KHRONOS™ GROUP

Vulkan DevU

The Vulkan Working Group

Vancouver
30 January 2017

Schedule - Morning

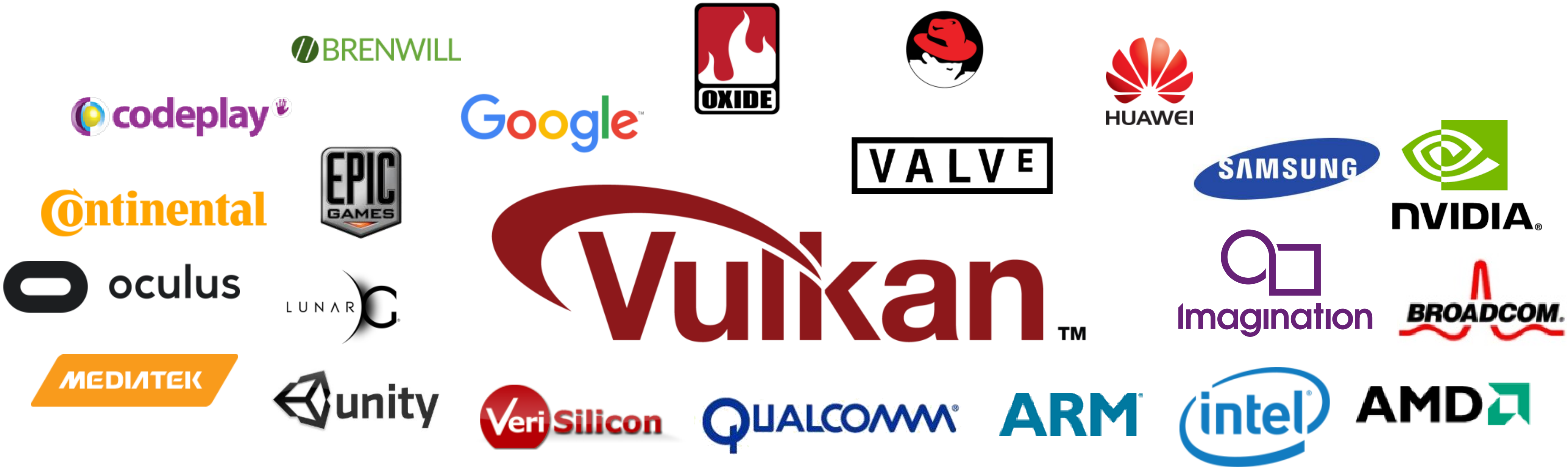
Time	Topic	Speaker(s)
9:00-9:45	Introduction	Tom Olson / ARM
	When and why should you use Vulkan?	Michael Worcester / Imagination
	Performance benefits of using Vulkan	Cort Stratton / Google
9:45-10:30	Getting started with Vulkan: Loader, layers, and other resources	Mark Young / LunarG
10:30-10:45	**BREAK**	
10:45-11:00	vkTrace for the desktop and Android	Karl Schultz / LunarG
11:00-11:15	Vulkan C++ bindings	Markus Tavenrath / NVIDIA
11:15-11:45	Making SPIR-V modules	John Kessenich / Google
11:45-12:45	**LUNCH**	

Schedule - Afternoon

Time	Topic	Speaker(s)
12:45-1:45	Vulkan Tutorial: Let's get into some code!	Karl Schulz / LunarG
1:45-2:15	Synchronization: Keep your GPU fed without getting bitten	Tobias Hector / Imagination
2:15-2:30	**BREAK**	
2:30-3:15	Android game development case study: Porting Vainglory to Vulkan	Alon Or-bach / Samsung
3:15-4:00	Bringing Vulkan to VR	Cass Everitt / Oculus VR
4:00-4:15	**BREAK**	
4:15-5:15	Q&A / Panel / Beer	Everyone
5:15-5:30	About Khronos	Neil Trevett / NVIDIA / Khronos President

- **Wifi: SSID <>, psw**

Who are we, and why are we here?



- **Members of the Vulkan Working Group include**
  - Game and middleware developers
  - CPU and GPU silicon IHVs
  - Consumer device manufacturers
  - Platform owners

Who are we, and why are we here?

Team Vulkan in Seattle, January 2016

What is Vulkan?

- **A 3D graphics API for the next 20 years**
  - Logical successor to OpenGL / OpenGL ES
  - *THE* cross-platform API for programming graphics HW
  - Modern, efficient design
  - An open, industry-controlled standard

- **Almost one year old!**
  - Released in February 2016
  - Widely supported on desktop
  - Exposed in Android 7.0

Problems with traditional 3D APIs

- **Programming model doesn't match GPU HW**
  - Driver magic hides the mismatch

- **CPU intensive**
  - Lots of state validation, dependency tracking

- **Complex, buggy, unpredictable drivers**
  - Different bugs and fast-paths on every GPU

- **Fundamentally single-threaded**
  - Can't use multi-core CPUs effectively

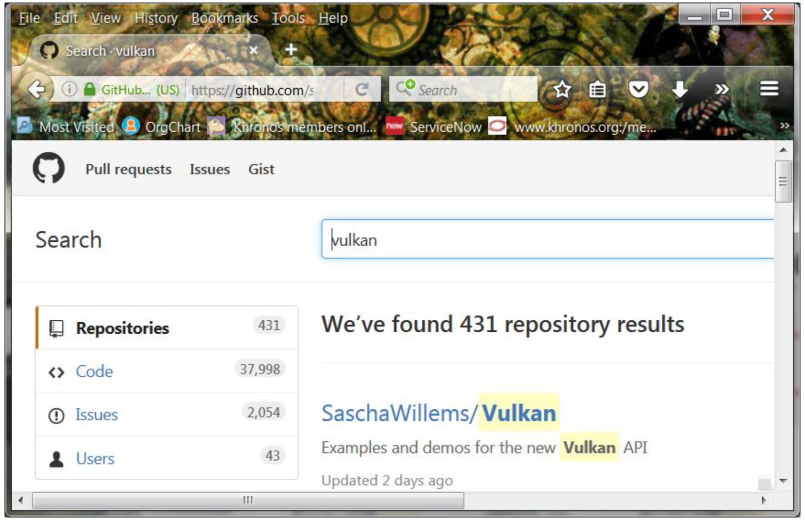
- **…not to mention twenty years of legacy cruft**

All of these problems are solved in Vulkan

© Copyright Khronos Group 2016

Vulkan and the Community

- **A strong commitment to openness**
  - All of our resources are in open source
  - …including the specification!
  - All of our SW is under Apache 2.0
  - See https://github.com/KhronosGroup/
- **Working hard to be responsive**
  - Weekly meeting devoted to GitHub input



vulkan

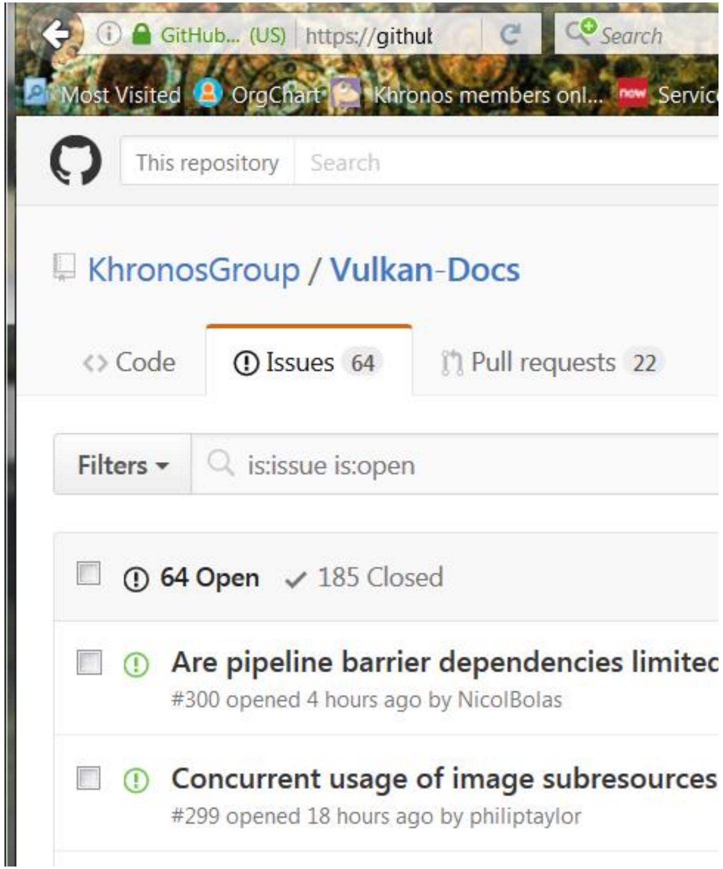
Repositories	431
Code	37,998
Issues	2,054
Users	43

We've found 431 repository results

SaschaWillems/Vulkan

Examples and demos for the new Vulkan API

Updated 2 days ago



KhronosGroup / Vulkan-Docs

is:issue is:open

□ 64 Open ✓ 185 Closed

□ Are pipeline barrier dependencies limitec
#300 opened 4 hours ago by NicolBolas

□ Concurrent usage of image subresources
#299 opened 18 hours ago by philiptaylor