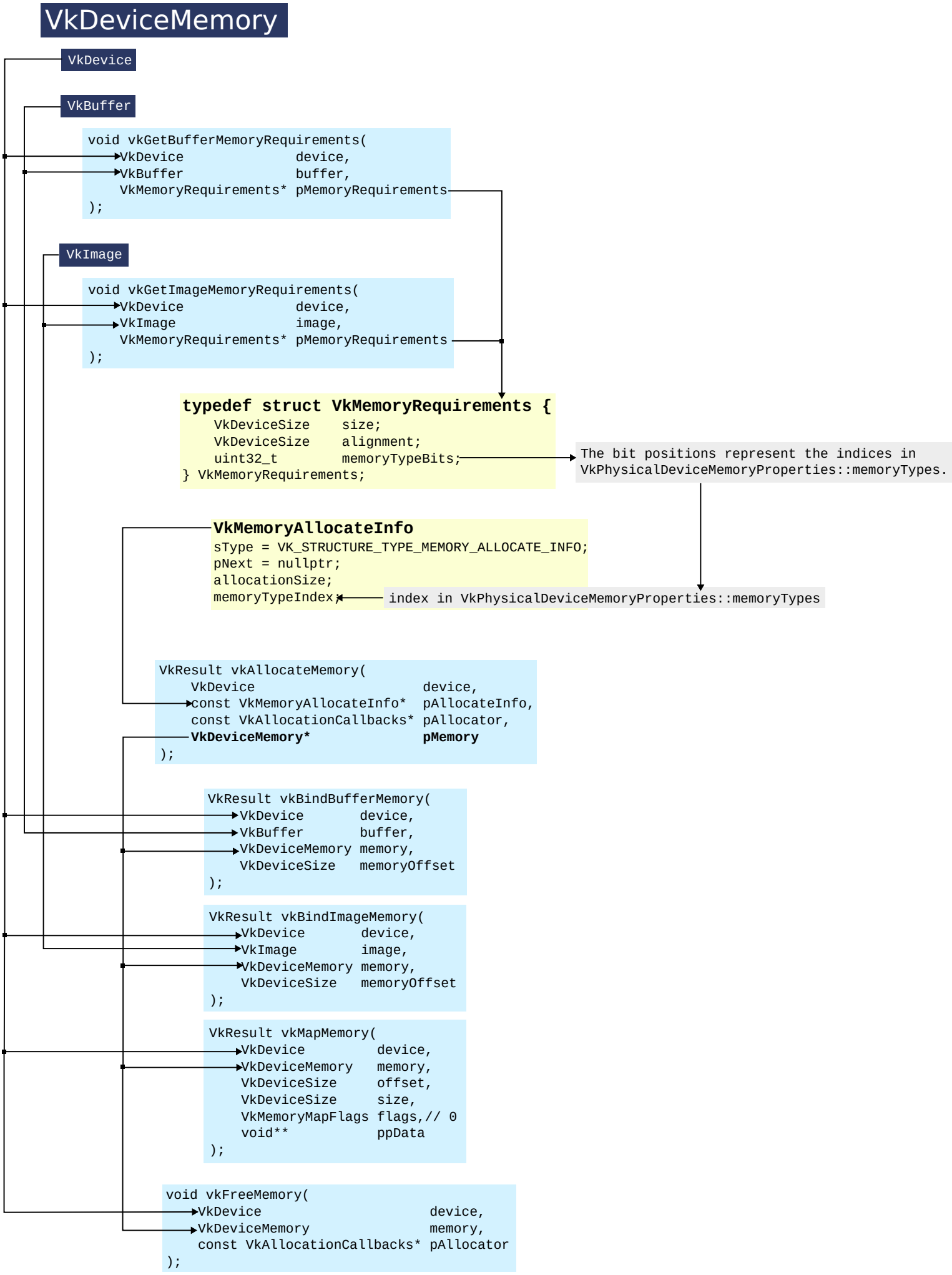
VkDeviceMemory
VkDevice
VkBuffer
void vkGetBufferMemoryRequirements(
VkDevice device,
VkBuffer buffer,
VkMemoryRequirements* pMemoryRequirements
);
VkImage
void vkGetImageMemoryRequirements(
VkDevice device,
VkImage image,
VkMemoryRequirements* pMemoryRequirements
);
typedef struct VkMemoryRequirements {
VkDeviceSize size;
VkDeviceSize alignment;
uint32_t memoryTypeBits;
} VkMemoryRequirements;
The bit positions represent the indices in VkPhysicalDeviceMemoryProperties::memoryTypes.
VkMemoryAllocateInfo
sType = VK_STRUCTURE_TYPE_MEMORY_ALLOCATE_INFO;
pNext = nullptr;
allocationSize;
memoryTypeIndex;
index in VkPhysicalDeviceMemoryProperties::memoryTypes
VkResult vkAllocateMemory(
VkDevice device,
const VkMemoryAllocateInfo* pAllocateInfo,
const VkAllocationCallbacks* pAllocator,
VkDeviceMemory* pMemory
);
VkResult vkBindBufferMemory(
VkDevice device,
VkBuffer buffer,
VkDeviceMemory memory,
VkDeviceSize memoryOffset
);
VkResult vkBindImageMemory(
VkDevice device,
VkImage image,
VkDeviceMemory memory,
VkDeviceSize memoryOffset
);
VkResult vkMapMemory(
VkDevice device,
VkDeviceMemory memory,
VkDeviceSize offset,
VkDeviceSize size,
VkMemoryMapFlags flags,// 0
void** ppData
);
void vkFreeMemory(
VkDevice device,
VkDeviceMemory memory,
const VkAllocationCallbacks* pAllocator
);