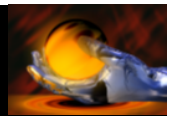


KHRONOS GROUP™

Khronos Mobile Graphics and Media Ecosystem

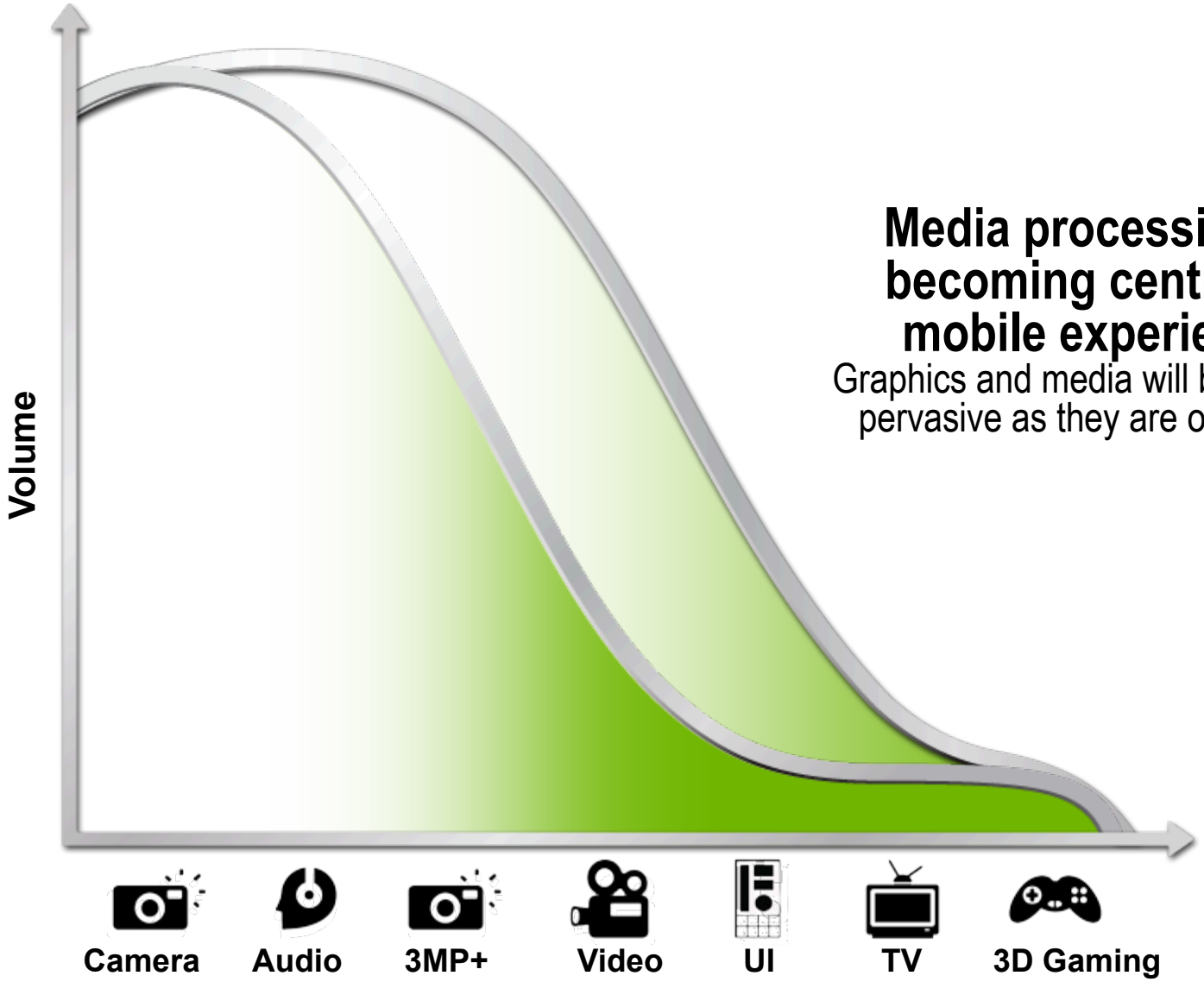
Neil Trevett
President, Khronos Group
Vice President Embedded Content, NVIDIA

Agenda for This Session

Mobile Graphics and Media

Speaker	Company	Start	Description
Neil Trevett	NVIDIA	15:15	Introduction Khronos Mobile Ecosystem
Dave Shreiner	ARM	15:50	Introduction to OpenGL ES
Tom McReynolds	NVIDIA	16:25	OpenGL ES in Automotive and Embedded Markets
Q&A	ALL	16:50	Audience Questions
End of day		17:00	DEMOS

Mobile Platform Media Evolution



Media processing is becoming central to mobile experience

Graphics and media will become as pervasive as they are on the PC

Mobile Media Fragmentation

- **Every handset is unique from the programmers perspective**
  - Differences in operating system functions, Java implementations and media functionality

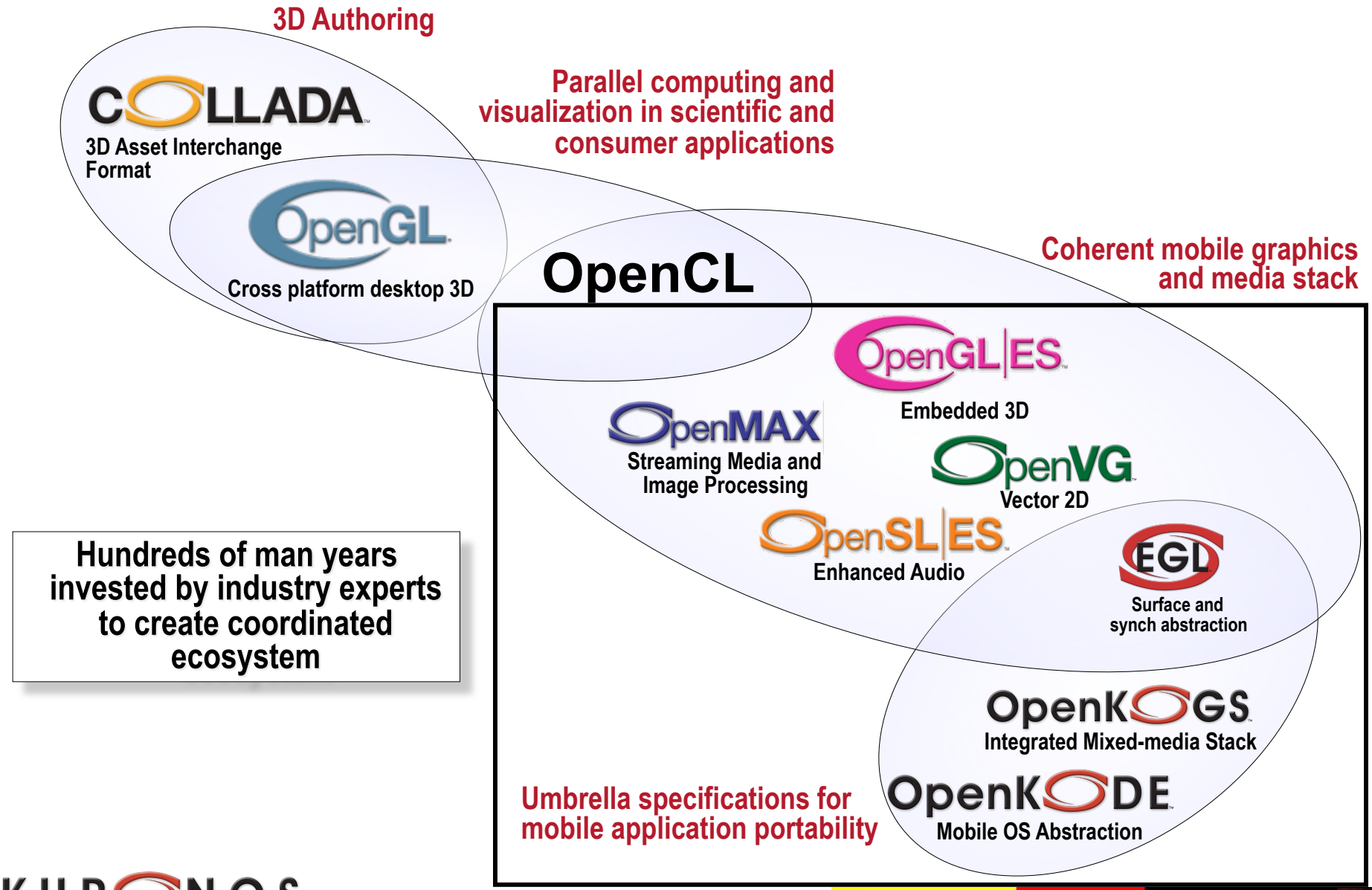
Severe platform fragmentation
ISVs need to port to and support 100s (even 1000s) of source variants of each title

What Standards Does Khronos Create?

- **"Foundation-Level" acceleration APIs**
  - Needed on every platform to support an ecosystem of middleware and applications
- **Low-level access to processor silicon**
  - Designed with strong silicon vendor participation
- **Cross-vendor software portability**
  - API abstractions just high enough to hide implementation specifics
- **Established focus on graphics, media and parallel compute acceleration**
  - 3D, vector 2D, video, imaging, audio, heterogenous parallel programming APIs…

Khronos APIs create the foundation of an ecosystem that enable applications to be PORTABLE and ACCELERATED on diverse silicon platforms

OpenCL and the Khronos Ecosystem



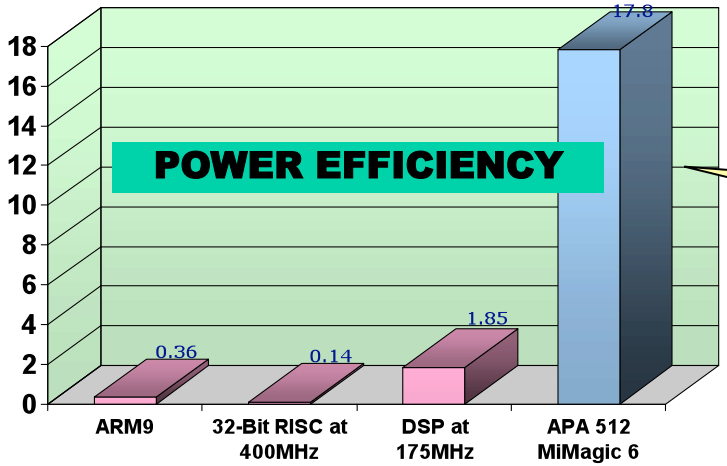
Hundreds of man years invested by industry experts to create coordinated ecosystem

Advantages of Mobile Graphics Acceleration

Faster Performance at Higher Quality
Hardware delivers smoother interaction with much better looking graphics

Software 3D

Accelerated 3D



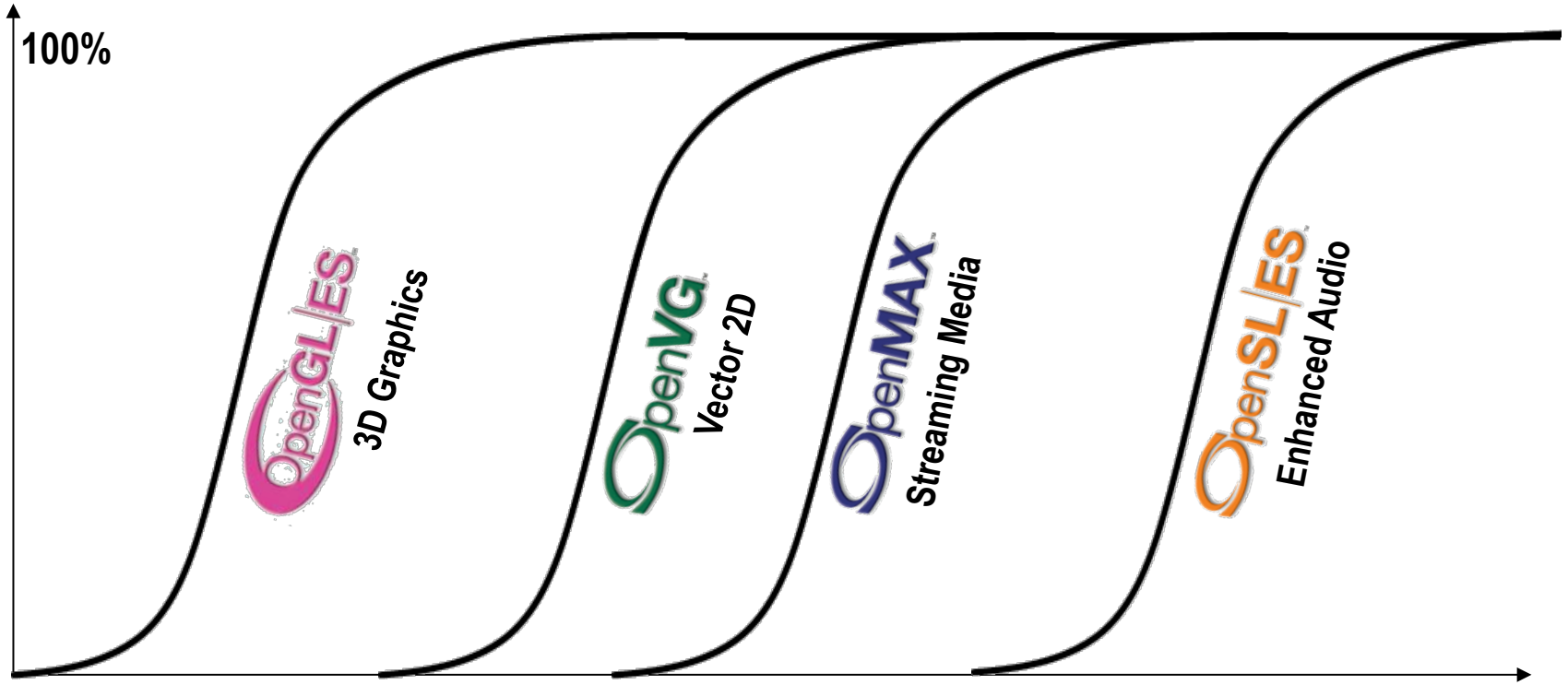
Less Power
Hardware accelerators exploit media pipeline parallelism and caching for a x10 increase in power efficiency over software

Better User Experience
Smaller screens need more advanced graphics processing per pixel

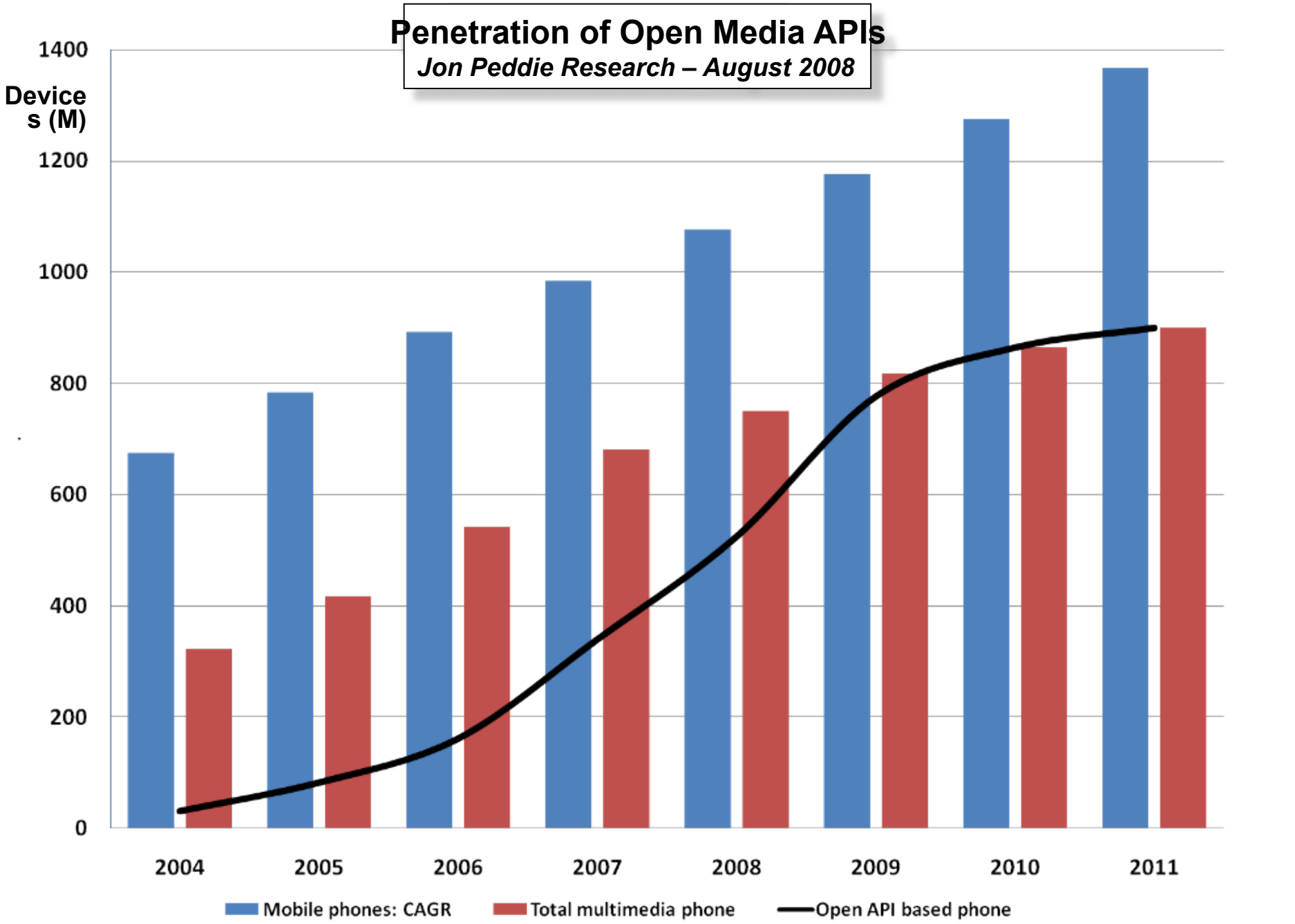
Adoption of Khronos Mobile APIs

- OpenGL ES 1.1 is becoming widespread, OpenGL ES 2.0 silicon has arrived
-OpenVG & OpenMAX IL – becoming widely implemented by silicon vendors
- OpenMAX AL and OpenSL ES – preliminary specs available – implementations started

Market adoption in media-accelerated handsets



Khronos Mobile Ecosystem



OpenGL ES

- **The leading 3D rendering API for mobile devices**
  - Based on desktop OpenGL – but optimized for mobile / handheld devices
  - Removes redundancy & rarely used features - adds mobile-friendly data types
  - The power of OpenGL distilled into a much smaller package
- **A smashing success!**
  - Widely used in mobile phone handsets from every major handset manufacturer
  - Personal Navigation, Personal Media Player, Automotive, Set-Top Box, Mobile Internet Device
  - Brew, Windows Mobile, Symbian, Android, iPhone OS, Limo
- **OpenGL ES has become the most widely deployed 3D API**
  - Used in diverse applications, devices and markets

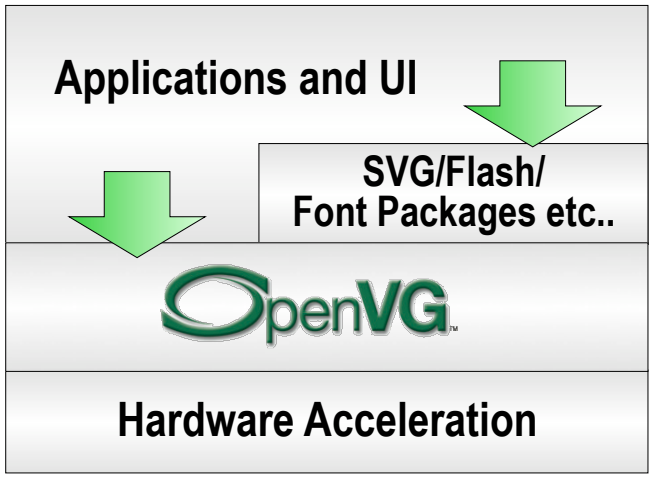
Mobile OS that have adopted OpenGL ES as their native 3D API

OpenVG - Accelerated Vector Graphics

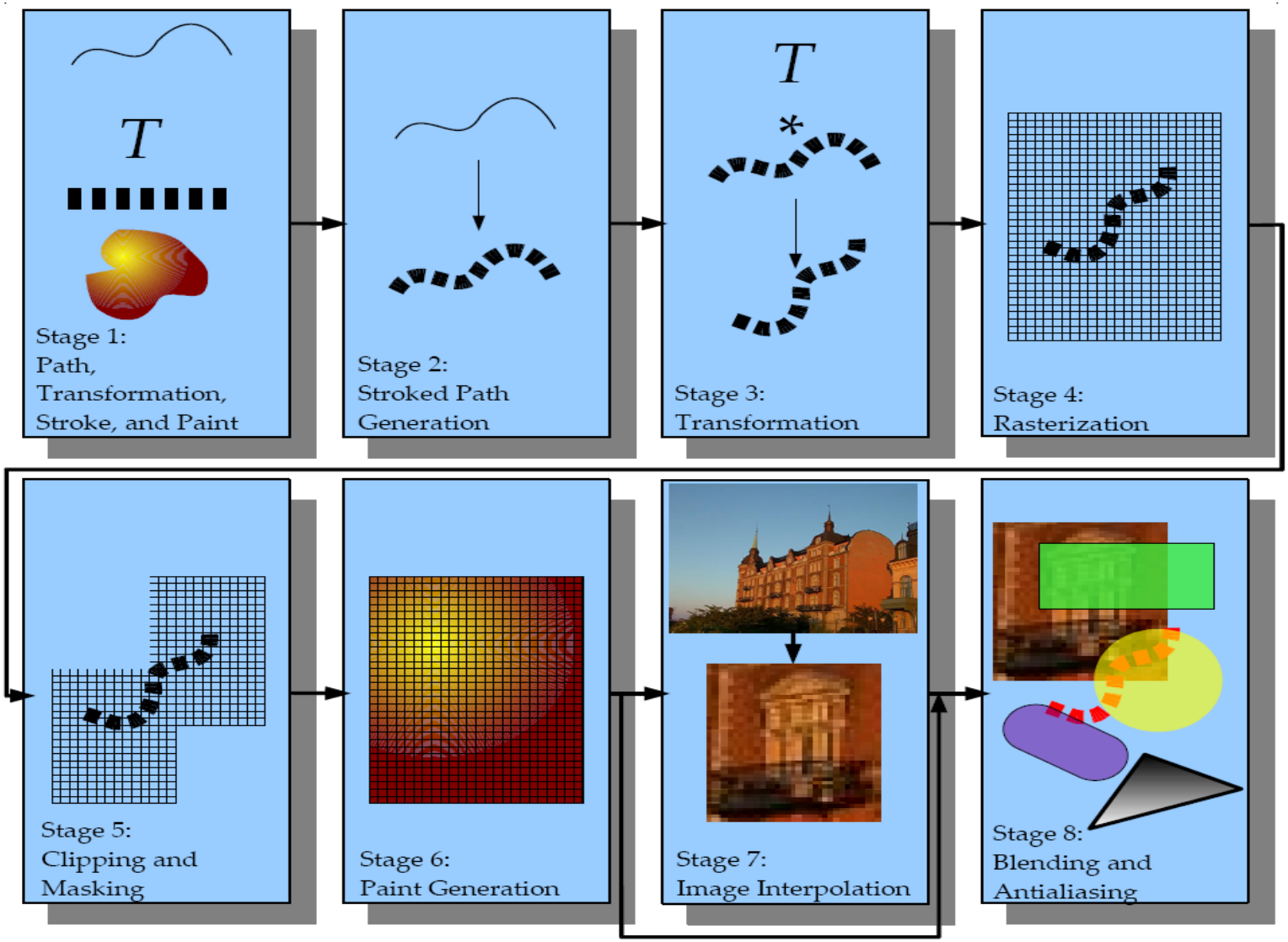
- **OpenVG is the industry's first native Bezier rendering API**
  - Enables a new class of hardware acceleration – Bezier primitives – not polygons
- **Primarily used to accelerate higher-level engines such as Flash and SVG**
  - OpenVG is an OpenGL-style, low-level API – but many artists prefer tools over programming

OpenGL ES
Accurately represents PERSPECTIVE and LIGHTING using 3D polygons

OpenVG
Accurately represents SHAPE and COLOR using 2D Bezier Curves



OpenVG Pipeline



OpenVG Adoption

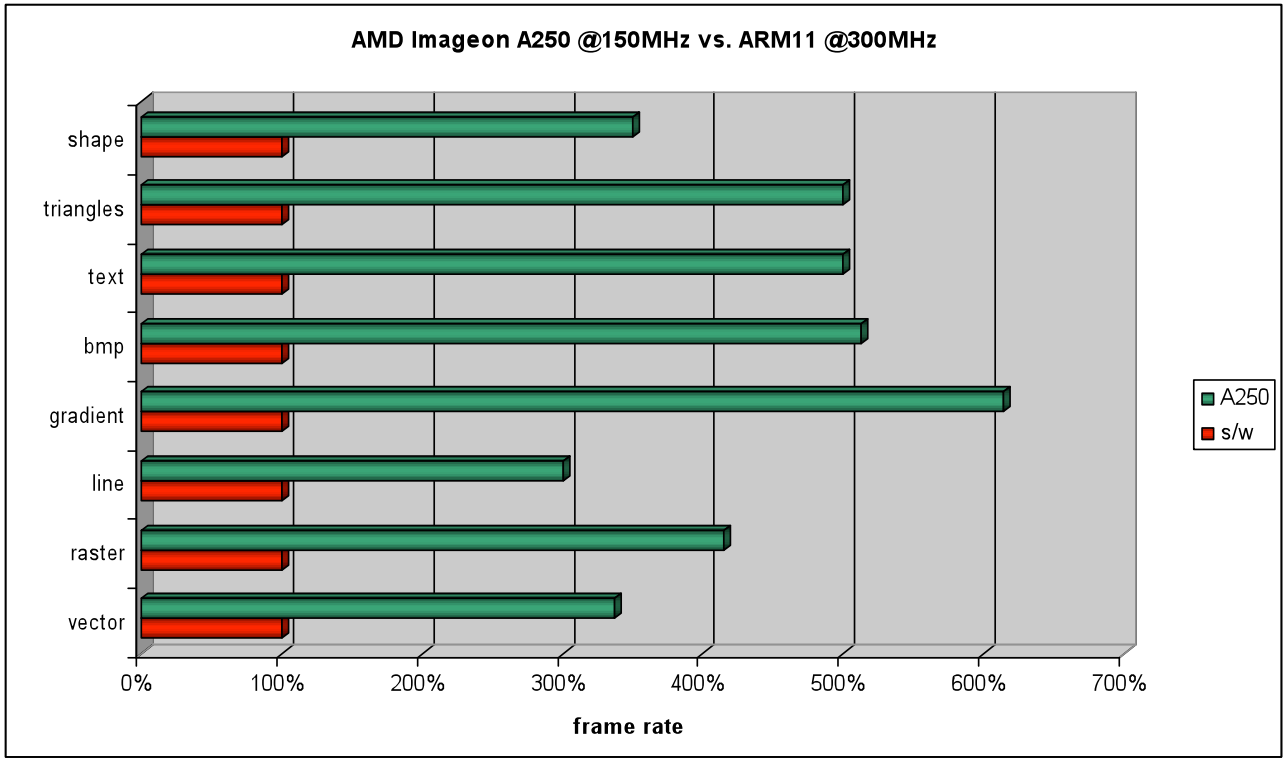
- **OpenVG 1.0 specification shipped in August 2005**
  - Open source sample implementation and conformance tests in March 2007
- **OpenVG developer momentum is growing strongly**
  - Cairo renderer in GTK+ widget toolkit, WebKit Browser engine, SVG acceleration
  - Native UI, mapping applications
- **OpenVG silicon acceleration is shipping**
  - Many graphics silicon vendors have announced support
  - 8 conformance submissions this year
- **Active work in OpenVG extensions**
  - NDS vendor extensions for advanced blending and filtering in HDTV market

OpenVG 1.1

- **OpenVG 1.1 specification publicly released at SIGGRAPH Asia 2008**
  - Together with open source sample implementation and full Conformance Test Suite
- **OpenVG 1.1 Conformance Test**
  - Comprehensive set of tests (~900 test cases)
  - Conformant implementation has to pass all of them
- **Major features**
  - Adobe Flash 7 / Flash Lite 3 support
  - Glyph API for hardware accelerated text rendering
  - Multi-sampled anti-aliasing

Flash Acceleration with OpenVG 1.1

- **Adobe is a strong participant in the OpenVG working group**
  - Helping to ensure that OpenVG 1.1 provides effective Flash acceleration
- **OpenVG support is now included in Flash Lite 3.1 engine**
  - Provides approximately 5x speed-up over software at 1/10 the power



Glyph API

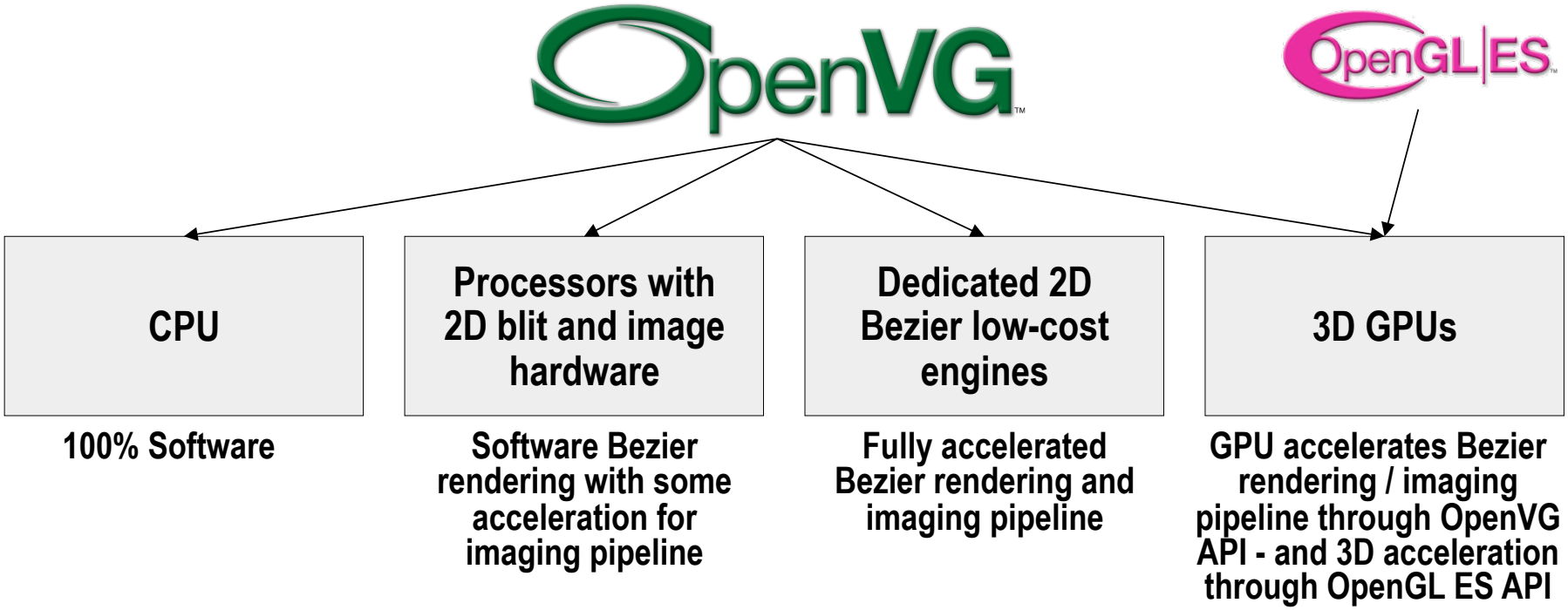
- **Many applications require high-quality scalable text**
  - User interfaces, SVG, PDF or E-book readers, GPS and mapping, web-browsers
- **Glyphs are complex 2D shapes**
  - Can be scaled down to very small sizes and minor variations in pixel coverage and / or glyph positioning may significantly affect legibility and perceived text quality
- **CPU and bandwidth-efficient commands to use cached font data**
  - Complex text layout support, including kerning, glyph variants and positional adjustments
  - Use both vector outlines and bitmap images to display characters
- **Glyph API enables text-specific rendering optimizations**
  - Auto-hinting of glyph outlines, modified filter kernels for anti-aliased text rendering
  - Missing pixel recovery for monochrome rendering mode

Unoptimized

Optimized using Glyph API

OpenVG Implementations

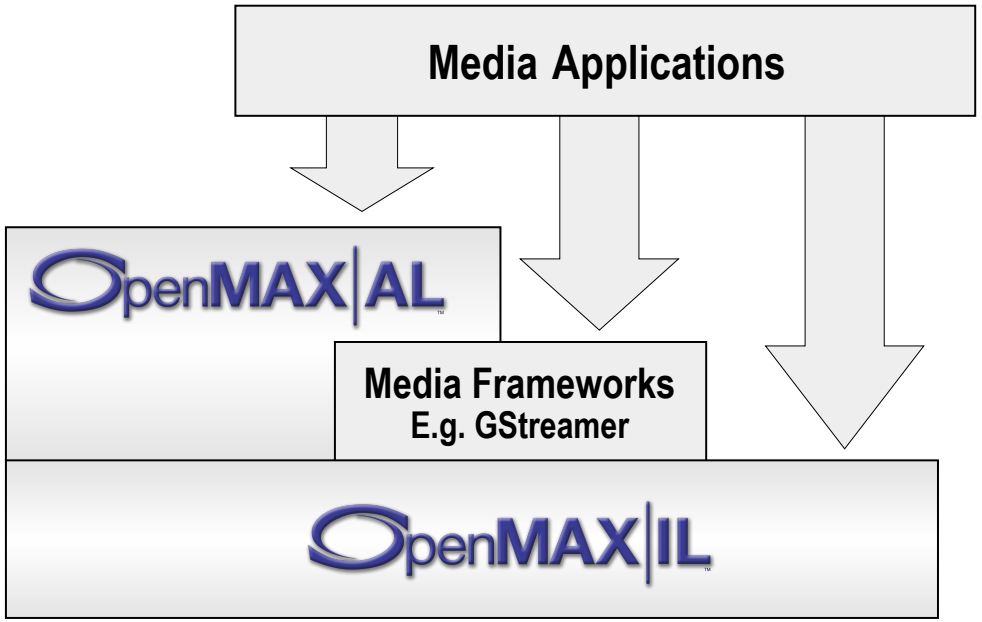
- **There are a wide range of OpenVG implementations possible**
  - Depending on the price point and target device architecture
- **3D GPUs can accelerate both OpenVG and OpenGL ES APIs**
  - With interoperability through EGL



OpenMAX Applications

- **OpenMAX AL provides fastest development and enhanced portability**
  - High-level, object-oriented abstractions for record/play applications
- **OpenMAX IL provides more flexibility and low-level hardware access**
  - For expert developers, no high-level or convenience functions
- **OpenMAX IL can be used by the native media framework**
  - Framework can be used to implement and extend OpenMAX AL

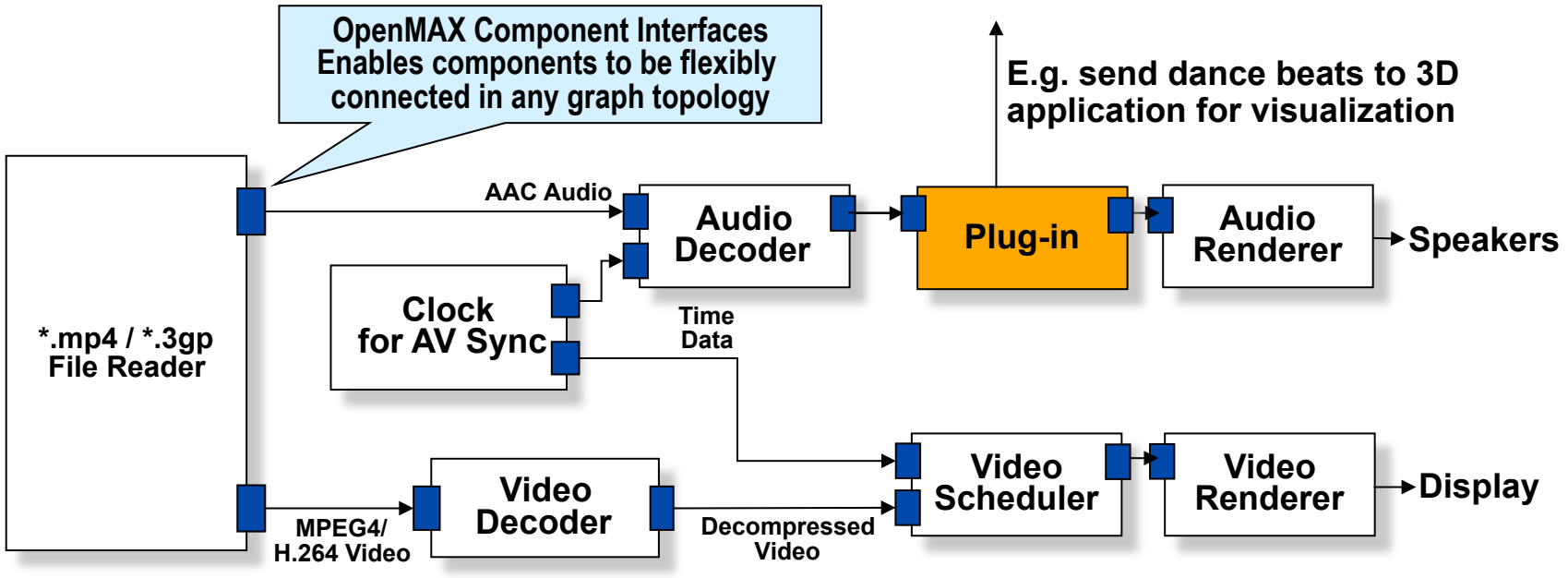
Applications may use OpenMAX AL, OpenMAX IL or the native media framework



OpenMAX AL may be implemented independently and not layered over OpenMAX IL

OpenMAX IL Example Graph

- Standardized component interfaces enable flexible media graphs
- Includes multi-stream synchronization
- Allows for custom plug-ins



Example: MPEG-4 video synchronized with AAC audio decode

Three OpenSL ES Profiles

- **Phone, Music and Game**
  - Sufficient to cover all the market segments of importance to OpenSL ES
- **Some profile overlap**
  - Mostly in basic audio functionality

Game-centric mobile devices
Advanced MIDI functionality, sophisticated capabilities such as 3D audio, audio effects, buffer handling

Music-centric mobile devices
High quality playback and record, support for multiple music codecs, streaming support

Game | **Music** | **Phone**

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

OpenSL ES

OpenMAX and OpenSL ES Priorities

- **OpenMAX IL 1.1.2 – released in September 2008**
  - Corrections and clarifications
  - Extension mechanism
- **Several extensions to OpenMAX IL 1.1 in discussion**
  - Imaging, 3D audio, content demuxer, content pipes
- **OpenMAX IL graph-level conformance tests**
  - Due for release early 2009
- **OpenSL ES and OpenMAX AL final specifications and conformance tests**
  - Due for release in first half 2009

Mobile Umbrella Specifications

- **Individual APIs define domain specific media acceleration**
  - OpenGL ES for 3D, OpenMAX for video and images etc.
- **Latest mobile applications want to MIX media types**
  - E.g. route live video into a composited 3D user interface
- **For portability of mixed-media need to define how the APIs work *together***
  - E.g. how to transfer video data from OpenMAX into OpenGL ES
- **Umbrella specs define and CONFORMANCE TEST trans-API operation**
  - Creates a reliable, cross-vendor media-stack definition

OpenKGS™

OpenKODE™

EGL

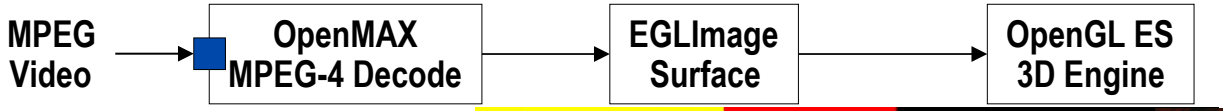
- **EGL abstracts access to rendering surfaces**
  - Interfaces Khronos rendering APIs to native platform window system
  - A derivative of the WGL Windows abstraction API
- **Emerging role as a communication hub between handheld APIs**
  - Sharing images via EGLImage extensions
  - EGLSync objects for inter-API fences and other signalling
- **Can create rendering surfaces into which multiple client APIs can draw**
  - Enables high-performance, accelerated, mixed-mode 2D and 3D rendering
  - Using OpenGL ES and OpenVG
- **EGL 1.3 was released in December 2006 – supports OpenGL ES**
  - OpenKODE 1.0 uses EGL 1.3 PLUS EGL extensions to integrate OpenVG PLUS
  - Lock Surface EGL extension for direct blitting of software rendering applications to the screen
- **EGL 1.4 integrated OpenVG and Lock Surface into core EGL**
  - In spring 2008

Directions for EGL 1.5

- **Create EGLImage within EGL**
  - Pre-declare uses => guaranteed image compatibility
- **Share images with OpenMAX**
  - Add EGLImage video data formats (YUV)
- **Stream images between APIs**
  - Queue of images with producer/consumer operations
- **EGLSync objects**
  - Inter-API fences and other signalling

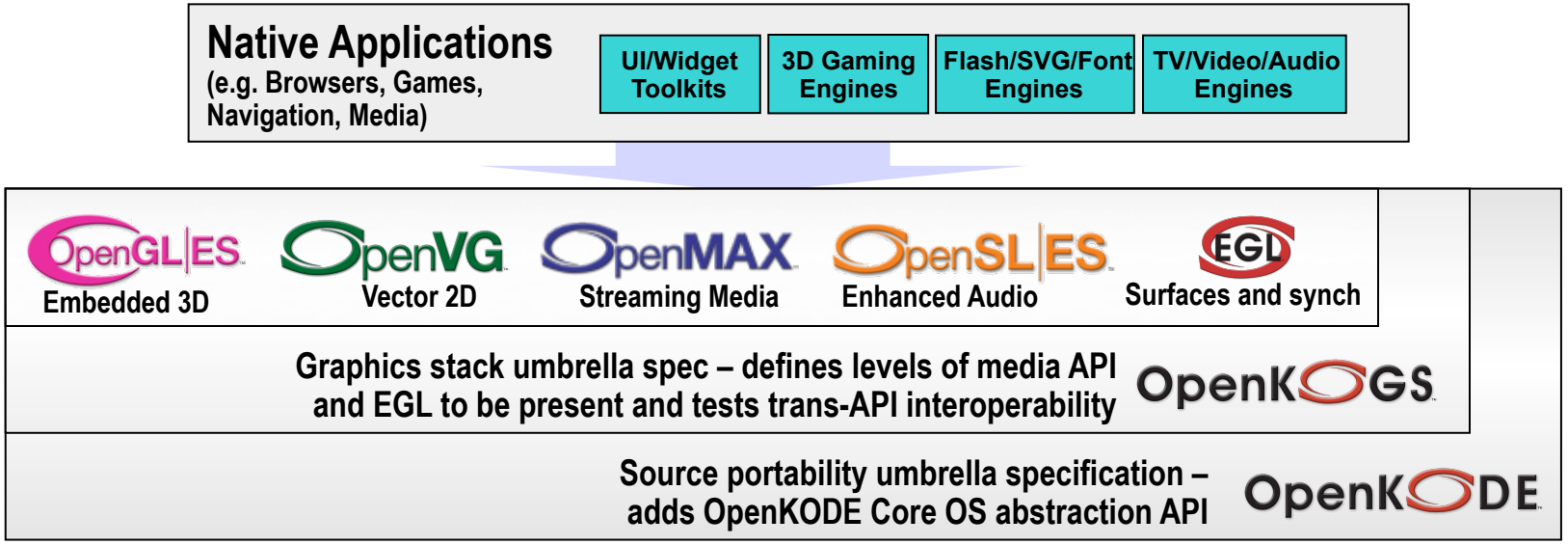
Without EGLImage the game must copy the OpenMAX buffer into an OpenGL ES texture

An EGLImage surface can be used as both the destination of the decode and a texture without copying the data



Khronos Mobile Umbrella Specs

- **Platform vendors can choose to ship more than just individual APIs**
  - Provide *conformance tested* multi-API programming platforms
- **1. OpenKOGS – integrated media stack – due in 2009**
  - Defines reliable trans-API interoperability through EGL for OpenGL ES, OpenVG, OpenMAX
- **2. OpenKODE = OpenKOGS *plus* OS abstraction API**
  - OpenKODE Core is Posix-like API for application portability across mobile operating systems



OpenKODE Core

- **OS Abstraction layer - familiar to POSIX and C Programmers**
  - "POSIX ES" – drawing on C89, C99
  - Some added functionality – events and IO functions
- **Genuinely implementable across all major mobile operating systems**
  - No small feat!
  - A lot of mobile OS knowledge encapsulated in the API
- **Threads, Events, Utilities, Math, Time and Timers, File system, Strings, Networking, Input/Output, Windowing**
  - Full functionally for rich-media applications
- **Lightweight – typically less than 100KB**
  - Adds no significant performance overhead
  - OpenKODE Core is NOT a layer between application and media APIs

OpenKODE – Source Portability

Compositing User Interfaces

Games and Game Engines

Flash and SVG Players

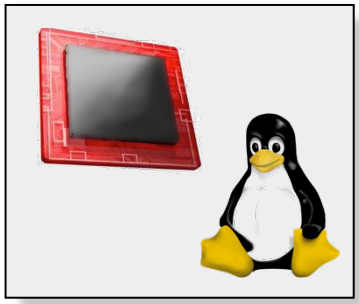
TV and Video Stacks

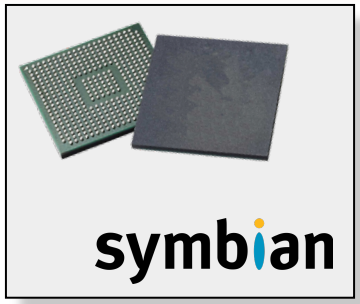
Camera Applications

Applications are portable to any OpenKODE-enabled platform

OpenKODE™ Royalty-free open standard for rich media source portability

Platforms can use any OS and silicon vendor





Raising 2D and 3D Visual Quality

State-of-the-art APIs enable compelling consumer displays
Advanced functionality, fast interactivity and extremely high quality

High-quality 2D graphics and test using OpenVG

Video and image processing with OpenMAX

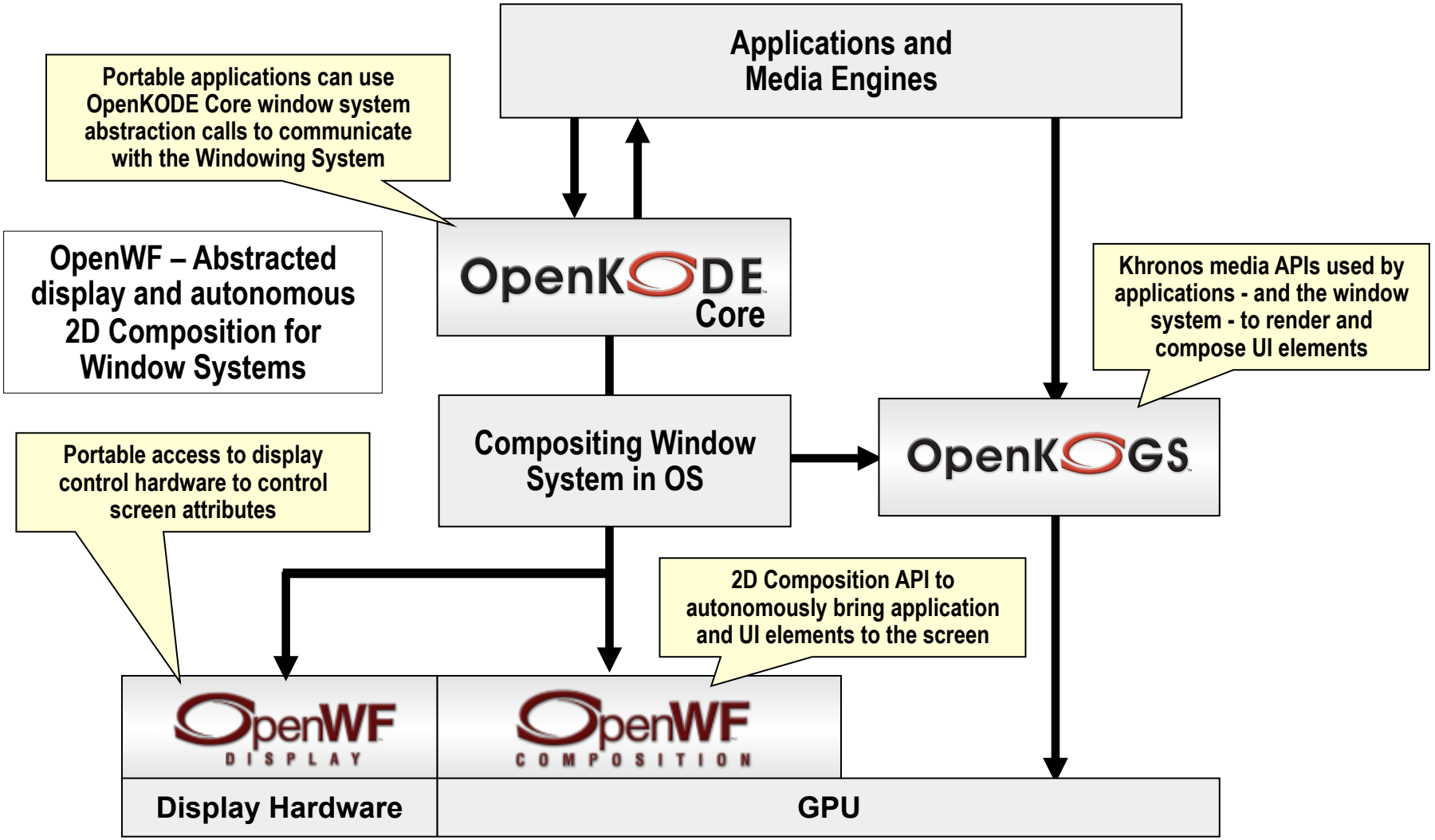
High-quality 3D displays using OpenGL ES

Older generation APIs
Provide rudimentary graphics functionality and quality

Composition and User Interface

- **The iPhone's success has demonstrates the importance of good UI**
  - Ease of discovery and use of device capabilities and resources drives revenue
- **Strong focus on using mobile graphics acceleration for advanced UI**
  - Accelerated rendering, window management and composition
- **Accelerated composition**
  - Bringing multiple screen elements together with complete flexibility on the display
- **Multiple Khronos initiatives underway to enable composition**
  - OpenWF – for autonomous, hardware-based 2D composition
  - KDWM – OpenKODE extensions to enable 3D composition using OpenGL ES

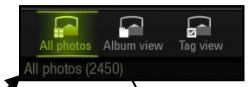
OpenWF – Windowing Foundation



OpenKODE Screen Composition

Client applications can behave as if they are writing directly to the display and render into off-screen, cross-process texture surfaces *using standard OpenGL ES, OpenMAX, EGL and OpenKODE calls*

Off-screen Buffers



Application

Application

Application

Compositor

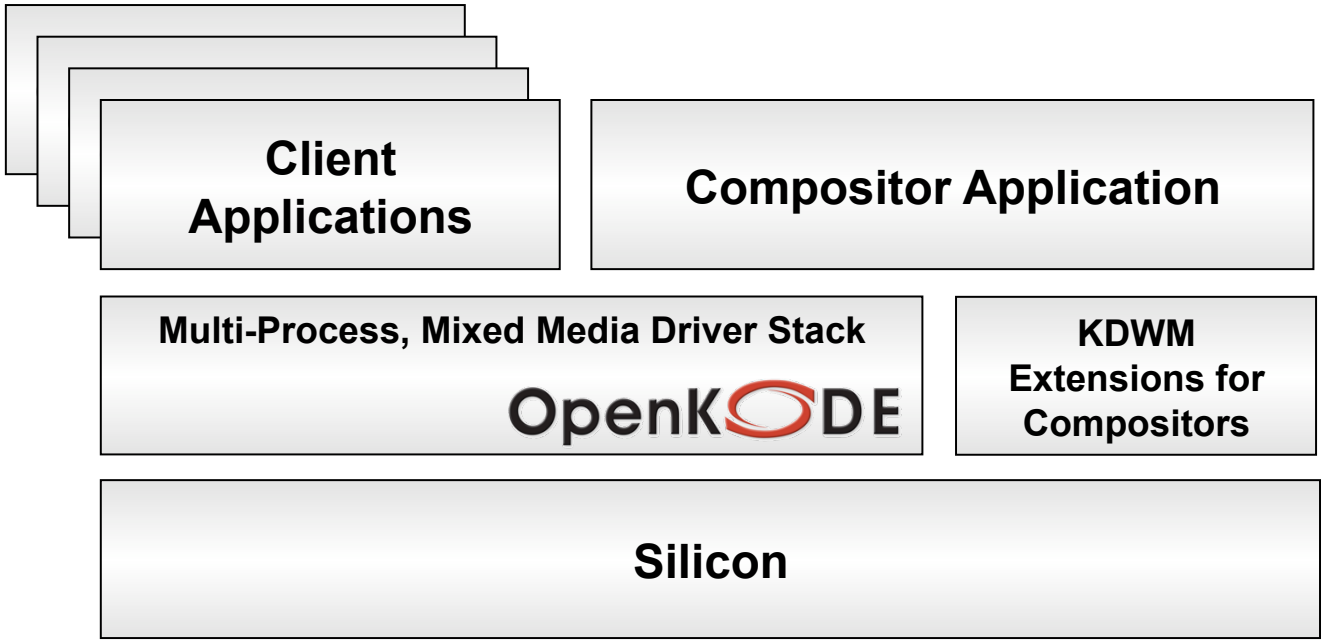
KDWM extensions to OpenKODE Core enables Compositor applications or window systems to communicate EVENTS and off-screen BUFFER HANDLES to client applications

Multiple off-screen surfaces are composited to the screen using OpenGL ES for flexible 3D screen layout and transitions

Physical Display

3D UI Composition Initiative

- **KDWM – "the Compositors OpenKODE"**
  - Extensions to OpenKODE Core - based on existing EGL and OpenKODE mechanisms
  - Off-screen surface allocation and cross process surface handle communication
  - Cross-process event dispatch
  - Proposed by NVIDIA as OpenKODE Core extension



Why Are Khronos Standards Key?

- **"Foundation Level" APIs to enable software to effectively use silicon**
  - State-of-the-art, fundamental functionality needed on every platform
  - Non-proprietary – application portability across many platforms
- **Architected to be cost and power efficient**
  - Ideal for embedded markets
- **Hundreds of man years invested**
  - Beyond any single company now to produce specifications of this breadth and depth
  - Leverage the investment of the silicon industry in creating standards-based drivers sets
- **Royalty-free**
  - Khronos is committed to generating market opportunities for its members and the industry
- **More information and slides at www.khronos.org**

KHRONOS GROUP