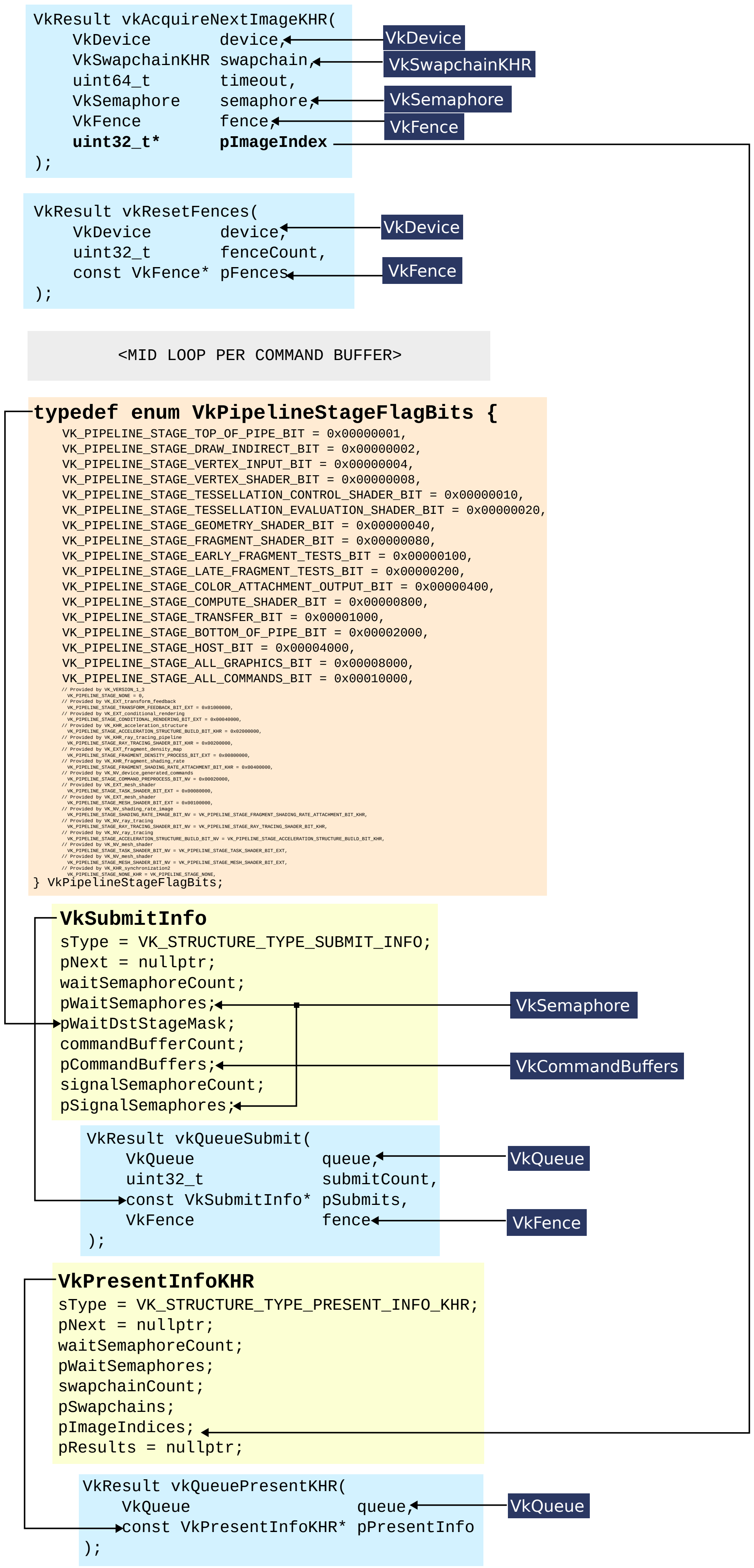
<OUTERMOST LOOP PER DRAW>



```
VkResult vkAcquireNextImageKHR(
    VkDevice        device,
    VkSwapchainKHR  swapchain,
    uint64_t        timeout,
    VkSemaphore     semaphore,
    VkFence         fence,
    uint32_t*       pImageIndex
);
```

VkDevice

VkSwapchainKHR

VkSemaphore

VkFence

```
VkResult vkResetFences(
    VkDevice        device,
    uint32_t        fenceCount,
    const VkFence*  pFences
);
```

VkDevice

VkFence

<MID LOOP PER COMMAND BUFFER>

```
typedef enum VkPipelineStageFlagBits {
   VK_PIPELINE_STAGE_TOP_OF_PIPE_BIT = 0x00000001,
   VK_PIPELINE_STAGE_DRAW_INDIRECT_BIT = 0x00000002,
   VK_PIPELINE_STAGE_VERTEX_INPUT_BIT = 0x00000004,
   VK_PIPELINE_STAGE_VERTEX_SHADER_BIT = 0x00000008,
   VK_PIPELINE_STAGE_TESSELLATION_CONTROL_SHADER_BIT = 0x00000010,
   VK_PIPELINE_STAGE_TESSELLATION_EVALUATION_SHADER_BIT = 0x00000020,
   VK_PIPELINE_STAGE_GEOMETRY_SHADER_BIT = 0x00000040,
   VK_PIPELINE_STAGE_FRAGMENT_SHADER_BIT = 0x00000080,
   VK_PIPELINE_STAGE_EARLY_FRAGMENT_TESTS_BIT = 0x00000100,
   VK_PIPELINE_STAGE_LATE_FRAGMENT_TESTS_BIT = 0x00000200,
   VK_PIPELINE_STAGE_COLOR_ATTACHMENT_OUTPUT_BIT = 0x00000400,
   VK_PIPELINE_STAGE_COMPUTE_SHADER_BIT = 0x00000800,
   VK_PIPELINE_STAGE_TRANSFER_BIT = 0x00001000,
   VK_PIPELINE_STAGE_BOTTOM_OF_PIPE_BIT = 0x00002000,
   VK_PIPELINE_STAGE_HOST_BIT = 0x00004000,
   VK_PIPELINE_STAGE_ALL_GRAPHICS_BIT = 0x00008000,
   VK_PIPELINE_STAGE_ALL_COMMANDS_BIT = 0x00010000,
   // Provided by VK_VERSION_1_3
     VK_PIPELINE_STAGE_NONE = 0,
   // Provided by VK_EXT_transform_feedback
     VK_PIPELINE_STAGE_TRANSFORM_FEEDBACK_BIT_EXT = 0x01000000,
   // Provided by VK_EXT_conditional_rendering
     VK_PIPELINE_STAGE_CONDITIONAL_RENDERING_BIT_EXT = 0x00040000,
   // Provided by VK_KHR_acceleration_structure
     VK_PIPELINE_STAGE_ACCELERATION_STRUCTURE_BUILD_BIT_KHR = 0x02000000,
   // Provided by VK_KHR_ray_tracing_pipeline
     VK_PIPELINE_STAGE_RAY_TRACING_SHADER_BIT_KHR = 0x00200000,
   // Provided by VK_EXT_fragment_density_map
     VK_PIPELINE_STAGE_FRAGMENT_DENSITY_PROCESS_BIT_EXT = 0x00800000,
   // Provided by VK_KHR_fragment_shading_rate
     VK_PIPELINE_STAGE_FRAGMENT_SHADING_RATE_ATTACHMENT_BIT_KHR = 0x00400000,
   // Provided by VK_NV_device_generated_commands
     VK_PIPELINE_STAGE_COMMAND_PREPROCESS_BIT_NV = 0x00020000,
   // Provided by VK_EXT_mesh_shader
     VK_PIPELINE_STAGE_TASK_SHADER_BIT_EXT = 0x00080000,
   // Provided by VK_EXT_mesh_shader
     VK_PIPELINE_STAGE_MESH_SHADER_BIT_EXT = 0x00100000,
   // Provided by VK_NV_shading_rate_image
     VK_PIPELINE_STAGE_SHADING_RATE_IMAGE_BIT_NV = VK_PIPELINE_STAGE_FRAGMENT_SHADING_RATE_ATTACHMENT_BIT_KHR,
   // Provided by VK_NV_ray_tracing
     VK_PIPELINE_STAGE_RAY_TRACING_SHADER_BIT_NV = VK_PIPELINE_STAGE_RAY_TRACING_SHADER_BIT_KHR,
   // Provided by VK_NV_ray_tracing
     VK_PIPELINE_STAGE_ACCELERATION_STRUCTURE_BUILD_BIT_NV = VK_PIPELINE_STAGE_ACCELERATION_STRUCTURE_BUILD_BIT_KHR,
   // Provided by VK_NV_mesh_shader
     VK_PIPELINE_STAGE_TASK_SHADER_BIT_NV = VK_PIPELINE_STAGE_TASK_SHADER_BIT_EXT,
   // Provided by VK_NV_mesh_shader
     VK_PIPELINE_STAGE_MESH_SHADER_BIT_NV = VK_PIPELINE_STAGE_MESH_SHADER_BIT_EXT,
   // Provided by VK_KHR_synchronization2
     VK_PIPELINE_STAGE_NONE_KHR = VK_PIPELINE_STAGE_NONE,
} VkPipelineStageFlagBits;
```

```
VkSubmitInfo
sType = VK_STRUCTURE_TYPE_SUBMIT_INFO;
pNext = nullptr;
waitSemaphoreCount;
pWaitSemaphores;
pWaitDstStageMask;
commandBufferCount;
pCommandBuffers;
signalSemaphoreCount;
pSignalSemaphores;
```

VkSemaphore

VkCommandBuffers

```
VkResult vkQueueSubmit(
    VkQueue             queue,
    uint32_t            submitCount,
    const VkSubmitInfo* pSubmits,
    VkFence             fence
);
```

VkQueue

VkFence

```
VkPresentInfoKHR
sType = VK_STRUCTURE_TYPE_PRESENT_INFO_KHR;
pNext = nullptr;
waitSemaphoreCount;
pWaitSemaphores;
swapchainCount;
pSwapchains;
pImageIndices;
pResults = nullptr;
```

```
VkResult vkQueuePresentKHR(
    VkQueue                 queue,
    const VkPresentInfoKHR* pPresentInfo
);
```

VkQueue