



The State of Gaming APIs

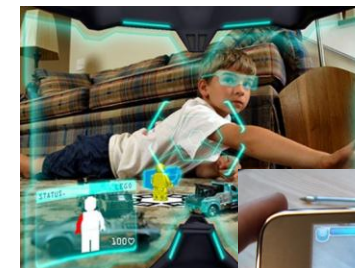
Neil Trevett
Vice President Mobile Content, NVIDIA
President, The Khronos Group

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

KHRONOS
GROUP
APIs enable software
developers to turn silicon
functionality into
rich end user experiences





K H R O N O S

GROUP

Over 100 members – any company worldwide is welcome to join

Board of Promoters



3D Evolution on PCs



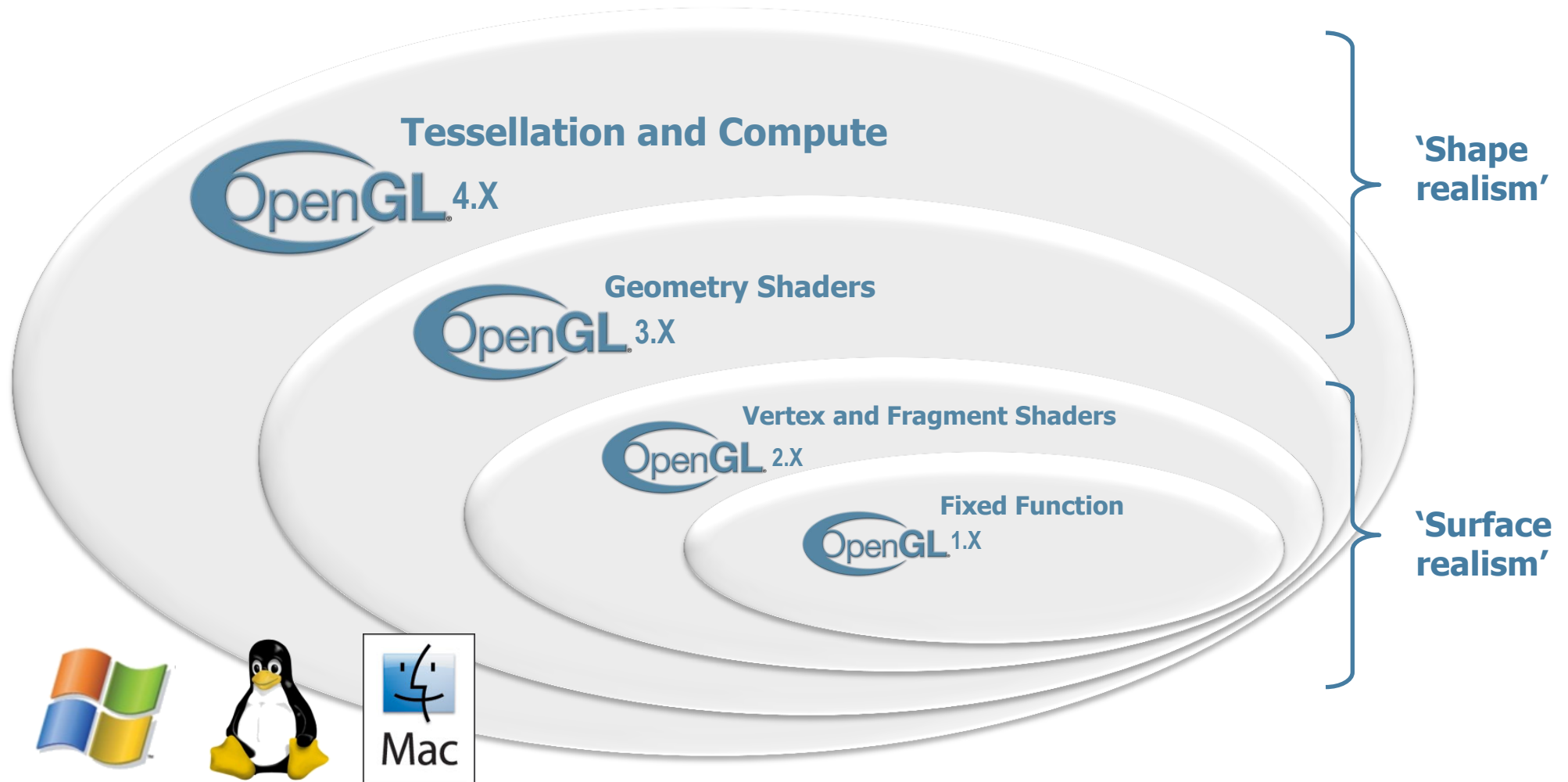
'Doom' on a PC – 1993
id Software



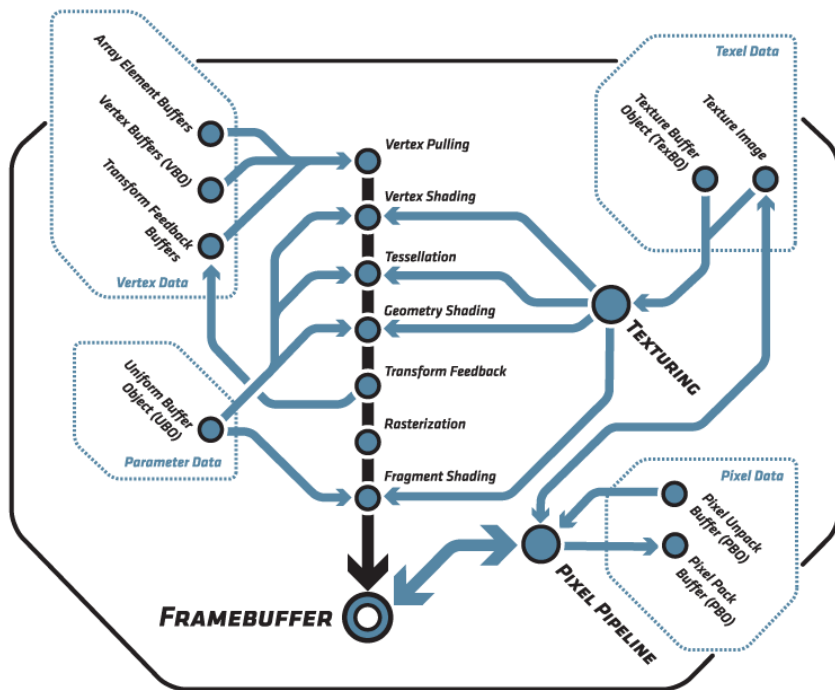
'Samaritan' Real-time Demo on a PC – 2011
Epic Unreal Engine

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 4.1

OpenGL 3.3/4.0

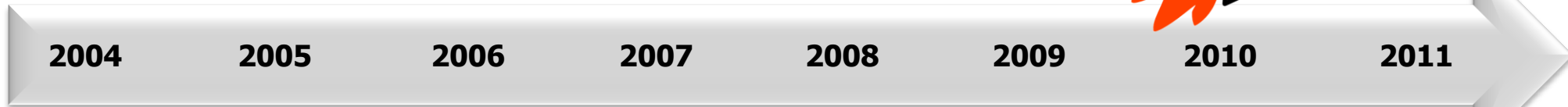
OpenGL 3.2

OpenGL 3.1

OpenGL 3.0

OpenGL 2.0

OpenGL 2.1



2004

2005

2006

2007

2008

2009

2010

2011

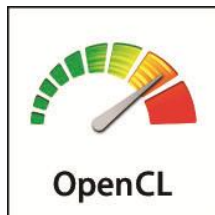
DirectX 9.0c

DirectX 10.0

DirectX 10.1

DirectX 11

Visual Computing Ecosystem



**Heterogeneous Parallel
Computation for physics
image processing etc.**

**High performance compute and
graphics interop – buffer and
events**

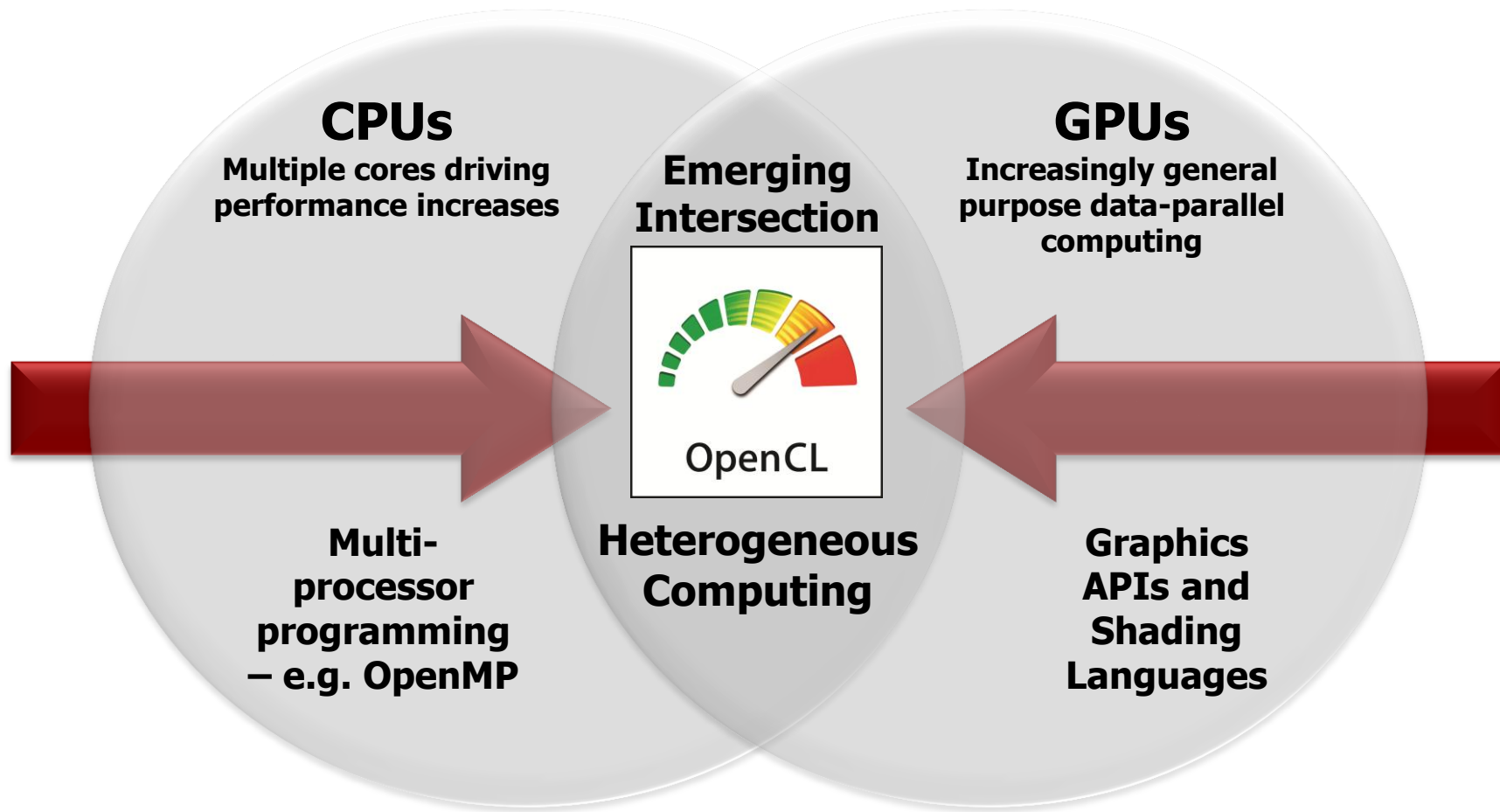


**Leading-edge functionality
developed on high-end systems**

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



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**



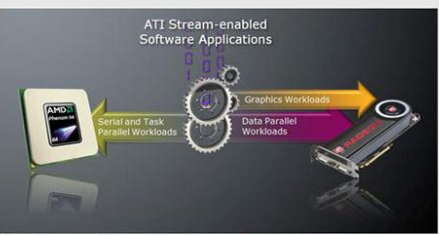
OpenCL Desktop Implementations

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


OpenCL™ Zone
Home > Zones > OpenCL™ Zone



OpenCL™ (Open Computing Language) is the first truly open and royalty-free programming standard for general-purpose computations on heterogeneous systems. OpenCL™ allows programmers to preserve their expensive source code investment and easily target multi-core CPUs, GPUs, and the new APUs.



Developed in an open standards committee with representatives from major industry vendors, OpenCL™ gives users what they have been demanding: a cross-vendor, non-proprietary solution for accelerating their applications on CPU, GPU, and APU.

VISUAL COMPUTING DEVELOPER COMMUNITY Intel® OpenCL SDK 

[Download Now](#)

About Intel® OpenCL SDK

About OpenCL™
OpenCL™ (Open Computing Language) is the first open, royalty-free standard for general-purpose parallel programming of heterogeneous systems. OpenCL provides a uniform programming environment for software developers to write efficient, portable code for client computer systems, high-performance computing servers, and handheld devices using a diverse mix of multi-core CPUs and other parallel processors.

About Intel® OpenCL SDK 1.1
Intel® OpenCL SDK 1.1 is Intel's implementation of the OpenCL standard optimized for Intel processors, running on Microsoft® Windows®, and Linux® operating systems. This SDK implementation is fully conformant with the OpenCL 1.1 specification for the CPU, and with Microsoft® Windows® 7 operating systems.

Developers are now able to use the Intel® OpenCL SDK to create and distribute OpenCL based applications optimized for Intel® Core™ and Intel® Xeon® processors.

Technical Content

Getting Started

- [Announce Intel® OpenCL SDK 1.1 New!](#)
- [Release Notes New!](#)
- [Installation notes](#)
- [Intel® OpenCL SDK 1.1 FAQ](#)
- [Intel® OpenCL SDK User Guide \(pdf\)](#)


Support and Feedback

[Intel® OpenCL SDK FAQ \(Frequently Asked Questions\)](#)
Answers to the most common questions asked by OpenCL developers

[Forums](#) - Get answers to your questions about Intel® OpenCL SDK from Intel engineers and other OpenCL developers

DEVELOPER ZONE
DEVELOPER CENTERS TECHNOLOGIES TOOLS RESOURCES COMMUNITY

OpenCL



OpenCL™ (Open Computing Language) is a low-level API for heterogeneous computing that runs on CUDA architecture GPUs. Using OpenCL, developers can write compute kernels using a C-like programming language to harness the massive parallel computing power of NVIDIA GPU's to create compelling computing applications. As the OpenCL standard matures and is supported on processors from other vendors, NVIDIA will continue to provide the drivers, tools and training resources developers need to create GPU accelerated applications.

In partnership with NVIDIA, OpenCL was submitted to the Khronos Group by Apple in the summer of 2008 with the goal of forging a cross platform environment for general purpose computing on GPUs. NVIDIA has chaired the industry working group that defines the OpenCL standard since its inception and shipped the world's first conformant GPU implementation of OpenCL for both Windows and Linux in June 2009.

NVIDIA has been delivering OpenCL support in end-user production drivers since October 2009, supporting OpenCL on all 300,000,000+ CUDA architecture GPUs shipped since 2006. OpenCL v1.1 support is included in publicly available NVIDIA drivers version 280.13 or later on the [driver download page](#)

- For OpenCL v1.1 support on Windows Server®, use the Windows 7 drivers
- Windows XP drivers with OpenCL v1.1 support are available for GeForce desktop products only

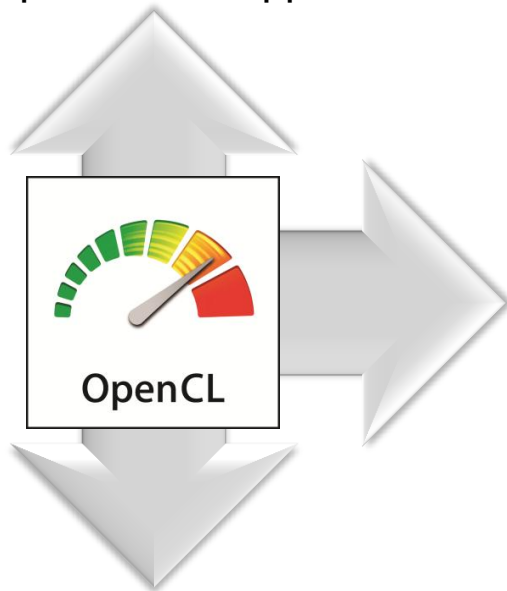
NVIDIA also provides powerful performance analysis tools for OpenCL developers, including NVIDIA [Parallel Nsight](#) for Visual Studio and NVIDIA [Visual Profiler](#) for Linux and MacOS.

On the same day Khronos Group announced the new OpenCL v1.1 specification update (June 14th, 2010), NVIDIA released OpenCL v1.1 pre-release drivers and SDK code samples to all GPU Computing registered developers. [Log in or apply for an account to download Intel/NVIDIA Drivers and Toolkits.](#)

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



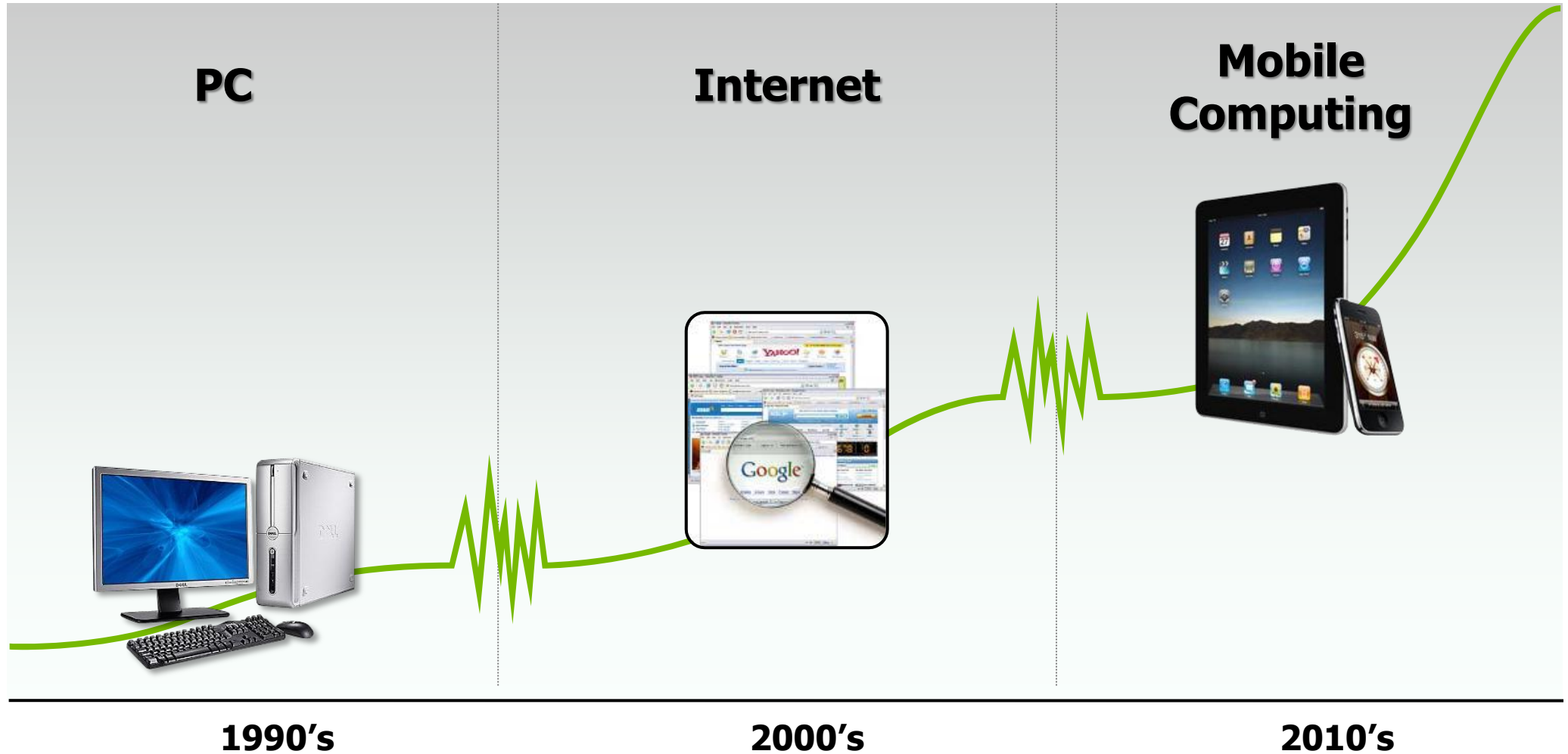
Long-term Core Roadmap

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

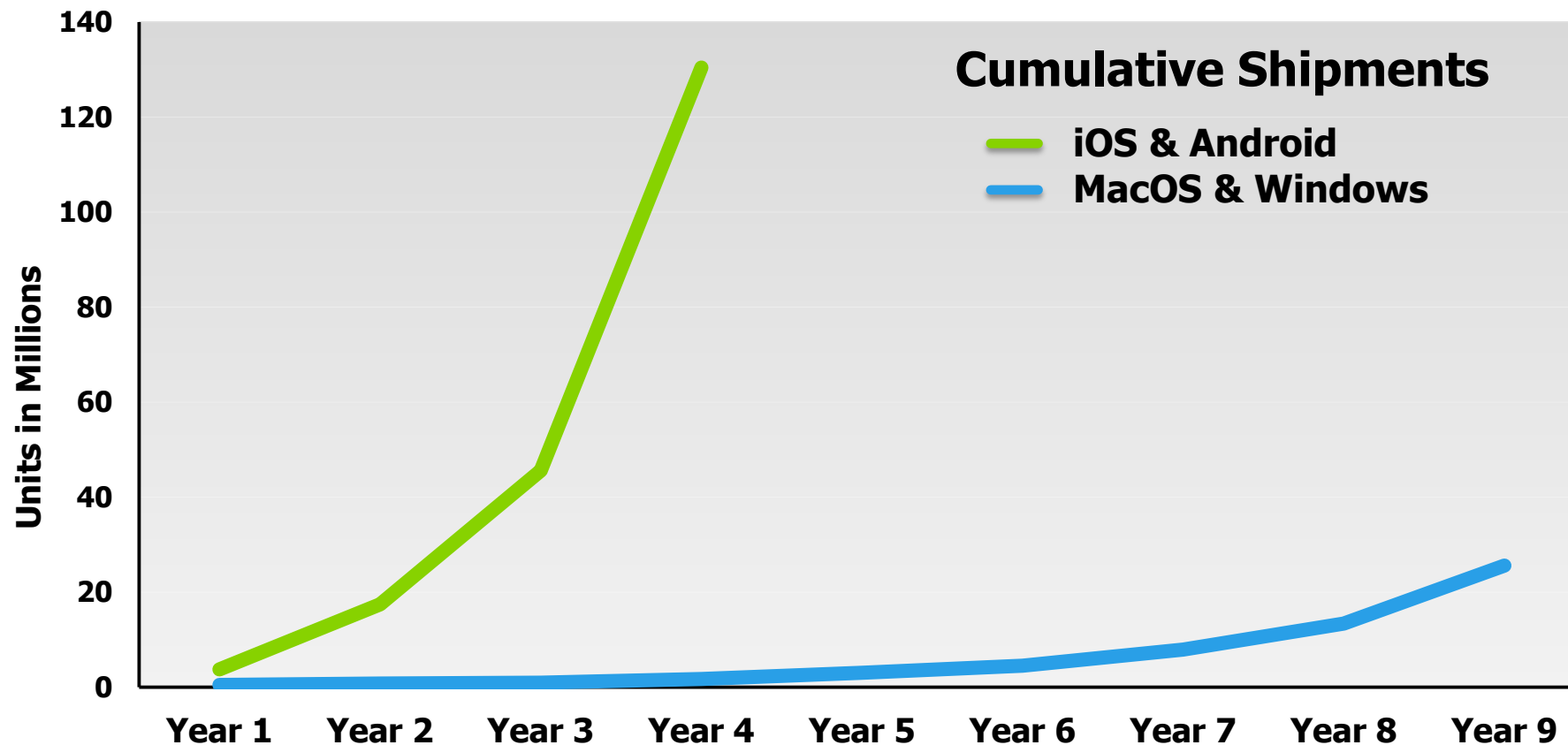
OpenCL-SPIR

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

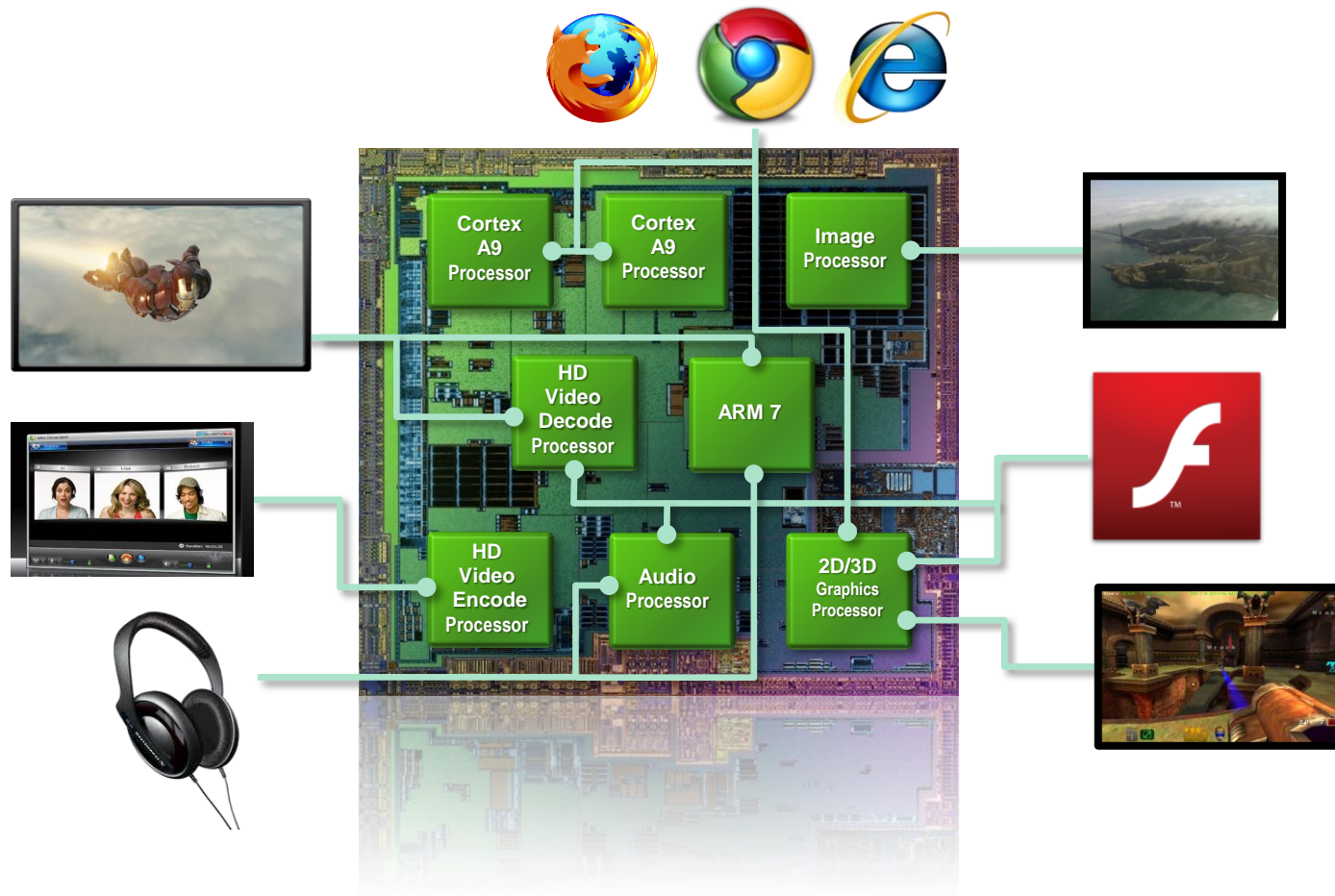


20 Years Faster to 100M Per Year

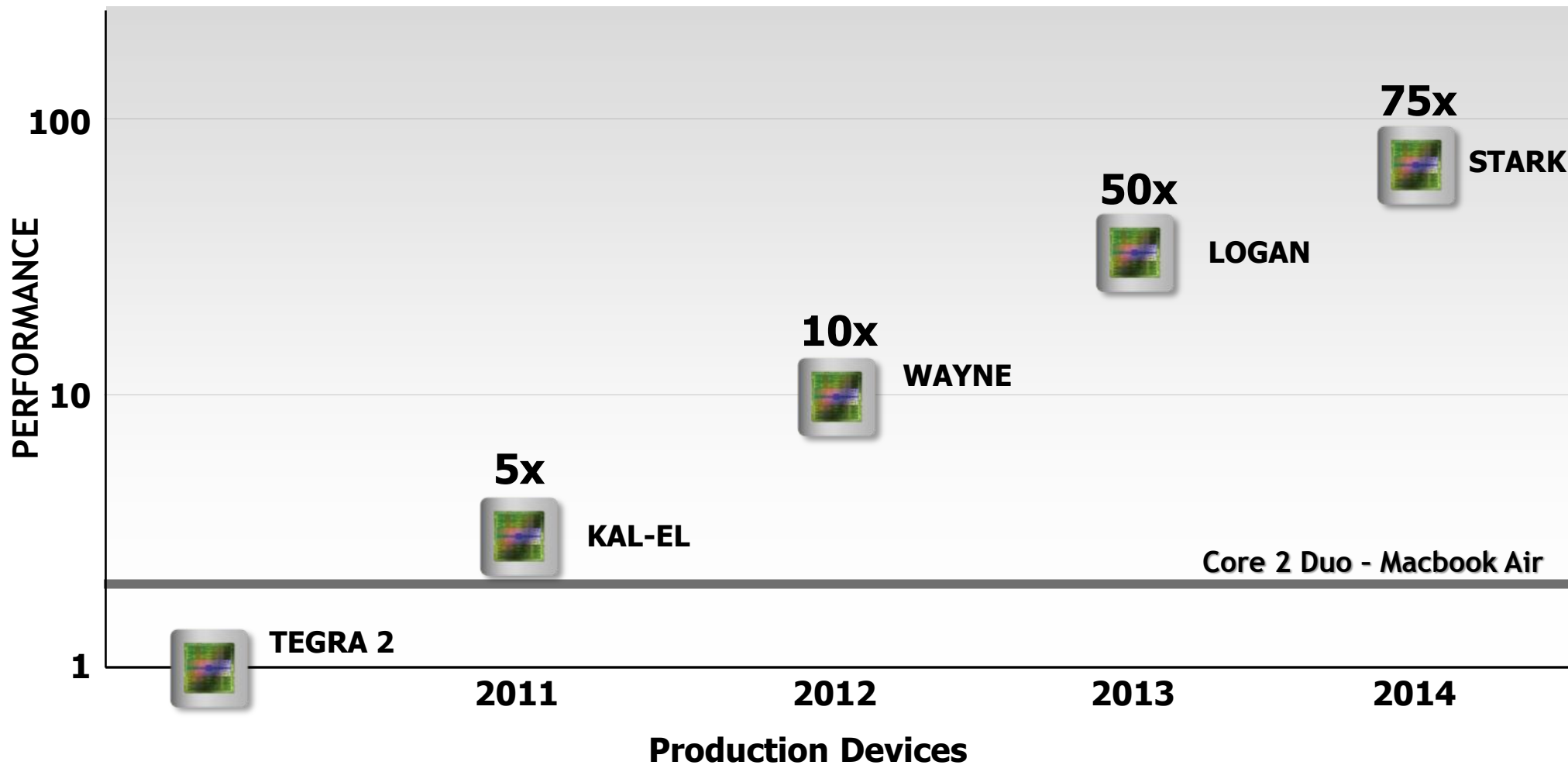


Source: Gartner, Apple, NVIDIA

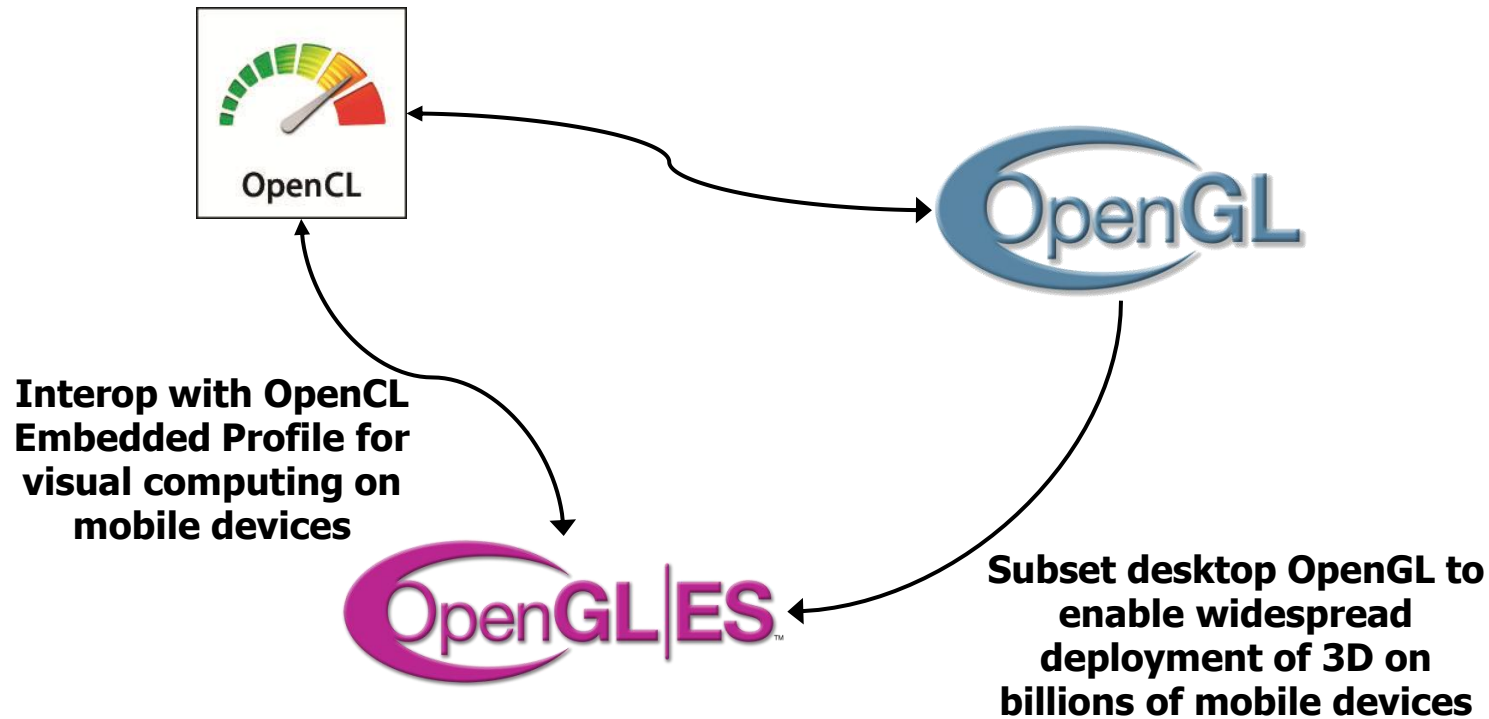
Mobile Silicon Experiential Processing



Mobile Roadmap Acceleration

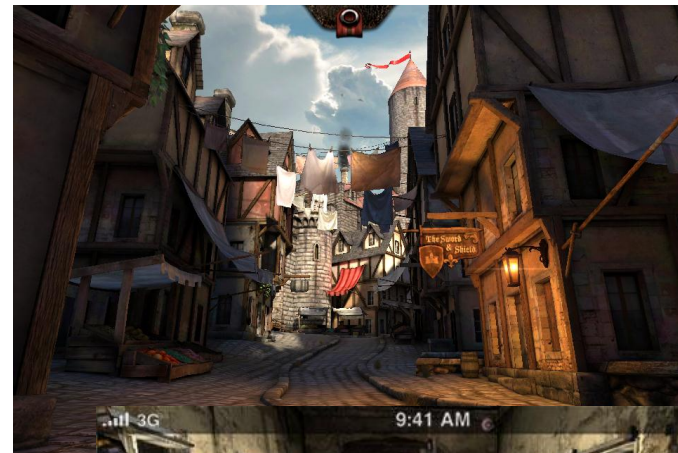


Visual Computing Ecosystem



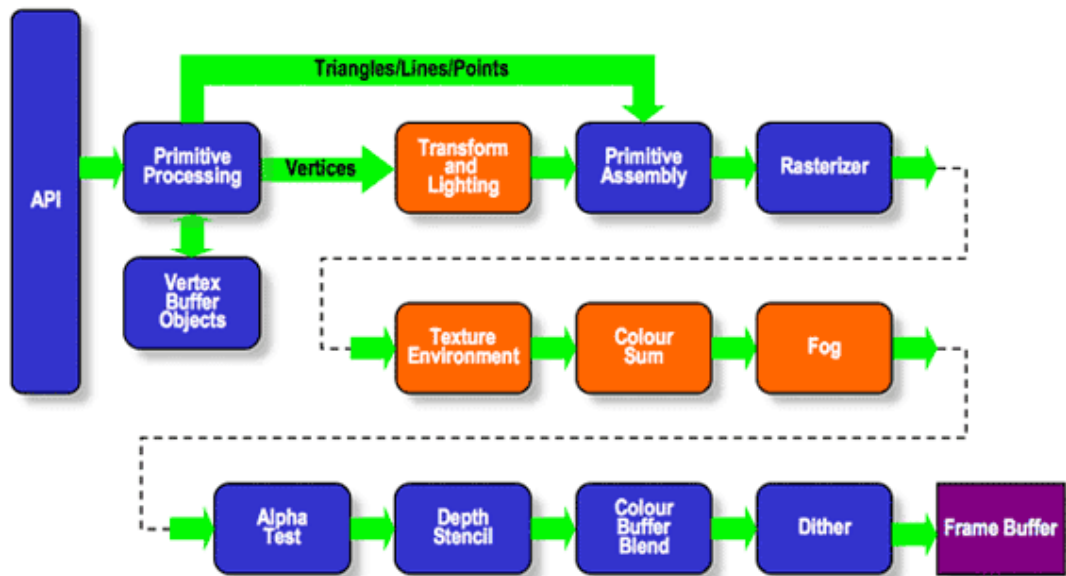
What is OpenGL ES?

- **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



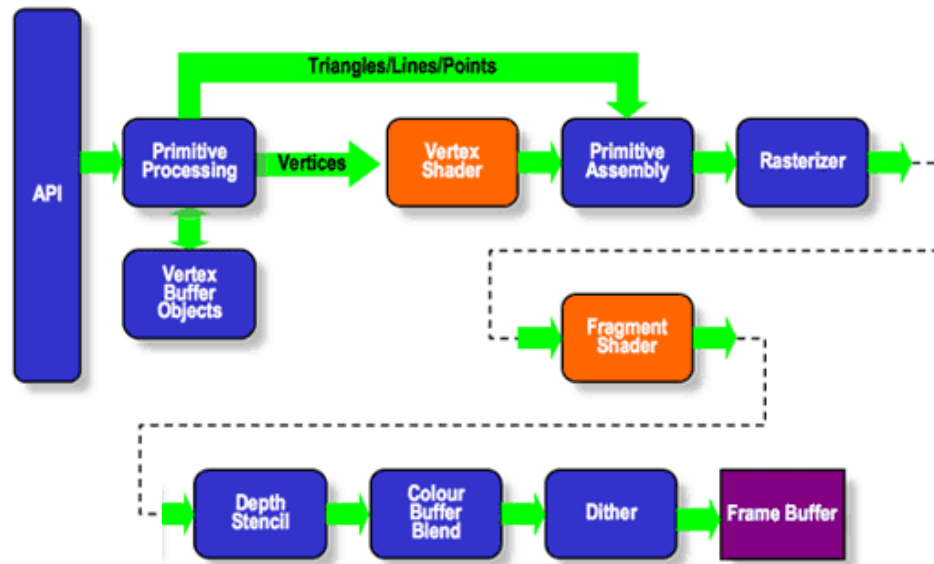
OpenGL ES Pipelines

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

OpenGL ES – Advanced Audio

- **OpenGL 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



OpenGL 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

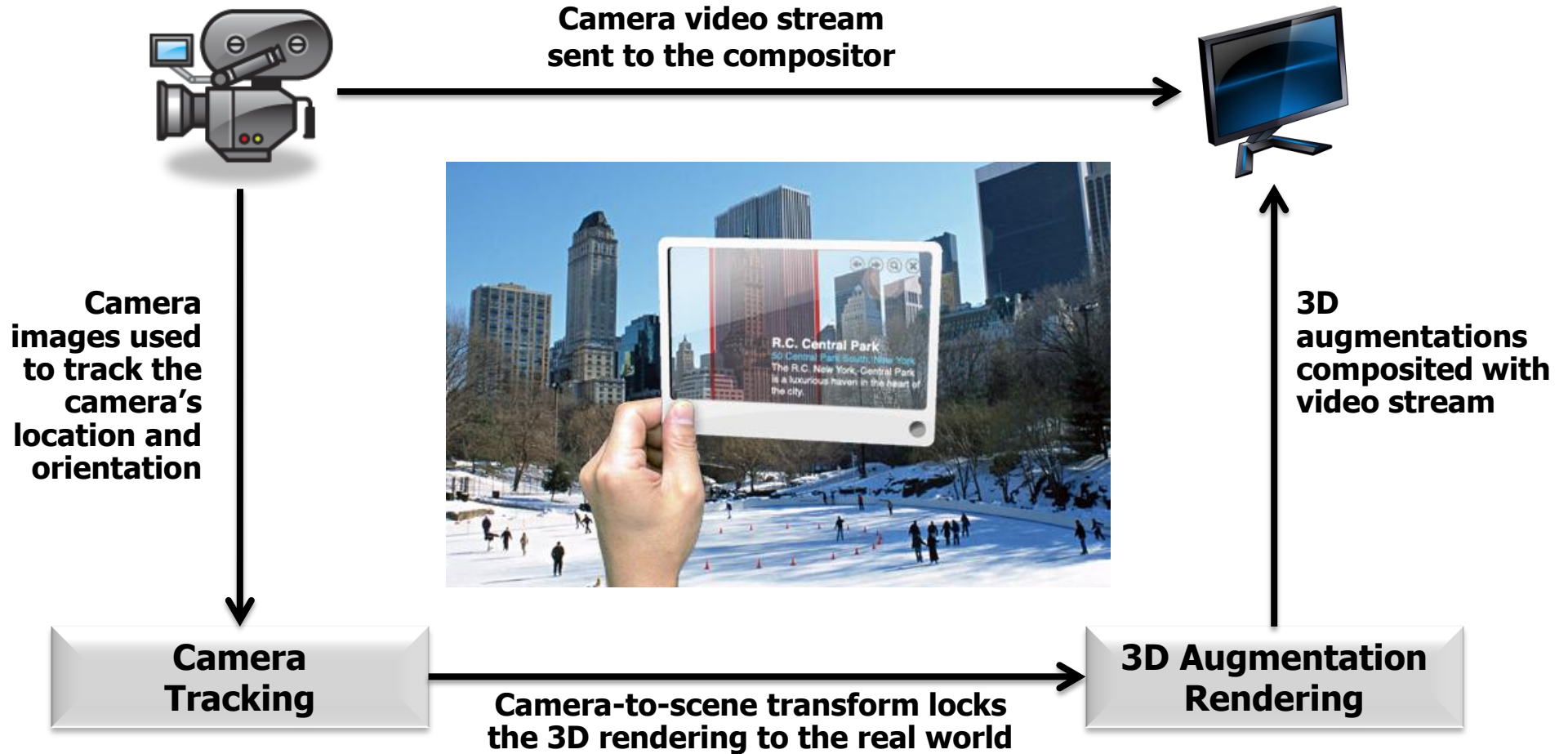


Basic mobile phones

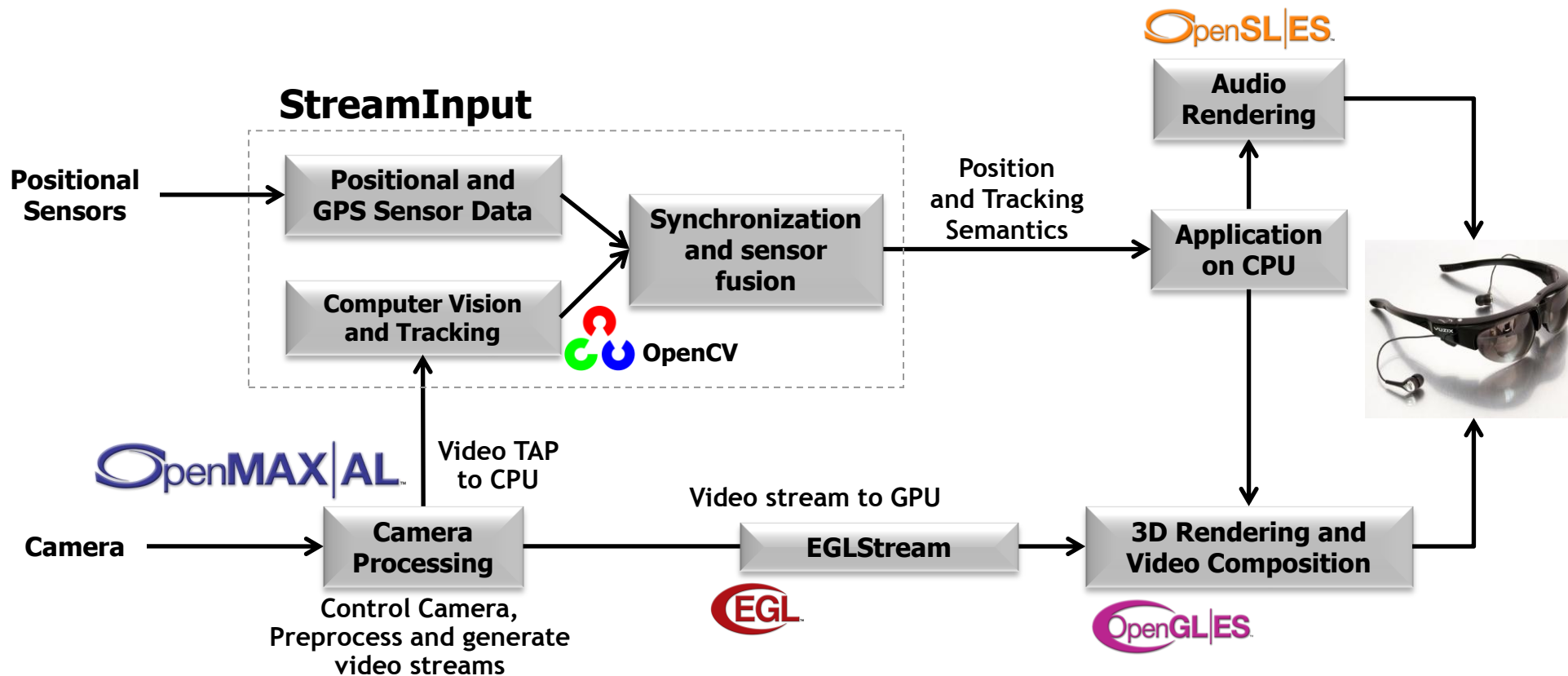
Ring tone and alert tone playback (basic MIDI functionality), basic audio playback and record functionality, simple 2D audio games

A device can implement any combination of profiles

Visual-based Augmented Reality



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™

Advanced image capture and photography

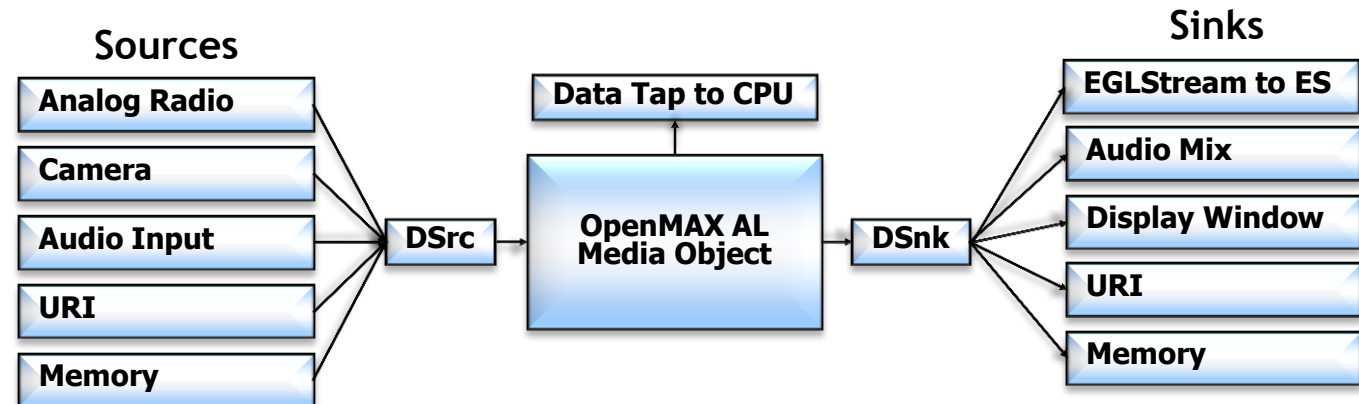
HD content playback with robust DRM

HD video teleconferencing

Augmented Reality

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



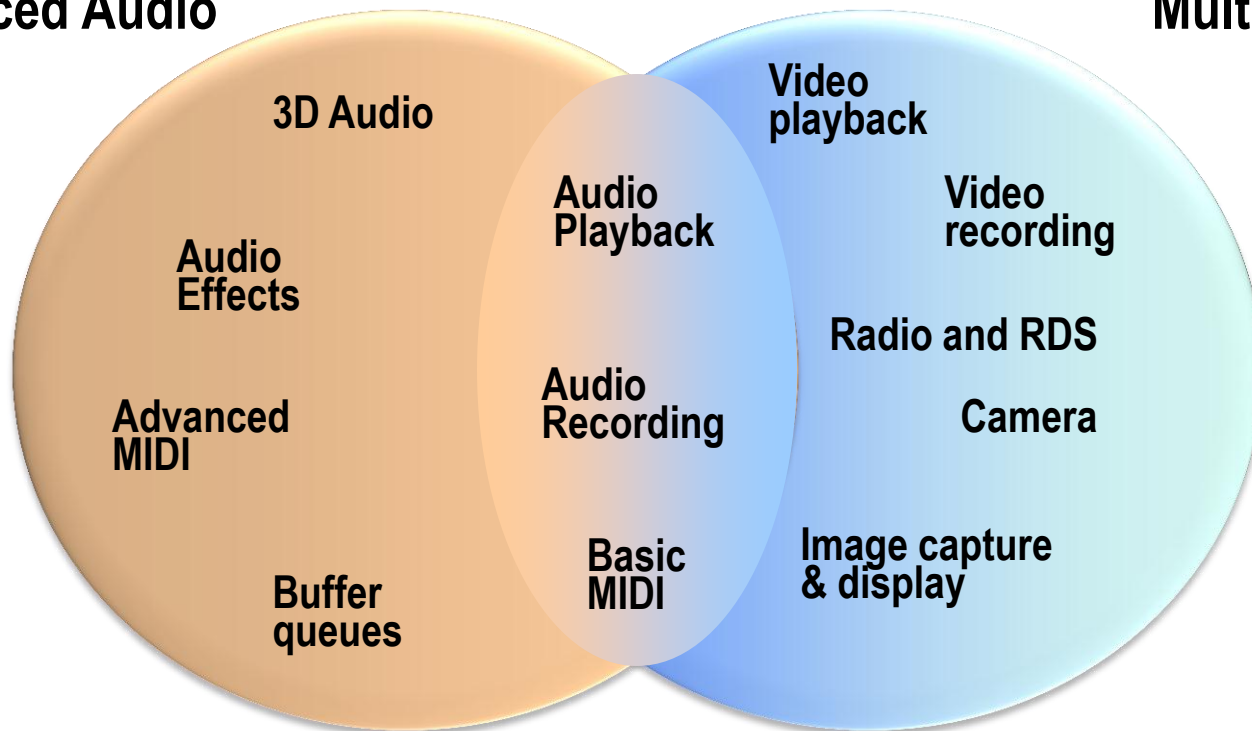
OpenGL ES and OpenMAX AL

OpenSL|ES

Advanced Audio

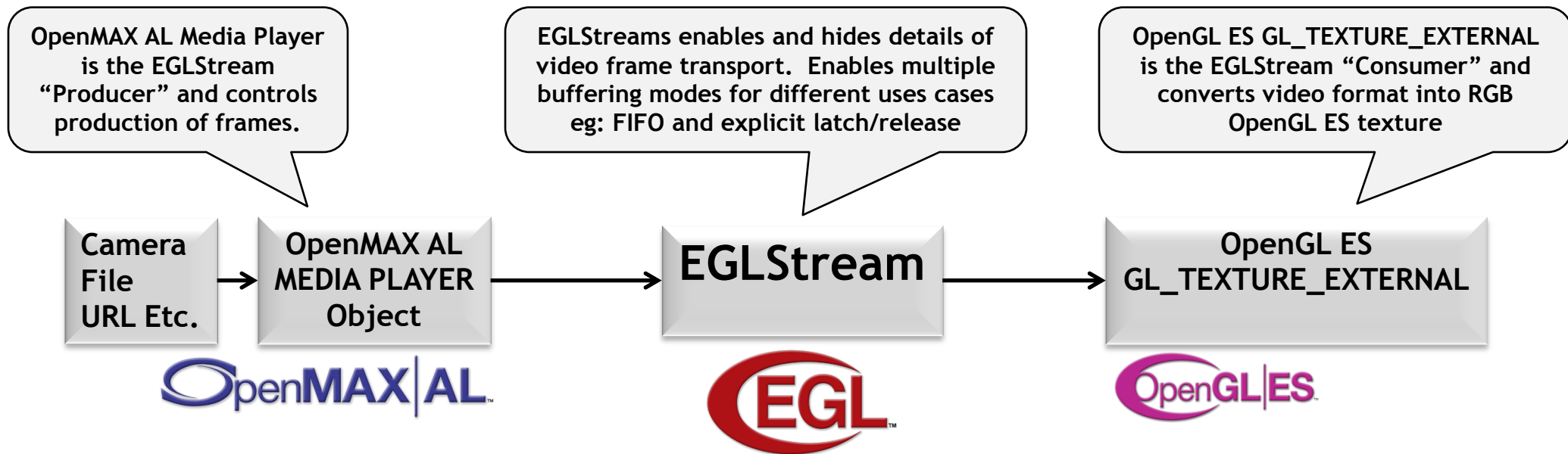
OpenMAX|AL

Multimedia



Common Object-oriented API Framework

EGLStream – Video/Graphics Interop



StreamInput Connects Sensors to Apps



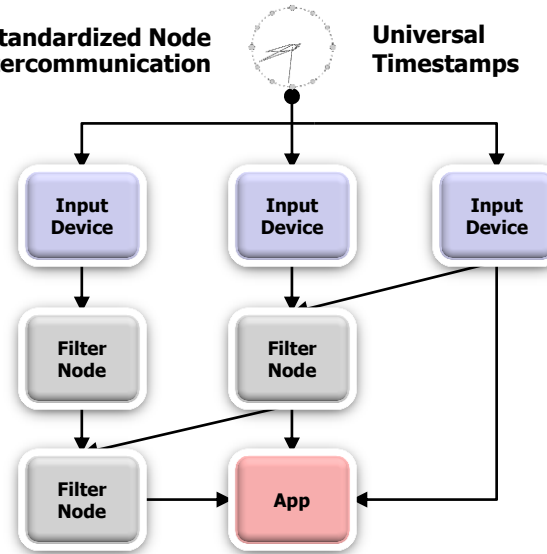
Advanced Sensors Everywhere

Standard cameras, depth cameras
motion and position, touch, microphones
wireless controllers

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

Standardized Node
Intercommunication

Universal
Timestamps



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

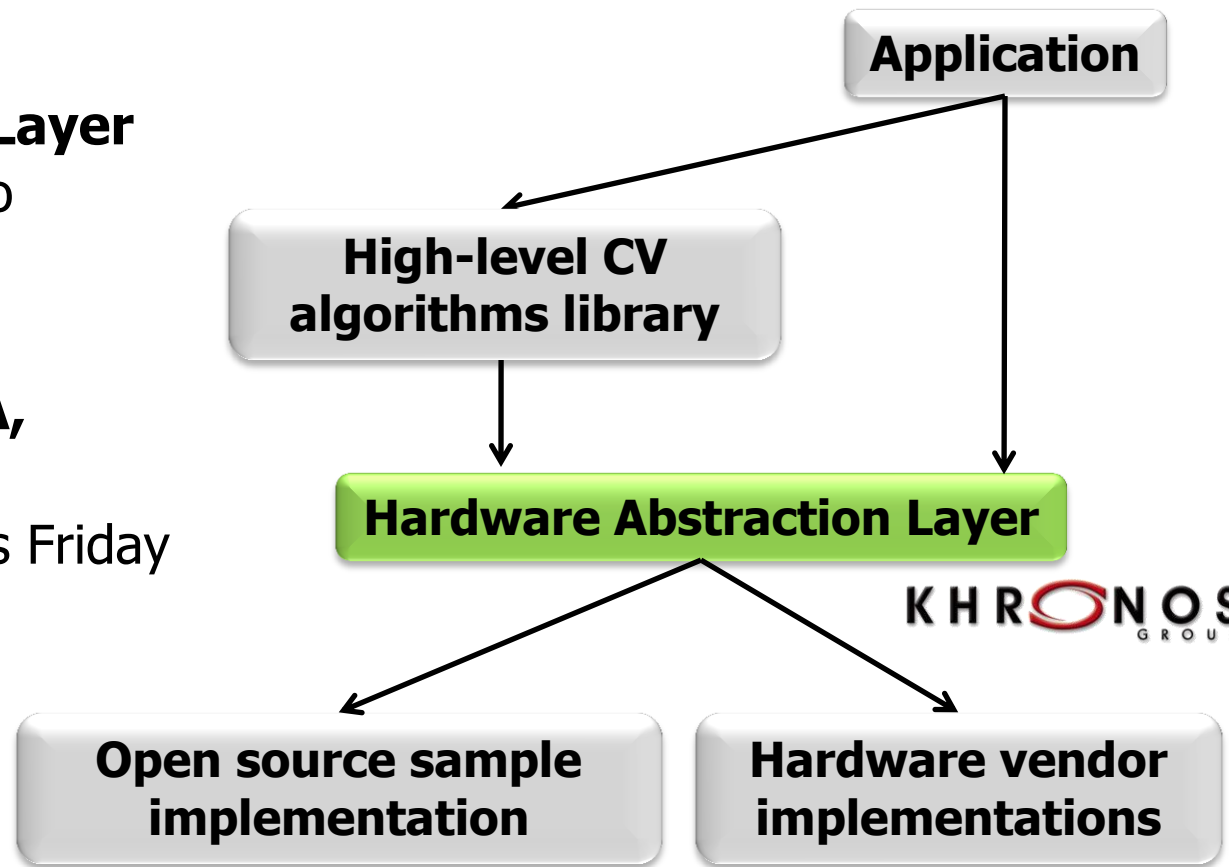
Current StreamInput Participants

- Aiming for production implementations in September 2012

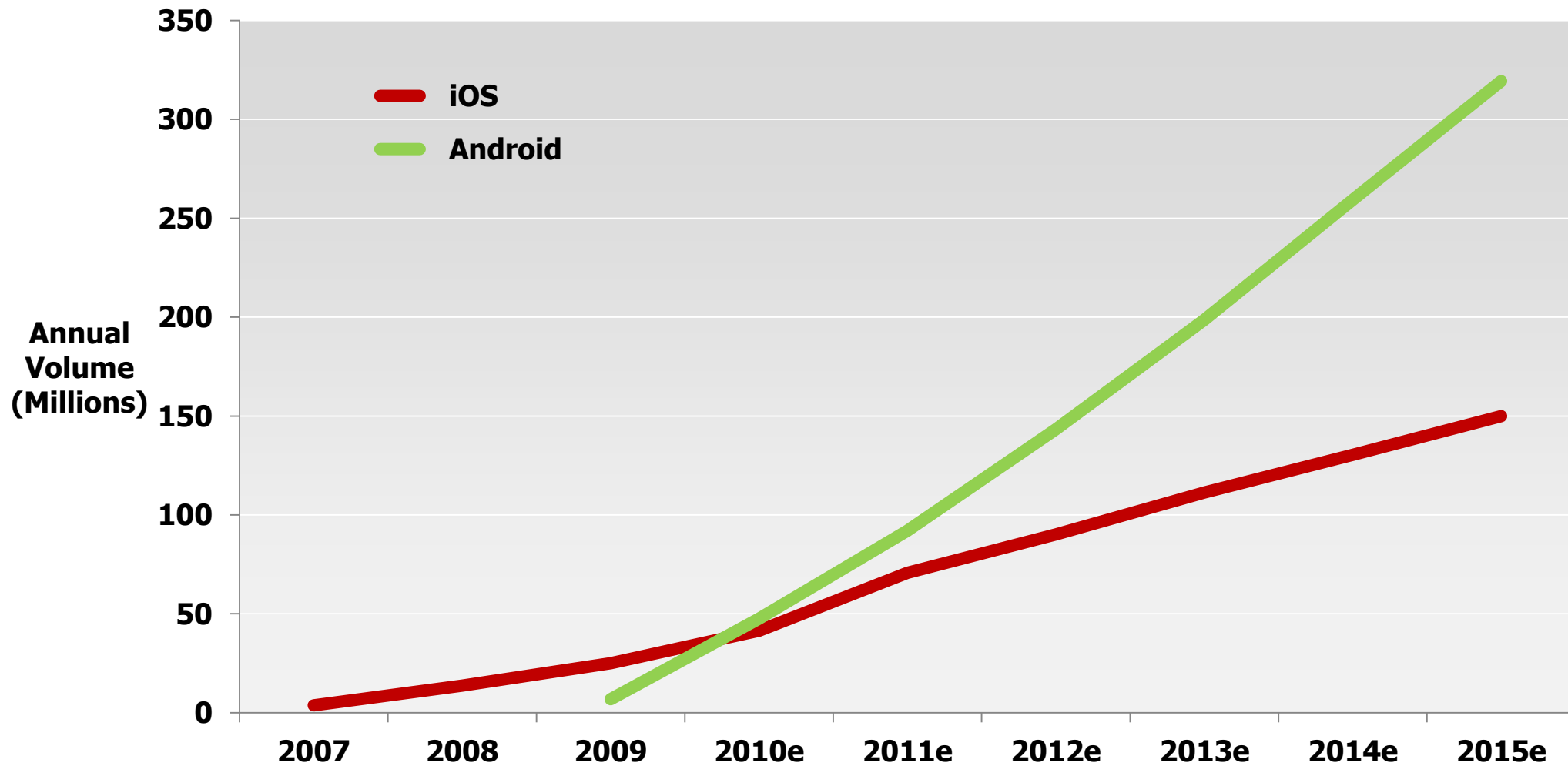


OpenCV as Potential Khronos Standard

- **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



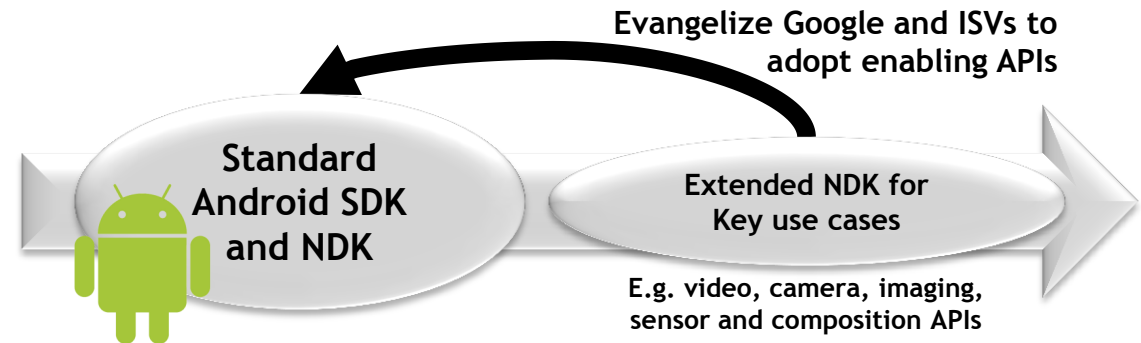
Mobile - Android Becoming Dominant OS














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 selectionMultiple apks behind multi-Apk SKU



Android Native API Adoption

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

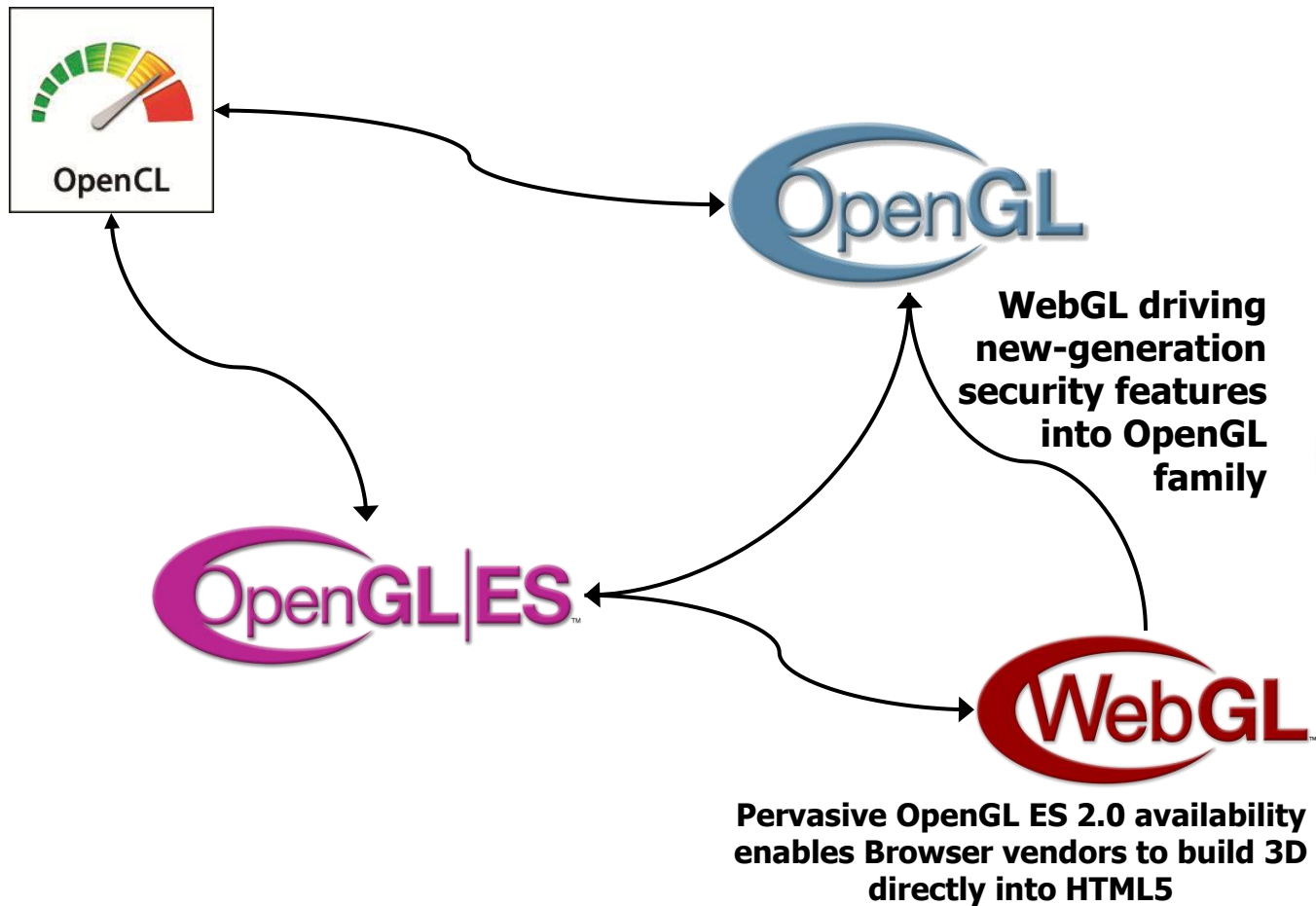


Mobile Web versus Apps

- **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 – 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

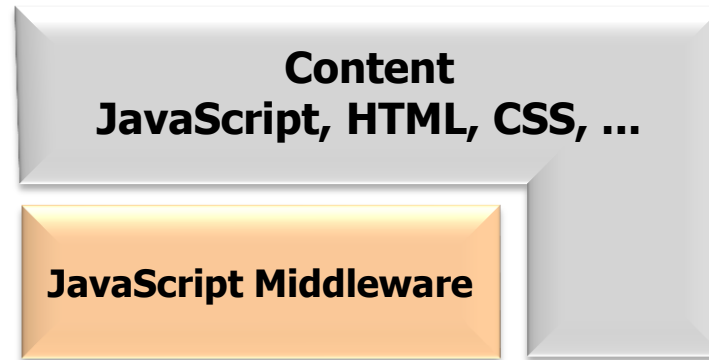


Being defined by major browsers and GPU vendors working together



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

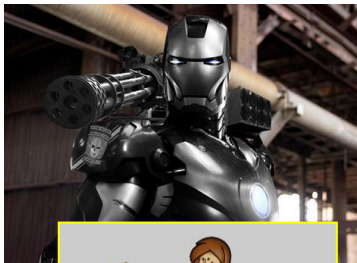
HTML content generated by layout engine 'on page'

Mollit Anim
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

lorem ipsum

All content passes through CSS layout

<video> tag



<canvas> tag



Composition of off-screen buffers

CSS Layout and Transforms

Composition needs to be GPU accelerated

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

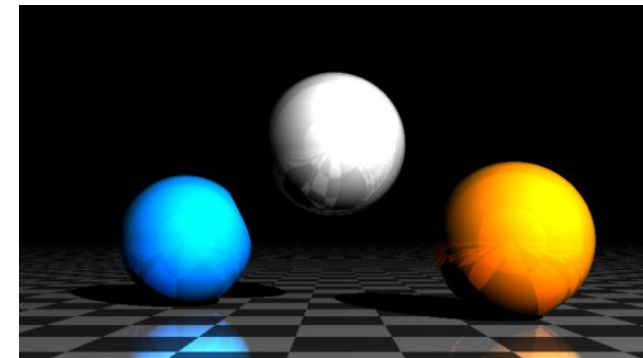
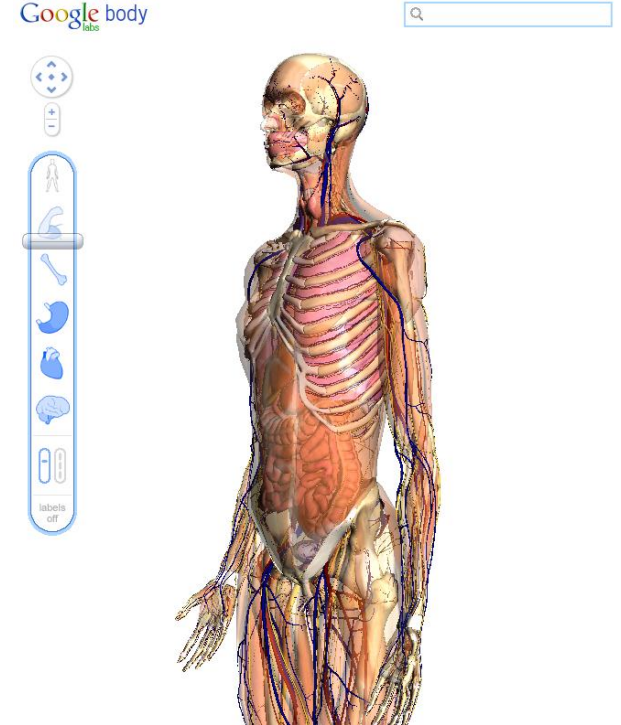


JavaScript drives interactivity for 2D and 3D graphics

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

Google body



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**



The screenshot shows the WebGL website's 'User Contributions' page. The page title is 'User Contributions' and it includes a sub-header 'This is a list of all the WebGL related activities happening on the web. If you...'. The main content is a list of frameworks and tools, organized into two sections: '1 Frameworks' and '2 Utilities & Debug Helpers'. The '1 Frameworks' section lists 18 items, including C3DL, CopperLicht, CubicVR, EnergizeGL, GammaJS, GLGE, GTW, Jax, O3D, PhiloGL, SceneJS, SpiderGL, TDL, Three.js, X3DOM, and WebGL Google Web Toolkit bindings. The '2 Utilities & Debug Helpers' section lists 5 items: WebGLU, WebGLTrace, WebGLDebugUtils, WebGLUtils, and gluUnProject. The page also features a 'WebGL' logo and a navigation menu with links like 'Main page', 'WebGL Message Board', 'Public Mailing List', 'Recent changes', 'Random page', and 'Help'.

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**



JavaScript

Native

 Native APIs shipping or working group underway

 JavaScript API shipping or working group underway

 Possible future JavaScript APIs

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

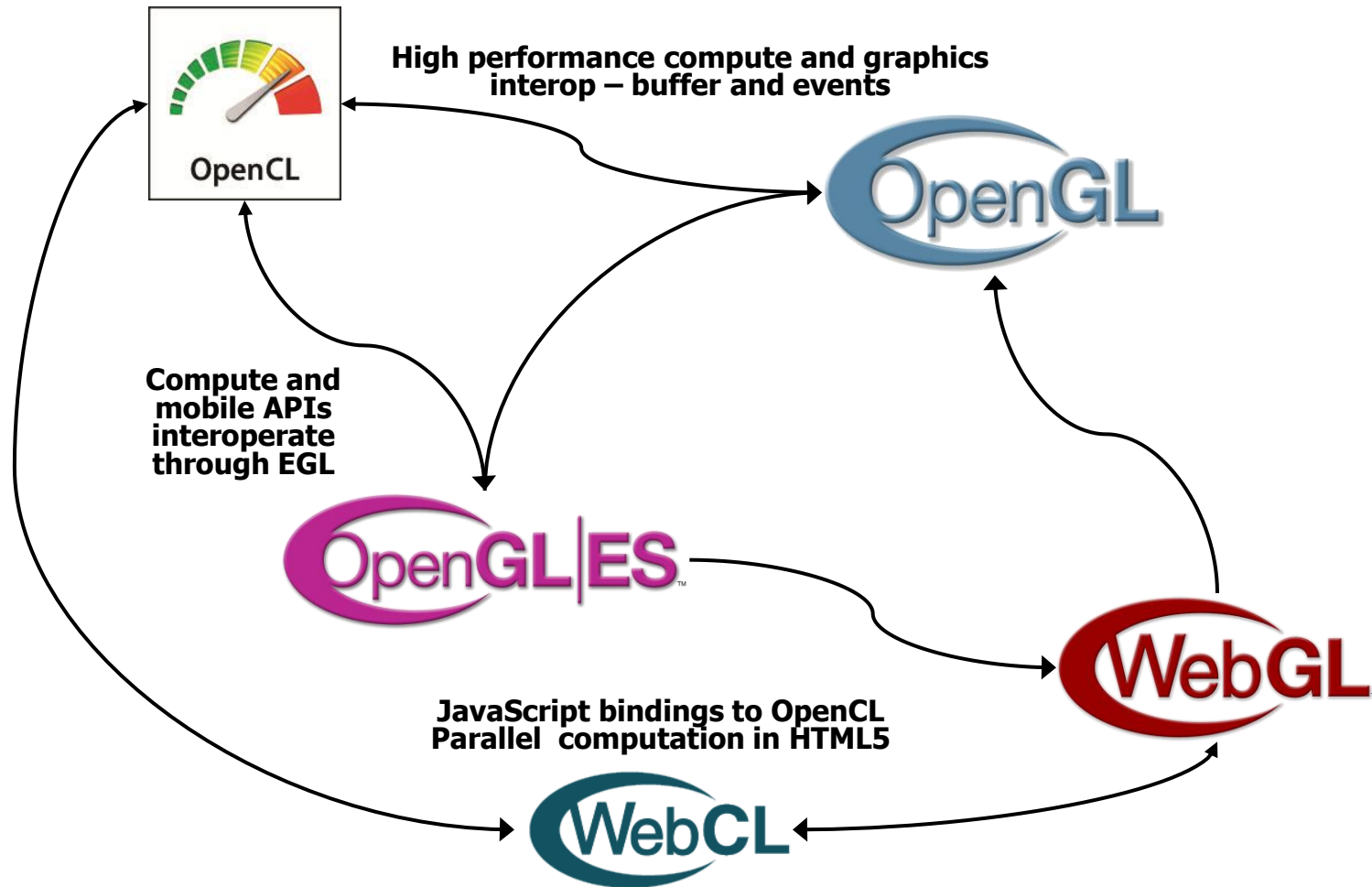
- **Khronos open process to engage Web community**
 - Public specification drafts, mailing lists, forums
 - <http://www.khronos.org/webcl/>
 - webcl_public@khronos.org
- **Khronos welcomes new members to define and drive WebCL**
 - info@khronos.org
- **Nokia open sourced prototype for Firefox in May 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!**

