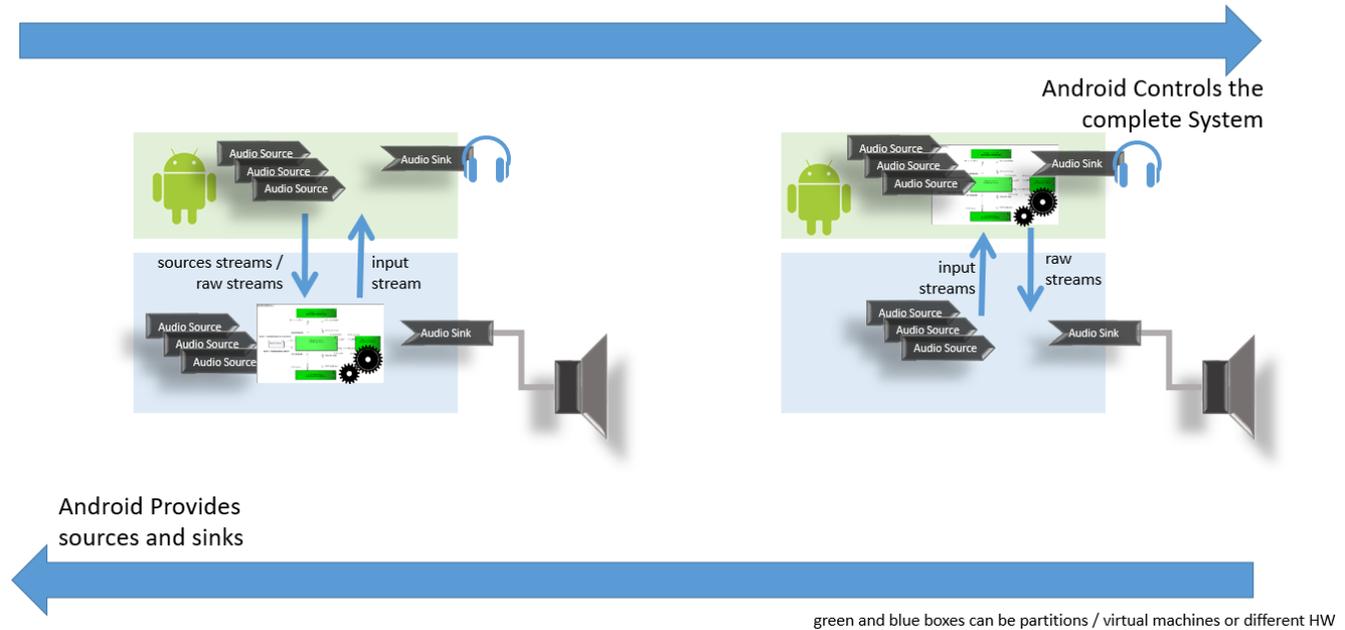


# Android and System Level Audio

- Investigation Answers

Possible partitioning over a hybrid approach  
 Definitions  
 Proof Of Concept - Getting Raw Streams  
 Android Automotive with Multizone and CarAudioService

## 2 Options



Android and System Level Audio.pptx

## Investigation Topics

Things that we need to investigate in order to make the evaluation of which approaches are feasible.

- Raw streams from Android.
  - Can we get the raw streams from any source, app, system, etc sent outside of the system without unnecessary preprocessing (see definition of Raw Stream below).
  - How is this done?
  - Are there limitations? For example, if apps themselves take decision on type, priority, effects, and even volume, then this is not a pure raw stream
  - What Raw Stream Metadata (definition below) can be extracted and delivered together with the stream to an outside system.
- Input streams to Android (external sources need to be brought into the Android head unit, if all mixing is done inside Android).
  - How is this done?
  - Are there limitations on the number of streams?
  - Can we bring the required metadata in (or at least associate "hard coded" metadata with the identity of a particular incoming stream. For example we know this one is a prioritized type of stream)
- Any particular support in Android for networked audio? Audio output device?
- Anything to consider about compressed audio / codecs? Support for hardware implementations?
  - Not in AOSP. The connection to hardware accelerators is expected to be implemented by vendor.

## Investigation Answers

- Audio devices visible to Android for both internal and external streams are defined in *audio\_policy\_configuration.xml* ([1] [2])
  - configuration is provided for each audio hardware module/driver (represented by Audio HAL module),
  - there should be at least Primary Audio HAL that will provide primary output from Android to the vehicle via AUDIO\_DEVICE\_OUT\_BUS and also that will expose primary input devices like FM TUNER or AUX IN,
  - if there is other audio output hardware, it is usually represented by other Audio HAL module (e.g. A2DP Audio HAL, USB Audio HAL, EAVB Audio HAL) - it also comes with corresponding input device, if available (e.g. A2DP Out, A2DP In).
- Raw streams:
  - Streams have properties. It seems any of these encodings might be used in the HAL interface: [https://source.android.com/devices/audio/data\\_formats](https://source.android.com/devices/audio/data_formats)

- Not all compressions might be supported in a particular platform, it should be expressed in the audio policy configuration which devices support direct output of offloading of compressed data to HW. PCM is considered as default, especially in Android Automotive.
  - By default all Android internal streams are going through Framework Mixer. This can be omitted by:
    - specific audio policy configuration where some devices are marked to handle direct output - then audio streams might be routed to that device (either by AudioManager decision or application preference) - ref. [example for HDMI multi-channel](#)
    - the app when using [AAudio API](#) in EXCLUSIVE mode based on MMAP buffers - Audio HAL implementation must support this.
  - Android Automotive is expected to be configured in fixed-volume, i.e. all internal streams have fixed volume. Volume control requests are sent directly to Audio HAL by separate API (setAudioPortConfig) - gain controllers are also defined in the audio policy configuration. Ref: [Configure volume](#) . NOTE: there's also dedicated API for controlling fader/balance ([AudioControl HAL](#) - extension to Audio HAL) but implementation is hardware specific and it's to the vendor how to utilize this.
  - What are referred to as metadata match the [Android term "attributes"](#), the two main attributes are :
    - **Usage** : (Communication, Alarm, Notification,...) the list is relatively long
    - **Content type** : (Unknown, Movie, Music, Sonification, Speech) and this one is important for post processing configuration
- Input streams to Android
  - Audio HAL exposes ALSA input device via abstracted audio device with particular type (e.g. AUDIO\_DEVICE\_IN\_FM\_TUNER) and properties defined in the audio policy configuration.
  - Can be handled via:
    - [createAudioPatch/releaseAudioPatch API](#) (no longer recommended since Android10)
    - [HwAudioSource Player](#) (aimed to replace above-mentioned API)
    - [AudioRecord/AudioTrack API](#) - kind of loopback, representing external stream via internal stream
  - No high-level limitation on the number of streams, limitation might come from particular kernel audio driver implementation or hardware.
  - A small presentation about it can be found uploaded [here](#)

## Analysis of both approaches

Function	Android Provides sources and sinks	Android Controls the complete system
How to get Raw Streams	applicable (individual streams or / streams per type ?)	applicable (difficulty ?) (individual streams not required / streams per type required ?)
Common HAL		
Mixing	then getting raw streams are required	

## Possible partitioning over a hybrid approach

Function	inside Android	outside Android

## Definitions

### Raw Stream:

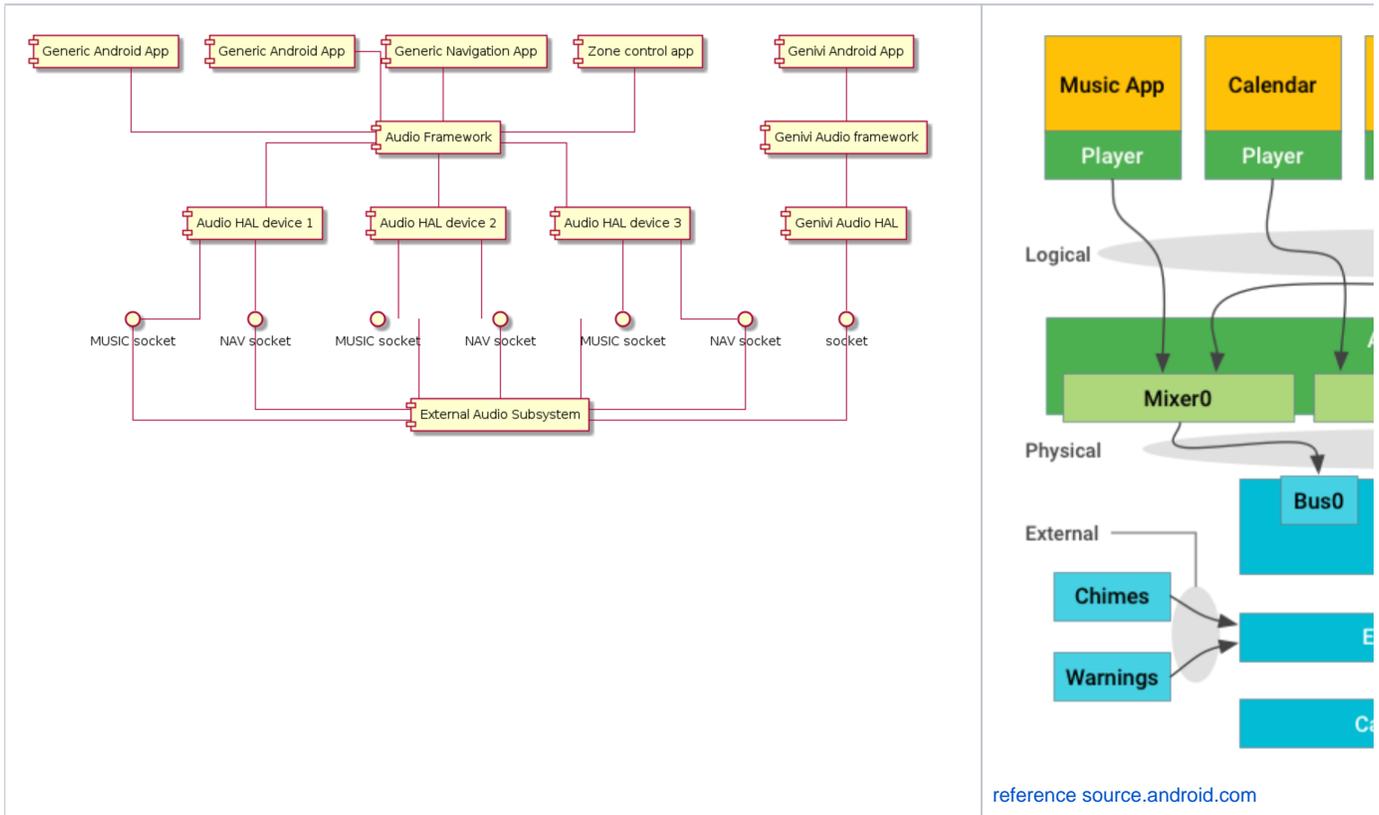
- = The audio stream as coming from a source (PCM format mostly, or other encoding such as Dolby for multi-channel, in that case indicate this).
  - There shall be **none** or **minimal** application of Gain/Volume (stream should be at nominal / maximum volume), Effects (no effects applied), or Mixing (one stream at a time).
  - We can however consider 2-ch stereo, or multi-channel, as a single stream as long as it is transferred so that it can be trivially separated for treating the channels separately in the outside system.

### Raw stream metadata:

- = Information *about* the stream.
  - Where it is coming from (an app? the HMI? the system?).
  - What type of audio is it (Audio track of a video playback? Music? Navigation prompt? Urgent and safety-critical warning message? HMI feedback because you pressed a button?).
- Metadata is used in an external mixing and effects system to take the decision about how to mix, how to apply volume and effects, and especially for priority of one sound over another.

## Proof Of Concept - Getting Raw Streams

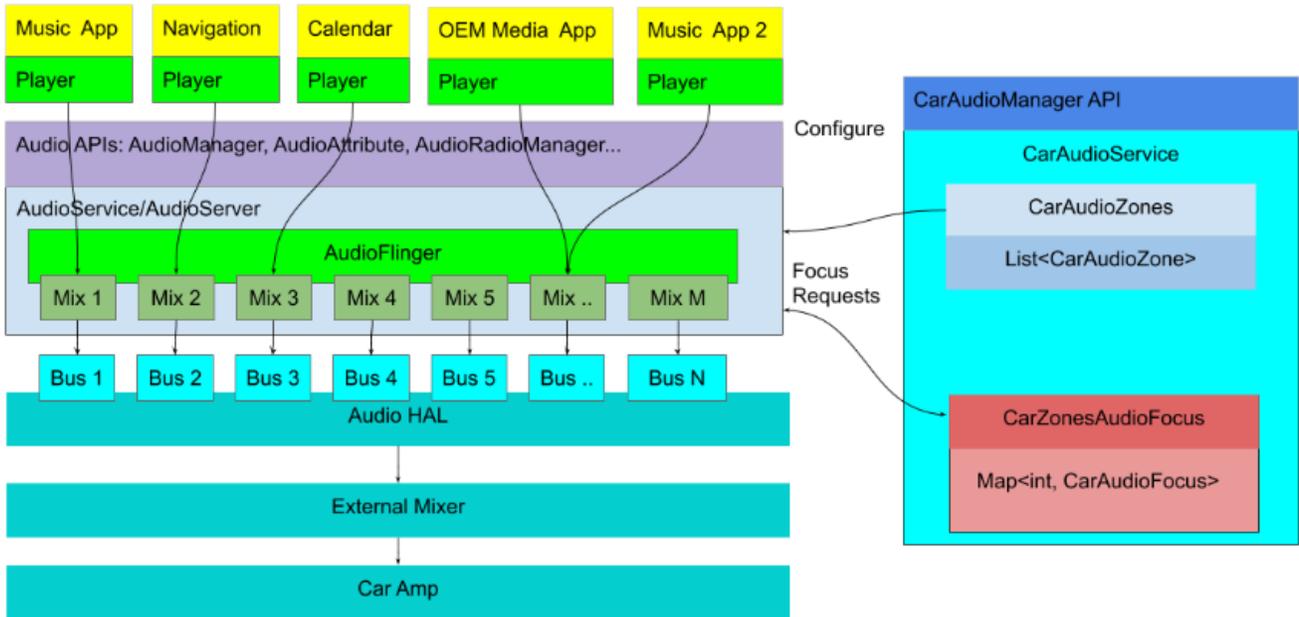
side by side with the Android Automotive sounds and streams



(screenshot from the attachment below - Piotr Krawczyk - Tieto Evry)

[Audio PoC.pptx](#)

## Android Automotive with Multizone and CarAudioService



[https://source.android.com/devices/automotive/audio/multi\\_zone/overview](https://source.android.com/devices/automotive/audio/multi_zone/overview)

