Ubiquitous Access to Digital Cultural Heritage

CHRISTIN SEIFERT, University of Passau
WERNER BAILER and THOMAS ORGEL, Joanneum Research
LOUIS GANTNER, Archeology and Museum Baselland
ROMAN KERN and HERMANN ZIAK, Know-Center GmbH
ALBIN PETIT and JÖRG SCHLÖTTERER and STEFAN ZWICKLBAUER and MICHAEL GRANITZER, University of Passau

The digitization initiatives in the past decades have lead to a tremendous increase of digitized objects in the cultural heritage domain. Although digitally available, these objects are often not easily accessible for interested users because of the distributed allocation of the content in different repositories and the variety in data structure and standards. When users search for cultural content they first need to identify the specific repository and then need to know how to search within this platform (e.g., usage of specific vocabulary). The goal of the EEXCESS project is to design and implement an infrastructure that enables ubiquitous access to digital cultural heritage content. Cultural content should be made available in the channels users habitually visit and be tailored to their current context without the need to manually search multiple portals or content repositories. In order to realize this goal open-source software components and services have been developed which can either be used as an integrated infrastructure or as modular components suitable to be integrated in other products and services. The EEXCESS modules and components comprise (i) web-based context detection, (ii) information retrieval-based, federated content aggregation, (iii) metadata definition and mapping, and (iv) a component responsible for privacy-preservation. Various applications have been realized based on these components that bring cultural content to the user in content consumption and content creation scenarios. Content consumption is for example realized by a browser extension generating automatic search queries from the current page context and the focus paragraph and presenting related results aggregated from different data providers. A Google Docs add-on allows to retrieve relevant content aggregated from multiple data providers while collaboratively writing a document. These relevant resources then can be included in the current document either as citation, an image or link (with preview), without having to leave disrupt the current writing task for an explicit search in various content providers' portals.

CCS Concepts: •**Applied computing** → **Digital libraries and archives; Document searching;** *Document metadata;*

Additional Key Words and Phrases: search aggregation, user context detection, metadata harmonization

ACM Reference Format:

Christin Seifert, Werner Bailer, Thomas Orgel, Louis Gantner, Roman Kern, Hermann Ziak, Albin Petit, Jörg Schlötterer, Stefan Zwicklbauer, and Michael Granitzer. 2016. Ubiquitous Access to Digital Cultural Heritage. *ACM J. Comput. Cult. Herit.* 000, 0000, Article 000 (2016), 26 pages.
DOI: 0000001.0000001

The presented work was developed within the EEXCESS project funded by the European Union Seventh Framework Programme FP7/2007-2013 under grant agreement number 600601.

Author's address: Christin Seifert, Jörg Schlötterer, Albin Petit, Stefan Zwicklbauer, Michael Granitzer, University of Passau, Germany, email: {firstname.lastname@uni-passau.at}; Louis Gantner, Archäologie und Museum Baselland, Amtshausgasse 7, CH-4410 Liestal; email: louis@gantner.cc; Werner Bailer, Thomas Orgel, JOANNEUM RESEARCH – DIGITAL, Steyrergasse 17, 8010 Graz, Austria, email: {firstname.lastname}@joanneum.at; Roman Kern, Hermann Ziak, Know-Center GmbH, Inffeldgasse 13/6, 8010 Graz, Austria; email: rkern@know-center.at, hziak@know-center.at.

DOI: 0000001.0000001

1. INTRODUCTION

In the past three decades GLAMs (Galleries, Libraries, Archives and Museums) invested a lot of efforts in various *digitization activities* within their institutions. A primary motivation of this *digital wave* was the need to introduce "object management systems" like databases (also termed "collection management systems" or "inventory systems"). Additionally, cultural heritage institutions aimed at digitizing parts of their physical collections using technologies like digital photography or scans (e.g., for paper documents, paintings, sculptures, photographs, maps etc.) along with the describing metadata in order to provide easy access and at the same time preserve the object from physical destruction (preservation and access initiatives).

Growing technological capabilities and the availability of the Internet triggered the next phase in the year 2000. The GLAM institutions started to open up their collections for the World Wide Web with the goal to connect with peer institutions and users directly. The Open Access movement also included the cultural domain and therefore broadened the requirements for online collections in science and the humanities. The subsequent activities resulted in the deployment of different sorts of web-access services like web-portals. The need for standardization was evident and in fact still remains a challenge. While standards had already existed for many years for libraries, especially the museums and archives were and still are challenged with the adoption and introduction of standards for both museums management processes and metadata schemas (e.g., CIDOC CRM[1], LIDO[2], EDM[3]).

In summary, the digitization initiatives can be viewed as overlapping and interacting processes responding to internal and external requirements. This affects the domain of metadata management, technical infrastructure, software design and rights-management facing fast advances in technical evolution (Internet and social media). From a scientific point of view, the topics of digital curation, online management, provision and exploitation of digital resources are embraced by the term and the research area Digital Humanities, which covers both the traditional disciplines of the humanities and the latest developments in computing (e.g., data and text mining, visualizations, digital mapping etc.).

1.1 Problem Setting

The current landscape for cultural heritage data can be characterized as follows: On the one hand, there are data silos in large and small institutions, that have been made accessible through APIs and their own portals, such as the Public Library of America[4], the University Libraries of Switzerland[5], or Europe's aggregator of cultural content Europeana[6]. On the other hand, there are the culturally interested users, students, librarians or researchers, who want to access, search, and discover content of their interest within those data silos. These actors are characterized by their personal interest and current task. The distribution of content in various data silos requires these actors to (i) find the relevant access point to the data silo (if available), and (ii) search the content within the silo. General purpose search engines (e.g., Google) realize a federated search through all (indexed) repositories and try to help users find the content, but only partly address the problem for cultural content. First, those search engines are optimized for "main stream content", while cultural content resides in the so-called long tail of the Internet [Barabási et al. 2000]. Second, not all content is available for indexing, due to licensing and rights-management issues.

[1] http://www.cidoc-crm.org/, last accessed 18.3.2016
[2] http://network.icom.museum/cidoc/working-groups/lido/what-is-lido/, last accessed 18.3.2016
[3] http://pro.europeana.eu/page/edm-documentation, last accessed 8.8.2016
[4] http://dp.la/, last accessed 23.3.2016
[5] https://www.swissbib.ch/, last accessed 23.3.2016
[6] http://europeana.eu/, last accessed 23.3.2016

Thus, for a contextualized personalized discovery of cultural content, the following problems have to be solved: First, a single point of discovery is necessary, that searches all related data silos, which in turn requires an intelligent aggregation of results and a harmonization of metadata. Second, users should be able to access the content from within their current task, and results should be contextualized towards the task and personalized based on the user interests and knowledge.

1.2 Approach

To achieve the goal of ubiquitous access to cultural heritage, we propose the following conceptual idea: The single point of discovery is realized by a federated search and recommendation approach, accessing relevant data silos and integrating their content into a single result list (*federated aggregation*). Responses from all accessed data silos are harmonized with respect to their metadata (*metadata harmonization*). The request to the access point is injected into the users current whereabouts, e.g., blogging frameworks (*content injection*), following the principle of bringing the content to the user [Granitzer and Seifert 2016]. In order to ensure user acceptance, methods for individually retaining the users privacy are employed while offering the full range of personalization (*privacy preservation*). Figure 1 illustrates this approach. When browsing a Web page, e.g., Wikipedia, relevant results aggregated from different data providers (e.g., the German digital library and Europeana) are presented in a single result list. The search is performed automatically based on the terms available on the page, in this example the terms “Rembrandt” and “Titus van Rijn’.

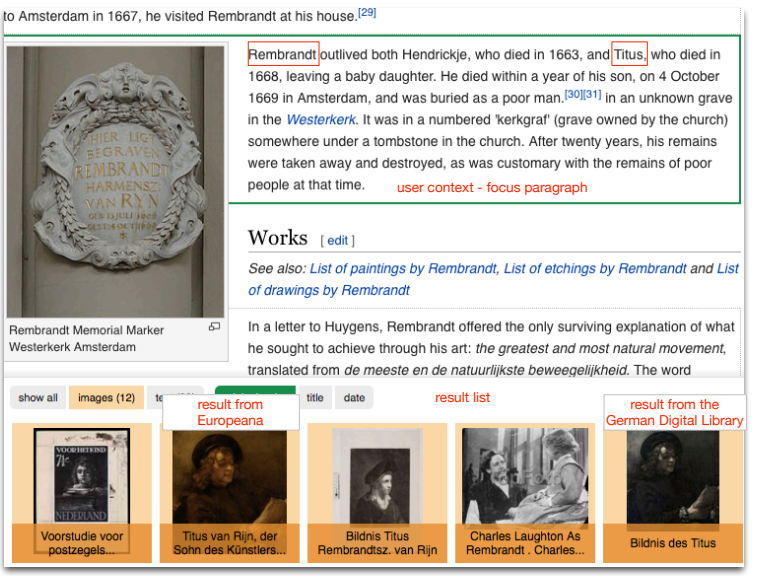

Fig. 1. Illustration of the proposed approach. Content is aggregated from different providers into a single result list and injected into the users’ current digital context (the Wikipedia page of Rembrandt).

In this article, we describe the EEXCESS[7] infrastructure for accessing cultural heritage content from the long tail of the Internet. This infrastructure comprises components for content aggregation from various sources, and content injection into various client platforms. Single components communicate with standard Web technologies and well-defined APIs, and can be used as stand-alone components with only minor modifications. The infrastructure and components are available as Open Source[8].

The article is organized as follows: In Section 2 we describe the overall architecture outlining the single components and their interaction. Section 3 provides more details and evaluation results for single components. Example application scenarios with a focus on content providers and content consumers of cultural heritage content are described in Section 4. Related work is discussed in Section 5. Section 6 presents a summary and outlook on future work.

[7]http://eexcess.eu, last accessed 18.3.2016
[8]https://github.com/EEXCESS, last accessed 18.3.2016

2. OVERALL ARCHITECTURE

The presented architecture is the basis for realizing (i) aggregation of various sources, (ii) provision of personalized content and (iii) dissemination of content into various channels. In this section we provide an overview of the architecture, the core components and their interactions. Further, we detail the workflow from user context detection to the provision of related resources.

An overview of the architecture is shown in Figure 2 with the conceptual components being described in the following. An overview of the repository locations of individual components is provided in Table I.

The **User Interface Clients** component is the conceptual component end-users directly interact with (more details see Section 3.4). The component contains interface elements for explicit search for related resources and/or mechanisms for implicitly issuing automatically generated searches (making use of the context detection component). The user interface is also responsible for presenting the retrieved resources, and might contain complex mechanisms for result interaction (e.g., advanced filtering tools). Available clients are, e.g., a Google Docs plugin (cf. Section 4.4), an extension to the Chrome browser (cf. Section 4.3), a Wordpress plugin, and a plugin for the Moodle e-Learning platform.

The **Context Detection** component is responsible for observing the user behavior on the client and generating user interest profiles and search queries. This component is available as a library and can be easily integrated in any user interface client as long as this client supports standard web-technologies (HTML, JavaScript). Context detection is described in detail in Section 3.4.2

The **Privacy Proxy** is responsible for removing, hiding and perturbing user sensitive information. All communication between client and server should go through the privacy proxy in order to ensure user's privacy. This component is described in detail in Section 3.5

The **Federated Aggregation** component is responsible for distributing the query to the **Partner Recommenders** (which access their own **Content Database**), retrieving the results from single partners, aggregating the result list and delivering the results in a common format to the client components. This component is described in detail in Section 3.2.

Metadata Harmonization contains the definition of a unified metadata model and a mapping component responsible for mapping partner's meta data to the unified data model. This component is described in Section 3.1. The **PartnerWizard** allows content-providers without programming skills to add their system to the EEXCESS ecosystem by creating partner recommenders via a guided user interface. This component is used in the partner setup phase and is explained in more detail in Section 3.3.

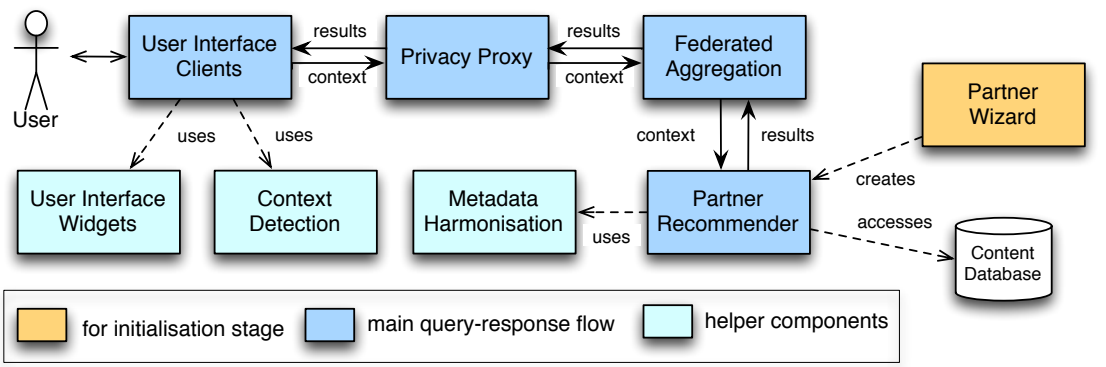

Fig. 2. Overview of the infrastructure outlining core components. The main query workflow is indicated with solid lines, helper calls are indicated with dashed lines.

Table I. Client applications and component overview.

Component	Source Code URL[1]	Language	Comment
Client Applications			
Chrome extension	*GHE*/chrome-extension	HTML, CSS, JavaScript	available from Chrome web store purl.org/eexcess/clients/chrome-extension
Google Docs plugin	*GHE*/gdocs-plugin	HTML, CSS, JavaScript	available from Google App Store purl.org/eexcess/clients/googledocs-plugin
Wordpress plugin	*GHE*/wordpress-plugin	HTML, CSS, JavaScript	available in Wordpress Store, plugin name ”EEXCESS“
Moodle plugin	*GHE*/MoodleServerPlugin, *GHE*/MoodleAttoEditorPlugin	PHP	installable versions available from purl.org/eexcess/install/moodle-server-plugin and purl.org/eexcess/install/moodle-atto-editor-plugin
Components			
Context Detection	*GHE*/c4	JavaScript	C4 library available via Bower[2]
User Interface Widgets	*GH*/visualization-widgets	HTML, CSS, JavaScript	provides simple result lists and more complex visualizations as modules
Privacy Protection	*GHE*/peas	JavaScript	client components for privacy preservation
Privacy Proxy	*GHE*/privacy-proxy	Java	server components for privacy preservation
Federated Aggregation	*GHE*/recommender	Java	includes federation and single partner recommenders
Partner Wizard	*GHE*/EEXCESS/PartnerWizard	Java	consists of a user interface and the partner recommender generating sub-component
Metadata Model	eexcess.eu/schema/eexcess.owl	OWL	metadata model (based on EDM and W3C PROV)
Metadata Quality Tools	*GHE*/data-quality	Java	library to determine data quality metrics

[1] *GHE* abbreviates the domain github.com/EEXCESS [2] http://bower.io

The *general workflow* for injecting personalized, cultural content into the client is as follows: The client detects the user's information need based on the current user context (e.g., the edited document, the visited web page, the browsing history). The client generates a search query and a user interest profile, which is sent through the privacy proxy to the federated aggregation component. The privacy proxy ensures user's unlinkability and indistinguishability. The federated aggregation component distributes the search query and the user profile to the partner recommenders, taking their availability and vocabulary into account. The partner recommenders retrieve relevant results from their content database and return them to the aggregator in a harmonized format making use of the metadata harmonization component. The aggregator integrates the results into a single result list, reranking based on the user profile and criteria such as diversity. The final result list is then returned to the client and presented to the users.

3. COMPONENTS

This section describes the components in more detail, starting with the metadata integration (in Section 3.1) and federated aggregation component in Section 3.2. The PartnerWizard, our solution for automatic integration is presented in section 3.3. Section 3.4.1 then describes the user interface components, including the widgets for displaying content and the modules for detecting user context for automatic search. The solution on privacy protection in the architecture is described in Section 3.5.

3.1 Metadata Harmonization

EEXCESS uses a federated model to collect query results from different data providers (each in the format returned from the provider's API), to combine and rerank these results and provide them to a range of client applications. Clearly, combining and ranking search results can only be effectively implemented, if at least a basic level of metadata harmonization between the data from different providers is achieved (e.g., mapping the title and year for each object in all sources to a unified metadata field). Thus, both a common metadata model and an approach for automatically transforming metadata from the format in which they were provided to the common model are needed. As EEXCESS aims to make also the long tail of niche content available, many records for a large set of assets will only be infrequently accessed. Thus transforming all content in advance is not useful, but the transformation happens during the retrieval of resources. Caching would be possible, but may be of limited use for the records that are only infrequently accessed, and would require creating the centralized infrastructure to store and manage the cached data.

The requirements on metadata models and approaches from the EEXCESS use cases differ partly from those of content provision to cultural heritage portals such as Europeana, most notably in three aspects. First, the metadata comes from a potentially large pool of different providers, so that a curated data provision step is not feasible. Second, the metadata of resources is not only automatically enriched from linked data sources, but also linked with social media information. Third, the information collected from different providers is selected by taking the user context into account and is presented and processed in a way that is specific for each of the use cases. This means that the model needs to provide high flexibility to cover these diverse requirements and to provide support for detailed provenance metadata due to risk of incompleteness and inconsistency from automatic mapping and enrichment.

3.1.1 *Metadata model.* Some of the models mentioned in Section 1 are candidates for the common metadata model. As EEXCESS deals with data from different domain, models that support cross-provider integration, but are limited to a specific domain (e.g., museums only) are not considered. Thus, the CIDOC Conceptual Reference Model (CRM) [ISO 21127 2014] and the Europeana Data Model (EDM) [Europeana Foundation 2015] are considered. CIDOC CRM is a quite comprehensive model to describe cultural heritage objects and their relations, and to integrate data models of different institutions. EDM has been developed as model for data provision to Europeana, and a number of providers already support this model. Both models supports multiple views to one real world resource, such as documentation of the object from different sources.

For EEXCESS, EDM was chosen because some metadata of our data providers are already in this format, and it is more basic, and thus it may be easier to define mappings. EDM lacks support for provenance metadata, thus we complement it by using the W3C PROV ontology (PROV) [Lebo et al. 2013]. The EEXCESS tools extract metadata from the original resource and use it for enriching their description, thus adding a new set of annotations. Following the linked data paradigm, EDM also defines a small set of core properties, while leaving room to add domain-specific properties from appropriate metadata schemes. The information from all sources is in the EDM description, but can still be separated by its source if needed. While this is possible, it is also important to provide more details about the provenance of each of these annotations, e.g., if they have been created manually or automatically, their creation and modification dates, the organization creating the annotation, etc. The core model of EDM does not include provenance information, and needs to be complemented by another metadata format. The use of the PROV vocabulary enables describing the different annotations added to an object in more detail. A client application can thus make the appropriate use of the available annotations and their provenance metadata.

3.1.2 *Mapping Definition.* As mentioned before, the tools for metadata transformation must be applied automatically on the fly, once such a mapping has been properly configured. We have therefore developed a tool with a focus on easy configurability of mappings, which can then be executed by an automatic service. A mapping can be defined from scratch, or a basic mapping created using the PartnerWizard (see Section 3.3) can be used as a starting point for refinement.

Our metadata mapping approach is based on a mapping ontology representing mapping information in order to solve a mapping problem between a pair of metadata formats. Elements and attributes of the metadata formats involved in the mapping can be linked by using drag & drop. Their data types can be specified (if not defined in an XML schema of the format). In addition, a hierarchy of contexts can be defined, in order to link metadata elements to the appropriate entity, for example, to discriminate titles of a series, a book and an article, which may be found in the source metadata.

All required mapping parameters are derived from this ontology and mapping instructions are created. The mapping instructions are encoded into an XSL document [Michal 2007], which represents the transformation between an input XML document and an output document. The mapping configuration tool includes the mapping quality approaches described below, so that users can obtain direct feedback on the applied mapping, using a set of sample files. Further details on the mapping tool and approach can be found in [Orgel et al. 2015].

3.1.3 *Quality Assessment.* In order to ensure the quality of the metadata processed and returned by the EEXCESS system, we need to address both the *source metadata quality*, i.e., the quality of records returned from a particular data provider, and the *mapping quality*, i.e., the completeness and fidelity of the metadata in the target common data model[9]. Metadata quality dimensions to be considered are completeness (the elements are provided and filled), accuracy (no syntactic errors), consistency (correct semantics and no logical errors), availability (referenced resources such as vocabularies can be accessed) and processability (structured and machine readable data). Further details on the implemented metrics and results on data sets from different providers can be found in [Orgel et al. 2016].

For assessing the mapping quality of metadata, the dimensions completeness and consistency can be considered. The metadata formats used differ a lot among the various data providers, and require appropriate mappings. Generally, these mappings may not be lossless. If some source formats are limited in their expressiveness, some loss of information or imprecision in mapping would be unavoidable. The aim of mapping quality assessment is thus to quantify the loss of completeness and consistency of metadata documents resulting from mappings, in order to provide feedback to the experts defining the mapping, and to keep this loss as small as possible. Due to the scale of the problem expert assessment of mappings for a range of formats and a significant number of metadata documents is not feasible. Thus, an automated method to assess the quality of mappings has been implemented and integrated into the tool for configuring mappings described above. The method is based on round trip mappings, i.e., testing the mapping from the source format to another format by performing this mapping and mapping back to the source format, so that the input and output of this process (both represented in the source format) can be compared. Two different variants of round trip mappings are considered. The first variant considers only the internal intermediate conceptual representation of metadata properties, while the second variant also includes a specific target metadata format. In the first case we would expect that input and output documents are identical if the mapping is correct and complete, while in the second case the expected loss or imprecision between a pair of formats needs to be specified by an expert once per metadata format in the configuration tool. The mapping quality assessment tools pro-

[9]Note that source metadata quality includes quality dimensions that support the findability of records, but does not address the retrieval performance for specific queries

vide feedback to a user defining a mapping, and can be run directly in the tool after any modification of the mapping. A more detailed description of the implemented quality assessment methods and results on actual queries can be found in [Höffernig et al. 2015].

3.2 Federated Aggregation

The task of the federated aggregation component is to take a user's query as input and create a single result list containing information from all data providers. This task is also called *aggregated vertical search in an uncooperative setting* [Lu and Callan 2005]. Therefore, the user's query needs to be adapted to the specific data providers (i.e., their vocabulary, their query language) and the results from the individual data providers need to be aggregated into a single result list. The three main challenges of this task are: (i) the queries themselves may be heterogeneous, e.g., may vary in length, (ii) the data providers behavior can only be indirectly steered, (iii) the results returned from the data providers vary greatly in terms of content and meta-data. Given the context of the EEXCESS project, the final aggregated result list should contain results, which are related to the user's information need, diverse in nature and also incorporate serendipitous elements. In addition, the whole process should return results in a short period of time, to keep the time the user waits for results as short as possible, i.e. the latency should be low.

3.2.1 *Source Selection.* The first part of the Federated Recommnender is the so called source selection module. Source selection is the task to find the set of matching data providers for any given query [Shokouhi and Si 2011]. This is motivated by the insight that not all data providers are equally suited to handle an user's information need. Therefore, we provides algorithms that try to predict which data providers are particularly well suited for a given user query.

The core of the source selection module is the category mapping function. This function takes a set of terms as input and produces a weighted set of categories. The input might be generated from a user's query or a document, as it is retrieved from one of the data providers. As soon as a data provider joins the EEXCESS system, the content of the data provider is probed. The module syntheses a set of automatically generated queries, resembling ambiguous user's queries, which are then sent to the newly registered data provider. The returned results are analyzed by invoking the category mapping function. After an amount of processed probing queries, a profile for the data provider can be inferred.

When a query is sent to the federated aggregation component, the category mapping function is applied to compute the categories for the query. Next the profiles of all registered data providers are matched against the query category. If the overlap between a data provider profile and the query category is too low, the data provider is completely excluded. This reduces the necessary resource consumption and helps to achieve a lower latency. In addition to the source selection based on categories, similar technique is also applied on the language of the query, the age span and other criteria.

3.2.2 *Query Processing.* The federation operates in a so-called uncooperative setting, i.e. the data providers are effectively treated as black boxes. The only way to steer the behavior of the individual data providers is via the issued queries. Therefore, each query is rewritten individually for each data provider to achieve optimal results. The federation component already provides a wide variety of different query generation strategies, ranging from disjunction queries to queries featuring data provider specific syntax elements.

In addition, the queries themselves may be heterogeneous. Therefore, we provide functionality to deal with short queries as well as long queries. For short queries additional, related query terms are added to the original query. This technique is known as query expansion. After thorough evaluation [Ziak and Kern 2015] we settled for a local query expansion technique based on pseudo relevance

feedback using an existing knowledge base. Our query expansion technique turned out not only to help in short queries, but was also shown to increase the diversity [Rubien et al. 2015].

For long queries the federated aggregation component foresees techniques to split the query into coherent smaller queries. These smaller queries are individually issued to the data providers and the results are combined. The most promising approach is based on a combination of query segmentation [Hagen et al. 2012] and the use of DBPedia to detect coherent concepts.

3.2.3 *Result List Aggregation.* Once results from all relevant data providers are retrieved they need to be combined into a single, ranked result list. At first, available metadata from the results are used to filter out unwanted content, e.g., to remove results in languages not understood by the user. Next the results are scanned from near-duplicates to avoid a situation where a user is confronted with a set of virtually identical results. For each of the data providers' result lists the overlap with the original query is computed. This allows to rank the individual data providers according to how well they matched the user's information need. The computed weight is used to control the likelihood of a data provider's result to be added to the final, aggregated result list. This way a weighted round robin scheme is implemented. In order to achieve a degree of diversity within the search result, results generated via our query expansion method can be additionally mixed into the result list. The same technique is optionally applied to trigger a serendipity effect. If available, the query is additionally expanded using the long term interest of a user. The aggregated result list is finally passed to the calling component.

3.2.4 *Runtime Behavior.* In order to assess the scalability of the federation one first needs to identify the main bottlenecks. Therefore, we conducted a number of experiments to measure the runtime behavior of the federation component and its sub-components. In this experiments we always submitted the same number of queries, but with different degree of parallelism. In Figure 3 an overview is given for a range of increasingly more parallel calls to the federation component. As expected, the total runtime decreases with more requests being conducted in parallel. At about 2000 parallel calls, some of the queries took longer than a predefined threshold, thus the respective data provider is ignored in the returned result list. Overall, one can state that the federation component itself is not the main bottleneck, instead the runtime behaviour is dominated by the individual data providers.

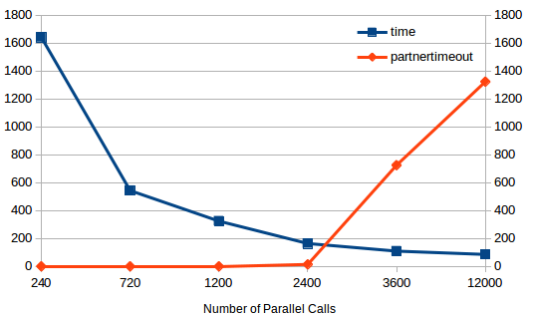

Fig. 3. Scalability behavior of the federation with 3 partners (right axis number of time-outs, left axis total time in sec). At about 2000 parallel calls the first time-outs occur. Overall, the scalability of the whole system is dominated by the latency of the data providers.

3.3 Data Provider Setup

For the success of EEXCESS the number and diversity of data providers is an important criteria. Many organizations provide an API in order to make their data available, but do not have the technical expertise in house to implement the necessary software components to connect to the EEXCESS system. We have thus developed a tool called PartnerWizard, which automatically creates the recommender component for a data provider that connects to the federation component and thus integrates with the EEXCESS framework. The PartnerWizard guides the user through a few easy and interactive configuration steps. The user specifies the URL of the endpoint of the data provider's API, and the XPath of the root of a record and to basic fields needed for metadata mapping. All parameters can be immedi-

ately tested with sample queries in order to see how changes to the configuration impact the handling of an actual response from the data provider's system.

The Recommender Query Generation is a further step in the PartnerWizard, which enables the user to optimize the search results for the data provider. This is also a web-based process, where different types of query processing strategies with multiple search terms and different strategies for combining search results are compared. The user selects the preferred result for some example queries, which then adjusts the source code in the recommender accordingly.

The PartnerWizard has been implemented using Apache Maven[10]. In particular, the archetype feature of Maven is used to create a template that is turned into the source code of the recommender for the data provider. As Maven fetches all required dependencies, the ready-to-use Java classes can be built automatically. Along with the implementation, unit tests are generated from the example data entered by the user during the configuration process.

3.4 User Interface Clients

The user interface clients component is the conceptual end-user interface, which can be instantiated by various applications. Several instantiations were already implemented within EEXCESS, like for example the Google Chrome Extension (c.f. section 4.3). However we do not aim to develop a client for each and every use case. Instead, we provide a modular platform, which allows the quick creation of new, and easy integration into existing client solutions. This platform consists of two major parts: modules for context detection and query extraction (c.f. section 3.4.2) and widgets to display results and interact with them (c.f. section 3.4.1).

3.4.1 *User Interface Widgets.* The widgets to display and interact with the results are self-contained web pages, which can be easily included via an iframe. With this architecture, developers do not need to care about configuring the widgets, but only need to implement the interface to communicate with them. The communication takes places via the Web-Messaging API[11]. Most importantly, a developer needs to implement the interface to pass retrieved results to the widget. As the interface is consistent across all widgets, they are easily interchangeable, once the interface is implemented.

Besides a basic search result view, as depicted in the screenshot in Figure 7, more advanced tools and alternative result views are available. Among them are the visualization dashboard [Tschinkel et al. 2015], providing several views on, and filter possibilities of the results, and the FacetScape [Seifert et al. 2014], a widget for result space exploration. We briefly describe the latter as an example for an alternative or additional widget to the basic search result view. Figure 4 shows a screenshot of the filter area of the FacetScape for a particular result set. In the screenshot, Swedish and English have been chosen in the *LANGUAGE* facet, hence only results in those languages are displayed. The selection has an impact on the available selection filters in other facets. Filters, which are not available in the current selection are displayed with less opacity. For example, *Swissbib* in the *PROVIDER* facet is grayed out, as no results in English or Swedish are available from this provider, opposed to *Europeana*, for which 23 results are available (indicated by the superscript number). When hovering over a selection filter, the amount of results that match this selection in the current set is displayed along with the total amount of results in the current set and the amount of results that would be removed or added by this filter. This situation is depicted for the *IMAGE* filter in the *MEDIATYPE* facet in the screenshot: 18 results in the currently selected set of 38 results are images and when applying this filter, 20 results would be removed from the current set.

[10] https://maven.apache.org/, last accessed 18.3.2016
[11] http://www.w3.org/TR/webmessaging, last accessed 18.3.2016

Fig. 4. Screenshot of the filter area of the FacetScape. Selection filters applied for English and Swedish in the language facet.

3.4.2 *User Context Detection and Query Generation.* Detecting the user's current context and generating a query out of it are the first steps on the way to retrieve relevant results. We provide modules for these steps, packaged in a Bower[12] repository called *C4* and adhering to the Asynchronous Module Definition[13]. This way, any desired modules can be loaded on demand. For example, in the Google Docs add-on (cf. Section 4.4), the context is defined by the editor's typed text and the context detection module does not apply.

3.4.2.1 *Context Detection.* In a web setting, the observable user context encompasses the web pages visited and additional information like the user's location. We focus on the visited web pages and hence account for the textual content of web documents in the context detection for query generation. Information like the user's location can be used to populate the user profile, while the user is in control about which information of this profile is disclosed (c.f. section 3.5). The textual context of visited web pages can be subdivided into five levels of granularity (from fine-grained to coarse): *terms*, *phrases*, *paragraphs*, *pages* and *sessions* [Schlötterer 2015]. To generate queries and present results according to the current context, we focus on the paragraph and phrase level. The session level is utilized to personalize the generated queries (c.f. section 3.4.2.3). Browser events, such as mouse movements or scroll position yield only limited accuracy in determining the phrase currently read by the user [Hauger et al. 2011]. Hence, we rely on explicit user interaction on the phrase level, i.e. a text selection, which is a strong indicator for reading focus [Hauger et al. 2011]. Treating terms as single term phrases, the term level is also covered by the phrase level context detection (which would also cover the page level, when a user selects the whole page content, but this is unlikely to happen). On the paragraph level, we first separate actual text passages from navigational menus, advertisements, etc. and afterwards identify the currently focused paragraph. In order to minimize the computational effort due to limited resources, we apply a heuristic based on a fixed length threshold of DOM text nodes for the extraction of paragraphs. To determine the focused paragraph, we first limit the set of candidate paragraphs to those currently in the viewport. Afterwards, the focused paragraph is determined by scroll position and highlighted. If the user does not agree with this selection, she can change it with a simple click on another paragraph. This paragraph will then be marked as focused until it leaves the viewport (due to scrolling).

[12] http://bower.io
[13] https://github.com/amdjs/amdjs-api/blob/master/AMD.md, last accessed 18.3.2016

3.4.2.2 *Query Generation.* We cannot influence the behavior of different providers' search engines integrated via the federation component. Therefore, we treat the search engines as black boxes and focus on the query side of retrieval. Since the context in our setting is defined by natural language text, e.g., a paragraph in the web site, keywords provide a compact representation of the paragraph and hence make up natural candidates to construct a query for related results. Keyword extraction algorithms with reasonable performance are readily available [Mihalcea and Tarau 2004; Rose et al. 2010]. However, over 71% of (user generated) search queries contain named entities [Guo et al. 2009] and Wikipedia page titles (i.e. named entities) have been shown to be beneficial to query segmentation [Hagen et al. 2012], a task for query optimization. In addition, named entity extraction can be seen as some kind of keyword extraction task, as the original text is represented by a smaller set of terms. For these reasons, we base our query construction approach [Schlötterer et al. 2016] on named entities.

In pretests with different digital library repositories, we discovered that in particular repositories that perform the search over meta data of repository objects return a very broad range of results. Moreover, it occurred that results, which were triggered by only a single keyword from the query suppressed results related to more keywords. We assume, that in those repositories, the keywords are combined via boolean OR queries. The problem hereby is that in most cases, the results, which were triggered by a single keyword only, did not fit the main topic of the text from which the query was constructed very well and hence were quite unrelated. To overcome this problem, we aimed for a solution that incorporates the overall topic and ensures it is well represented in the query, which resulted in a boolean query of the following form:

("main topic") AND ("keyword 1" OR "keyword 2" OR ...)

(Named) Entities located within a paragraph form the base for an underlying query. To extract these entities from a textual document we perform a (Named) Entity Annotation (NEA) that relies on two important subtasks: (Named) Entity Recognition and (Named) Entity Disambiguation. Entity recognition forms the first step of creating entity annotations. It identifies proper nouns (in the following denoted as surface forms) that can be linked to a semantic meaning. The task of entity disambiguation establishes links between identified surface forms and entities within a knowledge base (KB) and faces the problem of semantic ambiguity [Zwicklbauer et al. 2016b; 2016a].

To provide a robust NEA in terms of reliability and performance we apply the Named Entity Recognition and Named Entity Disambiguation system DBpedia Spotlight[14]. DBpedia Spotlight is one of the first semantic approaches (2011) and constitutes an entity-centric approach which is based upon DBpedia. Based on a vector-space representation of entities and using the cosine similarity, this approach has a public available web service. The service is able to recognize and disambiguate English and German language entities as determined in the request. Furthermore, we detect dates in documents and treat them like normal entities.

To determine the main topic of a paragraph we use Doc2Vec [Le and Mikolov 2014] as topic detection method. Generally based on Word2Vec, Doc2Vec produces a vector given a sentence or document. Hence, we use the entire input paragraph and infer a representative vector given a Doc2Vec model created on the Wikipedia corpus. We compare this vector with the vectors of the Wikipedia pages (entities) by computing the cosine similarity. The Wikipedia page (entity) with the highest similarity to the input paragraph represents the main topic. To significantly improve the performance we reduce the target entity set to those entities which have been annotated in the given paragraph.

[14]https://github.com/dbpedia-spotlight/dbpedia-spotlight/wiki, last accessed 18.3.2016

Table II. Evaluation Results of User Study for Context Detection and Query Generation

	Precision	Recall	F1-score	Extracted paragraph fit	Focused paragraph fit	Main topic fit
Automatic query	0.29	0.25	0.24	84%	65%	83%
User query	0.33	0.33	0.31			

3.4.2.3 *Query Adaption and Personalization.* The automatically generated queries are also displayed to the user and can be modified by the user. Possible modifications comprise filtering the keywords for locations or persons, adding or deleting keywords and editing the main topic. Keywords can be added either manually via textual input or via a text selection within the page. Similarly, the main topic can be set via a text selection within the page or via drag and drop of a keyword (manual editing is also possible).

In particular large paragraphs lead to a large amount of extracted keywords (named entities). To counter this fact, we subdivide large paragraphs into smaller sub-paragraphs and generate separate queries for each sub-paragraph. The sub-query with the highest overlap with the user profile is selected as query to be sent to the federated aggregation component. We took this approach as a sub-paragraph usually covers a particular aspect of the topic of the whole paragraph and by the highest overlap with the user profile, we deem this aspect to be the most interesting to the user. The user profile overlap is based on categories associated to the entities in the current and previously executed queries. The associated categories are provided by the category assignments in Wikipedia. For example, (amongst others) the category *Women in technology* is assigned to the entity *Ada Lovelace*. The categories of all previously executed queries for which the results have been viewed are stored in the user profile.

3.4.2.4 *Evaluation and Results.* The performance of context detection and query generation was evaluated in a user study with 77 participants. Each of the participants had to perform four tasks. In the first three tasks, user were free to choose a Wikipedia page from a predefined set of featured articles, while in the last task, the page was predefined and the same for all. Apart from this, the procedure was the same in every task: First, participants were instructed to navigate to a particular section within this page. Second, they had to check whether the paragraph identified as the currently focused paragraph by our context detection mechanism was correct. In case of an incorrect identification, they had to change the focused paragraph either by clicking on another extracted paragraph or by selecting the relevant piece of text. The first option applied when the extraction of paragraphs was correct, but the identification of the focused paragraph failed. The second option applied when the extraction failed already. After a potential correction of the focused paragraph, a query was generated automatically and participants had to provide relevance feedback for the results of this query. Then, they were instructed to adapt the query and rate the results of the modified query, until one of the following criteria was met: The participant is fully satisfied with the results, the participant does not believe the search engine being able to deliver any results on the topic or the time is up (around 8 minutes per task).

The evaluation was carried out on a cleaned dataset (e.g., queries that could not doubtlessly be assigned to a task were removed), resulting in 558 queries executed by 69 users in 228 tasks and 6985 relevance ratings for the results. The evaluation results are depicted in Table II.

The reported precision-, recall- and F1-scores are macro averaged over all queries (hence the F1-score is below the value obtained from the average precision- and recall-score). Clearly, we cannot measure the true recall value, as we do not have ground truth relevance feedback for the whole collection. Instead, we approximate the recall with all positive results retrieved via all queries executed in the context of a particular Wikipedia page. For the evaluation of user queries, we took the best query, a user was able to formulate for a paragraph. This means, that if the initial automatic query scored

better than all subsequent modifications by the user, we take the initial query. As can be seen from the table, the performance in general is quite low, with the user generated queries performing slightly better than the automatic queries.

Participants modified 16% of the extracted paragraphs, i.e. 84% of the paragraphs were extracted correctly or meaningful from a user perspective. From the correctly extracted paragraphs, the focused paragraph was identified correctly in 65% of the cases. These results are quite promising, in particular as they are based on a simple heuristic. However, they are only valid for Wikipedia pages, and we cannot draw any conclusions for other web pages.

We consider the chosen main topic as suitable when it is not changed by the user or a change does not lead to an improvement. In the main topic evaluation, we removed all cases, where we cannot make a definitive statement about the reasons, why the main topic was not changed. This comprises task, in which no stopping criterion was provided or the task was stopped due to time limitations. According to this evaluation, the main topic is appropriately chosen in 83% of the automatic queries.

3.4.2.5 *Outlook.* Given the promising results of paragraph extraction and detection on Wikipedia, we plan to evaluate the approach via a crowd-sourcing experiment on a larger variety of web pages and integrate further improvements. Even though the main topic choice is suitable with a high accuracy already, there is still room for improvement. In particular, we discovered, that the suggested main topic is less subject to modifications when it is identical to the topic of the page. Moreover, in 82% of the queries where a main topic modification resulted in an improvement, the originally suggested main topic was different from the page topic. These findings suggest, that the main topic extraction should be based on the whole page instead of the focused paragraph. Regarding the keywords (named entities besides the main topic) of the query generation process, we are researching optimizations to filter the extracted keywords in order to improve the query quality and add additional keywords besides the named entities.

3.5 Privacy Preservation

Sending users' personal information to the federated aggregation component raises serious privacy issues. For that reason, we introduce mechanisms to help users to control the privacy preservation of their data. Firstly, we design a user interface to enable users to define their own privacy policy (i.e., which information they agree to share with the federation component). Secondly, we design a privacy-preserving protocol between the client and the federation component to ensure that users' queries cannot be used to profile any users.

3.5.1 *Privacy Policy.* A user can explicitly specify which information from her user profile she agrees to pass to the system. Indeed, when a query is issued, it is expanded with the context of the user contained in her user profile (e.g., age, language, interests). Consequently, if the user chooses to not disclose specific information, this information is not added to the query. To avoid confusion for non-expert users, we created a simple interface as depicted in Figure 5. For each attribute contained in the user profile (e.g., age, language), a user specifies if she wants to hide or disclose the information to the system. Other attributes enable a more precise configuration (e.g., for location, the user can choose to share: her country, her city, or nothing). By default, all attributes are not disclosed.

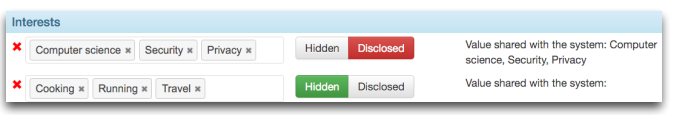

Fig. 5. Cutout of the user interface for defining the privacy settings.

3.5.2 *Query Protection.* The privacy-preserving protocol between users and the federated aggregation component consists in hiding the user identity and masking the user query. A user query contains two pieces of information: her identity (via her IP address) and her interests (via the content of the query). To successfully protect users' privacy, these two data need to be separated. This is the role of the Privacy Proxy. This entity is composed of two servers: the receiver and the issuer. The receiver knows the identity of the requester (with her IP address) without learning her interests, while the issuer accesses queries without knowing their provenance.

The privacy-preserving protocol works as follows: The receiver receives a message from the user, which contains an encrypted request (which can only be decrypted by the issuer). The receiver then forwards this message to the issuer. The issuer deciphers it and forwards the query to the federated aggregation component. Upon receiving a response, the issuer ciphers the results and forwards them to the receiver. Finally, the receiver forwards the message to the user who retrieves the results by deciphering the message.

Nevertheless, hiding the user identity is not enough to correctly protect users (as it has been shown in [Peddinti and Saxena 2014; Petit et al. 2016]). That is why we reinforce the user protection by masking the query with multiple fake queries. A message is then composed of the original query and k fake queries. Generating plausible fake queries is a difficult task. We perform this operation by reusing queries already sent to the system. This is made possible by the issuer which aggregates all queries it receives in a group profile. Queries are aggregated such that publishing the group profile does not leak information about individuals (i.e., publishing real past queries) but contains enough data to estimate real queries. As a consequence, fake queries generated by the Chrome Extension seem realistic as they are semantically coherent and their topics fit with the users in the system. However, sending fake queries has an impact on the accuracy. The federation aggregation component answers results about all queries (and not only the original one). For that reason, we also implement a filtering algorithm to only retrieve results corresponding to the initial request. Finally, the impact on the accuracy is relatively small as for 95% of queries, more than 80% of the expected results are correctly returned. Further details about the privacy-preserving protocol and the obfuscation algorithm are accessible in [Petit et al. 2015].

4. EXAMPLE APPLICATIONS

In this section we present examples how the presented infrastructure and components can be used to distribute cultural content to interested users and how content providers can be connected to the services. Section 4.1 describes the process for integrating cultural objects from small museums in Switzerland. The section describes an internal process and approach to establish multilateral readiness for the technical integration of the data. A fully automatic integration using the PartnerWizard component is described in Section 4.2 for content providers with available search API. Two content-related processes can be differentiated in the Web, namely *content consumption* (e.g., reading online news, researching information) and *content creation* (e.g., authoring web pages or creating blog entries) [Granitzer et al. 2013]. We present two client applications supporting these processes, a Chrome extension for content consumption in Section 4.3, and an add-on for Google Docs for content creation in Section 4.4.

4.1 Groundwork in a Cultural Heritage Institution

The following section describes the content-related work conducted in both a mid-sized and in about 20 small-sized museums on their journey to the world wide web. The state museum of the canton Baselland (AMBL) curates collections in the fields of natural history, ethnology, industrial history, archeology, arts and historical photography. Out of a total of around 2 million objects, records of about 400,000 objects have been migrated into a single system. Furthermore the state museum was leading

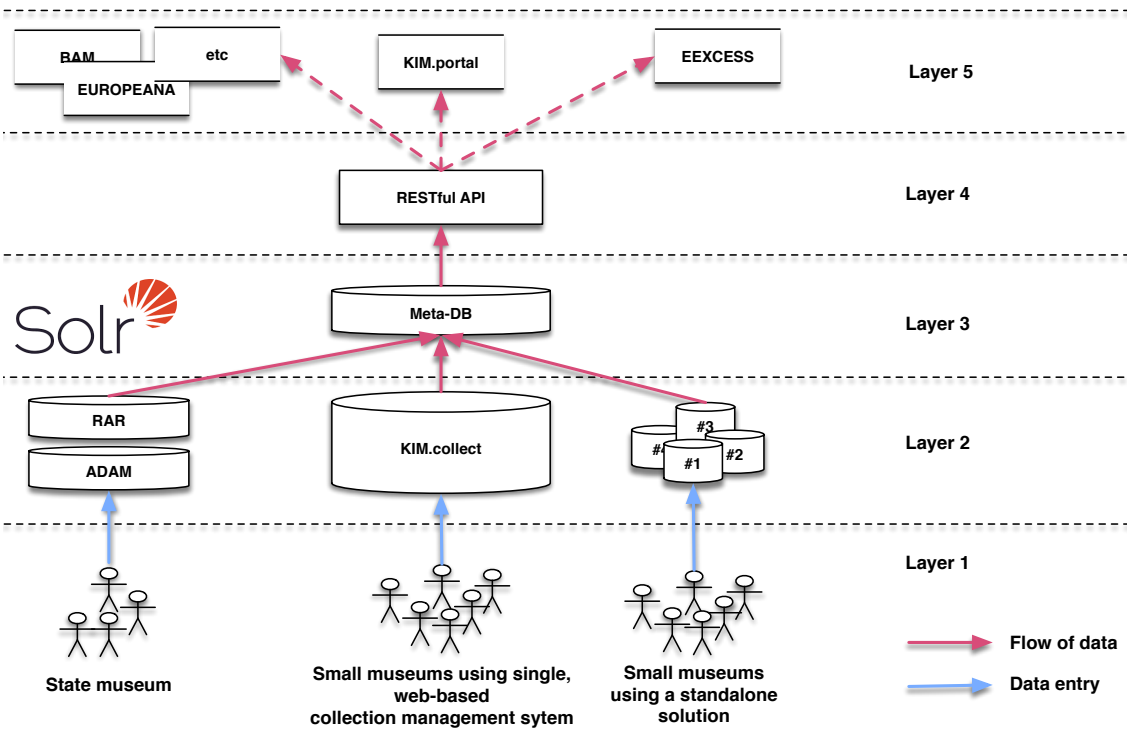

Fig. 6. Overview of the process for integrating the KIM.bl collection into EEXCESS: Curators in the museums manage their collections in their own museum management systems (Layer 1 = Data entry). The systems (Layer 2) provide their data (metadata and media files) using a dedicated export and import routine into the central Metadata Layer (Solr Server; Layer 3). A restful API on top of Layer 3, Layer 4, enables querying by different exploitation channels (Layer 5, e.g., EEXCESS, portal, Europeana) using XSLT routines.

a regional project (KIM.bl)[15] with the goal to establish a network amongst 45 small and large museums to harmonize and centralize their inventories using a central collection management system. The challenge to aggregate and exploit the heterogeneous mass of objects by the provision of one API has been met with the setup of a project that included technical and content-related sub tracks. Figure 6 illustrates both the technical and the content related tracks on high-level.

4.1.1 *Integration Process.* The process of content integration has been performed in four stages: i) target format definition, ii) collection of metadata requirements, iii) collection qualification and mitigation activities, and iv) iterative quality checks. In the following these four stages will be described in more detail. Lessons learned during the process will be outlined after.

Based on discussions with many partner institutions from a network across Europe, the decision had been taken to adhere to the LIDO XML harvesting schema as *target format* becuase this schema is intended for the delivery of metadata for the use in various online services. In addition, the strength of LIDO lies in its ability to support the full range of descriptive information on museum objects across all collection domains like art, architecture, cultural history, history of technology, and natural history.

In a next step, a list of *metadata requirements* was derived from user scenarios and best practices. The scenarios tailored for meeting the target channels in focus: The Museum Portal and EEXCESS.

[15] www.kgportal.ch, last accessed 18.3.2016

The analysis of the requirements translated into a list of necessary metadata fields which in turn were compared with the LIDO schema. All information had been aggregated into a flat ”Master metadata matrix”, which served as our look-up table to track progress during the whole project.

Subsequently, every collection in scope was screened and assessed for compliance with the matrix. Missing pieces were marked and translated into mitigation activities. Individual *quality control* activities per collection were identified and labeled with a timeline. The activities ranged from simple to complex IT-driven data-migration and completion routines. In other cases, profound discussions with subject matter experts were conducted and resulted in labor intensive handwork. Where necessary, external staff had to be hired in order to complete and harmonize the metadata. A separate track included the completion of photographs.

Collections that qualified for testing purposes were migrated to test environment. In joint collaboration with the partner teams from the project the results have been continuously assessed. That way missing pieces and gaps could be approached early in the project.

4.1.2 *Summary and Lessons Learned.* While implementing the outlined process, three main aspects turned out to be a persisting challenge: (i) engagement and awareness by curators, (ii) rights management and licensing, and (iii) technical implementation. In order to complete and harmonize the metadata per collection the engagement of subject matter experts is crucial. The establishment of a continuous dialogue with the experts ensures that they understand the goals of the projects and contribute with valuable inputs. The provision of metadata to a publicly available channel like EEXCESS asks for careful considerations different to the established internal curators work (e.g., from unstructured to structured information; transformation of time and periods; structured geo information; introduction of controlled vocabularies etc.).Workload for screening and complete metadata: These tasks are very labor intensive and often cannot be processed using routines. The involvement of specialized and skilled staff needs to be included into the calculation (time and budget).Going public asks for a broad discussion on rights management and licensing. The digital turn in GLAMs asks for a digital strategy of the institution. More information can be found on the webpage of the open GLAM initiative[16].

Besides these content related tasks it is important to state that a cultural heritage institution needs to have technical experts at its disposal. A museum management system typically does not feature the functionalities necessary to go online. A robust and flexible IT solution must be envisaged, realized and maintained in order to interface with a project like EEXCESS. This issue has been emphasized in a white paper targeted at potential new content providers[17].

4.2 EEXCESS Provider Integration Wizard

The manual process described above ensures the highest quality of data contribution to the EEXCESS framework. However, several steps in this process can be supported by (semi-)automatic tools. This is especially useful in cases where the required technical expertise is not available, which is often the case in smaller institutions. The following three components facilitate the data provider integration process.

The **PartnerWizard** introduced in Section 3.3 enables the basic integration of a data provider, i.e., ensuring that queries are correctly fed into their API, defining mapping of basic metadata fields and optimizing the way queries are constructed. All configuration is done in a web application and does not require detailed technical knowledge or programming skills. The PartnerWizard has been developed based on the manually integrated data providers (as described above), and the result of automatic integration has been validated against those from manual integration. In addition, eight data providers

[16] http://openglam.org/, last accessed 18.3.2016

[17] http://eexcess.eu/content-providers/, last accessed 18.3.2016

have so far been integrated with the PartnerWizard as a starting point. While the PartnerWizard is able to connect the API and provide basic mapping, it cannot leverage the full potential of the provided metadata if mappings require expert knowledge or adjustments to the data provided by the API are necessary.

The mapping created by the PartnerWizard covers the basic fields required by the EEXCESS framework. The APIs of many data providers return additional metadata, often in specific structures, that can improve the findability of assets. The **Mapping Configuration Tool** described in Section 3.1.2 provides the functionality to refine and extend the mapping for a data provider. Using this tool requires more skills than the PartnerWizard, in particular concerning the metadata model of the data provider. The tool can provide previews of transformed metadata records, so that the impact of changes to the mapping can be directly observed. An updated mapping can then be used by the recommender component for the respective data provider.

The **Quality Assessment Tools** described in Section 3.1.3 run as background process in the system, and provide valuable information for the data provider concerning the source metadata quality (i.e., the data returned from the provider's API) and the mapping to the EEXCESS model (i.e., the quality of the mapping definition). The assessment results can be used to improve the quality of the data, and thus the quality of the recommendations for the consumer. In some cases, even simple adjustments may have large impact on the quality (e.g., choice of default values, omitting empty fields, mapping additional fields). Addressing some recommendations may require changes to the data provider's search API, but has the potential to not only improve the quality of the data provided to EEXCESS, but also to other consumers of the API.

4.3 Client Application for Content Consumption

In the previous two sections we discussed possibilities for integrating new data sources into the presented infrastructure, either manually or automatically. In this and the next section we present example applications for distributing content to interested users, for consumption purposes (e.g., using related cultural resources as information consumption) and for content creation purposes (e.g., integrating cultural resources into newly created content).

For consuming cultural content related to the current (textual) user context, we developed an extension for the Chrome browser. The *Chrome extension* allows to access cultural content, which is relevant to the current Web page. While browsing the Web, reading news articles or blog entries, additional resources available in the content provider's data bases are automatically retrieved and suggested for further investigation.

When installed and activated[18], the extension is visible as a light gray bar at the bottom of every web page as shown in Figure 7. The paragraph automatically detected as active is outlined in green within the web page. The extracted keywords are visible at the bottom of the page, together with the main topic of the paragraph. Additional search tools, e.g., search for persons only, are available (bottom left). The main topic and the keywords are automatically sent as query to the federated aggregtion component, and the number of retrieved results is indicated on the bottom right. Keywords can be manually added, modified or deleted. A keyword can become the main topic by simply dragging it on the main topic field. When the result indicator is clicked, the search result list becomes visible. The search result view can be moved and changed in size. Search results can be sorted and filtered according to different metadata values (e.g., the media type of the result). Additionally, the search history is visualized (bottom right), and can be used to re-visit previous searches.

[18]https://purl.org/eexcess/clients/chrome-extension, last accessed 18.3.2016

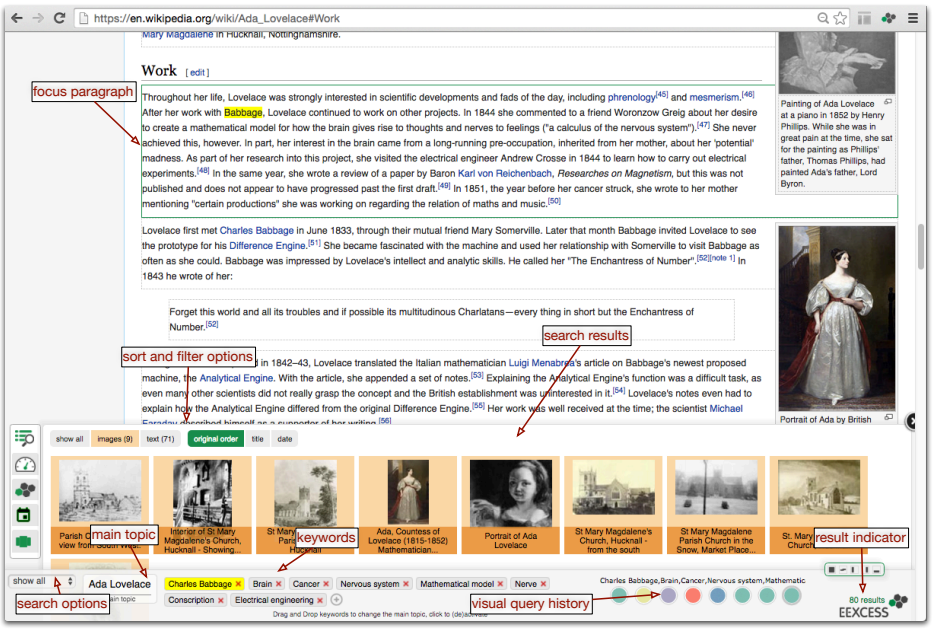

Fig. 7. Screenshot of the Chrome extension. When hovering over a keyword at the bottom the respective term in the focus paragraph is highlighted indicating the keyword context (yellow highlights).

To collect feedback about the general usability from a broader audience we generated a survey using the SUS scale [Brooke 1996].Answers are collected on a 5-point Likert scale (1 - strongly agree, 5 - strongly disagree). An open question was added to the standard questionnaire asking for comments and possible improvements. The survey was prepared in English and German and sent to project-internal mailing lists with a potential reach of 1231 people. During the time period of 24.11.2015 to 31.12.2015 we received 26 responses. The overall SUS score was 65.6, which is slightly below the average score of 68. The received qualitative feedback can be summarized as follows: The relevance of the results should be improved and certain parts (i.e., some visualizations included in the extension, but not described in this paper) of the user interface are too complex. Further, users were disappointed because they received only responses from specialized providers. Note, that this evaluation only considers usability issues of the current system with a fixed number of providers. We have not yet tested how user satisfaction changes with the number of providers.

4.4 Client Application for Content Creation

In the content creation scenario, additional cultural resources from the content providers are retrieved while users are creating their own (textual) content. To showcase this scenario, we implemented an add-on for the online word processor Google Docs[19]. The core idea of the *Google Docs add-on* is to have a collaborative way of writing documents, and – while writing – a simple way to include related images and citations. Real-time online collaboration is available in Google Docs per se, the EEXCESS add-on extends Google Docs with cultural resources, that can be inserted with one click.

[19]https://docs.google.com, last accessed 18.3.2016

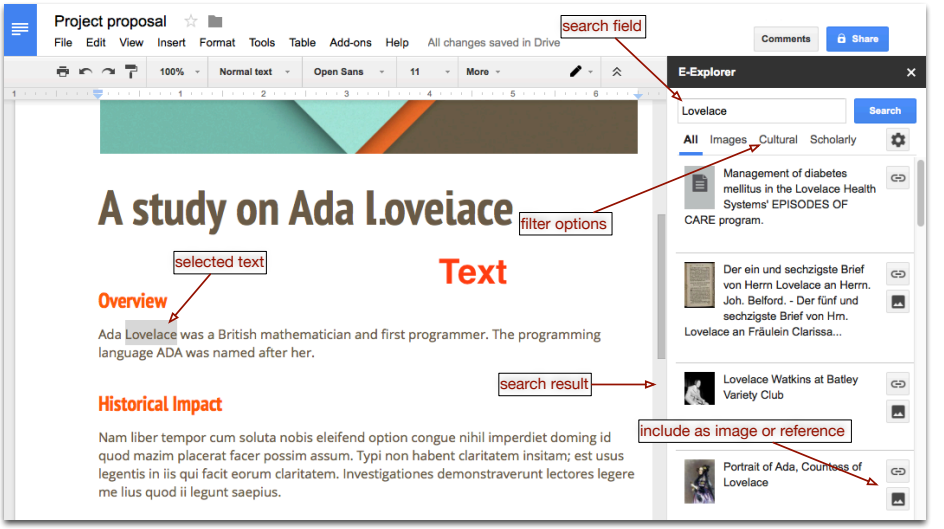

Fig. 8. Screenshot of the Google Docs plugin

The EEXCESS Google Docs add-on can be downloaded from the App-Store [20] and needs to be started for each document separately from the menu *Add-ons* → *E-Explorer* → *Start*. When activated, the add-on is visible on the right side of the currently edited document (see Figure 8). To get results, a phrase in the document can be highlighted (either by double clicking or keyboard selection) and is automatically added to the search field. The add-on searches all connected content providers via the federation component and displays the results as a list on the sidebar. The details of the results can be accessed by clicking on the list entry. This brings you to a page with more detailed information about the digital object. Resources can be included in the document either as a reference or an image (two icons on the right side of a result in Figure 8). The add-on has been evaluated with 25 students of a seminar at the Munich School of Philosophy using a questionnaire. The questionnaire asked about general usability, quality and variety of results and the perceived level of privacy. The students reported an overall good usability and overall relevance (e.g., ”[..] gives the user a bigger freedom of choice due to its ability to connect various articles regarding a certain topic.”). With respect to privacy students reported differing viewpoints: ”Compared to regular search services via sites like Google, I would say that EEXCESS is a possibility to find information from certain sources you trust, insofar as you can choose which providers you want to see. ” and ”Where is my personal data stored? How can I access it? Can I access it?”.

In summary, the Google Docs add-on is a usable tool providing access to cultural resources when collaboratively writing documents. Although the overall infrastructure ensures users’ privacy, it is not transparent what other stakeholders (in this case Google) can infer from user of the add-on.

5. RELATED WORK AND DISCUSSION

In this section we discuss works, projects and initiatives related to the presented infrastructure and single components w.r.t. (i) federated search and recommendation, (ii) metadata mapping, (iii) metadata quality assessment and (iv) privacy-preservation and (v) just-in-time-retrieval.

[20] https://purl.org/eexcess/clients/googledocs-plugin, last accessed 18.3.2016

A closely related initiative to the presented infrastructure is Europeana[21], the largest aggregator for cultural content in Europe providing access to approximately 44 million objects (beginning of 2016). While Europeana focuses on content aggregation, EEXCESS also provides content dissemination components, and relates cultural content with scientific content by integrating scientific data repositories as data providers. EEXCESS includes Europeana as a data source, and provides access to approximately 130 million objects, including scientific content.

5.1 Federated Search and Recommender Systems

Combining multiple search results into a single, consolidated search result is often referred to as meta search. Typically, the focus of such techniques has been on Web search with the goal to improve the quality and reduce unwanted content [Dwork et al. 2001], where results multiple Web search engines are combined. In such a scenario the results are expected to be homogeneous, sharing the same characteristics and metadata. If more heterogeneous results are to be combined, the term vertical search is commonly used, particularly for systems that display multiple search results side-by-side. For example, image and text search are both triggered to react on a specific information need. The term aggregated search is used to indicate that multiple search results are combined into a single one [Murdock and Lalmas 2008; Kopliku et al. 2014].

According to the literature [Shokouhi and Si 2011; Lu and Callan 2005], the three main challenges that aggregated search faces are: i) the selection of the appropriate sourcesii) the so called collection representation problem, i.e. inferring the key characteristics of a knowledge base while keeping the effort minimal iii) the aggregation of the results returned by the different sources.

An important aspect in an uncooperative, aggregated setting (where the underlying search engines can not be modified) is the query processing aspect. Here problems may arise due to short queries, which have to be expanded with related terms for better results [Montgomery et al. 2004]. At the opposite end of the spectrum are queries consisting of many terms with distinct concepts, which need to be split into segments, which are themselves coherent [Hagen et al. 2012].

There is a strong connection between recommender systems [Ricci et al. 2011] and information retrieval. One type of recommender systems are the so called content-based recommender systems [Lops et al. 2011]. The key aspect is here that the recommendations, i.e. search results, are provided without the users having to explicitly state their information need.

5.2 Metadata Mapping and Quality Assessment

There are several initiatives in the cultural heritage domain to provide services for mapping between library and museum metadata formats (e.g., the OCLC crosswalk service[22]). MINT[23] is a recent framework facilitating aggregation of cultural heritage information from heterogeneous sources and perform metadata mapping. This framework has been used in the Athena, Carare and EUscreen projects (all completed) as well as in recent projects such as LoCloud, AthenaPlus, and EUscreenXL to prepare cultural heritage metadata for ingestion into Europeana. The PrestoPRIME project[24] has developed a metadata mapping service for audiovisual metadata [Höffernig et al. 2010], including support for the Europeana Data Model (EDM). Most of these approaches are designed for the interactive definition of metadata mappings, which is the typical scenario when ingesting data into a cultural heritage portal. Such a workflow is not feasible in EEXCESS, as metadata transformation needs to be done on the fly.

[21] http://europeana.eu, last accessed 18.3.2016
[22] http://xwalkdemo.oclc.org/, last accessed 20.3.2016
[23] http://mint.image.ece.ntua.gr/redmine/projects/mint/wiki/Wiki, last accessed 20.3.2016
[24] http://www.prestoprime.org/, last accessed 20.3.2016

Most of the existing literature on metadata quality considers the metadata of single or multiple records of a collection, i.e., our source metadata. However, as EEXCESS access metadata from a range of different sources, there is no single application profile that can be checked against. The authors [Bruce and Hillmann 2004] define the following measures for quality: completeness, accuracy, provenance, conformance, logical consistency and coherence, timeliness and accessibility. A taxonomy of 22 measures for information quality has been proposed in [Stvilia et al. 2007], grouped into three categories: intrinsic, relational/contextual and reputational information quality. An approach that attempts automation has been proposed in [Bellini and Nesi 2013]. The authors start from viewing metadata quality as the fitness for use for a specific purpose and propose three metrics: completeness, accuracy and consistency. [Debattista et al. 2014] propose an extensible framework for assessing quality of Linked Open Data called Luzzu. One important contribution of their work is a data quality ontology (daQ), which is also used an input to W3Cs recent work on this topic. [Trippel et al. 2014] (and similarly [Reiche et al. 2014]) propose a quality assessment framework using similar criteria as earlier works, but they calculate a single score over all these criteria. The work of [Gavrilis et al. 2015] proposes an assessment framework called MQEM, but include a set of concrete metrics which yield numeric values. Another aspect of metadata quality is the use of controlled vocabularies, e.g., using the eight classes of criteria proposed in [Dröge 2012]).

5.3 Privacy-Preservation for Search

The main solutions to query a content provider or a recommender system in a privacy-preserving way can be classified in two categories: (i) protocols to ensure *unlinkability* between requesters and their queries, and (ii) systems guaranteeing *indistinguishability* of user interests. Unlinkability solutions hide user's identity to prevent the distant server to identify and analyze queries issued by a single user. Basic techniques consist in sending queries through a Proxy [Shapiro 1986] or a VPN server [Seid and Lespagnol 1998]. Unfortunately, these mechanisms only shift the privacy problem from the search engine to the relay (which is able to extract user's interests). To solve this issue, anonymous protocols (e.g., [Goldschlag et al. 1999; Dingledine et al. 2004; Ben Mokhtar et al. 2013]) ensure, with cryptographic primitives, that all third parties in the system cannot access both the user identity and the content of her query. However, most of these solutions rely on heavy cryptographic protocol or all-to-all communication that make them impractical for querying a recommender system. In [Castellà-Roca et al. 2009; Lindell and Waisbard 2010], authors propose a fully decentralized architecture in which users exchanged their queries and sent them on behalf of each other. But similarly to anonymous protocols, this solution produces significant overhead (more network traffic and higher latency).

Indistinguishability solutions consist in making all analysis of user queries inaccurate. For instance, TrackMeNot [Toubiana et al. 2011] periodically sends fake queries to obfuscate the user profile created by the distant server. In [Murugesan and Clifton 2009; Domingo-Ferrer et al. 2009], the user query is directly obfuscted by sending the initial query with k extra fake queries. As a result, the distant servers cannot distinguish which query (among the $k + 1$) is the correct one. Furthermore, Query Scrambler (presented in [Arampatzis et al. 2013]) protects users by generating and issuing queries related to the initial query. As the initial query is not sent, Query Scrambler reduces the leak of information but the accuracy of the recommendation is highly impacted. A recent approach, Dispa [Juarez and Torra 2015], reconciles personalization with user protection. It consists in making the distant server constructing multiple user profiles for a single user. Each user profile contains partial but accurate information about the user. As the distant server does not know that all small user profiles are related to the same user, this method reduces the amount of disclosed of information. However, this solution relies on the strong assumption that the distant server only uses cookies to identify users (and not other elements such as the IP address or the HTTP header).

Several privacy attacks [Peddinti and Saxena 2014; Gervais et al. 2014; Petit et al. 2016] against unlinkability and indistinguishability solutions have been published in the literature. However, most of es privacy solutions are not robust against these attacks, and therefore we proposed a new approach to unlinkability and indistinguishability.

5.4 Just-in-time-Retrieval

Proactive retrieval of resources relevant to the current user context and unobtrusive presentation of the retrieved results was first made popular by Rhodes as Just-in-Time Retrieval [Rhodes 2000]. Recently, Rhodes' research has been continued under the topic of zero effort queries [Allan et al. 2012] with special emphasis on mobile applications [Lee and Sumiya 2009]. Zero effort queries require minimal, ideally no, effort from the user in expressing her information need. While earlier work [Rhodes and Maes 2000; Lieberman 1997; Budzik and Hammond 1999] focused on document retrieval, a wide variety of contents is taken into account in more recent work [Shokouhi and Guo 2015]. However, these systems either treat the retrieval system as an integral part of the application or focus on domain-specific sets of information needs. EEXCESS is agnostic of the underlying retrieval system and can be applied to any search system whose contents are searchable via a REST-API.

6. SUMMARY

In this paper we presented an infrastructure for aggregating and distributing cultural content into various channels. Conceptually, the infrastructure realizes a personalized, privacy-preserving just-in-time retrieval of cultural content supported by federated aggregation of different sources and metadata harmonization. The single components communicate with standard Web technologies and well-defined APIs, and thus can be used as stand-alone components with only minor modifications. The EEXCESS infrastructure and all the components are available as Open Source[25]. The developed components comprise (i) metadata harmonization, (ii) partner recommenders and a federated aggregation component, (iii) privacy-preservation, (iv) user interface clients including presentation widgets and context detection, and (v) a component for automatic inclusion of new data providers.

The *metadata harmonization* component includes tools for creating, maintaining and applying metadata mappings, as well as performing metadata quality assessment. These components are not restricted to data provision to EEXCESS use cases, but can also efficiently support data providers in related tasks, e.g., data provision to Europeana or other cultural heritage portals. Together with the *PartnerWizard* they facilitate data provision from GLAM institutions.

The *federated aggregation* component is responsible for retrieving a single result list given a user's query as input querying partner sources in the process. Necessary steps within this component in order to create highly relevant and diverse result lists are the selection of appropriate sources (given the query), the query reformulation to adapt queries to single data providers and result list aggregation. Although these processes had been expected to be computationally intensive, they have been found to be quite efficient in practice.

The *privacy preservation component* tackles potential privacy issues that occur when a query is sent to the federated aggregation component. Mechanisms for ensuring users' privacy are a user interface to enable users to define their own privacy policy and a privacy-preserving protocol between the client and the federated aggregation component to ensure that users' queries cannot be used for user profiling.

The *user interface client component* comprises a module-based client architecture comprising of user interface widgets and a context-detection library. All client components can be easily integrated into client applications for presenting cultural content given the current web-based user context.

[25] github.com/eexcess, last accessed 20.3.2016

The integration of client components has been exemplified in two applications, the *Google Docs add-on* for supporting content creation processes and the *Chrome extension* for supporting content consumption processes. The Google Docs add-on allows to include related images and citations while collaboratively writing a document. The Chrome extension can be used to discover related cultural content while browsing the Internet. Evaluation results indicate an already overall satisfying usability. However, the following areas for improvement have been identified in a user study: (i) increase the result relevance by improving query generation and result ranking, (ii) user friendly guideline on scope of usage of the Chrome extension, (iii) improve the integration of the user interface components, and decrease the complexity for first-time users.

The result relevance and ranking has been targeted since the presented user evaluation. Further, a screencast has been prepared and is distributed along with the extension to communicate the purpose of the module. For complex user interface parts (e.g., the visualizations, which have not been described here), a tutorial has been implemented to guide users through the functions of the respective module.

The client modules have also successfully been implemented in other scenarios, e.g., as a Wordpress plugin [Seifert et al. 2015] and as a plugin for the Moodle e-Learning platform. Another example of adaption of components from the EEXCESS framework is the integration of a search visualization client in the *imdas pro*[26] collection management software.

REFERENCES

James Allan, Bruce Croft, Alistair Moffat, and Mark Sanderson. 2012. Frontiers, Challenges, and Opportunities for Information Retrieval: Report from SWIRL 2012. *SIGIR Forum* 46, 1 (5 2012), 2–32.

Avi Arampatzis, Pavlos S Efraimidis, and George Drosatos. 2013. A query scrambler for search privacy on the internet. *Information retrieval* 16, 6 (2013), 657–679.

Albert-Lásló Barabási, Réka Albert, and Hawoong Jeong. 2000. Scale-free characteristics of random networks: the topology of the world-wide web. *Physica A: Statistical Mechanics and its Applications* 281, 1–4 (2000), 69 – 77.

Emanuele Bellini and Paolo Nesi. 2013. Metadata Quality Assessment Tool for Open Access Cultural Heritage Institutional Repositories. In *Information Technologies for Performing Arts, Media Access, and Entertainment*. 90–103.

Sonia Ben Mokhtar, Gautier Berthou, Amadou Diarra, Vivien Quéma, and Ali Shoker. 2013. RAC: A Freerider-Resilient, Scalable, Anonymous Communication Protocol. In *Proc. IEEE International Conference on Distributed Computing Systems (ICDCS)*. 520–529.

John Brooke. 1996. SUS: A quick and dirty usability scale. In *Usability evaluation in industry*, P. W. Jordan, B. Weerdmeester, A. Thomas, and I. L. Mclelland (Eds.). Taylor and Francis, London.

Thomas R. Bruce and Diane I. Hillmann. 2004. *The Continuum of Metadata Quality: Defining, Expressing, Exploiting*. ALA Editions, Chicago, 238–256.

Jay Budzik and Kristian Hammond. 1999. Watson: Anticipating and Contextualizing Information Needs. In *Annual meeting of the american society for information science*. 727–740.

Jordi Castellà-Roca, Alexandre Viejo, and Jordi Herrera-Joancomartí. 2009. Preserving user's privacy in web search engines. *Computer Communications* 32, 13 (2009), 1541–1551.

J. Debattista, S. Londoo, C. Lange, and Sören Auer. 2014. LUZZU – A Framework for Linked Data Quality Assessment. http://arxiv.org/abs/1412.3750, *CoRR, abs/1412.3750* (2014).

Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-generation Onion Router. In *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13 (SSYM'04)*. USENIX Association, 21–21.

Josep Domingo-Ferrer, Agusti Solanas, and Jordi Castellà-Roca. 2009. h(k)-private information retrieval from privacy-uncooperative queryable databases. *Online Information Review* 33, 4 (2009), 720–744.

Evelyn Dröge. 2012. *Criteria for vocabulary evaluation and comparison*. Technical Report. Humboldt-Universität zu Berlin.

Cynthia Dwork, Ravi Kumar, Moni Naor, and D Sivakumar. 2001. Rank aggregation methods for the Web. *Proceedings of the 10th international conference on World Wide Web* (2001), 613–622. DOI:http://dx.doi.org/10.1145/371920.372165

Europeana Foundation. 2015. *Definition of the Europeana Data Model*. Technical Report. Europeana Foundation. http://pro.europeana.eu/page/edm-documentation

[26]http://culture.joanneum.at/

D. Gavrilis, D.-N. Makri, L. Papachristopoulos, S. Angelis, K. Kravvaritis, C. Papatheodorou, and P. Constantopoulos. 2015. Measuring Quality in Metadata Repositories. In *Research and Advanced Technology for Digital Libraries,*.

Arthur Gervais, Reza Shokri, Adish Singla, Srdjan Capkun, and Vincent Lenders. 2014. Quantifying web-search privacy. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 966–977.

David Goldschlag, Michael Reed, and Paul Syverson. 1999. Onion routing. *Commun. ACM* 42, 2 (1999), 39–41.

Michael Granitzer and Christin Seifert. 2016. Taking Cultural and Scientific Content to Users through the EEXCESS Project. *D-Lib Magazine* 22, 3/4 (3 2016). DOI:http://dx.doi.org/10.1045/march2016-content

Michael Granitzer, Christin Seifert, Silvia Russegger, and Klaus Tochtermann. 2013. Unfolding Cultural, Educational and Scientific Long-Tail Content in the Web. In *UMAP Extended Proceedings*, Shlomo Berkovsky, Eelco Herder, Pasquale Lops, and Olga C. Santos (Eds.), Vol. 997. http://ceur-ws.org/Vol-997/umap2013_project_1.pdf

Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. 2009. Named Entity Recognition in Query. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09)*. ACM, New York, NY, USA, 267–274. DOI:http://dx.doi.org/10.1145/1571941.1571989

Matthias Hagen, Martin Potthast, Anna Beyer, and Benno Stein. 2012. Towards Optimum Query Segmentation: In Doubt Without. In *Proc. Intern. Conf. on Information and Knowledge Management (CIKM)*. ACM, New York, NY, USA, 1015–1024.

David Hauger, Alexandros Paramythis, and Stephan Weibelzahl. 2011. Using Browser Interaction Data to Determine Page Reading Behavior. In *UMAP'11*. Springer-Verlag, 147–158. http://dl.acm.org/citation.cfm?id=2021855.2021869

Martin Höffernig, Werner Bailer, Günter Nagler, and Helmut Mülner. 2010. Mapping Audiovisual Metadata Formats Using Formal Semantics. In *SAMT*. 80–94.

Martin Höffernig, Thomas Orgel, Silvia Russegger, and Werner Bailer. 2015. Assessing Quality in Automated Metadata Aggregation and Mapping Services. In *Workshop on Cloud-based Services for Digital Libraries*.

ISO 21127. 2014. Information and documentation – A reference ontology for the interchange of cultural heritage information. (2014).

Marc Juarez and Vicenc Torra. 2015. DisPA: An Intelligent Agent for Private Web Search. In *Advanced Research in Data Privacy*. Vol. 567. Springer International Publishing, 389–405.

Arlind Kopliku, Karen Pinel-Sauvagnat, and Mohand Boughanem. 2014. Aggregated search: A new information retrieval paradigm. *ACM Computing Surveys (CSUR)* 46, 3 (2014), 41.

Quoc V. Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proc. of the 31th Int. Conf. on Machine Learning (ICML)*. Bejing, China, 1188–1196.

Timothy Lebo, Satya Sahoo, and Deborah McGuinness (eds.). 2013. PROV-O: The PROV Ontology. http://www.w3.org/TR/prov-o/. (2013).

Ryong Lee and Kazutoshi Sumiya. 2009. Zero-Effort Search and Integration Model for Augmented Web Applications. In *Proc. of the 9th Int. Conf. on Web Engineering (ICWE '9)*. Springer-Verlag, 330–339.

Henry Lieberman. 1997. Autonomous Interface Agents. In *Proc. of the ACM SIGCHI Conf. on Human Factors in Compu. Sys. (CHI '97)*. ACM, 67–74.

Yehuda Lindell and Erez Waisbard. 2010. Private web search with malicious adversaries. In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies (PETS'10)*. Springer-Verlag, 220–235.

Pasquale Lops, Marco De Gemmis, and Giovanni Semeraro. 2011. Content-based recommender systems: State of the art and trends. In *Recommender systems handbook*. Springer, 73–105.

Jie Lu and Jamie Callan. 2005. Federated search of text-based digital libraries in hierarchical peer-to-peer networks. In *Advances in Information Retrieval*. Springer, 52–66.

Kay Michal. 2007. XSL Transformations (XSLT) Version 2.0. W3C Recommendation. (2007). http://www.w3.org/TR/2007/REC-xslt20-20070123/

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Jesse Montgomery, Luo Si, Jamie Callan, and David A Evans. 2004. Effect of varying number of documents in blind feedback: analysis of the 2003 nrrc ria workshop bf_numdocs experiment suite. In *Proc. Intern. ACM SIGIR conference on Research and development in information retrieval (SIGIR)*. ACM, 476–477.

Vanessa Murdock and Mounia Lalmas. 2008. Workshop on aggregated search. *ACM SIGIR Forum* 42, 2 (2008), 80.

Mummoorthy Murugesan and Chris Clifton. 2009. Providing Privacy through Plausibly Deniable Search. *Proceedings of the 2009 SIAM International Conference on Data Mining* (2009), 768–779.

Thomas Orgel, Werner Bailer, Martin Höffernig, Werner Preininger, and Silvia Russegger. 2016. *Integration and Enrichment Services Final Prototype*. Technical Report EEXCESS Deliverable 4.4.

Thomas Orgel, Martin Höffernig, Werner Bailer, and Silvia Russegger. 2015. A Metadata Model and Mapping Approach for Facilitating Access to Heterogeneous Cultural Heritage Assets. *Int. J. Digit. Libr.* 15, 2-4 (April 2015), 189–207.

Sai Teja Peddinti and Nitesh Saxena. 2014. Web Search Query Privacy: Evaluating Query Obfuscation and Anonymizing Networks. *J. Comput. Secur.* 22, 1 (Jan. 2014), 155–199.

Albin Petit, Thomas Cerqueus, Antoine Boutet, Sonia Ben Mokhtar, David Coquil, Lionel Brunie, and Harald Kosch. 2016. *SimAttack: Private Web Search Under Fire.* Technical report. Institut National des Sciences Appliquées de Lyon ; Universität Passau. https://hal.inria.fr/hal-01289861

Albin Petit, Thomas Cerqueus, Sonia Ben Mokhtar, Lionel Brunie, and Harald Kosch. 2015. PEAS: Private, Efficient and Accurate Web Search. In *Trustcom/BigDataSE/ISPA, 2015 IEEE*, Vol. 1. IEEE, 571–580.

K. J. Reiche, I. Schieferdecker, and E. Höfig. 2014. Assessment and Visualization of Metadata Quality for Open Government Data. In *Proc. International Conference for E-Democracy and Open Government*.

Bradley James Rhodes. 2000. *Just-In-Time Information Retrieval*. Ph.D. Dissertation. Massachusetts Institute of Technology.

B. J. Rhodes and P. Maes. 2000. Just-in-time Information Retrieval Agents. *IBM Syst. J.* 39, 3-4 (July 2000), 685–704.

Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. *Introduction to recommender systems handbook*. Springer.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. *Automatic Keyword Extraction from Individual Documents*. John Wiley & Sons, Ltd, 1–20. DOI:http://dx.doi.org/10.1002/9780470689646.ch1

Raoul Rubien, Hermann Ziak, and Roman Kern. 2015. Efficient Search Result Diversification via Query Expansion Using Knowledge Bases. In *Proceedings of 12th International Workshop on Text-based Information Retrieval (TIR)*.

Jörg Schlötterer. 2015. From Context to Query. In *Proc. ACM Symposium on Applied Computing (SAC '15)*. ACM, New York, NY, USA, 1108–1109.

Jörg Schlötterer, Christin Seifert, and Michael Granitzer. 2016. Supporting Web Surfers in Finding Related Material in Digital Library Repositories. In *Proc. International Conference on Theory and Practice of Digital Libraries (TPDL)*.

H.A. Seid and A.L. Lespagnol. 1998. Virtual private network. (1998). US Patent 5,768,271.

C. Seifert, J. Jurgovsky, and M. Granitzer. 2014. FacetScape: A Visualization for Exploring the Search Space. In *Information Visualisation (IV), 2014 18th International Conference on*. 94–101.

Christin Seifert, Nils Witt, Sebastian Bayerl, and Michael Granitzer. 2015. Digital Library Content in the Social Web: Resource Usage and Content Injection. In *IEEE STCN Newsletter*, Vol. 3. https://sites.google.com/a/ieee.net/stc-social-networking/e-letter/stcsn-e-letter-vol-3-no-1/

Marc Shapiro. 1986. Structure and Encapsulation in Distributed Systems: the Proxy Principle. In *Proceedings of the 2013 IEEE 6th International Conference on Distributed Computing Systems (ICDCS'86)*. 198–204.

Milad Shokouhi and Qi Guo. 2015. From Queries to Cards: Re-ranking Proactive Card Recommendations Based on Reactive Search History. In *Proc. of the 38th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '15)*. ACM, 695–704.

Milad Shokouhi and Luo Si. 2011. Federated search. *Foundations and Trends in Information Retrieval* 5, 1 (2011), 1–102.

B. Stvilia, L. Gasser, and M. Twidale. 2007. A framework for information quality assessment. *J. American Society for Information Science & Technology* 58, 12 (2007).

Vincent Toubiana, Lakshminarayanan Subramanian, and Helen Nissenbaum. 2011. Trackmenot: Enhancing the privacy of web search. *arXiv preprint arXiv:1109.4677* (2011).

T. Trippel, D. Broeder, M. Durco, and O. Ohren. 2014. Towards Automatic Quality Assessment of Component Metadata. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation*.

Gerwald Tschinkel, Cecialia di Sciascio, Belgin Mutlu, and Vedran Sabol. 2015. The Recommendation Dashboard: A System to Visualise and Organise Recommendations. In *Proc. International Conference on Information Visualisation (IV)*. 241–244.

Hermann Ziak and Roman Kern. 2015. Evaluation of Pseudo Relevance Feedback Techniques for Cross Vertical Aggregated Search. In *Lecture Notes in Computer Science*. Springer, 91–102.

Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. 2016a. DoSeR – A Knowledge-Base-Agnostic Framework for Disambiguating Entities Using Semantic Embeddings. In *Proc. European Semantic Web Conference (ESWC)*.

Stefan Zwicklbauer, Christin Seifert, and Michael Granitzer. 2016b. Robust and Collective Entity Disambiguation through Semantic Embeddings. In *Proc. International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*.