Vulkan Video decoding

A brief overview of VK_KHR_video_{queue,decode}

Víctor Jáquez

vjaquez@igalia.com

17 September 2021

Who am I?

- Mexican.
- Igalian.
- GStreamer developer.
  - GStreamer-VAAPI maintainer (to be deprecated).
- I'm not a Mesa developer (...).

Graphics team

- Other Igalians are Mesa developers.
  - Mesa, Kernel, Vulkan, etc.

This talk

An overview of the concepts used to decode video with Video Vulkan extension, using as example the usage of the API in a GStreamer element.

Its purpose is merely educational.

Perhaps a new gallium frontend?

- Cross-platform API.
- Backed by Khronos.
- Aimed at graphics developers for a wide range of different tasks.
- Described by a common specification.
- Implemented by different drivers
  - Usually provided by GPU vendors
  - and Mesa 👍

Vulkan Video TSG

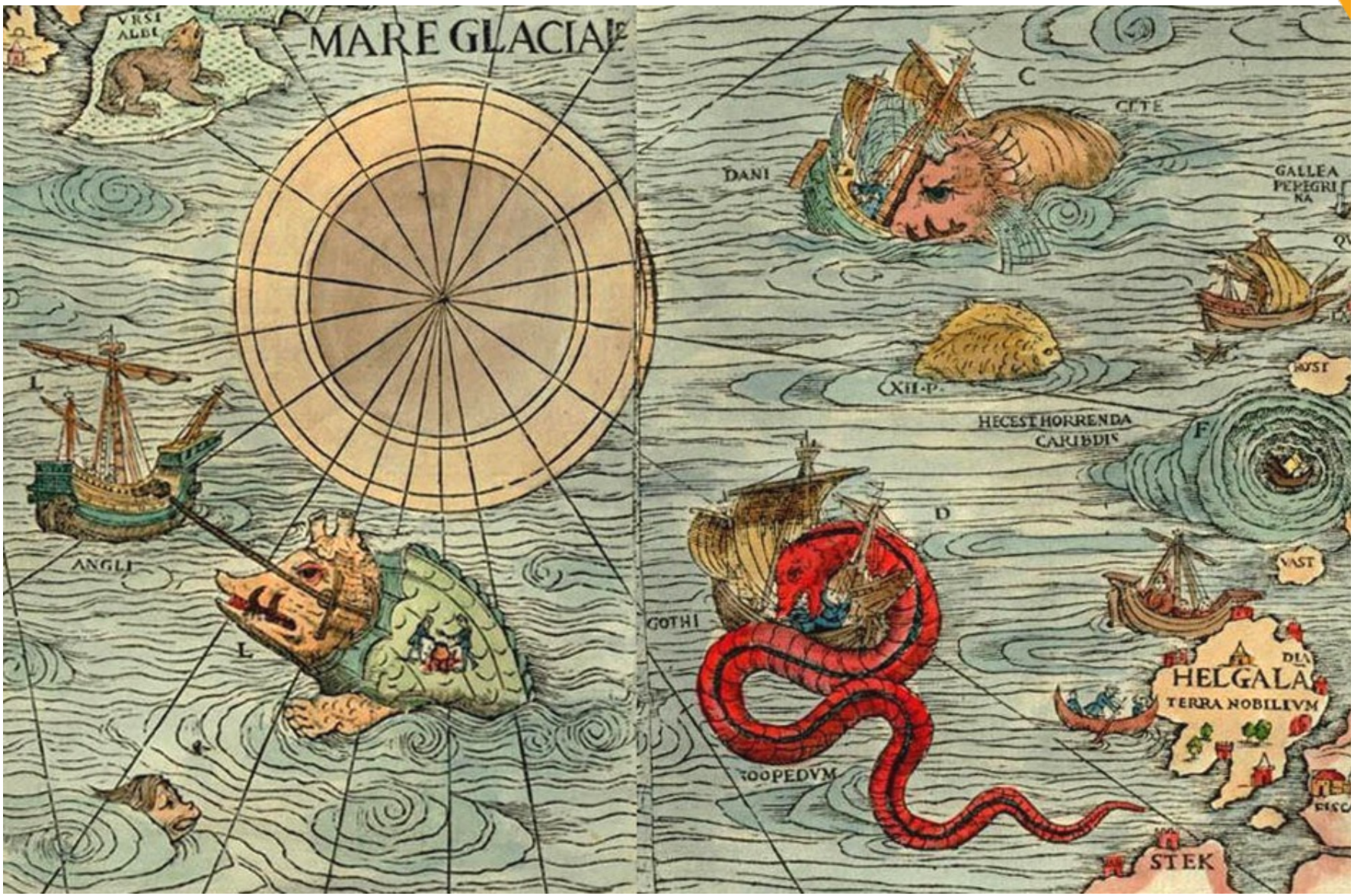
- In May 2018, the Vulkan Video Technical Specification Group was proposed:

To integrate hardware accelerated video compression and decompression into the Vulkan API.

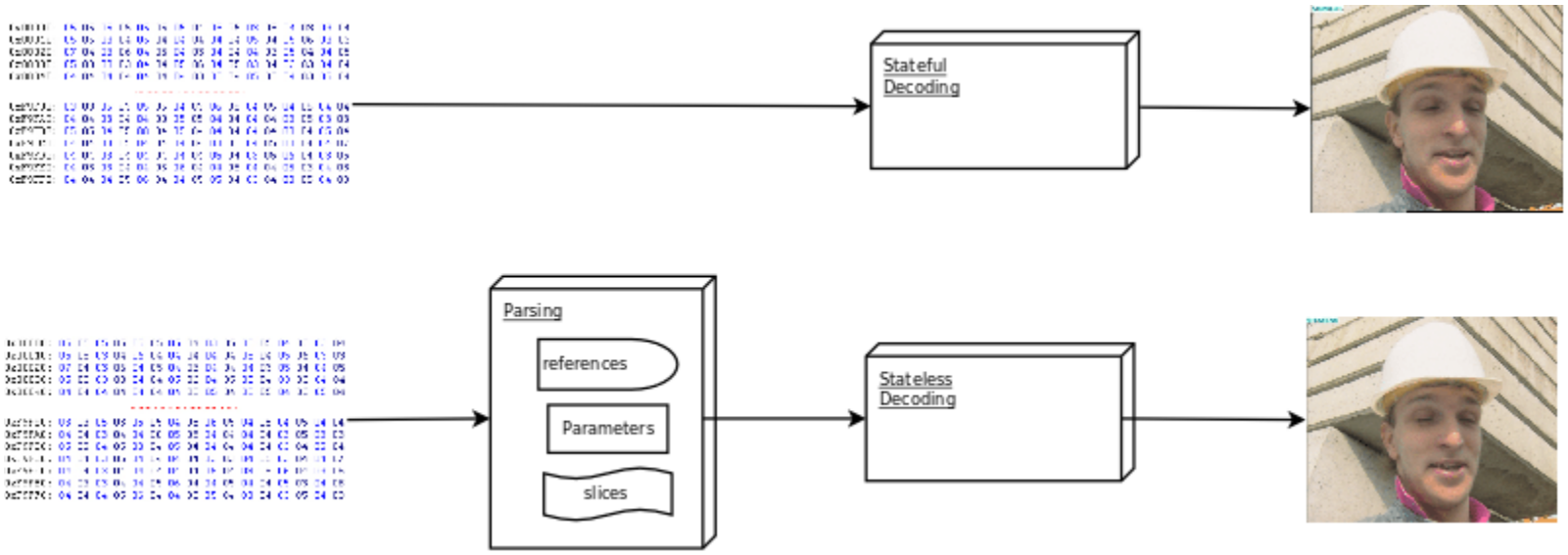
- In April of this year, the Vulkan Video TSG published an introduction to the specification:

https://www.khronos.org/blog/an-introduction-to-vulkan-video

Disclaimer: Hic Svnt Dracones



Stateful vs Stateless



- GStreamer Vulkan
- GStreamer parsers
- Codec state handlers

Current Support

- Vulkan Video Provisional Specification (Vulkan Spec with Video Extensions)

https://www.khronos.org/registry/vulkan/specs/1.2-extensions/html/vkspec.html

- NVIDIA latest driver (>=455.50.12)

https://developer.nvidia.com/vulkan-driver

- Vulkan Video samples

https://github.com/nvpro-samples/vk_video_samples

Vulkanh264dec

WIP Merge Request

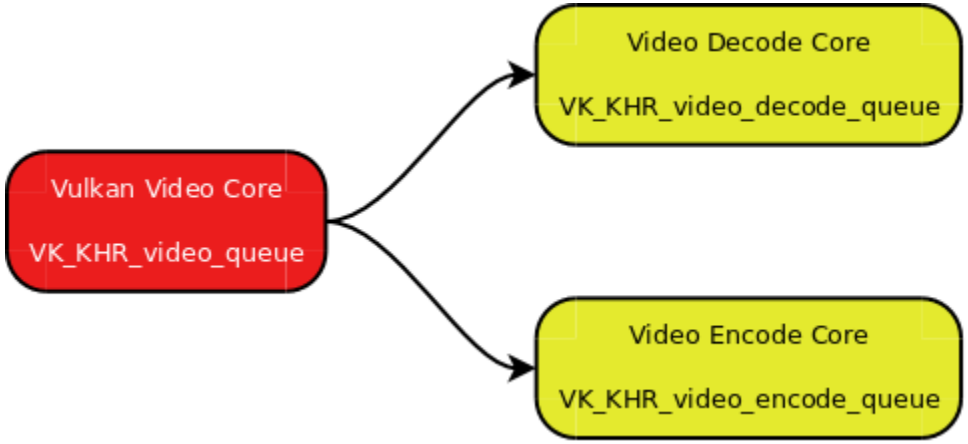
https://gitlab.freedesktop.org/gstreamer/gst-plugins-bad/-/merge_requests/2397

Vulkan Video Design Goals

- Low-level stateless management of hardware for efficiency and flexibility.
- Suitable for low-power/memory embedded devices to high-performance servers.
- Distribution of video processing across multiple CPU cores and video-codec devices.
- Closer integration with Vulkan Graphics and Displays.

Extension overview

- **VK_KHR_video_queue**: API calls that are common to decode and encode operations.
- **VK_KHR_video_decode_queue**: API calls used only in decode operations.
- **VK_KHR_video_encode_queue**: API calls used only in encode operations.



Setup

- Find a physical device with VK_KHR_VIDEO_QUEUE_EXTENSION_NAME and VK_KHR_VIDEO_DECODE_QUEUE_EXTENSION_NAME
- Find in that device a queue with VK_QUEUE_VIDEO_DECODE_BIT_KHR family and the H.264 codec operations.
- Query device's video capabilities and match them with your stream.
- Allocate bitstream VkBuffers and DPB VkImages.

VkVideoSession

- Maintains the context associated with the stream.
  - One per stream.
- Specifies the video profile and maximum parameters.
- Video Session object maintains the device memory heaps.
  - The application allocates and binds VkDeviceMemory objects to the Video Session object which uses it for its memory heaps.

VkVideoSessionParameters

- Contains processing parameters.
  - It belongs to a VkVideoSession.
- Use multiple VkVideoSessionParameters to process a stream.
  - An object can apply to the whole stream or a portion.
- Can add parameters to a the VkVideoSessionParameters object

vkCmdDecodeVideoKHR

- Only API call provided in the VK_KHR_video_decode_queue extension.
- Command buffers and bitstream data are built for the video device in memory before submission to the GPU.

Video command buffer

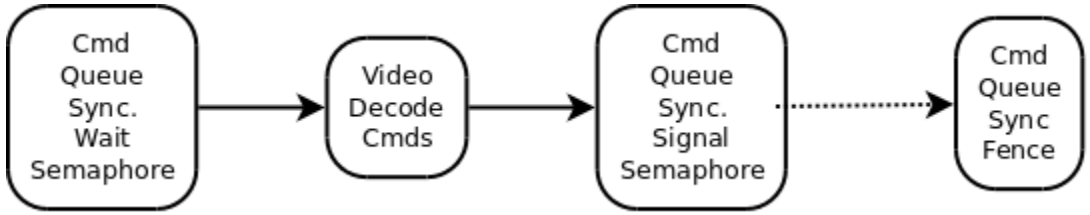
- Several decode or encode operations may be recorded in the same command buffer
  - All relying on the same set of resources and settings
  - API calls to mark the scope of video command control parameters during a session:

vkCmdBeginVideoCodingKHR

vkCmdEndVideoCodingKHR

VkCommandBuffers Queue Submission

- Regular Vulkan Queue Submit Sequence:
  - One or more command buffers can be submitted.
  - Command buffer sequences are synchronized by semaphores.
  - Command buffer sequences can be synchronized with the host CPU by a fence.



Questions?