

SENSORIS Specification

The **Sensor Interface Specification (SENSORIS)** defines an interface for requesting and sending vehicle sensor data from vehicles to clouds and across clouds. The specification and its standardization focus on the content and encoding of the interface.

For more information please visit the [SENSORIS webpage](#).

Change Log

SENSORIS v1.5.1

Bugfixes:

- In the `LocalizationCategory` the `VehicleSpeed` is fixed to set the vehicle speed to negative if the vehicle drives backwards.

SENSORIS v1.5.0

New enums have been added:

- Additional lane boundary types were added to `LaneBoundary` in `RoadAttributionCategory`:
`LINE_DOUBLE_DASHED_OR_DOUBLE_SOLID_DASHED`, `LINE_DOUBLE_DASHED_OR_DOUBLE_DASHED_SOLID`,
`LINE_DOUBLE_SOLID_OR_DOUBLE_SOLID_DASHED`, and
`LINE_DOUBLE_SOLID_OR_DOUBLE_DASHED_SOLID`.

A bugfix was necessary:

- Fix documentation of TrafficSign relations (left-over line was deleted).

SENSORIS v1.4.0

New enums have been added:

- In the `ObjectDetectionCategory` three new values were added to the `StaticObject` type:
`TRAFFIC_ENFORCEMENT_CAMERA`, `TRAFFIC_ENFORCEMENT_CAMERA_FIXED`, and
`TRAFFIC_ENFORCEMENT_CAMERA_MOBILE`.
- Static objects of type `TRAFFIC_ENFORCEMENT_CAMERA_MOBILE` which are hand-held can be related to a movable object of type `PERSON`.
- Static objects of type `TRAFFIC_ENFORCEMENT_CAMERA_MOBILE` which are vehicle-mounted can be related to a movable object of type `VEHICLE` or one of its sub-types.
- Additional lane boundary types were added to `LaneBoundary` in `RoadAttributionCategory`.

A few bugfixes were necessary:

- The unit of the base type `xyzVectorAndAccuracy` depends on the usage in a composite type. For the resolution this was described in the architecture document, but not for unit and range.

- Fix documentation of resolution of `RoadAttribution` > `InclinationAndCurvature` > `horizontal_curvature_and_accuracy`.
- Description for `THREE_D` for `NavigationSatelliteSystemStatus` of `VehiclePositionAndOrientation` is fixed.

SENSORIS v1.3.0

Some events have been added:

- Added the `VehicleDeviceCategory` to the SENSORIS `DataMessage`.
- In the `VehicleDeviceCategory` the `ExteriorLightStatus` event and `WiperStatus` events for the front and rear wiper have been added.
- In the `TrafficEventsCategory` the `ECallStatus` event has been added.

In addition package names for the Google Go programming language have been added throughout the specification.

A few bugfixes were necessary:

- For the states of a Job the UML state diagram in the interface architecture has been fixed to allow for termination from inactive state.
- Clarification of the spatial reference system for the `RotationAndAccuracy` type.

SENSORIS v1.2.0

In context of the EU Data Task Force several hazard events have been added:

- In the `TrafficEventsCategory` the `Hazard` event has been extended and the `RoadSurfaceCondition`, `RoadObstructionCondition` and `VulnerableRoadUserCondition` have been added.
- In the `WeatherCategory` the `AtmosphereCondition`, `WindCondition`, and `VisibilityCondition` have been added.
- In the `EventEnvelope` the `DetectionType` has been added, which is available for all events.

In addition, enumeration values have been extended for the `StaticObject` event in the `ObjectDetectionCategory`, for the `MarkingColorAndConfidence` in the `RoadAttributionCategory`, and for the `TrafficSign` event in the `TrafficRegulationCategory`.

SENSORIS v1.1.1

A few bugfixes were necessary:

- In the interface architecture document several minor issues have been fixed.
- The enumeration value `OTHER` has been deprecated affecting several events, as it complicates compatibility between versions.
- The value range for the relative speed of the `MoveableObject` event in the `ObjectDetectionCategory` has been fixed to allow also negative values.

- The example for the `AbsolutePaths` in the `DataMessage` has been fixed to point to the correct event.

SENSORIS v1.1.0

The Request Channel has been added. Job Messages are defined so that the cloud can request jobs to the vehicle. This version allows closing the loop, from Job Request to Data Messages upload.

Proto files corresponding to the Job Request Channel:

```
1  -- messages/
2      - job.proto
3  -- types/
4      - collection.proto
5      - job.proto
6      - job_validity.proto
7      - logical_expresion.proto
8      - map.proto
```

Some backwards compatible additions have been made with respect SENSORIS v1.0.0:

```
1  -- messages/
2      - data.proto
3  -- types/
4      - base.proto
5      - spatial.proto
```

SENSORIS v1.0.0

Focuses on the definition of Data Messages to transmit vehicle sensor data from the vehicle to the cloud.

Proto files corresponding to the Data Messages:

```
1  -- messages/
2      - data.proto
3  -- types/
4      - base.proto
5      - source.proto
6      - spatial.proto
7  -- categories/
8      - brake.proto
9      - driving_behavior.proto
10     - intersection_attribution.proto
11     - localization.proto
12     - map.proto
13     - object_detection.proto
14     - powertrain.proto
15     - road_attribution.proto
16     - traffic_events.proto
17     - traffic_maneuver.proto
```

```
18 | - traffic_regulation.proto
19 | - weather.proto
```

Protobuf

The SENSORIS specification uses [Google Protocol Buffers](#) (protobuf) version 3 for message encoding. The SENSORIS protobuf schemas are located in the src directory.

To compile the SENSORIS protobuf schemas for target languages, e.g. C++ and Java, some steps are required which are described below for a Linux based system.

First, download the [protoc compiler](#) for the target system and unzip it.

```
1 | wget https://github.com/google/protobuf/releases/download/v3.5.1/protoc-3.5.1-linux-
  | x86_64.zip
2 | unzip protoc-3.5.1-linux-x86_64.zip -d protobuf
```

Then clone the SENSORIS specification from GitHub and checkout the desired Git branch, e.g. feature/localization-category.

```
1 | git clone https://github.com/sensoris/specification.git
2 | cd specification
3 | git checkout feature/localization-category
4 | cd ..
```

The protobuf compiler requires that the output directories for the generated sources exist, e.g. for C++ and Java.

```
1 | mkdir specification-cpp specification-java
```

Finally, run the protobuf compiler to generate the sources for C++ and Java. For protoc all protobuf files have to be listed individually, which is automatically done by the find command.

```
1 | protobuf/bin/protoc --cpp_out=specification-cpp --java_out=specification-java --
  | proto_path=specification/src $(find specification -name '*.proto' -printf "%p ")
```

For reference the expanded version of the protobuf compiler call is listed below.

```
1  protobuf/bin/protoc --cpp_out=specification-cpp --java_out=specification-java --  
    proto_path=specification/src specification/src/sensoris/protobuf/types/base.proto  
    specification/src/sensoris/protobuf/types/source.proto  
    specification/src/sensoris/protobuf/types/spatial.proto  
    specification/src/sensoris/protobuf/messages/data.proto  
    specification/src/sensoris/protobuf/categories/traffic_regulation.proto  
    specification/src/sensoris/protobuf/categories/map.proto  
    specification/src/sensoris/protobuf/categories/brake.proto  
    specification/src/sensoris/protobuf/categories/weather.proto  
    specification/src/sensoris/protobuf/categories/intersection_attribution.proto  
    specification/src/sensoris/protobuf/categories/traffic_events.proto  
    specification/src/sensoris/protobuf/categories/traffic_maneuver.proto  
    specification/src/sensoris/protobuf/categories/localization.proto  
    specification/src/sensoris/protobuf/categories/driving_behavior.proto  
    specification/src/sensoris/protobuf/categories/powertrain.proto  
    specification/src/sensoris/protobuf/categories/object_detection.proto  
    specification/src/sensoris/protobuf/categories/road_attribution.proto
```

The generated source files for C++ and Java are created in the respective directories, e.g. specification-cpp and specification-java.