Innovation Action (IA)

ICT-14-2016-2017
H2020-ICT-2016-1

Enabling the European Business Graph for Innovative Data Products and Services

Deliverable D3.3:

Requirements Analysis, Architecture and API Specification for the euBusinessGraph Marketplace – v2

Date	15.08.2019
Author(s)	Brian Elvesæter (SINTEF), Bjørn Marius von Zernichow (SINTEF), Dumitru Roman (SINTEF), Ahmet Soylu (SINTEF), Philip Turk (SINTEF), Boyan Simeonov (Ontotext), Stancho Yordanov (Ontotext), Milena Yankova (Ontotext), Matteo Palmonari (UNIMIB), Vincenzo Cutrona (UNIMIB), Flavio De Paoli (UNIMIB), Matjaz Rihtar (JSI), Inna Koval (JSI), Miha Jenko (JSI)
Dissemination level	Public (PU)
Work package	WP3
Version	2.0

Document metadata

Quality assurors and contributors

Quality assuror(s)	Javier Paniagua (SDATI), Arnt-Henning Moberg (EVRY)
Contributor(s)	euBusinessGraph Consortium

Version history

Version	Date	Description
0.1	19.02.2018	Initial Table of Contents (ToC).
0.2	23.02.2018	Updated structure after WP3 concall.
0.3	12.03.2018	Updated the introduction and overview to align with the business model for the marketplace.
0.4	16.03.2018	Updated requirements analysis and architecture specification.
0.5	26.03.2018	Integrated input from WP3 partners.
0.6	27.03.2018	Draft ready for internal peer review.
1.0	28.03.2018	Final formatting and layout.
1.1	07.06.2019	Prepared updated draft to accommodate updated contributions from technical partners. The deliverable has been restructured in order to better align with deliverable D3.4.
1.2	18.06.2019	Integrated contributions from technical partners.
1.3	12.08.2019	Updated version addressing feedback from internal reviewers.
2.0	15.08.2019	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 technologies such as Data-as-a-Service and Linked Data.

The project will deliver a technology platform, the **euBusinessGraph Marketplace**, which 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.

This report focuses on the **euBusinessGraph Marketplace** and presents:

- An updated **requirements analysis** for the **euBusinessGraph Marketplace**, the relevant stakeholders of the marketplace and their requirements in terms of capabilities (tools and services) of the marketplace aligned with requirements from the business cases in WP4 and the initial business model for the euBusinessGraph Marketplace created in WP5;
- An updated **architecture and API specification** for the **euBusinessGraph Marketplace** in terms of the core components of the marketplace platform, and a set of APIs that will guide the development of the marketplace platform in the final phase of the project.

Table of Contents

1 Introduction

This report presents Deliverable D3.3 "Requirements Analysis, Architecture and API Specification for the euBusinessGraph Marketplace – v2" of the euBusinessGraph project. This deliverable is developed as part of Work Package 3 (WP3) "euBusinessGraph Provisioning" and provides an updated version of Deliverable D3.1 based on further analysis of requirements from the business cases in WP4 and the initial business model for the euBusinessGraph Marketplace created in WP5.

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 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. The tools and approaches will simplify the business graph data publication and consumption process, and analytics tasks on top of the business graph.

The aim of this deliverable is two-fold:

1. To provide an updated **requirements analysis** for the **euBusinessGraph Marketplace**, the relevant stakeholders of the marketplace and their requirements in terms of capabilities (tools and services) of the marketplace aligned with requirements from the business cases in WP4 and the initial business model for the euBusinessGraph Marketplace created in WP5.
2. To provide an updated **architecture and API specification** for the **euBusinessGraph Marketplace** in terms of the core components of the marketplace platform, and a set of APIs that will guide the development of the marketplace platform in the final phase of the project.

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 and Services** in month 12 and month 24 of the project as illustrated in Figure 1 below.

- Deliverable D3.1 (month 6) served as the specification for Deliverable D3.2 (month 12). The 1st version of the euBusinessGraph Marketplace and Services was based on initial requirements analysis from the business cases in WP4 (Deliverable D4.1), as well as some initial input from WP1 and WP2;
- Deliverable D3.3 (month 15) provides a revised specification for the 2nd version of the euBusinessGraph Marketplace and Services to be finalized in Deliverable D3.4 (month 24). This revised specification is based on the business case requirements (from Deliverable D4.1), dataset descriptions (from Deliverable D1.1), the design of the company model and identifier system (from Deliverable D2.1), and the initial business model for the marketplace (from Deliverable D5.2).

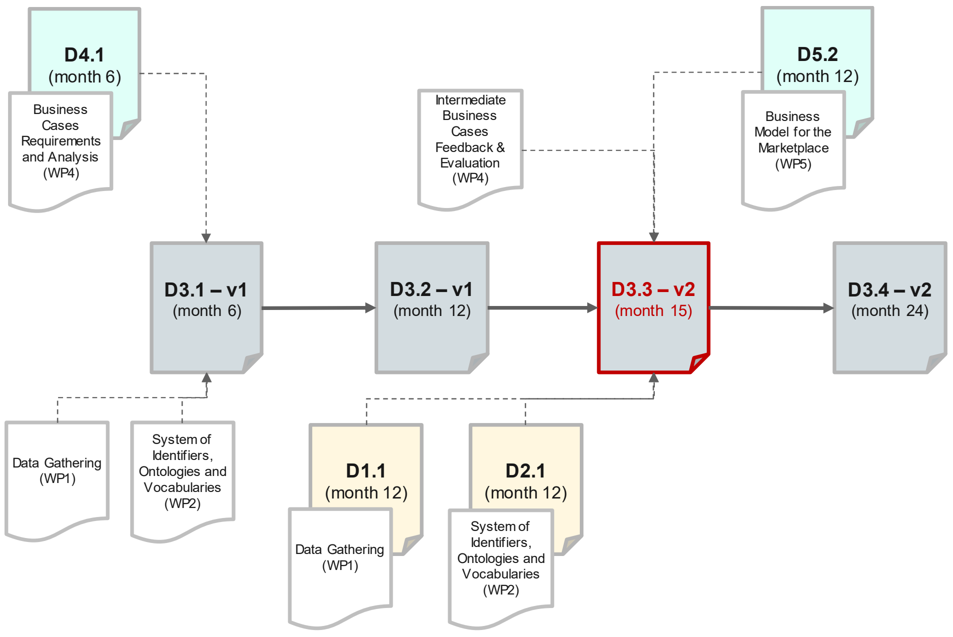

D4.1
(month 6)
Business Cases Requirements and Analysis (WP4)
Intermediate Business Cases Feedback & Evaluation (WP4)
D5.2
(month 12)
Business Model for the Marketplace (WP5)
D3.1 – v1
(month 6)
D3.2 – v1
(month 12)
D3.3 – v2
(month 15)
D3.4 – v2
(month 24)
Data Gathering (WP1)
System of Identifiers, Ontologies and Vocabularies (WP2)
D1.1
(month 12)
Data Gathering (WP1)
D2.1
(month 12)
System of Identifiers, Ontologies and Vocabularies (WP2)

Figure 1: Relationship to other work packages and deliverables

1.3 Document structure

The remainder of this report is structured as follows.

- Section 2 provides an overview of the euBusinessGraph Marketplace and the relevant customer segments identified in the business model for the marketplace. Next it outlines a revised architecture and API specification for the **euBusinessGraph Marketplace and Services** introducing new service categories. Finally, it presents an updated requirements analysis for the euBusinessGraph Marketplace structured according to the customer segments and the new service categories;
- Section 3 presents the development status and plans for the **Marketplace and data hosting services**, which (1) 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, and (2) provide a reliable hosting service for the business graph;
- Section 4 presents the development status and plans for the **Data ingestion services**, which provide (1) data preparation services that simplify data cleaning and transformation processes ensuring the availability of services to help with the generation of the business graph data, and (2) data interlinking services that provide necessary RDF-isation services for data transformation, enrichment and interlinking and metadata generation processes so that the business graph data can be published as Linked Data;
- Section 5 presents the development status and plans for the **Cross-cutting business analytics services**, which provide a set of analytics services on top of the business graph data;
- Finally, Section 6 concludes this report and provides a summary of the development plans for the next period of the project.

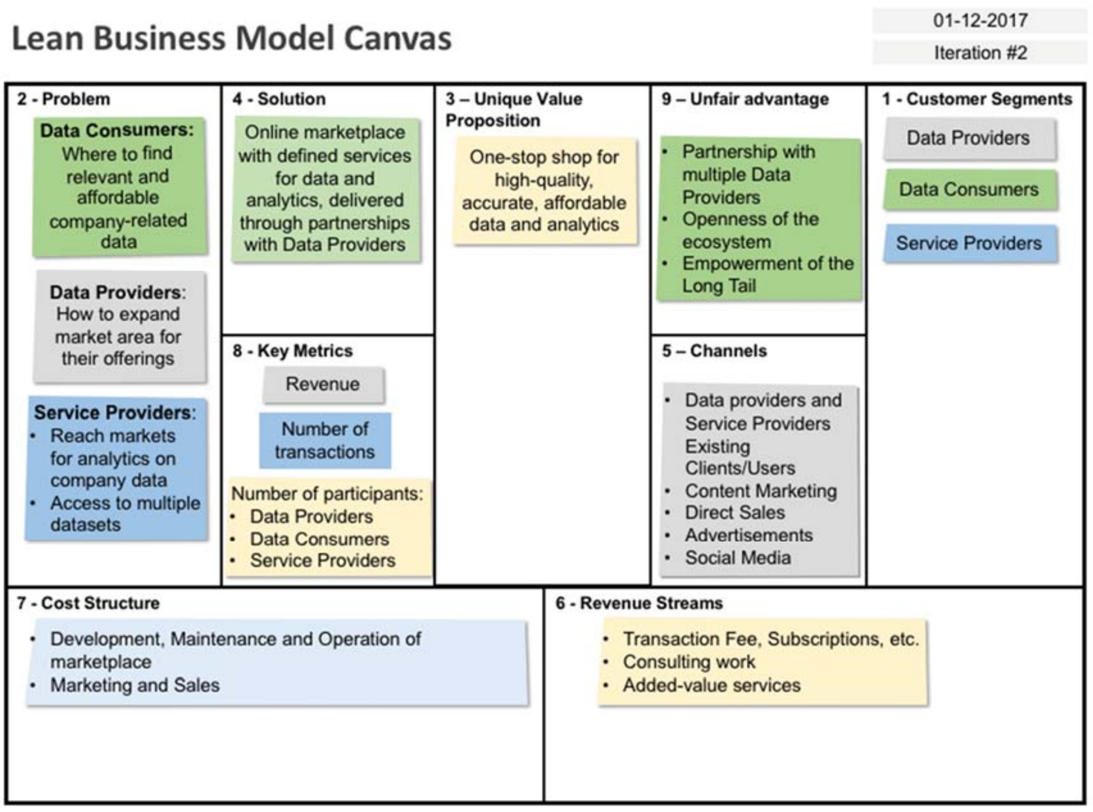
2 euBusinessGraph Marketplace Overview

This section gives an overview of the euBusinessGraph Marketplace, its customer segments and their requirements, and the revised architecture of the euBusinessGraph Marketplace and Services.

2.1 Business Model and Customer Segments

A business model for the euBusinessGraph Marketplace has been developed following the *lean business model canvas*. The business model is described in more details in Deliverable D5.2 "Exploitation and Dissemination Strategy". For the convenience of the reader, Appendix A of this report includes a copy of the business model description. In this section we provide a summary of the business model canvas, focusing on the three defined user segments.

The lean business model canvas shown in Figure 2 depicts the value propositions, customers and finances, the problem, the solution, key metrics, and competitive advantage of the euBusinessGraph Marketplace. The canvas is a fast and effective way to communicate the business model of the marketplace to internal and external stakeholders.



Lean Business Model Canvas

01-12-2017
Iteration #2

2 - Problem
- **Data Consumers:** Where to find relevant and affordable company-related data
- **Data Providers:** How to expand market area for their offerings
- **Service Providers:**
  - Reach markets for analytics on company data
  - Access to multiple datasets

4 - Solution
Online marketplace with defined services for data and analytics, delivered through partnerships with Data Providers

8 - Key Metrics
- Revenue
- Number of transactions
- Number of participants:
  - Data Providers
  - Data Consumers
  - Service Providers

3 – Unique Value Proposition
One-stop shop for high-quality, accurate, affordable data and analytics

9 – Unfair advantage
- Partnership with multiple Data Providers
- Openness of the ecosystem
- Empowerment of the Long Tail

5 – Channels
- Data providers and Service Providers Existing Clients/Users
- Content Marketing
- Direct Sales
- Advertisements
- Social Media

1 - Customer Segments
- Data Providers
- Data Consumers
- Service Providers

7 - Cost Structure
- Development, Maintenance and Operation of marketplace
- Marketing and Sales

6 - Revenue Streams
- Transaction Fee, Subscriptions, etc.
- Consulting work
- Added-value services

Figure 2: euBusinessGraph Marketplace – lean business model canvas

As can be seen, there are three distinct customer segments of the euBusinessGraph Marketplace, namely (1) data providers, (2) data consumers and (3) service providers. For each of these customer segments, we have defined a *value proposition canvas* that describes how the euBusinessGraph Marketplace is creating values to the customer. This type of canvas consists of a Customer Profile describing the user segments in the business model and a Value Proposition Map describing features of a specific value proposition for a product.

2.1.1 Data providers

This group of users has data which they would like to monetize. A data marketplace such as the euBusinessGraph Marketplace could provide a channel for them to achieve this. Figure 3 shows a value proposition canvas for a generic data provider. The data provider may have company data from one or more national registries that it wants to market and sell through the marketplace. Data providers can also provide other types of company data, e.g., an organization which has information on outstanding invoices in their Enterprise Resources Planning (ERP) solution could be interested in providing this information via the marketplace to credit scoring companies for a fee.

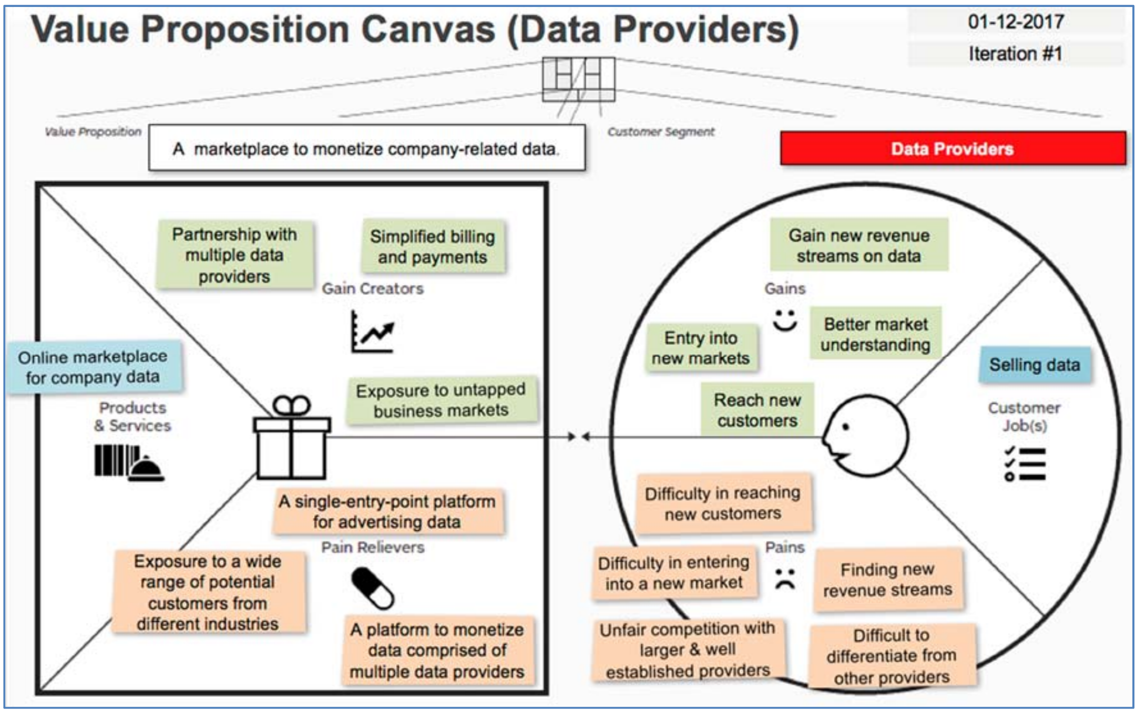

Value Proposition Canvas (Data Providers)
01-12-2017
Iteration #1
Value Proposition
A marketplace to monetize company-related data.
Customer Segment
Data Providers
Partnership with multiple data providers
Simplified billing and payments
Gain Creators
Online marketplace for company data
Products & Services
Exposure to untapped business markets
A single-entry-point platform for advertising data
Pain Relievers
Exposure to a wide range of potential customers from different industries
A platform to monetize data comprised of multiple data providers
Gain new revenue streams on data
Gains
Entry into new markets
Better market understanding
Selling data
Customer Job(s)
Reach new customers
Difficulty in reaching new customers
Pains
Difficulty in entering into a new market
Finding new revenue streams
Unfair competition with larger & well established providers
Difficult to differentiate from other providers

Figure 3: Data providers – value proposition canvas

2.1.2 Data consumers

This group of users requires certain types of business-related data in the daily operation of their organizations to help them make better business decisions. Figure 4 shows a value proposition canvas for a generic data consumer. An example of a data consumer is an organization that gets a credit scoring report at a fee to evaluate the strength of the financial footing of a potential customer or vendor.

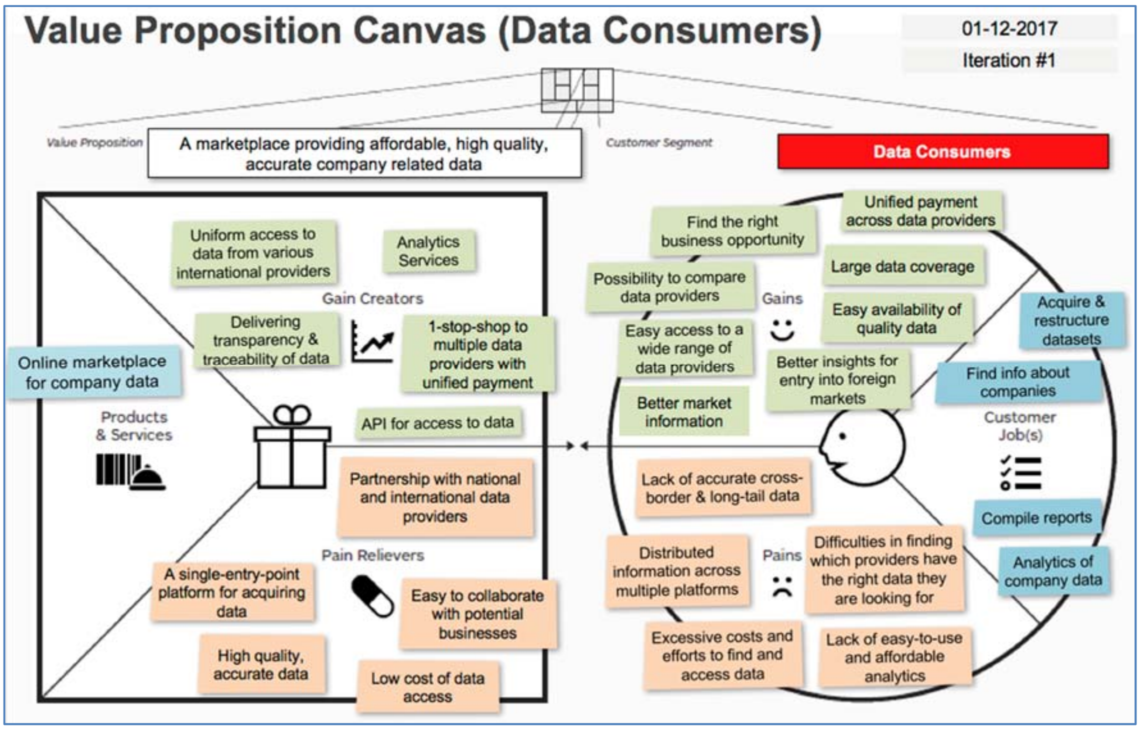

Value Proposition Canvas (Data Consumers)
01-12-2017
Iteration #1
Value Proposition
A marketplace providing affordable, high quality, accurate company related data
Customer Segment
Data Consumers
Uniform access to data from various international providers
Analytics Services
Gain Creators
Delivering transparency & traceability of data
1-stop-shop to multiple data providers with unified payment
Online marketplace for company data
Products & Services
API for access to data
Partnership with national and international data providers
Pain Relievers
A single-entry-point platform for acquiring data
Easy to collaborate with potential businesses
High quality, accurate data
Low cost of data access
Find the right business opportunity
Unified payment across data providers
Large data coverage
Possibility to compare data providers
Gains
Easy availability of quality data
Easy access to a wide range of data providers
Better insights for entry into foreign markets
Better market information
Acquire & restructure datasets
Find info about companies
Customer Job(s)
Lack of accurate cross-border & long-tail data
Compile reports
Distributed information across multiple platforms
Pains
Difficulties in finding which providers have the right data they are looking for
Analytics of company data
Excessive costs and efforts to find and access data
Lack of easy-to-use and affordable analytics

Figure 4: Data consumers – value proposition canvas

2.1.3 Service providers

Figure 5 shows the value proposition canvas for service providers. This group of users can provide better visualization of the data available in the data marketplace. They can potentially provide various data services on the vast amount of data available in a data marketplace such as data cleaning and data quality assurance or produce new data and insights that add value on top of the data that is already available.

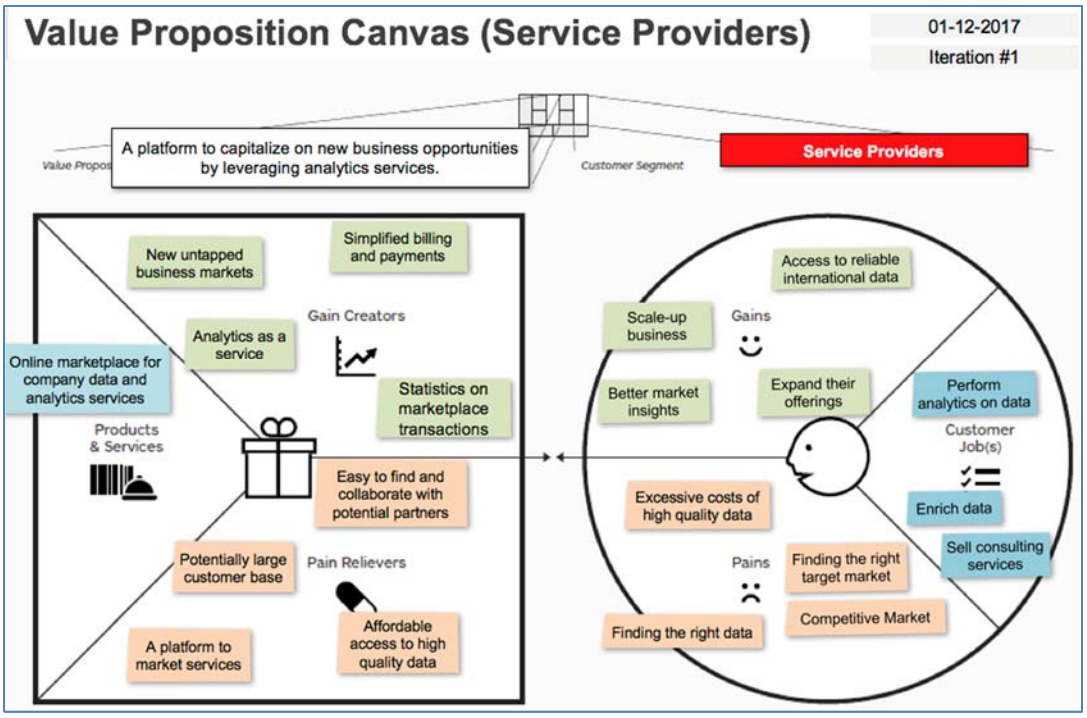

Value Proposition Canvas (Service Providers)
01-12-2017
Iteration #1
Value Propos
A platform to capitalize on new business opportunities by leveraging analytics services.
Customer Segment
Service Providers
New untapped business markets
Simplified billing and payments
Gain Creators
Analytics as a service
Online marketplace for company data and analytics services
Statistics on marketplace transactions
Products & Services
Easy to find and collaborate with potential partners
Potentially large customer base
Pain Relievers
Affordable access to high quality data
A platform to market services
Access to reliable international data
Scale-up business
Gains
Better market insights
Expand their offerings
Perform analytics on data
Customer Job(s)
Enrich data
Excessive costs of high quality data
Pains
Finding the right target market
Sell consulting services
Competitive Market
Finding the right data

Figure 5: Service providers – value proposition canvas

2.2 Marketplace and Services

In Deliverable D3.2 "euBusinessGraph Marketplace and Services – v1" we presented the first release of the software components implementing the euBusinessGraph Marketplace. The set of services were categorized according to the tasks of WP3. For the second release of the euBusinessGraph Marketplace and Services we have grouped these services according to three categories corresponding more closely with the three customer segments described above:

- **Marketplace and data hosting services** targeting **data consumers**: These services (1) 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, and (2) provide a reliable hosting service for the business graph;
- **Data ingestion services** targeting **data providers**: These services provide (1) data preparation services that simplify data cleaning and transformation processes ensuring the availability of services to help with the generation of the business graph data, and (2) data interlinking services that provide necessary RDF-ization services for data transformation, enrichment and interlinking and metadata generation processes so that the business graph data can be published as linked data;
- **Cross-cutting business analytics services** & **business case services** targeting **service providers** and **data consumers**: These services are examples of added-value services. The cross-cutting business analytics services provide a set of analytics services on top of the business graph data, while business case services provide data-driven applications focusing on focused application areas. These services can be provided by third party service providers and be made available as marketplace offerings to the data consumers. Analytics services developed by JSI and UNIMIB that have been developed in WP3 are included as part of the marketplace offerings. Additionally, the business services from the business cases can be offered through the marketplace. For more details on the business case services see Deliverable D4.2 "Evaluation Report of Business Cases Products & Services – v1".

Figure 6 illustrates the main service features that are available for these service categories.

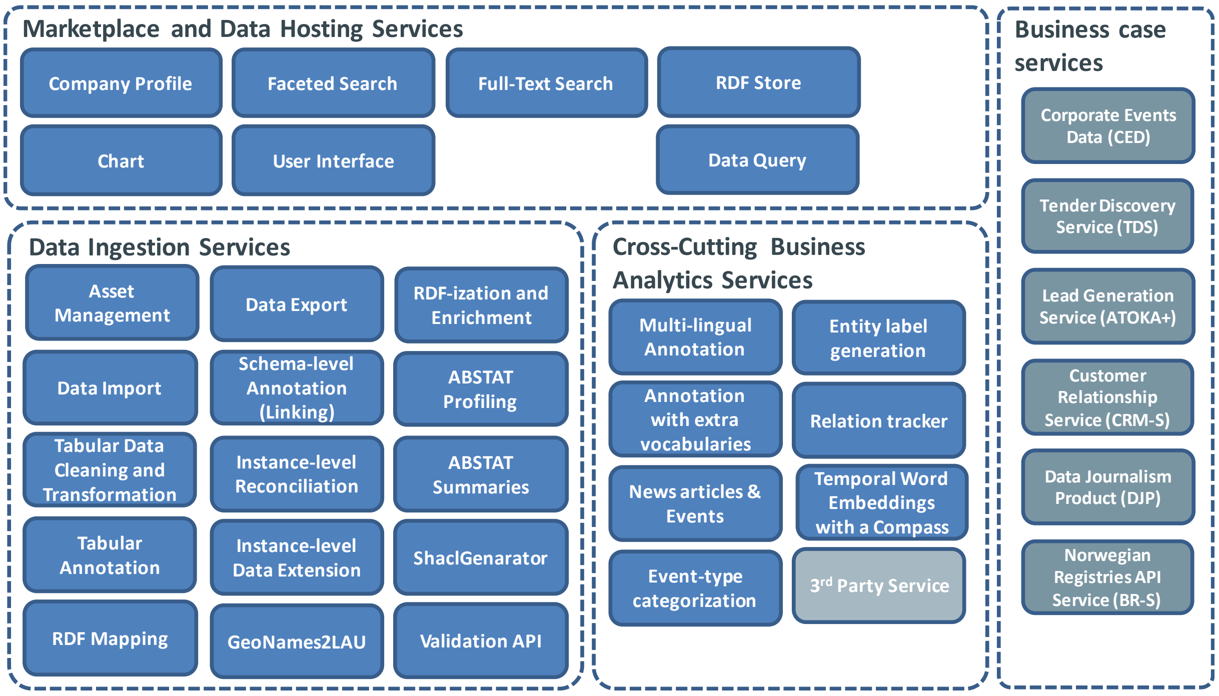

Marketplace and Data Hosting Services
Company Profile
Faceted Search
Full-Text Search
RDF Store
Chart
User Interface
Data Query
Business case services
Corporate Events Data (CED)
Tender Discovery Service (TDS)
Lead Generation Service (ATOKA+)
Customer Relationship Service (CRM-S)
Data Journalism Product (DJP)
Norwegian Registries API Service (BR-S)
Data Ingestion Services
Asset Management
Data Export
RDF-ization and Enrichment
Data Import
Schema-level Annotation (Linking)
ABSTAT Profiling
Tabular Data Cleaning and Transformation
Instance-level Reconciliation
ABSTAT Summaries
Tabular Annotation
Instance-level Data Extension
ShaclGenarator
RDF Mapping
GeoNames2LAU
Validation API
Cross-Cutting Business Analytics Services
Multi-lingual Annotation
Entity label generation
Annotation with extra vocabularies
Relation tracker
News articles & Events
Temporal Word Embeddings with a Compass
Event-type categorization
3rd Party Service

Figure 6: euBusinessGraph Marketplace and Services (revised)

2.2.1 Software components

Figure 7 overlays the software components that 1) have been developed and/or enhanced in WP3 during the first period of the project (i.e. D3.2), and 2) new components to be developed during the next period of the project (i.e. D3.4), including enhancements of existing components. Each of these software components is responsible for implementing the set of the service features listed.

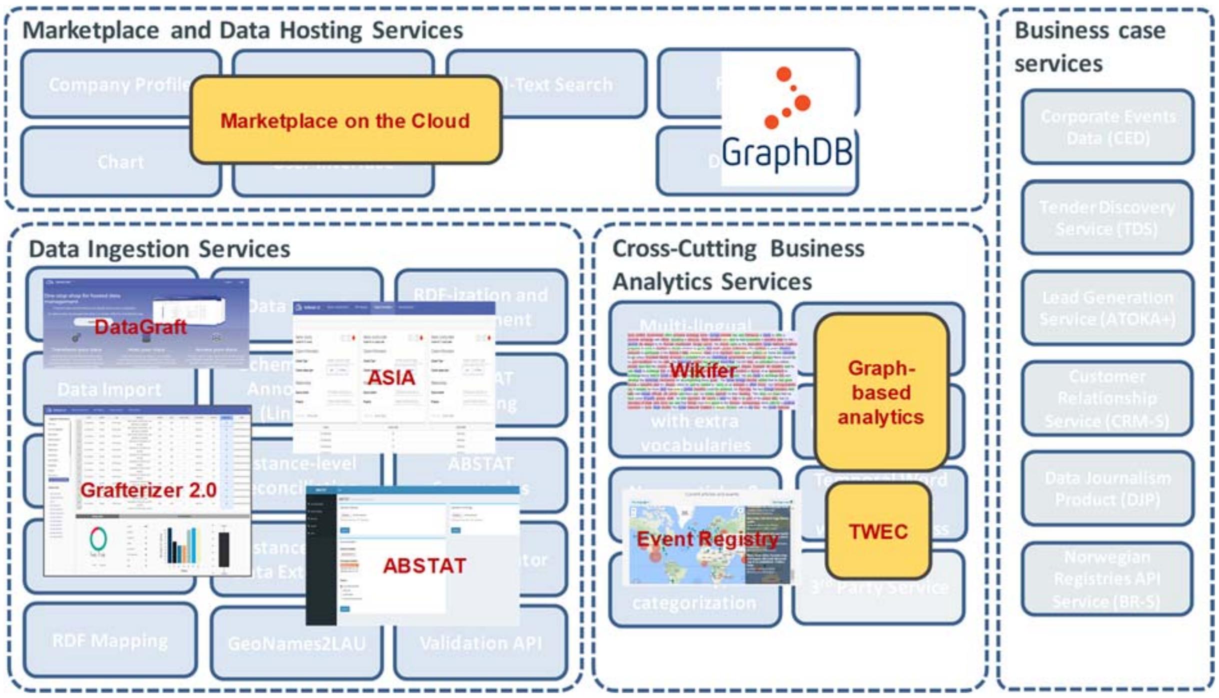


Figure 7: euBusinessGraph Marketplace and Services – software components[1]

More detailed descriptions for each software component, following the structure defined in Table 1, are given in Sections 3, 4 and 5.

Table 1: Software component description template

Subsection	Description
Factsheet	A short factsheet summarizing the functionality of the software component and its usage in the euBusinessGraph project. Additionally, the factsheet contains links to installation guides, user guides, API documentation, software license, and source code repository, and finally the name of the contact person.
Requirements	Short description of the relevant requirements from the business use cases being addressed by this software service/component.
Architecture	Architecture description of the implemented software service/component. Architecture component diagram.
Functionality	Description of the functionality provided by the software service/component. Screenshots of GUI if possible.
APIs	Short API description and links to online API documentation.
Development plans for 2nd release (D3.4)	Short description of the development plan for the 2nd release of the software service/component in month 24.

2.3 Requirements Analysis

After developing the business model for the marketplace, we revisited the initial requirements analysis from Deliverable D3.1 and revised them according to the three customer segments and the three service categories defined for the euBusinessGraph Marketplace and Services.

[1] Already available software components are shown using screenshots, while software components depicted using yellow boxes indicate new components that will be developed in the 2nd period of the project.

2.3.1 Data provider requirements

The **data providers** in euBusinessGraph are represented by OCORP, SDATI, BRC and ONTO. For the data providers the requirements analysis was 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 analyzed 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 have been summarized into the requirements table (Table 2).

Table 2: Data provider requirements (DPRs)

ID	Requirement	Requirement description
Data Ingestion 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).
DPR-03	Entity linking	Entity linking gets as input a text and produces a set of annotations.
DPR-04	Semantic labeling	Semantic labeling gets as input a (weakly) structured source and produces a set of annotations that are used for generating mappings.
DPR-05	Link discovery	Link discovery gets as input a knowledge graph and produces a set of mappings.
Marketplace and 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).
DPR-11	License models	Support for different license models for accessing and APIs for accessing and modifying datasets, including dual-license models that allow 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 is 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.
Cross-Cutting Business Cases Analytics Services		
DPR-10	Multi-lingual annotation	Support for multi-lingual annotation of text with links to relevant/common concepts.

2.3.2 Data consumer requirements

The **data consumers** in euBusinessGraph are represented by the business case partners CERVED, SDATI, EVRY and DW. They will access and use the data made available via the business graph for creation of data-driven products and services. For the data consumers the requirements analysis was 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 summarized into the requirements table (Table 3).

Table 3: Data consumer requirements (DCRs)

ID	Requirement	Requirement description
Marketplace and 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 predefined 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 integrate 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.
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.
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 computing similarity amongst entities, e.g., resolving ambiguity.
DCR-07	Relation extraction	Support for extracting relations between entities.

2.3.3 Service provider requirements

The **service providers** in euBusinessGraph are represented by OCORP, CERVED, SDATI, EVRY, DW, BRC and JSI. They provide value-added services in the marketplace. The value-added services can be in the form of business/application products that are being developed by the business case partners, i.e.:

- **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 standardized 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 at the start of the project only provided company-related data in Italy, to cover new jurisdictions, specifically company-related data in the United Kingdom and Norway;
- **EVRY 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.

Value-added service can also be more general cross-cutting business analytics services such as the analytics services provided by JSI (see Section 5).

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. However, the requirements of this customer segment can be said to include the data consumer requirements. In addition, there is a need for well-defined APIs and documentation describing how to access and use the business graph data and a flexible marketplace architecture supporting deployment and hosting of 3rd party services/applications.

Table 4: Service provider requirements (SPRs)

ID	Requirement	Requirement description
Marketplace and Data Hosting Services		
SRP-01	APIs and/or documentation	Well-defined APIs and documentation describing how to access and use the business graph data.
SRP-02	Deployment and hosting	Flexible marketplace architecture supporting deployment and hosting of 3rd party services/applications.

3 Marketplace and Data Hosting Services

Table 5 lists the software components that will implement the Marketplace and Data Hosting Services depicted in Figure 7.

Table 5: Software components – Marketplace and Data Hosting Services

Software component	Functionality	Implementation status
Marketplace on the Cloud	Search and discovery of company data • Full-text search; • Faceted search; • Company profile view; • Basic analytics; • Articles search; • Events search.	In development.
GraphDB on the Cloud	A scalable semantic graph database.	Developed.

The Marketplace and Operational Services as defined in Deliverable 3.1 have been our first iteration to collect requirements and address the design issues of the system. In the current deliverable, more work has been put into shaping the specification of the Marketplace and correspondingly we present our current version of the marketplace services design in terms of a component architecture diagram. This section also represents the current and respectively intermediate (working version) specification including definitions of the marketplace services API that will be finalized for the upcoming Deliverable 3.4.

The business graph of company data will be easier to manage and maintain if larger portions of it are centralized. The *Data Hosting Services* aim to provide just that, by provisioning a reliable hosting service for the business graph. They will be based on the Ontotext GraphDB triple store. The database will serve as the main provider of data hosting for the Data Marketplace.

The Data Hosting Services follow a microservices pattern where each component is an independent part of the whole. Communication is carried out in a standardized manner, via API calls. This ensures that the system is more manageable and eases maintenance and future expansions. All marketplace components utilize this microservices architecture.

The requirements specified in this section are based on the preliminary user group analysis that we carried out and described in Deliverable 3.1. Those requirements are still subject to change depending on changes within the vision of the stakeholders or on new data.

3.1 Marketplace

3.1.1 Factsheet

Table 6: Marketplace – Fact sheet

Marketplace	
Functionality	Search and discovery company data: • Full-text search; • Faceted search; • Company profile view; • Basic analytics; • Articles search; • Events search.
Usage in euBusinessGraph	The implemented marketplace serves as a proof-of-concept that demonstrates the company data and service capabilities that can be exploited by commercial marketplace providers.
Installation guide	Available online.

User guide	A detailed user guide will be provided to both data providers and data consumers.
API documentation	• http://ebg-fts.ontotext.com/swagger-ui.html • http://ebg-fsearch.ontotext.com/swagger-ui.html • http://ebg-company-profile.ontotext.com//swagger-ui.html • http://ebg-chart.ontotext.com/swagger-ui.html
Software license	The Marketplace is delivered with commercial license and freemium business model. In this model some of the data providers will sell their data, and some will provide open data under dual licensing: paid and free of charge for particular types of usage (e.g., for academic use).
Source code repository	Not open source.
Contact person	boyan.simeonov@ontotext.com

3.1.2 Requirements (usage in euBusinessGraph)

Detailed descriptions of the different customer segments and their high-level requirements are provided in 2. Here we will provide only requirements that are relevant for the Marketplace.

3.1.2.1 Data provider requirements

Here we present a refined set of requirements for the data providers.

Table 7: Marketplace – Data provider requirements (DPRs)

ID	Requirement	Requirement description
Marketplace Services		
DPR-11	License models	Support dual-license models that allow for both payment or share-alike for public-benefit use of the datasets uploaded by data providers.
DPR-11-a	Dataset-level license access control	Access control policy at a database or repository level.
DPR-11-b	Data item license access restrictions	Ability to configure types of data the consumer will receive after payment or agreeing with a particular license model.
DPR-11-c	Advertise company data	Revenue originating from data flows is shared.
DPR-11-d	Shared revenue	Security and access control for users.
DPR-12	Security and access control	Manage access of users to APIs.
DPR-13	User registration and access management	Manage access of users to APIs and groups.

3.1.2.2 Data consumer requirements

The data consumers have the following requirements:

Table 8: Marketplace – Data consumer requirements (DCRs)

ID	Requirement	Requirement description
Marketplace 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 allowing to query, navigate and download datasets.
DCR-10	User registration and access management	User sign-up, log-in, and profile management.

3.1.2.3 Service provider requirements

The service providers have requirements for data analytics services that need access to the following types of data:

- Company info;
- People info;
- Product/brand info.

3.1.3 Architecture

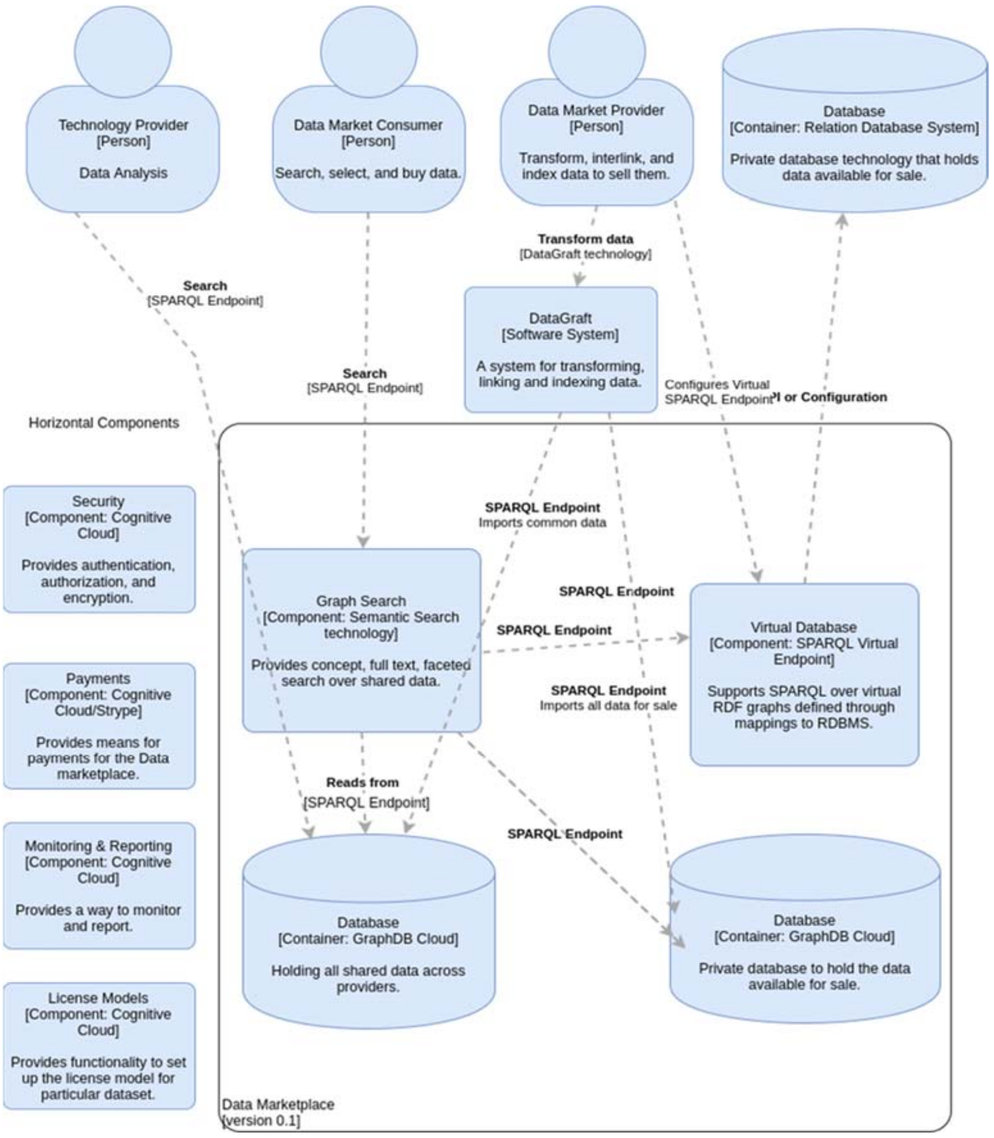

Technology Provider
[Person]
Data Analysis
Data Market Consumer
[Person]
Search, select, and buy data.
Data Market Provider
[Person]
Transform, interlink, and index data to sell them.
Database
[Container: Relation Database System]
Private database technology that holds data available for sale.
Transform data
[DataGraft technology]
Search
[SPARQL Endpoint]
DataGraft
[Software System]
A system for transforming, linking and indexing data.
Search
[SPARQL Endpoint]
Configures Virtual SPARQL Endpoint
API or Configuration
Horizontal Components
Security
[Component: Cognitive Cloud]
Provides authentication, authorization, and encryption.
SPARQL Endpoint
Imports common data
Graph Search
[Component: Semantic Search technology]
Provides concept, full text, faceted search over shared data.
SPARQL Endpoint
SPARQL Endpoint
Virtual Database
[Component: SPARQL Virtual Endpoint]
Supports SPARQL over virtual RDF graphs defined through mappings to RDBMS.
Payments
[Component: Cognitive Cloud/Strype]
Provides means for payments for the Data marketplace.
SPARQL Endpoint
Imports all data for sale
Reads from
[SPARQL Endpoint]
SPARQL Endpoint
Monitoring & Reporting
[Component: Cognitive Cloud]
Provides a way to monitor and report.
Database
[Container: GraphDB Cloud]
Holding all shared data across providers.
Database
[Container: GraphDB Cloud]
Private database to hold the data available for sale.
License Models
[Component: Cognitive Cloud]
Provides functionality to set up the license model for particular dataset.
Data Marketplace
[version 0.1]

Figure 8: Component Diagram of the Marketplace and Operational services[2]

[2] C4 model for software architecture, https://c4model.com/

The euBusinessGraph Marketplace provides different graphs of data which can be analyzed, searched and explored by invoking queries directly across a set of diverse datasets sharing common properties. The current understanding between the partners of euBusinessGraph is that the marketplace will adhere to a **common shared dataset** principle. That means that data providers joining the business graph will share a common subset of their data described according to a common data model.

The common shared data are kept in GraphDB database in the form of graphs of linked data as defined in the euBusinessGraph Company Data Model. The data providers of the marketplace can decide whether to transform and upload all of their data or to create a virtual database that acts as a proxy to their (relational) database instance. The proxy will hold the data available for sale on the marketplace. In both cases the data provider must transform their data into a canonical semantic form and link it according to the common shared data model, before making it available.

The marketplace consumer will search for data via search user interfaces and/or APIs provided by the graph search component. The available functionalities are:

1. Full-text search in the indexed shared data fields such as name, description, address, etc.
2. Concept search for the name of an organization, location in address, etc.
3. Faceted search to filter data based on various conditions such as organization name, country, postal code, various classifications, etc.

The search component will allow users to query the GraphDB database. Data consumers will use this component to search for data. When users choose to buy data, this component will be responsible for collecting all data from the data provider and making it available for download in multiple formats. These formats include but are not limited to various flavours of RDF, JSON, JSON-LD, and XML, etc.

The search component will combine intelligent scoring and ranking of results by applying different masking weights on the concepts and full-text individual rankings of results for the benefit of better retrieval of data.

The *security component* 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 the security component;
- Authentication and authorization: all access to the platform marketplace portal will be authenticated via a username and password, while all access to the REST services exposed on the platform will be authenticated via private API key/secret pairs;
- Datasets will offer different access levels: *public* for open source datasets, and *private* available to the data consumers after payment. Appropriate authorization measures will be taken to ensure that users have access only to appropriate data.

The *monitoring and reporting* component 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;
- 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.

The *license models* component is responsible for managing and enforcing data access based on license models and 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.

3.1.4 Functionality (to be used, enhanced or developed)

Table 9: Marketplace – Summary of features

Feature	Description	Existing[3]	New[4]
Company Profile Service	The Company Profile Service is responsible for serving all the information about a particular company - address, type, sector, active status, etc.		+
Faceted Search Service	The Faceted Search Service is responsible for the faceted search and the combination between facets and full-text search. It handles user-defined search criteria and returns appropriate results.		+
Full-Text Search Service	It is responsible for the autocomplete functionality and the full-text search.		+
Chart Service	It takes different parameters like country, activity, city, etc. as input and produces aggregations which then are represented as charts in the User Interface.		+
User Interface	The User Interface is the face of Marketplace service. All user interactions go through it. It is based on AngularJS.		+

The *Marketplace and Operational Services* will ensure the availability of a data brokerage system in the form of a data marketplace where data and datasets 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 Kubernetes and DataGraft.

The Marketplace will implement the following scenarios.

- **Scenario 1**. A data producer has data for sale. She will export the data from her relational database system that keeps the data and will transform and interlink to the Marketplace schema with DataGraft. Then she will upload the data that she wants to share with data consumers to the GraphDB database;
- **Scenario 2**. A data producer has data for sale, but there are business or regulatory limitations to share all of the data in a provisioned instance of GraphDB. She then will export the data she wants to share with the data consumer and will link and transform it into the master schema. She will then configure a separate repository on the Marketplace that will link her internal relational database with the search component;
- **Scenario 3**. A data consumer needs organizational data about companies from a particular sector in a London district. She will use the faceted search user interface to filter organizations by location and sector. There will be data about such companies from two providers. She will

[3] Existing software feature brought as background to be used or enhanced in the euBusinessGraph project.
[4] New or enhanced software feature developed as a new foreground during the euBusinessGraph project.

explore the available data by each provider and will choose to buy the data from one of the providers.

3.1.5 APIs

3.1.5.1 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. The API specification can be found in the GraphDB Workbench[5].

GraphDB implements the minimal Role Based Access Control (RBAC1) model with role hierarchy support. The model defines four entities:

- User - every user regardless if it is an anonymous or authenticated user;
- Session - the current session connections; every session always works in the context of a specific user;
- Roles - a group of permissions associated with a role; the roles may be organized in hierarchies i.e. Admin role give you all other roles without explicitly defining them;
- Permission - gives rights to execute a specific operation.

3.1.5.2 Usage reporting

The API for usage reporting provides a method that provides usage reports of the database resources (e.g., number of running queries, CPU usage, used memory) that are automatically logged by the euBusinessGraph platform.

3.1.6 Development plans for 2nd release (D3.4)

Currently, only the data hosting layer, master data, security, payments and billing, and monitoring components are implemented. The rest of the components, as depicted in Figure 8, will be implemented and configured in a fully working system for the second release.

3.2 GraphDB Cloud

The Ontotext Cloud provides:

- A scalable semantic graph database. The main data hosting provider for the Data Marketplace;
- Enterprise text analysis services for news, life sciences, and social media;
- Access to knowledge graphs such as DBpedia, Wikidata and GeoNames;
- Secure and fully managed service.

[5] http://data.businessgraph.io/webapi

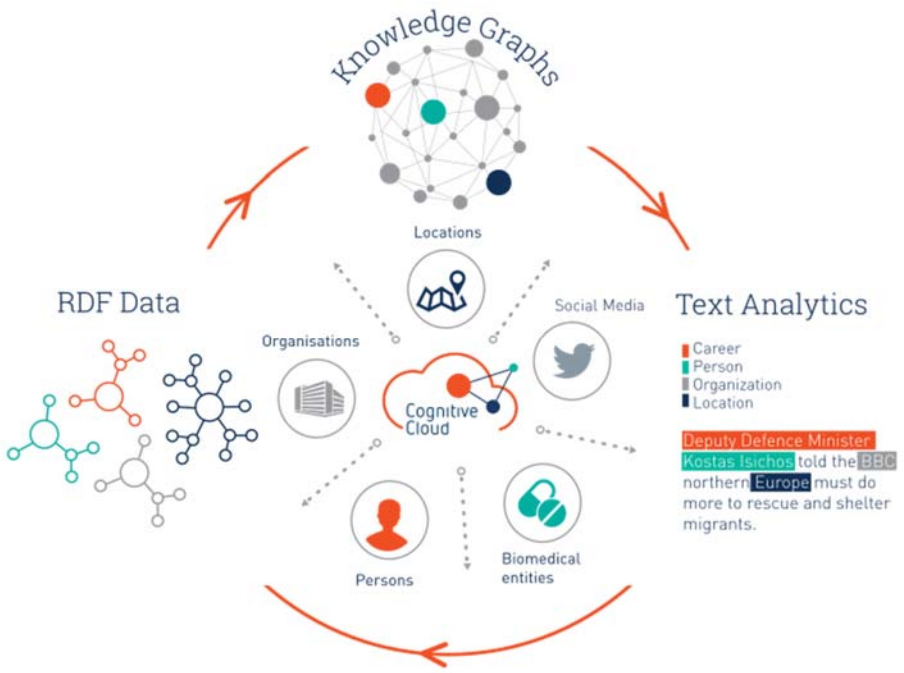


Figure 9: Ontotext Cloud

GraphDB Cloud is a product running on the Ontotext Cloud Platform that is designed to serve scenarios with small, medium and enterprise database size and query load.

With GraphDB Cloud, users can have their private database instance up and running within seconds. They will no longer need to deal with DBA specific tasks such as installation and upgrades, provisioning and deployment, backups and restores, as well as ensuring the database availability and security. GraphDB is instantly available and easily accessible so that users can build smart data prototypes and productions faster, and at a lower cost, without spending valuable time on installing, configuring, or developing their own infrastructure components.

The Ontotext Cloud platform is based on enterprise-grade technology from Ontotext including GraphDB and high-performance text mining solutions successfully applied at some of the largest enterprises in the world. We have taken proven technology used in publishing & media, government agencies and life sciences and made available with an accessible business model. The Ontotext Cloud is designed to deliver a fully managed, available, secure, and scalable solution that can be accessed via simple RESTful services.

The GraphDB Cloud, in particular will allow data providers from euBusinessGraph to reserve a database and then ETL (extract, transform, and load) their dataset by using the transformation services of the DataGraft platform. GraphDB Cloud will ensure that the euBusinessGraph platform will scale to many data providers.

3.2.1 Factsheet

Table 10: GraphDB Cloud – Fact sheet

GraphDB Cloud	
Functionality	A scalable semantic graph database.
Usage in euBusinessGraph	The main data hosting provider for the Data Marketplace.
Installation guide	https://www.ontotext.com/products/graphdb/
User guide	http://graphdb.ontotext.com/documentation/enterprise/
API documentation	http://graphdb.ontotext.com/documentation/enterprise/devhub/workbench-rest-api/index.html

Software license	GraphDB uses RDF4J as a library, taking advantage of its APIs for storage and querying, as well as the support for a wide variety of query languages (e.g., SPARQL and SeRQL) and RDF syntaxes (e.g., RDF/XML, N3, Turtle). The development of GraphDB is partly supported by SEKT, TAO, TripCom, LarKC, and other FP6 and FP7 European research projects.
Source code repository	GraphDB is proprietary software with a *freemium* business model.
Contact person	boyan.simeonov@ontotext.com

3.2.2 Requirements (usage in euBusinessGraph)

The requirements for the data hosting layer are summarized in the following tables.

Table 11: Data consumer requirements

ID	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 predefined 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.

Table 12: Data provider requirements

ID	Requirement	Description
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).

3.2.3 Architecture

GraphDB Cloud is designed to use the underlying cloud infrastructure of Amazon Web Services (AWS). GraphDB Cloud provides a flexible service, capable of dynamically expanding or shrinking the consumed cloud resources depending on the demand. The architecture is shown in Figure 10 below.

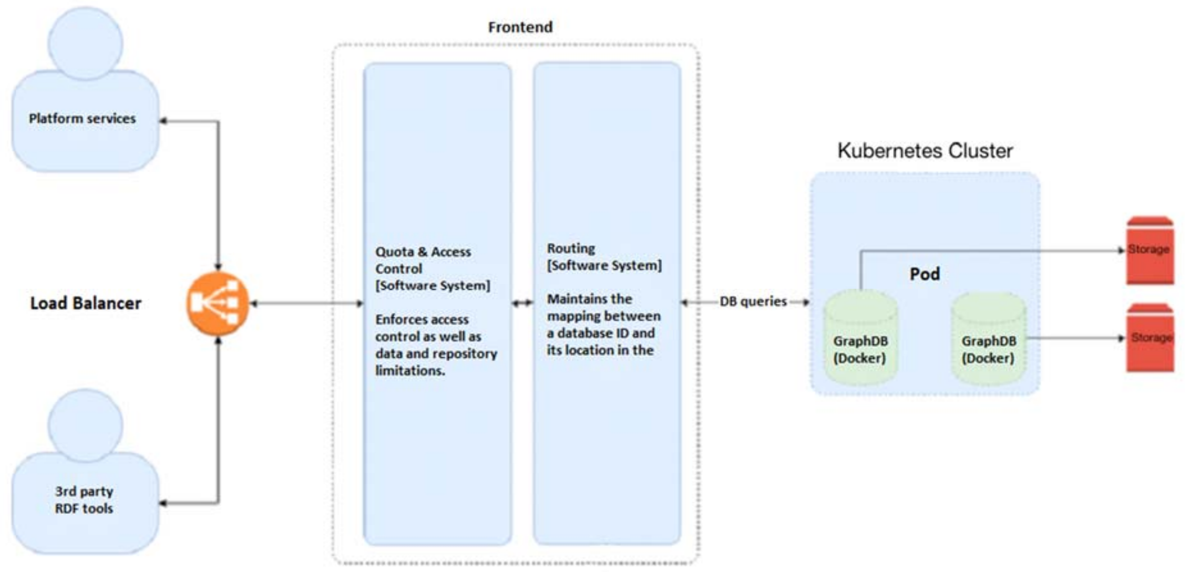

Frontend
Platform services
Load Balancer
3rd party RDF tools
Quota & Access Control [Software System]
Enforces access control as well as data and repository limitations.
Routing [Software System]
Maintains the mapping between a database ID and its location in the
DB queries
Kubernetes Cluster
Pod
GraphDB (Docker)
GraphDB (Docker)
Storage
Storage

Figure 10: GraphDB architecture

GraphDB Cloud architecture comprises of several components, in brief:

- Frontend: stateless node responsible for the interaction with the clients. The frontend serves as a router;
- Coordinator: orchestrates and keeps track of the health of the database cluster. Responsible for all database management actions, such as creation, deletion, upgrading, making backups, etc. Scales the number of data nodes as needed;
- Infrastructure services: services delivered by the Cloud infrastructure provider (in this case Kubernetes), such as load balancers, message queues (SNS), Docker container registries, backup storage, etc.

The architecture follows the microservices design principles where a complex application is decomposed into small independent processes that communicate with each other via RESTful APIs. The above architecture allows for:

- Flexible resource provisioning (scaling up and down) depending on the specific system demands;
- Minimal impact of system failures (component malfunction);
- Easier components replacement (failure recovery or upgrades).

All of the complexity of resource provisioning and software configuration of GraphDB Cloud is hidden behind management APIs. These APIs are used for integration with other systems. DataGraft uses them to provision and manage databases on demand. These APIs will be used when creating a new database for a euBusinessGraph user. This happens when a user creates their first SPARQL endpoint asset (i.e., their first RDF repository in their assigned database). After the database is created, again using the API, the platform obtains the URL of the newly provisioned repository and can change the availability of the SPARQL endpoint from public to private (which requires a specially assigned API key to issue queries, also produced by the management API). The management API supports a large number of other database and repository management tasks. Moreover, the DBaaS performs automated backups of the user data on regular intervals of time.

The hosted data are exposed by the DBaaS using two types of RESTful data access APIs:

- **Per repository**: SPARQL endpoint API – allows for issuing SPARQL SELECT queries; returns query results available in various formats;
- **Per database**: RDF4J (formerly known as Sesame) – allows for lower-level access; only accessible using a specially assigned API key. Each database instance is assigned a unique access URL that includes user account id and a database name.

The hosted data can be accessed via a user interface[6] that provides functionalities to query, search, navigate and explore the data.

3.2.4 Functionality (to be used, enhanced or developed)

Table 13: GraphDB Cloud – Summary of features

Feature	Description	Existing[7]	New[8]
RDF store	GraphDB Cloud which can contain any RDF-ized 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.	+	
Data query (SPARQL)	A standard SPARQL endpoint for querying data. A linked data endpoint, providing read-only access to RDF data. RDF4J API for querying of RDF data. RDF4J provides light-weight industry standard way for querying and processing RDF data.	+	

The Data Hosting Service, based on the Ontotext Cloud, as described within the introduction, will have to be improved and modified in order to conform to the needs of the Data Marketplace.

The hosting task improvements include scalability, performance, and reliability of a large number of semantic graph databases running in the Cloud. This way 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 API[9].

The incorporation of the cloud services with the overall Knowledge Marketplace means that the customers of the euBusinessGraph can easily process the results which they obtain from the marketplace. If the customers want to, they could utilize the components of the Cloud to explore, transform and load the data that they obtain from the Marketplace. Moreover, the Cloud's scalability and availability, as well as the already existent monitoring services, mean that the master database will be easier to manage and maintain than creating a custom solution for it.

The service components are:

- GraphDB Cloud which can contain any RDF-ized datasets and provides SPARQL query access to the data. For the purpose of improving database availability and resilience, the RDF

[6] http://graphdb.ontotext.com/documentation/standard/workbench.html
[7] Existing software feature brought as background to be used or enhanced in the euBusinessGraph project.
[8] New or enhanced software feature developed as a new foreground during the euBusinessGraph project.
[9] http://docs.rdf4j.org/rest-api/

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 exporting service, to access the dataset in a variety of standard formats.

3.2.5 APIs

The GraphDBCloud provides various REST APIs available online at:

- http://graphdb.ontotext.com/documentation/enterprise/using-the-workbench-rest-api.html

The API calls allow users to manage their accounts.

To use any of these APIs requires an API key for Basic HTTP authentication. The API keys are private for each user account and can be generated in Cloud Management.

The RESTful services accept JSON as input and return JSON as output.

3.2.6 Development plans for 2nd release (D3.4)

As a second release, we plan to add functionality to enable full-text search and faceted search as part of standard SPARQL of the fully managed database-as-a-service.

4 Data Ingestion Services

Table 14 lists the software components that will implement the Data Ingestion Services depicted in Figure 7.

Table 14: Software components – Data Ingestion Services

Software component	Functionality	Implementation status
DataGraft	DataGraft provides a user interface that enables user and account management, user assets cataloging and dataset and database management.	**Developed.** DataGraft is used to integrate a streamlined data onboarding process using the software components Grafterizer and ASIA.
Grafterizer 2.0	Grafterizer 2.0[10] is a web-based framework for tabular data cleaning and transformation and RDF mapping.	**In development.** Additional features and enhancements will be developed during the 2nd period of the project.
ASIA	ASIA is a semantic table enrichment tool integrated into Grafterizer 2.0, which provides Assisted Semantic Interpretation and Annotation of tables (CSV files).	**In development.** Additional features and enhancements will be developed during the 2nd period of the project.
ABSTAT	ABSTAT (Linked Data Summaries with ABstraction and STATistics) is a framework developed by UNIMIB to support users and machines in better understanding big and complex RDF datasets.	**In development.** Additional features and enhancements will be developed during the 2nd period of the project.

4.1 DataGraft

DataGraft provides a user interface that enables user data and account management, user assets cataloging and dataset and database management, whereas Grafterizer is an interactive tool for data cleaning and data transformation.

4.1.1 Fact sheet

Table 15: DataGraft – Fact sheet

DataGraft	
Functionality	DataGraft provides a user interface that enables user and account management, user assets cataloging and dataset and database management.
Usage in euBusinessGraph	DataGraft is used as a platform to integrate Grafterizer and ASIA.
Installation guide	https://github.com/datagraft/datagraft-portal
User guide	https://github.com/datagraft/datagraft-reference/blob/master/documentation.md
API documentation	https://datagraft.github.io/datagraft-API/dist/index.html?url=https://datagraft.github.io/datagraft-API/swagger.yaml
Software license	Eclipse Public License – v 1.0
Source code repository	https://github.com/datagraft/datagraft-portal
Contact person	dumitru.roman@sintef.no

[10] https://datagraft.io/

4.1.2 Requirements (usage in euBusinessGraph)

Most of the development effort in euBusinessGraph is related to re-implementing and adding new features to Grafterizer. DataGraft is used as a platform to integrate Grafterizer and ASIA. As DataGraft is mainly used as an integration framework we refer to the sections for Grafterizer, ASIA, and GraphDB for specific requirements related to euBusinessGraph. The software-specific requirements are mainly derived from the data provider requirements addressing the data provider customer segment.

4.1.3 Architecture

Both DataGraft and Grafterizer have been implemented through a microservice architecture consisting of a large number of sub-components, each contained in a Docker[11] container. The DataGraft and Grafterizer tool user interfaces have been integrated to provide a consistent user experience, whereby their connected microservices communicate with each other through REST. The individual components are illustrated in Figure 11.

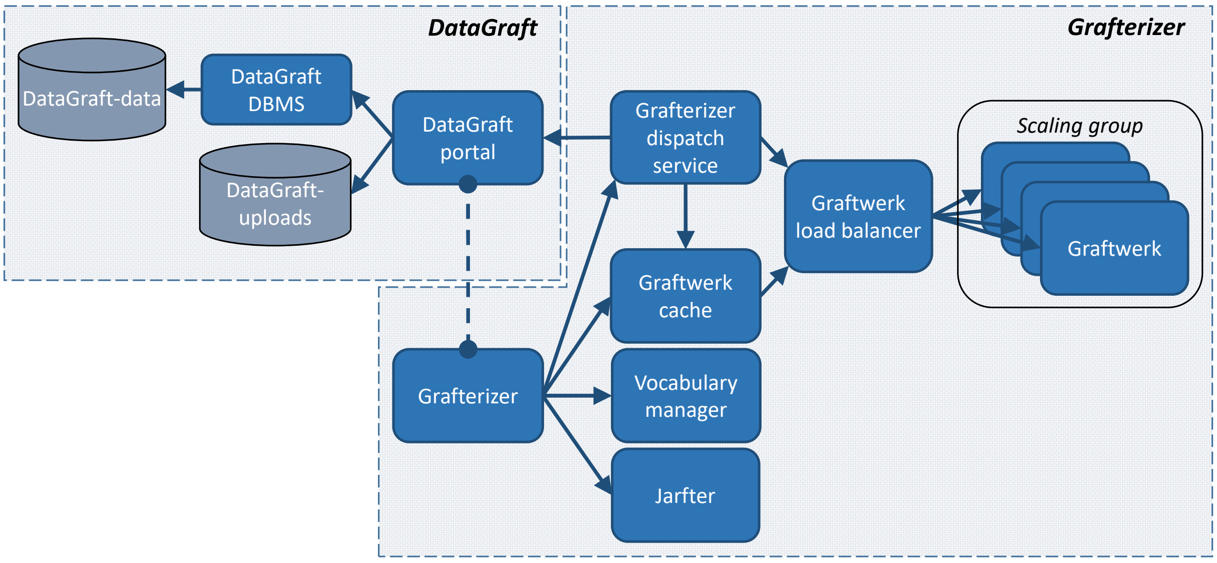


Figure 11. DataGraft and Grafterizer microservices architecture

DataGraft consists of the following components:

- **DataGraft portal:** The portal provides the web-based frontend that is used by the euBusinessGraph data publishers;
- **DataGraft DBMS:** This component represents the database management system (PostgreSQL[12]) for the user data and asset catalog. Data is stored in a separate volume (Docker volume or Amazon S3[13] in production).

Grafterizer includes the following sub-components:

- **Grafterizer frontend**: Component that implements the interactive graphical user interface for data cleaning, data transformation, and RDF mapping;
- **Grafterizer dispatch service**: A server component for the Grafterizer frontend that handles request authentication on its behalf (in order to ensure security) and dispatches requests for input and output across multiple services;

[11] https://www.docker.com/
[12] https://www.postgresql.org/
[13] https://aws.amazon.com/s3/

- **Graftwerk**: A sandboxed server component that executes data cleaning and transformation scripts that are generated by the Grafterizer frontend over a set of input data sent by the dispatch service. Graftwerk uses a proprietary load-balancing component in order to distribute the traffic when a larger number of users use the transformation tool;
- **Graftwerk cache**: A FIFO cache service for the Grafterizer frontend requests to Graftwerk;
- **Vocabulary manager**: RDF vocabulary management service for imported vocabularies used by the RDF mapping user interface of Grafterizer. Enables searching through concepts and import of vocabularies.

4.1.4 Functionality (to be used, enhanced or developed)

Table 16: DataGraft – Summary of features

Feature	Description	Existing[14]	New[15]
Asset management	Metadata, access management (public/private), sharing of assets, SPARQL endpoints, data transformations, queries.	+	
Data hosting and access	Data hosting and access services for all data stored on DataGraft.	+	
RDF database	Automatic reliable storage of files and RDF databases.	+	
Grafterizer 2.0	Interactive cleaning and transformation of tabular data, RDF generation, and tabular data enrichment.		+
GraphDB integration	Integration with GraphDB which hosts company data model and company graph data.		+
ASIA integration	Integration with ASIA to provide the functionality to enrich and annotate company data.		+
RDF data exploration	Interactive visual exploration of RDF data in the SPARQL endpoints published.	+	
REST API	REST-based access to all platform assets.	+	
Layered security	Encrypted user login information and SSL; asset security using OAuth2, API keys for semantic graph databases.	+	

DataGraft is used to integrate a streamlined data onboarding process using the software components Grafterizer and ASIA as illustrated in Figure 12 below. This process ensures a self-service data provisioning through:

- Interactive/Smart tabular data transformation (Grafterizer 2.0);
- Interactive/Smart data annotations (ASIA);
- Data provisioning as a service (DataGraft and GraphDB).

[14] Existing software feature brought as background to be used or enhanced in the euBusinessGraph project.
[15] New or enhanced software feature developed as a new foreground during the euBusinessGraph project.

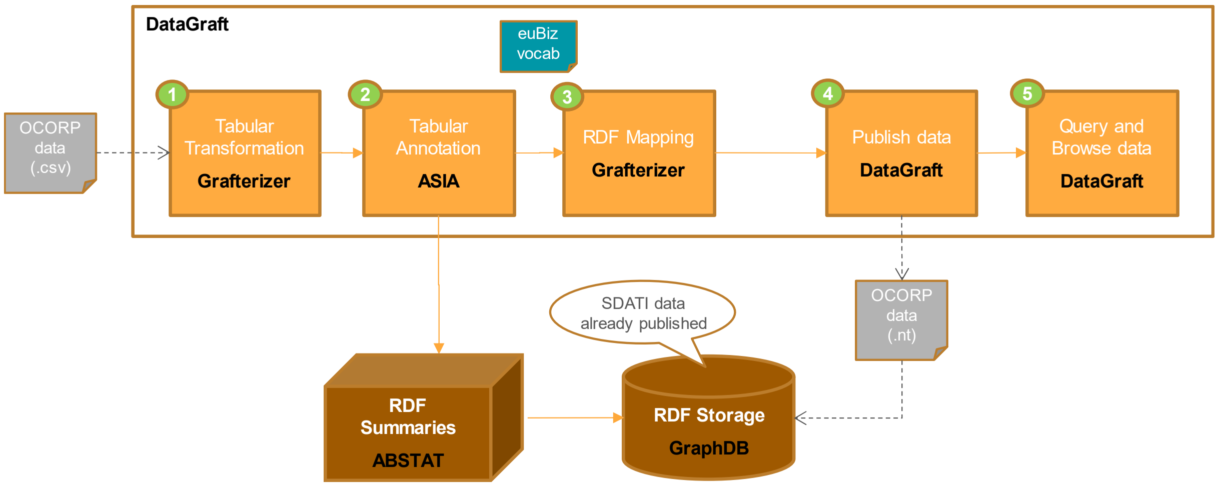


Figure 12: euBusinessGraph data provisioning

4.1.5 APIs

The REST API specification for DataGraft is maintained at:

- https://datagraft.github.io/datagraft-API/dist/index.html?url=https://datagraft.github.io/datagraft-API/swagger.yaml

4.1.6 Development plans for 2nd release (D3.4)

The focus of the DataGraft development in euBusinessGraph is on the Grafterizer and ASIA software components that are integrated into the DataGraft platform. Hence, some development effort is required on the DataGraft platform itself to ensure proper **integration of Grafterizer and ASIA** in order to provide a self-service data provisioning process for onboarding data to the business graph.

4.2 Grafterizer 2.0

Grafterizer 2.0[16] is a web-based framework for tabular data cleaning and transformation, as well as RDF mapping. *Deliverable 3.2 implemented the* ***alpha*** *version* of an effective approach for visual data profiling in Grafterizer that simplifies the process of preparing tabular data and contributes to improving data quality. Moreover, a prototype service has been implemented (i.e., Data Interlinking Services) in Grafterizer that supports semantic labelling of (weakly) structured CSV data sources.

Grafterizer is an integrated framework that provides functionality for the following services:

1. **Data import and data cleaning:** Service that provides functionality for cleaning and transformation of tabular CSV data.
2. **Tabular annotation:** Service that provides semantic annotation of tabular CSV data to RDF knowledge graphs (ASIA/ ABSTAT).
3. **RDF mapping:** Service that supports mapping of tabular CSV data to RDF knowledge graphs in a graphical tree-based interface.

Section 4.2 of this report covers the implementation of visual data profiling capabilities and suggestion-based data cleaning and transformation, while Section 4.3 includes the integration of semantic tabular annotation (ASIA/ABSTAT). Tabular annotation provides an alternative approach to the existing RDF mapping functionalities in Grafterizer.

[16] https://datagraft.io/

4.2.1 Factsheet

Table 17: Grafterizer 2.0 – Fact sheet

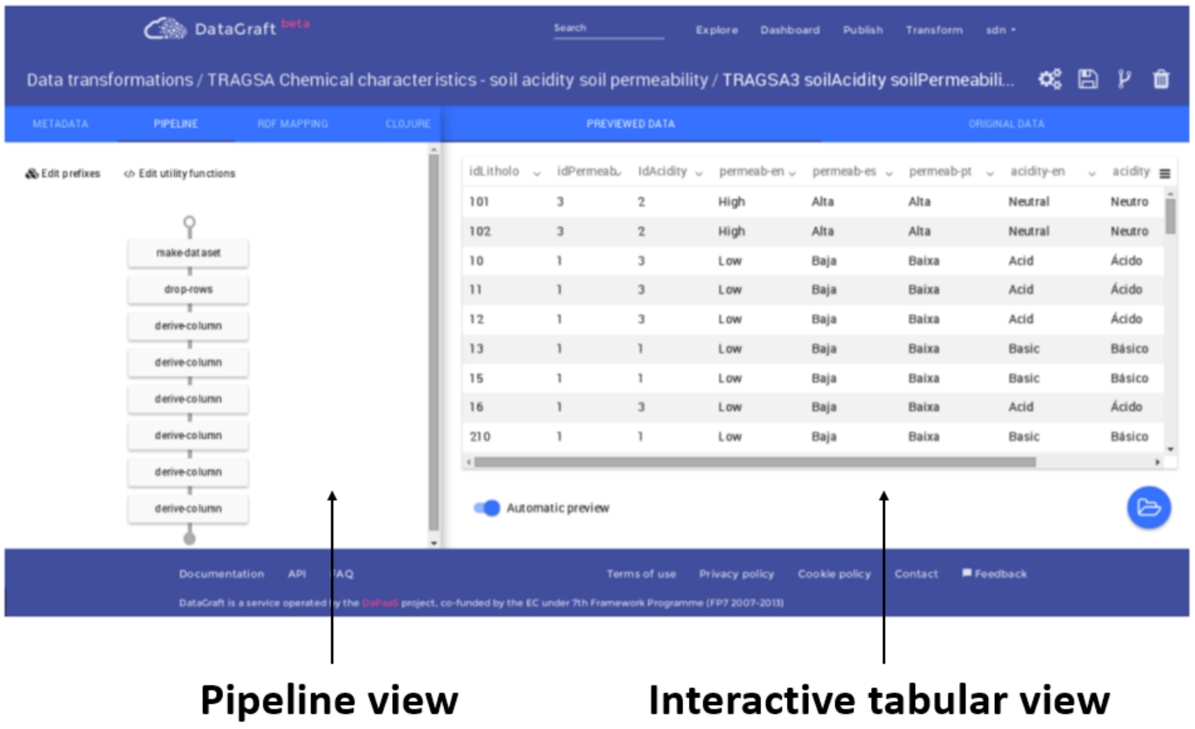
Grafterizer 2.0	
Functionality	In euBusinessGraph, we have extended the Grafterizer framework with interactive data cleaning and transformation functionality, and visual data profiling. The implementation provides functionality for: • Suggestion-based data cleaning and transformation; • Visual data profiling.
Usage in euBusinessGraph	Grafterizer is used together with ASIA to onboard the company data in the project, i.e. mapping the company data to the euBusinessGraph company data model in order to publish the data as a knowledge graph in GraphDB.
Installation guide	https://github.com/datagraft/grafterizer-2.0
User guide	https://github.com/datagraft/datagraft-reference/blob/master/documentation.md#transform
API documentation	https://datagraft.github.io/datagraft-API/dist/index.html?url=https://datagraft.github.io/datagraft-API/swagger.yaml
Software license	https://www.eclipse.org/legal/epl-v10.html
Source code repository	https://github.com/datagraft/grafterizer-2.0
Contact person	dumitru.roman@sintef.no

4.2.2 Requirements (usage in euBusinessGraph)

Previously gathered user feedback shows that Grafterizer has a steep learning curve and is complex to use. To meet the requirements of the euBusinessGraph to provide data preparation services that simplify cleaning and transformation processes, one of the goals of **D3.4** and the final period of the project will be to extend the current capabilities of Grafterizer by:

1. Adding additionally needed transformation functions.
2. Fix issues that emerge as a result of software- and usability testing.
3. Stabilize codebase with a focus on reliability, speed, and performance. The final result will be a stable beta release that can be used by data publishers to onboard data to the euBusinessGraph.

As a baseline for the implementation of the alpha version, Figure 13 shows the Grafterizer user interface at the time of D3.1. The user experience was characterized by a lack of overall assessment of data quality, difficulties in finding relevant data cleaning functionality, and limited interaction with the data table view.



Data transformations / TRAGSA Chemical characteristics - soil acidity soil permeability / TRAGSA3 soilAcidity soilPermeabili...

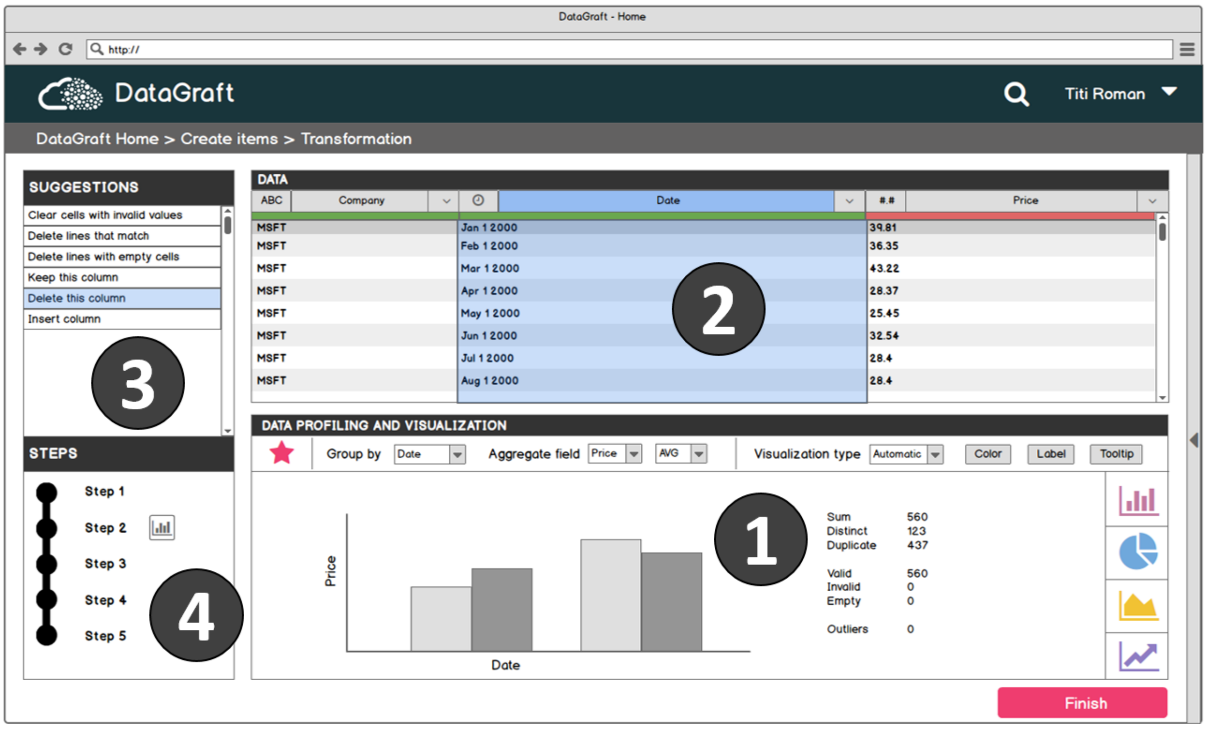
idLitholo	idPermeab.	IdAcidity	permeab-en	permeab-es	permeab-pt	acidity-en	acidity
101	3	2	High	Alta	Alta	Neutral	Neutro
102	3	2	High	Alta	Alta	Neutral	Neutro
10	1	3	Low	Baja	Baixa	Acid	Ácido
11	1	3	Low	Baja	Baixa	Acid	Ácido
12	1	3	Low	Baja	Baixa	Acid	Ácido
13	1	1	Low	Baja	Baixa	Basic	Básico
15	1	1	Low	Baja	Baixa	Basic	Básico
16	1	3	Low	Baja	Baixa	Acid	Ácido
210	1	1	Low	Baja	Baixa	Basic	Básico

DataGraft is a service operated by the DaPaaS project, co-funded by the EC under 7th Framework Programme (FP7 2007-2013)

Figure 13: Grafterizer user interface at the time of D3.1

Visual data profiling is the statistical assessment of datasets to identify and visualize potential quality issues such as data outliers or missing data values. The motivation behind the implementation of a visual data profiling approach is that the large datasets in euBusinessGraph require statistical assessment and analysis to achieve proper data quality. Visual data profiling has the potential to help data scientists make an informed decision on how to deal with data quality issues. Grafterizer implements this visual data profiling approach that provides capabilities such as a table view interface that lets the user manipulate columns and rows directly. Furthermore, relevant data cleaning and transformation functionality appropriately addresses the goals that the user tries to achieve.

To facilitate the requirements process, a wireframe was created to describe the user interface and functionality, and the needs of users, that lead to a set of requirements.



ABC Company	Date	#.# Price
MSFT	Jan 1 2000	39.81
MSFT	Feb 1 2000	36.35
MSFT	Mar 1 2000	43.22
MSFT	Apr 1 2000	28.37
MSFT	May 1 2000	25.45
MSFT	Jun 1 2000	32.54
MSFT	Jul 1 2000	28.4
MSFT	Aug 1 2000	28.4

Group by: Date
Aggregate field: Price, AVG
Visualization type: Automatic

Sum	560
Distinct	123
Duplicate	437
Valid	560
Invalid	0
Empty	0
Outliers	0

Figure 14: Grafterizer visual data profiling approach wireframe

The user interface of the visual data profiling approach illustrated in the wireframe in Figure 14, consists of the following main components and capabilities:

- A **visual data profiling** component (component 1);
- A tabular **table view** that provides data cleaning and transformation functionality (component 2);
- A sidebar that **suggests relevant data cleaning and transformation** actions to the user (component 3);
- A **steps pipeline** that reflects applied data cleaning and transformation steps (component 4).

The functionality of the visual data profiling approach in Figure 14 can be described in a sequence of steps that illustrate the visual data profiling cycle. As an example, consider a dataset with stock prices at various dates for five different companies (i.e. as shown in Figure 14):

1. The column 'Date' (component 2, middle column, highlighted) is selected in the table view.
2. Suggested transformations display in the 'Suggestions' sidebar (component 3).
3. The 'Data profiling and visualization' view provides a statistical assessment and profile of the data for the selected column 'Date' (component 1). A statistical profile of the data gives useful information to the user that can make an informed decision on how to deal with data quality issues in the dataset.
4. Optional: User may select any sections of the visualizations that will further suggest transformations for that specific section only, e.g., if the user selects a specific range of dates, the suggested transformations will be valid for this range only. This approach reduces the time taken to assess all the transformations that are applicable in that range of data.
5. Selecting a suggested transformation will add a transformation step to the pipeline and the table view will update to reflect the changes (component 4).
6. Repeat steps 1–5 to continue profiling, cleaning and transforming the dataset. The transformation sequence can be saved, and shared, for later reuse. The reuse of transformation sequences is particularly useful for periodical batch operations on data.

The functionality described in Figure 14 can be summarized as a set of requirements in Table 18 below.

Table 18: High-level requirements for Grafterizer 2.0

Req #	Description	Type
R1	Provide visual data profiling capabilities.	**F**
R2	Provide data cleaning and transformation functionality.	**F**
R3	Provide data cleaning and transformation suggestions.	**F**
R4	Provide a pipeline that reflects applied data cleaning and transformation steps.	**F**
R5	Provide a solution that is useful to the user.	**NF**
R6	Provide a solution that is easy to use.	**NF**
R7	Provide a robust beta release that can be deployed on the DataGraft platform.	**NF**
		F: Functional requirement **NF: Non-functional requirement**

4.2.3 Architecture

The high-level system architecture is based on a microservice architecture and uses the design principles of separation of concerns (SoC). SoC is traditionally achieved in layered architectures, e.g., in a 3-Tier architecture, by defining interfaces and encapsulating information. A 3-tier architecture (Figure 15) separates concerns into a presentation layer, an application tier, and a data layer.

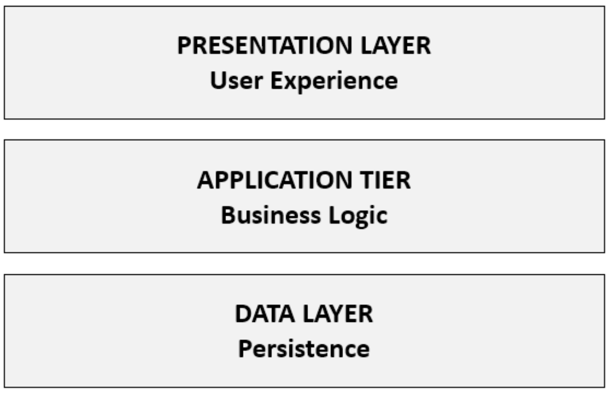


Figure 15: 3-tier architecture

Based on the considerations above, the architecture in Figure 16 shows a microservice approach that implements the design principles of separation of concerns (SoC). Angular 2 is selected as a development framework for building the frontend of Grafterizer 2.0 and its visual data profiling functionality, and the architecture is based on Angular 2 best practices for architectural patterns.

The green boxes in Figure 16 illustrate the backend services (application tier), while blue boxes show the frontend presentation layer. The data layer is implemented in the service through the integration with existing databases in DataGraft.

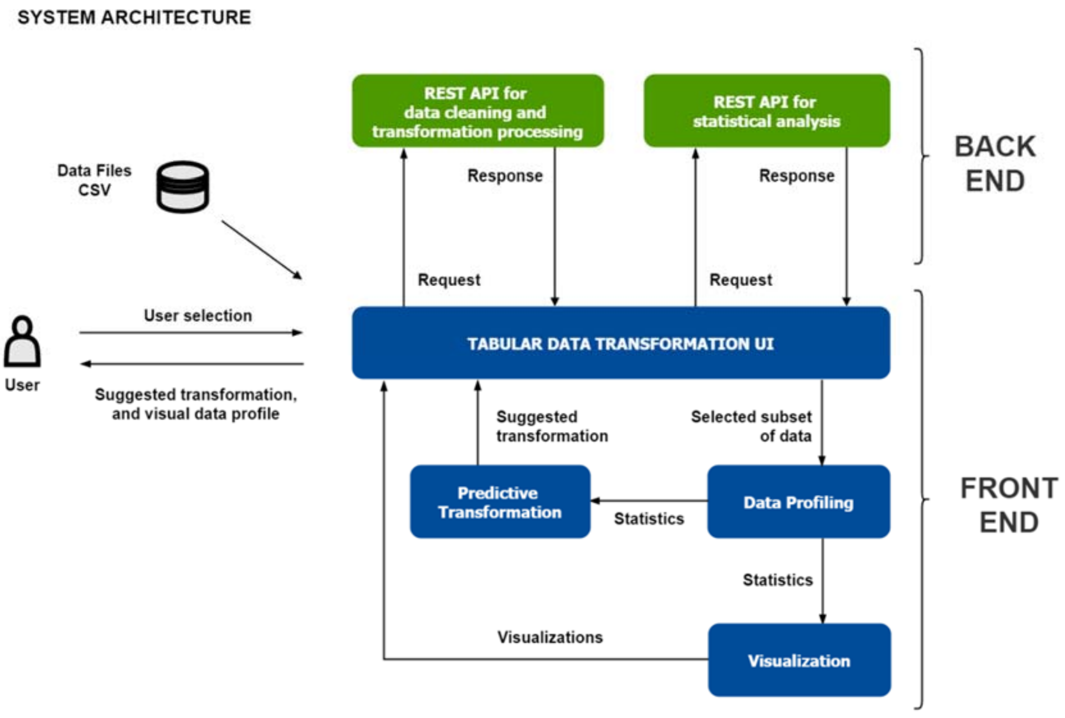


Figure 16: Grafterizer 2.0 visual data profiling microservice architecture

4.2.4 Functionality (to be used, enhanced or developed)

Table 19: Grafterizer 2.0 - Summary of features

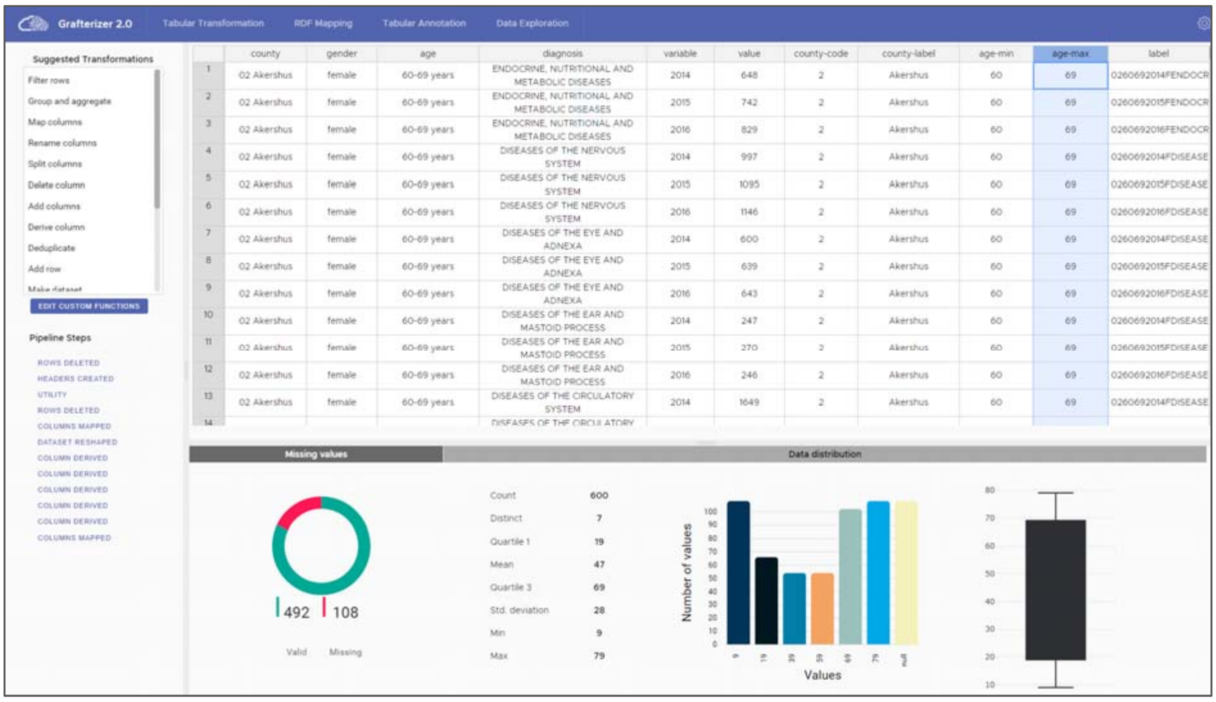
Feature	Description	Existing[17]	New[18]
Data Import	Grafterizer 2.0 currently supports import of CSV files. Files and transformations can be stored, shared, and reused in Grafterizer and DataGraft.	+	
Suggestion-based tabular data cleaning and transformation	Implementation of functionality for suggestion-based tabular data cleaning, visual data profiling for data quality assessment, and semantic annotation of CSV data to RDF knowledge graphs.		+
Tabular annotation	Implementation of service that provides semantic annotation of tabular CSV data to RDF knowledge graphs. This functionality is provided by ASIA and integrated into Grafterizer 2.0.		+
RDF mapping	Service that supports mapping of tabular CSV data to RDF knowledge graphs in a graphical tree-based interface.	+	
Data export	The output from Grafterizer can be exported to JAR executables, CSV, N-Triples and ArangoDB JSON collections.	+	

In euBusinessGraph, we have extended the Grafterizer framework with interactive data cleaning and transformation functionality, and visual data profiling. Figure 17 shows the **alpha version** (from

[17] Existing software feature brought as background to be used or enhanced in the euBusinessGraph project.
[18] New or enhanced software feature developed as a new foreground during the euBusinessGraph project.

deliverable D3.2) implementation of the wireframe from Figure 14. The implementation provides functionality for:

1. Suggestion-based data cleaning and transformation.
2. Visual data profiling.



	county	gender	age	diagnosis	variable	value	county-code	county-label	age-min	age-max	label
1	02 Akershus	female	60-69 years	ENDOCRINE, NUTRITIONAL AND METABOLIC DISEASES	2014	648	2	Akershus	60	69	0260692014FENDOCR
2	02 Akershus	female	60-69 years	ENDOCRINE, NUTRITIONAL AND METABOLIC DISEASES	2015	742	2	Akershus	60	69	0260692015FENDOCR
3	02 Akershus	female	60-69 years	ENDOCRINE, NUTRITIONAL AND METABOLIC DISEASES	2016	829	2	Akershus	60	69	0260692016FENDOCR
4	02 Akershus	female	60-69 years	DISEASES OF THE NERVOUS SYSTEM	2014	997	2	Akershus	60	69	0260692014FDISEASE
5	02 Akershus	female	60-69 years	DISEASES OF THE NERVOUS SYSTEM	2015	1095	2	Akershus	60	69	0260692015FDISEASE
6	02 Akershus	female	60-69 years	DISEASES OF THE NERVOUS SYSTEM	2016	1146	2	Akershus	60	69	0260692016FDISEASE
7	02 Akershus	female	60-69 years	DISEASES OF THE EYE AND ADNEXA	2014	600	2	Akershus	60	69	0260692014FDISEASE
8	02 Akershus	female	60-69 years	DISEASES OF THE EYE AND ADNEXA	2015	639	2	Akershus	60	69	0260692015FDISEASE
9	02 Akershus	female	60-69 years	DISEASES OF THE EYE AND ADNEXA	2016	643	2	Akershus	60	69	0260692016FDISEASE
10	02 Akershus	female	60-69 years	DISEASES OF THE EAR AND MASTOID PROCESS	2014	247	2	Akershus	60	69	0260692014FDISEASE
11	02 Akershus	female	60-69 years	DISEASES OF THE EAR AND MASTOID PROCESS	2015	270	2	Akershus	60	69	0260692015FDISEASE
12	02 Akershus	female	60-69 years	DISEASES OF THE EAR AND MASTOID PROCESS	2016	246	2	Akershus	60	69	0260692016FDISEASE
13	02 Akershus	female	60-69 years	DISEASES OF THE CIRCULATORY SYSTEM	2014	1649	2	Akershus	60	69	0260692014FDISEASE
14				DISEASES OF THE CIRCULATORY							

Count	600
Distinct	7
Quartile 1	19
Mean	47
Quartile 3	69
Std. deviation	28
Min	9
Max	79

Figure 17: Grafterizer alpha version implementation

Suggestion-based data cleaning and transformation recommends relevant next steps in the data preparation process. The data cleaning and transformation steps are incrementally applied in a pipeline approach. The transformations sidebar implements a rule-based system that suggests relevant data cleaning and transformation procedures by considering data type, and whether a column or row has been selected.

Table 20 shows some examples of the implementation of logic for suggesting transformations, based on a rules matrix.

Table 20: Rules matrix for examples of suggested transformations

String	Number	Date	Column	Row	Functionality
true	true	true	true	**false**	Insert a column to the right
true	true	true	true	**false**	Insert a column to the left
true	true	true	**false**	true	Insert a row above
true	true	true	**false**	true	Insert a row below
true	true	true	true	**false**	Delete a column
true	true	true	**false**	true	Delete a row
false	true	**false**	true	**false**	Set empty cells to zero
true	**false**	**false**	true	**false**	Set text to uppercase
true	true	true	**false**	true	Set the first row as headers
true	**false**	**false**	true	**false**	Pad digits
false	**false**	true	true	**false**	Reformat dates

The application checks the statistical data profile that is returned by the visual data profiling service against the rule's matrix illustrated in Table 20. Consider the following example:

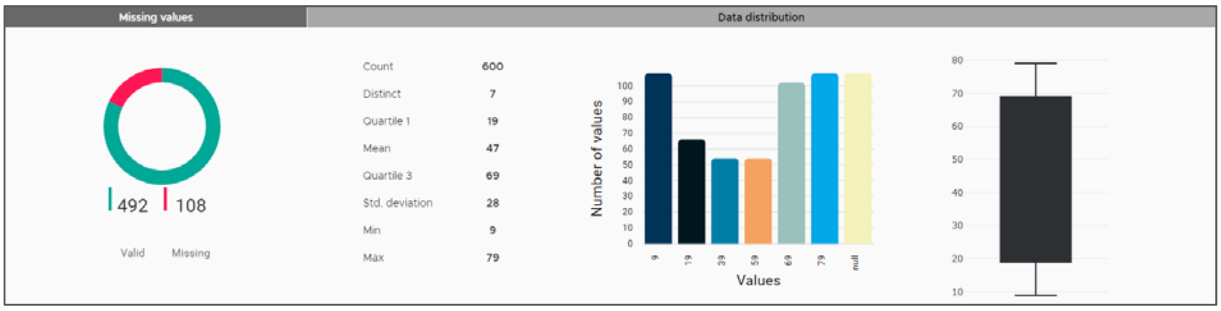
The data profiling service returns a profile of the current data selected in a column of string values. The String and Column values of the profile array are true, while the Number, Date, and Row values are false:

```
[true, false, false, true, false]
```

The application checks this profile against the enumerated list of transformations in Figure xx, and matches the profile array with the rules array function number 9, 'Set to uppercase', as a possible transformation that will be suggested:

```
[true, false, false, true, false],        // (9) Set to uppercase
```

Visual data profiling (Figure 18) 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.



Count	600
Distinct	7
Quartile 1	19
Mean	47
Quartile 3	69
Std. deviation	28
Min	9
Max	79

Figure 18: Grafterizer visual data profiling dashboard

The leftmost data profiling chart represents missing values and valid (non-null) values for the currently selected column. The three remaining charts (from left to right: table view, histogram, and box plot) represent the distribution of the currently selected column.

The visual data profiling service analyzes and assesses the quality of the dataset and returns a statistical profile. This profile is an essential part of the underlying core application logic that suggests transformations and renders profiling charts:

1. **Count**: the total number of values in the selected column.
2. **Distinct**: the number of unique values. As an example, a column attribute 'week' might count in total 1000 rows and 7 unique values, one for each day.
3. **Histogram**: an array containing one value for each histogram bin.
4. **Quality**: an array that contains three different values, one value representing valid entries, one for invalid entries and another one for outliers.
5. **Boxplot**: an array that contains all values necessary to render a boxplot chart, i.e., the first, second and third quartiles, and the median.
6. **Histogram labels**: labels for the histogram chart visualization.

4.2.5 APIs

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 and are maintained at:

- https://datagraft.github.io/datagraft-API/dist/index.html?url=https://datagraft.github.io/datagraft-API/swagger.yaml

4.2.6 Development plans for 2nd release (D3.4)

The second release – the **beta version** – will include implementation of the requirements listed in Table 21 which is needed to deploy Grafterizer 2.0 on the DataGraft platform.

Table 21: Detailed requirements for Grafterizer 2.0

Req #	Description	Type
R1	Provide visual data profiling capabilities: • Improve handling of datatypes to correctly display only charts and visualizations that are applicable to the user's current selection of data.	**F**
R2	Provide additional data cleaning and transformation functionality: • Add *Group and aggregate* pipeline function; • Add *Reshape dataset* pipeline function; • Add *Sort dataset* pipeline function; • Add *Derive column* pipeline function; • Add *Reshape dataset* pipeline function; • Add *Utility* pipeline function; • Add *Rename columns* pipeline function; • Add options for download of executable JARs (both CSV and RDF); • Add options for uploading user specific CSV files and accessing CSV files from the DataGraft assets repository.	**F**
R3	Provide data cleaning and transformation suggestions.	**F**
R4	Provide a pipeline that reflects applied data cleaning and transformation steps: • Implement pipeline functionality that allows the user to add, preview and delete steps of the transformation process.	**F**
R5	Provide a solution that is useful to the user.	**NF**
R6	Provide a solution that is easy to use: • Improve user experience and responsiveness by increasing performance and speed of Grafterizer. Alpha version is currently too slow and unresponsive.	**NF**
R7	Provide a robust beta release that can be deployed on the DataGraft platform: • Integrate Grafterizer 2.0 with the DataGraft platform; • Fix critical issues that are reported in Grafterizer GitHub repository; • Improve reliability and availability of microservices that Grafterizer depends on.	**NF**
	F: Functional Requirement **NF: Non-Functional Requirement**	

4.3 ASIA

ASIA is a semantic table enrichment tool integrated into Grafterizer 2.0, which provides Assisted Semantic Interpretation and Annotation of tables (.csv files). Semantic annotations create links from elements of a table to existing knowledge graphs (KGs). ASIA supports the creation of annotations at schema and instance level.

- **Schema-level annotations** link columns or column pairs of the table to types or properties used in a KG (in the following, we will use the term "type" to refer either to an RDFS class, e.g., foaf:Person, or to a datatype, e.g., xsd:double). For example, a user can annotate a column that contains company identifiers with the type dbo:Company, a class of the DBpedia ontology[19] used in a KG that describes companies. Schema-level annotations support what in D3.1 was referred to as the "schema alignment" step of semantic labeling;
- **Instance-level annotations** link individual values in the table to identifiers of entities described in a KG. For example, a user can annotate a column that contains the value

[19] http://wiki.dbpedia.org/services-resources/ontology

"Università Milano Bicocca" to the identifier dbr:University_of_Milano-Bicocca that describe UNIMIB in the DBpedia[20] KG. Instance-level annotations support what in D3.1 was referred to as "instances reconciliation" step of semantic labeling.

The annotations created for a table with ASIA support two functionalities:

- **RDF-ization**: annotations support the automatic transformation of the table into a KG represented in RDF, without requiring the specification of machine-readable mappings from CSV to RDF;
- **Content Enrichment**: instance-level annotations create links to identifiers of a reference KG; these linked identifiers can be used to fetch additional data from the KG or from other sources that use the same identifiers. For example, by linking company names to company identifiers in the euBusinessGraph KG, the user can use data enrichment widgets to fetch the CEO of the companies (where represented in the KG) and add this information to the table.

In ASIA, users can create annotations from a GUI, but they are assisted in this complex process by table interpretation algorithms. These algorithms are implemented as different interlinking services, which suggest annotations and help users make informed decisions when they transform the data into knowledge graphs.

In euBusinessGraph, ASIA is used for RDF-ization. Thus, in the rest of this section, we concentrate on this functionality.

Before describing requirements for ASIA in euBusinessGraph, we discuss the main characteristics of the RDF-ization process and the challenges for a semantic enrichment tool that aims at supporting this process.

Semantic table annotation for RDF-ization in euBusinessGraph. KGs represented using Semantic Web standards are represented using the RDF data model and are organized using shared vocabularies, possibly defined by ontologies modeled using RDFS and OWL languages. In euBusinessGraph, organizations that want to add their data to the business graph need to publish their data as KGs trying to interlink their data to the existing graph. In this process, two interlinking tasks are defined, each one with its own main objective.

- Schema-level interlinking has the objective of supporting providers to reuse, when possible, properties and classes and data types used in the business graph, so as to harmonize, at the schema-level, the new constituents of the business graph with the existing graph. If new data providers reuse vocabularies used in euBusinessGraph at a large extent, new data can be easily queried from the euBusinessGraph platform;
- Schema-level interlinking (alt. *value reconciliation*) has the objective of supporting providers to reconcile, when possible, values that occur in the table to standard or shared systems of identifiers used within the business graph data, e.g., identifiers of companies, locations, or administrative areas. When values are reconciled, the KGs published by the new data providers are linked to the rest of the graph at the instance level, thus supporting querying across different constituents of the business graph.

We add several important remarks to specify the objectives of table annotation in euBusinessGraph:

- **Need for assistance also for schema-level interlinking.** Performing even schema-level semantic annotation manually can be both resource- and time-consuming because it requires a domain expert that knows which are all the properties/types available in several KGs, and she has to choose one-by-one the correct property/type that better describe the semantics of each table column. The objective of ASIA is to ease the semantic annotation task by helping the user make an informed decision about the terminology to be used in the annotations, and, therefore, in the data that will be contributed to the business graph;
- **Schema-level interlinking and instance-level interlinking**. Schema-level interlinking is also useful to support the more difficult task of value reconciliation by making information available that is useful in the reconciliation process. For example, by specifying that a column lists values that belong to a type defined in a company ontology, a reconciliation algorithm may

[20] http://wiki.dbpedia.org/

compare the values in the column only against instances in the KGs that belong to the specified type (a technique often referred to as *blocking*). This and similar information (e.g., the specification of the RDF property of which values in the column will be objects) can improve scalability and accuracy of reconciliation algorithms;

- **Machine intelligence and the role of users in table annotation.** Table annotation and its subtasks, e.g., annotation of columns with RDF types and properties, reconciliation of values, etc., are complex and error-prone tasks. Automatic algorithms are needed to support the users in annotating a table, but users want to control the process and refine suggestions given by algorithms since it is well known that no automatic algorithm can be perfectly accurate. Table annotation in euBusinessGraph should therefore provide automatic algorithms that support the annotation process, as well as control via a user interface;
- **Flexibility in table annotation algorithms**. Reuse of existing vocabularies is important and useful for the reasons highlighted above, and algorithms used in table annotation should help users reuse existing vocabularies. However, there may be pieces of the data that are not adequately covered by existing vocabularies or data providers may prefer to use their own terminology in some cases, possibly by indirectly mapping it to other vocabularies (e.g., adding a new class as a subclass of an existing class). When a data provider introduces new terminology in a new piece of the business graph, this terminology may become useful to support new annotations in the future (by the same provider or other). In other words, usage of terminology in the business graph is an important driver for deciding which terminology to be used when publishing new data in the business graph; table annotation should encourage vocabulary reuse but allow for new terminology to be used.

ASIA vs STAN and ASIA 0.1. ASIA is developed starting from a previous tool named STAN (Semantic Table ANnotation), a prototype that suggests table annotations using only information about values occurring in the table, in such a way that: 1) the table annotation functionalities are integrated in Grafterizer 2.0, 2) the data manipulation tool used in the project, and 3) the logic described here above can be fully implemented. In this first release of ASIA, referred to as ASIA 0.1, we focus on the macro-level objectives listed here below:

- Develop ASIA as a table annotation component integrated into Grafterizer 2.0 and that can be modularly extended so as to provide schema-level and instance-level interlinking;
- Support schema-level interlinking incorporating information about terminology usage. To this end, ASIA is made interoperable with ABSTAT, a data summarization tool developed for RDF, which provides profiles of RDF datasets that report the usage of terminology in existing datasets.

4.3.1 Factsheet

Table 22: ASIA – Fact sheet

ASIA	
Functionality	ASIA (Assisted Semantic Interpretation and Annotation of tables) is a semantic table annotation tool to generate RDF knowledge graphs and enrich tabular data with semantic information possibly coming from third-party data sources. It supports *schema-level annotation* to map a table to an ontology in such a way that RDF data compliant with the ontology can be generated (RDF-ization), and *instance-level annotation* to reconcile values in the table to identifiers of entities described in existing knowledge or databases. The new identifiers are added to the original table to refine the RDF-ization process as well as to fetch even more data that can be found based on the new identifiers (data enrichment). To support schema-level and instance-level annotation, ASIA interoperates with vocabulary suggestion services (e.g., ABSTAT full-text search, Linked Open Vocabulary - LOV), reconciliation services (e.g., Wikifier or GeoNames) and data extension services (e.g., the GeoNames2LAU extension service). Finally, annotations are converted into data transformations operations that can be executed to deliver RDF-ization and data enrichment at scale. Semantic annotations are used to generate mappings from a table to a

	knowledge graph represented in RDF using a specific vocabulary, in such a way that the data contained in the table can be transformed into a Knowledge Graph. ASIA adopts a column-wise approach to semantic annotation, allowing users to define annotations for each column based on vocabulary suggestions provided by the tool. In the current version, ASIA incorporates suggestions from ABSTAT, a Knowledge Graph profiling tool, but can be configured to use LOV as well as other terminology recommendation services. ASIA is a new tool, with the following functionalities developed within euBusinessGraph: • User Interface for Schema-level Semantic Annotation of Tables: the user interface supports the creation of schema-level semantic annotation and is integrated into the Grafterizer tool to make the semantic annotation and RDF transformation processes integral to the data transformation steps; • Schema-level Vocabulary Suggestions: ASIA incorporates schema-level suggestions provided via API using ABSTAT, a knowledge graph profiling tool. It can be configured to use LOV and other terminology recommendation services; • Schema-level Data Transformations: based on the schema-level semantic annotations, data transformations are generated using the Clojure language, which can be executed by Grafterizer to generate RDF data. Features under development that are not deployed as part of the euBusinessGraph project: • User Interface for Instance-level Semantic Annotation of Tables: the user interface supports the creation of instance-level semantic annotation using reconciliation services that link values in the tables to identifiers of known knowledge graphs; • Instance-level Reconciliation Services: the reconciliation services support users in linking values in a table to identifiers of known knowledge graphs; • Data Transformations: based on the instance-level semantic annotations, data transformations generate linked knowledge graphs when transformations are executed.
Usage in euBusinessGraph	ASIA provides mapping suggestions that make it easier for new data publishers to map their data to the euBusinessGraph company data model.
Installation guide	https://github.com/UNIMIBInside/asia-backend
User guide	https://github.com/UNIMIBInside/asia-backend
API documentation	https://github.com/UNIMIBInside/asia-backend
Software license	GNU Affero General Public License v3.0
Source code repository	https://github.com/UNIMIBInside/asia-backend
Contact person	matteo.palmonari@unimib.it

4.3.2 Requirements (usage in euBusinessGraph)

The annotation of one column specifies the type for the values appearing in the column. In addition, if the column implicitly describes a relation, i.e., the values in the column are related to values of another column, the user needs to define this relation. In essence, to do so, she has to specify: 1) the RDF property that models the intended relation, and 2) the *source column* of the relation, i.e., the column from which subjects for the property will be selected. Thanks to this specification, it will be possible to generate a set of n RDF triples $< s_i, p, o_i >$ with $1 \leq i \leq n$ from the table such that:

- p is the RDF property used to represent the relation;
- values s_i are taken from the *source column*;
- values o_i are taken from the column that is being annotated;
- s_i and o_i occur in the i-th row of the table and will be respectively subject and object of the i-th triple generated from the table.

In other words, columns that describe values of a relation (and will provide objects for the relation) will be linked to the columns that describe the subjects of the relation. The ones described above are the basic elements of an annotation model, which specifies when annotations are valid, i.e., contain enough information to generate valid RDF triples. The model, whose details are omitted here, has been defined as an activity of euBusinessGraph.

Even if ASIA is integrated with Grafterizer 2.0, which allows users to manipulate tables before the RDF-ization, we cannot assume that users always want to RDF-ize the whole table manipulated with Grafterizer 2.0 - sometimes a user might be interested in RDF-ize only a subset of the table, without filtering it in advance. Since RDF-ization is a functionality provided by DataGraft, ASIA should produce annotations that can generate transformations compliant with the data transformation model used in Grafterizer 2.0.

Finally, ASIA should guide the user during the annotation task by providing procedures to *validate annotations* that ensure the validity of the RDF triples generated by the RDF-ization algorithm for a given set of annotations defined by a user.

All of the above requirements are summarized in Table 23.

Table 23: ASIA requirements

Req #	Description	Type
01	The user can choose arbitrarily which columns to annotate, and which ones not.	**F**
02	The annotation form must be validated in order to guide the user through the annotation task.	**F**
03	Columns that are not annotated must be filtered out from the RDF-ization process.	**F**
04	Each annotation must be complete enough to allow the RDF-ization process, that is each annotation form must be sufficient to produce a valid RDF.	**F**
05	The annotation model designed within ASIA must be compliant with the transformation model provided by Grafterizer 2.0, or alternatively, a mapping between the two models should be provided.	**F**
06	ASIA should provide suggestions about types and properties, and some statistics that can help the user in making decisions.	**F**
07	ASIA should provide automatic type- and properties annotations on columns based on information inferred from the headers.	**F**
08	Suggestions should be provided within an autocomplete function, that is the user will receive suggestions when she will start typing a property or a type.	**F**
09	The autocomplete functionality should fulfill real-time constraints.	**NF**
10*	A GeoNames reconciliation service to match location names to the GeoNames knowledge base must be available.	**F**
11*	A GeoNames2LAU data extension service must be created, which provides a complete map between locations of interest in Italy, Norway and UK to LAU codes for administrative level 4.	**F**
12*	Reconciliation services defined for OpenRefine, in particular the OCORP reconciliation service, must be usable in ASIA.	**F**
		F: Functional Requirement **NF: Non-Functional Requirement**

Some of the requirements in Table 23 (marked with an asterisk (*)) are related to specific reconciliation and extensions needs. In particular, it has emerged that there is a need to enrich the company data provided by OCORP and BRC with LAU codes. For this process to be supported, it is necessary to have the workings for reconciliation of location names against the GeoNames knowledge base and the acquisition of LAU codes that correspond to the found GeoNames identifiers available. Instance-level annotations are supported by reconciliation services. While a user can manually annotate the columns because they are limited in number, users require a service that automatically suggests links, which can then be refined by the users. The need for instance-level reconciliation is limited in euBusinessGraph, and most of the work to build instance-level functionalities is not carried out as a euBusinessGraph activity. However, these functionalities are available for usage in euBusinessGraph and are in fact needed to support an enrichment step in the data onboarding process: the addition of LAU code for administrative level 4 to the location specified in the data. Since owl:sameAs links between the GeoNames knowledge base and the LAU codes exist, and since ASIA provides a reconciliation service against GeoNames, to support this enrichment step we need a service that given a GeoNames id, which appears in a column i, returns a corresponding LAU code, and add this code in a column $i+1$, *by using* owl:sameAs links between GeoNames and LAU codes. We refer to this service as GeoNames2LAU extension. The links used in this service must be complete for the jurisdictions covered by BRC and OCORP data.

4.3.3 Architecture

Following the design principles of separation of concerns (SoC), ASIA is designed with a layered architecture. The Business Logic is developed within the ASIA Backend component, while the Presentation Layer contains the ASIA Frontend component. We describe in the following the two main components of ASIA, depicted in Figure 19:

- **ASIA Backend**, which contains the logic related to the annotation suggestion service and that is responsible for the intercommunication with third-party services (like ABSTAT), in particular:
    - ASIA Backend is currently integrated with LOV and ABSTAT autocomplete service via API; as a result, the Backend provides to ASIA, the real-time autocomplete service offered by ABSTAT as well as statistics computed by the same tool, thus fulfilling requirements 06, 07, 08.
- **ASIA Frontend**, which is the frontend customized and integrated within Grafterizer 2.0, and that contains two main modules:
    - Annotation Validation, which is the component that fulfills requirements from 01 to 04;
    - Annotation-To-Transformation, which is the component that fulfills the requirement 05.

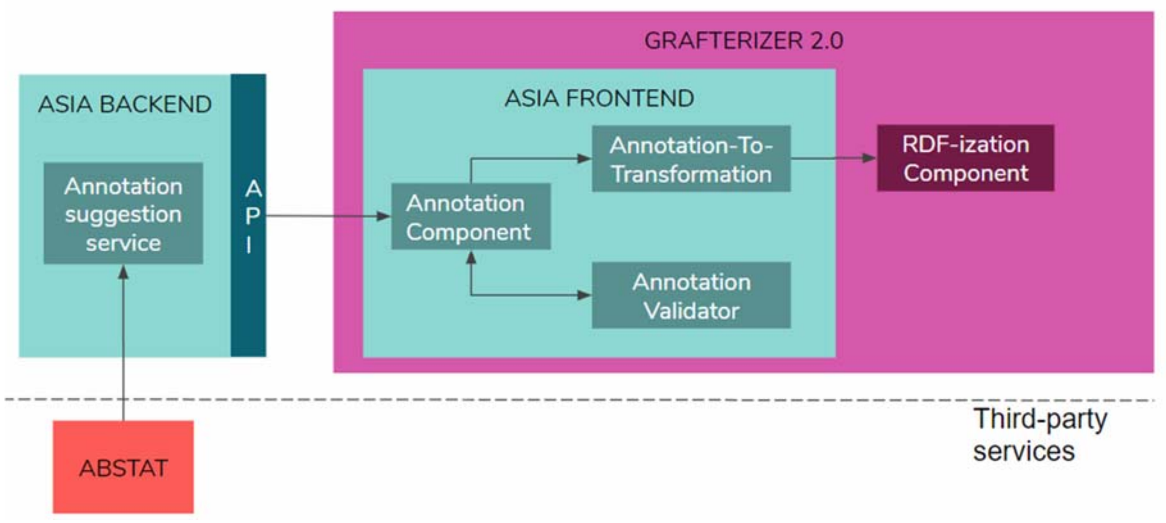


Figure 19: ASIA architecture

4.3.4 Functionality (to be used, enhanced or developed)

Table 24: ASIA - Summary of features

Feature	Description	Existing[21]	New[22]
Schema-level semantic annotation (schema-level linking)	Implemented in automated and manual mode. In the automated mode, each column is annotated (type and property) automatically based on its header. In the manual annotation mode, ASIA provides suggestions to the user. In both cases ASIA exploitsLOV and ABSTAT autocomplete services.		+
Instance-level semantic annotation: reconciliation[23]	Interface to use reconciliation services that are compliant with OpenRefine APIs (currently supported services are: Wikidata, Wikifier, OpenCorporate, GeoNames).	+	
Instance-level semantic annotation: data extension	Interface to use data extension services that are compliant with OpenRefine APIs (currently supported services are: Wikidata, OpenRefine, ECMWF Weather Enrichment, DBpedia, GeoNames, GeoNames2LAU).	+	
GeoNames2LAU extension service	Service that maps GeoNames identifiers to LAU codes (administration level 4) to enrich GeoNames data.		+
Data Transformation (RDF-ization and enrichment)	Annotations are converted into data transformations written in Clojure that can be applied to generate RDF data or enrich the input table.		+[24]

Given a table, ASIA provides an interface that guides the user through the annotation task using a column-wise approach. This approach allows users to define annotations for each column in such a way that the annotations encode enough information to automatically generate the transformations required for RDF-ization.

ASIA provides schema-matching functionalities both in automated and manual model. In the automated mode ASIA processes the header of each column to identify meaningful keywords. Space trimming, camelCase and underscore splitting are an example of the operations performed on the headers. Such keywords are used by the ASIA backend to invoke external services like LOV and ABSTAT to obtain a list of candidate annotations for each column. Both LOV and ABSTAT returns the candidates sorted on a likelihood criterion. ASIA selects the most likely annotation. The user is always responsible for validating the proposed annotation. Unvalidated annotations, in fact, will not be taken into account when creating the RDF of the table.

In many cases, however, the header is too cryptic and may not carry enough information to be exploited to get automatic annotation. For such cases, ASIA provides a syntactic autocomplete function, that is, an online tool capable to provide recommendations based on what the user is typing. In addition, along with suggestions ASIA provides additional information to help the user in making decisions (how to choose the correct property/type to be used?). ASIA provides suggestions and usage statistics by exploiting the LOV matching score and the ABSTAT summarization capability. An ABSTAT summary also includes statistics about patterns, properties, and types in the datasets.

[21] Existing software feature brought as background to be used or enhanced in the euBusinessGraph project.
[22] New or enhanced software feature developed as a new foreground during the euBusinessGraph project.
[23] In this list of features we do not list reconciliation and data extension services that are not developed within the euBusinessGraph project.
[24] Transformations based on schema-level annotations are the result of activities in euBusinessGraph, while transformations based on instance-level annotations are reused in euBusinessGraph.

Patterns and statistics model different aspects of the dataset, such as usage, domain, and quality. ABSTAT is useful for the semantic annotation as it supports a suggestion mechanism of predicates and types based on a context (the table headers) by looking on the summaries that it has produced.

When a user starts typing a type (but the same applies also for the properties case), ASIA suggests at most 15 types, which are the types that are most used in the KGs summarized by ABSTAT. Along with the type itself, ASIA provides also the name of the KG from which the type is kept, and the number of instances of that type that are found in the KG (occurrences). Types and properties statistics can help users to identify which suggestion is more probably to fit in the annotation context.

Each subsequent release of ASIA (and, eventually, the second release) will add layers of intelligence to its recommendation strategies, using richer evidence, for example, the relationships already stated between columns (that can be used to filter out some types/properties).

4.3.5 APIs

The ASIA Backend exposes a set of APIs, designed to reach high levels of modularity and maintainability. In particular, it is important to support the agile improvement of the interlinking algorithms used in the Backend of ASIA, so as to add layers of intelligence in the recommendation strategies. A complete description of ASIA API (which largely relies on OpenRefine) is available at https://github.com/UNIMIBInside/asia-backend.

4.3.6 Development plans for 2nd release (D3.4)

First, we will integrate LOV suggestions into ASIA. Second, starting from the next release, ASIA suggestions will be provided others recommendation strategies based on other pieces of evidence, for instance, the values occurring into a column or the relationships already stated between columns. The process of adding intelligence to ASIA is pursued with an agile methodology, being the main objective of the activities in this project to build a robust tool that can be used by the users (thus favoring implementation of tool functionalities over providing novel methods that work in academic experimental settings). In addition, instance-level linking services using company identifiers in the graph will be integrated into ASIA.

4.4 ABSTAT

ABSTAT (Linked Data Summaries with ABstraction and STATistics) is a framework developed by UNIMIB to support users and machines in better understanding big and complex RDF datasets. Given an RDF dataset and, optionally, an ontology (used in the dataset), ABSTAT computes a summary (profile) that provides an abstract but complete description of the dataset content and several statistical descriptors. Summaries are published and made accessible via web interfaces, in such a way that the information that they contain can be consumed by human users and machines (via APIs). ABSTAT also makes use of a minimalization mechanism to keep summaries complete but as small as possible.

ABSTAT summaries provide answers to such questions as: What types of resources are described in a dataset? What properties are used to link the resources? What types of resources are linked and by means of what properties? How many resources have a certain type and how frequent is the use of a given property? In practice, ABSTAT summaries describe the use of vocabularies in datasets. Moreover, statistics produced by ABSTAT can be used to evaluate the general quality of the dataset. ABSTAT is a backend infrastructure that is aimed at supporting different tasks:

- **Data understanding**: ABSTAT summaries provide a complete overview of the content of a dataset; this feature was proved to be useful to support, for example, SPARQL query formulation;
- **Vocabulary matching for table annotation**: Summaries record rich statistics about the usage of vocabularies/ontologies in datasets. Thus, we can use summaries to provide types and properties that match a string, for example, the header of a column in a table that we want to publish in RDF reusing existing vocabularies. In addition, statistics provide valuable information to algorithms aimed at suggesting the best properties and types to use when transforming a tabular data into RDF;

- **Data quality**: Summaries and cardinality profiles can detect data quality issues in RDF knowledge graphs. For example, to find quality issues in company-related data in DBpedia, one can look at ABSTAT's profiles. One would find that there is one instance of the concept owl:Thing that is specified as the object of the dbo:keyPerson property of as many as 5263 different companies (dbo:Company). If we combine this evidence with the evidence that the average number of distinct objects associated via the dbo:keyPerson property to instances of dbo:Company is three, we can conclude that there is an outlier in the dataset. A query to the knowledge graph would explain that the instance of owl:Thing specified as key person of 5263 is a generic instance named dbr:ceo, which is associated in DBpedia to companies that have no other key person associated with. To sum up, ABSTAT profiles can support the identification of anomalous schema-patterns.

The first version of ABSTAT is the result of research activities carried out at UNIMIB and was developed as a research prototype. Such a version is not suitable to be integrated into a production environment. For this reason, a new and more robust, version of ABSTAT is created in euBusinessGraph. The new version of ABSTAT will incrementally include algorithms developed for the research prototype. These algorithms compute for example more sophisticated statistics, such as cardinality of relations, number of instances (which is defined considered the semantics of the ontology where types and properties are defined) and more.

4.4.1 Factsheet

Table 25: ABSTAT – Fact sheet

ABSTAT	
Functionality	ABSTAT (Knowledge Graph Profiling with ABstraction and STATistics) is a tool to create, manage and provide access to semantic profiles of knowledge graphs. Semantic profiles describe structural properties of knowledge graphs and the vocabulary used therein. Semantic profiles are applied to support different tasks: data understanding and exploration, data quality, vocabulary recommendation and analytics. ABSTAT is currently developed to profile RDF-based knowledge graphs. Access to information stored in profiles is accessible to users, by means of web-based interfaces, as well as to machines, by means of suitable APIs. Features available before the project: • Core profiling algorithms: algorithms for the extraction of patterns and the computation of pattern-specific statistics; • ABSTAT Frontend: web application to access profiles (ABSTAT Browse, ABSTAT Search, ABSTAT Query). Features developed during the project: • ABSTAT Frontend: user-oriented web application to control the profiling process and compute, store and manage profiles; • Cardinality descriptors' algorithms: the algorithms to compute cardinality descriptors; • ABSTAT Distributed Architecture: the distributed architecture used to control, compute, and store profiles, as well as to represent and share them via the HTTP protocol. In particular, the new architecture uses NoSQL database to store profiles for structured access and ElasticSearch for full-text search over the profiles; • ABSTAT APIs: API to make profiles accessible for machines, including vocabulary suggestion APIs used in the ASIA semantic table annotation tool.
Usage in euBusinessGraph	ABSTAT provides profiles of company data already published in the business graph that are used by ASIA in order to provide mapping suggestions that makes it easier for new data publishers to map their data to the euBusinessGraph company data model. In addition, ABSTAT can be used to spot quality issues

	such as the occurrence of outliers or patterns that should not occur based on the expected schema.
Installation guide	A guide to installing ABSTAT can be found in the Readme.me file in the ABSTAT repository on Bitbucket (https://bitbucket.org/disco_unimib/abstat).
User guide	https://bitbucket.org/disco_unimib/abstat
API documentation	http://backend.abstat.disco.unimib.it/apis
Software license	https://www.gnu.org/licenses/agpl-3.0.html
Source code repository	https://bitbucket.org/disco_unimib/abstat/src
Contact person	matteo.palmonari@unimib.it

4.4.2 Requirements (usage in euBusinessGraph)

The requirements for ABSTAT are described in Table 26. Each requirement is identified by a unique code, so it can be referenced when needed.

Table 26: ABSTAT requirements

# Req	Description	Type
01	The system shall allow users to upload datasets and ontologies.	**F**
02	The system shall allow users to trigger a summarization with datasets and ontologies that were uploaded as specified in 01. Computed summaries should be archived in a persistent backend storage.	**F**
03	The summarization process can be configured by a user, who can specify parameters of the summarization (measures to compute, etc.).	**F**
04	A user shall be able to load a summary on a persistent storage, which supports structured queries over the summaries via API (09) or via a GUI (06).	**F**
05	The user shall be able to index a summary on a search engine in order to support full-text search via a GUI (07).	**F**
06	A GUI should support browsing of summaries. Browsing consists in executing constrained queries over one or more summaries specified by the user (e.g., by specifying a desired subject and/or predicate and/or object).	**F**
07	A GUI should support full-text search over one or more summaries specified by the users. Different ranking functions can be supported.	**F**
08	The system shall allow users to download summaries from ABSTAT backend storage even without prior 04 and 05.	**F**
09	The system should support constrained queries over one or more summaries via APIs.	**F**
10	The system should provide an API that implements an autocomplete mechanism using vocabularies that appear in the set of indexed datasets. Autocomplete should take a string as input and other optional pattern-level[25] constraints and return vocabulary terms (predicates and types) guaranteeing interactive time.	**F**
11	ABSTAT should support the identification of schema-level anomalous schema-pattern.	**F**
12	The backend storage (FR02) should contain summaries in a flexible and agnostic	**NF**

[25] Patterns, also named *abstract knowledge patterns* (akp), are the basic information primitive in ABSTAT summaries.

	data format in order to make easy post manipulations.	
13	The autocomplete API (FR10) should guarantee an interactive time as it should support real-time autocomplete.	**NF**
14	The key profiling processes (minimal pattern extraction, frequencies and cardinality descriptors) should be executed in a scalable manner to profile a large amount of data.	**NF**
		F: Functional Requirement **NF: Non-Functional Requirement**

4.4.3 Architecture

ABSTAT is designed to be modular and decoupled in order to support different scenarios of use. For example, in one scenario, summaries are computed, published, and made accessible to users using the same node in a network. In another scenario, one node may compute the summaries, which, once computed, are then transferred to a different node that publishes the summaries and makes them accessible to users. In Figure 20, we can distinguish five main components.

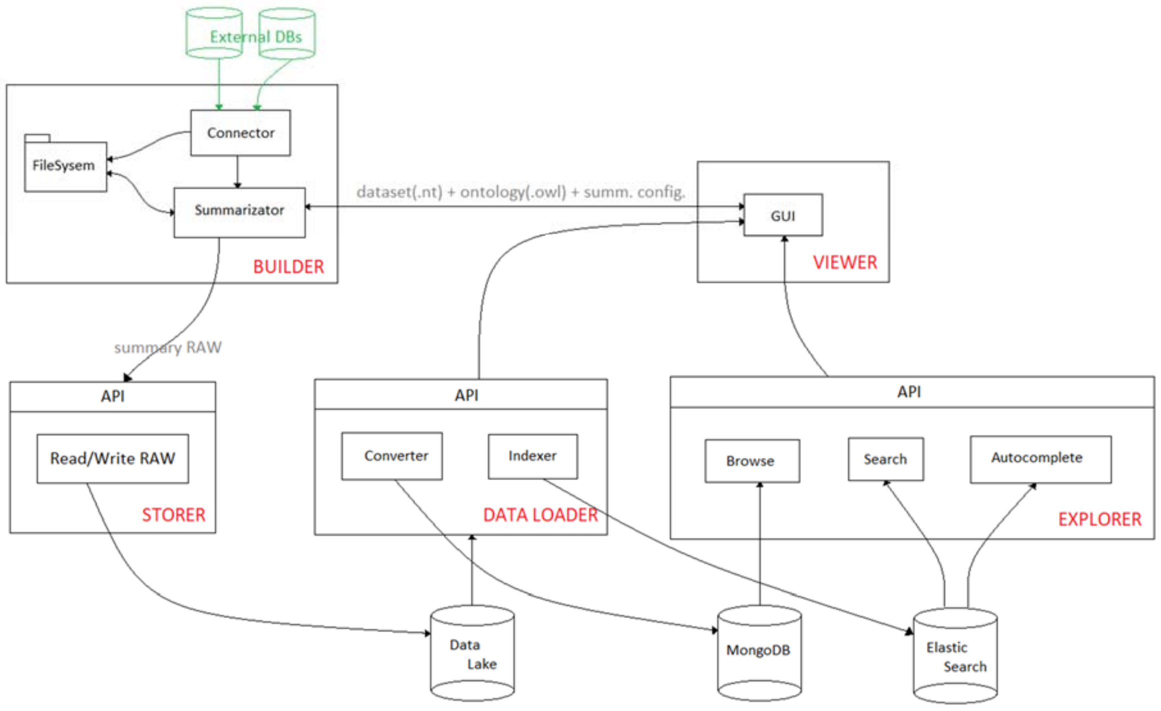


Figure 20: ABSTAT architecture

The ABSTAT Viewer provides a graphic user interface to serve different types of jobs such as summary exploration, execution of the summarization process using a wizard and indexing of summaries after execution. Summary exploration can be performed using constrained queries (by a desired subject and/or predicate and/or object) and full-text search. Both exploration modes can be run over one or more summaries. The summarization wizard provides a GUI to let users select datasets and ontologies from a list or using an upload module, configure the summarization process, and launch it. After the summary is computed, the user can load/index it on a persistent storage/search engine in order to support access to the summary through ABSTAT APIs or GUI. Indexing is required for full-text search.

The ABSTAT Builder is the component that executes the summarization algorithms and produces the summaries. The Summarizator sub-component requires as input a dataset (in N3[26] format) and an ontology (in owl format) along with the configuration chosen by the user. If the user does not provide

[26] https://www.w3.org/TeamSubmission/n3/

an input file because data are in a DB, the Connector sub-component allow for extracting a dump and storing it in the correct file to serve as input to the Summarizator.

The ABSTAT Storer component feeds a data lake storage with the raw data produced by the Builder. It also receives download requests from users who want to get raw summaries.

The ABSTAT Loader contains the Converter sub-components, which convert data available in formats different from N3 into the N3 format, so as to allow for summarization of RDF data represented using different formats. The Indexer sub-component indexes summaries in a search engine. Note that the Loader component receives the control input from the Viewer.

The ABSTAT Explorer is organized as a set of APIs to satisfy summary exploration requests from Viewer or users who want to use them directly.

Earlier versions of ABSTAT were developed for research purposes and made publicly available. However, the version of ABSTAT released as part of this deliverable can be considered the first official release made available on Bitbucket at https://bitbucket.org/disco_unimib/abstat with a license. For this reason, the version of ABSTAT released as part of this deliverable is numbered ABSTAT 0.1[27].

4.4.4 Functionality (to be used, enhanced or developed)

Table 27: ABSTAT – Summary of features

Feature	Description	Existing[28]	New[29]
Core profiling algorithms (pattern extraction and frequency) and statistics	Algorithms for the extraction of patterns and the computation of pattern-specific statistics like frequency.	+	
Full-text search	Full-text search over a set of indexed profiles.	+	
ABSTAT Frontend: Browse, Search, Query	Web application to access profiles (ABSTAT Browse, ABSTAT Search, ABSTAT Query).	+	
ABSTAT Frontend: Profiling Control Panel	A user-oriented web application to control the profiling process and compute, store and manage profiles.		+
Real-time vocabulary autocomplete using KG summaries	Implemented as a dedicated service accessible via API.		+
Full-text search and Browse APIs	Interface to use the ABSTAT full-text search component and access to profile information.		+
Cardinality descriptors algorithms	Algorithms to compute cardinality descriptors, i.e., statistics about the cardinality of properties for a given pattern.		+
ABSTAT Distributed Architecture	The distributed architecture used to control, compute, and store profiles, as well as to represent and share them via the HTTP protocol. In particular, the new architecture uses a NoSQL database to store profiles for structured access and Elastic Search for full-text search over the profiles.		+

[27] The architecture depicted in Figure 20 and its components have been developed for this release. Only core summarization algorithms have been reused from the earlier research prototype.
[28] Existing software feature brought as background to be used or enhanced in the euBusinessGraph project.
[29] New or enhanced software feature developed as a new foreground during the euBusinessGraph project.

ABSTAT (Linked Data Summaries with ABstraction and STATistics) is a framework that computes and provides access to synthetic descriptions of RDF datasets. These descriptions can be viewed as profiles of RDF datasets and are called summaries in ABSTAT.

The summary of a dataset describes its content by listing every schema-level pattern that occurs in the data and a set of statistics. A schema-level pattern, named also abstract knowledge pattern (akp) or, more simply, *pattern*, is a triple <Type_1,P,Type_2> that tells there are instances of Type_1 linked to instances of Type_2 with the property P. With the term *type*, we refer to either an ontology class (e.g., foaf:Person) or a datatype (e.g., xsd:DateTime). In addition to patterns, summaries provide several *statistics* for the patterns and their constituents, i.e., types and properties. The version of ABSTAT that is deployed in euBusinessGraph computes and includes statistics about the occurrence of patterns, types, and properties. Occurrence assigned to a pattern <Type_1,P,Type_2> tells how many instances of Type_1 are linked to instances of Type_2 with the property P.

From large datasets that use rich type hierarchies the number of patterns that can be extracted can be very large. To provide more compact summaries ABSTAT can use the ontology used in the dataset to extract a minimal set of specific patterns and discard patterns that are redundant. The minimal set of specific patterns (also referred to as *Minimal-Type Pattern Base - MTPB*) is sufficient to reconstruct the complete set of patterns that can be extracted from a dataset by inferring redundant patterns. The mechanism used in ABSTAT to compute the MTPB using an ontology specified by the user is known as *minimalization*. The ontology is used to define which are the most specific types of instances that occur in the summarized triples. In this way, it is possible to extract from each RDF triple only the patterns that are more specific, i.e., the patterns that contain the most specific types of the instances occurring in the triple. When minimalization is used, the intuitive meaning of a pattern <Type_1,P,Type_2> with occurrence n is that there are n RDF triples $< s, P, o >$ such that s and o have Type_1 and Type_2 respectively as minimal type.

If the user specifies the main pay-level domain of interest in the dataset (e.g., dbpedia.org for DBpedia_2014 dataset), ABSTAT can distinguish between resources (patterns, types and properties) that are *internal* (resources having the specified pay-level domain) and *external* (resources having a pay-level domain different from the one specified by the user). This distinction has the only purpose of letting users filter out patterns that include some external resource, which can be useful in some cases (e.g., the user may want to hide all patterns that contain the type foaf:person when looking at patterns extracted from DBpedia).

More details about the minimalization process, its effect on summaries' size, and an evaluation of the informativeness of summaries can be found in [1].

Functionalities of ABSTAT currently deployed in the euBusinessGraph therefore include:

- Summarization of RDF data in N3 format with the extraction of patterns and occurrence statistics (core functionality of the tool);
- Configuration and launch of the summarization algorithm via GUI with storage in MongoDB (addressing requirements 01, 02, 03, 04, 11);
- Indexing of computed summaries via GUI (addressing the requirement 05);
- Browsing and full-text search using a browser (addressing requirements 06, 07);
- Access to summaries via APIs (addressing the requirement 09);
- Autocomplete service over arbitrary strings (addressing requirements 10, 12).

Functionalities that will be implemented in the second release are described in Section 4.4.6.

References (scientific publications providing details about ABSTAT functionalities and algorithms):

[1] Azzurra Ragone, Paolo Tomeo, Corrado Magarelli, Tommaso Di Noia, Matteo Palmonari, Andrea Maurino, Eugenio Di Sciascio: **Schema-summarization in linked-data-based feature selection for recommender systems.** SAC 2017: 330-335.

4.4.5 APIs

The ABSTAT Storer exposes a set of APIs which allows the GUI and the users to navigate the available summaries. Here we provide a summary of the APIs. For more about APIs read the description on ABSTAT website[30].

4.4.5.1 Data consumption APIs

- **/api/v1/summaries**: Lists available summaries;
- **/api/v1/datasets**: Lists available datasets;
- **/api/v1/ontologies**: Lists available ontologies;
- **/api/v1/SolrSuggestions**: Allows the user to get suggestions based on the specified constraints;
- **/api/v1/browse**: Allows the user to list all AKPs which match the input constraints;
- **/api/v1/SPO**: Allows the user to list all the subject/predicate/object of a given summary;
- **/api/v1/search**: Allows a full-text search over concepts/datatypes/AKPs;
- **/api/v1/groupedExtractor**: Allows to retrieve the instances in a grouped way that violate a pattern shacl-constraint;
- **/api/v1/singleExtractor**: Allows to retrieve all the instances given an (entity/predicate/objectType) or (subject Type, predicate, object) triplet.

4.4.5.2 Control APIs

- **/api/v1/upload/ds**: Allows dataset upload;
- **/api/v1/upload/ont**: Allows ontology upload;
- **/api/v1/summarizator**: Allows the customization and submission of a summarization process;
- **/api/v1/consolidate**: Allows to consolidate the produced summary in the document-based storage or in the search-engine;
- **/api/v1/dataset/delete/{id}**: Removes chosen dataset;
- **/api/v1/ontology/delete/{id}**: Removes chosen ontology;
- **/api/v1/submitconfig/delete/{id}**: Removes chosen summary.

4.4.6 Development plans for 2nd release (D3.4)

In the next release, ABSTAT will include different features:

- Improved GUI, in particular for control of summarization and upload of datasets from other RDF storage systems (in particular, OntoText GraphDB);
- Inclusion of more advanced summarization options, previously included in the research prototype (pattern inference, cardinality profiles, minimalization over properties);
- Comparison of summaries and RDF shape specifications, e.g., in SHACL[31], for data anomalies detection and quality evaluation;
- Distributed architecture with more local Builders that compute summaries stored and accessed from centralized storage;
- Implementation of a distributed algorithm for the calculations of minimal patterns frequency, instances and cardinality descriptors.

[30] http://backend.abstat.disco.unimib.it/apis
[31] https://www.w3.org/TR/shacl/

5 Cross-Cutting Business Analytics Services

Table 28 lists the software components that will implement the Cross-Cutting Business Cases Analytics Services depicted in Figure 7.

Table 28: Software components – Cross-Cutting Business Cases Analytics Services

Software component	Functionality	Implementation status
Wikifier	Wikifier is a Web service which takes a text document as input and annotates text chunks with links to respective Wikipedia concepts.	The service is ready. Subsystems are in development.
Event Registry	Event Registry is a system that collects and analyses news articles from over 30.000 news publishers and identifies mentioned world events in the collected articles.	The service is ready. Subsystems are in development.
Graph-based analytics	The aim of the Graph-Based Analytics API service is to cover: • Event type categorization; • Entity extraction.	Prototype in development.
TWEC	Measure the similarity between words and entity identifiers.	Prototype in development.

5.1 Wikifier

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.

Wikipedia as a source of possible semantic annotations can be treated as a large and fairly general-purpose ontology. Each page is thought of as representing a concept, while the relations between concepts are represented by internal hyperlinks between different Wikipedia pages, as well as by Wikipedia's category membership and cross-language links. Because Wikipedia is available in a number of languages, with cross-language links being available to identify pages that refer to the same concept in different languages, it is much easier to support multilingual and cross-lingual annotation.

5.1.1 Factsheet

Table 29 presents the Wikifier factsheet, along with development plans.

Table 29: Wikifier – Fact sheet

Wikifier	
Functionality	Wikifier is a Web service which takes a text document as input and annotates text chunks with links to respective Wikipedia concepts.
Usage in euBusinessGraph	JSI’s semantic multilingual annotation service will be based on JSI Wikifier. JSI Wikifier is used by euBusinessGraph consortium partners for semantic text annotation. The development and usage plans for Wikifier service include the introduction of extra vocabularies (obtained from the business data, such as company data) and usage of Wikifier for the relation observation and relation extraction task.
Installation guide	http://wikifier.org/
User guide	http://wikifier.org/info.html
API documentation	• https://github.com/JSI-EuBusinessGraph/jsi-wikifier-api • http://wikifier.ijs.si/info.html

Software license	JSI Wikifier runs as a RESTful API service, for which a user key is required. The usage of JSI's Wikifier is governed by fair use.
Contact person	janez.brank@ijs.si

5.1.2 Requirements (usage in euBusinessGraph)

The requirements for Wikifier service are provided in Table 30 below.

Table 30: Requirements – Wikifier

ID	Requirement	Description
Wikifier		
Req1	Semantic annotation	The service should provide semantic annotation of textual input.
Req2	Cross-linguality and Multilinguality	The text document for semantic annotation should be in one of the supported languages. This list of supported languages consists of all languages for which, as of this writing, a localized Wikipedia with at least 10000 pages is available (from the List of Wikipedias), plus a small number of other languages.
Req3	Web API	The service is accessible via RESTful API (with an access key). The users should register at the Wikifier website.
Req4	Extra vocabularies	The service should embed a set of business-related extra vocabularies (such as company vocabularies).
Req5	Relation observation and relation tracking	The service should be used for observation and tracking of relations between entities - for instance, between companies and people.

5.1.3 Architecture

Figure 21 describes the main components of Wikifier architecture:

- WikiData (a free and open knowledge base);
- In-memory stores (that incorporate WikiData graph, Wordnet and Extra vocabularies. WordNet (https://wordnet.princeton.edu/) is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations;
- Processing and disambiguation mechanisms based on PageRank. PageRank (PR) is an algorithm used by Google Search to rank websites in their search engine results;
- User access via a Web interface and RESTful API.

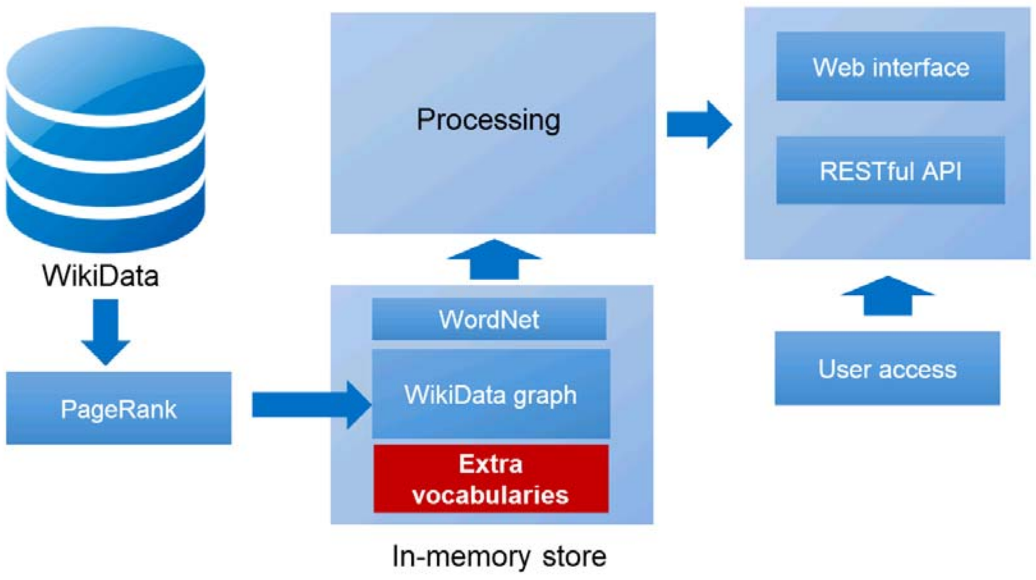


Figure 21: Wikifier architecture

5.1.4 Functionality (to be used, enhanced or developed)

Table 31: Wikifier – Summary of features

Feature	Description	Existing[32]	New[33]
Multi-lingual annotation	Semantic text annotation with Wikipedia concepts.	+	
Annotation with extra vocabularies	Vocabulary based annotation for the financial business domain.		+
Derived entity label generation	Wikifier preprocessor for deriving new entity labels from complex business entity labels, which increases text search recall.		+

JSI's Wikifier runs as a Web service and is accessible via API on the following URL: http://www.wikifier.org/annotate-article. When querying the service, many worker threads can run in parallel, as the internal data structures built from Wikipedia are read-only and can be shared between threads. This allows for highly efficient processing of a large number of documents. Our implementation currently processes on average more than 500K requests per day.

Wikifier currently supports a total of 134 languages. These are all languages in which a Wikipedia with at least 1000 pages is available. Annotations are returned in JSON format and can optionally include detailed information about mentions' support, alternative candidate annotations and WikiData/DBpedia class membership of the proposed annotations.

The motivation for derived entity label generation is two-fold:

- The entities from the datasets that we have analyzed in general come with only one complex string label. However, what is often found in unstructured data (e.g., news articles) is not necessarily the full title of a company or a product. For example, journalists often refer to company entities by their brand name, trademark or common name, which are often substrings of the full entity label. News search should support annotation of substring variations of entity labels;

[32] Existing software feature brought as background to be used or enhanced in the euBusinessGraph project.
[33] New or enhanced software feature developed as a new foreground during the euBusinessGraph project.

- Some entity labels have very common dataset-specific substrings, such as company titles having a suffix that describes their corporate status (e.g., INC.). These types of dataset-specific artifacts can lower search recall in unstructured data.

The figure below presents an illustration of the interactive decision-making utility for setting the threshold on the process of derived entity label generation.

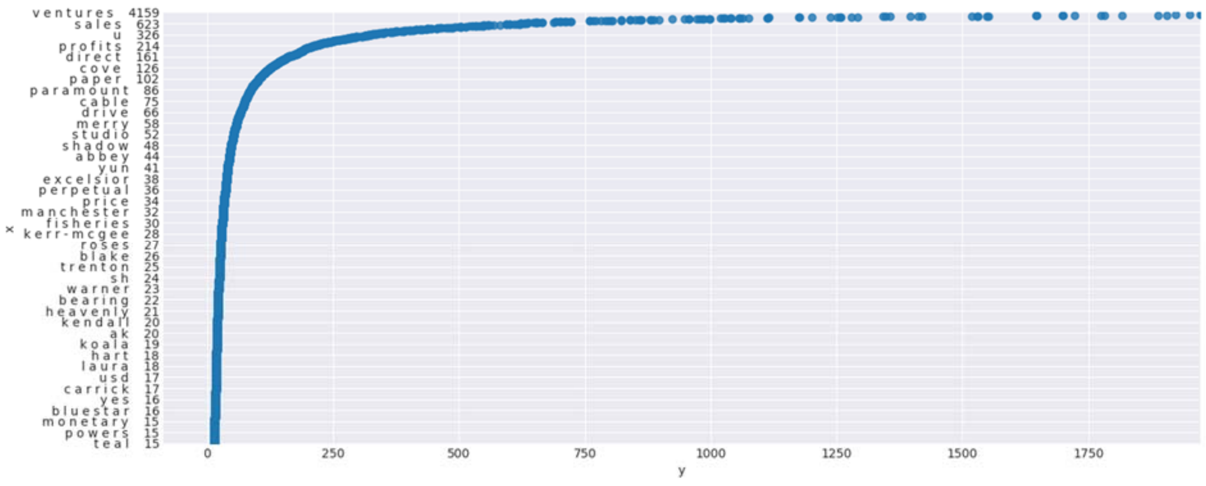


Figure 22: An interactive decision-making utility for setting the token frequency threshold

The output of the service is used as a preprocessing step by Event Registry and Graph-based analytics services. Since Wikifier is an important service that tackles cross-linguality and multilinguality, the news and events represented in different languages can be processed with this service and later analyzed in cross-cutting business cases analytics tasks.

5.1.5 APIs

For Wikifier, we are continuing to develop the Web API documentation, which is written according to the OpenAPI 2.0 standard and is available online.[34] Each request to the Web API must contain a `userKey` string value that users can obtain for free by registering on the service's Web site.[35]

If users include the correct **`extraVocabularies`** parameter value, Wikifier enables non-Wikipedia vocabularies (e.g., company names from company registers) for the annotation procedure. The Web API call then returns an additional array of entities in JSON under the key `extraVocabularies`. The annotations produced with additional vocabularies, for now, do not use context provided in entity properties (e.g., company description) for entity disambiguation. The accepted `extraVocabularies` values are not public, as the vocabularies are still under development – the feature can be used upon request.

JSI Wikifier can be called with HTTP GET request to a URL of the following form:

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

or a POST request with a Content-Type of application/x-www-form-urlencoded and pass the parameters (everything after the ? character) in the request body.

The Wikifier returns a JSON response of the following form:

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

[34] https://github.com/JSI-EuBusinessGraph/jsi-wikifier-api
[35] http://www.wikifier.org/register.html

```
  "ranges": [ ... ]
}
```

The API documentation can be found on the following links:

https://jsi-eubusinessgraph.github.io/jsi-wikifier-api and http://wikifier.ijs.si/info.html

Table 32: Wikifier API parameters

Parameter	Description
`userKey`	A 30-character string that uniquely identifies each user (register here to get one). This parameter is required.
`Text`	The text of the document that you want to annotate. Use UTF-8 and %-encoding for non-ASCII characters (e.g., `text=Beyonc%C3%A9`).
`Lang`	The ISO-639 code of the language of the document. Both 2- and 3-letter codes are supported (e.g., `en` or `eng` for English, `sl` or `slv` for Slovenian, etc.). You can also use `lang=auto`
`secondaryAnnotLanguage`	For each annotation, the Wikifier can report, in addition to its name and Wikipedia link in the Wikipedia for the language of the input document, also the name and link of the corresponding page in the Wikipedia for a “secondary” language; the `secondaryAnnotLanguage` parameter specifies the code of this secondary language (default: en, i.e. English).
`wikiDataClasses`	Should be `true` or `false`, determines whether to include, for each annotation, a list of wikidata (concept ID, concept name) pairs for all classes to which this concept belongs (directly or indirectly).
`wikiDataClassIds`	Like `wikiDataClasses`, but generates a list of concept IDs only (which makes the resulting JSON output shorter).
`Support`	Should be `true` or `false`, determines whether to include, for each annotation, a list of subranges in the input document that support this particular annotation.
`Ranges`	Should be `true` or `false`, determines whether to include, for each subrange in the document that looks like a possible mention of a concept, a list of all candidate annotations for that subrange. This will *significantly* increase the size of the resulting JSON output, so it should only be used if there's a strong need for this data.
`includeCosines`	Should be `true` or `false`, determines whether to include, for each annotation, the cosine similarity between the input document and the Wikipedia page corresponding to that annotation. Currently the cosine similarities are provided for informational purposes only and are not used in choosing the annotations. Thus, you should set this to `false` to conserve some CPU time if you don't need the cosines for your application.
`maxMentionEntropy`	Set this to a real number x to cause all highly ambiguous mentions to be ignored (i.e. they will contribute no candidate annotations into the process). The heuristic used is to ignore mentions where

	$H(link\ target\ \|\ anchor\ text\ =\ this\ mention)\ >\ x.$ (default value: -1, which disables this heuristic)
`maxTargetsPerMention`	Set this to an integer x to use only the most frequent x candidate annotations for each mention (default value: 20). Note that some mentions appear as the anchor text of links to many different pages in the Wikipedia, so disabling this heuristic (by setting $x = -1$) can increase the number of candidate annotations significantly and make the annotation process slower.
`minLinkFrequency`	If a link with a particular combination of anchor text and target occurs in very few Wikipedia pages (less than the value of `minLinkFrequency`), this link is completely ignored, and the target page is not considered as a candidate annotation for the phrase that matches the anchor text of this link. (default value: 1, which effectively disables this heuristic)
`pageRankSqThreshold`	Set this to a real number x to calculate a threshold for pruning the annotations on the basis of their pagerank score. The Wikifier will compute the sum of squares of all the annotations (e.g., S), sort the annotations by decreasing order of pagerank, and calculate a threshold such that keeping the annotations whose pagerank exceeds this threshold would bring the sum of their pagerank squares to $S\ \cdot\ x$.
`partsOfSpeech`	Should be `true` or `false`, determines whether to include information about parts of speech (nouns, verbs, adjectives, adverbs) and their corresponding WordNet synsets. This feature is only supported for English documents; the Brill tagger is used for POS tagging. (default value: `false`).
`verbs`	Like `partsOfSpeech`, but only reports verbs.
`nTopDfValuesToIgnore`	If a phrase consists entirely of very frequent words, it will be ignored and will not generate any candidate annotations. A word is considered frequent for this purpose if it is one of the $\mathrm{nTopDfValuesToIgnore}$ most frequent words (in terms of document frequency) in the Wikipedia of the corresponding language. (default value: 0, which disables this heuristic. If you want to use this heuristic, we recommend a value of 200 as a good starting point.)

5.1.6 Development plans for 2nd release (D3.4)

The next steps for JSI Wikifier development include the extension of available extra vocabularies.

The Wikifier service is going to be the basic service for annotation of news data while observing and tracking relation through time within Graph-Based Analytics services.

The next step for derived entity label generation is to upgrade the annotation process for Wikifier's business-related extra vocabularies extension.

5.2 Event Registry

Event Registry[36] is a system that can analyze news articles and identify in them mentioned world events. The system is able to identify groups of articles that describe the same event. It can identify groups of articles in different languages that describe the same event and represent them as a single event. From articles in each event, it can then extract an event's core information, such as event location, date, who is involved and what it is about. The extracted information is stored in a database. The user interface allows users to search for events using extensive search options, to visualize and aggregate the search results, and to inspect individual events and to identify related events.

5.2.1 Factsheet

Table 33 presents an Event Registry fact sheet.

Table 33: Event Registry – Fact sheet

Event Registry	
Functionality	Event Registry is a system that collects and analyses news articles from over 30.000 news publishers and identifies in them mentioned world events.
Usage in euBusinessGraph	In the euBusinessGraph, Event Registry tool is used as a source tool for providing news and events about companies, along with being a source for detection of business event types. In Graph-Based Analytics based on Event Registry data, the observation and tracking of relations between business entities will be performed.
Installation guide	http://eventregistry.org/
User guide	https://github.com/EventRegistry
API documentation	http://eventregistry.org/documentation/api
Software license	Event Registry can be accessed in free use and commercial use.
Contact person	gregor@eventregistry.org

5.2.2 Requirements (usage in euBusinessGraph)

The requirements for Event Registry service are provided in Table 34.

Table 34: Requirements – Event Registry

ID	Requirement	Description
Event Registry		
Req1	Data collection	The extensive coverage of different news sources for data collection (currently the system processes news from over 75.000 news sources).
Req2	Cross-linguality and Multilinguality	The articles in different languages are collected and processed by the service.
Req3	Preprocessing	The articles in the mentioned languages are then processed with a set of linguistic tools. A very important component for Event Registry is the named entity recognizer which detects the named entities mentioned in the articles and disambiguates them. Since events are associated with a date, we also try to identify in the text date mentions using a set of regular expressions for different languages.
Req4	Event construction	Groups of articles describing the same event are aggregated

[36] http://eventregistry.org/about

		in the event, information is extracted based on the event level.
Req5	Event categorization	Event categorization is an important requirement for business cases analytics services since it allows obtaining and analyzing the relevant subset of events for specific categories of interest (such as business and finance).
Req6	Web API	The service is accessible via API (with an access key). The users should register at the product website.
Req7	Web interface	Event Registry has a web interface that allows for displaying the search results, viewing trending concepts, clustering and tag clouds for events, displaying event and article information.

5.2.3 Architecture

Figure 23 presents Event Registry's pipeline.

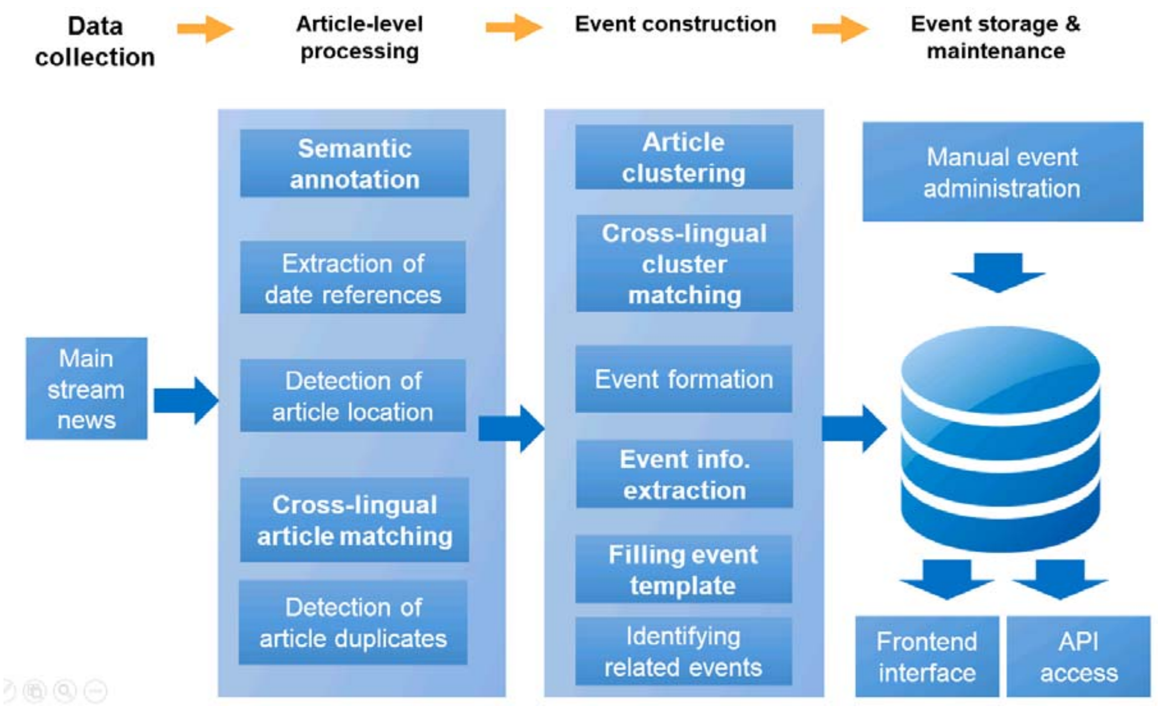

Data collection
Article-level processing
Event construction
Event storage & maintenance
Main stream news
Semantic annotation
Extraction of date references
Detection of article location
Cross-lingual article matching
Detection of article duplicates
Article clustering
Cross-lingual cluster matching
Event formation
Event info. extraction
Filling event template
Identifying related events
Manual event administration
Frontend interface
API access

Figure 23: Event Registry architecture

Event Registry architecture includes the following components:

- Mainstream news (the system is provided with news coming from thousands of mainstream sources);
- Article-level processing (including semantic annotation, extraction of date references and locations, addressing cross-lingual articles). The important part of this step is the detection of articles duplicates;
- The Event construction part includes articles clustering into events, addressing cross-lingual clusters, formation of events, categorization of event types, creation of events and identifying related events;
- Finally, the system is accessible via frontend interface and API access.

The current Event Registry set up is based on event categorization with DMOZ taxonomy[37]. However, within the euBusinessGraph project new mechanisms of event categorization are implemented, allowing more efficient analytics on top of news from a specific category (such as business news).

5.2.4 Functionality (to be used, enhanced or developed)

Table 35: Event Registry - Summary of features

Feature	Description	Existing[38]	New[39]
News article corpus	Real-time procurement and enrichment of news articles from news sources worldwide.	+	
Cross-lingual world event clustering	Clustering of similar news articles into Event instances.	+	
News visualizations	Article and Event instance visualizations.	+	
DMOZ categorization	DMOZ based text categorization model.	+	
Event-type categorization	Business taxonomy categorization model.		+

Data in Event Registry can be accessed through the web interface or directly through the available API. In order to access data in Event Registry, an API key is required. Accessing data through the API can be done by issuing HTTP GET requests with specific parameters. There are also Python and Node.js SDK libraries (https://github.com/EventRegistry/).

5.2.5 APIs

For Event Registry, we are continuing to develop the Web API documentation, which is written according to the OpenAPI 2.0 standard, and is available online:

- http://eventregistry.org/documentation/api

Below we describe the basics of Python SDK functionality.

`EventRegistry` is the main class to use in order to send a request to the Event Registry. `ReturnInfo` is a class that can be used to specify which properties should be returned for each returned data type (events, articles, concepts). Complete documentation of the Python SDK is available online.[40]

Table 36: Event Registry API parameters

Parameter	Description
`apiKey`	The API key for your account that should be used when making the requests.
`host`	The URL where the Event Registry is available.
`logging`	Should the executed web requests be logged in the logs folder in the package directory?
`minDelayBetweenRequests`	The minimum delay in seconds between two requests sent to Event Registry.
`repeatFailedRequestCount`	In case a request fails (for example, timeout), how many times should it be repeated before giving up.

[37] https://dmoz-odp.org/
[38] Existing software feature brought as background to be used or enhanced in the euBusinessGraph project.
[39] New or enhanced software feature developed as a new foreground during the euBusinessGraph project.
[40] https://github.com/EventRegistry/event-registry-python

`settingsFName`	If provided it should be a full path to `settings.json` file where `apiKey` and/or host can be loaded from. If `undefined`, we will look for the settings file in the `eventregistry` module folder.

Table 37: ReturnInfo Arguments

Parameter	Description
`ArticleInfoFlags`	Details about the articles to return.
`EventInfoFlags`	Details about the events to return.
`SourceInfoFlags`	Details about the news sources to return.
`CategoryInfoFlags`	Details about the categories to return.
`ConceptInfoFlags`	Details about the concepts to return.
`LocationInfoFlags`	Details about the locations to return.
`StoryInfoFlags`	Details about the stories to return.
`ConceptClassInfoFlags`	Details about the concept classes to return.
`ConceptFolderInfoFlags`	Details about the concept folders to return.

5.2.6 Development plans for 2nd release (D3.4)

The next steps of Event Registry development include improvements in event categorization for specific events. The Event Registry service is going to be used as a source of news and events data for observing and tracking business entities through time.

5.3 Graph-based analytics

The aim of graph-based analytics services is to cover:

- Relation observation between entities, such as companies and people;
- Relation tracking through time between entities, such as companies and people;
- Business event type categorization.

5.3.1 Factsheet

Tables below present the fact sheet for the services that compose Graph-Based Analytics – Event type categorization service and Relation Tracker.

Table 38: Graph-Based Analytics (Event type categorization) – Fact sheet

Graph-Based Analytics API	
Functionality	The aim of the Graph-Based Analytics API service is to cover: • Event type categorization; • Entity extraction.
Usage in euBusinessGraph	Event type categorization will be used by Event Registry when models become sufficiently reliable.
Installation guide	API and demo web app deployed at: http://gba.ijs.si/
User guide	Access to the relation extraction service is available through a SOAP-style API. In the future, user registration to obtain access will be required.
API documentation	https://github.com/JSI-EuBusinessGraph/jsi-graph-based-analytics-api
Software license	The software can be used in free and commercial modes.

Source code repository	Code is written in C, Python and Java, and stored in local SCM repositories.
Contact person	inna.koval@ijs.si

Table 39: Graph-Based Analytics (Relation Tracker) – Fact sheet

Relation Tracker	
Functionality	The aim of the Relation Tracker is to: • Model and extract relations with an unsupervised learning approach; • Observe relations through time; • Visualize the results in a web app; • Use cross-lingual and multilingual textual data; • Cluster entities.
Usage in euBusinessGraph	The relation extraction service will operate on Event Registry data and discover the relations between entities (companies, people), constructed from non-entities.
Installation guide	https://github.com/besher-massri/RelationTracker/wiki
User guide	Visualization interface deployed at: http://connection.ijs.si
Software license	The software is freely available.
Source code repository	https://github.com/besher-massri/RelationTrackerCode
Contact person	besher.massri@ijs.si

5.3.2 Requirements (usage in euBusinessGraph)

The requirements for Graph-based analytics service are provided in the table below.

Table 40: Requirements – Graph-based analytics

ID	Requirement	Description
Graph-based analytics		
Req1	Design Requirements/REST inspired	The service should provide a RESTful API.
Req2	Design Requirements/JSON	The service should provide results in JSON format.
Req3	Cross-linguality and multilinguality	The important aspects of the relation extraction service are related to the characteristics of the available datasets, such as multi-linguality and unstructured nature of the data.
Req4	Documentation Requirements	The API access should be clearly documented for external users. The documentation can have interactive character.
Req5	Process Requirements – API development	API should be developed and tested in phases.
Req6	Process Requirements – Usage of GitHub	GitHub repository can be used for managing the project.
Req7	Process Requirements – Documentation of changes	Changes and errors should be documented.

Req8	Relation extraction	The service should use machine learning and data mining methods for relation extraction.
Req9	Business cases analytics	The service should provide analytical solution on top of graph data.

5.3.3 Architecture

The majority of work on Graph-Based Analytics service has been currently concentrated at relation extraction tasks that are the basis for any analytical graph functionality.

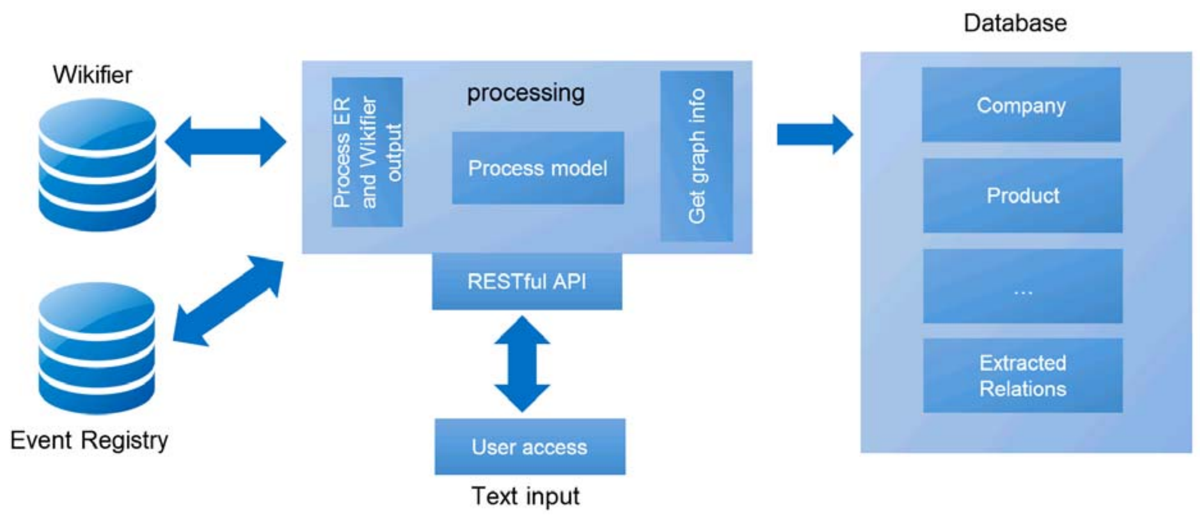


Figure 24: Graph-based analytics architecture

Figure 24 presents the architecture for the Graph-Based Analytics service.

The architecture of the prototype version of relation extraction services included the following components:

- Annotation of news with Wikifier (including synsets detection from Wordnet);
- Using events and articles from Event Registry for relation extraction;
- Database that stores information for companies and extracted relations;
- User access via RESTful API.

For entity extraction, we have introduced a **derived entity label generation** as an additional entity label preprocessing task. By enriching entity instances in Wikifier-ready datasets with derived entity labels, we can increase search recall of entities in unstructured data

During relation extraction development we have pivoted to acquiring and working with labeled data. This data was incorporated into our current approach to relation extraction modeling, outlined in Next steps.

For modeling the first version of relation extraction we have used word embeddings in a shallow linear neural network layer architecture.

5.3.4 Functionality (to be used, enhanced or developed)

Table 41 below presents the new features that should be developed for Graph-Based Analytics services.

Table 41: Graph-based analytics - Summary of features

Feature	Description	Existing[41]	New[42]
Event type categorization	Business event type classifier.		+
Relation Tracker	The series of desktop utilities includes: • Blacklist aggregation; • Event clustering; • Clusters processor; • Web-based visualization interface.		+

The developed methodology for event types categorization includes learning of specified relations (such as mergers and acquisitions, bankruptcies, etc.) based on a labeled dataset applying deep learning techniques.

The relation observing between entities and relation tracking through time is based on the Event Registry data, involving semantic annotation of cross-lingual and multi-lingual news and events with JSI Wikifier.

The methodologies and the applied techniques are provided in detail in the deliverable **D2.2 "Cross-lingual / Multi-lingual Data Management Approach for Structured and Unstructured Data"**.

5.3.5 APIs

For Graph-Based Analytics (Event type categorization), we are continuing to develop the API documentation, which is written according to the OpenAPI 2.0 standard, and is available online:

- https://jsi-eubusinessgraph.github.io/jsi-graph-based-analytics-api

The specifications are available at:

- JSON https://jsi-eubusinessgraph.github.io/jsi-graph-based-analytics-api/swagger.json
- YAML https://jsi-eubusinessgraph.github.io/jsi-graph-based-analytics-api/swagger.yaml

The Relation Tracker documentation is available at:

- https://documenter.getpostman.com/view/6188151/Rzn6uNR2?version=latest

5.3.6 Development plans for 2nd release (D3.4)

The next steps for the Graph-Based Analytics relation extraction service would be to improve the process of event type categorization, improve the deployed algorithms for relation tracking, and continue research into different machine learning techniques, statistical analysis, and pattern detection paradigms.

By using Event Registry's multilingual event clusters, we expect to be able to apply the relation extraction service in a multilingual scenario.

5.4 TWEC

The aim of these analytics services is to cover:

- Measure the similarity between words and entity identifiers, in particular words and entities relevant for company data analysis;
- Provide a method to train and compare word and entity embeddings based on different analytical dimensions such as time, location, and source, so as to support semantic comparative analysis of words and entities relevant for company-related data.

[41] Existing software feature brought as background to be used or enhanced in the euBusinessGraph project.
[42] New or enhanced software feature developed as a new foreground during the euBusinessGraph project.

5.4.1 Factsheet

Table 42 below presents the fact sheet for the TWEC tool.

Table 42: TWEC – Fact sheet

Graph-Based Analytics API	
Functionality	The objective of TWEC is to: • Measure the similarity between words and entity identifiers; • Provide a method to train and compare word and entity embeddings based on different analytical dimensions such as time, location, and source.
Usage in euBusinessGraph	TWEC is a tool that is used for semantic analysis of company-related words and entities. In particular, it is used for: • Measuring the similarity between words and entities related to company data and relevant to several business analyses; • Comparing words and entities related to company data as they appear in different media sources, countries and time periods.
Installation guide	The installation instruction will be provided in the associated repository.
User guide	Details on the use are present in https://github.com/valedica/twec
API documentation	
Software license	GNU Lesser General Public License, version 2.1
Source code repository	Code is written in Python and stored in https://github.com/valedica/twec
Contact person	matteo.palmonari@unimib.it

5.4.2 Requirements (usage in euBusinessGraph)

The requirements for TWEC are provided in the table below.

Table 43: Requirements – TWEC

ID	Requirement	Description	Type
TWEC			
Req1	Support training to build models over new corpora	The tool must provide algorithms to train embedding models that can be used by companies on their own corpora (e.g., news corpora).	F
Req2	Support for comparative semantic analysis along different analytical dimensions (time, location, media source)	The tool must provide a strategy to train embeddings in such a way that representations generated from different corpora or different sub-corpora of a larger corpus, can be compared.	F
Req3	Support for comparative semantic analysis also of entities in a company knowledge graph	The tool must support comparative semantic analysis of words and entities of in a knowledge graph.	F
Req4	Support operations for intra-corpus similarity evaluation	The tool must provide operations to evaluate the similarity between word/entity representations generated from a common corpus.	F
Req5	Support operations for inter-corpus similarity evaluation	The tool must provide operations to evaluate the similarity between word/entity representations generated from different corpora.	F

Req6	Process Requirements – Usage of GitHub	GitHub repository can be used for managing the project.	NF
Req7	Cases analytics	The tool must be tested on different data cases.	NF
			F: Functional Requirement **NF: Non-Functional Requirement**

5.4.3 Architecture

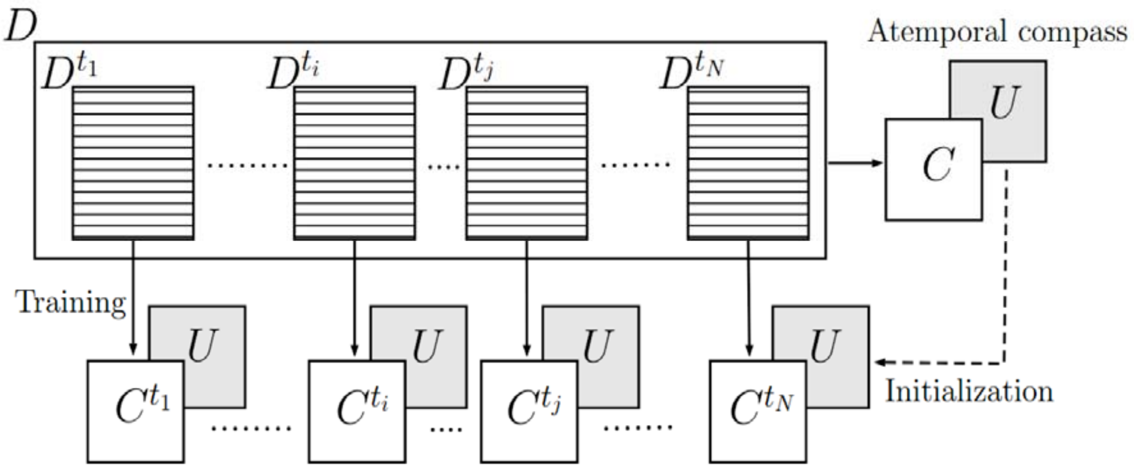


Figure 25: TWEC architecture

The conceptual architecture of the training process, which currently is under development, is sketched in Figure 25. Training is based on the following process:

- Different corpora of textual data are provided $(D^{t_1}, \dots, D^{t_N})$;
- A common representation is created from the aggregated data (C, U), that is, the corpus resulting from the aggregation of all the individual corpora. This common representation is referred to as Compass, as it is used for implicit alignment in the subsequent steps;
- For each corpus $(C^{t_1}, \dots, C^{t_N})$ a vector representation is obtained;
- Since vector representations are comparable, similarity can be obtained within the same dataset or across different corpora.

TWEC architecture is intended to be applied on top of vector-based representations of entities generated from text using Word2Vec. As a result, TWEC is an algorithm to support comparative analysis of words/entities along the time dimension and one another dimension, such as source, geography, etc.

5.4.4 Functionality (to be used, enhanced or developed)

Table 44 presents the new features that should be developed for TWEC.

Table 44: TWEC - Summary of features

Feature	Description	Existing[43]	New[44]
Training with compass-based alignment	An efficient method for training with different corpora with implicit representation alignment.		+
Intra-corpus similarity	Created embeddings must support the evaluation of similarity within each corpus.		+
Inter-corpus similarity	Embeddings coming from different sources must be comparable.		+

5.4.5 APIs

The tool is designed to be an extension of a word/entity embedding library so can be included in other software projects. The user guide documentation for the interface with other applications will be provided in the software repository (see D3.4).

5.4.6 Development plans for 2nd release (D3.4)

Next steps include completing the implementation of the algorithm, the release of the tool and the analysis of the representation performance on heterogeneous corpora.

[43] Existing software feature brought as background to be used or enhanced in the euBusinessGraph project.
[44] New or enhanced software feature developed as a new foreground during the euBusinessGraph project.

6 Conclusions

This document provides an overview of the euBusinessGraph Marketplace platform, introducing the customer segments of the platform, and outlining a set of requirements for the platform seen from the customer segments.

Furthermore, the document provides an updated architecture design of the marketplace platform and its software components. The architecture and software components will evolve during the course of the project. The requirements, architecture and software components outlined in this document will serve as the specification for the next implementation phase of the marketplace platform resulting in Deliverable D3.4 in month 24.

It should be noted that the technical results in the project should not be to solve the individual business cases, but rather focus on making the business graph and the marketplace services useful for a wide range of applications covering the business cases of the project.

A summary of the software components and their implementation status is given in Table 45. The table lists all the software components described in this report. For detailed descriptions on functionality that are implemented or planned see the respective sections in this document.

Table 45: Summary of software components and their implementation status

Software component	Functionality	Implementation status
Marketplace on the Cloud	Search and discovery company data: • Full-text search; • Faceted search; • Company profile view; • Basic analytics; • Articles search; • Events search.	In development.
GraphDB on the Cloud	A scalable semantic graph database.	Developed.
DataGraft	DataGraft provides a user interface that enables user and account management, user assets cataloging and dataset and database management.	**Developed.** DataGraft is used to integrate a streamlined data onboarding process using the software components Grafterizer and ASIA
Grafterizer 2.0	Grafterizer 2.0[45] is a web-based framework for tabular data cleaning and transformation and RDF mapping.	**In development.** Additional features and enhancements will be developed during the 2nd period of the project.
ASIA	ASIA is a semantic table enrichment tool integrated into Grafterizer 2.0, which provides Assisted Semantic Interpretation and Annotation of tables (CSV files).	**In development.** Additional features and enhancements will be developed during the 2nd period of the project.
ABSTAT	ABSTAT (Linked Data Summaries with ABstraction and STATistics) is a framework developed by UNIMIB to support users and machines in better understanding big and complex RDF datasets.	**In development.** Additional features and enhancements will be developed during the 2nd period of the project.
Wikifier	Wikifier is a Web service which takes a text document as input and annotates text chunks with links to respective Wikipedia concepts.	The service is ready. Subsystems are in development.

[45] https://datagraft.io/

Event Registry	Event Registry is a system that collects and analyses news articles from over 30.000 news publishers and identifies in them mentioned world events.	The service is ready. Subsystems are in development.
Graph-based analytics	The aim of the Graph-Based Analytics API service is to cover: • Event type categorization; • Entity extraction.	Prototype in development.
TWEC	Measure the similarity between words and entity identifiers.	Prototype in development.

Appendix A Business Model for euBusinessGraph Marketplace

A.1 Product Vision Board

Product envisioning describes what the future product will look like and do.

euBusinessGraph is positioned to be the key initiative to simplify the cross-border and cross-lingual collection, reconciliation, aggregation, and provisioning of company-related data from (authoritative and non-authoritative) public and private sector sources. The aim of euBusinessGraph is to enable cross-sectoral innovation based on company-related data.

The target customers of the data marketplace cover a wide range of business entities, which have processes, products or services that incorporate company-related data in their offerings and operations. This could potentially be customers that consume and provide data or service providers that perform various types of data service on the data available on the data marketplace.

In one sentence, the vision statement for euBusinessGraph is to deliver world-class company datasets and analytics across borders and languages (as shown in Figure 26).

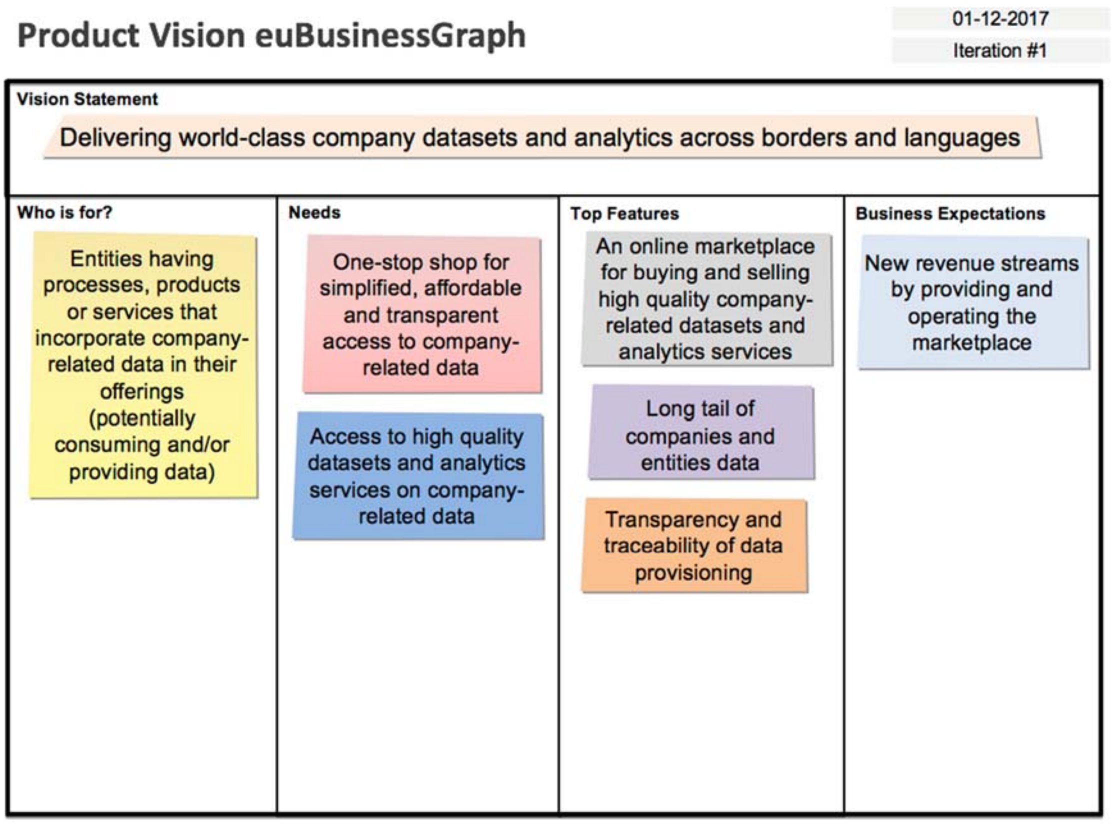
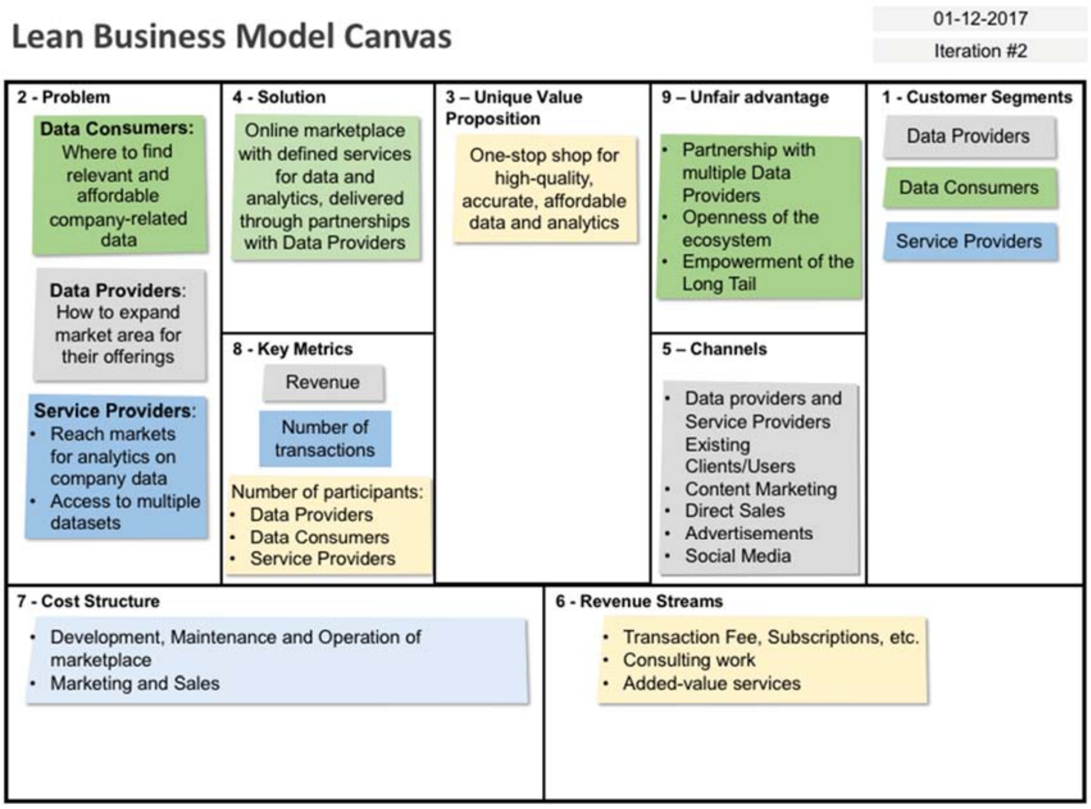

Product Vision euBusinessGraph
01-12-2017
Iteration #1
Vision Statement
Delivering world-class company datasets and analytics across borders and languages
Who is for?
Entities having processes, products or services that incorporate company-related data in their offerings (potentially consuming and/or providing data)
Needs
One-stop shop for simplified, affordable and transparent access to company-related data
Access to high quality datasets and analytics services on company-related data
Top Features
An online marketplace for buying and selling high quality company-related datasets and analytics services
Long tail of companies and entities data
Transparency and traceability of data provisioning
Business Expectations
New revenue streams by providing and operating the marketplace

Figure 26: euBusinessGraph Product Vision Canvas

A.2 Lean Business Model Canvas

The lean business model canvas is a chart with elements describing the euBusinessGraph Data Marketplace's value proposition, customers and finances, the problem the data marketplace is trying to solve, the solutions it presents, key metrics, and competitive advantage (Figure 27). This canvas is a fast and effective way to communicate the business model of the euBusinessGraph Data Marketplace in graphical form to internal and external stakeholders.



Lean Business Model Canvas

01-12-2017
Iteration #2

2 - Problem

Data Consumers: Where to find relevant and affordable company-related data

Data Providers: How to expand market area for their offerings

Service Providers:
- Reach markets for analytics on company data
- Access to multiple datasets

4 - Solution

Online marketplace with defined services for data and analytics, delivered through partnerships with Data Providers

8 - Key Metrics

Revenue

Number of transactions

Number of participants:
- Data Providers
- Data Consumers
- Service Providers

3 – Unique Value Proposition

One-stop shop for high-quality, accurate, affordable data and analytics

9 – Unfair advantage

- Partnership with multiple Data Providers
- Openness of the ecosystem
- Empowerment of the Long Tail

5 – Channels

- Data providers and Service Providers Existing Clients/Users
- Content Marketing
- Direct Sales
- Advertisements
- Social Media

1 - Customer Segments

Data Providers

Data Consumers

Service Providers

7 - Cost Structure

- Development, Maintenance and Operation of marketplace
- Marketing and Sales

6 - Revenue Streams

- Transaction Fee, Subscriptions, etc.
- Consulting work
- Added-value services

Figure 27: euBusinessGraph Lean Business Model Canvas

A.3 Value Proposition Canvas

The Value Proposition canvas describes how we are going to create values for the three user segments identified for the euBusinessGraph Data Marketplace. The three user segments are:

- Data Provider – this group of users has data which they would like to monetize, a data marketplace such as euBusinessGraph could provide a channel for them to achieve this. For example, an organization which has information on outstanding invoices in their Enterprise Resources Planning (ERP) solution and provide this information to credit scoring companies at a fee;
- Data Consumer – this group of users require certain types of business-related data in the daily operation of their organizations to help them make better business decisions. For example, an organization gets credit scoring report at a fee to evaluate the strength of the financial footing of a potential customer or vendor. Such a report can help them to analyze the risk they are being exposed to in every new business venture;
- Service Provider – this group of users can provide better visualization of the data available on the data marketplace. They can potentially provide various data services on the vast amount of data available on a data marketplace such as data cleaning and data quality assurance.

This type of canvas consists of a Customer Profile describing the user segments in the business model and a Value Proposition Map describing features of a specific value proposition for a product. We created three canvases (Figure 28, Figure 29 and Figure 30), one for each of the three user segments we identified above.

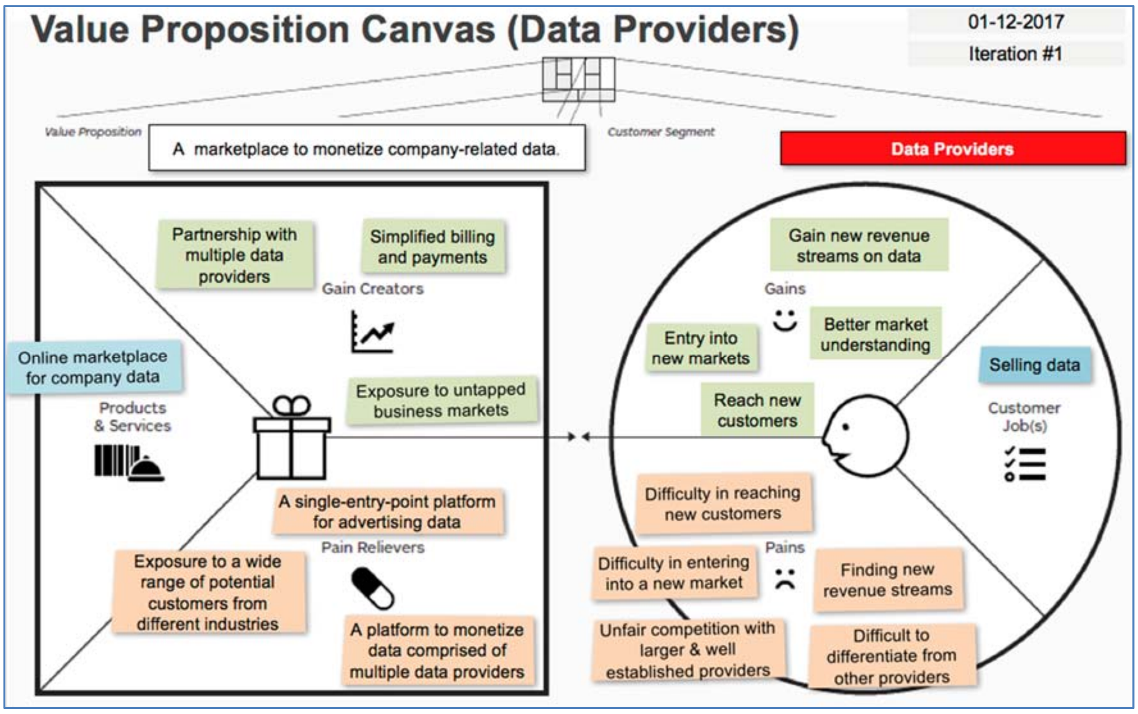

Value Proposition Canvas (Data Providers)
01-12-2017
Iteration #1
Value Proposition
A marketplace to monetize company-related data.
Customer Segment
Data Providers
Partnership with multiple data providers
Simplified billing and payments
Gain Creators
Online marketplace for company data
Products & Services
Exposure to untapped business markets
A single-entry-point platform for advertising data
Pain Relievers
Exposure to a wide range of potential customers from different industries
A platform to monetize data comprised of multiple data providers
Gain new revenue streams on data
Gains
Entry into new markets
Better market understanding
Selling data
Customer Job(s)
Reach new customers
Difficulty in reaching new customers
Difficulty in entering into a new market
Pains
Finding new revenue streams
Unfair competition with larger & well established providers
Difficult to differentiate from other providers

Figure 28: Value Proposition Canvas (Data Providers)

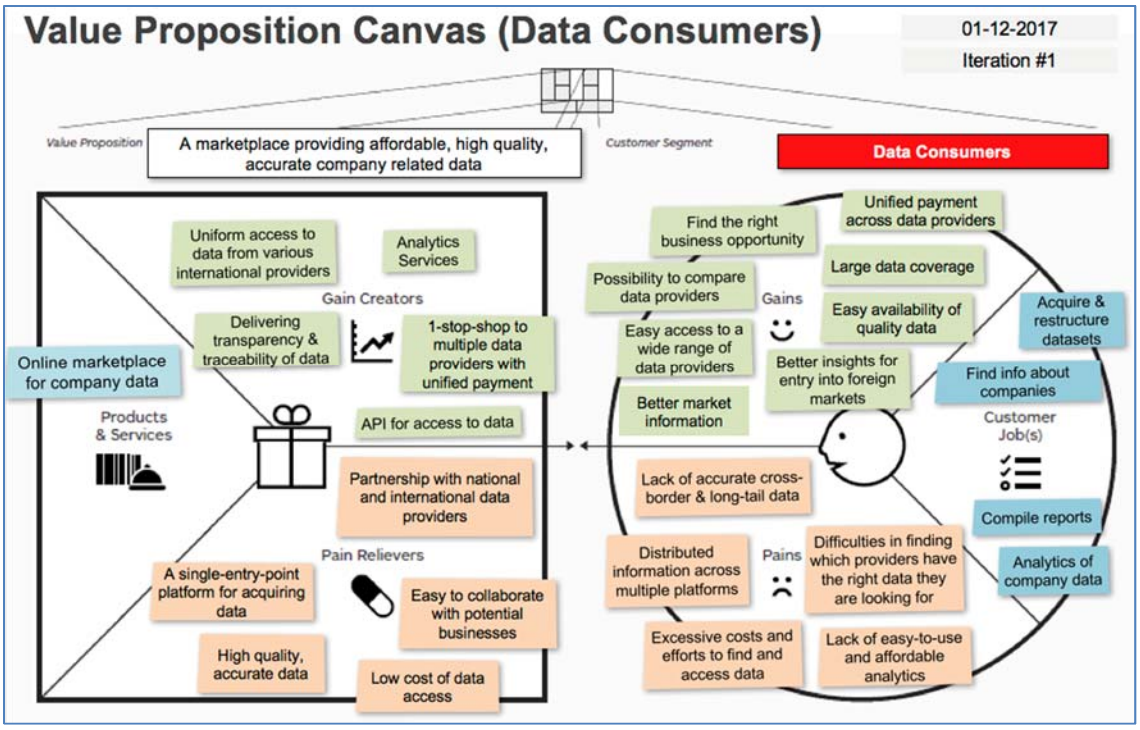

Value Proposition Canvas (Data Consumers)
01-12-2017
Iteration #1
Value Proposition
A marketplace providing affordable, high quality, accurate company related data
Customer Segment
Data Consumers
Uniform access to data from various international providers
Analytics Services
Gain Creators
Delivering transparency & traceability of data
1-stop-shop to multiple data providers with unified payment
Online marketplace for company data
Products & Services
API for access to data
Partnership with national and international data providers
Pain Relievers
A single-entry-point platform for acquiring data
Easy to collaborate with potential businesses
High quality, accurate data
Low cost of data access
Find the right business opportunity
Unified payment across data providers
Large data coverage
Possibility to compare data providers
Gains
Easy availability of quality data
Acquire & restructure datasets
Easy access to a wide range of data providers
Better insights for entry into foreign markets
Find info about companies
Better market information
Customer Job(s)
Lack of accurate cross-border & long-tail data
Compile reports
Distributed information across multiple platforms
Pains
Difficulties in finding which providers have the right data they are looking for
Analytics of company data
Excessive costs and efforts to find and access data
Lack of easy-to-use and affordable analytics

Figure 29: Value Proposition Canvas (Data Consumers)

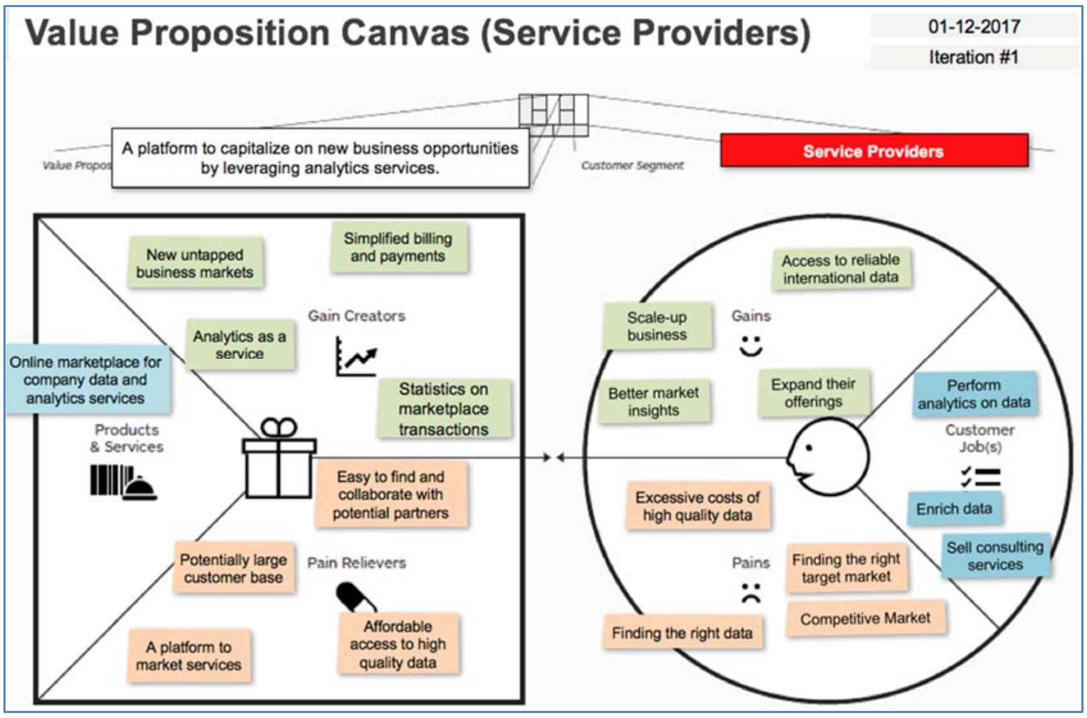

Value Proposition Canvas (Service Providers)
01-12-2017
Iteration #1
A platform to capitalize on new business opportunities by leveraging analytics services.
Value Propos
Customer Segment
Service Providers
New untapped business markets
Simplified billing and payments
Analytics as a service
Gain Creators
Online marketplace for company data and analytics services
Statistics on marketplace transactions
Products & Services
Easy to find and collaborate with potential partners
Potentially large customer base
Pain Relievers
A platform to market services
Affordable access to high quality data
Access to reliable international data
Scale-up business
Gains
Better market insights
Expand their offerings
Perform analytics on data
Customer Job(s)
Enrich data
Excessive costs of high quality data
Pains
Finding the right target market
Sell consulting services
Competitive Market
Finding the right data

Figure 30: Value Proposition Canvas (Service Providers)

A.4 Issues to be decided upon

A.4.1 Organization

It might be beneficial if the marketplace is established as a separate business entity, outside the consortium. The marketplace could either be established as a stockholder company or as a cooperative society. For transparency and ease of management, the former would be most appropriate.

Another possible model is to let a given partner set up and administrate the marketplace on behalf of the other consortium partners. This partner would administrate the governing of the marketplace and the sale of data on behalf of the other partners and would need to cover their costs out from the generated revenue. EVRY's Infotorg subsidiary is a good candidate for this model, as an established vendor of third-party business data.

A.4.2 Charter

The charter needs to have a clear statement on what the purpose of the marketplace is, the scope and the limitations as well as the statutes governing the organization of the marketplace. A governing board should be selected, drawing its members from the various organizations behind the marketplace.

A.4.3 Operating Model

Three different operating models have been suggested:
1. One partner in the consortium as the operator of the marketplace.
2. Two or more partners in the consortium as the operators of the marketplace.
3. An organization outside the consortium becomes the operator of the marketplace.

All three models have their limitations. The two main dimensions in this are the cost and profit-sharing structure, and the licensing and ownership of data. In order to ensure a fair and transparent structure, and to avoid the transfer of the ownership of data and responsibility from one partner to another, a separate governing body and separate organization are needed. This body must be owned by all partners with equal access and equal rights.

This governing body could then purchase needed services from either one or more of the partners in the consortium or from an external service provider. This would ensure a needed degree of flexibility to develop the marketplace autonomously outside the business interests of given partners and would also provide transparency with regards to the profits and costs.

A.4.4 Business Model

We suggest a subscription model for the marketplace, where customers choose a license fitting their needs, and is billed monthly, semi-annually or yearly for this license. The subscription cost should be competitive with vendors such as Dun & Bradstreet, etc.

The profit should be distributed to the shareholders either per share or per actual use of their data.

A.4.4.1 Licensing Models

The licensing models should be simple to understand and use. Some data are already public domain (such as the Norwegian Business Register) and should probably be offered for free, while other data is considered proprietary. Depending on the type of data and the ownership, several license tiers could be considered. Another issue is the number of monthly requests a user performs towards the service, which should be billed accordingly.

A.4.4.2 Suggested License tiers

- Free: Public domain data only
- Business: Public domain data as well as some value-added data
- Enterprise: Full access to everything

Other tiers should be considered.

A.4.4.3 Suggested Request tiers

- Free: Less than 5000 requests per month
- Small: 5000-10000 requests/month
- Medium:10000-50000 requests/month
- Large: 50000-250000 requests/month
- Enterprise: Per negotiation

Other tiers should be considered.

A.4.4.4 Considerations for income distribution

This licensing model is equivalent to what the competitors offer, with the exception that it is possible to use low volumes of the public data for no charge. This will make the service attractive for startups and non-commercial organizations that need the data but do not require large volumes.

Another issue is that the data should not be licensed per country. All data should be available through the same interface. This is to increase usability and to avoid a confusing and costly scheme where users are billed differently per country of use. The marketplace should instead keep track of the data usage for each subscription and divide the profit according to actual use.

For instance, Company Alpha has an enterprise data license, with a maximum number of requests set to 250k requests per month. 75 % of the made requests are for SpazioDati’s data, while the remaining 25 % is for OpenCorporates. The system should automatically calculate the revenue for the partner companies based on the usage share.

This is not necessarily complicated: several other services, such as for instance Spotify, perform similar calculations.

A.4.5 Ownership

The marketplace could be set up in many possible ways. It could either be a joint stock company owned by the consortium partners, allowing them full control of the revenue, or by a single entity either within or outside the consortium that will own and govern the marketplace, paying license fees to the consortium partners for the use of their data.

A.4.6 Service-level agreements

Business customers would most likely require some sort of service-level agreements in connection to their subscription. Such an agreement would detail both the required degree of uptime of the service, procedures regarding downtime, warning procedures, as well as requirements regarding the data quality. Each data provider should make a statement regarding their commitment and procedures on updates, error handling, data quality, etc.

A.4.7 Administration and governance

This section discusses some of the issues that need to be addressed in the establishment of a data marketplace governed by the consortium.

A.4.7.1 Membership

Membership concerns should be part of the statutes:

- **Rules for admitting new members:** Questions such as which organizations should be eligible for future partnership need to be addressed;
- **Rules when a member desires to withdraw:** The procedure for leaving the consortium should be established. It should contain points such as criteria for withdrawing, transition period and cost-sharing;
- **Exclusion rules:** When and how a member can be excluded from the consortium or organization, and the rights that the organization retains. This should only be applicable in certain cases where a partner fails to meet its obligations;
- **Minimum level of participation:** A minimum level of participation should be agreed upon. This includes items like the level of data sharing if applicable, the responsibility for costs and deficits and participation of the organizations;
- **Decision-making process:** In the following steps we will analyze questions related to:
  - How decisions will be reached?
  - Which issues will need consensus and which issues will need a simple or absolute majority?

A.4.7.2 Budgets

The rules for budgeting and reporting must be well defined and established.

- **Operations:** An operating budget should be decided upon.
- **Marketing:** A budget for marketing, if necessary, should be set aside.
- **Infrastructure and investments:** Depending on the operating model, a budget for infrastructure and other investments might be necessary.