Bluetooth® technology

from a Linux systems engineer's perspective

George Kiagiadakis, March 2025

$ whoami

- CSD UoC Alumnus (class06)
- Linux Systems / Multimedia Engineer
- Principal Software Engineer @ Collabora (14+ years)
- Collabora: Open Source Software Consulting

COLLABORA

pipewire

WirePlumber

Today's Agenda

Bluetooth: background & history

The Bluetooth protocol stack

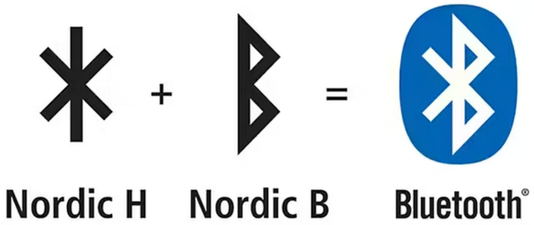
Recent advances in Bluetooth

Bluetooth on Linux

A few more things about HCI

Bluetooth®: Background & history

What is Bluetooth?



- Short-range wireless technology standard
- Data exchange between mobile devices up to 10m apart (nowadays more)
- Operating in the 2.4 GHz band
- Started in 1994 by Ericsson (idea from late 80's)
- Intended as a wireless replacement for RS-232 (serial) data cables
- Named after King Harald 'Bluetooth' Gormsson of Denmark (10th century AD) (intended as a codename...)

Version history

(1/3)

- **Bluetooth 1.0 & 1.0B** (1999): Data communication (no audio), 721 Kbps, many issues (!)
- **Bluetooth 1.1** (2001): Headset & Hands-Free profiles (HSP, HFP)
- **Bluetooth 1.2** (2003): Adaptive Frequency Hopping (AFH), Advanced Audio Distribution Profile (A2DP), Extended Synchronous Connections (eSCO), flow control & retransmissions
- **Bluetooth 2.0** (2004): Enhanced Data Rate (EDR) → 2.1 Mbps, Audio/Video Remote Control Profile (AVRCP)
- **Bluetooth 2.1** (2007): Secure Simple Pairing (SSP)
- **Bluetooth 3.0** (2009): Bluetooth High-Speed (HS) → switch to Wi-Fi for transfers (24 Mbps)

Version history

(2/3)

- **Bluetooth 4.0** (2010): Bluetooth Low Energy (BLE or simply LE)
- **Bluetooth 4.1** (2013): Incorporated various Core Specification Addenda (CSA), Wide band speech (mSBC)
- **Bluetooth 4.2** (2014): LE secure connection, link layer privacy, IPv6/6LoWPAN
- **Bluetooth 5** (2016): LE Long Range → 240m outdoors, higher speed at the expense of range → 50 Mbps (classic), 2 Mbps (LE)

Version history

(3/3)

- **Bluetooth 5.1** (2019): Angle of Arrival (AoA) and Angle of Departure (AoD) location tracking, Bluetooth Mesh
- **Bluetooth 5.2** (2020): LE Audio, LC3 codec, Auracast, LE Isochronous Channels, LE Power Control
- **Bluetooth 5.3** (2021): Stability, security and efficiency improvements
- **Bluetooth 5.4** (2023): Device advertisement improvements
- **Bluetooth 6.0** (2024): Channel Sounding (distance measurement), lower latency Isochronous Channels

Core & Profile specifications

- **Core Specification**
  - Basic building blocks
  - Low-level protocols, layers, features
- **Profile Specifications (“Profiles”)**
  - Specific use cases
  - Higher level protocols & procedures (on top of Core)
  - Follow their own versioning scheme

Some profiles...

Advanced Audio Distribution Profile	A2DP
Audio/Video Remote Control Profile	AVRCP
Hands-Free Profile	HFP
Headset Profile	HSP
Phone Book Access Profile	PBAP
Message Access Profile	MAP
Basic Imaging Profile	BIP
Human Interface Device Profile	HID
Basic Audio Profile	BAP
Generic Attribute Profile	GATT

Governance

- Bluetooth Special Interest Group (SIG)
  - Non-profit organisation, since 1998, established by Ericsson, IBM, Intel, Nokia, and Toshiba
- More than 35,000 member companies
- Owns Bluetooth standards and licenses the technology to member companies
- Certification program
  - Devices must pass qualification to use the Bluetooth trademark

SIG membership levels

https://www.bluetooth.com/develop-with-bluetooth/join/member-directory/

Adopter	• Free membership tier • Access to Bluetooth specifications and tools • Allows companies to implement Bluetooth technology in their products
Associate	• Paid membership tier with an annual fee (varies based on company size) • Provides additional benefits, such as early access to draft specifications • Eligible to participate in working groups and committees
Promoter	• Exclusive membership tier for a small group of companies (currently: **Apple, Ericsson, Intel, Lenovo, Microsoft, Nokia, and Toshiba**) • One seat (and one vote) each in the SIG's board of directors • Leadership role - considerable influence

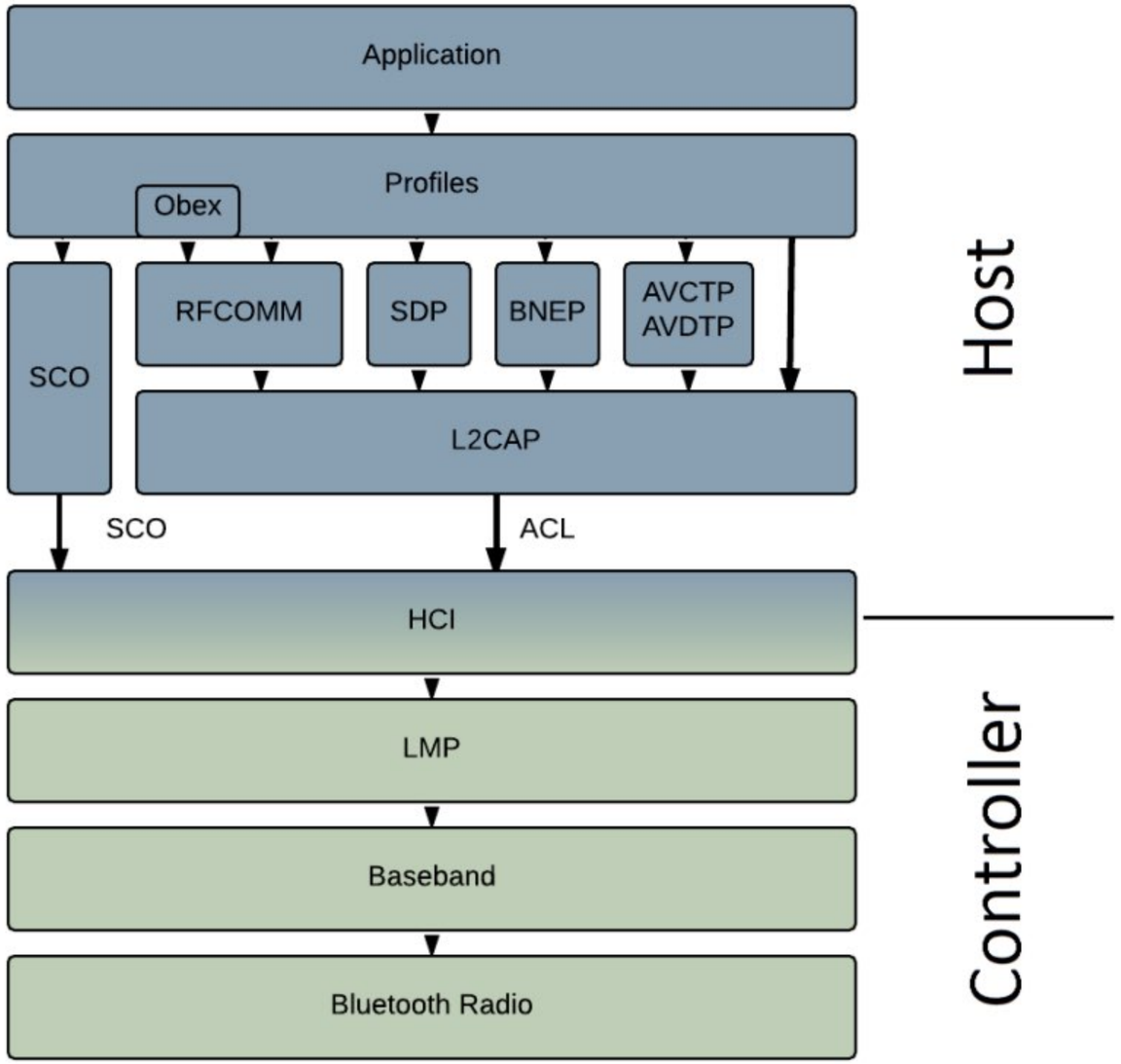
The Bluetooth® protocol stack

Once upon a time, there was a protocol stack

… until they became two

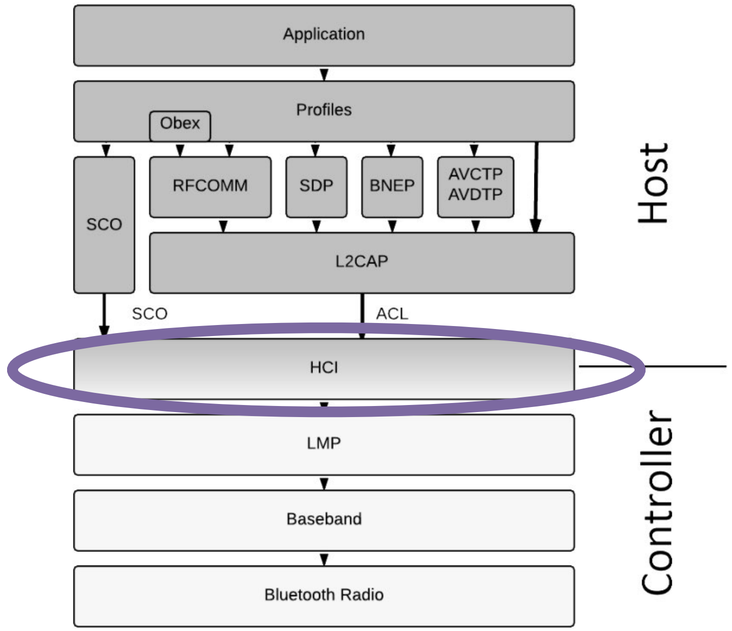
- **Bluetooth Classic:** Also referred to as “BR/EDR”
  - BR: Basic Rate (introduced in Bluetooth 1.0, 1999)
  - EDR: Enhanced Data Rate (introduced in Bluetooth 2.0, 2004)
  - Same protocol stack, radio & transport layers have differences
- **Bluetooth LE:** i.e. Low Energy (introduced in Bluetooth 4.0, 2010)
  - Entirely new protocol stack
  - Co-exists with Classic (served by the same controller)

The Classic stack



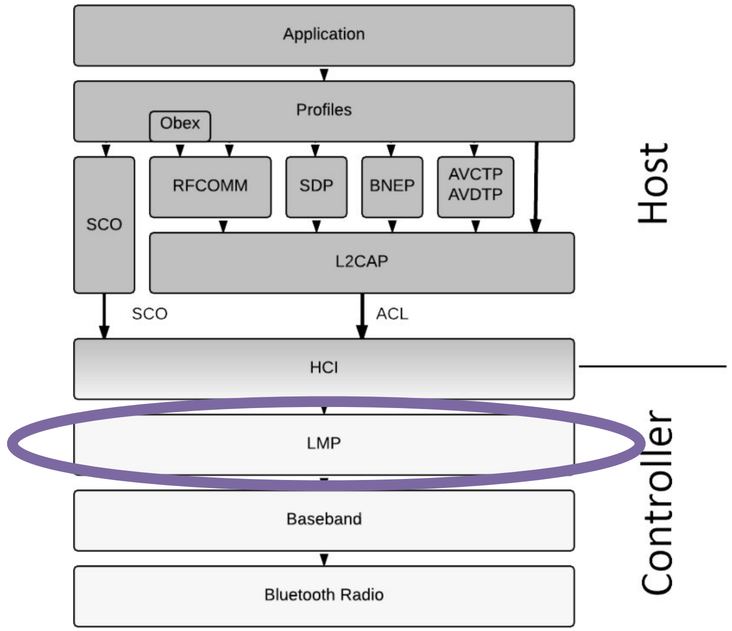
HCI: Host-Controller Interface

- Standardised interface to talk to Bluetooth controllers (“adapters”)
- Command / Reply / Event / Data packets
- Transported over USB (PCs) or UART (embedded)
- Never goes over-the-air
- Optional on very small devices (ex. headsets)



Link layer

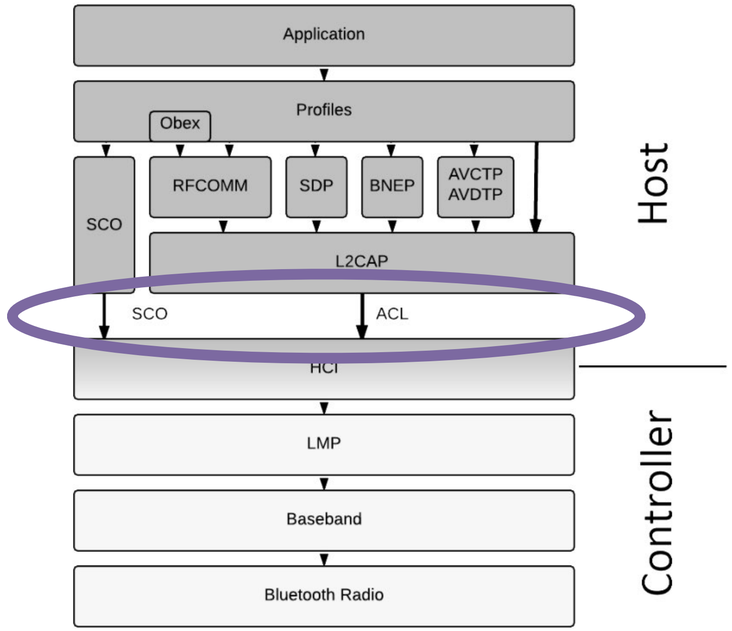
- Link Manager (LM)
  - Controls the radio link between the local controller and the remote Bluetooth device
  - Uses the Link Management Protocol (LMP) to negotiate the link
- Link Controller (LC)
  - Lower layer, provides services to the LM



Transport layer

Asynchronous Connection-oriented Logical transport (ACL)

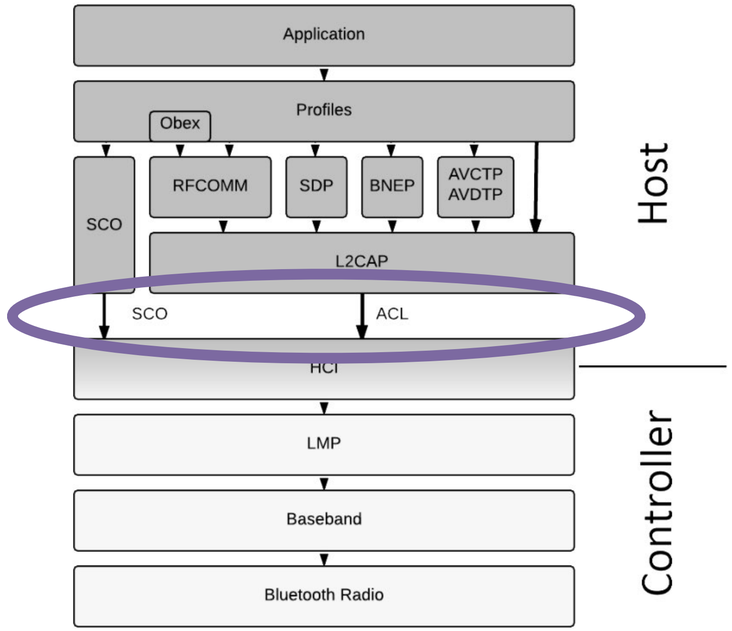
- Standard transport
- Uses TDMA (time division multiple access) to carry packets
- Packet acknowledgments & automatic retransmissions
- Configurable
  - Packet length (1, 3 or 5 time slots)
  - RF Modulation (EDR feature for higher data rate)
  - Forward Error Correction (lower rate, more reliable)



Transport layer

Synchronous Connection Oriented transport (SCO)

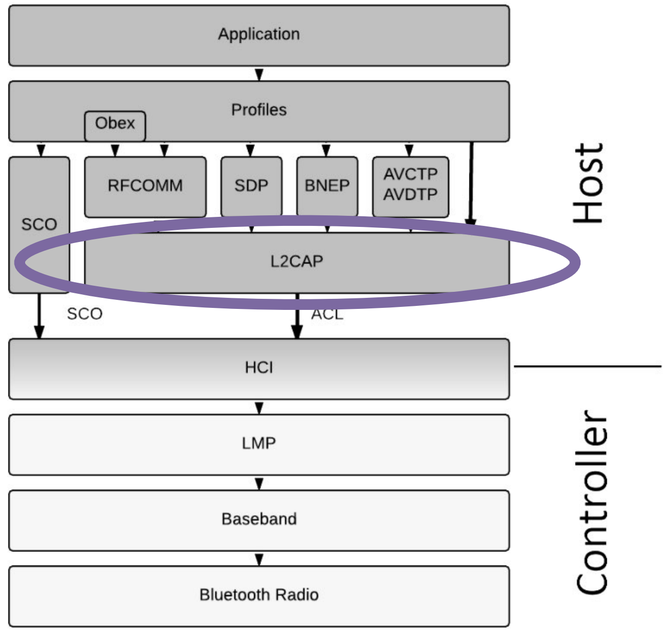
- Used for voice data
- Reserved time slots transmitted every Tsco
- No retransmissions, but optional FEC
- eSCO (enhanced SCO): with retransmissions, since Bluetooth 1.2



Middleware layer

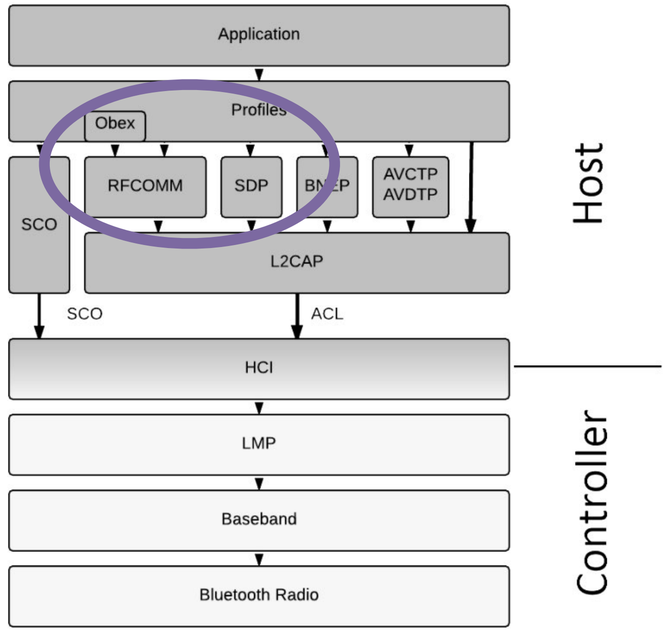
L2CAP: Logical Link Control and Adaptation Protocol

- Logical connections over ACL
- Provides multiplexing, segmentation and reassembly
- Quality of service (QoS) for higher layer protocols
- Encapsulates most other protocols
- (I would say, it's equivalent to TCP)



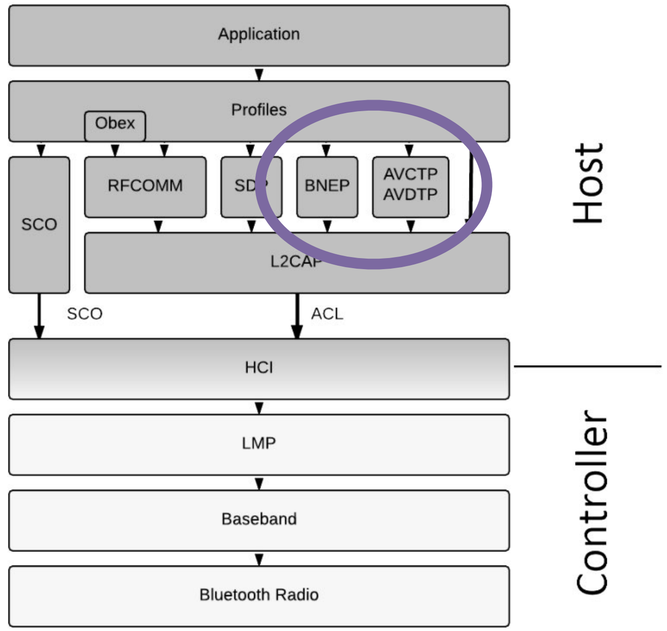
Application protocol layer

- SDP: Service Discovery Protocol
- RFCOMM: Radio Frequency Communications
  - Serial over L2CAP
  - OBEX: Object Exchange

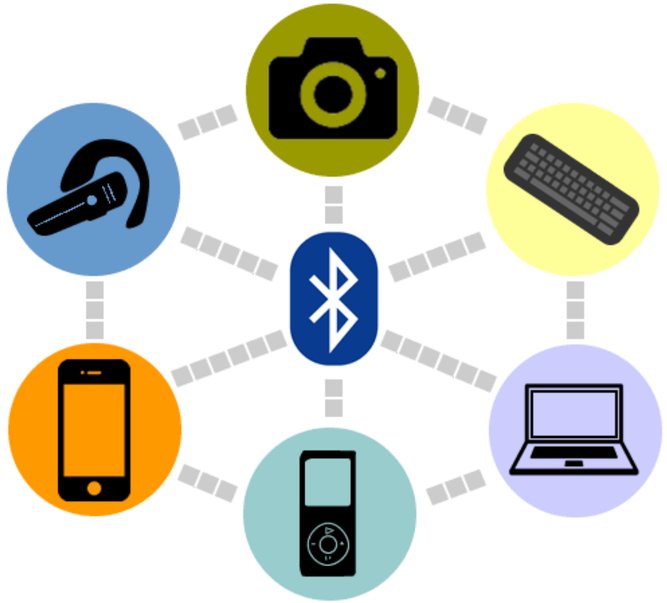


Application protocol layer

- AVDTP: Audio/Video Distribution Transport Protocol (RTP over L2CAP)
- AVCTP: Audio/Video Control Transport Protocol
- BNEP: Bluetooth Network Encapsulation Protocol (TCP/IP encapsulation)



Classic Profiles



Generic Access Profile

(GAP)

- Device discovery, connection & security management
- Supports multi-profile operation (discovery, arbitration)
  - Each profile role is listed with a UUID
- Discovery is inquiry-based
- Roles: Initiator & Acceptor
- Utilises the Secure Simple Pairing (SSP) protocol to establish secure connection

OBject EXchange
(OBEX)

- Client-server protocol designed for infrared (IrDA), operates over RFCOMM
- GET / PUT / ABORT operations, similar to HTTP
- *binary headers* and *stateful*, unlike HTTP
- File Transfer Profile (FTP)
  Phone Book Access Profile (PBAP)
  Message Access Profile (MAP)
  Basic Imaging Profile (BIP)
  Basic Printing Profile (BPP)

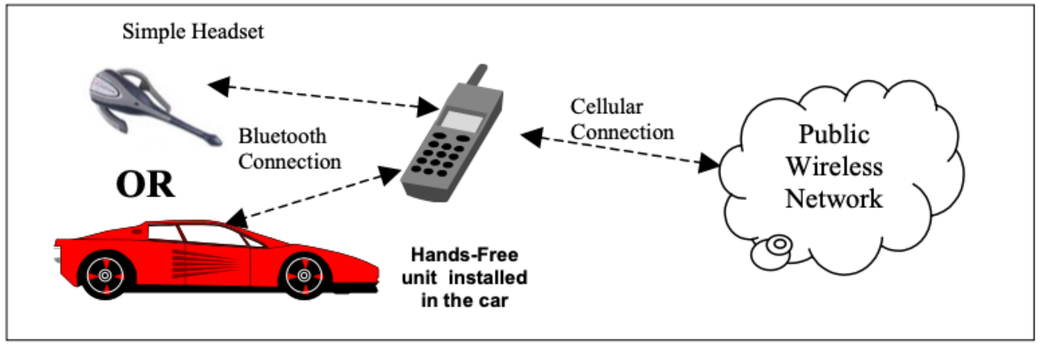
Advanced Audio Distribution Profile (A2DP)

- High-quality audio streaming (one direction)
- Usually for media playback to headphones/speakers or car audio systems
- Codecs: SBC (Sub-Band Codec, mandatory), AAC, aptX, Opus, LDAC, other proprietary...
- Uses AVDTP (i.e. RTP)
- Roles: Source (SRC), Sink (SNK)
- Often paired with the Audio/Video Remote Control Profile (AVRCP)

Headset Profile / Hands-Free Profile

(HSP / HFP)

- Bi-directional audio for headsets & hands-free units, intended for phone calls
- Uses SCO link for audio & RFCOMM for control (with AT commands!)
- Roles: Headset (HS), Audio Gateway (AG)

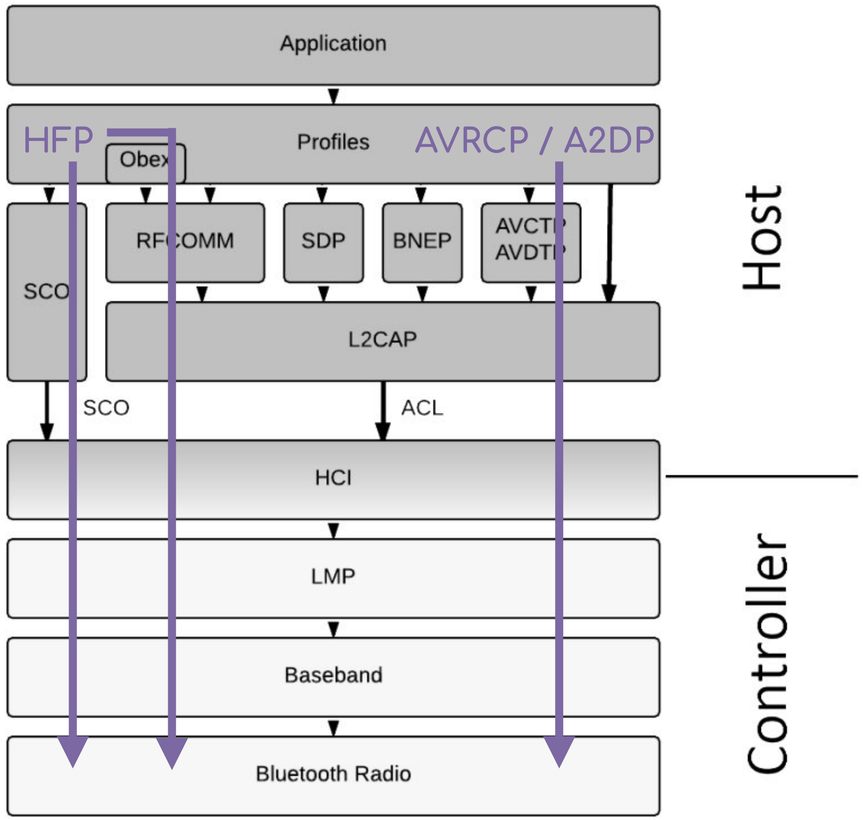


Audio Quality	Narrow Band (8 kHz, CVSD codec)	Wide Band (16 kHz, mSBC codec, since v1.6) Super Wide Band (32 kHz, LC3-SWB codec, since v.1.9)
Control Features	Basic (answer, hangup, volume)	Advanced (dial, hold, 3-way calls, voice activation, ...)
Use Case	Basic headset functionality	Hands-free devices in cars or advanced headsets
Audio Transmission	SCO	SCO or eSCO
Compatibility	Older devices	Modern devices, replacing HSP

Audio profiles

Comparison

- A2DP uses RTP over L2CAP & ACL
  (buffering, like typical network streaming)
- HSP/HFP use SCO
  (synchronous, almost like analog)
- Embedded controllers link SCO audio
  directly to the system's sound processor
  - Old school efficiency or good business?
  - PITA for developers



Notice any problems?

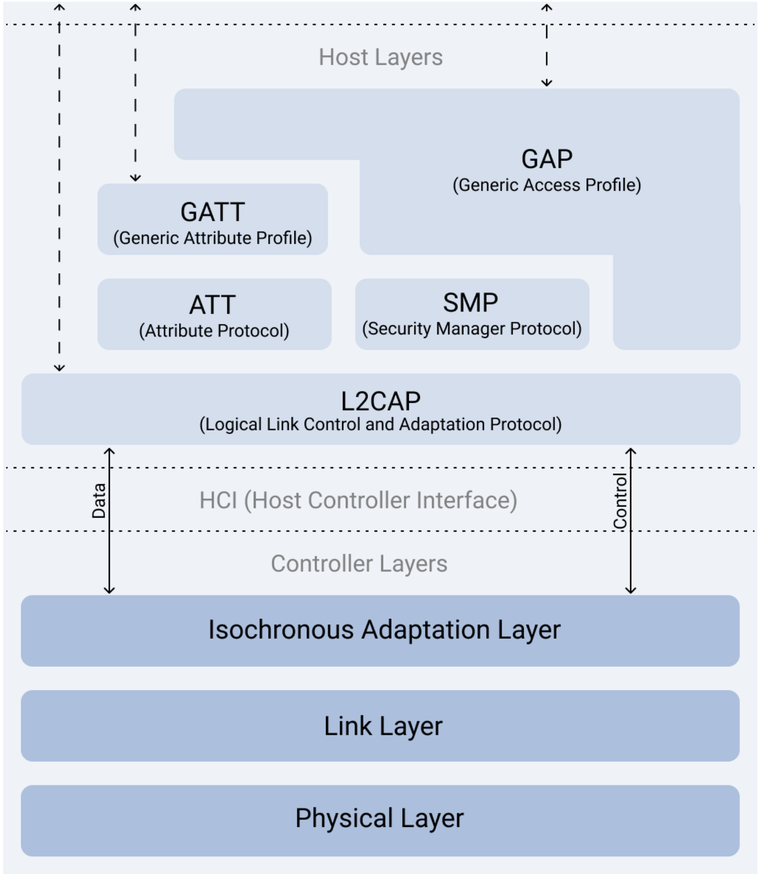
… as far as audio profiles are concerned

- Yes! High quality audio (A2DP) is uni-directional only
- There isn’t enough bandwidth (at least with the base SBC codec)
- Solutions exist, but few devices implement them (they are hacks, not standardised)
  - ex. FastStream codec (SBC variant) allows duplex mode in A2DP
- Headsets & earbuds switch to HFP (ideally with mSBC) when the microphone is in use
- Is this why it sounds terrible? …. yes and no
  - Headsets & earbuds also apply DSP to optimise for speech

The LE stack

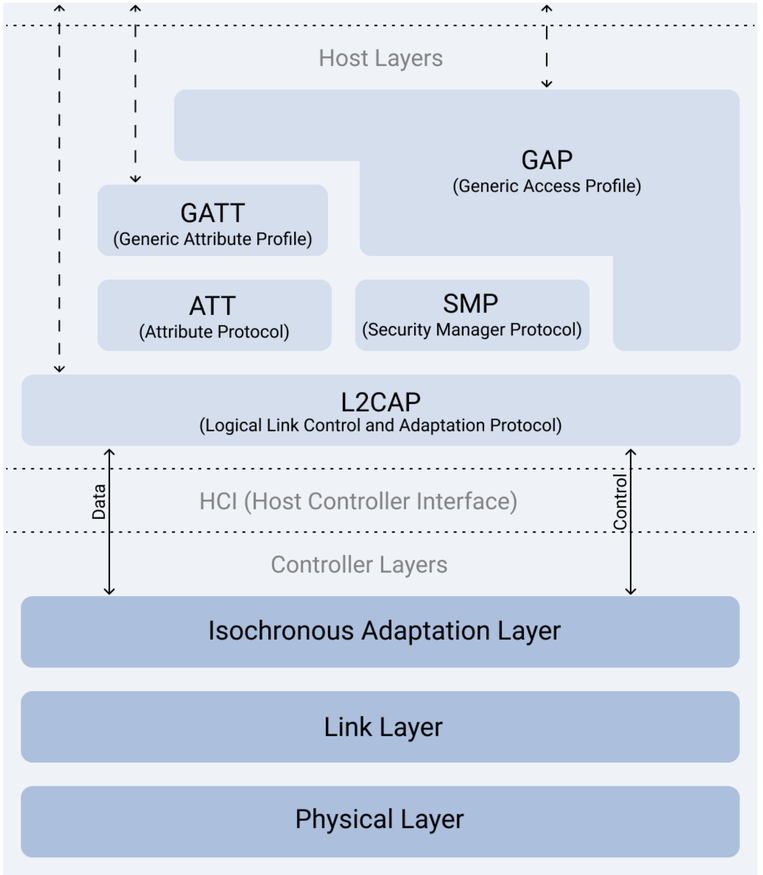
LE Background

- Designed for IoT, wearables and other devices needing low energy consumption
- Gradually replacing all the Classic functionality with modern alternatives
- Redesigned Physical Layer (PHY)
- Simplified Link Layer (LL)
- Same Host-Controller Interface (HCI) & L2CAP



Attribute Protocol (ATT)

- Data as “attributes”
  - Type, Handle, Permissions, Value
- Client - Server architecture
- Server may send notifications (no ACK) or indications (with ACK)
- Core component of GATT (Generic Attribute Profile)



Attribute Protocol

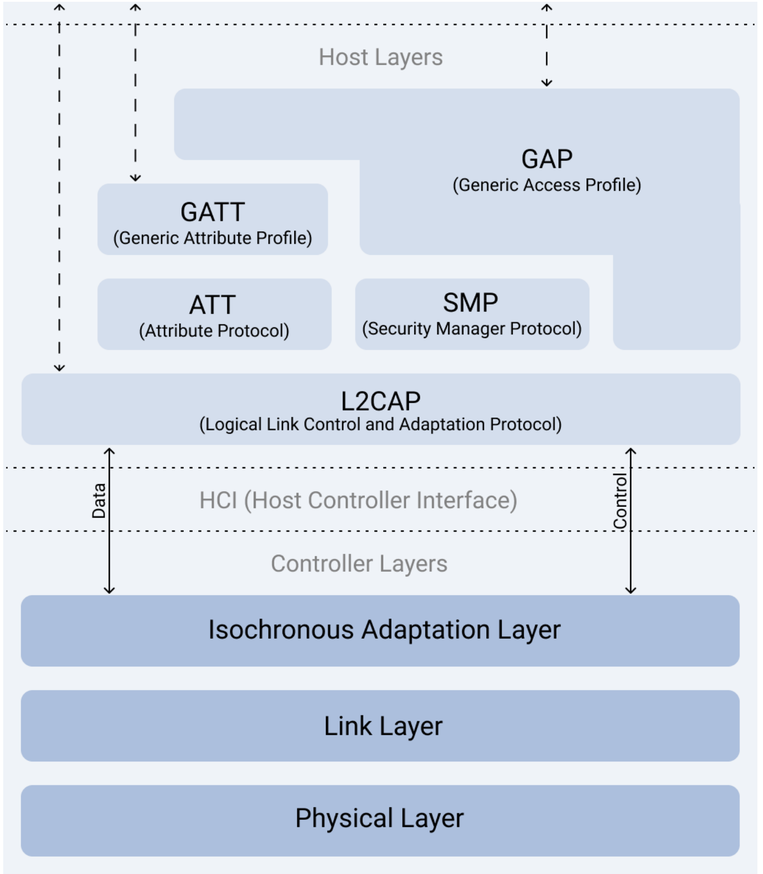
Profiles (based on GATT)

- Heart Rate Profile (HRP) → heart rate & location
- Health Thermometer Profile (HTP) → temperature
- Glucose Profile (GLP) → glucose measurements
- Find Me Profile (FMP) → alerts for lost devices
- Battery Service (BAS) → battery level & charge state
- Device Information Service (DIS) → manufacturer name, model, firmware version, ...
- Current Time Profile (CTS) → time information
- HID (Human Interface Device) over GATT Profile (HOGP) → input & control data (kb, mice, ...)

Generic Access Profile

(GAP)

- Advertisement, discovery, connection & security management
- Discovery is advertisement-based
- Security Manager Protocol (SMP, similar to SSP)
- Roles:
  - Broadcaster / Observer (no connection)
  - Peripheral / Central (connection, PtP or star topology)

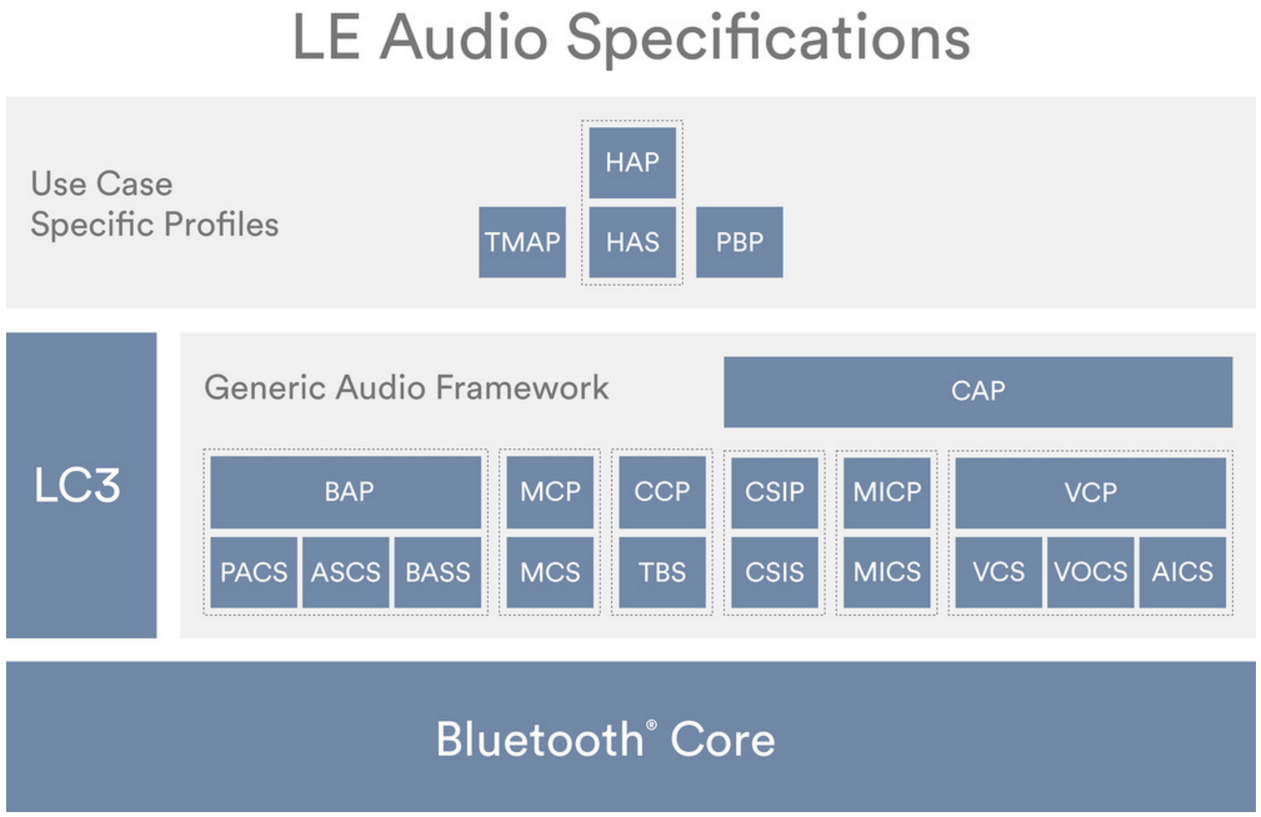


Recent advances in Bluetooth®

LE Audio

The future of wireless audio!

- Introduced in Bluetooth 5.2 (2020)
- Collection of profiles for next-gen audio communication
- Generic audio framework with abstract design
- Combines modern use cases
- Allows for future expansion
- Lower latency (20ms) & enhanced range



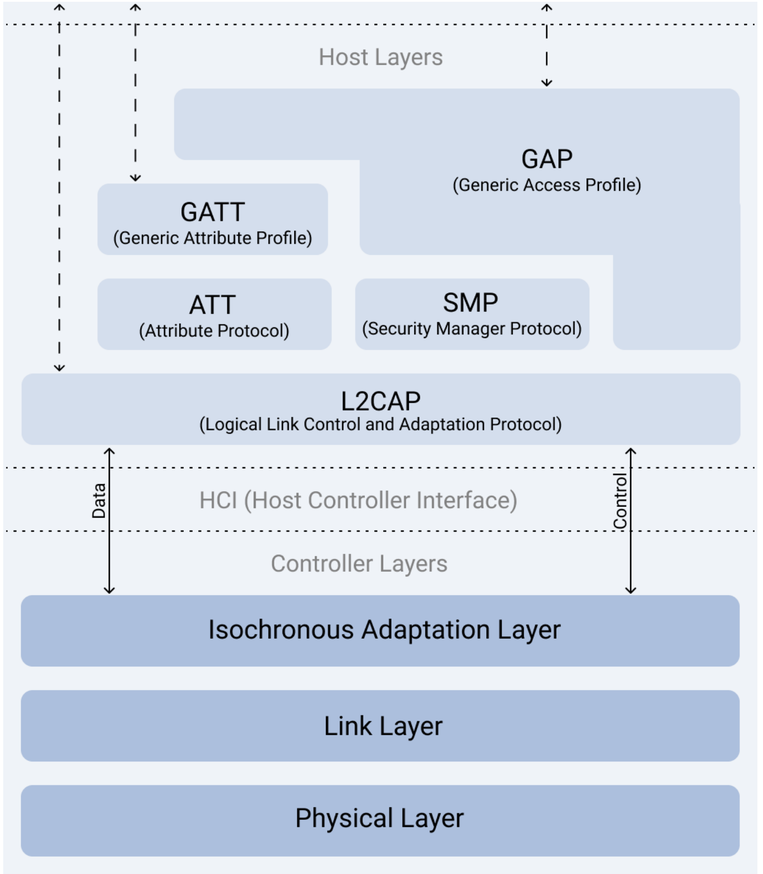
LE Audio

Better codecs

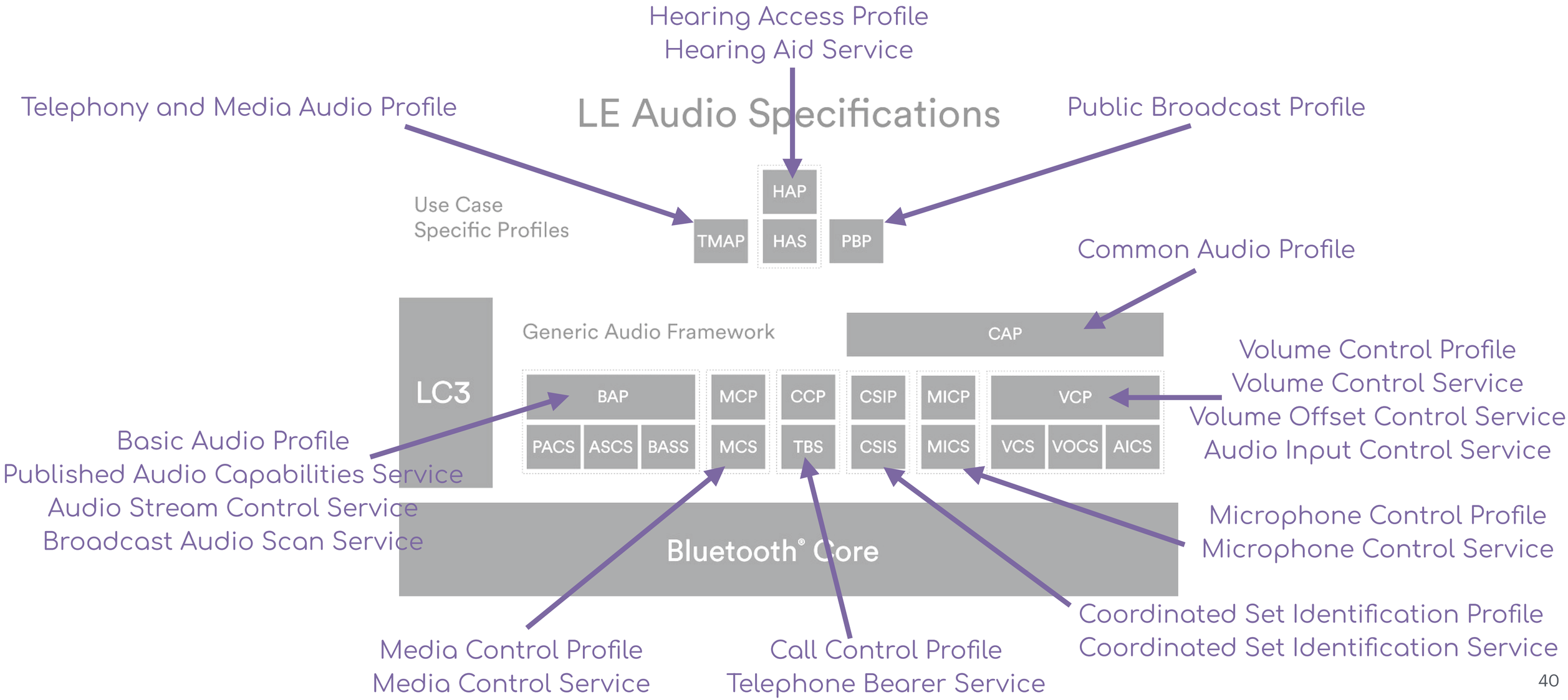
- Mandatory base codec: Low Complexity Communication Codec (LC3)
- Developed by *Fraunhofer IIS* and *Ericsson*
- Higher audio quality with lower bit rates
- Also available now in Classic:
  - Fraunhofer licenses LC3plus High Resolution for A2DP
  - HFP v1.9 allows LC3-SWB for Super Wide Band

Isochronous Adaptation Layer (ISOAL)

- Isochronous channels (ISO)
- Strict timing constraints to maintain synchronisation
- Segmentation & reassembly
- Connected and broadcast streams
- Core component of LE Audio



LE Audio


Hearing Access Profile
Hearing Aid Service
Telephony and Media Audio Profile
LE Audio Specifications
Public Broadcast Profile
Use Case
Specific Profiles
HAP
TMAP
HAS
PBP
Common Audio Profile
Generic Audio Framework
CAP
LC3
BAP
MCP
CCP
CSIP
MICP
VCP
PACS
ASCS
BASS
MCS
TBS
CSIS
MICS
VCS
VOCS
AICS
Volume Control Profile
Volume Control Service
Volume Offset Control Service
Audio Input Control Service
Basic Audio Profile
Published Audio Capabilities Service
Audio Stream Control Service
Broadcast Audio Scan Service
Bluetooth® Core
Microphone Control Profile
Microphone Control Service
Coordinated Set Identification Profile
Coordinated Set Identification Service
Media Control Profile
Media Control Service
Call Control Profile
Telephone Bearer Service

LE Audio

Features

- Unified telephony & media use cases
  - No more switching between A2DP and HFP
  - No more ambiguity or restrictions
  - Modern call control, voice activation, etc (no more 80s tech)
- Hearing Aids are first class citizens
- Multi-stream audio
  - True Wireless Earbuds (!) and other co-ordinated devices (HomePods anyone?)

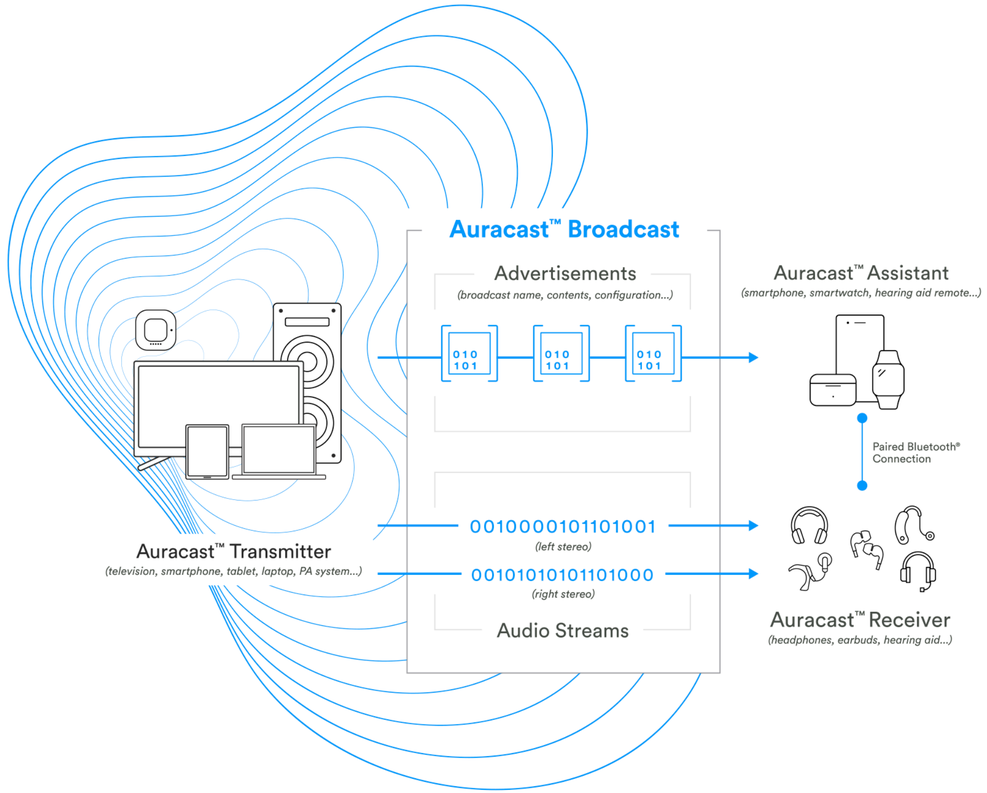
LE Audio

Broadcast

AURACAST™

A BLUETOOTH® TECHNOLOGY

- Marketed as Auracast™
  - “Share your audio”
  - “Unmute your world”
  - “Hear your best”
- Public Broadcast Profile (PBP) allows discovering and joining such sources
  - via phone companion app



Channel Sounding

- Introduced in Bluetooth 6.0 (September 2024)
- Distance awareness between devices (10 - 30cm accuracy)
- Uses advanced Phase-Based Ranging (PBR) technique together with Round-Trip Time (RTT)
- Security features to prevent signal manipulation or replay attacks
- New Physical Layer (PHY) with 72 channels (instead of 40) on the 2.4 GHz band
- Integrated in the Generic Access Profile (GAP)

Channel Sounding

Use cases

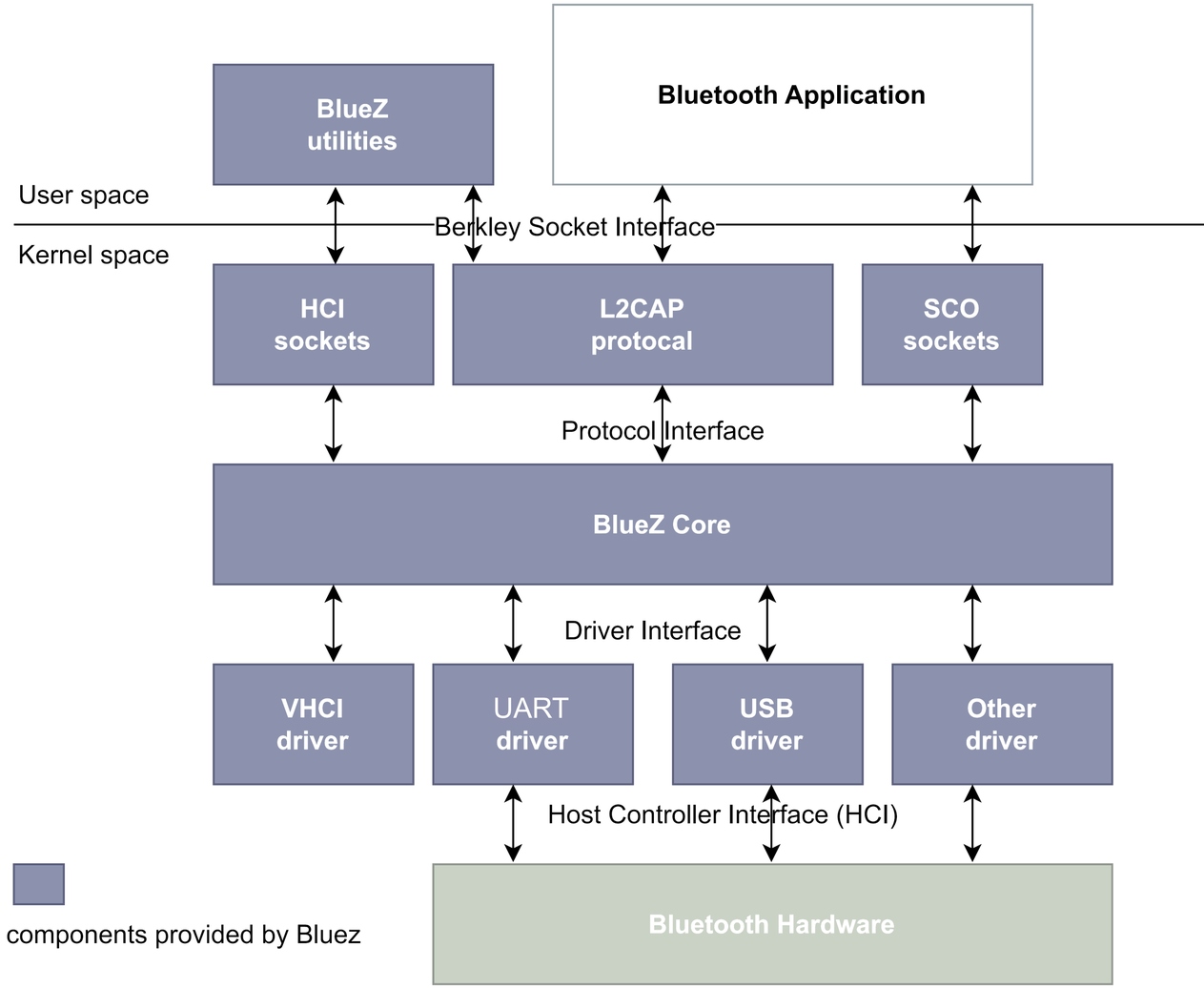
- Smart locks
- Indoor navigation (shopping malls, airports, ...)
- Personal item tracking
- IoT automation based on proximity

We already have all of that, but not with < 30cm accuracy and directional precision ;)

Bluetooth on Linux

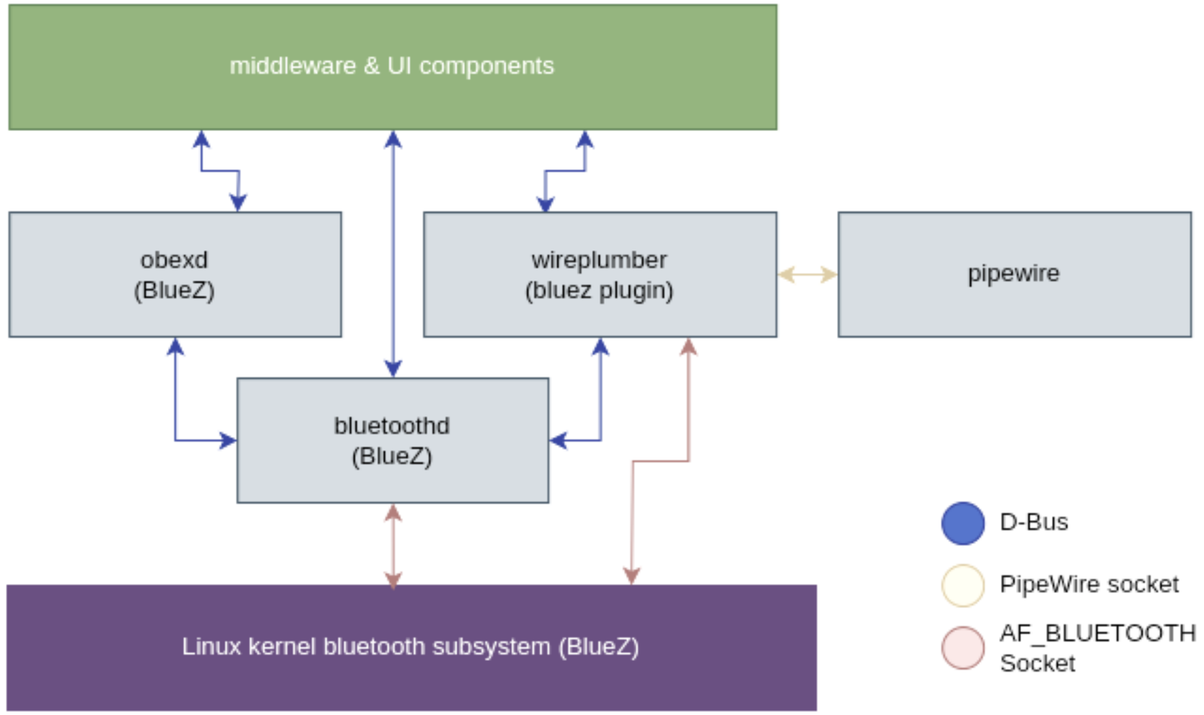
The BlueZ project

- Official Linux Bluetooth protocol stack
- Integrated with the Linux kernel since 2001
- Two components:
  - In-kernel subsystem: device drivers, core protocols
  - Userspace daemon(s) & utilities



Userspace parts

- **bluetoothd** → main manager, handles GAP and other core profiles & protocols
- **obexd** → optional, implements profiles based on the OBEX protocol
- **pipewire / wireplumber** → optional, implements the audio part of A2DP, implements all parts of HSP & HFP, implements BAP
  - alternatives: BlueZ-ALSA, oFono, PulseAudio



Utilities

- **bluetoothctl** → Interactive tool for managing Bluetooth devices
- **obexctl** → OBEX-based file transfers
- **btmon** → Monitors and logs Bluetooth events in detail (HCI packet dump ++)
- **hciattach / btattach** → Attach serial UART devices to the Bluetooth stack
  - `btattach --speed 115200 --device /dev/ttyS0`

```
❯ bluetoothctl
[bluetooth]# Agent registered
[bluetooth]# hci0 new_settings: powered connectable discoverable bondable ssp br/edr le secure-conn wide-band-speech
[bluetooth]# [CHG] Controller 9C:B6:D0:15:D9:18 Pairable: yes
[bluetooth]# AdvertisementMonitor path registered
[bluetooth]# help
Menu main:
Available commands:
-------------------
advertise                                 Advertise Options Submenu
monitor                                   Advertisement Monitor Options Submenu
scan                                      Scan Options Submenu
gatt                                      Generic Attribute Submenu
admin                                     Admin Policy Submenu
player                                    Media Player Submenu
endpoint                                  Media Endpoint Submenu
transport                                 Media Transport Submenu
mgmt                                      Management Submenu
monitor                                   Advertisement Monitor Submenu
assistant                                 Media Assistant Submenu
list                                      List available controllers
show [ctrl]                               Controller information
select <ctrl>                             Select default controller
devices [Paired/Bonded/Trusted/Connected] List available devices, with an optional property as the filter
system-alias <name>                       Set controller alias
reset-alias                               Reset controller alias
power <on/off>                            Set controller power
pairable <on/off>                         Set controller pairable mode
discoverable <on/off>                     Set controller discoverable mode
discoverable-timeout [value]              Set discoverable timeout
agent <on/off/auto/capability>            Enable/disable agent with given capability
default-agent                             Set agent as the default one
advertise <on/off/type>                   Enable/disable advertising with given type
set-alias <alias>                         Set device alias
scan <on/off/bredr/le>                    Scan for devices
info [dev/set]                            Device/Set information
pair [dev]                                Pair with device
cancel-pairing [dev]                      Cancel pairing with device
trust [dev]                               Trust device
untrust [dev]                             Untrust device
block [dev]                               Block device
unblock [dev]                             Unblock device
remove <dev>                              Remove device
connect <dev>                             Connect device
disconnect [dev]                          Disconnect device
menu <name>                               Select submenu
version                                   Display version
quit                                      Quit program
exit                                      Quit program
help                                      Display help about this program
export                                    Print environment variables
script <filename>                         Run script
[bluetooth]#
```

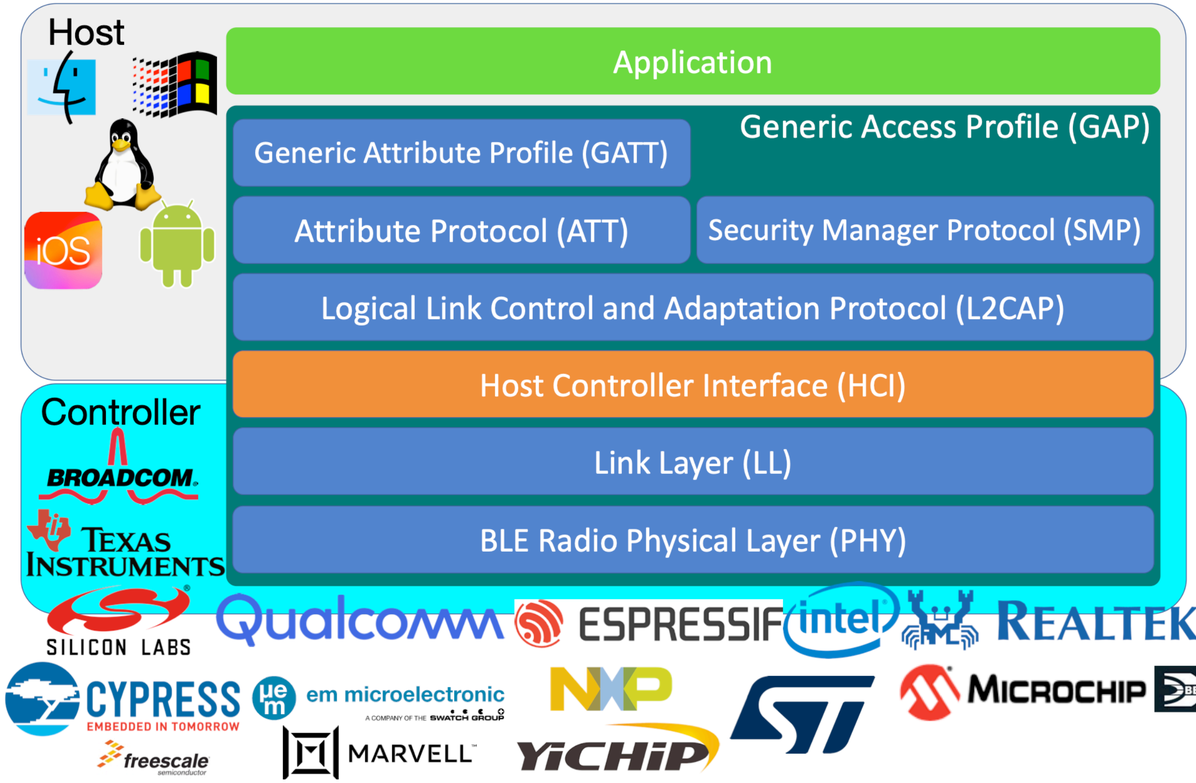
BlueZ vs proprietary stacks

- Embedded controllers are UART (serial) devices
- Proprietary stack → userspace daemon that attaches to serial port and speaks HCI
  - Nothing in kernel space!
  - Good for code reuse with other OSes, bad for security & performance
  - Easier to certify… BlueZ has multiple parts involved, each evolving differently
  - Microkernel-like approach (but Linux is a monolithic OS)

A few more things about HCI

Host <-> Controller

- Command / Reply / Event / Data packets
- Standardised command set
- Vendor-specific command (VSC) set…
  - Write firmware, debug it, access non-standard hardware features
  - Sometimes mistaken as backdoors (see last week…)
  - Can be a PITA or fancy DIY tool



Controller firmware

- Bluetooth controllers typically are not just Bluetooth:
  - also WiFi, Zigbee, Thread, Z-Wave, you name it…
- The same firmware controls all of them
- Some VSCs may affect non-Bluetooth parts
  - intentionally or unintentionally…
  - interesting CVEs do exist

Thank you!

Follow me: https://gkiagia.gr/