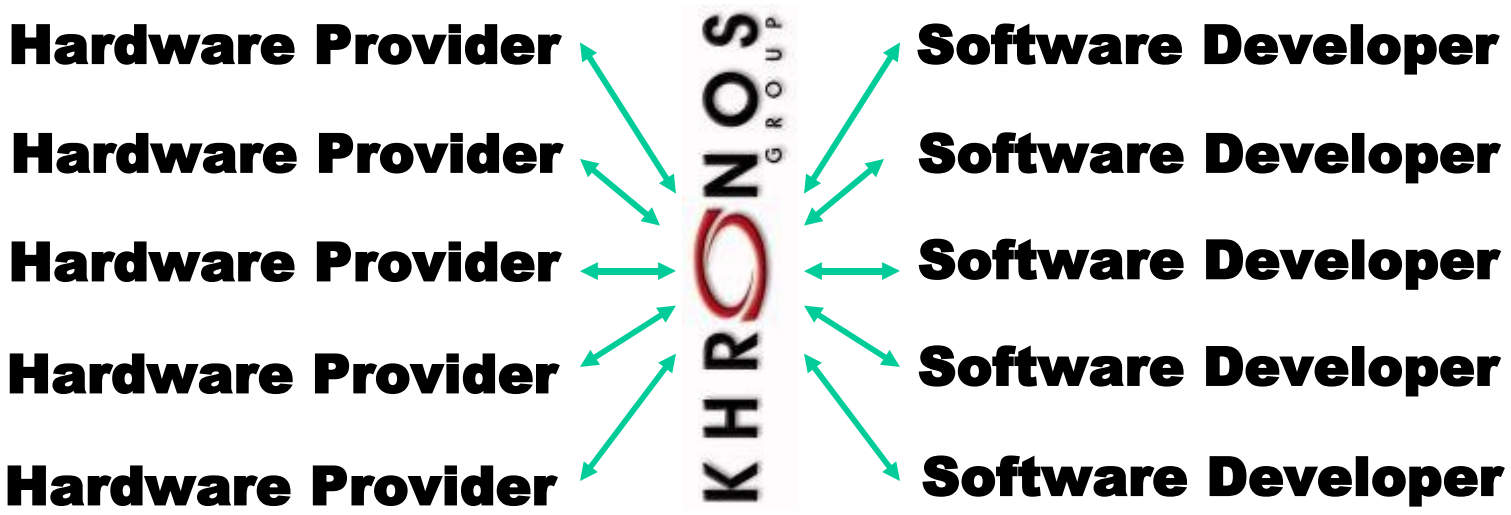
Creating the Embedded Media Processing Ecosystem

Ed Plowman
Graphics Product Manager, ARM
Board Of Promoters, Khronos
Founding Member, OpenGL ES Working Group

Media APIs Enable Market Growth

- An API is a contract between hardware and software worlds
  - Enabling both - everyone wins
- ISVs see reduced variability across multiple platforms
  - More software can reach market faster at a better level of functionality and quality
- Hardware vendors can accelerate many applications
  - Adding value to their platform



Khronos develops "Foundation-Level" APIs
As close-to-the-metal" as possible while providing portable access to hardware acceleration. Great performance. Good foundation for higher-level engines and middleware

KHRONOS GROUP

Creating Open API Standards

Open Membership
Any company is welcome
Funded by membership dues - $5K / year

Open Standards
Publicly available on web-site
Royalty-free

KHRONOS GROUP

Open Standard Platform for Embedded Rich Media Acceleration

Cross Platform
Enabling diverse handheld and embedded markets

Promoting Ecosystem
Conformance tests, tools, developer materials and outreach

Khronos has a PROVEN reputation for the TIMELY creation of HIGH-QUALITY, ROYALTY-FREE standards

How Does Khronos Make Money?

- **It doesn't!**
- **Khronos is purely a non-profit organization**
  - Funded by member dues – to cover costs
- **Our members make money by selling PRODUCTS enabled by standards**
  - NOT trying to charge for the standard itself

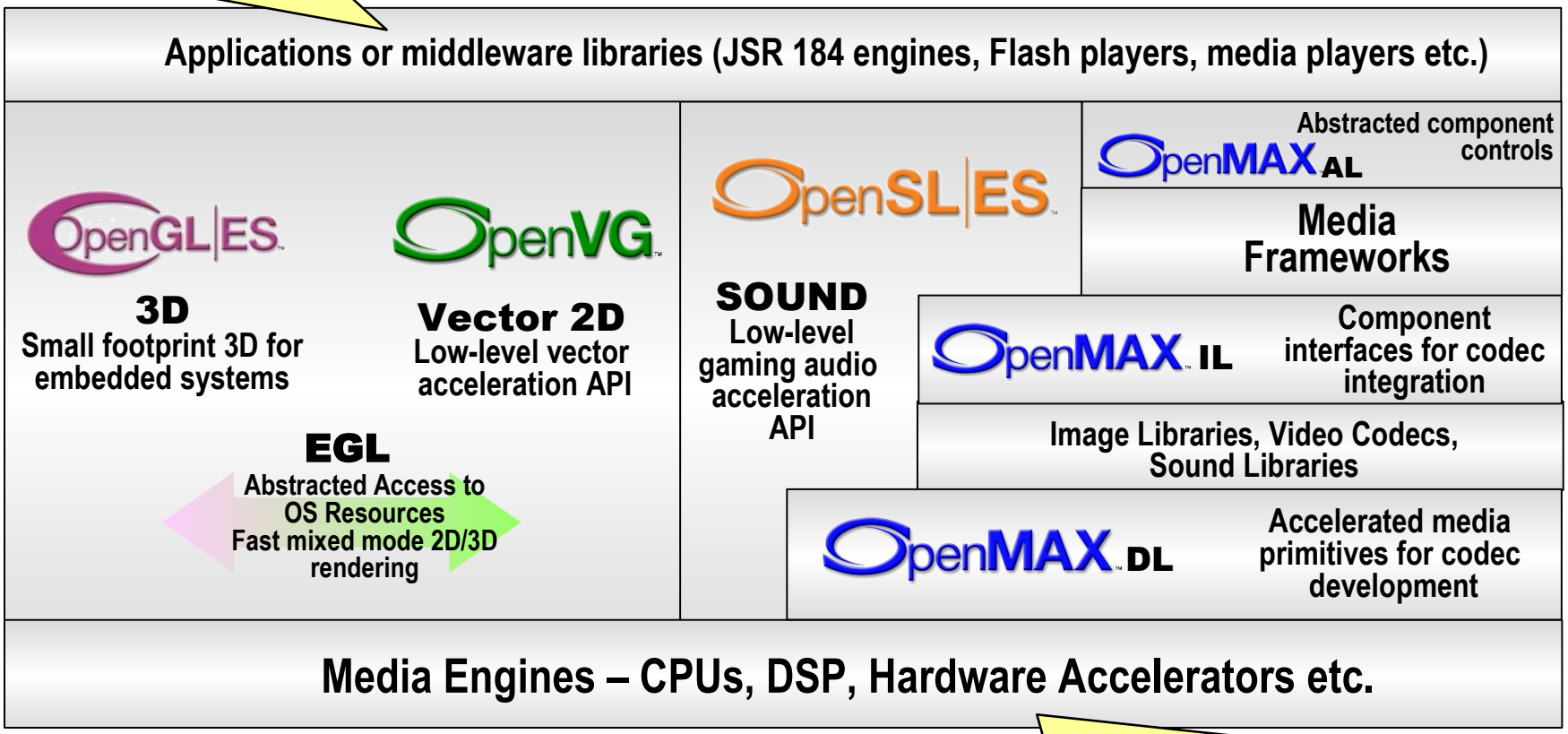
An open, royalty free API standard creates much larger market opportunities

Selling an API would generate relatively small amounts of revenue

Our members cooperate to create standards – and compete in the marketplace with products that use Khronos standards

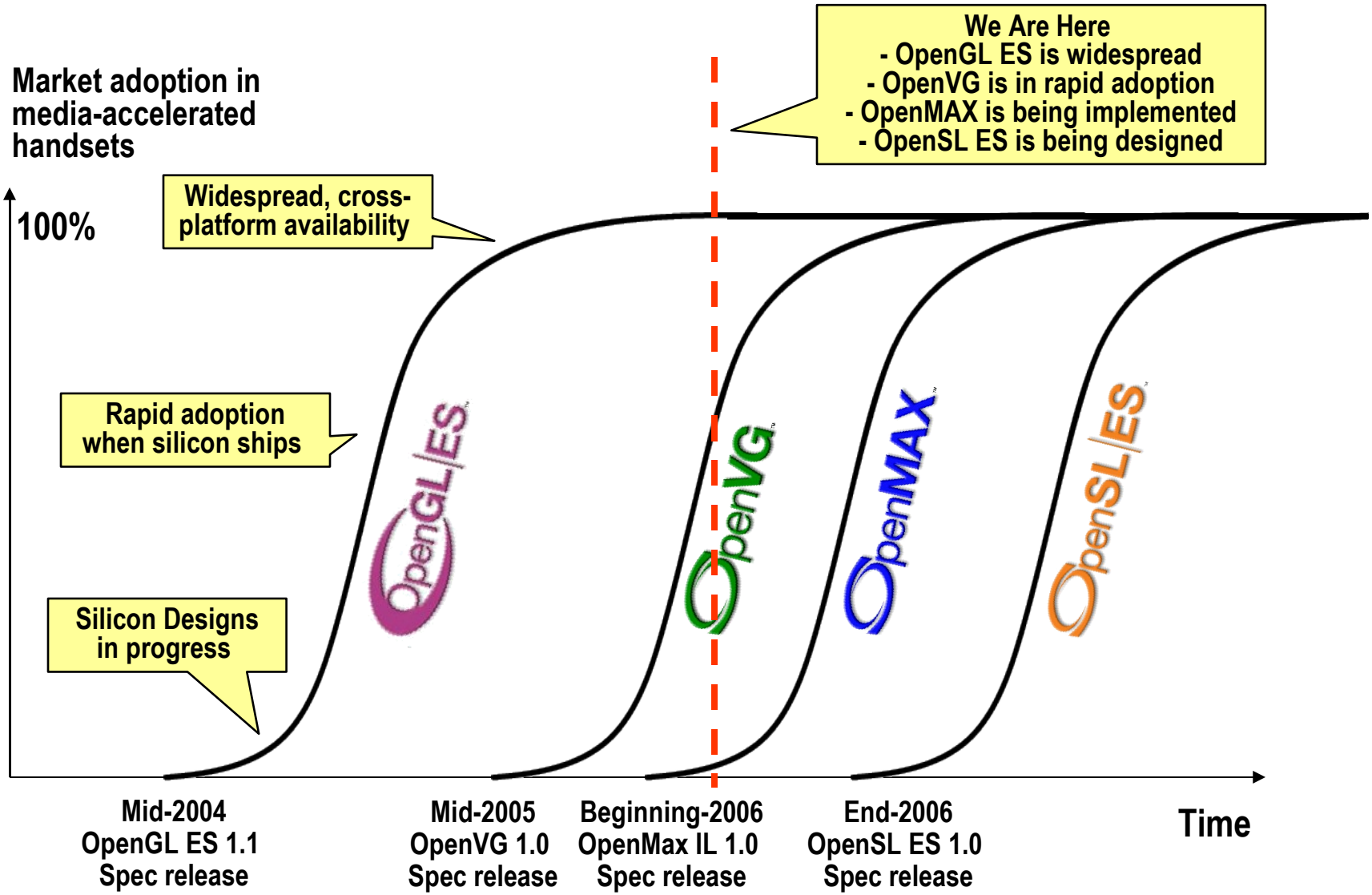
Khronos API Stack

The Khronos API family provides a complete ROYALTY-FREE, cross-platform media acceleration platform



Khronos defines low-level, FOUNDATION-level APIs.
"Close to the hardware" abstraction provides portability AND flexibility

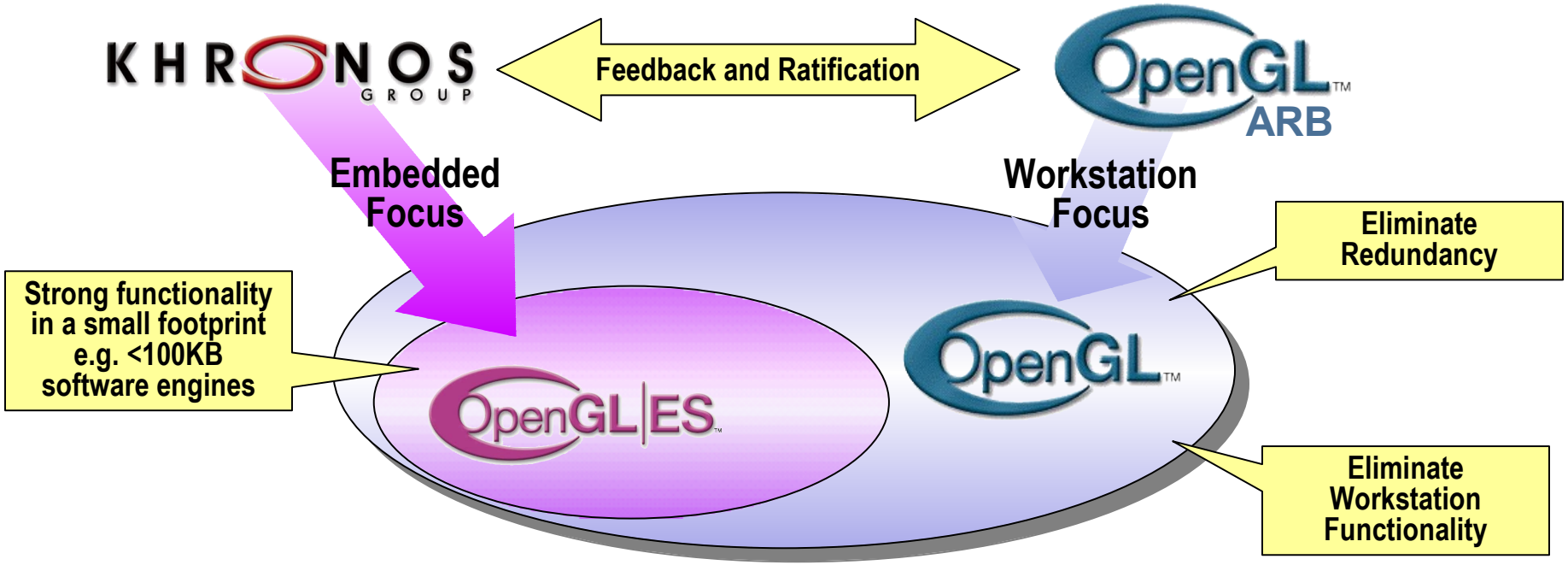
Adoption of Khronos APIs



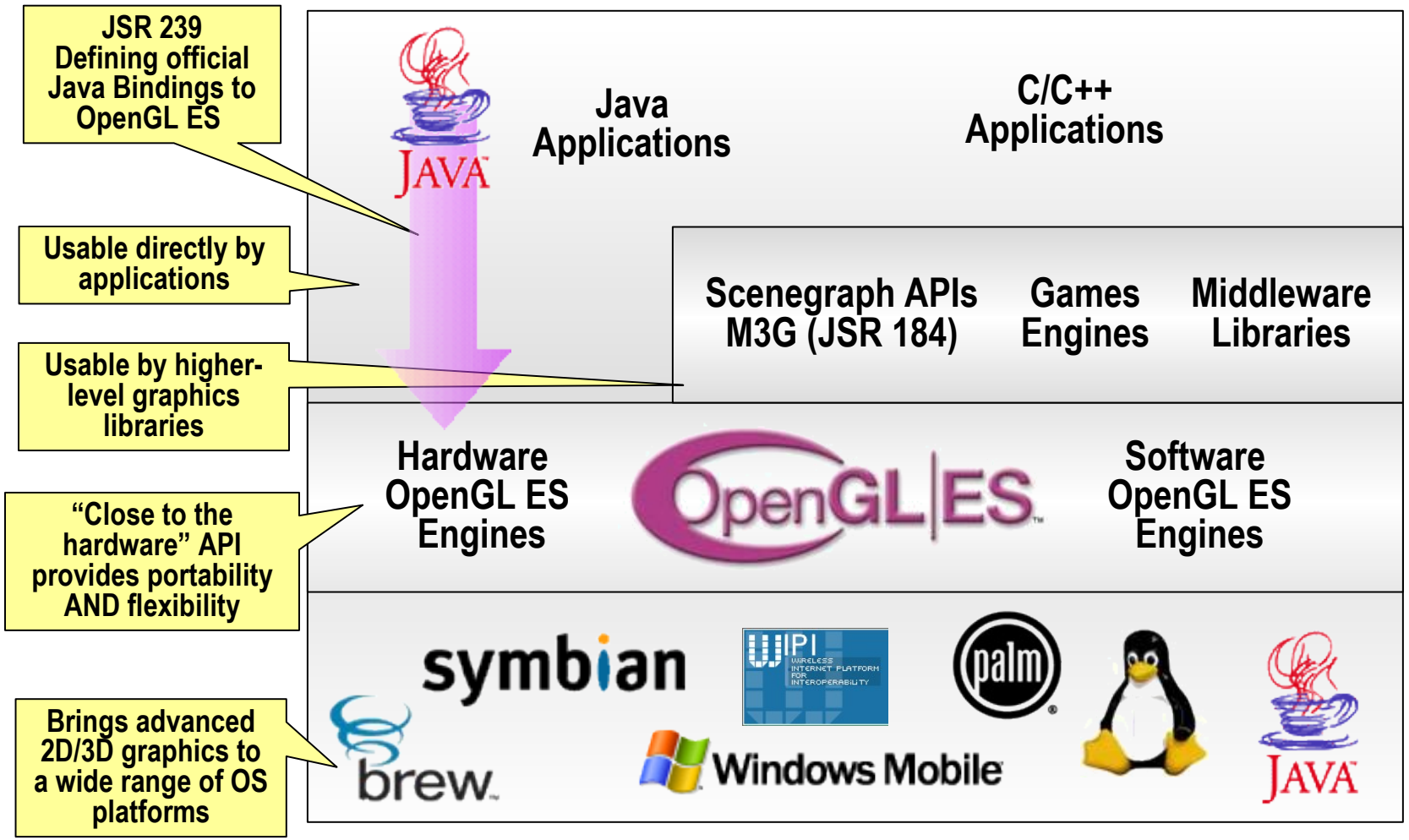
Industry Standard Embedded 3D

OpenGL ES API Standard

- **Small-footprint subset of OpenGL**
  - Created with the blessing and cooperation of the OpenGL ARB
- **Powerful, low-level API with full functionality for 3D games**
  - Available on all key platforms
- **Fully extensible**
  - Enables vendor differentiation and exploration of new functionality



OpenGL ES – Central to Embedded 3D



API Must Evolve at the Right Speed

Not too fast to prevent widespread adoption

Fast enough to encourage and expose new capabilities

OpenGL ES 1.0
3D running in software on CPU

OpenGL ES 1.1
Enhanced 3D running on fixed-function hardware

OpenGL ES 2.0
3D shaders running on programmable hardware –

The need for a standard every 12 months is decreasing. The move to programmability needs careful management

Shipping Products

2004/5 2005/6 2006/7

Shaders: Next Generation Mobile 3D

- **Graphics industry is in the middle of a programmable revolution**
  - Shader programs running on the GPU are enabling amazing new visual effects
- **Graphics APIs will need to support shading languages**
  - Enabling new visual effects to be created by developers

3D Today
Fixed Functionality

3D Tomorrow
- Shader Programmability

Doom 3's Zombies

Unreal's Rocks

Halo's Ice

Far Cry's Water

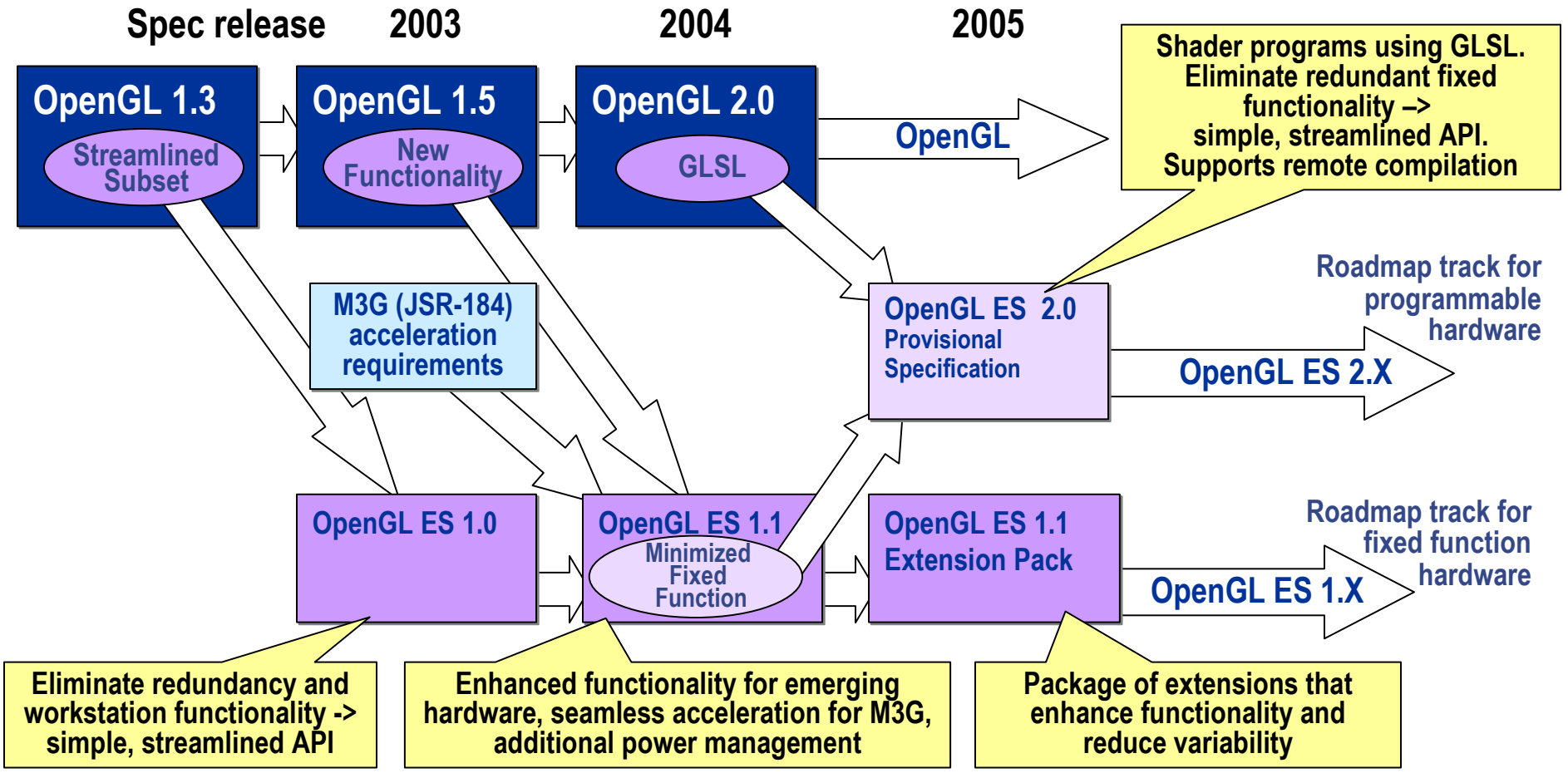
OpenGL ES – Two Track Standard

- **Two tracks - manage mobile graphics through programmable transition**
  - With maximized portability and minimized platform costs
- **OpenGL ES 2.0 ruthlessly eliminates redundancy – just like 1.X**
  - Deprecates all fixed functionality that can be replaced by shaders
  - Significant reduction in engine cost and driver complexity
- **Platforms can ship either or both 1.X and 2.X libraries**
  - Cheaper, more flexible than one large driver with both fixed and programmable functions
  - With full backwards compatibility maintained in each track
- **OpenGL ES 2.X does NOT replace OpenGL ES 1.X**
  - Will always need lowest cost, non-programmable hardware for certain high-volume devices

OpenGL ES 1.X – Fixed Function Acceleration
OpenGL ES 1.1
- For software and fixed functionality hardware
- All 1.X specifications are backwards compatible

OpenGL ES 2.X – Programmable Acceleration
OpenGL ES 2.0
- Vertex & pixel shaders through GLSL ES shading language
- All 2.X specifications will be backwards compatible

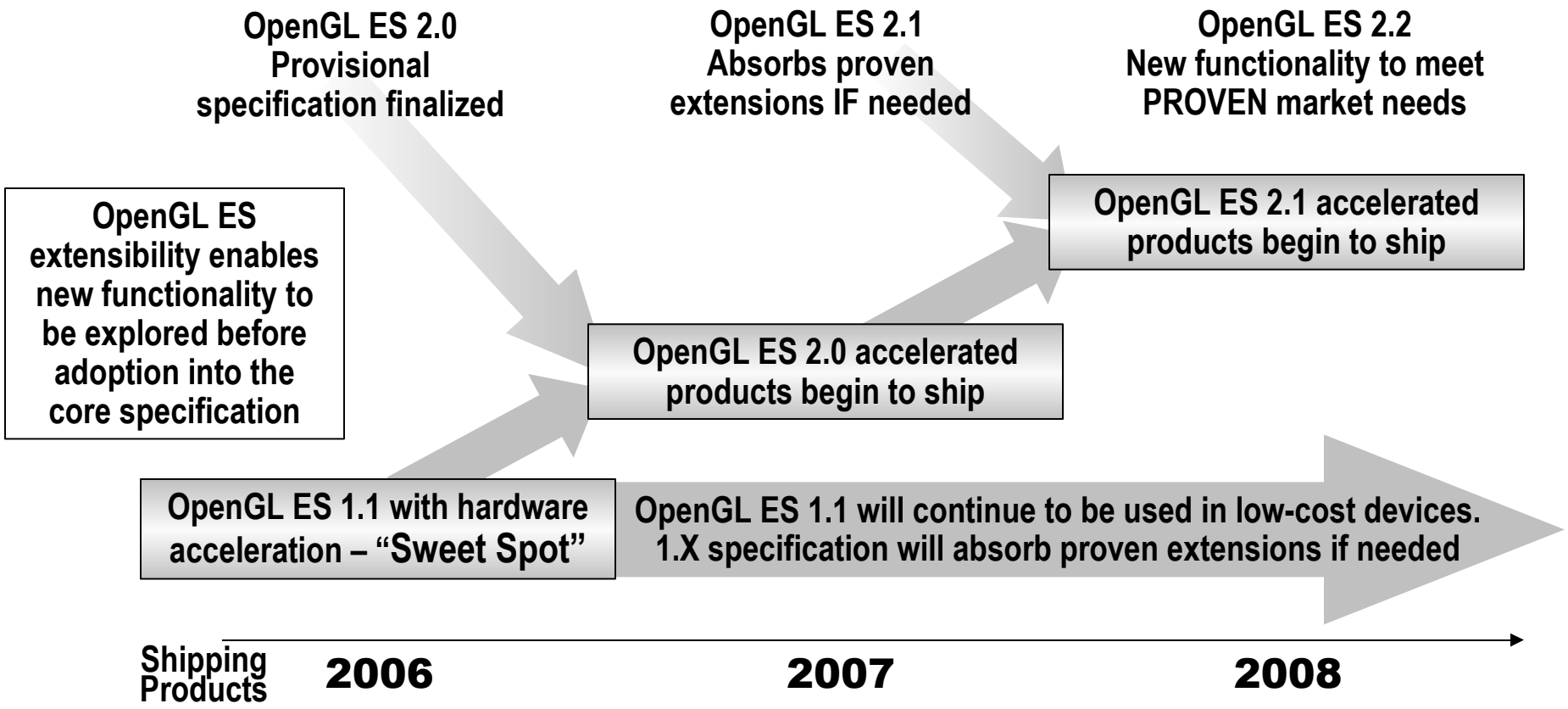
OpenGL ES DNA



Minimize differences between 1.X and 2.X tracks to ease programmable transition

OpenGL ES Roadmap

- **Stability and reducing fragmentation is currently the key concern**
  - More important than new functionality in the current phase of market development
  - Industry is still absorbing and implementing current OpenGL ES specifications



Mobile 3D Graphics Ecosystem

Synergistic Development of authoring and acceleration standards under one body. Third party tools and debuggers. gDEBugger ES announced at GDC 2006

KHRONOS GROUP

Market demand for Handheld 3D

Great 3D Applications

graphicREMEDY gDEBugger

Industry Cooperation

Tools

COLLADA

Khronos drives OpenGL ES roadmap to meet market needs

OpenGL|ES

Open API Standards

Benchmarks

FUTUREMARK CORPORATION

JBenchmark

Conformance Tests

High-quality platforms

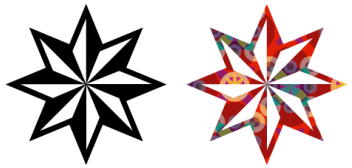
OpenGL ES 1.1 Conformance Tests released Aug'05 with peer review conformance process. Working Group is tightening conformance and Khronos is developing OpenGL ES 2.0 tests

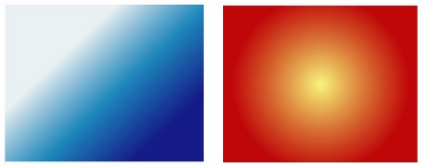
Futuremark 3DMarkMobile06 JBenchmark 3D. Soon see consumer benchmark interest as on the PC?

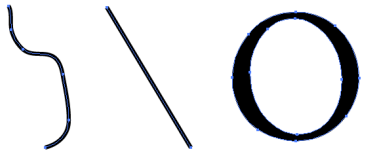
Accelerated Vector Graphics

Vector Graphics for Embedded Devices

- **Vector graphics used in many popular formats such as Flash, SVG, PDF**
  - High-quality user interfaces, screen savers, 2D Games
  - Portable mapping and GPS applications, E-book readers and text packages
- **High-quality 2D vector graphics use scalable Bezier curves**
  - Path based for scaling and positioning at full quality - not polygon based
  - Easy porting of full quality 2D content to different screen sizes
- **But all 2D vector graphics usually run un-accelerated!**
  - Works OK on high-performance PCs
  - Not effective on low-powered handset CPUs
- **We need to enable accelerated mobile 2D vector graphics!**





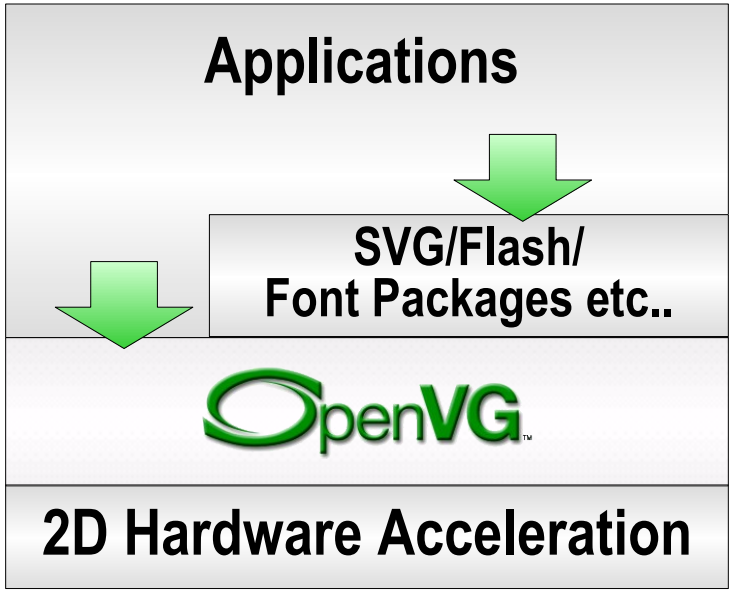


KHRONOS GROUP

OpenVG

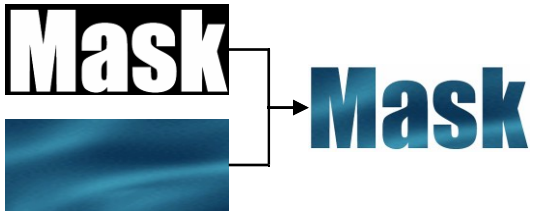
OpenVG – Accelerated Vector Graphics

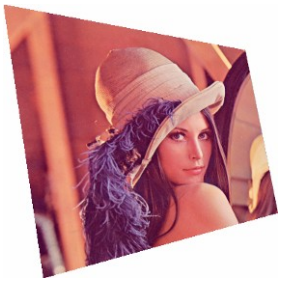
- **OpenVG ACCELERATES existing formats – such as Flash**
  - NOT a competitor to Flash, SVG etc.
  - Enables popular vector formats to run with faster performance and less power
- **OpenVG 1.0 released at Siggraph 2005**
  - Open, royalty-free standard
  - Developed in just 12 months
- **Uses OpenGL-style syntax**
  - Easy to learn for OpenGL developers
- **Will be supported by graphics silicon**
  - Dedicated 2D engines AND full 3D engines
- **Conformance tests under construction**

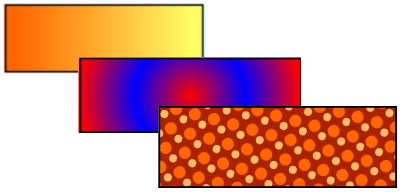


OpenVG Features

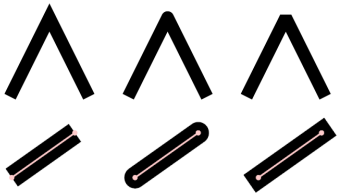
- **Core API**
  - Coordinate Systems and Transformations (Image drawing uses a 3x3 perspective transformation matrix)
  - Viewport Clipping, Scissoring and Alpha Masking
  - Paths - Line, Arc, Cubic Bezier, Quadratic Bezier, Stroked Path, Patter fill, gradation, Anti-aliasing, Path Interpolation
  - Images
  - Image Filters
  - Paint (gradient and pattern)
  - Blending

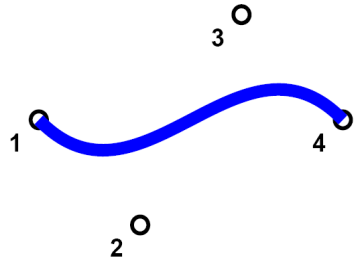




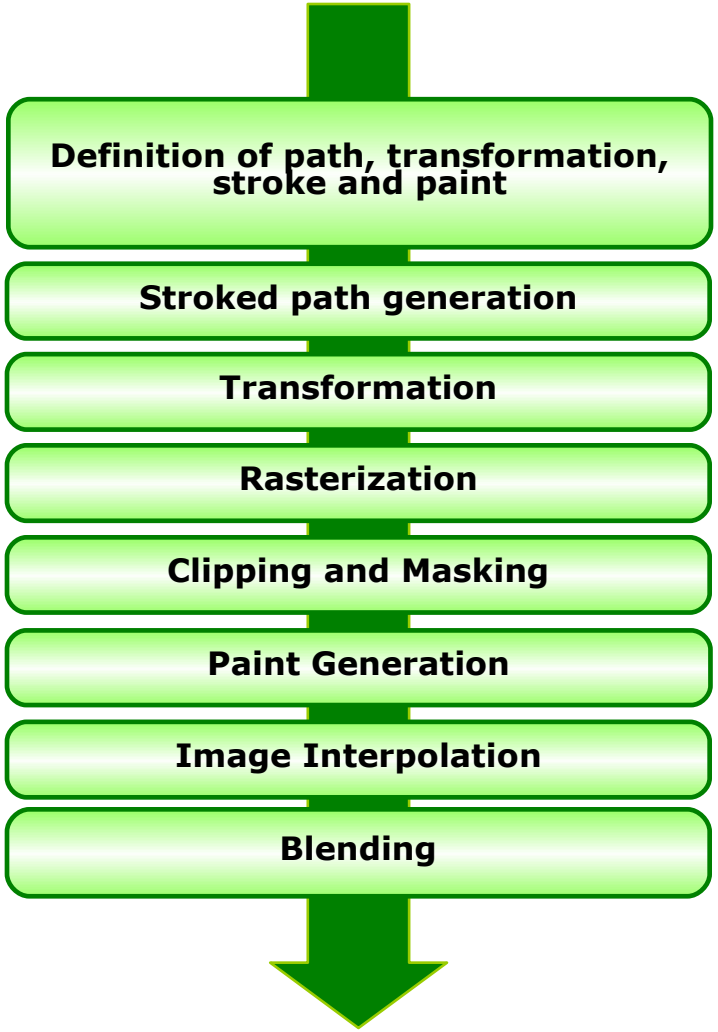


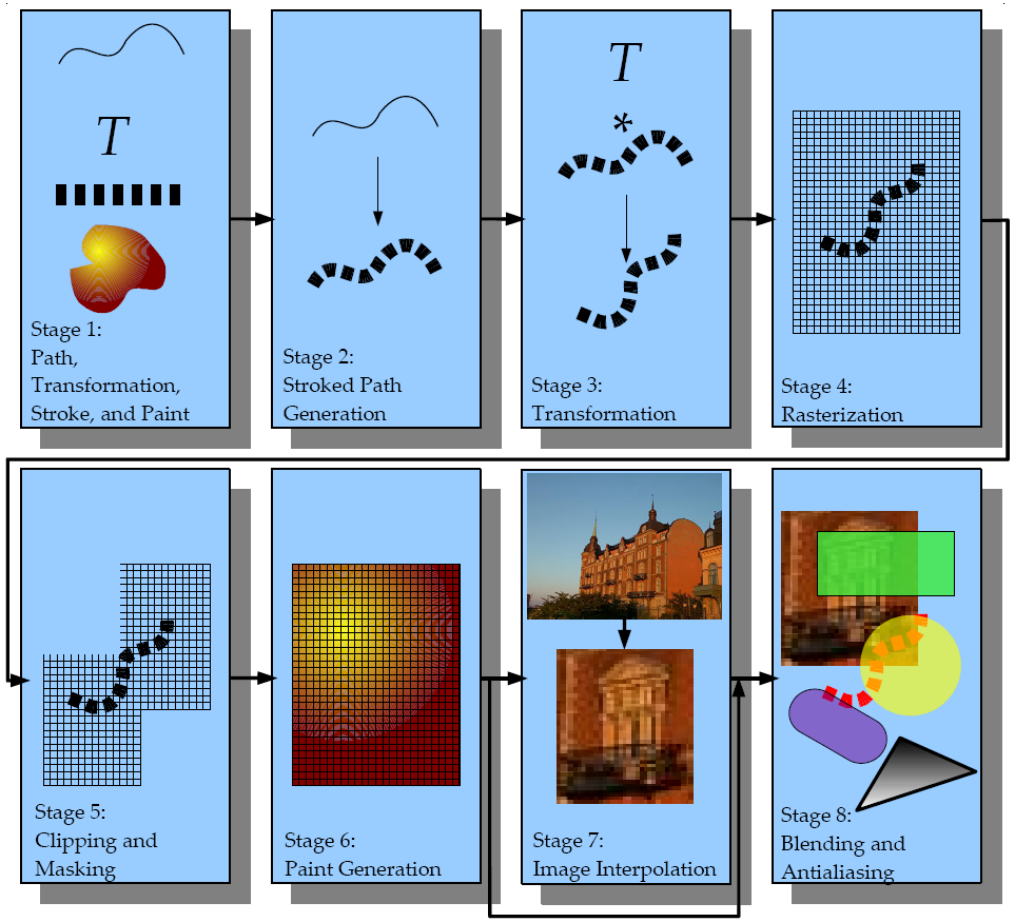
- **The VGU Utility Library**
  - Higher-level Geometric Primitives
  - Image Warping





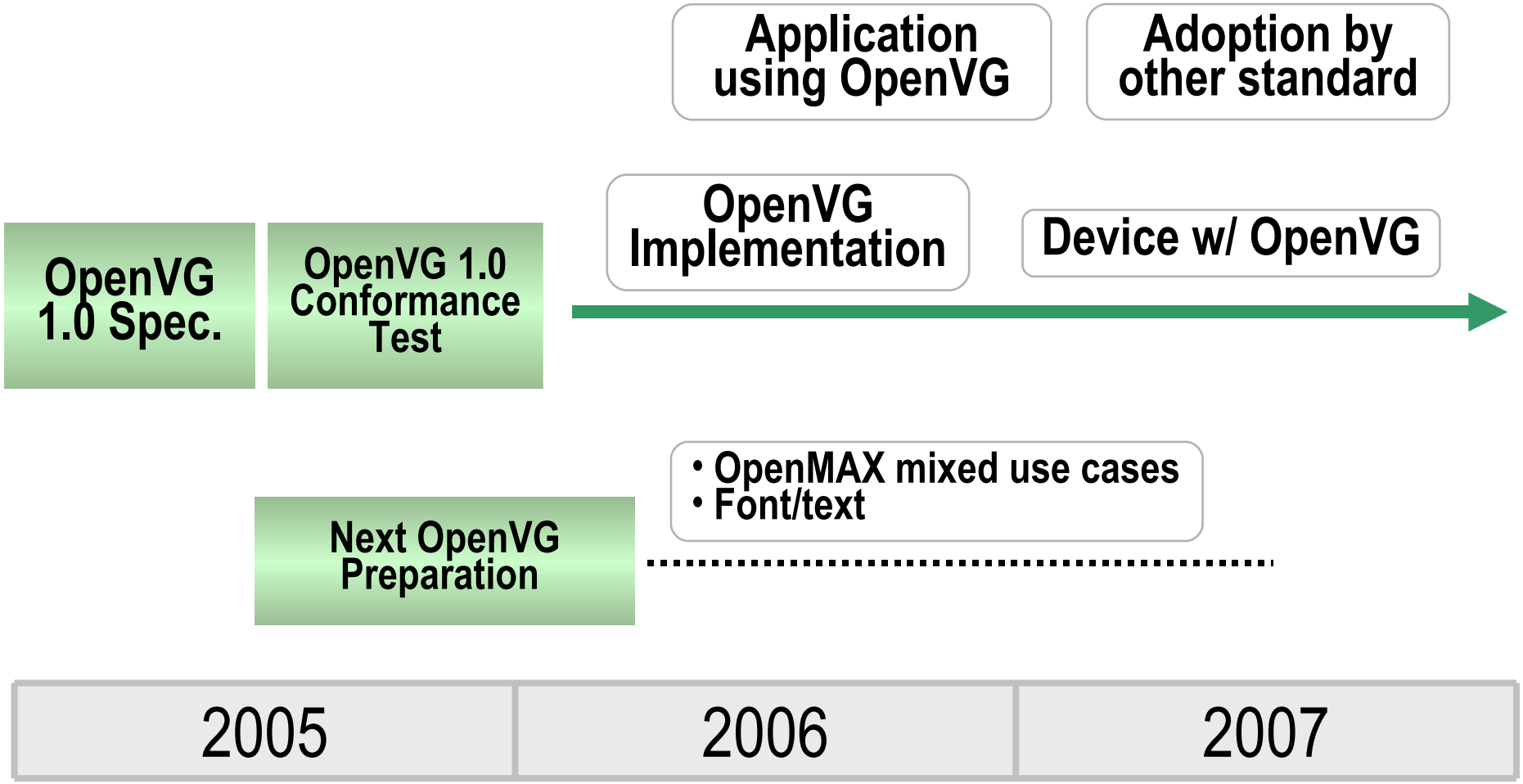
OpenVG Pipelines





OpenVG Roadmap

- Evolution continues!



EGL – Bringing it all together

OS abstraction, buffer and context management

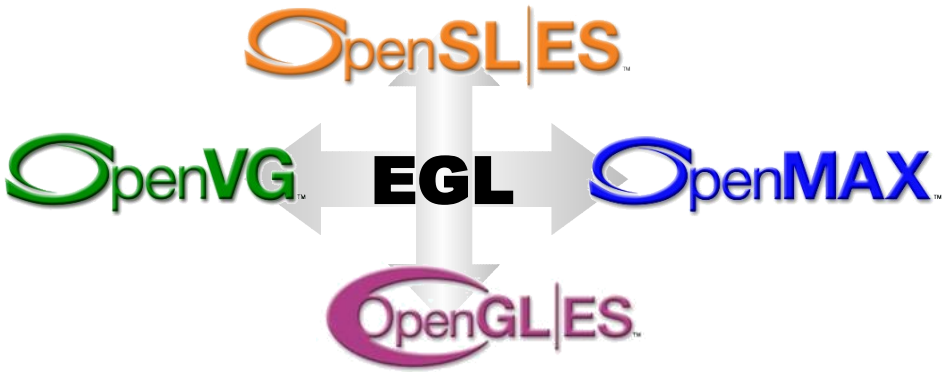
EGL – Central to OpenKODE

EGL is evolving into a central resource and display manager to enable sophisticated mixed-mode operation between Khronos APIs

EGL 2.0?
Considering adding FBO-based constructs for sophisticated multi-client buffer sharing with asynchronous / isochronous event handling.
Display compositing?

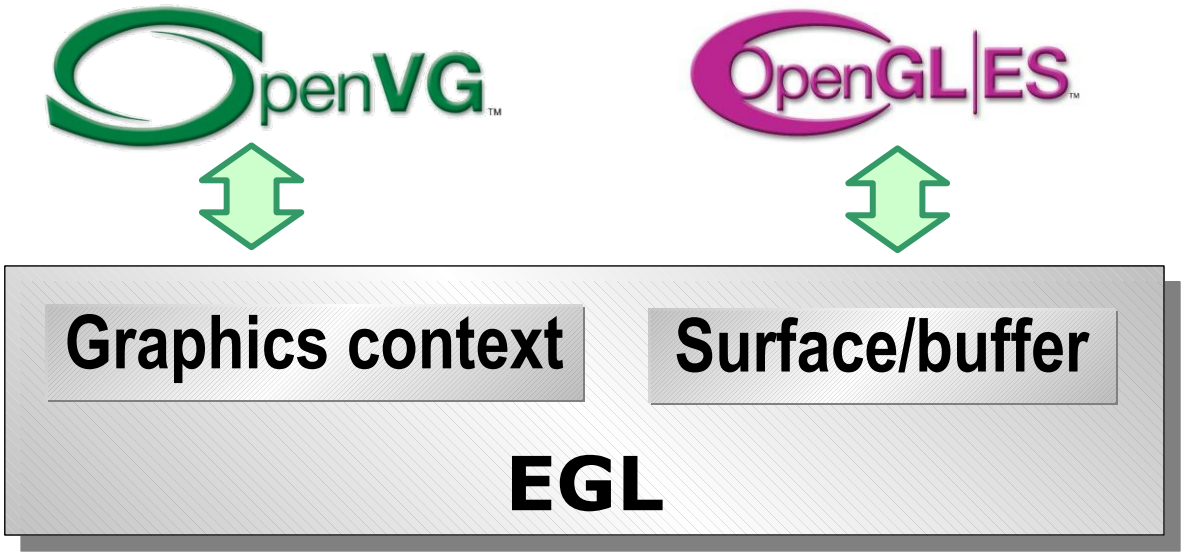
EGL 1.2
Added integrated buffer management for high-efficiency mixed mode rendering between OpenGL ES and OpenVG

EGL 1.0 / 1.1
Abstracted resource and display management for OpenGL ES.
Derived from WGL



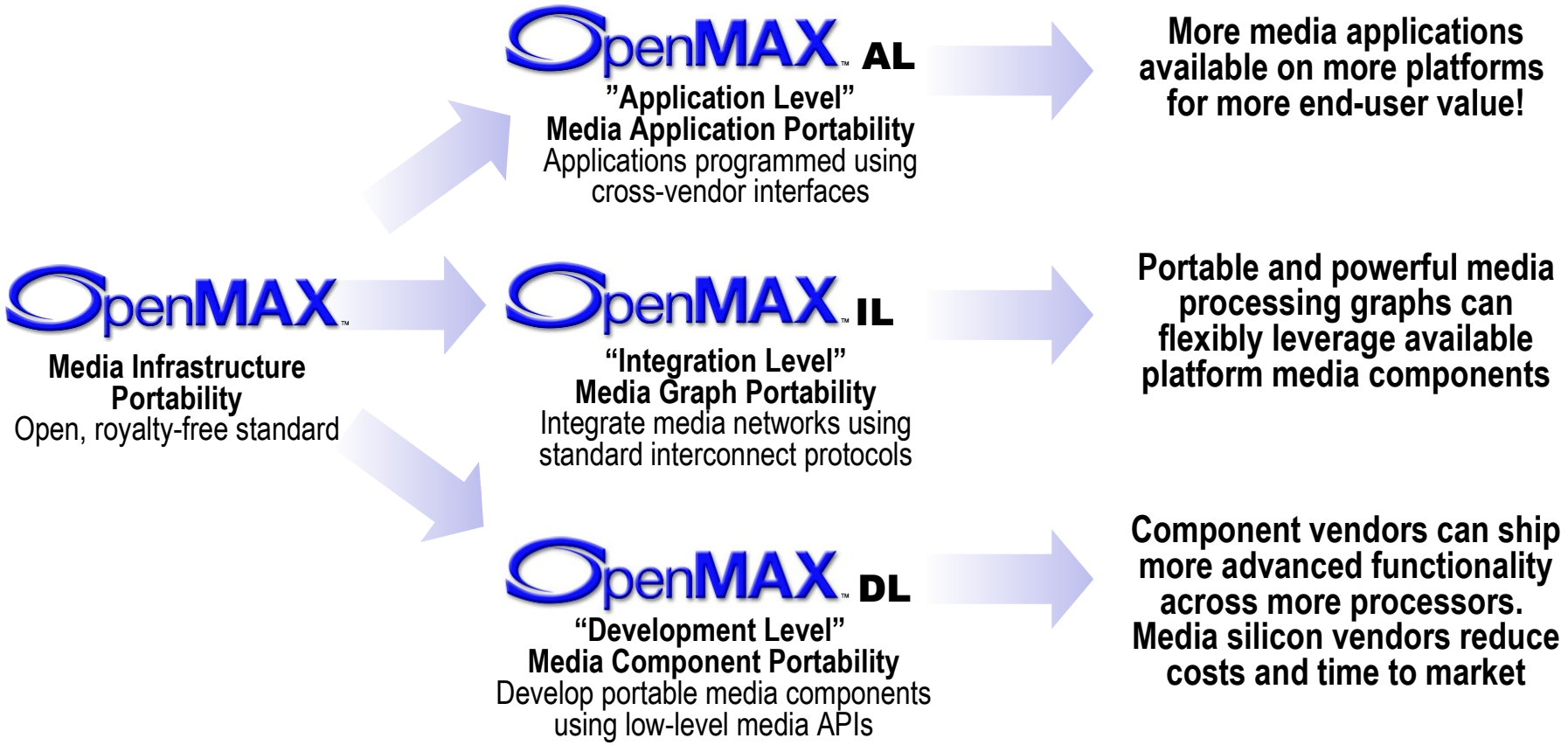
EGL encompassing more Khronos APIs

- **EGL– portable layer for graphics resource management**
  - Graphics context management
  - Surface/buffer binding
  - Rendering synchronization
- **The new EGL 1.2 supports OpenVG**
- **Use cases of OpenVG + OpenGL ES**
  - OpenVG renders front/back ground of OpenGL ES application (surface sharing)
  - OpenVG renders on OpenGL ES texture buffer
  - OpenGL ES renders on OpenVG image buffer



Streaming Media Portability

OpenMAX - Three Layer Solution



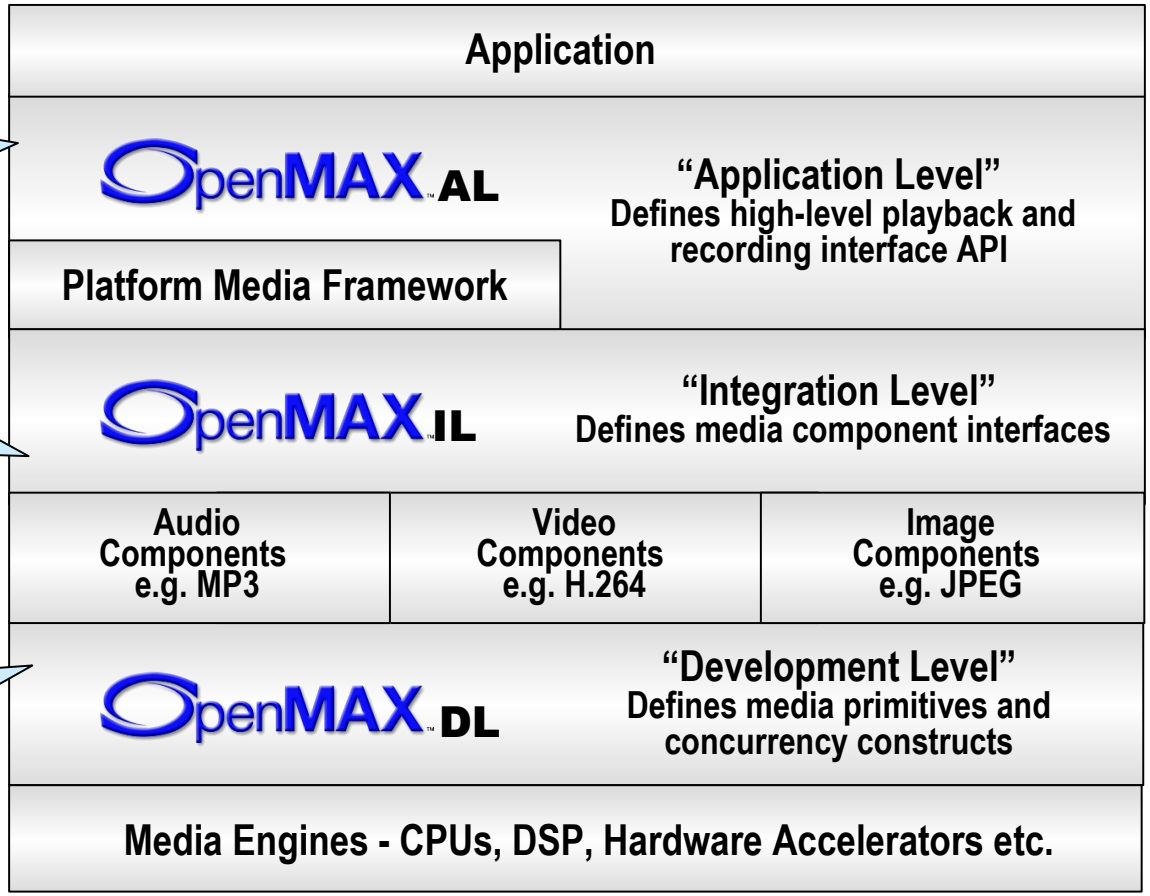
OpenMAX defines three holistically designed media open standards to provide complete media infrastructure portability

OpenMAX-based Media Stack

Media applications can be written portably, independent of the underlying media platform

Media components can be integrated into flexible media graphs for advanced streaming media processing

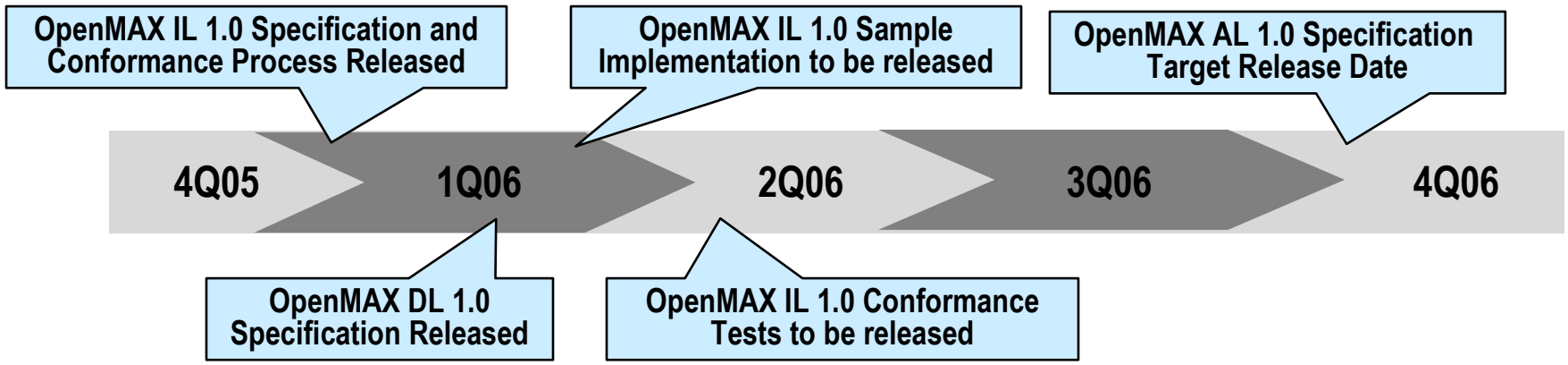
Media components can be written using primitives for portability across diverse parallel and serial silicon architectures



OpenMAX layers can be implemented together or independently from the other layers

OpenMAX Summary

- **Created with strong industry consensus and participation**
  - ARM, ATI, Beatnik, Broadcom, Emuzed, Fraunhofer, Freescale, Infineon, Intel, Motorola, Nokia, NVIDIA, Philips, SKY MobileMedia, Samsung, Sasken, Siemens, STMicroelectronics, Symbian, Texas Instruments
- **Specification is open and royalty-free using Khronos IP framework**
  - Delivered with sample implementations and conformance tests
- **Available on wide variety of architectures and operating systems**
  - To enable true streaming media portability



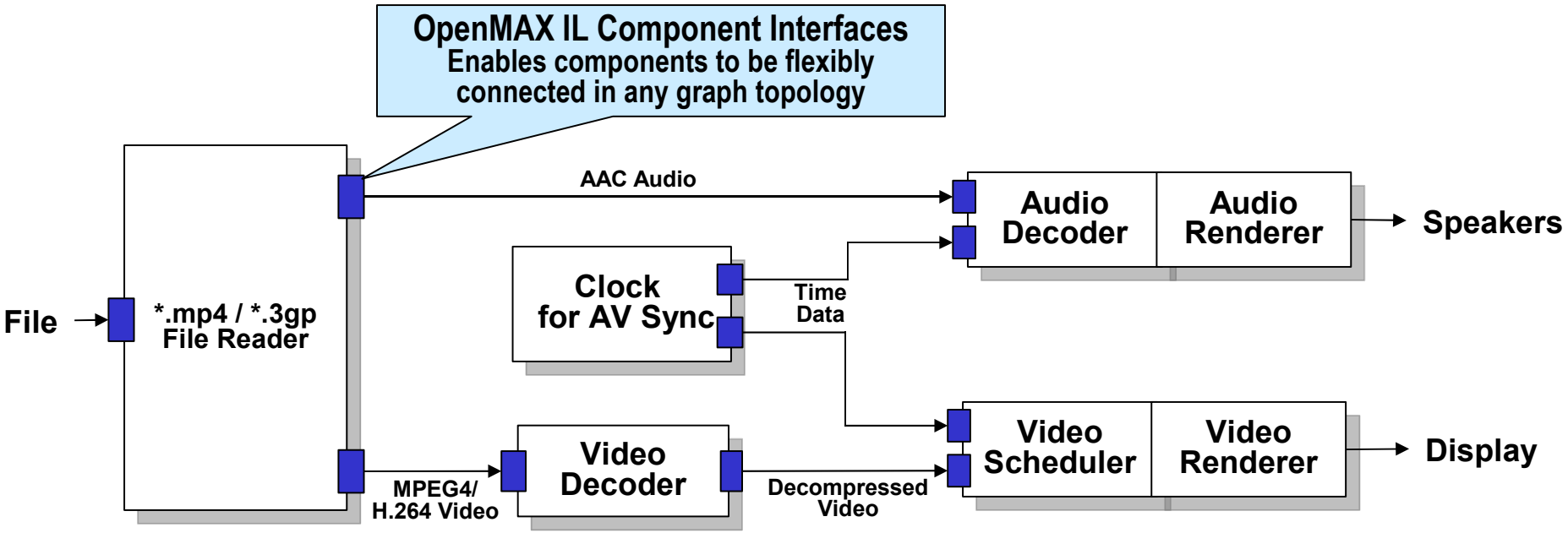
Integration Level

OpenMAX IL - Integration Level

- **Defines component interfaces to construct portable media graphs**
  - OpenMAX IL graphs are consistent across systems
- **Abstracts hardware architecture**
  - Processor specific code is encapsulated within components
  - Intelligently built components maximize system utilization
- **Reusable integration with major media frameworks**
  - Provides a uniform interface for framework integration across many architectures
  - Designed to sit below major frameworks - e.g. Symbian MDF, GStreamer, DirectShow, MMAPI
  - Defines a low level initialization and communication protocol
- **Extensible**
  - API extensions can be used to expose non standard features with only minor tweaks
- **Media graph use cases can be reused**
  - Use cases can be debugged in parallel on different projects and then shared
- **Enables Performance Comparisons and Optimization**
  - Common API allows benchmarking of different architectures, implementations and frameworks
  - Performance differences can be used by vendors to find areas for further optimizatio

OpenMAX IL Example Graph

- Standardized component interfaces enable flexible media graphs
- Includes multi-stream synchronization



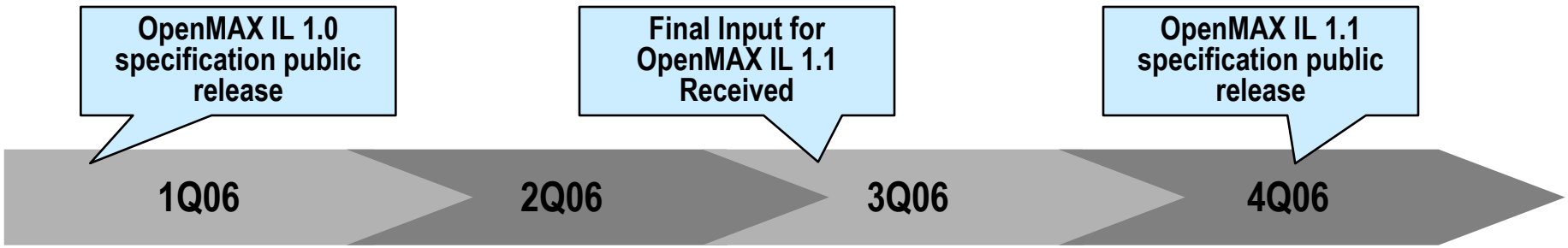
Example: MPEG-4 video synchronized with AAC audio decode

OpenMAX IL Deliverables

- **OpenMAX IL 1.0 specification - January 2006**
  - Publicly released
- **Conformance Tests – February 2006**
  - Component based with two profiles
    - Base Profile - to test the component's basic operation
    - Interop Profile - to test the component's interoperable behavior with a test component
  - Conformance tests will be validated on independently developed sample implementations
- **Linux sample implementation coded by TI – 1Q06**
  - Video - H.263
  - Audio - Narrow Band AMR
  - Image - Baseline JPEG
- **Anticipated frameworks for integration with OpenMAX IL 1.0**
  - Microsoft DirectShow, Symbian MDF, GStreamer, Java MMAPI, OpenML
- **Call for component submissions by members**
  - For refining conformance tests and OpenMAX 1.1 specification

Tentative OpenMAX IL Roadmap

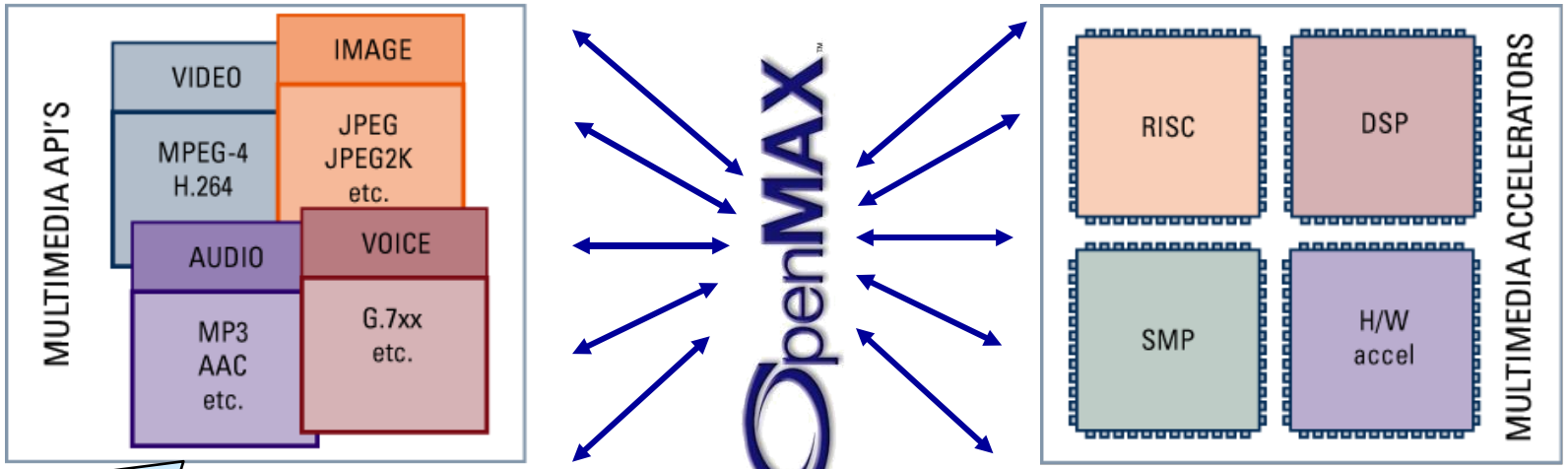
- **OS Services**
  - File I/O, Network I/O, Scheduling, Memory Management
- **Security**
  - DRM, Platform
- **Power Management**
  - Metrics, Hooks
- **Resource Management**
  - Metrics, Hooks



Development Level

OpenMAX DL – Low-Level Media API

- **OpenMAX DL is a library of key static primitive functions**
  - Designed to cover 80% of the processing required in a multimedia codec
- **Abstracts the ISA from the multimedia codec**
  - Enables faster codec development time and faster porting of existing codecs
- **Enables third party codec vendors to sell processor-agnostic codecs**
  - Multi-core architectures (i.e. ARM + DSP) gain greater code reuse between cores



An increasing number of multimedia API codecs for video, audio, graphics and images

Silicon vendors supply optimized OpenMAX DL library for rapid porting of codecs across multiple accelerators

A wide range of media acceleration silicon using many diverse architectures

OpenMAX DL Domains

- **Video Domain**
  - MPEG-4 SP/H.263 BL (encode and decode)
  - H.264 (encode and decode)
- **Image Codec Domain**
  - JPEG (encode and decode)
- **Image Processing Domain**
  - Color space conversion
  - Pixel packing/unpacking
  - De-blocking / de-ringing
  - Rotation, scaling, compositing, etc.
- **Multimedia Audio Domain**
  - MP3
  - AAC
- **Signal Processing Domain**
  - FIR
  - IIR
  - FFT
  - Dot Product

OpenMAX – Asynchronous DL (aDL)

- **API to group or chain multiple DL primitives together**
  - To form a single executing block
- **Enables vendors to accelerate key groups of primitives through:**
  - Specialized hardware
  - Co-processors
  - Hand-coded ISA optimizations
- **Enables a standard migration path between platforms**
  - With pure software and tightly coupled hardware
- **OpenMAX iDL**
  - Achieves same effect as OpenMAX aDL using OpenMAX IL constructs

OpenMAX DL Video Domain

- **Computationally intensive "hotspots" for video applications**
  - Basic video processing building blocks
- **Typical devices**
  - Digital still cameras, PDAs, Mobile Phones, Portable Media Players, Set-top-boxes, PCs, etc.
- **Example video primitive functions in OpenMAX DL 1.0**
  - 8x8 Add, Sub and 16X16 Add, Sub
  - 8x8 DCT+Q+Scan and 8x8 IDCT+Q+InvScan
  - MPEG-4 Variable Length Decode
- **Merged functions for improved performance on some architectures**
  - Motion Estimation, Motion Compensation, Deblocking
- **Video codecs covered by OpenMAX DL 1.0**
  - MPEG-4 SP/H.263 BL (encode & decode)
  - H.264 (encode and decode)
- **Can use aDL and iDL for video processing**
  - OpenMAX DL 1.1 will publish standard DL chains for aDL wrappers

OpenMAX DL Image Domain

- **Computationally intensive "hotspots" for imaging applications**
  - Basic image processing building blocks
- **Typical devices**
  - Digital still cameras, PDAs, Mobile Phones, Set-top-boxes, PCs, Printers etc.
- **Example image primitive functions in OpenMAX DL 1.0**
  - JPEG - encode and decode, 8x8 DCT and 8x8 IDCT, Quantization
    Merged DCT & quantization functions, Huffman encoding and decoding
  - Image Processing - color space conversion and packing/unpacking
    De-blocking / de-ringing filtering, Filtering, Moments, Block copy, rotation, mirroring and scaling
- **OpenMAX DL 1.1 will widen image functionality**
  - JPEG2000
  - Image Blending
  - Raw Camera data processing etc…

OpenMAX DL Speech / Audio Domain

- **Computationally intensive "hotspots" for audio applications**
  - Speech codecs are not supported since the standards are bit-exact
  - Other speech applications are supported indirectly with some signal processing APIs
- **Typical devices**
  - PDAs, Mobile Phones, Portable Media Players etc.
- **Example speech / audio primitive functions in OpenMAX DL 1.0**
  - Audio API - Unpacking of headers and bit-streams, Huffman decode, IMDCT and MDCT Polyphase filter, TNS and PNS processing
  - Signal Processing API - FFT and IFFT, FIR, IIR and Median filters, Dot product, Block exponent (finding minimal sign bits in array elements)
- **Example uses**
  - MP3 decoder, including low frequencies extensions, MPEG4-AAC decoder (LC/L TP profiles), Signal processing (FFT, digital filters, some math)
- **OpenMAX DL 1.1 will widen functionality**
  - Audio encoders
  - EAAC, EAAC+
  - LMS filters
  - Voice Recognition front-end …

Application Layer

OpenMAX AL - Application Level

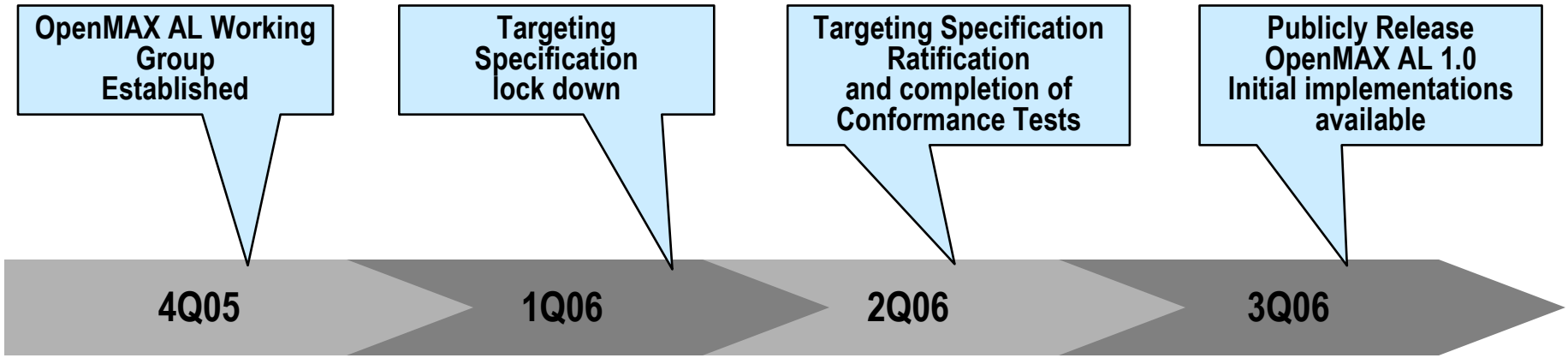
- **Enabling application developers to easily leverage OpenMax acceleration**
  - A simple high-level interface for common multimedia playback and capture use cases
- **Typical applications are found in:**
  - Mobile Phones
  - Mobile Music/Video Players
  - PDAs
  - Digital still cameras
  - Digital Media Adapters
  - STBs, PCs, etc…

OpenMAX AL 1.0 – Scope

- **Standard use cases**
  - Playback: play a video file, play a music file, display an image file
  - Recording: record a video file, record an audio file, capture an image file
- **Operational controls**
  - Playback: play, pause, stop, FF, RW
  - Recording: record, stop
- **Configuration control**
  - Audio output: volume, channels, etc
  - Video output: video window position, size, etc
- **Metadata controls**
  - Extract metadata from a playing stream
  - Insert metadata into a recording stream

OpenMAX AL - Milestones

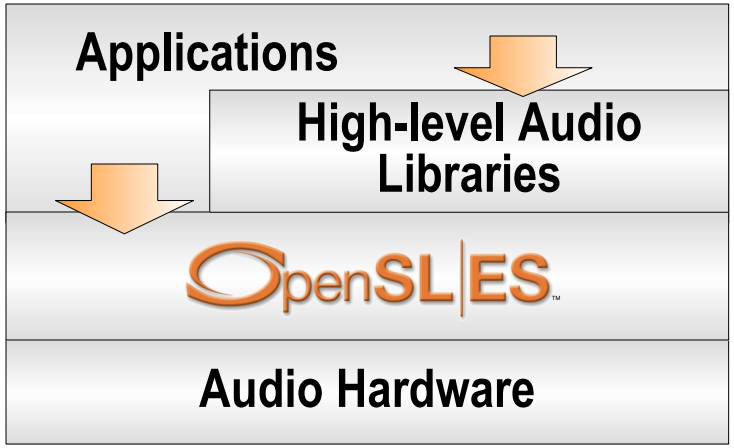
- **OpenMAX AL Taskforce formed in November 2005**
  - Membership included: ATI, Beatnik, Freescale, Nokia, NVIDIA, Symbian, SkyMobile Media, TI
  - Scoped intended functionality and investigated alternative solutions
  - Recommended formation of an OpenMax AL working group
- **OpenMAX AL Working Group formed in December 2005**
  - Call for widened working group participation
  - Official scope/requirements definition at face to face meeting in January 2006



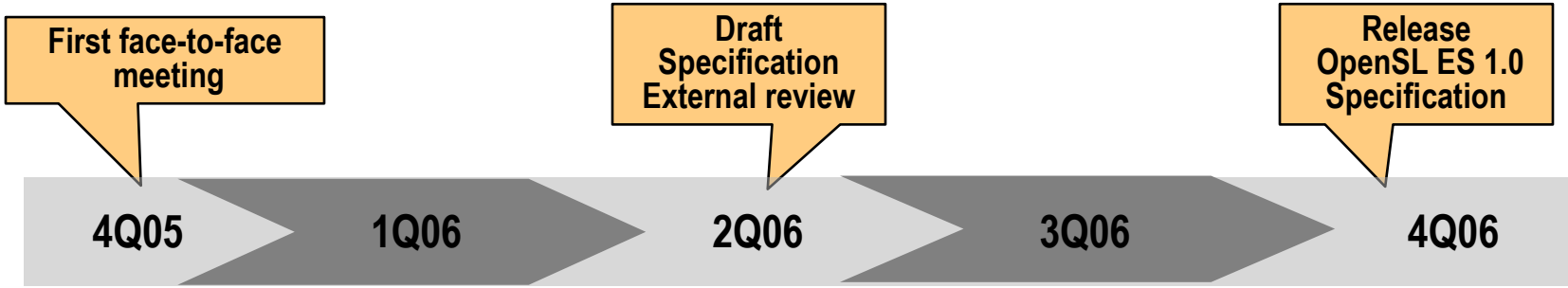
Accelerated Embedded Audio

OpenSL ES – Solving an Audio Crisis

- **Many different proprietary audio APIs**
  - Even playing a simple sound is different on different platforms
- **No standard way to access any available hardware**
  - Lots of work for developers to re-write code for every platform
- **Need a unified native audio acceleration API**
  - Targeting handheld devices
- **API for games developers - low-latency sound generation for games**
  - Advanced audio functionality: 3D positional audio, reverb, SP-MIDI
- **Cross-platform foundation for a wide range of higher-level audio APIs**
  - OpenAL, Java sound APIs (JSR-135 and JSR-234)



Working Group Milestones

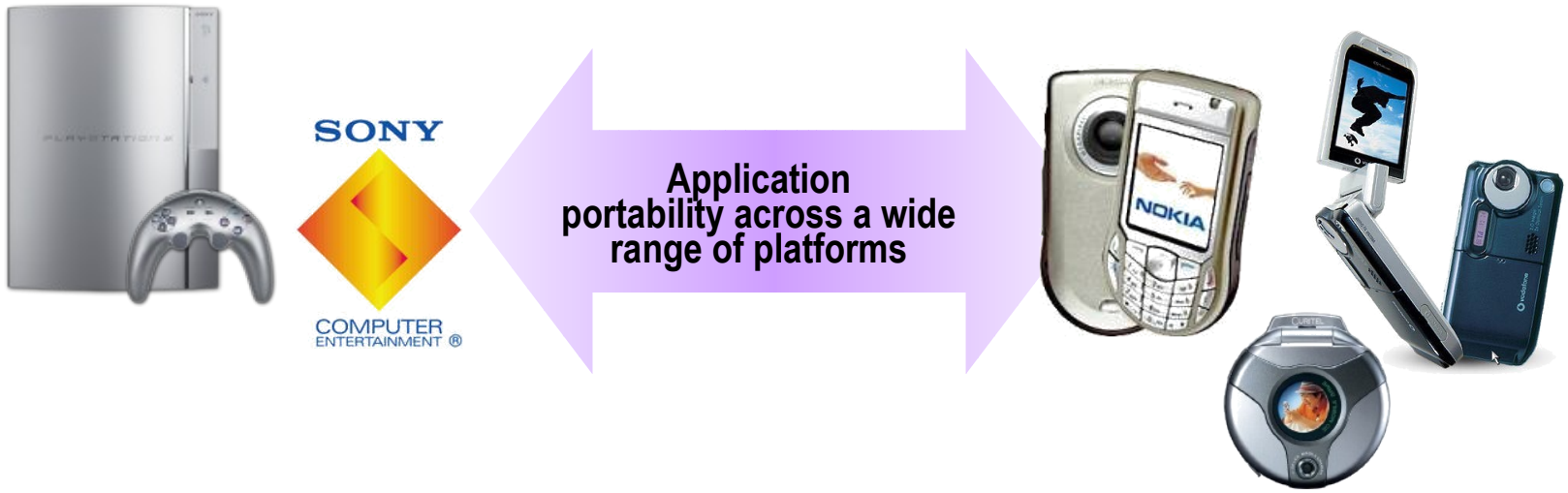



Strong industry quorum of leading audio hardware and software companies.
Open call for call for participation

3D Tools Ecosystem

OpenGL ES – More than Handhelds

- **OpenGL ES will be available for Playstation 3**
  - Sony made public announcement at GDC in San Francisco in March 2005
  - All interactive demos at E3 were using OpenGL ES
- **Powerful portability for console and handset titles**
  - Previous generation console games can be deployed on 100s of millions of cell phones



THE POWER OF COOPERATIVE OPEN STANDARDS
A growing infrastructure of OpenGL ES tools and platforms are creating one of the world's largest opportunity for games developers

COLLADA – 3D Creation & Delivery

- **Digital Asset Exchange Schema – making ISVs more productive**
  - Enables combination of 3D authoring tools to increase the power of tool chains
- **Packaging format for content delivery – including shaders and physics**
  - COLLADA FX and COLLADA Physics included in new COLLADA 1.4
- **Strong synergy between COLLADA and OpenGL ES 2.0**
  - Collada enables shaders to authored and packaged using OpenGL ES Shading Language

Collada enables leading 3D authoring tools to work effectively together – lossless interchange of assets

Collada 1.4 released in January 2006 – includes shader and physics data definitions to communicate advanced effects to OpenGL ES

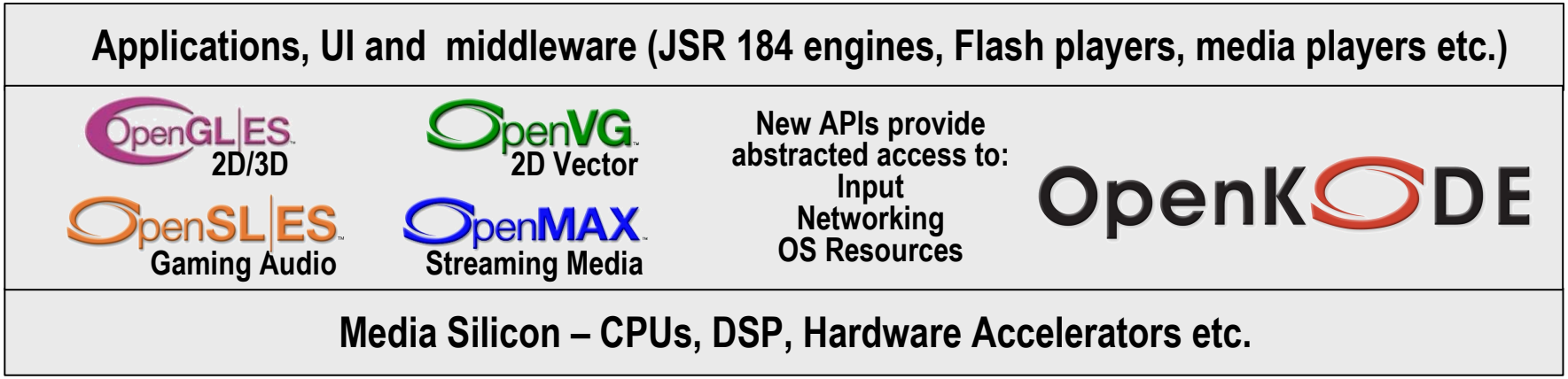
Khronos considering work on a COLLADA FX API and associated run-time

Tightly Integrated 3D Authoring and Deployment Stack

Khronos Open Development Environment

OpenKODE Native API Platform

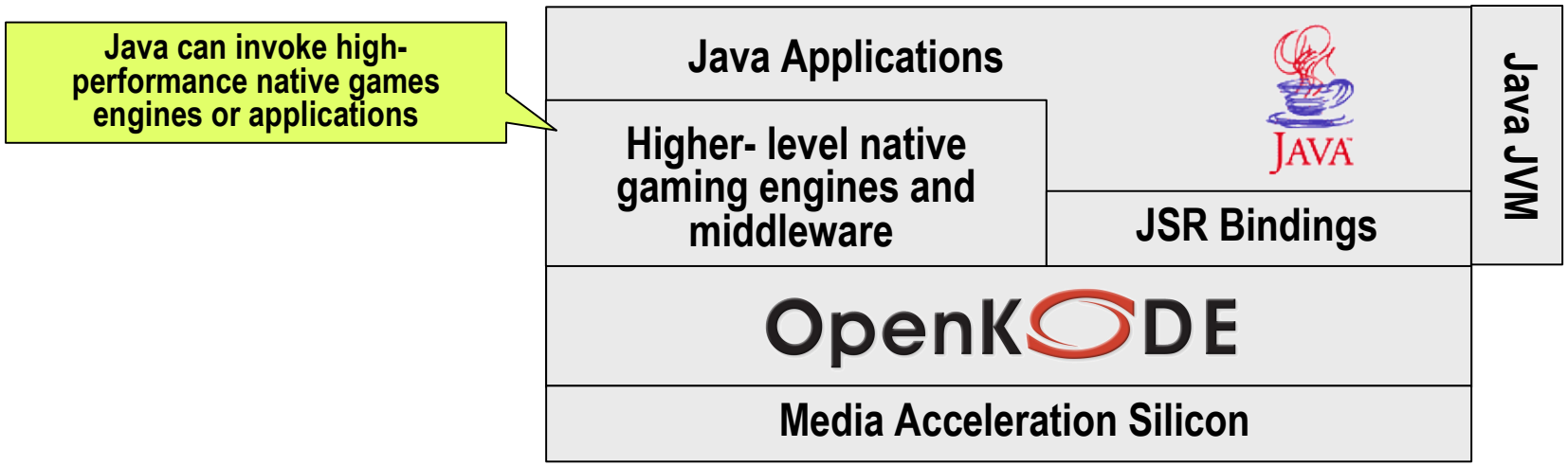
- **Collect Khronos media APIs into a single platform**
  - Rigorous conformance tests for robust reliability
- **New APIs for full portability**
  - Input, Network and OS Resources
- **Functionally similar to the DirectX platform**
  - Except cross-platform, royalty-free and streamlined



OpenKODE provides cross-platform, source level application portability

OpenKODE Additional Work Items

- **Interfaces to platform capabilities when present**
  - E.g. Java, Brew, OMA – for provisioning, billing etc.
  - C binding to CHAPI (JSR 248 – Content Handler API)
- **Enhanced conformance tests**
  - Including trans-API tests
- **Platform capability profiling tool - similar to WinSAT**
  - Query functions and measure key throughputs
  - Avoid "lowest common denominator syndrome"

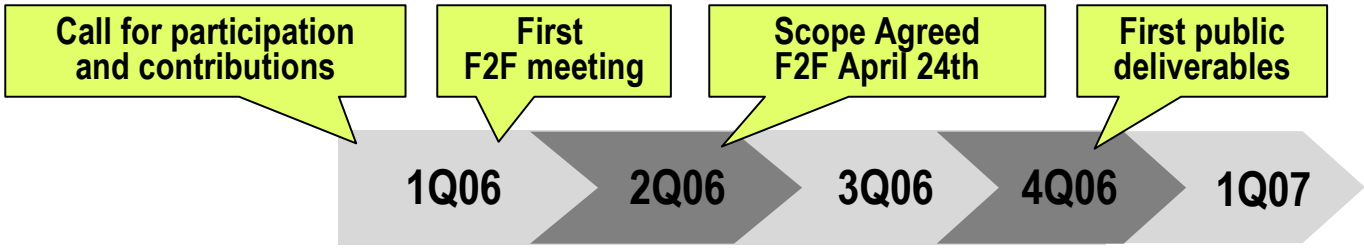


OpenKODE Bottom Line

- **Silicon vendors expose media acceleration through reliable APIs**
  - With full native performance
- **Carriers and OEMs have deployment control**
  - Can simply specify “OpenKODE Compliance”
  - Can use existing provisioning and billing infrastructure
  - Can use existing DRM and signature checking
  - Implement own certification programs
- **ISVs gain a wider market footprint**
  - ISVs use C and spend time developing not porting
  - Native applications run 3-5 times faster than Java
  - OpenKODE applications can ship on wide range of phones
- **More compelling applications – faster to market**

OpenKODE Milestones

- **40 companies participating in OpenKODE working group**
  - ARM, Ericsson, Freescale, Ideaworks 3D, Intel, Monotype, Nokia, NVIDIA, SUN, Tao, TI
- **Aiming for OpenKODE 1.0 release in 2006**
  - Keeping things as simple possible for first release while delivering significant benefit
- **OpenKODE is NOT:**
  - An operating system, DRM scheme, provisioning scheme or a carrier certification process
- **Maximize leverage of existing mobile infrastructure**
  - E.g. enable flexible Java invocation of native libraries, applications and engines
- **Working for industry adoption on many platforms**
  - Java, Symbian, WIPI, Linux, Windows, Brew, Nucleus
  - Simple build out of existing OpenGL ES/OpenVG adoption



Why Use Khronos Standards?

- **Khronos is creating a complete, coherent media acceleration platform**
  - To reduce development and deployment costs and increase market opportunity
- **"Foundation Level" APIs**
  - Close to the silicon – fundamental, flexible functionality needed on every platform
- **Designed by industry experts**
  - The industry leaders in media silicon, platforms and software are all Khronos members
- **Flexible, fast-track roadmap evolution**
  - Effective and streamlined process – specification updates every 12 months if needed
- **Royalty-free**
  - Khronos is committed to generating market opportunities for its members and the industry
- **Any company is welcome to join Khronos!**
  - Only $5,000 / year membership fees

KHRONOS GROUP

Khronos API's open source implementations

- **Vincent – OpenGL ES 1.1 software implementation**
  - Already available for PocketPC/Windows SP and Symbian
  - Next version on Linux for GP2x platform running ARM Embedded Linux
  - More information here: http://sourceforge.net/projects/ogl-es

- **Bellagio OpenMAX IL 1.0**
  - ST Microelectronics made available OpenMAX IL 1.0 implementation
  - Linux version available here: http://sourceforge.net/projects/omxil