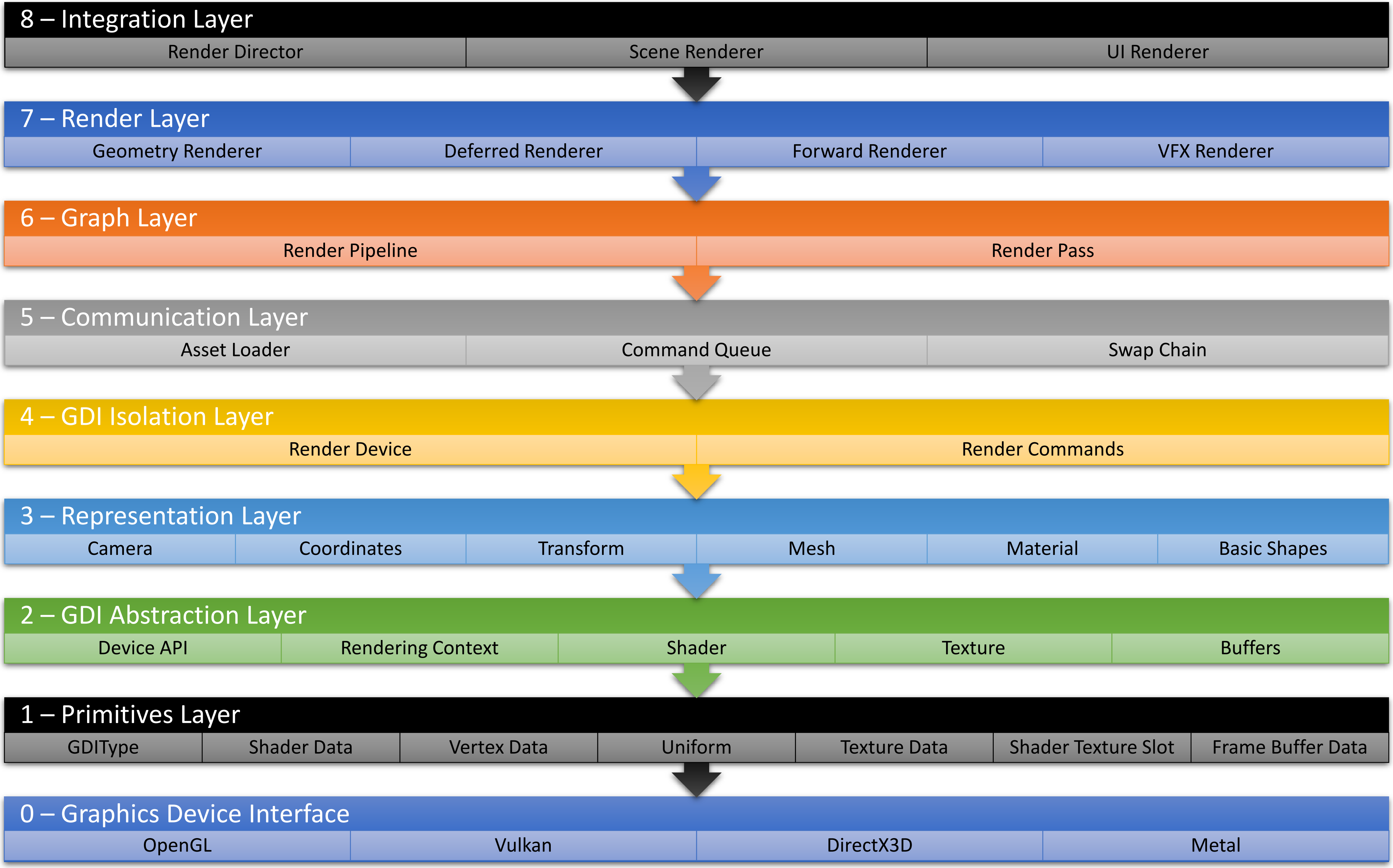
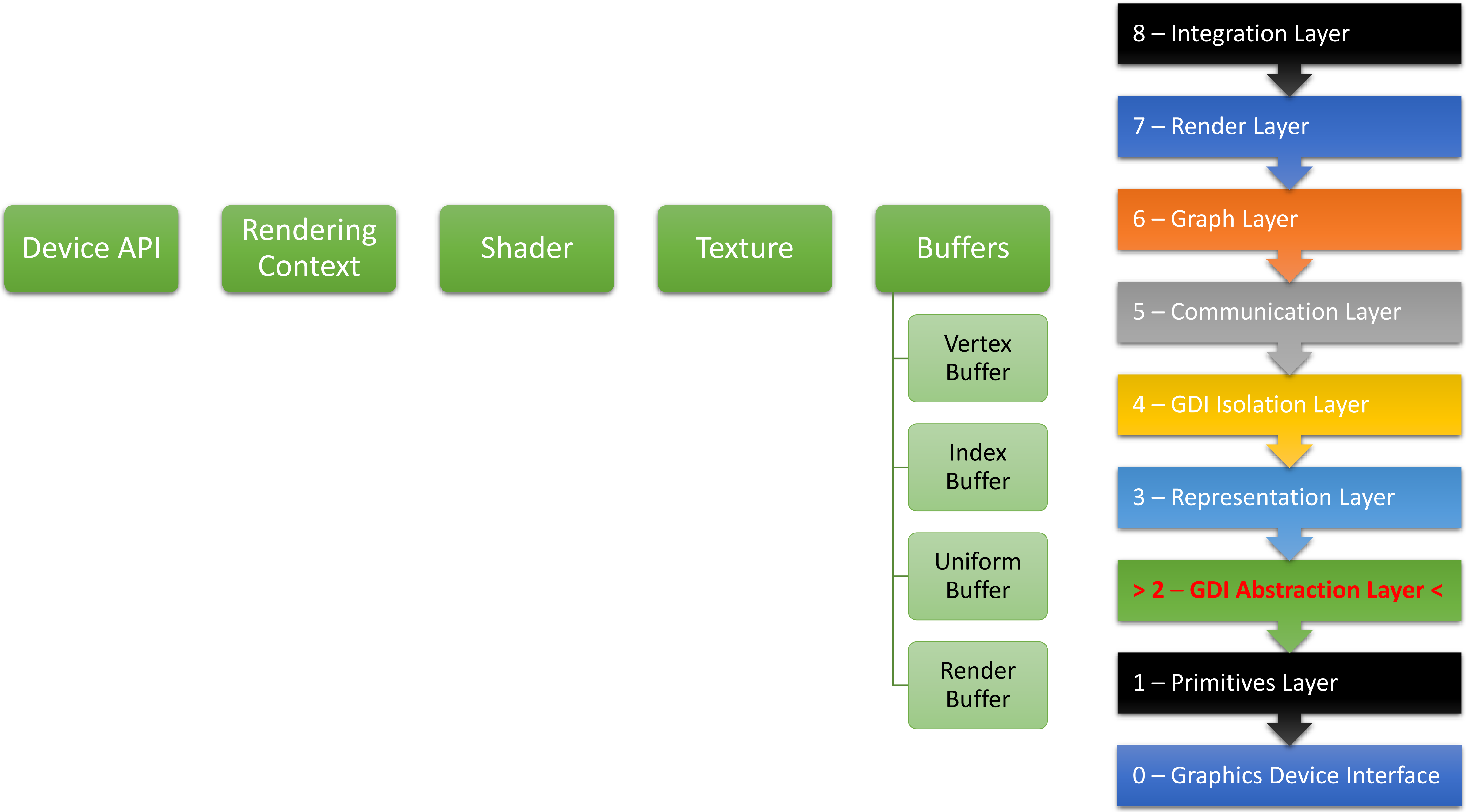
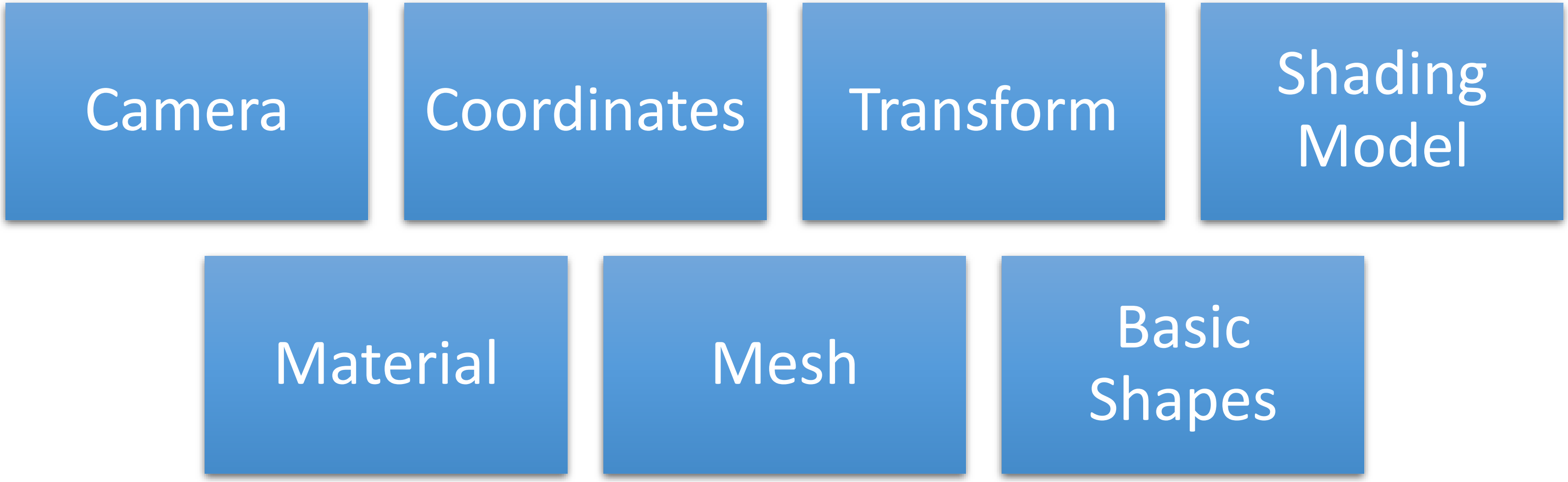
Planned Rendering Architecture of Fools Engine

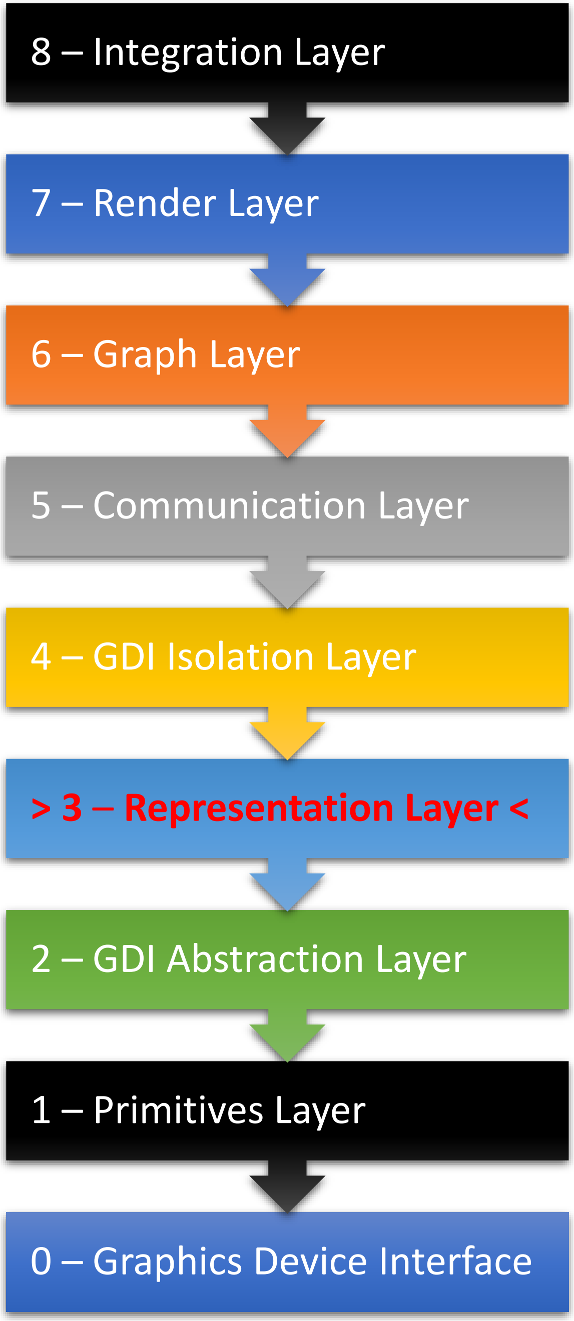
Work in progress


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
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

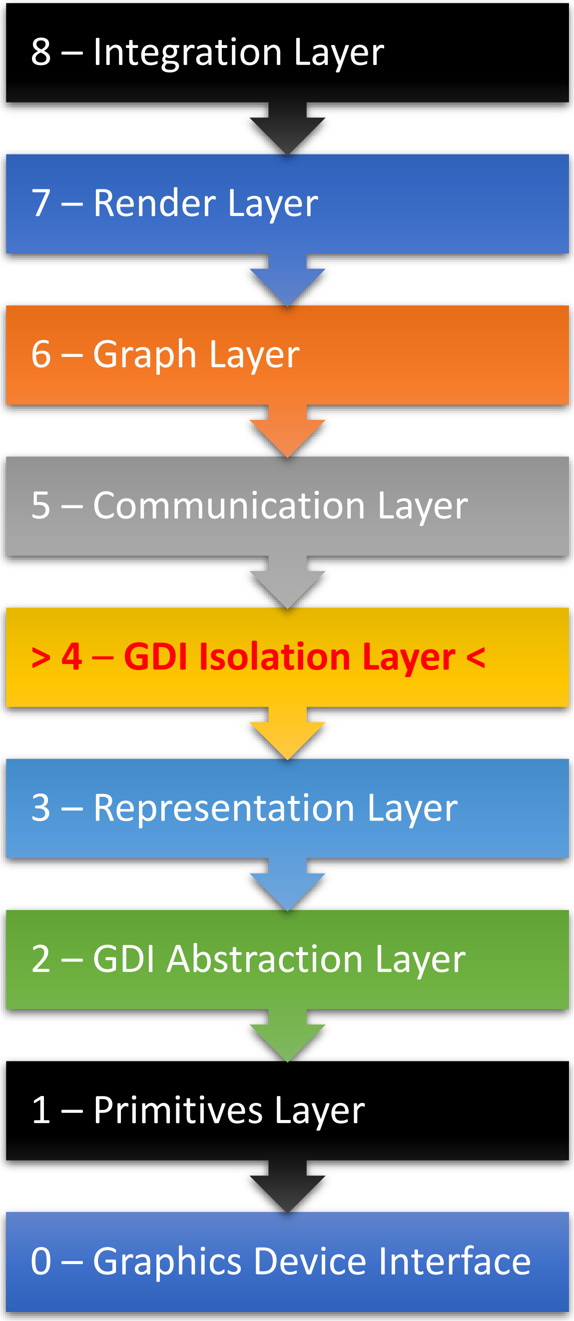

Device API
Rendering Context
Shader
Texture
Buffers
Vertex Buffer
Index Buffer
Uniform Buffer
Render Buffer
8 – Integration Layer
7 – Render Layer
6 – Graph Layer
5 – Communication Layer
4 – GDI Isolation Layer
3 – Representation Layer
> 2 – GDI Abstraction Layer <
1 – Primitives Layer
0 – Graphics Device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

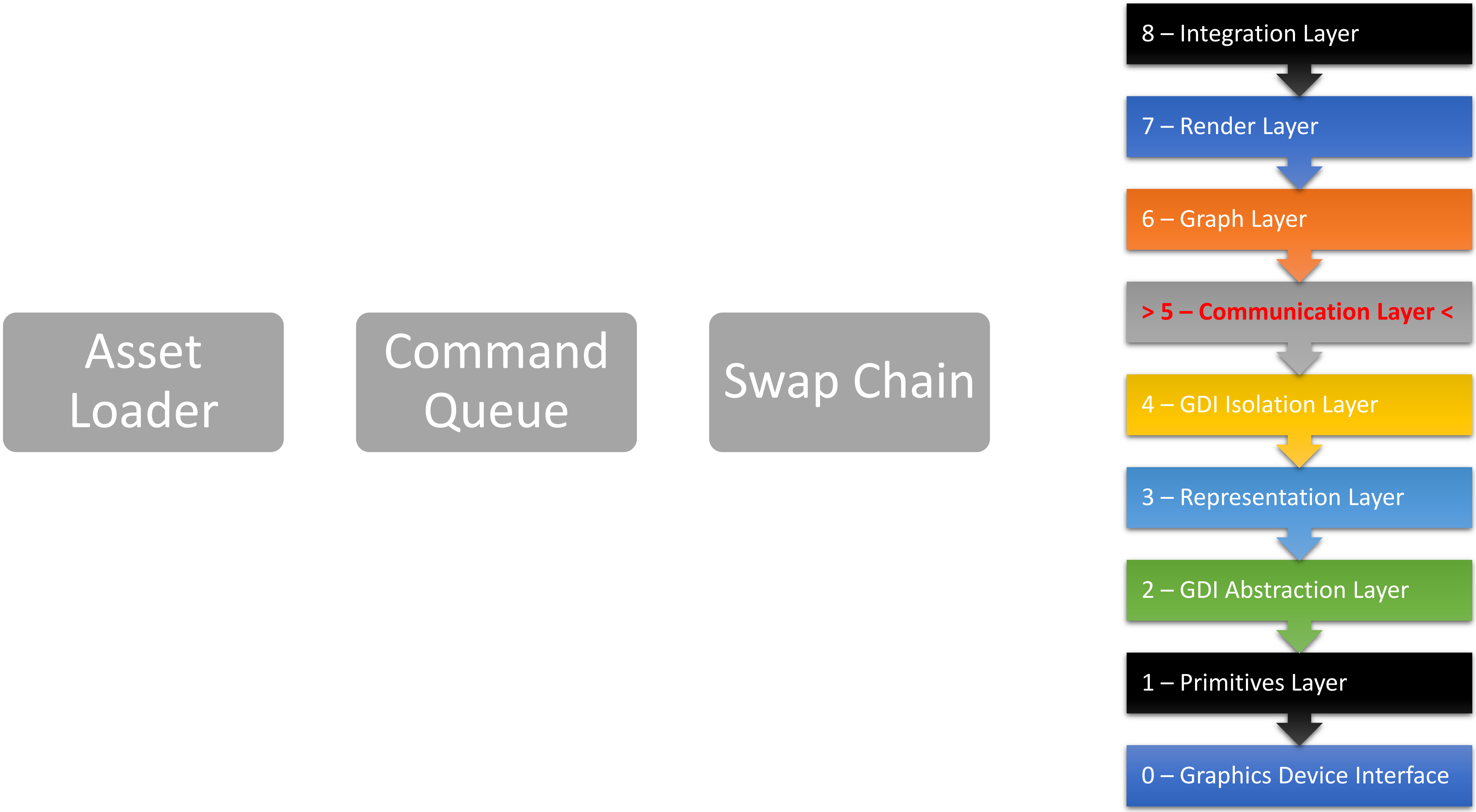

Camera
Coordinates
Transform
Shading Model
Material
Mesh
Basic Shapes


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




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

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

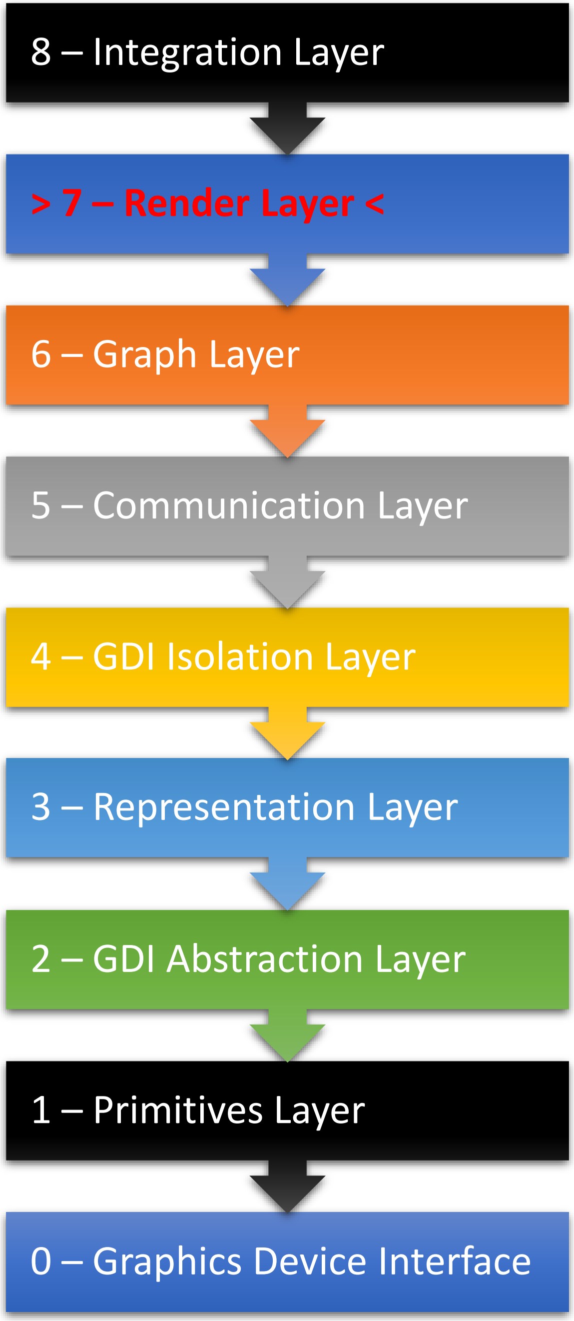
0 – Graphics Device Interface

Geometry Renderer

Deferred Renderer

Forward Renderer

VFX Renderer



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

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

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

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

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

Diffused

Fresnel-Schlick approximation:

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

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

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

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

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

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

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

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

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

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

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

$$\mathrm{L}_{d_g} = \int_\Omega (1 - P_i) L_i (n \cdot \omega_i)\, d\omega_i$$

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

Specular

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

$$\approx \frac{1}{4(\omega_o\cdot n)}\int_\Omega DF\frac{(\omega_o\cdot n)}{(\omega_o\cdot n)(1-k)+k}\times\frac{(\omega_i\cdot n)}{(\omega_i\cdot n)(1-k)+k}L_i\, d\omega_i =$$

$$= \frac{1}{4(\omega_o\cdot n)}\times\frac{(\omega_o\cdot n)}{(\omega_o\cdot n)(1-k)+k}\int_\Omega DF\frac{(\omega_i\cdot n)}{(\omega_i\cdot n)(1-k)+k}L_i\, d\omega_i =$$

$$= \frac{1}{4\big((\omega_o\cdot n)(1-k)+k\big)}\int_\Omega DF\frac{(\omega_i\cdot n)}{(\omega_i\cdot n)(1-k)+k}L_i\, d\omega_i \approx$$

$$\approx \frac{1}{4\big((\omega_o\cdot n)(1-k)+k\big)}\int_\Omega \frac{\alpha^2}{\pi\big((n\cdot h_i)^2(\alpha^2-1)+1\big)^2}F\frac{(\omega_i\cdot n)}{(\omega_i\cdot n)(1-k)+k}L_i\, d\omega_i =$$

$$= \frac{\alpha^2}{4\pi\big((\omega_o\cdot n)(1-k)+k\big)}\int_\Omega \frac{1}{\big((n\cdot h_i)^2(\alpha^2-1)+1\big)^2}F\frac{(\omega_i\cdot n)}{(\omega_i\cdot n)(1-k)+k}L_i\, d\omega_i =$$

$$= \frac{\alpha^2}{4\pi\big((\omega_o\cdot n)(1-k)+k\big)}\int_\Omega F\frac{L_i(\omega_i\cdot n)}{\big((n\cdot h_i)^2(\alpha^2-1)+1\big)^2\big((\omega_i\cdot n)(1-k)+k\big)}\, d\omega_i =$$

$$= \frac{\alpha^2}{4\pi\big((\omega_o\cdot n)(1-k)+k\big)}\int_\Omega F\frac{L_i(\omega_i\cdot n)}{R_i}\, d\omega_i \approx$$

$$\approx \frac{\alpha^2}{4\pi\big((\omega_o\cdot n)(1-k)+k\big)}\int_\Omega (F_0+(1-F_0)P_i)\frac{L_i(\omega_i\cdot n)}{R_i}\, d\omega_i =$$

$$= \frac{\alpha^2}{4\pi\big((\omega_o\cdot n)(1-k)+k\big)}\int_\Omega F_0\frac{L_i(\omega_i\cdot n)}{R_i}+(1-F_0)P_i\frac{L_i(\omega_i\cdot n)}{R_i}\, d\omega_i =$$

$$= \frac{\alpha^2}{4\pi\big((\omega_o\cdot n)(1-k)+k\big)}\left(\int_\Omega F_0\frac{L_i(\omega_i\cdot n)}{R_i}\, d\omega_i+\int_\Omega (1-F_0)P_i\frac{L_i(\omega_i\cdot n)}{R_i}\, d\omega_i\right)=$$

$$= \frac{\alpha^2}{4\pi\big((\omega_o\cdot n)(1-k)+k\big)}\left(F_0\int_\Omega \frac{L_i(\omega_i\cdot n)}{R_i}\, d\omega_i+(1-F_0)\int_\Omega P_i\frac{L_i(\omega_i\cdot n)}{R_i}\, d\omega_i\right)$$

Fresnel-Schlick approximation:

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

Smith's method with Schlick-GGX:

$$G = \frac{(\omega_o\cdot n)}{(\omega_o\cdot n)(1-k)+k}\times\frac{(\omega_i\cdot n)}{(\omega_i\cdot n)(1-k)+k}$$
$k = \frac{(\alpha+1)^2}{8}$ or $k = \frac{\alpha^2}{2}$

Trowbridge-Reitz GGX:

$$D = \frac{\alpha^2}{\pi\big((n\cdot h_i)^2(\alpha^2-1)+1\big)^2}$$

$$R_i = \big((n\cdot h_i)^2(\alpha^2-1)+1\big)^2\big((\omega_i\cdot n)(1-k)+k\big)$$

$$L_{s_{g_1}} = \int_\Omega \frac{L_i(\omega_i\cdot n)}{R_i}\, d\omega_i$$

$$L_{s_{g_2}} = \int_\Omega \frac{P_iL_i(\omega_i\cdot n)}{R_i}\, d\omega_i$$

$$\mathrm{L}_s(\mathrm{p},\omega_o) = \frac{\alpha^2}{4\pi\big((\omega_o\cdot n)(1-k)+k\big)}\Big(F_0L_{s_{g_1}}+(1-F_0)L_{s_{g_2}}\Big)$$

Recombination

$$k = \frac{(\alpha+1)^2}{8} \text{ or } k = \frac{\alpha^2}{2}$$
$$R_i = \left((n \cdot h_i)^2(\alpha^2 - 1) + 1\right)^2\left((\omega_i \cdot n)(1-k) + k\right)$$
$$P_i = \left(1 - (h_i \cdot \omega_o)\right)^5$$

$$\mathrm{L}_{d_g} = \int_\Omega (1 - P_i) L_i (n \cdot \omega_i)\, d\omega_i$$
$$L_{s_{g_1}} = \int_\Omega \frac{L_i(\omega_i \cdot n)}{R_i}\, d\omega_i$$
$$L_{s_{g_2}} = \int_\Omega \frac{P_i L_i(\omega_i \cdot n)}{R_i} d\omega_i$$

$$\mathrm{L}_d(\mathrm{p}, \omega_o) = (1 - m)\frac{c}{\pi}(1 - F_0)\mathrm{L}_{d_g}$$
$$\mathrm{L}_s(\mathrm{p}, \omega_o) = \frac{\alpha^2}{4\pi\left((\omega_o \cdot n)(1-k) + k\right)}\left(F_0 L_{s_{g_1}} + (1 - F_0) L_{s_{g_2}}\right)$$

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

We are going to calculate L_{d_g}, $\mathrm{L}_{s_{g_1}}$ and $\mathrm{L}_{s_{g_2}}$ first, looping through all illumination sources (lights / rays).

Then we are going to calculate 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.

Redeffered rendering

Lets split shading into two stages.

First stage (Illumination pass) will produce L_{d_g}, $\mathrm{L}_{s_{g_1}}$ and $\mathrm{L}_{s_{g_2}}$ values and store them in a dedicated buffer, let's call it I-buffer.
Second stage (Material pass) will use those values to calculate L_o.

I-buffer retains fundamental characteristic of light to be cumulative. This allows first stage 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 microscopic geometry). The variability of the I-buffer is expected to be low (imagine a white world with colorful lights). That makes this stage a good candidate for shading rate reducing techniques. 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.

$$k = \frac{(\alpha+1)^2}{8} \text{ or } k = \frac{\alpha^2}{2}$$

$$R_i = \left((n\cdot h_i)^2(\alpha^2-1)+1\right)^2\left((\omega_i\cdot n)(1-k)+k\right)$$

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

$$\mathrm{L}_{d_g} = \int_\Omega (1-P_i)L_i(n\cdot\omega_i)\,d\omega_i$$

$$L_{s_{g_1}} = \int_\Omega \frac{L_i(\omega_i\cdot n)}{R_i}\,d\omega_i$$

$$L_{s_{g_2}} = \int_\Omega \frac{P_iL_i(\omega_i\cdot n)}{R_i}\,d\omega_i$$

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

$$\mathrm{L}_s(\mathrm{p},\omega_o) = \frac{\alpha^2}{4\pi\left((\omega_o\cdot n)(1-k)+k\right)}\left(F_0L_{s_{g_1}}+(1-F_0)L_{s_{g_2}}\right)$$

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

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. We still consider geometry during the refining second stage, providing us with differentiation of pixels shaded together in the first pass. This fully covers artifacts of the reduced shading rate from the first stage, making them completely imperceptible.

Redeffered rendering - considerations

Bottleneck of memory throughput:

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 separate values used only in the second stage from those used in both stages

Light bleeding over geometry edges, aliased shadow edges and aliasing in strongly illuminated pixels due to greater aliasing in I-Buffer than in G-Buffer:

A. VRCS with dedicated shading rate attachments per illumination source (direct, shadows, reflections, indirect diffuse, etc.) and shading into an intermediary render target (to have a base for producing attachment texture)
B. Separate, specialized upscaling for diffuse and specular
C. Using G-Buffer as additional information for upscaling (edge preservation)
D. Temporal upscaling

Separate diffuse and specular aspects of illumination when used with temporal techniques separately improve quality (e.g. different types of reprojection: hit point and motion vector - [Tomasz Stachowiak: Stochastic all the things: Raytracing in hybrid real-time rendering])

Final frame still needs antialiasing

Other shading models may not be compatible with this technique!

Complete incompatibility with translucency.

$$k = \frac{(\alpha+1)^2}{8} \text{ or } k = \frac{\alpha^2}{2}$$

$$R_i = \left((n \cdot h_i)^2(\alpha^2 - 1) + 1\right)^2\left((\omega_i \cdot n)(1-k) + k\right)$$

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

$$\mathrm{L}_{d_g} = \int_\Omega (1 - P_i) L_i (n \cdot \omega_i)\, d\omega_i$$

$$L_{s_{g_1}} = \int_\Omega \frac{L_i(\omega_i \cdot n)}{R_i}\, d\omega_i$$

$$L_{s_{g_2}} = \int_\Omega \frac{P_i L_i(\omega_i \cdot n)}{R_i}\, d\omega_i$$

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

$$\mathrm{L}_s(\mathrm{p}, \omega_o) = \frac{\alpha^2}{4\pi\left((\omega_o \cdot n)(1-k) + k\right)}\left(F_0 L_{s_{g_1}} + (1 - F_0) L_{s_{g_2}}\right)$$

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

Specular importance sampling

$$k=\frac{(\alpha+1)^2}{8} \text{ or } k=\frac{\alpha^2}{2}$$
$$R_i=\left((n\cdot h_i)^2(\alpha^2-1)+1\right)^2\left((\omega_i\cdot n)(1-k)+k\right)$$
$$P_i=\left(1-(h_i\cdot\omega_o)\right)^5$$

$$L_{s_{g_1}}=\int\limits_{\Omega}\frac{L_i(\omega_i\cdot n)}{R_i}\,d\omega_i=\int\limits_{\Omega}\frac{L_i(\omega_i\cdot n)}{R_i}4(\omega_o\cdot h_i)\,dh_i=\int\limits_0^{2\pi}\int\limits_0^{\frac{\pi}{2}}\frac{L_i(\omega_i\cdot n)}{R_i}4(\omega_o\cdot h_i)\sin(\theta_{h_i})\,d\theta_{h_i}\,d\phi_{h_i}\approx$$
$$\approx\frac{1}{N}\sum_i^N\frac{L_i(\omega_i\cdot n)}{R_i}4(\omega_o\cdot h_i)\sin(\theta_{h_i})\frac{1}{D\cos(\theta_{h_i})\sin(\theta_{h_i})}=\frac{1}{N}\sum_i^N\frac{L_i(\omega_i\cdot n)}{R_i}4(\omega_o\cdot h_i)\frac{1}{D(n\cdot h_i)}=$$
$$=\frac{4}{N}\sum_i^N\frac{L_i(\omega_i\cdot n)(\omega_o\cdot h_i)}{R_i(n\cdot h_i)}\times\frac{1}{D}=\frac{4}{N}\sum_i^N\frac{L_i(\omega_i\cdot n)(\omega_o\cdot h_i)}{R_i(n\cdot h_i)}\times\frac{\pi\left((n\cdot h_i)^2(\alpha^2-1)+1\right)^2}{\alpha^2}=$$
$$=\frac{4\pi}{N\alpha^2}\sum_i^N\frac{L_i(\omega_i\cdot n)(\omega_o\cdot h_i)\left((n\cdot h_i)^2(\alpha^2-1)+1\right)^2}{\left((n\cdot h_i)^2(\alpha^2-1)+1\right)^2\left((\omega_i\cdot n)(1-k)+k\right)(n\cdot h_i)}=\frac{4\pi}{N\alpha^2}\sum_i^N\frac{L_i(\omega_i\cdot n)(\omega_o\cdot h_i)}{\left((\omega_i\cdot n)(1-k)+k\right)(n\cdot h_i)}$$

$$L_{s_{g_2}}=\int\limits_{\Omega}\frac{P_iL_i(\omega_i\cdot n)}{R_i}d\omega_i=\frac{4\pi}{N\alpha^2}\sum_i^N\frac{P_iL_i(\omega_i\cdot n)(\omega_o\cdot h_i)}{\left((\omega_i\cdot n)(1-k)+k\right)(n\cdot h_i)}$$

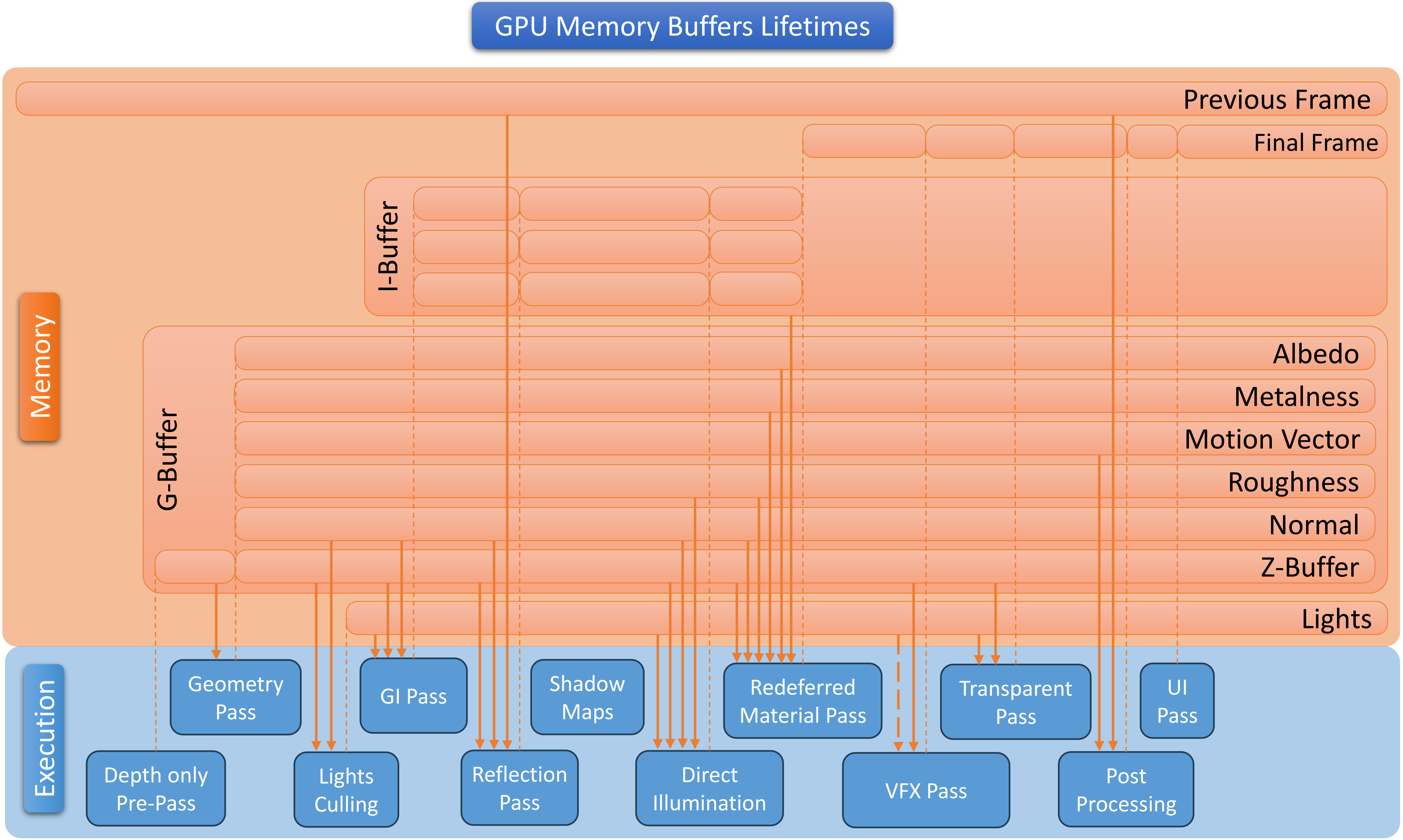
Trowbridge-Reitz GGX:

$$D=\frac{\alpha^2}{\pi\left((n\cdot h_i)^2(\alpha^2-1)+1\right)^2}$$

For low discrepancy sampled random A and B:

$$\theta_{h_i}=arc\cos\left(\sqrt{\frac{1-A}{A(\alpha^2-1)+1}}\right)$$
$$\phi_{h_i}=2\pi B$$

$$h_i=\begin{bmatrix}\sin(\theta_{h_i})\cos(\phi_{h_i})\\ \sin(\theta_{h_i})\sin(\phi_{h_i})\\ \cos(\theta_{h_i})\end{bmatrix}$$
$$\omega_i=2(\omega_o\cdot h_i)\,h_i-\omega_o$$


GPU Memory Buffers Lifetimes
Memory
Previous Frame
Final Frame
I-Buffer
G-Buffer
Albedo
Metalness
Motion Vector
Roughness
Normal
Z-Buffer
Lights
Execution
Geometry Pass
GI Pass
Shadow Maps
Redeferred Material Pass
Transparent Pass
UI Pass
Depth only Pre-Pass
Lights Culling
Reflection Pass
Direct Illumination
VFX Pass
Post Processing