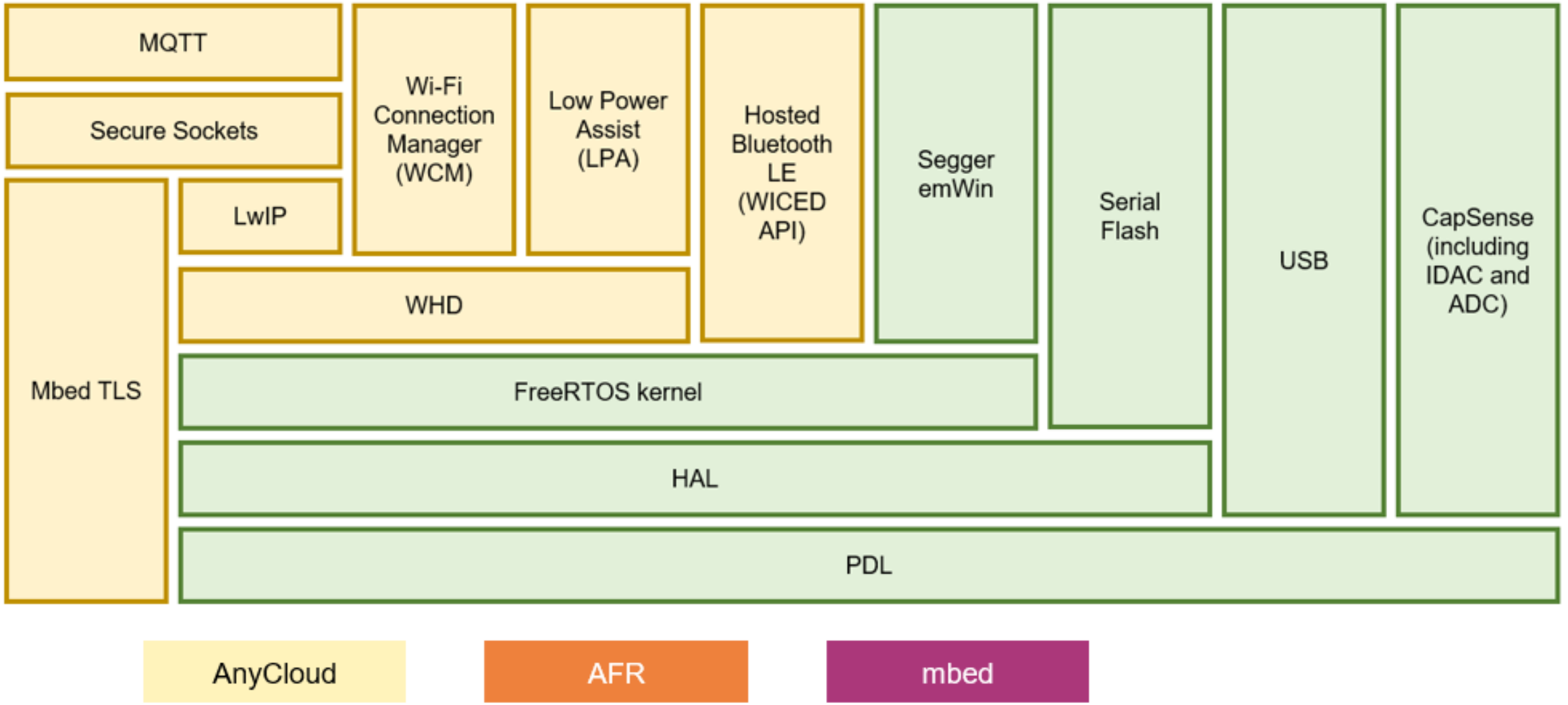
AnyCloud WiFi Design Flow

Aditi Bhatnagar
Murali Ramu
25-June-2020

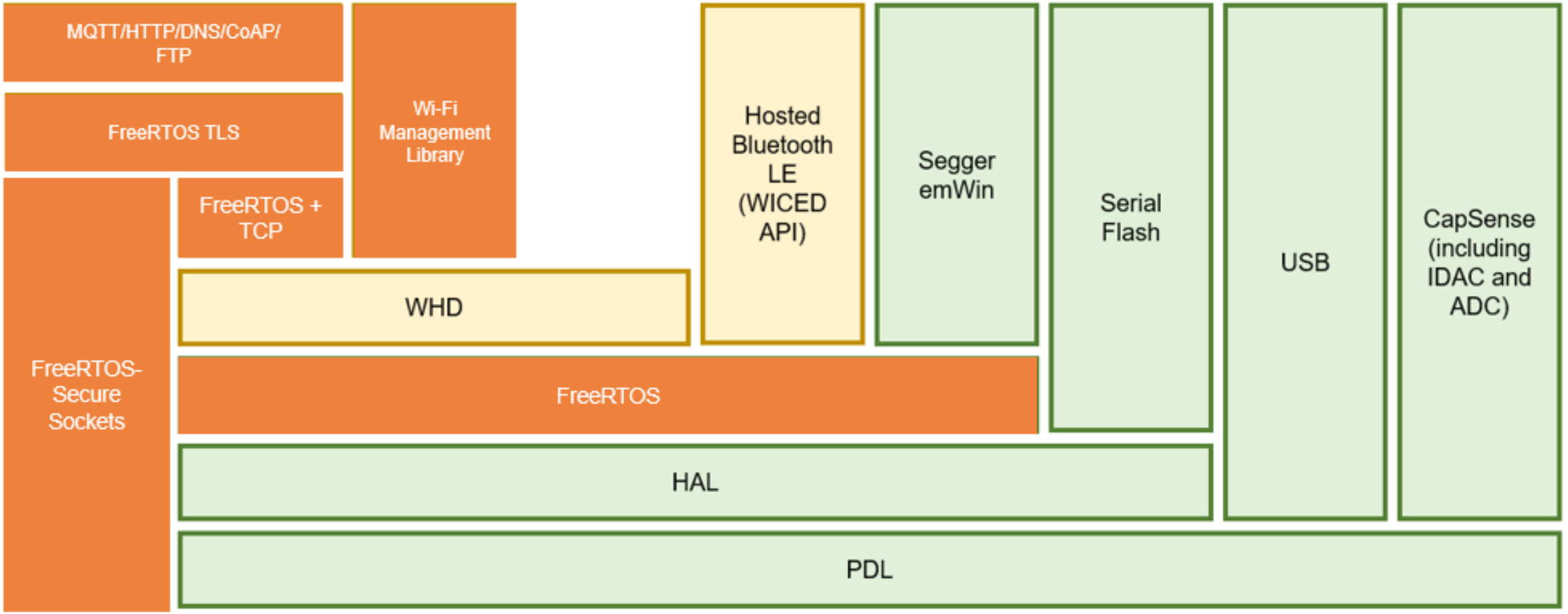
Agenda

- Software Architecture – AnyCloud / AFR / Mbed
- AnyCloud Stack
- AnyCloud WiFi Libraries
  - Using Library manager to add Wi-Fi to a PSoC 6 project in AnyCloud
  - Middleware Core Libraries
  - Wi-Fi Connection Manager (WCM)
  - Secure Sockets
- WICED Wi-Fi Driver
- WHD WiFi Host Driver
- Demo – TCP Client
- Q&A – Feel free to ask questions in the chat window even during the presentation.

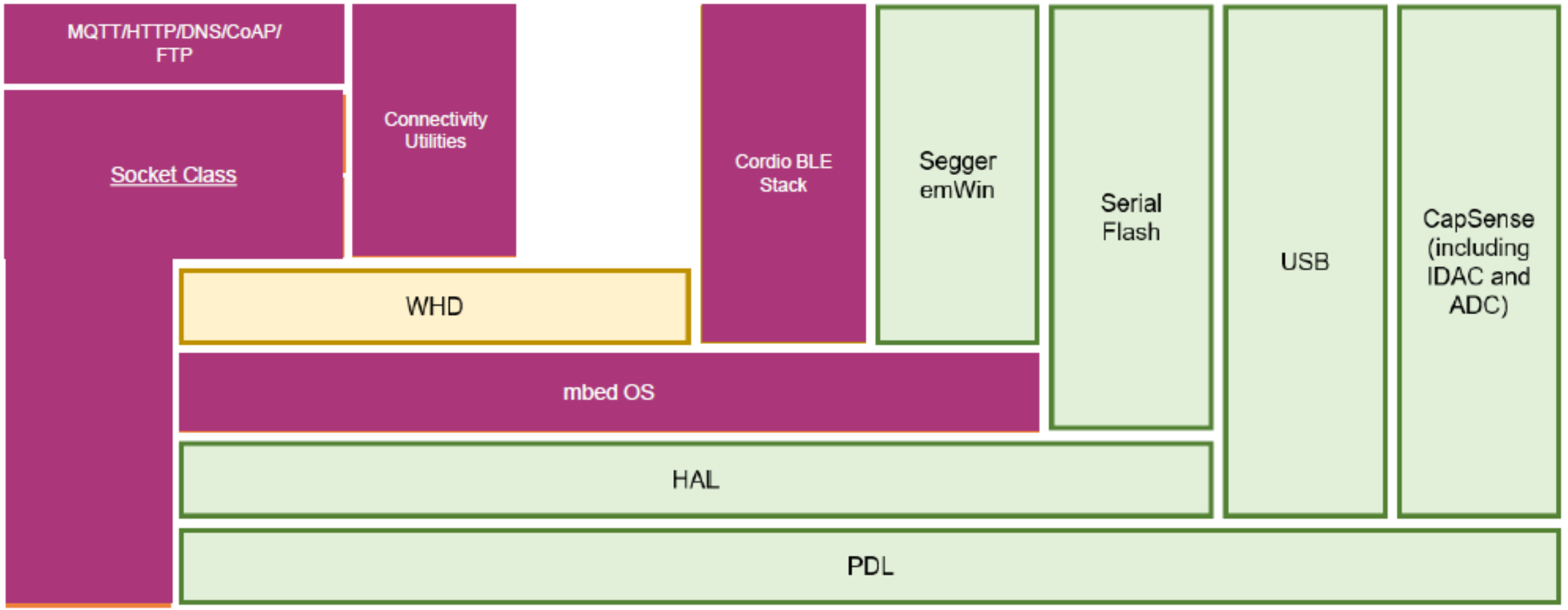
Software Architecture – AnyCloud / AFR / Mbed



Software Architecture – AnyCloud / AFR / Mbed

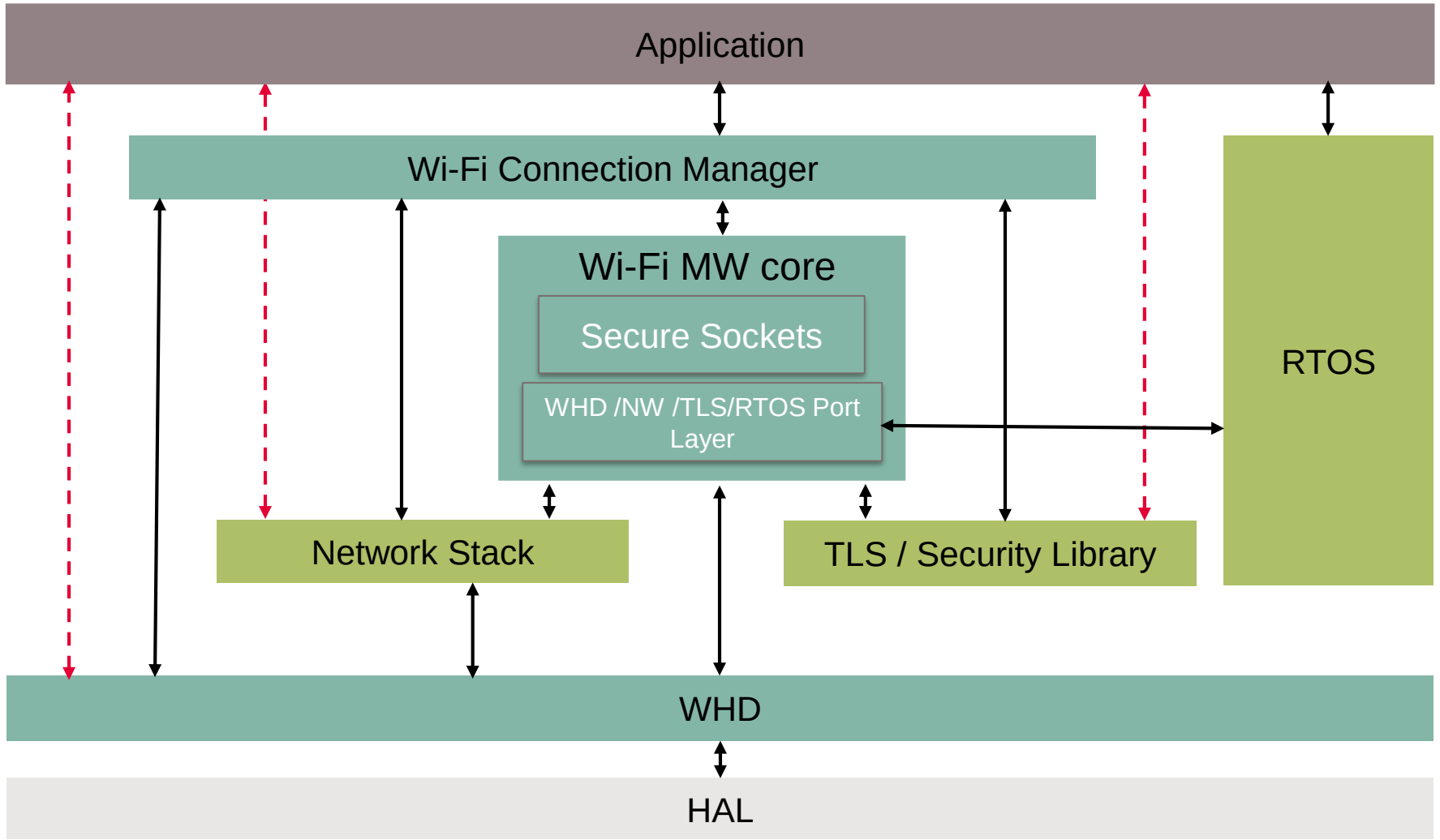


Software Architecture – AnyCloud / AFR / Mbed



AnyCloud Stack

- LwIP, mbedTLS & FreeRTOS are all pulled from their respective GIT Repos.
- RTOS, N/W Stack & Security (TLS) libraries can be substituted with alternatives in any combination.



Legend

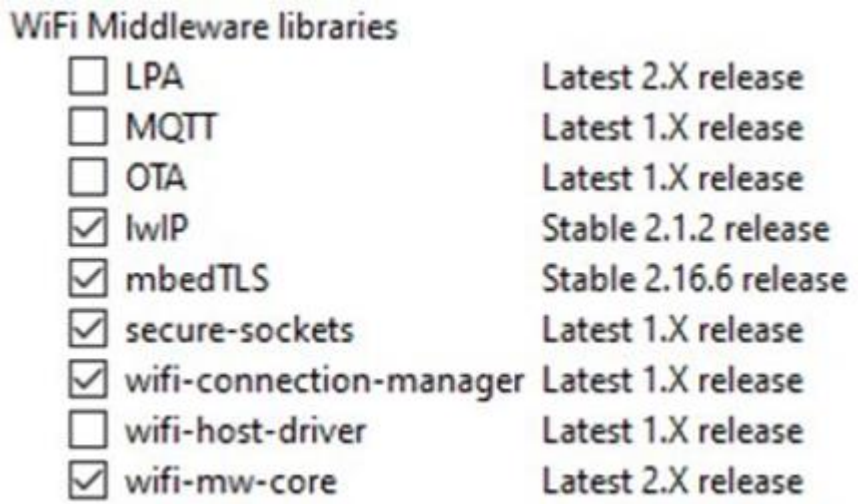
3rd Party Libraries

Infineon - AnyCloud Lib & Wi-Fi Driver

Host MCU HAL

WiFi Middleware Libraries

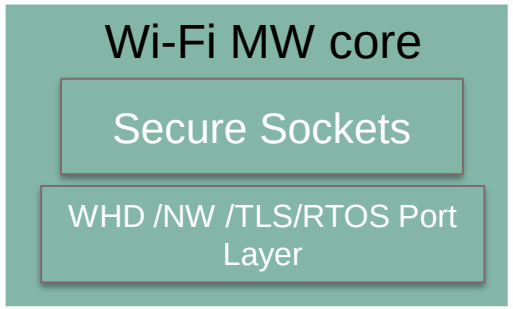
- Adding WiFi to an existing PSoC 6 project
  - Ensure the project is using FreeRTOS.
  - Add WCM from the library manager.
  - This will add / download all necessary libraries.
  - Make changes to the Makefile as below
  - Code should now compile



- Makefile Changes
  - `COMPONENTS=FREERTOS LWIP MBEDTLS`
  - `MBEDTLSFLAGS = MBEDTLS_USER_CONFIG_FILE='"mbedtls_user_config.h"'`
  - `DEFINES=$(MBEDTLSFLAGS) CYBSP_WIFI_CAPABLE CY_RTOS_AWARE`

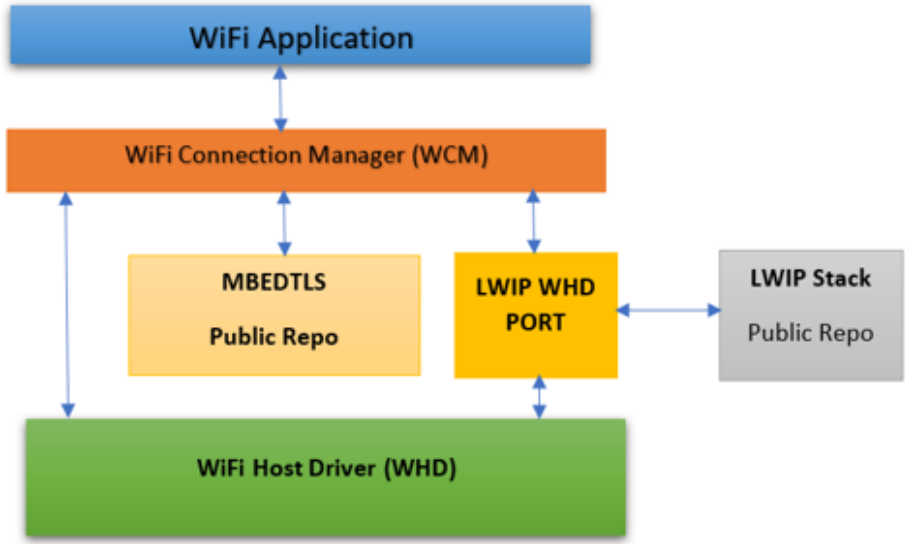
Wi-Fi Middleware Core Library

- Bundles all the core components required for Wi-Fi development.
- Adds LwIP, mbedTLS, FreeRTOS, WHD, and Secure Sockets as dependencies (.lib files)
- Adds the glue between LwIP and WHD



WiFi Connection Manager

- Provides API for Wi-Fi scan and connection management
- Application can register for disconnection and reconnection notifications
- Re-authenticates the connection with the AP on intermittent connection loss
- Implements Wi-Fi Protected Setup (WPS) Enrollee role
- Reduces code and no. of API calls in case of App level protocol. (For ex: MQTT).



- E.g.

```
cy_rslt_t cy_wcm_init(cy_wcm_config_t *config)
cy_rslt_t cy_wcm_register_event_callback(cy_wcm_event_callback_t event_callback)
```

Secure Sockets

- Abstraction API for the underlying network (LwIP) and security (mbedTLS) stacks
- Socket like API for secure (TLS) and non-secure socket communications
- Supports both client and server modes
- Ex: TLS handshake/secure connection.

mbedTLS APIs for the TLS handshake

```
mbedtls_platform_set_time
mbedtls_ssl_init
mbedtls_ssl_config_init
mbedtls_x509_crt_init
mbedtls_ctr_drbg_init
mbedtls_entropy_init
mbedtls_ctr_drbg_seed
mbedtls_ssl_config_defaults
mbedtls_ssl_conf_authmode
mbedtls_x509_crt_parse
mbedtls_ssl_conf_ca_chain
mbedtls_ssl_conf_rng
mbedtls_ssl_setup
mbedtls_ssl_set_hostname
mbedtls_ssl_set_bio
mbedtls_ssl_handshake
mbedtls_ssl_get_verify_result
```

Secure Sockets Library

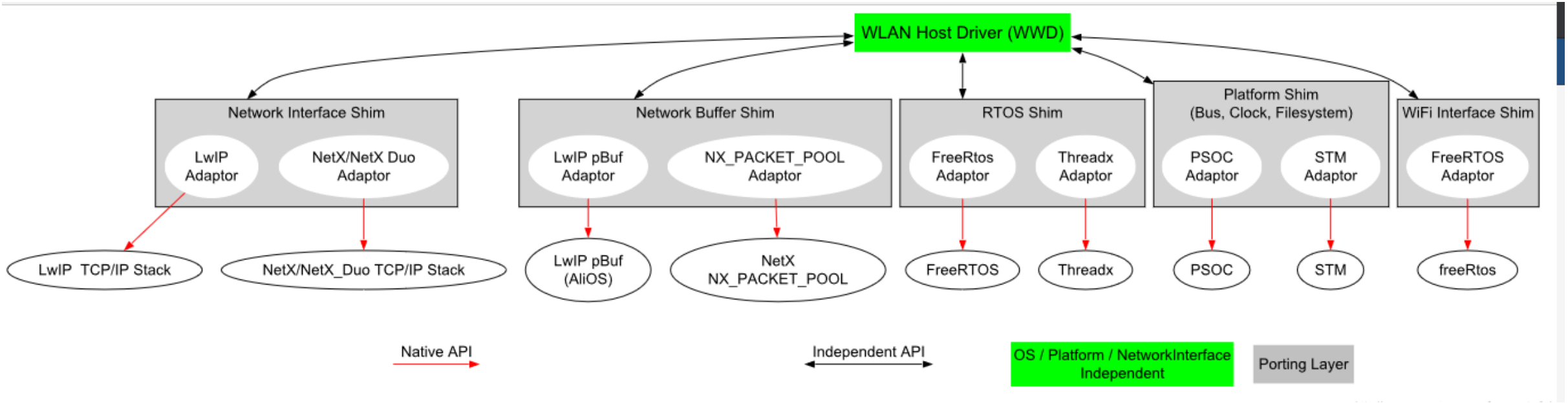
```
/**
 * Performs a TLS handshake and connects to the server.
 *
 * @param[in]  context Context handle for the TLS Layer created using \ref cy_tls_create_context.
 * @param[in] endpoint Endpoint type for the TLS handshake.
 * @return CY_RSLT_SUCCESS on success; an error code on failure.
 *         Important error code related to this API function is: \n
 *         CY_RSLT_MODULE_TLS_ERROR
 */
cy_rslt_t cy_tls_connect( cy_tls_context_t context, cy_tls_endpoint_type_t endpoint );
```

WiFi Driver

WiFi Driver

- Hardware needs underlying software which help them to interact known as device drivers
- Can be an audio driver, keyboard, Bluetooth, WiFi …
- Software allowing the CPU to interact with the WiFi/Network Card
- More specifically, software between the host and the WLAN chip
- In case of Cypress/Infineon → WWD, WHD

WICED WiFi Driver



- WICED WiFi Driver
- Intertwined with WICED
- Difficult to port
- WHD is used

WiFi Host Driver

- Used to interact with CY WLAN chips
- Modular with AnyCloud
- Portable

WHD Folder Structure

WHD Design

FW Download and WLAN Chip Logging Module

- Uses the HAL Resource API
- Doesn't use the services of control or data path module. It directly access the "WHD Bus Interface" to write the resources
- Once download is complete, allows the Chip to run
- Responsible for collecting WLAN chip logs

Control Module

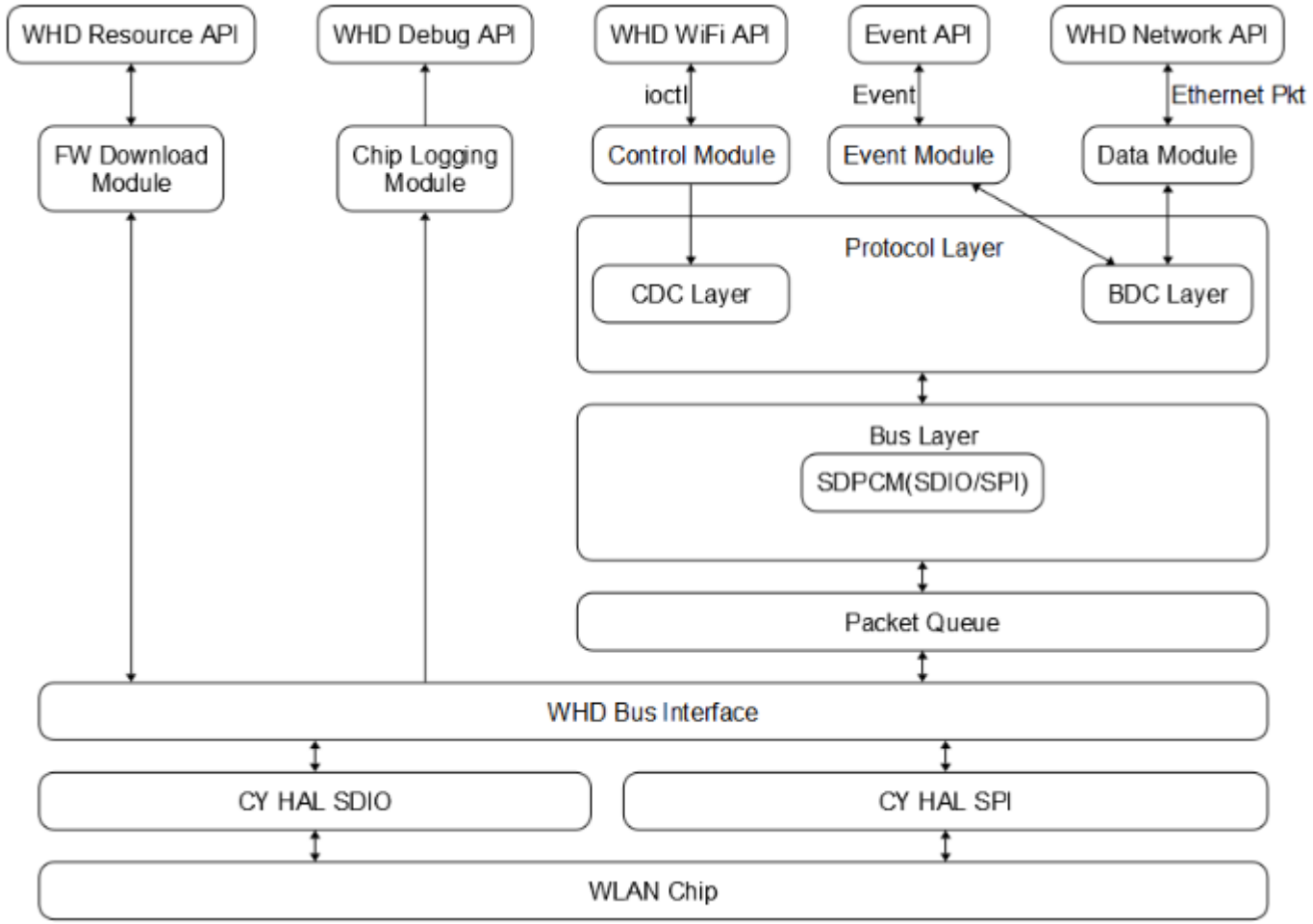
- Control access to WLAN Chip is done using IOCTLs

Data Module

- Responsible for handling User data received/sent in TCP/IP interface

Event Module

- Responsible for events generated from WLAN chip



WHD Design

Protocol layer

› Protocol layer is bus independent and is required for either SDIO/SPI.

› Packets sent to WLAN chip needs this header

CDC layer

› Control module sends messages

› Adds 16 byte header

CDC Header 16 bytes	IOCTL Message Variable size..

```
typedef struct
{
uint32_t cmd;    /* ioctl command value */
uint32_t len;/* lower 16: output buflen; upper 16: input buflen (excludes header)*/
uint32_t flags;  /* flag defns given in bcmcdc.h */
uint32_t status; /* status code returned from the device */
} cdc_header_t;
```

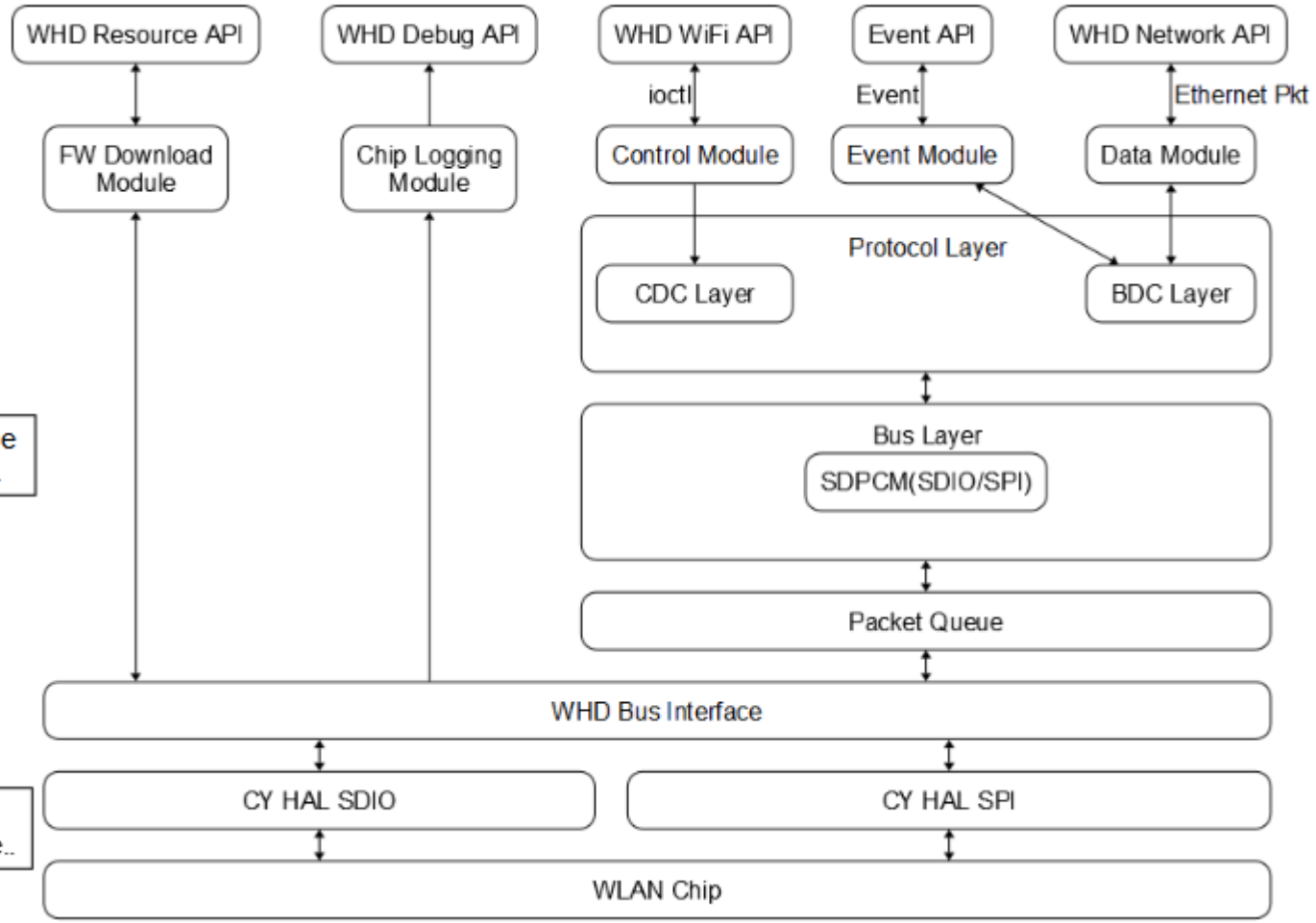
BDC layer

› Data module sends messages

› Adds a 4 byte header

BDC Header 4 bytes	User Data Variable size..

```
typedef struct
{
uint8_t flags;      /* Flags */
uint8_t priority;   /* 802.1d Priority (low 3 bits) */
uint8_t flags2;
uint8_t data_offset; /* Offset from end of BDC header to packet data, in 4-uint8_t
words.  Leaves room for optional headers.*/
} bdc_header_t;
```



WHD Design

BUS layer

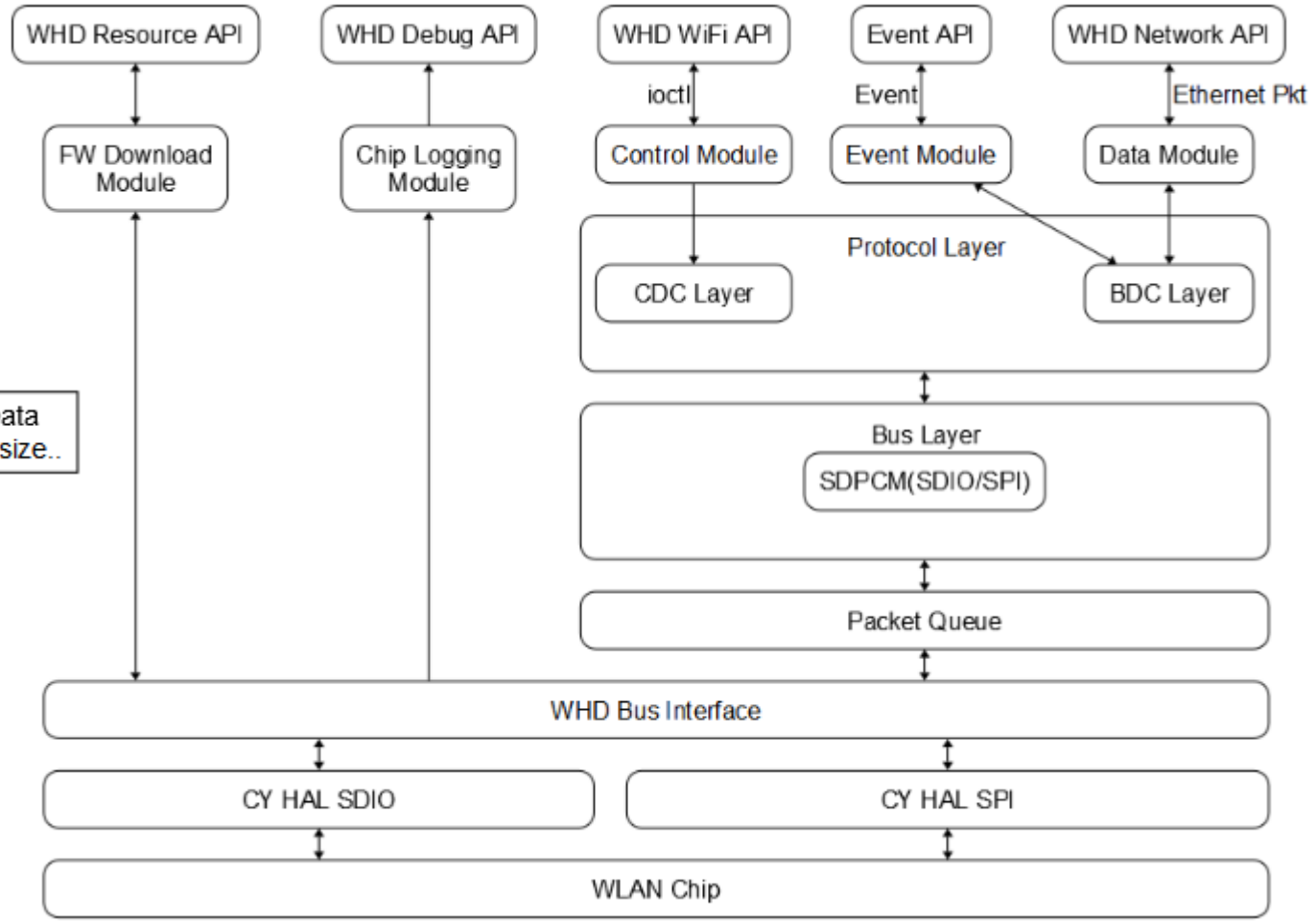
- Responsible for handling bus level protocol handling
- For SDIO, SDPCM is used

SDPCM - SDIO/SPI Bus Layer

- Adds sequence number to packet sent to WLAN Chip
- Flow control between WHD and WLAN Chip
- Adds 10 byte header

SDPCM Header 10 byte	CDC/BDC Header	User Data Variable size..

```
typedef struct
{
uint16_t  frametag[2]; /* SDPCM packet size */
uint8_t sequence; /* Sequence number of pkt */
uint8_t channel_and_flags; /* IOCTL/IOVAR or User Data or Event */
uint8_t next_length;
uint8_t header_length; /* Offset to BDC or CDC header */
uint8_t wireless_flow_control;
uint8_t bus_data_credit; /* Credit from WLAN Chip */
uint8_t _reserved[2];
} sdpcm_header_t;
```



WHD Design

Packet Queue

- Control/User data is queued in a link list in this layer
- Once Credit is available, sends data to WLAN chip

WHD Bus Interface

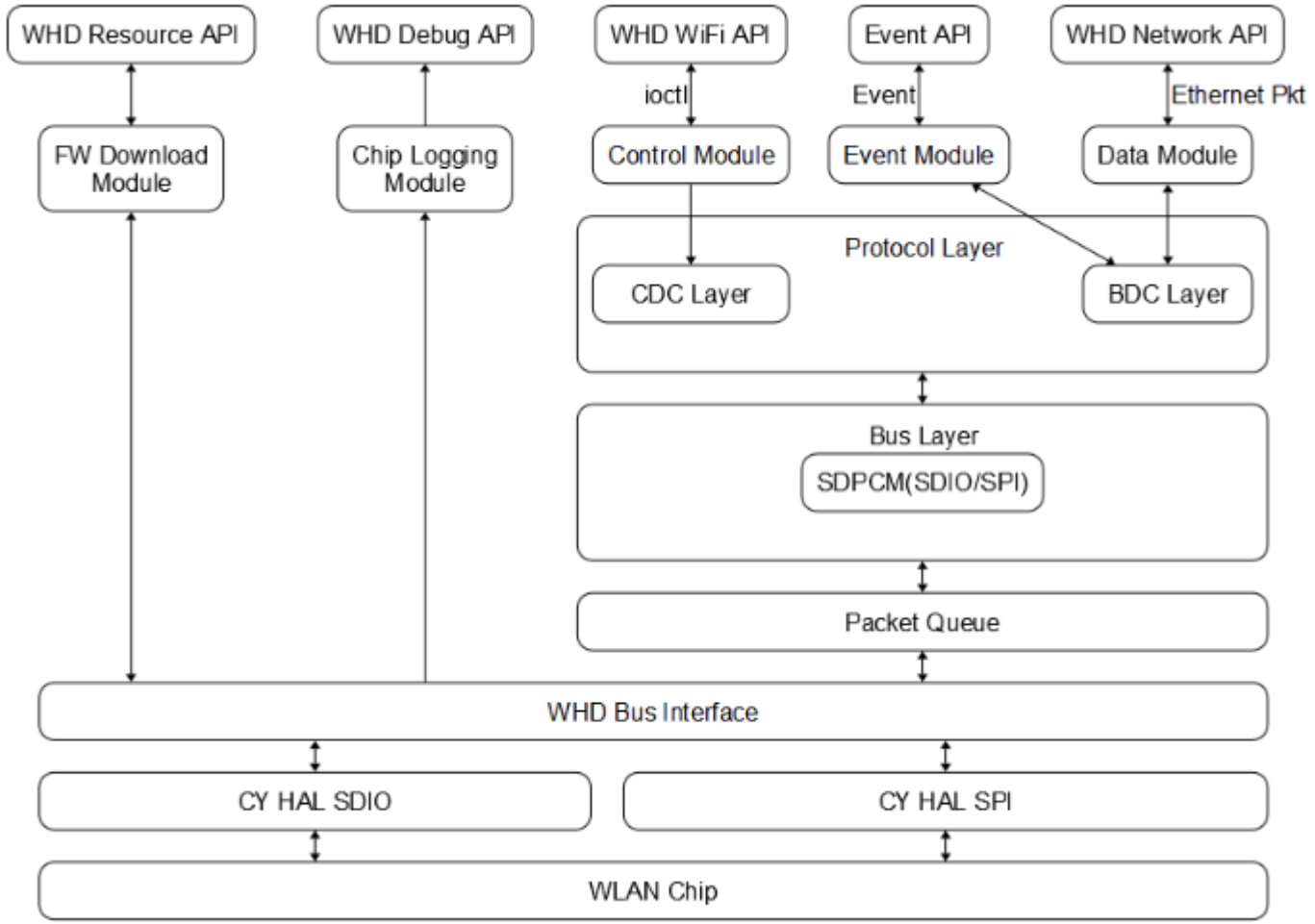
- Gives bus independent access functions to packet engine/sdpcm layer
- Primarily used to keep the access functions common between SDIO/SPI

SDIO HAL interface

- CY HAL interface to access the SDIO Host controller Hardware.
- External to WHD driver

SPI HAL Interface

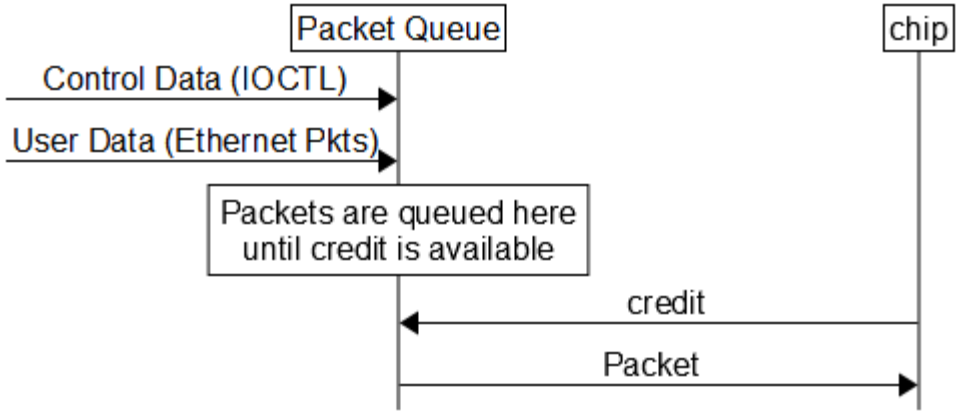
- CY HAL interface to access the SPI Host Controller Hardware.
- External to WHD driver

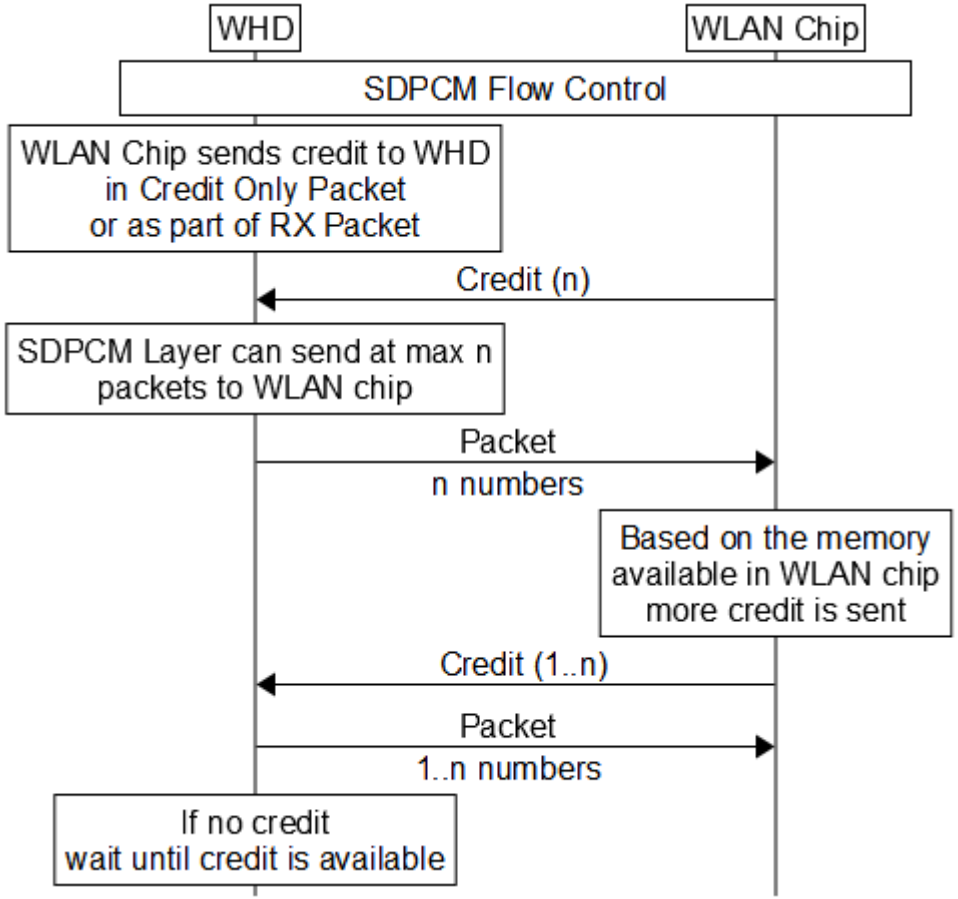


Flow Control Using SDPCM/ Packet Queue

Packet Queue

- Control/User data is queued in a link list in this layer
- Once Credit is available, sends data to WLAN chip





WHD Port

CY RTOS API

› Provides prototypes for functions that allow the WHD to use RTOS functionality

CY HAL Resource API

› Wi-Fi firmware, NVRAM, and CLM BLOB information are treated as resources to be downloaded onto the Wi-Fi chip

Buffer Interface API

› WHD requires packet buffers to exchange information between the host and Wi-Fi firmware

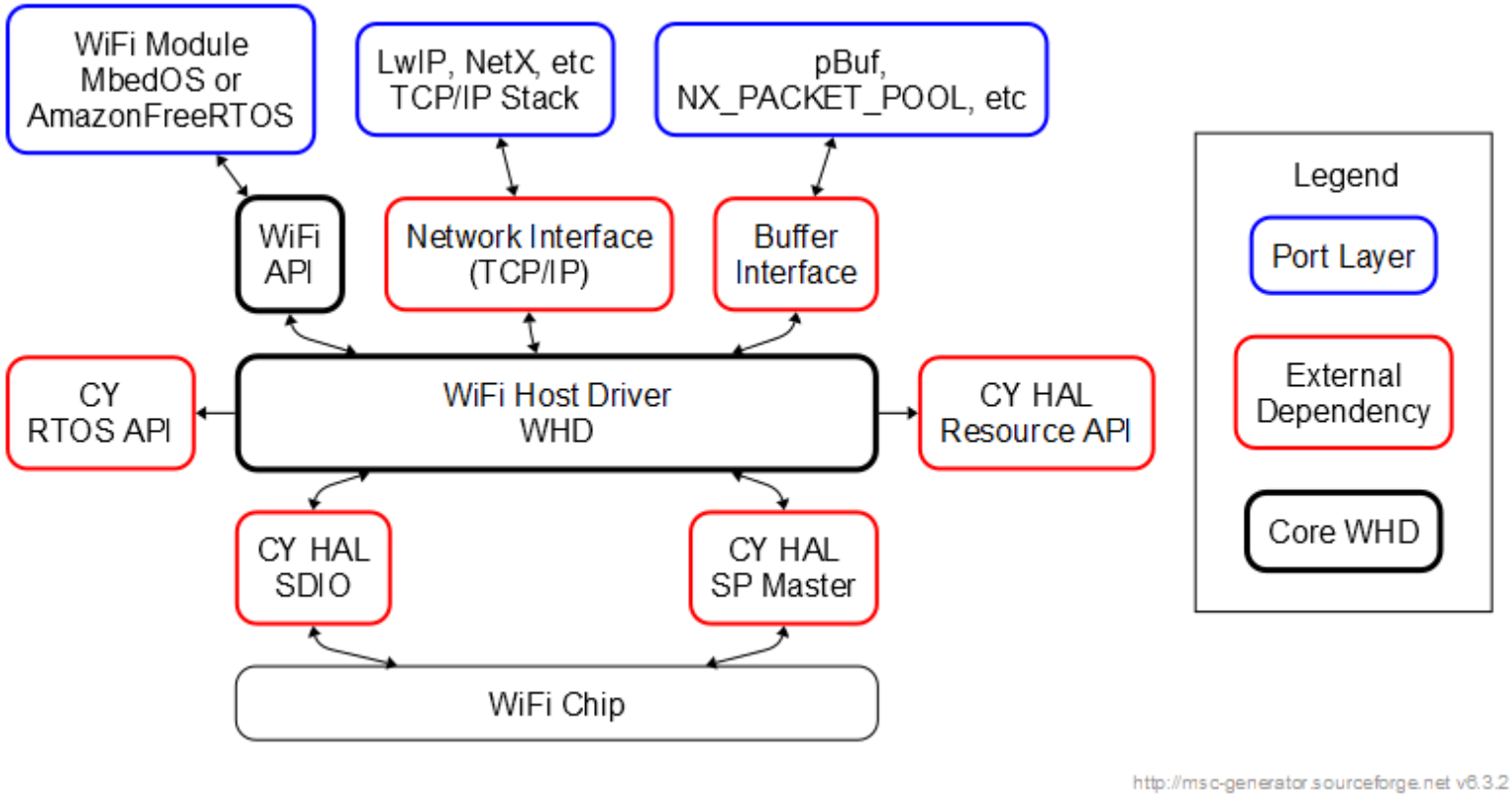
Network Interface API

› WHD calls this function pointer to pass the received TCP/IP data packet from WLAN

CY HAL SPI/SDIO Bus API

› WHD uses the following functions to access the host bus controller for SDIO or SPI buses

› Replace and use these functions appropriately to ensure bus operations



A world leader in semiconductor solutions

Our vision

We are the link between the real and the digital world.

Our values

We commit
We partner
We innovate
We perform

Our mission

We make life easier, safer and greener.

Part of your life. Part of tomorrow.

Abbreviations

- AFR – Amazon FreeRTOS
- PSoC – Programmable system on chip
- WCM – Wi-Fi Connection Manager
- WWD – Wiced Wi-Fi Driver
- WHD – Wi-Fi Host Driver
- TCP – Transmission Control Protocol
- TLS – Transport Layer Security
- HAL – Hardware Abstraction Layer
- LwIP – Lightweight Internet Protocol
- LPA – Low Power Assistant
- OTA – Over the Air (Programming)
- MQTT – Message Queuing Telemetry Transport
- WPS – Wi-Fi Protected Setup
- HTTP – Hyper Text Transfer Protocol
- DNS - Domain Name System
- CoAP – Constrained Application Protocol
- FTP – File Transfer Protocol

References

- https://os.mbed.com/docs/mbed-os/v6.0/apis/socket.html
- https://docs.aws.amazon.com/freertos/latest/userguide/secure-sockets.html
- https://docs.aws.amazon.com/freertos/latest/userguide/freertos-wifi.html
- https://github.com/cypresssemiconductorco/secure-sockets
- https://github.com/cypresssemiconductorco/wifi-connection-manager
- https://github.com/cypresssemiconductorco/connectivity-utilities
- https://github.com/cypresssemiconductorco/wifi-host-driver
- https://community.cypress.com/community/software-forums/anycloud

AFR, Mbed and Modus

TCP/IP Model	AFR	Mbed	Any
Application	HTTP/DNS/MQTT/CoAP/FTP....		
Transport	FreeRTOS: Secure Sockets	Socket Class	Secure Sockets Library
Network	FreeRTOS: Wi-Fi Management Library	Connectivity Utilities	Wifi-Connection Manager Library
Physical	WWD/WHD	WHD	WHD