

Vehicle Signal Specification + Ontology

Status update 04/22

DATA CENTRIC AS AN ANSWER TO INCREASING COMPLEXITY

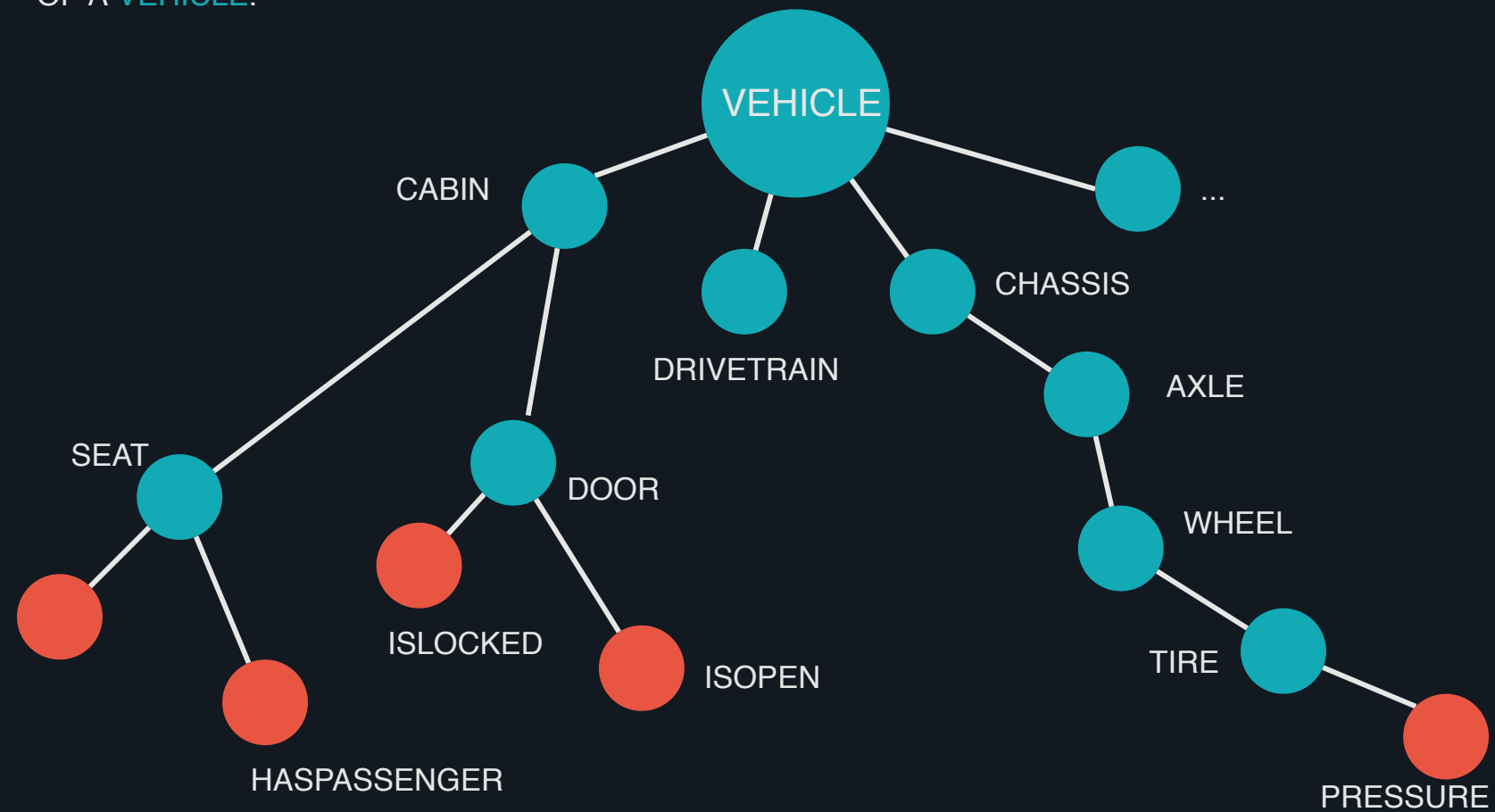
A good data-centric architecture is designed with **data sharing** in mind.

The **data model** is **permanent** and **enduring**, and applications come and go.

It's not something you **BUY**, it's something you **DO**!

VEHICLE SIGNAL SPECIFICATION (VSS)

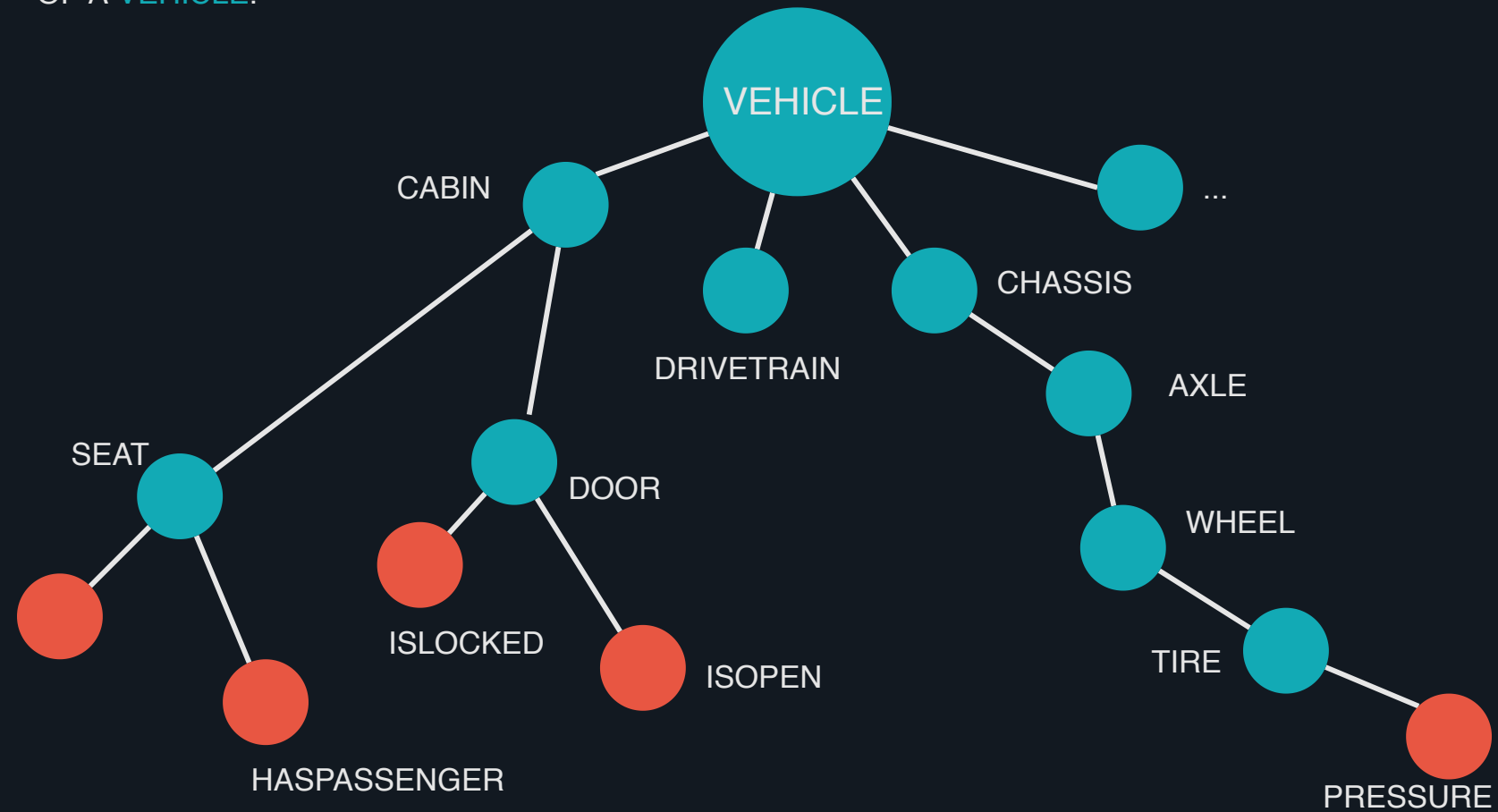
TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



VSS offers a simple, flexible and protocol agnostic way of describing vehicle signals.

VEHICLE SIGNAL SPECIFICATION (VSS)

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



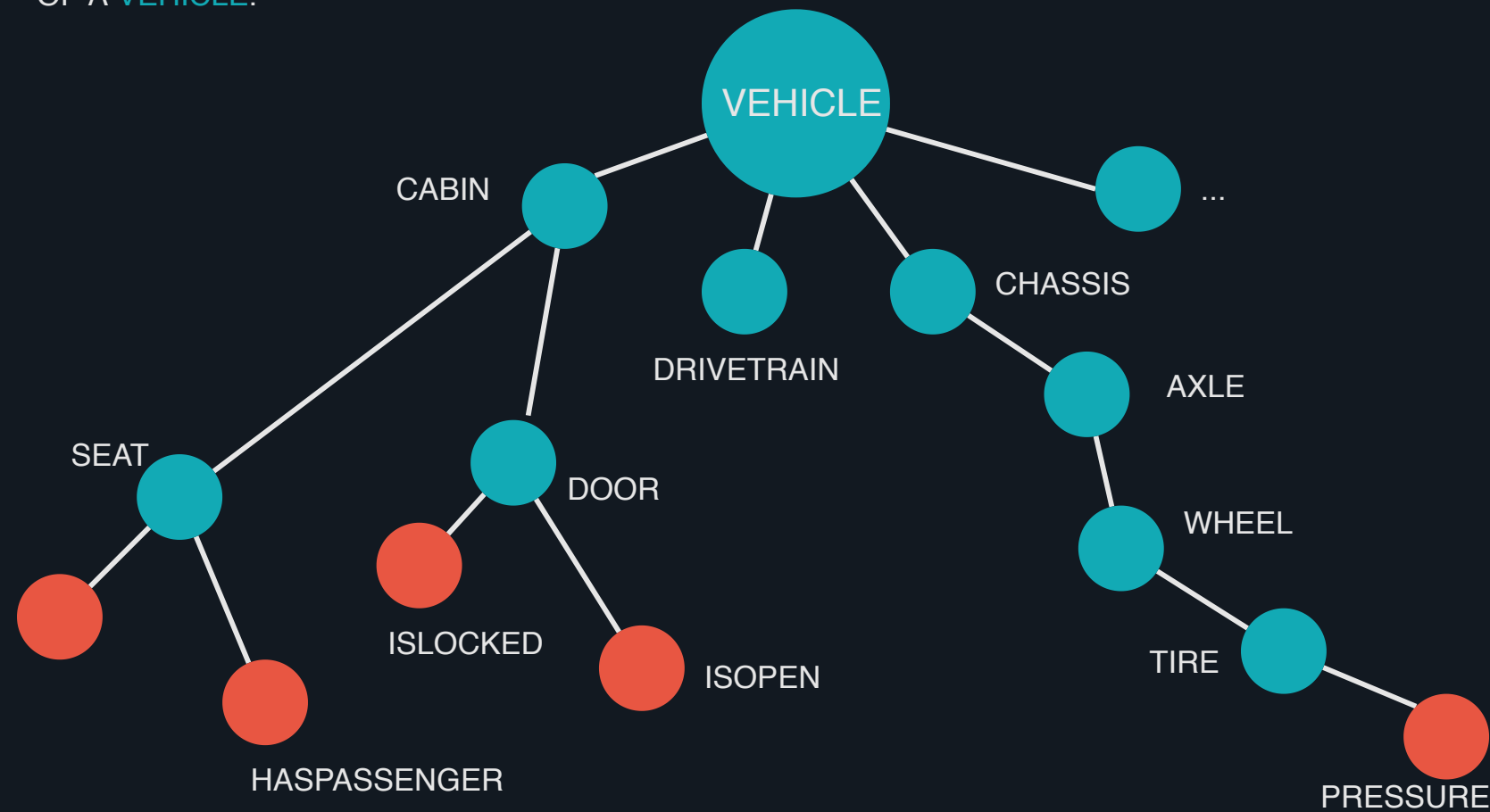
YAML SPECIFICATION

```
Vehicle.Drivetrain.Transmission.Speed  
type: sensor  
datatype: float  
unit: km/h  
description: The vehicle speed as measured by the drivetrain
```

VSS offers a simple, flexible and protocol agnostic way of describing vehicle signals.

VEHICLE SIGNAL SPECIFICATION (VSS)

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



YAML SPECIFICATION

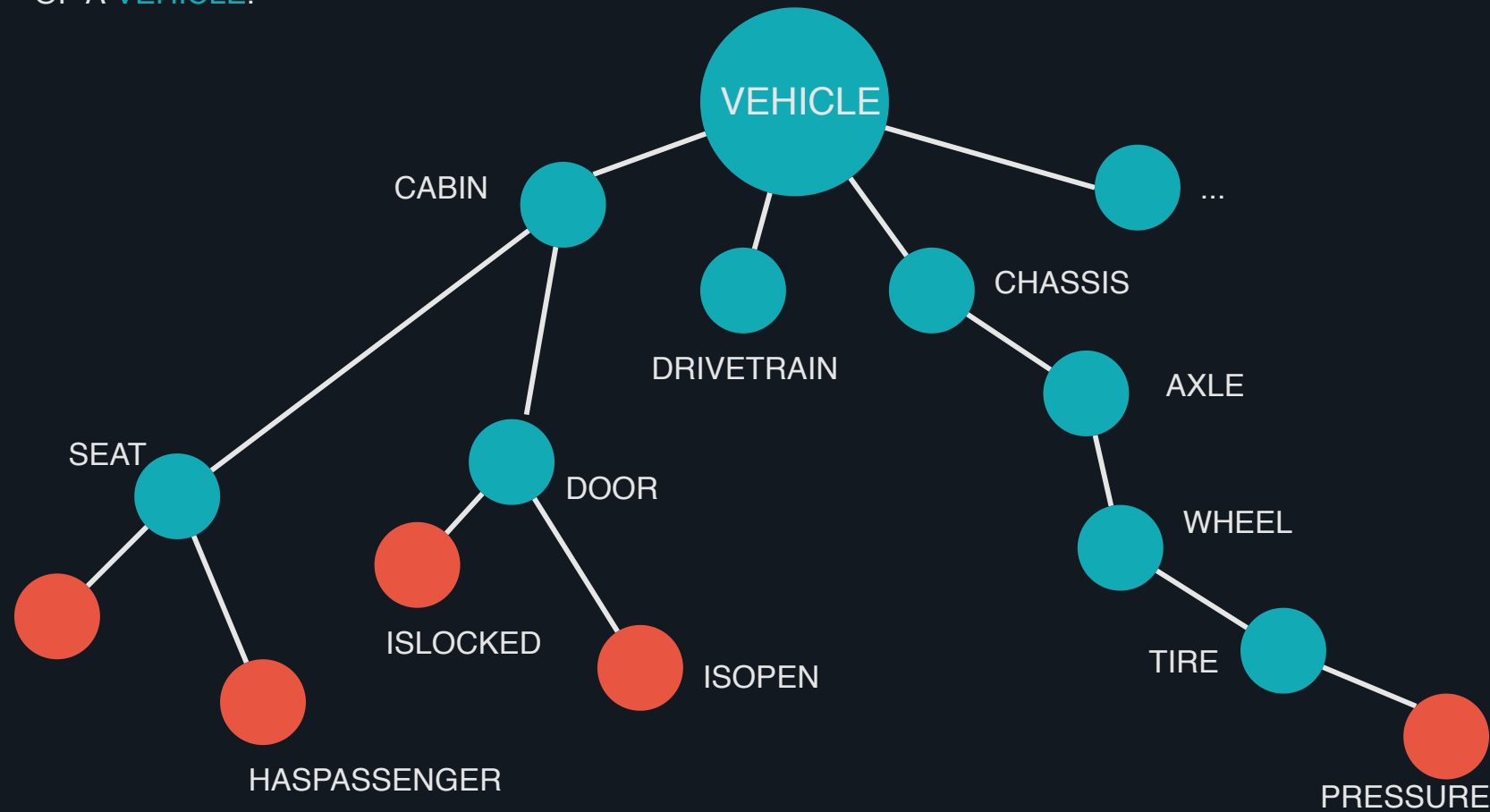
Vehicle.Drivetrain.Transmission.Speed

```
type: sensor
datatype: float
unit: km/h
description: The vehicle speed as measured by the drivetrain
```

VSS offers a simple, flexible and protocol agnostic way of describing vehicle signals.

VEHICLE SIGNAL SPECIFICATION (VSS)

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



YAML SPECIFICATION

Vehicle.Drivetrain.Transmission.Speed

type: sensor

datatype: float

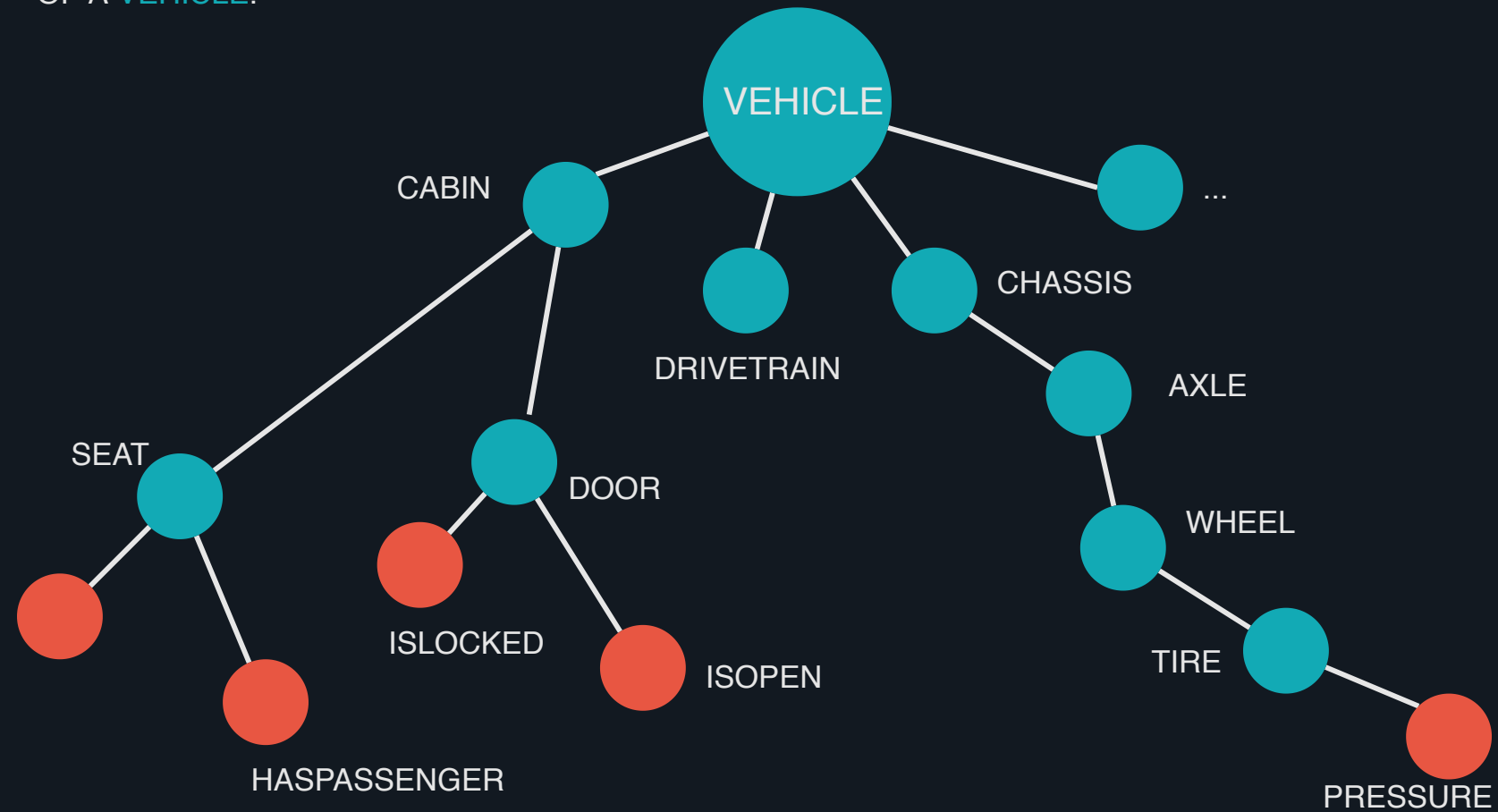
unit: km/h

description: The vehicle speed as measured by the drivetrain

VSS offers a simple, flexible and protocol agnostic way of describing vehicle signals.

VEHICLE SIGNAL SPECIFICATION (VSS)

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



YAML SPECIFICATION

```
Vehicle.Drivetrain.Transmission.Speed  
type: sensor
```

datatype: float

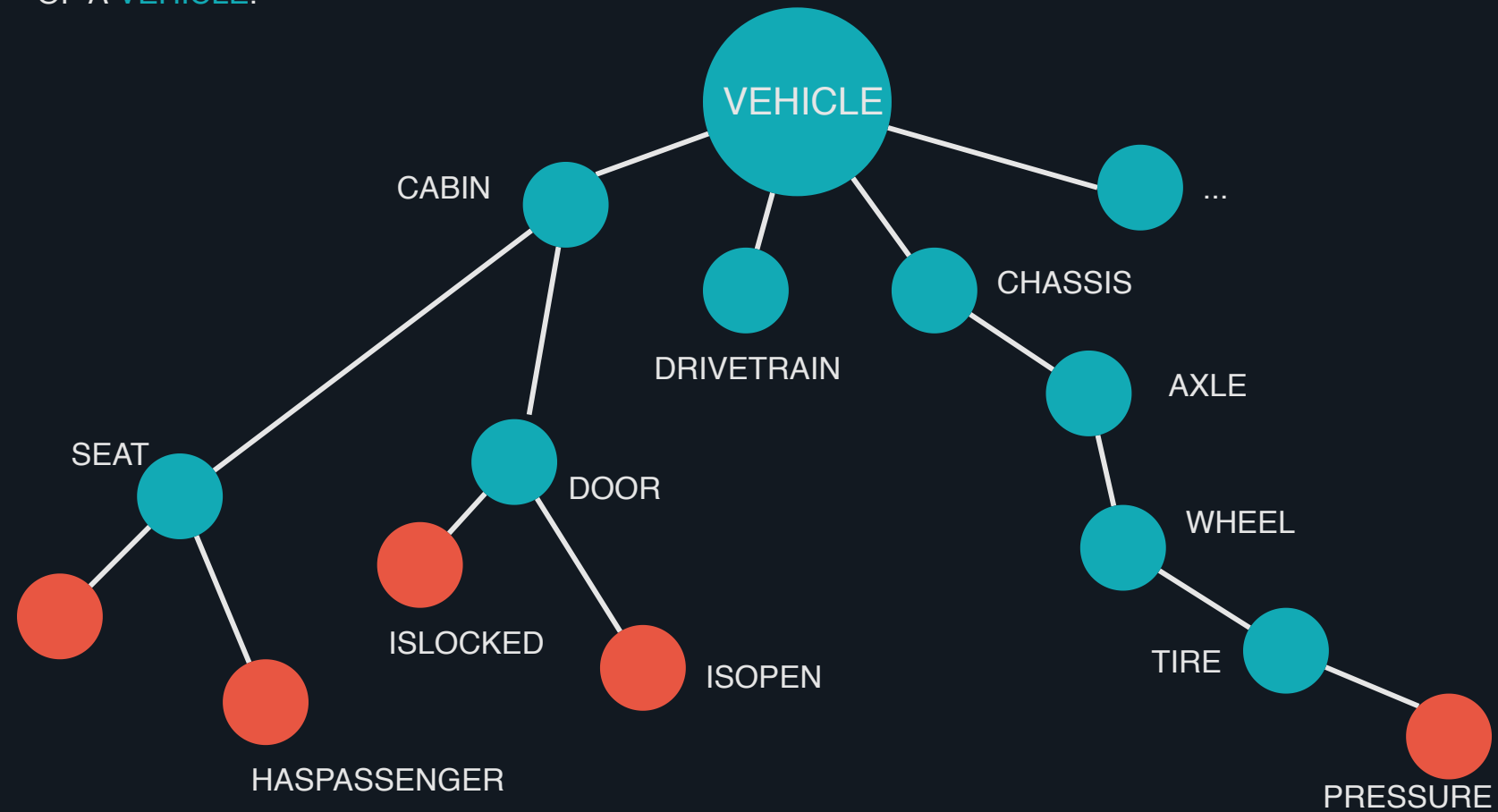
```
unit: km/h
```

```
description: The vehicle speed as measured by the drivetrain
```

VSS offers a simple, flexible and protocol agnostic way of describing vehicle signals.

VEHICLE SIGNAL SPECIFICATION (VSS)

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



YAML SPECIFICATION

```
Vehicle.Drivetrain.Transmission.Speed  
type: sensor  
datatype: float
```

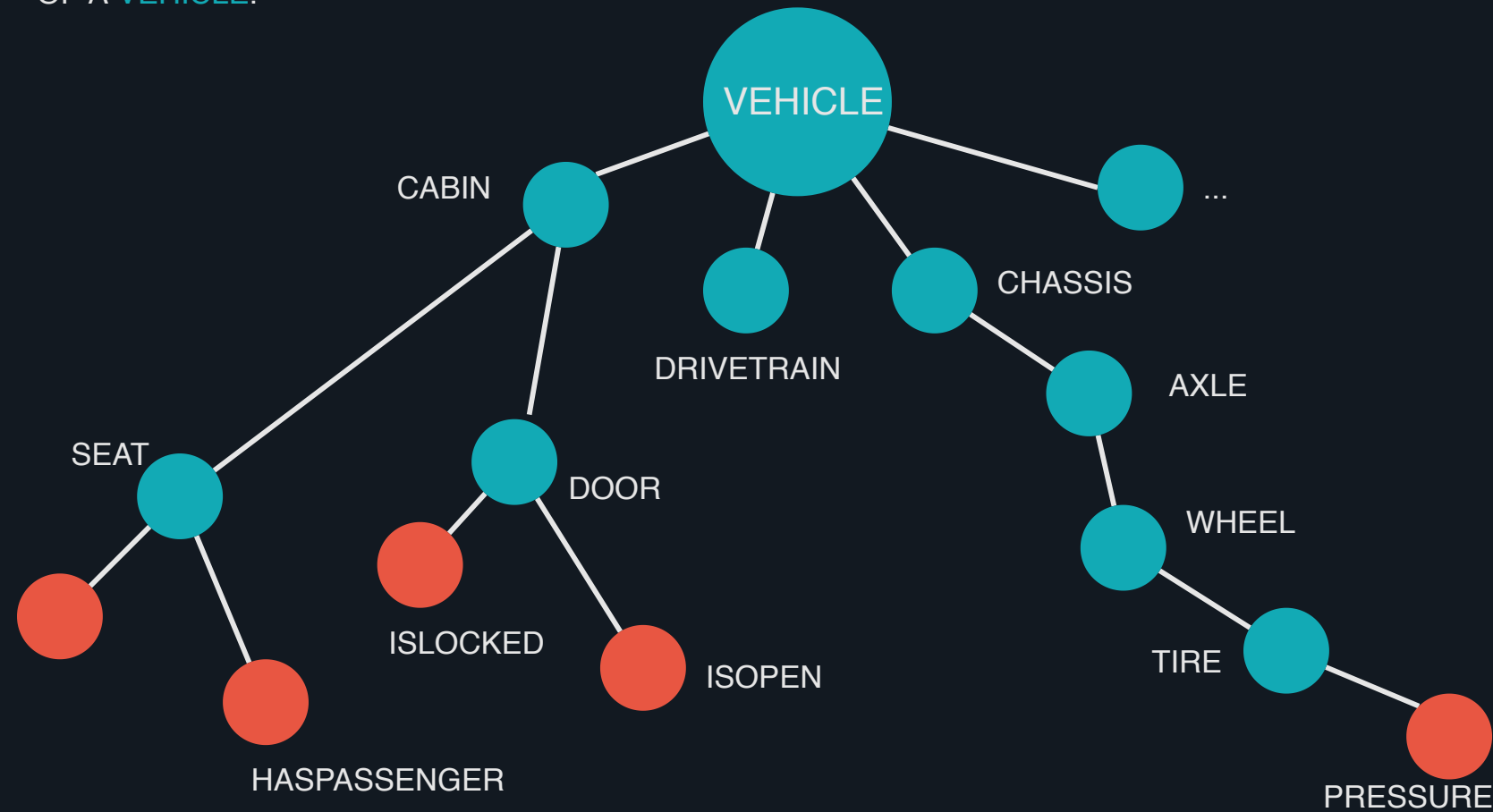
unit: km/h

description: The vehicle speed as measured by the drivetrain

VSS offers a simple, flexible and protocol agnostic way of describing vehicle signals.

VEHICLE SIGNAL SPECIFICATION (VSS)

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



YAML SPECIFICATION

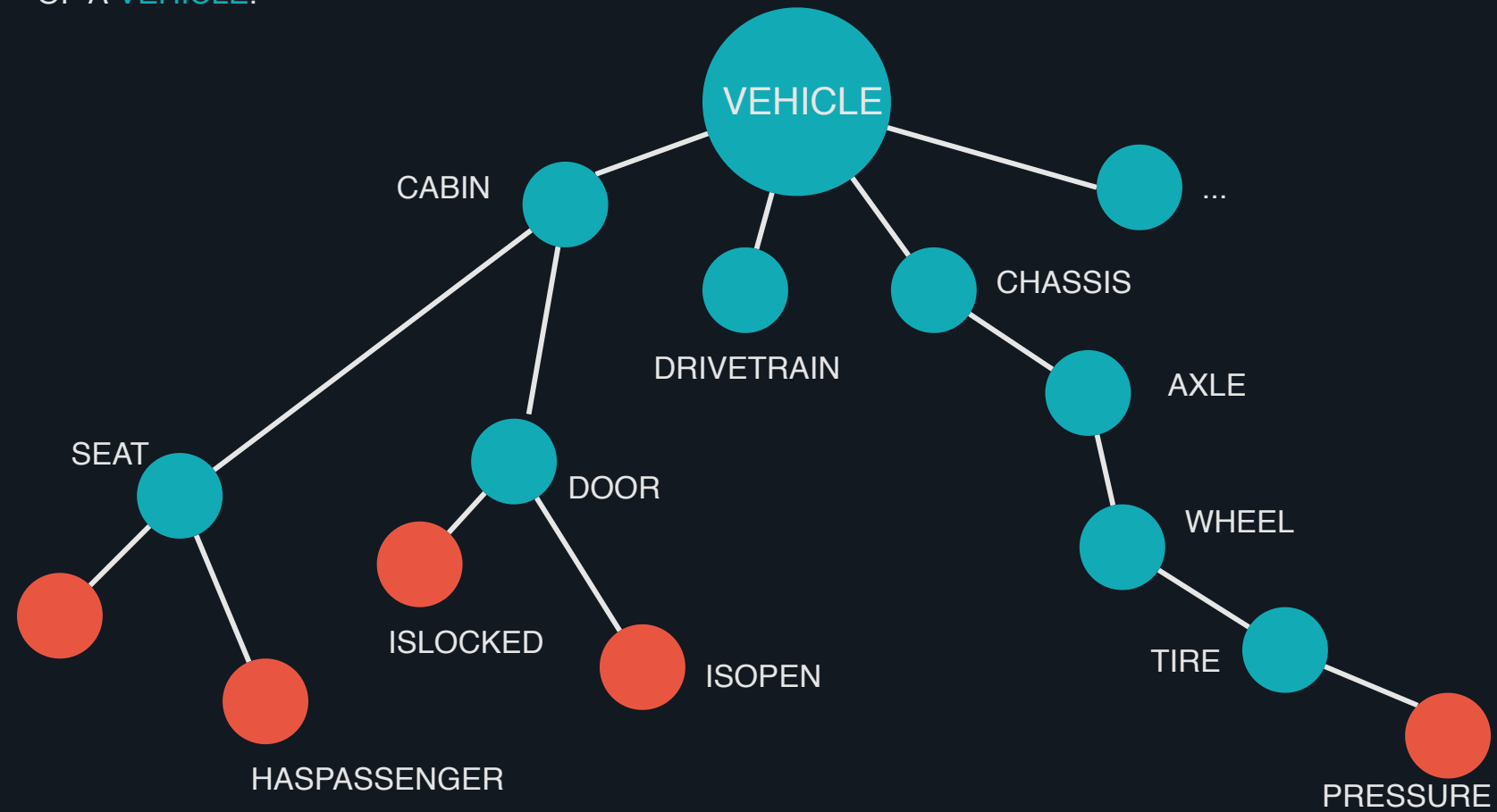
```
Vehicle.Drivetrain.Transmission.Speed  
type: sensor  
datatype: float  
unit: km/h
```

**description: The vehicle speed
as measured by the drivetrain**

VSS offers a simple, flexible and protocol agnostic way of describing vehicle signals.

VEHICLE SIGNAL SPECIFICATION (VSS)

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



YAML SPECIFICATION

*.csv

*.json

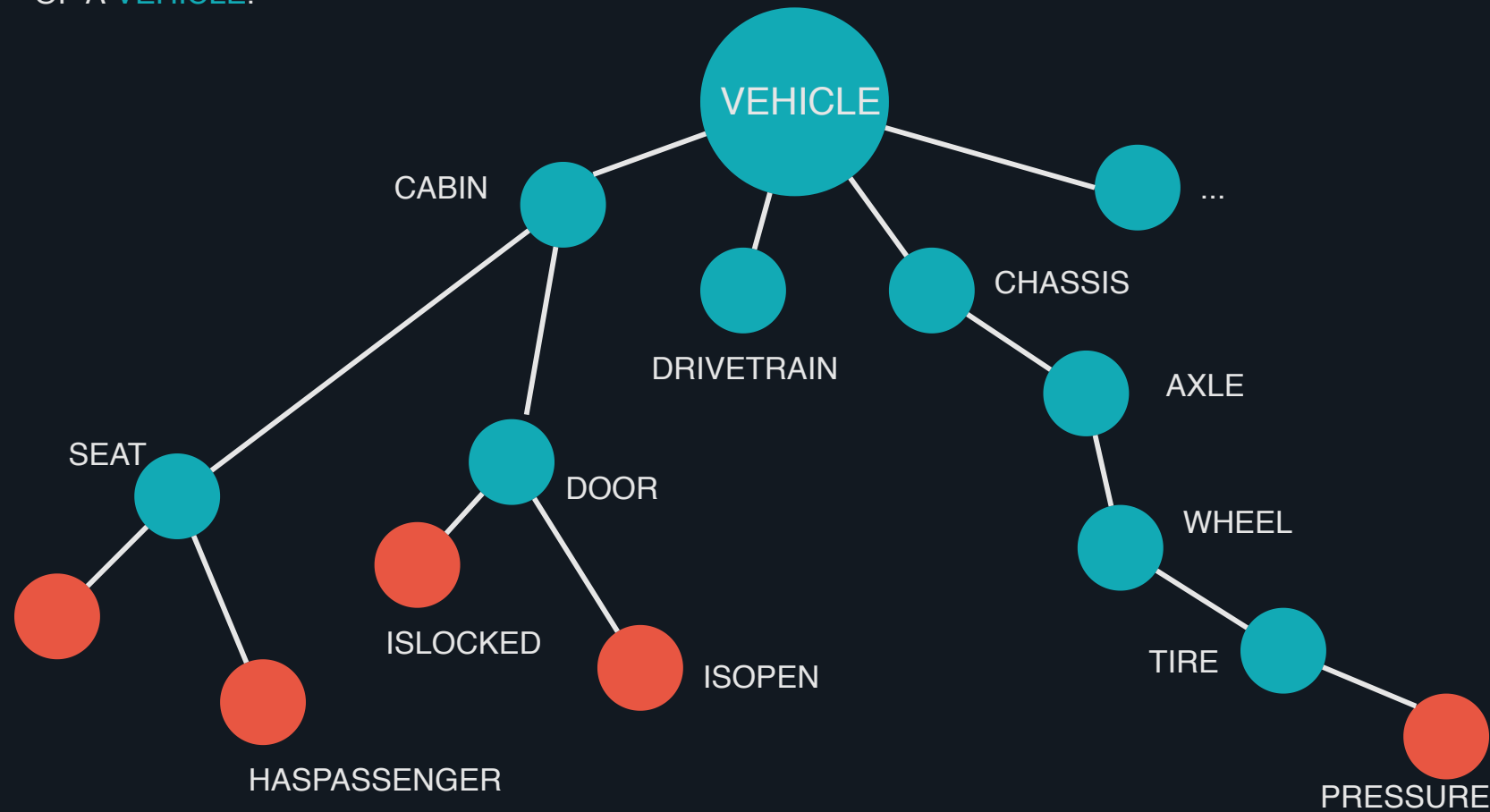
*.graphql

```
Vehicle.Drivetrain.Transmission.Speed
type: sensor
datatype: float
unit: km/h
description: The vehicle speed as measured by the drivetrain
```

VSS offers a simple, flexible and protocol agnostic way of describing vehicle signals.

VEHICLE SIGNAL SPECIFICATION (VSS)

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



VSS offers a simple, flexible and protocol agnostic way of describing vehicle signals.

YAML SPECIFICATION

*.csv

*.json

*.graphql

```
Vehicle.Drivetrain.Transmission.Speed
type: sensor
datatype: float
unit: km/h
description: The vehicle speed as measured by the drivetrain
```

What happened since last AMM?

Version 2.2

Besides various additions to the list of signals, this release focused on:

- Various updates and additions to the documentation
- Changed the signals in the spec files to valid YAML objects instead of list elements
- Moved to github actions for builds
- Changed naming from GENIVI to COVESA

Version 3.0 development

- Concept for layers
- Structural changes (enums, etc.)
- Cleanup and a lot of additions to signal tree

VSSo on W3C recommendation track

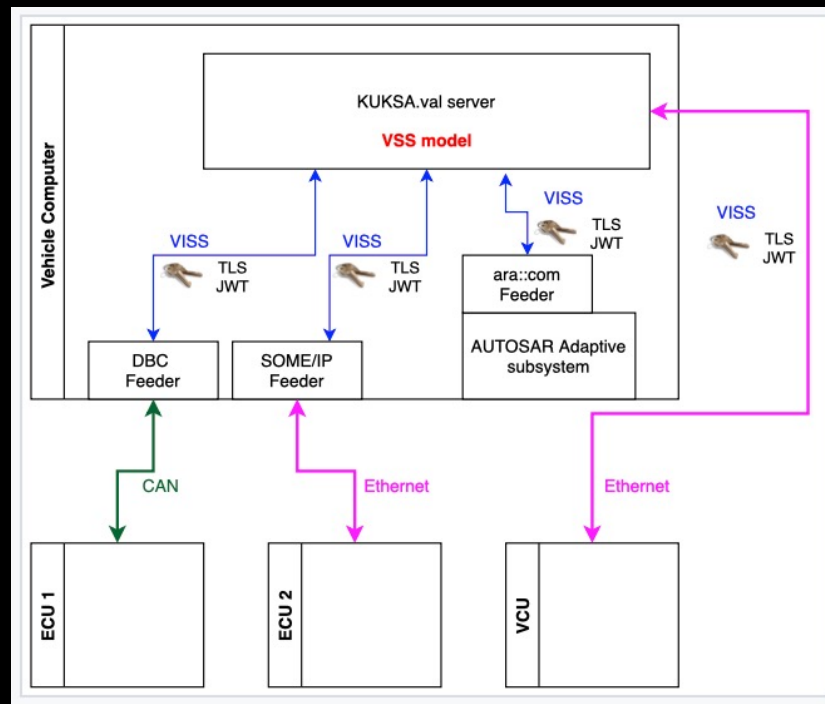
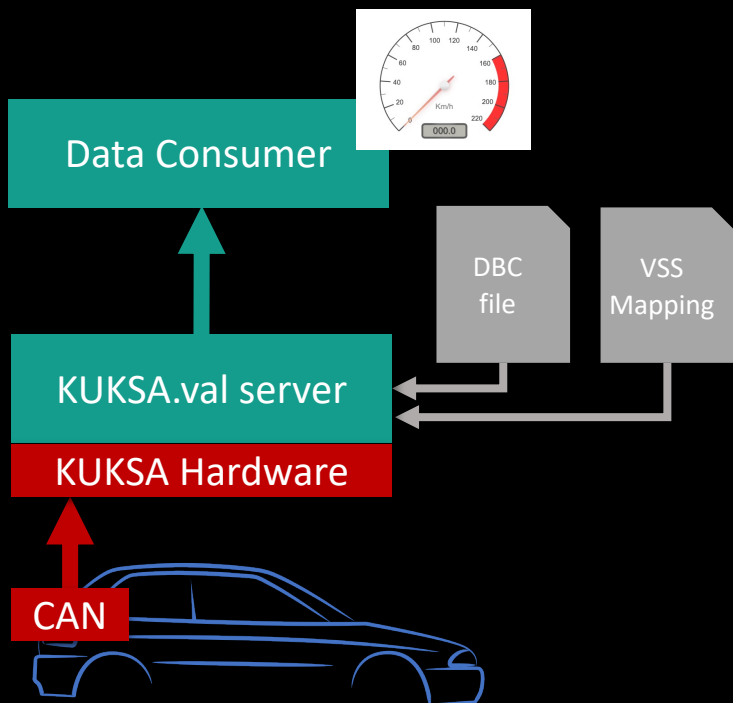
VEHICLE SIGNAL SPECIFICATION (VSS)

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.

But, where is it used?

VSS in Use: KUKSA.val

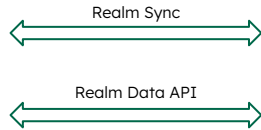
- In-vehicle “digital twin” of VSS data
- Written in C++, can run (containerized) on a Vehicle Computer
- App SDK for Python and GO available
- Example data feeders for CAN and GPS data available
- 100 % Open Source
- Secured access to data
- Applications access data using a version of the W3C VISS protocol



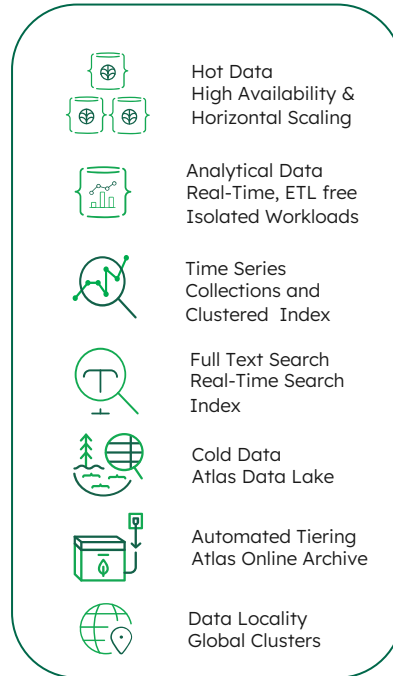
MongoDB Application Data Platform for Connected Vehicle



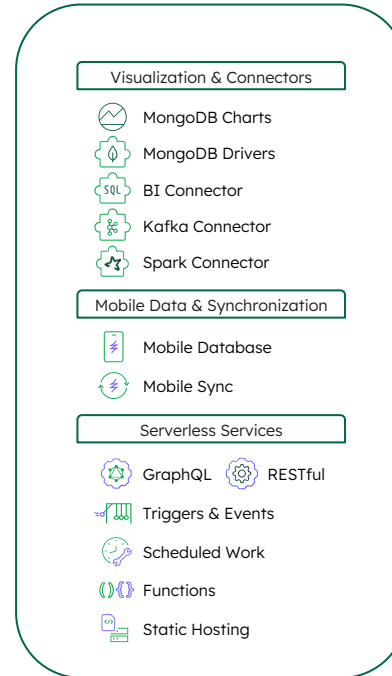
Connected Vehicle



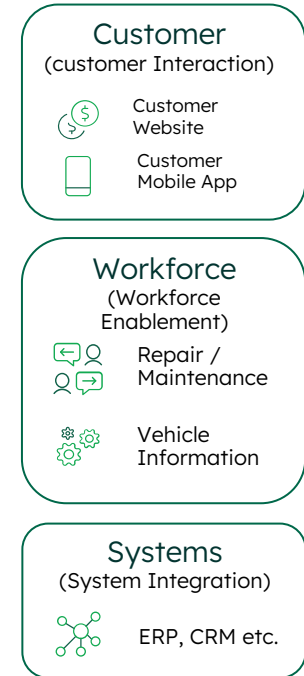
Cloud Data Platform



Integration Layer



Mobile / Web Clients





VSS @ Document Model

Onboard Realm DB(s)

- > VEHICLE
- > VEHICLE_ADAS
- ▼ VEHICLE_ADAS_ABS

```
export const Vehicle_ADAS_ABSSchema = {
  name: 'Vehicle_ADAS_ABS',
  embedded: true,
  properties: {
    Error: 'bool?',
    IsActive: 'bool?',
    IsEngaged: 'bool?',
  },
};
```

- ▼ VEHICLE_ADAS_CRUISECONTROL

```
export const Vehicle_ADAS_CruiseControlSchema = {
  name: 'Vehicle_ADAS_CruiseControl',
  embedded: true,
  properties: {
    Error: 'bool?',
    IsActive: 'bool?',
    SpeedSet: 'double?',
  },
};
```



Realm Sync:
 - Attribute level sync
 - Compressed
 - Deterministic
 - conflict resolution

MongoDB Data Platform



MongoDB Atlas

- Time Series
- Documents
- Search
- Graph
- Online Archive
- ...

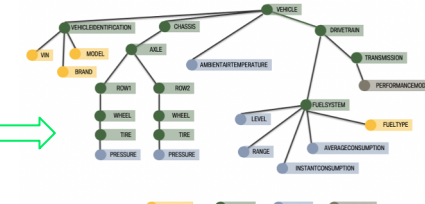


MongoDB Realm Serverless

- GraphQL
- MQL Data API
- Field lvl. RBAC
- ...



- ▼ ADAS: Object
 - ▼ ABS: Object
 - Error: false
 - IsActive: true
 - IsEngaged: false
 - ▼ CruiseControl: Object
 - Error: false
 - IsActive: true
 - SpeedSet: 0
 - ▼ ESC: Object
 - ▼ LaneDepartureDetection: Object
 - ▼ ObstacleDetection: Object
 - ▼ TCS: Object
 - ▼ Acceleration: Object
 - ▼ AmbientAirTemperature: 0
 - ▼ AngularVelocity: Object
 - ▼ AverageSpeed: 0
 - ▼ Body: Object
 - ▼ Cabin: Object
 - ▼ Chassis: Object
 - ▼ CurbWeight: 0
 - ▼ CurrentLocation: Object
 - ▼ CurrentOverallWeight: 0
 - ▼ DriveTime: 0
 - ▼ Driver: Object



```
> Drivetrain.Transmission.Speed:
  type: sensor
  attributes: adasabs
  unit: km/h
  min: 0
  max: 160
  description: The vehicle speed, as measured by the drivetrain.
```

Your Data Ecosystem

Mobile SDKs

- > VEHICLE
- > VEHICLE_ADAS
- ▼ VEHICLE_ADAS_ABS

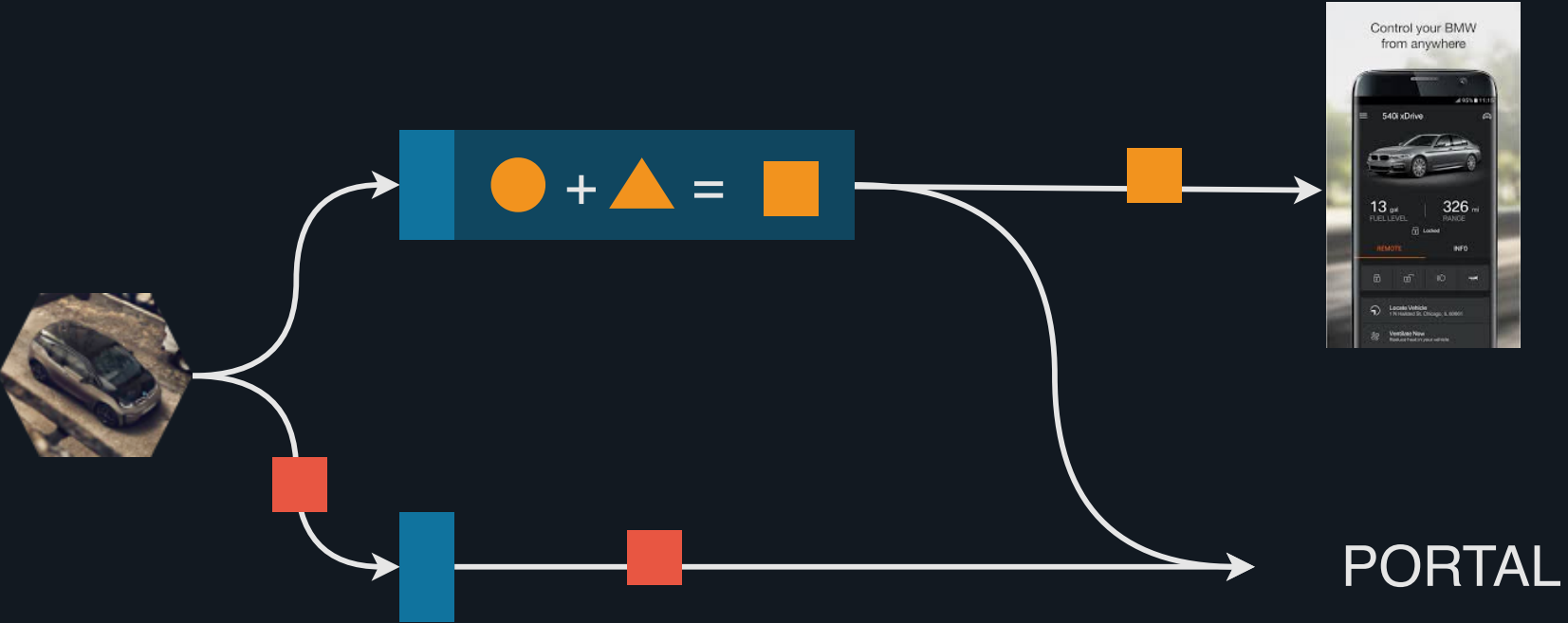
```
export const Vehicle_ADAS_ABSSchema = {
  name: 'Vehicle_ADAS_ABS',
  embedded: true,
  properties: {
    Error: 'bool?',
    IsActive: 'bool?',
    IsEngaged: 'bool?',
  },
};
```

GraphQL Endpoint

```
GraphQL
1 query {
2   vehicle {
3     adas {
4       abs {
5         Error
6         IsActive
7         IsEngaged
8       }
9       CruiseControl {
10        Error
11        IsActive
12        SpeedSet
13      }
14    }
15    Cabin {
16      Infotainment (HMI {
17        ..
18      }
19    }
20  }
21 }
```

VSS IN VEHICLE SHADOW

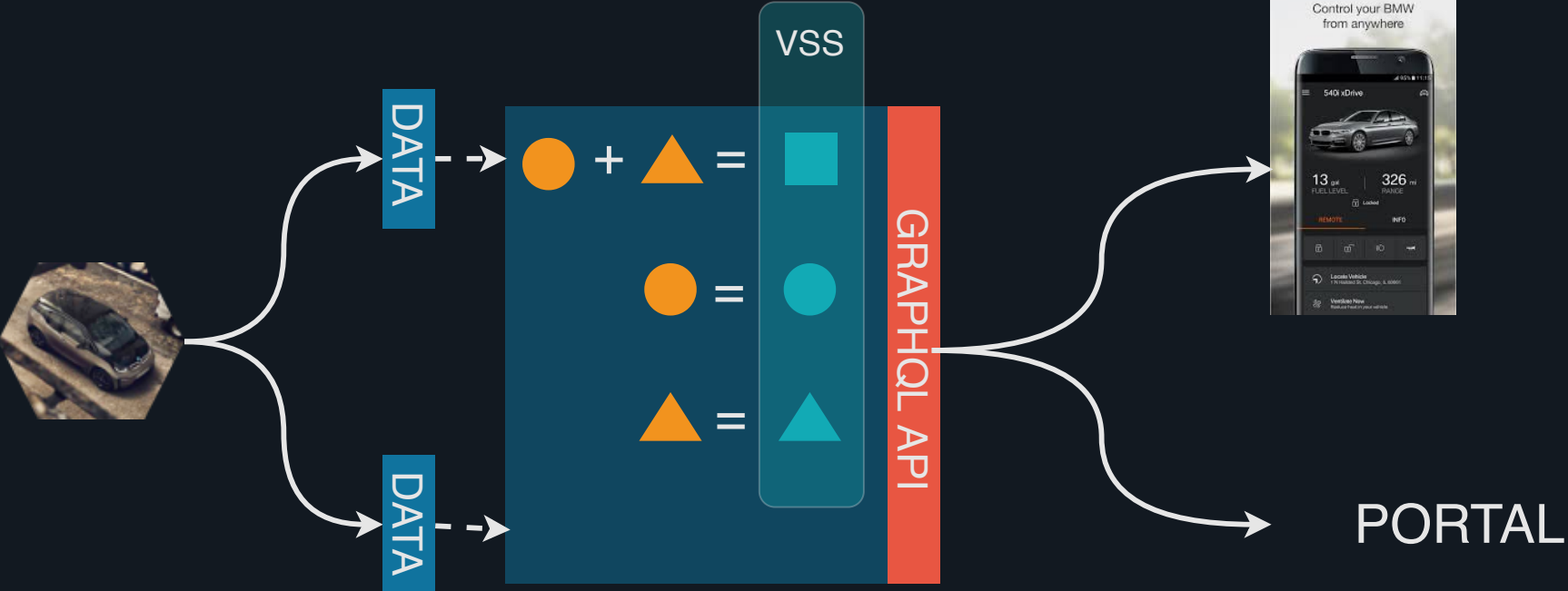
TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



- Unclear master system
- Mixed data models
- Multiple rule-based computations
- New services build new adapters
- Privacy hard to manage

VSS IN VEHICLE SHADOW

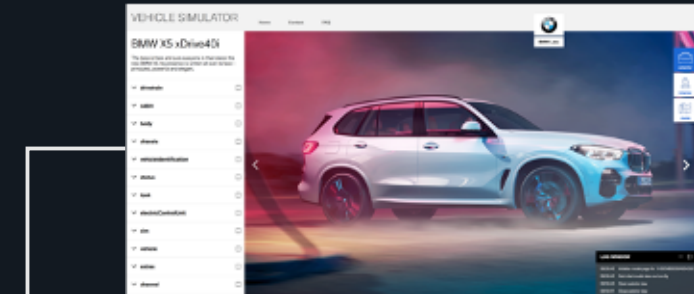
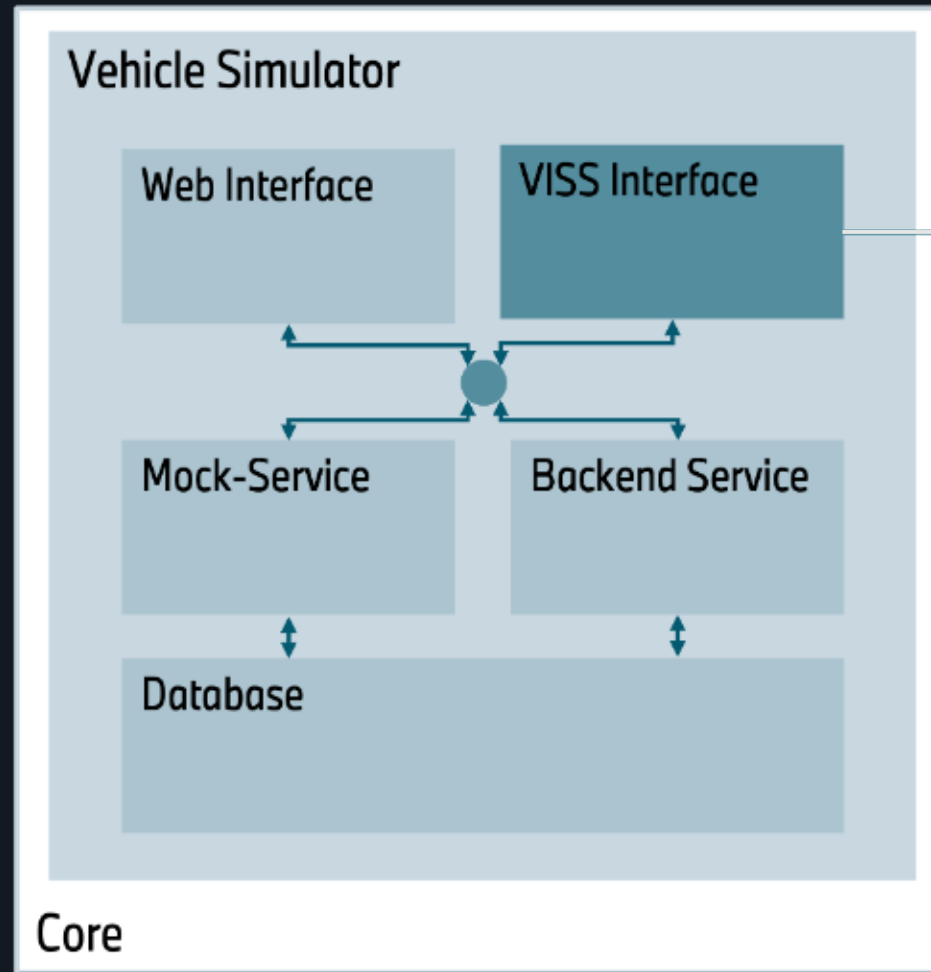
TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



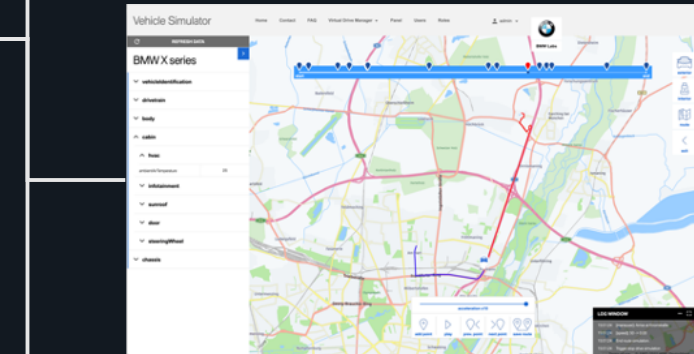
Unified Data Model
Single backend API
Improved privacy

VSS VEHICLE DATA SIMULATOR

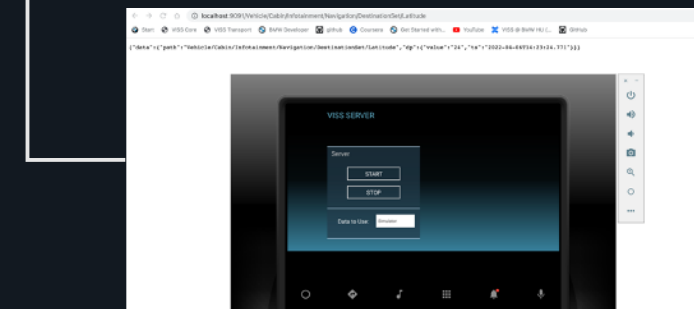
TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



static



dynamic



on-board

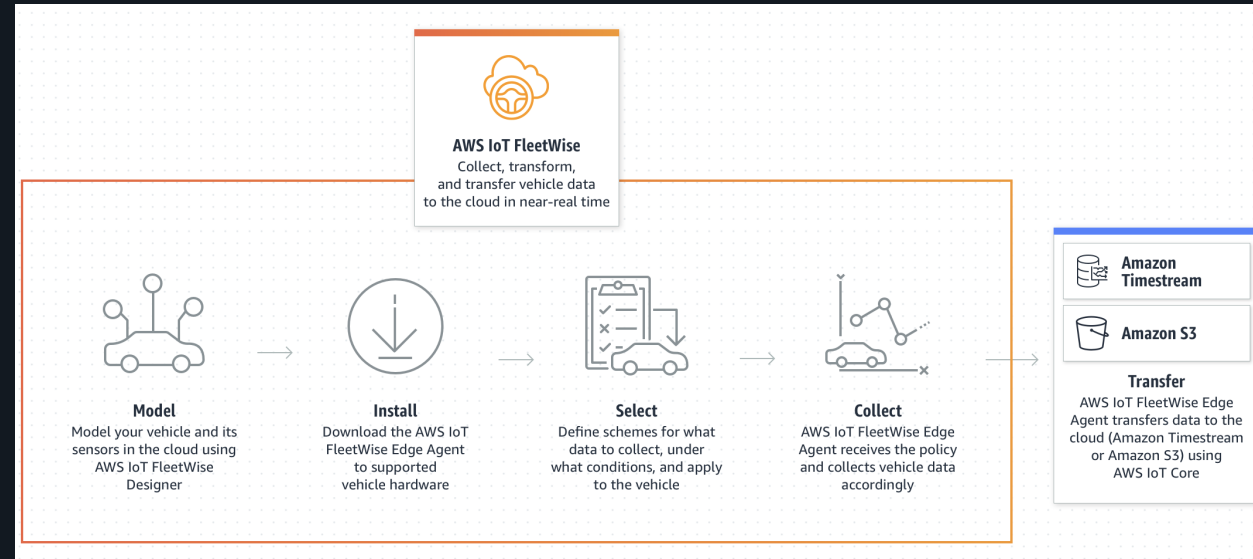


THURSDAY, 11 am

CVII Working Session - VISS
and W3C Alignment

VEHICLE SIGNAL SPECIFICATION (VSS)

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



Delivering the next generation user experience,
Through an open, standardized platform
And our Mutual Ecosystem of partners:



VISS version 2 - Core

W3C Working Draft 05 April 2022



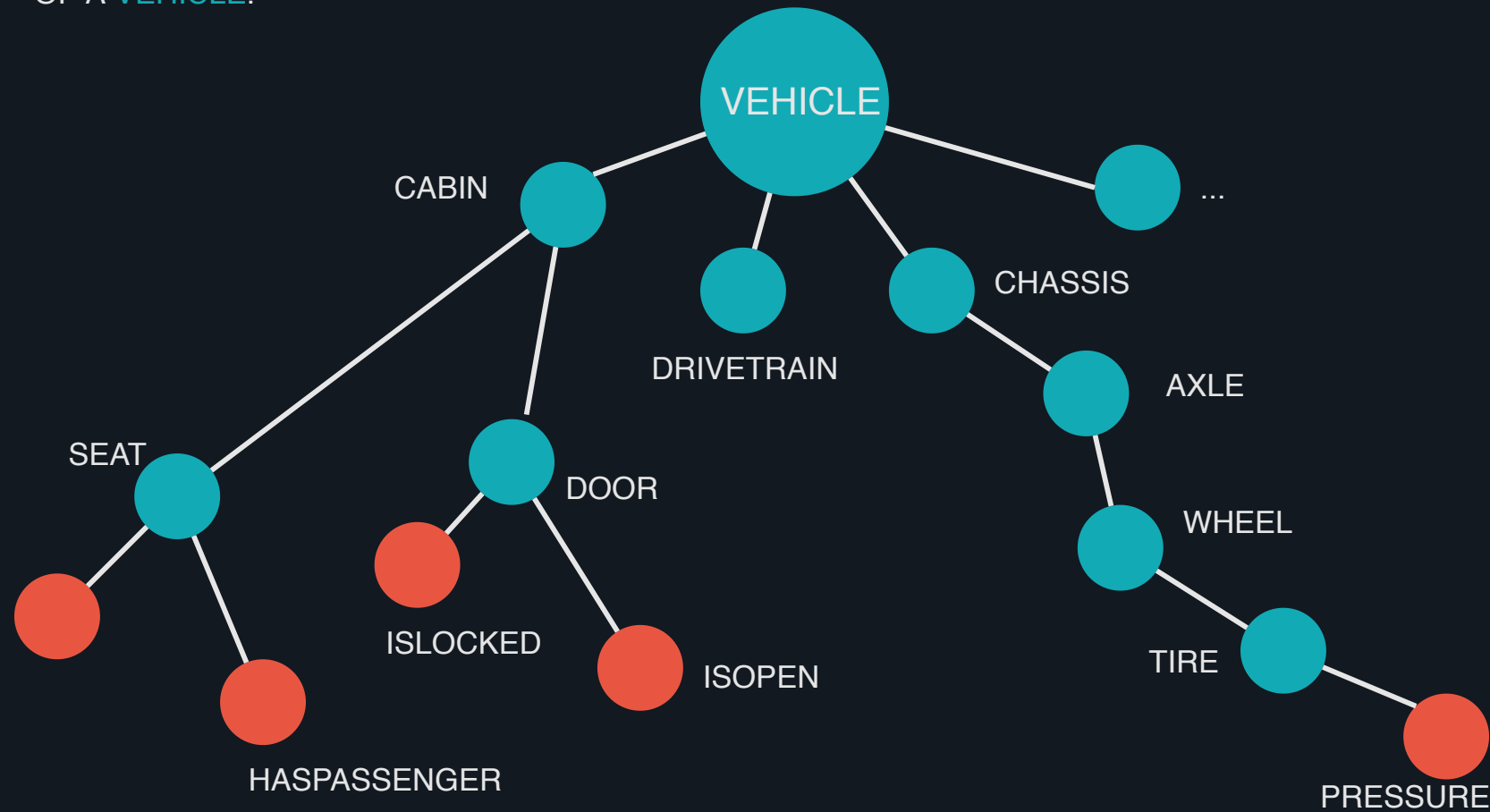
VEHICLE SIGNAL SPECIFICATION (VSS)

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.

**Great! But where are
the limits?**

VEHICLE SIGNAL SPECIFICATION (VSS)

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



VSS offers a simple, flexible and protocol agnostic way of describing vehicle signals.

YAML SPECIFICATION

*.csv

*.json

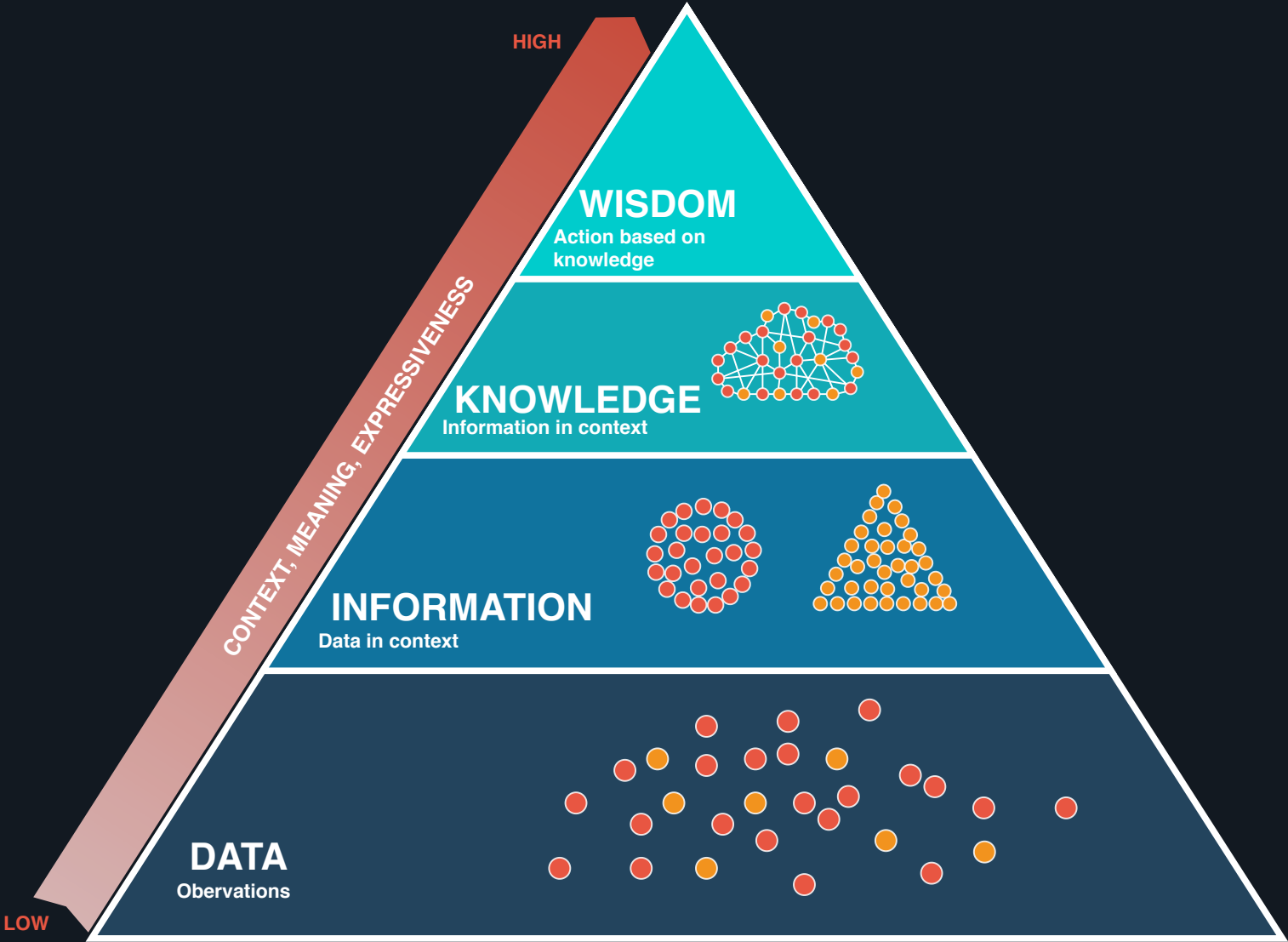
*.graphql

```
Vehicle.Drivetrain.Transmission.Speed
type: sensor
datatype: float
unit: km/h
description: The vehicle speed as measured by the drivetrain
```

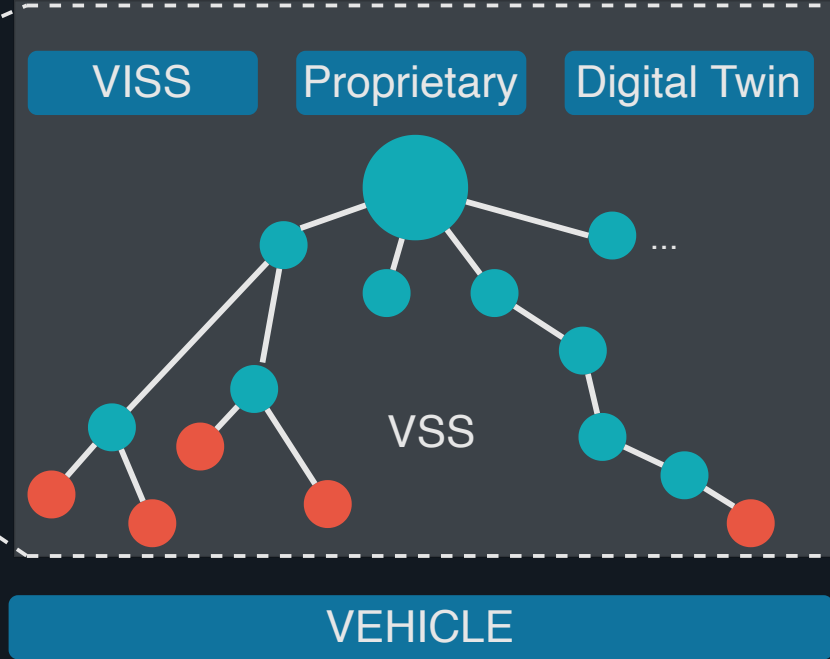
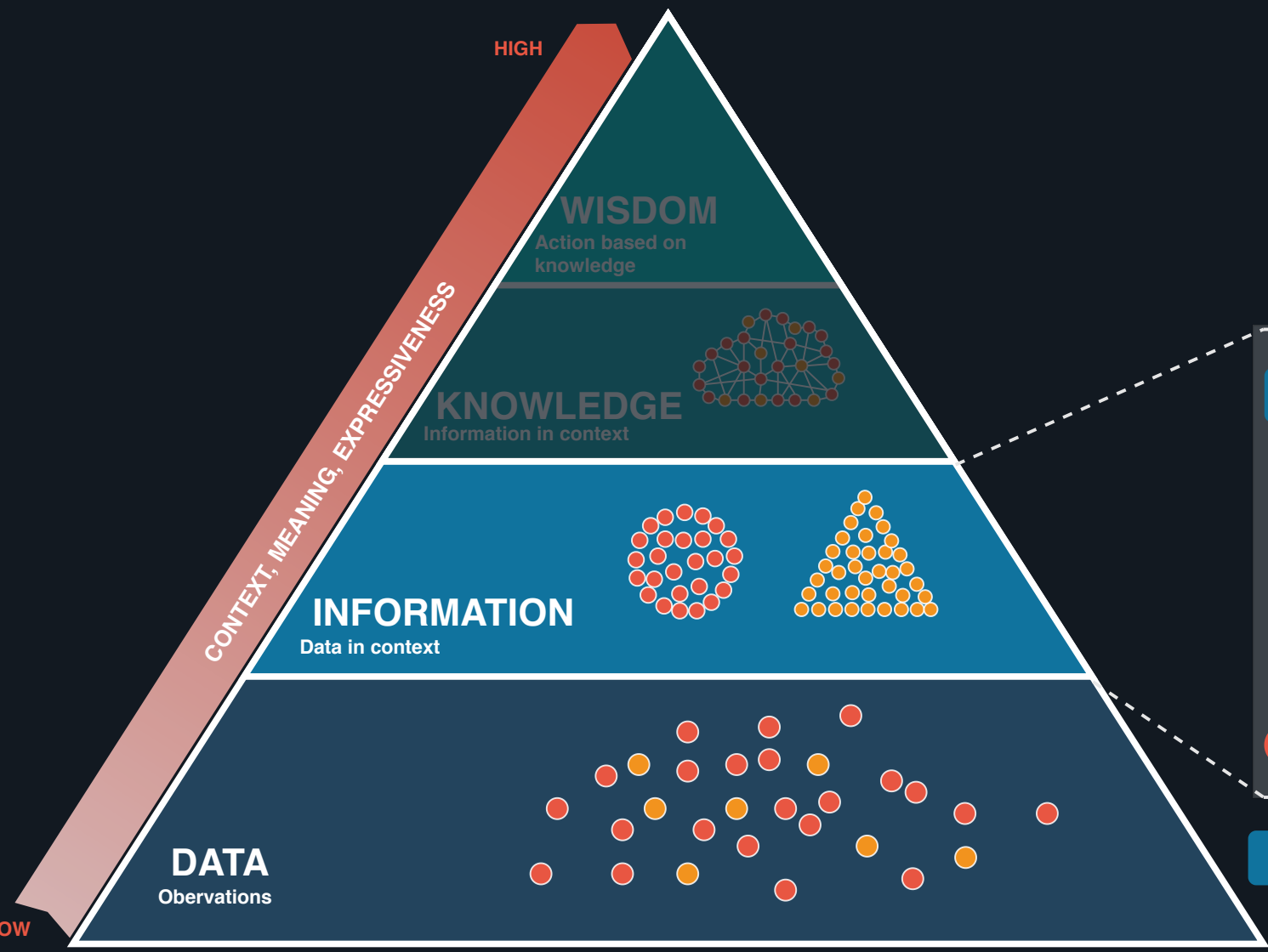
YAML SPECIFICATION PROs & CONS

- + Easy to read, parse and understand.
- + Tooling available and useable beyond vehicle signals.
- + Only text, well maintainable in common development tools and version management.
- But, limited modelling capabilities with regard to relationships.
- Hard to refer from one domain to another.

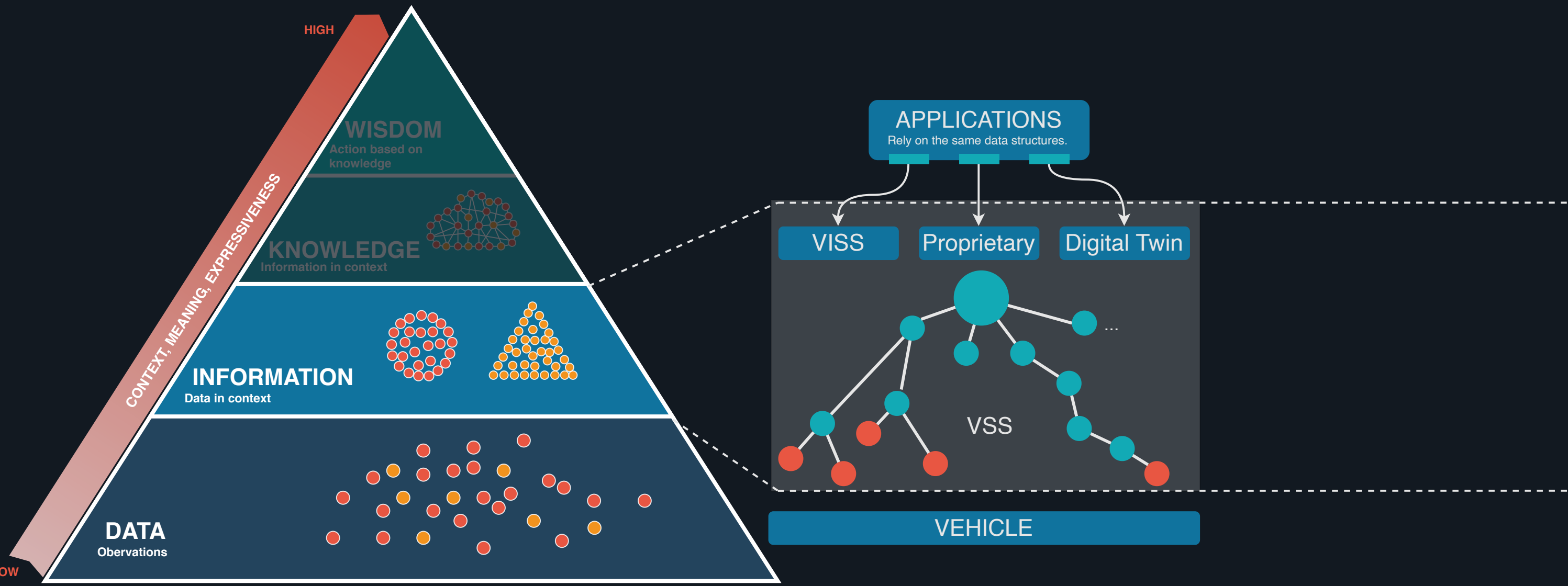
PRACTICAL ISSUES USING YAML IN MULTI-DOMAIN SCENARIOS.



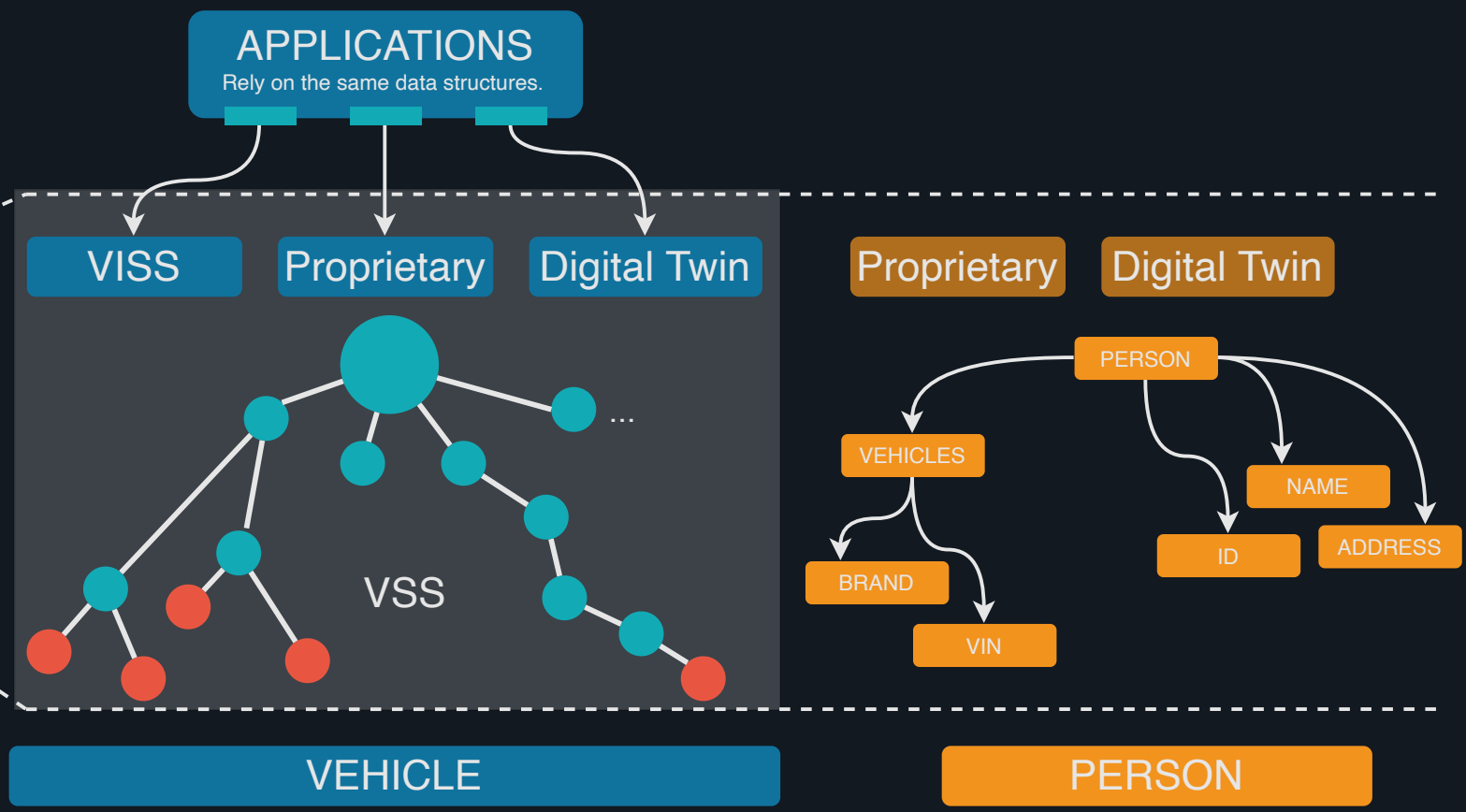
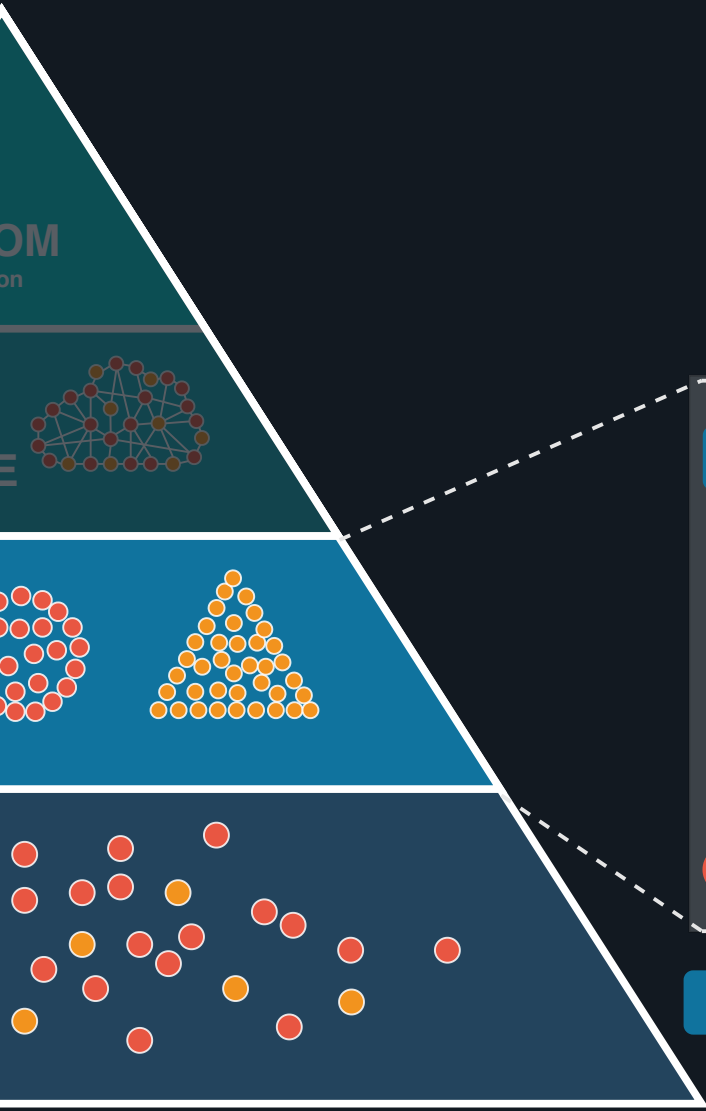
PRACTICAL ISSUES USING YAML IN MULTI-DOMAIN SCENARIOS.



PRACTICAL ISSUES USING YAML IN MULTI-DOMAIN SCENARIOS.

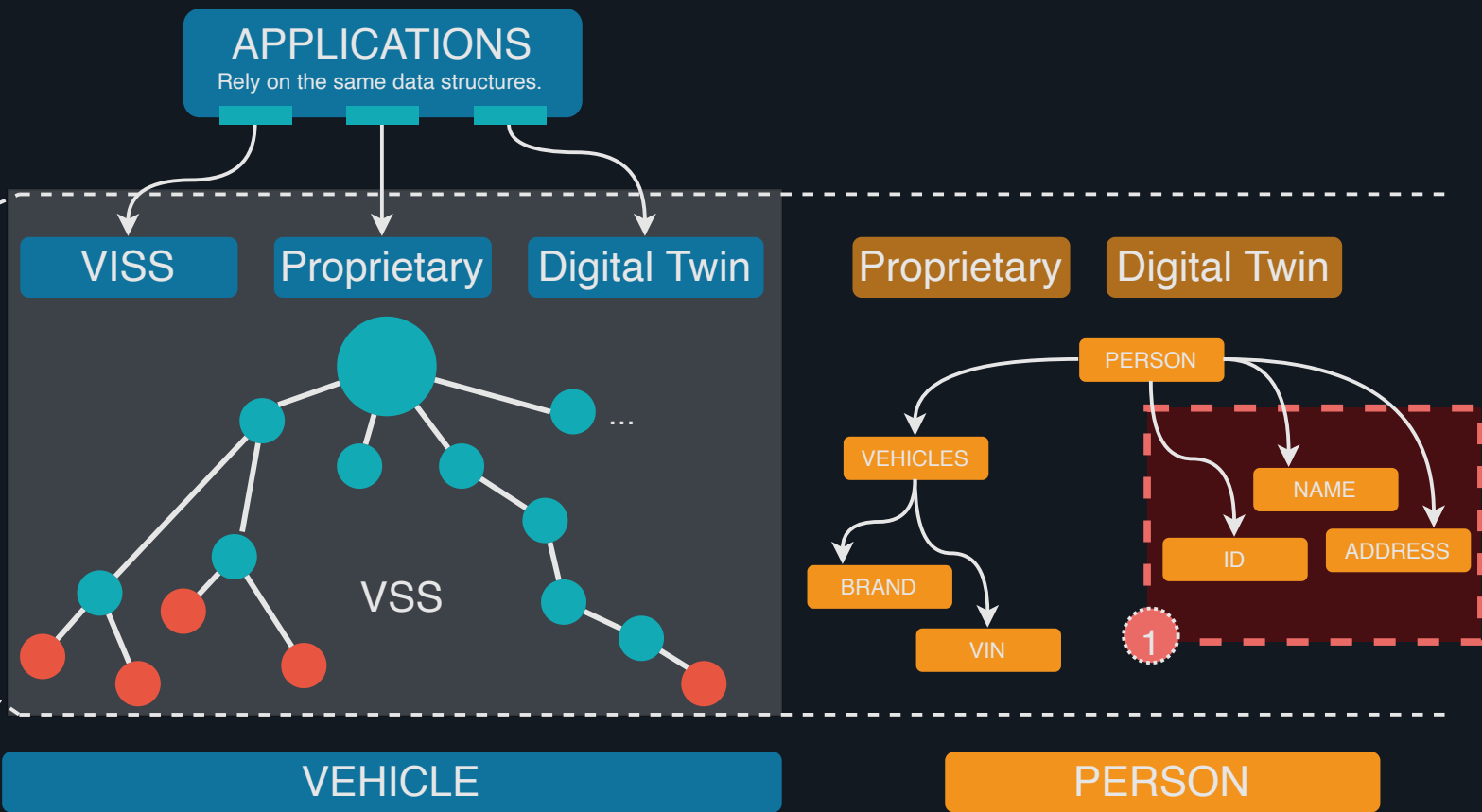
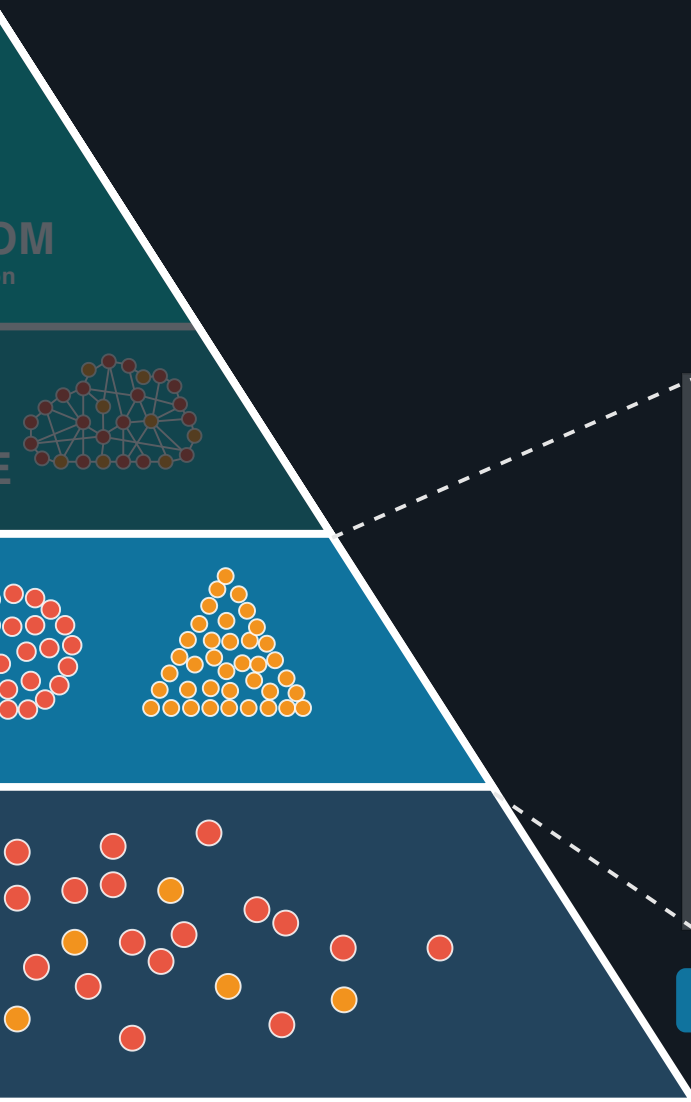


PRACTICAL ISSUES USING YAML IN MULTI-DOMAIN SCENARIOS.



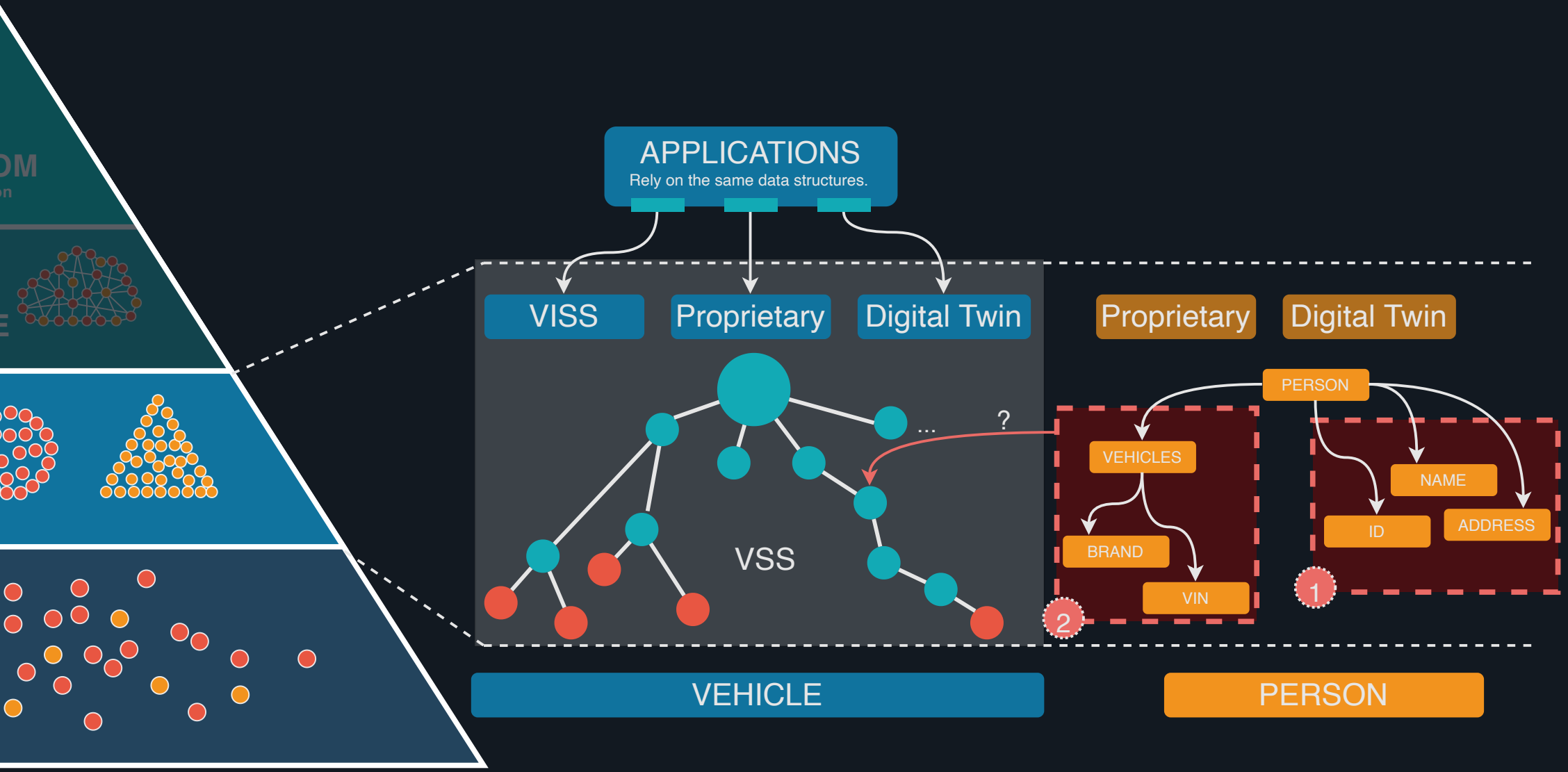
PRACTICAL ISSUES USING YAML IN MULTI-DOMAIN SCENARIOS.

1 One type of relation not enough in many other domains.



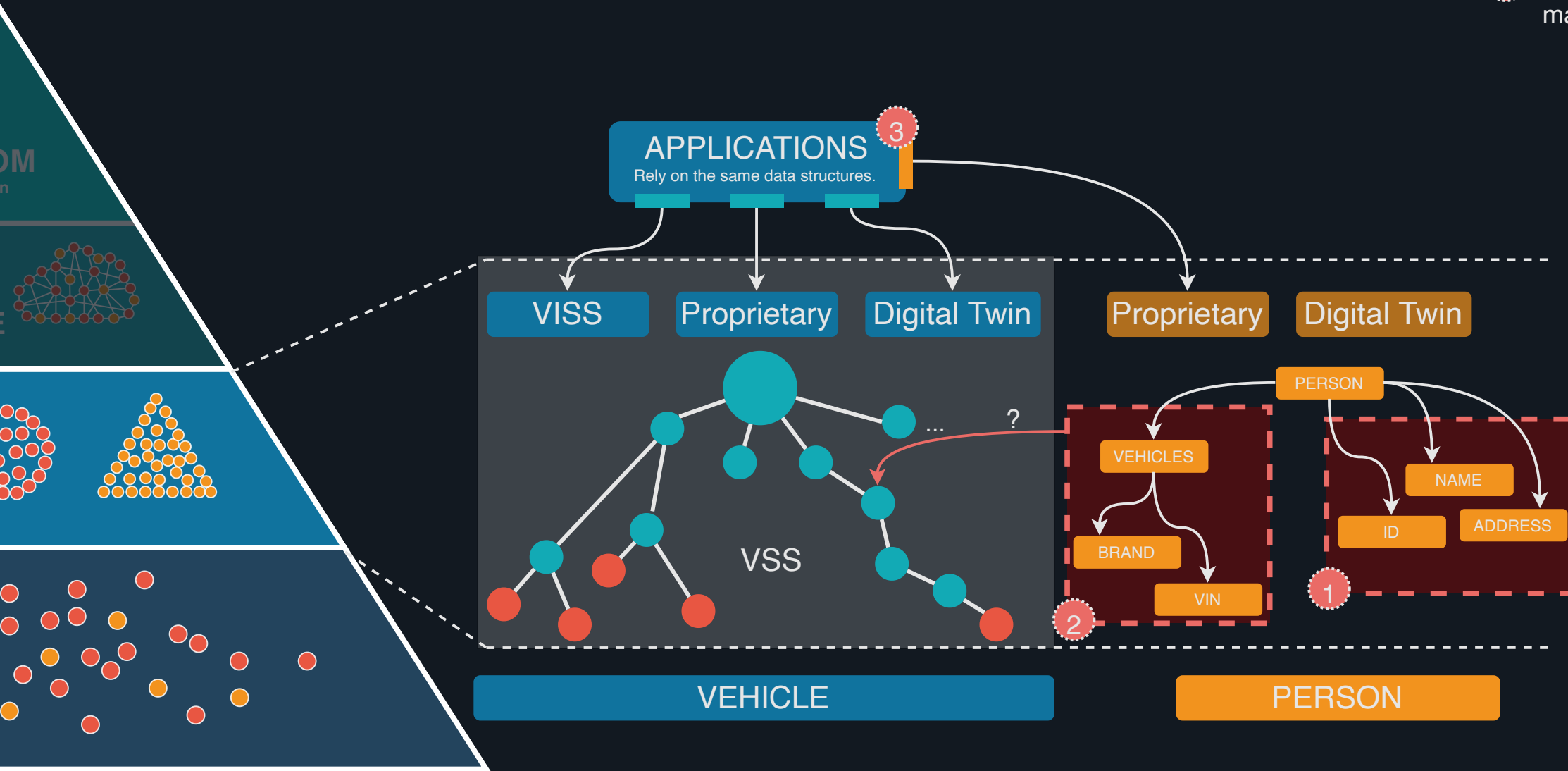
PRACTICAL ISSUES USING YAML IN MULTI-DOMAIN SCENARIOS.

- 1 One type of relation not enough in many other domains.
- 2 Relation to other domains with same technology stack hard to realise. Reference on existing standards not trivial (e.g. schema.org, FOAF)



PRACTICAL ISSUES USING YAML IN MULTI-DOMAIN SCENARIOS.

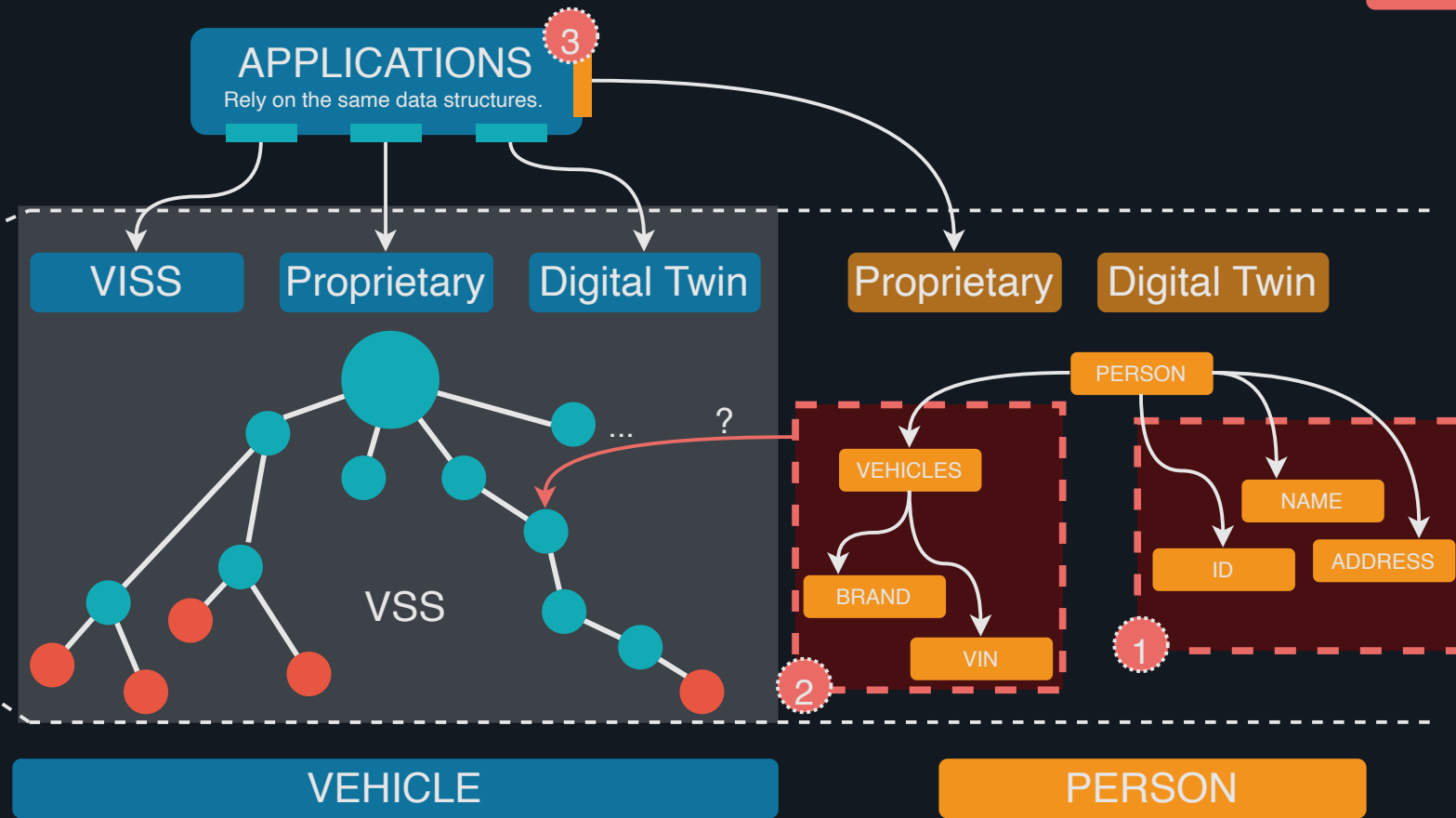
- 1 One type of relation not enough in many other domains.
- 2 Relation to other domains with same technology stack hard to realise. Reference on existing standards not trivial (e.g. schema.org, FOAF)
- 3 Manual work of integration and mapping.



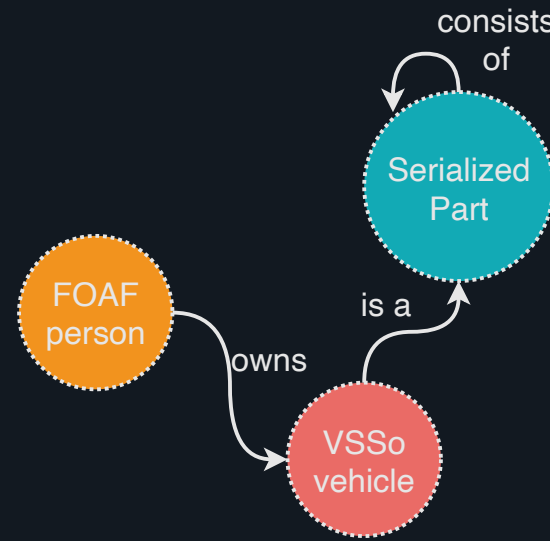
PRACTICAL ISSUES USING YAML IN MULTI-DOMAIN SCENARIOS.

- 1 One type of relation not enough in many other domains.
- 2 Relation to other domains with same technology stack hard to realise. Reference on existing standards not trivial (e.g. schema.org, FOAF)
- 3 Manual work of integration and mapping.

There are existing standards of the W3C tackling those issues.

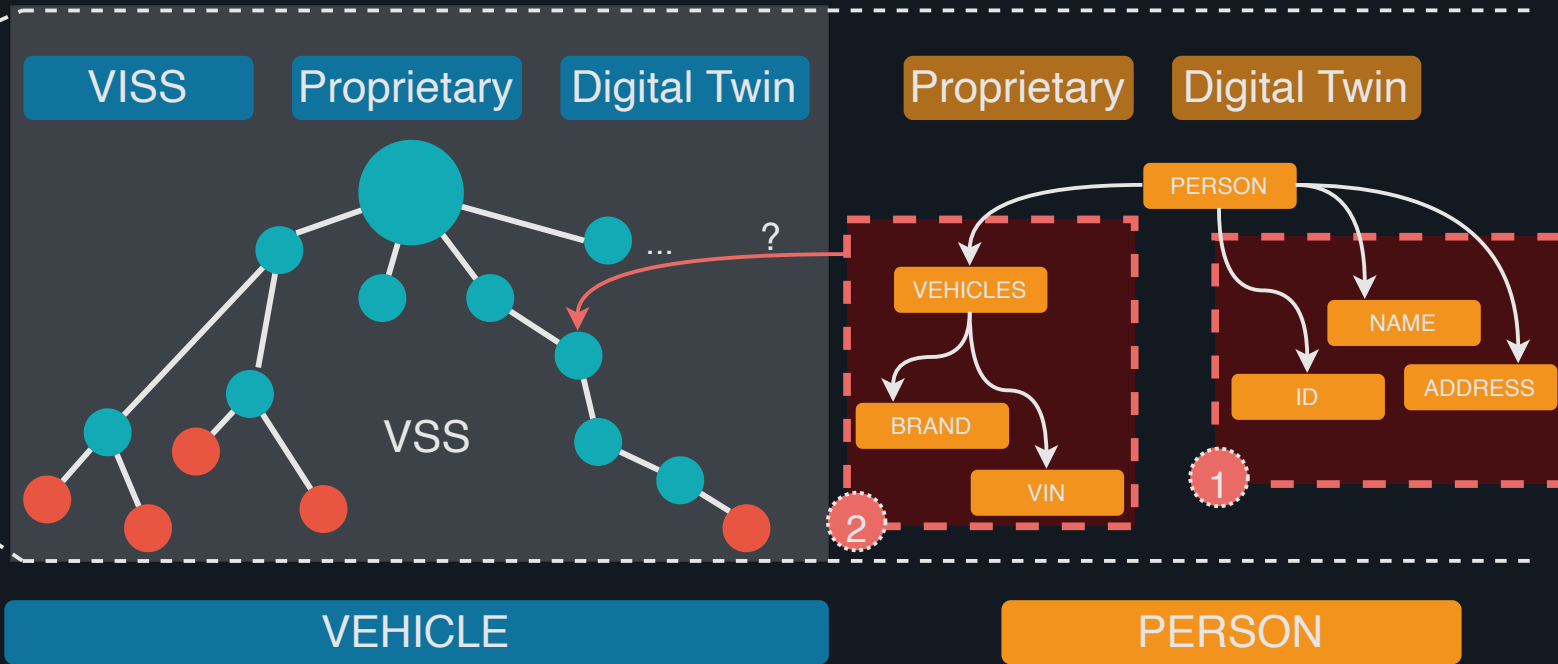
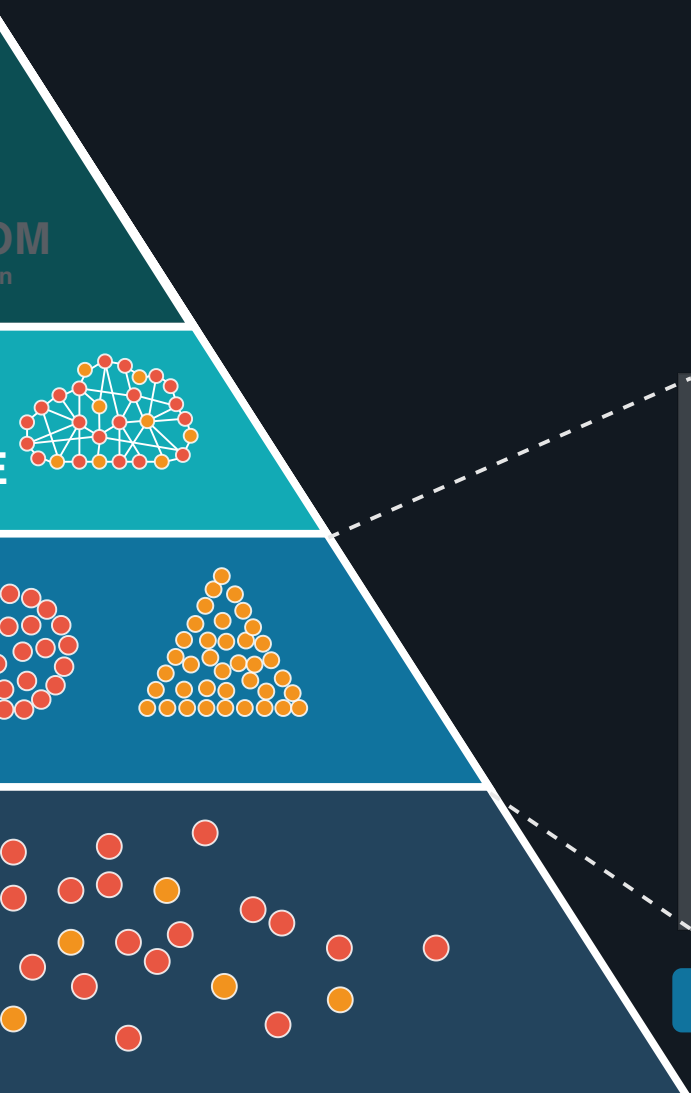


PRACTICAL ISSUES USING YAML IN MULTI-DOMAIN SCENARIOS.

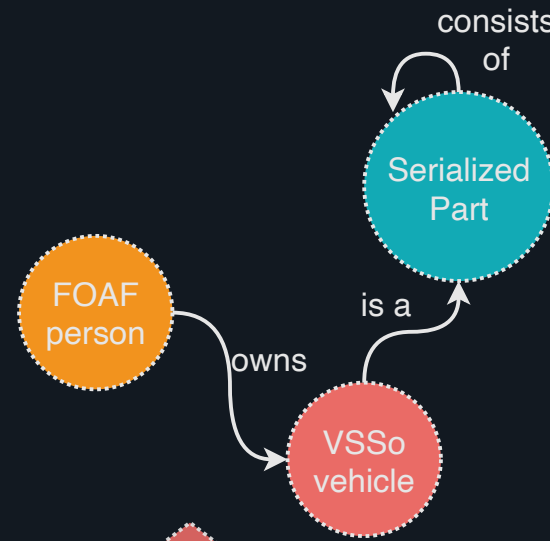
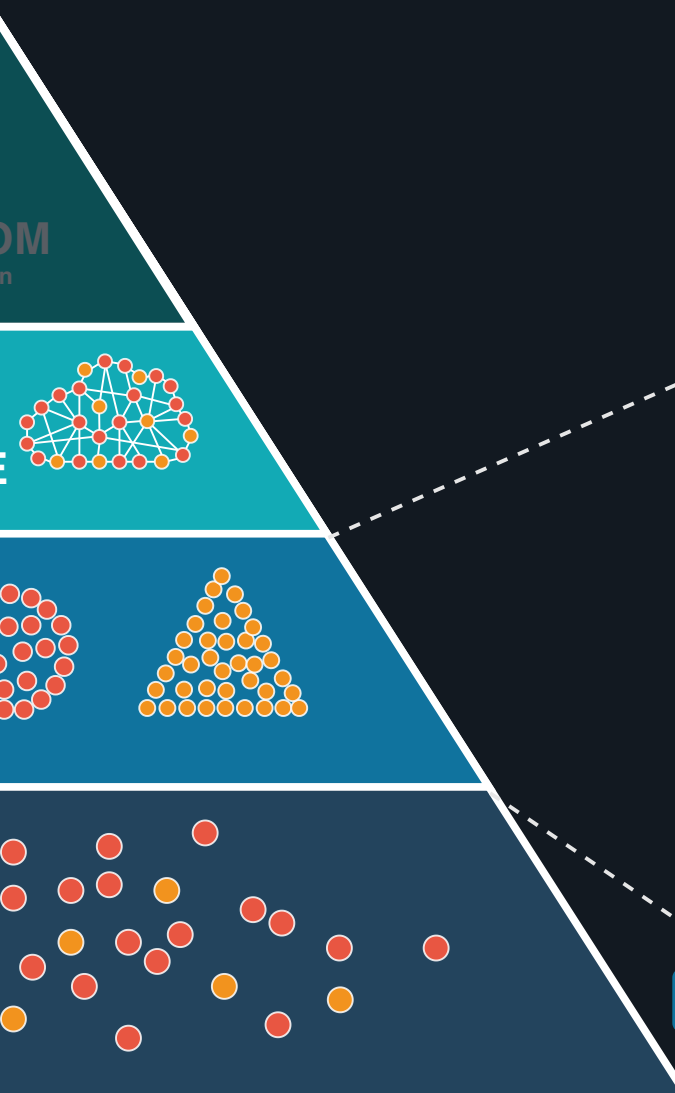


- 1 One type of relation not enough in many other domains.
- 2 Relation to other domains with same technology stack hard to realise. Reference on existing standards not trivial (e.g. schema.org, FOAF)
- 3 Manual work of integration and mapping.

There are existing standards of the W3C tackling those issues.

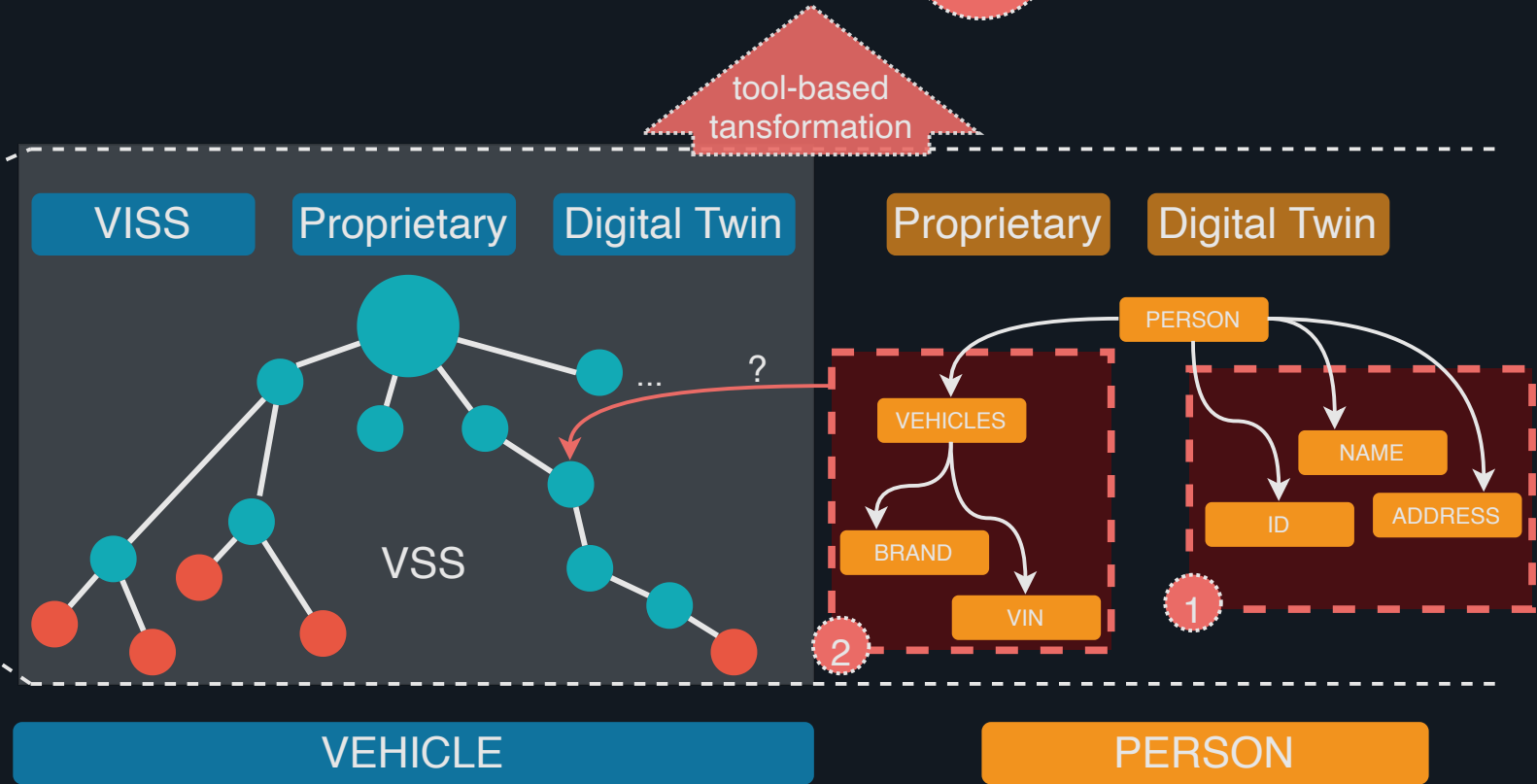


PRACTICAL ISSUES USING YAML IN MULTI-DOMAIN SCENARIOS.

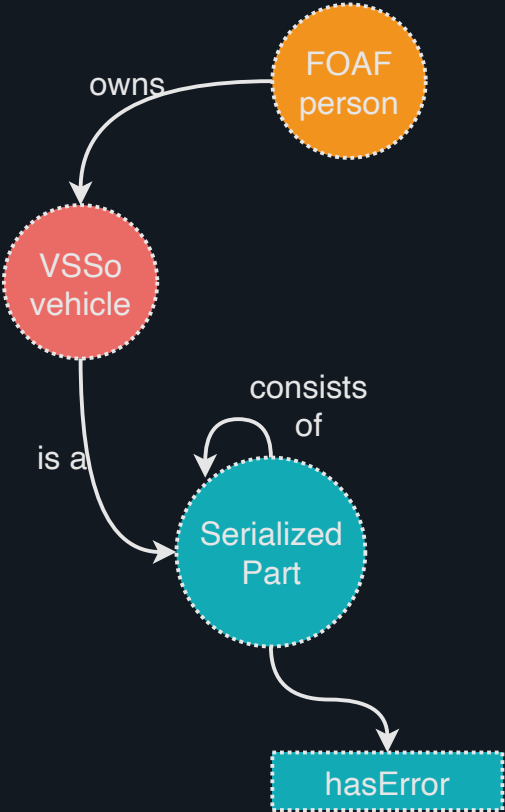


- 1 One type of relation not enough in many other domains.
- 2 Relation to other domains with same technology stack hard to realise. Reference on existing standards not trivial (e.g. schema.org, FOAF)
- 3 Manual work of integration and mapping.

There are existing standards of the W3C tackling those issues.

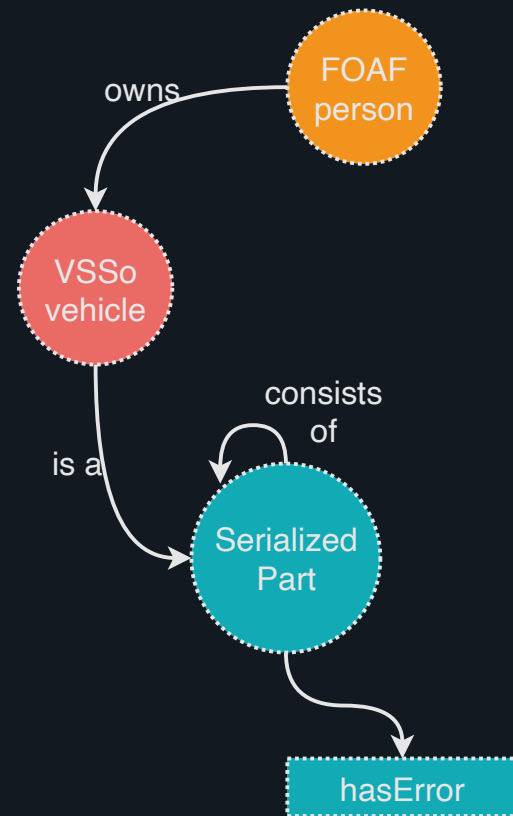


ONTOLOGY USE CASE



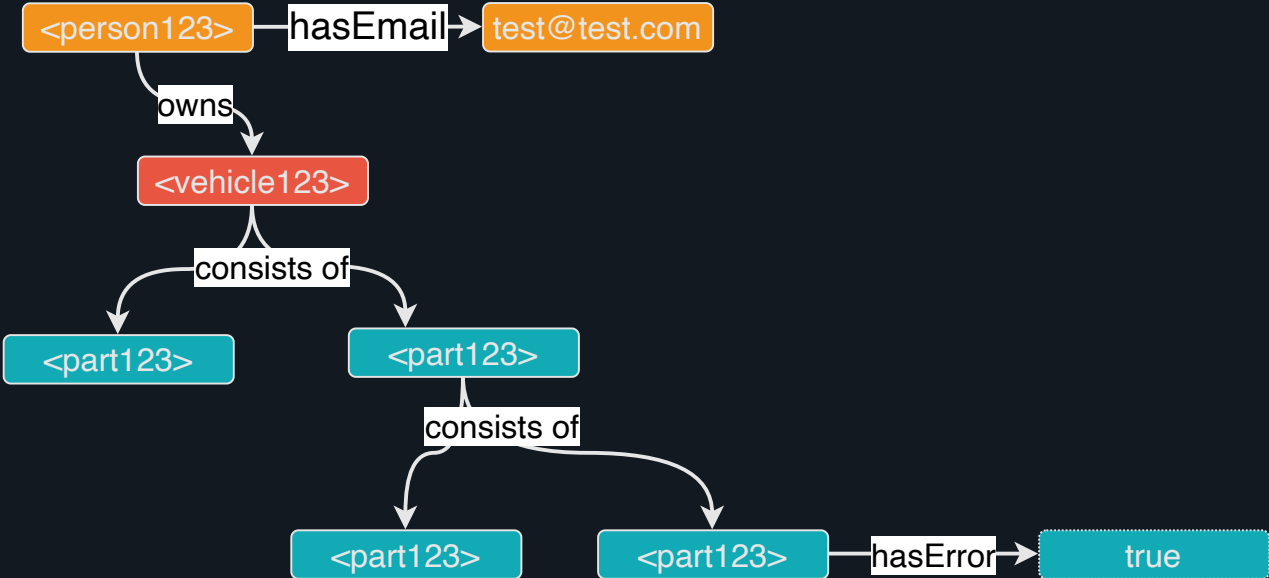
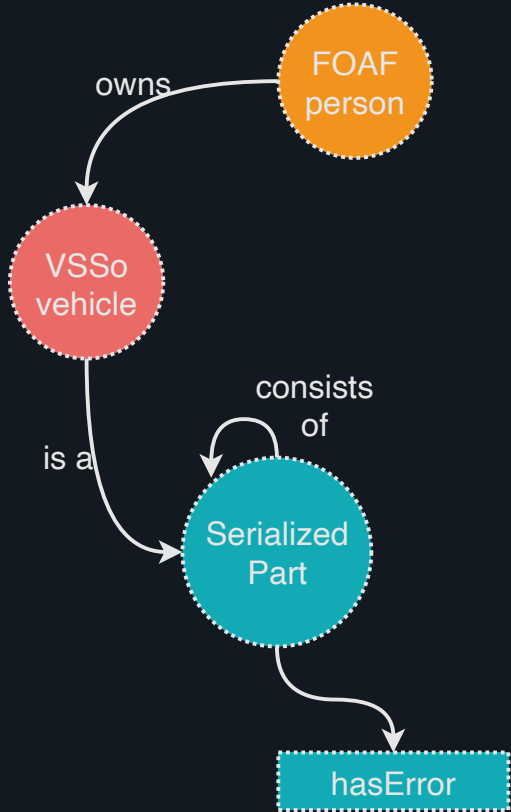
If a part of a vehicle has an issue, who is the customer to inform?

ONTOLOGY USE CASE



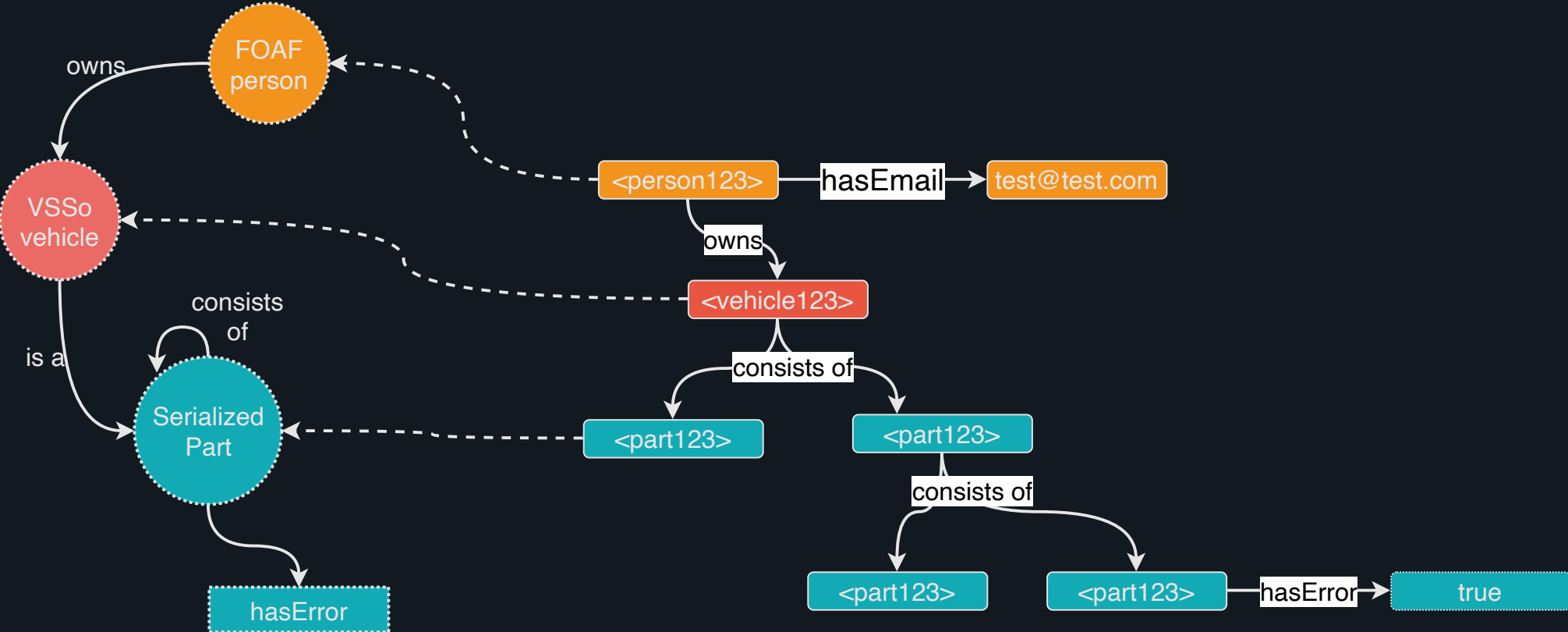
If a **part** of a **vehicle** has an issue, who is the **customer** to inform?

ONTOLOGY USE CASE



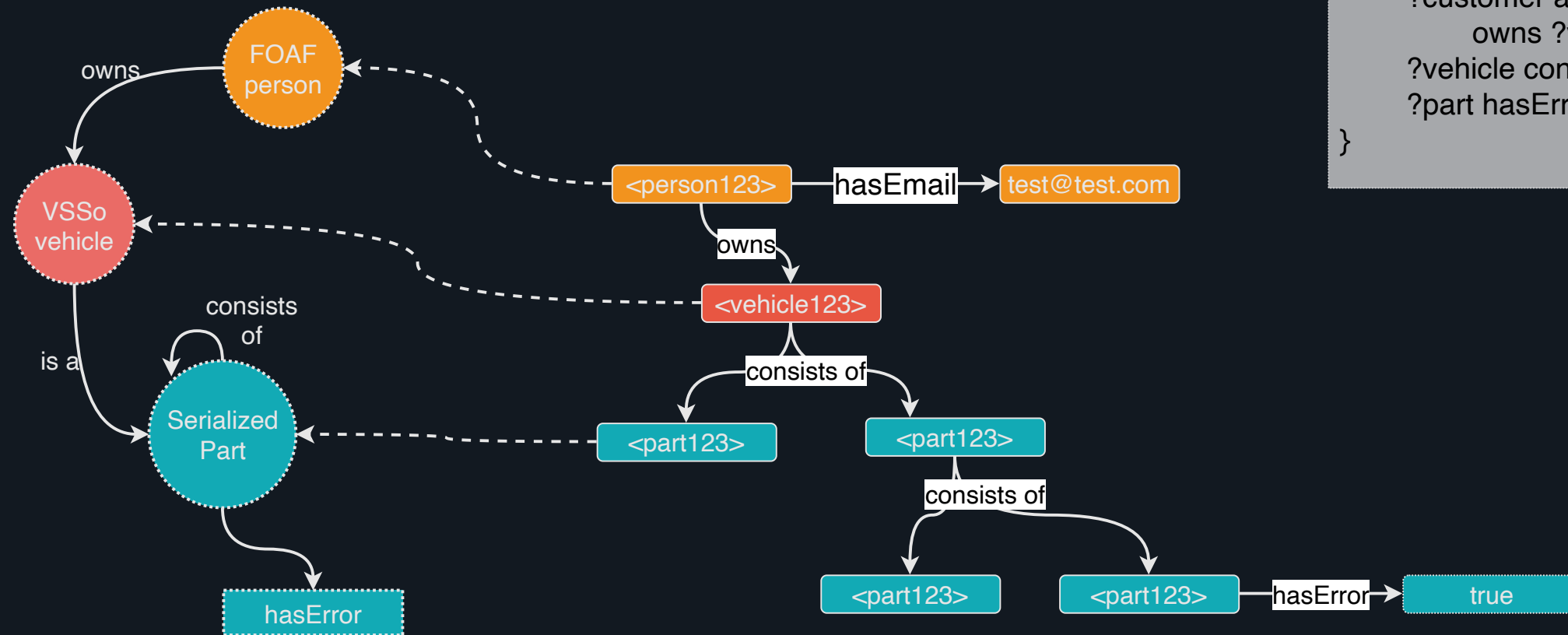
If a **part** of a **vehicle** has an issue, who is the **customer** to inform?

ONTOLOGY USE CASE



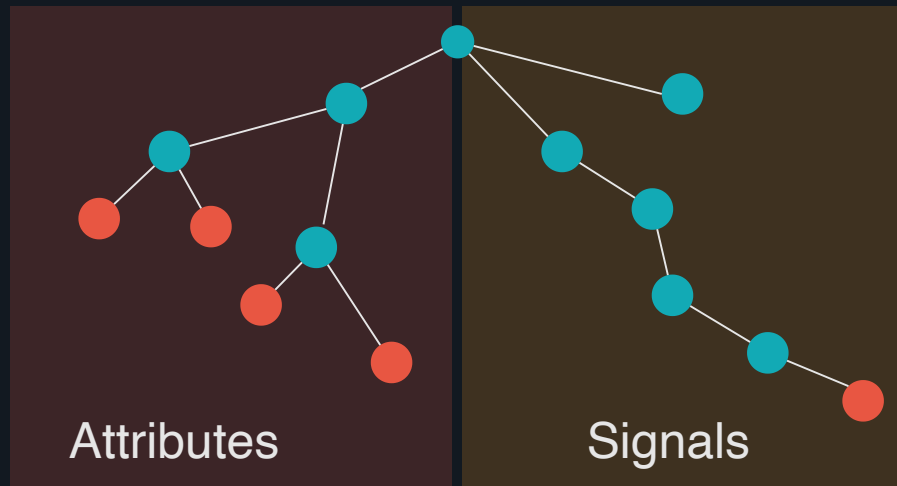
If a **part** of a **vehicle** has an issue, who is the **customer** to inform?

ONTOLOGY USE CASE



If a **part** of a **vehicle** has an issue, who is the **customer** to inform?

VEHICLE SIGNAL SPECIFICATION ONTOLOGY (VSS_o)



VSS

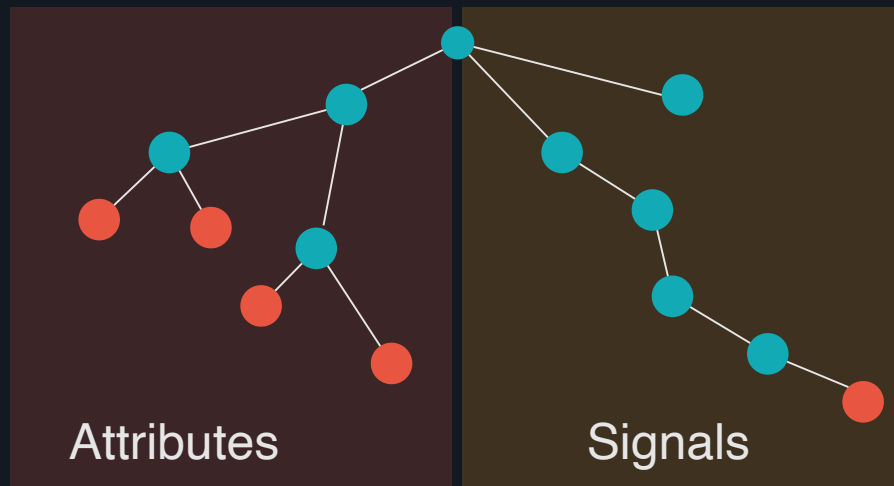
v1.0

2019

VEHICLE SIGNAL SPECIFICATION ONTOLOGY (VSSo)

Initial VSSo Development

Based on VSS v1.0 Benjamin Klotz created within his PhD the first version of VSSo backed by Eurescom and BMW. Creation and mapping mostly manual, extensions partially automatic. Results of his work influenced the development towards VSS v2.0.



VSS

v1.0

2019

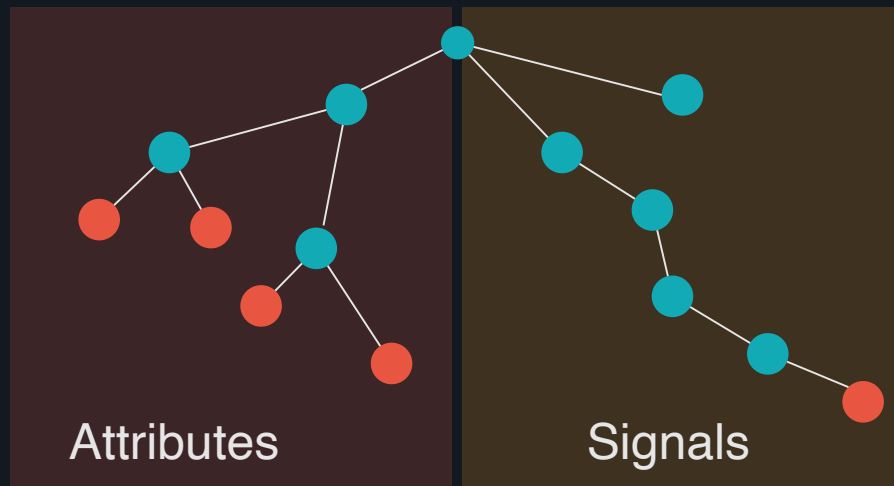
v2.0

2021

VEHICLE SIGNAL SPECIFICATION ONTOLOGY (VSSo)

Initial VSSo Development

Based on VSS v1.0 Benjamin Klotz created within his PhD the first version of VSSo backed by Eurescom and BMW. Creation and mapping mostly manual, extensions partially automatic. Results of his work influenced the development towards VSS v2.0.



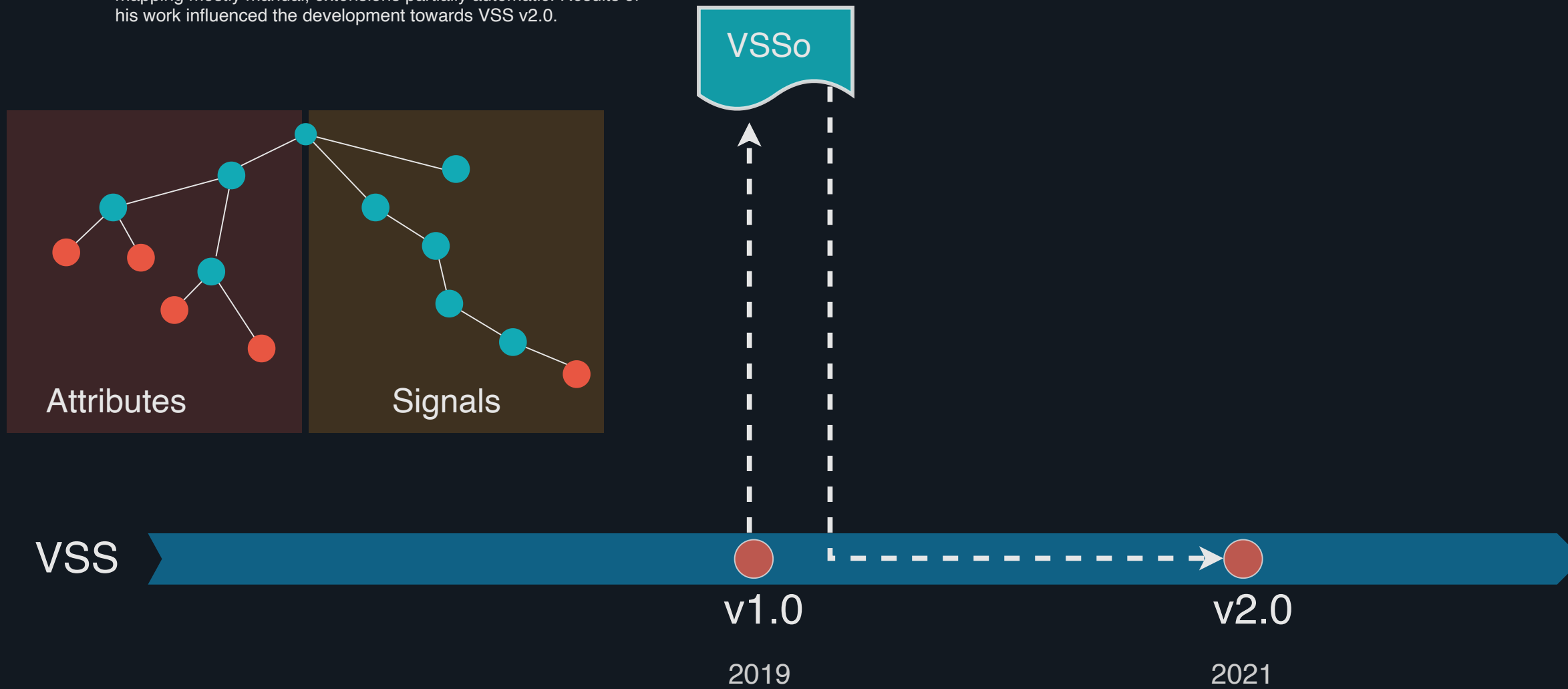
VSS

v1.0

2019

v2.0

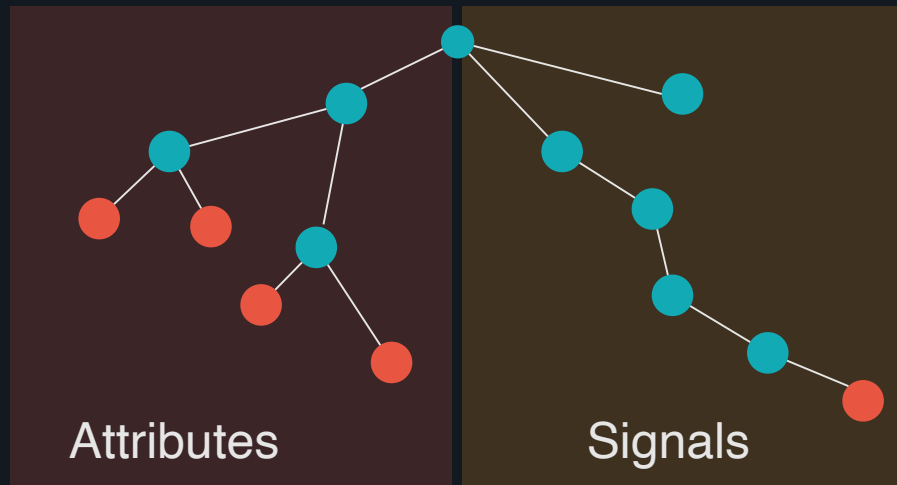
2021



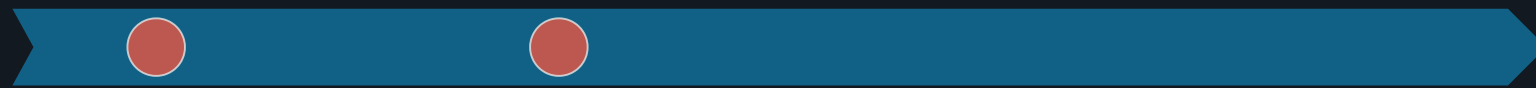
VEHICLE SIGNAL SPECIFICATION ONTOLOGY (VSSo)

Initial VSSo Development

Based on VSS v1.0 Benjamin Klotz created within his PhD the first version of VSSo backed by Eurecom and BMW. Creation and mapping mostly manual, extensions partially automatic. Results of his work influenced the development towards VSS v2.0.

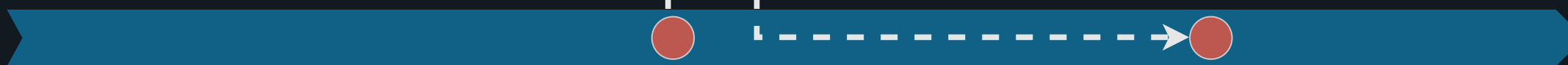


The W3C Automotive and Transportation Business Group has begun working on it.



VSSo was received as a member submission end of 2020.

VSS



v1.0

2019

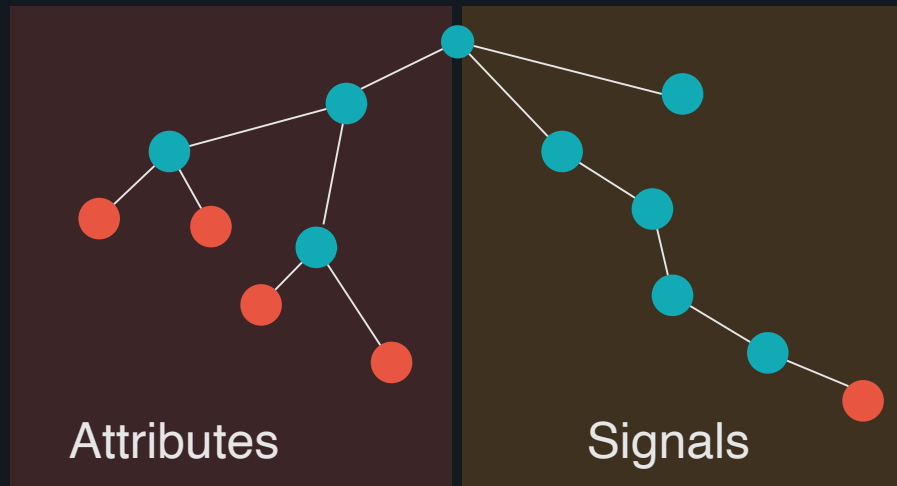
v2.0

2021

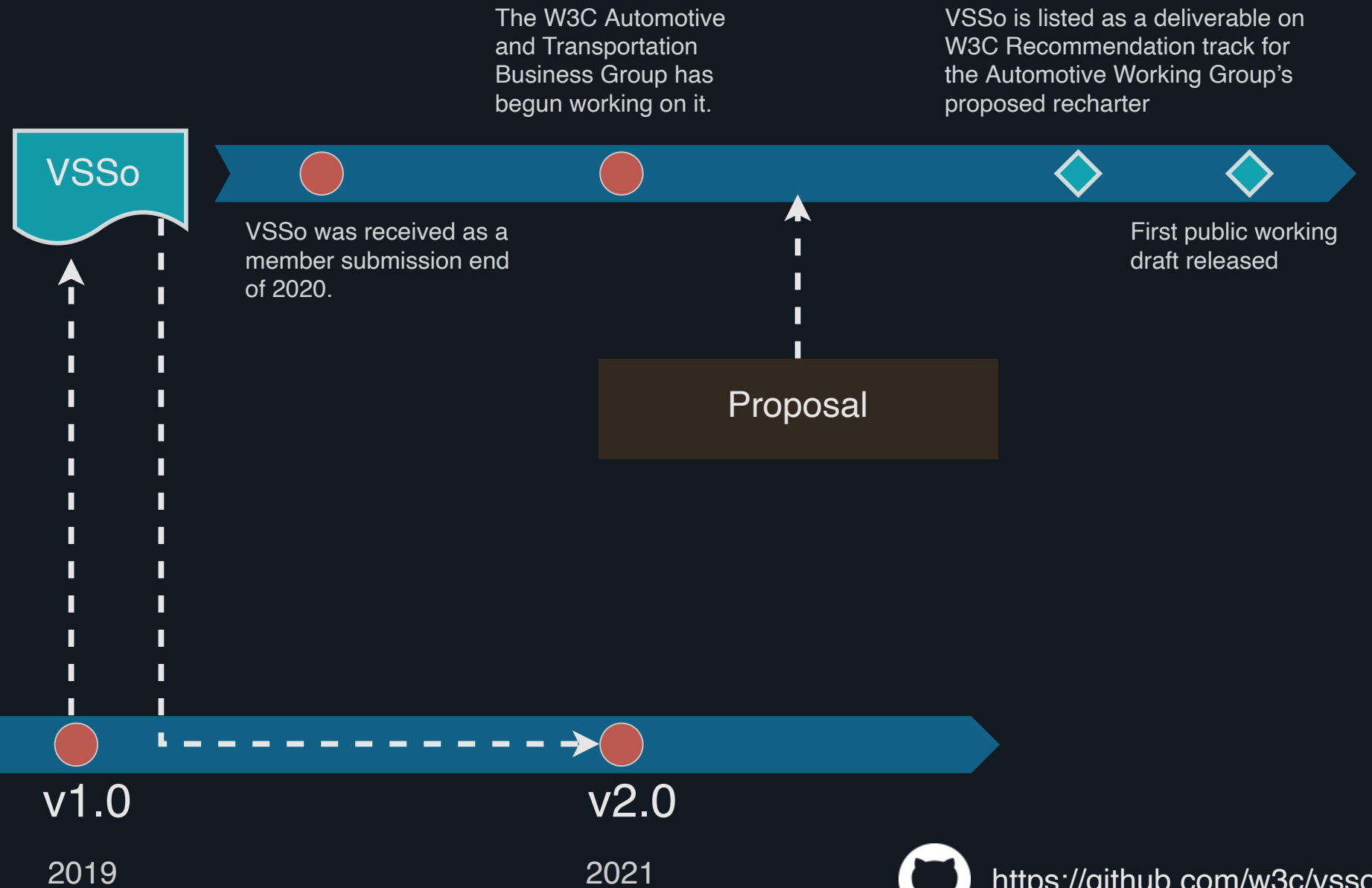
VEHICLE SIGNAL SPECIFICATION ONTOLOGY (VSSo)

Initial VSSo Development

Based on VSS v1.0 Benjamin Klotz created within his PhD the first version of VSSo backed by Eureka and BMW. Creation and mapping mostly manual, extensions partially automatic. Results of his work influenced the development towards VSS v2.0.



VSS

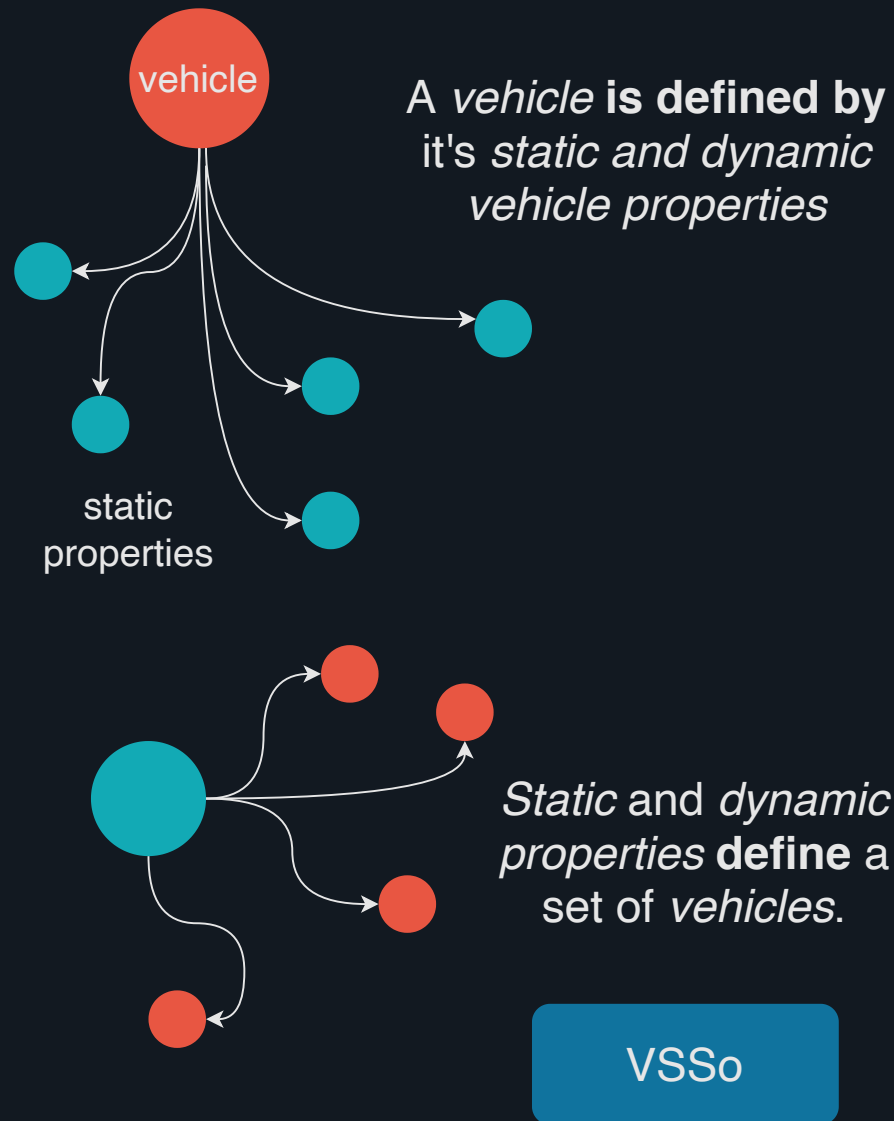


<https://github.com/w3c/vsso>

ANALYTICS

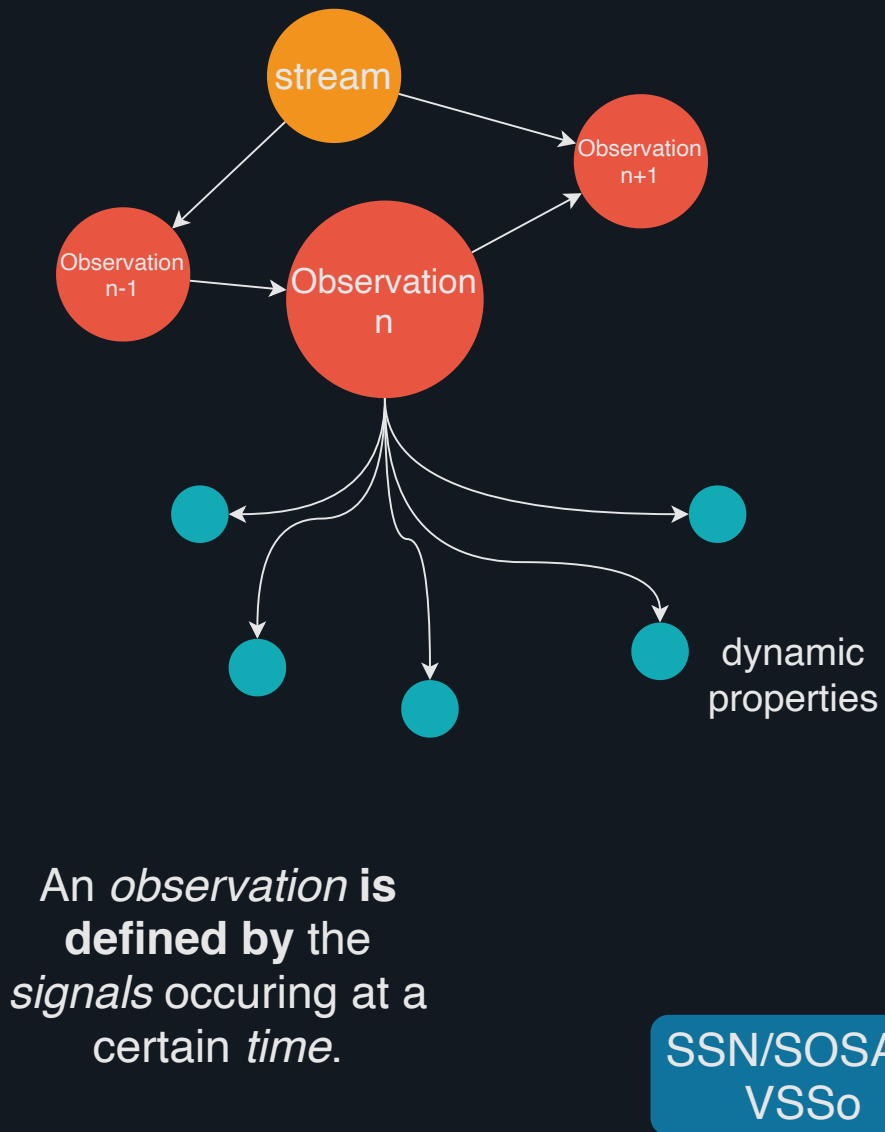
Current Vehicle Data

Find out what state a specific vehicle is in and how values of static and dynamic properties define a vehicle fleet.



Dynamic Vehicle Data over Time

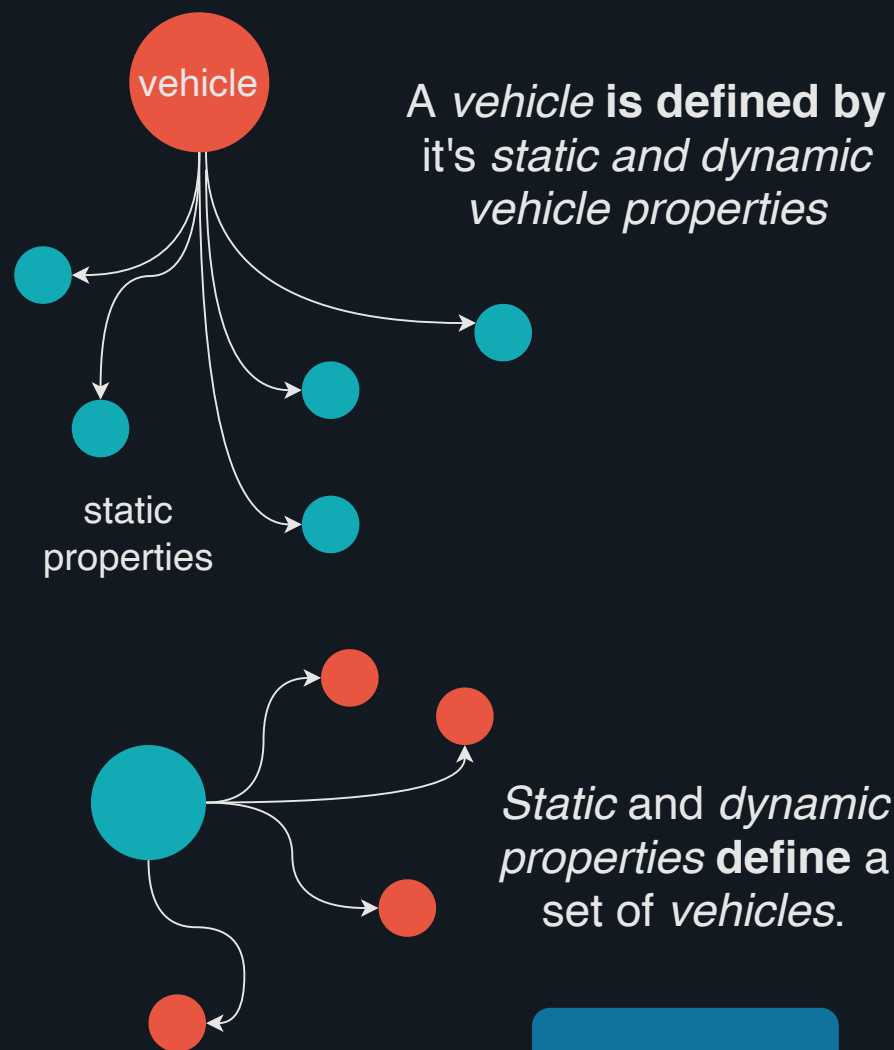
Which observations have been made at which point of time? How a data stream develops over time? Can I act on it?



ANALYTICS

Current Vehicle Data

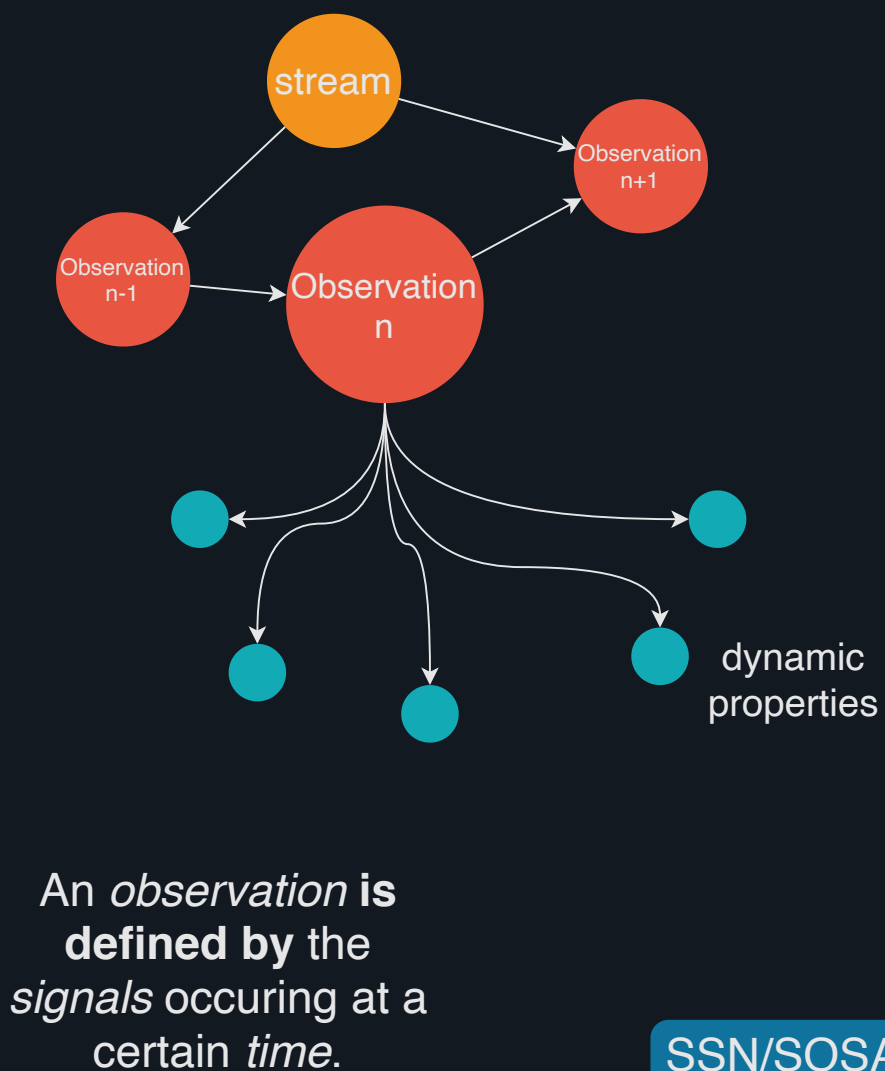
Find out what state a specific vehicle is in and how values of static and dynamic properties define a vehicle fleet.



VSSo

Dynamic Vehicle Data over Time

Which observations have been made at which point of time? How a data stream develops over time? Can I act on it?

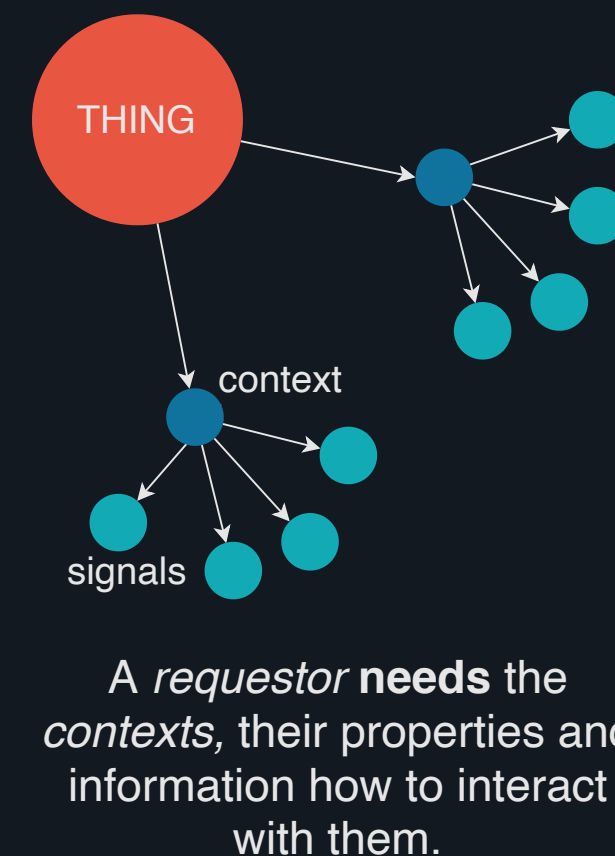


SSN/SOSA + VSSo

SERVICES

Interaction with Vehicle Data

Get specific values using a specific protocol of a specific data provider. Agreement on data exchange with unit type, etc.



WoT + VSSo

VEHICLE SIGNAL SPECIFICATION ONTOLOGY (VSSo)

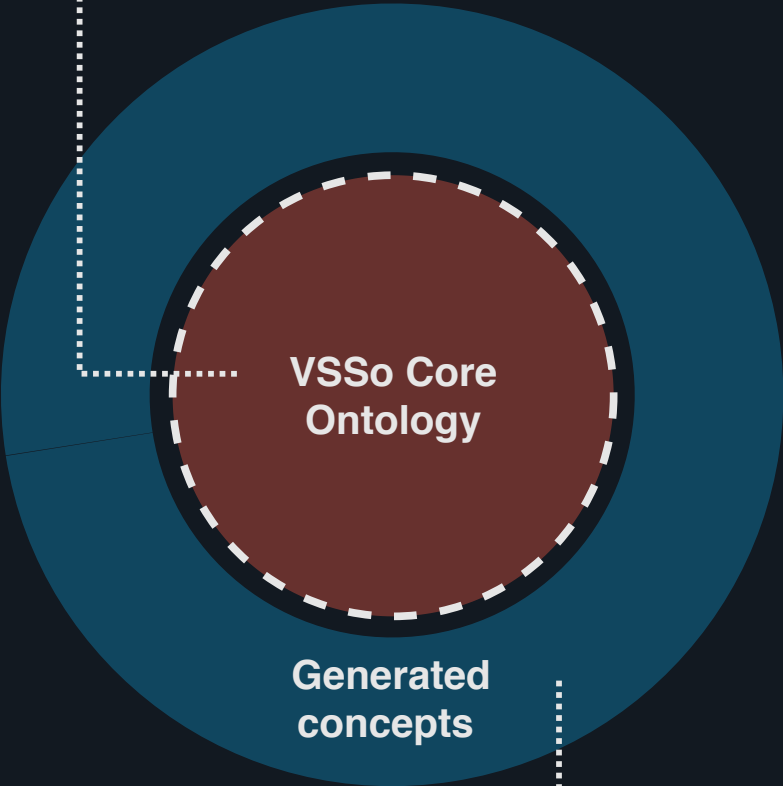
Define the core **structural** concepts of VSS (e.g. Branch, Attributes, Sensors, etc.)



VSSo Core Ontology

VEHICLE SIGNAL SPECIFICATION ONTOLOGY (VSSo)

Define the core **structural** concepts of VSS (e.g. Branch, Attributes, Sensors, etc.)



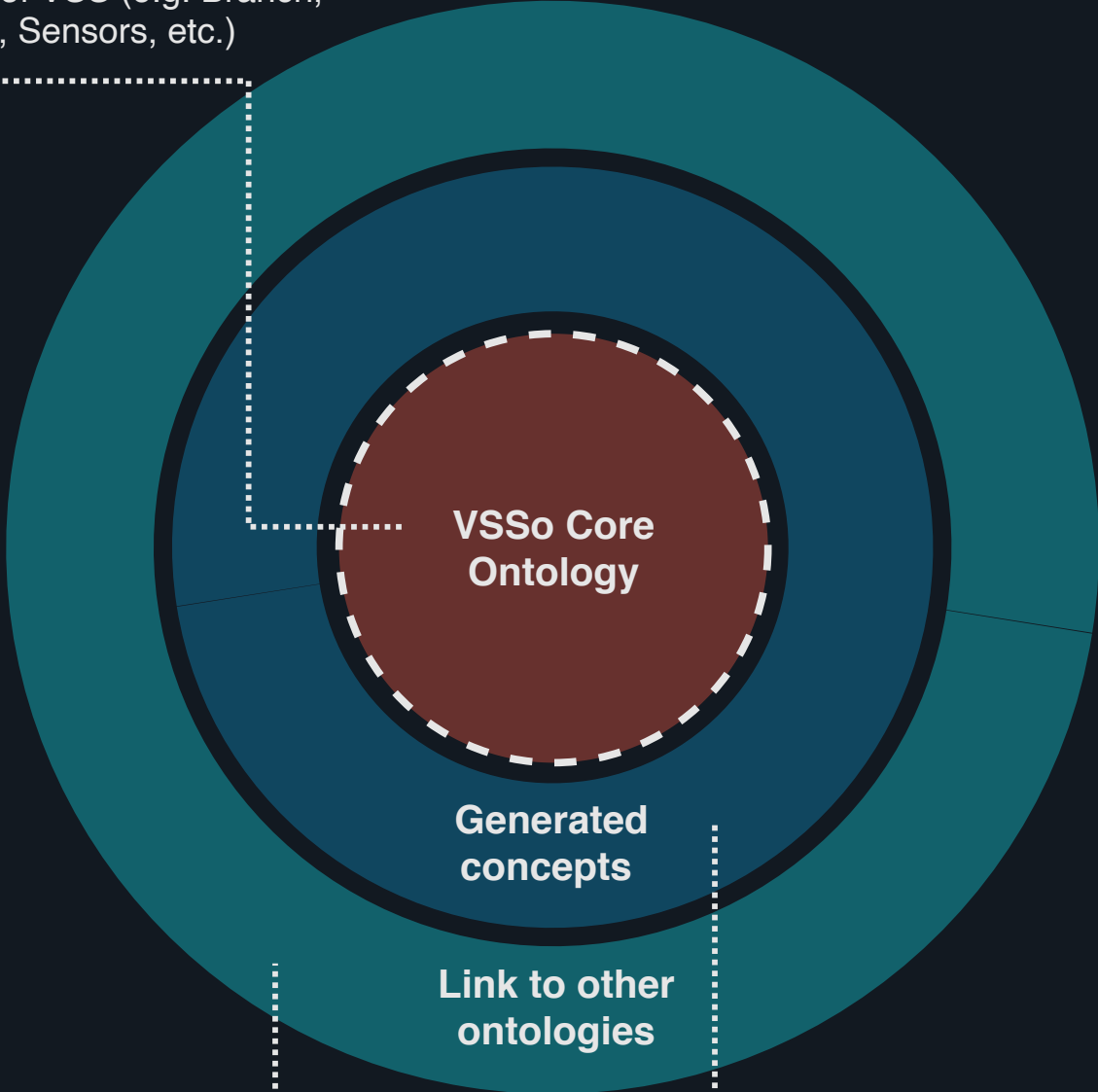
VSSo Core Ontology

Generated concepts

Generate the data definitions from VSS based on the core ontology.

VEHICLE SIGNAL SPECIFICATION ONTOLOGY (VSSo)

Define the core **structural** concepts of VSS (e.g. Branch, Attributes, Sensors, etc.)



VSSo Core Ontology

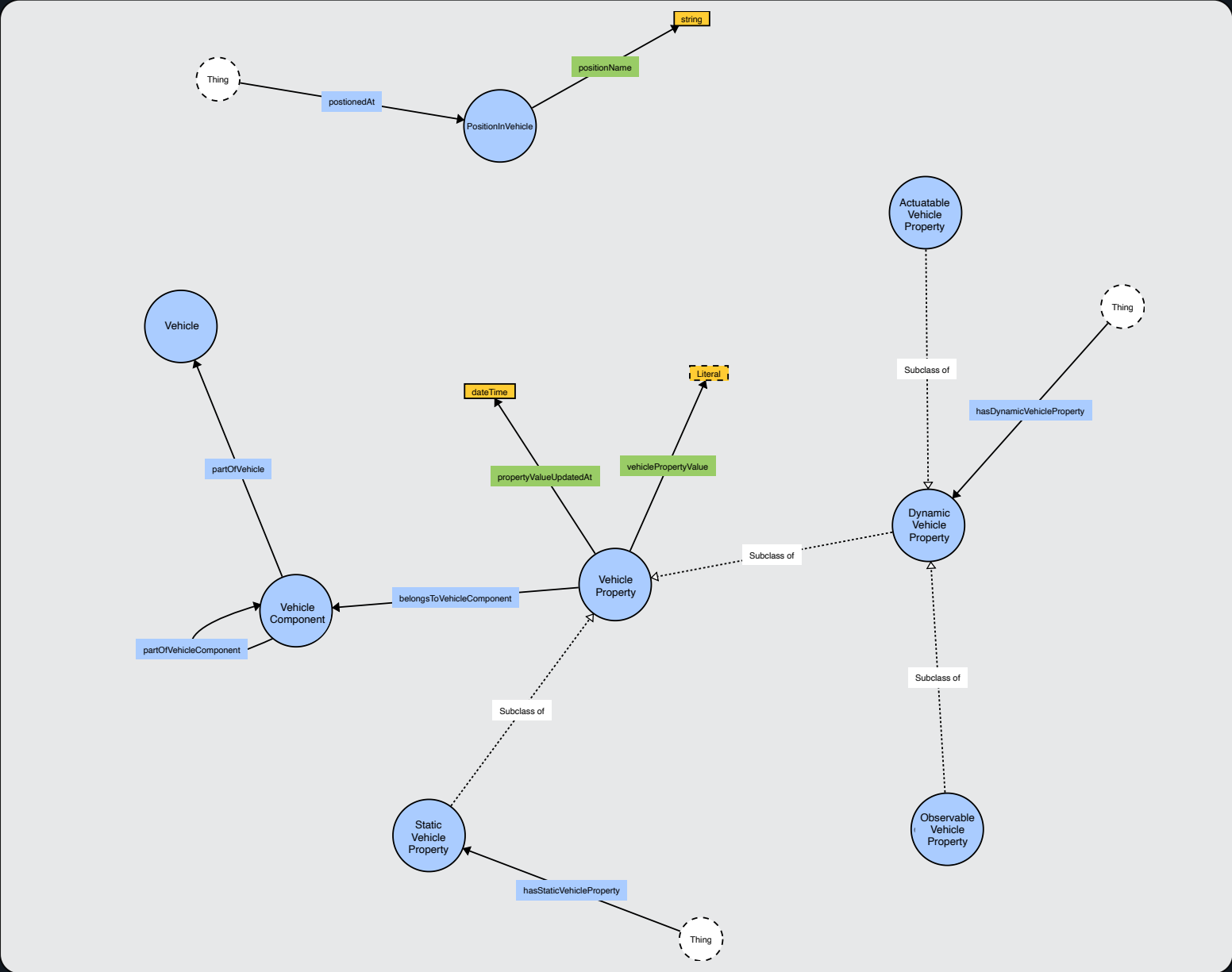
Generated concepts

Link to other ontologies

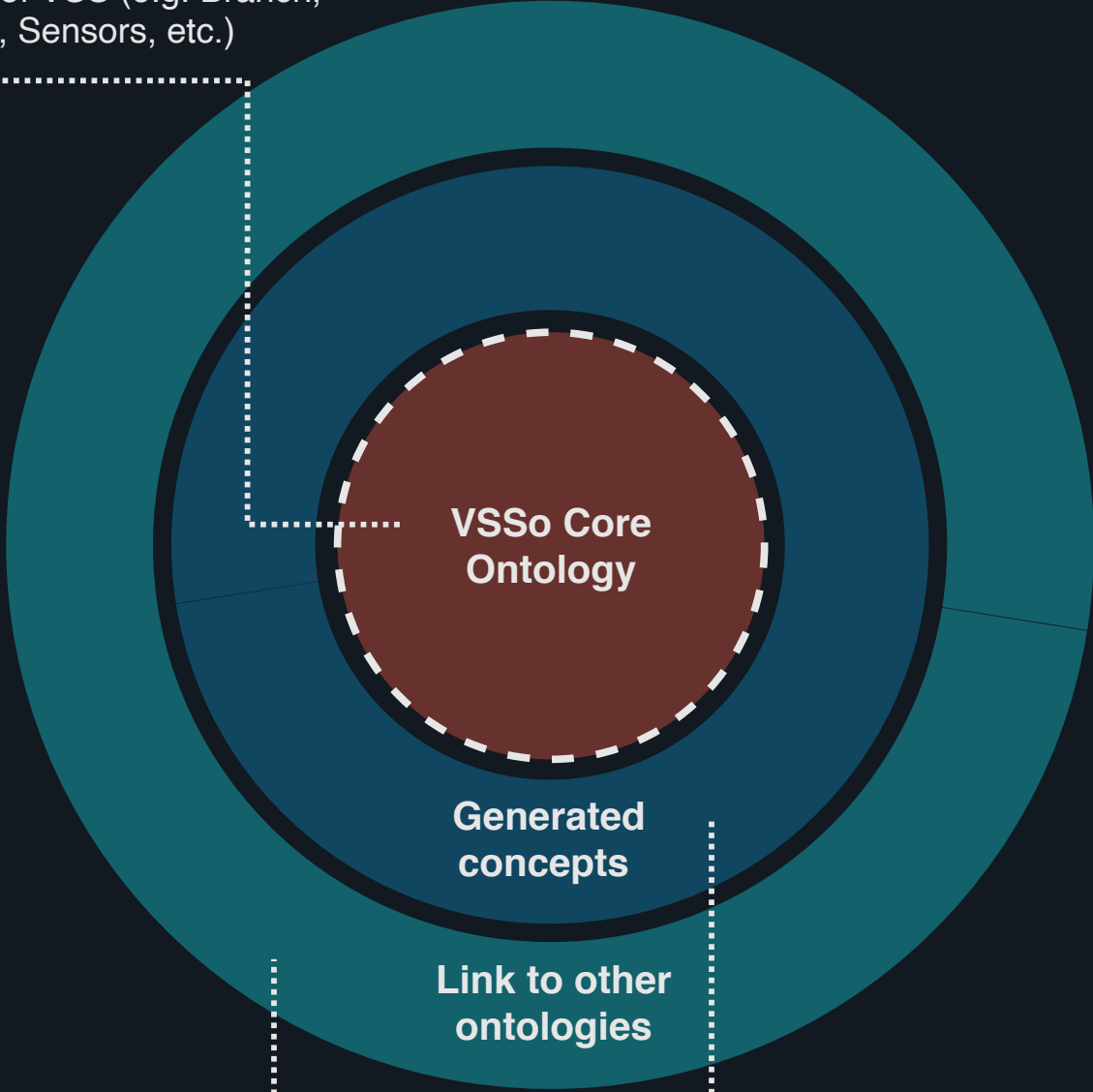
Use VSSo as domain ontology for other, widely adopted standards (SSN/SOSA).

Generate the data definitions from VSS based on the core ontology.

VEHICLE SIGNAL SPECIFICATION ONTOLOGY (VSSo)



Define the core **structural** concepts of VSS (e.g. Branch, Attributes, Sensors, etc.)



Use VSSo as domain ontology for other, widely adopted standards (SSN/SOSA).

Generate the data definitions from VSS based on the core ontology.