



Implementing A Bluetooth Stack on UEFI

Tony C.S. Lo
Senior Manager
American Megatrends Inc.



Agenda



Introduction

Bluetooth Architecture

UEFI Bluetooth Stack

Summary

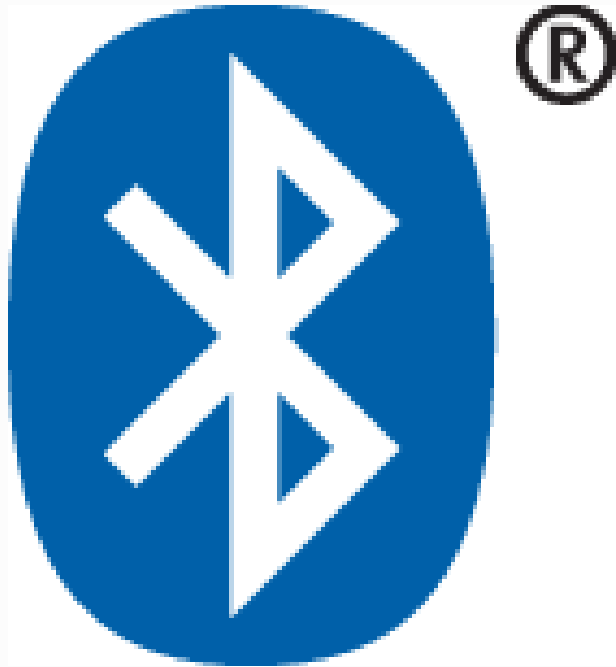







Introduction



Introduction



→ Bluetooth Key Features:

- ▶ Robustness 
- ▶ Low Power Consumption 
- ▶ Low Cost 

→ Two System Forms:

- ▶ **Basic Rate (BR) / Enhanced Data Rate (EDR)**
- ▶ Low Energy (LE)

BR/EDR Radio Specification



Frequency

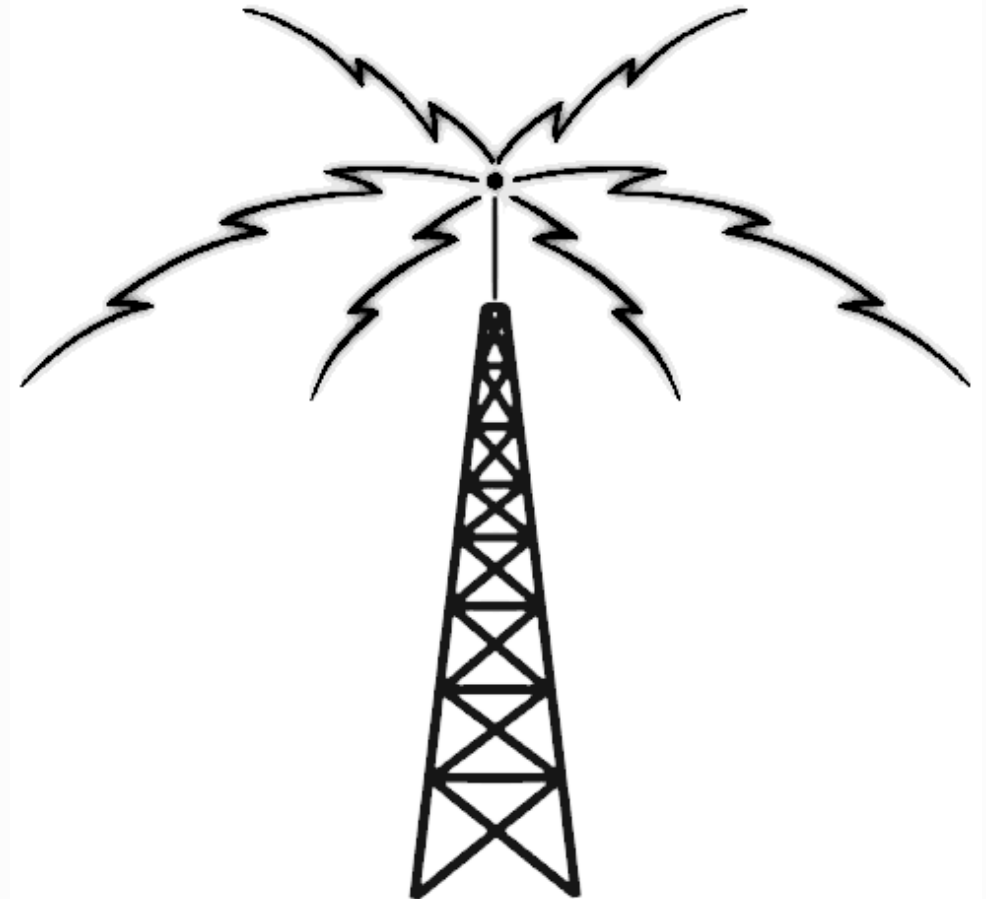
- Unlicensed 2.4Ghz ISM

Spread Spectrum

- Frequency Hopping Spread Spectrum (FHSS)

Data Rates

- BR 721.2 Kbps
- EDR 2.1Mbps
- 802.11 Alternative Mac/PHY (AMP)
54Mbps



Piconet Topology

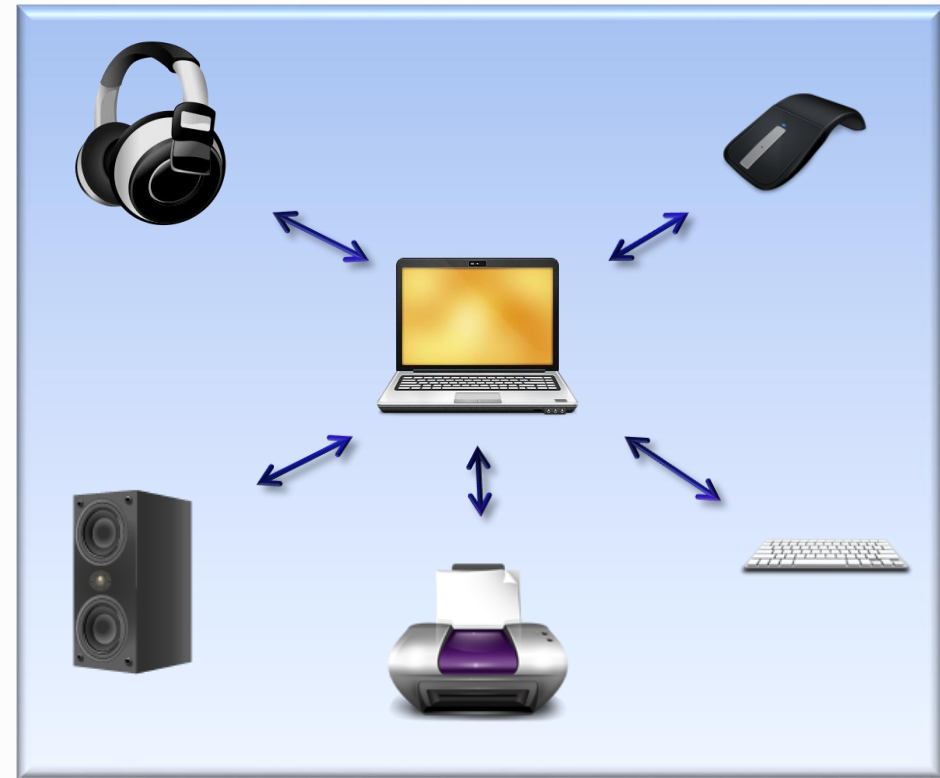


Master

- Provide the reference clock and frequency hopping pattern.
- Only one master within one piconet.

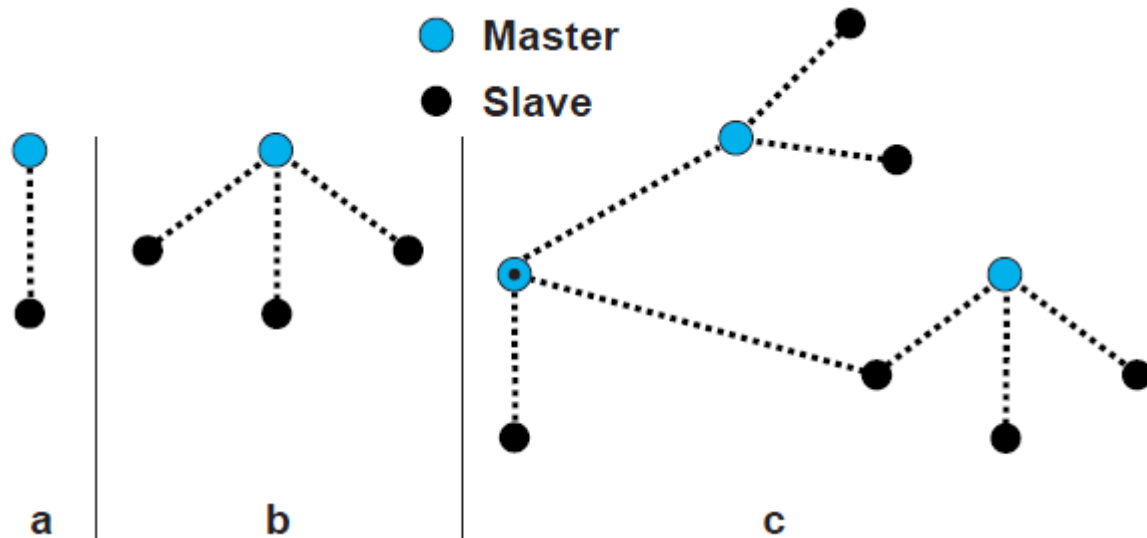
Slave

- Sync to master's clock and frequency hopping pattern.
- One to seven active slave devices are allowed in one piconet.
- 255 possible parked slave devices.



Single Master and 1-7 Slave Devices form a piconet

Scatternet



Multiple piconets that have common devices are called scatternet.

- Master device can act as slave in other piconets.
- Each Master owns one identical physical channel.

Net (a): Piconet

- Single master and single slave.

Net (b): Piconet

- Single master and multiple slave.

Net (c): Scatternet

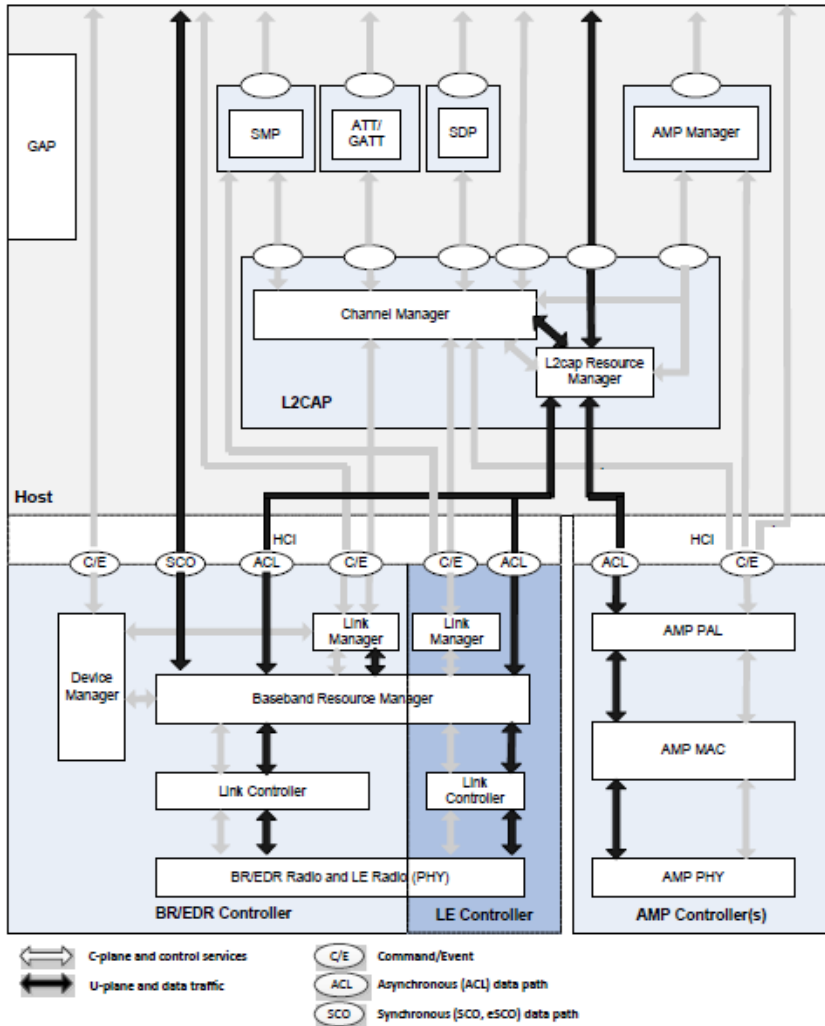
- Multiple piconets share devices



Bluetooth Architecture



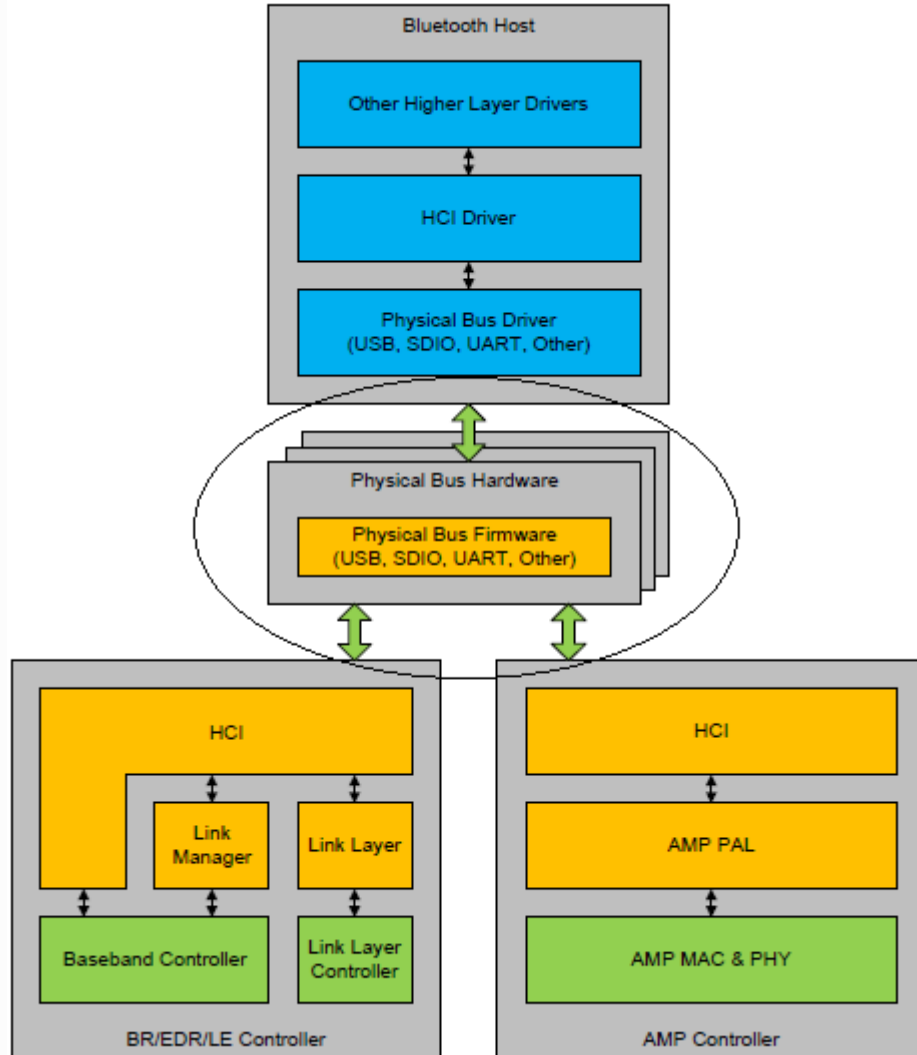
Bluetooth Core System Architecture



The Bluetooth Core system consists of a Host, a Primary Controller and zero or more Secondary Controllers.

The BR/EDR Core system includes support of Alternate MAC/PHYs (AMPs) including an AMP Manager Protocol and Protocol Adaptation Layers (PALs) supporting externally referenced MAC/PHYs.

Bluetooth Host Controller Interface



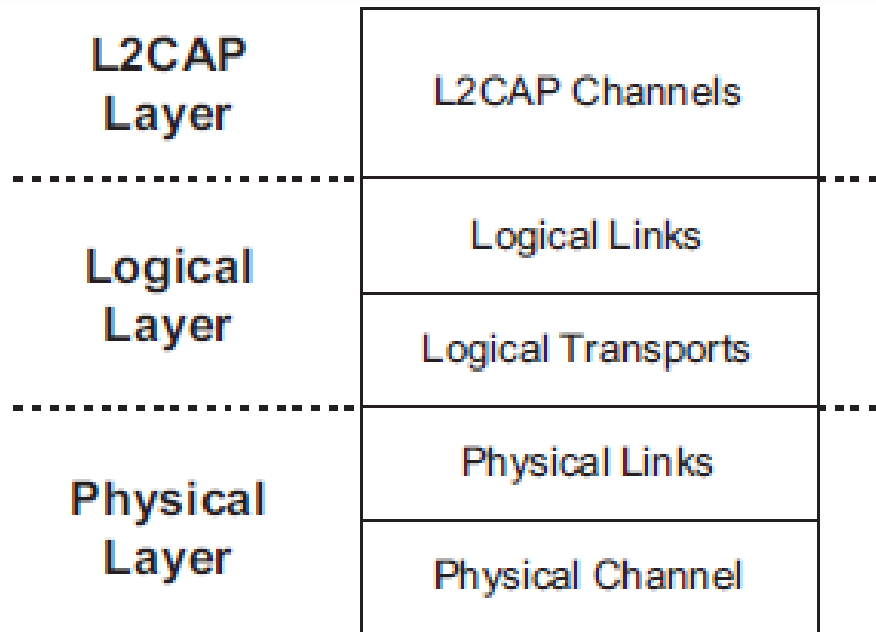
Three Types of Bluetooth Host Controller Interfaces are defined in Bluetooth Specification:

- UART
- USB
- SDIO

Data Transport Architecture



The Bluetooth data transport system follows a layered architecture. All Bluetooth operation modes follow the same generic transport architecture shown left.



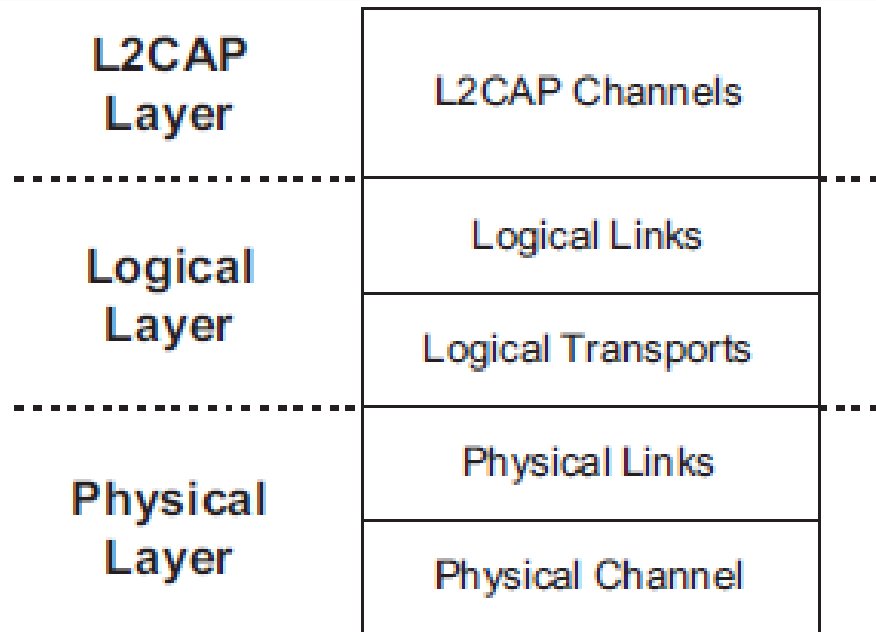
Physical Channel

- All Bluetooth physical channels are characterized by the combination of a pseudo-random frequency hopping sequence, the specific slot timing of the transmissions, the access code and packet header encoding.

Physical Links

- A physical link represents a baseband connection between devices. A physical link is always associated with exactly one physical channel.

Data Transport Architecture (Cont.)



Logical Transports

Between master and slave(s), different types of logical transports may be established, such as Synchronous Connection-Oriented (SCO), Asynchronous Connection-Oriented (ACL), eSCO, CSB, ASB, PSB.

Logical Links

There are six logical links. The control logical links LC and ACL-C are used at the link control level and link manager level, respectively. The ACL-U logical link is used to carry either asynchronous or isochronous user information. The SCO-S, and eSCO-S logical links are used to carry synchronous user information. The PBD logical link is used to carry profile broadcast data.

L2CAP

L2CAP provides connection-oriented and connectionless data services to the upper layer protocols with protocol multiplexing capability, segmentation and reassembly operation, and group abstractions.

General Packet Structure



Access Code			Header						Payload
Preamble	Sync Word	Trailer	LT_ADDR	TYPE	FLOW	ARQN	SEQN	HEC	

Preamble

- 4-bit symbols for DC compensation
- Sync Word
- 64-bit code word derived from LAP of BD_ADDR, to improve timing acquisition

Trailer

- Optional: 4-bit symbols for extend DC compensation

LT_ADDR

- 3-bit Logical Transport Address

TYPE

- 4-bit Type Code

FLOW

- 1-bit Flow Control

ARQN

- 1-bit Acknowledge Indication

SEQN

- 1-bit Sequence Number

HEC

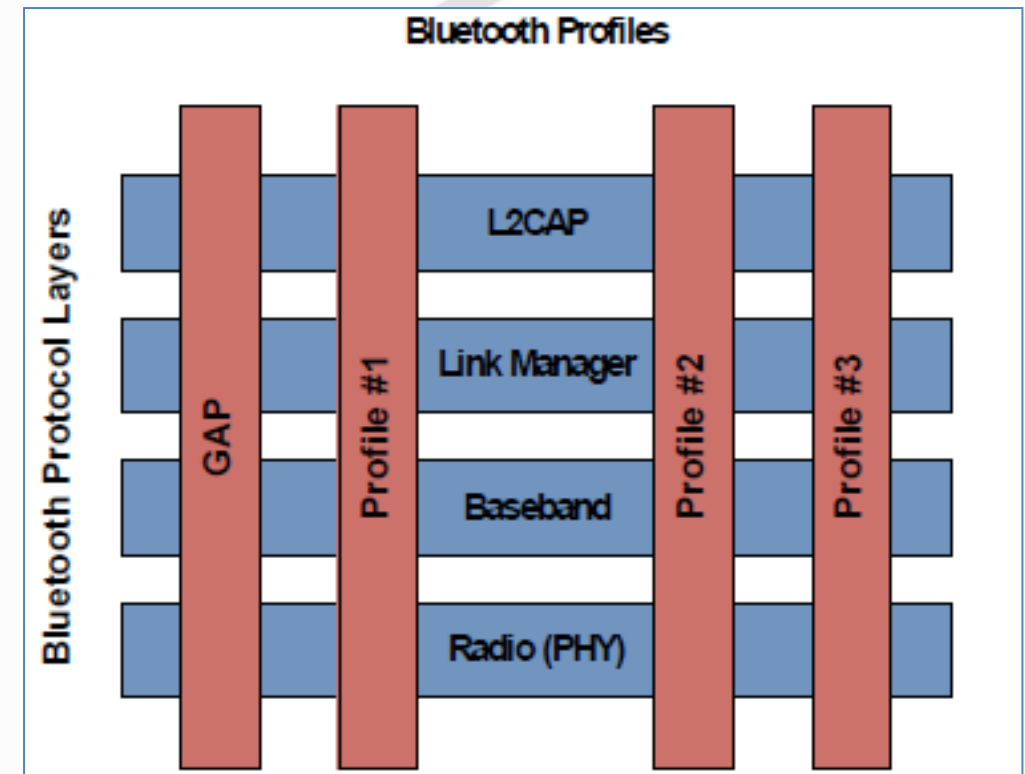
- 8-bit Header Error Check

Payload is the data payload to be transmitted over the air.

Protocol and Profiles



- Protocol is a layer stack.
- Application interoperability in the Bluetooth system is accomplished by Bluetooth profiles.
- Bluetooth profiles define the required functions and features of each layer in the Bluetooth system from the PHY to L2CAP and any other protocols outside of the Core specification.
- The profile defines the vertical interactions between the layers as well as the peer-to-peer interactions of specific layers between devices.



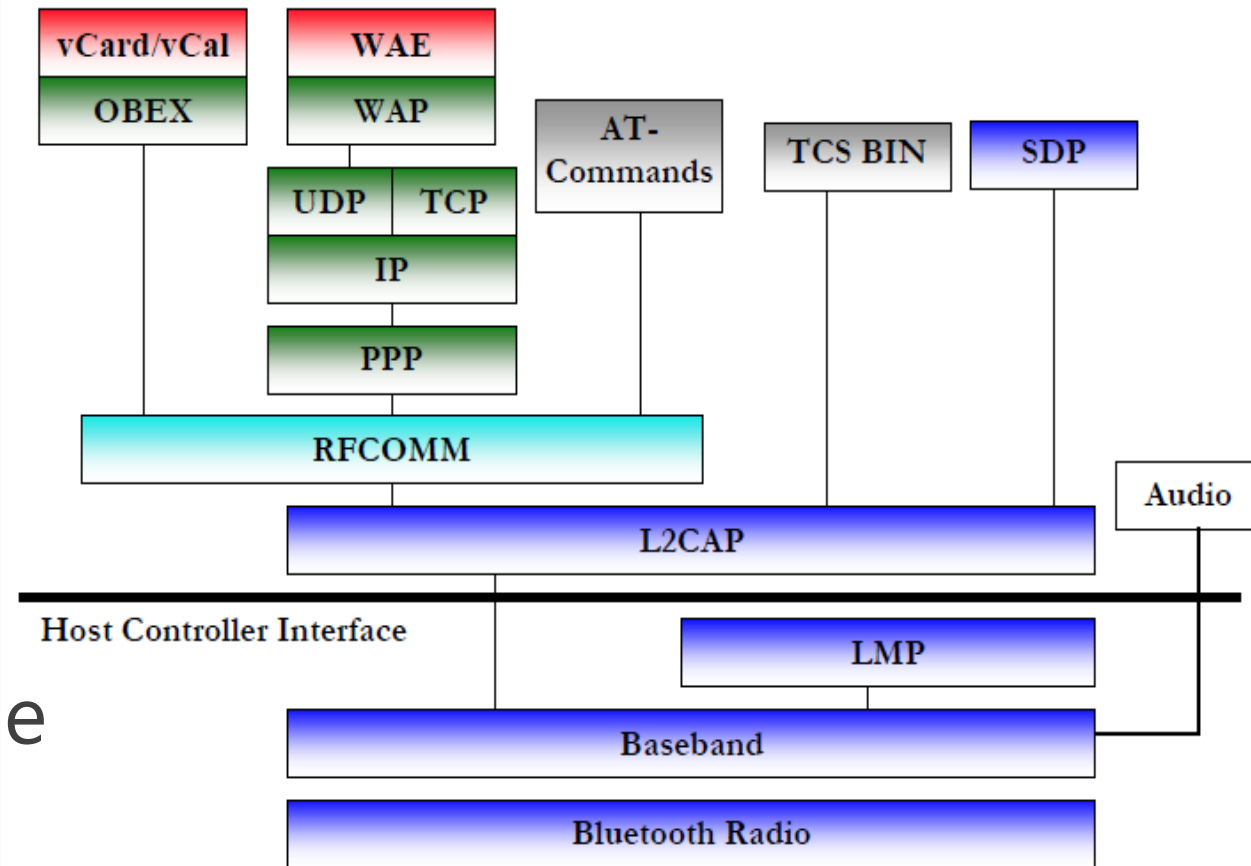
Bluetooth Protocols



Bluetooth Core Protocols are required by most of Bluetooth devices which include:

- Bluetooth Radio
- Baseband
- LMP
- L2CAP
- SDP

Other Protocols are only available when used by device.



Bluetooth Profiles



- Advanced Audio Distribution Profile (A2DP)
- Attribute Profile (ATT)
- Audio/Video Remote Control Profile (AVRCP)
- Basic Imaging Profile (BIP)
- Basic Printing Profile (BPP)
- Device ID Profile (DIP)
- Dial-up Networking Profile (DUN)
- Serial Port Profile (SPP)
- Generic Object Exchange Profile (GOEP)
- Human Interface Device Profile (HID)

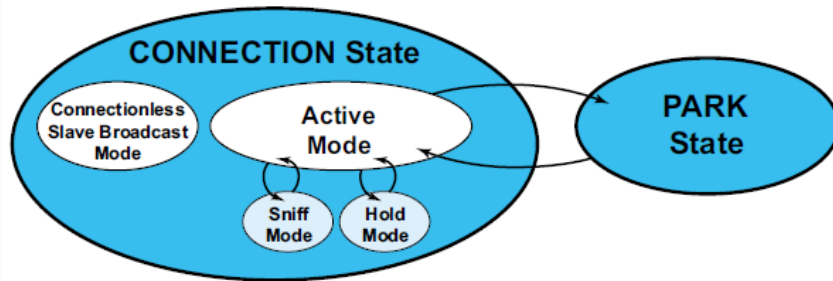


Bluetooth Device States

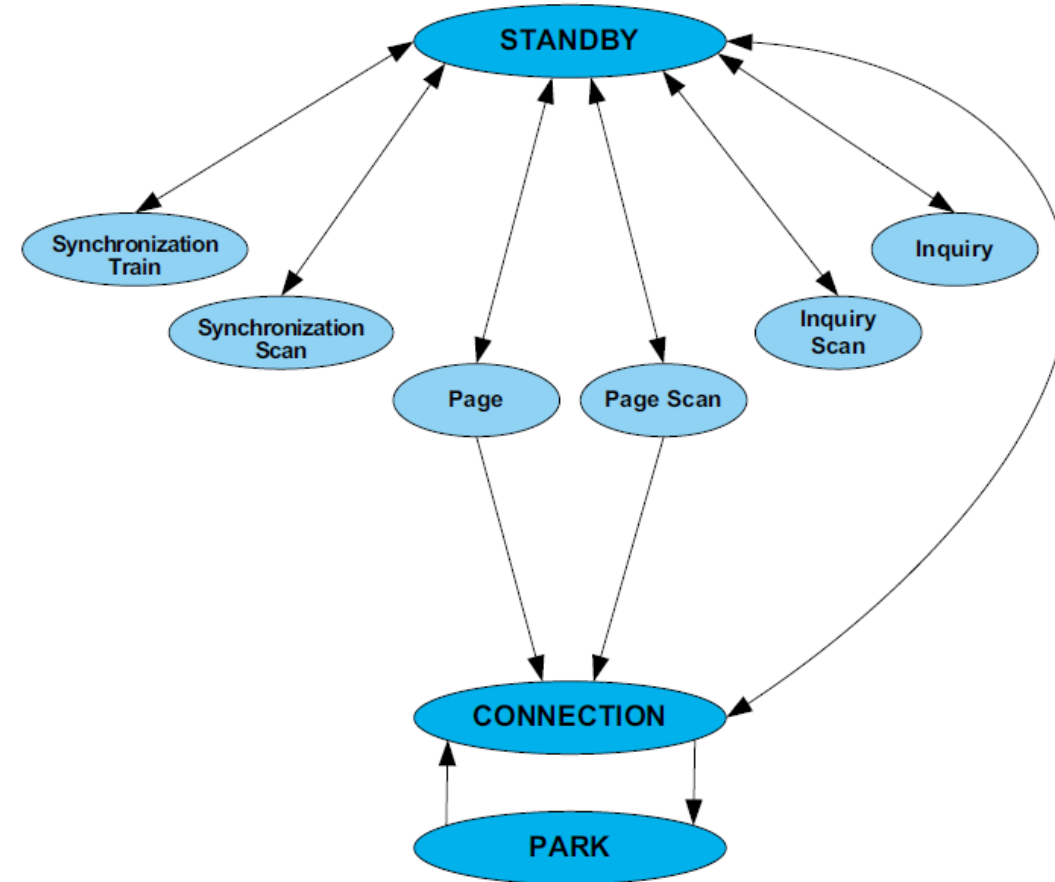


Several states of operation of the devices are defined to support the functions, such as how a piconet is established and how devices can be added to and released from the piconet. There are three major states:

- Standby
- Connection
 - Active
 - Sniff
 - Hold
- Park



Nine substates are defined in Bluetooth specification, not all of them are shown in the figure.

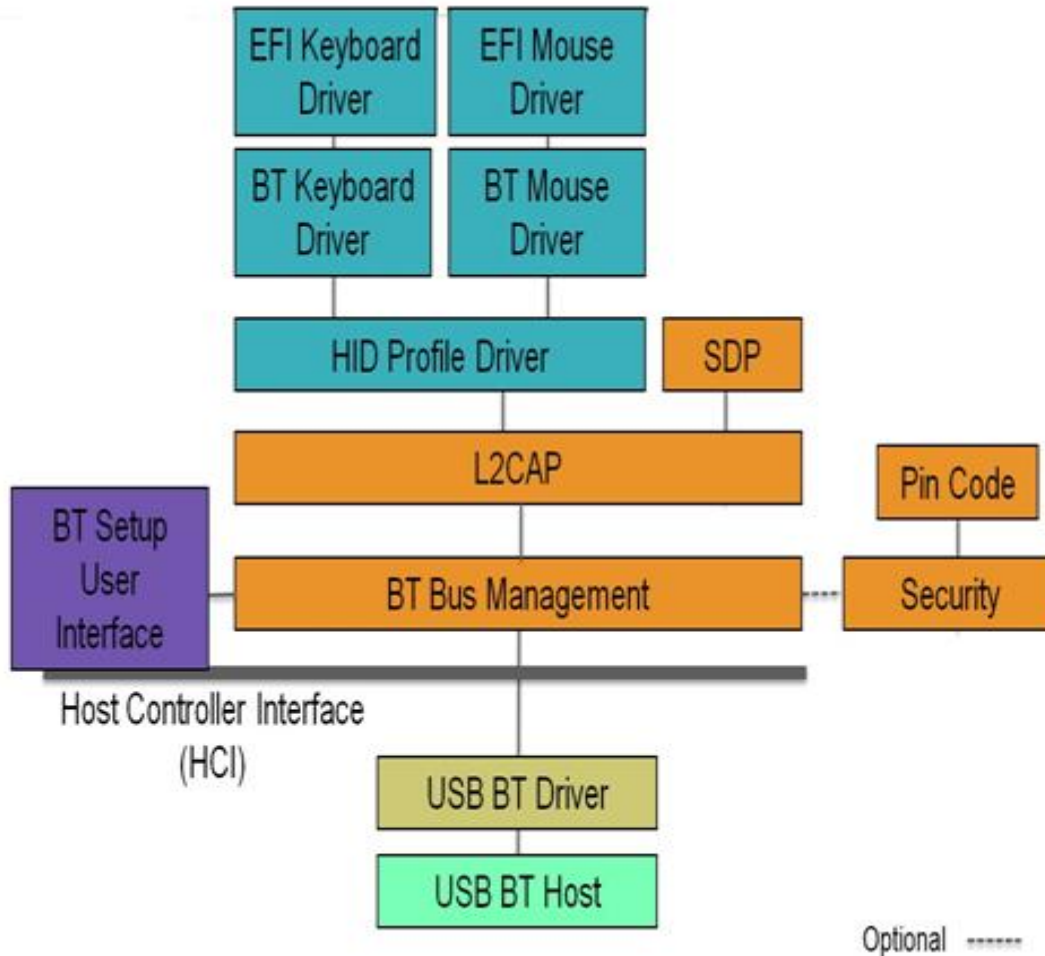




UEFI Bluetooth Stack



UEFI Bluetooth Stack Architecture



An implementation of Bluetooth stack on UEFI are shown which include:

Physical Bluetooth Hardware

USB BT Host

Bluetooth Bus Stack

BT Bus Management, L2CAP, SDP, PinCode, Security

Bluetooth Host Controller Specific

USB BT Driver

HID Profile Specific

HID Profile Driver, BT Keyboard Driver, BT Mouse Driver, EFI Keyboard Driver, EFI Mouse Driver

User Interface

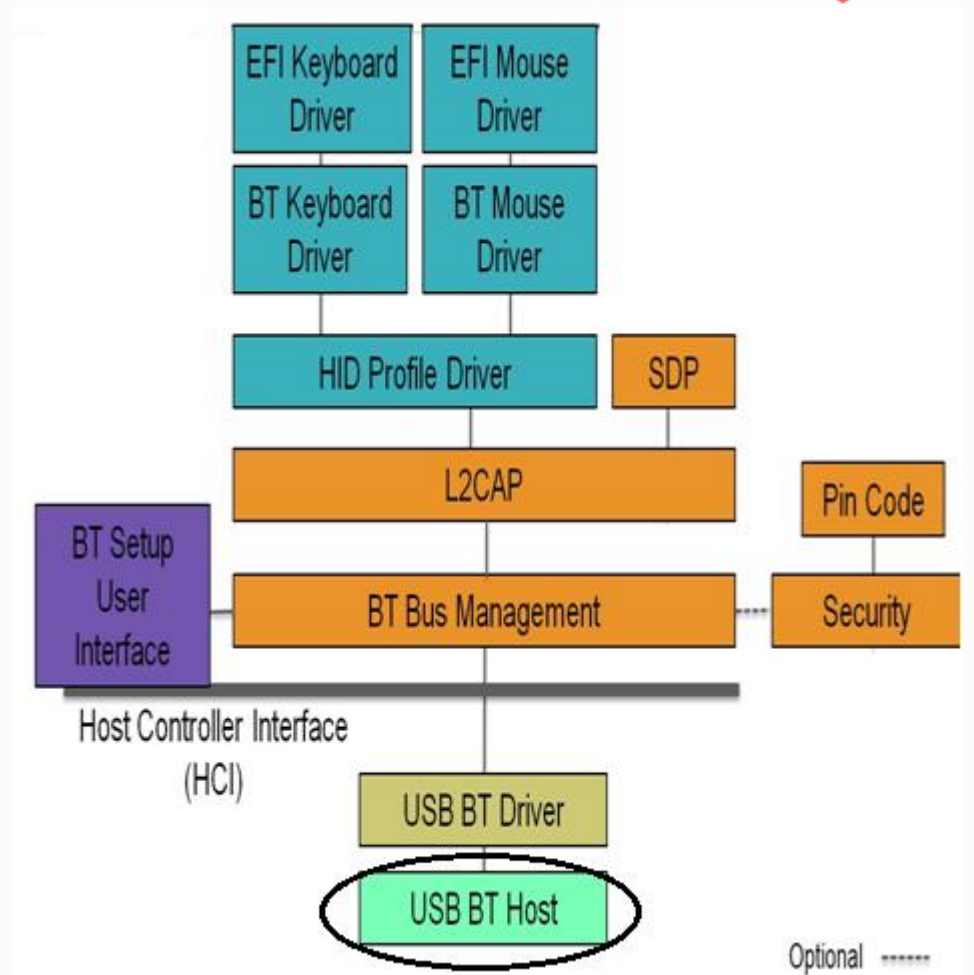
BT Setup User Interface

USB BT Host



The USB BT host can be embodied in one of several ways:

- As a USB dongle (e.g. cabled USB)
- As a USB module integrated into the product and connected internally via a cable or connector
- Integrated onto the motherboard of a notebook PC or other device and connected via circuit board traces with standard USB, Inter-Chip USB or High Speed Inter-Chip USB
- Integrated as a subsystem on a single-chip System-on-Chip (SoC) design connected on-chip as part of a compound device.

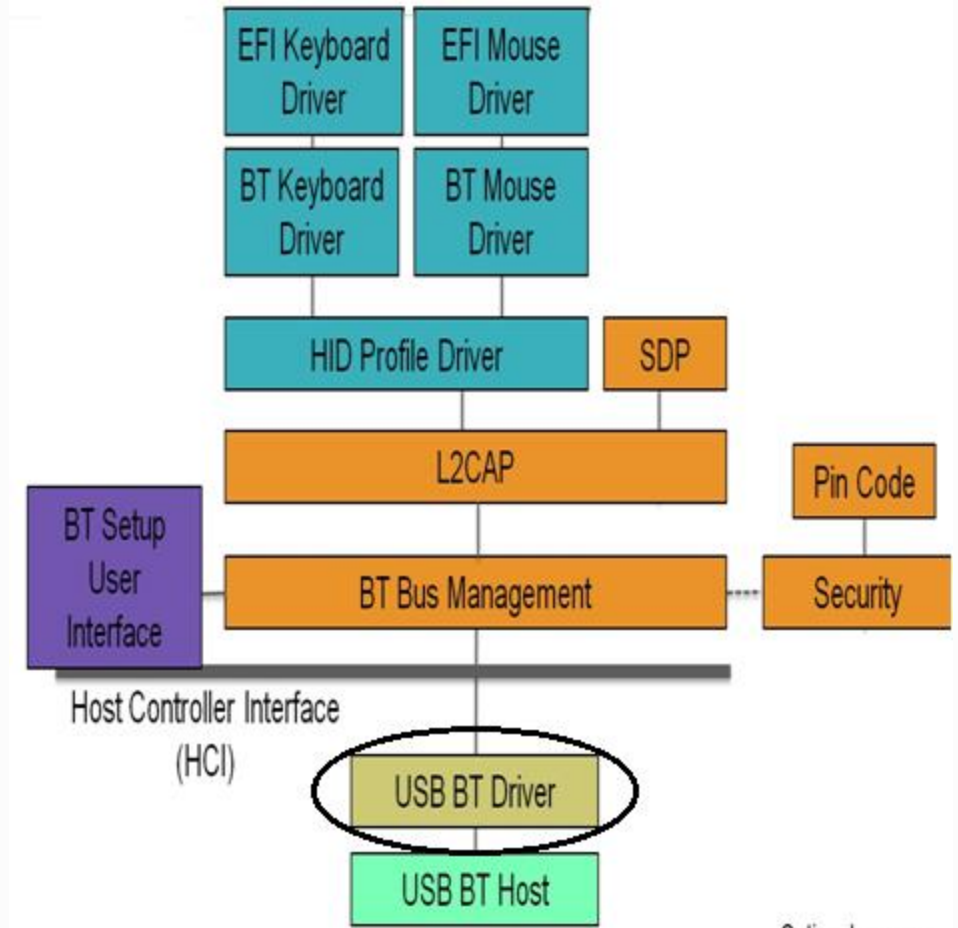


USB BT Driver



USB BT Driver is an interface between USB BT host and BT Bus Management driver which:

- Converts Bluetooth requests to USB packets then sending to USB BT Host.
- Redirects the received USB packets to BT Bus Management driver.



Optional -----

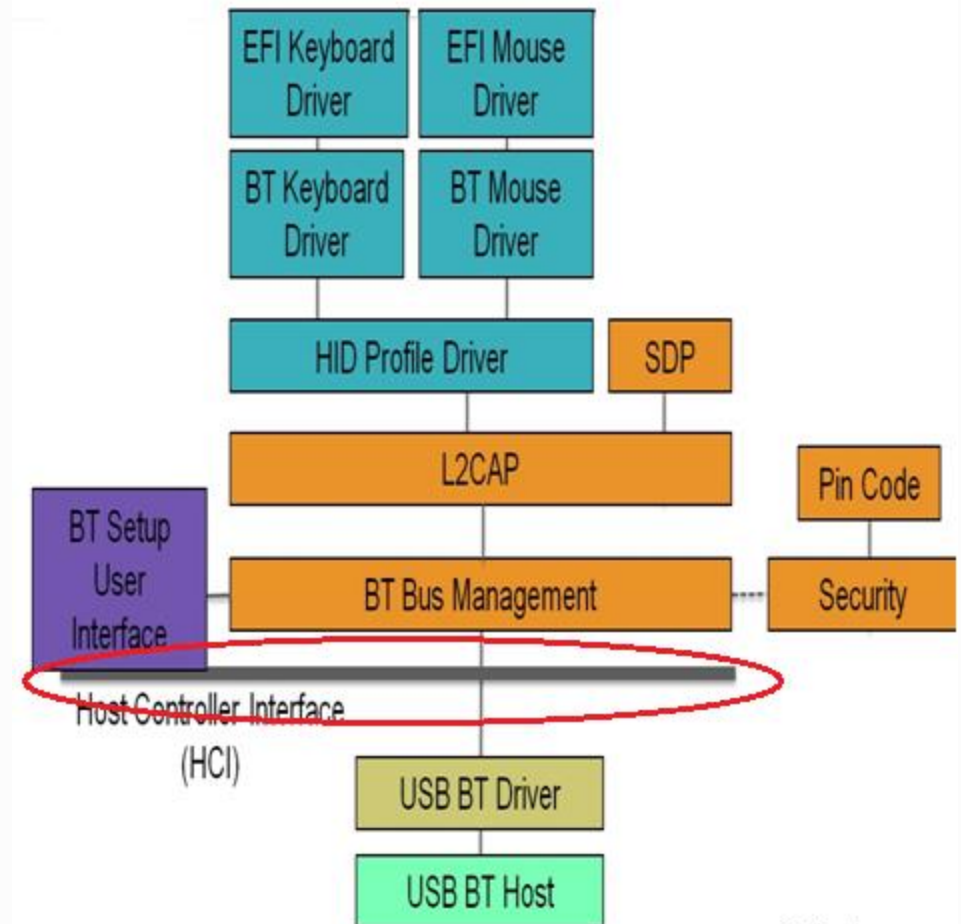
HCI – Host Control Interface



Standardized the communication between the host stack and the controller (the Bluetooth Host)

Provide below Commands:

- Link Control Commands
Inquire, Periodic Inquiry Mode,
- Link Policy Commands
Read Link Policy Settings, Write Link Policy Settings
- Controller and Baseband Commands
Write PIN Type, Read PIN Type
- Information Parameters
Read BD_ADDR, Read Buffer Size
- Status Parameters
Read Failed Contact Counter, Read Link Quality



BT Bus Management



Bluetooth Host Management

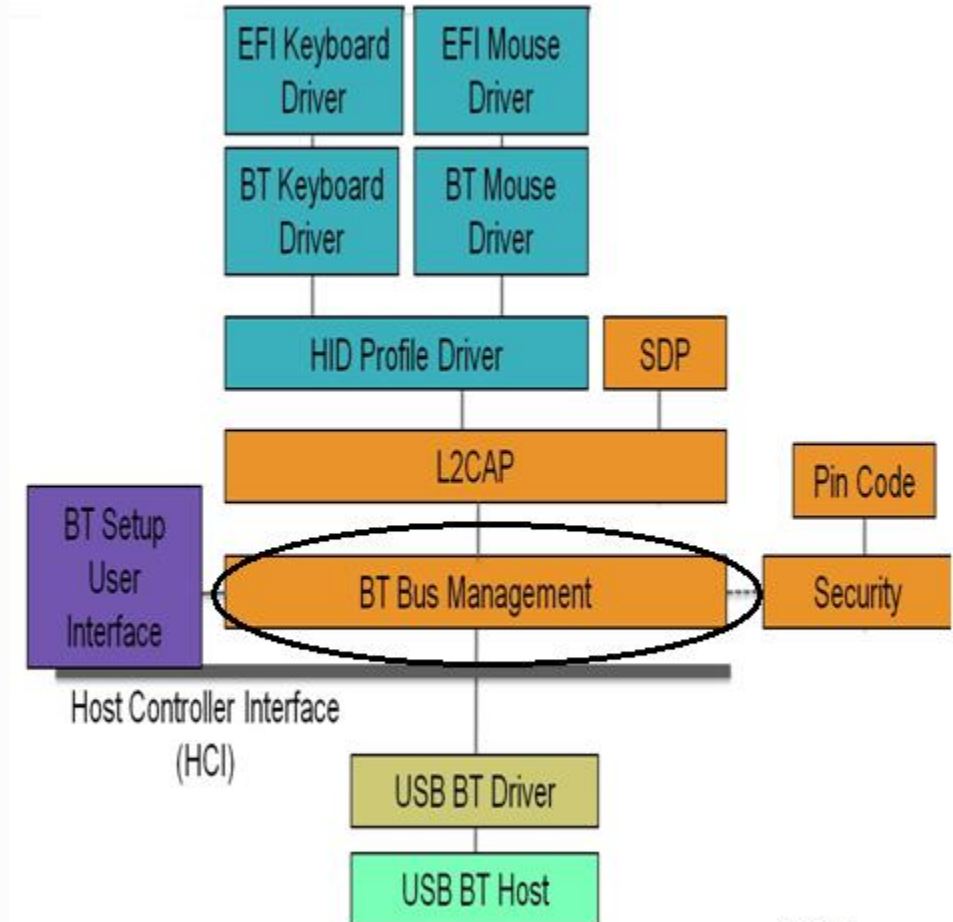
- Bluetooth host initialization

Bluetooth Device Management

- Add/Remove/Search device

Connect / Reconnect Devices

- Connect Bluetooth device which is specified by user.
- Process reconnect request for devices which had been paired.
- Process security(PIN code) request which is determined by user.



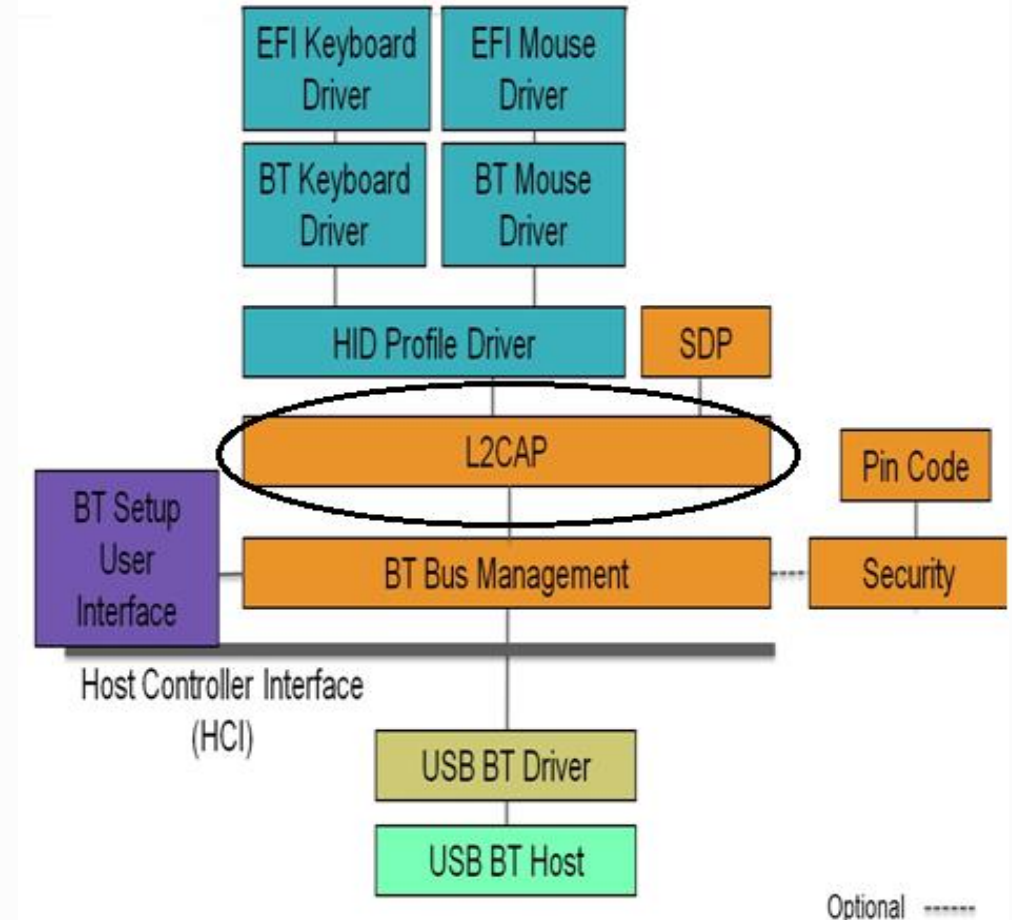
L2CAP - LOGICAL LINK CONTROL AND ADAPTATION PROTOCOL



L2CAP is used within the Bluetooth protocol stack. It passes packets to either the Host Controller Interface (HCI) or on a hostless system, directly to the Link Manager/ACL link.

L2CAP protocol functions include:

- Segmentation and Reassembly of packets.
- Providing one-way transmission management of multicast data to a group of other Bluetooth devices.



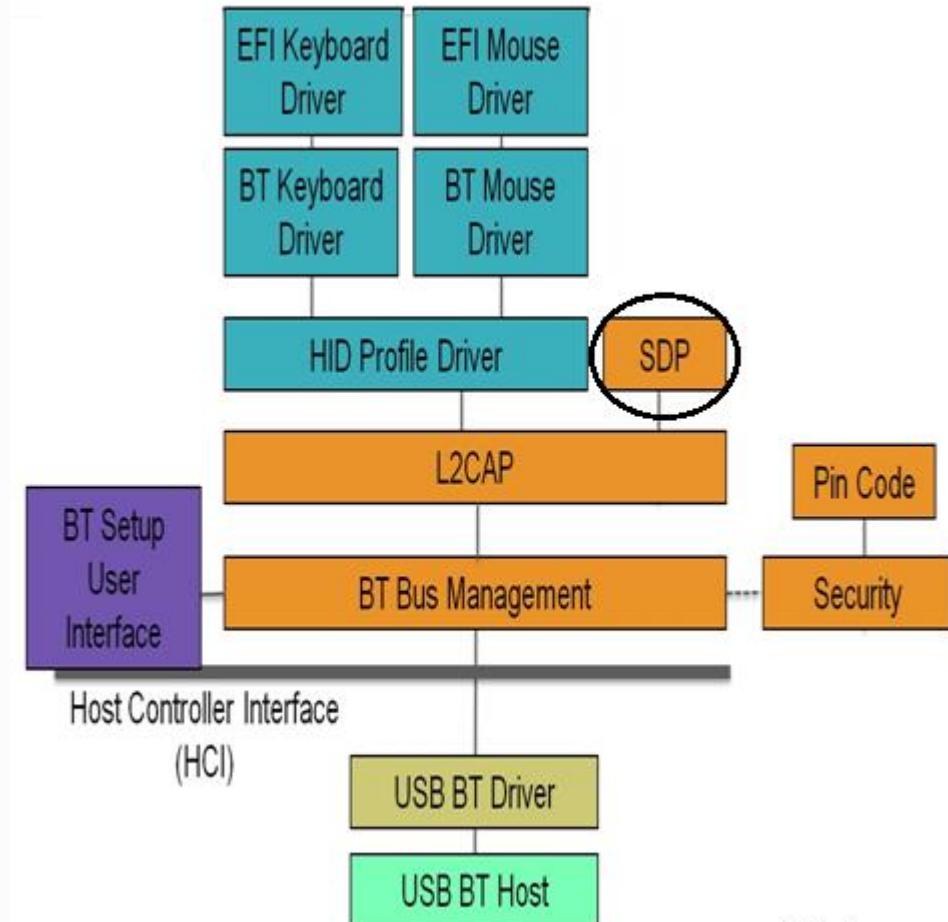
SDP – Service Discovery Protocol



Used to allow devices to discover what services each other support, and what parameters to use to connect between.

With HID profile, this function is utilized to request information of Keyboard/Mouse device, such as:

- Descriptors
 - contains the usage for unpacking report data from device.
- Device Attributes
 - Device Name
 - Reconnection Feature
 - Boot Protocol Feature
 - Version, Timeout, Device Subclass, Country Code... etc



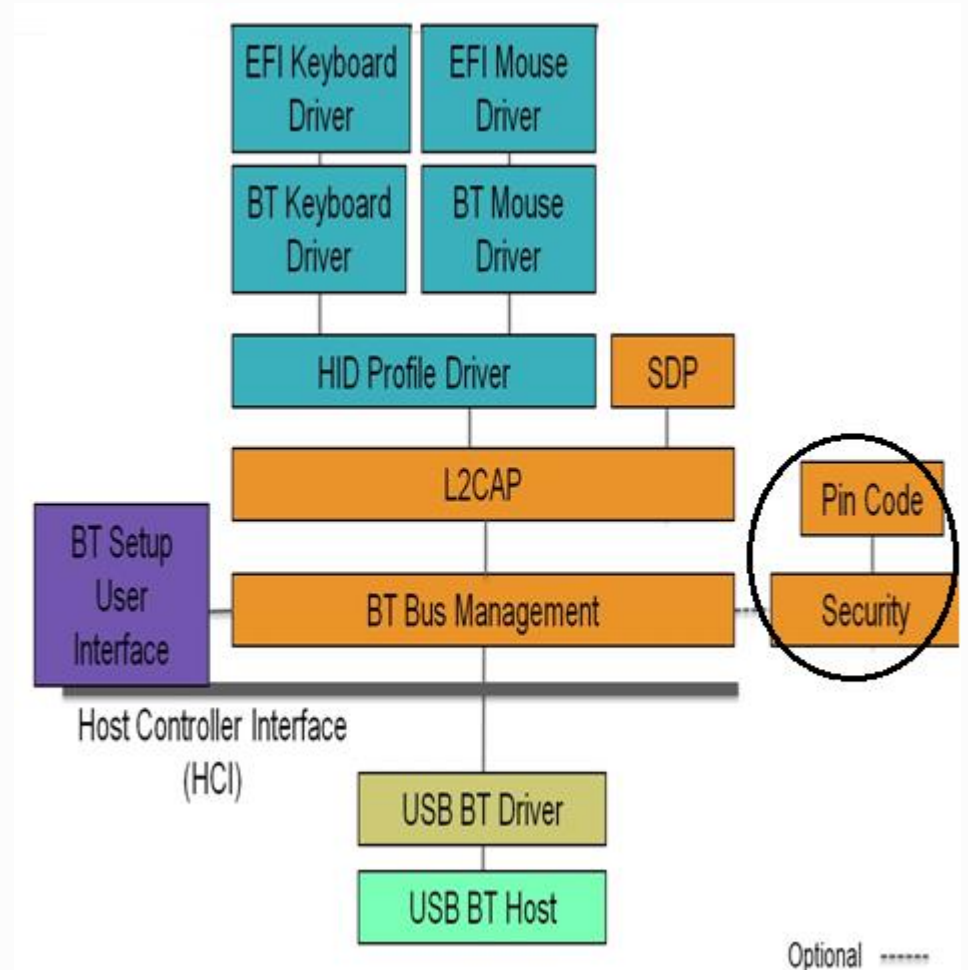
Security, PIN Code



Security is important to ...

- Protect against “man in the middle” attacks
- Process the reconnect event, if the Bluetooth device has been previously connected

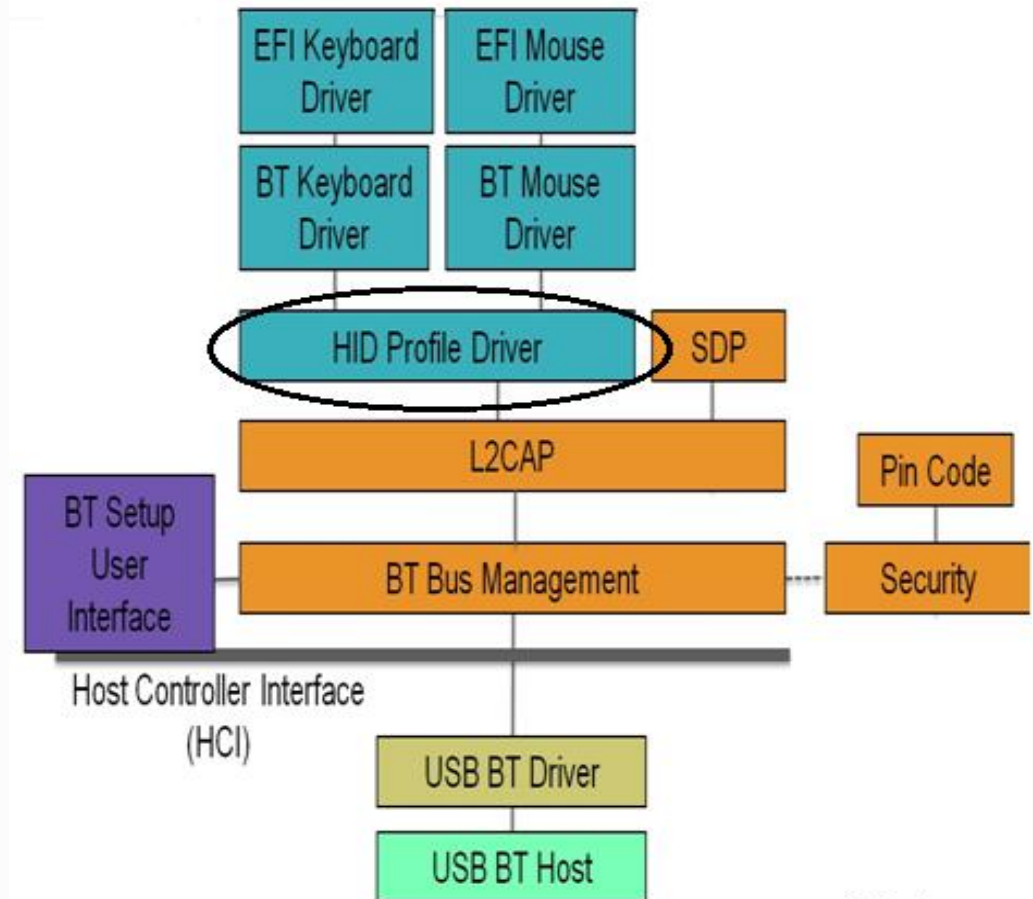
Security is provided as an optional function and can be enable/disable by user, because Keyboard and Mouse are low-level security risk of man in the middle attack. This also provide several advantages, such as simplifying the manufacturing process.



HID Profile Driver



- HID Profile Device functions, such as initialization, device specific requests, device management are supported through HID profile driver.
- This driver response for the management of higher layer drivers, "BT KeyboardDriver", "BT Mouse Driver".
- Manage HID profile common functions, get report descriptor, redirect usage specific packets to target driver.



Optional -----

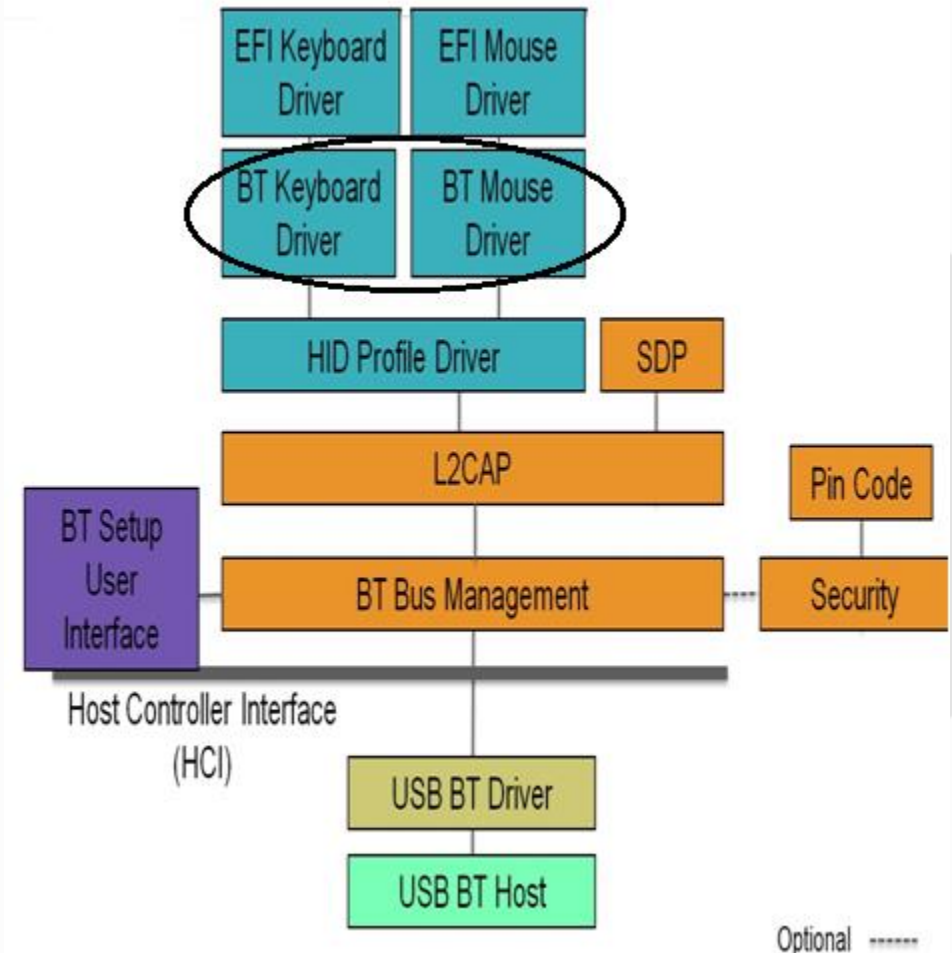
BT Keyboard/Mouse driver



Initialize Bluetooth
Keyboard/Mouse device and
processing received
Keyboard/Mouse report data.

Support HID boot protocol and
report protocol:

- Boot protocol
Use fixed data packet formats.
- Report protocol
Process report data according to
the definition of report descriptor.



EFI Keyboard/Mouse driver

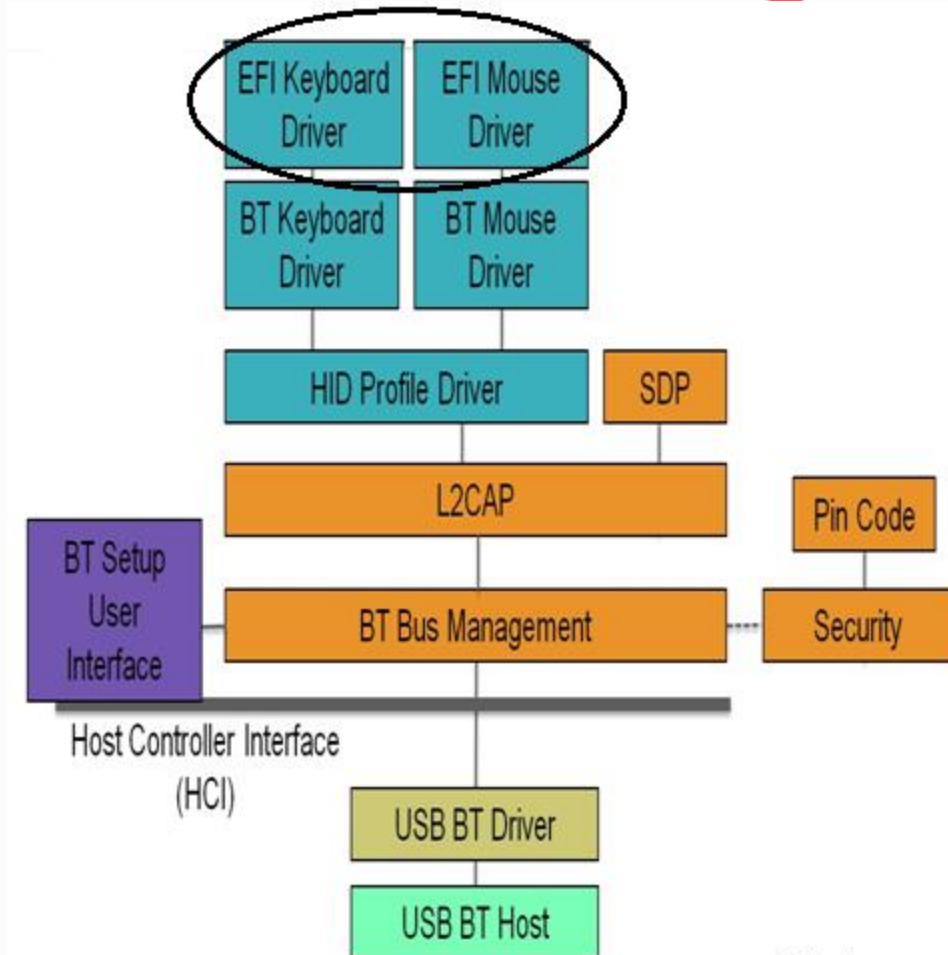


EFI keyboard driver produces

EFI Console I/O protocols:

- EFI Simple Text Input Protocol
- EFI Simple Text Input Ex Protocol

EFI mouse driver produces EFI Simple Pointer Protocol for application to consume.

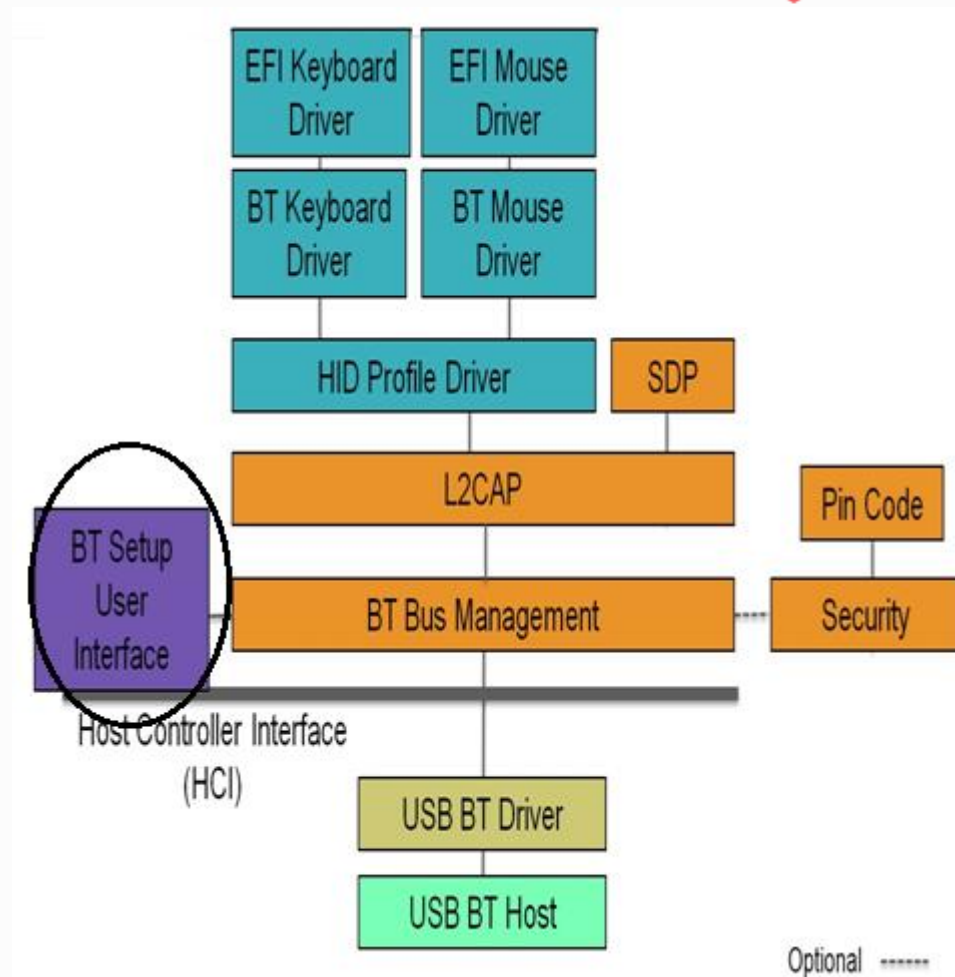


Bt Setup User Interface



The User Interface provides below functions:

- Device Enumeration
List all Bluetooth devices which are in the service range of host controller.
- Security Feature
Enable/Disable PIN code by user.
- Connect/Disconnect Device
Connect Bluetooth device which is specified by user.
- Notification for Connection Failure.



References



Bluetooth Core Specification v4.1

https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=282159

Bluetooth Protocol Architecture v1.0

Human Interface Device Profile v1.1

Summary



Bluetooth has many advantages ...

- Low power consumption & low cost
- Availability on mobile platforms
- Numerous devices already in the market

Implementing a UEFI Bluetooth stack enables many new applications



Q&A



For more information on the Unified EFI Forum and UEFI Specifications, visit <http://www.uefi.org>



presented by

