CloudRef – Towards Collaborative Reference Management in the Cloud

Oliver Kopp[1], Uwe Breitenbücher[2], and Tamara Müller[2]

[1] IPVS, University of Stuttgart, Germany, kopp@ipvs.uni-stuttgart.de
[2] IAAS, University of Stuttgart, Germany, {lastname}@iaas.uni-stuttgart.de

Abstract. With the help of literature management software, references can be collected, managed, and exported in bibliographies. Online resources offer functionalities to import references into reference management tools. However, the entries are often incomplete or faulty. CloudRef proposes to solve this issue by employing votings for new references and updates of references. To further foster collaboration, comments on PDFs can be shared among the users of CloudRef.

1 Introduction

When writing a scientific paper one has always to deal with an plethora of literature on the topic. Reference management software was invented to support researchers in that regard: The tools are used to collect literature, manage references, and export bibliographies. They provide an efficient way to keep an overview of a large amount of literature. Numerous tools provide the opportunity to manage knowledge about references inside comments, notes, or tags. Researching is often a collaborative task demanding that literature management software should support collaboration. This includes sharing references and comments with other users or people who use another literature management software. Sharing comments with others may be beneficial 1) to ease understanding the paper itself and 2) to ease finding relevant papers, because indexable text is provided.

There are multiple resources on the web offering searching for literature such as "Google Scholar"[3] or "The Collection of Computer Science Bibliographies"[4]. Many of them offer the functionality to import a reference into the preferred reference management software. However, they often provide incomplete or faulty reference entries. One exception is MathSciNet, where more than 20 persons take care of the quality of the references [8]. This quality assurance, however, is not implemented by all publishers.

A correct and complete entry is required for a correct reference list, which is a prerequisite for high-quality publications. Many programs for managing references provide a mechanism to detect missing required fields and highlight these entries to show the user that they are incomplete. However, this is not sufficient, because wrong information is not detected and it is tedious to find the correct missing information. The users have to check each reference entry manually to ensure correctness.

[3] https://scholar.google.com
[4] https://liinwww.ira.uka.de/bibliography

N. Herzberg, C. Hochreiner, O. Kopp, J. Lenhard (Eds.): 10th ZEUS Workshop, ZEUS 2018, Dresden, Germany, 8-9 February 2018, published at http://ceur-ws.org/Vol-2072

Hence, the goal of CloudRef is to provide a cloud-based web application for collaborative reference management with to main features:

1. CloudRef provides quality assurance by voting on bibliographical references to ensure complete and faultless references.
2. To support the cooperation of several people CloudRef enables to post comments to literature at different levels of visibility.

Sect. 2 presents related work on the field. Subsequently, Sect. 3 outlines the demonstration of CloudRef followed by a description of the implementation (Sect. 4). Finally, Sect. 5 provides a discussion and an outlook on future work.

2 Related Work

SoJa (Social JabRef [7]) introduces a social network among users. To establish a source of high-quality entries, for each topic, a user maintaining these entries has to be chosen in the community. There is no voting mechanism in place and comments can only be shared by embedding them into the BibTeX file. Haase et al. [10] assume that there are BibTeX databases with high-quality entries and that the issue is to identify duplicates and to find entries. To tackle these issues, they present the system Bibster. SharRef [22] focuses on sharing bibliographic data among groups and offers both a Web-based Client and a Java-based Desktop Client. Quality of entries is assured by having bibliography entries and automatically-updated shadows of them. There is no internal voting or commenting system in place. SocioBiblog [17] relies on the authors publishing their bibliographic data correctly on their homepages. There is no way presented on how to ensure quality of the resulting BibTeX entries. BibSonomy [3] and its variant PUMA [4] offer to collect publications. It is possible to edit bibliographic data [6]. While there is a history function[5], all edits are immediatly visible so there is no suggestion process as we propose. Academic search engines are surveyed by Ortega [16]. Additionally, there is OverCite [19] aggregating search results in a CiteSeer-like way. These tools offer search capabilities only and not a defined way to correct bibliographic entries. Beraka et al. [5] present a system for exchanging bibliographic information of scientific review and survey articles. Users can approve or disapprove bibliographic entries, but it is unclear how contradicting votes are treated. For presentation of surveys, SurVis [1] can be used. It is a read-only system without built-in functionalities of ensuring high-quality bibliographic data. Tkaczyk et al. [20] surveyed on reference parsers. They convert free reference text to a structured format. Thus, this is a way to get bibliographic data into a literature management system, but it is not ensured that the parsed data is of high quality itself. There is a movement on correctly citing software [18]. However, there is no quality control process proposed. Finally, we investigated 15 popular literature management systems[6] and none of them offers 1) a voting system on bibliography entries and 2) comments with dedicated visibility.

[5] Example: `https://www.bibsonomy.org/history/bibtex/57fe43734b18909a24bf5bf6608d2a09`

[6] `https://ultimate-comparisons.github.io/ultimate-reference-management-software-comparison`

3 Demonstration

After startup, the current prototype CloudRef shows a table of all references (Fig. 1). References with a green check mark have been reviewed and marked as high-quality. A new reference can be input using a form-based editor or by uploading a BibTeX file. A

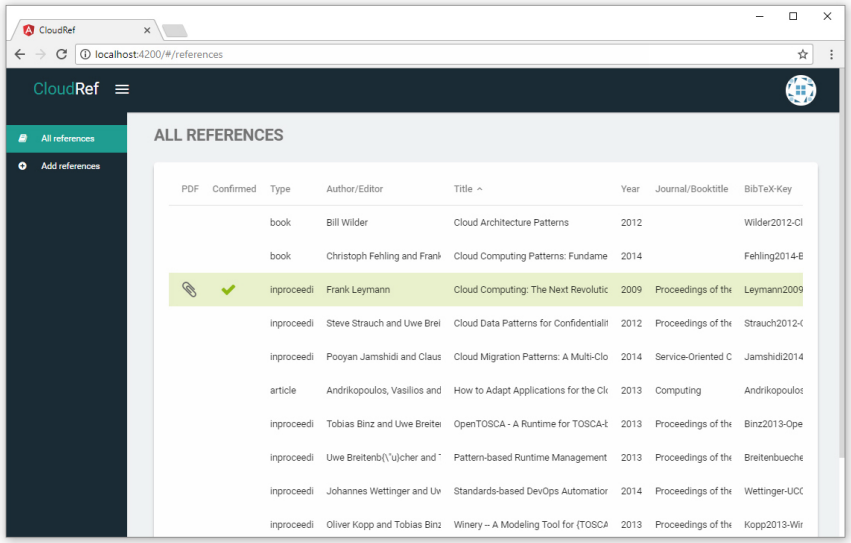

Fig. 1. Entry table showing references.

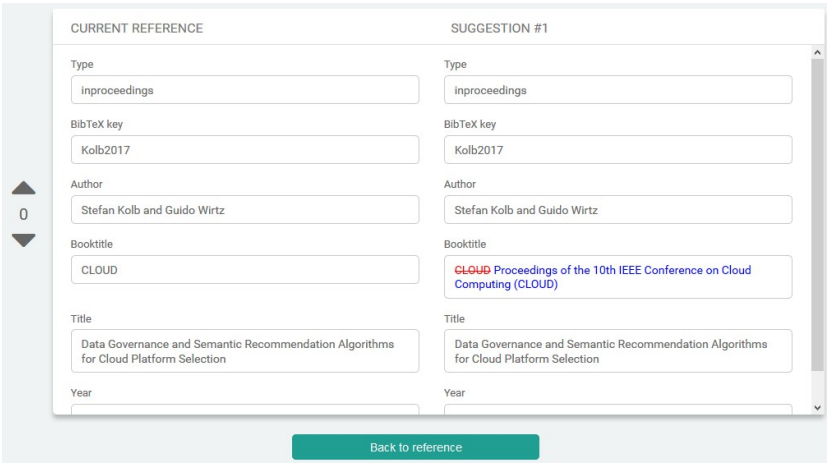

Fig. 2. Suggestion for modification with voting possibility. The dialog is based on JabRef's Merge Entries Dialog, `https://help.jabref.org/en/MergeEntries`.

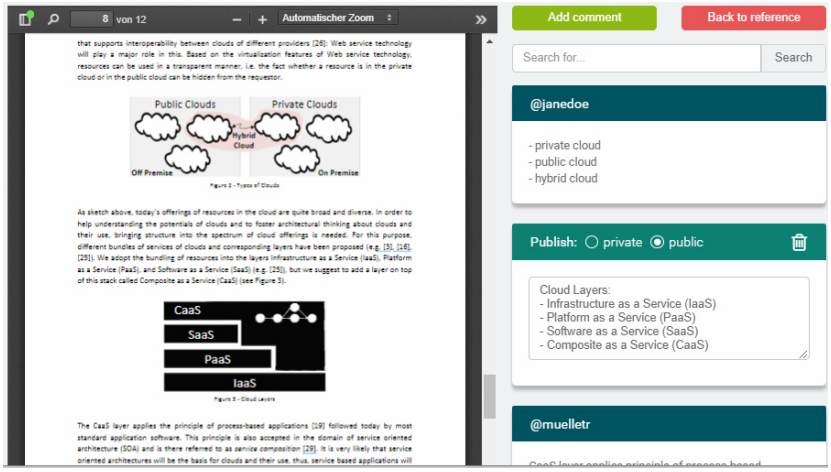

Fig. 3. PDF comments.

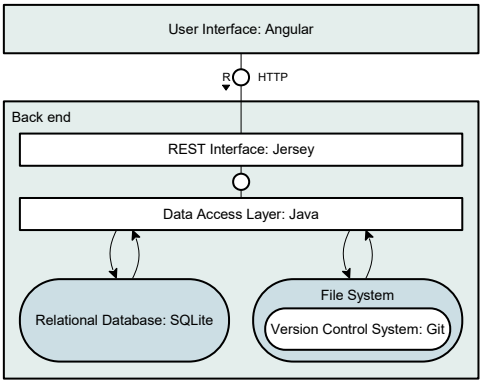

Fig. 4. Architecture of the CloudRef platform with implementation details. Notation: FMC [12].

suggestion for improvements can be done using the entry editor. A user can click on "See suggestions for modifications" and a respective dialog is shown (Fig. 2). On the left side, there are the voting buttons. The number indicates the number of votes. After five positive votes by different users have been reached, the suggestion is merged.

In case a PDF is attached, comments on the PDF can be made (Fig. 3). A user can set the visibility of his own comments to public or private. "Public" denotes that each logged in user can see the comment and "private" denotes that the user exclusively can see the comment. This helps newcomers to research to make private notes and the more experienced researches to share their comments.

4 Implementation

The architecture of CloudRef including the used technologies is presented in Fig. 4. We implemented CloudRef using Java and Angular. Regarding the storage, we decided to

put the comments into a SQL database, placing the PDFs directly into the filesystem, and versioning the bibliography entries using git. Each suggestion becomes a new branch in git. In case a suggestion is accepted, the branch is merged. This way, we did not have to reinvent the whole branching and merging concept, but could rely on git's possibilities. The implementation is published at `https://github.com/JabRef/cloudref/` and is offered as Docker image at `https://hub.docker.com/r/jabref/cloudref/`.

5 Discussion and Outlook

Currently, CloudRef is an initial idea for collaborative reference management. CloudRef is currently targeted at research groups with around ten persons. This ensures that quality standards established in a group is followed. The group, however, has to define its quality standards. For instance, there is currently no global agreement whether one should always put the abbreviation of a conference name in parentheses to the end of the conference title. The next natural step is to create a style guide for BibTeX and to integrate a checker into CloudRef similar to a GitHub pull request status checker [21].

When moving to a larger user base or offering CloudRef as public SaaS offering, additional concepts for reviewing and maintaining references have to be developed and the current voting concept has to be evaluated. For instance, the number of required positive votes is a variable to evaluate.

The history of each BibTeX entry is stored in the backend. Since there might be different views on a BibTeX entry, CloudRef should be able to show the history to the user. This enables him to propose another version of the entry based on historic proposals.

To provide more features for high-quality references, we plan to use the logic package of JabRef[7], transpile it to JavaScript using jsweet[8], and embed it in CloudRef. We also plan to integrate CloudRef's functionality into JabRef.

CloudRef is currently targeted as "Multiple Instances Service" [15]. To enable it being hosted using as "Arbitrary Instances Service," we are going to work on the backend implementation and make CloudRef a real cloud-native application [14]. This especially includes exchanging the storage layer by a PaaS one [13].

Finding related work is still a challenging task. To ease this, we aim for integrating a) the recommender system Mr. DLib [2] into CloudRef in a similar way it has been done for JabRef [9] or b) the user profile recommendations implemented by Bibster [11].

References

1. Beck, F., Koch, S., Weiskopf, D.: Visual analysis and dissemination of scientific literature collections with SurVis. IEEE Transactions on Visualization and Computer Graphics 22(1), 180–189 (Jan 2016)
2. Beel, J., Gipp, B., Langer, S., Genzmehr, M., Wilde, E., Nürnberger, A., Pitman, J.: Introducing Mr. DLib, a Machine-readable Digital Library. In: 11th annual international ACM/IEEE joint conference on Digital libraries (JCDL'11). ACM (2011)

[7] `https://www.jabref.org`

[8] `http://www.jsweet.org`

3. Benz, D., Hotho, A., Jäschke, R., Krause, B., Mitzlaff, F., Schmitz, C., Stumme, G.: The social bookmark and publication management system bibsonomy. The VLDB Journal 19(6), 849–875 (Dec 2010)
4. Benz, D., Hotho, A., Jäschke, R., Stumme, G., Halle, A., Lima, A.G.S., Steenweg, H., Stefani, S.: Academic Publication Management with PUMA – Collect, Organize and Share Publications. In: Research and Advanced Technology for Digital Libraries, pp. 417–420. Springer (2010)
5. Beraka, M., Al-Dhelaan, A., Al-Rodhaan, M.: A Bibliographic System for Review/Survey Articles. In: 2014 International Conference on Information Science & Applications (ICISA'14). IEEE (2014)
6. BibSonomy Team: Community posts (2018), https://www.bibsonomy.org/help_en/CommunityPosts
7. Datta, A.: SoJa: Collaborative reference management using a decentralized social information system. In: 6th International ICST Conference on Collaborative Computing: Networking, Applications, Worksharing (CollaborateCom 2010). IEEE (2010)
8. Dunne, E.: References and Citations (2015), https://blogs.ams.org/beyondreviews/2015/07/11/references-and-citations/
9. Feyer, S., Siebert, S., Gipp, B., Aizawa, A., Beel, J.: Integration of the Scientific Recommender System Mr. DLib into the Reference Manager JabRef. In: 39th European Conference on IR Research (ECIR 2017). Springer (2017)
10. Haase, P., Broekstra, J., Ehrig, M., Menken, M., Mika, P., Olko, M., Plechawski, M., Pyszlak, P., Schnizler, B., Siebes, R., Staab, S., Tempich, C.: Bibster – A Semantics-Based Bibliographic Peer-to-Peer System. In: 3rd International Semantic Web Conference (ISWC 2004). Springer (2004)
11. Haase, P., Stojanovic, N., Voelker, J., Sure, Y.: Personalized information retrieval in bibster, a semantics-based bibliographic peer-to-peer system. In: 5th International Conference on Knowledge Management (I-KNOW'05). p. 104 (2005)
12. Keller, F., Wendt, S.: FMC: an approach towards architecture-centric system development. In: 10th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems. IEEE (2003)
13. Kolb, S., Lenhard, J., Wirtz, G.: Application Migration Effort in the Cloud. Services Transactions on Cloud Computing (STCC) 3(4), 1–15 (2015)
14. Leymann, F., Wettinger, J., Wagner, S., Fehling, C.: Native Cloud Applications: Why Virtual Machines, Images and Containers Miss the Point! In: 6th International Conference on Cloud Computing and Services Science. SCITEPRESS (2016)
15. Mietzner, R., Leymann, F., Unger, T.: Horizontal and vertical combination of multi-tenancy patterns in service-oriented applications. Enterprise Information Systems 5(1), 59–77 (Feb 2011)
16. Ortega, J.L.: Academic Search Engines: A Quantitative Outlook. Chandos Publishing (2014)
17. Shakya, A., Takeda, H., Wuwongse, V., Ohmukai, I.: SocioBiblog: A decentralized platform for sharing bibliographic information. In: IADIS International Conference WWW/Internet (2007)
18. Smith, A.M., Katz, D.S., and, K.E.N.: Software citation principles. PeerJ Computer Science 2, e86 (sep 2016)
19. Stribling, J., Li, J., Councill, I.G., Kaashoek, M.F., Morris, R.: Overcite: A distributed, cooperative citeseer. In: 3rd Symposium on Networked Systems, Design & Implemention (NDSI'06) (2006)
20. Tkaczyk, D., Collins, A., Sheridan, P., Beel, J.: Evaluation and Comparison of Open Source Bibliographic Reference Parsers: A Business Use Case (2018), arXiv:1802.01168v1
21. Vasilescu, B., Yu, Y., Wang, H., Devanbu, P., Filkov, V.: Quality and productivity outcomes relating to continuous integration in GitHub. In: 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2015). ACM (2015)
22. Wilde, E.: Personalization of Shared Data: The ShaRef Approach. Journal of Digital Information 8(3) (2007)