Significant Khronos API Ecosystem Advances

- OpenGL 4.5 specification released
- First conformant OpenGL ES 3.1 Implementations
- OpenGL ES 3.1 + (AEP) Android Extension Pack mandated for Android L
- WebGL approaching pervasive availability across desktop and mobile
- SPIR v2.0 (OpenCL C intermediate representation) released
- Next Generation OpenGL Initiative - Call for Participation
Khronos Connects Software to Silicon

Open Consortium creating ROYALTY-FREE, OPEN STANDARD APIs for hardware acceleration

Defining the roadmap for low-level silicon interfaces needed on every platform

Graphics, compute, rich media, vision, sensor and camera processing

Rigorous specifications AND conformance tests for cross-vendor portability

Acceleration APIs BY the Industry FOR the Industry

Well over a BILLION people use Khronos APIs Every Day...
Access to 3D on Over 2 BILLION Devices

300M Desktops / year

1.9B Mobiles / year

1B Browsers / year

Worldwide Device Shipments by Segment (Thousands of Units)

<table>
<thead>
<tr>
<th>Device Type</th>
<th>2012</th>
<th>2013</th>
<th>2014</th>
<th>2015</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC (Desk-based and Notebook)</td>
<td>341,171</td>
<td>269,342</td>
<td>277,930</td>
<td>260,491</td>
</tr>
<tr>
<td>Tablet (Ultramobile)</td>
<td>119,429</td>
<td>179,911</td>
<td>243,450</td>
<td>325,565</td>
</tr>
<tr>
<td>Mobile Phone</td>
<td>1,746,177</td>
<td>2,804,334</td>
<td>1,853,425</td>
<td>1,964,788</td>
</tr>
<tr>
<td>Other Ultramobiles (Hybrid and clamshell)</td>
<td>9,144</td>
<td>17,195</td>
<td>19,635</td>
<td>63,835</td>
</tr>
<tr>
<td>Total</td>
<td>2,210,323</td>
<td>2,360,462</td>
<td>2,574,451</td>
<td>2,621,678</td>
</tr>
</tbody>
</table>

Source: Gartner (December 2013)
Continuing OpenGL Innovation

Bringing state-of-the-art functionality to cross-platform graphics

OpenGL 4.5
OpenGL 4.4
OpenGL 4.3
OpenGL 4.2
OpenGL 4.1
OpenGL 3.3/4.0
OpenGL 3.2
OpenGL 3.1

OpenGL 2.0 | OpenGL 2.1 | OpenGL 3.0


DirectX 9.0c | DirectX 10.0 | DirectX 10.1 | DirectX 11 | DirectX 11.1 | DirectX 11.2
What is new in OpenGL 4.5?

• Direct State Access (DSA)
  - Object accessors enable state to be queried and modified without binding objects to contexts - efficiency and flexibility for applications, tools and middleware

• Flush Control
  - Application can control flushing of pending commands before context switching - enabling high-performance multithreaded applications

• Robustness
  - Providing a secure platform for applications such as WebGL browsers e.g. preventing a GPU reset affecting any other running applications

• DX11 emulation features
  - Easier porting of applications between OpenGL and Direct3D

• OpenGL ES 3.1 API and shader compatibility
  - Enables development and execution of the latest OpenGL ES applications on desktop systems
OpenGL ES 3.1 Launched at GDC March 2014

- Headline features - Compute Shaders and Draw-Indirect
  - Compute shaders - general compute - can create geometry and rendering data
  - Draw-Indirect enables arguments from a buffer object that the GPU creates
- Backward compatible driver upgrade for many SOCs
  - Backward compatible with 2.0/3.0 so apps can incrementally adopt features
- Desktop OpenGL can be used for mobile development
  - OpenGL 4.5 enables use of a “OpenGL ES 3.1 context”
OpenGL ES 3.1 Adoption Momentum

- Widespread industry participation to release specification in March 2014
  - Tool and Game Engine Developers, GPU Designers, SoC Vendors
  - Platform Owners, End Equipment Makers, Middleware ISVs

- Khronos launched the OpenGL ES 3.1 Adopters program in June 2014
  - Broad set of conformance tests to ensure reliable cross-vendor operation

- Google announced that OpenGL ES 3.1 is standard in Android L
  - At Google IO June 2014

- First wave of GPU vendors conformant in July 2014
  - ARM, Imagination Technologies, Intel, NVIDIA, Qualcomm (in process) and Vivante
  - [http://www.khronos.org/conformance/adopters/conformant-products#opengles](http://www.khronos.org/conformance/adopters/conformant-products#opengles)
Google Android Extension Pack (AEP)

• Set of extensions for OpenGL ES 3.1
  - Accessible through a single query
  - Functionality to support AAA games

• Functionality from desktop OpenGL
  - Tessellation
    - Improves the detail of geometry rendered
  - Geometry shaders
    - Add details and shadows
  - ASTC Texture Compression
    - High quality texture compression

• Enables premium graphics effects
  - Deferred rendering
  - Physically-based shading
  - High Dynamic Range tone mapping
  - Global Illumination and reflection
  - Smoke and particle effects

Epic’s Rivalry demo using full Unreal Engine 4
Running in real-time on NVIDIA Tegra K1 with OpenGL ES 3.1 + AEP
https://www.youtube.com/watch?v=jRr-G95GdaM
Pervasive WebGL

- WebGL on EVERY major desktop browser
  - And coming to all mobile browsers
- Portable (NO source change) 3D applications are possible for the first time

http://caniuse.com/#feat=webgl
OpenCL as Parallel Language Backend

- **WebCL**: JavaScript binding for initiation of OpenCL C kernels
- **Halide**: Language for image processing and computational photography
- **C++ AMP**: MulticoreWare open source project on Bitbucket
- **aparapi**: Embedded array language for Haskell
- **Java**: Java language extensions for parallelism
- **River Trail**: Language extensions to JavaScript
- **OpenACC**: Compiler directives for Fortran, C and C++
- **PyOpenCL**: Python wrapper around OpenCL
- **Harlan**: High level language for GPU programming

OpenCL provides vendor optimized, cross-platform, cross-vendor access to heterogeneous compute resources
Widening OpenCL Ecosystem

OpenCL C Kernel Source

SPIR Generator (e.g. patched Clang)

Alternative Language for Kernels

Apps and Frameworks

OpenCL C Runtime

Device X  Device Y  Device Z

SPIR (Standard Portable Intermediate Representation)
First portable IR that includes support for parallel computation
Created in close cooperation with LLVM community
SPIR 1.2 Released January 2014 (uses LLVM 3.2)

SPIR is easier compiler target than C

OpenCL run-time can consume SPIR

SYCL
Programming abstraction that combines portability and efficiency of OpenCL with ease of use and flexibility of C++
SYCL 1.0 Provisional Released March 2014

SYCL

https://github.com/KhronosGroup/SPIR

SPIR is easier compiler target than C

SYCL

Programming abstraction that combines portability and efficiency of OpenCL with ease of use and flexibility of C++
SYCL 1.0 Provisional Released March 2014
SPIR Unleashes Language Innovation

• **Front-ends**
  - New language front-ends and programming abstractions for heterogeneous parallel programming target production quality OpenCL backends through SPIR

• **Back-ends**
  - New target platforms based on multicore, vector, VLIW or other technologies can reuse production quality language frontends and abstractions

• **Tooling**
  - Advanced program analysis and optimization of programs in SPIR form

• **E.g. OpenACC, C++ AMP and Python are targeting SPIR to access optimized back-ends across multiple vendors**

Front-end Languages and Frameworks  
Multiple Hardware Architectures Backends  
Tools
SPIR 2.0 Released August 2014

- Full support of OpenCL 2.0 “C” kernel language
  - Generic address space
    - Functions can be written without specifying a named address space for arguments
  - Device side kernel enqueue
    - Device kernels can enqueue kernels to the same device with no host interaction
  - C++11 atomics
    - Subset of C11 atomics and synchronization operations
  - Pipes
    - Memory objects that store data organized as a FIFO

- SPIR 2.0 uses LLVM 3.4 with restrictions and conventions
- SPIR 2.0 specification is provisional - requested developer feedback!
  - https://www.khronos.org/opencl/spir2_0_feedback_forum

If you can do it in OpenCL 2.0 - you can do it in SPIR 2.0!
Next Generation OpenGL Initiative

Ground up design of a modern 3D+compute API
Compatibility break with OpenGL
Unify GL and ES

Platform Diversity Increasing
Need for cross-platform standards increasing

After twenty two years - need ground up design of API for high-efficiency access to graphics and compute on modern GPUs
Key Design Decisions

- Explicit application control over GPU and CPU workloads
  - Application is expected to tell the driver what it wants
  - High performance and predictability

- Multithreading / multicore friendly API
  - Greatly reduced CPU overhead

- Common shading language intermediate representation
  - Allows improved shader reliability and portability
  - Good target for machine-generated shaders and specialized HLLs
  - Provides a measure of IP protection for shaders
  - Can use common compiler front end across multiple platforms

- Architecture Neutral
  - Full support for tile-based as well as direct renderers
  - Improved reliability and consistency between implementations
Cross Platform Challenge

One family of GPUs  One OS  One GPU on one OS  All Modern Platforms and GPUs

Participation of key players
Proven IP Framework
Battle-tested cooperative model
The drive to not let the 3D industry fragment
Portability

Streamlined API is easier to implement and test

Cross-vendor Portability

Standard intermediate language improves shader program portability and reduces driver complexity

Enhanced conformance testing methodology

WebGL 1.0.2 doubles conformance tests over 1.0.1
~21200 vs. ~8900
1.0.3 suite will contain ~20% more tests
Most contributed by community
Next Generation OpenGL Initiative

- Fast-paced work on detailed proposals and designs are already underway
- NOT a multi-year design-by-committee process
- Goal to unify OpenGL and OpenGL ES - hence one BOF!
- If this new API relevant to your business - PLEASE JOIN!
Who’s On Board?

- Participants come from all segments of the graphics industry
  - Including an unprecedented level of participation from AAA game engine ISVs

- Call for participation!
  - Any interested company is strongly encouraged to join Khronos for a voice and a vote in the development process: [http://www.khronos.org/members/join](http://www.khronos.org/members/join)