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

Defined by
- Google
- Genivi
- OEM / Tier1
- OSS

Virtual ECU (same / different ECU)
- Apollo GraphQL
- VSS Feeder
- VSS Database

Application layer
- WebSocket
- Manifest
- «App»
- «trace»

Framework layer
- Authentication Service
- Package Manager
- UnixDomainSocket
- WebSocket
- Package Manager

HAL layer

Component External Data Server

May 2020 | Copyright © GENIVI Alliance 2020
## Android Automotive SIG - Project overview & proof-of-concept demo

### Request for contributions

<table>
<thead>
<tr>
<th>Work Breakdown</th>
<th>Current contributors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a <strong>VSS feeder</strong> concept addressing <strong>dynamic connection to car network IPCs</strong> (e.g. CAN or/and SomeIP) and specify required <strong>VSS deployment extensions</strong>.</td>
<td>Stefan Wysocki (TietoEVRY)</td>
</tr>
<tr>
<td>Create a <strong>VSS Database</strong> concept addressing the <strong>storage</strong> of the data received from different car networks including <strong>cache</strong> strategy.</td>
<td></td>
</tr>
<tr>
<td>Create a <strong>VSS data server</strong> concept addressing app <strong>authentication</strong>, data access <strong>authorization</strong> based on permission groups, to be able to <strong>query vehicle data</strong>.</td>
<td>Alexander Domin (BMW)</td>
</tr>
<tr>
<td>Create a <strong>Authentication Service</strong> concept addressing token generation and secure key storage.</td>
<td>Stefan Wysocki (TietoEVRY)</td>
</tr>
<tr>
<td>Create a Data Server access concept addressing <strong>token generation process</strong> and a <strong>typesafe API</strong> to work with GraphQL servers by implementing a sample <strong>Android APP</strong>.</td>
<td></td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th>Work Breakdown</th>
<th>Current Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>VSS leaf to Google Vehicle properties mapping.</td>
<td></td>
</tr>
<tr>
<td>How and where to do the mapping?</td>
<td></td>
</tr>
<tr>
<td>How to deal with inconsistency between the standards</td>
<td></td>
</tr>
<tr>
<td>Google → FUEL_LEVEL (in „milliliters“) &amp; INFO_FUEL_CAPACITY (in „milliliters“)</td>
<td></td>
</tr>
<tr>
<td>Drivetrain.FuelSystem.TankCapacity (in „liter“)</td>
<td></td>
</tr>
<tr>
<td>How to deal with authentication of the VHAL implementation against GraphQL Server?</td>
<td></td>
</tr>
<tr>
<td>Create the VHAL component</td>
<td></td>
</tr>
</tbody>
</table>
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
<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):

  type: actuator
datatype: Boolean
description: Switches all ambient lighting effects in the cabin on (true) and off (false).

VSS deployment file (*.depl):

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"] )
}

May 2020 | Copyright © GENIVI Alliance 2020
Android Automotive SIG – JSON web token generation process

Data Server driven handling of permission groups

Apollo GraphQL

«App»
App 1

Manifest

Authentication Service

Package Manager

getPackageInfo(callerID)

readManifest()

:permission groups

generateToken(callerID, permissionGroups)

validateToken()

:getToken()

storeToken()

:token

retrievePermissionGroup()

processQuery()

queryData(query, token)

processQuery()
Android Automotive SIG - Permission groups specification

Definition of permission groups in a json files

VSS specification file (*.vspec):
  type: actuator
datatype: Boolean
description: Switches all ambient lighting effects in the cabin on (true) and off (false).

GraphQL schema file:

```graphql
type Vehicle_Cabin_Lights_AmbientLighting {
  """" Switches all ambient lighting effects in the cabin on (true) and off (false). """
  isLightingOn: Boolean
}
```

Permission groups file (*.json):

```json
{
  "vss.permission.AMBIENTLIGHTING_READ": {
    "description": "Permission group description",
    "protection_level": "normal",
    "attributes": [
    ]
  }
}
```
Android Automotive SIG – JSON web token generation process

Authentication Service driven handling of permission groups

- Apollo GraphQL
- «App» App 1
- Manifest
- Authentication Service
- Package Manager

*GetToken()*

*validateToken()*

*retrieveAttributes()*

*getPackageInfo(callerID)*

*permission groups*

*permission groups*

*getAttributesByGroups(permissionGroups)*

*generateToken(callerID, attributes)*

*processQuery()*

*retrieveData(query, token)*

*storeToken()*
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


- **Wiki**
  - Android Automotive Project Wiki: [https://at.projects.genivi.org/wiki/x/XgA4Ag](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](https://at.projects.genivi.org/wiki/x/RgXYAg)

  - Audio HAL - System Level Audio Wiki
    [https://at.projects.genivi.org/wiki/x/BAlyAw](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
Vehicle Data Access via Customized HAL - Google VHAL + OEM Extensions inside

Architectural proposal I (via custom HAL)
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