DE GRUYTER

Information. Wissenschaft & Praxis 2017; 68(2–3): 163–170

Open Science

Afshin Sadeghi*, Johannes Wilm, Philipp Mayr und Christoph Lange

Opening Scholarly Communication in Social Sciences by Connecting Collaborative Authoring to Peer Review

Abstract: The objective of the OSCOSS research project on "Opening Scholarly Communication in the Social Sciences" is to build a coherent collaboration environment that facilitates scholarly communication workflows of social scientists in the roles of authors, reviewers, editors and readers. This paper presents the implementation of the core of this environment: the integration of the Fidus Writer academic word processor with the Open Journal Systems (OJS) submission and review management system.

Descriptors: Author, Publishing, Workflow, Business process, Integrated, Social Sciences

Öffnung der wissenschaftlichen Kommunikation in den Sozialwissenschaften: Verbindung von kollaborativem Schreiben und Peer Review

Zusammenfassung: Das Ziel des Forschungsprojekts OSCOSS „Opening Scholarly Communication in the Social Sciences" ist es eine kohärente Kollaborationsumgebung zu erstellen, die den wissenschaftlichen Kommunikationsworkflow von Sozialwissenschaftlern in den Rollen Autor, Gutachter, Herausgeber und Leser ermöglicht. Der Artikel präsentiert die Implementierung der Kerninfrastruktur des Projekts: die Integration der wissenschaftlichen Textverarbeitung Fidus Writer mit dem Open Journal System (OJS) als Einreichungs- und Begutachtungssystem.

*Korrespondenzautoren: Afshin Sadeghi, University of Bonn, Römerstraße 164, 53117 Bonn, E-Mail: sadeghi@cs.uni-bonn.de
Dr. Johannes Wilm, GESIS–Leibniz Institute for the Social Sciences, Unter Sachsenhausen 6-8, 50667 Köln,
E-Mail: mail@johanneswilm.org
Dr. Philipp Mayr, GESIS–Leibniz Institute for the Social Sciences, Unter Sachsenhausen 6-8, 50667 Köln,
E-Mail: philipp.mayr@gesis.org
Dr. Christoph Lange, University of Bonn, Römerstraße 164, 53117 Bonn und Fraunhofer IAIS, Schloss Birlinghoven, 53754 Sankt Augustin, E-Mail: langec@cs.uni-bonn.de

Deskriptoren: Arbeitsablauf, Autor, Veröffentlichungswesen, Geschäftsprozess, integriert, Sozialwissenschaften

Ouvrir la communication scientifique en sciences sociales en reliant la création collaborative avec l'évaluation par les pairs

Résumé: L'objectif du projet de recherche «Open Scholarly Communication in the Social Sciences» (OSCOSS) est la construction d'un environnement cohérent et collaboratif qui facilite le flux de la communication scientifique des sciences sociales dans les rôles des auteurs, les réviseurs, les rédacteurs et les lecteurs. Cet article présente la mise en œuvre du noyau de cet environnement: l'intégration du traitement de texte universitaire Fidus Writer avec le système de gestion d'expédition et d'examiner Open Journal Systems (OJS).

Descripteurs: Auteur, Déroulement du travail, Edition, Processus de réalisation, Intégré, Sciences sociales

Introduction

The objective of the DFG-funded OSCOSS research project (Mayr and Lange 2016) on "Opening Scholarly Communication in the Social Sciences" is to build a coherent collaboration environment that facilitates scholarly communication workflows (Sompel et al. 2004) of social scientists in the roles of authors, reviewers, editors and readers. A collaborative writing environment (Whitehead 2005), for which we chose the Fidus Writer academic word processor[1], is the hub of the overall environment. Further components that we are currently integrating include databases of metadata about scientific publications and research data sets; we are also planning to include repositories hosting the source code of data analysis software. This paper presents the integration of Fidus Writer with a sub-

---

1 https://www.fiduswriter.org.

mission and review management system; for this, we chose Open Journal Systems (OJS)[2].

In the "Background" section, we give an overview about the scholarly authoring and reviewing process, our intended approach and the supported workflows including authoring workflow, journal/conference editing workflow and reviewing workflow. The "Related Work" section describes popular state of the art collaborative writing and reviewing systems. In the "Requirements" section we outline the requirements from the perspective of authors, editors and reviewers. The integration of Fidus Writer and OJS is technically described in "Implementation". The paper ends with our "Conclusion".

Background

Manuscripts submitted to journals and conferences are typically written using word processors. A breakthrough in this area in recent years has been a move from traditional desktop applications to collaborative, web-based versions. Modern online platforms such as Google Docs[3], ShareLaTeX[4] and Overleaf[5] have made collaborative authoring significantly easier. The entire writing process can be completed online – from the writing of the first draft to the production of a camera-ready document. On the other side of the life cycle of scientific articles is the reviewing workflow (Bornmann 2011; Bornmann and Daniel 2010). Mature submission and review management systems support this workflow include the open source solution Open Journal Systems (Smecher 2008) and the centrally hosted solution EasyChair[6] which follows a freemium usage model; both are web-based platforms.

These two processes together form the main parts of scientific writing, and they are intertwined, as, after reviewing, the control over a manuscript is typically given back to the authors, who will then revise it. Nevertheless, to our knowledge there is currently no mature system that supports both steps in combination. Therefore, the reviewing workflow is typically realized in a way that requires manual file transfers at several points: after the authoring process is finished, the authors have to export a copy of the manuscript from the collaborative authoring system and submit it to the submission and review management sys-

tem. There, the reviewers have to download the manuscript, comment on it, upload it back to the system, which will notify the journal editor or the conference PC chair, who will take a decision and notify the authors, who read the comments and, unless rejected, apply revisions to the manuscript before submitting it once more. This review cycle can be repeated two or more times. In every such cycle, authors as well as reviewers have to download and upload a document, authors have to apply revisions to a document according to comments that are given as plain text or, in the best typical case, as annotations to a copy of the document itself. Unless the manuscript is submitted in an editable format, such as an office word processor format, but in a read-only format such as PDF, the authors have to apply revisions by viewing side-by-side their own document in their word processor and the reviewers' annotations in a document viewer application. This procedure is error-prone for authors, as they may overlook comments or apply revisions in the wrong place of the document. It is also cumbersome for the majority of reviewers who are not using PDF annotation tools but write even minor revision requests into their plain-text overall summary of their review, as they have to refer to texts by approximate references such as "in the 2nd paragraph of page 7".

Approach

With the OSCOSS platform, we aim at linking the two steps of authoring and reviewing scholarly articles. The omnipresence of the Web means that the involved systems already have ways to connect. We choose the free open source systems Fidus Writer and Open Journal Systems (OJS), a submission and review management system, as they have open plugin APIs. This means that all that is left is extending the functionality of each system by a plugin capable of communicating with the other system, e.g., via RESTful web service interfaces. This not only allows the submission of the articles directly to the reviewing environment – it even lets authors and reviewers interact directly on specific sections of an article, which is much more fine-grained than peer review is typically done at the moment.

Supported Workflows

In this article, we use the term "workflow" to denote a largely predetermined procedure that users in different roles will take through the framework. It varies slightly from journal to journal and from conference to conference.

---

2 https://pkp.sfu.ca/ojs.

3 https://docs.google.com/.

4 https://de.sharelatex.com.

5 https://www.overleaf.com.

6 http://easychair.org.

In this section we will describe the workflows that we considered necessary to be addressed by our implementation of the OSCOSS platform.

Authoring workflow. An author starts writing about his research and invites his collaborating researchers to participate. Each person could be responsible for a part of the manuscript; some authors might just review text written by other authors. The corresponding author submits the article to a conference or journal. After the review, the authors receive feedback, usually including some concrete advice on how to correct or update parts of the writing.

Journal/conference editing workflow. The journal editor or conference PC chair typically assigns one to three reviewers to review every manuscript submitted. The editor also takes the final decision on whether to accept or reject a manuscript or to request the authors to resubmit a revision. He makes this decision based on the review of the reviews or sometimes his own, additional review of the manuscript.

Reviewing workflow. The reviewing process starts when a reviewer opens an assigned manuscript. His role is to assess the quality of the manuscript. He is expected to appraise the manuscript as a whole but can also give fine-grained feedback on specific issues in specific places of the manuscript. The reviewer may provide the authors with constructive feedback suggesting how they could improve, shorten or extend the manuscript. Finally, the reviewing workflow includes providing general recommendations to the journal editor to help him to quickly decide on whether to accept or reject the manuscript or to request a revision from the authors. Depending on the policy of the conference or journal, this cycle can be repeated several times.

Related Work

There are several collaborative writing and reviewing systems available. In our inventory we found that the existing systems all have limitations.

Among existing collaborative writing systems are: Microsoft Office Online[7], Zoho Office[8], Etherpad[9], Google Docs[10], ShareLaTeX[11], Overleaf[12], and Authorea[13] (see Table 1).

---

7 https://products.office.com/en/office-online.

8 https://www.zoho.com/docs/office-suite.html.

9 http://etherpad.org.

10 https://docs.google.com.

11 https://www.sharelatex.com.

12 https://www.overleaf.com.

13 https://www.authorea.com.

Fidus Writer, the word processor of our choice, combines the advantages of several of these systems: It has a classic word processor interface enabling also non-technical people to use it; Fidus Writer is developed in Python and JavaScript and is open source. It can be installed locally, it offers scientific features such as citation and figure management in a simple What-You-See-Is-What-You-Get (WYSIWYG) interface, yet the editing interface only allows semantic rather than stylistic changes to the text. This means that users can specify that a certain part of the text is a headline of the second level, a link, or emphasized text, but the user cannot specify the font, font size or line height the way word processors allow it. This removes a number of uncertainties such as the question of whether a bold text by itself on a single line should be interpreted as an emphasized paragraph or a small headline. It also removes the problem of users misusing the interface – for example by manually entering 25 line breaks to obtain a page break. The fact that the text is available in its semantic form at all times means that it can be converted fully automatically without the need of human interpretation of its meaning and it can therefore easily be dealt with in any publishing pipeline – even directly to PDF in a web browser with JavaScript and CSS (Murakami and Wilm 2015). Receiving the text in an unclean format where the semantics are open to interpretation is a major problem for publishers (Wilm 2015).

Other existing collaborative academic authoring systems range from ShareLaTeX and Overleaf (Perkel 2014), which allow for the editing of LaTeX code online and are used mainly by programmers and other technically minded people who do not mind reading and writing in code, to Google Docs, Zoho Office and Microsoft Office 365 (both Desktop and Online). These systems offer an interface fairly similar to the desktop version of the traditional Microsoft Word most users are familiar with, which makes them more usable for non-tech experts, but they lack features specific to scientific writing, and they produce output that is similarly complex to deal with in the publishing process as any desktop word processor.

Among the common limitations of such systems there is the software license and terms of use. Most systems store their documents on the servers of the company that operates them; they are typically not available for local installation, which would allow for a custom configuration and keep sensitive data, e.g., of medical studies, within the scope of the security and data protection policies of the authors' organisations. ShareLaTeX is different in that it is open source and can be installed locally. Etherpad is another example of an open source collaborative text editor,

but we did not consider it in our comparison as it lacks scientific features.

OJS with a history of more than 10 years from the first release is the submission and review management system of our choice[14]. It is open source and is implemented in PHP. At the time of developing our integration code, OJS version 3 was under development; therefore, our implementation is based on this version. At the time of writing this article, more than 10,000 journals used OJS[15]. In version 3, OJS features a more dynamic interface than previously and supports different peer review configurations including single blind and double blind reviewing. It has an extensive documentation and free tutorials available.

To the best of our knowledge, none of the authoring and reviewing systems mentioned above have ever been integrated with each other. The only existing solution is the ARPHA Writing Tool[16], which is trying to create both a writing and reviewing system. In comparison to Fidus Writer, ARPHA authoring tool does not support realtime collaborative editing. Their reviewing system, which is used by less than a handful of publishers, is not as mature as OJS in terms of documenting and ease of use. In the next section we explain the requirements that we considered for an integrated system.

Table 1: Collaborative online writing systems.

System / Feature	Open source	Academic content	WYSIWYG	Export formats
Google Docs	No	Formulas	Yes	DOCX, ODT, PDF, HTML
Microsoft Office Online	No	–	Yes	DOCX, ODT, PDF
Zoho Docs	No	Formulas	Yes	DOCX, ODT, PDF, HTML
ShareLaTeX	Yes	Formulas, Citations	No	LaTeX, PDF
Overleaf	No	Formulas, Citations	No	LaTeX, PDF
Authorea	No	Formulas, Citations	Yes	DOCX, LaTeX, PDF
Etherpad	Yes	–	Yes	HTML
Fidus Writer	Yes	Formulas, Citations	Yes	DOCX, ODT, PDF, LaTeX, HTML

---

14 Other existing submission and review systems are: EJPress, EPRESS, ePublishing Toolkit, ESPERE, FontisWorks, OJS, SmartPublishing, Rapid Review, ScholarOne, Editorial Manager, etc.

15 https://pkp.sfu.ca/ojs/ojs-usage/ojs-stats/.

16 http://arphahub.com/.

Requirements

From the perspective of authors, editors and reviewers, an integrated system needs to comply with certain requirements to be useful. By reflecting on our own experience in each of these three roles, and by talking to the editors of the GESIS journals mda[17] and HSR[18], which will serve as pilots for evaluating our implementation in the scope of the OSCOSS project, we obtained the following list of requirements.

– Ease of use. The integration of the two systems is supposed to make the two workflows easier, so it should reduce the number of manual steps instead of increasing the complexity of the workflow by adding steps.

– Continuity of the workflows through the two applications. When a function of one application requires to call a function of the other system, this call must be performed in the background. Jumping from one system to the other must appear seamless to the author. A user who is registered in one system must also be known in the other one (single sign-on). When a reviewer logs into OJS, he must also be able to see his assigned manuscripts in Fidus Writer without having to log in a second time.

– Each system must support its part of the peer review process. For example, comments whose visibility is restricted by role (e.g. reviewers' comments who are only visible to the authors once approved by the editor), must be available in Fidus Writer, but none of the currently existing peer review services of OJS should be stopped from working; therefore, OJS still needs to support users who are not using Fidus Writer.

– Conformance to technical standards, e.g., RESTful API best practice guidelines, OAuth standards, etc.

– Support the import of manuscripts in widely used formats such as Microsoft Word. This is crucial to attract authors with a background of using different authoring systems, as it enables them to continue working on manuscripts they have started drafting outside of our integrated environment.

– The system needs to be able to export to a variety of formats, such as Microsoft Word, PDF and LaTeX. This is important to support the publishing workflows of as many publishers as possible.

– Support widely used conference and journal publishing templates and layouts, e.g., those of ACM or Springer.

---

17 http://www.gesis.org/angebot/publikationen/zeitschrift-mda/.

18 http://www.gesis.org/hsr/aktuelle-hefte/.

- Handling of graphics, diagrams and tables. Graphics and Tables are present in almost of articles. Therefore support for formulas and tables is crucial as a usability factor for the integrated system.
- Security. The interactions of the Fidus Writer and OJS side of the system over the network should introduce no new security flaws.

Implementation

OJS already provides online support for the reviewing workflow. In the classic workflow, authors register in OJS system and deliver their article in the form of a PDF or Word file upload. Reviewers have to download the document, review it and give their general feedback to authors and journal editors. Given that OJS is extensible via a plugin mechanism, we were able to extend it with a RESTful interaction API without changing any core OJS code. Representational state transfer (REST) Web services are a method of providing interaction between computer systems on the Internet using a uniform and predefined set of stateless operations.

Our first change to the conventional review workflow affects authors. They no longer need to register on the OJS site. By their submission of a new article through the Fidus Writer interface, we register them as the corresponding author on the OJS site. If the authors are already known to OJS, we link the new article to the previously registered author in the system. And while reviewing, sending the general feedback of the reviewers, i.e. the one that addresses the manuscript as a whole rather than specific parts, to the registered email addresses of the authors removes the need for authors to visit OJS altogether.

Fidus Writer already supports a collaborative authoring workflow. It allows multiple users to edit a document at the same time (Wilm and Frebel 2015). It allows authors to discuss a general topic among each other using a chat interface. It also supports comments to tag and do internal reviews of the draft in the authoring phase. To support the reviewing workflow, we extended Fidus Writer by adding user roles to it, each having specific permissions. During the review process, comments can have different levels of visibility. Based on the policy of our targeted journals, we added a permission mechanism based on roles into Fidus Writer. For example, authors see the comments of authors and reviewers, but each reviewer does not see the comments of the other reviewers. It was necessary to match user roles and permission levels of Fidus Writer to those of OJS to make them compatible, as shown in Table 2.

Table 2: Mapping of roles between OJS and Fidus Writer (FW).

OJS	FW
Author	Author
Reviewer	Reviewer
Journal Editor	Admin

In both OJS and Fidus Writer, roles are defined per document and can differ across documents. While a role in OJS is customizable, in Fidus Writer, the same user can only have one role per document and the permissions of each role are set. For example, a user can be an author, reviewer and the journal editor at the same time in the OJS but he can only have one role in Fidus Writer in relation to one document. Figure 1 shows the matching of users and the roles and how the definition of a role affects the role of a user in the system at the other end. The role of an author in Fidus Writer causes the creation a new author role in OJS and the creation of a reviewer role in OJS causes a similar role to be assigned in Fidus Writer. This allows Fidus Writer to authenticate reviewers via our extension in OJS. As the number of journal editors and administrators is usually limited for the two systems, we did not introduce admin and journal editor roles within the integrated system.

Architecture of the integrated system

Fidus Writer is implemented based on the Django framework for Python and features a JavaScript rich client; it follows the model view controller pattern closely. The controllers are configured to allow internal interaction based on HTML. We could extend these controllers to accept requests from outside that are coming from OJS side. To send requests, we used jQuery RESTful calls in the JavaScript parts of the implementation.

On the OJS side we used the plugin API to implement a "general plugin". General plugins in OJS allow the inclusion of other types of plugins. This plugin can manipulate the user interface and the database, and it can be notified when a specific function or page is called in the system. We used this feature in OJS to receive a notification when an editor assigns a reviewer to a submission and to contact Fidus Writer to create, if necessary, user accounts for the reviewers, and to grant them the necessary permissions. General plugins are also capable of accepting connections from the Web. This let us develop a RESTful API for OJS to accept calls from Fidus Writer. An example of that was
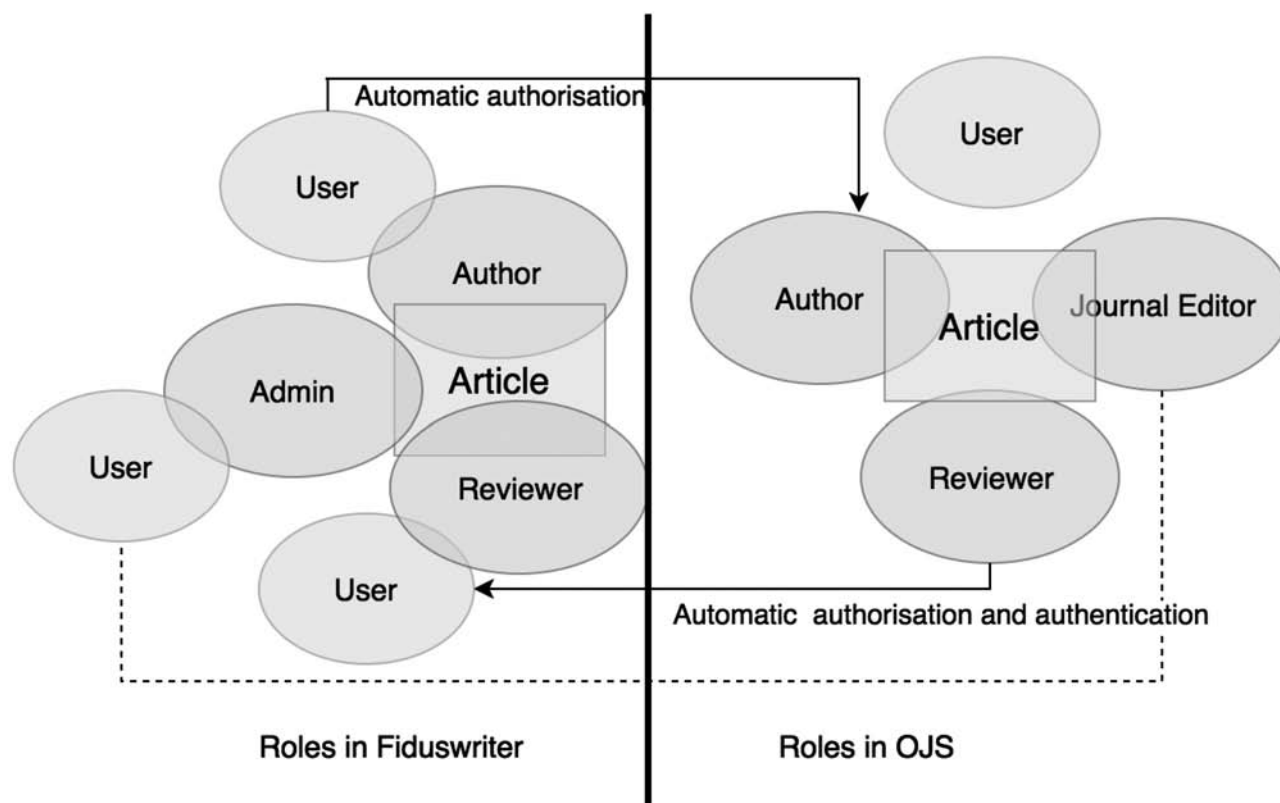
Figure 1: Matching of users and the roles in Fidus Writer and OJS.

providing a list of journals and their identifiers as the response to an HTTP GET request from Fidus Writer.

Figure 2 shows the interactions that connect the OJS and Fidus Writer of the integration system. We have recorded a video clip[19], that displays the interaction of Fidus Writer and OJS.

Conclusion and Outlook

We expect authors, reviewers and journal editors to be able to speed up their tasks related to authoring and reviewing manuscripts by using the integrated environment we have laid out in this article. We expect not only reduced time and thus cost but also fewer possible user mistakes (as could be made when emailing back and forth, converting, copying and uploading). Reduced administration time is another advantage of the integrated system because the accounts are created automatically and signing into one system does not make it necessary to log into the other system.

Our system is limited by the target environment for wich we designed our system. Over the Internet, the online connection of authors is influenced inherently by the network limitations, and the interface design is limited by being executing inside a browser. Rich text editing in browsers continues to be limited by a lack of standardization and development of relevant web technologies, as has been acknowledged even by the W3C and its member organizations (Berjon 2016).

We are planning to evaluate the system with social scientists from the communities of the mda and HSR journals, who have already published articles and worked with reviewing systems. In the future, we will mainly focus on following up methods to see whether reviwers' comments have affected the manuscript in order to achieve a closer integration of the authoring and the reviewing process.
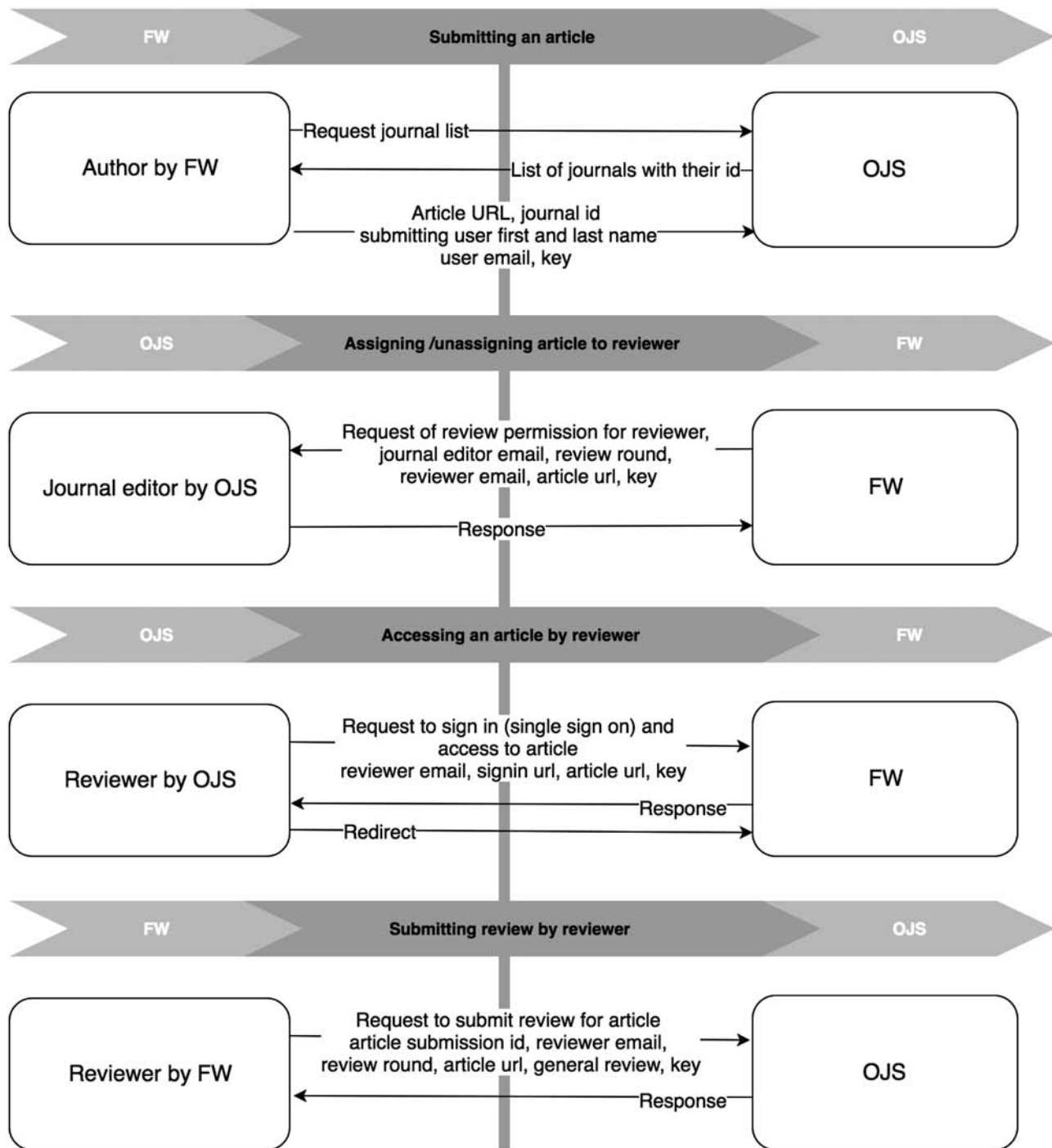
---

19 https://www.fiduswriter.org/ojs-integration/.

Figure 2: Interaction between Fidus Writer (FW) and OJS.

References

Berjon, Robin, ed. (2016). HTML Editing Task Force Charter. http://w3c.github.io/editing/tf-charter.html.

Bornmann, Lutz (2011). Scientific peer review. Annual Review of Information Science and Technology. http://doi.wiley.com/10.1002/aris.2011.1440450112.

Bornmann, Lutz, and Hans-Dieter Daniel (2010). The manuscript reviewing process: Empirical research on review requests, review sequences, and decision rules in peer review. Library & Information Science Research. http://linkinghub.elsevier.com/retrieve/pii/S0740818809001418.

Mayr, Philipp, and Christoph Lange (2016). The Opening Scholarly Communication in Social Sciences project OSCOSS. http://arxiv.org/abs/1611.04760.

Murakami, Shinyu, and Johannes Wilm (2015). Web Browser Based CSS Typesetting Engine – How Browser Based Typesetting Systems Can Be Made to Work Also for Printed Media. In: XML London 2015 Proceedings. Charles Foster, ed.

Perkel, Jeffrey M. (2014). Scientific writing: the online cooperative. Nature. http://www.nature.com/doifinder/10.1038/514127a.

Smecher, Alec (2008). The future of the electronic journal. Neuro-Quantology. http://www.neuroquantology.com/index.php/journal/article/view/150.

Sompel, Herbert Van de, Sandy Payette, John Erickson, Carl Lagoze, and Simeon Warner (2004). Rethinking Scholarly Communication. D-Lib Magazine. http://www.dlib.org/dlib/september04/vandesompel/09vandesompel.html.

Whitehead, E. James (2005). Collaborative Authoring on the Web: Introducing WebDAV. Bulletin of the American Society for Information Science and Technology. http://doi.wiley.com/10.1002/bult.107.

Wilm, Johannes (2015). Does Digital Publishing Need Standards? A History of Text File Standards and Outlook into the Future. Simon Worthington, ed. Dossier: Standards in Digital Publishing – Practitioners' Viewpoints. https://research.consortium.io/docs/publishing_standards_dossier/publishing_standards_dossier.html.

Wilm, Johannes, and Daniel Frebel (2015). Real-World Challenges to Collaborative Text Creation. In: Proceedings of the 2Nd International Workshop on (Document) Changes: Modeling, Detection, Storage and Visualization P. 8:1–8:4. DChanges' 14. Fort Collins, CO, USA: ACM. http://doi.acm.org/10.1145/2723147.2723154.

Dr. Johannes Wilm
GESIS–Leibniz Institute for the Social Sciences
Unter Sachsenhausen 6-8
50667 Köln
mail@johanneswilm.org

Dr. Johannes Wilm ist wissenschaftlicher Mitarbeiter im DFG-Projekt OSCOSS bei GESIS – Leibniz-Institut für Sozialwissenschaften.

Dr. Philipp Mayr
GESIS–Leibniz Institute for the Social Sciences
Unter Sachsenhausen 6-8
50667 Köln
philipp.mayr@gesis.org

Dr. Philipp Mayr ist Teamleiter und stellvertretender Abteilungsleiter bei GESIS – Leibniz-Institut für Sozialwissenschaften.

Afshin Sadeghi
University of Bonn
Römerstraße 164
53117 Bonn
sadeghi@cs.uni-bonn.de

Afshin Sadeghi ist wissenschaftlicher Mitarbeiter im DFG-Projekt OSCOSS an der Universität Bonn.

Dr. Christoph Lange
University of Bonn und Fraunhofer IAIS
Schloss Birlinghoven
53757 Sankt Augustin
langec@cs.uni-bonn.de

Dr. Christoph Lange ist Postdoc an der Universität Bonn und am Fraunhofer-Institut für Intelligente Analyse- und Informationssysteme IAIS.