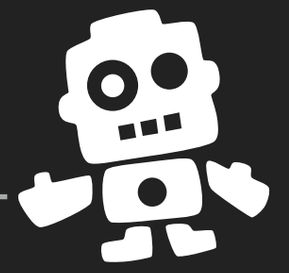
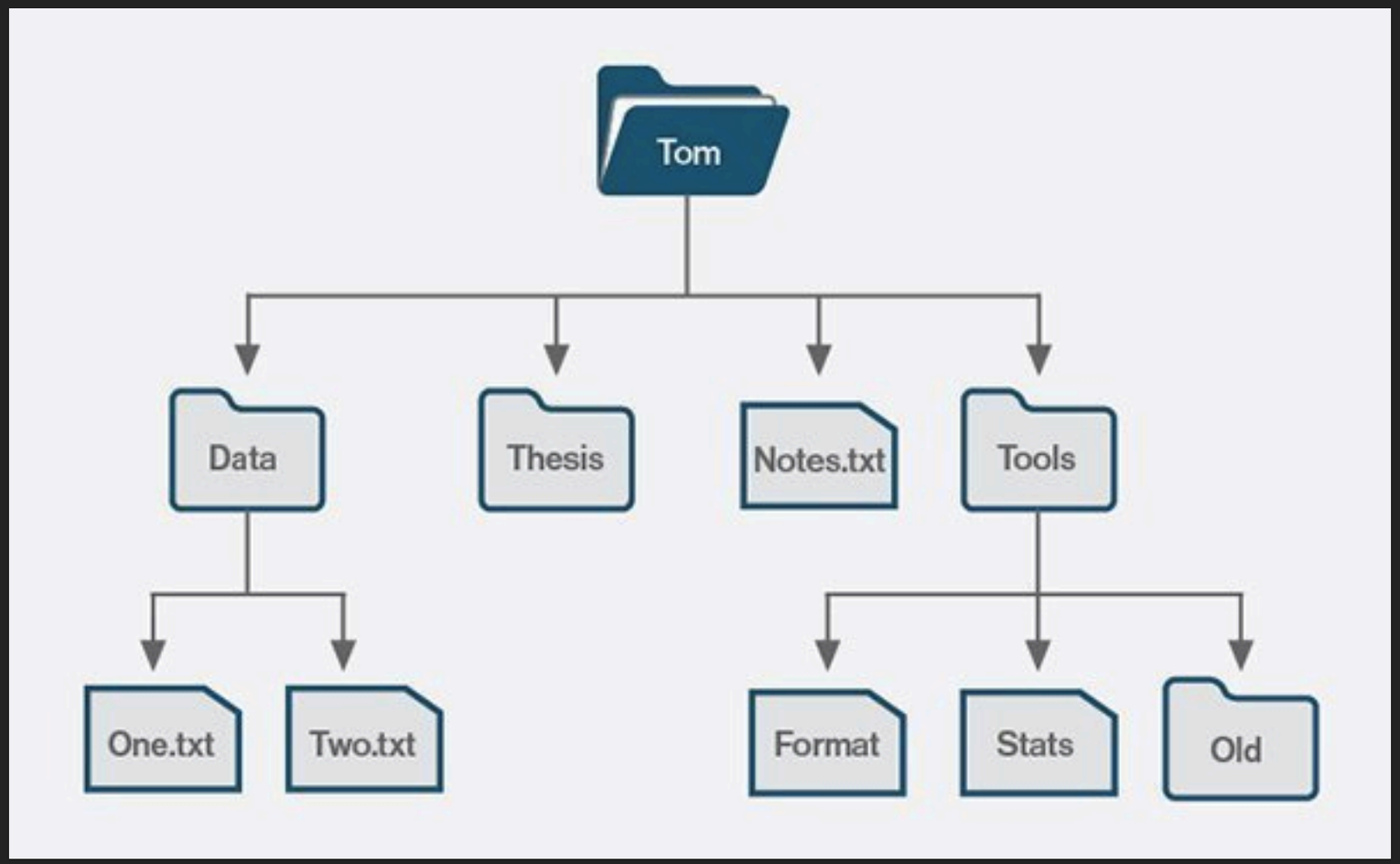
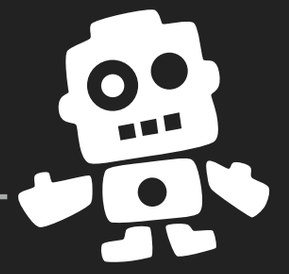


FILES AND FILE SYSTEMS

- In computing, a file system controls how data are stored and retrieved.





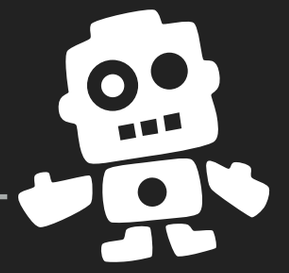
FILES AND FILE SYSTEMS

- Why do we need a file system?

  Not needed for data types like **`int`**, **`char`**, **`struct`**, and even **`class`**, but required for more complex data storage

  Without a file system it's difficult to tell where one piece of information stops and the next begins

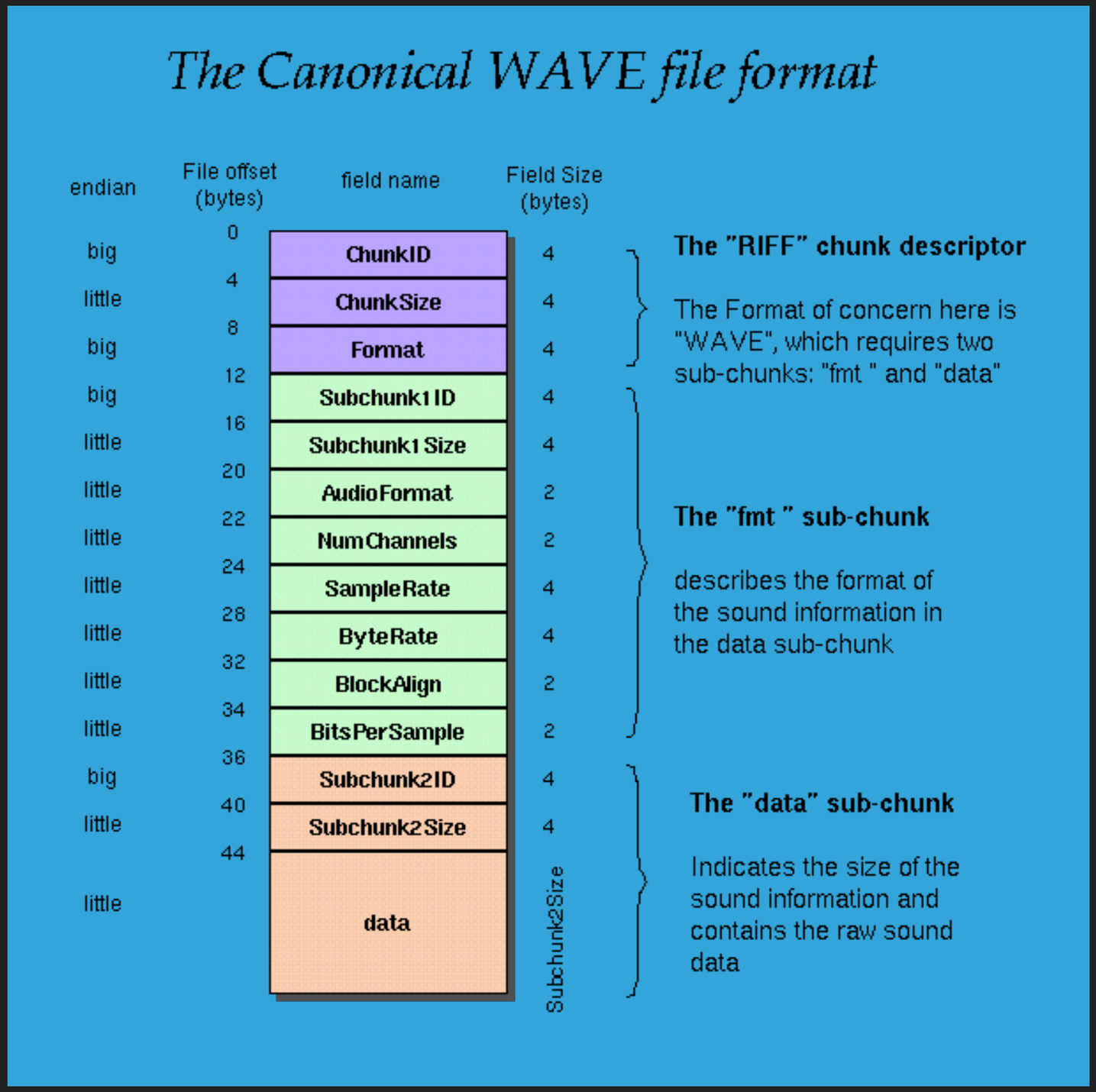
  Especially useful when using mass storage devices

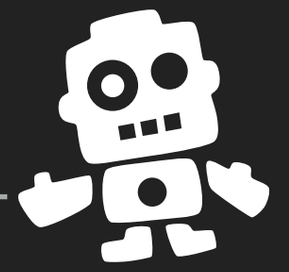


FILES AND FILE SYSTEMS

- What is a file?

A named collection containing data and metadata like *creation time*, *length*, and any number of other *attributes*





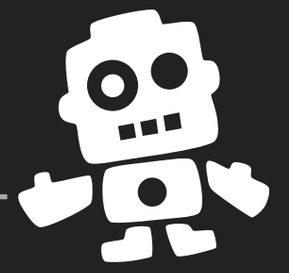
FILES AND FILE SYSTEMS

- What File System does the ESP8266 use?

  Computers (disks) use File Systems such as NTFS, EXT3, APFS, etc.

  Given their limited resources, microcontrollers and SoCs use especial File Systems such as SPIFFS

```
|--------------|-------|----------------|--|--|--|--|--|
^              ^       ^                ^     ^
SKETCH    OTA UPDATE   FILE SYSTEM   EEPROM  WIFI CONFIG (SDK)
```



FILES AND FILE SYSTEMS

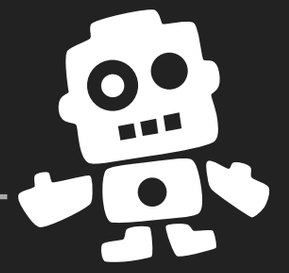
- Why use SPIFFS?

PROS:

Specifically designed for low ram usage

Built-in file system consistency checks

It can run on any NOR flash, not only SPI flash



FILES AND FILE SYSTEMS

- Why use SPIFFS?

CONS:

Does not support directories (uses a flat structure)

It is not a real-time stack. One write operation might last much longer than another.

Intended for small memory devices (< 128MB)