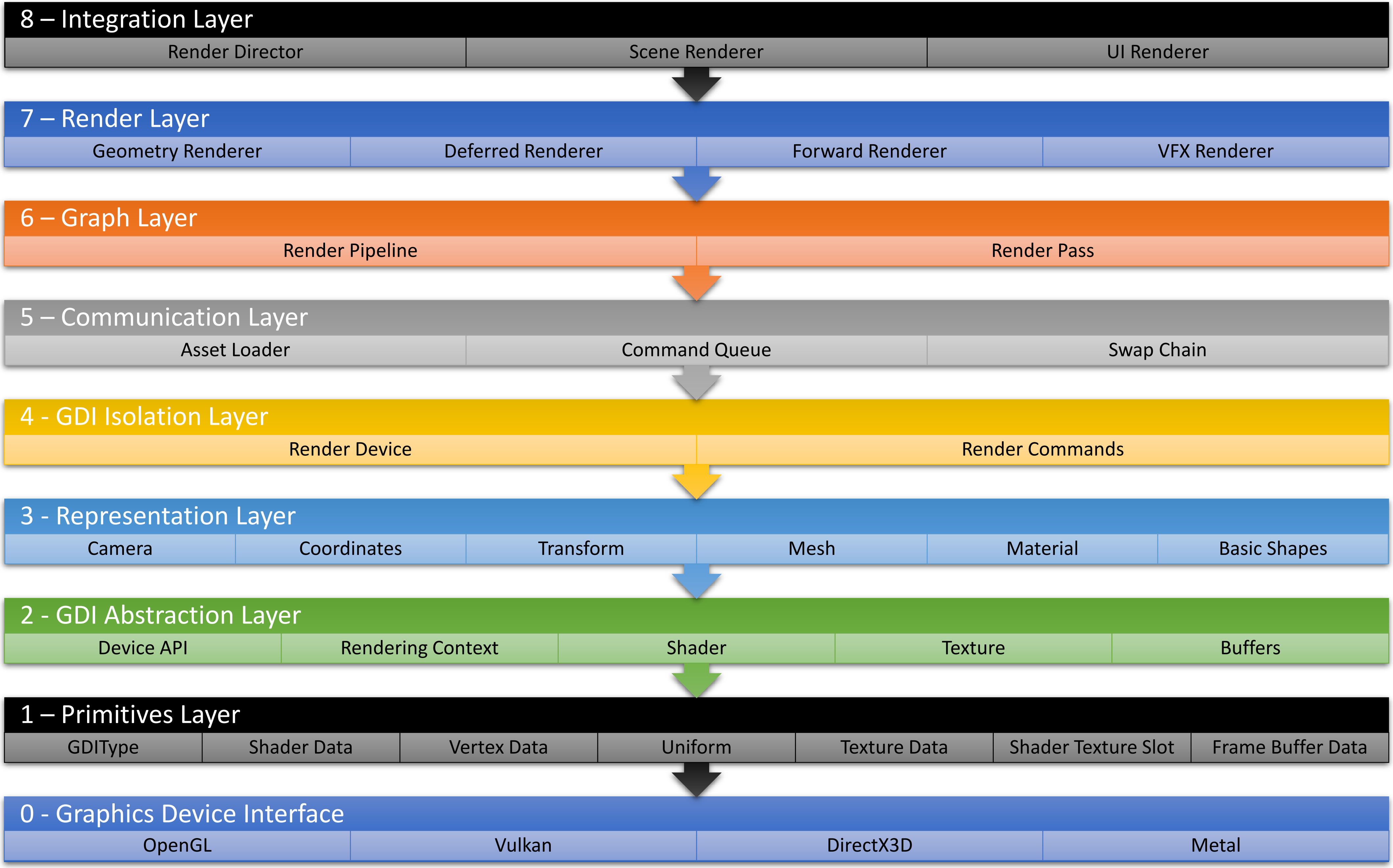
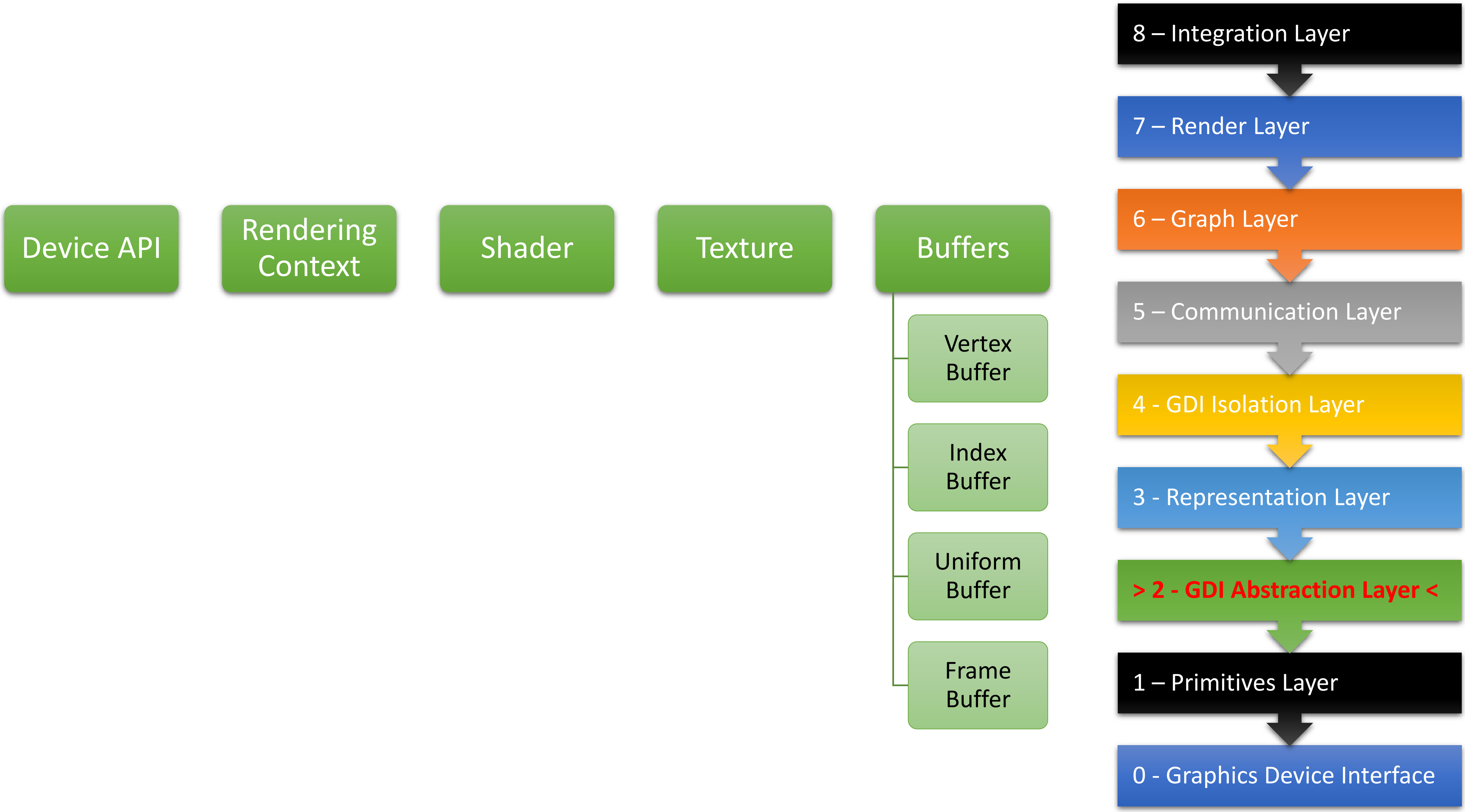
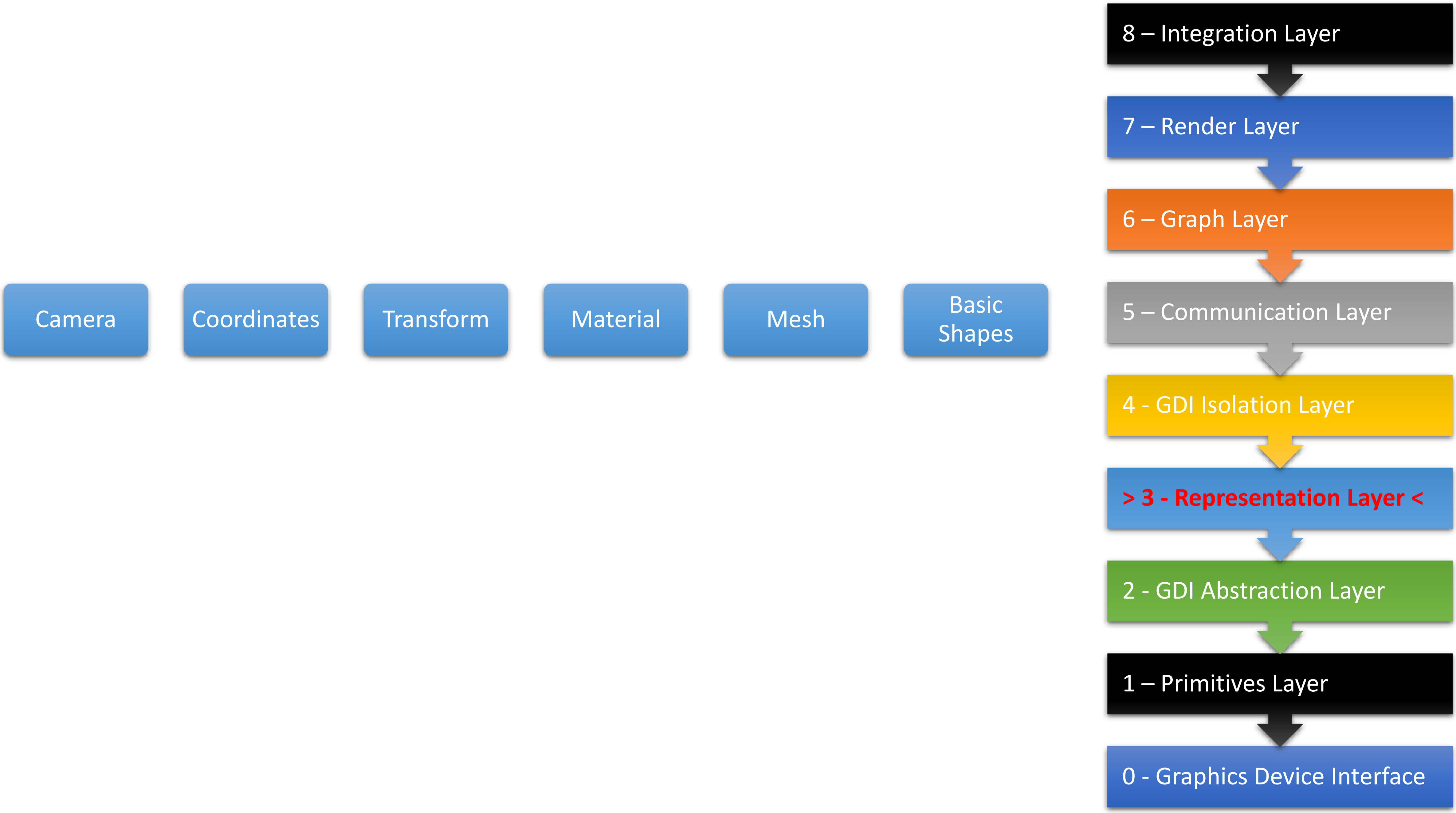
Planned Rendering Architecture of Fools Engine

Work in progress


8 – Integration Layer
Render Director
Scene Renderer
UI Renderer
7 – Render Layer
Geometry Renderer
Deferred Renderer
Forward Renderer
VFX Renderer
6 – Graph Layer
Render Pipeline
Render Pass
5 – Communication Layer
Asset Loader
Command Queue
Swap Chain
4 - GDI Isolation Layer
Render Device
Render Commands
3 - Representation Layer
Camera
Coordinates
Transform
Mesh
Material
Basic Shapes
2 - GDI Abstraction Layer
Device API
Rendering Context
Shader
Texture
Buffers
1 – Primitives Layer
GDIType
Shader Data
Vertex Data
Uniform
Texture Data
Shader Texture Slot
Frame Buffer Data
0 - Graphics Device Interface
OpenGL
Vulkan
DirectX3D
Metal


GDIType
Shader Data
Primitive
Structure
Type
Vertex Data
Element
Layout
Uniform
Texture Data
Type
Usage
Components
Format
Specification
Shader Texture Slot
Frame Buffer Data
Attachment Specification
Specification
Specification Builder
8 – Integration Layer
7 – Render Layer
6 – Graph Layer
5 – Communication Layer
4 - GDI Isolation Layer
3 - Representation Layer
2 - GDI Abstraction Layer
> 1 – Primitives Layer <
0 - Graphics Device Interface


Device API
Rendering Context
Shader
Texture
Buffers
Vertex Buffer
Index Buffer
Uniform Buffer
Frame Buffer
8 – Integration Layer
7 – Render Layer
6 – Graph Layer
5 – Communication Layer
4 - GDI Isolation Layer
3 - Representation Layer
> 2 - GDI Abstraction Layer <
1 – Primitives Layer
0 - Graphics Device Interface


Camera
Coordinates
Transform
Material
Mesh
Basic Shapes
8 – Integration Layer
7 – Render Layer
6 – Graph Layer
5 – Communication Layer
4 - GDI Isolation Layer
> 3 - Representation Layer <
2 - GDI Abstraction Layer
1 – Primitives Layer
0 - Graphics Device Interface

Render Device

Render Commands

8 – Integration Layer

7 – Render Layer

6 – Graph Layer

5 – Communication Layer

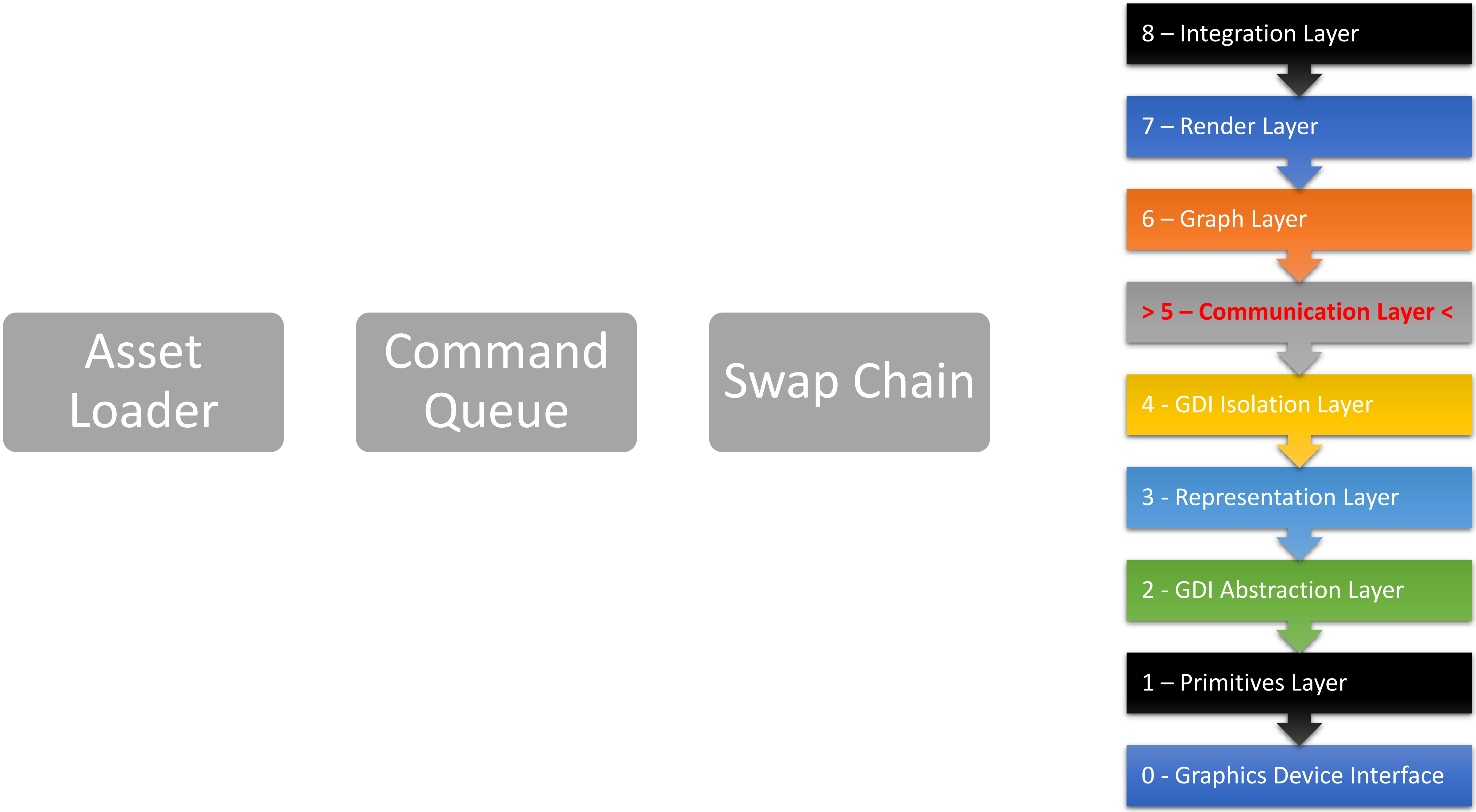
> 4 - GDI Isolation Layer <

3 - Representation Layer

2 - GDI Abstraction Layer

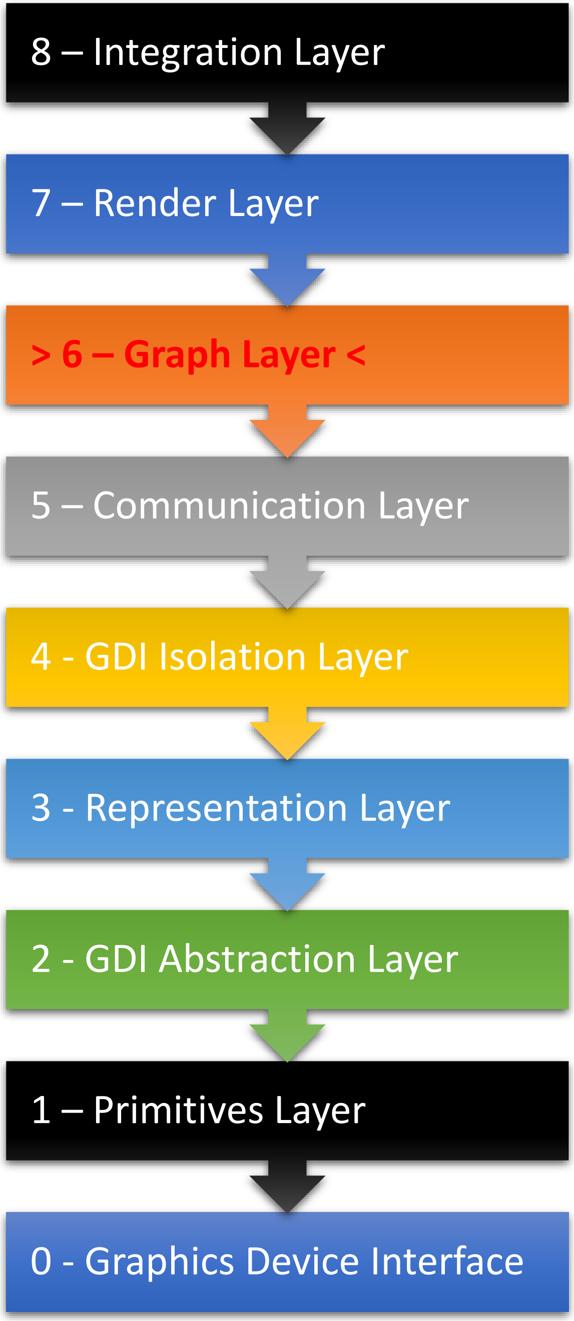
1 – Primitives Layer

0 - Graphics Device Interface


Asset Loader
Command Queue
Swap Chain
8 – Integration Layer
7 – Render Layer
6 – Graph Layer
> 5 – Communication Layer <
4 - GDI Isolation Layer
3 - Representation Layer
2 - GDI Abstraction Layer
1 – Primitives Layer
0 - Graphics Device Interface

Render Pipeline

Render Pass



Geometry Renderer

Deferred Renderer

Forward Renderer

VFX Renderer

8 – Integration Layer

> 7 – Render Layer <

6 – Graph Layer

5 – Communication Layer

4 - GDI Isolation Layer

3 - Representation Layer

2 - GDI Abstraction Layer

1 – Primitives Layer

0 - Graphics Device Interface

Render Director

Scene Renderer

UI Renderer

> 8 – Integration Layer <

7 – Render Layer

6 – Graph Layer

5 – Communication Layer

4 - GDI Isolation Layer

3 - Representation Layer

2 - GDI Abstraction Layer

1 – Primitives Layer

0 - Graphics Device Interface

Redeferred rendering – Experimental concept

Let's split diffused and specular contributions of the reflectance part of the rendering equation

$$
\begin{aligned}
&\mathrm{L}_o(\mathrm{p}, \omega_o) = \\
&= \int\limits_\Omega \left(k_d \frac{c}{\pi} + \frac{DFG}{4(\omega_o \cdot n)(\omega_i \cdot n)} \right) L_i(n \cdot \omega_i)\, d\omega_i = \\
&= \int\limits_\Omega \left(k_d \frac{c}{\pi} L_i(n \cdot \omega_i) + \frac{DFG}{4(\omega_o \cdot n)(\omega_i \cdot n)} L_i\, (n \cdot \omega_i) \right) d\omega_i = \\
&= \int\limits_\Omega \left(k_d \frac{c}{\pi} L_i(n \cdot \omega_i) + \frac{DFG}{4(\omega_o \cdot n)} L_i \right) d\omega_i = \\
&= \int\limits_\Omega k_d \frac{c}{\pi} L_i(n \cdot \omega_i)\, d\omega_i + \int\limits_\Omega \frac{DFG}{4(\omega_o \cdot n)} L_i\, d\omega_i
\end{aligned}
$$

$$\mathrm{L}_d(\mathrm{p}, \omega_o) = \int\limits_\Omega k_d \frac{c}{\pi} L_i(n \cdot \omega_i)\, d\omega_i$$

$$\mathrm{L}_s(\mathrm{p}, \omega_o) = \int\limits_\Omega \frac{DFG}{4(\omega_o \cdot n)} L_i\, d\omega_i$$

$$\mathrm{L}_o(\mathrm{p}, \omega_o) = L_d(\mathrm{p}, \omega_o) + L_s(\mathrm{p}, \omega_o)$$

Diffused

$$\mathrm{L}_d(\mathrm{p},\omega_o) = \int_\Omega k_d \frac{c}{\pi} L_i(n\cdot\omega_i)\, d\omega_i = \frac{c}{\pi}\int_\Omega k_d L_i(n\cdot\omega_i)\, d\omega_i =$$

$$= \frac{c}{\pi}\int_\Omega (1-k_s)(1-m)L_i(n\cdot\omega_i)\, d\omega_i =$$

$$= (1-m)\frac{c}{\pi}\int_\Omega (1-k_s)L_i(n\cdot\omega_i)\, d\omega_i \approx$$

$$\approx (1-m)\frac{c}{\pi}\int_\Omega \big(1-(F_0+(1-F_0)P_i)\big)L_i(n\cdot\omega_i)\, d\omega_i =$$

$$= (1-m)\frac{c}{\pi}\int_\Omega (1-F_0-(1-F_0)P_i)L_i(n\cdot\omega_i)\, d\omega_i =$$

$$= (1-m)\frac{c}{\pi}\int_\Omega \big((1-F_0)-(1-F_0)P_i\big)L_i(n\cdot\omega_i)\, d\omega_i =$$

$$= (1-m)\frac{c}{\pi}\int_\Omega \big((1-F_0)(1-P_i)\big)L_i(n\cdot\omega_i)\, d\omega_i =$$

$$= (1-m)\frac{c}{\pi}(1-F_0)\int_\Omega (1-P_i)L_i(n\cdot\omega_i)\, d\omega_i =$$

$$= (1-m)\frac{c}{\pi}(1-F_0)\int_\Omega 1-P_i\, d\omega_i \int_\Omega L_i(n\cdot\omega_i)\, d\omega_i =$$

$$= (1-m)\frac{c}{\pi}(1-F_0)\left(1-\int_\Omega P_i\, d\omega_i\right)\int_\Omega L_i(n\cdot\omega_i)\, d\omega_i$$

$$\mathrm{L}_{d_g} = \int_\Omega L_i(n\cdot\omega_i)\, d\omega_i$$

$$\mathrm{P} = \int_\Omega P_i\, d\omega_i$$

$$\mathrm{L}_d(\mathrm{p},\omega_o) \approx (1-m)\frac{c}{\pi}(1-F_0)(1-\mathrm{P})\mathrm{L}_{d_g}$$

Specular

$$\mathrm{L}_s(\mathrm{p},\omega_o) = \int_\Omega \frac{DFG}{4(\omega_o\cdot n)} L_i\, d\omega_i = \frac{1}{4(\omega_o\cdot n)}\int_\Omega DGL_iF\, d\omega_i \approx$$

$$\approx \frac{1}{4(\omega_o\cdot n)}\int_\Omega DGL_i(F_0+(1-F_0)P_i)\, d\omega_i =$$

$$= \frac{1}{4(\omega_o\cdot n)}\int_\Omega DGL_iF_0 + DGL_iP_i(1-F_0)\, d\omega_i =$$

$$= \frac{1}{4(\omega_o\cdot n)}\left(\int_\Omega DGL_iF_0\, d\omega_i + \int_\Omega DGL_iP_i(1-F_0)\, d\omega_i\right) =$$

$$= \frac{1}{4(\omega_o\cdot n)}\left(F_0\int_\Omega DGL_i\, d\omega_i + (1-F_0)\int_\Omega DGL_iP_i\, d\omega_i\right) =$$

$$= \frac{1}{4(\omega_o\cdot n)}\left(F_0\int_\Omega DGL_i\, d\omega_i + (1-F_0)\int_\Omega DGL_i\, d\omega_i\int_\Omega P_i\, d\omega_i\right) =$$

$$= \frac{1}{4(\omega_o\cdot n)}\int_\Omega DGL_i\, d\omega_i\left(F_0 + (1-F_0)\int_\Omega P_i\, d\omega_i\right)$$

$$\mathrm{L}_{s_g} = \int_\Omega DGL_i\, d\omega_i$$

$$\mathrm{P} = \int_\Omega P_i\, d\omega_i$$

$$\mathrm{L}_s(\mathrm{p},\omega_o) \approx \frac{1}{4(\omega_o\cdot n)}(F_0+(1-F_0)\mathrm{P})\mathrm{L}_{s_g}$$

Fresnel-Schlick approximation:

$$k_s = F = F_0 + (1-F_0)\big(1-(h_i\cdot v)\big)^5$$

$$P_i = \big(1-(h_i\cdot v)\big)^5$$

$$k_s = F = F_0 + (1-F_0)P_i$$

Recombination

$$\mathrm{L}_{d_g} = \int\limits_{\Omega} L_i (n \cdot \omega_i)\, d\omega_i$$

$$\mathrm{L}_{s_g} = \int\limits_{\Omega} DGL_i\, d\omega_i$$

$$P_i = \left(1 - (h_i \cdot v)\right)^5$$

$$\mathrm{P} = \int_{\Omega} P_i\, d\omega_i = f(\vartheta) \quad \vartheta = (v \cdot n)$$

An aquasion approximationg $f(\vartheta)$ can be developed by matching precomputed results of the integral for various angles ϑ

$$\mathrm{L}_d(\mathrm{p}, \omega_o) \approx (1-m)\frac{c}{\pi}(1-F_0)(1-\mathrm{P})\mathrm{L}_{d_g}$$

$$\mathrm{L}_s(\mathrm{p}, \omega_o) \approx \frac{1}{4(\omega_o \cdot n)}(F_0 + (1-F_0)\mathrm{P})\,\mathrm{L}_{s_g}$$

$$\mathrm{L}_o(\mathrm{p}, \omega_o) = L_d(\mathrm{p}, \omega_o) + L_s(\mathrm{p}, \omega_o)$$

We are going to calculate L_{d_g} and L_{s_g} first looping through all illumination sources (light / ray).
We are going to calculate P (as approximation of $f(\vartheta)$), L_d, L_s and finally L_o outside the loop.

A portion of calculations is now pushed out of the loop and performed only once, instead of on a per illumination source basis.

Metalness and base color are now not accessed inside the loop at all.

Depending on chosen approximations of normal distribution function and geometry function additional elements could be pulled out of the loop (e.g. $\frac{\alpha^2}{\pi}$ can be pulled out of Trowbridge-Reitz GGX approximation of NDF) although not fully.

Redeffered rendering

$$\mathrm{L}_{d_g} = \int_\Omega L_i(n \cdot \omega_i)\, d\omega_i$$

$$\mathrm{L}_{s_g} = \int_\Omega DGL_i\, d\omega_i$$

$$P_i = \left(1 - (h_i \cdot v)\right)^5$$

$$\mathrm{P} = \int_\Omega P_i\, d\omega_i = f(\vartheta) \qquad \vartheta = (v \cdot n)$$

$$\mathrm{L}_d(\mathrm{p}, \omega_o) \approx (1 - m)\frac{c}{\pi}(1 - F_0)(1 - \mathrm{P})\mathrm{L}_{d_g}$$

$$\mathrm{L}_s(\mathrm{p}, \omega_o) \approx \frac{1}{4(\omega_o \cdot n)}(F_0 + (1 - F_0)\mathrm{P})\mathrm{L}_{s_g}$$

$$\mathrm{L}_o(\mathrm{p}, \omega_o) = L_d(\mathrm{p}, \omega_o) + L_s(\mathrm{p}, \omega_o)$$

Lets split shading into two stages.

First stage will produce L_{d_g} and L_{s_g} values and store them in a dedicated buffer – I-buffer.
Second stage will use those values to calculate L_o.

I-buffer retains fundamental characteristic of light to be cumulative. This allows first step to be split into multiple passes using specialized shaders for different types of lights, shadows. Redeferred rendering can be used analogously with most of indirect illumination techniques too and their results can be accumulated to the same I-buffer with a common second stage to be applied.

The first stage depends only on lights and geometry (from the PBR point of view roughness is a simplified representation of subpixel geometry, not a property of a substance). The variability of the output buffer is expected to be low (imagine a white world with colorful lights). That makes this stage a good candidate for shading rate reducing technique (lower overall resolution of the buffer / VRS / VRCS / DACS / spatial upscaling / temporal upscaling etc.). We can safely apply a reduced shading rate everywhere unless there is an illumination discontinuity - hard shadow coming from an illumination source considered by a particular shader or geometry edge – this will require tuning based on visual quality.

The second stage does not have to iterate over illumination sources, so it will be used to refine results by integrating them with materials at full resolution. It will fully cover artifacts of reduced shading rate from the first stage, making them completely imperceptible. Suppose roughness is pulled out of the Trowbridge-Reitz GGX approximation of NDF. In that case, we have all geometry describing parameters of a given pixel from the G-buffer considered during the refining second stage, providing us with differentiation of pixels shaded together in the first pass.

Redeffered rendering - considerations

In case of memory throughput becoming a significant bottleneck:

A. Precalculating $(\omega_o \cdot n)$ in geometry pass – no need for fetching normal and depth in the second stage
B. Smart packing of G-Buffer to gather values used in the first stage

In case of light bleeding over edges and aliased shadow edges due to greater aliasing in I-Buffer than in G-Buffer:

A. Using the same resolution of I-Buffer as of G-Buffer
B. In case of VRCS, using dedicated shading rate attachments per illumination aspect (direct, shadows, reflections, indirect diffuse, etc.)
C. More sophisticated upscaling of I-Buffer:
    A. considering separately diffuse and specular
    B. Using G-Buffer

Separate diffuse and specular aspects of illumination can be saved across frames and used with temporal techniques separately, to improve quality:

A. Different types of reprojection: hit point and motion vector ([Tomasz Stachowiak: Stochastic all the things: Raytracing in hybrid real-time rendering]())
B. Ghosting additionally reduced by second shading stage
C. Using temporal techniques before second shading stage brings back texture details

$$\mathrm{L}_{d_g} = \int_\Omega L_i (n \cdot \omega_i)\, d\omega_i$$

$$\mathrm{L}_{s_g} = \int_\Omega DGL_i \, d\omega_i$$

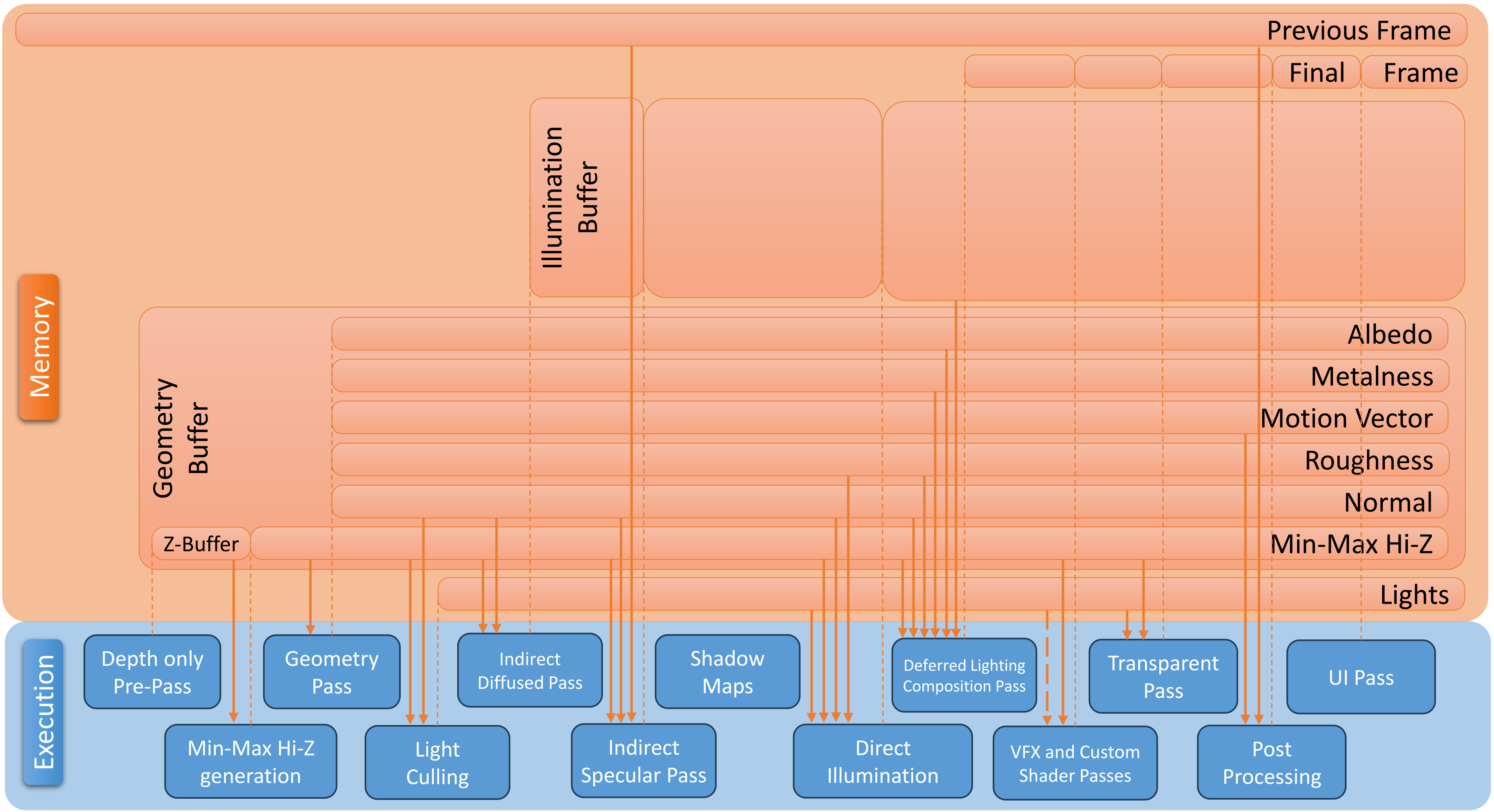
$$P_i = \left(1 - (h_i \cdot v)\right)^5$$

$$\mathrm{P} = \int_\Omega P_i \, d\omega_i = f(\vartheta) \qquad \vartheta = (\omega_o \cdot n)$$

$$\mathrm{L}_d(\mathrm{p}, \omega_o) \approx (1 - m)\frac{c}{\pi}(1 - F_0)(1 - \mathrm{P})\mathrm{L}_{d_g}$$

$$\mathrm{L}_s(\mathrm{p}, \omega_o) \approx \frac{\mathrm{L}_{s_g}}{4(\omega_o \cdot n)}(F_0 + (1 - F_0)\mathrm{P})$$

$$\mathrm{L}_o(\mathrm{p}, \omega_o) = L_d(\mathrm{p}, \omega_o) + L_s(\mathrm{p}, \omega_o)$$



Memory
Previous Frame
Final
Frame
Illumination Buffer
Geometry Buffer
Albedo
Metalness
Motion Vector
Roughness
Normal
Z-Buffer
Min-Max Hi-Z
Lights
Execution
Depth only Pre-Pass
Geometry Pass
Indirect Diffused Pass
Shadow Maps
Deferred Lighting Composition Pass
Transparent Pass
UI Pass
Min-Max Hi-Z generation
Light Culling
Indirect Specular Pass
Direct Illumination
VFX and Custom Shader Passes
Post Processing