# Embedded Linux System

# Out Line

- CH01 Introduction to Embedded System
- CH02 Embedded Linux (1)
- CH03 Embedded Linux (2)
- CH04 Basic Software And Tool
- CH05 Cross-compile Toolchain
- CH06 Introduction to Bootloader (u-boot)
- CH07 Embedded Linux Kernel
- CH08 RootFS
- CH09 Linux Device Driver
- CH10 Control Hardware Driver

# Introduction to Embedded System

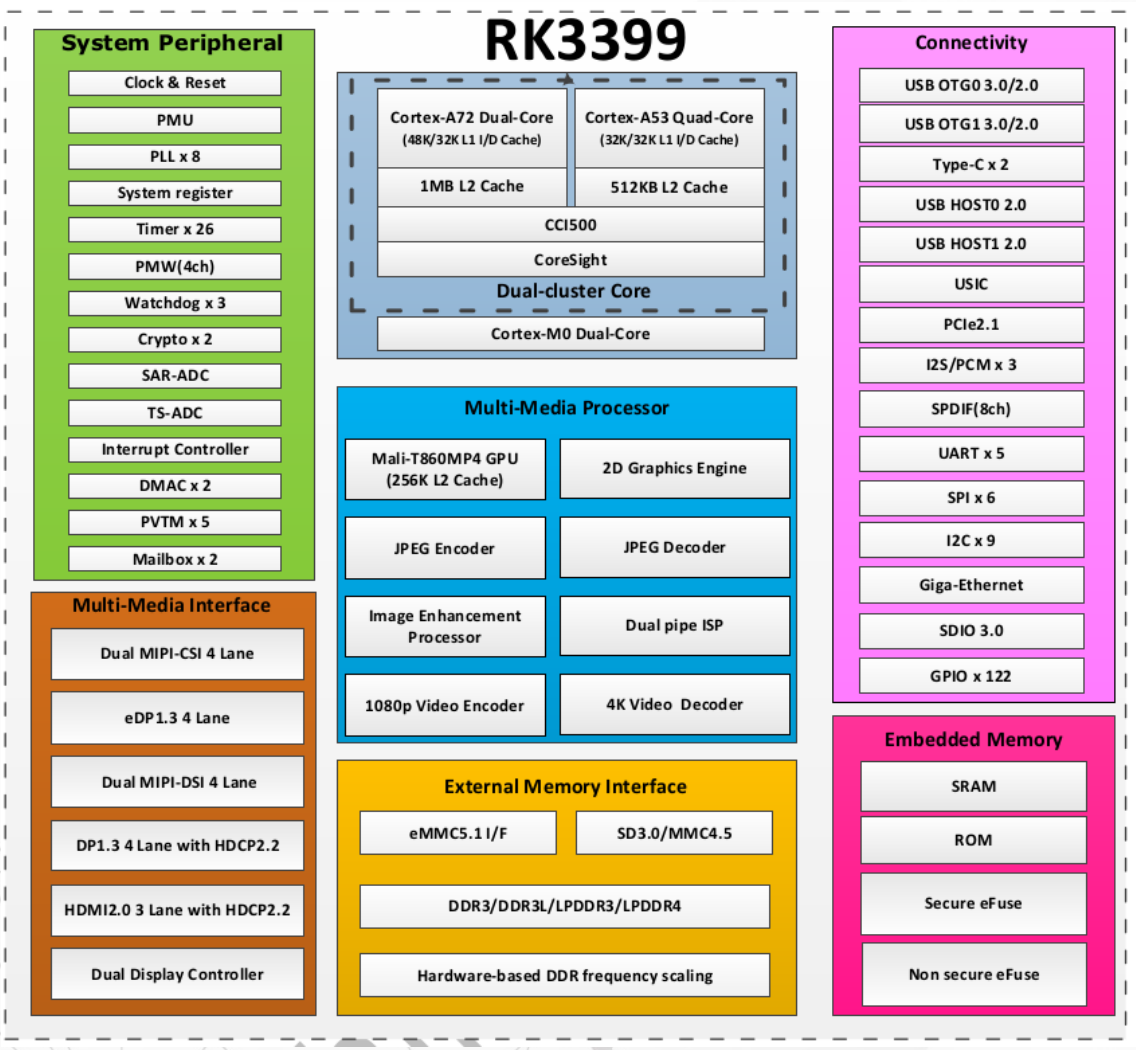
# Embedded System

- An embedded system
  - combination of computer hardware and software
  - specifically designed for a particular function
- Applications
  - Mobile phone
  - Digital camera
  - Smart TV
  - Navigation system

# Feature

- Designed to do some specific task
  - Low power
  - Small size
  - Special operating ranges
  - Low cost
- Install OS ?

# SOC RK3399



# SOC – System On Chip

- Processor

  → ARM, X86, MIPS ….

- RAM

  → 8MB ~ 4 GB

- Storagee

  → Nand, Nor flash

  → SD/MMC/eMMc

- System Bus

  → AMBA, AHB, APB, AXI ...

# SOC – System On Chip

- Communication
  - I2C, I2S, USB, PCI/PCIe ...
- Media system
  - JPEG, H.264 ..
- System component
  - DMA, RTC ..

# Embedded Linux ?

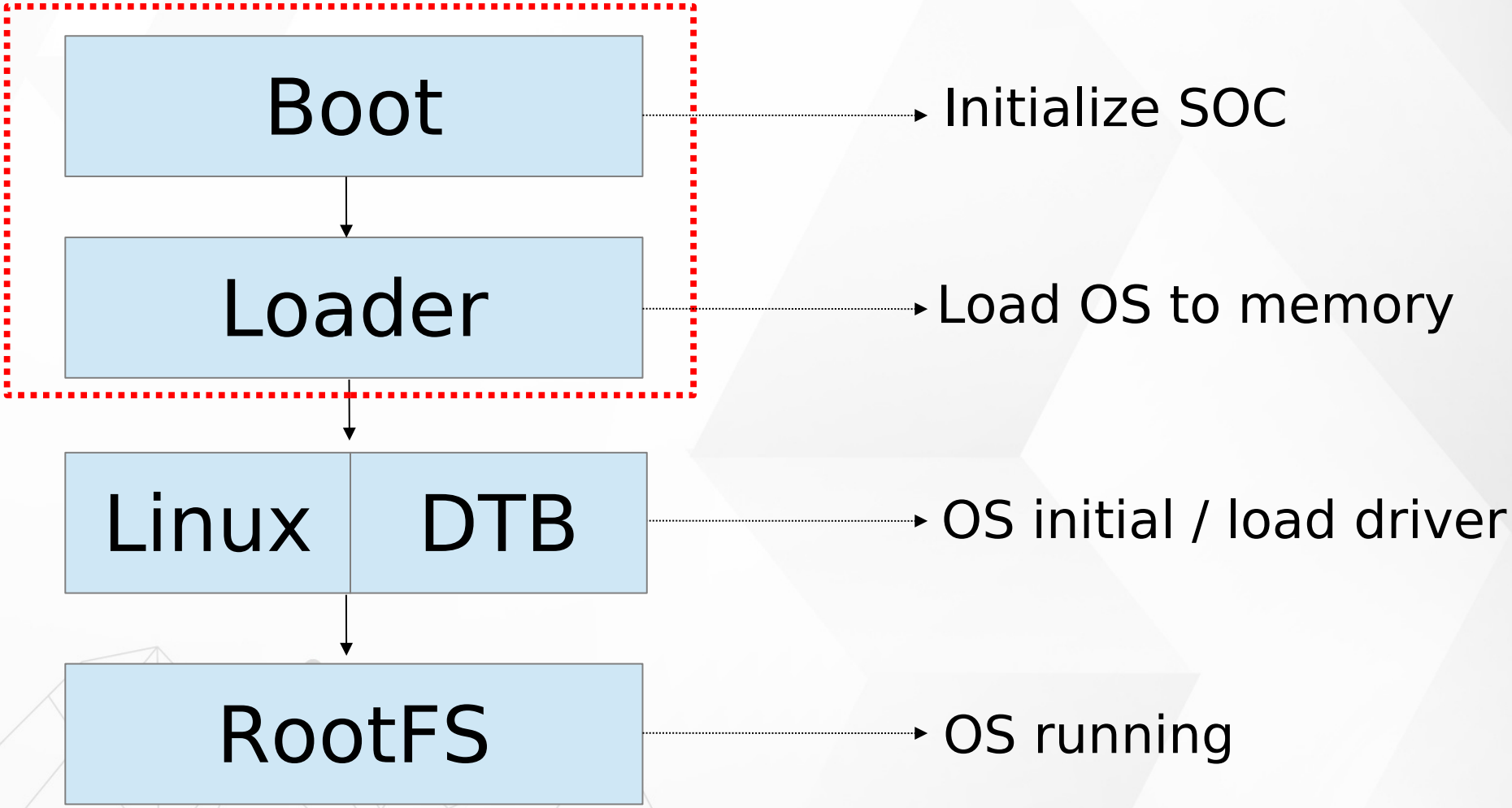
Embedded Linux is the usage of the Linux kernel and various open-source components in **embedded systems** (from Free Electrons)

# Linux Advantages

- Re-use components
- Quickly design and develop complicated products
- No need to re-develop components

  → TCP/IP stack, USB stack, PCI stack ...
- Allow you modify components

# Embedded Linux Booting



# Software Components

- Cross-compilation Toolchain
- Boot-loader
- Linux Kernel, DeviceTree
- RootFS
- C library
- Libraries and applications
- BSP (Board Support Package)

# Develop Environment

# Develop Environment

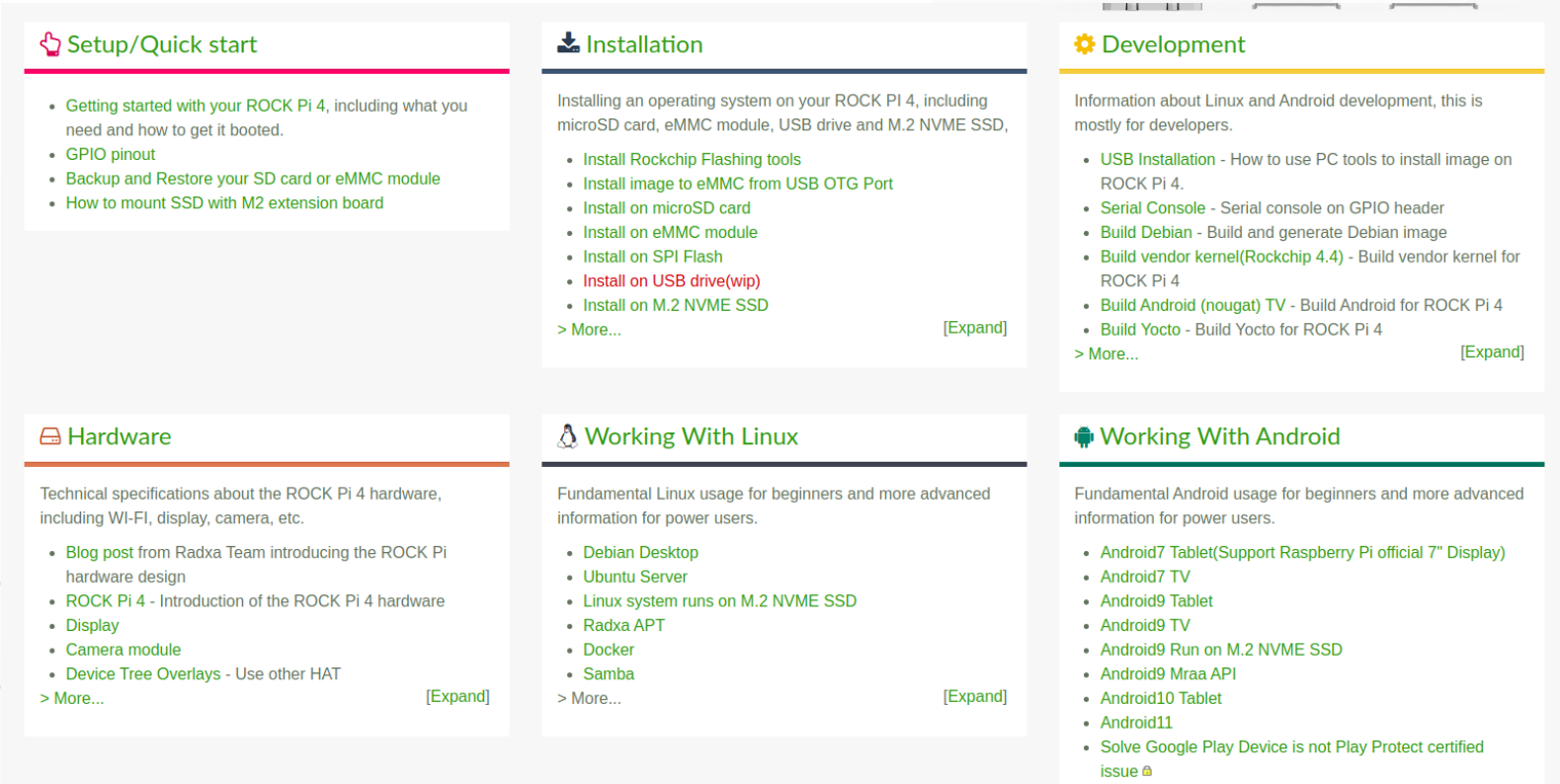
- Host PC (Linux)
- Toolchain
- BSP (Board Support Package)
- Target Board EVB (RockPi4)

# BSP

- Board Support Package
- From chip vendor
  - Distribution
    - Bootloader
    - Linux Kernel
    - Device Driver
    - Rootfs

# RK3399 Debian BSP (1)

- RK3399 Debian BSP
  - https://wiki.radxa.com/Rockpi4



### Setup/Quick start

- Getting started with your ROCK Pi 4, including what you need and how to get it booted.
- GPIO pinout
- Backup and Restore your SD card or eMMC module
- How to mount SSD with M2 extension board

### Installation

Installing an operating system on your ROCK PI 4, including microSD card, eMMC module, USB drive and M.2 NVME SSD,

- Install Rockchip Flashing tools
- Install image to eMMC from USB OTG Port
- Install on microSD card
- Install on eMMC module
- Install on SPI Flash
- Install on USB drive(wip)
- Install on M.2 NVME SSD

> More... [Expand]

### Development

Information about Linux and Android development, this is mostly for developers.

- USB Installation - How to use PC tools to install image on ROCK Pi 4.
- Serial Console - Serial console on GPIO header
- Build Debian - Build and generate Debian image
- Build vendor kernel(Rockchip 4.4) - Build vendor kernel for ROCK Pi 4
- Build Android (nougat) TV - Build Android for ROCK Pi 4
- Build Yocto - Build Yocto for ROCK Pi 4

> More... [Expand]

### Hardware

Technical specifications about the ROCK Pi 4 hardware, including WI-FI, display, camera, etc.

- Blog post from Radxa Team introducing the ROCK Pi hardware design
- ROCK Pi 4 - Introduction of the ROCK Pi 4 hardware
- Display
- Camera module
- Device Tree Overlays - Use other HAT

> More... [Expand]

### Working With Linux

Fundamental Linux usage for beginners and more advanced information for power users.

- Debian Desktop
- Ubuntu Server
- Linux system runs on M.2 NVME SSD
- Radxa APT
- Docker
- Samba

> More... [Expand]

### Working With Android

Fundamental Android usage for beginners and more advanced information for power users.

- Android7 Tablet(Support Raspberry Pi official 7" Display)
- Android7 TV
- Android9 Tablet
- Android9 TV
- Android9 Run on M.2 NVME SSD
- Android9 Mraa API
- Android10 Tablet
- Android11
- Solve Google Play Device is not Play Protect certified issue

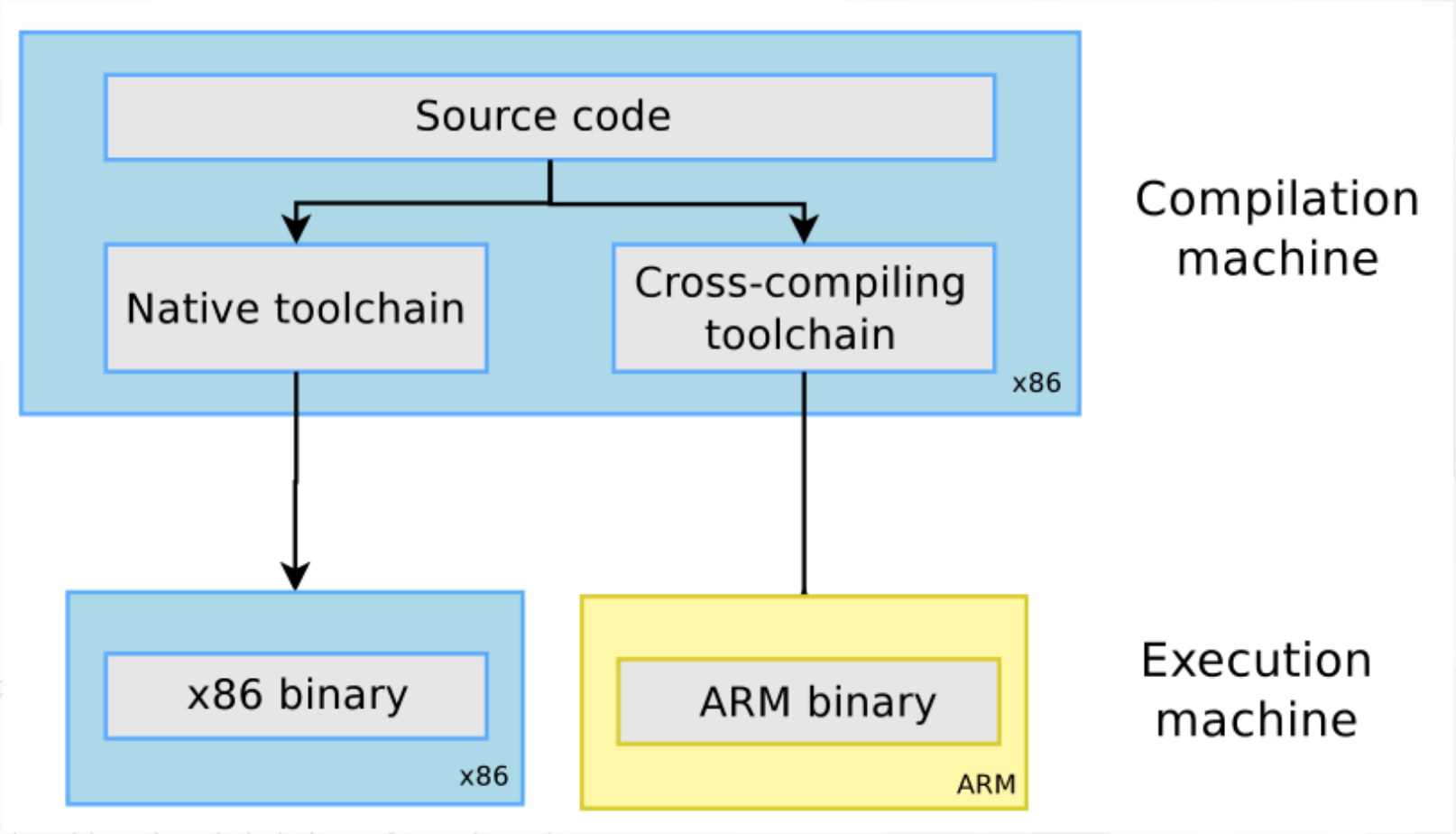
# RK3399 Debian BSP (2)

- Boot-Loader
  - RKBin, U-Boot
- Kernel
  - Linux Kernel source
- Rootfs
  - Debian File System
- Tool-Chain
  - Compile tool

# Cross Compilation

- Native Environment
  - Host x86PC (Linux)
  - gcc
- Cross-Compile
  - aarch64-linux-gnu-gcc

# Cross Compilation



# RK3399 and SOC

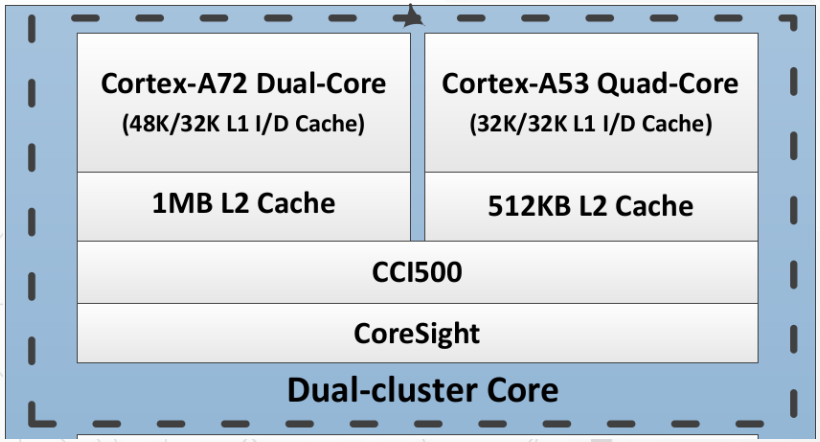
# RockPi WiKi

- Rock Pi4 Wiki
  - https://wiki.radxa.com/Rockpi4
- Rock Pi 4 Feture
  - https://wiki.radxa.com/Rockpi4/getting_started
- Rock Pi 4 Debin
  - https://wiki.radxa.com/Rockpi4/Debian
  - https://wiki.radxa.com/Rock4/downloads

# CPU

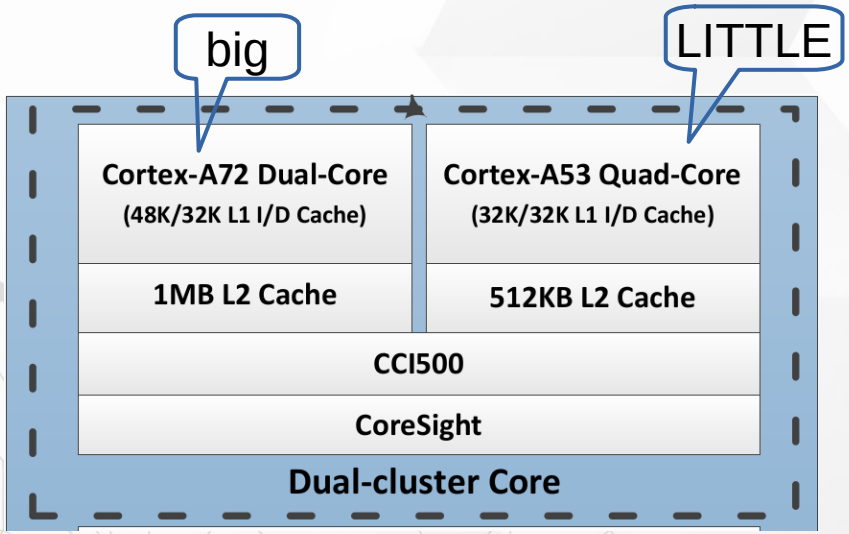
# CPU (1)

- Two CPU clusters
  - Big cluster with dual-core Cortex-A72
    - high-performance
  - Little cluster with quad-core Cortex-A53
    - low power



# CPU (2)

- Arm big.LITTLE technology
  - ”LITTLE” processors are designed for maximum power efficiency
  - ”big” processors are designed to provide maximum compute performance.



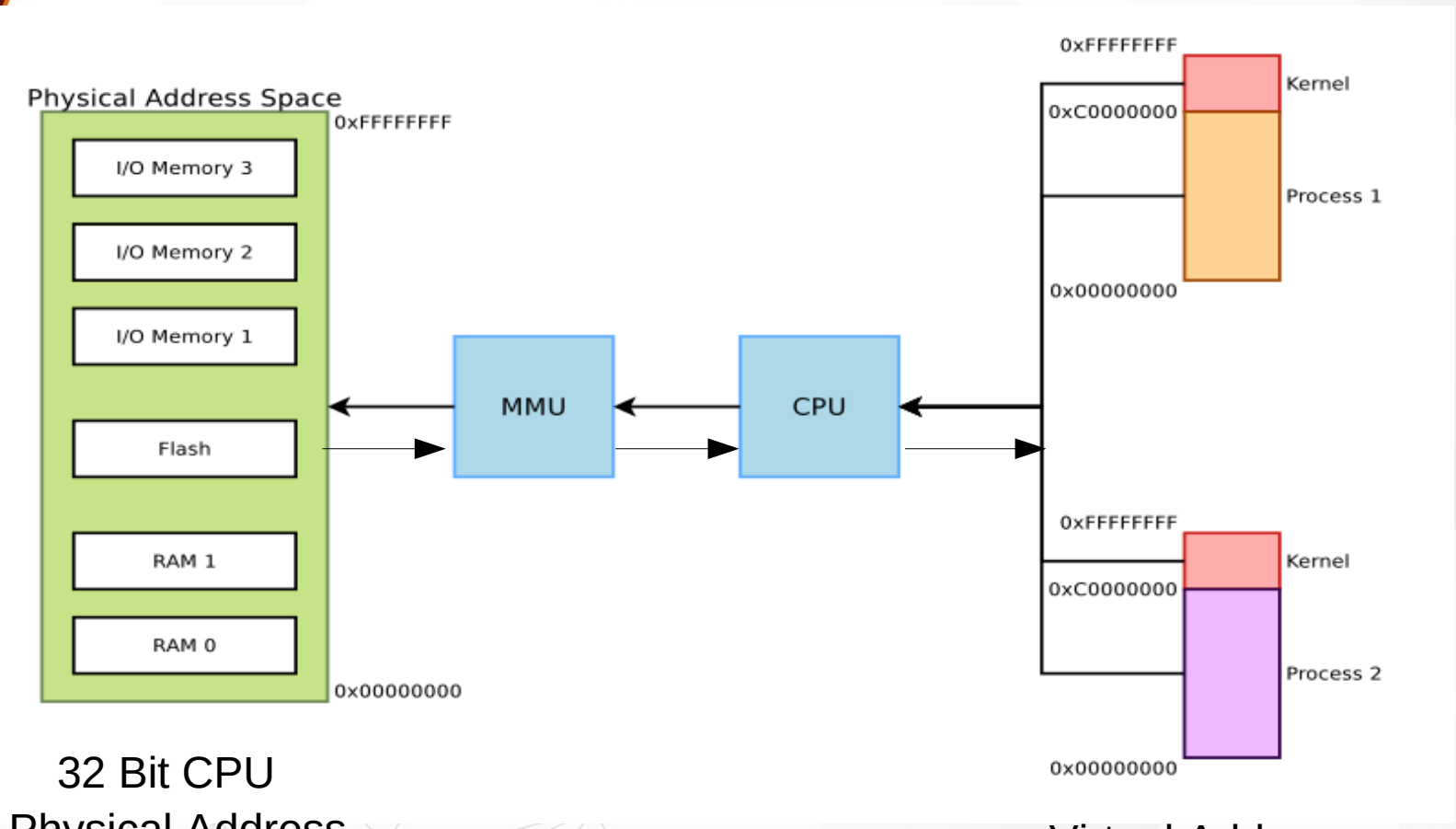

https://www.arm.com/zh-TW/technologies/big-little

# Memory

# Memory

- Internal ROM
  - Internal BootRom (Size : 32KB)
  - boot from
    - SPI, eMMC, SD/MMC
- Internal RAM
  - 200KB
- External
  - DDR3/DDR3L/LPDDR3/LPDDR4
  - SPI NOR/NAND Flash
  - EMMC5.1
  - SD3.0/MMC4.51

# MMU (1)

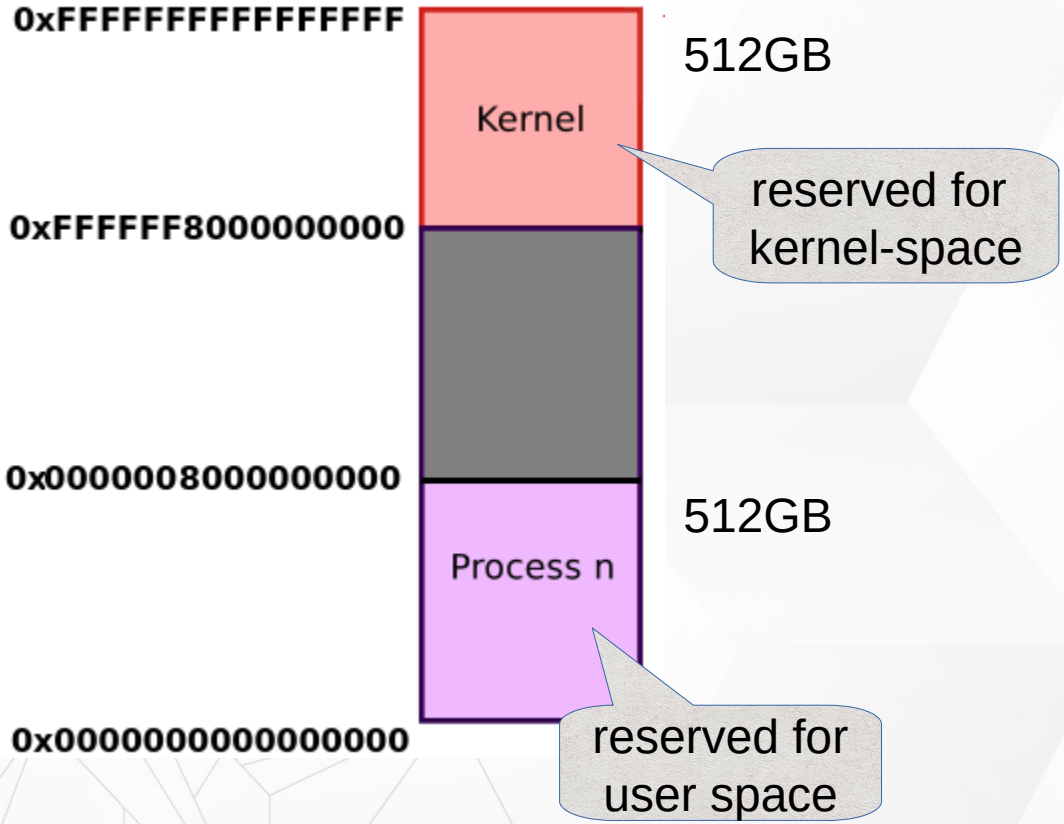



32 Bit CPU
Physical Address

Virtual Address

# MMU (2)

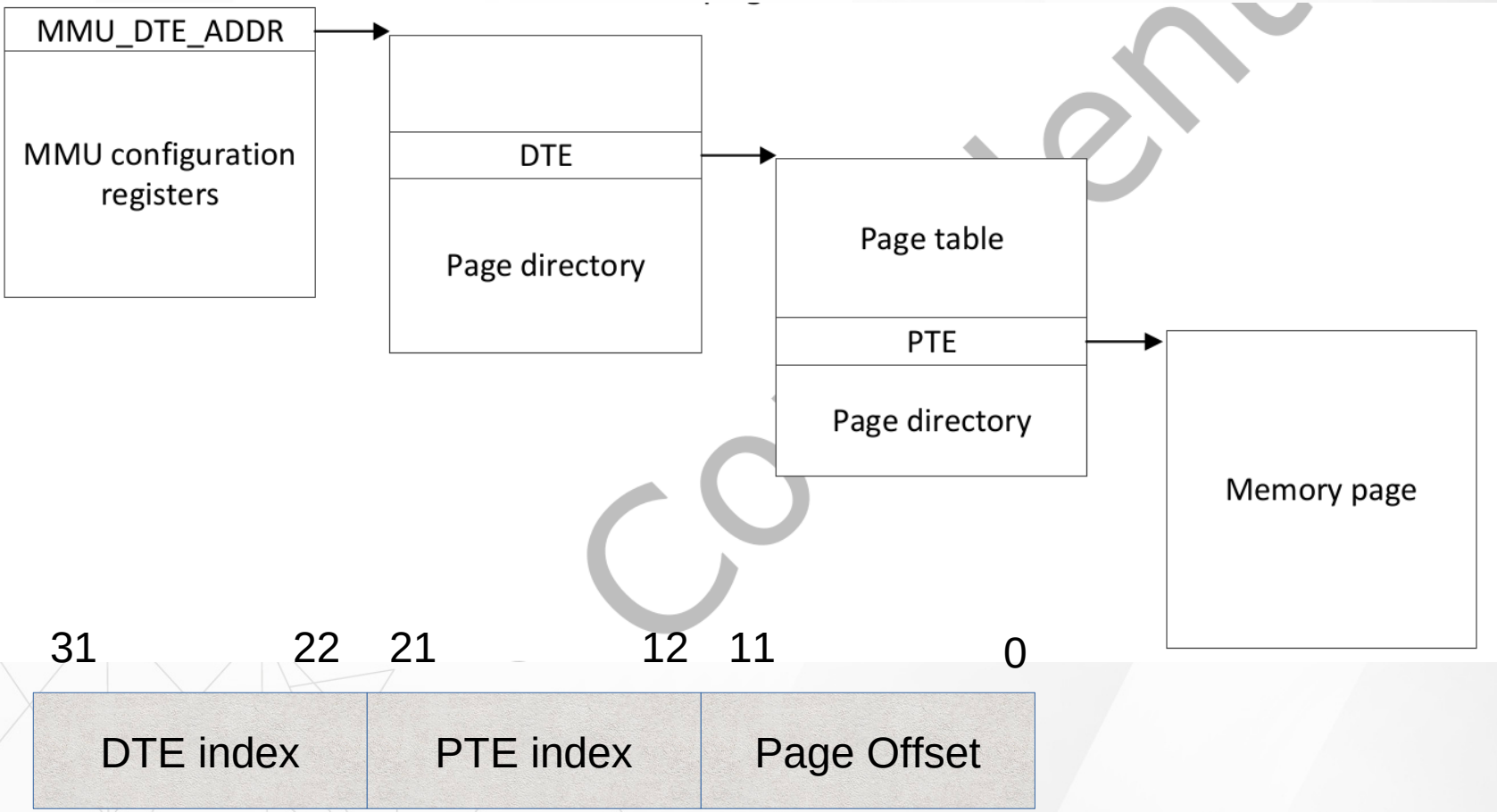
64 Bit CPU　AArch64 Linux memory layout with 4KB pages + 3 levels



# MMU (3)

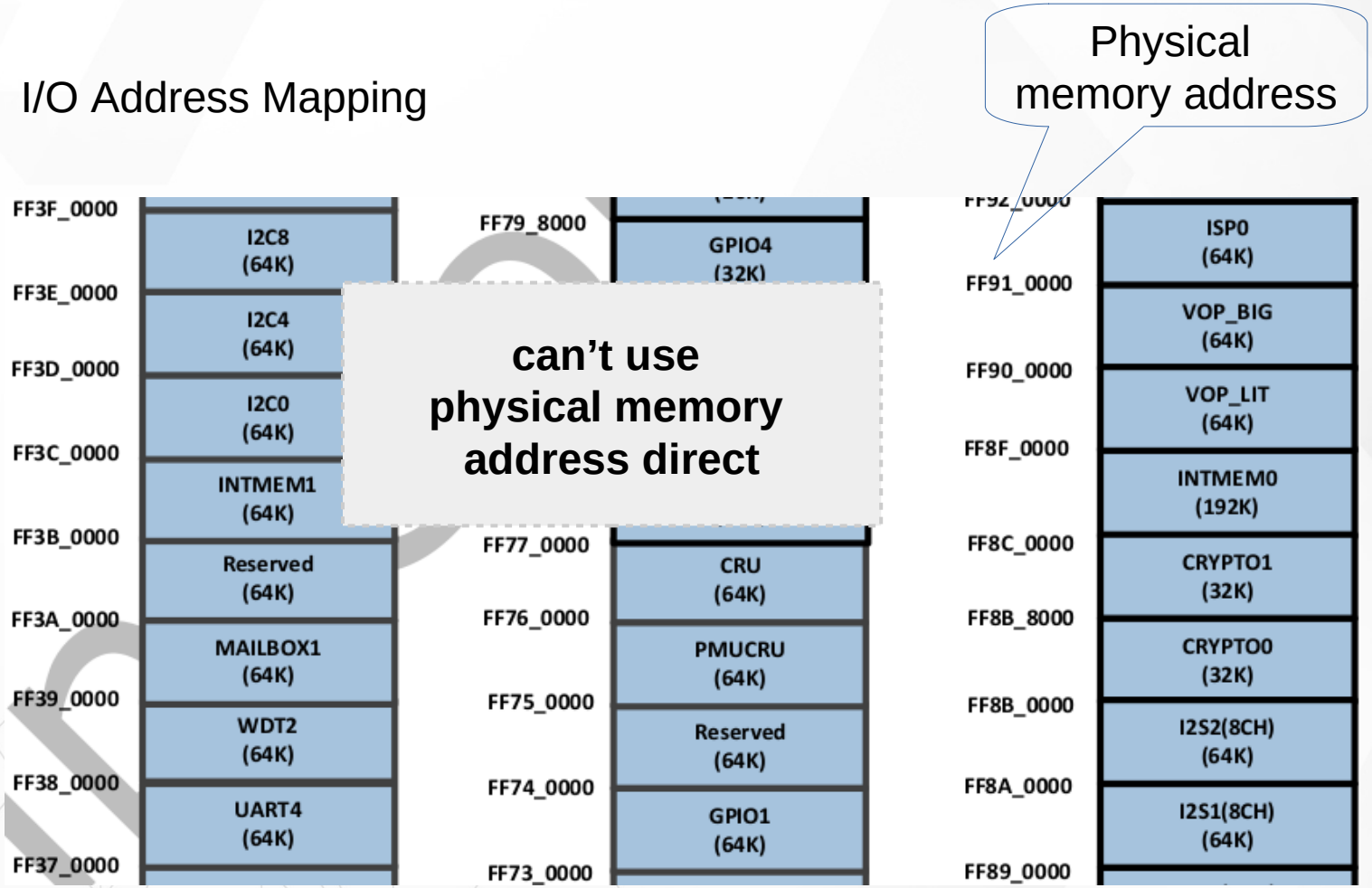
- The MMU divides memory into **4KB** pages
- 2-level page table structure
  - The First level
    - Page Directory consists of **1024** Directory  Table Entries (DTEs)
    - Each pointing to a Page Table.
  - The Second level
    - The Page Table consists of **1024** Page Table Entries (PTEs)
    - Each pointing to a page in memory

# MMU (4)



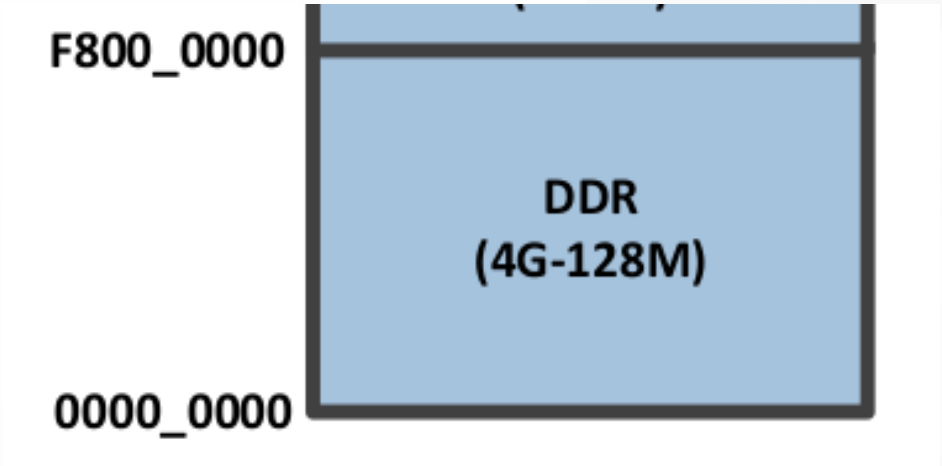
# MMU (5)

I/O Address Mapping



# MMU (6)

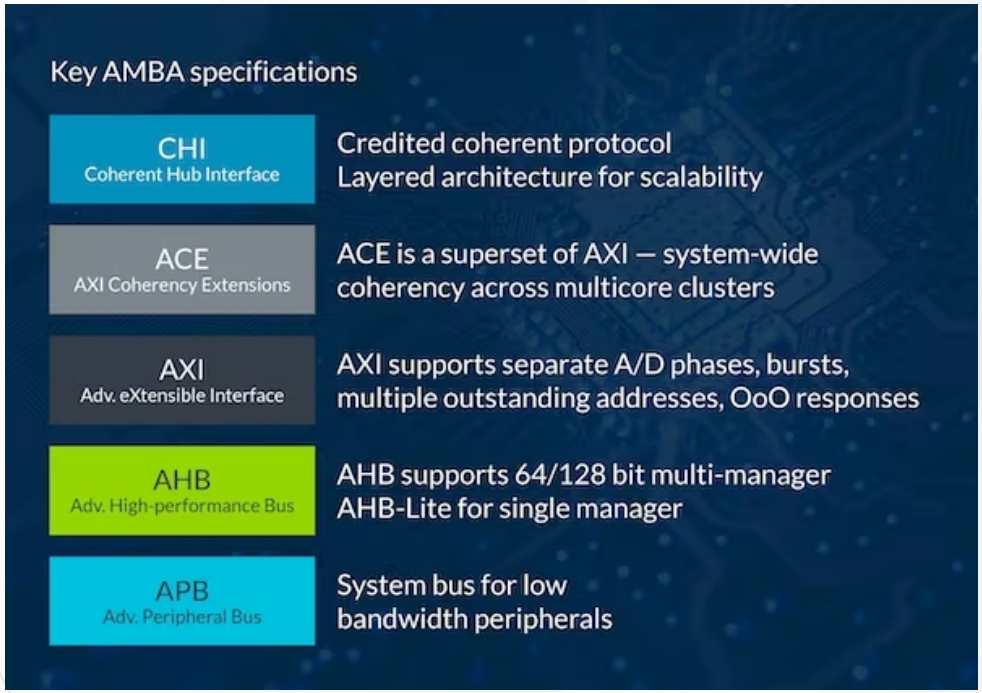
**DDR SD RAM Memory**



# Interconnect Connect

# Bus Architecture

- AMBA : Advanced Microcontroller Bus Architecture
  - AXI
  - AHB
  - APB




https://www.arm.com/zh-TW/architecture/system-architectures/amba

# GPU

# Graphics Engine (1)

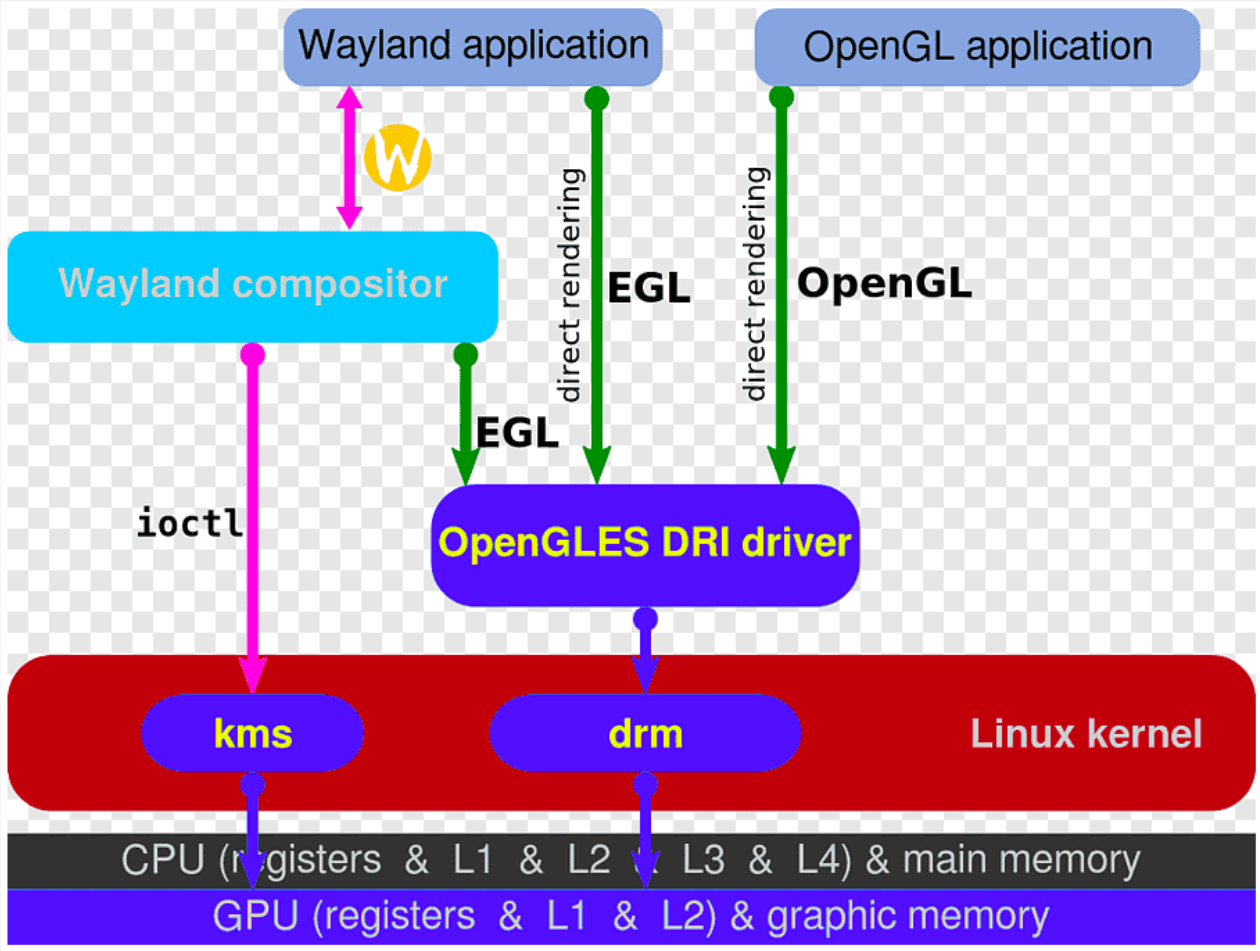
- Graphics Process Unit
- Mali-T860MP4 GPU
  - OpenGL ES1.1/2.0/3.0, OpenCL1.2,
  - 3DGraphics Engine
  - 2D Graphics Engine

# Graphics Engine (2)

- OpenGL
  - Open Graphics Library
- OpenGL ES
  - OpenGL for Embedded Systems
- EGL
  - Native Platform Graphics Interface

https://www.khronos.org/egl

# Graphics Engine (3)




https://en.wikipedia.org/wiki/EGL_%28API%29#/media/File:Wayland_display_server_protocol.svg

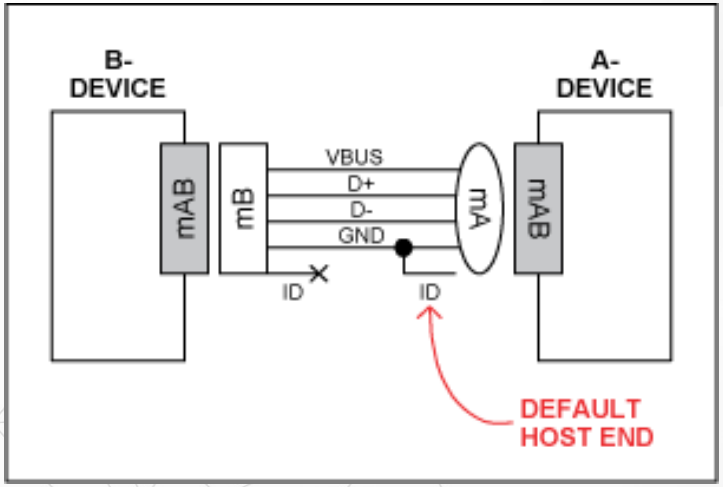
# Connect

# USB (1)

- USB Host
  - RK3399
  - OHCI : 1.1.  Hardware Complex
  - UHCI : 1.0,  1.1  Software Complex
  - EHCI : 2.0
  - XHCI : 3.0
- USB Device
  - USB Storage

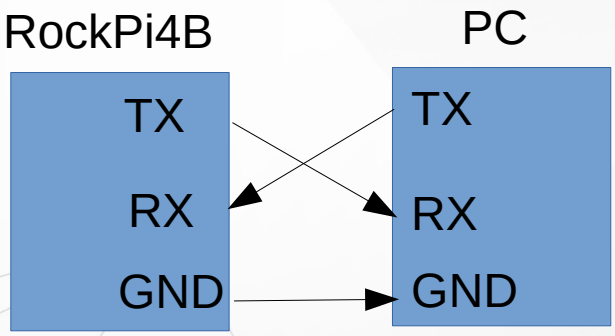
# USB (2)

- USB OTG
  - USB_ID 信號爲低時，該設備應作爲 Host
  - USB_ID 信號爲高時，該設備作爲 Slave



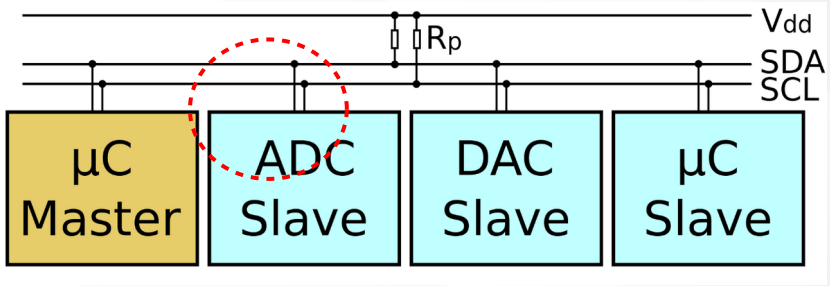
# UART

- The **U**niversal **A**synchronous **R**eceiver/**T**ransmitter
  - Write Data
    - CPU → Data → APB → UART
  - Read Data
    - Data → UART → APB → CPU

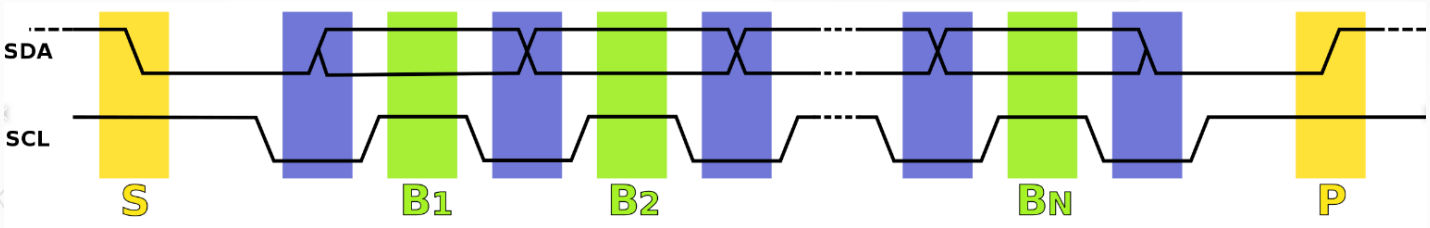


# I2C (1)

- Serial bus
  - SDA data line
  - SCL clock line



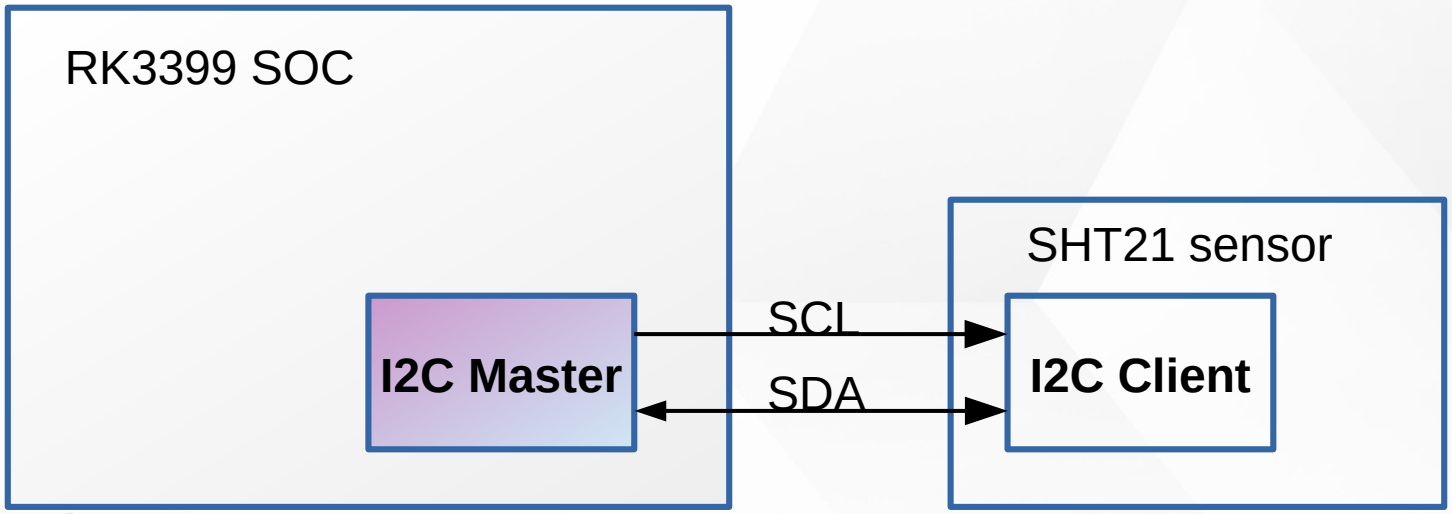

Protocol




https://en.wikipedia.org/wiki/I%C2%B2C

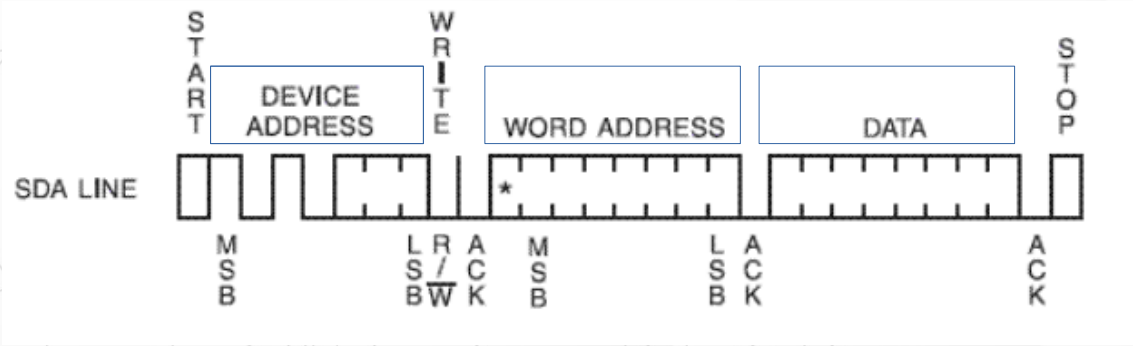
# I2C (2)

Master and Client



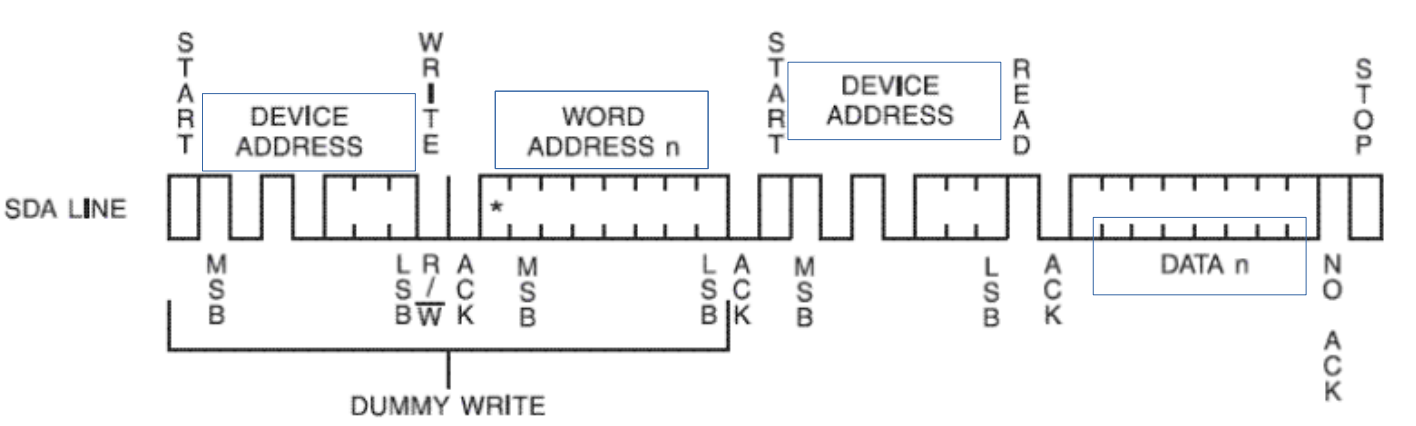
# I2C protocol - Write

- Write
  - byte write
  - page write
- Device address
- Read/write bit : 0
- ACK



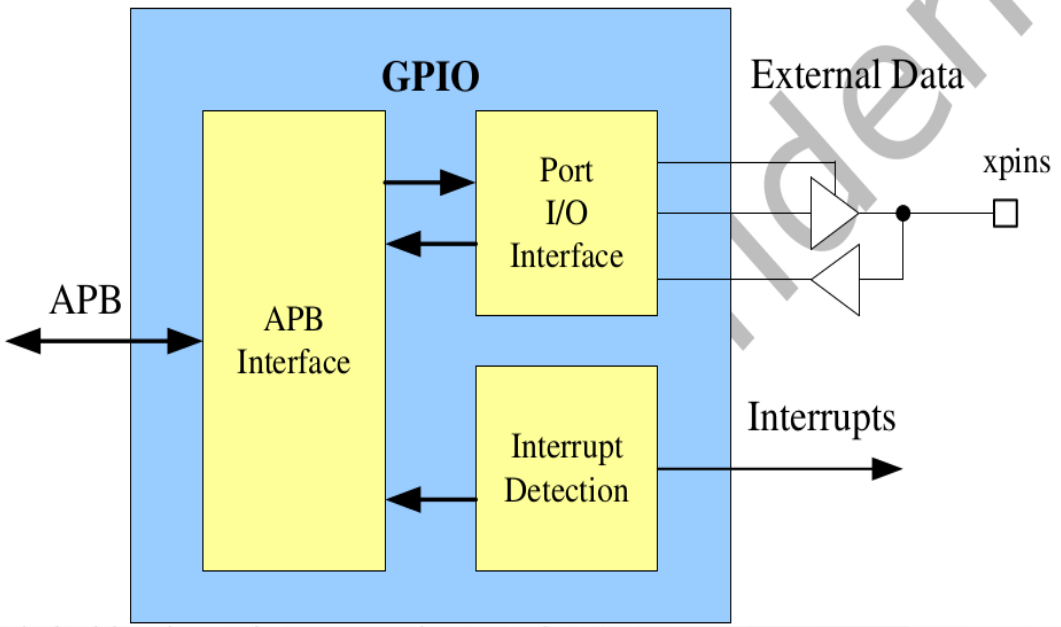
# I2C protocol - Read

- Read
  - byte read
  - page read
- Device address
- **Read/write bit : 1**
- ACK



# GPIO

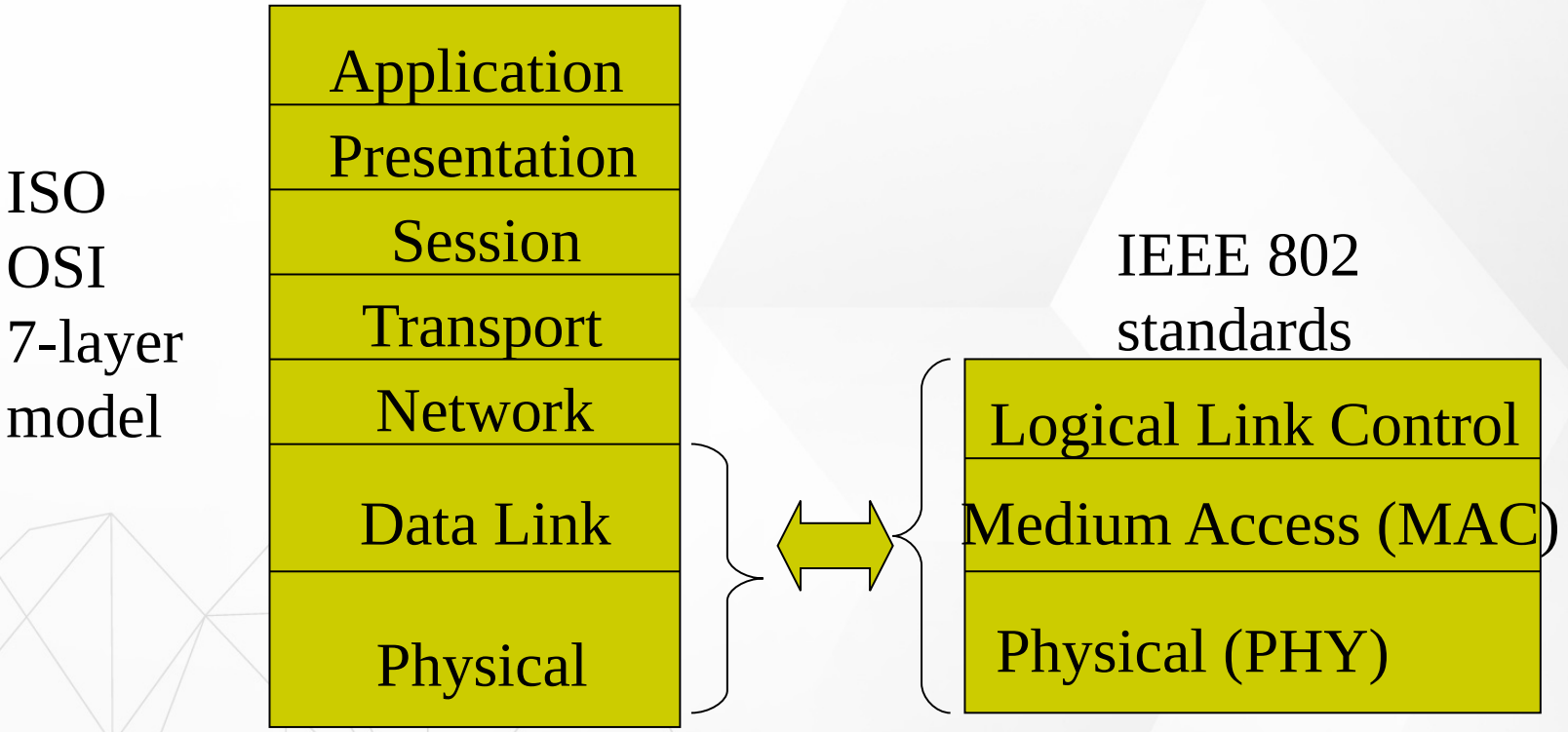
- General Purpose Programming I/O
- GPIO controls the output data and direction of external I/O pads



# WiFi Basic

# WIFI – 802.11 (1)

- IEEE 802.11 – 無線區域網路




https://ptolemy.berkeley.edu/projects/ofdm/ergen/docs/IEEE-802.11overview.ppt

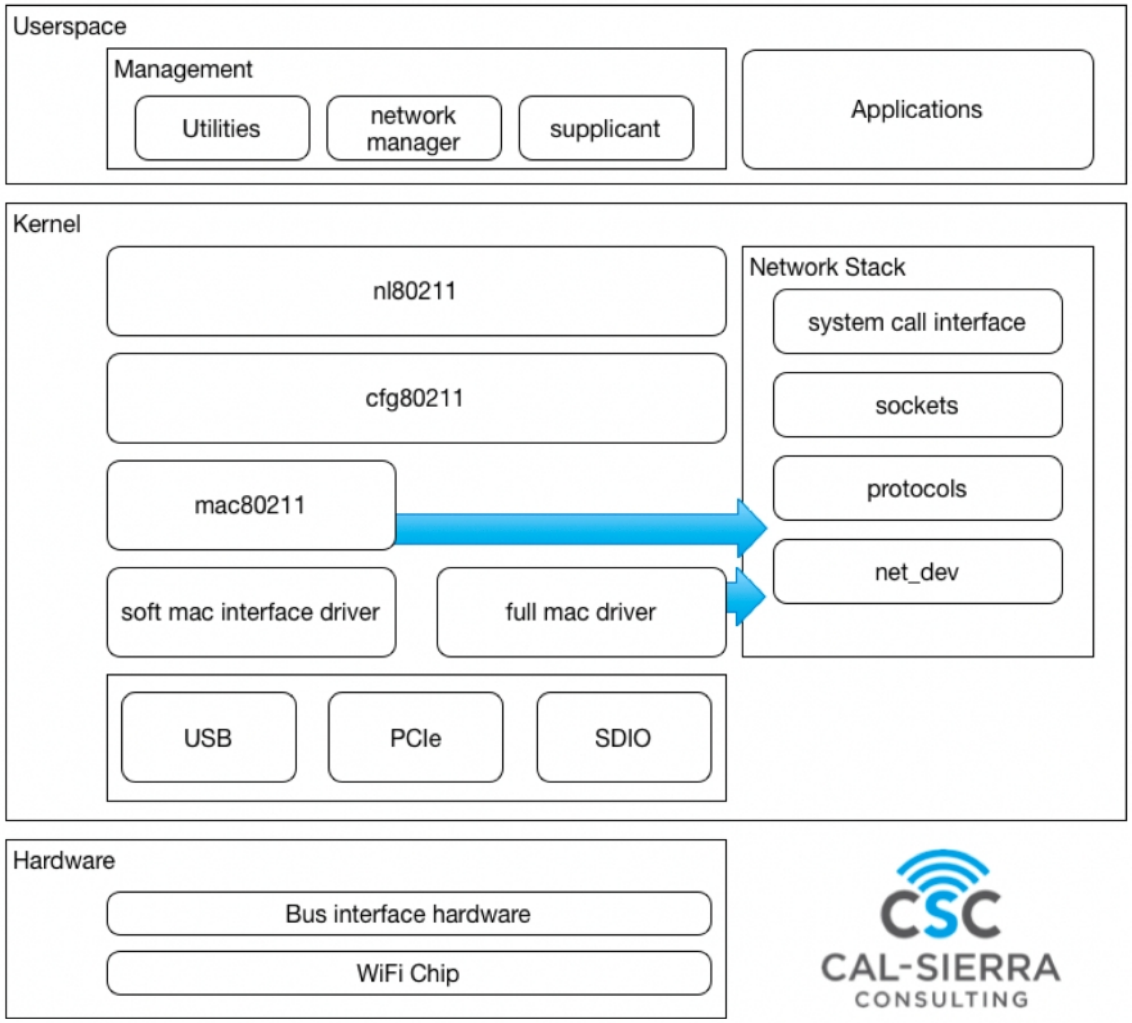
# WIFI – 802.11 (2)

| 世代名稱[註 1] | IEEE標準 | | 最大速率（Mbit/s） | 頻率（GHz） |
|---|---|---|---|---|
| | 名稱 | 發布年份 | | |
| **Wi-Fi 7** | 802.11be | (2024) [註 2] | 1376～46120 | 2.4/5/6 |
| **Wi-Fi 6E** | 802.11ax | 2020 | 574～9608[1] | 6 [2] |
| **Wi-Fi 6** | | | 2019 | 2.4/5 |
| **Wi-Fi 5** | 802.11ac | 2014 | 433～6933 | 5 [3] |
| **Wi-Fi 4** | 802.11n | 2008 | 72～600 | 2.4/5 |
| ***Wi-Fi 3***[註 3] | 802.11g | 2003 | 6～54 | 2.4 |
| ***Wi-Fi 2***[註 3] | **802.11a** | 1999 | 6～54 | 5 |
| ***Wi-Fi 1***[註 3] | 802.11b | 1999 | 1～11 | 2.4 |
| ***Wi-Fi 0***[註 3] | 802.11 | 1997 | 1～2 | 2.4 |

1. ^ Wi-Fi是Wi-Fi联盟的商標
2. ^ 預定的發布年份
3. ^ 3.0 3.1 3.2 3.3 Wi-Fi联盟未定義 Wi-Fi 0/1/2/3 的世代名稱[4][5]

https://zh.wikipedia.org/wiki/IEEE_802.11a-1999

# WIFI – Linux & 802.11 (1)




CSC CAL-SIERRA CONSULTING

https://hackmd.io/@0xff07/BJkDzazVK#iwd



中華行動數位科技 Chinese Action Digital Technology

# WIFI – Linux & 802.11 (2)

- mac80211
  - most associated to hardware offloading
  - the **802.11 protocol state machine lives here**
- cfg80211
  - middle-layer **Handles Everything Configurable**
- nl80211
  - The **API between user-land and kernel-land**
  - Relies on the **netlink** protocol to exchange messages between the two worlds

https://www.linuxembedded.fr/2020/05/emulating-wlan-in-linux-part-i-the-80211-stack

# WIFI – Tool

- Show / manipulate wireless devices and their configuration
  - iw
- For connecting to a WPA/WPA2 network
  - wpa_supplicant

# Audio

# ALSA Overview

- **A**dvanced **L**inux **S**ound **A**rchitecture
  - Linux kernel
  - Software framework
- Sound Servers
  - PulseAudio, JACK ...
- ALSA stream is a data flow representing sound
  - PCM (Pulse-code modulation)

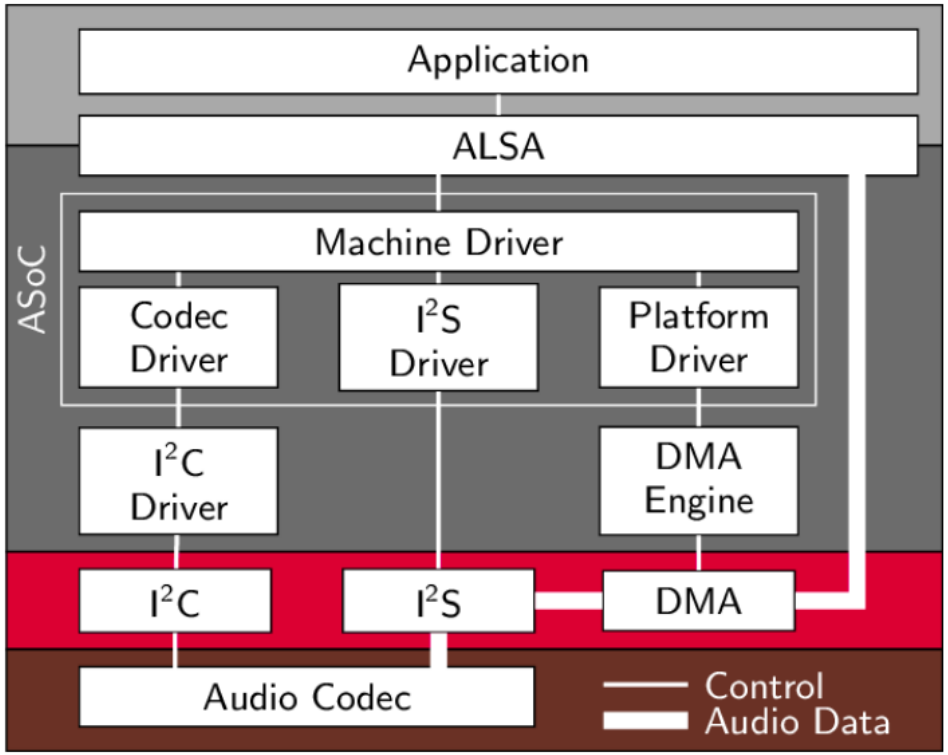
https://en.wikipedia.org/wiki/Advanced_Linux_Sound_Architecture

# ALSA Overview

- Audio Codec
  - Audio Codec 就是音樂訊號（ Audio ）壓縮 / 解壓縮 (Compress/DECompress)
    的演算法或程式，前後加起來就是 Audio Codec.
- Parameters of the hardware
  - sampling rate : 44100 Hz
  - sample width : 8 bit, 16 bit, 24 bit
  - sample encoding :  endianness
  - number of channels : 1 channel, 2 channel ...

https://en.wikipedia.org/wiki/Advanced_Linux_Sound_Architecture

# ALSA Overview



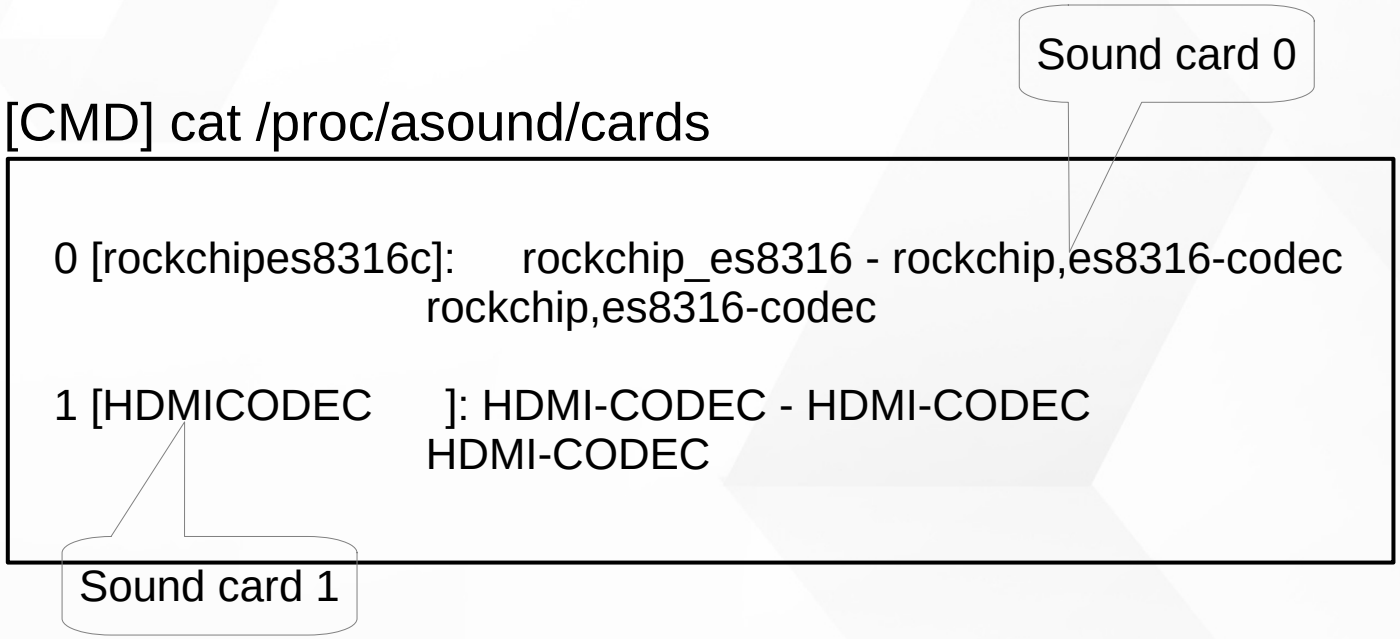

https://www.researchgate.net/figure/Structure-of-ASoC-and-the-embedment-into-the-Linux-audio-framework_fig2_262112720

# Sound Card in Linux (1)

[CMD] ls /proc/asound/ -l

```
lrwxrwxrwx      1 root root 5 Apr 30 06:51 HDMICODEC -> card1
dr-xr-xr-x      4 root root 0 Apr 30 06:51 card0
dr-xr-xr-x      3 root root 0 Apr 30 06:51 card1
-r--r--r--      1 root root 0 Apr 30 06:51 cards
-r--r--r--      1 root root 0 Apr 30 06:51 devices
-r--r--r--      1 root root 0 Apr 30 06:51 hwdep
-r--r--r--      1 root root 0 Apr 30 06:51 pcm
lrwxrwxrwx      1 root root 5 Apr 30 06:51 rockchipes8316c -> card0
dr-xr-xr-x 2 root root 0 Apr 30 06:51 seq
-r--r--r--      1 root root 0 Apr 30 06:51 timers
-r--r--r--      1 root root 0 Apr 30 06:51 version
```

# Sound Card in Linux (2)



# Sound Card in Linux (3)

rockchipes8316c

[CMD] ls -l /proc/asound/card0/

```
-r--r--r-- 1 root root 0 Apr 30 06:59 id
dr-xr-xr-x 3 root root 0 Apr 30 06:59 pcm0c
dr-xr-xr-x 3 root root 0 Apr 30 06:59 pcm0p
```

capture

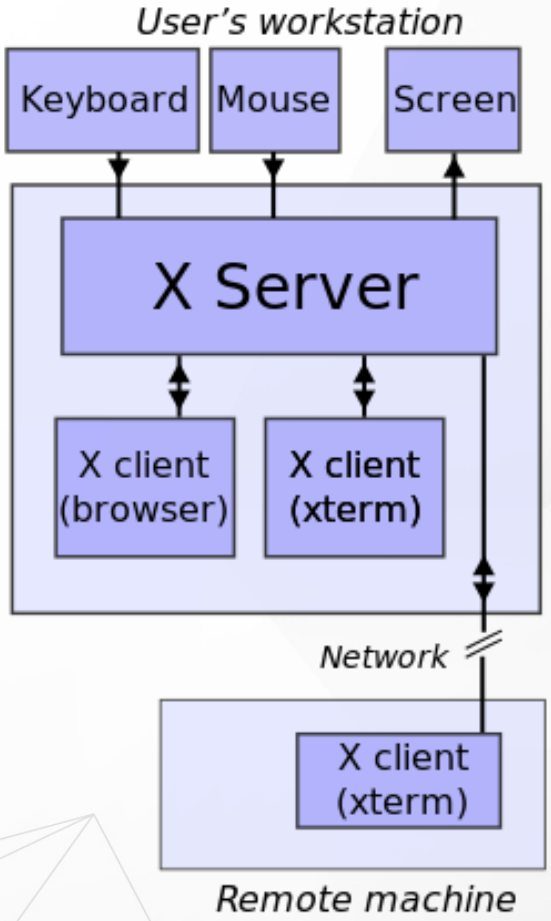
playback

[CMD] ls -l /proc/asound/card1/

```
-r--r--r-- 1 root root 0 Apr 30 07:01 id
dr-xr-xr-x 3 root root 0 Apr 30 07:01
pcm0p
```

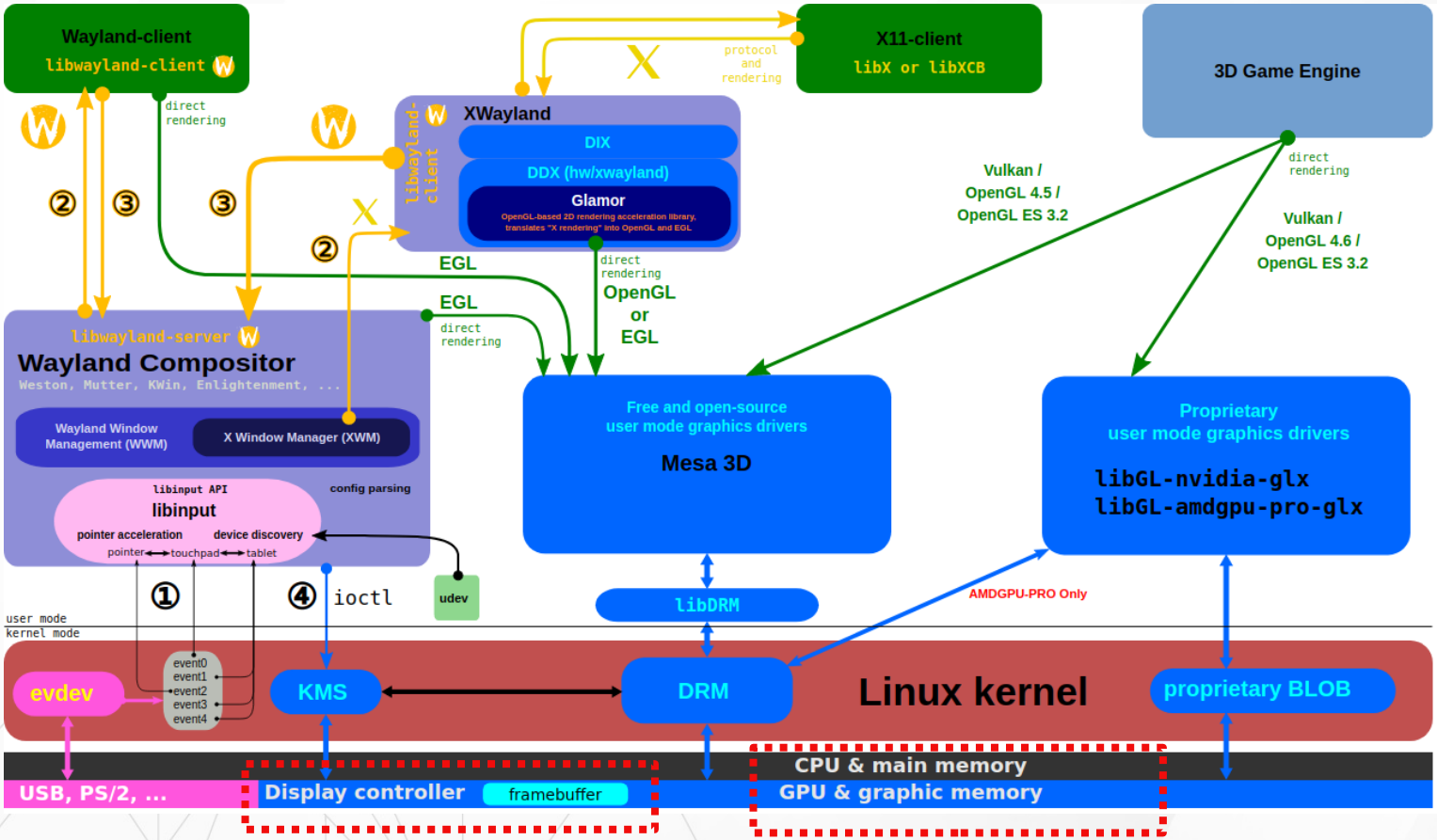
# Linux Display Subsystem

# Linux Windows System (1)




https://en.wikipedia.org/wiki/File:X_client_sever_example.svg

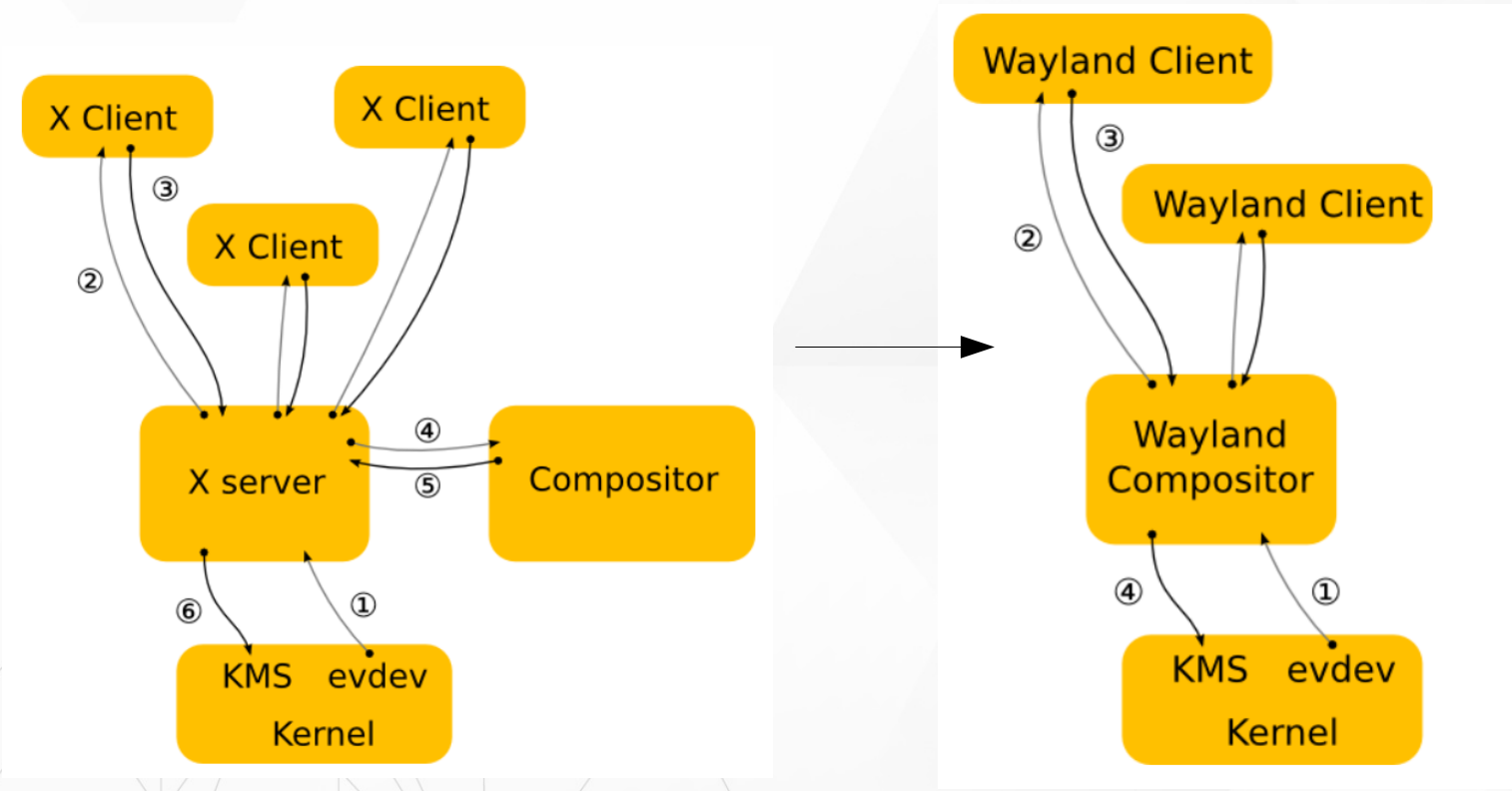
# Linux Windows System (2)




https://upload.wikimedia.org/wikipedia/commons/2/2d/The_Linux_Graphics_Stack_and_glamor.svg

# Linux Windows System (3)

Wayland is a replacement for the X11 window system protocol




https://wayland.freedesktop.org/architecture.html

# GTK and Gnome

- GTK
  - GTK (formerly GTK+) is a free and open-source cross-platform widget toolkit for creating graphical user interfaces (GUIs).
  -
- GNOME
  - GNOME is the default desktop environment of many major Linux distributions
  - originally an acronym for GNU Network Object Model Environment
  - free and open-source desktop environment for Linux and other Unix-like[10] operating systems

https://en.wikipedia.org/wiki/GNOME

https://en.wikipedia.org/wiki/GTK

# OpenGL and EGL

- EGL
  - EGL Native Platform Graphics Interface is an interface portable layer for graphics resource management.
  - works between rendering APIs such as OpenGL ES or OpenVG and the underlying native platform window system
- OpenGL
  - OpenGL (Open Graphics Library) is a cross-language, cross-platform application programming interface (API) for rendering 2D and 3D vector graphics
- Mesa
  - Mesa, also called Mesa3D and The Mesa 3D Graphics Library
  - it is an open source implementation of OpenGL, Vulkan, and other graphics API specifications

https://en.wikipedia.org/wiki/OpenGL

https://en.wikipedia.org/wiki/Mesa_(computer_graphics)

https://www.khronos.org/egl/

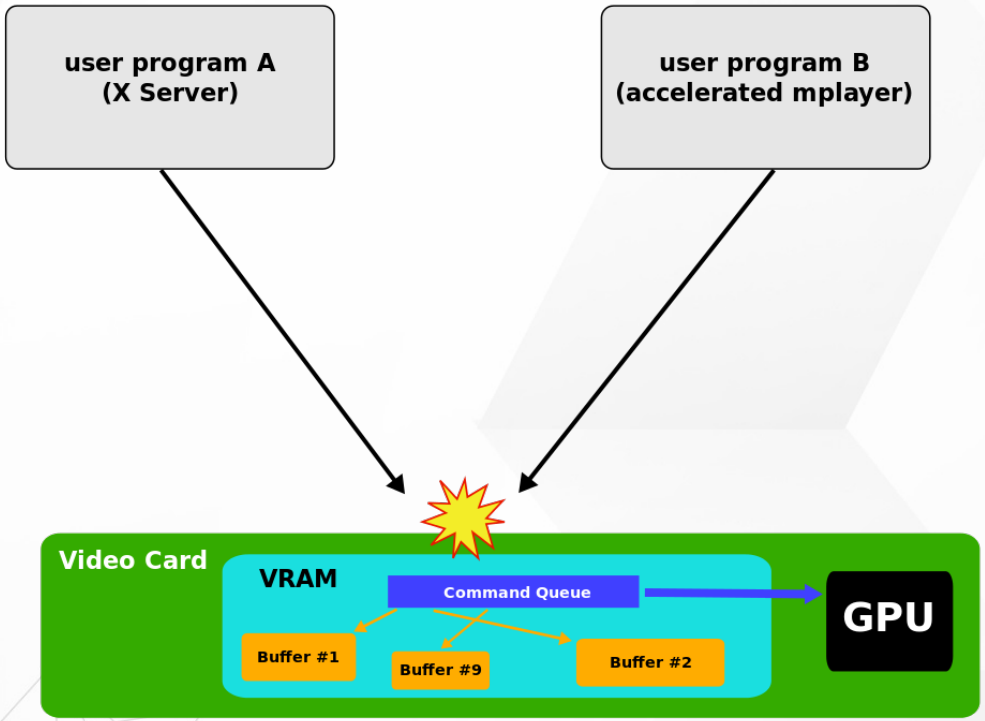
# Direct Rendering Manager (DRM)

- Direct Rendering Manager
  - Management of buffers and free space within that memory.
  - Solve Frame buffer driver cannot be used GPU and multi-user process.

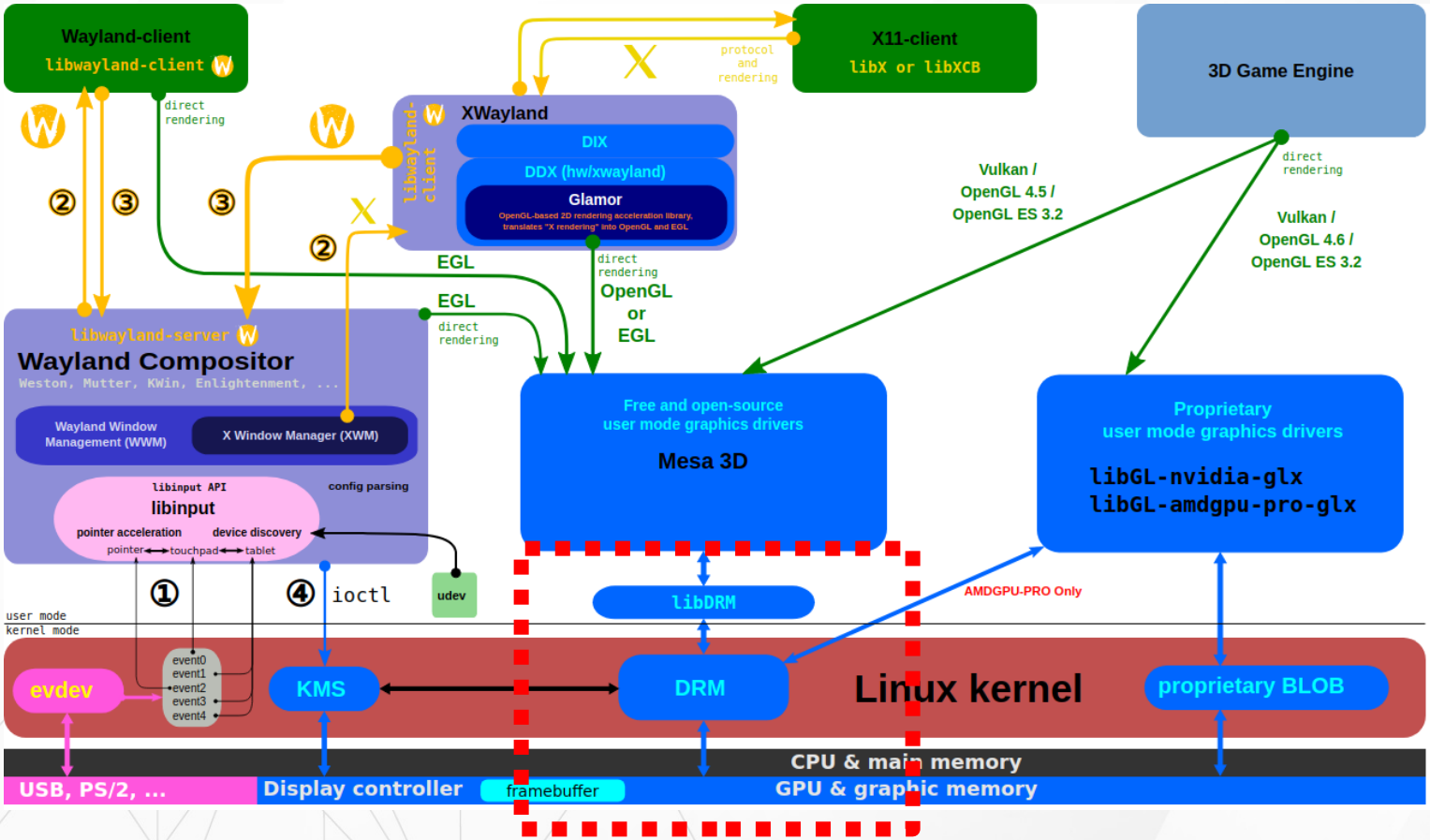
- DRM consists of
  - libdrm
    - libdrm provides a user space library for accessing the DRM
  - KMS : Kernel Mode Setting
    - Change resolution and depth
  - DRI : Direct Rendering Infrastructure
    - Interfaces to access hardware directly
  - GEM : Graphics Execution Manager
    - Buffer management
  - DRM Driver in kernel side
    - Access hardware

# Direct Rendering Manager (DRM)

If no use DRM



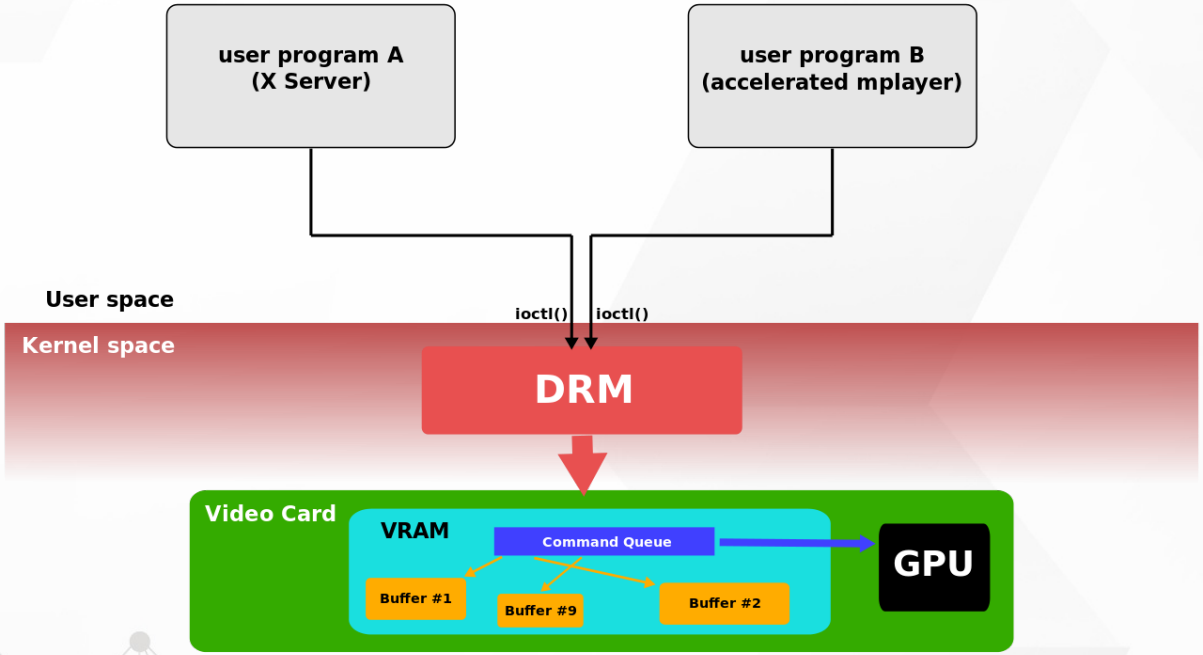
# Direct Rendering Manager (DRM)




https://upload.wikimedia.org/wikipedia/commons/2/2d/The_Linux_Graphics_Stack_and_glamor.svg

# Direct Rendering Manager (DRM)

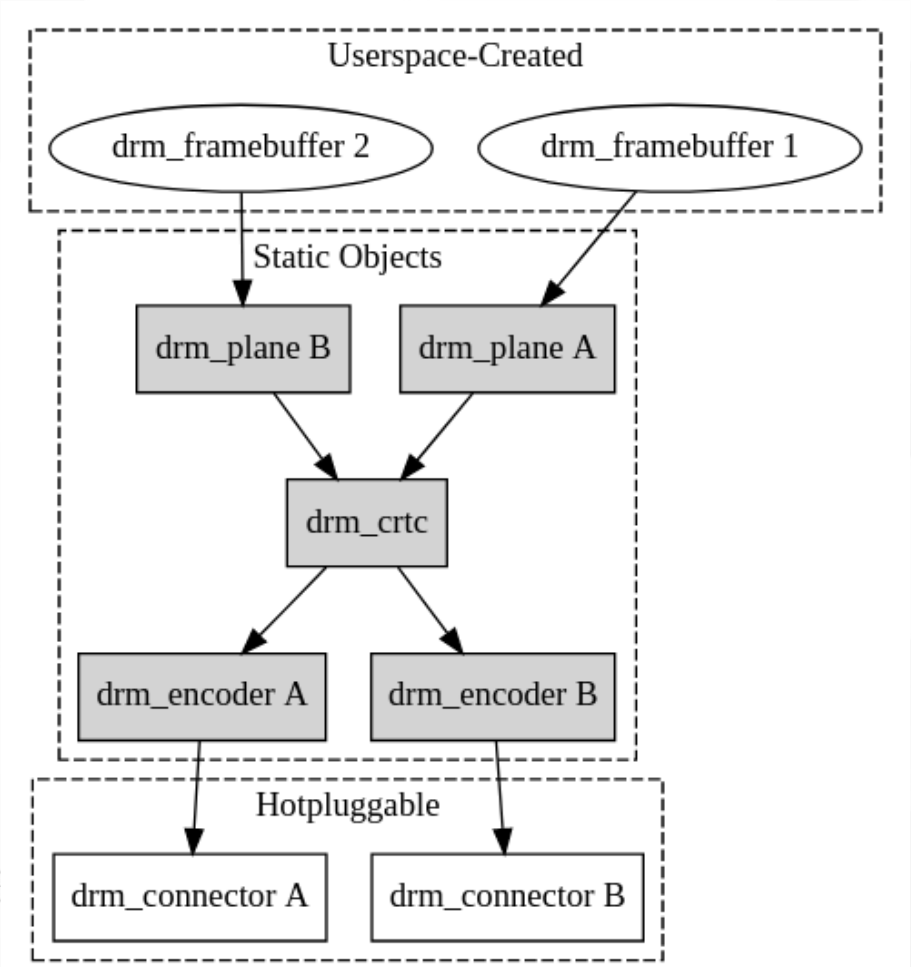
Use DRM



# Kernel Mode Setting (KMS)

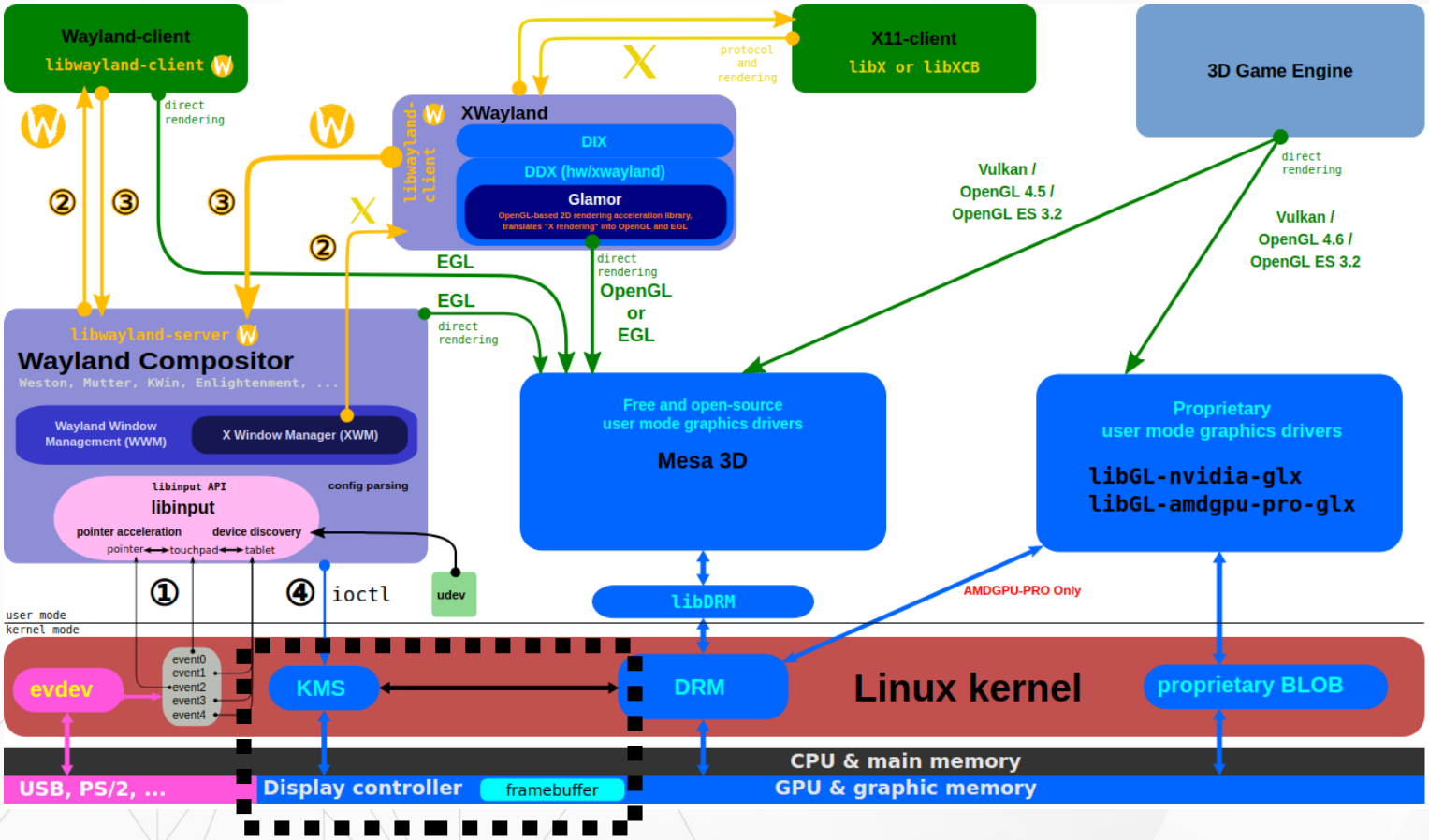
- KMS device model
  - CRTCs
  - Connectors
  - Encoders
  - Planes
- Kernel Mode Setting
  - screen resolution
  - color depth and
  - refresh rate

# Kernel Mode Setting (KMS)



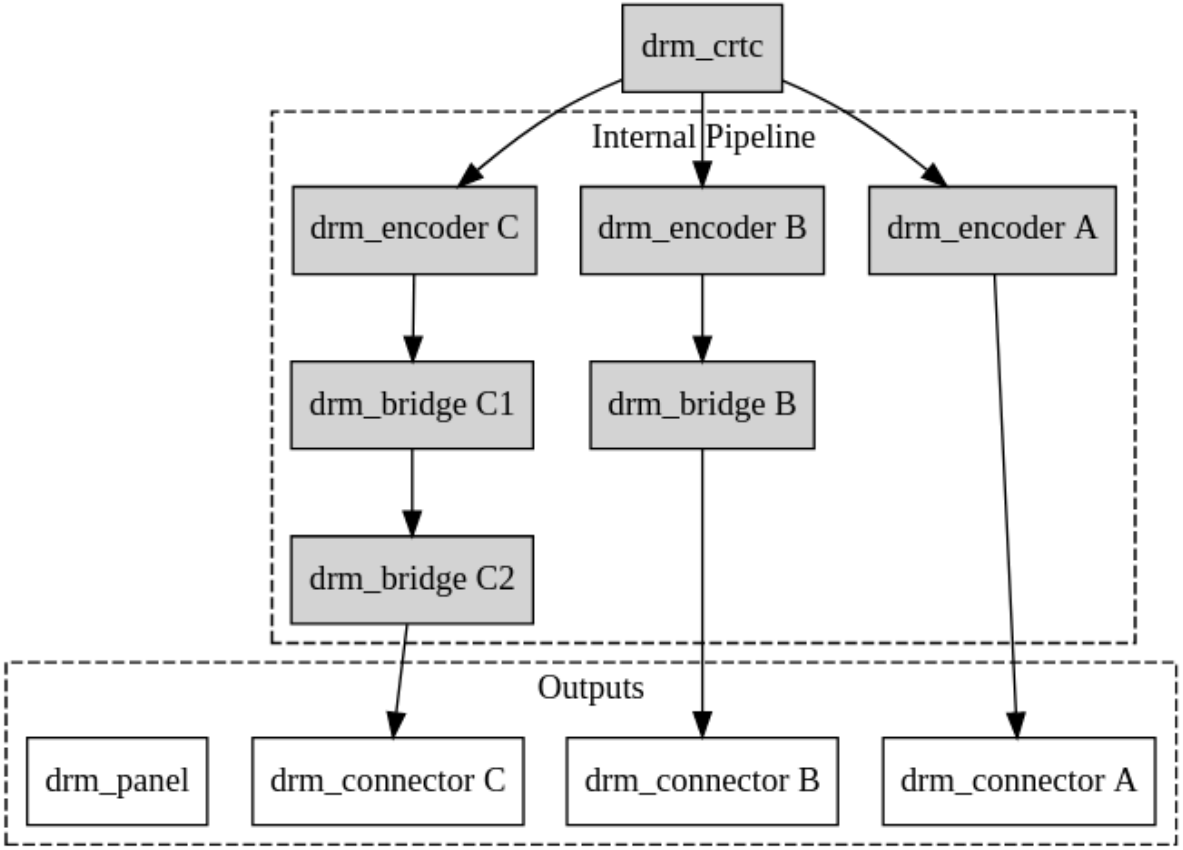

https://www.kernel.org/doc/html/v4.15/gpu/drm-kms.html

# Kernel Mode Setting (KMS)




https://upload.wikimedia.org/wikipedia/commons/2/2d/The_Linux_Graphics_Stack_and_glamor.svg

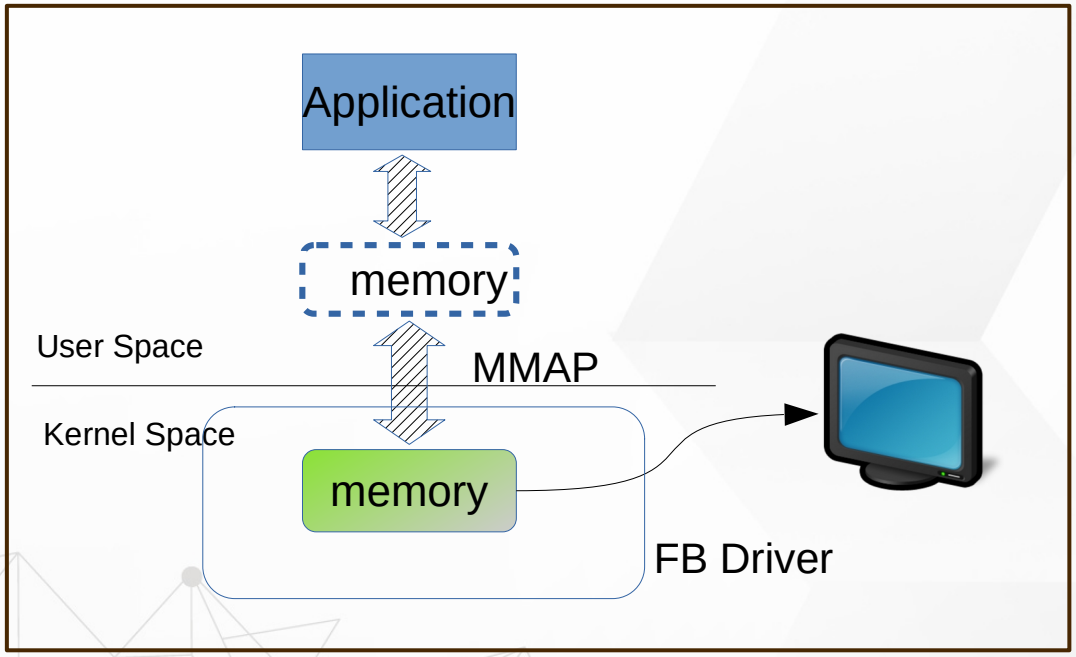
# Kernel Mode Setting (KMS)




https://www.kernel.org/doc/html/v4.15/gpu/drm-kms.html

# Video Frame Buffer

- The frame buffer device provides an abstraction for the graphics hardware.



# Kernel Mode Setting (KMS)