Vulkan SDK Tools
to Use and Create Vulkan Profiles

Christophe Riccio
LunarG, Inc.

SIGGRAPH 2022
VANCOUVER+ 8-11 AUG

Agenda

- A perspective on why Vulkan Profiles are relevant for Vulkan developers
- A brief presentation of the Vulkan Profiles Toolset components
- A tutorial on how to use the Vulkan Profiles Toolset components

Please, ask your questions in the chat. (Karen and John will moderate it)

- Based on The Vulkan Profiles Toolset solution whitepaper
  - https://www.lunarg.com/wp-content/uploads/2022/03/The-Vulkan-Profiles-Toolset-Solution-FEB2022.pdf
- Slides available on LunarG website
  - https://www.lunarg.com/news-insights/white-papers/vulkan-sdk-tools-to-use-and-create-vulkan-profiles/

Designing the Vulkan Profiles Toolset

What’s a Vulkan Profile?

- A collection of Vulkan Capabilities
    - Extensions
    - Features
    - Properties
    - Queue properties
    - Formats
    - Etc…

Creating portable Vulkan Applications in terms of Vulkan capabilities

Vulkan Profiles:

- A convention
  - **Explicit** Vulkan capability requirements (of engines) and/or supports (of platforms).
- A toolset
  - Easier Vulkan development for a selected range of **actual** ecosystem devices.

Vulkan Profiles usages:

- *Roadmap profiles*: to express guidance on the future direction of Vulkan devices.
- *Platform profiles*: to express the Vulkan support available on different platforms.
- *Device profiles*: to express the Vulkan support of a single Vulkan driver for a Vulkan device.
- *Architecture profiles*: to express the Vulkan support of a class of GPUs.
- *Engine profiles*: to express some rendering code paths requirements of an engine.
- Etc.

The Vulkan Profiles Toolset Solution

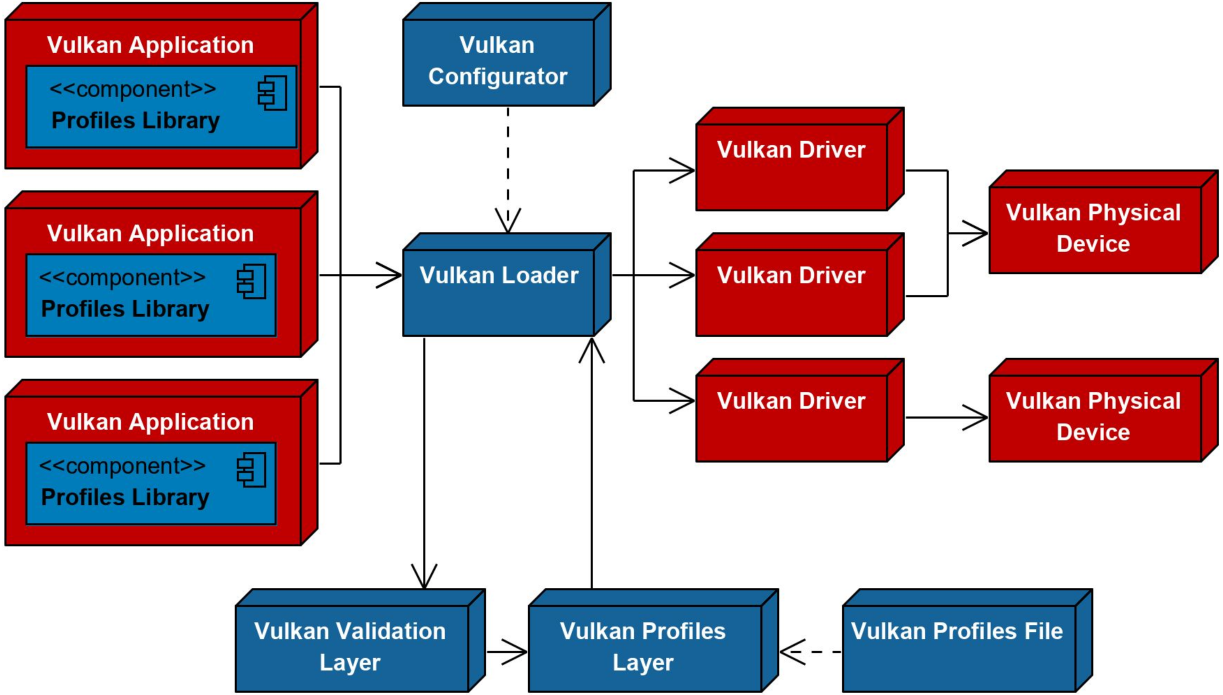
The Vulkan Profiles Toolset Components:

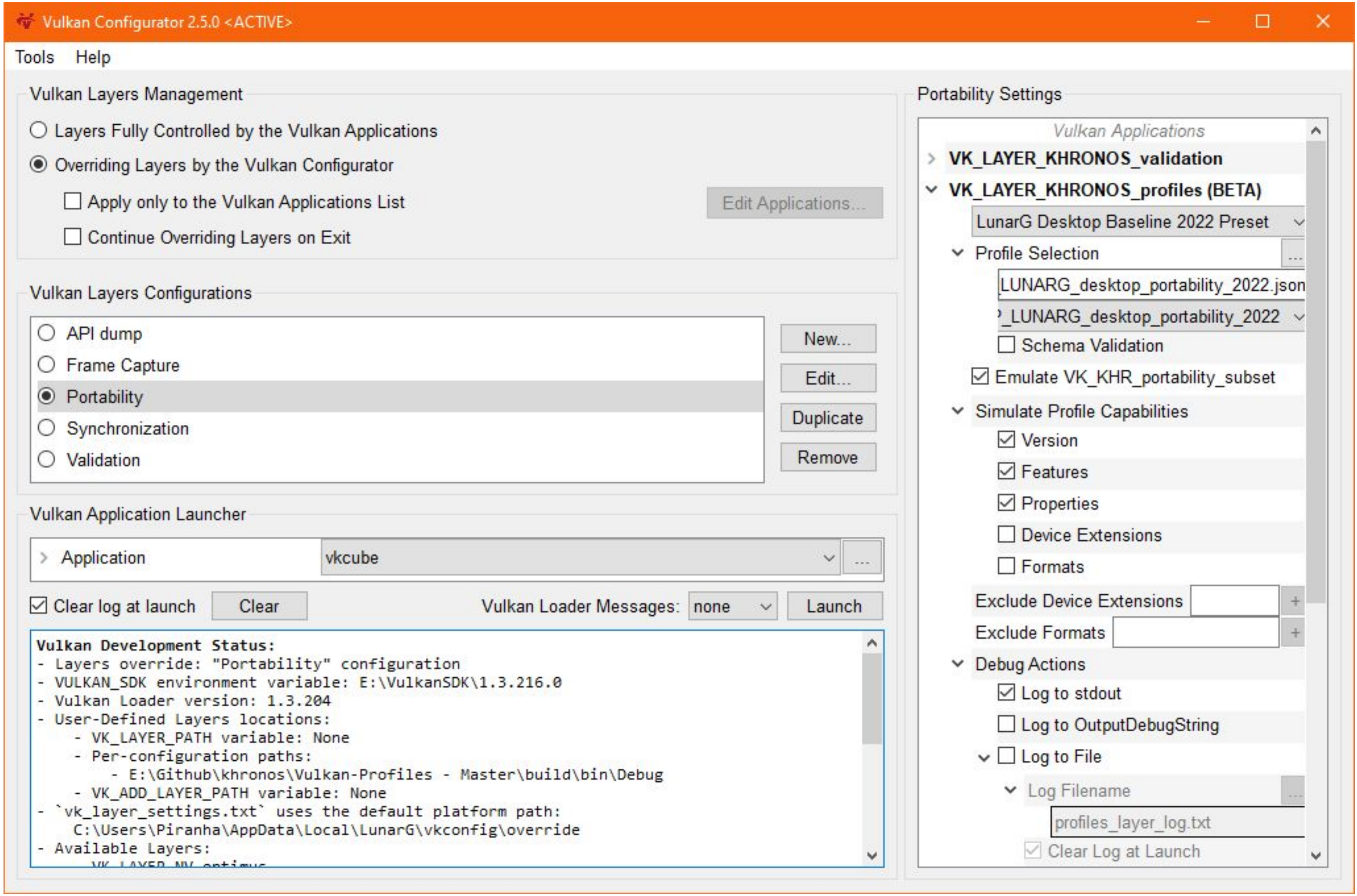
- Vulkan Profiles schema
- Vulkan Profiles file generation (new with August Vulkan SDK coming soon)
- Vulkan Profiles comparison table
- Vulkan Profiles layer
- Vulkan Profiles library

Requires a Vulkan 1.0 driver that supports the `VK_KHR_get_physical_device_properties2` extension.

Delivered at BETA development stage.

Deployment of the Toolset components





Vulkan Configurator 2.5.0 <ACTIVE>

Vulkan Layers Management

○ Layers Fully Controlled by the Vulkan Applications
● Overriding Layers by the Vulkan Configurator
□ Apply only to the Vulkan Applications List
□ Continue Overriding Layers on Exit

Vulkan Layers Configurations

○ API dump
○ Frame Capture
● Portability
○ Synchronization
○ Validation

Vulkan Application Launcher

Application: vkcube

☑ Clear log at launch

Vulkan Loader Messages: none

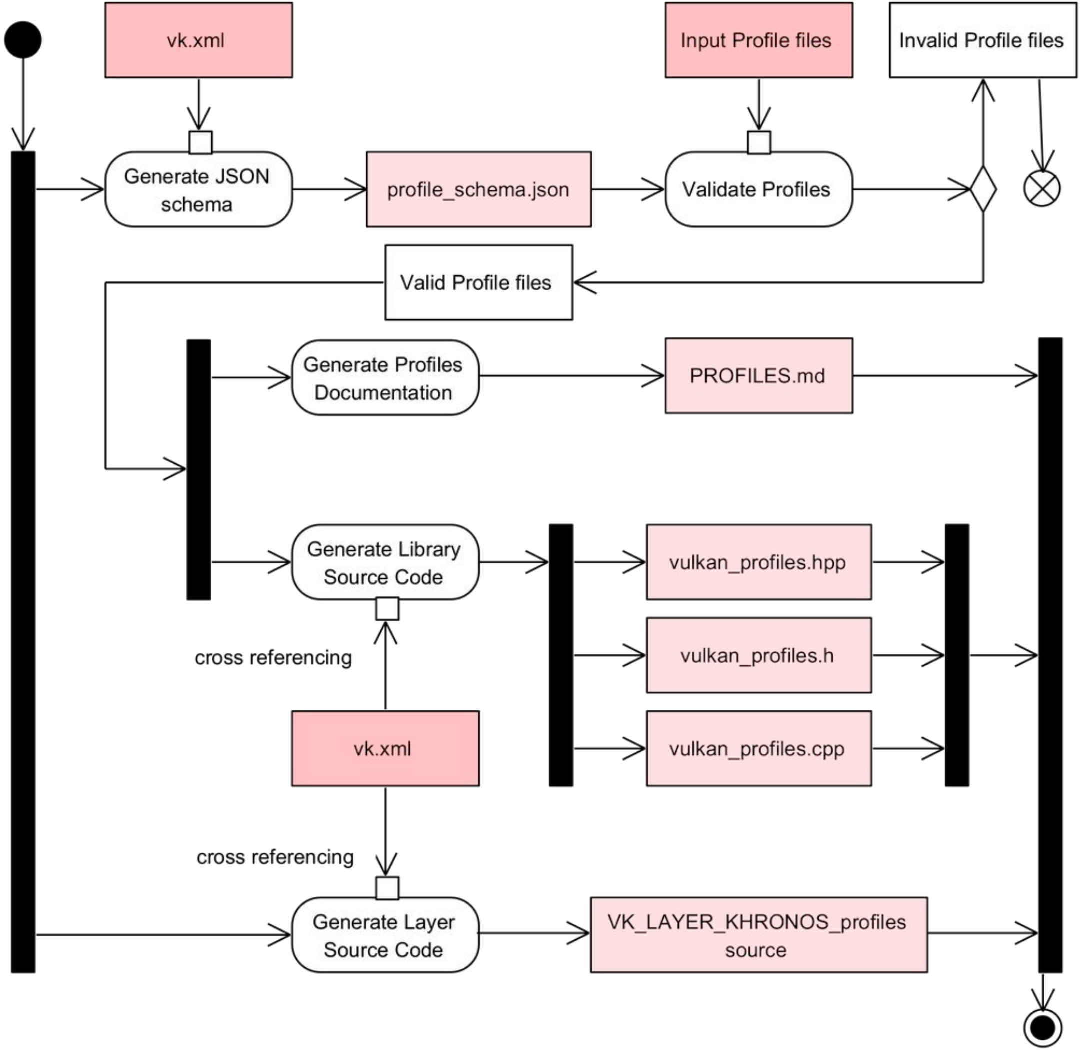
```
Vulkan Development Status:
- Layers override: "Portability" configuration
- VULKAN_SDK environment variable: E:\VulkanSDK\1.3.216.0
- Vulkan Loader version: 1.3.204
- User-Defined Layers locations:
    - VK_LAYER_PATH variable: None
    - Per-configuration paths:
        - E:\Github\khronos\Vulkan-Profiles - Master\build\bin\Debug
    - VK_ADD_LAYER_PATH variable: None
- `vk_layer_settings.txt` uses the default platform path:
    C:\Users\Piranha\AppData\Local\LunarG\vkconfig\override
- Available Layers:
    VK_LAYER_NV_optimus
```

Portability Settings

Vulkan Applications

- VK_LAYER_KHRONOS_validation
- VK_LAYER_KHRONOS_profiles (BETA)
    - LunarG Desktop Baseline 2022 Preset
    - Profile Selection
        - LUNARG_desktop_portability_2022.json
        - VP_LUNARG_desktop_portability_2022
        - □ Schema Validation
    - ☑ Emulate VK_KHR_portability_subset
    - Simulate Profile Capabilities
        - ☑ Version
        - ☑ Features
        - ☑ Properties
        - □ Device Extensions
        - □ Formats
    - Exclude Device Extensions
    - Exclude Formats
    - Debug Actions
        - ☑ Log to stdout
        - □ Log to OutputDebugString
        - □ Log to File
            - Log Filename
                - profiles_layer_log.txt
        - ☑ Clear Log at Launch

NARG


vk.xml
Input Profile files
Invalid Profile files
Generate JSON schema
profile_schema.json
Validate Profiles
Valid Profile files
Generate Profiles Documentation
PROFILES.md
Generate Library Source Code
vulkan_profiles.hpp
vulkan_profiles.h
vulkan_profiles.cpp
cross referencing
vk.xml
cross referencing
Generate Layer Source Code
VK_LAYER_KHRONOS_profiles source

The Vulkan Profiles Toolset Components

The Vulkan Profiles schema

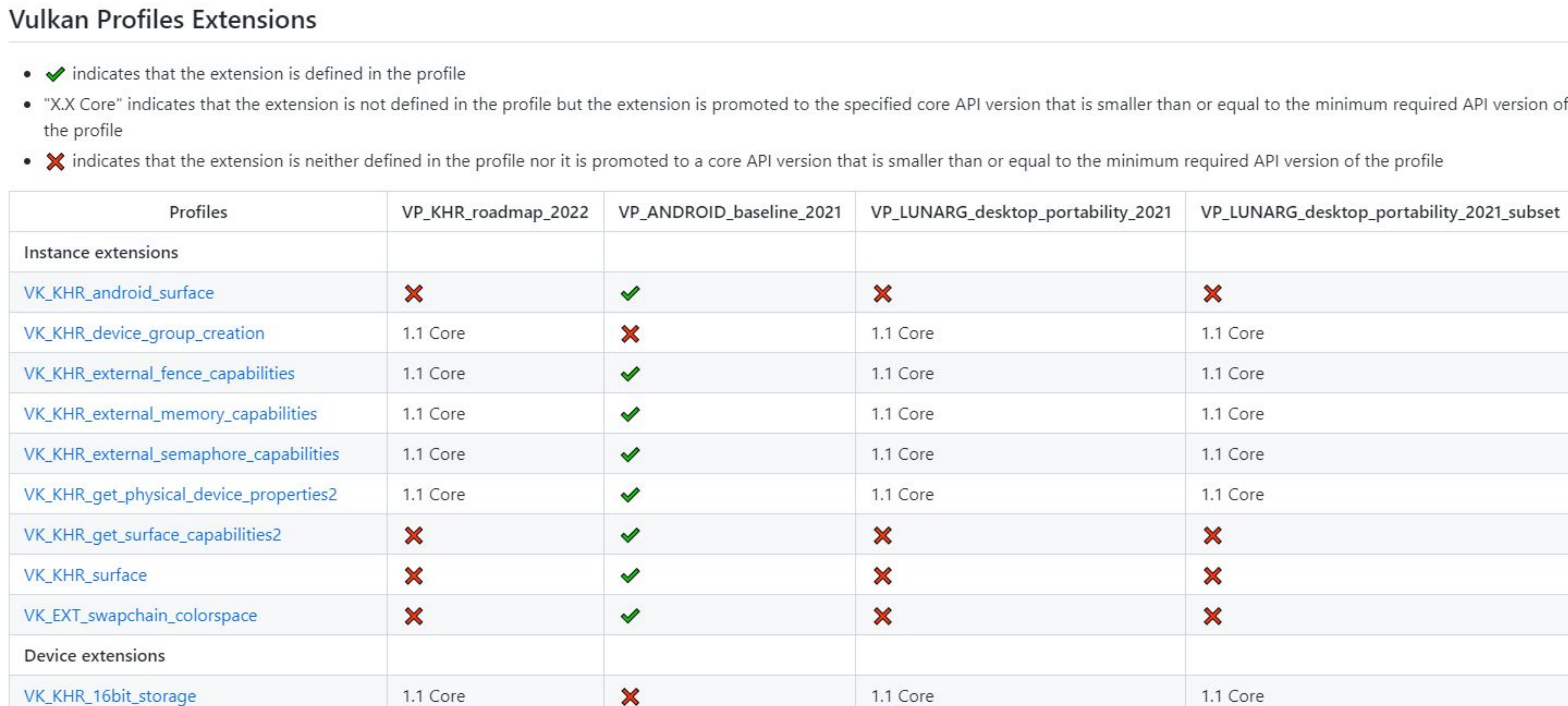
- A JSON data format to communicate about Vulkan capabilities:
    - extensions, features, properties, formats, and queue properties.
- Updated with each new Vulkan Header revision
    - Each revision of Vulkan API is represented by a schema that supersedes older versions of Vulkan API.

The Vulkan Profiles file generation

- Python script to combine multiple profiles files
  - Either “Intersect” or “Union”
- Used to generate:
  - VP_LUNARG_desktop_baseline_2022
  - VP_LUNARG_desktop_portability_2022 (baseline_2022 equivalent with portability extension)

- Vulkaninfo supports export of profiles JSON files
- GPUInfo.org supports export to profiles JSON files

The Vulkan Profiles comparison table



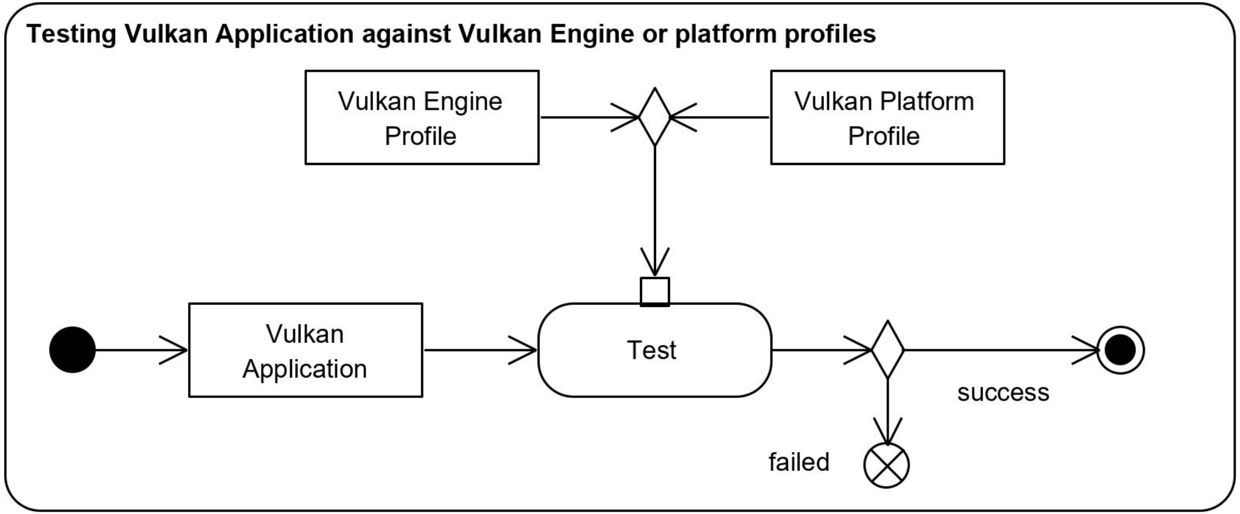
Vulkan Profiles Extensions

- ✓ indicates that the extension is defined in the profile
- "X.X Core" indicates that the extension is not defined in the profile but the extension is promoted to the specified core API version that is smaller than or equal to the minimum required API version of the profile
- ✗ indicates that the extension is neither defined in the profile nor it is promoted to a core API version that is smaller than or equal to the minimum required API version of the profile

Profiles	VP_KHR_roadmap_2022	VP_ANDROID_baseline_2021	VP_LUNARG_desktop_portability_2021	VP_LUNARG_desktop_portability_2021_subset
Instance extensions				
VK_KHR_android_surface	✗	✓	✗	✗
VK_KHR_device_group_creation	1.1 Core	✗	1.1 Core	1.1 Core
VK_KHR_external_fence_capabilities	1.1 Core	✓	1.1 Core	1.1 Core
VK_KHR_external_memory_capabilities	1.1 Core	✓	1.1 Core	1.1 Core
VK_KHR_external_semaphore_capabilities	1.1 Core	✓	1.1 Core	1.1 Core
VK_KHR_get_physical_device_properties2	1.1 Core	✓	1.1 Core	1.1 Core
VK_KHR_get_surface_capabilities2	✗	✓	✗	✗
VK_KHR_surface	✗	✓	✗	✗
VK_EXT_swapchain_colorspace	✗	✓	✗	✗
Device extensions				
VK_KHR_16bit_storage	1.1 Core	✗	1.1 Core	1.1 Core

The Vulkan Profiles layer

- Simulate vs Emulate:
  - Downgrade the Vulkan developer’s system capabilities
  - Only emulates VK_KHR_portability_subset
  - No emulation of mobile specific feature on desktop hardware
- Additional checking of the validity of a profile definition



Vulkan Profiles library

- Check Vulkan Profile support by the platform
- Create VkInstance and `VkDevice` with the profile features and extension enabled
  - Simplifies Vulkan initialization code
- Reflection of the Vulkan Profiles
- A KhronosGroup/Vulkan-Samples sample is available on github for demonstrating Vulkan Profiles library usage.

- The Vulkan Profiles API is not part of the Vulkan API
  - It ships as part of the Vulkan application code

Tutorial!