SALTED

Situation-Aware Linked
heTerogeneous Enriched Data

D2.1: Report on Data Linking and Enrichment Architecture

Work package	WP 2
Task	Task 2.1
Due date	30/06/2022
Submission date	30/06/2022
Deliverable lead	NEC
Version	1.0
Authors	Jonathan Fürst (NEC), Luis Sánchez (UC), Jorge Lanza (UC), Juan Ramón Santana (UC), Pablo Sotres (UC), Victor González (UC), Laura Martín (UC), Anja Summa (Kybeidos), Maren Dietzel (Kybeidos), Stephan Frenzel (Kybeidos), Guzmán Gutiérrez (AMPER), Noel Crespi (IMT), Praboda Rajapaksha (IMT), Amir Reza Jafari Tehrani (IMT)
Reviewers	Roberto Minerva (IMT), Ernö Kovacs (NEC), Luis Sánchez (UC)

This project is co-financed by the Connecting Europe Facility of the European Union under the Action Number 2020-EU-IA-0274.

Abstract	This document, developed by the SALTED project, represents the D2.1 deliverable of the data linking and enrichment architecture, in the following also refered to as Data Enrichment Toolchain (DET) architecture. The focus of this document is the identification of different components and interfaces within the SALTED architectural building blocks, as well as the different adaptations necessary for a successful use-case deployment. Further, D2.1 defines the role of each of the blocks as well as the required interfaces for the communication between blocks. The architectural design choices are based on the requirements in terms of the scope of available data sources phased in each use case.
Keywords	Data Linking and Enrichment, Data Enrichment Toolchain, Architecture

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

1.1 Scope of Document

The aim of Situation-Aware Linked heTerogeneous Enriched Data (SALTED) is to add value to existing datasets and data-streams by enriching them through the application of the principles of linked-data, semantics and Artificial Intelligence (AI) and publish the enriched data sets in NGSI-LD as Open Data. This document D2.1 provides an overview of the architecture of the Data Linking and Enrichment processes thus defining the *Data Enrichment Toolchain Architecture* (short DET architecture). Through that it describes the main project contributions that will be developed and evaluated during the lifetime of the project.

1.2 Target Audience

The Architecture Design Document is mainly intended for internal use, although it is publicly available. The target audience is the SALTED technical team including all partners involved in the delivery of Work Packages 1,2 and 3 but also it serves as reference for the developers of the situation-aware applications in Work Package 4. Moreover, it provides a thorough review of the key functionalities that the SALTED DET will support and how data will be processed in order to fulfil the overall project aim of publishing semantically enriched data.

1.3 Structure of the Document

We first describe the requirements based on the scope of available data sources in the SALTED use cases (Section 2). In Section 2 we provide an overview of the Data Enrichment Toolchain (DET) architecture, its main components, and interfaces (data and control flow). We then explain the internal working of each component in more detail in Section 3. Last, we take an initial look at possible instantiations of this architecture and its components for the SALTED use cases.

2 REQUIREMENTS

In this chapter we derive requirements for the SALTED DET based on the scope of available data sources.

2.1 Scope of Available Data Sources

Proliferation of data sources associated to Internet of Things (IoT) deployment as well as those bound to Open Data Portals (e.g., European Data Portal, Municipalities Open Data Portals, etc.) and Social Media platforms is creating an abundance of information that is called to bring benefits for both the private and public sectors, through the development of added-value services, increasing administrations' transparency and availability or fostering efficiency of public services. However, pieces of information without a context are significantly less valuable.

The *SALTED Data Enrichment Toolchain (DET)* main aim is to add value to existing datasets and data-streams by enriching them through the application of the principles of linked-data, semantics, and Artificial Intelligence (AI).

Precisely, the starting element of the DET are the *Injection Chains* that have to analyze raw data sources and generate the correspondingly normalized data elements using NGSI-LD as the information model. The heterogeneity of the original data sources and the need for harmonizing all of them following a systematic approach sets the initial requirements on the DET as it is necessary to define an architecture that can deal with such a heterogeneity.

To systematically address the design of the injection path within the DET architecture, an analysis of the planned data sources that will be leveraged during the project has been done. The objective of this analysis has been to make a catalogue of the features that the data sources, and the actual data that they produce, exhibit so that the design of the DET is made to fulfil the requirements imposed by these features.

Firstly, we consider the two application domains that will be targeted in the project as proof-of-concept scenarios. They are the Smart City and the Smart Agriculture domains. Regarding the first one, it comprises a wide variety of systems that makes it heterogeneous by itself. Moreover, the amount of businesses and people that participates in the scenario, makes it representative of the three types of data sources that SALTED DET is targeting (i.e. IoT deployments, Open Data and Social Media). Pertaining to the second scenario, it is representative of a more specialized environment in which the key characteristic is probably the volume of data (considering the potentially large extensions that it can cover) as well as the availability of data objects like text, and mainly images, that has specific conditions to be treated.

The second main aspect that have been considered in the analysis was the formats employed by the data sources to make information available. In this respect, DET has to be designed to support structured data (typical for IoT deployments whose information is accessible through data management platforms normally exporting web interfaces serving data as JSON objects), semi-structured data (typical for Open Data portals and Social Media, whose information has some pre-established format but might mix different information styles), and unstructured data (typical for information available on the Web including the semi-structured text of the web page as well as the visual data types like images or videos). Each kind of data format imposes different restrictions to the discovery and crawling part of the injection chain. Not only on the methods used to extract the information and define the mappers towards the common NGSI-LD information model, but also on the way the injection chain is parametrized to point to the appropriate data source (well-known interfaces with dedicated endpoints in the case of the sources of structured data, versus, a priori, undefined data sources in the case of the sources of unstructured or semi-structured data).

Another mayor feature of the data sources that affects the DET design is their static vs real-time nature. This is the fact that some data sources will provide static datasets while others will make available continuous data-streams. Since the core of the DET information handling is the NGSI-LD standard and it supports both access to real-time and historical values, the two types of data sources are supported. However, for the specification of the respective injection chains, particularly for the case of data sources service static datasets, it is necessary to define the procedure to be followed for the processing of such datasets and how the associated NGSI-LD entities and attributes are updated upon its historical evolution contained in the same dataset (i.e. the same entity and/or attribute might have multiple sequential instances at different timestamps within the same dataset).

Finally, the classical “Vs” characteristics such as volume, variety, velocity and veracity of the available data has been identified as particularly relevant for the specification of the DET architecture. In this sense, mainly the volume and velocity have been considered. The DET must concentrate on the enrichment of data avoiding unnecessary replication of redundant data. In this sense, the injection chains might have themselves their own knowledge extraction algorithms that produce the pieces of information that are to be handled within the DET and make available through it. However, the raw data used to feed these knowledge extraction modules shall only be temporarily handled at the DET and dismissed after the applied algorithm has concluded.

3 SALTED DATA ENRICHMENT TOOLCHAIN (DET)

The data linking and enrichment architecture mechanisms proposed by SALTED are supported by the combination of different Data Enrichment Toolchains (DET).

DETs can be defined as the composition of heterogeneous microservices which results in the progressive enhancement of the original information quality and value. Conceptually, a DET can be understood as a pipeline with different sets of components, each one targeting a specific step within a particular data source improvement cycle. This cycle is carried out in an iterative manner and some of these components can even be parametrized in a dynamic way.

Finally, it is important to note that, even though some specific components are reused and shared by different DETs, all deployed DETs within the SALTED ecosystem are managed as parallel and independent workflows. Note: reuse of components can be by instantiating the same code into different DETs or be linking a running components into different DETs. The last model is used to reduce redundant data access, storage, and computation processes.

Another major point is that enrichment functions and context awareness strongly depend on the type of data sets used and on the specific goals of the applications. For this reason, it is envisaged that formatting and data management functions will have a more general applicability compared to enrichment and situation awareness support features.

The objective of the architecture is to support the needs of different applications. In order to satisfy this requirement, the following functions have been identified as key enablers of the architecture:

- *data discovery*, i.e., the ability to discover and request the collection of sets and streams of data;
- *data formatting*, i.e., the transformation of raw data into well-formed and structured set of data accordingly to data models described in terms of NGSI-LD;
- *data curation*, i.e., the identification (and potential correction) of data that do not reflect the expected quality (outliers, errors in values and the like);
- *data linkage*, i.e., the ability to relate different data set accordingly to well established definition of relationships;
- *data enrichment*, i.e., the ability to understand and frame the data structures according to situations and contexts and the definition of functions that exploit this contextualization.

3.1 Architecture overview

SALTED architecture provides a scalable framework to address the particularities of the use of linked and enriched data, while providing a solution to enable the configuration of its modules. The architecture has been defined considering the following premises:

- Data sources can be *batch*, providing data under request (e.g. RESTful interfaces from a CKAN-based Open Data Portal); or *real-time*, providing data as soon as it is generated (e.g. under publish/subscribe based services). The SALTED architecture is flexible in order to collect and deal with data. Two operational modes are envisaged:

  - "batch mode" accesses already defined data sets and transform them into NGSI-LD data sets; or
  - "real-time mode" collects and process data from sources while they are produced.

- Data must be provided following the Linked Data Design Issues [1].
- Data must be curated, limiting the data garbage provided to SALTED application.
- Data can be dynamically linked based on changes in the data (e.g., a newly created entity might result in a sameAs link to an existing entity).
- Data can be enriched with new properties as they are created where properties can be new data attributes or new relationships, e.g. to express structural contexts relationships or dynamic situation descriptions.
- SALTED components must be configurable either by SALTED Apps or other SALTED components (e.g., a data collection component must be configurable with a new set of social media keywords to collect posts for).

The architecture can be then divided in two different planes, depending on whether it corresponds to the data or control flow. The data plane builds on the NGSI-LD standard and enables components to add/query or subscribe to data based on a well-defined API. The control plane enables a configuration of different components (e.g., configuration of websites to crawl) and enables their orchestration where an orchestration cannot be achieved through the data plane through subscribe mechanisms (e.g., a component can get notified when a new NGSI-LD entity is created). Both planes are described in detail in Section 3.1.1 and 3.1.2 respectively.

3.1.1 SALTED Data Plane

For the Data Plane, the architecture comprises a set of DET components that will provide well-defined functionalities. They will be using the Brokering Functionalities, represented as the core component (Context Broker). It provides the means for external apps to access the treated data from SALTED and it is in charge for allowing the communication of data among components of the application as well as the storage of well formatted and treated data. SALTED Data Plane architecture is shown in Figure 1.

SALTED data is generated through a process that leads to the linking and enrichment of the data that has been previously curated and transformed following the NGSI-LD information model and employing any of the Smart Data Models [2]. This data has been extracted from heterogeneous data sources. This process is carried out by the different DET components, which are a fundamental part of the architecture. They can be divided in two main groups: one group, so called *Injection Chain*, supports a data pipeline process that yields to the generation of curated NGSI-LD data from raw data; and the other, so-called *Enrichment Chain*, relates to the enrichment and linking NGSI-LD data through an iterative process, producing the so-called SALTED data.

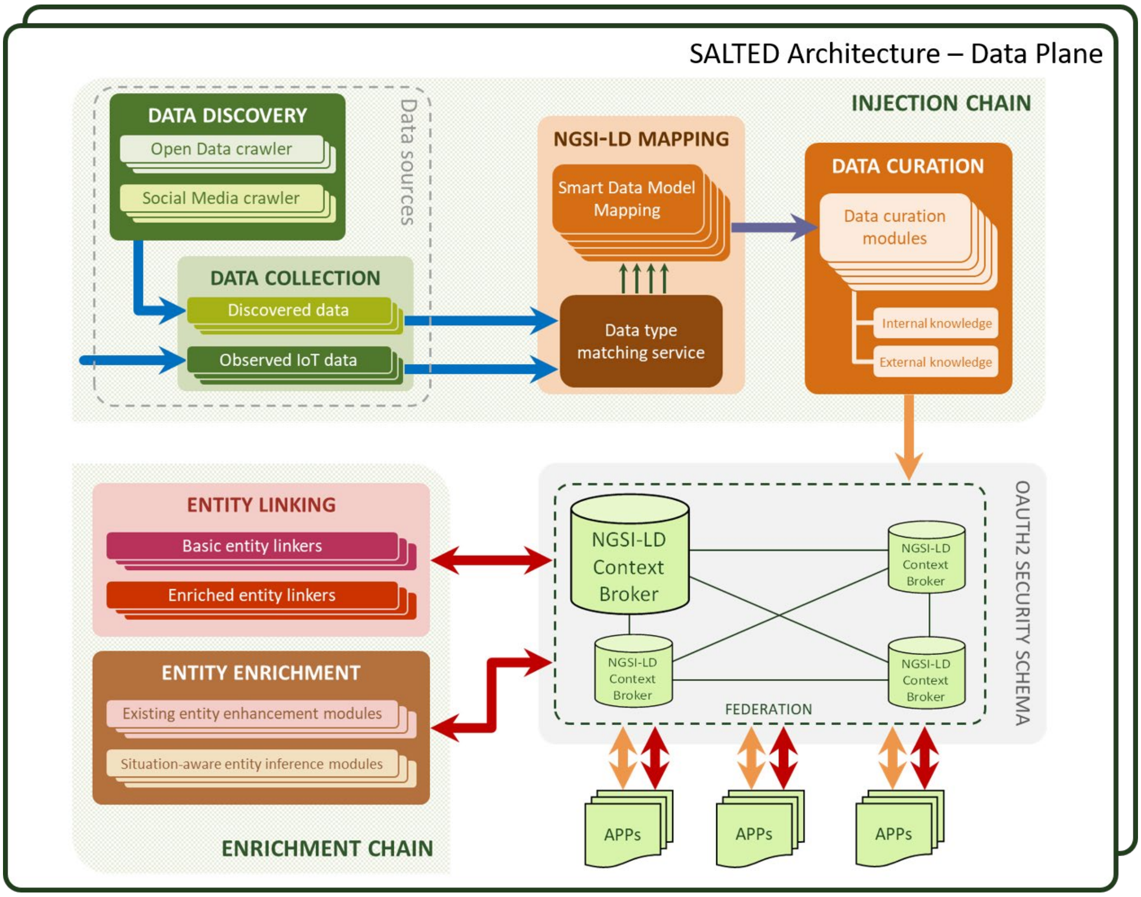


Figure 1. SALTED Data Plane architecture

Pipeline DET components, related to Heterogeneous data sources semantization (i.e. injection chain), are in charge of providing the tools to discover and collect the data from heterogeneous data sources; transform these data into NGSI-LD data, being compliant with existing data models; and curate the data before making it available to external applications by updating the Context Broker. DET components involved in this group are briefly described as follows:

- **Data Discovery and Data Collection**. This component provides the means to gather data from heterogeneous sources, such as crawling data from informational webpages, social media or Open Data platforms, or receiving IoT data from a Publish/Subscribe mechanism. In-depth details about this component, and the instances being implemented, can be found in Section 4.1.
- **NGSI-LD Mapping.** This component transforms the data collected from the Data Discovery and Data Collection component to NGSI-LD Data, mapping existing, either from custom or standard formats (e.g. NGSIv2). Information is forwarded to the Data Curation component. More information on the instances related to the NGSI-LD Mapping can be found in Section 4.2.
- **Data Curation**. It is in charge of assessing the quality of the data being injected into the NGSI-LD Context Broker, introducing relevant metadata information about each

observation or entity (e.g. precision or accuracy). Additional details on Data Curation components can be found in Section 4.3.

SALTED does not enforce the communication mechanisms employed by the different DET components that belongs to the data pipeline group. Different mechanisms may be used depending on the type of data (streams or batch for example). Only the final step has to conform to the interfaces and mechanisms defined by the NGSI-LD's Context Broker thought it is recommendet to use NGSI-LD as early as possible allowing to re-use intermediate results in a standard way. This way data will be NGSI-LD compliant, and it will be available to the NGSI-LD context broker. Besides, data must be as much as possible compliant with existing FIWARE Smart Data Models [2], proposing new ones or extensions to existing ones if they do not fit for the data being collected.

The enrichment and situational process comprises DET components (Situation-aware Linked-Data enrichment chain) in charge for data linking, and enrichment and contextualization. This process is carried out iteratively. As it will heavily depend on the data being collected in SALTED, it can result on the discovery of new links among data entities or the creation of new enriched properties or entities. Noticeably, both components must be NGSI-LD compliant, being at the same time consumers and producers of NGSI-LD data. There are two main components in charge of the entity linking and enrichment processes.

- **Entity Linking**. This component provides the means to link entities among each other. This way discovered and collected NGSI-LD data is interlinked, regardless its data source. A detailed overview of this component and the instances within SALTED can be found in Section 4.4.

- **Entity Enrichment**. In this case, the process carried out in this component is meant to enrich existing entities with new properties, or generate new entities based on the existing ones. A detailed overview of this component and the instances within SALTED can be found in Section 4.5.

Finally, the core component of the DET architecture is the NGSI-LD Context Broker, which implements the NGSI-LD interfaces. It enables the linking and enrichment process on top of curated NGSI-LD data, providing access to external applications retrieving and making use of SALTED data. Considering this component as the core of the SALTED architecture, security procedures will be implemented to ensure that the NGSI-LD interfaces exposed by the Context Broker (e.g. OAUTH2, JWT, TLS, …) are safe/secure.

Figure 2 depicts the entire process of transformation and enrichment for making raw data into SALTED data.

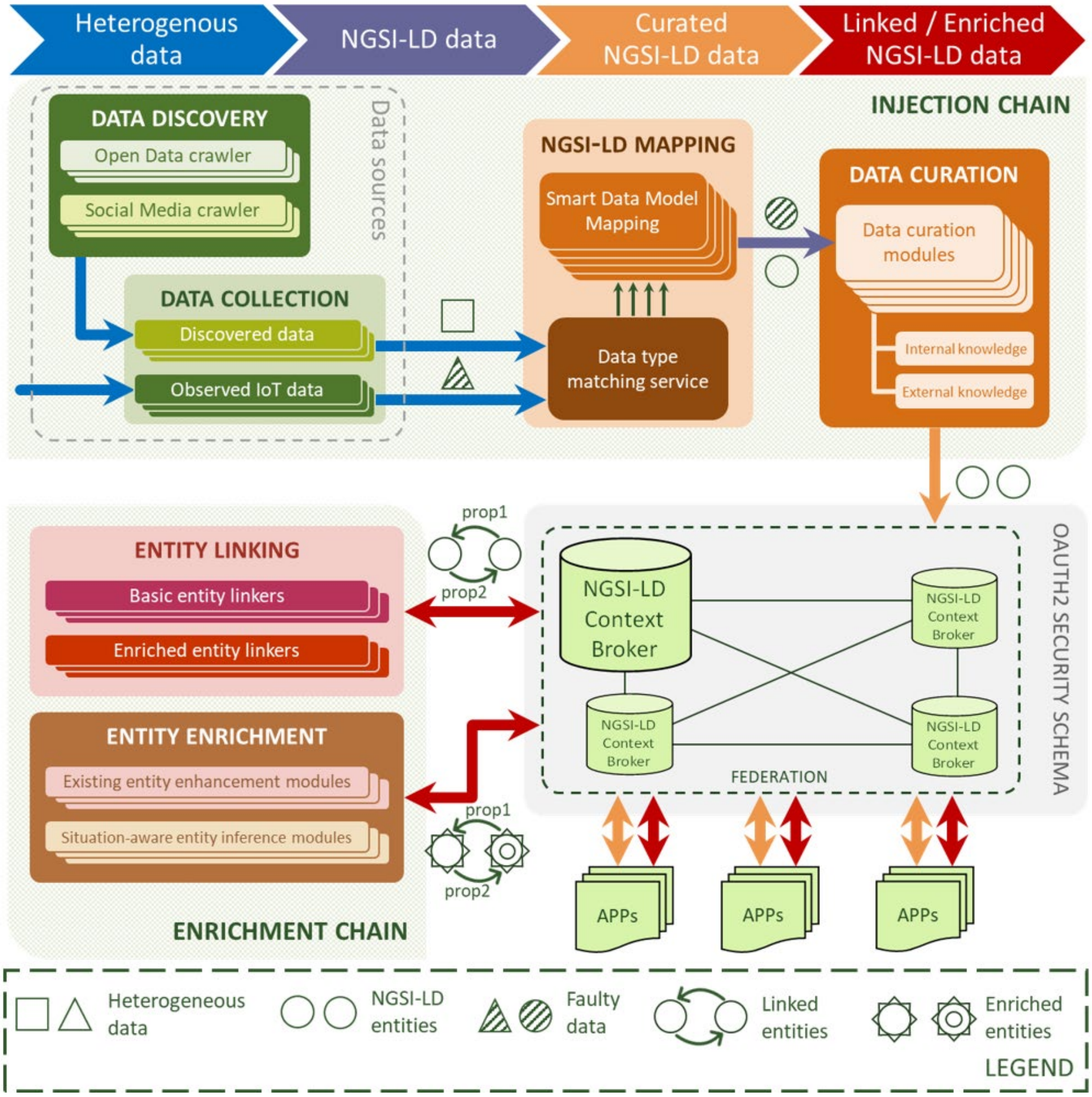


Figure 2. SALTED architecture data flow

3.1.2 SALTED Control Plane

SALTED architecture has been defined considering the provision of linked and enriched data from curated NGSI-LD data. Each DET component has a well-defined set of functionalities to process data coming from the NGSI-LD context broker eventually. Besides, some of the DET components are configurable, providing additional functionalities through parametrization, which can be modified by either external or internal applications. To this end, SALTED architecture envisions two different planes, one for data management and an additional one to enable applications to configure DET components, the Control Plane.

The Control Plane has been conceived following two premises. First, it should be completely decoupled from the Data Plane, thus avoiding sharing the same components for control-related functionalities. Second, control interfaces must be as simple as possible using a general approach, avoiding the need of implementing different interfaces depending on the component to be configured, easing the interaction between components and applications.

Figure 3 shows the control plane for the SALTED architecture. The main difference between data and control planes resides on the replacement of the NGSI-LD Context Broker by a specific component, namely the Control Broker, that provides the means for the communication between any DET component and the applications that might be interested in configuring it. Moreover, the Control Broker will be accessible by any other DET component, including those in the data pipeline from the Data Plane which did not have direct access to the NGSI-LD Context Broker.

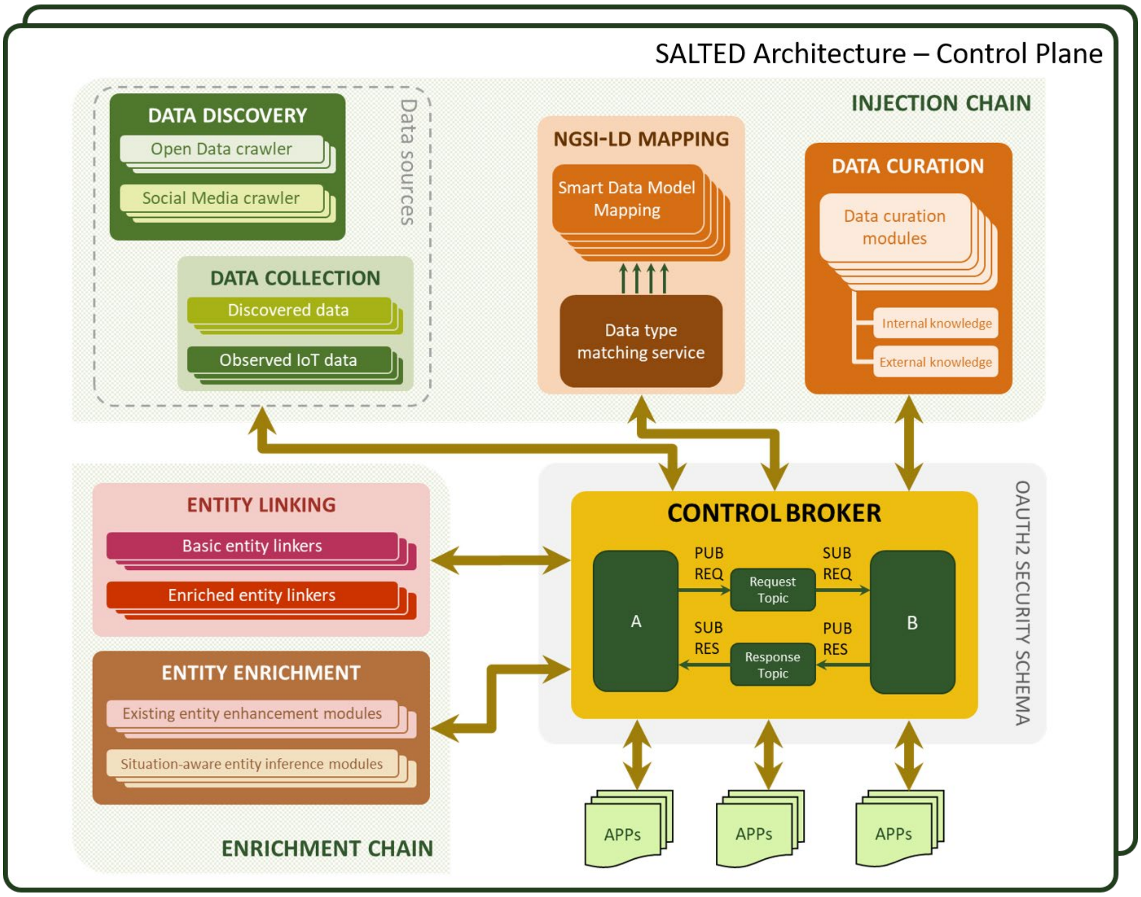


Figure 3. SALTED Control Plane architecture

Control management and configuration follows the Inversion of Control (IoC) pattern so as to enable the communication between DET components while avoiding them to know the specifics of the rest of the components or their endpoints. More precisely, Applications or any other DET component will not be required to call the components separately, but only through the Control Broker. Moreover, this way the security risks of the platform are also reduced as the individual components are not exposed to external applications.

Communication through the Control Broker will be carried out following the Pub/Sub event-based mechanism in which components will be subscribed to specific topics. These topics will be used to publish component-specific parametrization commands by any user or Application that might want to update or change the configuration of the DET component, using a predefined format depending on the configurable parameters available by each component.

More information about the proposed implementation can be found in Section 3.2.2.

3.2 Interfaces/Integration Plan

In this section, we explain how SALTED components interact through the Data Plane (Scorpio NGSI-LD broker) and through the Control Plane (MQTT).

3.2.1 Data Plane: Scorpio NGSI-LD Context Broker

In SALTED, NGSI-LD has been the agreed output format in which data will be published as Open Data. Further, NGSI-LD is the internal data protocol in which SALTED components exchange information.

The Scorpio context broker implements the standardized NGSI-LD protocol to represent and exchange linked data, especially in an IoT context. Through the NGSI-LD protocol, context producers and consumers can interact with each other. For example, a city can be equipped with IoT sensors to monitor various aspect from transportation to emissions. All these IoT devices act as context producers. On the context consuming side, there might be various applications that perform further processing of this produced context (e.g., they might do predictions, optimizations or enrich the data). The Scorpio Broker makes this scenario possible by connecting producers and consumers through the standardized NGSI-LD format and protocol.

Scorpio uses the NGSI-LD API and information model to model entities with their properties and relationships, thus forming a property graph with the entities as the nodes. It allows finding information by discovering entities, following relationships and filtering according to properties, relationships, and related meta-information. For data not directly represented in NGSI-LD like video streams or 3D models, links can be added to the model that allows consumers to directly access this information. In this way, Scorpio can provide a graph-based index to a data lake.

Scorpio provides several interfaces for querying the stored data so easily analytics can be done on the stored data, like it can be used to predict the situation of an ecosystem. Example: In a huge building there can be several fire sensors, temperature sensors, and smoke sensors. In case of a false fire alarm, it can be verified by the collected fire data, temperature data and smoke data of a particular area.

Functionalities

Scorpio implements the NGSI-LD API and information model, which enables the following interfaces:

- Create, update, append and delete context information: Providing NGSI-LD Entities for the actual information sharing.
- Query context information, including filtering, geographic scoping, and paging.
- Historical tracking of entity data.
- Subscribe to changes in context information and receive asynchronous notifications.
- Register and discover sources of context information, which allows building distributed and federated deployments.

These interfaces are used in SALTED to enable data flow between SALTED components. For example, the entity enrichment component might subscribe to changes in terms of available entities in Scorpio. When a change happens (e.g., a new entity has been created), the entity enrichment component gets notified and receives the newly created entity. It can then perform entity enrichment functions and update the entity with the enriched information.

Federation

Scorpio can be deployed in a federated setup, either to increase scalability or to achieve some isolation between different organizations, while still allowing some cross-organizational NGSI-LD queries and data flow (see Figure 4).

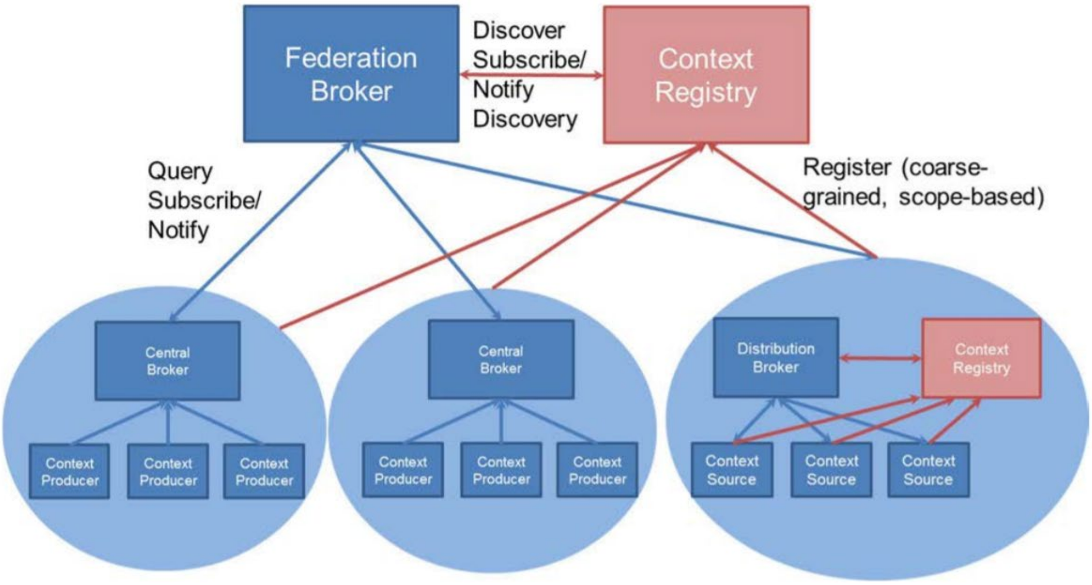


Figure 4: NGSI-LD Federated Setup

3.2.2 Control Broker

The Control Broker is the core component to support the functionalities envisioned in the Control Plane to support the configuration of DET components by applications. In this case, the solution followed consisted in the development of a tailored component following the principles stated in Section 3.1.2. The rationale behind implementing a dedicated module is to simplify the control process as much as possible, avoiding components with unnecessary additional functionalities, thus reducing the required computational resources as well. However, for the core functionalities, the Control Broker relies in well-known technologies, to refrain from reinventing the wheel. The component relies on the MQTT messaging protocol [3] to support such functionality through a lightweight messaging broker so-called MQTT Broker.

As aforementioned, Control Broker employs the PUB/SUB communication mechanism to implement the IoC pattern. The MQTT Broker is in charge of redistributing the messages throughout the clients subscribed to specific topics. Thus, anyone who is subscribed to a topic receives the message sent with that topic. In our case, the Control Broker relies on this feature to distinguish which component should address the request by the application. Besides, the one-to-many relation enabled by the broker will let applications to configure different instances of the same DET components at the same time, if required. However, most of the requests will be done against one particular component.

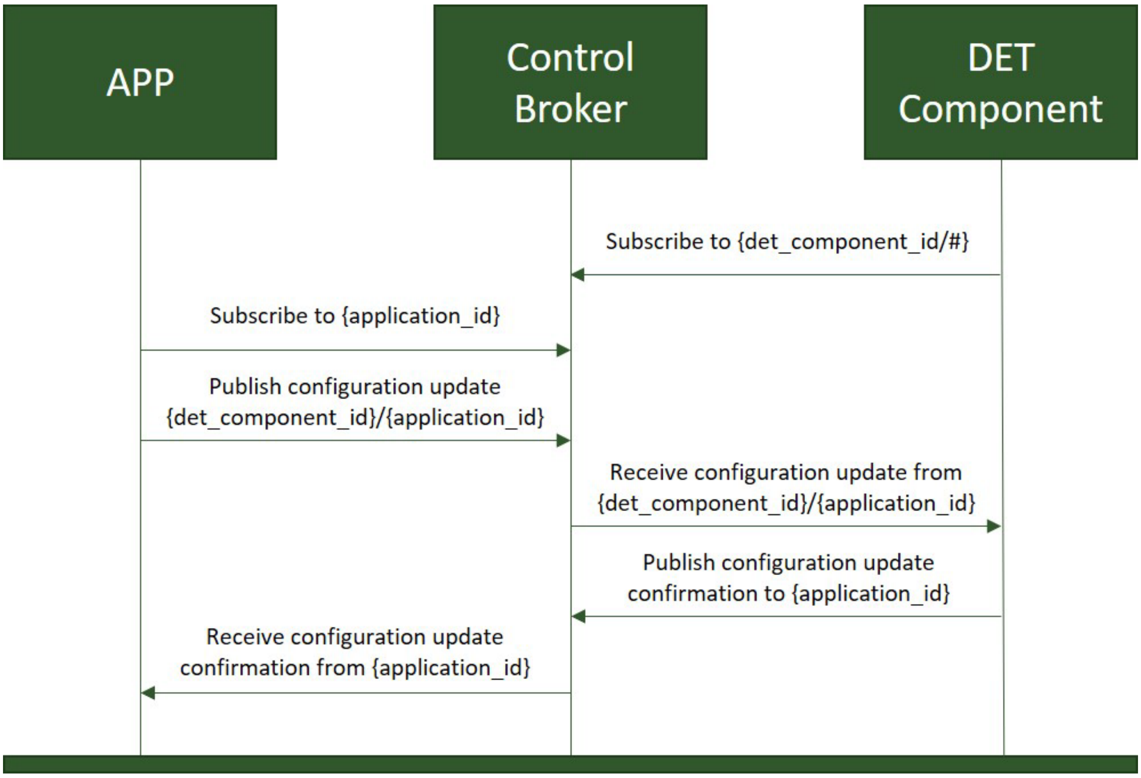

APP
Control Broker
DET Component
Subscribe to {det_component_id/#}
Subscribe to {application_id}
Publish configuration update {det_component_id}/{application_id}
Receive configuration update from {det_component_id}/{application_id}
Publish configuration update confirmation to {application_id}
Receive configuration update confirmation from {application_id}

Figure 5. Configuration update workflow for the Control Broker

The workflow followed by DET components and applications is shown in Figure 5, and described as follows:

1. DET components subscribe to the Control Broker to receive any configuration update requests. The topic consists in its own unique identifier along with a wildcard (i.e. {det_component_id}/#). This way, each DET component receives only their notifications, but from any application.
2. Similarly, an application updating the configuration of the component subscribes to its own identifier (i.e. {application_id}).
3. In case of a configuration update, the application publishes the configuration update with a topic consisting on the target DET component identifier and its own application identifier (i.e. {det_component_id}/{application_id}).
4. The specific DET component receives the configuration update request and publishes the response once the configuration updates are implemented. The topic to which the response is published is created with the application identifier (i.e. {application_id}), obtained from the message received.

All the content from the messages are expected to be sent in JSON format, although key/value pairs are specific for each of the DET Components as they will refer to the respective parametrization options that they support.

4 DET COMPONENTS

4.1 Data Discovery and Collection

4.1.1 Generic Purpose, Processing Flow and Sub components

The components aggregated under the terms data discovery and data collection serve as basis/source of the SALTED architecture as shown in Figure 6.

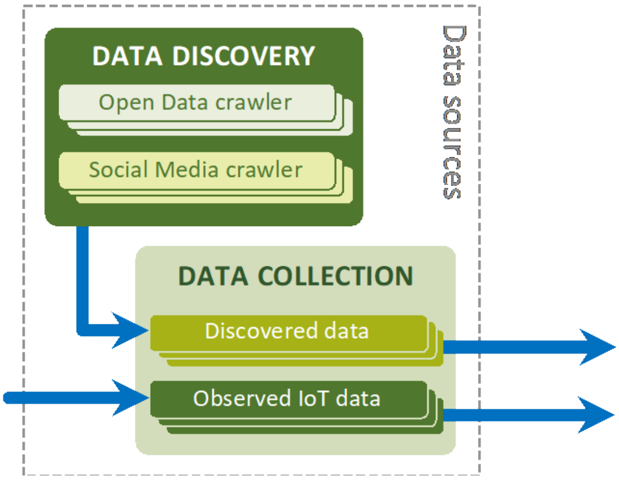


Figure 6 Data Discovery and Collection

The generic purpose or main objective of these components is at first the exploration and discovery of heterogeneous data sources such as IoT devices and sensors, web sources and social media, and open data. Data will be collected using different approaches, depending on the specification and features of the data sources. The output of these components needs to be fitting as input for the subsequent mapping to NGSI-LD.

Enabling a dynamic configuration of data collection (e.g., configuration of websites to crawl) prioritizes the clear decomposition into data discovery and data collection. At the same time, the collection components must clearly define a possibility of the parameterized request and document it for later applications. The goal is to introduce a certain dynamicity into the integration of new data into the SALTED pipeline.

From an organizational perspective within the SALTED project, the data discovery & data collection belong to the injection chain and are therefore headed and handled by the corresponding data owner individually. There is an agreement on the need to minimize the replication and redundancy of data and maximize the value at any given opportunity. The same individual responsibility is expected regarding the storage of the collected data. The data owners must comply with the applicable regulations and laws in each individual case.

The approach concerning the collection process can be specified as data source dependent. Therefore, different handling mechanisms are developed. For mainly structured, known IoT data, the collection process can be achieved through the implementation of interfaces for asynchronous and synchronous data collection from IoT sensors. Data available from Open Data portals also typically has formal structure both in terms of data modelling and access interfaces, however, it can have a wide variety from one portal to another even in catalogues referring to the same domains. For the part of the SALTED project, that will leverage the crawling of the web, the collection process is occupied with getting the information defining the scope of the later, to be crawled data. The following sections will give further insight into the different approaches within the project on how the data is discovered, collected, which pre-processing steps take place before the data can be mapped to NGSI-LD in the next step and what the intended request strategy is within the closed-loop approach.

4.1.2 DET Component Flavours

IoT Data Collector

The IoT Data Collector (IoTDC) is the component that supports the data collection and data discovery functionalities envisioned in the SALTED architecture for the IoT-related data (e.g., SmartSantander, Dublin and Madrid). It implements two interfaces, depending on whether the data sources are asynchronous or synchronous.

Asynchronous Data Collection

Asynchronous data collection refers to those data sources that provide a pub/sub service to access the information in (near) real-time once it is generated by the corresponding IoT device. This is the case for SmartSantander, which exposes an API that allows users to subscribe and receive notifications every time a new observation is registered by a sensor. Beyond the SmartSantander API, it also interacts with external platforms that can also provide asynchronous information, such as the “The Things Network” (TTN), which enables a MQTT broker to access data produced by LoRaWAN IoT devices from SmartSantander.

All in all, the IoT Data Collector subscribes to both the SmartSantander API and the SmartSantander TTN instance to gather the observations produced by any SmartSantander IoT Sensors. Nevertheless, other data sources can be easily added.

Synchronous Data Collection

Synchronous data collection refers to any data source (e.g., Open Data Portals) that provides static HTTP endpoints, through which (batch) data is normally gathered by periodic polling. E.g., in Madrid and Dublin, IoT related data sets are available as large CSV files.

IoTDC schedules periodic HTTP requests to a set of identified CKAN Open Data Portals (ODP) (e.g., Santander, Vitoria, Barcelona, Bilbao or Valencia ODP) providing relevant IoT data for SALTED. Data collection takes advantage of the RESTful API provided by CKANs, although the IoTDC can gather data from any HTTP endpoint. Once the data is gathered, it is split into smaller chunks or observations, and forwarded to the next component in the DET pipeline.

Note that in the course of the project, we might develop other synchronous data collection components. For example, GIS systems often provide their data in the form of shapefiles with embedded database tables. Those tables can easily be transformed into CSV files.

Social media Data Collector

Social media data collection refers to collecting various content such as blogs, posts, likes, followers, clicks, shares (reposts and retweets), comments, or engagement rates, from publicly available social media platforms by different users.

IMT is integrating the social media crawling into the SALTED project. To gather relevant information (conforming to general privacy and limitation rules), a programmable crawler of data can be operated within the project in order to retrieve the desired data. The crawlers can be launched, and specialized data can be gathered according to the limits and policies of the different social media networks. We divide these crawlers based on different data sources as follows:

Twitter crawler

Twitter is an online social media site that allows users to send and read short messages called "tweets" in real-time. Its popularity as a fast information dissemination platform has led to applications in various domains (e.g., business, disaster recovery, intelligent transportation, smart cities, etc.).

The Twitter data crawler uses Twitter's public APIs[1] for retrieving data from this platform. These APIs have limitations. With the "elevated" access, it is possible to retrieve 2 million Tweets per month. Moreover, Tweepy[2], which is one of the Python 3 libraries that can be used to crawl, is used in the Twitter crawler.

For crawling data from Twitter, we use the following approaches:

- Hashtag/keyword-based
- Specific public accounts

Instagram crawler

Instagram is a famous social media platform for generating content in the form of a post, story, or reel that contains text, image, or video. Retrieving public data in this platform can be used for different purposes in both industry and academic fields.

Other sources

Other social media sources can be considered for the collection of data, for example, Facebook offers a secure HTTP-based API. It allows developers to query public posts of specific users or organizations via authenticated HTTP calls. Moreover, Reddit offers APIs for programmatic control of virtually every function a user can perform on the site.

Web Crawler

The web crawler integrates the crawling of the web into the SALTED project. Crawlers (or spiders) are developed that "crawl" websites in order to make their content available for further processing services. In addition, meta-search engines are developed which use an automated query process on already implemented search engines e. g. Google or Bing. As already mentioned above, a more conservative approach is needed when facing unlimited data on the web, because it's not possible to crawl all information available. Therefore, the components handling the discovery and collection are focusing on setting the scope with a specific use case in mind. One use case, further explained in the elaborations on use cases in chapter 5, will generate a compliance map for public organizations offering compliance scores for different goals defined within public agendas, e. g. the sustainable development goals. Therefore, the discovery and collection only handle the detection of companies of interest, and their

[1] https://developer.twitter.com/en/docs/twitter-api

[2] http://docs.tweepy.org/en/latest/getting_started.html

corresponding URLs. This can be used for the initial creation of an entity representing the company of interest within the graph later. Any crawling actions or generation of additional information can be seen as enrichment steps later and are therefore explained in the corresponding chapter.

4.2 NGSI-LD Mapping

NGSI-LD mapping defines the process of transforming heterogeneous data into a standard, well known, easy to manipulate data format, such as NGSI-LD. This component takes the data collected in the previous stage (Data Discovery and Collection) as its input, and provides NGSI-LD compliant data as its output. The input, due to its heterogeneous nature, can be represented using several different data formatting standards such as CSV, JSON, or XML. Moreover, both the names of the properties and the values can be highly different from one another, as a result of the heterogeneous data sources and their internal policies, language, units, and several other factors. Thus, the main motivation behind this component is to unify all data kinds into a single format, NGSI-LD, which can then be further processed by the rest of the components in the pipeline.

We have identified three approaches that fulfil this purpose, which grow in complexity as their flexibility and scalability increase.

- The **manual mapping** (Figure 6) consists of a program or script that directly maps a specific kind of input data to NGSI-LD. This script requires full knowledge of the input data since its inner workings are dependent on elements such as the input format or the specific words used within the data. The main drawback of this approach is the need for a different mapper for every data source and data type, meaning that the number of scripts required can easily get too large to handle. On the other hand, its advantages include a simpler and more direct implementation and the high accuracy and low loss of data due to having a dedicated mapper for every input.

Figure 7. Manual mapping data flow

- The **template-based mapping** (Figure 7) consists on the use of a predefined set of templates depending on the data type and the formats used by the data sources. These templates are filled with the information extracted from the different data sources, generating NGSI-LD compliant output. Similarly, to the manual mapping approach, the drawbacks are, once again, the required previous knowledge of the data source and the need for a different template for every output data type. However, this methodology reports several advantages as a better reusability and a more efficient implementation. The program executing the mapping is generic and templates are only generated once a new source is handled, which means that no modification of source code is required, making this approach friendlier and more automatic than manual mapping.

Figure 8. Template-based mapping data flow

- The **AI-based mapping** (Figure 8) uses Machine Learning techniques to implement the mapping in an automated way. It can be split into three phases: type identification, template selection and transformation. In the type identification step, an AI model has been trained to classify the text (bunch of words) representing the input data into multiple categories, corresponding to the different output data models. Once they have been identified, the corresponding template is selected from the existing templates' pool (these can be reused from the previous approach). In the third and final phase, the input data is matched with the template automatically, resulting in the NGSI-LD output data. This approach has significant advantages in terms of scalability since once trained the translation process from any input data model to the generic NGSI-LD is almost automatic. The main drawback lies in its complexity, specially the fact that the AI models must be properly trained and then re-trained after new kinds of input data enter the pipeline.

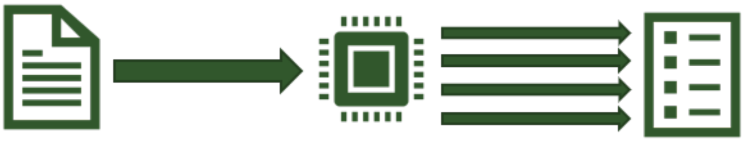


Figure 9. AI-based mapping data flow

4.2.1 DET Component Flavours

IoT Data Mapper

The IoT Data Mapper provides the NGSI-LD mapping functionality to transform heterogeneous data into the NGSI-LD flavor of the Smart Data Models. This module instance combines two of the aforementioned approaches, the template-based mapping and AI-based mapping.

It uses a Nearest Neighbor approach to individually map every field of the input document into its corresponding NSGI-LD property in the Smart Data Models. This mapping is based on the words used within that field. Once the NGSI-LD properties are identified, it uses a JMESpath template in order to transform the input document to the pertinent Smart Data Model. These templates work in tandem with a Python class that includes several custom functions to interact with the templates, allowing us to modify the output NGSI-LD document in a more complex manner. For instance, we are able to do type conversion within the document, change the date format, or generate unique identifiers.

However, the data received may not always be known. In order to deal with unknown data, we have implemented an AI-aided type identification module with the Tensorflow and Keras packages for Python. Firstly, we train a model with well-known SmartSantander IoT data and then this model is used to predict the type of the input heterogeneous data. After the type of input data has been identified, we send it to the Nearest Neighbor mapper along with the newly identified type, which then allows us to select the corresponding JMESpath template as mentioned before. The result is an automatically mapped NGSI-LD entity compliant with its corresponding Smart Data Model.

Social media NGSI-LD Mapping

The social media data mapper, with the goal of transforming retrieved raw data from different social media platforms, into the NGSI-LD architecture, uses smart data models.

For the SALTED project, a social media smart data model published by FIWARE[3] will be used and we will contribute that within the FIWARE GitHub repository.

Table 1 shows the entity types considered for social media data.

Table 1. Entity types for social media

Entity Types	User/Profile	associated with the description of a user of Social Media applications like Instagram/Twitter/Facebook etc.
	Post	associated with the description of a post (or a tweet in Twitter) created by a social media user
	Analysis	associated with the process of analysis of Social Media applications' posts
	Collection	associated with the process of collection of Social Media posts based on different subject (like a specific subject, hashtag, etc)
	RefLocation	associated with the description of a generic SM Reference Location

A visual representation of the social media smart data model is presented the appendix in Figure 17.

Web Data Mapping

The mapping procedure for entities created from crawled web data also differs from the mapping of defined sensor data sets from the IoT environment. The main approach here is an iterative one. Mapping occurs not once, but again, with every enrichment step, that provides a new attribute for an entity or even new linked entities.

Example, related to the Agenda Analytics use case: In the beginning, an initial mapping of the entities representing the “companies of interest” will take place. Two entity types will be used. The company as legal units will be mapped to the entity type Organization, which is specified within the schema.org definitions[4]. The aim within the project is to contribute that as smart data model within the FIWARE GitHub repository[5]. The geo-reference, where the legal unit is located at, can be represented by the entity type PointOfInterest, already defined within the smart data models[6]. Whether this generates added value remains to be tested. If it rather leads to an unnecessary overhead, since the geo data can also be mapped by the previously mentioned entity, then we will abstain from it. Within the enrichment steps, e. g. entities of the type

[3] https://github.com/smart-data-models/dataModel.SocialMedia
[4] https://schema.org/Organization
[5] https://github.com/smart-data-models
[6] https://github.com/smart-data-models/dataModel.PointOfInterest

KeyPerformanceIndicator[7] that hold information about calculated compliance scores, will be linked through relationships.

Regarding the technical approach of the mapping, firstly the usage of templates is targeted. Since the services leveraged for each enrichment step provide clear results, a mapping through templates is sufficient and no bias needs to be introduced through AI-based engines.

4.3 Data Curation

The aim of the Data Curation module is to check that the data received, exported by the mappers, is valuable and adequate to be stored in the SALTED NGSI-LD context broker. Either by tagging them with additional extra information (metadata) or by directly rejecting them in case non-valid data is detected, this module guarantees that users and applications gathering information from SALTED will have not only high-quality data, but a better understanding of its meaning.

The curation process can be considered the first step in the entity enrichment that SALTED envisaged. However, it is important to note that within the SALTED architecture, the Data Curation module is located after the NSGI-LD mapper and just before the data is fully accessible to external applications. It acts on the one hand as a kind of firewall that guarantees that only good and coherent data is stored, and on the other ensures that the information to be stored in this Broker has the highest possible informative quality.

As part of this workflow, the Data Curation Module not only accesses the Broker to create and update the necessary entities, but also requests information already stored in order to be able to perform its different processes. Regarding the data format at the output of the Data Curation Module, the original NGSI-LD entity provided by the NSGI-LD mapper is complemented with additional entities corresponding to the new properties generated, as can be seen Figure 9.

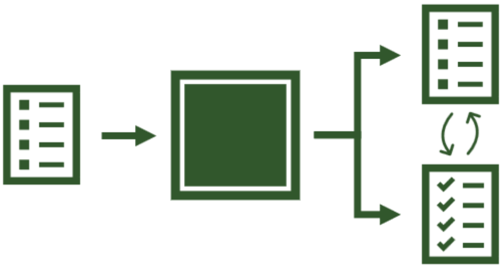


Figure 10. Data curation flow

To this purpose, different information processing functions are performed to obtain several properties that offer this quality improvement. The Data Curation Module is composed of several submodules that perform the different processes to obtain the new properties. The functions to be used to increase the quality of the data will be specific and dependent on the type of data source, including the possibility of reusing these functions among similar data sources. The different submodules provide functionalities such as detecting data that does not

[7] https://github.com/smart-data-models/dataModel.KeyPerformanceIndicator

fit with the rest, predicting missing data values or even calculating intrinsic characteristics of the data itself, such as frequency of arrivals, accuracy or availability.

4.3.1 DET Component Flavours

IoT Data Curator

The IoT Data Curator focuses on the information collected in real time by the sensors deployed around Santander (mainly linked SmartSantander project) as its Data Source. This implies that data curation is performed on NGSI-LD formatted numerical measurements of the environment, such as temperature or sound pressure level, which are obtained in SmartSantander proprietary format in the data collection phase but can be easily extended to other data formats. These two characteristics (real time and numerical measurement) significantly affect the structure of the Data Curation module, as well as the type of processes and analysis to be performed on the information to obtain the new quality properties.

The main functionality provided by the component is the novelty detection. It is a type of anomaly detection that only classifies as outlier or not the new observation received regarding a prepared dataset. This set is the train dataset of the machine learning algorithm that performs this classification and contains several dimensions such as location and timestamp, in addition to the numerical value itself.

Then the second step in the curation process, with is closely linked to data enrichment, is performed. It includes the addition of measurement quality properties like frequency, completeness, precision and/or accuracy.

Those enrichment functions or submodules can be easily executed at any time during the data lifecycle, not only during the curation phase. They will add or modify the measurement quality properties based on the currently available information in the SALTED NGSI-LD context broker. Thus, the description of these capabilities is moved to the corresponding Entity Enrichment section (Section 4.5).

Social media Data Curation

Data-driven analytics of social media data has become a vital asset for organizations to further improve their products and services. Since the raw social media data usually are noisy, performing accurate analysis of these data is requires curation before fed into analytics pipelines. This curation process transforms the raw data into contextualized data and knowledge. Generally, the following curations are proposed for social media data before sending data to the NGSI-LD broker:

Transforming emojis and emoticons

Emojis and emoticons convey emotional expression in a text message which are usually used by users on every social media platform. Removing the emojis/emoticons from the text for text analysis might not be a good decision since removing them may affect further enrichment quality and valuable information will be lost. They can give information about a text such as feeling expression, especially in Sentiment Analysis. A better approach is to convert emoji to word format so that it preserves the emoji information.

As an example of this approach, the emojis will be transformed into textual equivalent:

"I ♡ this flower. Thank you :)" → "I love this flower. Thank you with smiley face"

Removing URLs, emails, mentions, duplicate whitespaces, and punctuation from text

Mentioned items don't contain any valuable information, and removing them from text, helps improve performance on textual analysis.

Substitution of contractions

Using contractions is very popular in social media data, especially in the English language; e.g.: 'I'm'→'I am'.

Replacing the contractions for their actual words based on a list of contractions from Wikipedia is a useful data curation step.

Spell correction

In some natural language applications such as information retrieval, it's useful to correct spelling errors. For example, 'infromation' is normalized to 'information'.

Web Data Curation

The unstructured web data with unlimited scope imposes again additional challenges on the curation of data. Therefore, the curation will be implemented at each step of the pipeline handling web data. An example is the integration of new companies for the Agenda Analytics use case mentioned before (otherwise see chapter 5). The steps introduced for curating the data, need to make sure, that no company will be added as a duplicate to an already integrated one, e. g. when a name change occurs. Even with a unique number of companies integrated, the discovered and used domains (URLs) for each company need to be validated to ensure the match between company and URL. Finally, the crawled, processed, and incorporated data for each company needs to be transparent. The end-user of the NGSI-LD graph should be able, to trace, what data was used and see which actuality the information has, that the user sees in the graph.

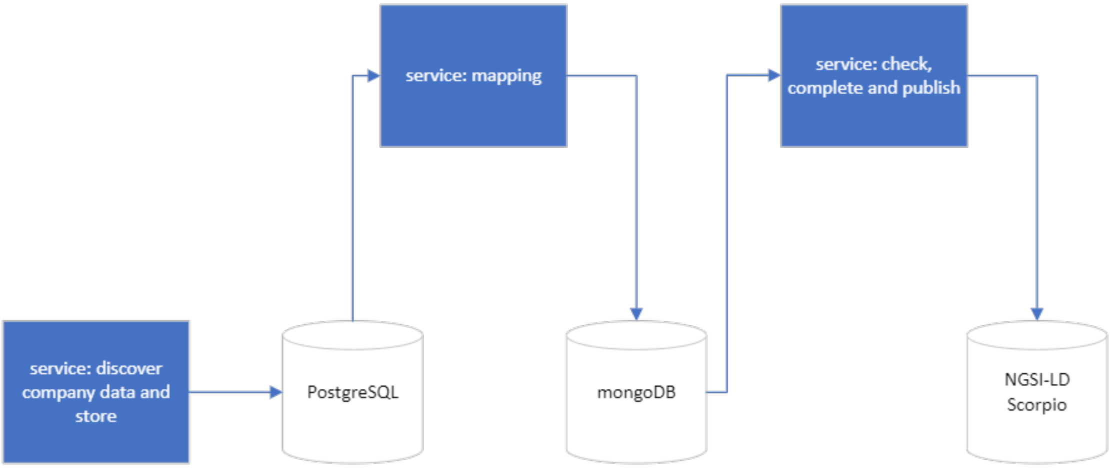

service: mapping
service: check, complete and publish
service: discover company data and store
PostgreSQL
mongoDB
NGSI-LD Scorpio

Figure 11. Company data collection workflow

One exemplary workflow for the collection of the initial company data (legal name and domain name to be used) addressing those issues is shown in Figure 10. As illustrated above, discovered company data in the raw form will be stored in a PostgreSQL database. In addition, storage of the associated metadata takes place (When was a company initially discovered? How often and when was the company again seen by the discovery service? …). This is implemented with help of the integrated quality assurance possible within e. g. PostgreSQL. One example is the possible implementation of data integrity with constraint functions, additionally to all aspects regarding structured databases. This approach is supposed to ensure the logging of how the data was discovered, for traceability and prove of origin.

In the next step, where mapping to the NGSI-LD format takes place, the metadata can be aggregated, so only customer-valuable information and a reference to the raw data will be included into the entity representation. Because of the JSON-like formatting the results will be stored in a MongoDB.

Before pushing those entities to the graph, content gets checked and completed to ensure high quality of the data represented within the graph.

4.4 Entity Linking

Entity linking aims to reconcile data records collected from heterogenous data sources and link them to the same real-world object. This is important because different data sources might describe the same real-world object differently (e.g., the same building might be referred to with different names).

In SALTED, we link NGSI-LD entities stored in an NGSI-LD context broker (Scorpio) through the Entity Linking component (see Figure 11). This has the advantage that the meta data-model and query interface is standardized trough the NGSI-LD specification. In the entity linking component, we define two sets of linkers:

(1) **Basic Entity Linkers** that discover sameAs links (i.e., equivalent, direct 1-1 matches) between two entities. E.g., these linkers might discover that a building entity with the street name “Kurfürstenanlage 36” is actually equivalent to one with “Kurfuerstenanlage 36”.

(2) **Enriched Entity Linkers** that further discover links/relationship of other meanings. E.g., e.g., a contains relationship (e.g., between an object/sensor and the building where it is contained) can be done based on inheritance, or location based.

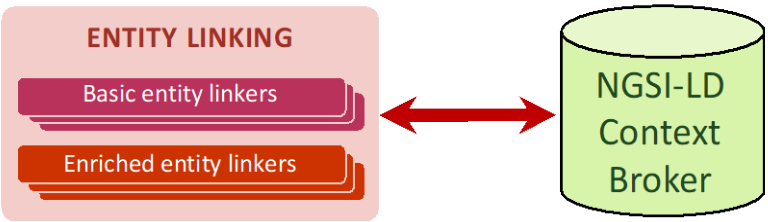


Figure 12 Entity Linking Component

The number of entities in the NGSI-LD Context Broker is expected to be large. As such, entity matching is facing a scalability problem, as potentially all combinations/pairs of entities need to be evaluated for sameAs or other relationships. This complexity is O(N^2), where N is the number of entities in the NGSI-LD broker. To deal with this problem, SALTED will develop a three-step pipeline: (1) A component will filter the entities based on their Smart Data Model type and the available properties in the schema. For the remaining entities, we create all possible combinations; (2) As these might still be too many combinations to perform entity linking in an adequate time-frame, we then block out entity combinations with low likelihood to be matches in a Blocking step; (3) Last, for the remaining candidate entity combinations, we perform a more compute intensive matching/linking step and then push the results back to the NGSI-LD broker.

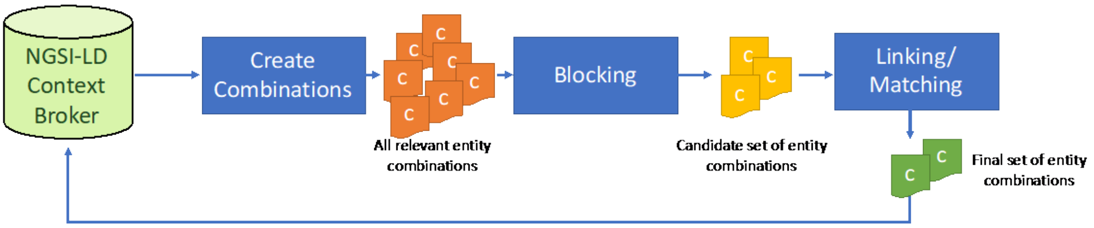

NGSI-LD Context Broker
Create Combinations
C
All relevant entity combinations
Blocking
Candidate set of entity combinations
Linking/ Matching
Final set of entity combinations

Figure 13 Entity Linking Details

4.4.1 DET Component Flavours

TrioNet

TrioNet is a data integration platform, with a set of matching methods for ontology, schema and entity matching (see Figure 13). TrioNet is developed outside of SALTED. It applies a combination of weak-supervision and active learning to facilitate these matching tasks through a human-in-the-loop approach. In SALTED, TrioNet will be made available as a service to enable the execution of matching tasks.

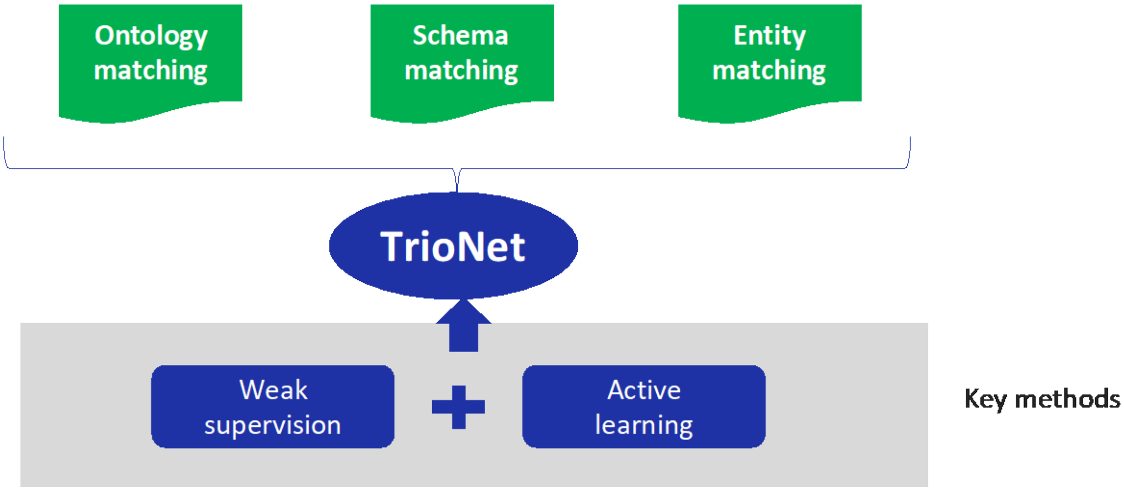


Figure 14 TrioNet: Build Human-in-the-loop AI pipelines to enable automated matching

4.4.2 Examples of linking between different IoT Data Sets

Smart City data set offer the possibility to study additional linking requirements. As an initial example, we may consider the traffic intensity data of several cities (e.g., Madrid, Santander and Dublin). Their linking feature will be TIME, and more specifically a time period. The available data about intensity data (aggregated or specialized for a set of sensors) in the different cities and for a time interval (e.g., from 7 a.m. to 10 a.m.) could be collected and aligned in terms of how frequently this data is collected (from instance in Madrid every 15 minute, while in Santander every 5 minutes). The linking could comprise functionalities to homogenise and recompose the collection time period in order to produce for each data set comparable data (in this case, for the Santander data, an average of traffic intensity every 15 minutes will be calculated). The goal here is identify how the traffic intensity is evolving over a certain period of time in the “linked” cities”.

Another example of linkage is for data sets generated in the same geographical location, i.e., linking or aggregating data with respect to a SPACE. In this case, space can refer to a single location. For example, the traffic intensity data of sensors close to a pollution station (e.g., within a distance minor that 500 meters) can be "linked" together in order to study the contribution of the evolution of vehicle traffic over the progression of pollution. Location can also refer to multiple positions. For example, the previous case can be extended by considering all the pollution station in a city and the corresponding traffic intensity sensors within a distance of 500 meters. The location and the related linking of data can be further extended by considering all the pollution stations in Madrid vs those of Santander to see patterns in the two cities. Location can also have a different meaning, the one of a well-defined space, e.g., the central zone of Madrid, or a particular quarter, or even a path between two points. So, data that are generated in the specified "regions" can be aggregated and presented to requesting applications.

Another type of linkers are those that will select from various sources some events or notification related to locations. For instance, many cities have a service (and related info published on a web site) related to authorized events in the city. The events can be of different types: protests, manifestations and events, incidents. They could be collected and represented as NGSI-LD formatted information. These data could be linked to the other data sets to lay the basis for a contextualized data collection. This contextualization could also be considered as a form of enrichment in which different type of data are aggregated to represent the situation. Enrichment functionalities could be triggered or executed when anomalies in the expected patterns are detected. As an example, if an anomalous traffic is detected in a region in Madrid, the association between region and events could be the basis for determining that a specific manifestation is having a great impact on the traffic evolution. Enrichment algorithms could be using the event data sets to simulate or predict some effects on traffic (or other phenomena) on the city.

4.5 Entity Enrichment

Data enrichment refers to the process of appending or otherwise enhancing collected data with relevant context obtained from additional sources.

Data enrichment is divided into the following subsections:

- Existing entity enrichment enhancement modules
- Situation-aware entity inference modules

For the SALTED project, the data enrichment is then the capability of augmenting and extending the information content of existing NGSI-LD formatted data by deriving additional information from or relationships between existing data. This component in the SALTED architecture is shown in Figure 14.

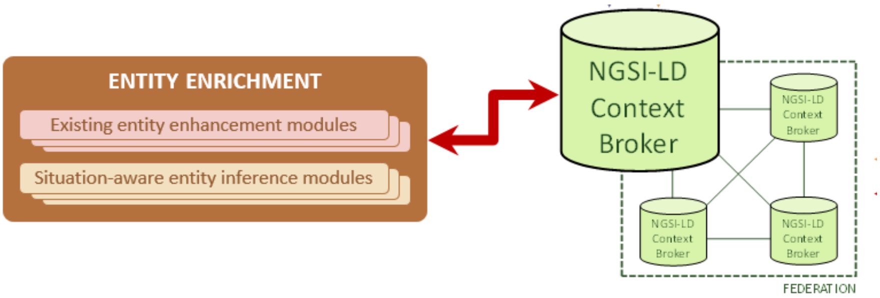


Figure 15. Entity enrichment component.

There are several options for data enrichment. The construction of knowledge graphs and the identification of situation-related dependencies are two major options for the enrichment of SALTED datasets. Other forms of enrichment are the applications of well-known AI algorithms on the available datasets. The goal is to derive new information or simply to visualize the available information. An example of this is the Weka Framework. It makes available many different AI-based tools that can be directly applied to data sets that are formatted accordingly to the Weka rules.

As seen, the creation of a knowledge graph is a path to data enrichment. The creation of Knowledge graphs starting from available data is an extremely important and valuable result. As described in [4], the construction of knowledge graphs is a process that can have increasing levels of difficulty and a different degree of “generality”. The construction of basic knowledge graphs (properties graphs) can exploit the definition of the data in terms of elements and their relationships. Labeling nodes and relationships could increase the information content and provide users with a clearer structuring of their datasets. A second step is to consider some taxonomies and to “organize” the datasets accordingly. This will provide even more structure to data (a hierarchy) and will make evident more relationships and “the process” behind data. The third step is to analyze the data and to reason about their definition and relationships by means of ontologies. In this case, a more consistent model for data representation can be built. Situation awareness (i.e., contextualized information) can be built by the application of ontologies and reasoning mechanisms.

As a rule of thumb, the generality of the method (from properties to ontologies) is decreasing. Properties can be derived and analyzed in a general fashion, while taxonomies and even more ontologies, over impose a view of the context of the dataset. This means that data are to be curated and prepared to have in mind the possible situations. This makes the possible solutions depend on the specific problem domain.

There are some products/solutions that can exploit the power of JSON-LD representation for building knowledge graphs (e.g., Allegro graph). This is encouraging also for SALTED because the NGSI-LD representation) emphasizing elements and their relationships could leverage the possibility of creating in a semi-automatic way basic knowledge graphs. Additional help could be the ability to label elements and relationships in order to better identify the relationships and the nodes that can be considered in a property graph). The Labeling of data could be seen as an initial step in the enrichment. In the context of SALTED, it could be part of the “curation” process (in which data are prepared and normalized for further usage; or in the enrichment part in which the SALTED system offers the possibility to users of labeling the data in order to create the basis for initial construction for knowledge graphs.

Being general-purpose in the "enrichment" phase may lead to weak and scarcely results. For this reason, the definition of enrichment functions within SALTED will be driven by the specific use cases and the related problem domain solutions. However, during the course of the project, some generalizations of the process towards the construction of a basic general-purpose knowledge graph will be attempted in order to validate the value and the problems related to a context-free enrichment. This needs some time for creating the right tools and the validation of relevant results. For instance, the NGSI-LD could be a promising format for property graph creation, but this possibility should be tried against the available solutions and their capability to fully exploit NGSI-LD entity and relation representation.

A similar approach will be pursued for general-purpose algorithms (in the WEKA style). SALTED will first focus on the specialized enrichment tools and techniques needed in the specific problem domains. In the meantime, the project will experiment in order to identify the possibility to provide generically applicable AI algorithms to the NGSI-LD formatted data.

As a final remark to this introduction, there is a strong need for situation-aware representation of data. Situations and contexts are intertwined and peculiar to the problem domain at hand (or the linked data considered). Focusing on the construction of capabilities supporting specific problem domains may be more fruitful than pursuing immediate generality in techniques and tools. As an example, considering IoT data related to traffic and linking them to "road construction" or accident data sets is very situation-related and can bring immediate and practical results.

4.5.1 DET Component Flavours

Social media Data Enrichment

Sentiment analysis

Sentiment analysis is an application of natural language processing (NLP) that reveals the emotional in textual data to help businesses to understand customer needs by analyzing the sentiment in customer feedback. A machine-learning-based sentiment analysis software can examine the positive or negative or neutral sentiment about the brand, product or any events. Specially in social medias, many people express their feeling through publishing post or comments about different topics so sentiment analysis on social media data can be a very useful enrichment component for SALTED.

Some of sentiment analysis use cases are:

- Social media monitoring and brand management
- Customer service response
- Product analysis
- Event analysis

With recently published language models such as BERT[8], the accuracy of predicting the true sentiment of text has been improved not only in English but also in other languages.

Named entity recognition (NER)

[8] https://arxiv.org/abs/1810.04805

Named entity recognition (NER) is a natural language processing (NLP) technique for identifying and classifying named entities within the text automatically in various entity groups. Such entities can be names of people, organizations, locations, times, quantities, monetary values, percentages, that can give key information to understand what a text is about.

By using machine learning algorithms and Natural Language Processing (NLP), firstly entities will be recognized and then categorized into the pre-defined categories. This service can get a text as an input and provide the results as an output.

Topic Analysis

Topic analysis/detection is a machine learning technique that uses natural language processing with to find patterns within text for helping data-driven decisions and understanding large collections of text's topic.

The two most common approaches for topic analysis with machine learning are NLP topic modeling and NLP topic classification.

Topic analysis can be applied at different levels of scope:

- Document-level: the topic model obtains the different topics from within a complete text.
- Sentence-level: the topic model obtains the topic of a single sentence.
- Sub-sentence level: the topic model obtains the topic of sub-expressions from within a sentence.

Word cloud

A word cloud is a simple yet powerful visual representation object for text processing, which shows the most frequent word within the text. This service can be used for the following cases:

- Top Hashtags on Social Media (Instagram, Twitter) about various events and incidents that are spreading in such platform.
- Hot Topics In Social Medias by analyzing the keywords in the headlines of news articles and posts to extract the top hot topics.

Compliance Engine

The compliance engine uses text as input data (e.g. from a Web Crawler component) and computes its matching score regarding to a predefined agenda, like the United Nations defined *Sustainable Development Goals*, SDGs. The results are used within the Agenda Analytics use case. This service is deployed as docker container. A more detailed explanation regarding the Natural Language Processing (NLP) methods used is given in the following:

The main purpose of the NLP compliance service is a coverage and similarity analysis of a freely chosen text and a predefined reference text. The reference text hereby addresses all relevant aspects and issues of a certain compliance task, for example SDG or ESG goals, but any text serving as a reference can be chosen.

The analysis is based on high-dimensional vector-space representations of both the reference and the test text which were divided into manageable slices of a maximum of 512 tokens. The usage of state-of-the-art transformer models based on the attention mechanism in combination with large language models allows for the generation of meaningful representations that incorporate a good deal of semantic information of the corpora. Once the representations are generated for both corpora, cosine similarity between both sets is computed and aggregated accordingly. The higher the similarity the better is the resemblance between a test and a compliance text slice. In practice a cutoff, for similarities to be considered as relevant, is introduced and only the topmost results are taken. By this procedure a filtering of the most important parts of the test text with respect to a reference corpus is done and can be visualized by a sunburst or network graph. Additionally, a determination of the fraction of the reference corpus addressed by the test text yielding a coverage measure is possible.

It was found, that reasonably good results can be obtained on average with high-level language models but some tasks require even more precision than currently available. For further improvements it is therefore considered to follow a multimodal approach including structural information obtained from the text and support of the NLP engine by lexical information. The latter aspect becomes important when dealing with texts using a highly domain-specific vocabulary.

Web Crawler

The web crawler itself is not an enrichment service but can be used by enrichment services to get raw data for further processing. An example is the compilation of company data for the compliance engine, that calculates the compliance scores for companies with respect to a predefined agenda, e. g. the SDGs.

The service for the crawling of web data is using StormCrawler[9], which is an open-source collection of resources for building low-latency, scalable web crawlers on Apache Storm. The additional setup of all implemented components within docker containers makes the service platform independent and portable. The service e. g. uses the seed URL and the desired depth of the crawling process as input and fetches all websites it comes across, recognizing restrictions regarding the robots.txt. For further processing the fetched data is stored within a MongoDB. Within the parameters of the service, it is also possible to specify the type of content to be fetched (e.g., text and/or pdf).

To implement a first step of pre-processing an additional service (as docker container) can be used on the text results, to heighten the quality of the crawled data. The to be applied steps can be configured through the parameters given to the service. Examples are the elimination of stop words, the conversion to lowercase.

For PDF documents a dedicated service (as docker container) is provided, that transforms these documents into plain text data.

Search Engine

The search engine service can be used, to leverage the Google or Bing search with a user set query. Also set up as a service with a defined API and corresponding parameters it can provide additional insights into the desired topics based on the extensive web directories from Google and Bing. Within the Agenda Analytics use case it is used to generate scores regarding the relevance of compliance as a topic with respect to the web presence of companies. The results are later used within the visual representation. This service can be deployed as docker container as well.

[9] http://stormcrawler.net/getting-started/

Numerical Measurements Enricher

Either using as input just generated valid NSGI-LD format from the mapper or NSGI-LD entities available at the NSGI-LD context broker, the IoT Data enrichment module will initially work over numerical measurements.

In this sense, the enrichment process, which is based on artificial intelligent or machine learning algorithms, will generate new measurement quality properties such as frequency, completeness, precision and/or accuracy are calculated. The first one, frequency, refers to the reporting frequency of the sensors, i.e., how often they report a measurement. The completeness parameter reflects the success rate of information transmission from the sensor to the corresponding SALTED DET in a sliding time window. Lastly, the precision and accuracy. The precision of a measurement shows how close the value is to the other reference measurements, and the accuracy indicates how close the measured value is to the ground truth. Both provide information about the measurement quality of the device.

Besides, it will perform the prediction of the values of the missed notifications. In this way, all the gaps of losses and errors in transmission are filled in, obtaining a complete record with real notifications and so-called synthetic notifications.

5 USE CASE VIEW/INSTANTIATION

5.1 Smart City

"Smart City is a collective term for holistic development concepts that aim to make cities more efficient, technologically advanced, greener and socially inclusive. These concepts include technical, economic and social innovations."[10]

An increasing amount of data is generated today by city residents and their infrastructure. This means that the conclusions that can be drawn from this data increases significant as well and could contribute to many scenarios when they would be enabled through some architecture.

Situation-aware smart city applications within the SALTED project are trying to fill that void, that is present today, where the data is theoretically available, but not used to support interested parties. The final goal is, that the applications will benefit by data from various data sources linked to objects into an urban environment, enabled through the DET of SALTED. Within the SALTED project, one of the focused areas is exactly this smart city scenario.

An application is to be implemented that supports decisions making for the transformation processes in the use of space. Specifically, front-end and back-end functions are to be developed that provide context- and location-related relevant information to various parties involved in urban development (cities, companies and people / civil society).

The further discovery of ideas for use cases could be subordinated to the statement "Cities have to learn to speak" - not only in the language of institutions, or programmers and developers, but also in the language of the conventional city residents. To give one creative example: In the future, everyone should, when he comes to an unknown city, invite this city (its bot) into his messaging and let it guide him through the city, explaining to him the places of places of interest will be explained to him. In principle, it would even be possible to interact with urban infrastructure. The implementation of ideas like that need to be validated and evaluated from the viewing point of data restrictions, architecture restriction and the interest of the end user.

Another important overarching theme is the discoverability of the EDP data. The SALTED Project itself feeds into the EDP, so it is all the more important to develop applications in this project that can access the huge existing data base. The integration of metadata query options in data portals and in the broker as a "meta-application" should be considered and included.

The first use case of the SALTED pipeline, that is validated and has potential users, is Agenda Analytics. As the name already reveals, this has the matching of given agendas to the activities of companies and public administration as its content. The obtained scores can be used for detailed reporting a visualization, as can be seen in the next two illustrations:

[10] https://de.wikipedia.org/wiki/Smart_City

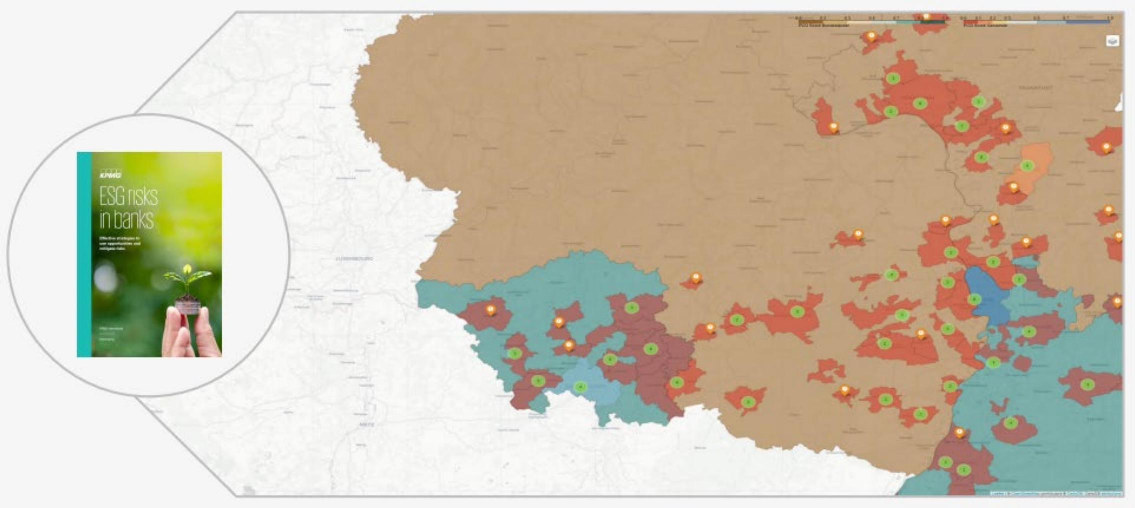
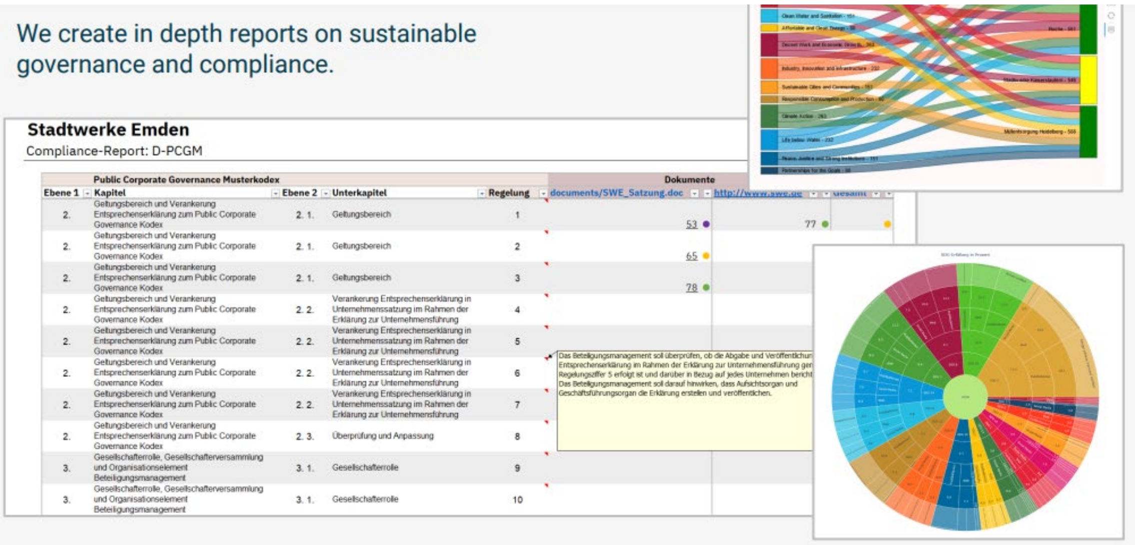


Figure 16: Agenda Analytics map



We create in depth reports on sustainable governance and compliance.

Stadtwerke Emden

Compliance-Report: D-PCGM

	Public Corporate Governance Musterkodex				Dokumente		
Ebene 1	Kapitel	Ebene 2	Unterkapitel	Regelung	documents/SWE_Satzung.doc	http://www.swe.de	Gesamt
2.	Geltungsbereich und Verankerung Entsprechenserklärung zum Public Corporate Governance Kodex	2. 1.	Geltungsbereich	1	53 ●	77 ●	●
2.	Geltungsbereich und Verankerung Entsprechenserklärung zum Public Corporate Governance Kodex	2. 1.	Geltungsbereich	2	65 ●		
2.	Geltungsbereich und Verankerung Entsprechenserklärung zum Public Corporate Governance Kodex	2. 1.	Geltungsbereich	3	78 ●		
2.	Geltungsbereich und Verankerung Entsprechenserklärung zum Public Corporate Governance Kodex	2. 2.	Verankerung Entsprechenserklärung in Unternehmenssatzung im Rahmen der Erklärung zur Unternehmensführung	4			
2.	Geltungsbereich und Verankerung Entsprechenserklärung zum Public Corporate Governance Kodex	2. 2.	Verankerung Entsprechenserklärung in Unternehmenssatzung im Rahmen der Erklärung zur Unternehmensführung	5			
2.	Geltungsbereich und Verankerung Entsprechenserklärung zum Public Corporate Governance Kodex	2. 2.	Verankerung Entsprechenserklärung in Unternehmenssatzung im Rahmen der Erklärung zur Unternehmensführung	6			
2.	Geltungsbereich und Verankerung Entsprechenserklärung zum Public Corporate Governance Kodex	2. 2.	Verankerung Entsprechenserklärung in Unternehmenssatzung im Rahmen der Erklärung zur Unternehmensführung	7			
2.	Geltungsbereich und Verankerung Entsprechenserklärung zum Public Corporate Governance Kodex	2. 3.	Überprüfung und Anpassung	8			
3.	Gesellschafterrolle, Gesellschafterversammlung und Organisationselement Beteiligungsmanagement	3. 1.	Gesellschafterrolle	9			
3.	Gesellschafterrolle, Gesellschafterversammlung und Organisationselement Beteiligungsmanagement	3. 1.	Gesellschafterrolle	10			

Das Beteiligungsmanagement soll überprüfen, ob die Abgabe und Veröffentlichun Entsprechenserklärung im Rahmen der Erklärung zur Unternehmensführung gen Regelungsziffer 5 erfolgt ist und darüber in Bezug auf jedes Unternehmen bericht Das Beteiligungsmanagement soll darauf hinwirken, dass Aufsichtsorgan und Geschäftsführungsorgan die Erklärung erstellen und veröffentlichen.

Industry, Innovation and Infrastructure
Sustainable Cities and Communities
Responsible Consumption and Production
Climate Action
Life below Water
Peace, Justice and Strong Institutions
Partnerships for the Goals

Figure 17: Agenda Analytics in-depth report

Agendas of interest could not only be the sustainable development goals[11] (SDGs) or environmental social governance risks in banks[12] (ESGs) but also the e. g. the public governance codex[13], addressing the emerging interest on governance topics in public companies.

Due to the wide range of possible agendas used and companies analyzed, the fields of application could be the:

- Identification of regional clusters of economic activity,
- Identification of options for inter-company cooperation,
- Analysis of text corpora regarding compliance with regulatory objectives,

[11] https://sdgs.un.org/goals
[12] https://home.kpmg/xx/en/home/insights/2021/05/esg-risks-in-banks.html
[13] https://publicgovernance.de/html/de/Public-Corporate-Governance-Kodizes-und-Beteiligungsrichtlinien.htm

- Monitoring of the reception and implementation of strategic objectives,
- Support of company internal and cross company knowledge management.

From a technical point of view, the use case leverages the SALTED pipeline in a way, where it subsequently leverages SALTED services for all steps taken. In the data discovery and data collection step, the companies of interest are supplied with the web URL representing their online presence. A mapping service can be used for evolving the information at hand into a proper NGSI-LD format. A publishing service can check for quality and post the entities into the graph. Later, enriching services like the compliance engine service or the search engine service are used to generate more information and evolve the representation in the graph. Finally, the Agenda Analytics application, the service providing the reports and visualization, depends on the graph as input data source, and only uses the NGSI-LD API.

This first use case serves to test the SALTED pipeline and, through its top-down approach, supports the inclusion of the application perspective in the development of the pipeline as a whole.

5.2 Smart Agriculture

“Smart farming is a management concept focused on providing the agricultural industry with the infrastructure to leverage advanced technology.”[14]

The advance of technology in the field of agriculture makes it possible to obtain a large amount of data. These data enriched in the right way can be used to provide useful information to the agricultural sector. The objective is to increase the performance of the agricultural installations.

The use case is to develop a proof-of-concept application involving Smart Agriculture. The application will consume the heterogeneous data already available on the SALTED platform that presents an apparent dashboard and that gives meaning to the application within Smart Agriculture.

The main data we are interested in is the carbon footprint, we can extract it at 200x200 scale. It will be presented in GEOJSON format. In each pixel of the raster there are four layers, the maximum value of CO2, the minimum, the median and the mean.

The main objective of the application is to relate the available data on the carbon footprint with various existing crops in a region, this information will be displayed on a map. Therefore, predictions can be made of which is the most convenient area to grow crops based on contamination.

[14] https://www.techtarget.com/iotagenda/definition/smart-farming

6 CONCLUSION

With this report, we provide the D2.1 deliverable on the SALTED Data Linking and Enrichment Architecture, which we refer to as Data Enrichment Toolchain (DET). The DET is build around NGSI-LD and consists of six loosly coupled components (Data Discovery, Data Collection, Data Curation, Entity Linking, Entity Enrichment). We describe the main requirements in terms of the scope of available data sources and their characteristics, which guided our DET design and architecture, our separation of data and contol plane and the working of the single components.

For each component we will implement different “component flavors” adapted to the specific data and application requirements.

This deliverable has provided an alignment between the SALTED project partners and has build some common ground from which we are implementing the individual components and, most importantly, setting the key common components that establish the interfaces among the components independently of the flavour to which they belong or the kind of data that they are actually processing. Moreover, it serves as basis for the SALTED applications for the Smart City and Smart Agriculture use cases.

7 BIBLIOGRAPHY

[1] T. Berners-Lee, “Linked Data - Design Issues,” [Online]. Available: https://www.w3.org/DesignIssues/LinkedData.

[2] F. F. T. F. I. and O. a. A. S. C. , “Smart Data Models,” [Online]. Available: https://smartdatamodels.org/.

[3] U. H. L. T. a. A. S.-C. Hunkeler, “MQTT-S—A publish/subscribe protocol for Wireless Sensor Networks,” in *3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08). IEEE.*, 2003.

[4] J. Barrasa, A. E. Hodler and J. Webber, Knowledge Graphs: Data in Context for Responsive Businesses, O'Reilly Media, 2021.

8 APPENDIX

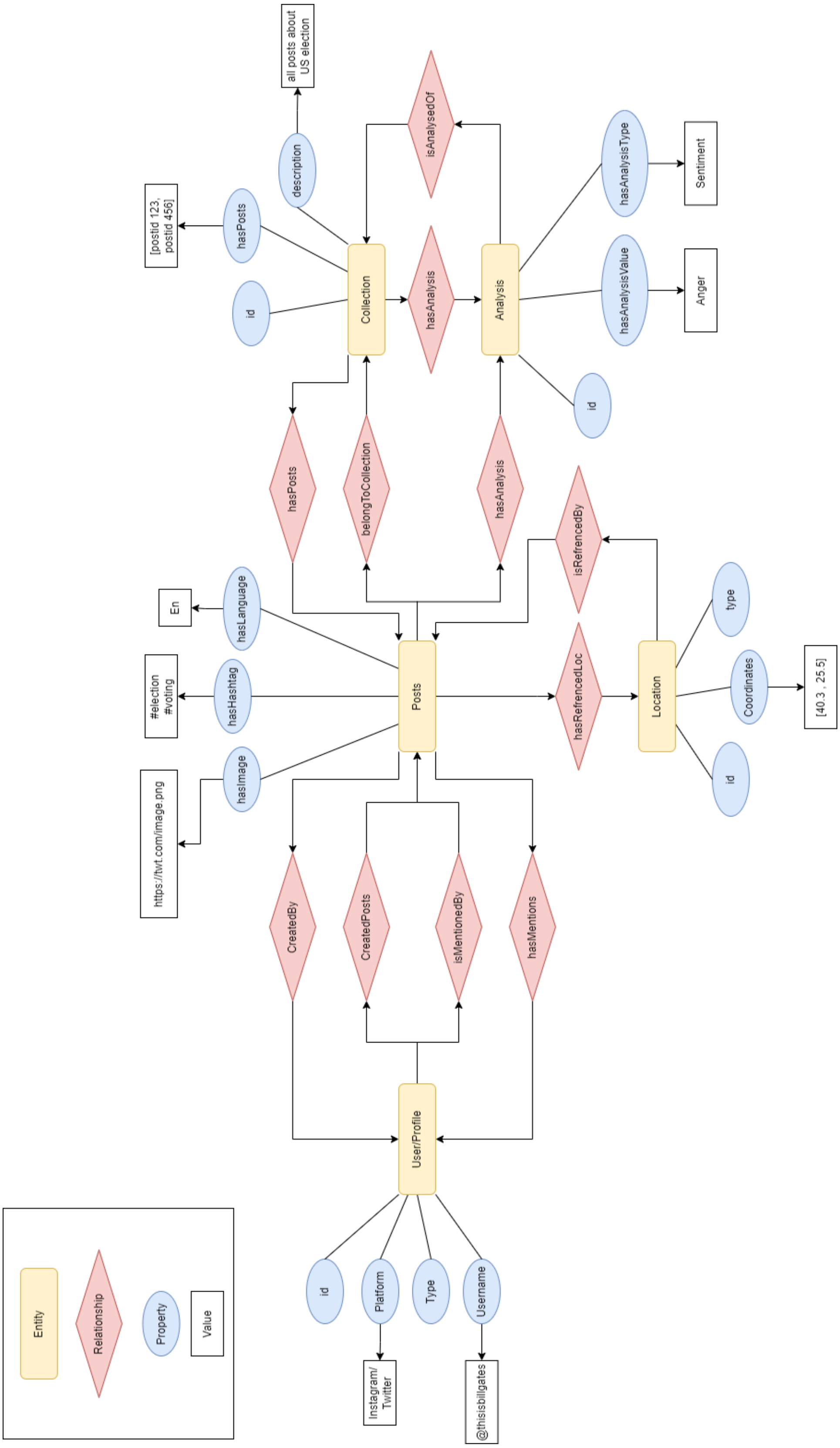


Figure 18. Social media smart data model