Analysis of the requirements for the Open Source infrastructure of the Open Research Europe Publishing Platform

Independent
Expert
Report

Research and Innovation

Analysis of the requirements for the Open Source infrastructure of the Open Research Europe Publishing Platform

European Commission
Directorate-General for Research and Innovation
Directorate A ERA & Innovation
Unit Open Science
Contact Victoria Tsoukala
Email victoria.tsoukala@ec.europa.eu
RTD-PUBLICATIONS@ec.europa.eu
European Commission
B-1049 Brussels

Manuscript completed in April 2023

PDF	ISBN 978-92-68-05646-2	doi:10.2777/13928	KI-03-23-367-EN-N

Luxembourg: Publications Office of the European Union, 2023

EUROPEAN COMMISSION

Analysis of the requirements for the Open Source infrastructure of the Open Research Europe Publishing Platform

Author:
Dimitrios Kouis

Research conducted between December 2022 and April 2023 for the European Commission, Directorate-General for Research and Innovation (DG RTD), Open Science Unit.

Dimitrios Kouis is Associate Professor at the Department of Archival, Library and Information Studies at the University of West Attica, Athens, Greece.

2023 Directorate-General for Research and Innovation

Table of contents

EXECUTIVE SUMMARY 5

I. Introduction 5

II. Analysis and Objectives 5

III. Methodology 5

IV. Results 6

IV.1 The landscape of existing state-of-the-art Open Source Software (OSS) for scientific publishing 6

VI.2 ORE publishing workflows 7

VI.2.1 Operating models 7

VI.2.2 ORE content organisation model 7

VI.2.3 User roles and their functions 8

VI.2.4 Pre-publication check and review workflows 8

VI.2.5 Functionality analysis and requirements for the new ORE publishing infrastructure 8

VI.2.6 Recommendations – Final Conclusions 9

CHAPTER 1. INTRODUCTION AND METHODOLOGY 10

1.1 Background 10

1.2 Scope and objectives 11

1.3 Methodology and tools 11

1.4 Limitations 12

1.5 Acknowledgements 13

CHAPTER 2. THE LANDSCAPE OF EXISTING STATE-OF-THE-ART OPEN-SOURCE SOLUTIONS FOR SCIENTIFIC PUBLISHING 14

2.1 Introduction – Aim 14

2.2 Evaluation framework 14

2.3 Solutions under investigation 20

2.3.1 OJS (PKP Solution) 20

2.3.2 PubPub 22

2.3.3 Kotahi 24

2.3.4 Janeway 26

2.3.5 Lodel 28

2.4 Summary – Conclusions 31

CHAPTER 3. THE ORE WORKFLOWS 33

3.1 Introduction – Aim 33

3.2 ORE content organisation model 34

3.3 User roles and their functions 35

3.4 Major workflows 37

3.4.1 Pre-publication check / Preprint workflow 39

3.4.2 Review workflow 40

3.4.3 Other ORE Workflows 41

3.5 Summary – Conclusions 41

CHAPTER 4. FUNCTIONALITY ANALYSIS AND REQUIREMENTS FOR THE NEW ORE PUBLISHING INFRASTRUCTURE 43

4.1 Introduction – Aim 43

4.2 Introduction – Aim 45

4.3 ORE Platform generic requirements 46

4.3.1 License, community and development requirements 46

4.3.2 Performance, availability, support and maintenance requirements 46

4.3.3 System Architecture 47

4.3.4 Security and data protection requirements 47

4.3.5 Data migration 48

4.3.6 EOSC compliance 49

4.3.7 GDPR compliance 50

4.4 ORE Platform publishing requirements 50

4.4.1 Organisational structure 51

4.4.2 User management - User roles 51

4.4.3 Scoping per Subject Area and User Role 51

4.4.4 Notification and communications – User dashboard – SDI Service 52

4.4.5 Integrations with third-party services - APIs – Data and Metadata standards 52

4.4.6 Import – Export 53

4.4.7 Administration features - Reporting 53

4.4.8 Publication activities 54
4.4.8.1 Submission 54
4.4.8.2 Pre-publication check 54
4.4.8.3 Review - Reviewer Management 54
4.4.8.4 Production - Manuscript preparation 55
4.4.8.5 Publishing 55
4.4.8.6 Presentation – Web interfaces 55
4.4.8.7 Commenting 56
4.4.9 Metrics 56
4.5 Summary – Conclusions 57
CHAPTER 5. RECOMMENDATIONS – FINAL CONCLUSIONS 59
APPENDIX I. ORE WORKFLOWS IN DETAIL 62
Pre-publication check workflow 62
Review workflow 65
Simplified workflow – Use of external service providers 69
Manuscript preparation workflow 72
ACRONYMS 74
REFERENCES 76

EXECUTIVE SUMMARY

I. Introduction

Open Research Europe (ORE) is the peer-reviewed open-access publishing platform of the European Commission. It follows the post-publication peer review model, aiming to promote scientific transparency and reuse. Currently, ORE is a service for the beneficiaries of Horizon 2020 and Horizon Europe, but the Commission plans to extend its use to all researchers as of 2026 in collaboration with other funders. Both articles and reviews are open-access and licensed under CC BY licenses (Creative Commons Attribution type). The platform adheres to strict scientific policies and guidelines that authors must follow and is supported by an international scientific advisory board. ORE launched in March 2021 and is currently supported through public procurement, currently through the service provider F1000 Research Ltd.

II. Analysis and Objectives

The present analysis was commissioned to help the European Commission identify and map in detail the workflows and functionalities that ORE should adopt and implement. Additionally, to analyse existing open-source software for scientific publishing.

The objectives of the analysis were the following:

- To analyse existing open-source software solutions with respect to their functional, operational and software perspectives.
- To investigate and document the necessary workflows and functionalities for the new ORE infrastructure.

Finally, the Author was asked to provide practical recommendations on the feasibility of using or combining existing solutions, keeping in mind the Commission's goal of developing an open-source software that will be adopted and enhanced by open-source communities, thereby increasing its sustainability.

III. Methodology

The objectives of this analysis were accomplished through the following tasks:

- **Task 1 - The landscape of existing state-of-the-art open-source solutions for scientific publishing:** A comprehensive examination of cutting-edge, open-source solutions for scientific publishing, such as OJS, PubPub, Kotahi, Janeway, and Lodel, focusing on their functional, operational, and software aspects.
- **Task 2 - ORE publishing workflows:** The publishing workflows were identified based on ORE’s current post-publication open-peer review model and then mapped in detail.

- **Task 3 - Functionality analysis and requirements for the new ORE publishing infrastructure:** A detailed list of features and their technical requirements for the new open-source infrastructure of ORE was provided.
- **Task 4 - Recommendations and Final Conclusions:** Recommendations on the potential for utilising, customising, enhancing, and/or merging existing state-of-the-art open-source solutions to achieve a fully functional infrastructure for ORE were provided.

IV. Results

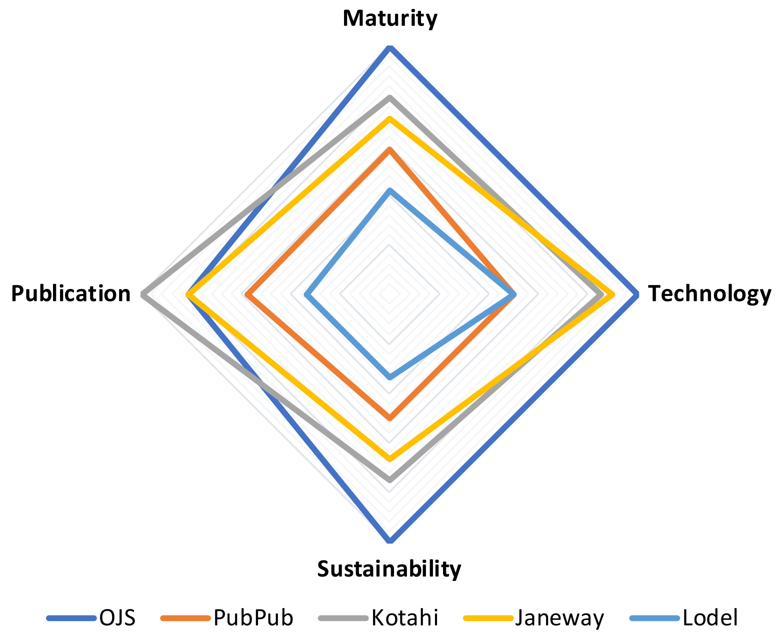
IV.1 The landscape of existing state-of-the-art Open Source Software (OSS) for scientific publishing

An evaluation framework has been developed consisting of more than **31 assessment** elements categorised into the following topics: **Technology (T), Sustainability (S), Maturity (M) and Publishing (P)**. An evaluation was performed for each system separately, and the rating was qualitative.

The most interesting findings are the following:

- None of the platforms examined can immediately support all of the ORE-specific needs and all require extensive development effort to be adapted.
- Three existing OSS for scientific publishing meet numerous ORE requirements, namely PKP OJS, Janeway and Kotahi.
- The advantage of OJS with reference to maturity and sustainability was anticipated, mainly due to the high penetration in the open-access publishing community and the many external vendors (software companies, universities etc.) involved in the development and support.
- If one of the existing solutions is chosen, the involvement of the relevant OSS community in the process may have an important/determining role. Preferably, a group of developers belonging to the OSS's core community may provide the development services for ORE's new platform.

The following diagram visually represents how the platforms under examination met the selected evaluation criteria.



VI.2 ORE publishing workflows

VI.2.1 Operating models

The functionality of the new open-source publishing platform must support the following ORE operating models:

- **Scenario 1** - ORE performs one or more of the publication functions (e.g., peer review) with the help of an external contractor using the new OSS platform or with staff hired for this purpose.
- **Scenario 2 -** an external contractor provides their services through their information systems and only updates the new ORE platform with the results/outcomes.

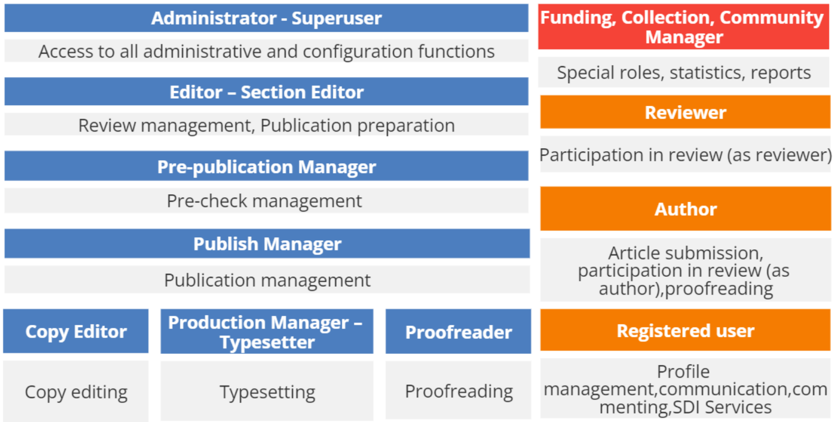
VI.2.2 ORE content organisation model

ORE is organised on the basis of the scientific subject area as the top-level entity and not on the basis of the scientific journal.

Articles should be classified and categorised using at least a two-level hierarchy (meaning that the main scientific topics are divided into sub-topics), the year of publication, unique identifiers, author names and other relevant information. The organization model requires a robust search environment with all the fields mentioned as filters and faceted navigation.

VI.2.3 User roles and their functions

The following figure presents the user roles and the available functions that the new ORE platform should provide.



VI.2.4 Pre-publication check and review workflows

The pre-publication check workflow is a new element vis-á-vis traditional publication flows. It ensures that articles submitted for publication meet ORE's policies and ethical guidelines. The outcome of the pre-publication check is the article's publication, with the peer review still pending. These articles will be referred to as preprints for the needs of the present analysis, although the extensive pre-publication checks add significantly to their value and quality. In addition, the ORE review workflow differs substantially from traditional review workflows that apply to single, double-blind, or Open Peer Review models. Specifically, the ORE workflow introduces the pre-publication check in addition to the main review phase. Both pre-publication check and review phases result in at least three publication actions: the manuscript publication after the pre-publication checks, the publication of the reviewers' comments and decision, the publication of the updated version of the manuscript and the authors' responses to the reviewers' comments and optionally the new comments from the reviewers.

Both workflows (pre-publication check and review) enhance transparency throughout the process and reduce the time for publishing the pre-print and the final versions of the research articles.

VI.2.5 Functionality analysis and requirements for the new ORE publishing infrastructure

The analysis of functionality and requirements offered insights on more than 20 technical elements, grouped into three main categories: (a) hosting infrastructure requirements, (b) general ORE platform requirements, and (c) publishing-specific requirements.

The most important findings are the following:

- The following aspects will necessitate structural modifications in existing open-source publication platforms: (a) The data organization model, (b) support for the pre-publication/preprint check workflow, (c) new forms for the submission of articles by authors and (d) support for the review workflow.
- In addition, content display and search will demand substantial transformation and a considerable investment in development. It is possible that the ORE public web portal may be implemented with software that is distinct from the one responsible for ORE's core publishing functions.

VI.2.6 Recommendations – Final Conclusions

The principal recommendations are summarised as follows:

- Leverage existing OSS instead of initiating the development of a new platform from the ground up. By building on pre-existing solutions, the process can be accelerated, development costs reduced, and the substantial risks commonly associated with large-scale software projects minimised.
- Ensure that the changes introduced by the ORE OSS are integrated into the core code of the selected open-source platform or as plug-ins/add-ons. Collaborate with the community, even if it necessitates modifying certain aspects of ORE.
- Determining which workflows the platform should support and which ones should be reserved for future development is essential for the initial development phase.
- Reevaluate whether the ORE content organisation model (and other workflows) remains valid – any modifications should be made before the ORE OSS platform is developed.
- Be prepared to "invest" in the community of the OSS on which the future ORE infrastructure will be based.
- Having a single entity serve as the provider for the development services of the new ORE platform software, the hosting infrastructure, and ongoing support and maintenance is advisable.

The final conclusions can be summarised as follows:

- Open-source software, particularly when tailored to needs and demands that may diverge from the strategy of the community that supports it, may incur substantial costs, which can be comparable to those associated with proprietary solutions.
- Choosing an existing OSS to adapt it to meet specific requirements, apart from the apparent benefits, also entails risks. For example, the required alterations may substantially restructure the software's code, thus creating a larger-than-expected development workload.
- Existing OSS partially meets the required functions to support the ORE scientific publishing workflow. Based on the findings of the present analysis, three existing OSS for scientific publishing meet numerous ORE requirements, namely: PKP OJS, Janeway and Kotahi.

CHAPTER 1. INTRODUCTION AND METHODOLOGY

1.1 Background

Open Research Europe (ORE) is the open-access peer-reviewed publishing platform offered by the Commission as an optional service to Horizon 2020 and Horizon Europe beneficiaries at no cost to them.

The platform enables researchers to publish in open access without spending from their research budgets and to comply with their open access obligations.

ORE launched in March 2021. Currently, it is supported through a four-year framework contract (until March 2024) with the publisher and publishing service provider F1000 Research Ltd., part of the Taylor & Francis publishing companies. The contractor provides their existing publishing platform customised for the needs of the Commission, as well as publishing, editorial and communication services.

ORE follows an innovative and transparent open-access publishing model for articles based on open peer review after publication (the so-called post-publication peer review). It supports reuse, as articles and reviews are open access and licensed under CC BY (Creative Commons Attribution type). The platform has robust scientific policies and guidelines that authors must follow and is steered by an international scientific advisory board.

Encouraged by Council Conclusions in 2022 and 2023, the Commission envisions gradually expanding ORE from a publication platform for EC beneficiaries only into a not-for-profit pan-European publishing platform.[1] This responsibility (governance and financing) could be shared with Member States, other funders, and possibly research organizations in a blended funding model that will not be based on article processing charges (APCs). Furthermore, ORE may become a service available through the European Open Science Cloud, and its content will be findable and accessible through it as well (European Commission. Directorate General for Research and Innovation., 2022).

Technological autonomy is essential in such a long-term vision: to avoid lock-ins, align to the Commission Open-Source Strategy 2020-2023 and enable cost-efficiency, the Commission will support the development of its publishing open-source infrastructure for ORE[2]. In such a context, the Commission requested the present analysis to advise on the open-source infrastructure underpinning ORE as a collective endeavour starting in 2026.

[1] The June 2022 Council Conclusions on '**Research assessment and implementation of open science**'and the forthcoming May 2023 Council Conclusions on **'High-quality, transparent, open, trustworthy and equitable scholarly publishing'** (final version 23 May 2023)**.**

[2] This procurement has been announced in the 2023-2024 Horizon Europe Work programme Horizontal Expenditure, item 9, p.21-22.

1.2 Scope and objectives

The present analysis has been commissioned to assist the European Commission in defining an efficient workflow and functionalities of the infrastructure basis of ORE in the future and to analyse existing OSS for scientific publishing.

More specifically, the objectives of the present analysis can be summarised in the following:

- Evaluate existing open-source software solutions for scientific publishing and assess whether they could be the base for developing ORE's new publishing platform.
- Describe a recommended publishing workflow that the forthcoming open-source infrastructure, intended for ORE, should satisfy, considering the platform's current post-publication open-peer review workflow, which the Commission wishes to continue to follow.
- Provide a list of functionalities the new open-source infrastructure for ORE must possess to accommodate the publishing process and associated tasks (such as submissions, publication, peer-review, preservation, communication among various roles, etc.).
- Make realistic recommendations on the possibility of using/combining existing solutions.

1.3 Methodology and tools

Four tasks were completed as part of this analysis. The tasks and their most important results are presented next.

Task 1: The landscape of existing state-of-the-art open-source solutions for scientific publishing: A thorough analysis of state-of-the-art open-source solutions for scientific publishing, such as OJS, PubPub, Kotahi[3], Janeway and Lodel, with respect to their functional, operational and software perspectives was performed – (**see Chapter 2**)

- An evaluation framework that covers aspects such as Technology (T), Sustainability (S), Maturity (M) and Publishing (P) characteristics was created.
- Each of the selected open-source software solutions was evaluated in a qualitative way against all aspects of the evaluation framework.

[3] eLife has recently stated that it will not participate in the development of Libero anymore. In contrast, the new integrated publishing platform, known as "Kotahi", which the Coko Community is developing, is expected to replace the Libero publishing suite (Reviewer, Publisher, and Producer). For this reason, evaluating the "Kotahi" application for this study was deemed necessary. See the eLife announcement here and here.

Task 2: ORE publication workflows and roles: Considering the platform's current post-publication open-peer review workflow, the essential publishing workflows that the new open-source infrastructure for ORE should fulfil were defined – (**see Chapter 3**).

- The “new“ workflows that ORE uses for the pre-publication checks and post-publication peer review were thoroughly analysed and documented.
- Alternative scenarios for supporting external publishing service providers are also presented.

Task 3: Functionality analysis and requirements: An extensive list of functionalities that the new open-source infrastructure of ORE should have to fulfil the current publishing workflow and relevant activities was provided – (**see Chapter 4**).

- A comprehensive list of functionalities and requirements concerning the hosting infrastructure, the publishing platform's general characteristics, and the features related to the publishing activities was provided.

Task 4: Recommendations – Final Conclusions: Recommendations were made regarding the possibility of using, customizing, improving and/or combining existing state-of-the-art open-source solutions to obtain a fully functional infrastructure for ORE in the future – (**see Chapter 5**).

Recommendations were provided regarding the feasibility of transitioning to a new open-source information system for the needs of ORE and the critical issues that may lie ahead. Furthermore, a series of conclusions are provided to assist the entire process.

The following tools were used to collect, analyse and produce the final recommendations:

- Literature review (for the evaluation framework of open-source publishing platforms).
- Desk research on open-source platforms for scientific publishing.
- Online meetings with publishing platforms representatives (Janeway, Kotahi, F1000).

Consultation with DG RTD staff members (three virtual meetings, one physical meeting).

1.4 Limitations

The present analysis is affected by the following limitations:

- The report was limited to only five OSS, which support publishing activities. Although the selected platforms are considered the most comprehensive and widely used for supporting scientific publications, many other tools could be examined as to whether they can be utilised for ORE-specific needs, although they are not specific to scientific publishing and its workflows (W. Maxwell et al., 2019).
- The evaluation framework comprises specific criteria and characteristics derived from the literature review. Another approach in forming the evaluation framework could

lead to different results. In addition, it was impossible to fully access the features and functions of the systems under assessment due to technical and time constraints.

- Additional meetings with more and different publishing software representatives in a more extended time period would result in an improved exchange of opinions and ideas.

Describing the most essential workflows of ORE is a creative activity and the basis for understanding the new platform functionality. Nevertheless, software implementation is the actual process of realising the design as a program. In this sense, the information provided in the present report should not be considered software design but an indicative first level of operational workflow analysis.

1.5 Acknowledgements

This work was commissioned by the European Commission’s Directorate-General for Research and Innovation (DG RTD). I gratefully acknowledge the guidance and support received from Victoria Tsoukala, Jean-François Dechamp, Dejan Dvorsek and Gabriella Leo on behalf of DG RTD.

The representatives of the publishing platforms shared their insights during our online meetings (Kotahi team, Janeway team and F1000 team), for which I am also grateful.

CHAPTER 2. THE LANDSCAPE OF EXISTING STATE-OF-THE-ART OPEN-SOURCE SOLUTIONS FOR SCIENTIFIC PUBLISHING

2.1 Introduction – Aim

This chapter presents several selected open-source platforms for managing scientific journal publications that could form the basis for developing the infrastructure that will underpin ORE in the future. Their main feature is their availability as open-source software (OSS) - this means that end users can access the code, install the platform on their infrastructure and customise its functionality and/or develop new functions and features. On the other hand, there is commercial software developed by Information and Technology (IT) companies and distributed with special usage licenses and cost payment models [Commercial Off-The-Shelf (COTS)].

The analysis of the OSS hosting platforms for scientific publications will be carried out by using an evaluation framework developed for the needs of the study, the details of which are described in the next section. The parameters/components of the evaluation framework focus on issues such as governance, software maturity, the extent of the community that supports/deals with it, security, reliability, the number of end-users/installations, etc.

Based on the analysis and data obtained, readers of this study can understand the extent to which each solution meets the requirements of each assessment parameter. The results of this qualitative analysis are useful, especially for institutions and organisations wishing to support scholarly publishing activities on one of the platforms studied, especially regarding development, support and future maintenance/sustainability.

At this point, it must be emphasised that all the systems studied are technologically mature solutions, which means that all of them are established software solutions used by many organisations. Moreover, it should be taken into account that the development plan and the functionality of each platform address the needs of the specific communities responsible for them, the entities that financed them, and their end-users. The data in this section provide information about the suitability of each platform for different scholarly publishing contexts, as well as the strengths and weaknesses of each solution in meeting the specific needs of diverse user groups.

2.2 Evaluation framework

Many studies in the relevant literature attempt to provide evaluation frameworks for software that belongs to the open-source category. Specifically, there are several assessment models such as C-OSMM (Capgemini Open Source Maturity Model), N-OSMM (Navica Open Source Maturity Model), QSOS (Methodology of Qualification and Selection of Open Source Software), Open BRR (Open Business Readiness Rating), among others (Adewumi et al., 2016; Umm-e-Laila et al., 2017).

The reasons for developing evaluation frameworks that focus on OSS are their different characteristics compared to commercial software. Specifically, OSS is free of charge, whereas COTS are third-party company products or components. In addition, OSS components are freely available in public repositories, sometimes making the selection and installation process more complicated and troublesome. Another difference is that OSS users are aware of the full functionality of the components, which is not possible in COTS as the code is not known / accessible. However, the main difference that led to the creation of particular frameworks for evaluating OSS is that the responsibility for the development lies with a community of volunteers who decide on the strategic plan and the licensing model. In addition, while in OSS, the skills of its members determine the quality of the code, in the COTS case, a company is responsible for the development, controls the strategic plan, makes significant decisions, has trained personnel, and usually applies strict quality assurance standards.

Therefore, the key evaluation dimensions of free software are its stability, code quality, project governance model, and market appeal. In a recent research study by Linåker et al. (2022), the concept of "OSS Health" was defined as follows: *OSS health is considered the project's capability to stay viable and maintained over time without interruption or weakening*. The same study identified 155 related articles to the topic through an extensive literature review. The study's authors categorised them into three broader thematic areas and 15 more specific thematic sub-areas with 107 characteristics that affect/define the concept of OSS Health.

Some thematic areas are related to software, project community/actors, and orchestration/governance issues. Similarly, other thematic areas are connected to funding sources, popularity/acceptance by the community, community development activities, licensing model, governance model, etc.

The findings in the research of Khatri and Singh (2016) were similar. Specifically, they identified evaluation areas such as age, market penetration, implementation and support, performance, reliability, etc.

Based on the above discussion, the multitude of characteristics that have been identified in the various studies, the type of software under examination (in our case, OS publishing platforms), and the availability/accessibility of information were the main factors that influenced the development of our evaluation framework. The following table (**Table 1**) presents the characteristics that have been selected as the most important ones for this assessment, along with their checkpoints necessary for a more profound understanding of a platform.

The evaluation concerns each system separately, and the rating will be qualitative and based on the experience of the author of this study. The decision to perform a non-quantitative evaluation was driven by the fact that it is impossible to determine numerical data for specific characteristics, as well as the fact that a qualitative analysis provides more nuanced insights into each of the elements analysed, which can be useful for organisations interested to learn more about these OSS.

The evaluation characteristics are further categorised into the following three topics: Technology (T), Sustainability (S) and Maturity (M).

Table 1. Publishing Platform Evaluation Framework (excluding functional modules capabilities)

Evaluation framework characteristics	Checkpoints
Age of the project (M)	▪ Year of the first stable version ▪ Frequency of major releases/updates
Community structure Governance model Finance (S)	▪ Existence of a leading organisation ▪ Decentralised or centralised governance ▪ Number of funding sources, financial stability and support ▪ Number and type of organisations involved
Model of licensing Level of platform or technology independence Integration of third-party services Interoperability (M)	▪ Open-source compatible license ▪ No restrictions/dependence on proprietary software ▪ Special requirements for installation/deployment ▪ Dependencies from / use of external services ▪ Compliance with standards
Development Community Future development Roadmap Documentation (M)	▪ Development activity (number of contributors, commits, forks etc.) ▪ Source code repositories - Tools used for reporting, bug solving etc. ▪ Programming language, DBs, frameworks etc. ▪ Infrastructure technologies ▪ Documentation completeness (development, end-user, installation/deployment) ▪ Documentation accuracy
Technical quality Modularity Collaboration with other products (T)	▪ Architecture – Deployment model (single installation, multiple instances, internal organisation) ▪ Component quality ▪ Plugin support/microservice support ▪ Compliance with interoperability standards ▪ Data/metadata standards, APIs
Performance Scalability Live installations (T)	▪ Performance metrics: traffic/response time/number of users ▪ Load scalability/balance mechanisms ▪ Number of live installations (size)
Security Data protection (T)	▪ Number of security vulnerabilities reported ▪ Vulnerability persistence - Platform and infrastructure security measures/tests (monitoring, firewalls, antivirus, servers, network – cloud infrastructure, application etc.) ▪ Backup mechanism – disaster recovery policies ▪ GDPR Compliance

Reliability / Availability (T)	▪ Correctness of functionality ▪ Availability and full functionality of the platform (e.g., >99.9% of the time)
Implementation and Support (S)	▪ Implementation and support* from the community ▪ Implementation and support from external contract vendors ▪ Large-scale implementation/support examples (* *meaning development, customisation, hosting, training, support services*)
Market penetration User Community (M)	▪ Total number of installations – estimation of end-users ▪ Existence of User-Communities ▪ Project popularity / Geographical diversity

In addition to the evaluation framework presented in the table above, the essential functional components of a platform supporting scientific publishing were also identified (see the following figure) and included as a second layer in the evaluation process. The full description of the functions and capabilities that each module should provide in the context of ORE is presented next. Through this extra set of aspects for evaluation (publishing characteristics), the study aims to identify whether each platform efficiently supports these specific functions in their "traditional operation" or not. This assessment suggests the "readiness" of each platform to support the requirements of ORE with considerable development effort or relatively easily.

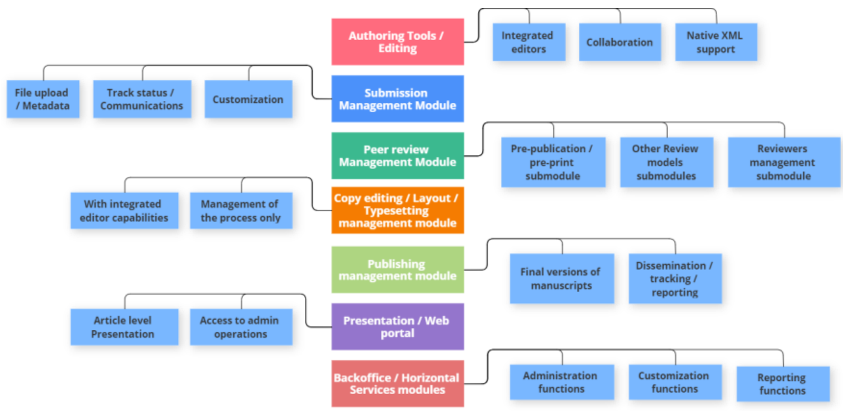


Figure 1. Main functional components of an end-to-end publishing platform

Authoring and editing tools in an end-to-end publishing platform provide a range of functions beyond uploading a manuscript in the form of Microsoft Word or Latex by authors. More specifically, they offer the following essential functions:

- The integrated WYSIWYG[4] editor enables authors to create and edit content without having to master HTML or other programming languages. It supports advanced

[4] WYSIWYG: "what you see is what you get".

formatting options (headings, fonts, colours and alignment), images, media, tables and mathematical equations, references management, etc.

- Authors can collaborate during manuscript preparation, review, and publication (e.g., adjusting content in response to reviewers' comments, typesetting and proofreading corrections, etc.).
- The ability to save versions and track changes to updated content.
- Native support for JATS[5]-based authoring and editing tools enables the automation of creating HTML and PDF versions of the manuscript through dedicated rendering engines.

Integrating authoring and editing tools into the publishing platform could minimise the need to upload and download different versions of the manuscript and significantly improve the speed of publication. *Nevertheless, despite the mature contexts of XML (such as JATS) in publishing, it seems to be a largely undisputed fact that "authors will write in Word"* (W Maxwell et al., 2019, p. 17).

In the next step, the **Submission Module** allows authors to upload the content (manuscript and additional files) and its metadata (author info, manuscript title, thematic categories, keywords, etc.). The **Submission Module** also allows authors to track the status of their submissions and communicate with editors, reviewers, proofreaders, etc. The information and details authors must fill in and the customisation capabilities offered to determine the degree of flexibility of the platform.

The **Peer Review Module** (often part of the Submission Module) is responsible for the peer review process for the submitted content. It includes the implementation of workflows such as: Assigning reviewers, managing reviewer qualifications (see Reviewer Management sub-module in **Figure 1**), managing review results, decision-making, overseeing revisions, tracking reviews/status, communication support, supporting quality checks (e.g., plagiarism checks) and more. The most important aspect of the peer review module is the ability to support different review models (e.g., single or double-blind peer review, Open Peer Review - OPR model or even Publish-Review-Curate-PRC[6] and other preprints models). The number of review models supported, their characteristics, and the flexibility offered for adapting them define the platform's maturity. It is important to note that no OSS publishing platform supports the ORE review model (as analysed in the next chapter).

The next module manages the workflow responsible for preparing the manuscript for publication. In particular, it administers the process of editing, typesetting/layout and proofreading. As explained above, it is necessary to support the uploading and downloading of different versions of the manuscript by the various actors (editors, typesetters, proofreaders and authors) if the authoring/editing tools are not part of the publishing platform. If an external service prepares the manuscript for publication, the complexity and number of functions are low and limited accordingly (see section **Other ORE Workflows**).

The **Publication Management Module** offers features such as uploading final manuscript content in various formats, metadata management tools for organising and categorising

---

[5] Standardized Markup for Journal Articles: [Journal Article Tag Suite (JATS).]()
[6] See Hyde et al. (2022).

content, licencing, access and permission issues, etc. It also supports integration with external services for content distribution, assignment of unique identifiers, etc. Finally, it manages communication and dissemination channels and tracking and reporting tools and services to monitor publication performance through various metrics and indicators.

The presentation layer of the publication platform is implemented through a **Web Portal**. Apart from the access to the management screens/functions (back-office operations), the capabilities the platform should offer regarding access and navigation to the content are more important. Information retrieval capabilities should include searching the full text and the metadata fields, thematic navigation, etc. Additionally, it should support an exclude/include or data filtering mechanism through faceted navigation/search[7]. Some commercial platforms that provide access to scientific content (e.g., Scopus, WoS) offer advanced search capabilities through constructing query strings, using proximity operators, and many more. Furthermore, at the article level view, information such as the following should be displayed:

- The manuscript of the article in various formats (at least HTML and pdf), including its metadata (title, authors, keywords, abstract, thematic topics)
- Review status and content (especially if the Open Peer Review model applies)
- Links to older / newer versions of the article
- Access to dissemination tools (share/export/download/cite/monitor)
- Article level metrics/indicators [e.g., views, downloads, visitors' characteristics, comments, cites from different citation indexes, the activity of datasets/repositories, social media (Twitter, Facebook) etc.]

Finally, although the Backoffice and other horizontal services of a publishing platform are not part of the publication process, they are considered essential for the publishers and determine efficiency. The Backoffice and other horizontal services may include administration functions (e.g., user and roles management, customisation of forms and workflows, communication configuration etc.), integration functions (APIs, third parties services etc.), reporting functions including export/import capabilities and many more.

This section presented a framework for the evaluation of the OS platforms in two levels, namely a first level regarding the Technology (T), Sustainability (S) and Maturity (M) aspects, and a second level, which concerns the main functional components of an end-to-end publishing platform Publishing (P) aspects. The following section presents the results of the assessment.

[7] Faceted search, also known as faceted navigation or guided navigation, is a search technique that allows users to refine and narrow down their search results by applying multiple filters or criteria. These filters, or facets, are typically organized into categories representing various aspects of the items under search.

2.3 Solutions under investigation

2.3.1 OJS (PKP Solution)

The world's most widely used OS journal management and publication system is the Open Journal Systems (OJS), with thousands of installations and a vibrant support community.

OJS has been created and maintained by the Public Knowledge Project (PKP - https://pkp.sfu.ca/). The easy process of installing OJS is why most end-users prefer local installation in their computational infrastructures. OJS supports the publishing process from submission to peer review and publication, offering at the same time a wide range of indexing, identification, discoverability, and preservation services.

The following table presents the comments and the results of the framework evaluation application on the OJS.

Table 2. PKP OJS Evaluation Framework Results

Evaluation framework characteristics	Comments
Age of the project	Since 2002, Major new release OJS 3.0, frequent releases/updates.
Community structure **Governance model** **Finance**	PKP is the leading organisation with more than 16 "strategic partners" (e.g., COAR[8], CrossRef, DOAJ[9] and more). Also, there are development partners and more than 100 financial contributors.
Model of licensing **Level of platform or technology independence** **Integration of third-party services** **Interoperability**	PKP OJS (and other PKP platforms) is open-source software under a GPLv2[10] license. There are no restrictions or dependencies from other proprietary software. PKP OJS is easy to install locally with detailed technical documentation. Many external services supported by plugins need a subscription (e.g., DOI services). Both content metadata and APIs for data exchange are compliant with international standards.

[8] COAR: Confederation of Open Access Repositories.
[9] DOAJ: Directory of Open Access Journals.
[10] See more details here.

Development Community Future development Roadmap Documentation	The source code repository of OJS can be found at the following URL: https://github.com/pkp/ojs The number of contributors is over 220, with more than 19.000 commits, 22 branches and 854 forks (OJS 3.X version). PKP OJS uses PHP, Javascript and the Smarty framework for its core software and is usually deployed in a LAMP[11] environment. Complete documentation, with tutorials and community support, is provided on the project website.
Technical quality Modularity Collaboration with other products	The single installation supports multiple journals. Journal is the top organisational unit. Plugin development and microservices are supported. Many add-ons already exist, providing access to external services and operational enhancements. Many APIs and metadata standards are supported.
Performance Scalability Live installations	Limited support for load balancing (internal cache support) – Infrastructure side mechanisms only. More than 25.000 Journals (single or multitenant installations – see Market penetration below).
Security Data protection	Support for security and backup/recovery mechanisms relies only on infrastructure tools. GDPR compliance is implemented only through some recommendations (https://docs.pkp.sfu.ca/gdpr/en/configuration-options).
Reliability / Availability	Support for availability relies only on infrastructure tools.
Implementation and Support	Implementation and support from the community are considered above average. Many commercial/external vendors worldwide offer hosting, implementation, and support services. PKP OJS has been tested in large installations (e.g., SciELO[12]). Also, while relying on OJS as the core of their journal-publishing platform, Ubiquity Press has made customisations for article review and relies on a different system for managing book editorial processes.
Market penetration User Community	The PKP community forum (https://forum.pkp.sfu.ca/) provides the tools for reporting/resolving bugs for all OJS users. The network of OJS end-users is extended with more than 25.000 active journals worldwide. Publishing around 1 million articles in 2021 (more than 6.1 million since 2010) (Huskisson, 2023; Khanna et al., 2022).

[11] LAMP: Linux, Apache, MariaDB / MySQL, PHP.
[12] SciELO: Scientific Electronic Library Online.

Based on the above, it is evident that the OJS platform constitutes a very mature technological solution, with a large community of programmers and external vendors providing support. Additionally, the OJS platform is used by many "small" publishers (e.g. Universities, scientific societies etc.), who manage journals primarily under the open-access model. Therefore, concerning the assessment dimensions of maturity and sustainability, the OJS platform presents excellent results.

Additionally, after the release of version 3 of the OJS software, it is easier to configure and customise all publishing workflows. The performance and reliability of the OJS platform have been verified through many installations supporting a large number of journals simultaneously (single installation – multiple journals).

Finally, the OJS platform satisfies most of the publishing functional area workflows to a large extent, except for the authoring and editing tools (although some efforts are taking place[13]). The OJS platform has many plugins that significantly increase the available features and demonstrate the great interest and experience of many programming teams in their development, which is considered positive regarding the technological dimension of evaluation.

It is a crucial point to what extent the ORE publishing model would be accepted by the community of OJS users and those who make strategic decisions for the platform's further development. If the ORE model ultimately gains acceptance, combined with the fact that many open-access publishers use OJS, the impact and influence on the publishing world will be enormous.

2.3.2 PubPub

The MIT Knowledge Futures Group and the MIT Press created an online authoring and publishing platform known as PubPub. It encourages "an integrated and iterative approach" for the community-based collaborative drafting, reviewing, and publication of scholarly work. Through PubPub, end-users can publish journals, books, and other material categories. PubPub is intended to be centrally hosted; thus, publishing services are offered through the hosting model.

The following table presents the results of the framework evaluation of PubPub.

Table 3. PubPub Evaluation Framework Results

Evaluation framework characteristics	Comments
Age of the project	Since 2018, At least ten development cycles (2018-2021 - https://notes.knowledgefutures.org/pub/pubpub-roadmap/release/29). Currently, PubPub implementation roadmap activities are being recorded here: https://github.com/orgs/pubpub/projects/9

[13] See Fidus Writer + OJS integration.

Community structure Governance model Finance	PubPub implementation is coordinated by the Knowledge Futures Group (KFG), a nonprofit organisation dedicated to building public knowledge infrastructure. KFG is governed by an independent board and supported and driven by Members. The financial stability of the PubPub endeavour is based on the contribution of several institutes and organisations (see https://www.knowledgefutures.org/about). Also, PubPub ensures community-driven implementation and sustainability through KFG Membership programmes and support services to end-users.
Model of licensing Level of platform or technology independence Integration of third-party services Interoperability	PubPub is open-source software under the GPLv2 license. There are no restrictions or dependencies from other proprietary software. PubPub is mainly provided as a hosted service. Connection with external services is limited. Both content metadata and APIs for data exchange are compliant with international standards.
Development Community Future development Roadmap Documentation	The source code repository of PubPub can be found at the following URL: https://github.com/pubpub The number of contributors is 22, with over 7.000 commits, 74 branches and 57 forks. PubPub uses mainly Typescript and Javascript for its core software development. Complete documentation, with tutorials and community support, is provided on the project website, although in a quick way.
Technical quality Modularity Collaboration with other products	Collection and Communities are the top organisational units. The single installation supports multiple communities and collections. No support for plugin development and microservices is provided. Some APIs and metadata standards are supported.
Performance Scalability Live installations	Support for load balancing relies upon infrastructure-side mechanisms. Only https://www.pubpub.org/ is tested with high-load traffic, hosting thousands of communities and end-users.
Security Data protection	Support for security and backup/recovery mechanisms relies on infrastructure tools. GDPR compliance is not mentioned.
Reliability / Availability	Support for availability relies on infrastructure tools.
Implementation and Support	Implementation and support from the community are considered moderate. Development and support services are provided through the KFG community services. The PubPub community forum (https://github.com/pubpub/pubpub/discussions) provides the tools for reporting/resolving bugs for all end-users.
Market penetration User Community	While PubPub is mainly provided only through a hosted live installation (https://www.pubpub.org/), many communities have selected it to implement their publishing activities.

On the basis of the information presented above, it is apparent that the PubPub platform is an emerging solution for meeting the needs of small and medium publishing efforts, focusing primarily on cost minimisation. Specifically, it offers a comprehensive solution for authoring, editing, and presenting content online. However, regarding administrative functions related to tasks such as article submission and peer-review management, the level of maturity is still relatively low. This is due to the ongoing development efforts and the focus on content creation and presentation in a modern way via the internet.

The PubPub platform is an ideal solution for publishing activities requiring α quick installation and deployment while managing workflows, such as article submission and evaluation, and other information systems. Consequently, at its present stage of development, Pubpub may not be able to satisfy the requirements of an end-to-end peer-review scholarly publishing process, including the ORE model.

2.3.3 Kotahi

Kotahi is a cutting-edge platform for scholarly publishing that can accommodate various end-to-end publishing workflows and was developed by a consortium of New Zealand universities, academic libraries, and government agencies to promote open access to research articles.

Kotahi offers a variety of workflows, including publishing, peer review, and writing/authoring. Kotahi emphasises assisting the creation and editing of articles within a web browser by including features like track changes, math, photos, captions, grammar checks etc. This idea allows Kotahi to provide a single source publishing workflow where all parties can access the same manuscript during content creation (peer review, editing, etc.) and production (copyediting, semantic tagging, etc.), reducing the need to upload and download different versions of the manuscripts.

The following table presents the comments and the results from the framework evaluation application on Kotahi.

Table 4. Kotahi Evaluation Framework Results

Evaluation framework characteristics	Comments
Age of the project	Since 2018. The latest version of Kotahi is 1.5. Kotahi implementation roadmap activities are being recorded here: https://kotahi.community/features/
Community structure **Governance model** **Finance**	Kotahi is a Coko[14] community project and aims to replace the Libero Publishing suite. The Coko team coordinates Kotahi implementation. The financial stability of the Kotahi endeavour is based on the contribution of the Coko Foundation community, which numbers more than 30 foundations, publishers, universities and other categories of institutes.

[14] Coco Foundation.

Model of licensing Level of platform or technology independence Integration of third-party services Interoperability	Kotahi is open-source software under the MIT license. There are no restrictions or dependencies from other proprietary software. Kotahi supports installation on docker. Connection with external services is adequate (e.g., ORCID, Crossref, SSO support etc.). Both content metadata and APIs for data exchange are compliant with international standards.
Development Community Future development Roadmap Documentation	The source code repository of Kotahi can be found at the following URL: https://gitlab.coko.foundation/kotahi/kotahi The number of contributors is 30, with over 4.300 commits and 310 branches. Kotahi uses mainly Javascript for its core software development and is based on projects produced by Coko[15] , such as PubSweet[16], xSweet[17], Wax[18], Paged.js[19] Complete documentation and community support are provided on the project website, although in a quick way.
Technical quality Modularity Collaboration with other products	Preprints, Journals, Books and other types of MicroPubs are the top organisational units. Single installation is currently available. Multitenancy with the support of unified or separated groups of publications is expected to be delivered at the end of 2023. Support for plugin development and microservices is provided. Many APIs and metadata standards are supported.
Performance Scalability Live installations	Support for load balancing relies on docker technology advantages. A rather limited number of installations.
Security Data protection	Support for security and backup/recovery mechanisms relies on docker technology advantages. Kotahi is GDPR compliant (no explicit reference).
Reliability / Availability	Support for availability relies on docker technology advantages.
Implementation and Support	Implementation and support from the community are considered moderate. A few external contract vendors support the platform and publishing services.

[15] See for details here.

[16] PubSweet.

[17] XSweet.

[18] WaX.

[19] Paged.js.

	Kotahi aims to replace end-users of the Libero Publishing Suite. In this sense, many publishers are expected/have to migrate/migrated to Kotahi (e.g., eLife[20], Sciety[21] etc.)
Market penetration User Community	See the previous comment.

The Kotahi platform presents interesting technical characteristics that make it a very reliable solution for implementing large-scale publishing projects. Specifically, it offers an easy and dynamic adaptation of workflows such as article submission and peer review. In addition, Kotahi can support different types of peer review within the same installation. Another strong aspect of the Kotahi solution is the end-to-end manuscript management within the platform (see discussion at the beginning of the section). The Kotahi platform development team also has experience implementing complex and sophisticated peer-review flows, including those supporting the preprint model and is currently experimenting with integrating AI support into various end-user functions. Another important aspect is that the Kotahi platform implements the presentation and content search level with the help of a new software called Flax[22], which is currently under development and promises advanced search features.

Finally, although the Kotahi platform has a small user base, it is mainly used by organisations responsible for large-scale publishing activities (e.g., eLife).

2.3.4 Janeway

Janeway is a digital platform for publishing scholarly articles online, from submission and review to editing, production and final publication.

Janeway is being developed and maintained by the Birkbeck Centre for Technology and Publishing for the Open Library of Humanities (OLH) at Birkbeck, University of London. Janeway is currently used by Iowa State Digital Press, Huddersfield University Press, ID Pro, UCL Press Michigan Publishing, the University of Essex and the University of West London and will soon be in use by California Digital Library to support preprints projects.

Janeway integrates various external services to provide a full-featured OA journal publishing platform.

The following table presents the comments and the results from the framework evaluation application on Janeway.

---

[20] eLife.

[21] Sciety.

[22] See Coko Platforms.

Table 5. Janeway Evaluation Framework Results

Evaluation framework characteristics	Comments
Age of the project	Since 2017. The latest version of Janeway is 1.4.3 Janeway's new implementation activities can be found in the source repository.
Community structure Governance model Finance	The Centre coordinates Janeway implementation for Technology and Publishing at Birkbeck, University of London. The financial stability of the Janeway endeavour is based on donors and clients that use it.
Model of licensing Level of platform or technology independence Integration of third-party services Interoperability	Janeway is open-source software under the GNU Affero General Public License v3.0 license[23]. There are no restrictions or dependencies from other proprietary software. Janeway supports installation on docker. Connection with external services is adequate (e.g., Crossref DOIs, ORCID sign-in etc.). Both content metadata and APIs for data exchange are compliant with international standards.
Development Community Future development Roadmap Documentation	The source code repository of Kotahi can be found at the following URL: https://github.com/BirkbeckCTP/janeway The number of contributors is 23, with over 4.500 commits, 106 branches and 54 forks. Janeway uses Python and utilises the Django framework for its core software development. Full documentation is provided on the project's website. https://janeway.readthedocs.io/en/latest/index.html
Technical quality Modularity Collaboration with other products	Journals and Preprints are the top organisational units. Single installation is currently available with the ability to operate multiple journals simultaneously. Support for plugin development and microservices is provided. Many APIs and metadata standards are supported.
Performance Scalability Live installations	Support for load balancing relies on docker technology advantages. A small number of installations.
Security Data protection	Support for security and backup/recovery mechanisms relies on docker technology advantages. GDPR compliance is not mentioned.

[23] The GNU Affero General Public License is a modified version of the ordinary GNU GPL version 3. It has one added requirement: if you run a modified program on a server and let other users communicate with it there, your server must also allow them to download the source code corresponding to the modified version running there.

Reliability / Availability	Support for availability relies on docker technology advantages.
Implementation and Support	Implementation and support from the community are considered moderate. Ten organisations provide Janeway as a service.
Market penetration User Community	See the previous comment.

The Janeway platform is a technically mature solution that fully supports all functional needs for publishing scientific content, except for the support of end-to-end article authoring and editing. The Janeway platform has similar operating characteristics to the PKP's OJS platform. The number of organisations using the Janeway platform is not extensive. Nevertheless, the majority of these organizations are prominent academic publishers, such as university presses, that handle extensive publishing operations and consider Janeway a reliable and complete solution.

2.3.5 Lodel

Lodel is a Content Management System (CMS) and not a complete platform that supports the end-to-end publishing cycle of scientific journals. Lodel development started in 2000 in France for the needs of the Revues.org platform, now known as OpenEdition.

Lodel was created to handle long, complex texts in a highly structured editorial environment. Lodel produces XML documents and conforms to the norms of the profession, especially W3C standards and other interoperability norms (Dublin Core, RSS, OAI). The evaluation of Lodel was based mainly on web resources and the findings of a report by Lucia Randone (Randone, 2019).

Table 6. Lodel Evaluation Framework Results

Evaluation framework characteristics	Comments
Age of the project	Since 2000, the current version is 1.05. Lodel implementation roadmap activities are being recorded here: https://lodel.hypotheses.org/chronologie
Community structure Governance model Finance	Lodel implementation is coordinated by the OpenEdition Center, which is responsible for its financial stability.
Model of licensing Level of platform or technology independence Integration of third-party services Interoperability	Lodel is open-source software under the GPLv2 license and can be locally installed. Nevertheless, the core publishing functions such as submission, review, etc. are offered through the OpenEdition Journals service, which is not part of the Lodel framework.

Development Community Future development Roadmap Documentation	The source code repository of Lodel can be found at the following URL: https://github.com/OpenEdition/lodel The number of contributors is 23, with over 5.200 commits and 14 branches. Lodel uses mainly PhP and HTML for its core software development. Full documentation, with tutorials and community support, is provided on the OpenEdition website.
Technical quality Modularity Collaboration with other products	Journals and Books are the top organisational units. Some APIs and metadata standards are supported.
Performance Scalability Live installations	Support for load balancing relies upon infrastructure-side mechanisms. A limited number of installations.
Security Data protection	Support for security and backup/recovery mechanisms relies on infrastructure tools. GDPR compliance is not mentioned.
Reliability / Availability	Support for availability relies on infrastructure tools.
Implementation and Support	Implementation and support from the community are provided mainly by OpenEdition Center members.
Market penetration User Community	A few installations exist, mainly in France

The Lodel platform does not support those publishing functions that would allow it to be characterised as a complete solution for implementing the publication of a peer-reviewed scientific journal. The Lodel platform lacks a journal-like content submission mechanism or peer-review management services as part of its core functionality. As mentioned earlier, the platform offers a reliable solution for displaying content in multiple formats, provided that it has followed specific specifications and standards during its creation. As it stands and by itself, it cannot support the complex peer-reviewed publishing activities required by the ORE workflow.

To conclude, the following diagram presents visually how the platforms under examination meet the selected evaluation criteria, as mentioned above and analysed in the previous paragraphs of this chapter. It visually displays each platform's strong and weaker points with respect to the four elements: technology, sustainability, publication, and maturity.

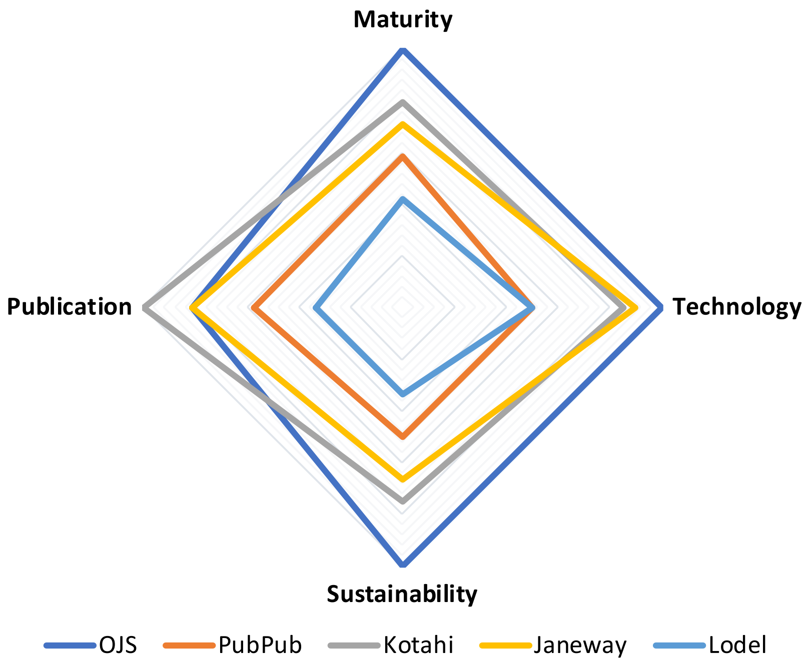


Figure 2. Platforms under examination and evaluation framework main criteria

2.4 Summary – Conclusions

The primary goal of this chapter was to understand and determine the "readiness" of a select number of OS platforms to support the requirements of ORE. This was done through the application of an evaluation framework devised for the needs of the present study to each of these platforms. The data presented on each platform help us understand their technological and functional maturity level. Additionally, determining the strengths and shortcomings of each solution in fulfilling specific requirements/aspects [*meaning technology (T), sustainability (S), maturity (M) and publishing (P) aspects as they were presented in the* ***2.2 Evaluation framework section***] is a way of providing an indicative estimation for the development effort necessary to meet the ORE needs.

The main conclusions drawn from the above analysis are as follows:

- None of the platforms examined can directly support the ORE particular needs. All platforms require extensive development effort to adapt them. Specifically, as is presented in more detail in the next chapter, none of the examined systems supports the ORE publication model in its entirety (pre-publication checks/preprint workflows). Additionally, most platforms use the "scientific journal" as the top-level entity for their internal organisational model (OJS, Janeway, Kotahi). Nevertheless, some support other types of publications, such as preprint repositories (Kotahi) or communities/collections (Pubpub). The above findings indicate that adaptation will require considerable development effort, which may impact the development and end-user communities, primarily regarding the acceptance of the new functions.

- OJS, Kotahi and Janeway have all the main functional components of an end-to-end scientific publishing platform in place [see section **Figure 1**], while Kotahi additionally features the authoring and editing tools. The three systems also fulfil most of the technical aspects presented in **Table 1** (Technology (T)). It can be concluded that they meet many of the ORE requirements in general.

- The maturity and sustainability aspects also play an essential role in choosing the most suitable solution to support publishing activities. The advantage of OJS in maturity and sustainability aspects was anticipated, mainly due to the high penetration in the open-access publishing community and the many external vendors (software companies, universities etc.) involved in the development and support. Of course, the popularity of OJS can also be an inhibitory factor in including new features (e.g., ORE review model) into the source code's main branch due to the complexity of the governance structure and the community's diverse needs. Maintaining the ORE development code as a separate fork can be an acceptable solution, if one of the above solutions is selected. However, it reduces the level of sustainability and impacts the end-user communities to some extent.

- The involvement of the relevant OSS community in the process may be an important/determining aspect of it. Theoretically, since ORE will be based on open-source software, any company or third-party provider could participate in developing the relevant infrastructure. However, the scenario with the greatest overall impact would be that a group of developers who belong to the core community develop the new ORE OSS platform. With the exception of OJS, whose development community includes many external vendors, in the remaining cases, the platforms are supported only by their development communities, whom it is important to involve in the process. This situation has certain advantages, such as that the developers from the software community can make changes relatively quickly because of their familiarity with the software. On the other hand, the limited options for providing development and support services may lead to phenomena similar to the dependence organizations face on proprietary software and, more specifically, the inability to use another provider without substantial switching costs.

CHAPTER 3. THE ORE WORKFLOWS

3.1 Introduction – Aim

This chapter describes a recommended end-to-end publishing workflow that the new open-source infrastructure that will be developed for ORE should fulfil, considering the platform's current post-publication open-peer review workflow, which the Commission wishes to continue to follow.

In this context and before presenting the major workflows, the **ORE content organisation model** to be followed, which affects both the workflows and certain functions of the platform (such as the presentation of data on the web), is also introduced (see section **3.2 ORE content organisation model**).

Additionally, a detailed description of **the roles of the users** that need to be supported by the system and the tasks/duties associated with these roles are presented (see section **3.3 User roles and their functions**).

It is noted that roles, the associated tasks and the content organisation model described in the following sections are indicative and can be adjusted/updated during the ORE new platform implementation stage.

It should also be emphasised that the analysis of the main workflows that the platform ORE must support is carried out by taking into account the following two operational scenarios/business models:

- **Scenario 1** - ORE performs one or more of the publication functions (e.g., peer review) with the help of an external contractor using the new OSS platform or with staff hired for this purpose.
- **Scenario 2 -** ORE assigns one or more publishing activities to an external contractor who provides its services through its information systems and only updates the new ORE platform with the results/outcomes.

In the first scenario, the new platform has to support complex workflows to monitor each publishing activity/function. In contrast, the situation is greatly simplified in the second scenario, as the ORE platform only needs to be updated with the results.

In any case, it is suggested that the new ORE platform provides both options for the core workflows presented in the upcoming sections, namely pre-publication or preprint check, per-review and final article versions preparation.

3.2 ORE content organisation model

Determining the data organisation model is critical in developing the ORE publishing platform. Based on the current status of the ORE platform, the highest level of organisation is the scientific field (Subject Area). Furthermore, the scientific articles are organised/associated using the concepts of Gateway[24], Community Gateways[25] and Collections[26] (see figure below for more details). The existing organisation model entails specific requirements for implementing the scoping capabilities (see section **4.4.3 Scoping per Subject Area and User Role**) concerning (a) the ORE functionality, (b) the user management, (c) data presentation, and (d) SDI services[27].

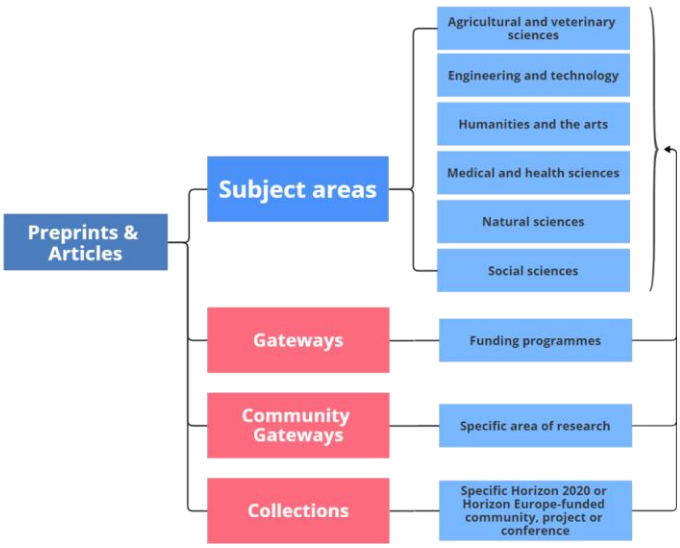


Figure 3. ORE content organisation model - Subject areas, Gateways, Community Gateways and Collections

As mentioned earlier, the main difference in how data is organised in the ORE model is that the scientific subject area constitutes the top-level entity and not the scientific journal. This means "traditional" metadata fields, such as issue, volume, page numbers, etc., are not used in ORE. Articles are classified and categorised using scientific topics and subtopics (at least

---

[24] Gateways provide dedicated portals for the different funding programmes, programme areas and their thematic sections. In each gateway, you will find articles published on Open Research Europe that are linked to projects funded under each funding.

[25] Community Gateways are dedicated hubs within Open Research Europe to bring together all content related to a specific area of research.

[26] Collections are compilations of content relating to a specific Horizon 2020 or Horizon Europe-funded community, project or conference.

[27] Selective Dissemination of Information (SDI) service is a type of "alerting service" meant for individual users, mainly specialists, which is exclusively restricted to the area of interest of the user concerned [see in Chatterjee (2017)].

with two levels of depth), the year of publication, unique identifiers, author names and other related information. The organization model depicted above requires a robust search environment with all the fields mentioned as filters and faceted navigation.

In addition, it is essential to offer the ability to organise/correlate scientific articles with specific concepts, primarily for extracting statistical indicators and grouping articles that might interest a particular community of researchers.

At this stage, six primary scientific subject areas have been defined at the top level of the hierarchy (first level) and the concepts of Gateways, Communities Gateways and Collections, whose function is briefly described in **Figure 3**. It is highly likely that more first-level subject areas and new concepts for categorising/classifying articles will be required in the future.

3.3 User roles and their functions

The essential user roles that need to be supported by the new ORE platform are the following:

- **Registered users**: Registered users have access to the communication functions offered by the publishing platform, including messaging and commenting capabilities and managing their profiles. They can also subscribe to SDI services provided by the publication platform based on their profile information (e.g., scientific areas/topics selected).
- **Authors**: Authors can perform manuscript submission and track the status of the peer review. Also, authors can respond to reviewing comments and participate in the publishing workflow (e.g., proofread the final version of the manuscript before publication).
- **Reviewers**: The Reviewer's primary role is to accept (or reject) a review assignment, access the content under evaluation and provide feedback to both the Editor and the authors through special web forms.
- **Pre-publication Manager (or Preprint Manager)**: The duties of the Pre-publication Manager are directly related to the ORE publication model. The Pre-publication Manager checks if submitted articles represent scholarly communications that adhere to author guidelines and ethical and editorial policies, including data policies. Also, the Pre-publication Manager is responsible for communicating with the authors and managing the workflow steps leading to the preprint publication (see section **3.4.1 Pre-publication check / Preprint workflow**). It is noted that Pre-publication Manager can handle all necessary checks for articles even if they are submitted in different subject areas.
- **Editor – Section Editor**[28]: Editors (and consequently Section Editors) are responsible for managing the editorial process, including the further checks of the manuscripts,

[28] The role of the Section Editor is identical to the role of the Editor for a specific subject area.

coordinating peer review, making editorial decisions, and overseeing the production and publication of the final version.

- **Copy Editor**: The Copy Editor ensures the manuscript is clear, concise, consistent and error-free. Copy editors thoroughly check the article for grammar, punctuation, spelling, syntax and style issues.
- **Production Manager – Typesetter**: The typesetter is responsible for formatting the manuscript for publication. The typesetting task includes taking the final version of the manuscript and designing the layout, selecting fonts and adding graphic elements such as pictures, tables, equations and figures.
- **Proofreader**: The Proofreader's role is to check the final version of the manuscript to ensure that it is error-free and ready for publication. Proofreading involves carefully checking the manuscript for spelling, grammar, punctuation, and formatting errors that may have been overlooked in a previous edit.
- **Publish Manager**: Publish managers are responsible for overseeing the production and publication of the article. Publish managers handle the final review (including metadata completion, Search Engine Optimisation – SEO techniques, social media etc.) of papers, the publishing schedule, copyright compliance and promotion/dissemination.

The new ORE platform should also support additional roles related to the content organisation model. Specifically, it should support the user accounts for **Funding, Community, and Collection Managers**. Through their accounts, each of these roles can access a range of statistical data generated by analysing metadata of articles included in the corresponding concept to which they are related (e.g., Gateway, Community, and Collection).

Moreover, the **System Administrator (or superuser)** role is also essential, and as expected, these users can perform the administrative and configuration capabilities.

It is noted here that depending on the workload (related to the number of articles submitted within a certain period), a user may have more than one role, or the functions of a role can also be assigned to more than one user.

The following figure presents the functionalities the new ORE platform should provide per user role.

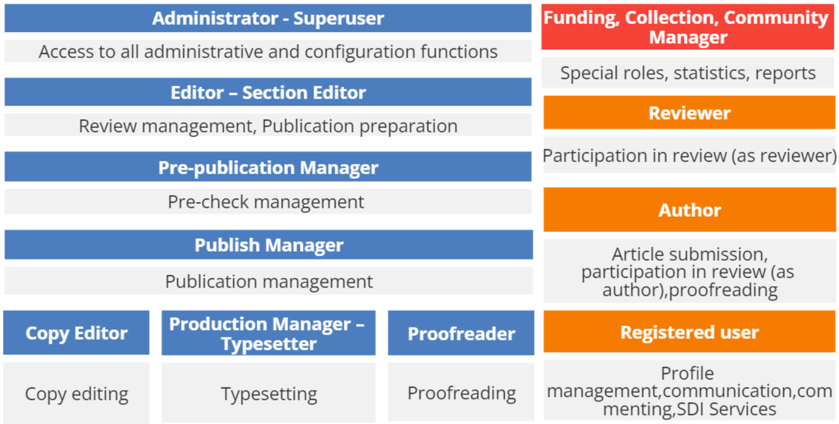


Figure 4. Roles and functionality – Inheritance hierarchy

3.4 Major workflows

This part of the study provides in detail the essential workflow processes of ORE. Candidate publishing platforms should meet, or be customised to meet, these ORE model workflow requirements. The steps of the proposed workflows are based on an analysis of the current ORE information system.

Emphasis is initially placed on presenting the workflow that concerns whether a scientific article is accepted as a preprint, known as pre-publication check. This process ensures that articles submitted for publication meet all policies, ethics and guidelines a specific subject area sets[29,30]. The outcome of the pre-publication check is the article's publication in the preprint state with the peer review still pending (awaiting peer review state). The article's publication entails formatting into a typeset version, DOI attribution, and indexing by Google Scholar and other similar citation indexes.

For a better understanding of the Open Peer Review model used by ORE, the following should be taken into account:

- The authors are obliged to provide a list of potential reviewers immediately after completing the pre-publication checks, thus publishing the article as a preprint. All reviewers must meet a set of prerequisites[31] such as: (a) to have at least three publications in the same scientific field as the article under review, (b) to have an appropriate level of qualification, and to be affiliated with a recognised institution, and (c) to not have any conflict of interest with the authoring team. Moreover, the Pre-

[29] See Articles Guidelines.
[30] See ORE Policies.
[31] See here for Reviewer Criteria.

publication Manager must thoroughly check the list of potential reviewers and reject any cases that do not meet the abovementioned criteria. At the end of this process, the Editor must send at least five invitations to review the article, through the information system, to the corresponding reviewers. The failure to find the minimum required number of reviewers or their non-positive response to invitations for review may lead the entire process to restart. The Pre-publication Manager can use reviewers already registered on the ORE's internal lists only after the authoring team accepts them.

- The ORE Open Peer Review model also assumes that the reviewers' comments and their identities should be published along with the article. In addition, according to ORE's policy, when two reviews are received and accepted by the Editor, the peer review process for the article is considered complete. At this point, the article's status changes to peer-reviewed, and the possible review states are depicted in **Table 7**. This phase of the peer review process concludes with the article's content being updated with the addition of the review comments and the publication on the ORE platform. If the review outcome is "Approved," then the article is indexed by all collaborating citation indexing services, considering its subject area (e.g., PubMed, Europe PMC, Scopus, ERIH+, British Library, CrossRef, DOAJ, INSPEC etc.).

- From the above analysis, it becomes apparent that an article is published in preprint status quickly, with the peer review process starting after the publication. Additionally, the article's evaluation results are posted after the peer review, without waiting for the author team's response to the reviewers' comments. This way, transparency in the review process is enhanced, and the indexing of the article by all citation databases is accelerated, positively impacting the research dissemination.

- For both cases of the review outcome ("Approved" and "Not approved"), the article authors can proceed with revising the content and submitting a new version. In addition to the revised version of the article, they should also respond to the reviewers' comments. As before, after performing a mandatory check, the Editor publishes the revised version of the article. The immediate publication of the revised version aims to avoid the delay of re-evaluation by the reviewers, as happens in traditional review workflows. This ensures transparency throughout the process and reduces the time for publishing the new, revised version of the article.

- The article's reviewers are informed about the revised version and the authors' responses and are called upon to proceed with re-examination and a new evaluation. Depending on their decisions for the article, the review rounds may continue or stop if the "Approved" status is achieved.

Table 7. ORE model review results

Review results	Article Review State
An article receives two "Approved"	It is considered "Approved", and it is indexed by all cooperating citation indexes.
An article receives one "Approved" and two "Approved with Reservations"	It is considered "Approved", and it is indexed by all cooperating citation indexes.

All other combinations of results	The article is considered "Not Approved", meaning the authors should revise it to respond to the reviewers' concerns.

The above analysis shows that the ORE review workflow differs substantially from traditional review workflows that apply to single, double-blind, or Open Peer Review models. It is emphasised here that the authoring team can initiate an article revision process, even if it is in an "Approved" state. The reasons for updating an article's content could be related to the need to include new results or could be triggered by comments from readers. In this case, the participation of the Editor and the reviewers is again necessary for the initial check of the revised version and the subsequent review.

In addition to the two basic workflows mentioned (pre-publication check and peer review), this chapter presents the workflow for the processing and preparation of an article for publication. Also, a simplified workflow for all processes is depicted where external service providers are used to support the ORE publishing functions (pre-publication check, peer review, copy editing, etc.).

3.4.1 Pre-publication check / Preprint workflow

In the previous section, the most important differences between the ORE model and traditional publication models regarding submitting an article, the pre-publication check process (a process that leads to the acceptance of the article as a preprint), and the review process were analysed.

This section presents a concise overview of the steps involved in the Pre-publication check workflow.

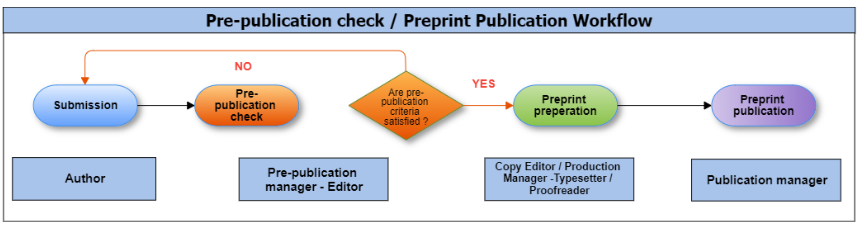


Figure 5. Pre-publication check workflow in brief

The diagram above (**Figure 5**) depicts the pre-publication check process stages and the role of the users involved. Specifically, in the first step, the author submits the article through the publication system. The Pre-publication Manager performs all necessary checks and requests the necessary adjustments/changes, if deemed necessary, from the authors. If the article fails to pass the pre-publication check, the author has two options: to resolve the comments raised or withdraw their submission and end the process. In the case that the pre-publication assessment is successfully completed, meaning the article will be published in ORE, it is sent to the production team for copy editing, typesetting and proofreading and finally published in the preprint state.

A comprehensive description of all the steps in this particular workflow can be located in **Appendix I – ORE Workflows in detail**.

3.4.2 Review workflow

In this section, the steps of the peer review workflow, based on the model followed in ORE, are presented briefly. As discussed in section **3.4 Major workflows**, the review process is divided into two phases, mainly for transparency reasons and to accelerate the publication time (see **Figure 6**).

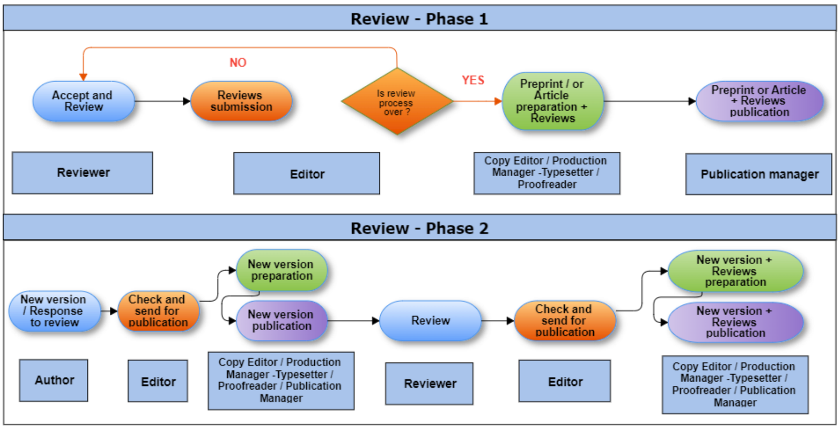


Figure 6. Review workflow Phases 1 and 2 in brief

The first phase includes all necessary steps up to the completion of the evaluation by the reviewers and the publication of the preprint version, along with the reviewers' comments, their rating, and whether the article completed the assessment successfully (got the "Approved status"). In more detail:

- The Reviewer accepts or rejects the review assignment. In case of rejection, the review process for the specific reviewer ends at this point. Otherwise, the review process starts if the reviewer agrees with the assignment.
- The Editor receives the review and accepts or rejects its outcome. If two valid reviews are submitted, the Editor sends the review results (plus the preprint version) to the production team.
- The preprint version and the review comments are published on the public portal. Authors, reviewers and registered users are notified via email with the updates.

The second phase involves the author's response to the review comments (especially when the latter is not Approved). The same phase includes the steps followed to initiate the next round of evaluation by the reviewers immediately after posting the Author's revised version of the article. In more detail:

- The Author prepares a revised version of the article, accompanied by the response to reviewer comments and submits them to the Editor.

- The Editor checks the revised version and the response to the reviewer's comments and sends them to the production team for publication. The revised version, the reviewer's comments, and the Author's response to them are published in the public portal.
- Reviewers are notified that a revised version of the manuscript has been published and are encouraged to re-review if they had initially given an 'Approved with Reservations' or 'Not Approved' status.
- The Editor receives the review(s) and sends the review results to the production team for publication.

A comprehensive description of all the steps in this particular workflow can be found in ***Appendix I -* Review workflow**.

3.4.3 Other ORE Workflows

In addition to the basic workflows presented in the previous sections, it was deemed appropriate to study and analyse in-depth two more workflows. Specifically, in Appendix I, the following workflow cases are also included:

Simplified workflow – Use of external service providers. The goal of this particular workflow is to present an alternative scenario for the operation of ORE, where most of the functions of the publishing cycle are assigned to external service providers without using the ORE platform. The scenario is indicative and valuable in providing the interaction points between external systems and the ORE platform. The functions that could be assigned to external service providers are the following:

- Pre-publication check
- Review
- Copy editing / Typesetting and Proofreading (Production)

Manuscript preparation workflow. This workflow aims to present the steps followed for producing ready-to-publish manuscript versions. An external collaborator will provide this specific task as part of ORE's operation. Nevertheless, it is recommended to implement it to enable the realisation of Scenario 1, as analysed in section **3.1 Introduction – Aim**.

Appendix I - Manuscript preparation workflow and Manuscript preparation workflow sections comprehensively describe the abovementioned workflows.

3.5 Summary – Conclusions

The main points in which the ORE publishing platform differs from a traditional scientific publication platform are as follows:

- The model for organising content focuses on the scientific domain (Subject Area) instead of the scientific journal.

- The Pre-publication check is an entirely new workflow and results in a publication action (preprint publication).
- The realisation of the review workflow requires two separate stages and results in three publication actions (the reviewers' comments and decision, the updated version of the manuscript and the authors' responses to the reviewers and finally, the new comments from the reviewers).

These new capabilities – workflows affect many aspects of the platform, such as data submission forms, user roles, content discovery, and content presentation.

Additionally, outsourcing certain parts of the publishing cycle to external contractors, especially at the first steps of ORE service, may simplify some workflows and require less development effort.

Based on the above, this chapter presents an analysis of the main features and characteristics that should be considered during the design phase of the new ORE publishing platform. Specifically:

- A cohesive data organisation model (scientific articles) based on Subject Areas and the use of Gateways and Communities as an alternative way of presenting content.
- The essential user roles and their capabilities.
- New workflows, analysed step by step.
- Additional simplified workflow, enabling collaboration with external contractors.

From all the above, a series of conclusions can be drawn, importantly:

- The new ORE Publishing Platform should include the workflows for pre-publication check and review in its implementation plan. These specific workflows should be able to support both Scenario 1 and Scenario 2 (see section **3.1 Introduction – Aim**)
- The new ORE Publishing Platform may preferably not implement the workflows for producing and creating ready-to-publish files, considering that this service will be provided exclusively by an external contractor outside of the ORE publishing infrastructure.
- The new ORE Publishing Platform should fully support all functions for publishing content, as analysed in the workflows presented in the previous sections.

CHAPTER 4. FUNCTIONALITY ANALYSIS AND REQUIREMENTS FOR THE NEW ORE PUBLISHING INFRASTRUCTURE

4.1 Introduction – Aim

This chapter aims to collect and encode as much information as possible regarding the functionality the new ORE Publishing Platform should offer and the mandatory and optional requirements for its implementation. Some general aspects, such as those analysed below, must be considered to achieve this goal.

The Commission's strategic choice is to build[32] the ORE Publishing Platform upon an existing publishing software that follows the open-source code use and distribution model. There are several advantages to making this choice. Specifically, building upon existing solutions could speed up the whole process, reduce development costs, and minimise the significant risk often faced in large-scale software projects. Additionally, collaboration with an active community of programmers, end-users and stakeholders for supporting and further developing existing software by adding new capabilities and functions could significantly improve the degree of sustainability. Furthermore, implementing the ORE publishing model as part of an open-source solution could immediately boost its exploitation by other publishing efforts, further promoting the Commission's vision for innovative ways of publishing scientific knowledge.

On the other hand, choosing an existing solution to adapt it to meet the needs of a new publishing workflow entails a series of risks. There is always the possibility that the required changes will restructure the software's code and functionality to an extended degree. The consequences of such a development are twofold. Specifically, the developing workload might be significantly more extensive than anticipated, and it is no longer feasible to include the modified source code in the open-source software's main codebase. In this case, the ORE Publishing Platform will be developed in parallel with the original open-source publishing software as a separate product, deviating from the initial goal for high sustainability and impact.

A balanced approach must be embraced regarding the complexity and the level of detail in describing the technical specifications and requirements for developing the ORE Publishing Platform. A set of complex and demanding requirements/conditions will result in significant developing effort, which should be avoided during the initial stages of the implementation. Furthermore, a detailed description of specific technical requirements is not recommended, as more solutions/approaches for implementing similar functionality exist. For example, as mentioned in Chapter 2, there are platforms that could be used as the basis for implementing the ORE Publishing Platform. They provide similar functionality for all publishing activities by following different logical and technical approaches. In other words, it is suggested that the level of detail in describing the technical specifications and requirements should be kept low and leave plenty of space for alternative solutions and proposed implementation methods.

[32] Meaning to further develop / adopt / customise.

Furthermore, as expected, the operating model that will be followed to support the ORE Service directly affects the implementation of the ORE Publishing Platform. Specifically, for at least the first phase of the ORE Service operation, it may be envisioned that external contractors conduct a large part of the publishing cycle processes, as also suggested in a recent Commission-supported analysis on "Operationalising Open Research Europe as a collective publishing enterprise" (European Commission. Directorate General for Research and Innovation., 2022, p. 53). At the same time, the in-house staff of the ORE Service may have mainly administrative and supervisory roles. As analysed in Chapter 3, the external contractor could use the ORE platform to manage all publishing activities or to use its system and manually upload the process outcomes to ORE (Scenario 2 - see section **3.1 Introduction – Aim**). Alternatively, an API for data exchange could be developed to achieve interoperability of the ORE Publishing Platform with the contractor's information systems. Whatever the choice, it directly impacts the development and implementation of the ORE Publishing Platform. For example, copy editing, typesetting and proofreading services are currently outsourced to an external contractor. According to Scenario 2, communication between the contractor and the ORE Publishing Platform will be limited to a dual message exchange: (a) the assignment of a task along with the material at the external contractor and (b) the delivery of the ready-to-publish set of files with the relevant comments from the contractor. Consequently, the execution of the production workflow steps, as they were presented in section **3.4.3 Other ORE Workflows**, will not be managed by the ORE Publishing Platform. Therefore, deciding the workflows that will be supported through the ORE Publishing Platform in the first development phase and which will be left as pending tasks for the future is essential.

Considering the above observations, the following sections present the requirements and functionalities that should be part of the new ORE Publishing Platform. These requirements concern (a) the hosting infrastructure, (b) the publishing platform's general characteristics, and (c) the features related to the publishing activities. The following terms are used in the section:

- **Proposed Publishing Platform:** The OSS Publishing Platform that is selected for implementing the ORE Publishing Platform.
- **ORE Publishing Platform:** The new publishing platform resulting from the adaptation of the proposed publishing platform.
- **ORE IT Solution (Infrastructure or hardware):** The environment for hosting the new ORE Publishing Platform. The ORE IT Solution consists of hardware and software (operating system, database management systems, virtualisation software etc.).
- **ORE Service Solution:** Refers to the ORE IT Infrastructure and ORE Publishing Platform.
- **ORE Service Contractor:** Refers to the selected contractor to develop, operate, support and maintain the ORE Service Solution.

It should be noted here that some of the requirements for the proposed ORE OS platform are the same as they were presented in the 2019 ORE Tender Specifications document[33].

[33] See tender RTD/2019/OP/006 - Open Research Europe — The European Commission Open Research Publishing Platform / Tender Specifications.

Finally, it should be emphasised that the requirements that follow are indicative.

4.2 Introduction – Aim

The IT solution for hosting the **ORE Publishing Platform** must be operated on a corresponding IT infrastructure or hardware for which different design types are feasible (e.g., physical client-owned machines, virtualised machines, cloud services etc.). The hardware specifications must comprise backup, restore, performance, security, scalability, and data storage location.

The setup and maintenance of the operation of the **ORE IT Solution** need to be externally hosted in a professional, reliable, and secure manner to fulfil the given requirements by the European Commission. The **ORE Service Contractor is responsible for proposing** the type of the **ORE IT Solution**, the mode of operation, and its performance characteristics and for providing maintenance and support services.

The provision of data centre services should include a reliable and efficient backup mechanism in case of any data loss or breakdown of the service. The backup scenario should be adjusted to the level of criticality of different data categories and the number of transactions. Different backup approaches could be decided based on the software requirements (e.g., item-level or image-level backup in case of IT infrastructure that utilises virtualised machines). For the ORE needs, daily backups are mandatory unless a narrower time window is decided.

The solution for the IT infrastructure selected by the **ORE Service Contractor** to implement the **ORE IT Solution** shall be equipped with up-to-date security and auditing standards. The operations of the IT infrastructure should be confirmed through external certification organisations to guarantee the prevention of data loss and data protection based on internationally defined and comparable standards such as ISO or tier topology.

The place of data storage shall be defined within the documentation of the **ORE IT Solution** as it affects how the General Data Protection Regulation is implemented, particularly in cases where personal data are processed. There may be a requirement to hold and store the data in a specific geographical area or country (preferably a member of the European Union).

The ORE Publishing Platform should be based on open-source software. Nevertheless, some parts of the infrastructure software (e.g., operating system, database management software, virtualisation software, etc.) may rely on a commercial solution for quality, reliability, and better performance. In any case, the ORE Publishing Platform software should be compatible with open-source software solutions for implementing the installation environment.

The **ORE Service Contractor** should prove the fulfilment of the infrastructure requirements in the best possible way, explain the hardware/software concept in detail, explain how the performance indicators are complied with, and provide details on the backup scenario and security measures.

4.3 ORE Platform generic requirements

The implementation of the **ORE Publishing Platform** should be based on an existing open-source publishing platform (**Proposed Publishing Platform**) that complies with the following general sets of requirements. After the development/adaptation ends, the new **ORE Publishing Platform should also comply with these requirements.**

4.3.1 License, community and development requirements

- The **Proposed Publishing Platform** must adopt an open-source license for using and distributing the code without any restrictions.
- A large and active community must support and maintain the proposed open-source publishing platform, including programmers, institutes, funders and end-users. The community characteristics and interest in the platform can be demonstrated by its population, the number of different installations, downloads, and bug-fix response time.
- The **Proposed Publishing Platform** must present adequate software releases/upgrades in the last three years.
- The **Proposed Publishing Platform** must have a concrete, long-term plan for future changes and upgrades.
- The **Proposed Publishing Platform** must use popular, modern programming languages and software frameworks for code development without restrictions.
- The **Proposed Publishing Platform** must be compatible with as many infrastructure environments as possible (operating systems, databases, etc.).
- The **Proposed Publishing Platform** must follow international standards for storing data and metadata and implementing data exchange and communication interfaces *in line with the European Interoperability Framework*[34]. *Propriety/exclusive data and metadata formats that are unavailable or not portable to other solution providers are unacceptable*.
- The **Proposed Publishing Platform** must provide complete documentation for the source code, database schemas, and application programming interfaces (APIs).
- The **Proposed Publishing Platform** must provide complete administrative operations and functional modules documentation.

4.3.2 Performance, availability, support and maintenance requirements

The **ORE Publishing Platform** must meet Requirements 5 to 9, Part A. Infrastructure Requirements of the 2019 Tender Specifications for ORE. In brief, these requirements concern the following:

---

[34] See here.

- The platform shall be accessible and functional 99.9% of the time, ensuring a minimum capacity of 300 concurrent users with a response time not exceeding two seconds during peak traffic.
- Scheduled maintenance should not result in unavailability periods exceeding 4 hours for standard maintenance or 12 hours for major upgrades and should be scheduled outside working hours.
- During regular working hours, IT support will be provided via an online helpdesk featuring a ticketing service, and responses should be addressed within one working day.

4.3.3 System Architecture

- The **ORE Publishing Platform** must allow module-based software architecture with direct communication between modules for the core system/application. Also, the proposed publishing platform must allow the integration of additional functionality to the core application through plugin modules.
- The **ORE Publishing Platform** must offer a responsive, web-based interface for accessing all administrative and publishing functions.
- The **ORE Publishing Platform** must offer the ability to **operate in any of the following modes:**
    - (**Preferable**) **Mode 1:** Core system single installation - multiple instances, one per subject area, supported by an advanced scoping mechanism allowing end-users to access different publication functions and parts of the platform.
    - **Mode 2:** Core system multiple installations, one per subject area, supported by a unified end-user management mechanism.

4.3.4 Security and data protection requirements

The **ORE Publishing Platform** must be secure in various aspects to protect user data and content adequately. Such aspects must have been considered during the software design and optimally integrated into the business processes.

- A dedicated login shall secure the access limitation to the **ORE Publishing Platform** to a specified group of users. The login options can have different steps and levels.
    - (**Mandatory**) Users can use at least their **ORCID**, **eduGAIN**[35] and **EU login** accounts as authentication mechanisms to register and sign into the ORE Platform.
    - (**Optional**) username and password generated by the user with final confirmation by administrator **or** username and password, where username is given by administrator and password set by a system user **or** username and password where the operator sets username and password, but the

[35] See here.

password is subject to change by the user itself **or** support for a Multi-Factor Authentication (MFA) mechanism requesting two or more pieces of information for the login to the system (e.g., username, password, and additionally a PIN received via a phone number or email).

- Role-based security allows end-users to access different publication functions and parts of the platform based on the subject area hierarchy model, enabling scoping capabilities.

In addition, the **Proposed Publishing Platform** must meet Requirements 10 to 11 and 13 to 17 as they are described in *Part A. Infrastructure Requirements of the 2019 Tender Specifications for ORE*. In brief, these requirements concern the following:

- The data and information within the publishing platform must be segregated from other unrelated information. A strategy should be developed, and appropriate actions should be implemented to avoid failures in this isolation as part of the security plan.

- The preservation of information and data generated through the platform's usage (such as transaction logs and internal communications) must be ensured.

- The implementation of security incident response procedures is mandatory.

4.3.5 Data migration

The **ORE Service Contractor** must ensure data portability, namely the content and its administrative information between the current ORE Information System and the new **ORE Publishing Platform**.

The data that needs to be transferred to the new ORE Publishing Platform are the following:

- The structure and the content of the internal organization scheme (Subject areas, Gateways, Collections, etc.).
- Articles and all accompanying data (all versions, metadata, comments, ratings, etc.).
- Administrative data for each article is in the form of a log file.

To achieve the above goal, a collaboration between the responsible parties of the existing information system of ORE and the new one is necessary.

4.3.6 EOSC compliance

The European Open Science Cloud (EOSC) initiative provides European researchers and professionals access to a wide range of open scientific data, tools, and services. To be compatible with the EOSC, a service should meet the following requirements[36]:

- **Compliance with FAIR principles:** Services should ensure that the data they generate or manage is Findable, Accessible, Interoperable, and Reusable (FAIR). This means providing precise metadata, persistent identifiers, and standardised data formats to promote data discovery and sharing.
- **Open standards and protocols:** Services should adopt open standards and protocols to facilitate seamless integration with other services within the EOSC ecosystem. This includes using standard APIs, data formats, and communication protocols.
- **Data privacy and security:** Services should ensure the protection of user data and adhere to relevant data protection regulations, such as the General Data Protection Regulation (GDPR). This includes implementing appropriate access controls, encryption, and other security measures.
- **Interoperability:** Services should be designed to work with other services and tools within the EOSC ecosystem. This includes using standard vocabularies, ontologies, and data models to enable data exchange and integration across different services.
- **Transparent and open licensing:** Services should make their data and code available under open licenses that promote reuse and collaboration. This includes adopting licenses such as Creative Commons (for data) and open-source licenses (for software).
- **Sustainability:** Services should have a transparent business model and long-term plan to ensure their sustainability and continued availability within the EOSC ecosystem. This includes securing appropriate funding and support from stakeholders.
- **User support and documentation:** Services should provide comprehensive documentation, tutorials, and support to help users effectively use the service. This includes maintaining up-to-date user guides, FAQs, and providing responsive support channels.
- **Quality assurance and monitoring**: Services should implement quality assurance processes to ensure the reliability and accuracy of the data and tools they provide. This includes regular monitoring and evaluation of the service's performance and impact.
- **Inclusivity and accessibility:** Services should be designed to be accessible to all users, regardless of their technical expertise or background. This includes providing user interfaces that are easy to use and accessible to users with disabilities.

---

[36] Vitlacil, Dejan, Livenson, Ilja, Riungu-Kalliosaari, Leah, & Lukkarinen, Ari. (2020). D3.1 EOSC Service compliance checklist and maturity model. Zenodo. https://doi.org/10.5281/zenodo.5541618.

- **Adherence to EOSC policies and guidelines:** Services should adhere to the policies, guidelines, and governance structures established by the EOSC. This includes contributing to developing and implementing EOSC standards, guidelines, and best practices.

A service can become valuable to the EOSC ecosystem by meeting these requirements, enabling researchers and professionals to access and share data and resources more effectively.

4.3.7 GDPR compliance

The General Data Protection Regulation (GDPR) is an all-encompassing data privacy regulation applicable to organisations functioning within the European Union or handling the personal information of EU citizens. For an information system to be compliant with GDPR, it needs to fulfil the following essential requirements[37]:

- Provide data protection mechanisms across all information system functions related to data processing.
- Support encrypting, pseudonymizing, or anonymising personal data.
- Support mechanisms for detecting data protection incidents.
- Support functionalities for the users that include:
  - Obtaining all the information, the platform has about them in a format that can be effortlessly transferred to another application.
  - Allowing them to correct or update inaccurate or incomplete information.
  - Allowing them to delete their personal data.
  - Allowing them to request to cease processing their data.

By incorporating these requirements into the design, development, and operation of software, organizations can ensure that their software is GDPR-compliant and appropriately manages the personal data of EU citizens.

4.4 ORE Platform publishing requirements

This section briefly presents the new ORE publishing platform's basic requirements.

[37] The checklist offered in this section is a representative selection derived from here, primarily concentrating on the technical aspects of information systems. Achieving GDPR compliance is a more intricate process that necessitates consultation with experts in the field.

4.4.1 Organisational structure

The data organisation model of the ORE platform was presented in detail in section **3.2 ORE content organisation model**. The highest level of organisation is the Subject Area. Furthermore, the scientific articles are organised using the concepts of Gateway, Community Gateways and Collections. The ORE platform should offer the ability to add more Subject Areas and configure Gateways and Collections via the administration panel.

4.4.2 User management - User roles

The user management module should allow users to set permissions and access rights based on their responsibilities within the publication cycle. The new ORE platform should support the user roles as they were defined in section **3.3 User roles and** their functions. Each role should have a predefined set of permissions, which determine what actions the user can perform within the ORE platform. The User management module should also provide the registration and authentication mechanisms as they were described in section **4.3.4 Security and data protection requirements**. Users should be able to log in using ORCID, eduGAIN and EU login accounts.

Additionally, the user management module should support the creation and update of user profiles where user information will be recorded, such as:

- Personal details
- Bio information
- Contact details
- Subject Areas of interest (useful for the SDI service and reviewers matching)
- Unique identifiers (e.g., ORCID ID, Scopus ID, etc.) and social media accounts

4.4.3 Scoping per Subject Area and User Role

Scoping in the ORE publishing platform defines the specific features, functionalities, and access permissions available to back-office users based on Subject Area and user role. The following table provides an indicative scenario for scoping in particular user roles.

Table 8. ORE model review results

Role	Scoping
Reviewers	Reviewers can peer-review articles from all Subject Areas
Pre-publication Managers (or Preprint Managers)	They can be assigned to manuscripts from all Subject Areas
Editors – Section Editors	They can manage manuscripts only for a specific Subject Area

Copy Editors, Production Managers – Typesetters, Proofreaders	They can be assigned manuscripts from all Subject Areas
Publish Managers	They can be assigned manuscripts from all Subject Areas

It is evident from the above scenario that only Editors are associated with a specific Subject Area, meaning that the system allows access only to the manuscripts submitted on this scientific topic. In any case, the system should offer the ability to configure more complex scoping scenarios for other role types and multiple Subject Areas.

4.4.4 Notification and communications – User dashboard – SDI Service

The new ORE publishing platform should offer a complete notification and communication module for sending predefined messages to users participating in its operation. These messages will be sent via email and will concern all possible tasks and functions of the various publishing stages. The sender and/or administrator must be able to modify their content. The message and the communication logs should be accessible to users and administrators.

A dashboard that provides a centralised view of important information and functionality, notifications and pending tasks should be available for all user types. The purpose of a user dashboard is to provide a convenient and intuitive way for users to interact with the software and stay informed about their activities and progress.

Finally, a Selective Dissemination of Information (SDI) module should be provided to individual alert users of newly published content in their interest.

4.4.5 Integrations with third-party services - APIs – Data and Metadata standards

The new ORE publishing platform should support a plugin framework to make adding new elements/functions to the core software easier. Also, the ORE publishing platform should offer a range of integrations with the following third-party services:

- Assignment of DOIs to various types of digital objects (manuscripts, preprints, articles, versions of articles, research funding organisations, reviews, etc.) with one of the existing management services (e.g., CrossRef, DataCite, etc.).
- Information retrieval from various services such as ORCID, RESEARCH ID, etc.
- Connection for depositing research data or retrieving information (e.g., Zenodo).
- Connection for depositing or retrieving content in repositories (at least from Zenodo or other repositories compatible with OPENAire, EuropePMC[38], and PubMed).

[38] See here for EuropePMC.

- Connection with major citation indexing services (Google Scholar, Semantic Scholar[39], Scopus, Web of Science, etc.).
- Connection with services for retrieving statistical data at the article level, author level, etc. (Publons[40], PlumAnalytics[41], OpenAIRE UsageCount[42], etc.).
- Connection with a plagiarism detection service.
- Connection with a service for long-term archiving of the content externally (e.g., Portico[43], LOCKSS[44] etc.).

The new ORE publishing platform should support public APIs for metadata harvesting (e.g., OAI PMH), use standard metadata formats [e.g., Dublin Core, MARC], and popular content XML standards [e.g., JATS-based XML].

To implement controlled vocabularies, taxonomies and other classification schemes, the ORE Publishing Platform should rely on existing European and International standards, such as the OECD Frascati Manual.

4.4.6 Import – Export

The new ORE publishing platform should support import and export functions for all data. The formatting of exports should follow standard XML styles.

4.4.7 Administration features - Reporting

The Administration features of ORE Publication Platform include functions such as:

- User Management,
- System configuration (e.g., Subject Areas creation)
- Various customisations (web pages, submission forms, workflows etc.)
- Plug-in management
- Internal reports and statistics
- Export / Import functions

---

[39] See here for Semantic Scholar.
[40] Publon in now part of Web Of Science platform.
[41] See here for PlumAnalytics.
[42] See here OpenAIRE UsageCount.
[43] See here for Portico.
[44] See here for LOCKSS.

4.4.8 Publication activities

4.4.8.1 Submission

The submission of a manuscript should at least include the following information:

- Manuscript upload
- Supplementary files upload (e.g., images, figures etc.)
- Complete set of descriptive metadata
    - Article and authors' information, authors' contribution, keywords, subject headings, special declarations, funding details, notes to editors, etc.
- Licensing information
- Open Peer Review Support
    - Quality check guidance during the submission
- Support for external integrations
    - ID services [author login, profile data – ORCID, etc.]
    - Data repositories (input/output)
    - Other services, e.g., PIDs [funders/research data…]
- Reference management
- Reviewers list

4.4.8.2 Pre-publication check

The ORE Publishing Platform must support the entire pre-publication check workflow presented in section **3.4.1 Pre-publication check / Preprint workflow**.

4.4.8.3 Review - Reviewer Management

The ORE Publishing Platform must support the entire review workflow presented in section **3.4.2 Review workflow**.

In addition, the ORE Publishing Platform must provide a mechanism for managing the reviewers with capabilities such as:

- Searching for reviewers in various ways, including the Subject Areas they have declared in their profiles.
- The ability to view their evaluation history and statistical data.

4.4.8.4 Production - Manuscript preparation

The ORE Publishing Platform must support the entire production/manuscript preparation workflow presented in section **3.4.3 Other ORE Workflows.**

4.4.8.5 Publishing

The functions of publishing are mandatory before uploading/posting a manuscript on the public website of the platform. The actions involved in publishing management are listed next and immediately impact the manuscript presentation (see next section).

- Handling various content formats (e.g., pdf, XML, HTML etc.) and different versions.
- Final correction/completion of metadata, primarily those related to the subject classification.
- Curation of intellectual property issues and usage licenses for the content.
- Assigning unique identifiers to the content (manuscript, reviews, research data etc.).
- Completion of proper metadata and application of techniques to support SEO.
- Further dissemination of the publication in third-party repositories and citation indexes.
- Promotion of the article through social networks.
- Monitor the article's metric and manage comment moderation.

Upon completing the above steps that allow an article to be published, all processes related to end-user notification (e.g., author group, reviewers, registered users, etc.) are triggered.

4.4.8.6 Presentation – Web interfaces

The presentation and search interface for the content is one of the most important aspects of the new ORE Publishing Platform. It should offer visually appealing content representation and have advanced search and navigation capabilities. The content discovery and retrieval should be able to be conducted in the following ways:

- All content from all Subject Areas
- Per Subject Area
- Per Gateway or Collection

For each one of the previous cases, the options presented below should be offered to end-users:

- Full-text indexing and searching.
- Advanced Search in all metadata fields and in the full text.
- Multi-facet search support [e.g., by Year, Author, Status, Subject area, Gateway, Topic, Affiliation, Funder etc.].
- Expand/restrict options per facet element.

- For all web pages SEO techniques should apply.

In addition, the article's web page should include:

- Review status / all metadata / Reviews / Full text in various formats (at least pdf, XML and HTML) / links between versions/links with research data / external repositories.
- Dissemination tools (share/export/download / cite / monitor).
- Commenting - Discussion tools.
- Metrics/indicators [Article level].
  - Internal metrics: views, downloads, visitors' characteristics, comments etc.
  - External metrics: cites (from various sources), the activity of datasets/repositories, altMetrics, social media (e.g., Twitter, Facebook), etc.

4.4.8.7 Commenting

Registered users should be able to log in using their credentials and leave comments for an article. The Commenting module should at least offer capabilities such as:

- Reply and threading
- Comment moderation
- Spam prevention

Also, commenting should trigger the notification and communication mechanism for alerting related users (authors, editors, reviewers etc.) of new comments or responses.

4.4.9 Metrics

The ORE Publishing Platform must support performance metrics and indicators for Subject Areas, Gateways and Collections. The list below presents an indicative scenario for such metrics:

- Aggregated data from all articles
- Internal metrics: views, downloads, visitors' characteristics, comments etc.
- External metrics: cites (from various sources), the activity of datasets/repositories, altMetrics, social media (e.g., Twitter, Facebook), etc.

Performance indicators

- The number of articles submitted/approved/revised etc.
- Review time

4.5 Summary – Conclusions

A set of conclusions were reached by combining the findings from Chapter 2 (maturity of existing open-source publication platforms) and 3 (new workflows) with the operational and technical requirements outlined in Chapter 4. Specifically:

The following aspects will necessitate structural modifications in the current open-source publication platforms:

- The data organization model, as analysed in section **3.2 ORE content organisation model**.
- Support for the Pre-publication/preprint check workflow.
- New forms for the submission of articles by authors.
- Support for the Review workflow.

Moreover, how content is searched for and displayed to users on the current ORE platform and the related search environments of the open-source solutions, explored in Chapter 2, indicate that this specific feature demands substantial transformation and a considerable investment in development. The solution that ORE ought to implement must support the features and capabilities detailed in section **4.4.8.6 Presentation – Web interfaces** while emulating search environments from reputable scientific content providers (e.g., OpenAire[45], Scopus[46], WoS[47], VuFind[48], etc.). Drawing from this analysis, we conclude that the ORE public web portal might utilise software distinct from the one responsible for the publishing platform's core functions. This alternative/additional software would be given to the community to enable further enhancements to the user experience within existing systems.

It was subsequently discovered that existing open-source publishing platforms partially offer several features essential for the best possible user experience on ORE. In this sense, additional development and enhancement are required. These specific features include the following:

- User management – Additional login options
- GDPR requirements
- SDI Service support
- Integrations with third-party services, such as
    - Pre-print repositories,

[45] **OpenAIRE** search web portal.
[46] **Scopus** web portal requires subscription – See here a quick reference guide.
[47] **WoS** (Web of Science) web portal requires subscription – See here a quick reference guide.
[48] **VuFind** is a versatile, open-source discovery platform that allows users to explore and search a wide range of scientific resources, including books, journals, etc. Its goal is to improve the user experience by providing features such as faceted search, advanced search capabilities, and individualized user accounts for storing searches and organizing favorites.

    - Research data repositories
    - Statistical and other metrics services
- Support of vocabularies, taxonomies and other classification schemes
- Metrics support, as described in section **4.4.9 Metrics**

Finally, as stated in the introduction, a balanced approach should be adopted concerning the intricacy and depth of detail in outlining the technical requirements for developing the ORE Publishing Platform.

A collection of complicated and challenging requirements may lead to considerable development efforts, which should be circumvented during the initial implementation stages. Additionally, providing an in-depth description of particular technical requirements is not advised, as multiple solutions or approaches for implementing similar functionality exist.

CHAPTER 5. RECOMMENDATIONS – FINAL CONCLUSIONS

In the concluding chapter of this research, key recommendations and suggestions, along with a range of conclusions, are put forward. It is worth mentioning that the information offered here has already been thoroughly examined and outlined in earlier chapters. The principal proposals can be summarised as follows:

- Leverage existing open-source software (OSS) instead of initiating the development of a new platform from the ground up. By building on pre-existing solutions, the process can be accelerated, development costs reduced, and the substantial risks commonly associated with large-scale software projects minimised.
- Incorporating the ORE publishing model into an open-source solution has the potential to rapidly increase its adoption by other publishing endeavours, thereby further advancing the Commission's vision for innovative approaches to disseminating scientific knowledge.
- Integrating various software solutions to implement new ORE platform components may reduce development efforts and costs; however, it could also heighten complexity and diminish sustainability. An exception to the above observation could be the public web presentation layer of the content. Specifically, a software solution specialising in data representation, rather than the one responsible for the publishing platform's core functions, could provide an enhanced end-user experience.
- Ensure that the changes introduced by the ORE implementation are integrated into the core code of the selected open-source platform or as plug-ins/add-ons. Collaborate with the community, even if it necessitates modifying certain aspects of ORE.
- The business model for the ORE Service, which depends on external contractors for publishing processes and in-house personnel for administrative tasks, influences the platform's implementation. Determining which workflows the platform should support and which should be reserved for future development is essential for the initial development phase. For instance, concentrate solely on submission, pre-publication check, review, and web portal presentation – adhere to Scenario 2 for all other workflows.
- Re-evaluate whether the ORE content organisation model (and other workflows) remains valid – any modifications should be made before the ORE implementation platform.
- Pursue a phased implementation model - include implementation services in a maintenance contract (e.g., development days).
- Ownership of the initiative of developing a new OSS ORE platform necessitates extra workload from the Commission side. Be prepared to add additional personnel to the ORE management team.
- Be prepared to support and sponsor the relevant OSS community to ensure the investment's sustainability and perennity. For example:
    - Assume a central role and finance/support community activities as an integral part of ORE's strategic sustainability plan.

    - Facilitate the integration of authoring tools into the platform and assist with the implementation of toolkits for migration to the ORE publication model, available to other publication endeavours.

- Since the ORE publishing initiative deviates from traditional models (e.g., journal-style publication), introducing new incentives for authors to publish their work in ORE is essential. Emphasise features such as:
    - Indexing ORE articles in additional citation databases.
    - Providing metrics and statistics for each article and author.
    - Integrating with external services for enhanced information and content exchange.
    - Developing and introducing new indicators that promote and encourage publication on ORE.
- Operating an information system necessitates readiness for minor or major adjustments and even its replacement for different reasons (e.g., as seen with eLife and Libero). Consequently, it is essential to continually assess factors such as the strategy employed by the community that offers software support or the external contractor and the degree to which the solution is at risk of becoming technologically obsolete.
- In conclusion, while not obligatory, having a single entity serve as the provider for the development services of the new ORE platform software, the hosting infrastructure, and ongoing support and maintenance is advisable. Although employing multiple partners might sometimes be necessary, it can considerably complicate daily operations and allocate responsibilities when dealing with errors and failures.

Some important conclusions are reserved for the end.

- Open-source software, particularly when tailored to needs and demands that diverge from the strategy of the community that supports it, can incur substantial costs, which are frequently on par with those associated with proprietary solutions.
- Choosing an existing OSS to adapt it to meet specific requirements, while recommended, also entails a series of risks.
    - If the modifications required are far more extensive than initially anticipated, incorporating the altered source code into the main codebase of the open-source software becomes infeasible. Consequently, the forked version will be developed alongside the main (version) as a distinct product, moving away from achieving high sustainability and impact.
    - Similar to the case of proprietary software, there is a lock-in possibility with the external development/support contractor.
- Existing open-source software partially meets the required functions to support the ORE scientific publishing workflow. Based on the findings of Chapters 2 to 4, three existing OSS for scientific publishing meet numerous ORE requirements, namely PKP OJS, Janeway and Kotahi.

In conclusion, we are at a pivotal juncture regarding how research is carried out and its outcomes disseminated. The majority of evidence suggests that the struggle between Open

Science and Open Access versus the misdirected protection of intellectual property rights is at its peak.

In this sense, all technical and technological barriers to open-access scientific publication initiatives must be removed. In addition, operating expenses for those seeking to promote open-access scientific publishing endeavours should be minimised without imposing additional financial burdens on authors. The ORE Service should play a central role in this direction.

APPENDIX I. ORE WORKFLOWS IN DETAIL

Pre-publication check workflow

Pre-publication check workflow steps (see Figure 7)

Step 1. The Author starts a new paper submission process. All necessary information is typed in a submission form (article's metadata, manuscript file, supplementary files, licensing options, PIDs, etc.). Submission form elements may differ according to the needs of each subject area.

Step 2. The Author finally submits the article.

Step 3. The Pre-publication Manager (or Editor) receives a notification and starts the pre-publication check procedures. The pre-publication check entails plagiarism detection through an external service and confirmation that author guidelines, ethical and editorial policies, data policies, language quality, figure appearance etc., are followed by the article under examination.

Step 4. The Pre-publication Manager prepares a response based on the outcome of the check process, including the decision, and notifies the authors.

Step 5. If the submitted article fails to pass the pre-publication check, the Authors have two options: to resolve the comments raised (**Step 6.1**) or withdraw their submission (**Step 6.2**, where the process ends and the submission is archived). If the pre-publication check is successfully completed, the process continues at **Step 8**.

Step 6.1. The Author works on resolving pre-publication comments.

Step 7. The Author submits a new version of the manuscript and/or a response to pre-publication comments. The pre-publication process starts again with **Step 3**.

Step 8. In this step, two parallel tasks are initiated. The first task is drafting the candidate reviewer's list with the authors. The second task is responsible for the article preparation by the Copy Editor / Production Manager (preprint version). The methodology for drafting the candidate reviewer's list has already been discussed in the previous section.

Steps 9 and 10. The Copy Editing / Production / Typesetting / Proofreading team receives the pre-publication material and starts preparing the preprint version of the article.

Step 11. The preprint version of the article is sent to the Editor for a check.

Step 12. The Editor checks the preprint and sends it to the authors for final proofreading.

Step 13. The Author performs proofreading and, based on the findings, submits comments/remarks.

Step 14. The Author's comments are received by the production team (the Editor is also notified), and the preparation of the final version of the preprint version is performed.

Step 15. The Production team sends the final version to the Editor for a last check and publication.

Step 16. The Editor forwards the preprint version to the Publication Manager.

Step 17. The Publication Manager processes the material for publication (e.g., add metadata, information related to Search Engine Optimisation, DOIs, other PIDs, etc.).

Step 18. The preprint version is published at the public portal. Authors and registered users (according to their SDI service settings) are notified via email. The preprint is uploaded to third-party preprint servers (optional), and selected citation indexes (e.g., Google Scholar) are updated with the new publication. Social media and other means of dissemination are automatically getting also updated.

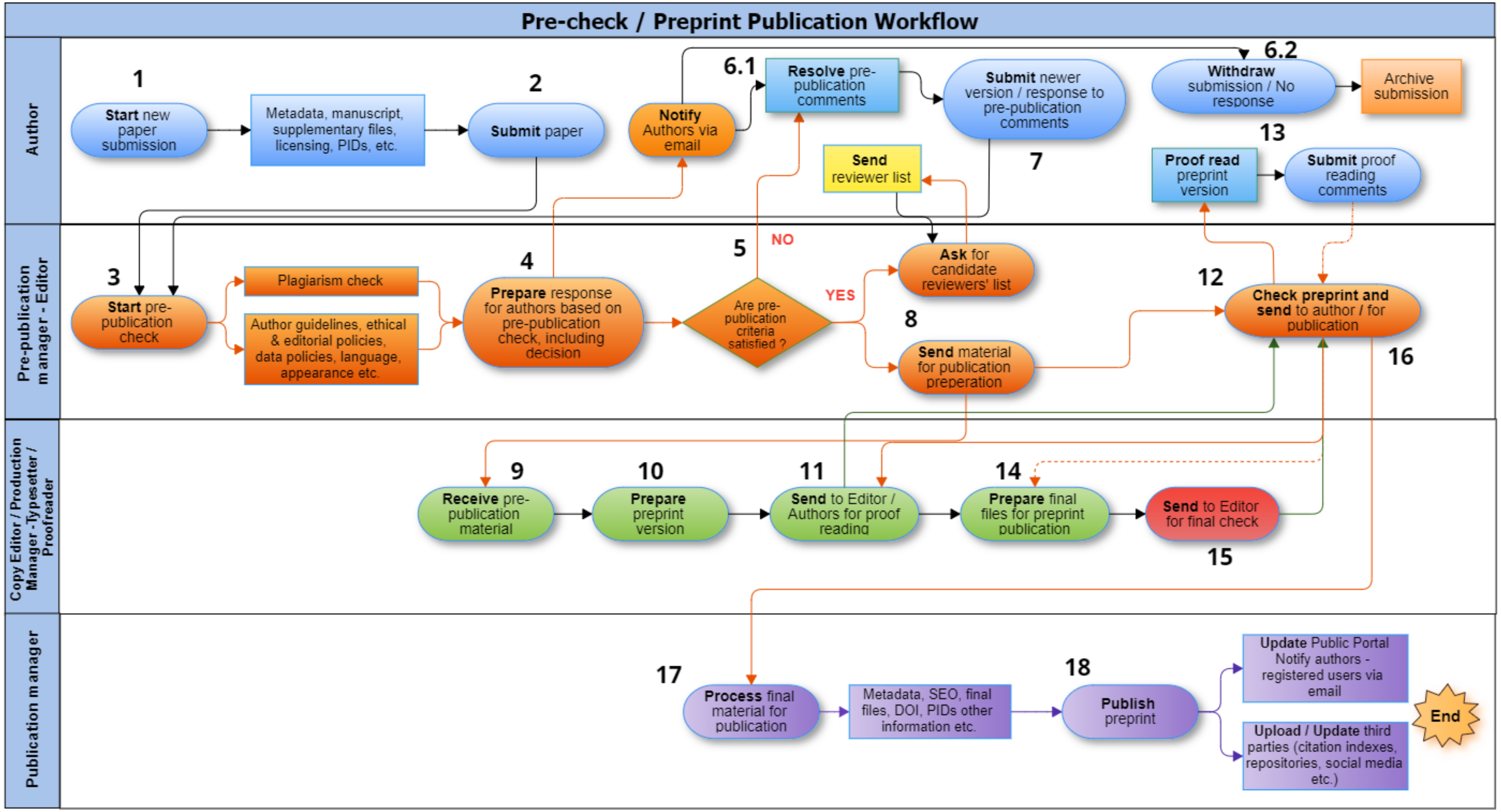


Figure 7. Pre-publication check / preprint workflow

Review workflow

Review workflow steps – Phase 1 (see Figure 8)

Step 1. The Author provides the Editor with the recommended reviewer list.

Step 2. The Editor checks/validates the Author's Reviewer list according to the ORE model guidelines (see **3.4 Major workflows**).

Step 3. If the five potential Reviewers have not successfully been identified as a minimum, the Editor asks the Author to supply additional contacts.

Step 4 (optional). The Editor may utilise the internal Reviewer database to reach the minimum number required. If a Reviewer is selected who was not on the Author's list, they must first be approved by the latter (**Step 5 – optional**).

Step 6. All five Reviewers are notified via email with the request to accept the review assignment.

Step 7. The Reviewer creates an account on the ORE platform and accesses the review assignment details. The Reviewer also has access to the necessary information about the review process and the article under review.

Step 8. The Reviewer accepts or rejects the review assignment. In case of rejection, the review process for the specific reviewer ends at this point. Otherwise, the review process starts if the reviewer agrees with the assignment.

Step 9. The reviewer acquires access to the entire submission material and submits the review, including the decision for the article under assessment ("Approved", "Approved with Reservations", and "Not Approved").

Step 10. The Editor receives the review and accepts or rejects its outcome.

Step 11. Suppose more reviews are needed (at least two reviews are necessary). In that case, the Editor can send reminders to the already selected Reviewers or start a new process of obtaining more Reviewer contact details from Authors (back to **Step 3**).

Step 12. If two valid reviews are submitted, the Editor sends the review results (plus the preprint version) to the production team.

Steps 13 and 14. The Copy Editing / Production / Typesetting / Proofreading team receives the material and starts preparing the preprint version of the article, including the review results (comments from the reviewers).

Step 15. The production team sends the updated preprint version to the Editor for a last check.

Step 16. The Editor checks the new version of the preprint and sends it to Publication Manager.

Step 17. The Publication Manager processes the material for publication (e.g., add metadata, information related to Search Engine Optimisation, DOIs, other PIDs, etc.).

Step 18. The preprint version and the review comments are published on the public portal. Authors, Reviewers and Registered Users (according to their SDI service settings) are notified via email with the updates. Third-party preprint servers (optional) and selected citation indexes (e.g., Google Scholar) are also informed of the updates. Especially, If the review outcome is "Approved," then the article is indexed by all collaborating citation indexing services, taking into account its subject area (PubMed, PubMed Central, Europe PMC, Scopus, ERIH+[49], British Library, CrossRef, DOAJ, and IET Inspec).

Review workflow steps – Phase 2 (see Figure 9)

Step 1. The Author is notified about the review results. Depending on the review outcome, the Author decides if a new version (revision) of the article is needed, including a response to comments raised by the reviewers. For

[49] See here https://kanalregister.hkdir.no/publiseringskanaler/erihplus/

example, if the review's outcome is "Approved" or the comments are of minor significance, the Author may decide there is no need for a revised version of the article. Otherwise, the Author prepares a revised version of the article, accompanied by the response to reviewers' comments and submits them to the Editor.

Step 2. The Editor checks the revised version and the response to the Reviewer's comments and sends them to the production team.

Steps 3 and 4. The Copy Editing / Production / Typesetting / Proofreading team receives the revised article and starts preparing a new version, including the review comments and the Author's response.

Step 5. The revised version of the article is sent to the Editor for a check. The Editor checks the revised version, and the production team is notified in case of any errors. Finally, the Editor sends the revised version to the Author for final proofreading.

Step 6. The Author proofreads and submits comments/remarks based on the findings.

Step 7. The Production Team receives the authors' comments, and the final version of the revised article is prepared. The Editor is also notified.

Step 8. The Production Team sends the final version to the Editor for a last check and publication.

Step 9. The Editor forwards the revised version to the Publication Manager.

Step 10. The Publication Manager processes the material for publication (e.g., add metadata, information related to Search Engine Optimisation, DOIs, other PIDs, etc.).

Step 11. The revised version, the review comments, and the Author's response to them are published at the public portal. Authors, reviewers and registered users (according to their SDI service settings) are notified via email with the updates. Additionally, all third-party services are also informed of the updates.

Step 12. Reviewers are notified that a revised version of the manuscript has been published. Consequently, reviewers are encouraged to re-review if they had initially given an 'Approved with Reservations' or 'Not Approved' status. They are asked to assess whether the work has been sufficiently improved to achieve better approval status.

Step 13. The Editor receives the review for each reviewer and accepts or rejects the outcome.

Step 14. If two valid reviews are submitted, the Editor sends the review results to the production team.

Steps 15. The Copy Editing / Production / Typesetting / Proofreading team receives the material and performs **Steps 13 to 16** of the *review workflow - phase 1*.

Steps 10 and 11. The Publication Manager receives the new version of the article (revised versions, initial reviews and author response, second round of review) and publishes it.

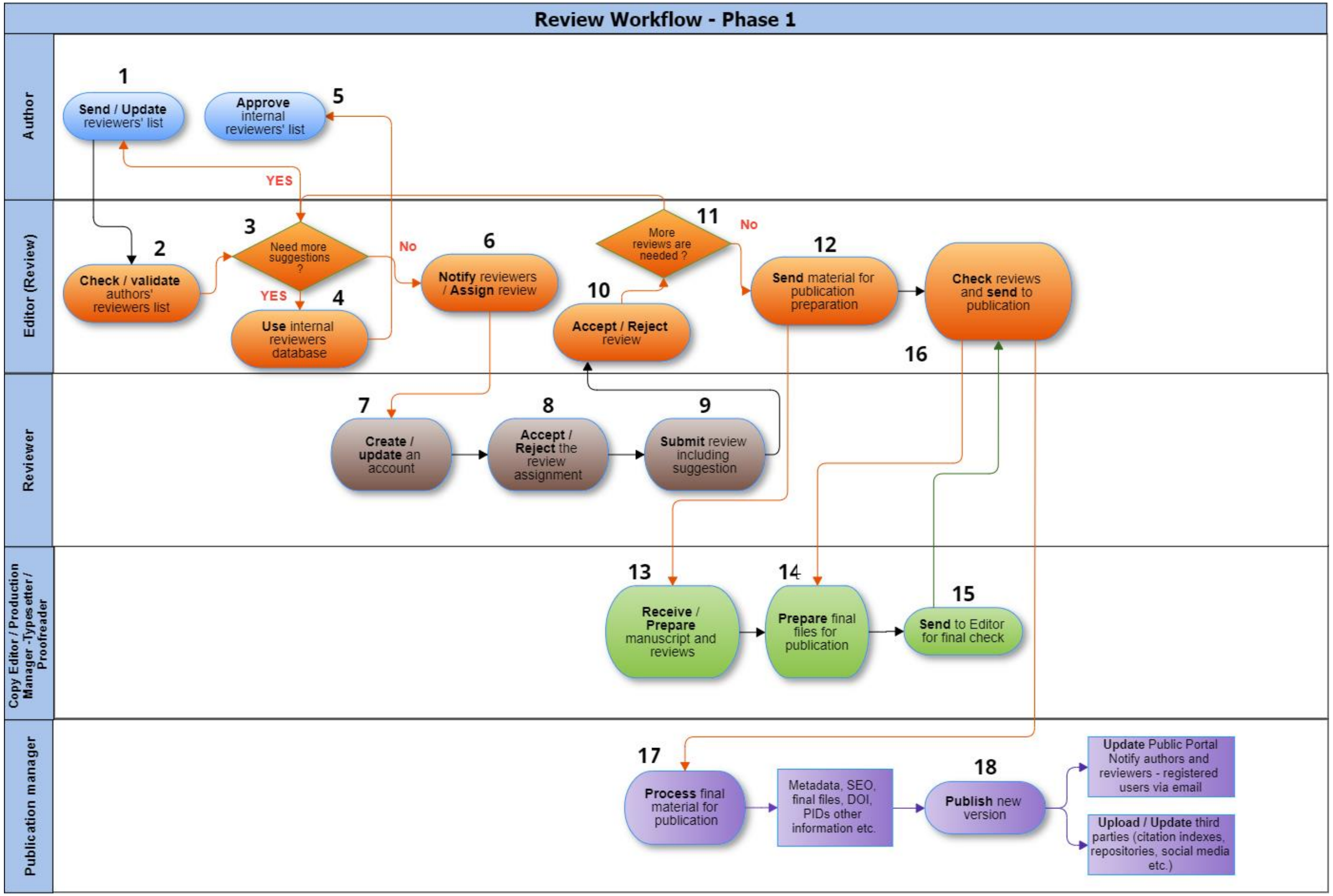


Figure 8. Review workflow (Phase 1)

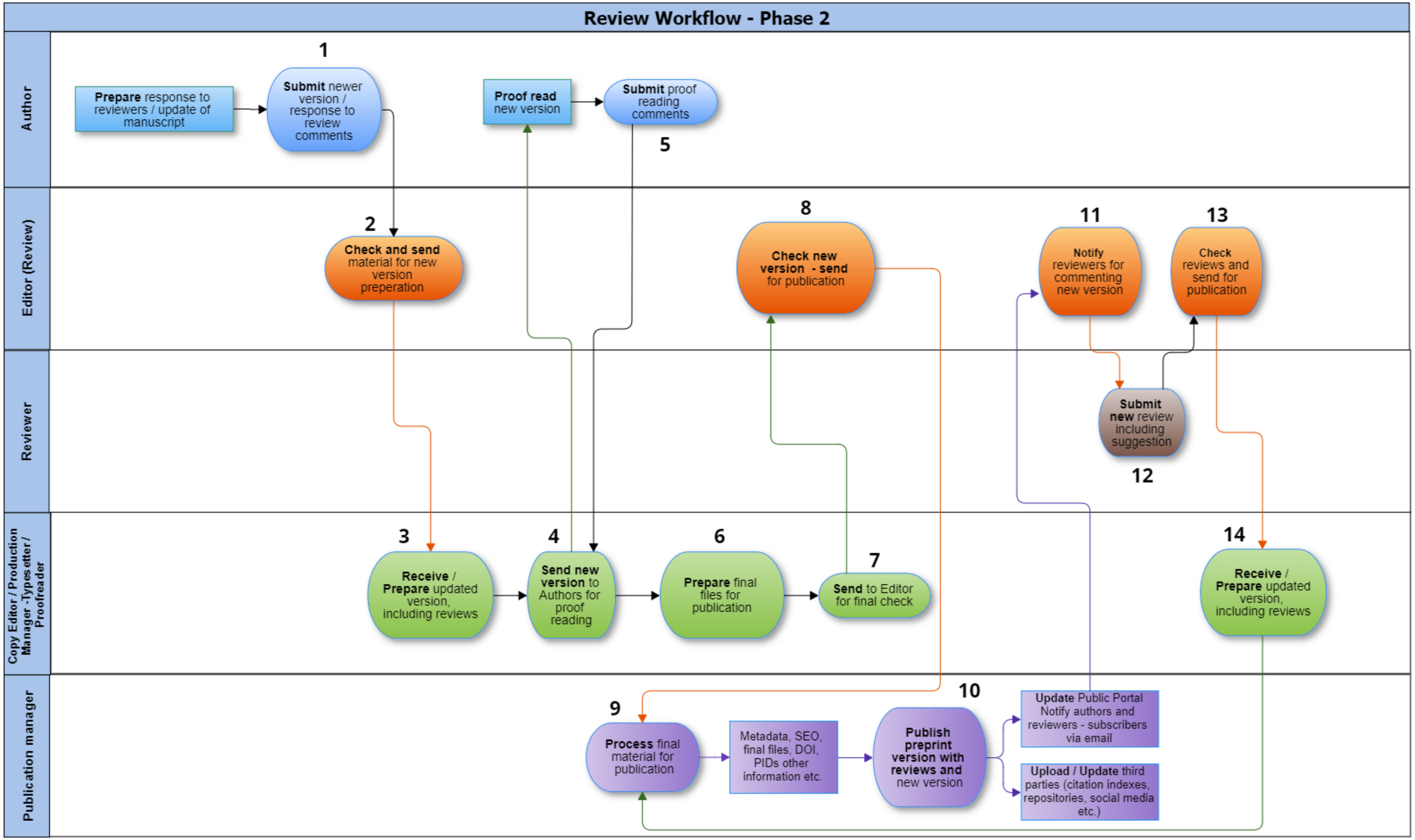


Figure 9. Review workflow (Phase 2)

Simplified workflow – Use of external service providers

Simplified workflow – Use of external service providers/steps for pre-publication check (see Figure 10)

Step 1. The Author starts a new paper submission process. All necessary information is typed in a submission form (article's metadata, manuscript file, supplementary files, licensing options, PIDs, etc.). Submission form elements may differ according to each subject area's needs.

Step 2. The Author finally submits the article.

Steps 3 and 4. Pre-publication Manager (or Editor) receives a notification and assigns the pre-publication check to an external service provider

Step 5. The Pre-publication external service provider performs the task, including communicating with the Author (**Step 6**).

Step 6. The Author works on resolving pre-publication comments and submits the response to the external service provider.

Step 7. The Pre-publication external service provider sends the check results to the Pre-Publication Manager.

Step 8. The pre-publication check results are checked and accepted by the Pre-Publication Manager.

Step 9. In this step, two parallel tasks are initiated. The first task is drafting the candidate reviewer's list with the authors. The second task is sending all necessary material to an external service provider for the ready-to-publish preprint preparation.

Steps 10 and 11. The external service provider receives the pre-publication material, prepares the preprint version of the article and sends it to the Author and the Pre-publication Manager (and the Editor) for proofreading.

Step 12. The Author performs proofreading and, based on the findings, submits comments/remarks.

Steps 13 and 14. The external service provider prepares the final version and sends it to the Editor for a last check and publication.

Step 15. After checking the final preprint version, the Editor sends it to the Publication Manager.

Step 16. The Publication Manager processes the material for publication (e.g., add metadata, information related to Search Engine Optimisation, DOIs, other PIDs, etc.).

Step 17. The preprint version is published at the public portal. Authors and registered users (according to their SDI service settings) are notified via email. The preprint is uploaded to third-party preprint servers (optional), and selected citation indexes (e.g., Google Scholar) are updated with the new publication. Social media and other means of dissemination are automatically getting also updated.

Step 18. The Editor sends all the necessary material for the review, including the reviewers' list, to an.

Step 19. The external service provider responsible for the review performs the process and sends the results to the Editor.

Step 20. The Editor receives the review and accepts or rejects its outcome. If two valid reviews are received, the Editor sends the review results (plus the preprint version) to the external service provider responsible for preparing the ready-to-publish manuscript.

Step 21. The external service provider receives the review material, prepares the final version, and sends it to the Editor for a last check and publication.

Step 22. The Editor checks the new version of the preprint and sends it to Publication Manager.

Step 23. The Publication Manager processes the material for publication (e.g., add metadata, information related to Search Engine Optimisation, DOIs, other PIDs, etc.).

Step 24. The preprint version plus the review comments are published at the public portal. Authors, Reviewers and Registered Users (according to their SDI service settings) are notified via email with the updates. Third-party preprint servers (optional) and selected citation indexes (e.g., Google Scholar) are also informed of the updates. Especially, If the review outcome is "Approved," then the article is indexed by all collaborating citation indexing services, taking into account its subject area (PubMed, PubMed Central, Europe PMC, Scopus, ERIH+, British Library, CrossRef, DOAJ, and INSPEC).

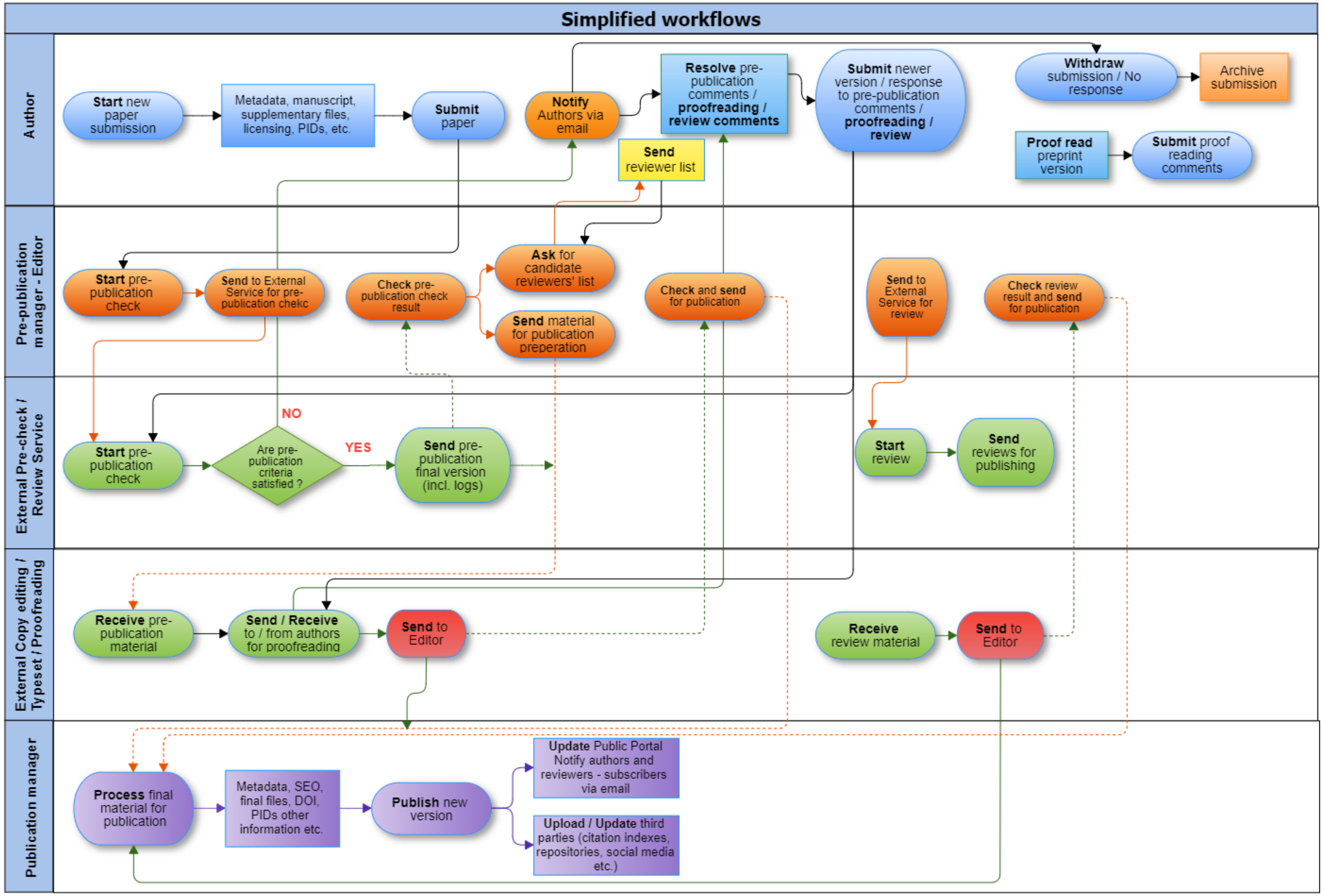


Figure 10. Simplified workflow – Use of external service providers

Manuscript preparation workflow

Manuscript preparation workflow steps (see Figure 11)

Step 1. The Editor sends the to the Copy Editor the manuscript material.

Step 2. The Copy Editor performs the copy editing and uploads the produced files on the publishing platform. The Editor gets notified about the copy editing process completion. The Copy Editor (or the Editor) sends the manuscript's material (after copy editing) for typesetting to the Typesetter.

Step 3. The Typesetter performs the typesetting and uploads the produced files on the publishing platform. The Editor gets notified about the typesetting process completion. The Typesetter (or the Editor) sends the manuscript's material (after typesetting) for proofreading.

Step 4. The Proofreader proofreads the manuscript and uploads the produced files on the publishing platform. If it is necessary, corrections are applied by the Typesetter. The Editor gets notified about the proofreading process completion.

Steps 5 and 6. The Editor checks the ready-to-publish manuscript and sends it to the authors for final proofreading. If necessary and before the proofreading by the Author, corrections are applied by the Typesetter upon request from the Editor.

Step 7. The Author performs proofreading and submits comments/remarks based on the findings to the Editor / Production team.

Steps 8 and 9. The authors' comments are received by the Editor and sent to the production team (Typesetter)

Step 10. The preparation of the final version of the manuscript is performed.

Step 11. The Typesetter sends the final version of the manuscript to the Editor for a last check and publication.

Step 12. The Editor forwards the final version of the manuscript to the Publication Manager.

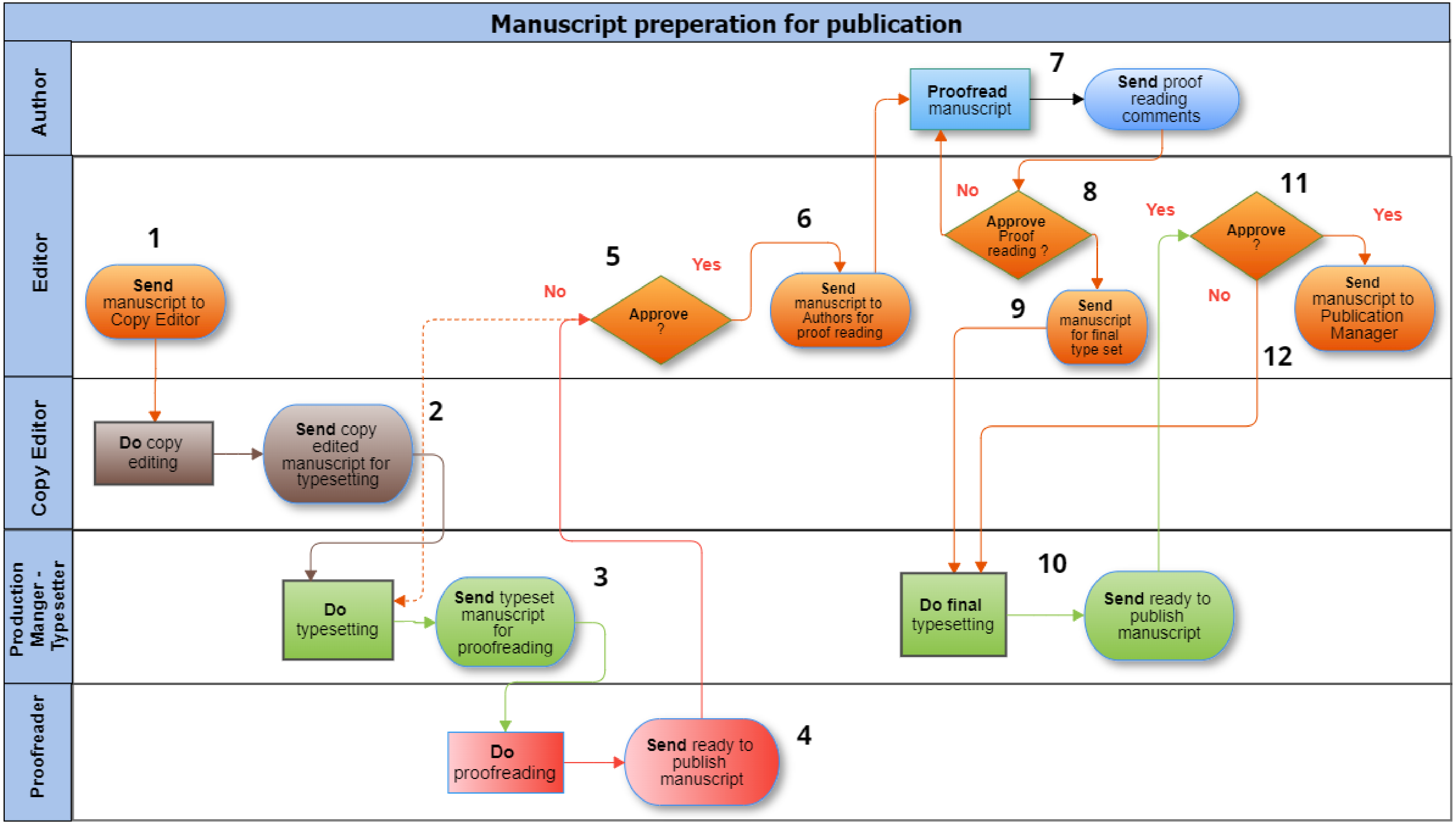


Figure 11. Manuscript preparation for publication workflow

ACRONYMS

APIs	Application Programming Interfaces
CMS	Content Management System
COAR	Confederation of Open Access Repositories
C-OSMM	Capgemini Open-Source Maturity Model
COTS	Commercial Off-The-Shelf
DBs	Databases
DG RTD	European Commission’s Directorate-General for Research and Innovation
DOAJ	Directory of Open Access Journals
DOI	Digital Object Identifier
EOSC	European Open Science Cloud
ERIH	European Reference Index for the Humanities
Europe PMC	Europe PubMed Central
FAIR	Findable, Accessible, Interoperable, Reusable
FAQs	Frequently Asked Questions
GDPR	General Data Protection Regulation
GPLv2	GNU General Public License
HTML	Hypertext Markup Language
IET Inspec	The Institute of Engineering and Technology Inspec databases
IT	Information and Technology
JATS	Journal Article Tag Suite
KFG	Knowledge Futures Group
LAMP	Linux, Apache, MariaDB / MySQL, PHP
LOCKSS	Lots Of Copies Keep Stuff Safe
MFA	Multi-Factor Authentication
N-OSMM	Navica Open-Source Maturity Model
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
OJS	Open Journal Systems
OLH	Open Library of Humanities
Open BRR	Open Business Readiness Rating
OPR	Open Peer Review
ORCID	Open Researcher and Contributor IDentifier
ORE	Open Research Europe
OSS	Open-Source Software

PDF	Portable Document Format
PKP	Public Knowledge Project
PRC	Publish-Review-Curate
QSOS	Methodology of Qualification and Selection of Open-Source Software
RSS	Really Simple Syndication
SciELO	Scientific Electronic Library Online
SDI	Selective Dissemination of Information
SEO	Search Engine Optimisation
SSO	Single sign-on
URL	Uniform Resource Locator
WoS	Web of Science
WYSIWYG	what you see is what you get
XML	extensible markup language

REFERENCES

Adewumi, A., Misra, S., Omoregbe, N., Crawford, B., & Soto, R. (2016). A systematic literature review of open source software quality assessment models. *SpringerPlus*, *5*(1), 1936–1936. https://doi.org/10.1186/s40064-016-3612-4

Chatterjee, A. (2017). Selective Dissemination of Information. In *Elements of Information Organization and Dissemination* (pp. 117–123). Elsevier. https://doi.org/10.1016/B978-0-08-102025-8.00008-9

European Commission. Directorate General for Research and Innovation. (2022). *Operationalising Open Research Europe as a collective publishing enterprise.* Publications Office. https://data.europa.eu/doi/10.2777/061886

Huskisson, M. (2023, February 16). *Guest Post – Scholarly Publishing as a Global Endeavor: Leveraging Open Source Software for Bibliodiversity*. The Scholarly Kitchen. https://scholarlykitchen.sspnet.org/2023/02/16/guest-post-scholarly-publishing-as-a-global-endeavor-leveraging-open-source-software-for-bibliodiversity/

Hyde, A., Pattinson, D., & Shannon, P. (2022). Designing for Emergent Workflow Cultures: ELife, PRC, and Kotahi. *Commonplace*. https://doi.org/10.21428/6ffd8432.ef6691ea

Khanna, S., Ball, J., Alperin, J. P., & Willinsky, J. (2022). *Recalibrating the Scope of Scholarly Publishing: A Modest Step in a Vast Decolonization Process*. SciELO Preprints. https://doi.org/10.1590/SciELOPreprints.4729

Khatri, S. K., & Singh, I. (2016). Evaluation of open source software and improving its quality. *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 114–119. https://doi.org/10.1109/ICRITO.2016.7784936

Linåker, J., Papatheocharous, E., & Olsson, T. (2022). How to characterize the health of an Open Source Software project? A snowball literature review of an emerging practice.

Proceedings of the 18th International Symposium on Open Collaboration, 1–12. https://doi.org/10.1145/3555051.3555067

Randone, L. (2019). *OJS and Lodel possible Interoperability within Open Access Publications: Some indications from Turin-based Italian Journals Use Cases*. Zenodo. https://doi.org/10.5281/ZENODO.2536598

Umm-e-Laila, Zahoor, A., Mehboob, K., & Natha, S. (2017). Comparison of open source maturity models. *Procedia Computer Science*, *111*, 348–354. https://doi.org/10.1016/j.procs.2017.06.033

W Maxwell, J., Hanson, E., Desai, L., Tiampo, C., O'Donnell, K., Ketheeswaran, A., Sun, M., Walter, E., & Michelle, E. (2019). *Mind the Gap: A Landscape Analysis of Open Source Publishing Tools and Platforms* (1st ed.). PubPub. https://doi.org/10.21428/6bc8b38c.2e2f6c3f

GETTING IN TOUCH WITH THE EU

In person

All over the European Union there are hundreds of Europe Direct centres. You can find the address of the centre nearest you online (european-union.europa.eu/contact-eu/meet-us_en).

On the phone or in writing

Europe Direct is a service that answers your questions about the European Union.
You can contact this service:

- by freephone: 00 800 6 7 8 9 10 11 (certain operators may charge for these calls),
- at the following standard number: +32 22999696,
- via the following form: european-union.europa.eu/contact-eu/write-us_en.

FINDING INFORMATION ABOUT THE EU

Online

Information about the European Union in all the official languages of the EU is available on the Europa website (european-union.europa.eu).

EU publications

You can view or order EU publications at op.europa.eu/en/publications. Multiple copies of free publications can be obtained by contacting Europe Direct or your local documentation centre (european-union.europa.eu/contact-eu/meet-us_en).

EU law and related documents

For access to legal information from the EU, including all EU law since 1951 in all the official language versions, go to EUR-Lex (eur-lex.europa.eu).

EU open data

The portal data.europa.eu provides access to open datasets from the EU institutions, bodies and agencies. These can be downloaded and reused for free, for both commercial and non-commercial purposes. The portal also provides access to a wealth of datasets from European countries.

Open Research Europe (ORE) is the peer-reviewed open-access publishing platform of the European Commission. It follows the post-publication peer review model to promote scientific transparency and reuse. The Commission plans to develop an infrastructure to underpin ORE in the future that is based on open source software following the open-source code use and distribution model.

The present analysis was commissioned to determine if open-source software (OSS) solutions can be used as a foundation for developing the new publishing platform and to document the necessary workflows and functionalities of the new platform.

After conducting a thorough analysis, it has become evident that utilizing existing open-source software has its own advantages and disadvantages. Although some risks are associated with this approach, our research has identified a few mature existing solutions that could be further developed to support the future ORE platform.

Studies and reports

Publications Office
of the European Union