Bluetooth®

How to Build an Auracast™ Assistant

This document covers the basic design principles for building an Auracast™ assistant, explaining the features and the configuration requirements. It also discusses a range of different options for physical form factors.

Author: Market Development

Version: 1.0

Revision Date: 14 May 2024

Table of Contents

Table of Contents

1.0 Designing Auracast™ Assistants

The Auracast™ assistant is an embodiment of the Broadcast Assistant – a totally new concept in Bluetooth® audio. It exists to support the new Auracast™ broadcast audio use cases, allowing users to discover and select an audio stream, yet it does not transmit, receive, or render audio itself. The Broadcast Assistant is defined as a role in BAP [1], which can be implemented in devices as diverse as phones, remote controls, fobs, or charging cases, where it adds new functionality to an existing product. The Bluetooth LE Audio specifications do not give any guidance on its physical form factor, its user interface, or the resulting user experience, leaving that up to the imagination and innovation of product and application designers. Without its presence, the broadcast user experience is substantially diminished. This document explains why Auracast™ assistants are needed and provides guidance on how they they can be implemented.

1.1 Nomenclature

The Bluetooth LE Audio specifications each introduce different names for the 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 the 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] .

2.0 An Overview of the Auracast™ Topology

The three devices that make up the Auracast™ topology are shown in Figure 2.1. They work in very different ways.

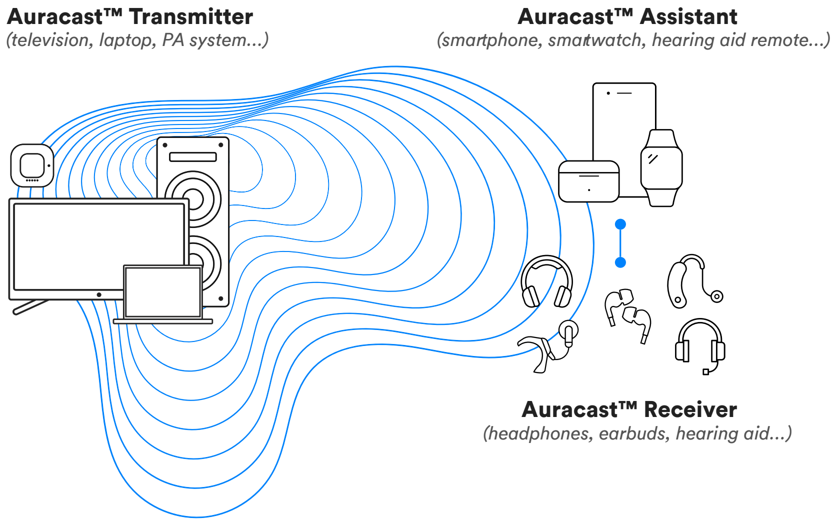
Auracast™ Transmitter
(television, laptop, PA system...)
Auracast™ Assistant
(smartphone, smartwatch, hearing aid remote...)
Auracast™ Receiver
(headphones, earbuds, hearing aid...)

Figure 2.1: The three elements of the Auracast™ broadcast audio topology

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, hearingaids, speakers, etc.) to start receiving and rendering the appropriate broadcast streams. It also takes charge of keeping pairs of Auracast™ receivers synchronized with each other, so that a change on either receiver is immediately reflected on the other.

There are three important points to note in this diagram.

- There is no connection between any of the Auracast™ transmitters and the receivers
- There is no connection between the Auracast™ transmitters and the Auracast™ assistants
- There is no Bluetooth® connection 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.

2.1 The Anatomy of an Auracast™ Assistant

Figure 2.2 shows the key elements of the Auracast™ assistant. It performs five main functions.

- It takes over the relatively power-hungry scanning task of discovering Auracast™ transmitters away from devices like earbuds and hearing aids, which have small batteries
- It provides a better user experience by maintaining a list of available Auracast™ streams, which may be accessible through an enhanced user interface
- It controls the selection, termination, and changing of the broadcast stream
- It is responsible for ensuring that left and right Auracast™ receivers work in unison, receiving the correct broadcast stream
- It can provide a decryption key to Auracast™ receivers when the broadcast streams are encrypted

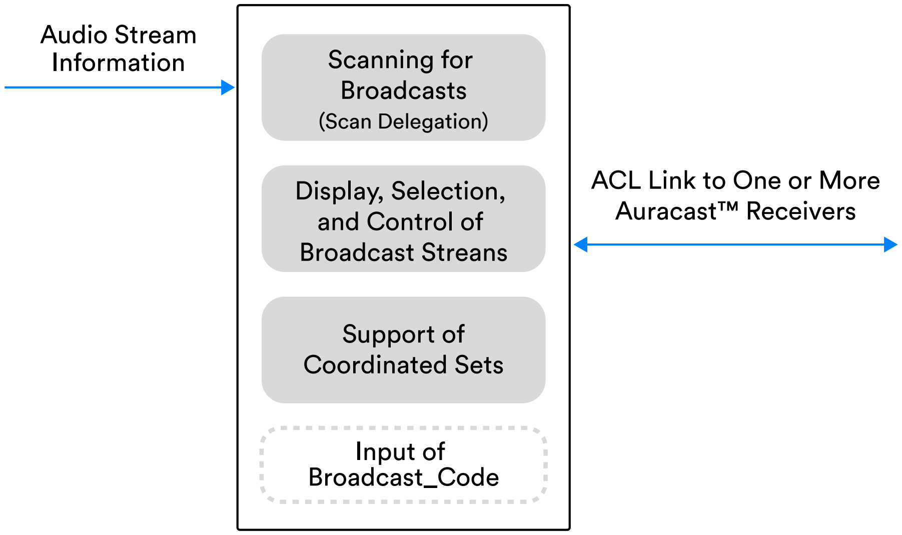
Audio Stream Information
Scanning for Broadcasts (Scan Delegation)
Display, Selection, and Control of Broadcast Streans
Support of Coordinated Sets
Input of Broadcast_Code
ACL Link to One or More Auracast™ Receivers

Figure 2.2: The key elements of an Auracast™ assistant

Each Auracast™ assistant needs to be paired with its Auracast™ receiver. If there are two Auracast™ receivers in a coordinated set, it needs to pair with both of them. Any number of Auracast™ assistants can be used by a pair of earbuds or any other Auracast™ receiver, operating on a first-come, first-served basis.

2.2 The Elements of an Auracast™ Assistant

All of the functionality of an Auracast™ assistant is based on the Broadcast Assistant role defined in the Basic Audio Profile (BAP) [1]. An Auracast™ assistant needs to be aware of the characteristics defined in the Broadcast Audio Scan Service (BASS) [9] and the Published Audio Capabilities Service (PACS) [9], both of which are implemented on Auracast™ receivers. It also needs to be aware of the Public Broadcast Announcement defined in the Public Broadcast Profile (PBP) [2], as well as the Common Audio Profile (CAP) [3] procedures for dealing with coordinated sets (which are defined in the Coordinated Set Identification Profile [11]).

Figure 2.4 shows the specifications within Bluetooth® LE Audio which are required to implement an Auracast™ assistant. Other specifications, such as the Volume Control Profile, Volume Control Service, Volume Offset Control Service, and Audio Input Control Service, may also be incorporated in an Auracast™ assistant device to provide additional, collocated user functionality.

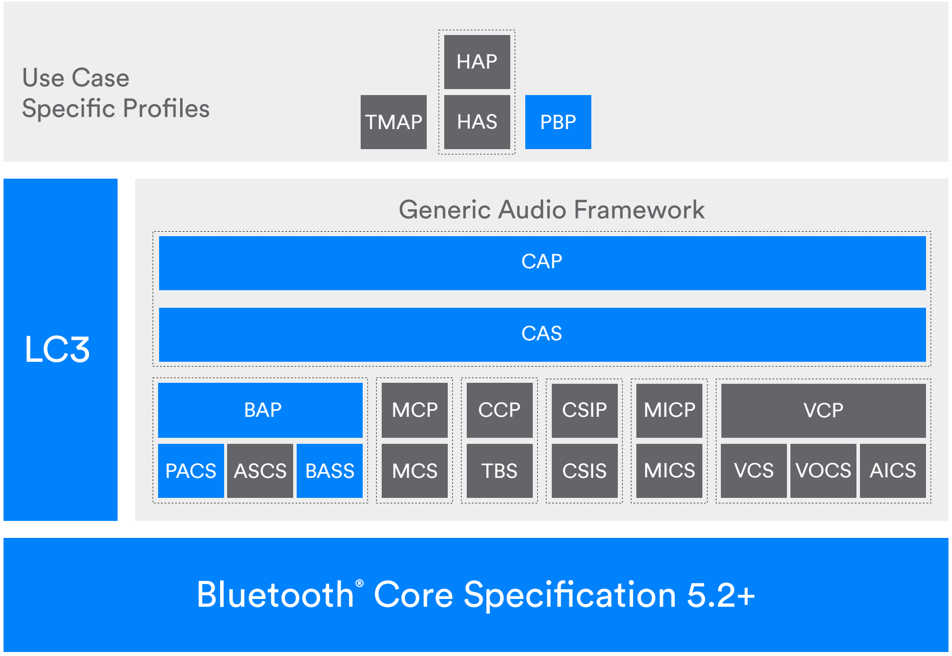

Figure 2.4: Profiles and services used by an Auracast™ assistant.

As we've already noted, none of the Bluetooth LE Audio specifications give any guidance on the design of the user interface, leaving designers total freedom in how they present the user experience. Nor do they specify how Auracast™ assistants use the data they acquire from discovering Auracast™ transmitters. The remainder of this document provides recommendations on how to ensure a good user experience across multiple form factors.

back to contents

3.0 Auracast™ Assistant Form Factors

The Bluetooth® LE Audio specifications give manufacturers complete freedom to design Auracast™ assistants in whatever form they want. They can be standalone products, smartphone applications, or integrated into other products. Despite this, and regardless of their form factor, all Auracast™ assistants are universal and will work with any Auracast™ receivers. However, design choices may limit or differentiate their functionality. Multiple Auracast™ assistants can work on behalf of an Auracast™ receiver at the same time, on a first-come, first-served basis.

Before we look at the implementation details of Auracast™ assistants, it's useful to familiarize ourselves with the variety of forms which they can take as that will influence their functionality. Many designers are unaware of how diverse these physical implementations can be. The following are a few examples of how the functionality may be implemented.

3.1 Application-Based Implementations

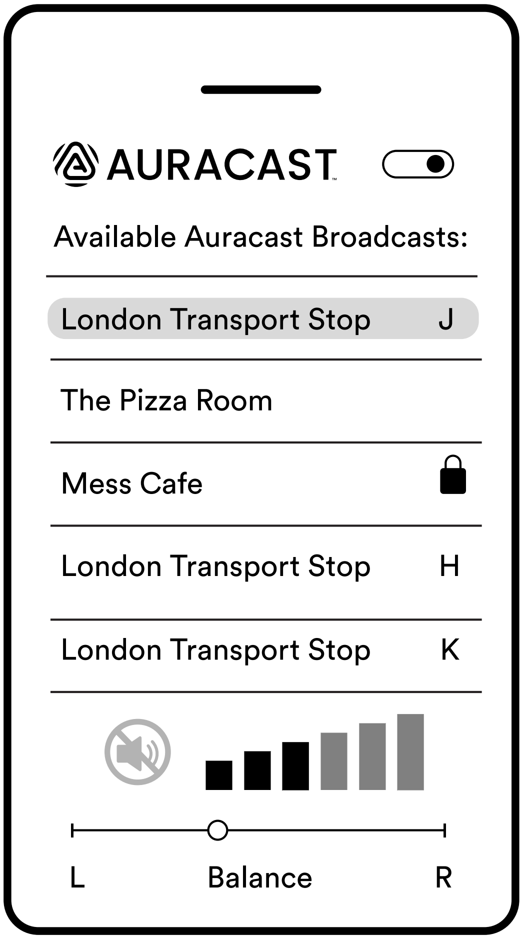

Figure 3.1: An example of an Auracast™ assistant application on a smartphone

There is an expectation that smartphone will become the most common implementation choice for an Auracast™ assistant. In time, the major phone operating systems will provide native support for discovering Auracast™ transmitters and are likely to include APIs to allow app developers to integrate Auracast™ capabilities into their applications.

3.1.1 Smartphone Applications

Figure 3.1 shows an example of an Auracast™ assistant application on a smartphone, which discovers and lists the available Auracast™ streams, allowing a user to select one and start or stop it. This example also shows the application implementing volume control and balance for a pair of earbuds.

An identical interface and user experience can be provided by an app for a legacy, non-scanning smartphone. Although such legacy phones are not able to receive or broadcast Bluetooth LE Audio streams or scan for their presence, they can work with the BASS characteristics on their paired Auracast™ receivers. Information on this option is provided in the Developing Auracast™ Receivers with an Assistant Application for Legacy Smartphones [12].

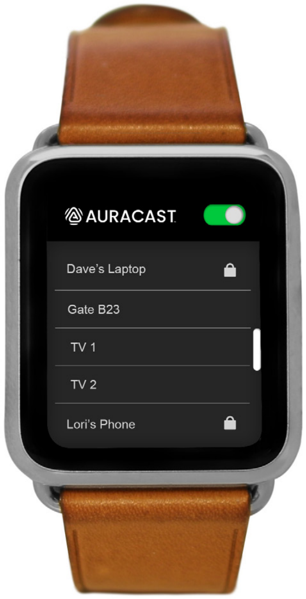

Figure 3.2: A simple Auracast™ assistant selection app for a smartwatch

3.1.2 Smartwatches

In a similar manner, an Auracast™ assistant can be implemented as an application in a smartwatch. This solution is useful for situations where users may not be carrying their phones but they need to select a broadcast, such as when they are training in a gym.

Figure 3.2 illustrates a simple Auracast™ assistant selection app for a smartwatch, displaying three TV streams in a gym. In this case, the gym owner would have identified the three TVs with signs denoting their numbers.

3.2 Dedicated Hardware Implementations

Much of the discussion around Auracast™ assistants has focused on implementations such as smartphone or smartwatch apps. That obscures the fact that the Bluetooth® LE Audio specifications allow them to be implemented in many different forms. The following examples illustrate just a few possible options. These can be implemented as scanning Auracast™ assistants with a Bluetooth LE only chipset which supports Bluetooth Core Specification version 5.2 or above or as non-scanning versions that use a GATT connection to a paired Auracast™ receiver. For new designs, the scanning version makes the most sense.

3.2.1 Charging Cases

Many charging cases for earbuds already contain a Bluetooth chip, and there is a trend to add remote control functionality and/or displays to provide a richer user interface. So, it makes sense to include Auracast™ assistant functionality as well.

Figure 3.3 is an example of a possible implementation where the display on the charging case provides the user with the Auracast™ assistant functionality. Other screens could provide volume and media control using other Bluetooth LE Audio profiles.

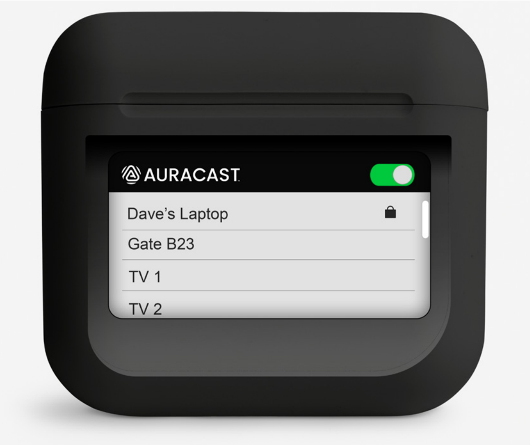

Figure 3.3: A charging case including an Auracast™ assistant

One additional advantage of this approach is that because the earbuds are likely to be paired with the charging case during manufacture, they can operate with the integrated Auracast™ assistant as soon as the earbuds are taken out of the case. This gives a superb unboxing experience where the user only needs to take the earbuds out of the case and place them in their ears, at which point they can select an Auracast™ transmission and immediately receive their first Auracast™ audio stream.

back to contents

3.2.2 Remote Controls

Auracast™ assistants don't need to be complex. While the smartphone is the most logical Auracast assistant, a significant number of users still do not use smartphones[2]. However, they will still need and want an easy way to scan through available Auracast streams.

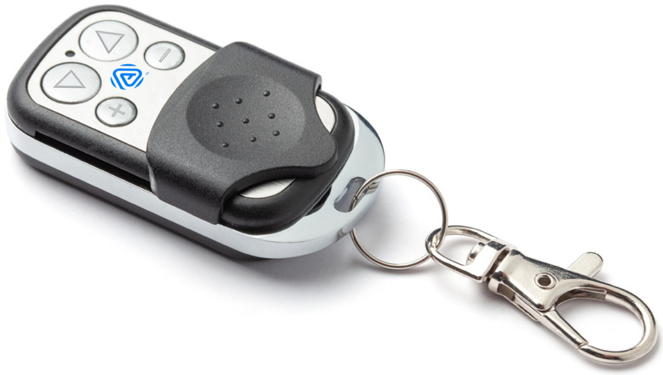

Figure 3.4: A basic remote control style of Auracast™ assistant

In Figure 3.4, a simple keyfob remote control has been used for an Auracast™ assistant implementation. It allows a user to move through the available Auracast™ audio streams using the lower two buttons. In this example, each time a button is pressed it will select the next instance of its stored Auracast™ transmitter details and request the earbuds or hearing aids to start rendering that stream. Pressing the central Auracast™ button can either stop the stream or start scanning for new Auracast™ transmitters. This implementation also includes buttons for volume control.

An implementation like this one might work best in a situation where there are a limited number of Auracast™ transmitters, including applications such as day centers and care homes where users may be less likely to have access to a smartphone.

3.2.3 Lanyards

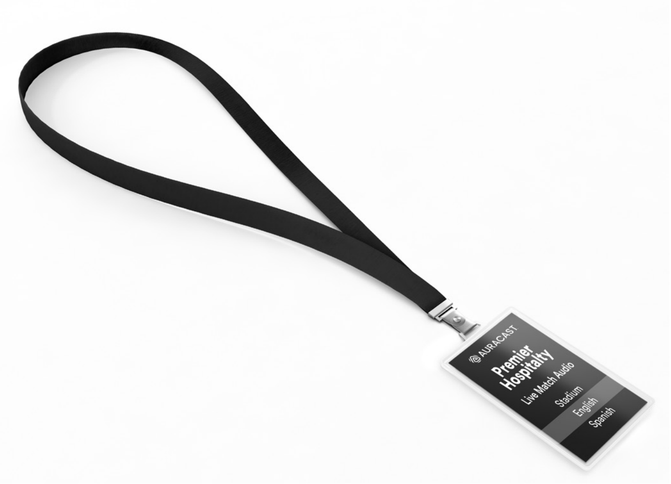

Figure 3.5: A lanyard-based Auracast™ assistant

A similar approach to a low-cost remote control is shown in Figure 3.5, but in this case, a basic touch display has been integrated into a credit card-sized device which can be used with a lanyard.

In this example, it has been configured to select an audio stream for attendees in a hospitality box at a sporting event, allowing them to hear the basic stadium sound of the game or to access a range of different commentaries in the language of their choice. It is likely to be provided along with a pair of earbuds by the event organizer as part of the hospitality package.

The same concept is equally valid for the use cases of the previous example, providing a simple way for residents of a care home to select their preferred audio entertainment.

3.3 Integrated Auracast™ Assistants

All of the examples above cover Auracast™ assistants which only operate as scanning and selection devices. However, there are also good reasons to collocate Auracast™ assistants with Auracast™ transmitters where they can help users connect to those devices.

3.3.1 A Smartphone With Auracast™ Transmitter Functionality

In the future, most smartphones will not only operate as Auracast™ assistants but will also be Auracast™ transmitters, allowing users to share their audio with friends. Collocating the two functions allows the transmitter role to provide information directly to the Auracast™ assistant, including the current value of the Broadcast_Code. The audio application on the phone could include a QR code scanner, making it easy to transition to listening to a friend's Auracast™ stream if they want to share their favorite audio in return.

By combining the two roles, the Auracast™ assistant can either scan for other Auracast™ transmitters or offer an enhanced user interface design when the phone is operating as an Auracast™ transmitter. In this case, an application can generate a QR code which contains all of the information that is required to allow a friend to connect by scanning the screen with their own smartphone. As illustrated in Figure 3.6.

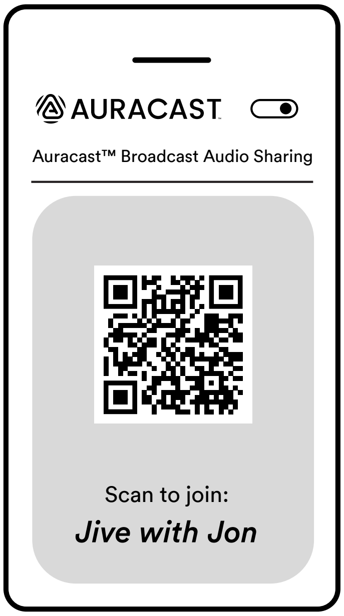

Figure 3.6: An example of a QR Code being generated by a personal music-sharing app

This option would normally be integrated into an audio application, such as a streaming app, which would transition a unicast stream to a broadcast stream when the user wanted to move from personal listening to sharing with friends or colleagues.

3.3.2 A TV With an Integrated Auracast™ Assistant

Consumer TVs are expected to be early adopters of Auracast™ transmitter capability, offering the options of silent listening as well as transmitting multiple audio channels supporting different languages or assisted listening streams. Because TVs are likely to use encryption to protect the audio streams from eavesdropping, they need a mechanism to convey the Broadcast_Code to listeners.

To accomplish this, TVs are expected to combine the Auracast™ transmitter and assistant roles, using the collocation to share information which allows a more intuitive user interface.

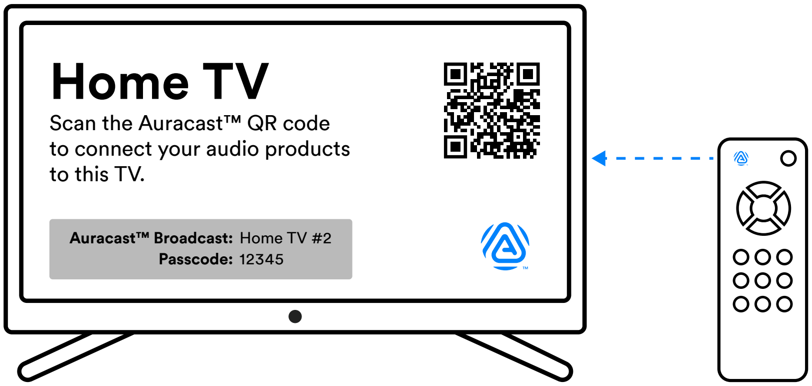
Home TV
Scan the Auracast™ QR code to connect your audio products to this TV.
Auracast™ Broadcast: Home TV #2
Passcode: 12345

Figure 3.7: A potential interface for a home TV

Figure 3.7 shows a potential interface for a TV with a collocated Auracast™ assistant. Users paired with the TV would be able to automatically connect whilst a dynamically generated QR Code, displayed by pressing an Auracast™ button on the TV remote, would let friends join when they come around.

4.0 Designing Auracast™ Assistants

The examples in the previous section show that there are many different ways to provide Auracast™ assistant functionality, but the basic functions that they need are common. There are differences in their user interface and how they convey information, but each needs to support the same functions. In this section, we look at each of those, explaining how the different aspects of an Auracast™ assistant perform and work with each other. These cover seven main categories:

- Pairing and discovering the Auracast™ receiver capabilities
- Responding to solicitations from an Auracast™ receiver
- Scanning for Auracast™ transmitters
- Maintaining a database of Auracast™ transmitters
- Selecting and terminating Auracast™ streams
- Working with coordinated sets
- Managing Broadcast_Codes for encrypted audio streams

4.1 Pairing and Discovering the Auracast™ Receiver Capabilities

Auracast™ assistants need to be paired with each of the Auracast™ receivers they intend to work with. Auracast™ receivers can be headphones or a pair of hearing aids, earbuds, or speakers. The pairing process follows the normal pairing procedure for Bluetooth® LE devices, taking into account any coordinated set where all of the set members need to be paired.

A user is most likely to use an Auracast™ assistant with a single set of Auracast™ receivers, but there is no limitation on the number of different Auracast™ receivers it can work with. An Auracast™ assistant can also operate on behalf of multiple sets at the same time. If an Auracast™ assistant is designed to operate with more than one set of Auracast™ receivers, its user interface should make clear which device or set of devices it is currently managing. In the following discussion, we will assume the Auracast™ assistant is dedicated to a single coordinated set of earbuds.

4.1.1 Discovering the Auracast™ Receiver Properties

After the devices have initially paired, the Auracast™ assistant should read the Sink PAC records of each Auracast™ receiver. These will tell it the range of codec configurations that each supports. With this information, the Auracast assistant can determine whether a transmitter's streams are compatible with the Auracast™ receiver. It should also read the Sink Audio Locations characteristic of each Auracast™ receiver to determine which rendering locations it supports so it can assign the correct Broadcast Isochronous Stream (BIS) to each device when a user selects a stream. If the Sink Audio Locations characteristic is not present, the Auracast™ assistant should assume that it only supports mono streams.

The reason for reading this information is that it provides details of the capabilities of the Auracast™ receivers which the Auracast™ assistant can use to filter the information that it acquires when scanning for Auracast™ transmitters. Table 4.1 lists the information that an Auracast™ assistant should retrieve from each of its Auracast™ receivers.

Characteristic	Parameter	Reason
Sink PAC	Codec_Specific_Capabilities (related to the Auracast™ receiver's LC3 support)	To determine whether the Auracast™ receiver supports the High Quality streams defined in PBP. If these are not supported, then the Auracast™ assistant can ignore any 48kHz streams and confine its database to 16kHz and 24kHz Standard Quality streams1.
Sink PAC	Metadata	Metadata may be included in the Sink PAC to indicate customer preferences for streams. Examples include, but are not limited to: ● Preferred Audio Contexts ● Language ● Parental Rating Metadata LTVs are defined in the Assigned Numbers document [1].
Sink Audio Locations	Sink_Audio_Locations	This identifies the Audio Location(s) supported by the Auracast™ receiver. For coordinated sets, there are generally the physical location of each device. When selecting a stream by writing to the Broadcast Audio Scan Control Point characteristic, the Auracast™ assistant should select the stream with the matching Audio_Channel_Allocation value in the BASE structure. If no Sink_Audio_Locations characteristic is present, this is interpreted as meaning that the Auracast™ receiver expects a mono stream.

Table 4.1: PACS parameters that should be read and stored by an Auracast™ assistant

An Auracast™ assistant may also be able to infer this information from discovering the top-level profile support of the Auracast™ receiver. It should not assume that the information is identical for all members of a coordinated set.

[1] *If an Auracast™ assistant detects a Public Broadcast Announcement which only denotes a 48kHz High Quality stream, it can alert the user to request that the Auracast™ Transmitter change its broadcast configuration to a 16 or 24kHz Standard Quality stream. All compliant Auracast™ transmitters must support this functionality.*

If the Auracast™ assistant does not cache this information, it should reread the Sink PAC record and Sink Audio Locations characteristic each time it reconnects to the Auracast™ receiver(s).

4.2 Responding to Solicitations From an Auracast™ Receiver

It is normally up to an Auracast™ receiver to decide whether it wants to use an Auracast™ assistant to assist it with scanning for Auracast™ transmitters. The Auracast™ receiver may make this decision either to offload the scanning task or to add an additional scanning resource to supplement its own efforts.

When it wants to use an Auracast™ assistant, an Auracast™ receiver uses extended advertising PDUs to solicit any available Auracast™ assistant. Figure 4.1 shows the process.

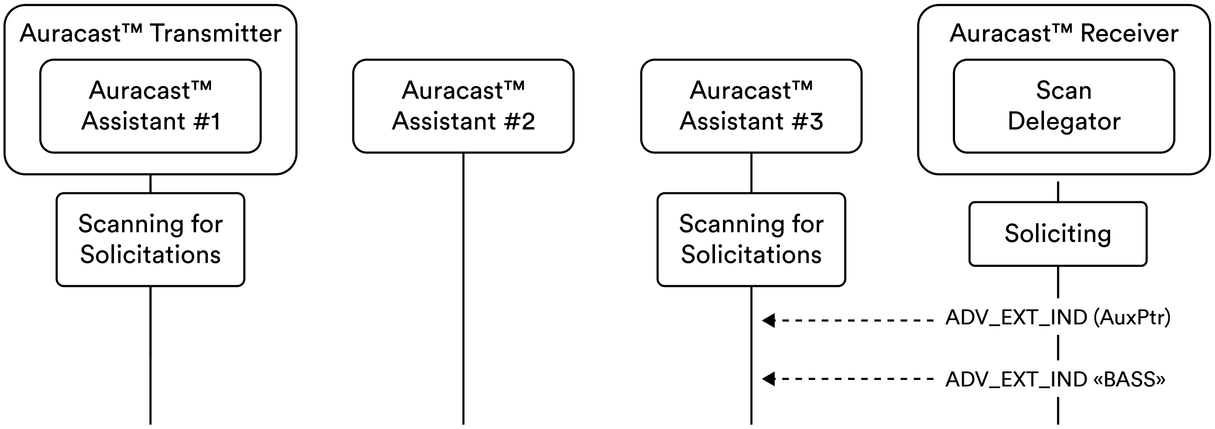

Figure 4.1: The solicitation process

In this example, the Auracast™ receiver uses its Scan Delegator to send out extended advertisements which include the Broadcast Audio Scan Service UUID (0×184F), indicating that it is looking for an Auracast™ assistant. At this point, it has no knowledge of whether any Auracast™ assistants are available to scan for advertisements. Sending out these advertisements is normally triggered by a user action on the Auracast™ receiver, such as a button press.

Auracast™ assistants may perform regular scans to look for solicitations or may only scan as a result of a user action. That is implementation specific. In the example above, only two of the Auracast™ assistants within range are currently scanning and are therefore capable of responding to the solicitation advertisements.

If an Auracast™ assistant wishes to scan on behalf of the Scan Delegator, it responds by establishing an ACL connection and writing to the Broadcast Audio Scan Control Point characteristic of each Auracast™ receiver with the operation code of 0×01, denoting that it has started remote scanning on behalf of that Auracast™ receiver. If that receiver is a member of a coordinated set, which the Auracast™ assistant will determine by reading its Coordinated Set Size characteristic (defined in CSIS [11]), then the Auracast™ assistant will scan on behalf of all set members. Only one member of the coordinated set needs to solicit an Auracast™ assistant. If the Auracast™ assistant finds no

Auracast™ transmitters within a set time, it may inform the Auracast™ receiver(s) that it is no longer scanning on their behalf and terminate its scanning to conserve power.

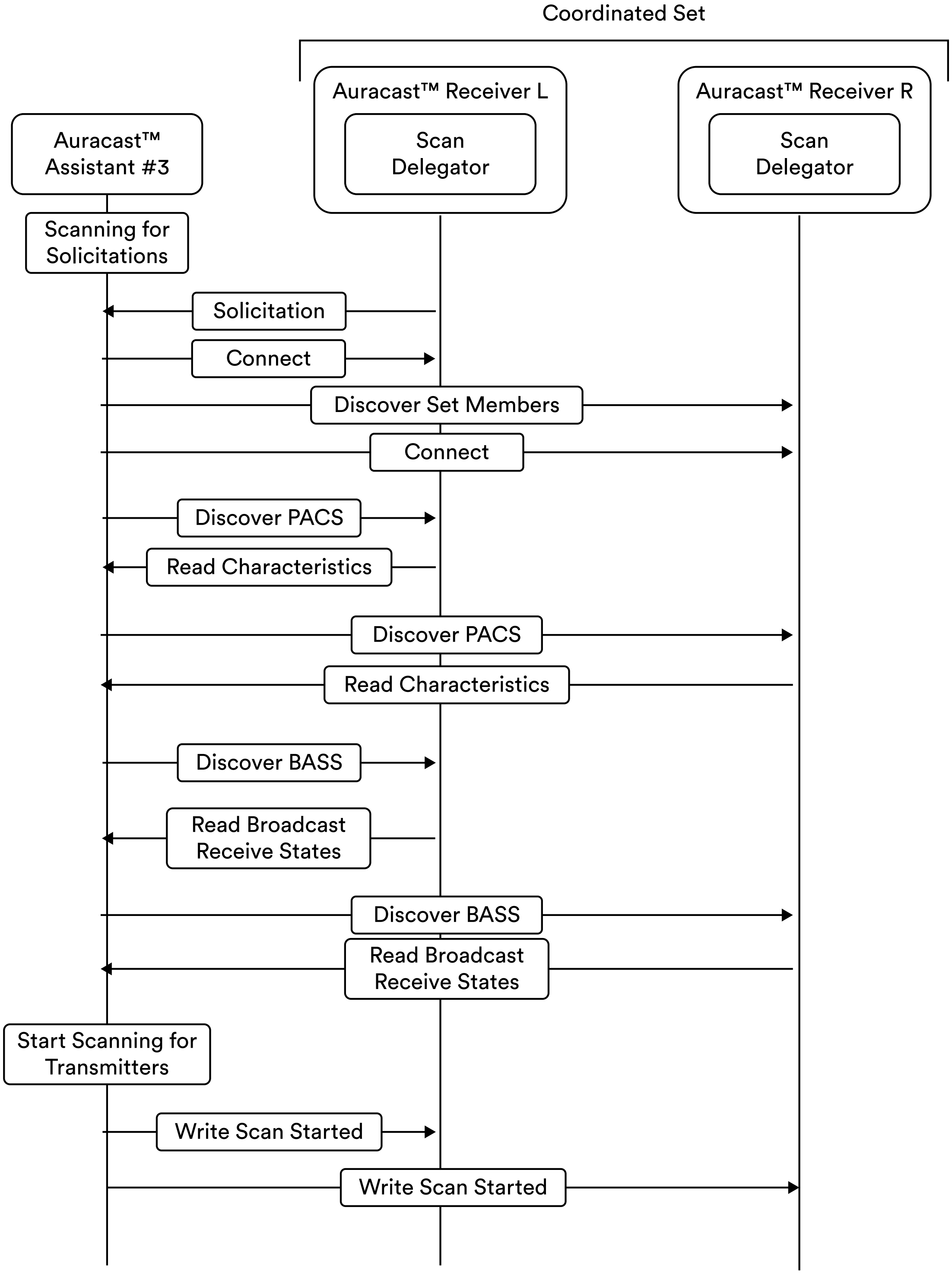

Figure 4.2: The process of registering with members of a coordinated set

Figure 4.2 shows the steps of the process where the Auracast™ assistant is scanning on behalf of a pair of earbuds which form a coordinated set. Before it starts scanning and writing to the Broadcast Audio Scan Control Point characteristic, unless it has its values cached, it needs to connect to both Auracast™ receivers and read their PAC records and Sink Audio Locations characteristics. Only when that is complete, and it has started scanning, should it confirm to each that it is scanning on their behalf.

In this example, only the Auracast™ assistant #3 has responded, but any number of Auracast™ assistants can respond and work with a set of Auracast™ receivers at the same time. When any Auracast™ assistant performs a Broadcast Audio Scan Control Point operation to add or modify a Broadcast Receive State characteristic of one of its Auracast™ receivers, it should perform the same update on each member of the coordinated set. Any other active Auracast™ assistants will receive notifications of these changes and should update their internal records.

If an Auracast™ assistant becomes aware that an Auracast™ transmitter is no longer available (which may be because it has been turned off or is out of range), the Auracast™ assistant should remove it from the database of any connected Auracast™ receiver by writing the Remove Source opcode to the Broadcast Audio Scan Control Point characteristic of each of them.

An Auracast™ assistant can inform an Auracast™ receiver that it is no longer scanning for transmitters on its behalf by writing the Remote Scan Stopped opcode (0×00) to the Broadcast Audio Scan Control Point characteristic. There is no method defined for an Auracast™ receiver to request that an Auracast™ assistant should stop scanning. However, it is at liberty to ignore any writes to its Broadcast Audio Scan Control Point characteristic.

An Auracast™ assistant may make a unilateral decision to scan on behalf of an Auracast™ receiver if it has an active ACL connection by writing the Remote Scan Started opcode to the receiver's Broadcast Audio Scan Control Point characteristic. This can simplify the user experience by allowing them to start the scanning and selection process on their Auracast™ assistant.

4.3 Scanning for Auracast™ Transmitters

The primary task of an Auracast™ assistant is to discover Auracast™ transmitters on behalf of an Auracast™ receiver by scanning for extended and periodic advertisements. An Auracast™ assistant can scan continuously, but it is more likely to do so on request to conserve its battery. Scanning may be initiated by a user action on either an Auracast™ assistant or an Auracast™ receiver scanning autonomously.

Auracast™ transmitters use extended advertisements to publish information about their transmissions. Figure 4.3 illustrates the structure of these advertisements and the key packets of data within them which are captured by an Auracast™ assistant or receiver during the scanning process.

An Auracast™ assistant may decide to regularly scan in the background so that it can immediately provide a current list of Auracast™ transmitters to a user. This is an implementation decision.

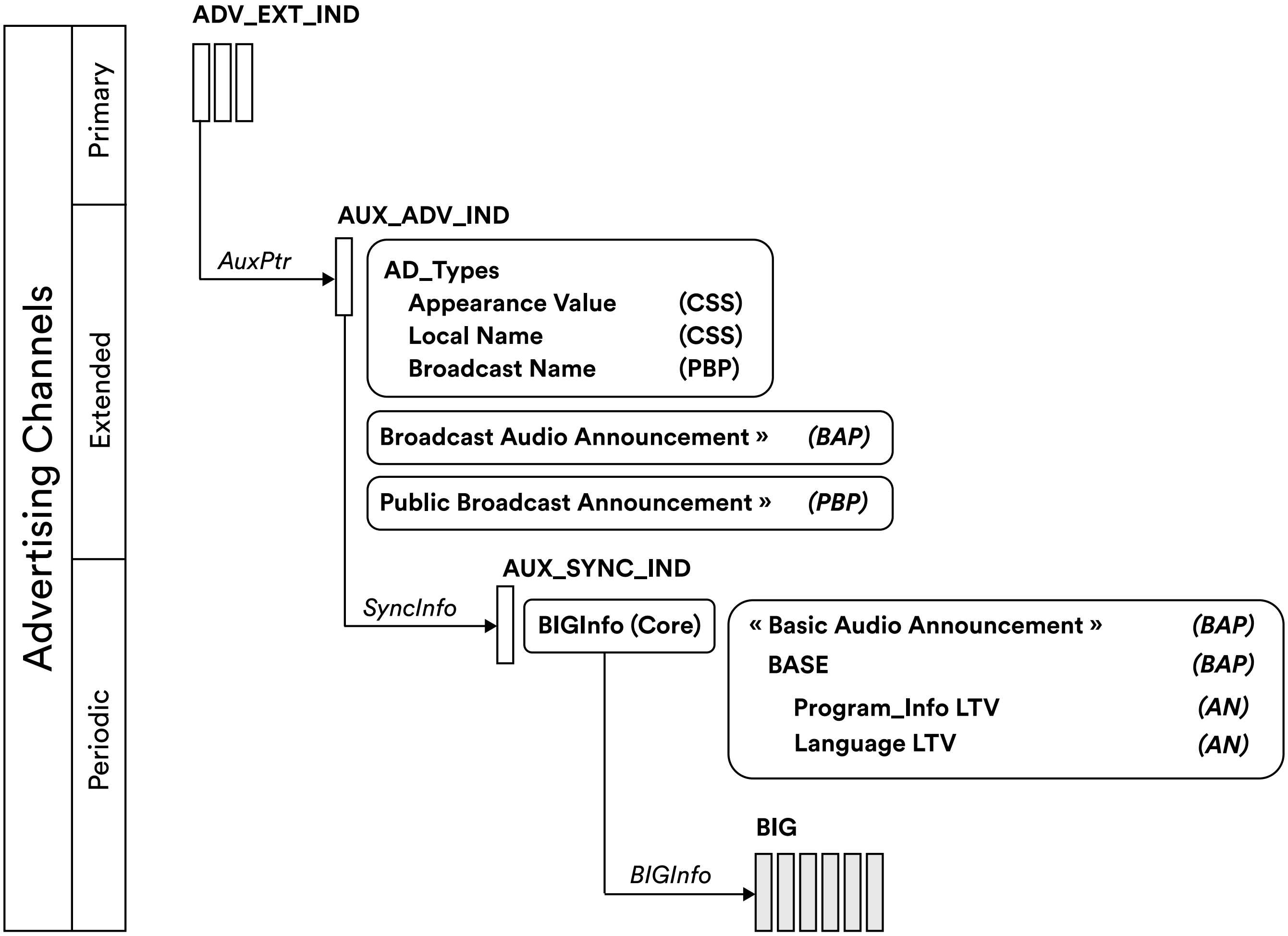

Figure 4.3: The Structure of Auracast™ information from Auracast™ transmitters2

4.3.1 The Scanning Process

An Auracast™ assistant scanning for Auracast™ transmissions needs to work its way through the various elements of these advertising sets for all of the transmitters it can find. For each set of data, it should parse the contents to decide which broadcast streams are relevant and which can be ignored. It makes these decisions based on:

- The information it has determined about its currently connected Auracast™ receivers from reading their PAC records.
- Any local policies that a user has set in the Auracast™ assistant. These are implementation specific.

Figure 4.4 illustrates the basic decision flow during scanning as an Auracast™ assistant or receiver works its way through the information contained in a hierarchy of advertisements.

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

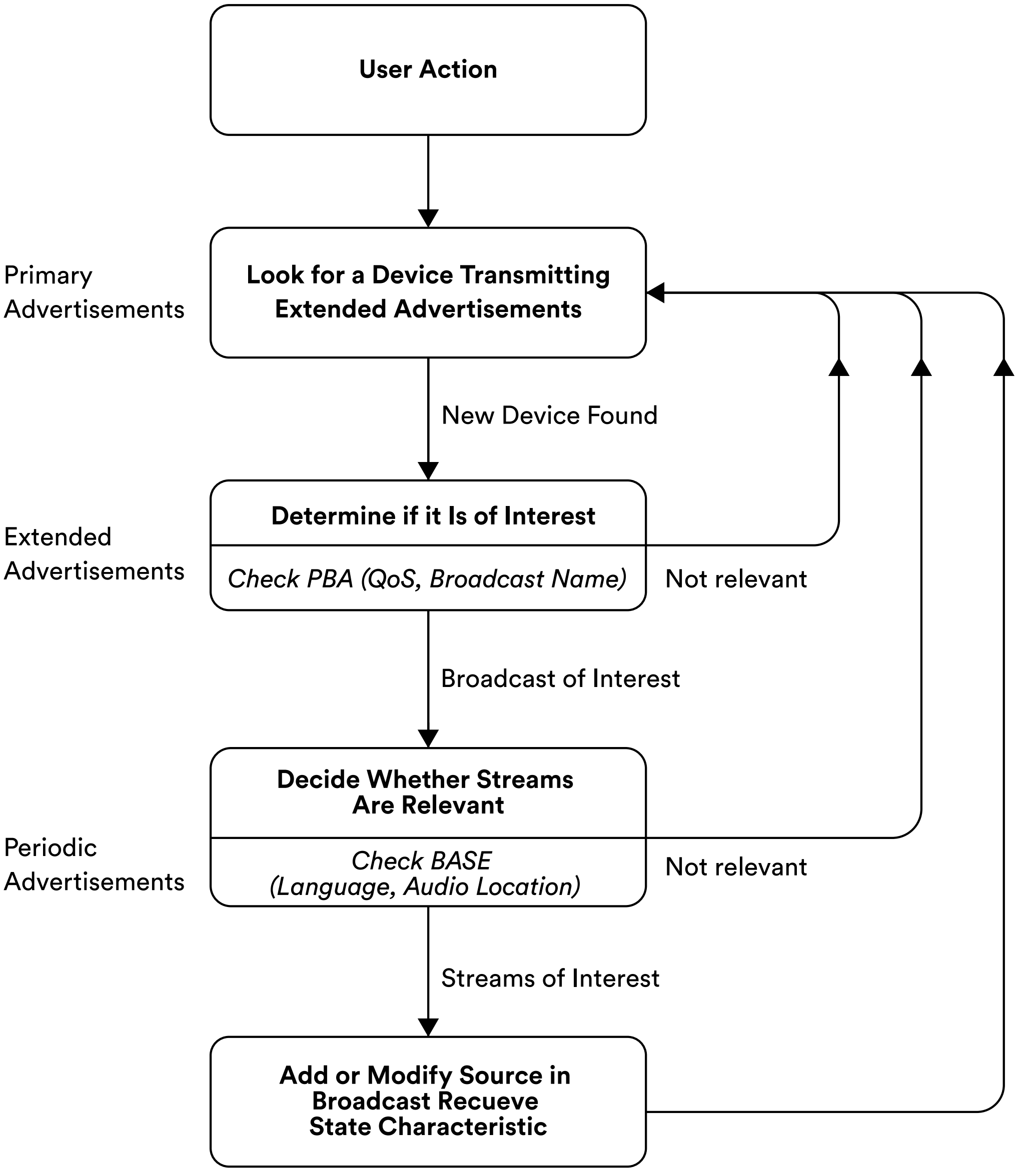

Figure 4.4: Basic scanning process

At each stage of the scanning process, the Auracast™ assistant will be able to make decisions about whether it should skip to the next Auracast™ transmitter or proceed to the next stage of parsing the stream information in the extended advertisement.

4.3.2 Parsing the Extended Advertisement

Referring back to Figure 4.4, a scanning device 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. If it is present, the scanner will then look for the extended advertisement to see if it includes a Broadcast Audio Announcement, signifying that it is a Broadcast Source. The Broadcast Audio Announcement also includes the Broadcast _ID for the BIG. If it is present, the scanning device should parse the Public Broadcast Announcement. This will always be present if at least one of the streams within the BIG claims to be Auracast™ compliant[3].

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
- Whether a 16/24kHz or 48kHz stream is currently being transmitted[4]

The Auracast™ assistant should also retrieve the Broadcast_Name of the Auracast™ transmitter. This is a human-readable name for the stream which is used where the Auracast™ assistant can convey the name of the stream to the user, typically through an integrated display. It is also used to identify known streams or user favorites. The Auracast™ assistant should also 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.

On the basis of this information, the Auracast™ assistant 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 AUX_SYNC_IND or move on to look for other transmitters.

4.3.3 Parsing the Periodic Advertisements

Once synchronized to the periodic advertisement, the Auracast™ assistant needs to acquire the Basic Audio Announcement, which provides detailed information about the individual BISes in each BIG using the Broadcast Audio Source Endpoint (BASE) structure.

The information in the BASE can be split into three categories:

- Content information which helps indicate whether the stream is relevant, such as language
- Practical information, including codec configuration 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 an Auracast™ assistant with more information about the streams and how they should be treated.

Figure 4.5 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.

[3] *If the Broadcast Transmitter 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.*

[4] *If the Public Broadcast Announcement does not show that a 16 kHz or 24kHz stream has been configured, and a user's hearing aid or earbuds cannot support the current 48kHz stream, the user may need to ask for a 16 kHz or 24kHz stream to be enabled. All Auracast™ compliant transmitters must support this functionality.*

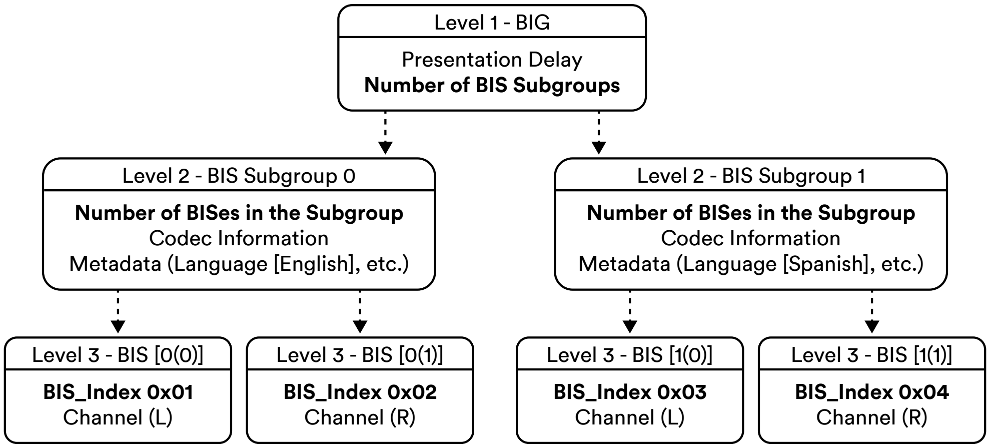

Figure 4.5: A simplified example of the BASE structure and contents

Having collected this information, an Auracast™ assistant may decide, based on what it knows of its receivers and any local policies, that it is not interested in any of the streams in any of the subgroups, in which case it can discard this BIG to move on and look for more transmitters or additional BIGs from the same transmitter. If any of the streams described in the BASE are of interest, the Auracast™ assistant should write the information to each of its Auracast™ receivers. At this stage, it is just populating or updating an instance of a Broadcast Receive State characteristic on each of its Auracast™ receivers by performing an Add Source of Modify Source operation on its Broadcast Audio Scan Control Point. Before performing this operation, the Auracast™ assistant must check the current status of each Auracast™ receiver, as it is not allowed to create duplicate entries of the Broadcast Receive State characteristic. If an instance of the Broadcast Receive State characteristic already exists for this Auracast™ transmitter, it must modify the existing entry.

An Auracast™ assistant should be aware that its connected Auracast™ receivers may be scanning independently at the same time, so the state of their Broadcast Receive State characteristics may have changed. They should also be aware that the receive sensitivity of their Auracast™ receivers may be poorer than their own, which means that their Auracast™ receivers will not be able to synchronize with them. An Auracast™ assistant may take this into consideration, limiting its display to discovered Auracast™ transmitters with an RSSI value that is above a locally determined threshold. However, this is an implementation decision.

4.4 Maintaining a Database of Auracast™ Transmitters

The Bluetooth® LE Audio specifications do not require an Auracast™ assistant to do anything with its scanning data other than writing appropriate information to its Auracast™ receivers. However, it makes sense for it to maintain a local database of the Auracast™ transmitters it has found.

How this is implemented is up to individual implementations, but an obvious approach is to maintain a local database which is a copy of the Broadcast Receive State characteristics from its Auracast™ receivers. These will be notified if there is any change, either from local scanning or inputs from an Auracast™ assistant, allowing an Auracast™ assistant to update its own database as well as ensuring that any other Auracast™ assistant has a synchronized set of instances5 of the Broadcast Receive State characteristics.

4.4.1 Writing to the Broadcast Receive State Characteristic

An Auracast™ assistant writes or modifies an instance of the Broadcast Receive State characteristic by using the Write or Modify operations in the Broadcast Audio Scan Control Point characteristic. There is a lot of data which is included in each Broadcast Receive State characteristic. Basically, it contains:

- A unique identifier 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, which BIS(es) it is currently receiving
- Whether it is encrypted and the Auracast™ receiver has the correct code to decrypt it
- Any metadata which may be relevant in deciding whether to synchronize to it

The full list of parameters is defined in Table 3.2 of BASS and all of these features are mandatory. When any change is made to any instance of the Broadcast Receive State characteristics, it will be notified. This includes changes made autonomously by the Auracast™ receiver.

The Auracast™ assistant should perform an Add Source or Modify Source operation on the Broadcast Audio Scan Control Point of its receiver(s) as soon as it has determined that it has found an Auracast™ transmitter which meets its policy criteria. At this stage, it will not normally request that its Auracast™ receiver(s) synchronize to it. That normally requires a user to make a specific selection.

The available space and number of Broadcast Receive State characteristic instances will differ between different Auracast™ assistants and Auracast™ receivers. Both need to implement strategies to manage the number of instances using the Remove Source operation defined for the Broadcast Audio Scan Control Point to keep the Auracast™ receiver's list of Broadcast Receive State characteristics relevant and fresh.

4.5 Selecting and Terminating Auracast™ Streams

While an Auracast™ assistant may just be used to perform scanning on behalf of an Auracast™ receiver, leaving the user to select which stream they want to receive on their earbuds, it is far more likely that the user will take advantage of a more complex interface on the Auracast™ assistant to make that selection.

To select a stream which will already be present in an instance of the Broadcast Receive State characteristic on its receivers, the Auracast™ assistant needs to invoke the Modify Source operation in the Broadcast Audio Scan Control Point to request that the Auracast™ receiver synchronizes with the selected stream. This involves two parameters. The first is the request to synchronize with

[5] *There may be minor differences, as each Auracast™ receiver assigns its own value to the Source_ID, and the BIS_Sync values may reflect different Audio Locations. For this reason, an Auracast™ assistant should keep copies of the values for each Auracast™ receiver on whose behalf it is scanning.*

the BIG identified by the Source_ID. It would normally default to requesting that this occur using PAST where the Auracast™ receiver can request the syncinfo data, allowing it to directly access the periodic advertising train and the BIGInfo data that it needs. The second is to set the appropriate bits in the BIS_Sync parameter, which determines which individual BIS or BISes that device should use. An Auracast™ assistant can write the value 0xFFFFFFFF (unspecified) to the BIS_Sync, which is interpreted as allowing the Auracast™ receiver to make its own choice.

An Auracast™ assistant should always perform the same operation on every Auracast™ receiver in the coordinated set with the appropriate parameter values.

An Auracast™ receiver may make an autonomous decision to synchronize to a BIG. In this case, it should update the relevant Broadcast Receive State characteristic and notify that change. An Auracast™ assistant receiving that notification should update any other Auracast™ receivers in the same coordinated set with which it currently has a connection. An Auracast™ receiver may ignore the instruction contained in an Add or Modify Source request based on its local policy. Devices need to include strategies to deal with this, which would be triggered by a failure to detect an expected Broadcast Receiver State characteristic notification resulting from a Broadcast Audio Scan Control Point operation.

4.6 Working with Coordinated Sets

As the pair of earbuds (or hearing aids) have no connection between them, notifications become extremely important. They are used to ensure that all Auracast™ assistants are kept up to date with the status of both of the earbuds. This is the mechanism that ensures that 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 notifiable characteristic or by an autonomous action on the earbud, that change is notified to all Auracast™ assistants, which will use their knowledge of the coordinated set members to update the information on the other earbud. This should, 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 4.6. An event at the right Auracast™ receiver results in a notification being produced. This is received by the Auracast™ receiver, which then writes to the other member of the coordinated set to keep its value synchronized.

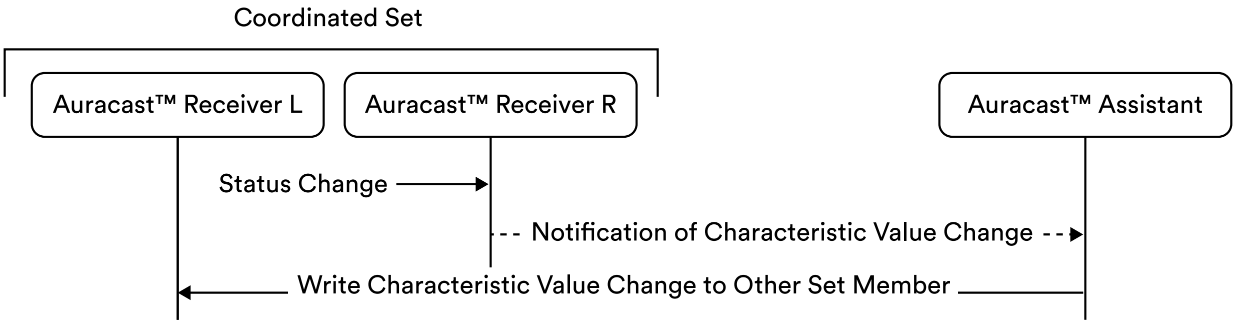
Coordinated Set
Auracast™ Receiver L
Auracast™ Receiver R
Auracast™ Assistant
Status Change
Notification of Characteristic Value Change
Write Characteristic Value Change to Other Set Member

Figure 4.6: The basic principle of using notifications in coordinated sets

Notifications are also used for other control functions, such as volume, ensuring that any changes are applied at the same time to both earbuds. Where earbuds are using more than one Auracast™assistant, it ensures that both Auracast™assistants are aware of the latest state of both earbuds.

A consequence of this approach is that if a pair of earbuds do not both 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. All state information is stored independently on each earbud and Auracast™ assistants are responsible for ensuring that any change on one is reflected to the other.

Many earbuds and hearing aids include a proprietary sub-GHz radio to manage this synchronization. Auracast™ assistants have no way of knowing whether or not such a mechanism exists and should always respond to notifications on the assumption that such a proprietary link is not present. An Auracast™ may be able to infer the presence of a proprietary link if it receives notifications from both members of a coordinated set without having written to both of their Broadcast Audio Scan Control Point characteristics. But this is an implementation feature which is outside the scope of the Bluetooth® LE Audio specifications.

4.7 Managing Broadcast_Codes for Encrypted Audio Streams

Auracast™ assistants are expected to supply Broadcast_Codes to Auracast™ receivers in response to a notification asking for them. If the Auracast™ assistant has a copy of the current Broadcast_Code for the requested BIG, it should send it by writing the Set Broadcast_Code operation to its Broadcast Audio Scan Control Point.

The Auracast™ receiver should check whether this code works, and, if so, set the BIG_Encryption value in the relevant Broadcast Receive State characteristic to 0×02 indicating successful decryption, which will be notified. If it fails, it should set the value to 0×03 (Bad_Code) indicating that it has an incorrect encryption key. The Auracast™ assistant will receive these values in a notification. In the case of a Bad_Code, the Auracast™ assistant should supply a correct value.

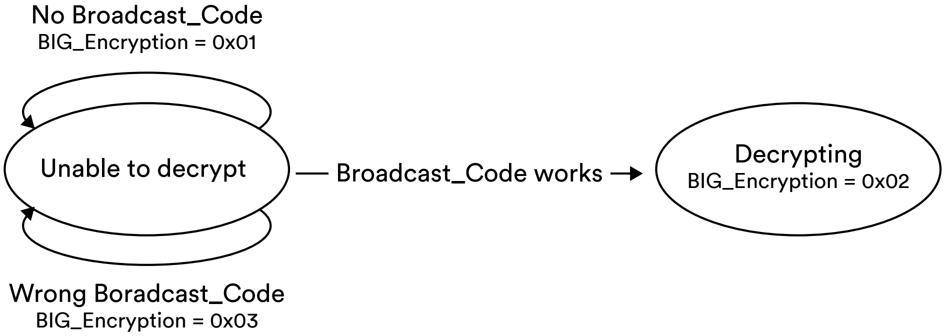

Figure 4.7: The Broadcast_Code encryption state machine

Figure 4.7 illustrates the state machine used by an Auracast™ receiver to control its requests for a Broadcast_Code.

4.7.1 Finding the Broadcast_Code

There is currently no specified method for an Auracast™ assistant to find the Broadcast_Code. It is assumed that most methods will involve some form of out-of-band transfer. The following options are possible approaches.

4.7.1.1 Manual Entry of a Code by a User

This follows the method in common use for Wi-Fi where an application on an Auracast™ assistant asks a user to manually enter a passcode, which is typically printed and displayed in the locality of the transmitter. The method of entry will be down to the implementation, but the Broadcast_Code should be an appropriate alphanumeric string.

4.7.1.2 Entry of a Code by an Automatic Method

An Auracast™ assistant may support an automatic method of entering a Broadcast_Code, such as by scanning a QR code or tapping an NFC tag. Currently, there is no standard format for this information, but Bluetooth Working Groups have developed recommendations.

4.1.1.3 Collocation With Auracast™ Transmitters

Some Auracast™ transmitters, such as TVs may be collocated with an Auracast™ assistant. In this case, the Broadcast_Code can be passed internally from the Auracast™ transmitter to the Auracast™ assistant implementation. This is a particularly useful approach when the Broadcast_Code is changed regularly. An Auracast™ transmitter cannot change the Broadcast_Code during the life of a BIG. If it needs to be changed, the BIG must be terminated and reestablished.

5.0 Conclusion

Auracast™ assistants are a vital part of the Auracast™ ecosystem, allowing Auracast™ receivers to select from a growing number of Auracast™ transmitters in a growing number of Auracast™ broadcast audio locations. The Bluetooth® LE Audio specifications provide a toolbox for designers to develop a wide range of different implementations, ranging from applications running on smartphones and smartwatches to a diverse range of innovative hardware solutions.

Although most new hardware designs will be based on Bluetooth Core Specification version 5.2 or later versions of chipsets which support scanning for Auracast™ transmitters, it is also possible to design GATT-based devices using earlier Bluetooth chips, providing the same rich user experiences for consumers. This allows the market to grow and innovation to proceed prior to the availability of native support in major phone and PC operating systems. For more information on this approach, please refer to the Developing Auracast™ Receivers with an Assistant Application for Legacy Smartphones [12].

6.0 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] How to Design an Auracast™ Earbuds

[8] How to Design an Auracast™ Transmitter

[9] Broadcast Audio Scan Service (BASS)

[10] Published Audio Capabilities Service (PACS)

[11] Coordinated Set Identification Profile (CSIP)

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

[13] Assigned Numbers (AN)