
# An Overview of the Khronos Application-level Multimedia API

(Note: best viewed in slide-show mode)

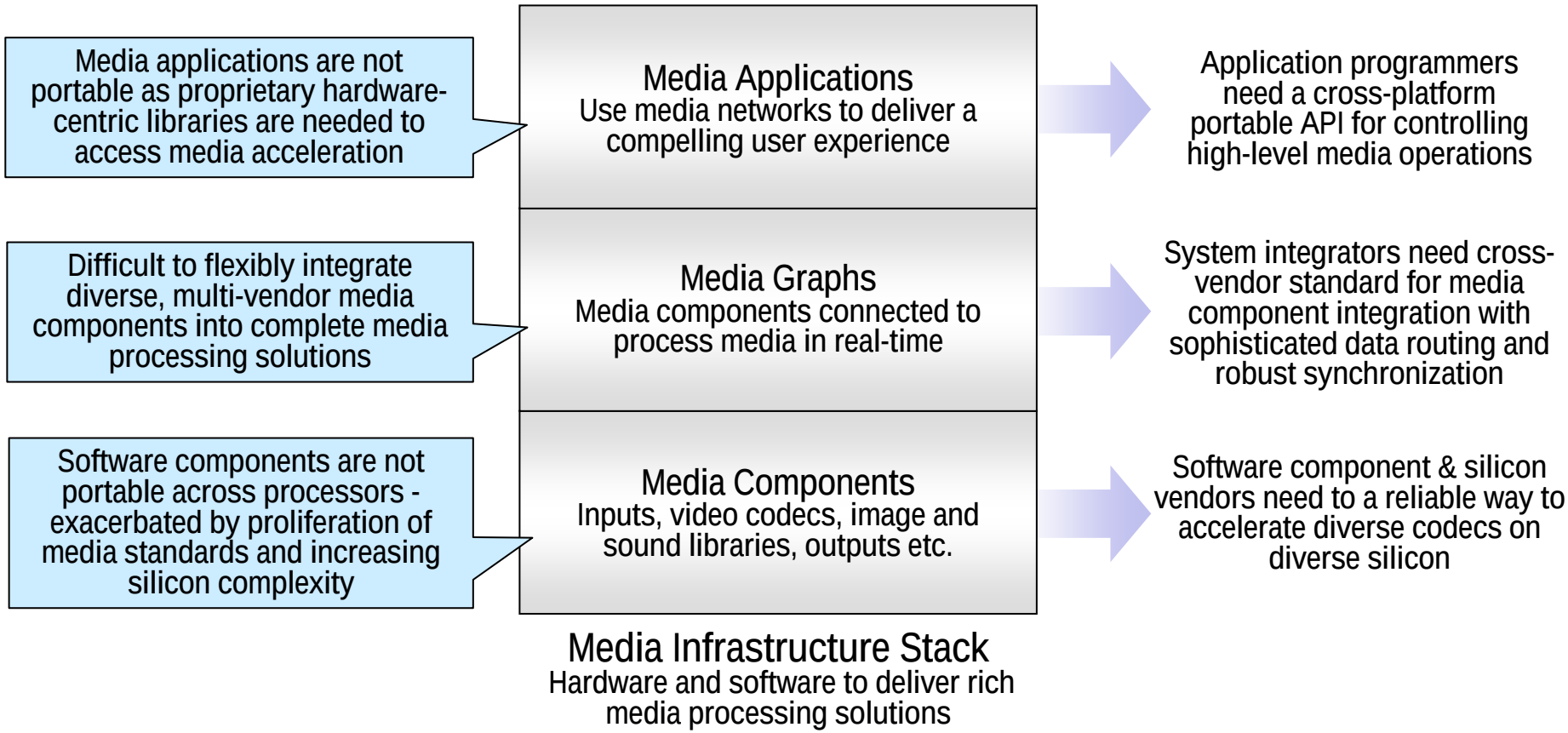
KHRONOS GROUP

# Introduction

- This slide-set provides a brief technical summary of the OpenMAX AL v1.0 specification and highlights its main architectural features
- An overview of the OpenMAX suite of APIs and links to the respective specifications are available here: http://www.khronos.org/openmax

# Problem: Streaming Media Portability

- Media infrastructure portability is a multi-level industry problem
  - Media infrastructure is time-consuming and expensive to develop, integrate and program

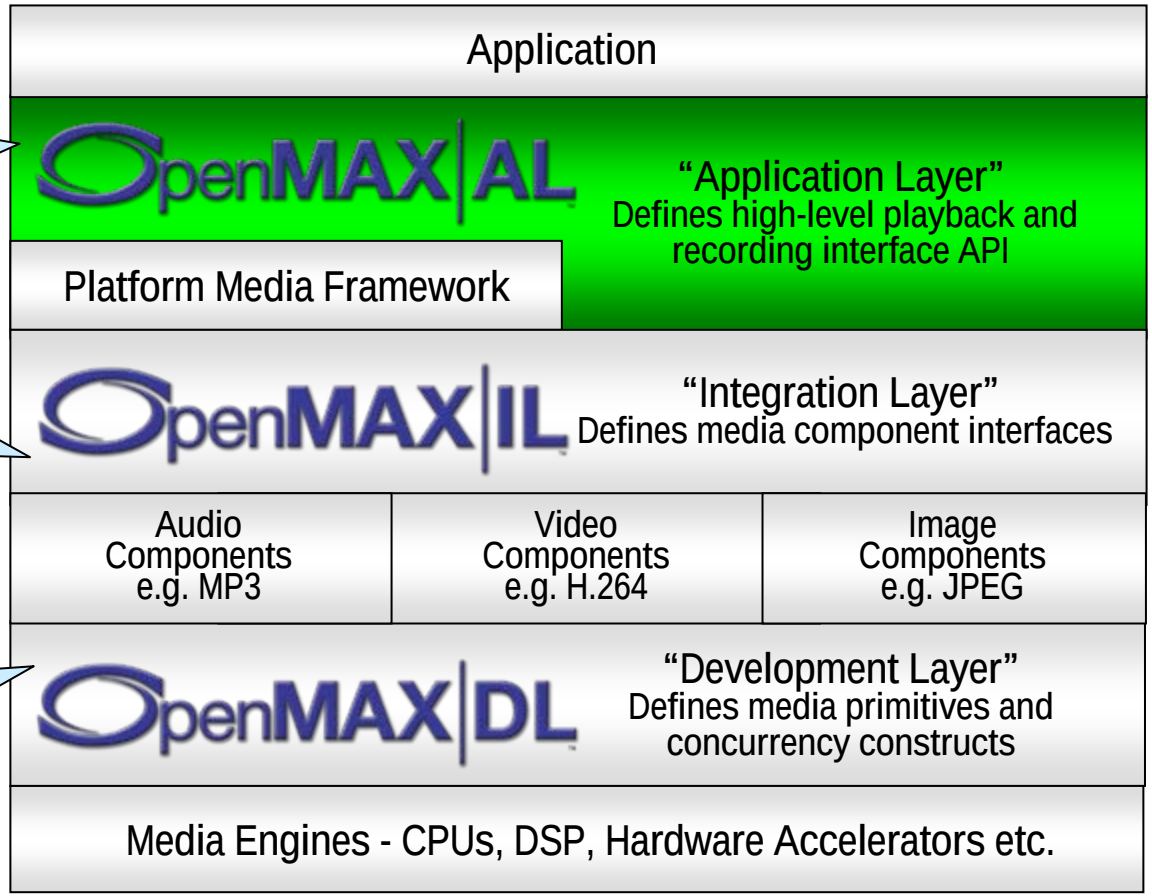


# OpenMAX – Streaming Media

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 AL

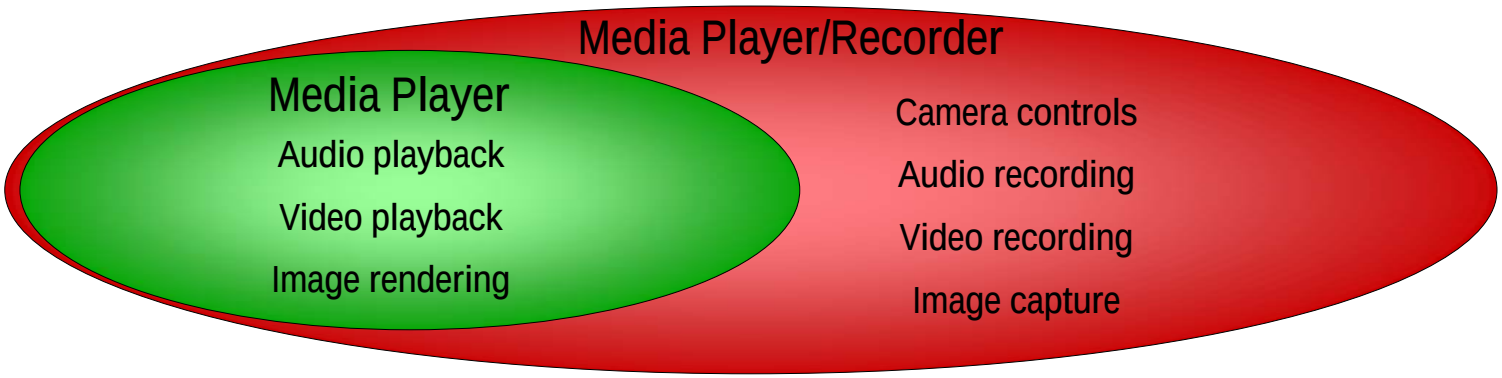
- OpenMAX AL is targeted to the application developer
  - Easy to use, high-level access to rich multimedia functionality
- OpenMAX IL is powerful but complex
  - More power than most application developers require
- Most application developers just want to playback and record media
  - Specifying where the content comes from
  - Specifying where the content should be rendered to
  - Manipulate a few playback controls
  - Have simple configurability
- That's what OpenMAX AL provides!
  - An application level multimedia API for playback and recording use cases

# API Profiles Explained

- Definition: "*An API profile is a defined subset of features satisfying typical use cases for a given market segment. Any feature may be included in any profile. Any device may support any number of profiles.*"
- Profiles are good for developers
  - Less fragmentation – all functionality in a profile must be present on all devices supporting that profile
  - Easy application portability across devices
- OpenMAX AL will be deployed on a wide range of devices catering to different market segments
  - The entire OpenMAX AL feature-set is large and not suitable for all devices
  - Profiles allow OpenMAX AL to be targeted towards specific market segments

# OpenMAX AL Profiles and Extensions

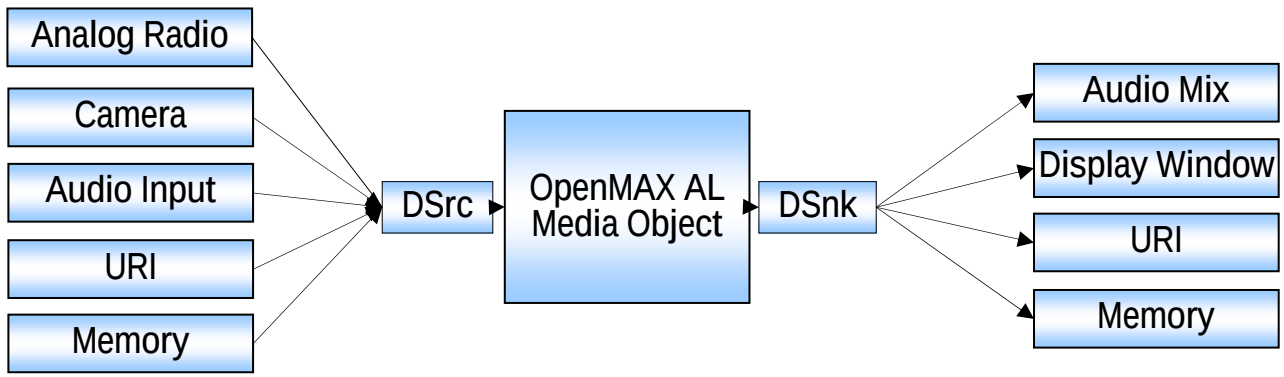
- Two profiles:
  - Media Player – designed for media playback-only devices
  - Media Player/Recorder – designed for full-featured media devices




- Optional extensions
  - Some features are optional in all profiles, mainly because of hardware considerations or use case constraints
  - Standardized for API consistency when the hardware is available or the use cases warrant it
  - Examples: Vibra, LED, Analog Radio and MIDI are optional extensions of OpenMAX AL
- Vendor-specific extensions supported - successful extensions may be integrated into later versions of the APIs

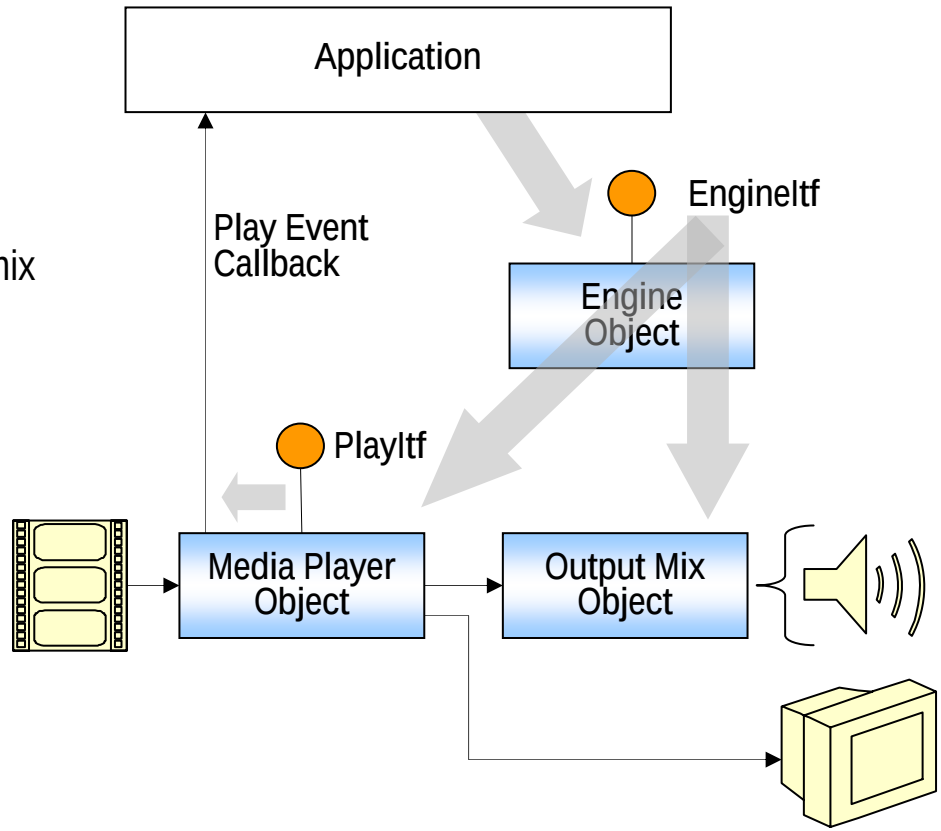
# OpenMAX AL – Object Oriented Media

- OpenMAX AL has an object-oriented programming model
  - Simplifies common use cases – but also extensible
- Engine Objects are central to any OpenMAX AL session
  - Media Objects created using methods on the Engine Object interfaces
- OpenMAX AL Media Objects enable PLAY and RECORD of media
  - Perform some operation on an input and emit the result as output
  - Can handle audio, images, video with synchronized audio
- Objects have control interfaces
  - Play, Seek, Rate, Audio, Display Region, Metadata Extraction
  - Record, Camera, Video Encoder, Audio Encoder, Metadata Insertion, Radio, MIDI
- Media Objects input and output to Devices



# OpenMAX AL Video Playback Example

- Create Engine object
  - To drive this session
- Create Audio Output Mix object
  - Method on Engine interface
  - Mix object drives audio output devices
- Create Media Player object
  - Method on Engine interface
  - Input is URI pointing to a local media file
  - Output drives display and audio output mix
- Register event callback
  - Method on Media Player interface
- Set PlayState to Playing
  - Method on Media Player interface
- Wait for end of file event
  - Via registered callback



# Other OpenMAX AL Features

- Extensive camera controls
  - Flash modes
  - Focusing controls
  - Metering modes
  - Exposure compensation
  - ISO Sensitivity
  - Shutter speed & Aperture
  - White balance controls
  - Zoom (digital and optical)


- Analog radio controls
  - Tuning
  - RDS


- Audio routing
  - Application-selectable audio inputs and outputs, based on location, connectivity, etc.
  - I/O device capability querying
- Metadata extraction and insertion
  - Ability to search/extract metadata in a variety of file formats
  - Ability to insert/overwrite metadata into a variety of file formats

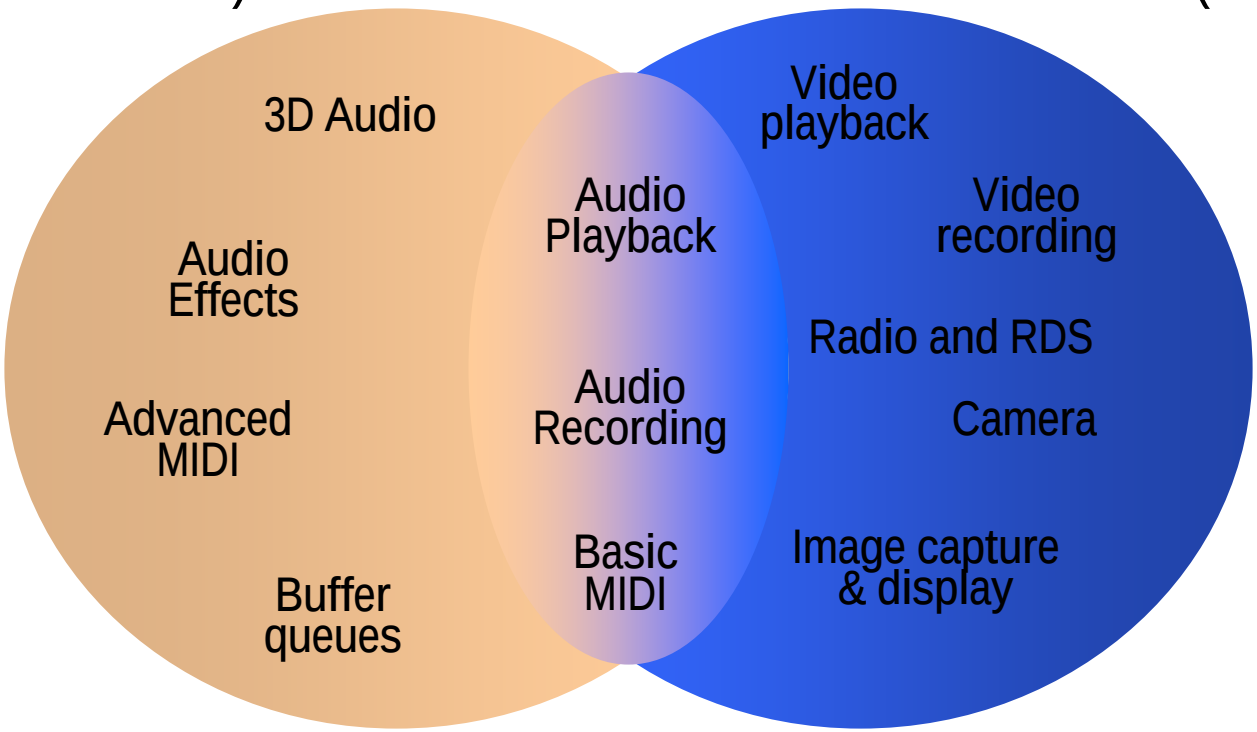
# OpenSL ES and OpenMAX AL (1/2)

OpenSL ES

(Enhanced audio API)

OpenMAX AL

(Multimedia API)




- Working groups collaborated to define the common API functionality.

# OpenSL ES and OpenMAX AL (2/2)

- Independent
  - No dependency between the APIs – either one can exist by itself
  - A device may support any combination of the APIs that most suits the device:
    - OpenMAX AL only (Media Player/Recorder profile)
    - OpenSL ES only (Phone, Game, Music profiles)
    - OpenMAX AL + OpenSL ES (Media Player/Recorder + Music)
    - OpenMAX AL + OpenSL ES (Media Player/Recorder + Game)
    - ….
- Compatible
  - Working groups collaborated to make sure the APIs work together well
- Consistent
  - Identical API architecture
  - Identical APIs for same functionality
- Distinct
  - OpenMAX AL represents basic multimedia functionality (audio, video and image)
  - OpenSL ES represents advanced audio-only functionality

For more information and links to the specification:
http://www.khronos.org/openmax