

SENSORIS - Study Summary

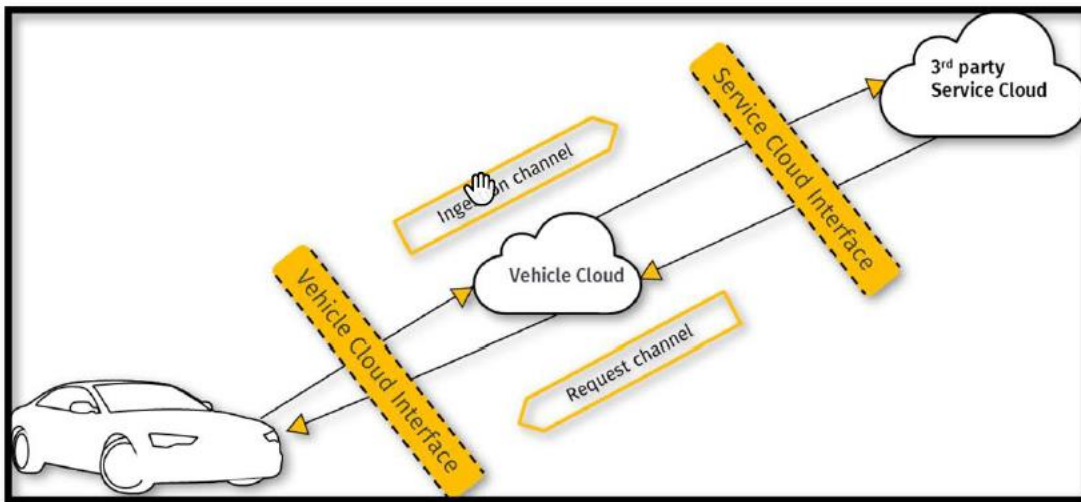
(Draft)

Contents

- Overview**2
- Background Info :**3
- Scope**3
 - in scope3
 - out of scope3
- Messages**4
 - Data Format Structure4
 - DataMessage envelope4
 - Event Relationship types5
 - Data Encoding6
 - Weather Category6
 - IntersectionAttribution Category7
 - Build and Src Folder structure8
- Members**9
- Working Groups**9
- References** 10

Overview

Sensor Interface Specification Innovation Platform, SENSORIS, is an open group of significant actors from the global vehicle industry, map and data providers, sensors manufacturers and telecom operators who joined forces, under the form of this Innovation Platform, driven by the common vision and belief.



SENSORIS specify the interface for exchanging information between in vehicle sensors and dedicated cloud as well as between clouds to:

- enable broad access, delivery & processing of vehicle sensor data
- enable easy exchange of vehicle sensor data between all players
- enable enriched location based services and automated driving

Main objective is to deliver and maintain technical specifications defining the format and content of sensor and campaign data in the cases mentioned in the in-scope section:

- vehicle-to-cloud data upload format (vehicle-based data only)
- cloud-to-cloud data exchange format (vehicle-based data and other data needed for mobility services)
- cloud-to-vehicle 'campaign' request format (request for specific data at specific locations and times only)

The specification and its standardization focus on the content and encoding of the interface.

The documentation of the SENSORIS schema is part of the protobuf schema itself, i.e. schema definition and documentation are located together. Documentation is written as protobuf comments. The comments in the protobuf schema are taken over automatically to the auto-generated data classes by the protobuf compiler.

Background Info :

- 06/2015: First specifications released by HERE
- 06/2016: SENSORIS platform created & coordinated by ERTICO with major industrial stakeholders
- 12/2017: Draft of new specifications ready
- 06/2018: New specifications as de-facto industrial standard.

Scope

in scope

- vehicle-to-cloud data upload format (vehicle-based data only)
- cloud-to-cloud data exchange format (vehicle-based data and other data needed for mobility services)
- cloud-to-vehicle 'campaign' request format (request for specific data at specific locations and times only)
- conformance to data authorization/authentication process
- conformance to data privacy regulations
- conformance to approved security regulation (N.B. 'cloud' can be an intermediate server or aggregation server or a service provider input gateway)

out of scope

SENSORIS will not:

- define infrastructure or architecture
- establish commercial agreement frameworks for data exchange
- define data exchange for v2v, v2i, i2v (cooperative data) exchange
- define cloud-to-vehicle services

Messages

Data Format Structure

Data format has:

- Events (vehicle data, localization, road condition, weather, traffic signs, ...)
- Sources (sensors like GNSS or Camera, sensor fusion like algorithms)
- Relations between Events and Events or Events and Sources

Flexible data content according to use cases:

- Single event content (e.g. 1x Position, 1x Speed)
- Full drive data (e.g. 5h of position, speed, traffic signs, road attribution and weather)

DataMessage envelope

Datamessage envelope is defined to ids (follows google protobuf string, int etc..), field resolution, sensoris vehicle co-ordinate system, map_identification (If the map of the vehicle is known to the cloud, then the relations can be interpreted by the cloud.), events to map objects, priority extensions

Field	Type	Description
ids	DataMessage.Envelope.Ids	Identifiers.
field_resolution_override	repeated DataMessage.Envelope.FieldResolutionOverride	Field resolution override.
vehicle_dimensions	DataMessage.Envelope.VehicleDimensions	Vehicle dimensions in reference to SENSORIS vehicle coordinate system.
map_identification	DataMessage.Envelope.MapIdentification	Map identification. The combination of the map identification and event relations of type MAP_REFERENCE defined in the data message enable to relate events to map objects defined by map references in the map category. If the map of the vehicle is known to the cloud, then the relations can be interpreted by the cloud.
extension	repeated google.protobuf.Any	Proprietary extension.

One event can contain

Field	Type	Description
envelope	EventGroup.Envelope	Envelope.
localization_category	sensoris.protobuf.categories.localization.LocalizationCategory	Localization category.
object_detection_category	sensoris.protobuf.categories.objectdetection.ObjectDetectionCategory	Object detection category.
weather_category	sensoris.protobuf.categories.weather.WeatherCategory	Weather category.
driving_behavior_category	sensoris.protobuf.categories.drivingbehavior.DrivingBehaviorCategory	Driving behavior category.
intersection_attribution_category	sensoris.protobuf.categories.intersectionattribution.IntersectionAttributionCategory	Intersection attribution category.
road_attribution_category	sensoris.protobuf.categories.roadattribution.RoadAttributionCategory	Road attribution category.
traffic_regulation_category	sensoris.protobuf.categories.trafficregulation.TrafficRegulationCategory	Traffic regulation category.
traffic_events_category	sensoris.protobuf.categories.trafficevents.TrafficEventsCategory	Traffic events category.
traffic_maneuver_category	sensoris.protobuf.categories.trafficmaneuver.TrafficManeuverCategory	Traffic maneuver category.
brake_category	sensoris.protobuf.categories.brake.BrakeCategory	Brake category.
powertrain_category	sensoris.protobuf.categories.powertrain.PowertrainCategory	Powertrain category.
map_category	sensoris.protobuf.categories.map.MapCategory	Map category.

Event Relationship types

☰ **DataMessage.EventRelation.Type**


Type.

Name	#	Description
UNKNOWN_TYPE	0	Unknown.
EQUAL	1	Same real world entity described through to and from events.
GROUP	2	Accumulation of from events without to relation.
CONTAIN	3	Accumulation of from events to whole to events.
IMPACT	4	To events add context to or enrich the from events.
START_AT	5	From events start at the to events, e.g. for spatial or temporal relations.
END_AT	6	From events end at the to events, e.g. for spatial or temporal relations.
LEFT_OF	7	From events are spatially left of the to events in the context of the spatial reference system.
RIGHT_OF	8	From events are spatially right of the to events in the context of the spatial reference system.
TRIGGERED_BY	9	From events are consequences of the to events.
DERIVED_FROM	11	To events are derivations or interpretations of the from events.
MAP_REFERENCE	12	From events reference to objects from a map.

Data Encoding


- Google protobuf v3
 - Efficient encoding
 - Language support
 - Field Options – variable resolution interpretation
 - Version compatibility
 - Including documentation generator
- Individual resolution and variable bit encoding

Weather Category

 WeatherCategory

Weather category.


Field	Type	Description
envelope	sensoris.protobuf.types.base.CategoryEnvelope	Envelope.
precipitation	repeated Precipitation	Precipitation.


 Precipitation.TypeAndConfidence.Type

Type.

Name	#	Description
UNKNOWN_TYPE	0	Unknown.
NONE	1	No precipitation.
RAIN	2	Rain.
MIXED_RAIN_SNOW	3	Mixed rain and snow.
SNOW	4	Snow.
HAIL	5	Hail, sleet or freezing rain.

IntersectionAttribution Category


 **sensoris/protobuf/categories/intersection_attribution.proto**
sensoris.protobuf.categories.intersectionattribution

 **IntersectionAttributionCategory**

Intersection attribution category.

Field	Type	Description
envelope	sensoris.protobuf.types.base.CategoryEnvelope	Envelope.
traffic_signal_bulb	repeated TrafficSignalBulb	Bulb of a traffic signal, i.e. the individual traffic light.
traffic_signal	repeated TrafficSignal	Traffic signal of one or more traffic lights for an intersection scenario. Traffic signals also might be part of no intersection, e.g. on pedestrian crossings.

Any

 **Any**

Any contains an arbitrary serialized protocol buffer message along with a URL that describes the type of the serialized message.

Protobuf library provides support to pack/unpack Any values in the form of utility functions or additional generated methods of the Any type.

The pack methods provided by protobuf library will by default use 'type.googleapis.com/full.type.name' as the type URL and the unpack methods only use the fully qualified type name after the last `.` in the type URL, for example "foo.bar.com/x/y.z" will yield type name "y.z".

Field	Type	Description
type_url	string	A URL/resource name whose content describes the type of the serialized protocol buffer message.
value	bytes	Must be a valid serialized protocol buffer of the above specified type.

The categories for other ECU's signals is grouped under that domain/ECU

e.g.

sensoris/protobuf/categories/powertrain.proto

sensoris.protobuf.categories.powertrain

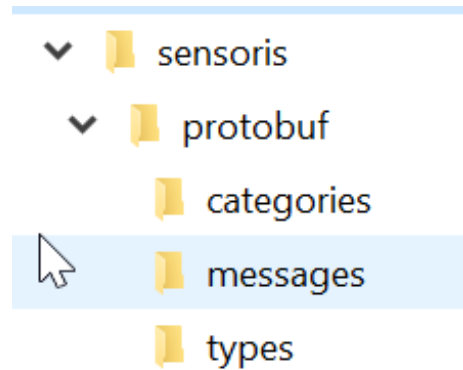
e.g. engine status, engine speed, transmission status etc.

Build and Src Folder structure

The source and documentation is available in [github Reference\[2\]](#)

The sources are grouped under Category, Messages and types following the Google protobuf format.

The files has .proto extension.



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

```
```bash
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.

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

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

Members

| Main membership category | SENSORIS member | Main membership category | SENSORIS member |
|--|----------------------------|---------------------------------|--------------------------------------|
| Vehicle manufacturers (6) | Audi | ADAS manufacturers (5) | AISIN AW |
| | BMW AG | | Continental Automotive GmbH |
| | Daimler AG | | DENSO Ten (Europe) GmbH |
| | Jaguar Land Rover Limited | | LG Electronics |
| | Nissan | | Valeo Comfort and Driving Assistance |
| | Volvo Car | | |
| Location content & Service providers (7) | AutoNavi Software Co. Ltd. | Navigation System Suppliers (7) | DENSO |
| | Baidu | | Elektrobit Automotive GmbH |
| | HERE Global B.V. (Chair) | | Harman |
| | INRIX Inc. | | Hyundai Mnsoft |
| | NavInfo Co.Ltd. | | NNG |
| | TomTom International B.V. | | PIONEER Co. |
| | Zenrin | | Robert Bosch GmbH |
| Telecom & Cloud Infrastructure Providers (1) | IBM | Other (2) | ICCS
CTAG |

Working Groups

The SENSORIS platform work is organised in four Working Groups (WG):

| WG# | Working Group title and short description |
|------------|---|
| WG1 | <p>Needs & requirements
 Leader = HERE
 Alignment of existing Sensor Data Standards and summarizing the needs of content and details of sensor data from the LBS provider view.
 This is composed of but not restricted to:</p> <ul style="list-style-type: none"> - List of Sensor Data Attributes, Classified by topics - Meta-Requirements on information per attribute, as e.g. quality metrics <p>Requirements on sensor data representation</p> |
| WG2 | <p>Availability of sensor data (OEM/suppliers)
 Leader = Daimler
 Gathering of available in-vehicle sensor data including their specific properties</p> |
| WG3 | <p>Interface architecture and high level design
 Leader = Elektrobit
 Overall architecture and workflow of data between vehicle and cloud (and cloud to cloud). Proposed communication paradigms and security approach.</p> |
| WG4 | <p>Interface specifications
 Leader = Continental
 Provide Data Definitions for Sensor Data including Quality Metrics and Location References according Requirements from WG1.
 Provide Data Definition for Data Request Channel.
 Starting point is the existing Draft SENSORIS Specification.</p> |

References

| Sl No | Description | Link / source |
|-------|---------------|---|
| [1] | Sensoris | https://sensor-is.org/ |
| [2] | Github source | https://github.com/sensoris/specification.git |