Public

Android Audio HAL

Audio overview

Senior Software Engineer Piotr Krawczyk

tieto

About

Me:

- Senior Software Engineer in Tieto;
- 7 years of experience with Android platform;
- 3 years of experience with Android Automotive;

Tieto Product Development Services

Building the Connected world

Cloud Born

Hybrid & Multi vendor, NFVi, Cloud Native

Dynamic

Software defined Infrastructure, Zero Touch

Ever reaching

5G NR, HetNet, NB-IoT, Cloud RAN

Autonomous

Orchestration, Automation, Slicing, Open source

Enabling Tomorrow's Mobility

Connected

Telematics, V2X, Connected car

Safe

ADAS, ISO 26262

Aware

HERE technologies, Location based services

As a service

Fleet management, Mobility Services, Use based insurance

Innovating for Smart Living

Mobile

Platforms, Apps, AI, Multimedia

Fun

Multimedia, Multiroom Audio, Smart TV apps

Cozy

Smart Home, Homekit, Alexa, Google Home

Healthy

Wearable, DevOPS, Automated

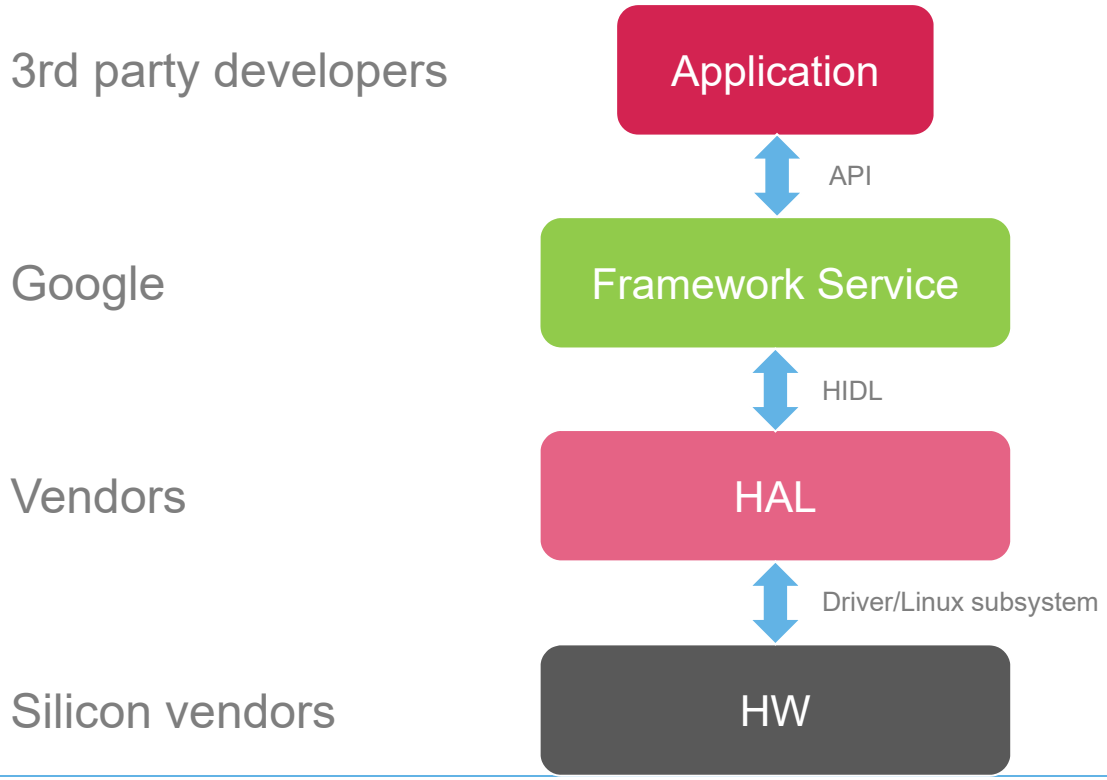
Agenda

- Audio in Android – overview;
- Integration challenges;
- Possible improvements;
- Impact of Android 10;

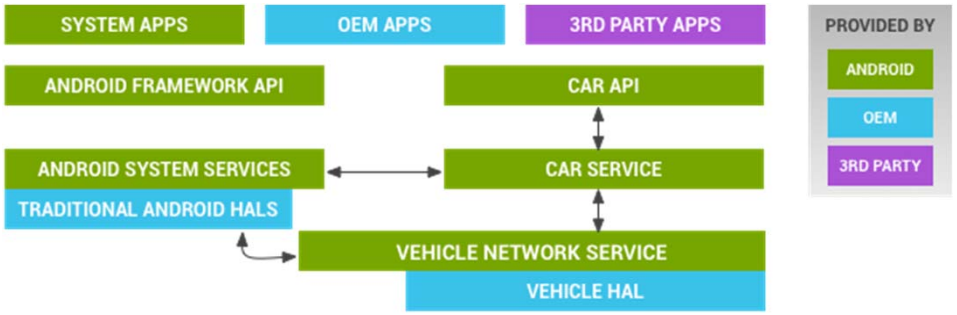
Audio overview

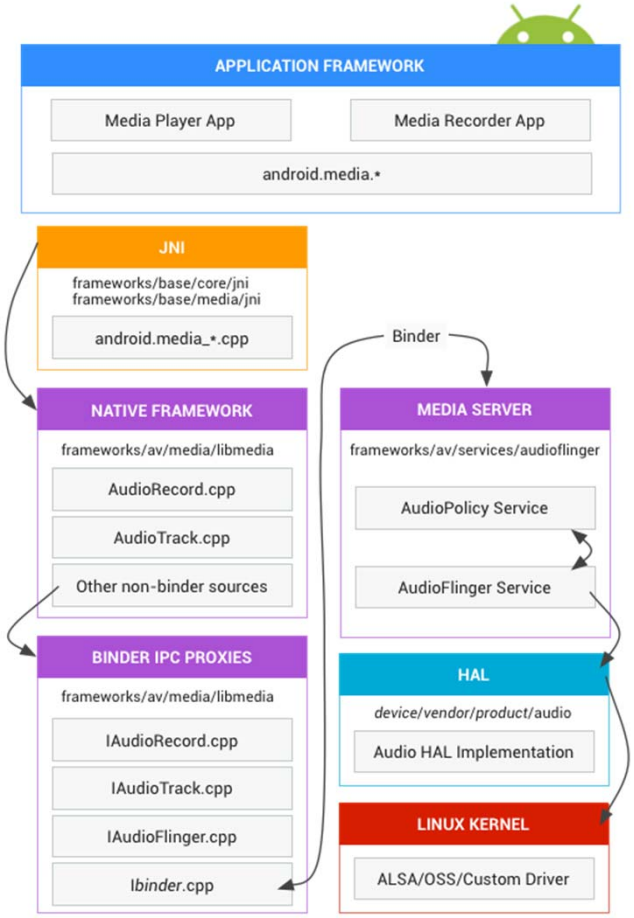
tieto

Typical Android flow

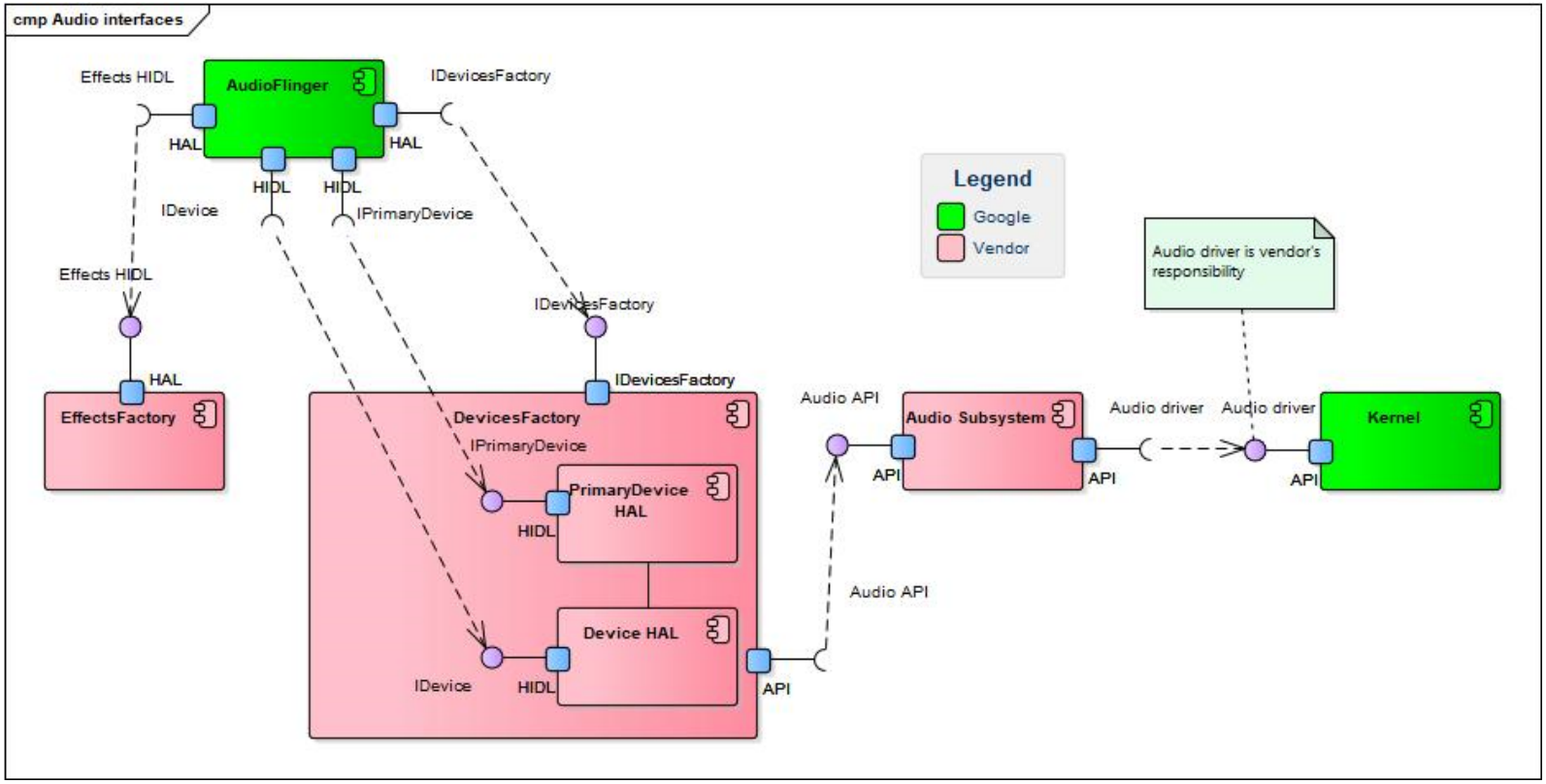


Audio in Android





Audio HAL interfaces



Audio HIDL

- **IDevice** – represents Audio HW module (e.g. primary, USB, A2DP);
- **IDevicesFactory** – connect to one of the Audio HW modules;
- **IPrimaryDevice** – interface for primary Audio HW module, extends IDevice;
- **IStream** – controls audio streams;
- **IStreamIn** – specialization for input streams;
- **IStreamOut** – specialization for output streams;
- **IAudioControl** - interacts with the car's audio subsystem to manage audio sources and volumes.

Audio Effects

Control effect lifecycle:

- **IEffectsFactory**

Generic effect interface:

- **IEffect**

Effect specializations (defined by Google):

- **IAcousticEchoCancelerEffect**
- **IAutomaticGainControlEffect**
- **IBassBoostEffect**
- **IDownmixEffect**
- **IEffectBufferProviderCallback**
- **IEnvironmentalReverbEffect**
- **IEqualizerEffect**
- **ILoudnessEnhancerEffect**
- **INoiseSuppressionEffect**
- **IPresetReverbEffect**
- **IVirtualizerEffect**
- **IVisualizerEffect**

tieto

Integration

tieto

Configuration challenge

Configurable:

- attached audio output/input devices,
- audio effects,
- audio codecs configuration,
- audio hardware paths,
- audio features (CDD),
- default sounds.

Methods:

- global settings (for all processes),
- user / profile settings,
- car variant specific,
- static vs dynamic.

Related files:

- audio_policy_configuration.xml (bus address definition),
- AudioControl.cpp (sContextToBusMap),
- car_volume_groups.xml (mapping of buses to volume groups),
- config.xml (audioUseDynamicRouting),
- audio_effects.[xml|conf],
- media_profiles_*.xml,
- media_codecs_*.xml,
- other.

What if framework needs to be changed?

- **Overlays**
  - Resource overlay – configuration changes;
  - Fork application – LOCAL_OVERRIDES_PACKAGES (Android.mk);
- **Manual changes in framework**
  - Keep Treble compliance – keep API, keep HIDL;
  - Adding new interfaces is allowed, keep ABI compliance.

Places for improvements

tieto

Common solution

Challenge: Audio HALs are provided by silicon vendors or 3rd party suppliers.

Improvement: create common Audio HAL that will work on different platforms.

Why:

- common usage of TinyALSA as audio subsystem, ALSA and OSS is rarely used;
- HIDL interfaces guaranteed by VTS.

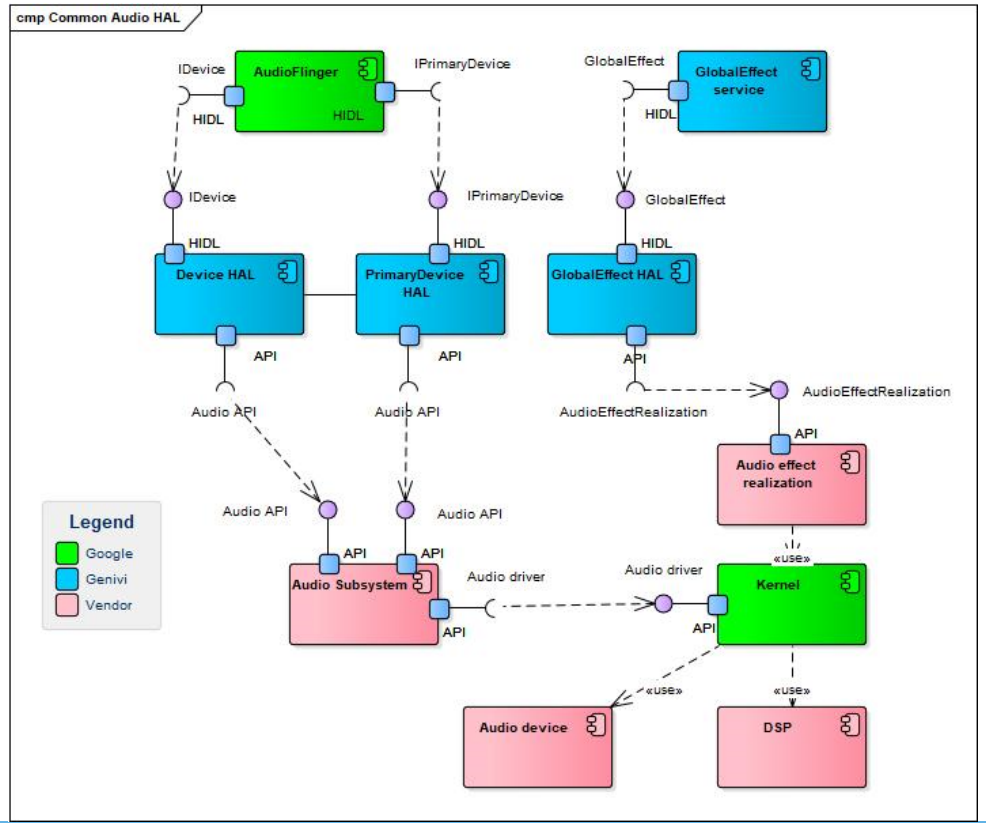
Limitations:

- HW dependency for audio effects,

Common solution benefits

- Add new features to Audio HAL – shared between platforms, e.g.
  - Global effect handling;
  - Speed dependent volume control;
  - Audio HW module direct communication;
- End user tool portability between platforms, e.g.
  - Audio calibration tool;
- Configuration helpers;
- Faster integration – short time to market;
- Less dependency to external deliveries.

Common HAL idea



tieto

Configuration helper

Challenge: Configuration is scattered, poorly documented, difficult to maintain.

Improvement: create set of configuration helpers tool.

Why:

- common usage of TinyALSA as audio subsystem, ALSA and OSS is rarely used,
- verification checks can assure dependency between configuration modules.

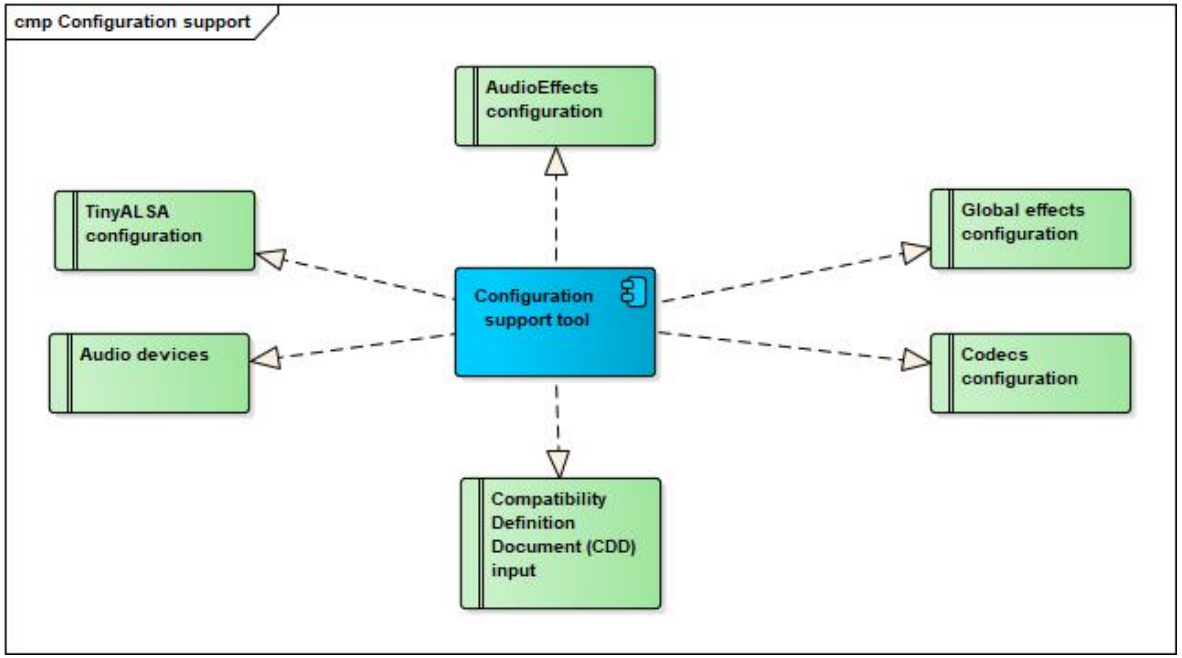
Limitations:

- Track changes between Android updates.

Configuration helper benefits

- Automatic dependency check between configuration modules;
- Faster bring-up by inheriting configuration from other platforms;
- Easier comparison of products;
- Easier control of features;
- Better visibility on product configuration.

Configuration helper idea



tieto

Public

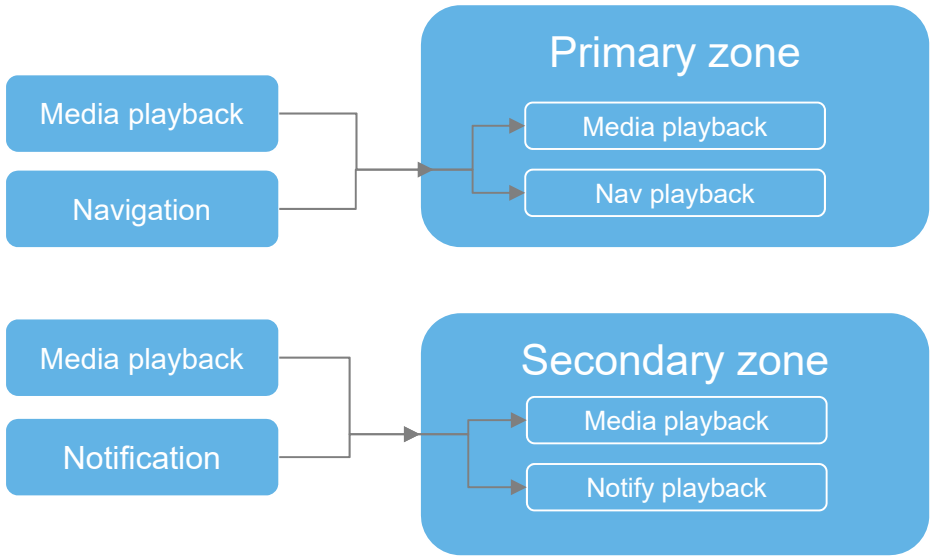
Android 10

New features, possible impact

tieto

New in Android 10

- Multi-Zone



- Application can be played in zone
- Zone audio contains devices
- Zones have separate volume
- Zone can be requested by app
- In future zone could be binded with display

New in Android 10

CarAudioFocus interaction matrix

- Row selected by playing sound (labels along the right)
- Column selected by incoming request (labels along the top)

R – reject
E – exclusive
C – concurrent

	Music	Nav	Voice	Ring	Call	Alarm	Notification	System
Music	E	C	E	E	E	E	C	E
Nav	C	C	E	C	E	C	C	C
Voice	C	R	C	E	E	R	R	R
Ring	R	C	C	C	C	R	R	C
Context	R	C	R	C	C	C	C	R
Alarm	C	C	E	E	E	C	C	C
Notification	C	C	E	E	E	C	C	C
System	C	C	E	E	E	C	C	C