AOSP App Framework Standardization Expert Group Status
AUTOMOTIVE AOSP APP FRAMEWORK STANDARDIZATION

EXPERT GROUP

OCTOBER 11TH, 2023
Antitrust Note Well

Before we begin, we would like to make clear that COVESA is committed to compliance with the antitrust laws in all of its activities, and that it expects all participants to similarly comply with the antitrust laws. We will not engage in--and members must refrain from--any discussion of, or understandings regarding competitively sensitive topics. If you have any doubts regarding whether a matter is appropriate for discussion, please consult with your antitrust counsel.
Agenda

OCTOBER 11TH, 2023
9:00AM – 9:45AM EST

COVESA
Accelerating the future of connected vehicles
Agenda

- Why is there the need for an AOSP App Framework Standardization Expert Group?
- What already happened on the topic at COVESIA
- Status update on ongoing work streams
- What’s Next
- Questions
Why is there the need for an AOSP App Framework Standardization Expert Group?
**Traditional Approach vs. Platform Ecosystem Approach.**

- Develop app once, deploy to many OEMs via an App Store.
- No need for OEM specific development work.

### Traditional Approach
- App Developer
- OEM A
- OEM B
- OEM C

### Platform Ecosystem Approach
- App Developer
- Store Provider
- OEM A
- OEM B
- OEM C

*COVESA*
With an AOSP-based App Store, OEMs support a relevant 3rd Party Ecosystem to emerge.

Platform ecosystems need to attract and coordinate two different target groups – app developers and OEM customers.

In order to grow the portfolio of the AOSP based App Store we need to focus on the App Providers as an important customer group.
Goals Automotive AOSP App Framework Standardization.

1. Enabling innovation and rich experiences.
2. Frictionless onboarding and testing for app developers.
4. Avoid fragmentation.
What already happened on the topic at COVESA
April 2023: AMM in Portugal

- Presentation of the initiative related to Automotive AOSP App Framework Standardization Expert Group to COVESA members
- Attendance of several OEMs, App Store Providers, Tier 1s and app developers
- Interest expressed by the attendees
- Jointly identified TOP3 topics:
  - Non-GAS Emulator (independent of OEM)
  - Push Notifications
  - Camera API
F2F Meeting June 2023: Munich

ACHIEVEMENTS

- Election of Chairs
  - BMW (Melina Mascolo)
  - General Motors (Richard Fernandes)
  - Forvia / Faurecia Aptoide Automotive (Camille Ghibaudo)

- Definition of main work streams
  - Camera API
  - Push Notifications
  - Cross-OEM emulator

- Presentation of potential reference implementation for In-Vehicle Camera Access
Status update on ongoing work streams
In-Vehicle Camera Access: Deliverable & Status

DELIVERABLE
1. Defining default in-vehicle camera for Developers to access
2. Transparency for Developers which in-vehicle cameras can be accessed (+ its location)
3. Definition reference implementation along existing Google standard (across OS versions)
4. Ensurance for developers that this reference implementation works across multiple OEMs
5. Future: Enabling 3rd Party Developers to test camera integration with Cross-OEM emulator

Feedback on reference implementation
Demo of POC / solution
Clarification timeline OEM Platform Support
Publication reference implementation

Positive feedback already provided by:

faurecia aptoide
HARMAN gm
In-Vehicle Camera Access: Reference Implementation Deep Dive

Guidelines
- AUTOMOTIVE_LENS_FACING only exists in Camera2, not CameraX
- App targetSdk should be 33 where AUTOMOTIVE_LENS_FACING was introduced (Android 13)
- App minSdk should be 24 because PackageManager hasSystemFeature API is required

Implementation logic

Check if system is AAOS and API level >= 33
PackageManager.hasSystemFeature(FEATURE_AUTOMOTIVE)
Android.os.Build.VERSION.SDK_INT >= ...

In-Cabin Camera
AUTOMOTIVE_LENS_FACING_INTERIORSeat_ROW_1_CENTER
Front Ext. Camera
AUTOMOTIVE_LENS_FACING_EXTERIOR_FRONT
FALLBACK

In-Cabin Camera
LENS_FACING_FRONT
Front Ext. Camera
LENS_FACING_BACK

Obs: these are only for the default lens facing apps can expect in automotive, but they can still use Camera2 APIs to enumerate the different cameras and characteristics/metadata.
Push Notifications: Deliverable & Status

DELIVERABLE
Agreeing upon a common service/protocol of how 3rd Parties can provide push notifications.

INTERIM STEPS
1. Deciding who is building and financing the reference implementation
2. Deciding who is hosting and financing the necessary backend service
   → COVESA vs. every OEM individually
   → If OEM specific implementation: Test suite to ensure compatibility for 3rd Party Developer

Definition possible solutions
First POC
Working POC on basis of Unified Push

OEM & app feedback
Currently app feedback is being gathered

Decision implementation logic

Implementation & publishment solution

Android Automotive VHAL
Status update: Push Notifications

• Push Notification POC from Maximilian Galanis (BMW)
Status update: Emulator

- Two possible directions for Cross-OEM availability:
  - Snapp:
    - Agreed to provide a generic emulator
  - Harman
    - Have a more customizable emulator and currently working to define next steps.
What's next?
What’s Next

• At COVES A AMM
  – Brainstorming session
  – Working session on push notifications
  – Round table

• Opportunity for an event in the frame of CES 2024 on January, 9th
  – Joint podium discussion
  – Vehicle demonstration
Questions
Thank You For your Time
Back Up Slides
Ecosystem: AAOS Apps framework standardization initiative

The concept
- Android with Google built-in
- Whatever the Android Automotive flavour
- Whatever the distribution channel
- Whatever the Voice Assistant
- Whatever the Maps Provider
- The same version of the Android Automotive apps distributed to several OEMs

The purpose
- Avoid fragmentation
- Increase attractiveness of automotive market for Android Apps developers

Integration into COVESA
- Launch of the Expert Group
- Jan 2023
- 16th May 2023
- Garching F2F meeting
- Work on Push notifications

The involved OEMs and Tier1
- GM
- Hyundai
- Kia
- Ford
- Stellantis
- Renault
- Harman
- Siemens

Identified priorities
- AOSP Emulator
- Push notifications
- Vehicle data collection and Sensor interface definition, with initial focus on sensors and properties which are not supported yet by AAOS (e.g. ambient lights)
Status update: Camera API

CLAIRIFICATIONS OEMS.

In-Vehicle Camera Access.

Camera libraries and APIs in Android have a set of characteristics, one of them being the LENS_FACING. This property informs 3rd-party developers which camera to select.

- **AAOS (Android 13 or higher):**
  - Introduction of AUTOMOTIVE_LENS_FACING, allowing to access variety of in-cabin cameras
  - **Precondition:** FEATURE_AUTOMOTIVE system flag (PackageManager.hasSystemFeature API)
  - **Standardization:**
    - In-Cabin Camera: INTERIOR_SEAT_ROW_1_CENTER
      - Should this be the default or rather LEFT / RIGHT?
    - Front-facing exterior Camera: EXTERIOR_FRONT
    - LENS_FACING can be supported in addition to AUTOMOTIVE_LENS_FACING

- **Android or AAOS (up to Android 12):**
  - **Standardization:**
    - In-Cabin Camera: LENS_FACING_FRONT
    - Front-facing exterior Camera: LENS_FACING_BACK