Track 1

Use case:

- Application wants to read the current fuel level and the tank capacity

Requirements:

- Read access for signals needs to be protected by a permission: `org.genivi.vss.permission.FUEL_SYSTEM_READ`
- Permission needs to be granted by an authority which is secure
- Data needs to be defined and structured according to VSS
- Data needs to be accessible by the framework as well
1. App requests the token
2. Auth. Service querying PM for granted permissions and includes it into the token
3. App sends a query for VSS leaves
4. Server checks which permissions are needed for given leaves and compares it with the ones provided in token

VSS Feeder constantly supplies the DB with VSS structured properties from OpenDS Simulator
Track 2

Android defined properties + VSS as "vendor property"

AOSP format

VSS format
**Track 2 - vehicle HAL implementation**

- Flat the VSS hierarchy
- Map some of the "fields" to the one supported by hidl
- Loose "branch" type for grouping the properties
- Translation module for properties that are defined differently by both standards (translation of units or datatypes)

```python
Vss2android.py
```

```python
# Tire
#
- Tire:
  type: branch
  description: Tire signals for wheel

- Tire.Pressure:
  datatype: uint8
type: sensor
  unit: kpa
  description: Tire pressure in kilo-Pascal
```

```python
TIRE_PRESSURE = (0x0309 |
  | VehiclePropertyGroup:SYSTEM |
  | VehiclePropertyType:FLOAT |
  | VehicleArea:WHEEL),
```
Contributing

Weekly telcos:
Tuesdays – 17:00 CET (US friendly time) – Vehicle Data APIs / VHAL
Android Automotive Project Wiki:
https://at.projects.genivi.org/wiki/x/XgA4Ag
Thank you!

Contact W3C Transport and Automotive groups:
ted@w3.org
https://www.w3.org/auto/

Visit GENIVI:
http://www.genivi.org
http://projects.genivi.org