Platform high-level architecture and data flows

The Data4Impact Knowledge Base Platform (D4IP)

Big Data approaches for improved monitoring of research and innovation performance and assessment of the societal impact in the Health, Demographic Change and Wellbeing Societal Challenge

Project acronym: Data4Impact

Grant Agreement no: 770531

Deliverable: D3.1

Deliverable information	
Deliverable number and name	D3.1 Platform high-level architecture and data flows
Due date	Month 6
Delivery	Month 6
Work Package	WP3
Lead Partner for deliverable	CNR
Author	Paolo Manghi (CNR), Claudio Atzori (CNR)
Reviewers	Omiros Metaxas (ARC), Johan Eklund (HB)
Approved by	All partners
Dissemination level	Confidential
Version	1.0

Document revision history

Issue Date	Version	Comments
12/03/2018	0.1	First draft.
20/04/2018	0.2	Updated by Paolo Manghi and Claudio Atzori
26/04/2018	0.3	Revised
27/04/2018	1.0	Revised and corrected final version

Disclaimer

This document contains description of the **Data4Impact** project findings, work and products. Certain parts of it might be under partner Intellectual Property Right (IPR) rules so, prior to using its content please contact the consortium coordinator for approval.

In case you believe that this document harms in any way IPR held by you as a person or as a representative of an entity, please do notify us immediately.

The authors of this document have taken any available measure in order for its content to be accurate, consistent and lawful. However, neither the project consortium as a whole nor the individual partners that implicitly or explicitly participated in the creation and publication of this document hold any sort of responsibility that might occur as a result of using its content.

This publication has been produced with the assistance of the European Union. The content of this publication is the sole responsibility of Data4Impact consortium and can in no way be taken to reflect the views of the European Union.

The European Union is established in accordance with the Treaty on European Union (Maastricht). There are currently 28 Member States of the Union. It is based on the European Communities and the member states cooperation in the fields of Common Foreign and Security Policy and Justice and Home Affairs. The five main institutions of the European Union are the European Parliament, the Council of Ministers, the European Commission, the Court of Justice and the Court of Auditors. http://europa.eu.int/)

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 770531.

Abbreviations

DMP – Data Management Plan
DoW – Description of Work
EC – European Commission
PPMI – Public Policy and Management Institute
UoB – University of Borås
WP – Work Package
D4IP - Data4Impact Knowledge Base Platform

Table of Contents

Summary

This document describes the Data4Impact knowledge base platform (D4IP) high level architecture and its data flows, aimed to support the automatic computation of indicators on the impact of science, technology and innovation policies.

CNR prepared this document as the partner leading WP3 with inputs from other consortium partners and with a review from ARC and HB.

1. Introduction

In this document we present the the first iteration of the D4I Knowledge Base Platform (D4IP) design, a system supporting the automatic computation of indicators on the impact of science, technology and innovation policies. As a consequence, we also present the data flows implemented within the platform and involved in the computation of such indicators (which are extensively described in D2.1).

The design process of the D4IP collects requirement from WP2, WP4, WP5 and WP6 in order to identify the best deployment configuration and settings so as to collect outputs of data analysis and aggregation in support of indicator measurement and in support of other data analysis workflows (intermediate analysis results). The D4IP will operate as the common deposition and access source for input and output data of data analysis workflows and the source of aggregated data interrogation and calculation of indicators.
Three main kind of data sources will be integrated to serve as "raw" and intermediate data for data analysis workflows:

- Funder and grant data: programmes, projects; national and EU level;
- Scientific output: article metadata and full texts, patents metadata and full texts;
- Impact data: company data; monitoring data on finalised EU projects; online/social media blog and health-related forum data; and policy documents / data.

Once calculated, the data resulting from the analysis workflows will be made available for programmatic access to end user interfaces to be developed in the context of WP6.

2. D4IP High level architecture

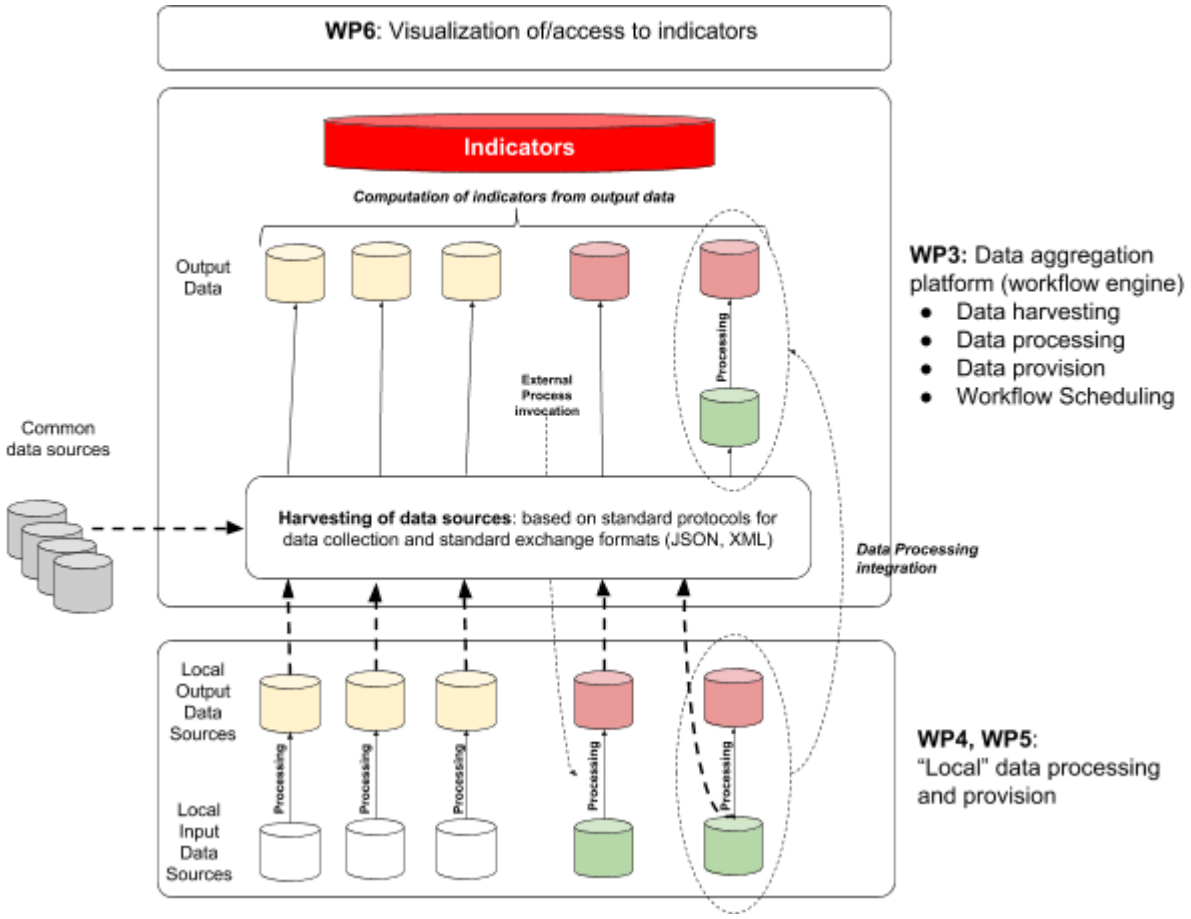


Figure 1 D4IP general architecture

2.1. D4IP: integration patterns

The diagram in Figure 1 describes the D4IP system, which consists of a combination of local infrastructures (i.e. systems at different partners' sites) and a data aggregation platform. The aggregation platform acts as both (i) the central place where all workflow outputs are collected and processed to deliver indicators, and (ii) the central place where *common data sources* or workflow *output data* can be shared and accessed by partners. The generic local infrastructure builds a workflow that processes *local input data sources*, processes them, and generates *local output data sources* (see Figure 1). The data aggregation platform can aggregate local infrastructure content according to the following patterns.

Data source integration: harvesting common data sources and output data sources of a local processing workflow.

- Common data sources: OpenAIRE projects and PubMed. In this case the data source is directly harvested by D4IP and offered to the task leader's local infrastructure to download or to D4IP workflows to reuse.

- Output data sources: the data is collected and processed by local infrastructures in the context of WP4 and WP5 workflows. The solution has low technological integration cost, but does not easily support "continuous generation of indicators". Agreements on the frequency of harvesting must be taken between WP output data production and D4IP harvesting workflows.

Workflow integration: integration of business logic into D4IP, requires technology integration into the platform workflow engine. The platform will then harvest the primary data and execute the process locally. This solution supports "continuous generation of indicators" but at the cost of integrating heterogeneous technologies.

External workflow integration: external workflows can be executed by the D4IP. This solution supports "continuous generation of indicators" mitigating the cost if integrating heterogeneous technologies.

The data aggregation platforms also offers the possibility for local infrastructure workflows to be migrated and executed centrally. Given the considerable heterogeneity of technologies and data models/formats involved in the project, the task to harmonise and unify all models and schemas will be significant and most likely absorb all partners' effort, hence workflow integration will not be the main focus. This possibility may however be considered for future developments so as to better enforce project sustainability.

For this reason, in order support the automatic computation of the first set of indicators identified in D2.1 the data aggregation platform will collect the output data sources from local infrastructures, where workflows will be executed.

3. D4IP Common data sources

In the following section we describe the data sources that can be used to serve as input in different data analysis workflows. Such data sources are aggregated by the data aggregation platform and made available as input to the local infrastructure workflows. For each source the relative protocol, data model, schema, exchange format is provided.

EC Projects

The set of FP7 and H2020 projects will be retrieved from the CORDIS api to collect a LOD representation and from the OpenAIRE api[1] to collect an XML representation.

Publications from PubMed and OpenAIRE (ARC)

Scientific outputs (publications) and related metadata is collected from OpenAIRE and Open Access PubMed. From OpenAIRE the publications of interest are all those that contain funding related information, and the field. For all publications the we collect the following:

- Title
- Abstract
- Full Text (whenever available)
- Venue
- Publication Date

[1] http://api.openaire.eu

- Internal Citations (for OpenAIRE only publications)
- MeSH terms (whenever available for PubMed Publication)
- Keywords
- Funding Info (funder or grant for OpenAIRE publications)

To collect above mentioned data we utilize related APIs and services from OpenAIRE and PubMed. The goal is to analyse such data based on the Topic Modelling workflow in WP4 (together with patent related data and project related data from CORDIS). All data are accessible through specific REST APIs and SQL drivers (e.g., JDBC driver for Postgres) and the detailed data model is presented in following section describing the WP4 data flow. The data will be then updated every three months.

4. Data processing workflows in WP4

4.1 Topic modelling

The work carried out in WP4 consider the analysis of data from various sources:

- Scientific outputs and related metadata (OpenAIRE, Open Access PubMed);
- Project reports, abstracts, evaluation summary and other information from CORDIS;
- Patents (IPR) outputs and related metadata.

In addition, the plan is to analyse additional data as collected in WP5 including project related content from web sites, company data, social data and content related to EU directives and guidelines.

The overall WP4 objectives can be summarized as follow:

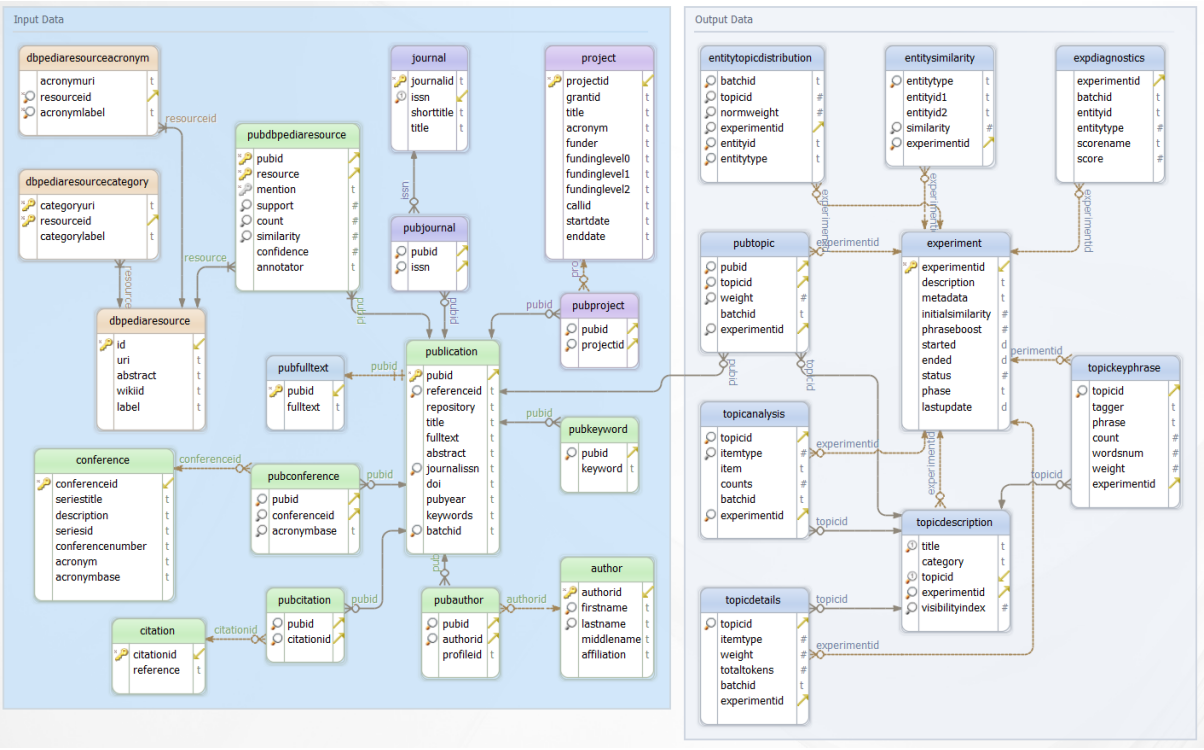
- identify active areas of research: discover hidden themes (topics) understand what is actually produced: calculate topic distributions per document / entity (e.g., project, author, call);
- analyse active research areas on several dimensions (e.g., compare geographic regions, funders, etc.);
- discover clusters and communities, assess research collaboration: topic based similarity analysis;
- identify emerging research areas: topic based trend analysis;
- assess coverage, identify gaps or new challenges: compare funded research;
- assess the impact of research in the society using new indicators as described in D2.1 section 4.3.

Following a specific data integration process, all the collected data is harmonized, transformed and stored to a PostgreSQL RDBMS. Such database contains both input (publication, patent, project related data) and output (results from the topic modelling flow) data. Result from the topic modelling flow include:

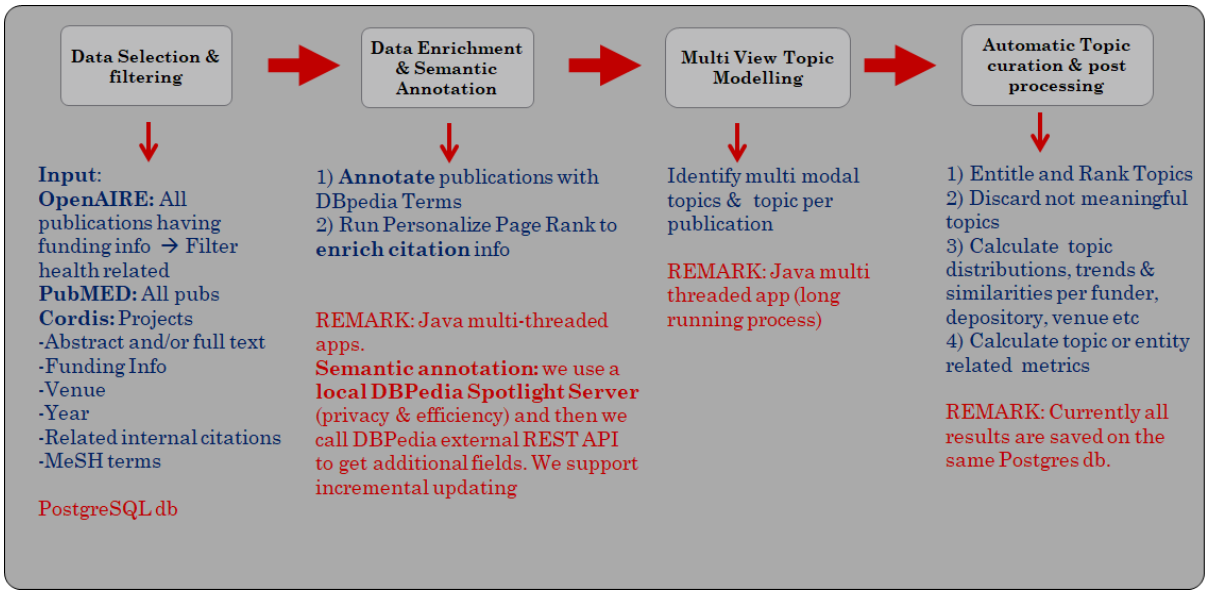
- Identified Topics: Words, Wikipedia terms, Keywords, Mesh terms, phrases per topic;
- Topics Per Document (publication, report, patent etc.);
- Topics Per other entities of interest (e.g., projects/grants, funder, etc.);
- Similarity among topics;

- Similarity among different types of entities (e.g., grants, E.C. calls, funders, documents etc.);
- Topic trends;
- Several topic related metrics: Topic weight, topic exclusivity on one specific dimension (e.g., grants), topic trend index, topic coherence, etc.

The schema of such Topic Modelling DB is shown below:



The whole WP4 topic modelling flow (shown in the diagram below) is running in a local infrastructure, all results are stored in a PostgreSQL database and are available through a well-defined REST API and SQL drivers.



Local input data sources

Common data sources: EC Projects, Publications from PubMed and OpenAIRE

Patents in PATSTAT and FP7 patents dataset (ARC / PPMI)
PATSTAT contains bibliographical data relating to more than 100 million patent documents from leading industrialised and developing countries. This is extracted from the European Patent Office (EPO)'s databases and is either provided as raw data or can be consulted online. It also includes the legal status data from more than 40 patent authorities contained in the EPO worldwide legal status database (INPADOC).

PATSTAT is produced by EPO on behalf of the OECD Taskforce on Patent Statistics. Much of the raw data is extracted from the EPO's master bibliographic database DOCDB, also known as the EPO Patent Information Resource. ATHENA RC have acquired the 'PATSTAT Biblio - single edition spring 2017' raw data. If we publish analyses based on this statistical database, we must cite the source of the data including the name of the current version, i.e., 'EPO Worldwide Patent Statistical Database - 2017 Spring Edition'. The copyright to this database as distributed by the EPO remains with the EPO. "PATSTAT" is a registered trademark.

A complete description of the PATSTAT data catalogue for our version of the database can be found online[2]. The logical model diagram (schema) with its 27 tables is shown in Figure 2. PATSTAT contains the metadata of the patents and only Abstracts of patents (in table TLS203_APPLN_ABSTR). Since in Data4Impact we may require the full text of patents for topic modelling (Abstracts might be insufficient), we have also ordered the EP full-text raw data collection which includes the full text in machine-readable format (XML) of all patent applications and granted patent specifications published by the EPO since it was set up in 1978.

The PATSTAT data are originally provided as a set of multiple CSV files (corresponding to each table in the schema). These have been imported into an SQLite database in one of ATHENA RC's servers for internal processing. The 'EPO Worldwide Patent Statistical Database - 2017 Spring Edition' that we have purchased is a single edition dataset and does not get updated. Updated editions would require purchasing EPO's annual subscription.

5. Data processing workflows in WP5

5.1 Clinical guideline citation context analysis

Description of content: Citation context analysis of cited references to project publications in clinical guidelines.

This is intended to supplement the strict bibliometric analysis with a contextual 'quali-quantitative' analysis of how citing is done to better evaluate the role of the cited document in the clinical guideline.

How it is produced and why: matched citations found in the clinical guidelines will be probed for textual content around the reference in-text and various text based methods will be used evaluate the role of the reference. This will be done by analysing contextual fragments related to cited references, "references-in-context" (RIC) to identify the topic distributions of the textual neighbourhood of cited references in the clinical guidelines.

Data model: From the text based analysis of the cited references within the clinical guidelines analyses of the following kind will be done:

2 http://documents.epo.org/projects/babylon/eponot.nsf/0/D0CE63CD32DC9A9EC12581C2003661F7/$FILE/data_catalog_register_v3.05_en.pdf

- Reference function (e.g. sentiment). [textual classes, text];
- Correlation between cited references and named entities. [interval values, float];
- Triangulation between cited references, named entities and reference functions. [interval values, float].

How it can be accessed: The information will be stored in a relational database that could be queried and or performance data could be exported in suitable metadata formats according to needs. Upon activity, updates can be produced continuously. Set dates for data delivery can be set according to needs.

Topic modelling of clinical guidelines data
Description of content: Text data from matched clinical guidelines, linked to research topics.

Latent links between clinical guidelines research output produced by FP7 / H2020 projects and other programmes will be identified from text data from matched clinical guidelines, linked to research topics found in Task 4.

Local input data sources

Common data sources: EC Projects.

Clinical guidelines data (UoB)
Data from the clinical guidelines (Task 5.3) will be collected and processed externally from the D4IP by UoB with the help from external contributor Minso Solutions. The Clinical Impact database will be queried for matches in clinical guidelines based on Document identifiers (DOI, CrossRef ID, PubMed ID, Scopus ID or Web of Science ID) based on project publications.

Resulting Clinical guideline citation impact metrics will be treated by normal bibliometric conventions such as field normalization and author fractionalization and made possible for aggregation at different levels.

Research impact data in the form of aggregated bibliographical data on researchers institutional, country and research affiliations will be used to develop impact measures and map how EU financed research and the national funded research has come to use in the actual professional health sector setting. Indicators will be developed to measure performance at various levels. Clinical guideline citation impact at:

- project level [interval values, float];
- programme level [interval values, float];
- intra-national (affiliation) [interval values, float];
- national level [interval values, float];
- Collaboration at the intra-national and national levels [interval values, float].

Impact data will be kept in a relational database that could be queried by SQL calls. Data could also be shared as TSV or any other text based metadata standard. Exact format has not been established at this point but it would be good to establish a standard common for the project.

For clinical guideline data the following would be needed (not conclusive):

- Document identifiers (DOI, CrossRef ID, PubMed ID, Scopus ID or Web of Science ID);
- Project identifier: EU/National projects, Call, Project ID, Project title;
- Clinical guideline set (country), providing organization, guideline group within provider;
- Impact measures according to schema above.

Frequency of update: upon activity, continuously. Set dates for data delivery can be set according to needs.

5.2 Company Data Indicators (PPMI)

Company data indicators are produced by processing CORDIS data sources, crawling company web sites, collecting text data wherein and identifying innovation mentions in the textual data. Innovation mentions are then classified by innovation stage (pre-market / market) and type (intangible products/ tangible products). Finally, unique innovations are identified from the innovation mentions and counted.

This process follows a simple linear workflow:

1. Relevant companies are identified from the CORDIS data by selecting companies which participated in the projects in the domain of Health (project list compiled in Task 5.2);
2. Company web-sites are crawled and text data is collected. Data is stored in PPMI until it is fully analysed and then deleted;
3. From the selected texts, those mentioning innovations are identified and selected;
4. Texts with innovation mentions are classified by innovation stage and type;
5. Mentions of innovation commercialization are also identified;
6. From the classified innovations mentions, unique innovations are identified and counted;
7. Final data with company innovation counts is handed over to D4IP.

More specifically, the resulting indicators will contribute to measuring both Economic Output and Economic Impact, as specified in D2.1 and illustrated in Table below:

Indicator Type	**Input data source**	**Indicator**
Economic Output	Company websites	Number of companies introduced an innovation (regardless of stage and type) after received a funding
Economic Impact	Company websites	Number/share of companies introduced an innovation to the market after received a funding
		The number/share of companies that commercialized the innovations they produced.

Company data indicators will be generated by the local services of PPMI and be made available to the D4IP every 200 days. The frequency of the update is 200 days because of two reasons:

1. The company websites are not updated on the daily basis;
2. It takes several weeks for data collection and processing.

Hence, more frequent data updates would be technically demanding and not very useful.

The data will follow the schema outlined in the table below. The table will contain anonymized company-level data. Each company will be given a unique ID, which will be the same as company's PIC number from CORDIS (thus allowing to integrate company level data with

project and call data). The table will feature a count of unique innovations produced by a company by stage and type as well as a count of commercialization instances.

Column	Data type
Company number (PIC number from Cordis)	Integer
Company Innovation Score	Integer
Tangible Innovation count: Pre-market Stage	Integer
Tangible Innovation count: Market Stage	Integer
Intangible Innovation count: Pre-market Stage	Integer
Intangible Innovation count: Market Stage	Integer
Count of commercialization/valorisation activities	Integer

The data will be made available using the JSON format, using PIC numbers as keys (see example below).

```
{999999999: {‘Company Innovation Score’: 10, ‘Tangibles pre market’: 3, ‘Tangibles market’: 2, ‘Intangibles pre market’: 0, ‘Intangibles market’: 0, ‘Commercialization’: 1},
999999998: {‘Company Innovation Score’: 12, ‘Tangibles pre market’: 0, ‘Tangibles market’: 0, ‘Intangibles pre market’: 0, ‘Intangibles market’: 4, ‘Commercialization’: 2}}
```

The final data outlined above will be uploaded to the D4IP from PPMI every 200 days.

Local input data sources

CORDIS Company data: participants information collected from CORDIS relative to companies.

Data collected from Company websites (PPMI)
Data from the company websites (Task 5.1) will be collected and processed externally from the D4IP by PPMI. The data will be collected by crawling the websites of enterprises which participated in relevant projects and collecting all text data wherein, except for *a priori* specified decisions of the websites (such as ‘/contacts’), which might contain sensitive personal data. The textual data collected from company websites will constitute the ‘Initial Data Table’.

These data will be processed using appropriate natural language processing techniques and specific URLs in a company domain, containing mentions of innovations will be isolated. These then will be classified by innovation stage (pre-market/market) and type (tangible product/ intangible product). The isolated URLs with innovation mentions and designations of innovation stage and type will constitute the ‘Interim Data Table’.

Then these innovation mentions will be filtered to identify unique innovations. The unique innovation counts by stage and type will constitute the Final data, which will be handed from PPMI to D4IP.

Column	**Data type**
Company number (PIC number from Cordis)	Integer
Company Innovation Score	Integer
Tangible Innovation count: Pre-market Stage	Integer
Tangible Innovation count: Market Stage	Integer
Intangible Innovation count: Pre-market Stage	Integer
Intangible Innovation count: Market Stage	Integer
Count of commercialization/valorisation activities	Integer

Every 200 days, PPMI will share the data update with the D4IP in the form of JSON file. The sample structure of the JSON file is presented below. It will use PIC numbers as keys (see example below).

```
{99999: {‘Company Innovation Score’: 10, ‘Tangibles pre market’: 3, ‘Tangibles market’: 2, ‘Intangibles pre market’: 0, ‘Intangibles market’: 0, ‘Commercialization’: 1},

999998: {‘Company Innovation Score’: 12, ‘Tangibles pre market’: 0, ‘Tangibles market’: 0, ‘Intangibles pre market’: 0, ‘Intangibles market’: 4, ‘Commercialization’: 2}}
```

The entire schema of PPMIs project database is presented below.

Initial Data	**Interim Data**	**Final data (uploaded from PPMI to D4IP)**
Company_id (int)	Company_id (int)	Company number (PIC number from Cordis) (int)
Site_url (varchar)	Site_url (varchar)	Company Innovation Score (int)
Site_text (text)	Innovation_mention (bool)	Tangible Innovation count: Pre-market Stage (int)
Text_language (varchar)	Tangible_pre_market (bool)	Tangible Innovation count: Market Stage (int)

Extraction_date (date-time)	Tangible_market (bool)	Intangible Innovation count: Pre-market Stage (int)
	Intangible_pre_market (bool)	Intangible Innovation count: Market Stage
	Intangible_market (bool)	Count of commercialization/valorisation activities (int)
	Commercialization_evidence (bool)	

5.3 EC Projects portfolios (ARC)

The data that will be gathered in this task (task 5.2) are the portfolios of the finalised EU projects in the field of Health, Demographic Change and Wellbeing. A so called **Project Portfolio** will be constructed, composed of the following data:

- The actual Call e-booklet;
- Project Description along with metadata concerning participants, funding etc.);
- Project Documentation (Final reports, brief results etc.);
- Project Publications;
- Project Patents;
- Policy Documents.

Project Portfolios are generated from various sources: CORDIS, Pubmed publications, Patents. The Data4Impact Text Mining workflows consume these data and process them in order to extract useful metadata and insights. Both metadata and insights are exposed via Restful APIs. The raw data are kept and curated in the internal archive.

APIs

All APIs expect a post request in a json format including all the parameters needed.

API #1: Data Stats API

The first API returns all the necessary data in aggregated form. The API call should be XXXX and the response is a json formatted text as shown in the example below. The field **total_projects** provides the total number of projects stored in the database. The other fields provide aggregate stats on Calls, diseases, domain and subject Fields respectively.

Each key can be used in the API #02 to refine the search. The value of each key indicates the number of Projects this key was found in (project portfolios).

Request example:

Response example:

```
{
	'calls': [
		{
			'FP7-HEALTH-2007-B': 155
		},
		{
			'FP7-HEALTH-2007-A': 122
		},
		...
	],
	'diseases': [
		{
			'cancer': 282
		},
		{
			'diabetes': 168
		},
		...
	],
	'domain': [{ 'FP7-HEALTH': 874}],
	'subjects': [
		{
			'Medicine and Health': 420
		},
		{
			'Life Sciences': 343
		},
		...
	],
	'projects_with_insights': 648,
	'total_projects': 874
}
```

API #2: Project data in Batches API

The API request data must be json formatted (an example is given below). The user must provide a value to the fields "from" and "size". The API will skip as many instances as indicated by the field "from" and return the following instances of the database. A number of instances equal to the "size" parameter will be returned. The user can further narrow her/his search space, using the optional fields "calls", "diseases", "domain", "subjects". When these fields are used, only matching projects will be returned. API #01 and #02 can be used jointly to fill in these parameters.

Finally, one can define what type of data should be returned using the field "return". There are three possible values: 'project_metadata', 'paper_metadata' and 'annotations'.

"Project_metadata" enforces project metadata to be returned: (namely the fields: 'acronym', 'funded_under', 'project_id', 'call', 'title', 'cordis_link', 'date_from', 'date_to', 'NumberOfPublications', 'eu_contribution', 'total_cost', 'subjects', 'topics', 'participants', 'coordinators'). With "Paper_metadata", the metadata of the Project Publications are returned (namely the fields: 'paper_chemicals', 'paper_keywords', 'paper_mesh_terms').

Finally, "annotations" enforces automatic annotations and insights to be returned: (namely the fields: 'organizations', 'insights', 'diseases', 'terms').

In the following example the API will return the first 20 instances of the database that match all the conditions. In order to get the next 20 instances, the user has to put the value 20 in the field "from" and the value 20 in the field "size".

Request example:

```
{
 'from': 0,
 'size': 20,
 'calls'       : ['FP7-HEALTH-2007-B', ... ],
 'diseases'    : ['cancer', ... ],
 'domain'      : ['FP7-HEALTH', ... ],
 'subjects'    : ['Life Sciences', ... ],
 'return'      : [ 'project_metadata', 'paper_metadata', 'annotations' ]
}
```

API #02 results:

The API response data is a well-formatted json object (an example is presented below).

In the field "results" one can find a list of data in json format. Each json object in the "results" field refers to a Project found in the database. Finally, the fields: "total", "from" and "size" emphasize the values of the parameters used in the json request.

Response example:

```
{
 'results' : [
 {
  'acronym'                  : 'PREPARE',
  'funded_under'             : 'FP7-HEALTH',
  'project_id'               : '602525',
  'call'                     : 'FP7-HEALTH-2007-B',
  'title'                    : 'Platform foR European Preparedness Against (Re-)emerging
Epidemics',
  'cordis_link'              : 'http://cordis.europa.eu/project/rcn/110174_en.html',
  'date_from'                : '2014-02-01',
  'date_to'                  : '2019-01-31',
  'NumberOfPublications'     : 5,
  'eu_contribution'          : 23992375.0,
  'total_cost'               : 31130188.06,
  'subjects'                 : [ 'Life Sciences', ... ],
  'topics'                   : [ 'HEALTH.2013.2.3.3-1 - Clinical management of patients in
severe epidemics', ... ],
  'coordinators': [
   {
    'activity_type'    : 'Higher or Secondary Education Establishments',
    'contribution'     : 4119132.0,
    'country'          : 'Belgium',
    'name'             : 'UNIVERSITEIT ANTWERPEN'
   },
  ...
  ],
  'participants' : [
   {
    'activity_type'  : 'Higher or Secondary Education Establishments',
    'contribution'   : 2074696.0,
    'country'        : 'Netherlands',
    'name'           : 'Academisch Medisch Centrum bij de Universiteit van Amsterdam'
   },
  ...
  ],
  'paper_chemicals'   : [ 'Immunoglobulin E', 'purothionin', 'Triclosan', ... ],
  'paper_keywords'    : [ 'maternal genetic effects', 'IL-4', 'IL-6', 'rhinitis'.. ],
  'paper_mesh_terms'  : [
   {
    'id'       :  'E05.393.673',
    'mesh_term':  'Pedigree'
   },
   ...
```

```
  ],
  'organizations': [
   {
    'mentions' : 2,
    'name'     :  'EARL'
   },
   ...
  ],
  'insights'  : [
   {
    'text': 'Standardized approaches for TIL evaluation',
    'type': 'Approach'
   },
   ...
  ],
  'diseases'      : [
   {
    'mentions' : 15,
    'name'     : 'cancer'
   },
  ],
  'terms'         : [ 'infectious', 'health', 'influenza', ... ],
  },
  ...
 ],
 'from'      : 0,
 'size'      : 20,
 'total'     : 983
}
```

Local input data sources

Common data sources: Pubmed publications, EC projects.
Patents in PATSTAT and FP7 patents (see 4.1)

5.4 Project mentions in social media / blog / health-related forum data

For each EU-funded project a dataset will be produced that contains the mentions per medium regarding the project: news articles, blog posts, fora posts, tweets.

For each project (H2020 & FP7) we create a set of keywords of the named entities that characterize the project. The queries will be created and executed on Qualia's search engine and will collect mentions of the project's keywords. A mention is a news article or a blog post or a forum post or a tweet that contains any of the project's keywords.

Data model: The format of the queries can be of two types, xml or json, depending on how expressive they need to be:

xml format

```
<query weight="10" Machine="CRAWLER" StemQuery="false" ExclusionTermList="" ExclusionStem=""
SourceFilter="" MaxDistance="100" UIText="" Lang="en" Country="de,uk">VascuBone</query>
```

json format

```
{"Queries":
[{"name":"Rule1",
  "AndClauses":
    [{"text":"AstraZeneca","AsIs":false,"proximity":0,"isPrefix":false},
     {"text":"Pfizer","AsIs":false,"proximity":0,"isPrefix":false}],
  "OrClauses":
    [[{"text":"innovation","AsIs":false,"proximity":0,"isPrefix":false},
     {"text":"strategy","AsIs":false,"proximity":0,"isPrefix":false},
     {"text":"new markets","AsIs":false,"proximity":0,"isPrefix":false},
     {"text":" partnership","AsIs":false,"proximity":0,"isPrefix":false},
     {"text":"algorithms","AsIs":false,"proximity":0,"isPrefix":false},
     {"text":"deep learning","AsIs":false,"proximity":0,"isPrefix":false},
     {"text":"artificial intelligence","AsIs":false,"proximity":0,"isPrefix":false},
     {"text":"big data","AsIs":false,"proximity":0,"isPrefix":false},
     {"text":"patent","AsIs":false,"proximity":0,"isPrefix":false}]],
  "NotClauses":
    [{"text":"ad","AsIs":false,"proximity":0,"isPrefix":false}],
     "languageExclusions":null,
     "languageInclusions":["en"],
     "domainExclusions":null,
     "domainInclusions":null,
     "selectedMedia":{"News":true,"Blogs":true,"Twitter":true,"Discussions":true}}],
  "id":1}
```


The result of the search will be a .csv file containing for each query the following metadata:

- Total Buzz: number of total mentions
- Sentiment Analysis: 3 numbers (% percentages) for positive/neutral/negative
- Source impact measurement: average global rank (in case of news sites, blogs), number of followers (tweets) in search results per query

The input queries will be used by Qualia's platform for searching. The detected results will be internally processed and the final output will be a .csv file containing the selected metrics (indicators).

Frequency of update: Once in the project, after the full set of project keywords are produced.

5.5 Social media/media metrics for SweCRIS and FP7 / H2020 projects

Twitter conversation threads will be collected and processed externally from the D4IP by UoB. The analysis will be done to be able to elicit in-depth performance measures of specific project papers that generate traction by rendering twitter conversations that span the mere 'mention' of the paper in the 'twitterverse'. By monitoring the Twitter API for document identifiers and noting 'hits' for project publications in our dataset, we can analyse the Twitter threads that occur in real time.

The Twitter streaming API will be utilised to filter the streams for tweets mentioning or referencing unique document identifiers, such as DOIs, CrossRef ID, Pubmed ID, Scopus ID or Web of Science UID, as well as the follow-on conversations connected to these tweets. These will then be matched to project publications using the same document identifiers.

The Twitter streaming API will also be utilized to follow the activities of a given number of accounts selected based on a set of criteria relevant for research. The conversations these accounts participate in will be collected.

The dataset will consist of tweets (actual text) including metadata such as timestamp, included hashtags, URLs, tweet replied to (if any), geolocation (if included by Twitter user). For those

tweets including or referring to a document identifier (such as DOI) or posted by a selected account that sparks a reaction in the form of replies, the conversation thread will be included in the dataset where each tweet is accompanied by the metadata described above.

Data model

- Retweets: number of users retweeting, number of retweets [integer or float depending on if data will be normalized, fractionalized];
- Conversation: number of users participating in the conversation and number of tweets in the conversation connected to the tweet [integer or float depending on if data will be normalized, fractionalized].

Altmetrical scores based on - Twitter mentions to project publications.

- Sentiment analysis, automatic classification of mention type based on tweet text. [textual classes, text];
- Semantic modelling between article abstract and twitter conversation contents. (E.g. a mere mention of a document or a copy-paste of the title is not as meaningful as actual interaction with the material.) [textual classes, text];
- Reach/spreadability, a measure based on number of followers, number of retweets and number of retweeters. [integer or float depending on if data will be normalized, fractionalized].

Impact data will be kept in a relational database that could be queried by SQL calls. Data could also be shared as tsv or any other text based metadata standard.

Exact format has not been established at this point but it would be good to establish a standard common for the project.

For Twitter conversation analyses the following would be needed (not conclusive):

- Document identifiers (DOI, CrossRef ID, PubMed ID, Scopus ID or Web of Science ID);
- Project identifier: EU/National projects, Call, Project ID, Project title;
- Twitter ID:s, part of Twitter conversation thread. Metadata for tweets, hashtag, retweet, meta-data at the user level. All personal data will be anonymized and shared data will be at the aggregate level, based on projects, affiliation data, national level or collaboration between different entities (as with the clinical guideline data);
- Impact measures according to schema above.

Upon activity, the data can be updated continuously. Dates for data delivery can be set according to needs.

Local input data sources

Social media data from websites/twitter (Qualia)

The data gathered in this task aims at building a dataset that contains all the articles and posts in the health-related web collection. For this purpose we compile a set of health-related sources: news sites, blogs, fora.

We collect all the articles and posts that are contained in this domain dependent dataset. The objective is to compare the ranking of the EU-funded project topics in terms of funding with the ranking of the health topics in terms of number of mentions (buzz).

The dataset consists of a collection of xml files. The schema of the files for news sites and fora follows:

news.xsd

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="NewsML">
      <xs:complexType>
      <xs:sequence>
      <xs:element name="NewsEnvelope">
      <xs:complexType>
            <xs:sequence>
            <xs:element type="xs:float" name="Version"/>
            <xs:element type="xs:dateTime" name="CreationTime"/>
            <xs:element type="xs:dateTime" name="LastUpdate"/>
            <xs:element type="xs:string" name="Comment"/>
            <xs:element type="xs:string" name="Creator"/>
            <xs:element name="Tool">
            <xs:complexType>
            <xs:simpleContent>
                  <xs:extension base="xs:string">
                  <xs:attribute type="xs:float" name="version"/>
                  </xs:extension>
            </xs:simpleContent>
            </xs:complexType>
            </xs:element>
            </xs:sequence>
            <xs:attribute type="xs:string" name="type"/>
      </xs:complexType>
      </xs:element>
      <xs:element name="NewsIdentifier">
      <xs:complexType>
            <xs:sequence>
            <xs:element type="xs:anyURI" name="ProviderId"/>
            <xs:element type="xs:string" name="NewsItemId"/>
           <xs:element type="xs:anyURI" name="NewsURL"/>
            <xs:element type="xs:dateTime" name="FirstCreated"/>
            <xs:element type="xs:dateTime" name="ThisRevisionCreated"/>
            <xs:element type="xs:anyURI" name="Source"/>
            <xs:element type="xs:string" name="Language"/>
            <xs:element type="xs:string" name="Country"/>
            <xs:element type="xs:string" name="Encoding"/>
            </xs:sequence>
      </xs:complexType>
      </xs:element>
      <xs:element name="NewsComponent">
```

```
        <xs:complexType>
                <xs:sequence>
                <xs:element type="xs:string" name="HeadLine"/>
                <xs:element type="xs:string" name="SummaryLine"/>
                <xs:element type="xs:string" name="TopicId"/>
                <xs:element type="xs:byte" name="Tdt"/>
                <xs:element type="xs:string" name="ClassificationScheme"/>
                <xs:element type="xs:string" name="ClassificationFile"/>
                <xs:element type="xs:byte" name="ClassificationLabel"/>
                <xs:element type="xs:string" name="DataContent"/>
                </xs:sequence>
        </xs:complexType>
        </xs:element>
        </xs:sequence>
        </xs:complexType>
  </xs:element>
</xs:schema>
```

forum.xsd

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="NewsML">
      <xs:complexType>
      <xs:sequence>
      <xs:element name="NewsEnvelope">
      <xs:complexType>
            <xs:sequence>
            <xs:element type="xs:float" name="Version"/>
            <xs:element type="xs:string" name="CreationTime"/>
            <xs:element type="xs:string" name="LastUpdate"/>
            <xs:element type="xs:string" name="Comment"/>
            <xs:element type="xs:string" name="Creator"/>
            <xs:element name="Tool">
            <xs:complexType>
            <xs:simpleContent>
                  <xs:extension base="xs:string">
                  <xs:attribute type="xs:float" name="version"/>
                  </xs:extension>
            </xs:simpleContent>
            </xs:complexType>
            </xs:element>
            </xs:sequence>
            <xs:attribute type="xs:string" name="type"/>
      </xs:complexType>
      </xs:element>
      <xs:element name="NewsIdentifier">
      <xs:complexType>
            <xs:sequence>
            <xs:element type="xs:anyURI" name="ProviderId"/>
            <xs:element type="xs:string" name="NewsItemId"/>
            <xs:element type="xs:string" name="NewsURL"/>
            <xs:element type="xs:string" name="FirstCreated"/>
            <xs:element type="xs:string" name="ThisRevisionCreated"/>
            <xs:element type="xs:anyURI" name="Source"/>
            <xs:element type="xs:string" name="Language"/>
            <xs:element type="xs:string" name="Country"/>
            </xs:sequence>
      </xs:complexType>
      </xs:element>
      <xs:element name="NewsComponent">
      <xs:complexType>
            <xs:sequence>
```

```
            <xs:element type="xs:string" name="HeadLine"/>
            <xs:element type="xs:string" name="SummaryLine"/>
            <xs:element type="xs:string" name="TopicId"/>
            <xs:element type="xs:string" name="ClassificationScheme"/>
            <xs:element type="xs:string" name="ClassificationFile"/>
            <xs:element type="xs:string" name="ClassificationLabel"/>
         <xs:element type="xs:string" name="DataContent"/>
            </xs:sequence>
      </xs:complexType>
      </xs:element>
      <xs:element name="SMAnalytics">
      <xs:complexType>
            <xs:sequence>
            <xs:element name="actor">
            <xs:complexType>
            <xs:sequence>
                  <xs:element type="xs:string" name="screen_name"/>
                  <xs:element type="xs:string" name="id"/>
                  <xs:element type="xs:string" name="klout_score"/>
            </xs:sequence>
            </xs:complexType>
            </xs:element>
            <xs:element type="xs:string" name="urls"/>
            <xs:element type="xs:string" name="user_mentions"/>
            </xs:sequence>
      </xs:complexType>
      </xs:element>
      </xs:sequence>
      </xs:complexType>
  </xs:element>
</xs:schema>
```

The dataset will be a typical textual corpus in xml format, organized by media type, which will be provided as is for processing and once operational.

Frequency of update: Once in the project, after the full set of articles and posts is collected.