



## Android Automotive SIG - Vehicle Data APIs / VHAL

**Alexander Domin from BMW**

**Stefan Wysocki from TietoEVRY**

GENIVI Virtual Technical Summit | May 13, 2020

The content on this presentation is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)



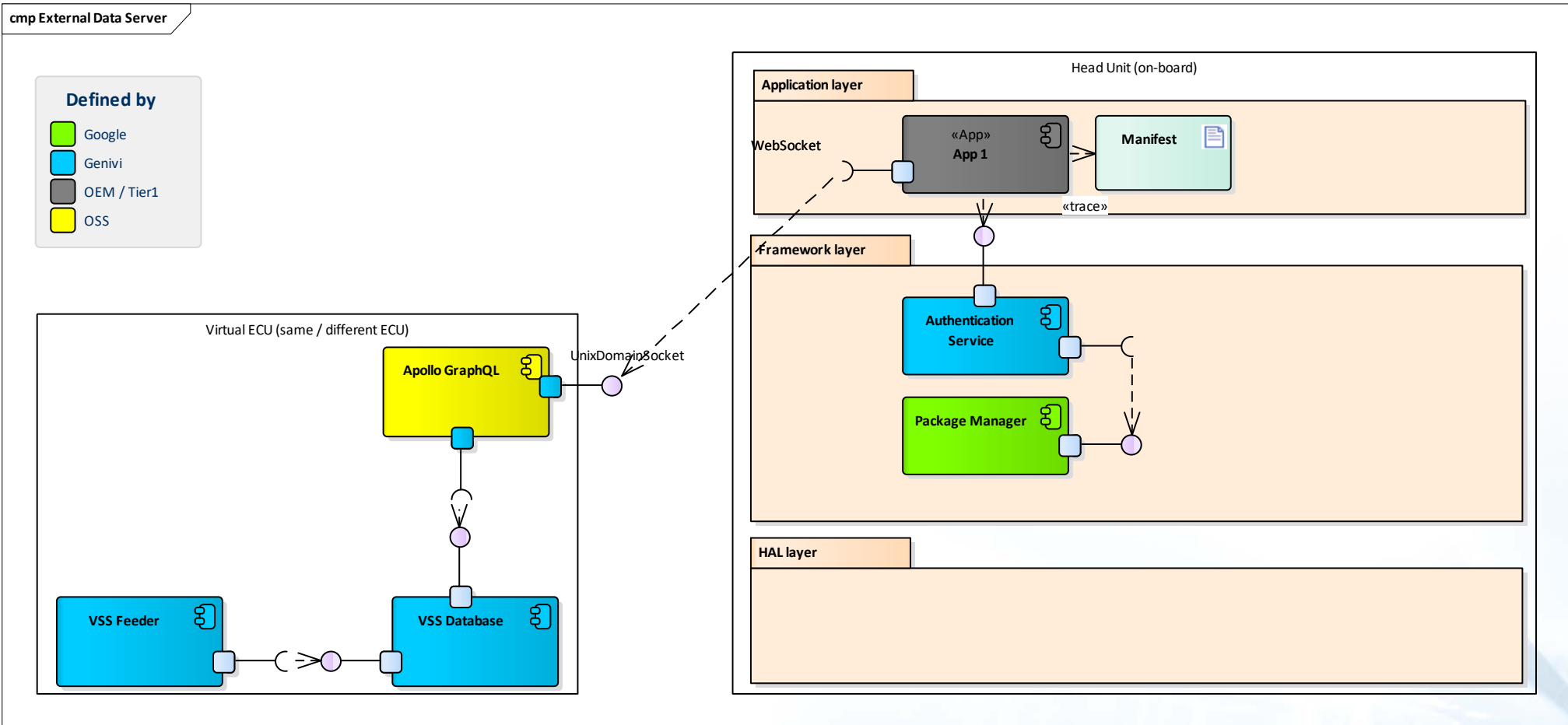
## Workshop Agenda

- Project overview & proof-of-concept demo
- Topics discussion
  - Google Vehicle Properties Implementation based on GraphQL Service.
  - Permission groups specification.
  - Translation of permission groups.
  - JWT Token what will be included and how it will be done? And generation process?
- Feature content definition for next milestones
- Technical readiness level assessment and discussion on how and when to reaching out to Google



# Android Automotive SIG - Project overview & proof-of-concept demo

Get common understanding on the External Data Server concept



# Android Automotive SIG - Project overview & proof-of-concept demo

## Request for contributions



Work Breakdown	Current contributors
Create a <b>VSS feeder</b> concept addressing <b>dynamic connection to car network IPCs</b> (e.g. CAN or/and SomIP) and specify required <b>VSS deployment extensions</b> .	Stefan Wysocki (TietoEVERY)
Create a <b>VSS Database</b> concept addressing the <b>storage</b> of the data received from different car networks including <b>cache</b> strategy.	
Create a <b>VSS data server</b> concept addressing app <b>authentication</b> , data access <b>authorization</b> based on permission groups, to be able to <b>query vehicle data</b> .	Alexander Domin (BMW)
Create a <b>Authentication Service</b> concept addressing token generation and secure key storage.	Stefan Wysocki (TietoEVERY)
Create a Data Server access concept addressing <b>token generation process</b> and a <b>typesafe API</b> to work with GraphQL servers by implementing a sample <b>Android APP</b> .	

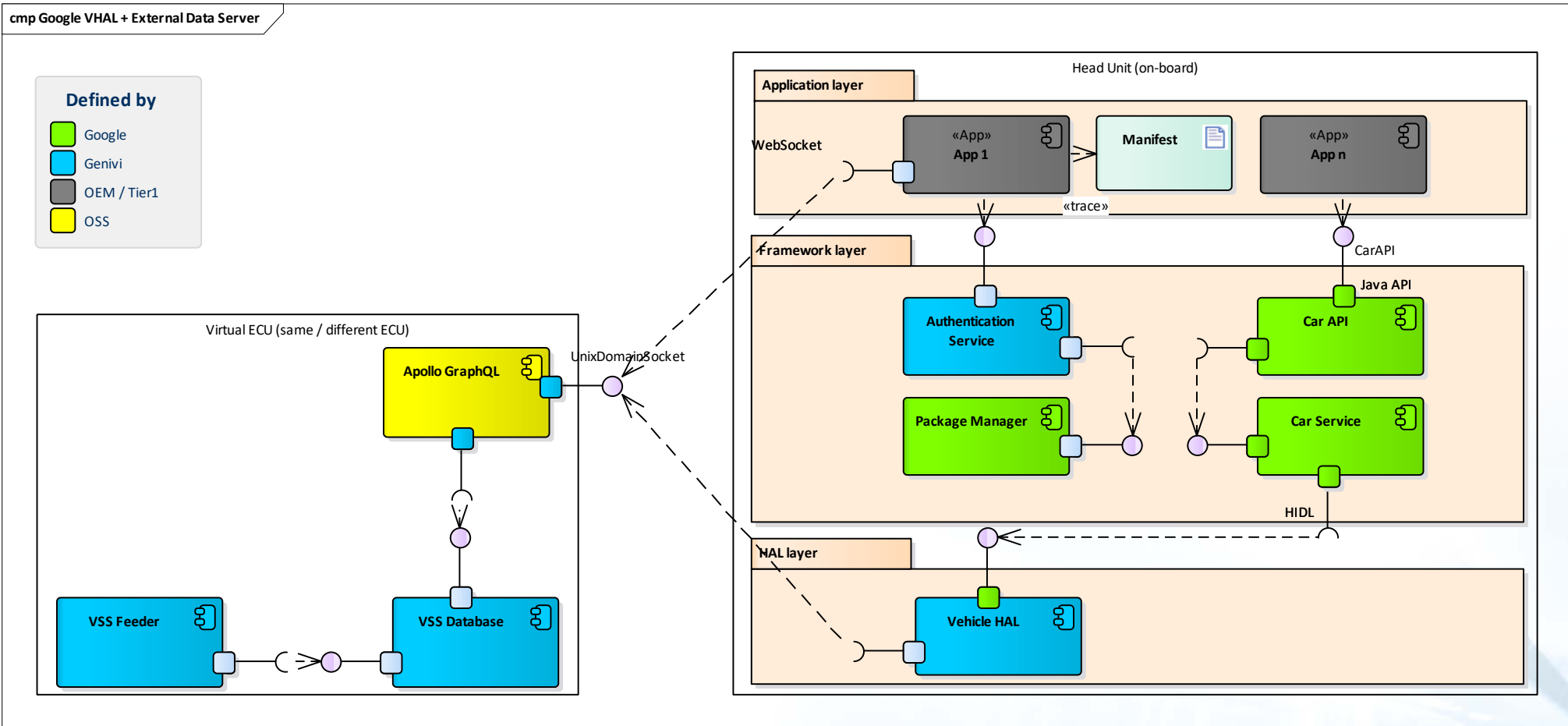
# Android Automotive SIG - Project overview & proof-of-concept demo

## Showcase Proof-of-Concept Demo



# Android Automotive SIG - Google Vehicle Properties Implementation

Get common understanding on the GENIVI Google VHAL concept



# Android Automotive SIG - Google Vehicle Properties Implementation

## Identification of work packages for the Proof-of-Concept



Work Breakdown	Current Contribution
VSS leaf to Google Vehicle properties mapping. How and where to do the mapping?	
Howto deal with inconsistency between the standards Google → FUEL_LEVEL (in „milliliters“) & INFO_FUEL_CAPACITY (in „milliliters“) VSS → Vehicle.Drivetrain.FuelSystem.Level ( in „percent“) & Vehicle.Drivetrain.FuelSystem.TankCapacity (in „liter“)	
Howto deal with Authentification of the VHAL implementation against GraphQL Server?	
Create the VHAL component	

# Android Automotive SIG - Permission groups specification



## Share experience made specifying permission groups

The AASIG agreed to Introduce permission groups to limit access to vehicle data, aligning to the concept provided by Android

e.g.:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
  package="org.genivi.ambientlight">
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission android:name="android.car.permission.CAR_IDENTIFICATION" />
  <uses-permission android:name="vss.permission.AMBIENTLIGHTING_READ" />
  ...
</manifest>
```

### Open questions:

- Using which syntax permission groups shall be specified?
- The right location to maintain this specification?
- How permission groups will be handled in our architecture?



# Android Automotive SIG - Permission groups specification



## Concept of directives as a permission validation in GraphQL.

- A directive is an identifier preceded by a **@** character, optionally followed by a list of named arguments, which can appear after almost any form of syntax in the GraphQL query or schema languages.
- Imagine “**@hasPermissions**” directive that takes an argument “**permissions**”, which contains a list of permissions. This “**@hasPermissions**” directive can appear on individual fields, to enforce field-specific “**@hasPermissions**” restrictions:

e.g. GraphQL Schema example:

```
""" Ambient lighting signals and sensors """
```

```
type Vehicle_Cabin_Lights_AmbientLighting {
```

```
    """ Switches all ambient lighting effects in the cabin on (true) and off (false). """
```

```
    isLightingOn: Boolean
```

```
    @hasPermissions(permissions: ["*_READ"])
```

```
}
```

# Android Automotive SIG - Permission groups specification



## Definition of permission groups in yaml deployment files

### VSS specification file (\*.vspec):

- Vehicle.Cabin.Lights.AmbientLighting.**IsLightingOn**:  
type: actuator  
datatype: **Boolean**  
description: **Switches all ambient lighting effects in the cabin on (true) and off (false).**

### VSS deployment file (\*.depl):

- Vehicle.Cabin.Lights.AmbientLighting.**IsLightingOn**:  
permissions:
  - vss.permission.AMBIENTLIGHTING\_READ
  - vss.permission.AMBIENTLIGHTING\_WRITE
  - vss.permission.VEHICLE\_READ

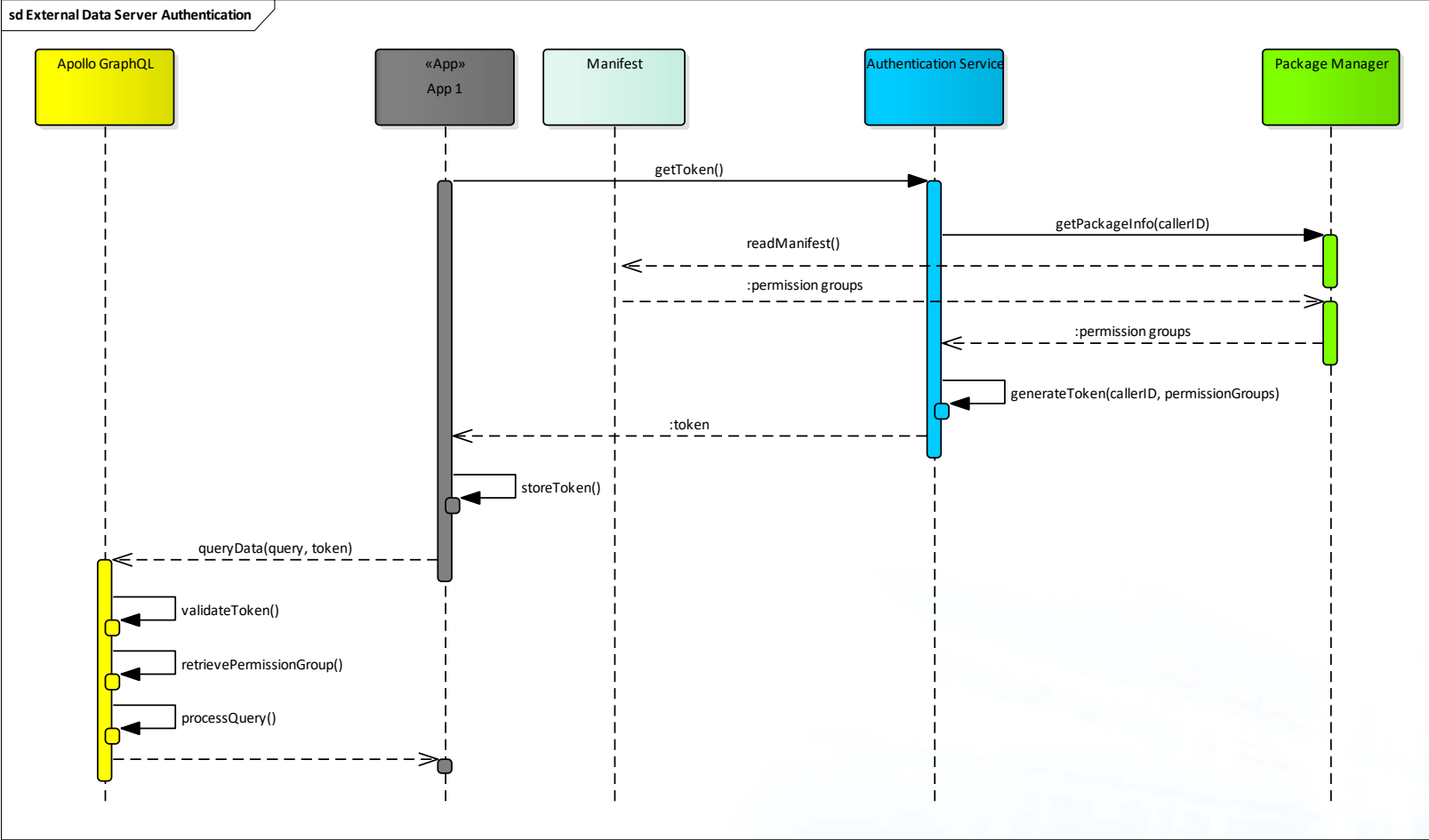
### GraphQL schema file:

```
type Vehicle_Cabin_Lights_AmbientLighting {  
  """ Switches all ambient lighting effects in the cabin on (true) and off (false). """  
  isLightingOn: Boolean  
  @hasPermissions( permissions: [ "vss.permission.AMBIENTLIGHTING_READ",  
    „vss.permission.AMBIENTLIGHTING_WRITE“,  
    „vss.permission.VEHICLE_READ“ ] )  
}
```

# Android Automotive SIG – JSON web token generation process



## Data Server driven handling of permission groups



# Android Automotive SIG - Permission groups specification



## Definition of permission groups in a json files

### VSS specification file (\*.vspec):

- Vehicle.Cabin.Lights.AmbientLighting.IsLightingOn:  
type: actuator  
datatype: **Boolean**  
description: **Switches all ambient lighting effects in the cabin on (true) and off (false).**

### GraphQL schema file:

```
type Vehicle_Cabin_Lights_AmbientLighting {  
  """ Switches all ambient lighting effects in the cabin on (true) and off (false). """  
  isLightingOn: Boolean  
  @hasPermissions( permissions: [„Vehicle.Cabin.Lights.AmbientLighting.IsLightingOn_READ“,  
    „Vehicle.Cabin.Lights.AmbientLighting.IsLightingOn_WRITE“  
  ] )  
}
```

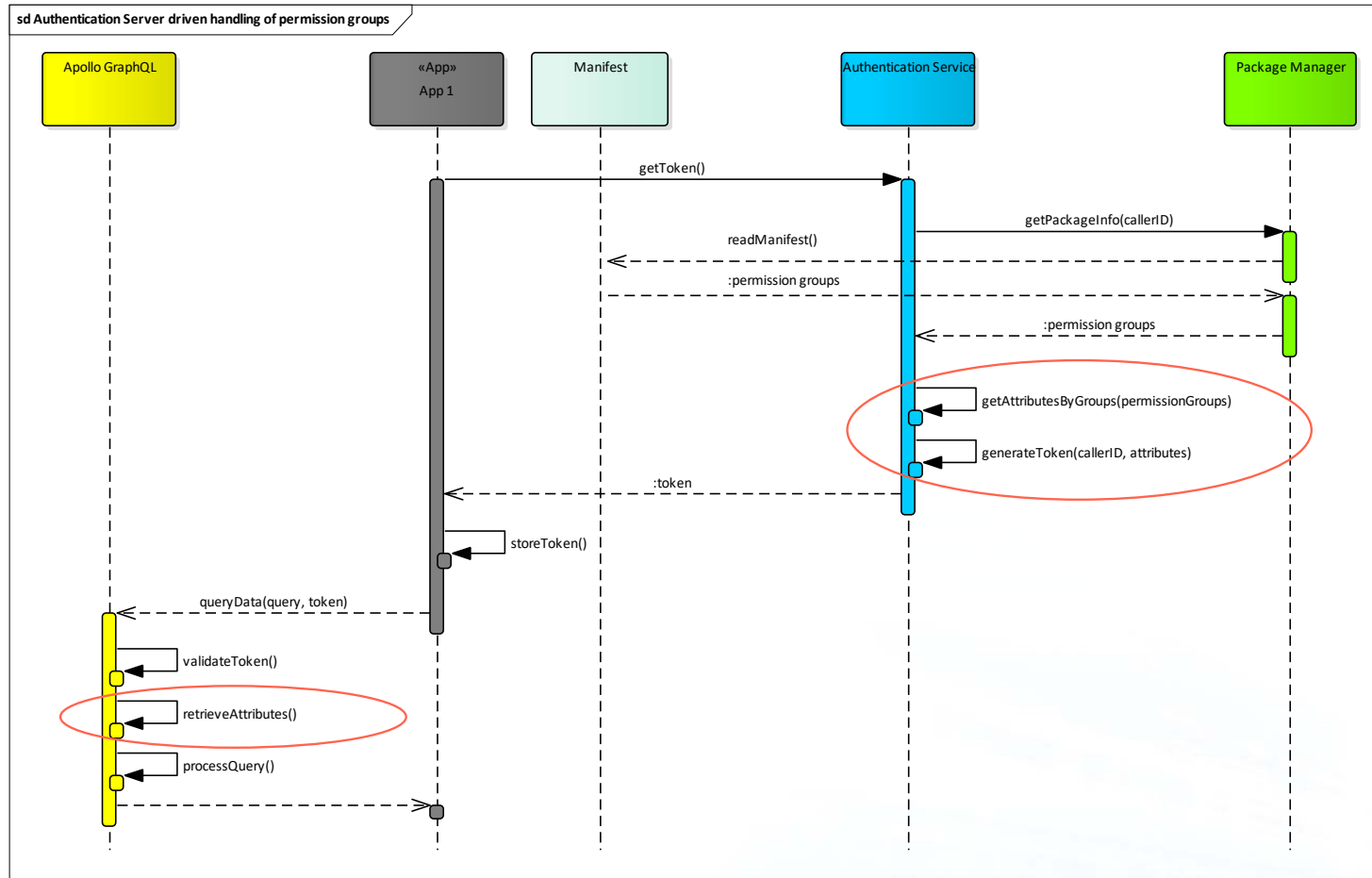
### Permission groups file (\*.json):

```
{  
  "vss.permission.AMBIENTLIGHTING_READ": {  
    "description": "Permission group description",  
    "protection_level": "normal",  
    "attributes": [  
      "Vehicle.Cabin.Lights.AmbientLighting.IsLightingOn_READ"  
      "Vehicle.Cabin.Lights.AmbientLighting.IsDimmedAtNightOn_READ"  
    ] }  
}
```

# Android Automotive SIG – JSON web token generation process



## Authentication Service driven handling of permission groups



# Timeline

## Timeline

- Milestone 1 – **GENIVI Virtual Technical Meeting** (12-14 May)
- Milestone 2 - Internal milestone (early Q3 - July)
- Milestone 3 - Fall All Member Meeting, Leipzig, Germany (last week of October)
- Milestone 4 - CES 2021, Las Vegas, USA (early January 2021)



# Contributing



- **Weekly telcos**

- Tuesdays – 17:00 CET (US friendly time) – Vehicle Data APIs / VHAL

- Thursday - 11:30 CET (India & Asia friendly time) – Audio HAL

- Mailing list: [http://genivi.emwd.com/mailman/listinfo/aa-sig\\_lists.genivi.org](http://genivi.emwd.com/mailman/listinfo/aa-sig_lists.genivi.org)

- **Wiki**

- Android Automotive Project Wiki : <https://at.projects.genivi.org/wiki/x/XgA4Ag>

- Vehicle Data Access / VHAL - External Data Server Proof-of-Concept Work Breakdown Wiki  
<https://at.projects.genivi.org/wiki/x/RgXYAg>

- Audio HAL - System Level Audio Wiki

- <https://at.projects.genivi.org/wiki/x/BAlyAw>

# Thank you!

**Visit GENIVI:**

<http://www.genivi.org>

<http://projects.genivi.org>

**Contact us:**

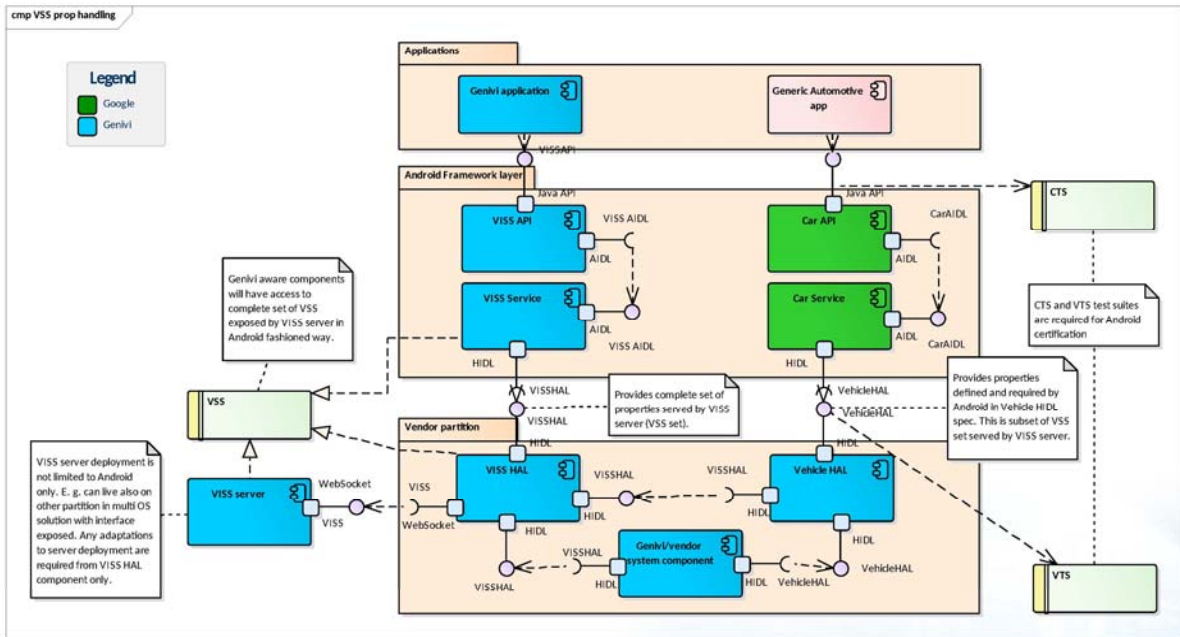
[help@genivi.org](mailto:help@genivi.org)





# Vehicle Data Access via Customized HAL - Google VHAL + OEM Extensions inside

## Architectural proposal I (via custom HAL)



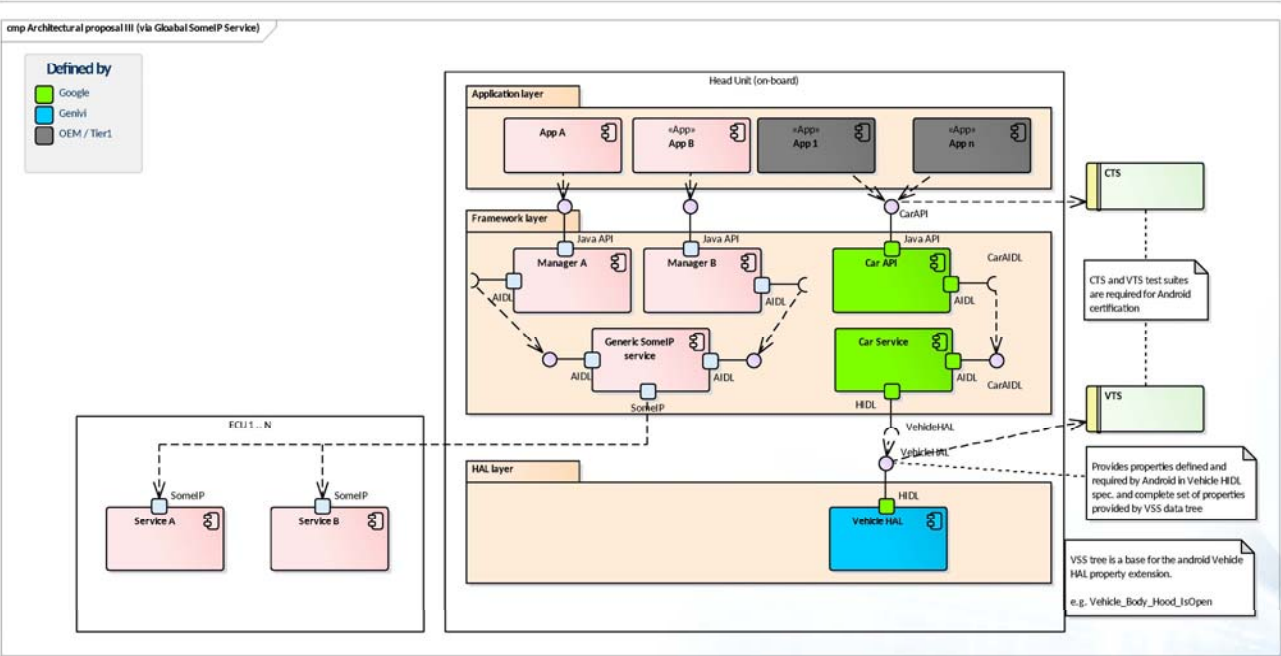
29 October 2019 | Copyright © GENIVI Alliance 2019

# Vehicle Data Access – via Some/IP Service - SomeIP stack inside the Framework



Look at vsomeip port to AOSP : <https://github.com/GENIVI/vsomeip/pull/107>

## Architectural proposal III (via Global SomeIP Service)



29 October 2019 | Copyright © GENIVI Alliance 2019

14