

Vulkanised 2025

The 7th Vulkan Developer Conference
Cambridge, UK | February 11-13, 2025

Vulkan Safety Critical.

Mark Bellamy, Vulkan SC Chair
Arm



Agenda

- Intro to Vulkan SC
- Usecases
- Porting Vulkan to Vulkan SC
- Platform Availability
- Tools and Resources



What is Vulkan Safety Critical?



Vulkan SC is an API designed to enable the latest GPU technology into use cases where a GPU (or GPU driver) failure is more than an inconvenience to the user.

Vulkan SC 1.0 is:

- Specifically targeted to be deployed in devices that require functional safety certification
- Based on Vulkan 1.2
- Modified to be deterministic and robust API
- Designed to enable state-of-the-art GPU-accelerated graphics and computation

Vulkan Safety Critical Changes?



Offline Pipeline Compilation

- Pipeline Compilation is performed during Application development for target GPU
- Applications provide GPU specific pipeline cache
- Removes large complex compiler from certified driver code base

Device Creation

- Applications provide information on max number of Vulkan objects
- Applications provide information on Pipeline Pools needed
- Applications provide information on Command Pools and Buffers needed
- Avoids need for dynamic memory support

Additions

- Fault Reporting API
- Object Refresh (optional extension)

Why Vulkan SC versus OpenGL SC?

Vulkan has some great features

	Zero copy by default	Predictable command execution	Multithreading	Modern Shading Language and Features	AI/GPU Compute
OpenGL SC	If careful	No	No	No	No
Vulkan SC	Yes	Yes	Yes	Yes	Yes

... and Vulkan SC stays current:

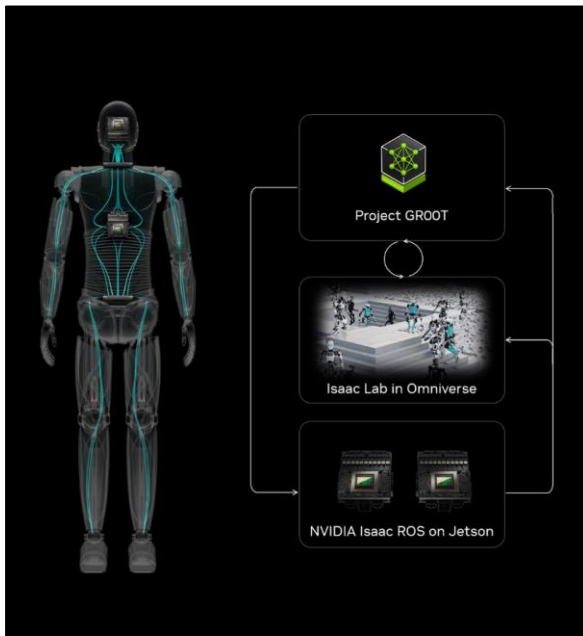
API	Last spec update
OpenGL SC	2019-07
Vulkan	2025-02
Vulkan SC	2024-12

Cameras -> AI and Visualization is everywhere you look

... and Vulkan SC is well positioned to help



Surround View System
NVIDIA DRIVE IX



Robotic environments



Avionics Graphics and Compute
CoreAVI VkCoreSC

Vulkan SC example use cases

- HMI
 - Tell tales on Vehicle Clusters
 - Visualization of ADAS or Autonomous Driving systems
 - Avionics Cockpits
 - Industrial Machine Control panels
- GPU Compute Acceleration
 - Classic Computer Vision
 - ADAS Workloads
 - Autonomous Driving Systems
- Anything where more reliability, robustness or fixed footprint is needed
 - Control Panels
 - Commercial Devices
- Not write once deploy anywhere and everywhere
 - App Markets
 - Gaming

Visualizing what your self driving car sees

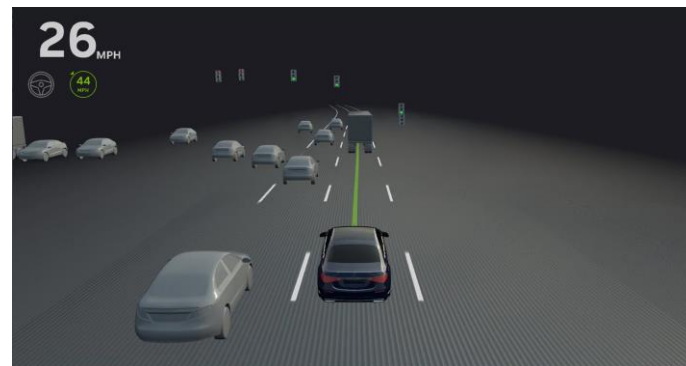
A modern car is handling and visualizing huge amounts of data in realtime:

- Backup camera
- Surround view
- Confidence View

This data needs visualizing!

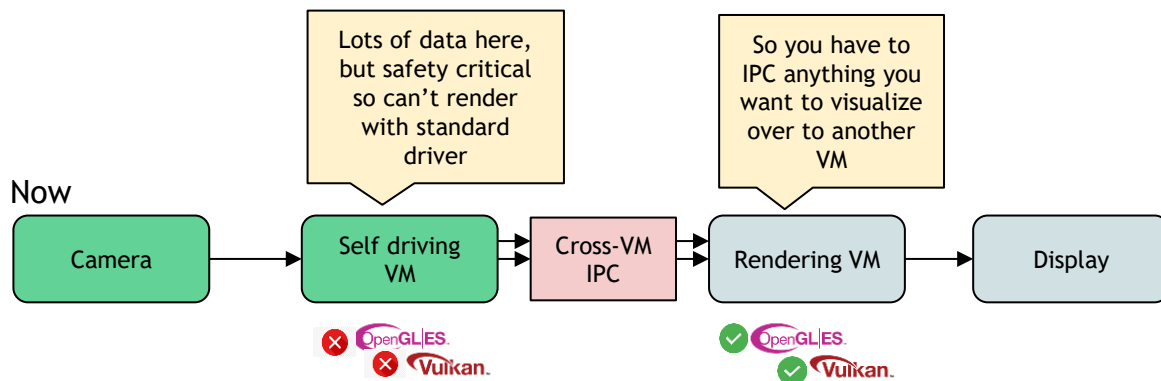
All at 60fps+.

And, the trend is toward ever-bigger data.



How sensor visualization is done today...

Today, self driving VMs are safety certified, so anything in that VM needs to also be provably safe. Thus:

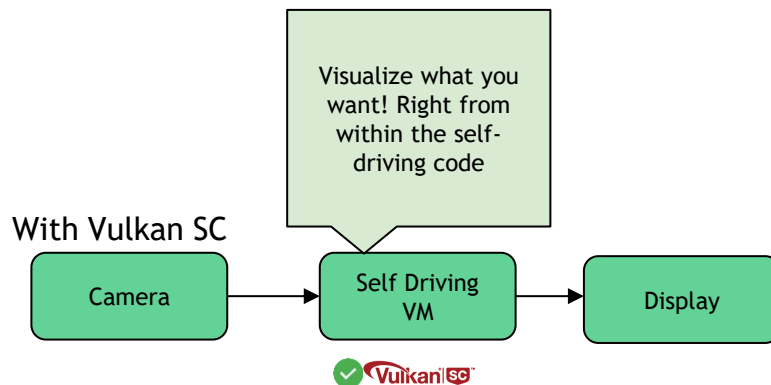


Example:

- Surround view requires streaming real time camera data across VM boundaries
- Need to decide which side of this boundary to do stitching

Vulkan SC: The right way

With Vulkan SC, you can access graphics right inside the safe VM:



Rich graphics that won't interfere with AI workloads on the same VM - magic!

Tell tales on Vehicle Clusters

- Tell Tales are safety related information
- Covered by ISO26262 certification
- Currently Either
 - Small Certified Render
 - Safety Validation Application
- Limits visual effects
- Vulkan SC gives full flexibility
 - Overlap with other content
 - Animation effects

Separate Applications for Tell Tales to main Cluster

Tell Tales

Cluster Application

Display



“Tell Tales”



Combined Application

Tell Tales

Display



Porting Vulkan to Vulkan SC is easy

Develop in Vulkan! Then:

1. Shift your pipeline and shader compilations to be done offline (~= pipeline cache)
2. Shift from dynamic object and command pool reservation to static reservation during initialization

That's it! If you want more tech details, see the Khronos website!

It doesn't take much:

Example	Purpose	# lines changed
vk_triangle	Draw simple triangle	50
Lime SC	NVIDIA's reference confidence view	1000

Vulkan SC is available on these production platforms

Core AVI

Avionics

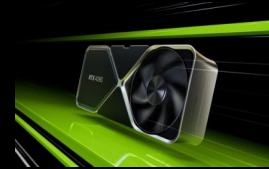
Automotive

Industrial



NVIDIA

Discrete GPU



Jetson




DriveOS



See this board running Vulkan SC content here today...

If you have an NVIDIA GPU, you can use Vulkan SC NOW



3 Easy Steps

1. Download SDK at NVIDIA:
<https://developer.nvidia.com/vulkan-sc/>
2. Build
3. Run it!

* GPU RTX 20 Series and newer

If you have a Vulkan Driver, you can still try Vulkan SC

- **Vulkan SC Emulation**

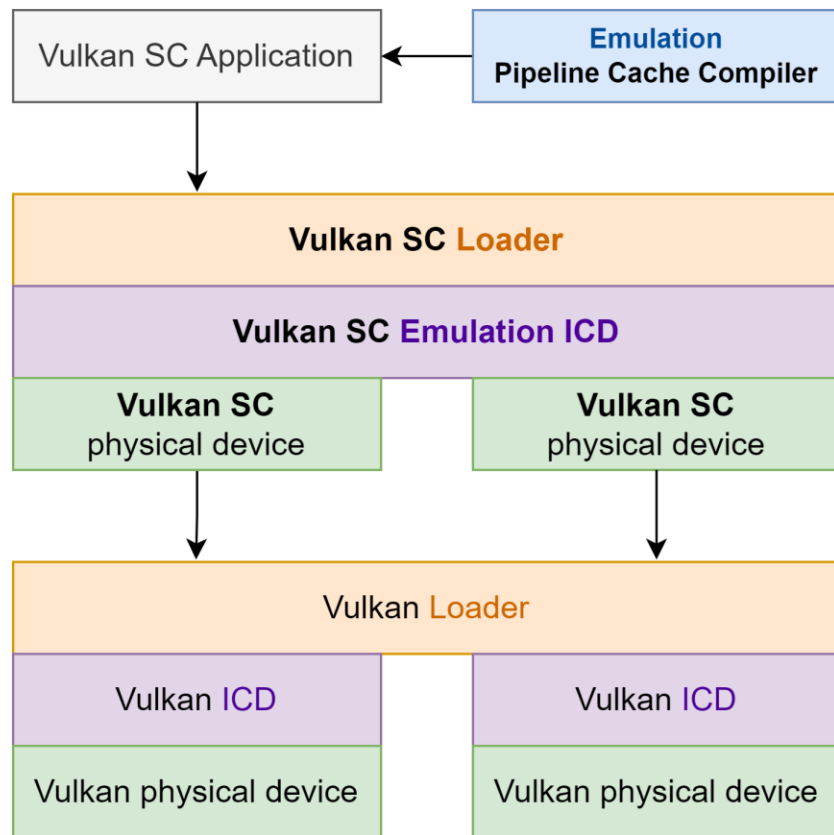
- enables running Vulkan SC apps on top of Vulkan driver stacks

- **Source code and instructions available on Github**

- <https://github.com/KhronosGroup/VulkanSC-Emulation>

- **Vulkan SC Cube demo also available**

- https://github.com/KhronosGroup/VulkanSC-Tools/tree/sc_main/cube-vksc



Vulkan SC Tools and Resources

Specification	Vulkan SC 1.0 Specification and Extensions (PDF , HTML)
VulkanSC-Headers	Official API headers
VulkanSC-Loader	ICD loader, documentation and tests (Linux, Windows and QNX)
VulkanSC-ValidationLayers	Validation layers to verify applications are making correct use of the API
VulkanSC-Tools	Tools and utilities: vulkanscinfo command line tool, the mock ICD, and the device simulation layer.
Emulation Layer	Develop Vulkan SC application on top of Vulkan Implementations
Cube Sample	Vulkan SC Sample application that renders a Cube.

Questions?

For more info:

CoreAVI

https://coreavi.com/product_category/safety-critical-graphics-and-compute/



Khronos

<https://www.khronos.org/vulkansc/>



<https://khr.io/vkscdiscord>



NVIDIA

<https://developer.nvidia.com/vulkan-sc/>

