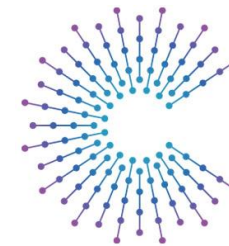




Project Proposal
COVESA Automotive Android App Host

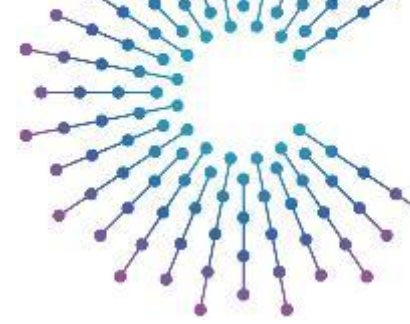
Richard Fernandes, Gabriel Gautron, José Freitas, Sabine Hofschien -
Automotive AOSP App Framework Standardization Expert Group



COVESA

Accelerating the future of connected vehicles

Requirements of in-car infotainment apps



What are the needs for an in-car app?

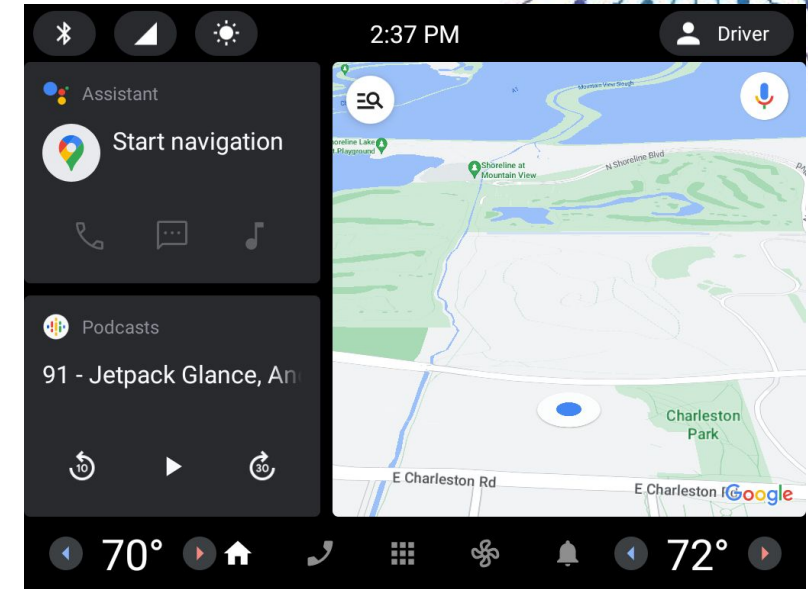
- **Safe user experience** - no distraction during driving
- 3rd party apps to be **visually and consistently integrated** in the OEM's infotainment system.
- **Easy way for app developers** to
 - develop UIs along these needs,
 - to test the apps in the COVESA emulator and
 - to develop the app once & deploy it at any OEM.

How are these needs fulfilled?

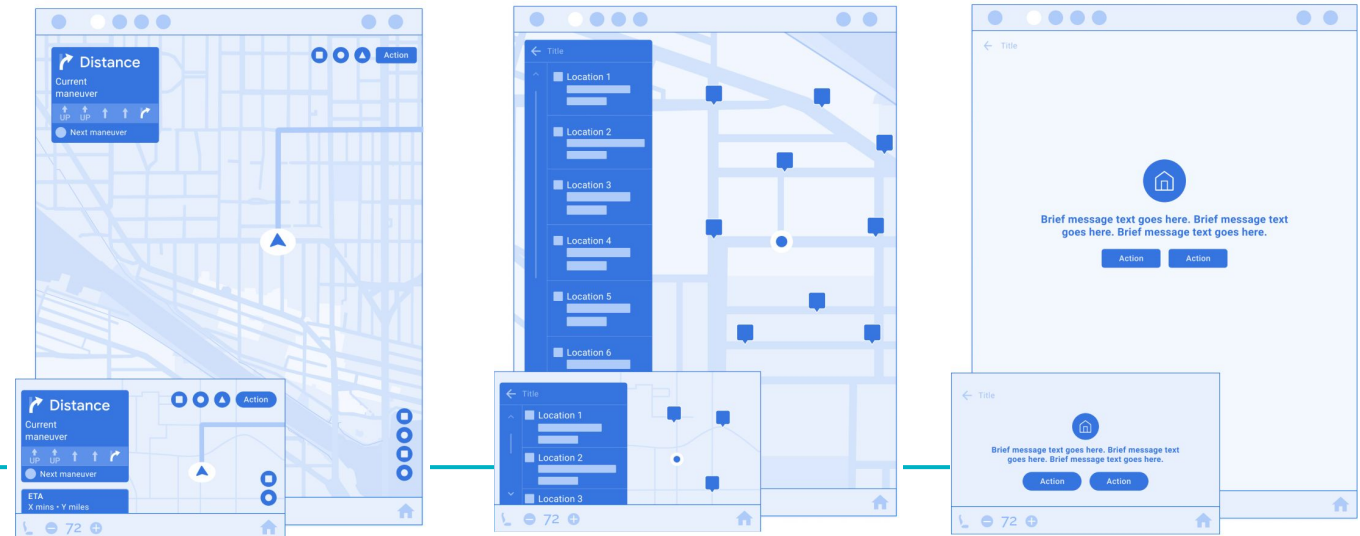
- **UIs are rendered by the system**, not by the app itself.
- An API for app developers - the **Car App Library CAL** - provides standard **UI templates** which minimize driver distraction.
- The templated apps have to be supported in the **emulator**, so as to let app devs test their apps.

What is the AAOS Automotive App Host

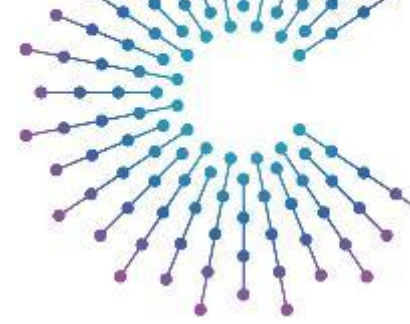
- A system application responsible for rendering apps that use the Car App Library* to ensure they are safe for driving.
- App developers have no control over the look and feel of components rendered by the app host.
- This provides a more cohesive and safer UX for applications that can be used when driving.



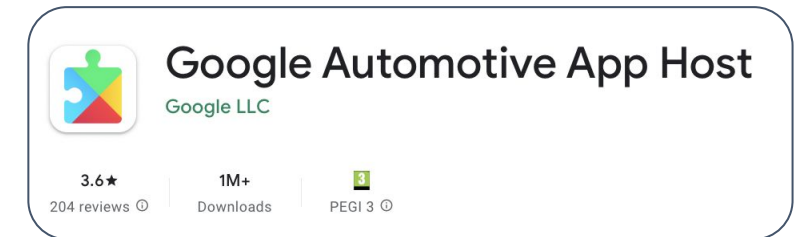
*Car App Library CAL is an API for app developers that provides standard UI **templates** which are optimized to not distract the driver.



The evolution of the Automotive App Host



- The initial public release was done in 2022
by Google
- Google regularly updates its GAS Host since then
- However, **the AOSP App Host has never been updated**
- ⇒ **There is now a large gap between the GAS Host and the AOSP host**
 - The GAS Host keeps adding templates and features
 - While the AOSP Host stays the same

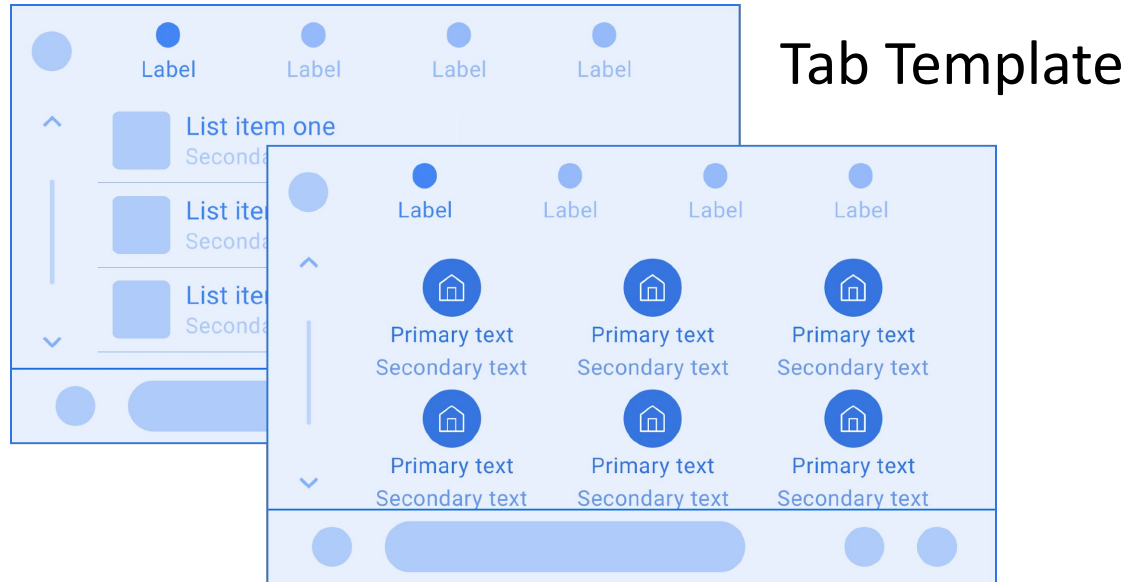
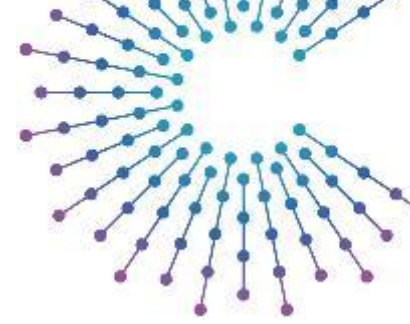


How do App developers interact with the Host?

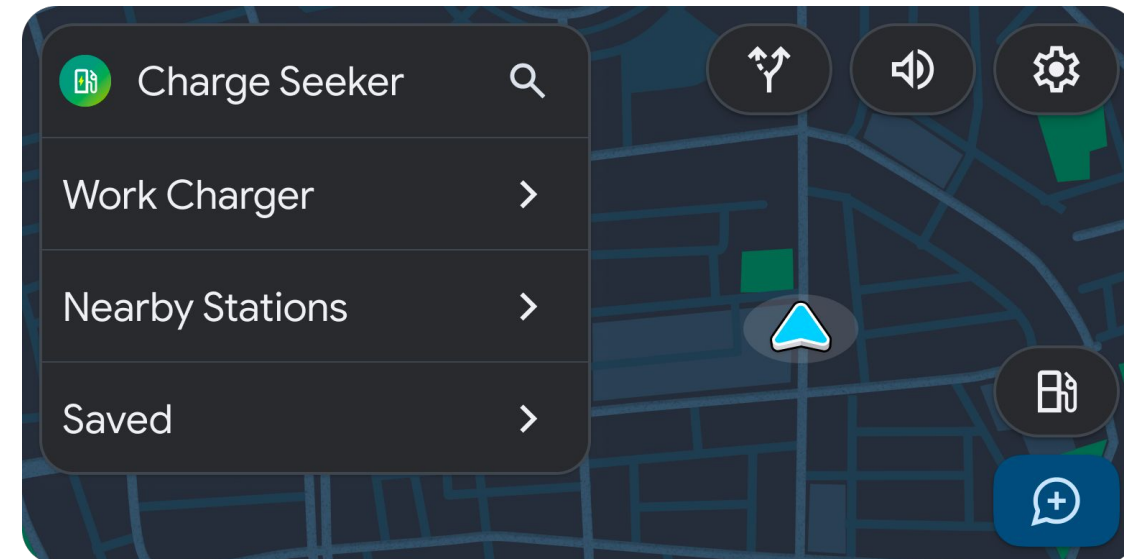
App developers implement the Car App Library in their app.

The CAL relies on the Host to render the UI of some templated elements, such as Maps.
The current CAL works seamlessly with the GAS host, but not with the AOSP that is outdated

Some templates supported by the GAS Host only



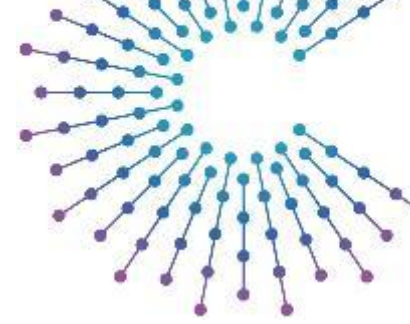
Map with Content Template



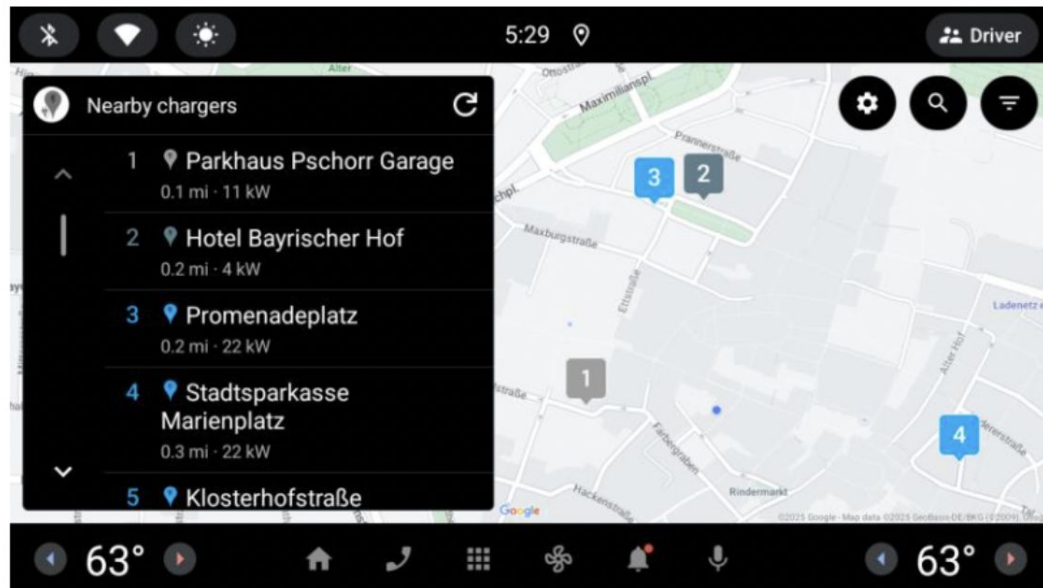
& more components...

What happens when an application is built targeting a newer App Host

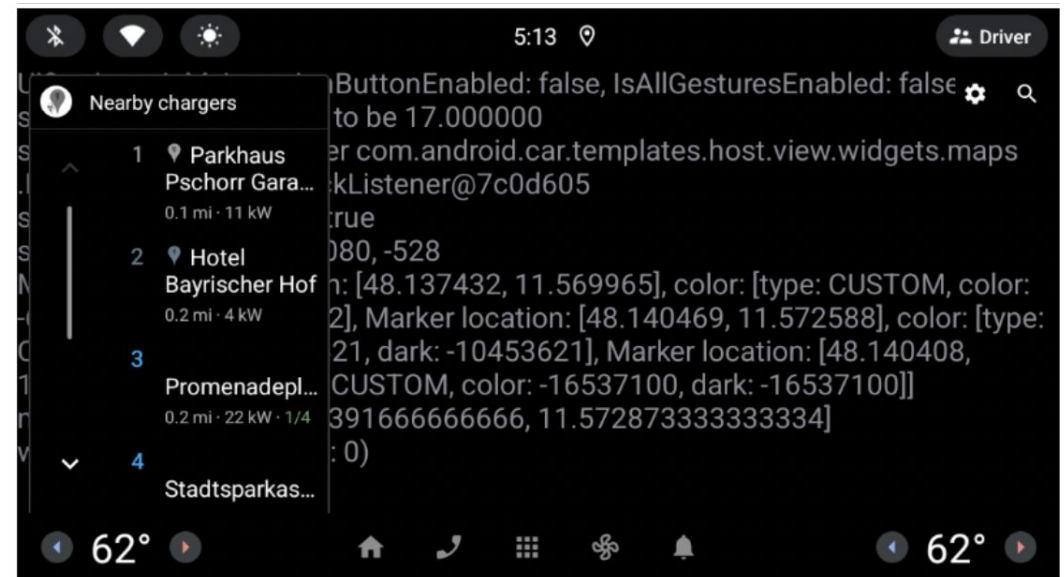
ie. when implementing a newer version the Car App Library



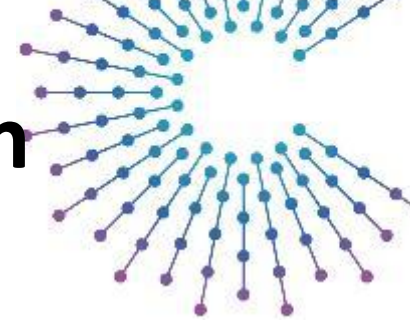
An app. rendered with the GAS Host 



The same app with the AOSP Host 

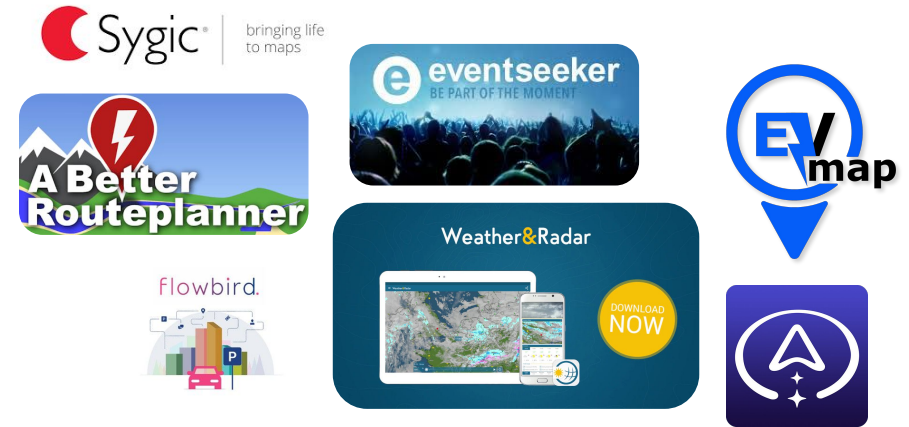


The Outdated AOSP App Host creates Fragmentation



For App Developers

- **65% of apps using the CAL cannot be published**
% of apps uploaded on the Appning portal that require a newer App Host
- They need to develop **2 versions**
One for the GAS host, one for the AOSP host



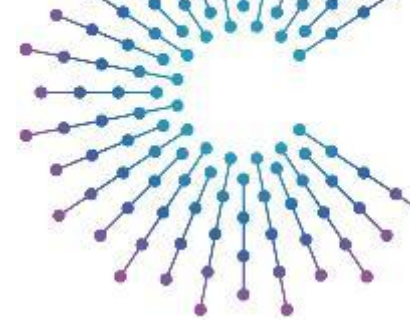
For OEMs

- **No up-to-date Host*** or costly development
**At the exception of BMW that introduced the map template*
- **Inconsistent and non-safe UX**

For App Store providers

- More manual testing of every application ⇒ **longer validation processes**

Proposal: Update the AOSP App Host to bridge the gap with the GAS App Host and Reduce Fragmentation



- Develop a COVESA AOSP Host
- Make it available to all OEMs
- Integrate it to the COVESA emulator for seamless testing of apps

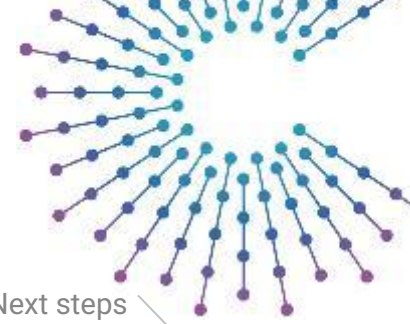
Benefit for App developers

- Develop one app only
- Test apps in the COVESA Emulator

Benefit for OEMs and App Store providers

- **Free Host** that can be installed in cars
⇒ *The Host would be distributed by the app store providers*
- **More applications** available in the stores
- **Safer** applications
- **Faster** app validation process

Project Proposal: Building a COVESA AOSP App Host



Preliminary work

Detailing the funds

Get Proposals

Board Approval

WP1: Start

Project setup

Implementation of the map template

Integration in COVESA emulator

WP2: CAL API support

Support Car App Library API up to level 6

WP3: CAL API support

Support Car App Library API up to level 7

Documentation for OEMs

Cont. Improvements

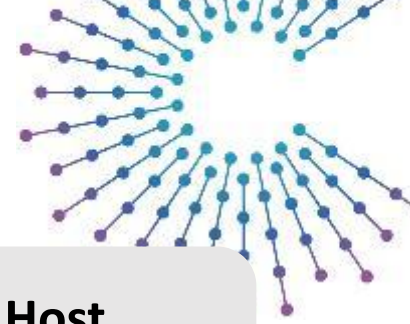
Maintenance, upgrade to upcoming CAL APIs

Maintaining compatibility with newer CAL API updates

⇒ Will require board approval

Needed today
 General Decision on project Launch
 How to proceed to raise the funds and to issue the tender

What are the risks of each options



Stop project: Not Updating the AOSP Host

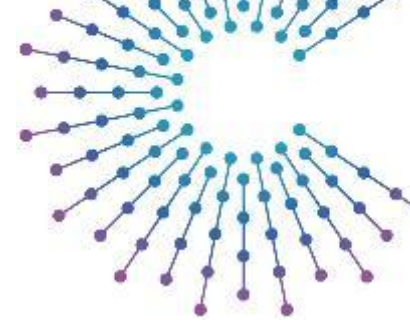
- Gap with GAS will increase
- **Maintain fragmentation**
- App developers to concentrate on GAS
- Less safe UX
- OEMs to each develop their own Host or keep stuck on the old one

Start project: Updating the AOSP Host

- **Google to start updating the AOSP host**
⇒ *Making our dev obsolete*
- Google to make the CAL more proprietary
By hardcoding dependencies
We would need to fork the CAL
- Google to change or drop the interface mechanism between the Host and the CAL

We strongly believe this project is essential to make the AOSP ecosystem competitive with GAS

Motivation



Motivation

- Fully featured support for templated apps in COVESA emulator
- Open-source fork of Template Host serves as base for integration into OEM systems
- Simplified customization for OEMs
- Visibility to 3rd party developers of templated apps
-> App store operators can refer to COVESA emulator as resource for testing
- CAL Samples provided in emulator simplify development with templates

Backup

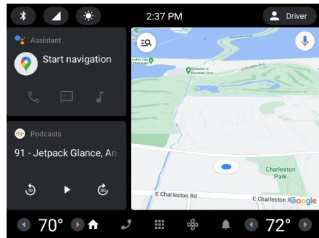
-

COVESA Template Host vs. Google Automotive App Host

The GAP - Analysis provided by PARADOXCAT

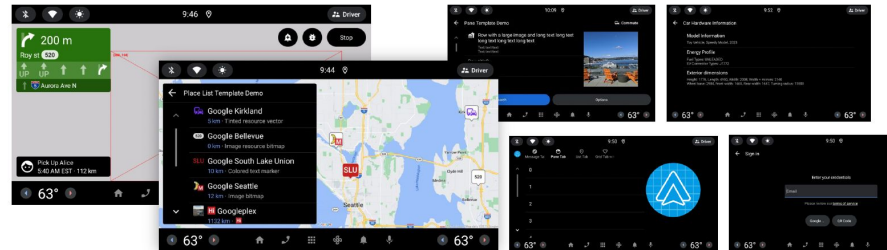
Android for Cars

- Targeting both Android Auto and Android Automotive OS (AAOS)
- Split by categories:
 - Media (audio) using **MediaBrowserService** and **MediaSession**
 - Messaging using **Notifications**
 - Navigation, POI, IoT, weather (+ experimental features for messaging, media) using **Car App Library**
 - Parked apps: video, games, browsers (AAOS only)



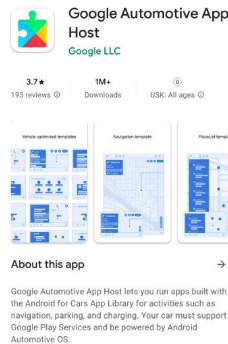
Car App Library

- Car App Library (CAL) is API towards Android application developers
- Standardized templates, distraction optimized
- It is open source, up-to-date, and documented
- Relies on a **Template Host** (or Android Auto) to render app's UIs on their behalf
- Some templates are restricted to certain app categories (navigation, POI)



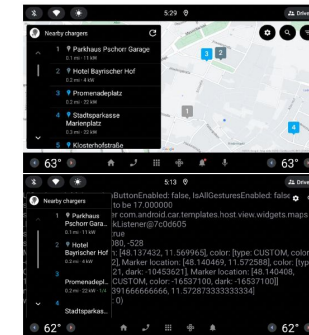
Template Host

- a.k.a. Automotive App Host / Templates Host
- Separate Android system application
- Responsible for rendering apps which use Car App Library
- Google Automotive App Host
 - Closed-source
 - Comes pre-installed with GAS and only works with GAS
 - Up-to-date, distributed via Play Store
- AOSP Templates Host
 - Open source in AOSP „[packages/apps/Car/Templates](#)“
 - First released in March 2022, **not updated since then**



Differences between GAS and AOSP Template Host

- Map implementation (GAS: Google Maps, AOSP: stub implementation using TextView)
- Increased content limits (e.g., maximum list length)
- Minor UI customizations (e.g., rounded buttons)
- Major new features in Car App Library since the AOSP release:
 - New templates:
 - API 5: **MapTemplate** (superseded by API 7 **MapWithContentTemplate**)
 - API 6: **TabTemplate** (can contain multiple other templates)
 - API 7: **MapWithContentTemplate** (self-rendered map for nav/POI/weather apps)
 - API 8 (experimental): **SectionedItemTemplate**, **MediaPlayerTemplate**
 - New features in existing templates:
 - API 5:
 - Refresh** in **PlaceListMap/PlaceListNavigationTemplate**
 - Up to 4 action buttons** in map templates
 - Enabled/disabled state** for Action, Row, Toggle
 - Navigation apps: **SuggestionManager**, **Alert**, **map click listener**
 - API 6:
 - App-driven refresh**
 - Top-level actions (FAB)** for ListTemplate & GridTemplate
 - Actions & badges** for ListTemplate rows
 - Navigation apps: map rendering on **instrument cluster** (optional)
 - API 7:
 - ListTemplate: **ConversationItem** for messaging apps (unclear if relevant for AAOS?)
 - Reusable **Header** component for multiple templates
 - API 8 (experimental):
 - GridTemplate: **badge, size, image shape** (circle)



COVESA Template Host vs. Google Automotive App Host

Proposal for a Solution - *Analysis provided by PARADOXCAT*

Proposal: Open Source COVESA fork of Template Host



1. Project setup, upgrade to latest Gradle & dependencies
2. Implement maps
 - Maps SDK - our proposal: [MapLibre](#) (permissively licensed fork of Mapbox SDK)
 - Map data hosting needed (API key)
3. Integrate Template Host (and CAL samples) into COVESA emulator
4. Update Host to support latest Car App API Levels, implement new features & renderers
 - Gradually add support for API level 5 -> 6 -> 7 (-> 8)
 - Test suite:
 - Google's Car App Library samples
 - Common 3rd party apps
 - If needed, build our own test app?
5. Documentation & services for OEMs (how to fork, what can/needs to be customized)
6. Future maintenance

Rough estimation



1. Project setup, upgrade to latest Gradle & dependencies
 2. Implement maps
 3. Integrate Template Host (and CAL samples) into COVESA emulator: 1 week
 4. Update Host to support latest Car App API Levels, implement new features & renderers
 - API Level 5: 8 weeks
 - API Level 6: 8 weeks
 - API Level 7: 4-6 weeks
 - (API Level 8: TBD)
 5. Documentation for OEMs: 2 weeks
 6. Future maintenance:
 - New API level – depends on complexity, see above
 - General maintenance, bugfixing
- } 2 weeks

Benefits

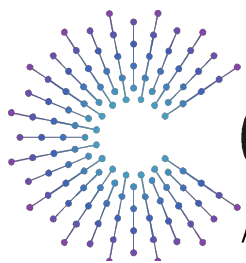


- Fully-featured support for templated apps in COVESA emulator
- Open Source fork of Template Host in COVESA GitHub serves as base for integration into OEM systems
- Simplified customization for OEMs using documentation and services
 - Custom map implementation
 - UI theme customization
 - Modification of content limits
- Visibility to 3rd party developers of templated apps
-> App store operators can refer to COVESA emulator as resource for testing
- CAL Samples provided in emulator simplify development with templates

Risks



1. Google releases updated AOSP Template Host and continues maintaining it
 - Our fork would likely be obsolete - might not be needed anymore or would need to be updated
2. Google hardcodes GAS dependencies into Car App Library
 - We would need to fork the Car App Library
 - Fragmentation :(
3. Google changes connection mechanism between Car App Library and Template Host
 - If the change is too complex, it may be hard to adapt our Template Host (because we would only see the client-side changes, not the host changes)
 - This is rather unlikely due to backwards compatibility promise
4. Google abandons and replaces the Car App Library / Template Host mechanism
 - Our solution will continue to work, but app developers might switch to the new solution



COVESA

Accelerating the future of connected vehicles

