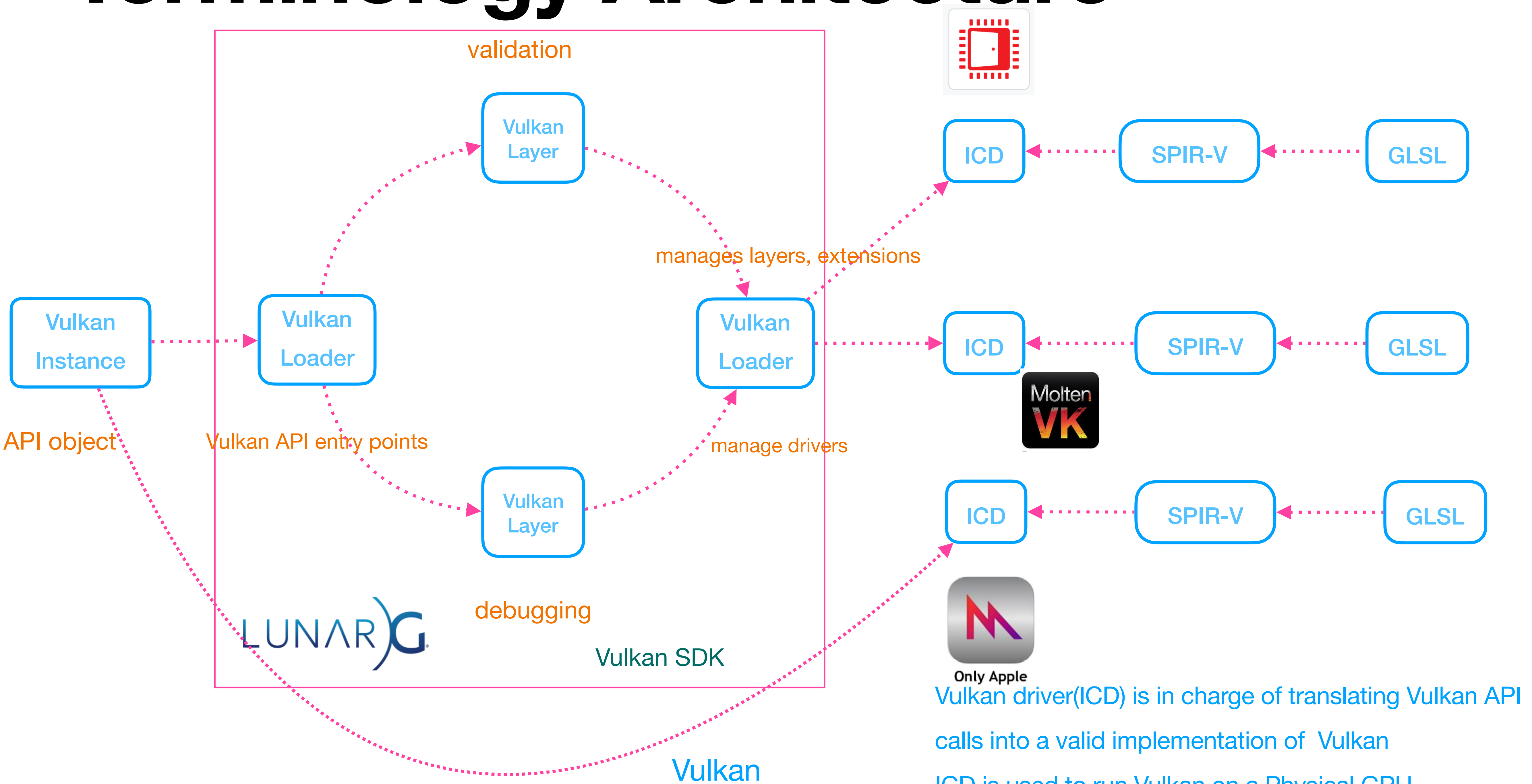
# Terminology Architecture



Vulkan driver(ICD) is in charge of translating Vulkan API calls into a valid implementation of Vulkan

ICD is used to run Vulkan on a Physical GPU

Vulkan SDK includs a MoltenVK runtime library for macOS

# Install Vulkan SDK on Mac

Lina Liu Nov.12 2022

# Vulkan SDK Introduction

## What is Khronos Vulkan API?

1. What is Khronos Vulkan API?
    1. Vulkan API is explicit
    2. Vulkan API is low-overhead
    3. Vulkan API is cross-platform graphics API
    4. Vulkan API is cross-platform compute API
    5. Vulkan API over operating system
    6. Vulkan API on wide variety of devices as PC/mobile/embeded platforms

# Vulkan SDK Introduction

## What does Vulkan SDK do?

2. What does Vulkan SDK do?
    1. Vulkan SDK enables Vulkan Developer to develop Vulkan applications
3. What does Vullkan SDK include?
    1. Vulkan API usage validation
    2. Vulkan Layer configuration
    3. SPIR-V shader compilation
    4. SPIR-V shader optimization
    5. SPIR-V shader validation
    6. Vulkan System report

# MoltenVK

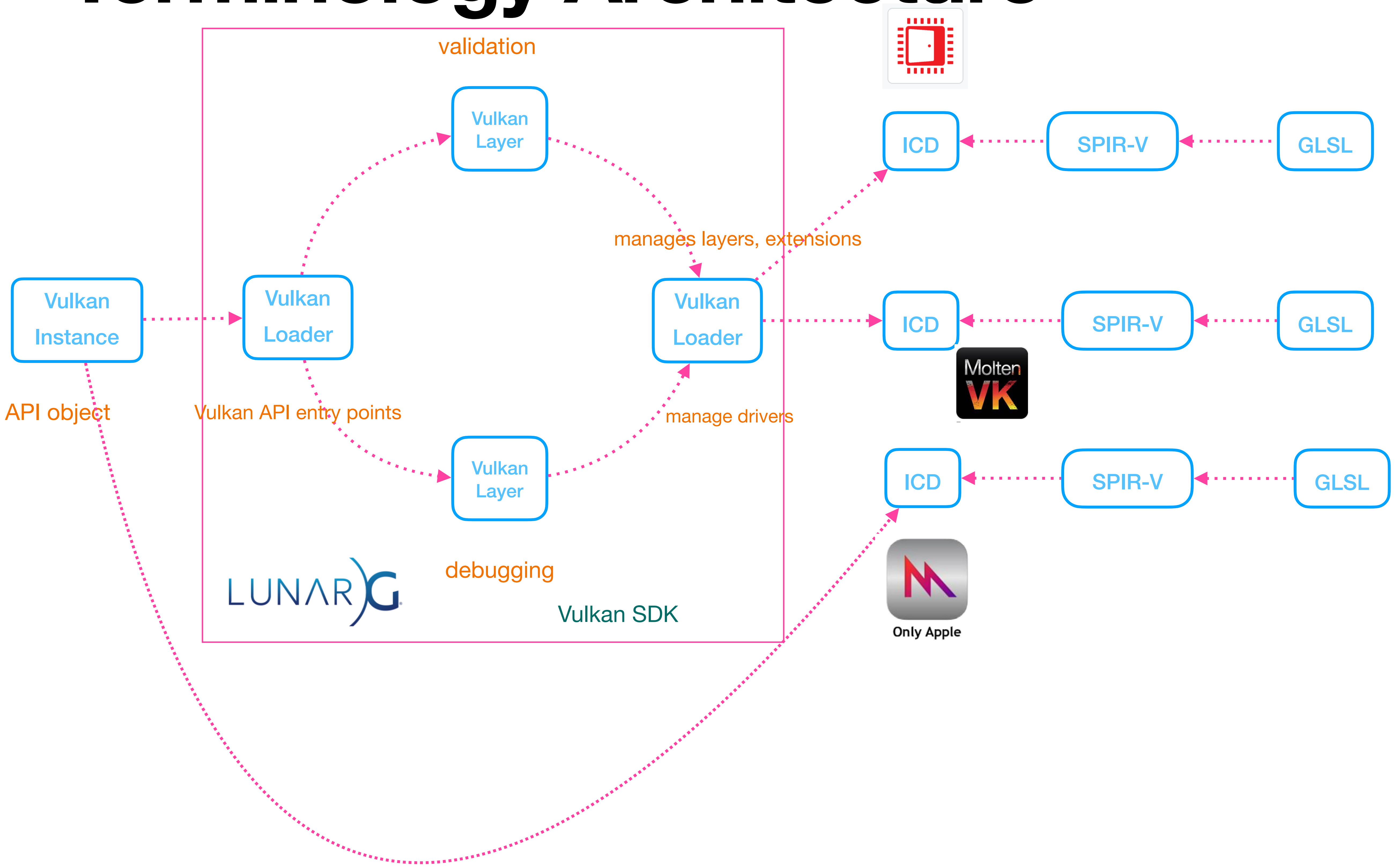
## Vulkan SDK that used on macOS, iOS platforms

4. How to use MoltenVK?
    1. Link directly to the MoltenVK static or dynamic library
        1. You can direct access to the Vulkan API
        2. It is not practical if you wish to maintain portability of your Vulkan rendering code across platforms
        3. You will sacrifice the ability to use the Vulkan validation layers
        4. However, this is the only way to use MoltenVK on mobile devices.
        5. On mobile devices, XCFramework is provided as a static library that can be linked directly to your application.
    2. Use MoltenVK dynamic library in conjunction with the Vulkan loader.
        1. This is generally used on desktop applications
        2. In this mode, you link only to the Vulkan loader, and not the MoltenVK library directly.
        3. You will include the MoltenVK and the Vulkan Loader dynamic libraries in your application bundle when distributing your software.
        4. Use Vulcan loader and Vulcan validation layers instead of linked to the static MoltenVK library, why?
            1. You can debugging your Vulkan rendering code by using tremendous boon

# Terminology

| Term | Description |
|---|---|
| **ICD** | Installable Client Driver, on desktop side, the MoltenVK take this role |
| **GLSL** | OpenGL Shader Language |
| **Vulkan Instance** | The Vulkan API object that stores all per-application states |
| **Vulkan Layer** | A library designed to work as a plug-in for the loader.<br>Provide validation and debugging functionality to applications |
| **Vulkan Loader** | A library 1. implements the Vulkan API entry points and<br>2. manages layers, extensions, and drivers<br>It is found in the independent hardware vendor driver installs |
| **SPIR-V** | Standard Portable Intermediate Representation<br>A cross-API intermediate language(IL) represents parallel compute and graphics programs |

# Terminology Architecture



## Developing Vulkan Application for macOS

5. Vulkan SDK includs a MoltenVK runtime library for macOS
6. How to integrate the MoltenVK runtime library into a game or application?
    1. https://github.com/KhronosGroup/MoltenVK/blob/master/Docs/MoltenVK_Runtime_UserGuide.md

# Use of the Vulkan SDK

## SDK Versioning

7. SDK version: v.w.xx.0
    1. “v” Vulkan major version
    2. “w” Vulkan minor version
    3. ‘xx’ vulkan patch version
8. Vulkan SDK and Vulkan Application must be matched.
9. Vulkan Instance Version.
    1. This is the version of Vulcan loader, run vulkaninfo to see the version
10. Physical device has apiVersion, run vulkaninfo to see the version

# Install the SDK

11. Download the SDK
    1. https://vulkan.lunarg.com/sdk/home
12. What happened when Installing the SDK?
    1. Mounting a disk image file
    2. Double-click the InstallVulkan icon to start the installer
    3. Choose the destination folder to install the Vulkan SDK
    4. Update the Vulkan Loader and MoltenVK libraries in /usr/local
    5. Optional components be automatically downloaded from the cloud and installed them as part of the SDK.
        1. Com.lunarg.vulkan.usr
            1. Copy loader, icd, and tools into system wide /usr/ locations
        2. com.lunarg.vulkan.sdl2
            1. SDL2 both 32/64-bit library
        3. com.lunarg.vulkan.glm
            1. GLM ( 3D Math Library) headers
        4. com.lunarg.vulkan.volk
            1. Volk(Vulkan Meta Loader) library
        5. Com.lunarg.vulkan.vma
            1. Vulkan Memory Allocator library.
13. Install from command line
    1. sudo ./InstallVulkan.app/Contents/MacOS/InstallVulkan --root "installation path" --accept-licenses --default-answer --confirm-command install
    2. Sudo .install_vulkan.py
14. Uninstall from the command line
    1. sudo ./MaintenanceTool.app/Contents/MacOS/MaintenanceTool --confirm-command purge
    2. Sudo ./uninstall.sh

# SDK System Paths

15. SDK system paths is : /usr/local
    1. Vulkan installer program will copy following items to /usr/local:
        1. Loader
        2. Command line tools
        3. MoltenVK ICD
    2. If you install Vulcan SDK through cmd line, then you need to set PATH, DYLD_LIBRARY_PATH, VK_LAYER_PATH, and VK_ICD_FILENAMES environments.

# Directories inside Vulkan SDK

| Directory | Description |
|---|---|
| **Applications** | Standalone Vulkan demos and tools |
| **MoltenVK** | MoltenVK frameworks and libraries for macOS |
| **macOS** | VULKAN_SDK tree; |
| **macOS/bin** | Vulkan and Shader tool executables |
| **macOS/share/vulkan/explicit_layer.d** | Explicit layers that can be referenced with the VULKAN_LAYER_PATH environment variable |
| **macOS/share/vulkan/icd.d** | MoltenVK ICD manifest JSON file |
| **macOS/share/vulkan/registry** | Home of the vk.xml valid usage.json and files |
| **macOS/share/vulkan/config** | Home of the Vulcan profiles json files |
| **macOS/Frameworks** | Vulkan loader framework for Xcode |
| **macOS/include** | Vulkan and Shader tool header files |
| **macOS/lib** | Vulkan and Shader tool libraries. |

# Set up the runtime environment manually

1. Set convenience variable to the SDK
    1. export VULKAN_SDK=$vulkansdk/macOS
2. Add bin directory to path to make it easy to run Vulcan info or cube tools
    1. export PATH=$VULKAN_SDK/bin:$PATH
3. Add lib directory to your DYLD_LIBRARY_PATH so that programs find the Vulkan Loader library.
    1. export DYLD_LIBRARY_PATH=$VULKAN_SDK/lib:$DYLD_LIBRARY_PATH
4. Tell the Vulkan Loader where to find a Vulkan Driver:
    1. export VK_ICD_FILENAMES=$VULKAN_SDK/macOS/share/vulkan/icd.d/MoltenVK_icd.json
5. Tell the Vulkan Loader where to find the Vulkan SDK layers:
    1. export VK_LAYER_PATH=$VULKAN_SDK/macOS/share/vulkan/explicit_layer.d