Adnan Bekan
Senior Research Engineer: Data Architecture, Connected Vehicle, Standards & Platforms
BMW Group

Intro to Vehicle Signal Specification (VSS)

COVESA
Accelerating the future of connected vehicles

Introduction to Vehicle Signal Specification

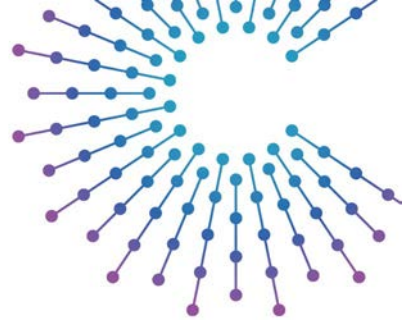
ALL MEMBER MEETING

October 10-12, 2023

COVESA

Accelerating the future of connected vehicles

Agenda



What will be covered:

- What VSS is and what it is NOT.
- The value of VSS.
- Adoption Paths.
- Resources.
- Current Release.
- Next steps.
- Example with service architecture.

What will not be covered:

- Technical deep dive.
- Detailed implementations.

COVESA
Accelerating the future of connected vehicles

What is VSS ?

COVESA's Vehicle Signal Specification (VSS) is a common approach for describing vehicle data.

What does it offer: It is a widely adopted; developer friendly, extensible data model & catalog with industry supported tooling. VSS provides a common understanding across the value chain of the connected vehicle.

What value does it bring: This enables improved interoperability and integration which ultimately saves time and cost thus allowing companies to focus on business value creation and differentiating solutions.

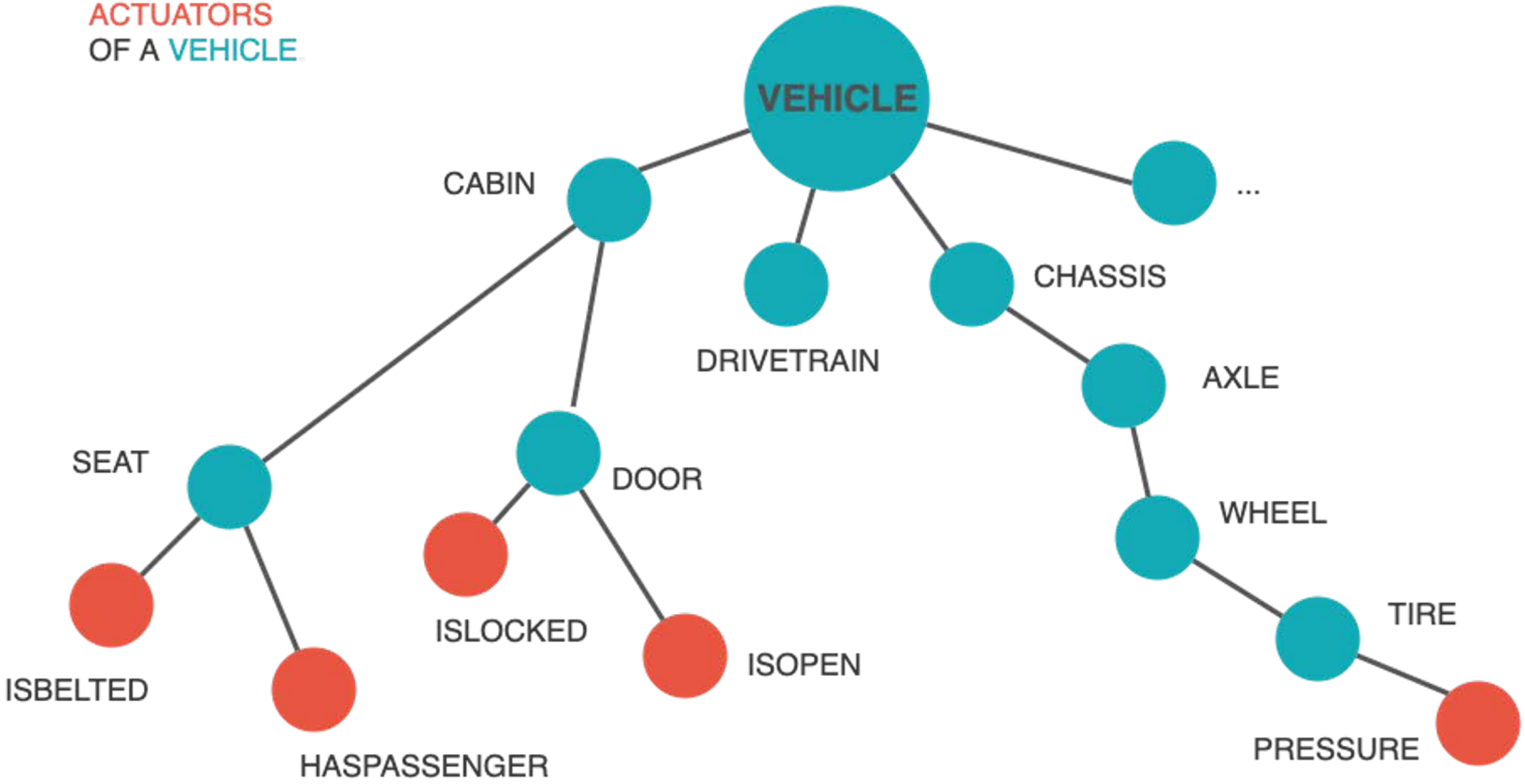
What is VSS - Point by Point

- A simple, flexible and protocol agnostic common approach for describing vehicle data for machines & humans.
- Extensible data model & catalog with industry supported tooling.
- Provides a common understanding across the value chain of the Connected Vehicle.
- Enables improved interoperability and integration, saving time and cost.
- Can be used in car, cloud, edge, or wherever you need it.
- Allows focus on business value creation and differentiating products and solutions.
- Widely adopted.
- Developer friendly.

COVESA
Accelerating the future of connected vehicles

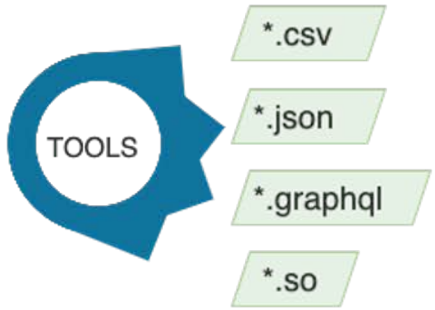
VEHICLE SIGNAL SPECIFICATION

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE




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

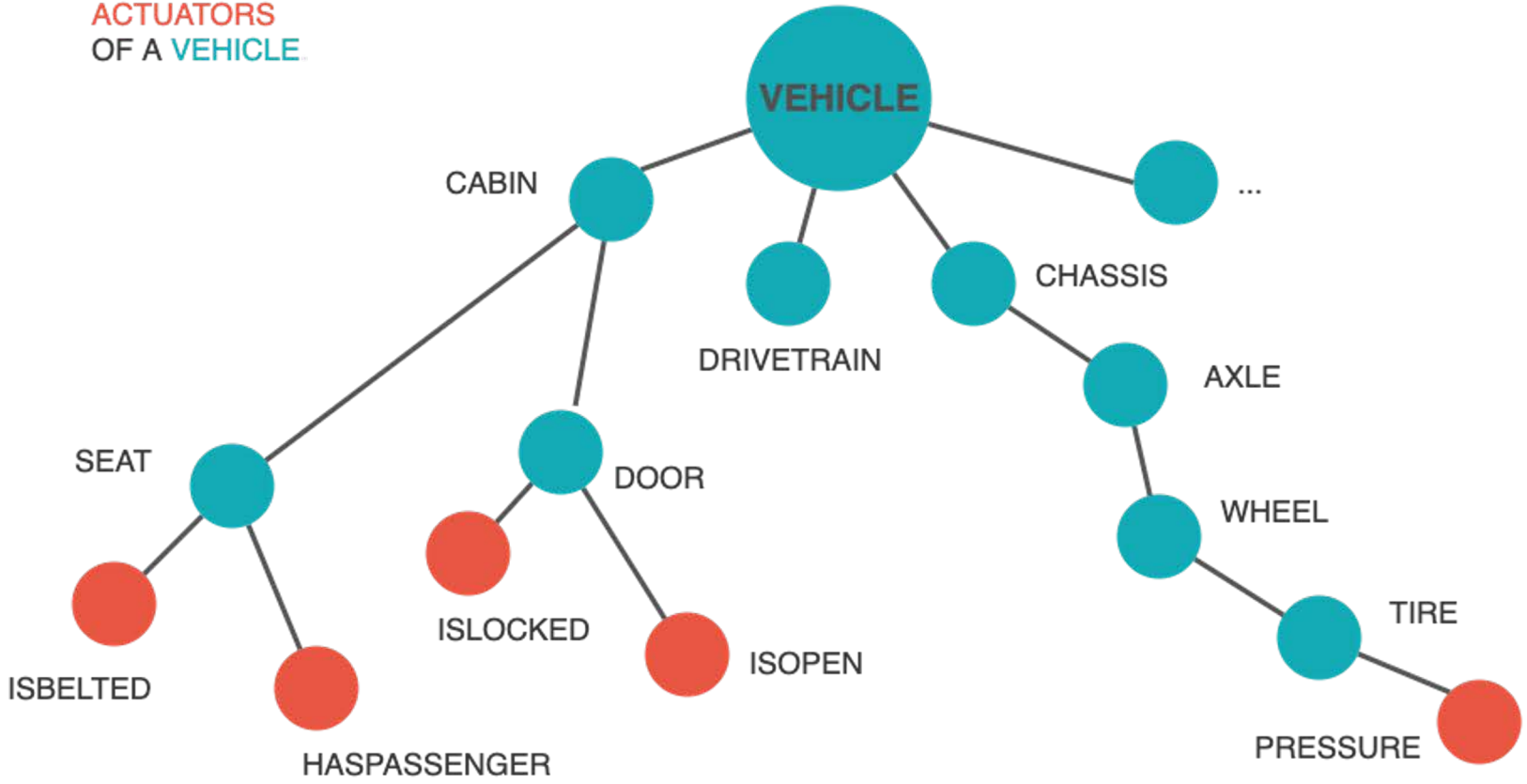
YAML SPECIFICATION



COVESA
Accelerating the future of connected vehicles

VEHICLE SIGNAL SPECIFICATION

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



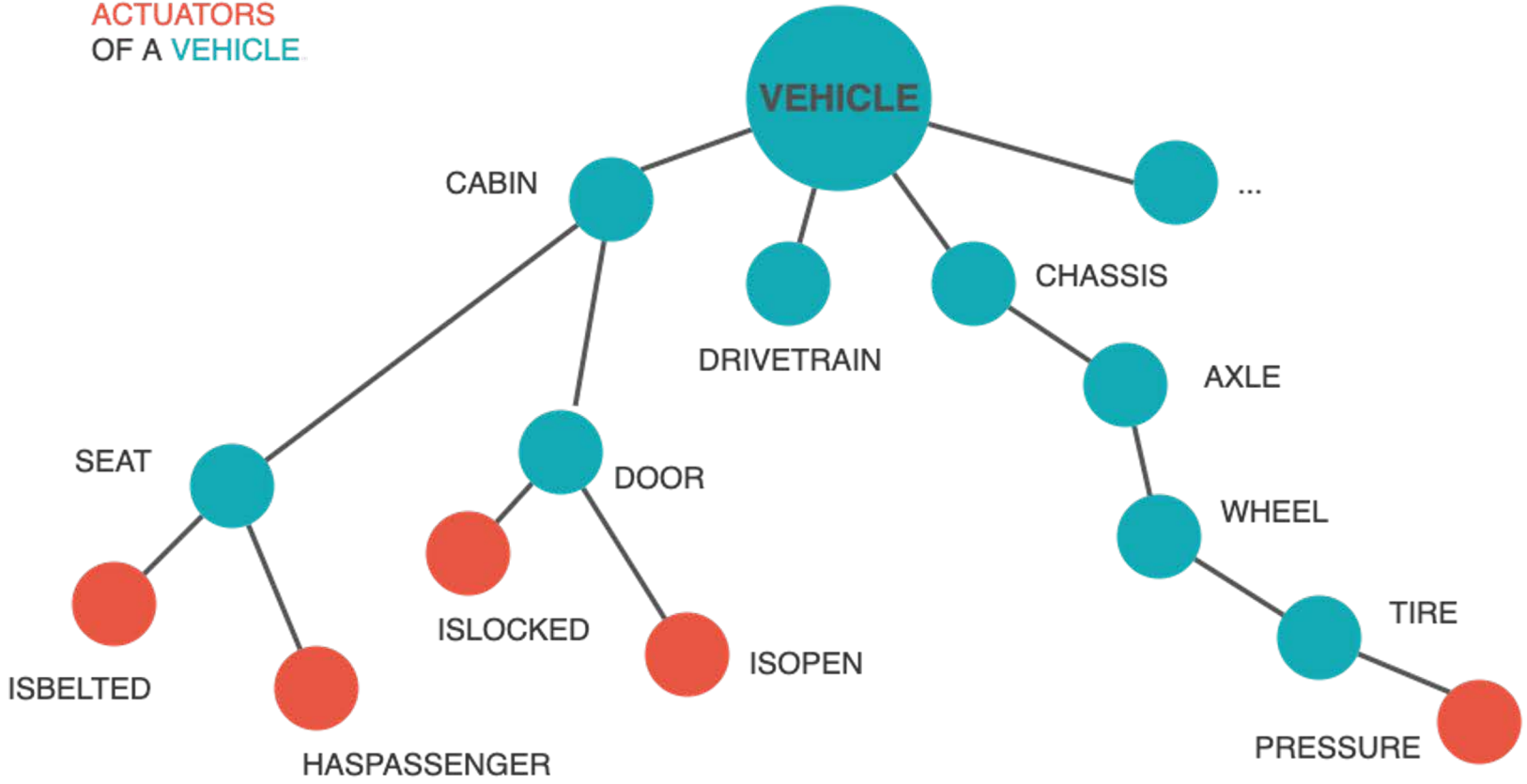
Vehicle.Drivetrain.Transmission.Speed
type: sensor
datatype: float
unit: km/h
description: The vehicle speed as measured by the drivetrain

YAML SPECIFICATION

COVESA
Accelerating the future of connected vehicles

VEHICLE SIGNAL SPECIFICATION

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.

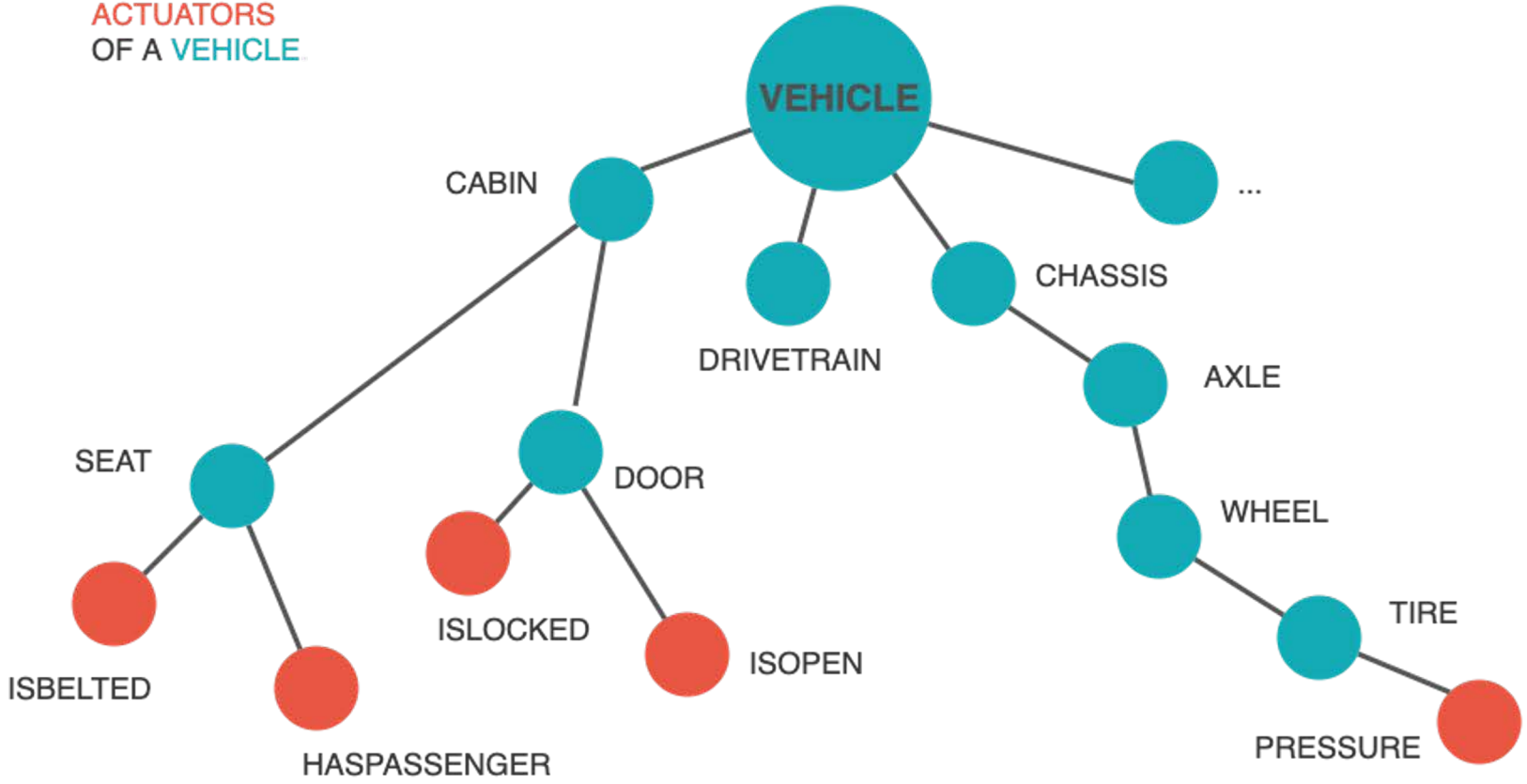


Vehicle.Drivetrain.Transmission.Speed
type: sensor
datatype: float
unit: km/h
description: The vehicle speed as measured by the drivetrain

YAML SPECIFICATION

VEHICLE SIGNAL SPECIFICATION

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.

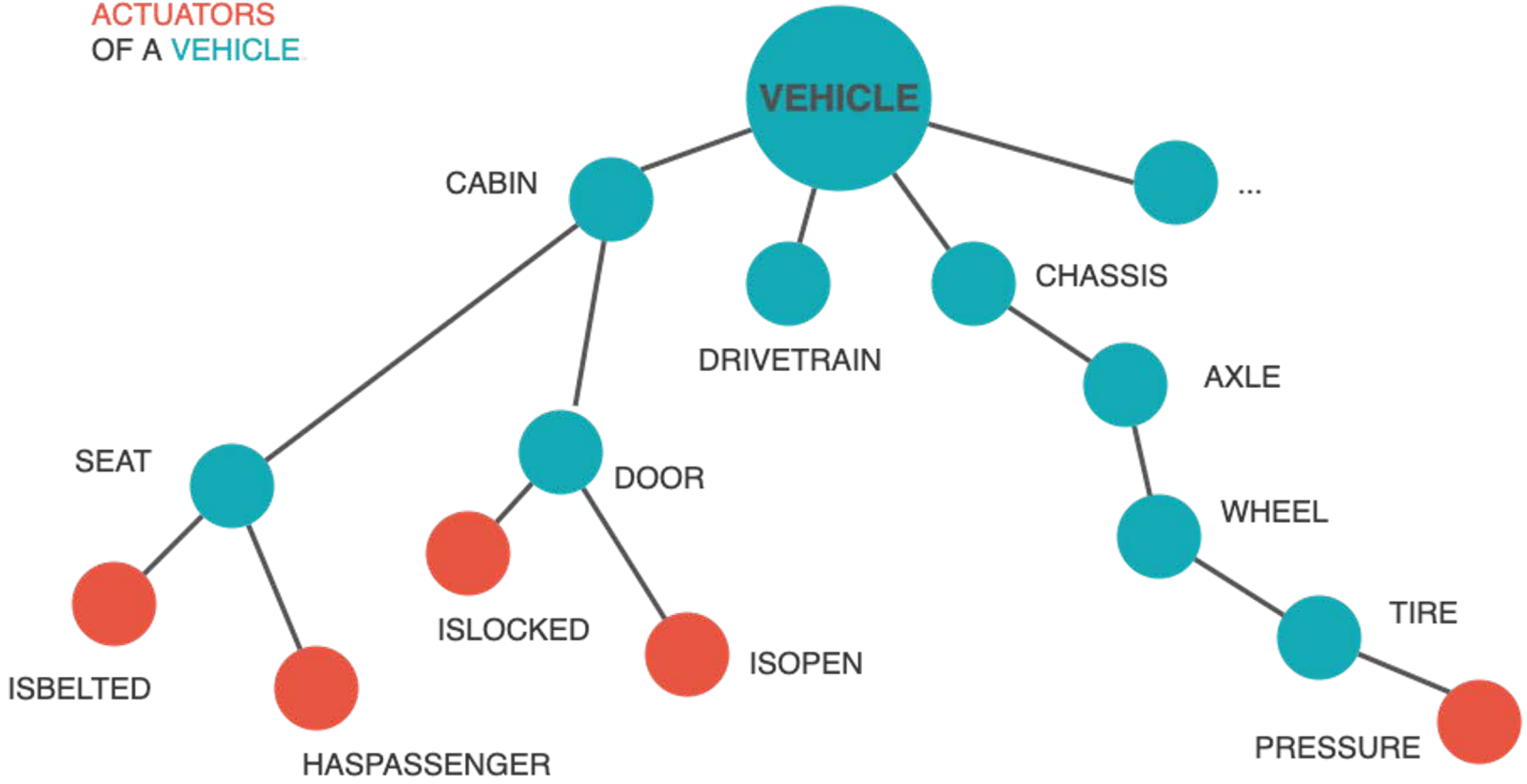


Vehicle.Drivetrain.Transmission.Speed
type: sensor
datatype: float
unit: km/h
description: The vehicle speed as measured by the drivetrain

YAML SPECIFICATION

VEHICLE SIGNAL SPECIFICATION

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.

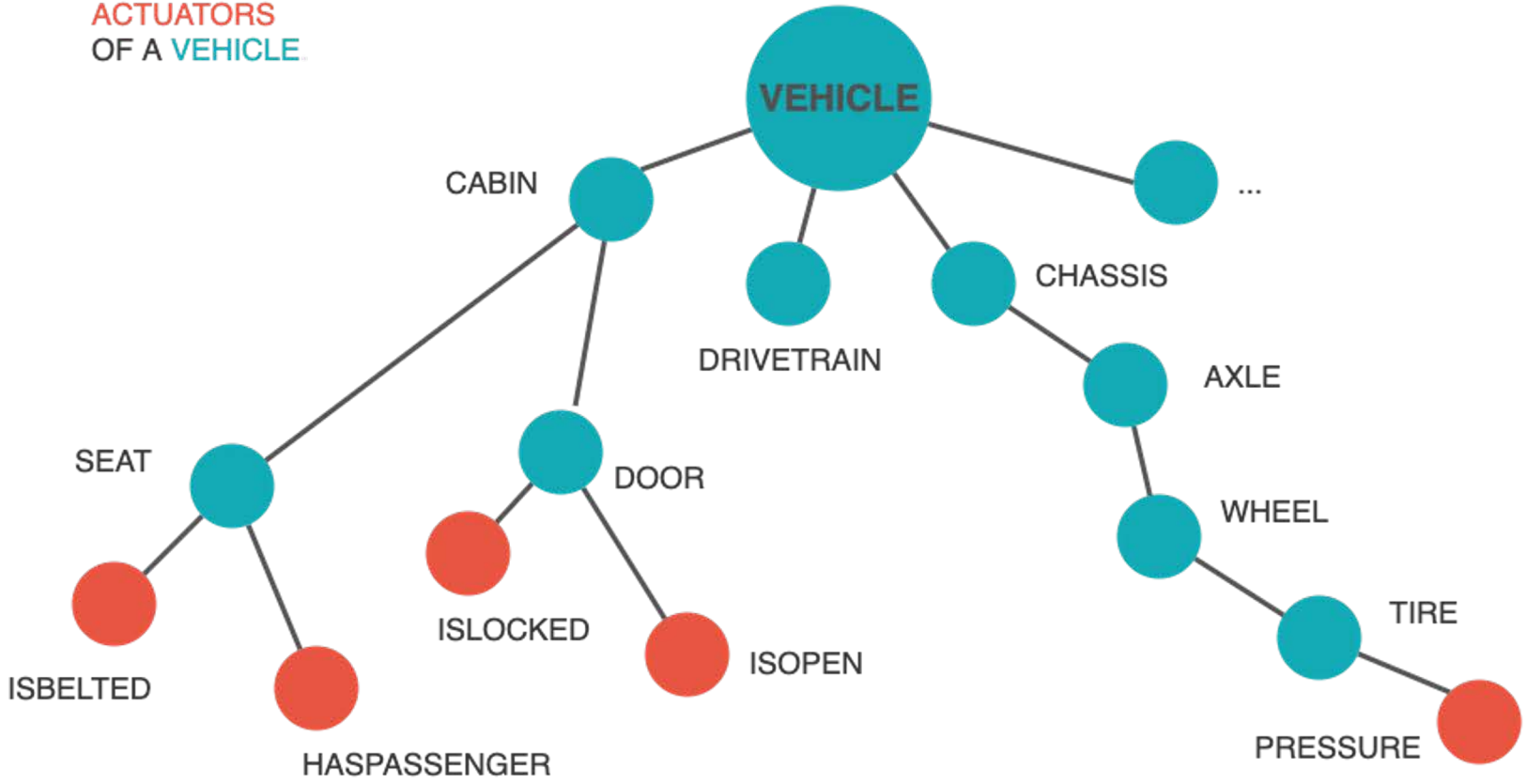


Vehicle.Drivetrain.Transmission.Speed
type: sensor
datatype: float
unit: km/h
description: The vehicle speed as measured by the drivetrain

YAML SPECIFICATION

VEHICLE SIGNAL SPECIFICATION

TAXONOMY
FOR
ATTRIBUTES,
SENSORS AND
ACTUATORS
OF A VEHICLE.



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

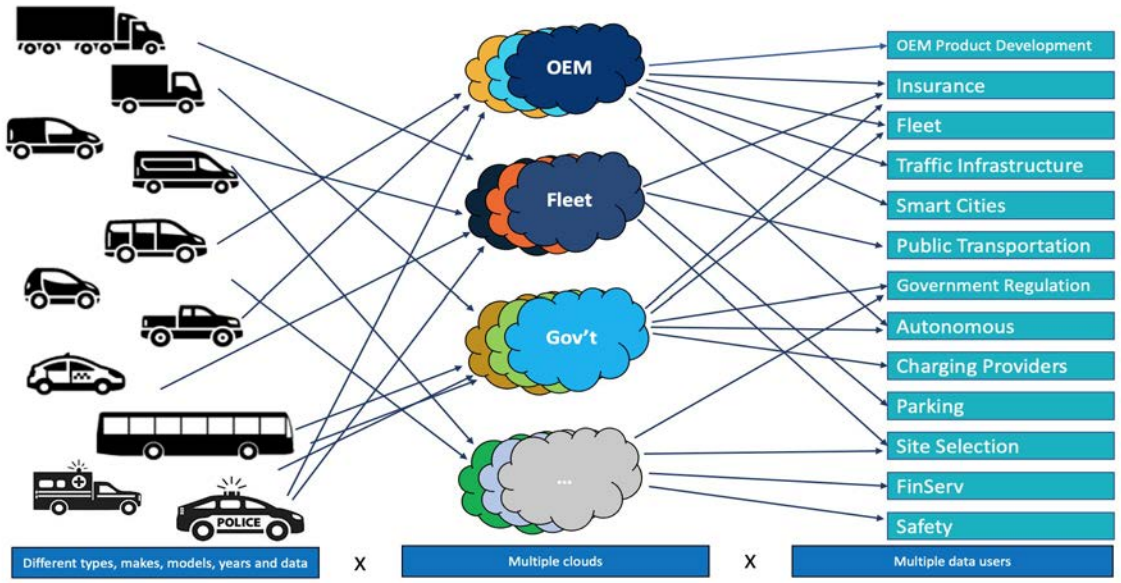
YAML SPECIFICATION

VSS is not…

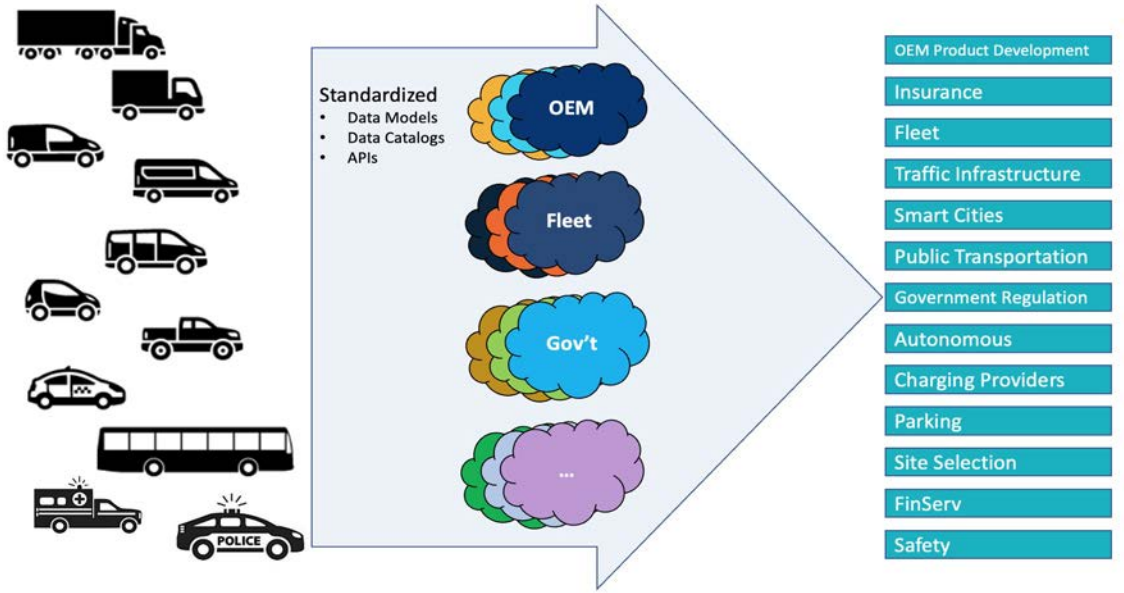
- a protocol
- a serialization format
- an API specification
- Interface Definition Language
- a model for everything in the world
- limited to car, cloud, edge, or…
- a server specification
- an ontology

Why Now?

Today: Integration efforts are complex and unmaintainable.



VSS + Consistent Interface reduce complex integration effort and promotes focus on value



Current Reality

- 100 million connected vehicles today
- 400 million connected vehicles by 2025
- Vehicles generate hundreds of GB of data a day or more
- Average vehicle has thousands of signals
- Multiple vehicle models, model years and variation per OEM
- Multiple Clouds (OEM, Fleet, Gov't…)
- Multiple data consumers: OEM, Fleet, Smart City, Insurance, Traffic Infrastructure, Finserv, Charging Providers, …

Challenges

- Siloed data by organization and domain
- Complex integration and an interoperability
- Estimated 80% of time focused on integration and 20% focused on value
- A lot of noise!

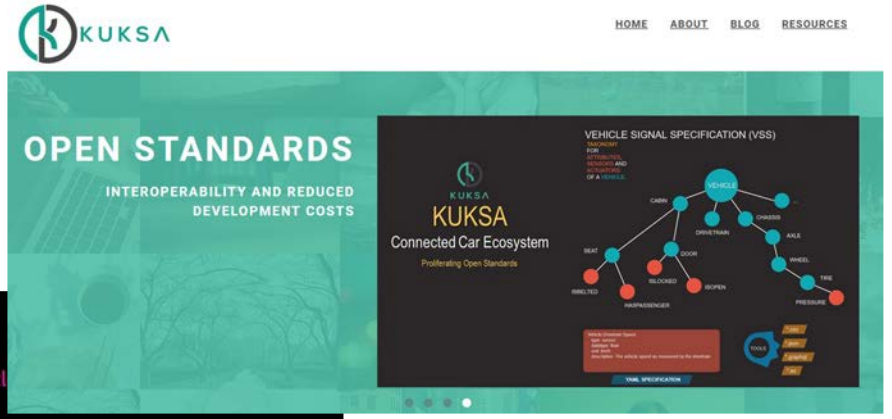
Solutions

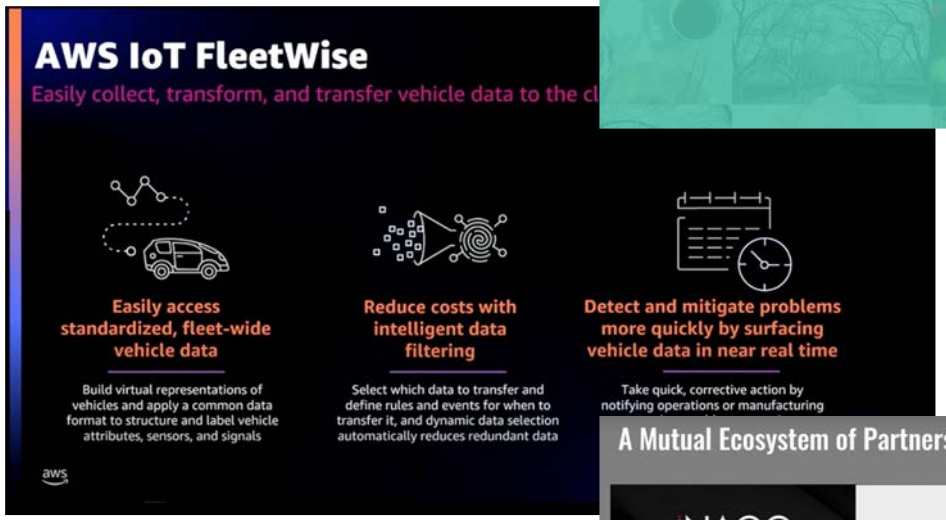
- Standardize data models
- Standardize data catalogs
- Standardize APIs

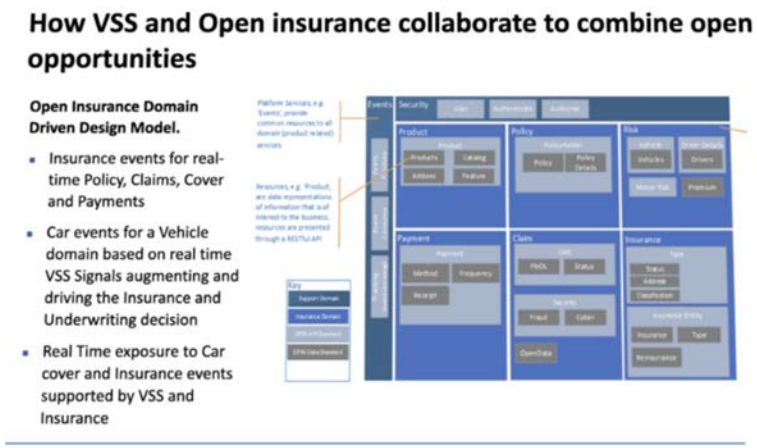
COVESA
Accelerating the future of connected vehicles

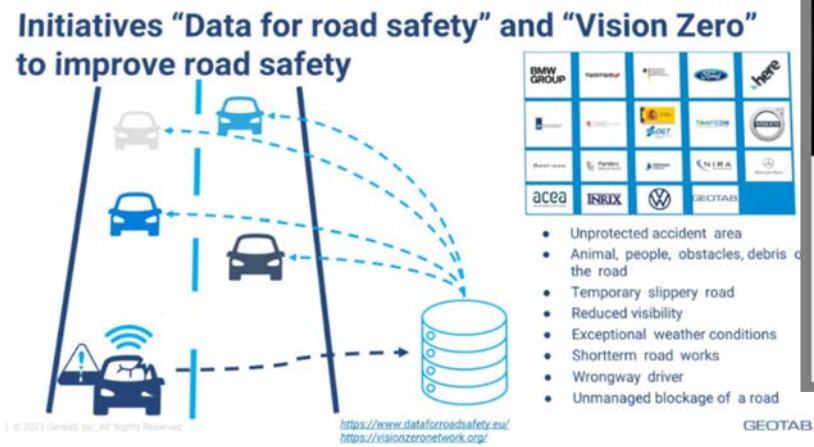
But, where it is used?

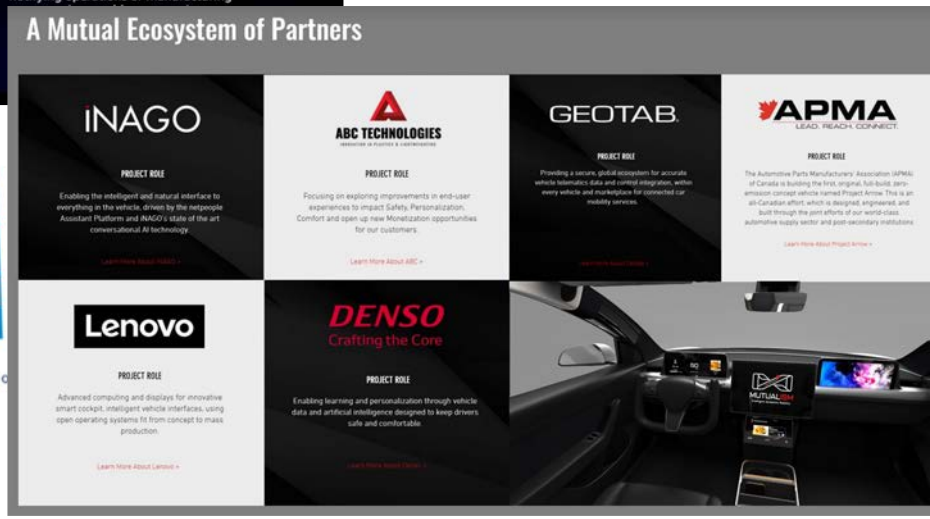
- OEM
- Fleet Management
- Usage Based Insurance integrations
- Road Safety
- Traffic Infrastructure and Planning
- Large government projects
- Use by other standards











COVESA
Accelerating the future of connected vehicles

Adopters

COVESA
Accelerating the future of connected vehicles

Relationship to Other Initiatives

W3C Vehicle Information Service Specification (VISS)	VSS is the data model used by W3C VISS
Vehicle Signal Specification Ontology	VSSo is directly correlated with the VSS catalog
Common Vehicle Interfaces	VSS will be a data model used in API definition of services.
Data Architecture	VSS as data model for Vehicle Domain.

Paths to Adoption

Build your own

- Use common catalog to “get started”
- Extend your own catalog privately
- Create a custom catalog

Buy/Use a solution

- Use existing VSS data server
- Buy something VSS-ready…

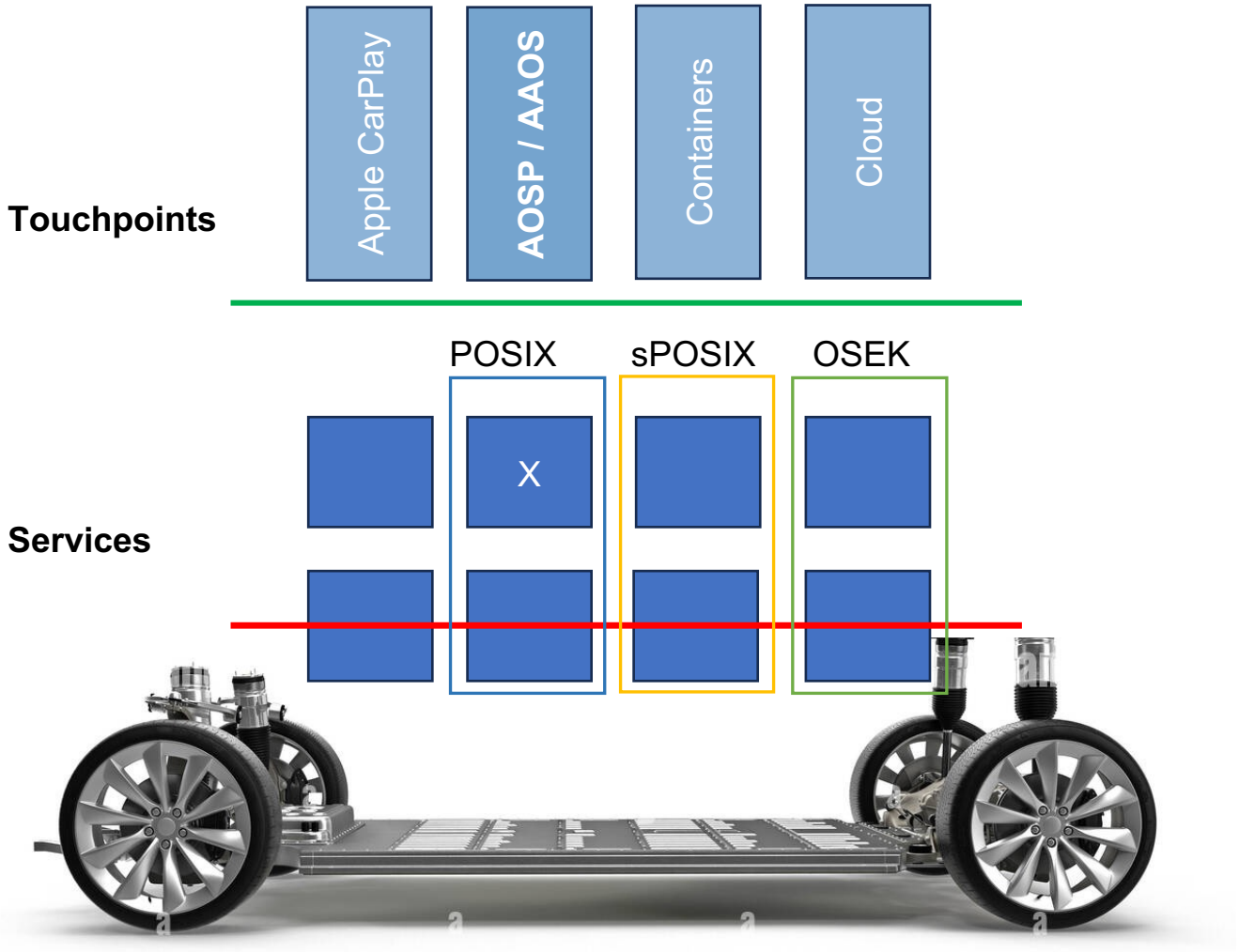
Contribute & grow

- Become part of the community and help to shape VSS for what you need

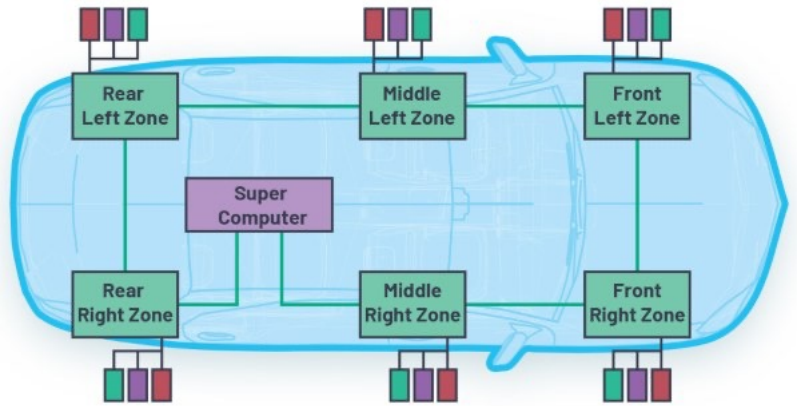
Resources

Documentation	https://covesa.github.io/vehicle_signal_specification/ https://wiki.covesa.global/display/WIK4/VSS+Resources+at+a+Glance	Please read the documentation first. It answers most questions and points you to where to find other information.
Specification	https://github.com/COVESA/vehicle_signal_specification	To see specific nodes such as Cabin or Powertrain drill down into https://github.com/COVESA/vehicle_signal_specification/tree/master/spec
Tools	https://github.com/COVESA/vss-tools	Tools for building and processing VSS
GitHub Issues and Wiki	https://github.com/COVESA/vehicle_signal_specification/issues https://github.com/COVESA/vehicle_signal_specification/wiki	The team uses GitHub Issues and Wiki heavily
Vehicle Signal Specification ontology	https://www.w3.org/TR/vsso/	Ontology built with VSS
Vehicle Information Service Specification	https://www.w3.org/TR/viss2-core/#:~:text=The%20Vehicle%20Information%20Service%20Specification,Vehicle%20Signal%20Specification%20(VSS).	Server specification for accessing vehicle information represented by VSS

Example 1/5 – HLA

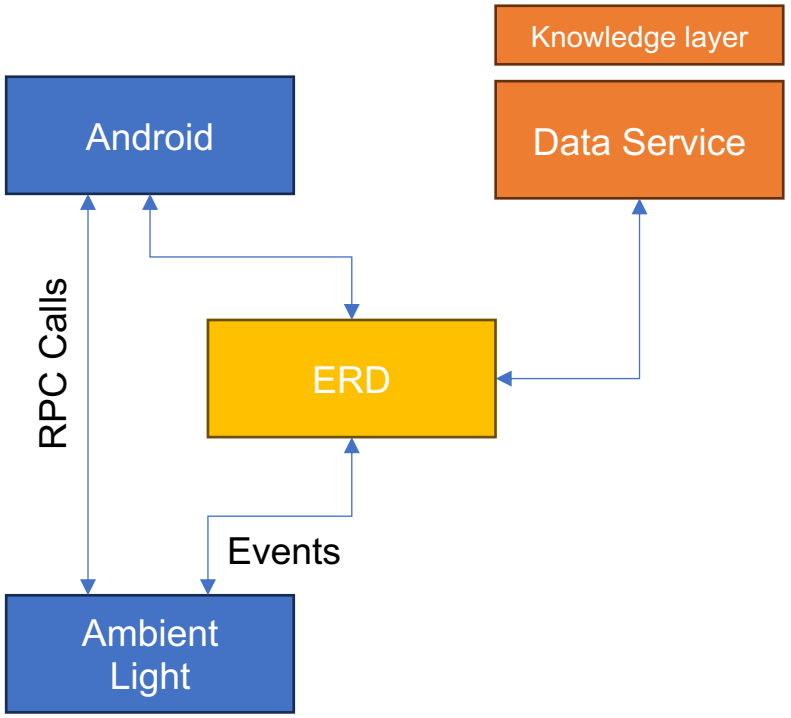


What type of arch?
- Client-Server
- Event-Driven
….

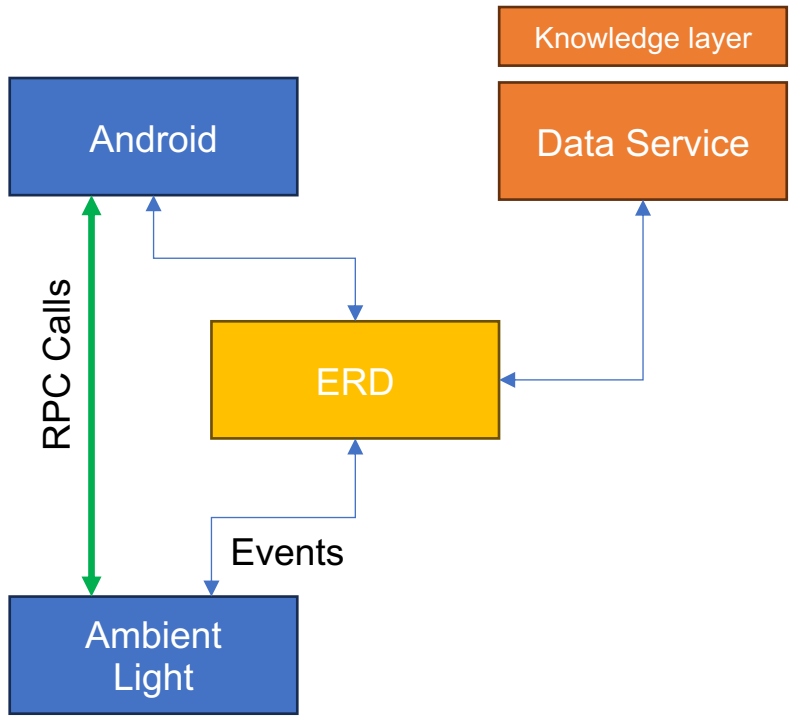


Most common challenges are
Fragmentation, Usability, Integration
(Test and Validation)

Example 2/5 -HLA



Example 3/5 – “RPC”

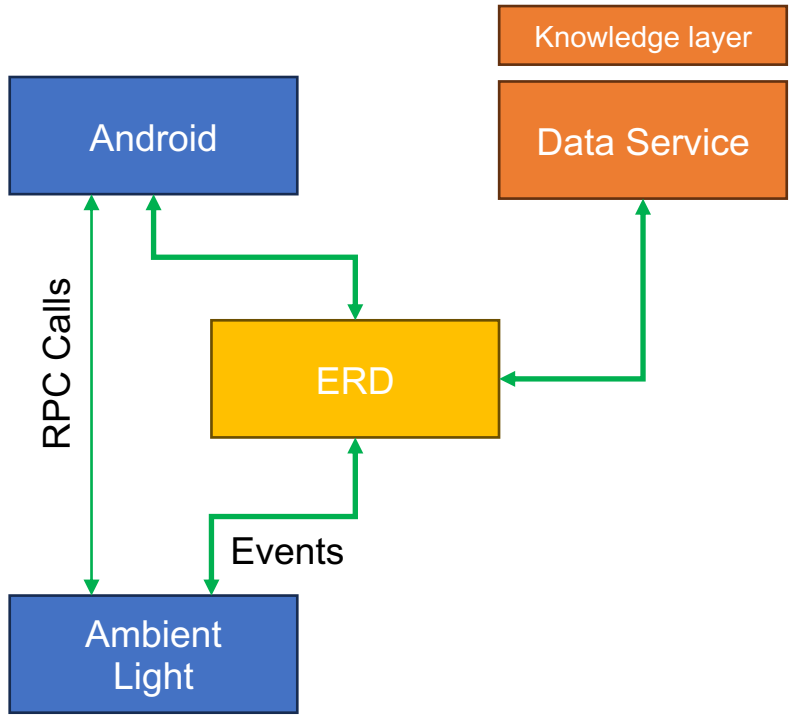



```yaml
openapi: 3.0.0
info:
  title: Vehicle Cabin Ambient Light API
  version: 1.0.0
paths:
  /Vehicle/Cabin/InteriorLights/AmbientLight/Segments:
    get:
      summary: Get the list of segments
      responses:
        200:
          description: List of segments retrieved successfully
          content:
            application/json:
              schema:
                type: array
                items:
                  $ref: '#/components/schemas/Segment'
  /Vehicle/Cabin/InteriorLights/AmbientLight/setSegmentsPermanentColor:
    post:
      summary: Set permanent color for segments
      requestBody:
        required: true
        content:
          application/json:
            schema:
              type: object
              properties:
                segments:
                  type: array
                  items:
                    $ref: '#/components/schemas/Segment' # Reference to the Segment object
                Fading:
                  type: string
                  format: duration
                ColorSpace:
                  type: string
                  enum: [sRgb888Gamma22D65]
      responses:
        200:
          description: Color set successfully
        400:
          description: Bad request

  /Vehicle/Cabin/InteriorLights/AmbientLight/segments/{segmentId}/lamps:
```

Use generator to convert to protobuf.

Example 4/5 – Events



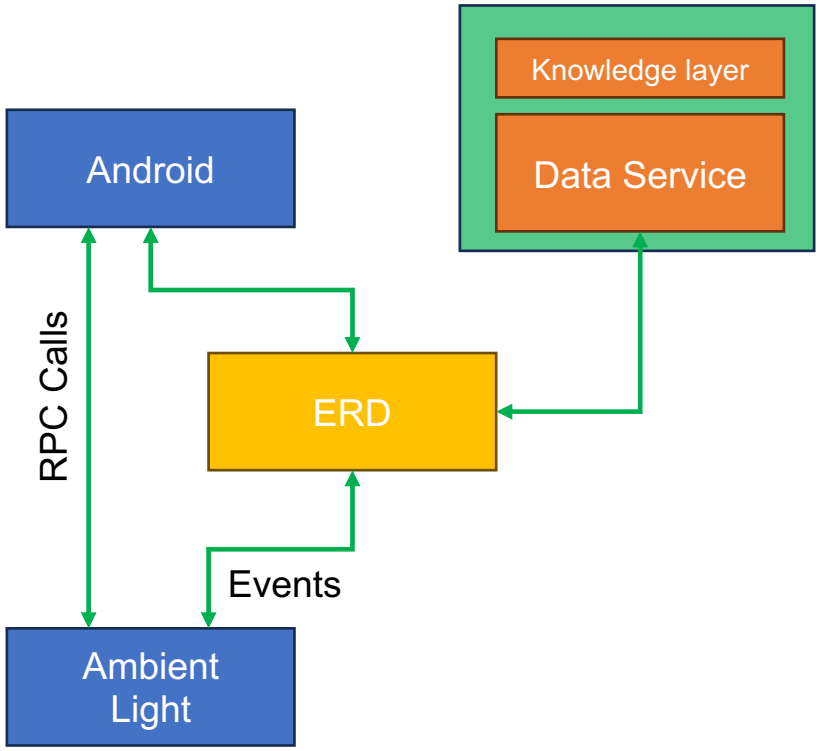

```yaml
asyncapi: '2.6.0'
info:
  title: Vehicle Cabin Ambient Light Notifications
  version: '1.0.0'
  description: |
    This service provides notifications related to ambient lighting in a vehicle cabin.

channels:
  Vehicle/Cabin/InteriorLights/AmbientLight/Segments/Row1/Left/Color:
    description: The topic for color change notifications on the left side of Row1.
    publish:
      summary: Color Change Notification for Row1 Left
      operationId: publishColorChangeForLeft
      message:
        $ref: '#/components/messages/Color'

  Vehicle/Cabin/InteriorLights/AmbientLight/Segments/Row1/Right/Color:
    description: The topic for color change notifications on the right side of Row1.
    publish:
      summary: Color Change Notification for Row1 Right
      operationId: publishColorChangeForRight
      message:
        $ref: '#/components/messages/Color'

  Vehicle/Cabin/InteriorLights/AmbientLight/Segments/Row1/Left/Status:
    description: The topic for status change notifications on the left side of Row1.
    publish:
```

Example 5/5 -HLA




```
{
  "_id": "Vehicle/Cabin/InteriorLights/AmbientLight/Segments/Row1/Left/Colo
  "description": "The topic for color change notifications on the left side
  "summary": "Color Change Notification for Row1 Left",
  "operationId": "publishColorChangeForLeft"
}
{
  "_id": "Vehicle/Cabin/InteriorLights/AmbientLight/Segments/Row1/
  "description": "The topic for color change notifications on the
  "summary": "Color Change Notification for Row1 Right",
  "operationId": "publishColorChangeForRight"
}
```

```
{
  "_id": "Color",
  "summary": "Notification of color change for Segment Row1",
  "payload": {
    "type": "object",
    "properties": {
      "color": {
        "type": "object",
        "properties": {
          "red": {
            "type": "integer",
            "minimum": 0,
            "maximum": 255
          },
          "green": {
            "type": "integer",
            "minimum": 0,
            "maximum": 255
```

```
{
  "_id": ObjectId("5ff5b6eae1f913a25c000001"),
  "timestamp": ISODate("2023-10-15T12:30:45Z"),
  "channelId": "Vehicle/Cabin/InteriorLights/AmbientLight/Segments/Row1/Lef
  "value": {
    "color": {
      "red": 150,
      "green": 50,
      "blue": 50
    }
  }
```

Questions?

COVESA

Accelerating the future of connected vehicles