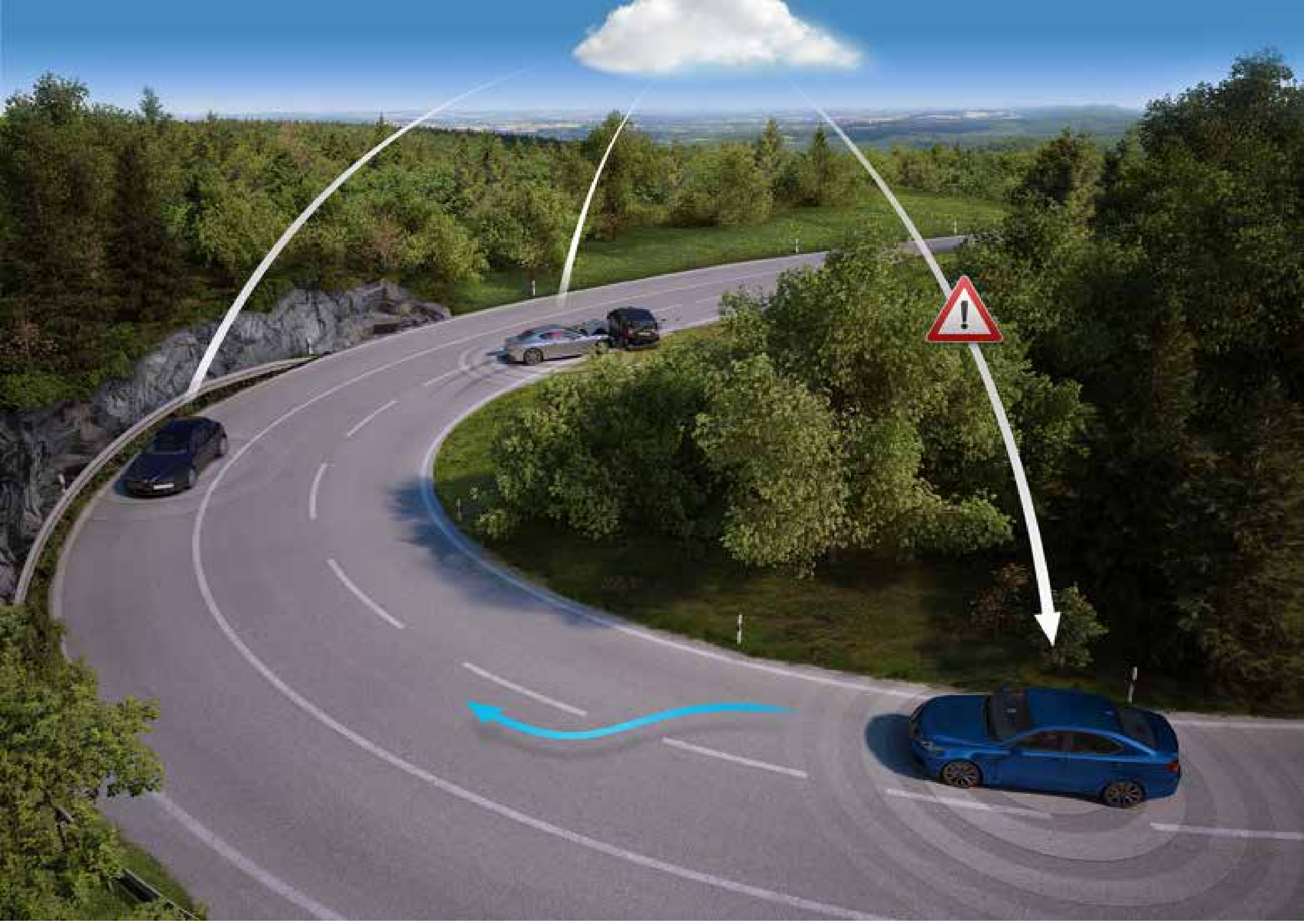

Vehicle Sensor Data Cloud Ingestion Interface Specification (v2.0.2)

Table of contents

1. General Considerations

Vehicles driving on the road are equipped with a magnitude of sensors. These Sensor Data may be transferred over any kind of technology from the vehicle to an Analytic Processing Backend. Between individual vehicles and the Analytic Processing Backend, an OEM-, or System Vendor-Backend may be located as a proxy. The Sensor Data Interface Specification defines the content of Sensor Data Messages and their encoding format as they are submitted to the Analytic Processing Backend. However, the specification may be used between other components as well.

Sensor Data are submitted as messages with various type of content. Common to all kind of submitted messages is that they are related to one or multiple locations. Sensor Data Messages may be time critical and submitted near real time but also may be of informational value and submitted with an acceptable delay accumulated within other Data. Neither priority, nor requirements on latency are part of the Specification.

The content and format of the Sensor Data Messages is independent of the submission latency (in near real time or delayed).

2. Data Elements

All data elements have units according to the International System of Units (SI) unless otherwise explicitly noted.

2.1 General

Data Types within this paragraph describe multiply used base and complex types.

2.1.1 Timestamp

A timestamp is used to indicate an absolute time value. All timestamps used in any of the sensor ingestion API are based on UTC and measured in milliseconds since January 1, 1970. Please note that in the context of sensor data collection a GPS time may be used during data collection. GPS time is different from UTC. At the time of this writing, the GPS time is 16 seconds ahead of UTC. It is the responsibility of the data submitter to convert any time stamp into UTC before data submission. The timestamp does not count in leap seconds. Any necessary transformation is within the responsibility of the data submitter.

Name	timeStampUTC_ms
Unit	ms (= seconds/1.000)
Range	[0; ∞ [
Resolution	1ms

It is an absolute requirement that the timestamp is consistent at all times throughout all event messages. Hereby, the timestamp represents the point of time at which the sensor data has been captured and not the time at which the sensor data is available. By way of example, a position estimate of at time t=0 is made available at t=1 and a map matching algorithm provides a matched position at t=5, then the map matched position estimate is provided as sensor data at t=6 with the data content “t=0”.

2.1.2 Position Offset

The Position Offset describes a position delta from the vehicle reference point as defined in the vehicle metadata.

Name	Position Offset

Elements:

Data type	Element	Mandatory or Optional
double	lateralOffset_m	optional
LateralOffsetSimpleEnum	lateralOffsetSimple	optional
double	longitudinalOffset_m	optional
LongitudinalOffsetSimpleEnum	longitudinalOffsetSimple	optional
double	verticalOffset_m	optional
VerticalOffsetSimpleEnum	verticalOffsetSimple	optional
double	lateralOffsetAccuracy_m	optional
double	longitudinalOffsetAccuracy_m	optional
double	verticalOffsetAccuracy_m	optional

2.1.3 Wheel reference Bitfield

A bitfield referencing a combination of the 4 Wheels. A vehicle equipped with more than 3 axis is referencing the first (front) and the last (rear) axis. A vehicle equipped with a trailer does reference to the towing vehicle, only.

Name	Position Offset

Elements:

Bitfield identifier	Bit	Description
frontAxleLeft	0	Referencing the front left wheel
frontAxleRight	1	Referencing the front right wheel
rearAxleLeft	2	Referencing the rear left wheel
rearAxleRight	3	Referencing the rear right wheel

2.1.4 Textual Key-Value Pairs

Parts of the interface are using generic key value pairs. Hence, a data type is defined that allows such key value pairs.

Name	KeyValuePairString

Elements:

Bitfield identifier	Element	Mandatory or Optional
string	key	Mandatory
string	value	Mandatory

2.1.5 Vector3D

A 3D-vector containing double values. Vector3D is used for anything requiring a three dimensional vector with optional accuracy, e.g. an acceleration-vector. According to the ENU (East-North-Up) System, a value is positive in right, up, front direction and negative in left, down, rear direction. When representing rotations then the value represents the rotation along the named axis in angular speed with the rotation in mathematical positive direction facing the positive end of the axis is represented by a positive value. From the vehicles reference point, increasing any dimension of a rotation vector results in "yaw to the right" (heading), "roll to the left" (cross-slope) and "pitching to the front" (slope).

Longitudinal value always refers to the front-rear-axis in longitudinal direction, transversal value refers to the left-right axis, and vertical refers to the up-down axis.

Data type	Element	Mandatory or Optional
double	longitudinalValue	mandatory
double	transversalValue	mandatory
double	verticalValue	mandatory
double	longitudinalAccuracy	optional
double	transversalAccuracy	optional
double	verticalAccuracy	optional

2.2 Envelope

2.2.1 Version

A textual value describing the version of interface specification in use. For example: "1.0"

Name	Version
Values	a textual value

2.2.2 Submitter

An OEM specific information that describes the submitter. It is the name of the company that submits the data (e.g. an OEM name or a System Vendor name). It is not an individual vehicle identifier. The same company may use different values for the submitter text value for different purposes (e.g. production group, pre-development group).

Name	Submitter
Values	a textual value

2.2.3 Vehicle Meta Data

An OEM specific information that describes the submitter. It is the name of the company that submits the data (e.g. an OEM name or a System Vendor name). It is not an individual vehicle identifier. The same company may use different values for the submitter text value for different purposes (e.g. production group, pre-development group).

Name	VehicleMetaData

Elements:

Data type	Element	Mandatory or Optional
VehicleTypeGenericEnum	vehicleTypeGenericEnum	Mandatory
KeyValuePairString	vehicleSpecificMetaData	Repeated element (which allows zero entries)
double	vehicleReferencePointDeltaAboveGround_m	Mandatory
double	vehicleLength_m	Optional
double	vehicleWidth_m	Optional
double	vehicleHeight_m	Optional
double	primaryFuelTankVolume	Optional
FuelTypeEnum	primaryFuelType	Optional (Mandatory if FuelTank is provided)
double	secondaryFuelTankVolume	Optional
FuelTypeEnum	secondaryFuelType	Optional (Mandatory if FuelTank is provided)

2.2.4 Transient Vehicle ID

The Transient Vehicle ID is a numeric and optional value. If the transient vehicle ID value is submitted then each submission during the drive cycle of a single vehicle shall receive the same transient vehicle ID value. This allows stitching together multiple smaller path submissions of a vehicle during a drive. A different drive of the same vehicle (e.g. on the next day) should receive a new transient vehicle ID. Within all submissions of a "Submitter" the ID space shall remain unique.

Name	transientVehicleID
Values	An integer value

2.2.5 Vehicle Profile ID

The Vehicle Profile ID is a numeric and optional value that is unique for a vehicle. If the vehicle Profile ID value is submitted then each submission including the profile ID is identifiable with the vehicle. This allows stitching together multiple single submissions of a vehicle during over a multitude of drives. The vehicle profile ID should be provided to events as e.g. fuel events where the fuel profile for a single vehicle should be analyzed and provided back to the single vehicle through a different interface.

Name	vehicleProfileID
Values	An integer value

2.3 Path

Path is the mandatory logical data type containing one or more position estimates of the vehicle ordered by timestamp. The array of Position Estimates contains linear interpolateable locations.

2.3.1 Position Estimate

The position estimate data type combines information related to the position of a vehicle.

Name	PositionEstimate

Elements:

Data type	Element	Mandatory or Optional
int64	timeStampUTC_ms	Mandatory
PositionTypeEnum	positionType	Mandatory
boolean	interpolatedPoint	optional
double	longitude_deg	Mandatory
double	latitude_deg	Mandatory
double	horizontalAccuracy_m	Mandatory
double	altitude_m	optional
double	heading_deg	optional
double	speed_mps	optional
double	altitudeAccuracy_m	Optional, Mandatory if Altitude_m is presenttude_m is present
double	headingAccuracy_deg	Optional, Mandatory if Heading_deg is present
double	speedAccuracy_mps	Optional, Mandatory if Speed_ mps is present
SpeedDetectionEnum	speedDetectionType	Optional
HeadingDetectionEnum	speedDetectionType	Optional
Int	currentLaneEstimate	Optional

2.4 Path Events

This complex Data Type contains necessary information to notify about detected events that the vehicle or OEM deduces from their raw sensor data. Every element in Path Event is a complex Data Type described in their respective paragraphs. The order within each list of Path Event Types is by timestamp ascending.

Data type	Element	Mandatory or Optional
VehicleStatus	vehicleStatus	Repeated
VehicleDynamics	vehicleDynamics	Repeated
SignRecognition	signRecognition	Repeated
LaneBoundaryRecognition	laneBoundaryRecognition	Repeated
ExceptionalVehicleState	exceptionalVehicleState	Repeated
ProprietaryInfo	proprietaryInfo	Repeated
EnvironmentStatus	environmentStatus	Repeated
ObjectDetection	objectDetection	Repeated
ADServiceAndSensorState	adServiceAndSensorState	Repeated

2.4.1 Vehicle Status

The vehicle status data type combines information related to the status of a vehicle

Name	VehicleStatus

Elements:

Data type	Element	Mandatory or Optional
int64	timeStampUTC_ms	Optional
TransmissionModeEnum	transmissionMode	Optional
LightStateBitfield	lightStateBitfield	Optional
WiperStateEnum	wiperState	Optional
WiperSpeed	wiperSpeed_wpm	Optional
WheelReferenceBitfield	driveWheelReference	Optional
double	chassisClearance_m	Optional
double	mileage	Optional
double	primaryFuelState	Optional
double	primaryFuelStateAccuracy	Optional
Int32	estimatedPrimaryRange_km	Optional
double	secondaryFuelState	Optional
double	secondaryFuelStateAccuracy	Optional
Int32	estimatedSecondaryRange_km	Optional

2.4.2 Vehicle Dynamics

The vehicle dynamics data type combines information related to the dynamic movements of the vehicle.

Name	VehicleDynamics

Elements:

Data type	Element	Mandatory or Optional
int64	timeStampUTC_ms	Mandatory
double	curvature_1pm	Optional
double	slope_percent	Optional
double	springTravelRange_m	Optional
double	springTravelAccuracy_m	Optional
Vector3D	averageGravityVector	Optional

2.4.3 Sign Recognition

The combination of information used to report a recognized (or not detected) sign.

Name	SignRecognition

Elements:

Data type	Element	Mandatory or Optional
int64	timeStampUTC_ms	Mandatory
PositionOffset	positionOffset	Optional
RoadSignTypeEnum	roadSignType	Mandatory
RoadSignPermanencyEnum	roadSignPermanency	Optional
string	roadSignValue	Optional
RoadSignDependenciesEnum	roadSignDependencies	Optional
RoadSignValidityEnum	roadSignValidity	Optional
RoadSignValidityValue	roadSignValidityValue	Optional
RoadSignRecognitionType	roadSignRecognitionType	Optional
Int64	detectedObjectID	Optional
Int64	mediaID	Optional
Int32	signRecognitionConfidence_percent	Optional

2.4.4 Lane Boundary Recognition

The combination of information used to report a recognized lane. This complex element is used to describe a single lane boundary. The Position Offset data element provides positional information relative to the vehicle for the reported lane boundary. The Curvature data element is used to report lane boundary curvature in case this can be detected on board.

Name	LaneBoundaryRecognition

Elements:

Data type	Element	Mandatory or Optional
int64	timeStampUTC_ms	Mandatory
PositionOffset	positionOffset	Mandatory
LaneBoundaryTypeEnum	laneBoundaryType	Optional
LaneBoundaryColorEnum	laneBoundaryColor	Optional
double	curvature_1pm	Optional
int32	laneMarkerWidth_mm	Optional
int32	laneMarkerWidthAccuracy_mm	Optional
double	laneDeclination_deg	Optional
double	laneDeclinationAccuracy_deg	Optional
Int32	laneBoundaryTypeConfidence_percent	Optional

2.4.5 Exceptional Vehicle State

The combination of information used to report exceptional vehicle states. These vehicle states are of rare nature and typically indicate a non-regular condition (e.g. tires slipping, crash detected, strong breaking).

Name	ExceptionalVehicleState

Elements:

Data type	Element	Mandatory or Optional
int64	timeStampUTC_ms	Mandatory
TireSlippageEvent	tireSlippageEvent	Optional (deprecated)
CrashDetectionEvent	crashDetectedEvent	Optional
EmergencyBrakingEvent	emergencyBrakingEvent	Optional
ElectronicStabilityControlEvent	electronicStabilityControlEvent	Optional
AntiLockBrakingSystemEvent	antiLockBrakingSystemEvent	Optional (deprecated)

2.4.6Proprietary Information

The combination of information to report OEM proprietary information. This is typically used by OEMs that intend to transports their proprietary data to the Analytic Processing Backend where customer-specific analyses can be done and exclusively provided back to the customer. The proprietary data is submitted as key value pairs in string form. Should binary data need to be transmitted it may be converted using Base64 or other technologies in order to utilize the string key value pairs.

Name	ProprietaryInfo

Elements:

Element	Mandatory or Optional
timeStampUTC_ms	Mandatory
N x keyValuePairs	Mandatory

2.4.7 Environment Status

The combination of information covering the outside environment

Name	Environment Status

Elements:

Data type	Element	Mandatory or Optional
int64	timeStampUTC_ms	Mandatory
LightConditionsEnum	lightConditions	optional
double	externalAirTemperature_DegC	optional
double	externalAirTemperatureAccuracy_DegC	optional
PrecipitationEnum	Precipitation	optional
double	visibleDistance	optional
double	roadSurfaceTemperature_DegC	optional
double	roadSurfaceTemperatureAccuracy_DegC	optional
RoadSurfaceTypeEnum	roadSurfaceType	optional

2.4.8 Object Detection

The information from object detection sensors providing information about moving and static objects.

Name	ObjectDetection

Elements:

Data type	Element	Mandatory or Optional
int64	timeStampUTC_ms	Mandatory
int	detectedObjectID	Optional
PositionOffset	positionOffset	Optional
Vector3D	movingVector_mps	Optional
ObjectTypeEnum	objectType	Optional
Vector3D	objectSize_m	Optional
Vector3D	objectSizeAccuracy_m	Optional
Int64	mediaID	Optional

2.4.9 Assisted Driving: Service and Sensor State

The information from object detection sensors providing information about moving and static objects.

Name	ADServiceAndSensorState

Elements:

Data type	Element	Mandatory or Optional
int64	timeStampUTC_ms	Mandatory
bool	adSpeedControl	Optional
bool	adBrakeControl	Optional
bool	adSteeringControl	Optional
bool	adConnectionAvailable	Optional
bool	sensorObjectRecognition	Optional
bool	sensorSignRecognition	Optional
bool	sensorLaneRecognition	Optional
bool	sensorRoadSurface	Optional
bool	sensorEnvironment	Optional

2.5 Position Estimate

A position estimate contains at least the coordinates of the estimated position and optionally additional attributes.

2.5.1 Timestamp

For general description of the timestamp element see paragraph 2.1.1

2.5.2 Position Type

The position type is an enumeration used to differentiate between different types of absolute positions.

Name	PositionTypeEnum

Values:

Enum Position	Position Type	Description
1	RAW_GPS	Raw GPS position without usage of other sensors and no map matching applied.
2	FILTERED	GPS data filtered or fused with inertial data but not map matched
3	MAP_MATCHED_REGULAR_MAP	Position based on sensors such as GPS and inertial fused and matched to a road network model
4	MAP_MATCHED_HD_MAP	Position based on sensors such as GPS and inertial fused and matched to a HD Map on centerline level
5	MAP_MATCHED_HD_MAP_LANE	Position based on sensors such as GPS and inertial fused and matched to a HD Map on lane level

2.5.3 Interpolated Point

The interpolated point flag is used to flag points in the path that were created artificially by interpolating “real” measurements.

Name	interpolatedPoint
Values	True / False

2.5.4 Longitude

Longitude is part of an absolute position. Longitude is used for different position types, such as raw GPS position, fused position based on absolute and inertial measurements (dead reckoning), and map matched position. Reference system is WGS84. The required resolution is at least 5 decimal. Wherever possible a higher resolution should be used such as 7 decimal.

Name	longitude_deg
Unit	degrees
Range	[-180; 180 [
Resolution	<= 0.00001 deg

2.5.5 Latitude

Latitude follows the definition of Longitude.

Name	latitude_deg
Unit	degrees
Range	[-90; 90 [
Resolution	<= 0.00001 deg

2.5.6 Horizontal Accuracy

The calculated standard deviation for the horizontal position (longitude and latitude combined).

Name	horizontalAccuracy_m
Unit	m
Range	[0; 10000]
Resolution	<= 0.01m

2.5.7 Altitude

Altitude follows the definition of Longitude. It is based on the WGS84 reference ellipsoid and not based on a mean sea level. The altitude is expected to be the altitude at street level, not the altitude of the position of the GPS antenna. Alternatively a different altitude can be provided, e.g. altitude at the roof of the vehicle but then an altitude offset compared to the street level needs to be provided in the vehicle meta data.

Name	altitude_m
Unit	m
Range	[-1000; 9000]
Resolution	<= 0.01m

2.5.8 Heading

The heading of the vehicle. Clockwise measured from North (0 deg), East (90 deg),...

Name	heading_deg
Unit	degrees
Range	[0; 360 [
Resolution	<= 1 deg

2.5.9 Speed

The speed of the vehicle. Speed for raw GPS position has no sign. Other positions are required to be signed where positive means forward and negative means backwards.

Name	speed_mps
Unit	m/s
Range	[-150; 150]
Resolution	<= 0.1 m/s

2.5.10 Altitude Accuracy

The calculated standard deviation for the altitude.

Name	altitudeAccuracy_m
Unit	m
Range	[0; 10000]
Resolution	<= 0.01m

2.5.11 Heading Accuracy

The calculated standard deviation for the heading.

Name	headingAccuracy_deg
Unit	degrees
Range	[0; 360]
Resolution	<= 1 deg

2.5.12 Speed Accuracy

The calculated standard deviation for the vehicle speed.

Name	speedAccuracy_mps
Unit	m/s
Range	[0; 150]
Resolution	<= 0.1 m/s

2.5.13 Speed Detection Type

The technology the speed is detected with.

Name	SpeedDetectionType

Values:

Enum Position	Position Type	Description
1	SPEED_RAW_GPS	Accuracy derived from two consecutive location detection.
2	SPEED_WHEEL_TICKS	Speed derived by the rotation speed of the vehicle
3	SPEED_RADAR_SONAR	Speed derived by detection of returned signals from high frequency sensors

2.5.14 Heading Detection Type

The technology the speed is detected with.

Name	HeadingDetectionType

Values:

Enum Position	Position Type	Description
1	HEADING_RAW_GPS	Accuracy derived from two consecutive location detection.
2	HEADING_MAGNETIC_SENSOR	Heading fusioned by GPS additionally with magnetic sensors.
3	HEADING_MULTI_SENSOR_FUSION	Heading fusioned by multiple inputs including driving distance, steering angle, etc.
4	HEADING_BY_MAP	The heading is derived from the map link on which a vehicle was mapped.

2.5.15 Vehicle Referenced Orientation Vector

Information of the vehicles orientation from the reference horizontal north position as yaw/roll/pitch-rotation value.

Name	vehicleReferencedOrientationVector_rad
Unit	Degrees
Range	3*[-180; +180]
Resolution	<= 0.01

By way of example, a vector of {lateral=-30,longitudinal=45, vertical=90} corresponds to a rotation of 90° to the right (EAST), a roll of 45° along the longitudinal axis (leaning left) and a pitch of -30° along the lateral axis (pointing upwards).

2.5.16 Current Lane Estimation

The current lane number as estimation according to the vehicle internal sensors where lane 0 is most inner lane (left on right hand traffic and right on left hand traffic)

Name	currentLaneEstimate
Unit	N/A
Range	[0; 20]
Resolution	1

2.6 Vehicle Meta Data

The vehicle metadata provides information about the vehicle that is valid for the entire path. This includes vehicle type information, the vehicle reference point. All absolute positions (longitude / latitude) that are reported to the Sensor Data Ingestion Interface are expected to be at the center of the vehicle. All offsets that are reported are expected to be offsets from this center point of the vehicle. Altitude that is reported to the Interface are expected to be altitude on the ground (not the altitude of the location of the GPS antenna). Instead of providing the altitude on the ground, it is possible to report a different altitude with a constant offset. This offset from the ground must be provided through the vehicle metadata.

2.6.1 Vehicle Type (Generic)

The generic vehicle type is an enumeration used to provide information what type of vehicle the sensor data produced.

Name	VehicleTypeGenericEnum

Values

Enum Position	Vehicle Type	Description
1	BUS	A bus
2	DELIVERY_TRUCK	A delivery truck (typically a smaller truck)
3	EMERGENCY_VEHICLE	For example an ambulance, police, fire-truck or similar which is allowed to break normal traffic rules

Enum Position	Vehicle Type	Description
4	MOTORCYCLE	A motorcycle
5	PASSENGER_CAR	A passenger car
6	TAXI	A taxi
7	TRANSPORT_TRUCK	A transport truck (typically a larger truck)

2.6.2 Vehicle specific Meta Data

Some of the data submitted may be very OEM or vehicle specific. For this purpose a generic mechanism of string key value pairs is included that allows submission of such generic information. Based on information provided through such key value pairs in conjunction with individual OEM agreements different processing assumptions may be made.

Data Type	Name	vehicleSpecificMetaData
KeyValuePairString	Values	A repeated field of keys and values in textual form

2.6.3 Vehicle Referenced Point Delta above Ground

This attribute provides the difference between the altitude values and the street level. Altitude values should always be provided at street level and hence this value here should be zero. However in case the altitude is provided at a different level (e.g. at the roof of the vehicle) then this delta value shall provide the distance between the altitude points and the street level.

Name	vehicleReferencePointDeltaAboveGround_m
Unit	m
Range	[0; 20]
Resolution	<= 0.01m

2.6.4 Curvature Accuracy (depr.)

The standard deviation for the curvature. Curvature Accuracy is deprecated in Vehicle Meta Data and will be provided with the Curvature Value within Vehicle Dynamics

Name	curvatureAccuracy_1pm
Unit	m-1
Range	[-1 ; 1]
Resolution	<= 0.00001m-1

2.6.5 Slope Accuracy (depr.)

The standard deviation for the slope. Slope Accuracy is deprecated in Vehicle Meta Data and will be provided with the Slope Value within Vehicle Dynamics.

Name	slopeAccuracy_percent
Unit	%
Range	[-100%; 100%]
Resolution	<= 0.01%

2.6.6 Vehicle Length

The length of the vehicle from most front part to the most rear part (including e.g. bumpers and spoilers)

Name	vehicleLength_m
Unit	m
Range	[0; 100]
Resolution	<= 0.001m

2.6.7 Vehicle Width

The registered width of the vehicle from the most left part to the most right part (including e.g. mirrors)

Name	vehicleWidth_mt
Unit	m
Range	[0; 100]
Resolution	<= 0.001m

2.6.8 Vehicle Height

The total width of the vehicle from lowest part (surface) to the highest part (e.g. antenna)

Name	vehicleHeight_m
Unit	m
Range	[0; 100]
Resolution	<= 0.001m

2.6.9 Primary Fuel Tank Volume

The volume of the fuel tank

Name	primaryFuelTankVolume
Unit	According to Fuel Type volume (l), mass (kg), or energy (Ah)
Range	[0; 1000]
Resolution	<= 0.01l

2.6.10 Primary Fuel Type

The type of the primary fuel source

Name	primaryFuelType
DataType	FuelTypeEnum

Values:

Enum Value	Road Sign Type	Description
1	FUEL_TYPE_OTHER	Refers to any other non-specified Fuel Type. (Unit: percentage)
2	FUEL_TYPE_GASOLINE_L	Refers to Fuel used for petrol engines with ignition systems. (Unit: liter)
3	FUEL_TYPE_DIESEL_L	Refers to Fuel used in engines with spontaneous combustion (Unit: liter)
4	FUEL_TYPE_AUTOGAS_KG	Refers to liquid petrol gas (Unit: kilograms)
5	FUEL_TYPE_BATTERY_AH	Refers to an energy accumulator (Unit: amperehours)
6	FUEL_TYPE_HYDROGEN_KG	Refers to hydrogen used in fuel cell vehicles (Unit: kilograms)

2.6.11 Secondary Fuel Tank Volume

The volume of the secondary fuel tank. Units are according to the Fuel Type

Name	secondaryFuelTankVolume
Unit	According to Fuel Type volume (l), mass (kg), or energy (Ah)
Range	[0; 1000]
Resolution	<= 0.01l

2.6.12 Secondary Fuel Type

The type of the primary fuel source

Name	secondaryFuelType
DataType	FuelTypeEnum

Values:

Enum Value	Road Sign Type	Description
1	FUEL_TYPE_OTHER	Refers to any other non-specified Fuel Type. (Unit: percentage)
2	FUEL_TYPE_GASOLINE_L	Refers to Fuel used for petrol engins with igntion systems. (Unit: liter)
3	FUEL_TYPE_DIESEL_L	Refers to Fuel used in engines with spontaneous combustion (Unit: liter)
4	FUEL_TYPE_AUTOGAS_KG	Refers to Liquid petrol gas (Unit: kilograms)
5	FUEL_TYPE_BATTERY_AH	Refers to an energy accumulator (Unit: amperehours)
6	FUEL_TYPE_HYDROGEN_KG	Refers to hydrogen used in fuel cell vehicles (Unit: kilograms)

2.7 Vehicle Status

2.7.1 Timestamp

For general description of the timestamp element, see paragraph 2.1.1

2.7.2 Transmission Mode

The Transmission Mode is an enumeration about the status of the transmission of the vehicle.

Name	TransmissionModeEnum

Values:

Enum Value	Transmission Mode	Description
1	PARK	Transmission is in park mode. This also applies if the engine is permanently switched off.
2	COASTING	Transmission is in coasting mode and engine is not permanently switched off

Enum Value	Transmission Mode	Description
3	DRIVE	Transmission is in drive mode and engine is not permanently switched off
4	REVERSE	Transmission is in reverse mode and engine is not permanently switched off

2.7.3 Light State

The light state is a bitfield that encodes which lights of the vehicle are turned on and off

Name	LightStateBitfield

Values:

Light State	Bit	Description
LOWBEAMS	0	The low beam light
HIGHBEAMS	1	The high beam light
FOGLAMP_FRONT	2	The front fog light
FOGLAMP_REAR	3	The rear fog light
HAZARD	4	Hazard lights are on (e.g. both turn lights blinking)
LEFT_TURN	5	Left Turn indicated activated
RIGHT_TURN	6	Right Turn indicator activated

2.7.4 Wiper State

The wiper state is an enumeration about the status of the front wiper. This does not necessarily reflect the status of the wiper switch. The value must be derived from actual wiper operation. For example, vehicles that have a rain sensor need to indicate the wiper state as a result of the rain sensor. If the rain sensor is switched on and no rain is detected then the value should be WIPING_OFF.
Note that wiping activity due to windshield washing activities is not in scope of the wiper state.

Name	WiperStateEnum

Values:

Wiper State	Description
WIPING_OFF	No wiping activity (e.g. wiper completely switched off, or rain sensor has not detected any rain)
WIPING_SLOW	Slow wiping activity (e.g. switch is set to slow, or rain sensor has detected light rain)
WIPING_MEDIUM	Medium speed wiping activity (e.g. switch is set to medium, or rain sensor has detected medium rain)

Wiper State	Description
WIPING_FAST	Fast wiping activity (e.g. switch is set to fast, or rain sensor has detected heavy rain)
WIPING_INTERVALL	The Wiping activity is provided by an additional interval value derived by the unit wipes per minute.

2.7.5 Temperature (deprecated)

The outside temperature measured by the vehicle. The value is in degrees Celsius. Temperature within Vehicle State is deprecated due to existing Temperature Value in Environment Status.

Name	temperatureExternal_cel
Unit	°C
Range	[-100°C ; 100°C]
Resolution	<= 1°C

2.7.6 Wiper Speed

The wiper speed is an additional value to measure objective wiper speed as the count of wipes per second.

Name	wiperSpeed_wps
Unit	Wipes per minute
Range	[0,120]
Resolution	= 1

2.7.7 Drive Wheel Reference

According to a fixed setting or to a selected Wheel Drive Setting, this Data object references the currently driven wheels. WheelReferenceBitfield is defined in 2.1.3

2.7.8 Chassis Clearance

According to the vehicles setting to estimated clearance between road surface and the lowest part of the vehicle excluding the tires and wheels.

Name	chassisClearance_m
Unit	m
Range	[0 ; 10]
Resolution	<= 0.001m

2.7.9 Mileage

Information about the current mileage count of the vehicle

Name	mileage_km
Unit	km
Range	[0;32.000.000]
Resolution	<= 0.01 l

2.7.10 Primary Fuel State

Information about the primary fuel content.

Name	primaryFuelState
Unit	Units according to fuel type
Range	[0;10.000]
Resolution	<= 0.01 l

2.7.11 Primary Fuel State Accuracy

Information about the accuracy of the primary fuel state depending on the internal shape of the tank and the accuracy of the sensors.

Name	primaryFuelStateAccuracy
Unit	liters [l]
Range	[0;10.000]
Resolution	<= 0.01 l

2.7.12 Estimated Range on Primary Fuel

The vehicle based information on the estimated Range with the usage of secondary Fuel only.

Name	estimatedPrimaryRange_km
Unit	km
Range	[0;10.000]
Resolution	<= 0.01 l

2.7.13 Secondary Fuel State

Information about the secondary fuel content.

Name	secondaryFuelStatekm
Unit	Units according to fuel type
Range	[0;10.000]
Resolution	<= 0.01 l

2.7.14 Secondary Fuel State Accuracy

Information about the accuracy of the secondary fuel state depending on the internal shape of the tank and the accuracy of the sensors.

Name	secondaryFuelStateAccuracy
Unit	Units according to fuel type
Range	[0;10.000]
Resolution	<= 0.01 l

2.7.15 Estimated Range on SecondaryFuel

The vehicle based information on the estimated Range with the usage of secondary Fuel only.

Name	estimatedSecondaryRange_km
Unit	kilometer
Range	[0;10.000]
Resolution	<= 0.01 l

2.7.16 Maintencance Light State

The maintenance light state is a bitfield that encodes which maintenance lights or warnings are active – not during control cycle upon ignition start-up.

Name	MaintenanceLightStateBitfield

Values:

Light State	Bit	Description
WARNING_ENGINE_CONTROL	0	Warning that the engine has sent a DTC.
WARNING_OIL_PRESSURE	1	Warning of Oil pressure is on
WARNING_COOLANT_TEMP	2	Warning of too high coolant temperature or malfunctioning coolant systems
WARNING_VEHICLE_SERVICE	3	Indicates that the service warning light is on any module indicates that it needs service (e.g. malfunctioning)
WARNING_BATTERY_CHARGING	4	Indicates that the battery charging alert is on.
WARNING_TIRE_PRESSURE	5	Indicates that the tire pressure alarm is on.
WARNING_LAMP_OUT	6	Indicates that one or more light bulbs may not be illuminated
WARNING_OTHER_HIGH_PRIO	7	Other than stated warning is lit. The priority is high and the vehicle should be stopped as soon as possible
WARNING_OTHER_LOW_PRIO	8	Other thatn stated warning is lit. The priority is normal. No immediate stop is required.

2.8 Vehicle Dynamics

Vehicle Dynamics are measurements beyond the position of a vehicle. Typically vehicle dynamics information is measured by the vehicle using on board sensors at high frequency compared to positions (e.g. 5Hz or 10Hz). Depending of the set of sensors in the vehicle different values could be provided. In order to keep the complexity at a manageable level these raw measurements must be converted into meaningful values and hence are a result of calculations either in the vehicle or in the OEM or System Vendor backend.

2.8.1 Timestamp

For general description of the timestamp element see the Timestamp paragraph (p.4)

2.8.2 Curvature

The curvature as measured (calculated) by the vehicle at the location of the vehicle. A positive value means a curve to the right in driving direction.

Name	curvature_1pmracy
Unit	m-1
Range	[-1 ; 1]
Resolution	<= 0.0001m-1

2.8.3 Curvature Accuracy

The standard deviation for the curvature.

Name	curvatureAccuracy_1pm
Unit	m-1
Range	[-1 ; 1]
Resolution	<= 0.0001m-1

2.8.4 Slope

The slope as measured (calculated) by the vehicle at the location of the vehicle. A positive value means uphill slope in driving direction.

Name	slope_percent
Unit	%
Range	[-100%; 100%]
Resolution	<= 0.1%

2.8.5 Slope Accuracy

The standard deviation for the slope.

Name	slopeAccuracy_percent
Unit	%
Range	[-100%; 100%]
Resolution	<= 0.01%

2.8.6 Suspension Travel (1s)

Over a duration of 1 seconds the accumulated travel of the suspensions (averaged over all 4 wheels) is provided in this attribute.

Name	suspensionTravel_mm
Unit	mm
Range	[0; 10000]
Resolution	== 1mm

2.8.7 Average Acceleration Vector (1s)

Over a time range of 1 second, the average acceleration vector is provided in this complex data type.

Name	averageAccelerationVector_mps2
Unit	3 dimensional vector of meter per square second
Range	3*[-100; 100]
Resolution	<= 0.5 mps2

2.8.8 Average Rotation Rate Vector (1s)

Over a timerange of 1 second the average rotation vector is provided in this complex data type as a yaw,pitch,roll rotation rate. A value of 2*PI ~ 6.28 equals a rotation rate of one full rotation per second

Name	averageRotationRateVector_omega
Unit	3 dimensional vector of angular velocity (rad/second)
Range	3*[-314;+314]
Resolution	<= 1

2.9 Position Offset

2.9.1 Lateral Offset

The lateral offset value is used to describe a distance to the side of the vehicle from the vehicle reference point (which is the absolute position of the vehicle). A positive value is to the right of the vehicle in driving direction.

Name	lateralOffset_m
Unit	m
Range	] -∞; ∞ [
Resolution	<= 0.01m

2.9.2 Lateral Offset Simple

The simple lateral offset is an enumeration to indicate positions compared to the vehicle reference point.

Name	LateralOffsetSimpleEnum

Values:

Lateral Offset Simple	Description
LEFT	Towards the left
MIDDLE	Approximately in the middle
RIGHT	Towards the right

2.9.3 Longitudinal Offset

The longitudinal offset value is used to describe a distance in front (positive) or to the back of the vehicle from the vehicle reference point.

Name	longitudinalOffset_m
Unit	m
Range	] -∞; ∞ [
Resolution	<= 0.01m

2.9.4 Longitudinal Offset Simple

The simple longitudinal offset is an enumeration to indicate positions compared to the vehicle reference point.

Name	LateralOffsetSimpleEnum

Values:

Longitudinal Offset Simple	Description
FRONT	In front of the vehicle
CENTER	Approximately at the vehicle
BACK	Behind the vehicle

2.9.5 Vertical Offset

The vertical offset value is used to describe a distance above or below the vehicle from the vehicle reference point.

Name	verticalOffset_m
Unit	m
Range	] -∞; ∞ [
Resolution	<= 0.01m

2.9.6 Vertical Offset Simple

The simple vertical offset is an enumeration to indicate positions compared to the vehicle reference point.

Name	VerticalOffsetSimpleEnum

Values:

Longitudinal Offset Simple	Description
ABOVE	Above the vehicle
AT_LEVEL	Approximately at the level of the vehicle
BELOW	Below the vehicle

2.9.1 Lateral Offset Accuracy

The lateral offset accuracy value is used to describe the confidence of the lateral offset.

Name	transversalOffsetAccuracy_m
Unit	m
Range	] -∞; ∞ [
Resolution	<= 0.01m

2.9.2 Longitudinal Offset Accuracy

The longitudinal offset accuracy value is used to describe the confidence of the longitudinal offset

Name	longitudinalOffsetAccuracy_m
Unit	m
Range	] -∞; ∞ [
Resolution	<= 0.01m

2.9.3 Vertical Offset Accuracy

The vertical offset accuracy value is used to describe the confidence of the vertical offset

Name	longitudinalOffsetAccuracy_m
Unit	m
Range	] -∞; ∞ [
Resolution	<= 0.01m

2.10 Sign Recognition

2.10.1 Timestamp

For general description of the timestamp element see paragraph 2.1.1

2.10.2 Position Offset

This complex object describes the position in reference to the vehicles center point. For detailed description, see section 2.1.2

2.10.3 Road Sign Type

Enumeration of various types of road signs that may be detected by a camera for example.

Name	RoadSignTypeEnum

Values:

Enum Value	Road Sign Type	Description
1	SPEED_LIMIT_START	Speed Limit Sign
2	SPEED_LIMIT_END	Sign that annuls previous speed limit signs
3	NO_OVERTAKING_PASSENGER_CARS_START	Passenger cars are not permitted to overtake
4	NO_OVERTAKING_PASSENGER_CARS_END	End of no overtaking restriction for passenger cars
5	NO_OVERTAKING_TRUCKS_START	Trucks are not permitted to overtake
6	NO_OVERTAKING_TRUCKS_END	End of no overtaking restriction for trucks
7	ALL_RESTRICTIONS_END	End of all restrictions (speed, no overtaking)
8	CITY_START	Start of city sign
9	CITY_END	
10	MOTORWAY_START	
11	MOTORWAY_END	
12	CONSTRUCTION_START	
13	CONSTRUCTION_END	
14	PROTECTED_OVERTAKING_EXTRALANE	
15	PROTECTED_OVERTAKING_EXTRALANE_RIGHTSIDE	
16	PROTECTED_OVERTAKING_EXTRALANE_LEFTSIDE	
17	LANE_MERGE_RIGHT	
18	LANE_MERGE_LEFT	
19	LANE_MERGE_CENTER	
20	RAILWAY_CROSSING_PROTECTED	

21	RAILWAY_CROSSING_UNPROTECTED	
22	ROAD_NARROWS	
23	SHARP_CURVE	
24	SHARP_CURVE_LEFT	
25	SHARP_CURVE_RIGHT	
26	WINDING_ROAD_STARTING_LEFT	
27	WINDING_ROAD_STARTING_RIGHT	
28	STEEP_HILL	
29	STEEP_HILL_UPWARDS	
30	STEEP_HILL_DOWNWARDS	
31	STOP_SIGN	
32	LATERAL_WIND	
33	GENERAL_WARNING	
34	RISK_OF_GROUNDING	
35	ANIMAL_CROSSING	
36	ICY_CONDITIONS	
37	SLIPPERY_ROAD	
38	FALLING_ROCKS	
39	SCHOOL_ZONE	
40	TRAMWAY_CROSSING	
41	CONGESTION_HAZARD	
42	ACCIDENT_HAZARD	
43	PRIORITY_OVER_ONCOMING_TRAFFIC	
44	YIELD_TO_ONCOMING_TRAFFIC	
45	PREFERENCE_ROAD_START	
46	PREFERENCE_ROAD_END	

2.10.4 Road Sign Permanency

Enumeration used to indicate the permanency of a sign (static sign versus a variable (electronic) message sign).

Name	RoadSignPermanencyEnum

Values:

Road Sign Permanency	Description
STATIC	Sign is a fixed sign
VARIABLE	Variable sign, for example an electronic sign, or a gantry with electronic signs

2.10.5 Road Sign Value

The value of a sign. For example a speed limit sign has a speed limit value. The sign value does not have any unit but does reflect the text, that is written on the sign.

Name	roadSignValue
Values	A textual value

By way of example:

Sign meaning	Text value
Speed limit shows ‘30’	30
Speed limit shows ‘45 mph’	45 mph
Maximum height ‘12”’	12”
Maximum width ‘2,2m’	2,2m
Maximum weight ‘12 to’	12 to

2.10.6 Road Sign Dependencies

Some road signs have dependency information. For example, a speed limit may only be in effect if it rains.

Name	RoadSignDependenciesEnum

Values:

Road Sign Dependency	Description
RAIN	Only applicable if it rains
SNOW	Only applicable if there is snow
TIME	A time dependency exists
SEASON	Only valid during a specific season
FOG	Only applicable if fog is present-season
SCHOOL	Applicable for a school area
TRUCKS	Only applicable for trucks
TRAILER	Only applicable for vehicles with a trailer

2.10.7 Road Sign Validity

Enumeration used to indicate the validity of a sign.

Name	RoadSignValidityEnum

Values:

Enum Value	Road Sign Permanency	Description
1	STARTING_IN	Sign is a fixed sign
2	VALID_FOR	Variable sign, for example an electronic sign, or a gantry with electronic signs
3	IN_RIGHT_DIRECTION	
4	IN_LEFT_DIRECTION	
5	ZONE	

2.10.8 Road Sign Validity Value

The optional value of a validity sign. Provided with the sign itself.

Name	roadSignValidityValue
Values	A textual value

2.10.9 RoadSignRecognitionType

The sign event type indicates whether a sign was detected or a sign was not detected. Not detecting a sign is a meaningful information to report signs that are no longer present.

Name	RoadSignRecognitionTypeEnum

Values:

Enum Value	Sign Recognition Type	Description
1	SIGN_DETECTED	A sign was detected
2	SIGN_NOT_DETECTED	No sign was detected

2.10.10 Detected Object

If the Sign Recognition is combined with an object Recognition, then here, the reference to the recognized object is given.

Name	detectedObjectID
Unit	
Range	[0 ; MAX_INTEGER]
Resolution	N/A

2.10.11 Media ID reference

If the Sign Recognition is provided with a media content then here, the reference to the media is given.

Name	mediaID
Unit	
Range	[0 ; MAX_INTEGER]
Resolution	N/A

2.11 LaneBoundaryRecognition

2.11.1 Timestamp

For general description of the timestamp element, see paragraph 2.1.1

2.11.2 Position Offset

This complex object describes the position in reference to the vehicles center point. For detailed description, see section 2.1.2

2.11.3 Lane Boundary Type

Information about what type of lane boundary is reported.

Name	LaneBoundaryTypeEnum

Values:

Enum Value	Lane Boundary Type	Description
1	SINGLE_SOLID_PAINT	Single solid line painted
2	DOUBLE_SOLID_PAINT	Double solid line painted
3	LONG_DASHED_PAINT	
4	SHORT_DASHED_PAINT	
5	SHADED_AREA_PAINT	
6	DASHED_BLOCKS	
7	DOUBLE_LINE_DASHED_SOLID_PAINT	A line with dashed(left) and solid (right) line markings in driving direction
8	DOUBLE_LINE_SOLID_DASHED_PAINT	A line with solid(left) and dashed(right) line markings in driving direction
9	PHYSICAL_DIVIDER	
10	DOUBLE_DASHED_LINE	
	//additional types to be defined later	

2.11.4 Lane Boundary Color

Information about what color the lane marking has.

Name	LaneBoundaryColorEnum

Values:

Enum Value	Lane Boundary Type	Description
1	WHITE	A white lane marking
2	YELLOW	A yellow lane marking
3	BLUE	
	// additional colors to be defined later	

2.11.5 Lane Curvature

The curvature as measured from the lane detection algorithm. Defining the curvature of the lane at the proximity of the vehicle.

Name	curvature_1pm
Unit	m-1
Range	[-1 ; 1]
Resolution	<= 0.0001m-1

2.11.6 Lane Marker Width

the width of the detected lane marker at the proximity of the vehicle.

Name	laneMarkerWidth_mm
Unit	mm
Range	[0; 1000]
Resolution	== 1

2.11.7 Accuracy of Lane Marker Width

The accuracy of the lane Marker width depending on the quality of the sensor readings.

Name	laneMarkerWidthAccuracy_mm
Unit	mm
Range	[0; 1000]
Resolution	== 1

2.11.8 Lane Declination

The measured declination between the lane marker at the proximity of the vehicle and the vehicle itself in mathematical rotation direction (positive == the lane is rotated to the right in reference to the vehicle)

Name	laneDeclination_deg
Unit	Degree
Range	[0; 90]
Resolution	<= 0.1°

2.11.9 Lane Declination Accuracy

The accuracy of the measured lane Declination as it can be biased by noise in the sensors (e.g. dirt on the road, bad weather ...)

Name	laneDeclinationAccuracy_deg
Unit	Degree
Range	[0; 90]
Resolution	<= 0.1°

2.11.10 Lane Boundary Type Confidence

An OEM internal confidence value providing the confidence that the recognized sign type is correct.

Name	laneBoundaryTypeConfidence_percent
Unit	%
Range	[0 ; 100]
Resolution	<= 1%

2.12 Crash Detection

This object within the ExceptionalVehicleState contains necessary attributes if a crash has been detected and stored in the object CrashDetectedEvent.

2.12.1 Drivable Condition

If the vehicles sensor decide that the vehicle is still drivable, the sensor data notifies on this event.

Name	vehicleIsDriveable
Values	True/False

2.12.2 Deployment of Airbags

This attribute contains the information if any of the airbags has been deployed.

Name	airbagsDeployed
Values	True/False

2.12.3 Activated eCall

If the vehicle decides to activate an eCall, the sensor data notifies on this event.

Name	eCallActivated
Values	True/False

2.12.4 Obstacle on the Road

If the vehicle detects that it is located on the road and a potential obstacle to other vehicles, this state is set.

Name	vehicleIsObstacleOnRoad
Values	True/False

2.12.5 Maximal Acceleration during Crash

The maximum acceleration is contains in the 3 values of the 3D Vector.

Name	maxAccelerationVector_mps2
Unit	Meter per square second
Range	[-100; 100]
Resolution	<= 0.1 m/s²

2.13 Emergency Braking Event

This object contains information required to identify and validate the emergency braking event.

2.13.1 Maximal Acceleration Vector

Contains a vector with the maximum acceleration in all three dimension referring to the vehicles lateral, longitudinal, and vertical axis.

Name	maxAccelerationVector_mps2
Unit	Meter per square second
Range	[-100; 100]
Resolution	<= 0.1 m/s²

2.14 Electronic Stability Control

This complex data type contains flags for any occurred event that is counter measured by the ESC.

2.14.1 Anti Slip Action Event

Signaling if the wheels spin at acceleration and counter measures are active (e.g. throttling or de-clutching).

Name	antiSlipActionEvent
Values	True/False

2.14.2 Anti Lock Action Event

Signaling if the wheels block at braking and counter measures are active (e.g. ABS).

Name	antiLockActionEvent
Values	True/False

2.14.3 Electronic Stability Control Event

Signaling if the vehicle turns out of control and ESC-actions are active to bring the vehicle under control.

Name	electronicStabilityControlEvent
Values	True/False

2.14.4 Requested Acceleration Vector

Informs about the requested Acceleration Vector in contrast to the measured Acceleration Vector.

Name	requestedAccelerationVector_mps2
Unit	Meter per square second
Range	[-100; 100]
Resolution	<= 0.1 m/s²

2.14.5 Requested rotation rate vector

Informs about the requested Rotation Vector in contrast to the measured rotation Vector

Name	requestedRotationRateVector_omega
Unit	Angular velocity omega = rad/second
Range	[-100; 100]
Resolution	<= 0.1 m/s²

2.15 Exceptional Vehicle State

2.15.1 TireSlippageEvent (deprecated)

The content when a tire slippage is detected and counter measures are activated. Tire Slippage is deprecated and being handled through the electronic stability control event in section 2.15.4.

2.15.2 Crash Detection

This Event contains relevant information about a detected crash.

Name	CrashDetectionEvent

Elements:

Data Type	Element	Mandatory or Optional
bool	vehicleIsDrivable	Optional
bool	airbagsDeployed	Optional
bool	eCallActivated	Optional
bool	vehicleIsObstackleOnRoad	Optional
Vector3D	maxAccelerationVector_mps2	Optional

2.15.3 Emergency Braking Event

When the vehicle is facing a braking event that goes beyond the comfortable behavior, this event is triggered. The content contains a 3D-Vector of the maximum detected acceleration.

Name	EmergencyBrakingEvent

Elements:

Data Type	Element	Mandatory or Optional
Vector3D	maxAccelerationVector_mps2	Optional

2.15.4 Electronic Dynamic Stability Control Event

The content when the DSC is activated by detecting an unnatural vehicle behavior and activating single breaks to control the motion of the vehicle.

Name	Electronic Stability Control Event

Elements:

Data Type	Element	Mandatory or Optional
bool	antiSlipActionEvent	Optional
bool	antiLockActionEvent	Optional
bool	electronicStabilityControlEvent	Optional
Vector3D	requestedAccelerationVector_mps2	Optional
Vector3D	requestedRotationRateVector_omega	Optional

2.15.5 AntiLockBrakingSystemEvent (deprecated)

The event when an anti-lock braking event is provided within the ElectronicStabilityControlEvent in section 2.15.4.

2.16 Environment Status

2.16.1 Timestamp

For general description of the timestamp element, see paragraph 2.1.1

2.16.2 Light Conditions

Provides the current environmental light conditions according to environmental sensors.

Name	LightConditionEnum

Values

Enum Position	Light Condition Type	Description
1	GENERAL_DAYLIGHT	Describing light environment
2	GENERAL_DARK	Describing dark environment (usually lights are turned on)
3	DAYLIGHT_SUN	Describing daylight with sun shining conditions
4	DAYLIGHT_CLOUD	Describing daylight with diffuse light conditions
5	DUSK_OR_DAWN	Describing a halfway light and dark situation with intermediate visibility
6	NIGHT_LUMINATED	Describing a night situation with illumination (tunnel or urban areas)
7	NIGHT_DARK	Describing a night situation with no illumination of the road and surrounding

2.16.3 External Air Temperature

Contains the degrees of the external air temperature

Name	externalAirTemperature_DegC
Unit	°C
Range	[-100,+100]
Resolution	<=1°C

2.16.4 External Air Temperature Accuracy

Contains the accuracy of the sensor measurement of the external Air Temperature

Name	externalAirTemperatureAccuracy_DegC
Unit	°C
Range	[-100,+100]
Resolution	<=1°C

2.16.5 Precipitation

This Enumeration contains the possible precipitation types that a vehicle can provide.

Name	LightConditionEnum

Values

Enum Position	Precipitation Type	Description
1	NONE	No precipitation detected
2	RAIN	Refers to rain
3	HAIL	Refers to hail
4	SNOW	Refers to snow

2.16.6 Visible Distance

Contains the detected distance of visible light.

Name	visibleDistance_m
Unit	m
Range	[0,10000]
Resolution	==1m

2.16.7 Road Surface Temperature

Contains the degrees of the external air temperature

Name	roadSurfaceTemperature_DegC
Unit	°C
Range	[-100,+100]
Resolution	<=1°C

2.16.8 Road Surface Temperature Accuracy

Contains the accuracy of the sensor measurement of the external Air Temperature

Name	roadSurfaceTemperatureAccuracy_DegC
Unit	°C
Range	[-100,+100]
Resolution	<=1°C

2.16.9 Road Surface Type

This enum provides the detected road type.

Name	RoadSurfaceEnum

Values

Enum Position	Precipitation Type
1	ASPHALT
2	CONCRETE
3	PAVED
4	GRAVEL
5	ICEORSNOW
6	UNKNOWN

2.17 Object Detection Types

2.17.1 Timestamp

For general description of the timestamp element, see paragraph 2.1.1

2.17.2 Detected Object ID

A vehicle unique identification of an object, if the vehicle is capable of following single object detection over time. “0” if no ID given.

Name	detectedObjectID
Unit	
Range	[0, MAX_INTEGER]
Resolution	

2.17.3 Position Offset

Location of the detected object relative to the current vehicle position identified by the complex Object Position Offset

2.17.4 Moving Vectors

A moving 3D vector of the object referring to the vehicles reference axis lateral, longitudinal and vertical

Name	movingVector_mps
Unit	3 dimensional vector of speeds in meter per second
Range	3*[-1000;+1000]
Resolution	<= 1

2.17.5 Object Type Enum

Information about the recognized object type.

Name	ObjectTypeEnum

Values

Enum Position	Recognized Object Type	Description
1	MOVING_GENERAL	ASPHALT
2	STATIC_GENERAL	ASPHALT
3	STATIC_GENERAL_VERTICAL	A static object, with a prevailing vertical extent
4	STATIC_GENERAL_TRANSVERSAL	A static object, with a prevailing transversal extent
5	STATIC_GENERAL_LONGITUDINAL	A static object with a prevailing longitudinal extent
6	MOVING_VEHICLE	A movable (standing or moving) object identified as vehicle
7	MOVING_TRUCK	A movable (standing or moving) object identified as Truck
8	MOVING_BIKE	A movable object identified as bike (bicycle or motorbike)
9	MOVING_PERSON	A movable object identified as person
10	STATIC_BRIDGE	A static object identified as bridge over the road
11	STATIC_TUNNEL	A static object identified as the entrance of a tunnel, or the walls of a tunnel
12	STATIC_POLE	A pole
13	STATIC_BAR	A bar, crossing the road
14	STATIC_TREE	A tree at the side of a road
15	STATIC_WALL	A wall along the road
16	STATIC_BOLLARD	A bollard on the road
17	STATIC_GUIDERAIL	A guiderail along the raod
18	STATIC_TRAFFICISLAND	A trafficisland on the road
19	STATIC_SIGN	A traffic Sign object
20	STATIC_TRAFFIC_LIGHTS	A traffic Light object

2.17.6 Object Size

A moving 3D vector of the object referring to the vehicles reference axis lateral, longitudinal and vertical. The size is a BBOX-value building a box from the Position Estimate point. By way of example, an object with the longitudinal offset of 100m and a length of 10m is located in the range of 100m to 110m. An object with the offset of 100m and a length of -10m is located in the range of 90m to 100m.

Name	objectSizeVector_m
Unit	3 dimensional vector of the object size in m
Range	3*[-1000;+1000]
Resolution	<= 0.01

2.17.7 Object Size Accuracy

When objects are not rectangular and detection can be imprecise, the accuracy of the detection is given here.

Name	objectSizeAccuracyVector_m
Unit	3 dimensional vector of the accuracy of the object size
Range	3*[-1000;+1000]
Resolution	<= 0.01

2.17.8 Reference to a media ID

When media content is provided with the object, a mediaID as reference can be provided with this object.

Name	mediaID
Unit	
Range	[0;MAX_INTEGER]
Resolution	

2.18 Automated Driving Service and Sensor State

2.18.1 Timestamp

For general description of the timestamp element, see paragraph 2.1.1

2.18.2 Speed Component

Contains the information if the vehicle is actively using a component that controls the speed of the vehicle

Name	adSpeedControl
Values	True / False

2.18.3 Brake Component

Contains the information if the vehicle is actively using a component that controls the brakes of the vehicle

Name	adBrakeControl
Values	True / False

2.18.4 Steering Component

Contains the information if the vehicle is actively using a component that controls the steering of the vehicle

Name	adSteeringControl
Values	True / False

2.18.5 Online Connection

Contains the information if the vehicle is able to use the an online connection or if the online connection is available.

Name	adConnectionAvailable
Values	True / False

2.18.6 Surrounding Object Detection

Contains the information if the vehicle is able to detect object around the vehicle

Name	sensorObjectRecognition
Values	True / False

2.18.7 Sign Recognition Detection

Contains the information if the vehicle is able to use the an online connection or if the online connection is available.

Name	sensorSignRecognition
Values	True / False

2.18.8 Lane Recognition Detection

Contains the information if the vehicle is able to detect lanes.

Name	sensorLaneRecognition
Values	True / False

2.18.9 Road Surface Sensor

Contains the information if the vehicle is able to detect the state of the road surface.

Name	sensorRoadSurface
Values	True / False

2.18.10 Environmental Sensor

Contains the information if the vehicle is able to detect environmental conditions as air temperature, precipitation, light.

Name	sensorEnvironment
Values	True / False

2.19 Media Container

2.19.1 Timestamp

For general description of the timestamp element see paragraph 2.1.1

2.19.2 Media Type

Defining the type of media

Name	mediaType

Values

Enum Position	Media Tyoe	Description
1	OTHER	Any other type
2	IMAGE	A content providing an image snapshot
3	VIDEO	A video content over a duration of timesnapshot
4	AUDIO	An audio content over a duration of time

2.19.3 MediaFormat

Contains information about the Media Format in text form (e.g. JPG)

Name	mediaFormat
Values	String

2.19.4 MediaContent

The data content

Name	mediaContent
Values	Binary

2.19.5 MediaID

Identifies the media throughout all media contents and allows to reference a media from other events as e.g. Object recognition.

Name	mediaID
Values	int64

2.19.6 SensorOffset

The position offset of the vehicles sensor to the vehicles reference point (center of the vehicle)

2.19.7 Sensor Direction

A vector defining the direction of the sensor view in relation to the vehicle.

2.19.8 Media Content Duration

If the media content is a video or anything similar then the duration of the media is contained in this attribute

Name	duration_s
Unit	Seconds
Range	[0;+86400]
Resolution	= 1

2.19.9 Sensor Vertical Viewing Angle

Name	verticalViewingAngle_deg
Unit	Degree
Range	[0,360]
Resolution	<= 0.1°

2.19.10 Sensor Horizontal Viewing Angle

Name	horizontalViewingAngle_deg
Unit	Degree
Range	[0,360]
Resolution	<= 0.1°

3. Logical Data Model

Each sensor data submission is a Message. A Message has an Envelope, a Path, and optionally (but likely) Path Events and optionally Path Media.

Message		
	1 x Envelope	Mandatory
	1 x Path	Mandatory
	N x Path Events	Optional
	N x Path Media	Optional

3.1 Envelope

The envelope bears fundamental information about the individual sender (the vehicle) but not to a level that owner of the vehicle can be identified or different messages can be identified that originate from a single vehicle. Elements:

Element	Description	Mandatory or Optional
1 x version	Indicator which version of interface specification is used	Mandatory
1 x submitter	A text. The name of the company that submits the data (e.g. an OEM name or a System Vendor name)	Mandatory
1 x vehicleMetaData	A set of parameters that provide information about the sensors of the vehicle	Mandatory
1 x transientVehicleID	A vehicle ID that shall be reused for submissions of an individual vehicle during one drive (not during multiple different drives).	Optional

3.2 Path

The fundamental data element for rich data sensor submission is a Path. A Path is a list of Position Estimates. The position estimates are ordered starting with the oldest position estimate towards the newest position estimate. A path can be very short, for example for near real time events which are transmitted immediately after they occur. A path can be very long, for example an entire drive over many hours that records the vehicle trace and events for later submission.

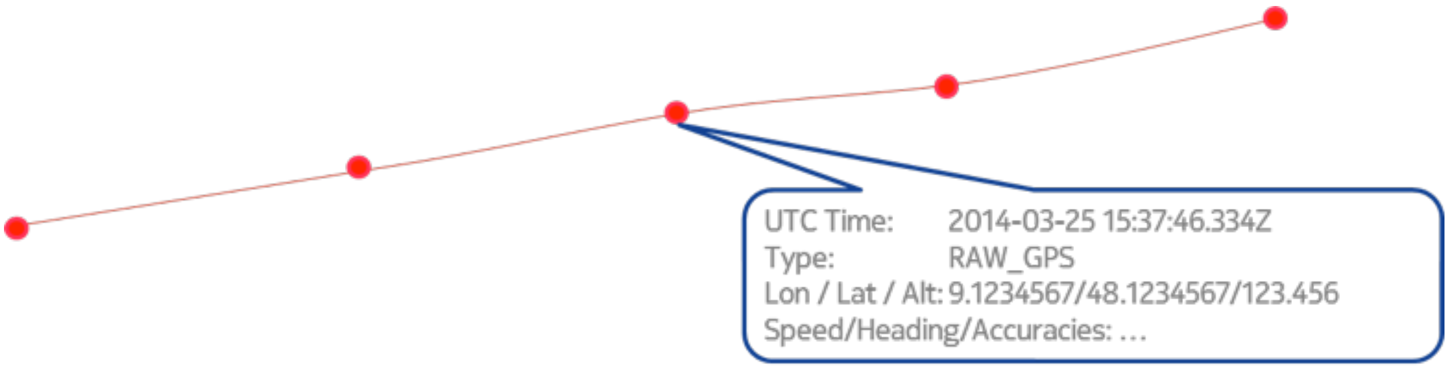
UTC Time: 2014-03-25 15:37:46.334Z
Type: RAW_GPS
Lon / Lat / Alt: 9.1234567/48.1234567/123.456
Speed/Heading/Accuracies: …

Figure 2: A Path with a number of Position Estimates

A Path can have a mixture of Position Estimates of different Position Types. For example raw GPS positions may always be included while map matched positions may only be included when available.

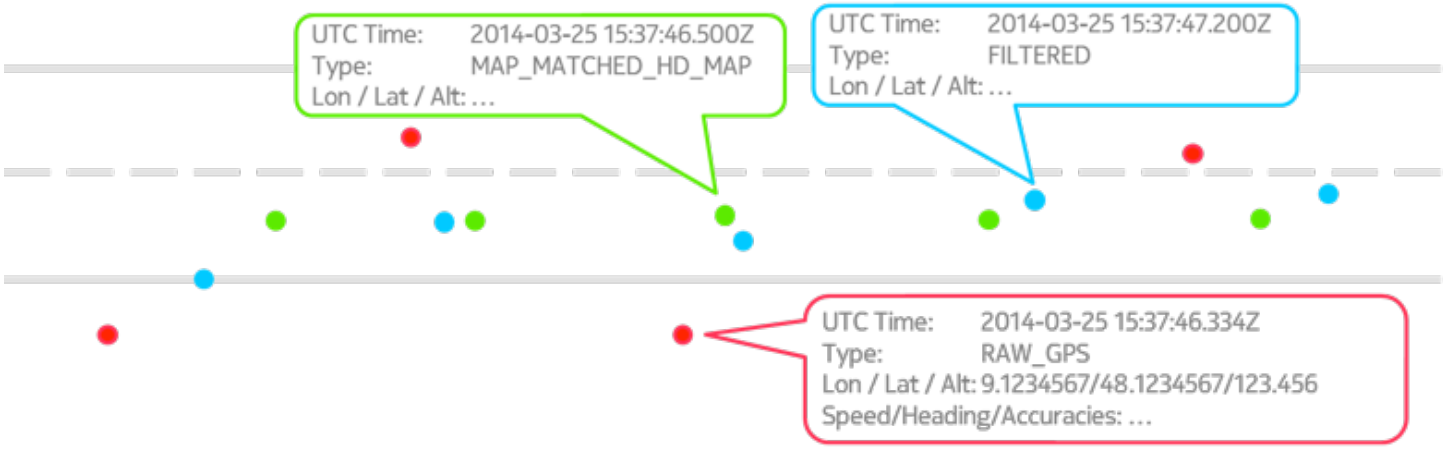
UTC Time: 2014-03-25 15:37:46.500Z
Type: MAP_MATCHED_HD_MAP
Lon / Lat / Alt: …
UTC Time: 2014-03-25 15:37:47.200Z
Type: FILTERED
Lon / Lat / Alt: …
UTC Time: 2014-03-25 15:37:46.334Z
Type: RAW_GPS
Lon / Lat / Alt: 9.1234567/48.1234567/123.456
Speed/Heading/Accuracies: …

Figure 3: A path including different types of Position Estimates

A Path must have at least one Position Estimate.

Elements:

Element	Description	Mandatory or Optional
N x PositionEstimate	The list of position estimates that make up the path	Mandatory

3.3 Path Events

Path Events are providing additional information along a path. This may be singular events such as a change of vehicle operation mode or information about an electronic stability program event. It may also be continuously collected information such as velocity or curvature measurements. Path Events are of different types and may be mixed in the Path Events list. Path Events are referencing the Path based on the timestamp of each Path Event. This timestamp may not exactly match a timestamp in the Path which means interpolation may be required during processing of the information in the Analytic Processing Backend. Note that Path Events should not be collected before the first Position Estimate in the Path and not collected after the last Position Estimate in the Path.

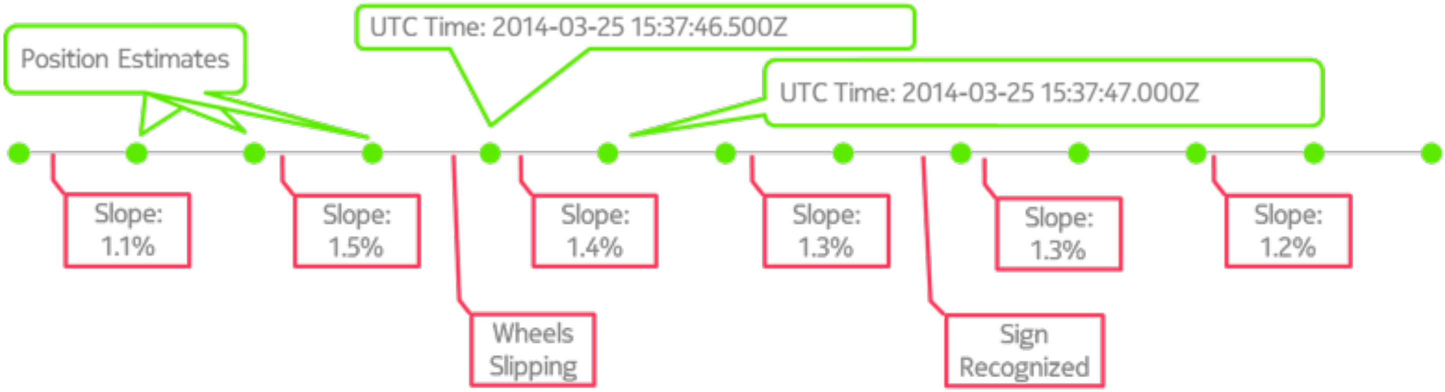

Figure 4: Continuous Path Events (Slope) and Singular Path Events (Wheels Slipping, Sign Recognized)

Path Events are optional in the Sensor Data Message. In such case, only a Path is provided which may have a meaning of its own without actual Path Events. Nevertheless, given that the Analytic Processing Backend is supposed to collection Sensor Data it is expected that a number of Path Events is typically included. Path Events are included in the Sensor Data Message as an ordered list by timestamp starting with the oldest Path Event. T they are not grouped by type. A Path Event is typically a complex data type, which has a number of mandatory and optional components. Vehicles need to be configured to report the components consistently. For example if the vehicle is reporting transmission mode and temperature as part of the Vehicle Status complex type then it must report these two attributes at the very beginning of the path initially. Whenever the transmission mode is changing, another Path Event must be reported that indicates the new transmission mode value in the Vehicle Status complex type. Whenever temperature is changing, another Path Event must be reported that indicates the new temperature value in the Vehicle Status complex type. If by chance two Path Events are reported at the exact same timestamp, then all Information can be merged into one combined Path Events, as long as both events do not have conflicting information.

Element	Description	Mandatory or Optional
N1 x VehicleStatus	Vehicle status has various elements describing the status of the vehicle.	Optional
N2 x VehicleDynamics	Vehicle dynamics has various elements describing the dynamic movements of the vehicle. Vehicle Dynamics Path Events may be present at a much higher frequency than any other event and also at higher frequency than Position Estimates	Optional
N3 x SignRecognition	Information about a recognized (or missing) road sign	Optional
N4 x LaneBoundaryRecognition	Information about a recognized lane boundary	Optional
N5 x ExceptionalVehicleState	Information about an exceptional vehicle state	Optional
N6 x ProprietaryInfo	OEM specific proprietary data typically only used by Platform as a Service customers	Optional
N7 x EnvironmentStatus	Information about the environ-ment	Optional
N8 x ObjectDetection	Identified position of objects (moving or static)	Optional
N9 x ADSensorAndServiceState	Information about the activated Sensors and Controls for AD-feature of the vehicle	Optional
N9 x Fuel Event	Information about fuel state	Optional

3.4 Path Media

Path Media are providing additional media content along a path. It may be collected information referring to a certain Path Events, but also may be content requested at a certain location by vehicle or backend configuration. Similar to Path Events, Path Media does contain a repeated amount of Media Container. Each Container can hold exactly one media as e.g. an Image or a Video-Clip of an image sensor.

Element	Description	Mandatory or Optional
N1 x MediaContainer	mediaContainer	Optional

4 Submission Considerations

Vehicles may have different collection policies for different types of sensor data. According to individual priorities of Sensor Data information, a message could be compiled and sent out the moment a specific Sensor Data reading is detected or the events could be accumulated into one Message and submitted after a given amount of time. By way of example, this could be due to reducing computational performance in critical moments while driving, due to reduction of transmission volume through cell phone connection or due to accumulation and thus referencing all detected Path Events onto one single path. The latter case may be given if the referencing of multiple Path Events to one single vehicle is needed and the usage of Transient Vehicle ID is not supported or not wished.
The following sections provide implementation guidance for different collection policies.

4.1 Delayed / Batch Submission

For delayed submission, it is recommended to record Position Estimates during the entire drive. Should there be a need for gaps in the path for privacy reasons then it is important to separate the individual parts before and after the gap completely. The part after a gap should be treated as a completely new path and therefore collected in a separate submission.
Some of the data elements are added to a path when a change occurs. For example, if the status of the vehicle lights changes then a Vehicle Status Path Event should be added describing the new status. At the beginning of a path, such information also must be provided as initialization independent whether the information just changed. This information would be added at the beginning of the path as Path Event. For delayed submission, some of the data elements described in this document may seem to be not important to be included. However, if not already provided as near real time submission it may be required to include these data also into batch submissions as the data can be required by analyze processes By way of example it may be useful to identify areas where certain events happen frequently.

4.2 Near Real Time Submission

For near real time submissions the paths may be very short. Such submissions are triggered by events in the vehicle that result in the creation of a specific Path Event. For example, a speed limit sign may be recognized which results in the creation of a Sign Recognition Path Event. This may result in the need of an instant submission of this information from the vehicle to the OEM / System Vendor backend to the Analytics Processing Cloud.

In such cases a short Path must be included which is referenced by the Path Event(s). This Path should start at least 10 Position Estimates and at a minimum 100m before the actual event and should include 2 Position Estimates after the event. For such small Paths it is also important that at the beginning of the Path all other Path Events are included that describe the status of the vehicle and that are supplied during at least one Sensor Data Submission, independent whether the information changed or not.

5 Encoding

Messages are encoded using specific serialization mechanisms. The primary mechanism is Protocol Buffers. The Protocol Buffer definitions are available as a separate *.proto file. A binary encoded Protocol Buffer example file is available.

6 Units

All physical values in the Sensor Data Messages as specified in this document are given in metrical base units.

Measurement	Unit	Description
Length	m	Meter
Length	mm	Millimeter
Length	km	Kilometer
Degree	deg	Degree in [0,360] or[180,180]
Speed	mps	Meter per second
Share	%	Percent in [0..100]
Acceleration	mps2	Meter per square second
Rotating speed	Omega	Radiants per second
Volume	l	liter
Energy	Ah	ampere-hours

7 Protobuf Schema Definition

```
message KeyValuePairString
{
required string key = 1;
required string value = 2;
}

message Vector3D
{
        required double longitudinalValue = 1;
        required double lateralValue = 2;
        required double verticalValue = 3;
        optional double longitudinalAccuracy = 4;
        optional double lateralAccuary = 5;
        optional double verticalAccuracy = 6;
}

message VehicleMetaData
{
        enum VehicleTypeGenericEnum
        {
                BUS = 1;
                DELIVERY_TRUCK = 2;
                EMERGENCY_VEHICLE = 3;
                MOTORCYCLE = 4;
                PASSENGER_CAR = 5;
                TAXI = 6;
                TRANSPORT_TRUCK = 7;
        }
        optional VehicleTypeGenericEnum vehicleTypeGeneric = 1;
        repeated KeyValuePairString vehicleSpecificMetaData = 2;
        required double vehicleReferencePointDeltaAboveGround_m = 3;
        optional double curvatureAccuracy_1pm = 4 [deprecated = true];
        optional double slopeAccuracy_percent = 5 [deprecated = true];
        optional double vehicleLength_m = 6;
        optional double vehicleWidth_m = 7;
        optional double vehicleHeight_m = 8;
        enum FuelTypeEnum
        {
                FUEL_TYPE_OTHER = 1;
                FUEL_TYPE_GASOLINE_L = 2;
                FUEL_TYPE_DIESEL_L= 3;
                FUEL_TYPE_AUTOGAS_KG= 4;
                FUEL_TYPE_BATTERY_AH= 5;
                FUEL_TYPE_HYDROGEN_KG= 6;
        }
        optional double primaryFuelTankVolume = 9;
        optional FuelTypeEnum primaryFuelType = 10;
        optional double secondaryFuelTankVolume = 11;
        optional FuelTypeEnum secondaryFuelType = 12;
}

message Envelope
{
        required string version = 1;
        required string submitter = 2;
        required VehicleMetaData vehicleMetaData = 3;
        optional int64 transientVehicleID = 4;
        optional int64 vehicleProfileID = 5;
}

message PositionEstimate
{
        required int64 timeStampUTC_ms = 1;
        enum PositionTypeEnum
        {
                RAW_GPS = 1;
                FILTERED = 2;
                MAP_MATCHED_REGULAR_MAP = 3;
                MAP_MATCHED_HD_MAP = 4;
                MAP_MATCHED_HD_MAP_LANE = 5;
        }
required PositionTypeEnum positionType = 2;
        optional bool interpolatedPoint = 3;
        required double longitude_deg = 4;
        required double latitude_deg = 5;
        required double horizontalAccuracy_m = 6;
        optional double altitude_m = 7;
        optional double heading_deg = 8;
        optional double speed_mps = 9;
        optional double altitudeAccuracy_m = 10;
        optional double headingAccuracy_deg = 11;
        optional double speedAccuracy_mps = 12;
        enum SpeedDetectionEnum
        {
                SPEED_RAW_GPS = 1;
                SPEED_WHEEL_TICKS = 2;
                SPEED_RADAR_SONAR = 3;
        }
        optional SpeedDetectionEnum speedDetectionType = 13;
        enum HeadingDetectionEnum
```

```
{
        HEADING_RAW_GPS = 1;
        HEADING_MAGNETIC_SENSOR = 2;
        HEADING_MULTI_SENSOR_FUSION = 3;
        HEADING_BY_MAP = 4;
    }
    optional HeadingDetectionEnum headingDetectionType = 14;
    optional Vector3D vehicleReferencedOrientationVector_deg = 15;
    optional int32 currentLaneEstimate = 16;
}

message PathSegment
{
    repeated PositionEstimate positionEstimate = 1;
}

message Path
{
    repeated PositionEstimate positionEstimate = 1;
    repeated PathSegment positionSegments = 2;
}

enum WheelReferenceBitfield // Bitwise OR of LightStateBitfield values
{
    frontAxleLeft = 0x1;
    frontAxleRight = 0x2;
    rearAxleLeft = 0x4;
    rearAxleRight = 0x8;
}

message VehicleStatus
{
    required int64 timeStampUTC_ms = 1;
    enum TransmissionModeEnum
    {
        PARK = 1;
        COASTING = 2;
        DRIVE = 3;
        REVERSE = 4;
    }
    optional TransmissionModeEnum transmissionMode = 2;
    enum LightStateBitfield
    {
        LOWBEAMS = 0x0001;
        HIGHBEAMS = 0x0002;
        FOGLAMP_FRONT = 0x0004;
        FOGLAMP_READ = 0x0008;
        HAZARD = 0x0010;
        LEFT_TURN = 0x0020;
```

```
                RIGHT_TURN = 0x0040;
        }
        optional int64 lightStateBitfield = 3; // Bitwise OR of LightStateBitfield values
        enum WiperStateEnum
        {
                WIPING_OFF = 1;
                WIPING_SLOW = 2;
                WIPING_MEDIUM = 3;
                WIPING_FAST = 4;
                WIPING_INTERVALL = 5;
        }
        optional WiperStateEnum wiperState = 4;
        optional double temperatureExternal_cel = 5 [ deprecated = true ];
        optional int32 wiperSpeed_wps = 6;
        optional int64 driveWheelReference = 7;
        optional double chassisClearance_m = 8;
        optional double mileage_km = 9;
        optional double primaryFuelState = 10;
        optional double primaryFuelStateAccuracy = 11;
        optional double estimatedPrimaryRange_km = 12;
        optional double secondaryFuelState = 13;
        optional double secondaryFuelStateAccuracy = 14;
        optional double estimatedSecondaryRange_km = 15;
        enum MaintenanceLightStateBitfield
        {
                WARNING_ENGINE_CONTROL  = 0x0001;
                WARNING_OIL_PRESSURE    = 0x0002;
                WARNING_COOLANT_TEMP    = 0x0004;
                WARNING_VEHICLE_SERVICE = 0x0008;
                WARNING_BATTERY_CHARGING= 0x0010;
                WARNING_TIRE_PRESSURE   = 0x0020;
                WARNING_LAMP_OUT        = 0x0040;
                WARNING_OTHER_HIGH_PRIO = 0x0080;
                WARNING_OTHER_LOW_PRIO  = 0x0100;
        }
        optional MaintenanceLightStateBitfield maintenanceLightState = 16;
}

message VehicleDynamics
{
        required int64 timeStampUTC_ms = 1;
        optional double curvature_1pm = 2;
        optional double slope_percent = 3;
        optional double curvatureAccuracy_1pm = 4;
        optional double slopeAccuracy_percent = 5;
        optional int32 averageSuspensionTravel_mm = 6;
        optional Vector3D averageAccelerationVector_mps2 = 7;
        optional Vector3D averageRotationRateVector_omega = 8;
}
```

```
message PositionOffset
{
        optional double lateralOffset_m = 1;
        enum LateralOffsetSimpleEnum
        {
                LEFT = 1;
                MIDDLE = 2;
                RIGHT = 3;
        }
        optional LateralOffsetSimpleEnum lateralOffsetSimple = 2;
        optional double longitudinalOffset_m = 3;
        enum LongitudinalOffsetSimpleEnum
        {
                FRONT = 1;
                CENTER = 2;
                BACK = 3;
        }
        optional LongitudinalOffsetSimpleEnum longitudinalOffsetSimple = 4;
        optional double verticalOffset_m = 5;
        enum VerticalOffsetSimpleEnum
        {
                ABOVE= 1;
                AT_LEVEL = 2;
                BELOW= 3;
        }
        optional VerticalOffsetSimpleEnum verticalOffsetSimple = 6;
        optional double lateralOffsetAccuracy_m = 7;
        optional double longitudinalOffsetAccuracy_m = 8;
        optional double verticalOffsetAccuracy_m = 9;
}

message SignRecognition
{
        required int64 timeStampUTC_ms = 1;
        optional PositionOffset positionOffset = 2;
        enum RoadSignTypeEnum
        {
                SPEED_LIMIT_START = 1;
                SPEED_LIMIT_END = 2;
                NO_OVERTAKING_PASSENGER_CARS_START = 3;
                NO_OVERTAKING_PASSENGER_CARS_END = 4;
                NO_OVERTAKING_TRUCKS_START = 5;
                NO_OVERTAKING_TRUCKS_END = 6;
                ALL_RESTRICTIONS_END = 7;
                CITY_START = 8;
                CITY_END = 9;
                MOTORWAY_START = 10;
                MOTORWAY_END = 11;
                CONSTRUCTION_START = 12;
                CONSTRUCTION_END = 13;
                PROTECTED_OVERTAKING_EXTRALANE = 14;
                PROTECTED_OVERTAKING_EXTRALANE_RIGHTSIDE = 15;
                PROTECTED_OVERTAKING_EXTRALANE_LEFTSIDE = 16;
                LANE_MERGE_RIGHT = 17;
                LANE_MERGE_LEFT = 18;
                LANE_MERGE_CENTER = 19;
                RAILWAY_CROSSING_PROTECTED = 20;
                RAILWAY_CROSSING_UNPROTECTED = 21;
                ROAD_NARROWS = 22;
                SHARP_CURVE = 23;
                SHARP_CURVE_LEFT = 24;
                SHARP_CURVE_RIGHT = 25;
                WINDING_ROAD_STARTING_LEFT = 26;
                WINDING_ROAD_STARTING_RIGHT = 27;
                STEEP_HILL = 28;
                STEEP_HILL_UPWARDS = 29;
                STEEP_HILL_DOWNWARDS = 30;
                STOP_SIGN = 31;
                LATERAL_WIND = 32;
                GENERAL_WARNING = 33;
                RISK_OF_GROUNDING = 34;
                ANIMAL_CROSSING = 35;
                ICY_CONDITIONS = 36;
                SLIPPERY_ROAD = 37;
                FALLING_ROCKS = 38;
                SCHOOL_ZONE = 39;
                TRAMWAY_CROSSING = 40;
                CONGESTION_HAZARD = 41;
                ACCIDENT_HAZARD = 42;
                PRIORITY_OVER_ONCOMING_TRAFFIC = 43;
                YIELD_TO_ONCOMING_TRAFFIC = 44;
                PREFERENCE_ROAD_START = 45;
                PREFERENCE_ROAD_END = 46;
        }
        required RoadSignTypeEnum roadSignType = 3;
        enum RoadSignPermanencyEnum
        {
                STATIC = 1;
                VARIABLE = 2;
        }
        optional RoadSignPermanencyEnum roadSignPermanency = 4;
        optional string roadSignValue = 5;
        enum RoadSignDependenciesEnum
        {
                RAIN = 1;
                SNOW = 2;
                TIME = 3;
                SEASON = 4;
                FOG = 5;
                SCHOOL = 6;
                TRUCKS = 7;
                TRAILER = 8;
        }
        optional RoadSignDependenciesEnum roadSignDependencies = 6;
        enum RoadSignValidityEnum
        {
                STARTING_IN = 1;
                VALID_FOR = 2;
                IN_RIGHT_DIRECCTION = 3;
                IN_LEFT_DIRECTION = 4;
                ZONE = 5;
        }
        optional RoadSignValidityEnum validity = 7;
        optional double validityValue = 8;
        enum RoadSignRecognitionTypeEnum
        {
                SIGN_DETECTED = 1;
                SIGN_NOT_DETECTED = 2;
        }
        optional RoadSignRecognitionTypeEnum roadSignRecognitionType = 9;
        optional int64 detectedObjectID= 10;
        optional int64 mediaID = 11;
    }

    message LaneBoundaryRecognition
    {
        required int64 timeStampUTC_ms = 1;
        required PositionOffset positionOffset = 2;
        enum LaneBoundaryTypeEnum
        {
                SINGLE_SOLID_PAINT = 1;
                DOUBLE_SOLID_PAINT = 2;
                LONG_DASHED_PAINT = 3;
                SHORT_DASHED_PAINT = 4;
                SHADED_AREA_PAINT = 5;
                DASHED_BLOCKS = 6;
                DOUBLE_LINE_DASHED_SOLID_PAINT = 7;
                DOUBLE_LINE_SOLID_DASHED_PAINT = 8;
                PHYSICAL_DIVIDER = 9;
                DOUBLE_DASHED_LINES = 10;
                // Additional types to be defined later
        }
        optional LaneBoundaryTypeEnum laneBoundaryType = 3;
        enum LaneBoundaryColorEnum
        {
                WHITE = 1;
                YELLOW = 2;
                BLUE = 3; //in some countries used as reflectors on bridge (icying
    conditions)
                // Additional colors to be defined later
        }
        optional LaneBoundaryColorEnum laneBoundaryColor = 4;
        optional double curvature_1pm = 5;
        optional int32 laneMarkerWidth_mm = 6;
        optional int32 laneMarkerWidthAccuracy_mm = 7;
        optional double laneDeclination_deg = 8;
        optional double laneDeclinationAccuracy_deg = 9;
        optional int32 laneBoundaryTypeConfidence_percent = 10;
    }

    message TireSlippageEvent
    {
        //[ deprecated = true ]
    }

    message CrashDetectedEvent
    {
        optional bool vehicleIsDrivable = 1;
        optional bool airbagsDeployed = 2;
        optional bool eCallActivated = 3;
        optional bool vehicleIsObstackleOnRoad = 4;
        optional Vector3D maxAccelerationVector_mps2 = 5;
    }

    message EmergencyBrakingEvent
    {
        optional Vector3D maxAccelerationVector_mps2 = 1;
    }

    message DynamicStabilityControlEvent
    {
        // To be defined later
    }

    message AntiLockBrakingSystemEvent
    {
        //  [deprecated = true]
    }

    message ElectronicStabilityControlEvent
    {
        optional bool antiSlipActionEvent = 1;
        optional bool antiLockActionEvent = 2;
        optional bool electronicStabilityControlEvent = 3;
        optional Vector3D requestedAccelerationVector_mps2 = 4;
```

```
        optional Vector3D requestedRotationRateVector_omega = 5;
}
message ExceptionalVehicleState
{
        required int64 timeStampUTC_ms = 1;
        optional TireSlippageEvent tireSlippage= 2 [deprecated = true];
        optional CrashDetectedEvent crashDetected= 3;
        optional EmergencyBrakingEvent emergencyBraking= 4;
        optional DynamicStabilityControlEvent dynamicStabilityControl= 5 [deprecated
= true];
        optional AntiLockBrakingSystemEvent antiLockBrakingSystem= 6 [deprecated =
true];
        optional ElectronicStabilityControlEvent electronicStabilityControl= 7;
}

message ProprietaryInfo
{
        required int64 timeStampUTC_ms = 1;
        repeated KeyValuePairString keyValuePairs = 2;
}

message EnvironmentStatus
{
        required int64 timeStampUTC_ms = 1;
        enum LightConditionsEnum
        {
                GENERAL_DAYLIGHT = 1;
                GENERAL_DARK = 2;
                DAYLIGHT_SUN = 3;
                DAYLIGHT_CLOUD = 4;
                DUSK_OR_DAWN = 5;
                NIGHT_LUMINATED = 6;
                NIGHT_DARK = 7;
        }
        optional LightConditionsEnum lightConditions = 2;
        optional double externalAirTemperature_DegC = 3;
        optional double externalAirTemperatureAccuracy_DegC = 4;
        enum PrecipitationEnum
        {
                NONE = 1;
                RAIN = 2;
                HAIL = 3;
                SNOW = 4;
        }
        optional PrecipitationEnum precipitation = 5;
        optional double visibleDistance_m = 6;
        optional double roadSurfaceTemperature_DegC = 7;
        optional double roadSurfaceTemperatureAccuracy_DegC = 8;
        enum RoadSurfaceTypeEnum
```

```
{
        ASPHALT = 1;
        CONCRETE = 2;
        PAVED = 3;
        GRAVEL = 4;
        ICEORSNOW = 5;
        UNKNOWN = 6;
    }
    optional RoadSurfaceTypeEnum roadSurfaceType = 9;
}
message ObjectDetection
{
    enum ObjectTypeEnum
    {
        MOVING_GENERAL= 1;
        STATIC_GENERAL = 2;
        STATIC_GENERAL_VERTICAL = 3;
        STATIC_GENERAL_TRANSVERSAL = 4;
        STATIC_GENERAL_LATERAL = 5;
        MOVING_VEHICLE = 6;
        MOVING_TRUCK = 7;
        MOVING_BIKE = 8;
        MOVING_PERSON = 9;
        STATIC_BRIDGE = 10;
        STATIC_TUNNEL = 11;
        STATIC_POLE = 12;
        STATIC_BAR = 13;
        STATIC_TREE = 14;
        STATIC_WALL = 15;
        STATIC_BOLLARD = 16;
        STATIC_GUIDERAIL = 17;
        STATIC_TRAFFICISLAND = 18;
        STATIC_SIGN = 19;
        STATIC_TRAFFIC_LIGHT = 20;
    }
    required int64 timeStampUTC_ms = 1;
    optional int64 detectedObjectID = 2;
    optional PositionOffset positionOffset = 3;
    optional Vector3D movingVector_mps = 4;
    optional ObjectTypeEnum objectType = 5;
    optional Vector3D objectSize_m = 6;
    optional Vector3D objectSizeAccuracy_m = 7;
    optional int64 mediaID = 8;
}

message ADServiceAndSensorState
{
    required int64 timeStampUTC_ms = 1;
    optional bool adSpeedControl = 2;
```

```
        optional bool adBrakeControl = 3;
        optional bool adSteeringControl = 4;
        optional bool adConnectionAvailable = 5;
        optional bool sensorObjectRecognition = 6;
        optional bool sensorSignRecognition = 7;
        optional bool sensorLaneRecognition = 8;
        optional bool sensorRoadSurface = 9;
        optional bool sensorEnvironment = 10;
}

message PathEvents
{
        repeated VehicleStatus vehicleStatus = 1;
        repeated VehicleDynamics vehicleDynamics = 2;
        repeated SignRecognition signRecognition = 3;
        repeated LaneBoundaryRecognition laneBoundaryRecognition = 4;
        repeated ExceptionalVehicleState excpetionalVehicleState = 5;
        repeated ProprietaryInfo proprietaryInfo = 6;
        repeated EnvironmentStatus environmentStatus = 7;
        repeated ObjectDetection objectDetection = 8;
        repeated ADServiceAndSensorState adServiceAndSensorState = 9;
}

message MediaContainer
{
        enum MediaTypeEnum
        {
                OTHER = 1;
                IMAGE = 2;
                VIDEO = 3;
                AUDIO = 4;
        }
        required int64 timeStampUTC_ms = 1;
        required MediaTypeEnum mediaType = 2;
        required string mediaFormat = 3;
        required bytes mediaContent = 4;
        optional int64 mediaID = 5;
        optional PositionOffset sensorOffset = 6;
        optional Vector3D sensorDirection = 7;
        optional int32 duration_s = 8;
        optional double verticalViewingAngle = 9;
        optional double horizontalViewingAngle = 10;
}

message PathMedia
{
        repeated MediaContainer mediaContainer = 1;
}
```

```
message Message
{
        required Envelope envelope = 1;
        required Path path = 2;
        optional PathEvents pathEvents = 3;
        optional PathMedia pathMedia = 4;
}
```