



---

## **Deliverable 4.3:** Final responsive platform prototype, modules and common communication protocol

---

Robert Lorenz, Paul Grunewald, Sabine Barthold/TUD  
Alexandros Pournaras, Chrysa Collyda/CERTH  
Andrea Zielinski, Peter Mutschke/GESIS  
Angela Fessler, Ilija Šimić/KC  
Aitor Apaolaza/UMAN  
Ahmed Saleh, Iacopo Vagliano, Till Blume/ZBW

29/03/2019

Work Package 4: Iterative MOVING platform development

**TraininG towards a society of data-saVvy inforMation  
prOfessionals to enable open leadership INnovation**

Horizon 2020 - INSO-4-2015  
Research and Innovation Programme  
Grant Agreement Number 693092

Dissemination level	PU
Contractual date of delivery	31/03/2019
Actual date of delivery	29/03/2019
Deliverable number	4.3
Deliverable name	Final responsive platform prototype, modules and common communication protocol
File	moving_d4.3-v1.0.pdf
Nature	Report
Status & version	Released v1.0
Number of pages	59
WP contributing to the deliverable	4
Task responsible	TUD
Other contributors	CERTH, GESIS, KC, UMAN, ZBW
Author(s)	Robert Lorenz, Paul Grunewald, Sabine Barthold/TUD Alexandros Pournaras, Chrysa Collyda/CERTH Andrea Zielinski, Peter Mutschke/GESIS Angela Fessler, Ilija Šimić/KC Aitor Apaolaza/UMAN Ahmed Saleh, Iacopo Vagliano, Till Blume/ZBW
Quality Assessors	Angela Fessler, Ilija Šimić/KC
EC Project Officer	Hinano SPREAFICO
Keywords	Prototype, Responsive Design, MOVING platform

## Table of contents

<b>Executive Summary</b>	<b>5</b>
<b>Abbreviations</b>	<b>6</b>
<b>1 Introduction</b>	<b>7</b>
1.1 History of the document . . . . .	7
1.2 Purpose of the document . . . . .	7
<b>2 Platform overview</b>	<b>7</b>
2.1 Instances . . . . .	7
2.2 Architecture . . . . .	8
<b>3 MOVING web application</b>	<b>9</b>
3.1 Search . . . . .	9
3.1.1 Elasticsearch configurations . . . . .	9
3.1.2 Search history . . . . .	10
3.1.3 Search results visualizations . . . . .	10
3.2 Communities . . . . .	21
3.2.1 Badges . . . . .	28
3.3 Learning environment . . . . .	32
3.4 Responsive design . . . . .	35
<b>4 Recommender System</b>	<b>39</b>
4.1 The HCF-IDF model . . . . .	40
4.2 Building user profiles . . . . .	40
4.3 Implementation . . . . .	41
<b>5 Adaptive Training Support</b>	<b>41</b>
5.1 "Learning-how-to-search" widget . . . . .	41
5.2 "Curriculum reflection" widget . . . . .	42
5.2.1 "Curriculum learning and reflection" part . . . . .	43
5.2.2 "Overall progress" part . . . . .	44
<b>6 User interaction tracking</b>	<b>45</b>
6.1 Interaction tracking . . . . .	45
6.2 WevQuery . . . . .	45
<b>7 Data acquisition and processing</b>	<b>45</b>
7.1 Data acquisition . . . . .	45
7.1.1 Data accessible through the MOVING platform . . . . .	45
7.1.2 Web crawling . . . . .	46
7.1.3 Bibliographic Metadata Injection . . . . .	47
7.2 Data processing . . . . .	51
7.2.1 Data Integration Service . . . . .	51
7.2.2 Author Name Disambiguation . . . . .	52
7.2.3 Deduplication . . . . .	52
7.2.4 Named Entity Recognition and Linking . . . . .	53
7.2.5 Video analysis . . . . .	53
7.3 Explore . . . . .	54
<b>8 Conclusion</b>	<b>57</b>
<b>References</b>	<b>58</b>

## List of Figures

1	MOVING platform architecture . . . . .	8
2	Search history view . . . . .	11
3	Concept Graph visualization . . . . .	12
4	Concept Graph – entity to document relevance . . . . .	12
5	Concept Graph – entity to entity relevance . . . . .	13
6	Concept Graph – improved ring-menu . . . . .	13
7	Concept Graph – node aggregation . . . . .	14
8	Concept Graph – mobile view . . . . .	14
9	uRank visualization . . . . .	15
10	uRank – mobile view . . . . .	17
11	Top Properties visualization . . . . .	18
12	Top Properties – mobile view . . . . .	19
13	Tag Cloud visualization . . . . .	19
14	Tag Cloud – mobile view . . . . .	20
15	MOVING communities . . . . .	21
16	Create a new community . . . . .	22
17	MOVING MOOC community . . . . .	23
18	MOVING MOOC - week 4 . . . . .	24
19	MOVING MOOC - wiki . . . . .	25
20	MOVING MOOC - badges . . . . .	26
21	MOVING user dashboard . . . . .	27
22	Overview of the Open Badge workflow . . . . .	28
23	Assertion of a MOVING badge . . . . .	31
24	MOVING learning environment . . . . .	32
25	Learning tracks for information literacy start page . . . . .	33
26	Microlearning session card . . . . .	34
27	Learning tracks page on screens of different devices . . . . .	36
28	Communities page viewed by an administrator on screens of different devices . . . . .	37
29	Search results page on screens of different devices . . . . .	38
30	The Recommender System widget . . . . .	39
31	Overview of the MOVING Recommender System . . . . .	39
32	A concept tree . . . . .	40
33	Learning-how-to-search widget . . . . .	42
34	Curriculum reflection widget: curriculum learning part . . . . .	43
35	Curriculum reflection widget: reflection part . . . . .	43
36	Overall progress widget . . . . .	44
37	MOVING Crawler architecture. . . . .	47
38	System overview of the Bibliographic Metadata Injection service . . . . .	48
39	The functionality and data flow of the Bibliographic Metadata Injection service . . . . .	50
40	Basic functionality of the Data Integration Service . . . . .	52
41	The interaction of the various components of the MOVING platform for the <i>Explore</i> functionality . . . . .	55
42	The UI and the Concept Graph visualization of the uploaded documents . . . . .	56

## List of Tables

1	Comparison of the MOVING platform's data with respect to EconBiz and VideoLecture.NET . . . . .	46
---	---	----

## Executive summary

This Deliverable D4.3 is an update of Deliverable D4.2 "Initial responsive platform prototype, modules and common communication protocol". The aim of this document is to report the development results of the MOVING platform up to M36 of the project. Furthermore, we detail the status of the individual components integrated into the MOVING platform and show their final results. For this reason, this deliverable provides a final update on the MOVING web application as well as on the Recommender System, the Adaptive Training Support, the user interaction tracking, and the data acquisition and processing components.

The MOVING web application, described in Section 3, has been subject to major changes in order to improve and implement the platform requirements and community building functionalities gathered in the Deliverables D1.3 "Initial evaluation, updated requirements and specifications" and D2.2 "Updated curricula and prototypes for adaptive training support and introductory MOVING MOOC for community building" (Apaolaza et al., 2018; Günther et al., 2018). The newly integrated Recommender System (Section 4) and the significantly improved Adaptive Training Support (Sections 5) supports users in suggesting further documents and providing learning material according to the user's needs respectively. Extensive changes in the user interaction tracking by improving its scalability, as well as the privacy of the data are explained in Section 6. The final status of the integration of the data acquisition and processing components are emphasized in Section 7. Specifically, the web crawlers (Section 7.1.2) as well as the Bibliographic Metadata Injection service (Section 7.1.3) have been further updated to increase the data quantity accessible through the MOVING platform search. To improve the data quality of the MOVING index, the Data Integration Service (Section 7.2.1) as well as the services for Author Name Disambiguation (Section 7.2.2), Deduplication (Section 7.2.3), Named Entity Recognition and Linking (Section 7.2.4), and video analysis (Section 7.2.5) have been integrated and completed. Furthermore, in Section 7.3 we are presenting the novel *Explore* functionality which enables users to visually analyse PDF files by showing connections between extracted entities and related documents in the MOVING platform. An in-depth insight into the functionalities of the tracking component as well as the data acquisition and processing components can be found in Deliverable D3.3 "Technologies for MOVING data processing and visualisation v3.0" (Vagliano et al., 2019).

Lastly, we conclude the final implementation of the MOVING platform prototype in Section 8, highlighting the achievements of the third year of the project.

## Abbreviations

Abbreviation	Explanation
AND	Author Name Disambiguation
API	Application Programming Interface
ATS	Adaptive Training Support
BMI	Bibliographic Metadata Injection
CF-IDF	Concept Frequency-Inverse Document Frequency
DCMI	Dublin Core Metadata Initiative
DFXP	Distribution Format Exchange Profile
DIS	Data Integration Service
D3	Data-Driven Documents
FDC	Focused web Domain Crawler
GVF	Graph Visualisation Framework
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HCF-IDF	Hierarchical Concept Frequency-Inverse Document Frequency
IDF	Inverse Document Frequency
JS	JavaScript
JSON	JavaScript Object Notation
JVM	Jave Virtual Machine
LOD	Linked Open Data
MOOC	Massive Open Online Course
NER&L	Named Entity Recognition and Linking
NPM	Node Package Manager
ORCID	Open Researcher Contributor Identification
QA	Quality Assurance
RDF	Resource Description Framework
REST	REpresentational State Transfer
RS	Recommender System
SEC	Search Engine-based web Crawler
SPARQL	SPARQL Protocol and RDF Query Language
SSM	Social Stream Manager
SSOAR	Social Science Open Access repository
SVG	Scalable Vector Graphics
TF-IDF	Term Frequency-Inverse Document Frequency
TOC	Table Of Contents
TS	TypeScript
URL	Uniform Resource Locator
VPN	Virtual Private Network
WevQuery	Web Event Query Tool

# 1 Introduction

## 1.1 History of the document

Date	Version
13.02.2019	v0.1: Initial TOC draft
15.02.2019	v0.2: Second TOC draft
08.03.2019	v0.8: QA ready
22.03.2019	v0.9: QA comments addressed
29.03.2019	v1.0: Final version

## 1.2 Purpose of the document

This document describes the updated and final implementation of the MOVING platform prototype up to M36 with respect to what was described in the previous Deliverable D4.2 "Initial responsive platform prototype, modules and common communication protocol" (Gottfried, Pournaras, et al., 2017). First, we give an overview of the updated platform architecture (Section 2). Then we describe the final version of the components which have been integrated in the MOVING platform. This includes the MOVING web application (Section 3), the Recommender System (RS) (Section 4), the Adaptive Training Support (ATS) (Section 5), the user interaction tracking (Section 6) and the data acquisition and processing components including the novel *Explore* functionality (Section 7).

# 2 Platform overview

In this section we give an overview of all available MOVING instances and the final architecture of the MOVING platform. The architecture was further developed on the basis of D4.2 (Gottfried, Pournaras, et al., 2017) in order to improve and implement the platform requirements and community building functionalities gathered in the Deliverables D1.3 "Initial evaluation, updated requirements and specifications" (Apaolaza et al., 2018) and D2.2 "Updated curricula and prototypes for adaptive training support and introductory MOVING MOOC for community building" (Günther et al., 2018). We briefly describe the newly integrated components and show their relationships to the other platform components. The aim of this section is to show the architectural changes of the last months and how they have been integrated into the existing platform. For in-depth information, please refer to the respective sections or deliverables.

## 2.1 Instances

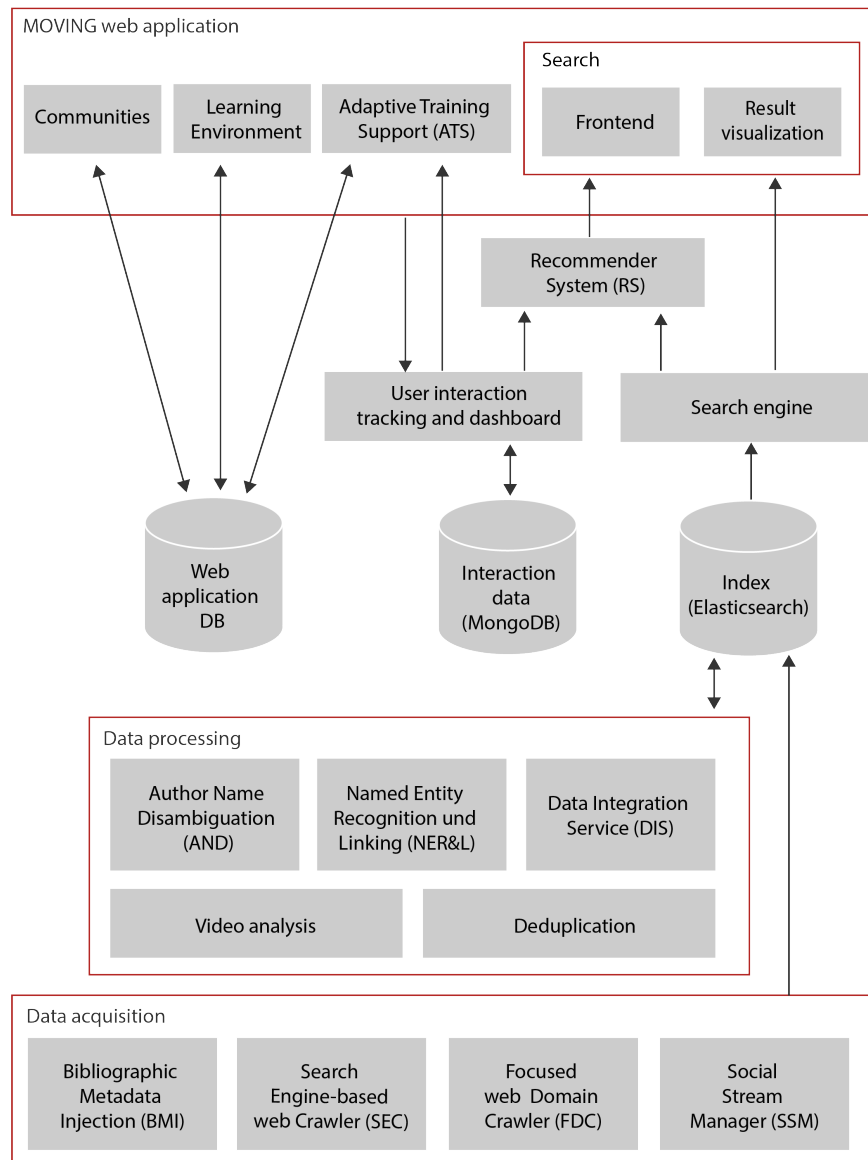
At the moment, there are three instances of the platform installed on the servers of TU Dresden. Each of them has its own branch in our MOVING-repository, which allows us to deploy the servers independently.

- Instance 1 is accessible under the address <https://moving.mz.tu-dresden.de>. This is the main platform which was released to the public in M21. Since then, it was constantly updated and improved and will be used mainly by use case partner 2.
- The second platform installation is reachable under the address <https://moving2.test.mz.tu-dresden.de><sup>1</sup>. It was cloned from the first instance and we are working on making it available to the public as well. Next to the improvements for the first installation it contains additional functionalities and a different web design specific to use case partner 1. On both platforms, only stable components and improved functionalities are deployed to not impair the user experience while developing and testing. The deployments are only executed when they are absolutely necessary, i.e. when the development of new functionalities or components has been completed or important updates have to be installed.
- Moreover, there is a third installation reachable under the address <https://moving.test.mz.tu-dresden.de><sup>1</sup> where changes to the platform are constantly deployed. This version contains ongoing developments of the platform and its components. It is used by the consortium members only to implement and test new features on a regularly, i.e. daily basis.

<sup>1</sup>The access is limited to the network of the TU Dresden. We can provide access to this instance for interested parties outside of the TUD network via a VPN connection. For this, please contact Robert Lorenz ([robert.lorenz4@tu-dresden.de](mailto:robert.lorenz4@tu-dresden.de)).

## 2.2 Architecture

Figure 1 shows the updated and final platform architecture that is equal to all installed instances. It shows the most important components and their relationships to each other. The main components are still as follows: the MOVING web application, the search, the data acquisition and processing components as well as the user interaction tracking.



**Figure 1:** MOVING platform architecture

The MOVING web application is the core of the platform which represents the interface to the user (see Section 3). It has undergone extensive changes in order to improve and implement the platform requirements and functionalities. During the last months, the project environment was revised to integrate the community features (see Section 3.2), the learning environment was implemented in its full specification (see Section 3.3), and the user interface and its responsive design were improved (see Section 3.4). Part of the web application is the search interface consisting of the search frontend and the search results visualizations (see Section 3.1). The frontend was improved by adding a search history that shows the last performed user searches (see Section 3.1.2). The visualizations were extensively enhanced and are now able to show the results set as a Concept Graph, a dynamic document ranking view (uRank), a Tag Cloud or a Top Properties representation (see Section 3.1.3). Moreover, the Adaptive Training Support widget which is shown next to the search results



was significantly improved (see Section 5). It supports users to learn-how-to-search and provide learning material according to the user's needs. In addition, the Recommender System widget was added below the ATS. It is capable referring to suitable documents to the users of the platform by evaluating the last search queries (see Section 4).

The web application and its widgets pull data from two sources directly: the web application database holds the data for the communities, the learning environment and the ATS, while the MOVING search engine is responsible for the search results and the recommended documents. The user interaction tracking in turn extracts data from the web app and stores it in a separate database (see Section 6). The tracking of the users provides additional data for the ATS as well as the RS, which in turn provides the basis for the user support by these two widgets.

The MOVING index for the search interface is again populated by various data acquisition components like the web crawlers (see Section 7.1.2) as well as the Bibliographic Metadata Injection service (see Section 7.1.3) to increase the data quantity accessible through the MOVING platform search. In addition, data processing components have been incorporated to improve the data quality of the search results. The Data Integration Service (see Section 7.2.1) as well as the services for Author Name Disambiguation (see Section 7.2.2), Deduplication (see Section 7.2.3), Named Entity Recognition and Linking (see Section 7.2.4), and video analysis (see Section 7.2.5) refine and enrich the documents stored in the index.

### 3 MOVING web application

In the months since the preceding Deliverable D4.2 "Initial responsive platform prototype, modules and common communication protocol" (Gottfried, Pournaras, et al., 2017) we extensively updated and extended the MOVING web application. We increased the usability and the efficiency of the MOVING search and its results visualizations (see Section 3.1). We fully redesigned the project environment to add community features (see Section 3.2), added a learning environment (see Section 3.3) and updated the responsive design (see Section 3.4) as well.

#### 3.1 Search

In Section 3.1.1, we describe how we configured our MOVING cluster to enhance the efficiency and increase the availability of our MOVING index. In Section 3.1.2, we present the new search history feature. In addition to this, we revised and improved the search results visualisations (see Section 3.1.3).

##### 3.1.1 Elasticsearch configurations

As we described in the Deliverables D4.1 (Gottfried, Grunewald, et al., 2017) and D4.2 (Gottfried, Pournaras, et al., 2017), our MOVING search engine is based on Elasticsearch. In MOVING, we index dozens of millions of documents. In order to speed up the retrieval process of Elasticsearch, we changed the following configurations of our elasticsearch Index:

- **Heap size.** One of the key strengths of Elasticsearch is that it benefits from high amount of RAM, both for query cache and for operating system cache of Memory Mapped Files. The heap size controls the maximum amount of memory that Elasticsearch can use. Elasticsearch recommends allocating half of the memory to Elasticsearch Heap. Furthermore, they recommended not to exceed 32 GB in order to benefit from the Java Virtual Machine (JVM) object compression<sup>2</sup>. Therefore, we changed our heap size configuration from the default value 1 GB to 32 GB.
- **Number of shards.** Shards are instances of Elasticsearch nodes (more details in Deliverable D4.2). Each shard has data that need to be kept in memory and use heap space. The sharding process itself allows Elasticsearch to distribute operations across shards, thus increasing performance/throughput<sup>3</sup>. In the last version of our MOVING's search engine, that has been described in Deliverable 4.2, we used only one shard. However, one shard is not sufficient to handle the increasing amount of data in our MOVING index efficiently. Therefore, we increased the number of shards to 4<sup>4</sup>.

<sup>2</sup><https://www.elastic.co/guide/en/elasticsearch/guide/2.x/heap-sizing.html>

<sup>3</sup>[https://www.elastic.co/guide/en/elasticsearch/reference/6.2/\\_basic\\_concepts.html](https://www.elastic.co/guide/en/elasticsearch/reference/6.2/_basic_concepts.html)

<sup>4</sup><https://www.elastic.co/blog/howmanyshardsshouldihaveinmyelasticsearchcluster>

### 3.1.2 Search history

We created a new view to allow the users of our MOVING platform to view and replicate their previous searches. The new search history view is connected with WevQuery, the tracking tool of MOVING which logs user-interaction data. For more information, please see Deliverables D3.1 (Blume et al., 2017), D3.2 (Vagliano et al., 2018), and D3.3 (Vagliano et al., 2019). From WevQuery, we get the information about the previous user searches, at which time the user performed the search query, and the number of documents that has been retrieved to him. The retrieved information from WevQuery are represented in JSON format, as shown in Listing 1.

```

1  [
2  {
3  {
4    "_id": {
5      "episodeCount": 3,
6      "searchID": 6
7    },
8    "timestamps": 1537808910520,
9    "query": "quantum mechanics",
10   "docCount": "1",
11   "urlFull": "http://localhost:3000/search?utf8=%E2%9C%93&search_domain=research&view_mode=results&q=quantum
12   +mechanics&filters%5Bpersons%5D%5B%5D=Ferner%252C+A."
13 },{
14   "_id": {
15     "episodeCount": 3,
16     "searchID": 5
17   },
18   "timestamps": 1537807212532,
19   "query": "advanced_basic",
20   "docCount": "0",
21   "urlFull": "http://localhost:3000/search?utf8=%E2%9C%93&search_domain=research&advanced_query%5Btitle%5D=
22   advanced_basic&advanced_query%5Babstract%5D=Abstract&advanced_query%5Bfulltext%5D=&advanced_query%5B
23   person%5D=person"
24 },{
25   "_id": {
26     "episodeCount": 2,
27     "searchID": 5
28   },
29   "timestamps": 1537802411045,
30   "query": "advanced_basic",
31   "docCount": "0",
32   "urlFull": "http://localhost:3000/search?utf8=%E2%9C%93&search_domain=research&advanced_query%5Btitle%5D=
33   advanced_basic&advanced_query%5Babstract%5D=Abstract&advanced_query%5Bfulltext%5D=safagasg
34   &advanced_query%5Bperson%5D=person"
35 },{
36   "_id": {
37     "episodeCount": 2,
38     "searchID": 4
39   },
40   "timestamps": 1537802226442,
41   "query": "fulltext",
42   "docCount": "0",
43   "urlFull": "http://localhost:3000/search?utf8=%E2%9C%93&search_domain=research&advanced_query%5Btitle%5D=
44   &advanced_query%5Babstract%5D=Abstract&advanced_query%5Bfulltext%5D=fulltext&advanced_query
45   %5Bperson%5D=person"
46 }
47 ]

```

**Listing 1:** Example of the search history of a user retrieved by WevQuery

The retrieved information from WevQuery are then utilized to build the search history view. Figure 2 shows an example of the search history view of a user.

### 3.1.3 Search results visualizations

The following visualizations have been implemented in the final version of the MOVING Platform:

**Concept Graph** For the discovery and exploration of relationships between documents and their properties.

**uRank** A tool for the interest-driven exploration of search results.

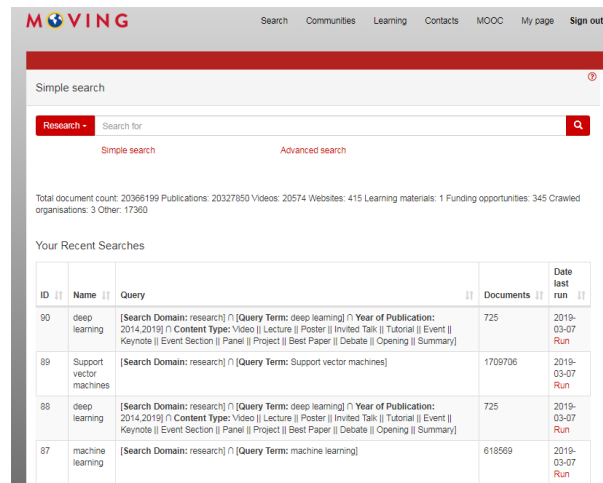


Figure 2: Search history view

**Top Properties** A bar chart displaying aggregated information about the properties of the retrieved documents.

**Tag Cloud** A visualization for the analysis of keyword frequency in the retrieved documents.

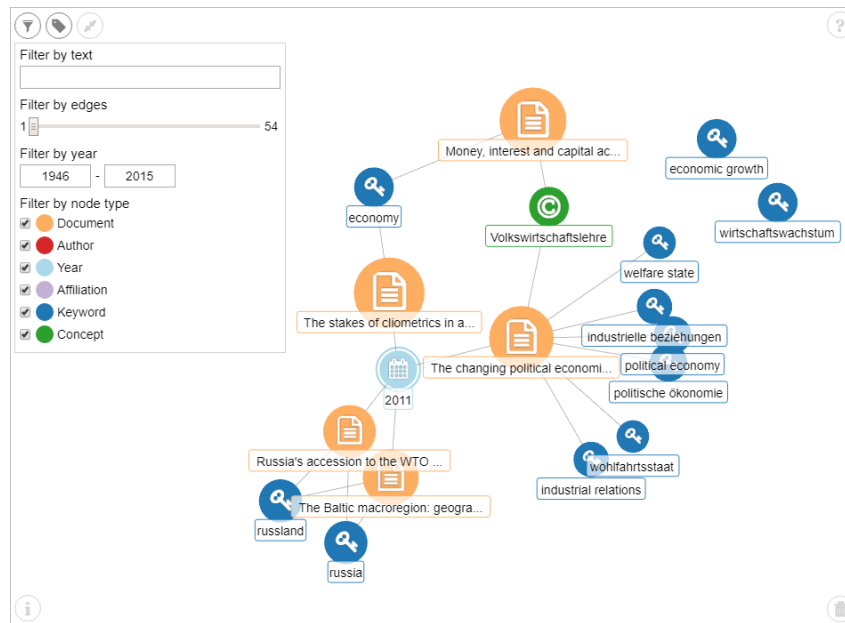
The functionalities of each of the visualizations in their latest form have been described in D3.3 "Technologies for MOVING data processing and visualisation v3.0" (Vagliano et al., 2019), therefore, only a technical description of all visualizations will be given here.

Switching between visualizations is now done asynchronously so that there is no need to refresh the whole page but only the wished visualizations. This ensures a better user experience for all MOVING users. For this purpose, the same container is used for the ranked result list as well as all of the visualizations. When switching from the ranked result list to one of the visualizations, or between the different visualizations, the container gets emptied and is replaced with the new selected visualization. To implement this, the data retrieval for all visualizations is implemented in the same way. First, the visualization source code gets loaded in the container. Second, a request is triggering the search engine on the server to retrieve the first 50-100 of the most relevant search results for the currently initiated search query. To minimize the amount of data that is sent back from the server to the client, the visualization explicitly requests from the search engine only those result fields that are necessary to display the visualization. This significantly reduces the size of the response sent from the server to the client, since, for example none of the visualizations require the full text of the results. All of the visualizations were designed in a responsive manner. They fully use the available width of the container in which they are drawn and adapt immediately if the size of their container changes.

**Concept Graph: an interactive network visualization** After the Concept Graph (Figure 3) is loaded into its dedicated container, it triggers a request to the search engine for 50 results. The results in the response are returned as an array of result objects. The Concept Graph is built by going through each of these result objects, and creating nodes for the results and its properties (meta-data and extracted entities). While creating the nodes, connections are created between the node representing the result and the nodes representing properties. If a node of a property already exists in the graph, only the connection to the existing node will be created. In addition to the connections between the documents and their properties, there are also connections between properties. These connections include those of the co-authors, and those produced by the co-occurrence analysis of the extracted entities. After all the result objects have been analyzed and the graph has been built in the background, the initial node layout is displayed.

Since, the Graph Visualization Framework (GVF) of the Concept Graph has been described in D4.2 (Gottfried, Pournaras, et al., 2017), and the available functionalities have been summarized in D3.3., we will focus here on the technical description of the newly added features.

**Node types** - The following types are available in the Concept Graph: Document, Author, Year, Affiliation (Organization), Keyword, Concept and Entity. Additionally, the entity nodes are further divided into 9 sub-types: person, location, organization, SAGE-METHOD, SAGE-RESEARCHFIELD, SAGE-THEORY, SAGE-DATATYPE, SAGE-MEASURE and SAGE-TOOL. These sub-types of the entity nodes are visu-

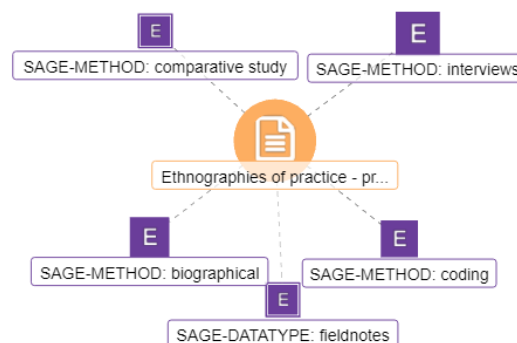


**Figure 3:** Concept Graph with the opened filter menu

ally not distinguishable from each other, but a filter can be applied to get only those nodes that are of interest for the user.

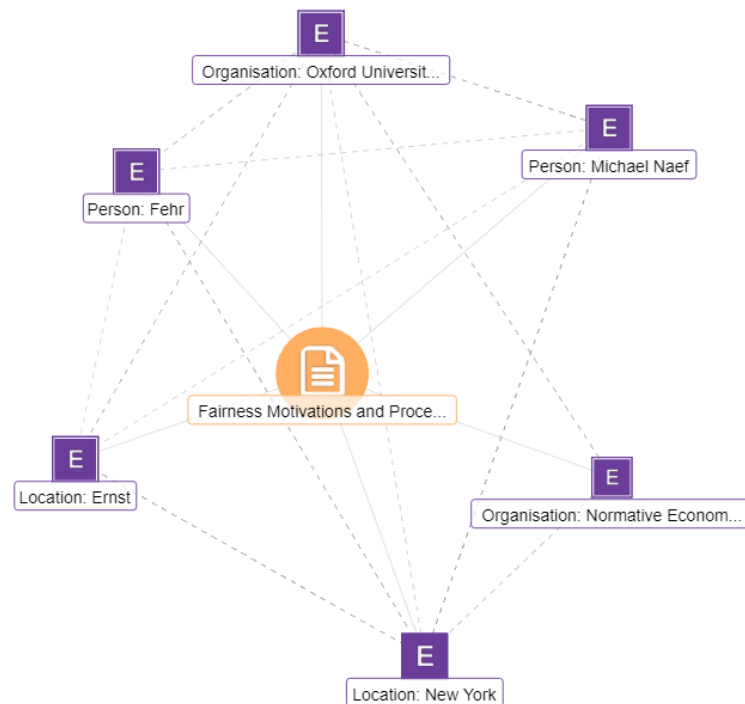
*Data retrieval for graph creation* - Since more node types are now supported by the graph, the corresponding data request needed was also extended. It includes the titles of the results, the document types, URLs, keywords, concepts, publishing dates, authors, organizations and entities extracted from the full text of the result.

*Entities, co-occurrence analysis and relevance edges* - The entities extracted from the documents of the results might contain information about the relevance of the entity to the document (TF-IDF score), or the positions, which are represented as an array of indexes, where the entities occur in the document of the result. The information about the relevance of the entity to the document is used to display a relevance edge between the document and the entity, which can be seen in Figure 4. Edges representing relevance are drawn with a dashed line, while the relevance is shown through the opacity of the line. The darker the dashed line, the more relevant the entity is in the document.



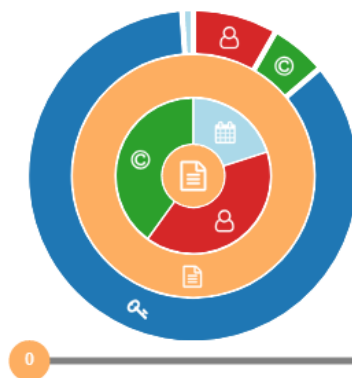
**Figure 4:** Concept Graph – entity to document relevance

The positions of the entities in the document are on the other hand used to display the relevance between entities (Figure 5). The closer two entities occur in a document, the darker the edge will be between them. If the distance exceeds a certain threshold, no relevance edge will be displayed.



**Figure 5:** Concept Graph – entity nodes and relevance edges: entity to entity relevance

*Ring-menu* - Figure 6 shows the improved ring-menu of the graph implemented with the D3 JavaScript library<sup>5</sup>. The goal of the improvement was to also support the selection of node types. In addition, choosing the types of nodes which should be expanded multiple hops away, a long press on one of the sections reveals a slider which allows to select how many nodes on this level should be revealed.

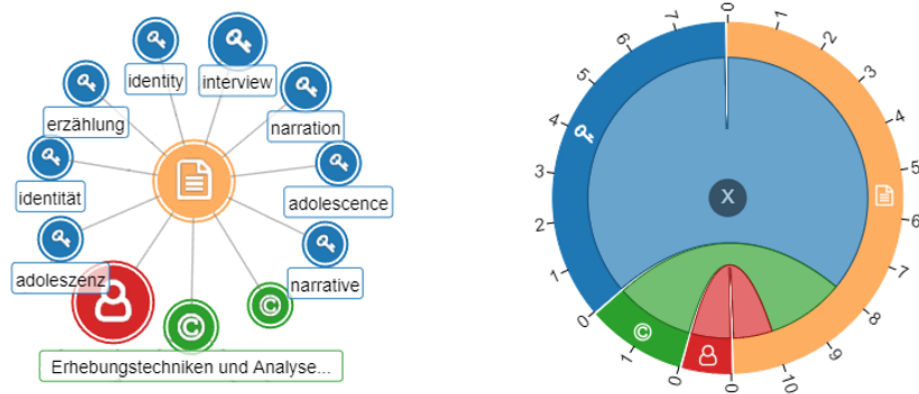


**Figure 6:** Concept Graph – improved ring-menu: holding the left click on one of the ring sections reveals a slider which lets the user choose the number of nodes to show

*Node Aggregation* - The aggregation of nodes is also a new functionality in the graph, which was also developed using D3 and SVG<sup>6</sup>. It allows the user to aggregate a whole subgraph into a single meta-node. Figure 7 shows on the left a subgraph of an expanded node and on the right the aggregated meta-node of this subgraph. The meta-node shows the number of connections between the different node-types in the subgraph. To create a meta-node the user has to first select a subgraph and then click on a button in the graphs toolbar to create it.

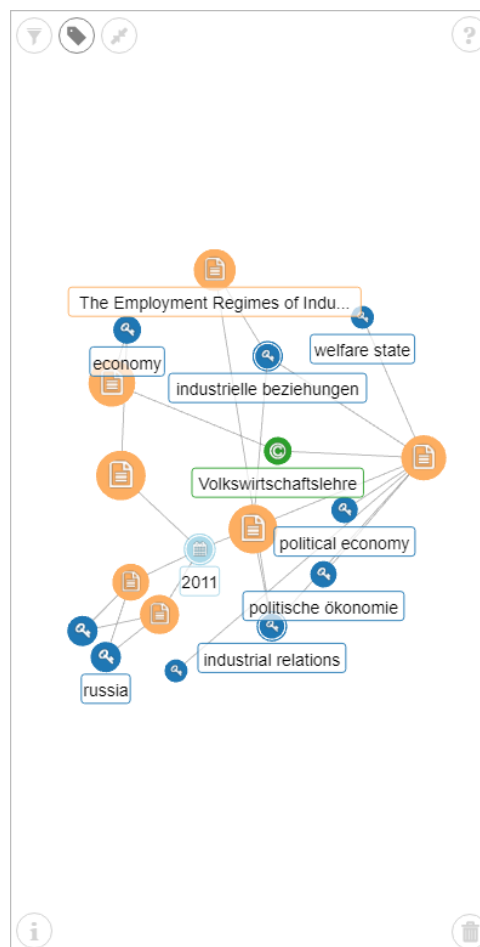
<sup>5</sup><https://d3js.org/>

<sup>6</sup><https://www.w3.org/TR/SVG2/>



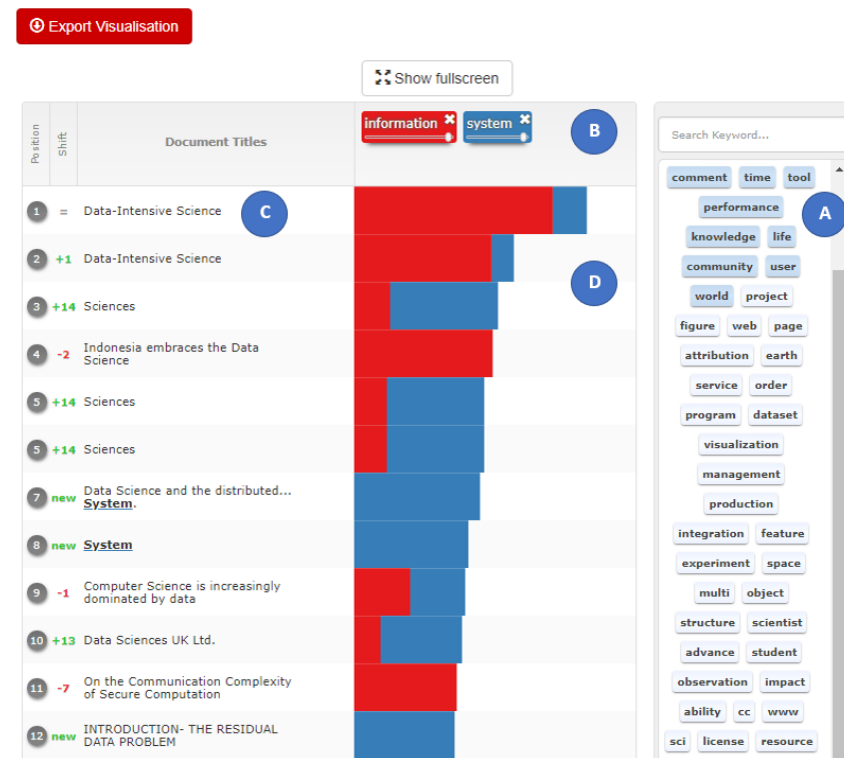
**Figure 7:** Concept Graph – node aggregation: left – a subgraph of an expanded node, right – a meta-node representing the subgraph

*Responsiveness* - Since the Graph Visualization Framework used in the Concept Graph always uses all of the available space (width & height), it was already responsive. Figure 8 shows the Concept Graph on a mobile device. Because the sizes of the buttons and the font-sizes of the labels are set by using `em` as a unit for size, all of the menu elements scale nicely when the meta viewport tag is set in the HTML. However, some functionalities are disabled in the graph when being visualized on a touch device. The disabled functionalities are the ring-menu, the node aggregation functionality and the removal of selected nodes, because they use actions in the desktop version, which are not suitable to be carried out with a finger.



**Figure 8:** Concept Graph – mobile view

**uRank: interest-based result set exploration** Based on the search query, the top 100 retrieved results are displayed as a ranked list. The keywords extracted from the results are presented in the tag cloud in the right sidebar of uRank (Figure 9, point A). By selecting keywords of interest, the results in the result list (Figure 9, point C) get re-ranked in such a way that the results containing the selected keyword move to the top. The ranking view (Figure 9, point D) provides visual feedback of the relevance of the result. It is possible to select multiple keywords and even fine-tune the importance by using the slider under the selected ones (Figure 9, point B). By clicking on a result, a dialog pops up which presents additional information about the retrieved document. Additionally, the user can also export the current view of uRank, with the current search configuration, by clicking on the "Export visualization" button, which initiates a download of a zip-file, containing an image and a report text file.



**Figure 9:** uRank and its components – A) tag cloud, B) tag box, C) result list, D) ranking view

*Integration into the MOVING Platform* - as described in D3.2(Vagliano et al., 2018) uRank was first developed as part of the EEXCESS<sup>7</sup> project. The concept and the implementation were adapted for and integrated into the MOVING platform. Currently, uRank's keyword extraction performs Natural Language Processing (NLP) on the retrieved results in the web browser. This NLP processing includes stemming and the removal of stop words for the English language.

The adaptations for MOVING include the changes needed to accommodate the data model provided by MOVING's search engine, initial handling of German language (mainly stop word elimination), changes in the look & feel of uRank to fit the design of the platform, and numerous small adjustments to the UI because of the limited space.

The uRank visual interface, as it is integrated in the source code of the MOVING Platform, consists of two main parts. The first part is the uRank NPM<sup>8</sup> package, which contains the core uRank functionality. The second is the uRank initializer, which requests the data from MOVING's search engine and initializes uRank with this data. The reason why uRank was created as a standalone package with an initializer was to ensure modularity and reusability.

*Data retrieval* - The necessary data for uRank includes the titles, the abstracts, the document types, the language, and the URLs of the results.

<sup>7</sup><http://eexcess.eu/>

<sup>8</sup><https://www.npmjs.com/>

*Transforming the response* - After getting the response, the uRank initializer passes the results to uRank as an array of objects, specifying what attributes in the results should be used for the keyword extraction and what attributes should be used in the results preview dialog. The attributes that are used for the keyword extraction are the title and the abstract of the result. For the result preview, the language and the URLs to the result are additionally used.

*View initialization* - After receiving the data, uRank first performs a keyword extraction on all results, before counting the occurrences. The keywords, which occur in *uRank*'s tag cloud have to appear at least twice in the title or abstract of a result to be displayed at all. After the tag cloud has been created, the results are initially displayed in the order returned from the search engine.

*View updates* - The result list gets updated through the following three actions:

- Selection of a keyword from the tag cloud
- Changing the weight of a keyword
- Removal of a keyword from the tag box

Every time one of those actions is performed, the result list is re-ranked. For each selected keyword, the relevance of the result is calculated by checking the frequency of this keyword in each of the results, and adjusting it by the set weight of the keyword. If multiple keywords are selected, the results of those calculations are added. The results are then ordered by this relevance.

*Export* - The export functionality in uRank creates an image of the current view of uRank and a report file describing all the applied configurations, which are then compressed and downloaded. All this functionality is implemented on the client-side, speeding up the process and not burdening the server. For generating the image of the visualization, `Html2Canvas`<sup>9</sup> was used. This library renders an HTML element into a canvas that can then be used to extract a picture in a .png format. The generated report file contains the current search settings, the applied filters, and a list of selected keywords with their weights. The content of such a report file might look like this:

```

1 Visualization: uRank
2 Query: data science
3 Active Filters:
4   Content Type
5     Document
6     Full-text
7     Law-regulation
8     Journal Article
9     Book
10    Book Article
11    RDF
12   Language
13     English
14 Selected tags:
15   [ Weight: 1.0, Tag: information ]
16   [ Weight: 1.0, Tag: quality ]
17   [ Weight: 0.8, Tag: education ]

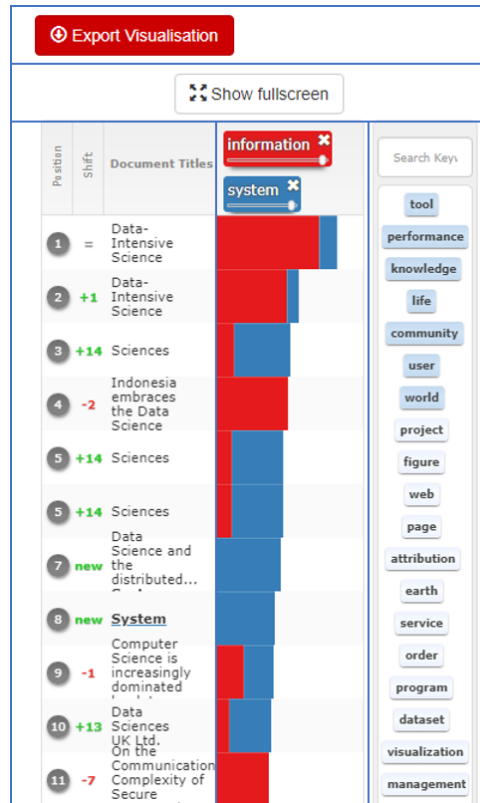
```

*Responsiveness* - uRank, like the other visualizations, was designed in a responsive manner. It uses the available width of the parent container, and has a dynamic height. Figure 10 shows uRank as it is seen on a mobile device, and how the visualization is divided into three rows, each of which uses 100% width of the parent container. The first two rows are only for the "Export Visualisation" and "Show Fullscreen" buttons, while the third row contains the uRank visual interface. This row is divided into three columns, where each column takes a fixed percentage of the horizontal space. Therefore, when changing the size of the container, uRank will automatically adapt to it.

**Top Properties** The Top Properties visualization uses 100 of the most relevant results from the current search query. It shows a bar chart visualization presenting one of the following properties of the available results: Authors, Keywords, Concepts, Sources and Publication Year. The results are ordered according the

<sup>9</sup><https://html2canvas.hertzen.com/>





**Figure 10:** uRank – mobile view: the uRank component in the MOVING Platform is divided into three rows using all of the container width. The third row containing uRank itself is divided into three columns with relative width

most frequent values of the selected property as it can be seen in Figure 11. Only in the case that the publication year gets selected, the sorting order changes so that the years are displayed in a chronological order to make it easier, for the user to see the changes on a year-by-year basis. Clicking on one of the bars shows the results associated with this property in a small dialog. The results in this dialog are sorted in the order provided originally by the search engine. The Top Properties visualization supports also an export functionality, which exports the current view of the visualization with its search configuration, as a zip-file containing an image and a report text file.

*Data retrieval* - To ensure fast switching between the available properties in the visualization, all of the available properties are retrieved with a single request.

*Transforming the response* - Every time the selected property changes, the data has to be mapped into the right format for the visualization. The visualization expects an array of objects, where each object in this array represents a single bar. The objects have the following format:

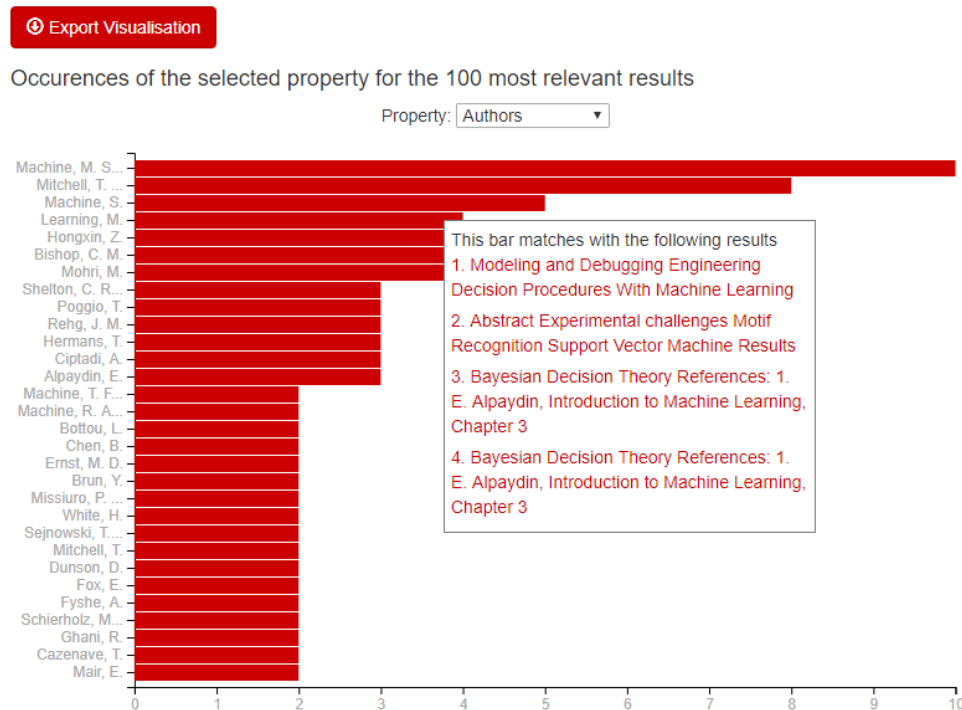
```

1 {
2   "key": <string>,
3   "value": <integer>
4 }
```

Where the "key" attribute represents the label on the y-axis, while the value attribute represents the length of the bar.

In addition to the array of objects representing the individual bars, the visualization also gets provided the raw data from the response. Based on this raw data, every time a user clicks on a bar, the ranked list of results can be generated. This is done by going through all the results from the original response and showing only the results that contain this property.

For creating the bar chart the library D3.js was used.



**Figure 11:** The Top Properties visualization with the dialog showing the result list for a bar

*Export* - The export functionality works in a similar way as the export functionality of uRank. For generating the image of the visualization, Html2Canvas<sup>10</sup> was used. In addition to the query data and the selected visualization, the selected property can be found in the export report, which looks like this:

```
1 Visualization: Top properties
2 Query: data science
3 Selected property: Authors
```

*Responsiveness* - The Top Properties visualization, like the other visualizations, was designed in a responsive way. It uses the available width, but is using a static height. Figure 12 shows the visualization as it is seen on a mobile device, and how it is divided into rows, where each row uses all available width of the container. The export visualization button is on the top, with the title message below, which wraps and is displayed in two lines, and the property dropdown menu stays centered. In the bar chart, the y-axis still has enough space to show the labels of the bars clearly, while the bars fill up the rest of the available width.

**Tag Cloud** The Tag Cloud visualization (Figure 13) retrieves 100 of the most relevant results from the search query and displays them by showing the most frequent keywords that occur in the corresponding titles and abstracts. The displayed keywords are initially sorted by their frequency and can be filtered by occurrence, year or by text. Clicking on one of the keywords shows the results associated with this property. As in the Top Properties visualization, the results are sorted in the order provided originally by the search engine.

*Data retrieval* - The necessary data for the tag cloud includes the titles of the results, the abstracts, the publishing dates and the URLs to the resources.

*Transforming the response* - After getting the response, uRank's keyword extractor is used to analyze the keywords in the titles and abstracts of the retrieved documents. As in uRank, this keyword extractor performs stemming on all the keywords and removes all the stop words before counting the occurrences. Additionally, the keywords which occur in the Tag Cloud have to appear at least twice in the title or abstract of a result to be displayed at all.

<sup>10</sup><https://html2canvas.hertzen.com/>

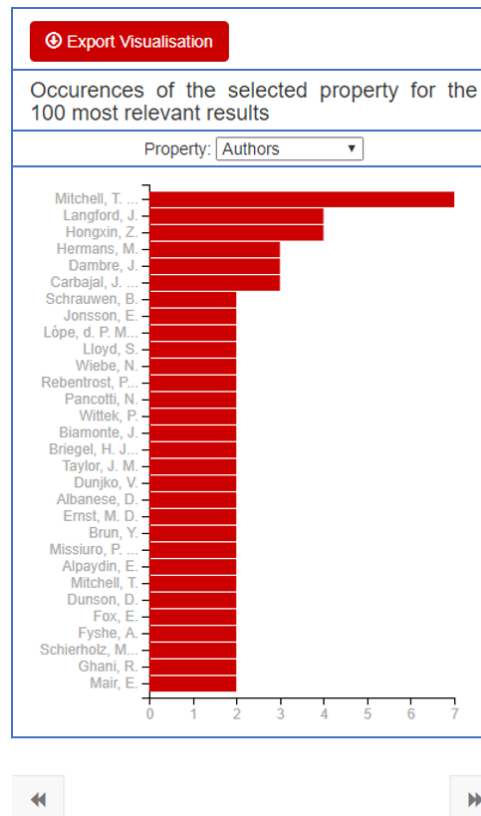


Figure 12: Top Properties – mobile view with the highlighted individual rows that make up the visualization

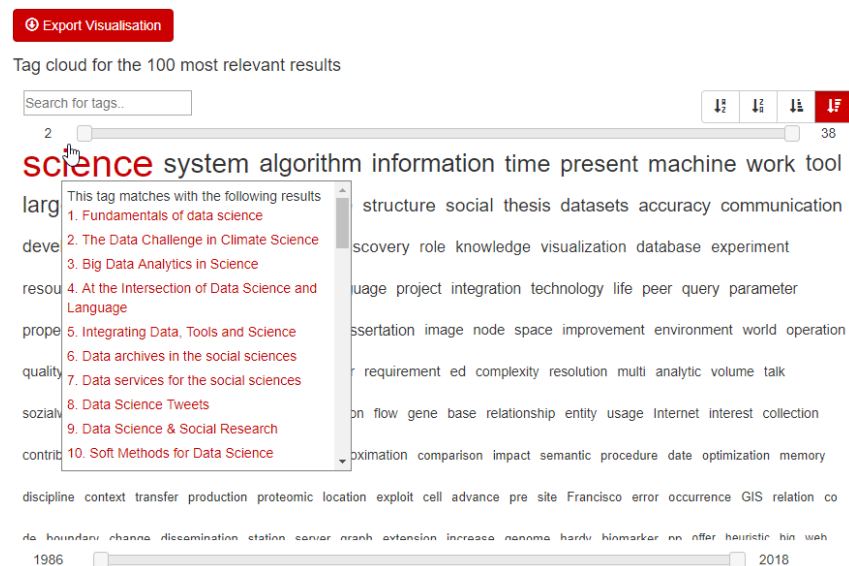


Figure 13: The Tag Cloud visualization with the dialog showing the result list for a keyword

*Creating the visualization* - After the keywords have been extracted, the tag cloud can be built. In addition to the keywords and their occurrence, additional data of the response has to be used to allow the user to get to the documents behind the keywords. In addition to the frequency, each keyword in the tag cloud also contains the information in what results it was found and what the publishing dates of those results are.

*Filtering* - The keywords in the Tag Cloud can be filtered in three ways, namely "by frequency", "by year", and "by text". The keywords can be filtered "by frequency" by adjusting the slider over the tag cloud. All of the keywords not fulfilling the minimum and maximum frequency constraints will be hidden. Filtering

the keywords "by year" actually filters the results that are associated with this keyword. If no results satisfy the year constraints, the keyword gets hidden. Lastly, filtering the keywords "by text" performs a partial string match on all the keywords, thus when entering a text in the filter text input field, all keywords that do not contain this text get hidden.

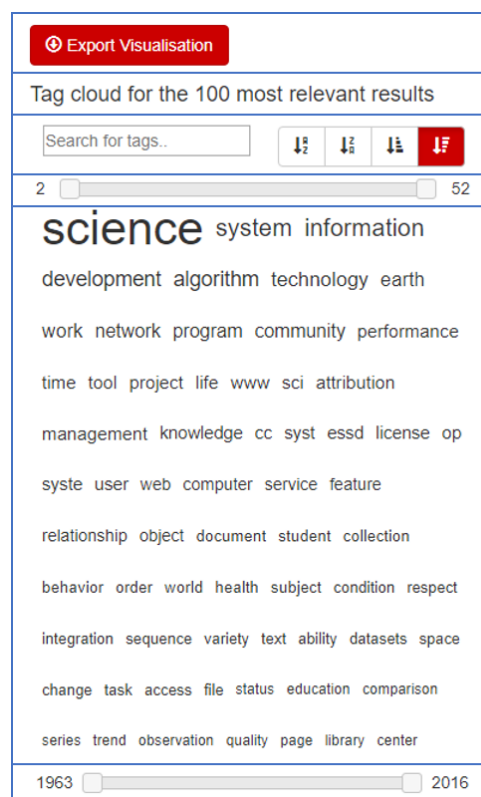
*Export* - For generating the image of the visualization, again, Html2Canvas<sup>11</sup> was used. In addition to the query data and the selected visualization, all of the applied filters can be found in the export report, which might look like this:

```

1 Visualization: Tag cloud
2 Query: "machine learning"
3 Textfilter: machine
4 Minimum frequency: 12
5 Maximum frequency: 100
6 Minimum year: 1990
7 Maximum year: 2017
8 Sorted by: Frequency, Descending

```

*Responsiveness* - In Figure 14, it can be seen how the Tag Cloud visualization looks like on a mobile device, and how it is divided into rows, where each row uses all available width of the container. The export button and visualization title behave in the same way as in the Top Properties visualization. The text filter and the keyword sort buttons come closer when the width of the container shrinks and if they become too close, the keyword sort buttons will wrap into the next row. The two filter sliders adapt to the available width too. While the container with the keywords has a dynamic width, the height is static. Therefore, when the container shrinks, the number of displayed keywords in the Tag Cloud will be reduced. Still all of the keywords are available, the user will only need to apply the necessary filters, or change the sort order.

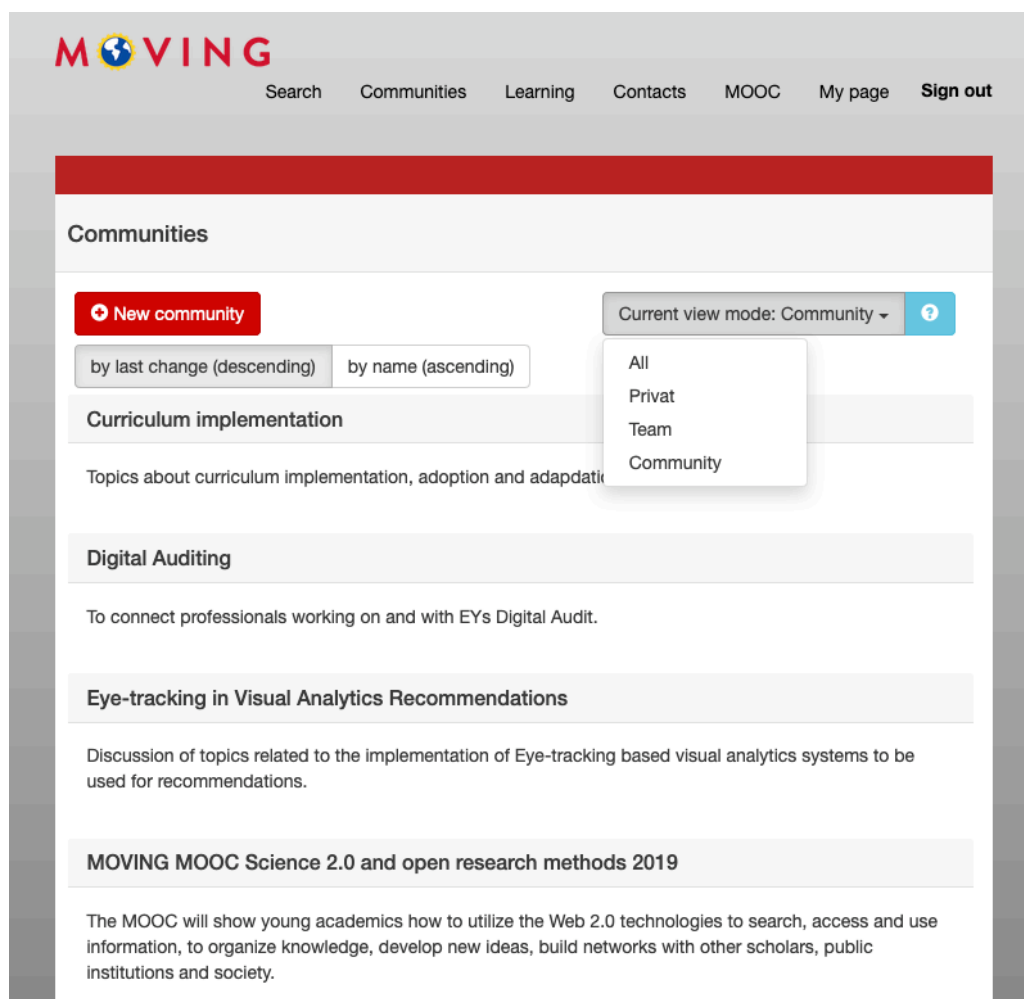


**Figure 14:** Tag Cloud – mobile view with the highlighted individual rows that make up the visualization

<sup>11</sup><https://html2canvas.hertzen.com/>

## 3.2 Communities

Open collaboration and communication are the foundations of open innovation and open science. With MOVING communities users are offered a powerful tool to organize group collaboration and communities of practice on the MOVING platform (see Figure 15). MOVING communities are part of the working environment of the platform and offer a range of social technologies and tools for knowledge and information management including wikis, forums, blog functions and group news. MOVING communities are based on the project management tools and technologies of the *eScience* platform that was taken as the foundation for the development of MOVING. For further information see Deliverable 4.1 "Definition of platform architecture and software development configuration" (Gottfried, Grunewald, et al., 2017). The existing *eScience* modules, which enabled the cooperation in closed teams of researchers, were adapted to the goals of MOVING to provide an open innovation environment and foster open collaboration and communication as well as the exchange of knowledge between users of the platform.



**Figure 15:** MOVING communities

Registered users who want to create a new community are offered different options (see Figure 16): firstly, users can create *public* communities that are visible to everyone in the MOVING platform and can be accessed and edited by anyone interested in the topic of the community. And secondly, users who want to organize specific groups like project teams or research groups can create *private* communities that users have to join first before they can access and edit content. Private communities are not visible to other users but can be shared to collaborators via email.

The MOVING CK Editor<sup>12</sup> allows the creation of formatted text and the integration of multimedia content in HTML pages that are created by the users in the MOVING communities. Videos, pictures, GIFs or documents

<sup>12</sup><https://ckeditor.com/>, last accessed on 6/3/2019

**MOVING** Search Communities Learning Contacts MOOC My page Sign out

**Communities**

**New community**

**General**

**Name\***

**Identifier\***

Length between 1 and 100 characters. Only lower case letters (a-z), numbers, dashes and underscores are allowed. Once saved, the Identifier cannot be changed.

**Public\*** ☒ Visible for everyone in the community

**Subcommunity of\***

**Summary\***

**Description**

Source

Styles  **B** *I* U ~~S~~ <sup>x<sub>e</sub></sup> <sup>x<sub>e</sub></sup> *I<sub>x</sub>*

body

**Homepage**

**Course** ☐ Enable smaller course features

**Modules**

☒ News ☒ Documents ☒ Wiki ☒ Forums ☒ Badges

**Figure 16:** Create a new community

as well as social media content from i.e. Twitter<sup>13</sup> and YouTube<sup>14</sup> can be easily integrated via an *iframe* (inline frame). Features like the *accordion* and the option to include expandable items make it easy to structure content in the page. It is an WYSIWYG-editor (*What You See Is What You Get*) hence making it easy and comfortable even for users that are not familiar with HTML to create and edit web-based content within MOVING communities. The wiki module is useful to create and collaboratively manage large knowledge repositories with an community. The forum module provides space for open communication and the exchange of information - precondition for open innovation processes. The forum module contains a user rating functionality that allows the community to publicly rate the content of individual forum entries. Users can vote posts and replies up and down, based on the quality of the contribution. The highest rated input is highlighted to help users in finding the best response in a thread. Furthermore, the summarized score for all received votes is shown on each user profile. The ranking functionality is helpful in the self-organisation of communities and peer assessment of user generated content. Community administrators can also choose to assign badges to their community to reward users, or motivate them to get actively engaged. Badges can be assigned automatically

<sup>13</sup><https://twitter.com/>, last accessed on 6/3/2019

<sup>14</sup><https://www.youtube.com/>, last accessed on 6/3/2019

or manually.

The ease of user-generated content creation and integration combined with the social features of MOVING communities open up a wide range of possible applications for MOVING users. Users can organize group work in small project teams, or create open communities around scientific or technical topics to discuss research or ask questions to an expert community. MOVING communities can be organized as an open innovation tool but also as a learning management system as the following example shows. One practical application of MOVING communities is the four-week MOVING MOOC (massive open online course) *Science 2.0 and open research methods* that was organized on the MOVING platform in M32/33 and M34/35 (see Figure 17). The MOOC is organized on the platform as a private *team* community, so that participants have to register to gain access to the learning materials and the forums the course. For each week of the MOOC we created a subcommunity containing learning materials in different media formats as well as weekly assignments (see Figure 18). The forums were used to organize group communication and allow users to share their assignment results. A wiki was created and contained additional information about the course, its learning goals and how to proceed, as well as technical information about the use of the editor or about the MOOC badges that users can earn for participating and completing the course (see Figures 19 and 20). The badges that users earn will be displayed in their user profile *My page* along with their profile information and contact details (profile picture, science field, skills, hometown, institution, email, ORCID<sup>15</sup>). The dashboard also displays the membership in communities and an individual stream of recent user activities (see Figure 21). For a detailed description about badges, see Section 3.2.1.

The screenshot shows the MOVING MOOC community page. At the top, the MOVING logo is on the left, and 'Register' and 'Sign In' buttons are on the right. Below this is a banner image of a keyboard with a central graphic that says 'MOVING MOOC SCIENCE 2.0 AND OPEN RESEARCH METHODS'. Under the banner, a green bar contains course information: 'Course Information (#MoMoScience20)', 'Start date: January 21, 2019', 'Effort: 2-4 hrs/week', 'Language: English', 'Duration: 4 weeks', 'Level: Beginner', 'Credentials: Certificate of participation', and an 'Enrol now' button. Below this, the page is divided into two columns. The left column has a section 'What is this course about?' with a paragraph about Science 2.0, a 'Keywords' section listing Science 2.0, Open Science, Open Access, Open Data, OER, Web 2.0 technologies, Creative Commons, open peer review, Altmetrics, and social media, and a 'Who is this course for?' section stating it is for PhDs, Post-docs, and students at an advanced level. The right column features a video player titled 'MOVING MOOC Science 2.0 Teaser' with a 'Watch later' and 'Share' button. The video player shows a thumbnail with icons for 'Science 2.0 & open research', 'Tools & technologies', and 'open & collaborative'. Below the video player is a section titled 'Organizers'.

Figure 17: MOVING MOOC community

<sup>15</sup><https://orcid.org/>, last accessed on 6/3/2019

## Week 4: Make your research open!

### 11 - 17 February 2019

Hello and welcome to the fourth and last week of the MOOC. In this week, you will learn about the advantages of openness and transparency in your research process and get relevant information about Open Access Publishing and Creative Commons licensing. To wrap up the MOOC we have illustrated an Open Science workflow in a short video. In this weeks assignment you are going to think about your research process and with which tools you're plan to make it more open.

#### Week 4 Forum

Go here to discuss and ask questions about week 4!

#### Learning goals

After this week you will...

- have an overview about Open Access publishing
- know how to license your work using Creative Commons-licensing
- understand the Open Science research workflow

#### Open Access

##### Infographic: Opening up the research workflow (approx. 5min)

In week 2 you were already introduced into opening up your research workflow, focusing on the first three steps. Now we will focus on the fourth and fifth step of the research workflow and we'll give you some tips on which platforms you can perform these steps:

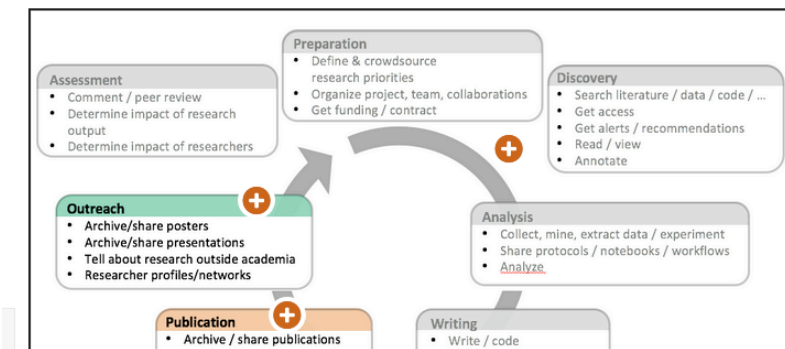


Figure 18: MOVING MOOC - week 4



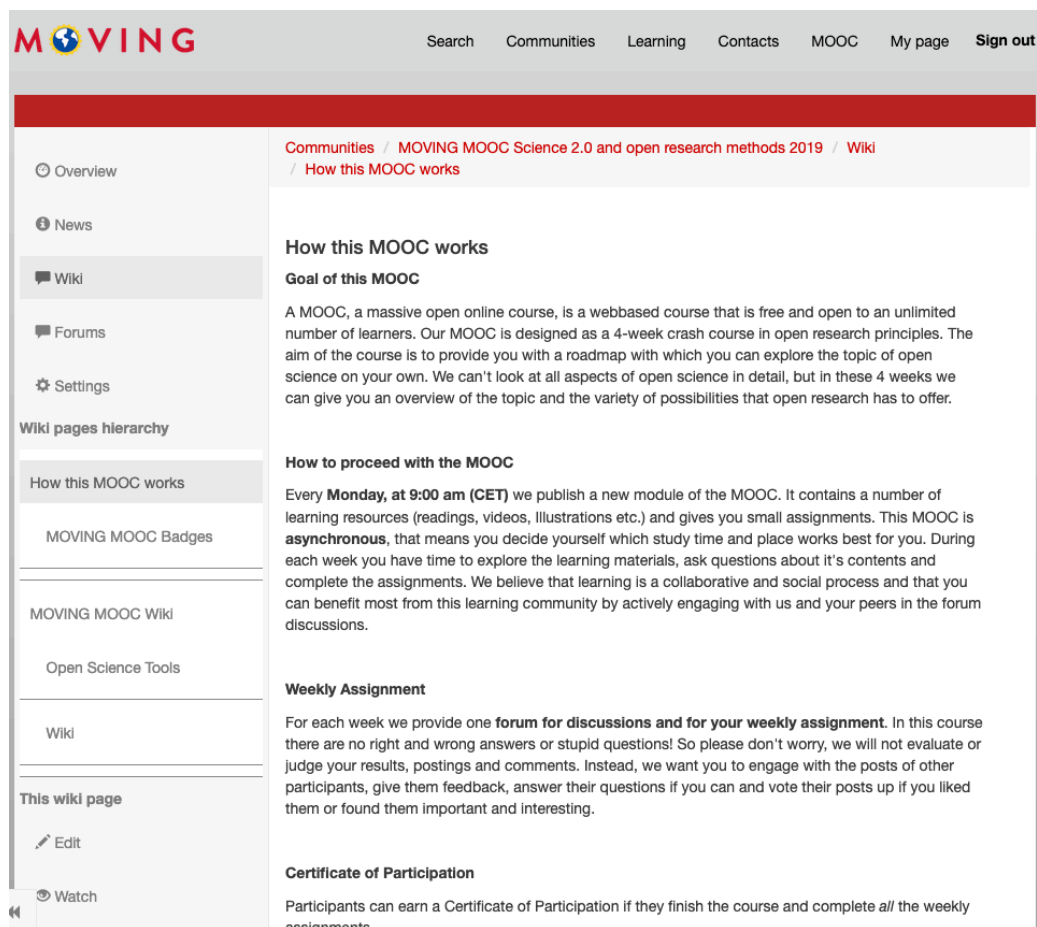


Figure 19: MOVING MOOC - wiki

### Which Badges can I get in the MOOC and what do I have to do?

In the MOVING-MOOC you can get the following Badges:

#### MOVING



The MOVING-Badge is one badge for all MOVING MOOC participants. All you have to do is to enroll to the MOVING-MOOC. By getting the MOVING-Badge you show others, that you are interested in Open Science and MOOCs. If you like to promote the MOVING-MOOC backpack the MOVING-badge and show others this great opportunity to learn and discuss about open Science.

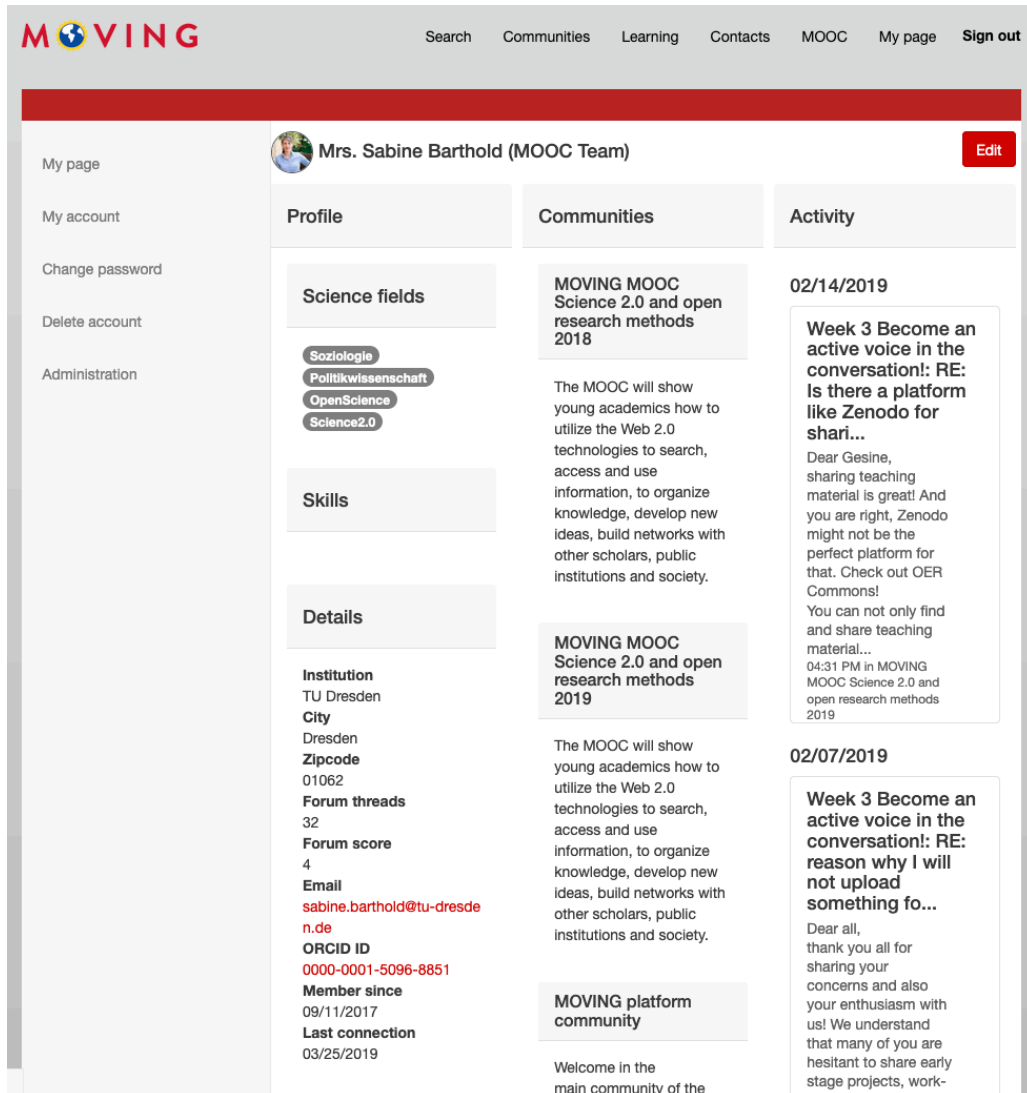
#### OPEN SCIENCE AFICIONADO



The Open Science Aficionado Badge shows that you:

- understand the concept of Open Science and know the fundamentals of open research methods
- know how you can use open research methods for scientific collaboration
- understand why and how social technologies improve open scholarly communication and help you to reach a wider audience for your research
- know the open science research workflow and how it relies on the use of digital and social technologies

**Figure 20:** MOVING MOOC - badges



The screenshot displays the MOVING user dashboard for Mrs. Sabine Barthold (MOOC Team). The interface features a top navigation bar with links for Search, Communities, Learning, Contacts, MOOC, My page, and Sign out. A left sidebar contains links for My page, My account, Change password, Delete account, and Administration. The main content area is divided into three columns: Profile, Communities, and Activity.

**Profile Column:**

- Science fields:** Soziologie, Politikwissenschaft, OpenScience, Science2.0.
- Skills:**
- Details:**
  - Institution:** TU Dresden
  - City:** Dresden
  - Zipcode:** 01062
  - Forum threads:** 32
  - Forum score:** 4
  - Email:** sabine.barthold@tu-dresden.de
  - ORCID ID:** 0000-0001-5096-8851
  - Member since:** 09/11/2017
  - Last connection:** 03/25/2019

**Communities Column:**

- MOVING MOOC Science 2.0 and open research methods 2018:** The MOOC will show young academics how to utilize the Web 2.0 technologies to search, access and use information, to organize knowledge, develop new ideas, build networks with other scholars, public institutions and society.
- MOVING MOOC Science 2.0 and open research methods 2019:** The MOOC will show young academics how to utilize the Web 2.0 technologies to search, access and use information, to organize knowledge, develop new ideas, build networks with other scholars, public institutions and society.
- MOVING platform community:** Welcome in the main community of the

**Activity Column:**

- 02/14/2019:** Week 3 Become an active voice in the conversation!: RE: Is there a platform like Zenodo for shari...  
Dear Gesine, sharing teaching material is great! And you are right, Zenodo might not be the perfect platform for that. Check out OER Commons! You can not only find and share teaching material...  
04:31 PM in MOVING MOOC Science 2.0 and open research methods 2019
- 02/07/2019:** Week 3 Become an active voice in the conversation!: RE: reason why I will not upload something fo...  
Dear all, thank you all for sharing your concerns and also your enthusiasm with us! We understand that many of you are hesitant to share early stage projects, work-

Figure 21: MOVING user dashboard

### 3.2.1 Badges

Badges are used to showcase one's achievement like participating in social groups or events, publishing in a scientific field or succeeding in a sport event. As badges are already a well known and accepted concept in the real world, they also have found its way in the digital world in many different online communities. Albeit there are many online communities that have some kind of system to show one's achievements (often accompanied with a visual representation and a explanatory text), the display and usage of those badges suffer from different problems. Key information about the earner, the issue date, the achievement itself and its requirements are often missing or it is not easily possible to verify the accreditation of the earner who of a badge.

The *Open Badges* standard<sup>16</sup>, initially proposed by the Mozilla Foundation, addresses the aforementioned problems and allows other features like being able to transfer badges to a so-called Backpack to showcase badges from different places at a single one. Figure 22 shows the overall workflow demonstrating the possibilities of using Open Badges<sup>17</sup>. This workflow illustrates a life-cycle, with different roles and interactions in the context of badges. *Badge Classes* describe the badge itself, which can be awarded to so *Earners*. The resulting *Assertion* (stating who has been awarded what) can be hosted, shown and verified by other parties. Note, that a system can play *multiple* roles, for instance the MOVING platform can create, award, store and display its own badges. A backpack-supporting system, such as Badgr<sup>18</sup>, on the other hand allows to store and display badges from *any* system.

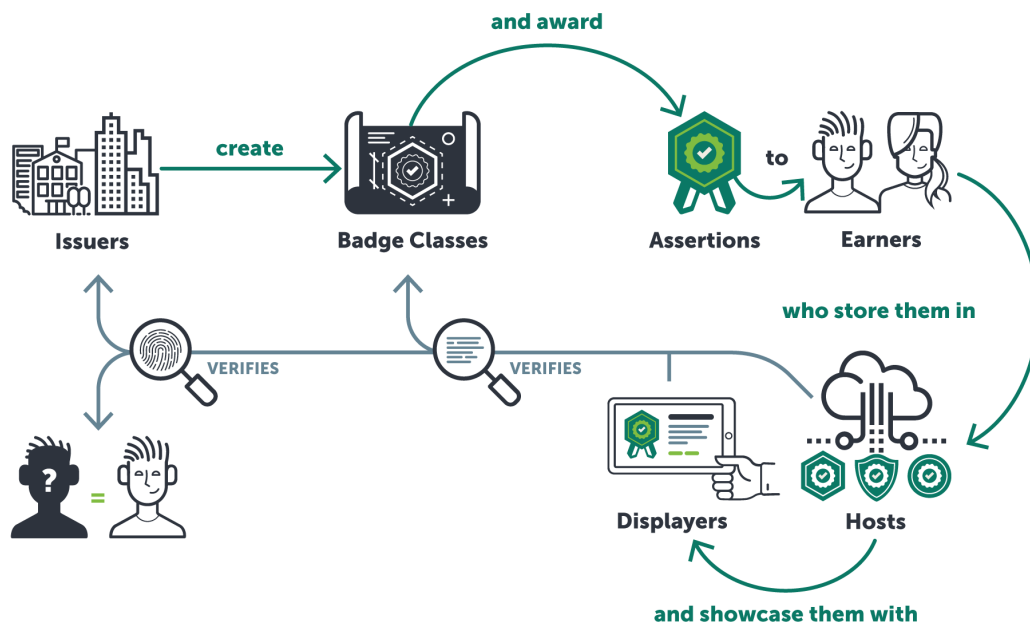


Figure 22: Overview of the Open Badge workflow

The MOVING platform implements the latest revision of this standard (i.e. v2.0, at the time of writing) to recognize primarily learning success of individual users and empowering the users to share this success with others. After enabling the Badges Module in the community settings menu, the community's administrator can create own badges after providing the following information:

- a title,
- an image (selected from a given collection),
- a general description about the badge,
- a human-readable description of the criteria,
- whether or not there should be a mail notification and
- one of the methods that determines, how a badge is issued to the user.

<sup>16</sup><https://openbadgespec.org/>, last accessed on 10/03/2019

<sup>17</sup>Source: <https://www.imsglobal.org/sites/default/files/Badges/OBv2p0Final/impl/index.html>, last accessed on 10/03/2019

<sup>18</sup><https://badgr.io/>, last accessed on 10/03/2019

There are three kinds of methods: manual, semi-automatic and automatic ones. With the manual method any user can receive a badge by manual selection, whereas with semi-automatic methods the eligible users are filtered out and with automatic methods badges are issued completely without administrative effort. All these methods are defined within the code base of MOVING and can be selected by the administrator. So if dealing with user base size comparable to a MOOC with potentially thousands of participants, it is advised to implement a (semi-)automatic method to drastically ease the issuing process.

When a badge has been issued, the recipient might be notified by mail and is now able to inspect the newly awarded badge at two different places: The first one is the profile page of the user, which provides a list of all badges received in any of the communities of the MOVING platform with only basic information (title of badge, date issued, community, etc.). For detailed information there is a link on the profile page, that leads to a summary page for the issued badge. The summary page serves multiple purposes: It describes how this user acquired this badge making it transparent for outstander in a readable way. This page itself can be shared in various social media channels for greater exposure. It also teaches the recipient how Open Badges — just like the one currently being presented — can be added to a Backpack service and allows to perform the verification process in accordance to the Open Badges standard, in case an individual claims to have earned this badge.

Following the Open Badges standard for implementing badges does not only dictate how the data model is being designed, but also enforces which *endpoints* have to be provided by a badge issuing platform such as MOVING. These endpoints are meant to be consumed by other platforms, which would like to programmatically verify the badge they have been presented.

The following endpoints have been implemented:

**Badge Class:** GET /communities/:id/badges/:badge\_id::json

This endpoint returns the metadata of a Badge Class (:badge\_id) of a given community (:id). An example response is given in listing 2.

**Badge Assertion:** GET /users/:id/badges/:badge\_issue\_id::json

This endpoint returns an assertion about an issued badge (receiver, issuer, badge, date of issue). The assertion is retrieved via the user (:id) and the unique issue number (:badge\_issue\_id). An example response is given in listing 3.

**Issuer Profile:** GET /communities/:id/badges\_issuer::json

This endpoints returns the metadata of the issuer of badges of a given community (:id). An example response is given in listing 4.

```
1 {
2   "@context": "https://w3id.org/openbadges/v2",
3   "id": "https://moving.mz.tu-dresden.de/communities/moving_mooc_2019/badges/1:json",
4   "type": "BadgeClass",
5   "name": "Movling",
6   "description": "The MOVLING-Badge is a badge for all MOVING MOOC participants. You got this Badge because you have
7     ↳ enrolled in the MOVING MOOC. With the MOVLING-Badge you show others that you are interested in Open
8     ↳ Science. If you want to promote the MOVING MOOC outside the MOVING platform, add the MOVLING-Badge to
9     ↳ your backpack and show others this great opportunity to learn more about Open Science.",
10  "image": "https://moving.mz.tu-dresden.de/assets/badges/badge1.png",
11  "issuer": "https://moving.mz.tu-dresden.de/communities/moving_mooc_2019/badges_issuer:json",
12  "criteria": {
13    "narrative": "Enrolling to the MOOC"
14  }
15 }
```

**Listing 2:** JSON response of Badge Class metadata

```

1 {
2   "@context": "https://w3id.org/openbadges/v2",
3   "id": "https://moving.mz.tu-dresden.de/users/34/badges/275.json",
4   "type": "Assertion",
5   "recipient": {
6     "type": "email",
7     "hashed": true,
8     "salt": "2860395236ac55aaa7e45e7b84a62fd7",
9     "identity": "sha256$6158b3810236129c1fe4135bc0279d5708c82fc7274930167e16bc13296a0f81"
10  },
11  "evidence": "https://moving.mz.tu-dresden.de/users/34/badges/275",
12  "badge": "https://moving.mz.tu-dresden.de/communities/moving_mooc_2019/badges/1.json",
13  "verification": {
14    "type": "hosted"
15  },
16  "issuedOn": "2019-02-01T09:03:47Z"
17 }

```

**Listing 3:** JSON response for a Badge Assertion

```

1 {
2   "@context": "https://w3id.org/openbadges/v2",
3   "id": "https://moving.mz.tu-dresden.de/communities/moving_mooc_2019/badges_issuer.json",
4   "type": "Profile",
5   "name": "MOVING MOOC Science 2.0 and open research methods 2019",
6   "url": "https://moving.mz.tu-dresden.de/communities/moving_mooc_2019",
7   "description": "The community \"MOVING MOOC Science 2.0 and open research methods 2019\" is part of the MOVING
8     ↪ platform, which allows individuals to gather, learn and share information with others. For further information please
9     ↪ refer to the community page.",
10  "verification": {
11    "type": "hosted"
12  }
13 }

```

**Listing 4:** JSON response for a Issuer Profile

With all the aforementioned endpoints, it is now possible to verify if a user is the rightful holder of a badge<sup>19</sup>:

The identity section of a badge Assertion includes the Recipient's email address, a hash of the Recipient's email address, or a salted hash of the Recipient's email address. A Displayer can compare this value to the expected email(s) of the Recipient claiming the badge to establish authenticity. This facilitates verification, while preventing the need to store the earner's email explicitly within the badge Assertion, providing protection against routine forms of unauthorized access.

As every community within the MOVING platform is able to use the badge system, the community for the MOVING curriculum named 'MOVING MOOC Science 2.0 and open research methods 2019' has heavily applied badges with more than 300 Badge Assertions for different tasks. To make users familiar with the concept of badges and the Open Badges ecosystem, we have devised a beginner badge called 'MOVING', that is awarded to every user, that joins the MOVING curriculum. The figure 23 illustrates the summary page for a MOVING badge.

For an in-depth analysis about the exploitation of badges and other community instruments please refer to Deliverable 2.3 (Günther et al., 2019).

<sup>19</sup>Source: Open Badges Developer Guide, <https://openbadges.org/developers/#verification>, last accessed on 10/03/2019

## Movling

### Description

The MOVLING-Badge is a badge for all MOVING MOOC participants. You got this Badge because you have enrolled in the MOVING MOOC. With the MOVLING-Badge you show others that you are interested in Open Science. If you want to promote the MOVING MOOC outside the MOVING platform, add the MOVLING-Badge to your backpack and show others this great opportunity to learn more about Open Science.

### Requirements for earning this badge

Enrolling to the MOOC

### Recipient

Mr. Paul Grunewald

### Date of approval

02/01/2019

### Issuer of this badge

Administrators of the community named **MOVING MOOC**  
**Science 2.0 and open research methods 2019**



**Figure 23:** Assertion of a MOVLING badge

### 3.3 Learning environment

MOVING offers a unique combination of working and training features in one platform. Heart of the training programs is the MOVING learning environment. Here, all the learning content is organized and directly accessible to the users. The landing page (see Figure 24) gives an overview of the learning materials including the *platform demo videos* and video tutorials, the *Learning tracks for information literacy 2.0* and the MOVING MOOC *Science 2.0 and open research methods*. The platform demos are videos hosted by JSI on [videlectures.net](http://videlectures.net) and are embedded in the learning environment so users can learn about the different platform features and technologies developed within the MOVING project. Users can improve their data and information literacy as well as digital competences through the MOVING *Learning Tracks for Information Literacy 2.0* (see Figure 25).

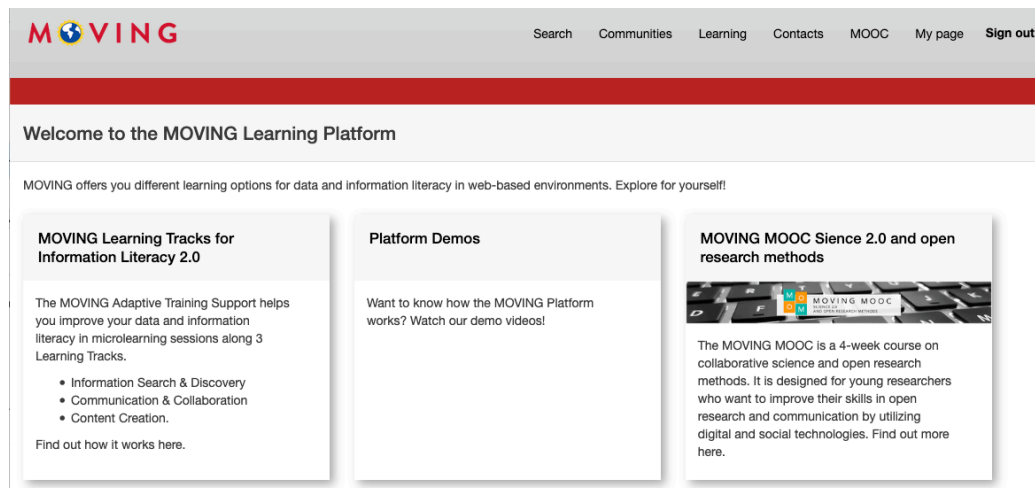
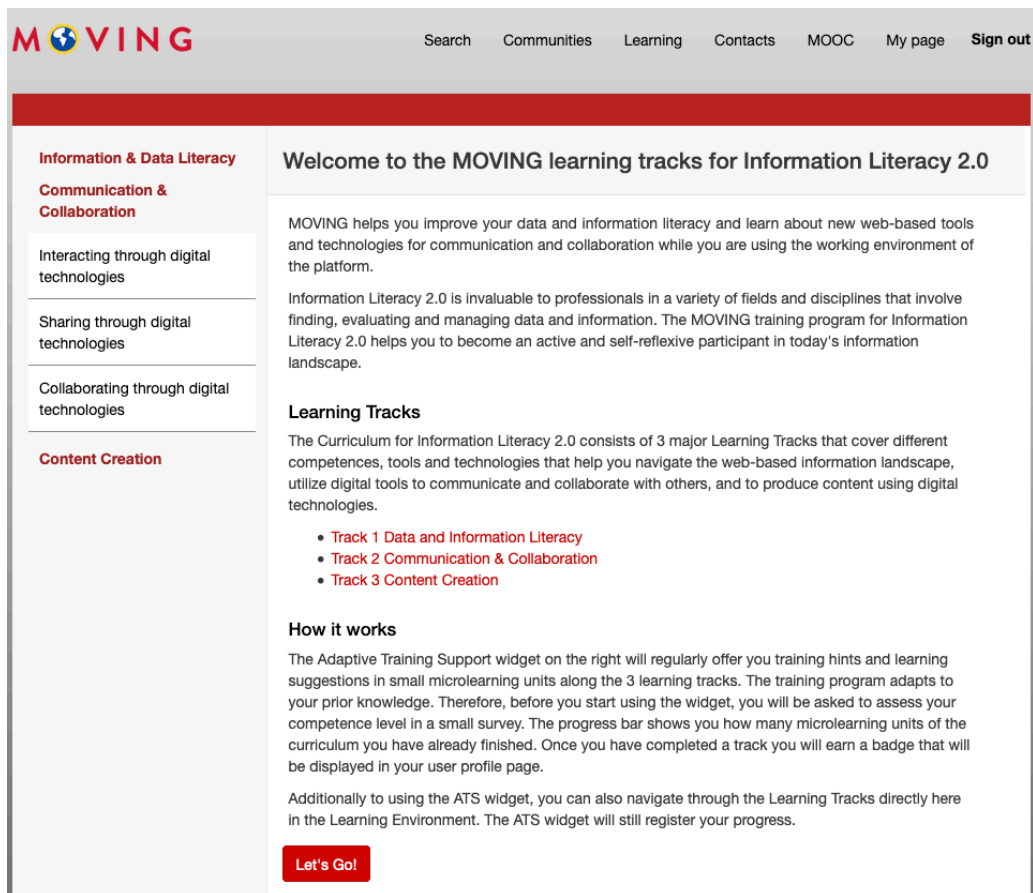


Figure 24: MOVING learning environment

The *Curriculum reflection* widget of the Adaptive Training Support (ATS) (see Section 5.2) in the working environment of the platform regularly offers learning prompts to users based on their prior knowledge and guides them to the microlearning sessions in the learning environment. These small sessions are called *learning cards* and are based on the MOVING curriculum described comprehensively in the Deliverables D2.2 and D2.3 (Günther et al., 2018; Günther et al., 2019) and contain learning material in different media formats such as text, video and infographics (see Figure 26). The MOVING MOOC *Science 2.0 and open research methods* is a four week online course hosted on the MOVING platform designed to give young scholars from a wide range of disciplines a comprehensive introduction to open science methods and open research workflows (Günther et al., 2019) and was implemented in the communities environment of the MOVING platform (see Section 3.2).





The screenshot shows the MOVING platform interface. At the top is a navigation bar with the MOVING logo and links for Search, Communities, Learning, Contacts, MOOC, My page, and Sign out. Below this is a red header bar. The main content area is divided into a left sidebar and a main panel. The sidebar has a red header 'Information & Data Literacy' and 'Communication & Collaboration'. It lists three categories: 'Interacting through digital technologies', 'Sharing through digital technologies', and 'Collaborating through digital technologies'. Below these is a red header 'Content Creation'. The main panel has a red header 'Welcome to the MOVING learning tracks for Information Literacy 2.0'. It contains a paragraph about MOVING's purpose, a paragraph about Information Literacy 2.0's value, a section 'Learning Tracks' with a list of three tracks, a section 'How it works' with a paragraph about the Adaptive Training Support widget, and a red button 'Let's Go!'.

**Information & Data Literacy**  
**Communication & Collaboration**

Interacting through digital technologies

Sharing through digital technologies

Collaborating through digital technologies

**Content Creation**

## Welcome to the MOVING learning tracks for Information Literacy 2.0

MOVING helps you improve your data and information literacy and learn about new web-based tools and technologies for communication and collaboration while you are using the working environment of the platform.

Information Literacy 2.0 is invaluable to professionals in a variety of fields and disciplines that involve finding, evaluating and managing data and information. The MOVING training program for Information Literacy 2.0 helps you to become an active and self-reflexive participant in today's information landscape.

### Learning Tracks

The Curriculum for Information Literacy 2.0 consists of 3 major Learning Tracks that cover different competences, tools and technologies that help you navigate the web-based information landscape, utilize digital tools to communicate and collaborate with others, and to produce content using digital technologies.

- **Track 1 Data and Information Literacy**
- **Track 2 Communication & Collaboration**
- **Track 3 Content Creation**

### How it works

The Adaptive Training Support widget on the right will regularly offer you training hints and learning suggestions in small microlearning units along the 3 learning tracks. The training program adapts to your prior knowledge. Therefore, before you start using the widget, you will be asked to assess your competence level in a small survey. The progress bar shows you how many microlearning units of the curriculum you have already finished. Once you have completed a track you will earn a badge that will be displayed in your user profile page.

Additionally to using the ATS widget, you can also navigate through the Learning Tracks directly here in the Learning Environment. The ATS widget will still register your progress.

**Let's Go!**

Figure 25: Learning tracks for information literacy start page

Lesson 1 out of 7


**Reliability of information**

Evaluating information > Information & Data Literacy

Science-related resources, such as scholarly articles, books, etc., are subject to scientific quality criteria and research ethical principles.

Articles published in scientific journals are always peer reviewed, i.e. their authorship, formats, presentation methods and intentions are checked and evaluated by peers, that means by other experts, before publication.

In addition to scientific publications, there are also other digital sources of information, mainly websites and articles in social media and social networks. For these contents other evaluation criteria apply. They require a certain degree of personal information literacy. You will find out how you can evaluate the reliability of these sources in a later lesson.



What is Peer Review?

Watch later Share

If accepted, the journal then sends the article to experts in the subject area.

Image by Rhoda Baer, National Institutes of Health

Previous Lesson

Next Lesson

**Figure 26:** Microlearning session card

### 3.4 Responsive design

As already described in the previous Deliverables D4.1 (Gottfried, Grunewald, et al., 2017) and D4.2 (Gottfried, Pournaras, et al., 2017), we are using the Bootstrap framework<sup>20</sup> in order to implement the responsive design making the platform easily and efficiently accessible via multiple types of devices, from desktop web clients to smart phones and tablets. For implementing the initial version of the responsive design, the MOVING platform mockups were used as a starting point. They were developed with the Balsamiq Software<sup>21</sup> and described in detail in the Section 7 of Deliverable D1.1 (Bienia et al., 2017). Consequently, and by continuing using the Bootstrap framework we implemented the responsive design in all screens of the MOVING platform by using the implemented environment of the MOVING web application.

We continued to organise the different functionalities in the same way in all the views of the application. More specifically:

1. Navigation button *Search* which follows the three-column view as already described in subsection 3.3 of the previous Deliverable D4.2, adapts the single-column view in the mobile and tablet device.
  - (a) For *Research* results we implemented the dropdown menus by using multiple checkboxes for each single facet. Especially, for the Year of publication facet we implemented a canvas tag that changes according to the size and capabilities of the device.
  - (b) For *Funding* and *Training materials* results the responsive implementation is similar to the *Research* results.
2. Navigation buttons *Communities*, *Learning*, *Contacts*, *MOOC* and *My page* which follow the two-column view also adapt the single-column view.

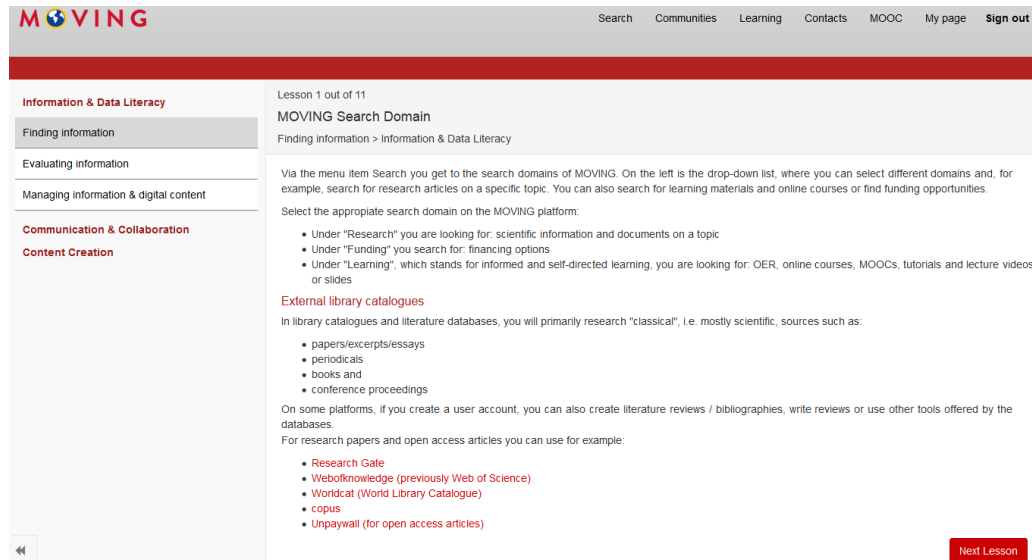
Moreover, the widgets in the Adaptive Training Support, which is located in the right sidebar of the MOVING platform, always uses all of the available width to display their content. Since the MOVING platform itself is responsive, when resizing the browser window, the widgets of the ATS adapt as well. Finally, the various visualizations of the MOVING platform that are displayed in the search results of the navigation tab *Search* (i.e. the Concept Graph, uRank, Tag Cloud and Top Properties) act in a similar way. Given that they are integrated in the central column of the *Search* tab, they also use the available width of their parent container. Therefore, when resizing the page, the container will resize too and the visualizations will adapt to it. An unresolved issue is still present in the Concept Graph, which has compatibility issues with browsers other than Google Chrome (desktop and mobile), due to recent updates of their viewing engines.

Finally, it is worth mentioning that we employed the responsive design for all newly developed functions and pages of the MOVING platform (e.g in all pages in the navigation tab *Learning* as showed in Figure 27) and we introduced responsiveness for the pre-existing platform functions of the e-Science platform used in the MOVING platform (e.g in the administrative page communities in Figure 28).

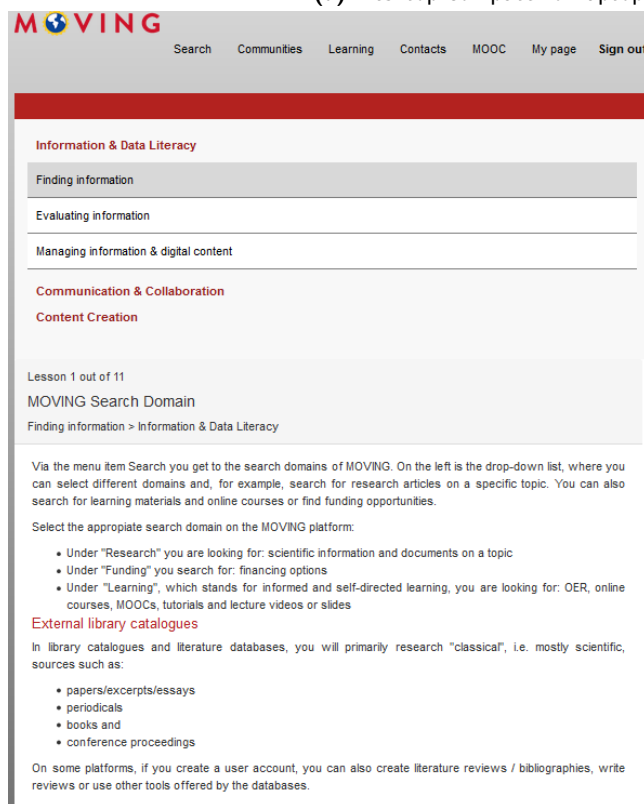
In overall, all MOVING platform views adapt to different screen sizes, automatically changing the layout according to the size and capabilities of the device. For example, on a desktop computer screen, the users see the content in a three-column view, as depicted in Figure 29(a), whereas on a mobile phone the content is presented in a single-column view, and on a tablet the same content is displayed with the menus on the top of the screen, as shown in Figure 29(b) and 29(c).

<sup>20</sup><http://getbootstrap.com>, last accessed on 8/3/2019

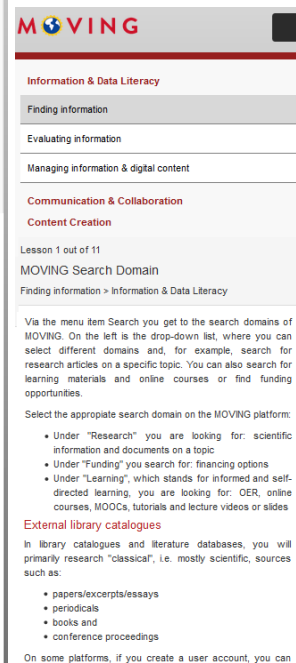
<sup>21</sup><https://balsamiq.com>, last accessed on 8/3/2019



(a) Desktop computer or laptop

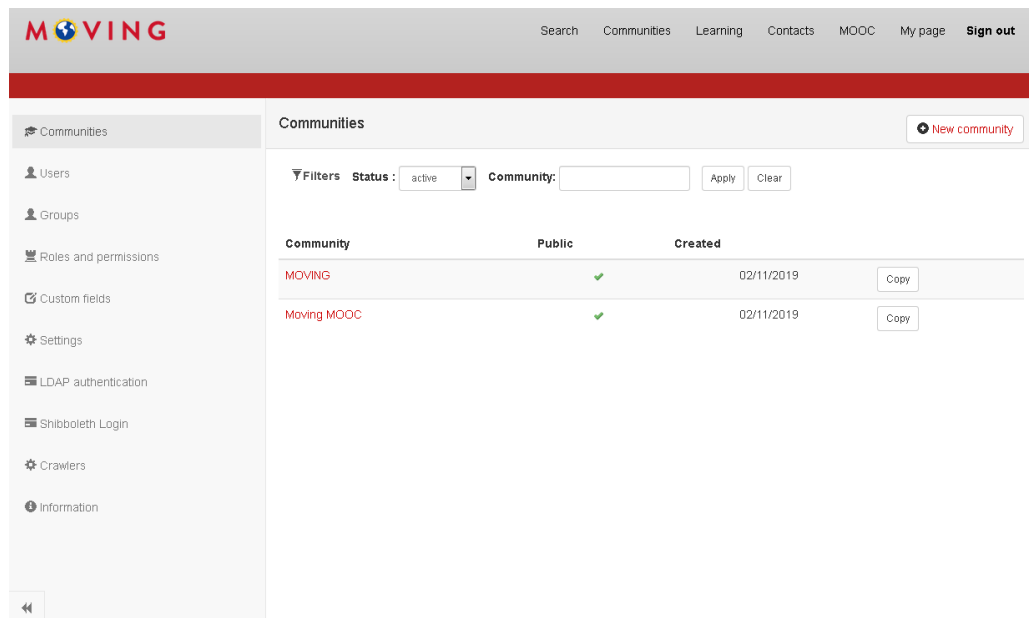


(b) Tablet

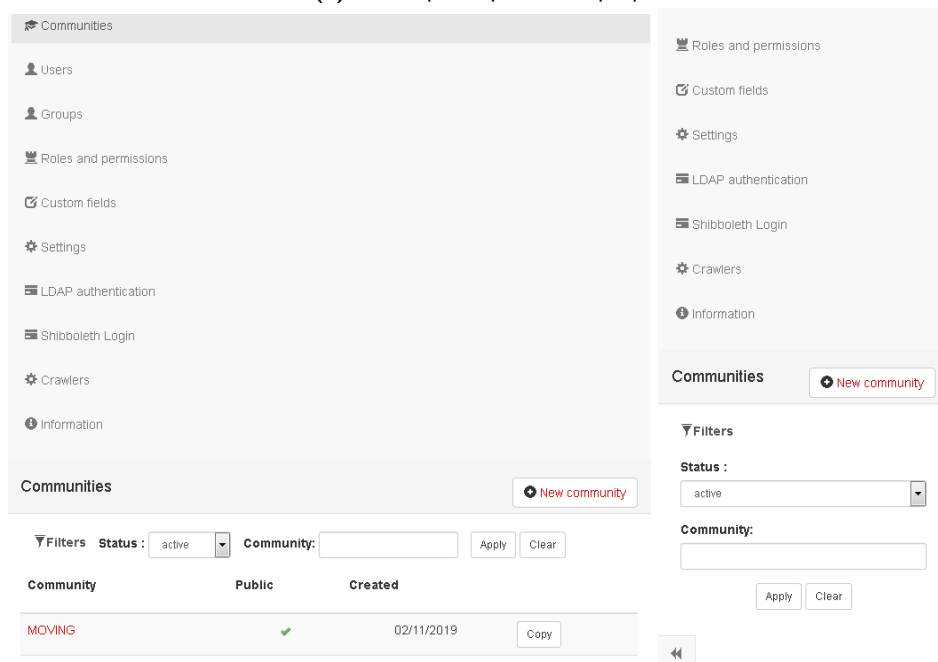


(c) Smartphone

Figure 27: Learning tracks page on screens of different devices



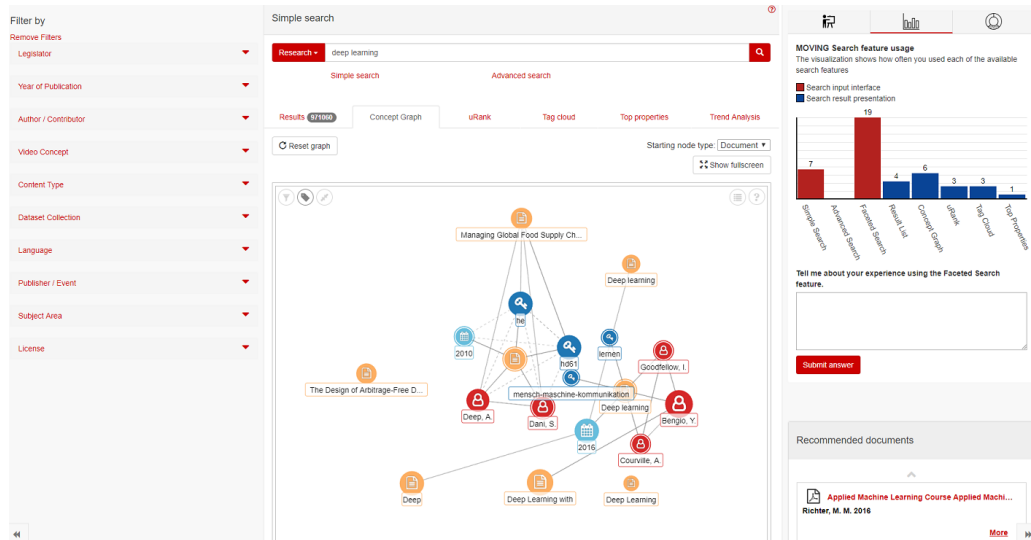
(a) Desktop computer or laptop



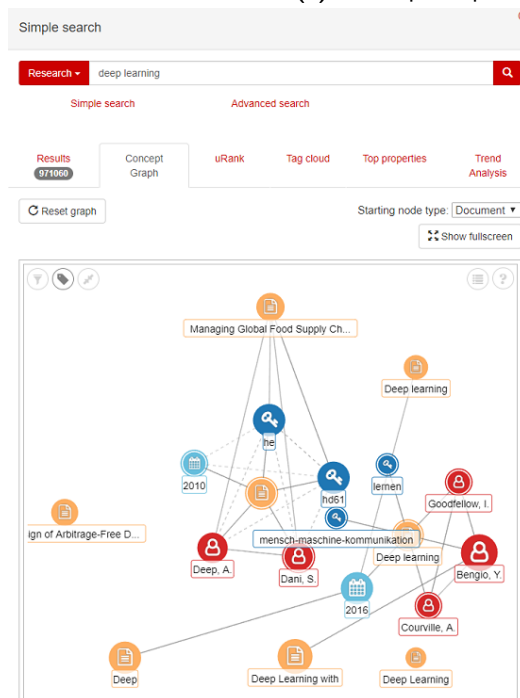
(b) Tablet

(c) Smartphone

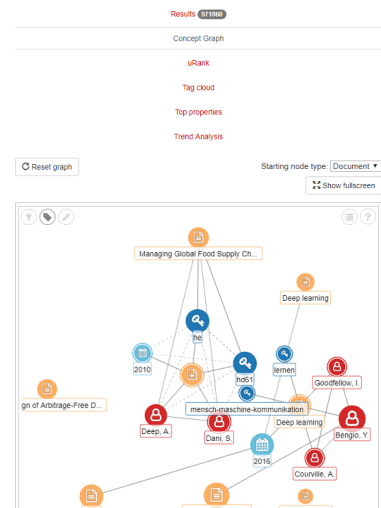
**Figure 28:** Communities page viewed by an administrator on screens of different devices



(a) Desktop computer or laptop



(b) Tablet

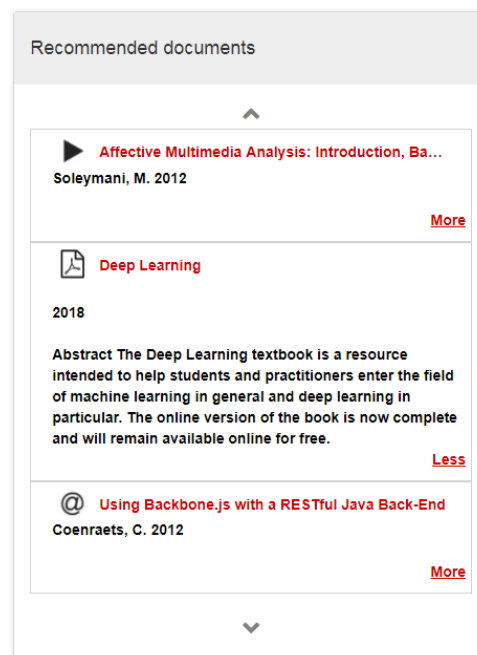


(c) Smartphone

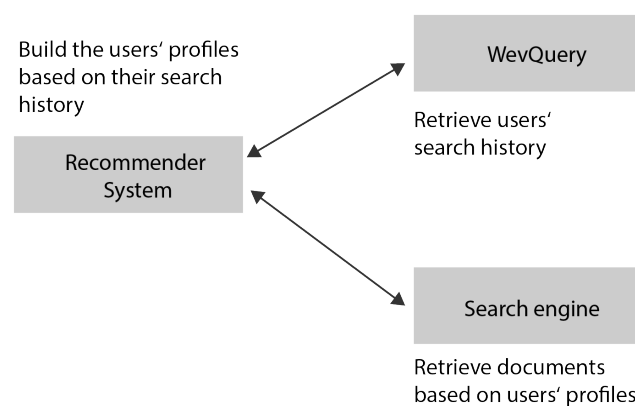
Figure 29: Search results page on screens of different devices

## 4 Recommender System

The Recommender System widget, depicted in Figure 30, is part of the search page of the MOVING platform. Thanks to it, users can receive additional suggestions possibly discovering useful resources of which they were not aware. The Recommender System has been introduced in Deliverable D2.1 (Fessl et al., 2017), while Deliverables D2.2 (Günther et al., 2018) and D2.3 (Günther et al., 2019) present advanced recommendation techniques which can improve the system. The Recommender System interacts with both the search engine and user interaction tracking and dashboard (WevQuery, see Deliverable D3.1 (Blume et al., 2017)), as illustrated in Figure 31. To build users' profiles based on their search history, it obtains the search history from the user data previously logged through WevQuery, and then it retrieves in the index the documents to suggest depending on the user's profile. In this deliverable, after recalling the key features of our profiling method, Hierarchical Concept Frequency-Inverse Document Frequency (HCF-IDF), we detail how the user profile is built and we show how the Recommender System is implemented.



**Figure 30:** The Recommender System widget



**Figure 31:** Overview of the MOVING Recommender System

## 4.1 The HCF-IDF model

The MOVING Recommender System is based on HCF-IDF (Nishioka & Scherp, 2016), a novel semantic profiling approach which can exploit a thesaurus or ontology to provide better recommendations. The HCF-IDF method extends the Concept Frequency-Inverse Document Frequency (CF-IDF) approach (Goossen, Ijntema, Frasinca, Hogenboom, & Kaymak, 2011), which in turn is an extension of the classical Term Frequency-Inverse Document Frequency (TF-IDF) model (Salton, Wong, & Yang, 1975). A more extensive description of HCF-IDF is available in Deliverable D2.1 (Fessl et al., 2017).

Given a set of  $m$  documents  $\mathbb{D}$  and a set of  $n$  users  $\mathbb{U}$ , the typical recommendation task is to model the spanned space,  $\mathbb{U} \times \mathbb{D}$ . With documents, we intend the multimedia resources available in the MOVING platform, i.e. textual documents (e.g. articles, books, regulations), videos and social media data. We model our recommendation problem as the Top- $N$  recommendation problem (Cremonesi, Koren, & Turrin, 2010). Specifically, the goal is returning the set of Top- $N$  documents which have the highest similarity with a user  $u_i$ , for each user  $u_i \in U$ . Typically users and documents represented with user and document profiles, respectively. In our case, user profiles are sets of terms previously searched by the user ordered by time and frequency of search (more details are provided in Section 4.2), while documents profiles consist of concepts preassigned to the documents.

HCF-IDF takes into account the hierarchy of concepts. This enables the model to consider related concepts not directly mentioned in a text. To do so, it applies spreading activation (Crestani, 1997) over a given concept tree and through the IDF component it prevents very generic concepts accounts for high weights.

As an example, if a user profile includes the concept *Open innovation* and given the concept tree shown in Figure 32, then HCF-IDF assigns non-zero weights to the concepts *Innovation management* and *Management*, even if they are not directly mentioned in the document. In this way, if *Innovation management* is part of the user profile, then also the documents related to *Open innovation* can be recommended. Similarly, if *Open innovation* is part of the user profile, then also the documents concerning *Innovation management* can be suggested. This helps to the system to generate more diverse recommendations since documents not directly related to the user profile but still relevant to it are considered.

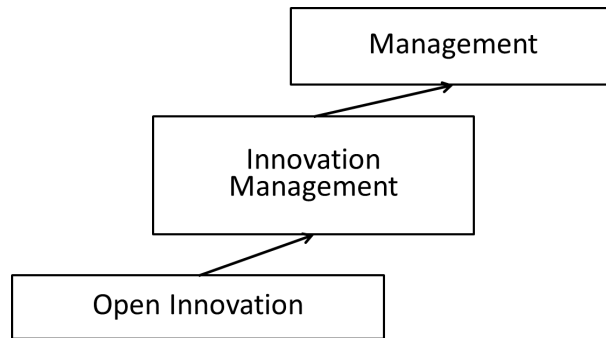


Figure 32: A concept tree

The weights in HCF-IDF are computed as defined in Equation 1 (Nishioka & Scherp, 2016), where  $BL(c, d)$  is the BellLog spreading-activation function (Kapanipathi, Jain, Venkataramani, & Sheth, 2014), which is described in Equation 2. The function  $h(c)$  returns the level where a concept  $c$  is located in the concept tree, while  $nodes$  counts the concepts at a given level in the tree. For example, with the tree showed in Figure 32,  $h(Innovation\ management)$  returns 2 and  $nodes(h(Innovation\ management) + 1)$  returns 1.  $C_l$  is the set of concepts located in one level lower than the concept  $c$  considered. Referring to Figure 32,  $C_l$  is equal to  $\{Open\ innovation\}$  for *Innovation management*.

$$w_{hcf-idf} = BL(c, d) \cdot \log \frac{|D|}{|\{d : c_i \in d\}|} \quad (1)$$

$$BL(c, d) = \frac{n_{i,j}}{\sum_{c_k \in d_j} n_{k,j}} + \frac{1}{\log_{10}(nodes(h(c) + 1))} \cdot \sum_{c_j \in C_l} BL(c_j, d) \quad (2)$$

## 4.2 Building user profiles

The user profile is a group of information that best describes a given user. In our case, it is the history of searches performed with the MOVING platform. Every term has a weight associated, which depends on how



many times and how recently the user has looked for a term. More formally, the weight  $w$  of a term  $k$  is defined as  $w = \alpha_t \cdot f_t + \alpha_h \cdot f_h$ . The time coefficient,  $\alpha_t$ , and the hit coefficient,  $\alpha_h$ , weight the time and frequency of each term in the profile. The time factor of a term,  $f_t$ , is the timestamp ( $t$ ) of its last search, normalized by the current time ( $T$ ):  $f_t = \frac{t}{T}$ . The hit factor of a term,  $f_h$ , is the number of times the term has been looked up by the user ( $h$ ) divided by the total number of searches made by the user ( $H$ ),  $f_h = \frac{h}{H}$ . The user profile is a set of pairs term-weight,  $\langle k_i, w_i \rangle$ , where  $k_i$  is a term and  $w_i$  a weight.

We set both  $\alpha_t$  and  $\alpha_h$  to 0.5 and we decided to limit the user profile to the top 25 terms, as considering more terms does not significantly improve the recommendations while increases the response time. This last parameter can also be configured, similarly to  $\alpha_t$  and  $\alpha_h$ .

### 4.3 Implementation

In the MOVING platform, a search engine allows users to search the data indexed, while WevQuery (Apaolaza & Vigo, 2017) tracks the users' behavior on the platform by capturing UI events. The Recommender System interacts with both the search engine and WevQuery. To build users' profiles based on their search history, it obtains the search history from the user data previously logged through WevQuery, and then it retrieves in the index the documents to suggest based on the user's profile. After building the user profile, it sorts the terms based on their weights in descending order and appends them in a space-separated string to build a query to generate the list of recommendations through the search engine, using HCF-IDF. The search engine is based on Elasticsearch<sup>22</sup>, as described in Deliverable D4.1 (Gottfried, Grunewald, et al., 2017). We have implemented HCF-IDF as an Elasticsearch plugin.

We implemented the Recommender System as a RESTful web service. An HTTP GET `/recommendations` issues the execution of the `get_recommendations` method, which serves the request taking a `user_id` as an argument, and returns the list of recommendations in the JSON format. For building the user profile, we use the information stored in WevQuery: the Recommender System retrieves all the searches made by the user with the specified `user_id` through the WevQuery web API. The user profile is sent to the HCF-IDF plugin, which generates the list of recommendations.

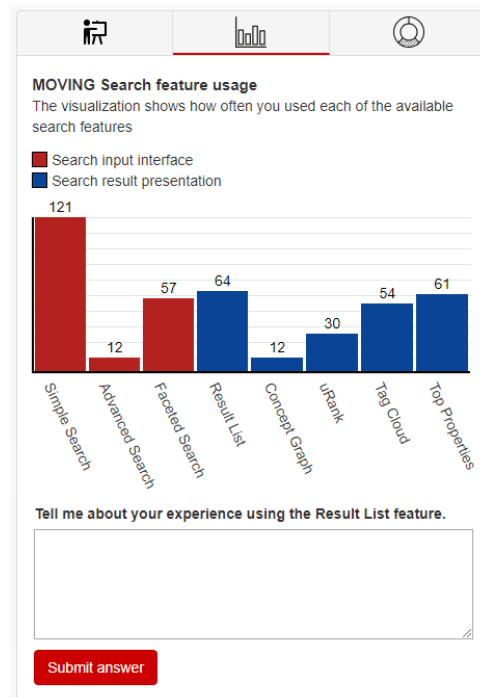
## 5 Adaptive Training Support

The adaptive training support (ATS) does not contain anymore only the *Learning-how-to-search* widget, but was further extended with the *Curriculum Reflection* widget that consists of two parts: the *curriculum learning and reflection part* and the *overall progress* part. Since the usage and the functionalities of both the widgets are already covered in Deliverable D2.3 (Günther et al., 2019), we will focus here only on the technical implementation of the functionalities of the widgets.

### 5.1 "Learning-how-to-search" widget

The *Learning-how-to-search* (figure 33) widget visualizes information about the use of features collected by the MOVING platform. The widget presents users their feature usage in a bar chart to motivate them to explore the provided features and reflect about their usage behavior.

<sup>22</sup><https://www.elastic.co/products/elasticsearch>



**Figure 33:** Learning-how-to-search widget: The tracked features are separated into features of the Search input interface and Search result presentation

Since Deliverable D4.2. (Gottfried, Pournaras, et al., 2017) the used features that are tracked have been enhanced and are divided into two categories: usage of features w.r.t. the *Search input interface*, and the usage of features w.r.t. the *Search result presentation*. Features of the *Search input interface* consist of *Simple Search*, *Advanced Search* and *Faceted Search*, while the *Search result presentation* features are *Result List*, *Concept Graph*, *uRank*, *Tag Cloud* and *Top Properties*.

The features are still being tracked through the WevQuery interaction tracking service of the MOVING Platform, coupled with the ATS Engine Service (Gottfried, Pournaras, et al., 2017, Section 5.1), on which the learning-how-to-search widget relies on to get accurate data to display the feature usage. Each of the features is being counted as used if the following criteria is met:

- Simple Search – The user enters a query in the simple search input bar and submits the form by clicking on the search button or by pressing enter
- Advanced Search – The user uses the advanced search interface and after configuring the advanced search submits the form by clicking on the search button or by pressing enter
- Faceted Search – A simple or advanced search had to be performed before and the user is on the page with the results. The user applies one or more of the filters from the left sidebar
- Result List - The user clicks on the *Result* tab on the results page
- Concept Graph - The user clicks on the *Concept Graph* tab on the results page
- uRank - The user clicks on the *uRank* tab on the results page
- Tag Cloud - The user clicks on the *Tag cloud* tab on the results page
- Top Properties - The user clicks on the *Top properties* tab on the results page

**Reflective guidance** – The reflective guidance functionality of the learning-how-to-search widget was designed independently from the tracking of the used features, thus, the functionality behind presenting the reflective prompts has not been updated since the version presented in D4.2.

## 5.2 "Curriculum reflection" widget

The *Curriculum reflection* widget is a new addition to the ATS and consists of two parts: the *curriculum learning and reflection* part and the *overall progress* part.

### 5.2.1 "Curriculum learning and reflection" part

The curriculum reflection and learning part consists of two main areas. The upper area contains either a learning prompt, suggesting to learn more about the next topic that would be the next in the current sub-module of their curriculum, and a button which opens the respective learning unit in a new tab (see figure 34), or it presents a reflective question that motivates the user to think about the currently learned topic (see figure 35). In addition to the two main areas, the progress for the of the current sub-module is displayed on the bottom of the widget.

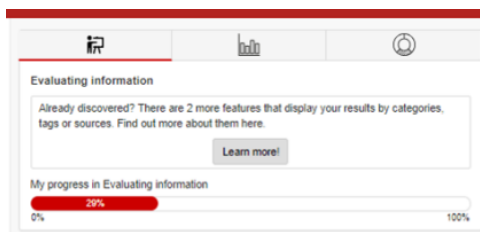


Figure 34: Curriculum reflection widget: curriculum learning part

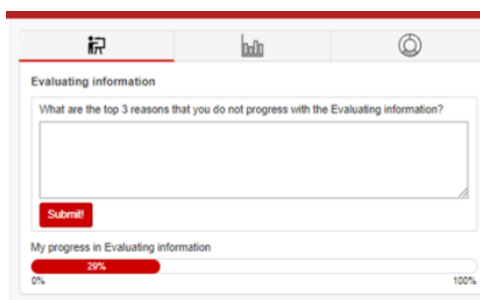


Figure 35: Curriculum reflection widget: reflection part

**Curriculum learning area** – The micro learning prompts, in the curriculum learning part of the *Curriculum reflection* widget, are displayed in the order according to the curriculum. Even if the user skips learning units or jumps across modules in the learning environment, the *Curriculum reflection* widget will give the user the micro learning prompt associated with the next not completed learning unit, starting from the beginning of the curriculum. Every time a user completes a learning unit and gives an answer to the reflective question, the next micro learning prompt will be displayed.

Since the learning units in the learning environment are very short, user interaction tracking cannot be applied there to infer the engagement level of the user with the page. Therefore, a learning unit counts as completed, if the user clicks on the "Next" button of the learning unit. Every time a user "learns" a unit, an appropriate record of the active learning unit will be created in the database with the users id, the learning unit id, the time the user completed this learning unit and that the question is completed but not yet answered.

**Reflection area** – If the user has completed a learning unit, a reflective prompt is displayed. The reflective questions consist of a set of general questions, which should bring the user to reflect about the last learning unit. Every reflective question contains a placeholder, which is set dynamically based on the topic of the last unit learned. Taken the following prompt:

"Did you notice any motivating moments during this week for progressing with the  
<b>{{sub\_competence}}</b> competence?"

The {{sub\_competence}} might be replaced with the name of the sub competence *Finding information*, which would result in the following reflective question:

"Did you notice any motivating moments during this week for progressing with the **Finding information** competence?"

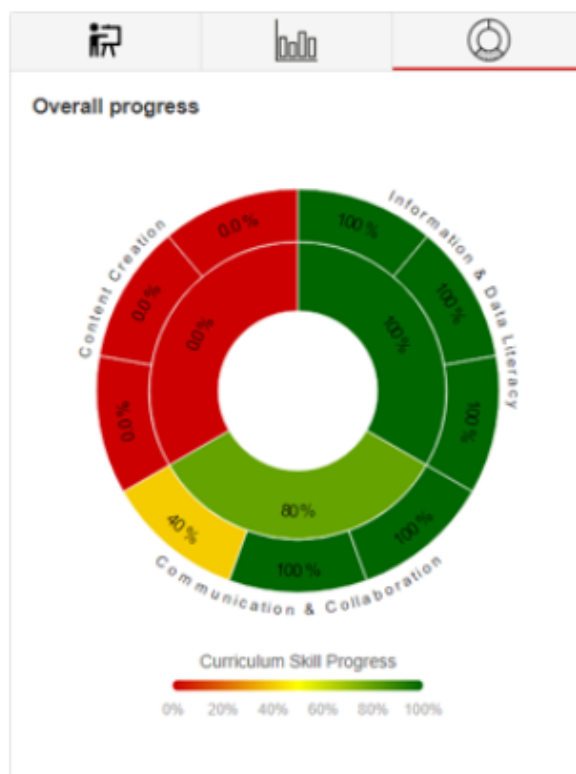
If the user answers this question, the answer will be stored in the database in a table as a record which contains the users id, question id, matching learning unit and time when it was answered. Additionally, the record of the active learning unit will be updated that it was answered.

In the case that the user goes through multiple learning units in the learning environment, the reflective question will always be directed to the last learned unit.

**User Progress** – The progress bar at the bottom of the widget shows the user's progress for the current submodule of the curriculum. The progress is calculated by taking the number of lessons for this submodule which the user has completed and dividing it by the total number of learning units in this submodule.

### 5.2.2 "Overall progress" part

The overall progress part of the widget shows the user's learning progress with regard to the curriculum using a sunburst visualization. In figure 36, it can be seen that the curriculum is divided into three modules. Furthermore, a fourth module is available for auditors of EY. Each module is represented as section in the inner circle of the visualization and each module is additionally divided into three sub-modules (outer circle). Every time a user completes a new learning unit, the percentage in the respective section in the sunburst diagram is updated. Furthermore, the progress in each sub-module is encoded by color. If the user has not completed any learning units in a sub-module (0%) the respective section will be red. Making progress in a sub-module will turn the section to yellow (50%) and finally, by completing a sub-module, the section will turn green (100%). This is also explained by the legend below the visualization. Moreover, the sections in the sunburst diagram are ordered to mirror the structure of the curriculum. Starting from the top, the sub-modules get completed clock-wise, slowly turning the visualization green.



**Figure 36:** Overall progress widget: The first module was completed and the second module is in progress

The sunburst visualization of the *Overall progress* part was implemented using D3.js<sup>23</sup>, and is dynamic and responsive. It is dynamic in the sense, that it adapts to a varying number of modules and submodules. Therefore, if changes in the curriculum happen, the visualization will handle the update itself, without the need to do any further development. It is responsive in the way, that it uses SVG<sup>24</sup> with a viewbox for the visualization. The SVG always fills the available width, while the viewbox guarantees that the sunbursts proportions stay intact.

<sup>23</sup><https://d3js.org/>

<sup>24</sup><https://www.w3.org/TR/SVG2/>

## 6 User interaction tracking

MOVING's interaction tracking has captured more than seven million events from visitors to the MOVING platform. As the platform has naturally evolved since the tracking was deployed, several modifications have been made to ensure the interaction tracking was reliable and satisfied the needs of other MOVING modules.

### 6.1 Interaction tracking

UCIVIT (Apaolaza, Harper, & Jay, 2013) has been modified extensively to improve its scalability, as well as the privacy of the data. Event requests were changed from GET to POST, so they would take advantage of MOVING's Web site's secure connection. The behaviour of the interaction tracking with modern browsers has been tested. Slight changes, such as the way the mousewheel was captured, were necessary to ensure full compatibility. We also identified another issue caused by third party software. In some cases the use of ad-blockers would prevent the execution of the interaction tracking software, without triggering any visible error. To prevent this, a notification was added so visitors are aware of the problem, urging them to deactivate their ad-blocker. The possibility of deactivating the interaction tracking altogether was always available to the users, in the case that they were not willing to deactivate their ad-blocker.

New events have enabled the addition of further user-specific widgets. For example, Section 4.1.3 in Deliverable D3.3 (Vagliano et al., 2019) introduced the use of a custom event that augmented a regular event with a particular field from the results page, allowing the search history to show the users how many results had been displayed for each query. These additions have also allowed us to extend the knowledge acquired about users' behaviour. In the case of visualisations, UCIVIT has been modified to accept visualisation events (Vagliano et al., 2019, Section 4.1.3), which could then be used for evaluations (Vagliano et al., 2019, Section 5.4). Interaction with externally sourced videos has also been included. The detection of these events requires the use of specific APIs<sup>25,26</sup>, with different behaviour. However, we ensured that generic video events (*play*, *pause*, *skip*) were covered by both of them, so they could be used as additional engagement metrics in the study carried out in D1.4 (Apaolaza et al., 2019).

### 6.2 WevQuery

In the case of WevQuery, developers used the interface presented in Section 6 in D4.2 (Gottfried, Pournaras, et al., 2017) to create queries specific to their needs. However, in the case of the search history, a custom REST query had to be added, due to the complexity of identifying the latest query from a series of iterations (Vagliano et al., 2019, Section 4.1.3).

## 7 Data acquisition and processing

This section presents the final version of the components that are integrated in the MOVING platform for retrieving data coming from external sources. The collected data is then enriched through processing components which are either updated or newly integrated since the Deliverable D4.2 (Gottfried, Pournaras, et al., 2017). Furthermore, in Section 7.3 we are presenting the novel *Explore* functionality which enables users to visually analyse PDF files.

### 7.1 Data acquisition

In the following section, we are giving an overview about the documents accessible through the MOVING platform. Additionally, we are describing the services that are pulling the data from external sources and integrating them to the platform.

#### 7.1.1 Data accessible through the MOVING platform

In March 2018, the MOVING platform is hosting over 22.3 million documents metadata. These metadata represent various kind of documents, including books, scientific articles, laws and regulation, funding opportunities, as well as videos, such as lectures and tutorials, and social media posts. This amount is constantly growing because new documents are regularly crawled from the web. Specifically, the MOVING platform contains the following data:

<sup>25</sup>[https://developers.google.com/youtube/iframe\\_api\\_reference](https://developers.google.com/youtube/iframe_api_reference), last accessed on 8/3/2019

<sup>26</sup><https://developer.vimeo.com/>, last accessed on 8/3/2019

**ZBW Economics dataset.** It consists of about 6.4 million metadata records and 413,097 full-text scientific publications in Economics. These data include about 1.8 million records from the Munich Personal RePEc Archive<sup>27</sup> which are also indexed in the EconBiz service<sup>28</sup> provided by ZBW.

**GESIS dataset.** It includes almost 4 million metadata records and 49,653 open access documents in social sciences, as well as descriptions of roughly 50,000 social science research projects which comprise about 10,000 institutions.

**Videolectures.NET.** It consists of 20,574 metadata records of educational video lectures with transcripts.

**Laws and regulations dataset.** It collects 1,687 metadata records, each containing information about a law and its changes over time. The dataset mostly covers the see-related regulations in Germany and the European Union. This dataset was provided by Wolters Kluwer<sup>29</sup> within the H2020 ALIGNED project<sup>30</sup>.

**CORE.** The CORE repository<sup>31</sup> includes 123 million metadata records and 9.8 million open access full-texts. The CORE integration is currently in progress since many documents need to be added and processing the full-texts requires some more time. At now almost 11 million documents have been included and, overall, we plan to integrate 74.5 million documents from CORE. Some documents from this dataset are skipped either because they are duplicates, e.g. the same documents were already added into the platform from another source as it is the case for the Munich Personal RePEc Archive, or because the metadata do not comply with the common data model (see D3.3 for further details on the common data model).

**Crawled documents.** The platform hosts 1.1 million of documents from the Web. This include social media posts, metadata from the Linked Open Data cloud<sup>32</sup> and web pages. New data are continuously being crawled. About 50,000 new documents per day are added.

Additional information on the data hosted in the MOVING platform can be found in the deliverable D6.2 "Data management plan" (Collyda et al., 2017).

ZBW Economics, VideoLecture.NET and CORE data can also be through the EconBiz, VideoLecture.NET<sup>33</sup> and CORE services, respectively. As it can be seen from Table 1, the MOVING platform allows its users to access a greater amount of data than EconBiz and VideoLectures.NET. Since roughly 63.5 million documents from CORE are being integrated, we expect to have soon over 80 million documents in the MOVING, without counting the new crawled documents continuously harvested. Furthermore, while EconBiz and CORE provide textual data only and VideoLecture.NET focuses on videos, MOVING data are heterogeneous. Finally, while EconBiz and CORE contains publications only, the MOVING platform also hosts other type of resources, such as social media posts, blogs, funding opportunities, laws and regulations, tutorials and video lectures.

**Table 1:** Comparison of the MOVING platform's data with respect to EconBiz and VideoLecture.NET

Platform	Amount of data	Data types	Domain
<b>MOVING</b>	22.3 M	Text, Video, metadata, social media posts	Cross-domain
EconBiz	10.7 M	Text, metadata	Economics
CORE	123.0 M	Text, metadata	Cross-domain
VideoLectures.NET	20.5 k	Video	Cross-domain

### 7.1.2 Web crawling

Web crawling in the MOVING platform is performed by three distinct crawlers that run as background system services. The Focused web-domain Crawler (FDC) continuously crawls websites that are specified by the platform's administrator. The Social Stream Manager (SSM) is responsible for crawling social media sources and the Search-Engine based Crawler (SEC) utilizes the Google API to search the web. Both the SSM and the SEC collect data for a given set of topics by the platform's administrator. For the insertion of websites and topics there is an interface that can be accessed only by the platform's administrator. Technical details

<sup>27</sup><https://mpra.ub.uni-muenchen.de/>

<sup>28</sup><https://www.econbiz.de/>

<sup>29</sup><https://wolterskluwer.com/>

<sup>30</sup><http://aligned-project.eu/>

<sup>31</sup><https://core.ac.uk/>

<sup>32</sup><https://lod-cloud.net/>

<sup>33</sup><http://videolectures.net/>

about the crawlers have been outlined in the previous Deliverables D4.1, D4.2 (Gottfried, Grunewald, et al., 2017; Gottfried, Pournaras, et al., 2017).

An additional crawler for the Horizon 2020 funding calls has been added and is integrated within the SSM (see Figure 37). An adaptive crawling mechanism has also been implemented for the FDC. This mechanism adjusts the crawling frequency for each website according to the webpage update rate of each website and the number of clicks for each website in the platform's search results. Moreover a new, more sophisticated, duplicate detection method has been implemented and is applied on all the crawlers. The method compares the fulltext of a candidate-for-indexing document with the fulltexts of already indexed documents with similar URLs. All the above are described in detail in D3.3 (Vagliano et al., 2019).

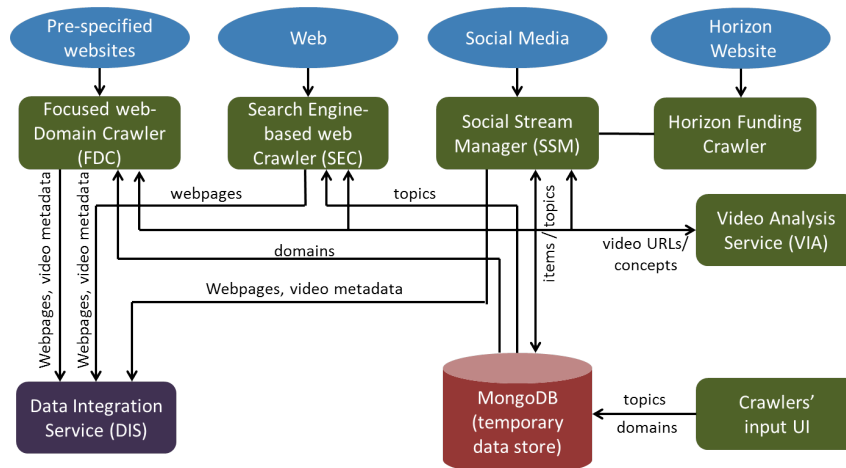


Figure 37: MOVING Crawler architecture.

### 7.1.3 Bibliographic Metadata Injection

The Web is a widely accepted source of information for various purposes. The information is primarily presented as text embedded in HTML documents to improve the readability for humans. However, alongside the commonly known Web of (HTML) documents, there exists an ongoing trend of publishing and interlinking data in machine readable form following the Linked Data principles<sup>34</sup>. Thus, forming the so called Linked Open Data (LOD) cloud<sup>35</sup> or also referred to as the Web of Data. Among other features, LOD allows embedding semantics into search operations. While in the Web of documents we can search for occurrences of the term *book*, in the Web of Data we can explicitly search for resources which are of type *book*. Please find more information about this in Deliverable D3.3 (Vagliano et al., 2019).

Similarly to classic web search, finding information in the Web of Data is a challenging task since there is a vast amount of data available, which is distributed over various data sources. Furthermore, since there is no central authority responsible for the Web of Data, there is a large variety in how the data is modelled, i.e., which vocabulary terms are used in which combination to model the data. In the Bibliographic Metadata Injection service (BMI), we address these challenges using schema-level indices. In particular, we use the data search engine LODatio (Blume & Scherp, 2018; Gottron, Scherp, Kraye, & Peters, 2013) to find data sources providing relevant information. Furthermore, we exploit relationships between vocabulary terms found in the Web of Data to automatically extend mappings from vocabulary terms found in the Web of Data to vocabulary terms in our common data model. With this approach, we are able to find, harvest, and integrate bibliographic metadata from the Web of Data into our MOVING platform. To implement the service, we developed the framework IMPULSE (Integrate Public Metadata Underneath professional Library SERVICES)<sup>36</sup>. The interaction of the different components is depicted in Figure 38. Each step is further explained in the following subsections.

**Input** The service requires as input a SPARQL<sup>37</sup> query using a combination of RDF<sup>38</sup> types and/or properties. SPARQL is the de facto standard query language for Linked Data. This query needs to be carefully chosen

<sup>34</sup><https://www.w3.org/DesignIssues/LinkedData.html>, last accessed on 05/03/2019

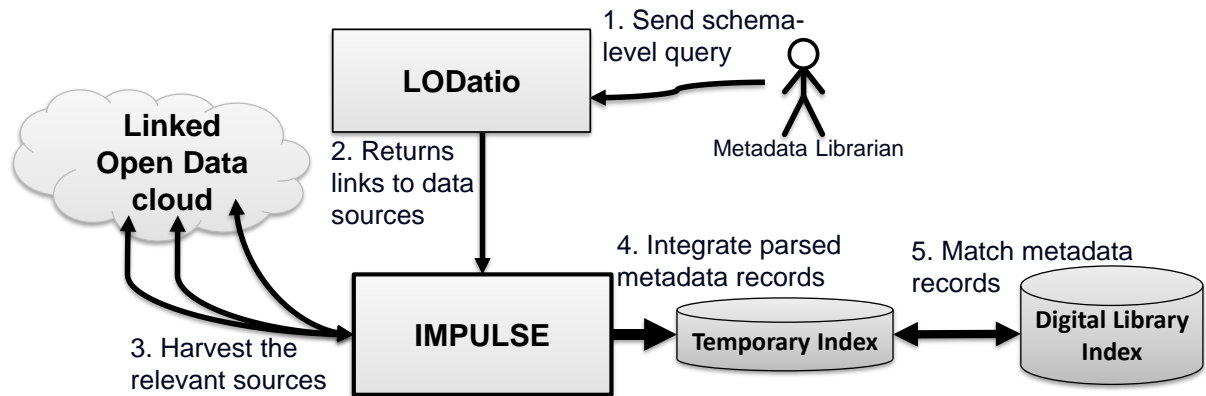
<sup>35</sup><http://lod-cloud.net/>, last accessed on 05/03/2019

<sup>36</sup><https://github.com/t-blume/impulse>

<sup>37</sup><https://www.w3.org/TR/rdf-sparql-query/>, last accessed on 05/03/2019

<sup>38</sup><https://www.w3.org/RDF/>, last accessed on 05/03/2019





**Figure 38:** System overview. Step 1: send a query to the data search engine LODatio. Step 2: LODatio returns a set of links to data sources. Step 3: IMPULSE accesses all data sources and harvests metadata. Step 4: All metadata is parsed and integrated in an temporary index. Step 5: Metadata from the temporary index is integrated in the existing database.

in order to reliably identify relevant information. In particular, two important aspects need to be considered: the vocabularies and the combination of properties and types from those vocabularies. In the context of LOD, “A vocabulary consists of classes, properties and datatypes that define the meaning of data.” (Vandenbussche, Ateamezing, Poveda-Villalón, & Vatant, 2017). Thus, choosing properties and types from domain specific vocabularies is crucial to avoid false results. In Listing 5, we present an exemplifying query using the Bibliographic Ontology (bibo)<sup>39</sup> and DCMI Metadata Terms (dcterms)<sup>40</sup>, which are two domain-specific vocabularies. However, such vocabularies are often used in a non-conform way, e.g. in a different context (Meusel, Bizer, & Paulheim, 2015). Thus, the combination for specific properties and types from the desired vocabularies is also important. To sufficiently address this issue, we rely on previous analyses of how to explicitly model bibliographic metadata as LOD (Jett, Nurmikko-Fuller, Cole, Page, & Downie, 2016).

```

1 SELECT ?x
2 WHERE {
3   ?x rdf:type bibo:Document .
4   ?x dcterms:title [].
5   ?x dcterms:description [].
6   ?x dcterms:creator [].
7 }

```

**Listing 5:** SPARQL query using the RDF type bibo:Document and three properties from the DCMI Metadata Terms vocabulary to search for bibliographic metadata.

Furthermore, we automatically generate additional queries by using the ontology inferencing feature implemented in LODatio (Blume & Scherp, 2018; Gottron et al., 2013). These additional queries exploit known RDF Schema<sup>41</sup> relations between properties and types. With RDF Schema, one can describe how two properties are related, e.g., describe a hierarchical relationship between them. For example, `bibo:authorList rdfs:subPropertyOf bibo:contributorList`. We can use such statements to generate variations of the original query, e.g., by interchanging properties. The benefit of inferencing is that one may catch more metadata records than the original query. These inferred schema-level queries allow us to flexibly find data sources containing bibliographic metadata.

```

1 SELECT ?x
2 WHERE {
3   ?x dcterms:title ?a .
4   ?x bibo:contributorList ?b
5 }

```

**Listing 6:** Base BIBO-query using title and author.

```

1 SELECT ?x
2 WHERE {
3   ?x dcterms:title ?a .
4   ?x bibo:authorList ?b
5 }

```

**Listing 7:** Generated inferred query for Listing 6.

In order to automatically process the meta data found in the Web of Data, a mapping file is required. In this mapping file one defines which information is used for which attribute within our common data model (see Deliverable D3.3 “Technologies for MOVING data processing and visualisation v1.0”). The example presented

<sup>39</sup><http://lov.okfn.org/dataset/lov/vocabs/bibo>, last accessed on 05/03/2019

<sup>40</sup><http://lov.okfn.org/dataset/lov/vocabs/dcterms>, last accessed on 05/03/2019

<sup>41</sup><https://www.w3.org/TR/rdf-schema/>, last accessed on 05/03/2019



in Listing 8 maps the example SPARQL query to the common data model. Please note, in this file we do not use the common namespaces (dcterms, bibo), but fully qualified URIs. We can assume that all properties used in the query appear in the data sources we harvest. The IMPULSE framework is able to automatically extend the mapping file when in the ontology inferencing feature is enabled. This means, the mappings for the additionally generated queries are also generated automatically.

The mapping file also offers to include optional properties, e.g. `http://purl.org/dc/terms/language`. This information can be present in some cases, but does not necessarily have to.

```
1 {"BiblItemMapping":{
2   "title":["http://purl.org/dc/terms/title"],
3   "abstract":["http://purl.org/dc/terms/description"],
4   "author":["http://purl.org/dc/terms/creator"],
5   "startDate":["http://purl.org/dc/terms/date"],
6   "venue":["http://purl.org/dc/terms/isPartOf"],
7   "language":["http://purl.org/dc/terms/language"],
8   "keyword":["http://purl.org/dc/terms/subject"]
9 },
10 "AuthItemMapping":{
11   "name":["http://www.w3.org/2000/01/rdf-schema#label",
12     "http://xmlns.com/foaf/0.1/name"]
13 }
14 }
```

**Listing 8:** Example mapping file used to transform bibliographic metadata modelled as Linked Data into MOVING's common data model

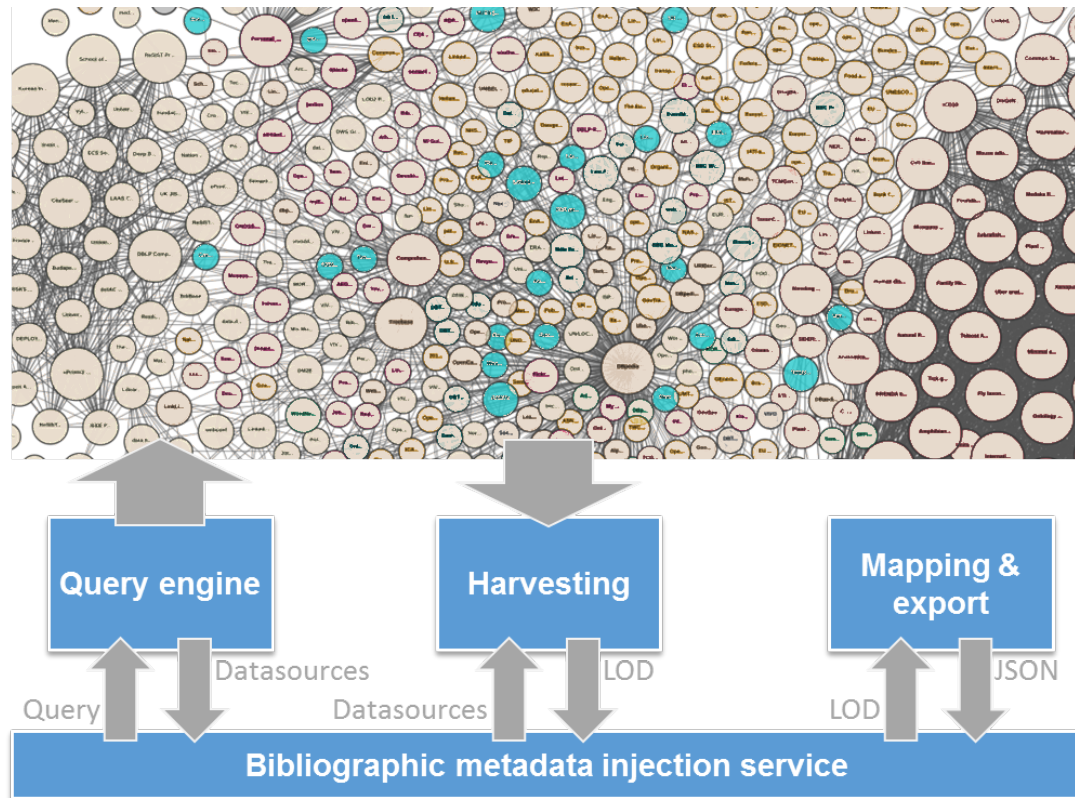
Finally, the service requires as input data from the LOD cloud. In the current version, this data is provided as a (crawled) dataset, which is available on a local storage device. Accessing the data directly in the LOD cloud using a LOD crawler is possible and can be implemented in the future.

**Functionality** This service operates in three steps: (1) query execution, (2) harvesting, and (3) mapping and export. For each step there exists a component responsible for its execution. A high-level view of the components and the data flow is depicted in Figure 39.

As a first step, the query is sent to the LOD search engine LODatio (Gotttron et al., 2013). LODatio supports the execution of queries using a combination of RDF types and properties, since it uses a schema-level index. More details on schema-level indices are available in Section 3.5.4 of Deliverable D3.3 “Technologies for MOVING data processing and visualisation v1.0”. The LOD search engine returns a list of identified data sources within the LOD cloud. The LOD cloud can be seen as a distributed knowledge graph, while this list can be considered a sub-graph comparable to a list of HTML websites in classic web search. For instance, in Figure 39, the relevant sub-graph consists of the highlighted nodes. In the subsequent step, the datasources are harvested by means that the contained information is extracted. The information is parsed and converted as specified in the mapping file. Please note, each data source possibly contains additional information which may be not relevant, e.g. non bibliographic content. However, the mapping file ensures we only parse the desired information. Finally, the converted data is exported into JSON objects following the common data model.

**Results** The bibliographic metadata injection service is able to retrieve bibliographic metadata modelled as Linked Open Data and transform it into our common data model. Thus, it returns JSON objects, which can be ingested into the Elasticsearch index as additional content. An excerpt of the first generated dataset is presented below. For the final version of the MOVING platform, we ingested 1,117,671 bibliographic metadata records harvested from a crawled snapshot of the LOD cloud called Billion Triple Challenge Dataset 2014<sup>42</sup>.

<sup>42</sup><http://km.aifb.kit.edu/projects/btc-2014/>, last accessed on 05/03/2019



**Figure 39:** The BMI service acquiring additional content from the LOD cloud using its core components. The highlighted datasources in the LOD cloud are identified using the query engine, subsequently harvested, and their respective content mapped and exported to JSON objects following the common data model.

```

1 {
2   "identifier": "http://bnb.data.bl.uk/id/resource/009098350",
3   "URL": "http://bnb.data.bl.uk/doc/resource/009098350?_metadata=all,views,formats,execution,bindings,site",
4   "title": "China investment guide :the North-east :establishing and operating a business",
5   "authors": [
6     {
7       "identifier": "http://bnb.data.bl.uk/id/person/MorarjeeRachel",
8       "URL": "http://bnb.data.bl.uk/doc/resource/009098350?_metadata=all,views,formats,execution,bindings,site",
9       "name": "Morarjee, Rachel"
10    }
11  ],
12  "venue": {
13    "identifier": "http://bnb.data.bl.uk/id/
14      ↪ resource/009098350/publicationevent/HongKongLondonEconomistIntelligenceUnit1997",
15    "URL": "http://bnb.data.bl.uk/doc/resource/009098350?_metadata=all,views,formats,execution,bindings,site",
16    "name": "Hong Kong ; London :Economist Intelligence Unit, 1997"
17  },
18  "source": "BTC2014",
19  "docType": "document/RDF",
20  "language": "en",
21  "keywords": [
22    "Manchuria%28China%29Economicconditions",
23    "BusinessenterprisesChinaManchuria",
24    "InvestmentsChinaManchuria",
25    "Manchuria%28China%29",
26    "Manchuria(China)",
27    "Manchuria(China)Economicconditions"
28  ]
29 }

```

**Listing 9:** Example bibliographic metadata record harvested from the Billion Triples Challenge (BTC) 2014 dataset and converted to the common data model.

## 7.2 Data processing

In the following section, we are presenting all services of the MOVING platform that are processing the acquired data (7.1) to harmonize or enrich them.

### 7.2.1 Data Integration Service

The MOVING platform provides access to a large variety of documents coming from different data sources (see Section 7.1.1). To acquire new data, we follow two main approaches. First, we integrate publicly available data sources on the web using crawlers and harvesting tools (see Sections 7.1). Second, we parse available data collections in a batch process (see Deliverable 6.2 "Data management plan" (Collyda et al., 2017)). Therefore, our service for integrating the data needs to support an interface for the continuously running crawlers and harvesting tools, as well as an interface to integrate large collections of data in a batch process. It is critical for the data consistency to provide both interfaces in a single service. The common data model for MOVING lays the foundation for data integration in the project (Vagliano et al., 2019). All types of documents are stored in the index based on the common data model and are subject to validity and minimum quality checks before being indexed. Therefore, we apply normalization patterns in order to ensure a smooth processing of the data within the platform. For example, for fields like *source* and *docType*, we maintain a list of all valid values to avoid ambiguity. In order to ensure that the data in our index full-fills all our requirements to the data quality, we developed the Data Integration Service (DIS).

The main purpose of the DIS is to validate documents before they are integrated into our index. Validating against the common data model enables a minimum quality threshold for each document, i.e., mandatory fields are present and used correctly. The quality checks are performed by the DIS using JSON schema. The full list of quality checks is defined in the JSON schema document found in Deliverable D3.3 (Vagliano et al., 2019).

For valid documents, we also generate additional metadata fields and default values. For example, we implemented a rule to determine which type of document qualifies as research, learning, or funding material. For each document, the following fields are generated or updated.

**Document identifier** Documents in Elasticsearch have an `"_id"` field that uniquely identifies each document. In the DIS, we use this value generated by Elasticsearch to initialise the `"identifier"` value of each document. This is needed for the de-duplication process. Two documents in Elasticsearch representing the same real-world document will have two different `"_id"` values but the same `"identifier"` value.

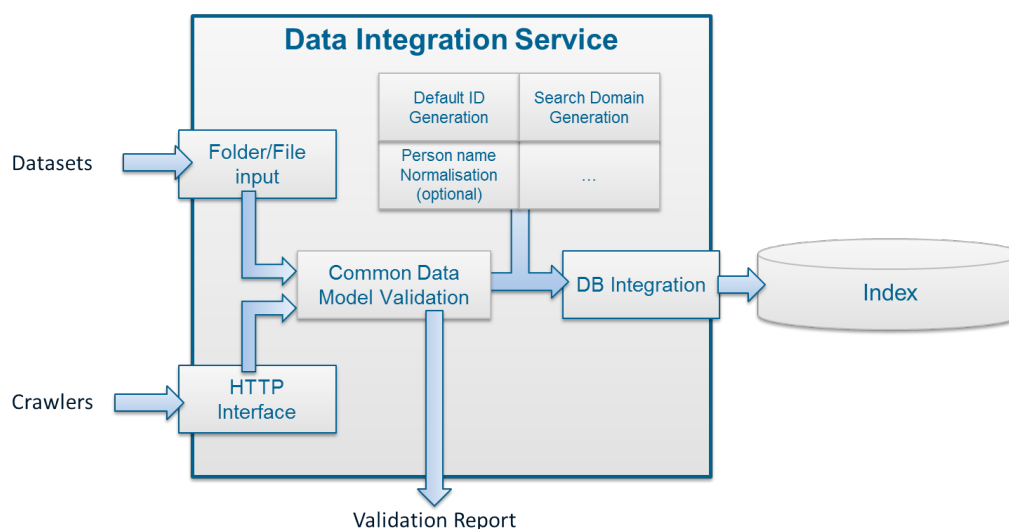
**Search domain** We distinguish the search domains `"research"`, `"learning"`, and `"funding"` in MOVING. We identified a set of document types that qualify for the respected search domain. Thus, for each document, the correct search domain is added based on the document type by the DIS.

**Metadata identifier** For the provided metadata fields for *author*, *affiliation*, and *location*, the DIS generates default identifiers. Default identifiers are compound of their type (*person*, *organisation*, *location*) and the parsed name. These default identifiers are updated by the Author Disambiguation Service (see Section 7.2.2), e.g., when there are two different persons with same name in the MOVING index.

**Named entity recognition** For the Auditor's use case, it is required to extract entities from the full-text of documents on-the-fly. In this context, on-the-fly means without storing the document in the index and within a reasonable response time. Therefore, the DIS has also the optional feature of extracting entities (person, organisation, location) from the full-text. This is implemented using the Stanford CORE NLP Framework with the pre-trained German and English language models. Furthermore, we tailored the solution to the Auditor's use case by adding a rule base with known DAX 20 company names. More details are presented in D3.3 (Vagliano et al., 2019). The platform offers a different module for handling entities relevant to the scientific use case (see Section 7.2.4).

**Implementation** The architecture of the DIS is depicted in Figure 40. The DIS can be started in batch mode to index one or more JSON files containing one or more documents. Each document is validated according to the latest version of the JSON schema document. Valid documents are integrated into Elasticsearch and are automatically included in the MOVING platform's search. The document log keeps track of all indexed documents. This log file of document identifiers enables the author disambiguation service to incrementally process the documents. Furthermore, the entity log keeps track of each extracted entity. Invalid documents are rejected and a detailed error report is provided in the validation log.

The DIS can be started as a background service accepting HTTP POST requests. The DIS responds to three REST interfaces. First, JSON documents following the common data model can be integrated in the index if sent to the interface listing by default on `localhost:8000/dis`. Second, PDF files can be integrated in the index if sent to the interface listing by default on `localhost:8000/dis/pdf-upload`. Third, PDF files can be processed on-the-fly if sent to the interface listing by default on `localhost:8000/dis/pdf-process`. The latter two interfaces are used for the *Explore* functionality which is described in more detail in Section 7.3.



**Figure 40:** Basic functionality of the Data Integration Service

### 7.2.2 Author Name Disambiguation

Author name disambiguation addresses the problem to map a single author name on a document to the right real-world author. The problem occurs when an author is referred to just by a string of characters and not by an identifier. The problem increases with the size of the document collection. For this problem, GESIS has provided a novel method which decides for a set of author mentions with the same name which of them belong to the same author and which do not. This is a clustering problem over the author mentions which we solved by applying agglomerative clustering based on document features extracted from the document collection for the mention under study (such as affiliation, co-authors, referenced authors, email addresses, keywords, publication years). The approach has been described in detail in Deliverable D3.1 “Technologies for MOVING data processing and visualisation v1.0”, Section 3.7. The method has been fully integrated into the MOVING platform and applied to the entire MOVING corpus by providing and executing a number of Python scripts that can be run by an administrator on the platform. The scripts implementing the author disambiguation service have been described in detail in Deliverable D4.2 “Initial responsive platform prototype, modules and common communication protocol”, Section 7.5. The scripts are now integrated in a way that the author name disambiguation service starts automatically whenever new documents are added to the corpus. The service is then performed incrementally on the basis of the author names that appear in the newly added documents. As a result, each disambiguated author is assigned a unique internal authorID such that documents having the same author name but belong to different real-world author can be distinguished. In order to make the disambiguation results available to the user, we have furthermore implemented a feature into the MOVING platform interaction layer that allows the user to click on the name of an author given for a selected document. This click event then triggers an Elasticsearch query for the authorID assigned to the respective mention such that only documents authored by the author having the authorID in question appear on the result page.

### 7.2.3 Deduplication

Document Deduplication becomes particularly relevant when in a large document space, as provided by MOVING, documents originate from different sources. The duplicate problem appears when two (or more) metadata entries refer to the same real-world document, but might describe it in a slightly different way. To keep the index as small and clean as possible, GESIS has developed a method for identifying duplicates in a document collection. For this, we defined a similarity measure that distinguishes a duplicate pair from a non-duplicate

pair of documents. The measure reuses the similarity measure developed for author name disambiguation described above. In order to reduce the number of pairs to be compared we defined a method for partitioning the collection into blocks such that only document pairs inside the same block need to be compared. Both the similarity measure and the blocking procedure have been formalized in the MOVING project as efficient linear algebra operations on sparse matrices in order to make an application of the deduplication on the entire MOVING index more successful in terms of runtime and memory consumption. The method has been described in detail in the Deliverable D3.2 "Technologies for MOVING data processing and visualisation v2.0", Section 3.4.

The duplicate detection is implemented as a pipeline of Python scripts. The pipeline (1) first fetches all the features required from the Elasticsearch index and stores them into an SQLite database to be able to process structured queries needed for the following steps. It is possible to specify a list of documents for which to fetch the features. Otherwise features will be fetched for all documents. As the feature extraction script actually updates the feature database, it is only executed on the document IDs that have been added since the last run. In the next step (2) the duplicate detection script is executed. Detected duplicate pairs are stored in a SQLite database together with a confidence value. Next (3), the results of the duplicate detection process are inserted into the Elasticsearch index of the MOVING platform. In the next step (4) we rank all documents within a cluster of duplicates according to a) the relevance of the source, b) the relevance of the document type, and c) the timestamp of the document. This procedure, which is described more detailed in Deliverable D3.3 "Technologies for MOVING data processing and visualisation v3.0" (Section 2.1), also provides general statistics on the duplicates (i.e., number of clusters, number of members per cluster, checks if clusters of duplicates are disjoint) and computes the best ranked candidate within a cluster. The results are stored in a SQLite database. Thereafter (5), the ranking values for all duplicates are inserted into the Elasticsearch index. Finally (6), all duplicates except from the best ranked documents are removed from the Elasticsearch index.

#### 7.2.4 Named Entity Recognition and Linking

Named entity recognition and linking (NER&L) aims at semantically annotating unstructured sources by additional information extracted from the text, such as mentions of data sets, associated research methods, tools, measurements, etc. By this, a document can be enriched by additional context information which might help to improve search and retrieval. We applied NER&L to the scientific use case of MOVING, specifically to the case of information retrieval in the Social Sciences. For this, we defined six basic entity types, semantic classes respectively, relevant for the Social Sciences, namely *Research Method*, *Research Theory*, *Research Tool*, *Research Measurement*, *Research Dataset*, and *Research Field* which we extracted from fulltexts provided by Social Science Open Access repository (SSOAR). To determine the relevance of terms identified TF-IDF scores are calculated. Extracted entities are linked to the SAGE Thesaurus<sup>43</sup> as an external knowledge base which we extended by automatically extracting further terms from SSOAR fulltexts, such as abbreviations, synonyms and related terms. The entire extraction pipeline is described more detailed in the Deliverable D3.3 "Technologies for MOVING data processing and visualisation v3.0" (Section 2.2). We inserted all extracted entities together with a relevance score to the entity fields of the Common Data Model (see D3.2) of the Elasticsearch Index at the MOVING platform to be incorporated in the Concept Graphs provided by KC, such that a user is able to explore the entities related to a document under study. A separate module in the platform, included in the DIS, addresses the NER&L task for the auditors' use case (see Section 7.2.1).

#### 7.2.5 Video analysis

For the processing of videos CERTH provides the MOVING platform with 2 different technologies which run as external services.

**Video Analysis Service** The Video Analysis Service (VIA) is a REST service that performs visual analysis, in the form of temporal fragmentation and concept detection, on crawled videos. It can download and process videos hosted on social media platforms such as Youtube and Twitter, and web repositories. The crawlers use the service to enrich the video metadata with the analysis results before indexing them. The VIA component is described in detail in D4.2.

**Lecture video fragmentation REST service** The lecture video fragmentation REST service, exposes a method which utilizes the textual information derived from a transcript of a lecture video, to extract meaningful textual cues such as phrases or terms that the original text contains. These cues are characteristic of the original

---

<sup>43</sup><http://methods.sagepub.com/>



text as they capture very concisely the essence and the meaning of that text. The transcript text is used as input to our method, which outputs a set of time boundaries of the video fragments. The method is accessible via a REST service, which takes the transcript file of a lecture video in DFXP format as input, and outputs the results of the procedure in JSON format. To call the lecture fragmentation REST service a GET request is issued:

```
GET http://160.40.51.36:8000/LTF?file=<dfpx_file_URL>
```

The output of the service is a JSON file containing the following fields:

**Slug** The slug name of the lecture

**Video** The video number of the lecture

**Duration** The duration of the lecture video

**Number of Fragments** The number of the calculated fragments

**Fragments** An array containing the list of the calculated fragments; each fragment contains 4 fields:

**id** The identifier of the fragment

**Start time** The time the fragment begins

**End time** The time the fragment ends

**Keywords** An array containing the top-10 extracted keywords of each fragment in descending order; each keyword contains 2 fields:

**Keyword** The keyword itself

**Relevance** The relevance score between the keyword and the corresponding fragment

**Lecture video keywords** An array that contains the top-10 extracted keywords of the entire lecture video in descending order; each keyword contains 2 fields:

**Keyword** The keyword itself

**Relevance** The relevance score between the keyword and the entire lecture

More details about the lecture video fragmentation REST service and its underlying technologies can be found in D3.3.

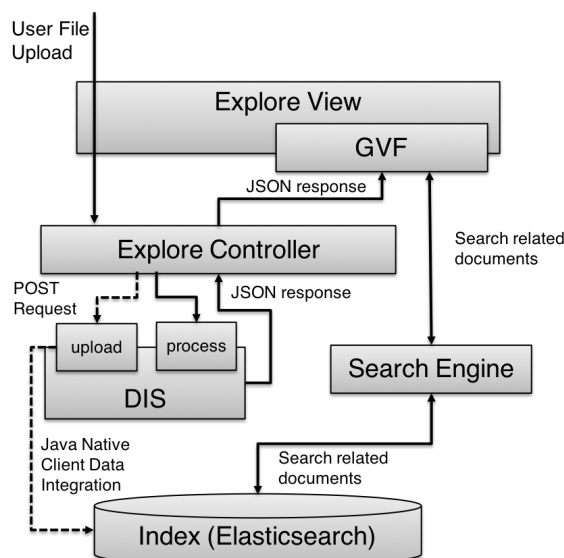
## 7.3 Explore

The *Explore* view of the MOVING platform provides simplified access to the innovative features of the MOVING platform. It enables users to visually analyse PDF files that can be transferred to the platform. To this end, the *Explore* functionality visualises the metadata included in the PDF files and the extracted entities from the full-text. Furthermore, connections between extracted entities and related documents in the MOVING platform are shown.

The *Explore* form is kept as simple as possible: it contains a mandatory field for the local path to the file to be transferred to the platform, a mandatory selection field for the language of the uploaded document, and a submit button. Optionally, a checkbox can be activated that enables the integration of the uploaded files into the MOVING index (see Figure 42). Upon submitting the files via the HTTP POST method, the *explore* controller of the MOVING platform sends an HTTP POST request to the REST interface of the Data Integration Service (DIS) (see Section 7.2.1). The DIS extracts then the metadata fields from the transferred file and analyses the full-text to extract further entities.

If the checkbox is not activated, which is the default case, the POST request is sent to the `/pdf-process` interface of the Data Integration Service. This way, neither the DIS nor the platform stores any permanent copies of the uploaded files or the extracted metadata and entities. When the checkbox is activated, the POST request is sent to the `/pdf-upload` interface of the DIS. Therefore, the extracted metadata, the extracted entities, and the full-text will be integrated into the MOVING index as a new document. By default, each submitted file is listed in the search facet *data collection* as *user uploads*. The uploaded files are immediately accessible to all registered users of the platform via the search interface (see Section 3.1). Figure 41 gives an overview of the involved components and how they interact for the *Explore* functionality.

Both REST interfaces of the DIS respond with JSON files following the common data model. In order to avoid unnecessary data overhead, the JSON response does not include abstracts or full-texts. The *explore* controller of the MOVING platform then sends the received response to the Graph Visualisation Framework



**Figure 41:** The interaction of the various components of the MOVING platform for the *Explore* functionality

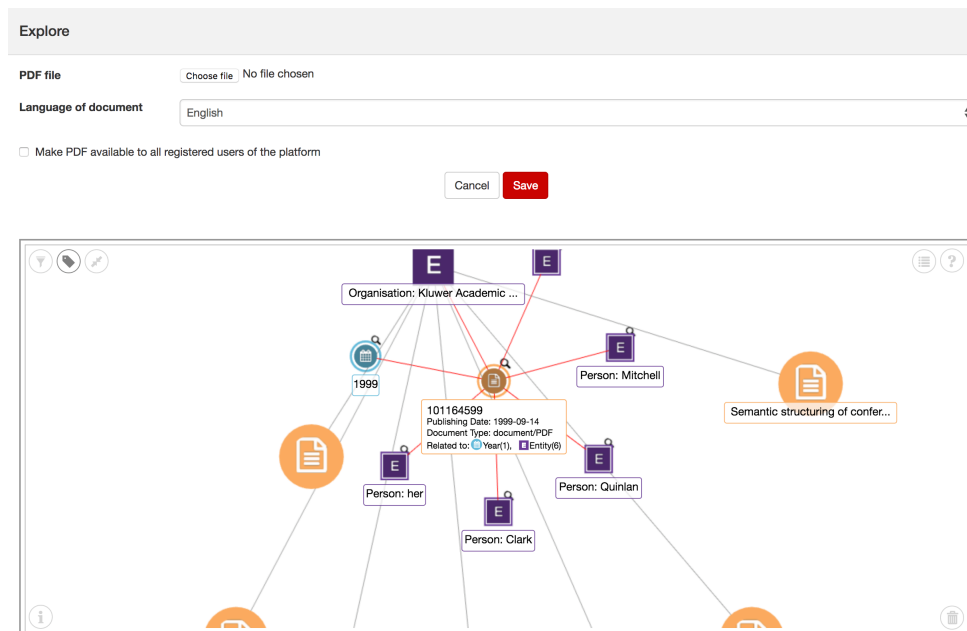
(GVF), which in turn renders the metadata and extracted entities from the JSON response using the Concept Graph Visualisation.

Furthermore, the uploaded files are connected to the most relevant documents in the MOVING index using a designated interface of the Concept Graph. This interface consists of a single function called `gvfExplore` which expects an array of objects as a parameter. These objects, which represent the JSON response documents of the uploaded files, follow the common data model. They contain all the metadata and extracted entities from the full-text of the uploaded documents. This document representation resembles the regular search results, when retrieved via the MOVING platform's search engine. To visualize the response from the DIS, the `gvfExplore` function performs three steps to retrieve additional results based on the file.

Firstly, (1) the ten most occurring entities per type (*organization*, *location*, *person*) are identified. The entities provided by the Data Integration Service include an array of positions, where they appeared in the full-text. Thus, we can count the number of occurrences by counting the number of positions in the full-text.

Secondly, (2) the extracted entities are used as search queries for the MOVING search engine which in turn delivers the related search results. The results are then ranked by the number of different entities covered by each document and the number of occurrences of each entity within the document.

As the last step, (3) the top 50 retrieved documents are visualised in the Concept Graph alongside the original JSON documents generated from the uploaded files. The document nodes representing the uploaded documents in the graph, and the nodes of the properties and entities of such documents have additionally a magnifying glass attached to them (see Figure 42). This magnifying glass makes it easier for the user to discern which nodes belong to the uploaded documents, and which were retrieved by the MOVING search engine.



**Figure 42:** The UI and the Concept Graph visualization of the uploaded documents: The nodes with the magnifying glass are from the uploaded document, while the ones without it are retrieved by the search engine



## 8 Conclusion

This report shows the final implementation of the MOVING platform prototype and gives an update of the Deliverable D4.2 "Initial responsive platform prototype, modules and common communication protocol" (Gottfried, Pournaras, et al., 2017). We showed how the platform was improved within the last months, which features were enhanced and which components were newly integrated. Both, the overall platform architecture and responsive design got updated to reflect this changes in functionality and user experience until its finished version.

To this end, we extensively updated and extended the MOVING web application with its search interface and novel results visualizations, community features and learning environment. The Adaptive Training Support was revised and comprises of the established *Learning-how-to-search* and a separate *Curriculum Reflection* widget. They both help users to explore the platform functionalities more easily by showing either statistics about their platform use or in finding information by providing a learning guidance. The Recommender System widget was added to the search page to support users in quickly accessing relevant multimedia resources available to the MOVING search engine. It is based on the novel ranking algorithm HCF-IDF. Part of the input for the ATS and RS is coming from the advanced user interaction tracking system which captures interaction data from the users of the platform. Finally, we incorporated five data processing components to improve the data quality stored in the index by validating documents before integrated, removing document duplicates, disambiguate equal document author names and enrich the index data by additional context information of documents or concepts of videos. All remaining data acquisition components and its interfaces were brought to a finalized version to import appropriate external multimedia content to the index.

In summary, we showed how the final MOVING platform prototype has been built and how its integrated training and working environment is capable of generating data-savvy information professionals and enabling users to improve their information literacy skills.

## References

- Apaolaza, A., Bienia, I., Maas, A., Wiese, M., Günther, F., Barthold, S., & Vigo, M. (2018). *Deliverable 1.3: Initial evaluation, updated requirements and specifications* (Tech. Rep.). MOVING.
- Apaolaza, A., Gledson, A., Fessler, A., Bienia, I., Apournaras, A., Blume, T., ... Vigo, M. (2019). *Deliverable 1.4: Final implementation of user studies and evaluation* (Tech. Rep.). MOVING.
- Apaolaza, A., Harper, S., & Jay, C. (2013). Understanding users in the wild. In *Proc. of the 10th international cross-disciplinary conference on web accessibility* (pp. 13:1–13:4). Retrieved from <http://doi.acm.org/10.1145/2461121.2461133> doi: 10.1145/2461121.2461133
- Apaolaza, A., & Vigo, M. (2017, June). WevQuery: Testing hypotheses about web interaction patterns. *Proc. ACM Hum.-Comput. Interact.*, 1(EICS), 4:1–4:17.
- Bienia, I., Fessler, A., Günther, F., Herbst, S., Maas, A., & Wiese, M. (2017). *Deliverable 1.1: User requirements and specification of the use cases* (Tech. Rep.). MOVING.
- Blume, T., Bösch, F., Galke, L., Saleh, A., Scherp, A., Schulte-Althoff, M., ... Gottron, T. (2017). *Deliverable 3.1: Technologies for moving data processing and visualisation v1.0* (Tech. Rep.). MOVING.
- Blume, T., & Scherp, A. (2018). Towards flexible indices for distributed graph data: The formal schema-level index model fluid. In G. Klassen & S. Conrad (Eds.), *Proceedings of the 30th gi-workshop Grundlagen von Datenbanken, Wuppertal, Germany, May 22–25, 2018*. (Vol. 2126, pp. 23–28). CEUR-WS.org. Retrieved from <http://ceur-ws.org/Vol-2126/paper3.pdf>
- Collyda, C., Mezaris, V., Herbst, S., Grunewald, P., Köhler, T., Fessler, A., ... Skulimowski, A. (2017). *Deliverable 6.2: Data management plan* (Tech. Rep.). MOVING. Retrieved from [http://moving-project.eu/wp-content/uploads/2016/10/moving\\_d6.2\\_v1.0.pdf](http://moving-project.eu/wp-content/uploads/2016/10/moving_d6.2_v1.0.pdf)
- Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-n recommendation tasks. In *Recsys '10* (pp. 39–46). ACM.
- Crestani, F. (1997). Application of spreading activation techniques in information retrieval. *Artificial Intelligence Review*, 11(6), 453–482.
- Fessler, A., Thalmann, S., Pammer-Schindler, V., Saleh, A., Nishioka, C., Scherp, A., ... Günther, F. (2017). *Deliverable 2.1: Initial conceptual framework-curricula and technical prototypes for adaptive training support* (Tech. Rep.). MOVING. Retrieved from [http://moving-project.eu/wp-content/uploads/2017/04/moving\\_d2.1\\_v1.0.pdf](http://moving-project.eu/wp-content/uploads/2017/04/moving_d2.1_v1.0.pdf)
- Goossen, F., Ijntema, W., Frasinca, F., Hogenboom, F., & Kaymak, U. (2011). News personalization using the CF-IDF semantic recommender. In *Wims '11* (pp. 10:1–10:12). ACM.
- Gottfried, S., Grunewald, P., Pournaras, A., Collyda, C., Fessler, A., Hasitschka, P., ... Scherp, A. (2017). *Deliverable 4.1: Definition of platform architecture and software development configuration* (Tech. Rep.). MOVING. Retrieved from [http://moving-project.eu/wp-content/uploads/2017/02/moving\\_d4.1-v1.0.pdf](http://moving-project.eu/wp-content/uploads/2017/02/moving_d4.1-v1.0.pdf)
- Gottfried, S., Pournaras, A., Collyda, C., Mezaris, V., Backes, T., Wertner, A., ... Saleh, A. (2017). *Deliverable 4.2: Initial responsive platform prototype, modules and common communication protocol* (Tech. Rep.). MOVING.
- Gottron, T., Scherp, A., Kray, B., & Peters, A. (2013). LODatio: using a schema-level index to support users infinding relevant sources of linked data. In *Proceedings of the 7th international conference on knowledge capture* (pp. 105–108). Banff, Canada: ACM. doi: 10.1145/2479832.2479841
- Günther, F., Barthold, S., Bienia, I., Maas, A., Wiese, M., Fessler, A., ... (JSI, T. Z. D. (2018). *Deliverable 2.2: Updated curricula and prototypes for adaptive training support and introductory moving mooc for community building* (Tech. Rep.). MOVING. Retrieved from [http://moving-project.eu/wp-content/uploads/2018/03/moving\\_d2.2\\_v1.0.pdf](http://moving-project.eu/wp-content/uploads/2018/03/moving_d2.2_v1.0.pdf)
- Günther, F., Barthold, S., Herbst, S., Zawidzki, J., Helbig, A., Bienia, I., ... Draksler, T. Z. (2019). *Deliverable 2.3: Final conceptual framework, curricula and moving mooc for community building* (Tech. Rep.). MOVING.
- Jett, J., Nurmikko-Fuller, T., Cole, T. W., Page, K. R., & Downie, J. S. (2016). Enhancing scholarly use of digital libraries: A comparative survey and review of bibliographic metadata ontologies. In *Proc. of the JCDL* (pp. 35–44). ACM. doi: 10.1145/2910896.2910903
- Kapanipathi, P., Jain, P., Venkataramani, C., & Sheth, A. (2014). User interests identification on Twitter using a hierarchical knowledge base. In *Eswc*. Springer.
- Meusel, R., Bizer, C., & Paulheim, H. (2015). A web-scale study of the adoption and evolution of the schema.org vocabulary over time. In *WIMS* (pp. 15:1–15:11). ACM.
- Nishioka, C., & Scherp, A. (2016). Profiling vs. time vs. content: What does matter for top-k publication recommendation based on twitter profiles? In *Jcdl '16* (pp. 171–180). ACM.

- Salton, G., Wong, A., & Yang, C.-S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11), 613–620.
- Vagliano, I., Abdel-Qader, M., Blume, T., Bösch, F., Galke, L., Saleh, A., ... Mutschke, P. (2018). *Deliverable 3.2: Technologies for moving data processing and visualisation v2.0* (Tech. Rep.). MOVING.
- Vagliano, I., Blume, T., Galke, L., Mai, F., Saleh, A., Pournaras, A., ... Mutschke, P. (2019). *Deliverable 3.3: Technologies for moving data processing and visualisation v3.0* (Tech. Rep.). MOVING.
- Vandenbussche, P., Atemezing, G., Poveda-Villalón, M., & Vatan, B. (2017). Linked open vocabularies (LOV): A gateway to reusable semantic vocabularies on the web. *Semantic Web*, 8(3), 437–452. doi: 10.3233/SW-160213