VSSo: a car signal ontology

April 18, 2018 | Extension of the Vehicle Signal Specification

Benjamin Klotz

PhD student, BMW Group, EURECOM

Initial context

{"name":"accelerator_pedal_position","value":0,"timestamp":1361454211.483000}
{"name":"fuel_level","value":23.478279,"timestamp":1361454211.485000}
{"name":"torque_at_transmission","value":1,"timestamp":1361454211.488000}

- Adapt infotainment from online use favorite playlists and volume.
- Drive your car in front of the building you are leaving
- …

Temperature sensor

Adaptive cruise control

Front camera

Radar

Blind spot detection

Wheel speed sensor

Oil temperature sensor

Steering angle sensor

Park assistant

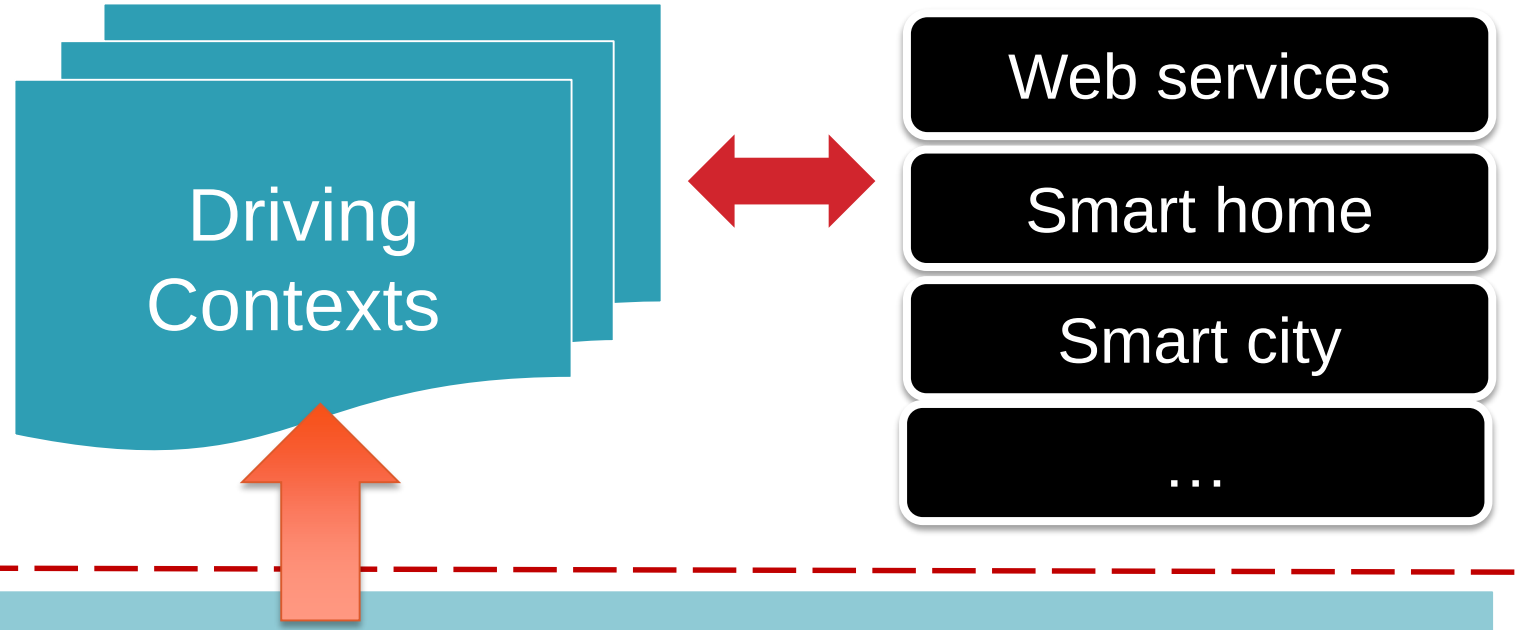
Tire pressure sensor

Vehicle height sensor

GENIVI®

Driving context

Goals:
- Personalization
- Awareness
- Data fusion



GENIVI®

Enabling interoperability through the signal data model?

VSS in a nutshell

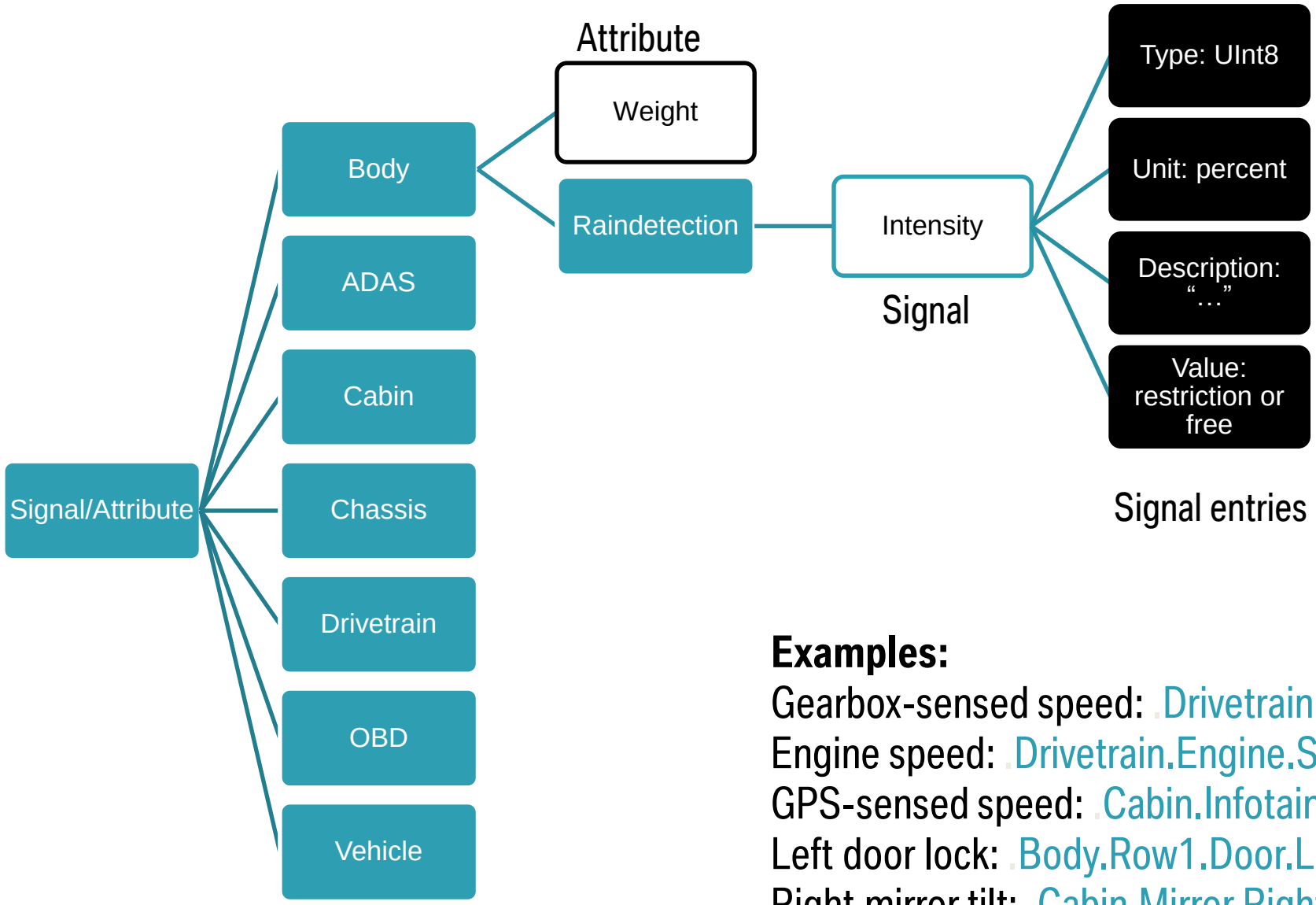


Figure:

- 451 branches
- 1103 leaves:
  - 43 attributes
  - 1060 signals: including
    - (700 seat-related),
    - 268 with unit

Examples:

Gearbox-sensed speed: Drivetrain.Transmission.Speed
Engine speed: Drivetrain.Engine.Speed
GPS-sensed speed: Cabin.Infotainment.Speed
Left door lock: Body.Row1.Door.Left.IsLocked
Right mirror tilt: Cabin.Mirror.Right.Tilt

GENIVI®

Ontologies for (car) signals [1]

- A W3C and OGC recommendation (19 October 2017)

The **Semantic Sensor Network (SSN)** ontology is an ontology for describing

- **sensors**
- and their **observations**,
- the involved **procedures**,
- the studied **features of interest**,
- the **samples** used to do so,
- and the **observed properties**,
- as well as **actuators, actuations...**

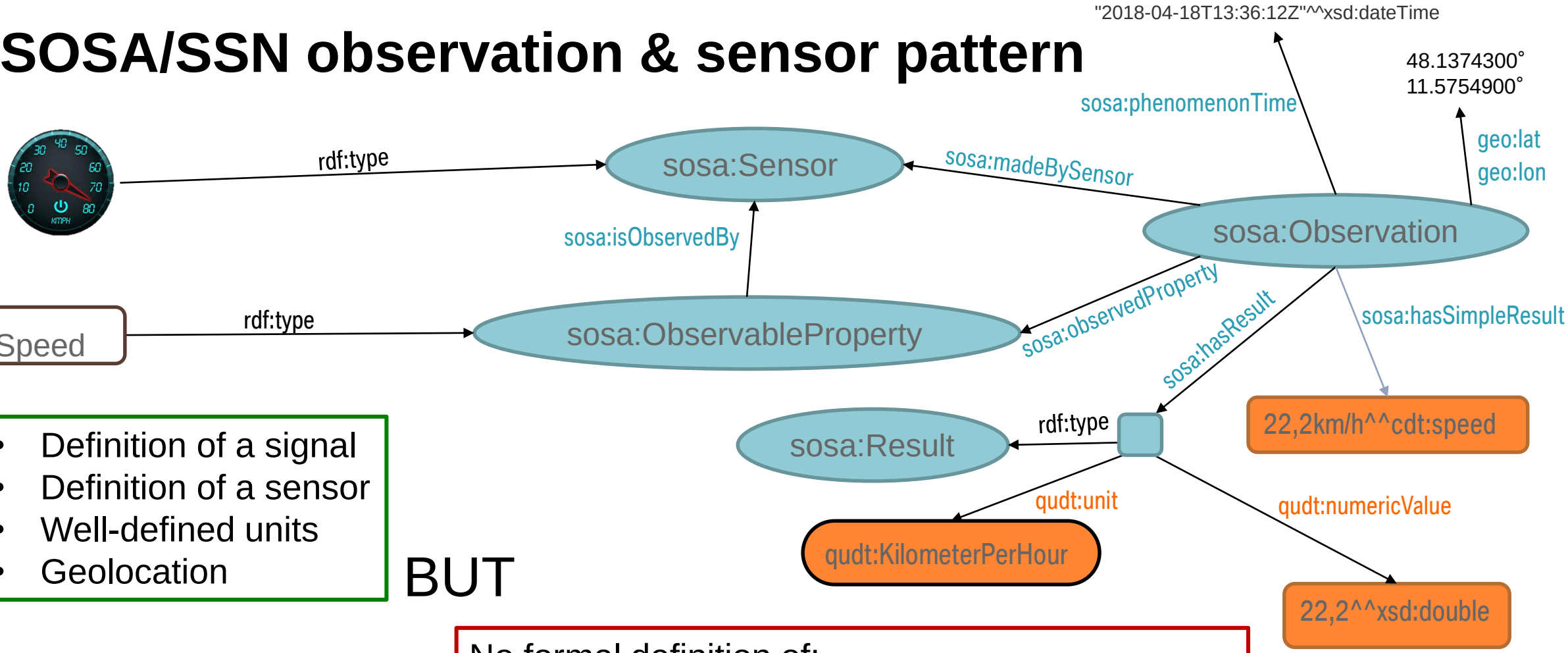
SSN follows a horizontal and vertical modularization architecture by including a lightweight but self-contained core ontology called **SOSA (Sensor, Observation, Sample, and Actuator)** for its elementary classes and properties.

- Supports a wide range of applications and use cases, including satellite imagery, large-scale scientific monitoring, industrial and household infrastructures, social sensing, citizen science, observation-driven ontology engineering, and the Web of Things.

[1] https://www.w3.org/TR/vocab-ssn/

GENIVI®

SOSA/SSN observation & sensor pattern



- Definition of a signal
- Definition of a sensor
- Well-defined units
- Geolocation

BUT

No formal definition of:

- “speed” or other observable properties
- “speedometer” or other car sensors/actuators
- “Car” or vehicle parts

SSN/SOSA with a VSS ontology

rdf:type

vss:Speedometer

rdfs:subClassOf

sosa:Sensor

sosa:madeBySensor

sosa:Observation

sosa:isObservedBy

sosa:Observable Property

sosa:observedProperty

sosa:hasResult

sosa:hasSimple Result

22,2km/h^^cdt:UCUM

rdfs:subClassOf

vss:ObservableSignal

sosa:Result

rdf:type

qudt:unit

qudt:numericValue

qudt:Unit

22,2^^xsd:double

rdfs:subClassOf

vss:Speed

vss:Drivetrain

vss:hasSignal

vss:partOf

vss:partOf

:Speed

rdf:type

vss:Transmission

vss:Internal Combustion

rdfs:subClassOf

rdfs:subClassOf

rdfs:subClassOf

vss:Branch

vss:maxTorque

4Nm^^cdt:torque

What is required ?

From VSS spec to the VSS ontology

Map to existing Ontologies
- SSN/SOSA
- QUDT (unit)

Generate definition of VSS concepts

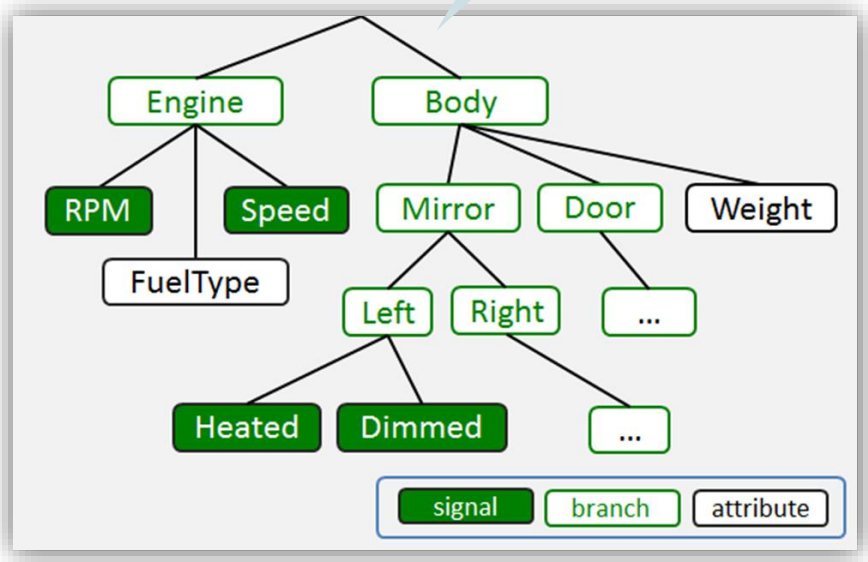
Apply **consistency policies**

Manually validate and clean the generated ontology

VSSontology

Add sensors and actuators

VSS



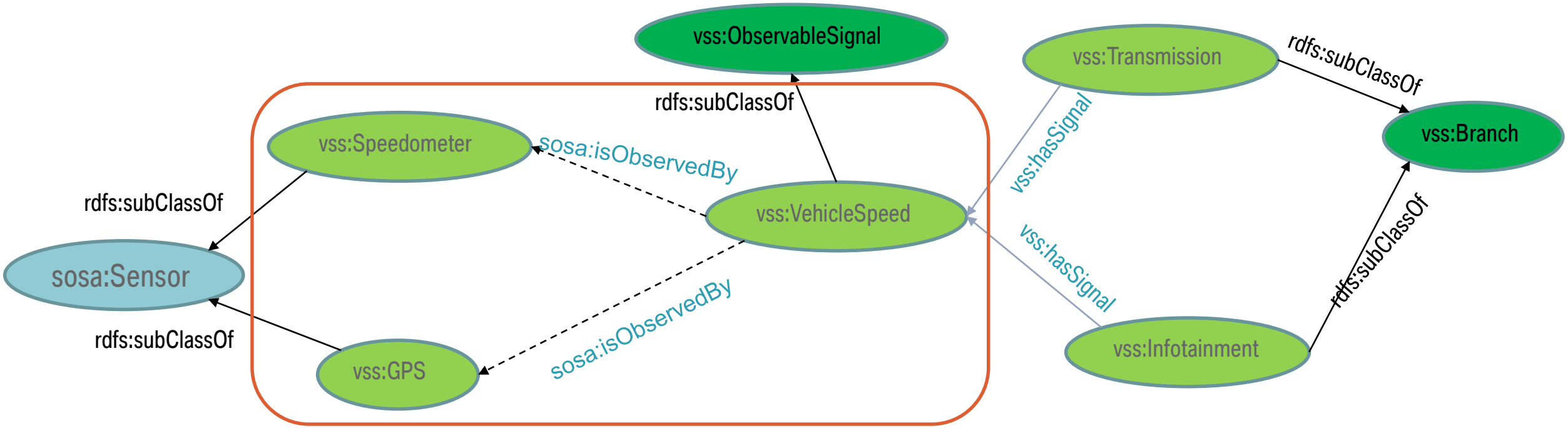
Modeling policies

1. VSS concepts have unique names
2. All signals are either observable, actuable or both
3. All signals have either a sensor or an actuator
4. All branches are part of the top “vss:Vehicle” branch
5. All position-dependent branches have a property “position”

GENIVI®

Modeling choice 1: unique names for unique concepts

- Some signals represent the same phenomenon, sensed by different sensors
  - Ex: Drivetrain.Speed (sensed by the **gearbox**) and Infotainment.Speed (sensed by the **GPS**)



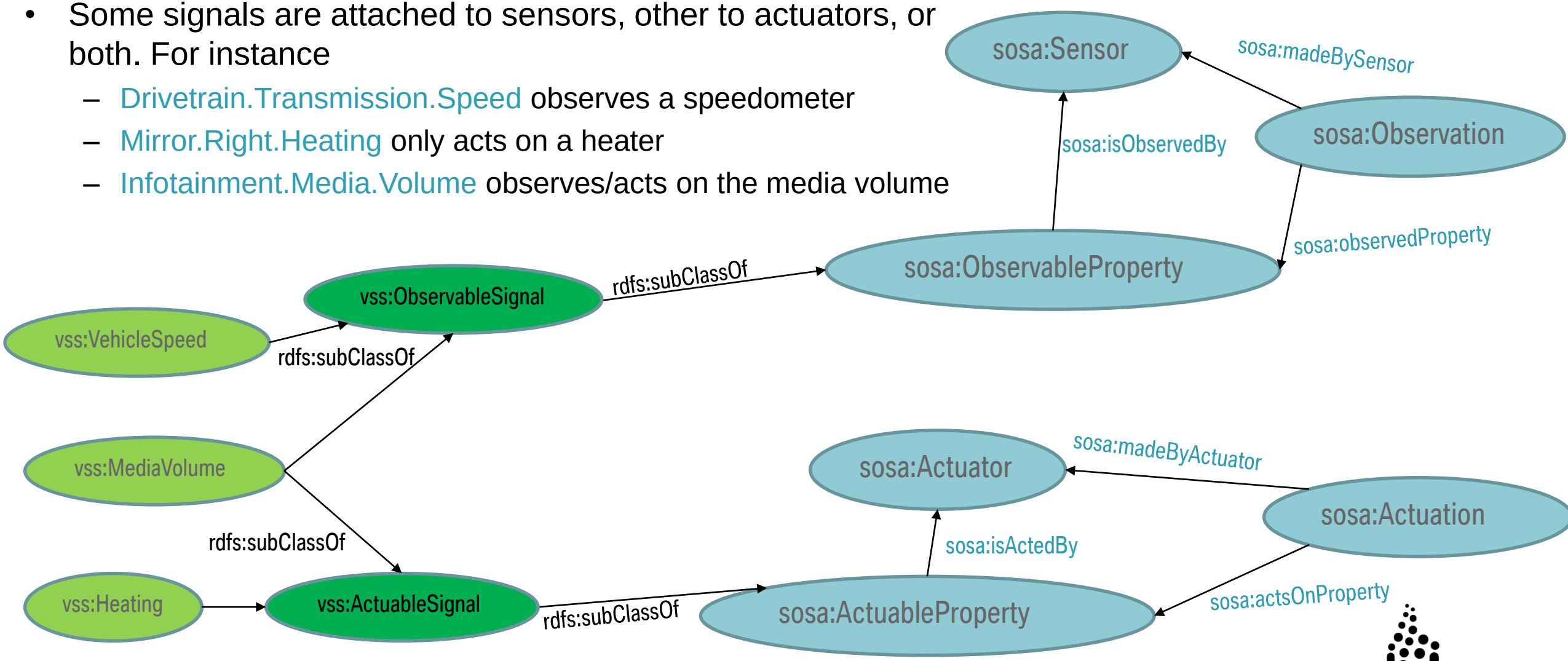
"vss:VehicleSpeed" is a unique phenomenon
↳ observed by different sensors
↳ Producing different signals

Names are clarified to avoid homonymy

GENIVI®

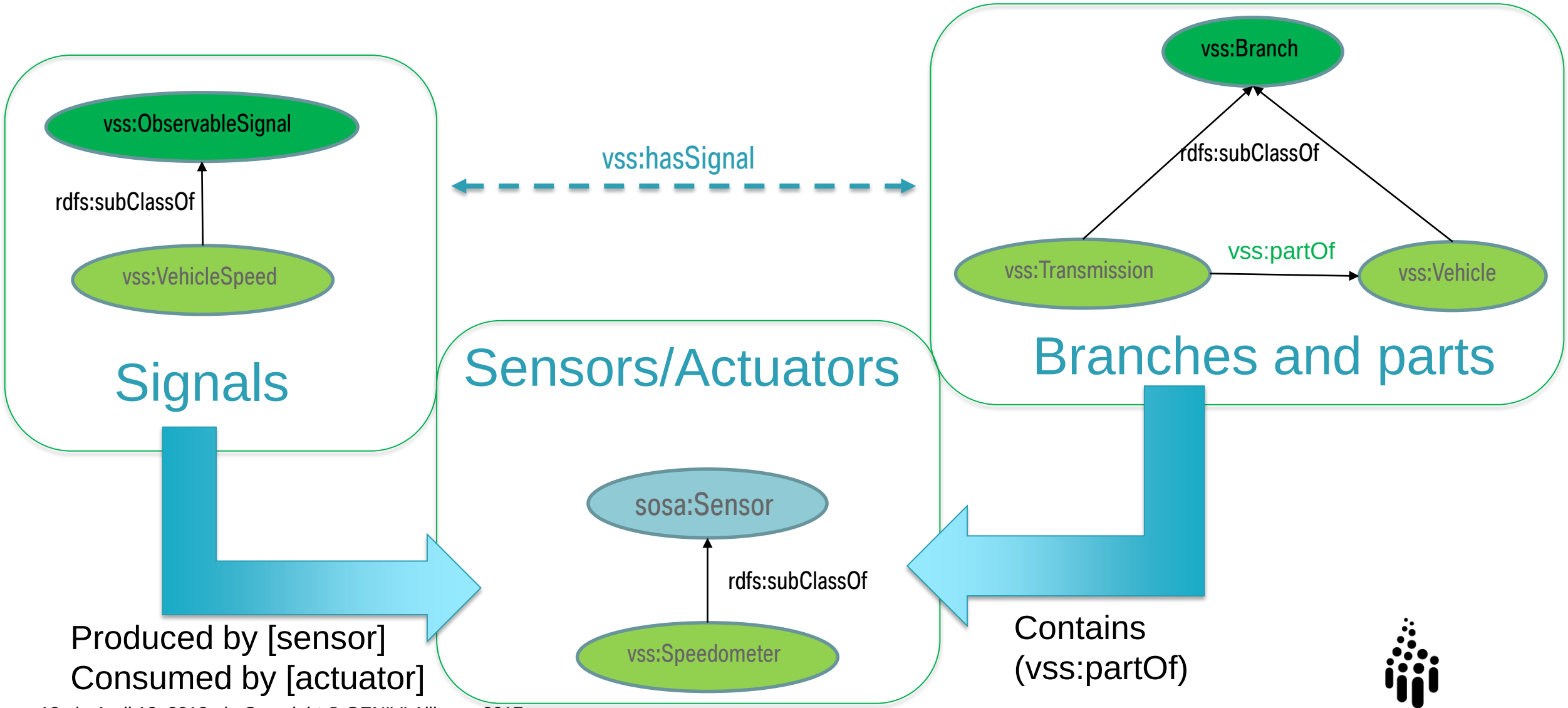
Modeling choice 2: signals are observable, actuable or both

- Some signals are attached to sensors, other to actuators, or both. For instance
  - Drivetrain.Transmission.Speed observes a speedometer
  - Mirror.Right.Heating only acts on a heater
  - Infotainment.Media.Volume observes/acts on the media volume

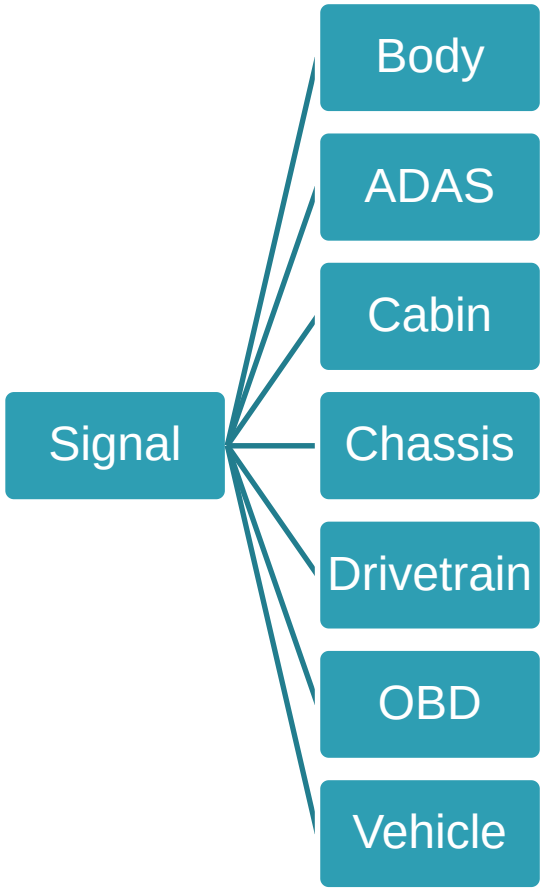


GENIVI®

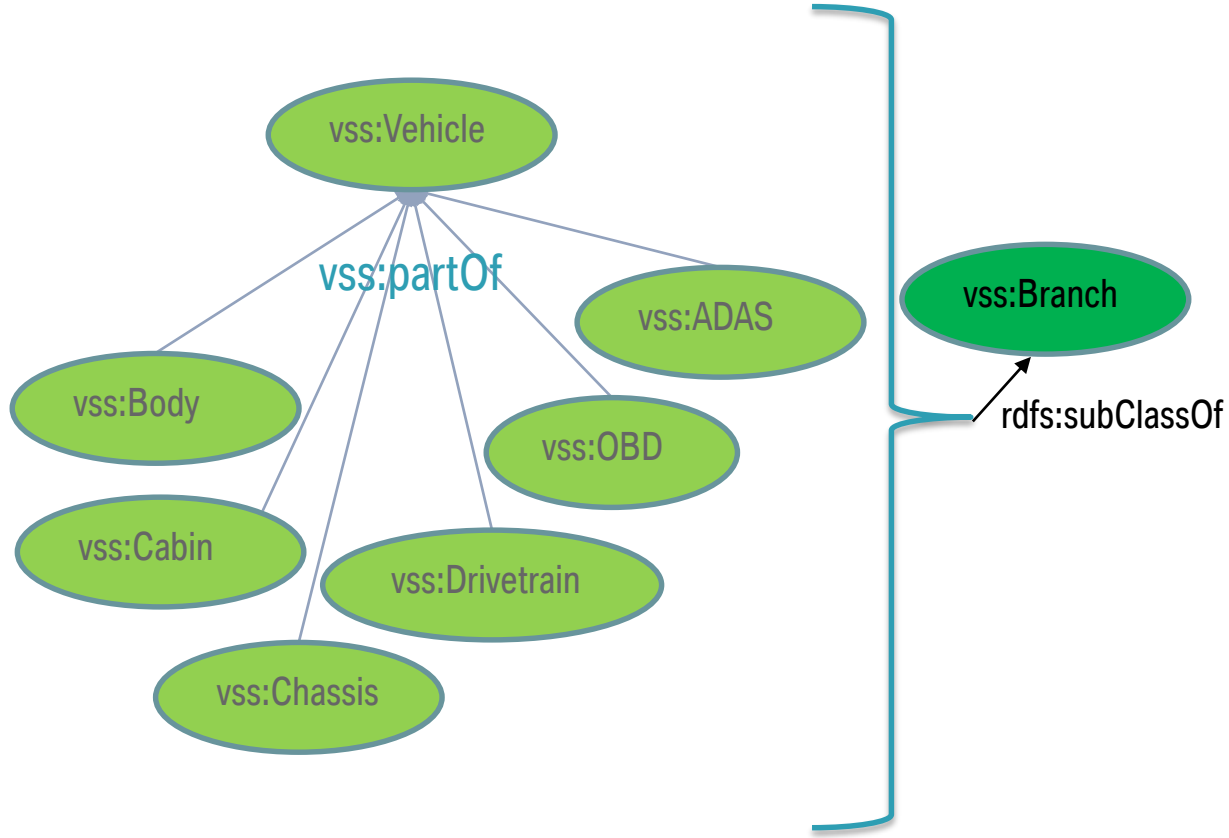
Modeling choice 3: all signals have a sensor or actuator



Modeling choice 4: all branches are vss:partOf vss:Vehicle



Current VSS structure

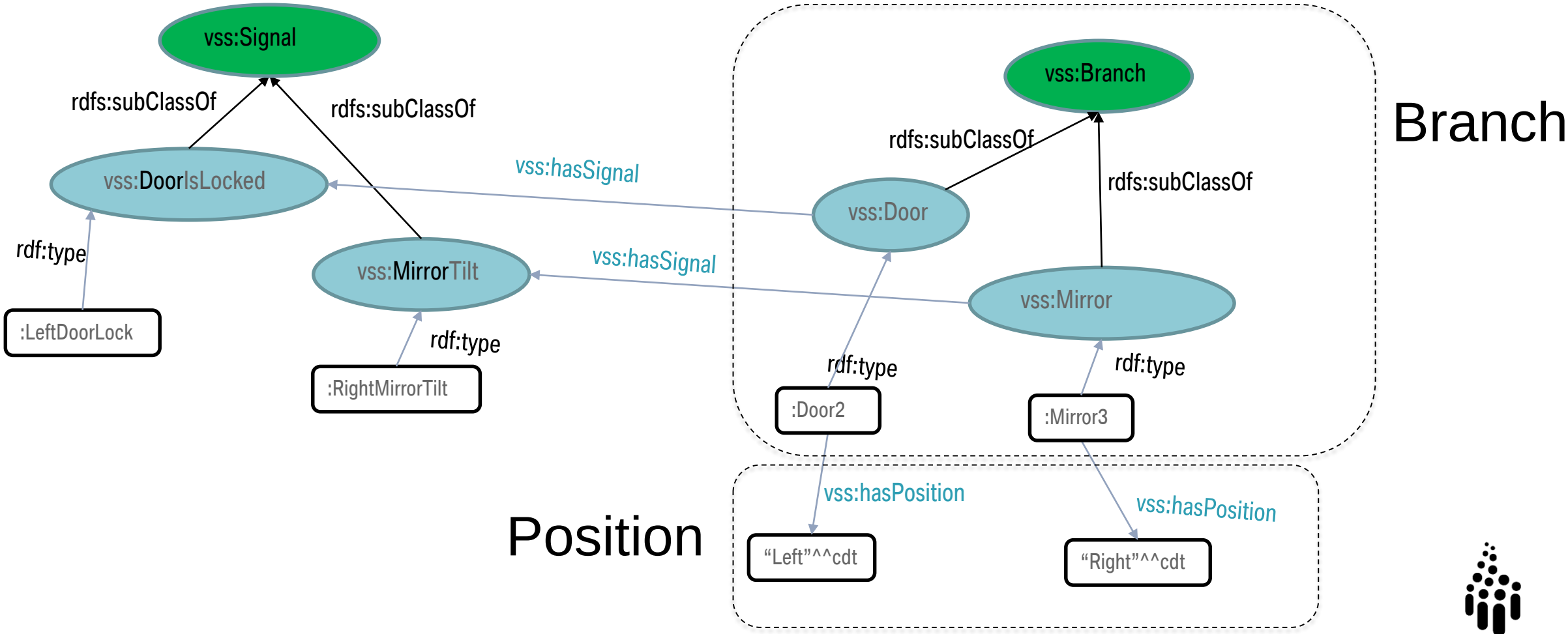


VSSontology structure

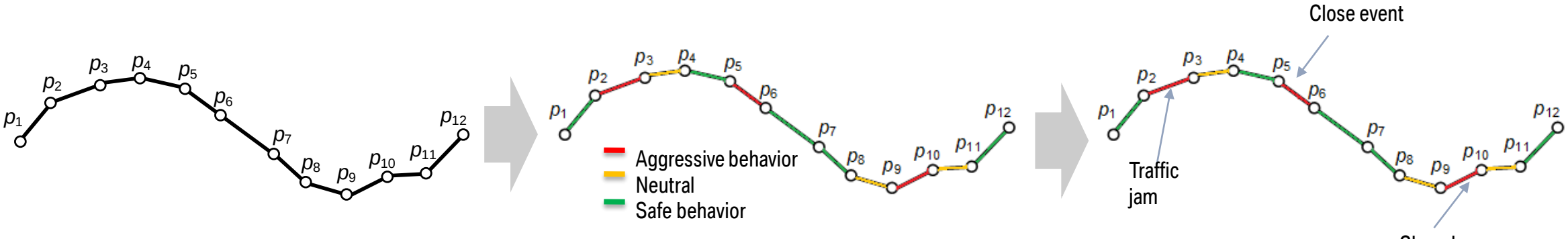
GENIVI®

Modeling choice 5: position ≠ branch

- Ex: Door.Left.IsLocked, Mirror.Right.Tilt
- Branches vss:Door and vss:Mirror have **vss:hasPosition** property with limited potential values (“Left”,”Right”,”Row1”, …)

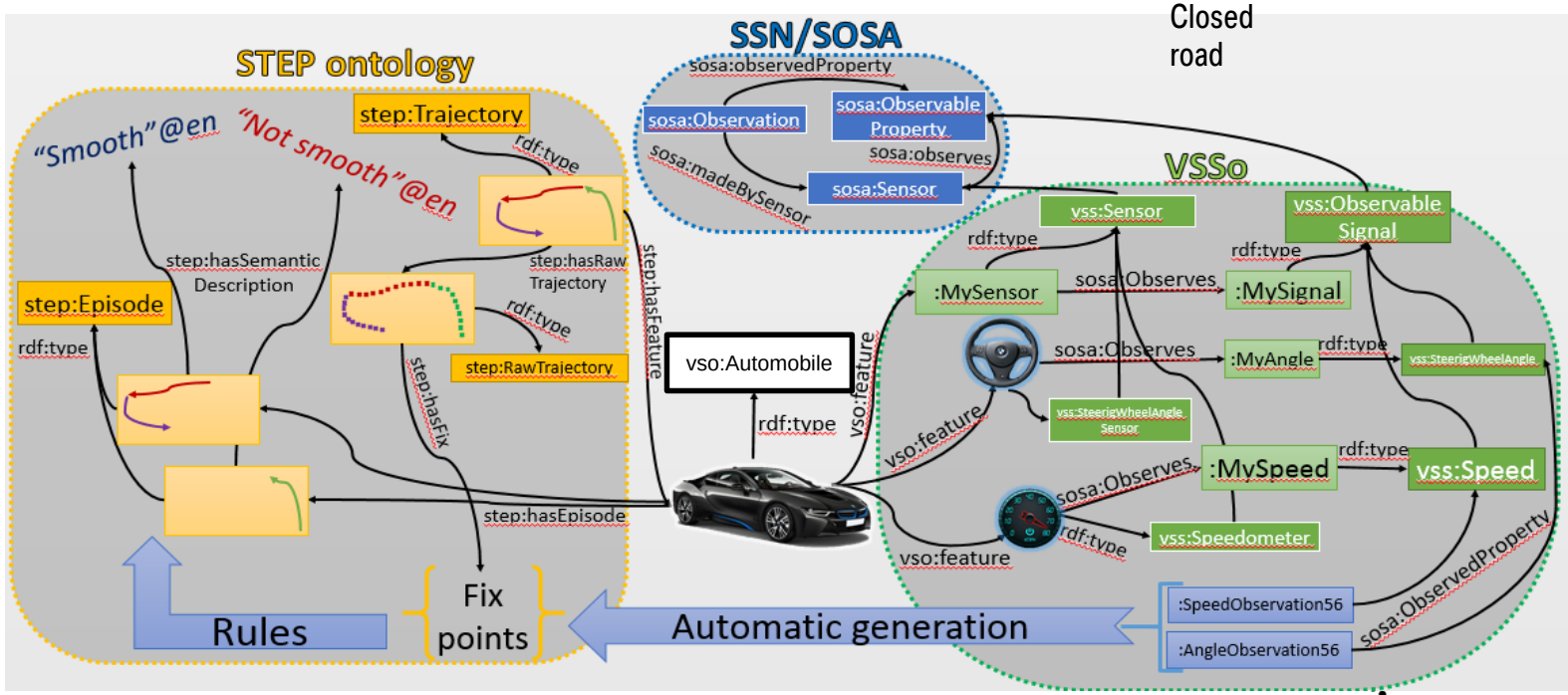


Application: linking signals to a heterogeneous context



Driving Contexts

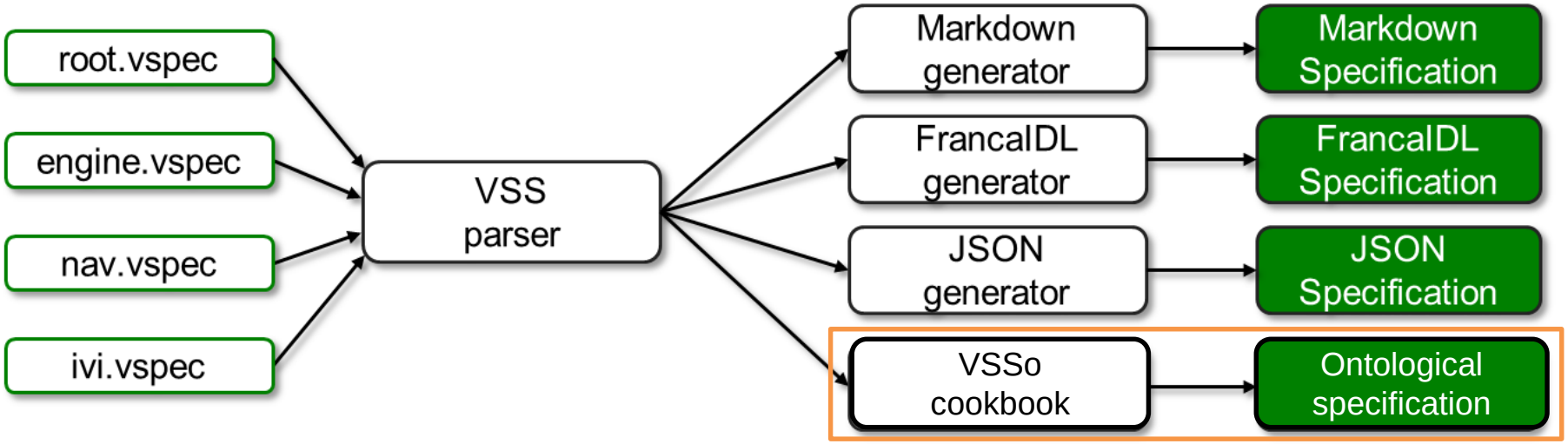
Web services
Smart home
Smart city



GENIVI®

What about OEM-specific concepts ?

Ontological specification

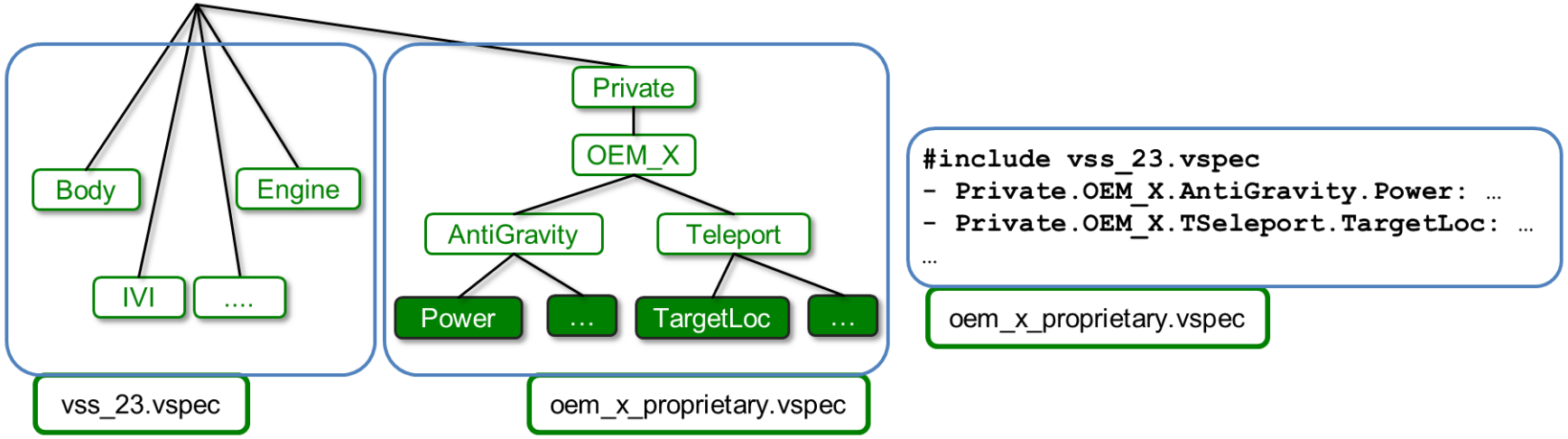


Base VSSo

- 84 formally-defined vehicle parts
- ~300 formally-defined signals
- ~40 formally-defined attributes

GENIVI®

VSS private branch



Private branch

- OEM-specific concepts
- Extension of VSS
- Merged into VSS when generating specifications

Private OEM-specific ontology cookbook:

1. Write VSS-compliant specification of private concepts (new signals, attributes and branches)
   - Follow the VSS policies just as when creating a private branch
2. Generate the ontology using the existing tool
3. Validate the ontology
   - Check the unicity of concepts and definitions (in the private branch and if possible with VSSo)
4. Define a private namespace for your ontology integrating VSSo

GENIVI®

Thank you for your attention

Do you have questions ?

Branch examples

```
vss:Branch a rdfs:Class, owl:Class;
  rdfs:label "Branch"@en;
  rdfs:comment "Branch of the vehicle. Either a component (Body, Chassis...) or the complete vehicle"@en.

vss:ObstacleDetection a rdfs:Class, owl:Class;
  rdfs:subClassOf vss:Branch;
  rdfs:label "ObstacleDetection"@en;
  rdfs:comment "Signal/Attribute.ADAS.ObstacleDetection : Signals form Obstacle Sensor System"@en;
  rdfs:subClassOf [
    a owl:Restriction;
    owl:onProperty vss:partOf;
    owl:allValuesFrom vss:ADAS
  ];
  rdfs:subClassOf [
    a owl:Restriction;
    owl:onProperty vss:hasSignal;
    owl:allValuesFrom [owl:unionOf vss:ObstacleDetectionIsActive, vss:ObstacleDetectionError]
  ].
```


GENIVI®

Signal examples

```
vss:ObservableSignal a rdfs:Class, owl:Class;
  rdfs:subClassOf sosa:ObservableProperty;
  rdfs:label "Observable signal"@en;
  rdfs:comment "All observable signals that can dynamically be updated by the vehicle"@en.

vss:AmbientAirTemperature a rdfs:Class, owl:Class;
        rdfs:subClassOf vss:ObservableSignal;
        rdfs:label "AmbientAirTemperature"@en;
        rdfs:comment "Signal.Vehicle.AmbientAirTemperature : Ambient air temperature"@en;
        rdfs:subClassOf [
                a owl:Restriction;
                owl:onProperty sosa:isObservedBy;
                owl:allValuesFrom vss:Thermometer
        ];
        rdfs:subClassOf [
                a owl:Restriction;
                owl:onProperty qudt:unit;
                owl:allValuesFrom vocab:DegreeCelcius
        ].
```


GENIVI®

Signal examples

```
vss:attribute a owl:ObjectProperty;
  rdfs:label "Attribute"@en;
  rdfs:comment "Attribute signals that do not change during the power cycle of a vehicle."@en;
  rdfs:domain vss:Branch.

vss:driveType a owl:DatatypeProperty;
        rdfs:subPropertyOf vss:attribute;
        rdfs:label "DriveType"@en;
        rdfs:comment "Attribute.Drivetrain.Transmission.DriveType : Drive type."@en;
        rdfs:domain vss:Transmission;
        rdfs:range [owl:oneOf("unknown"@en "Front-wheel drive"@en "Rear-wheel drive"@en
"All-wheel drive"@en)].
```


GENIVI®

Thank you!

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

Contact us: help@genivi.org