IWOCL 2020
SYCLcon

The 8th International Workshop on OpenCL and the SYCL Developer Conference

KHRONOS GROUP

Live Webinar: **APRIL 28 (16:00 BST)**

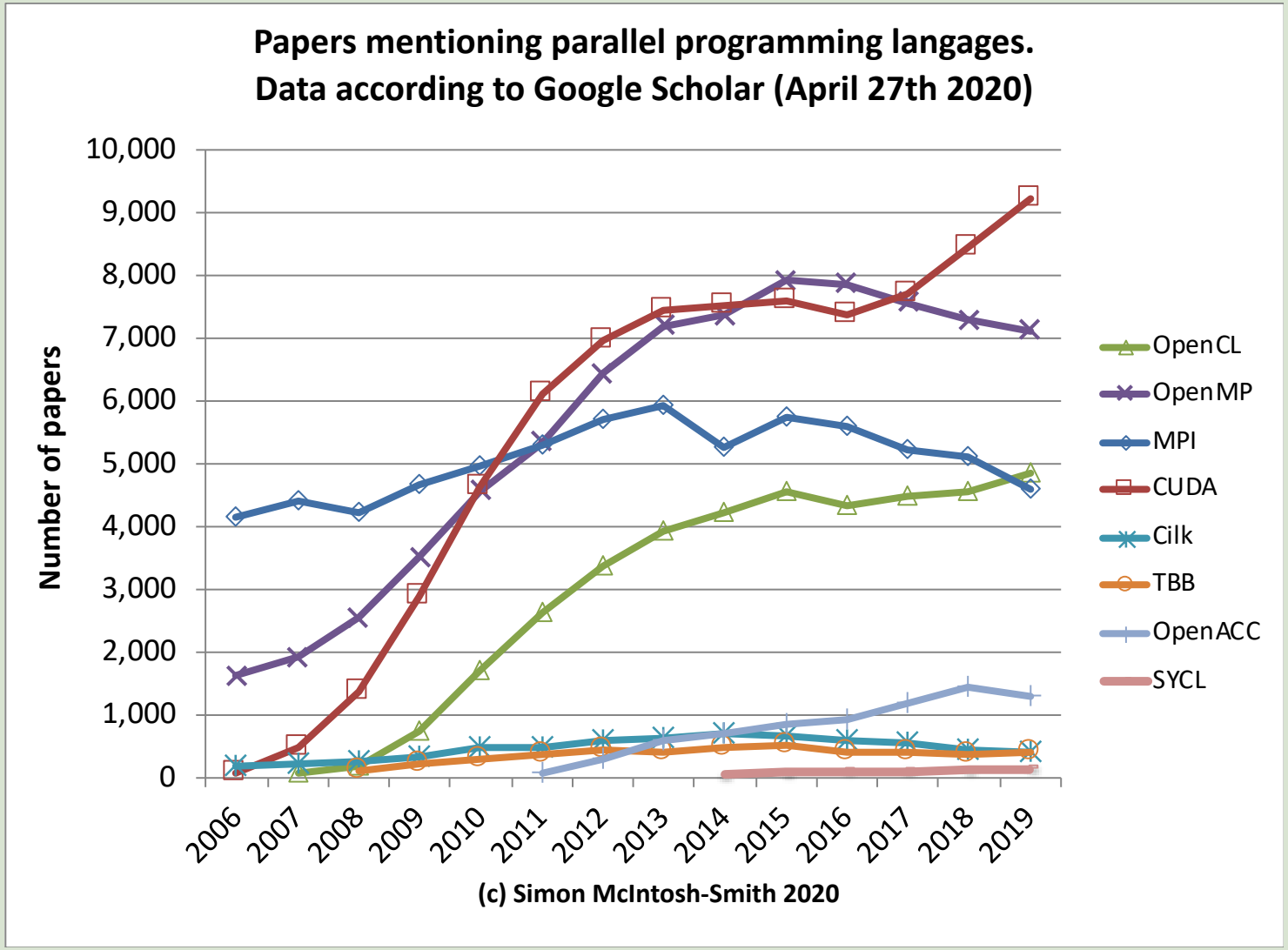
Register to Join: www.iwocl.org/iwocl-2020/conference-program/#panel

Thanks to our sponsors

KHRONOS GROUP intel codeplay University of BRISTOL TUM lrz SCIENTIFIC COMPUTING WORLD hgpu.org

IWOCL 2020
SYCLcon

Adoption of parallel programming languages



IWOCL 2020
SYCLcon

Agenda

- **Introduction** by Simon McIntosh-Smith, IWOCL / SYCLcon chair, University of Bristol

- **OpenCL state of the union and announcements** by Neil Trevett, VP at NVIDIA and Khronos President

- **SYCL status and updates** by Michael Wong, SYCL working group chair and VP of software at Codeplay

OpenCL 3.0

Neil Trevett
Khronos President
OpenCL Working Group Chair
NVIDIA VP Developer Ecosystems
ntrevett@nvidia.com | @neilt3d

April 2020

OpenCL is Widely Deployed and Used

The industry's most pervasive, cross-vendor, open standard for low-level heterogeneous parallel programming

Desktop Creative Apps

Adobe, otoy, RED DIGITAL CINEMA REDCINE-X PRO, DASSAULT SYSTEMES, SONY, Modo, darktable, AUTODESK, CHAOSGROUP, CyberLink, ptc, Vegas Pro, RADEON PRORENDER, REALFLOW, GIMP, Capture One, ArcSoft, Blackmagicdesign, SILHOUETTE, blender, LuxCoreRender OPEN SOURCE PHYSICALLY BASED RENDERER, SideFX, acdsee

Parallel Languages

SYCL, OpenACC DIRECTIVES FOR ACCELERATORS, aparapi, PyOpenCL

Linear Algebra Libraries

SYCL-BLAS, ViennaCL, CLBlast

Machine Learning Libraries and Frameworks

Intel OPENVINO™, Intel clDNN, Xiaomi MACE Mobile AI Compute Engine, SYCL-DNN, Caffe, NNAPI, Acuity ML/AI Software Foundation, VeriSilicon, Qualcomm Neural Processing SDK for AI, Synopsis MetaWare EV, TI DL Library (TIDL), Arm Compute Library

Molecular Modelling Libraries

siremol.org, OpenMM, CHARMM Chemistry at HARvard Macromolecular Mechanics, GROMACS FAST. FLEXIBLE. FREE., FOLDING @HOME, ForceBalance

Machine Learning Compilers

plaidML, tvm, Glow

Vision and Imaging Libraries

FAST, OpenCV, OpenVX, VisionCpp, Metashape, Halide, FFmpeg

Math and Physics Libraries

BULLET PHYSICS LIBRARY, GNU Octave, Wolfram Mathematica, ArrayFire C, C++, Fortran, Matlab

---

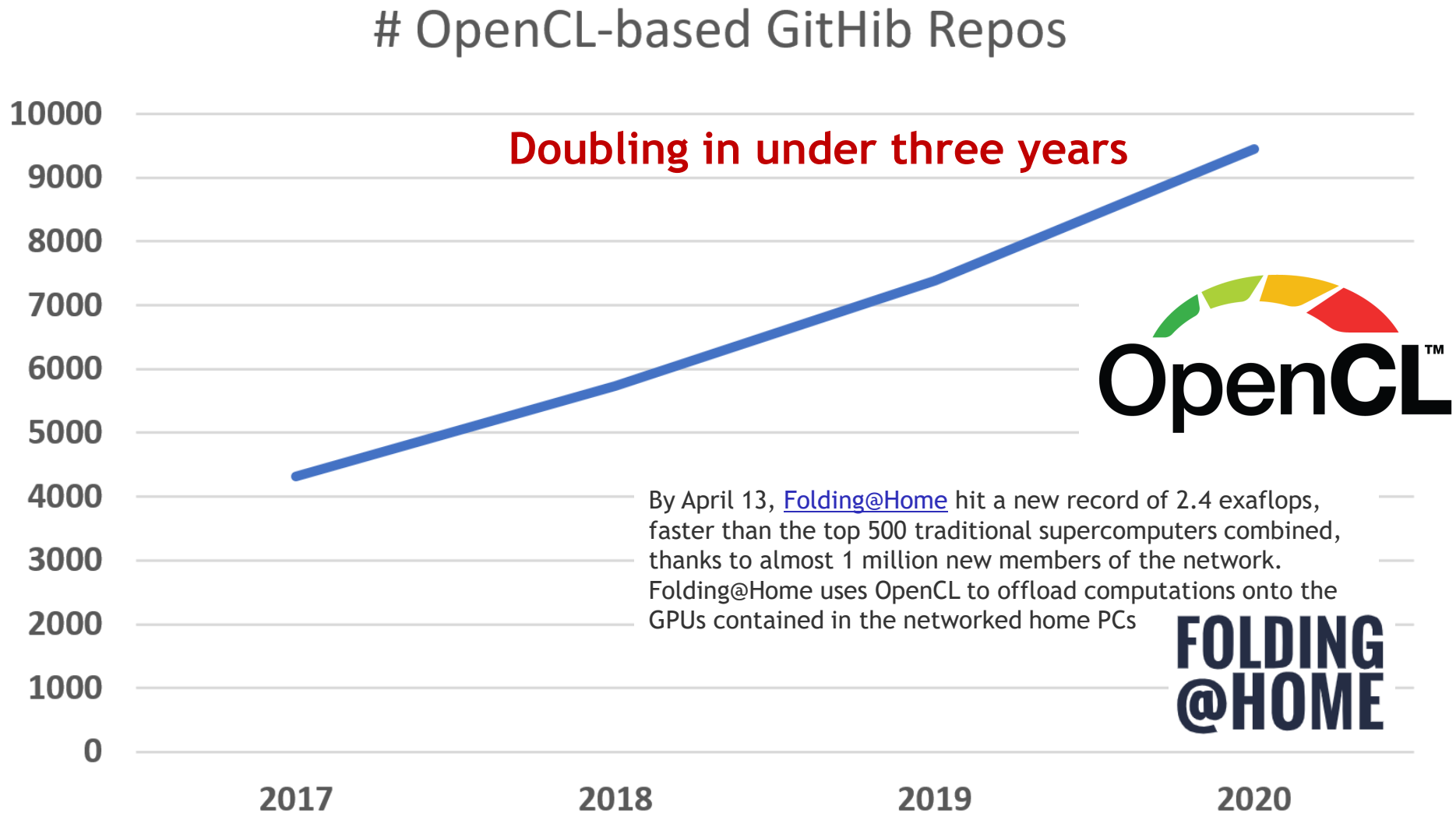
Apple, ALTERA, AMD, arm, OpenCL™, Imagination, intel, MARVELL, MEDIATEK, QUALCOMM, NVIDIA, SAMSUNG, ST, Texas Instruments, VeriSilicon, XILINX

Accelerated Implementations

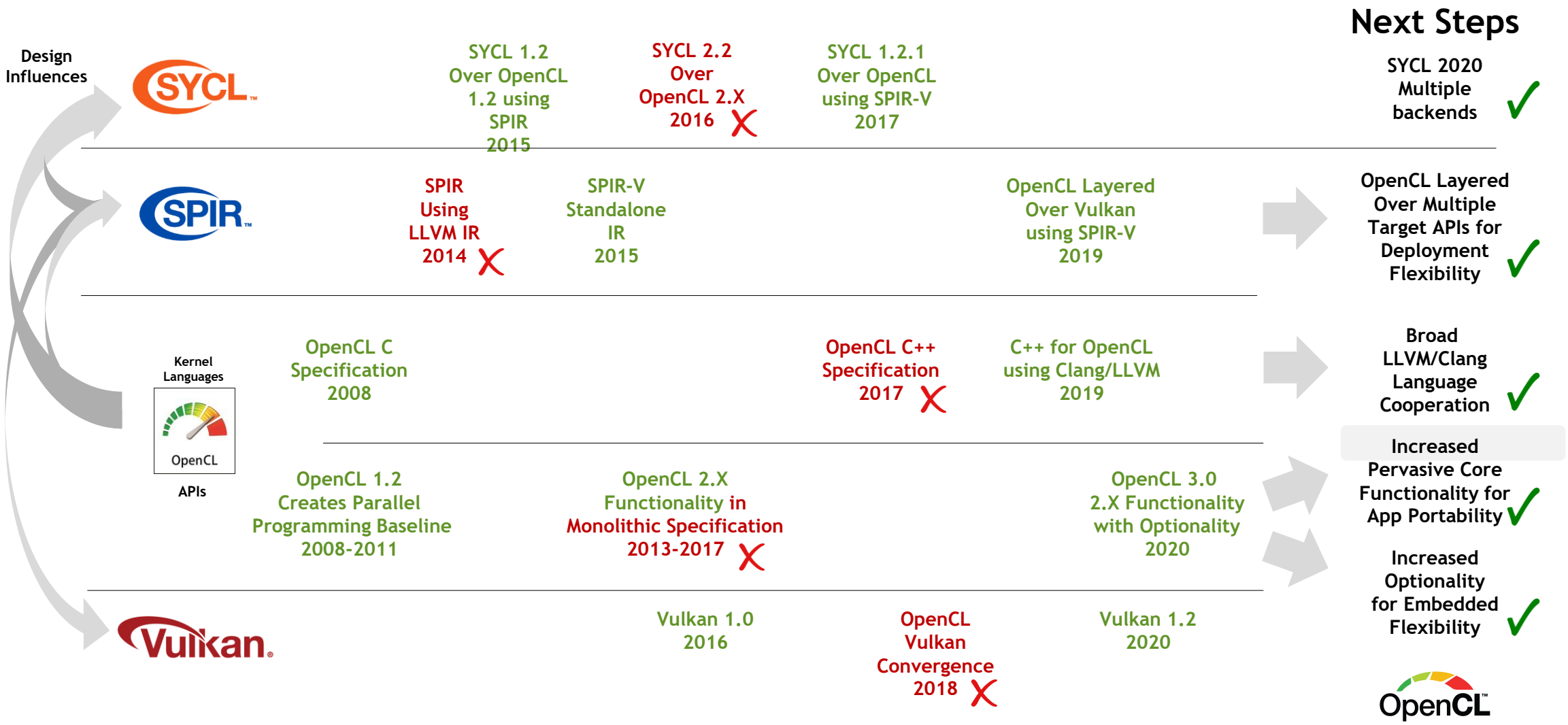
https://en.wikipedia.org/wiki/List_of_OpenCL_applications

KHRONOS GROUP

OpenCL Open Source Ecosystem Momentum



OpenCL Standards Evolution



OpenCL 3.0

Increased Deployment Flexibility

All functionality beyond OpenCL 1.2 is optional
Unified API specification slices OpenCL 2.X functionality into coherent, queryable, optionality
OpenCL C 3.0 language specification adds macros for optional language features

Subgroups with SPIR-V 1.3

New (optional) core functionality

Asynchronous DMA extension

Enabling a new class of Embedded Processors

OpenCL C++ not included

Ecosystem has transitioned to open source C++ for OpenCL

Easy for Developers to upgrade to OpenCL 3.0

NO code changes necessary if all used functionally is present
Applications encouraged to query used OpenCL 2.X functionality for future portability

Easy for Implementers to upgrade to OpenCL 3.0

Add queries for OpenCL 2.X functionality - missing or present
Update reported version and add minor entry points for improved app portability

OpenCL 3.0
April 2020

KHRONOS GROUP

OpenCL Roadmap

C++ for OpenCL
Open source C++ kernel language front-end leveraging Clang and LLVM

Regular Maintenance Updates
Clarifications, formatting, bug fixes

Unified API Specification
All OpenCL versions documented in one place
Tightly organized queries for all 2.X functionality
OpenCL C 3.0 Language - macros for optional features

Subgroups and SPIR-V 1.3
New (optional) core functionality

Asynchronous DMA extension
Enabling a new class of Embedded Processors

OpenCL 3.0
April 2020

Extension Pipeline
Extended Subgroups
Device UUID Query
Extended Debug Info
External Memory Sharing
Vulkan/OpenCL Interop
Recordable Command buffers?
Machine Learning Primitives?
Indirect Dispatch?
Device Topology?

Khronos OpenCL SDK
Headers, Utility Libraries, Documentation, Samples, ICD Loader

Open Source Ecosystem
Tools, Domain Libraries

SPIR-V 1.4/1.5 ingestion
Compiler efficiency and expressiveness

Regular Maintenance Updates
Clarifications, formatting, bug fixes

New Pervasive Functionality in Core Specification
Integrate proven, widely adopted extensions

Flexible Profile
Finer-grain optional functionality for embedded processors

'Layering' Profile?
Defined queries and conformance for layered implementations?

'OpenCL Next'
Faster core release cycle

Time

API Layering

Enabled by growing robustness of open source compiler ecosystem

Layers Over	Vulkan	OpenGL	OpenCL	OpenGL ES	DX12	DX9-11
Vulkan		Zink	clspv clvk	GLOVE Angle	vkd3d	DXVK WineD3D
OpenGL	gfx-rs Ashes			Angle		WineD3D
DX12	gfx-rs	Microsoft 'GLOn12'	Microsoft 'CLOn12'			Microsoft D3D11On12
DX9-11	gfx-rs Ashes			Angle		
Metal	MoltenVK gfx-rs		clspv + SPIRV-Cross?	MoltenGL Angle		

ROWS Benefit Platforms by adding APIs

Enable content without additional kernel level drivers

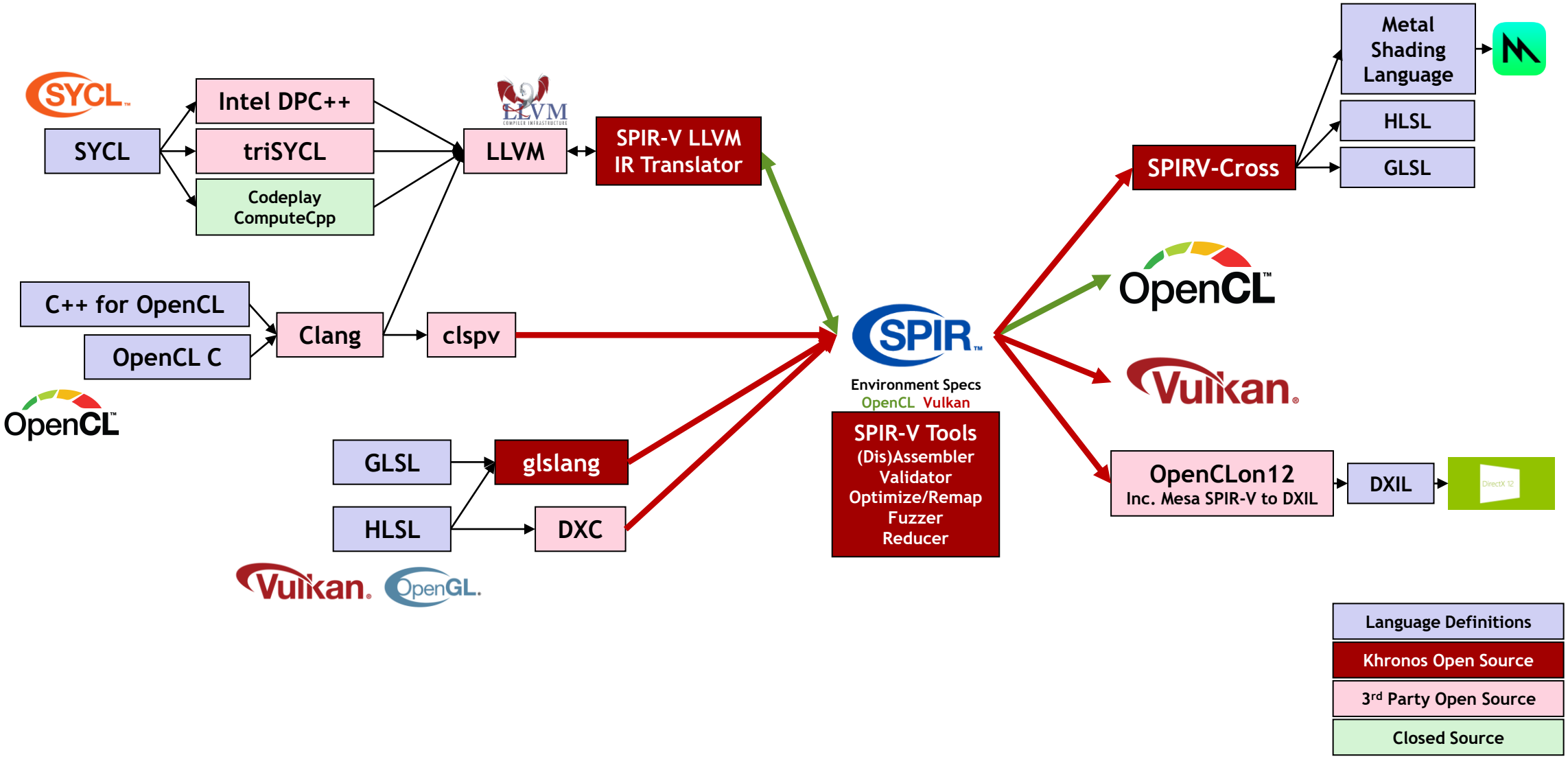
Vulkan PORTABILITY

COLUMNS Benefit ISVs by making an API available everywhere

Application deployment flexibility by fighting platform fragmentation

Making an API available across multiple platforms even if no native drivers available

SPIR-V Language Ecosystem



Developers - Please Give Us Feedback!

- **Is the set of optional features sliced too finely, or too coarsely?**
  - Are they easy to understand?
- **Which optional features do you expect to use in your application or library?**
  - Usage data drives which optional features should be made mandatory in future
- **What new features do you most need?**
  - We will use extensions to prove new functionality before adding to core specification
  - What extensions would you like to see in the second half of 2020?

OpenCL Working Group has maximized information in Khronos public GitHub to accelerate finalization

Provisional OpenCL 3.0 Specification sources released on GitHub
https://www.khronos.org/registry/OpenCL/

Spec feedback and pull requests welcome on GitHub
https://github.com/KhronosGroup/OpenCL-Docs/issues

OpenCL 3.0 Conformance Tests WIP sources released on GitHub
https://github.com/KhronosGroup/OpenCL-CTS

Tests feedback and pull requests welcome on GitHub
https://github.com/KhronosGroup/OpenCL-CTS/issues

Vendor OpenCL 3.0 Implementations in flight

OpenCL™

Urgency to Finalize and Ship
Finalized OpenCL 3.0 Specifications
Completed Conformance Tests
Multiple Shipping Conformant Implementations

SYCL WG
State of the Union 2020

Michael Wong
SYCL WG Chair
Codeplay VP of R&D
ISOCPP Director & VP
ISO C++ Directions Group Chair
michael@codeplay.com | wongmichael.com/about

SYCL Present and Future Roadmap (May Change)

C++11 | C++14 | C++17 | C++20 | C++23

SYCL 1.2
C++11 Single source programming

SYCL 1.2.1
C++11 Single source programming

SYCL 2020
C++17 Single source programming
Many backend options

SYCL 2021-?
C++20 Single source programming
Many backend options

OpenCL

2011

OpenCL 1.2
OpenCL C Kernel Language

OpenCL

2015

OpenCL 2.1
SPIR-V in Core

SPIR

OpenCL

2017

OpenCL 2.2

SPIR

OpenCL

2020

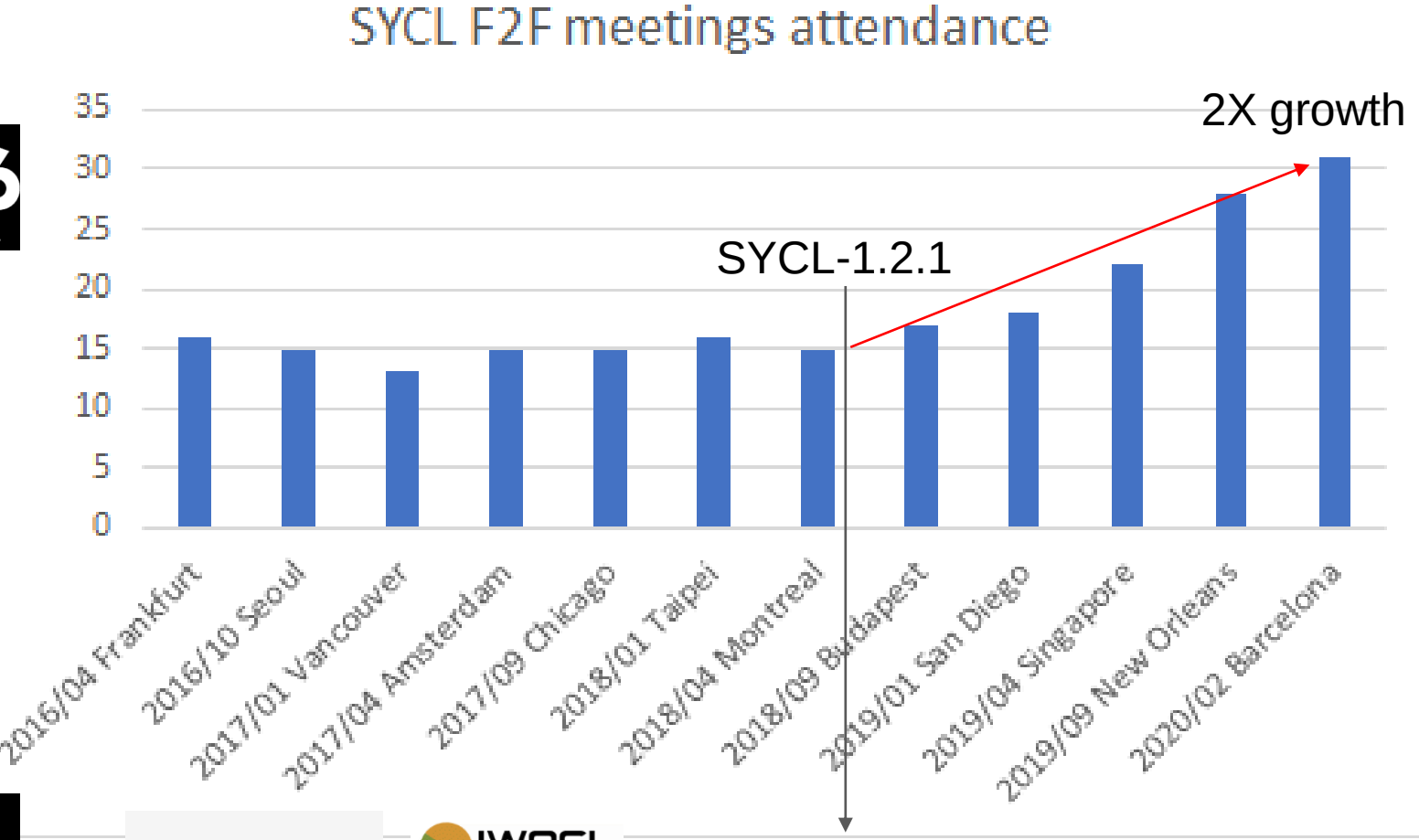
OpenCL 3.0

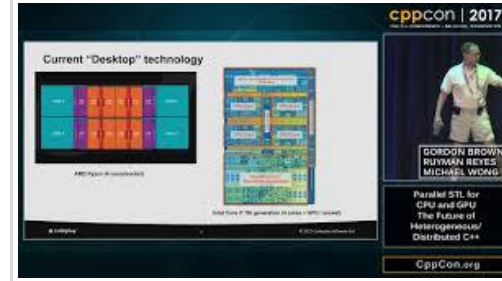
SPIR

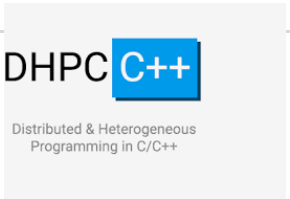
2021-????

KHRONOS GROUP

SYCL community is vibrant







SYCL Evolution

SYCL 2020 Potential Features

Generalization (a.k.a the Backend Model) presented by Gordon Brown
Unified Shared Memory (USM) presented by James Brodman
Improvement to Program class Modules presented by Gordon Brown
Host Task with Interop presented by Gordon Brown
In order queues, presented by James Brodman

SYCL 2020 compared with SYCL 1.2.1

Easier to integrate with C++17 (CTAD, Deduction Guides...)
Less verbose, smaller code size, simplify patterns
Backend independent
Multiple object archives aka modules simplify interoperability
Ease porting C++ applications to SYCL
Enable capabilities to improve programmability
Backwards compatible but minor API break based on user feedback

Integration of successful Extensions plus new Core functionality

Converge SYCL with ISO C++ and continue to support OpenCL to deploy on more devices

CPU
GPU
FPGA
AI processors
Custom Processors

SYCL 2020 Roadmap (WIP, MAY CHANGE)

2017
SYCL 1.2.1

Target 2020
Provisional Q3 then Final Q4

Improving Software Ecosystem

Tool, libraries, GitHub

Expanding Implementation

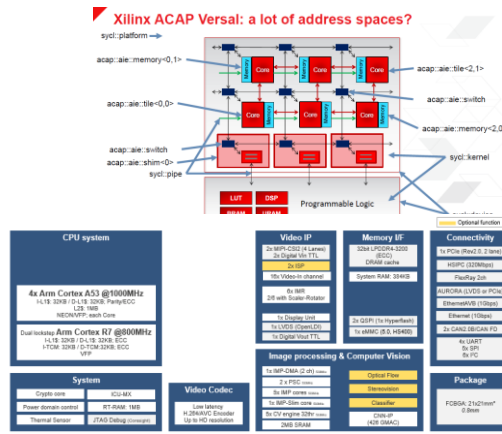
DPC++
ComputeCpp
triSYCL
hipSYCL

Regular Maintenance Updates

Spec clarifications, formatting and bug fixes
https://www.khronos.org/registry/SYCL/

Selected Extension Pipeline aiming for SYCL 2020 Provisional Q3

Reduction
Subgroups
Accessor simplification
Atomic rework
Extension mechanism
Address spaces
Vector rework
Specialization Constants



Repeat The Cycle every 1.5-3 years

SYCL Ecosystem, Research and Benchmarks

Implementations

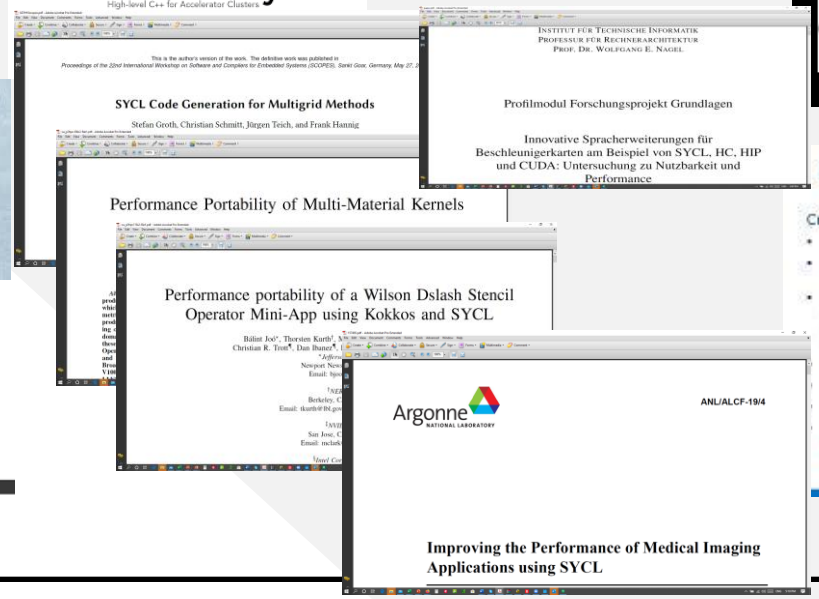
oneAPI

ComputeCpp

hipSYCL

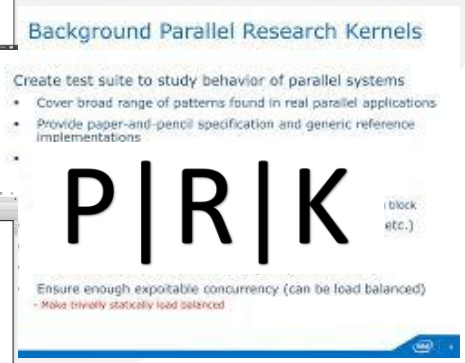
Research

Celerity
High-level C++ for Accelerator Clusters



ECP
EXASCALE COMPUTING PROJECT

Benchmarks



RSBench

Linear Algebra Libraries

SYCL-BLAS

Eigen

Machine Learning Libraries and Parallel Acceleration Frameworks

SYCL-ML

TensorFlow

SYCL-DNN

oneMKL

SYCL Parallel STL

STREAM HPC

Argonne NATIONAL LABORATORY

QUALCOMM

arm

SYCL™

intel

XILINX

University of BRISTOL

AMD

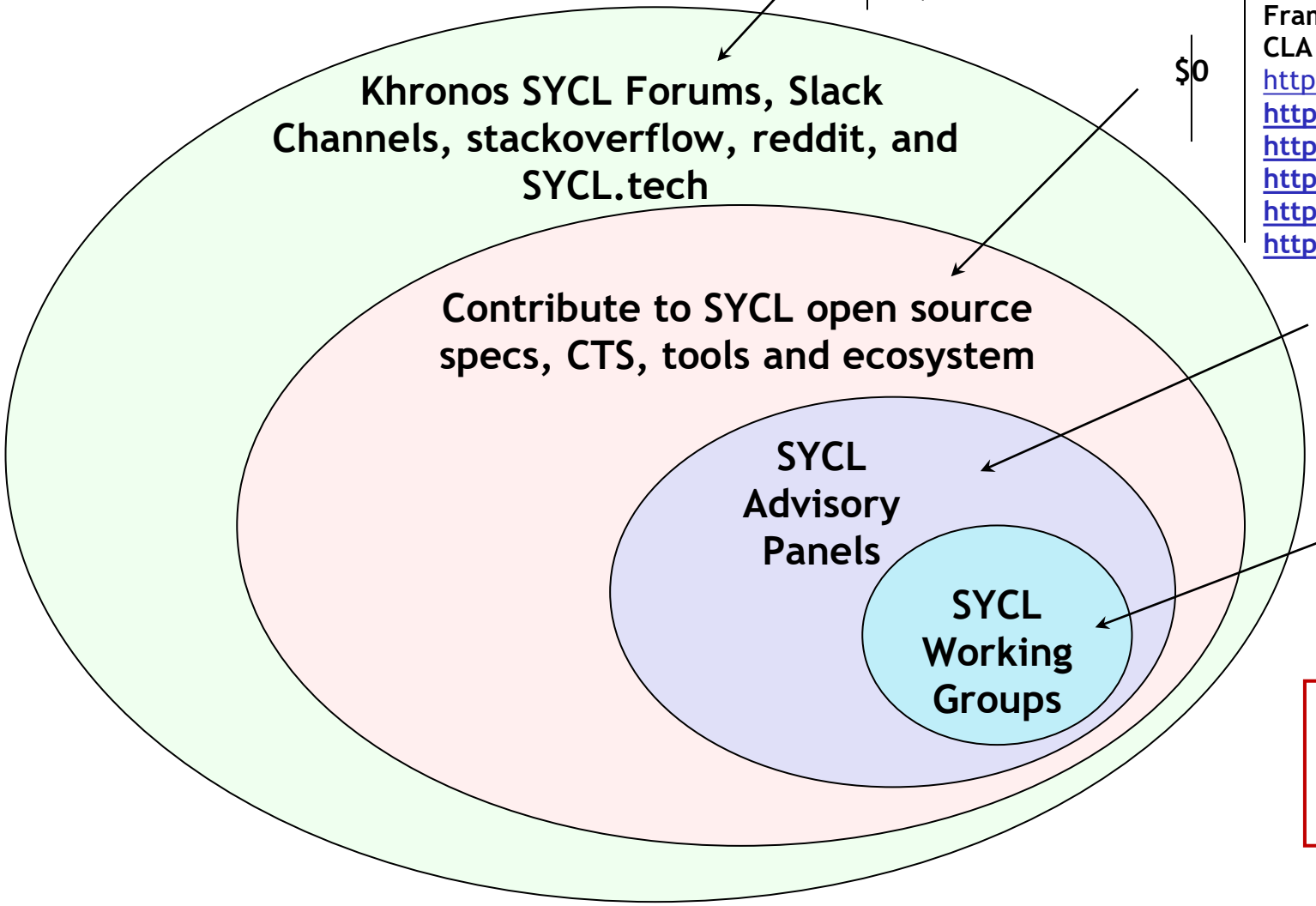
codeplay

Active Working Group Members

SYCL 2020 Provisional is coming

- In a few months, SYCL 2020 provisional will be released
- We need your feedback asap
  - https://app.slack.com/client/TDMDFS87M/CE9UX4CHG
  - https://community.khronos.org/c/sycl
  - https://sycl.tech
- What features are you looking for in SYCL 2020?
- What feature would you like to aim for in future SYCL?
- How do you join SYCL?

Engaging with the Khronos SYCL Ecosystem



$0

Open to all!

https://community.khronos.org/www.khr.io/slack
https://app.slack.com/client/TDMDFS87M/CE9UX4CHG
https://community.khronos.org/c/sycl/
https://stackoverflow.com/questions/tagged/sycl
https://www.reddit.com/r/sycl
https://github.com/codeplaysoftware/syclacademy
https://sycl.tech/

$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
https://github.com/KhronosGroup/SYCL-CTS
https://github.com/KhronosGroup/SYCL-Docs
https://github.com/KhronosGroup/SYCL-Shared
https://github.com/KhronosGroup/SYCL-Registry
https://github.com/KhronosGroup/SyclParallelSTL

$0

Invited Advisors under the Khronos NDA and IP Framework can comment and contribute to requirements and draft specifications

https://www.khronos.org/advisors/

$

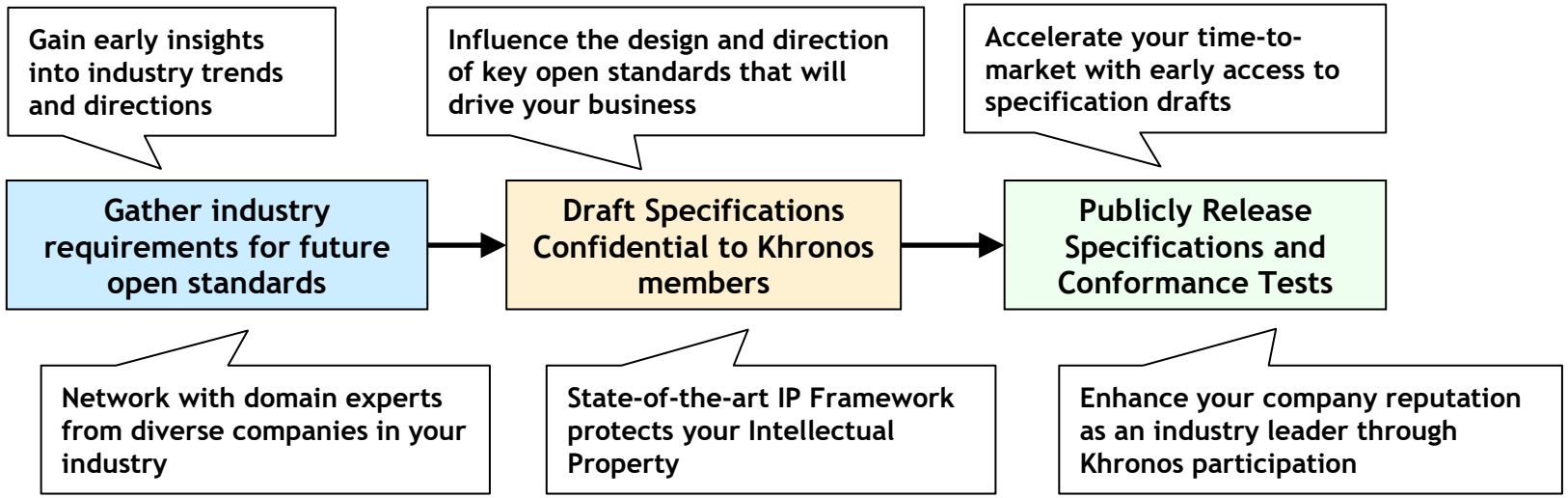
Khronos members under Khronos NDA and IP Framework participate and vote in working group meetings. Starts at $3.5K/yr.

https://www.khronos.org/members/
https://www.khronos.org/registry/SYCL/

Any member *or* non-member can propose a new SYCL feature or fix

Thank You!

- **Khronos SYCL is creating cutting-edge royalty-free open standard**
  - For C++ Heterogeneous compute, vision, inferencing acceleration
- **Information on Khronos SYCL Standards:** https://www.khronos.org/sycl
- **Any entity/individual is welcome to join Khronos SYCL:** https://www.khronos.org/members
- **Join the SYCLCon Tutorial Monday and Wednesday Live panel** : Wednesday Apr 29 15:00-18:00 GMT
  - Have your questions answered live by a group of SYCL experts
- **Michael Wong: michael@codeplay.com | wongmichael.com/about**



Benefits of Khronos membership

IWOCL 2020
SYCLcon

KHRONOS GROUP

Panel Discussion & Announcements

Live Webinar: **APRIL 28 (16:00 BST)**

Register to Join: www.iwocl.org/iwocl-2020/conference-program/#panel

Panel Chair and Announcements by:

Simon McIntosh-Smith
University of Bristol
Conference Chair

Neil Trevett
OpenCL Working Group Chair
Khronos President, VP NVIDIA

Michael Wong
SYCL Working Group Chair
VP of R&D, Codeplay

Alastair Murray
Codeplay
Principal SW Eng. Compilers

Ben Ashbaugh
Intel
Principal Engineer

Dennis Adams
Sony Creative Software
Director of Technology

Eric Berdahl
Adobe
Senior Engineering Manager

Hal Finkel
Argonne National Laboratory
Lead for Compiler Technology

Jeremy Kemp
Imagination
Snr. Software Design Engineer

Kévin Petit
Arm
Principal Software Architect

Martin Schreiber
Technical University of Munich
Researcher, IWOCL Local Chair

Ronan Keryell
Xilinx
Principal Software Engineer