Progress towards Federated Logistics through the Integration of TEN-T into A Global Trade Network

D2.8 EGTN Transport Data and Knowledge Models final version

Document Summary Information

Grant Agreement No	860274	**Acronym**	PLANET
Full Title	Progress towards Federated Logistics through the Integration of TEN-T into A Global Trade Network		
Start Date	01/06/2020	**Duration**	36 months
Project URL	www.planetproject.eu		
Deliverable	D2.8 EGTN Transport Data and Knowledge Models final version		
Work Package	WP2		
Contractual due date	30.09.2022	**Actual submission date**	30.09.2022
Nature	other	**Dissemination Level**	Public
Lead Beneficiary	SIR		
Responsible Author	Andrey Tagarev (SIR)		
Contributions from	Andrey Tagarev (SIR), Mihail Radkov (SIR), Moises Sanchez(IBM)		

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement No 860274.

Revision history (including peer reviewing & quality control)

Version	Issue Date	% Complete [1]	Changes	Contributor(s)
V0.1	29/06/2022	0%	Initial Deliverable Structure	Andrey Tagarev
V0.2	07/08/2022	5%	Added ToC	Andrey Tagarev
V0.3	08/08/2022	8%	ToC updated	Andrey Tagarev
V0.4	20/08/2022	10%	ToC updated	Andrey Tagarev
V0.5	31/08/2022	40%	Contribution to chapters	Mihail Radkov
V0.6	05/09/2022	80%	Contributed to all chapters	Andrey Tagarev
V0.7	15/09/2022	85%	Added Summary, Introduction and Conclusions	Andrey Tagarev
V0.8	16/09/2022	90%	Receive comments regarding the deliverable	Moises Sanchez(IBM)
V1.0	20/09/2022	100%	Internal Review version	Andrey Tagarev

Disclaimer

Copyright message

[1] According to PLANET's Quality Assurance Process

Table of Contents

List of Figures

List of Tables

Glossary of terms and abbreviations used

Abbreviation / Term	Description
JSON	JavaScript Object Notation
KG	Knowledge Graph
GTIN	Global Trade Identification Number
SGTIN	Serialized Global Trade Identification Number
LGTIN	Lot Global Trade Identification Number
SOML	Semantic Object Modelling Language
T&L	Transport and Logistics
EGTN	Eu-Global T&L network
EPCIS	Electronic Product Code Information Services
WWIS	World Weather Information Service
API	Application Programming Interface
SOML	Semantic Object Modelling Interface

1 Executive Summary

The subject of this deliverable is to present the final form of the unified model behind the EGTN Connectivity Infrastructure. It follows on directly from D2.7 EGTN Transport Data and Knowledge v1 which presented the initial version of the model and the dataset selection process.

The work described here focuses on the data science task behind extending the model's ontological coverage, coordinating the data format between partners deploying IoT devices in the field, building custom datasets that enable the development and improvement of prediction algorithms and using the collected knowledge to gain higher order insight into the collected data through inference.

The final form of the unified model centres on the use of EPCIS 2.0 events for tracking of shipments and representing sensor readings. The other datasets such as the route model, train carbon footprint estimation, weather data and IoT event inference all tie into this central dataset collected from sensors deployed within the PLANET project. This deliverable focuses on the logic behind these connections and the specific data work required to ensure they can be made for the use cases relevant to the living labs of the project.

The purpose of this document is to not simply present the finalised model but explain the reasoning behind its development and the way in which it is intended to be used. It can serve as a guide to partners integrating with the EGTN Connectivity Infrastructure and as a guide to anyone developing similar applications that incorporate IoT devices, transport and logistics tracking, weather data, route modelling and carbon footprint estimations.

The ontological model presented in this deliverable is ideally suited for reusability by its very nature. For example, the model is not specific for train transportation and can be adjusted and reused for other transportation means such as trucks, ships and airplanes. Similarly, the EPCIS2.0 events and IoT device modelling are directly applicable in any comparable T&L context.

2 Introduction

This deliverable is part of task T2.2 EGTN IoT and Connectivity Infrastructure Components. The objective of the task is to identify, collect and integrate T&L data and models to be used in creating T&L services. The task will identify, collect and curate such models adhering to the Project's Data Management Plan.

2.1 Mapping PLANET Outputs

Purpose of this section is to map PLANET's Grant Agreement commitments, both within the formal Deliverable and Task description, against the project's respective outputs and work performed.

Table 1: Adherence to PLANET's GA Deliverable & Tasks Descriptions

PLANET GA Component Title	PLANET GA Component Outline	Respective Document Chapter(s)	Justification
DELIVERABLE			
D2.8 EGTN Transport Data and Knowledge Models final version	Final version of the strategic, meso and micro transport models	Section 6	Section 6 covers the final state of the complete transport data model that fully describes the content of the Knowledge Graph. This includes both the contents from D2.7 and the extensions developed and described in D2.8
D2.8 EGTN Transport Data and Knowledge Models final version	China-Poland train route model representation extension to model	Section 3	Section 3 covers the various aspects of modelling related to the China-Poland route. This includes the route descriptions, technical details of the vehicles, weather data stations, carbon emissions data, typical transit times and so on.
D2.8 EGTN Transport Data and Knowledge Models final version	Fully-integrated EPCIS 2.0 system with sample data	Section 4	Section 4 covers the work on a detailed example of using the newly released EPCIS 2.0 standard for IoT devices. It was directly used to inform partners in LL1 and LL3 how to handle the data-transmission aspect of their sensor deployment.
D2.8 EGTN Transport Data and Knowledge Models final version		Section 5	Section 5 continues on from the previous section and discusses the additional data modelling work based on raw sensor events. This ties into trip tracking, delivery time estimates, carbon emissions predictions and so on.
TASKS			
ST2.2.3 EGTN Transport Data and Knowledge Models as a Service	Incorporation of global standards and major ontologies in the modelling	Chapter 4	Chapter 4 describes the implementation of the newly released EPCIS 2.0 standard for IoT device data sharing, the development of sample outputs and the coordination with partners on deploying services using it.
ST2.2.3 EGTN Transport Data and Knowledge Models as a Service	Strategic, meso and micro transport models that correspond to key transport routes and corridors	Chapter 3	Chapter 3 details the extension of the route model with additional data including connection to weather stations from the weather dataset, per country modelling of

			sample train composition and the application to carbon footprint calculations
ST2.2.3 EGTN Transport Data and Knowledge Models as a Service	Single data-sharing intelligence space in the Cloud	Chapter 5 & 6	Chapter 5 follows up from Chapter 4 to describe how the collected IoT data can be leveraged to infer unique insights into the transport and logistics process. Chapter 6 presents the complete final version of the unified data model.

2.2 Deliverable Overview and Report Structure

The objective of this deliverable is to describe the progress on ST2.2.3 including the extension of the unified data model underlying the EGTN Connectivity Infrastructure, the definition of JSON formats for IoT Device output transmissions and the collection of specific external datasets required to support specific needs of partner services and use cases.

- Section 3 presents the extensions to the China-Poland route model. It focuses on the inclusion of weather stations, train composition modelling and the technical details on locomotives and cars. Finally, it describes the method for calculating the input for the delivery estimate and carbon footprint prediction algorithms.
- Section 4 presents the fully integrated EPCIS 2.0 system that follows the full transportation cycle of goods. This includes everything from company and product specifications to transportation steps such as loading and unloading, transhipping and the creation of logistic units. By utilising the EPCIS 2.0 innovations, it also covers sensor installation and readings transmission.
- Section 5 presents the event inference based on the data discussed in Section 4. This focuses on building knowledge on sensors, returnable assets and vehicles as well as tracking statistics on the scale of delivery trips.
- Section 6 presents an overview of the complete unified model underlying the EGTN Connectivity Infrastructure
- Annex I contain the detailed data samples of the EPCIS 2.0 system described in Section 4

The actual implementation of the services that ingest and process data inputs for the EGTN Connectivity Infrastructure is found in D2.6 EGTN Connectivity Infrastructure final version. Similarly, the description of the GraphQL access endpoint based on the unified model is to be found in the same deliverable.

2.3 Business Value and Reusability

The network of global transportation is an incredibly complex system made up of a huge number of moving parts. It includes multinational corporations; tracking of multi-modal globe-spanning delivery routes; details of consignments, shipments, and their related parties; details on all kinds of delivery vehicles and their current loads and possibly much more besides that. A main goal of the EGTN data work was to ensure integration between different T&L stakeholders, as well as to enable data analysis and prediction through making sure that the various participants "speak the same language." To achieve this goal, we relied on the new EPCIS 2.0 extension of the GS1 EPCIS standard as a means to enable disparate applications and organizations to uniformly exchange and process semantic data in a global transportation context. This resulted in building a global Logistics Knowledge Graph using Linked Data and semantic technologies that assists in tracking shipments, throughput of goods, emissions, and anticipation of the effects of disruptive events.

The improvement in end-to-end visibility is the most direct and obvious benefit of creating the EGTN Knowledge Graph. The combination of data sources allows answering all questions about the journey of the goods:

- What: barcode scans of goods, containers, and vehicles (received through EPCIS events)
- When and where: IoT devices physically present in the container/on the vehicle
- How: the abstract shipment route, vehicle specifications and specific knowledge of the world

Directly following on from the above is the effect on emissions reductions and increased throughput of goods. Predicting the carbon footprint or transit time of a specific package relies on prediction analytics algorithms trained on the data within the KG but the graph directly contains data on average transit times and footprints. That means it can directly answer questions such as “What will be the effect of replacing these old locomotives with this newer model?” or “How has the throughput of goods over a segment varied over time?”

The most abstract but still important benefit is the reduction of the effect of disruptions on shipment. A robust KG containing historical data can give quick and reliable answers to complex questions about the effects of road closures, extreme weather, purchase of new vehicles, changes in timetables and so on. For example, what would be the best alternative if the typically preferred route is temporarily unavailable or what would be the long-term effect of railroad repairs that decrease transit time between two major stations on an important route.

The ontological model presented in this deliverable is ideally suited for reusability by its very nature. For example, while Section 3 discusses the specific application to the China-Poland rail route modelling relevant to PLANET partners, the actual model is not specific to this. It can be used without change for train transportation over the route or other model of transport such as trucks, ships, and airplanes. Similarly, Section 4 and 5 give examples specific to the context of the project but are directly applicable in any similar T&L context.

2.4 Data Workflow Within and Beyond the EGTN Connectivity Framework

Our model provides a mechanism for harmonization of the data in the EGTN infrastructure, which further allows and improves the searchability of information provided by heterogenous sources. The standardization of the data processed by the EGTN infrastructure allows for smooth integration, analysis and prediction that can only be unlocked when the various participants “speak the same language” as defined by the unified data model.

Error! Reference source not found. shows the overall flow of the data into, through and beyond the Connectivity I nfrastructure. On top, we have the various sources feeding data into the EGTN Cloud including partner real-time streaming data (EGTN IoT Infrastructure), external real-time streaming data (Weather service), partner historical data (IoT readings), custom built static data (route model) and so on. All these sources are ingested into the Knowledge Graph and undergo a process of integration and enrichment which builds connections between the various datasets and infers extra data about them. This in turn enables the calculation of trip level data and statistics that speeds up querying and analysis of the EGTN Infrastructure knowledge hub and makes it easier for end users.

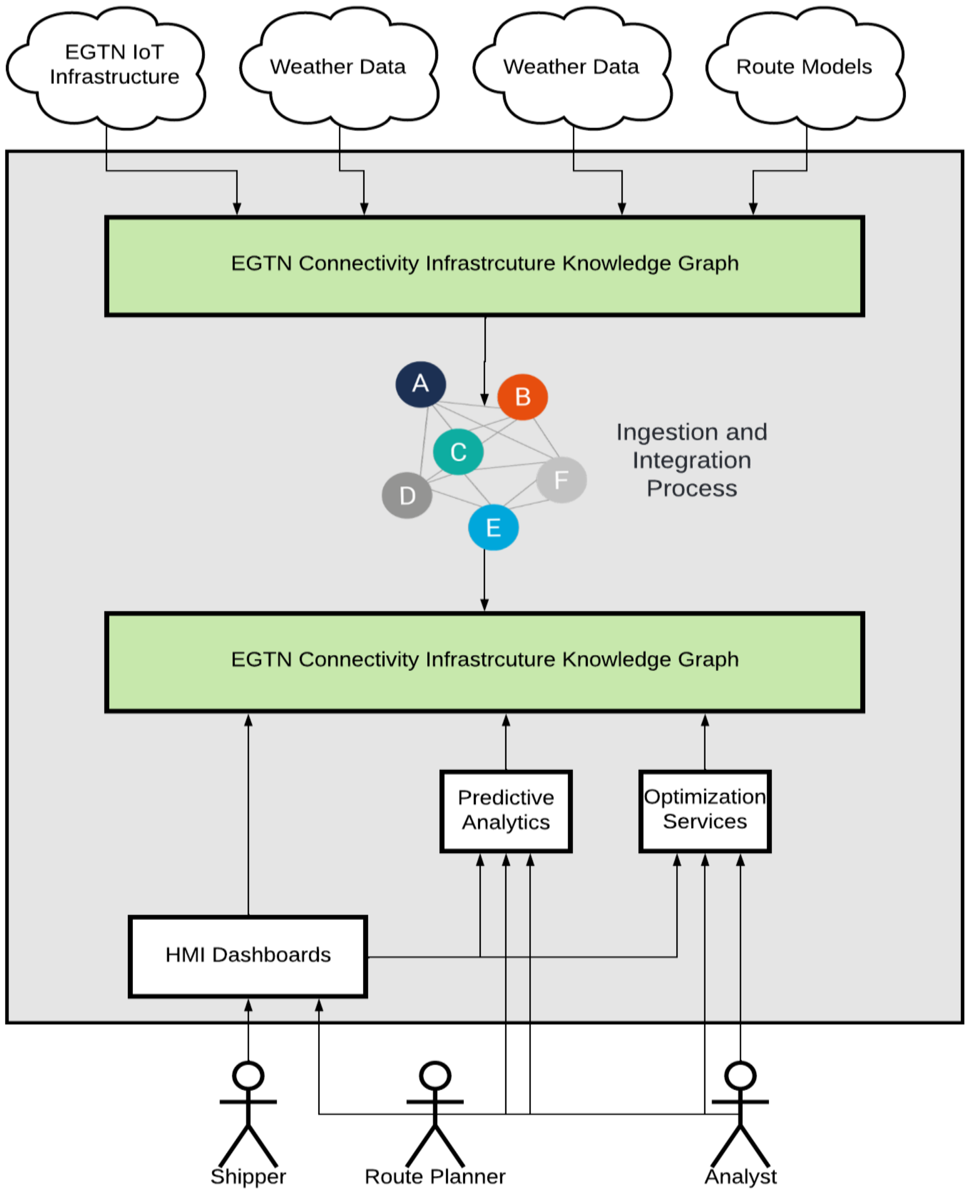

Figure 1: Connectivity Data Flow

At the bottom we have various users accessing the contents of the knowledge graph. These include predictive analytics algorithms and optimization services using detail real-time data to make predictions; route planners and analysts accessing large-scale historical data to train algorithms, build simulations and perform complex analyses; shippers and other end users accessing data through the HMI dashboards to gain immediate insight and answers.

Overall, this increased visibility serves the needs of the PLANET decision support system which allows transport and logistics stakeholders using the EGTN system to optimise planning and re-planning based on up-to-date information. There are many other opportunities for analytics and predictive systems that estimate delivery costs, suggest optimal routes according to different criteria, do real-time rerouting, calculate environmental impact, issue early warnings to inform recipients of relevant events and so on.

3 Route model extended

The initial version of the route model was described in D2.7 EGTN Transport Data and Knowledge v1. The sections there covered the actual train route including segments, major train stations and border crossings that define the overall route. The deliverable also covered the railway and transport ontologies examined and chosen for the modelling.

This deliverable covers the progress on developing the route model which includes several important ways. There is the connection between the route model and the weather dataset by associating major train stations and border crossings with appropriate nearby weather stations. The details of the route segments were significantly enriched with additional data including distance, typical transit time and average speed of trains. On a per country basis, the model now includes available locomotives with associated technical details, typical train compositions and information on typical carbon emissions in energy productions in the country. This data is combined to reach a description of the expected train compositions in each country.

This additional data was developed for the prediction of transit times and carbon footprints by providing the raw data for an initial estimate for both.

3.1 Connection to weather stations

The connection between rail and weather stations is the most straightforward addition to the route model. D2.7 described the model of both sources and the means of connecting them. The work on extending the dataset was performed in two simple steps:

1. Identify the coordinates of the rail station or border crossing
2. Using the WWIS API[2], find the closest weather station with streaming data

There were some minor challenges such as identifying the correct rail station in larger cities and finding a weather station that is not only nearby but providing streaming and historical data. The final result is that appropriate weather stations within no more than 20km were identified for each location.

This part of the data model is simple which makes the mapping very easy to extend and modify. At any time, a new connection is added, the weather ingestion service described in D2.6 EGTN Connectivity Infrastructure final version simply collects historical data. Any connections that are removed or modified, simply disconnect the given station's weather data from the route model.

[2] https://climatedata-catalogue.wmo.int/explore

3.2 Train composition simulations

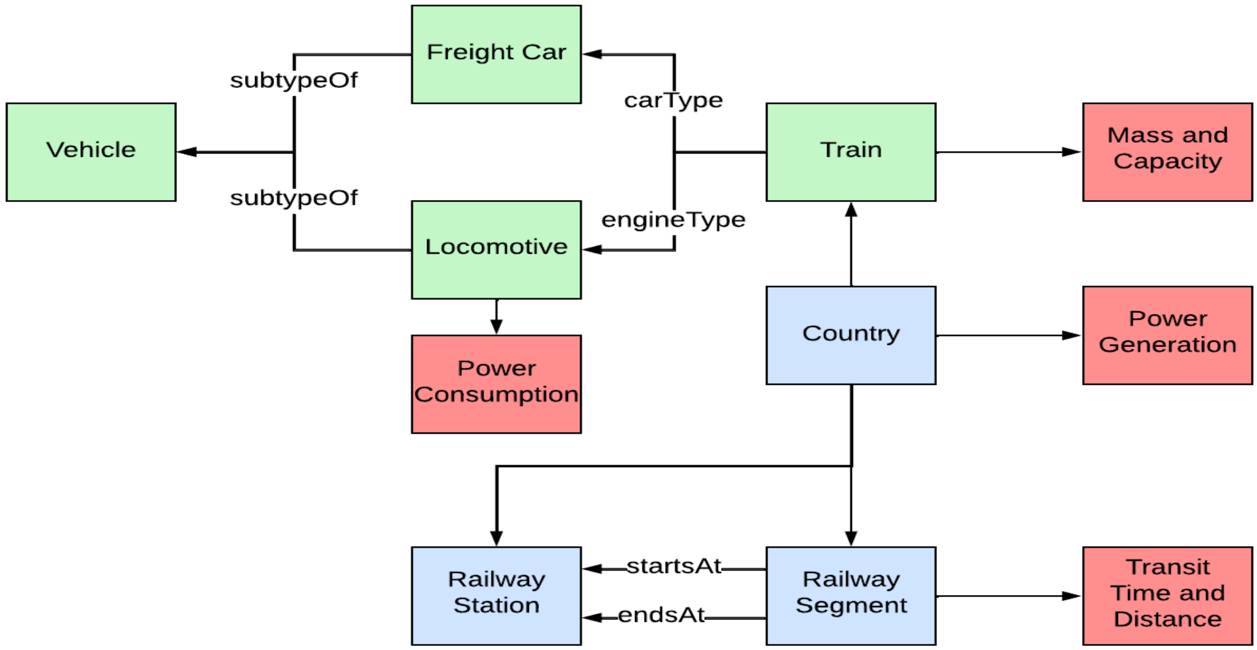

Figure 2: Extended Route Model

The largest extension of the railway model is the addition of typical compositions of trains. As Figure 2 shows in green, each country has a few typical train compositions along with information on how common they are. Each train composition includes technical information on the locomotive and freight cars including weight, carrying capacity and power consumption as well as the number of freight cars in the train.

The technical information is used to calculate electricity consumption per km, the mass of the train and the net weight of it at full capacity. Research carried out by our partners at ILIM also provided reliable estimates on the carbon emissions of electricity production per country, the numbers of each type of locomotive available for freight transport and the typical transit time between major stations. These values are added as new properties to already existing objects within the graph but are presented in red in Figure 2 to make their addition clear.

3.3 Carbon footprint estimate

The above graph is used to support carbon footprint estimation and the calculation of initial average values. The following values obtained through the efforts explained in Subsection 3.2 are used:

- Locomotive power consumption per kilometre
- Segment distance and transit time
- Average carbon emissions per KWh of electricity production
- Empty weight and carrying capacity

This allows straightforward calculation of total carbon emissions for an entire train covering a given railway segment. It also provides data necessary to calculate a per package value but leaves the actual fractional attribution up to the user.

In short, the China-Poland train route was extended with details on the train composition, technical specification of vehicles and electricity production emissions. These are all necessary to support the prediction of delivery time and carbon footprint of package transportation.

4 Fully-integrated EPCIS 2.0 system

In this section, we examine the application of the most recently released GS1 standards in the development of a fully integrated tracking system incorporating IoT sensors and real-time tracking of all assets. Within the scope of the project this is directly used for package processing and IoT devices on delivery trucks in Spain and rail transportation between China and Poland. For more information on the deployment of the physical devices and the direct data collection, see D2.4 EGTN IoT Infrastructure final version, D3.2 LL1 EGTN Solution description and test results and D3.6 LL3 EGTN Solution description and test results.

Sirma's participation in the development of the EPCIS 2.0 was presented in D2.7 EGTN Transport Data and Knowledge v1. Development work was completed and the standard finalised since publishing the previous deliverable. This section will be dedicated to defining the workflow of an end-to-end delivery tracking system using EPCIS 2.0 events both in its application to the specific needs of PLANET partners and as a general approach that can be reused in other contexts.

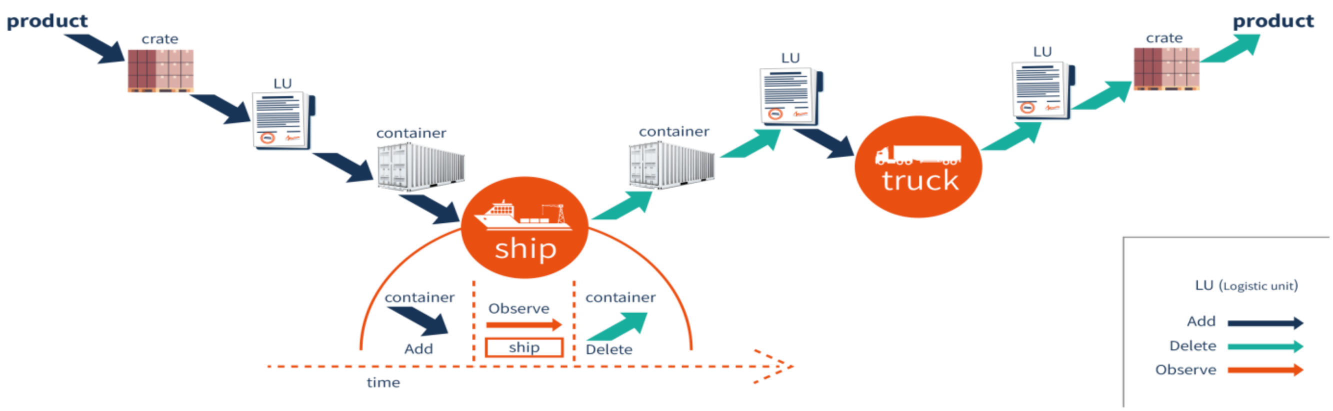

Figure 3: EPCIS 2.0 Events example

Figure 3 shows a visualisation of all the steps in the transportation of a specific product across the world using multiple steps and forms of transport. The example presented here covers everything from company description and manufacturing certification through use of vehicles, crates and containers to real-time tracking by IoT devices installed in vehicles and reusable assets.

4.1 Companies, products and certificates

While not explicitly depicted in the figure, GS1 can support and provide technical descriptions of companies, products and certification of both factories and products. The centralised company database, GEPIR[3], and product database, GDSN[4], still have limited coverage currently but their contents are growing. In the long term, it will be possible to simply reference them but in the short term it is still possible to describe the relevant content locally.

Keeping track of this information has many possible practical applications, ranging from the straightforward (e.g. generate a list of parties affected by a transportation vehicle delay) to the increasingly complex (e.g. combine product specifications with sensor temperature readings to detect issues with storage during transportation). This example includes mocked up samples of company, product and certificate objects but for practical reasons they are out-of-scope for the real-world data streaming from partners. In short, the system demonstrates and supports these data types but no real data of this type.

[3] https://www.gs1us.org/tools/gs1-company-database-gepir

[4] https://www.gs1.org/services/gdsn

4.2 Products, crates and containers

The largest part of the image is dedicated to the transition of the product between containers and transportation vehicles. These steps, typically tracked by scanning barcodes, have been a part of the EPCIS standard since its original version but here we are putting them in the context of a fully integrated system.

As the image indicates, each blue arrow represents an ADD event and each cyan arrow represents a DELETE event. These ADD and DELETE events come in pairs so at the end of the trip, each ADD comes with a corresponding DELETE. Their meaning depends on the type of the object they reference.

Firstly, the definition of products, crates and containers needs to be clarified. Product is used to mean any package that is being transported between the initial sender and final recipient. For practical purposes, any item with a GTIN (i.e., barcode) can be treated as a product. The crates and containers are simply movable storage for one or more products; as far as the system is concerned, they only differ in their identifier formatting, otherwise they are simply reusable storage.

When an EPCIS event references a crate or container, it is an Aggregation event which means it represents putting or removing an object from storage. In essence, it allows the system to know that for a given timespan a certain object was within the container and therefore any EPCIS events or sensor readings related to the container for that period also apply to the object.

Then we have the vehicle types. As far as the system is concerned, they are another type of returnable asset similar to a crate or container. Loading and unloading items is represented as Aggregation events, sensor can be installed and provide readings. The technical description of the asset will specify details on the vehicle but from a workflow perspective, the only significant difference is that transport vehicles aren't themselves loaded onto other vehicles.

Finally, we have the Logistic Units. These do not correspond to a physical object precisely. They are "created" with an Object ADD event and "destroyed" with an Object "DELETE" event. They represent the delivery of one or more returnable assets between two destinations and only exist for the duration of the trip. Their importance to the overall workflow will become clear in Section 5.

4.3 Integrated IoT sensor tracking

The figure also references the Object OBSERVE events emitted by IoT devices. These sensor readings don't occur at a specific time of the transportation but rather come in two types. One is the regular readings emitted at a certain frequency (15 minutes and 2 hours are common choices) which include the current values outputted by each sensor at that specific time. These make up the bulk of the data collected and are not very informative individually but building up large quantities of historical data is key to advanced insights.

The other type is the extreme readings prompted by the sensor detecting predefined anomalous activity. These can be triggered by excessive acceleration, extreme temperature, or high quantities of light. In each case, they are unusual events that might prompt a specific reaction or at least issuing an alert. This is important in the tracking of smart contract conditions.

There are also Association events that refer to sensors. An ADD event of that type signifies the installation of a sensor in a returnable asset while a DELETE indicates its removal. These are crucial for connecting the sensor readings to the actual transportation network since the sensor readings themselves have no reference to the object it is in.

This is a brief description of the integrated EPCIS 2.0 system that has been developed and implemented in the project. Here we presented the intended workflow of the system and Appendix I gives data samples for the EPCIS events we discussed. As mentioned, the system has been implemented in multiple partners and the EGTN Infrastructure has real world data following this format and used for collecting current and historical transportation data.

5 IoT Device Event Inference

Based on the data available in the EPCIS 2.0 events described in Section 4, the system performs inference in order to create higher order objects or enrich the contents of already existing objects.

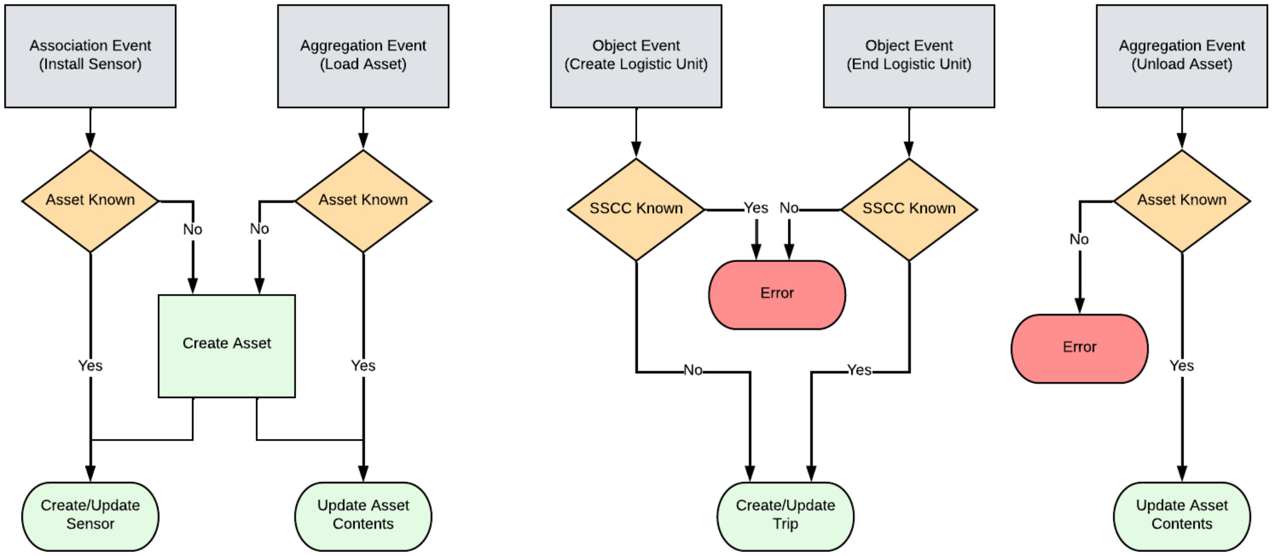

Figure 4: IoT Device Events Inference

Figure 4 shows the workflows performed for IoT events inference. Reading from left to right is the intended order of receiving events with a sensor first being installed then wares being loaded onto a returnable asset, a logistics unit being created and lasting for the duration of the trip and finally the wares being unloaded from the asset.

The different aspects of the system are colour-coded to distinguish their nature. On the top row in grey are the EPCIS 2.0 events sent to the EGTN Cloud Infrastructure. The format and use of these was discussed in Section 4. The next row in yellow involves checking the already existing contents of the Knowledge Graph in order to determine which objects will be affected by the newly ingested event. Finally, the bottom row in green is the newly inferred information added to the Knowledge Graph. The red states represent invalid data inputs that do not allow inference to take place.

5.1 Logistic Unit and Trip data

The second grouping in Figure 4 deals with the ADD and DELETE events that span the existence of a logistic unit. When properly used, each logistic unit ID should appear in one and only one pair of these events which inform us when and where the delivery trip began and when and where it ended. Based on that, the Knowledge graph can create a Trip object that uses any associated IoT devices and to calculate trip-level data and store it. This includes:

- trip duration, distance travelled and average speed
- list of extreme sensor readings
- list of returnable assets and products on the trip

These statistics are simply a useful summary that makes querying and analysing the data much quicker and easier. Technically they can be replaced with simply "heavier" queries that make the calculation on query time but the difference in accessibility and monitoring capability is significant.

5.2 Sensor Installation

The leftmost input in Figure 4 is the ingestion of an Association Event ADD event which indicates an IoT device is being installed on a returnable asset. The corresponding DELETE event would correspond to uninstalling the device and would eventually be incorporated into the system to account for sensor replacements or movements between assets. That said for simplicity the DELETE event has been omitted from the initial demonstration workflow.

Upon the installation of the sensor platform, the system checks for two things. One is any encounters of the reusable asset id- if it has not been encountered before, a new entry for the reusable asset is created although this entry will not include any technical information on the nature of the asset (e.g. whether its a container, truck, ship, etc.). The other is any encounters of the sensor platform id- if it has been encountered before, its updated to reflect its relationship to the asset. If it has not, a basic entry is created but will later be updated when the first OBSERVE events come in and give information about the sensors installed on the platform.

5.3 Asset Loading and Unloading

The logistic unit existence which defines the shipment transport is surrounded by aggregation events that define the loading (ADD) and unloading (DELETE) of returnable assets. These are required to enable the tracking of individual packages- by following the loading and unloading events we can create a list of all items contained in the shipment.

The inference steps presented in this chapter are all focused on extending the knowledge and improving the usability of the IoT device readings collected by the EGTN Connectivity Infrastructure. The goal is to ensure the device input is not only correct on a per-event level but also follows the proper workflow. This is done to ensure that the dataset is useful for the prediction analytics services that rely on the connections between individual packages/assets and the information collected by IoT devices.

6 Finalised data model

The finalised data model presents all ontologies and objects discussed in deliverables 2.7 and 2.8. The actual full data model is implemented as a SOML file which defines the interaction with the GraphQL endpoint as described in D2.6 EGTN Connectivity Infrastructure final version. The model itself is available in its entirety on the EGTN gitlab here[5].

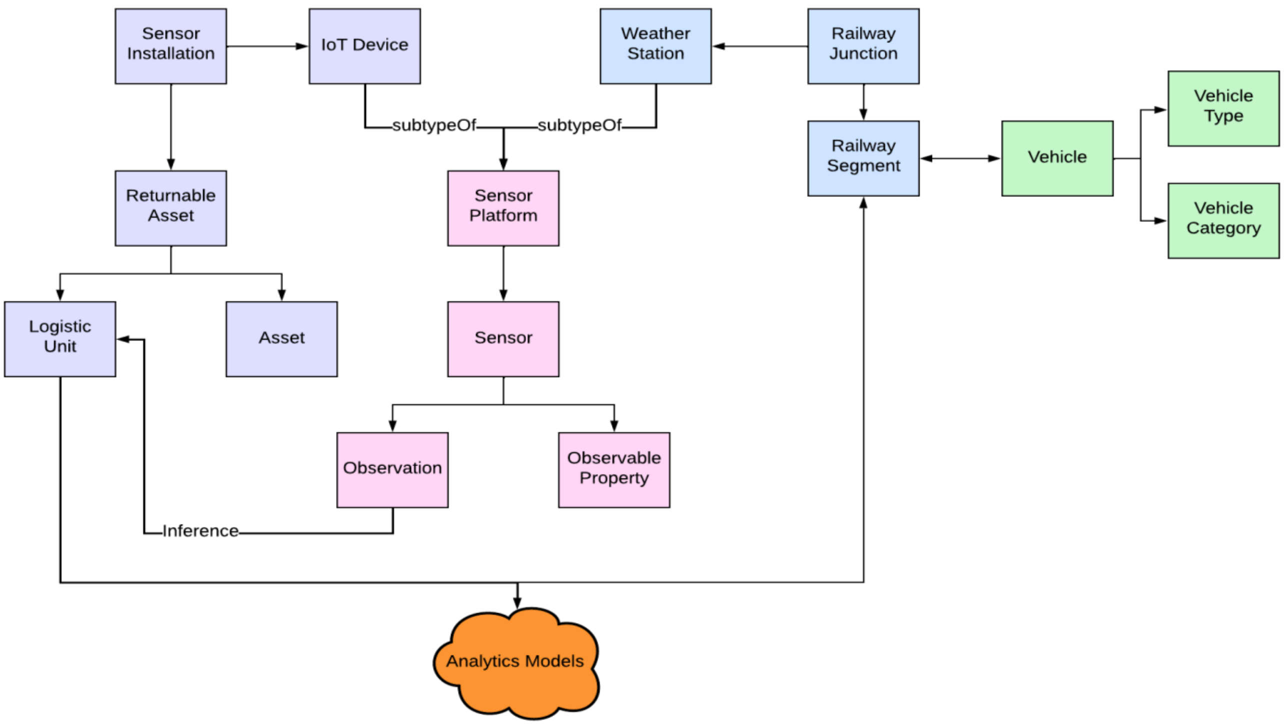

Figure 5: Simplified presentation of final data model

Figure 5 presents a simplified view of all the parts of the final data model. On the left in purple are the objects represented by the EPCIS 2.0 integrated systema (Section 4) and IoT Device event inference (Section 5). In the centre in pink are the objects representing the time series ingested from sensor platforms including both IoT devices and weather stations (D2.7). In blue is the China-Poland railway route layout (D2.7) along with mapping to weather stations and delivery time/carbon footprint averages (Subsections 3.1 and 3.3). On the right in green are the per country train composition models and locomotive/freight car technical specifications (Subsection 3.2). Finally, at the bottom we have the connection to the prediction and analytics models which ties into multiple parts of the model- on one hand historical training data and latest data for upcoming predictions is accessed through the logistic unit/trip node created from IoT devices and event inference; on the other hand, the route model provides baselines for transit times and carbon footprint and can store updated values based on newer version of the analytics models.

It is also worth going over the data sourcing for each section of the graph. The EPCIS events in purple come indirectly from living lab partners doing train (Rohlig Suus and Polish Post) and truck (DHL, Cosco and CityLogin). The time series sensor readings cover two types of data- IoT device streaming data (NGS and ILIM) and weather data (external source, World Meteorological Organisation). The route definition in blue is a custom dataset developed based on data provided by Rohlig Suus with additional manual work and the vehicle and carbon emissions data in green is a static dataset based on research performed by colleagues at ILIM.

[5]https://gitlab.ontotext.com/planet/integrated-example/-/blob/222e740e168be22a2be55b720cab928e036a8c0f/model/schema.yaml

7 Conclusions

This deliverable presented the final version of the unified model that defines the contents of the EGTN Connectivity Infrastructure. It covers the addition of vehicle technical specifications, train compositions and electricity production carbon emissions per country to the China-Poland route model; the development of a blueprint for a fully integrated EPCIS 2.0 IoT device tracking system with data samples; the building of knowledge within the Knowledge Graph based on inference and calculation of statistics. Finally, it presents a clear overview of the finalised unified model including connections between parts of the graph and the sourcing of data.

The extended route model in Section 3 along with event inference in Section 5 build a crucial base for the training and functionality of prediction and analytics services. These are currently geared to support ETA/ETD prediction and CO2 emission estimation. On one hand, the route model collects data work by partners and external knowledge for the various segments of delivery journeys. On the other hand, the event inference processes IoT device output and other time series to produce high quality easily processable trip-scale statistics and historical data for algorithm training and real-time streaming data to enable prediction.

The fully integrated EPCIS 2.0 system presented in Section 4 along with the implemented data samples included in Annex I is an important guide to the implementation of T&L systems that perform global tracking and the deployment of many IoT devices. The workflow, interaction logic, event inference described in Section 5 and JSON -ld examples would be useful for any system trying to fulfil a similar purpose.

The data model is extensible and can be easily customized by EGTN infrastructure users in case additional needs arise. It can also be reused in different contexts as part of future exploitation activities. Possible directions may include modelling a delivery vehicle fleet (like the train compositions modelled in Section 3), extending the route model to cover further rail networks or other modes of transport, and extending the event inference to make connections between sensor alarms and technical specifications of products to indicate handling issues. It also enables tying the model to existing large datasets of RDF data such as European Railway Agency's databases which cover everything from vehicle specification and country-specific laws to piece-by-piece tracking of railway infrastructure.

Reusability then comes in two forms. The model can be directly reused with no changes for the same kinds of data even if they are obtained from different sources (e.g., new IoT devices deployed to vehicles by a different shipper) or it can be extended to support new but similar use cases as mentioned before. Even when the model needs to be slightly extended to support new kinds of use cases, the newly added parts will only be minor modifications that follow the structure and logic of the already existing parts.

Annex I: EPCIS 2.0 Data Example

The data samples are provided in the json-ld format which is used for communication between the systems of the EGTN Cloud Infrastructure. While the Figure 6 presents a sample sensor installation event. The childEPCs lists the sensor platforms being installed and the parentID gives the returnable asset in which they are being installed.


```
{
  "@context": [
    "https://gs1.github.io/EPCIS/epcis-context.jsonld",
    { "gs1": "https://gs1.org/voc/" },
    { "cbv": "https://ns.gs1.org/cbv/"  },
    { "ngs": "https://ngs-sensors.it/" }
  ],
  "type": "EPCISDocument",
  "schemaVersion": "2.0",
  "creationDate": "2022-07-29T14:00:00.000000",
  "epcisBody": {
    "eventList": [
      {
        "@id": "ni:///sha-256;00?ver=CBV2.0",
        "@type": "epcis:AssociationEvent",
        "epcis:action": "ADD",
        "epcis:bizLocation": { "@id": "https://id.gs1.org/414/095200001237" },
        "epcis:readPoint": { "@id": "https://id.gs1.org/414/095200001237/254/235" },
        "epcis:bizStep": { "@id": "cbv:BizStep-assembling" },
        "epcis:disposition": { "@id": "cbv:Disp-in_progress" },
        "epcis:parentID": { "@id": "https://id.gs1.org/8003/00952000000150000123" },
        "epcis:childEPCs": { "@id": "ngs:D5:D1:B8:17:F8:18" },
        "epcis:eventTime": {
          "@value": "2022-05-01T00:00:00.000+00:00",
          "@type": "xsd:dateTimeStamp"
        },
        "epcis:recordTime": {
          "@value": "2022-05-01T00:00:00.000+00:00",
          "@type": "xsd:dateTimeStamp"
        },
        "epcis:eventTimeZoneOffset": "+00:00"
      }
    ]
  }
}
```

Figure 6: Sensor Installation Event

Figure 7 presents a sample event of loading goods on a returnable asset. The childEPCs lists the objects loaded and the parentID gives the returnable asset on which they are being loaded.


```
{
 "@context": [
    "https://gs1.github.io/EPCIS/epcis-context.jsonld",
    { "gs1": "https://gs1.org/voc/" },
    { "cbv": "https://ns.gs1.org/cbv/" },
    { "ngs": "https://ngs-sensors.it/" }
  ],
  "type": "EPCISDocument",
  "schemaVersion": "2.0",
  "creationDate": "2022-07-29T14:00:00.000000",
  "epcisBody": {
    "eventList": [
      {
        "@id": "ni:///sha-256;01?ver=CBV2.0",
        "@type": "epcis:AggregationEvent",
        "epcis:action": "ADD",
        "epcis:bizLocation": { "@id": "https://id.gs1.org/414/095200001237" },
        "epcis:readPoint": { "@id": "https://id.gs1.org/414/095200001237/254/235"
},
        "epcis:bizStep": { "@id": "cbv:BizStep-assembling" },
        "epcis:disposition": { "@id": "cbv:Disp-in_progress" },
        "epcis:parentID": { "@id":
"https://id.gs1.org/8003/009520000000150000123" },
        "epcis:childEPCs": { "@id": "https://id.gs1.org/01/0952012345678" },
        "epcis:eventTime": {
          "@value": "2022-07-15T08:00:00.000000",
          "@type": "xsd:dateTimeStamp"
        },
        "epcis:recordTime": {
          "@value": "2022-07-15T08:00:00.000000",
          "@type": "xsd:dateTimeStamp"
        },
        "epcis:eventTimeZoneOffset": "+08:00"
      }
    ]
```


Figure 7: Aggregation Event ADD

Figure 8 presents a sample event of unloading goods from a returnable asset. The childEPCs lists the objects unloaded and the parentID gives the returnable asset from which they are being unloaded.


```
{
 "@context": [
   "https://gs1.github.io/EPCIS/epcis-context.jsonld",
   { "gs1": "https://gs1.org/voc/" },
   { "cbv": "https://ns.gs1.org/cbv/" },
   { "ngs": "https://ngs-sensors.it/" }
  ],
  "type": "EPCISDocument",
  "schemaVersion": "2.0",
  "creationDate": "2022-07-29T14:00:00.000000",
  "epcisBody": {
    "eventList": [
      {
        "@id": "ni:///sha-256;05?ver=CBV2.0",
        "@type": "epcis:AggregationEvent",
        "epcis:action": "DELETE",
        "epcis:bizLocation": { "@id": "https://id.gs1.org/414/095200001237" },
        "epcis:readPoint": { "@id": "https://id.gs1.org/414/095200001237/254/235"
},
        "epcis:bizStep": { "@id": "cbv:BizStep-assembling" },
        "epcis:disposition": { "@id": "cbv:Disp-in_progress" },
        "epcis:parentID": { "@id":
"https://id.gs1.org/8003/009520000000150000123" },
        "epcis:childEPCs": { "@id": "https://id.gs1.org/01/0952012345678" },
        "epcis:eventTime": {
          "@value": "2022-07-29T12:30:00.000000",
          "@type": "xsd:dateTimeStamp"
        },
        "epcis:recordTime": {
          "@value": "2022-07-29T12:30:00.000000",
          "@type": "xsd:dateTimeStamp"
        },
        "epcis:eventTimeZoneOffset": "+02:00"
      }
```


Figure 8: Aggregation Event DELETE

Figure 9 presents a sample event of beginning a delivery trip. The childEPCs lists the returnable assets comprising the shipment and the parentID gives the SSCC of the logistic unit that defines the trip.

```
{
"@context": [
   "https://gs1.github.io/EPCIS/epcis-context.jsonld",
   { "gs1": "https://gs1.org/voc/" },
   { "cbv": "https://ns.gs1.org/cbv/" },
   { "ngs": "https://ngs-sensors.it/" }
  ],
  "type": "EPCISDocument",
  "schemaVersion": "2.0",
  "creationDate": "2022-07-29T14:00:00.000000",
  "epcisBody": {
   "eventList": [
    {
     "@id": "ni:///sha-256;02?ver=CBV2.0",
     "@type": "epcis:ObjectEvent",
     "epcis:action": "ADD",
     "epcis:bizLocation": { "@id": "https://id.gs1.org/414/095200001237" },
     "epcis:readPoint": { "@id":
"https://id.gs1.org/414/095200001237/254/235" },
     "epcis:bizStep": { "@id": "cbv:BizStep-assembling" },
     "epcis:disposition": { "@id": "cbv:Disp-in_progress" },
     "epcis:parentID": { "@id": "https://id.gs1.org/sscc/106141412345678908"
},
     "epcis:childEPCs": { "@id":
"https://id.gs1.org/8003/00952000000150000123" },
     "epcis:eventTime": {
      "@value": "2022-07-15T08:01:00.000000",
      "@type": "xsd:dateTimeStamp"
     },
     "epcis:recordTime": {
      "@value": "2022-07-15T08:01:00.000000",
      "@type": "xsd:dateTimeStamp"
     },
     "epcis:eventTimeZoneOffset": "+08:00"
    }
   ]
```


Figure 9: Object Event ADD

Figure 10 presents a sample event of completing a delivery trip. The childEPCs lists the returnable assets comprising the shipment and the parentID gives the SSCC of the logistic unit that defines the trip.

```
{
"@context": [
   "https://gs1.github.io/EPCIS/epcis-context.jsonld",
   { "gs1": "https://gs1.org/voc/" },
   { "cbv": "https://ns.gs1.org/cbv/" },
   { "ngs": "https://ngs-sensors.it/" }
  ],
  "type": "EPCISDocument",
  "schemaVersion": "2.0",
  "creationDate": "2022-07-29T14:00:00.000000",
  "epcisBody": {
   "eventList": [
    {
     "@id": "ni:///sha-256;04?ver=CBV2.0",
     "@type": "epcis:ObjectEvent",
     "epcis:action": "DELETE",
     "epcis:bizLocation": { "@id": "https://id.gs1.org/414/095200001237" },
     "epcis:readPoint": { "@id":
"https://id.gs1.org/414/095200001237/254/235" },
     "epcis:bizStep": { "@id": "cbv:BizStep-assembling" },
     "epcis:disposition": { "@id": "cbv:Disp-in_progress" },
     "epcis:parentID": { "@id":
"https://id.gs1.org/sscc/106141412345678908" },
     "epcis:childEPCs": { "@id":
"https://id.gs1.org/8003/00952000000150000123" },
     "epcis:eventTime": {
      "@value": "2022-07-29T12:00:00.000000",
      "@type": "xsd:dateTimeStamp"
     },
     "epcis:recordTime": {
      "@value": "2022-07-29T12:00:00.000000",
      "@type": "xsd:dateTimeStamp"
     },
     "epcis:eventTimeZoneOffset": "+02:00"
```


Figure 10: Object Event DELETE

Figure 11 presents a sample event of a senso reading including an alarm condition for anomalous value.

```
{
  "@context": [
    "https://gs1.github.io/EPCIS/epcis-context.jsonld",
    { "gs1": "https://gs1.org/voc/" },
    { "cbv": "https://ns.gs1.org/cbv/" },
    { "ngs": "https://ngs-sensors.it/" }
  ],
  "type": "EPCISDocument",
  "schemaVersion": "2.0",
  "creationDate": "2022-07-29T14:00:00.000000",
  "epcisBody": {
    "eventList": [
      {
        "@id": "ni:///sha-256;03?ver=CBV2.0",
        "@type": "epcis:ObjectEvent",
        "epcis:action": "OBSERVE",
        "epcis:bizStep": "cbv:BizStep-sensor_reporting",
        "epcis:eventTime": "2022-07-22T19:11:16+00:00",
        "epcis:eventTimeZoneOffset": "00:00",
        "epcis:readPoint": { "id": "geo:51.2069,71.2940" },
        "sensorElementList": [
          {
            "sensorMetadata": { "time": "2022-07-22T19:11:16", "deviceID":
"ngs:D5:D1:B8:17:F8:18" },
            "sensorReport": [
              {
                "type": "gs1:Temperature", "value": 18.31, "uom": "CEL"
              },
              {
                "type": "gs1:RelativeHumidity", "value": 51.2069, "uom": "P1",
                "exception": "gs1:ALARM_CONDITION"
              }
            ]
          }
```


Figure 11: Object Event OBSERVE