

# **OpenCL BOF**

**Siggraph 2012**

**Neil Trevett, NVIDIA**

**OpenCL Chair**

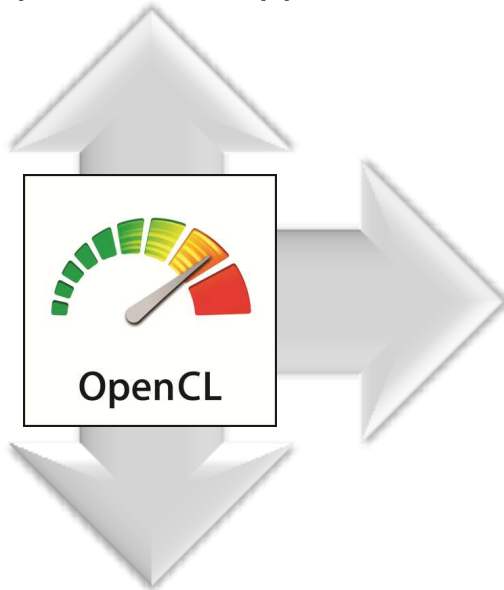
# Welcome!

- **AMD - Mike Houston**
  - OpenCL 1.2 Intro
  - OpenCL and AMD
- **Intel - Adam Lake**
  - CLU – OpenCL Utility Library
  - Intel and OpenCL
- **Adobe - Eric Berdahl, Dave McGavran and Sarah Kong**
  - Using OpenCL
- **Q&A**

# OpenCL Roadmap

## OpenCL-HLM (High Level Model)

Exploring high-level programming model, unifying host and device execution environments through language syntax for increased usability and broader optimization opportunities



## Long-term Core Roadmap

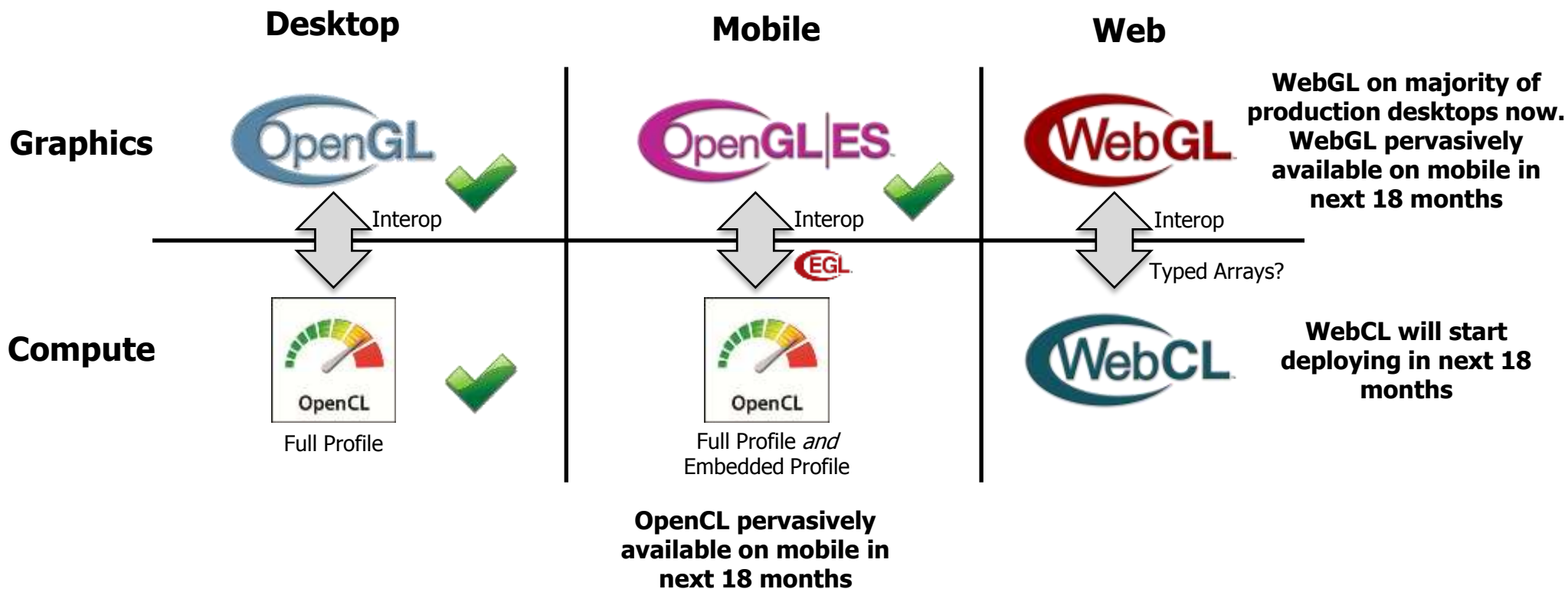
Exploring enhanced memory and execution model flexibility to catalyze and expose emerging hardware capabilities

## OpenCL-SPIR (Standard Parallel Intermediate Representation)

Exploring LLVM-based, low-level Intermediate Representation for code obfuscation/security and to provide target back-end for alternative high-level languages

**OpenCL-SPIR Provisional Specification to be released soon for feedback from LLVM community**

# Expanding Platform Reach for Graphics and Computation



# OpenCL and OpenGL Compute Shaders

- **OpenGL compute shaders and OpenCL support distinctly different use cases**
  - OpenCL provides a significantly more powerful and complete compute solution

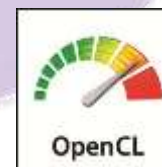
1. Fine grain compute operations inside OpenGL
2. GLSL Shading Language
3. Execute on single GPU only

1. Full ANSI C programming of heterogeneous CPUs and GPUs
2. Utilize multiple processors
3. Coarse grain, buffer-level interop with OpenGL

Enhanced 3D  
Graphics apps  
"Shaders++"

Imaging  
Video  
Physics  
AI

Pure compute  
apps touching  
no pixels





# Over to Mike..