ECEN 5803

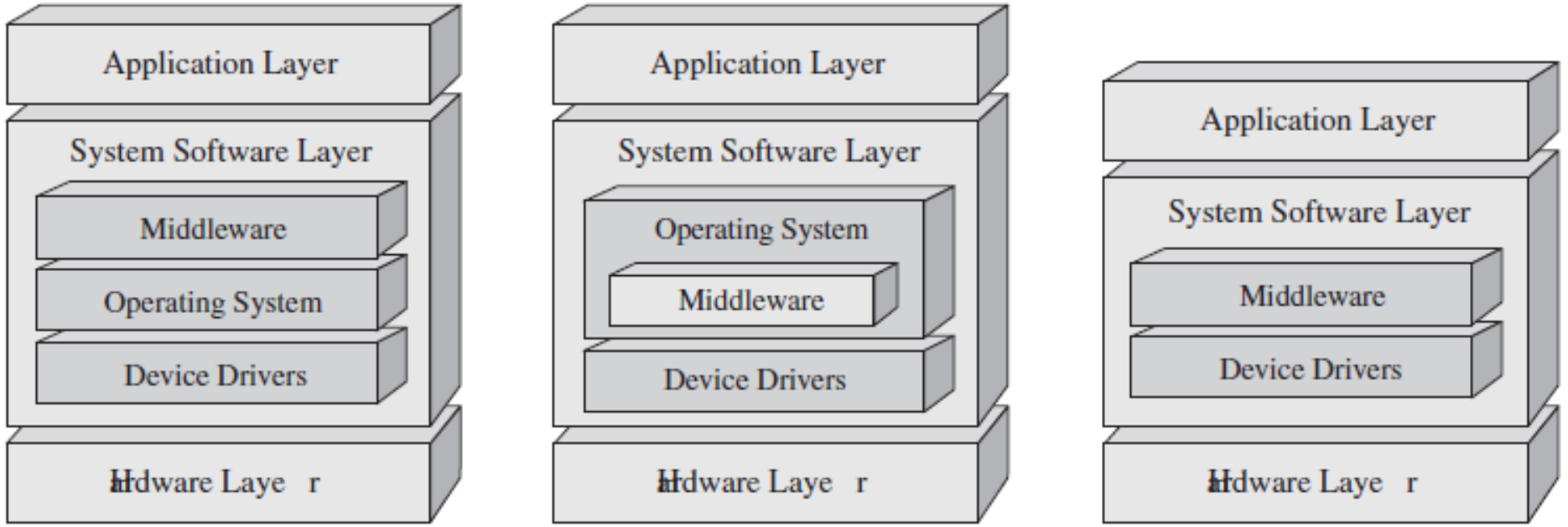
Mastering Embedded Systems Architecture

Be Boulder.

University of Colorado Boulder

Middleware

Middleware software is any system software that is not the OS kernel, device drivers, or application software. Note that some OSes may integrate middleware into the OS executable. In other words, in an embedded system middleware is system software that typically sits on either the device drivers or on top of the OS, and can sometimes be incorporated within the OS itself.

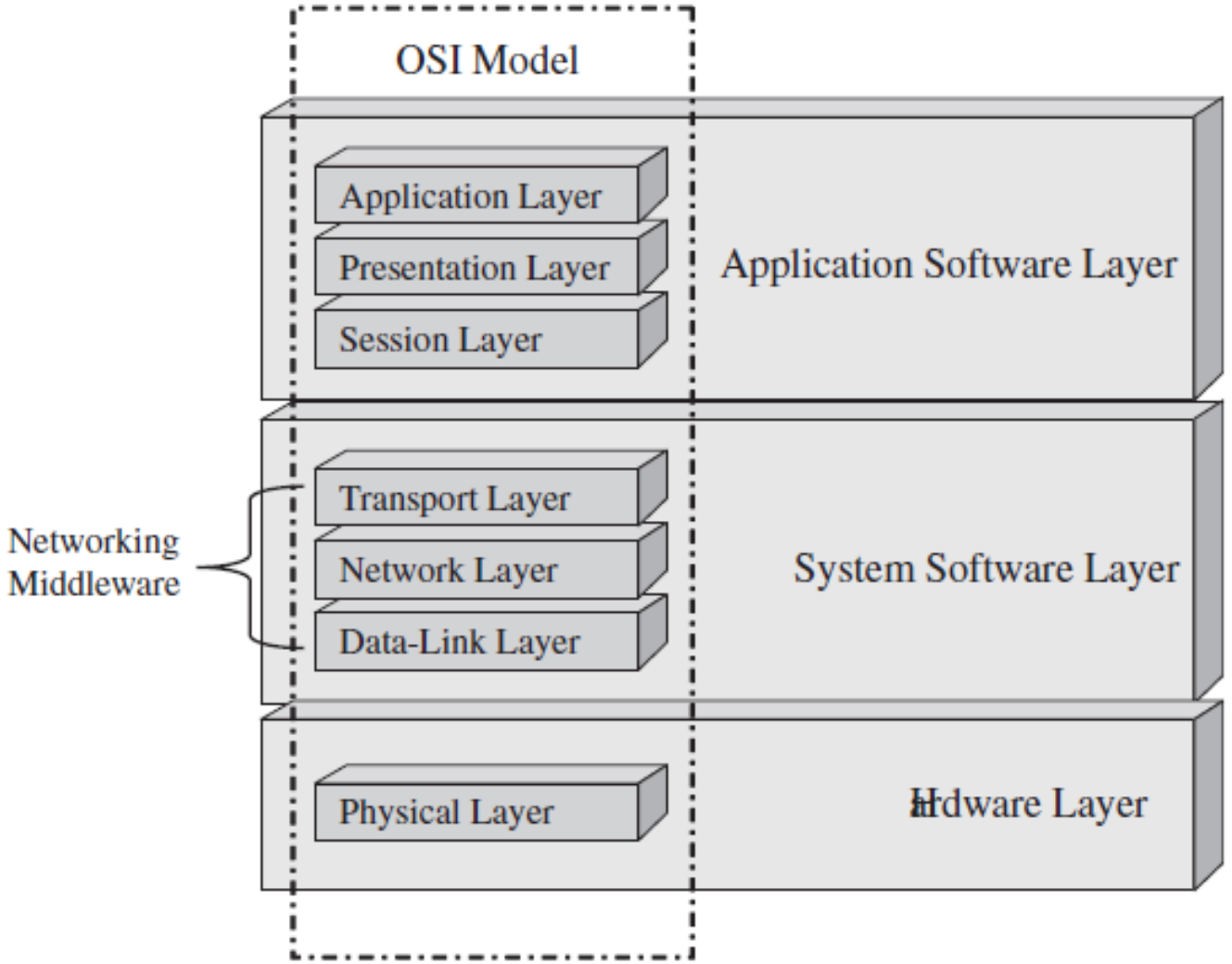


"Embedded Systems Architecture", 2013, Noergaard, pp. 453-455.

Middleware

Middleware Examples?

TCP/IP
.NET
JVM
USB stack
Real Time Clock
USB Audio Class Driver
CAN Driver
FAT File System
Common FLASH Interface

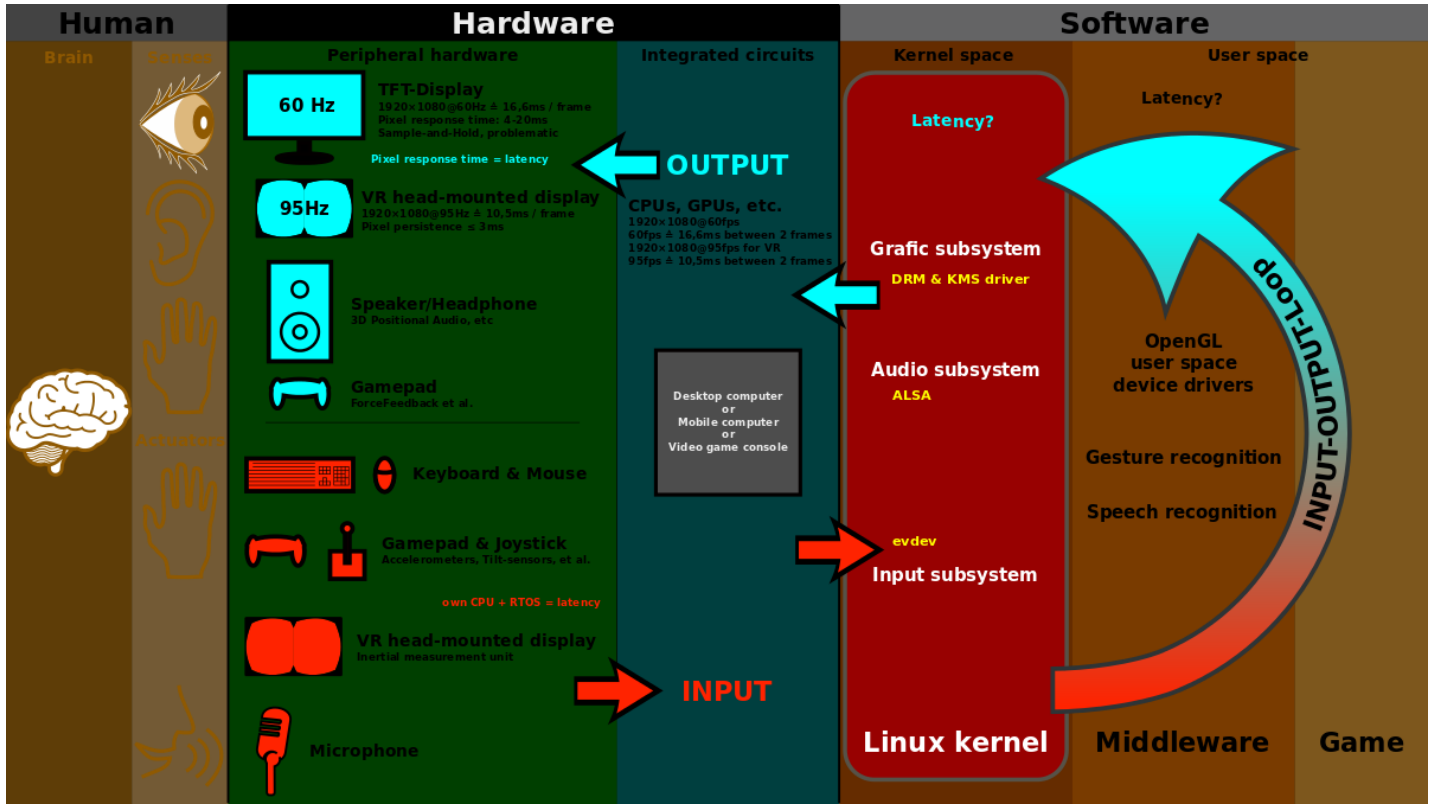


Middleware

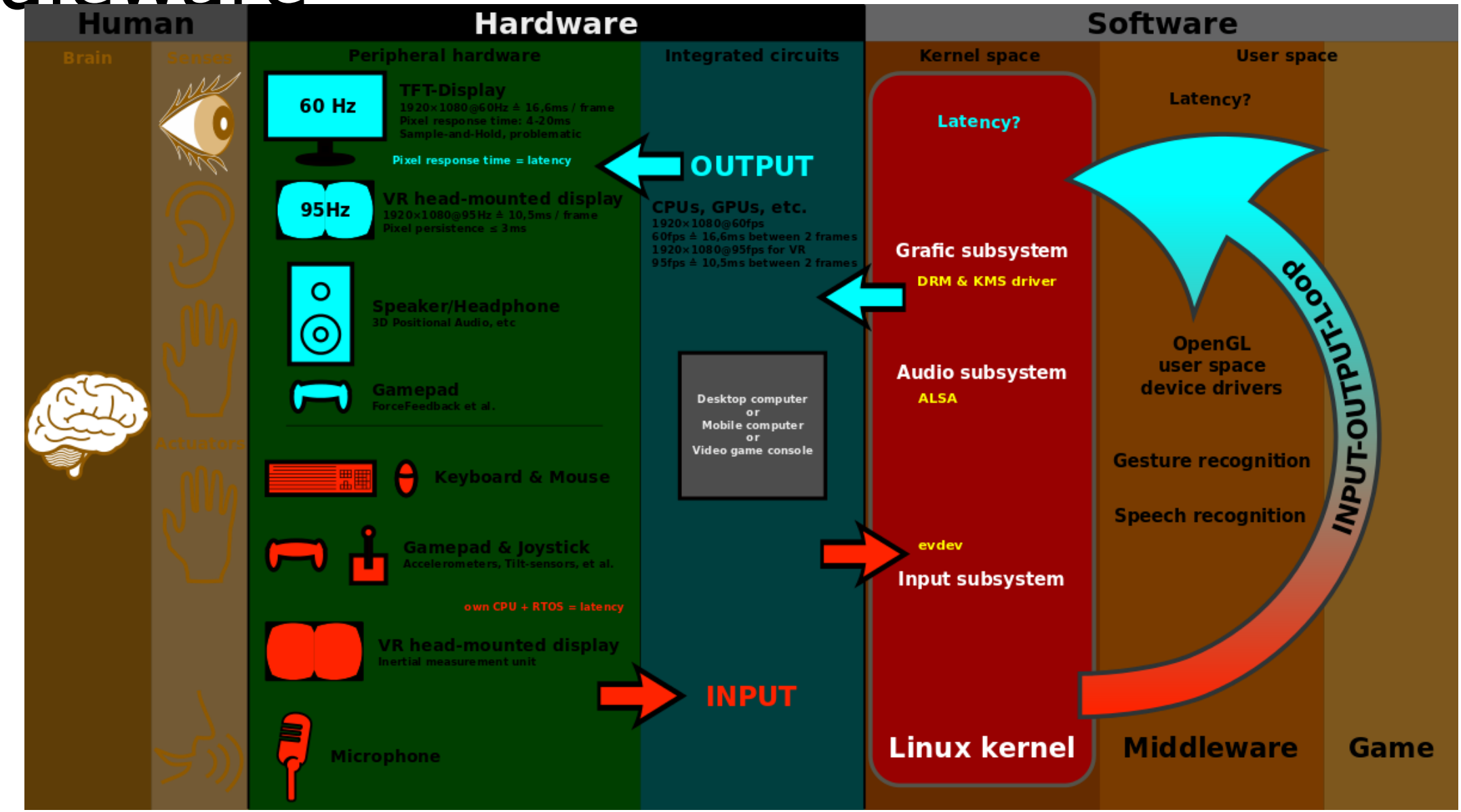
Middleware Sources

Historically, 4 primary sources of Middleware:

1. Build your own.
2. Buy from a 3rd party vendor.
   There are dozens.
3. Acquire by OS selection.
4. Open Source.



Middleware


Human
Hardware
Software
Brain
Senses
Actuators
Peripheral hardware
Integrated circuits
Kernel space
User space
60 Hz
TFT-Display
1920×1080@60Hz ≙ 16,6ms / frame
Pixel response time: 4-20ms
Sample-and-Hold, problematic
Pixel response time = latency
OUTPUT
95Hz
VR head-mounted display
1920×1080@95Hz ≙ 10,5ms / frame
Pixel persistence ≤ 3ms
CPUs, GPUs, etc.
1920×1080@60fps
60fps ≙ 16,6ms between 2 frames
1920×1080@95fps for VR
95fps ≙ 10,5ms between 2 frames
Latency?
Grafic subsystem
DRM & KMS driver
Speaker/Headphone
3D Positional Audio, etc
Gamepad
ForceFeedback et al.
Desktop computer
or
Mobile computer
or
Video game console
Audio subsystem
ALSA
Keyboard & Mouse
Gamepad & Joystick
Accelerometers, Tilt-sensors, et al.
evdev
Input subsystem
own CPU + RTOS = latency
VR head-mounted display
Inertial measurement unit
INPUT
Microphone
Linux kernel
Latency?
OpenGL
user space
device drivers
Gesture recognition
Speech recognition
INPUT-OUTPUT-LOOP
Middleware
Game

Middleware

Middleware Sources - Buy

Jungo –

- CarPlay and Android Auto for Car Infotainment
- MediaCore - Infotainment Multimedia and Connectivity Middleware
- USB Host Stack
- USB Device Stack
- DriverCore - PC USB Drivers
- WinDriver - PCI/USBDevice Driver Development Tool

See http://www.jungo.com/st/products/mediacore-embedded-ivi-multimedia-software-middleware/

Middleware

Middleware Sources - Buy

Clarinox – **Products**

- Jannal IDE
- Bluetooth Stack
- WiFi Stack
- ClarinoxSoftFrame
  - Supported Hardware
  - Supported OS/RTOS
- Koala EVM
  - Specifications
  - FAQs
- Services
- RFID Middleware
- GPRS/3G AT Engine

See http://www.clarinox.com/clarinoxblue

Middleware

Keil Middleware allows you to develop robust applications using a wide variety of communication protocols

Develop robust and powerful applications fast

The RTX kernel, together with its sources, gives you all the resources you need to create and control multi-threaded, real-time applications and can be tailored to your exact system requirements. Most Keil Middleware libraries can be used with or without RTX.

Do only what you need

MDK-Professional includes libraries to enable TCP/IP networking, CAN, USB, and Flash file-system support. Use these existing resources to allow you to focus on the development of the core of your application. There's no need to waste time re-implementing low-level functions.

Take advantage of Keil's expertise

All Keil middleware libraries have been designed, tested, and optimized by ARM and Keil engineers for ARM platforms. The libraries have a modular design with simple APIs. Documentation and examples make it easy to re-use the work done by our experts.

Source Code

The TCP/IP networking, USB, and Flash file-system components are available as source code as a separate product.

Middleware

Other benefits of Keil middleware libraries include:

•**Scalability** - their modular structure means that applications are easy to design and maintain. Libraries can be used and re-used in multiple applications, from the most simple to the highly complex.

•**Abstraction** - by removing the requirement to implement low-level peripheral and task management functions you allow your engineers to concentrate on what they know best - developing the core functionality of your application.

•**Reliability** - when you take advantage of the Keil middleware code you are working with the confidence that all of these components have been rigorously tested by ARM engineers and third party users.

•**Cost** – MDK-Professional middleware libraries are supplied Royalty-Free.

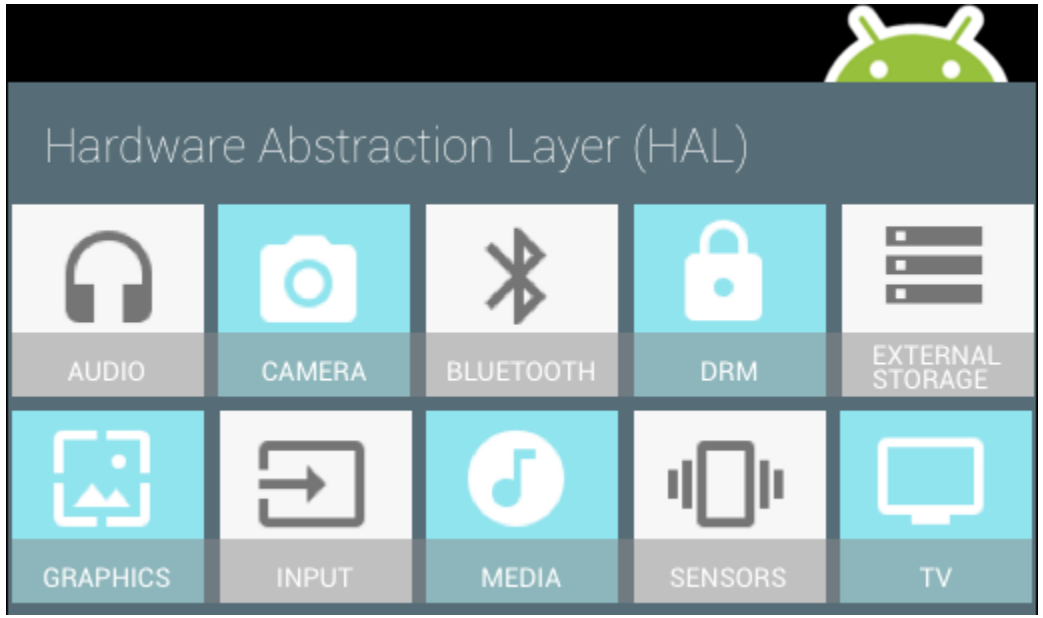
While it is possible to develop similar functionality yourself, this approach rarely reduces cost and in most cases leads to unexpected delays, particularly if your engineers are working in unfamiliar areas. MDK-Professional enables you to reduce cost and development time.

•**Support** - There are extensive online resources including examples and templates to enable you to quickly start your project. MDK-ARM is fully supported by Keil to ensure that if you have questions during your development then you are not alone.

Middleware - Android

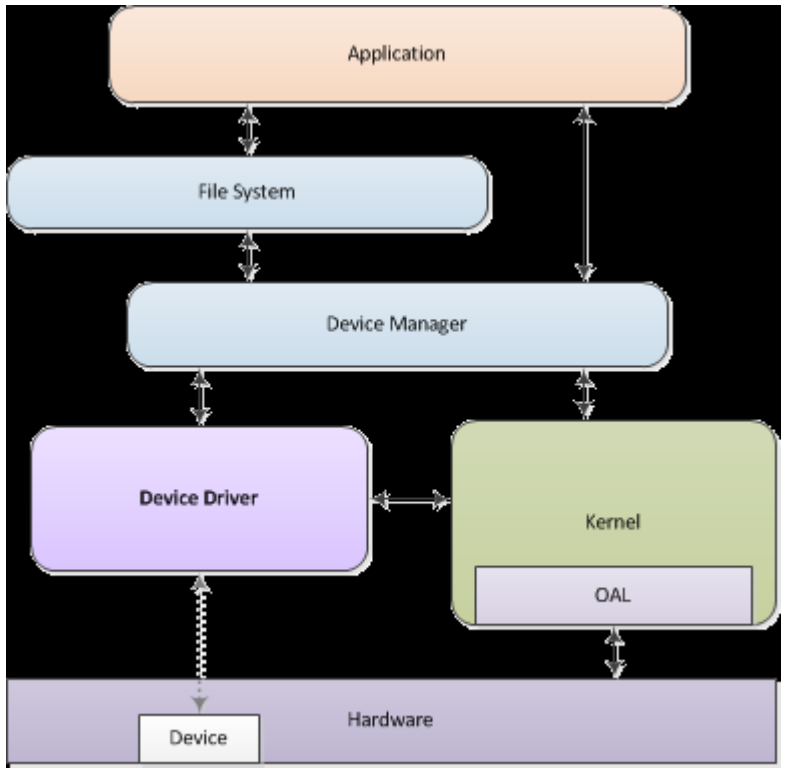
Android is Middleware

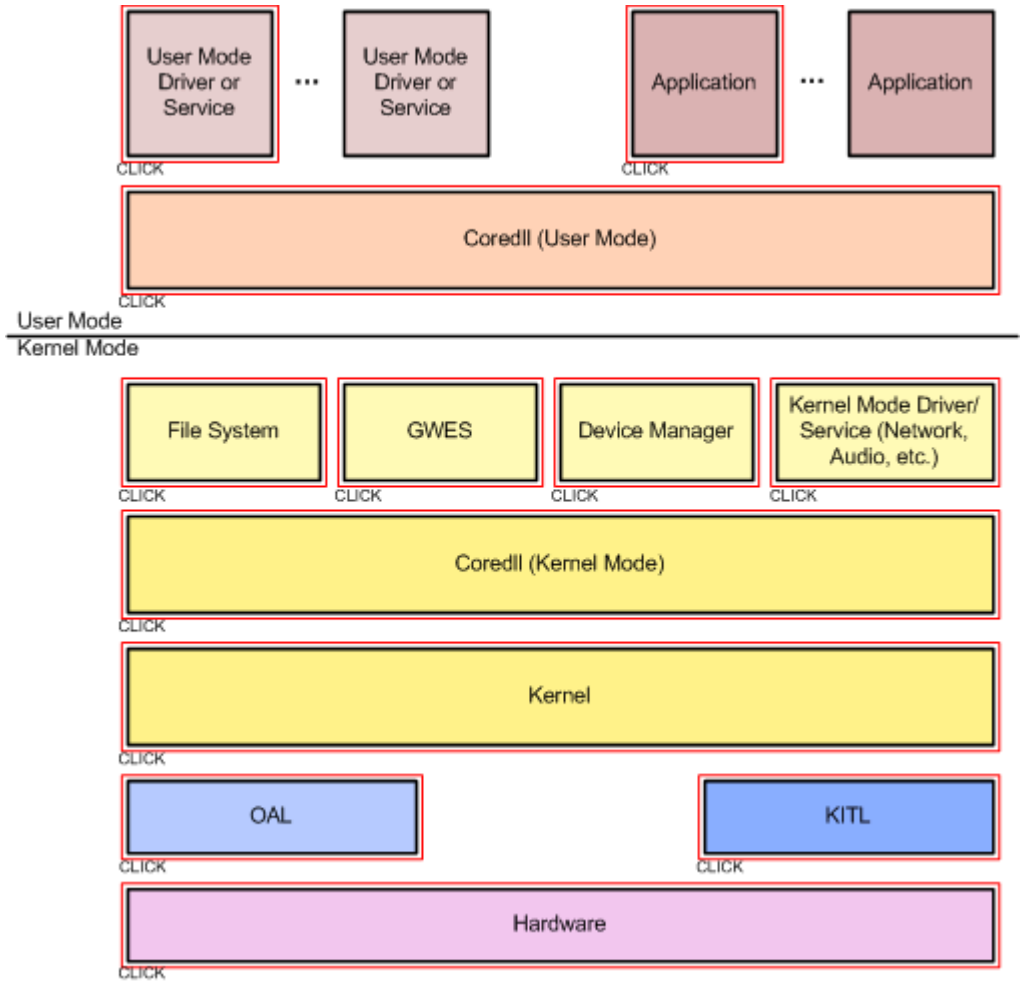
Built on Linux, with a JVM for Applications and extensive set of drivers and protocol stacks, Android enables products by middleware use.



Middleware – Windows Embedded

No one has more Middleware than Windows – one advantage of using Windows Embedded.





Middleware – Windows Embedded

With the same code base as Windows Embedded Compact 7, **Windows Embedded Automotive 7** is a platform that allows automakers and suppliers to deliver visually rich in-car experiences with a shorter time to market. Windows Embedded Automotive 7 includes a large set of integrated and flexible middleware components that allow automotive solutions to scale across a broad range of automotive makes and models. Drivers benefit from the rich user interface and features including state-of-the-art hands-free Bluetooth phone communications, speech commands, touch input, advanced dashboard systems and more.

Middleware – Windows Embedded

Windows Embedded Compact 7 Features

- Deterministic hard real-time operating system
- Separate kernel-mode and user-mode spaces, keeping critical code separate from noncritical code to enhance security and stability
- FAT12, FAT16, FAT32, and exFAT file systems
- UDF and UDFS v2.5 file systems to support CD, DVD, and HDDVD
- Graphic and multimedia core components (GUI)
- Touch and Gesture user interface
- Video and Audio capture pipeline
- VoIP, Cellcore components to access mobile communication resources.
- Web, FTP and file servers
- Networking with Bluetooth, NDIS, TCP/IP, and Wi-Fi
- USB Host and Device