D1.1
Component and Experiment Design Report

ABSTRACT

This task will define the overall requirements and architecture of AffectUs, containing descriptions of the components that will be developed, their relation to the existing TagItSmart platform, the integration approach as well as interconnections and the purpose of their functionality. Interface specifications, communication patterns, expected information flows will be analyzed and described, along with the necessary highlight of the needed semantic structures. This design will take under consideration requirements deriving from end users, the TagItSmart current implementation and design as well as information from the TIS use case scenarios.

Delivery Date: 31.10.2017

Work Package: 1

Task: T1.1

Dissemination Level: Public

Type of Deliverable: Report

Lead partner: ELKE/HUA

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

CONTRIBUTORS

	ELKE/HUA
Authors	George Kousiouris, Stylianos Tsarsitalidis, Stavros Koloniaris, Evangelos Psomakelis, Konstantinos Tserpes, Dimosthenis Anagnostopoulos

DOCUMENT LOG

Issue	Date	Comment	Author/Partner
0.1	1/9/2017	Table of contents	George Kousiouris
0.2	10/9/2017	Incorporation of comments from Vaasa Workshop in ToC	George Kousiouris
0.3	18/09/2017	Chapters 2.3.1, 2.3.2., Use case diagrams	V. Psomakelis, G. Kousiouris, D. Anagnostopoulos, K. Tserpes
0.3.1	25/09/2017	Chapter 3.2	Stylianos Tsarsitalidis
0.3.2	26/09/2017	Merge of changes (all chapters) Subtle changes in Chapters 3.2, 3.3	Stylianos Tsarsitalidis
0.3.3	28/09/2017	Changes and merges (all chapters) Some more changes in Chapter 3.3, Chapter 4, inclusion of linked data cases	S. Tsarsitalidis, V. Psomakelis, S. Koloniaris
0.3.4	5/10/2017	Addition of sequence and component diagrams, events description in secton 2	G. Kousiouris, D. Anagnostopoulos, K. Tserpes
0.4	15/10/2017	Finalization of introduction, executive summary and conclusions	G. Kousiouris
0.5	20/10/2017	Finalization of OWL definition	Stylianos Tsarsitalidis
1.0	29/10/2017	Final version	G. Kousiouris, S.Tsarsitalidis

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

ACRONYMS

Acronym	Description
API	Application Programming Interface
HTTP	HyperText Transfer Protocol
KB	Knowledge Base
LCC	Large Crowd Concentration
NE	North East
OATH	Open Authentication
OWL	Ontology Web Language
RDF	Resource Description Framework
REST	Representational State Transfer
ST	Smart Tag
SW	South West
TIS	Tag It Smart
UC	Use Case
UI	User Interface
UML	Unified Modelling Language
URL	Uniform Resource Locator
VE	Virtual Entity

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

Contents

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

INDEX OF TABLES

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

INDEX OF FIGURES

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

Executive Summary

AffectUs is an extension module for the TIS platform, that aims to cover a set of functionalities aiming to

- Detect abnormal conditions with relation to the state of a supply chain, based on models created from historical data regarding transition times from stage to stage
- Link and forward externally identified generic events (coming from open data sources, smart city data and external developers) to affected entities of the chain
- Increase link between verticals through identifying and implementing links between generic events and effects on a given Thing/product

Through the joint usage of semantics, events identification and forwarding, AffectUs may enable this cross-fertilization of event notifications between verticals, thus enriching application context. Automated discovery of what products are affected by the predicted events is part of this process.

In this document, the main requirements, scenario needs and design considerations for the AffectUs system of services are highlighted. Initially in Section 2 an analysis is performed on the types of events that need to be described and detected through the system, based on the innovation goals and end user feedback received during the early stages of the project. This results in 6 requirements identified, that need to be addressed by AffectUs, and have their origin from the necessary multitude of sources (end users, innovation goals, technical requirements etc.) and can be used to drive system use cases and implementation. Furthermore, external data sources are analyzed, while requirements towards the TIS UCs are posed in order to identify the ideal candidates for the application of AffectUs.

One of the main aspects of the approach is a specialized semantic structure that is needed in order to cover dependencies between products, the concept of the supply chain and the description of externally available events, while giving the ability to cross-correlate between all these concepts. The concepts included in this ontology definition are presented in Section 3, along with an initial view on the ontology and the rationale behind its structure. The ontology has respective fields in order to model a supply chain, map its stages with specific tag ids, define dependencies between different tag ids and with external concepts (such as weather conditions) but also hold the concrete rules that dictate the event detection in the end. Furthermore, aspects of modelling that can be used to abstractively create prediction cases from the available historical data, post their correlation with the supply chain concepts, and inclusion of the identified rule limits as part of the ontology is included in Section 4.

Finally, in Section 5 the main system use cases that are created in order to satisfy the requirements of Section 2 are highlighted, while specifying also the number and type of roles (Product Owner, Product Consumer, Application Developer, Analytics Developer) that are expected to interact with the system. 6 such use cases are identified and analyzed, that aim at covering the intended functionality of the system as well as raise abstraction levels to the user, through extensive UI elements and reusable flows. Following, the system component diagram is presented, including the necessary sequences of operation

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project
co-funded by the European Union (EU) Horizon 2020 program.

(1 per system use case) for implementing the functionality dictated by the system use cases. In Section 6 we conclude the design specification with a mapping between the dictated requirements and the system use case and sequence that has undertaken its implementation, in an exercise to check coverage of the requirements by the envisioned system functionality.

1 Introduction, Objectives and Software design methodology

For the software design process we will follow the UML paradigm, in order to capture the main aspects of the extension module and its link to the TIS platform. In principle, we will use the following UML diagrams:

- Use case diagrams to indicate external actors and needed concrete functionalities from the extension. These functionalities are sourced on the initial innovation vision and the respective technical requirements, as well as requirements coming from the existing platform (in the form of needed integration) and open innovation principles (e.g. co-creation workshops to determine aspects such as needed events etc.)
- Component diagrams including TIS platform components in order to highlight integration points with the latter and necessary interfaces
- Sequence diagrams for the processes defined in the UC diagrams in order to detect call sequences and create the implementation guideline

The generic process appears in the following figure (Figure 3). Design considerations include the main innovations targeted by AffectUs:

- Increase link between verticals through identifying and implementing links between generic events and effects on a given Thing/product
- Create generic prediction mechanisms and enable predictive analytics on TagItSmart platform, used in an abstract manner by developers
- Through the joint usage of semantics, events identification and forwarding AffectUs may enable this cross-fertilization of event notifications between verticals, thus enriching application context. Automated discovery of what products are affected by the predicted events is part of this process.
- Enable entities such as external analytics developers that produce generic notifications (such as detection of various social conditions in a city, traffic conditions etc.) to disseminate their events in whatever interested entity such as a supply chain that is affected by the analysis outcomes.

Feedback received from the consortium, participation in external events and from the organization of the internal events is expected to be used in order to enhance the outcome of the extension, either as feedback received during the design time or for evaluation after the implementation end.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

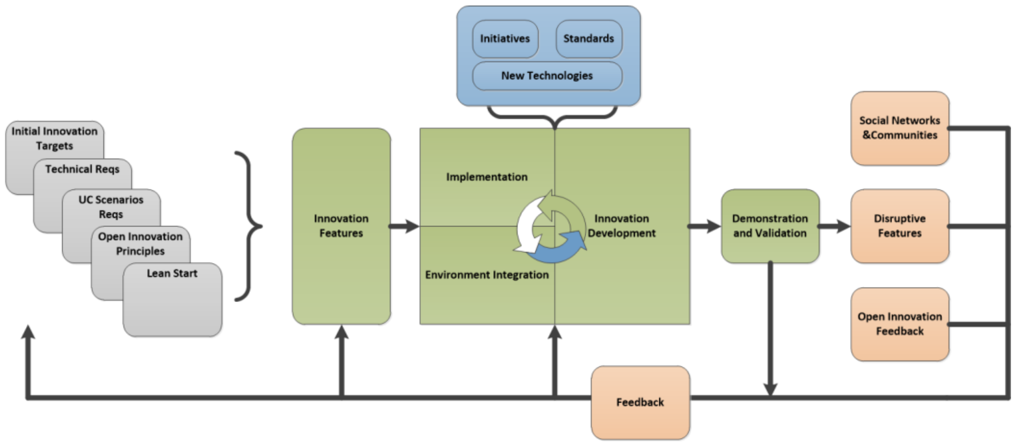


Figure 1: Generic Implementation Methodology of AffectUs

1.1 *Tasks breakdown*

In terms of tasks breakdown (in an Agile-like manner), we foresee the following main stages:

- Data Sources Definition
  - The available input data are a key consideration since they may define the types and richness of the identified events. In terms of data sources we have two major ones, the external TIS sources (any kind of open data available through the internet) and the TIS based ones, which refer to the included products. The latter include the generic information (such as scan time, location and product ID) but also the smart sensor tag values, that depend on each scenario and smart tag capability.
  - Data acquisition methods incorporation and storage consideration for the data retrieved from both sources (TIS products or external data)
- Needed events identification
  - ○ Definition of the events is critical since it dictates to a large extent the needed prediction methods as well as the needed data sources
- Identification of semantic extensions
  - The need from the main innovations (such as automatically detecting affected entities from an event) implies that the respective relationships will have been declared and populated in a semantic structure. Thus the definition of needed semantic placeholders and queries is important for enabling the functionalities of the prototype. This needs also to include TIS concepts and entities in order to automate the end to end functionality (such as detecting the dependency and pushing the event in the respective notification endpoint).
- Prediction framework design
  - The need to merge and call prediction methods from the main data sources is a feature that needs consideration, especially in the form of the necessary sequences that need to be in place in order to implement the end to end functionality. Developers need to be able to specify easily the selected model as well as the data inputs that need to be part of this model

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

1.2 *Structure of the document*

The document is structured as follows. In Section 2, aspects of the scenario are identified, such as requirements that stem from innovation goals, a candidate list of events to be identified based also on the feedback received during a co-creation workshop in the context of TIS, as well as a list of candidate data sources to be used. In Section 3 the needed semantic extensions and concepts are highlighted, in an effort to capture dependencies between products and/or external circumstances. In Section 4 aspects of the prediction framework are highlighted while in Section 5 the main design of the AffectUs module is presented, consisting of the roles, system use cases and sequences that are needed. Finally, Section 6 concludes the document.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

2 Scenario description

2.1 *Needed events definition through co-creation*

One of the key aspects of the extension is the ability to combine different data sources in a manner that will concentrate and generate more knowledge and proactive management on behalf of the stakeholders in the system. In order to strengthen these aspects, AffectUs participated in the co-creation workshop of TIS, through the communication with which a number of requirements have been produced. The most important of the latter is to be able to combine not only external data (e.g. weather information) or generic information coming from the tag analysis, such as location, time and tag id in a scan, but also the inner sensor values included in the tags themselves, in order to better illustrate the process of tag usage.

From this interaction as well as from subsequent communications, the following list of candidate events has been compiled:

- Events that have to do with delays in the transition between stages of the supply chain and can be detected through the exploitation of historical data (e.g. for the extraction of average times or other statistics characteristic of a transition). These events are agnostic to the internal sensor values of the tags, they mostly rely on typical location-timestamp-tagid values but have especially impact in the cases where timely delivery is essential. These events can be triggered if a respective scan in the target (next stage) location does not arrive within a certain timeframe.
- Abnormal events in the sense of an unfeasible or illegal sequence of appearance, for example:
    - scanned twice at selling point
    - scanned outside a designated region of sale
    - scanned at recycling point while not scanned at selling point
    - scanned at recycling point and then resold at selling point
    - scanned at selling point while scanned in the past and found violation in the temperature threshold. This event also captures the requirement for using the internal sensor values of the tags
- Batch recall. In one of the scenarios (champagne manufacturing) there is a documentation of the parts that the final product consists of. Therefore, one event coming in related to one of the parts that indicates defective batch, needs to be associated to the selling points of that batch in order to speed up the recall process.
- Another interesting feature (related to the first item in this list) is also the fact that personnel needs to be trained and monitored in order not to forget to scan products at designated locations. Lack of scanning may be attributed at some cases also to suspicious behavior (e.g. if at that specific stage storage requirements for sensitive goods were not met, the person responsible might avoid scanning in order to break the sequence of responsibility that can lead back to the specific point in the chain) or it may be due to simple negligence. Thus relevant notifications should be issued in order to remind about the need to scan at given points in time or location.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

- Special offers for products that are near expiration date but with not enough selling rate in order to reduce the stocked quantity
- With relation to the recycling case, one aspect is the prediction of the containers reaching their maximum capacity at a given recycling point. Especially if this is linked to quantities consumed through the previous parts of the chain (e.g. selling points) within the area, or with relation to external weather conditions (e.g. hot weather affecting the increase in ice-cream sales)

External data sources that may be used include:

- Social network data usage, in order to detect large crowd concentrations in a given area, that might affect a number of aspects such as:
  - Increased consumption of the given product
  - Increased delays in the transportation phase that include the locations around which the event has been detected
- Weather data that can affect:
  - Transport times
  - Production especially of agricultural products that may be used as raw material in the production of a TIS product

During the implementation stage of the project, at least two of the aforementioned events/categories will be incorporated in the demonstration of the AffectUs extension.

2.2 *Requirements drafting*

The following generic requirements have been identified either from the AffectUs goals or from the integration scope with TIS or from the end users (in our case the project UCs) and outcomes of Section 2.1.

Table 1: Requirement for external analytics events consideration

ID	Affect_REQ1
Definition	External events developers that offer generic notifications to potential interested entities (e.g. incoming bad weather, current bad traffic, large crowd concentration in parts of a city) need to be able to declare their offered events in a conceptual manner (e.g. through semantic concepts) so that these can be later on correlated by the platform with a specific subset of the products that might be affected by the events identification
Origin of requirement	From the envisioned business model (e.g. eventflows.com) and the respective innovation goal regarding mixing different types of information that is of interest, as well as the abstraction goal
Affected Entities	• Semantic structure needs to capture these generic events declaration • UI needs to be adapted so that developers can express what they offer in a graphical manner • Queries during runtime need to correlate a given product (and instance) with

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

	the offered events
Success Criteria	1 external event should be included in the demonstration, as well as its declaration through the structure and automated discovery of effect by AffectUs between the specific event and a given product.
Priority	SHOULD

Table 2: Requirement for shared IDs between TIS and AffectUs

ID	Affect_REQ2
Definition	Shared IDs rationale between TIS platform and AffectUs. Given that AffectUs needs to understand from the disseminated scan info the type of product class, we need to follow the rationale of TIS platform during ID assignment, which includes the product class ID and then a unique ID that dictates the specific instance of the given product class.
Origin of requirement	Technical integration need for identification of product class from baseline TIS data and implementation of the respective semantic queries following data acquisition
Affected Entities	Semantic Structure and relevant link flow (Naming convention for storing data in the KB)
Success Criteria	Effectively understand at the AffectUs side to which product class and instance the scan data refers to
Priority	MUST

Table 3: Requirement for abstracted supply chain analysis

ID	Affect_REQ3
Definition	Abstract and adaptable supply chain delays analysis functionalities should exist, in order to retrieve and extract statistics from a given supply chain instance, based on generic methods and data processing functions. For this reason it needs to correlate on the fly locations with supply chain stages in order to understand the positioning of each product at any given time with relation to the chain.
Origin of requirement	Innovation goal of affected delays and generic modelling template
Affected Entities	• Semantic concepts definition (needs to include dependencies description between products and chains and between locations and stages) • AffectUs UI (needs to incorporate relevant page for declaration by the involved entity) • Modelling method for analyzing delays

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

Success Criteria	Identify state of the chain with only the initial declaration of information from the Application Developer
Priority	SHOULD

Table 4: Requirement for UI based dependency insertion

ID	Affect_REQ4
Definition	UI insertion for dependencies needs to exist in order to hide declaration specifics as well as the need to understand the underlying semantic concepts and functionalities. Insertion should either be performed by the external analytics developer in order to declare what type and characteristics their event has, or by vertical owners/consumers, in order to describe inputs and outputs of their supply chain and relevant information (such as location of each stage if applicable).
Origin of requirement	Innovation goal of abstraction
Affected Entities	• AffectUs front end functionalities and UIs • Backend Node-RED flows in order to gather declared information and implement the respective semantic queries to store relationship data
Success Criteria	Feedback from external users during demonstration activities
Priority	SHOULD

Table 5: Requirement for packaging considerations

ID	Affect_REQ5
Definition	Designed services should be as finegrained as possible to enable the DevOps microservices approach to apply. This includes their joint configuration, parameterization and in most cases exposure of the respective service interfaces for the individual components to act as a service bundle.
Origin of requirement	Innovation goal of flexible management and dockerization
Affected Entities	All AffectUs components
Success Criteria	Deployment through dockerized services of AffectUs
Priority	COULD

Table 6: Requirement of addition of sensor values in event generation

ID	Affect_REQ6
Definition	Supply chain events should also take under consideration the smart tag “inner” sensor values in order to detect specific events of the given product class.
Origin of requirement	Co-creation workshop feedback
Affected Entities	• AffectUs front end functionalities for the product owner to declare rule on the limit value for a given sensor value • Semantic structure to hold and indicate the rule based on the declared limit value
Success Criteria	Participant should be able to declare and force checks based on the given rule to generate events based on the rule
Priority	SHOULD

2.3 Data sources review and selection

2.3.1 TIS data sources review and selection

With relation to the requirements needed to exist in order for a TIS Use Case to be the target of AffectUs, the following points of attention need to be mentioned:

- The selected data source/use case needs to be incorporated in the EVRYTHNG platform, given that this is the main integration point between TIS and AffectUs. Digital beer and recycling (with icecream products that are also temperature sensitive) are the main candidates from this point of view and based on project discussions. These can also be linked with external events such as LCC that may affect consumption of these products.
- Data need to be available in the next months and definitely within the duration of the AffectUs extension implementation, thus until February 2018.
- We assume that each physical/conceptual object is associated with a unique tag type id (besides the tag ID differentiating the tag from other similar tags), so that we can indicate abstract dependencies needed for the production of one product from basic elements

2.3.2 Identification of candidate external data sources

External data sources can provide us with additional information, enabling us to identify or even predict external events that can affect the tracked products in various ways, either positively or negatively. They also can provide us with enhanced contextual information, leading to more accurate identification of the events that affect each product. For example we could use the external input that a large crowd of people are marching on a street to notify the driver of a truck that if he chooses to follow this path he will be delayed, possibly damaging the products he is delivering. For a more clear view of the identified external data sources we have split them into the following categories.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

2.3.2.1 Weather Data

Weather greatly affects a large variety of products, in direct or indirect ways. For example if we are tracking an ice-cream product and we are notified that a heat wave is coming, we can notify all members of the production and transfer line to enhance the functionality of their fridges in order to avoid damage to the product. At the same time, we can notify the shop owners that an increase in ice-cream demand is to be expected due to the heat wave.

As data sources for Weather data, we can depend on a number of free to use weather web sites that provide their predictions through APIs. Such websites include, but they are not limited to, "darksky.net", "openweathermap.org", "www.wunderground.com". All these data sources are available to the public without charge, through the provided APIs and each one has strengths and weaknesses as seen in the following table. The final choice of which one to use depends on the use case we are working on each time.

Source	Endpoints	Input Parameters	Historical Conditions	Current Conditions	Future Conditions	Maximum Prediction Time
openweathermap	Nodered + RestFul	City, Country or Coordinates	Not Provided	• Description • Weather • Tempc • Tempk • Humidity • Windspeed • Winddirection • location	• epoch timestamp • pressure • humidity • wind speed • wind direction • cloudiness • temp	5 days
wunderground	Nodered + RestFul	City, Country or Coordinates	Not Provided	• Description • Weather • Tempc • Tempf • Tempk • Humidity • Windspeed • Winddirection • Location • Forecast • Epoch	Same as current	12 hours
darksky	Nodered + RestFul	Coordinates, time, metrics	Same as current	• Weather • Detail • Humidity • Maxtemp • Mintemp • Wind speed • Wind direction • Longitude • Latitude • Clouds • Precipitation • Sunrise	Same as current	Unspecified

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

				• Sunset • Units		

Table 7: Weather Data Sources

2.3.2.2 Social Data

The social networks are always providing us with user created data that can help us identify or predict various events related to the tracked products either directly or indirectly. For example, we could estimate the consumption of a product per location for a duration of time by looking at how many people are talking about it. We could also identify vulnerabilities to a product by performing sentiment analysis on the public posts that mention this product, notifying the corresponding part of the production or transfer line that this problem is becoming noticeable by the consumers.

As social data sources we can use the Twitter which provides an easy to use API and most of the post made on it are public so we can access them. Other possible data sources include Facebook, LinkedIn. If these data are not enough for our purposes we can extend our search in less public media, such as various fora that are relevant to each specific use case. By looking at the following comparison, Twitter seems the easier to use and the most effective means of gathering textual data from the public.

Source	Endpoint	Input Parameters	Results
Twitter	Nodered + RestFul	Hashtags, Users or Keywords	List of tweets
Facebook	RestFul	Hashtags, Users or Keywords	List of public posts
LinkedIn	RestFul	Users or Keywords	List of public posts

Table 8: Social Data Sources

In addition to the social data collected by the mentioned sources, we will be using other sources for sentiment analysis tasks. The sentiment analysis will receive the textual data that we will be gathering from the social networks and analyse them in order to extract information about the opinions of their authors. For comparison we have listed three of the most known sentiment analysis APIs that provide their analysis free of charge through restful APIs. Google seems to be the obvious choice here but the other two APIs provide a more basic functionality, which makes them faster and easier to use. The final choice will depend on each specific use case and its needs.

Source	Endpoint	Input Parameters	Supported Languages
sentiment140	RestFul	Text, target, id, language	• English • Spanish
Google Cloud	RestFul	Text, Encoding	• Chinese (Simplified) • Chinese (Traditional) • English • French • German • Italian • Japanese • Korean

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

			• Portuguese • Spanish
getsentiment	RestFul	Text	• English

Table 9: Sentiment Analysis APIs

2.3.2.3 Large Crowd Concentrations

The third category of external data source relates to the concentration of crowds in specific places. These crowds can gather for any reason in any place at any time, affecting different parts of a product's life cycle. For example, a large crowd gathered in a square to attend a concert can mean an increase in beer demand for that area's shops, so we should notify the shopkeepers to restock and fill their fridges.

An indicative example of large crowds concentration comes from the outcomes of the Cosmos European project, which has produced a set of Node-RED flows for detection of these events in real time, using Twitter, based on a typical lambda architecture setup[15]. In this case, the number of Tweets in a city are stored with geolocation data. Upon request, a specific location of interest is selected, statistics with relation to the "normal" number of Tweets are processed and in real time the current number of tweets are maintained. In case they violate the normal limit, the relevant event is triggered.

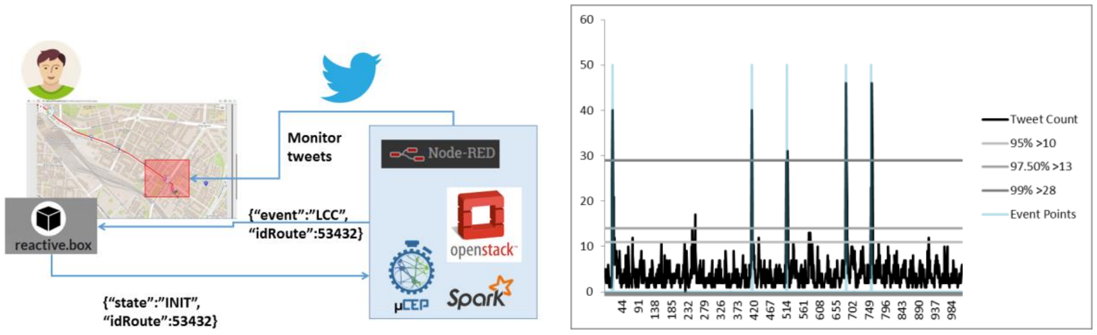


Figure 2: COSMOS Large Crowd Concentration Event[15]

2.3.2.4 Traffic

Traffic data are important and can affect the products in a number of ways. For example, if we could identify the traffic in each road leading from the factory to the shop, we could estimate an optimal route for the truck and notify the driver about this path. Then we could estimate the time of arrival for the path chosen by the driver and set the temperature of the fridge, if applicable to the specific use case, in a way that preservers the products and wastes as little energy as possible. After that, we could notify the shopkeeper about the estimated time of arrival in order for her to prepare the storage space and arrange her work to create an empty timeslot during the expected time of arrival.

As a traffic data source we can use the "Here Traffic API" which provides information about traffic flow and incidents in various countries or the Google Services which provides route information taking into account the traffic patterns that it has distinguished. Both of them are free to use and provide APIs that enable us to get the data programmatically but Google also has a NodeRed client ready to use and suits our needs of estimating delays and time of arrival so it is the most relevant choice.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

3 Identification of semantic extensions

3.1 Definition of needed semantic queries from main innovation principles

Based on the described vision, requirements and innovation targets, the following cases of queries can be identified, which are a direct input to what kind of semantics are needed (Section 3.2):

- Ability to document (and query afterwards) the link between affected categories of entities and events (at the class level), but also at the value level. As an example, a product may be affected by a weather condition, but only in a given location of interest. Thus a characterization of the event based on a data range would be needed.
- A documentation and link between which data sources, in the form of "which model id uses which data source", in order to automate the process of event detection.

  e.g. [Data source] Included in [model id]
- In order to create links between semantic stages, a potential description (even high level) of a production line, and/or a supply chain, would be necessary. This would also imply associating physical locations with production stages, so that the link to the semantic of the stage is done programmatically and at the data level.
  - Domain model of states/stages in a production line ordered->delivered->produced->shipped->delivered at destination
  - Instances of locations mapped to a stage definition, so that when a scan with the details of {location,tag_id,timestamp} arrives, it is possible to correlate which tags, to which amount and how long it takes to go through the stages.
- A description of the event concept, with needed characteristics such as location and time, as well as generic links to what entities it affects. The instantiation of this information is expected to be performed by the user (potentially through a guided UI). This aspect will enable the programmable link between the event detection and the notification process.

The above basic concepts are the basis upon which the ontology will be built. This ontology describes the connections between a causality (e.g. weather, prolonged time) and possible effects (e.g. product quality in danger). To make these connections precise and usable, events and product types have to be interconnected, so that the queries return useful and decent results, by making inferences.

3.2 Usage and Domain of Ontology

The information that flows from various sources, real-time or not, is semi-continuous. Information about each physical {location,tag_id,timestamp} "instance of a" product, as well as external affecting data, have to be separate from the logic encoded in the ontology.

The ontology consists of the ground truths about event and product (metadata), along with the additional information provided by the user. The UI has to intuitively allow a user to add specific information about events and types of products that are of interest, so as to extend the information that can be queried. One possibility is to have an ever-expanding web of information, which may extend well beyond the boundaries of the initial ideas and be tailored to the users' needs. Reasoning may also play a

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

role here, to validate the users' "claims", and possibly find additional information. Then, by using semantic queries, complex logic is simplified. Inferences that come from semantic queries provide knowledge about the types of products that are affected, and how they are affected.

Using the aforementioned semantic queries, certain rules or conditions are inferred. With these rules, the spectrum of products affected is narrowed down to the product (class). Additionally, "conditions" that need to be satisfied at the level of a physical item (unique tag_id), so that the right parties can be alerted or notified about the potential danger, change or happening (event), are available. The conditions are programmatically applied to each arrival of an event.

For example, the weather is not the same in every location at every time. An event about weather changing into a potentially harmful state for certain types of product is flowing throughout the platform. Through the steps described above, only the interested/affected parties are notified about it.

We can also use semantic queries to simplify the very instantiation of an event. For example, change in the state of the weather may, or may not be of interest. This is also true about social data, as it may not be known if an analysis has results that are of interest to all parties. To make sure that something may truly affect an entity (product, person etc.) we can use inferencing on semantics in a similar fashion as above, to instantiate an event, which will in turn have a better target spectrum. The event declaration can be simplified in a similar fashion, as the analytics developer may declare the potential states of the event, and the product owner may dictate only some, or even one, specific event type to be of interest to her.

So, to generalize, we can have a stream of data moving through a workflow (Node-Red), in which queries (SPARQL) are made to hit the triple store (JENA) that holds the ontology. The triple store can be enhanced with additional reasoning capability and linkage with data from other sources throughout the web. The inferences returned are used for event processing and directing notifications to the right parties. This approach provides the intelligence needed to achieve the described innovation targets. The domain of the ontology has to be the logic to enable the combination of data from all the aforementioned sources effectively, and produce conclusions.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

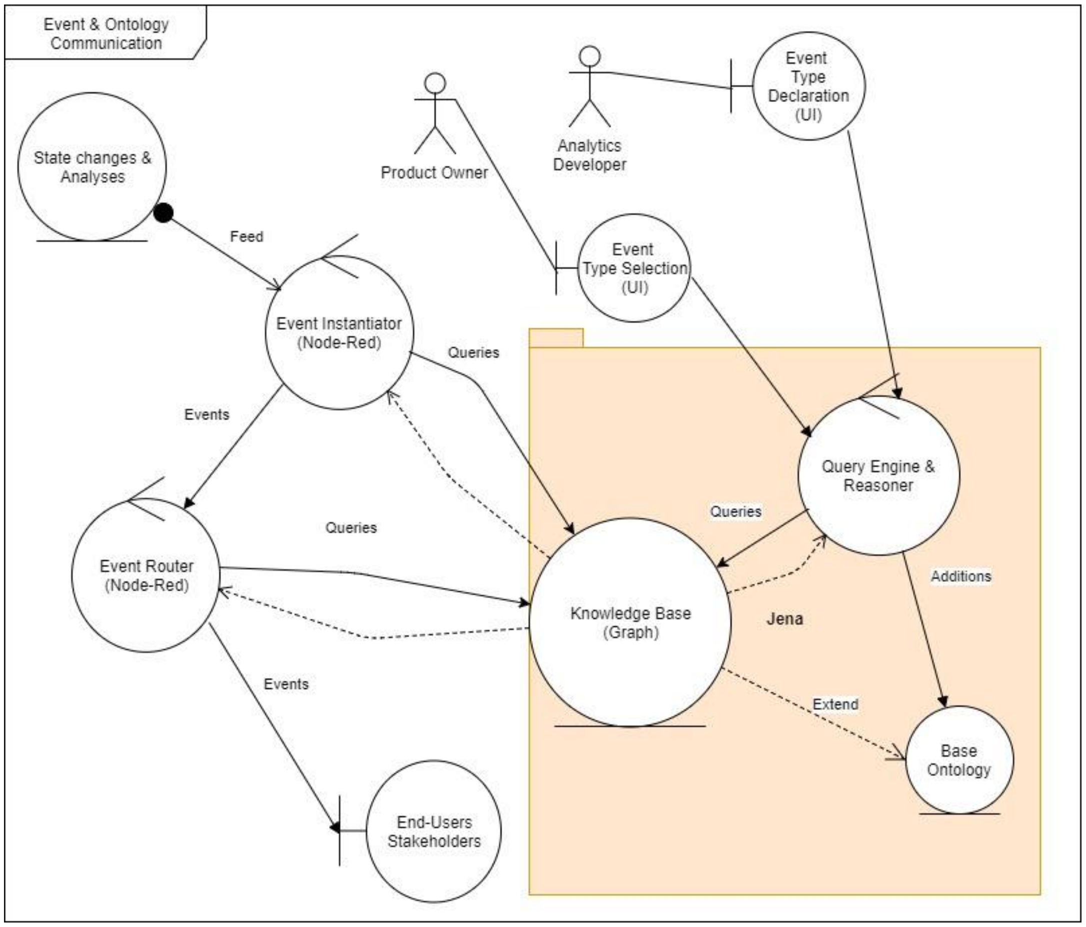


Figure 3: General Use of Ontology and Knowledge Base

3.3 Semantic concepts needed

When mentioning the term “Event” we mean the occurrence of a case that satisfies a condition (that may trigger some other action), an outcome, an issue, a result. Event types are interconnected, as well as connected with other concepts, through predicates, which are a high-level representation of the “state changes” that need to take place, for entities to be affected in some way. Those entities may be or consist of either products, producers or customers/consumers. The term Event takes many different interpretations, throughout the semantic web. In our case, the most basic definition is adapted to the needs of a complex event and entity framework. Resources (Subjects or Objects) that have the term “Instance” in them represent the physical product/item, state etc. at a specific place and/or time, and not the concept, product or state type etc.

In our case, events can be of two types. Either generic chain events with primarily VEs as subjects and particularly referring to the case of the supply chain stages concepts described in the following table, and analytics events with the majority of characteristics referring to eventCondition concepts. In the first case of supply chain events, identified events can be more generic such as ARRIVAL_OUT_OF_BOUNDS, MISSED_STAGE, WRONG SEQUENCE etc. (based also on the list of identified cases of section 2.1).

As such, an event should be associated with the following concepts:

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

Subject	Predicate	Object	Linked Data
EventConditionInstance	rdfs:subclassOf	EventCondition	EventCondition represents a concept such as e.g. humidity. EventConditionInstance is a specific instantiation of the humidity rule
EventConditionInstance	happenedAt	Time concept	Reuse from existing time concept
EventConditionInstance	hasLocation	Location / Space concept	Reuse existing location/space concepts
EventCondition	hasExternalInputs	DataSource External data sources feed (e.g. weather sites, social networks etc.) in a form of endpoint or id that represents a readymade nodered flow for that endpoint (parameters such as location for weather data should be dynamic)	Reuse from existing concepts, especially the TIS ontology coming from the COSMOS project regarding endpoints.
VE (or EVRYTHNG product)	isAffectedBy	EventCondition or other VE, in this case VE id should be obtained given that it is for the specific batch of VEs that might get delayed.	A VE is in general affected by humidity. Inference should be made from the EventConditionInstances that have a similar location and time parameters with the product instance
EventCondition	predictedBy	Model id	Necessary for referencing the model, if approach of rule is selected then potentially not needed
EventConditionInstance	hasMinValue	Numeric	This may be populated in a rule rationale by the modelling methods and/or the analytics developer

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

			declaration) or it can be used in the rationale of the product not having a value over/under a limit.
EventConditionInstance	hasMaxValue	Numeric	Same as above
EventConditionInstance	refersToSmartTagSensor	Sensor field	Optional, is needed for referencing which sensor field should have the limit mentioned above
EventConditionInstance	hasLimitValue	Numeric	In case of a value discovered through e.g. clustering
EventCondition	isAscending	Boolean	Implies that higher is better
EventCondition	publishedAt	Endpoint concept, where one could obtain the information on the appearance of such an event	Reuse from TIS or other, we assume the same endpoint for all the same conditions class given that they would in principle be created by the same analytics developer.
Smart tag id	ConstructedBy	Smart tag id(s)	Reuse from TIS concepts. May be obsolete if based on VE declaration and inference of smart tags from VEs
VE	isAssociatedWith	Smart tag id part or EVRYTHNG product ID	Assume that there is such a VE to ST relationship concept
Producer	Produces	VE	Necessary to abstract at the VE level. Inference can be made as to whether a given smart tag refers to the VE class, therefore associate smart tags with producers
Consumer	Consumes	VE	Same as above
Producer/Consumer	isPartOf	SupplyChainEntity	SupplyChainEntity enumeration? (roles or participants)

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

Chainstage	belongsTo	SupplyChainInstance	SupplyChain entity instantiated for this chainstage
SupplyChainInstance	rdfs:subclassOf	SupplyChain	A differentiation between general concept and specific entity, added by user
ChainStage	isOfType	StageEnumeration	Examples: Raw Material, Manufacturing, retail, end user
Chainstage	hasLocation	Location concept	Optional if applicable, location object would be in the form of lat,lon coordinates or geobox (lat,lon of NE and SW corners). All stages but the end users one should have one
Chainstage	hasPrevious	Chainstage	Necessary to indicate the proper sequence of stages (dictated also by the need for out of sequence scans mentioned in Section 2.1
Chainstage	hasNext	Chainstage	Same as above
Chainstage	hasAverageDurationTime (or any other type of statistic e.g. hasStdDev)	Interval concept	This is populated from queries to the historical data and then the results stored in the KB for this supply chain and chainstage During runtime upon rule launch, we infer the limit and this is put on the runtime rule
Chainstage	hasInputs	VEs or smart tags from which VEs are inferred	More than one instances of this relation can exist in case of N inputs
Chainstage	hasOutputs	VEs or smart tags from which VEs are inferred	More than one instances of this relation can exist in

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

			case of M outputs
Product ID	isOrderedBy	Consumer	This is necessary in order to have information about which batch is ordered by which entity
Product class	isManufacturedBy	Producer	In this case, we are only interested in the product class, given that for a given product the same producer will exist

Table 10: Identified Semantic Concepts

The core ontology concepts form a graph. For convenience, Vowl is used for its visualization.

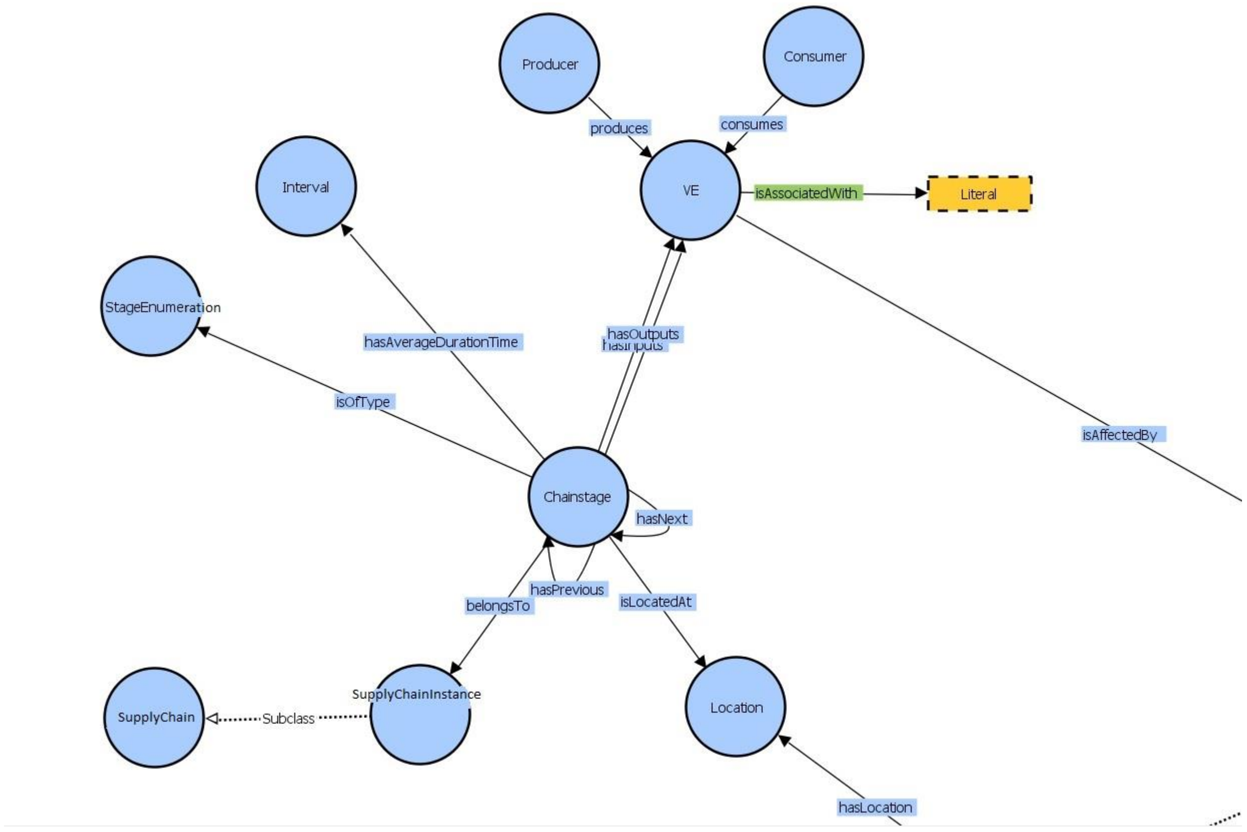


Figure 4: Core Ontology (Part 1 of 2)

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

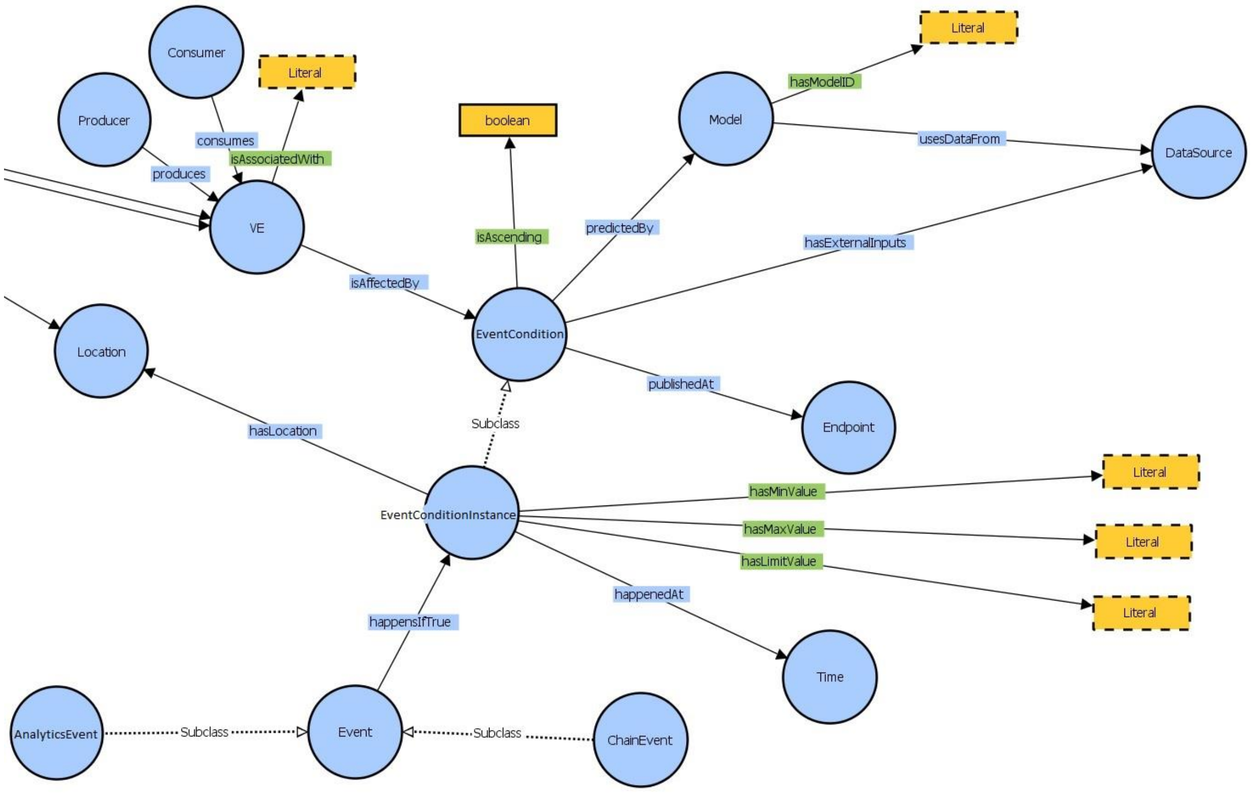


Figure 5: Core Ontology (Part 2 of 2)

3.4 Investigation of external Linked Data sources

Having described the needed concepts, we investigate in this section potential external ontologies and related entities that can be directly referenced, in the Linked Data Paradigm, in order to simplify the semantic structure and enhance reusability

Table 11: Event related concepts in existing vocabularies

Name:	EventCondition
Scope:	An Event concept is necessary in order to hold information about the nature of the occurrence, including aspects of dependencies between input and output tags, location and time occurrence and retrieval information
Existing work	[eventOntology-escience2008] argues on creating event based ontologies for describing flooding effects. Too specific on the flooding case [http://dbpedia.org/ontology/Event] has a number of nice features (such as CausedBy, Damage, timing aspects), however it has also a number of other not needed features targeting at describing human-defined events (such as gatherings, concerts etc) [http://purl.org/NET/c4dm/event.owl#] is a much simpler case, that while again focusing on e.g. concerts, it has much wider concepts such

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

	as location[http://www.w3.org/2003/01/geo/wgs84_pos#] , time[http://www.w3.org/2006/time#] , active agents, factors (that can be considered as the key inputs in our case) and products (potentially the affected outputs in our case) ,as well as an iterative subevent that may be very useful also for defining positive and negative cases of events (and according value ranges that dictate them). Time and location are candidates for reutilization even if the other concepts are not finally chosen. This vocabulary is also heavily utilized by 36 other vocabularies [http://linkedevents.org/ontology/] defines an Event class and 7 properties that may be helpful, including aspects of place and time, as well as involved entities An interesting addition is [http://semanticweb.cs.vu.nl/2009/11/sem/], which also extends from the basic event vocabulary, but has nice generic formalizations that are more descriptive, such as eventType, ActorType, abstract constraints etc., while including a variety of time intervals.

Table 12: Relevant Supply Chain or Entity dependencies

Name:	Supply Chain concepts and dependencies
Scope:	Description of relationships between entities is a key part of the ontology, in order to enhance the description and identification of dependencies
Existing work	With relation to description of relationships between entities, one candidate is The Prov ontology [https://www.w3.org/TR/2013/NOTE-prov-overview-20130430/]. It contains features such as ActivityInfluence that can be used to indicate dependency. It also has time features (such as generatedAt or endedAt). Another candidate is the PRIMER organization ontology [https://www.w3.org/TR/vocab-org/#bib-OWL2-PRIMER]. This case however is primarily intended for describing affiliated entities such as organizations, even though there are location concepts (such as Site) in order to include position of the organizations and linkedTo properties to indicate connection between organizations. Finally, GoodRelations [http://www.heppnetz.de/ontologies/goodrelations/v1.html] contains concepts that can describe a product or contents of a store, thus it is more focused on the retail part of the description. Some aspects on which it is based (like the "http://schema.org/latitude or longitude concepts) can be easily reused in our case .

Table 13: Weather related concepts in existing vocabularies

Name:	Weather

Scope:	Description of weather conditions, or weather metrics etc. will enhance the specific semantics capacity of the ontology, as well as ability to perform more abstract queries and the ability of the product stakeholder to declare specific characteristics that influence their product (e.g. humidity etc.) enabling more complex reasoning.
Existing work	One candidate is the Phenonet Sensor Network ontology [https://www.w3.org/2005/Incubator/ssn/ssnx/meteo/phenonet] that has some interesting features in terms of temperature, wind (and their units) and/or potentially the structure of the sensors. However it is missing other aspects such as humidity or rain, that can be found in the Ontology for Meteorological sensors [https://www.w3.org/2005/Incubator/ssn/ssnx/meteo/aws]

Table 14: Location related descriptions in existing vocabularies

Name:	Location concept
Scope:	The location concept is critical for a number of purposes, the most predominant of which is to correlate scanned tag info (that includes location) with the positioning in the supply chain (relevant stage) or the identification of which external events are relevant (in terms of affecting geoproximity) to the specific product ID.
Existing work	One of the most basic vocabularies for declaring simplistic (in the form of lat, lon coordinates) location features is included in WGS84 Geo Positioning [https://www.w3.org/2003/01/geo/] which is definitely one of the candidate cases. A more sophisticated approach is included in GeoNames [http://www.geonames.org/ontology/documentation.html], containing over 11 million geonames toponyms now have a unique URL with a corresponding RDF web service. Other services describe the relation between toponyms. While very detailed, one of the drawbacks in this case is the coarse grained nature of places. In our case, we may be interested in much more finegrained aspects, similar for example to the geobox concept in Twitter, in which a bounding box with coordinates in the SW and NE corner are given in a simple lat,lon format. Finally the LinkedGeoData project [Description: http://linkedgeodata.org/About] is an LOD initiative that exposes over 1.2 billion locations, including aspects such as how many places of interest exist, e.g. restaurants, service points etc in a location. This could be potentially very useful in case we consider dissemination of the respective identified event information in an ad-hoc manner to all affected entities in a region. Thus the relevant service places may be retrieved inside the given region.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

4 Prediction framework definition

4.1 Review of baseline methods for event identification

Our work in AffectUs is based on the identification of events both in the present and in the future. The main principle of these two aspects is the same. We have to create models that recognize the normal behavior of a production line in order to locate abnormalities in that procedure. The models created can identify these abnormities both in the present time and in the future because they "know" how the production line works under normal conditions, what affects these conditions and how a change in these conditions is causing (or will cause) abnormities.

The first step in creating these models is acquiring the dataset. The dataset is the basis of this whole process because all models are built from it and go back to it in order to compare the current state of the world with the state they already have learned. These data of course, rarely are in a form digestible by the algorithms used to create the models. A middle step is needed, which is called data pre-processing in the literature [1]–[6]. This step is trying to prepare the data by performing cleansing and feature extraction tasks, or in our case enriching also and transcending from the simple location characteristic to the chainstage abstraction.

By cleansing we refer to the process of removing duplicate and erroneous entries, removing irrelevant data that could confuse the models and possibly formatting the data into a more friendly format for the algorithms (html to csv for example). Feature extraction on the other hand is referring to the process of extraction second level data from the basic data contained in the dataset. For example, in a dataset containing Tweets, a second level datum may contain the average length of the words, the number of words and the average difference to another Tweet, which are all numeric entries that can be used by mathematical functions in order to create a model.

After pre-processing the dataset, we are using various machine learning algorithms (also called classifiers or clusterers) and categorization tools. Some of the most common ones include: Support Vector Machines (SVMs), Naïve Bayesian Networks, C4.5, Functional Trees, Best-First Trees, Multilayer Perceptrons and Logistic Regression algorithms which are used in isolation and in combinational experiments.

In detail, Naïve Bayesian Networks, Functional Trees, Best-First Trees and the C4.5 algorithms try to create categorization trees by iteratively splitting the possible values of each tested variable. Their differences are in the way they are deciding when and how to split a decision point, thus creating another branch of the tree. For example C4.5, which is based on the ID3 algorithm, is using the Information Gain theory, which in turn is based on the Entropy theory, in order to perform the less possible splits, creating the smallest possible tree [7]. On the other hand, Naïve Bayesian Networks do not care about the size of the produced tree so they are using a probabilistic model in order to maximize their accuracy [8]. Finally, Best-First Trees are trying to decide the "best" point to split the tree by using a more arbitrary function, specified in each case by the user in order to minimize the impurity between each new split [9].

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

Moving on, both Logistic Regression and Support Vector Machines are trying to create a mathematical function that can mimic the correlation between the features and the target class. The Logistic Regression is using various logistic functions in order to calculate a function with a graphical representation that has the minimum distance from each of the training data points in the multidimensional space, as defined by the features provided [10]. On the other hand, Support Vector Machines are using different mathematical functions, such as the minimum square function, in order to create new functions with similar graphical representation to the one created by the training dataset [11].

Finally, the Multilayer Perceptrons are a basic build of the artificial neural networks. They consist of a number of nodes, grouped in different fully interconnected layers [12]. Each node is using a pre-defined function in order to map its input values into a set of output values. This way each feature will be processed by one or more of node trees, containing one node per layer. The outputs of the first layer will become the inputs of the second and so on until at least one node of each layer has been activated for each feature. That way each one of the variables can contribute into the final classification decision with an intelligently calculated weight. A more recent advancement in this field is presented by the Deep Learning algorithms [13], that are using a more thick and rigid configuration of the neural networks, demanding more resources but becoming more powerful in pattern recognition and clustering tasks.

4.2 Adaptation of input/output

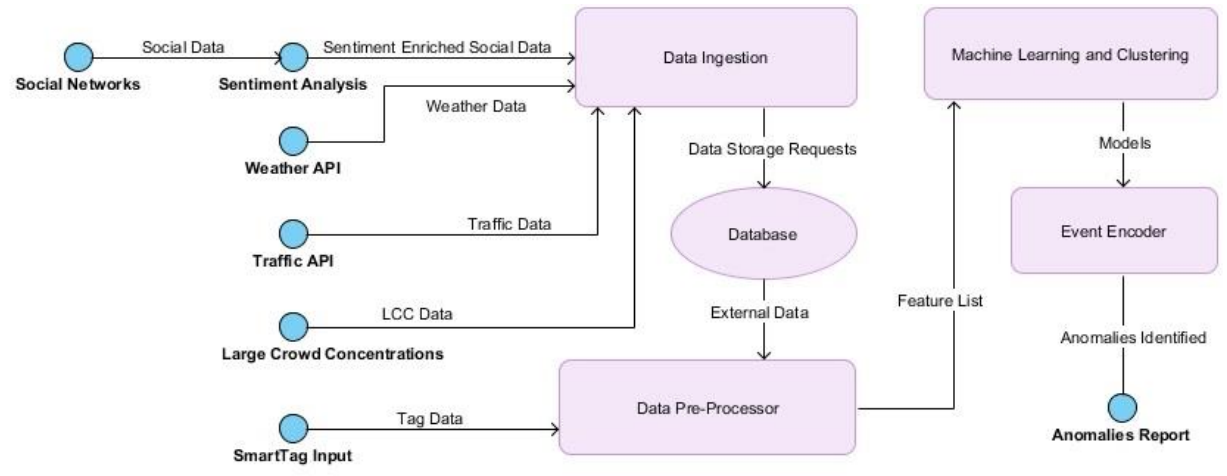


Figure 6: Modeling Process General Architecture

4.3 Generic Supply Chain Modelling

In case one needs to build a generic process of a supply chain (Figure 7), the different stages need to be modelled as well as the respective tags that have to do with each part of the chain. Then based on the scanned info, specific statistics per stage can be extracted, such as duration in a stage. Furthermore, based on available scanning of tags per layer, this chain may be simplified (e.g. remove Produced and Packaged) while based on historical data, measured times for transition between stages may be acquired, thus being able to detect an event of Delayed /Normal/AheadOfTime status. Finally, specific

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

aspects of correlation investigation between external events and observed delays (e.g. external traffic status and transition delay between Shipped and Arrived) could be considered in order to quantify an effect.

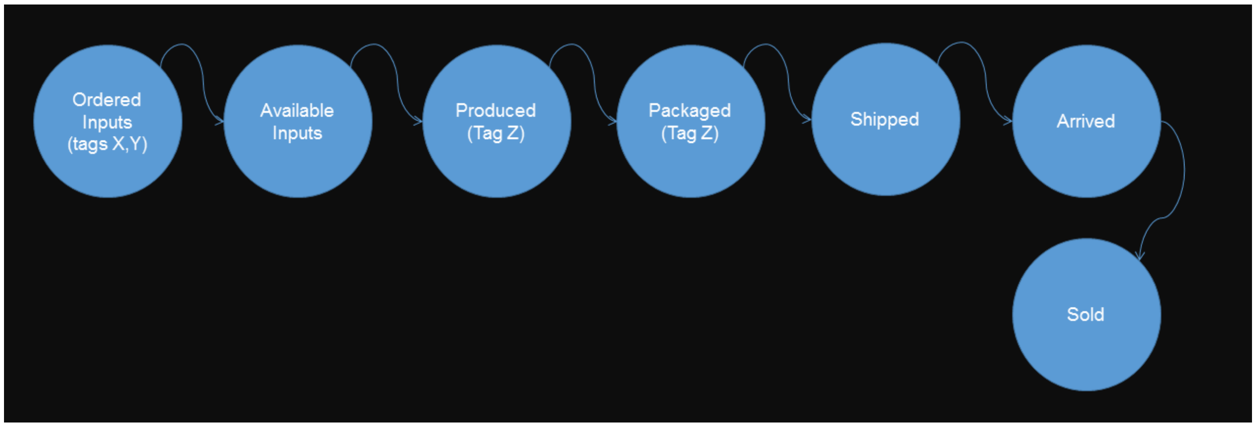


Figure 7: Example of supply chain

4.4 Time Series Modelling

Another potential candidate for incorporation may include the rationale of time series modelling[16], in order to predict trends in a given metric, based on previous values of the same or different metrics. This case could prove useful especially for events like recycle bin expectation time to be filled, as mentioned in the events described in section 2.1. Even if the used algorithm is based on simple aspects such as trend analysis or periodicity analysis (through e.g. a Fourier series decomposition), it may prove useful for the given use case and/or act as a placeholder for more sophisticated methods in the future.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

5 AffectUs Extension Design

Based on the aforementioned analysis of requirements, types of events and flow of information, sketching of AffectUs extension components can be performed. Initially we formally declare the anticipated roles in the AffectUs ecosystem, followed by the system use cases for the extension and the design of the block diagrams and functional sequences. The latter are also mapped to the initially defined requirements of Section 2.2 in order to check their coverage.

5.1 *Formal Role Definition and Objectives*

The main roles that are envisioned in the AffectUs ecosystem are illustrated below.

Product Stakeholder: this role is immediately affected by any event that occurs and affects a given product A. The role may be further specialized to **Product Producer** (EVRYTHNG Account/Product Owner) or **Product Consumer** (i.e. can be any end user or another Product Producer that expects product A as part of their supply chain in order to create in turn their owned product B). Delays in product A are assumed to affect both the Product producer and the product consumer. A product can be affected by other products or by external data sources (e.g. weather). A Product stakeholder can be affected by one or more other products. On detecting an event, and based on the aforementioned link, one can trace from the event the affected products and from these the affected product stakeholders. Product Stakeholders are foreseen to be logistics officers or business managers. Therefore a generic sequence could be the following:

On Event-> Retrieve which products are affected-> retrieve which stakeholders are affected-> Notify stakeholders

AffectUs Application Developer: acts as a broker/extension to the EVRYTHNG platform and has the role of supporting the aforementioned stakeholder roles in modelling and providing notifications based on the identified events. The Application Developer should have a clear UI to perform the customization of the flows needed for each case, as well as relevant abstractions that can help them in speeding up the process.

Analytics Developer: this entity is an external, independent developer that may utilize externally available data (such as weather, social media etc. sources) and produce generic events around abnormal situations with relation to these data. The related events can also be of help to the Application Developer, in order to enrich the level of notifications or detect dependencies between their managed products and this data source (e.g. traffic conditions in a specific area affecting the transportation time between two stages of a chain).

AffectUs system: is the set of components needed to support the AffectUs developers in their endeavours. This system should interact primarily with the EVRYTHNG platform of TIS and coordinate actions in order to achieve the envisioned functionality as well as provide the necessary abstractions and customizable flows to enable the process.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

The aforementioned roles are depicted in Figure 8.

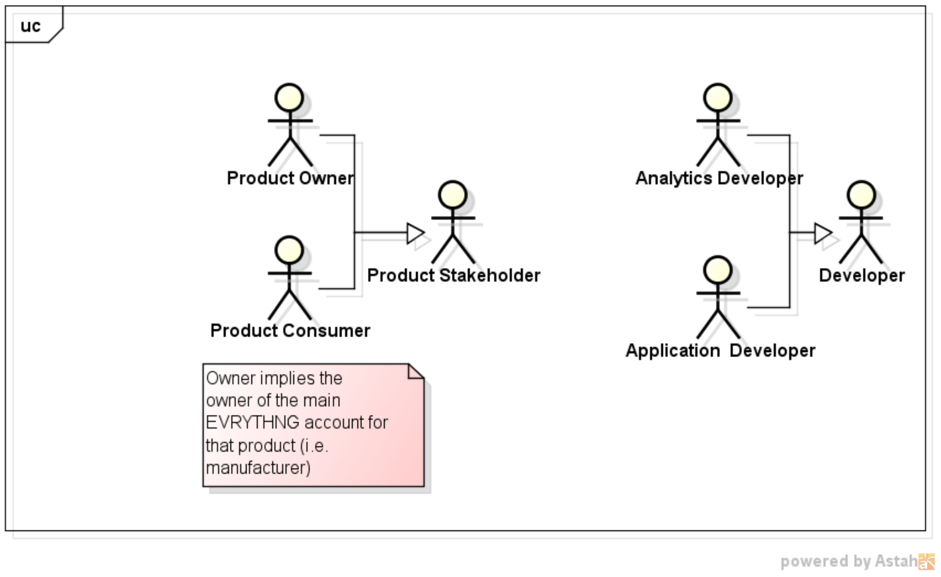


Figure 8: Roles summary and relationships

Through the envisioned extension, the following objectives are targeted:

- Enable easier association between concepts in the ecosystem, through defining and populating semantic structures that can enhance reasoning and inference
- Create coordination frameworks that can aid in merging and synchronizing the necessary functionality, while embedding modelling techniques that may be used in event detection from a multitude of data sources
- Combine the two previous cases in order to properly link the roles that need to be notified, as well as implement the respective bridging framework for this purpose

5.2 Link with the existing TIS platform

The overall blocks of the AffectUs extension in relation to the main TIS platform appear in the following figure, colour coded to indicate their relation to the existing TIS software stack .

The Workflow layer is implemented via Node-RED and represents the coordinating flows that integrate the two systems and the overall sequences (described in detail in Section 5.5). The association logic includes the semantic definitions and Knowledge base, and is populated either through the UI based inputs of the respective role, or through integrated flows that capture raw data from the TIS messaging services, map them to the semantic concepts and transform them into KB instance data.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

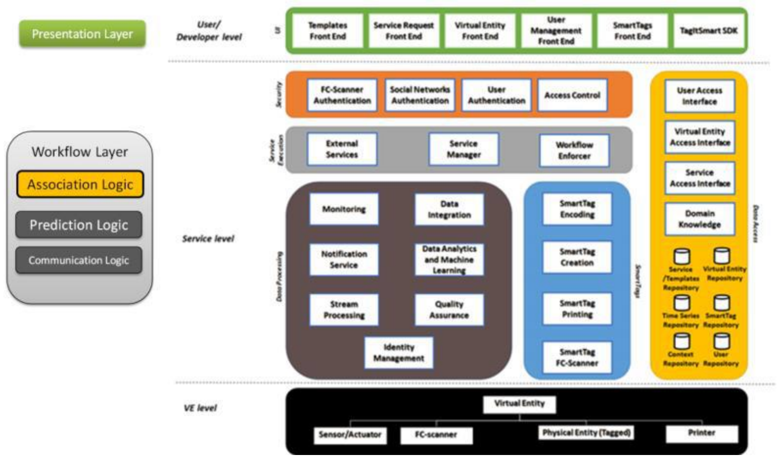


Figure 9: Main AffectUs functional blocks and relation to main TIS platform

The prediction mechanisms layer is responsible for the implementation of the main methods for estimation and specifically for exposing the interfaces so that these mechanisms can be used in real time (thus acquiring input conditions from TIS messaging services and transforming them to model input data, while producing the output estimated metric).

The communication logic layer is responsible for obtaining the aforementioned prediction output, correlating it with the needed limits and the affected products (through querying for the respective relationships stored in the KB) and generating the according events, while at the same time linking to the TIS notification services mechanism in order to forward the specific events to the interested entities.

The main integration point is with the EVRYTHNG platform of TIS, which represents the majority of the needed functionalities we need from the TIS Service level stack, like:

- Declaring a product, a tag etc., and the links between them
- Has a rules functionality to associate values with alerts from the base products
- Provides an API, which can be used in our case in a Node-RED flow with the API calls in order to automate the respective parts
- Has custom fields in the description of an entity, which maybe an indirect way to add our semantic extensions
- Can have redirects, which implies that we can detect a rule there and push the notification to a Node-RED endpoint for further processing, distribution based on our approach, or as input data to the prediction mechanisms, adding the product id as an argument in e.g. a post/get call
- Can define new action of our own and push the action data from Node-RED

5.2.1 Needed foreseen data inputs/outputs

The foreseen dependencies for automated data flows and inputs based on the envisioned functionalities include:

- Retrieval of all the available products from the linked accounts, so that the Product Stakeholder (or the AffectUs developer acting on their behalf) can correlate and structure the dependencies in their supply chain
- Retrieval of historical data from the Thngs (instances) of the dependent products in order to create/train the models and event detectors
- Retrieval of real time values for the Thngs (instances) in order to check the current status of the situation
- Access to the notification endpoints of the involved entities that need to be alerted based on the event effects of the first bullet

In relation to the necessary inputs from the external roles, these can be summarized in the following points:

- The product stakeholder needs to be able to declare specific aspects of the supply chain, such as location of each stage (if applicable), sequence of stages and potential dependencies from external products (e.g. used as raw material), whose info (or events) should also be propagated to the chain of this product
- The analytics developer needs to declare aspects of their event identification, such as general semantics as well as characteristics (e.g. different types of states identified etc.)
- The product stakeholder should also select which of these external events (and relevant states) are of interest to them (given that this affects their product in some sense)
- The application developer also inserts information, but not in the same abstract sense as the previous roles, given that they need also to specify specific flows e.g. in Node-RED. Thus they are more considered as users of some abstracted functionalities (such as model templates) but not in the same overall abstraction level that needs to be achieved for the other roles. The application developer needs to have a clear understanding of the overall AffectUs system in order to create the logic behind a given AffectUs instance.

5.2.2 Adaptation to the permissions model of the existing TIS framework

The main entry point in the TIS framework for AffectUs is considered to be the EVRYTHNG platform, which offers the majority of the functionalities in the Data Processing layer needed by AffectUs. EVRYTHNG has a variety of roles and access policies, based on OATH authentication, that are also extensible and adaptable to a given scenario and the need for access to specific resources. Therefore it is highly dynamic and suitable for an AffectUs developer to undertake one of the available roles and have access to a subset of the data of the Product Owner. More information on the details can be found in [14], however the role that fits best to our scope, given that the AffectUs developer (and system) is actually a collaborating but external entity to the product owner, is to undertake the Application Developer role of EVRYTHNG, with a project read capability (even at a limited subset of the project

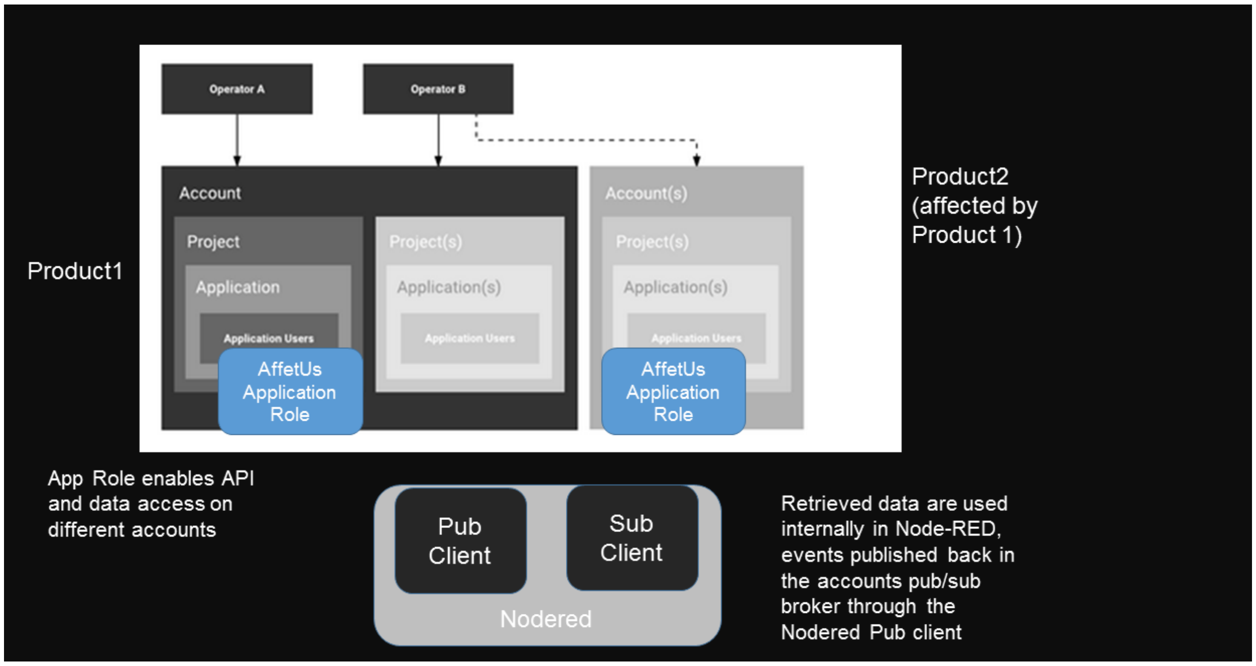
The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

resources). This enables us to register and consume data from the Product and Thng level, while it enables the product owner to conceal the level of information that they need not exposed.

The EVRYTHNG access model appears in the following figure (Figure 10). Assuming that AffectUs can undertake the application developer role in two different accounts (with account 2 being the account of a product that depends on the product of account 1, e.g. due to the fact that the latter as an input to the manufacturing process of product 2), this implies access to a set of predefined resources (configurable at which level by OATH). This way AffectUs can plug in the data flow and historical data repositories of product 1 scans, create the prediction models and then use the real time values of the flow in order to detect various types of events (such as delay in a specific stage, e.g. transport). The generated event can then be propagated to the EVRYTHNG data broker endpoint of account 2, in order to inform the according operator/account owner. The necessary connections/dependencies between products are defined by each product owner (in this case by product owner 2, after being displayed a list with the supported products of AffectUs, in which product 1 also resides). Thus AffectUs can act as the bridging middleware between different and dependent verticals.

5.2.3 Technical considerations on data retrieval

Based on the EVRYTHNG platform capabilities, and assuming that the permissions described in 5.2.2 are granted, the AffectUs extension can communicate via typical REST interfaces supported by EVRYTHNG in order to acquire the necessary data. The only part in this process that needs to be considered is whether the application role defined in a specific EVRYTHNG account has indeed the necessary configuration at the resource level needed and given the fact that this configuration is dynamic can be set at will per resource URL and HTTP verb.



The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

Figure 10: EVRYTHNG access model and relation to AffectUs

5.3 System Use Cases (Functional View)

The system use cases based on the UML standard that can be highlighted for the system are documented in the following chapters, based on the expected functionality that needs to be provided.

5.3.1 Generic Role interaction with AffectUs and EVRYTHNG

Based on the aforementioned analysis on the AffectUs positioning with relation to TIS and the authorization model that is followed by EVRYTHNG, the main integration point with TIS, the high level interactions between the roles, TIS EVRYTHNG platform and AffectUs extension are depicted in Figure 11.

The product owner follows the normal process on the EVRYTHNG platform and declares the characteristics of the product. Data related to this entity will be also needed by the Application Developer role, therefore the owner needs to exploit the OAuth authorization framework and enable the developer to retrieve data related to the product (main EVRYTHNG system). Furthermore, the product consumer needs to utilize the UI offered by AffectUs in order to declare the relationship and dependency from that product. Further information may be needed such as documenting and linking information e.g. from location to which chain stage the specific location relates to.

The Application developer in general has the main task of creating a generic model, exploiting the various available data sources (from dependent products or external events identification performed by the Analytics Developer) and combining them in a workflow fashion.

5.3.2 Create Model Use Case (Actor: Application Developer)

This is the primary use case of the Application Developer, aiming to extract a model that can investigate the various stages of the supply chain and detect events that have to do with transitions between the stages or with the positioning of the products related to them. In this case, they will exploit the captured relationships of the previous UC (as defined by the product stakeholders in e.g. location vs chain stage concepts) and select existing abstract methods in order to model timing transitions between stages. In this case the data retrieval is key (further specialized in the following subsection).

5.3.3 Retrieve Data Specialization (Actor: Developer)

The Retrieve Data functionality is separately presented, in order to remove some information from the previous UC and enhance readability, also because it includes the more generalized role of Developer. Data retrieval is one key aspect in the integration between verticals or with external sources. In the foreseen abstraction, the developer will be able to specify different means of obtaining the data, ranging from historical data to real time streams, potentially utilizing enriched semantic structures with relation to retrieval endpoints.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

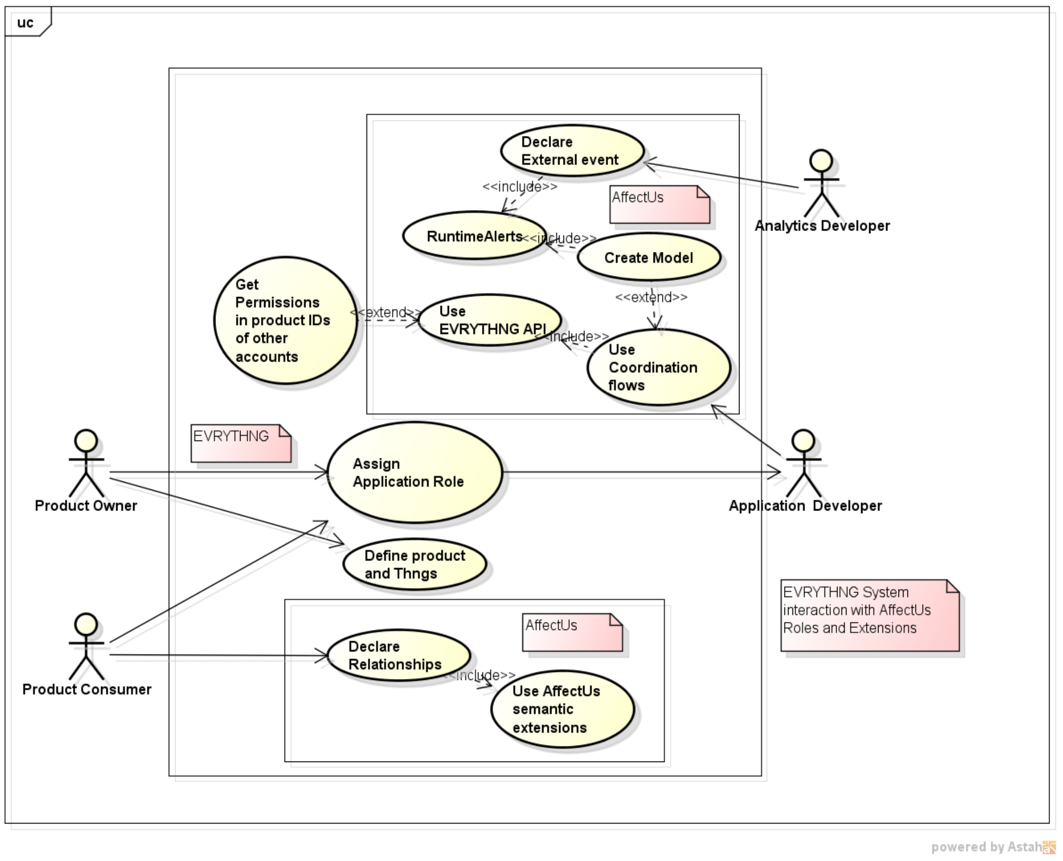


Figure 11: Generic Role interaction with the AffectUs system

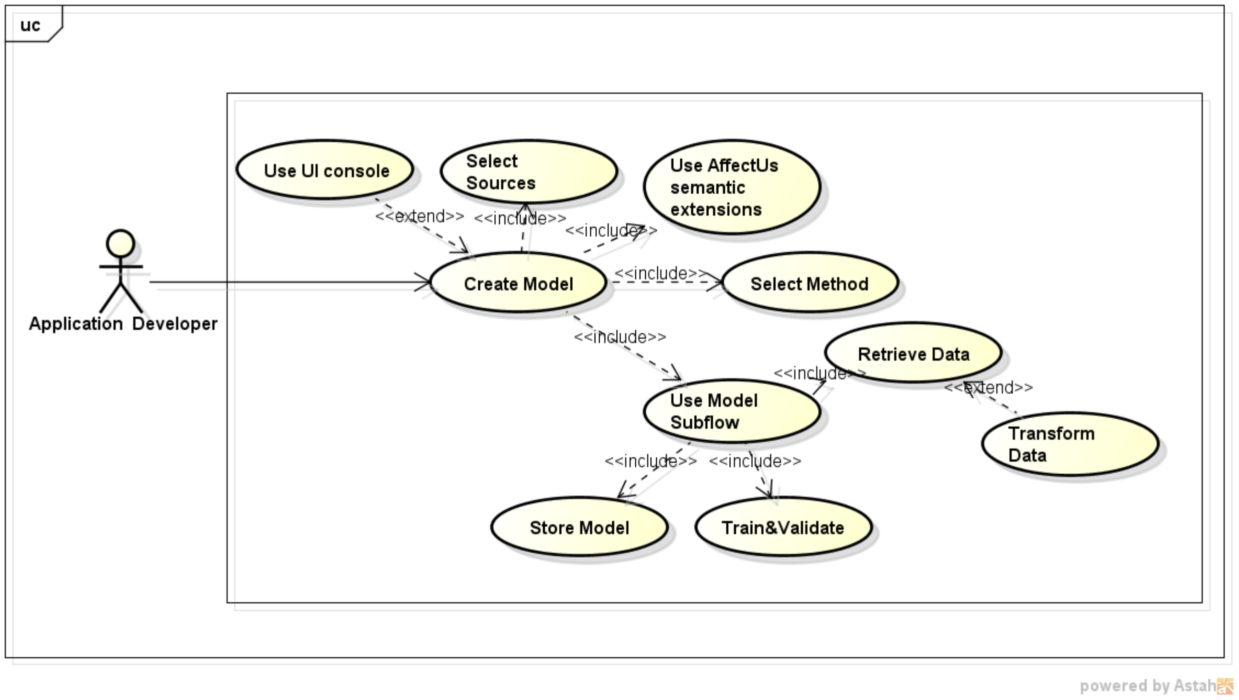


Figure 12: Create Model UC for the Application Developer

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

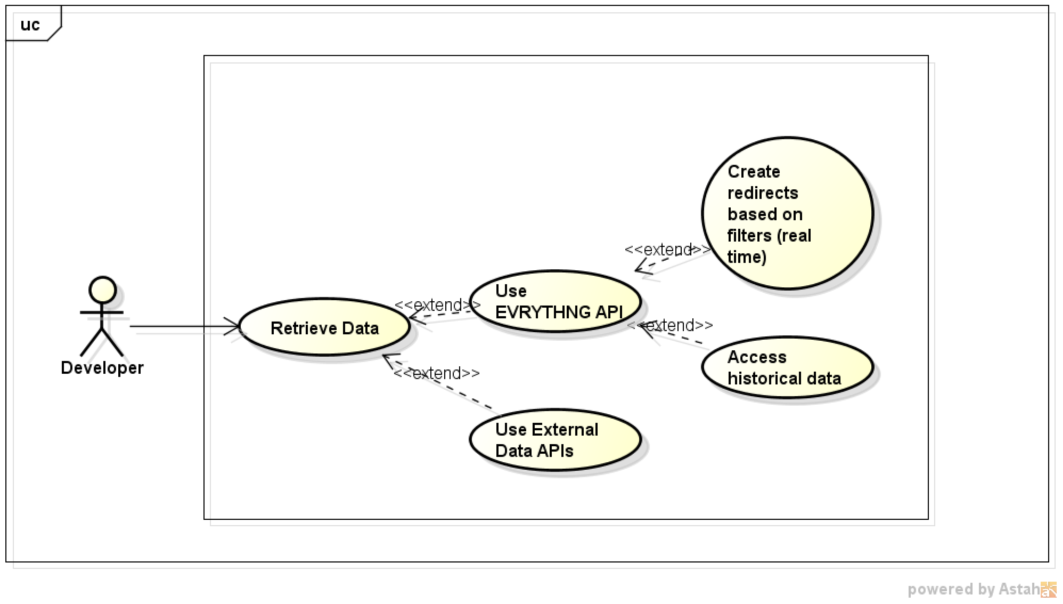


Figure 13: Retrieve Data Specialization Use Case

5.3.4 Create Coordination Flows (Actor: Application Developer)

In the specific Use Case the application developer is responsible for creating the overall coordination flow that will integrate the overall rationale of event detection and propagation through the respective entities. In this case pre-constructed abstract semantic queries and AffectUs subflows will be utilized in order to aid the developer in their task and speed up integration logic creation.

5.3.5 Receive Runtime Alerts (Actor: Product Stakeholder)

In this Use Case the product stakeholder may activate the runtime notifications coming from the AffectUs models. This should hide the flows created in Section 5.3.4 that aim at enabling all the needed backend functionalities, from retrieving the base data needed, to launching the model rule and detecting in real time the specific event, while pushing the final notification in the initially declared endpoint.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

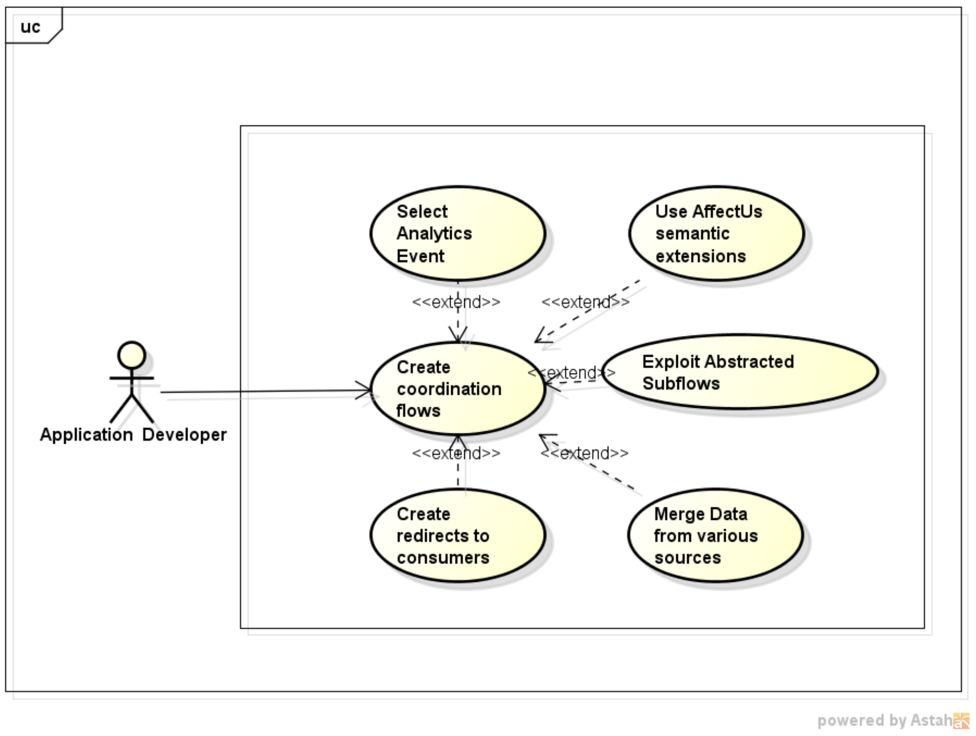
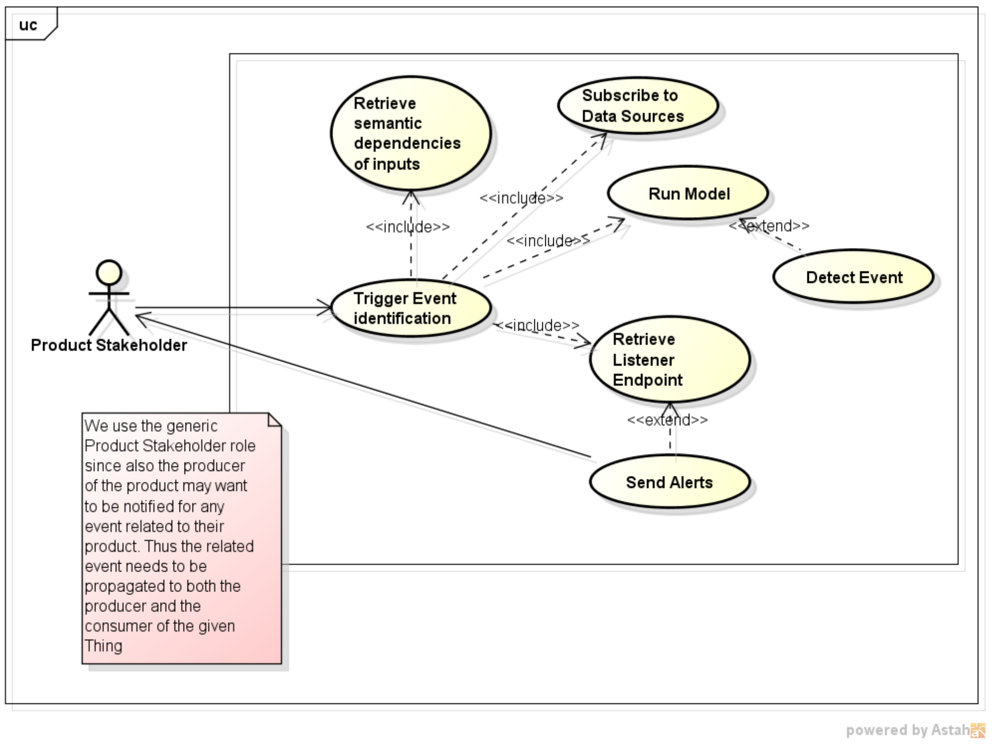

uc
Select Analytics Event
Use AffectUs semantic extensions
<<extend>>
<<extend>>
Create coordination flows
Exploit Abstracted Subflows
<<extend>>
Application Developer
<<extend>>
<<extend>>
Create redirects to consumers
Merge Data from various sources
powered by Astah

Figure 14: Create Coordination Flows



We use the generic Product Stakeholder role since also the producer of the product may want to be notified for any event related to their product. Thus the related event needs to be propagated to both the producer and the consumer of the given Thing

Figure 15: Receive Runtime Alerts Use Case

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

5.3.6 Declare External Event Use Case (Actor: Analytics Developer)

The external Analytics Developer may create the process of their event detection in whatever manner they see fit (or can reutilize abstract flows created by AffectUs), but the main point of involvement is the declaration on their behalf of the type of event generated, as well as some practical details such as publishing endpoint etc.). The purpose of this step is that this knowledge is documented in AffectUs KB, so that it can be combined by the Application Developer during the actions of model creation or the creation of coordination flows to redirect data.

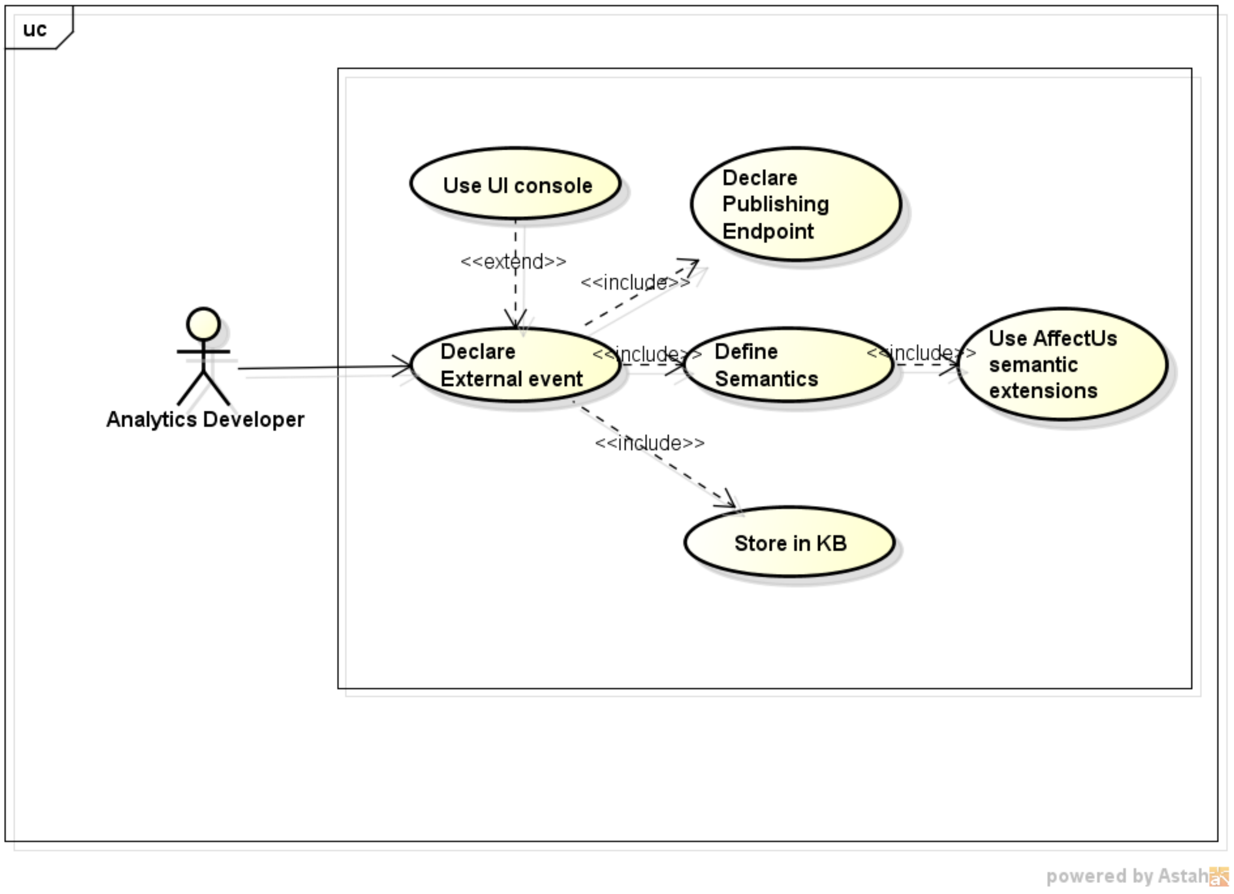


Figure 16: Declare External Event Use Case

5.4 *Component Diagram (Structural View)*

The AffectUs component diagram appears in the following figure (Figure 17). Interfaces are omitted for better visibility, given that they will also be included in the sequence diagrams of the next section. Instead we provide an overall description of the purpose and functionality of each component, starting from the UI elements and describing their relationship to the backend AffectUs components.

Declare external events: this element is created for the Analytics Developer in order to express the specific aspects of the provided notification. It should provide capabilities for declaring the general

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

category of the event (e.g. weather etc.), specific enumerations that have to do with the number and type of states declared, and parameters of the produced notification (such as time and location).

Declare Relationships: this UI element is to be used by the Product Stakeholder in order to declare dependencies between products. To this end it needs to be supported by the UI populator, which is responsible for retrieving product lists from the VE repositories and access interfaces of EVRYTHNG, as well as specific relationships (e.g. predicates) from the ontology definition behind the KB. This will populate drop down lists that will aid the stakeholder to proceed with the intended declarations. However the validity of these declarations needs also to be checked, for consistency purposes, either in the final submission stage or gradually during menu selection processes (thus if the subject is chosen from an initial list, the predicates and objects will need to follow the legal values based on the ontology definition).

Create Model: this UI is expected to be used by the Application Developer in order to declare aspects of the model to be created, as well as launch the model creation instance. One key aspect here is also the Data Processing component of the main system, which is responsible for transforming the raw data coming in from the EVRYTHNG Pub/Sub system, in order to apply aspects such as cleaning, normalization etc. of data, but primarily to map the input information to other necessary types (e.g. mapping the location in the scanned info with one of the supply chain stages). For this reason, the specific functionality needs to be integrated with semantic queries in order to retrieve this map and apply it for storing the data in the local DB in the proper form (including stage definition). When a model is created the result of the rule is stored as RDF. Each ontology relationship type is associated to an ACTION/RULE template, that can be either detected locally as a Node-RED flow (in case of a specific not supported aspect by EVRYTHNG alerts such as “not arrived scan”) or can be propagated to the EVRYTHNG platform and be submitted as a rule for value limit based alerts. On detection triggering, the specific rule template is populated with the given limit (as retrieved from the rdf triple of the specific case) and submitted via REST call to EVRYTHNG.

The Monitor State UI element is an optional dashboard element to indicate the overall state. The optional nature is due to the fact that notifications may be directly propagated to the respective publication structures of EVRYTHNG in order to reach the interested entities without the need for a further visualization stage.

In all the cases of the UI elements the respective information needs to be used in order to enrich the Ontology definition behind the KB or insert instances in the latter. Thus the UI translator component is used in order to adapt the respective information, check its potential validity (through the reasoner) and finally submit the data. Furthermore, and especially for the case of model creation, in which arbitrary logic may be inserted by the developer, a Flows repository functionality will be provided that contains template subflows that may be reused for generic purposes.

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

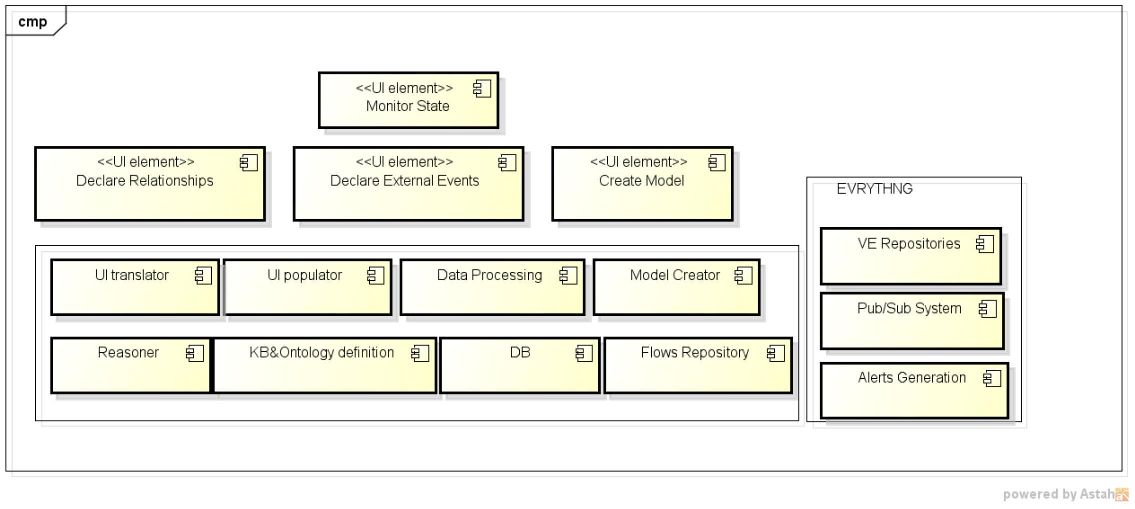


Figure 17: AffectUs generic component diagram

5.5 Sequence Diagrams (Behavioral View)

The behavioral view of the AffectUs system in order to support the aforementioned system use cases appears in the following paragraphs, supported by the components described in Section 5.4

5.5.1 Declare relationships

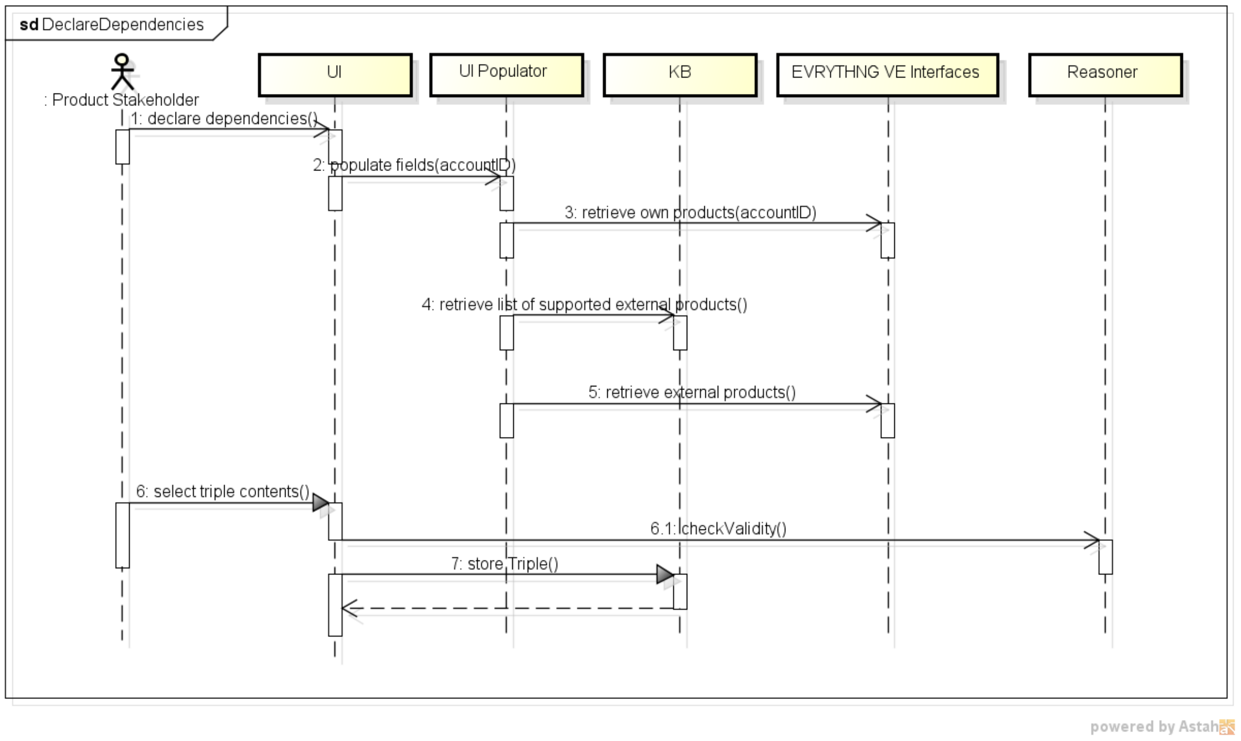

sd DeclareDependencies
: Product Stakeholder
UI
UI Populator
KB
EVRYTHNG VE Interfaces
Reasoner
1: declare dependencies()
2: populate fields(accountID)
3: retrieve own products(accountID)
4: retrieve list of supported external products()
5: retrieve external products()
6: select triple contents()
6.1: checkValidity()
7: store Triple()
powered by Astah

Figure 18: Declare Relationships sequence diagram

5.5.2 Create model functionality

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

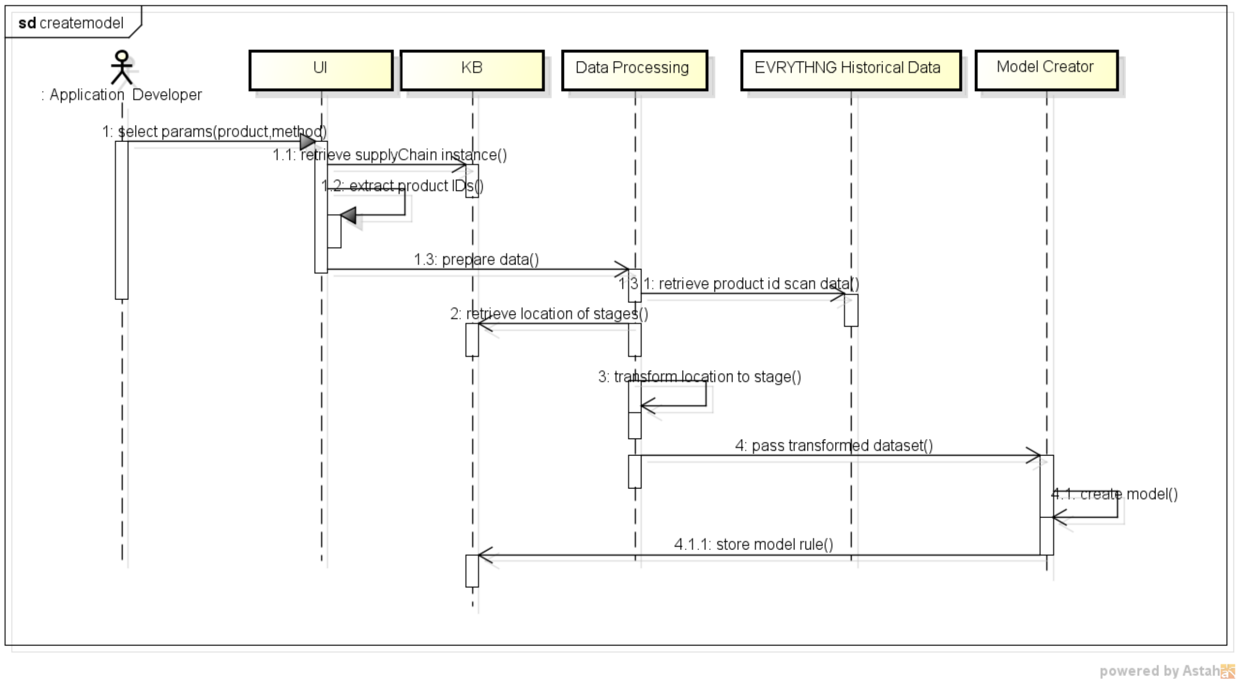

sd createmodel
: Application Developer
UI
KB
Data Processing
EVRYTHNG Historical Data
Model Creator
1: select params(product,method)
1.1: retrieve supplyChain instance()
1.2: extract product IDs()
1.3: prepare data()
1.3.1: retrieve product id scan data()
2: retrieve location of stages()
3: transform location to stage()
4: pass transformed dataset()
4.1: create model()
4.1.1: store model rule()
powered by Astah

Figure 19: Create model sequence diagram

5.5.3 Create coordination flows

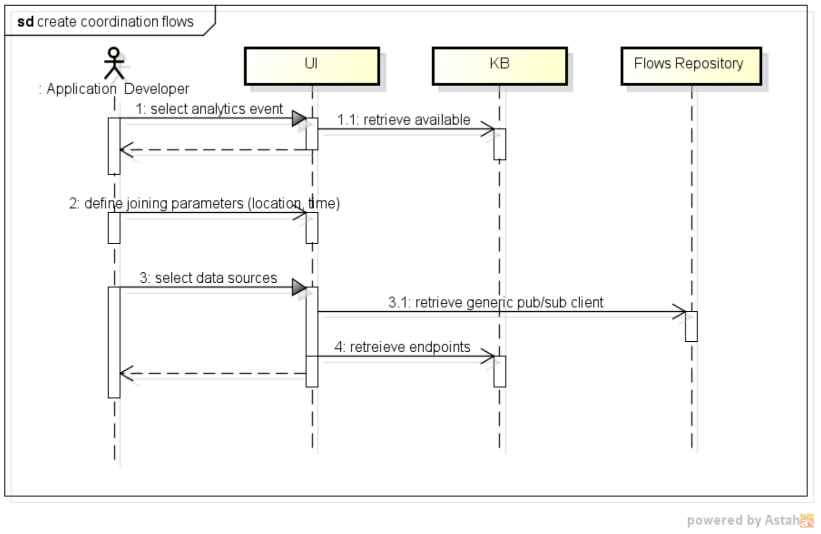

sd create coordination flows
: Application Developer
UI
KB
Flows Repository
1: select analytics event
1.1: retrieve available
2: define joining parameters (location, time)
3: select data sources
3.1: retrieve generic pub/sub client
4: retreieve endpoints
powered by Astah

Figure 20: Create coordination flows sequence diagram

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

5.5.4 Declare external event

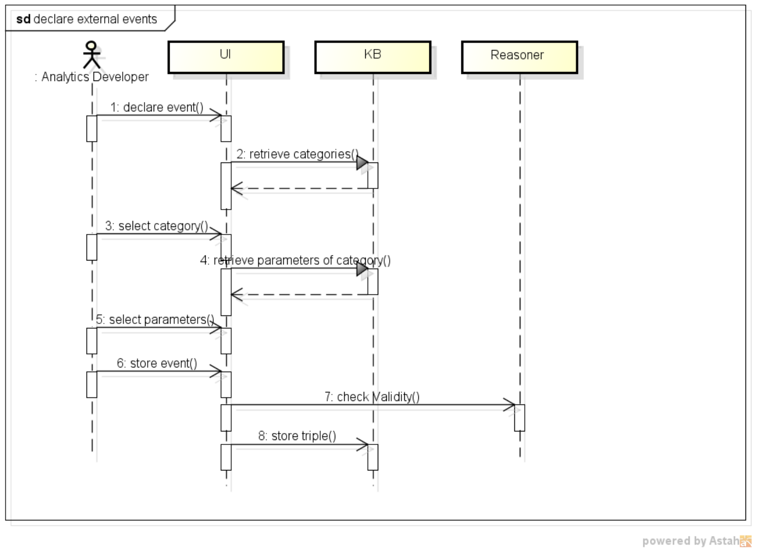


Figure 21: Declare external event sequence diagram

5.5.5 Trigger event identification

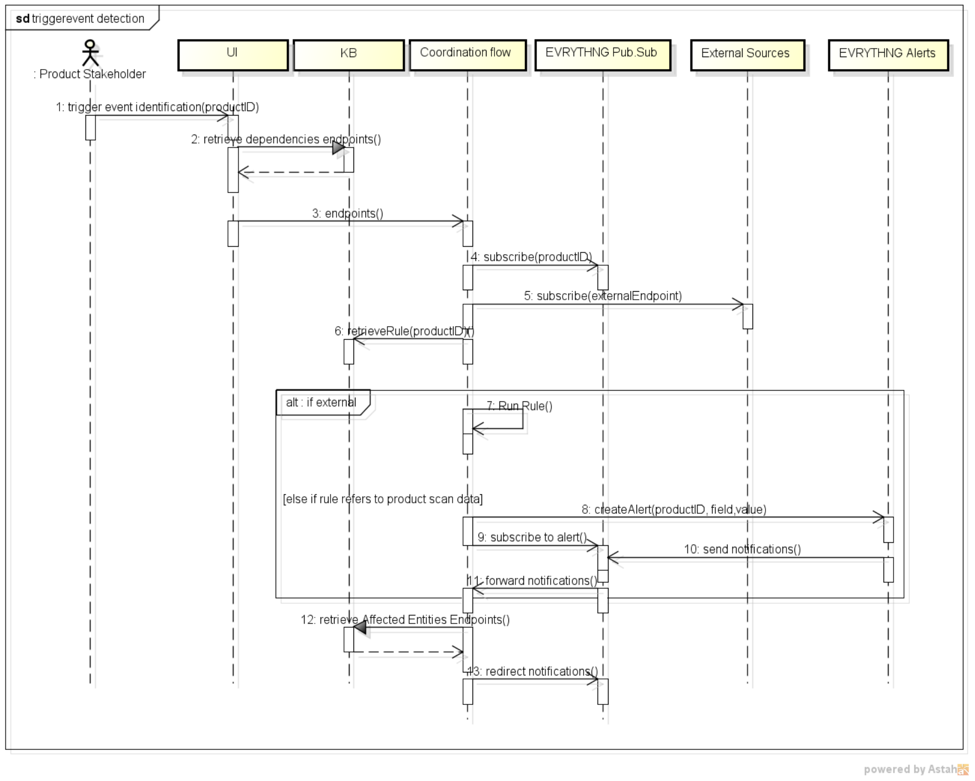

sd triggerevent detection
: Product Stakeholder
UI
KB
Coordination flow
EVRYTHNG Pub.Sub
External Sources
EVRYTHNG Alerts
1: trigger event identification(productID)
2: retrieve dependencies endpoints()
3: endpoints()
4: subscribe(productID)
5: subscribe(externalEndpoint)
6: retrieveRule(productID)()
alt : if external
7: Run Rule()
[else if rule refers to product scan data]
8: createAlert(productID, field,value)
9: subscribe to alert()
10: send notifications()
11: forward notifications()
12: retrieve Affected Entities Endpoints()
13: redirect notifications()
powered by Astah

Figure 22: Trigger event identification sequence diagram

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

6 Conclusions

In order to reach the initial version of AffectUs design specification, the process was based on the UML paradigm, starting from requirements drafting and reaching the level of system sequences in order to aid implementation efforts.

During this process, all sources of requirements, stemming from innovation goals, external audiences, existing TIS capabilities and features have been taken under consideration in order to reach a complete system design, including an integration scope with the existing TIS functionality.

In terms of coverage with relation to the main functional requirements identified (REQS1 to 4 and 6), the following table summarizes in which system UC and respective sequence they have been addressed. With relation to the non functional REQ5 (dockerization), this is addressable by creating one instance of AffectUs per application developer, while having multitenant versions of specific components like the DB backend.

Table 15: Requirements Traceability Matrix

Requirements	System UC (by name)	Component	Sequence	Other
Affect_REQ1 (Inclusion of external events developers that offer generic notifications)	Declare External Events	According UI element, KB rule, messaging flow	Declare External event	Incorporation in Ontology concepts
Affect_REQ2 (Shared IDs rationale between TIS platform and AffectUs)	Generic role interaction UC	KB, data processing, coordination flows	Declare relationships, Create model, Trigger event identification	N/A
Affect_REQ3 (Abstract and adaptable supply chain analysis)	Create model	According UI element, Model Creator, KB structure, DB	Create Model	Incorporation in Ontology concepts for correlating chainstage to location
Affect_REQ4 (UI insertion for dependencies)	Declare External Events, Declare Relationships	According UI elements, Reasoner, KB	Declare Relationships, Declare external event	N/A

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

Affect_REQ6 (smart tag “inner” sensor values consideration)	Receive runtime alerts	KB, coordination flows	Trigger Event identification	Definition of events in Section 2.1, incorporation in ontology concepts for storing as an EventConditionInstance and the relation to which sensor field it references

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.

7 References

[1]. E. Psomakelis, K. Tserpes, D. Anagnostopoulos, and V. Theodora, “Comparing Methods for Twitter Sentiment Analysis,” Proc. 6th Int. Conf. Knowl. Discov. Inf. Retr. - KDIR 2014, pp. 225–232, 2014.

[2]. E. Psomakelis, F. Aisopos, A. Litke, K. Tserpes, M. Kardara, and P. M. Campo, “Big IoT and social networking data for smart cities: Algorithmic improvements on Big Data Analysis in the context of RADICAL city applications,” ArXiv Prepr. ArXiv160700509, 2016.

[3]. F. Aisopos, G. P. 0001, K. Tserpes, and T. A. Varvarigou, “Content vs. context for sentiment analysis: a comparative analysis over microblogs.,” in HT, 2012, pp. 187–196.

[4]. B. Pang and L. Lee, “Opinion mining and sentiment analysis,” Found. Trends Inf. Retr., vol. 2, no. 1–2, pp. 1–135, 2008.

[5]. B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment Classification Using Machine Learning Techniques,” in emnlp2002, Philadelphia, Pennsylvania, 2002, pp. 79–86.

[6]. B. Pang and L. Lee, “A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts,” in In Proceedings of the ACL, 2004, pp. 271–278.

[7]. J. R. Quinlan, “Improved use of continuous attributes in C4.5,” J. Artif. Intell. Res., vol. 4, pp. 77–90, 1996.

[8]. G. H. John and P. Langley, “Estimating continuous distributions in Bayesian classifiers,” in Proceedings of the Eleventh conference on Uncertainty in artificial intelligence, 1995, pp. 338–345.

[9]. H. Shi, “Best-first Decision Tree Learning,” Department of Computer Science, University of Waikato, 2007.

[10]. D. A. Salazar, J. I. Vélez, and J. C. Salazar, “Comparison between SVM and Logistic Regression: Which One is Better to Discriminate?,” Número Espec. En Bioestad., vol. 35, pp. 223–237, 2012.

[11]. T. Mullen and N. Collier, “Sentiment Analysis Using Support Vector Machines with Diverse Information Sources,” Barcelona, Spain, 2004, pp. 412–418.

[12]. S. Haykin, Neural Networks: A Comprehensive Foundation. New York: Macmillan College Publishing, 1994.

[13]. Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[14]. EVRYTHNG Roles and permissions: https://developers.evrythng.com/docs/roles-and-permissions

[15]. George Kousiouris, Adnan Akbar, Juan Sancho, Paula Ta-Shma, Alexandros Psychas, Dimosthenis Kyriazis, Theodora A. Varvarigou: An integrated information lifecycle management framework for exploiting social network data to identify dynamic large crowd concentration events in smart cities applications. Future Generation Comp. Syst. 78: 516-530 (2018)

[16]. George Kousiouris, George Vafiadis and Theodora Varvarigou "Enabling Proactive Data Management in Virtualized Hadoop Clusters Based on Predicted Data Activity Patterns ", in Proceedings of P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2013 Eighth International Conference on , vol., no., pp.1,8, 28-30 Oct. 2013

The AffectUs project is a TagItSmart (Grant Number 688061) Open Call 1 Extension project co-funded by the European Union (EU) Horizon 2020 program.