



HORIZON 2020 - ICT-14-2016-1

AEGIS

Advanced Big Data Value Chains for Public Safety and Personal Security

WP2 - Core Data Value Chain Transformation and Handling Methods



D2.3 - Update on Semantic Representation and Data handling and Analytics Methods

Due date: 30.06.2018

Delivery Date: 06.07.2018

Author(s): Spiros Mouzakitis, Evmorfia Biliri (NTUA), Konstantinos Perakis, Dimitrios Miltiadou (UBITECH), Yury Glikman, Fabian Kirstein (Fraunhofer), Mahmoud Ismail, Alexandru Ormenisan (KTH), Elisa Rossi (GFT), Sotiris Koussouris, Spiridon Koussouris, Penelope Dima, Marios Phinikettos (SUITE5), Gianluigi Viscusi (EPFL)

Editor: Elisa Rossi (GFT)

Lead Beneficiary of Deliverable: GFT

Dissemination level: Public **Nature of the Deliverable:** Report

Internal Reviewers: Yury Glikman (Fraunhofer), Dimosthenis Tsagkrasoulis (Hypertech)

EXPLANATIONS FOR FRONTPAGE

Author(s): Name(s) of the person(s) having generated the Foreground respectively having written the content of the report/document. In case the report is a summary of Foreground generated by other individuals, the latter have to be indicated by name and partner whose employees he/she is. List them alphabetically.

Editor: Only one. As formal editorial name only one main author as responsible quality manager in case of written reports: Name the person and the name of the partner whose employee the Editor is. For the avoidance of doubt, editing only does not qualify for generating Foreground; however, an individual may be an Author - if he has generated the Foreground - as well as an Editor - if he also edits the report on its own Foreground.

Lead Beneficiary of Deliverable: Only one. Identifies name of the partner that is responsible for the Deliverable according to the AEGIS DOW. The lead beneficiary partner should be listed on the frontpage as Authors and Partner. If not, that would require an explanation.

Internal Reviewers: These should be a minimum of two persons. They should not belong to the authors. They should be any employees of the remaining partners of the consortium, not directly involved in that deliverable, but should be competent in reviewing the content of the deliverable. Typically this review includes: Identifying typos, Identifying syntax & other grammatical errors, Altering content, Adding or deleting content.

AEGIS KEY FACTS

Topic:	ICT-14-2016 - Big Data PPP: cross-sectorial and cross-lingual data integration and experimentation
Type of Action:	Innovation Action
Project start:	1 January 2017
Duration:	30 months from 01.01.2017 to 30.06.2019 (Article 3 GA)
Project Coordinator:	Fraunhofer
Consortium:	10 organizations from 8 EU member states

AEGIS PARTNERS

Fraunhofer	Fraunhofer-Gesellschaft zur Förderung der angewandten Forschung e.V.
GFT	GFT Italia SRL
KTH	Kungliga Tekniska högskolan
UBITECH	UBITECH Limited
VIF	Kompetenzzentrum - Das virtuelle Fahrzeug , Forschungsgesellschaft-GmbH
NTUA	National Technical University of Athens - NTUA
EPFL	École polytechnique fédérale de Lausanne
SUITE5	SUITE5 Limited
HYPERTECH	HYPERTECH (CHAIPERTEK) ANONYMOS VIOMICHANIKI EMPORIKI ETAIREIA PLIROFORIKIS KAI NEON TECHNOLOGION
HDIA	HDI Assicurazioni S.P.A

Disclaimer: AEGIS is a project co-funded by the European Commission under the Horizon 2020 Programme (H2020-ICT-2016) under Grant Agreement No. 732189 and is contributing to the BDV-PPP of the European Commission.

The information and views set out in this publication are those of the author(s) and do not necessarily reflect the official opinion of the European Communities. Neither the European Union institutions and bodies nor any person acting on their behalf may be held responsible for the use which may be made of the information contained therein.

© Copyright in this document remains vested with the AEGIS Partners

EXECUTIVE SUMMARY

The document at hand, entitled “Update on Semantic Representation and Data handling and Analytics Methods”, constitutes the final report of work package 2, documenting the results of its four tasks.

At first, updated definitions of the AEGIS Data Value Chain Bus following the incoming needs from the other WPs (mainly WP3), as well as of its main processes (data harvesting and annotation) are provided (T2.3). The means for collecting, harmonising and processing data and metadata following both of the technical and the user requirements are investigated, driving to a redefinition of the AEGIS data harmonisation approach.

The final design of the infrastructure for accessing and sharing Linked Big Data Vocabularies and Metadata is presented (T2.1): it is built on top of the Linda Workbench¹ infrastructure and existing standards and schemas adapted to the needs of the AEGIS stakeholders. To access the complete list of the vocabularies and ontologies available within the AEGIS Platform, please refer to D2.1.

Complete and detailed lists of the big data analytics and business intelligence functionalities as well as algorithms for visualisation are provided (T2.4), although the intention is not to consider them as final lists, since further needs of the AEGIS stakeholders or demonstrators could lead to further updates.

Finally, an updated definition of the Data Policy and Business Brokerage Frameworks (T2.2) is provided. The AEGIS Data Policy Framework is a conceptual realisation of the approach of the project towards categorising assets in the AEGIS platform. The Business Brokerage Framework (BBF) formally dictate data-related assets transaction terms and oversee the smooth and rightful execution of them.

¹ <http://linda.epu.ntua.gr/>

Table of Contents

EXPLANATIONS FOR FRONTPAGE.....	2
AEGIS KEY FACTS	3
AEGIS PARTNERS.....	3
EXECUTIVE SUMMARY.....	4
LIST OF FIGURES	6
LIST OF TABLES	6
ABBREVIATIONS	7
1. INTRODUCTION.....	8
1.1. OBJECTIVE OF THE DELIVERABLE	8
1.2. INSIGHTS FROM OTHER TASKS AND DELIVERABLES	8
1.3. STRUCTURE	8
2. AEGIS DATA VALUE CHAIN BUS	10
3. DATA HARVESTING	11
3.1. GOAL	11
3.2. REQUIREMENTS	12
3.3. HARVESTER DESIGN.....	13
4. DATA AND METADATA HARMONISATION	15
4.1. GOAL	15
4.2. REQUIREMENTS.....	15
4.3. AEGIS DOMAIN VOCABULARIES AND VOCABULARY REPOSITORY	16
4.4. DATA HARMONISATION	19
4.5. METADATA HARMONISATION	22
5. KNOWLEDGE EXTRACTION & BUSINESS INTELLIGENCE.....	24
5.1. DIMENSIONALITY REDUCTION – FEATURE EXTRACTION HYPOTHESIS TESTING.....	25
5.2. NLP FUNCTION ALGORITHMS.....	28
5.3. RECOMMENDERS	31
5.4. CLUSTERING ALGORITHMS	32
5.5. CLASSIFICATION/REGRESSION ALGORITHMS	34
6. VISUALISATION.....	45
6.1. TOOLS AND LIBRARIES	45
6.1.1. Jupyter	46
6.1.2. Highcharts	46
6.1.3. Folium	46
6.2. VISUALISATIONS TECHNIQUES IN AEGIS	47
6.3. VISUALISATION TECHNIQUES RELEVANCE TO THE AEGIS DEMONSTRATORS	52
6.4. AEGIS REQUIREMENTS MET	54
7. DATA POLICY AND BUSINESS BROKERAGE FRAMEWORKS	56
7.1. DATA POLICY FRAMEWORK UPDATES	56
7.2. BUSINESS BROKERAGE FRAMEWORK UPDATES.....	61
8. CONCLUSION.....	65

LIST OF FIGURES

Figure 1: AEGIS Data Value Chain Bus Components	10
Figure 2: The Vocabularies page (Vocabularies, Classes and Properties views)	17
Figure 3: Part of the visualization of the DCMI Metadata Terms vocabulary	18
Figure 4: Creating a new vocabulary through the UI	19
Figure 5: Example of a scatter plot in AEGIS	48
Figure 6: Example of pie chart in AEGIS	48
Figure 7: Example of bar chart in AEGIS	49
Figure 8: Example of line chart in AEGIS	49
Figure 9: Example of a box plot in AEGIS	50
Figure 10: Example of histogram in AEGIS	50
Figure 11: Example of time series in AEGIS	51
Figure 12: Example of heatmap chart in AEGIS	51
Figure 13: Example of a bubble chart in AEGIS	52
Figure 14: Example of plotting on a map in AEGIS	52
Figure 15: High-level Data Asset Policy Concept in AEGIS	57
Figure 16: AEGIS DPF Properties	60
Figure 17: AEGIS BBF Concept	62
Figure 18: AEGIS BBF JSON Example	63

LIST OF TABLES

Table 1: Harvesting Requirements	12
Table 2: Data and Metadata Harmonisation Requirements	15
Table 3: Metadata Repository Requirements	15
Table 4: DPF Properties Split amongst AEGIS DataStore and Brokerage Engine	60

ABBREVIATIONS

CO	Confidential, only for members of the Consortium (including the Commission Services)
D	Deliverable
DoW	Description of Work
H2020	Horizon 2020 Programme
FLOSS	Free/Libre Open Source Software
GUI	Graphical User Interface
IPR	Intellectual Property Rights
MGT	Management
MS	Milestone
OS	Open Source
OSS	Open Source Software
O	Other
P	Prototype
PU	Public
PM	Person Month
R	Report
RTD	Research and Development
WP	Work Package
Y1	Year 1

1. INTRODUCTION

The scope of the current section is to introduce the deliverable and familiarise the user with its contents. Towards this end, this section summarises the objective of the deliverable, its relation to the other work packages and analyses its structure.

1.1. Objective of the deliverable

D2.3 is the final report of WP2 ending at M18 and it documents the efforts undertaken within the context of the work package.

The current deliverable provides the final definition of the semantic representations and linked data vocabularies necessary for describing the data within the AEGIS platform (T2.1), as well as the main methods and data schemas for the AEGIS Data Policy and Business Mediator Frameworks (T2.2). Moreover, the final design of the AEGIS Data Value Chain Bus, the methods for processing and harmonising heterogeneous data and metadata from different sources (T2.3), as well as the algorithms for knowledge extraction and visualisation (T2.4), is described.

1.2. Insights from other tasks and deliverables

The current deliverable builds directly on top of the previous two WP2 deliverables, namely D2.1 “Semantic Representations and Data Policy and Business Mediator Conventions” and D2.2 “AEGIS Data Value Chain Bus Definition and Data Analysis Methods”, and contains the final decisions of the AEGIS partners regarding all the tasks of the work package.

The content of the following sections is an updated version of the content of the aforementioned WP2 deliverables and is mainly related to the decisions taken within the contexts of WP1 concerning the AEGIS Big Data Value Chain, WP3 and WP4 concerning the technical development of the AEGIS platform.

In addition, the feedback of the demonstrators (WP5) is relevant in particular regarding the algorithms for knowledge extraction and visualisation.

1.3. Structure

Deliverable 2.3 is organised in eight main sections as indicated in the table of contents.

The first section introduces the deliverable, documenting its scope and briefly describing how it is structured. It also documents the relation of the current deliverable with the other deliverables, and how the knowledge produced in the other deliverables and work-packages served as input to the current deliverable.

After a conceptual description of the AEGIS Data Value Chain Bus (section 2), the Data Harvesting and the Data and Metadata Harmonisation are investigated (section 3 and 4), starting from the goal and the requirements to meet within the AEGIS context. The final design and methodologies offered by the Data Harvester, as well as the Vocabularies and Vocabularies Repository are presented. It is important to point out that the proposed solutions are based on the re-use of advanced and proven tools and methodologies, ensuring interoperability with existing approaches.

The section 5 and section 6 present respectively an overview of the analytics algorithms and of the visualisation techniques, tools and libraries considered for the project, with an accent on the needs of the three AEGIS demonstrators, while section 7 focuses on the design of the Data Policy Framework as well as the Business Brokerage Framework.

Section 8 concludes the deliverable, outlining the final statements of WP2, which will guide the future research and technological efforts of the consortium in the technical work packages that will continue in the forthcoming months.

2. AEGIS DATA VALUE CHAIN BUS

AEGIS aspires to foster data-driven innovation in various public safety and personal security applications, a vision that requires the effective handling of the inherent data heterogeneity, both in terms of content and format, of the relevant underlying domains. The updated AEGIS architecture, presented in D3.3 "Technical and User Requirements and Architecture v2.00", provides the technical descriptions and specifications for the components that constitute the AEGIS data value chain bus, i.e. the components that are involved in the realisation of the initial steps of the AEGIS data value chain, specifically the ones related with connecting data to the AEGIS system and subsequently harmonising and semantifying them, in order to make them available to more advanced business intelligence and visualisation tasks.

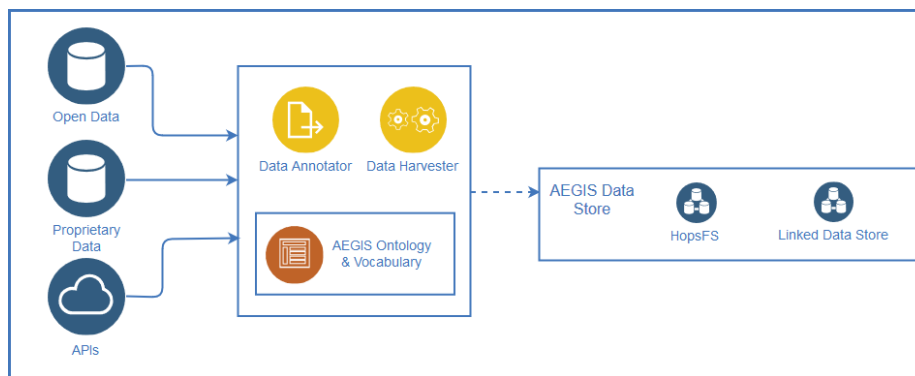


Figure 1: AEGIS Data Value Chain Bus Components

The data value chain bus is in essence a conceptual component that includes mainly the processes of data harvesting and annotation, as well as a set of harmonisation services. The following two sections provide detailed descriptions of how these processes are designed and delivered in AEGIS.

3. DATA HARVESTING

Harvesting is the process of extracting or fetching data from a data source. This process includes all necessary means to make data from a specific source available for further processing. Within the AEGIS Data Value Chain this process is the first step of the Data Acquisition process. This definition applies for data and metadata as well.

3.1. Goal

The accessibility to a wide range of data and metadata is one of the key features of the AEGIS platform. Even beyond the initial requirements of the pilot scenarios, the value and sustainability of the platform correlates with the availability of useful data. Hence, the main objective of harvesting in the AEGIS platform is to provide a methodology and mechanisms to access a large variety of data sources.

In deliverable D2.2 characteristics of data sources were defined and described. These characteristics need to be respected in the design of the data harvester. The characteristics are still valid and do not require an update. Below a short summary:

- **Protocol**
The following basic protocols were identified as the most relevant for the AEGIS platform: HTTP, HTTPS, FTP and SFTP. This list may be not complete at this stage and the final implementation should be generic, allowing the extension to further protocols.
- **Serialisation format**
Where the protocol defines on which way the data is transported, the serialisation format states how the data is actually structured. AEGIS has to support and deal with a variety of serialisation formats. Based on the findings of D1.1 and the defined scenarios of the client applications the most relevant formats are: tabular formats (XLSX, XLS, CSV, TSV), textual formats (TXT, RTF), XML, Web Map Service (WMS), JSON and relational database formats.
- **Semantics**
The various serialisation formats mainly describe the syntax of the data. In addition, some formats do support derivatives, allowing to further describe the meaning of the data, hence the semantics.
- **Offline or real-time**
This characteristic describes how the actual creation time of a dataset relates to its time of availability. Offline describes datasets that are not available at the creation time, but a significant amount of time later. Real-time datasets refer to an availability in the very instance of creation or seconds after it.
- **Update interval**
Closely connected to the offline and real-time data characteristic is the update interval of a dataset. Assuming an offline dataset, it affects directly the frequency of the harvesting process. Since the update interval differs from source to source, the design should consider this as a customisable setting to adjust the process to the circumstances.
- **Pull or push**
A fundamental difference exists in the manner of retrieving the data – either by a pull or by a push mechanism. In the first cases, the harvester actively calls for a resource to

retrieve the data. The push mechanism shifts the control to the provider of the data. The harvester needs to provide an endpoint, which is invoked, whenever the publisher sees fit. Typically pull mechanisms are used for offline data with a small update frequency, whereas push mechanisms are applied for real-time data, especially in the context of the Internet of Things. The AEGIS platform should support both paradigms, since both have relevant use cases in the context of the project.

- **Size**

This characteristic describes the actual size of the data, which needs to be harvested. Since the harvester and the following transformation have to process the data, the size depicts an important property. The AEGIS platform will be designed under the premise of working with Big Data. Hence, the harvester design has to consider a very big size.

- **Security**

The interface of a data source may be completely open or require authentication and/or authorisation by the user. The latter needs to be considered in the design of the harvester, since the retrieving mechanism needs to support the applied security methods.

- **Language**

Datasets may be available in multiple language. How this is presented and implemented by the data source needs to be taken into account for the harvester, since this data should not get lost during the process.

3.2. Requirements

The requirements of deliverable D2.2 were updated and revised. Mostly the need for supporting the harvesting from database systems was removed. The development of the use cases has shown that there is no demand for this functionality. Data from internal database systems is pre-processed and provided via other protocols, like HTTP.

Table 1: Harvesting Requirements

ID	HV Requirements	Previous Requirements of Reference
HV1	The harvesting process should be able to retrieve data via the following application protocols: HTTP, HTTPS, FTP, SFTP, WSS, WS.	TR1-9
HV3	The harvesting process has to parse the following tabular file formats: XLS, XLSX, CSV and TSV.	TR1-9
HV4	The harvesting process has to read the following text formats: TXT, RTF, DOC and DOCX.	TR1-9
HV5	The harvesting process has to read the structured formats XML and JSON.	TR1-9
HV6	The harvesting process has to deal with the following Linked Data formats: JSON-LD and RDF/XML.	TR1-9
HV8	The harvesting process has to support the guided extraction of semantic information from the data sources.	TR1-9

HV9	The harvesting process has to support the data retrieval from offline and real-time data sources.	TR1-9
HV10	The harvesting process has to be adjustable to the given update interval of a data source.	TR1-9
HV11	The harvesting process should support both, pull and push mechanisms for collecting data.	TR1-9
HV12	The harvesting process should facilitate the fetching of data in the size of several megabyte.	TR1-9
HV13	The harvesting process has to support the data source and protocol specific authentication and authorisation schemas.	TR1-9
HV14	The harvesting process should support the HTTP authentication by API-key and htaccess.	TR1-9
HV15	The harvesting process should be able to process multi-lingual data sources, either encoded in the data itself or by distinct addresses.	TR1-9

3.3. Harvester Design

The design of the harvester was updated and extended to fulfil the requirements. The updates reflect the experiences made implementing the AEGIS platform and pilots' use cases.

In deliverable D2.2 a generic, programmatic approach for developing custom connectors for different data sources was described. This approach was not feasible in the scope of the AEGIS project. The real characteristics of various data sources differ too much. This led to a revised architecture based on interchangeable microservices for reflecting the basic harvesting workflow. This workflow consists out of four stages, where each stage is represented by a microservice. Each service stands alone and has to implement a specified interface based on RESTful principles and uses JSON as serialisation format. In the following, the microservices are described:

Importer

An importer implements all functionalities for retrieving data from a specific data source. It needs to specifically support the characteristic of that data source, including protocol, serialisation format, security etc. It has to export the harvested data as JSON to the next stage.

Transformer

A transformer converts the retrieved data from an importer into the target format of the AEGIS platform, hence, a tabular format. It is specific for each data source. A basic transformer is available, allowing to provide scripts (JavaScript) for creating custom transformations from source to target.

Aggregator

An aggregator collects converted data from a transformer over a configurable time interval. It allows to adjust the granularity of the available data in one file within the AEGIS platform. A basic implementation is available.

Exporter

The exporter uploads transformed and/or aggregated data to the AEGIS platform. In addition, it creates the corresponding metadata in the AEGIS metadata store. There will be only one implementation for the exporter.

The four microservices are orchestrated with a web application, where the specific implementation for each stage can be chosen and configured.

4. DATA AND METADATA HARMONISATION

4.1. Goal

The goal of the harmonisation is to decrease the heterogeneity of metadata and data to be managed in the AEGIS platform. It will be done by transforming and refining the collected (harvested) raw data and metadata to the AEGIS adopted data format(s), data structure(s) and vocabularies. This simplifies further processing and management of data and metadata in AEGIS.

4.2. Requirements

This section presents the list of requirements of the data and metadata harmonisation and the metadata repository.

Table 2: Data and Metadata Harmonisation Requirements

ID	Data and Metadata Harmonisation Requirements	Previous Requirements of Reference
DMH1	AEGIS should be able to harmonise metadata describing data coming from sensor nodes, including wearable devices, smartphones, smart home devices, car sensors and other IoT devices to linked data format.	TR1-9
DMH2	AEGIS has to be able to harmonise tabular data from file formats: XLS, XLSX, CSV and TSV to a selected format in which data will be stored in AEGIS.	TR1-9
DMH3	AEGIS should include a tool for refining and interlinking of metadata.	TR1-9
DMH4	AEGIS should be able to harmonise temporal information, both low-level (e.g. UNIX timestamps) and high-level (e.g. year)	TR1-9

Table 3: Metadata Repository Requirements

ID	Data and Metadata Harmonisation Requirements	Previous Requirements of Reference
MDR1	Insert new vocabularies / ontologies in the Metadata Repository	TR1-9
MDR2	Delete vocabularies / ontologies from the Metadata Repository	TR1-9
MDR3	Update vocabularies / ontologies in the Metadata Repository	TR1-9
MDR4	Insert metadata about vocabularies / ontologies in the Metadata Repository	TR1-9

MDR5	Search vocabularies / ontologies in the Metadata Repository based on different criteria and keywords	TR1-9
MDR6	Evaluate SPARQL queries over the Metadata Repository to collect metadata about vocabularies / ontologies	TR1-9
MDR7	Evaluate SPARQL queries over the Metadata Repository to retrieve classes and properties of vocabularies / ontologies	TR1-9
MDR8	Search and identify PSPS datasets semantically enriched with particular vocabularies / ontologies	TR1-9
MDR9	Ensure persistence of the Metadata Repository	TR1-9
MDR10	Ensure web-based access and availability of the Metadata Repository	TR1-9
MDR11	Download data dumps of the Repository vocabularies / ontologies	TR1-9
MDR12	Provide a recommendation system on top of the Metadata Repository	TR1-9
MDR13	Search for related vocabularies / ontologies in the Metadata Repository	TR1-9
MDR14	Search pilots using particular vocabularies / ontologies	TR1-9

4.3. AEGIS Domain Vocabularies and Vocabulary Repository

The AEGIS Vocabulary Repository, which is built on top of the Linda Workbench infrastructure², allows registering, describing, and searching vocabularies and thus significantly facilitates the exploration of ontologies and vocabularies that can be utilised in the diverse AEGIS applications. The repository enables the discovery of potentially useful vocabularies for the various data sources and also accelerates the design and update of the AEGIS ontology. It also supports a variety of more advanced capabilities like transformation to RDF, analytics, visualizations, and more. The list of vocabularies previously available in LinDa (which were more than 300) has been augmented with additional vocabularies and ontologies related to the public safety and personal security domains, which indicatively include the ones described in Table 3.5 of Deliverable D2.1.

The vocabulary repository serves the purpose of presenting the final user with various ontologies, supporting the transformation of traditional data formats to Linked Data by suggesting classes and properties. The usage of the repository can take place with actions that can be grouped in the following categories:

² <http://linda.epu.ntua.gr/>

- 1. Navigation:** Actions that let the user search for vocabularies and entities inside them, read vocabulary descriptions, download the vocabulary RDF documents in various formats and get access to vocabulary visualisations and best usage practices.

When users navigate to the “Vocabularies” page, they are shown a catalogue of all repository entities, which they can select to view by vocabularies, classes, or properties:

LinDA - Vocabulary search

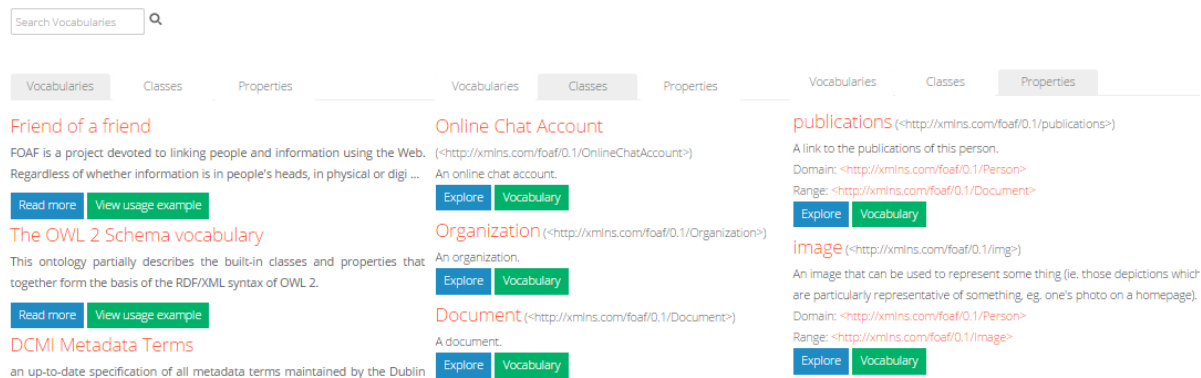


Figure 2: The Vocabularies page (Vocabularies, Classes and Properties views)

By selecting a vocabulary, users get access to a page with more details about the selected vocabulary, which also allows them to perform actions on it, depending on their current role and permissions on the website.

The vocabulary page contains:

- Some basic information about the vocabulary, like its namespace URI, the prefix that is commonly used for it, a link to the website where it is defined (like a W3C recommendation document or a website dedicated to the vocabulary) and a short description of its purpose and contents.
- Links to the source vocabulary document, both in its original version and in an automatically created RDF in all supported serializations (RDF/XML, n3 and NTriples), as well as a link to an automatically created vocabulary visualisation.
- Metadata about the vocabulary owner and when it was created.
- Information about classes and properties that it defines.
- Feedback controls, including rate and comment capabilities for authenticated users.
- A usage example that indicates how the major entities defined in the vocabulary are supposed to be used in order to create semantically correct RDF documents (optional).
- A button that opens the visualization of the vocabulary and offers a quick view of the ontology (the number of elements that can be visualized in the web page is limited in order to avoid information overload)

DCMI Metadata Terms

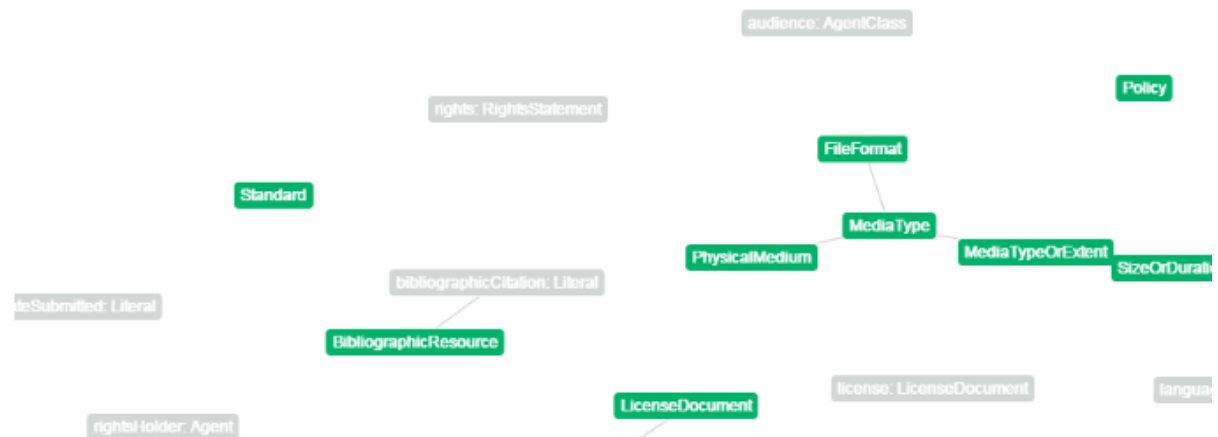
[Back to info](#)

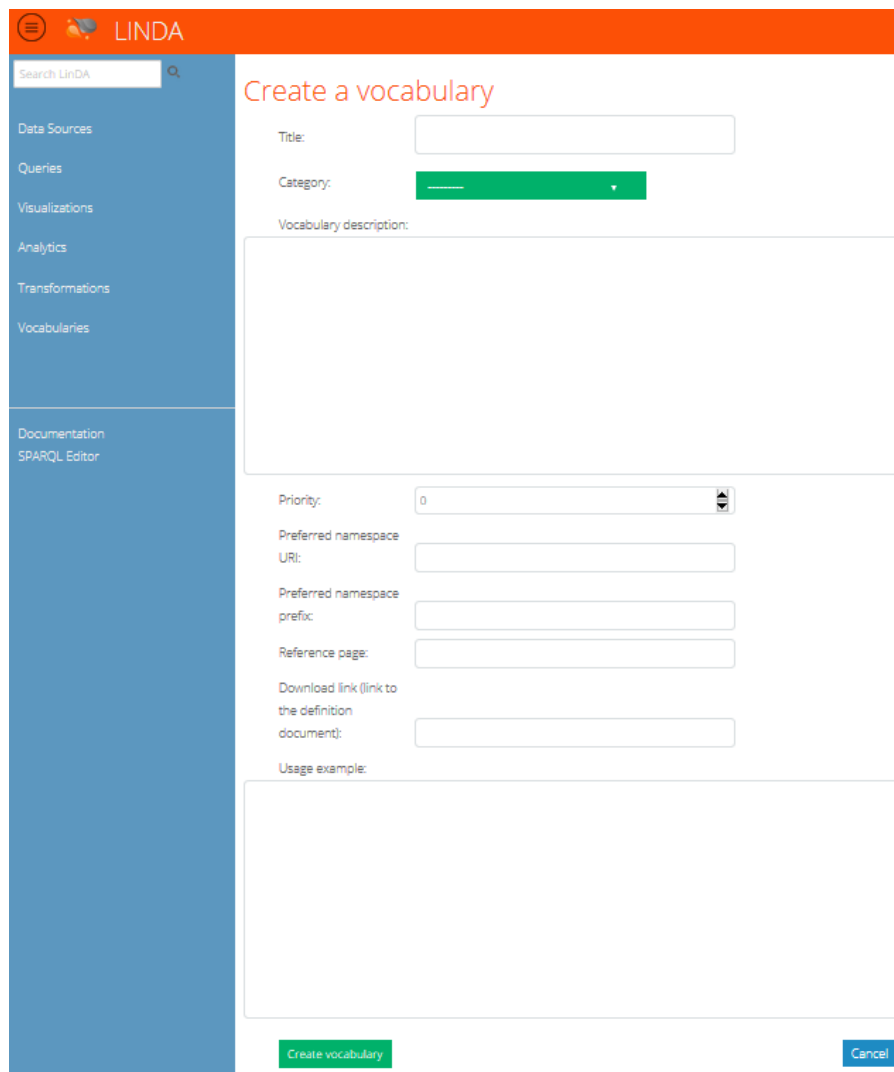
Figure 3: Part of the visualization of the DCMI Metadata Terms vocabulary

Users can also view the details of both classes and properties that have been extracted by the installed vocabularies.

2. **Usage feedback:** Evaluation of vocabularies, discussions and commenting, that expose the advantages and disadvantages of choosing a vocabulary's terminology to create transformation plans and guide the user base of an enterprise to vocabularies best representing its structure, operations and needs.

Evaluation of the presented material and community discussions are useful tools in order to promote the appropriate material according to each enterprise community needs and solve questions and problems that end users may face. In the LinDA Vocabulary and Metadata repository, two main mechanisms are available to let users express feedback and interact with each other: vocabulary rating and vocabulary discussions.

- Repository enrichment:** Authenticated users may create and upload new vocabularies containing ontologies that do not exist to the initial repository or are specific to the enterprise. Vocabulary owners may further update their vocabularies at any times. The repository automatically extracts metadata information contained in the vocabulary RDF document like classes and properties, as well as their relations.



The screenshot shows the LINDA web interface. On the left is a blue sidebar with a search bar and a menu containing: Data Sources, Queries, Visualizations, Analytics, Transformations, Vocabularies, Documentation, and SPARQL Editor. The main content area has an orange header with the LINDA logo and a search bar. Below the header, the title 'Create a vocabulary' is displayed in red. The form includes the following fields: Title (text input), Category (dropdown menu with a green bar), Vocabulary description (large text area), Priority (input with value 0 and a spinner), Preferred namespace URI (text input), Preferred namespace prefix (text input), Reference page (text input), Download link (link to the definition document) (text input), and Usage example (large text area). At the bottom of the form are two buttons: 'Create vocabulary' (green) and 'Cancel' (blue).

Figure 4: Creating a new vocabulary through the UI

- 4. Term suggestion:** Web API methods pick the most prevalent vocabulary terms that describe real world objects and relationships.

4.4. Data Harmonisation

AEGIS aims to support numerous data sources and heterogeneous data formats. This makes data analysis and visualisation of data complex tasks, especially when data from different sources and in different arbitrary formats have to be visualised in one chart or somehow analysed together on the request of the end user.

A pragmatic approach to this issue is to perform transformation of raw data to bring it in the format(s) acceptable by the AEGIS data analytics and visualisation tools. The data transformation can be seen as part of the data harvesting process. In case of a lossless transformation, there is no need to keep the original raw data in the AEGIS platform. In other cases, AEGIS should save the original raw data in own data store to enable its processing in future.

The Big Data Cluster provides the means to store and process big data files within the context of the AEGIS project. The Big Data Cluster will be running the software stack provided by Hops Hadoop. The core component of Hops Hadoop is the distributed file system (HopsFS) which is a reliable highly scalable distributed file system that stores massive volumes of data across thousands of machines. On top of HopsFS, multiple processing frameworks such as Spark and Flink can be easily used. Due to the nature of the file system, a user can store any type of data as is without any constraints being placed on how the data is processed. A file in HopsFS can vary in size from kilobytes growing potentially to terabytes of data. Under the hood, HopsFS divides the files into multiple blocks and reliably replicates these blocks across the machines in the cluster.

In the Hadoop/Hops world, users simply store raw data into the system, and then impose the structure at the processing time based on the application requirements. This approach is called Schema-on-Read, an alternative to a well-known approach, Schema-on-Write, which is widely used in traditional data management systems. In the Schema-on-Write approach, the data structure is imposed beforehand at the time of writing the data, that makes it not as agile and as flexible as the Schema-on-Read approach.

Although a user can potentially store any type of data with any kind of format on Hops, there are some considerations need to be taken into account such as how big are the files, what kind of processing and query tool will be used, and what are the performance requirements for read and write.

The following standard file types and specific file formats can be used in Hops/Hadoop:

1. **Standard File Types.** As noted before, a user can store any kind of data on Hops regardless of format. A file could contain text data (such as comma separated files, emails, or log files), structured text data (such as XML files), and binary data (such as image, and videos).
 - a) **Text Data** comes in many forms for example comma separated files (CSV), or unstructured data such as emails or server logs. It can very quickly consume considerable amount of storage space on the cluster. That is, storing data as text is not always efficient for example storing integer as text require more space and conversion tools are needed for converting from string to integer and vice versa. Usually, compression is a good idea with such formats but the user must also take into account the usage pattern of the data and the performance of the compression algorithm. In most of the cases, transforming these data into SequenceFile or Avro format is preferable since these formats provide better compression support and are splittable.
 - b) **Structured Text Data** is a more specialized form of text data such as XML and JSON. These formats impose a tricky challenge while storing since they are not splittable by nature, meaning that it is not possible to split the file into disjoint blocks to exploit the parallel processing nature of frameworks such as MapReduce, Spark, and Flink. Luckily, Transforming such data into Avro format, for example, provide a more compact and efficient way to process data.
 - c) **Binary data** such as images, videos or more generally speaking any sequence of bytes can be stored also in Hops. These data can be stored as is, or in a container format such as Avro.
2. **Hadoop File Types.** MapReduce and most of the parallel processing frameworks leverage the idea of data decomposability. That is, decomposing the data into smaller chunks, and

then work in parallel on each chunk. There are several Hadoop-specific file formats that were specifically created to work well with such frameworks. These formats supports common compression formats and are splittable. They include file-based data structures such as Sequence Files, serialization formats such as Avro, and columnar formats such as Parquet.

- a) **File-based data structures.** These formats include SequenceFiles, MapFiles, SetFiles, ArrayFiles, and BloomMapFiles. The SequenceFile format is the most commonly used format in Hadoop. These formats are well supported within the Hadoop eco-system. The SequenceFile is a flat file consisting of binary key/value pairs. There are three available formats for records stored in SequenceFile; uncompressed key/value records, record compressed key/value records, and block compressed key/value records. The SequenceFiles are well supported within the Hadoop ecosystem, however outside of the ecosystem their support is limited. Moreover, they are only supported in Java.
- b) **Serialization formats.** Data Serialization is the process of translating data into a byte stream that can be stored or transferred over the network. Similarly, deserialization is the counter process of turning the data back into the original format. The main serialization format used by Hadoop is Writable, but it suffers from many limitations for example it is not easily extendable. There are different serialization frameworks such as Thrift, Protocol Buffers, and Avro that were developed to address the limitations of Hadoop writables.
 - **Thrift** is an interface definition language that was developed by Facebook for scalable cross-language service development. It provides a cross-language serialization that can be used to translate a single interface between different languages. It is used sometimes as a data serialization framework within Hadoop, however, it lacks support for internal record compression, it is not splittable, and lacks native support in MapReduce.
 - **Protocol Buffers** was developed by Google, and it is also used for serializing data structures between different languages similar to Thrift. It can be used for data serialization in Hadoop but it lacks support for internal record compression, is not splittable, and lacks native support in MapReduce.
 - **Avro** is a data serialization framework that was developed to address the limitations of Hadoop Writable format. Like Thrift and Protocol Buffers, it uses a language independent format to describe data. However, it has a better native support for MapReduce since Avro data files are compressible and splittable.
- c) **Columnar formats.** The conventional wisdom was to store the data into a row-oriented fashion. This makes sense for queries reading all the fields from a bunch of rows. Nevertheless, for queries working on a subset of columns, row-oriented formats will not be that efficient. In addition, usually a lot of repetition happen within the values of a column which impose the need for compression on columns. There are different column-oriented formats such as RCFile, ORC, and Parquet.
 - **Record Columnar File (RCFile)** is a column-oriented data storage format that was developed to be used by MapReduce applications. It is used in Hive as one of the data storage formats. It writes the data into row splits, and within a split it writes the columns in a column-oriented format. It has some limitations in terms of query performance and compression that encouraged the move to better columnar formats such as ORC and Parquet.

- **Optimized Row Columnar (ORC)** is a column-oriented data storage format that was developed to overcome the shortcomings of the RCFile format. It provides a lightweight always on-compression, and predicates push down for efficient query processing. It writes files into stripes, where each stripe is independent of all other stripes. Within each stripe, the data is written in a column-oriented fashion. It is also supported by Hive.
- **Parquet** is a column-oriented data storage format that shares the same design goals as ORC, but it is designed to be a more general-purpose storage format for Hadoop. It provides efficient compression that can be specified on per-column level. It also supports complex nested data structures. It is compatible with most of the data processing frameworks in Hadoop eco-system such as Hive, Pig, Impala, and Spark. In addition, it can be easily used with Avro and Thrift since they fully support reading/writing Parquet files.

The main input raw data formats, which AEGIS has to support are the following:

- Tabular file formats: XLS, XLSX, CSV and TSV;
- Text formats: TXT, RTF, DOC and DOCX;
- Structured formats XML and JSON;
- Linked Data formats: JSON-LD and RDF/XML.

Although Hadoop supports rich tabular formats, the optimal choice for storing tabular raw data in AEGIS is CSV. It represents a very simple and common format, which can be easily processed and understood by users of AEGIS. A later conversion within the AEGIS platform to richer formats is always possible. Some XML and JSON files can be transformed to CSV as well if they contain tabular data. In case if lossless transformation is not possible the original raw data files will be saved in the data store as well. Linked Data harvested as JSON-LD and RDF/XML files will be stored in AEGIS in a triplestore. Textual files will be saved in the AEGIS platform in their original raw format. Raw data in XML and JSON format if it cannot be saved a CSV will be stored in the original format in the AEGIS data store.

Apart, from transforming raw data to a selected in the AEGIS data format some data elements can be harmonised as well, for example, the date and time writing style. Date and time formats used in data and metadata can vary depending on the data source or even the concrete dataset. AEGIS will provide a possibility to automatically harmonise date and time data by transforming them to the subset of the ISO 8601³ format.

4.5. Metadata Harmonisation

All metadata collected by the AEGIS harvester will be transformed to linked data using the DCAT-AP ontology as well as the AEGIS ontology describing the structure and content of data as well as domain specific ontologies as it is explained in the deliverable D2.1⁴. The metadata transformation service works in a pipeline together with the harvester. That means each harvesting pipeline starts with an importer, addressing the specific needs of the source portal, a

³ http://www.loc.gov/standards/datetime/iso-tc154-wg5_n0039_iso_wd_8601-2_2016-02-16.pdf

⁴ D2.1 – Semantic Representations and Data Policy and Business Mediator Conventions

transformation running a transformation script on each single metadata/data record, and an exporter feeding the AEGIS data store with the transformed (meta)data record.

The achieving high harmonisation and interlinking of metadata, however, requires manual refinement of metadata. AEGIS will support it with its Data Annotator tool. In order to achieve interlinking of data and connecting them with other data across the web, the AEGIS Data Annotator will use the Named Entity Recognition (NER) service developed in the LinDA project. It will help users to replace the literals in metadata by appropriate URIs (e.g., the string “Athens” is replaced by an URI that unambiguously refers to the Greek capital, e.g. <http://dbpedia.org/resource/Athens>).

5. KNOWLEDGE EXTRACTION & BUSINESS INTELLIGENCE

D2.2 indicated a large number of algorithms that range from basic statistics to deep learning algorithms, which have been considered for implementation over the AEGIS platform. After evaluating these algorithms, the needs of the demonstrators and the technical possibilities for the AEGIS platform, this section provides an overview of the analytics algorithms that are supported by AEGIS in the current version and which are based on the Spark MLlib (<https://spark.apache.org/mllib/>) library of algorithms for big data analysis.

Spark provides a powerful, unified engine that is both fast and easy to use. This allows data engineers and data scientists to solve their machine learning problems, as well as graph computation, streaming, and real-time interactive query processing. Spark provides different language choices, including Scala, Java, Python, and R.

Spark provides a general-purpose machine learning library, MLlib. It is designed for simplicity by providing data scientists coming from R and Python world with simple and familiar APIs. Also, scalability is provided natively by allowing developers to seamlessly run their ML code on a single machine or a big cluster without breaking down. Spark provides easy integration with other large variety of tools and languages. With the scalability, language compatibility, and speed of Spark, data scientists can solve and iterate through their data problems faster. The key benefit of MLlib is that it allows data scientists to focus on their data problems instead of solving the complexities surrounding distributed data such as infrastructure, and configurations. The data engineers can focus on distributed systems engineering while the data scientists can leverage the scale and speed of Spark.

Alternatively, data scientists use popular languages such as Python and R due to the large number of modules or packages that are readily available to help them solve their data problems. However, traditional uses of these tools are often limiting, as they process data on a single machine where the movement of data becomes time consuming, and moving from development to production environments requires extensive re-engineering. Using Spark, data scientists don't have to deal with the previously described problems. Moreover, there exist good integration support with notebook environments such as Zeppelin and Jupyter.

As visible also in the AEGIS platform, the algorithms have been placed overall in 5 algorithmic families which are the following:

- Dimensionality Reduction – Feature Extraction and Hypothesis testing
- NLP Functions
- Recommenders
- Clustering
- Classification/Regression

ALGORITHM	CATEGORIES
FEATURE EXTRACTION AND HYPOTHESIS TESTING -DIMENSIONALITY REDUCTION	
(1) Principal Component Analysis (PCA)	Feature extraction
(2) Pearson's chi-squared tests	Hypothesis testing
NATURAL LANGUAGE PROCESSING	

(3) Term frequency-inverse document frequency (TF-IDF)	Natural language processing
(4) Word2vec	Natural language processing
(5) Tokenizer	Natural language pre-processing
(6) N-gram	Natural language pre-processing
RECOMMENDATION SYSTEMS	
(7) Collaborative filtering (CF)	Recommendation systems
CLUSTERING METHODS	
(8) K-Means	Clustering
(9) Gaussian Mixtures	Clustering
(10) Latent Dirichlet Allocation (LDA)	Clustering
CLASSIFICATION-REGRESSION METHODS	
(11) Ordinary Least Squares (OLS)	Regression, Feature Selection
(12) Decision Trees (DT)	Classification, Regression
(13) Random Forest (RF)	Classification, Regression
(14) Multi-layer Perceptron (MLP)	Classification, Regression
(15) Naive Bayes (NB)	Classification
(16) One-vs-Rest classifier (a.k.a. One-vs-All)	Classification
(17) Isotonic regression	Regression
(18) Multinomial Logistic Regression	Classification, Regression

5.1. Dimensionality Reduction – Feature Extraction Hypothesis testing

5.1.1.1. Principal Component Analysis (PCA)

Brief description

Principal component analysis is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. It finds a rotation such that the first coordinate has the largest variance possible, and each succeeding coordinate in turn has the largest variance possible.

Purpose

Dimensionality reduction

Variables

- k: number of top features (int)

Typical Examples

Neuroscience, medicine, atmospheric science etc.

Application Examples (from the PSPS domain)

- In large datasets, it is very likely that subsets of variables are highly correlated with each other. The accuracy and reliability of a classification or prediction model will suffer if we include highly correlated variables or variables that are unrelated to the outcome of interest because of over fitting. To overcome this problem, PCA is used selecting a subset of variables together with a low complexity method for classification or regression. Because the subset size is predefined, the optimal number of selected variables is unknown. It can be fixed applying PCA many times to whole dataset and checking the accuracy of a classification or a regression method when employed with the resulted subset of variables.
- We suppose that the K-means algorithm described in a following paragraph is doing to implemented to cluster a data set obtained by smart home sensors. As known, K-means uses as similarity measure the Euclidean distance. However, in high-dimensional spaces, Euclidean distances tend to become inflated and the data points essentially become uniformly distant from each other. Due to this limitation, PCA is a useful relaxation of k-means clustering

Limitations

PCA is a linear algorithm. It will not be able to interpret complex polynomial relationship between features.

References

- Fukunaga, Keinosuke. 'Introduction to Statistical Pattern Recognition'. Elsevier. ISBN 0-12-269851-7 (1990)

5.1.1.2. Chi-Squared

Brief Description

The chi-square goodness of fit test is applied to binned data (data put into classes). The chi-square test statistic depends on how the data are binned and it requires a sufficient sample size in order for the chi-square approximation to be valid.

$$\chi^2 = \sum_{i=1}^k (O_i - E_i)^2 / E_i$$

After computing the chi value and comparing it with a table value we can decide whether to accept or reject a hypothesis.

Purpose

Hypothesis testing is a powerful tool in statistics to determine whether a result is statistically significant, whether this result occurred by chance or not. The chi-square test is used to test if a sample of data came from a population with a specific distribution.

Variables

- k: number of top features (int)

Typical Examples

Provided a hypothesis and result from an experiment, does the result suggest we should accept or reject initial hypothesis?

Application Examples (from the PSPS domain)

Say we are working on a problem that involves a hospital and the number of patients for each day of the week. We are provided with the distribution of patients for each of the week days (this is our hypothesis). From our measurements, however, we experience 200 patients within a particular week.

	M	T	W	T	F	S	S
Distribution (Hypothesis)	0.1	0.2	0.15	0.15	0.2	0.1	0.1
Experienced patients	20	40	40	30	20	30	20
Expected Patients (Hypothesis)	20	40	30	30	40	20	20

$$\chi^2 = \frac{(20 - 20)^2}{20} + \frac{(40 - 40)^2}{40} + \frac{(40 - 30)^2}{30} + \frac{(30 - 30)^2}{30} + \frac{(20 - 40)^2}{40} + \frac{(30 - 20)^2}{20} + \frac{(20 - 20)^2}{20}$$

$$\chi^2 = 18.(3)$$

Based on the computed chi value and a predetermined table, this goodness of fit test will tell us whether we should accept or reject our hypothesis based on the experienced data.

Established Variations

- Anderson-Darling Goodness of Fit
- Kolmogorov-Smirnov Test
- Shapiro-Wilk Normality Test
- Probability Plots
- Probability Plot Correlation Coefficient Plot

References

- Snedecor, George W. and Cochran, William G. (1989), Statistical Methods, Eighth Edition, Iowa State University Press.

5.2. NLP Function Algorithms

5.2.1.1. Term Frequency-Inverse Document Frequency (TF-IDF)

Brief Description

Term frequency-inverse document frequency numerical statistic is a weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection or corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus.

Purpose

Text mining, natural language processing

Variables

- K: number of features (int)
 - minDocFrequency: minimum number of documents in which a term should appear (int)
-

Typical Examples

Search engines

Application Examples (from the PSPS domain)

- Scoring and ranking a documents' relevance. Consider a document containing 100 words wherein the word cat appears 3 times. The term frequency (i.e., tf) for cat is then $(3 / 100) = 0.03$. Now, assume we have 10 million documents and the word cat appears in one thousand of these. Then, the inverse document frequency is calculated as $\log(10,000,000 / 1,000) = 4$. Thus, the Tf-idf weight is the product of these quantities: $0.03 * 4 = 0.12$.
- Assumed the problem of event detection from social media and we want to find specific events happened in one day. We can apply Tf-idf to several sites and compute the measure of a word for example 'fire' in several sites or in social media.

References

- Rajaraman, A.; Ullman, J. D. "Data Mining". pp. 1–17. ISBN 978-1-139-05845-2 (2011)

5.2.1.2. Word2vec

Brief Description

Word2vec is used to produce word embeddings. It is a two-layer neural networks that is trained to reconstruct linguistic contexts of words. Word2vec takes as its input a large corpus of text and produces a vector space, typically of several hundred dimensions, with each unique word in the corpus being assigned a corresponding vector in the space. Word vectors are positioned

in the vector space such that words that share common contexts in the corpus are located in close proximity to one another in the space.

Purpose

Text classification, natural language processing

Variables

- VectorSize: The size of the Vector (int)
- MinCount: The minimum count (int)
- numPartitions: The number of partitions to provide (int)
- maxSentenceLength: The maximum sentence length (int)
- maxIter: The maximum number of iterations to run (int)

Typical Examples

Natural language processing

Application Examples (from the PSPS domain)

- Let say we want to identify a word which doesn't belong in a list with several other words. As an example, in the list: ride, come, drive, run, crash, all the words except crash are verbs of transportation, so the answer would be crash
- Creating a model with word2vec using documents specified by insurance sector and million tweets, it can be improved the language used in advertisements or it can be created new offering strategies.
- Language modeling and feature learning techniques

References

- Mikolov, Tomas; Yih, Wen-tau; Zweig, Geoffrey. "Linguistic Regularities in Continuous Space Word Representations.". HLT-NAACL: pp. 746–751 (2013)

5.2.1.3. Tokenizer

Brief Description

The process of taking text (such as a sentence) and breaking it into individual terms (usually words) is called tokenization. As such, a Tokenizer is the instantiation of such a functionality to split text into different subsets.

In AEGIS we use RegexTokenizer that converts the input string to lowercase, removes stop words and then splits it by white spaces.

Purpose

The purpose of the Tokenizer is to extract the different words out of a text and to prepare the text for keyword extraction and identification in NLP applications

Variables

None

Typical Examples

Typical examples include text cleansing and sentiment analysis.

Application Examples (from the PSPS domain)

- The Tokenizer can be used to prepare input coming out of unstructured datasets (such as text) to conduct analysis and identification of keywords, which can be used for event detection.

Limitations

- Relying on simple heuristics and with tokenisation happening at word level, it is not always easy to define the meaning of a word.
- There are many edge cases such as contractions, hyphenated words, emoticons, and larger constructs such as URIs (which for some purposes may count as single tokens).

References

- Trim, C. (2013). The Art of Tokenization. Developer Works. IBM.

5.2.1.4. N-gram

Brief Description

An n-gram is a sequence of n tokens (typically words) for some integer n. This function can be used to transform input features into n-grams, taking as input a sequence of strings (e.g. the output of the Tokenizer) and the output will consist of a sequence of n-grams where each n-gram is represented by a space-delimited string of n consecutive words.

Purpose

Identification of most used items, word (keywords) and phrases, prediction of next word/phrases/item.

Variables

- n: number of Terms (int)

Typical Examples

Typical examples of n-grams can be met in speech recognition, computational linguistics, in DNA sequencing, etc.

Application Examples (from the PSPS domain)

- In the PSPS, n-grams can be used in NLP applications that are relevant to event and sentiment analysis detection and prediction. Moreover, those can be used to identify sensor reading outliers, working against prediction coming from n-grams.

Limitations

- Out of vocabulary words are ignored
- Long range dependency is not possible, as it the longest one is based on (n-1) tokens
- Practical but no complete linguistic knowledge modelling, as any Markov model

References

- A. Broder; S. Glassman M. Manasse; G. Zweig (1997). "Syntactic clustering of the web". Computer Networks and ISDN Systems. 29 (8): pp. 1157–1166.

5.3. Recommenders

5.3.1.1. Collaborative filtering

Brief description

Collaborative filtering (CF) is a technique used by recommender systems It filters information by using techniques involving collaboration among multiple agents, viewpoints, data sources or people. Most collaborative filtering systems apply the so-called neighbourhood-based technique. In this approach, a number of individuals is selected based on their similarity to an active individual. A recommendation for the active individual is made by calculating a weighted average of the decision of the selected individuals.

Purpose

Recommendation

Variables

- rank: rank of the matrix factorization (int)
- maxIter: max number of iterations (int)
- alpha: param for the alpha parameter in the implicit preference formulation (float)

Typical Examples

E-service personalization, e-commerce, e-learning, e-government etc

Application Examples (from the PSPS domain)

- Collaborative filtering can provide real time recommendations to drivers with the safest and faster routes. Using historical driving data with information come from social media and breaking news, the optimal route with controlled traffic in the safest area can be estimated.

- Insurance Product Recommender helps underwriters and brokers identify industry-specific client risks; pinpoint cross-selling and up-selling opportunities by offering access to collateral insurance products, marketing materials, and educational materials that support a complete sales cycle. As more products are sold to an ever-increasing customer base, the recommendations become more reliable, resulting in an exponential increase revenue realization.

Established Variations

Item-to-item approach is simply an inversion of the neighbourhood-based approach using the correlation the individual decisions.

Classification approach. The individuals are grouped using a classification method.

References

- F. Ricci, L. Rokach and Bracha Shapira, 'Introduction to Recommender Systems Handbook', Recommender Systems Handbook, Springer, 2011, pp. 1-35

5.4. Clustering Algorithms

5.4.1.1. K-Means

Brief Description

K-means is a popular algorithm for cluster analysis. It clusters the data points into predefined number of clusters, namely it constructs k clusters from N observations. The algorithm is commonly employed via an iterative refinement procedure and converge quickly to a local optimum.

Purpose

Clustering

Variables

- k : number of clusters (int)
- maxIterations : Max number of iterations (int)

Typical Examples

Entities recognition, grouping events, etc.

Application Examples (from the PSPS domain)

Can be applied to any scenario requires typical clustering results. For instance, Demonstrator 1 can describe the grouping of data into a number of driving styles, whereas Demonstrator Scenario 3 may require a grouping and categorization of customer habits, which would eventually lead to personalized offers.

Established Variations

Fuzzy C-means is a popular variation where each data point belongs to more than one cluster to a fuzzy degree.

Hierarchical variations of k-means attempt to determine the optimum number of clusters automatically starting with a small number and then adding or splitting clusters according to the method's logic.

Limitations

- *k-means converges when the clusters has comparable spatial extent.*
- *k-means does not perform well in high dimensional datasets*

References

- E.W. Forgy (1965). "Cluster analysis of multivariate data: efficiency versus interpretability of classifications". *Biometrics*. **21**: 768–769.
- MacQueen, J. B. (1967). Some Methods for classification and Analysis of Multivariate Observations. Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1. University of California Press. pp. 281–297.

5.4.1.2. Gaussian Mixtures

Brief Description

A Gaussian mixture is a probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. One can think of mixture models as generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians.

Purpose

Clustering

Variables

- *k*: number of clusters (int)
- *maxIterations*: Max number of iterations (int)

Typical Examples

Image segmentation, text processing, handwriting recognition, etc.

Application Examples (from the PSPS domain)

- Same as k-means

Limitations

Insufficient input points lead to algorithm divergence.

References

- McLachlan, G.J. (1988), "Mixture Models: inference and applications to clustering", Statistics: Textbooks and Monographs, Dekker

5.4.1.3. Latent Dirichlet Allocation (LDA)

Brief Description

Latent Dirichlet Allocation (LDA) is an example of a topic model and is included amongst the NLP algorithms for clustering similar data. Its generative statistical model views each document as a mixture of various topics and allows words to be assigned to each topic in a probabilistic manner.

Purpose

LDA is used in topic modelling

Variables

- k: number of clusters (int)
- maxIterations: Max number of iterations (int)

Typical Examples

Information retrieval, image clustering

Application Examples (from the PSPS domain)

Can be applied to any Demonstrator requires topic modelling coming from textual data, such as any Event detection scenarios.

Limitations

- The ordering of words (context) is not taken into account. Each word is treated individually.
- Can be a slow process when the number of documents is large

References

- Blei, D; Ng, A; Jordan, M. (2003). Lafferty, John, ed. "Latent Dirichlet Allocation". Journal of Machine Learning Research. 3 (4–5): pp. 993–1022.

5.5. Classification/Regression Algorithms

5.5.1.1. Ordinary Least Squares

Brief Description

Ordinary least squares is the most common formulation for regression problems. It minimizes the residual sum of squares between the observations in a dataset, and the model responses established the linear combinations between the input data and their corresponding outputs. Mathematically it solves a problem of the form:

$$\min_w ||Xw - y||_2^2$$

The coefficients w are estimated with the singular value decomposition method or with the stochastic gradient descent algorithm that is more suitable for a large scale problem.

The performance of ordinary least squares relies on the independence of the input variables. When the inputs are correlated and have an approximate linear dependence, the least-squares estimate becomes highly sensitive to random noise in the observed response, producing a large variance. To address some of the problems of ordinary least squares generalization methods can be applied.

Purpose

Regression, Feature Selection

Variables

- regParam: regularization parameter (float)
- maxIterations: Max number of iterations (int)
- elasticNetParam: Param for the ElasticNet mixing parameter(float)

Typical Examples

Analysis of variance, measuring accuracy, financial forecasting, etc

Application Examples (from the PSPS domain)

- Assumed that someone need to predict the CO2 and the only available data are the temperature timeseries. He would like to know if this problem could be solved using only temperature as the only input. He should choose a simple method running in short time. He could apply linear regression together with a regularization method to figure out if the predictability of this problem is acceptable.
- An insurance company tries to predict loss amounts based on the variables obtained by its policyholders' profiles. Because the dataset is high dimensional, it should be applied a method to describe how correlate each variable to the loss amounts. The Lasso regression method can give this information providing a set of weights which indicate the importance of each variable.

Established Variations

Ridge regression addresses some of the problems of ordinary least squares by imposing a penalty on the size of coefficients. The ridge coefficients minimize a penalized residual sum of squares.

Lasso reduces the number of variables upon which the given solution is dependent. For this reason, the Lasso and its variants are fundamental to the field of feature selection. Under certain conditions, it can recover the exact set of non-zero weights. Since reducing parameters to zero removes them from the model.

ElasticNet is a linear regression model combining ridge regression with lasso. This combination allows for learning a sparse model where few of the weights are non-zero like Lasso, while still maintaining the regularization properties of Ridge. *ElasticNet* is useful when there are multiple features which are correlated with one another. Lasso is likely to pick one of these at random, while elastic-net is likely to pick both.

Limitations

It performs well for feature selection, but it is not recommended for estimation. The input data should be normalized.

References

- Björck, Å. Numerical methods for least squares problems. Philadelphia: SIAM, 1996. ISBN 0-89871-360-9.

5.5.1.2. Decision Trees (Regression and Classification)

Brief description

Decision tree is built incrementally breaking down a dataset into smaller and smaller subsets that contain instances with similar values (homogenous). The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. A leaf node represents the decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data. The core algorithm for building decision trees called ID3 by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 uses Entropy and Information Gain to construct a decision tree. ID3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one. The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches).

Purpose

Classification, Regression

Variables

- **maxDepth:** Maximum depth of the tree (int)
- **maxBins:** Maximum number of bins used for discretizing continuous features
- **minInstancesPerNode:** Minimum number of samples each child node must have after split (int)

- minInfoGain: Minimum information gain for a split to be considered per node (float)

Typical Examples

Agriculture, manufacturing and production, astronomy etc.

Application Examples (from the PSPS domain)

- Due to their rule-based architecture, decision trees can detect efficiently events from data without noise. Using streaming data from a vehicle such as X-Y-Z acceleration, speed, and location, a decision tree can detect road damages.
- Similarly, with the above example, decision trees could be implemented to create alerts for an event. After processing messages from social media and news using a natural language processing method like word2vec, decision trees can establish the rules for an event detection

Limitations

Low generalization ability

References

- L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth, Belmont, CA, 1984

5.5.1.3. Multi-Layer Perceptron

Brief Description

The multi-layer Perceptron (MLP) is a popular category of feedforward neural networks able to solve non-linear problems. It consists of more than one layer of nodes, (in contrast to the single layer perceptron) and utilizes a supervised learning technique called backpropagation for training.

MLPs were very popular during the '80s especially in the fields of speech processing and image recognition, but this interest degraded gradually due to the appearance of faster and simpler algorithms (e.g. SVNs). Today, the enthusiasm towards multi-layer networks has returned as an outcome of the success of *deep learning*.

Purpose

Classification, Regression, Deep learning

Variables

- maxIterations: maximum number of iterations (int)
- step: Step size to be used for each iteration of optimization(float)
- convergenceTolerance: the convergence tolerance for iterative algorithms (float)
- layers: all layers' sizes (array)
- seed: param for random seed (long)

Typical Examples

Function approximation, image recognition, speech processing, etc.

Application Examples (from the PSPS domain)

- MLPs and their variations should be assigned with tasks suitable for deep learning, such as the road damage classification task in Demonstrator 1, the discrimination of driving styles in *safe* or *unsafe* manner, or the recognition of the alerting conditions in Demonstrator Scenario 2.

Established Variations

Convolutional Neural Networks (CNNs) is a class of deep, feed-forward neural networks, whose design emulates the vision processing in living organisms. They consist of an input and an output layer, with multiple hidden layers which are either convolutional, pooling or fully connected. They have wide applications in speech, image and video processing, as well as in recommenders and NLP.

Limitations

Problem of overfitting or underfitting the data

References

- Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961
- Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundation. MIT Press, 1986

5.5.1.4. Naive Bayes

Brief Description

Naïve Bayes (NB) is a simple multiclass classification algorithm with the assumption of independence between every pair of features. Naive Bayes can be trained very efficiently. Within a single pass to the training data, it computes the conditional probability distribution of each feature given label, and then it applies Bayes' theorem to compute the conditional probability distribution of label given an observation and use it for prediction.

Purpose

Classification

Variables

- Lambda: Smoothing parameter (float)

Typical Examples

Spam filtering, Image recognition etc.

Application Examples (from the PSPS domain)

- Obtained numerical weather prediction by a local meteorological office, the day-ahead weather conditions can be described. Usually numerical weather predictions consist of many variables. Naïve Bayes can be employed to classify the numerical weather predictions and to contribute to form recommendations about the outdoor conditions of the next day (Scenario 2.1)
- Naïve Bayes can be used for traffic incident detection such as accidents, disabled vehicles, spilled loads, temporary maintenance and construction activities, signal and detector malfunctions, and other special and unusual events that disrupt the normal flow of traffic and cause motorist delay. Using social media, smart phones devices and vehicles black boxes GPS data naïve Bayes can estimate various incidents that have recorded.
- Using several features recorded in real time by wearables, naïve Bayes can be implemented for emotional recognition. The user can record his emotions once per day for one-month period. The data gathered by its wearable is harmonized with the recorded emotions. Then naïve Bayes learn this data set and inform the user about his emotions.
- Using positioning information obtained from mobile phones or wearable devices and recorded behavioural routines, naïve Bayes can identify irregularity patterns. A large dataset with all above information can be created. Due to every person acts differently, a dimensionality reduction or a feature selection method needs to be applied. In follows, naïve Bayes algorithm can find the conditions where a patient acts irregular.

Established Variations

- *Gaussian Naive Bayes* is employed when the likelihood of the features is assumed to be Gaussian.
- *Multinomial Naive Bayes* implements the naive Bayes algorithm for multinomially distributed data.
- *Bernoulli Naive Bayes* is applied to data that is distributed according to multivariate Bernoulli distributions; i.e., there may be multiple features but each one is assumed to be a binary-valued variable. Therefore, this class requires samples to be represented as binary-valued feature vectors.
- *Compliment Naive Bayes* estimates the parameters from all classes except the one which it is going to be evaluated.

References

- Hastie, Trevor, Robert Tibshirani, and J Jerome H Friedman. The Elements of Statistical Learning. Vol.1. N.p., Springer New York, 2001

5.5.1.5. Random Forest (RF)

Brief Description

Random forests (RF) are ensembles of decision trees. RFs are one of the most successful machine learning models for classification and regression. They combine many decision trees in order to reduce the risk of overfitting. Like decision trees, random forests handle categorical features, extend to the multiclass classification setting, do not require feature scaling, and are able to capture non-linearities and feature interactions. Random forests train a set of decision trees separately, so the training can be done in parallel. Each decision tree is constructed subsampling the original dataset (bootstrapping) and considering different random subsets of features to split. Each feature is analysed using the most discriminative thresholds that increase the information gain. Combining the predictions from each tree reduces the variance of the predictions, improving the performance on test data.

Purpose

Classification, Regression

Variables

- numTrees: The number of decision trees to be used in training (int)
- maxDepth: The maximum depth of the trees (int)

Typical Examples

Medical applications, computer vision, forecasting, fault detection etc.

Application Examples (from the PSPS domain)

- The decision tree drawbacks related to the generalization ability is solved using random forests. Random forests can analyse noisy data and perform classification with great accuracy. Using vehicle data, random forests can estimate damages on a road. Random forests create trees selecting features based on the variable importance, a metric that indicates the potential of each variable.
- Random forests can effectively balance the indoor ambient conditions in a smart house. It can manipulate sensor data, device measurements (CO₂, VOC) and energy pricing data to ensure optimal comfort levels and compliance with health requirements.
- As support vector machines, random forests can detect accurately a distress situation, a cognitive deterioration and frailty status in a smart home environment. Applying data from sensors, smart phone and wearables to random forests, daily living activities can be supervised, and emerging incidents can be identified.

Established Variations

Extremely Randomized Trees uses random thresholds for each candidate feature and the best of these randomly-generated thresholds is picked as the splitting rule.

Gradient Tree Boosting follows a gradient descent like procedure to minimize a differential loss function by adding new weak learners (decision trees).

Limitations

- Gradient Tree Boosting can train one tree at a time, so they can take longer to train than random forests. Random Forests can train multiple trees in parallel.
- Training more trees in a Random Forest reduces the likelihood of overfitting while training more trees with Gradient Tree Boosting increases the likelihood of overfitting.

References

- Hastie, T.; Tibshirani, R.; Friedman, J. H. (2009). "The Elements of Statistical Learning", New York: Springer. pp. 337–384. ISBN 0-387-84857-6.

5.5.1.6. One-vs-Rest Classifier (a.k.a. One-vs-All)

Brief Description

This algorithm involves training a single classifier per class, with the samples of that class as positive samples and all other samples as negatives. This requires base classifiers to produce a real-valued confidence score for its decision, rather than just a class label. In addition to its computational efficiency, a big advantage of One-vs-Rest is its interpretability, since each class is represented by one and one classifier only, it is possible to gain knowledge about the class by inspecting its corresponding classifier.

Purpose

Classification

Variables

- maxIterations: Max number of iterations (int)
- convergenceTolerance: the convergence tolerance for iterative algorithms (float)

Typical Examples

Image classification

Application Examples (from the PSPS domain)

- This algorithm can be applied to any classification problem in the Demonstrator scenarios where the number of classes is relatively small, and the accuracy is not the primary goal.

Limitations

The scale of the confidence values may differ between the binary classifiers.

Also, if the class distribution is balanced in the training set, the binary classification learners see unbalanced distributions.

It is a reasonable approach to use only when the total number of classes is small.

References

- Nasrabadi, N. M. (2007). Pattern recognition and machine learning. *Journal of electronic imaging*, 16(4), 049901.

5.5.1.7. Isotonic Regression

Brief Description

Isotonic regression (often encountered as monotonic regression also), finds a non-decreasing approximation of a function while minimizing the mean squared error on the training data. Essentially the algorithm tries to fit a free-form line to a sequence of observations considering the following: the free-form line has to be non-decreasing everywhere and it needs to lie as close to observation as possible. The benefit of such a model is that it does not assume any form for the target function such as linearity.

Purpose

The benefit of such a model is that it does not assume any form for the target function such as linearity and is free of any tuning parameters.

Variables

- None

Typical Examples

Classifier and Recommendation models calibration

Application Examples (from the PSPS domain)

- It can be employed to calibrate the recommendation systems for both Driving routes and Insurance products recommendation scenarios.
- It can be utilized as a simple and fast approach for regression tasks or for the calibration of classifiers' output for categorical probabilities.

Limitations

- One or several points at the ends of the interval are sometimes noisy and result in low quality results
- Works better than other approaches when there is big enough statistics ($n \gtrsim 103$)

References

- Kruskal, J. B. (1964). "Nonmetric Multidimensional Scaling: A numerical method". *Psychometrika*. 29 (2): 115–129. doi:10.1007/BF02289694.

- Leeuw, Jan de; Hornik, Kurt; Mair, Patrick (2009). "Isotone Optimization in R: Pool-Adjacent-Violators Algorithm (PAVA) and Active Set Methods". Journal of Statistical Software. 32 (5): 1–24. doi:10.18637/jss.v032.i05. ISSN 1548-7660.
- Best, M.J.; Chakravarti N. (1990). "Active set algorithms for isotonic regression; a unifying framework". Mathematical Programming. 47: 425–439.

5.5.1.8. Multinomial Logistic Regression

Brief Description

MLR (known by a variety of other names, including polytomous LR multiclass LR, softmax regression, etc.) is a classifier that generalizes logistic regression to multiclass problems, i.e. with more than two possible discrete outcomes. As such, MLR is an extension of logistic regression, which analyses dichotomous (binary) dependents. In more detail MLR is a linear regression analysis that is used when the dependent variable is nominal with more than two levels.

Purpose

Like all linear regressions, the multinomial regression is a predictive analysis. Multinomial regression is used to describe data and to explain the relationship between one dependent nominal variable and one or more continuous-level (interval or ratio scale) independent variables.

The fundamental condition, in multinomial logistic regression is that the outcome variable is categorical (e.g.- vanilla, strawberry, chocolate) and the predictor variables tend to a linear relationship. Predictive variables (also called independent variables) can be either categorical or continuous (numerical).

Variables

- MaxIter: Maximum Number of Iterations
- RegParam: Regularization Parameter
- ElasticNetParam: Elastic net Parameter

Typical Examples

Typically used for Risk Analysis predictions, in the fields of marketing (campaign performance), banking (credit risk analysis), manufacturing and politics.

MLR is used as an alternative to Naive Bayes in NLP applications, though the models are trained slower.

Application Examples (from the PSPS domain)

- MLR is used for predicting probabilities of the different possible outcomes (more than 2 states) of a categorically distributed dependent variable. An example of this can be the suggestions offered in Demonstrator 2 for a Smart Home's comfort levels or the various

recommended treatments based on health monitoring, where the influence(probability) of each outcome is important.

References

- Greene, W (2012). *Econometric Analysis* (Seventh ed.). Boston: Pearson Education. pp. 803-806. ISBN 978-0-273-75356-8.
- Engel, J. (1988). "Polytomous logistic regression". *Statistica Neerlandica*. 42 (4): 233.

6. VISUALISATION

Due to the progress in computational power and storage capacity in existing platforms, as well as the emergence of new platforms that generate new data over the last decades, nowadays data is produced at an incredible rate and in unprecedented volume. The flood of data resulted in the arising need from analysts and decision makers from several industry sectors for tools that will help them organise their data, generate overviews and explore the information space towards the aim of rapidly extracting useful information. The rise of advanced analytics was a consequence of this need and several powerful analytics software tools were implemented to fill in this gap. These tools are able to run complex algorithms on large datasets and perform a variety of analyses depending on the user needs and requirements.

Nevertheless, the ability to provide useful information in an easily and quickly perceivable manner from a conducted analysis heavily relies on the data visualisation. In general, data visualisation is the presentation of the data in a pictorial or graphical format. Visualisation is aiming at bringing the gap of extracting useful information from the results of an analysis or the dataset itself by employing more intelligent means in the established analysis process. The basic idea of the visualisation is to visually represent the information, thus allowing the human to directly interact with it, gain insight, draw conclusions and make better decisions. It enables decision makers and analysts to see analytics visually represented facilitating the grasp of the difficult concepts or new pattern identification. Through visualisation, the process of analysing big data sets is simplified and knowledge extraction and business intelligence are accessed more easily through eye-catching and easy-to-understand formats. With the use of visualisations, the complex cognitive work needed is significantly reduced and the information contained on the datasets can be synthesized so that useful insights are derived from massive, dynamic and heterogeneous data. Moreover, the interactive visualisations can take the analysis one step further as they can offer the possibility to display more real-time info about the specific points or areas currently explored, plus possibility of highlighting/hiding specific content changing interactively what data you see and how it's processed.

An in-depth analysis for the goals of visualisation in the data analysis, as well the corresponding requirements, were documented in detail in deliverable D2.2. The forthcoming paragraphs are focusing on the tools and libraries that were incorporated in the AEGIS platform towards the aim of enabling the visualisation capabilities of the platform, the visualisation types offered by the platform and the relevance of these visualisations to the AEGIS demonstrators.

6.1. Tools and libraries

Within the context of AEGIS, the consortium further analysed the designed conceptual architecture, as well as the core and the demonstrator requirements, and made the necessary refinements towards the aim of providing an updated architecture that will further ensure that the designed integrated platform is addressing all the requirements and will provide the envisioned added value services to the AEGIS platform's stakeholders. These decisions have driven also the specifications in terms of utilisation and integration of several tools and libraries in regards to the visualisation capabilities of the platform. As such, the selection of these tools and libraries is highly associated with their capability of offering a variety of visualisation formats with several layers of information and customisation. The following tools and libraries were selected as the ones that will be integrated and exploited in order to address the advanced visualisations requirements of the AEGIS platform.

6.1.1. *Jupyter*⁵

The Jupyter Notebook is an open-source web application that is supporting the creation and sharing of documents that contain live code, equations, visualisations and narrative text. Jupyter provides capabilities for data cleaning and transformation, numerical simulation and statistical modelling, data visualisation and machine learning among others. It currently supports more than 40 programming languages including Python, R, Scala and Julia. Additionally, it offers integration with big data frameworks in order to leverage of their capabilities such as Apache Spark. Jupyter incorporates a large variety of features, such as the in-browser editing for code or rich text with automatic syntax highlighting and indentation, as well as the ability to execute code from the browser with the results of computations attached to the code that generated them. Moreover, Jupyter can produce rich interactive outputs using rich media representations such as HTML, PNG, SVG, LaTeX and custom MIME types. One major feature of the Jupyter notebook is the ability to display plots that are the output of running code cells. Additionally, the majority of the Python plotting libraries can be easily embedded within Jupyter to enable interactive visualisations where real-time information is displayed about the specific points or areas currently explored, in addition to the possibility of highlighting/hiding specific content. One more advantage of Jupyter is the embracement of web technology and the support for JavaScript, which further expands the offered capabilities in terms of user interface development within the notebooks.

6.1.2. *Highcharts*^{6,7}

Highcharts is an SVG-based, multiplatform charting library written in pure JavaScript, offering an easy way of adding interactive, mobile-optimised charts to a web site or web application. Highcharts is backend-agnostic as it can be used with any back-end database or server stack where data can be provided in any form, loaded or updated live, and offers wrappers for most popular programming languages such as Python, R and Java. Highcharts is also big data ready, offering an optimised engine to support the rendering of millions of data points in the browser. Highcharts currently supports line, spline, area, areaspline, column, bar, pie, scatter, angular gauges, arearange, areasplinerange, columnrange, bubble, box plot, error bars, funnel, waterfall and polar chart types. Highcharts is solely based on native browser technologies and does not require client-side plugins like Flash or Java. Through a full API, the user can add, remove or modify series and points or modify axes at any time after chart creation. Numerous events supply hooks for programming against the chart. This opens for solutions like live charts constantly updating with values from the server, user supplied data and more.

6.1.3. *Folium*⁸

Folium is a Python wrapper for the leading open source JavaScript library for mobile-friendly interactive maps called leaflet.js⁹. The leaflet.js library is designed based on the principles of simplicity, performance and usability. It provides compatibility with all desktop and mobile platforms available and it is extendable with a large variety of plugins. Folium is combining the

⁵ <http://jupyter.org/>

⁶ <https://www.highcharts.com/products/highcharts/>

⁷ <https://github.com/highcharts/highcharts>

⁸ <http://python-visualization.github.io/folium/docs-v0.5.0/>

⁹ <https://leafletjs.com/>

powerful strengths of the Python programming language in terms of data wrangling with the mapping strengths of the leaflet.js library. With Folium the data can be easily processed and transformed leveraging Python's data manipulation capabilities and then can be visualised on an interactive leaflet map. Folium enables both the binding of data to a map in order to provide choropleth visualisations, as well as providing rich vector/raster/HTML visualisations as markers on the map. The library is offering a large number of built-in tilesets from OpenStreetMap, Mapbox and Stamen, while also supporting custom tilesets with Mapbox or Cloudmade API keys.

6.2. Visualisations techniques in AEGIS

Within the context of AEGIS, the consortium identified the need to offer a large variety of visualisations techniques in order to successfully address the AEGIS platform stakeholders' needs. After carefully analysing the core requirements, as well as the requirements originating from the demonstrators of the platform, the list of the visualisation techniques that will be offered by the AEGIS platform was compiled. Going beyond simple standard static charts, this list also includes interactive charts that enable users to manipulate them or drill into the data for querying and analysis.

In deliverable D2.2, an indicative non-exhaustive list of the visualisation techniques that could be deployed in the context of the AEGIS was presented. This list was evaluated from the consortium and in conjunction with the feedback received by the AEGIS demonstrators, as well as the in-depth analysis performed on the needs of PSPS domain concerning visualisations, the visualisation techniques that will be utilised and offered by the AEGIS platform were selected. The list of visualisation techniques that are employed in the context of AEGIS includes the following:

- Scatter plot
- Pie chart
- Bar chart
- Line chart
- Box plot
- Histogram
- Time series
- Heatmap
- Bubble chart
- Map

The details and characteristics for each of the aforementioned visualisation techniques were documented in detail in Section 6.3 of the deliverable D2.2. The following figures illustrate indicative examples of these visualisation techniques as they are implemented in the AEGIS platform.

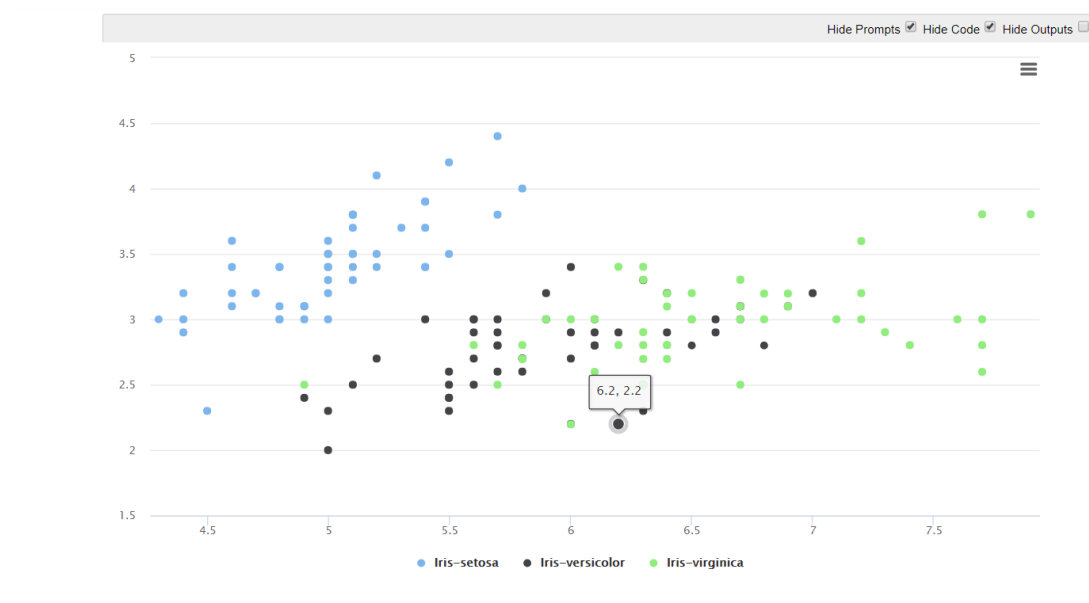


Figure 5: Example of a scatter plot in AEGIS

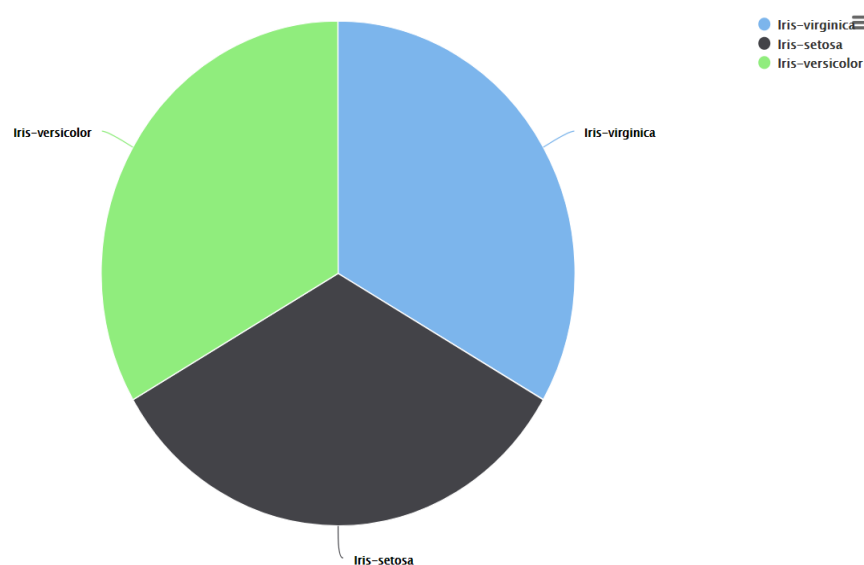
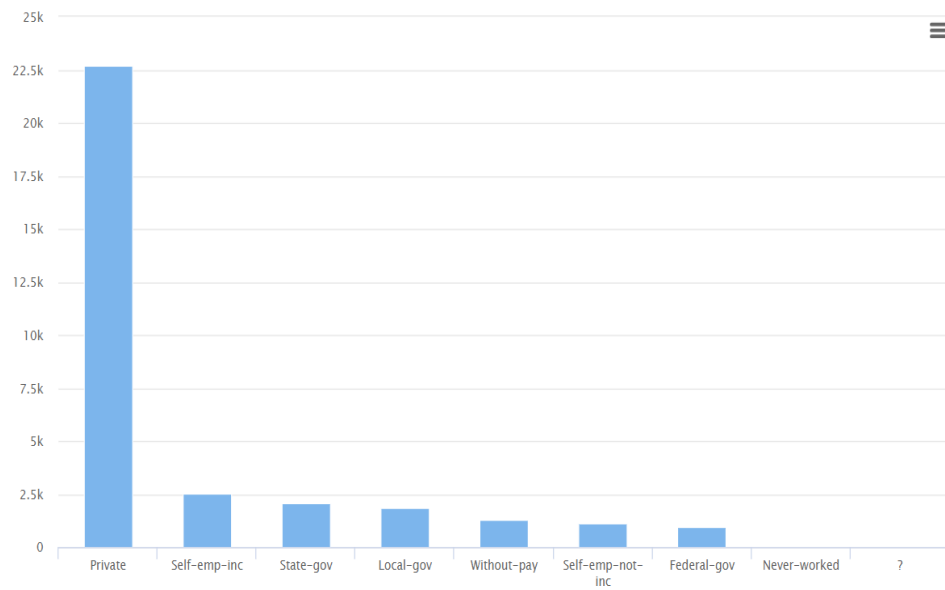
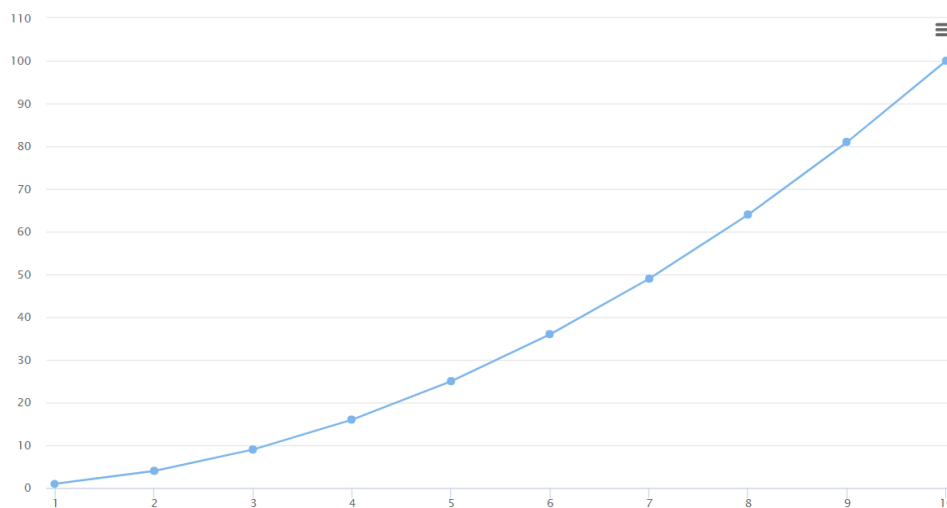


Figure 6: Example of pie chart in AEGIS

**Figure 7: Example of bar chart in AEGIS****Figure 8: Example of line chart in AEGIS**

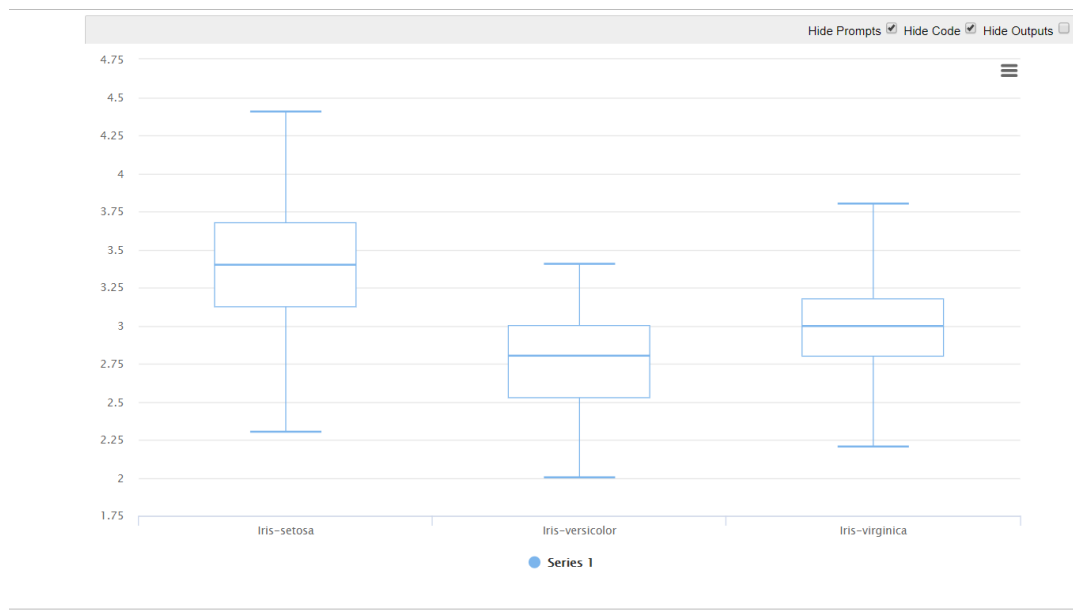
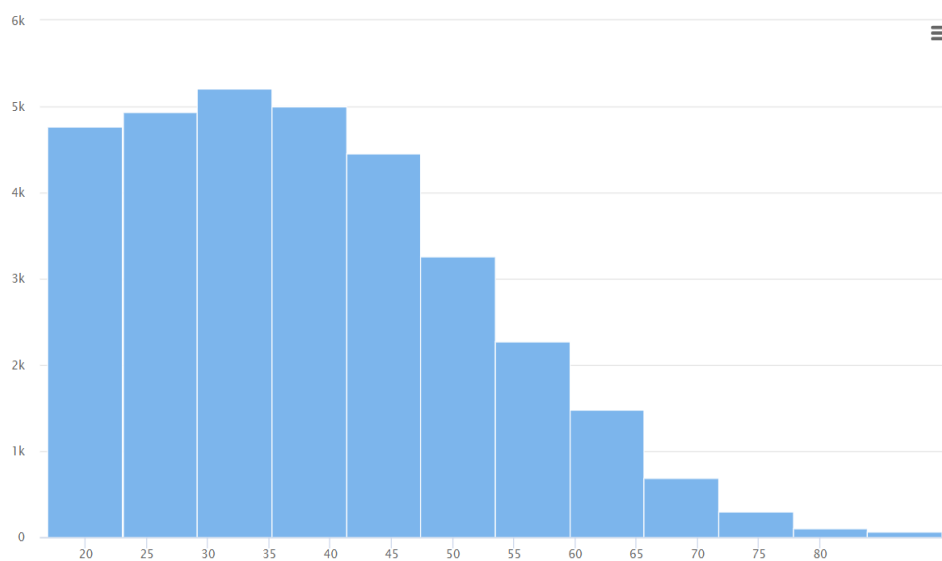
**Figure 9: Example of a box plot in AEGIS****Figure 10: Example of histogram in AEGIS**



Figure 11: Example of time series in AEGIS

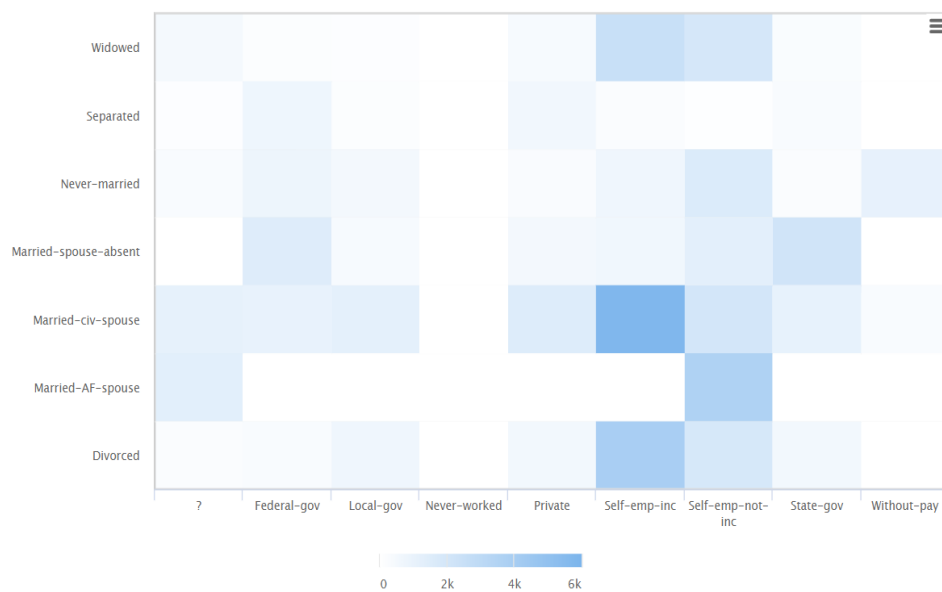


Figure 12: Example of heatmap chart in AEGIS

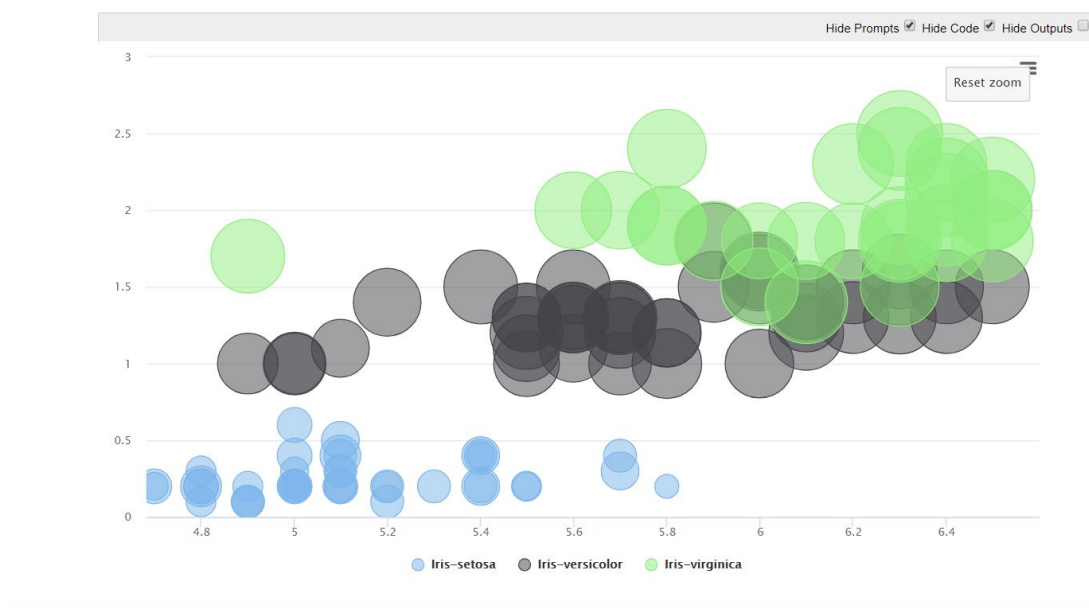


Figure 13: Example of a bubble chart in AEGIS

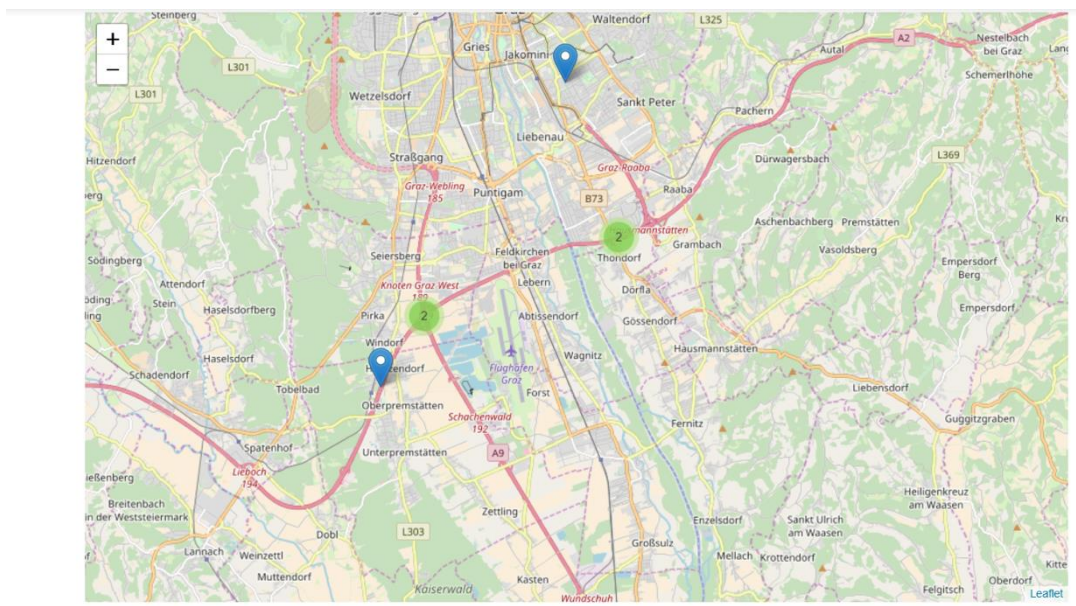


Figure 14: Example of plotting on a map in AEGIS

It should be noted at this point that the list presented is not the final list of visualisations that will be supported by the platform as it is foreseen that the list will be further expanded with additional visualisation techniques as the project evolves and additional feedback will be received especially from the AEGIS demonstrators.

6.3. Visualisation techniques relevance to the AEGIS Demonstrators

The selection of the visualisation techniques that are employed in the AEGIS platform was based, among other criteria, on the successful fulfilment of the requirements originating from the demonstrators of the platform. In the current subsection a mapping is performed between

the visualisation techniques, the relevant demonstrators and aspired usage of the visualisation technique by the demonstrator. In the first row of the table, the list of the visualisation techniques that will be available in the context of the AEGIS platform is represented: the middle row documents the demonstrator that will utilise the specific visualisation technique, while the third row describes how the visualisation technique will be used from the relevant demonstrator(s). Ultimately, the visualisation techniques offered by the AEGIS platform will be able to support additional or new services for both the demonstrators and the stakeholders of the platform.

Visualisation Techniques	Demonstrator Relevance	Usage
Scatter plot	<ul style="list-style-type: none"> • Smart Home and Assisted Living demonstrator • Insurance demonstrator 	<ul style="list-style-type: none"> • Visualise correlations between the various collected measurements • Visualise the results of the clustering analysis performed
Pie chart	<ul style="list-style-type: none"> • Smart Home and Assisted Living demonstrator • Insurance demonstrator 	<ul style="list-style-type: none"> • Visualise proportional distribution and percentages of the collected measurements • Visualise business and market analysis
Bar chart	<ul style="list-style-type: none"> • Insurance demonstrator 	<ul style="list-style-type: none"> • Visualise the trend of some parameters of interest for the company in diverse years/places
Line chart	<ul style="list-style-type: none"> • Insurance demonstrator 	<ul style="list-style-type: none"> • Visualise the trend of some parameters of interest for the company in diverse years/places • Visualise business and market analysis
Box plot	<ul style="list-style-type: none"> • Smart Home and Assisted Living demonstrator 	<ul style="list-style-type: none"> • Visualise the outliers in the collected measurements
Histogram	<ul style="list-style-type: none"> • Smart Home and Assisted Living demonstrator • Insurance demonstrator 	<ul style="list-style-type: none"> • Visualise the distribution of values for the collected measurements over predefined intervals • Visualise business and market analysis
Timeseries	<ul style="list-style-type: none"> • Automotive demonstrator • Smart Home and Assisted Living demonstrator 	<ul style="list-style-type: none"> • Visualise patterns or behaviour of vehicle /driving data over time • Visualise patterns or behaviour in the

		collected measurements data over time
Heatmap	<ul style="list-style-type: none"> Automotive demonstrator Insurance demonstrator 	<ul style="list-style-type: none"> Visualise variance of vehicle /driving data and reveal any patterns Visualise customer density policy per type
Bubble chart	<ul style="list-style-type: none"> Automotive demonstrator Smart Home and Assisted Living demonstrator Insurance demonstrator 	<ul style="list-style-type: none"> Visualise the correlations between the vehicle /driving data Visualise the correlations between the various collected measurements Visualise the clustering analysis performed
Map	<ul style="list-style-type: none"> Automotive demonstrator Insurance demonstrator 	<ul style="list-style-type: none"> Display the results of the analysis (events, patterns) on a map Display the customers based on their location on the map and risk-based maps

6.4. AEGIS Requirements met

In deliverable D2.2 a mapping between the requirements regarding visualisation, and the core and demonstrator, functional, non-functional and technical requirements, as identified and documented in deliverable D3.1, was performed. The current subsection provides the updated information in regards to the aforementioned mapping.

The first row of the table represents the unique code of the visualisation requirement, the middle row provides the description of the requirements formulated based upon the description of the requirements documented in D3.1, updated based upon the goals and needs for visualisations as analysed in the current section, while the third row maps the specific requirement to the set of requirements from D3.1.

ID	Visualisation Requirements	Previous Requirement of Reference
VC_R1	Feature a user-friendly interface which provides an overview of supported kind of visualizations	NFR7
VC_R2	Act as a unified front end to multiple databases and (big) data types. The visualisation component should be able to handle (visualise / create advanced graphs) large datasets, queries spanning multiple datasets, and scale horizontally	TR14 (NFR5), TR41 (CFR9, NFR1), TR55 (CFR5, CFR9, CFR18)

VC_R3	Preview a small selection of the results of the generated query so as to extract some initial insights out of the foreseen analytics	TR53 (CFR18, NFR3)
VC_R4	Update the visualisation by re-running the query contained in it	FR_RT12
VC_R5	Automatically update the visualisation if a data source is modified, without the need to re-run the query <i>Note: The realisation of this requirement might involve multiple elements of the AEGIS platform besides the ones offering visualisation capabilities</i>	FR_RT14
VC_R6	Render data in as many ways as possible / support different kinds of visualisations, based on different types of input datasets formats. Provide means for visualizing different data modalities (e. g. special, temporal, statistical) and provide an overview of the supported kinds of visualization	FR_RT1, TR54 (CFR5, CFR9, CFR18)
VC_R7	Save the results of the visualisation in the form of various graphic formats (such as PNG, JPEG PDF, SVG)	FR_RT2
VC_R8	Export the results of the analysis and visualisations, so that they can be in turn consumed by external applications and/or entities (in the form of various graphic formats such as PNG, JPEG PDF, SVG)	FR_RT3

7. DATA POLICY AND BUSINESS BROKERAGE FRAMEWORKS

7.1. Data Policy Framework updates

Considering now the state-of-the art approaches, challenges, and requirements for big data with regard to *Intellectual Property Rights (IPR)*, *trust*, *security*, and *quality* defined in the deliverable D2.1 (sections 4.1.1.1, 4.1.1.3 and 4.1.1.4), in this Section we discuss the AEGIS Data Policy Framework (DPF), which aims at facilitating the PSPS stakeholders and users to answer questions like:

- Do we have the right to use a specific dataset, algorithm or data-as-a- service?
- Do the qualities of the data asset meet the qualities mentioned in the agreement between the data asset provider and interested user/consumer?
- Do we have the right to republish an intelligence report built on a data asset or a collection of data assets?

The AEGIS DPF is a conceptual realisation of the approach of the project towards categorising assets which are placed on the AEGIS platform. Most of those assets refer to datasets (hence the title of “Data Policy Framework”) and the main asset type in AEGIS is defined as AEGIS::data, yet for reason of completeness other types of assets are defined, such as algorithms, intelligence reports, visualisations, etc. which are outputs of work which can be performed with the aid of the overall AEGIS infrastructure and that can be shared over the platform as well by their owners.

The DPF, as identified in D2.1, is an extensible model that is used to describe certain properties of assets, which are deemed necessary for following an approach that allows monetisation based on offered assets (by their owners/producers). Accordingly, on the one hand, the DPF enables the value creation capacity of AEGIS, by keeping its non-profit nature; on the other hand, it promotes external transactions which lead to value generation and value capture within the ecosystem of collaborating parties.

The proposed AEGIS DPF provides a complete set of the asset policies, where some of which are to be used by the brokerage engine and the harvester of the AEGIS platform, towards demonstrating how such assets can be described to cater for trusted, transparent and easy to follow business transactions between different organisations in a value chain model.

As shown in the next figure, an Asset complies with a specific Asset Policy that governs every contract / Transaction among an Asset Provider and Asset Consumer. In this context, an Asset Policy in AEGIS thus aims at:

- Defining the detailed terms according to which an asset can be used, on the basis that any use outside the policy terms would constitute an infringement.
- Specifying the expected asset quality, as well as the delivery and payment terms.
- Clarifying the liability of asset providers and consumers (e.g. in case of failure of the provided asset).

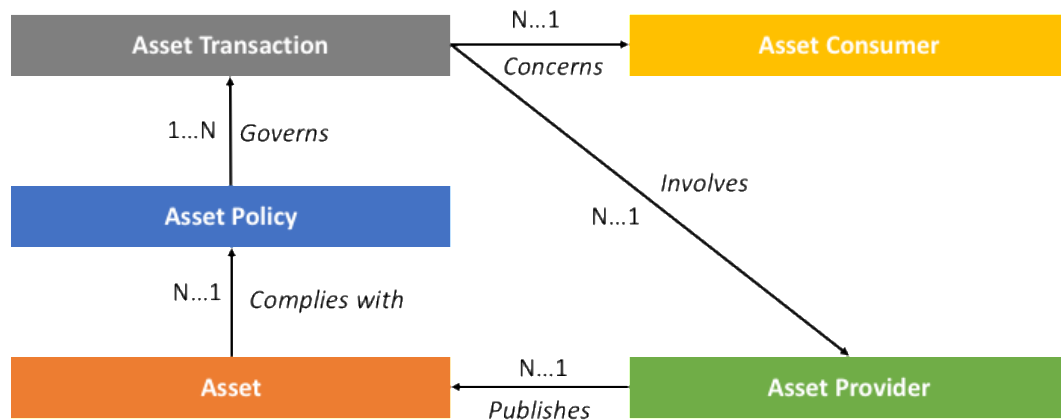


Figure 15: High-level Data Asset Policy Concept in AEGIS

For the Data Policy Framework, the AEGIS consortium has chosen to follow the Open Digital Rights Language (ODRL) W3C Recommendation¹⁰, which is a policy expression language that can be used for the description of assets. It is accompanied by an information model (currently ORDL Information Model 2.2¹¹) and by a vocabulary and expression document (currently ORDL Vocabulary & Expressions 2.2¹²) that describes how the terms in ODRL are to be used and encoded. The use of this model is able to accommodate the concepts that are presented in the AEGIS DPF and is also able to facilitate the Brokerage scheme envisaged in the project, as parts of the ODRL expressed DPF units will be stored in the AEGIS blockchain.

In general, the AEGIS DPF includes the following properties:

- **Assets Rights (AR)** encapsulating the rights that the data asset provider authorises the consumer to exercise for the specific data asset in order to clarify and assure the corresponding intellectual property rights. In accordance with the Creative Commons Rights Expression Language (CC REL)¹³, the set of common data right terms for data assets offered by the AEGIS platform are classified in the following categories:
 - **Permissions** including actions on the data asset that may or may not be allowed or desired, i.e.: Distribution (restricted or unrestricted publication and distribution of a data asset); Reproduction (from a given data asset, temporary or permanent reproductions can be created by any means and in any form, in whole or in part, including of any derivative data assets or as a part of collective data assets); Derivative Works (creation and distribution of any update, adaptation, or any other alteration of a data asset or of a substantial part of the data asset that constitutes a derivative data asset); Sharing (that permits Open-Public-Group based-Named-Internal access¹⁴ to a data asset).

¹⁰ W3C ODRL - <https://www.w3.org/community/odrl/>

¹¹ W3C (2018). ORDL Information Model 2.2 Available at: <https://www.w3.org/TR/odrl-model/>

¹² W3C (2018). ORDL Vocabulary & Expressions 2.2 Available at: <https://www.w3.org/TR/odrl-vocab/>

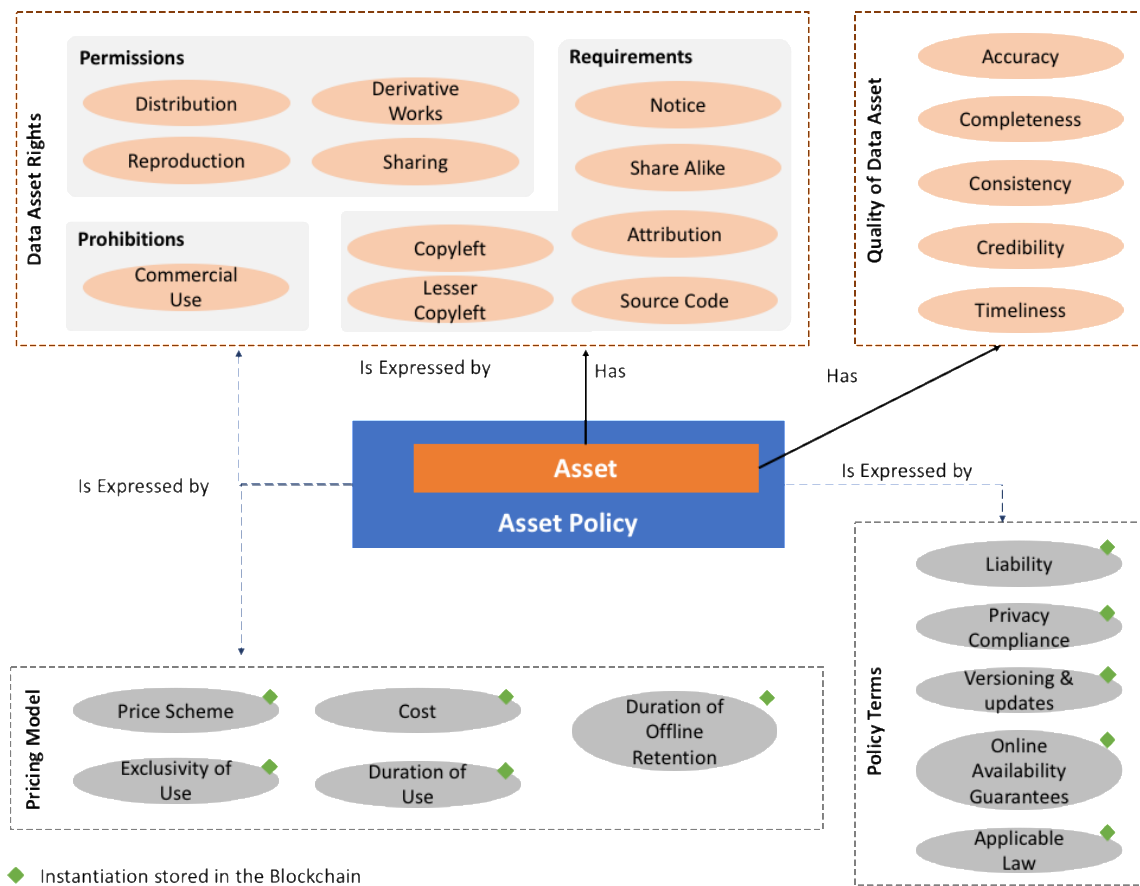
¹³ <https://creativecommons.org/ns>

¹⁴ <https://theodi.org/data-spectrum>

- **Requirements** including actions that may or may not be requested of the data asset consumer, i.e.: Notice (copyright and license notices to be kept intact); Attribution (credit to be given to copyright holder and/or provider); Share Alike (derivative works to be licensed under the same terms or compatible terms as the original work); Source Code (to be provided when exercising some rights granted by the license); Copyleft (derivative and combined works must be licensed under specified terms, similar to those on the original work); Lesser Copyleft (derivative works must be licensed under specified terms, with at least the same conditions as the original work; combinations with the work may be licensed under different terms).
- **Prohibitions** including actions a data asset consumer may be asked not to do, i.e.: Commercial Use (exercising rights permitting or forbidding use of a data asset for commercial purposes).
- **Quality of Assets (QoA)**, a complex concept that, depending on the asset type consists of different following facets, which are in most of the cases subjectively identified and measured. However, in large scale environment, such metrics can be calculated by the total perception of users of those specific assets, making them more objective. In AEGIS, we have identified the following QoA metrics, which will be measured through a 5star ranking:
 - **Accuracy** as a measure of correctness and precision (e.g. whether the dataset is error-free or the performance of an algorithm in terms of results is satisfactory).
 - **Completeness** defining the degree to which a data asset is sufficient in depth, breadth and scope.
 - **Consistency** by ensuring internal validity, i.e. two or more values do not conflict with each other.
 - **Credibility** as the degree to which a data asset is considered as trustworthy, traceable and reliable (e.g. through provenance, through the reputation of the data asset provider, by publishing the identity of the provider).
 - **Timeliness** as a measure of how sufficiently up-to-date a data asset (e.g. a dataset or data-as-a-service) is for a certain task, representing the timespan that such a data asset remains valid.
- **Pricing Model** that consists of:
 - **Price Scheme** including transaction, PAYG (pay-as-you-go) and subscription schemes. In detail, the transaction model allows data asset providers to charge for each single use of a data asset. The PAYG model is applicable in the case of data-as-a-service (provided through APIs) and allows charging the data asset consumers every time they call the provided APIs to retrieve data. The subscription model allows consumers to purchase data assets for a fixed period (e.g., a week, a month, or a year) and only pay once for this period with or without maximum limitations for how frequent they access a data asset.
 - **Cost** reflecting the exact amount to be paid for a certain period of time for use and/or offline retention.
 - **Exclusivity of use** that defines whether the data asset consumer requires exclusive use and the corresponding data asset becomes unavailable in the AEGIS platform (as long as the relevant data contract is active).

- **Duration of use**, the time period for which the data consumer has paid for use of the data asset in case of a subscription scheme.
- **Duration of offline retention**, the time period for which the data consumer is allowed to have offline / local access to the data asset. Nevertheless, in AEGIS there exist no way for checking the compliance of users against the duration timespan that an asset is allowed to be stored offline, however, this is a term that can be incorporated in a contract and users shall follow it.
- **Terms** consisting of more detailed terms regarding a data asset's evolution, support, indemnification, and limitation of liability. In this version of the AEGIS DPF, the following policy terms are defined:
 - **Liability** defining the data liability disclaimer and conditions.
 - **Privacy Compliance** to indicate whether and how the privacy aspects of a data asset have been appropriately handled through anonymization, fabrication, synthetisation, etc. depending on the level the data asset belongs to, e.g. Level 0 – Open data assets without any privacy aspects, Level 1 – Data assets with small privacy concerns, Level 2 – Data assets with significant privacy concerns and Level 3 – Data assets with severe privacy concerns.
 - **Online Availability Guarantees** describing the expected Quality of Service in case of data-as-a-service assets.
 - **Versioning & updates** whether the data asset consumer has access to updates and latest versions of the data asset.
 - **Applicable Law** including the regulatory framework of the country that is responsible for settlement of any disputes.

As depicted in the figure below, the AEGIS DPF is the linking point between the AEGIS assets and the Transactions that will be facilitated via the Brokerage Engine, combining information coming from the AEGIS Data Store (the assets), the AEGIS Harvester (the metadata of the assets) and of the Brokerage Engine (existing transactions and transaction eligibility).

**Figure 16: AEGIS DPF Properties**

Following the above, the decision taken by the project was to split parts of the DPF amongst the Brokerage engine and the Metadata Harvester. The following a clear separation of the different DPF properties identified (see following table), facilitates smoother integration from the IT perspective, and is able to serve users with the requested information easier and faster.

Table 4: DPF Properties Split amongst AEGIS DataStore and Brokerage Engine

DPF Property	Stored on the AEGIS DataStore as Metadata	Stored in the AEGIS Brokerage Engine
Assets Rights (AR)	Information stored in the AEGIS DataStore as rights metadata accompanying the asset	-
Quality of Assets (QoA)	Information stored in the AEGIS DataStore as quality metadata accompanying the asset	-

Pricing Model	Information stored in the AEGIS DataStore as pricing metadata accompanying the asset	Instantiation during a transaction stored in the blockchain
Terms	Template Document should exist in the DataStore	Stored in the payload of the transaction as a reference document containing the terms

Using this approach, it becomes easier for AEGIS stakeholders to describe their assets with DPF properties only in case they would like to make them shareable (thus DPF properties are only required when a dataset becomes available for sharing and not in case it is used for private operations), while at the same time condition/policy enforcement is executed more rapidly, and any transaction is stored in an immutable ledger which supports multi-cluster AEGIS deployments.

7.2. Business Brokerage Framework updates

In AEGIS, as sharing, utilisation, and exploitation of data-related assets is amongst the core aims that would lead to novel business opportunities and would strengthen the data value chain concept of the project, a lightweight Business Brokerage Framework (BBF) has been designed in order to formally dictate transaction terms and oversee the smooth and rightful execution of them. In general, the AEGIS BBF is thought of acting as a supervisor method in asset sharing operations performed over the platform, as those are defined in the AEGIS DPF. The DPF acts as a reference point for the BBF, towards identifying important assets' properties, which are essential for a transaction (such as the originators, the pricing schemes, etc.). In line with this, the BBF is implemented as a layer that oversees transactions performed between the AEGIS clusters.

In principle, data sharing and exchange is being performed without a supervisor, as long as there is no monetary transaction and as long as both parties (the “producer” and the “consumer”) respect certain rules, as those implied by the licenses of the assets they will exchange (for example the license might prohibit the “consumer” to extend an asset, or it might oblige the “producer” to provide certain guarantees regarding the asset’s functionalities) as defined with the help of the AEGIS DPF. However, this practise is solely based on the good intentions and will of both parties and does not generally meet the criteria of building a “trusted” environment of asset exchange. In this respect, the AEGIS BBF comes as a methodology used to strengthen trust between parties over the whole AEGIS value chain and create a distributed ledger of transactions that can be used to log all the transaction that might happen between the different AEGIS clusters that may be deployed, resulting in the deployment of a transaction network that can be used for validating and executing transaction between diverse actors. As such, the AEGIS BBF is envisioned to be supported by a blockchain implementation that will allow the different users to perform transactions (whether these include a type of payment or not) and all comply to the same rules.

What is of importance for the implementation of the proposed solution is the definition of various nodes within the AEGIS ecosystem that will play the role of the miners and that will

hold the distributed ledger of the transactions. Following the exploitation plan of the project and the overall architectural decision, AEGIS will be installed in multiple clusters owned by different organisations, to run private analyses with private data, but also to expose publicly data that they can offer for free or with a paid license. This distributed architecture is very well aligned with blockchain architectures, where the different clusters can play the role of the different nodes which are necessary to strengthen the overall network.

The following figure provides an illustrative presentation of the BBF framework that can be applied to the AEGIS platform and be implemented with the help of blockchain technology.

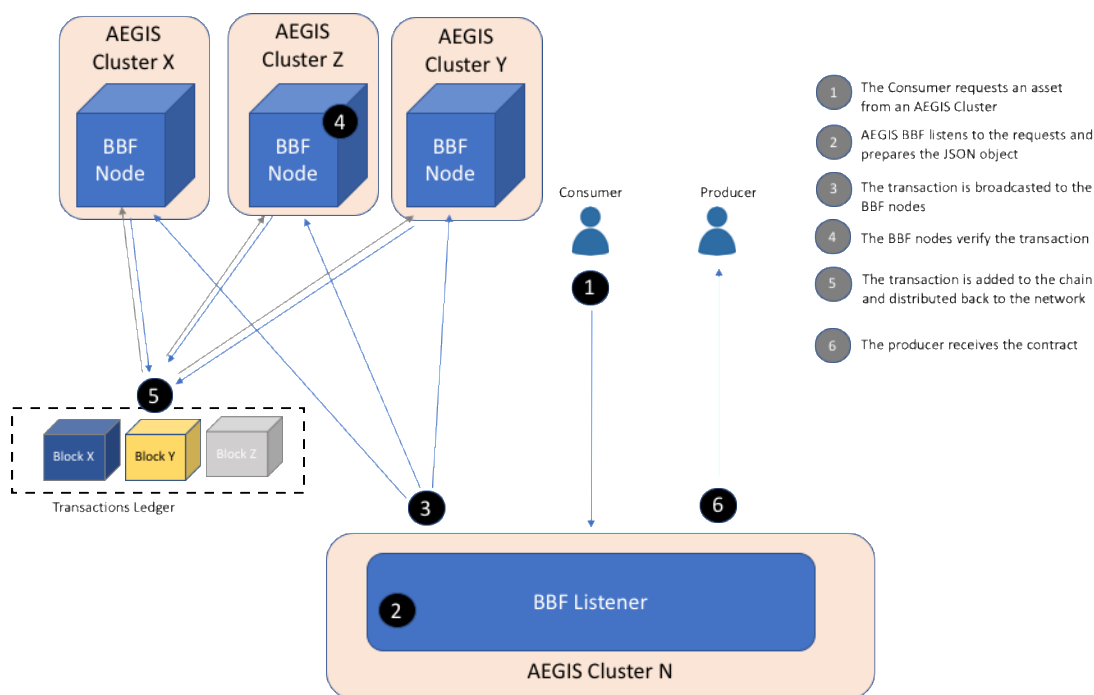


Figure 17: AEGIS BBF Concept

Taking into account the underlying approach of the blockchain protocol and the data trust and security aspects, the AEGIS BBF contains a small set of rules that are in place to better describe each type of transaction and offer a mutual understanding about the code of conduct of each transaction amongst the involved parties. Those rules are the following:

- #1 Each transaction is unique and can be identified as unique
- #2 Each transaction should have one, and only one seller
- #3 Each transaction should have one, and only one customer
- #4 Each transaction refers to one, and only one asset
- #5 Each transaction can be automatically, or manually executed, based on the asset under transaction
- #6 Each transaction is accompanied by the date and time that it has been executed. In case of automatic transaction, time is specified by the system based on the system request. In case of manual transactions, the transaction's data and time is considered as the time of acceptance of the transaction by a "seller"
- #7 Each transaction is accompanied by a contract document specifying certain aspects for the transaction
- #8 Each transaction is public
- #9 The specifics of each transaction contract may not be disclosed publicly

#10 No other parties gain any benefit from a transaction that they are not part of

The rules identified above are mostly “guidelines” and suggest what users should expect from the BBF that is present over the platform. With regards to the benefits for the “seller”, these can be described when publicly setting an asset, resulting into the creation of automatically executed transactions (in case licensing and pricing is straightforward), or in manual transaction (where there has to be an exchange of documents between both parties until they finally agree).

Each transaction to be performed over the platform can be described with the following conceptual JSON object in the BBF, which in reality is translated to the blockchain models that are provided in WP4.

```
{ "AEGISBBFTransaction": {  
  
  "transactionid": idoofTransaction,  
  
  "datetime": "TimetampofTransaction",  
  
  "executiontype": "Automatic/Manual",  
  
  "assetconcerned": "AssetURI",  
  
  "parties": {  
  
    {"producer": PersistentUserIDofProducer},  
  
    {"consumer": PersistentUserIDofConsumer}  
  }  
  
  "contract": "ContractTemplateURI"  
}}
```

Figure 18: AEGIS BBF JSON Example

The main elements as expressed in rules #1-#6 are present in the JSON object and are the ones that seal a transaction as valid.

With regards to rule #6, it is noted that the automatic execution of micro-contracts, as those envisioned in AEGIS, can be supported by the provision of contract template documents. As such, the JSON object above contains the “contract” field that is used to point to a template document of that kind. Such document templates should be present in the platform to facilitate rapid execution of baseline transactions, while more complicated transactions could be supported by specific contracts that may be uploaded to the platform by the “seller” in each case. As such, each transaction will be accompanied by such a template document, which will be filled in with the related data for each transaction in alignment with the AEGIS DPF and that would specify the obligations of each party.

As identified above, the AEGIS BBF comes as a lightweight solution that is able to oversee transactions and log them in order to enhance trust to the platform’s operations and more importantly between the different users of the platform. However, certain limitations apply and are not tackled by the project. Such issues have to do with IPRs and automatic license compatibility checking of, data transportation and usage rights and obligations in the different member states.

8. CONCLUSION

The objective of this deliverable is to document the final decisions of the AEGIS consortium related to WP2, leading to the definition of the semantics, the data handling algorithms and the overall logic that will combine data from various sources and deliver value for the engaged stakeholders. The steps of the AEGIS Big Data Value Chain are investigated mostly from a theoretical point of view, highlighting the connection between the decisions taken within the context of WP2 with the technical needs of WP3 and WP4, and the stakeholders' and demonstrators' needs arisen from WP1 and WP5.

The AEGIS Data Value Chain Bus definition has been updated following the AEGIS platform architecture defined in D3.3, and the methods to define the appropriate patterns for harmonising and processing the data within the AEGIS platform are described. The characteristics of the data sources have been defined and described in D2.2, while the methodology and mechanisms for harvesting and harmonising data and metadata simplifying their further processing and management have been updated.

The semantic vocabularies and ontologies as well as their own repository have been listed and described in D2.1. The Linda Workbench infrastructure¹⁵ is the basis of the AEGIS Vocabulary Repository, and it has been enhanced to join the AEGIS needs, for instance the list of vocabularies previously available in LinDa has been augmented considering the public safety and personal security domains.

The format for storing tabular raw data in AEGIS has been identified as the CSV, although the platform will allow a later conversion of the file, while the DCAT-AP ontology has been identified as reference for describing the structure and content of data for all collected by the AEGIS harvester metadata. The achieving of high harmonisation and interlinking of metadata, however, needs a manual refinement of metadata; the AEGIS component that provides this service is the Data Annotator tool.

Updated lists of the analytic algorithms and visualisation tools and techniques supported by the AEGIS platform have been drawn up, even if they cannot be considered as final, since it is foreseen that, following the feedback received from the stakeholders and the demonstrators, further expansions could be done.

The Data Policy and Business Brokerage Frameworks are the components that provide to the AEGIS stakeholders a mean for offer their assets following rules dependent on the asset to be exchanged and guaranteeing data quality. The data-value chain of the project will be enriched adding a "trusted" environment of asset exchange based on blockchain technology.

¹⁵ <http://linda.epu.ntua.gr/>