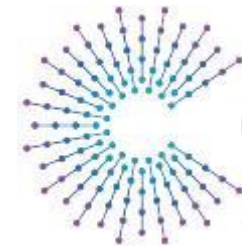




AUTOMOTIVE AOSP APP FRAMEWORK
STANDARDIZATION EXPERT GROUP

COVESA AMM Update

September 25th , 2024



COVESA

Accelerating the future of connected vehicles

ANTITRUST NOTE

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.

CO-CHAIRS STANDARDIZATION EXPERT GROUP



Richard Fernandes

Connectivity Standards Engineer



Melina Mascolo

Product Owner 3rd Party Apps



Gabriel Gautron

Innovation Project Manager



TECHNICAL EXPERTS



Maximilian Galanis



José Freitas



A decorative background at the top of the slide featuring a network diagram. It consists of numerous circular nodes connected by thin lines, forming a complex web. The nodes and lines are rendered in a light blue color that fades out towards the bottom of the image.


WHAT IS HAPPENING IN THE AUTOMOTIVE APP ECOSYSTEM?

WHITELABEL APP STORES ARE ALL AROUND AUTOMOTIVE. ... AND AROUND DROIDCON.. ;)

Home > Mercedes-Benz > E-Class > Features

Hands On With Mercedes' New Third-Gen MBUX Infotainment With Android Apps

The 2024 Mercedes-Benz E-Class' infotainment will bring Angry Birds, TikTok to the road.



1.

**MOTOR1 ORIGINAL
HANDS-ON WITH MBUX 3.0**

Feb 28, 2023 at 1:07pm ET

How Lotus partnered with Dolby, Faurecia Aptoide, and TIDAL to pave the way for in-vehicle infotainment app developers



HOME · CARS · NEWS

CES 2023: BMW is going all-in on Android Auto Open Source – here's why



By Nate Swanner
January 5, 2023

SHARE

TRANSPO / VOLKSWAGEN / CARS

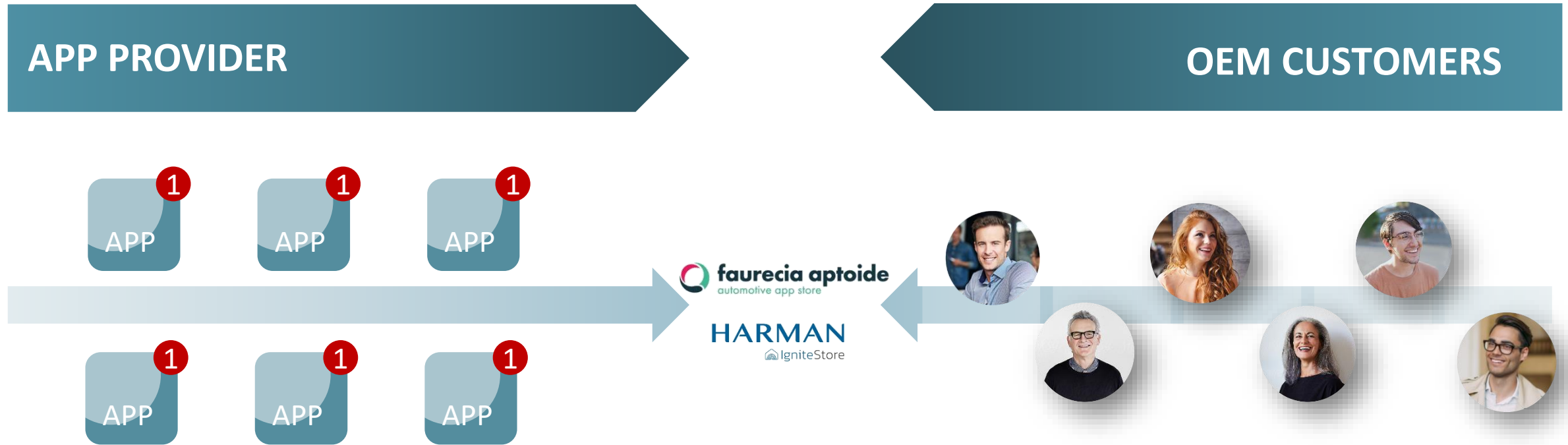
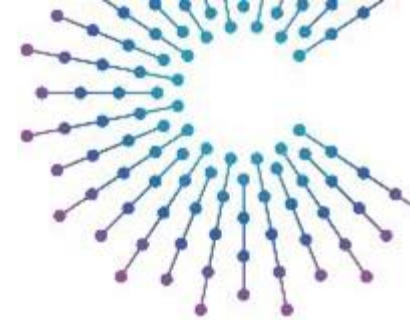
Volkswagen, Audi, and Porsche are getting their own in-car app store – and yes, that includes TikTok



Volkswagen Group is getting on the apps. TikTok, Spotify, The Weather Channel, and more are coming to VW's new in-car app store.

By PATRICK GEORGE
Mar 1, 2023, 1:45 PM GMT+1 | 32 Comments / 10 News

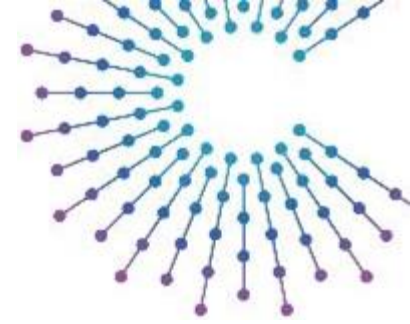
WITH AN AOSP-BASED APP STORE, OEMS SUPPORT A RELEVANT 3RD PARTY ECOSYSTEM TO EMERGE.



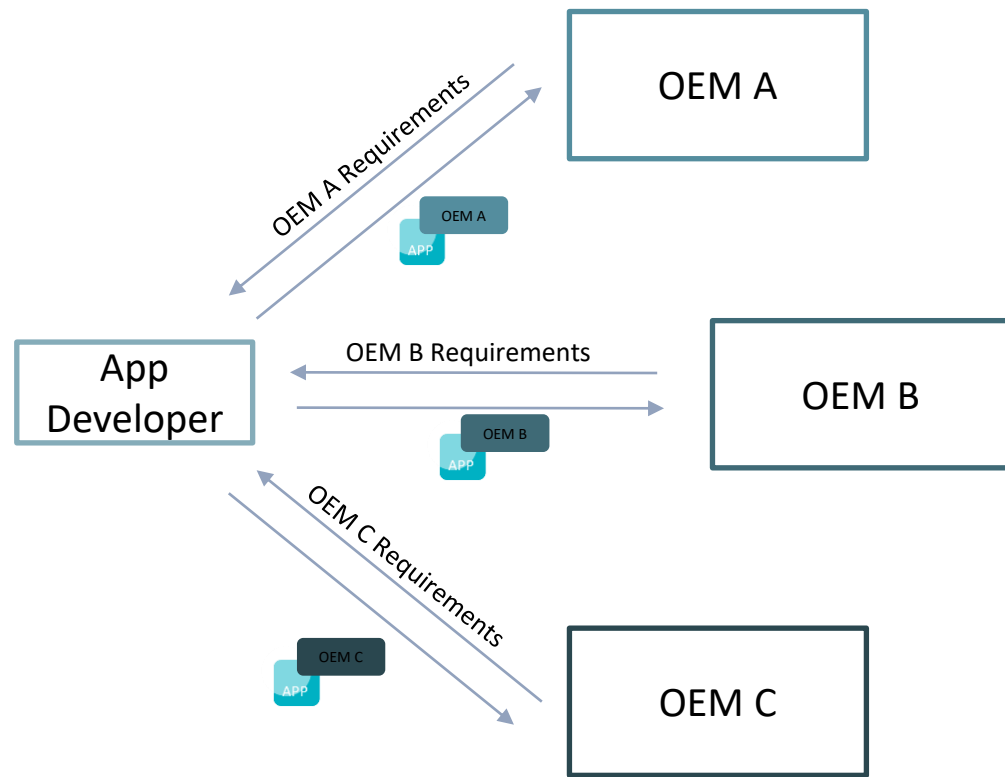
Platform ecosystems **need to attract and coordinate two different target groups** – app developers and OEM customers.

The OEMs need to start focusing on the **App Providers as an important customer** group in order to grow their content portfolio.

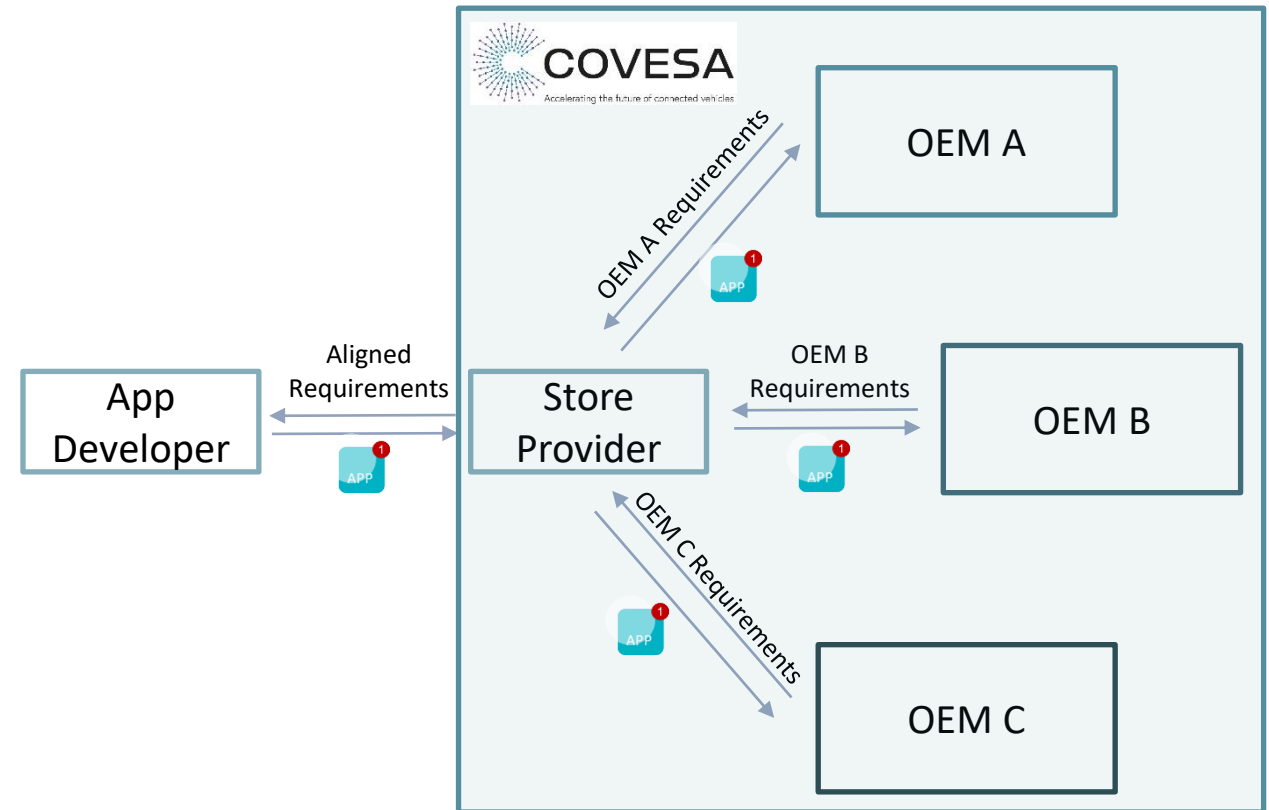
TRADITIONAL APPROACH VS. PLATFORM ECOSYSTEM APPROACH.



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

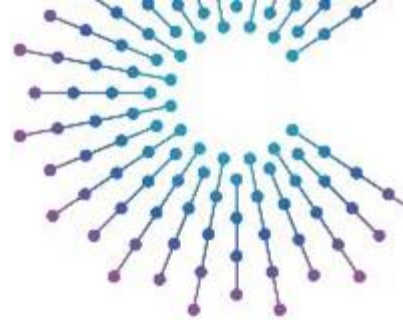


Traditional Approach



Platform Ecosystem Approach

GOALS AUTOMOTIVE AOSP APP FRAMEWORK STANDARDIZATION.



1 Enabling innovation and rich experiences.

2 Frictionless onboarding and testing for app developers.

3 Building a Cross-OEM App Ecosystem.

3 Avoid fragmentation.

A decorative graphic at the top of the slide consists of a network of interconnected nodes and lines. The nodes are represented by small circles in various shades of blue and purple, connected by thin lines. The overall pattern is a complex, web-like structure that spans the width of the slide.

WHAT ARE WE DOING WITHIN COVESA?

AUTOMOTIVE AOSP APP FRAMEWORK STANDARDIZATION EXPERT GROUP.



OEMs



& many others

Apps



& many others

Automotive AOSP App Framework Standardization

Store Provider



Tier 1

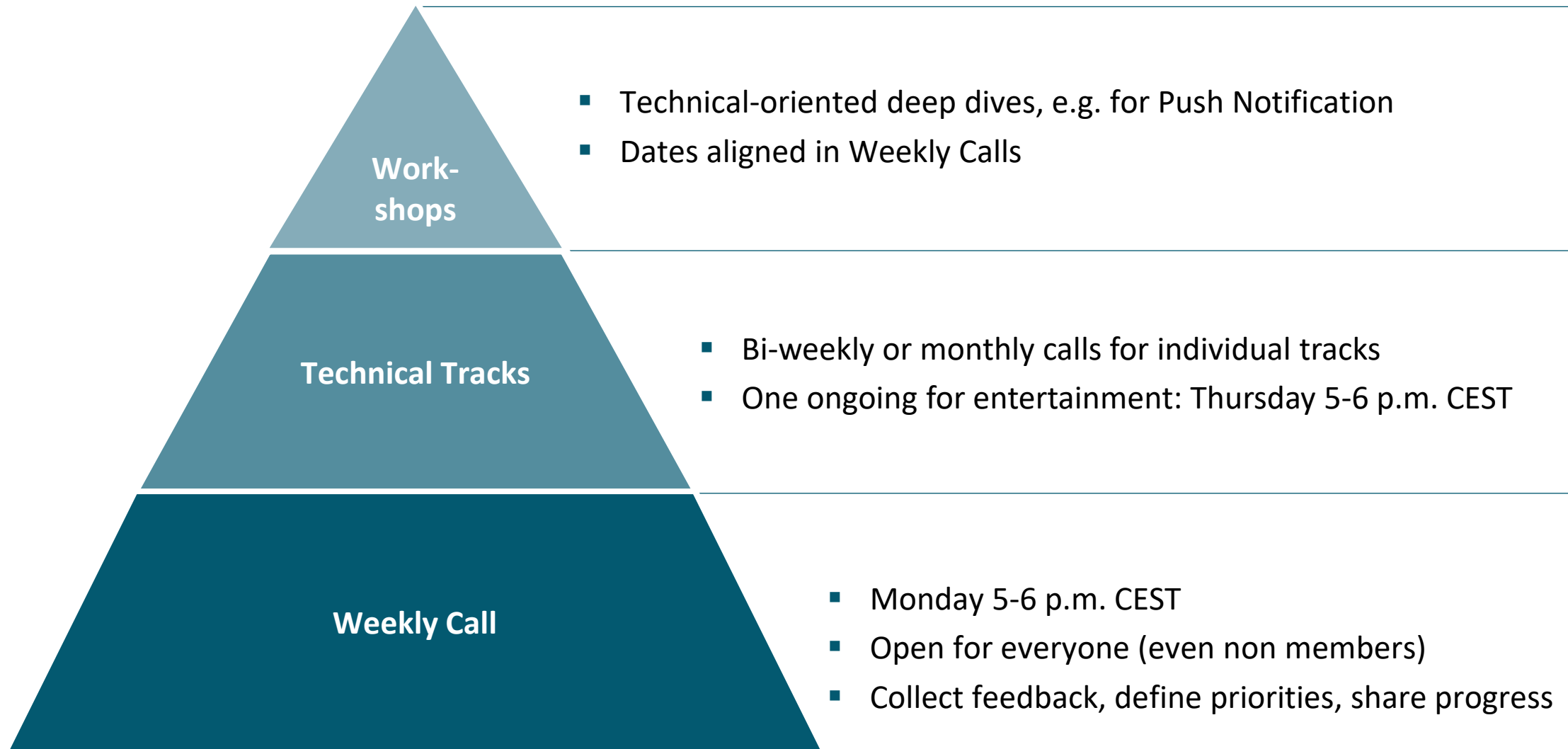


.. and a lot more

A decorative graphic at the top of the page consists of a network of interconnected nodes and lines. The nodes are represented by small circles, and the lines are thin, connecting the nodes in a complex, web-like pattern. The colors of the nodes and lines transition from a dark blue on the left to a light blue on the right.

ONGOING WORK STREAMS.

WORKING MODE OF THE EXPERT GROUP.



FIRST SUCCESS STORIES.



Release of the **COVESA-AOSP SDK 1.0**



Alignment on **UnifiedPush as Open Source solution** for Push Notifications



Definition of requirements and specification of UnifiedPush extension



Zoom as first 3rd Party App Provider to support **UnifiedPush**



Secured funding to start reference implementation and specs for UnifiedPush solution

ON GOING WORK STREAMS IN THE EXPERT GROUP.



PUSH NOTIFICATIONS

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

Deliverables: Push server reference implementation, Distributor App, Client Lib, Documentation



ENTERTAINMENT

Reducing access barriers for content providers by standardizing the technical implementation in the car (e.g. geolocalization)

Deliverable: Deliverable: Client SDK, Documentation



COVESA SDK

Collection of all “COVESA libraries” allowing 3rd party apps to access automotive features in a way that is agnostic of manufacturer and model. Available in Open Source Repo, (soon) open for contributions.

Deliverable: Open source code, Build on Maven

ON GOING WORK STREAMS IN THE EXPERT GROUP.

Projects started

- Push notifications
- Covesa SDK / ambient lights
- Entertainment
- In-vehicle Camera API access

Official candidates

- Emulators

Proposals

- App host
- Documenting fragmentation

PARTICIPATION TO INDUSTRY EVENTS

Droidcon (Berlin) on July 4th

The image shows a screenshot of a Droidcon Berlin session page. The session is titled "How 'Write-Once / Deploy Everywhere' is Becoming a Reality in Automotive Applications" and is presented by Gabriel Gautron, Daniele Bonaldo, Jose Freitas, and Melina Mascolo. The description states that Covesa standardizes Android Automotive to make it easy to create applications and new experiences for the automotive industry. Below the session details, there is a social media post from Chris Simmonds, a professional geek and consultant in embedded Linux and embedded systems. The post mentions that Melina Mascolo, Gabriel Gautron, and Daniele Bonaldo spoke at #dcb24 about Android standardisation within the Connected Vehicle Systems Alliance (COVESA) to avoid fragmentation. The post includes a video thumbnail showing a presentation slide titled "The Connected Vehicle Systems Alliance" and a photo of the speakers on stage.

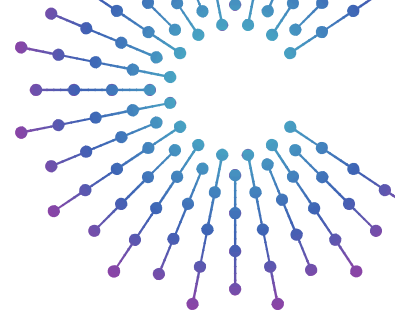
IBC24 (Amsterdam) on Sept 2024

The image shows a screenshot of an IBC24 event page. The page is divided into three main sections: Workshop Objectives, General Information, and Attendees. The Workshop Objectives section lists the following points: Connect leading car OEMs with Content Providers; Discuss the role of the car as the next platform of entertainment; Identify ways to remove friction to accelerate content adaptation in the car; and Freely exchange views on the main use cases and onboarding bottlenecks between Content Partners and car OEMs. The General Information section provides the following details: Friday, September 13th, 2024 (onsite + remote); 2:00 pm - 4:00 pm CET; Meeting Room G107 at RAI Amsterdam (On top of the main/central halls (Hall 2)); Online: The link to be sent prior the event; and Beach Party: 5:00 pm CET after the workshop (Drinks & Snacks) -> Details to be shared. The Attendees section states: Multiple leading global and local content providers; and COVESA Car OEM members and non-members. A map of the RAI Amsterdam venue is shown at the bottom right, highlighting the location of the event.



PUSH NOTIFICATION

PROBLEM STATEMENT – PUSH NOTIFICATIONS.



“A push notification (also known as a server push notification) is the delivery of information to a computing device from an application server where the request for the transaction is initiated by the server rather than by an explicit request from the client“ [1]

❖ Common use-cases:

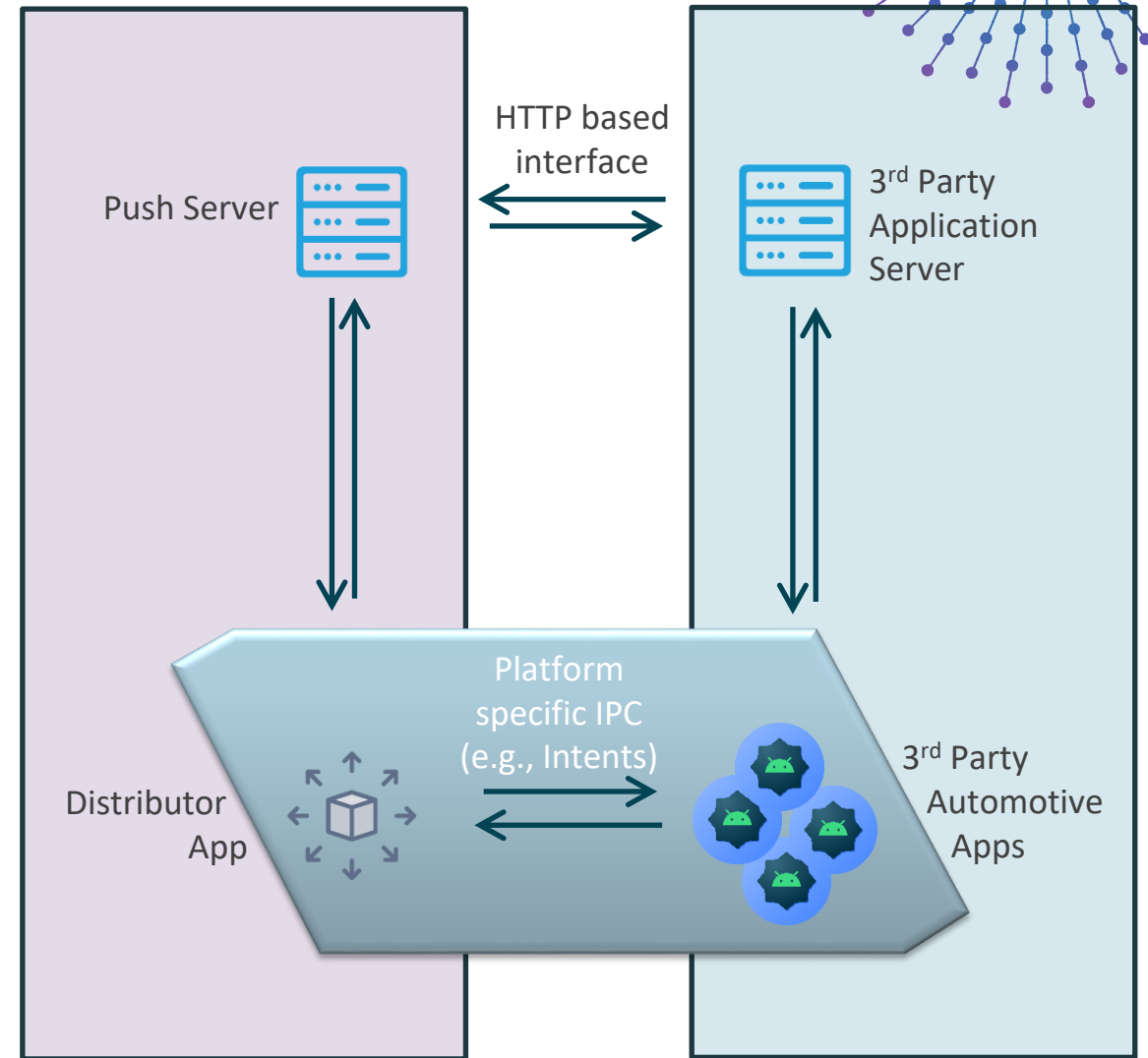
- Delivery of messages
- Incoming VoIP calls
- Live updates in a running app (e.g., update list of upcoming meetings)

Problem

- Apps get killed by the system and cannot rely on being able to communicate with their backend
- There exists no standardized push notification service for AOSP/AAOS

THE ARCHITECTURE OF A PUSH NOTIFICATION SERVICE.

- ❖ A push notification service (PNS) generally consists of four parts
- ❖ Notifications are triggered by an HTTP POST request to the respective push-server
- ❖ The exact type of the connection between the distributor and the push server varies
- ❖ The mechanism used by the distributor to talk to the application depends on the platform



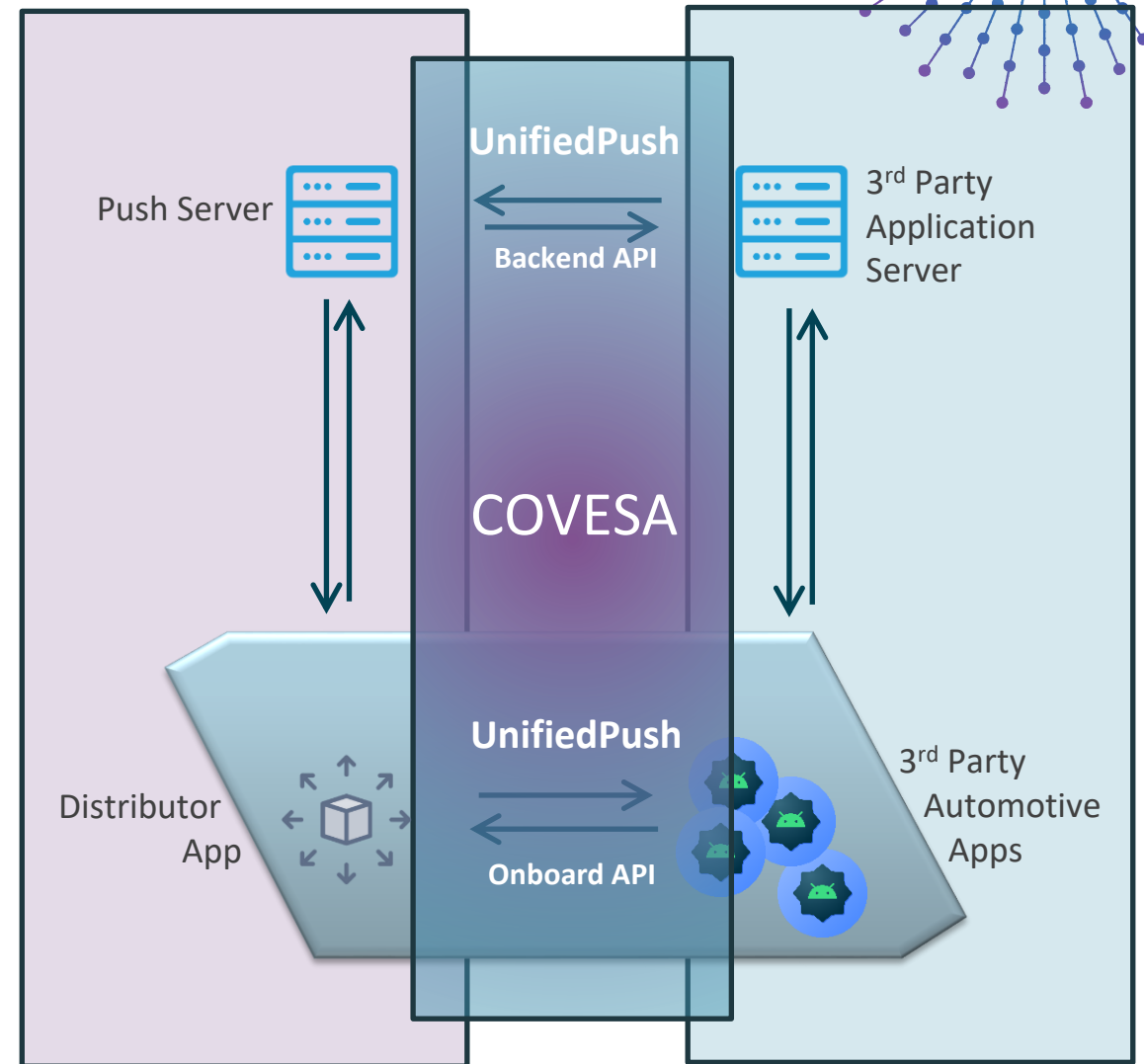
PUSH NOTIFICATIONS – UNIFIED PUSH

❖ From previous analysis performed independently by different parties **UnifiedPush** is the most promising solution:

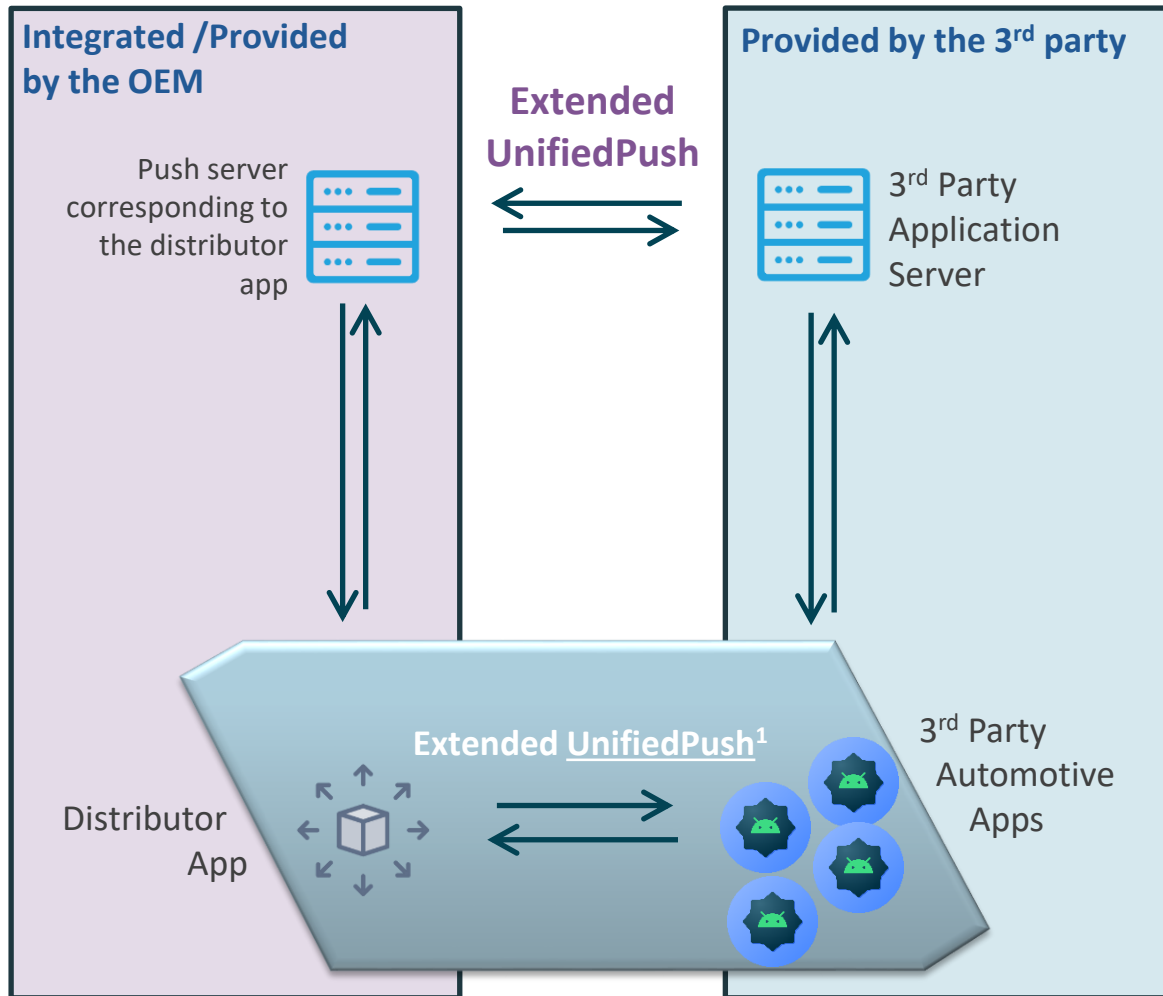
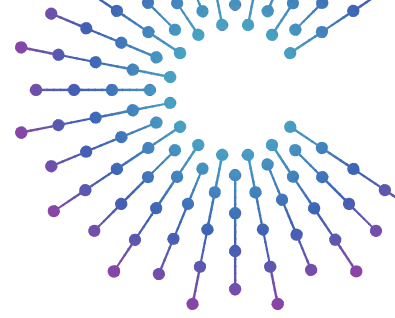
- Provides a specification
- Apache 2.0 licensed, same as AOSP

❖ However, it lacks some important features:

- **No support** for message receipts
- **No support** for time-to-live
- **No support** for urgency
- **No support** for collapsible messages
- **No support** for VAPID



OVERVIEW: HOW WE ARE EXTENDING UNIFIED PUSH



❖ Taken care of:

- Libraries and the Example App
- Specification for push server and distributor
- Reference implementation

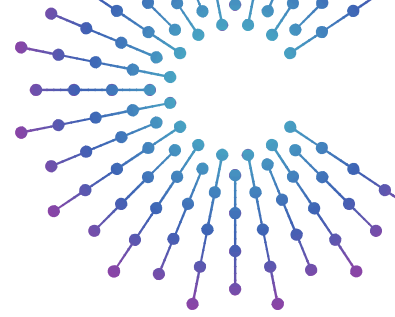
❖ **TODO OEMs:**

- Provide a distributor app in the car / Host the reference implementation

❖ **TODO 3rd parties:**

- Implement UnifiedPush support in their app and their application backend

OVERVIEW: CURRENT STATUS



❖ Unified Push Notification Project Time Plan:

1. UnifiedPush Specification:
 - a. Implementation: September 2nd, 2024 – September 20th, 2024
 - b. Review Period: September 20th – October 4th
2. UnifiedPush Connector & UnifiedPush Example App:
 - a. Implementation: September 23rd, 2024 – October 25th, 2024
 - b. Review Period: November 8th, 2024 – November 22nd, 2024
3. Reference Implementation:
 - a. Implementation: October 28th, 2024 – December 20th, 2024
 - b. Review Period: December 23rd, 2024 – January 17th, 2025



The screenshot shows the Codeberg repository page for 'UnifiedPush / specifications'. The page includes navigation tabs for Code, Issues (1), Pull requests (2), and Activity. It displays repository statistics: 70 commits, 3 branches, 0 tags, and 186 KIB. A file browser shows the 'main' branch with a search bar and a list of files: 'S1m' (6806fe8b2e, Update link to codeberg, last month), 'specifications' (Update link to codeberg, last month), 'definitions.md' (Update link to codeberg, last month), 'LICENSE' (Update LICENSE, 3 years ago), and 'README.md' (Update README.md, 4 years ago). Below the file list is a preview of the 'README.md' file, showing the title 'UP-spec' and the text 'UnifiedPush Specifications'. The URL <https://codeberg.org/UnifiedPush/specifications> is displayed at the bottom of the screenshot.



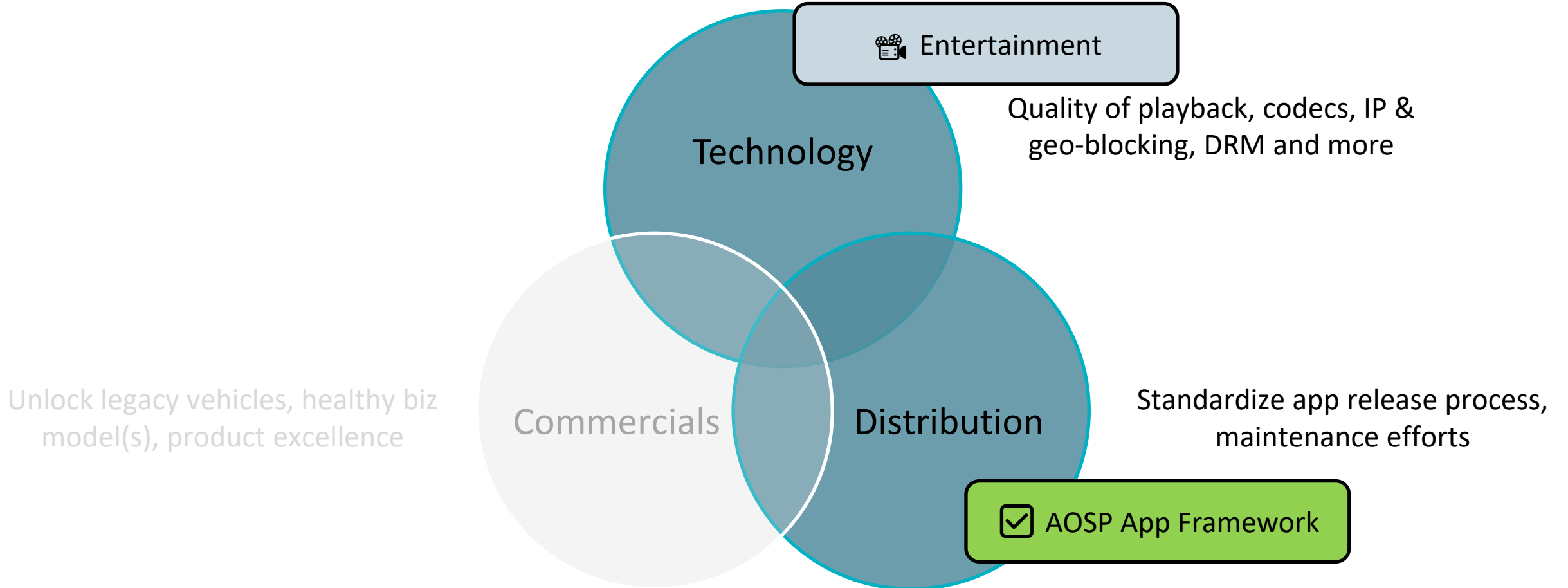
ENTERTAINMENT

COVESA Entertainment Working Group Goal

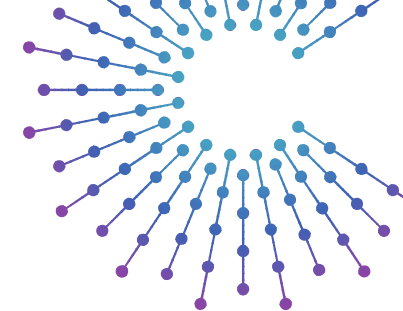
Reducing access barriers for content providers by standardizing the technical implementation in the car.

What can we impact within COVESA?

Reduce entry barriers to explore with in-vehicle video streaming.



ENTERTAINMENT WORKING GROUP SCOPE



1. DRM

- Challenge: closing the gap between GAS Systems with Widevine DRM L1 vs. non-GAS
- Solution: defining the minimal requirements and standardizing the certification process

2. GEOLOCATION

- Challenge: using IP for geo-blocking streaming content is not feasible when using a local IP hub
- Solution: standardize non-IP-based location sharing following privacy laws

3. CERTIFICATION

- Challenge: clustered and undefined process per car brand
- Solution:
 1. Introducing a standardized way to certify entertainment apps for automotive
 2. introducing readiness on non-GAS systems (streaming technologies, codecs, frame rates, ...)

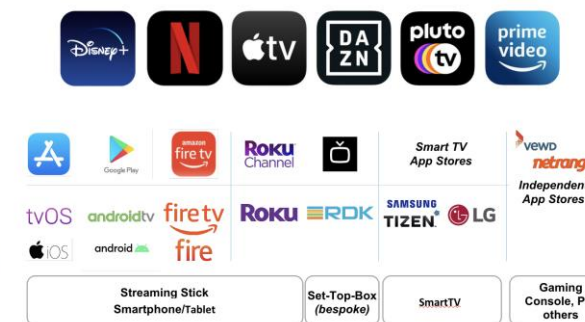
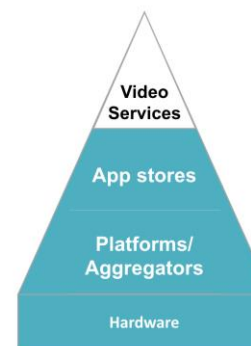
4. ANALYTICS

- Challenge: lack of device & app monitoring capabilities
- Solution:
 1. standardization of streaming analytics data and reporting formats to easily track
 2. standardization of the quality of experience across all automotive platforms (average bitrate, error rate, ...)

5. TECHNOLOGY

- Challenge: automotive hardware not seen as a relevant content distribution platform
- Solution:
 1. defining the minimum requirements on hardware performance and software testing & specs
 2. defining the requirements to become attractive for global streaming providers
 3. Multi-screen behaviour & the right aspect ratio & sidepanel reusage for wider display (gradient/ambient)

Selected topic



IBC 2024
Sep 13th - 16th

COVESA
Accelerating the future of connected vehicles

faurecia aptoide
automotive app store

3SS

The Car as a New Entertainment Platform

Connect With Top Content Partners And Car OEMs!



ROLLS-ROYCE
MOTOR CARS LTD



HYUNDAI
MOTOR GROUP

HONDA
The Power of Dreams



STELLANTIS KARMA



Workshop Stats

First workshop between Content Providers and OEMs to discuss In-Car Entertainment

- 2h workshop during IBC Amsterdam
- 60 signups
- 2 x presentation from Content Providers:
 - **YouTube** presentation on content distribution across vehicle platforms and YouTube's technology outlook
 - **BBC** on distribution and usage in terrestrials
- Discussion panel between 3SS, BMW, Mercedes, BBC, Faurecia and CARIAD on in-car entertainment



Participants

Car Makers

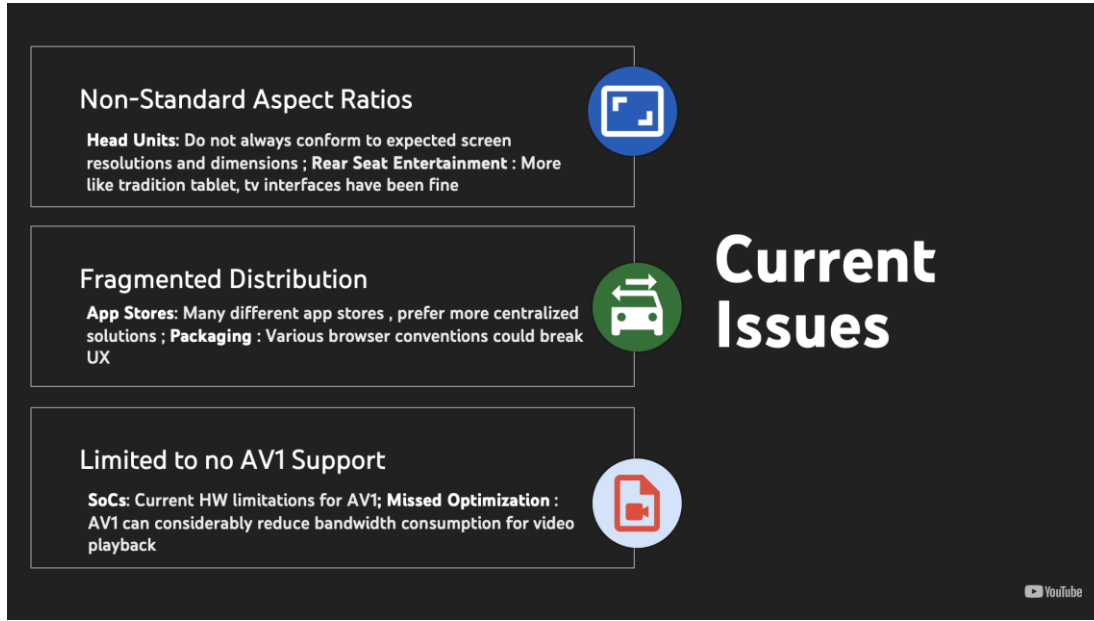


C A R I A D

Content Providers



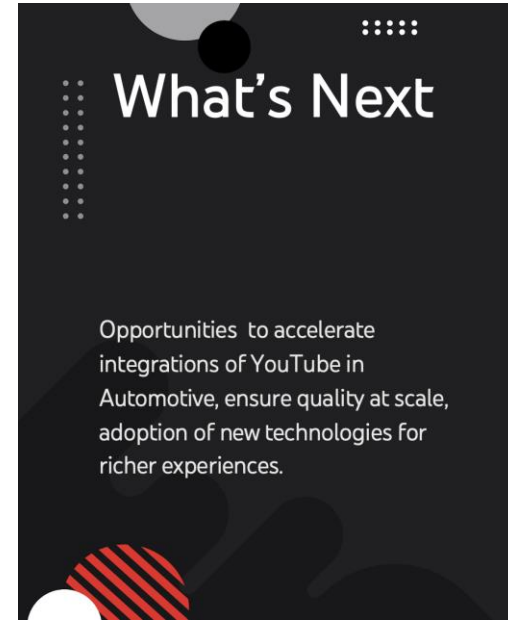
Global Streamer: YouTube insights



Current Issues

- Non-Standard Aspect Ratios**
Head Units: Do not always conform to expected screen resolutions and dimensions ; Rear Seat Entertainment : More like tradition tablet, tv interfaces have been fine
- Fragmented Distribution**
App Stores: Many different app stores , prefer more centralized solutions ; Packaging : Various browser conventions could break UX
- Limited to no AV1 Support**
SoCs: Current HW limitations for AV1; Missed Optimization : AV1 can considerably reduce bandwidth consumption for video playback

YouTube



What's Next

Opportunities to accelerate integrations of YouTube in Automotive, ensure quality at scale, adoption of new technologies for richer experiences.

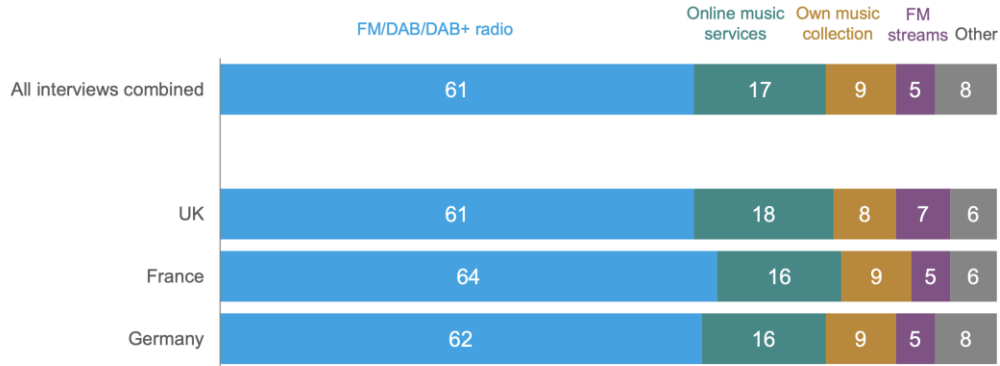


- Quality at scale for user experiences greatly benefit with the use of **AV1**
- Supporting **IAMF** (Spatial Audio) for YouTube content enriches the in-car experience
- Seeking **common solutions** to reduce fragmentation : packaging & certification

- More board adoption of AV1 (faster video startup, fewer rebuffers, higher quality while using less bandwidth)
- Supporting new technologies for spatial audio such as IAMF (Immersive Audio Model and Formats)
- Reducing implementation fragmentation by providing common solutions - mainly for Android based infotainment systems.

Public Broadcaster: BBC insights

Which do you typically listen to most often in a car/vehicle?



Base: Typically spend any time in a car/vehicle

Source: Dashboard Dialogue Study 2023: Online survey of recent/prospective car buyers



BBC's plans for cars through 2026

	Off product	On product
2024	<p>Launch Android Automotive BBC Sounds with partners</p>	<p>Improve data to better understand audience behaviour</p>
2025	<p>Launch Android Automotive BBC World Service app with partners</p> <p>Acquisition via collaboration, e.g.</p> <ul style="list-style-type: none"> • Deep links from radio • Pre-installs, Quick-start, etc • Promo feeds in partner UX 	<p>Validate use cases for integrating key News, Weather, Sport, iPlayer moments to in-car app UX</p>
2026		<p>Build value for drivers via continuous discovery and incremental improvement</p>

BBC PARTNERSHIPS

- Needs to work closer with OEM on improving content discovery and convenience
- Radio / Audio experience remains super important
- Integrating new use cases via native AAOS apps

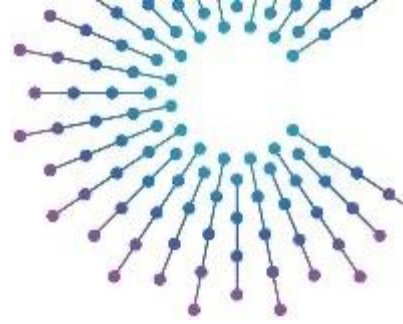


Outcome: More cooperation between OEMs and Content Partners to leverage vehicle platform and to create new value for our customers

We need more OEMs joining
the COVESA Entertainment
Group to convince Content
Partners to join us!



Building Car App 2.0, together!



Use Case #1 : Contextualized Recommendations

Situation:

- Vehicles are moving sensors
- People use cars to solve a task

Solution:

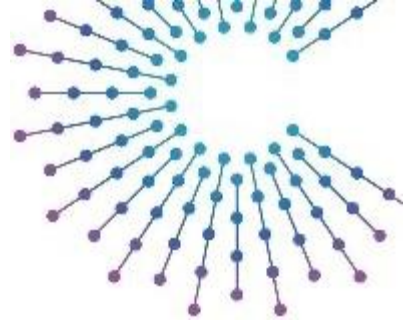
- Use vehicle data to trigger and contextualize the app content
- App content is adjusted to the right situation

Outcome:

- A new type of personalization
- The new data point for App Developers to create new automotive content

Example data points → Route information, ETA, driving state

Building Car App 2.0, together!



Use Case #2 : Vehicle APIs

Situation:

- Many features are not known in the car

Solution:

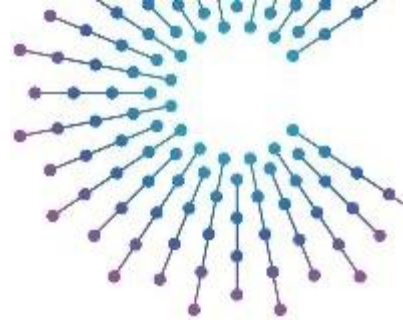
- Use specific scenarios to guide customers to explore what's available in the car
- Use Notifications for App Activation

Outcome:

- Car as a smart companion
- Crafting valuable services for customers that ease their life

Example data points → EV charge state, remaining charging time

What's next?



- Define the short-term objective for the COVESA Entertainment Working Group
 - 3SS proposal from last work stream meetings -> Geolocation
- Involve more Content Partners in COVESA Entertainment Working Group
- Create a dedicated COVESA session between Car Makers and Content Providers during CES 2025 -> to be confirmed

How to join?

COVESA Bi-weekly / Monthly Sync

- When?
 - Thursday 11:00 am ET
- Write us message or go to [COVESA Common Calendar](#) to add it to your calendar



CES 2025

- Preparing another meeting between **Content Providers & Carmakers**
 - To be confirmed



Contact Us



Felix Walter

Head of Automotive

felix.walter@3ss.tv



Tomasz Dzikowski

Product Manager

tomasz.dzikowski@3ss.tv



Robert Glas

Director Technology

robert.glas@3ss.tv

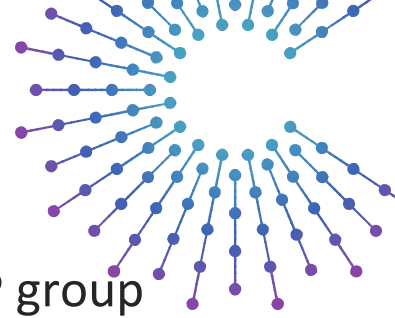


A decorative graphic at the top of the page consists of a network of interconnected nodes and lines. The nodes are represented by small circles, and the lines are thin, connecting the nodes in a complex, web-like pattern. The colors of the nodes and lines transition from a dark blue on the left to a light blue on the right, creating a gradient effect.

COVESA SDK PROOF OF CONCEPT

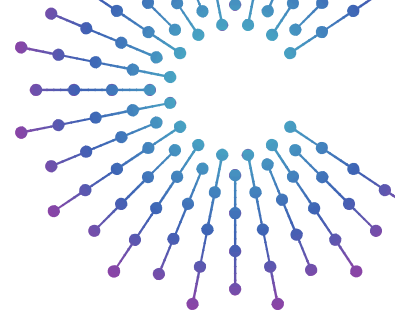
WHAT IT IS

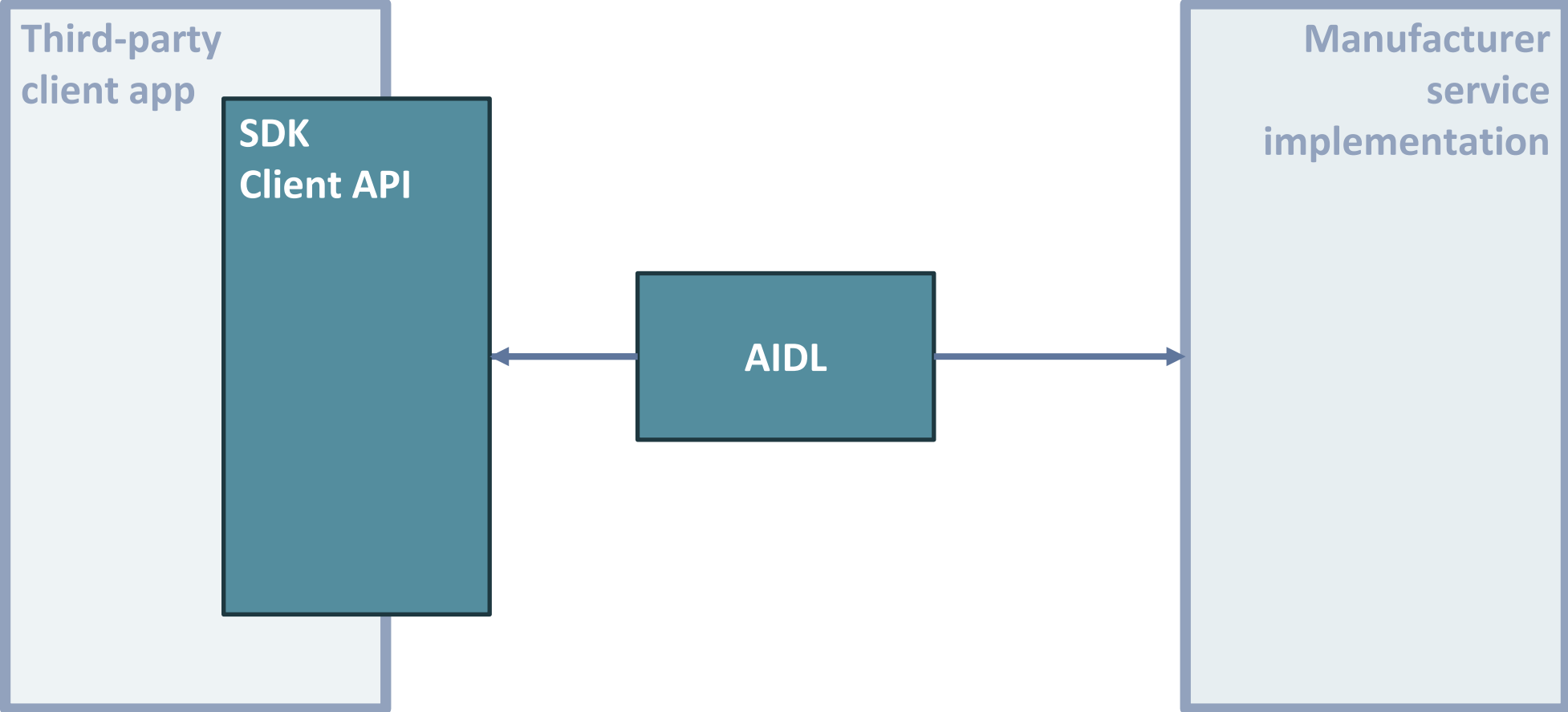
- A single Android SDK that integrates the standardization work developed by the Covesa AOSP group
 - Soon : Push notification client
 - Ambient light feature
 - ...
- The SDK will be made available on Maven
- Github page: <https://github.com/COVESA/covesa-aosp-sdk>



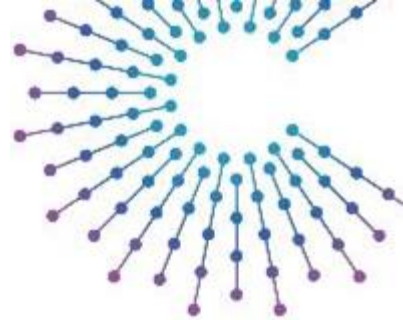
RESOURCES

- Group wiki on covesa wiki: [link](#)
- Covesa github: [link](#)
 - Project to create an AOSP group repository with documentation and specifications
- SDK github: [link](#)





POC – LIGHTS SERVICE



Third-party client apps can:

- Set lights state
 - Zone (driver, passenger, front, rear, all, etc)
 - Color
 - Brightness
- Get updates about lights statuses
- Get version of the installed lights service

FOR APP DEVELOPERS

- Client Library
 - AIDL definition
 - .aar
 - Soon to be in Maven
- Samples
- Code
- Documentation
- Ambient light service implementation
- Mock version of services
 - APK
 - Runs on any emulator

COVESA SDK Client sample

This first version of the COVESA SDK client sample shows how to use the COVESA Lights Service to update the values of internal lights and receive and updated state of all the lights.

Client SDK artifacts

The client SDK library is composed (at the moment) of two AAR files: `aidl-debug.aar` and `client-debug.aar`.

These need to be added as dependencies for third-party applications. Please refer to the samples/client module for an example of setup.

Services Catalog API

The main API entry point is `global.covesa.sdk.api.client.ServicesCatalog`. It is the service that informs a client at runtime about the other installed COVESA services.

It can be created with a Context constructor parameter and it exposes the following:

- `getInstalledServices()` returns a list of String, each one representing the action needed to connect to a supported COVESA Service via Intent.
- `getServiceApiVersion()` returns a nullable Int specifying the API level exposed by the service or `null` if the service is not installed

Lights Client API

LightsService

The main API entry point is `global.covesa.sdk.api.client.LightsService`. It can be created with a Context constructor parameter and it exposes the following:

- `lightsStates`: it's a [kotlin flow](#) of `LightState`. Clients should collect from this flow to get reactive updates about the current state of all the lights.
- `getServiceApiVersion()` returns a nullable Int specifying the API level exposed by the service or `null` if the service is not installed
- `setInternalLight(lightState: LightState)` which should be used to update the value of a specific zone lighting.

Due to the remote client-service communication at the base of the SDK, all methods are `suspend` and for that reason should be invoked from a [CoroutineScope](#).

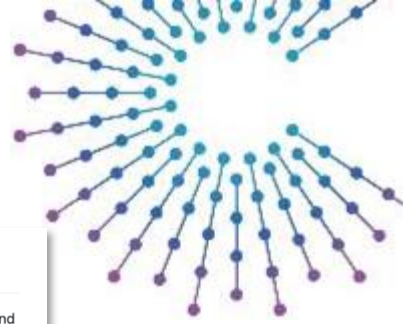
LightState

`global.covesa.sdk.api.lights.LightState` is a Parcelable which describes a single light.

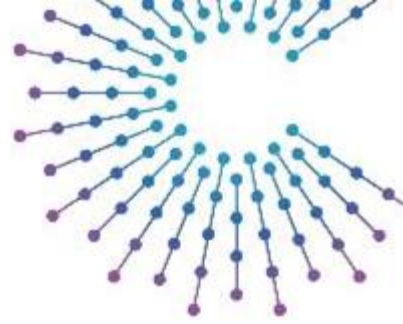
It specifies the zone relative to the light, the RGB color (in the form of `global.covesa.sdk.api.lights.LightColor`) and the brightness.

The zone can be one of the following:

- `ALL_ZONES = 0`
- `ZONE_DRIVER = 1`

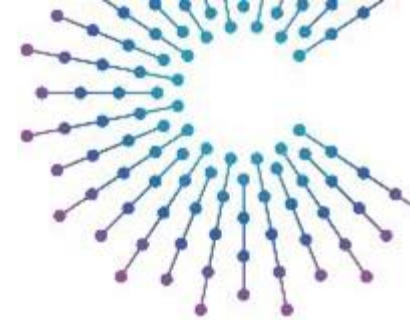


AIDL



```
interface ICovesaLightsRemoteService {  
    const int API_VERSION = 1;  
    int getApiVersion();  
    void setInternalLight(in LightState state);  
    void registerLightsStateListener(in ILightsStateListener listener);  
    void unregisterLightsStateListener(in ILightsStateListener listener);  
}
```

CLIENT API



```
private var serviceConnection: ServiceConnection? = null

try {
    check(serviceConnection == null) { "service is already connected" }

    Log.i(TAG, "connecting to remote service")
    serviceConnection = object : ServiceConnection {
        override fun onServiceConnected(name: ComponentName, service: IBinder) {
            trySend(interfaceFromBinder(service))
        }

        override fun onServiceDisconnected(name: ComponentName) {
            cancel("connection to service lost")
        }

        override fun onNullBinding(name: ComponentName?) {
            Log.i(TAG, "received null binding from remote service")
            cancel(
                "connection to remote service failed",
                NullPointerException("received null binding from remote service")
            )
        }

        override fun onBindingDied(name: ComponentName?) {
            Log.i(TAG, "connection died from remote service")
            cancel("connection to remote service lost")
        }
    }

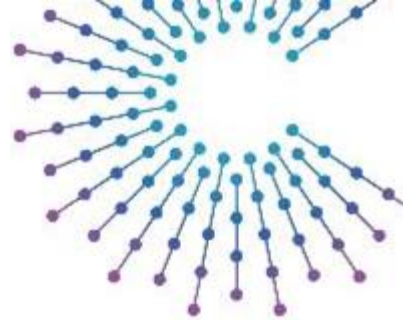
    val serviceBound = context.bindService(
        serviceIntent(),
        serviceConnection!!,
        Context.BIND_AUTO_CREATE
    )
    check(serviceBound) { "Service was not found" }
} catch (e: SecurityException) {
    cancel("connection to remote service failed", e)
    Log.w(TAG, "could not connect to remote service", e)
    serviceConnection = null
} catch (e: IllegalStateException) {
    cancel("connection to remote service failed", e)
    Log.w(TAG, e.message, e)
}
}
```

```
val listener = object : ILightsStateListener.Stub() {
    override fun onLightsStateUpdate(states: MutableList<LightState?>) {
        if (states != null) {
            trySend(states)
        }
    }
}

service.registerLightsStateListener(listener)

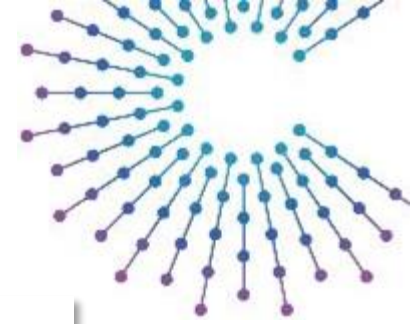
awaitClose {
    try {
        service.unregisterLightsStateListener(listener)
    } catch (e: DeadObjectException) {
        Log.w(TAG, "service died before unregistering lights listener")
    }
}
```

CLIENT API



```
class LightsServiceClient {  
    val lightsStates: Flow<List<LightState>>  
    suspend fun setInternalLight(lightState: LightState)  
    suspend fun getServiceApiVersion(): Int?  
}
```


FOR CAR MANUFACTURERS



- Shared Library
 - AIDL definition
 - .aar
 - Soon to be in Maven
- Samples
- Code

COVESA SDK Server sample

This version of the COVESA SDK server sample shows how to implement the COVESA Lights Service to listen for lights update requests and Catalog Service to provide clients information about the installed services.

Server SDK artifacts

The server SDK library is composed (at the moment) of an AAR file: ``aidl-debug.aar``.

These need to be added as dependencies for third-party applications. Please refer to the `samples/client` module for an example of setup.

Services Catalog

This service needs to be exported and have an `intent filter` defined in its manifest with the action `global.covesa.sdk.server.CovesaCatalogService.BIND`. It's possible to find a sample implementation in `global.covesa.sdk.server.CovesaCatalogService`.

The default implementation should list an action for each of the installed services among the known ones. The list of all the known COVESA services is defined in the AIDL file `global.covesa.sdk.api.ICovesaCatalogRemoteService`.

Lights Service

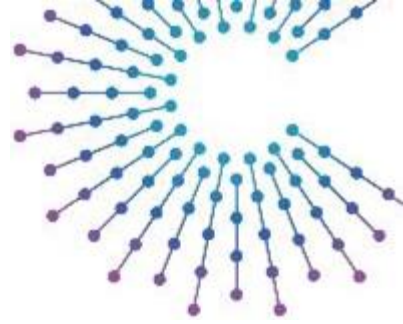
This service needs to be exported and have an `intent filter` defined in its manifest with the action `global.covesa.sdk.server.CovesaLightService.BIND`. It's possible to find a sample implementation in `global.covesa.sdk.server.CovesaLightService`.

The service must expose, in its binder, several methods defined in the AIDL:

- `registerLightsStateListener(...)` / `unregisterLightsStateListener(...)`: these allow clients to register for updates about the state of all the available lights zones
- `setInternalLight(LightState: LightState)`: this allows clients to update the light state for a given zone

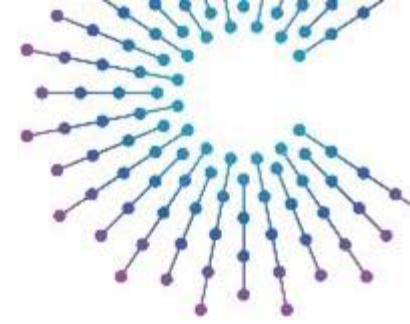
The sample implementation correctly notifies the registered listeners every time a client invokes `setInternalLight(...)`, and internally logs and shows a Toast which can be used by client applications developers to test their client implementation. A manufacturer implementing their version of the service should instead invoke the lower-level methods to interact with the hardware, where the sample app invokes `logReceivedState()`.

SERVICE SAMPLE



```
class CovesaLightService : Service() {  
  
    private val lightsListeners = mutableListOf<ILightsStateListener>()  
  
    private var currentLights = mutableMapOf<Int, LightState>()  
  
    ...  
  
    private val binder = object : ICovesaLightsRemoteService.Stub() {  
  
        override fun getApiVersion(): Int = ICovesaLightsRemoteService.API_VERSION  
  
        override fun setInternalLight(lightState: LightState?) {  
            // update internal state of lights  
            // notify listeners  
        }  
  
        override fun registerLightsStateListener(listener: ILightsStateListener?) {  
            // add a new listener  
        }  
  
        override fun unregisterLightsStateListener(listener: ILightsStateListener?) {  
            // remove a listener  
        }  
    }  
}
```

OPEN-SOURCE REPOSITORY



⑩ <https://github.com/COVESA/covesa-aosp-sdk>

The screenshot shows the GitHub repository page for `covesa-aosp-sdk`. At the top, it indicates the repository is public and shows interaction options: Edit Pins, Unwatch (5), Fork (0), and Star (1). The repository is currently on the `main` branch with 1 branch and 0 tags. A search bar and 'Add file' / 'Code' buttons are visible.

The file list shows the following structure:

File/Folder	Description	Last Updated
<code>api</code>	First PoC with lights service and services catalog	2 months ago
<code>gradle-plugins</code>	First PoC with lights service and services catalog	2 months ago
<code>gradle</code>	First PoC with lights service and services catalog	2 months ago
<code>samples</code>	First PoC with lights service and services catalog	2 months ago
<code>.gitignore</code>	First PoC with lights service and services catalog	2 months ago
<code>LICENSE</code>	Create LICENSE	2 months ago
<code>README.md</code>	Update README.md	last week
<code>build.gradle.kts</code>	First PoC with lights service and services catalog	2 months ago
<code>gradle.properties</code>	First PoC with lights service and services catalog	2 months ago
<code>gradlew</code>	First PoC with lights service and services catalog	2 months ago
<code>gradlew.bat</code>	First PoC with lights service and services catalog	2 months ago
<code>settings.gradle.kts</code>	First PoC with lights service and services catalog	2 months ago

The `README.md` file is selected, showing the following content:

STATUS **INCUBATING**

COVESA AOSP SDK

Objective of this SDK

The main goal of this SDK is to provide a set of libraries which allow third-party applications to access some automotive features in a way that is agnostic of manufacturer and model.

SDK components

About
No description, website, or topics provided.

- Readme
- Apache-2.0 license
- Activity
- Custom properties
- 1 star
- 5 watching
- 0 forks

Releases
No releases published
[Create a new release](#)

Packages
No packages published
[Publish your first package](#)

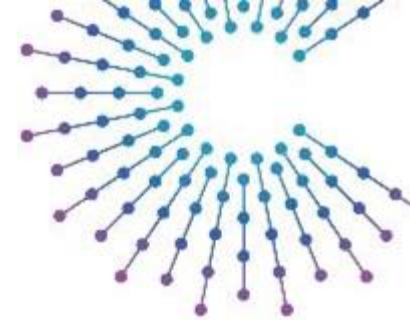
Contributors 2

- `paulboyes` Paul J. Boyes
- `danybony` Daniele Bonaldo

Languages

- Kotlin 85.0%
- AIDL 15.0%

NEXT STEPS



COVESA / covesa-aosp-sdk

Type to search

Code Issues Pull requests Discussions Actions Projects Security Insights

Filters is:pr is:open Labels 9 Milestones 0 New pull request

0 Open 0 Closed

Author Label Projects Milestones Reviews

There aren't any open pull requests.

You could search [all of GitHub](#) or try an [advanced search](#).

ProTip! Follow long discussions with [comments:>50](#).

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

COVESA / covesa-aosp-sdk

Type to search

Code Issues Pull requests Discussions Actions Projects Security Insights

Queries related to SDK design rationales ?

General · renjithrajagopal-sudo

Click to go back, hold to see history

Search all discussions Sort by: Latest activity Label Filter New discussion

Categories Discussions

View all discussions ↑ 2 **Queries related to SDK design rationales ?** 1

[renjithrajagopal-sudo](#) started 3 weeks ago in [General](#)

- Announcements
- General
- Ideas
- Polls
- Q&A
- Show and tell

Most helpful

Be sure to mark someone's comment as an answer if it helps you resolve your question — they deserve the credit! 🍷

[Community guidelines](#)

[Community insights](#)

JOIN US!

Contact us

- [COVESA chair: melina.mascolo@bmw.de](mailto:melina.mascolo@bmw.de)
- [COVESA chair: gabriel.gautron@forvia.com](mailto:gabriel.gautron@forvia.com)
- [COVESA chair: richard.fernandes@gm.com](mailto:richard.fernandes@gm.com)

Join our weekly call



zoom

General info





COVESA

Accelerating the future of connected vehicles

QUESTIONS

