SEMANTIC TECHNOLOGIES FOR CONNECTED VEHICLES IN A WEB OF THINGS ENVIRONMENT

Raphaël Troncy

✉raphael.troncy@eurecom.fr

BMW GROUP THE NEXT 100 YEARS

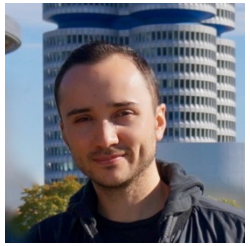
EURECOM Sophia Antipolis

Rolls-Royce Motor Cars Limited

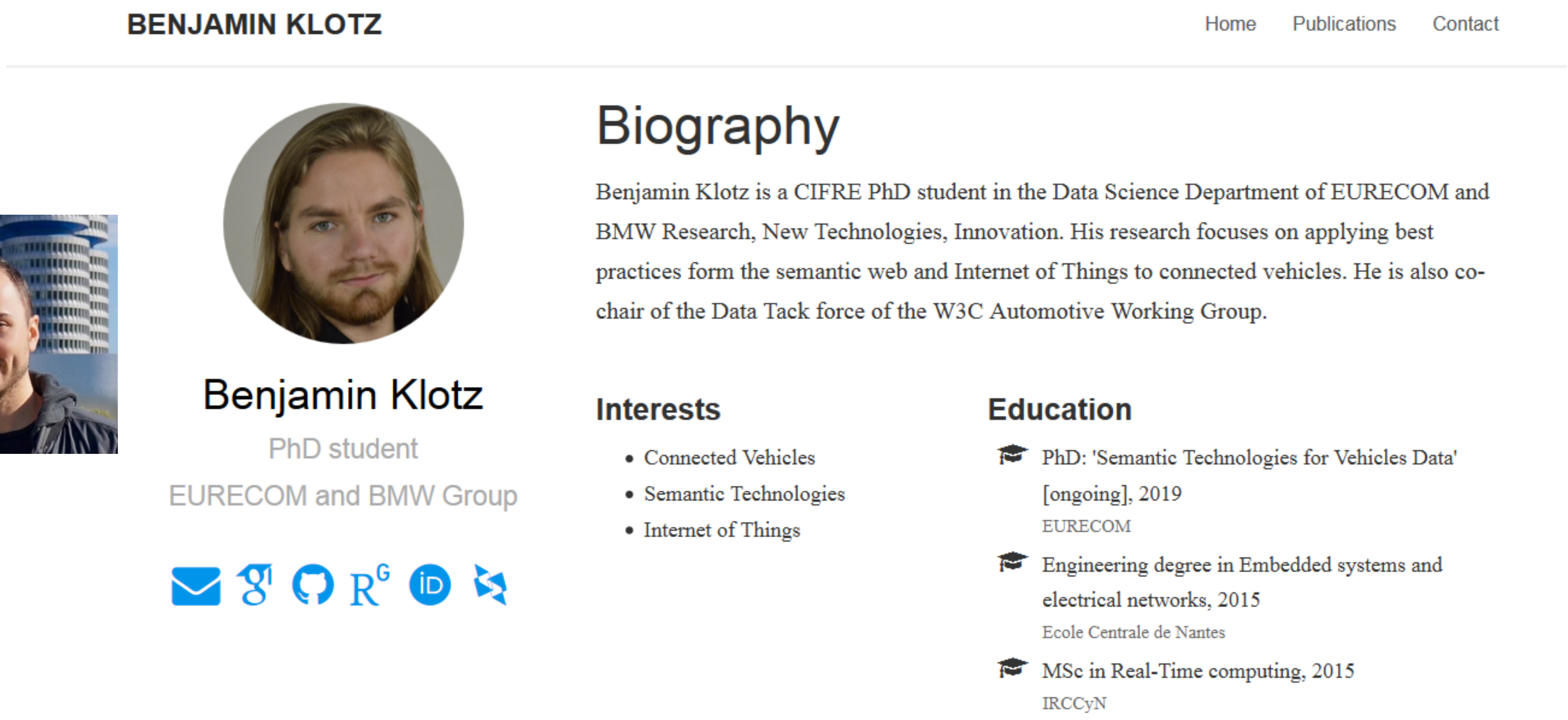
CREDITS

– Dr. Benjamin Klotz (EURECOM / BMW)

– Daniel Alvarez Coello
(PhD Candidate, Uni. Oldenburg)
https://www.linkedin.com/in/jdacoello/

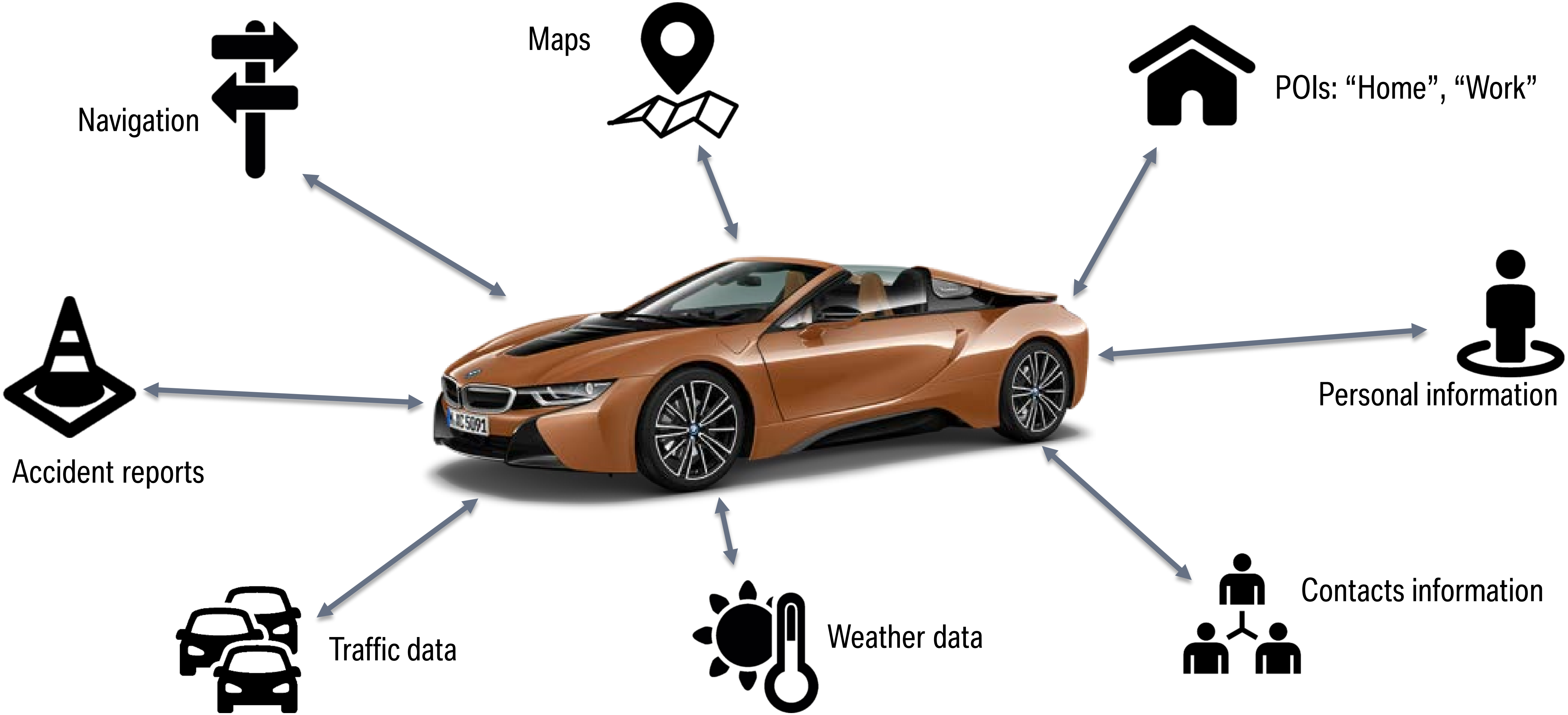


– Dr. Daniel Wilms (BMW researcher)
https://www.linkedin.com/in/danielwilms/



http://www.eurecom.fr/~klotz/

DATA AROUND THE AUTOMOTIVE DOMAIN



SENSOR DATA IN THE AUTOMOTIVE DOMAIN

{"acceleratorPedal":{"position":"4095","ecoPosition":"3"},"brakeContact":"16","speedActual":"0“}, "timeStamp":"2018-01-10T17:01:27.297Z",}

{"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}

Signal name?
Units?
Datetime?

Temperature sensor

Adaptive cruise control

Front camera

Radar

Blind spot detection

Wheel speed sensor

Oil temperature sensor

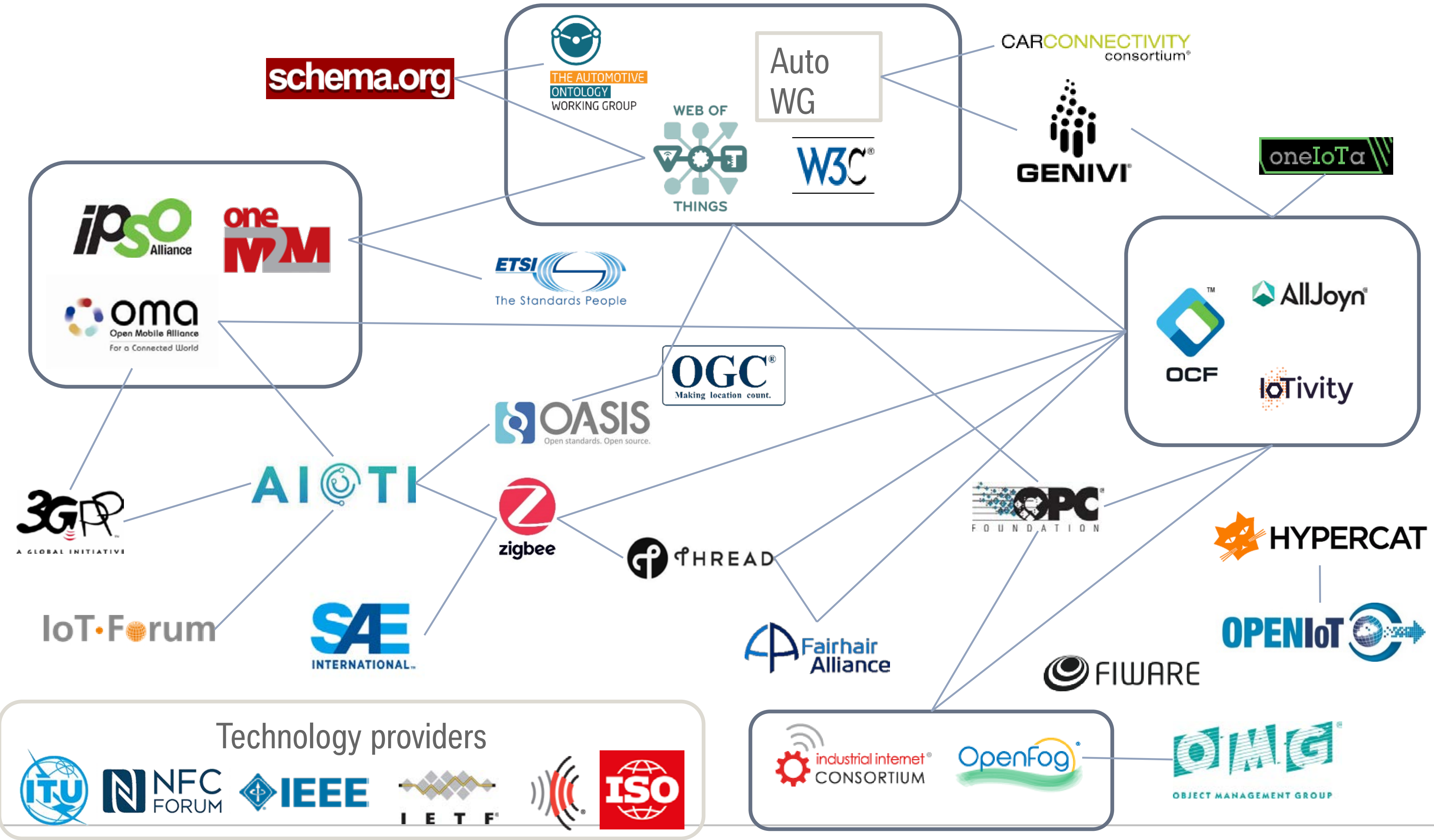
Tire pressure sensor

Steering angle sensor

Park assistant

Vehicle height sensor

INTEROPERABILITY IN A FRAGMENTED IOT ECOSYSTEM



HOW CAN PROVIDE INTEROPERABLE DESCRIPTION OF VEHICLE DATA? REQUIREMENTS: COMPETENCY QUESTIONS

Get information about attributes and signals on connected vehicles

32 competency questions…

Attributes

What type of fuel does this car need?
What is the model of this car?
How old is this car?
What type of transmission does this car have?

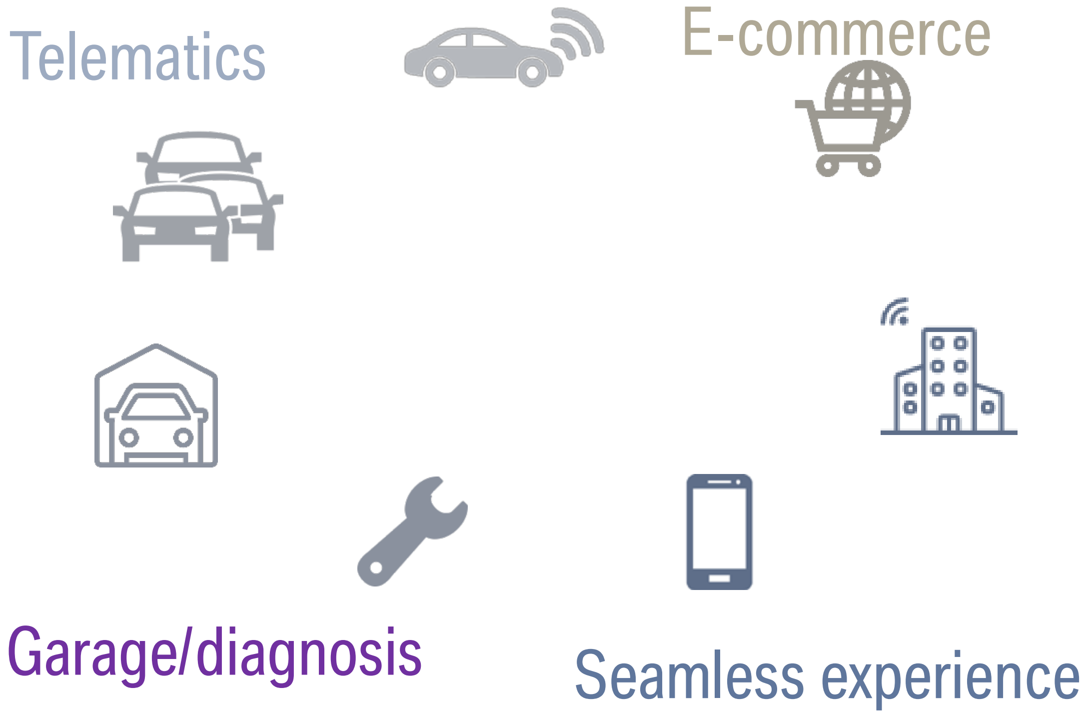
Signals and sensors

Is there a signal measuring the steering wheel angle?
How many different speedometers does this car contain?

Dynamic signals

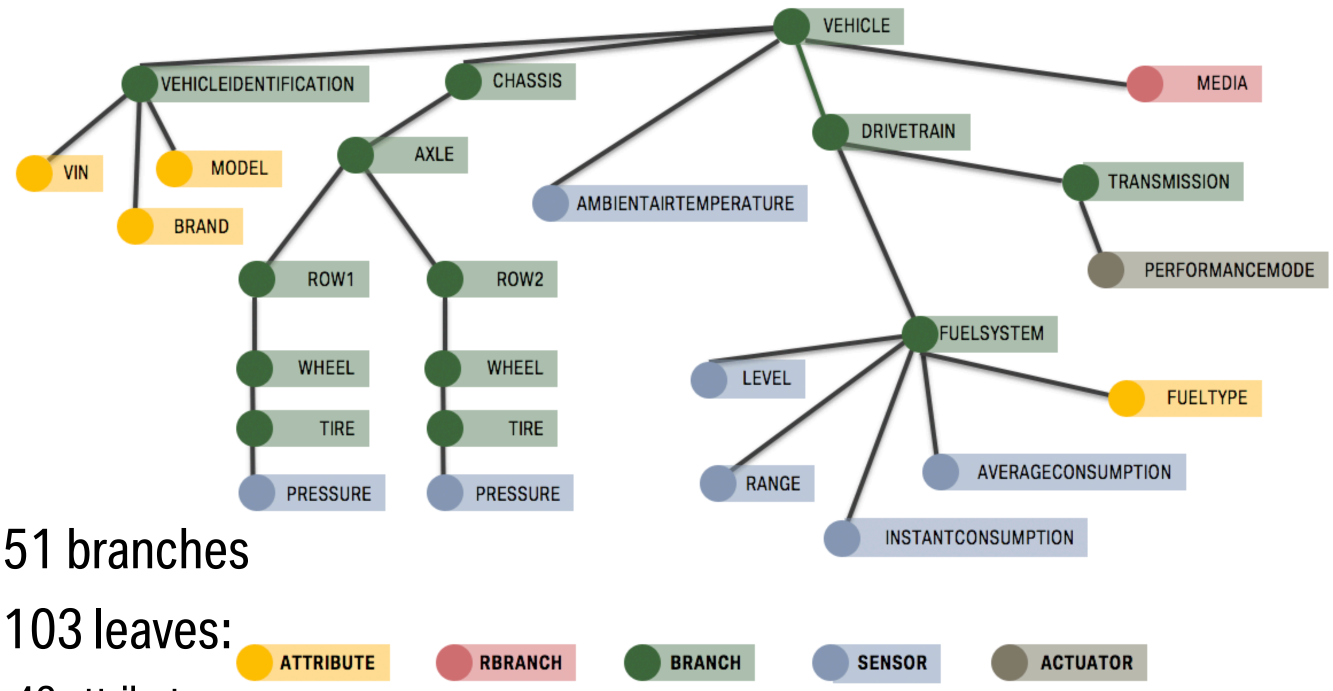
What is the current gear?
What is the local temperature on the driver side?

… generated from domain needs on vehicle signals and attributes



Complete list: https://github.com/klotzbenjamin/vss-ontology

VSS IN A NUTSHELL



– 451 branches

– 1103 leaves:

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

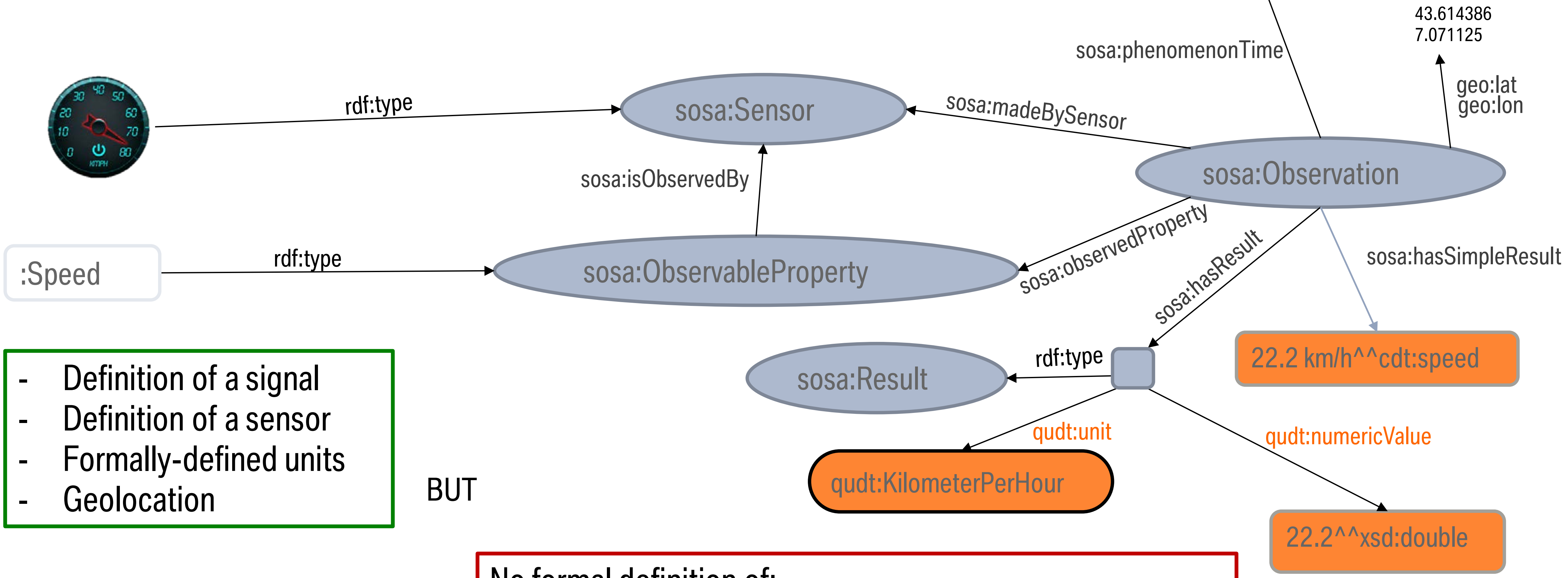
```
- Drivetrain.Transmission.Speed:
  type: sensor
  datatype: uint16
  unit: km/h
  min: 0
  max: 300
  description: The vehicle speed, as measured by the drivetrain.
```

VSS

- Data model: data structure for attributes, sensors and actuators of vehicles
- Specifies uniform structure for data description
  - Path / name
  - (Data-)Type
  - Unit
  - Range
  - Description
- Extensible and suitable for multi user collaboration

https://github.com/GENIVI/vehicle_signal_specification

HYPOTHESIS: SOSA PATTERN FOR VEHICLE SIGNALS



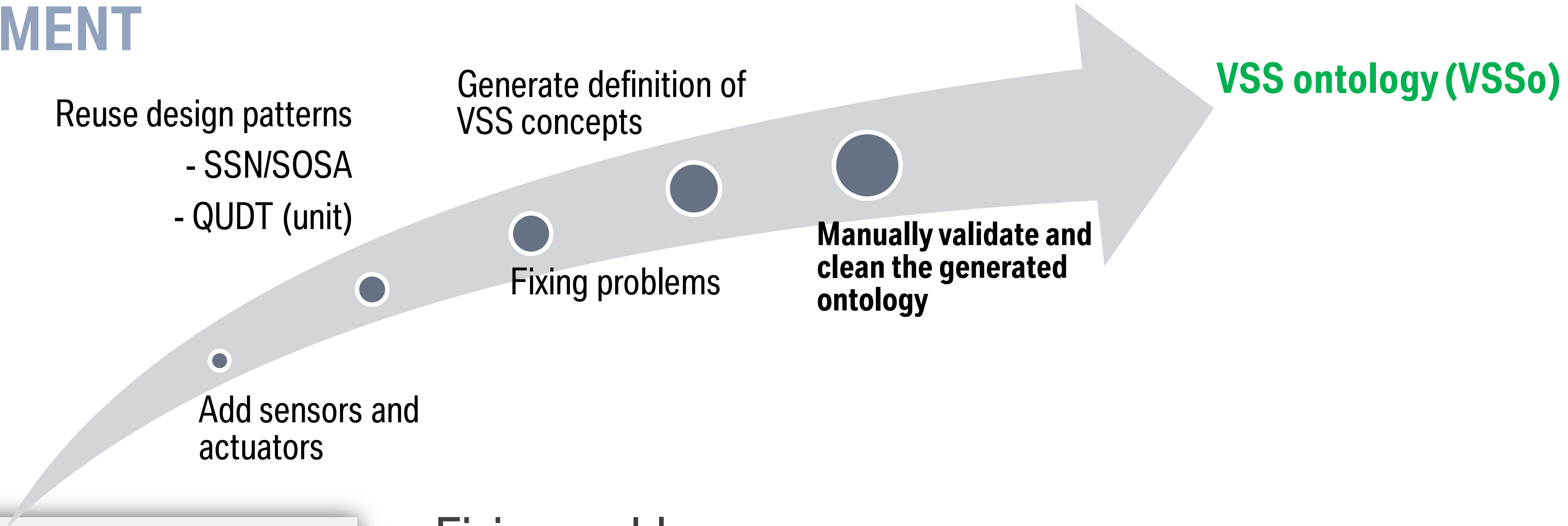
- Definition of a signal
- Definition of a sensor
- Formally-defined units
- Geolocation

BUT

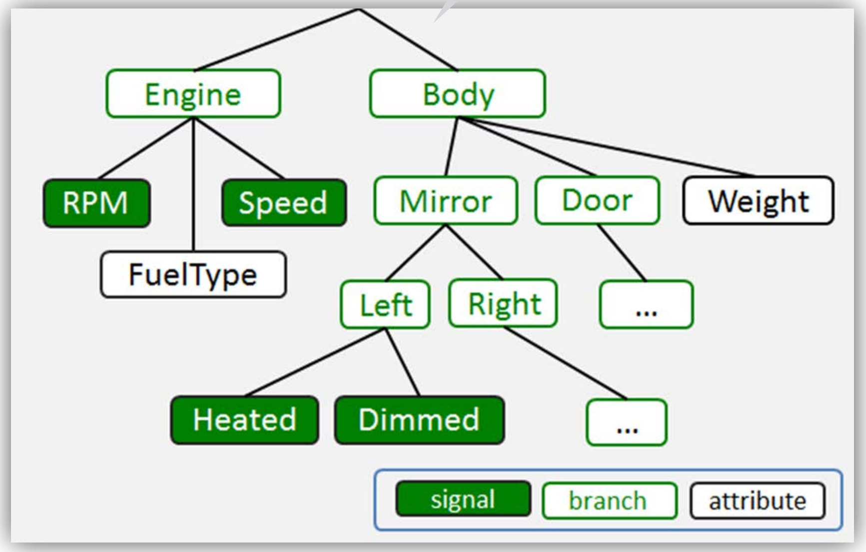
No formal definition of:

- "speed" or other observable properties
- "speedometer" or other car sensors/actuators
- "Vehicle" or vehicle parts

VSSo DEVELOPMENT



VSS

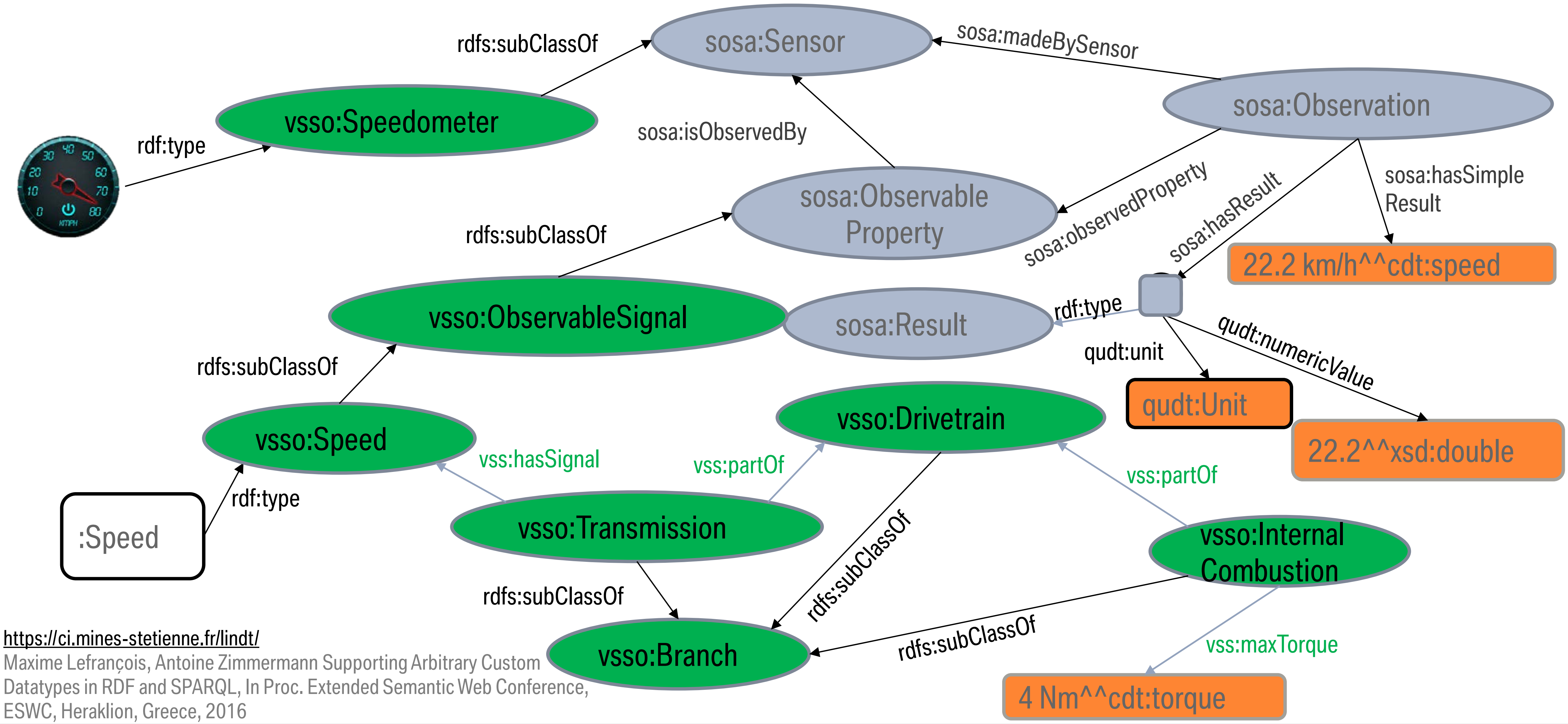


Fixing problems

1. VSS concepts have unique names
2. All signals are either observable, actuatable or both
3. Different signals can yield the same phenomenon (e.g. speed)
4. All branches are part of the top “vsso:Vehicle” branch
5. All position-dependent branches have a property “position”

Benjamin Klotz, Raphael Troncy, Daniel Wilms, and Christian Bonnet. VSSo: A Vehicle Signal and Attribute Ontology. In 9th International Semantic Sensor Networks Workshop (SSN), Monterey, California, October 2018.

VSSo EXAMPLE



https://ci.mines-stetienne.fr/lindt/

Maxime Lefrançois, Antoine Zimmermann Supporting Arbitrary Custom Datatypes in RDF and SPARQL, In Proc. Extended Semantic Web Conference, ESWC, Heraklion, Greece, 2016

VSSo SUMMARY

VSSo: a Vehicle Signal and Attribute ontology (http://automotive.eurecom.fr/vsso)

- OWL ontology of DL expressivity: ALUHOI+
- 483 classes (~300 signals); 63 properties (~50 attributes)
- Reuse SSN/SOSA modeling patterns

Evaluation:

Hypothesis: VSSo data enables SPARQL queries answering the set of competency questions

Dataset: simulated (random) values for 19 signals and 23 fixed attributes on a sliding window of 3 seconds

Experiment: set 2 SPARQL endpoints with VSSo data (with 1 vehicle, with a fleet of 3 vehicles)

http://automotive.eurecom.fr/simulator/query
http://automotive.eurecom.fr/simulator/fleetquery

VSSo expressivity: most requirements can be translated into SPARQL queries

What are the dimension of this car?

```
SELECT ?length ?width ?height
WHERE { ?branch vsso:length ?length;
            vsso:width ?width;
            vsso:height ?height.}
```

90% of competency questions can be answered

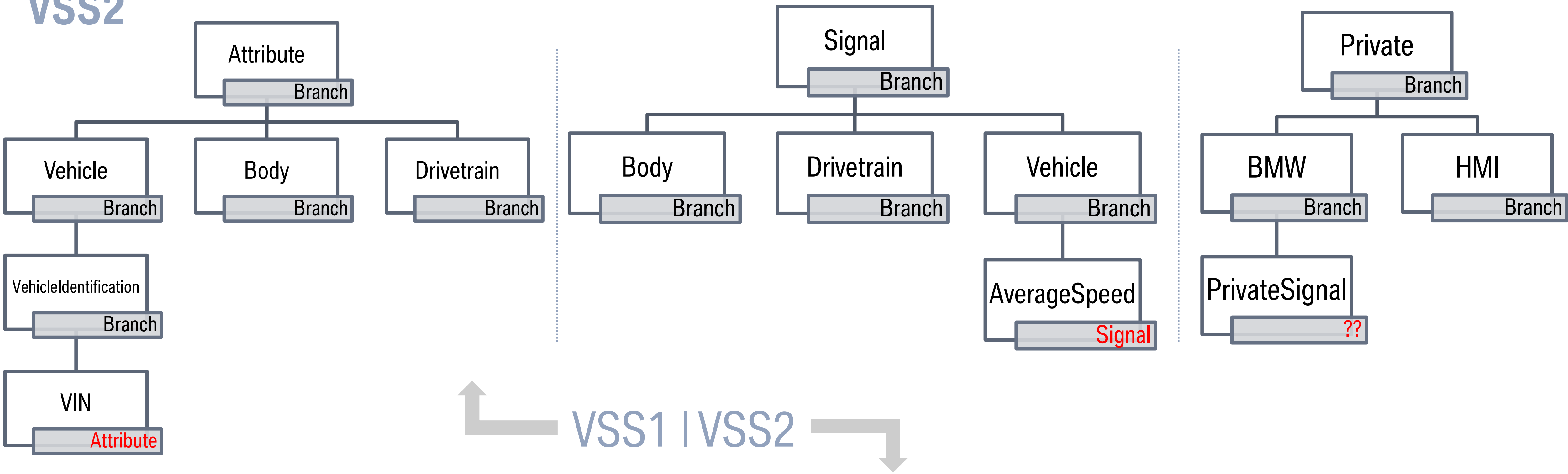
http://automotive.eurecom.fr/simulator/query
http://automotive.eurecom.fr/simulator/fleetquery

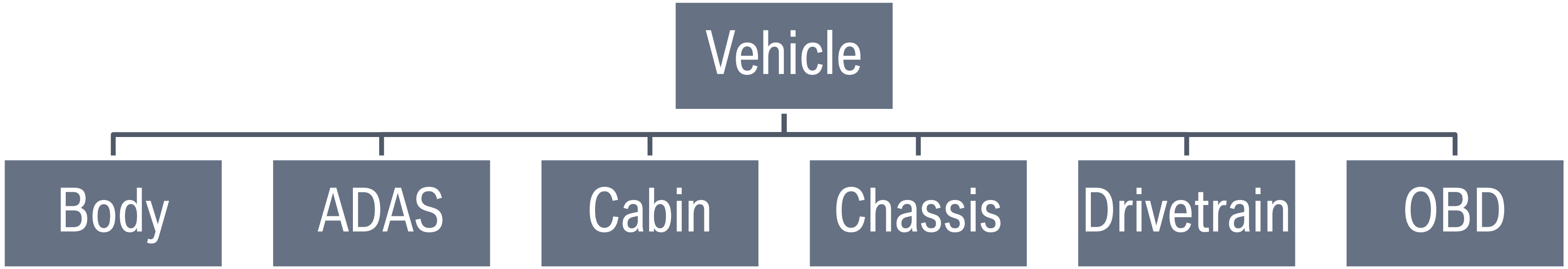
What is the current temperature on the driver side?

```
SELECT DISTINCT ?localTemperature ?value ?position ?time
WHERE { ?wheel a vsso:SteeringWheel;
vsso:steeringWheelSide ?steeringWheelSide.
?branch a vsso:LocalHVAC;
vsso:position ?position;
vsso:hasSignal ?localTemperature.
?localTemperature a vsso:LocalTemperature.
FILTER regex(str(?steeringWheelSide),str(?position))

?obs a sosa:Observation;
        sosa:observedProperty ?localTemperature;
        sosa:hasSimpleResult ?value;
        sosa:phenomenonTime ?time.
}
ORDER BY DESC(?time)
LIMIT 1
```

VSS2





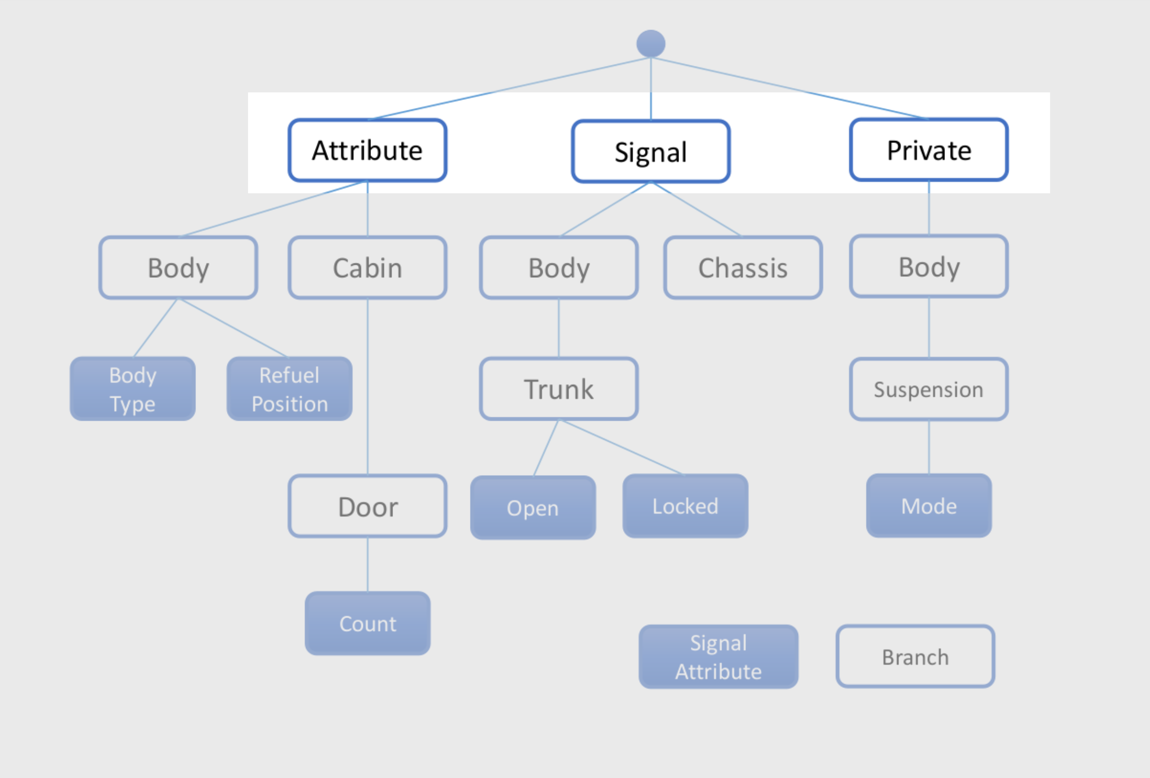
Private branches and leaves should:

- Overwrite pre-existing concepts
- Extend the VSS tree

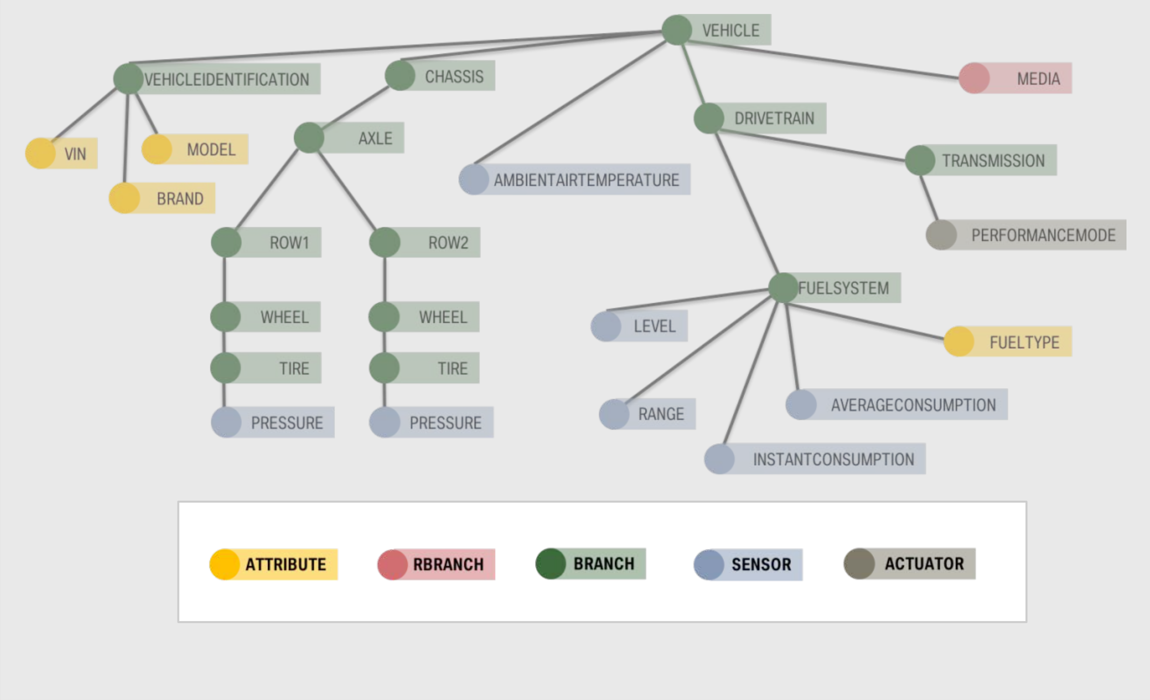
VSS needs consistent position patterns

TYPES

VSS 1 - Attribute/Signal Branch: Attributes and signals were handled as separate branches from the root node, which lead to:
- Duplication in the tree structure
- Leaf properties handled as branches

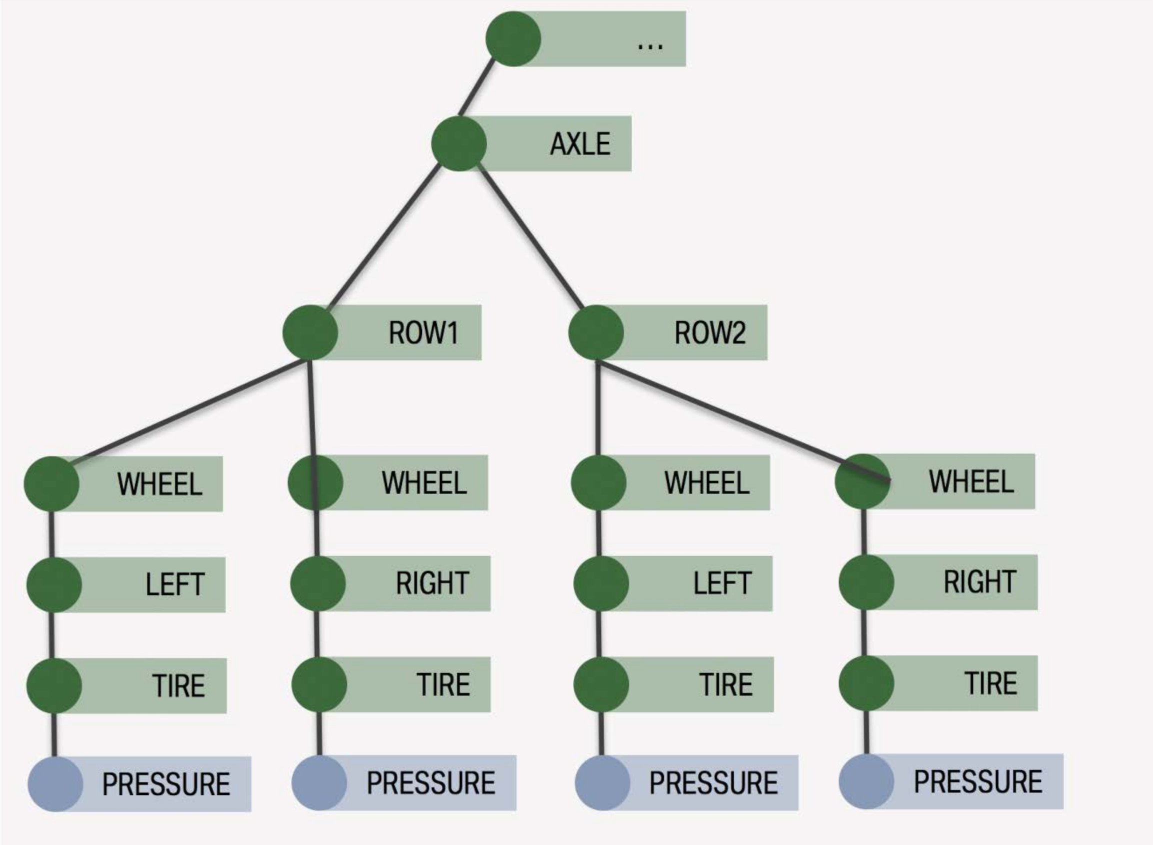


VSS 2 - Introducing new types: To avoid duplication and to add the properties to the leaf, new types were introduced and datatypes got their own property.
- Branch: Node in the tree, which has subnodes
- Sensor: Read-only, which updates in some interval x
- Actuator: sensor + write
- Attribute: read-only and static

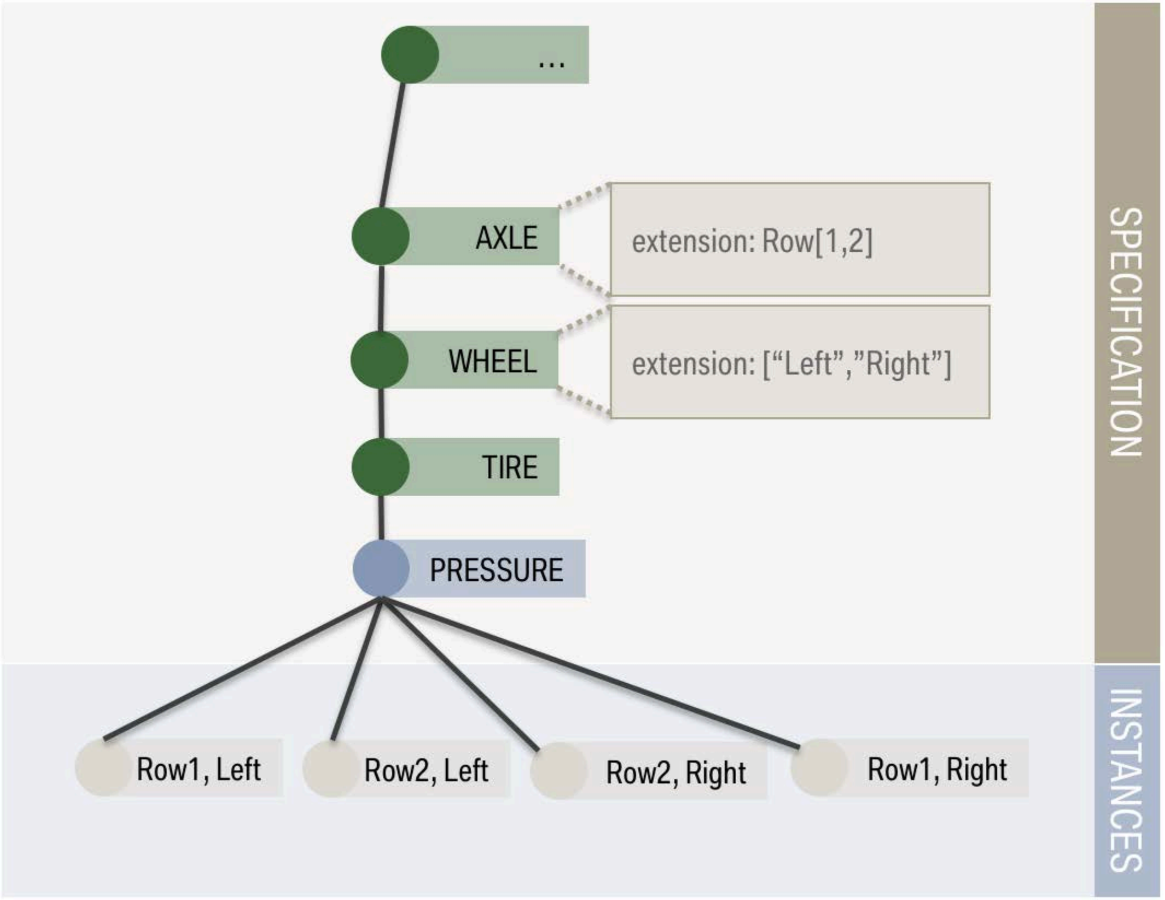


EXTENSIONS

VSS 1 – Extensions as branches: Extensions, often used for positioning, are modeled in the path of the tree. This leads to duplication in the resulting tree (not in tooling), which makes the tree hard to read. Further it hardens filtering and zoning, e.g. like all left tire pressures.

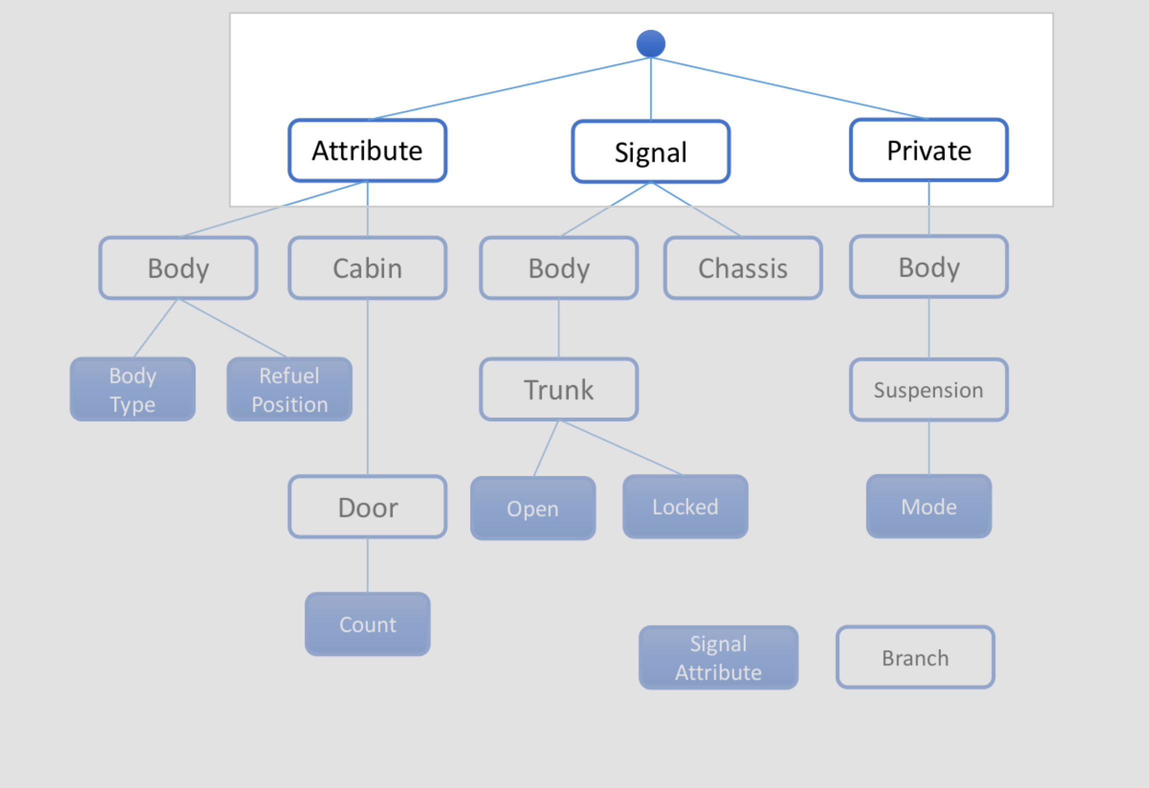


VSS 2 – Extensions as attributes: In VSS 2 extensions are modeled as attributes where they occur. The specification is a straight path to the sensor. The sensor description itself can be seen as “class” and the realization as it’s “instances”. This allows for more flexibility, a cleaner graph representation and zoning and filtering.

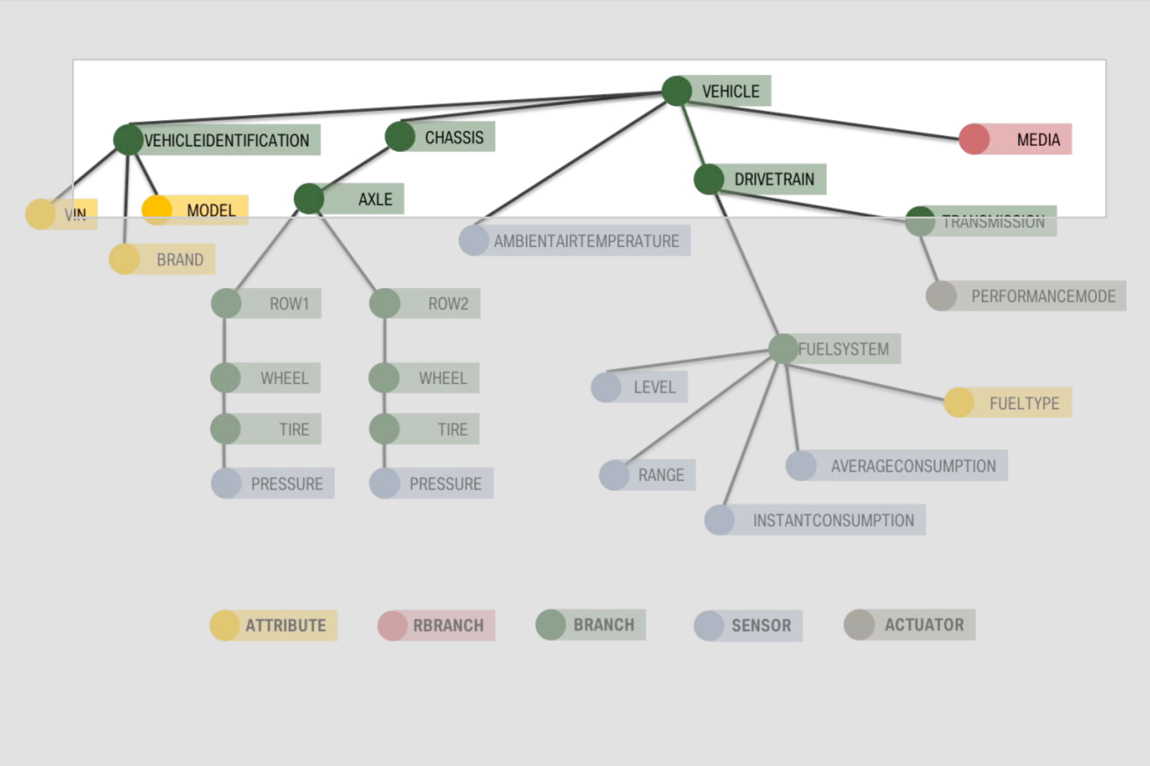


ROOT NODE

VSS 1 - Attribute/Signal Branch: Attributes and signals were handled as separate branches from the root node, which lead to:
- Duplication in the tree structure
- Leaf properties handled as branches



VSS 2 - Introducing new types: To avoid duplication and to add the properties to the leaf, new types were introduced and datatypes got their own property.
- Branch: Node in the tree, which has subnodes
- Sensor: Read-only, which updates in some interval x
- Actuator: sensor + write
- Attribute: read-only and static



CAN WE MAKE RELIABLE PREDICTION FROM VEHICLE DATA?

Vehicle machine learning

In-car learning

Fleet learning

Behavior

- Aggressiveness
- Drowsiness
- Driving style
- Diagnosis

Mental State

- Emotions
- Stress
- Mental load
- Frustration
- Distraction

Environment

- Topology
- Marks
- Potholes
- Obstacles
- Weather

Trajectory patterns

- Maneuvers
- Intents

Data sources:
- Car sensors
- Smartphones/cameras
- Physiological sensors

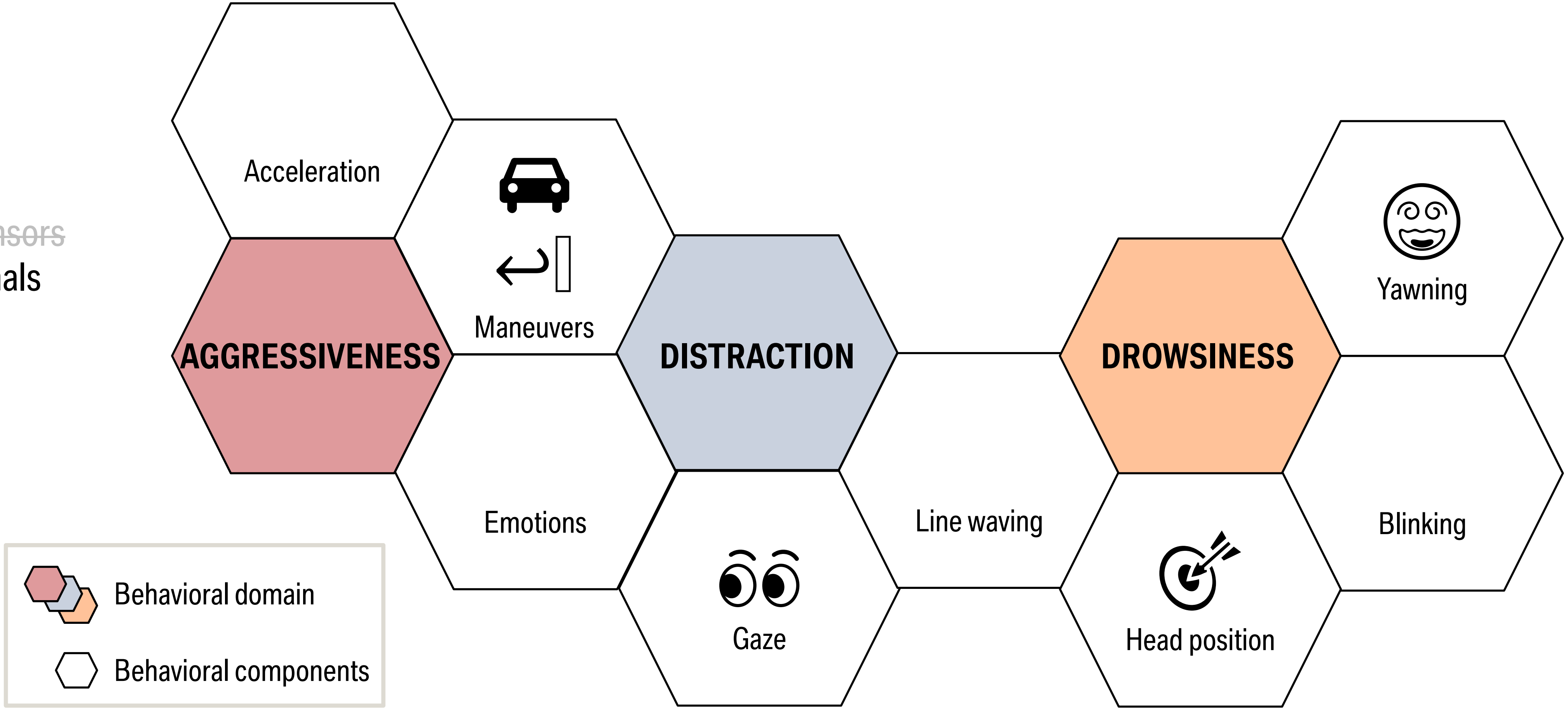
Introduction

Approach

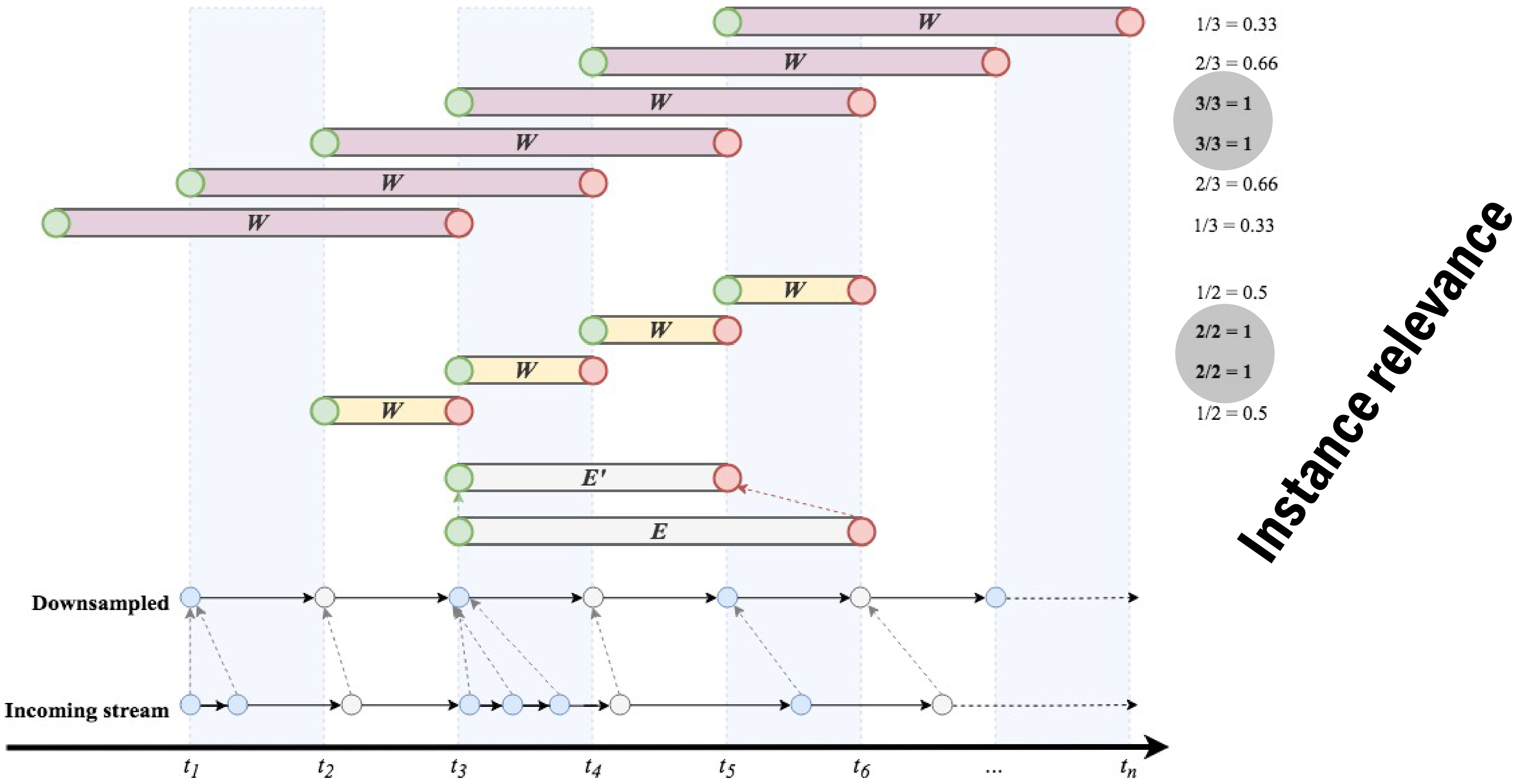
Implementation & Results

Conclusion

Sources:
~~Cameras~~
~~External sensors~~
Vehicle signals

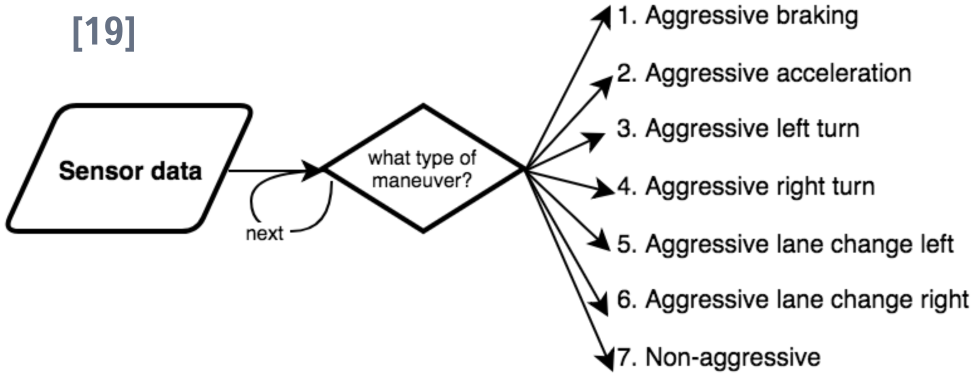


E	Driving event
E'	Driving event after downsampling
W	Time window
t	Time step
(green circle)	Start of a time window
(red circle)	End of a time window
W	Hopping window, where $W_{size} > E'_{size}$
W	Hopping window, where $W_{size} < E'_{size}$

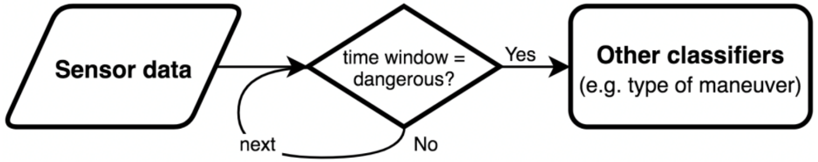


Reference

[19]



Base classifier



Features

[26], [30]

Continuous signals	Categorical signals
Lateral acceleration	Acceleration efficiency
Longitudinal acceleration	Gear *
Accelerator pedal position	Brake pressed
Actual speed	Brake Dynamic Stability Control (DSC) state *
Speed displayed *	
Engine consumption	
Engine RPM speed	
Engine torque	

Random Forest

[19]

Custom parameters	
Window size [frames]	{2, 3, 4, ..., 10}
Minimum instance relevance	{0.1, 0.2, ..., 1.0}
Random Forest	
Number of estimators	{10, 11, 12, ..., 25}
Maximum features	{10, 15, "log2"}
Maximum depth	{5, 10, 15}

RNN

[20]

Recurrent Neural Network	
Number of hidden layers	{1, 2}
Number of recurrent units in the hidden layer	{10, 15, 16, 32, 64, 128}
Recurrent unit type	{LSTM, GRU}
Dropout	{0.1, 0.2}
Recurrent dropout	{0.1, 0.2}

Dataset:
- Manually labeled
- Drivers → 2
- Driving events → 183 (maneuvers)
- Classes → 6
  - Aggressive Turns (L & R)
  - Aggressive Lane change ~~(L & R)~~
  - Aggressive Brake
  - Aggressive Acceleration
  - Normal
- Down sample and aggregate data to 0.5s

- ~13 hours of recorded data
- ~3,5 hours of aggressive driving

Training considerations:
- Evaluation metric → Area Under the ROC Curve (AUC)
- Loss function:
  - Binary classification → Binary cross-entropy
  - Maneuver classification → Categorical cross-entropy
- Input data:
  - Random Forest → Statistical features extracted
    - Mean, median, std. deviation, trend
  - RNN → Min-Max normalization (w.r.t., sensor specs.)

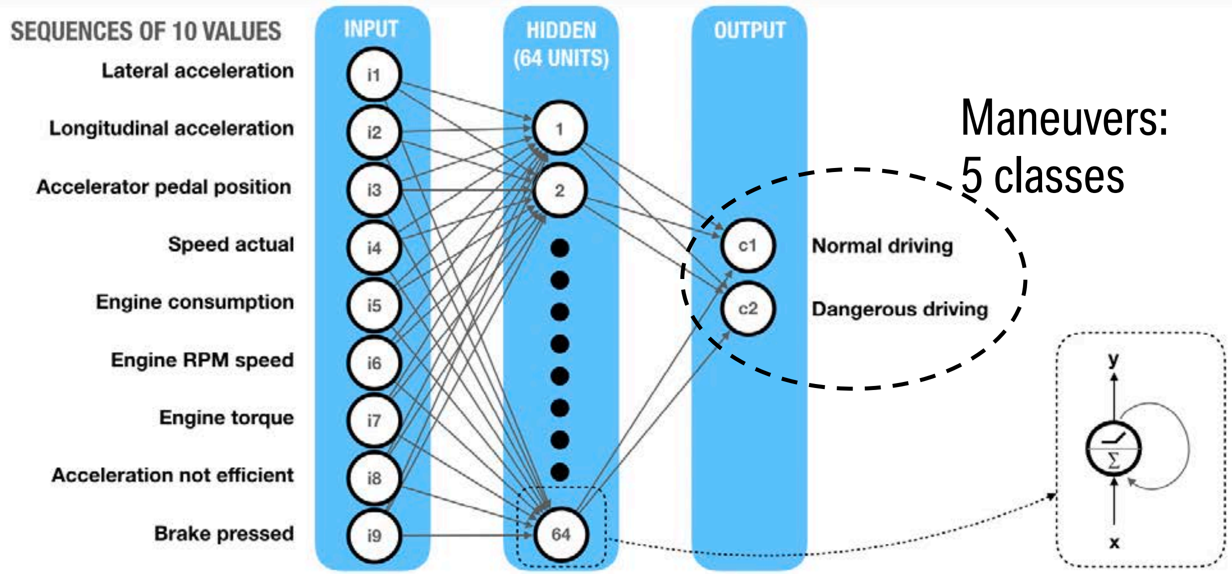
Best found parameters

Random Forest

Parameter \Classifier	Base	Maneuver
Window size [frames]	10	10
Minimum instance relevance	0.9	0.8
Number of estimators	15	24
Maximum features	5	"log2"
Maximum depth	10	15

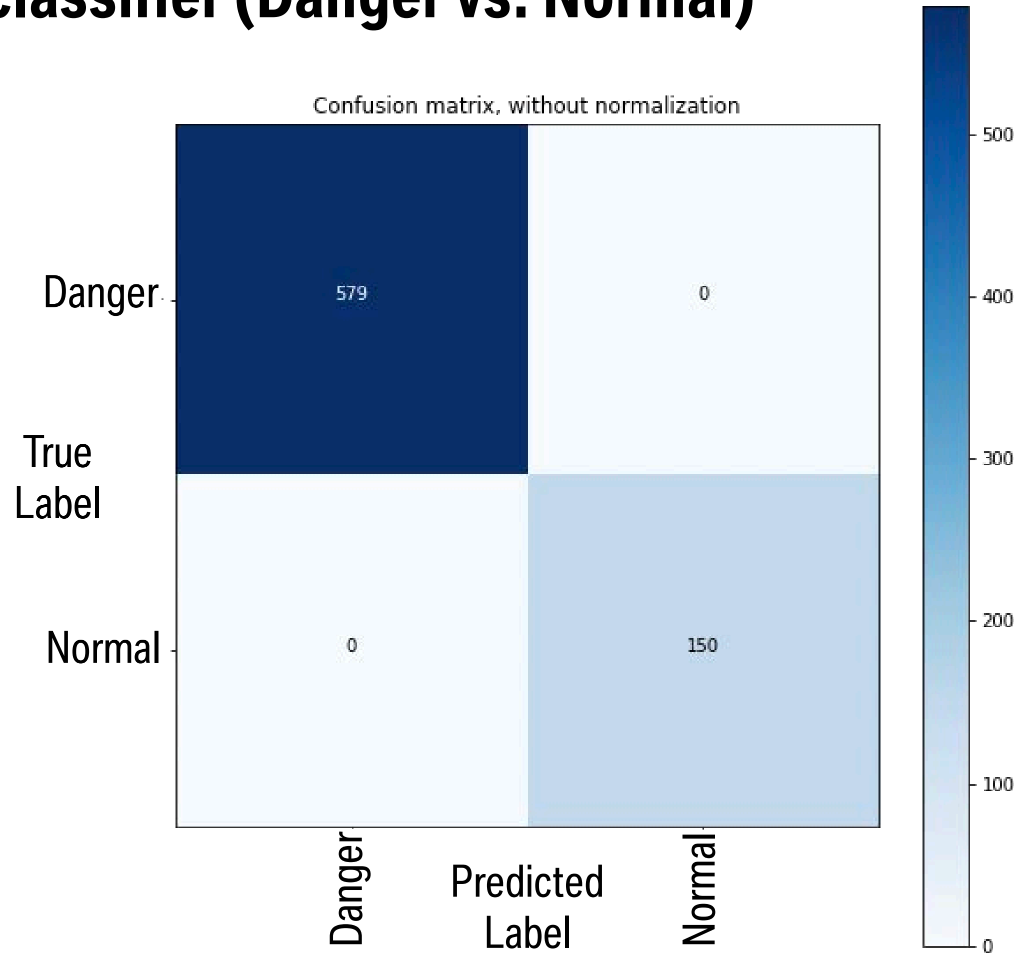
RNN

- LSTM cells
- Input → 9 features
- Hidden layer → x1 (64 units)
- Output → 2 and 5 respectively

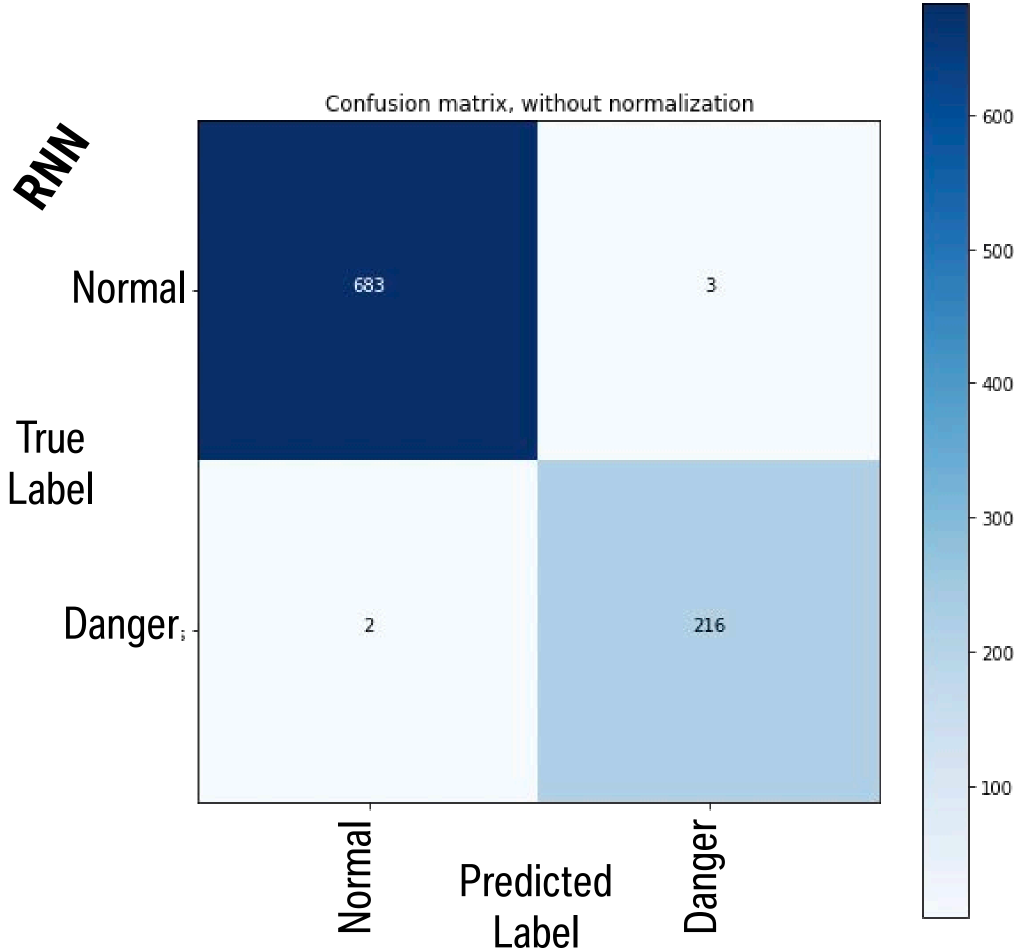


Base classifier (Danger vs. Normal)

Random Forest

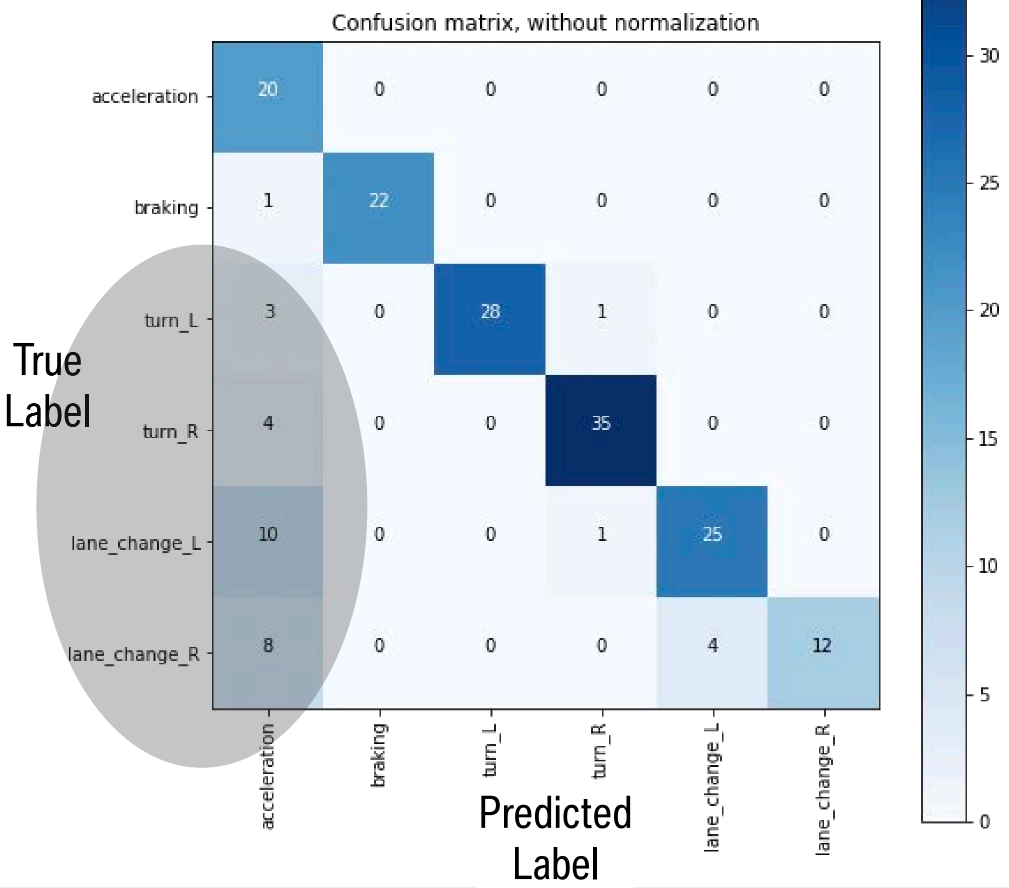


RNN

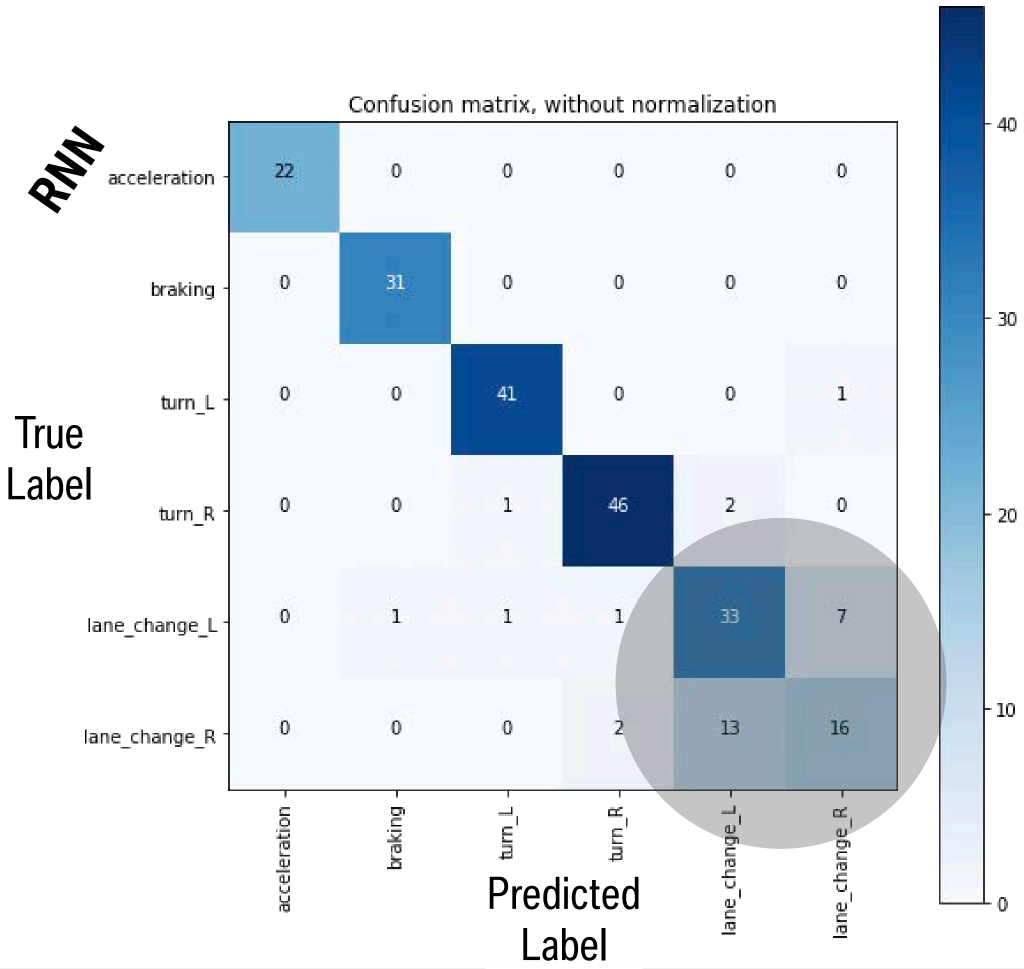


Maneuver classification (6 classes)

Random Forest

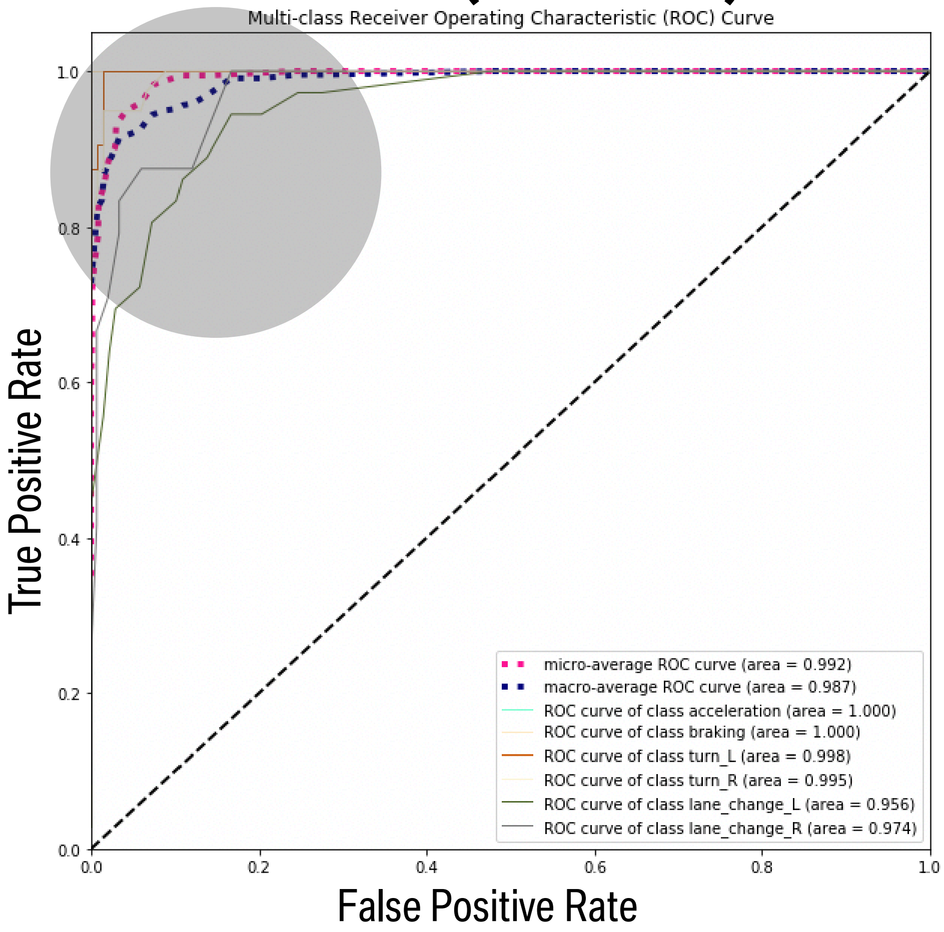


RNN

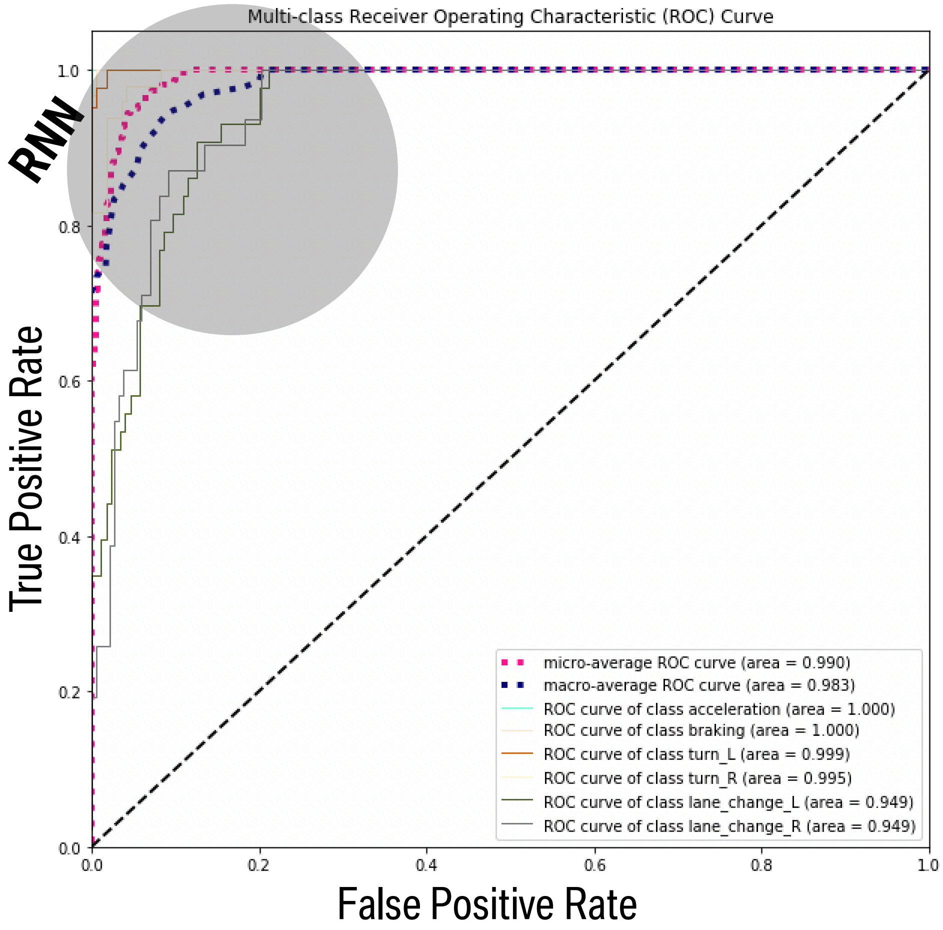


Maneuver classification (6 classes)

Random Forest

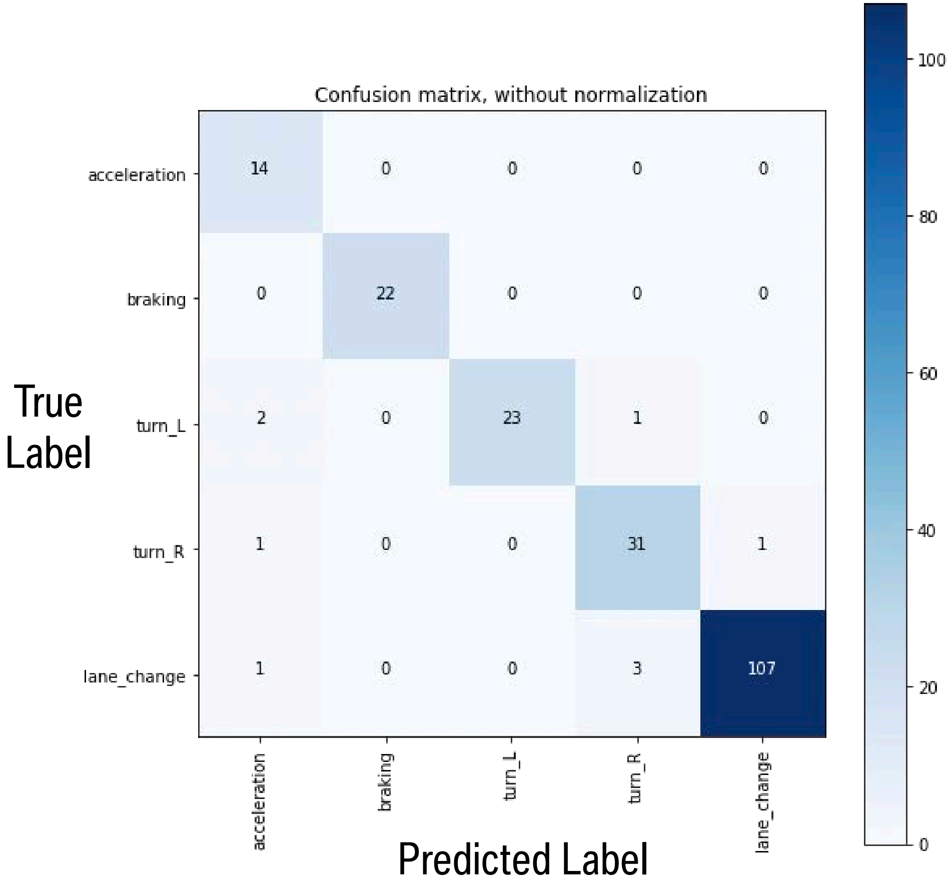


RNN

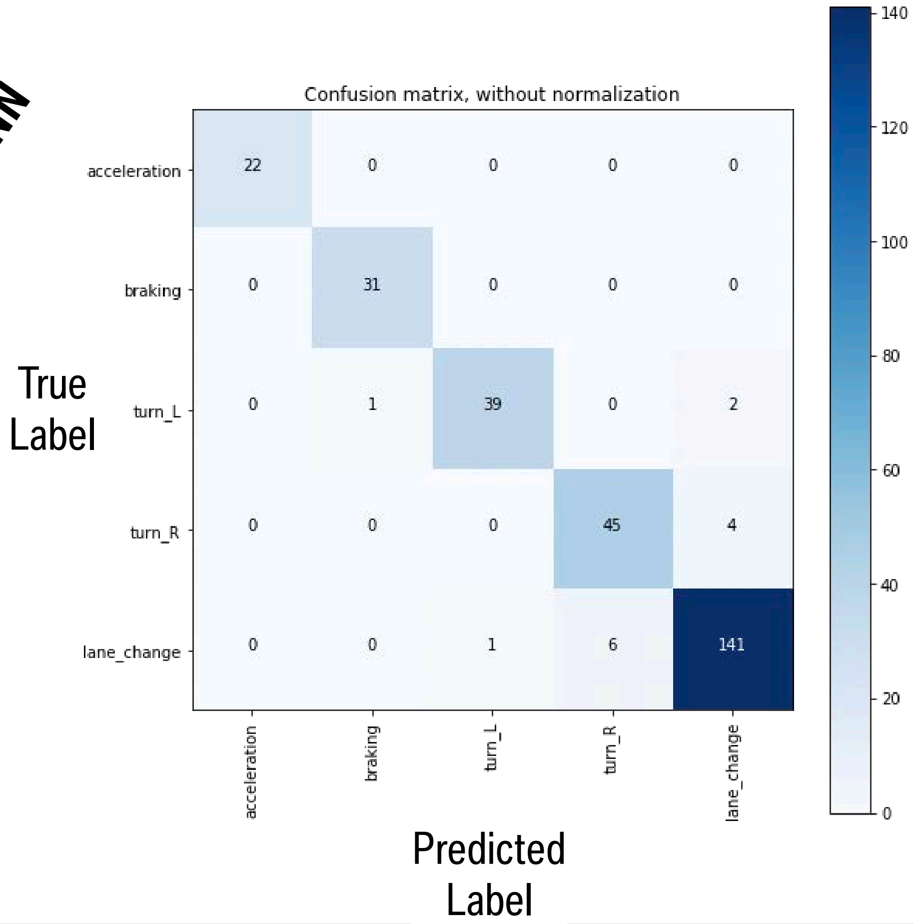


Maneuver classification (5 classes)

Random Forest

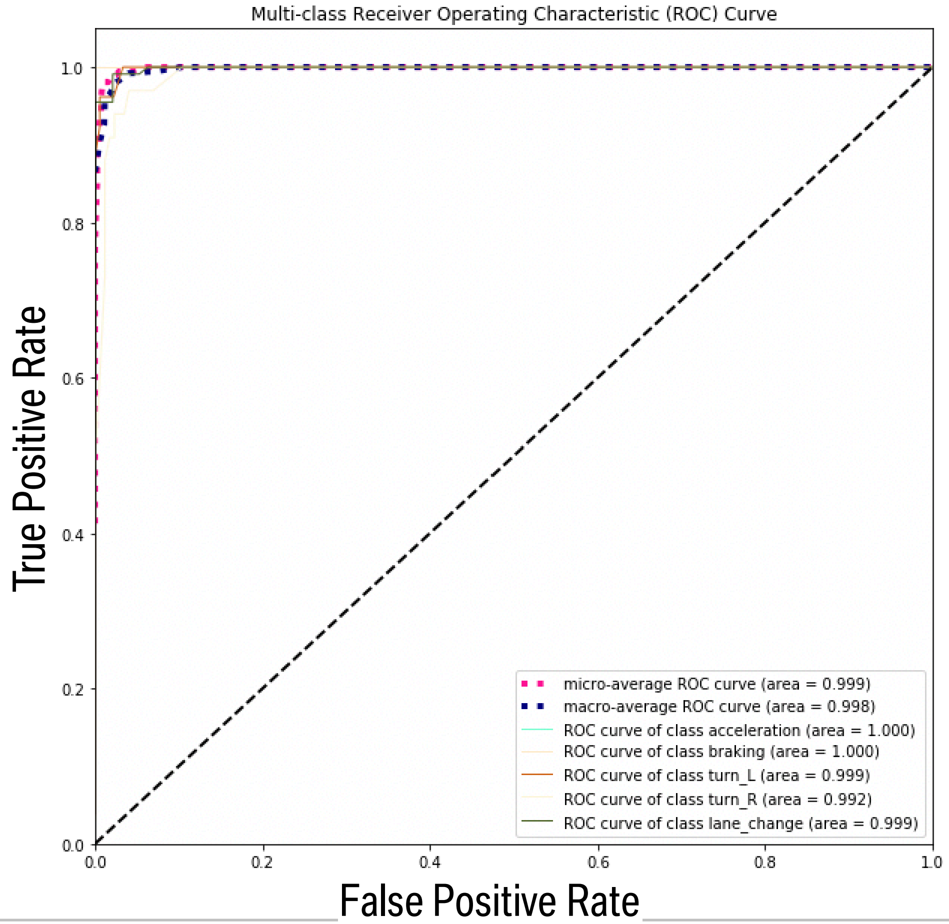


RNN

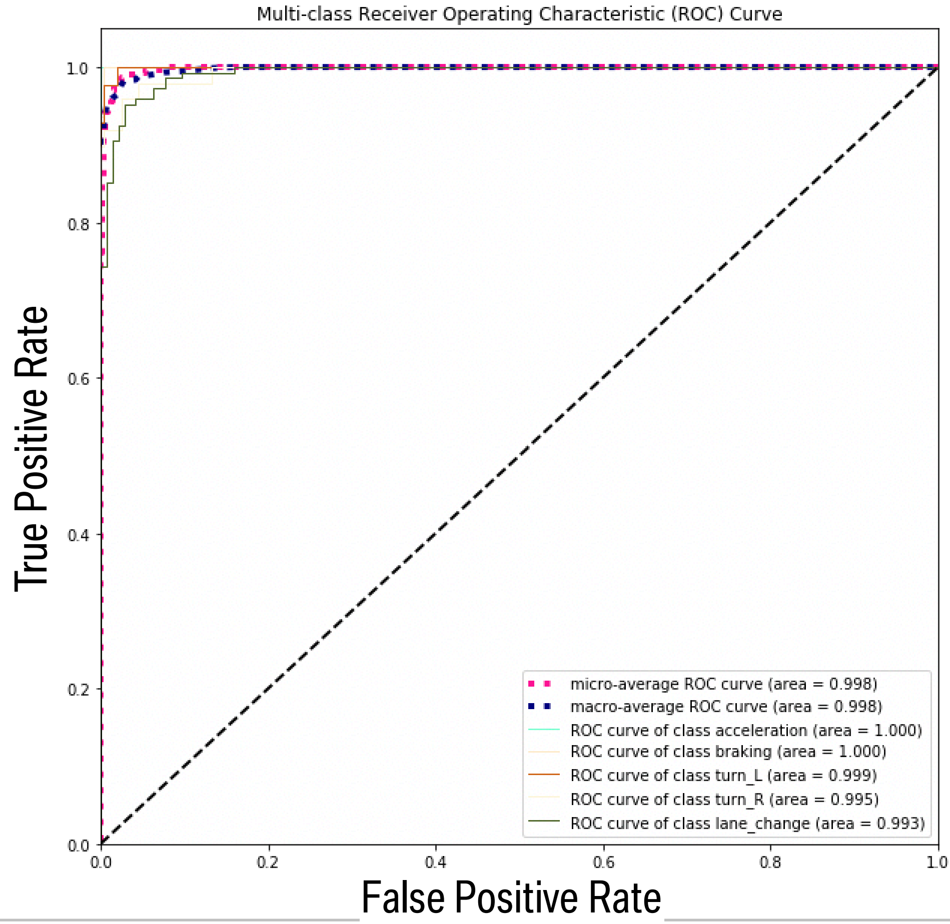


Maneuver classification (5 classes)

Random Forest

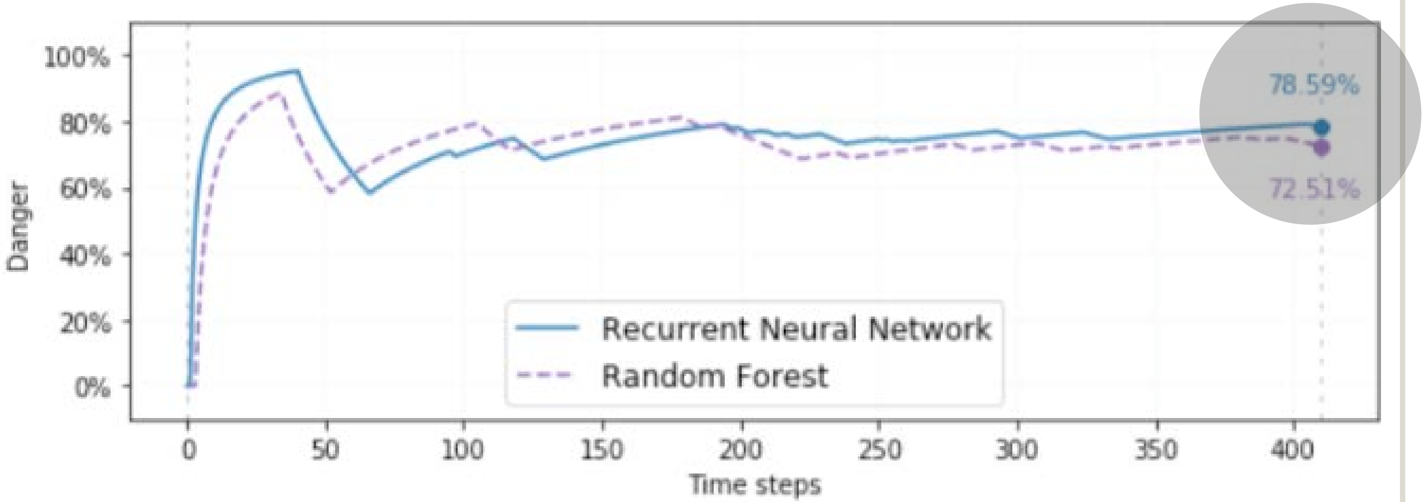


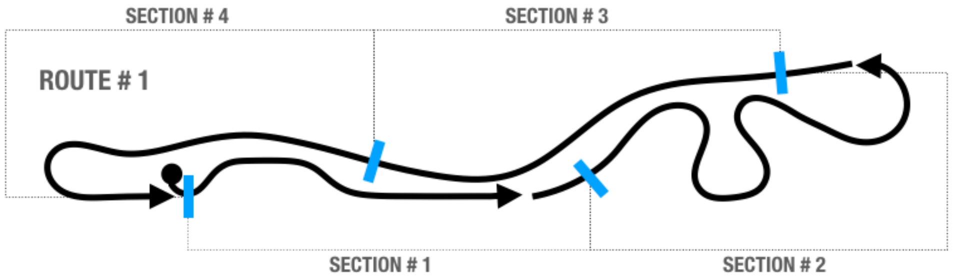
RNN



Instructions

Route	Driver	Instruction
1	A	3 driving styles
1	B	3 driving styles
1	B	2 laps of free driving
2	A	3 driving styles

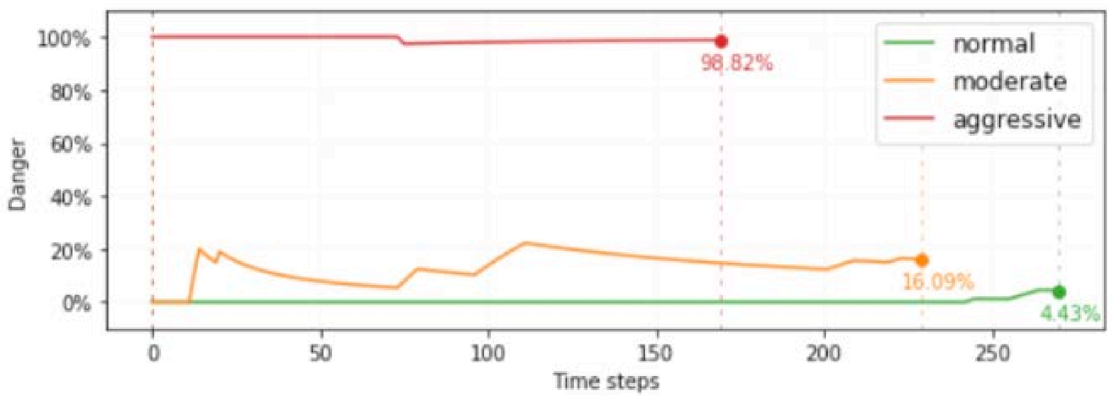




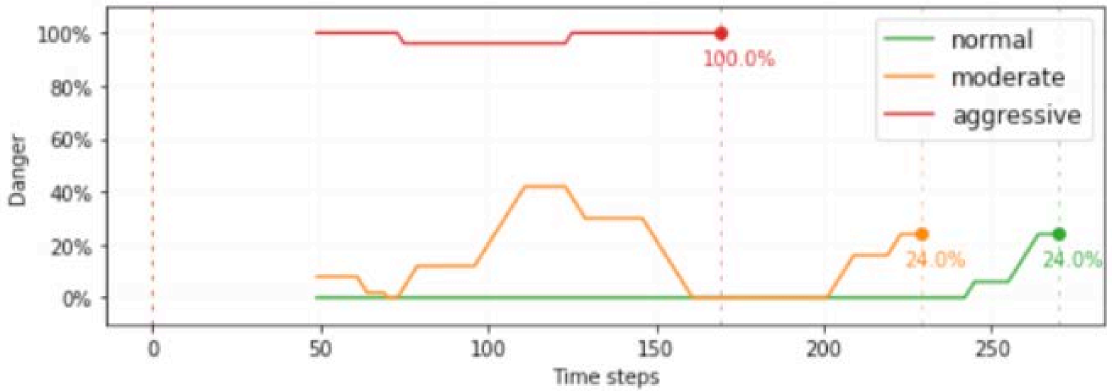
	Section	Co-pilot C	Co-pilot D
Lap 1	1	4	4
	2	2	2
	3	3	2
	4	3	3
Lap 2	1	1	2
	2	2	1
	3	4	3
	4	4	3
Danger perceived		23	20
Maximum possible danger (4 sections x 2 laps x 4)		32	32
Danger perceived [%]		71,875 %	62,5 %
Average danger perceived		67,1875 %	

Instructions

Route	Driver	Instruction
1	A	3 driving styles
1	B	3 driving styles
1	B	2 laps of free driving
2	A	3 driving styles

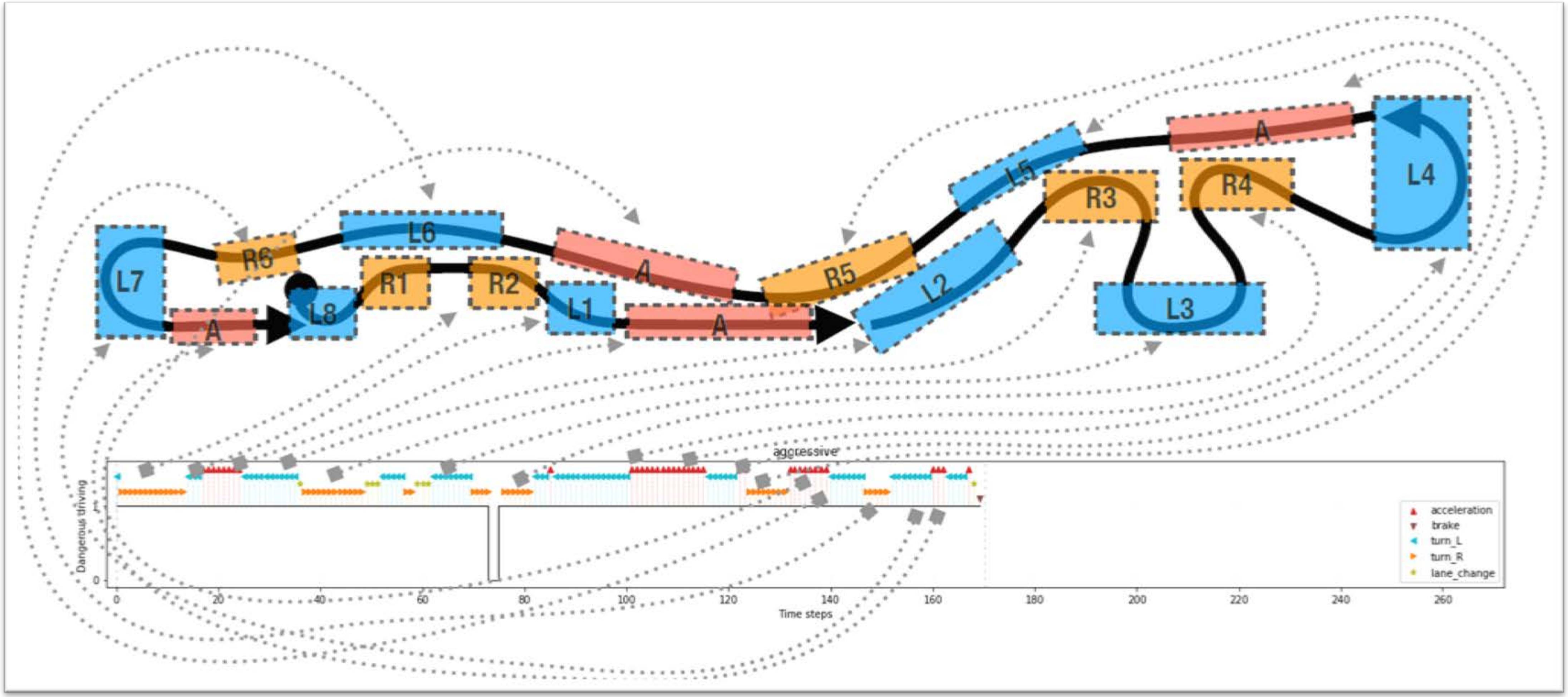


(a) Overall danger score



(b) Score of the past 50 frames

RECONSTRUCTION OF DANGEROUS SITUATIONS



2 biases

Important difference between safe and aggressive

Track shape and speed far from public roads

Daniel Alvarez Coello, Benjamin Klotz, Daniel Wilms, Jorge Marx Gómez, and Raphaël Troncy. Modeling dangerous driving events based on in-vehicle data using Random Forest and Recurrent Neural Network. In 1st International Workshop on Data Driven Intelligent Vehicle Applications (DDIVA), Paris, France, 2019

WEB OF THINGS DEVELOPMENT

Web

IoT platforms

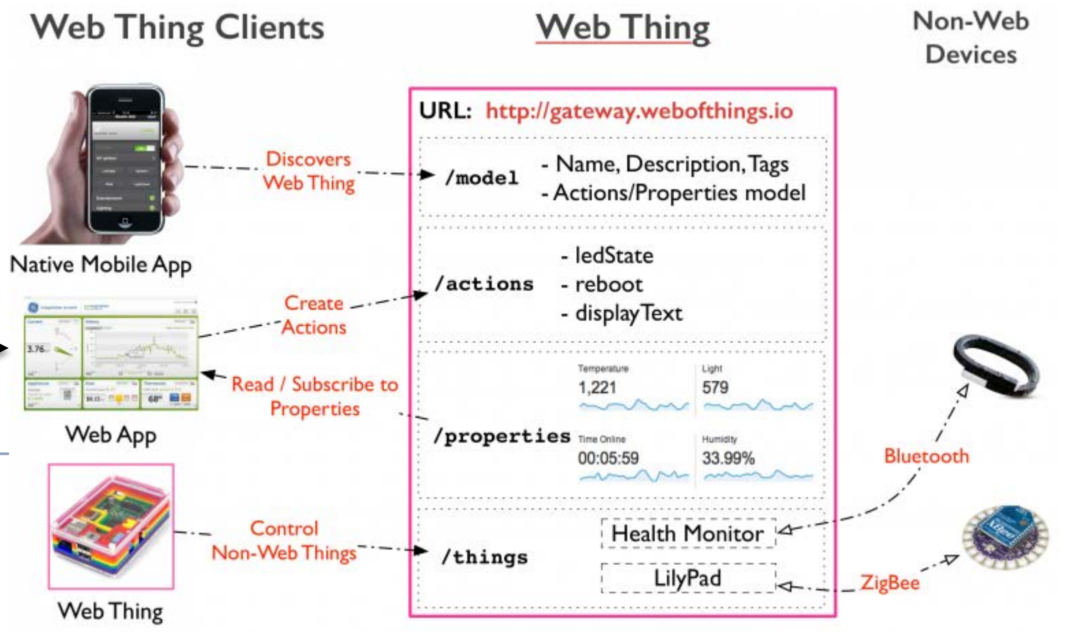
Devices

SSN/SOSA Domain ontology

W3C WoT servient

Limited vehicle sensor/actuator access

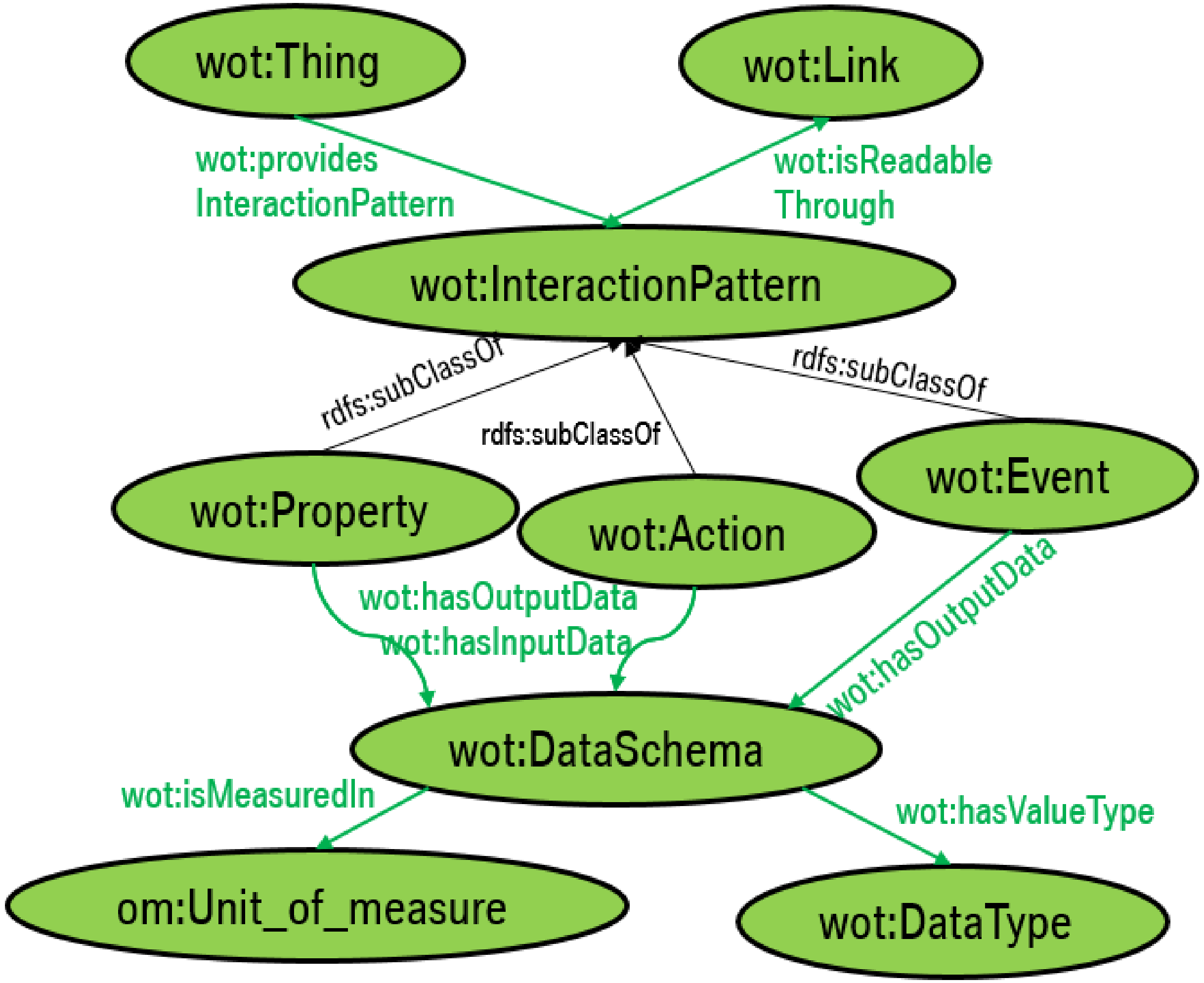
Auto WG



Source: Building the Web of Things: book.webofthings.io
Creative Commons Attribution 4.0

WOT ONTOLOGY

- Define a wot:Thing
- Centered on wot:interactionPattern
  - Properties
  - Actions
  - Events
- Use dataSchema
  - Literal value
  - wot:DataType
  - om:Unit_of_measure



http://iot.linkeddata.es/def/wot/index-en.html

AUTOMOTIVE WEB THINGS: CHALLENGES

Domain vs Nature of

– Things
– Interactions

```
"@id": "property/acceleration“,
"@type": ["Property","vsso:LongitudinalAcceleration","iot:Property"],
```

Complexity of vehicles

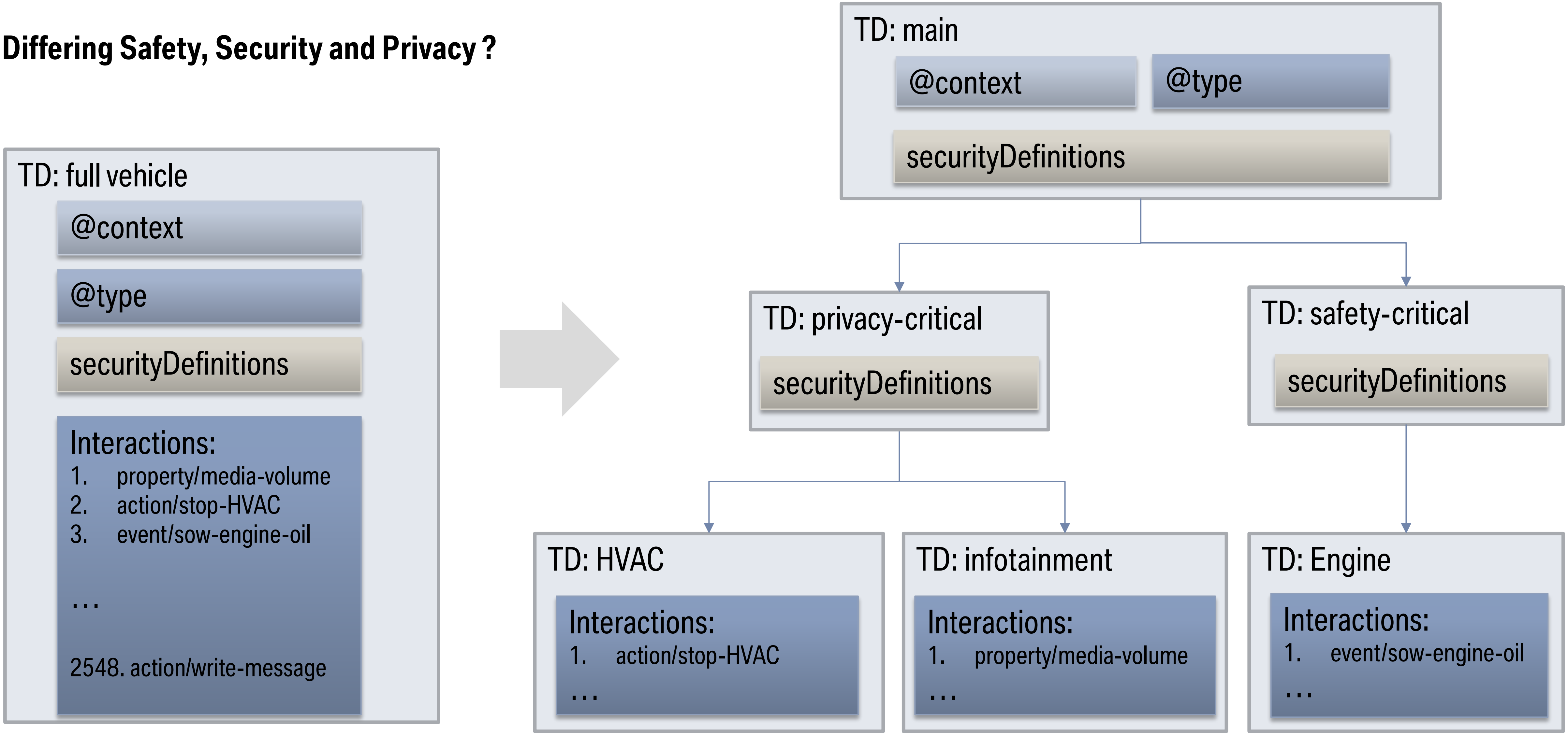
– Different access control and security
– Different expertise

Data access

– External hardware
– Legacy solutions

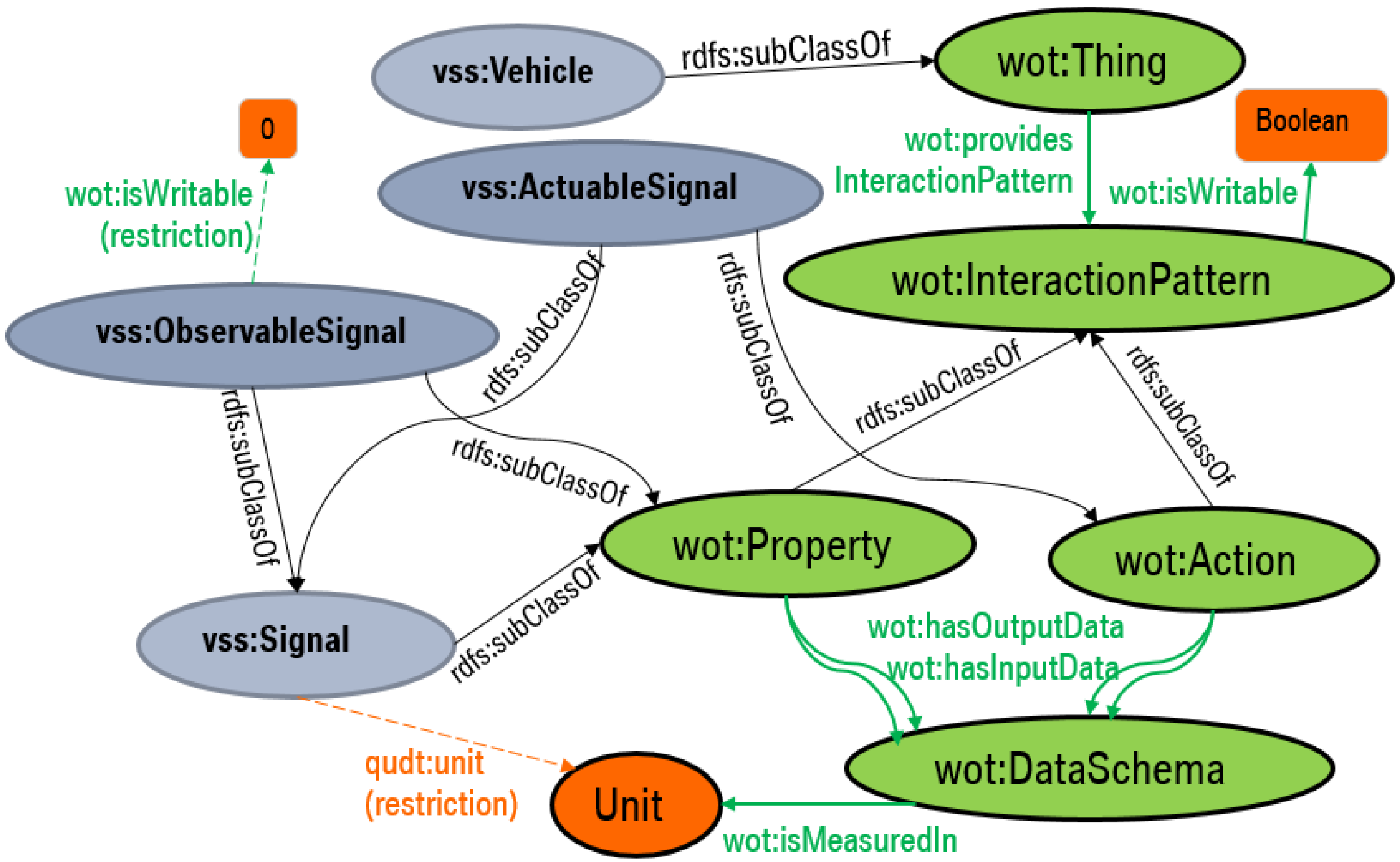
DESCRIBING COMPLEX THINGS IN THE WOT

Differing Safety, Security and Privacy ?



Requirements included in the latest charter update (June 2019)

ALIGNING WOT – SOSA FOR THE AUTOMOTIVE DOMAIN



Modeling pattern:

i. Vehicles are Things

ii. Signals are properties

Read-write depending on the signal type

iii. Actuatable signals are actions

iv. DataSchema use the domain Units

Benjamin Klotz, Soumya Kanti Datta, Daniel Wilms, Raphael Troncy, and Christian Bonnet. A car as a semantic web thing: Motivation and demonstration. In 2nd Global Internet of Things Summit (GIoTS'18), Bilbao, Spain, June 2018.

VEHICLE THING DESCRIPTION

Thing

```
"@context": ["https://www.w3.org/2019/wot/td/v1",
    {"auto": "https://auto.schema.org/" ,
    "iot": "https://iotschema.org/" ,
    "vsso": "https://automotive.eurecom.fr/vsso#" ,
    "qudt":"http://www.qudt.org/1.1/vocab/unit#"}
  ],
  "@type" : ["Thing","auto:Car","vsso:Vehicle"],
  "id": "http://10.159.160.74:5001/WBY8P61020VD33272/",
"title": "MyCarThing",
  "name": "MyCarThing",
  "auto:brand":"BMW",
  "auto:model":"i3",
  "vsso:vin":"WBY8P61020VD33272",
```

Properties

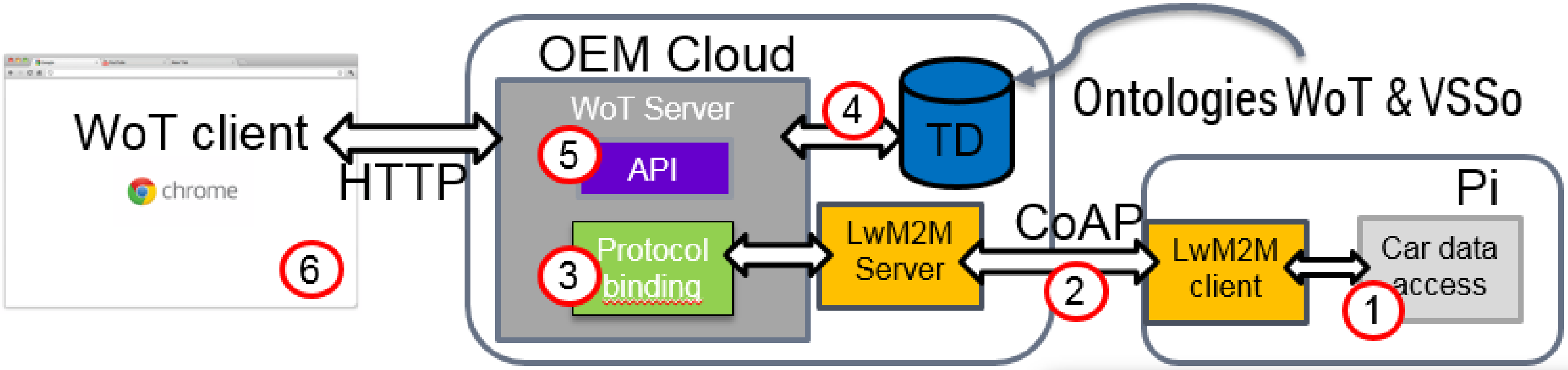
```
"properties": {
    "secured": {
      "@type" : ["iot:Property“,”vsso:DoorLock”],
      "description" : "Shows the current lock status of the car",
      "type": "string",
      "forms": [{
       "href": "property/secured",
        "contentType": "application/json",
        "op":"readproperty"
      }]
```

Actions

```
"actions": {
    "write-message": {
      "@type" : ["Action“,”iot:ChangePropertyAction”],
      "description" : "Send message to the vehicle HMI",
      "safe": false,
      "idempotent": false,
      "input": {
        "type": "object",
        "properties": {
          "subject": {
            "type": "string"
          },
          "message": {
            "type": "string"
          }
        },
        "required": ["subject","message"]
      },
      "forms": [{
        "href": "action/message",
        "contentType": "application/json",
        "op":"invokeaction"
      }]
    },
```

Semantics | Communication | Access

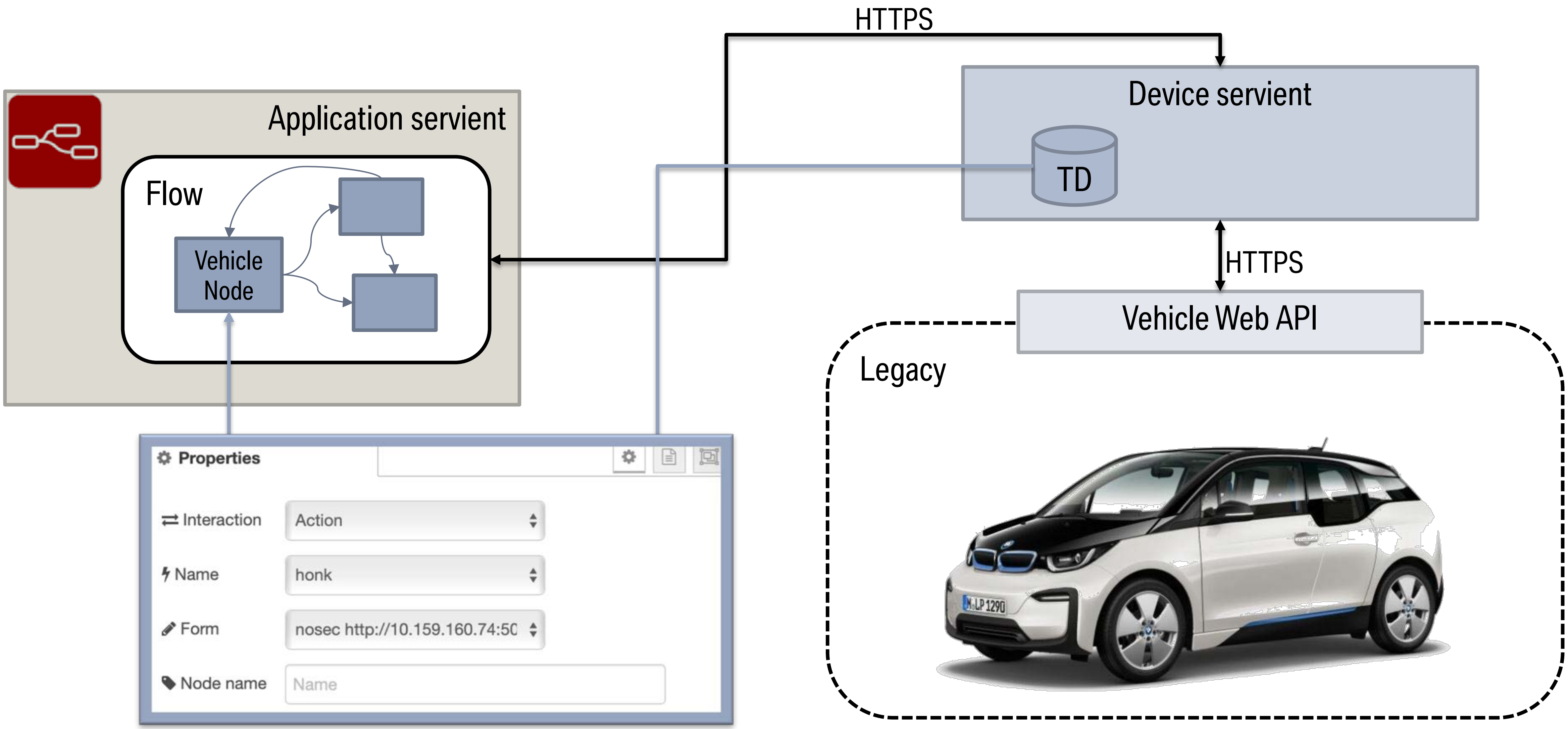
DEMONSTRATION: CONTROLLING YOUR CAR WITH YOUR WEB BROWSER (2017)



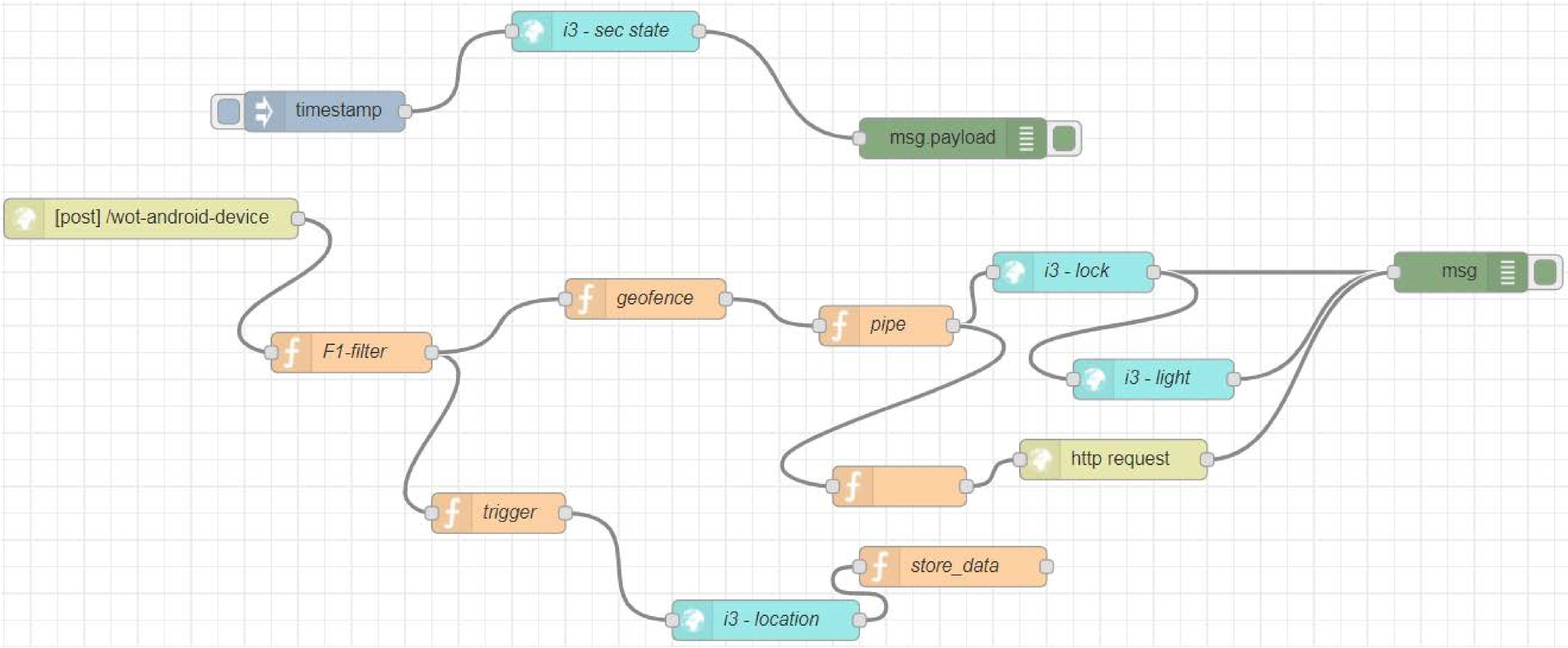
Usage:

Behind a web browser, we can control the windows, doors and honk

DEMONSTRATION: INTUITIVE INTERFACE ON LEGACY VEHICLES



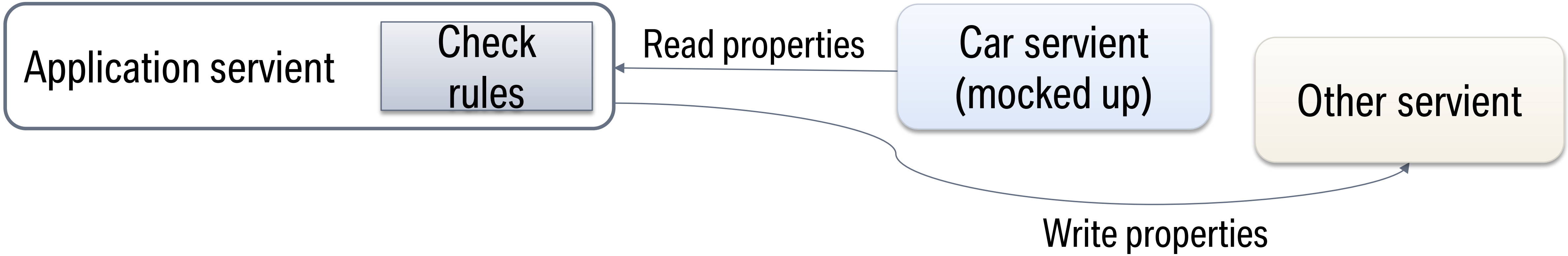
FLOW EXAMPLE (OPEN DAY)



When the user is far from the unlocked vehicle, the vehicle lock is activated

DEMONSTRATION: CAR INTERACTION IN THE WEB OF THINGS

Triggering actions based on rules:
- If a door is open while speed>0 then trigger a warning
- If longitudinal and lateral acceleration are high (dangerous driving), then turn on a red LED
- If the coordinates of the car are close to a fixed destination, control a garage door an light



https://www.youtube.com/watch?v=pjgTLPlAsKQ
https://www.youtube.com/watch?v=zkL8Cdgy8PE

OUTLOOK AND PERSPECTIVES

Outlook: a step to make connected vehicles more interoperable

- **Ontology for the Automotive domain / alignment with the Web of Things**
  - VSSo (http://automotive.eurecom.fr/vsso) … being further developed in the W3C Automotive Business Group
  - Alignment VSSo/VDC/WoT (axioms to publish)
- **Datasets: semantic trajectories**
  - VSSo data: 5 trajectories, 8 signals, 16k observations (http://automotive.eurecom.fr/trajectory)
  - VDC data: 2 trajectories, 16 states, 16k observations, 130 events
- **Classifiers (Internal BMW)**
  - Aggressive driving prediction / Maneuvers prediction
- **WoT vehicles: https://www.youtube.com/watch?v=zkL8Cdgy8PE**

Many remaining challenges:

- **Next generation vehicle server: VISS server (implementations, w3c recommendation), RSI (or Gen2)**
- **In-vehicle machine learning**
- **Scalability of graph-based data representation (vehicle shadow data access using GraphQL)**
- **How much your vehicle should remember your driving patterns?**
- **Security / Privacy / Ethics / add yours …**