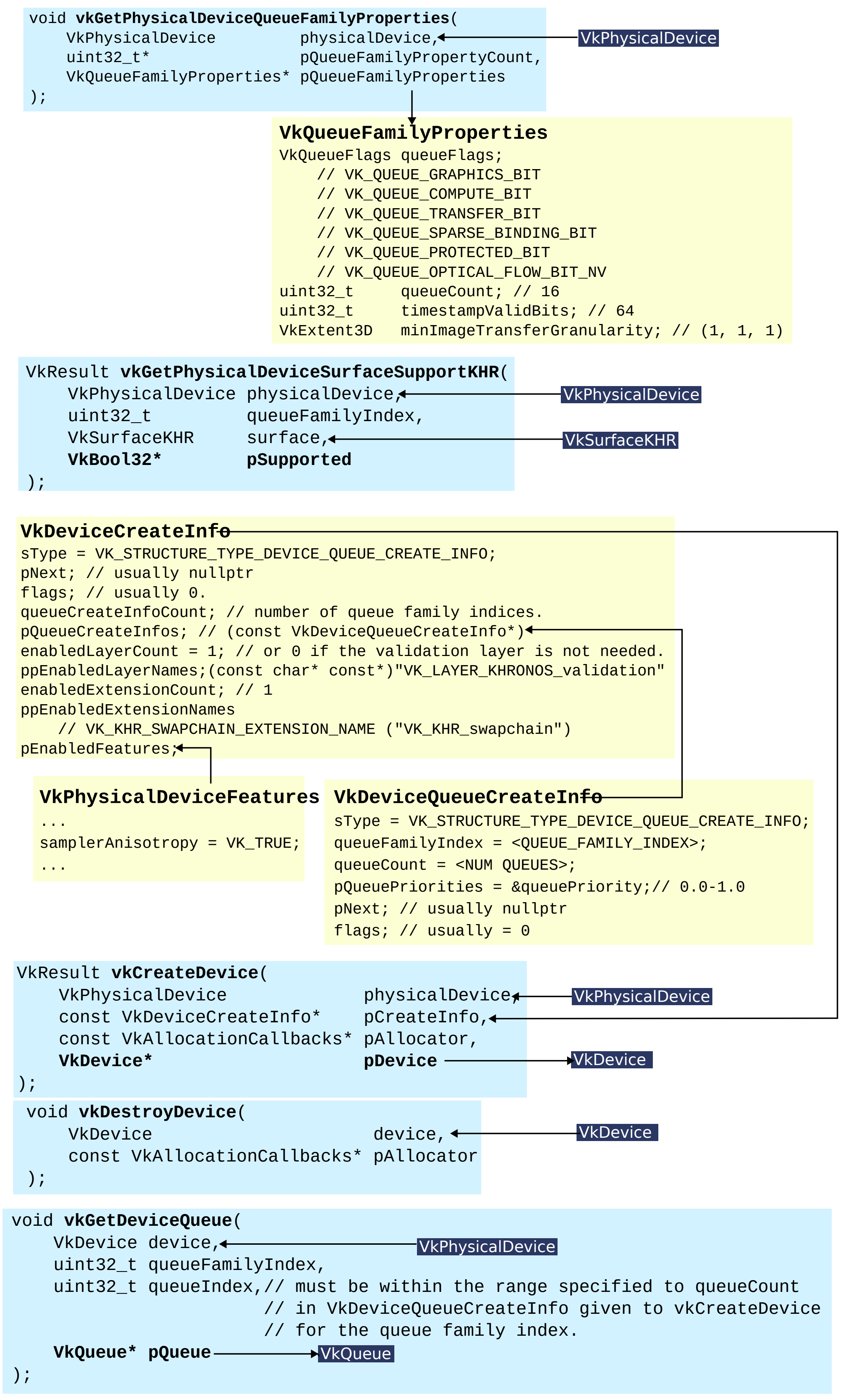
VkDevice & VkQueue



```
void vkGetPhysicalDeviceQueueFamilyProperties(
    VkPhysicalDevice         physicalDevice,
    uint32_t*                pQueueFamilyPropertyCount,
    VkQueueFamilyProperties* pQueueFamilyProperties
);
```

```
VkQueueFamilyProperties
VkQueueFlags queueFlags;
    // VK_QUEUE_GRAPHICS_BIT
    // VK_QUEUE_COMPUTE_BIT
    // VK_QUEUE_TRANSFER_BIT
    // VK_QUEUE_SPARSE_BINDING_BIT
    // VK_QUEUE_PROTECTED_BIT
    // VK_QUEUE_OPTICAL_FLOW_BIT_NV
uint32_t     queueCount; // 16
uint32_t     timestampValidBits; // 64
VkExtent3D   minImageTransferGranularity; // (1, 1, 1)
```

```
VkResult vkGetPhysicalDeviceSurfaceSupportKHR(
    VkPhysicalDevice physicalDevice,
    uint32_t         queueFamilyIndex,
    VkSurfaceKHR     surface,
    VkBool32*        pSupported
);
```

```
VkDeviceCreateInfo
sType = VK_STRUCTURE_TYPE_DEVICE_QUEUE_CREATE_INFO;
pNext; // usually nullptr
flags; // usually 0.
queueCreateInfoCount; // number of queue family indices.
pQueueCreateInfos; // (const VkDeviceQueueCreateInfo*)
enabledLayerCount = 1; // or 0 if the validation layer is not needed.
ppEnabledLayerNames;(const char* const*)"VK_LAYER_KHRONOS_validation"
enabledExtensionCount; // 1
ppEnabledExtensionNames
    // VK_KHR_SWAPCHAIN_EXTENSION_NAME ("VK_KHR_swapchain")
pEnabledFeatures;
```

```
VkPhysicalDeviceFeatures
...
samplerAnisotropy = VK_TRUE;
...
```

```
VkDeviceQueueCreateInfo
sType = VK_STRUCTURE_TYPE_DEVICE_QUEUE_CREATE_INFO;
queueFamilyIndex = <QUEUE_FAMILY_INDEX>;
queueCount = <NUM QUEUES>;
pQueuePriorities = &queuePriority;// 0.0-1.0
pNext; // usually nullptr
flags; // usually = 0
```

```
VkResult vkCreateDevice(
    VkPhysicalDevice             physicalDevice,
    const VkDeviceCreateInfo*    pCreateInfo,
    const VkAllocationCallbacks* pAllocator,
    VkDevice*                    pDevice
);
```

```
void vkDestroyDevice(
    VkDevice                     device,
    const VkAllocationCallbacks* pAllocator
);
```

```
void vkGetDeviceQueue(
    VkDevice device,
    uint32_t queueFamilyIndex,
    uint32_t queueIndex,// must be within the range specified to queueCount
                        // in VkDeviceQueueCreateInfo given to vkCreateDevice
                        // for the queue family index.
    VkQueue* pQueue
);
```