Public

Permissions of custom properties

Vendor extension of the set

Stefan Wysocki

stefan.wysocki@tieto.com

tieto

Table of contents

Background

tieto

Stories

- As a User I want to have the access to wide range of compatible applications to be installed on IVI
- As a 3rd party Developer I want to create my application once and port it into multiple platforms in easy way
- As a OEM I want to provide as much properties as I want
- As a OEM I want to control and restrict the accesses of the provided data

Way to extend the set

- Extend android.hardware.automotive.vehicle@2.0 interface

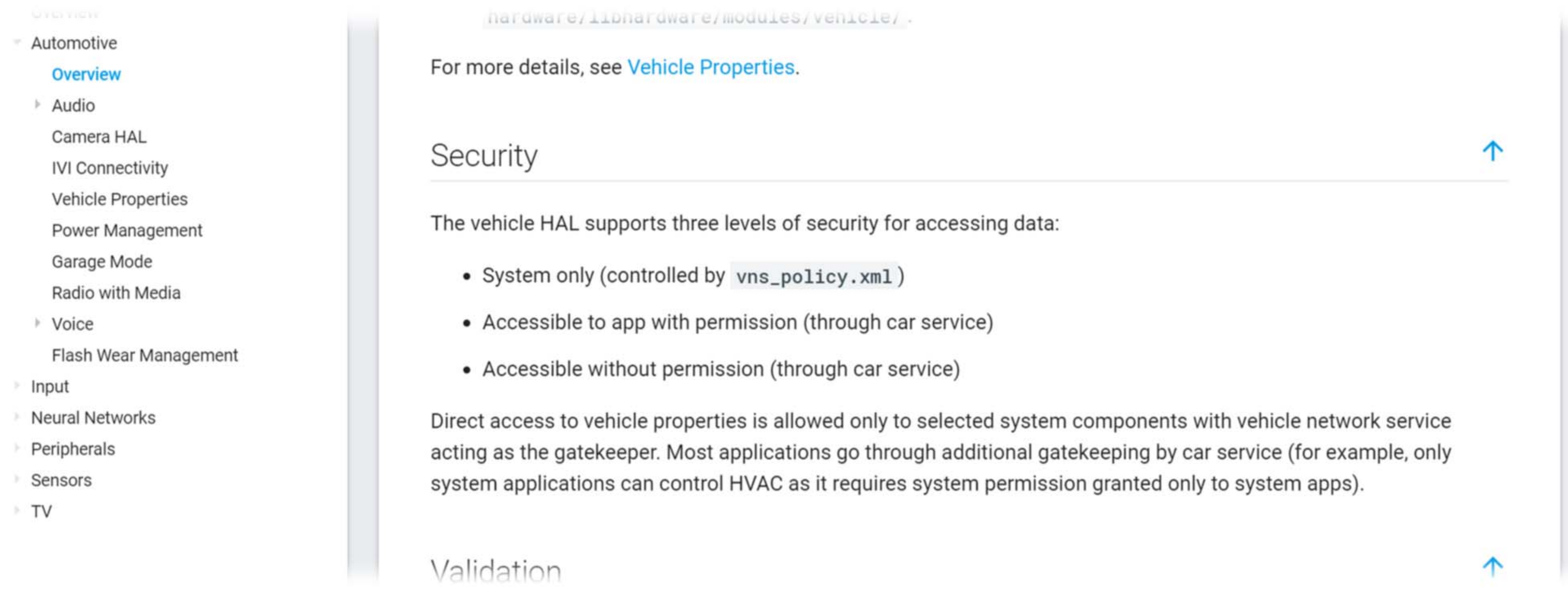
```
enum VehicleProperty: @2.0::VehicleProperty {
        VENDOR_FOO= (
         0xf100
         | VehiclePropertyGroup:VENDOR
         | VehiclePropertyType:STRING
         | VehicleArea:GLOBAL),
};

// types.hal
```

Public

AOSP solution

Accessing Vehicle Data

Documentation



hardware/libhardware/modules/vehicle/ .

For more details, see Vehicle Properties.

Security

The vehicle HAL supports three levels of security for accessing data:

- System only (controlled by `vns_policy.xml`)
- Accessible to app with permission (through car service)
- Accessible without permission (through car service)

Direct access to vehicle properties is allowed only to selected system components with vehicle network service acting as the gatekeeper. Most applications go through additional gatekeeping by car service (for example, only system applications can control HVAC as it requires system permission granted only to system apps).

Validation

https://source.android.com/devices/automotive#security

tieto

1^{st} level: System only

- controlled by vns_policy.xml
  - Last seen Android 7.1
  - Whole „Vehicle Network Service Concept” removed

```
Remove Vehicle Network Service

It was replaced by Vehicle HAL

Bug: b/34361484

Test: removing dead code
Change-Id: Ib6f9641bc98d53ea1fe87d245aa7792be5650532
```

https://android.googlesource.com/platform/packages/services/Car/+/41aa219%5E%21/

2nd level: Accessible to app with permission

- through car service
- Defined static set of Permissions for Default Properties [1]
  - Some of those annotated with @SystemApi/@hide
- Mapped such permission to properties [2]/[3]
- Will be replaced by PropertyPermissionMapping? Preparation of custom Permissions? [4]

[1] /packages/services/Car/car-lib/src/android/car/Car.java
[2] /packages/services/Car/service/src/com/android/car/hal/PropertyHalServiceIds.java
[3] /packages/services/Car/service/AndroidManifest.xml
[4] /packages/services/Car/car-lib/src/com/android/car/internal/PropertyPermissionMapping.java

Current state of Vendor Extension

```
    <!-- Allows an application to access the vehicle vendor channel to exchange vendor-
specific information.
         <p>Protection level: signature|privileged
    -->
    <permission
        android:name="android.car.permission.CAR_VENDOR_EXTENSION"
        android:protectionLevel="signature|privileged"
        android:label="@string/car_permission_label_vendor_extension"
        android:description="@string/car_permission_desc_vendor_extension" />
```

PropertyPermissionMapping.java

```java
@Nullable
private Permission getPermission(int propId) {
    return isVendorExtension(propId) ? VENDOR_PERMISION : mPermissions.get(propId);

}

private static boolean isVendorExtension(int propId) {
    return (propId & VENDOR_MASK) == VENDOR_MASK;
}
```

Permission check

- App -> CarPropertyManager -> CarPropertyService:

```java
// CarPropertyService.java
@Override
public CarPropertyValue getProperty(int prop, int zone) {
    if (mConfigs.get(prop) == null) {
        // Do not attempt to register an invalid propId
        Log.e(TAG, "getProperty: propId is not in config list:0x" + toHexString(prop));
        return null;
    }
    ICarImpl.assertPermission(mContext, mHal.getReadPermission(prop));
    return mHal.getProperty(prop, zone);
}
// ICarImpl.java
public static void assertPermission(Context context, String permission) {
    if (context.checkCallingOrSelfPermission(permission) != PackageManager.PERMISSION_GRANTED) {
        throw new SecurityException("requires " + permission);
    }
}
```

3rd level: Accessible without permission

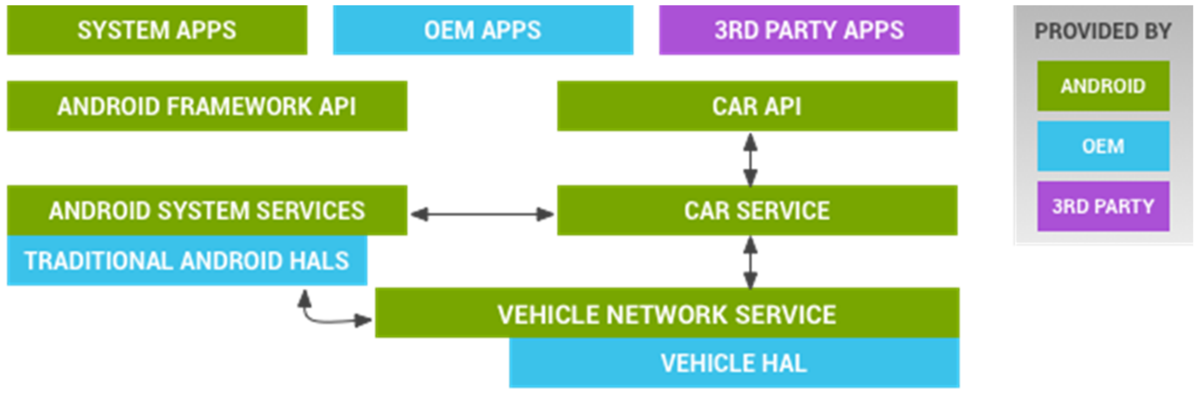
- (through car service)?
- A chance for the vendor permission specialization?

Public

Side notes

Documentation

- For automotive: poorly maintained



Automotive Vehicle HAL changes

- Be aware:
  - VHal .hal definition not frozen yet!

AUTOMOTIVE.VEHICLE@2.0 ON ANDROID P

!=

AUTOMOTIVE.VEHICLE@2.0 ON ANDORID Q

```
commit 62fd03d8e64f368fadce4a972ead88404f94b21d
Author: Scott Randolph <randolphs@google.com>
Date:   Mon Jun 4 14:16:21 2018 -0700

    Explicitly unfreeze the Automotive HALs

    Per discussion with Treble team, keeping HALs unfroze
n is suitable for
    platforms whos functionality is evolving and which do
 not require hard
    gaurantees of cross version vendor/system interoperab
ility.

    Test: build owl
    Bug: 109674659
    Change-Id: Idc2a391b4bd7c2edbf9cdd7fc099b9d4a6fcf95a
```

Public

tieto

Stefan Wysocki

Senior Software Engineer
stefan.wysocki@tieto.com