Intro to Vehicle Signal Specification (VSS)
Introduction to Vehicle Signal Specification

ALL MEMBER MEETING

October 10-12, 2023
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.
What is VSS?

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

Vehicle.Drivetrain.Transmission.Speed

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

YAML SPECIFICATION
VEHICLE SIGNAL SPECIFICATION

Vehicle.Drivetrain.Transmission.Speed
- datatype: float
- unit: km/h
- description: The vehicle speed as measured by the drivetrain
VEHICLE SIGNAL SPECIFICATION

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
Today: Integration efforts are complex and unmaintainable.

- 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
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
Adopters
Relationship to Other Initiatives

<table>
<thead>
<tr>
<th>Initiative</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>W3C Vehicle Information Service Specification (VISS)</td>
<td>VSS is the data model used by W3C VISS</td>
</tr>
<tr>
<td>Vehicle Signal Specification Ontology (VSSo)</td>
<td>VSSo is directly correlated with the VSS catalog</td>
</tr>
<tr>
<td>Common Vehicle Interfaces</td>
<td>VSS will be a data model used in API definition of services.</td>
</tr>
<tr>
<td>Data Architecture</td>
<td>VSS as data model for Vehicle Domain.</td>
</tr>
</tbody>
</table>
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

In-cloud — In-Vehicle
# Resources

<table>
<thead>
<tr>
<th><strong>Resources</strong></th>
<th><strong>Links</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Documentation</strong></td>
<td><a href="https://covesa.github.io/vehicle_signal_specification/">https://covesa.github.io/vehicle_signal_specification/</a></td>
<td>Please read the documentation first. It answers most questions and points you to where to find other information.</td>
</tr>
<tr>
<td></td>
<td><a href="https://wiki.covesa.global/display/WIK4/VSS+Resources+at+a+Glance">https://wiki.covesa.global/display/WIK4/VSS+Resources+at+a+Glance</a></td>
<td></td>
</tr>
<tr>
<td><strong>Specification</strong></td>
<td><a href="https://github.com/COVESA/vehicle_signal_specification">https://github.com/COVESA/vehicle_signal_specification</a></td>
<td>To see specific nodes such as Cabin or Powertrain drill down into <a href="https://github.com/COVESA/vehicle_signal_specification/tree/master/spec">https://github.com/COVESA/vehicle_signal_specification/tree/master/spec</a></td>
</tr>
<tr>
<td><strong>Tools</strong></td>
<td><a href="https://github.com/COVESA/vss-tools">https://github.com/COVESA/vss-tools</a></td>
<td>Tools for building and processing VSS</td>
</tr>
<tr>
<td><strong>GitHub Issues and Wiki</strong></td>
<td><a href="https://github.com/COVESA/vehicle_signal_specification/issues">https://github.com/COVESA/vehicle_signal_specification/issues</a></td>
<td>The team uses GitHub Issues and Wiki heavily</td>
</tr>
<tr>
<td></td>
<td><a href="https://github.com/COVESA/vehicle_signal_specification/wiki">https://github.com/COVESA/vehicle_signal_specification/wiki</a></td>
<td></td>
</tr>
<tr>
<td><strong>Vehicle Signal Specification ontology</strong></td>
<td><a href="https://www.w3.org/TR/vsso/">https://www.w3.org/TR/vsso/</a></td>
<td>Ontology built with VSS</td>
</tr>
</tbody>
</table>
Most common challenges are Fragmentation, Usability, Integration (Test and Validation)

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

Touchpoints
- Apple CarPlay
- AOSP / AAOS
- Containers
- Cloud

Services
- POSIX
- sPOSIX
- OSEK

Example 1/5 – HLA
Example 2/5 - HLA

Android → ERD → Data Service

ERD → Knowledge layer

Ambient Light → Events → RPC Calls

ERD
Example 3/5 – “RPC”

Use generator to convert to protobuf.
Example 4/5 – Events

Android

Data Service

Knowledge layer

ERD

Ambient Light

RPC Calls

Events

```
asyncapi: '2.6.8'
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/Color",
    "description": "The topic for color change notifications on the left side",
    "summary": "Color Change Notification for Row1 Left",
    "operationId": "publishColorChangeForLeft"
}

{
    "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("5ff5b0eae1f913a25c000001"),
    "timestamp": ISODate("2023-10-15T12:30:45Z"),
    "channelId": "Vehicle/Cabin/InteriorLights/AmbientLight/Segments/Row1/Left",
    "value": {
        "color": {
            "red": 150,
            "green": 50,
            "blue": 50
        }
    }
}
```
Questions?