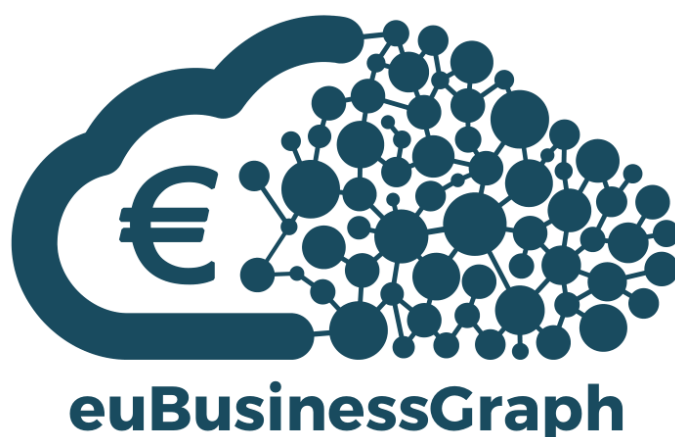


Innovation Action (IA)

ICT-14-2016-2017

H2020-ICT-2016-1

Enabling the European Business Graph for Innovative Data
Products and Services



Deliverable D3.1:

Requirements Analysis, Architecture and API Specification for the euBusinessGraph Marketplace – v1

Date	30.06.2017
Author(s)	Brian Elvesæter (SINTEF), Gencer Erdogan (SINTEF), Vincenzo Cutrona (UNIMIB), Flavio De Paoli (UNIMIB), Matteo Palmonari (UNIMIB), Stefan Dragnev (ONTO), Matjaz Rihtar (JSI), Aljaz Kosmerli (JSI)
Dissemination level	Public (PU)
Work package	WP3
Version	1.0

Document metadata

Quality assurers and contributors

Quality assessor(s)	Javier Paniagua (SDATI), Chris Taggart (OCORP)
Contributor(s)	euBusinessGraph Consortium

Version history

Version	Date	Description
0.1	22.05.2017	Initial Table of Contents (ToC).
0.2	12.06.2017	Updated structure and initial input from SINTEF and ONTO.
0.3	22.06.2017	Input from JSI.
0.4	23.06.2017	Input from UNIMIB.
0.5	26.06.2017	Updated input from SINTEF and ONTO.
0.6	26.06.2017	Initial draft ready for 1 st round of internal peer review.
0.7	27.06.2017	Updated input from UNIMIB. Addressed comments from 1 st round of internal peer review.
0.8	28.06.2017	Updated draft ready for 2 nd round of internal peer review.
0.9	29.06.2017	Updated input from ONTO. Addressed comments from 2 nd round of internal peer review.
1.0	30.06.2017	Final formatting and layout.

Executive summary

The main goal of the euBusinessGraph project is to create the foundations of a European cross-border and cross-lingual business graph through aggregating, linking, and provisioning (open and non-open) high-quality company-related data, thereby demonstrating innovation across sectors where company-related data value chains are relevant. This is achieved by leveraging the power of the emerging technologies such as Data-as-a-Service and Linked Data.

This report describes the initial architecture of the business graph provisioning of the **euBusinessGraph Marketplace** platform. The platform aims to:

- Integrate, host, and sustain a scalable business graph data marketplace, with capabilities for data cleaning, enrichment, integration, interlinking, publication and hosting;
- Serve as an entry point for company-related data discovery, exploration and analytics; and
- Grow an ecosystem of 3rd party applications and company-related data services.

The focus of this document is on the euBusinessGraph Marketplace platform and provides:

- An overview of the **euBusinessGraph Marketplace platform**, the relevant stakeholders of the platform, and their requirements for the capabilities of the platform;
- A **preliminary design and architecture of the platform** in terms of the core components, and a set of APIs that will guide the development of the platform in the next phase.

Table of contents

1	INTRODUCTION	6
1.1	OBJECTIVE	6
1.2	RELATIONSHIPS TO OTHER WORK PACKAGES AND DELIVERABLES	6
1.3	DOCUMENT STRUCTURE	7
2	EUBUSINESSGRAPH STAKEHOLDERS.....	8
2.1	APPROACH	8
2.2	STAKEHOLDERS.....	9
2.2.1	<i>Data providers.....</i>	9
2.2.2	<i>Data consumers (business products and services)</i>	9
2.2.3	<i>Marketplace technology providers.....</i>	10
3	REQUIREMENTS SPECIFICATION	11
3.1	DATA PROVIDERS REQUIREMENTS	11
3.2	DATA CONSUMERS REQUIREMENTS	12
3.3	MARKETPLACE TECHNOLOGY PROVIDERS REQUIREMENTS	14
3.3.1	<i>Data analytics.....</i>	14
3.3.2	<i>Shared data models.....</i>	14
3.3.3	<i>System of Identifiers requirements.....</i>	15
4	PLATFORM ARCHITECTURE	16
4.1	DATA PREPARATION SERVICES.....	16
4.1.1	<i>Data Import</i>	16
4.1.2	<i>Data Cleaning and Transformation (RDF-ization).....</i>	16
4.2	DATA INTERLINKING SERVICES.....	17
4.2.1	<i>Named Entity Linking (Text2KG)</i>	19
4.2.2	<i>Semantic Labelling (Tabular2KG).....</i>	20
4.2.3	<i>Link Discovery (KG2KG).....</i>	22
4.3	DATA HOSTING SERVICES.....	23
4.4	CROSS-CUTTING BUSINESS CASES ANALYTICS SERVICES	24
4.5	MARKETPLACE AND OPERATIONAL SERVICES	25
4.5.1	<i>License Models</i>	25
4.5.2	<i>Security and Access Control.....</i>	25
4.5.3	<i>System Monitoring and Reporting.....</i>	25
4.5.4	<i>Platform Administration.....</i>	26
5	API SPECIFICATION.....	27
5.1	DATA PREPARATION SERVICES.....	27
5.1.1	<i>Data import, cleaning and transformations.....</i>	27
5.2	DATA INTERLINKING SERVICES.....	28
5.3	DATA HOSTING SERVICES.....	30
5.3.1	<i>Data hosting platform services.....</i>	30
5.4	CROSS-CUTTING BUSINESS CASES ANALYTICS SERVICES	31
5.4.1	<i>API for semantic multilingual annotation service.....</i>	31
5.4.2	<i>API for Event Registry's events detection.....</i>	32
5.4.3	<i>API for graph based analytics.....</i>	33
5.4.4	<i>API for relation extraction</i>	34
5.5	MARKETPLACE AND OPERATIONAL SERVICES	34
5.5.1	<i>Security and access control</i>	34
5.5.2	<i>Usage reporting.....</i>	36
6	CONCLUSIONS	37
APPENDIX A	REQUIREMENTS MATRIX.....	38
APPENDIX B	REQUIREMENTS FOR COMMON DATA MODEL	41

APPENDIX C	INITIAL LIST OF SYSTEMS OF IDENTIFIERS	43
APPENDIX D	MINIMUM VIABLE PRODUCT (MVP)	45
D.1	CURRENT VERSION OF THE MVP	45
D.1.1	<i>Federated search</i>	45
D.1.2	<i>View and compare information about a specific company</i>	46
D.2	WHAT WE PLAN TO DO FOR THE FUTURE VERSIONS OF THE MVP	48

1 Introduction

This report presents Deliverable D3.1 "Requirements Analysis, Architecture and API Specification for the euBusinessGraph Marketplace – v1" of the euBusinessGraph project. This deliverable is developed as part of Work Package 3 (WP3) "euBusinessGraph Provisioning".

1.1 Objective

The objective of WP3 is to develop, integrate, deploy and maintain the technical infrastructure, methods, tools and services for reliable provisioning of the business graph by applying the Data-as-a-Service (DaaS) paradigm. WP3 will cover aspects related to simplifying the onboarding and population of the business graph (including data preparation and transformation mechanisms and semi-automated interlinking of data), cost-effective hosting of business graph data, support for cross-cutting business cases analytics tasks, and creation and maintenance of a data marketplace for the business graph.

WP3 aims to adapt tools and approaches for tasks such as data transformation, cleaning and integration, enrichment and interlinking, storage and scalable querying, access control, methodologies for data publishing, etc., in order to simplify the business graph data publication and consumption process, and analytics task on top of the business graph. Work in the WP will include customisation of existing approaches and tools, where necessary extensions of components and methods needed to offer an integrated data infrastructure that the marketplace can be built upon. WP3 will reuse and build upon results from the FP7 DaPaaS and Horizon 2020 proDataMarket projects for data preparation, hosting, and marketplace services.

The aim of this deliverable is two-fold:

1. To introduce the **euBusinessGraph Marketplace platform**, the relevant stakeholders of the platform, and their requirements for the capabilities of the platform;
2. To provide a **preliminary design and architecture of the platform** in terms of the core components, and a set of APIs that will guide the development of the platform in the next phase.

1.2 Relationships to other Work Packages and Deliverables

The development in WP3 follows an iterative approach resulting in two versions of the euBusinessGraph Marketplace platform released in month 12 and month 24 of the project as illustrated in Figure 1 below. Deliverable D3.1 serves as the specification for the first release (Deliverable D3.2 in month 12). An updated document (Deliverable D3.3 in month 15) will serve as the specification for the second release (Deliverable D3.4 in month 24).

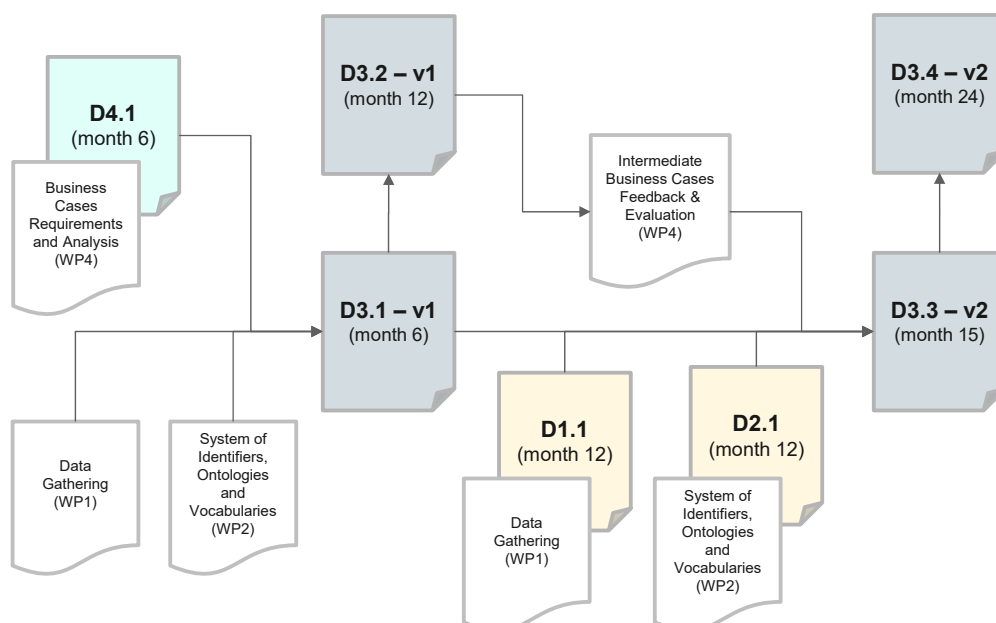


Figure 1: Relationship to other work packages and deliverables

The iterative approach allows us to develop the first prototype based on initial results in the project and adjust the development mid-way based on updated technical results and intermediate feedback and evaluation. The requirements analysis for the first specification (Deliverable D3.1) is based on the business cases requirements and analysis from WP4 (Deliverable D4.1), initial data gathering results in WP1 and initial system of identifiers and vocabularies concepts developed in WP2. Updated results from WP1 (Deliverable D1.1) and WP2 (Deliverable D2.1), and intermediate business cases feedback and evaluation, will be taken as input for the updated specification (Deliverable D3.3).

1.3 Document structure

The remainder of this report is structured as follows:

- Section 2 provides an overview of the euBusinessGraph Marketplace platform and the relevant roles played in the euBusinessGraph context.
- Section 3 presents a set of requirements for the euBusinessGraph Marketplace platform collected through analysis of dataset templates, business case descriptions and requirements input from the business cases.
- Section 4 outlines the initial design and architecture of the euBusinessGraph Marketplace platform services.
- Section 5 provides a preliminary set of APIs for the euBusinessGraph Marketplace platform services.
- Section 6 provides a brief summary and concludes this report.

2 euBusinessGraph stakeholders

This section gives an overview of the euBusinessGraph Marketplace platform and the relevant roles played in the euBusinessGraph context

2.1 Approach

An overview of the euBusinessGraph approach is provided in Figure 2 below, depicting the connection between data consumers and providers of the business graph data, through the enhanced environment developed in euBusinessGraph.

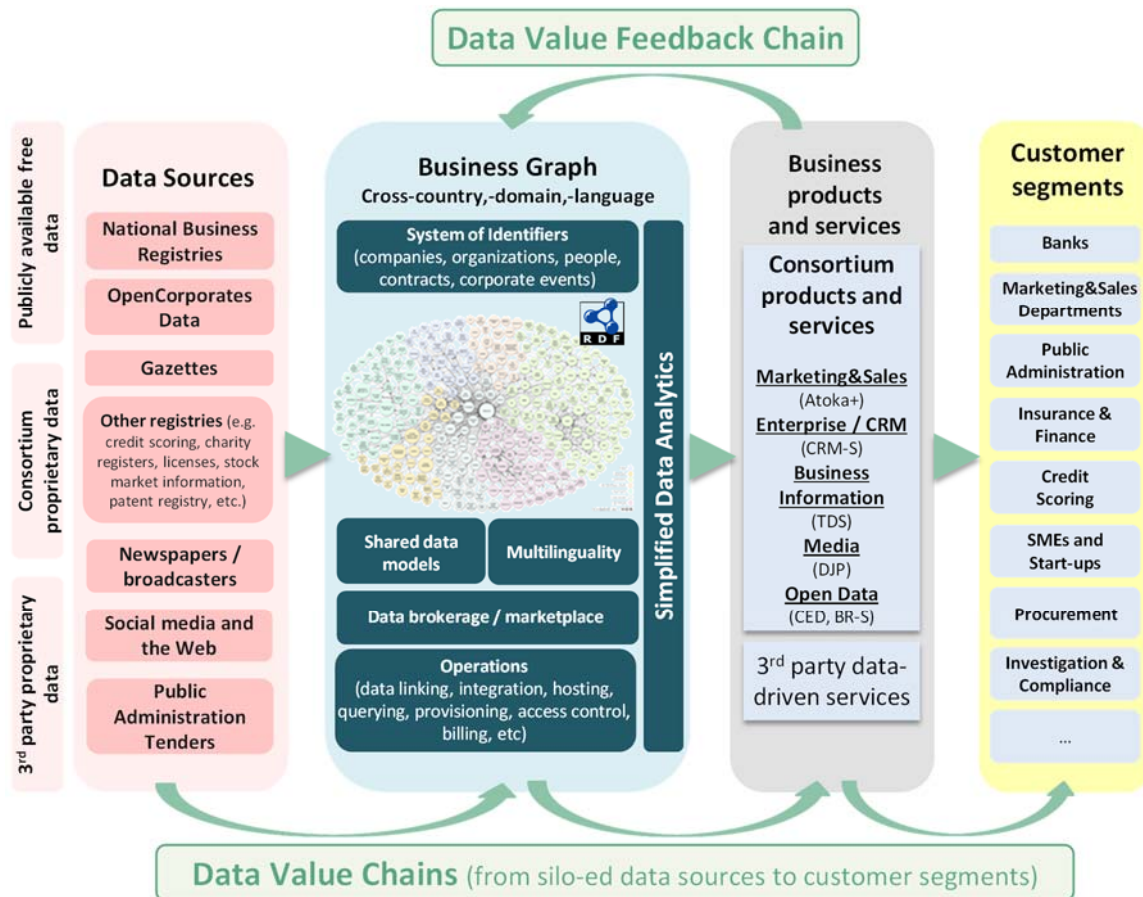


Figure 2: euBusinessGraph approach

The core element of the approach is the **business graph** component (second from the left in Figure 2). WP2 addresses the design of the business graph, which includes a system of identifiers for company-related data, shared data models and multilingual aspects. WP3 address the provisioning of the business graph, which includes support for data transformations, onboarding, hosting, access, analytics, data marketplace, and operations. In the process of transforming, integrating, publishing, and reusing data, two types of data value chains are enabled by euBusinessGraph:

- **Data value chain #1 – from silo-ed data sources to customer segments** (at the bottom of Figure 2): the value chain spans from the silo-ed data sources to various customer segments. The value is created through the set of capabilities provided by the business graph provisioning infrastructure, which are in turn leveraged to establish a set of innovative business products and services.
- **Data value chain #2 – data value feedback chain** (at the top of Figure 2): consists of the data insights generated by the proposed products and services being fed back into the business graph, therefore enhancing the value and scope of the data in the business graph. This is euBusinessGraph's mechanism to ensure that the business graph can host highly valuable and high-quality information, while at the same time increasing the chances of a self-sustainable business graph.

2.2 Stakeholders

A stakeholder in this context represents a group or organization that has interests or concerns in the euBusinessGraph Marketplace Platform. For the scope of the requirements analysis we consider the following stakeholders:

- **Data providers** that own and/or manage company and company-related data. As such, they can offer (i.e. provide, market and/or sell) their data via the business graph, or enrich and link their data using other relevant data offered via the business graph.
- **Data consumers**, i.e., the developers of business products and services, that want to use the data being offered via the business graph.
- **Marketplace technology providers**, i.e., the technology transfer providers that develop and maintain marketplace platform services to support the business graph data.

2.2.1 Data providers

The **data providers** in euBusinessGraph are OCORP, CERVED, SDATI, DW, BRC and JSI. They provide the *data sources* (the left part of Figure 2) that will be offered via the business graph:

- *National Business Registries*: Data from authoritative business registries providing official information about companies at the national level (e.g., identifiers of companies, addresses, industry codes, etc.)
- *Gazettes*: Public records of company-related legal notices.
- *Other registries*: Data from registries created in various sectors such as credit bureau registries, charities, licenses, stock markets, patent registries, etc.
- *OpenCorporates*: Open data about over 92 million companies, primarily collected from public sources.
- *Newspapers/broadcasters*: Unstructured company-related data from media sources in various languages.
- *Social media and the Web*: Unstructured or semi-structured company-related data from companies' websites and social media.
- *Public Administration Tenders*: data about companies participating in public tenders.

2.2.2 Data consumers (business products and services)

The **data consumers** in euBusinessGraph aim to develop *business products and services* (the second from the right in Figure 2) that will access and use the data made available via the business graph for creation of data-driven products and services. Below we only list and provide a brief description of the business products and services to be developed in the business cases. Further details can be found in Deliverable D4.1.

- **OCORP Corporate Events Data Access Service (CED)** is a new product to provide cross-jurisdictional data and alerts about changes in companies, deriving these from official primary sources (primarily company registers and government gazettes), and making them available in a standardised form.
- **CERVED Tender Discovery Service (TDS)** is a new set of algorithms and services easing and facilitating discovery and participation of business companies in public administration tenders.
- **SDATI Atoka+** will extend the Atoka service, which currently only provides company-related data in Italy, to cover new jurisdictions, specifically company-related data in the United Kingdom and Norway.
- **EVERY CRM Service (CRM-S)** is a novel service utilizing machine-learning algorithms to deliver insights using data from the business graph to their customers' databases through an API.
- **DW Data Journalism Product (DJP)** is envisaged to be a web-based application that supports journalists in dealing with complex and large volumes of company related data across the three journalistic workflows: search, monitoring and content production.

- **BRC Norwegian Registries API Service (BR-S)** is a complete set of services to authoritative business data from Norway.

The **customer segments** (the first from the right in Figure 2) represent customers of the *business products and services*. The focus here is on the value created to the end users of the business products and services. This stakeholder is considered out of scope for the requirements analysis in WP3, as any requirements from such a stakeholder should be covered indirectly in the requirements from the business cases.

2.2.3 Marketplace technology providers

The marketplace technology providers in euBusinessGraph are SINTEF, OCORP, SDATI, ONTO, JSI and UNIMIB. They are the providers of the marketplace services such as system of identifiers, shared data models, multilingual support, data transformations, data onboarding, data hosting, data access, data analytics, data marketplace, and operations.

In addition to the requirements from the data consumers' and data providers' point of views, there may be technical requirements and constraints from the technology providers' point of view that must be taken into account for the development of the marketplace services. In particular, any design requirements and constraints from WP2 should be taken into consideration. As can be seen in Figure 3, the *system of identifiers* and *shared data models*, as well as data *analytics services* may impose requirements and constraints on the development of the euBusinessGraph provisioning services. For this first version of the euBusinessGraph Marketplace platform specification, we have taken preliminary working results from WP2 into account.

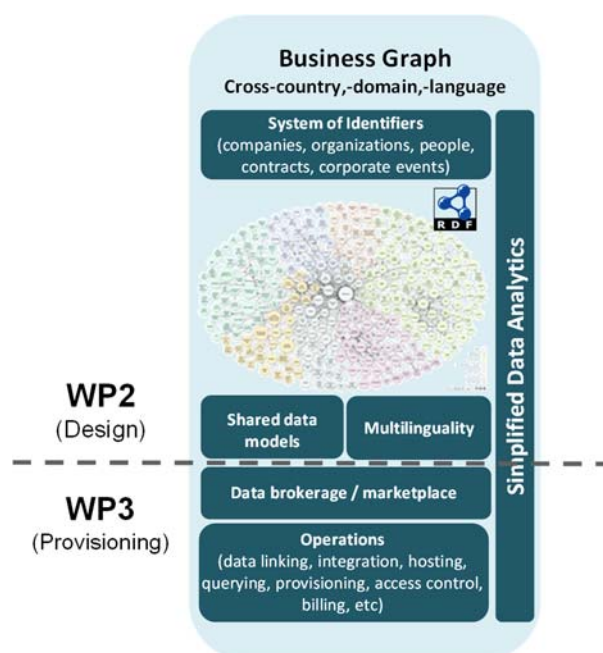


Figure 3: euBusinessGraph design and provisioning

3 Requirements specification

This section provides details on the requirements analysis and the resulting requirements for the euBusinessGraph Marketplace platform.

3.1 Data providers requirements

For the data providers (OCORP, CERVED, SDATI, DW, BRC and JSI), the requirements analysis were based on:

- Discussions through meetings and concalls.
- Requirements input collected for each business case in the project's Wiki collaboration platform.
- Analysis of the business cases descriptions in Deliverable D4.1.
- A set of initial dataset templates as part of Work Package 1 (WP1). The dataset templates covered aspects such as:
 - Data collection – how the data is collected, e.g., mandatory by law;
 - Data structure – description of the data structure;
 - Standards and metadata – standards followed and metadata available;
 - Dataset license and availability – Under which license the dataset is made available;
 - Data access – how the data can be accessed – e.g. queries endpoints, RESTful APIs, original data import, database dump, etc.;
 - Size, frequency, coverage and language – dataset size, update frequency, time and spatial coverage and language;
 - Data identification – the identifier used for the data;
 - Data privacy and sharing – is there sensitive datasets or parts of the datasets that should remain private, what kinds of access restrictions should be implemented – both for input datasets and the output (integrated) dataset for the use case;

These inputs were analysed with respect to technical requirements for the euBusinessGraph Marketplace platform. The technical focus for this analysis was on the data provider role that aims to share its data to the business graph. The results of this analysis has been summarised into the requirements table below.

Table 1: Data provider requirements (DPRs)

ID	Requirement	Requirement description
Data Preparation Services		
DPR-01	Dataset import	Dataset import for relevant data formats (e.g. RDF, CSV, JSON, XML, REST service).
DPR-02	Dataset cleaning and transformation	Data cleaning and transformation activities (RDF-ization).
Data Interlinking Services		
DPR-03	Entity linking	Entity linking get as input a text and produces a set of annotations.
DPR-04	Semantic labelling	Semantic labelling get as input a (weakly) structured source and produces a set of annotations that are used for generating mappings.
DPR-05	Link discovery	Link discovery get as input a knowledge graph and produces a set of mappings.

Data Hosting Services		
DPR-06	Data access	Specifying access through different channels (e.g. SPARQL endpoint, original data download, REST APIs, reporting service).
DPR-07	Data updates	APIs for accessing and modifying (updating) datasets.
DPR-07-a	API for incremental update	API for incremental update of data.
DPR-07-b	API for bulk update	API for bulk update of data.
DPR-08	Dataset metadata	Management of metadata, such as standardized name, description, language, including company-specific extensions, such as jurisdictions.
DPR-08-a	Common vocabulary	Ability to describe data through a common/standard vocabulary.
DPR-09	Big data storage	Storage capability for large data volumes (RDF).
Cross-Cutting Business Cases Analytics Services		
DPR-10	Multi-lingual annotation	Support for multi-lingual annotation of text with links to relevant/common concepts.
Marketplace and Operational Services		
DPR-11	License models	Support for different license models for accessing and APIs for accessing and modifying datasets, including dual license models that allows for both payment plus share-alike for public-benefit use.
DPR-11-a	Dataset-level license access control	Access control policy at dataset-level.
DPR-11-b	Data item license access restrictions	Access control policy at data-item level.
DPR-11-c	Advertise company data	Ability to advertise private graphs and direct consumers to them.
DPR-11-d	Shared revenue	Revenue originating from data flows are shared.
DPR-12	Security and access control	Security and access control for users and groups.
DPR-13	User registration and access management	Manage access of users to APIs and groups.
DPR-14	Secure access policy specification	Specify policy through different API keys for different users or groups.

3.2 Data consumers requirements

For the data consumers (CERVED, SDATI, EVRY and DW), the requirements analysis were based on:

- Discussions through meetings and concalls.
- Requirements input collected for each business case in the project's Wiki collaboration platform.
- Analysis of the business cases descriptions in Deliverable D4.1.

It should be noted that amongst the business cases there are partners that represent both the data provider and data consumer roles (i.e., CERVED, SDATI and DW) and thus see things from both perspectives. The result of this analysis has been summarised into the requirements table below.

Table 2: Data consumers requirements (DCRs)

ID	Requirement	Requirement description
Data Hosting Services		
DCR-01	Multiple dataset programmatic access channels	Access through different channels (e.g. SPARQL endpoint, original data download, REST APIs, reporting service).
DCR-01-a	Data dump	Ability to download large volumes of data as data dumps.
DCR-02	Searching and exploring existing datasets	Dataset search and browse/explore functionality.
DCR-02-a	Single access point	Single access point to information about company data.
DCR-02-b	High availability and efficient querying	Efficient querying based on indexation of the data based on pre-defined sets of indexes. High availability should be ensured for the indexed data by the hosting platform.
DCR-02-c	Extended company profile	Ability to access extended company profiles that integrates data from multiple data sources.
DCR-03	Access to dataset metadata information	Access to metadata information.
DCR-03-a	Detailed information about data	Detailed information about data that can be found in other data repositories.
Cross-Cutting Business Cases Analytics Services		
DCR-04	Multi-lingual annotation	Support for multi-lingual annotation of text with links to relevant/common concepts.
DCR-05	Event	Support for event detection.
DCR-06	Graph based analytics	Support for clustering and similarities amongst entities, e.g. resolving ambiguity.
DCR-07	Relation extraction	Support for extracting relations between entities.
Marketplace and Operational Services		
DCR-08	License models	Support for different license models for accessing and APIs for accessing and modifying datasets.
DCR-08-a	Shared agreement models	Integrated navigation to data with shared business/revenue agreements
DCR-09	Secure access to platform APIs	API keys for specifying access policies for different users.
DCR-10	User registration and access management	User sign-up, log-in and profile management.

3.3 Marketplace technology providers requirements

As written in Section 2.2.3, we have also considered technology providers requirements for the data analytics services and preliminary working results related to the development of the system of identifiers and the common data model in WP2.

3.3.1 Data analytics

The data analytics of the euBusinessGraph Marketplace platform are supported by the *Cross-Cutting Business Cases Analytics Services* that are specified in sections 4.4 and 5.4. These services, provided either by partners or collected/generated locally, will need the following data:

- Company info (source: partners, local data gathering)
- People info (source: partners, local data gathering)
- Product/brand info (source: partners, local data gathering)
- Annotated articles and events (source: JSI Event Registry and JSI Wikifier)
- Relations from news (source: JSI relation extraction)
- Relations between companies, people and products/brands (source: partners, JSI relation extraction)

3.3.2 Shared data models

For sharing of data in the business graph we have discussed three different options.

1. **All data is shared:** Data providers joining the business graph share all their data.
 - **Advantages:** Single point of access to all data. Eases data integration and allows analytics services to run on the data shared in the business graph infrastructure.
 - **Disadvantages:** Big volume of data. Scalability problems with frequent updates etc.
2. **A common data subset is shared:** Data providers joining the business graph share a common subset of their data described according to a common data model.
 - **Advantages:** Single point of access to all data, but only a subset is stored in the euBusinessGraph infrastructure. Overcomes some big data scalability challenges when compared to option 1 (where all data is shared). Data can be indexable, which allows for efficient data discovery, faceted search, ranking, etc.
 - **Disadvantages:** Simple analytics services can run on the data shared in the business graph infrastructure, while more advanced analytics services will require additional (local) data and must be run locally.
3. **No data is shared (common data model + pointers to data):** Data providers joining the business graph share descriptions of their data (metadata) with pointers to the actual data (search and access APIs). This represents a federated approach. All data that is offered via the business graph must be described according to a common vocabulary, i.e. the common data model.
 - **Advantages:** Does not require data providers to share any data. Can be viewed as a virtual knowledge graph whose storage is distributed in different physical databases – where different parts are accessed under different licenses.
 - **Disadvantages:** Performance issues with federated search and data retrieval. Makes it problematic to support faceted search, merging result lists, ranking, linking, or reliable pagination. Will require data providers to implement a common search and data access API so that results can be ranked similarly amongst all data providers.

Based on the initial discussions and requirements input collected from the business cases, there were different views on the sharing of data. While some data providers were willing to share data and making all their data indexable (e.g. to support faceted search), others were more concerned and wanted to use the euBusinessGraph Marketplace platform more as a means of enriching, promoting and marketing their data instead of sharing their data fully.

One particular requirement raised by the business cases was to make it simple for data providers to join the business graph. Thus rather than requiring data providers to share all their data, a better strategy would be to give data providers a choice of how to join the business graph. Data providers can choose whether they want to share all data, a common subset of data or only provide pointers to their data.

Data providers that primarily want to market their data, but not share it directly through the euBusinessGraph infrastructure, would be required to describe their data according a common vocabulary and implement a common search API that allows the euBusinessGraph Marketplace to query data that are being offered. Data consumption in this case will be through the specific APIs of the data providers.

Data providers that are willing to share their data can use the data hosting facilities of the euBusinessGraph Marketplace platform, make use of cleaning and transformation services to map the data to RDF linked data, and make use of data interlinking services to enriched and linked their data with other data sources. Some data providers may want to fully host and share their data in the euBusinessGraph infrastructure, while for other data providers it make more sense to only provide and share a minimum common subset. This choice may depend on the size of the dataset, data governance, business policies, etc.

For the first release of the euBusinessGraph platform we are exploring support for option 3 and 2. Initial work in WP2 is now developing a common vocabulary for the common data model. This data model can be used to both describe the data being offered and used to share a common data subset. The model is a foundation for the business graph, whose purpose is to distribute company-related data from various data providers in a uniform way and facilitate implementation of business products and services on top of this data. The first users of the business graph are the business cases of the project that intend to consume data from the graph.

3.3.3 System of Identifiers requirements

Work Package 2 (WP2) of euBusinessGraph will create and maintain a system of *shared identifiers* for companies in Europe, together with a mechanism for mapping to existing proprietary identifier systems, while at the same time relying on common conceptual models (e.g. shared vocabularies and ontologies) to address semantic heterogeneity and cross-lingual problems in company-related data.

An initial survey in WP2 has identified a set of external identifiers used in the relevant datasets for the business cases in the project. See Appendix C for the list of external identifiers. The common data model must be able to represent the identifiers used by the dataset, and the euBusinessGraph Marketplace platform must provide services that allows to create mappings between these identifiers.

4 Platform architecture

This section describes platform architecture covering the data preparation, data interlinking, data hosting, cross-cutting business cases analytics, and marketplace and operational services. For each of these service categories we are planning on developing a set of features as shown in Figure 4 below.

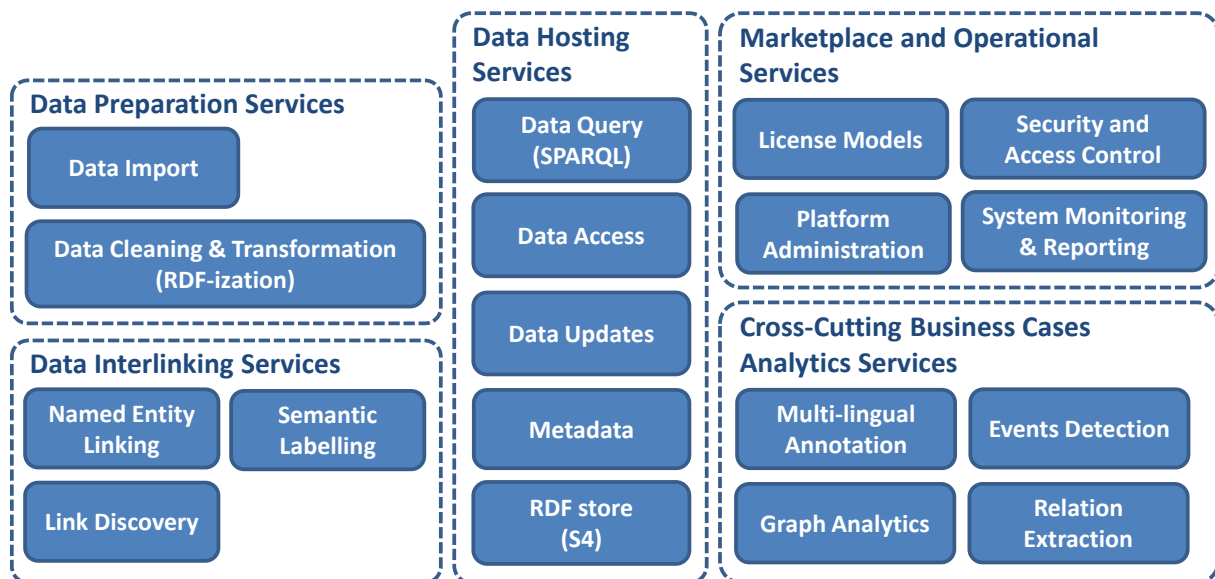


Figure 4: Platform services

4.1 Data Preparation Services

The *Data Preparation Services* will include reuse and customisation (including extensions where needed), of the Grafterizer¹ data cleaning and transformation approach (service deployed as part of the DataGraft platform²). This is meant to simplify data cleaning and transformation processes, and ensure the availability of services to help with the generation of the business graph data. The focus of these services will be on intelligent support in data cleaning and transformations using approaches such as predictive interactions in data transformation pipelines, support for data profiling and automated data quality issues resolutions.

4.1.1 Data Import

DataGraft comprises services and GUI tools that facilitate the importing of datasets into the platform. These include:

- A graphical tool for specifying data mappings and transformations;
- A framework for generating repeatable and executable data transformation services;
- The actual data transformation services deployed on the platform, which can import data in various formats and clean it and/or publish it into RDF;
- APIs for managing the uploading of data using standard read/write operations, such as OpenRDF API³ and a Linked Data Platform (LDP) API⁴.

4.1.2 Data Cleaning and Transformation (RDF-ization)

The Grafterizer component of the DataGraft platform was initially developed to support general-purpose data cleaning and transformation operations so that it could be applied in many different settings. Data cleaning and transformation in DataGraft platform is performed with the help of a “pipeline” concept. To

¹ <https://github.com/datagraft/grafterizer>

² <https://github.com/datagraft>

³ <http://rdf4j.org/>

⁴ <http://www.w3.org/TR/ldp/>

begin with, each single transformation step is defined as a pipe – a function that performs simple data conversion on its input. In DataGraft you are able to see the partial preview of the transformation on each step. Last option makes it possible to see how the transformed data looks like for every stage of transformation. The Grafterizer component also supports creation of RDF mappings. After having defined the pipeline for the data cleaning you can start creating RDF mappings.

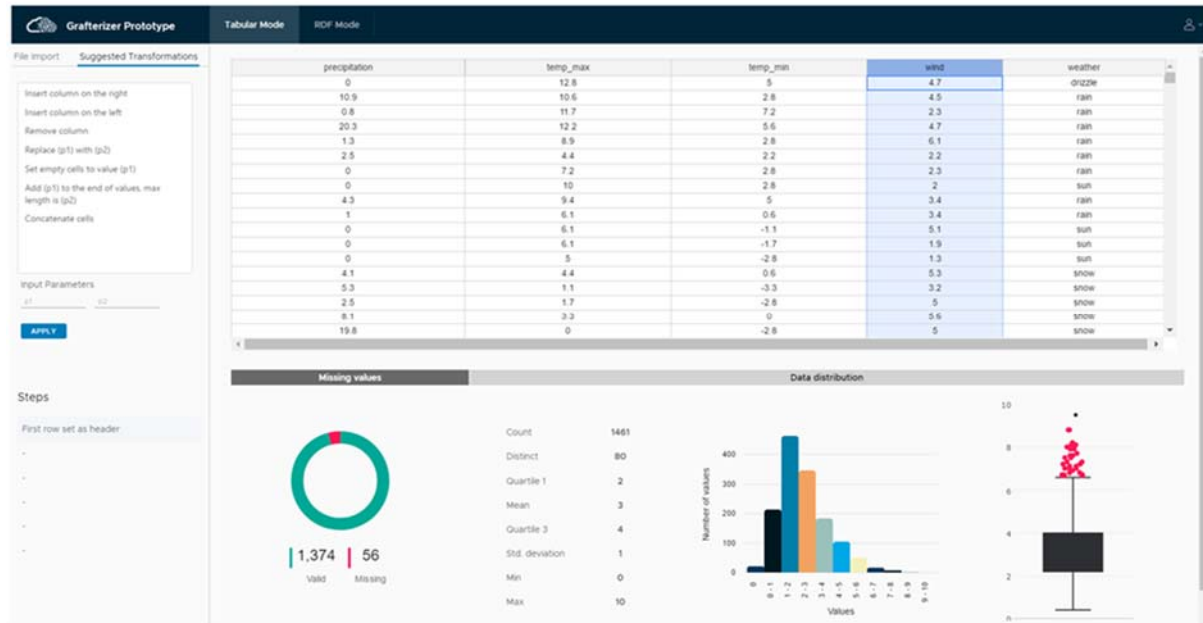


Figure 5: Visual data profiling extensions to Grafterizer

In euBusinessGraph we will extend the Grafterizer component with predictive interactions in data transformation pipelines, support for data profiling and automated data quality issues resolutions. An initial prototype demonstrating the visual data profiling extensions has been developed at SINTEF. It extends the functionality of the current data cleaning and transformation framework of Grafterizer. The prototype features capabilities for integrated, interactive data cleaning and transformation, and visual data profiling as shown in Figure 5.

Visual data profiling is the statistical assessment of datasets to identify and visualize data quality issues such as outliers or missing data values. The approach has the potential to help data scientists and data workers make an informed decision on how to deal with data quality issues, and reduces effort spent on cleaning and transforming data.

The framework has been validated in terms of usefulness and ease of use, and will be publicly available in future versions of Grafterizer. The following capabilities will be implemented:

- **Predictive, intelligent data cleaning and transformation** that recommends domain-specific and relevant next steps in the data preparation process. The data cleaning and transformation steps are incrementally applied in a pipeline approach.
- **Visual data profiling** that analyses and determines data quality based on statistical properties, semantics and structure of data. The data quality assessment is presented to the user by means of statistical and scientific charts and visualizations.
- **Direct manipulation table** in a spreadsheet style table view that dynamically integrates with the user interface. The table view can be manipulated interactively by the user.

4.2 Data Interlinking Services

Reaching the main goal of euBusinessGraph project, i.e. the creation of a cross-lingual business graph through aggregating, linking, and provisioning (open and non-open) high-quality company-related data, requires to integrate and link several data sources. Data interlinking is the task of establish semantic links between pieces of information represented in one or more than one independent sources. Without loss of generality, when establishing links between two data sets, we distinguish between a *source* piece of information (the data that have to be linked) and a *target* piece of information (the data to which links are established).

Different problems of data interlinking can be devised depending on the types of data that are considered. In euBusinessGraph interlinking is considered only against a reference Knowledge Graph (KG), i.e., the target data are represented as a KG. Table 3 shows different kind of data interlinking problems, based on the types of sources considered: Entity Linking, Semantic Labelling and Link Discovery.

Table 3: Data Interlinking tasks depending on the type of data sources

Source Data	Target Data	Task	Relevant in euBusinessGraph
Text	KG	Entity Linking	✓
Multimedia (Audio/Video/Image)	KG	Entity Linking	✗
Weakly structured source (e.g. JSON, XML, CSV)	KG	Semantic labelling	✓
Knowledge Graph	KG	Link Discovery / Ontology Matching / Entity co-resolution	✓
Structured source (e.g. relational database)	KG	Semantic labelling	✓

A Data Interlinking Service should get as input any piece of data (e.g. a document or a full dataset) and produce a set of annotations or mappings as output, depending on the specific task accomplished:

- Entity Linking get a text as input and produces a set of annotations;
- Semantic labelling get a (weakly) structured source as input and produces a set of annotations that are used for generating mappings;
- Link discovery get KGs as input and produces a set of mappings.

Figure 6 depicts an overview of the data interlinking process. Each service needs some built-in additional resources in order to complete its own task: the reference KG (considered as target) and a set of classification indexes. In the following sections, we will discuss deeply about the three main problems that the Data Interlinking Services address.

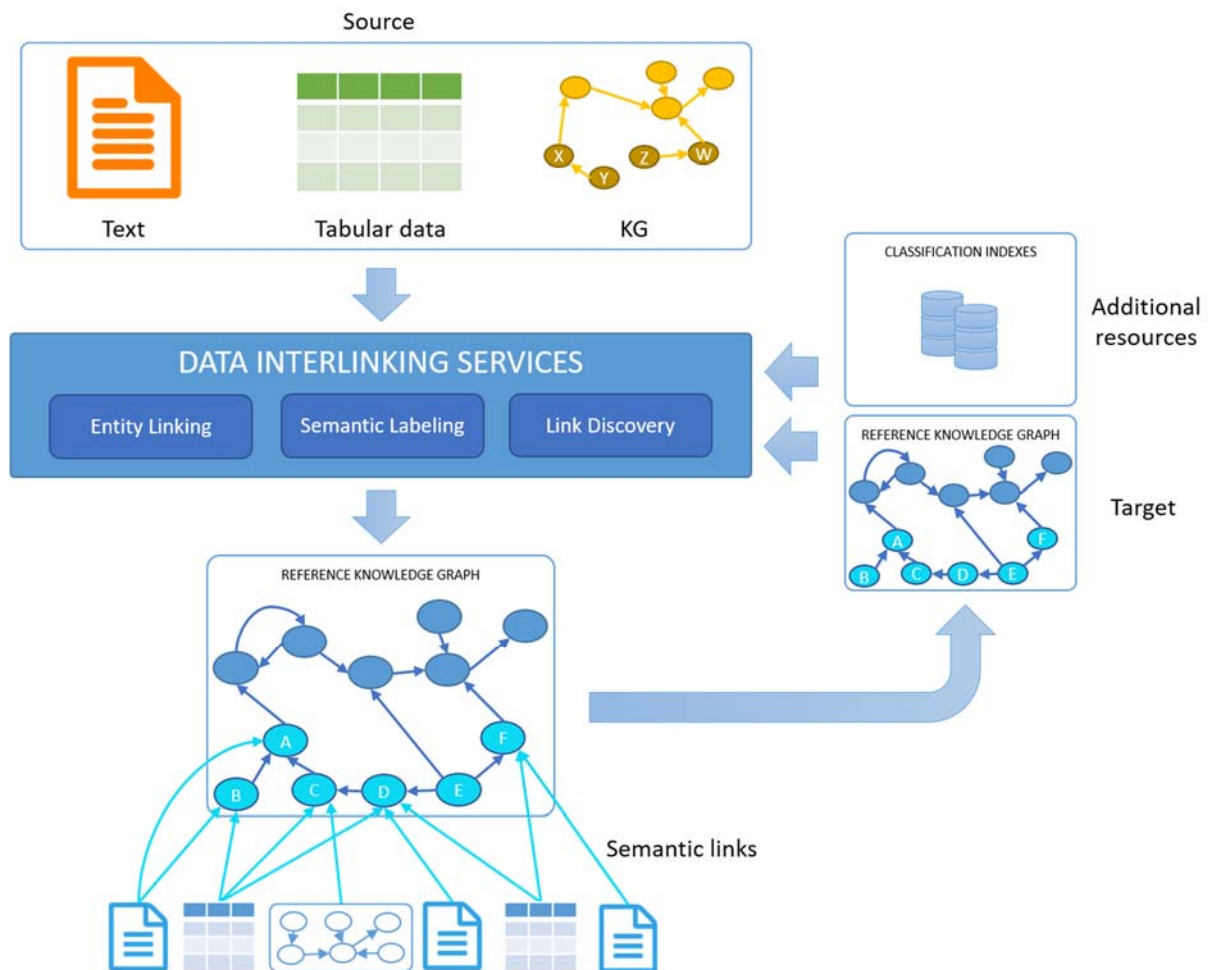


Figure 6: Data Interlinking Process

4.2.1 Named Entity Linking (Text2KG)

Given a text, the Named Entity Linking (NEL) task consists in finding all named entities (sequences of words in the text that are names of things, such as person and company names, or gene and protein names) mentioned inside the text that are also represented in a KG. This task is divided into two different subtasks:

- **Named Entity Recognition:** to find the mentions of entities inside the text (and, possibly to assign them with a class label, e.g., Person)
- **Named Entity Disambiguation:** to link the above mentions to entities in a KG (usually, instances of some class in the KG). This linking task must consider the meaning of the mentions: since a named entity mention can refer to multiple entities (named candidate entities), this task has to resolve the appropriate meaning (disambiguation phase) in the considered context.

Figure 7 shows the complete conceptual pipeline that has to be supported in EuBusinessGraph: given a text, the **Named Entity Recognition** task returns a set of annotation candidates, each one specifying the portion of text that is considered a mention of a named entity. A class such as 'person' or 'organization' could be assigned to the identified entity at this point (several state-of-the-art existing tools accomplish this task, e.g. GATE, OpenNLP, Stanford Named Entity Recognizer, etc.). However, since we aim to perform full linking to an established KG, which contains this information about the linked entities, this step can be skipped without any loss. The output of this step will be refined in the next step.

The candidate named entities are used as input for the **Named Entity Disambiguation** task (Step 2 in Figure 7), whose goal is to link each identified entity mention to an entity described in the reference KG. This task includes deciding which candidate entity, if any, has to be linked. The result of the disambiguation task may be a refusal to add a link for the candidate entity mention, if disambiguation is too uncertain to be trusted. Disambiguation needs to take into account the context of candidate entity

mentions and the relations between entities in the KG to produce the most likely set of linked entities. For example, if the text mentions the Louvre, Notre Dame and Avenue des Champs-Élysées than the string 'Paris' in this text is much more likely to refer to the French capital than miss Paris Hilton.

Some established systems perform only the Named Entity Recognition task, while most of systems that perform the full NEL pipeline do not share information about the output of subtasks performed by their algorithms, e.g., of the Named Entity Recognition task. In euBusinessGraph we need to link texts to entities described in KGs, thus we need to perform the full NEL pipeline. Several services performing NEL are available (e.g., DBpedia Spotlight, Babelify, AIDA), none of which focus on company names, one of the key target class of entities in euBusinessGraph. Two partners of euBusinessGraph provide their own NEL services:

- **Wikifier**⁵ (JSI): it links entity mentions to their Wikipedia concepts, i.e. the URLs of their related Wikipedia pages. For a specific language it is built using the corpus of Wikipedia pages in that language. It supports languages of the top 100 largest Wikipedias (i.e. largest corpus size in the sense of the number of pages).
- Entity Extraction in **Dandelion API**⁶ (SpazioDati): it finds mentions of places, people, brands and events in documents and social media using SpazioDati reference KG. It support fetching of additional data about the entities. It supports many languages.

In euBusinessGraph the following additional requirements can be defined for NEL services:

- perform linking considering texts in different languages;
- allow user to set a confidence parameter (more tags vs. more precision);
- return detailed information for each annotation found, where an annotation specifies for a source piece of data (e.g. sequence of words in a text) at least:
 - the URI of the linked resource;
 - the confidence score;
 - the class/type of the linked resource;

The NEL services described above fulfil also these additional requirements.

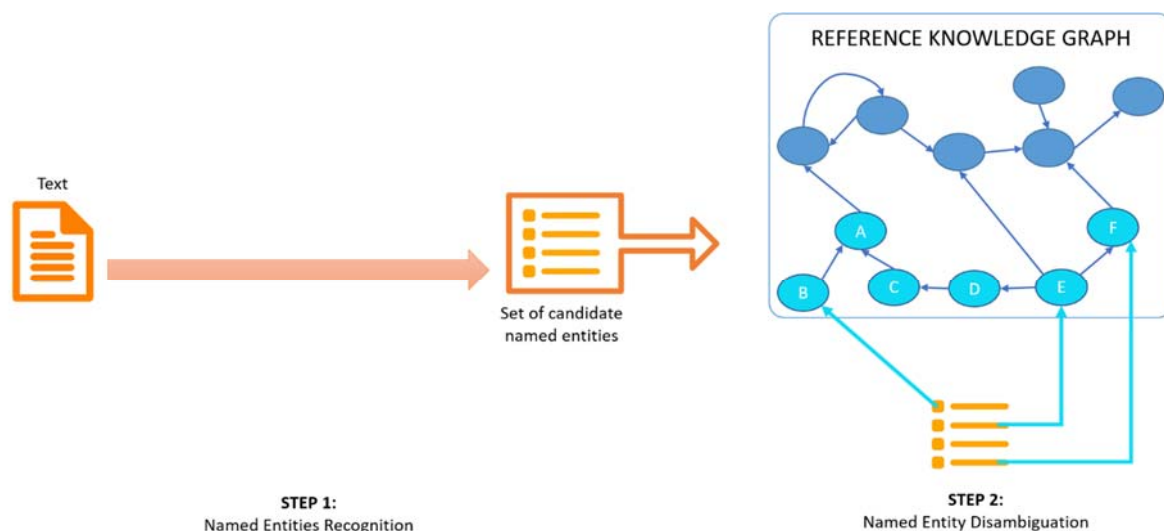


Figure 7: Entity Linking conceptual pipeline

4.2.2 Semantic Labelling (Tabular2KG)

The main goal of semantic labelling approaches (sometimes called also Table Interpretation or Table Annotation approaches) is to map different (weakly) structured data sources to a common KG. As (weakly) structured data sources we can consider CSV, JSON, Relational Data Base table (RDB), XML,

⁵ <http://wikifier.org/info.html>

⁶ <https://dandelion.eu/>

etc. Since we can transform all of these sources and obtain a CSV, in the following we consider only CSV as sources (using also “tabular data” as interchangeable name).

In the academic research state of the art this problem is well known and several studies try to address it in different way; in general, the process is divided into two main steps:

- schema alignment, which is related to mapping each attribute of a tabular data source with a semantic label (provided by an ontology);
- instances reconciliation, which is responsible of mapping cells value to entities stored in some knowledge bases. The instances reconciliation is an important step because allows the data enrichment phase, whose exploits the links between entities to find new knowledge.

Some of existing Semantic Labelling tools focus on the schema alignment task (e.g. Karma, STAN, Datalift and Juma); some other tools focus on the instances reconciliation (e.g. OpenRefine). There exist also tools that perform both (e.g. TableMiner+ and Linda). These tools can operate at different level of automatization. For the euBusinessGraph platform purposes, UNIMIB will provide a *semi-automatic Semantic Labelling Service* (SLS) based on the STAN (Semantic Table ANnotation) tool, able to accomplish the Semantic Labelling task either at schema and instances levels. This service will also rely on other services and resources:

- ABSTAT, an external service (provided by UNIMIB) for obtaining some information about classes and properties related to entities stored in different knowledge bases;
- a service for entities disambiguation focused on tabular data; since NEL techniques require significant adaptation to work in tables, UNIMIB proposes to rely on JSI’s Wikifier tool and adapt it to work in tables context;
- indexes to represent knowledge about classifications (if required).

The SLS gets a well-formed table as input, i.e. a table that is compliant with the following requirements:

- table can contain at most one row header;
- header cannot contain nested headers;
- table cannot contain nested tables;
- there exist some relations between columns;
- each cell contains atomic information as string;
- all elements included in a column are of the same format (e.g. for date and number).

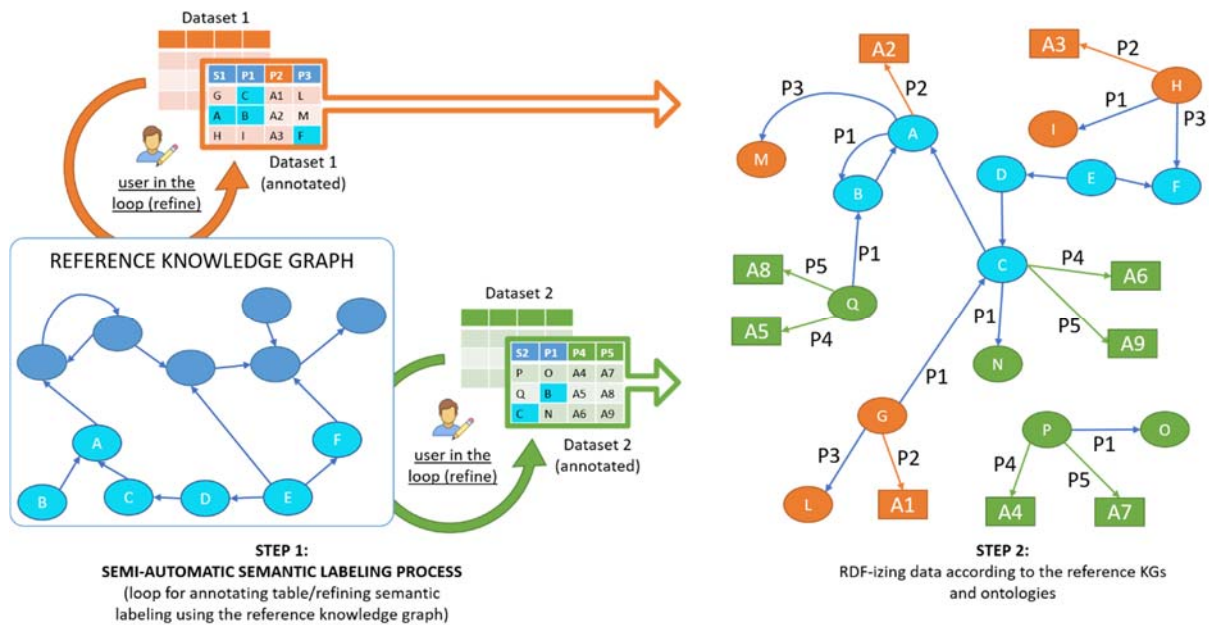


Figure 8: Semantic labeling task executed on different datasets

The SLP is a semi-automatic process (see Figure 8) that includes the user in the suggest/refine annotations loop to reach high-quality annotations with a low effort. The tool takes advantage from the summarization capabilities of ABSTAT to support user in decision making.

The result of SLS is a set of punctual annotations (that can be used for defining the mappings among the elements of dataset and KG), each one associated with an element of the table:

- Column Annotation, which includes the role in a RDF triple (subject and/or object) and the type (URI, literal or 'blank node');
- Value Annotation, which includes the URI of the linked resource, the confidence score and the class/type of the linked resource.

The information carried by the annotation must be those that are needed to generate mapping useful for the RDF-ization task.

Additional requirements for the SLS are:

- the capability to annotate tables written in different languages.

4.2.3 Link Discovery (KG2KG)

Link Discovery refers to linking tasks where source and target data are represented as KGs, when a commitment to a specific relation represented by the discovered links is not specified. When the task focus on specific kinds of relations, more specific names are often used, such as Ontology Matching, Entity co-resolution and Classification Matching:

- Ontology Matching is the task of discovering and establish links, also referred to as *mappings*, both at schema and instance level, which are represented by means of well-known ontological properties like owl:sameAs (links between instances that denote a same real-world entity), owl:equivalentClassOf, rdfs:subclassOf (links between two classes, where they are equivalent or one is subclass of the other one), owl:equivalentPropertyOf and owl:equivalentPropertyOf (links between two properties, where they are equivalent or one is subproperty of the other one);
- Entity co-resolution is the task of linking instances that denote a same real-world entity (frequently using the owl:sameAs property);
- Classification Matching is a particular case where the KGs considered are classifications (e.g. taxonomies).

In general, the Link Discovery task must perform using two generic KGs, one as input and one as target. In the euBusinessGraph context, we can assume that this task will perform using frequently the

reference KG as target. Figure 9 shows an example where the Link Discovery task is applied on 3 different KGs, using the reference KG as target. The blu arcs indicate properties declared in the target KG ontology. KG1, KG2 and KG3 use properties of the target ontology at different level of usage (total, partial and absent, respectively), showing that the Link Discovery Service must accept as input KGs based on different ontologies.

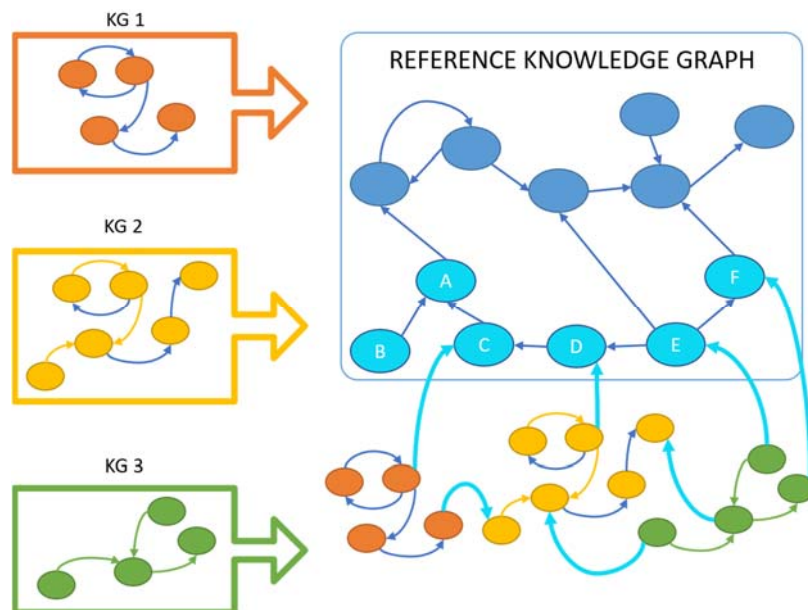


Figure 9: Application of Link Discovery task using different KGs as input

There exist several state-of-the-art tools that accomplish this task; the most used tools are LINES and Silk. LINES is a link discovery framework for the Linked Data, that uses also time-efficient and dimension-scalable approaches. Silk is an open source framework for integrating heterogeneous data sources, focused also on generating links between related data items within different Linked Data sources. Both require to set configuration parameters, such as the properties to be discovered, the similarity metrics to be used, and so on.

In euBusinessGraph UNIMIB will provide a Link Discovery service that gets a generic KGs as input and produces a set of mappings between entities represented also in the reference KG. The service has to be configurable, so the end user should be able to:

- declare properties to be considered while performing task;
- set the similarity metric to be used;
- choose a different target KG;
- set a threshold to decide when a discovered link can be automatically set or has to be reviewed by the user.

The result of this task is a set of mappings between entities represented in both the KGs considered.

These mappings can be exploited to make knowledge grow:

- by creating new links between resources so as to make one connected KG;
- by merging KGs into one larger KG.

4.3 Data Hosting Services

The *Data Hosting Services* will provide a reliable hosting service for the business graph. It will be based on the Ontotext S4 service suite⁷, which provides a semantic graph (triplestore) database-as-a-service that will be into and deployed as part of the DataGraft platform.

⁷ <https://console.s4.ontotext.com/>

The hosting task covers improvements of the scalability, performance and reliability of a large number of semantic graph databases (triplestores) running in the Cloud, so that large volumes of data can be managed and simultaneous queries and data access requests can be supported. The hosted datasets are accessible to third-party applications via various standard data access mechanisms: SPARQL query and Linked Data endpoints, as well as various RESTful APIs. Data may also be uploaded into the platform via standard read/write APIs for managing RDF data, such as the RDF4J API⁸.

The service components are:

- An RDF store that contains any RDF-ised datasets and provides SPARQL query access to the data. For the purpose of improving database availability and resilience, the RDF warehouse will not comprise a single database instance but a distributed set of database instances on a Cloud infrastructure;
- A standard SPARQL endpoint for querying data;
- RDF4J API for querying of RDF data. RDF4J provides light-weight industry standard way for querying and processing RDF data.
- A linked data endpoint, providing read-only access to RDF data;
- A Data Catalog Vocabulary (DCAT)⁹ based catalogue API providing simple metadata for dataset descriptions;
- A data exporting service, to access the dataset in original format.

4.4 Cross-Cutting Business Cases Analytics Services

The *Cross-Cutting Business Cases Analytics Services* provide a set of analytics services on top of the business graph data. These services are meant to be generic and customisable, and so will be reused in more than one business case. These services aim to simplify analytics tasks in the business cases, and will include deployment of machine learning techniques, statistical analysis and pattern detection on graph data, as well as information extraction and natural language processing techniques on unstructured data (news content).

The *Cross-Cutting Business Cases Analytics Services* will be based on JSI's **Event Registry**¹⁰, JSI's **Wikifier**¹¹, which will both be extended with new functionality during this project, and newly developed service for relation extraction. When the components of the business graph will be more precisely defined, an additional service for analytics on the graph will be added.

With these services we will cover:

1. semantic annotation of news and text related to companies
2. categorization of news into event types
(detection of event types like "Company A dissolved", ...)
3. information extraction about individual business events
(relation extraction)

Additionally, on the business graph, we will try to cover:

4. community/similarities detection
(clustering of companies based on some common criteria, finding connections with a top level (umbrella) company, ...)
5. "value" of the company, propagated via graph
(like pagerank for companies, based on predefined feature vectors for companies)

All data needed to run these services will reside locally, especially for the graph, because of specific requirements of used algorithms. All these services will be loosely coupled and available via API. Data

⁸ <http://docs.rdf4j.org/rest-api/>

⁹ <http://www.w3.org/TR/vocab-dcat/>

¹⁰ <http://eventregistry.org/>

¹¹ <http://wikifier.org/info.html>

needed for the annotation, event categorization and graph functionality will be inputted via API and the results of the services will be available via API.

When we will generate some new data (like new nodes in the graph), we could push the new data back to the main graph, which would need to have an API for that.

4.5 Marketplace and Operational Services

The *Marketplace and Operational Services* will ensure the availability of a data brokerage system in the form of a data marketplace where data that are part of the business graph can be provisioned and accessed. The focus here will be on the implementation of a mechanism for controlled access to business graph data, together with services for user management and data access mechanisms. In addition, the operational services needed for the marketplace will be addressed. Components for platform monitoring, availability, administration quota enforcement, branding and billing will be created. The underlying infrastructure of the marketplace will be based on DataGraft.

As stated in Section 3.3.2 we are exploring support for different models of sharing data. Current we are implementing a minimum viable product (MVP) supporting discovery and search of data made available via the euBusinessGraph Marketplace platform. The current version of the MVP provides an initial feature for a federated search and a feature to view and compare company data provided by different data providers. Further details of the MVP are found in Appendix D.

4.5.1 License Models

The *license models* sub-system is responsible for managing and enforcing data access based on license models and agreed agreements. The platform should support different license models such as:

- Free access to public open data;
- Payment models restricting access to data based on payments; and
- Dual license models, allowing payment plus share-alike for public-benefit use.

According to the requirements from the data providers, license models for full datasets are not sufficient, as there is a need to have different licenses models at the property level in the datasets.

4.5.2 Security and Access Control

The *security and access control* sub-system is responsible for ensuring the proper authorization and authentication for accessing all public service on the platform. As an additional security measure, all access to the platform services will utilize transport encryption (SSL/TLS). The components of the system include:

- Account management – all users will have personal accounts for accessing the euBusinessGraph platform that will be managed by this component;
- Authentication and authorisation – all access to the platform marketplace portal will be authenticated via a user name and password, while all access to the REST services exposed on the platform will be authenticated via private API key/secret pairs. Furthermore, datasets will offer different access levels: public, private, restricted. Appropriate authorisation measures will be taken to ensure that users have access only to appropriate data.

4.5.3 System Monitoring and Reporting

The *system monitoring and reporting* sub-system is responsible for monitoring all operations on the platform. Its components include:

- Service monitoring – for ensuring that services are operational and operate within expected performance levels (platform operators should be notified if services are down or operating with deteriorated performance);
- Usage monitoring and logging – all access to platform resources should be logged, so that various reports, billing information and audits may be prepared;
- Quota management – usage should be restricted within the predefined quota limits for the various services;

- Reporting – various system reports will be available to platform operators with daily/weekly/monthly statistics about platform usage. These reports will be generated based on input parameters and a set of APIs supported by the platform;
- Billing – this component is responsible for aggregating all the usage of a particular account on a monthly basis. In that way, based on predefined billing metrics such as volume of data accessed, number of queries, or number of datasets accessed, a usage cost can be calculated for the account and packaged as a monthly bill.

4.5.4 Platform Administration

The *platform administration* sub-system combines a set of tools for platform operators:

- Automation scripts that can be invoked by the platform operator in order to deploy/upgrade/restart a specific platform service (DevOps);
- Various GUIs providing administrative capabilities to platform components (e.g. GUI for RDF database management and configuration, cloud infrastructure management UIs, etc.).

5 API specification

This section describes the API specifications for the data preparation, data interlinking, data hosting, cross-cutting business cases analytics, and marketplace and operational services. The APIs, names and parameters are provisional and will evolve throughout the development of the euBusinessGraph platform. Inputs and outputs will be further refined and concretised as the platform API evolves.

5.1 Data Preparation Services

5.1.1 Data import, cleaning and transformations

The euBusinessGraph Marketplace platform will be implemented with the capability of generating, publishing and invoking executable transformation services. The API specification for the data workflows and publishing defines methods for transformation resources. Transformation services themselves will be capable of periodically triggering remote downloads, thus satisfying the requirement for continuous updates of the dataset.

Table 4: Data workflows and publishing APIs

Methods	Resource	Description
GET	/transformations/catalogue	Gets a list of transformations. <ul style="list-style-type: none"> Input: Desired serialization format (RDF or JSON-LD) of the response. This is applicable to most of the methods listed in the table. Output: List of transformation catalogue records (RDF or JSON-LD).
GET	/transformations/search	Searches for transformations (on metadata). <ul style="list-style-type: none"> Input: Query to search with in the transformations catalogue for one or more transformation(s) Output: List of transformation catalogue records.
GET, POST, PUT, DELETE	/transformations	GET retrieves a particular transformation description. <ul style="list-style-type: none"> Input: identifier of the transformation (taken from the catalogue) Output: Complete transformation. POST creates a new transformation. <ul style="list-style-type: none"> Input: Transformation metadata. Output: Identifier of the new transformation. The system will generate an identifier. PUT updates a transformation. <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> Transformation metadata. The code of the transformation. Output: HTTP result code. DELETE deletes a transformation. <ul style="list-style-type: none"> Input: Transformation identifier – identifier of the transformation to be removed. Output: HTTP result code.
GET, DELETE	/transformations/code	GET retrieves an executable transformation. <ul style="list-style-type: none"> Input: Transformation identifier – identifier of the transformation description. Output: The code of the executable transformation

		<p>DELETE deletes an executable transformation</p> <ul style="list-style-type: none"> Input: Transformation identifier – identifier of the transformation Output: HTTP result code.
POST	/transformations/execute	<p>Performs a data workflow, and loads the result in certain repository.</p> <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> Result file name – the name for the result file. Result type – mime type of the result. Data workflow identifier – the id of the specific data workflow to be used. Dataset – A file attachment of the input file. Output: Operation completion status and resulting data.
POST	/transformations/execute/rdf	<p>Performs a transformation to RDF and loads the result in certain repository.</p> <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> Repository graph – graph in the repository to store the result. RDF transformation identifier – the identifier of the specific RDF mapping transformation to be used. Input file – A file attachment of the input file. Output: Operation completion status or resulting RDF data as textual file.

5.2 Data Interlinking Services

The API specification for the data interlinking services defines methods for table, KG and document resources. We suppose that the table to be annotated is already available and accessible through a known identifier (ID) (if not, we can also provide an extra method to upload the table that returns as output the ID). We assume the same also for documents and KGs.

Table 5: Data Interlinking APIs

Methods	Resource	Description
GET, POST, PUT	/tables/<ID>/annotations/schema	<p>GET returns annotations at schema level for table <ID></p> <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> col_index (optional) – return annotation of the selected column (0-based) threshold (optional) – return annotation only if the confidence level is higher than threshold Output: the annotated schema <p>POST creates new annotation on selected column for table <ID></p> <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> col_index – the selected column annotation Output: HTTP result code <p>PUT updates annotation on selected column for table <ID></p>

		<ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> col_index – the selected column annotation Output: HTTP result code
GET	/tables/<ID>/annotations/instances	<p>Returns annotations at instance level for table <ID></p> <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> col_index (optional) – return annotation of the selected column (0-based) threshold (optional) – return annotation only if the confidence level is higher than threshold (value in [0,1]) Output: the annotated instances
GET	/tables/<ID>/annotations/instances/uncertain	<p>Returns instances with confidence degree lower than threshold for table <ID></p> <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> threshold – value in [0,1] Output: a list of the uncertain annotations
GET	/tables/<ID>/annotations/instance	<p>Return annotation for the given value wrt the specified context for table <ID></p> <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> row_index – the index of the row (1-based, 0 reserved to the header) col_index – the index of the column (0-based) context – it can be R (considering the row), C (considering the column), B (considering both the row and the column) or N (no context) Output: the annotated instance
GET	/KGs/<ID>/mappings	<p>Return a list of mappings between KG <ID> and a target KG</p> <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> target_kg (optional): a reference to the KG to be considered as target; if empty, the reference KG is used as target properties: a list of properties (declared with URI) to be used threshold (optional): set the minimum degree of confidence (value in [0,1]) needed to create a mapping Output: a list of mappings
GET	/documents/<ID>/links	<p>Return a list of target KG's entities that are mentioned in document <ID></p> <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> threshold (optional) – set the minimum degree of confidence (value in [0,1]) needed to select an entity language (optional) – the language of the document Output: a set of annotations

5.3 Data Hosting Services

5.3.1 Data hosting platform services

The API specification for the data hosting platform services defines methods for dataset and repository resources. We focus on the DCAT vocabulary APIs and the way that SPARQL endpoints are managed.

Table 6: Data hosting platform APIs

Methods	Resource	Description
GET	/datasets/catalogue	Gets a list of all datasets available to a user. <ul style="list-style-type: none"> Input: Serialization format (RDF or JSON-LD). This is applicable to most of the methods listed in the table. Output: List of catalogue records (RDF or JSON-LD) for the datasets.
GET	/datasets/search	Searches for datasets (on metadata). <ul style="list-style-type: none"> Input: Query to search with in the dataset catalogue for one or more dataset(s) Output: List of catalogue dataset records
GET, POST, PUT, DELETE	/datasets	Performs operations on dataset descriptions: GET retrieves a dataset description. <ul style="list-style-type: none"> Input: A dataset identifier taken from the catalogue. Output: A complete dataset description POST creates a dataset description. <ul style="list-style-type: none"> Output: Identifier of the new dataset in the format. The system will generate a new identifier automatically whenever the API is called. PUT updates a dataset. <ul style="list-style-type: none"> Input: Dataset description as RDF or JSON-LD. Output: Operation result indication (HTTP result code) DELETE deletes a dataset. <ul style="list-style-type: none"> Input: 'dataset-id' – URI of dataset to be removed. Output: Operation result indication (HTTP result code)
GET	/repositories	Gets a list of available repositories. <ul style="list-style-type: none"> Output: List of repository records with properties such as identifier, title, read- and write access parameters for each listed repository.
GET	/repositories/<ID>?query=...	Query a specific RDF database repository endpoint, identified by the input identifier, with a SPARQL query <ul style="list-style-type: none"> Input: SPARQL query to evaluate Output: Result of query
DELETE	/repositories/<ID>	Deletes a repository with a particular identifier. <ul style="list-style-type: none"> Output: Operation result indication (HTTP result code)
GET, POST, PUT, DELETE	/distributions	Perform operations on DCAT distributions. A distribution may contain the raw dataset uploaded to the platform or a transformed dataset available on the platform. The API is format-independent, meaning that it resolves the underlying file/database format based on the distribution data/metadata.

		<p>GET retrieves a distribution</p> <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> Distribution identifier – identifier of the distribution taken from the dataset description. Format – file format of the output Output: Full description of the desired distribution according to the DCAT vocabulary. <p>POST creates a distribution. Uses a multipart HTTP request with form parameters for the input.</p> <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> Metadata – metadata of the transformation according to the DCAT vocabulary. Raw file or endpoint reference representing the dataset. Output: Identifier of the new distribution. <p>PUT updates a distribution. Uses a multipart HTTP request with form parameters for the input.</p> <ul style="list-style-type: none"> Input (same as in POST): <ul style="list-style-type: none"> Metadata – metadata of the transformation according to the DCAT vocabulary. Raw file or endpoint reference representing the dataset. Output: Operation result indication (HTTP result code) <p>DELETE deletes a distribution</p> <ul style="list-style-type: none"> Input: Distribution identifier of the distribution to be deleted Output: Operation result indication (HTTP result code)
--	--	--

5.4 Cross-Cutting Business Cases Analytics Services

All services for Business Cases Analytics will have an REST API, with which data exchange will be possible. There will be separate APIs for:

- semantic multilingual annotation service
- Event Registry's events' detection
- graph based analytics:
 - relation extraction
 - clustering
 - similarities
 - connections between companies

Some of these APIs will require input and all will provide a response with results.

5.4.1 API for semantic multilingual annotation service

JSI's semantic multilingual annotation service will be based on JSI Wikifier. JSI Wikifier is a web service which takes a text document as input and annotates it with links to relevant Wikipedia concepts.

To use the JSI Wikifier, you would issue a HTTP GET request to the Wikifier URL in the form:

- <http://www.wikifier.org/annotate-article?text=...&lang=...&...>

Each request to the API must be accompanied with an API (user) key which you can obtain for free by registering on the Web site.

Parameters specified would typically include the text of the document that you want to annotate and language of the document. There exists a list of all the languages currently supported by the JSI Wikifier. One could also use the language parameter auto to let the system auto detect the language. Other parameters, if specified, would influence the amount and type of information returned by the JSI Wikifier.

The JSI Wikifier returns a JSON response of the following form:

```
{
  "annotations": [ ... ],
  "spaces":["", " ", " ", " ", "."],
  "words":["New", "York", "City"],
  "ranges": [ ... ]
}
```

The spaces and words arrays show how the input document has been split into words.

- annotations is an array of objects pointing to the relevant Wikipedia pages with additional information like title, url, language, ...
- ranges is an array of objects which support/extend the information in the annotations array.

5.4.2 API for Event Registry's events detection

Event Registry has a Python package which can be used to easily access the data available in the Event Registry through the provided API. Each request to the API must be accompanied with an API key which you can obtain for free by registering on the Web site (<http://eventregistry.org/register>).

The current API allows for querying of:

- **Events**
For searching for events that match various search criteria, such as relevant concepts, keywords, date, location or others.
- **Articles**
For searching for articles based on the publisher's URL, article date, mentioned concepts or others.
- **Trending concepts**
For finding concepts which are currently trending the most in the news.
- **Most shared articles and events on social media**
For finding the list of articles that have been shared the most on Facebook and Twitter on a particular date, or the most relevant event based on shares on social media.
- **Daily mentions and sentiment of concepts and categories**
For finding how often a particular concept or category was mentioned in the news in the previous years, or the sentiment expressed on social media about some person or event.

The API described below is based on Python package, which is easy to use. Python package itself is using a raw HTTP request protocol via JSON input formatted data, which is varying regarding the requested functionality and needed input parameters. This protocol won't be described here.

5.4.2.1 Searching for events

For searching for events two python classes are available: `QueryEvents` and `QueryEventsIter`.

`QueryEvents` class returns a list of events with complete information about the matching events in various forms, possibly including a timeline distribution of the matching events over time, distribution of matching events into predefined categories, list of top concepts in the matching events, etc.

`QueryEventsIter` class offers an iterator, with which you can easily iterate over all events that match the specified conditions. The information returned is similar to the `QueryEvents` class.

More details about these two classes are available on [Searching for events](#).

The returned information about events is a JSON formatted data with elements from the Event data model, which is explained in detail on [Event data model](#).

5.4.2.2 Searching for Articles

For search for articles two python classes are available: `QueryArticles` and `QueryArticlesIter`.

`QueryArticles` class returns a list of articles with complete information about the matching articles in various forms, possibly including a time distribution when articles were published, distribution of top news sources that wrote the matching articles, top concepts mentioned in the articles, etc.

`QueryArticlesIter` class offers an iterator, with which you can easily iterate over all articles that match the specified conditions. The information returned is similar to the `QueryArticles` class.

More details about these two classes are available on [Searching for articles](#).

The returned information about articles is a JSON formatted data with elements from the Article data model, which is explained in detail on [Article data model](#).

5.4.2.3 Trending concepts

A concept in Event Registry's terminology is an annotation which can be assigned to an article or event. Concepts can represent entities (people, locations, companies) or non-entities/keywords (things like phone, computer, cars, ...). A concept is associated with the article if it's mentioned in it or with event if it appears in the containing articles.

The concepts' trends in Event Registry are computed by comparing how many times a particular concept appears in the articles in the last two days compared to the last two weeks.

For getting a list of currently top trending concepts a python class `GetTrendingConcepts` is available.

More details about this are available on [Trends](#).

The returned information about top concepts is a JSON formatted data with elements from the Concept data model, which is explained in detail on [Concept data model](#).

5.4.2.4 Most shared articles and events on social media

For every article or event stored in Event Registry a lot of additional information (metadata) is stored. Among them is also a number how many times an individual article or event is being shared on social networks, such as Facebook or Twitter.

For getting this information you would use python class `QueryArticles`, setting the parameter `socialScore` in `ArticleInfoFlags` to `True`. The list of the resulting articles will then contain information about the number of shares of the article.

More details about this are available on [Social shares](#).

Based on the shares of the articles in social media Event Registry also computes a social score for events. A social score of an event is computed by selecting all articles assigned to the event and sorting them by decreasing social score. Maximum top 30 articles are then selected and an average social score of them is computed and used as the event's social score.

5.4.2.5 Daily mentions and sentiment of concepts and categories

For getting the information how often a particular concept is mentioned in the news or social media on a particular date you can use the python class `GetCounts`.

`GetCounts` class returns a JSON formatted data with a list of dates and number of mentions of matching concept on that date.

More details about this are available on [Number of mentions in news or social media](#).

5.4.3 API for graph based analytics

For any meaningful data analytics on the constructed graph the basic agreed data from the data providers should be made available to store offline. The business analytics services will operate on this data separately of other components and will offer functionality like the following:

- clustering

- similarities
- connections between companies, people and products/brands

The specific functionality will be based on use cases, for example:

1. Find out all companies who might be interested in a specific tender
The input to the service would be tag words describing tender. The service would then find all companies, who are dealing with the similar products/services as the specified tag words and return a list of such companies along with the basic data and pointers to the providers of the data.
2. Find out what companies are relevant to a specific product/service
The input to the service would be the description of the product/service. The service would return all companies dealing with or related to the specified product/service along with their basic data.
3. Find out all companies with similar product portfolio
The service would return all companies with similar products/services grouped in clusters along with their basic data.
4. Find the relationship of a specific person to companies if it exists
The input would be the person's name and the output would be the relations of this person to the companies (i.e. "Person A is a CEO of company B").
5. Find all companies connected to a specific company
The input would be the company's name and the output would be a list of companies connected in any way to a specified company along with their basic data.

The exact list of implemented functionality will be finalized as soon as all business case requirements are known and defined. The details of the API like the REST url, the input and the output format will be finalized in the next version of this document.

5.4.4 API for relation extraction

The relation extraction service will operate on local data separately of other components and will find out the relations between entities (companies, people and products/brands). It will generate additional information for the graph which could be pushed back to the main graph, which would need to have an API for that. This additional information will then serve as input in some of the graph based analytics.

The API will be defined in the next version of this document.

5.5 Marketplace and Operational Services

5.5.1 Security and access control

The API specification for security and access control defines methods for account and API key resources. If an error occurs, a 'message' containing an explanation of what went wrong (e.g., "incorrect username or password", "expired session", etc.) will be included in the resulting output.

Table 7: Security and access control APIs

Methods	Resource	Description
PUT	/accounts/login	<p>Performs log-in with username and password or uses a pre-existing social network ID (Google+ ID, Twitter ID or Facebook ID).</p> <ul style="list-style-type: none"> • Input: <ul style="list-style-type: none"> ○ 'username' – username. ○ 'password' – password. ○ 'google_id' – Google+ ID (alternative). ○ 'twitter_id' – Twitter ID (alternative).

		<ul style="list-style-type: none"> ◦ 'facebook_id' – Facebook ID (alternative). • Output: HTTP result code (with 'message').
POST	/accounts	<p>Creates a new account with username and password or associates a pre-existing one (Google+ ID, Twitter ID or Facebook ID).</p> <ul style="list-style-type: none"> • Input: <ul style="list-style-type: none"> ◦ 'username' – user name. ◦ 'password' – password. ◦ 'google_id' – Google+ ID (alternative). ◦ 'twitter_id' – Twitter ID (alternative). ◦ 'facebook_id' – Facebook ID (alternative). ◦ 'role' – user role. ◦ 'name' – full name of user. ◦ 'email' – email address of user. • Output: HTTP result code (with 'message').
PUT	/accounts/logout	<p>Performs log-out.</p> <ul style="list-style-type: none"> • Output: HTTP result code (with 'message').
GET	/accounts/login_status	<p>Gets login status.</p> <ul style="list-style-type: none"> • Output: 'status' – "authenticated" or "not authenticated"
GET, PUT	/account/details	<p>GET retrieves account details.</p> <ul style="list-style-type: none"> • Output: Account details with username, email, name, role, phone, address, etc. <p>PUT updates account details:</p> <ul style="list-style-type: none"> • Input: <ul style="list-style-type: none"> ◦ 'username' – user name. ◦ 'password' – password. ◦ 'role' – user role. ◦ 'name' – full name of user. ◦ 'email' – email address of user. ◦ 'phone' – phone number ◦ 'address' – address • Output: HTTP result code (with 'message').
PUT	/accounts/password	<p>Changes the password.</p> <ul style="list-style-type: none"> • Input: 'new_password' – new password. • Output: HTTP result code (with 'message').
POST	/accounts/password/reset	<p>Requests a password reset.</p> <ul style="list-style-type: none"> • Input: 'email' – email address. • Output: HTTP result code (with 'message').
PUT	/accounts/password/confirm	<p>Confirms a password reset request.</p>

		<ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> 'email' – email address. 'new_password' – new password. 'token' – verification token. Output: HTTP result code (with 'message').
GET, POST	/api_keys/	<p>GET retrieves API Keys.</p> <ul style="list-style-type: none"> Output: List of API keys <p>POST creates new API key</p> <ul style="list-style-type: none"> Output: New API key record <ul style="list-style-type: none"> 'api_key' – key id 'secret' – secret phrase
POST	/api_keys/temporary	<p>Creates a temporary API Key (expires after 24h).</p> <ul style="list-style-type: none"> Output: New API key record <ul style="list-style-type: none"> 'api_key' – key id 'secret' – secret phrase
PUT	/api_keys/<api_key>/enable	<p>Enables an API Key.</p> <ul style="list-style-type: none"> Output: HTTP result code (with 'message').
PUT	/api_keys/<api_key>/disable	<p>Disables an API Key.</p> <ul style="list-style-type: none"> Output: HTTP result code (with 'message').
DELETE	/api_keys/<api_key>	<p>Deletes an API Key.</p> <ul style="list-style-type: none"> Output: HTTP result code (with 'message').

5.5.2 Usage reporting

The API for usage reporting provides a method that provides usage reports of the platform resources (e.g., number of requests, incoming/outgoing traffic) that are automatically logged by the euBusinessGraph platform.

Table 8: Usage reporting APIs

Methods	Resource	Description
GET	/usage	<p>Gets the usage report in the given period.</p> <ul style="list-style-type: none"> Input: <ul style="list-style-type: none"> 'from' – from date. 'to' – to date. 'resources' – list of resource types (or all) that should be included in the report. Output: Usage report with user information (user ID, name, email) and usage records (start time, end time, number of requests, incoming traffic, outgoing traffic, SPARQL requests, SPARQL traffic, transformation requests, transformation traffic, etc.)

6 Conclusions

This document provides an overview of the euBusinessGraph Marketplace platform, introducing the relevant stakeholders of the platform, and outlining a set of requirements for the platform from the perspectives of the stakeholders.

Furthermore, the document provides an initial architecture design for the platform, together with a preliminary set of APIs for the components of the architecture. The architecture, components, and APIs will evolve during the course of the project, as the business cases will become more mature.

The requirements, architecture, and APIs outlined in the document will serve as input for the implementation of the platform in the next phase.

It should be noted that the technical results in the project should not be to solve the individual business cases, but focus on making the business graph and its services as smart and useful as possible to the business cases. The different business cases are different in nature, but should represent a diverse set of needs and requirements that can be used as input for the technical solutions to be developed in the project.

Appendix A Requirements matrix

Description		OCORP	CERVED	SDATI	EVERY	DW	BRC	JSI
Stakeholder role (DP=Data Provider, DC=Data Consumer)		DP	DP, DC	DP, DC	DC	DP, DC	DP	DP
Data preparation services								
Dataset Import	Support for file formats (CSV, JSON, XML, RDF) and REST service	JSON, REST APIs	CSV, JSON, REST APIs	CSV, REST API	-	JSON, REST API	CSV, JSON, XML, RDF	?
Data Cleaning & Transformation	Support for data cleaning & transformation	✓	✓	✓	-	✓	✓	?
RDF-ization	Mapping to RDF	✓	✓	✓	-	✓	✓	?
Data Interlinking Services								
Named entity linking	Support for named entity linking	?	?	?	-	✓	?	-
Semantic labelling	Support for semantic labelling	✓	✓	✓	-	✓	?	-
Link discovery	Support for link discovery	✓	✓	✓	-	✓	?	-
Data Hosting Services								
Data queries	Support for data queries	✓	✓	✓	✓	✓	✓	✓
Company data search/discovery	Search for (type) of company data available	✓	✓	✓	✓	extended company profile	✓	✓
Faceted search	Faceted and filters	-	✓	✓	✓	brands	-	-
Data access	Support for data access	✓	✓	✓	✓	✓	✓	✓
SPARQL endpoint	Data can be queried through a SPARQL endpoint	-	-	-	-	-	✓	-
Data dump	Data dump available for download/transfer	✓	✓	✓	✓	?	✓	✓
REST service	Data is available through RESTful web services	✓	✓	✓	✓	✓	✓	✓
Data updates	Support for data updates	✓	✓	✓	-	✓	✓	✓
Frequency	Frequency of updates (Y=Yearly, M=Monthly, W=Weekly, D=Daily, V=Variable, C=Continuously)	D, V	D	W	-	D	C	M, Y
Incremental update	API for incremental update	✓	✓	✓	-	✓	✓	✓
Bulk update	API for bulk update	✓	✓	✓	-	-	-	-
Metadata	Support for metadata	✓	✓	✓	-	✓	✓	✓

Spatial coverage	Jurisdictions (e.g. countries) covered by the dataset (UK=United Kingdom, IT=Italy, N=Norway, R=Russia)	UK	IT	UK, IT, R	-	European, Global	N	✓
Data store (RDF)	Support for big data store	✓	✓	✓	-	✓	✓	✓
Dataset size	Size of dataset	1.3 billion records (1TB)	GBs per month	GBs per month	-	35.000 articles per year per language	1 million entities	?
Cross-Cutting Business Case Analytics Services								
Multi-lingual annotation	Support for multi-lingual annotation	✓	✓	✓	✓	✓	✓	✓
Language	Language of the dataset ¹² (en=English, it=Italian, no=Norwegian, ru=Russian)	en	it	en	✓	articles in 30 different languages	no	✓
Events detection	Support for events detection	?	?	?	-	✓	?	-
Graph analytics	Support for graph based analysis	?	?	?	-	✓	?	-
Relation Extraction	Support for extracting relations between entities	?	?	?	-	✓	?	-
Marketplace and Operational Services								
License Models	Dataset license ¹³ (F=Free, P=Payment, D=Dual (payment plus share-alike for public-benefit use))	D	P	P	P	P	F	F
Open data	Data available as open data	✓	-	-	-	-	NLOD	-
Dataset-level access models	Access models to datasets (P=Public, M=Matching only, R=Restricted to license model)	P, R	M, R	M, R	-	R	P	-
Data property-level access model	Access models to data properties (P=Public, M=Matching only, R=Restricted)	P, R	M, R	M, R	-	R	P	-
Security & Access Control	Support for authentication and user management	✓	✓	✓	✓	✓	✓	✓

¹² <http://data.okfn.org/data/core/language-codes>

¹³ https://en.wikipedia.org/wiki/Creative_Commons_license

System Monitoring & Reporting	Support for monitoring and reporting	✓	✓	✓	✓	✓	✓	✓
Platform Administration	Support for platform administration	✓	✓	✓	✓	✓	✓	✓

Appendix B Requirements for common data model

Business Case Partner (Business Case)	Stakeholder role (DP=Data Provider, DC=Data Consumer)	Common data model properties	Other requirements (related to the euBusinessGraph infrastructure)
OCORP	DP	-	<ul style="list-style-type: none"> Data access: <ul style="list-style-type: none"> OC API or bulk, as well as euBG API to public interest, not commercial use, journalism, NGO, etc All OC data searchable through euBG, with links to OC to euBG partners or external third parties for remuneration
CERVED	DP, DC	-	<ul style="list-style-type: none"> Similar to SDATI
SDATI	DP, DC	-	<ul style="list-style-type: none"> Bulk access to UK and NOR data Handle trustworthiness/fuzziness of the information. Atoka has different business cases that require different degree of trust, for example, merge/acquisition is reflected both in the news and in Companies Register, but in the news it happens before, hence, it is interesting for the kind of users for whom knowing this information asap is more important than trustworthiness of the information. This especially relevant for non-authoritative sources and the users for whom credibility of the information is important. Capture different types of events, e.g., merge, acquisition, etc, from different sources: <ul style="list-style-type: none"> mentions of ppl/companies in the news events in the news: news article being about a merger or acquisition or a product launch events on a company's website: new company website, a company website has changed its e-commerce technology, there's a new link on the corp. events coming from auth sources: a company has been awarded a new tender, a company has changed its trading address, company shares info has changed
EVRY (CRM-S)	DC	-	<ul style="list-style-type: none"> Have data stored and synced in their DB Store a subset of the data in their DB (this data will be used to train a model that will indicate actual area of business, related business and events that indicate potential or risk) The subset of the data is the same as DW defined for their advanced search To get access to this subset -- a licensing scheme implemented in euBG or a common agreement between the partners. If not they will start working directly with the data providers. BRC will be the first one.
DW (JDP)	DP, DC	<ul style="list-style-type: none"> Advanced search by attributes: <ul style="list-style-type: none"> Name Company type Country Jurisdiction code Address Key Manager Profit/Loss Turnover Tax Paid Number of employees Website Wikipedia URL VAT 	<ul style="list-style-type: none"> Resolve companies' names ambiguity Brand recognition? Information about data that can be found in other data providers: <ul style="list-style-type: none"> what fields and their meaning free or paid? registration or OAuth? API/web? Payment system used?

		<ul style="list-style-type: none"> ○ Certified e-mails ○ Other locations ○ RSS/Atom feeds ○ Web languages ○ Publicly traded? 	
BRC	DP	<ul style="list-style-type: none"> ● Alignment to core vocabularies ● Describing all their data through a common vocabulary (DCAT-AP) - see pilot http://portal-fdk.tt1.brreg.no/?lang=en 	<ul style="list-style-type: none"> ● Data access: BRC RESTful API or the euBG API; Linked Data URIs, bulk download will be available, but not a preferred options as data in bulk might be outdated ● Not linking to non-authoritative data sources ● Interlinking with public authoritative data, e.g., basic company data and yearly accounts

Appendix C Initial list of systems of identifiers

Identifier name	Brief description	Example	More information	OCORP	CERVED	SDATI	EVERY	DW	BRC
NO orgnr	Organization number (eleven digits) for all entities in Norway (e.g. legal entity and branches)	974760673	Manually look up data at (do not use for scraping) https://brreg.no/home/ (english). Open data service at http://data.brreg.no/oppslag/enhetsregisteret/enheter.x.html . Each entity have their own restful URI http://data.brreg.no/enhetsregisteret/enhet/974760673.xml (or .json or .csv)	✓	✓	✓	✓	✓	✓
News Id	URI which uniquely identifies news article	3403979	https://github.com/EventRegistry/event-registry-python/wiki/Data-models					✓	
Events Id	URI which uniquely identifies event	eng-4635724	https://github.com/EventRegistry/event-registry-python/wiki/Data-models					✓	
Categories Id	URI which uniquely identifies news'/events' category (might point to Wikipedia/Wikidata)	dmoz/Society/Issues/Warfare_and_Conflict	https://github.com/EventRegistry/event-registry-python/wiki/Data-models	✓				✓	
OCORP Company id in multiple jurisdictions	Company numbers (aka Business Register IDs) issued by over 100 different registers	gb/7444723	List of all the registers we hold company numbers for is at https://opencorporates.com/registers	✓	✓	✓			
LEI	LEI issued by Global Legal Entity Identifier Foundation	549300MC3GOXY2ACZD66	https://www.gleif.org	✓	✓	✓			
US Federal EIN (TIN)	US Federal Employee Identification Number	47-1406623	http://www.irs.gov/businesses/small/article/0,,id=98350,00.html	✓				✓	
Greece TIN	Tax Identification Number (Αριθμός Φορολογικού Μητρώου - ΑΦΜ)		https://ec.europa.eu/taxation_customs/tin/pdf/en/TIN_-_country_sheet_EL_en.pdf					✓	
News ID	URI which uniquely identifies news article	37748254	http://www.dw.com/api/detail/article/37748254					✓	
Category ID	URI which uniquely identifies news'/events' category (pay attention: not used perfectly systematic)	"categoryName" : "Kultur"	http://www.dw.com/api/detail/article/37748254					✓	
EU VAT number	standardised VAT number in EU	GB164331133	http://ec.europa.eu/taxation_customs/vies/	✓	✓	✓		✓	
Montenegro TIN	Montenegro Tax Identification Number (Poreski identifikacioni broj - PIB)		http://www.crps.me/	✓					
US SEC CIK	Central Index Key issued by SEC to filing entities		http://www.sec.gov/edgar/searchedgar/cik.htm	✓					
England & Wales Charity Number	Issued by Charity Commission for England & Wales		http://www.charitycommission.gov.uk/	✓					

Swiss Federal Statistical Office Enterprise Identification Number			https://www.uid.admin.ch/	✓					
France National Associations Register Identifier			http://www.associations.gouv.fr/le-rna-repertoire-national-des-associations.html	✓					
UK FCA Mutuals Register Number			https://www.fca.org.uk/	✓					
Atoka ID	ID for companies, locations, people within Atoka	"6da785b3a df2"	https://developers.atoka.io/v2/companies.html		✓	✓			
Italian Tax Code	ID issued by the Italian Revenue Agency (Agenzia delle entrate) to identify citizens and entities	"022418902 23"	http://www1.agenziaentrate.gov.it/english/italian_taxation/tax_code.htm		✓	✓			
CCIAA + REA	REA: Repertorio Economico Amministrativo. Number assigned by the chamber of commerce (per province where there is at least one location of a company) CCIAA: Camera di Commercio, Industria, Artigianato e Agricoltura. ID of the chamber of commerce that issues the REA in that province a pair REA - CCIAA - REA can be used to identify a company. Companies can have more than one CCIAA - REA, since they can be registered to more than one chamber of commerce	TN210089	(in italian) https://help.infocert.it/risposte/cosa-sono-le-sigle-cciaa-rea/		✓	✓			
RUI	ID handed by Italy's registry of insurance and reinsurance intermediaries	A000101292	https://www.ivass.it/operatori/intermediari/rui/index.html?com_dotmarketing.htmlpage.language=3		✓	✓			
CONI	ID for sports organisations handed by Italian National Olympic Committee	268397	http://www.coni.it/en/coni-eng.html		✓	✓			
IPA	ID for organisations in the Italian Public Sector	ardsu_to	(in italian) http://www.indicepa.gov.it/documentale/n-opensdata.php		✓	✓			

Appendix D Minimum Viable Product (MVP)

In euBusinessGraph, we are developing a Minimum Viable Product (MVP) to provide end users, via a graphical user interface, core features supported by the euBusinessGraph Marketplace platform outlined in this deliverable. The MVP is a web-based application developed in Ruby on Rails¹⁴.

The current version of the MVP provides an initial feature for a federated search and a feature to view and compare company data provided by different data providers. In the following, we first provide an overview of the current version of the MVP, and then we explain how we plan to improve the current version in order to facilitate the envisioned features of the euBusinessGraph Marketplace platform. However, the latter is explained at a high level of abstraction and does not cover detailed technical aspects. This is because the technical details of the MVP are evolving during the course of the project, in line with the architecture, components, and the APIs of the euBusinessGraph Marketplace platform.

D.1 Current version of the MVP

As mentioned above, the current MVP version supports two main features: federated search and view/compare information about specific companies.

D.1.1 Federated search

Figure 10 shows a screen shot of the federated search in which we have searched for the company name "lipton". The search feature also has filtering capabilities, but is currently filtering only on selected countries (Norway, Italy, Great Britain). In the example in Figure 10, we have restricted the search to Great Britain.

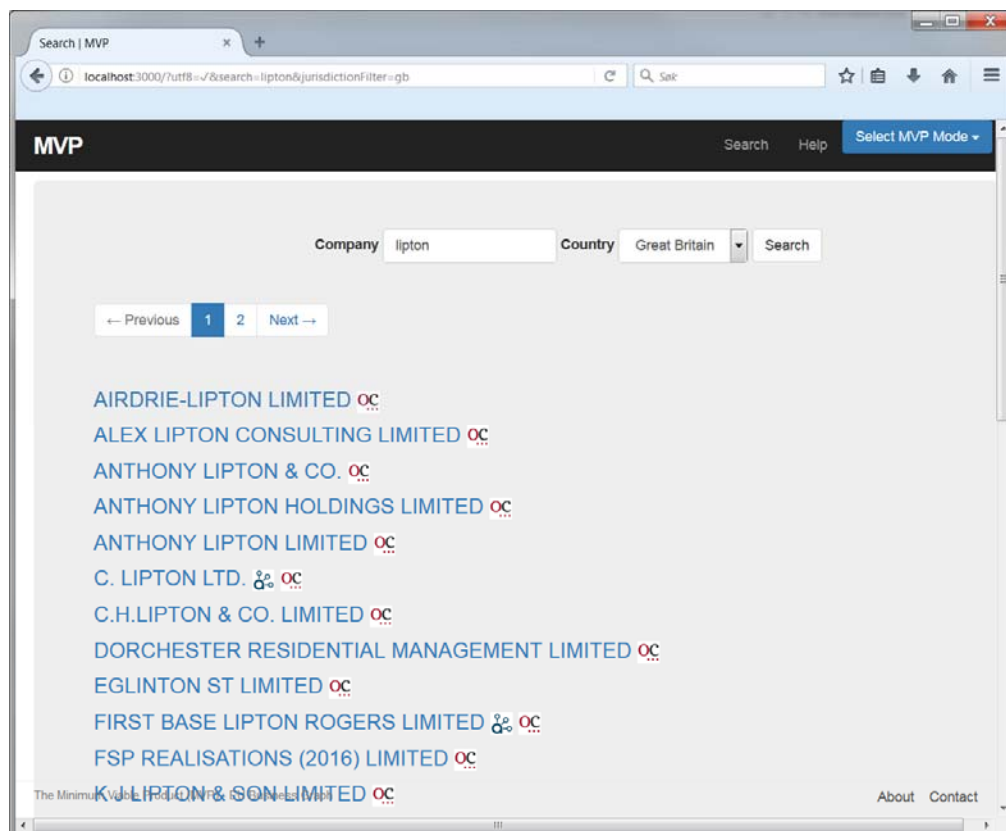


Figure 10: Screen shot of the federated search

Figure 11 illustrates a conceptual model of the federated search. In the following, we explain how the search feature works using the search result in Figure 10 as an example. First, a user types in a company name in the Company field of the search form ("lipton" in the above example), selects which

¹⁴ <http://rubyonrails.org/>

country to filter on ("Great Britain"), and finally presses the search button on the search form. The country filter is optional.

These input values are passed to a class (Search) which assembles an API search query based on the Company name and the Country filter, and then executes the search by passing the search query to the data providers. Notice that, currently, the MVP executes queries only on the Open Corporates API and the Atoka API.

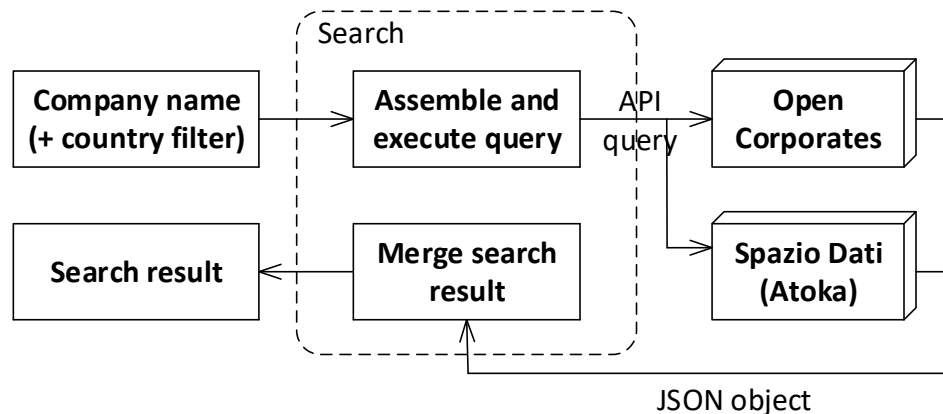


Figure 11: Federated search

Having executed the search queries, the data providers respond with a JSON object containing the search result. The Search class captures and merges the responses into a single search result. Figure 10 shows the merged search result from searching on "lipton" restricted to "Great Britain". Notice that one or two icons follow each company name in the search result, where each icon represent a data provider. For example, for search result "C. LIPTON LTD.", we see that both Open Corporates and Atoka may provide information about the company.

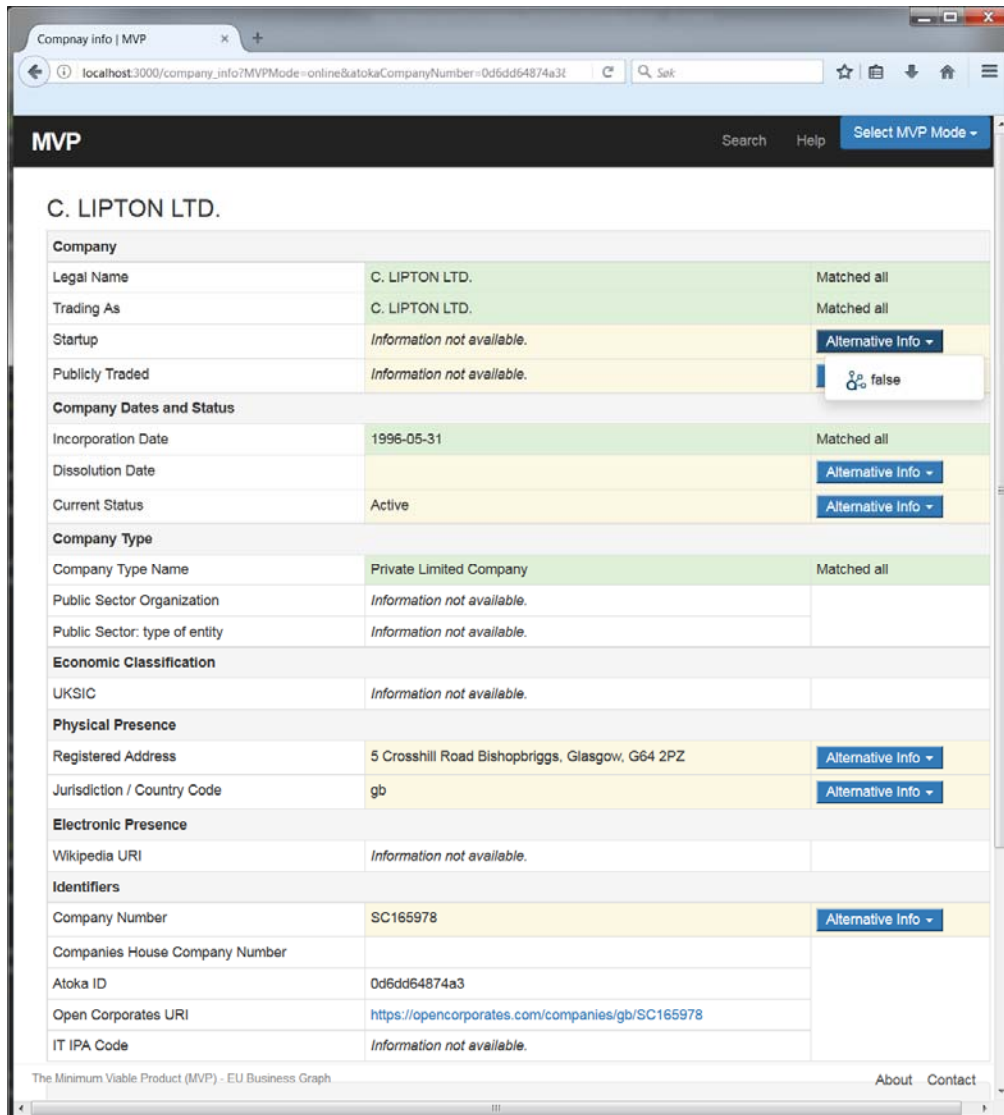
D.1.2 View and compare information about a specific company

In order to retrieve information about a specific company, the user has to click on a company name in the search result. Assume we click on the company "C. LIPTON LTD." in Figure 10. The result is shown in Figure 12. It is beyond this deliverable to go into the details of the company information given in Figure 12. In the following, we therefore first explain the layout of the information for a specific company as represented in Figure 12, and then we explain the corresponding conceptual model for retrieving company info (Figure 13).

The first column in Figure 12 lists the properties of a company. A property is a type of information about a company. For example, the legal name of the company, the public identifier of the company, etc. This is a preliminary list of properties (at the time of writing) and will be updated as soon as the euBusinessGraph ontology model is defined.

The second column shows available data for each property. The idea is that the information shown in the second column is retrieved from a "base data provider" (which may be user defined), and used to compare against information retrieved from all other data providers about the same company. In this example, the "base data provider" is Open Corporates. That is, the name "C. LIPTON LTD." in the second column is retrieved from Open Corporates. This is compared to information retrieved from "all other" data providers, which in this case is only Spazio Dati (Atoka). The row Legal Name is highlighted green because the legal name for company "C. LIPTON LTD." is identical in all data providers (Open Corporates and Spazio Dati).

Whenever a row is highlighted yellow, the corresponding information is either missing, or it is different from the information retrieved from other data providers. For example, we see that the row Startup is highlighted yellow including the text "Information not available". This means that Open Corporates do not have this information. The user may then click on the "drop-down" button on the third column to see whether other data providers provide this information. We see from Figure 12 that Atoka provides this information and the information is "false" meaning that "C. LIPTON LTD." is not a startup company.



The screenshot shows a web browser window with the URL `localhost:3000/company_info?MVPMode=online&atokaCompanyNumber=0d6dd64874a3f`. The page title is "Company info | MVP". The main content area displays information for "C. LIPTON LTD." under the heading "MVP". The information is organized into several sections:

- Company**
 - Legal Name: C. LIPTON LTD. (Matched all)
 - Trading As: C. LIPTON LTD. (Matched all)
 - Startup: Information not available. (Alternative Info)
 - Publicly Traded: Information not available. (false)
- Company Dates and Status**
 - Incorporation Date: 1996-05-31 (Matched all)
 - Dissolution Date: (Alternative Info)
 - Current Status: Active (Alternative Info)
- Company Type**
 - Company Type Name: Private Limited Company (Matched all)
 - Public Sector Organization: Information not available.
 - Public Sector: type of entity: Information not available.
- Economic Classification**
 - UKSIC: Information not available.
- Physical Presence**
 - Registered Address: 5 Crosshill Road Bishopbriggs, Glasgow, G64 2PZ (Alternative Info)
 - Jurisdiction / Country Code: gb (Alternative Info)
- Electronic Presence**
 - Wikipedia URI: Information not available.
- Identifiers**
 - Company Number: SC165978 (Alternative Info)
 - Companies House Company Number: (empty)
 - Atoka ID: 0d6dd64874a3
 - Open Corporates URI: <https://opencorporates.com/companies/gb/SC165978>
 - IT IPA Code: Information not available.

The footer of the page reads "The Minimum Viable Product (MVP) - EU Business Graph" and includes links for "About" and "Contact".

Figure 12: Information about one specific company (in this case C. LIPTON LTD.)

Figure 13 illustrates a conceptual model of the MVP's feature to retrieve and display information about one specific company as illustrated in Figure 12.

As mentioned initially, a user has to click the name of a company to retrieve information about that company. Each hyperlink in Figure 10 contains the unique company ID as defined in the databases of the data providers (Open Corporates and Spazio Dati). These unique IDs are then passed to the Search class that assembles and executed an API query.

The data providers respond with a JSON object containing information about the specific company. The Search class retrieves the JSON object from each data provider and stores the JSON object retrieve from Open Corporates as the "base" information. Then, for each property in the base information, the Search class checks for whether the information exists, is equal to, or is different from the corresponding information retrieved from all other data providers. Finally, the information is displayed to the user as represented in Figure 12. Notice that the choice of using Open Corporates is not intentional, but only used as an initial "base" as part of the development of the MVP.

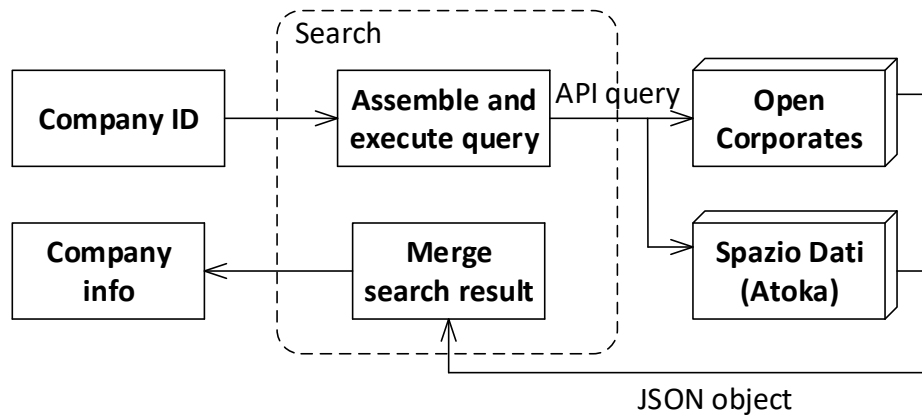


Figure 13: Retrieve and view company info

D.2 What we plan to do for the future versions of the MVP

As pointed out above, the technical details of the MVP are evolving during the course of the project, in line with the architecture, components, and the APIs of the euBusinessGraph Marketplace platform. It is therefore not possible to foresee all possible improvements of the MVP at the time of writing. However, in the following, we list the improvements we plan to implement.

With respect to the federated search, the current version of the MVP is executing search on different APIs provided by different data providers. This approach will not scale in the end. As mentioned in Section 3.3.2, for the first release of the euBusinessGraph platform, we are therefore implementing a common data model that all data providers can use to share a common data subset. For data that data providers are unwilling to share, we may use pointers to their respective source.

With respect to viewing company information, the current version of the MVP is limited to show whether the underlying information about a company exists, is equal to, or is different from the information provided by other data providers. We plan to extend this feature to support other useful data analytics.