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

Vulkan Video Decoding with .NET8 leveraging Silk.NET and Avalonia

Andrew Bennett
Senior Software Engineer
L3Harris ForceX

Vulkanised 2025

Looking for a cross platform and multiple OS solution

AMD

NVIDIA

intel

Vulkan

.NET 8.0

Windows

OS	Versions	Architectures	Lifecycle
Nano Server	2022, 2019	x64	Lifecycle
Windows	11 24H2 (IoT), 11 24H2 (E), 11 24H2, 11 23H2, 11 22H2 (E), 10 22H2, 10 21H2 (E), 10 21H2 (IoT), 10 1809 (E), 10 1607 (E)	Arm64, x64, x86	Lifecycle
Windows Server	23H2, 2022, 2019, 2016, 2012-R2, 2012	x64, x86	Lifecycle
Windows Server Core	2022, 2019, 2016, 2012-R2, 2012	x64, x86	Lifecycle

Linux

OS	Versions	Architectures	Lifecycle
Alpine	3.20, 3.19, 3.18	Arm32, Arm64, x64	Lifecycle
CentOS Stream	9	Arm64, ppc64le, s390x, x64	Lifecycle
Debian	12	Arm32, Arm64, x64	Lifecycle
Fedora	40	Arm32, Arm64, x64	Lifecycle
openSUSE Leap	15.6, 15.5	Arm64, x64	Lifecycle
Red Hat Enterprise Linux	9, 8	Arm64, ppc64le, s390x, x64	Lifecycle
SUSE Enterprise Linux	15.6, 15.5	Arm64, x64	Lifecycle
Ubuntu	24.10, 24.04, 22.04, 20.04	Arm32, Arm64, x64	Lifecycle

Android

OS	Versions	Architectures	Lifecycle
Android	15, 14, 13, 12.1, 12	Arm32, Arm64, x64	Lifecycle

Video Decoders

Video Decoders used before Vulkan, other ones to note are VAAPI and VDPAU

SDK	LINUX	WINDOWS	H264	HEVC
NVIDIA Video SDK	X	X	X	X
Intel Media SDK (Legacy)	X	X	X	X
Intel One API - OneVPL	X	X	X	X
Microsoft Media Foundation		X	X	X
Direct Show		X	X	
DXVA		X	X	
Vulkan SDK	X	X	X	X

Decoders	SPS	PPS
DXVA	Required	Required
VAAPI	Required	Required
VDPAU	Required	Required
VULKAN	Required	Required

Graphics API used with .NET before Vulkan

- OpenGL

Vulkanised 2025

Silk.NET takes the released Khronos specifications and generates the C# code required to use the Vulkan API as well as other technologies:

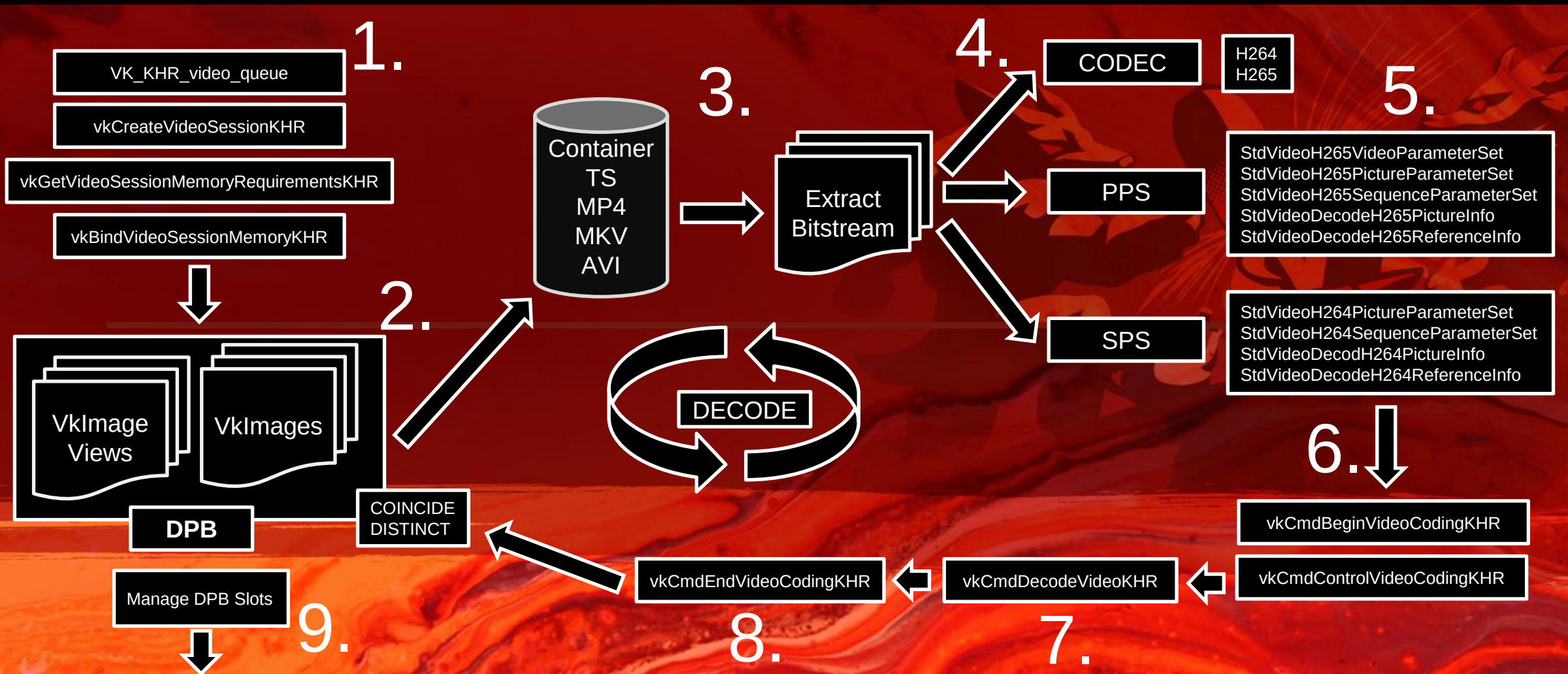
- Compute
- Graphics
- Video

Getting Started with Video Decoding

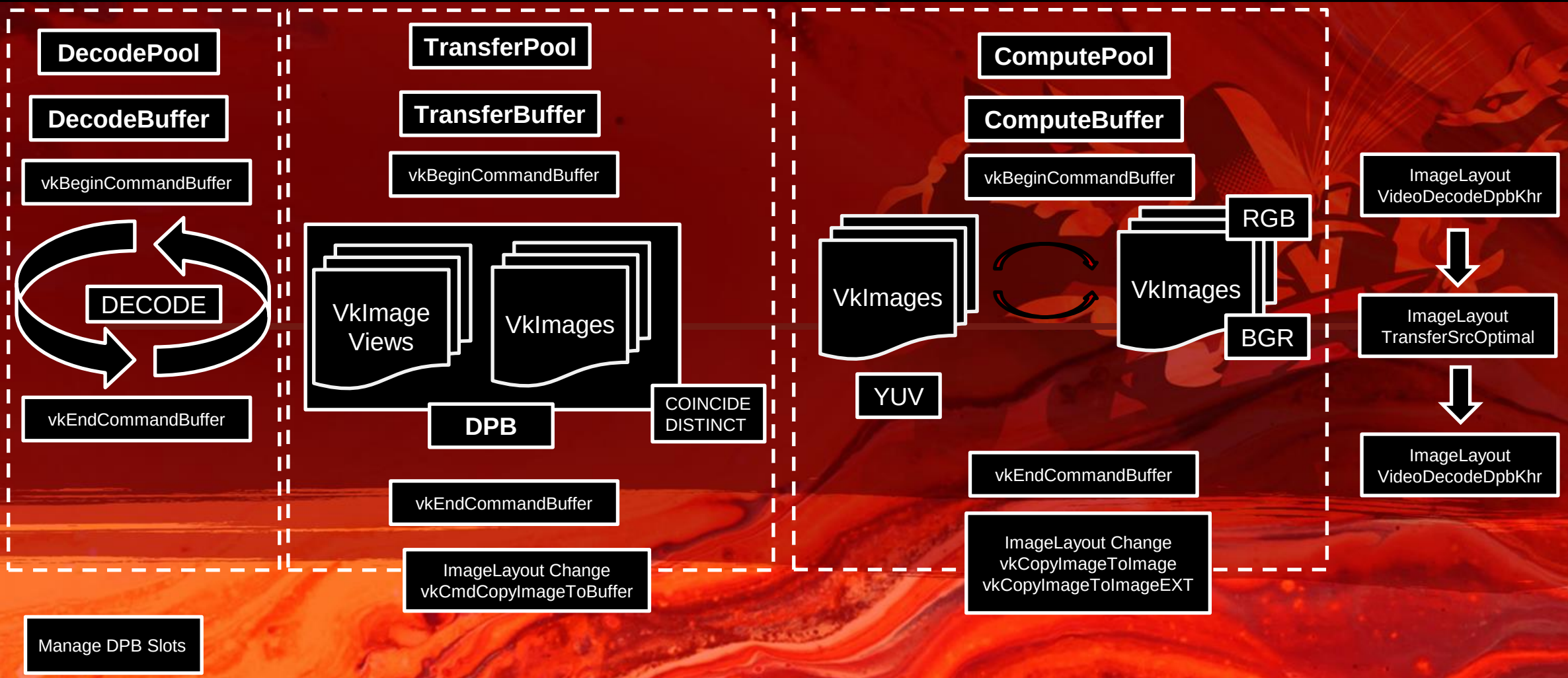
- First Step extract video codec(H264, HEVC) from a video container:
  - TS
  - MP4
  - MKV
  - AVI
- Next need to fill the following Vulkan structures depending on Codec

Vulkanised 2025

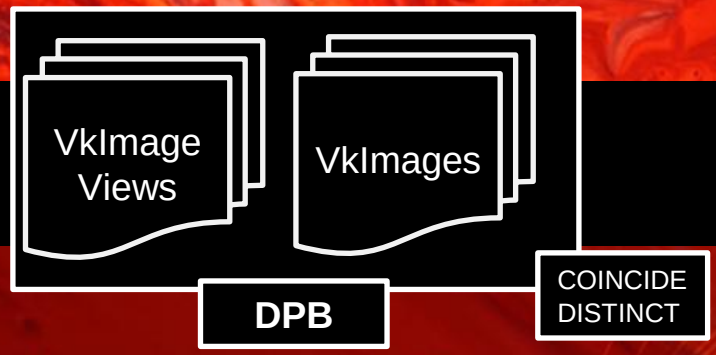
Getting Started with Video Decoding



Submitting all queued work

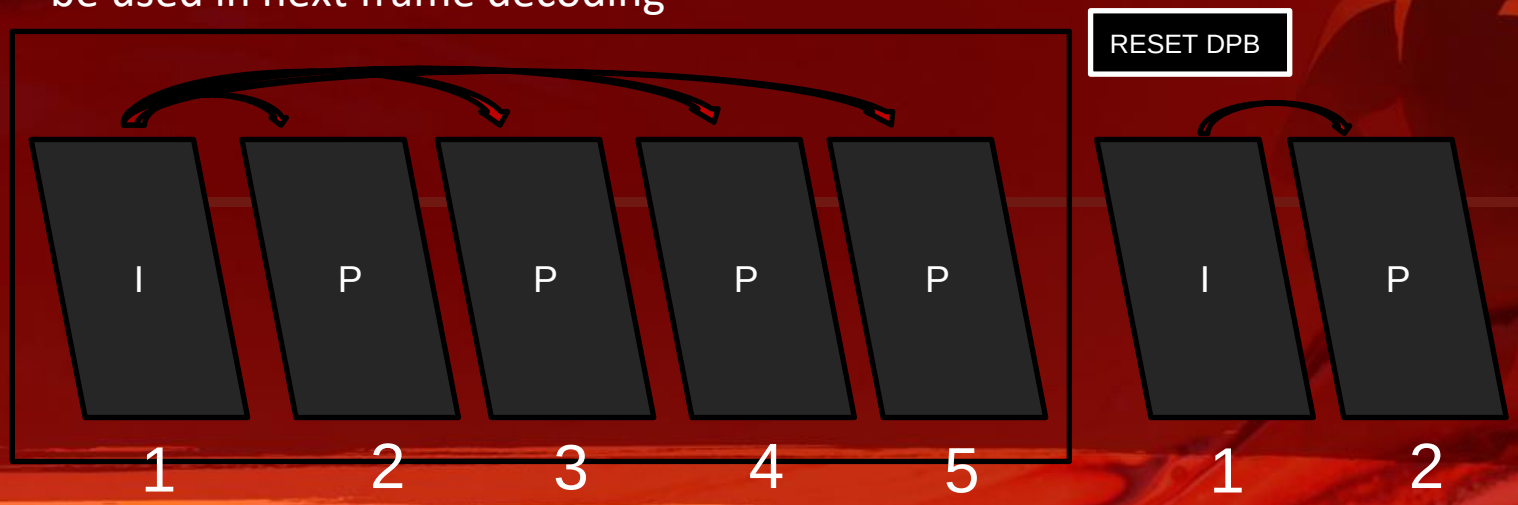


Vulkanised 2025



Decoded Picture Buffer

After each Image that was decoded the ImageLayout will be changed and then need to set ImageLayout back to VideoDecodeDpbKhr to be used in next frame decoding



In the VkGetPhysicalDeviceVideoCapabilitiesKHR call this will determine if the same image can be used for DPB and decoder output

- VkVideoDecodeCapabilityFlagBitsKHR
- DPB_AND_OUTPUT_DISTINCT_BIT_KHR
  - INTEL
  - AMD
- DPB_AND_OUTPUT_COINCIDE_BIT_KHR
  - NVIDIA

References:

- The H.264 Advanced Video Compression Standard by IAIN E. Richardson
- https://www.itu.int/rec/T-REC-H.265-202108-S/
- https://www.itu.int/rec/T-REC-H.264-202108-I/

Vulkanised 2025

Required Avalonia Packages

```xml
<Project Sdk="Microsoft.NET.Sdk">
  <PropertyGroup>
    <OutputType>WinExe</OutputType>
    <TargetFramework>net8.0</TargetFramework>
    <Nullable>enable</Nullable>
    <BuiltInComInteropSupport>true</BuiltInComInteropSupport>
    <ApplicationManifest>app.manifest</ApplicationManifest>
    <AvaloniaUseCompiledBindingsByDefault>true</AvaloniaUseCompiledBindingsByDefault>
    <AllowUnsafeBlocks>true</AllowUnsafeBlocks>
  </PropertyGroup>

  <ItemGroup>
    <Folder Include="Models\" />
    <AvaloniaResource Include="Assets\**" />
  </ItemGroup>

  <ItemGroup>
    <PackageReference Include="System.Reactive" Version="5.0.0" />
    <PackageReference Include="Avalonia" Version="11.2.1" />
    <PackageReference Include="Avalonia.Desktop" Version="11.2.1" />
    <PackageReference Include="Avalonia.Themes.Fluent" Version="11.2.1" />
    <PackageReference Include="Avalonia.Fonts.Inter" Version="11.2.1" />
    <!--Condition below is needed to remove Avalonia.Diagnostics package from build output in Release configuration.-->
    <PackageReference Include="Avalonia.Diagnostics" Version="11.2.1">
      <IncludeAssets Condition="'$(Configuration)' != 'Debug'">None</IncludeAssets>
      <PrivateAssets Condition="'$(Configuration)' != 'Debug'">All</PrivateAssets>
    </PackageReference>
    <PackageReference Include="CommunityToolkit.Mvvm" Version="8.2.1" />
  </ItemGroup>
```

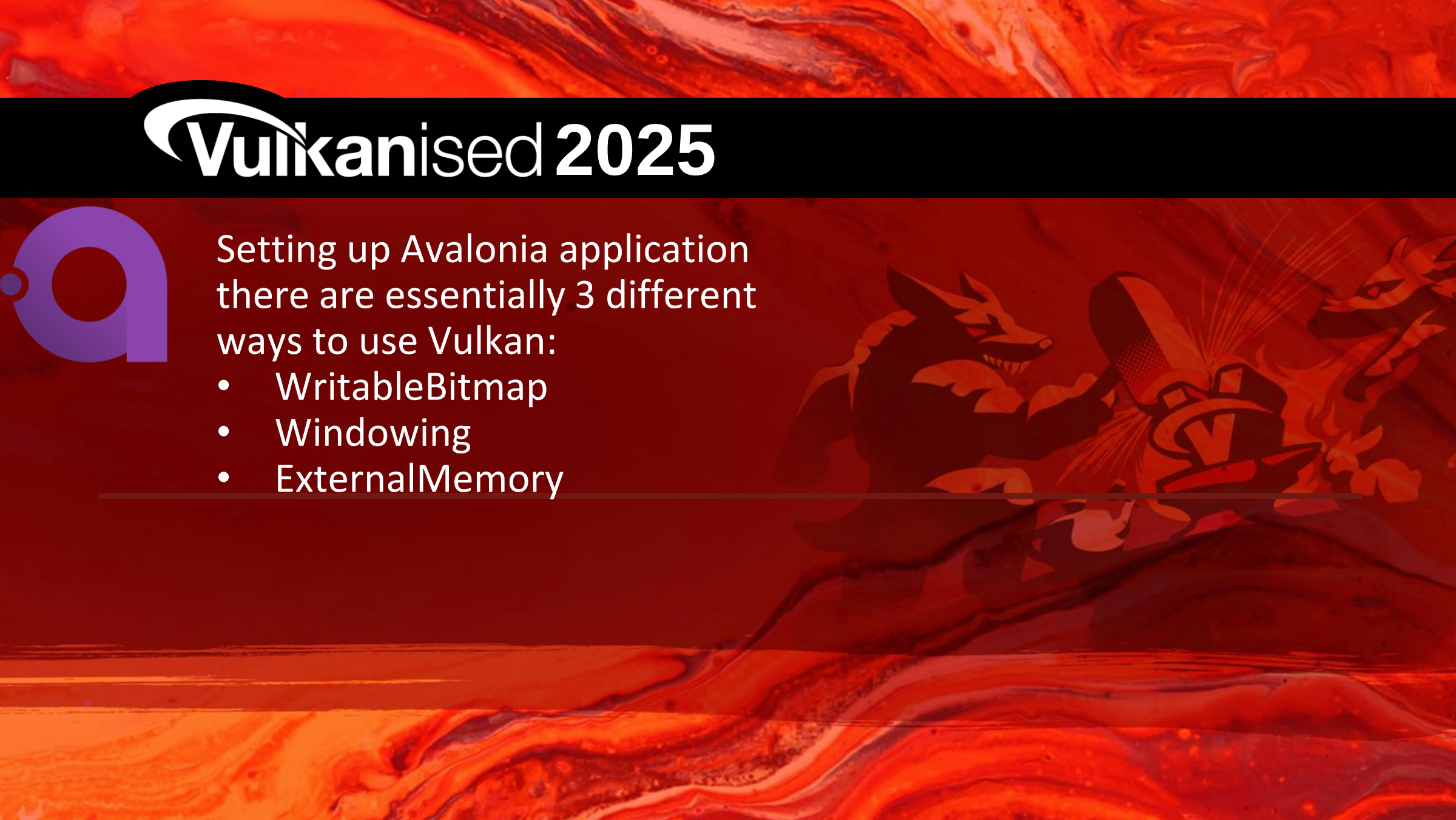
Vulkanised 2025

Required Silk.NET Packages(WritableBitmap, ExternalMemory)

- Silk.NET.Vulkan
- Silk.NET.Vulkan.Extensions.EXT
- Silk.NET.Vulkan.Extensions.KHR
- Silk.NET.Core

Optional Silk.NET Packages(Windowing)

- Silk.NET.GLFW
- Silk.NET.Windowing

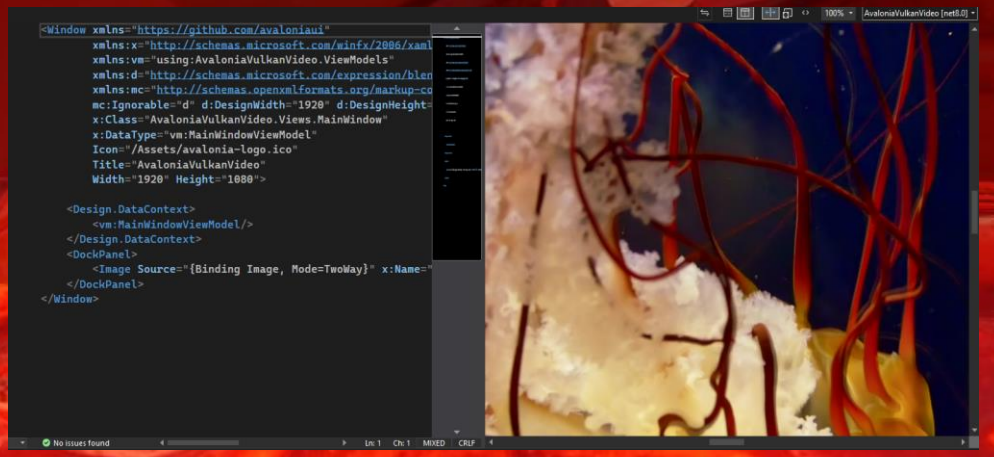


Setting up Avalonia application there are essentially 3 different ways to use Vulkan:

- WritableBitmap
- Windowing
- ExternalMemory

WritableBitmap


```
[ObservableProperty] private WriteableBitmap _image;
0 references | 0 changes | 0 authors, 0 changes
private unsafe void UpdateVideo(nint vkBufferPtr, int width, int height)
{
    using (var buffer = Image.Lock())
    {
        var size = width * height * 4;
        System.Buffer.MemoryCopy((void*)vkBufferPtr, buffer.Address.ToPointer(), size, size);
    }
    OnPropertyChanged(nameof(Image));
}
```



Windowing with GLFW

Use GLFW to manage the windowing system on the platform

```
Silk.NET.GLFW.Glfw glfwApi = Silk.NET.GLFW.Glfw.GetApi();
WindowHandle* glfwWinHandle = glfwApi.CreateWindow(1920, 1080, $"GlfwWindow", null, null);
var glfwWin = new GlfwNativeWindow(glfwApi, glfwWinHandle);
```

Parenting

- Windows
  - User32.SetParent(glfwWin.Win32.Value.Hwnd, IPlatformHandle.Handle)
- Linux
  - Xlib.XReparentWindow(display, glfwWin.X11.Value.Window, IPlatformHandle.Handle)

Avalonia.Platform.IPlatformHandle represents the native platform control handle that represents the parent of the GLFW Window

ExternalMemory

Configuring the compositor and drawing surface to support external memory that can be used for a creating VkImage

```
namespace AvaloniaVulkanVideo
{
    0 references | AndrewPBennett, 1 day ago | 1 author, 1 change
    internal sealed class Program
    {
        // Initialization code. Don't use any Avalonia, third-party APIs or any
        // SynchronizationContext-reliant code before AppMain is called: things aren't initialized
        // yet and stuff might break.
        [STAThread]
        0 references | AndrewPBennett, 1 day ago | 1 author, 1 change
        public static void Main(string[] args) => BuildAvaloniaApp()
            .StartWithClassicDesktopLifetime(args);

        1 reference | AndrewPBennett, 1 day ago | 1 author, 1 change
        public static AppBuilder BuildAvaloniaApp() =>
    AppBuilder.Configure<App>()
        .UsePlatformDetect()
        .WithInterFont()
        .With(new Win32PlatformOptions
        {
            RenderingMode = new[]
            {
                Win32RenderingMode.Vulkan
            }
        })
        .With(new X11PlatformOptions()
        {
            RenderingMode = new[]
            {
                X11RenderingMode.Vulkan
            }
        })
        .With(new VulkanOptions()
        {
            VulkanInstanceCreationOptions = new VulkanInstanceCreationOptions()
            {
                UseDebug = true
            }
        })
        .LogToTrace(LogEventLevel.Debug, "Vulkan");
    }
}
```

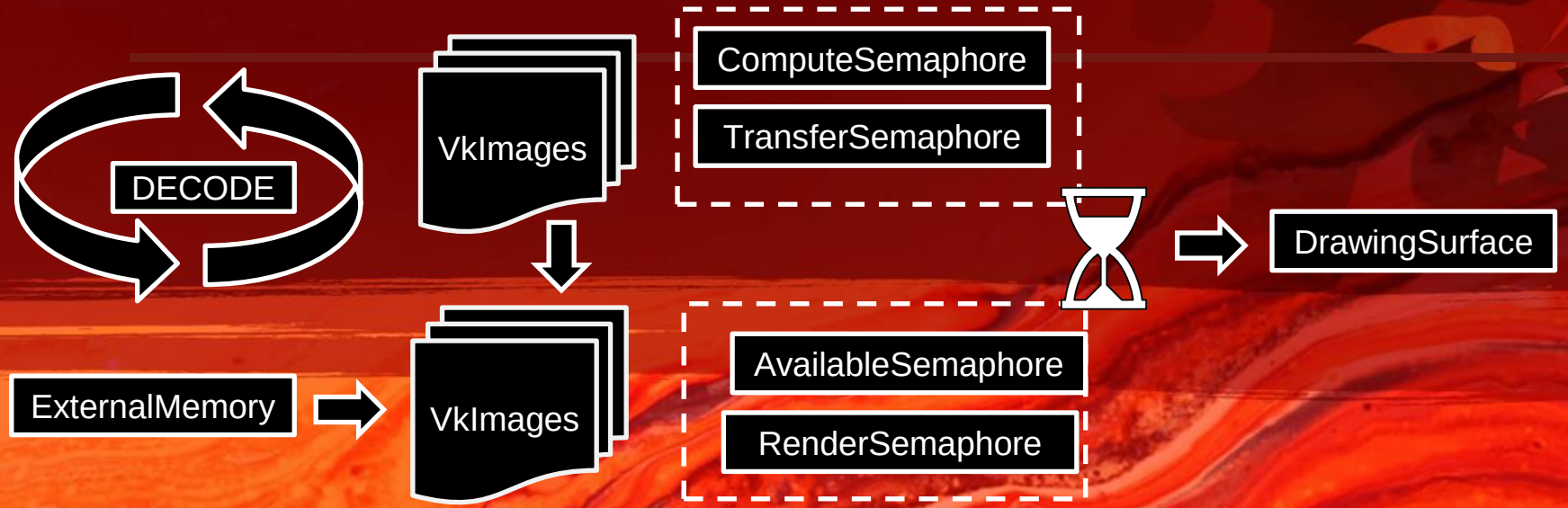
ExternalMemory

Configuring the compositor and drawing surface to support external memory that can be used for a creating VkImage

VkExternalMemoryImageCreateInfoKHR-HandleTypes

- VK_EXTERNAL_SEMAPHORE_HANDLE_TYPE_OPAQUE_FD_BIT
- VK_EXTERNAL_SEMAPHORE_HANDLE_TYPE_OPAQUE_WIN32_BIT

VkImageCreateInfo



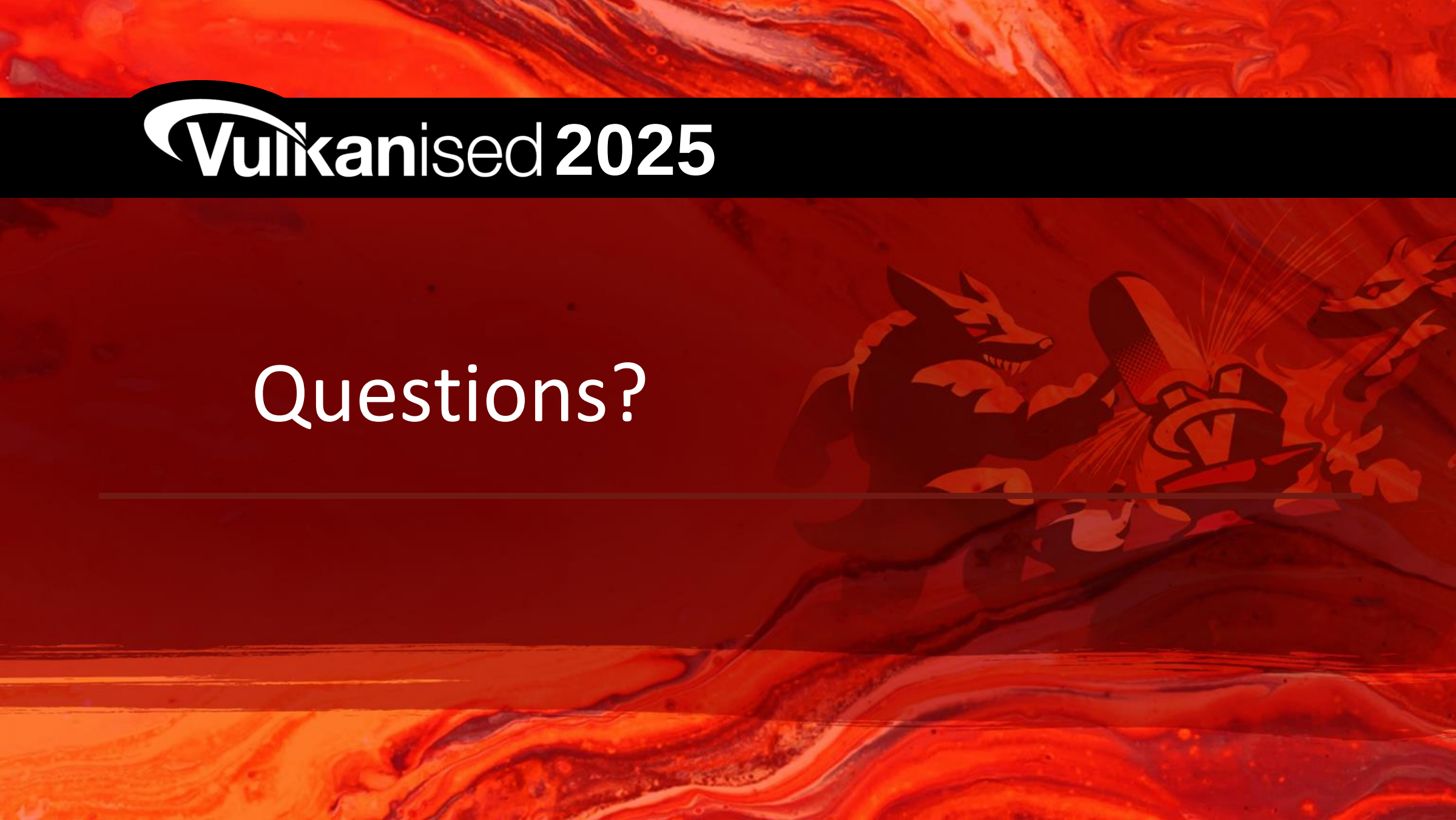
Final Result

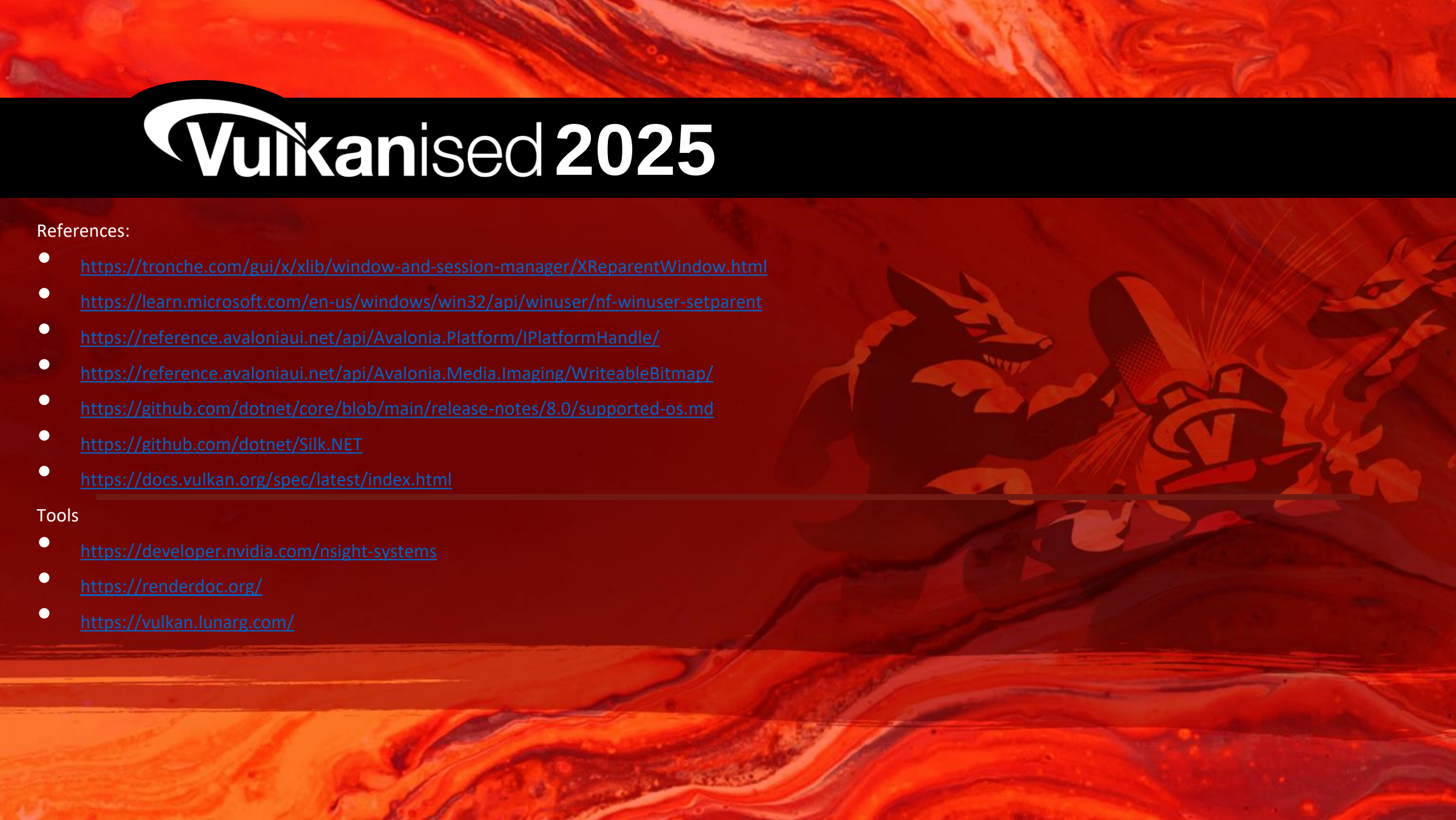
Video Files used for testing:

- https://test-videos.co.uk/

Lessons learned and challenges

- Tools for analyzing without a rendering surface are limited
  - RenderDoc and Nsight are great tools but need a graphics pipeline to use
  - Not possible to debug Vulkan Decoders or Encoders without graphics pipeline, if developing standalone software implementations
- Validation Layers helped immensely when implementing
- Decoded Picture Buffer is challenging to implement and requires a lot of different standards
- Video Capabilities vary from hardware to hardware changing the way the DPB and Decoded output is managed


Vulkanised 2025
Questions?



References:

- https://tronche.com/gui/x/xlib/window-and-session-manager/XReparentWindow.html
- https://learn.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-setparent
- https://reference.avaloniaui.net/api/Avalonia.Platform/IPlatformHandle/
- https://reference.avaloniaui.net/api/Avalonia.Media.Imaging/WriteableBitmap/
- https://github.com/dotnet/core/blob/main/release-notes/8.0/supported-os.md
- https://github.com/dotnet/Silk.NET
- https://docs.vulkan.org/spec/latest/index.html

Tools

- https://developer.nvidia.com/nsight-systems
- https://renderdoc.org/
- https://vulkan.lunarg.com/