Vulkanised 2025

The 7th 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?

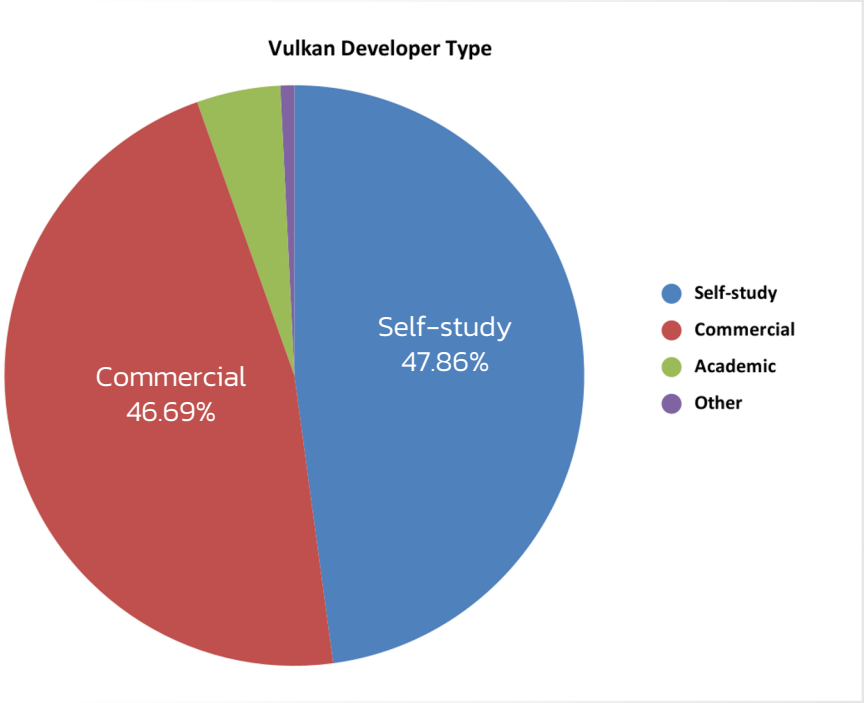
Karen Ghavam, CEO
LunarG, Inc.

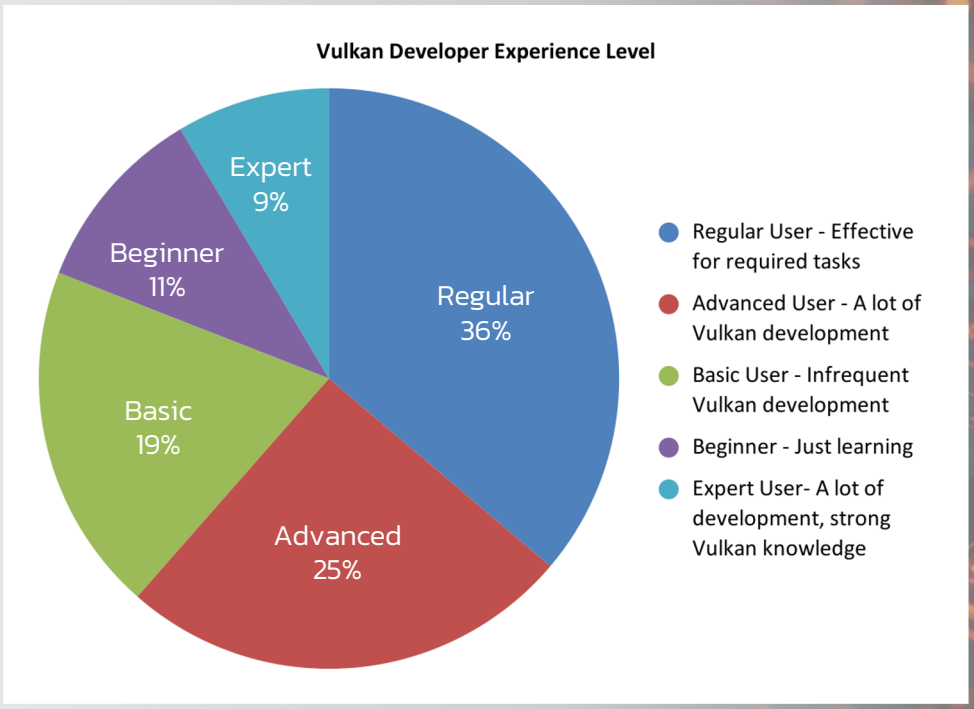
Who is LunarG?

- 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
  - ...

LUNARG

2024 Ecosystem Survey - Highlights





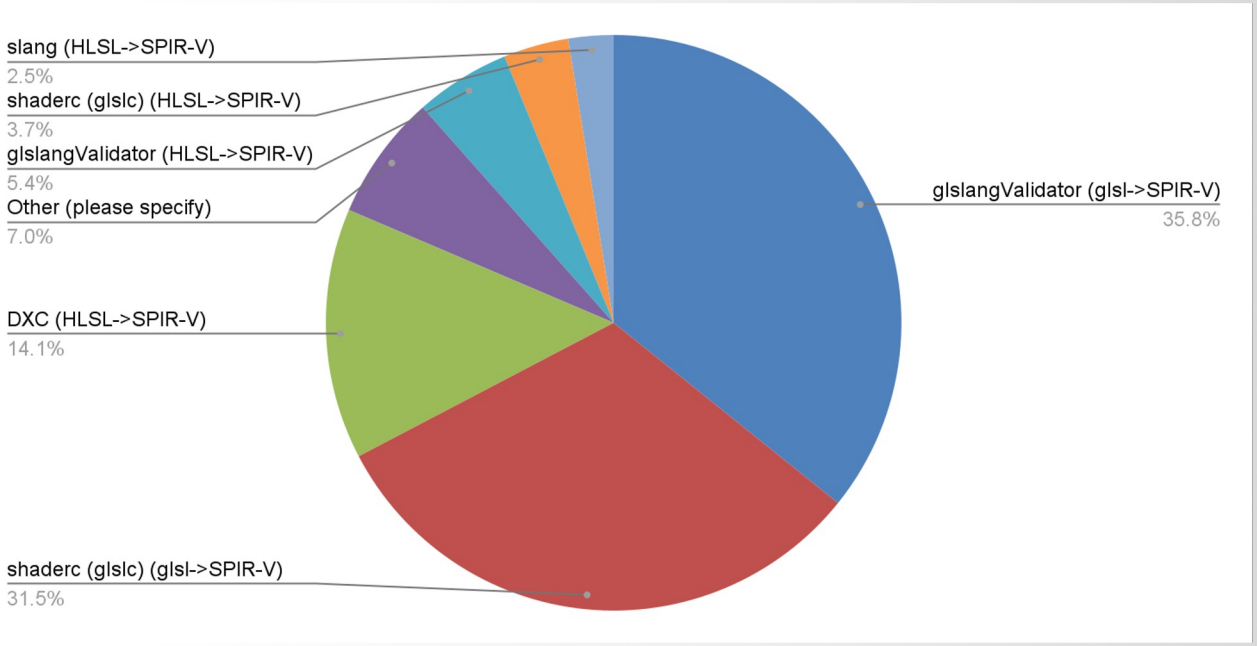
- 258 respondents
- 47%/48% commercial/non-commercial
- 70%/30% (regular, advanced, expert developers)/beginner developers

LUNARG

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



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

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

LUNARG

2024 Survey – End of Year Progress

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

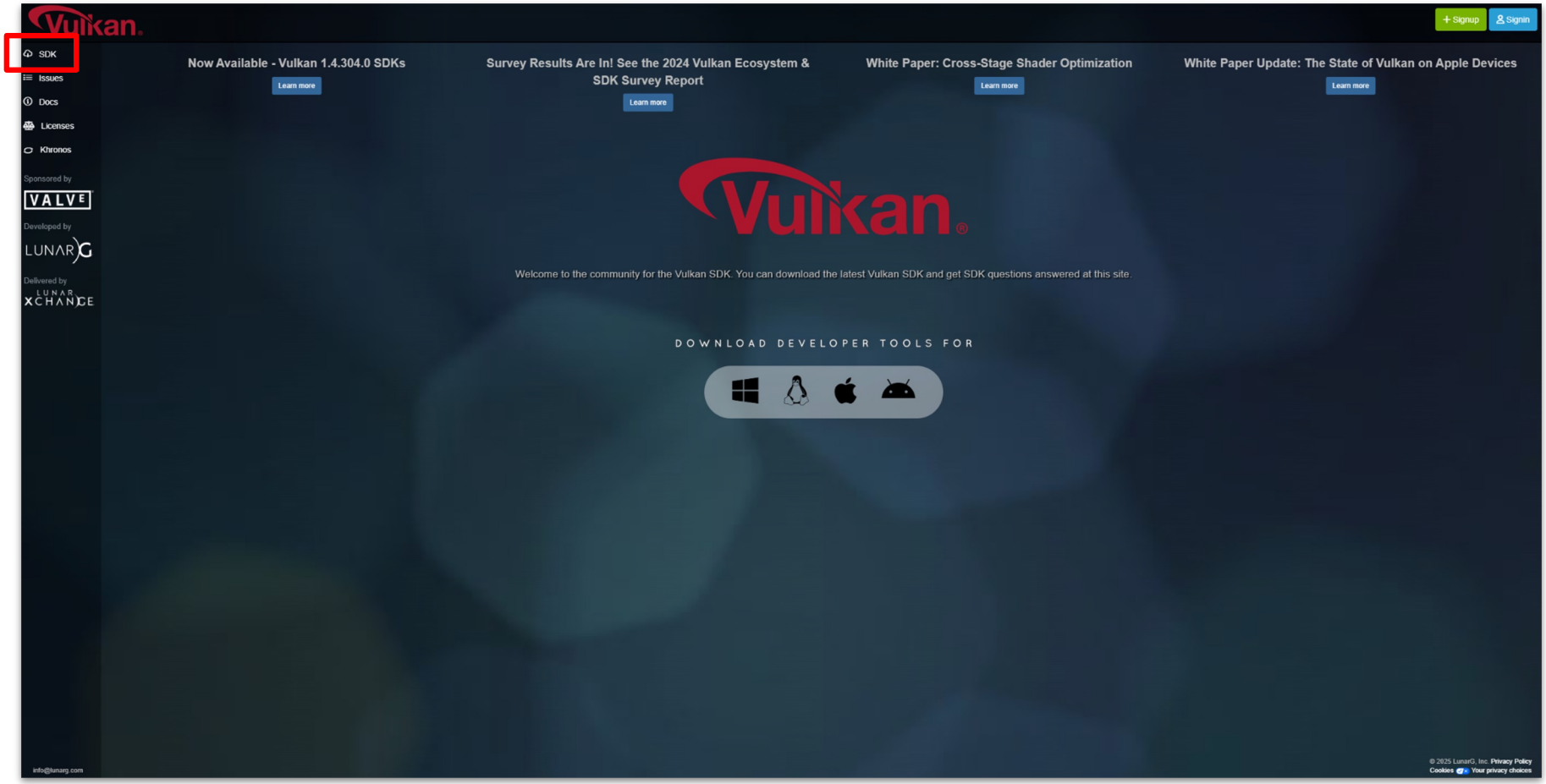
LUNARG

2025 Ecosystem Survey is Live!

Take the Annual Developers Survey

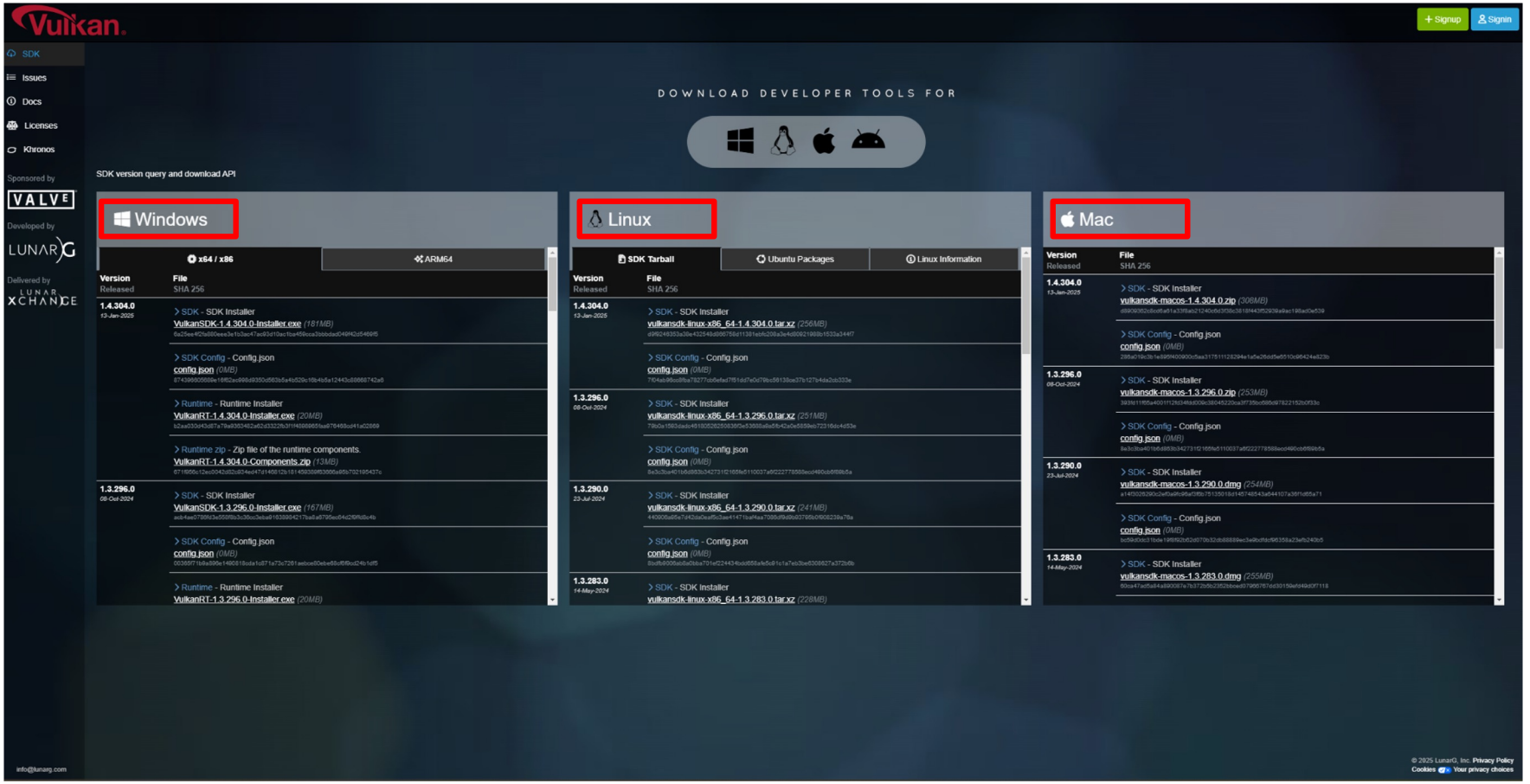
https://khr.io/1cq

The Vulkan SDK (Vulkan.lunarg.com)



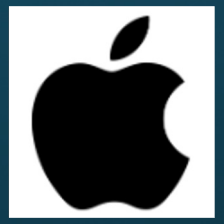
Delivered by LunarG in close coordination with the Khronos Vulkan working group

The Vulkan SDK (Vulkan.lunarg.com)



LUNARG

The Vulkan SDK



	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
VKVIA	apidump	Vulkan-HPP	Screenshot	VMA
MoltenVK	SPIR-V Visualizer	SDL & GLM	Monitor	GFX Reconstruct

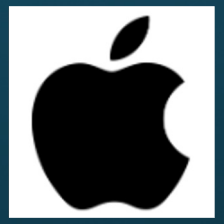
LUNARG

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

LUNARG

The Vulkan SDK – Windows 11 on ARM



	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
VKVIA	apidump	Vulkan-HPP	Screenshot	VMA
MoltenVK	SPIR-V Visualizer	SDL & GLM	Monitor	GFX Reconstruct

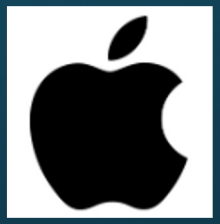
- Windows SDK install: "copy only" option

LUNARG

The Vulkan SDK – Ubuntu Version Upgrade

Windows

Ubuntu 22.04 & 24.04



Windows On arm

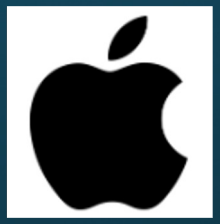
	Vulkan Loader	vkconfig	Validation Layer	
SPIR-V Optimizer	SPIR-V Tools	Crash Diagnostic Layer	vulkaninfo	Extension Layers
shaderc	SPIR-V Validator	Profiles Toolset	GPUInfo	VOLK
DXC	SPIR-V Reflect	glslang	SLANG	SPIR-V Cross
VKVIA	apidump	Vulkan-HPP	Screenshot	VMA
MoltenVK	SPIR-V Visualizer	SDL & GLM	Monitor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

LUNARG

The Vulkan SDK – iOS as a Target

Windows

Ubuntu 22.04 & 24.04



iOS Support

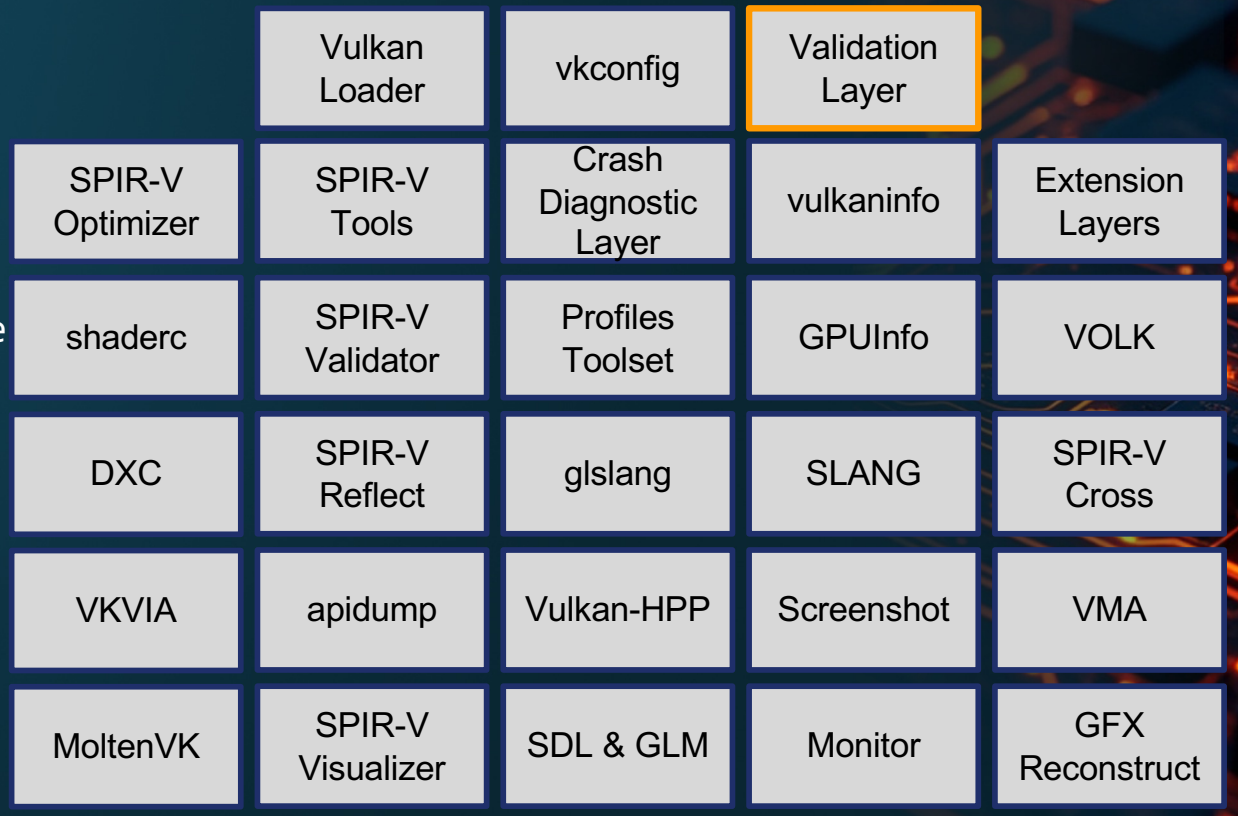
Windows On arm

	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
VKVIA	apidump	Vulkan-HPP	Screenshot	VMA
MoltenVK	SPIR-V Visualizer	SDL & GLM	Monitor	GFX Reconstruct

LUNARG

The Vulkan SDK – Validation Layer

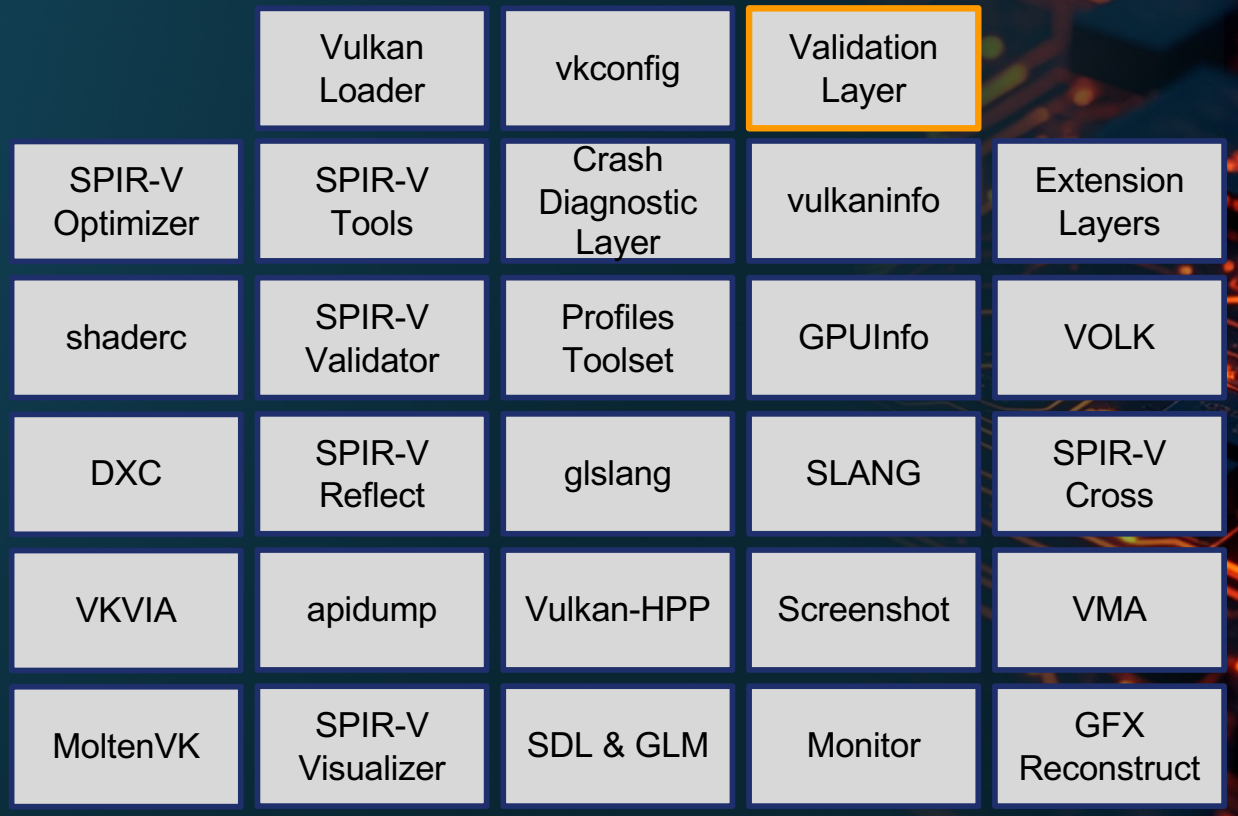
- Synchronization Validation for VK_KHR_timeline_semaphore



LUNARG

The Vulkan SDK – Validation Layer

- debugPrintf and GPU-AV can now be used simultaneously



LUNARG

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
VKVIA	apidump	Vulkan-HPP	Screenshot	VMA
MoltenVK	SPIR-V Visualizer	SDL & GLM	Monitor	GFX Reconstruct

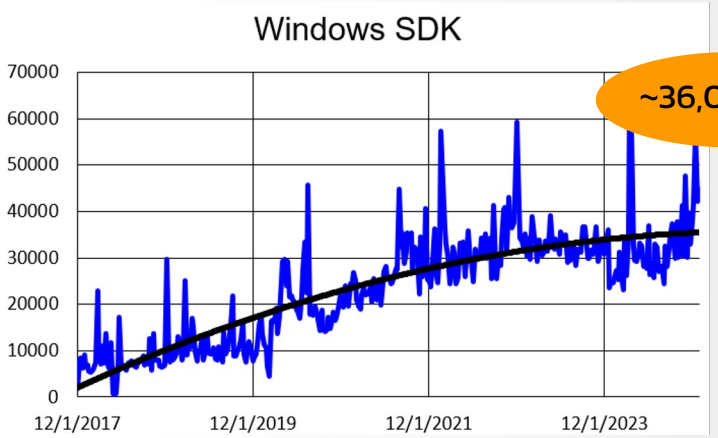
The Vulkan SDK – Validation Layer

- Validation support "coincident" for any new extensions

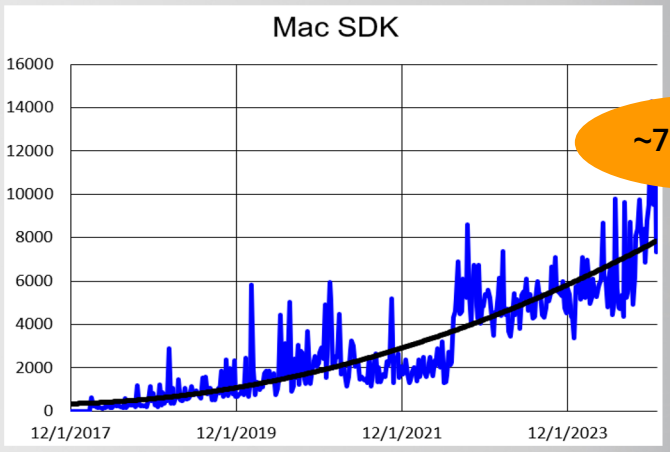
	Vulkan Loader	vkconfig	Validation Layer	
SPIR-V Optimizer	SPIR-V Tools	Crash Diagnostic Layer	vulkaninfo	Extension Layers
shaderc	SPIR-V Validator	Profiles Toolset	GPUInfo	VOLK
DXC	SPIR-V Reflect	glslang	SLANG	SPIR-V Cross
VKVIA	apidump	Vulkan-HPP	Screenshot	VMA
MoltenVK	SPIR-V Visualizer	SDL & GLM	Monitor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

LUNARG

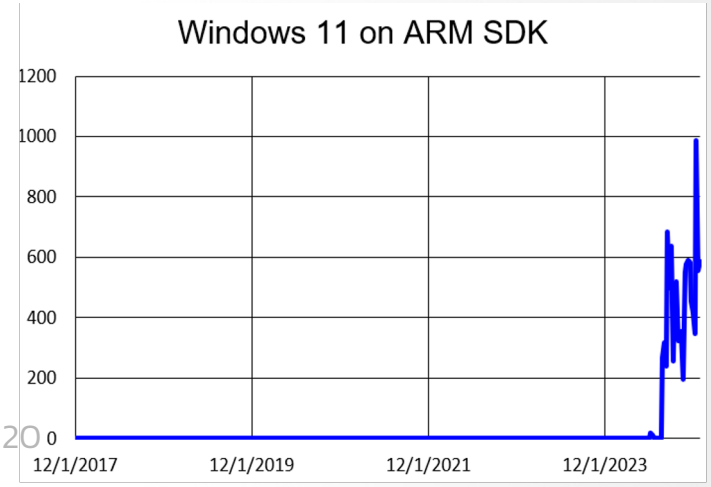
SDK Download Rates

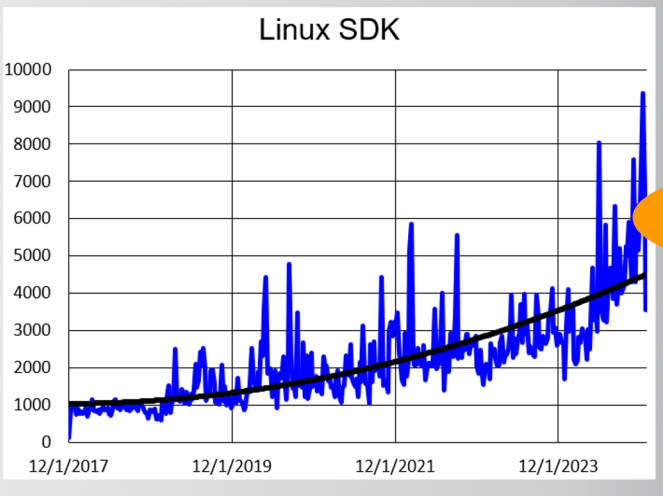


~36,000/wk



~7800/wk





~4500/wk

Consistent Vulkan SDK Layer Settings

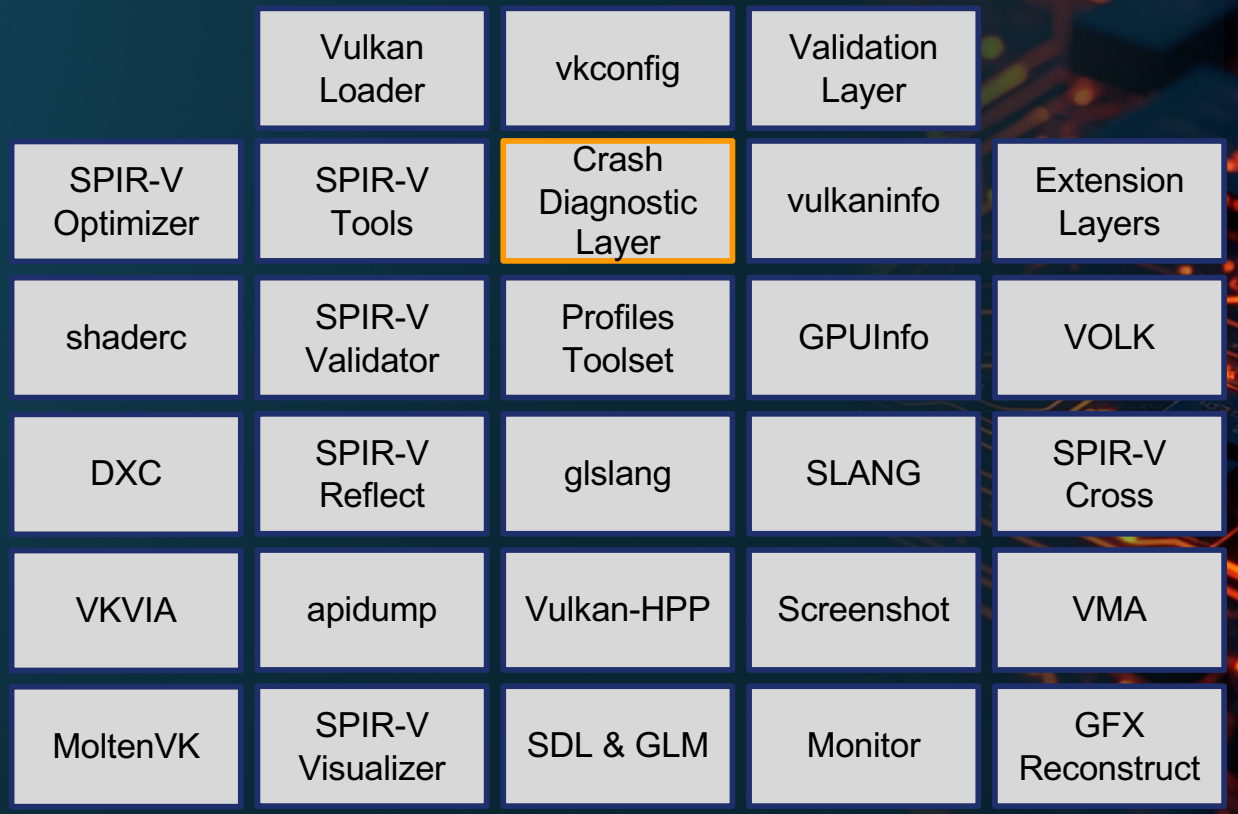
- Enables configuring any layer programmatically from the Vulkan application
- Three approaches to layer configuration
  - Vulkan Configurator
  - Environment variables
  - Programmatically with VK_KHR_layer_settings
- Consistent behavior across SDK Layers
  - Validation layer
  - Extension Layers
 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

Documentation: https://github.com/KhronosGroup/Vulkan-Utility-Libraries/blob/main/README.md

LUNARG

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!



LUNARG

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	vulkaninfo	Extension Layers
shaderc	SPIR-V Validator	Profiles Toolset	GPUInfo	VOLK
DXC	SPIR-V Reflect	glslang	SLANG	SPIR-V Cross
VKVIA	apidump	Vulkan-HPP	Screenshot	VMA
MoltenVK	SPIR-V Visualizer	SDL & GLM	Monitor	GFX Reconstruct

LUNARG

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/

LUNARG

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

LUNARG

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!

LUNARG

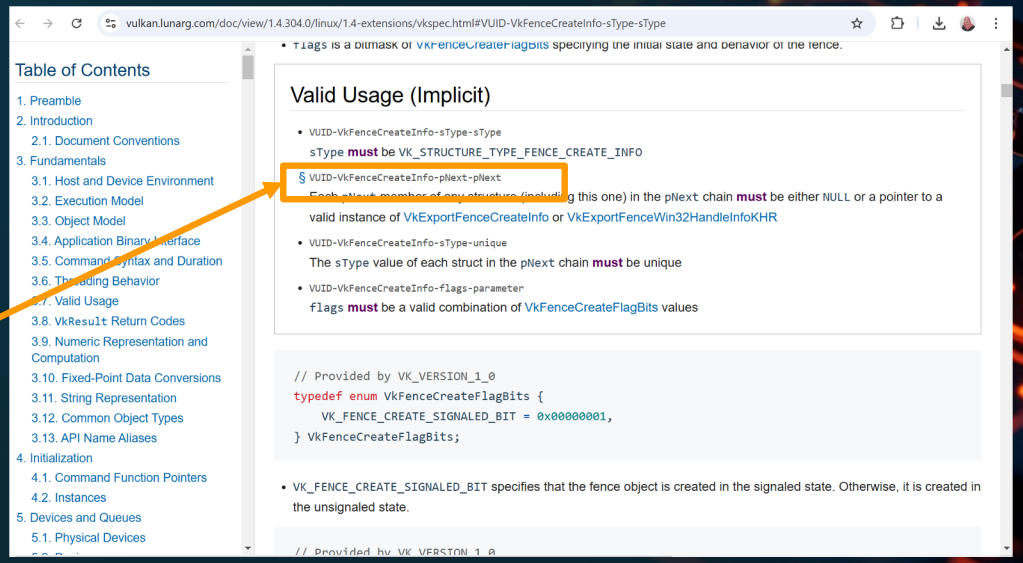
Slow Vulkan Spec Load Times!

- 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-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/1.4.304.0/linux/1.4-extensions/vkspec.html#VUID-VkFenceCreateInfo-sType-sType)

HTML Single File Specification:



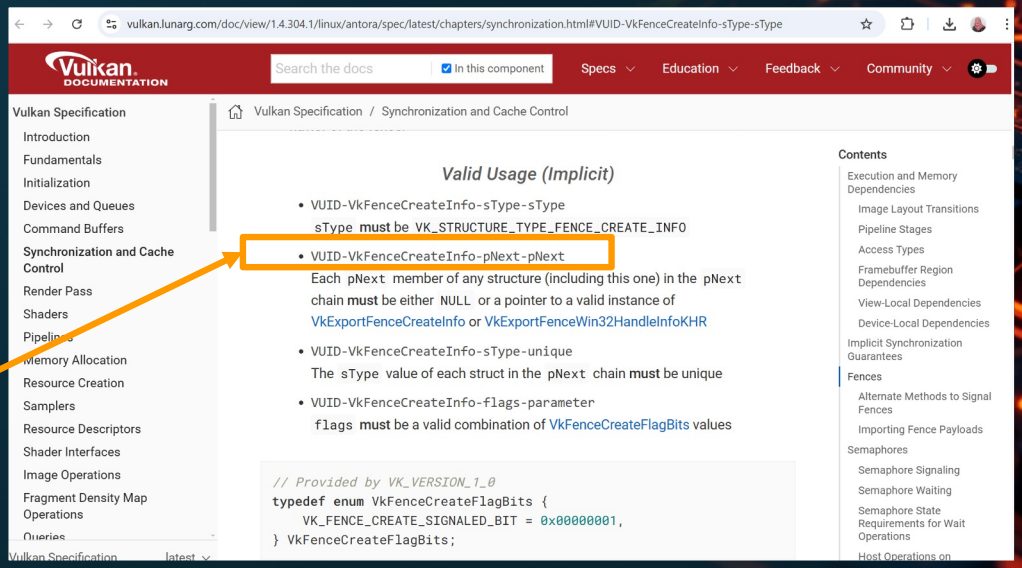
NO MORE Slow Vulkan Spec Load Times!

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

SDK Validation Layer Error:

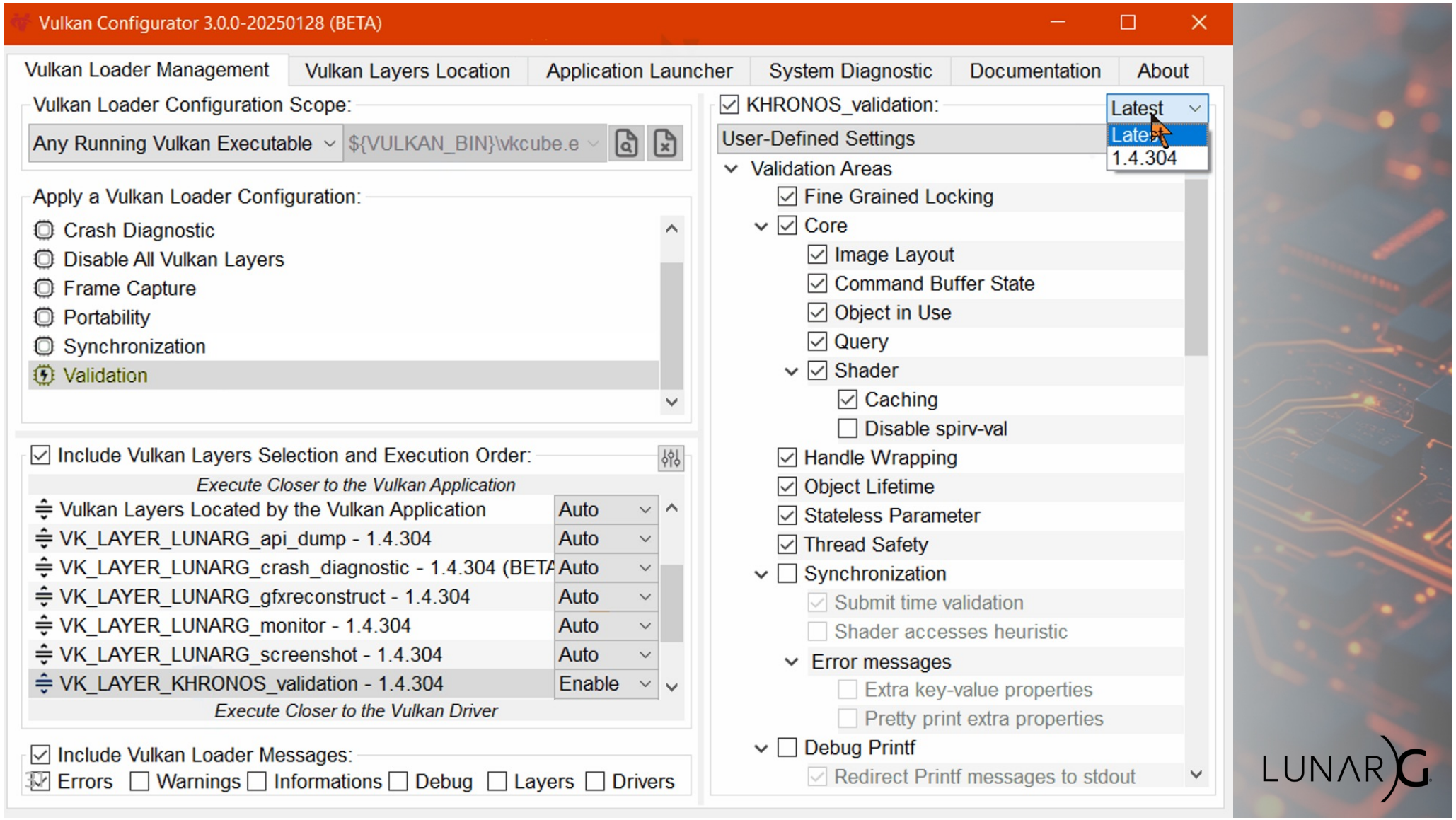
VUID-VkImageViewCreateInfo-pNext-pNext(ERROR / SPEC): msgNum: 151814321 - Validation Error: [VUID-VkImageViewCreateInfo-pNext-pNext] | MessageID = 0x90c80b1 | vkCreateImageView(): pCreateInfo->pNext chain includes a structure with unexpected VkStructureType VK_STRUCTURE_TYPE_IMAGE_CREATE_INFO. This error 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 VkSamplerYcbcrConversionInfo
(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:



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
VK_LAYER_LUNARG_api_dump - 1.4.304
VK_LAYER_LUNARG_crash_diagnostic - 1.4.304 (BETA
VK_LAYER_LUNARG_gfxreconstruct - 1.4.304
VK_LAYER_LUNARG_monitor - 1.4.304
VK_LAYER_LUNARG_screenshot - 1.4.304
VK_LAYER_KHRONOS_validation - 1.4.304
Auto
Enable
Execute Closer to the Vulkan Driver
Include Vulkan Loader Messages:
Errors
Warnings
Informations
Debug
Layers
Drivers
KHRONOS_validation:
Latest
1.4.304
User-Defined Settings
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 Printf
Redirect Printf messages to stdout
LUNARG

Gratitude for their support

LUNARG

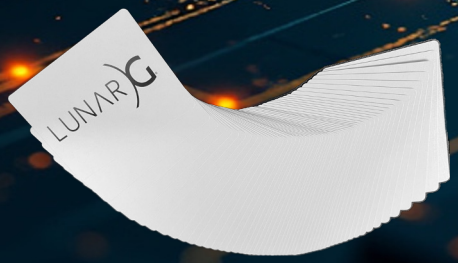
Thank you!

Actions

Download this Presentation

https://khr.io/1cr

Talk to us and get Swag!



Visit the
LunarG Sponsor Table

Take the Annual Developers Survey

https://khr.io/1cq

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)