COLLABORA

Implementing Bluetooth LE Audio & Auracast on Embedded Linux Systems

George Kiagiadakis
Principal Software Engineer

OSS EU - August 2025

Open First



$ whoami

- Linux Systems Engineer with focus on multimedia
  - More specifically: Audio & Bluetooth
- Principal Software Engineer @ Collabora
- GStreamer, PipeWire & recently BlueZ
- Author & maintainer of WirePlumber
  - PipeWire's default session manager
- Automotive Grade Linux contributor



COLLABORA

Outline

- Bluetooth® Classic audio: history, fundamentals & limitations
- Bluetooth® Low Energy (LE) audio
- Bluetooth® LE Audio on Linux: software stack, configuration, commands, expectations



COLLABORA

Bluetooth® classic audio

Bluetooth® Core vs Profiles

- 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

Bluetooth® audio: origin

- **Bluetooth 1.0** (1999): initial version, no audio
- Idea from the late 80's
  - A wireless replacement for RS-232 (serial) data cables
- Development started in 1994 by Ericsson
  - Focus on mobile phone use cases

Bluetooth® audio: history

- **Bluetooth 1.1** (2001): introduction of HSP & HFP
- Headset & Hands-Free profiles (HSP, HFP)
  - Bi-directional audio for calls / voice data
- Synchronous Connection Oriented (SCO) transport
  - Steady transmission rate with reserved TX time slots / capped bandwidth

Headset / Hands-Free Profiles (HSP, HFP)

	HSP	HFP
Use Case	Headsets (old ones…)	Hands-free in cars / modern headsets
Control Features	**Basic** (answer, hangup, volume ctl)	**Advanced** (dial, hold, 3-way calls, voice activation, …)
Audio Quality	Narrow Band (8 kHz, CVSD codec)	Narrow Band (8 kHz, *CVSD*) Wide Band (16 kHz, ***mSBC***, since v1.6 - **2013**) Super Wide Band (32 kHz, ***LC3-SWB***, since v1.9 - **2023**)

- Control: AT commands over RFCOMM
  - *RFCOMM*: Serial over Bluetooth
  - *Hayes AT command set*: Dial-up modem commands (Hayes Smartmodem, 1981)

Bluetooth® audio: history

- **Bluetooth 1.2** (2003)
  - Advanced Audio Distribution Profile (A2DP)
    - High quality audio streaming – in ***one direction***
  - Extended Synchronous Connections (eSCO)
    - Quality improvements for SCO
- **Bluetooth 2.0** (2004)
  - Audio/Video Remote Control Profile (AVRCP)
    - Remote media player controls

Advanced Audio Distribution Profile (A2DP)

- **Media playback** to headphones/speakers
- Uses AVDTP (*A/V Distribution Transport Protocol*)
  - It's basically RTP (*Real-time Transport Protocol*) over Bluetooth
  - Codecs: SBC (Sub-Band Codec), AAC, aptX, LDAC, Opus, ...
- Controlled via the *Audio/Video Remote Control Profile* (AVRCP)



COLLABORA

Bluetooth® classic audio: issues

- High quality audio (A2DP) is uni-directional only
  - Non-standard solutions exist, few implement them
- Devices switch to HFP when mic is in use
  - Sounds terrible (but also due to speech-optimized DSPs)
- High latency (100-200ms)
- Outdated tech not keeping up to modern use cases
  - Co-ordinated devices (earbuds), spatial audio, audio sharing, voice assistants, hearing aids, much more...

Hearing Aids

- Hearing aids need long battery life & low latency
- Non-standard solutions:
  - Google's Audio Streaming for Hearing Aids (ASHA) – open spec
  - Apple's MFi – proprietary
- Based on Bluetooth Low Energy (LE), but not LE Audio
  - Both developed before LE Audio

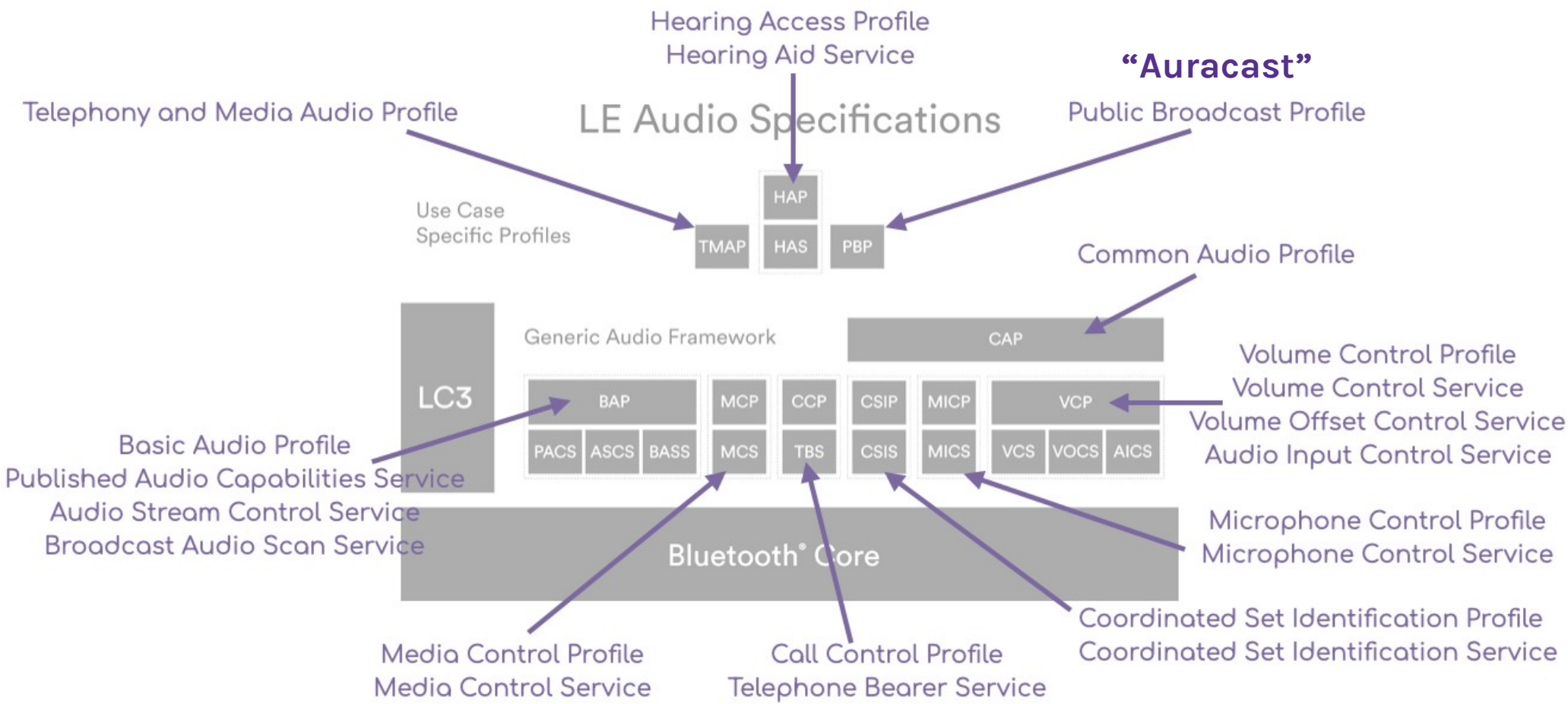


COLLABORA

Bluetooth® LE Audio: the future of wireless audio!

Bluetooth® LE Audio

- **Bluetooth 5.2** (2020), based on Bluetooth LE stack
- Lower power consumption (due to the “Low Energy” stack)
- Lower latency (down to 20ms)
- Collection of profiles for next-gen audio communication
  - Generic audio framework, modular design
  - Combines & unifies all modern use cases, while allowing for expansion


Hearing Access Profile
Hearing Aid Service
"Auracast"
Telephony and Media Audio Profile
LE Audio Specifications
Public Broadcast Profile
HAP
Use Case
Specific Profiles
TMAP
HAS
PBP
Common Audio Profile
Generic Audio Framework
CAP
LC3
BAP
MCP
CCP
CSIP
MICP
VCP
Volume Control Profile
Volume Control Service
Volume Offset Control Service
Audio Input Control Service
PACS
ASCS
BASS
MCS
TBS
CSIS
MICS
VCS
VOCS
AICS
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: Core

- New codec: *Low Complexity Communication Codec* (LC3)
  - Also “backported” to HFP (LC3-SWB) and A2DP (LC3 Plus)
- Transport: Isochronous Streams
  - Time-synchronized audio delivery implemented on the lower layers
  - *Connected Isochronous Streams* (CIS) & *Broadcast Isochronous Streams* (BIS)



COLLABORA

Bluetooth® LE Audio on Linux

Bluetooth® on Linux

- BlueZ: the default Linux Bluetooth “host” stack
  - Kernel drivers
  - Userspace daemons/tools
- PipeWire: multimedia IPC
  - Default **audio server** nowadays across all distributions
  - BlueZ plugin to enable access to Bluetooth audio devices
  - Implements A2DP, HSP, HFP, ASHA, LE MIDI & LE Audio !

LE Audio with PipeWire & BlueZ

- BAP: full support (unicast and broadcast sink & source)
- Also handled: VCP (volume control profile)
- Device sets (ex earbuds) supported via WirePlumber
  - Each device is an independent node, grouped via a property
  - *module-combine-sink* is used to group them
- Broadcast control in WirePlumber config (needs work…)
- See: https://gitlab.freedesktop.org/pipewire/pipewire/-/wikis/LE-Audio-+-LC3-support

Requirements

- Disclaimer: LE Audio & Auracast support are still WIP
- Latest* Linux kernel (bare minimum: 6.4)
- Latest* BlueZ & PipeWire (compiled with liblc3)

* = the later the better, but can be a bit older too...

“stable” is fine

Requirements: adapter

- Bluetooth adapter with LE Audio support
  - Recent Intel (BE200 for Auracast), NXP, MediaTek (also as AMD), Qualcomm, Nordic
  - At least Bluetooth 5.2; Auracast support depends on firmware

```
[bluetoothctl]> menu mgmt
[bluetoothctl]> info
Index list with 1 item
hci0:   Primary controller
        addr [redacted] version 12 manufacturer 29 class 0x4c010c
        supported settings: powered connectable fast-connectable discoverable bondable link-security ss
p br/edr le advertising secure-conn debug-keys privacy configuration static-addr phy-configuration wide
-band-speech cis-central cis-peripheral
        current settings: powered fast-connectable bondable ssp br/edr le secure-conn wide-band-speech
cis-central cis-peripheral iso-broadcaster sync-receiver
        name [redacted]
```

Enable LE Audio

- BlueZ
  - In `/etc/bluetooth/main.conf`:
    - `Experimental = true`
    - `KernelExperimental = ...,<ISO socket UUID>`
- PipeWire
  - In `wireplumber.conf.d/bluetooth.conf` (or similar):
  - Ensure `bluez.roles` in `monitor.bluez.properties` is either undefined (the default) or contains: `[ bap_sink bap_source bap_bcast_sink bap_bcast_source ]`
  - Broadcast needs more config (see `man pipewire-props`)

Enable LE Audio: verify

- If everything has gone well…

```
[bluetoothctl]> show
Controller [redacted] (public)
        Manufacturer: 0x001d (29)
        UUID: Generic Access Profile    (00001800-0000-1000-8000-00805f9b34fb)
        UUID: Handsfree Audio Gateway   (0000111f-0000-1000-8000-00805f9b34fb)
        UUID: Audio Source              (0000110a-0000-1000-8000-00805f9b34fb)
        UUID: Published Audio Capabil.. (00001850-0000-1000-8000-00805f9b34fb)
        UUID: Broadcast Audio Scan      (0000184f-0000-1000-8000-00805f9b34fb)
        ExperimentalFeatures: BlueZ Experimental ISO... (6fbaf188-05e0-496a-9885-d6ddfdb4e03e)
```

Requirements: peripheral(s)

- Peripheral supporting LE Audio (or a 2nd BlueZ/PW machine)
- Remember: earbuds are 2 devices...

```
[Galaxy Buds3 (CC3B)]> info
Device [redacted](public)
        Name: Galaxy Buds3 (CC3B)
        Alias: Galaxy Buds3 (CC3B)
```

```
        UUID: Handsfree                 (0000111e-0000-1000-8000-00805f9b34fb)
        UUID: PnP Information           (00001200-0000-1000-8000-00805f9b34fb)
        UUID: Volume Control            (00001844-0000-1000-8000-00805f9b34fb)
        UUID: Microphone Control        (0000184d-0000-1000-8000-00805f9b34fb)
        UUID: Audio Stream Control      (0000184e-0000-1000-8000-00805f9b34fb)
        UUID: Broadcast Audio Scan      (0000184f-0000-1000-8000-00805f9b34fb)
        UUID: Published Audio Capabil.. (00001850-0000-1000-8000-00805f9b34fb)
        UUID: Common Audio              (00001853-0000-1000-8000-00805f9b34fb)
        UUID: Telephony and Media Aud.. (00002b51-0000-1000-8000-00805f9b34fb)
```

Connect LE Audio instead of A2DP/HFP

- Preferred bearer: select stack per device
  - `bearer <device_addr> le / bredr / last-used`
  - Experimental D-Bus API and explicit bluetoothctl info available on master (BlueZ 5.84)
- UX issue: not a simple audio device profile change

```
> info XX:XX:XX:XX:XX:XX
...
BREDR.Paired: yes
BREDR.Bonded: yes
BREDR.Connected: no
LE.Paired: yes
LE.Bonded: yes
LE.Connected: no
...
[CHG] BREDR XX:XX:XX:XX:XX:XX Connected: yes
```



COLLABORA

WIP / TODO

- Improve general LE Audio & Auracast usability (tools, UX, etc)
- Desktop UI integration
  - Auracast control, LE Audio switching, Call/Telephony UI
- Move HFP control to BlueZ & impl Call Control Profile (CCP)
  - Common abstraction, easier use & qualification
- There’s more coming to the LE Audio spec soon… !

Community

- BlueZ
  - Slack: go to https://www.bluez.org/contact/ -> click on the slack invitation
  - Mailing list: linux-bluetooth@vger.kernel.org (patches & discussions)
  - GitHub: https://github.com/bluez (code, CI & tickets)
- PipeWire
  - Matrix: #pipewire:matrix.org
  - GitLab: https://gitlab.freedesktop.org/pipewire/
  - Mailing list is unused; file tickets instead



Thank you!