Vulkanised!
Vulkan 1.3 Overview
Ralph Potter, Samsung
Vulkan 1.3

Supportable on OpenGL ES 3.1-class hardware
Same hardware baseline as Vulkan 1.0

Integrates significant requested functionality proven as extensions
23 extensions promoted to the Vulkan 1.3 Core API
Mandated support for new functionality ensures availability on all Vulkan 1.3 implementations

Hardware vs. Software (mostly)
API usability and infrastructure changes in Vulkan 1.3 Core API
(programming model changes, API interfaces, compiler etc.)
New hardware functionality in extensions and Roadmap
Vulkan 1.3

VK_EXT_4444_formats
VK_EXT_extended_dynamic_state
VK_EXT_extended_dynamic_state2
VK_EXT_image_robustness
VK_EXT_inline_uniform_block
VK_EXT_pipeline_creation_cache_control
VK_EXT_pipeline_creation_feedback
VK_EXT_private_data
VK_EXT_shader_demote_to_helper_invocation
VK_EXT_subgroup_size_control
VK_EXT_texel_buffer_alignment
VK_EXT_texture_compression_astc_hdr

VK_EXT_tooling_info
VK_EXT_ycbcr_2plane_4444_formats
VK_KHR_copy_commands2
VK_KHR_dynamic_rendering
VK_KHR_format_feature_flags2
VK_KHR_maintenance4
VK_KHR_shader_integer_dot_product
VK_KHR_shader_non_semantic_info
VK_KHR_shader_terminate_invocation
VK_KHR_synchronization2
VK_KHR_zero_initialize_workgroup_memory
Dynamic Rendering

- Render passes have been a recurring area of developer feedback
- API improvements for single pass render passes
- Removes coupling to VkRenderPass and VkFramebuffer objects
- Single pass performance is equivalent to the existing API

Streamlining Render Passes Blog: [https://www.khronos.org/blog/streamlining-render-passes](https://www.khronos.org/blog/streamlining-render-passes)
Pipeline Creation

● **VK_EXT_pipeline_creation_cache_control**
  ○ Gives developers more control over how and when pipelines are compiled

● **VK_EXT_pipeline_creation_feedback**
  ○ Provides information about the compiled pipelines, making it easier to profile and debug pipeline issues

Extension Specifications:
https://www.khronos.org/registry/vulkan/specs/1.3-extensions/man/html/VK_EXT_pipeline_creation_cache_control.html
https://www.khronos.org/registry/vulkan/specs/1.3-extensions/man/html/VK_EXT_pipeline_creation_feedback.html
Extended Dynamic State

- Static state changes require VkPipeline objects
- Promoted VK_EXT_extended_dynamic_state and VK_EXT_extended_dynamic_state2
- Making state dynamic reduces the number of VkPipeline objects
- Dynamic state for patch control points and logic op are not promoted in Vulkan 1.3

Extension Specifications:
https://www.khronos.org/registry/vulkan/specs/1.3-extensions/man/html/VK_EXT_extended_dynamic_state.html
https://www.khronos.org/registry/vulkan/specs/1.3-extensions/man/html/VK_EXT_extended_dynamic_state2.html
Image Formats

- **VK_EXT_texture_compression_astc_hdr**
  - VK_FORMAT_ASTC_4x4_SFLOAT_BLOCK
  - ...

- **VK_EXT_4444_formats**
  - VK_FORMAT_A4R4G4B4_UNORM_PACK16
  - VK_FORMAT_A4B4G4R4_UNORM_PACK16

- **VK_EXT_ycbcr_2plane_444_formats**
  - VK_FORMAT_G10X6_B10X6R10X6_2PLANE_444_UNORM_3PACK16
  - VK_FORMAT_G12X4_B12X4R12X4_2PLANE_444_UNORM_3PACK16
  - VK_FORMAT_G16_B16R16_2PLANE_444_UNORM
  - VK_FORMAT_G8_B8R8_2PLANE_444_UNORM

- The formats added by these extensions are optional in core
- But implementations will now recognize the enums and give valid results to queries
Shader Extensions

- **VK_KHR_shader_integer_dot_product**
  - Particularly useful for ML applications, but available for general usage

- **VK_KHR_shader_non_semantic_info**
  - Adds support for SPIR-V instructions that are ignored by the driver, but may aid tools such as debuggers

- **VK_KHR_shader_terminate_invocation**
  - Replace SPIR-V OpKill with better specified behaviour
  - This is a better match for the “discard” instruction in GLSL

- **VK_EXT_shader_demote_to_helper_invocation**
  - This is a better match for the “discard” instruction in HLSL
Usability, Infrastructure and Maintenance

- **VK_KHR_synchronization2**
  - Usability improvements for synchronization

- **VK_EXT_image_robustness**
  - Defined behaviour for out-of-bounds operations

- **VK_EXT_inline_uniform_block**
  - Store uniforms in descriptor set storage

- **VK_EXT_private_data, VK_EXT_tooling_info**
  - Improvements to layer and tooling support

- **VK_EXT_subgroup_size_control**
  - Greater control/introspection of subgroup behaviour

- **VK_EXT_texel_buffer_alignment**
  - Alignment queries for uniform and storage texel buffers

- **VK_KHR_copy_commands2, VK_KHR_format_feature_flags2**
  - Infrastructure extensions to supporting future API extension

- **VK_KHR_maintenance4**
  - Bugfixes and usability improvements

- **VK_KHR_zero_initialize_workgroup_memory**
  - Zero-initialized memory for security conscious use cases
Other Changes

- Support for SPIR-V version 1.6
- Support for the following features is now mandatory:
  - bufferDeviceAddress
  - vulkanMemoryModel
  - vulkanMemoryModelDeviceScope
  - vulkanMemoryModelAvailabilityVisibilityChains

- maxInlineUniformTotalSize limit is added to provide the total size of all inline uniform block bindings in a pipeline layout
More Information

**Vulkan:** [https://www.vulkan.org/](https://www.vulkan.org/)
- Press Release: [https://khr.io/xj](https://khr.io/xj)
- Final specification: [https://khr.io/vulkan13spec](https://khr.io/vulkan13spec)
- Spec GitHub Repo: [https://khr.io/vulkan13github](https://khr.io/vulkan13github)
- Discord Link for community discussion: [https://khr.io/vulkan13discord](https://khr.io/vulkan13discord)
- Vulkan 1.3 Blog post: [https://www.khronos.org/blog/vulkan-1.3-and-roadmap-2022](https://www.khronos.org/blog/vulkan-1.3-and-roadmap-2022)
- Google Android Baseline 2021 Profile blog: [https://www.khronos.org/blog/vulkan-1.3-and-roadmap-2022](https://www.khronos.org/blog/vulkan-1.3-and-roadmap-2022)

Another Vulkanised Webinar coming after SDK release

Will cover SDK features in detail