CROSS-CPP

Ecosystem for Services based on integrated Cross-sectorial Data Streams from multiple Cyber Physical Products and Open Data Sources

CIDM

Common Industrial Data Model

CIDM Common Industrial Data Model

Open Specification

November 2020

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 780167

Foreword

The key motivation of the Cross-CPP project is to develop a unified set of services with which to man-age cross-sectorial data streams. This would enable its exploitation through an IT environment in which the data is collected and stored, serving as both a provider of features (e.g. detection of anomalies in on-board sensors) and an enhancer of those which already exist (e.g. weather forecasting, energy optimisation…).

Thus, this platform bridges the gap between industries (from a wide range of ambits, such as automotive or IoT) and the access to valuable, high volume data that is being continuously generated by products of their interest (vehicles, smart home devices, etc.). This uninterrupted stream of data from massively produced items will enable both the combination of data streams from various sources and an interdisciplinary Big Data-driven business potential for both manufacturers and IT organisations.

However, the viability of these business opportunities is currently hindered by how fragmented the industrial environment is with respect to proprietary software approach that prevails. This prevents from developing clear models and tools for the mentioned cross-sectorial collaboration, as it limits access to data from devices of interest and impedes the development of agreed terms of use, privacy, etc.

In contrast, the Cross-CPP project focuses on developing a readily available source of data and associated services to the outside world. Therefore, the establishment of an operative cross-sectorial Big Data Ecosystem is founded upon three fundamental pillars. Namely, its openness for the integration of diverse data providers (including the provision of a standardized cross-sectorial data model), its operation via a Big Data marketplace and a controlled access to the available data streams (including data ownership concerns).

Cross-CPP consortium partners

Executive Summary

In accordance with the purpose of collecting data from vehicles and buildings (property of various manufacturers), Cross CPP requires a common understanding of data and information. Within the scope of the project, the Common Industrial Data Model (CIDM) was developed as open and highly scalable automotive big data format. The CIDM that is the extension of the CVIM of the Automat project. The CIDM provides a brand-independent and transparent data model, which harmonizes proprietary data into generic datasets. However, the CIDM is not rigid, rather representing a living data structure, where in reference to the needs of the service provider community the amount of signals to be recorded as well as the type of measurement channels can be modified or extended.

This document comprises the specification of the CIDM and describes a key result of the Cross CPP project. A machine-readable CIDM definition is publicly available as part of the SDK.

This specification document is structured as follows:

1 Terminology and Format of the CIDM Specification

1.1 Terminology of Requirements

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119.

1.2 Specification Format

The specification is provided in text form throughout the following sections and in a machine-readable format as part of the SDK, which is publicly available at github. This enables automated testing and validation of CIDM data and Cloud Storage Provider Interfaces. The CIDM specification and definition exploits the following formats and standards:

- CIDM Relationship Modelling: Unified Modeling LanguageTM (UML), Version 2.5
- CIDM Model Definition: JSON Schema, Version 4
- CIDM Reference and Prototype Implementation: JavaScript Object Notation (JSON)
- Cloud Storage Provider Interface Specification: OpenAPI, Version 3.0
- Cloud Storage Provider Prototype Implementation: Python, Version 3.6

1.3 JSON Schema Datatype Extension

JSON Schema allows detailed and rich data model descriptions. Nevertheless, it supports the following seven primitive data types for JSON values:

- Array – a JSON array
- Boolean – Value may be true or false
- Integer – Number without fraction or exponent part
- Number – Any number, Number includes integer
- Null – Null (empty) value
- Object – JSON object
- String – Text value

To improve and validate contents of fields the JSON Schema Validation defines semantic validation using the format attribute. Table 1 further extends this format attribute to provide a more detailed description of data types within the CIDM definition and Cloud Storage Provider specification

Table 1. Extension of JSON Schema Format Attributes.

Value Type (JSON Schema) “type” attribute	Format “format” attribute	Description	Example
integer	int8	signed 8-bit integer	- 128
integer	int16	signed 16-bit integer	32767
integer	int32	signed 32-bit integer	- 2147483648

integer	int64	signed 64-bit integer	1
integer	uint8	unsigned 8-bit integer	255
integer	uint16	unsigned 16-bit integer	65535
integer	uint32	unsigned 32-bit integer	4294967295
integer	uint64	unsigned 64-bit integer	0
number	double	single precision floating point value representing a 32-bit binary format IEEE 754 value	-3.1415926, +Infinity
number	double	double precision floating point value representing a 64-bit binary format IEEE 754 value	3.1415926, - Infinity
string	date	As defined by full-date in RFC3339	2016-04-07
string	date-time	As defined by date-time in RFC3339	2016-04- 07T11:45:13.01Z
string	uuid	Universally unique identifier (UUID representing a 128-bit unique value in a hexadecimal representation	664f4826-9563-4a56- 9d73-c8cf55572fd5
string	email	E-Mail address representation	mail@cross-cpp.eu
string	version	CIDM Version Format with Schema a.b.c: - a: Major Version - b: Minor Release - c: Revision	1.2.1

2 The Cross-CPP Cross-Industrial Data Model

This section defines the Cross CPP data model, that is the extension of the CVIM of the Automat project. The main changes of the data model involve:

- Extension of CVIM towards CIDM, to support various CPP data sources
- Inclusion of new signal list related to CPP type 'Smart Building'
- Extension by new types of measurement channels (event based & Basic CPP Information)
- Creation of CPP specific measurement channel lists (extended vehicle lists & new smart building lists)
- Modification of the timestamp requirements to support single value data packages

The following sections 3.2, 3.3 and 3.4 describe the Signal layer, the Measurement layer and the Data layer respectively.

2.1 Model Architecture

The FP CIDM architecture has the same three layers architecture as the CVIM. The Signal layer contain Signals that are the information providers within the CPP devices like vehicles or smart buildings. Signals observe the environment and generate data, they detect physical and chemical phenomenon, for example, speed, temperature, charge state level, etc.

The Measurement layer provides signals' data aggregation. The data needs to be pre-processed since raw sensor data exceeds the available storage and transferring capacity, to reduce the size of data down-sampling and histograms methods are provided. The CVIM specification provide detailed information about the applied methods.

The Data Layer aggregates data inside data packages to store and transfer. "*One data package contains data from exactly one signal measured with one Measurement Channel. In addition to the actual data, Data Packages contain header information ("meta data"). This header information provides ownership of the data and gives quality of signal indications by OEM signatures or describes parameters of the measurement (e.g. time, rough position estimate, etc.)*".

Figure 1 shows the high-level architecture of the CIDM. The bottom layer contains Signals. The next layer is the Measurement Layer. The top layer is the data layer.

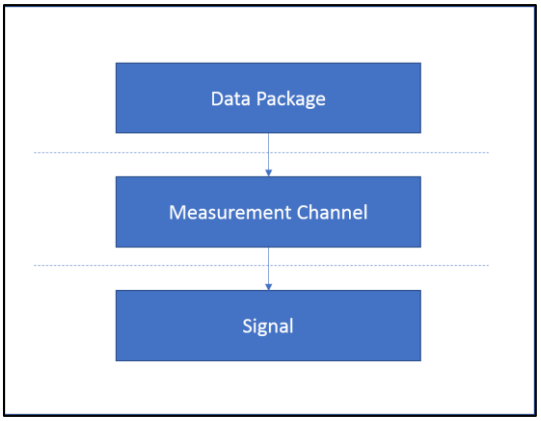

Figure 1: Layered High-level View of the Common Industrial Data Model (CIDM)

2.2 Signal Layer Specification

Sensors are the perception organs of CPP devices like vehicles and buildings. It is their main duty to detect physical phenomenon and chemical quantities by transferring them into electrical signals. The signal layers describe different types of signals and formats represented in the system. A new property is needed to group signals regarding the signal source type, cpp-type. Figure 2 shows the UML modelling of the signals for CIDM.

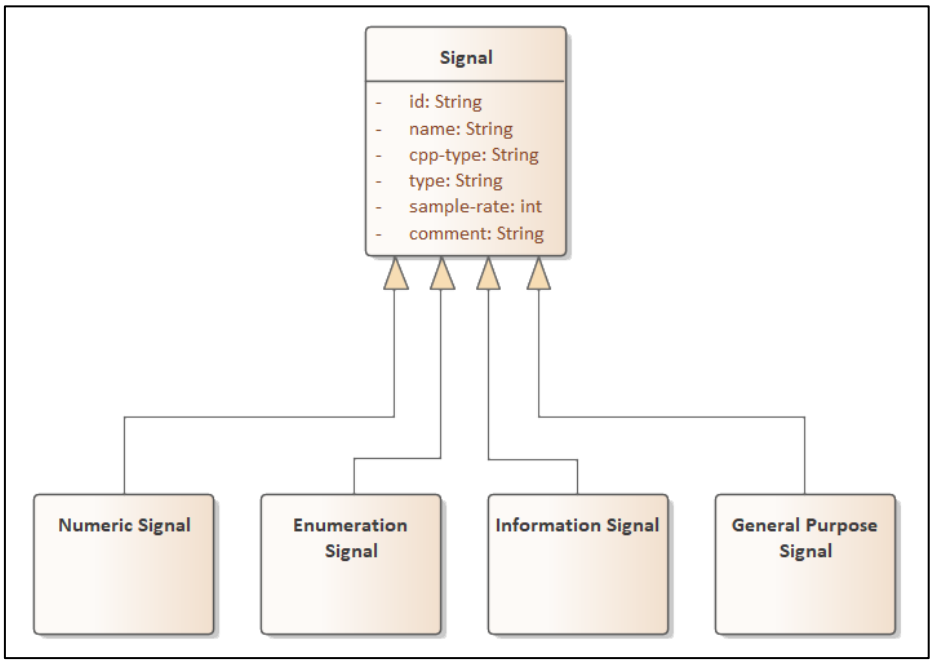

Figure 2. Signal UML Model.

The cpp-type is a required property that must be one of the two values, “vehicle” or “building”. Table 2 shows the complete definition of the Signal.

Table 2. Signal property definition.

Property	Occurrence	Type	Format	Description
Common Properties				
`id`	Required	String		Unique Identifier of the Signal
`name`	Required	String		Name of the Signal
`cpp-type`	Required	String	one of: - vehicle - building	Type of the CPP
`type`	Required	String	one of: - numeric - enumeration - information - general-purpose	Type of the Signal

format	Optional	String		Signals representation format
sample-rate	Required	Numeric	double	Sample rate in Hz (Samples per Second). Must be larger than or equal to zero.
comment	Optional	String		Description of the signal
Numeric Signal				
type	Required	String	"numeric"	Type of the Signal needs to be numeric
format	Required	String	*<numeric formats>*	Signal's numeric representation (e.g. uint8, double, etc.)
min	Required	Number	<according to format>	Minimum Signal value
max	Required	Number	<according to format>	Maximal Signal value
resolution	Required	Number	<according to format>	Signals resolution
unit	Required	String		Unit of the Signal (e.g. ºC)
Enumeration Signal				
type	Required	String	"enumeration"	Signal's type attribute needs to be "enumeration"
items	Required	Array	String	String array with possible Signal values
Information Signal				
type	Required	String	"information"	Signal's type attribute needs to be "information"
format	Required	String		Signals representation format (e.g. VIN, etc.)
General Purpose Signal				
type	Required	String	"general-purpose"	Signal's type attribute needs to be "general-purpose"
*	Optional	Any	No	May be extended with further attributes

2.3 Measurement Layer Specification

"*The measurement layer defines how sensor signals are captured and processed. One Measurement Channel describes how samples from one (or more - in the case of multidimensional histograms) sensor signal are aggregated and measured*" (TUDO, Consortium Partners, 2017). Figure 3 shows the Measurement Channel UML model.

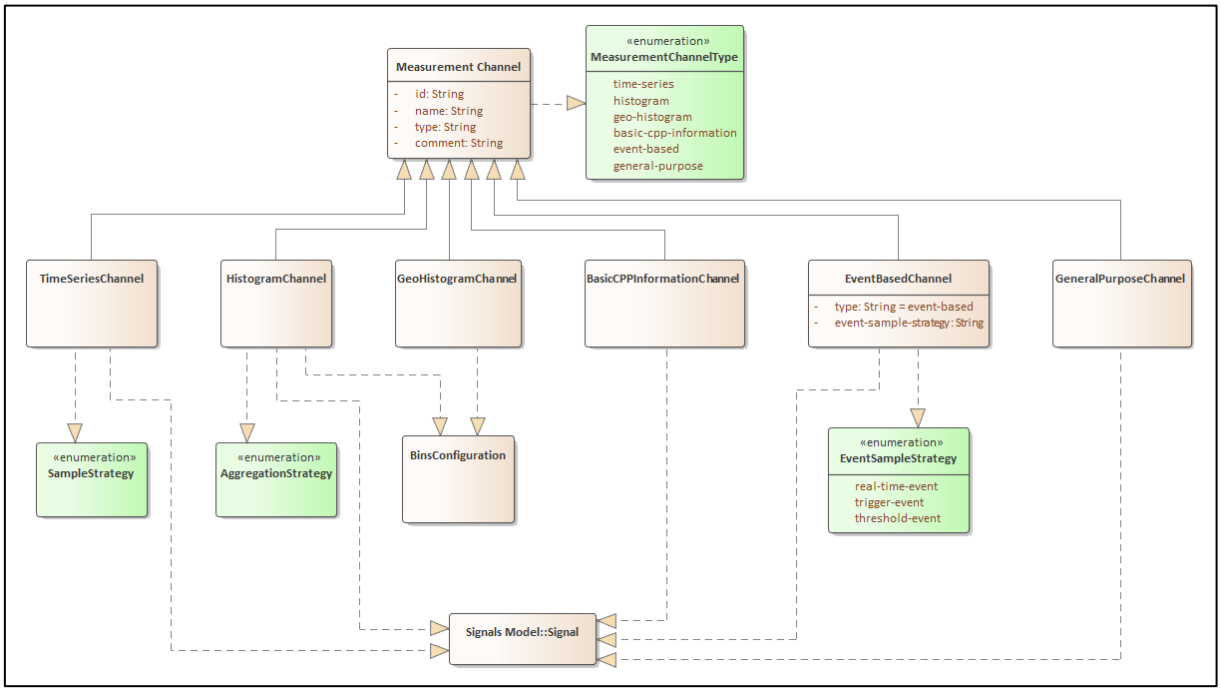

Figure 3. Measurement Channel UML Model.

The basic CPP information channel provides static information that is not measured by sensors but provides information about the CPP device, like the colour of a vehicle, the number of floors of a building, the identification number of a car, etc.

The event-based measurement channel provides information of events that occurs when the value of a measurement gets to a specific value (real-time-event) or when the value passes a specific threshold (threshold) and the last one that provides information about the event and the data-packages that have triggered the event. Table 3 details the measurement channel specification.

Table 3: Measurement channel definition

Property	Occurrence	Type	Format	Description
Common Properties				
`id`	Required	String	No	Unique Identifier of the Measurement Channel
`name`	Required	String	No	Name of the Measurement Channel
`type`	Required	String	one of: - time-series - histogram	Type of Measurement Channel

			- geo-histogram - general-purpose - event-based - basic-cpp-information	
comment	Optional	String	No	Description of the signal
Time Series Measurement Channel				
type	Required	String	“time-series”	Type of the Measurement Channel needs to be time-series
format	Required	String		Data type format of the samples
dimension	Optional	Number	uint32	Dimension of the Time-series. If dimension is not given, one-dimensional is assumed
capture-interval	Required when on-change is false	Number	double	Capture interval between two samples in seconds. Only required, when on-change is false.
on-change	Required	Boolean		Does Measurement-Channel only record changes in signal
sample-strategy	Required	String	one of: - min - max - average - last-known-value	Signal sampling strategy
signal	Required	Object	Array of Signal Object	See section 4.3 for Signal object definition
Histogram AND geo-Histogram Measurement Channel				
type	Required	String	one of: - histogram - geo-histogram	Type of the Measurement Channel needs to be histogram or geo-histogram
aggregation-strategy	Required	String	one of: - time - count - min - max	Histogram values aggregation strategy
capture-interval	Required	Number	double	Capture Interval of one Histogram. Needs to be

				larger than zero. +Infinity is valid (see IEEE 754).
dimensions	Required	Number	uint32	Dimensions of the Histogram
bins	Required	Array	Bin-Configuration Object	Array of bin configurations. Array needs to contain exactly one configuration for every dimension/axes of the histogram. See also section 4.4.1.1
Geo-Histogram Measurement Channel				
type	Required	String	“geo-histogram”	Type of the Measurement Channel needs to be geo-histogram
geo-resolution	Required	Numeric	double	Zoom level of the geo-histogram according to section 4.2.1.3
Basic CPP Information Measurement Channel				
type	Required	String	“basic-cpp-information”	Type of the Measurement Channel needs to be basic-cpp-information
signal	Required	Object	Signal Object	See section for Signal object definition
Event Based Measurement Channel				
type	Required	String	“event-based”	Type of the Measurement Channel needs to be event-based
format	Required	String		Data type format of the samples
event-sample-strategy	Required	Event Sample	one of: - real-time-event - trigger-event - threshold-event	Event sampling strategy
comment	Optional	String	No	Description of Event Strategy
General Purpose Measurement Channel				
type	Required	String	“general-purpose”	Type of the Measurement Channel needs to be general-purpose

`signal`	Required	Any		See section 0 for Signal definition

2.4 Data Layer Specification

"*Data Packages contain the actual data of Signal measurements. As Signals are the information providers and Measurement Channels define the process of data acquisition from those Signals, Data Packages provide a structure for storing the data. In addition, they provide meta / header information containing time of recording, data ownership information, etc. Data Packages contain data from exactly one Measurement Channel*". This leads to six different types of Data Packages that are defined similar as the Measurement Channels:

- Time Series Data Package
- Histogram Data Package
- Geo-Histogram Data Package
- Event based data Package
- Basics CPP information Data Package
- General Purpose Data Package

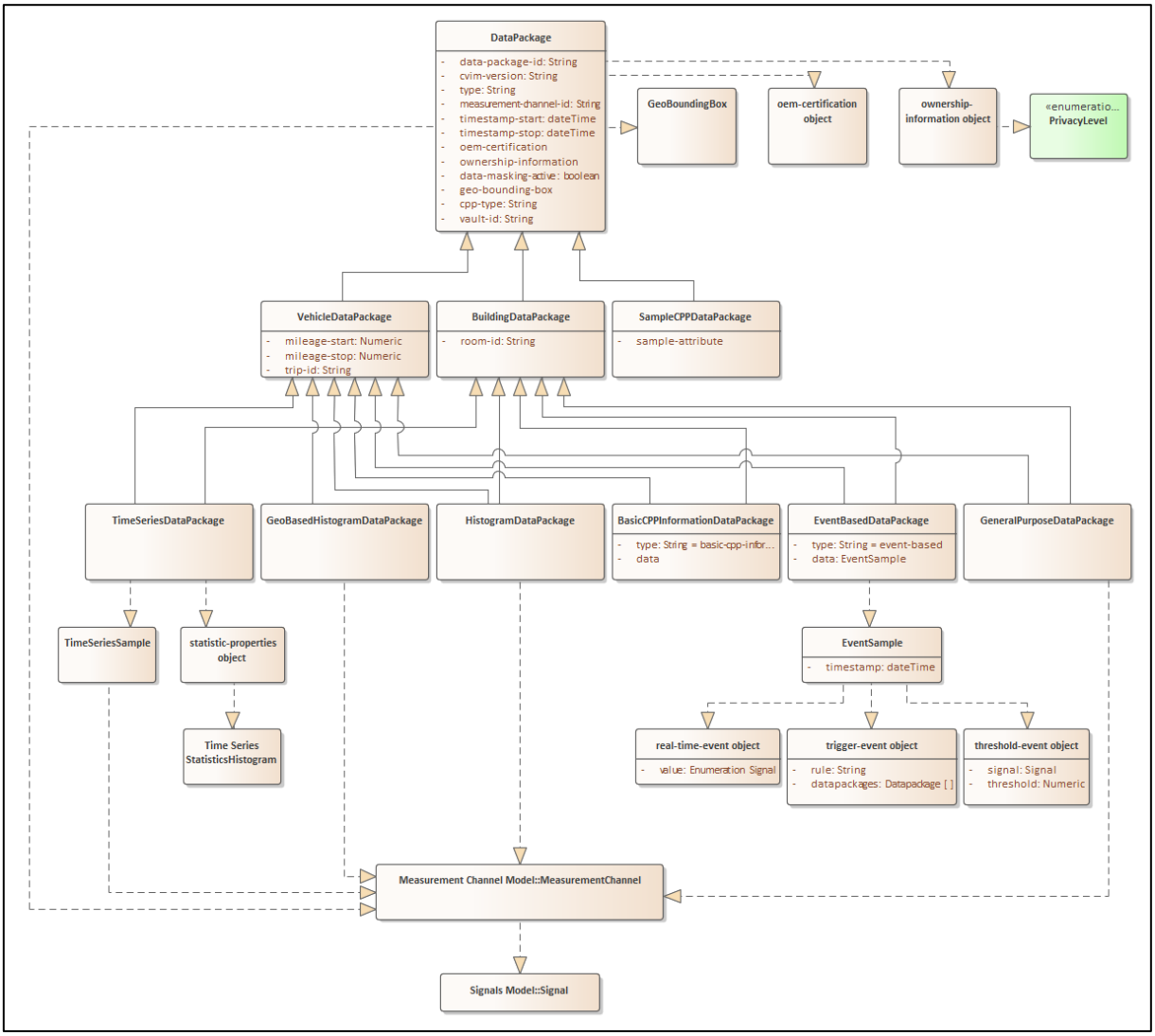

Figure 4. Data Package UML Model.

The new data-package definition includes cpp-type to indicate is the data-package belongs to a “building” or a “vehicle” cpp and two new definition of data-package type, basics-cpp-information data-package and the event-based data-package.

The data of basics-cpp-information data-package depends of the signal type definition of the measurement channel - numeric, enumeration, information or general-purpose.
Event-based data-package has an additional property, “event-sample-strategy”, to indicate three different type of event, real-time, trigger and threshold. According to the type of event, the “data” object has two mandatory properties, “timestamp” and “value”, and an optional property named “datapackages” that is an array of the data-packages that triggers the “trigger-event”. See Table 4 and Table 5 for the entire specification.

The “building” CPP devices provides a set of sensors distributed along the rooms of the building, in order to identify the devices of the same room a new property has been included, “room-id”.

Table 4. Data Package definition.

Property	Occurrence	Type	Format	Description
Common Properties				
data-package-id	Required	String	UUID	Identifier of the Data Package. Unique per Cloud Storage Vault, Set by Cloud Storage Provider
cvim-version	Required	String	version	The name of the property is for backward compatibility with CVIM. Version 0.0.1 to 1.2.0 are CVIM and CIDM from 1.2.1
type	Required	String	one of: - time-series - histogram - geo-histogram - general-purpose - event-based - basic-cpp-information	Type of the Data Package
vault-id	Required	String	UUID	ID of the Cloud Storage Vault, where the data is stored in.
cpp-id	Optional	String	any	ID of the CPP
cpp-type	Required	String	one of: - vehicle - building	Type of the CPP
trip-id	Optional	String	any	Trip-ID of the User
room-id	Optional	String	any	ID of the room in a building where the measurement data was collected
measurement-channel-id	Required	String		Identifier of the Measurement Channel whose data is inside this data package
mileage-start	Optional	Number	double	Mileage at the start of the measurement in kilometres (km)
mileage-stop	Optional	Number	double	Mileage at end of measurement (km)
geo-bounding-box	Optional	Object	Geo-Bounding-Object	Geographic bounding box, see section 4.5.1.1

Property	Occurrence	Type	Format	Description
Common Properties				
location	Optional	Object	Location-Object	Single location including latitude and longitude
oem-certification	Optional	Object	OEM-Certification-Object	OEM Certification, see section 4.5.1.2
data-ownership-information	Optional	Object	Ownership-Information - Object	Data Ownership Information, see section 5.4.1.3
expiration-date	Optional	String	date-time	Data expiration date
data-masking-active	Optional	Boolean		Indicates status of data-masking (true = active)
Time Series Data Package				
type	Required	String	“time-series”	Type of the Measurement Channel needs to be time-series
timestamp-start	Required	String	date-time	Measurement start time
timestamp-stop	Required	String	date-time	Measurement stop time
number-of-samples	Required	Number	uint32	Number of samples that are stored in data
statistic-properties	Optional	Object	statistic-properties-object	Provides statistic properties about the data, see section 4.5.1.4
data	Required	Array	time-series key-value-pair - object	Array of time-series-data Objects. The size of the array mist equal number of samples, see section 0
Histogram Data Package				
type	Required	String	histogram	Type of the Measurement Channel needs to be histogram
timestamp-start	Required	String	date-time	Measurement start time
timestamp-stop	Required	String	date-time	Measurement stop time

Property	Occurrence	Type	Format	Description
Common Properties				
`data`	Required	(Multi-dimen-sional) Array	Number	Array containing the bin counts. Size of array must match the dimension and bin configuration of the related Measurement Channel. Number format depends on Histogram aggregation-strategy
Geo-Histogram Data Package				
`type`	Required	String	geo-histogram	Type of the Measurement Channel needs to be geo-histogram
`timestamp-start`	Required	String	date-time	Measurement start time
`timestamp-stop`	Required	String	date-time	Measurement stop time
`data`	Required	(Multi-dimen-sional) Array	Number	Array containing the bin counts. Size of array must match the dimension and bin configuration of the related Measurement Channel. Number format depends on Histogram aggregation-strategy. The outer most dimension is the geo-dimension. It must match in its size the size of the geo-tiles array.
`geo-tiles`	Required	Array	Geo-Tile Object	Array of geo-tile objects. Only visited tiles are included.
Basic CPP Information Data Package				
`type`	Required	String	“basic-cpp-information”	Type of the Measurement Channel needs to be basic-cpp-information
`timestamp`	Required	String	date-time	Measurement date time
`data`	Required	Any		Data depends on the type of signal of the measurement channel

Property	Occurrence	Type	Format	Description
Common Properties				
Event Based Data Package				
`type`	Required	String	“event-based”	Type of the Measurement Channel needs to be event-based
`timestamp`	Required	String	date-time	Measurement date time
`event-sam-ple-strategy`	Required		one of: - real-time-event - trigger-event - threshold-event	Event sampling strategy
`data`	Required	Object	Event Sample Object	event-based data Object indicating an event
General Purpose Data Package				
`type`	Required	String	“general-purpose”	Type of the Measurement Channel needs to be general-purpose
`timestamp`	Required	String	date-time	Measurement date time
`data`	Required	Any	time	Datatype depends on Measurement Channel

Table 5. Event Sample Object

Property	Occur-rence	Type	Format	Description
`timestamp`	Required	timestamp	date-time	Timestamp of the event
`value`	Required	string		e.g. “*Ignition On”, “Wipers Off”*
`datapackages`	Optional	Array of data-package		

3 Annex

3.1 OpenAPI Specification of the Company Backend REST API (yaml)

```
---
swagger: "2.0"
info:
  version: 3.0.0
  title: Cloud Storage Provider API Definition
  contact:
    name: Elisa Herrmann
    url: http://www.cross-cpp.eu
    email: elisa.herrmann@atos.net
host: cloudstorage-api.datagora.eu
basePath: /
schemes:
- https
consumes:
- application/json
produces:
- application/json
security:
- api_key: []
paths:
  /users:
    get:
      tags:
      - User Management
      summary: Retrieve all users
      description: This functionalities provides a list of all users within this Cloud Storage Pro-
vider. This API call is not specified and may only be used by the Cloud Storage Provider for
internal user management. Returns an array of `User` objects.
      operationId: usersGET
      parameters: []
      responses:
        "200":
          description: OK
          schema:
            type: array
            items:
              $ref: '#/definitions/FullUser'
        "401":
          description: Unauthorized
        "403":
          description: Forbidden
        "404":
          description: Not found
        default:
          description: Unexpected error
      x-swagger-router-controller: Default
    post:
```

```
    tags:
    - User Management
    summary: Create new user
    description: API call registers a new user and creates new Cloud Storage Vault at the Pro-
vider. This API call is not specified and is intended to be used for internal user management.
    operationId: usersPOST
    parameters:
    - in: body
      name: User
      description: Defines full name, login and password of the user.
      required: true
      schema:
        $ref: '#/definitions/User'
    responses:
      "200":
        description: OK
        schema:
          $ref: '#/definitions/FullUser'
      "400":
        description: Bad request
      "401":
        description: Unauthorized
      "403":
        description: Forbidden
      "409":
        description: Conflict - User already exists
      default:
        description: Unexpected error
    x-swagger-router-controller: Default
  /users/access:
    post:
      tags:
      - Contract Management
      summary: Grant access permission
      description: Acquires access permissions to the `api_key` that is used in this request. If
`key`is a `vault-write-key` `write permission`is granted, if `key`is `vault-read-key` `read permis-
sion`is granted.
      operationId: usersAccessPOST
      parameters:
      - in: body
        name: key
        description: (Read or write) Access key to one Cloud Storage Vault.
        required: true
        schema:
          $ref: '#/definitions/key'
      responses:
        "200":
          description: OK
          schema:
            $ref: '#/definitions/inline_response_200'
        "400":
          description: Bad request
        "401":
```

```
        description: Unauthorized
      "403":
        description: Forbidden
      "404":
        description: Not found
      default:
        description: Unexpected error
    x-swagger-router-controller: Default
  /users/access/{key}:
    get:
      tags:
      - Contract Management
      summary: Validate access token
      description: tbd
      operationId: usersAccessGET
      parameters:
      - name: key
        in: path
        description: Access key for Cloud Storage Vault
        required: true
        type: string
        format: uuid
      responses:
        "200":
          description: OK
          schema:
            $ref: '#/definitions/inline_response_200_1'
      x-swagger-router-controller: Default
    delete:
      tags:
      - Contract Management
      summary: Release access permissions
      description: Releases access permissions of the `api_key` that is used in this request. If
`key`is a `vault-write-key` `write permission`is released, if `key`is `vault-read-key` `read per-
mission`is released.
      operationId: usersAccessKeyDELETE
      parameters:
      - name: key
        in: path
        description: Access key for Cloud Storage Vault
        required: true
        type: string
        format: uuid
      responses:
        "200":
          description: OK
        "400":
          description: Bad request
        "401":
          description: Unauthorized
        "403":
          description: Forbidden
        "404":
```

```
        description: Not found
      default:
        description: Unexpected error
    x-swagger-router-controller: Default
  /datapackages:
    post:
      tags:
      - Data Storage Interface
      summary: Push data packages into Cloud Storage
      description: This API call enables OEMs to write data packages into the cloud. Data packages are stored within a CIDM container structure. Every data package needs to contain the correct vault-id. The OEM needs write permission to the user's Cloud Storage Vault. The Cloud Storage Provider assignes an unique datapackage-id to every delivered data package.
      operationId: datapackagesPOST
      parameters:
      - in: body
        name: Data Package Container
        description: CIDM container structure containing valid and json encoded CIDM data packages
        required: true
        schema:
          type: array
          items:
            $ref: '#/definitions/DataPackage'
      responses:
        "200":
          description: OK
        "400":
          description: Bad request
        "401":
          description: Unauthorized
        "403":
          description: Forbidden
        default:
          description: Unexpected error
      x-swagger-router-controller: Default
  /datapackages/{datapackageid}:
    get:
      tags:
      - Data Provisioning Interface
      summary: Get data package
      description: Retrieve one data package with `datapackage-id`.
      operationId: datapackagesIdGET
      parameters:
      - name: datapackageid
        in: path
        description: Unique data package identifier
        required: true
        type: string
      - name: metadata
        in: query
        description: Retrieve *only* metadata, default=false
        required: false
```

```
        type: boolean
        default: false
      responses:
        "200":
          description: OK
          schema:
            $ref: '#/definitions/DataPackage'
        "400":
          description: Bad request
        "401":
          description: Unauthorized
        "403":
          description: Forbidden
        "404":
          description: Not found
        default:
          description: Unexpected error
      x-swagger-router-controller: Default
  /datapackages/query:
    post:
      tags:
      - Data Provisioning Interface
      summary: Query data packages
      description: Allows searching for data packages. Cloud Storage Provider will only include
Cloud Storage Vaults into search where `api_key` has read access.
      operationId: datapackagesQueryPOST
      parameters:
      - in: body
        name: query
        description: Query for data packages
        required: true
        schema:
          $ref: '#/definitions/Query'
      responses:
        "200":
          description: OK
          schema:
            type: array
            items:
              $ref: '#/definitions/DataPackage'
        "400":
          description: Bad request
        "401":
          description: Unauthorized
        "403":
          description: Forbidden
        "404":
          description: Not found
        default:
          description: Unexpected error
      x-swagger-router-controller: Default
  /datapackages/query_stream:
    post:
```

```
    tags:
    - Data Provisioning Interface
    summary: Query data packages
    description: Allows searching for data packages. Cloud Storage Provider will only include
Cloud Storage Vaults into search where `api_key` has read access.
    operationId: datapackagesQueryStreamPOST
    parameters:
    - in: body
      name: query
      description: Query for data packages
      required: true
      schema:
        $ref: '#/definitions/Query'
    responses:
      "200":
        description: OK
        schema:
          type: array
          items:
            $ref: '#/definitions/DataPackage'
      "400":
        description: Bad request
      "401":
        description: Unauthorized
      "403":
        description: Forbidden
      "404":
        description: Not found
      default:
        description: Unexpected error
    x-swagger-router-controller: Default
  /datapackages/basicCppQuery:
    post:
      tags:
      - Data Provisioning Interface
      summary: Query basic CPP information data packages
      description: Allows searching for data packages. Cloud Storage Provider will only include
Cloud Storage Vaults into search where `api_key` has read access.
      operationId: basicCppQueryPOST
      parameters:
      - in: body
        name: query
        description: Query for data packages
        required: true
        schema:
          $ref: '#/definitions/Query'
      responses:
        "200":
          description: OK
          schema:
            type: array
            items:
              $ref: '#/definitions/DataPackage'
```

```
        "400":
          description: Bad request
        "401":
          description: Unauthorized
        "403":
          description: Forbidden
        "404":
          description: Not found
        default:
          description: Unexpected error
      x-swagger-router-controller: Default
  /datapackages/basicCppQuery_stream:
    post:
      tags:
      - Data Provisioning Interface
      summary: Query basic CPP information data packages
      description: Allows searching for data packages. Cloud Storage Provider will only include
Cloud Storage Vaults into search where `api_key` has read access.
      operationId: basicCppQueryStreamPOST
      parameters:
      - in: body
        name: query
        description: Query for data packages
        required: true
        schema:
          $ref: '#/definitions/Query'
      responses:
        "200":
          description: OK
          schema:
            type: array
            items:
              $ref: '#/definitions/DataPackage'
        "400":
          description: Bad request
        "401":
          description: Unauthorized
        "403":
          description: Forbidden
        "404":
          description: Not found
        default:
          description: Unexpected error
      x-swagger-router-controller: Default
  /notifications:
    post:
      tags:
      - Notifications
      summary: Subscribe Push Notification
      description: Functionality of the marketplace to subscribe to push notification events. Push
notifications are sent, whenever users put data into their Cloud Storage Vaults.
      operationId: notificationsPOST
      parameters:
```

```
    - in: body
      name: config
      description: Push Notification URL
      required: true
      schema:
        $ref: '#/definitions/config'
    responses:
      "200":
        description: OK
      "400":
        description: Bad request
      "401":
        description: Unauthorized
      "403":
        description: Forbidden
      default:
        description: Unexpected error
    x-swagger-router-controller: Default
  delete:
    tags:
    - Notifications
    summary: Unsubscribe Push Notification
    description: No more push notifications will be sent from the Cloud Storage Provider to the
marketplace.
    operationId: notificationsDELETE
    parameters: []
    responses:
      "200":
        description: OK
      "401":
        description: Unauthorized
      "403":
        description: Forbidden
      "404":
        description: Not found
      default:
        description: Unexpected error
    x-swagger-router-controller: Default
 /access:
  get:
    tags:
    - Contract Management
    summary: Validate authentication token
    description: t.b.d.
    operationId: accessGET
    parameters: []
    responses:
      "200":
        description: OK
        schema:
          $ref: '#/definitions/inline_response_200_2'
      "403":
        description: Forbidden / No authorization token in header
```

```
        "404":
          description: Authorization token not found!
        "500":
          description: Internal Error
      x-swagger-router-controller: Default
securityDefinitions:
  api_key:
    description: Provides authentification for OEM and Marketplace. Must be sent in HTTP
header.
    type: apiKey
    name: Authentication
    in: header
definitions:
  FullUser:
    allOf:
    - $ref: '#/definitions/User'
    - {}
  User:
    type: object
    required:
    - full-name
    - login-name
    - password
    properties:
      full-name:
        type: string
        description: Full name of the user
      login-name:
        type: string
        format: email
        description: Login name of the user (e-mail)
      password:
        type: string
        description: Password of the user
    example:
      password: password
      full-name: full-name
      login-name: login-name
  DataPackage:
    type: object
    required:
    - cpp-type
    - cvim-version
    - data
    - measurement-channel-id
    - type
    - vault-id
    properties:
      vault-id:
        type: string
        description: Cloud Storage Vault ID, where the data packages are pushed into.
      datapackage-id:
        type: string
```

```
      description: Unique identifier of the data package. Property is set by Cloud Storage Provider.
      readOnly: true
    data:
      type: object
      description: CIDM data
      properties: {}
    cvim-version:
      type: string
      description: Version of the CIDM protocol >=1.1.2
    type:
      type: string
      description: Type of the Data Package
    cpp-type:
      type: string
      description: Type of the CPP device, "vehicle" or "builging"
    cpp-id:
      type: string
      description: CPP ID for identifying vehicles or bulding of the same owner and vault-id
    measurement-channel-id:
      type: string
      description: Identifier of the Measurement Channel whose data is inside this data package
    timestamp:
      type: string
      description: Measurement timestamp (basic-cpp-information and event-based)
    timestamp-start:
      type: string
      description: Measurement start time
    timestamp-stop:
      type: string
      description: Measurement stop time
    mileage-start:
      type: number
      description: Mileage at the start of measurement in kilometres (km)
    mileage-stop:
      type: number
      description: Mileage at the end of measurement in kilometres (km)
    geo-bounding-box:
      type: object
      description: geographic bounding box (see reference manual section 6.5.1.1)
      properties: {}
    room-id:
      type: string
      description: Identifier of the room in a building cpp-type
    oem-certification:
      type: object
      description: OEM certification (see reference manual section 6.5.1.2)
      properties: {}
    ownership-information:
      type: object
      description: Data Ownership Information (see reference manual section 6.5.1.3)
      properties: {}
```

```
    expiration-date:
      type: string
      description: Data expiration date
    data-masking-active:
      type: boolean
      description: Indicates status of data-masking (true = active)
  example:
    datapackage-id: datapackage-id
    vault-id: vault-id
    data: '{}'
    data-masking-active: true
    cvim-version: cvim-version
    type: type
    mileage-stop: 6.027456183070403
    measurement-channel-id: measurement-channel-id
    timestamp-start: timestamp-start
    geo-bounding-box: '{}'
    mileage-start: 0.8008281904610115
    oem-certification: '{}'
    ownership-information: '{}'
    timestamp-stop: timestamp-stop
    expiration-date: expiration-date
Query:
  type: object
  properties:
    datapackage-id:
      type: array
      description: Array of Data Package IDs
      items:
        type: string
        description: Data Package ID
    measurement-channel-id:
      type: array
      description: Array of Measurement Channel IDs
      items:
        type: string
        description: Measurement Channel ID
    vault-id:
      type: array
      description: Array of Cloud Storage Vault IDs
      items:
        type: string
    submit-time:
      $ref: '#/definitions/Query_submittime'
    metadata:
      type: boolean
      description: Request only metadata, default=off
      default: false
  example:
    datapackage-id:
    - datapackage-id
    - datapackage-id
    submit-time:
```

```
      min: 2000-01-23T04:56:07.000+00:00
      max: 2000-01-23T04:56:07.000+00:00
    metadata: false
    vault-id:
    - vault-id
    - vault-id
    measurement-channel-id:
    - measurement-channel-id
    - measurement-channel-id
key:
  type: object
  properties:
    vault-access-key:
      type: string
      format: uuid
      description: Access key for Cloud Storage Vault
config:
  type: object
  properties:
    handler-url:
      type: string
      format: uuid
      description: URL for push notifications
    level:
      type: string
      description: Defines the level of the notification ('id-only', 'metadata' or 'full')
Query_submittime:
  properties:
    min:
      type: string
      format: date-time
      description: Earliest Data Package submission time
    max:
      type: string
      format: date-time
      description: Latest Data Package submission time
  description: Data Package submission time
  example:
    min: 2000-01-23T04:56:07.000+00:00
    max: 2000-01-23T04:56:07.000+00:00
inline_response_200:
  type: object
  properties:
    full-name:
      type: string
    vault-id:
      type: string
      description: ID of the user's Cloud Storage Vault.
inline_response_200_1:
  type: object
  properties:
    full-name:
      type: string
```

```
    vault-id:
      type: string
      format: uuid
      description: ID of the user's Cloud Storage Vault.
    type:
      type: string
      description: '`read` or `write` access key'
    in-use:
      type: boolean
      description: '`true` when key is already in use, otherwise `false`'
inline_response_200_2:
  type: object
  properties:
    name:
      type: string
    type:
      type: string
```

3.2 CIDM v1.2.1 jsonSchema


```
{
    "$schema": "http://json-schema.org/draft-04/schema#",
    "description": "Common Industrial Data Model",
    "type": "object",
    "properties": {
        "Signal": {
            "$ref": "#/definitions/Signal"
        },
        "MeasurementChannel": {
            "$ref": "#/definitions/MeasurementChannel"
        },
        "DataPackage": {
            "$ref": "#/definitions/DataPackage"
        }
    },
    "additionalProperties": false,
    "definitions": {
        "TimeSeriesMeasurementChannel": {
            "title": "TimeSeriesChannel",
            "type": "object",
            "properties": {
                "type": {
                    "type": "string",
                    "enum": [
                        "time-series"
                    ]
                },
                "capture-interval": {
                    "type": "number"
                },
                "on-change": {
                    "type": "boolean"
                },
                "sample-strategy": {
                    "type": "string",
                    "enum": [
                        "min",
                        "max",
                        "average",
                        "last-known-value"
                    ]
                },
                "signal": {
                    "$ref": "#/definitions/Signal"
                },
                "format": {
                    "type": "string"
                },
                "dimension": {
                    "type": "number"
                }
```

```
            },
            "required": [
                "type",
                "capture-interval",
                "on-change",
                "sample-strategy",
                "signal"
            ]
        },
        "MeasurementChannel": {
            "type": "object",
            "properties": {
                "id": {
                    "type": "string"
                },
                "name": {
                    "type": "string"
                },
                "type": {
                    "type": "string",
                    "enum": [
                        "time-series",
                        "histogram",
                        "geo-histogram",
                        "general-purpose",
                        "event-based",
                        "basic-cpp-information"
                    ]
                },
                "comment": {
                    "type": "string"
                }
            },
            "required": [
                "id",
                "name",
                "type"
            ],
            "oneOf": [
                {
                    "$ref": "#/definitions/TimeSeriesMeasurementChannel"
                },
                {
                    "$ref": "#/definitions/HistogramMeasurementChannel"
                },
                {
                    "$ref": "#/definitions/GeoBasedHistogramMeasurementChan-
nel"
                },
                {
                    "$ref": "#/definitions/GeneralPurposeMeasurementChannel"
                },
```

```
                {
                    "$ref": "#/definitions/BasicCppInformationMeasurement-
Channel"
                },
                {
                    "$ref": "#/definitions/EventBasedMeasurementChannel"
                }
            ]
        },
        "HistogramMeasurementChannel": {
            "type": "object",
            "properties": {
                "type": {
                    "type": "string",
                    "enum": [
                        "histogram",
                        "geo-histogram"
                    ]
                },
                "aggregation-strategy": {
                    "type": "string",
                    "enum": [
                        "time",
                        "count",
                        "min",
                        "max"
                    ]
                },
                "capture-interval": {
                    "type": "number"
                },
                "dimensions": {
                    "type": "integer",
                    "minimum": 1
                },
                "bins": {
                    "type": "array",
                    "minItems": 1,
                    "items": {
                        "type": "object",
                        "properties": {
                            "type": {
                                "type": "string",
                                "enum": [
                                    "linear",
                                    "logarithmic",
                                    "custom"
                                ]
                            },
                            "lower-bound": {
                                "type": "number"
                            },
```

```
        "upper-bound": {
            "type": "number"
        },
        "signal": {
            "$ref": "#/definitions/Signal"
        },
        "number-of-bins": {
            "type": "integer",
            "minimum": 0
        },
        "alternative-bin-labels": {
            "type": "array",
            "items": {
                "type": "string"
            }
        }
    },
    "required": [
        "type",
        "lower-bound",
        "upper-bound",
        "signal",
        "number-of-bins"
    ],
    "oneOf": [
        {
            "type": "object",
            "properties": {
                "type": {
                    "type": "string",
                    "enum": [
                        "linear",
                        "logarithmic"
                    ]
                }
            },
            "required": [
                "type"
            ]
        },
        {
            "type": "object",
            "properties": {
                "type": {
                    "type": "string",
                    "enum": [
                        "custom"
                    ]
                },
                "custom-bounds": {
                    "type": "array",
                    "items": {
```

```
                                    "type": "number"
                                }
                            }
                        },
                        "required": [
                            "type",
                            "custom-bounds"
                        ]
                    }
                ]
            }
        }
    },
    "required": [
        "type",
        "aggregation-strategy",
        "capture-interval",
        "dimensions",
        "bins"
    ],
    "additionalProperties": false
},
"GeoBasedHistogramMeasurementChannel": {
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "geo-histogram"
            ]
        },
        "geo-resolution": {
            "type": "number"
        }
    },
    "required": [
        "type",
        "geo-resolution"
    ],
    "additionalProperties": false,
    "allOf": [
        {
            "$ref": "#/definitions/HistogramMeasurementChannel"
        }
    ]
},
"GeneralPurposeMeasurementChannel": {
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
```

```
                    "general-purpose"
                ]
            },
            "signal": {
                "$ref": "#/definitions/Signal"
            }
        },
        "required": [
            "type",
            "signal"
        ],
        "additionalProperties": false
    },
    "BasicCppInformationMeasurementChannel": {
        "type": "object",
        "properties": {
            "type": {
                "type": "string",
                "enum": [
                    "basic-cpp-information"
                ]
            },
            "signal": {
                "$ref": "#/definitions/Signal"
            }
        },
        "required": [
            "type",
            "signal"
        ],
        "additionalProperties": false
    },
    "EventBasedMeasurementChannel": {
        "type": "object",
        "properties": {
            "type": {
                "type": "string",
                "enum": [
                    "event-based"
                ]
            },
            "format": {
                "type": "string"
            },
            "event-sample-strategy": {
                "type": "string",
                "enum": [
                    "real-time-event",
                    "trigger-event",
                    "threshold-event"
                ]
            },
```

```
            "comment": {
                "type": "string"
            }
        },
        "required": [
            "type",
            "signal",
            "event-sample-strategy"
        ],
        "additionalProperties": false
    },
    "DataPackage": {
        "type": "object",
        "properties": {
            "datapackage-id": {
                "type": "string"
            },
            "vault-id": {
                "type": "string"
            },
            "trip-id": {
                "type": "string"
            },
            "cpp-id": {
                "type": "string"
            },
            "cvim-version": {
                "type": "string",
                "enum": [
                    "1.0.0",
                    "1.0.1",
                    "1.2.0",
                    "1.2.1"
                ]
            },
            "cpp-type": {
                "type": "string",
                "enum": [
                    "vehicle",
                    "building"
                ]
            },
            "type": {
                "type": "string",
                "enum": [
                    "time-series",
                    "histogram",
                    "geo-histogram",
                    "general-purpose",
                    "event-based",
                    "basic-cpp-information"
                ]
```

```
},
"measurement-channel-id": {
    "type": "string"
},
"submit-time": {
    "type": "string",
    "format": "date-time"
},
"mileage-start": {
    "type": "number"
},
"room-id": {
    "type": "string"
},
"mileage-stop": {
    "type": "number"
},
"geo-bounding-box": {
    "type": "object",
    "properties": {
        "latitude-min": {
            "type": "number"
        },
        "latitude-max": {
            "type": "number"
        },
        "longitude-min": {
            "type": "number"
        },
        "longitude-max": {
            "type": "number"
        },
        "altitude-min": {
            "type": "number"
        },
        "altitude-max": {
            "type": "number"
        }
    },
    "additionalProperties": false
},
"location": {
    "type": "object",
    "properties": {
        "latitude": {
            "type": "number"
        },
        "longitude": {
            "type": "number"
        }
    },
    "additionalProperties": false
```

```
},
"oem-certification": {
    "type": "object",
    "properties": {
        "signature": {},
        "checksum": {},
        "sequence-number": {}
    },
    "additionalProperties": false
},
"expiration-date": {
    "type": "string",
    "format": "date-time"
},
"data-ownership-information": {
    "type": "object",
    "properties": {
        "privacy-veto-rights": {
            "type": "object",
            "properties": {
                "consent-level": {
                    "type": "string",
                    "enum": [
                        "public",
                        "shared",
                        "private"
                    ]
                },
                "data-format": {
                    "type": "string",
                    "enum": [
                        "time-series",
                        "histogram"
                    ]
                },
                "jurisdiction": {
                    "type": "string",
                    "enum": [
                        "Europe",
                        "any"
                    ]
                },
                "storage-constraint": {
                    "type": "string",
                    "enum": [
                        "OEM storage",
                        "Personal storage"
                    ]
                }
            },
            "required": [
                "consent-level"
```

```
        ]
    },
    "copyright-stakeholders": {
        "type": "array",
        "items": [
            {
                "type": "object",
                "properties": {
                    "name": {
                        "type": "string"
                    },
                    "status": {
                        "type": "string"
                    }
                },
                "required": [
                    "name",
                    "status"
                ],
                "additionalProperties": false
            }
        ]
    },
    "data-stakeholders": {
        "type": "array",
        "items": [
            {
                "type": "object",
                "properties": {
                    "name": {
                        "type": "string"
                    },
                    "status": {
                        "type": "string"
                    }
                },
                "required": [
                    "name",
                    "status"
                ],
                "additionalProperties": false
            }
        ]
    },
    "data-privacy-level": {
        "type": "string",
        "enum": [
            "public",
            "shared",
            "private"
        ]
    }
```

```
            },
            "additionalProperties": false
        },
        "data-masking-active": {
            "type": "boolean"
        },
        "signatures": {
            "type": "array",
            "items": [
                {
                    "type": "object",
                    "properties": {
                        "signatory": {
                            "type": "string"
                        },
                        "checksum": {
                            "type": "string"
                        },
                        "signature": {
                            "type": "string"
                        }
                    },
                    "required": [
                        "signatory",
                        "checksum",
                        "signature"
                    ],
                    "additionalProperties": false
                }
            ]
        }
    },
    "required": [
        "cvim-version",
        "type",
        "measurement-channel-id",
        "vault-id",
        "cpp-type"
    ],
    "oneOf": [
        {
            "type": "object",
            "properties": {
                "type": {
                    "type": "string",
                    "enum": [
                        "time-series"
                    ]
                },
                "number-of-samples": {
                    "type": "integer",
                    "minimum": 1
```

```
},
"statistic-properties": {
    "type": "object",
    "properties": {
        "min": {
            "type": "number"
        },
        "max": {
            "type": "number"
        },
        "average": {
            "type": "number"
        },
        "histogram": {
            "type": "object",
            "properties": {
                "measurement-channel-id": {
                    "type": "string"
                },
                "data": {
                    "type": "array",
                    "minItems": 0,
                    "maxItems": 10,
                    "items": {
                        "type": "number"
                    }
                }
            },
            "required": [
                "measurement-channel-id",
                "data"
            ]
        }
    }
},
"data": {
    "type": "array",
    "minItems": 1,
    "items": {
        "type": "object",
        "properties": {
            "timestamp": {
                "type": "string",
                "format": "date-time"
            },
            "value": {
                "type": [
                    "array",
                    "string",
                    "number",
                    "boolean"
                ],
```

```
                        "items": {
                            "type": [
                                "string",
                                "number",
                                "boolean"
                            ],
                            "minLength": 1
                        }
                    }
                }
            }
        },
        "timestamp-start": {
            "type": "string",
            "format": "date-time"
        },
        "timestamp-stop": {
            "type": "string",
            "format": "date-time"
        }
    },
    "required": [
        "type",
        "number-of-samples",
        "data",
        "timestamp-start",
        "timestamp-stop"
    ]
},
{
    "type": "object",
    "properties": {
        "type": {
            "type": "string",
            "enum": [
                "histogram",
                "geo-histogram"
            ]
        },
        "data": {
            "type": "array",
            "items": {
                "type": [
                    "number",
                    "array"
                ],
                "minItems": 1,
                "items": {
                    "type": "number"
                }
            }
        },
```

```
            "timestamp-start": {
                "type": "string",
                "format": "date-time"
            },
            "timestamp-stop": {
                "type": "string",
                "format": "date-time"
            }
        },
        "required": [
            "type",
            "data",
            "timestamp-start",
            "timestamp-stop"
        ]
    },
    {
        "type": "object",
        "properties": {
            "type": {
                "type": "string",
                "enum": [
                    "general-purpose",
                    "basic-cpp-information"
                ]
            },
            "data": {},
            "timestamp": {
                "type": "string",
                "format": "date-time"
            }
        },
        "required": [
            "type",
            "data",
            "timestamp"
        ]
    },
    {
        "type": "object",
        "properties": {
            "type": {
                "type": "string",
                "enum": [
                    "event-based"
                ]
            },
            "data": {
                "type": "object",
                "properties": {
                    "event-type": {
                        "type": "string",
```

```
                                    "enum": [
                                        "real-time-event",
                                        "trigger-event",
                                        "threshold-event"
                                    ]
                                },
                                "event-data": {
                                    "type": "object",
                                    "properties": {
                                        "value": {
                                            "type": "string"
                                        },
                                        "event-datapackages": {
                                            "type": "array",
                                            "minItems": 1,
                                            "items": {
                                                "$ref": "#/defini-
tions/DataPackage"
                                            }
                                        }
                                    },
                                    "required": [
                                        "value"
                                    ]
                                }
                            },
                            "required": [
                                "event-type",
                                "event-data"
                            ]
                        },
                        "timestamp": {
                            "type": "string",
                            "format": "date-time"
                        }
                    },
                    "required": [
                        "type",
                        "data",
                        "timestamp"
                    ]
                }
            ]
        },
        "Signal": {
            "title": "CIDM Signal",
            "description": "Signals are the perception organs of vehi-
cles. It is their main duty to detect physical phenomenons and chemical quan-
tities by transferring them into electrical signals. They observe the envi-
ronment and gener-ate the data that is exchangeable at AutoMat's market-
place. They are one of the core components of the AutoMat project. Fig-
```

ure 11 shows the UML modelling of the signals. Within AutoMat all information pro-viders are modelled as Signal. They can be classified as static signals or changing/non-static signals, having a sample rate larger than zero. ",

```
            "type": "object",
            "properties": {
                "id": {
                    "type": "string"
                },
                "name": {
                    "type": "string"
                },
                "cpp-type": {
                    "type": "string",
                    "enum": [
                        "vehicle",
                        "building"
                    ]
                },
                "type": {
                    "type": "string",
                    "enum": [
                        "numeric",
                        "information",
                        "enumeration",
                        "general-purpose"
                    ]
                },
                "format": {
                    "type": "string"
                },
                "sample-rate": {
                    "type": "number",
                    "minimum": 0
                },
                "comment": {
                    "type": "string"
                }
            },
            "required": [
                "id",
                "name",
                "type",
                "sample-rate",
                "cpp-type"
            ],
            "oneOf": [
                {
                    "$ref": "#/definitions/NumericSignal"
                },
                {
                    "$ref": "#/definitions/EnumerationSignal"
```

```
            },
            {
                "$ref": "#/definitions/InformationSignal"
            },
            {
                "$ref": "#/definitions/GeneralPurposeSignal"
            }
        ]
    },
    "GeneralPurposeSignal": {
        "type": "object",
        "properties": {
            "type": {
                "type": "string",
                "enum": [
                    "general-purpose"
                ]
            }
        },
        "required": [
            "type"
        ]
    },
    "InformationSignal": {
        "type": "object",
        "properties": {
            "type": {
                "type": "string",
                "enum": [
                    "information"
                ]
            },
            "format": {
                "type": "string"
            }
        },
        "required": [
            "type",
            "format"
        ]
    },
    "EnumerationSignal": {
        "type": "object",
        "properties": {
            "type": {
                "type": "string",
                "enum": [
                    "enumeration"
                ]
            },
            "items": {
                "type": "array",
```

```
                    "items": {
                        "type": "string"
                    }
                }
            },
            "required": [
                "type",
                "items"
            ]
        },
        "NumericSignal": {
            "type": "object",
            "properties": {
                "type": {
                    "type": "string",
                    "enum": [
                        "numeric"
                    ]
                },
                "format": {
                    "type": "string",
                    "enum": [
                        "int",
                        "uint",
                        "float",
                        "double"
                    ]
                },
                "min": {
                    "type": "number"
                },
                "max": {
                    "type": "number"
                },
                "resolution": {
                    "type": "number"
                },
                "unit": {
                    "type": "string"
                }
            },
            "required": [
                "type",
                "format",
                "min",
                "max",
                "resolution",
                "unit"
            ]
        }
    }
}
```

CROSS-CPP

About Cross-CPP

The objective was to establish an IT environment for the integration and analytics of data streams coming from high volume (mass) products with cyber physical features, as well from Open Data Sources, aiming to offer new cross sectorial services and focusing on the commercial confidentiality, privacy and IPR and ethical issues using a context sensitive approach. The project addresses cross-stream analysis of large data volumes from mass cyber physical products (CPP) from various industrial sectors such as automotive, and home automation. The business objective of the research was to allow for analyses of such data streams in combination to other (non-industrial, open) data streams and for the establishment of diverse enhanced sectorial and cross-sectorial services. The project developed: (i) New models for integration and analytics of data streams coming from multi-sectorial CPP, including shared systems of entity identifiers applicable to multi-sectorial CPP (as well as the definition of agreed data models for data streams from multiple CPP aiming at defacto standard; (ii) Ecosystem, including a common Marketplace, and methodology to use such models to build multi-sectorial cloud based services, (iii) Toolbox for real-time and predictive cross-stream analytics, context modelling and extraction, and dynamically changing security policy, privacy and IPR conditions/rules and (iv) set of services such as services based on a combination of data streams from home automation and (electrical) vehicles to provide enhanced local weather forecast and predict and optimise energy consumptions in households. The project has built upon the results from past and current projects, where results from the project AutoMat, addressing services developed based on data streams from vehicles, were used as a basis for Cross-CPP development extend it to integrated, cross-sectorial data streams analytics. More information is available at https://cross-cpp.eu

Funded by the Horizon 2020 Framework Programme of the European Union

Every effort has been made to ensure that all statements and information contained herein are accurate, however the Cross-CPP Project Partners accept no liability for any error or omission in the same.

© 2020 Copyright in this document remains vested in the Cross-CPP Project Partners.