Towards Cloud-Based Knowledge Capturing Based on Natural Language Processing

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2015) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

HOLACONF - Cloud Forward: From Distributed to Complete Computing

Towards Cloud-Based Knowledge Capturing Based on Natural Language Processing

Christian Nawroth,[a] Matthäus Schmedding,[a] Michael Fuchs,[b] Holger Brocks,[b] Michael Kaufmann,[c] Matthias Hemmje[a*]

[a]*University of Hagen, Faculty for Mathematics and Computer Science, Hagen, Germany*
[b]*Research Institute for Communiction and Cooperation, Dortmund, Germany*
[c] *Lucerne University of Applied Sciences and Arts, School of Engineering and Architecture, Horw, Switzerland*

Abstract

The organized capturing and sharing of knowledge is very important, and a lot of tools, such as wikis, social communities and knowledge-management or e-learning portals, exist for supporting this purpose. The community content- and knowledge-capturing, management and sharing portal of the European project "Realising an Applied Gaming Eco-system" (RAGE)[†] combines such tools. The goal of the RAGE project is to boost the collaborative knowledge asset management for software development in European applied gaming (AG) research and development (R&D). To support this process, the so-called RAGE ecosystem implements a portal to support the related asset, content and knowledge exchange between diverse actors in AG communities. Therefore, the community portal in RAGE is designed as a so-called ecosystem and is intended to provide its users different tools for the capturing, management, and sharing of knowledge. In this study, we rely on the term and model definition of spiraling knowledge exchange between explicit and tacit knowledge given by Nonaka and Takeuchi.[1] To achieve the goal of extracting, i.e., externalizing and explicitly representing and sharing this knowledge to its users, we propose to generate a taxonomy for faceted search automatically by extracting named entities form the knowledge sources and to classify documents using Support Vector Machines (SVM). In this paper we present our architectural approach for the NLP-based IR concepts and discuss how cloud services based on data distribution and cloud computing can improve the outcome of our system.

Peer-review under responsibility of Institute of Communication and Computer Systems.

Keywords: Storage Cloud; Scientific Cloud; Natural Language Processing; Named Entity Recognition; Support Vector Machines; Knowledge Management

* Corresponding author.
E-mail address: Matthias.Hemmje@fernuni-hagen.de
† www.rageproject.eu

Peer-review under responsibility of Institute of Communication and Computer Systems.

1. Introduction and Motivation

The EU-based industry for applied gaming is an emerging business field. However, it is still fragmented and needs to build critical mass for global competition. The actors in the game development industry, namely the developers, resellers, users, researchers, etc., are not working together optimally because they do not yet have an integrated platform for knowledge interchange. The European Commission (EC)-funded RAGE project will help overcome this fragmentation and aims to exchange knowledge through an online portal and social space that will connect research, gaming industries, intermediaries, education providers, policy makers and end users.

The goal of this online portal, which is called the RAGE ecosystem, is to allow its participants

- to get hold of advanced, usable gaming assets (technology push);
- to get access to associated business cases (commercial opportunity);
- to create bonds with peers, suppliers and customers (alliance formation);
- to advocate their expertise and demands (publicity);
- to develop and publish their own assets (trade) and
- to contribute to creating a joint agenda and roadmap (harmonization and focus).

Seen as a whole, RAGE is a technology- and know-how-driven research and innovation project. Its main purpose is to be able to equip industry players (e.g., game developers) with a set of technology resources (so-called assets, i.e., software and related knowledge resources) and strategies (i.e., knowledge management support features) to strengthen their capacities to penetrate a market (non-leisure) that is new for most of them, and to consolidate a competitive position in it.

Knowledge-interchange portals mean a lot of human effort to accumulate useable information. Automated knowledge extraction can improve the efficiency of the portal because it generates useful content automatically. Therefore, RAGE will support knowledge extraction from different knowledge sources such as a social network, the software repository, the media archive and the learning management system. In that context, we have identified the following general research questions for which possible answers are described in sections 3 and 4, respectively:

(1) How can AG knowledge be captured automatically, and how can this knowledge be shared optimally?

(2) How can cloud technology support knowledge capturing in the RAGE ecosystem?

Usually, knowledge management is performed by the usage of assistance tools, e.g., wikis, blogs, e-learning portals, etc. The RAGE ecosystem's initial design already consists of such tools, and we want to go a step further. A lot of implicit and tacit knowledge, as introduced by Nonaka and Takeuchi,[1] is produced by the communities interacting with and within such a portal. For instance, with contents that are commonly used together, e-learning module users have to pass to reduce colleagues' knowledge gaps, etc. Considering this description, RAGE is a system that contains multiple kinds of content and functionality. The storage of heterogeneous contents and knowledge management in RAGE require an especially high input of resources due to the complexity of the related tasks. For this, a cloud-based approach relieves the user from hosting and setting up the complex RAGE (NLP-IR) environment and transfers this task to a specialized cloud provider. This is our main motivation for investigating the need and the advantages for a cloud-based NLP-IR architecture in RAGE.

The main components of the RAGE ecosystem considered as knowledge sources are:

- A social network system-integration component will enable the RAGE ecosystem users to collaborate with each other in social network systems and across those and the RAGE ecosystem, which is developed and described by Salman et al. in two papers.[2,3] Thus, the users can directly exchange their knowledge, e.g., in chats or other forms of interaction, but they can also use the RAGE ecosystem for content and knowledge management and sharing.
- The software repository of RAGE will contain so-called assets that represent, e.g., advanced technology components for the support of game development.
- The media archive contains multimedia content, such as video instructions, and the digital library includes documents for example, scientific research papers, n digital form.
- The learning management system enables ecosystem users to author, access, and consume online courses to build up knowledge for the usage of specific assets or reduce knowledge gaps among project participants.

In our cloud-based architecture, these four knowledge sources are the blueprint for the structure of the RAGE storage cloud, which is an abstract concept of data distribution, which we introduce in section 4.4. Before we describe the cloud architecture, we will take a short look at our hybrid approach, which uses methodologies from

several fields of computer science, such as recommender systems (RS), natural language processing (NLP) with its sub-discipline named-entity recognition (NER) and information retrieval (IR) for the capturing and sharing of such knowledge.

Due to the high number of different components of the RAGE ecosystem and the corresponding content types, it is very important to manage the flow of knowledge in the RAGE ecosystem. There are different ways by which knowledge can be captured, represented, managed, further developed and re-used, i.e., shared and disseminated to other users. Thus, it is not sufficient to provide tools that already exist in the initial design of the RAGE ecosystem. It is important to know which kind of content types are commonly used together, which content items rely on other content items, whom to ask if ecosystem users have questions regarding specific assets, and much more. Therefore, we have decided to develop a hybrid approach that aims at extracting tacit user knowledge and implicit content knowledge from data, and share this emergent knowledge with the users of the RAGE ecosystem. One pipeline in this hybrid is the NLP-IR system. In this paper we will introduce our overall concept for this system. In our cloud-based architecture, the NLP-IR system is part of the RAGE computing cloud, which focuses on high-performance computing (HPC) tasks to support all functionality to provide the NLP-IR services defined by the use cases. Therefore, in this paper we describe RAGE, our overall NLP-IR approach and combine both with a data distribution and cloud computing based architecture.

Before we describe the cloud architecture, we take a short look at our hybrid approach, which uses methodologies from several fields of computer science, such as recommender systems (RS), natural language processing (NLP) with its sub-discipline named-entity recognition (NER) and information retrieval (IR) for the capturing and sharing of such knowledge.

2. Related Work

The base technology of the RAGE ecosystem will be built upon the so-called Educational Portal (EP) tool suite that was developed by the software company GLOBIT.[4] The EP tool suite was developed as a solution for the growing need of scientific communities to manage their document and media collections as well as their educational and other kinds of knowledge resources. Furthermore, one of its additional purposes is to support continuous professional education and training of practitioners, experts and scientists who are members of professional communities of practice or scientific communities. The EP tool suite has been developed in order to provide an internet-platform which allows for interaction among community members and access to their digital assets. Such assets include, e.g., published papers, lecture videos, abstracts and entire online courses. A reference application of the EP tool suite is, e.g., the United European Gastroenterology (UEG) association's educational platform 2. In order to find information in the EP, a UEG community member can either search for specific words in all texts or set a filter for specific categories provided by the communities. UEG experts have created domain-specific, hierarchical categories to which a given document can belong. In order to make use of this feature, the need to assign specific categories to given documents arises.

Assigning categories to digital documents can be done either manually or automatically. The manual approach yields the best results but has two intuitive drawbacks: domain experts' time is a sparse resource and every categorization created is subjective to the individual expert's opinion. The automated assignment of a specific text to a given category is commonly called text categorization (TC, also known as text classification or topic spotting).[5] The initial class paper is largely based on Mohri et al.[6] The UEG's EP-based application solution is based on the TYPO3 content management system (CMS). The underlying architecture paradigm is the service-oriented architecture paradigm. Before the work of Mohri et al.,[6] the EP employed an automated TC approach using the Apache Solr 3 enterprise search engine to calculate scores for individual terms. The higher the score, the more representative of the document the term is. The terms used were the names of the available categories. If the score exceeded a predefined threshold, the document was automatically assigned to the category represented by the term.

Besides its application by the UEG's EP-based application solution, the EP tool suite was already used in the European project Alliance Permanent Access to the Records of Science in Europe Network (APARSEN).[7] The EP tool suite offers a wide variety of tools. This includes a web-based, user-friendly course-authoring tool, and content & knowledge management tools for the management of documents, multimedia objects, taxonomies and more. Figure 1 displays the components and services in the Educational Portal tool suite underlying the RAGE ecosystem. The EP's three-tier architecture is divided into core functions, extensions and the data storage.

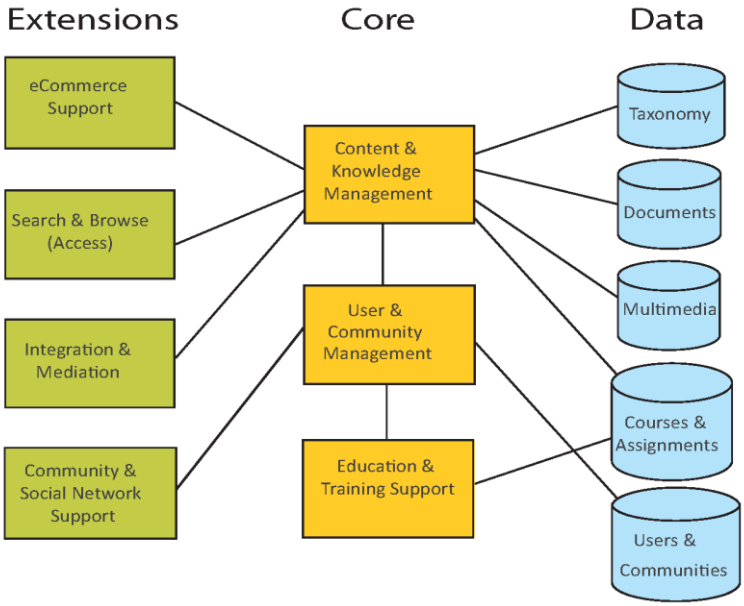

Fig. 1. Architecture of the EP

In the NLP-IR, part our work is motivated by several concepts that use NLP to support knowledge capturing in domain-specific environments: Nobata, Cotter et al.[8] present an integrated IR system for biology that is built upon conditional random field (CRF)-based named-entity recognition. Ananiadou, Thompson et al.[9] developed an IR system based on text-mining techniques that provides access to documents for the education community. The NLP-IR component has to address the heterogeneous structure of the RAGE ecosystem and will rely on a couple of base techniques: "Text classification based on support vector machines has been known for one and a half decades. Our classification approach for the wiki and other text-based components relies on the base ideas of SVM text classification as published by Joachims[10] and Sebastiani.[5] Filipova and Hall[11] demonstrated how to classify videos based on textual meta-data as well as on video content, a concept which could be used in the media archive component. Swoboda[12] developed an SVM-based classifier for the medical domain, which works with the bag of words approach. Our work will extend his approach, not using the bag of words approach but using features extracted from the RAGE ecosystem's content, knowledge resources and software assets. Ritter, Clark et al.[13] present how to use NER in social media (Twitter), which we will use to extend the knowledge-capturing features of the social network system integration support and for integrating knowledge-capturing features in the LMS of the RAGE ecosystem. Dascalu, Dessus et al.[14] developed ReaderBench, an NLP-based system tailored to the needs of learning and educational environments. ReaderBench will also be a part of the RAGE ecosystem's core system. Dit, Revelle et al.[15] provide a comprehensive overview of concepts for feature extraction in source code to be used, for example, in software repositories.

Our ideas about cloud are based on several concepts. First of all, we reference GATECloud, developed by Tabland, Roberts et al.[16] GATECloud is a cloud-based NLP solution of the GATE framework, which we mentioned before. Klenner, Bergmann et al.[17] present UIMA-HPC, an unstructured information management applications

(UIMA)-based NLP system that uses grid-based HPC. This project shows which amount of CPU capacities may be used for large-scale NLP purposes. We use it as a starting point to transfer this grid-based UIMA approach to a cloud-based service. Evangelinos, Hill et al.[18] give an example of how HPC applications can be transferred to cloud-based services. Our approach on cloud-based SVM classification is primarily based on Pouladzadeh, Shirmohamaddi et al.,[19] who present a cloud-based SVM classifier for object classification. They also show that in their approach the cloud-based SVM classifier outperforms a traditional implementation with LIBSVM. Apache SolrCloud is a cloud-based implementation of the well-known open-source search server Apache Solr.[20] SolrCloud could be the base of our IR subsystem described in the architecture section.

3. NLP-based Information Retrieval Support (NLP-IR)

3.1. NLP-IR Use Cases (as a Cloud Service)

The NLP-IR component of RAGE will provide a search functionality for RAGE. Due to the heterogeneous structure of the content, knowledge resources and software assets in the RAGE ecosystem, the NLP-IR component will cover two dimensions. On one side, the NLP-IR component will have to deal with the multiple types of content in the different RAGE modules, for example text documents, multimedia content, source code, social media content etc. On the other side, an adaptation of the functional requirements of the applied gaming sector is necessary, represented by 36 RAGE asset categories and existing taxonomies. So from the NLP-IR point of view, we talk about two dimensions of NLP-IR support in this project. Based on these assumptions, we defined two major use cases for NLP-based IR support, which are keyword search and faceted search. In the case of keyword search, the user specifies one or more keywords and is able to filter and refine his search results by a multi-dimensional categorization scheme, which covers both dimensions mentioned above. In a faceted search, the user does not specify any keywords but uses solely the categorization to access assets in the RAGE ecosystem.
From a user perspective, both use cases have the potential to be "classical" use cases for a cloud implementation. While all the data as well as the functionality may be covered by two cloud systems (RAGE Storage Cloud and RAGE Computing Cloud), on the user side just a simple interface (browser, app) is necessary to provide ubiquitous and smart user access.

3.2. NLP-IR - Working Hypothesis and Research Questions

To help explain our NLP-IR approach, we give a short description of three working hypotheses that are the basis of NLP-IR. Our first hypothesis is that a bag of words approach is not the appropriate concept for IR support in the RAGE ecosystem due to its heterogeneous content structure. As there are assets that will not work with full-text IR (e.g., videos or software libraries) we try to extract the relevant features using NLP either from text documents as well as from meta-data. Our idea of using NER for this kind of knowledge extraction and document classification is primarily based on Guo, Xu et al.[21] These authors found that 71% of user search queries contain named entities. Based on this observation, our second hypothesis is that in addition to queries, named entities are also a good feature to extract from the content to support search and access. Our third hypothesis is that in domain-specific content, the weight of named entities as an IR feature is even higher than in multi-purpose IR. Nadeau and Sekine[22] show several examples of successful domain-specific named-entity recognition. Because we work in the highly specialized domain of applied gaming, we believe that a domain-specific approach similar to that presented in the paper mentioned before will work in this case too. Based on these hypotheses, the NLP component should rely on state-of-the-art named-entity recognition to identify named entities and deal with heterogeneous content. Due to the two dimensions of NLP-IR support mentioned above, there will be an adaptation of the NER component to the different types of content and the domain-specific asset categories necessary. As these three (in this context yet unproven) working hypotheses are the main anchor of the NLP-IR sub-project, the development of an appropriate evaluation strategy will be a central aspect of our future work. The three working hypotheses lead to the following main research question of the overall project:

(3) Is an NLP- and classification-based IR approach in a domain-specific software repository indeed more efficient than traditional bag-of-words-based IR-approaches?

Hence, from a technical perspective we will investigate which kinds of named entity categories besides the standard categories (Name, Place and Organization) are suitable for the applied gaming sector. First (yet-unevaluated) examples for this are "Story-Character," "Player-Class" and "Emotion." Named-entity recognition, besides the standard categories, requires the use of NER components that can be trained on special named-entity classes of the applied gaming sectors. For this we use NER based on conditional random fields (CRF). One of the state-of-the-art named-entity recognizers fulfilling this requirement is the NER component of Stanford CoreNLP, which we will use in this project.[23] To realize interoperability and to support our modular approach, we are going to integrate this NER component in a standard NLP framework such as GATE[24] or Apache UIMA[25] using the appropriate wrappers.

3.3. NLP-IR Architecture

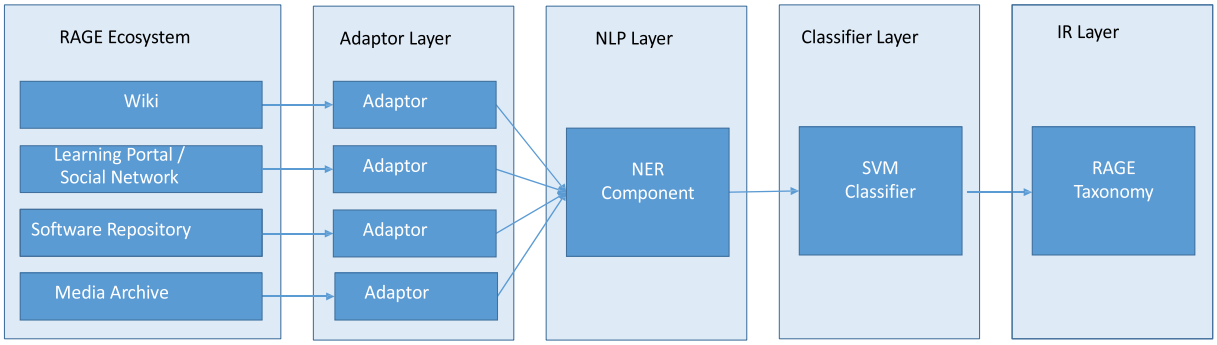

Fig. 2. NLP-IR Workflow

The main idea is to extract named entities from the data and use these to automatically categorize documents in RAGE so that they can be used as taxonomy in faceted search. Figure 3 displays a pipeline representing our multi-layer architecture for the NLP-IR workflow. The primary layer is the RAGE ecosystem with its sub-components, as described above. The adapter layer is intended to connect the relevant RAGE ecosystem subsystems containing different types of content with the NER component. Adoption in this context essentially means to extract (textual) meta-data from the non-text assets such as multimedia content or software components and to normalize unstructured content, e.g., from social network systems and the learning management system, to get consistent input for the NER component. In the following NLP layer, a tailored state-of-the-art NLP component will be used to extract the named entities (NE-chunks) from the textual input. The output of this layer is a vector whose features are NE-chunks to be processed in the following classifier layer. During the project we will investigate whether a binary or a weighted output vector is suitable for the following classifier layer. The classifier layer, which contains the SVM-based classifier, is used for the document classification within a domain-specific taxonomy. Finally, the pipeline is concluded by the information-retrieval layer, which supports keyword search as well as faceted search based on the taxonomy built by the classifier. The output of the IR layer is merged with the result sets of the recommender system (RS), addressing our hybrid approach, and will be presented to the user on the GUI. Our main goal with this layered architecture is to allow a modular and independent development of the several components after inter-layer interface specification.

3.4. RAGE Asset Categories

To support access to the heterogeneous contents of the RAGE ecosystem, we introduce a hierarchical classification scheme, which will be used to provide a structure for the user to browse through. Therefore, on one side our categorization scheme is initially based on 36 categories of software assets that support applied game development and will be delivered through the RAGE ecosystem.

3.5. SVM Classifier

To categorize the documents in RAGE using the extracted categories in the NER component, an SVM classifier is applied. Due to the potentially complex structure of the assets, we prefer an SVM-based approach to straightforward approaches of text categorization such as probabilistic or linear techniques as described by Sebastiani.[5] To categorize the RAGE content based on the classification scheme outlined above, an SVM classifier will be trained on the domain-specific NE-chunks produced by the NER component. With this approach we hope to increase the quality of the classification compared to the full-text approach with regard on the RAGE use cases. To implement the SVM classifier we will rely on the state of the art SVM library *LibSVM*.[26]

3.6. Evaluation

Although the evaluation strategy will be presented in detail in a future paper, here we share the main idea of how we are going to evaluate our designs. During the implementation we are going to evaluate cloud user experience using structured interviews to discover the benefits and drawbacks of our cloud-based approach. For this we will design standard-use cases and a test collection to be tested with users of the RAGE project in a case study. The user-based evaluation is intended to compare the cloud and the non-cloud approach. Besides utilizing those stakeholder-interviews to evaluate the cloud approach, we are going to use a semi-supervised standard evaluation methodology to evaluate the quality of the NLP-based classification and IR system based on test collections of the RAGE project. These standard evaluation methodologies (e.g., recall, precision, F-measure) are described by, among others, Croft, Metzler et al.[28] and Manning, Raghavan et al.[29] and are used to answer research question 3, whether an NLP (NER)- and classification-based IR approach outperforms the bag-of-words IR approach.

4. Towards Cloud Support

In the previous sections we described the basic ideas of the RAGE ecosystem and the NLP-IR support for knowledge capturing in the heterogeneous content within it. Now we want to discuss how cloud technology can enhance our non-cloud approach to improve the outcome of the overall system. In the first step we take a look at the several sub-systems of the NLP-IR architecture and discuss how they can be implemented using cloud technology. In the second step we present a common architecture that integrates both cloud and non-cloud subsystems to implement a complete system. Our discussion is primarily based on the NIST cloud definition.[27] In this paper we focus on cloud support in the NLP-IR system, but we also address aspects of data distribution in the RAGE storage cloud.

4.1. Use Cases for Cloud Support

In the following section we investigate which possible use cases for cloud support exist within the NLP-IR system. For that, we consider some of the "Essential Characteristics" proposed by NIST in their widely used definition of cloud computing. Taking a look at them, it comes to mind that the "Rapid Elasticity" characteristic in particular may be usefully applied to our overall NLP-IR and will lead to a reasonable use case. Both NER and SVM classification are tasks that may require a high volume of CPU time depending on the volume and complexity of the input data. At the same time, both components are only in use when new (heterogeneous) content is added to the RAGE ecosystem, which has to be classified in order to support knowledge capturing and access support, as described above. This means that there is a high volatility in this component: While most of the time there is no workload on the components, there are other (relatively short) times where the CPU load is relatively high when new content has to be analyzed and classified. By using rapid elasticity, an efficient and goal-oriented application of the resources can be ensured. Combined with the measured service characteristic in a third-party model, rapid elasticity may reduce costs for computing power in our use case while increasing usability by providing a suitable amount of CPU time at the same time. As the RAGE ecosystem and NLP-IR in the actual architecture are designed as non-cloud web services that are used by thin clients, the broad network access characteristic of cloud computing also may be applied in a cloud-based approach. For the end user there will be no distinction between a cloud and a

non-cloud RAGE/NLP-IR system. In the use case described above, the two other characteristics "resource pooling" and "on demand self-service" in our architecture also primarily refer to provision and use of CPU resources for NER and SVM. If the user needs them, he doesn't have to care about the underlying resources (on demand), while the provider, in house or third party, may share the resources using resource pooling with other (scientific) cloud projects. Based on these considerations, there are good reasons to develop a cloud-based version of NLP-IR in RAGE. Therefore, the strongest characteristic indeed is the rapid elasticity, which was mentioned first.

4.2. Service Model

The next step is to decide which service paradigm proposed by NIST should be used in the architecture. For both the RAGE storage cloud with data distribution and the RAGE computing cloud with NLP-IR-INPUT, NLP-IR-HPC and NLP-IR-OUTPUT system we decided to utilize the platform as a service (PaaS) paradigm throughout the complete pipeline. This paradigm allows us to use a suitable abstraction of the underlying hard- and software resources while allowing us to develop our own solution based on the technology of the runtime environment. As the candidates for the underlying NLP/NER framework are both based on Java technology, we will have to choose a scalable cloud-based Java runtime environment.

4.3. Deployment Model

The NIST definition gives four different deployment models of clouds: public, private, community and hybrid. In our architecture we have chosen the community cloud, as the applied gaming developer community is a "classic example" of a community described by NIST: "by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations)."[27] The cloud may be managed by a stakeholder of the applied gaming developer community or an external cloud provider, depending on the resources needed for the implementation. The question of which kind of provider model is chosen influences if the characteristics "measured service" and "resource pooling" may be used to add value to the project, as both primarily refer to providers who operate a broad cloud infrastructure. As indicated, the community cloud approach allows the users to deal with their core business (AG development) without having to deal with IT management, as all necessary services are provided by the RAGE storage cloud and the NLP-IR cloud. If the community cloud is operated by a trusted third party, it is even possible to operate multiple distinct instances of the RAGE cloud services for competing communities in one cloud environment. This allows the deployment of distinct RAGE instances as a whole in different sub-communities of the applied gaming community.

Besides the pure community cloud approach, a hybrid approach seems to be also constructive. In this deployment the mission-critical resources and specializations are deployed in a community cloud. In the use case, this applies to the NLP and IR resources as specialized, for example, as well as to the storage of secret and business-critical company information. While these critical assets remain internal to the community, public cloud services may be utilized to integrate open-source services for software development, presentation and storage of non-critical assets.

4.4. Cloud-based NLP-IR Architecture

Our architecture is divided into four cloud-based subsystems: the STORAGE CLOUD, NLP-IR-INPUT, HPC and NLP-IR-OUTPUT subsystems. In our actual approach, these four subsystems are distinct and distributed due to their heterogeneous structures. The four systems may be divided into two subgroups: data distribution and cloud computing. The data distribution subgroup contains the RAGE contents, whereas the cloud computing subgroup is used to provide the services of the NLP-IR architecture; the NLP-IR system has to deal with a high I/O load. The focus of the HPC system is primarily to provide CPU time. The NLP-IR-OUTPUT system will be optimized for database and index access. In a later development stage after evaluating cloud platforms, we may come to the conclusion that two or three subsystems may be combined on one common cloud platform.

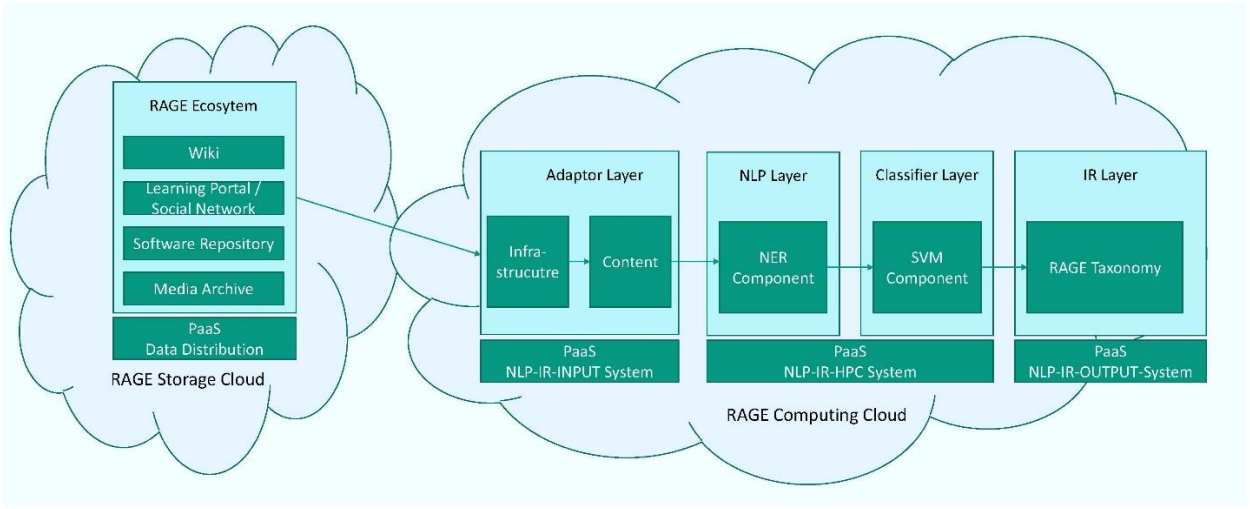

Fig. 3. NLP-IR Cloud Architectural Overview

The cloud storage offers all the data sources of the RAGE system in a distributed environment. A data distribution approach follows the classification of the RAGE knowledge sources shown in section 1. The advantages of the cloud storage compared to non-cloud storage are the scalability and adaptability of the storage regarding the heterogeneous content. The storage infrastructure is hidden from the user and tailored to the contents, such as multimedia with high volume and I/O load or textual documents with smaller file sizes. The end user and following cloud services just use standard service-based interfaces. In the RAGE storage cloud we will rely on state-of-the-art cloud storage techniques.

The task of the NL-IR-INPUT subsystem is similar to the task of the adaptor layer of our non-cloud approach. Its main objective is to provide an interface to the cloud storage and the RAGE subsystems within it. Compared to our original approach, a second component has been added to the NLP-IR-INPUT subsystem: the infrastructure adaptor. This component is used to connect the cloud-based NLP-IR system to cloud- as well as non-cloud-based RAGE ecosystem repositories using appropriate abstraction techniques. Besides the infrastructure adaptor, you find the content adaptor, which fulfills the normalization of the heterogeneous contents for NER as described above.

The HPC subsystem is the core system of our cloud-based NLP-IR architecture, as it provides the main functionality of the system and which has the most benefit of the cloud deployment due to the rapid elasticity characteristic. It contains both the NER and the SVM components, which require high CPU load while in use. The integration of NER and SVM in this layer will follow state-of-the-art NLP/SVM cloud-based implementations, as mentioned in related work.

The OUTPUT subsystem contains the IR components (index, taxonomy, user interface). Here we are planning to use a state-of-the-art cloud-based IR technology, as provided by Apache SOLR, for example. Due to our special needs, this server application will have to be tailored to our domain-specific issues.

5. Conclusion

We have introduced a design for a cloud-based architecture for knowledge extraction in a game industry knowledge-sharing platform. The intended architecture will extract named entities from the data and use these named entities for document categorization so that they can be applied in a taxonomy for faceted search. Regarding this architecture, we have identified three research questions. Also, we have hinted at a possible evaluation of our design to answer these research questions. The research is still in an early stage, and the evaluation scenarios will be elaborated further in the near future.

As we have shown, the NLP-IR in the RAGE ecosystem has the potential to be implemented using cloud technology to optimize use cases, particularly because in use cases that contain high volumes of heterogeneous data and require easy and fast scalability a cloud support is recommended. From an administrator's perspective, a cloud-based implementation offers more flexibility and usability due to the reduction in installation and operation efforts. This allows users to concentrate on their primary objective of developing applied games (which is the main goal of the RAGE project). As a first consequence, the component design of our non-cloud based approach is tailored so that we will use standard components which are "cloud ready," as mentioned above, to ultimately ensure easy implementation of a non-cloud solution as well as a cloud-based solution.

Acknowledgements and Disclaimer

This publication has been produced in the context of the RAGE project. The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 644187. However, this paper reflects only the author's view and the European Commission is not responsible for any use that may be made of the information it contains.

References

1. Nonaka I, Takeuchi H. *The knowledge-creating company: How Japanese companies create the dynamics of innovation*. New York: Oxford University Press; 1995.
2. Salman M, Hemmje M, Heutelbeck D. Towards Social Network Support for an Applied Gaming Ecosystem. In: *Accepted for: ECGBL, European Conference on Games Based Learning*; 2015
3. Salman M, Hemmje M, Heutelbeck D, Fuchs M, Brocks H. Towards Social Media Platform Integration with an Applied Gaming Ecosystem. In: *Submitted to: SOTICS, The Fifth International Conference on Social Media Technologies, Communication, and Informatics*; 2015.
4. Globit.com. Educational Portal; 2015. Available at: http://www.globit.com/products-services/educational-portal/ [accessed 13.05.2015].
5. Sebastiani F. Machine Learning in Automated Text Categorization. *ACM Comput. Surv.* 2002;**34**(1):1–47.
6. Mohri M, Rostamizadeh A, Talwalkar A. *Foundations of Machine Learning:* The MIT Press
7. Schrimpf S. APARSEN - Alliance Permanent Access to the Records of Science in Europe Network. *Dialog mit Bibliotheken* 2014;**26**(2):52–3.
8. Nobata C, Cotter P, Okazaki N, Rea B, Sasaki Y, Tsuruoka Y, et al. Kleio: A Knowledge-enriched Information Retrieval System for Biology. In: *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM; 2008, p. 787–788.
9. Ananiadou S, Thompson P, Thomas J, Mu T, Oliver S, Rickinson M, et al. Supporting the education evidence portal via text mining. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 2010;**368**(1925):3829–44.
10. Joachims T. Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec C, Rouveirol C, editors. *Machine Learning: ECML-98:* Springer Berlin Heidelberg; 1998, p. 137–142.
11. Filippova K, Hall KB. Improved Video Categorization from Text Metadata and User Comments. In: *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM; 2011, p. 835–842.
12. Tobias Swoboda. Towards effectivity augmentation of automated scientific document categorization by continuous feedback, Master Thesis. Hagen; 2014.
13. Ritter A, Clark S, Mausam, Etzioni O. Named Entity Recognition in Tweets: An Experimental Study. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA Association for Computational Linguistics; 2011, p. 1524–1534.
14. Dascalu M, Dessus P, Trausan-Matu Ş, Bianco M, Nardy A. ReaderBench, an Environment for Analyzing Text Complexity and Reading Strategies. In: Lane H, Yacef K, Mostow J, Pavlik P, editors. *Artificial Intelligence in Education:* Springer Berlin Heidelberg; 2013, p. 379–388.
15. Dit B, Revelle M, Gethers M, Poshyvanyk D. Feature location in source code: a taxonomy and survey. *Journal of Software: Evolution and Process* 2013;**25**(1):53–95.
16. Tablan V, Roberts I, Cunningham H, Bontcheva K. GATECloud. net: a platform for large-scale, open-source text processing

on the cloud. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 2013;**371**(1983):20120071.
17. Klenner A, Bergmann S, Zimmermann M, Romberg M. Large scale chemical patent mining with UIMA and UNICORE. *Journal of Cheminformatics* 2012;**4**(1):1–2.
18. Evangelinos C, Hill C. Cloud computing for parallel scientific HPC applications: Feasibility of running coupled atmosphere-ocean climate models on Amazon's EC2. *ratio* 2008;**2**(2.40):2–34.
19. Pouladzadeh P, Shirmohammadi S, Bakirov A, Bulut A, Yassine A. Cloud-based SVM for food categorization. *Multimedia Tools and Applications* 2015;**74**(14):5243–60.
20. Apache Foundation. SolrCloud; 2014. Available at: https://cwiki.apache.org/confluence/display/solr/SolrCloud [accessed 01.07.2015
21. Guo J, Xu G, Cheng X, Li H, editors. *Named entity recognition in query:* ACM; 2009.
22. Nadeau D, Sekine S. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 2007;**30**(1):3–26.
23. Manning CD, Surdeanu M, Bauer J, Finkel J, Bethard SJ, McClosky D. The Stanford CoreNLP Natural Language Processing Toolkit. In: *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*; 2014, p. 55–60.
24. Cunningham H, Maynard D, Bontcheva K, Tablan V. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*; 2002.
25. Ferrucci D, Lally A. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering* 2004;**10**(3-4):327–48.
26. Chang C, Lin C. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2011;**2**(3):27:1 - 27:27.
27. Mell P, Grance T. The NIST definition of cloud computing. *National Institute of Standards and Technology* 2009;**53**(6):50.
28. Croft WB, Metzler D, Strohman T. *Search engines: Information retrieval in practice*. Boston: Addison-Wesley; 2010.
29. Manning CD, Raghavan P, Schütze H. *Introduction to information retrieval*. New York: Cambridge University Press; 2008.