List of prioritized topics for the Audio HAL

> **WARNING**
>
> WORK IN PROGRESS

- 8
- 9
- 13

	Topics / Priority (high, medium, low)	Category (all TBC)	workshop	Suhasini	Andrey	Mobis all TBC	Stefan K	Piotr	Wassim Alexander	Gunnar	Chris

1	Networked audio devices
	
	• need for a better support (of the configuration) of networked audio devices.
	• currently Android can view only a single device / sound card.
	• an audio network in car might have multiple devices that need to be controlled by the Android OS.
	• related need: a silicon vendor has a portfolio of audio processors and an automotive audio bus to distribute audio within the car, and want to configure their proprietary audio bus SW stack to use it with AA as they did for Linux
	• No easy way to support external amplifiers
	  ○ Audio data interface (PCM Stream, AVB, A²B,...)
	  ○ Focus on Control interface (1 to 1 mapping, including local logic in the local HAL) depends on other topics analysis
		Source Management		High	High			Medium	High		Medium

2	Overall configuration management		+	Medium				High	Low		Low
	configuration of automotive overlay, there is not much tool to configure the AA system	**Configuration**			Low						
	configuration management component for TinyALSA	**Configuration**			Medium						
	configuration challenge (look at slide 12 of tech summit deck)	**Configuration**			Low						
3	Common Audio HAL HAL extension / custom (requires to modify the service layer first adding hidl definitions breaks the vts) • new vendor service has to interoperate with other services. • back channels (app / HAL, sync audio flinger and app commands) • worth looking into the audio effects library (might be handled as audio effects plugins,..) • e.g. mic arrays : limitation of the framework as the hardware only can detect from where the voice is coming. • (look at slides 15-20 of tech summit deck) relates to extracting stream topic.	**Common HAL**		High	Medium			Medium	High		Medium

#	Description	Category									
4	**Audio data transfer to DSP** • streaming to a co-processor for post processing of audio from the HAL / other Android layers • related, need: we want to determine whether we have a problem of bandwidth when using our proprietary network with AA, one approach is to use shared memory transfer in the audio HAL relates to extracting stream topic	Common HAL		High	High			Medium			
5	**Equalization** • there is no way to control the equalization, i.e. no simple way to control global effects for output streams (by default Android application controls its own tracks) Might be outdated by Android 11 relates to Controlling Audio Effects	Common HAL	-	High	High			High			
6	**Audio Calibration** • there is no Audio calibration interface relates to Equalization	Common HAL	-+	Medium	Medium			High			

7	**Controlling Audio Effects**
	
	• there is a need for a generic interface for controlling audio effects at HAL level, global effects are designed for input streams but control over them is limited by the available interface
	
	Examples Virtual surround,
	
	Tied to the DSP function.
	
	• If the DSP is on a different device would the interface be the same.
	    ◦ recommended to start with design to clarify the question including an architecture.
	    ◦ some API exists for real DSPs only need to extend over the network.
	
	Remark: while doing research, Unknown User (bartoszbialek) noticed that Android team is tackling this and that they will be offering API to control the global effect. This topic is no longer relevant.
	
	To consider Android 11
		Common HAL	-	High	High			High		High	Medium

8	Latencies **Remark:** We can use the Android Audio API that Unknown User (bartoszbialek) once mentioned in _AUDIO_HAL_CW16_ _Full audio path latency, should have a measure on the PoC of latency_	Performance	+++	High	High			Medium			
9	Multiple audio channels • the challenge is to adapt the AA framework to HW I/O, e.g. there are 4 audio channels that need to be presented as 2 audio channels to AA could be a matter of configuration Android Down mixes in case you'd like to route through Flinger to 16 PCM, otherwise mixing would be outside. Default behavior (Audio Flinger can distinguish other use cases, compress data : offloaded, specific API AAudio) Sound Open Firmware, relates to ALSA	Common HAL	+++	High	High			Low			

10	Early audio • for RVC (Rear View Camera) or other services (warning audio signal) **Remark:** This is addressed by the Android Automotive team here. Android Early Audio and External System early audio. Topic that requires handshake between Android and an external Audio system.	**Common HAL**	++	Low	Low			Medium			
11	Source Management • Multisource multisink • Stream types – not audio zones • Still limited to one sink • Limited number of sources **Remark**: This can be addressed with the new multi-zone concept that started since Android 9.	**Source Management**		High	High			Low			
12	Audio Focus • doesn't forbid to interrupt Audio, Android 10 provides additional interface for automotive applications to solve this problem • Audio focus can be lost at any time Topic relates to slip and relying on external Audio system. Question to Android 11, if it provides more control ? relates to control priorities	**Source Management**	++	Low	Medium			Low			
13	Bluetooth: patches for handsfree management, combination of BT stream with other streams, how to enable the correct audio routing, documentation on the Audio HAL is missing, it is difficult to find examples on line **Remark:** This can be related to the current POC. The BT can be mixed inside or outside of Android Automotive to check what are the shortcomings of both strategies. The need for low latency BT should also be considered in the POC How will it be addressed? Telephony, latency, control messages to HW acceleration?/ routing DSP. **HW DEP** (generic interface of HW) (can be done in SW / adv disadvantage power consumption efficiency, latency)	**Multi-Source Management**	++++	Low	Medium			Low			
14	*?* *Android does not implement all features required by customer.* **(bullet points below will be distributed over the table)** • Limited number of priorities -2 • No priorities of phone call types. **Remark:** It seems that the Android Automotive team is addressing this. In Android 10 there is a matrix to control the priories. For now it's hard-coded but it might get more flexibility. relates to Audio Focus	**Multi-Source Management**	++	High	High			-			

15	Safety Audio Android Audio subsystem is developed only for infotainment purposes. Safety-related features need to be implemented in another RTOS • Need to share the same hardware between 2 OSes • Running Android as a virtual machine inside RTOS leads to problems with scheduling of audio processes **Remark:** This can be tackled in the POC. This going towards the other extreme of making a system outside of Android to control safety-related features. It should be address in the POC as well. extracting raw streams helps with this point (handing over also relates to this)	**Multi-OS Integration**	-	High	Medium			Medium			
16	Modifying AOSP - goal to avoid changes Android way of extending its functionality is developing vendor extensions without modification of the framework. • But in some cases you need to modify framework. There is a process for this, but slow. And changes needed to one customer / developer is not needed for another. Even if you do this, you have to wait until Tier 1 merges this change to its drop • You have to retain compatibility for third-party apps	**Extensions**	-	Medium	Medium			Low			