How to Design Auracast™ Earbuds

This document covers the basic design principles for building Auracast™ earbuds and other receiver devices, explaining their features, their configuration requirements, and how they interact with other Auracast™ devices. These principles also apply to speakers, hearing aids, headphones, and soundbars. The focus on earbuds illustrates the additional design issues encountered when two independent Auracast™ receivers need to synchronize to the same Auracast™ transmitter.

Author: Market Development

Version: 1.0

Revision Date: 14 May 2024

Table of Contents

1.0 Purpose of This Document

This document takes designers through the process of designing a pair of Auracast™ earbuds, describing available options and the considerations that affect the user experience. These principles apply to any Broadcast Sink, whether that's an earbud, speaker, hearing aid, headphone, or soundbar, but earbuds illustrate the additional design issues encountered when two independent devices need to synchronize to the same Broadcast Source.

This discussion is confined to the broadcast aspects of the Bluetooth® LE Audio specifications. Although the majority of products on the market will support the reception of both unicast and broadcast streams, broadcast introduces a number of new concepts which some developers may find challenging. By walking through the design process, the aim is to help clarify the design decisions that need to occur. As we will only deal with the broadcast aspects, we can ignore the presence of any microphones in these products.

1.1 Nomenclature

The Bluetooth LE Audio specifications each introduce a variety of different roles taken by the three devices that comprise the Auracast™ ecosystem. As using all of these can be confusing when providing an overview, this document introduces and uses the generic descriptive names of Auracast™ transmitter, Auracast™ receiver, and Auracast™ assistant. These are used to describe both physical devices and their roles. They are based on the definitions of roles in the Basic Audio Profile but also encompass the roles identified in various higher level specifications which are listed in Table 1.1, which are the Basic Audio Profile (BAP) [1], the Public Broadcast Profile (PBP) [2], the Common Audio Profile (CAP) [3], the Telephony and Media Audio Profile (TMAP) [4], and the Hearing Access Profile (HAP) [5]. Note that roles are always defined in profile documents, even when they apply to the device acting as the server.

Name	Includes the Role of	Specification
Auracast™ Transmitter	Broadcast Source Public Broadcast Source Initiator Broadcast Media Sender	BAP Section 2.2.2.1 PBP Section 3.1 CAP Section 2.1.1 TMAP Section 3.5.2
Auracast™ Receiver	Broadcast Sink Public Broadcast Sink Acceptor Hearing Aid Broadcast Media Receiver	BAP Section 2.2.2.2 PBP Section 3.2 CAP Section 2.1.2 HAP Section 3.2 TMAP Section 3.5.2
Auracast™ Assistant	Broadcast Assistant Public Broadcast Assistant Commander	BAP Section 2.2.2.3 PBP Section 3.3 CAP Section 2.1.3

Table 1.1: Underlying specification roles covered by the Auracast™ terminology in this document

1.2 Audience

This document assumes that readers have a basic knowledge of Bluetooth® Low Energy (LE) and are familiar with concepts like GATT, characteristics, and client/server architectures. A basic understanding of Bluetooth LE Audio and isochronous channels is also assumed. An overview of the Auracast™ ecosystem is available in the companion document An Overview of Auracast™ Broadcast Audio [6].

1.3 The Building Blocks of an Auracast™ Receiver

The task of an Auracast™ receiver is fairly simple. It needs to find an Auracast™ transmission, then synchronize to it and render it. As Auracast™ only involves broadcast, we can ignore any microphone in the receiver. That means there are a very limited number of functions which an earbud needs to support. They are:

- Scanning for Auracast™ transmitters
- Optionally delegating the scanning function to an Auracast™ assistant
- Selecting an Auracast™ audio stream to acquire, then synchronizing to it
- Rendering that stream at the correct point in time
- Managing reception (terminating or changing it)
- Notifying any changes to support the other earbud in its coordinated set

These functions are illustrated in Figure 1.1. In the case of pairs of Auracast™ receivers, these functions will exist in both along with support for coordinated sets.

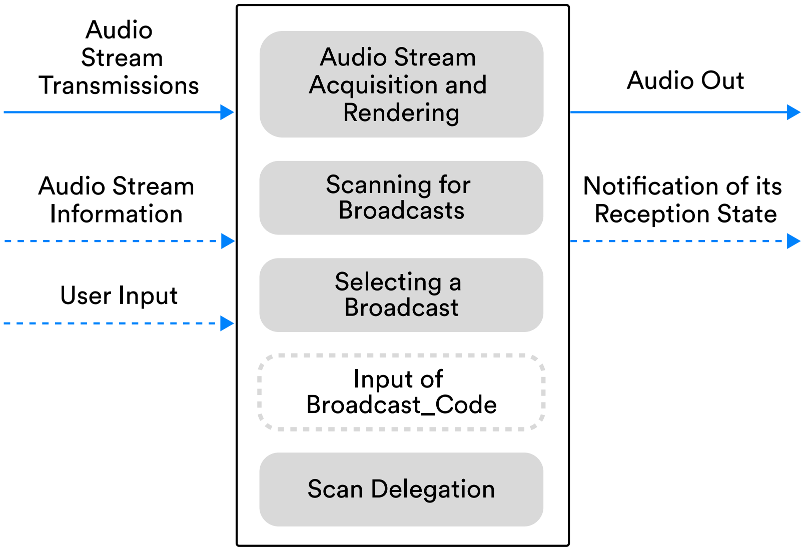


Figure 1.1: The main functions of an Auracast™ receiver

An Auracast™ receiver can perform these functions autonomously without making a connection to any other device. If it is paired with an Auracast™ assistant, which acts as a GATT client, it can receive and act on Control Point writes, which can provide a richer user experience. Otherwise, it only requests or exposes information through notifications or advertisements.

There is no state machine for an Auracast™ receiver. It acts autonomously or enlists the help of an Auracast™ assistant to discover, select, and synchronize to one of potentially many Auracast™ transmitters.

An Auracast™ receiver requires only a limited number of Bluetooth LE Audio specifications to support the broadcast capability, which are.shown in Figure 1.2. In many cases, an earbud will also include include other specifications to support unicast functionality, such as microphones and bidirectional audio streams, as well as media and telephony control. But it is possible to build Auracast™ compliant public address (PA) speakers using just the specifications shown below.

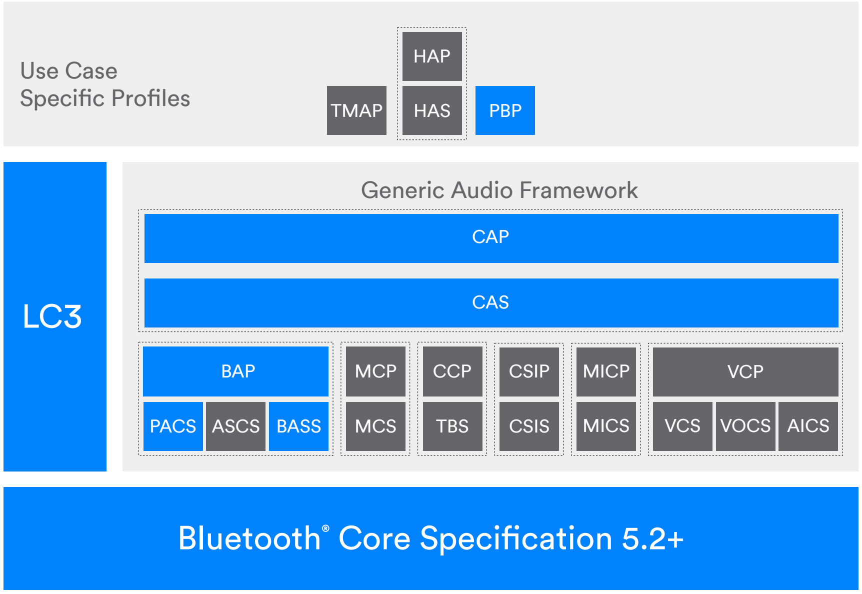


Figure 1.2: The Bluetooth LE Audio specifications required for Auracast™ devices

Every Auracast™ receiver needs to include:

- A controller which supports the Bluetooth® Core Specification at version 5.2 or above, as well as an implementation of the LC3 codec [6]
- BAP – the Basic Audio Profile, which specifies the main requirements of Bluetooth LE Audio for broadcast
- PACS – the Published Audio Capabilities Service, which is essentially a database that exposes what the receiver (earbuds, speakers, etc.) can do
- BASS – the Broadcast Audio Scan Service [8], which defines how Auracast™ receivers can work with Auracast™ assistants to discover and select broadcast streams

- CAP – the Common Audio Profile adds additional features and is necessary where devices come in sets, such as earbuds, hearing aids, and speakers
- PBP - the Public Broadcast Profile, which defines configurations for enhanced interoperability

Many of the other Bluetooth LE Audio specifications can be implemented to add extra functionality, but we'll be concentrating on BAP and BASS, where the important decisions are made.

In many cases, earbuds will be designed to support both Bluetooth Classic and Bluetooth LE Audio, so that they can operate with both legacy and Bluetooth LE Audio audio sources. This document confines itself to the Bluetooth LE Audio broadcast functionality.

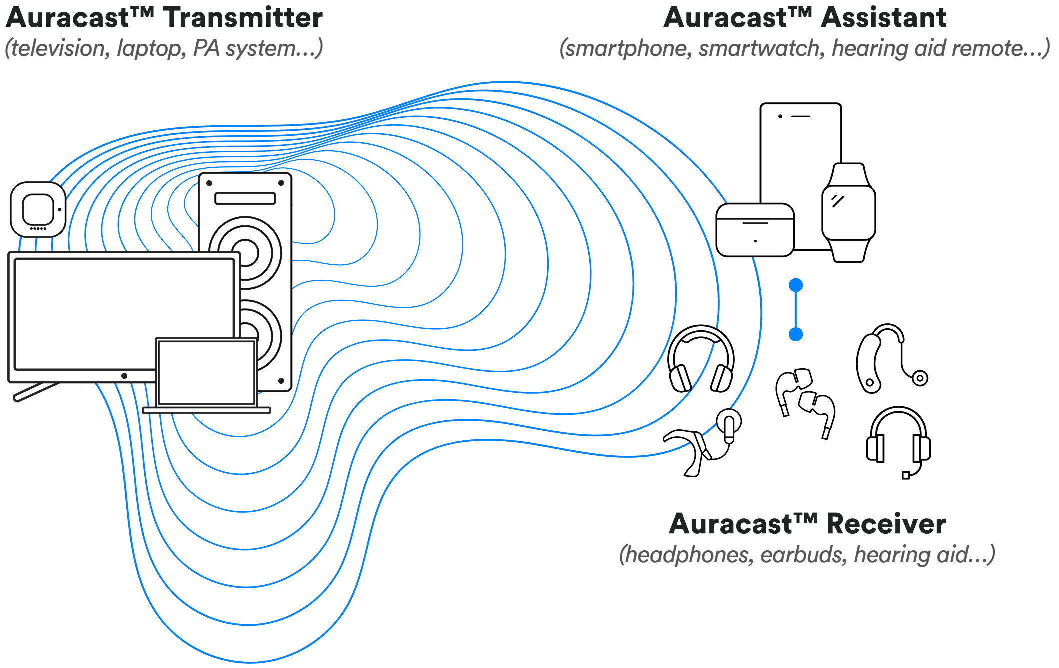

Auracast™ Transmitter
(television, laptop, PA system...)
Auracast™ Assistant
(smartphone, smartwatch, hearing aid remote...)
Auracast™ Receiver
(headphones, earbuds, hearing aid...)

Figure 1.3 The basic topology of Auracast™ broadcast audio

1.4 An Overview of the Auracast™ Topology

Before we start on the design of earbuds for Auracast™ broadcast audio, it is important to understand the Auracast™ topology as the topology of broadcast is totally different from that of any other Bluetooth audio connection. Figure 1.3 illustrates a typical configuration.

The Auracast™ transmitter operates autonomously, with no idea of whether any other device is listening to it. Any number of Auracast™ receivers can listen to it, but the transmitter never knows whether any of them are.

Auracast™ receivers are the devices which receive and render the transmissions from an Auracast™ transmitter. They comprise of any device which can produce sound, such as earbuds, headphones, sound bars, speakers, and headphones. Where multiple Auracast™ receivers are designed to work together, such as earbuds of speakers, they each receive the specific audio stream for their location, such as left and right, along with synchronization information which enables them to render the different audio channels at exactly the same time.

If more than one Auracast™ transmitter is within range of an Auracast™ receiver, then the receiver needs to decide which Auracast™ transmitter to listen to. An Auracast™ receiver can do that itself, but that's often not a great user experience, as earbuds and hearing aids have a very limited user interface available to let the user make that choice. The Broadcast Assistant role was defined to address this and forms the basis of every Auracast™ assistant.

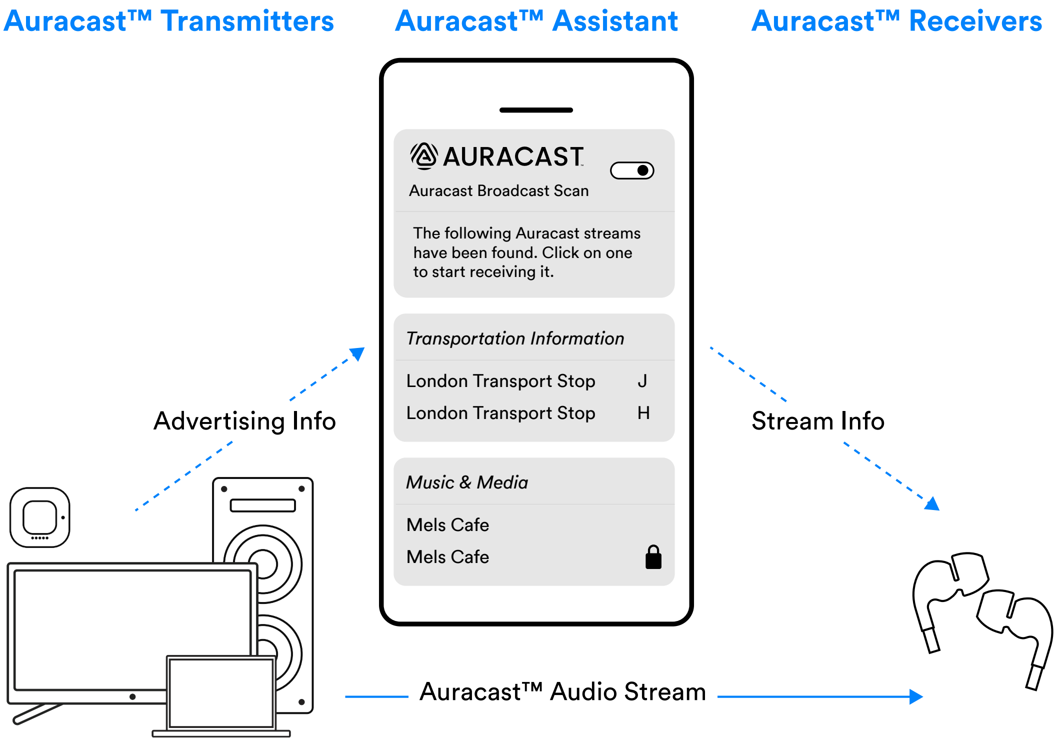
Auracast™ assistants provide a remote user interface to make that choice easier by scanning for Auracast™ transmitters and providing a method for the user to choose which broadcast to listen to. They allow an Auracast™ receiver to offload its scanning function, relying on the Auracast™ assistant to do the work for it. Once the Auracast™ assistant has built up a list of available Auracast™ transmitters, it lets the user choose which one to listen to and instructs the receiver (e.g., earbuds, hearing aids, speakers, etc.) to start receiving and rendering the appropriate broadcast streams.

There are three important points to note in this diagram:

- There is no paired connection between any of the Auracast™ transmitters and the receivers
- There is no paired connection between any of the Auracast™ transmitters and the Auracast™ assistants
- There is no Bluetooth® connection required between the left and right receivers within a coordinated set, like earbuds

This last point surprises many developers, but there is a practical reason for this. The head absorbs the 2.4 GHz transmissions of the Bluetooth® signal extremely well. The result is that for many applications, particularly with miniature earbuds with small antennas, it is nearly impossible for a Bluetooth signal to get reliably from your left to right ear, particularly if you are outside where there are no radio reflections from walls or ceiling. While many earbuds and hearing aids today contain a proprietary sub-GHz radio to connect the left and right earbuds, the Bluetooth standard does not specify or expect this. Instead, the Bluetooth LE Audio specification provides other mechanisms to ensure that two earbuds remain synchronized, principally through the use of notifications and support for the coordinated set features within Auracast™ assistants.

At the point where a pair of earbuds select and synchronize to a particular Auracast™ transmitter, the audio they receive over the Broadcast Isochronous Streams is received directly from the Auracast™ transmitter, as shown in Figure 1.4. No ACL connection is made with the Auracast™ transmitter.



The following Auracast streams have been found. Click on one to start receiving it.

Figure 1.4: Data flow in Bluetooth LE Audio broadcast

The phone shown in Figure 1.4 can also be used for controlling volume and other features of the earbuds, but that is independent of broadcast.

1.4.1 The Importance of Notifications

As the pair of earbuds (or hearing aids) have no connection between them, notifications become extremely important. They are used to ensure all Auracast™ assistants are kept up to date with the status of both of the earbuds. This is the mechanism that ensures a pair of earbuds receive streams from the same Auracast™ transmitter. Each earbud maintains its state using a set of characteristics. When a change is made, either by the action of an Auracast™ assistant writing to a control point or by an autonomous action on the earbud, that change is notified to all Auracast™ assistants which will use their knowledge of the devices in the coordinated set to update the information on the other earbud, which will, in turn, notify its new, updated state. Auracast™ assistants need to maintain a record of the current values for the devices on whose behalf they are scanning to prevent multiple notifications from being sent.

The process is shown in Figure 1.5. An event at the right Auracast™ receiver results in a notification being produced. This is received by the Auracast™ assistant, which then writes to the other member of the coordinated set to keep its value synchronized.

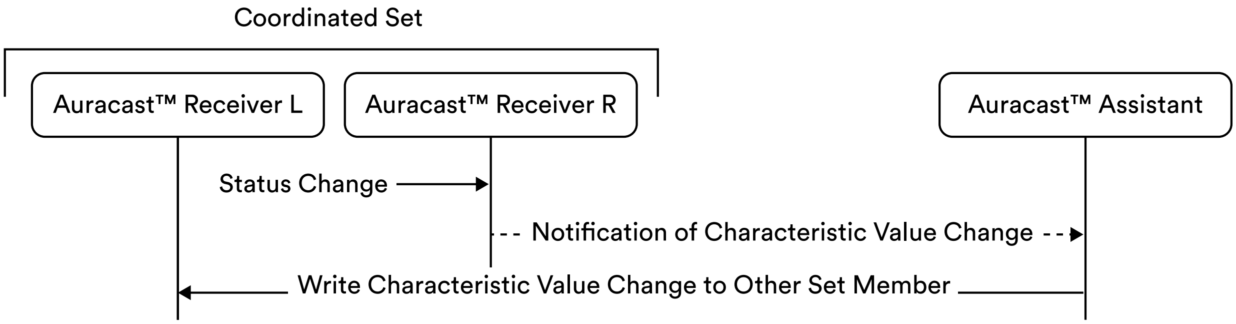

Coordinated Set
Auracast™ Receiver L
Auracast™ Receiver R
Auracast™ Assistant
Status Change
Notification of Characteristic Value Change
Write Characteristic Value Change to Other Set Member

Figure 1.5: The basic principle of using notifications in coordinated sets

A consequence of this approach is that if a pair of earbuds do not individually have an active ACL connection to the same Auracast™ assistant, then any change made on one of them will not be reflected on the other. If a manufacturer expects a pair of earbuds to work together without the presence of an Auracast™ assistant, they need to include a proprietary mechanism to allow the two devices to keep track of each other's state, normally through the provision of a sub-GHz radio link. Where a proprietary link between the earbuds exists, each earbud should still notify any change, which will be interpreted by an Auracast™ assistant as an autonomous action.

Notifications are also used for other control functions, ensuring that changes are applied at the same time to both earbuds. When earbuds are using more than one Auracast™ assistant, this ensures that all Auracast™ assistants are aware of the latest state of both earbuds.

1.5 Broadcast Characteristics

The Bluetooth® features of an Auracast™ receiver are controlled through a number of GATT characteristics which are used by Auracast™ assistants to perform the scanning and selection operation and to keep the pair of earbuds synchronized.

For an Auracast™ earbud operating without an Auracast™ assistant, there is no connection to any device to implement the normal Client-Server relationship that characteristics need. That means that the Auracast™ receiver operates autonomously. However, it still needs to implement these characteristics for two reasons:

- Without them, it is incompatible with Auracast™ assistants, which is not allowed
- The Broadcast Receive State characteristic is the foundation of an Auracast™ receiver's database of known Auracast™ transmitters

Table 1.2 lists all of the characteristics required for an Auracast™ receiver. Not all of the characteristics are used for broadcast. The two most important characteristics are the Broadcast Receive State characteristic, which can be used as the basis for building a database of known Auracast™ transmitters, and the Broadcast Audio Scan Control Point characteristic, which Auracast™ assistants use to manage their Auracast™ receivers.

back to contents

Characteristic	Defined in	Req'd	Used for Broadcast	Instances	Properties	
					Mandatory	Optional
Broadcast Audio Scan Control Point	BASS	M	Y	1	Write, Write w/o response	None
Broadcast Receive State	BASS	M	Y	varies [1]	Read, Notify	None
Sink PAC	PACS	M	Y	varies [2]	Read	Notify
Sink Audio Locations	PACS	O	Y	1 [2,3]	Read	Notify, Write
Available Audio Contexts	PACS	M	N [5]	1	Read, Notify	None
Supported Audio Contexts	PACS	M	N [5]	1	Read, Notify	None
Coordinated Set Size	CSIS [9]	O	Y	1 [2,4]	Read	Notify
Set Identity Resolving Key	CSIS	M	Y	1 [2,4]	Read	Notify

[1] If an earbud can synchronize to more than one BIG, the minimum value represents the maximum number of simultaneous BIGs that it can support. However, to enable support for rich user interfaces on Auracast™ assistants and other non-scanning selection applications, it is suggested that a minimum of ten instances of the Broadcast Receive State are provided, even when only one BIG is supported.

[2] These characteristics are used for both unicast and broadcast applications.

[3] In most cases, this should be included for an Auracast™ receiver.

[4] Should be included for an Auracast™ receiver which is a member of a coordinated set.

[5] Although it is mandatory to support this characteristic, it is only used for unicast applications.

Table 1.2: Characteristics required to support a pair of earbuds configured as a coordinated set.

2.0 Finding Auracast™ Transmitters

The decision to find an Auracast™ transmitter usually begins with a user action. This may be by pressing a button on an earbud or performing an action on an Auracast™ assistant. The earbuds may be in an idle state where they have no active Bluetooth® connection, which is common for hearing aids – they will only look for a broadcast when they need it. Alternatively, they may have a Bluetooth connection to an Auracast™ assistant, but no current audio stream (either BR/EDR or Bluetooth LE Audio), or they may be engaged in another Bluetooth audio application which could be HFP, A2DP, Bluetooth LE Audio unicast, or broadcast.

Whatever their state, an action by the user instructs an earbud or its Auracast™ assistant to start scanning to find an Auracast™ transmitter. If an Auracast™ assistant is not available, only one of a pair of earbuds needs to scan, as it will use the notification process (or a proprietary link) to inform the other of what it has discovered.

2.1 How to Discover Auracast™ Transmitters

Before an Auracast™ receiver can render a broadcast stream, it needs to find it. It does this by scanning. All Auracast™ receivers must be able to scan, although they will normally delegate this task to one or more Auracast™ assistants. Having discovered one or more relevant broadcast audio streams, they need to decide which, if any, of these streams they wish to listen to.

Auracast™ transmitters use extended advertisements to publish this information. Figure 2.1 illustrates the structure of these advertisements and the key packets of data within them.

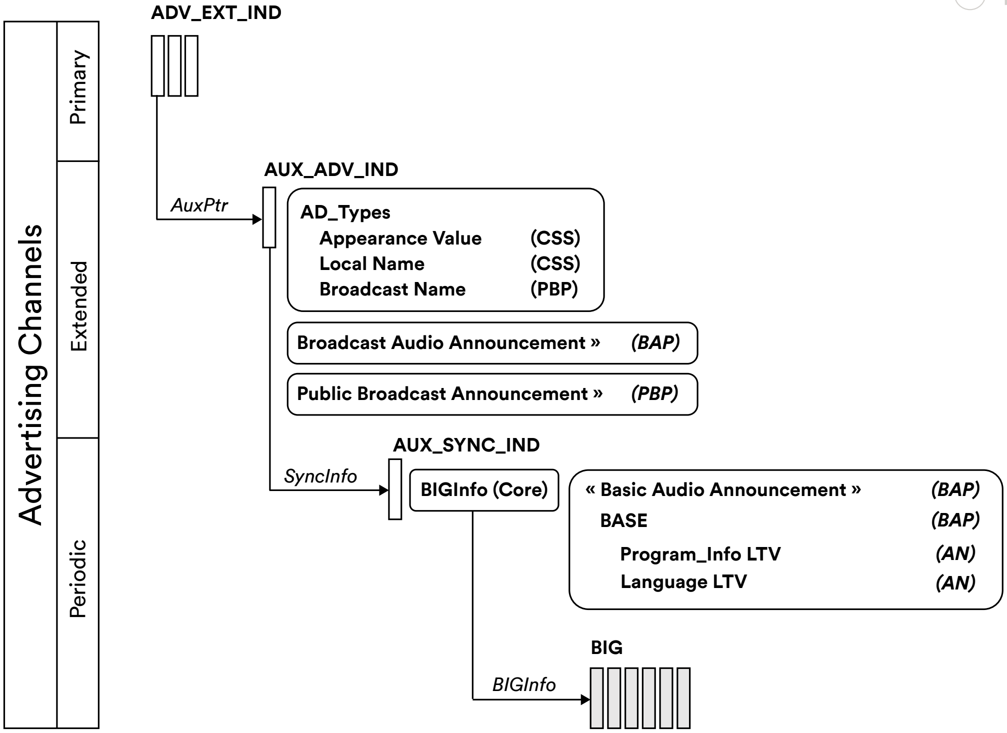


Figure 2.1: The structure of Auracast™ information from Auracast™ transmitters[1]

2.2 The Scanning Process

A device scanning for Auracast™ transmissions needs to work its way through the various elements of these advertisements, collecting the data it will use to populate each instance of the Broadcast Receive State characteristic. At each step, it should parse the contents to decide which broadcast streams are relevant and which can be ignored. How many streams are discovered before a decision is made to select one to listen to is down to the specific implementation. This may be set by a policy on the earbud or the Auracast™ assistant. For example, an earbud may be configured to automatically select a known broadcast as soon as it is discovered. Alternatively, it may wait until a manual selection is made on an Auracast™ assistant. This is a user interface design decision which may be configurable by the user.

Figure 2.2 illustrates the basic decision flow during scanning as a scanning device works its way through the information contained in a hierarchy of advertisements. Although an Auracast™ receiver may delegate this task to an Auracast™ assistant, it also needs to be able to perform this process autonomously. The User Action in Figure 2.2 is an action on the Auracast™ receiver or Auracast™ assistant to start scanning for transmitters. The decision to perform that autonomously, to delegate the task, or to perform both in parallel is down to the Auracast™ receiver implementation.

[1] *CSS = Core Specification Supplement, PBP = Public Broadcast Profile, BAP = Basic Audio Profile, and AN – Assigned Numbers.*

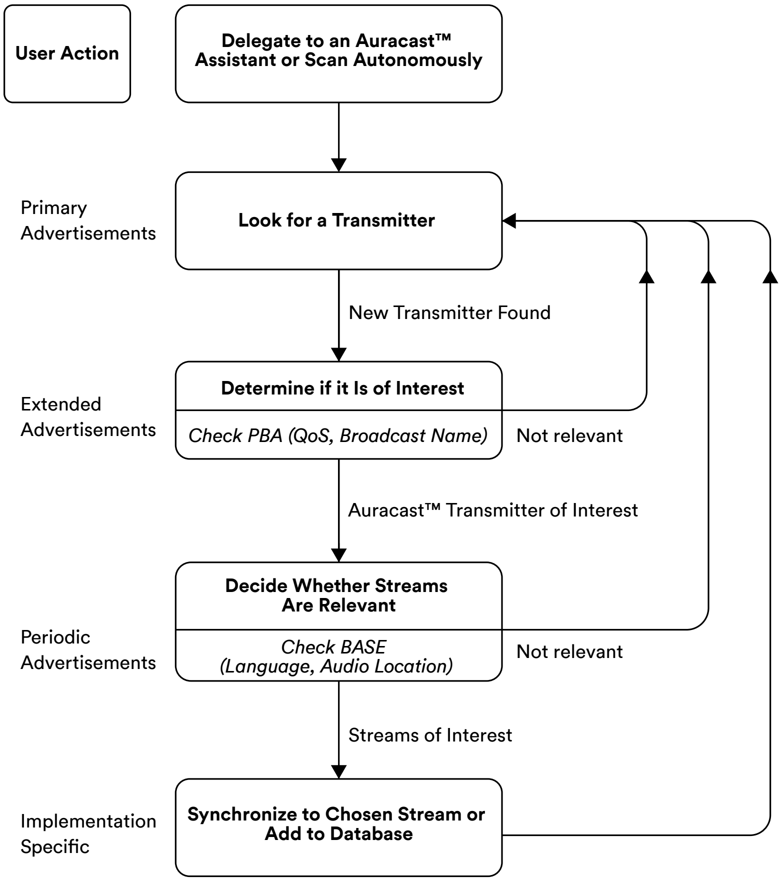

User Action
Delegate to an Auracast™ Assistant or Scan Autonomously
Primary Advertisements
Look for a Transmitter
New Transmitter Found
Extended Advertisements
Determine if it Is of Interest
Check PBA (QoS, Broadcast Name)
Not relevant
Auracast™ Transmitter of Interest
Periodic Advertisements
Decide Whether Streams Are Relevant
Check BASE (Language, Audio Location)
Not relevant
Streams of Interest
Implementation Specific
Synchronize to Chosen Stream or Add to Database

Figure 2.2: Basic scanning process

2.3 Parsing the Extended Advertisements

Scanners will start by looking for primary advertisements which contain an Auxiliary Pointer (AuxPtr), signifying the existence of extended advertising data. If the AuxPtr is not present, it is not a broadcaster, so can be ignored. The scanner will then look for the Extended Advertisement associated with that AuxPtr to see if it includes a Broadcast Audio Announcement, signifying that it is a Broadcast Source. If it is present, it should look for and parse the Public Broadcast Announcement (defined in the Public Broadcast Profile). This will always be present if at least one of the streams within the BIG is Auracast™ compliant[2]. The Public Broadcast Announcement provides some high-level compatibility information.

- Whether the stream is encrypted, in which case the user needs to obtain a Broadcast_Code, and
- Whether a 16kHz or 24kHz stream is currently being transmitted[3]

[2] *If the Broadcast Source does not claim Auracast™ compliance, the Public Broadcast Announcement may not be present, in which case a scanner would need to retrieve the BASE data from the Periodic Advertisement to determine whether it is capable of decoding the broadcast audio streams.*

[3] *If the Public Broadcast Announcement does not show that a 16kHz or 24kHz stream has been configured, the user may need to ask for it to be enabled. All Auracast™ compliant transmitters must support this functionality.*

The scanning device should also retrieve the Broadcast_Name of the transmitter. The Broadcast_ Name is a human-readable name for the stream, which is also used where there is a physical display of the stream information. The scanning device should check for any metadata LTV structures in the Public Broadcast Announcement, as these may contain high-level information about the contents of the streams or how they should be rendered.

Based on this information, the scanning device should decide whether or not to retrieve further information about the audio streams by synchronizing to the periodic advertisements using the SyncInfo information included in the extended advertisement.

2.4 Parsing the Periodic Advertisements

Once it has obtained the periodic advertisement, the scanning device needs to acquire the Basic Audio Announcement, which provides detailed information about the individual BISes in each BIG. These are described in the Broadcast Audio Stream Endpoint (BASE) structure in Section 3.7.2.2 of BAP [1].

- The information in the BASE can be split into three categories:
- Content information, which helps indicate whether the stream is relevant, such as its language
- Practical information, including encoding and rendering details
- The arrangement of the BISes within the BIG, including their intended audio locations, (e.g., left, right, and mono)

The BASE can also contain metadata which provides a scanning device with more information about the streams and how they should be treated.

Figure 2.3 is a simplified representation of a BASE structure, showing an example of a BIG which contains two subgroups of stereo streams, one in English and one in Spanish.

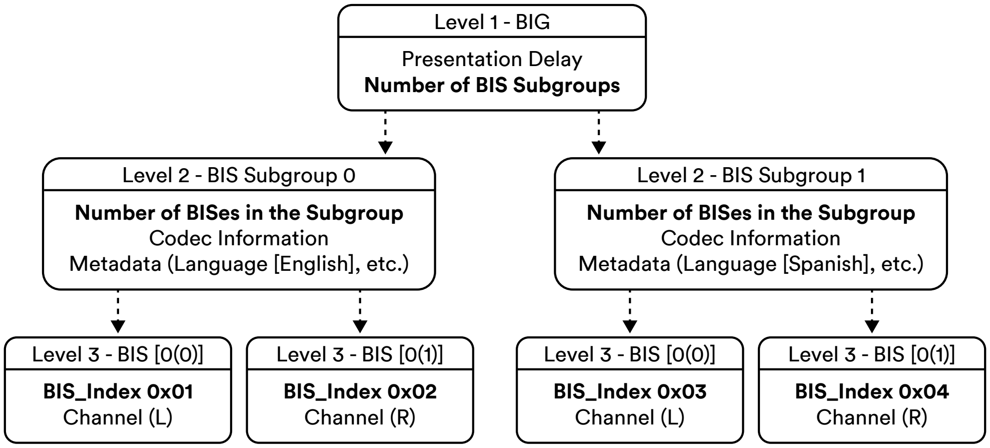


Figure 2.3: A simplified example of the BASE structure and contents

Having collected this information, the scanning device may decide that it is not interested in any of the streams, in which case it can move on and look for more transmitters, or for further BIGs from the same transmitter. If it is interested, then it should store the information before making a decision about synchronizing and receiving one or more of the streams. This will be autonomous if it is the Auracast™ receiver scanning, or as a result of an operation on the Broadcast Audio Scan Control Point if done by an Auracast™ assistant. Storing the information in an instance of the Broadcast Receive State characteristic will trigger a notification that can be used to update all the Auracast™ assistants which are currently scanning on behalf of the Auracast™ receiver.

2.5 The Auracast™ Receiver's Database of Auracast™ Transmitters

The Bluetooth® LE Audio specifications make no mention of what an Auracast™ receiver does with the information it discovers about Auracast™ transmitters within range. They may have been discovered by the earbud, or by an Auracast™ assistant acting on its behalf, which sends information about them to the earbud. It is possible, and allowable, to qualify an earbud which only stores information on the most recent Auracast™ transmitter it has found or been told about. That might work reliably in the early days of Auracast™ deployments when a user will rarely find more than one transmitter, but it is not a scalable experience for a user, particularly if they want to swap between different transmitters, as it would mean that every change to a new transmitter would require a new scan.

It is clear that to cope with growing numbers of Auracast™ transmitters, Auracast™ receivers will need to maintain a database of known transmitters within range, updating it whenever the user requires. The BASS specification contains the features to implement this, using multiple instances of the Broadcast Receive State characteristic. These reside within the Scan Delegator role which is part of every Auracast™ receiver. It is the role that is responsible for interacting with Auracast™ assistants, using the Broadcast Audio Scan Control Point characteristic.

There's a lot of data which is included in each Broadcast Receive State characteristic. Basically, it contains:

- A unique identifier (the Source_ID)4 for each BIG assigned by the Auracast™ receiver
- Information on how to find that BIG
- Whether the Auracast™ receiver is currently synchronized to it, and, if so, to which BIS(es)
- Whether the audio streams are encrypted and the Auracast™ receiver has the correct code to decrypt them
- Any metadata which may be relevant in deciding whether to synchronize to the stream

Table 2.1 shows the main items which are important to understand. The full list is in Table 3.9 of BASS and all of these features are mandatory.

[4] *The Source_ID is unique to each Auracast™ receiver, so it may be different in the Broadcast Receive State characteristics of left and right earbuds. It is not the same parameter as the BIG_ID which identifies individual BIGs on an Auracast™ transmitter.*

Field	Description
Source_ID[1]	A local, unique reference for each Broadcast Isochronous Group (BIG)[2] generated locally by the Auracast™ receiver. There will be a different instance of the Broadcast Receive State characteristic for each Auracast™ transmitter that the Auracast™ receiver is aware of.
Source_Address	The Bluetooth® address of the Auracast™ transmitter.
PA_Sync_State	Identifies whether the Auracast™ receiver is currently receiving a stream from the BIG with this Source_ID. The common values are: 0×00 = Not synchronized to the Periodic Advertising train for this BIG 0×02 = Synchronized to the Periodic Advertising train for this BIG (Only one instance of the Broadcast Receive State characteristic would normally have this field set as being synchronized, indicating which audio stream is currently being received and rendered.)
Encryption	Shows whether the stream is encrypted and whether the Auracast™ receiver needs a code to decrypt it.
BIS_Sync_State [3]	Identifies which of the individual BISes in the BIG the Auracast™ receiver is currently receiving. The BIS_Sync_State is a bitfield covering the 31 possible BISes in each BIG. For an earbud, speaker, or hearing aid, this will normally be 1; for headphones, it will be 2; and for soundbars, will be 2 or more.
Metadata [3]	A variety of user-readable information which can be used to populate the user interface of an Auracast™ assistant.
Broadcast_Name	The name of the Auracast™ transmitter, such as Pub TV (This is the same as the value in the Broadcast_Name AD Type.).
Program_Info	And optional LTV to denote the content of the audio stream, for example, "Bollywood Films", "Cricket", "Football", etc. (In some cases, where it is not expected to change, the user may set the Broadcast_Name to the value of the content and omit this LTV.).
Language	The language of the stream. This is generally only used when multiple different languages are being transmitted.

[1] In the case of a coordinated pair, each will assign their Source_ID values independently. Auracast™ assistants need to keep track of these differences.

[2] Each Broadcast Receive State characteristic relates to a different Broadcast Isochronous Group (BIG) and includes information about every Broadcast Isochronous Stream (BIS) within that BIG. If an Auracast™ transmitter is broadcasting two different BIGs, then each would be represented by a separate Broadcast Receive State characteristic with a different Source_ID.

[3] The BIS_Sync_State and Metadata are contained in arrays – one for each subgroup in the BIG.

Table 2.1: The main Auracast™ related components of the Broadcast Receive State characteristic

These instances can be thought of as database entries for each BIG from an Auracast™ transmitter which the Auracast™ receiver considers it might want to connect to, as depicted in Figure 2.4. It shows four instances with the instance identified as Source_ID 3 currently being received and with its PA_Sync_State having a value of 0×02 (Synchronized to PA).

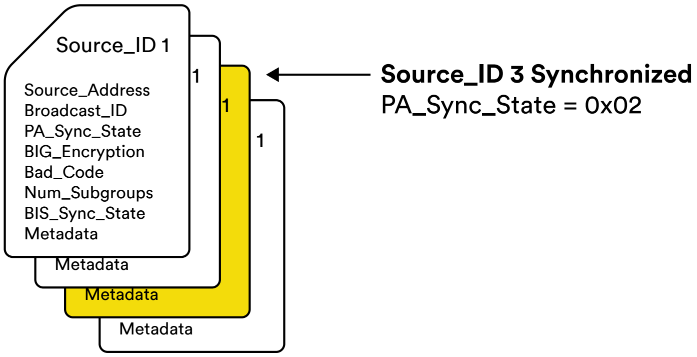


Figure 2.4: A visualization of multiple Broadcast Receive State characteristic instances

To provide a good user experience, it is recommended that multiple instances of the Broadcast Receive State characteristic are supported on every earbud – at least six, and preferably ten, to allow each earbud to maintain a list of available Auracast™ transmitters which are within range. This provides a number of practical advantages:

- Local controls on an earbud can be used to cycle through broadcast streams
- Multiple Auracast™ assistants can use the data to synchronize their lists of available streams
- Non-scanning Auracast™ assistants (e.g., older, non- Bluetooth® LE Audio phones) can also be supported[5]

An Auracast™ receiver can populate these instances independently or delegate the scanning to one or more Auracast™ assistants, which will populate them using the Broadcast Audio Scan Control Point operations.

Many earbuds will have configuration options which use the information they discover to decide whether to add an Auracast™ transmitter to their database. Obviously, stream configurations which cannot be rendered should be ignored and not added to the database. Other filtering decisions may be made based on the metadata, such as parental controls and language.

2.6 Selecting an Auracast™ Audio Stream

The Bluetooth® LE Audio specifications define the methods for discovering Auracast™ transmitters but not the decision process of which transmitter to accept. That allows for innovation and differentiation in the marketplace. How that choice is made may be automatic, configurable, or left to an Auracast™ assistant. At the simplest level, it can be a button on an earbud which selects the first Auracast™ transmitter it finds, looks for one it has connected to before, or toggles through each Auracast™ transmitter it is aware of based on the instances of the Broadcast Receive State characteristic.

[5] *For information on how this is achieved, see the Integrating Legacy Smartphones Into the Auracast™ Experience document [11].*

At any point after an Auracast™ receiver has found an Auracast™ transmitter with streams which it can render, it can stop scanning and proceed to synchronize to and render a stream. This is normally the result of a user action on the earbud or an Auracast™ assistant. Once it has synchronized successfully, an earbud would normally terminate its scanning to conserve its battery.

If an Auracast™ assistant is used, it will request each earbud to synchronize to a stream by writing a value to the BIS_Sync field of a new or existing instance of the Broadcast Receive State characteristic. It should set the value of the PA_Sync field to 0x01 if PAST is available or 0x02 if it is not. The Auracast™ receiver should then attempt to synchronize to that BIG.

2.6.1 Handling BASE Metadata

Metadata may be present in both the Public Broadcast Announcement and the Level 2 subgroup structure of the BASE structure. The applicable Metadata LTV structures are defined in Section 6.12.6 of the Assigned Numbers document.

If relevant metadata is found in the Public Broadcast Announcement or BASE, it should be written into the appropriate instance of the Broadcast Receive State characteristic. Auracast™ receivers need to be aware that metadata can be changed by an Auracast™ transmitter during the life of a BIG. If an Auracast™ receiver or assistant wants to follow any changes in metadata, it should regularly retrieve the PBP and BASE information from the Auracast™ transmitter. The frequency of reading these is implementation specific.

Metadata Name	Description
Preferred_Audio_Contexts	Not used in broadcast.
Streaming_Audio_Contexts	Not used in broadcast.
Program_Info	Unlikely to be used by an Auracast™ receiver unless it has a display.
Language	Indicates the language of the content being broadcast in a subgroup. An earbud may be configured with language preferences, in which case it will use this information to choose between multiple streams.
CCID_List	Not normally used in broadcast applications, unless the Auracast™ transmitter contains media control functionality.
Parental Rating	Describes the suggested minimum age for listeners of the broadcast content. It may exist in the Public Broadcast Announcement or the BASE. Some Auracast™ receivers may use this to block content from being rendered. If this is supported, the Auracast™ receiver should allow configuration of the accepted ages.
Program_Info_URI	A URI link which a scanner can use to obtain more information about the content of an audio stream. Unlikely to be used by an Auracast™ receiver unless it has a display.
Audio_Active_State	Indicates whether or not audio data is currently being transmitted. This is used where audio data is only transmitted intermittently, such as with public transport announcements. An Auracast™ receiver may use this to determine whether or not to synchronize.

Metadata Name	Description
Broadcast_Audio_Immediate_Rendering_Flag	If present, an Auracast™ receiver can ignore the Presentation Delay in Level 1 of the BASE and render the audio content as soon as possible. The same Presentation Delay value must be used by all members of a coordinated set, which implies a proprietary mechanism to ensure synchronization.
Assisted Listening Stream	Indicates the presence of an audio stream which has been preprocessed to enhance the audio intelligibility for a user with hearing loss. This is predominantly of use to users of earbuds and Personal Sound Amplification Products (PSAPs). Hearing aids normally carry out personalized sound processing on the original stream.

Table 2.2: Available metadata structures and their usage

2.6.2 Choosing Which BIS to Use (Mono vs Stereo)

A pair of earbuds will normally have their Sink Audio Locations characteristics set to front left or front right to denote which ear they are in. They will use this information to choose the appropriate stream to receive, using the information in the Audio_Channel_Allocation LTV metadata located in the Level 3 structure of the BASE to identify the appropriate BIS. The Level 3 structure also contains the BIS_index which defines the order of the BISes within the BIG. This is used by the receiver to determine which BIS events to receive.

A manufacturer may provide an option for a user to change the allocation of which BIS a device should render. There are a number of practical reasons to do this, such as swapping left and right to render a mirror image of the sound or preferring a mono stream where one is available, which may be appropriate if the user is wearing a mix of a hearing aid and an earbud. These decisions are up to the implementation. See Section 6.2 for more information on this topic.

An Auracast™ transmitter denotes a mono stream by omitting the Audio_Channel_Allocation LTV metadata from the Level 3 information for a BIS. If no stream is available for a receiver matching the value in its Sink Audio Locations characteristic, it should consider synchronizing to an appropriate mono stream if one is available.

2.6.3 Receiving Multiple BISes

There is no reason why an earbud cannot receive more than one stream other than practical limitations in terms of its resources. Devices like headphones and soundbars will obviously want to do so to reproduce stereo signals. However, receiving redundant information is an unnecessary drain on battery life. Bluetooth® LE Audio has been designed to optimize the resources required by Auracast™ receivers by supplying them with just the audio data they need to do their job. An Auracast™ receiver should choose to receive only the streams which carry the data it needs to render. It can turn its receiver off while an Auracast™ transmitter is sending other BISes.

Auracast™ transmitters should not transmit multiplexed audio streams. These are streams where multiple audio streams, such as left and right, are combined into a single BIS. Support for these multiplexed streams is not mandatory in BAP or HAP, which means that not all Auracast™ receivers can handle them.

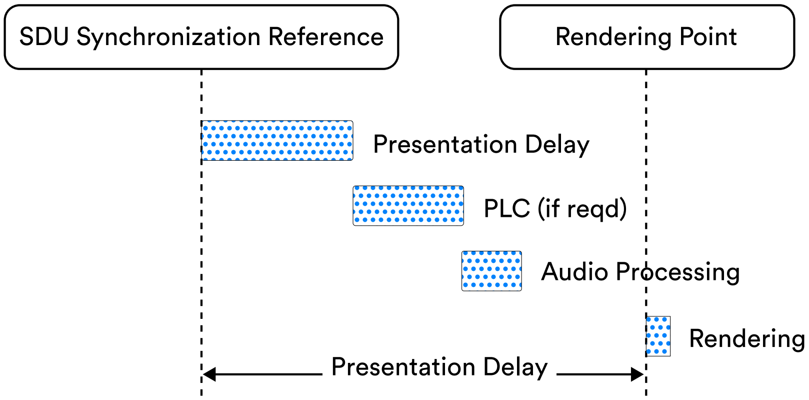


Figure 3.1: The use of Presentation Delay

While some Auracast™ receivers may be able to receive both left and right streams and downmix them to mono, this is not a mandatory requirement and should not be assumed. If an Auracast™ transmitter is supporting an application where mono may be needed, it should transmit mono as a separate stream so that receivers do not need to mix multiple audio streams. This should exist in a separate subgroup to the one containing the two stereo BISes (left and right) and must signify it is mono by not including an Audio_Channel_Allocation LTV at Level 3.

Auracast™ receivers may often have the choice of synchronizing to stereo or mono streams. Implementers should include a policy for the circumstances in which each is preferred. For example, if a user is only wearing one earbud, or the battery in one has gone flat, the single working earbud may decide to render a mono stream rather than a single left or right stream from a stereo pair. The decision is implementation specific and may or may not be exposed to the wearer. It should be made on the basis of providing the most appropriate stream for a variety of different circumstances to provide a fail-safe experience.

2.6.4 Working With Multiple BIGs

An Auracast™ transmitter may contain more than one BIG. If it does, each will have a unique set of advertisements with different Broadcast_IDs. However, they may share a common Broadcast_Name if they contain the same content which differs only in codec configuration. As each Broadcast Receive State characteristic instance applies to a single BIG, the presence of two BIGs would require two separate instances. An Auracast™ assistant may decide to store and present both options to the user, but, in most cases, it would probably only present one based on a local configuration preference for a codec configuration. This is implementation dependent.

The specifications do allow an Auracast™ receiver to synchronize to more than one BIG at a time. However, this can be resource intensive and is not expected to be supported in Auracast™ receivers in the short term.

3.0 Rendering a Broadcast Stream

Regardless of which BIS an earbud is receiving, it knows the point in time when the last BIS in the BIG will have finished being sent, which is the SDU Synchronization Reference. The earbud has to decode and process its received audio packet, rendering it at an exact time after that reference, defined by the Presentation Delay. The typical steps in this process are shown in
Figure 3.1.

The value of Presentation Delay that an Auracast™ receiver should use is set by the Auracast™ transmitter. It is calculated to help achieve the desired overall latency (of which Presentation Delay is only a part) whilst providing time for the device to decode the audio packet, apply Packet Loss Concealment (PLC), and any additional audio processing, such as Active Noise Cancellation (ANC). Auracast™ receivers need to support buffering to meet the Presentation Delay and ensure that their controller clock, which receives the SDU Synchronization Reference, is aligned with the higher layer clock controlling audio rendering.

All Auracast™ receivers must be able to complete these tasks within a value of 40ms (the default defined in BAP). However, this is a relatively relaxed value which produces an overall latency that may be too long for use cases with live sound or lip-sync requirements. Because of this, HAP and TMAP require faster processing times, allowing Presentation Delay to be set at 20ms. Auracast™ receivers supporting HAP or TMAP should therefore be able to support any value from 20ms to 40ms.

Auracast™ transmitters may set values outside this range. Manufacturers should set the range of values for Presentation Delay that their Auracast™ receivers can support without introducing audio artifacts due to incomplete processing or buffering limitations and always work within this range. If they receive a value outside that range, they should revert to the closest value that they can support. An Auracast™ receiver should not reject a broadcast stream based on the value of Presentation Delay.

If an Auracast™ receiver detects that the Broadcast_Audio_Immediate_Rendering_Flag has been set in either the Public Broadcast Announcement or the BASE structure, they should use the lowest value of Presentation Delay which they support[6]. This flag is typically present where an Auracast™ transmitter's content is live and a low latency is required.

When sets of earbuds or hearing aids use an alternative value for Presentation Delay, either because they support the Broadcast_Audio_Immediate_Rendering_Flag or have been presented with a value outside their range of support, both devices must apply the same value of Presentation Delay to ensure that they render in sync. This may be a value set at manufacture or a dynamic value if they have a means of communicating with each other. In both cases, this is implementation specific.

[6] *The main purpose of the Broadcast_Audio_Immediate_Rendering_Flag is for low-latency applications in ear-worn devices. Speakers and devices which render at a distance are advised not to support it.*

4.0 Working With Encrypted Streams

If an Auracast™ receiver detects that the streams in a BIG are encrypted, it will need a Broadcast_Code to decrypt them. The Auracast™ receiver should check that the streams in a BIG are encrypted by examining the length of the BIGInfo7. For unencrypted streams, the length of the BIGInfo is 33 octets; for encrypted streams, it is 57 octets. Either all streams are encrypted or none are encrypted. The encryption status cannot be changed during the lifetime of a BIG.

An Auracast™ receiver normally uses an Auracast™ assistant to acquire a Broadcast_Code. It does this by notifying the value of the BIG_Encryption field in the Broadcast State Characteristic in accordance with the simple state machine shown in Figure 4.1. It should notify a value of 0×01 in the BIG_Encryption field if it has not received a Broadcast_Code and a value of 0×03 if its currently stored value of Broadcast_Code does not work.

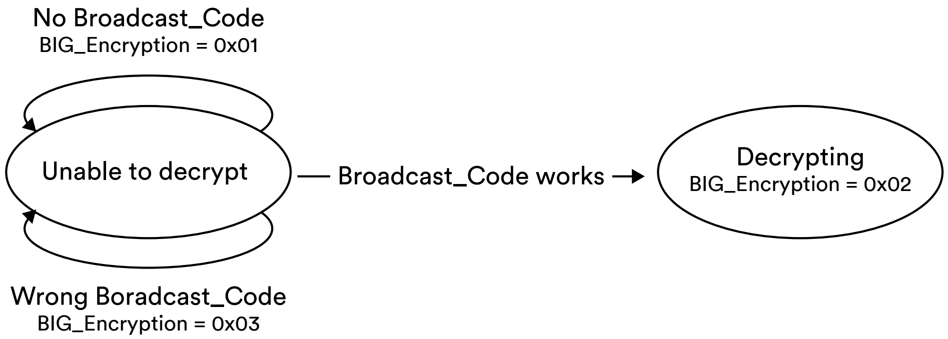


Figure 4.1: The Broadcast_Code encryption state machine

An Auracast™ receiver should store the value of Broadcast Code for at least the duration of its synchronization to the stream and preferably until the associated instance of the Broadcast Receive State characteristic is deleted. This value may persist across multiple synchronizations with a BIS.

[7] *An Auracast™ transmitter will also set the encryption bit in the Public Broadcast Announcement to indicate that the BIG is encrypted, reflecting the content of the BIGInfo. The size of the BIGInfo should be taken as the definitive statement of encryption for a BIG.*

5.0 Working With Auracast™ Assistants

In most circumstances, an earbud would use its Scan Delegator role (from BASS) to look for one or more Auracast™ assistants to perform the scanning function, reducing the power consumption requirements of the earbud. Speakers and headphones, which have larger batteries, may decide not to delegate this task but perform the scanning themselves. The process for scan delegation and soliciting is described in BAP. More information on their use is provided in the How to Design an Auracast™ Assistant document [10].

As the Broadcast Receive State characteristic defined in BASS is a GATT characteristic, it can be read by devices which do not perform scanning. Broadcast Receive State characteristics can be created or updated via the Broadcast Audio Scan Control Point characteristic, which is a GATT characteristic. This means that a pseudo Broadcast Assistant functionality can be provided on legacy phones, tablets, and PCs, working with the Auracast™ transmitter data they read from their Auracast™ receiver(s) Broadcast Receive State characteristics. This process is described in detail in the Developing Auracast™ Receivers with an Assistant Application for Legacy Smartphones document [11].

6.0 Other Design Considerations

Although this document has confined itself to broadcast applications, there are generic Bluetooth® LE Audio features that are likely to be implemented in earbuds, which apply to both unicast and broadcast applications. These are briefly described in this section.

6.1 Volume Control

Features such as volume control are expected to be supported on Auracast™ receivers. The volume model for Auracast™ broadcast audio is that the gain is always applied at the Auracast™ receiver and not at an Auracast™ transmitter. This allows multiple listeners to adjust their received volume to meet their preferred level. Auracast™ transmitters should encode audio at a fixed level in order to provide the best dynamic range for receivers.

Volume control is implemented using the Volume Control Service and, optionally, the Volume Offset Control Service and the Audio Input Control Service. These are independent of broadcast streaming and apply to both unicast and broadcast streams.

6.2 Published Audio Capabilities and Audio Locations

Every Bluetooth LE Audio device acting as an Acceptor (which is the case of earbuds and other receivers) must implement one or more PAC records, which expose the capabilities of the device across both unicast and broadcast applications. The PAC records provide a list of supported codecs and codec configurations. They can also include metadata LTV structures which are relevant to the device's operation.

Auracast™ receivers that wish to receive streams other than mono must include a Sink Audio Locations characteristic, which lists the available rendering location(s). Any value implicitly includes support for mono. The choice of whether to prefer a mono stream is down to the implementation. If the Auracast™ receiver allows a user choice of rendering location (for example, through a left/right/mono switch on a speaker), then the Sink Audio Locations characteristic must be present and use a value of 0×00000000 when mono is preferred. If an Auracast™ receiver only supports mono, then the Sink Audio Locations characteristic can be omitted.

The information included in these characteristics is used by Auracast™ assistants to determine the capabilities of an Auracast™ receiver, allowing them to discard Auracast™ transmitter streams which do not match the receiver's capabilities. More information on their use is provided in the How to Build an Auracast™ Assistant document [10].

6.3 Support for Multiple Retransmissions

An Auracast™ transmitter retransmits its audio data packet unconditionally, as it has no way of knowing whether any devices in range have received them. If it is set for four retransmissions, it will always transmit the original audio data packet followed by four retransmissions. Auracast™

receivers are made aware of the number of retransmissions from the information contained in the BIGInfo structure.

An Auracast™ receiver will normally stop listening for retransmissions once it has received a valid packet, saving unnecessary power consumption. It may also decide to limit the number of subevents on which to listen for retransmissions as a way of conserving its battery life. However, this carries a risk that in a noisy environment, it may fail to receive a packet, resulting in audible artifacts. For this reason, Auracast™ receivers should always be able to support the recommended number of reception slots for retransmissions for the mandatory codec configurations of BAP. BAP expects Broadcast Sinks to be able to support four retransmissions (i.e., a total of five transmissions of each packet).

Auracast™ Transmitters may use other Link Layer parameters to adjust the time diversity of retransmissions in an attempt to increase robustness to interference. Support for these is optional in both Broadcast Sources and Broadcast Sinks, but Auracast™ transmitters may use them to increase the robustness of their transmissions. Table 6.1 summarizes examples shown in version 5.4 of the Bluetooth® Core Specification, which transmitter designers may decide to implement.

Core Reference Version 5.4 Volume 6, Part B	LL Parameter				
	ISO_Interval	BN	NSE	IRC	PTO
Fig 4.34	20ms	2	4	3	0
Fig 4.35	10ms	1	5	3	2
Fig 4.36	20ms	2	6	2	4

Table 6.1: Core examples for the use of BN, IRC, and PTO

Transmissions using these parameters will generally result in longer latencies, but that is determined by the transmitter. Auracast™ receiver designs should ensure their implementation has sufficient buffer to support them or strategies to accept a reasonable number of retransmissions if Link Layer parameters of BN, IRC, and PTO are used to increase robustness.

6.4 Packet Loss Concealment

All Auracast™ receivers should support Packet Loss Concealment (PLC). This may use the PLC specified in Appendix B of the LC3 specification or a vendor-specific implementation.

7.0 References

[1] Basic Audio Profile (BAP)

[2] Public Broadcast Profile (PBP)

[3] Common Audio Profile (CAP)

[4] Telephony and Media Audio Profile (TMAP)

[5] Hearing Access Profile (HAP)

[6] An Overview of Auracast™ Broadcast Audio

[7] Low Complexity Communications Codec (LC3)

[8] Broadcast Audio Scan Service (BASS)

[9] Coordinated Set Identification Service (CSIS)

[10] How to Design an Auracast™ Assistant

[11] Developing Auracast™ Receivers with an Assistant Application for Legacy Smartphones