

# Android Automotive OS

*A short introduction to  
Google's AAOS*

**Anna-Lena Marx**

*March 12th, 2024 · Munich*



VOL  
AUDIO

# Anna-Lena Marx



[Anna-Lena Marx](#)



[anna-lena.marx@inovex.de](mailto:anna-lena.marx@inovex.de)



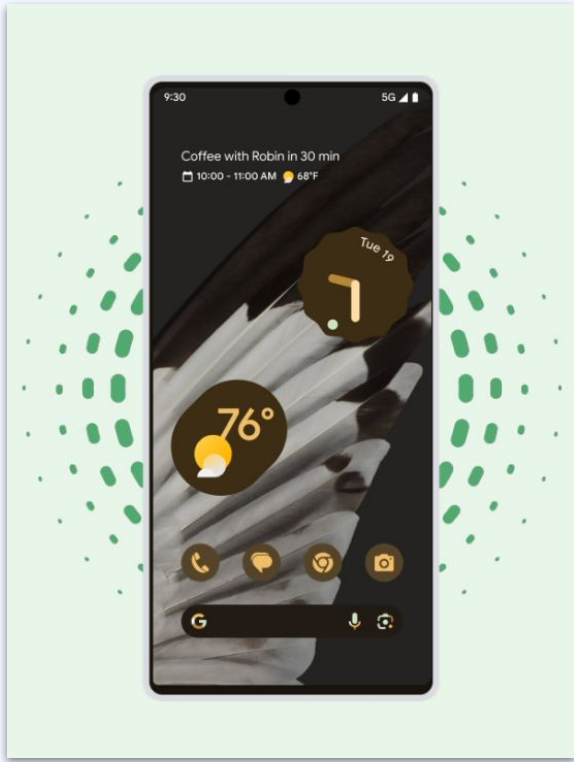
[marx.engineer](#)

## Embedded Systems Developer

- since 2015 with inovex
- has a Master's degree in Embedded Systems
- studies Electrical Engineering as a hobby

## Main Topics

- Embedded Systems
- Yocto Linux
- Linux Kernel
- AOSP/AAOS
- IoT



# Android

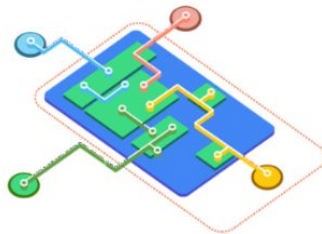


# Android Open Source Project

Android unites the world. Use the open source Android operating system to power your device.

Get source

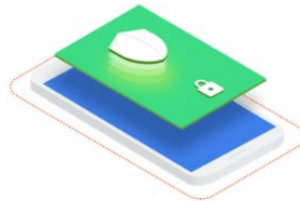
Search



## Interfaces and architecture

Learn how the pieces fit together, from the kernel to the HALs to updatable system components.

Understand architecture



## Android security is essential

Find out how the Android security program works and learn how to implement the latest features.

Implement security



## Design compatible devices

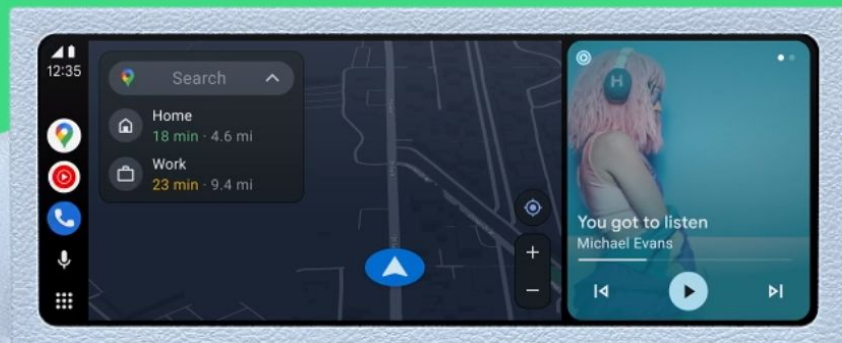
Offer consistent experiences across different Android-powered devices for users and app developers.

Test devices

Android Auto

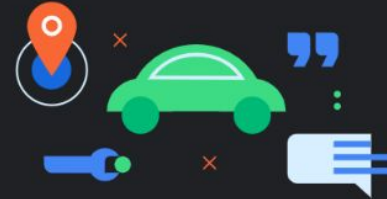
# On your car display.

Get Started



# Android Automotive

Android Automotive is a full-stack, open source, highly customizable platform running directly on in-vehicle hardware.



[Learn More](#)



## Guidelines for Development

Learn about our extensive ecosystem of guidelines specific to the development of Automotive apps.



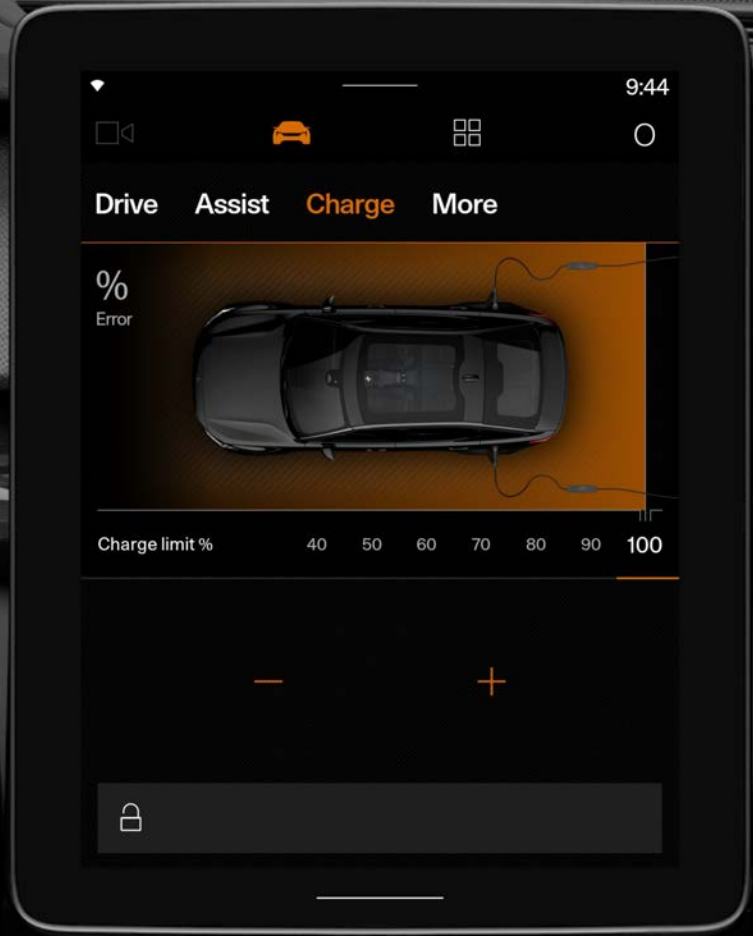
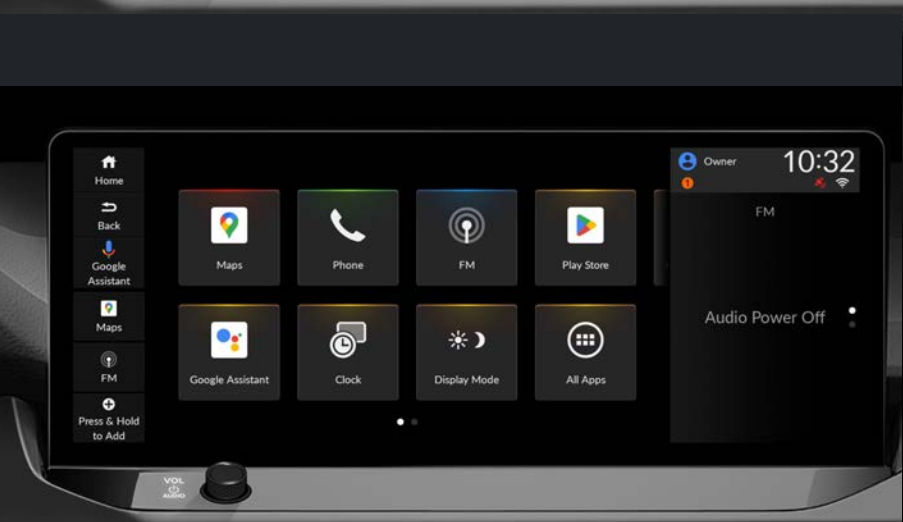
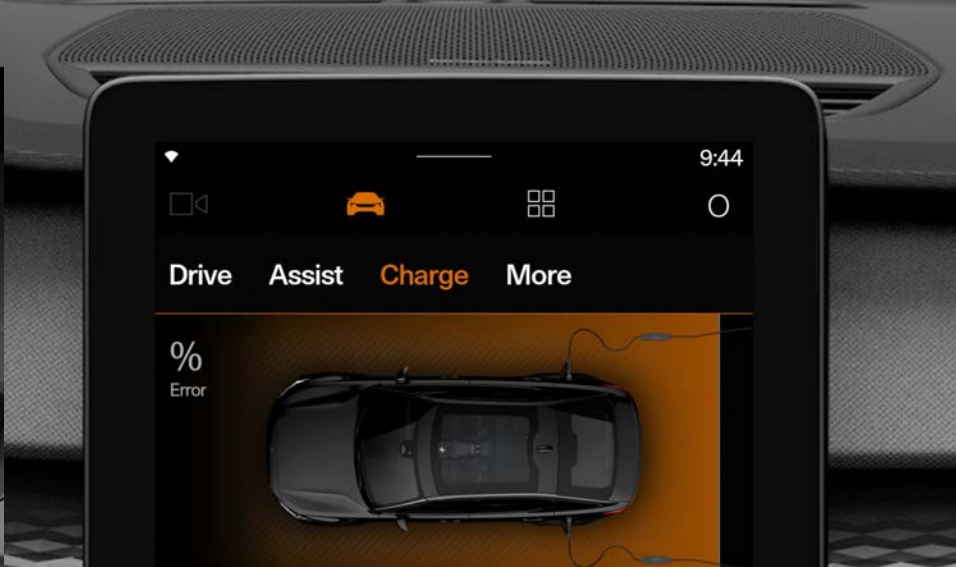
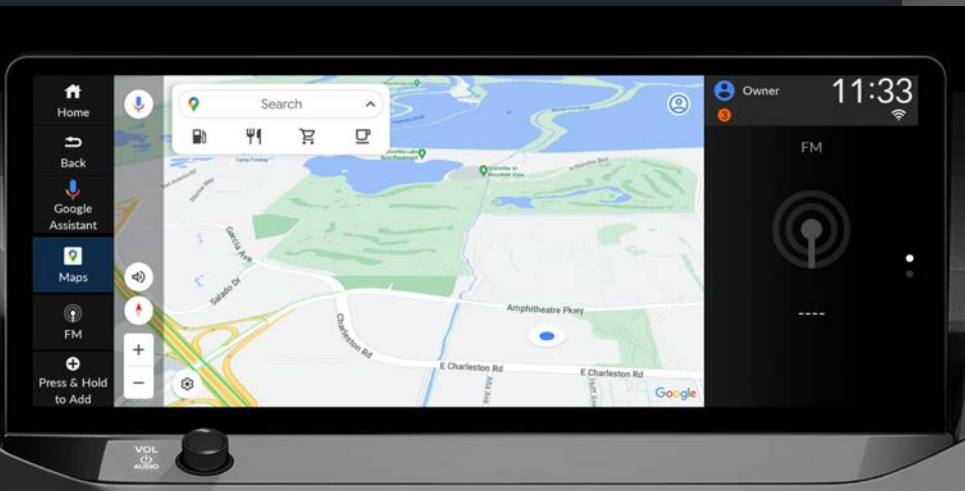
## Development Tools

Read about the tools we provide to support your development of AAOS-based apps.



## Testing Tools

See the scalable infrastructure and robust set of testing tools we provide so you can maximize your efficiency and ensure compliance.

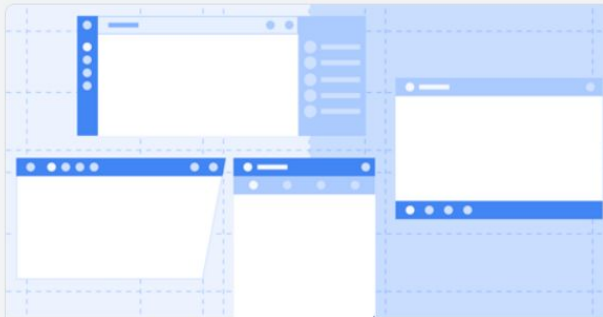




## Android Auto

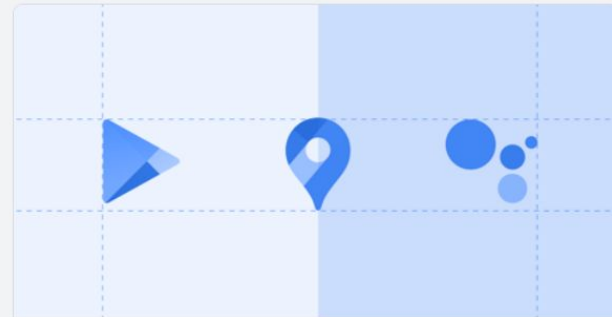
Android Auto provides users of Android phones with an app experience designed for cars. If a car's head unit supports Android Auto, users can access apps directly on their car's display by connecting their phone.

App developers can support Android Auto by adding services to their phone apps. Android Auto then uses those services to display an interface designed to work well on the screen configurations of all compatible cars.



## Android Automotive OS (AAOS)

AAOS is an infotainment system built into vehicles by car makers. Android apps that support AAOS requirements can be downloaded from the Google Play Store to any car manufactured by a Google Automotive Services (GAS) partner. Instead of using a particular phone app, users install a version of the app that is designed for cars directly onto the head unit.



## Google Automotive Services (GAS)

Car makers who wish to provide Google services such as Maps, Play Store, Assistant, and so on can work directly with Google to support GAS. GAS consists of software built on top of Android Automotive OS for the purpose of delivering such services.



# The Android Open Source Project

Two levels of compatibility:

## AOSP compatibility

defined by  
Compatibility Definition Document (CDD)

## Android compatibility

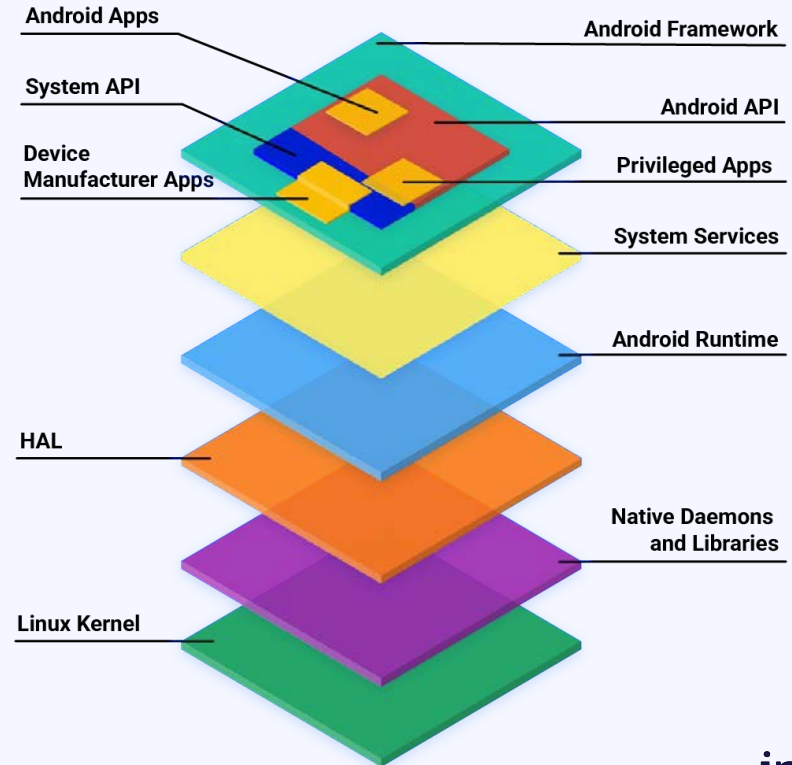
CDD plus

- Vendor Software Requirements (VSR)
- Vendor Test Suite (VTS)
- Compatibility Test Suite (CTS)

Additional step:

## Licensing Google Mobile Services (GMS)

- Google Apps (Youtube, Maps, Gmail etc.)



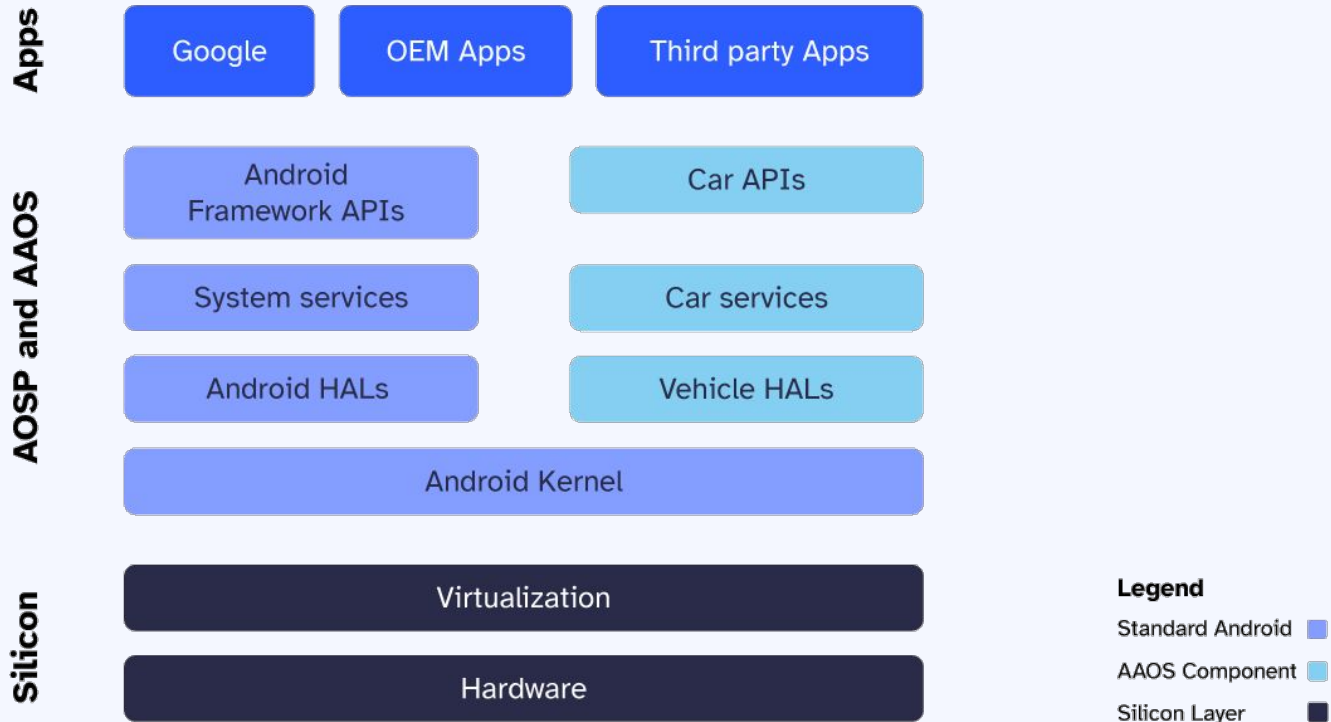
**And where does  
the car come in?**



VOL  
AUDIO



# It's already there!

































## AAOS

Source code for [Android Automotive OS](#).

## Structure

[Load more](#)

### Files and Directories

- |   |   |   |   |  |  |
|---|---|---|---|--|--|
|  FrameworkPackageStubs/ |  car-evs-helper-lib/ |  car-maps-placeholder/ |  cpp/            |  packages/          |  tests/                   |
|  apex_car_framework/    |  car-helper-lib/     |  car-test-lib/         |  data/           |  procsfs-inspector/ |  tools/                   |
|  car-admin-ui-lib/      |  car-lib/            |  car-usb-handler/      |  experimental/   |  service/           |  vehicle-hal-support-lib/ |
|  car-builtin-lib/       |  car-lib-module/     |  car_product/          |  obd2-lib/       |  service-builtin/   |  |
|  .clang-format          |  Android.mk          |  CleanSpec.mk          |  OWNERS_networking |  README.md          |  |
|  .gitignore             |  CPPLINT.cfg         |  OWNERS                |  PREUPLOAD.cfg     |  TEST_MAPPING       |  |

# Structure

car_product/	- AAOS product
car-builitn-lib/	- A helper library for CarService to access hidden framework APIs
car-lib/	- Car API
car-lib-module/	- Car API module
cpp/	- Native services
experimental/	- Experimental Car API and services
packages/	- Apps and services for cars
service/	- Car service module
service-builitn	- Platform builitn component that runs CarService module
tests/	- Tests and sample apps
tools/	- Helper scripts

## C++

Native (C++) code format is required to be compatible with .clang-format file. The formatter is already integrated to repo tool. To run manually, use:

```
git clang-format --style=file --extension='h,cpp,cc' HEAD~
```

Note that clang-format is *not* desirable for Android java files. Therefore the command line above is limited to specific extensions.

## Debugging CarService

Dumpsys and car shell can be useful when debugging CarService integration issues.

### dumpsys

```
adb shell dumpsys car_service # to dump all car service information
adb shell dumpsys car_service --services [service name] # to dump a specific service information
adb shell dumpsys car_service --list # get list of available services
```

Dumpsys for CarService includes the following (more information is available in dumpsys, below are just highlights):

Repository root

- ▾ interfaces
  - apexkey
  - atrace
  - audio
  - authsecret
  - ▾ **automotive**
    - audiocontrol
    - can
    - evs
    - ivn\_android\_device
    - occupant\_awareness
    - remoteaccess
    - sv
    - vehicle
    - 📄 OWNERS
    - 📄 README.md
    - 📄 TEST\_MAPPING

## Automotive HALs

### Overview:

The automotive HAL tree is used by Android Automotive to discover and operate hardware specific to a car. The HALs are not (yet) frozen, as the HAL definition is expected to evolve between Android releases.

### Files and Directories

- |                 |                       |                       |            |
|-----------------|-----------------------|-----------------------|------------|
| 📁 audiocontrol/ | 📁 evs/                | 📁 occupant_awareness/ | 📁 sv/      |
| 📁 can/          | 📁 ivn_android_device/ | 📁 remoteaccess/       | 📁 vehicle/ |
| 📄 OWNERS        | 📄 README.md           | 📄 TEST_MAPPING        |            |

Files Outline

<|

1.0

Links ▾

Repository root

▾ interfaces

▸ apexkey

▸ atrace

▸ audio

▸ authsecret

▾ automotive

▸ audiocontrol

▾ can

▾ 1.0

▸ default

▸ hidl-utils

▸ tools

▸ vts

📄 Android.bp

📄 ICanBus.hal

📄 ICanController.hal

📄 ICanErrorListener.hal

^ Files and Directories

📁 default/

📁 hidl-utils/

📁 tools/

📁 vts/

📄 Android.bp

📄 ICanController.hal

📄 ICanMessageListener.hal

📄 types.hal

📄 ICanBus.hal

📄 ICanErrorListener.hal

📄 ICloseHandle.hal

**Why is Automotive  
interesting for  
Google?**



VOL  
AUDIO

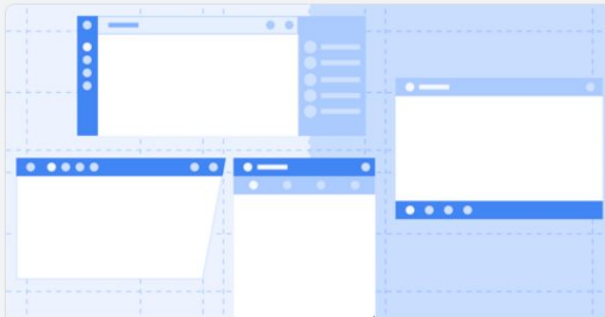




## Android Auto

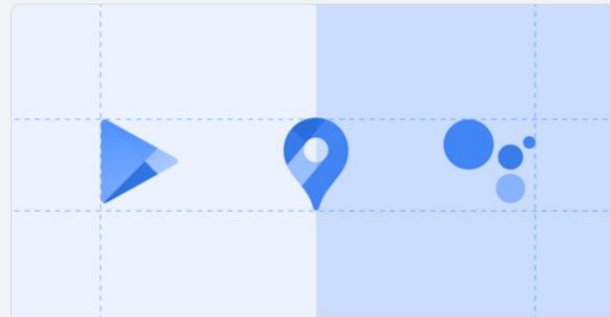
Android Auto provides users of Android phones with an app experience designed for cars. If a car's head unit supports Android Auto, users can access apps directly on their car's display by connecting their phone.

App developers can support Android Auto by adding services to their phone apps. Android Auto then uses those services to display an interface designed to work well on the screen configurations of all compatible cars.



## Android Automotive OS (AAOS)

AAOS is an infotainment system built into vehicles by car makers. Android apps that support AAOS requirements can be downloaded from the Google Play Store to any car manufactured by a Google Automotive Services (GAS) partner. Instead of using a particular phone app, users install a version of the app that is designed for cars directly onto the head unit.



## Google Automotive Services (GAS)

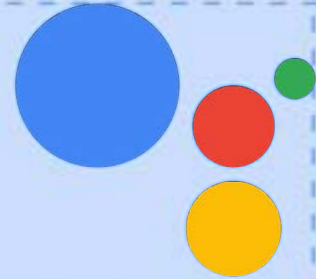
Car makers who wish to provide Google services such as Maps, Play Store, Assistant, and so on can work directly with Google to support GAS. GAS consists of software built on top of Android Automotive OS for the purpose of delivering such services.



Google Play



Google Maps



Google Assistant

**Why is AAOS  
interesting for  
car manufacturers?**



VOL  
AUDIO



**Android is an ECOSYSTEM**

# AAOS is a platform to build vehicles

## Matured code infrastructure of the AOSP

- graphics, video and media support
- connectivity stack
  - WiFi
  - Bluetooth
  - cellular
- security mechanisms
  - verified boot
  - SELinux
  - app isolation – secure runtime for untrusted apps
- updates mechanism

# AAOS is familiar to app developers and users

## Matured application development ecosystem

- an UI/UX concept people are already familiar with
  - platform focuses on UX!
- well-known app developer ecosystem
  - standardized **APIs** with a good abstraction level
  - extensive set of system and third-party **libraries**
  - pretty good and in depth **documentation**
  - large eco system of existing apps
  - lots of *good* Android **app developers**

## ... and it's open

### There are alternatives to the Google Automotive Services

- Automotive focused app stores



- Navigation providers



**Who is using  
Android  
Automotive?**



VOL  
AUDIO





# Android Automotive OS with Google Automotive Services



Polestar



RENAULT



# Only Android Automotive OS



C A R I A D



# Challenges



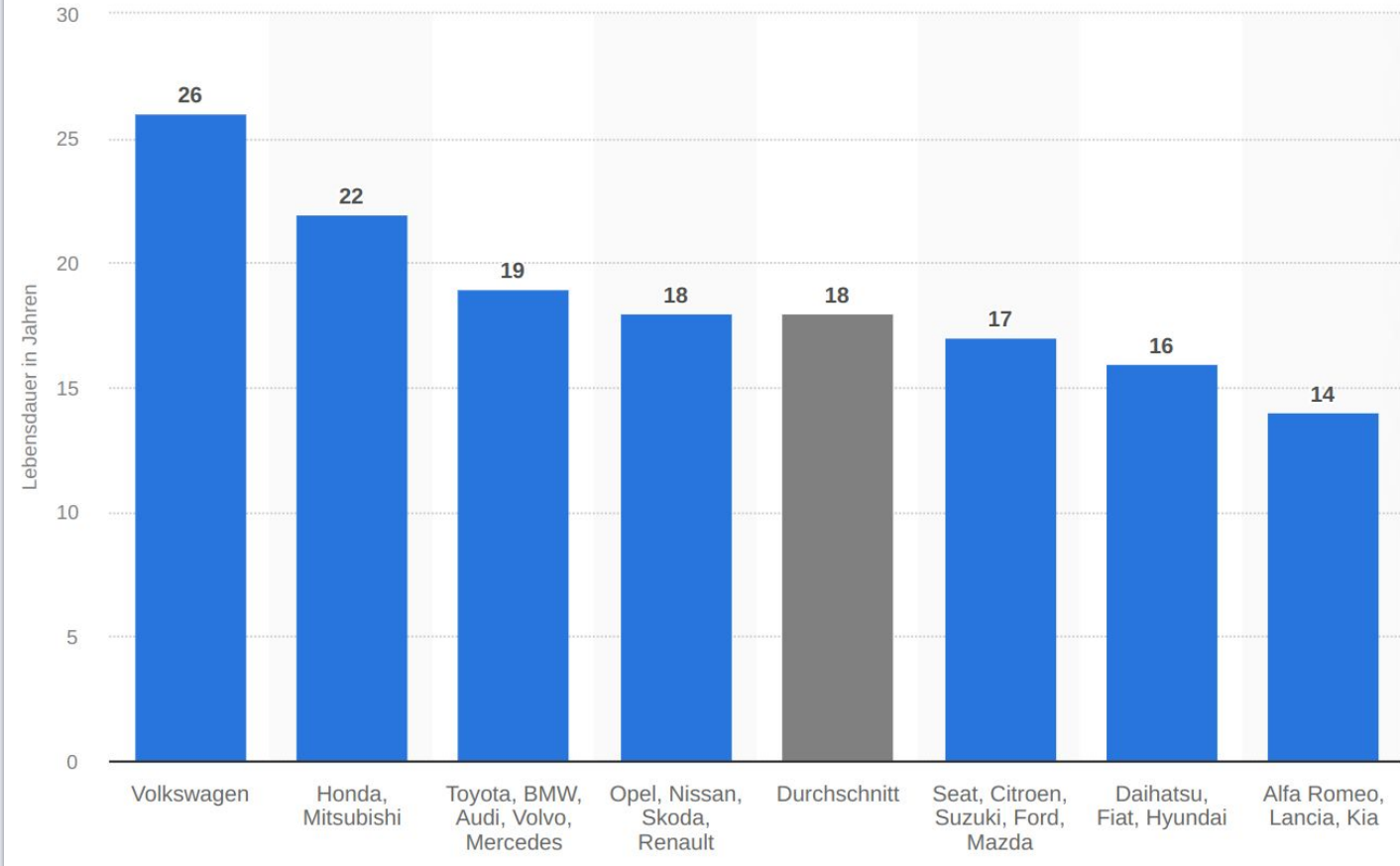
VOL  
AUDIO

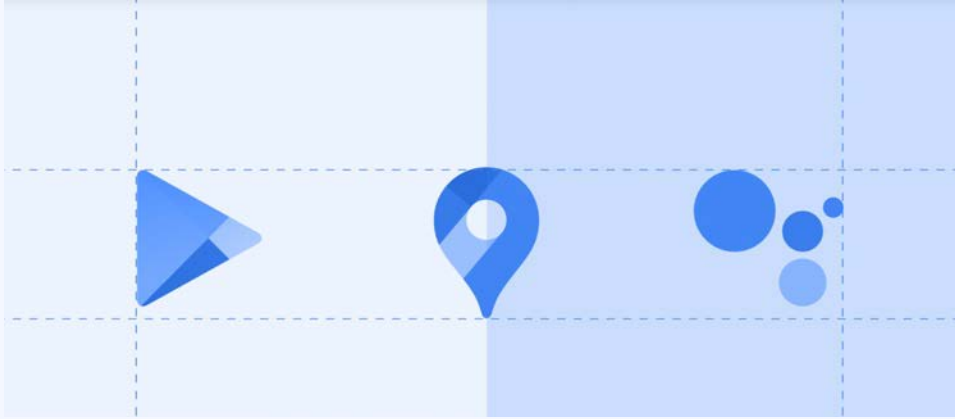


- lives 5-6 years as a maximum
- typical Google support range:  
~ 2-3 years

- average lifetime: 18 years

2014





**Google Automotive Services**



**vs.**

**Third-Party Providers**



**inovex**

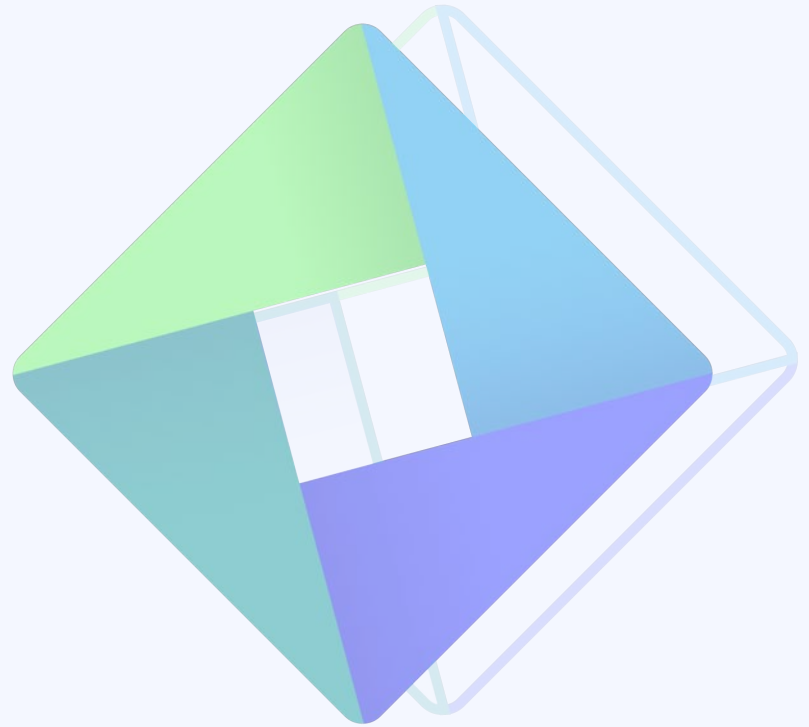
**And where do the  
necessary AOSP  
devs come from?**



VOL  
AUDIO



# Let's discuss!



**Anna-Lena Marx**  
*Embedded Systems Dev*

Ludwig-Erhard-Allee 6  
76131 Karlsruhe

[anna-lena.marx@inovex.de](mailto:anna-lena.marx@inovex.de)



## Further Reading

- [What is Android Automotive?](#)
- [A perspective on Android Automotive \(AAOS\) from an Android TV guy](#)
- [Android Automotive, the Real Android Fragmentation](#)
- [The state of Android Automotive in 2024](#)