



**Vulkanised 2025**

The 7<sup>th</sup> Vulkan Developer Conference  
Cambridge, UK | February 11-13, 2025

## **Vulkan SDK and Ecosystem: Enhancements Over the Last Year**

---

Karen Ghavam, CEO  
LunarG, Inc.



# Who am I?



Karen Ghavam, CEO  
LunarG, Inc.



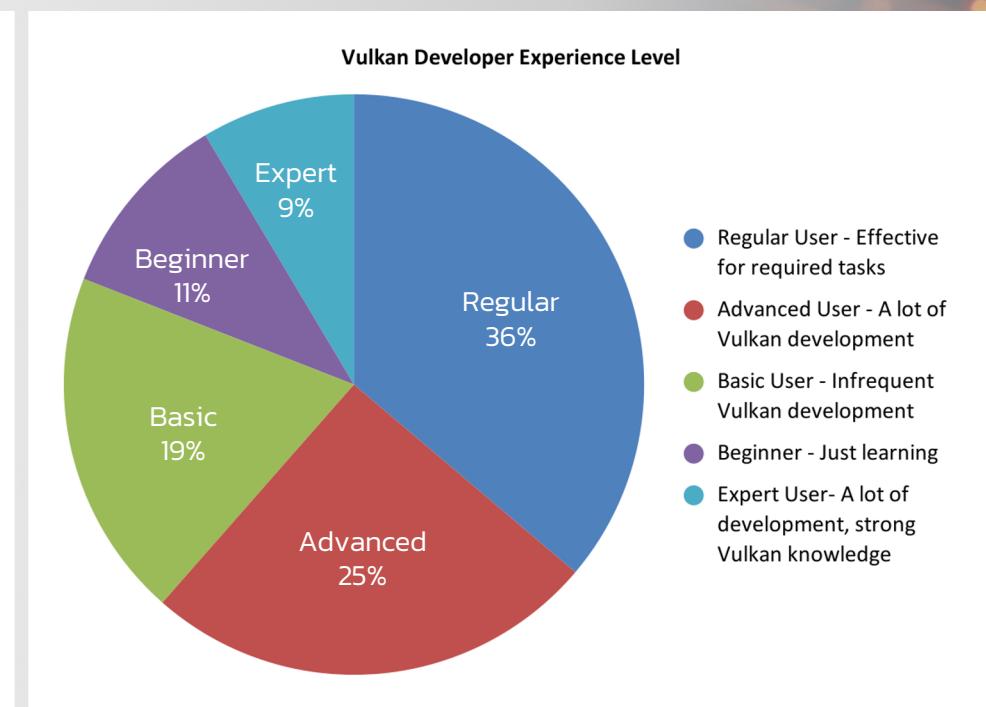
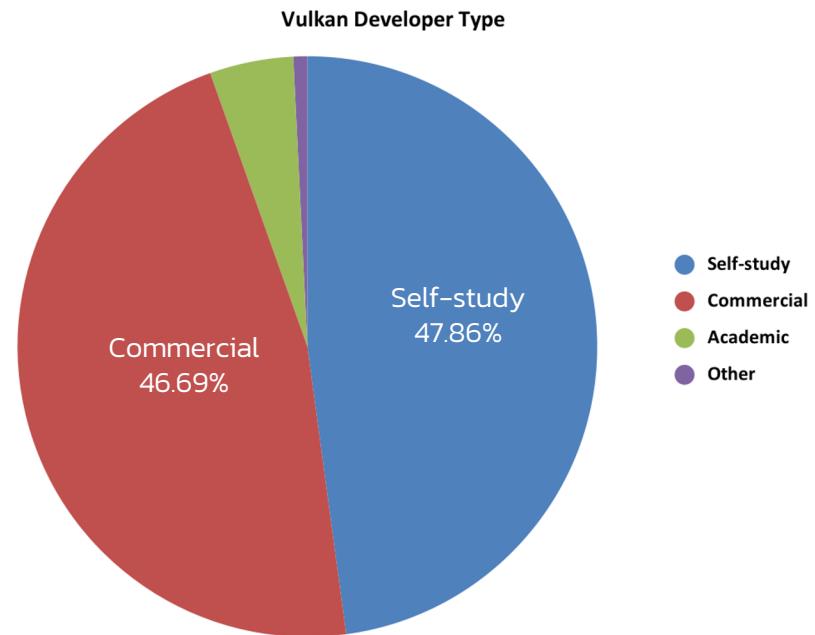
LUNAR)G®

GPU SOFTWARE SPECIALISTS

# Who is LunarG?

- An Independent, private company with Khronos membership
  - GPU Software Specialists
- Developing Vulkan Ecosystem components since 2015
  - Generous sponsorship from Valve and Google
- Vulkan Ecosystem Projects
  - Vulkan SDK
  - Vulkan Loader
  - Vulkan Validation Layers
  - Vulkan Profiles Toolset
  - Vulkan Extension Layer
  - GFXReconstruct
  - glslang
  - ...

# 2024 Ecosystem Survey – Highlights

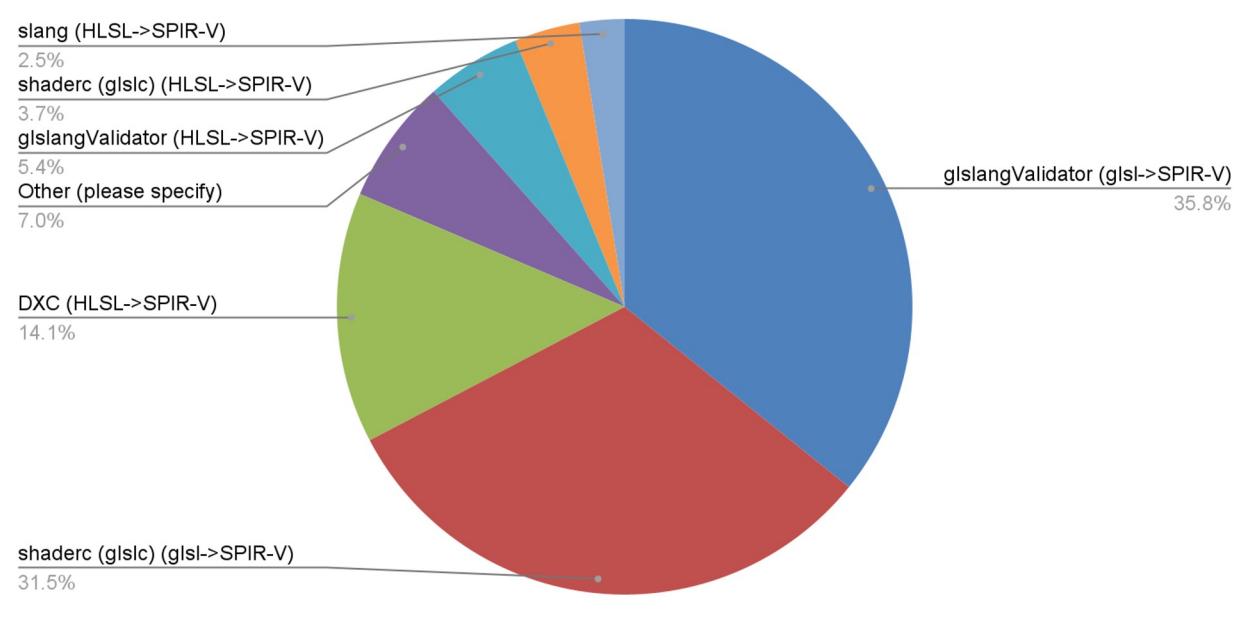


- 258 respondents
- 47%/48% commercial/non-commercial
- 70%/30% (regular, advanced, expert developers)/beginner developers

# 2024 Survey – Themes

- People rely on the validation layer
  - More trust that there are not defects in the validation
- Validation of synchronization extremely useful for finding application bugs
- Synchronization validation for Timeline semaphore is in high demand
- DebugPrintF is used a lot, but has not been maintained at the same level as GPU-AV recently
- Error messages can be very verbose, poorly formatted, or poorly worded
- Specific complaints about the ray tracing extension missing a lot of validation
- More timely delivery of validation layer support for new extensions
- Validation Layer error messages link to the HTML spec. SLOW TO LOAD!
- GPU crashes and timeouts very difficult to debug

# 2024 Survey – Shader Compiler Themes



- GLSL->SPIR-V usage is most common
  - ~67%

- Shader toolchain improvements needed
  - Better options
  - Improved HLSL support
  - Next generation languages
  - DXC – complicated code base with bugs

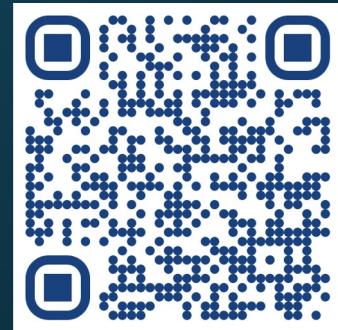
# 2024 Survey – End of Year Progress



<https://www.lunarg.com/2024-ecosystem-survey-progress-report-released/>

# 2025 Ecosystem Survey is Live!

Take the  
Annual  
Developers  
Survey



<https://khr.io/1cq>

# The Vulkan SDK (Vulkan.lunarg.com)



**Vulkan.**

Now Available - Vulkan 1.4.304.0 SDKs [Learn more](#)

Survey Results Are In! See the 2024 Vulkan Ecosystem & SDK Survey Report [Learn more](#)

White Paper: Cross-Stage Shader Optimization [Learn more](#)

White Paper Update: The State of Vulkan on Apple Devices [Learn more](#)

SDK

Issues

Docs

Licenses

Khronos

Sponsored by

**VALVE**

Developed by

**LUNAR**G

Delivered by

**LUNAR**XCHANGE

info@lunarg.com

Welcome to the community for the Vulkan SDK. You can download the latest Vulkan SDK and get SDK questions answered at this site.

DOWNLOAD DEVELOPER TOOLS FOR

Windows Linux macOS Android

© 2025 LunarG, Inc. [Privacy Policy](#) [Cookies](#) [Your privacy choices](#)

A screenshot of the Vulkan SDK website homepage. The page features a dark background with orange circuit board patterns on the right. The Vulkan logo is prominently displayed in red. The main navigation menu includes links for the SDK, Issues, Docs, Licenses, and Khronos. It also highlights the availability of the Vulkan 1.4.304.0 SDKs, survey results, white papers, and developer tools for Windows, Linux, macOS, and Android. The page is sponsored by Valve and developed by LunarG, and delivered by LunarXchange. A footer provides contact information and legal links.

# The Vulkan SDK ([Vulkan.lunarg.com](https://Vulkan.lunarg.com))

The screenshot shows the Vulkan SDK download page. At the top, there are links for '+ Signup' and 'Sign In'. Below that, there's a navigation bar with 'SDK', 'Issues', 'Docs', 'Licenses', 'Khronos', and a 'Sponsored by' section featuring 'VALVE' and 'LUNARG'. The main content area has three tabs: 'Windows' (selected), 'Linux', and 'Mac'. Each tab displays a table of versions with columns for 'Version', 'Released', 'File', and 'SHA 256'. The 'Windows' tab shows versions 1.4.304.0 (13-Jan-2025) and 1.3.296.0 (08-Oct-2024). The 'Linux' tab shows versions 1.4.304.0 (13-Jan-2025), 1.3.296.0 (08-Oct-2024), 1.3.290.0 (23-Jul-2024), and 1.3.283.0 (14-May-2024). The 'Mac' tab shows versions 1.4.304.0 (13-Jan-2025), 1.3.296.0 (08-Oct-2024), 1.3.290.0 (23-Jul-2024), and 1.3.283.0 (14-May-2024). Each row in the table contains links to download the SDK installer, config.json, and runtime zip files.

DOWNLOAD DEVELOPER TOOLS FOR			
SDK version query and download API			
<b>Windows</b>			
Version	Released	File	SHA 256
1.4.304.0	13-Jan-2025	<a href="#">SDK - SDK Installer</a> <a href="#">VulkanSDK-1.4.304.0-Installer.exe (161MB)</a>	<a href="#">config.json (1MB)</a> 87439980598e1f9c2e998d93505b53b45b526-1b4-48e1-2443-d0866742e0
		<a href="#">SDK Config - Config.json</a>	
		<a href="#">config.json (1MB)</a>	
		<a href="#">Runtime zip - Zip file of the runtime components.</a>	
		<a href="#">VulkanRT-1.4.304.0-Components.zip (13MB)</a>	
		<a href="#">VulkanRT-1.4.304.0-Components.exe (13MB)</a>	
1.3.296.0	08-Oct-2024	<a href="#">SDK - SDK Installer</a> <a href="#">VulkanSDK-1.3.296.0-Installer.exe (167MB)</a>	<a href="#">config.json (1MB)</a> 003897fbde39e1400818dab1d071e73e7201a6c0e80e8b080cd24b1cf5
		<a href="#">SDK Config - Config.json</a>	
		<a href="#">config.json (1MB)</a>	
		<a href="#">Runtime zip - Zip file of the runtime components.</a>	
		<a href="#">VulkanRT-1.3.296.0-Components.zip (20MB)</a>	
		<a href="#">VulkanRT-1.3.296.0-Components.exe (20MB)</a>	
1.3.290.0	23-Jul-2024	<a href="#">SDK - SDK Installer</a> <a href="#">VulkanSDK-1.3.290.0-Installer.exe (241MB)</a>	<a href="#">config.json (1MB)</a> 8d303e4c10b08936a4731d1659e5110037a9222778500ed480e970f5a
		<a href="#">SDK Config - Config.json</a>	
		<a href="#">config.json (1MB)</a>	
		<a href="#">Runtime zip - Zip file of the runtime components.</a>	
		<a href="#">VulkanRT-1.3.290.0-Components.zip (20MB)</a>	
		<a href="#">VulkanRT-1.3.290.0-Components.exe (20MB)</a>	
1.3.283.0	14-May-2024	<a href="#">SDK - SDK Installer</a> <a href="#">VulkanSDK-1.3.283.0-Installer.exe (228MB)</a>	<a href="#">config.json (1MB)</a> 8d0f8a09e1ca0a701e224434bd655afef0@1ca7ea3b6300927a372bdf
		<a href="#">SDK Config - Config.json</a>	
		<a href="#">config.json (1MB)</a>	
		<a href="#">Runtime zip - Zip file of the runtime components.</a>	
		<a href="#">VulkanRT-1.3.283.0-Components.zip (20MB)</a>	
		<a href="#">VulkanRT-1.3.283.0-Components.exe (20MB)</a>	

info@lunarg.com

© 2025 LunarG, Inc. [Privacy Policy](#) [Cookies](#) [Your privacy choices](#)



# The Vulkan SDK

Windows	Vulkan Loader	vkconfig	Validation Layer	
	SPIR-V Optimizer	SPIR-V Tools	Crash Diagnostic Layer	vulkaninfo
	shaderc	SPIR-V Validator	Profiles Toolset	Extension Layers
	DXC	SPIR-V Reflect	glslang	GPUInfo
	VKVia	apidump	Vulkan-HPP	VOLK
	MoltenVK	SPIR-V Visualizer	SDL & GLM	SLANG
			Monitor	SPIR-V Cross
				VMA
				GFX Reconstruct

# Why Use the SDK?

- An installation process that is easy and fast
  - Windows (x86 & ARM), Linux, and macOS versions
- Pre-built tools installed into system locations, ready for use.
  - Tools from 22 repositories!
- Vetted and curated content to ensure compatibility and seamless integration
- Ready-to-use versions of the Vulkan Configurator
- SDK release notes and user documentation
- License Registry
  - Details ALL of the open-source licenses present in the SDK

# The Vulkan SDK – Windows 11 on ARM

	Vulkan Loader	vkconfig	Validation Layer	
	SPIR-V Optimizer	SPIR-V Tools	Crash Diagnostic Layer	vulkaninfo
	shaderc	SPIR-V Validator	Profiles Toolset	VOLK
	DXC	SPIR-V Reflect	glslang	SLANG
	VKVia	apidump	Vulkan-HPP	Screenshot
	MoltenVK	SPIR-V Visualizer	SDL & GLM	Monitor
				GFX Reconstruct

- Windows SDK install: "copy only" option

# The Vulkan SDK - Ubuntu Version Upgrade



Ubuntu 22.04 & 24.04



Vulkan Loader	vkconfig	Validation Layer
SPIR-V Optimizer	SPIR-V Tools	Crash Diagnostic Layer
shaderc	SPIR-V Validator	Profiles Toolset
DXC	SPIR-V Reflect	glslang
VKVia	apidump	Vulkan-HPP
MoltenVK	SPIR-V Visualizer	SDL & GLM
		Monitor
		GFX Reconstruct

# The Vulkan SDK - iOS as a Target

	Vulkan Loader	vkconfig	Validation Layer
 Ubuntu 22.04 & 24.04	SPIR-V Optimizer	SPIR-V Tools	Crash Diagnostic Layer
	shaderc	SPIR-V Validator	Profiles Toolset
	DXC	SPIR-V Reflect	GLSLang
	VKVia	apidump	Vulkan-HPP
	MoltenVK	SPIR-V Visualizer	SDL & GLM
iOS Support			Monitor
			GFX Reconstruct

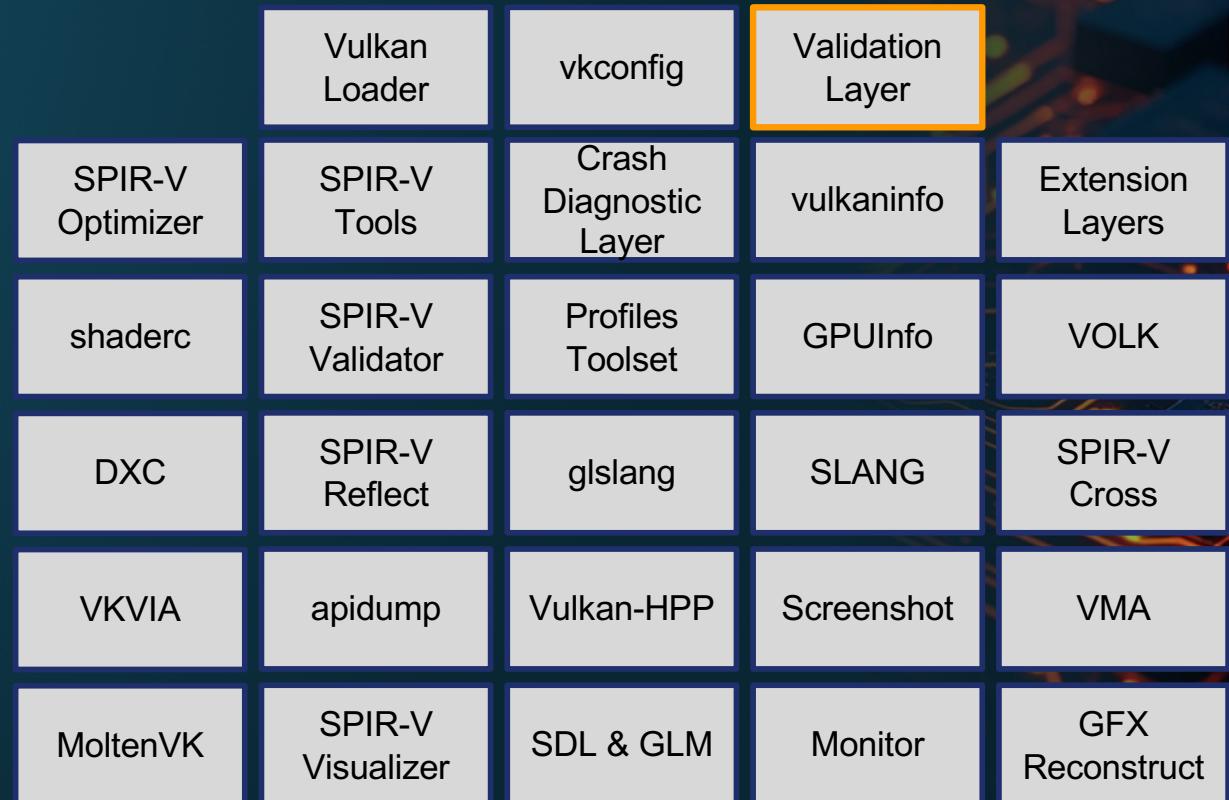
# The Vulkan SDK - Validation Layer

- Synchronization Validation for VK\_KHR\_timeline\_semaphore

Vulkan Loader	vkconfig	Validation Layer		
SPIR-V Optimizer	SPIR-V Tools	Crash Diagnostic Layer	vulkaninfo	Extension Layers
shaderc	SPIR-V Validator	Profiles Toolset	GPUInfo	VOLK
DXC	SPIR-V Reflect	glslang	SLANG	SPIR-V Cross
VKIA	apidump	Vulkan-HPP	Screenshot	VMA
MoltenVK	SPIR-V Visualizer	SDL & GLM	Monitor	GFX Reconstruct

# The Vulkan SDK - Validation Layer

- debugPrintf and GPU-AV can now be used simultaneously



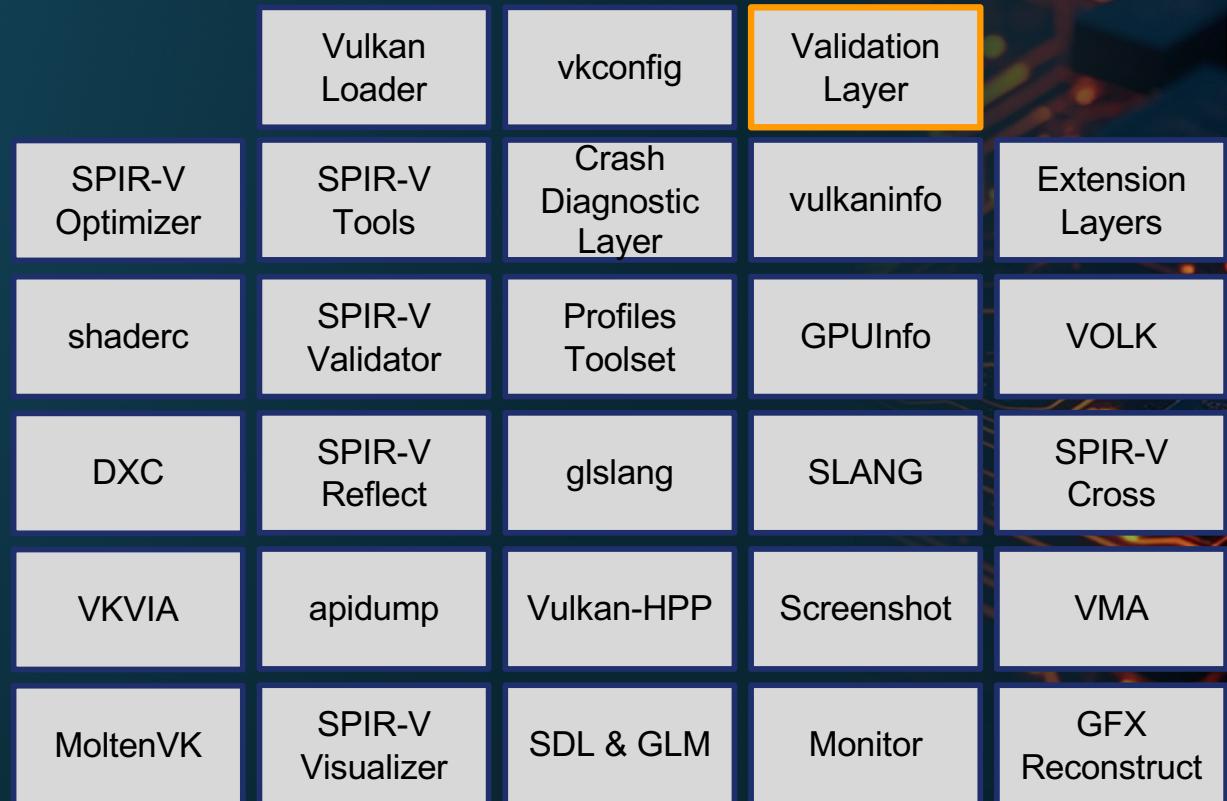
# The Vulkan SDK - Validation Layer

- All the core validation for 1.0, 1.1, and 1.2 completed
- All CPU based ray tracing validation completed

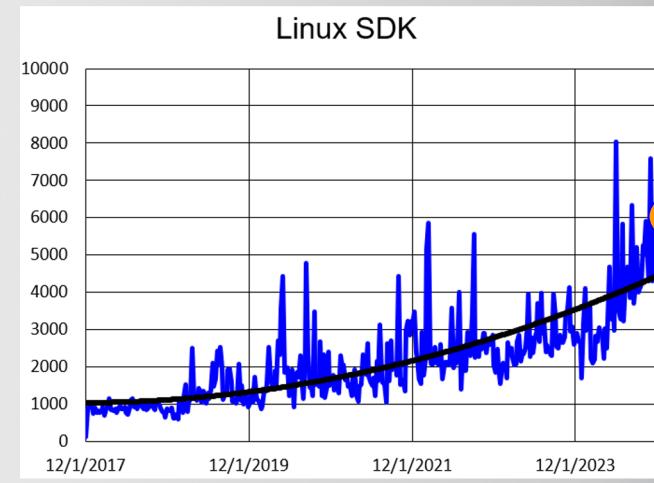
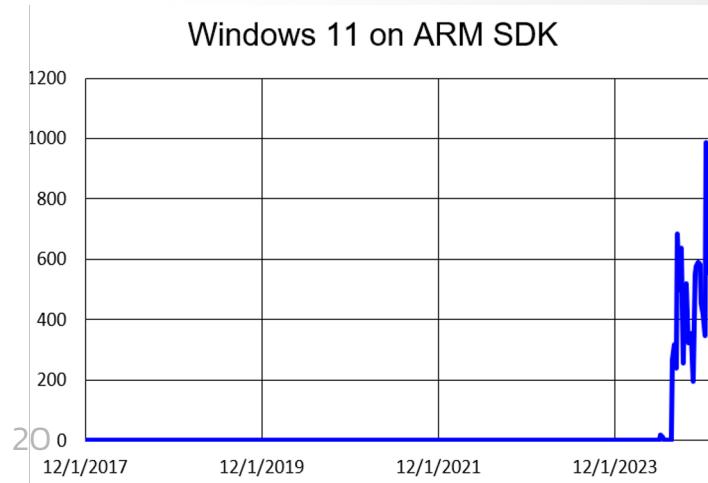
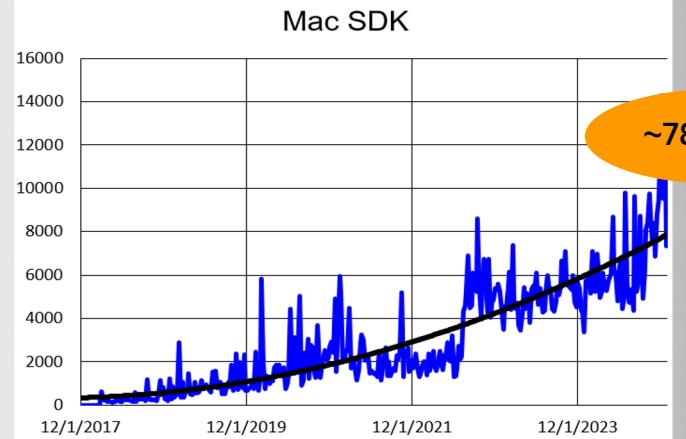
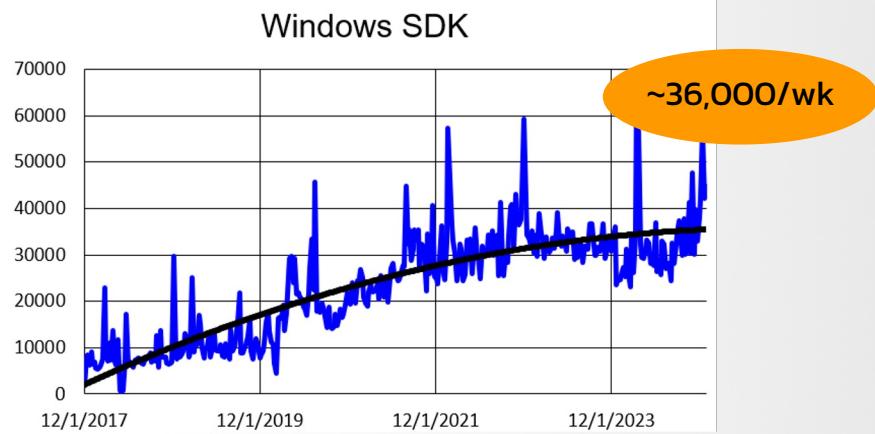
Vulkan Loader	vkconfig	Validation Layer		
SPIR-V Optimizer	SPIR-V Tools	Crash Diagnostic Layer	vulkaninfo	Extension Layers
shaderc	SPIR-V Validator	Profiles Toolset	GPUInfo	VOLK
DXC	SPIR-V Reflect	glslang	SLANG	SPIR-V Cross
VKIA	apidump	Vulkan-HPP	Screenshot	VMA
MoltenVK	SPIR-V Visualizer	SDL & GLM	Monitor	GFX Reconstruct

# The Vulkan SDK - Validation Layer

- Validation support "coincident" for any new extensions



# SDK Download Rates



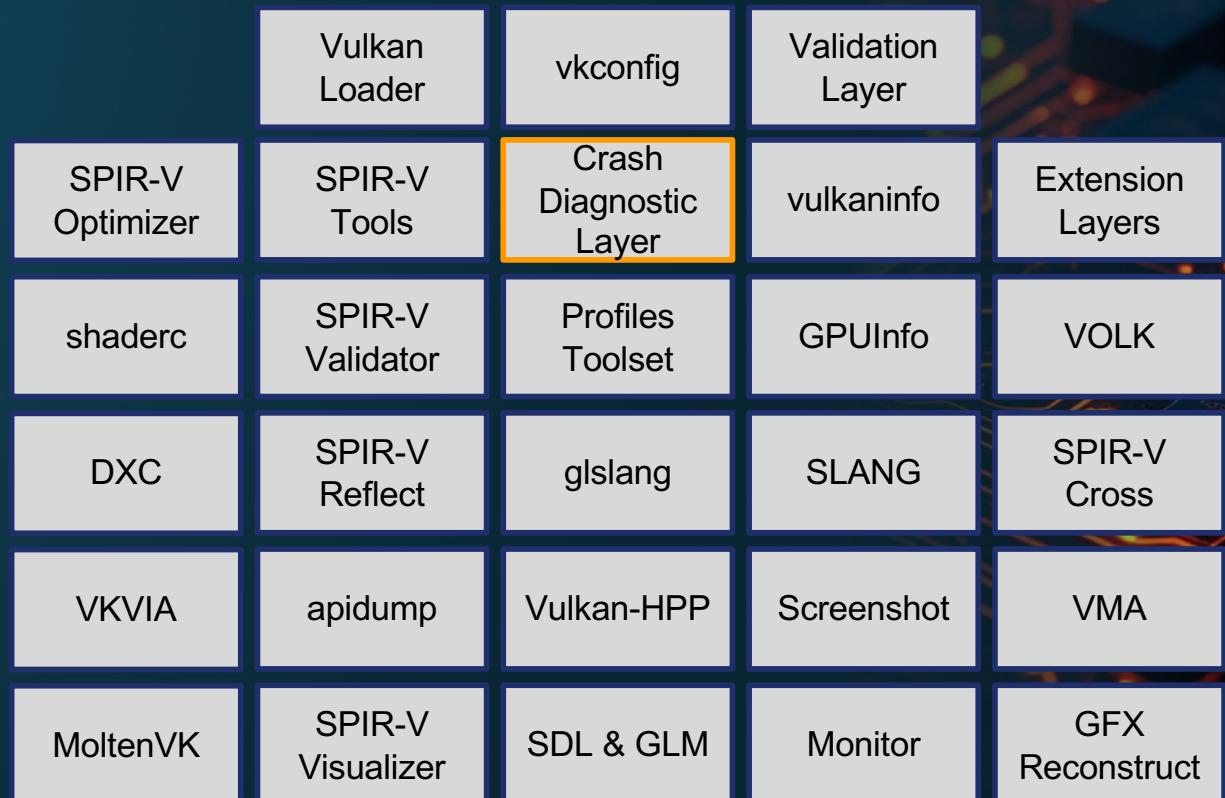
LUNAR)G

# Consistent Vulkan SDK Layer Settings

- Enables configuring any layer programmatically from the Vulkan application
- Three approaches to layer configuration
  - Vulkan Configurator
  - Environment variables
  - Programmatically with VK\_KHR\_layer\_settings
- Consistent behavior across SDK Layers
  - Validation layer
  - Extension Layers
  - Profiles Layer
  - LunarG utility layers (apidump, etc)

# The Vulkan SDK - Crash Diagnostic Layer

- GPU crash and hang debugging
- Command buffer instrumentation with completion checkpoints
- Dump file
- Strong user demand Debugging Device Lost errors very difficult!





# The Vulkan SDK - SLANG

- Slang shading language and compiler
  - SPIR-V
  - HLSL
  - GLSL
  - WGSL for WebGPU
  - Metal Shading Language

Vulkan Loader	vkconfig	Validation Layer
SPIR-V Optimizer	SPIR-V Tools	Crash Diagnostic Layer
shaderc	SPIR-V Validator	Profiles Toolset
DXC	SPIR-V Reflect	glslang
VKVia	apidump	Vulkan-HPP
MoltenVK	SPIR-V Visualizer	SDL & GLM
		Monitor
		GFX Reconstruct

# Compiler Explorer (Godbolt.org)

- Online tool
  - Write code in various programming languages
  - See the resulting assembly code in real-time
- LunarG (Spencer) added support for
  - GLSL
  - spirv-opt
  - spirv-val
  - spirv-reflect
  - slang!



Blog post:  
<https://shader-slang.com/blog/2024/12/17/slang-support-in-godbolt/>

## 1.4.304.0 Vulkan SDK

- Live January 14, 2025
- Windows – Automatic installation of SDK version of Vulkan Loader
- glslang shared library
- Addition of GFXReconstruct to the macOS SDK
- 32-bit versions of the Vulkan layers have been removed
  - A 32 bit runtime (Vulkan Loader, vulkaninfo) is still included

## 1.4.304.1 Vulkan SDK

- Just Released!
- Three really cool enhancements:
  - Portable Ray tracing support in GFXReconstruct
  - Fast loading of Vulkan Specification from SDK Validation Layer
  - Vulkan Configurator 3

# Portable Ray Tracing

- What is Portable Ray Tracing?
  - Tracking at capture time, then translated at replay time for device independence:
    - Buffer Device Addresses
    - Shader Group Handles
    - Acceleration Structures
- How is this useful for the Vulkan application developer?
  - Bug-reporting across different vendors/drivers
- Additional benefit of portable raytracing work:
  - Other GPU-centric commands result in device independent address translations
    - Buffer Device Addresses increasingly used:
      - in shader-code
      - gpu-centric extensions like Device-Generated Commands
- White paper: <https://www.lunarg.com/portable-raytracing-with-gfxreconstruct>

**Demo at the LunarG table!**

LUNAR G

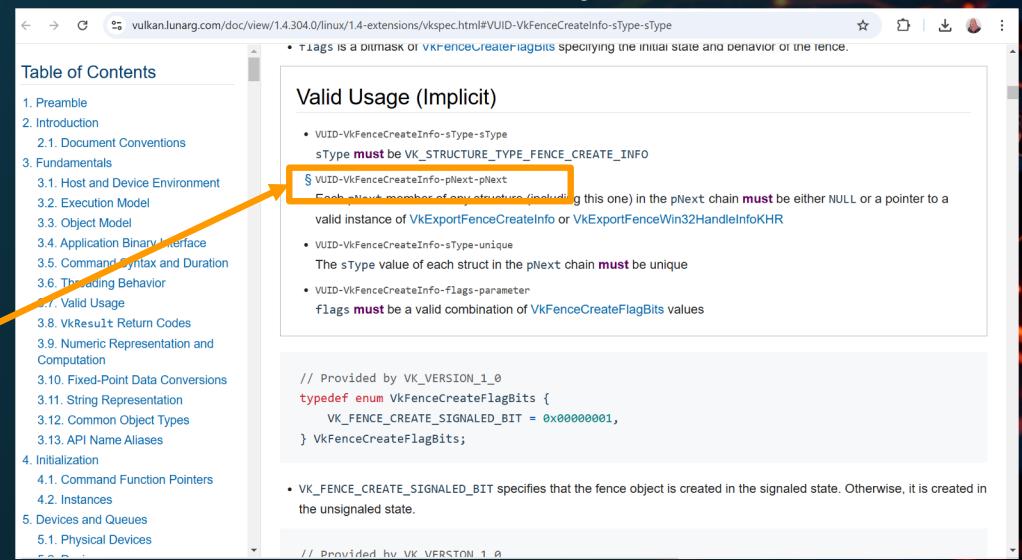
# Slow Vulkan Spec Load Times!

- SDK Validation Layer Error Messages
  - Linking to specific violated VUID in the specification
  - Referencing the HTML single file version
    - SLOW to LOAD in your browser!

## SDK Validation Layer Error:

```
VUID-VkImageViewCreateInfo-pNext-pNext(ERROR / SPEC): msgNum: 151814321 - Validation  
Error: [ VUID-VkImageViewCreateInfo-pNext-pNext ] | MessageID = 0x90c80b1 |  
vkCreateImageView(): pCreateInfo->pNext chain includes a structure with unexpected  
VkStructureType VK_STRUCTURE_TYPE_IMAGE_CREATE_INFO. This error is based on the Valid  
Usage documentation for version 304 of the Vulkan header. It is possible that you are using a  
struct from a private extension or an extension that was added to a later version of the Vulkan  
header, in which case the use of pCreateInfo->pNext is undefined and may not work correctly  
with validation enabled.  
The Vulkan spec states: Each pNext member of any structure (including this one) in the pNext  
chain must be either NULL or a pointer to a valid instance of VkExportMetalObjectCreateInfoEXT,  
VkImageViewASTCDecodeModeEXT, VkImageViewMinLodCreateInfoEXT,  
VkImageViewSampleWeightCreateInfoQCOM, VkImageViewSlicedCreateInfoEXT,  
VkImageViewUsageCreateInfo, VkOpaqueCaptureDescriptorDataCreateInfoEXT, or  
VkSamplerYcbcrConversionInfo (https://vulkan.lunarg.com/doc/view/1.4.304.0/linux/1.4-extensions/vkspec.html#VUID-VkImageCreateInfo-pNext-pNext)  
VUID-VkFenceCreateInfo-sType(ERROR / SPEC): msgNum: 913590280 - Validation Error: [  
VUID-VkFenceCreateInfo-sType-sType ] | MessageID = 0x36744808 | vkCreateFence():  
pCreateInfo->sType must be VK_STRUCTURE_TYPE_FENCE_CREATE_INFO.  
The Vulkan spec states: sType must be VK_STRUCTURE_TYPE_FENCE_CREATE_INFO  
(https://vulkan.lunarg.com/doc/view/1.4.304.0/linux/1.4-extensions/vkspec.html#VUID-VkFenceCreateInfo-sType-sType)
```

## HTML Single File Specification:



The screenshot shows a browser window displaying the Vulkan specification. The URL is <https://vulkan.lunarg.com/doc/view/1.4.304.0/linux/1.4-extensions/vkspec.html#VUID-VkFenceCreateInfo-sType-sType>. The page content includes a Table of Contents and a 'Valid Usage (Implicit)' section. The 'Valid Usage (Implicit)' section contains several bullet points, one of which is highlighted with a red box and an orange arrow pointing from it to the corresponding error message in the SDK log. The highlighted text reads: '\$ VUID-VkFenceCreateInfo-pNext-pNext Each <code>pNext</code> member of any structure (including this one) in the <code>pNext</code> chain **must** be either NULL or a pointer to a valid instance of <code>VkExportFenceCreateInfo</code> or <code>VkExportFenceWin32HandleInfoKHR</code>'. Below this, there is C code for the <code>VkFenceCreateInfo</code> structure.

# NO MORE Slow Vulkan Spec Load Times!

- SDK Validation Layer Error Messages
  - Now referencing Antora build of Vulkan specification

## SDK Validation Layer Error:

```
VUID-VkImageViewCreateInfo-pNext-pNext(ERROR / SPEC): msgNum: 151814321 - Validation  
Error: [ VUID-VkImageViewCreateInfo-pNext-pNext ] | MessageID = 0x90c80b1 |  
vkCreateImageView(): pCreateInfo->pNext chain includes a structure with unexpected  
VkStructureType VK_STRUCTURE_TYPE_IMAGE_CREATE_INFO. This error is based on the Valid  
Usage documentation for version 304 of the Vulkan header. It is possible that you are using a  
struct from a private extension or an extension that was added to a later version of the Vulkan  
header, in which case the use of pCreateInfo->pNext is undefined and may not work correctly  
with validation enabled.  
The Vulkan spec states: Each pNext member of any structure (including this one) in the pNext  
chain must be either NULL or a pointer to a valid instance of VkExportMetalObjectCreateInfoEXT,  
VkImageViewASTCDecodeModeEXT, VkImageViewMinLodCreateInfoEXT,  
VkImageViewSampleWeightCreateInfoQCOM, VkImageViewSlicedCreateInfoEXT,  
VkImageViewUsageCreateInfo, VkOpaqueCaptureDescriptorDataCreateInfoEXT, or  
VkCompuerYoderConversionInfo.  
(https://vulkan.lunarg.com/doc/view/1.4.304.1/linux/antora/spec/latest/chapters/resources.html#VUID-VkImageViewCreateInfo-pNext-pNext)  
VUID-VkFenceCreateInfo-sType-sType(ERROR / SPEC): msgNum: 913590280 - Validation Error: [  
VUID-VkFenceCreateInfo-sType-sType ] | MessageID = 0x36744808 | vkCreateFence():  
pCreateInfo->sType must be VK_STRUCTURE_TYPE_FENCE_CREATE_INFO.  
The Vulkan spec states: sType must be VK_STRUCTURE_TYPE_FENCE_CREATE_INFO  
(https://vulkan.lunarg.com/doc/view/1.4.304.1/linux/antora/spec/latest/chapters/synchronization.html#VUID-VkFenceCreateInfo-sType-sType)
```

## Antora built specification:

The screenshot shows the Vulkan Documentation website (<https://vulkan.lunarg.com/doc/view/1.4.304.1/linux/antora/spec/latest/chapters/synchronization.html#VUID-VkFenceCreateInfo-pNext-pNext>). The page title is "Valid Usage (Implicit)". The left sidebar lists chapters such as Introduction, Fundamentals, Initialization, Devices and Queues, Command Buffers, Synchronization and Cache Control, Render Pass, Shaders, Pipelines, Memory Allocation, Resource Creation, Samplers, Resource Descriptors, Shader Interfaces, Image Operations, Fragment Density Map Operations, Queries, and Vulkan Specification. The right sidebar contains a "Contents" section with links to various Vulkan topics like Execution and Memory Dependencies, Image Layout Transitions, Pipeline Stages, Access Types, Framebuffer Region Dependencies, View-Local Dependencies, Device-Local Dependencies, Implicit Synchronization Guarantees, Fences, Alternate Methods to Signal Fences, Importing Fence Payloads, Semaphores, Semaphore Signaling, Semaphore Waiting, Semaphore State Requirements for Wait Operations, and Host Operations on. The main content area displays the "Valid Usage (Implicit)" section for the `VUID-VkFenceCreateInfo-pNext-pNext` rule, which states that each `pNext` member of any structure in the `pNext` chain must be either `NULL` or a pointer to a valid instance of `VkExportFenceCreateInfo` or `VkExportFenceWin32HandleInfoKHR`. Below this, it specifies that the `sType` value of each struct in the `pNext` chain must be unique and that the `flags` parameter must be a valid combination of `VkFenceCreateFlagBits` values. A code snippet at the bottom provides the definition of `VkFenceCreateFlagBits`:

```
// Provided by VK_VERSION_1_0
typedef enum VkFenceCreateFlagBits {
    VK_FENCE_CREATE_SIGNALLED_BIT = 0x00000001,
} VkFenceCreateFlagBits;
```

# Vulkan Configurator – Version 3

- Improved layers loading and selection
  - Loading multiple versions of the same layer
  - Explicit selection of the layer version used by a configuration
  - Per-layer enabling of layer settings
- Improved layers ordering
  - Ordering of all layers executed by Vulkan application
- Improved executable list
  - Multiple set of options per executable
  - Per-application layers configuration
- *Vulkan Loader* logging enhancements
  - Aids in system diagnosis

## Vulkan Configurator 3.0.0-20250128 (BETA)

Vulkan Loader Management    Vulkan Layers Location    Application Launcher    System Diagnostic    Documentation    About

Vulkan Loader Configuration Scope:

Any Running Vulkan Executable    \${VULKAN\_BIN}\vkcube.e

Apply a Vulkan Loader Configuration:

- Crash Diagnostic
- Disable All Vulkan Layers
- Frame Capture
- Portability
- Synchronization
- Validation

Include Vulkan Layers Selection and Execution Order:

Execute Closer to the Vulkan Application

Vulkan Layers Located by the Vulkan Application	Execution Order
VK_LAYER_LUNARG_api_dump - 1.4.304	Auto
VK_LAYER_LUNARG_crash_diagnostic - 1.4.304 (BETA)	Auto
VK_LAYER_LUNARG_gfxreconstruct - 1.4.304	Auto
VK_LAYER_LUNARG_monitor - 1.4.304	Auto
VK_LAYER_LUNARG_screenshot - 1.4.304	Auto
VK_LAYER_KHRONOS_validation - 1.4.304	Enable

Execute Closer to the Vulkan Driver

Include Vulkan Loader Messages:

Errors     Warnings     Informations     Debug     Layers     Drivers

KHRONOS\_validation:    Latest    User-Defined Settings    1.4.304

Validation Areas

- Fine Grained Locking
- Core
  - Image Layout
  - Command Buffer State
  - Object in Use
  - Query
- Shader
  - Caching
  - Disable spirv-val
- Handle Wrapping
- Object Lifetime
- Stateless Parameter
- Thread Safety
- Synchronization
  - Submit time validation
  - Shader accesses heuristic
- Error messages
  - Extra key-value properties
  - Pretty print extra properties
- Debug Print
  - Redirect Printf messages to stdout

LUNAR G

**Gratitude for their support**

**VALVE**

**Google**

**SAMSUNG**

**Qualcomm**

**arm**

**AMD**

# Thank you!

## Actions

Take the  
Annual  
Developers  
Survey



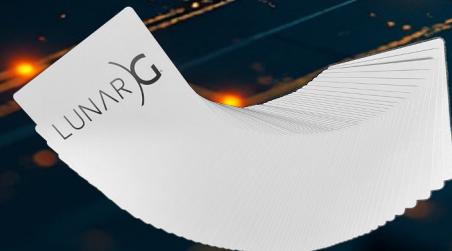
<https://khr.io/1cq>

Download this  
Presentation



<https://khr.io/1cr>

Talk to us and  
get Swag!



Visit the  
LunarG Sponsor Table

Your Feedback  
Matters!

Survey Results

- Are shared with the Khronos Vulkan Working Group
- Are used to drive development priorities throughout 2025

Survey Closes  
Wednesday, Feb. 19, 2025  
(GMT-7)