14th International Workshop on OpenCL and SYCL

IWOCL 2026

OpenCL: State of the Union 2026

Ben Ashbaugh, Intel

(Representing the Khronos OpenCL Working Group)

May 6-8, 2026 | Heilbronn, Germany | iwocl.org

KHRONOS GROUP

Let’s think back briefly to 2020…

This work is licensed under a Creative Commons Attribution 4.0 International License

This work is licensed under a Creative Commons Attribution 4.0 International License

This work is licensed under a Creative Commons Attribution 4.0 International License

This work is licensed under a Creative Commons Attribution 4.0 International License

This work is licensed under a Creative Commons Attribution 4.0 International License

This work is licensed under a Creative Commons Attribution 4.0 International License

OpenCL 3.1

- **Raises the Bar for Heterogeneous Compute Acceleration**
  - Incorporates Features with Proven Value, Use, and Support
- **Expecting Wide Adoption in 2026**
  - Including open source and layered implementations
  - Arm, CLVK, Imagination, Intel, Mesa Rusticl, PoCL, Qualcomm, …
- **Continuing to Innovate via Extensions**
  - Driven by multiple markets: mobile, embedded, desktop, HPC, AI
  - Finalizing: Command Buffers, USM, Cooperative Matrix, more!
  - In development: AI data formats, image tiling controls, more!

KHRONOS GROUP

This work is licensed under a Creative Commons Attribution 4.0 International License

Khronos Connects Software to Silicon

Founded in 2000
~ 135 Members | ~ 40% US, 30% Europe, 30% Asia

KHRONOS GROUP

Non-profit Standards Consortium creating open, royalty-free standards

Focused on runtime APIs and file formats for 3D, XR, AI, vision, and parallel compute acceleration

Member-driven, open to any company

This work is licensed under a Creative Commons Attribution 4.0 International License

Khronos Compute Acceleration

Choice of programming models to meet the needs of diverse developers

Higher-level applications, libraries, and languages often access hardware acceleration through lower-level APIs

Higher-level Languages and APIs
Streamlined development and performance portability

SYCL

Single source C++ programming with compute acceleration

OpenVX

Graph-based vision and inferencing acceleration

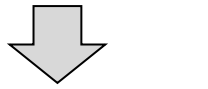
PyTorch, gstreamer, OpenCV, FFmpeg

Third party vision, streaming and inferencing libraries

Lower-level Languages and APIs
Explicit hardware control

Vulkan

GPU rendering + compute acceleration



GPU

Shaders

SPIR-V

Kernels

Intermediate Representation (IR) language compiler target supporting parallel execution and graphics

OpenCL

Heterogeneous compute acceleration

FPGA | CPU | GPU | DSP

Custom HW | AI/Tensor HW

This work is licensed under a Creative Commons Attribution 4.0 International License

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

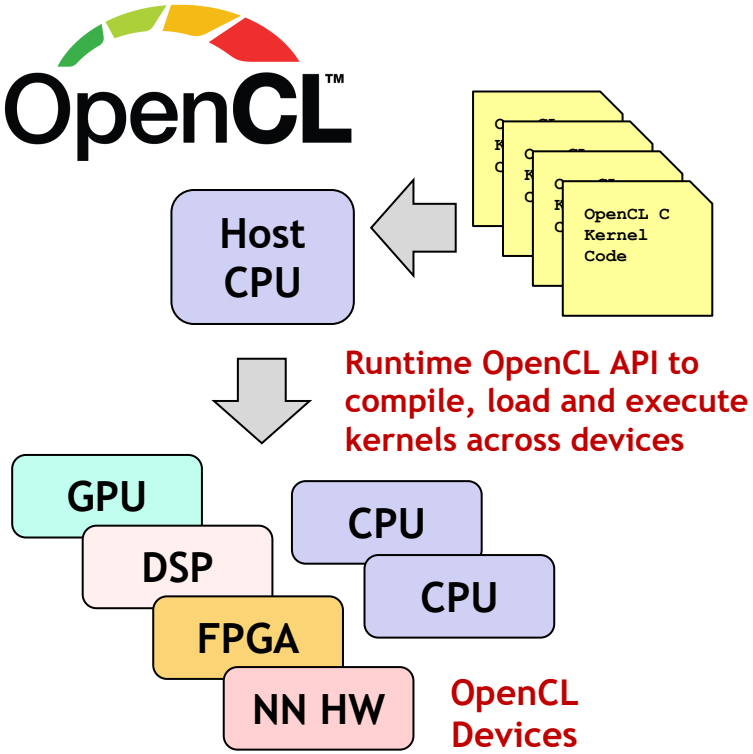
Query, select and initialize compute devices

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

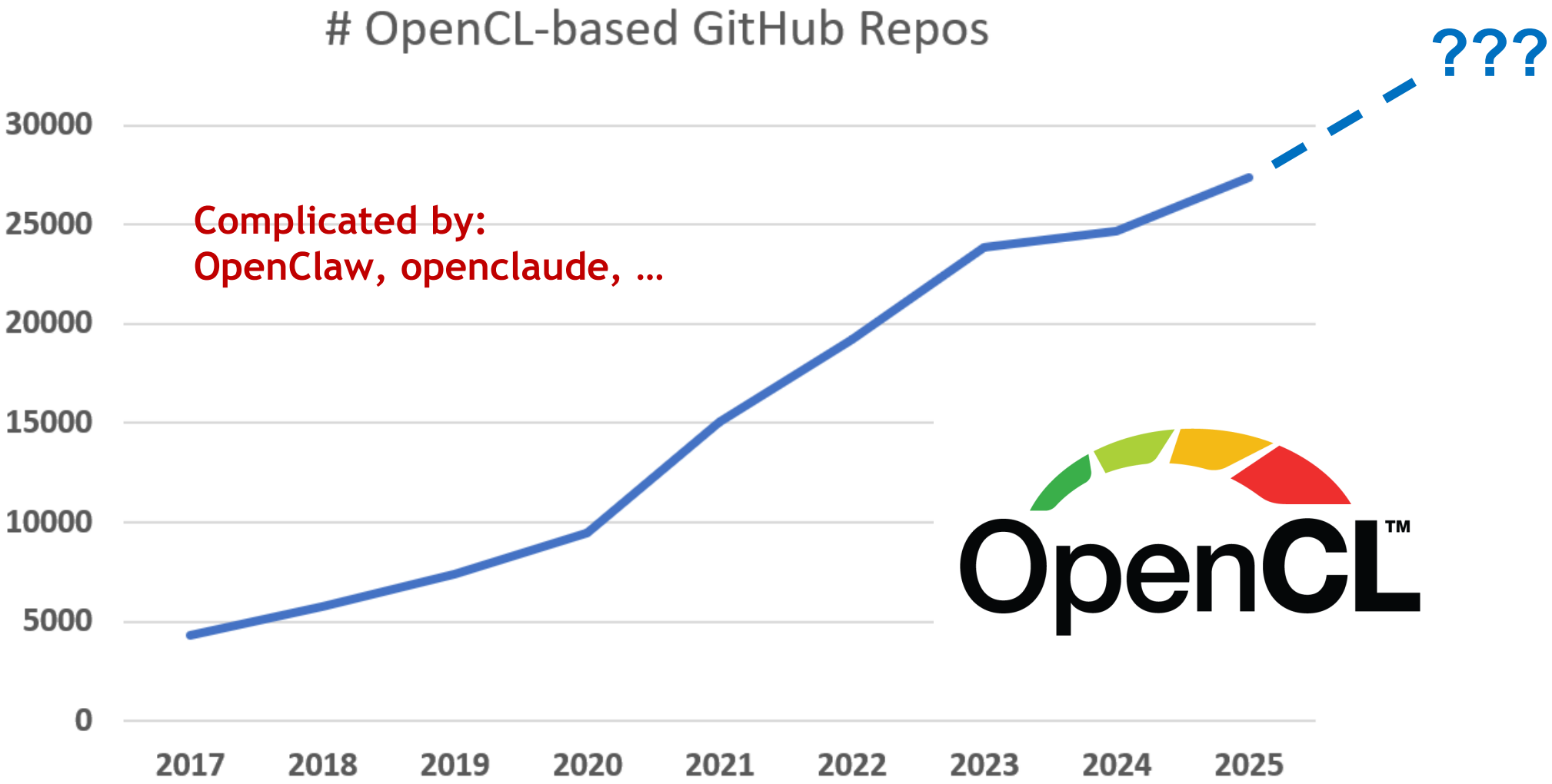


Complements GPU-only APIs

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

This work is licensed under a Creative Commons Attribution 4.0 International License

OpenCL Open-Source Project Momentum



KHRONOS GROUP

This work is licensed under a Creative Commons Attribution 4.0 International License

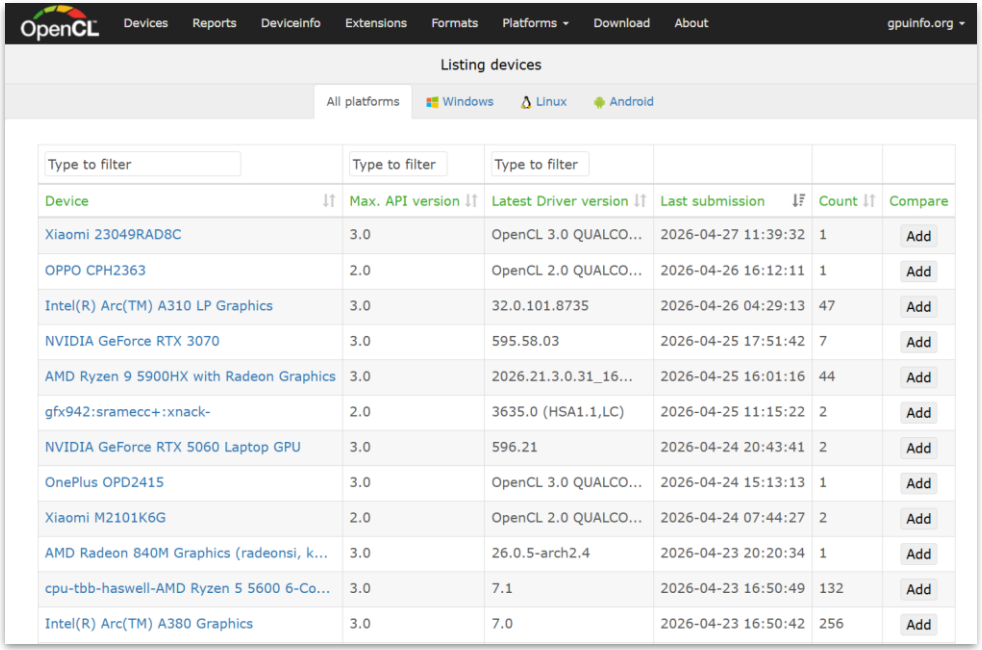
OpenCL on GPUInfo.org

The online GPUinfo.org database is populated using the [OpenCL Hardware Capability Viewer] application

Available for Windows, Linux and Android

Reads and displays OpenCL information and uploads to the database

Please download and run to help populate the database!



This work is licensed under a Creative Commons Attribution 4.0 International License

API Layering

Enabled by growing robustness of open-source compiler ecosystem using SPIR-V

Layers Over	**Vulkan**	**OpenGL**	**OpenCL**	**OpenGL ES**	**DX12**	**DX8-11**
Vulkan		Zink	clspv + clvk/Angle Rusticl + Zink	Angle GLOVE	vkd3d-Proton vkd3d	DXVK WineD3D
OpenGL	Ashes			Angle		WineD3D
DX12	Dozen	Microsoft 'GLOn12'	Microsoft 'CLOn12'			Microsoft D3D11On12
DX9-11	Ashes			Angle		
Metal	MoltenVK			Angle MoltenGL		

ROWS Benefit Platforms by adding APIs

COLUMNS Benefit ISVs by making an API available everywhere

This work is licensed under a Creative Commons Attribution 4.0 International License

Layered OpenCL Implementations

clspv + clvk

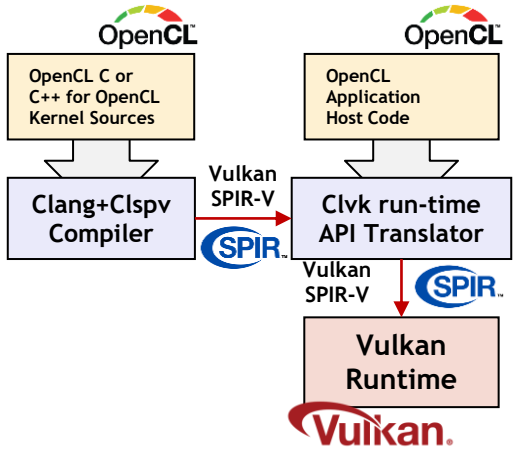
OpenCL over Vulkan
Google

clspv 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

Android chromeOS Google



clspv + Ancle

OpenCL over Vulkan
Samsung

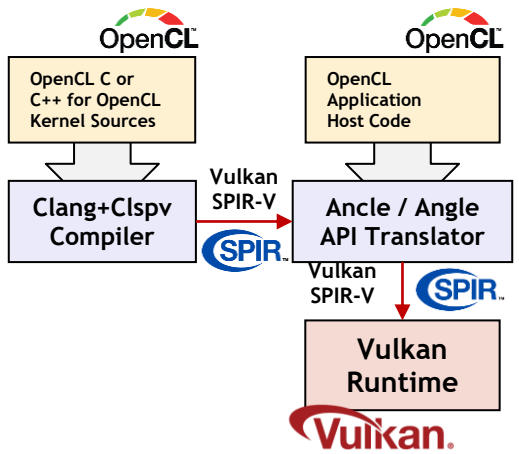
Integrates clspv and OpenCL runtime into Angle code base

Samsung Motivation
"OpenCL is widely used and deployed and is making a comeback thanks to ML"

"OpenCL is a favored high-level (front-end) compute language! Easier to write than Vulkan"

Ancle makes OpenCL a first-class citizen in Android by relying on Vulkan as its Native Driver"

Android SAMSUNG

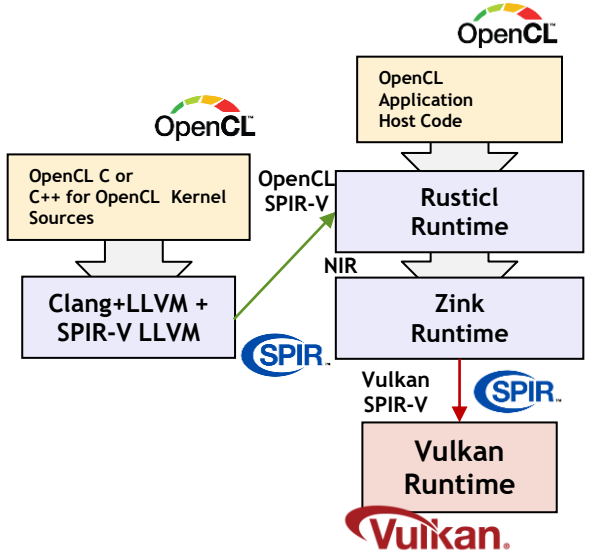


Rusticl over Zink

OpenCL over Vulkan
Mesa

The Zink Gallium driver emits Vulkan API calls and now supports OpenCL Kernels

MESA

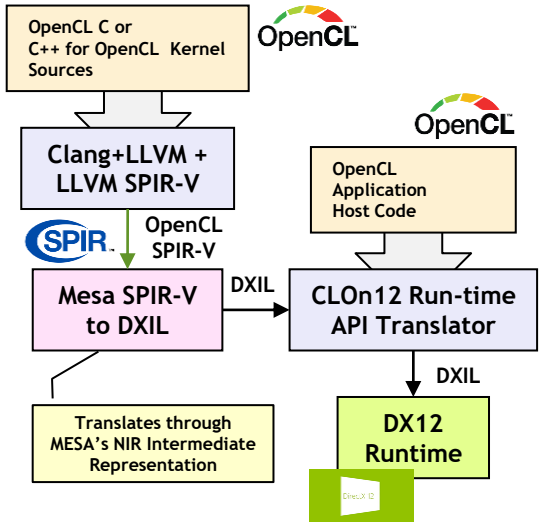


OpenCLOn12

OpenCL over DX12
Microsoft

GPU-accelerated OpenCL on any DX12 PC and Cloud instance (x86 or Arm)

Microsoft COLLABORA



This work is licensed under a Creative Commons Attribution 4.0 International License

OpenCL 3.1 Feature Deep Dive

This work is licensed under a Creative Commons Attribution 4.0 International License

Key OpenCL 3.1 Feature:

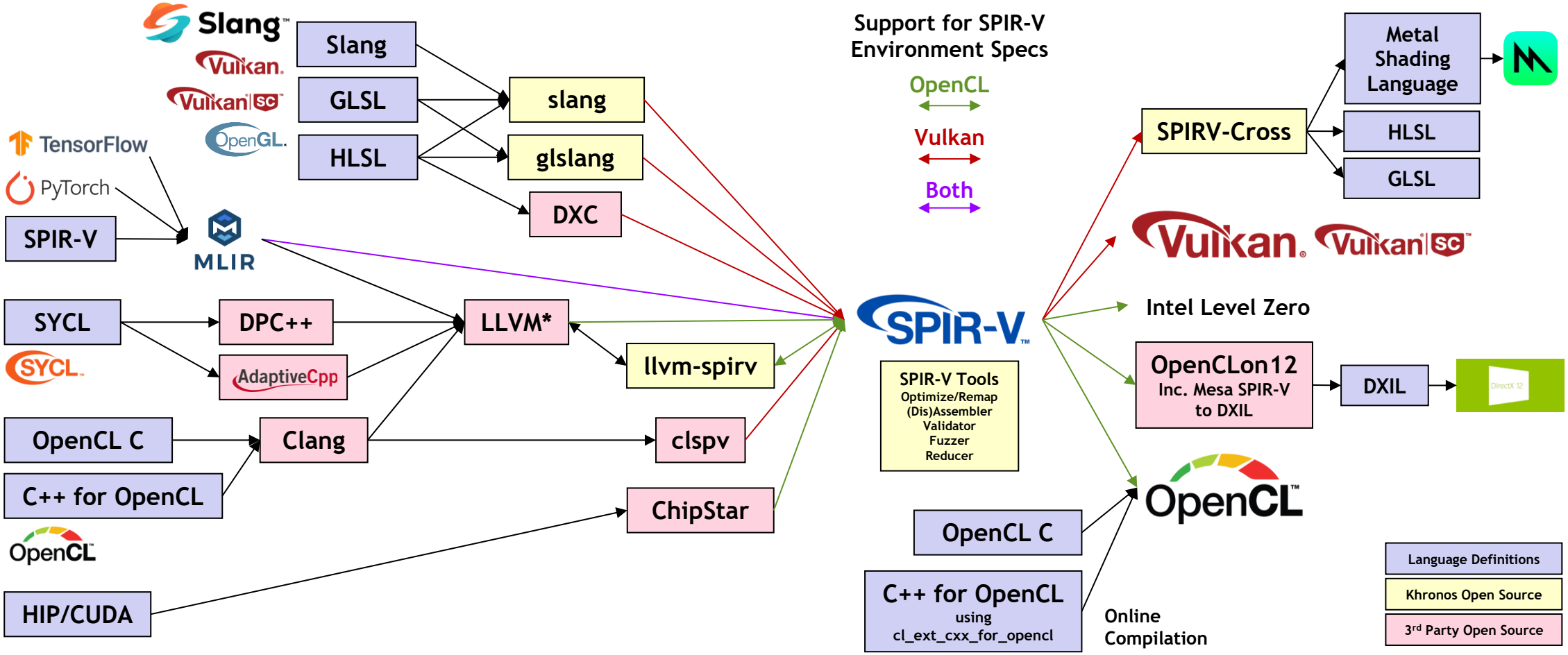
- **OpenCL 3.1 requires support for SPIR-V consumption!**
  - Significantly enhances the viability of OpenCL as a language target
  - One of the most requested OpenCL features
  - Builds upon open source compiler and testing investments

- **OpenCL 3.1 also requires support for SPIR-V queries!**
  - Directly query the SPIR-V features supported by a device
  - Simplifies support for new SPIR-V extensions and capabilities

KHRONOS GROUP

This work is licensed under a Creative Commons Attribution 4.0 International License

SPIR-V Ecosystem



MLIR is part of the LLVM compiler infrastructure

This work is licensed under a Creative Commons Attribution 4.0 International License

Key OpenCL 3.1 Feature:

- **OpenCL 3.1 requires support for sub-groups!**
  - Commonly used feature in highly tuned kernels, including AI kernels
  - Supported operations include shuffles, rotations, reductions, and more
  - Important building block for additional extensions in the future

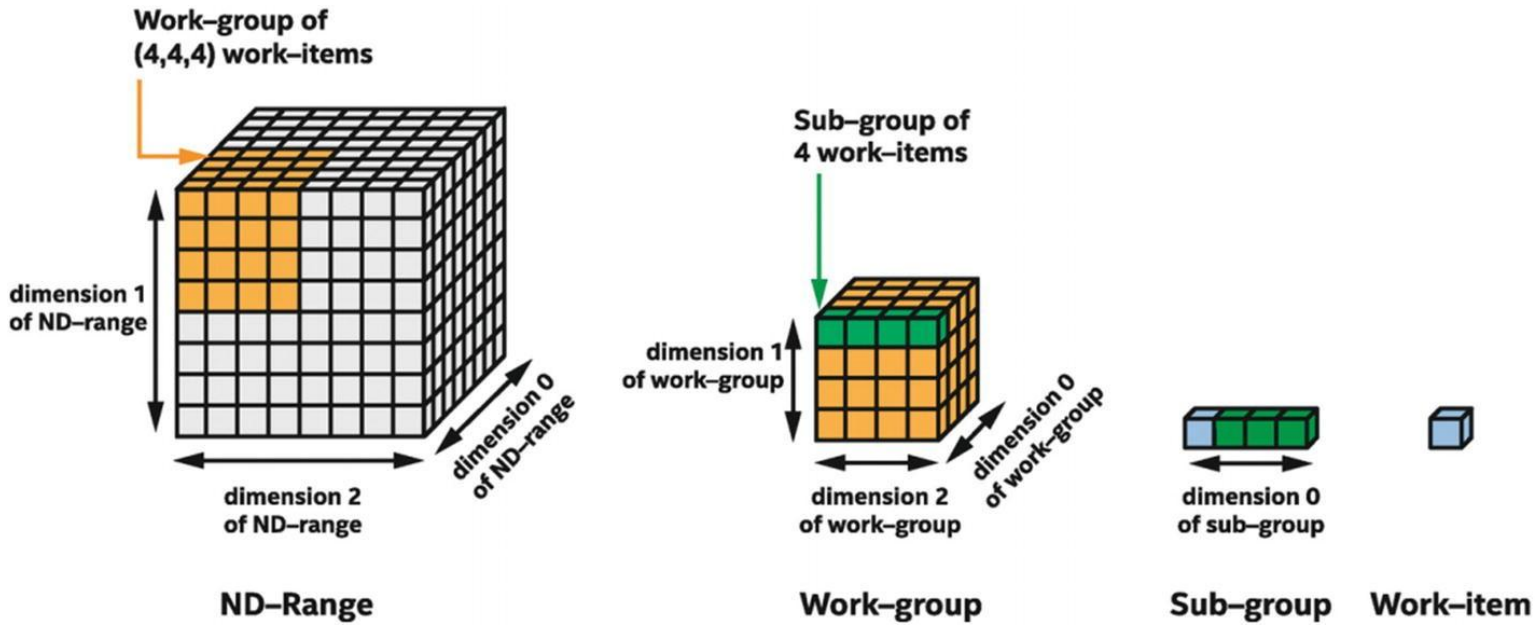
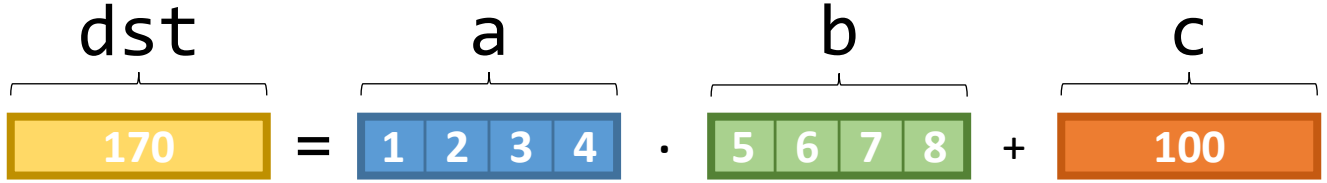


Diagram source: https://link.springer.com/chapter/10.1007/978-1-4842-9691-2_4/figures/12

This work is licensed under a Creative Commons Attribution 4.0 International License

More OpenCL 3.1 Features:

- **Integer Dot Products**
  - Dot product of small integer vectors, with saturation and accumulation
  - Building block for larger matrix multiplications



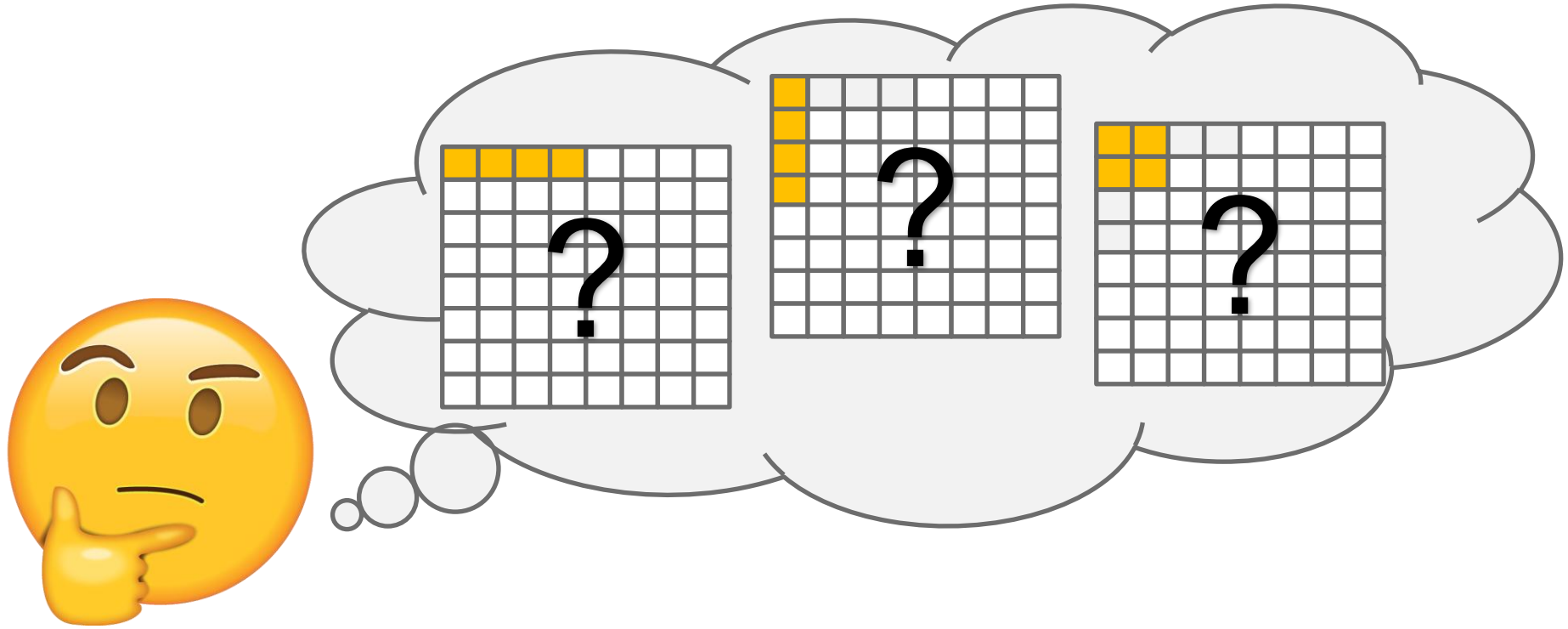
- **Extended Bit Operations**
  - Insert, Extract, and Reverse bits in a bitfield
  - Implemented with dedicated instructions on many devices

KHRONOS GROUP

This work is licensed under a Creative Commons Attribution 4.0 International License

More OpenCL 3.1 Features:

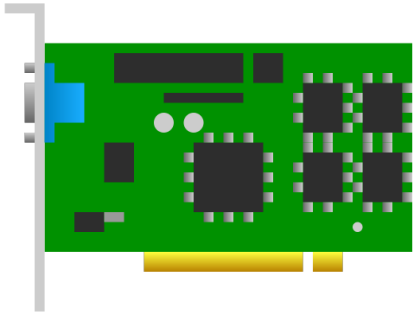
- **Suggested Local Work-Group Size**
  - Reduces overhead by amortizing determination over multiple enqueues
  - Provides insight to profilers when the local work-group size is NULL



This work is licensed under a Creative Commons Attribution 4.0 International License

More OpenCL 3.1 Features:

- **Device UUID**
  - Uniquely identify devices in the system, including across APIs
  - Important for multi-device systems and external memory sharing

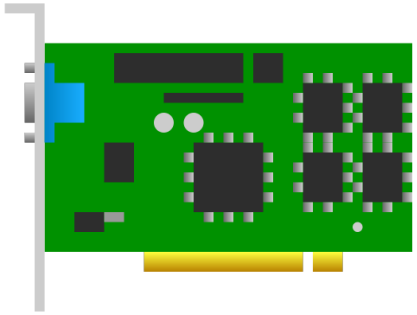


OpenCL™

CL_DEVICE_UUID_KHR	86800BE2-0000-0000-7500-000000000000

Vulkan®

VkPhysicalDeviceIDProperties::deviceUUID	86800BE2-0000-0000-7500-000000000000



OpenCL™

CL_DEVICE_UUID_KHR	86800BE2-0000-0000-7500-000000000001

This work is licensed under a Creative Commons Attribution 4.0 International License

Even More OpenCL 3.1 Features:

- **OpenCL C 3.1 Kernel Language**
  - Use new language features without relying on extensions
- **OpenCL C `printf`**
  - Adds support for `z` (`size_t`) and `t` (`ptrdiff_t`) length modifiers
- **`CL_DEVICE_HOST_UNIFIED_MEMORY`**
  - Clarified meaning; can be used to query integrated vs. discrete GPUs
- **Local Memory Kernel Arguments**
  - Can be set to zero to indicate no local memory is needed

KHRONOS GROUP

This work is licensed under a Creative Commons Attribution 4.0 International License

Even More OpenCL 3.1 Changes:

- **Execution Model Improvements**
  - Observing that an event is `CL_COMPLETE` is now a sync point
  - Matches intuition, previously needed to explicitly wait on the event

- **Memory Model Improvements**
  - Relaxed definition of “inclusive scopes” – scopes do not need to match
  - Also matches intuition, previous definition was unnecessarily strict

OpenCL 3.1 improves ergonomics, smooths out many rough edges!

KHRONOS GROUP

This work is licensed under a Creative Commons Attribution 4.0 International License

Observations about OpenCL 3.1 Features:

There is very little that is truly new in OpenCL 3.1.

We think this is a good thing, not a bad thing!

We want core OpenCL APIs to be widely supported and used!

Innovate via Extensions, Standardize Best Practice

This work is licensed under a Creative Commons Attribution 4.0 International License

What's next?

Where do we go after OpenCL 3.1?

Lots of different options!

We would love your feedback!

This work is licensed under a Creative Commons Attribution 4.0 International License

Continue Innovation via Extensions

Finishing and finalizing experimental extensions:

- **Command Buffers**
  - Record groups of commands to lower overhead, increase performance
- **Unified SVM (Unified Shared Memory / USM)**
  - Represent allocations as pointers, not as handles (`cl_mems`)
  - More control over memory placement and migration
- **Cooperative Matrices**
  - Access dedicated matrix hardware for AI applications
- **Improved External Memory Sharing**
  - More handle types for better interoperability

KHRONOS GROUP

This work is licensed under a Creative Commons Attribution 4.0 International License

Continue Innovation via OpenCL Extensions

Extensions in development:

- **New AI Data Types**
  - bfloat16, fp8, microscaling formats, …
- **More Sub-group Controls**
  - Required sub-group size, required sub-group lane mappings
- **External Memory Improvements**
  - Improved tiling controls, DRM format modifiers, …
- **And more!**

Today's OpenCL extensions are candidates for tomorrow's OpenCL core APIs!

This work is licensed under a Creative Commons Attribution 4.0 International License

OpenCL as a Substrate API

- **Continuing to see usage of OpenCL indirectly!**
  - As a heterogeneous Hardware Abstraction Layer (HAL)

Today:

chipStar

Vendor Inference Engines

Possible Future?

OpenVX™ ? OpenMP® ? **others?**

KHRONOS GROUP

This work is licensed under a Creative Commons Attribution 4.0 International License

OpenCL for Safety Critical Applications?

- OpenCL is increasingly desired in Safety Critical markets
- Some member companies are exploring OpenCL SC
- Possible substrate API for SYCL SC?

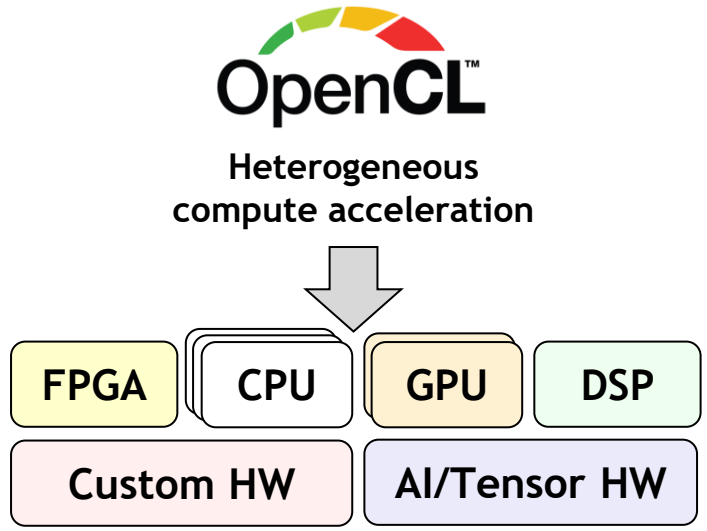
This work is licensed under a Creative Commons Attribution 4.0 International License

OpenCL for More Devices?

- **Key OpenCL Strength: Device Diversity**
  - Can we expand OpenCL support to even more devices?
  - Via custom devices? More profiles?

- **Expanding interest in OpenCL from RISC-V**

- **Expanding interest in OpenCL in China**



This work is licensed under a Creative Commons Attribution 4.0 International License

Summary and Wrap-Up

This work is licensed under a Creative Commons Attribution 4.0 International License

OpenCL State of the Union

- **OpenCL 3.1 raises the bar!**
  - Adds features that are widely supported with proven value: SPIR-V consumption, sub-groups, much more…
  - A significant step forward for the heterogeneous compute ecosystem!

- **Extension pipeline remains active:**
  - Continuing to innovate via extensions
  - Driven by mobile, embedded, desktop, HPC, and AI markets

- **OpenCL is a key part of a layered ecosystem:**
  - OpenCL is a popular substrate layer for higher-level models - especially SYCL
  - OpenCL is a popular usability layer over lower-level models – especially Vulkan

KHRONOS GROUP

This work is licensed under a Creative Commons Attribution 4.0 International License

Developer and Users:

Please Provide Your Feedback!

- **Give us your feedback on the OpenCL spec GitHub**
  - What could be added to the ecosystem to make you more productive?
  - What API and Language features do you most need?
  - https://github.com/KhronosGroup/OpenCL-Docs
- **Engage via the Khronos Discord Server**
  - https://www.khr.io/khrdiscord
- **Please download and run the GPUinfo OpenCL Hardware Capability Viewer**
  - https://opencl.gpuinfo.org/download.php

KHRONOS GROUP

This work is licensed under a Creative Commons Attribution 4.0 International License