

# Connected Services: W3C VISS/VSI specification status



This is a historical page. As part of preparation work for the CCS project and the initial stages of working towards a unified approach to vehicle data, all related projects were surveyed and this was the "snapshot" view at the time. By now these projects have progressed, and we have continued convergence and collaboration in this area. For up to date status of the VISSv2 and other related projects in W3C and elsewhere, we recommend starting at the [CVII initiative](#).

Links to readable specifications (from the README in the [W3C Automotive & Transportation WG GitHub repository](#))

- [Vehicle Information Service Specification version 2 - CORE](#) (V2 git master, current working copy)
- [Vehicle Information Service Specification version 2 - TRANSPORT](#) (V2 git master, current working copy)
- [Vehicle Information Service Specification](#) (V1, released but being replaced by V2)



## Snapshot status (2019-06-24)

### Introduction

As part of the investigation of all relevant standards and technologies in the [Vehicle Connected Services project](#), this is a brief summary of the current (2019-06-24) state of the W3C Automotive working group standard for Vehicle Services Interface specification (formerly VISS), version 2.0.

The **VISS** version 1.0 is still in Candidate Recommendation status. The **version 2.0** specification is currently named Vehicle Services Interface (VSI?). Version 2 is currently being defined and often referred to inside group discussions simply as "**Gen 2**".

GENIVI Alliance participates actively in the W3C specification definition, in part through shared member companies, but also with GENIVI representatives (GENIVI & W3C have a collaboration agreement and GENIVI is set up as an official member of W3C).

### Two-part Specification

The W3C specification is split into two parts: Core and Transport. The Core specification "*describes the VSI messaging layer*" whereas the Transport specification includes the "*mapping of the messaging layer to selected transports*".

### Content

NOTE: The intention here is to give a current snapshot, not to (re)-describe the group charter or intended scope of the final version 2.0 specification.

---

## 1) The Core specification includes



*Describing current status, not planned scope*

- General explanation of the underlying "Data Model" (**NOTE:** The terms **data model**, **taxonomy**, and **related terms** are [currently being discussed](#) in the working group and formal definition created, to avoid confusion that seems to happen between participants during discussions)
- The data model is described as a *tree-like logical taxonomy of the whole vehicle. It is represented by a directed acyclic graph*
- The current structure of the [Vehicle Signal Specification \(VSS\)](#) including branches and rbranches (resource branches), attributes, and signals are used as the illustration. *There are unfinished discussions in the group to support some alternative databases of signals.*
- Short description of **search** mechanisms: **Wildcards** (defined) and **Queries** (to be defined)
- General description of Create/Read/Update/Delete (**CRUD principle**), plus **Subscribe/Unsubscribe**.
- **Authorization** (very brief)

Note about current status:

Clarification of the split and formal definition of data taxonomy and data model ongoing. It is slightly unclear about the final intention and *how* to define one or several "mandated" data model content (e.g. VSS) and possible alternative data descriptions (mandated, optional, proprietary/private or unknown to others).

More to do to define service discovery (which is a wanted feature).

The chapters regarding Security seem to be not yet written, except for the brief Authorization note.

---

## 2) The Transport specification includes:

⚠ *Describing current status, not planned scope*

- Status/error codes (following HTTP standards)
- "The payload **shall** have **JSON format**."
- Two protocols noted so far: **HTTPS** (the main driver for VISS v2) and "**Secure Websockets**" (like VISSv1)
- **Wildcards** and **Queries**
- The HTTPS chapter includes examples how to access data in the described data tree, such as:

```
GET VSS-Root/Car/Drivetrain/InternalCombustionEngine/RPM HTTP/1.1
```

- ...and Wildcards:

```
/VSS-Root/Cabin/Door/**/IsLocked HTTP/1.1
```

⚠ **NOTE:** Some discussions whether wildcards should be completely replaced by new Queries approach.

Note about current status:

The appendix defining the exact JSON format is not yet written. (VISS v1 standards could be reused as basis. There are likely other options.) Discussions on other protocols tend to pop up from time to time (think for example MQTT). But should it be in scope?

---