Vulkan™ Overview
February 2016
www.khronos.org/vulkan/
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

**Software**
- **Royalty-Free** specifications
- State-of-the-art IP framework protects members AND the standards

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

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

**Khronos**
- 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...**
The Genesis of Vulkan

Khronos members from all segments of the graphics industry agree the need for new generation cross-platform GPU API.

Significant proposals, IP contributions and engineering effort from many working group members.

Including an unprecedented level of participation from game engine developers.

18 months A high-energy working group effort.

Khronos’ first API ‘hard launch’

Specification, Conformance Tests, SDKs - all open source...
Reference Materials, Compiler front-ends, Samples...
Multiple Conformant Drivers on multiple OS

Vulkan Working Group Participants

© Copyright Khronos Group 2016 - Page 4
The Need for a New Generation GPU API

- **Explicit**
  - Open up the high-level driver abstraction to give direct, low-level GPU control

- **Streamlined**
  - Faster performance, lower overhead, less latency

- **Portable**
  - Cloud, desktop, console, mobile and embedded

- **Extensible**
  - Platform for rapid innovation

OpenGL has evolved over 25 years and continues to meet industry needs - but there is a need for a complementary API approach.

GPUs are increasingly programmable and compute capable + platforms are becoming mobile, memory-unified and multi-core.

GPUs will accelerate graphics, compute, vision and deep learning across diverse platforms: FLEXIBILITY and PORTABILITY are key.
Vulkan Explicit GPU Control

Vulkan 1.0 provides access to OpenGL ES 3.1 / OpenGL 4.X-class GPU functionality but with increased performance and flexibility.

Vulkan Benefits

- Simpler drivers:
  - Improved efficiency/performance
  - Reduced CPU bottlenecks
  - Lower latency
  - Increased portability

- Resource management in app code:
  - Less hitches and surprises

- Command Buffers:
  - Command creation can be multi-threaded
  - Multiple CPU cores increase performance

- Graphics, compute and DMA queues:
  - Work dispatch flexibility

- SPIR-V Pre-compiled Shaders:
  - No front-end compiler in driver
  - Future shading language flexibility

- Loadable Layers:
  - No error handling overhead in production code
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.
Next Generation GPU APIs

- DirectX 12: Only Windows 10
- Vulkan: Cross Platform
- SteamOS
- Ubuntu
- Tizen
- Windows XP
- Windows 7
- Windows 8
- Windows 10
- Only Apple

© Copyright Khronos Group 2016 - Page 8
Vulkan - No Compromise Performance

Retains Traditional Binding Model
(but missing functionality such as Tessellation and Geometry Shaders)

Potential Performance Gain

Amount of work to port from traditional OpenGL and OpenGL ES
Which Developers Should Use Vulkan?

- Vulkan puts more work and responsibility into the application
  - Not every developer will need or want to make that extra investment
- For many developers OpenGL and OpenGL ES will remain the most effective API
  - Khronos actively evolving OpenGL and OpenGL ES in parallel with Vulkan

Vulkan provides more choice to developers and can be used to create new classes of end-user experience
The Power of a Three Layer Ecosystem

Applications can use Vulkan directly for maximum flexibility and control.

Application uses utility libraries to speed development.

Utility libraries and layers

Game Engines fully optimized over Vulkan

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

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

Similar ecosystem dynamic as WebGL
A widely pervasive, powerful, flexible foundation layer enables diverse middleware tools and libraries.
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

- GL_KHR_vulkan_glsl spec released - adds the GLSL features needed for Vulkan
  - Descriptor sets, push constants, specialization constants
  - Separate images/samplers, sub pass input images...
  - Updated front-end open source compiler in Khronos GitHub

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

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

Khronos will open source these tools and translators

Third party kernel and shader Languages

OpenCL C
OpenCL C++

SPIR-V Tools
SPIR-V Validator
SPIR-V (Dis)Assembler

LLVM

GLSL

SPIR-V (Dis)Assembler
LLVM to SPIR-V Bi-directional Translator

IHV Driver Runtimes

Other Intermediate Forms

SPIR-V Version 99
Builder's Magic #: 0x051a00BB
OpMemoryModel
Logical
GLSL450
OpEntryPoint
Fragment shader
function <id> 4
OpTypeVoid
<id> is 2
OpTypeFunction
<id> is 3
return type <id> is 2
OpFunction
Result Type <id> is 2
Result <id> is 4
0
Function Type <id> is 3

© Copyright Khronos Group 2016 - Page 16
Vulkan Developer Resources at Launch

Khronos.org
Canonical Resources
Specifications, Header Files
Feature Set Definitions
(Windows and Linux - post developer feedback)
Quick Reference and Reference Pages
Conformance Test Source and Test Process
Materials to Build SDKs and Tools
Compiler toolchain sources
Validation Layer Source
Loader Source
Layers and Loader documentation
(open source resources in github.com/KhronosGroup)

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

Third Party Websites
Layers, Samples etc.

IHV Websites
Drivers and Loader
Vendor tools and layers

Everything needed to create SDKs for any platform or market
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

Production Path (Performance)

- Vulkan-based Title
- Vulkan’s Common Loader
- IHV’s Installable Client Driver
- Debug Layers
- Validation Layers
- Interactive Debugger

Debug Layers can be installed during Development

Debug information via standardized API calls
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
Conformant Vulkan Drivers at Launch

- 30 Driver submissions passed conformance at Vulkan 1.0 launch
  - ARM: Linux
  - Imagination Technologies: Linux
  - Intel: Linux
  - NVIDIA: Android 6.0, Linux (desktop and embedded), Windows 7-10
  - Qualcomm: Android 6.0
  - www.khronos.org/conformance/adopters/conformant-products

- Drivers in test submission review at Vulkan 1.0 launch
  - AMD: Windows

Khronos and Android leveraging and merging Vulkan tests with the Android Open Source Project (AOSP) and DrawElements Quality Program (dEQP) framework

Open source Vulkan conformance test suite hosted on GitHub

Implementers must pass Test Suite

Enable developers to provide direct feedback and contributions to help resolve cross-vendor inconsistencies

Copyright Khronos Group 2016 - Page 20
One Week Since Launch of Vulkan 1.0

Valve’s SteamOS now supports Vulkan, the cross-platform alternative to DirectX 12.

Imagination introduces PowerVR Series8XE GPUs, brings OpenGL ES 3.2 and Vulkan to the ultra-affordable market.

Ubuntu 16.04 LTS to Ship with Full Support for Vulkan in Mir Display Server.
Vulkan Ecosystem Active at Launch

“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!”

“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

“Talos Principle on Steam has beta Vulkan back-end
Vulkan at GDC!

- Many deep dive sessions
  - Much more detail than we have time for today

- Vulkan sessions at GDC - March 14-18
  - [http://schedule.gdconf.com/search-sessions/vulkan](http://schedule.gdconf.com/search-sessions/vulkan)

- Khronos Sessions co-located with GDC - March 16 - free - no need for GDC Badge!
  - All Khronos sessions will be live streamed and posted
**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**

**OpenCL-class Heterogeneous Compute to Vulkan runtime:**
- C++ Shading Language
- Tiered precision
- Shared virtual memory
- Dynamic parallelism...

**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

1990’s
Workhorse cross-platform professional 3D apps & gaming

2000’s
Ubiquitous mobile gaming & graphics apps

2005
Safety Critical Graphics

2008
Heterogeneous parallel compute

Portable intermediate representation for graphics and parallel compute

2014
High-efficiency GPU graphics and compute for performance critical apps

2016

LATEST STATUS

New Extensions to enable latest desktop graphics capabilities

OpenGL ES 3.2 released to bring AEP functionality to core

New Safety Critical Working Group - Call for Participation

OpenCL 2.0 specification update and C++ Headers released

Provisional Spec Update and significant open source activity

Adopted by Android and other platforms. Building ecosystem