The State of Gaming APIs

Neil Trevett
Vice President Mobile Content, NVIDIA
President, The Khronos Group

State of Gaming APIs–the Role of Khronos

High-end graphics technology is created on high-end platforms

New generation games will embrace mobility's strengths – not just treat phones as small consoles e.g. augmented reality

Mobile is the new platform for games innovation. Mobile APIs are needed to unlock hardware potential while conserving battery life

As platforms diversify – mobile, TV, embedded – HTML5 will become increasingly important as a universal app platform

Khronos - Connecting Software to Silicon

- **Creating open, royalty-free API standards**
  - Focus on graphics, dynamic media, compute and sensor hardware
- **Low-level - just above raw silicon**
  - "Foundation" functionality needed on every platform
- **Safe forum for industry cooperation**
  - 'By the industry for the industry'
  - Open to any company to join
  - IP framework to protect members and industry

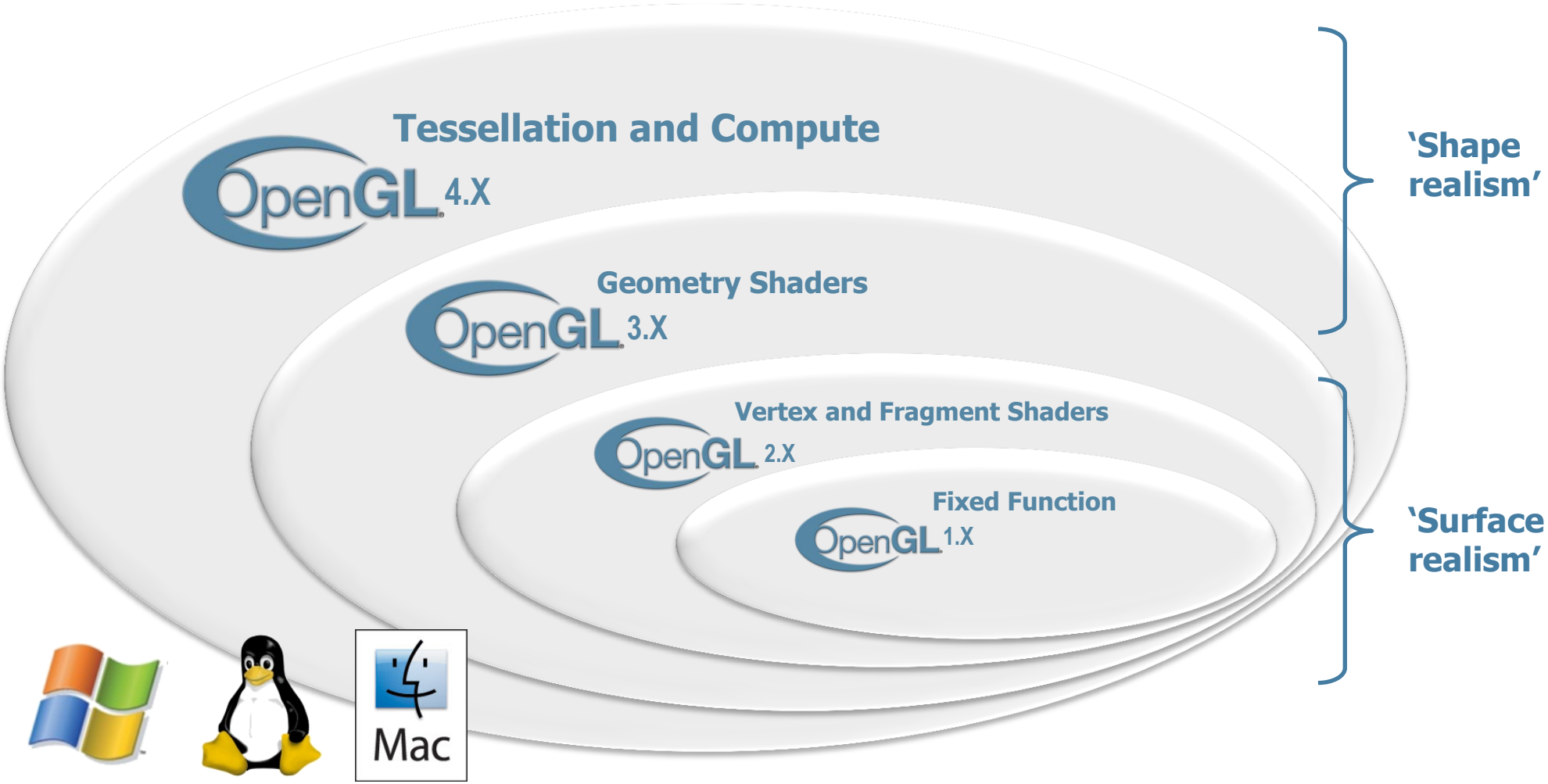
3D Evolution on PCs

'Doom' on a PC – 1993
id Software

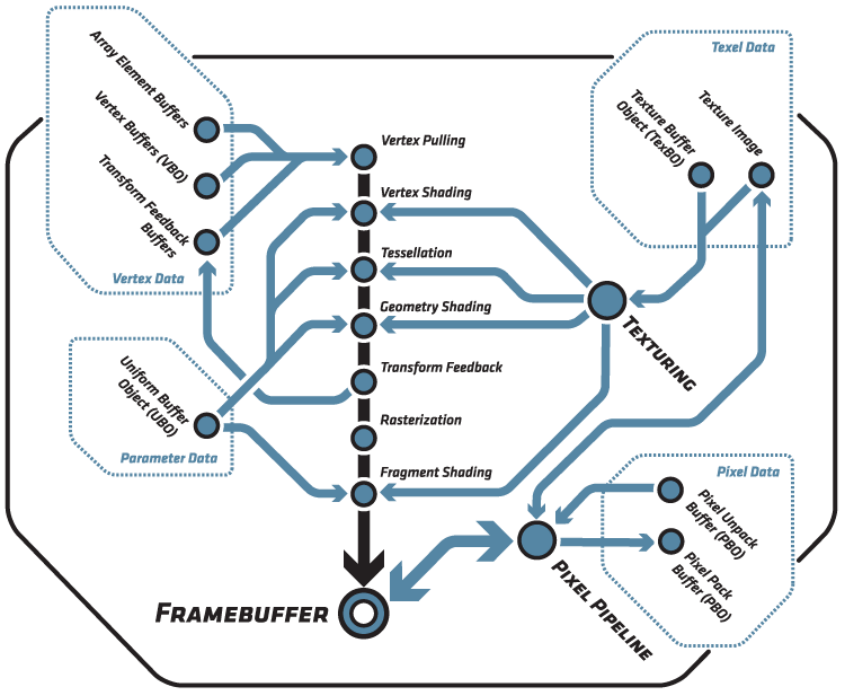
'Samaritan' Real-time Demo on a PC – 2011
Epic Unreal Engine

http://www.youtube.com/watch?v=RSXyztq_0uM

OpenGL for Each Hardware Generation



Accelerating OpenGL Innovation



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

OpenGL 4.2

OpenGL 4.1

OpenGL 3.3/4.0

OpenGL 3.2

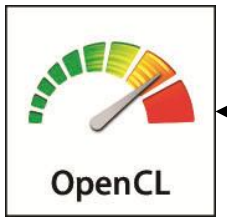
OpenGL 3.1

OpenGL 2.0 | OpenGL 2.1 | OpenGL 3.0

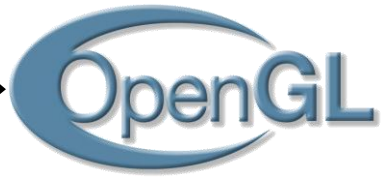
2004 | 2005 | 2006 | 2007 | 2008 | 2009 | 2010 | 2011

DirectX 9.0c | DirectX 10.0 | DirectX 10.1 | DirectX 11

Visual Computing Ecosystem



High performance compute and graphics interop – buffer and events



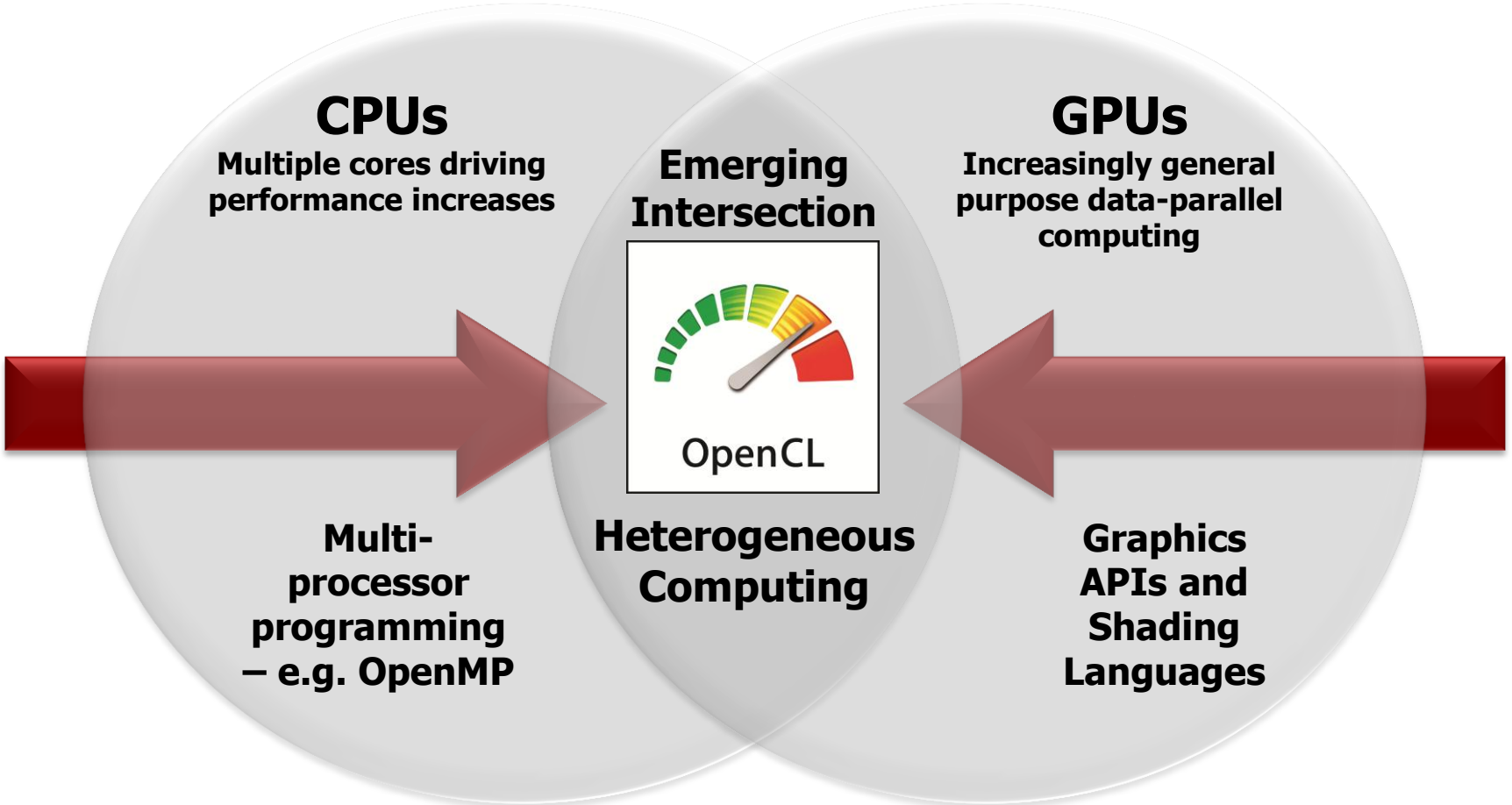
Heterogeneous Parallel Computation for physics image processing etc.

Leading-edge functionality developed on high-end systems

An API Ecosystem needs more than a list of APIs – it needs those APIs to efficiently interoperate

KHRONOS GROUP

Processor Parallelism



OpenCL is a programming framework for heterogeneous compute resources

The BIG Idea behind OpenCL

- **OpenCL execution model …**
  - Define N-dimensional computation domain
  - Execute a kernel at each point in computation domain
- **C Derivative to write kernels – based on ISO C99**
  - APIs to discover devices in a system and distribute work to them
- **Targeting many types of device**
  - GPUs, CPUs, DSPs, embedded systems, mobile phones.. Even FPGAs

Traditional loops

```
void
trad_mul(int n,
         const float *a,
         const float *b,
         float *c)
{
  int i;
  for (i=0; i<n; i++)
    c[i] = a[i] * b[i];       }
```

Data Parallel OpenCL

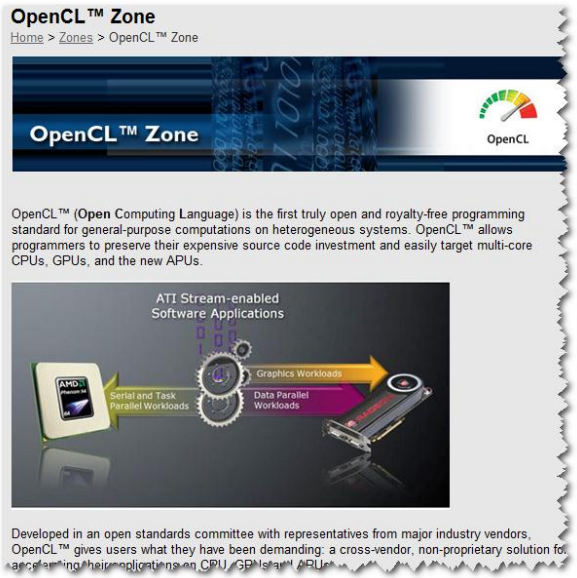
```
kernel void
dp_mul(global const float *a,
       global const float *b,
       global float *c)
{
  int id = get_global_id(0);

  c[id] = a[id] * b[id];

} // execute over "n" work-items
```

OpenCL Working Group Members

- **Diverse industry participation – many industry experts**
  - Processor vendors, system OEMs, middleware vendors, application developers
  - Academia and research labs, FPGA vendors
- **NVIDIA is chair, Apple is specification editor**

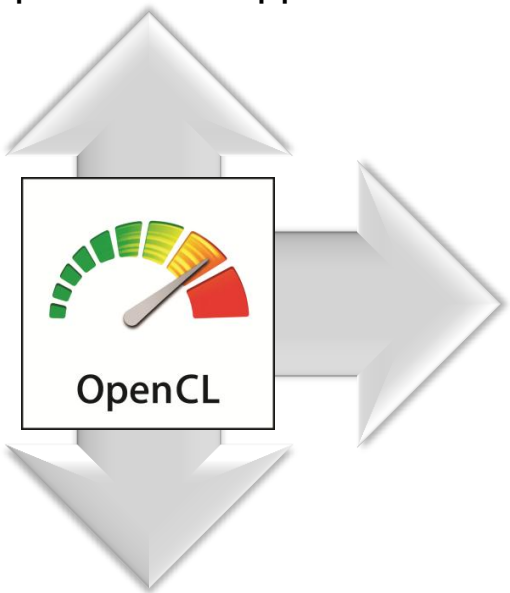
OpenCL Desktop Implementations

- http://developer.amd.com/zones/OpenCLZone/
- http://software.intel.com/en-us/articles/opencl-sdk/
- http://developer.nvidia.com/opencl



Looking Forward

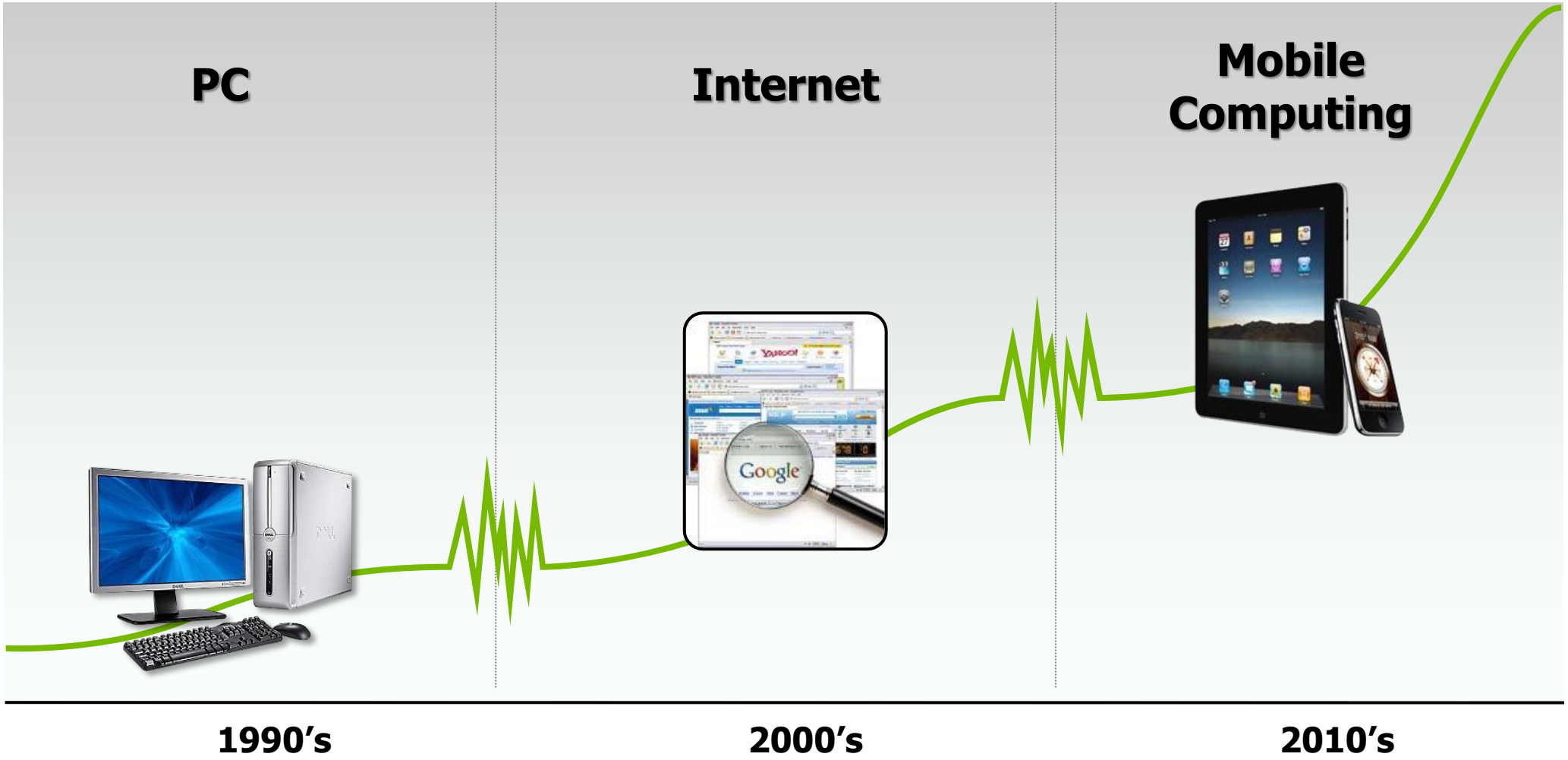
OpenCL-HLM
Exploring high-level programming model, unifying host and device execution environments through language syntax for increased usability and broader optimization opportunities



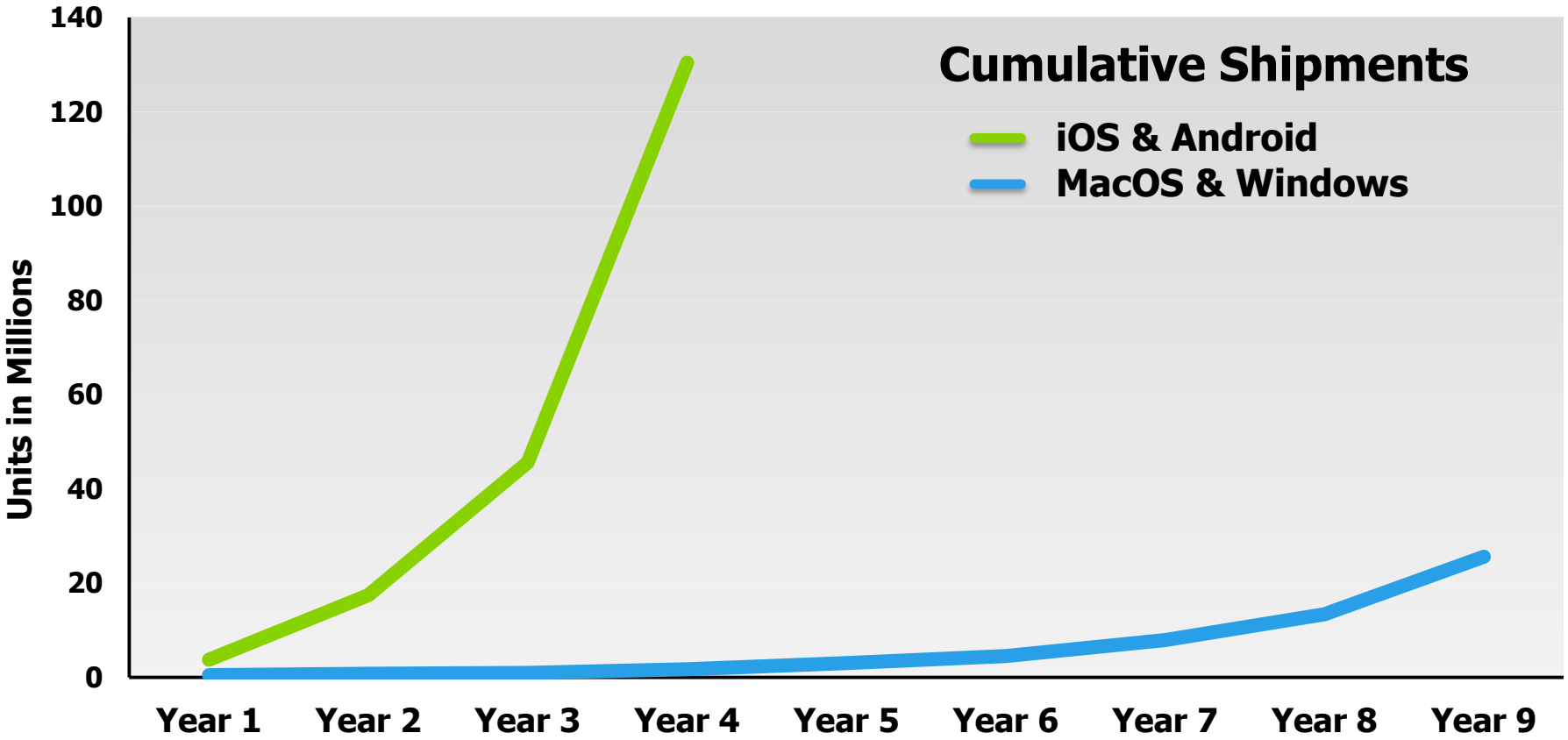
Long-term Core Roadmap
Exploring enhanced memory and execution model flexibility to catalyze and expose emerging hardware capabilities

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

A New Era in Personal Computing

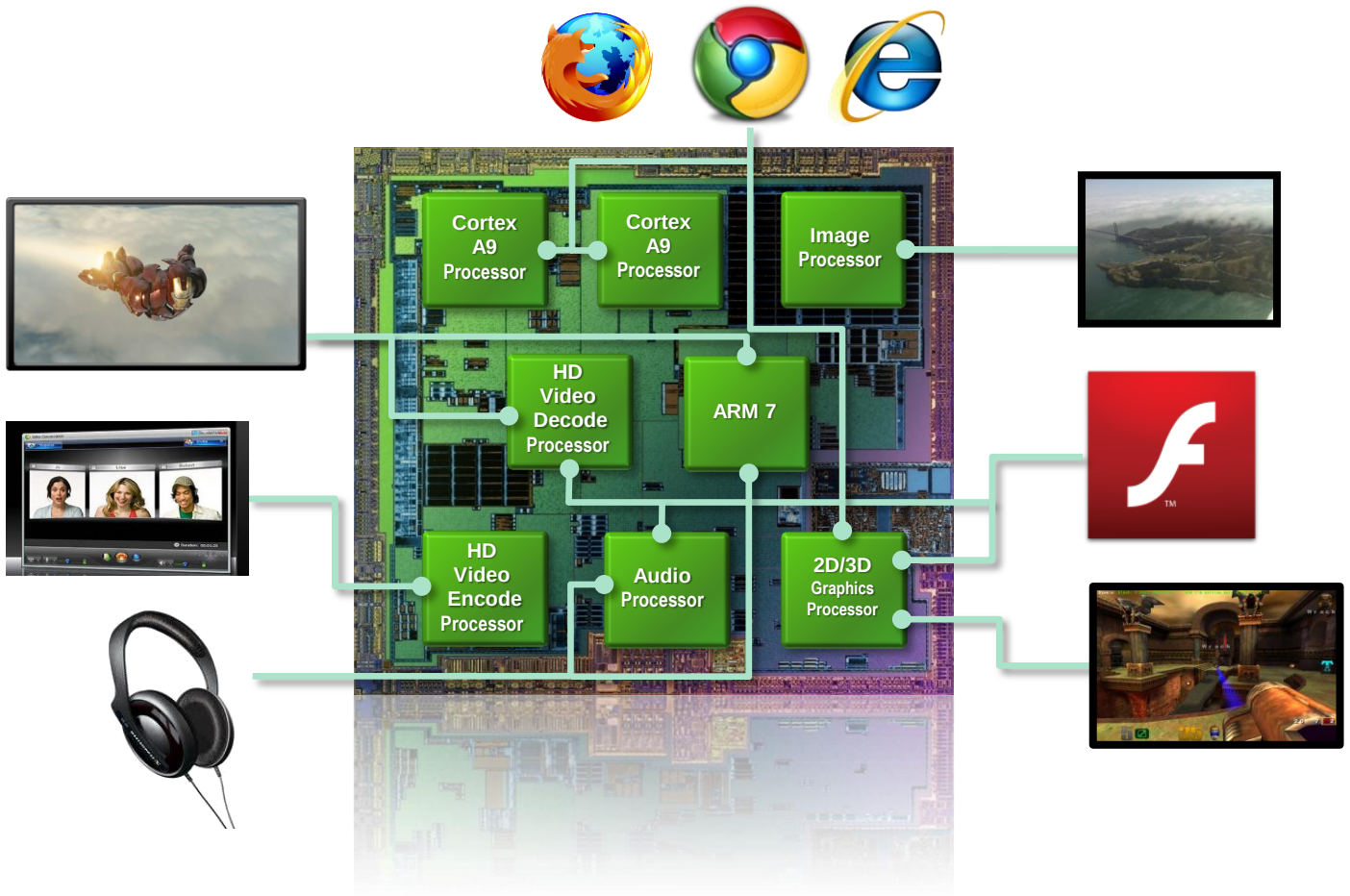


20 Years Faster to 100M Per Year

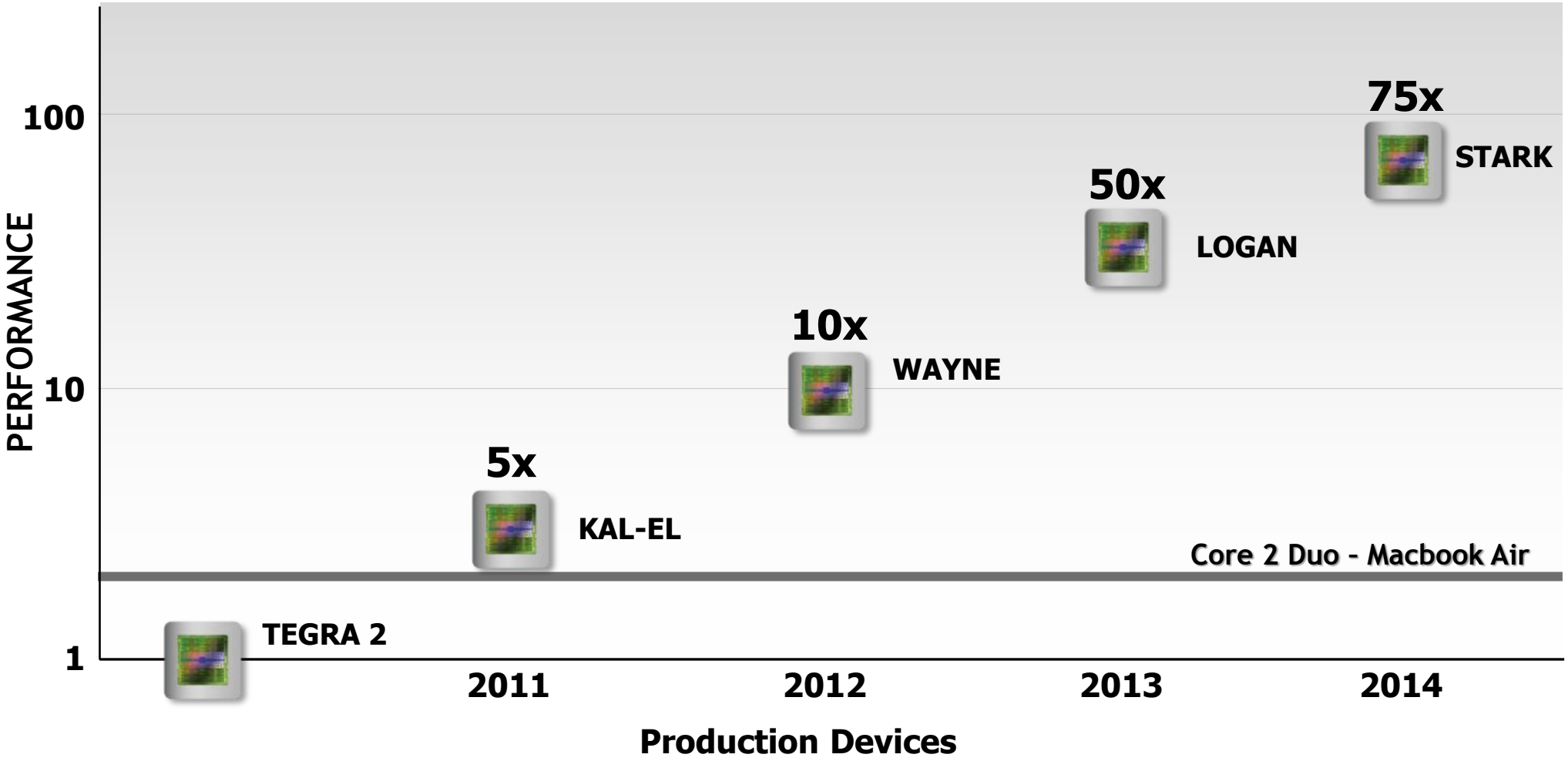


Source: Gartner, Apple, NVIDIA

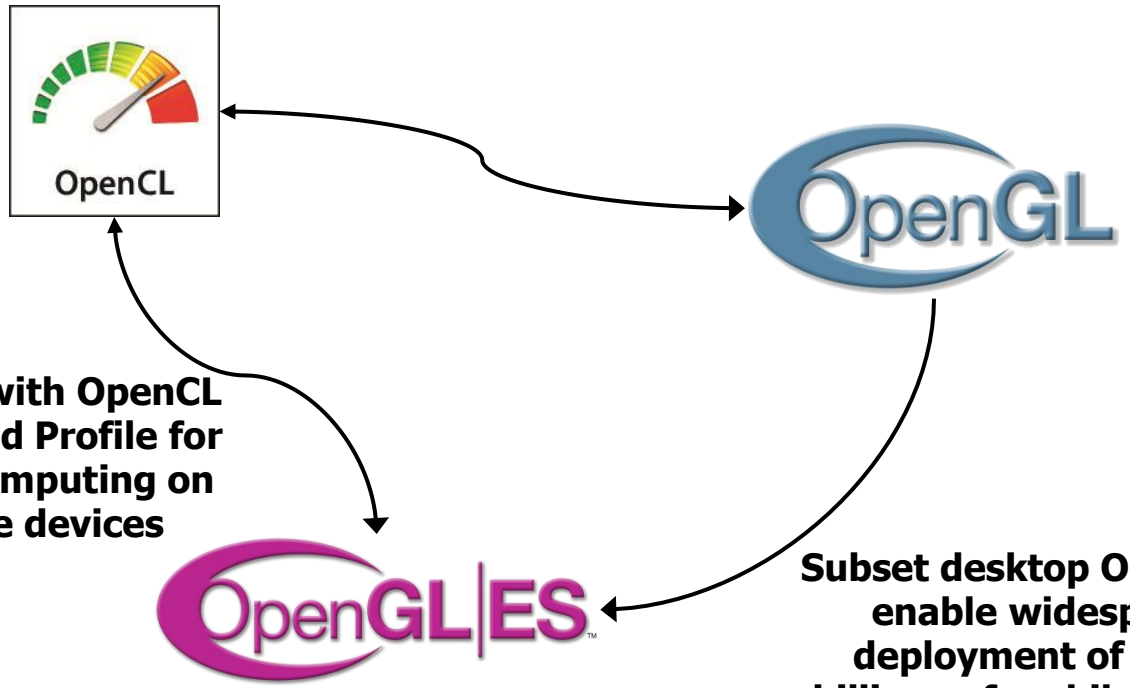
Mobile Silicon Experiential Processing



Mobile Roadmap Acceleration



Visual Computing Ecosystem

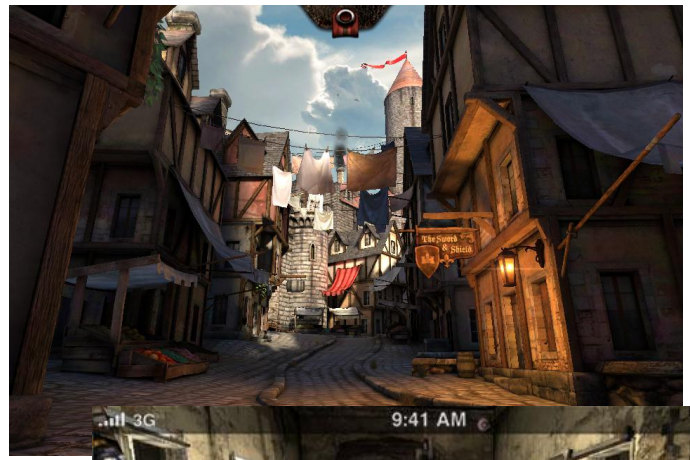


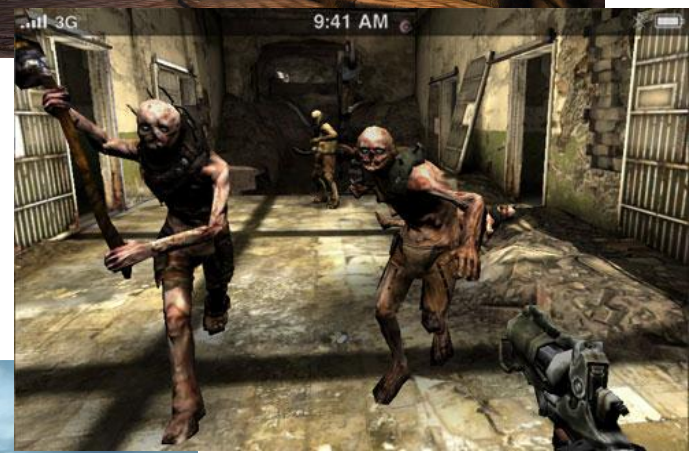
Interop with OpenCL Embedded Profile for visual computing on mobile devices

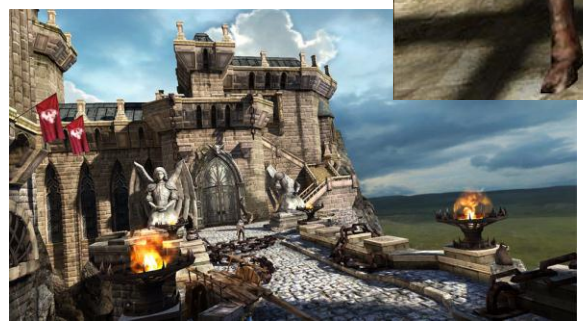
Subset desktop OpenGL to enable widespread deployment of 3D on billions of mobile devices

What is OpenGL ES?

- **OpenGL for embedded and mobile devices**
  - Eliminates redundant and legacy features
  - Adds extensions to make it mobile-friendly
- **The dominant 3D API for mobile devices**
  - Widely adopted for STB, DTV, automotive,…
  - Hundreds and hundreds of millions shipped
- **Runs high-end content and engines**
  - UE3, Unity, Unigine, Rage

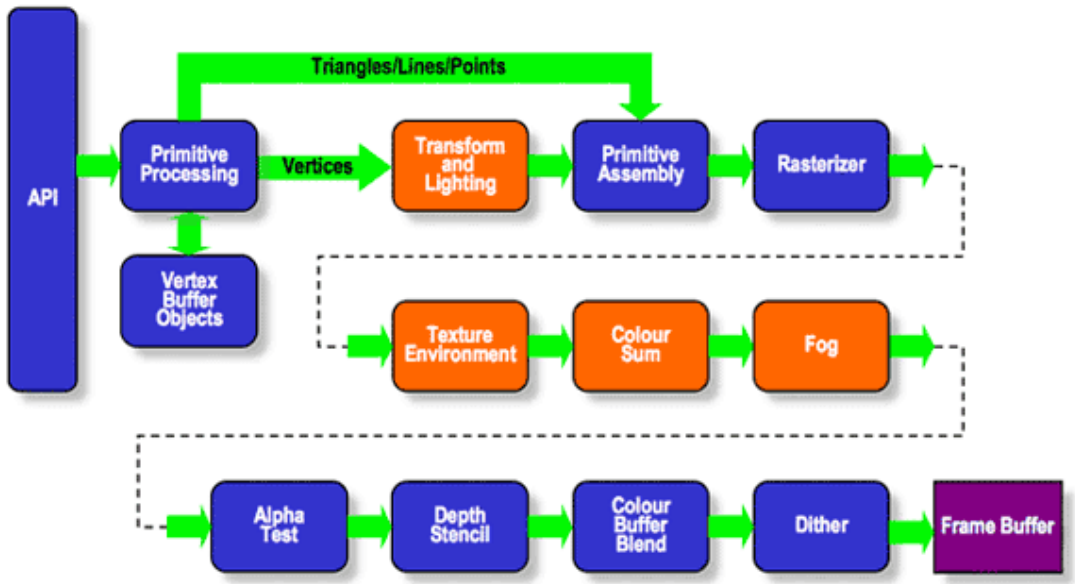






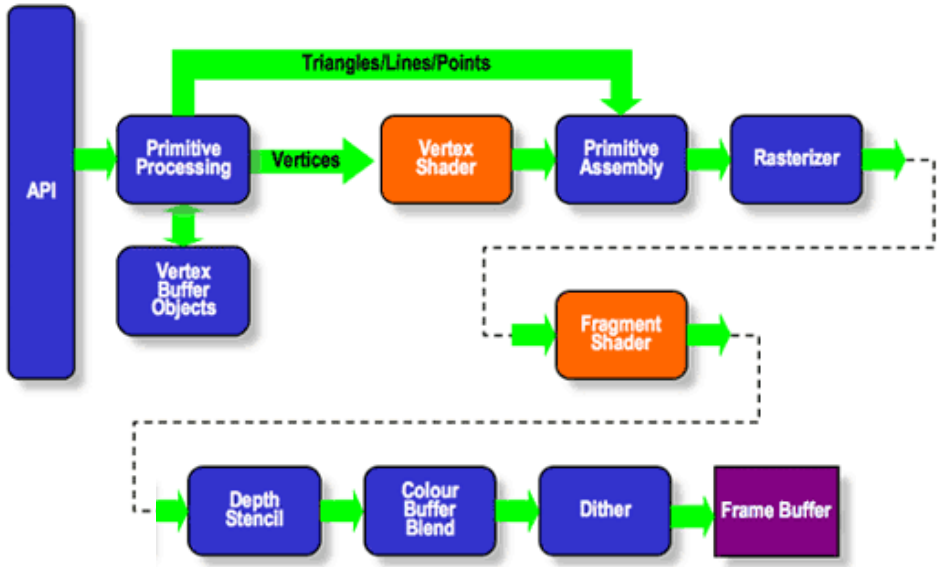
OpenGL ES Pipelines

OpenGL ES 1.x Fixed Function Pipeline



Based on OpenGL 1.5
Vertex Arrays / Buffer Objects
Transform & Lighting
Multi-texturing (min 2 units)

OpenGL ES 2.0 Programmable Pipeline



Based on OpenGL 2.0
Removes fixed function pipeline
High level language (GLSL ES)
Super-compact, efficient API

OpenSL ES – Advanced Audio

- **OpenSL ES does for audio what OpenGL ES does for graphics**
  - Advanced audio functionality from simple playback to 3D audio
- **Object-based native audio API for simplicity and high performance**
  - Reduces development time
- **Same API regardless of underlying implementation**
  - Software or hardware accelerated
- **Cross OS portability**
  - Preserves application investment

OpenSL ES™

OpenSL ES Profiles

Game-centric mobile devices
Advanced MIDI functionality, sophisticated audio capabilities such as 3D audio, audio effects, ability to handle buffers of audio, etc.

Music-centric mobile devices
High quality audio, ability to support multiple music audio codecs, audio streaming support

OpenSL ES game

OpenSL ES music

OpenSL ES phone · music · game

OpenSL ES phone

Basic mobile phones
Ring tone and alert tone playback (basic MIDI functionality), basic audio playback and record functionality, simple 2D audio games

A device can implement any combination of profiles

Visual-based Augmented Reality

Camera video stream
sent to the compositor

Camera
images used
to track the
camera's
location and
orientation

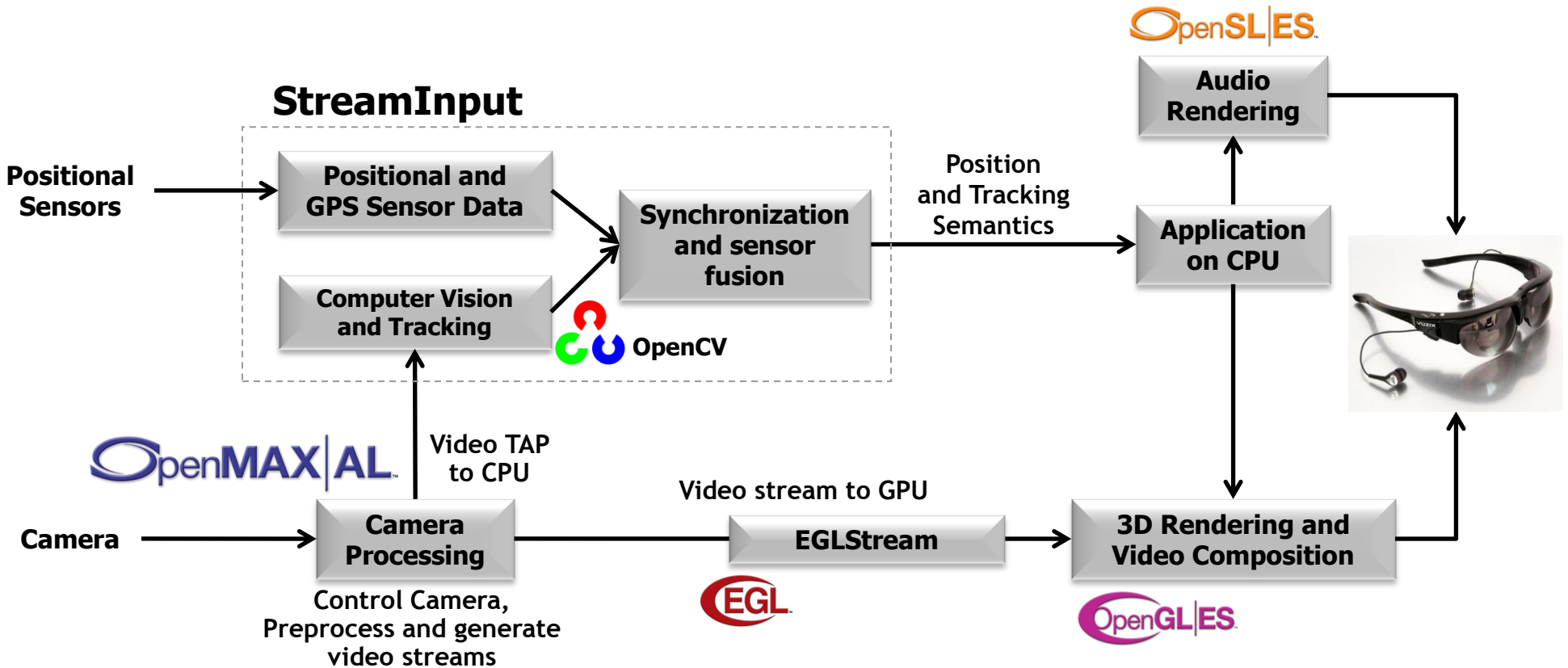
3D
augmentations
composited with
video stream

Camera
Tracking

Camera-to-scene transform locks
the 3D rendering to the real world

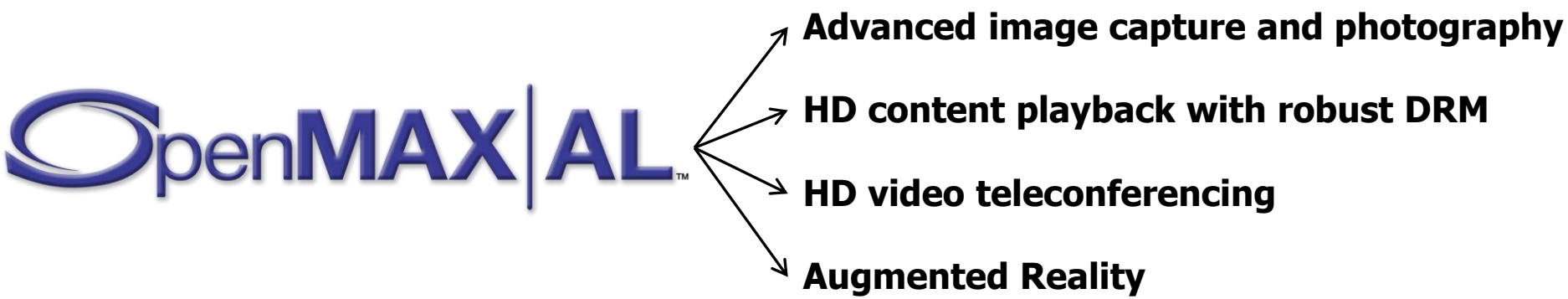
3D Augmentation
Rendering

Augmented Reality Functionality



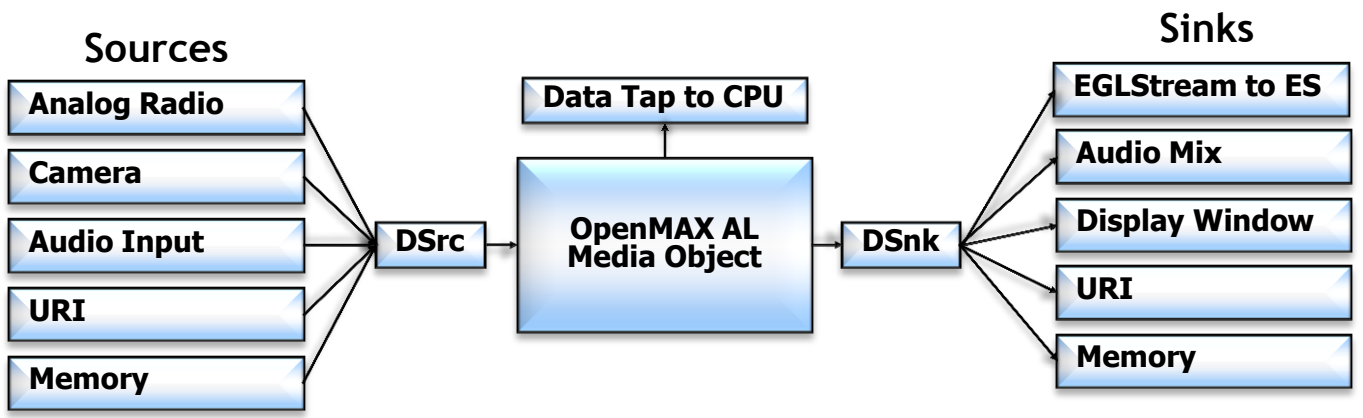
OpenMAX AL

- **Enables key image, camera and video use cases**
  - Allows optimal hardware acceleration with app portability
- **OpenMAX AL is designed as application API**
  - OpenMAX IL is for system integration – more flexibility, less portability



OpenMAX AL - Object Oriented Media

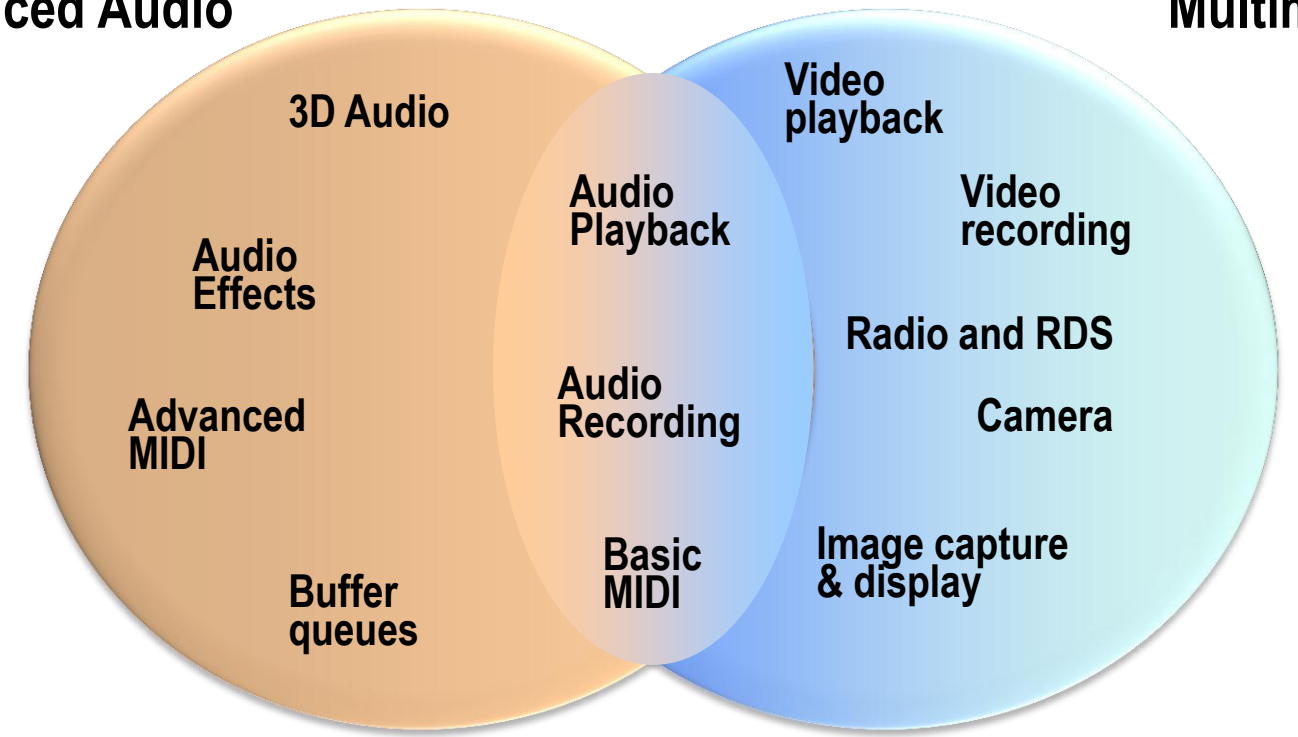
- **Create Media Objects to process images and video with AV sync**
  - Connect to variety of input and output objects to PLAY and RECORD media
- **Object control interfaces**
  - Sources: Mix control, Seek, Rate, Metadata Extraction, Camera Controls
  - Sinks: Encode control, Tuning, MIDI, Metadata Insertion
- **Extensions for data routing**
  - EGLStream - to send images to OpenGL ES on the GPU
  - Data Tap Object – to send images to CPU memory



OpenSL ES and OpenMAX AL

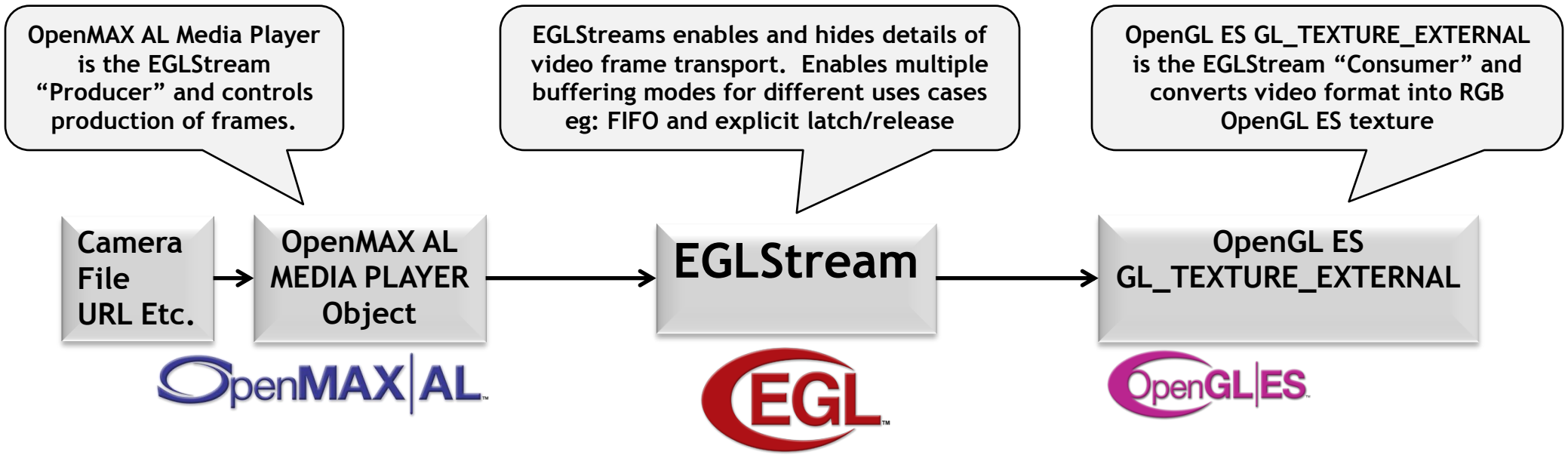
OpenSL ES
Advanced Audio

OpenMAX AL
Multimedia



Common Object-oriented API Framework

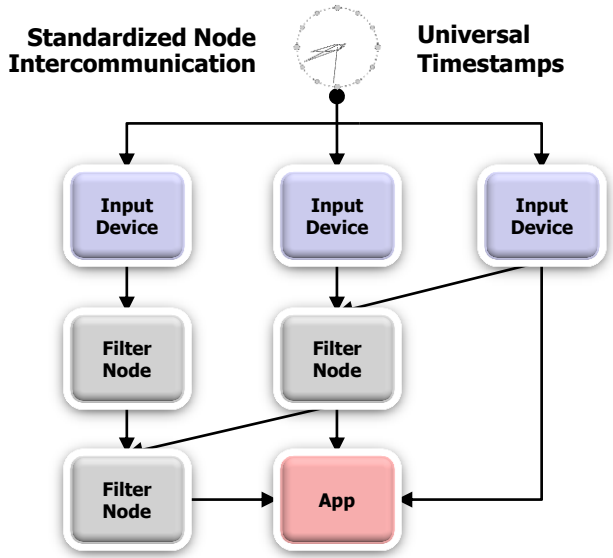
EGLStream – Video/Graphics Interop



StreamInput Connects Sensors to Apps

Apps request semantic sensor information
StreamInput defines list of possible semantic requests
"Am I in an elevator?" "Give me gestures and face position"

Advanced Sensors Everywhere
Standard cameras, depth cameras
motion and position, touch, microphones
wireless controllers



Apps Need Sophisticated Access to Sensor Data
Without coding to specific systems
or sensor hardware

Sensor graph created to provide sensor information
StreamInput defines graph creation API and node interconnects
Low-level sensor processing encapsulated in nodes – unleashes fusion innovation
Apps gain 'magical' situational awareness

Unique Aspects of StreamInput

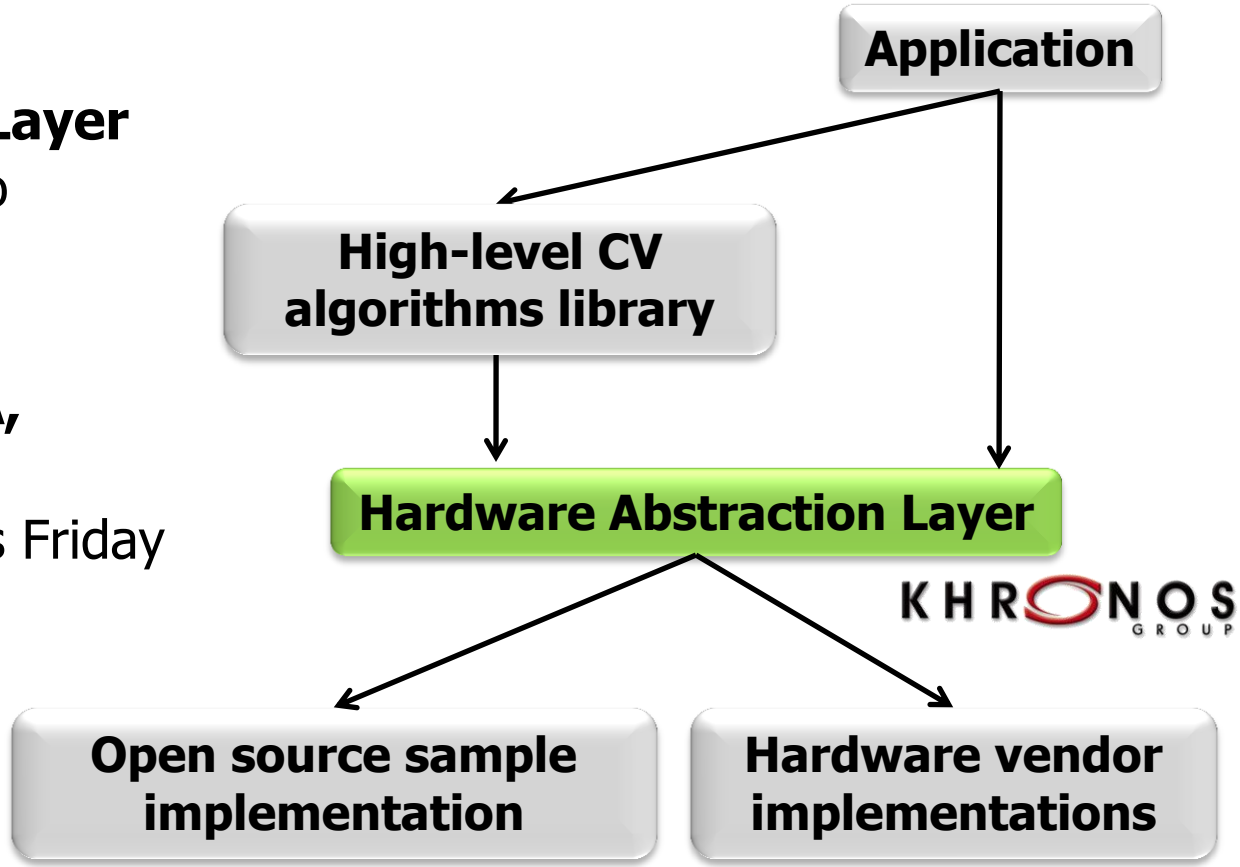
- **Defines semantic information flow - not low-level sensor processing**
  - Removing the barriers to sensor discovery and sensor code portability
- **Enables sensor/middleware vendors to innovate**
  - Sensor processing nodes can ship as firmware or middleware 'black boxes'
  - Nodes can be discovered and integrated into StreamInput graph
  - Nodes can fuse data from multiple sensors – increasing data quality
- **Covers broad range of sensor types**
  - Multi-axis motion/position, cameras, depth cameras, touch, audio, mechanical
  - Industry-leading experts from each category at the working group
- **Time-stamps for inter-sensor synchronization**
  - Camera frames, audio samples, sensor samples, display buffer switch times
  - App can detect and compensate for sensor/pipeline/rendering delays

Current StreamInput Participants

- **Aiming for production implementations in September 2012**

OpenCV as Potential Khronos Standard

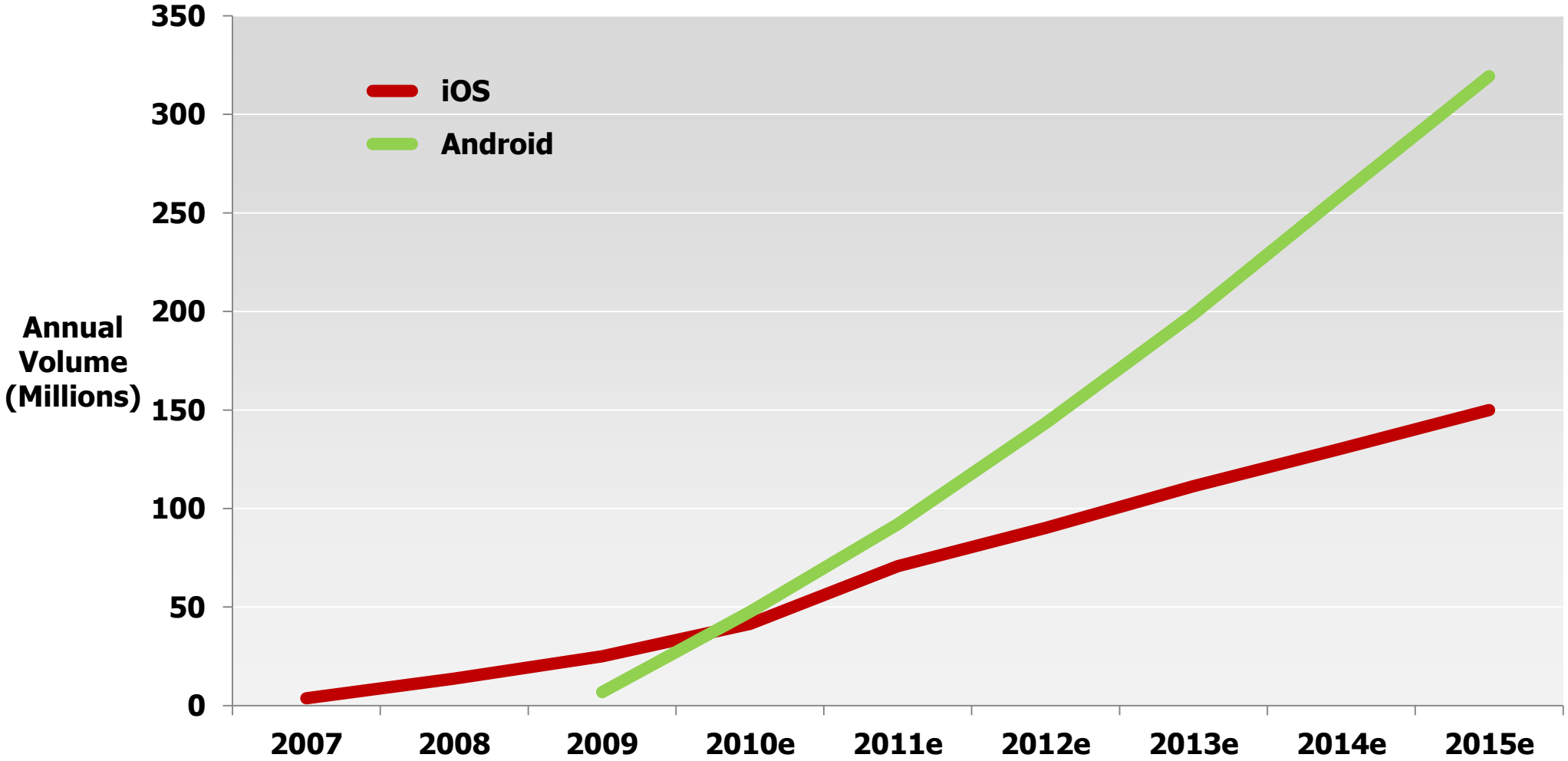
- **Maintain OpenCV's open source momentum**
- **Add Hardware Abstraction Layer**
  - Enables hardware vendors to provide accelerated imaging and vision modules
- **Being sponsored by NVIDIA, Itseez and Willow Garage**
  - Board decision to initiate this Friday



OpenCV

KHRONOS GROUP

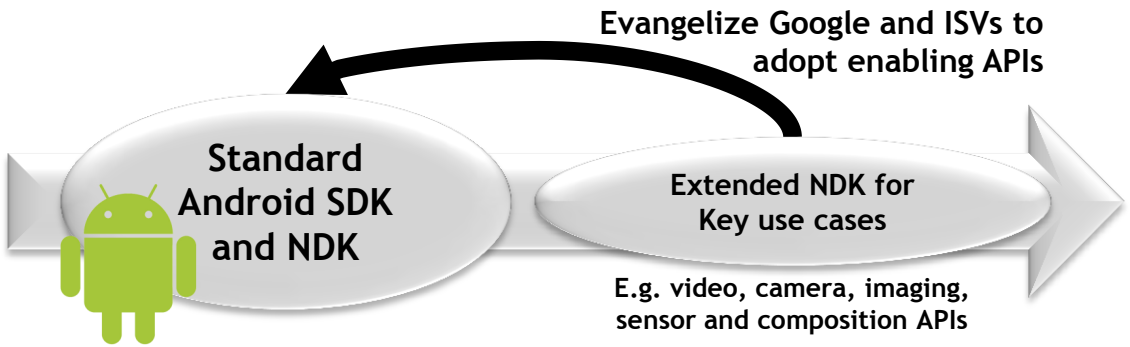
Mobile - Android Becoming Dominant OS



Source: Gartner, NVIDIA

Native APIs on Android

- **Khronos developing open cross vendor standards for key use cases**
  - Khronos APIs have strong momentum in the silicon community
- **Drive adoption - add these APIs to Android NDK**
  - Do not break/change existing Google APIs
  - Evangelize Google to possibly adopt into standard platform
- **Extended APIs can be used by:**
  - Bundled apps, Market apps with API selection
    Multiple apks behind multi-Apk SKU



Android Native API Adoption

OpenGL ES	OpenGL ES	OpenGL ES 2.0 Shipping - Android 2.2
OpenSL ES	OpenSL ES	OpenSL ES 1.0 Shipping – Android 2.3
OpenMAX AL	OpenMAX AL	OpenMAX AL 1.0 Shipping - Android 4.0
EGL	EGL	EGL 1.4 Shipping on SDK
OpenCL	OpenCL	Not yet adopted
StreamInput	StreamInput	Working group will ship in 2012
	OpenCV	Khronos voting to establish WG

Google Chrome OS

symbian

brew

bada

RIM

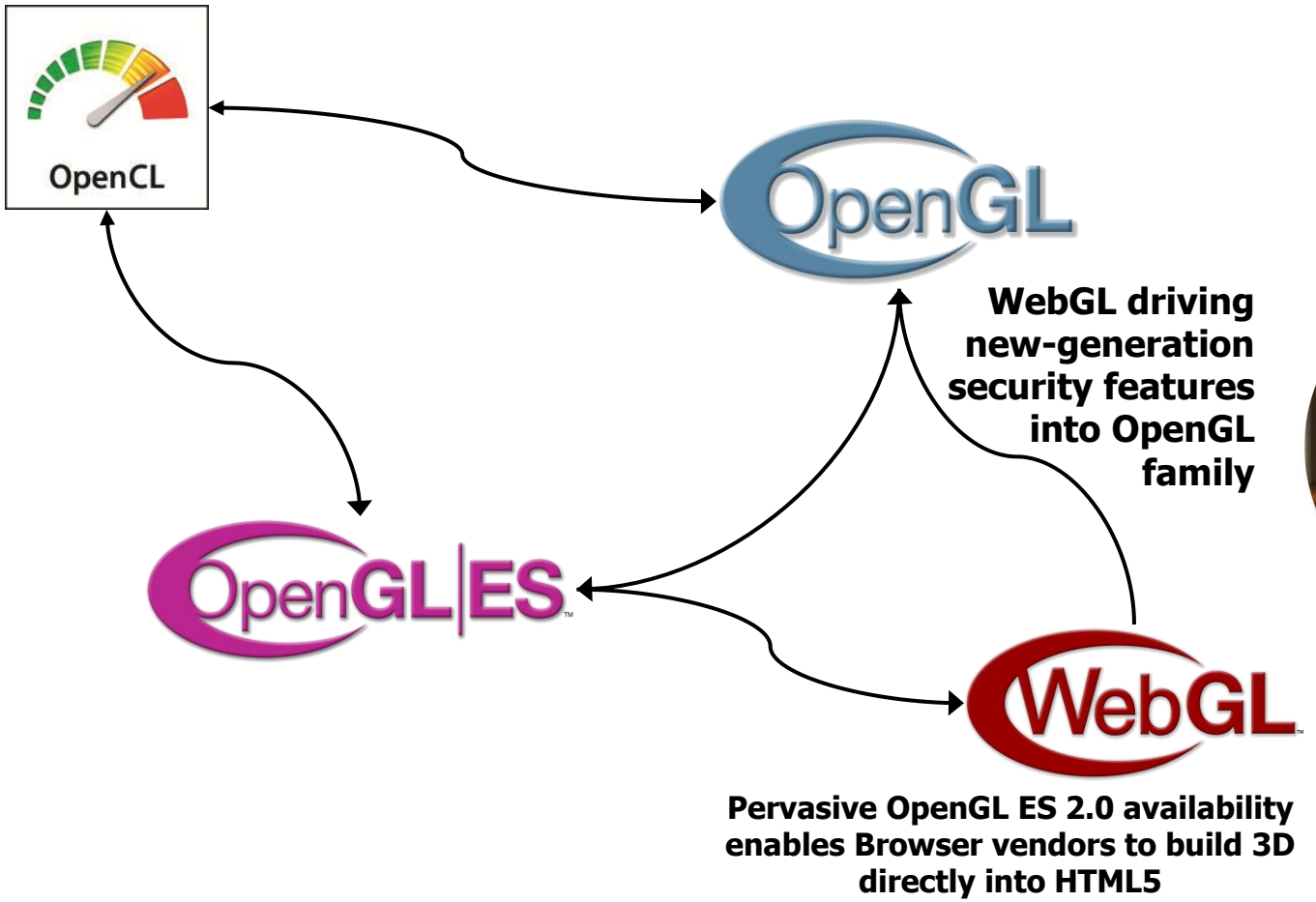
BlackBerry

?

Mobile Web versus Apps

- **Mobile Apps have functional and aesthetic appeal**
  - Beautiful, responsive, focused
- **HTML5 with Advanced 3D could provide the same level of "App Appeal"**
  - Highly interactive, rich visual design
- **Using HTML5 to create 'Web Apps' has many advantages**
  - Portable to any browser enabled system
  - Same code can run as app or as web page
  - Web app is searchable and discoverable through the web
  - Not a closed app store – no app store 'tax'

Visual Computing Ecosystem



WebGL driving new-generation security features into OpenGL family

Pervasive OpenGL ES 2.0 availability enables Browser vendors to build 3D directly into HTML5

WebGL – 3D on the Web – No Plug-in!

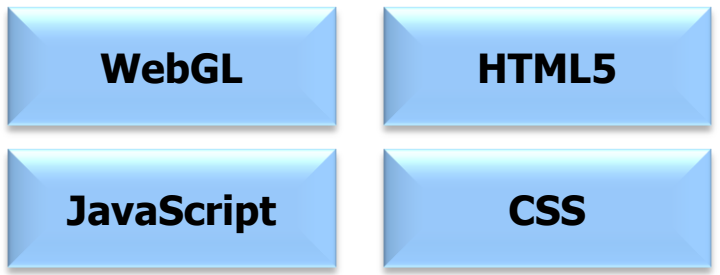
- **Historic opportunity to bring accelerated 3D graphics to the Web**
  - WebGL defines JavaScript binding to OpenGL ES 2.0
- **Leveraging HTML5 and uses <canvas> element**
  - Enables a 3D context for the canvas
- **JavaScript is easily fast enough now for visual computing**
  - Plus OpenGL ES 2.0 enables local geometry caching and GPGPU computation

Being defined by major browsers and GPU vendors working together

WebGL Implementation Anatomy

Content downloaded from the Web. Middleware can make WebGL accessible to non-expert 3D programmers

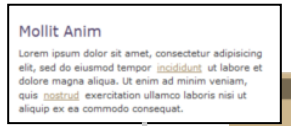
Browser provides WebGL functionality alongside other HTML5 specs - no plug-in required



OS Provided Drivers. WebGL on Windows can use Google Angle to create conformant OpenGL ES 2.0 over DX9

HTML5 Content Architecture

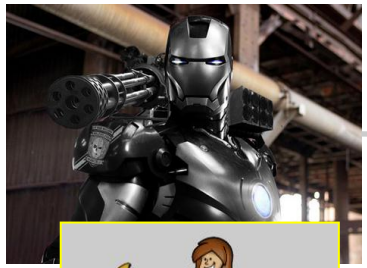
HTML content generated by layout engine 'on page'



All content passes through CSS layout

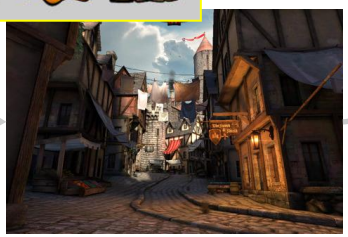
CSS Layout and Transforms

<video> tag



<canvas> tag

JavaScript drives interactivity for 2D and 3D graphics



Composition of off-screen buffers

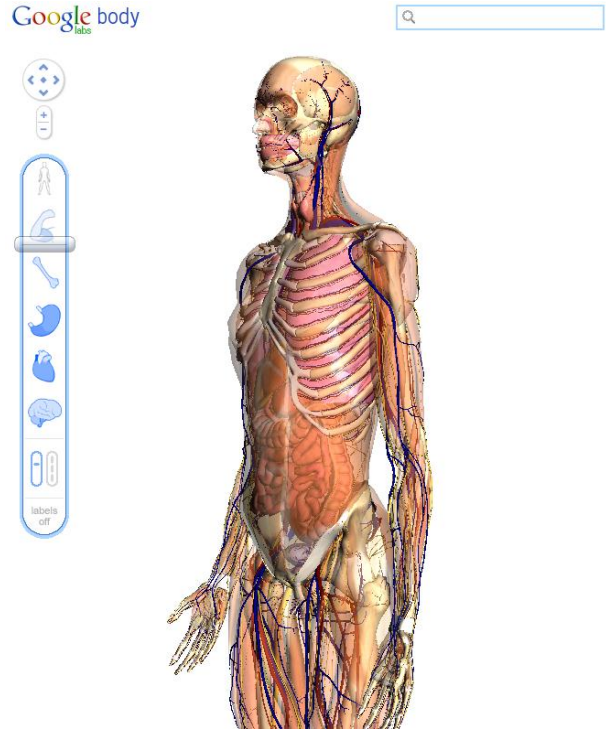
Composition needs to be GPU accelerated

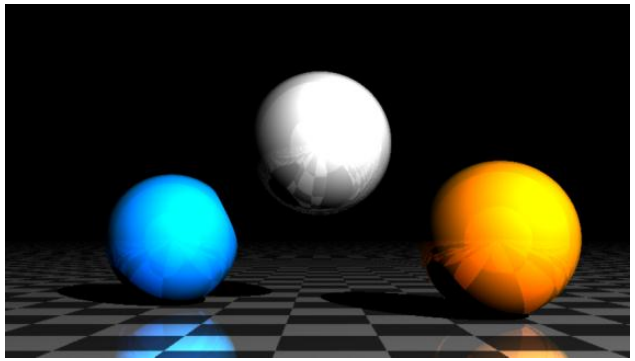
Video, Vector Graphics and 3D created off-screen buffers

WebGL

WebGL and HTML Interaction

- **3D is not trapped in a rectangular window**
  - 3D can overlay and underlay HTML content
  - Easy to make HUDs or user interfaces
- **Strong ties with other advanced HTML5**
  - WebGL can use HTML5 <video>
    or canvas as a texture
- **Can use 3D for core Web UI – as well as content**
  - Advanced transforms and special effects
- **Render HTML DOM sub-tree as texture**
  - Support user interaction when in 3D
  - Mozilla and Google prototyping as extension
- **WebGL is democratization of 3D**
  - Accessible, pervasive, enabling
  - Spawning amazing innovation





WebGL Deployment

- **WebGL 1.0 Released at GDC March 2011**
  - Mozilla, Apple, Google and Opera working closely with GPU vendors
- **Typed array 1.0 spec ratified by Khronos in May**
  - Supporting bulk data transfer between threads (workers)
  - Many use cases - background mesh loading, generation, deformation, physics ...
- **1.0.1 release of WebGL spec and conformance suite imminent**
  - 100% robust stance on security
  - Fixing bugs in 1.0.0 conformance suite
  - Implementations will report getContext("webgl") (not experimental)

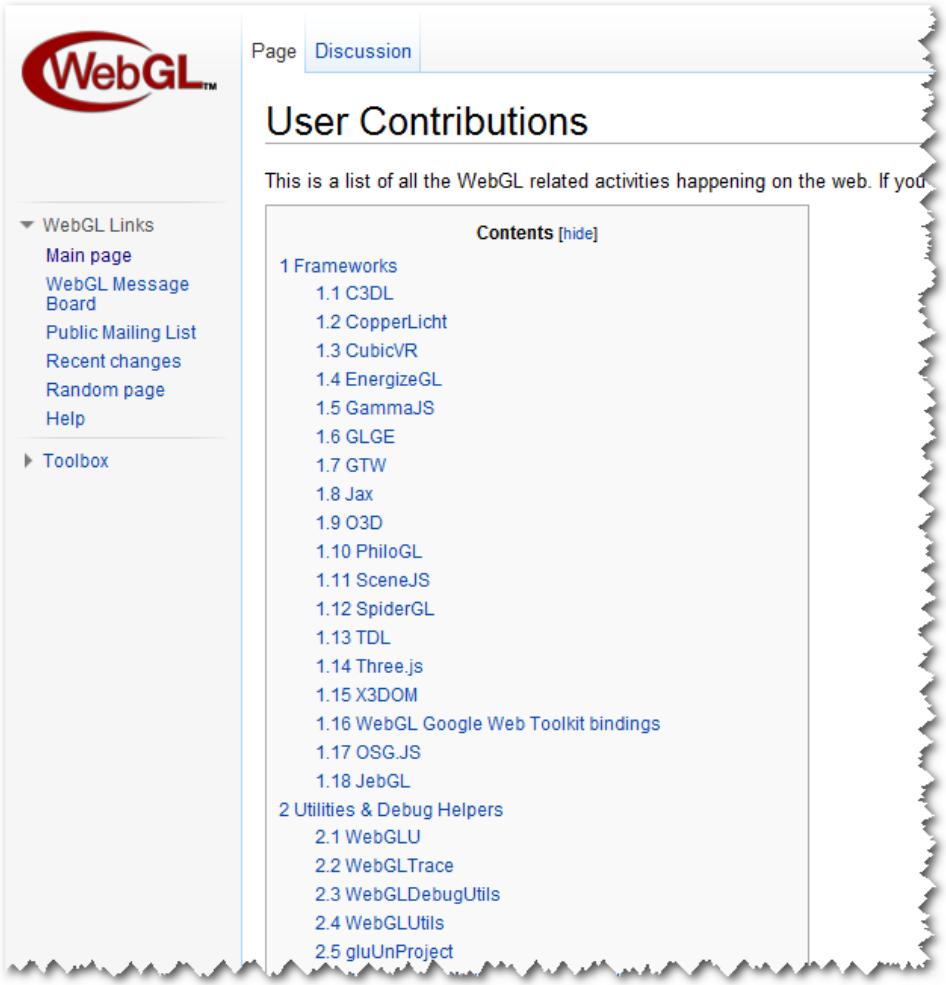
Show all versions	IE	Firefox	Safari	Chrome	Opera
3 versions back	6.0	3.6	3.2	10.0	10.6
2 versions back	7.0	4.0	4.0	11.0	11.0
Previous version	8.0	5.0	5.0	12.0	11.1
Current	9.0	6.0	5.1	13.0	11.5
Near future		7.0		14.0	12.0
Farther future	10.0	8.0	6.0	15.0	12.1

WebGL is not enabled by default in Safari

http://caniuse.com/#search=webgl

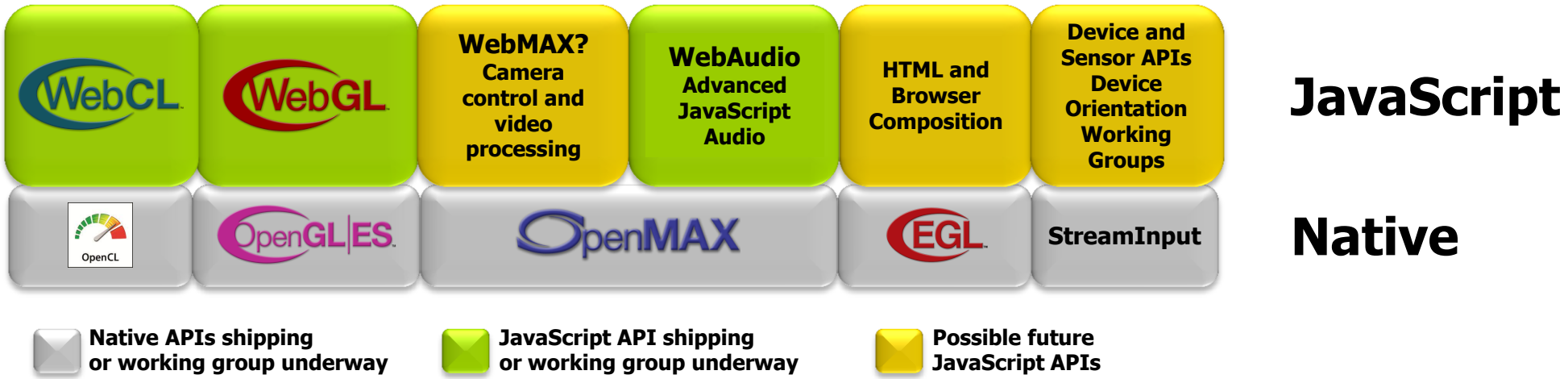
Frameworks and Tools

- **WebGL is deliberately low level to enable the full power and flexibility of OpenGL ES 2.0**
- **If you are not an expert 3D programmer – don't panic!**
- **WebGL is perfect foundational layer for JavaScript middleware frameworks**
- **Lots of utilities and tools already appearing**



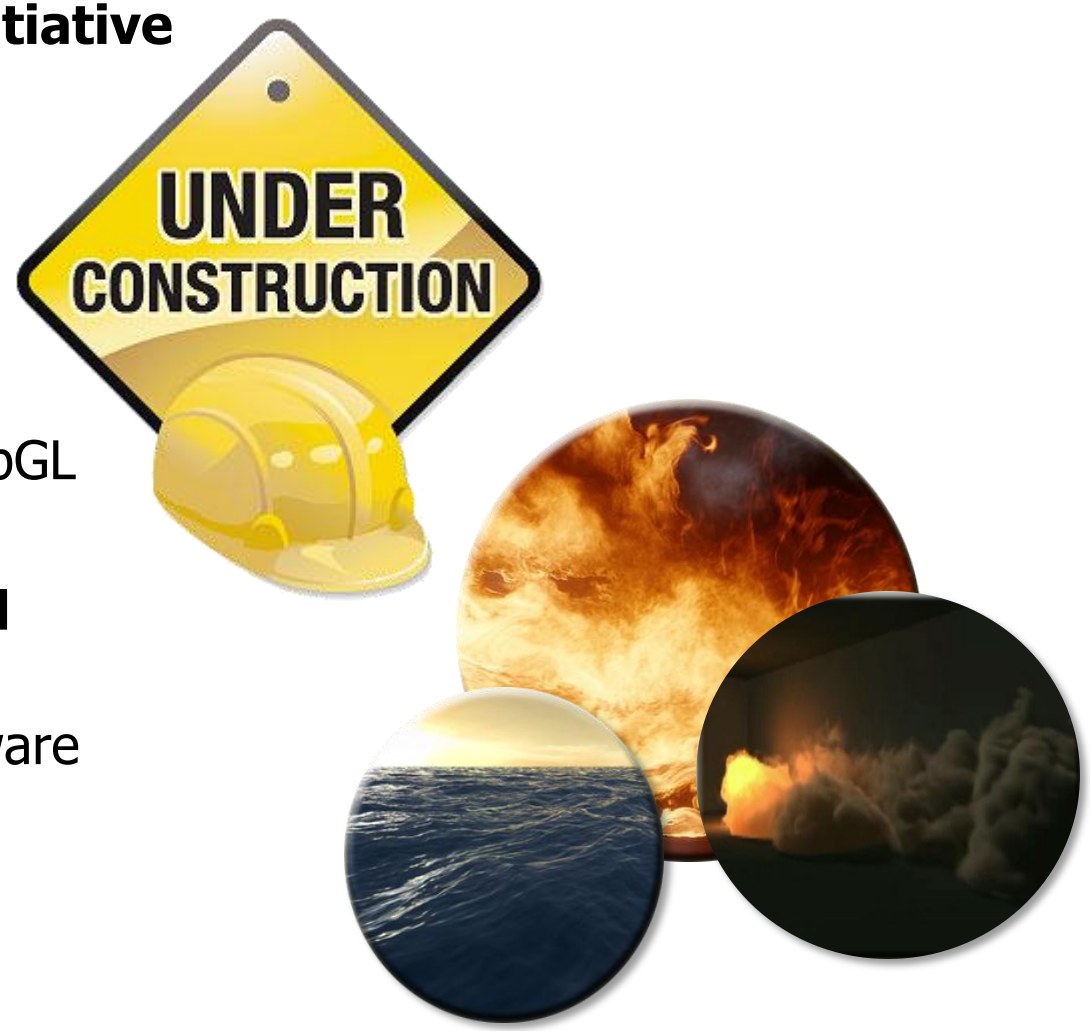
Leveraging Native API Investment into HTML5

- **HTML5 evolving into cross-platform programming platform**
  - Gradually exposing complete system capabilities
- **Opportunity to synergize Web and native APIs development**
  - Leverage native API investments, reduce developer learning cycles
- **Khronos and W3C creating close liaison**



WebCL – Parallel Computing for the Web

- **Khronos launching new WebCL initiative**
  - First announced in March 2011
  - API definition already underway
- **JavaScript binding to OpenCL**
  - Security is top priority
- **Many use cases**
  - Physics engines to complement WebGL
  - Image and video editing in browser
- **Stay close to the OpenCL standard**
  - Maximum flexibility
  - Foundation for higher-level middleware

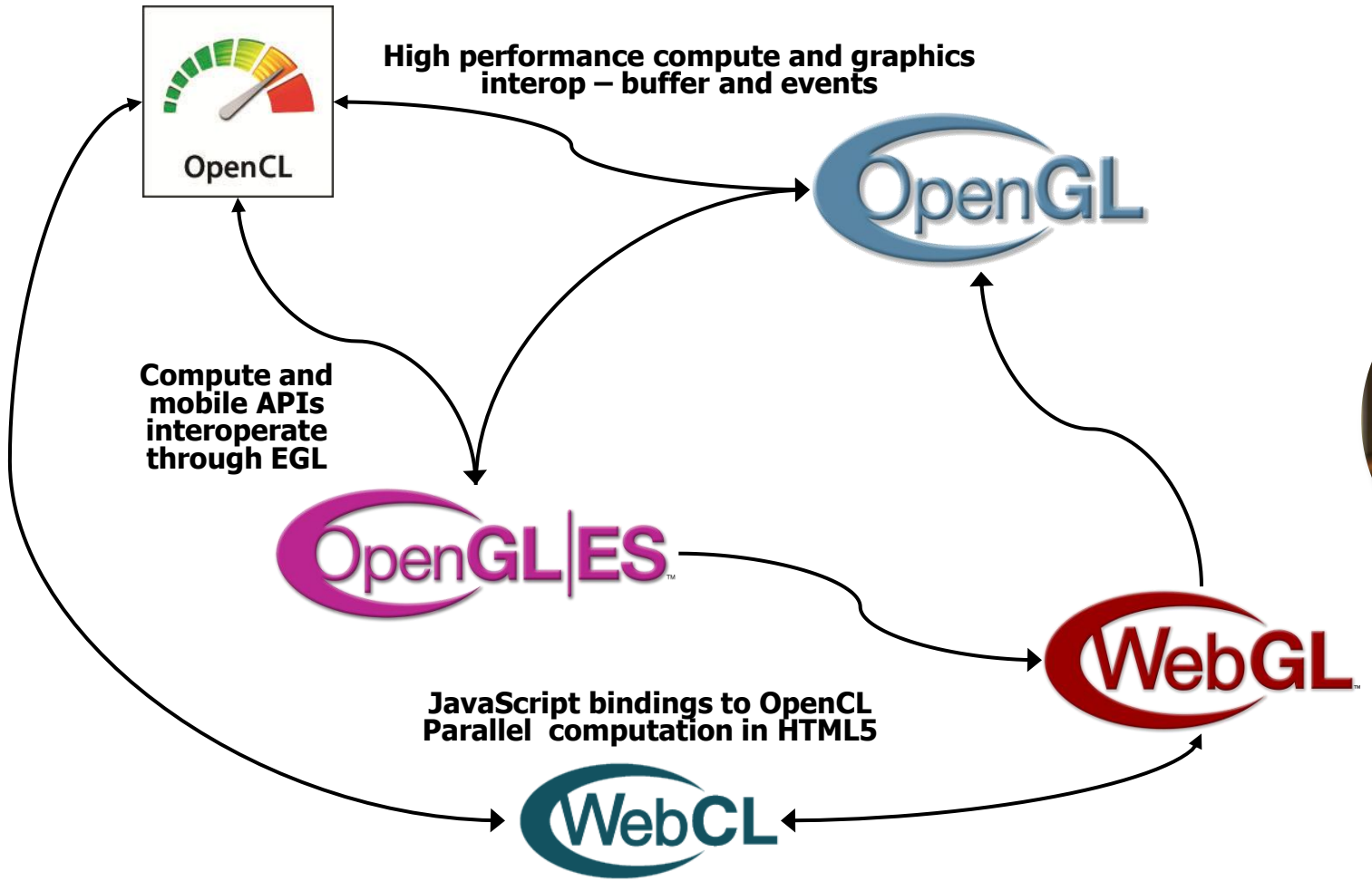


WebCL Open Process and Resources

- **Khronos open process to engage Web community**
  - Public specification drafts, mailing lists, forums
  - http://www.khronos.org/webcl/
  - webcl_public@khronos.org
- **Khronos welcomes new members to define and drive WebCL**
  - info@khronos.org
- **Nokia open sourced prototype for Firefox in May 2011 (LGPL)**
  - http://webcl.nokiaresearch.com
- **Samsung open sourced prototype for WebKit in July 2011 (BSD)**
  - http://code.google.com/p/webcl/

- **Deformation Demo:**
- Calculates and renders transparent and reflective deformed spheres on top of photo background
- Performance comparison on Mac
  - JS: ~1 FPS
  - WebCL: 87-116 FPS
- http://www.youtube.com/user/SamsungSISA#p/a/u/1/9Ttux1A-Nuc

Visual Computing Ecosystem



In Summary

- APIs are key to enable compelling applications on advanced hardware – APIs developed on high-end hardware are now enabling mobile devices
- APIs no longer exist alone – they are interoperating to form platform ecosystems for advanced content
- Significant cooperation between native and Web APIs to bring advanced visual computing to HTML5
- Khronos is driving open standards for hardware acceleration
  Join, change the industry AND get the inside edge for your products!

KHRONOS GROUP