KHRONOS GROUP

Vulkan Launch Briefing

February 2016

Neil Trevett | Khronos President
NVIDIA Vice President Developer Ecosystem
ntrevett@nvidia.com | @neilt3d

BOARD OF PROMOTERS

KHRONOS GROUP

Over 100 members worldwide
any company is welcome to join

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

Software

Silicon

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

Low-level silicon APIs
needed on almost every platform:
graphics, parallel compute,
rich media, vision, sensor
and camera processing

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 GROUP

The Need for New Generation GPU APIs

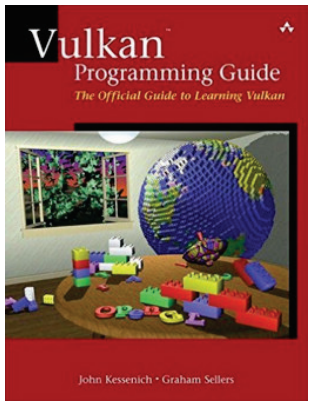
OpenGL has evolved over 25 years - API complexity can obscure optimal performance path and hinder portability

GPUs are increasingly compute AND graphics capable + platforms are becoming unified and multi-core

GPUs accelerate graphics, compute, vision and deep learning across diverse platforms: PORTABILITY is key

Vulkan 1.0 Launched!

- **Khronos' first API 'hard launch'**
  - Specification, Conformance Tests, SDKs - all in open source
  - Reference Materials, Compiler front-ends, Samples…
- **Conformant Drivers from multiple hardware vendors**
  - Across multiple OS
- **LunarG SDK for Vulkan on Windows and Linux (Android soon)**
  - Free and open source
- **A Vulkan game on Steam: Talos Principle Vulkan back-end in beta**



For preorder on Amazon

Image Courtesy Croteam

Image Courtesy NVIDIA

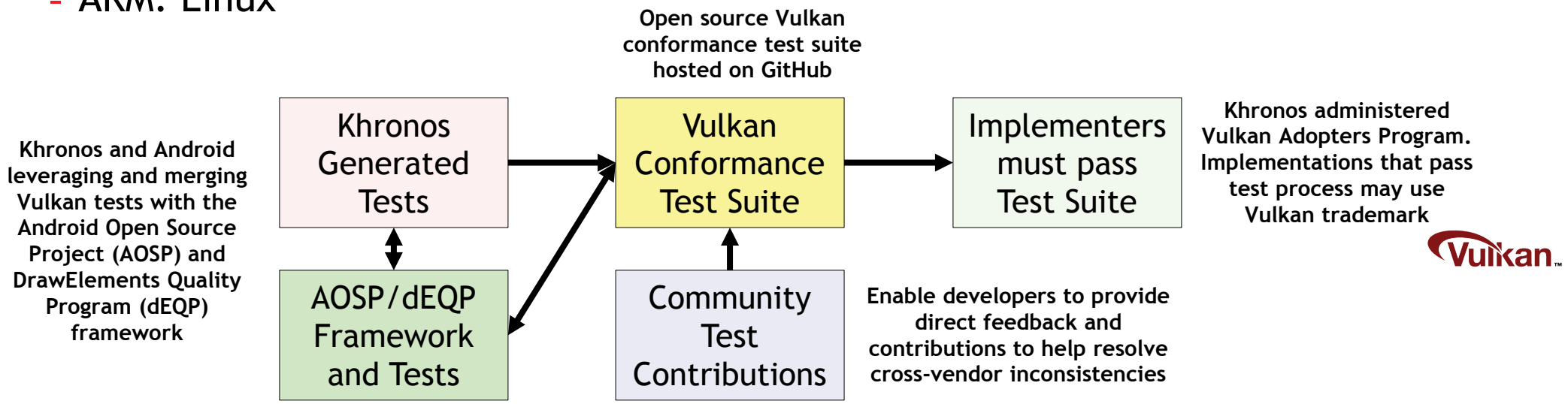
Vulkan Working Group

- **Vulkan 1.0 specification and implementations created in 18 months**
  - More energy and participation than any other API in Khronos history
- **Significant proposals and IP contributions received from members**
  - A true working group effort
- **Participants come from all segments of the graphics industry**
  - Including an unprecedented level of participation from game engine ISVs

Working Group Participants

Conformant Vulkan Drivers at Launch

- **30 Driver submissions passed conformance at Vulkan 1.0 launch**
  - Imagination Technologies: Linux
  - Intel: Linux
  - NVIDIA: Android 6.0, Linux (desktop and embedded), Windows 7-10
  - Qualcomm: Android 6.0
- **Drivers in test Submission review at Vulkan 1.0 launch**
  - AMD: Windows
  - ARM: Linux



Vulkan Developer Resources at Launch

KHRONOS GROUP

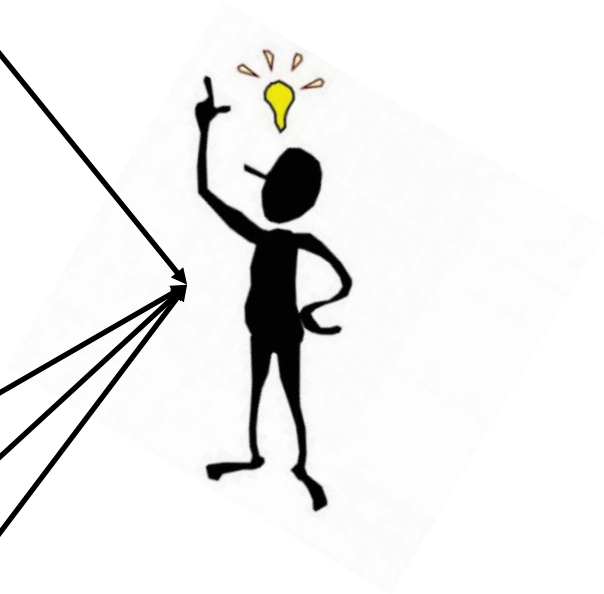
Khronos.org
(open source resources in github.com/KhronosGroup)
Specifications Source
Header Source
Feature Set Definitions
(Windows and Linux - post developer feedback)
Quick Reference
Reference Pages (Vulkan and WSI)
Conformance Test Source and Test Process
Compiler toolchain sources
Validation Layer Source
Loader Source
Layers and Loader documentation
Community Contributions

Everything needed to create SDKs for any platform or market

LunarG
Windows and Linux Installable SDKs
Loader and Validation Layer binaries
Tools Layers - source and binaries
Samples - source and binaries
Windows get started guide

IHV Websites
Drivers and Loader
Vendor tools and layers

Third Party Websites
Layers, Samples etc.



KHRONOS GROUP

LunarG SDK for Vulkan

- **Valve sponsored LunarG to develop a free, open source SDK for Vulkan**
  - Utilities, samples, debugging tools, documentation
  - For Windows and Linux on launch – Android coming soon
- **Validation Layer – checks many aspects of Vulkan code:**
  - Device limits, draw state, parameter values
  - Multi-thread object access rules, texture and render target formats
  - Object Tracker, Memory Tracker
- **Other SDK Tools**
  - Trace and replay tools
  - GLSL Validator
  - SPIR-V Disassembler and Assembler
- **RenderDoc Graphics debugger**
  - Free and open source
  - Adding Vulkan support
  - https://github.com/baldurk/renderdoc

vulkan.lunarg.com

Vulkan Ecosystem Already Active

Molten

THE BRENWILL WORKSHOP
Graphics Technology Expertise

MetalVK MetalGL

Vulkan and OpenGL ES over Metal - in development

"By building your application or game using the Vulkan API, you can run your modern graphics application or game unchanged across an entire industry of platforms and development tools"
Brenwill Workshop

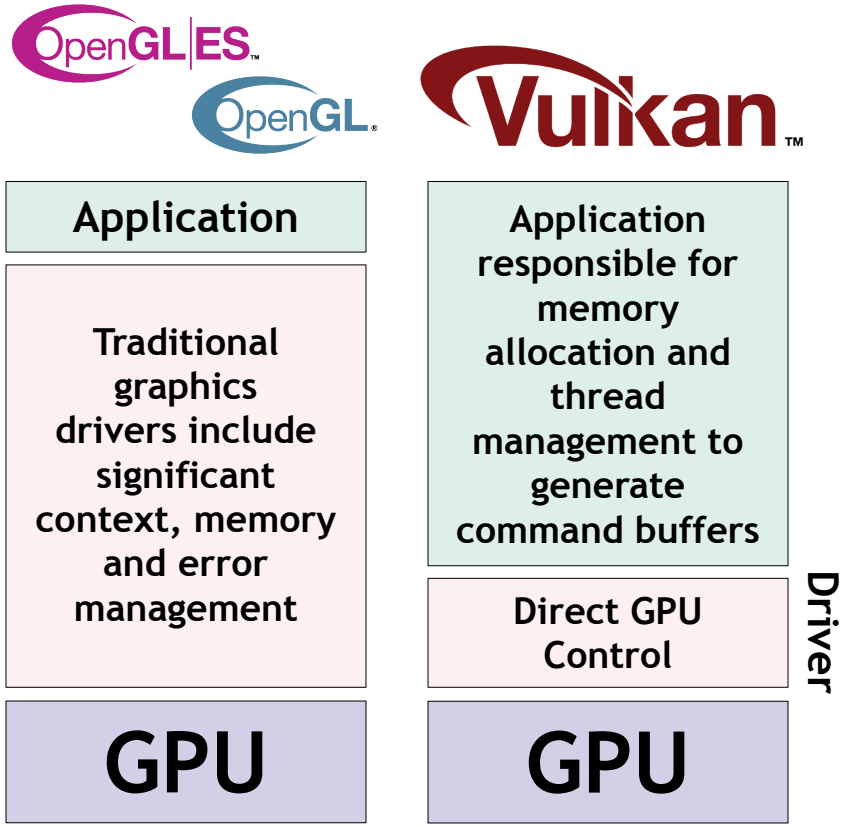
"Vulkan has a huge potential! We're only scratching the surface of what can be done with it, and porting The Talos Principle to Vulkan should be seen as a proof of concept," said Dean Sekulic graphics engine specialist at Croteam. "Vulkan in just one sentence? The endless war between performance and portability is finally over!"

Talos Principle on Steam has beta Vulkan back-end

CROTEAM

KHRONOS GROUP

Vulkan Explicit GPU Control



Complex drivers lead to driver overhead and cross vendor unpredictability

Error management is always active

Driver compiles full shading language source

Simpler drivers for low-overhead efficiency and cross vendor consistency

Layered architecture so validation and debug layers can be loaded only when needed

Run-time only has to ingest SPIR-V intermediate language

What Developers have been asking for…

… at least developers that need and can benefit from explicit control over GPU operation

Leading Edge Graphics Functionality
Equivalent to OpenGL in V1.0

Same API for mobile, desktop, console and embedded
Defined 'Feature Sets' per platform
No need for 'Vulkan ES'

General Purpose Compute
Graphics AND compute queues in a single API

Explicit API
Direct control over GPU and memory allocation for less hitches and surprises

Efficient Multi-threading
Use multiple CPU cores to create command buffers in parallel

Precompiled Shaders
SPIR-V for shading language flexibility including C++ Programming (future)

Clean, Streamlined API
Easier to program, implement and test for cross-vendor consistency

Low Driver Overhead
Thinner, simpler driver reduces CPU bottlenecks and latency

FUNCTIONALITY

PERFORMANCE

PORTABILITY

KHRONOS GROUP

Next Generation GPU APIs

Only Windows 10

Only Apple

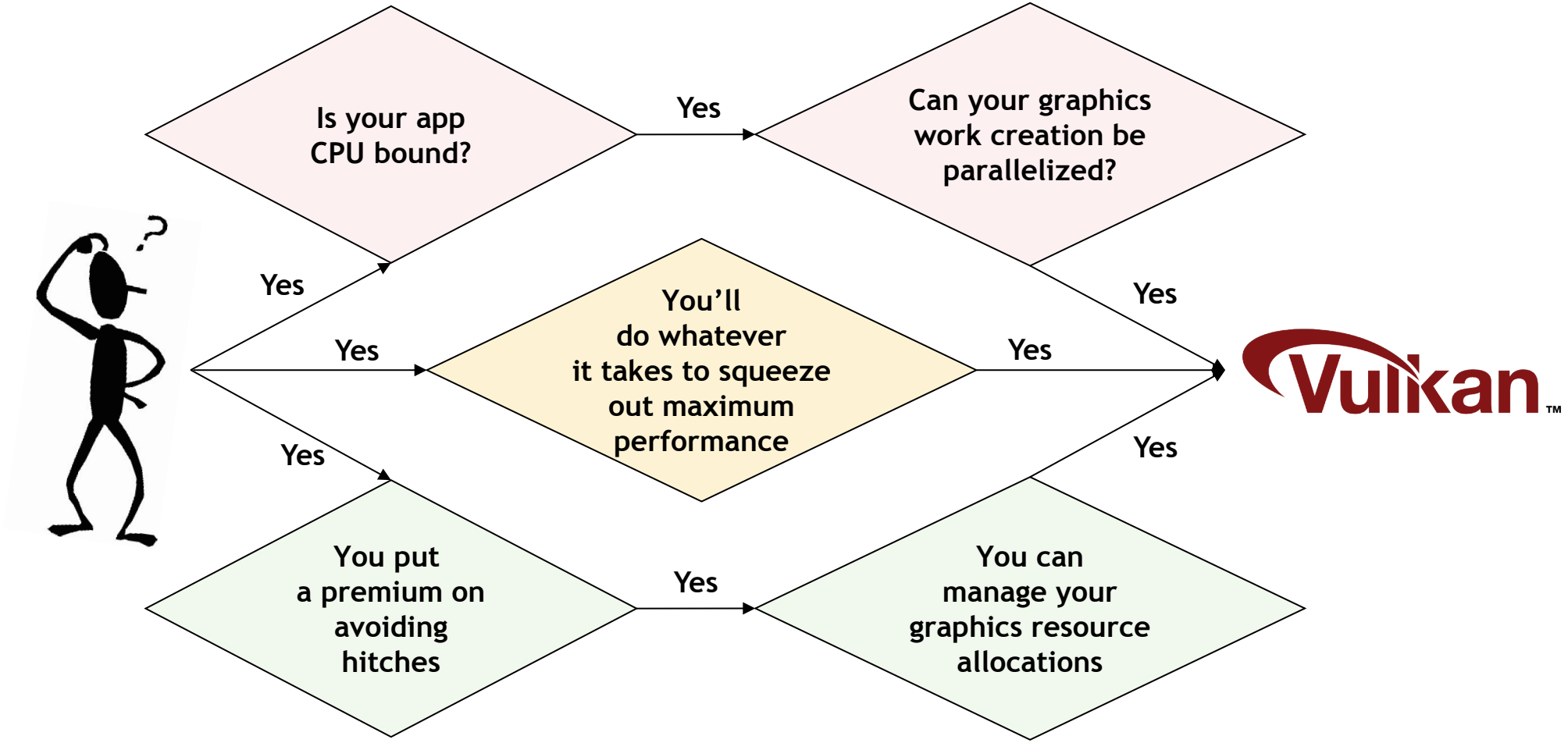
Cross Platform

Any OpenGL ES 3.1/4.X GPU

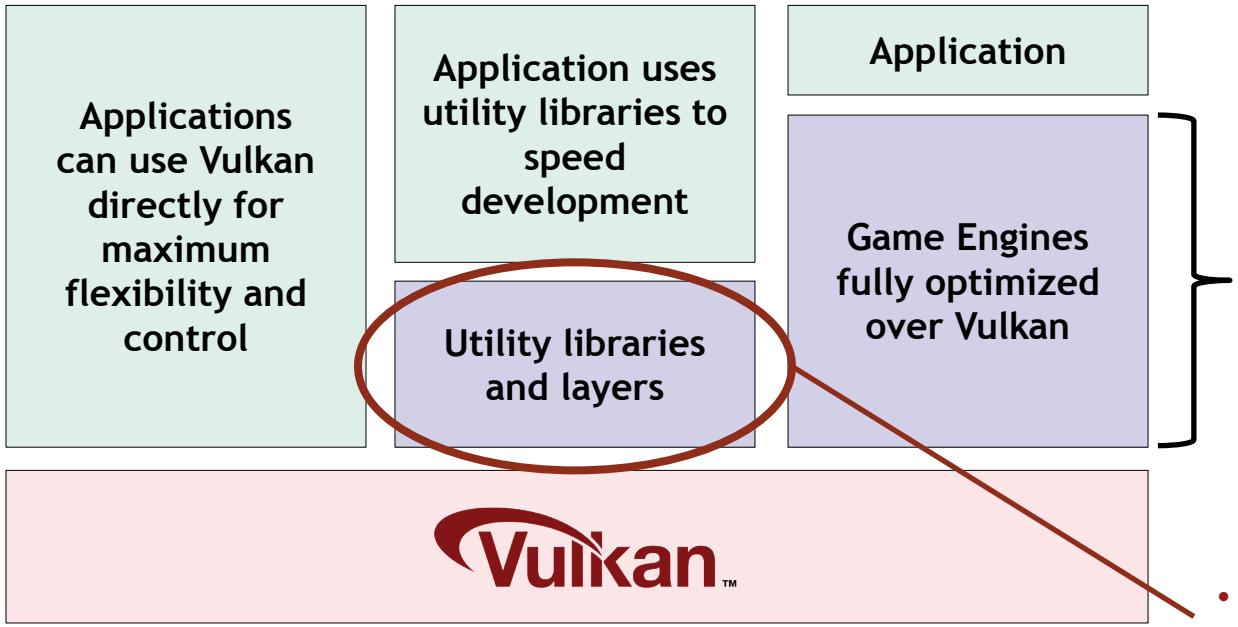
Vulkan - No Compromise Performance

Which Developers Should Use Vulkan?

- For many developers OpenGL and OpenGL ES will remain the most effective API



The Power of a Three Layer Ecosystem



Applications using game engines will automatically benefit from Vulkan's enhanced performance

OXIDE · LUCASFILM Ltd · BLIZZARD ENTERTAINMENT · EA · unity · EPIC GAMES · VALVE

Rich Area for Innovation

- Many utilities and layers will be in open source
- Layers to ease transition from OpenGL
- Domain specific flexibility

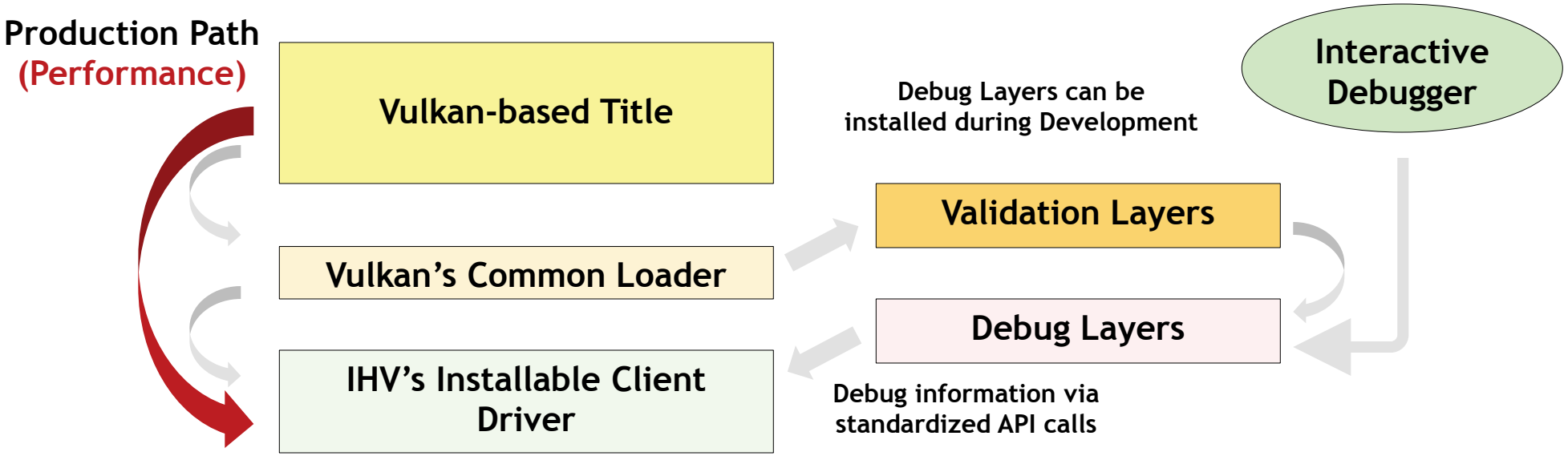
The same ecosystem dynamic as WebGL

A widely pervasive, powerful, flexible foundation layer enables diverse middleware tools and libraries

KHRONOS GROUP

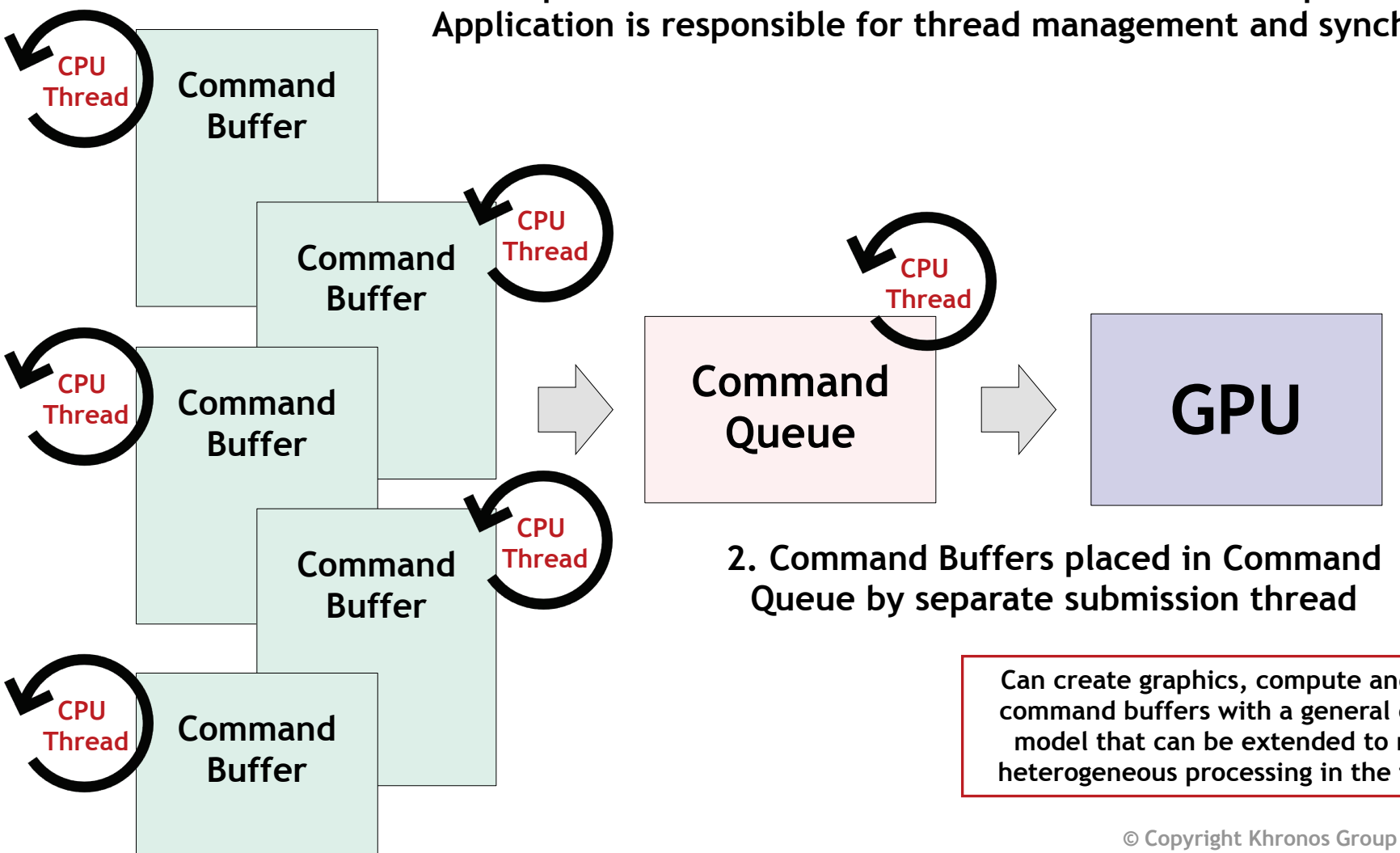
Vulkan Tools Architecture

- **Layered design for cross-vendor tools innovation and flexibility**
  - IHVs plug into a common, extensible architecture for code validation, debugging and profiling during development without impacting production performance
- **Khronos Open Source Loader enables use of tools layers during debug**
  - Finds and loads drivers, dispatches API calls to correct driver and layers



Vulkan Multi-threading Efficiency

1. Multiple threads can construct Command Buffers in parallel
Application is responsible for thread management and synch



2. Command Buffers placed in Command Queue by separate submission thread

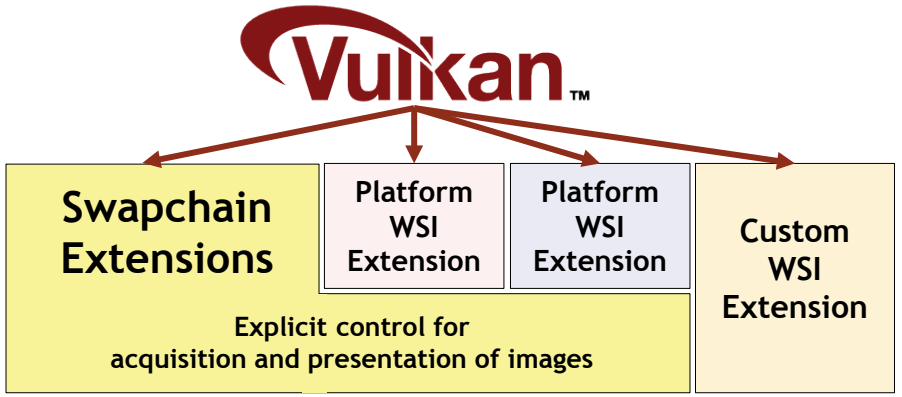
Can create graphics, compute and DMA command buffers with a general queue model that can be extended to more heterogeneous processing in the future

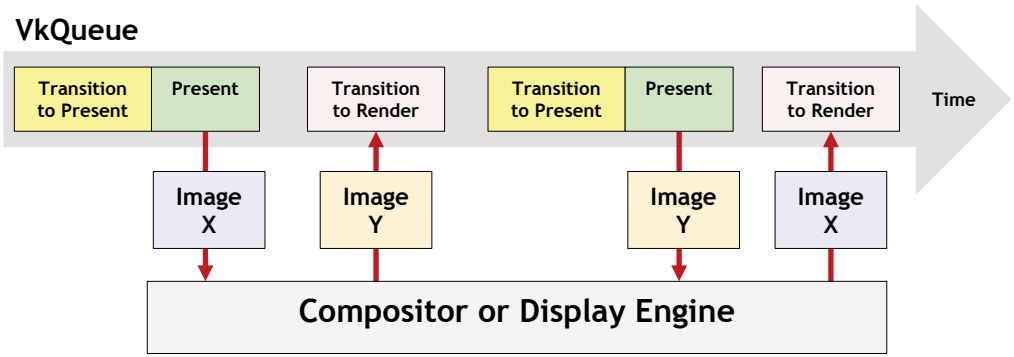
Vulkan Feature Sets

- **Vulkan supports hardware with a wide range of hardware capabilities**
  - Mobile OpenGL ES 3.1 up to desktop OpenGL 4.5 and beyond
- **One unified API framework for desktop, mobile, console, and embedded**
  - No "Vulkan ES" or "Vulkan Desktop"
- **Vulkan precisely defines a set of "fine-grained features"**
  - Features are specifically enabled at device creation time (similar to extensions)
- **Platform owners define a Feature Set for their platform**
  - Vulkan provides the mechanism but does not mandate policy
  - Khronos will define Feature Sets for platforms where owner is not engaged
- **Khronos will define feature sets for Windows and Linux**
  - After initial developer feedback

Vulkan Window System Integration (WSI)

- **Explicit control for acquisition and presentation of images**
  - Designed to fit the Vulkan API and today's compositing window systems
  - Cleanly separates device creation from window system
- **Platform provides an array of persistent presentable images = Vulkan Swapchain**
  - Device exposes which queues support presentation
  - Application explicitly controls which image to render and present
- **Standardized extensions - unified API for multiple window systems**
  - Works across Android, Mir, Windows (Vista and up), Wayland and X (with DRI3)
  - Platforms can extend functionality, define custom WSI stack, or have no display at all





SPIR-V Transforms the Language Ecosystem

- **First multi-API, intermediate language for parallel compute and graphics**
  - Native representation for Vulkan shader and OpenCL kernel source languages
  - https://www.khronos.org/registry/spir-v/papers/WhitePaper.pdf
- **Cross vendor intermediate representation**
  - Language front-ends can easily access multiple hardware run-times
  - Acceleration hardware can leverage multiple language front-ends
  - Encourages tools for program analysis and optimization in SPIR form

Multiple Developer Advantages

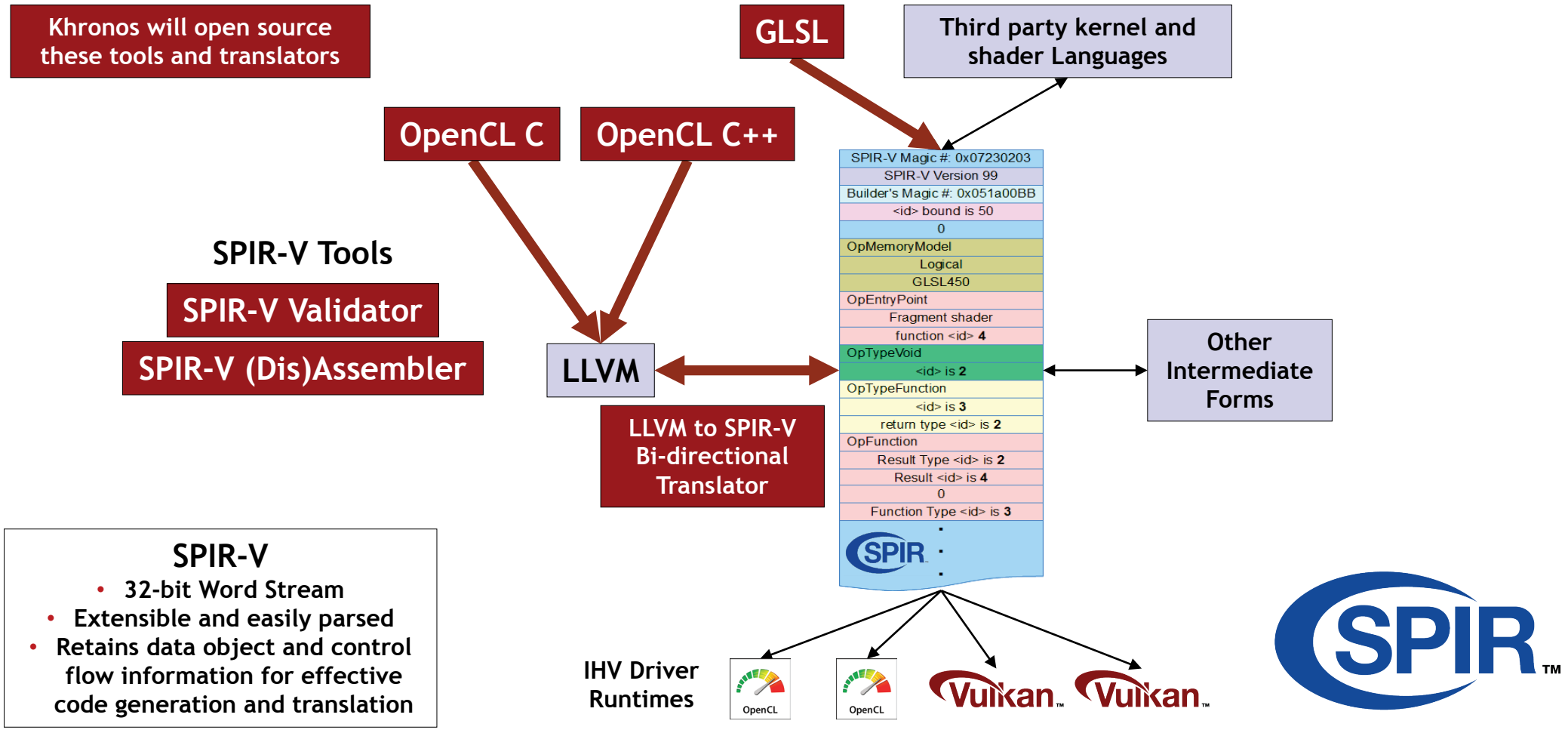
Same front-end compiler for multiple platforms
Reduces runtime kernel compilation time
Don't have to ship shader/kernel source code
Drivers are simpler and more reliable

Evolution of SPIR Family

- **SPIR-V is first fully specified Khronos-defined SPIR standard**
  - Does not use LLVM to isolate from LLVM roadmap changes
  - Includes full flow control, graphics and parallel constructs beyond LLVM
  - Khronos will open source SPIR-V <-> LLVM conversion tools

SPIR	SPIR 1.2	SPIR 2.0	SPIR-V 1.0
LLVM Interaction	Uses LLVM 3.2	Uses LLVM 3.4	100% Khronos defined Round-trip lossless conversion
Compute Constructs	Metadata/Intrinsics	Metadata/Intrinsics	Native
Graphics Constructs	No	No	Native
Supported Language Feature Set	OpenCL C 1.2	OpenCL C 1.2 OpenCL C 2.0	OpenCL C 1.2 / 2.0 OpenCL C++ GLSL
OpenCL Ingestion	OpenCL 1.2 Extension	OpenCL 2.0 Extension	OpenCL 2.1 Core
Vulkan Ingestion	-	-	Vulkan Core

Driving the SPIR-V Open Source Ecosystem



SPIR-V

- 32-bit Word Stream
- Extensible and easily parsed
- Retains data object and control flow information for effective code generation and translation

Upcoming Vulkan Events

- **Khronos Vulkan Webinar - February 18**
  - https://www.khronos.org/news/events/vulkan-webinar
- **Vulkan sessions at GDC – March 14-18**
  - http://schedule.gdconf.com/search-sessions/vulkan
- **Khronos Sessions co-located with GDC – March 16**
  - https://www.khronos.org/news/events/2016-khronos-sessions-san-francisco

Khronos Vulkan sessions co-located with GDC

Time	Descripton	View Session	Register
Wednesday, March 16th 9:00am – 10:00am	Jon Peddie Research Press Sessions	View	
Wednesday, March 16th 12:00pm – 1:00pm	WebGL + glTF: Mobile Graphics	View	Register
Wednesday, March 16th 1:00pm – 2:00pm	Khronos Chapters Lunch	View	Register
Wednesday, March 16th 2:00pm – 7:00pm	Vulkan: The API for Graphics & Compute	View	Register
Wednesday, March 16th 7:00pm – 9:30pm	Khronos Evening Social	View	Register

Khronos Safety Critical Working Group

OpenGL|SC™ 2005 OpenGL SC 1.0 Fixed function graphics subset

OpenGL|SC™ 2016 (planned) OpenGL SC 2.0 Programmable shader pipeline subset

OpenGL|ES™ 2003 OpenGL ES 1.0 Fixed function graphics

OpenGL|ES™ 2007 OpenGL ES 2/3 Programmable shader pipeline

OpenVX™ EGL™ Vulkan™

New Generation API for safety certifiable graphics AND compute

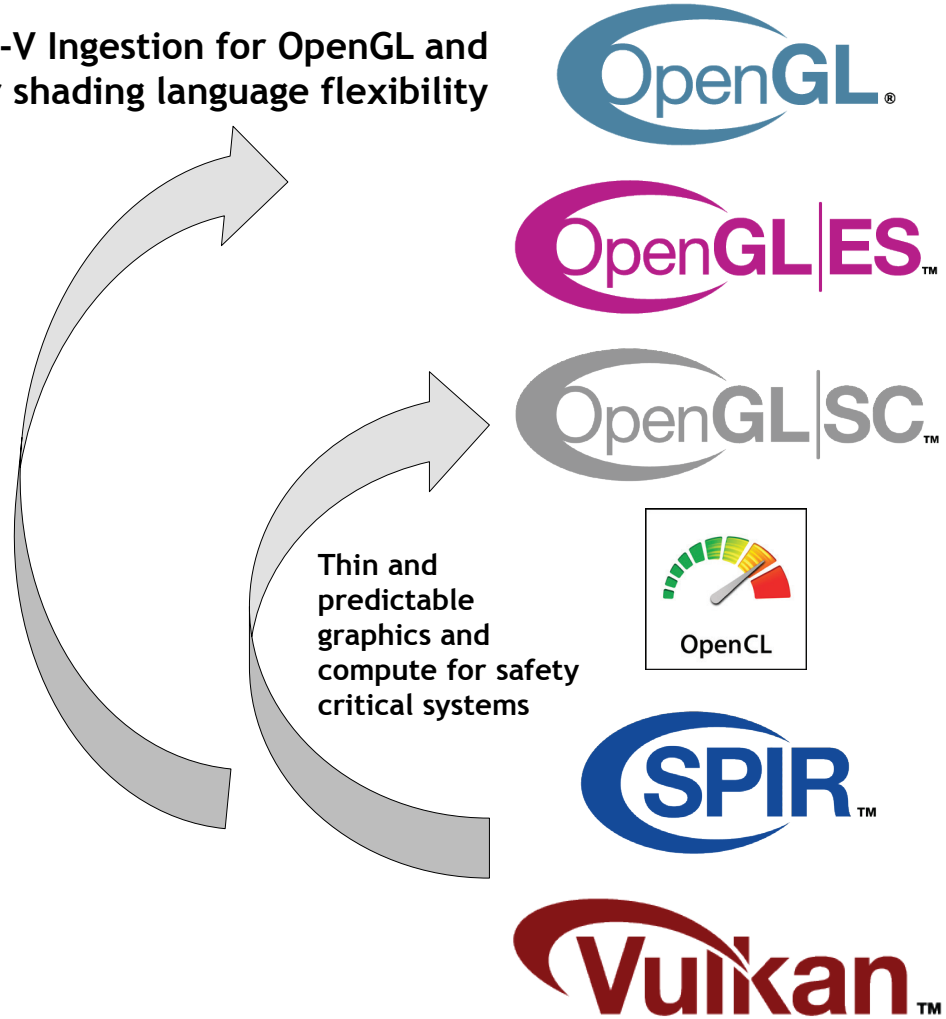
Many future safety critical use cases involve vision and compute acceleration (e.g. neural nets)

New Khronos Safety Critical Advisory Panel

Defining guidelines for creating specifications for ISO 26262 and DO-178B/C certification

Khronos Roadmap Discussions

SPIR-V Ingestion for OpenGL and OpenGL ES for shading language flexibility

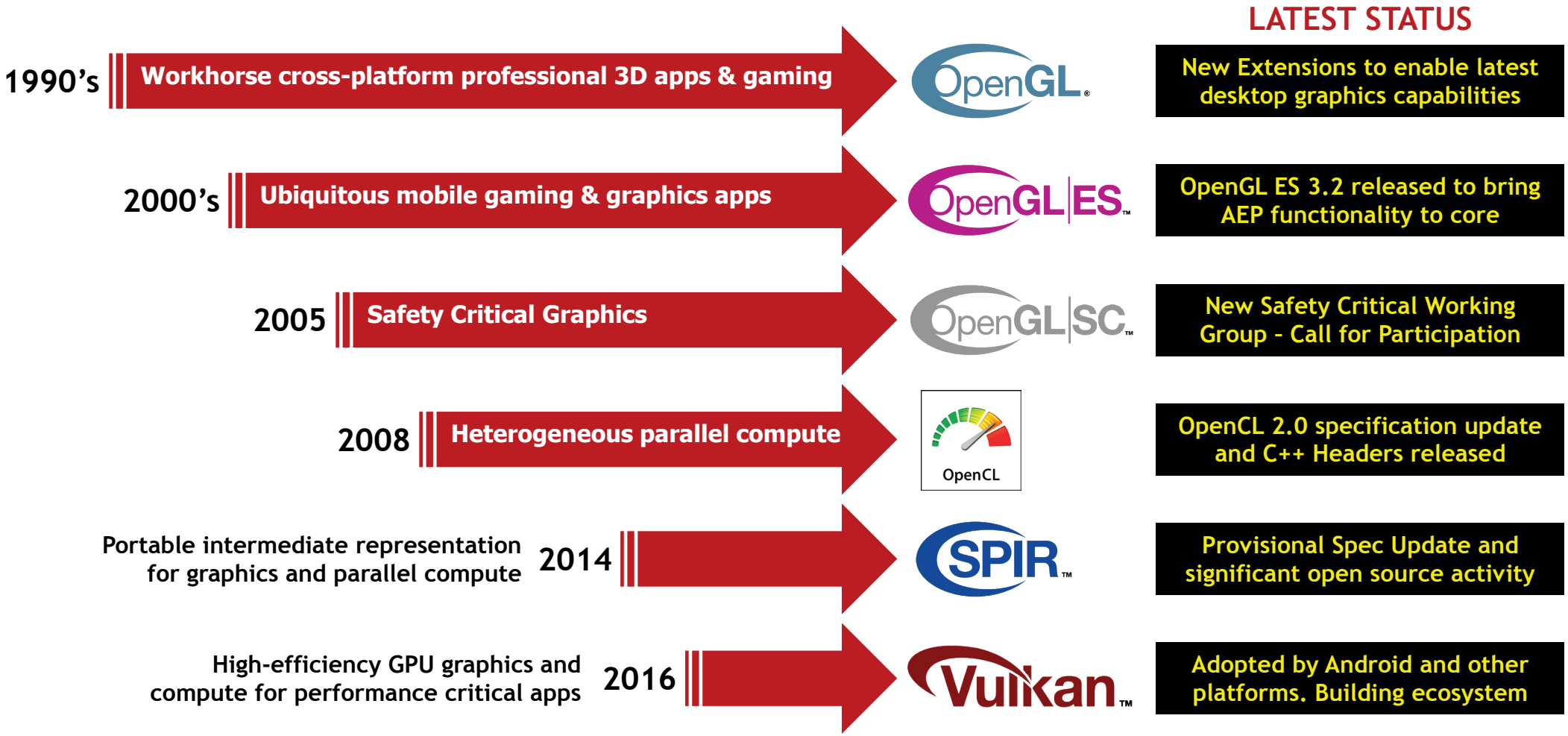


Thin and predictable graphics and compute for safety critical systems

C++ Shading Language and OpenCL-class Heterogeneous Compute to Vulkan runtime

Khronos members decide how to evolve and mix and match a rich set of APIs and technologies to meet market needs

Khronos Open Standards for Graphics and Compute



KHRONOS GROUP