

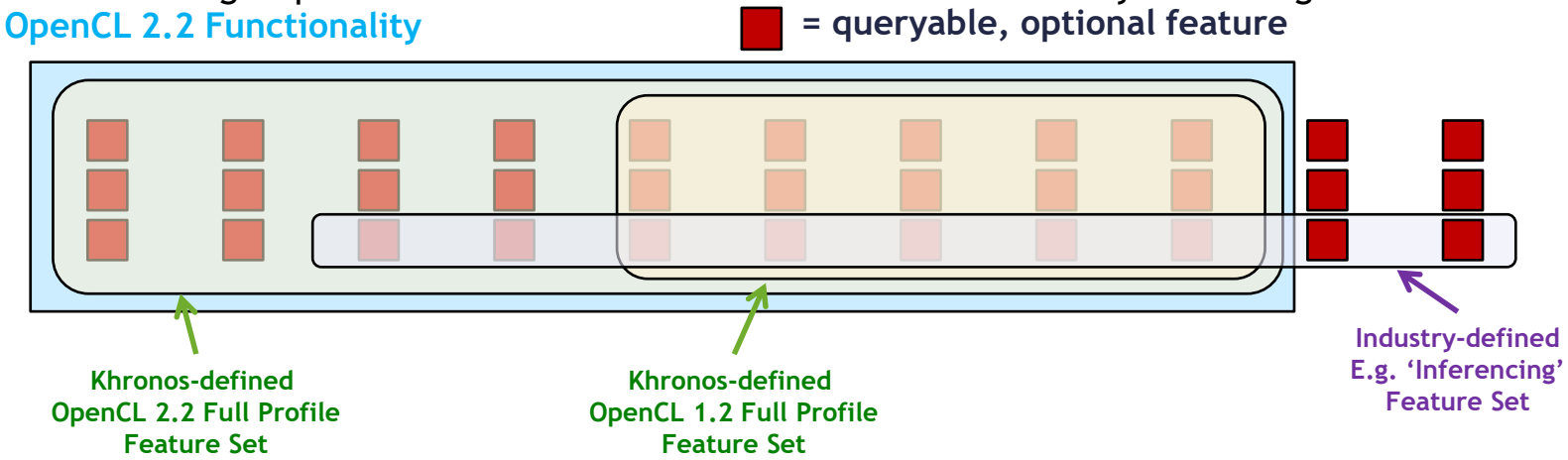
Adobe

codeplay

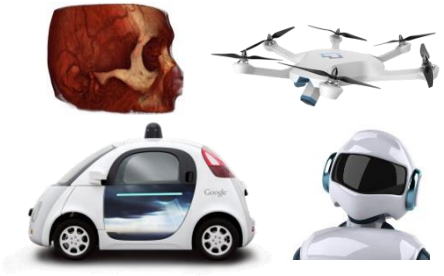


OpenCL Next – Deployment Flexibility

- Vendors can support ANY combination of features to suit their hardware/market
  - If all exposed features are conformant – the implementation is conformant
- Khronos will define Feature Sets equivalent to current profiles
  - Existing profiles and device types not going away! No changes to existing applications
- Opportunity to coalesce industry support around market-focused feature sets
  - Khronos aiming to provide Feature Set infrastructure for the industry to leverage



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

Khronos Safety Critical Advisory Forum

Industry outreach and cooperation

AESIN
Automotive ADAS & AV + security
https://aesin.org.uk

MISRA C++
C++ WG23 Programming Vulnerabilities
ISO C Safe and Secure SG
ISO C++ Vulnerabilities Safety Critical SG

KHRONOS SAFETY CRITICAL ADVISORY FORUM | SC™

Generating guidelines for designing safety critical APIs to ease system certification.
Open to Khronos member AND industry experts
https://www.khronos.org/advisors/kscaf

We are inviting safety critical experts to join KSCAF!
No cost or work commitment

Khronos SC Activities

OpenCL

OpenCL SC TSG
Working on OpenCL SC
Gathering requirements

SYCL™

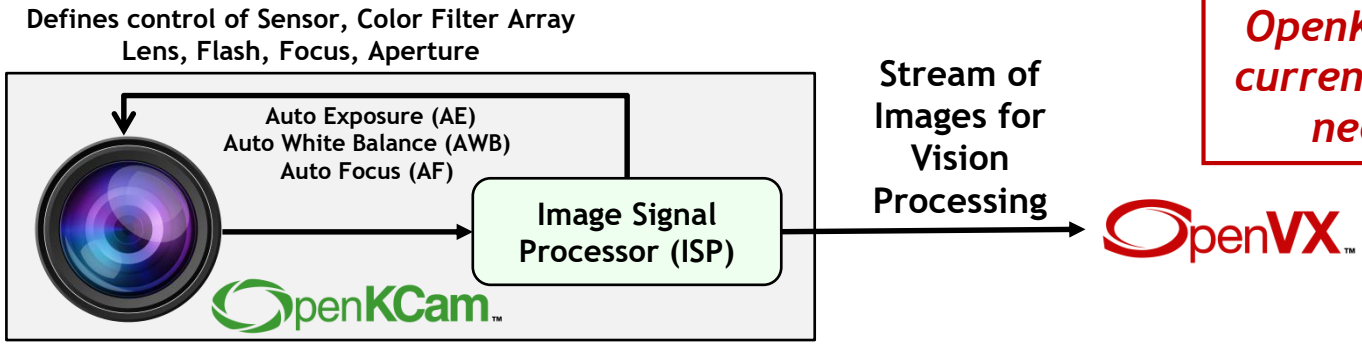
SYCL SC
Guidelines to augment Industry
First Safe and Secure Parallel and Heterogeneous C++
Safe AI for Automotive

OpenVX™

OpenVX SC 1.1 - May 2017
Restricted "deployment" implementation only executes pre-compiled binary format

Need for Camera Control API?

- Advanced control of sensor and camera subsystem - with cross-platform portability
  - Generate sophisticated image stream for advanced imaging & vision apps
- No platform API currently fulfills all developer requirements
  - Portable access to growing sensor diversity: e.g. depth sensors and sensor arrays
  - Cross sensor synch: e.g. synch of 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



OpenKCAM standard is currently on ice - do we need to restart?

Key Takeaways and What's Next?

- Vision tools and API ecosystem becoming increasingly sophisticated
  - Layering libraries, languages and run-times
  - Machine learning stacks continue to grow in flexibility
- Exchange formats and compiler technologies essential to flexible deployment
  - Open standards evolving alongside proprietary solutions
  - For developers that value cross-platform deployment
- Safety-critical APIs becoming increasingly essential
  - Many vision applications need system certification
- Still no cross-vendor camera APIs?
  - Is the time yet right for this to be a target for standardization?
- Please join if your company interested in Khronos open standards
  - Neil Trevett | ntrevett@nvidia.com | @neilt3d