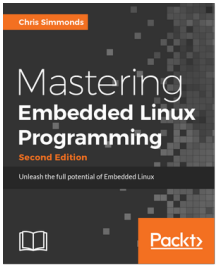
# An Introduction to Android Automotive OS

Chris Simmonds

Embedded World 2021

# License

# About Chris Simmonds




- Consultant and trainer
- Author of *Mastering Embedded Linux Programming*
- Working with embedded Linux since 1999
- Android since 2009
- Speaker at many conferences and workshops

"Looking after the Inner Penguin" blog at `https://2net.co.uk/`

@2net_software

`https://uk.linkedin.com/in/chrisdsimmonds/`

# Google and me

- I have no direct contact with Google
- I do not represent Google's point of view
- I have not signed any NDAs with Google

# Agenda

- Android and automotive
- Vehicle HAL
- Car service
- Exterior cameras
- Audio
- Conclusion


The Polestar 2 is the first vehicle with Android Automotive OS

# Android and IVI

- 2014: **Android Auto**
  - https://www.android.com/auto/
  - Screen cast from smart phone to head unit display
  - An SDK integrated into the head unit (which is usually not running Android)
  - Apple CarPlay is a similar concept
- 2017: **Android Automotive OS**
  - https://source.android.com/devices/automotive/
  - Android running in the head unit

Android has been used in IVI for a long time, e.g. Honda (based on JB 4.2) and Hyundai (based on GB 2.3).

# The Android Open Source Project

- The core of Android is developed and released as the **Android Open Source Project** (AOSP)
- Android Automotive OS is part of AOSP
- But, AOSP is not a production-ready solution
- you need front-end apps, a home screen, back-end services
- Google has a solution ...

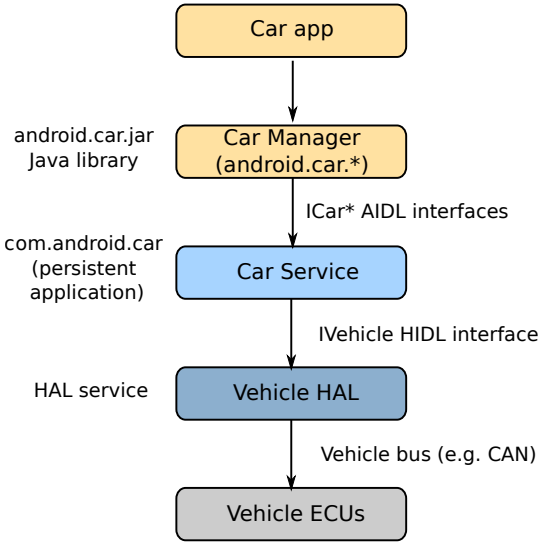
# Google Automotive Services (GAS)

- Non-free services on top of Android Automotive
  - (i.e. GMS for Automotive)
- Includes
  - Play Store
  - Google Assistant
  - Google Maps
- Need to pass the Automotive Test Suite (ATS)
  - Minimum hardware requirements: 4 GiB RAM; 32 GiB storage
  - Must enable Assistant

# No GAS

- Without GAS, you need to find alternative apps and services
  - typically a combination of in-house and third party
  - some tier one companies have SDKs that you can use

# Architecture of Android Automotive



- Android and automotive
- Vehicle HAL
- Car service
- Exterior cameras
- Audio
- Conclusion

# Vehicle HAL

- The Vehicle HAL stores information as **Vehicle Properties**
- Most properties are linked to signals on the vehicle bus, for example:
  - speed: a float value in metres per second
  - heating control setting: a float value in degrees Celsius
- Properties may be changed
  - by the signal changing on the bus
  - programmatically from an Android application
- The Vehicle HAL has an interface named **IVehicle**

# System Property Identifiers

- **System** property identifiers are marked with `VehiclePropertyGroup:SYSTEM`
- In Android 11 there are over 130, for example:

```
enum VehicleProperty: int32_t {
    /**
     * HVAC, target temperature set.
     *
     * @change_mode VehiclePropertyChangeMode:ON_CHANGE
     * @access VehiclePropertyAccess:READ_WRITE
     * @unit VehicleUnit:CELSIUS
     */
    HVAC_TEMPERATURE_SET = (
        0x0503
        | VehiclePropertyGroup:SYSTEM
        | VehiclePropertyType:FLOAT
        | VehicleArea:SEAT),
```

Code: `hardware/interfaces/automotive/vehicle/2.0/types.hal`

# IVehicle

- Functions exposed by the Vehicle HAL
  - `getAllPropConfigs()`
  - `getPropConfigs(props)`
  - `get(VehiclePropValue)`
  - `set(VehiclePropValue)`
  - `subscribe(IVehicleCallback, SubscribeOptions)`
  - `unsubscribe(IVehicleCallback, propId)`

Code: `hardware/interfaces/automotive/vehicle/2.0/IVehicle.hal`

- Android and automotive
- Vehicle HAL
- Car service
- Exterior cameras
- Audio
- Conclusion

# Car service

- Wraps Vehicle Properties and presents them as a number of APIs useful to applications
- Implemented as a system service in a **persistent**, **system** app named **com.android.car**
- Service name is `car_service`
- Interface `android.car.ICar`
- Dump command `dumpsys car_service`
  - `-h` for a list of options

Code: `packages/services/Car/service`

# Car Manager

- In Android, the API to a service is implemented as a **manager**
- Car Manager consists of the android.car.* classes, which form the API for Android Automotive
  - https://developer.android.com/reference/android/car/classes
- Car Manager is a platform library which is installed on the device in `/system/framework/android.car.jar`

Code for Car Manager: `packages/services/Car/car-lib`

# Car Manager

- Car Manager provides 23 interfaces:

CAR_INPUT_SERVICE
INFO_SERVICE
APP_FOCUS_SERVICE
PACKAGE_SERVICE
AUDIO_SERVICE
CAR_NAVIGATION_SERVICE
CAR_OCCUPANT_ZONE_SERVICE
CAR_INSTRUMENT_CLUSTER_SERVICE
DIAGNOSTIC_SERVICE
CAR_TRUST_AGENT_ENROLLMENT_SERVICE
CAR_WATCHDOG_SERVICE
POWER_SERVICE
PROPERTY_SERVICE
PROJECTION_SERVICE
BLUETOOTH_SERVICE
TEST_SERVICE
CAR_DRIVING_STATE_SERVICE
CAR_UX_RESTRICTION_SERVICE
OCCUPANT_AWARENESS_SERVICE
CAR_CONFIGURATION_SERVICE
CAR_MEDIA_SERVICE
CAR_BUGREPORT_SERVICE
STORAGE_MONITORING_SERVICE

The next slides expand on just a few of these: PROPERTY_SERVICE, INFO_SERVICE, and CAR_UX_RESTRICTION_SERVICE

# Digression: Android permissions

- Applications need to be granted **permissions** to access services
- Car Service has only a few that can be granted to 3rd party apps

```
CAR_INFO
READ_CAR_DISPLAY_UNITS
CONTROL_CAR_DISPLAY_UNITS
CAR_ENERGY_PORTS
CAR_EXTERIOR_ENVIRONMENT
CAR_POWERTRAIN
CAR_SPEED
CAR_ENERGY
```

- The others are marked as **signature** | **privileged**
  - which are only granted to apps built by the OEM and shipped as part of the platform

# PROPERTY_SERVICE (CarPropertyManager)

- A simple wrapper for Vehicle HAL properties, has methods to enumerate, get, set and listen to any property
- Permissions are checked per property
  - e.g. to access vendor properties, apps need `PERMISSION_VENDOR_EXTENSION`, level "signature|privileged"

Code: `packages/services/Car/car-lib/src/android/car/hardware/property/CarPropertyManager.java`

# INFO_SERVICE (CarInfoManager)

- Retrieves various static information from the car (VID, model, year, fuel type, etc.)
- Permission `PERMISSION_CAR_INFO`, level "normal"

Code:
`packages/services/Car/car-lib/src/android/car/CarInfoManager.java`

# CAR_UX_RESTRICTION_SERVICE (CarUxRestrictionsManager)

- Indicates whether there is a requirement to be Distraction Optimized. Uses information from CarDrivingStateManager

Code: `packages/services/Car/car-lib/src/android/car/drivingstate/CarUxRestrictionsManager.java`

# Car apps

- Demo apps are in `packages/apps/Car/*` and `packages/services/Car/*`
- Examples:

| Name | description |
|---|---|
| CarLauncher | Car home screen |
| CarHvacApp | Heating, ventilation and A/C |
| CarRadioApp | Radio |
| CarDialerApp | Car dialer |
| CarMapsPlaceholder | Navigation would go here! |
| LocalMediaPlayer | Media player |
| CarMessengerApp | Messages and notifications |
| CarSettings | Settings |
| EmbeddedKitchenSinkApp | Lots of demos! |

# Third party apps

- Apps in Play Store for Auto and Automotive can only access a subset of the Car Manager APIs
- They are orientated towards media (audio) and messaging
- Written to minimize driver distraction and tested by Google

*"Important: Google takes driver distraction very seriously. Your app must meet specific design requirements before it can be listed on Google Play for Android Automotive OS and Android Auto"*

`https://developer.android.com/training/cars/start`

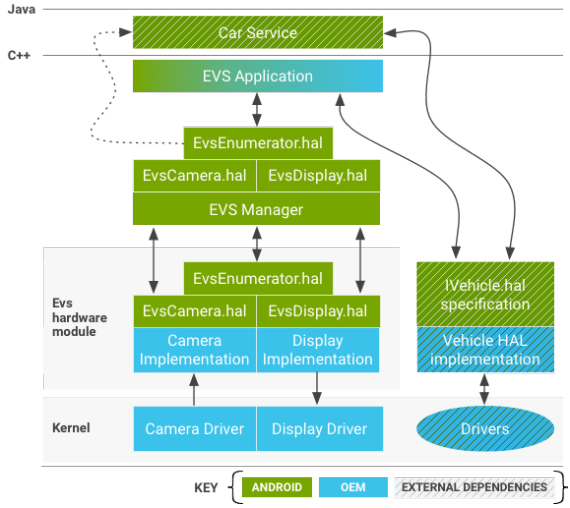
`https://developer.android.com/docs/quality-guidelines/car-app-quality`

- Android and automotive
- Vehicle HAL
- Car service
- Exterior cameras
- Audio
- Conclusion

# Exterior cameras

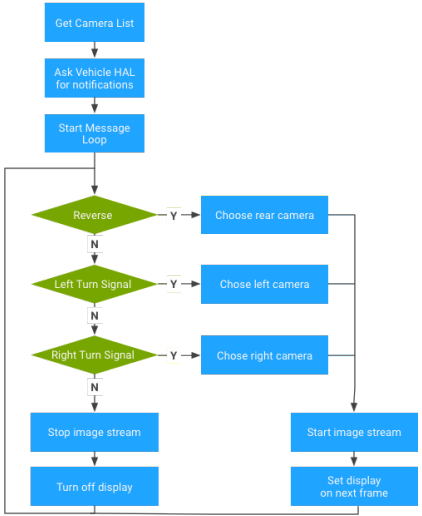
- Problem: the rear view camera must be able to display images within 2 seconds of starting the ignition
- But, Android takes 10's of seconds to boot
- Solution: the **Exterior View System** (EVS)
  - EVS is a self contained application written in C++
  - has few dependencies on the Android operating system
  - so, EVS can be active within 2 seconds, long before Android has finished booting

# Architecture




Reference: https://source.android.com/devices/automotive/camera-hal

# Typical control flow




Reference: `https://source.android.com/devices/automotive/camera-hal`

# Display sharing

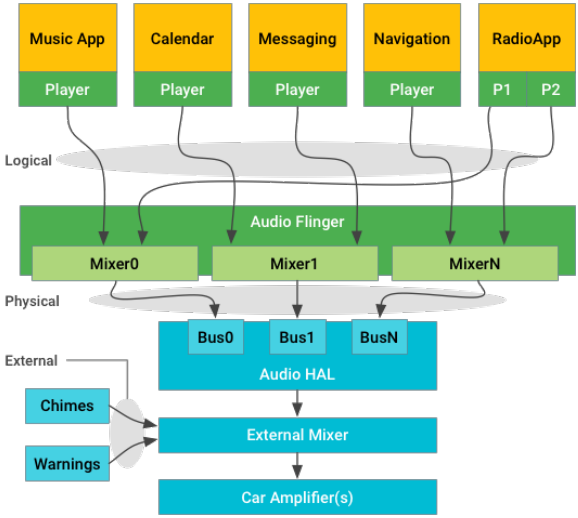
- EVS has priority over the main display (usually the centre console)
- It can grab the display whenever an exterior camera needs to be shown
  - e.g. when reverse gear is selected
- There is no mechanism that allows EVS and Android to display content at the same time

- Android and automotive
- Vehicle HAL
- Car service
- Exterior cameras
- Audio
- Conclusion

# What is special about audio in vehicles?

- Many audio channels with special behaviours
- Critical chimes and warning sounds
- Interactions between audio channels
- Lots of speakers

# Automotive sounds and streams




Reference https://source.android.com/devices/automotive/audio

# Audio contexts

| | |
|---|---|
| MUSIC | Music playback |
| NAVIGATION | Navigation directions |
| VOICE_COMMAND | Voice command session |
| CALL_RING | Voice call ringing |
| CALL | Voice call |
| ALARM | Alarm sound from Android |
| NOTIFICATION | Notifications |
| SYSTEM_SOUND | User interaction sounds (button clicks, etc) |

# Physical streams, contexts and buses

- AudioFlinger uses the **context** to mix logical streams down to to **physical** streams called a **buses**
- Many to one: several logical streams may be mixed into one bus
- `IAudioControl::getBusForContext` maps from context to bus
- A bus is an output channel, typically fed to the car mixer/amplifier
  - For example, the NAVIGATION context could be routed to driver's side speakers

# Chimes and warnings

- Regulatory chimes and warnings **are not played through Android**
  - Android does not have an early audio path
  - Android is not a safety critical operating system
- Regulatory sounds must be generated outside Android and mixed later in the output chain

- Android and automotive
- Vehicle HAL
- Car service
- Exterior cameras
- Audio
- Conclusion

# Conclusion

- Android Automotive is Android adapted for the car
- New VHAL, Car Service, and Car Manager
- New services for external cameras
- Additions to audio, including zones (buses) and context based routing

Slides at `https://2net.co.uk/slides/EW21/introduction-to-aaos-csimmonds-ew-2020.pdf`

Embedded Android+Automotive: a 5-day deep dive into Android Automotive
`https://2net.co.uk/training/embedded-android-automotive`

Questions?