

# Vehicle Simulators as example data providers

Genivi Vehicle Simulator and OpenDS

Cloud & Connected Services workshop

GENIVI Virtual Technical Summit | 14 May 2020

## Overview

# Original purpose of this investigation

- To check if the simulators are applicable as a presentation layer and „real” data provider for Android Framework and Applications
- Invent some usecases and verify it against the ones agreed in face-to-face workshop
- Highlight the problems or identify blockers that need to be resolved together
- Kickoff VSS Feeder implementation!

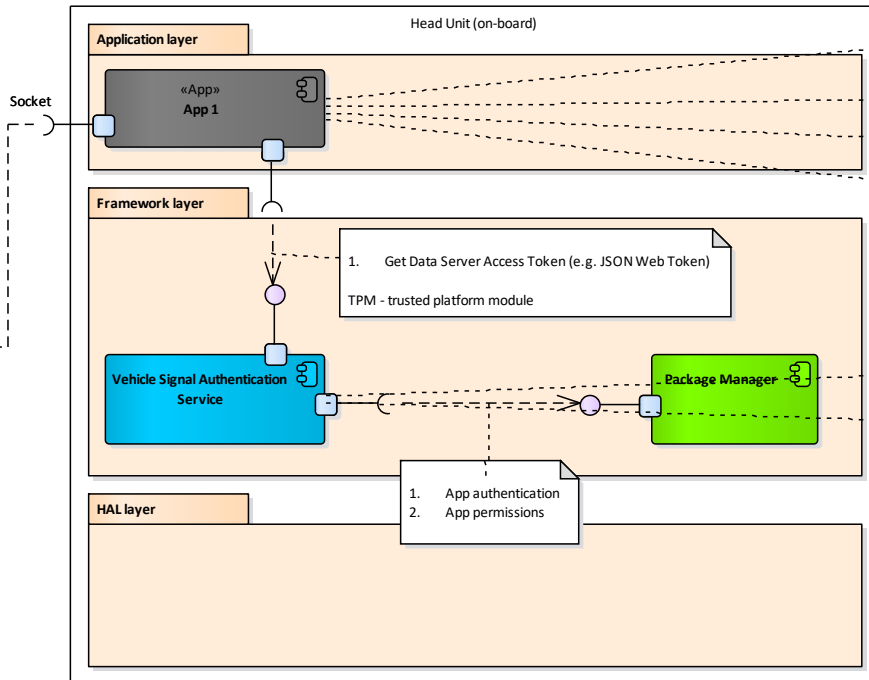
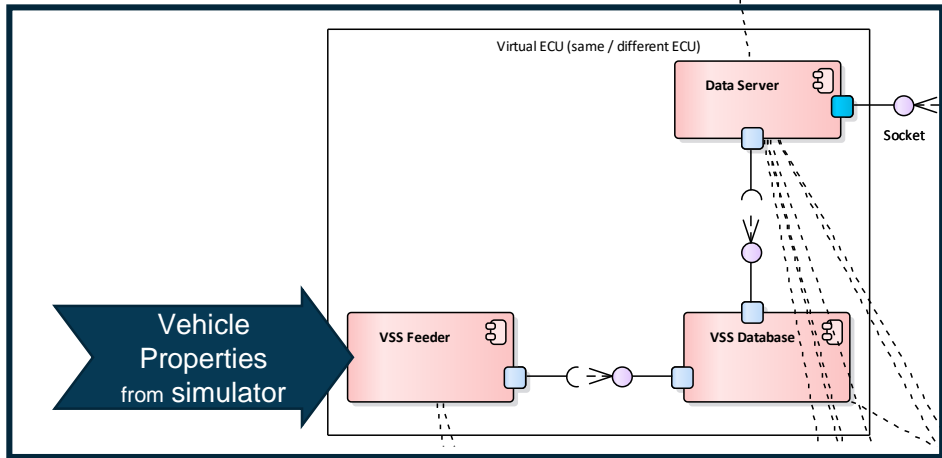
# Overview

**Defined by**

- Google
- Genivi
- OEM / Tier1

In order to decode the token being sent by the application, app can first verify if JWT was issued by a valid agent and then proceed to decode it.

TPM - trusted platform module



# OpenDS

## OpenDS

- <https://opends.dfki.de/>
- Configurable by set of XML
- Written in java (with native parts, sometimes closed-source), ~~however did not tried to compile it locally~~
- Officially 5.0 sourcecode distributed with executables in release bundle
- Unofficially multiple forks hosted on Github under the name openDS-oss (4.5)
- Based on opensource and free libraries

# OpenDS

- Able to provide data through tcp/ip sockets:
  - Speed (km/h)
  - Engine state (on/off)
  - Engine rpm
  - Actual gear
  - Steering wheel angle
  - Gas/brake pedal state
  - Fuel consumption (l/100km)
  - Fuel - max amount
  - Fuel - current amount
  - Headlights (high/low/off)
  - GPS position (lon, lat, alt, orientation)

## OpenDS

- To enable this data to be accessible configuration xml needs to be modified for:

```
<settingsControllerServer>  
  <startServer>true</startServer>  
  <port>5678</port>  
</settingsControllerServer>
```

# OpenDS

## Workflow:

- Connect to socket
- Send xml message to establish connection
- Server starts sending the properties



# OpenDS

Example frame

```
<Message>  
  <Event Name="SubscribedValues">  
    <root>  
      <thisVehicle>  
        <exterior>  
          <fueling> ...  
        </fueling>  
        <engineCompartment>  
          <engine>  
            <Properties>  
              <running>1</running>  
              <actualRpm>3178</actualRpm>  
            </Properties>  
          </engine>  
        </engineCompartment>  
        <lights> ...  
      </lights>  
      <gearUnit> ...  
    </gearUnit>  
  </exterior>  
  <physicalAttributes> ...  
</physicalAttributes>  
  <interior> ...  
</interior>  
</thisVehicle>  
</root>  
</Event>  
</Message>
```

# DEMO

# Genivi Vehicle Simulator

## Genivi Vehicle Simulator

- <https://github.com/GENIVI/genivi-vehicle-simulator>
- Use Unity as an engine
- Written in C#, developed in Unity IDE
- Sends values as a map, every frame includes property name bound with value and timestamp

# Genivi Vehicle Simulator

Example frame

```
EMSSetSpeed, 0.0000, 33.99587  
EngineSpeed, 3740.9430, 33.99587  
GearPosActual, 4.00, 33.99587  
GearPosTarget, 4.00, 33.99587  
AcceleratorPedalPos, 0.0000, 33.99587  
DeceleratorPedalPos, 0.0000, 33.99587  
RollRate, 0.0000, 33.99587  
SteeringWheelAngle, 0.0000, 33.99587  
VehicleSpeed, 95.1472, 33.99587  
VehicleSpeedOverGnd, 95.1472, 33.99587  
WheelSpeedFrL, 44872.8800, 33.99587  
WheelSpeedFrR, 45907.8800, 33.99587  
WheelSpeedReL, 45003.8100, 33.99587  
WheelSpeedReR, 45866.3700, 33.99587  
YawRate, 0.0000, 33.99587
```

# Genivi Vehicle Simulator

## Properties comparison

### GVS Exclusive

Target gear

Wheel speed

### Common

Speed (km/h)

Engine rpm

Actual gear

Steering wheel angle

Gas/brake pedal state

### OpenDS Exclusive

Engine state (on/off)

Fuel consumption  
(l/100km)

Fuel - max amount

Fuel - current amount

Headlights (high/low/off)

GPS position (lon, lat,  
alt, orientation)

## Reference and contacts

Weekly telcos of AASIG Vehicle Api subgroup

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

AASIG Mailing list

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

Reference VSS feeder implementation

- <https://github.com/genivi/vss-feeder>

Whole demo setup including GraphQL schema from VSS + GrapQL server  
(work in progress, README.md to be updated for docker-compose)

- <https://github.com/stefanwysocki/vss-graphql/tree/vss-server-main-repo>

**tieto** *EVRY*

Stefan Wysocki  
[stefan.wysocki@tietoevry.com](mailto:stefan.wysocki@tietoevry.com)