**COPIM • Books Contain Multitudes: Exploring Experimental Publishing
(2022 update)**

Part 3: Technical Workflows, Tools, and Platforms for Experimental Publishing, Interaction, and Reuse of Books

Tobias Steiner¹, Marcell Mars, Simon Bowie¹, Janneke Adema¹

¹Centre for Postdigital Cultures, Coventry University

Published on: Apr 26, 2022

DOI: 10.21428/785a6451.174760b2

License: [Creative Commons Attribution 4.0 International License \(CC-BY 4.0\)](https://creativecommons.org/licenses/by/4.0/)

Table of Contents

Table 1
<u>Review and Analysis of Key Studies and Resources</u>
<u>Proposed Methodology for an Online Resource to Support Experimental Publishing</u>
<u>Categorising Tools: On ‘Open Source’ Tools</u>
<u>Collaborative Writing Tools</u>
— <u>Pads & platforms: EtherPad, CryptPad, CodiMD/Hedgedoc</u>
— <u>Git-based Collaboration</u>
<u>Annotation Tools</u>
— <u>Overview of annotation-specific standalone tools</u>
— <u>Platform-agnostic overlay annotation tools</u>
— <u>“Standalone” fixed-ecosystem annotation tools</u>
<u>Remix and Adaptation</u>
— <u>Licensing and Copyright</u>
— <u>Alternative Licensing Models: CopyLeft, CopyFarLeft, CopyFair, CC4R, Traditional Knowledge</u>
<u>Semantic Web, Linked Data, Translatability, Citations</u>
<u>On Re-Using Third-Party Material in your Research Publication</u>
<u>Versioning and Forking of Books</u>
<u>Computational Publishing Tools</u>
— <u>Examples of Computational Books</u>
<u>The ExPub Compendium</u>
<u>Conclusion</u>

Works Cited

For this third part of the scoping report, we will be looking at the technical developments around experimental book publishing. We will be doing so in a three-fold manner in these subsequent sections. First, instead of conducting a landscape study ourselves, we will be reviewing a number of studies and resources that have recently been released and that have tried to categorise, analyse, and map the open source publishing tools and platforms currently available to support open access (book) publishing. Our focus in this analysis will predominantly be on those tools and technologies that can support the kinds of experimental publications that we have identified in the first two parts of this scoping report. With the current version of the report updated in 2022, we have substantially expanded this third part and included segments that had previously been covered in our *Promoting and Nurturing Interactions with Open Access Books: Strategies for Publishers and Authors* report ([Adema et al., 2021](#)), while also adding new examples and information that further research has revealed over the past year.

Secondly, in [section 2](#), we will outline a proposed methodology to analyse and categorise the currently available tools and technologies to support the creation of an online resource or Compendium for publishers and authors in year 3 of the COPIM project. This Compendium will include the technological support and workflows available to enable more experimental forms of book publishing, whilst showcasing examples and best practices for different levels of technical know-how.

Thirdly, in [section 3](#), we will make an initial attempt at categorising a selection of tools following this proposed methodology, where we will be focusing on collaborative writing tools and on annotation tools—and the software, platforms, and workflows that support these—in first instance. The choice for these tools is driven by the [pilot projects](#) we are supporting as part of the COPIM Experimental Publishing and Reuse Work Package, which focus on several experimental practices, including collaborative writing, annotation, remix, versioning, open peer review, and computational publishing.

Review and Analysis of Key Studies and Resources

Maxwell, J. W., 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*. PubPub. <https://doi.org/10.21428/6bc8b38c.2e2f6c3f>

The first resource or environmental scan we looked at was the *Mind the Gap* report, conducted by John Maxwell et al. at Simon Fraser University in Vancouver on behalf of the MIT Press after they secured a grant from the Mellon foundation in 2018. As they state in the report, the award was to

‘conduct a landscape analysis of open source publishing systems, suggest sustainability models that can be adopted to ensure that these systems fully support research communication and provide durable alternatives to complex and costly proprietary services.’ (2019).

As they note, the last few years have seen an increase in the number of open source publishing platforms (many well-developed, stable, and supported) or, in other words, production and hosting platforms for both scholarly books and journals. The report argues that this is evidence of an infrastructure ‘ecology’ emerging which includes complementary, non-competitive service technologies instead of proprietary and often bespoke software systems. This is of particular relevance for our work with COPIM, as

‘at a more ambitious level, they may even form a layer of community infrastructure that rivals—or at least provides a functional alternative—to the commercial infrastructure run by a small number of for-profit entities’ (2019).

Mind the Gap provides a guidebook through this proliferating yet noisy landscape, as they work to help ‘the university press community and other mission-focused enterprises’ (2019) with decision-making and project planning. Next to being a catalogue of open source publishing tools, the report also examines the ecosystem in which these tools and projects exist. The element of community infrastructure and interoperability is key here, as a ‘system in which these components can be mobilized to serve larger goals’ (2019).

Part II of the report serves as a catalogue of open source publishing projects. For each open source project, Maxwell et al. provide a summary description plus details on the host organisation, the project's principal investigator or leadership, funders, partners (both strategic and development), date of original release, and current version, plus some basic data drawn from the projects’ Github/Gitlab repositories, including development language, license, and number of contributors. As part of their methodology, they looked at tools and projects that were ‘available, documented open

source software relevant to scholarly publishing’ and that ‘were ‘still alive’—that is, with evidence of active development’ ([2019](#)). They emphasise however that this is a dynamic space, and that their cataloguing is a snapshot of a specific moment in time. As such, Maxwell et al.’s analysis is not only based on individual tools but on a consideration of the dynamic landscape as a whole. Their categorising is mainly based on exclusion, where they did not include tools and projects that were closed-source, cloud-based services, research (instead of publishing) tools, library infrastructure, DIY ad-hoc toolchains, and dormant projects.

The key themes that informed their research were sustainability, scale, collaboration, and ecosystem integration. One key research question was ‘who will care about these projects?’ In other words, ‘care enough to fund, contribute, promote, use, and ultimately further their useful life? What are the values and mechanisms that cause people—especially external stakeholders—to care enough about these projects to keep them alive, and even thriving, going forward?’ ([2019](#)). The gap that they have noticed as part of their research is one of co-ordination and integration between and among projects. In other words, there is a lack of interoperability and incentives for collaboration between projects.

In Maxwell et al.’s mapping of the tools and projects they emphasise a few main characteristics:

- Difference between journal publishing and book publishing
- Centralised vs distributed models
- Old projects and new projects
- Functional scope (i.e., development across hypothetical workflow stages)
- Operational details (development features, languages and frameworks, licenses, and funding)
- Traditional functions vs. new capacities (i.e., interactive scholarly works)
- Technological approaches and trends (approaches to XML, conversion and ingestion strategies)
- Workflow modeling and management
- Innovating new possibilities

Key findings were issues of:

- ***Siloed development***, with the recommendation that ‘where possible, collaboration, standardization, and even common code layers can provide considerable benefit to project ambitions, functionality, and sustainability’ ([“Prospects,” pp. 20-1](#)).

- *The organisation of **the community-owned ecosystem*** itself, where the recommendation is that ‘neither a chaotic plurality of disparate projects nor an efficiency-driven, enforced standard is itself desirable, but mediating between these two will require broad agreement about high-level goals, governance, and funding priorities—and perhaps some agency for integration/mediation’ ([“Prospects,” pp. 20-1](#)).
- **Funding**, where the question was ‘what would project funding look like if it prioritized community governance, collaboration, and integration across a wider ecosystem?’ ([“Prospects,” p. 24](#)).
- **Longevity and maintenance**, with the recommendation that ‘if the care and upkeep of projects could be extended to multiple groups, multiple institutions, then not only is there a larger and more diverse set of people who care, but opportunities for resourcing increase, and also, when one group’s priorities inevitably shift, it is less likely that a project is simply abandoned’ ([“Prospects,” p. 24](#)).
- **Ecosystem integration**, with the reminder that ‘if the goal of community-owned infrastructure is to succeed, then structural attention needs to be paid to the integration of projects, goals, and development efforts across the ecosystem’ ([“Prospects,” p. 24](#)).
- Whether we need **centralised or distributed options, or a tertiary service provider**? With the recommendation that ‘if longer-term funding for sustainability is needed, then a mediating layer might productively function as a broker of such funding, assuming overhead costs remain low’ ([“Prospects,” p. 28](#)).
- **Scale**, where almost all of the projects they examined are too small, niche or specialised to be sustainable on their own. Additional funding will be needed.
- **The importance of trust** in open scholarly communication, which presents challenges for scalability. Recommendation that ‘community coordination may go some distance towards addressing this [issue]’ ([“Prospects,” p. 28](#)).

Lewis, D. W. (2020). *A Bibliographic Scan of Digital Scholarly Communication Infrastructure* / Educopia Institute. Educopia Institute. <https://educopia.org/mapping-the-scholarly-communication-landscape-bibliographic-scan/>

The second resource we looked at is a Bibliographic Scan by David W. Lewis on behalf of the Educopia Institute. The blurb accompanying this resource summarises its aims quite well:

This Bibliographic Scan by David W. Lewis provides an extensive literature review and overview of today’s digital scholarly communications ecosystem, including information about 206 tools, services, and systems that are instrumental to the

publishing and distribution of the scholarly record. The Bibliographic Scan includes 67 commercial and 139 non-profit scholarly communication organizations, programs, and projects that support researchers, repositories, publishing, discovery, preservation, and assessment.

The review includes three sections: 1) Scholarly citations of works that discuss various functional areas of digital scholarly communication ecosystem (e.g., Repositories, Research Data, Discovery, Evaluation and Assessment, and Preservation); 2) Charts that record the major players active in each functional area; and 3) Descriptions of each organization/program/project included in the Bibliographic Scan. This work has been produced as part of the “[Mapping the Scholarly Communication Infrastructure](#)” project (Andrew W. Mellon Foundation; Middlebury College, 2018-20).

The second and third part of the report list and describe projects, programs, and products (as well as listing some key literature on these), and categorises them according to Researcher Tools (Reading, Writing, Annotation, and Collaboration), Repositories, Publishing, Discovery, Evaluation and Assessment, Preservation, and General Services. This categorisation also indicates whether the organisation hosting the project or product is non-profit (NP) or for-profit (P).

Confederation of Open Access Repositories (COAR), & Next Generation Libraries Publishing. (2021). *SComCaT: Scholarly Communication Technology Catalogue*. <https://www.scomcat.net/>

The third resource we looked at is the *Scholarly Communication Technology Catalogue* (*ScomCat*), a catalogue or database of open tools, platforms, and technologies that identifies relationships and dependencies between them. Developed by [Antleaf](#) for the [Confederation of Open Access Repositories \(COAR\)](#) as part of the [Next Generation Libraries Publishing project](#), the catalogue maps these technologies according to adoption levels, functions, categories, governance, and readiness. This catalogue has now been made openly available since January 2021. Our thanks go out to the Next Generation Libraries Publishing Project for sharing the early catalogue-in-progress version with us. From the catalogue’s home page:

SComCat comprises a catalogue (knowledge base) of scholarly communication open technologies where the term "technologies" is defined to include software and some essential running services. The aim is to assist potential users in making decisions about which technologies they will adopt by providing an overview of

the functionality, organizational models, dependencies, use of standards, and levels of adoption of each technology.

The scan includes tools, platforms, and standards that can be locally adopted to support one or more of functions of the lifecycle of scholarly communication, which is conceptualized as including the following activities: creation, evaluation, publication, dissemination, preservation, and reuse. ([COAR & NGLP, 2021](#))

Radical Open Access Collective. (n.d.). *Information Portal: OA Publishing Tools*. <https://radicaloa.disruptivemedia.org.uk/resources/publishing-tools/>

The fourth resource we looked at is the Radical Open Access Collective's Information Portal, which includes a list of Open Access Publishing Tools. This page contains a list of open source tools, software, and platforms for scholar-led approaches to open access publishing. It lists all-in one platforms or services as well as more targeted solutions. It provides descriptions of the tools and links to their home pages and to other resources related to the tools or platforms.

Kramer, B., & Bosman, J. (n.d.). *400+ Tools and innovations in scholarly communication*. Google Docs. <https://bit.ly/innoscholcomm-list>

The fifth resource is a shared crowd-sourced database of tools and technologies in scholarly communications, that grew out of the "101 innovations in scholarly communication" project led by Bianca Kramer and Jeroen Bosman at Utrecht University in the Netherlands. As they explain:

When we published the 101 list of selected innovations our database already contained some 200 innovations/tools. The 101 selection was strictly on innovativeness and thus did not contain recent tools if they were not innovative compared to older ones with the same functionality, even if the more recent ones were more popular or well-known. The database shared here has dropped that strict innovativeness criterion and thus contains multiple tools offering basically the same functionality. ([Kramer & Bosman, n.d.](#))

Tools are identified by workflow phase (preparation, discovery, analysis, writing, publication, outreach, assessment) and short descriptions of each tool are provided.

Tennant, J. P., Bielczyk, N., Tzovaras, B. G., Masuzzo, P., & Steiner, T. (2020). Introducing Massively Open Online Papers (MOOPs). *KULA: Knowledge Creation, Dissemination, and Preservation Studies*, 4(1), 1. <https://doi.org/10.5334/kula.63>

This sixth resource is included here due to its approach to identifying and discussing common traits of collaborative writing tools: while the main focus of “Introducing Massively Open Online Papers (MOOPs)” is on ‘collaboratively author[ing] research articles in an openly participatory and dynamic format’ (Tennant et al., 2020), the workflows that are explored in the paper and the steps taken to identify common features to evaluate a variety of tools along a set of predefined criteria (see the paper’s Table 2) that are posited as user requirements for collaborative writing platforms, are introduced here in a concise fashion that warrants further adoption and expansion to fit the needs of experimental book publishing.

Categories introduced by this paper that might also inform our discussion of experimental publishing tools (Authorea, CryptPad, Google Docs, Overleaf, HackMD¹) include:

- Sustainability² model (FLOSS (open source, self-hostable), freemium [basic functionality for free, premium add-ons], proprietary but free-to-use (via user account/login)).
- Based on Open Source platform (yes, no - open repository of software code available).
- Option to export to open formats, (if yes, which kind of output format - markdown, git, Word, Open Document Text, html).
- Interactive multi-user collaboration (commenting, editing, etc.) .
- Integration of Reference Management solutions (i.e., using Zotero and other RefManager tools with your collaborative writing tool).
- Predefined Formatting / Layout styles to fit journal house styles where possible.

Proposed Methodology for an Online Resource to Support Experimental Publishing

In year 3 of the COPIM project, we are delivering an online resource to support authors and publishers in publishing more experimental long-form works. As part of this research and scoping report, we propose a methodology or a set of methodologies to support the development of this resource, which we hope will become community-maintained in the future. By publishing this report and updates to it, we hope to receive further feedback from publishers, authors, technologists, and platform providers on this proposed methodology and on the set-up and usefulness of the online resource. We then hope to be able to incorporate this feedback to further develop and

fine-tune the ideas presented in this report over the next couple of years (as part of various updated versions of this report).

The first aspect we are focusing on is identifying those open source tools, platforms, and technologies that are particularly useful for more experimental forms of publishing (because they support the creation of experimental books, for example). In the first instance, we use the resources listed in the previous section to identify those tools that are currently available. As part of our subsequent analysis of these tools we propose the following methodology or set-up for the online resource:

- An **introductory part/glossary** that defines what we mean when we refer to open source tools, and how—within the category of open source tools—one can differentiate between software packages and hosted solutions, and between the commercial, not-for-profit, and other underlying business models (e.g., institutional support) that support these services or platforms.
- A **review of those tools we deem most useful to support the publication of experimental books**. Next to providing a basic description of the tool and its purpose and usage, this review considers collaborative capabilities and features (e.g., synchronous editing, in-document change-tracking and versioning) and its availability as a stand-alone tool and/or platform, while also focusing on the skills level of both publishers and authors, focusing on the technical knowledge required to install and use the tool, software, or platform discussed. In addition to this, the review focuses on the longevity and stability (sustainability) of the tools under review. For example, we explore who is maintaining them under which conditions and in what way, and how many times they have been successfully implemented.
- A **categorisation/tagging of tools according to the main experimental publishing functionalities** we have identified (i.e., annotation, collaborative writing, open peer review, multimodal publishing, versioning, enhancing existing documents). Our aim with this categorisation is to provide authors and publishers with a range of tools to choose from if they are interested in experimenting with, for example, open peer review or multi-modal publishing. But we also want to outline the difference in functionality between tools, and the skills-level required to implement the specific tool in the research or publishing workflow, and show what you can do with the tools based on your skills level. (From a developer's perspective, for example, how easy is it to install and run the tool locally or on a VPS.)
- Work backwards from a few key examples of previously published experimental books to analyse which tools and workflows were used to produce those experimental books (while linking back to potential alternative tools, or new tools or

updates to tools released after the example book was published). This would include user experiences or stories/narratives (where available) about the research and publishing process involved in their creation. In other words, our aim is to map tools and technologies onto real examples of OA experimental books to showcase what you can do with these tools and to show proof of concept.

This proposed methodology comes with certain risks and unknowns that we hope to more clearly map and identify when we request community feedback on this scoping report. These are some of the risks we have identified up to now:

- How to involve the community of technologists, software, and platform providers in the set-up of this online resource (again, as a community-led endeavour), while at the same time being able to provide an assessment / review of the tools discussed as part of the online resource? One way to resolve this is by looking at clear categories to base our assessment on, which can be devised with the aid of the technologists involved.
- How to make sure we adequately capture researchers' and publishers' workflows or are able to suggest software stacks that can be implemented in publishing or research workflows? One of the ways we hope to achieve this is by first of all requesting feedback from the ScholarLed presses involved in the COPIM project; and second of all by requesting feedback from other presses (for example, via workshops and interviews).
- How to ensure the online resource will be maintained after the project ends? As we are keen to develop this online resource from the start as a community-led project, we hope to involve the community of authors and publishers interested in the publishing of experimental books in the set-up of this online resource. We imagine that in the future it can be maintained by a community of volunteers (led by an Advisory Board, for example), or can be integrated in the wider COPIM infrastructural provision, for example as a service connected to the Open Book Collective. As the tools and resources we will be describing and analysing as part of this online resource will be highly dynamic, it is crucial that we design this online resource as a processual endeavour that can easily be updated and maintained by the scholarly and publishing community. As part of the research for this online resource (and in collaboration with the COPIM Governance Work Package) we will be studying the governance of similar projects and resources (such as the [Electronic Literature Directory](#)) that have been able to achieve a certain level of longevity.

Categorising Tools: On ‘Open Source’ Tools

To make a head start on the proposed methodology for an online resource around experimental book publishing described in the previous section, we want to outline both here for this report and for any future work based on our research, some of the principles and concepts that underlie our work, as well as what we feel would be desired aspects for technical workflows to have in the context of experimental book publishing. Similar to Maxwell et al. ([2019](#)), our approach to ‘open source’ is informed by the understanding encapsulated in the (F/L)OSS acronym, i.e., the notion of Free/Libre and Open Source Software that is ‘developed in such a way that its source code is open and available online, and explicitly licensed as such’ ([“Setting Context,” 2019](#)). Hence, we limit our selection to those tools that have been made available as self-hostable packages under the premise of open, permissible licensing (e.g., GPL, Apache 2.0). We also highlight the underlying value system and *modus operandi* chosen by each of the tools so as to make visible the features that may prove conducive for inclusion in a curated selection of such tools, as we seek to do in the COPIM project.

From a historical perspective, it seems pertinent to keep the underlying factions of the struggle to define open software in mind: while the Free/Libre Open Source Software (FLOSS/FOSS) camp has postulated four fundamental freedoms that are governing its value-based proposition, this is not necessarily true for the open source approach to software, which is more occupied with the practical means of software production/development following a ‘bazaar’ model of collaboration ([Raymond, 1998](#)), which in turn does not explicitly enshrine the Free Software movement’s fundamental freedoms.³

Graphical User Interfaces vs Command Line Interfaces

Many interesting experiments happen (both in digital scholarship and publishing) when using and combining different tools together in new ways. If these attempts are successful there is a significant chance the newly introduced (combined) technique will become a feature of existing tools or even a tool in its own right. To encourage scholars and publishers to start experimenting with new digital tools and technologies as part of their research and publishing practices, we want to make the argument that it is productive, from a technical perspective, to understand and capture this process as a sequence of steps, performed by orchestrated human labour and/or software tools, moving from the beginning to the end of a specific work (or research or publishing) process. This is what is commonly called a *workflow*. A workflow’s sequence consists

of distinctive repeatable patterns, and those patterns might overlap throughout authoring and publishing workflows.

Most distinctive operations in the sequence of a workflow are exposed to the user through a user interface. The most popular and wide-spread one is the so-called 'point & click' graphical user interface, with its iconic drop-down menus where one can choose which operation to be performed by the tool.⁴ In general, people know how to point & click in the drop-down menu of Microsoft Word, LibreOffice, or Google Docs, for example, and open a file, select text, apply italic or bold font styling, and save the file in one of the available file formats the tool offers. If we would have to express the level of user expertise needed in order to work with these kinds of tools, we could classify them as 'a regular user.'

Authoring tools such as Microsoft Word, LibreOffice, or Google Docs expect a user to open a certain number of supported input file formats such as ods, doc, md,⁵ and export or save them in, again, a certain number of supported output file formats. Almost everything a user can do in these kinds of tools is supposed to be done manually by pointing & clicking on drop-down or contextual (i.e., right-click on one's mouse/pointing device) menus. If, for example, a user needs to process digital photos, she can use a similar GUI tool such as Photoshop. Following the suggested workflow sequence, she would then open a photo, point & click on menus in Photoshop, and save the graphics into a file format (e.g., jpg, png) that text authoring tools such as MS Word are able to import.

These tools can be used in a sequence of steps and following distinctive patterns of use, but due to the design principles that many of these GUI-based tools follow,⁶ their role in an open workflow potentially involving a set of interchangeable tools/applications is doubtful.

While there is nothing in a graphical user interface that would make a single tool in a workflow less interoperable with other tools, both the evolution of proprietary file format standards and corresponding developments pushed by commercial software companies to make their GUIs uniquely fit their distinguished user group, has led to substantial problems with regards to interoperability that, through years of use of these GUIs by its users, have led to a profound silo-isation of GUI tools.⁷

However, an alternative culture does exist, one mostly built around the so-called 'command line interface', which preceded the GUI era. This culture derives from and is based on decades of development of the Unix operating systems ecosystem. In

summary, this culture’s underlying philosophy states: ‘Write programs that do one thing and do it well. Write programs to work together. Write programs that handle text streams, because that is a universal interface’ ([Salus, 1994, p. 52](#)). In Unix, interoperability is key, where it is expected that the output of one tool (for example the ‘cat’ tool, which outputs text of a given text file) can be used as an input for another tool. This tool’s output could then, again, become the input for yet another tool, a third, fourth or as many tools as one would want to link together in a pipeline of *interoperable* tools to form what is generally called a *toolchain*.

This flexibility comes with a price, however. Not all users are happy or familiar with typing commands into a terminal (aka the ‘command line’), especially when their usual interactions with a computer have been solely mediated through GUI-based desktop applications. The most widely used proprietary desktop operating systems, Microsoft Windows and Apple’s macOS, both obfuscate their terminals from the average user to strongly discourage the use of command-line functions.

However, if one wants to explore experimental research or publishing pipelines, forms of automation such as batch processing—including the automated generation of different output formats from one source format; automated and streamlined layouting along a pre-defined set of rules; and/or massive conversion of files such as the transformation of image files to one compatible format for web publications—would really benefit from command line tools/utilities, which are also often developed years before these kinds of features get implemented in mainstream GUI authoring tools.⁸ As such, research teams or publishing operations that are open to typing lines of commands into the terminal will most likely be able to get things done much quicker.⁹ Command-line based tools such as Pandoc, PDFtk, Xpdf-utils or Sphinx, Jekyll, and Hugo are able to manipulate, extract, convert, and process PDFs, plain text, LaTeX, HTML or Markdown files into all kinds of documents, websites, or publications ready to be served to end users or just passed further down the tools pipeline. To be able to really explore the many possibilities experimental publishing and experimental books can offer, we would therefore always recommend research teams and publishing projects familiarise themselves with the basics of the command line interface.¹⁰

Desired Aspects of Technical Workflows

From a technical perspective, we at COPIM are committed to open source solutions. To accommodate the creation of experimental books in the best way possible, we recommend that any technical research or publishing workflow takes into consideration the following desired aspects:

- The code used within the workflow should be **open source** available in a version control system.^{[11](#)}
- The workflow should be **user friendly** (ideally when working with both command line and graphical user interfaces).
- The workflow should be **easily installable/deployable** in a cross-platform environment (available for a variety of Computer Operating Systems including Linux / Unix, Apple's macOS, Microsoft Windows, Google Android, Apple's iOS, as well as taking different types of platforms such as desktop computers / laptops, mobile phones, tablets, and web servers and cloud services into account),
- The workflow should be **modular**, so that any work done as part of one certain phase/step of the workflow can be re-used further down the pipeline of another compatible workflow. This translates to an operationalisation of steps that can be actioned by (sets of) commands in the CLI to be combined in a modular way.
- The workflow should be **interoperable** and support established standards such as xml-based document formats (ods, odt, xml, epub) or plain text markups such as HTML and Markdown, both for its inputs and outputs. This would be to enable the workflow to follow up on what has already been done in another compatible workflow; or to enable its output(s) to be used as (an) input(s) for another compatible workflow.
- It should be **possible to build distributed services** around/on top of a given workflow, meaning that it:
 - can be installed and run on your own computer/server,
 - can be installed and run as a node in a federated network (such as email infrastructure, the Mastodon social network, PeerTube video delivery, or the XMPP instant messaging protocol),
 - can be installed and run as a node in a peer2peer/mesh network (such as BitTorrent content delivery, the [Tor](#) anonymity network, or the [Freifunk](#) wireless community network).
- A workflow's sources should **remain human-readable** and should **not require idiosyncratic (versions of the) software** in order to use the workflow (i.e., this would be an argument for using Markdown documents over Rich Text formats that tend to bury information relevant for text output in the depths of their xml-based document structure). This would also make source materials easier to archive.
- The workflow should be **collaborative** in either an asynchronous or synchronous way.^{[12](#)}
- The workflow should **track the edits/versions** of who, when, and what changed in a (collaborative) document.

- The workflow should **allow for (interoperable) annotations and/or comments**. This means that, ideally, annotations and/or comments are available as human readable, versioned source materials that include contextual information/metadata about e.g., their relation to the annotated text.
- The workflow should render/transform user input into results/output(s) that manifest in an online and/or offline-ready website, EPUB, PDF or other formats ready to be read, edited, annotated, commented, widely distributed, preserved, archived, and used by other compatible workflows.

We are aware that it will be difficult for any technical workflow to cover or include all of the aspects listed here. In most research and publishing contexts, workflows are chosen based on criteria of speed, ease of use, and availability. Familiar user interfaces therefore have a better chance to be picked up in the first instance (which also explains the continued preference for print-based interfaces and workflows in digital scholarship and publishing). Similarly, through our institutional settings, we have grown accustomed to working with commercial software solutions (e.g., provided by Microsoft, Apple, Google). This is why, for example, interfaces that are similar to Google Docs (often used to support collaborative writing projects) will be the starting point for many collaborative research projects. However, as a piece of software, Google Docs is proprietary, cloud based, not installable/deployable, and hardly modular or interoperable. Still, even the option of being able to export a given document via "Save as" into different formats can present a first step and an entry point to opening up publishing to experiments, as this output can then be used as a starting point to follow-up with workflows that cover more of the desired aspects listed here.

Plenty of alternatives to GoogleDocs exist in the free & open source world. For example, within the COPIM project we use [ONLYOFFICE](#) integrated with our own instance of the file hosting service [Nextcloud](#), an open source alternative to Microsoft's SharePoint. Both projects are open source, interoperable, support established standards, are well integrated, relatively easy to set up and to run on a server. Nextcloud has a fairly modular architecture which has attracted a whole ecosystem of plugins that can address different tasks, among which sits ONLYOFFICE, which follows the familiar paradigm of the Microsoft Office Suite. Experimental books or publishing projects that involve elements of (collaborative) writing and editing, just as is the case in proprietary office suites, will most likely, benefit most from the possibility to save their outputs in a variety of output formats, giving them the flexibility to incorporate that output into another (follow up) workflow again.

Some of the desired workflow aspects listed previously are only achievable if they are set up, ran, and maintained by publishers or researchers who have a certain (minimal) level of computer literacy and skills (which is often lacking, as Adema and Stone have shown ([2017](#))). But for some of these steps only a few basic tweaks to software settings are needed to achieve the desired set up or results. In some cases, as explained, this involves being familiar with a command line interface (including reading the documentation about option flags which should be added to the software in order to make it do something specific, for example).

If publishers or researchers are able to connect to a server via SSH and to edit in the server's shell (configuration) text files or if they can run command line tools on their desktop computer, a lot more options for experimental work are opened up and become possible. We feel that these basic skills, together with the openly available documentation that accompanies many of the tools and technologies we will discuss in this report, should be enough for authors and publishers to experiment with these tools and adapt them according to their needs. One of the things we want to start to explore with this research and scoping report, is how we can aid in this process of enabling researchers and publishers to use and adapt the tools needed to create experimental books.

The more expert knowledge of system administrators and programmers is primarily needed when experiments fail or get stuck. However, recent trends around cultures of software deployment, which were introduced by the use of virtual machines in the cloud, followed by the acceptance of light virtualisation aka. [containerisation](#), greatly improved the testing and usage of software tools. These days any software tool developed to be run on a server should come with decent accompanying documentation and should in most cases only need a few lines pasted into the command line to use the tool according to one's needs. To support the uptake of tools and software that can help publishers and authors in the creation and publication of experimental books, we will in this report, where appropriate, try to describe the basic competencies needed (as a basic or regular user, an advanced user, or an expert user) to successfully test different types of software.

Collaborative Writing Tools

Within COPIIM we are running a series of [pilot projects](#) focused on creating experimental books together with a selection of authors and publishers. In this section we will focus on tools that support a variety of practices or modes of research that accompany or form the basis of various experimental publishing projects, namely

collaborative writing and annotation tools. Other practices covered in this section include the facilitation of remix and re-use of content through open licensing, versioning and forking, as well as computational publishing.

Collaborative real-time writing / editing as an idea was introduced in 1968 by Douglas Engelbart in *The Mother of All Demos*¹³ but it took another forty years to be implemented in such a way that people could work collaboratively from their personal computers and rely on the service to keep their documents in place. In order for that to happen, Google played an important role by first acquiring Writely in 2006 and then in 2009 the team of AppJet created the, at that time, very impressive EtherPad application (mostly as a demo for their underlying technology). AppJet's engineers joined the Google Wave team and EtherPad was made available by Google as open source software.¹⁴

Pads & platforms: EtherPad, CryptPad, CodiMD/Hedgedoc

In the following decade we witnessed the development of a new culture of collaborative writing/editing that developed around so-called ‘pads’. The common denominator of pads is that their source text is always available in some simple human readable form (most recently Markdown) and their features have been mostly developed to support the communities using the tool.

EtherPad Lite was a rewrite of EtherPad, aiming to make it less resource hungry. It was written in a popular programming language (Javascript), making EtherPad Lite easy to install on one’s own server—i.e., EtherPad Lite can be installed via Linux distribution package managers or via Docker. Many activist organisations have chosen to use EtherPad.¹⁵

One notable project which follows the pad paradigm is **CodiMD (now HedgeDoc)**.¹⁶ In CodiMD’s Software-as-a-Service rendition HackMD, the platform is focused on providing an online space for collaborative text editing by integrating an account login system with popular online services (Google, Facebook, Twitter, Dropbox, GitHub...) and integration with GitHub for easier development of documentation. This wide range of log-ins makes the platform an interesting exemplar for experiments in the field of publishing, as it facilitates potential participation across a wide range of stakeholders. Next to the platform offer, and similar to Etherpad, self-hosted instances of CodiMD have grown popular in and beyond the HE context.¹⁷

Another example of a collaborative writing pad is the employee-owned French company XWiki SAS, which has developed a suite of tools focusing on cryptography

including [CryptPad](#), following the ‘zero knowledge’ approach where every web browser encrypts its own pad content so that even the owners of the server serving the web app to the web browser cannot decipher the encrypted content. This whole ecosystem of apps can also be installed on one’s own server.

The following table displays a list of recent tools that can be used to facilitate collaborative writing in a variety of ways. The list is limited to collaborative writing tool solutions that are under active maintenance (i.e., updated in the recent past), and available under an open-source license. This spreadsheet and the spreadsheet listing annotation tools added to the next section of this report, are works-in-progress and will continuously be updated.

Collaborative Writing Tools	Short description	Real-time editing with multiple collaborators	Rich-text editor	Real-time track changes	Commenting	Standalone tool or platform-based	Level of expertise to run as a self-hosted web service	Input / Ingest	Output formats	Landing / Project page URL	Repository URL	SaaS (Software-as-a-Service)
FiduWriter	web-based collaborative text editor made for academics with support for citations and formulas	yes	yes	yes	yes	standalone (self-host or SaaS)	regular	Fidus file	DOCX (epub, HTML, LaTeX, JATS, ODT)	www.fiduwriter.com	github.com/fiduwriter	www.fiduwriter.com
JupyterLite	A full JupyterLab distribution running in the browser	yes	depending on the choice of module (Markdown, text, Jupyter Notebook)			standalone (self-host)	Advanced	Admin knowledge of certain recommended, can be run via GIP Pages		https://jupyterlite.rtfd.io	https://github.com/jupyter/jupyterlite	https://blog.jupyter.org/up-to-date-jupyter-for-notebook-notebook-velocity-for-notebook-python-the-jupyterlab
PubPub	supports community-based collaborative drafting, review, and publication of scholarly work “using an integrated and iterative process”	yes	yes	no	yes	standalone	regular	7 March 2022	docx, epub, html, markdown, epub, html, odt, lat, JATS, LaTeX, tex	www.pubpub.org	github.com/pubpub/pubpub	www.pubpub.org
Overleaf	a collaborative cloud-based LaTeX editor used for writing, editing, and publishing scientific documents	yes	yes, either LaTeX-based writing or Rich Text available	yes	yes	standalone (self-host or SaaS)	Advanced / expert	junior spreadsheet able to run desktop image	PDF, LaTeX project as zip, optional bib file for bibliography	https://www.overleaf.com	github.com/overleaf/overleaf	https://www.overleaf.com
Wikibooks		yes	limited	no		platform						

Figure 1

Overview of open source Collaborative Writing Tools considered in this study. View this spreadsheet at <https://tiny.cc/copim-collab-writing>.

Git-based Collaboration

The world of collaborative software development was revolutionised by *Git*, which was developed by Linus Torvalds in 2005. Git was developed primarily for Torvalds' needs in maintaining one of the largest software collaborations ever—the Linux kernel. The approach and architecture of Git is also known and described as a distributed version-control system for tracking changes in source code during software development. The history of changes keeps its consistency and reproducibility by generating cryptographic hashes¹⁸ for every change of the content. The whole repository with its history of changes is then cloned for every user of the system. Future synchronisations of a code repository could thus be done in between any of the software instances, which allows for a true so-called ‘[peer2peer topology](#)’. With Git's internal architecture

and forking/branching mechanism,¹⁹ Torvalds addressed another well-known problem in software collaboration: the issue of experimenting and introducing new features or even rewriting code. Creating new forks and branches of code, while providing synchronisation with the others became much easier with the introduction of Git, resulting in drastic changes in the world of software development.

But this change did not happen more generally until [GitHub](#) (2008) made a proprietary web frontend for Git, enabling software developers to use it through a user-friendly web interface. GitHub also wrote an extensive documentation and a recorded series of screencasts explaining how to actually use Git (both in the command line and using one's own web user interface).

Now in its 14th year of existence, [GitHub](#) has become an essential part of the infrastructure of storage and history of changes in the development of open source software. While GitHub itself is now a commercial entity owned by Microsoft ([2018](#)), throughout its history it did introduce a number of important and influential open source projects, namely: Atom (a text editor),²⁰ Electron (a web browser engine as desktop application),²¹ and Jekyll (a static site generator).

Many powerful and popular text editors, such as Emacs and Vim,²² which have been used for decades in software development, are also known to have a steep learning curve. However, due to again decades of customisation, these editors are often the first to provide support for new technologies—including technologies needed for scholarly research and writing. Many scientists in particular started to use Emacs or Vim because they wanted to have support for LaTeX, BibTeX and/or other bibliographic and citation management options.

The popularity of Atom, together with the ever-growing popularity of web technologies, fuelled the development of text editing components for the web (and for desktop via Electron). Some of the most powerful and elegant amongst these, such as [CodeMirror](#) and [ProseMirror](#) by Marijn Haverbeke, have supported a new generation of web-based text editors. These text editors share their underlying technology with ProseMirror and/or CodeMirror, and based on feedback from their users would, usually, iteratively grow into specific niche contexts.

Due to the latest developments of the CSS standard,²³ web browser engines are becoming increasingly an environment where well-structured content can be processed into a PDF publication with user control over the required layout (header, footer, margins) and pagination (links to specific pages etc.). Free software libraries

that have been helping developers to integrate these features include [paged.js](#), developed by Cabbage Tree Labs in their endeavour to provide the underlying technology for Editoria. [Editoria](#) is a full-stack²⁴ web-based publishing workflow, supported with its own underlying set of technologies, including Wax, which is an online rich text editor (component) based on ProseMirror and paged.js for its typesetting,²⁵ and the [XSweet](#) converter, which converts Microsoft Word documents to HTML (and vice versa).

Also relying on ProseMirror, and combining this with [Vivliostyle](#), another established open source library for typesetting/rendering PDFs, is [Fidus Writer](#) — ‘an online collaborative editor especially made for academics who need to use citations and/or formulas.’ ([n.d.](#)) It proposes semantic editing, which is focused on the structure of the document rather than its look and feel. If the document is developed following the proposed semantic editing, Fidus Writer is able to render and export its output in different formats (HTML, Epub, LaTeX, Journal Article Tag Suite (JATS), docx, odt and PDF). It supports citations via drag'n'drop or copy-paste of BibLaTeX²⁶, easily exported from a reference manager such as [Zotero](#) and from text into the text editing area. Fidus Writer uses ProseMirror as its underlying text editing component, and Vivliostyle for typesetting and it can be easily installed locally (or on a server) as a docker container.

GitHub not only took care of educating people about and simplifying the use of Git, it also changed the way tutorials and documentation look. GitHub tried to encourage developers to add basic documentation for projects in their README.md files, to enable the repository page to open as a nicely designed HTML page with lists of the directories and files and below that the content of the Markdown formatted README.md file, processed automatically on GitHub's server. A well-designed front page, functioning as basic documentation, made software projects distinctive and more comprehensible if compared to other web frontends for version control systems.

In 2008, in its early days, GitHub introduced [GitHub Pages](#) based on the [Jekyll](#) static website generator. This allowed users —predominantly software developers at that time — a simple way to create a web site. The existence of themes would help people to choose the design and layout of their website, in a similar way as they would do in WordPress. The content creation in GitHub Pages was based on [Markdown](#) markup, a human readable syntax to structure the content of a given web page. The hierarchy of documents would follow the hierarchy of the directory structure. With a simple configuration file inside a repository, Jekyll would know how to create a menu for the

website and render the rest of the website. The web site would be rendered as a simple HTML, CSS and maybe some basic Javascript, easily served by GitHub servers with no hassle for developers to maintain their project's website or any web server.

In 2011, [GitLab](#) started as a project that would be able to provide the efficiency of code management that had been introduced by Github while also allowing more control over where a project's code is stored. Today, GitLab is available in two distinct flavours; while its Enterprise Edition (GitLab EE) is the software-as-a-service (SAAS) branch, the Community Edition (GitLab CE) follows the open source route of making its codebase available for others so that everyone has the ability to run one's own self-hosted GitLab server. And similar to the earlier-described publishing interface of GitHub Pages, such a set-up is also possible with [GitLab Pages](#).²⁷

Next to the static site generators mentioned above—Jekyll, GitHub Pages, and GitLab Pages—the [Jamstack](#) approach has led to the rise of a plethora of static site generator variants,²⁸ including Hugo, which the COPIM project is using for its website. Many of these generators have eventually found their respective ways into open publishing workflows, for journals, books, as well as fully-digital, experimental modes of publishing.²⁹

Annotation Tools

From its early days, the World Wide Web has been perceived as a medium enabling everyone and anyone to participate. It seemed that the limitations that Brecht found unacceptable for radio—as a public medium, to be merely *unidirectional*—and called for a transformation ‘*from a distribution apparatus into a communication apparatus*,’ ([Brecht & Silberman, 2020](#)) could now finally be realized with the World Wide Web.

Following this perception, it was easy to imagine that anyone could write their prose in HTML and have it published online; that one could share a URL to a comment or threaded discussion; that one could do everything we are used to do in text and/or literary criticism, with the promise of endless possibilities to expand even further. In other words, the idea that anyone, not just experts, could edit any web page, was, at the time, inseparable from the idea of Word Wide Web. It was reflected in everything from [WikiWikiWeb](#), created in 1995 by Ward Cunningham as a user-editable website, to the ‘View source’ button, which was a prominent menu item in the original web browser written by Tim Berners Lee, a feature that since then has been inherited by all other web browsers.

The history of annotation tools proved once again that many simple and elegant ideas become difficult to implement and sustain once they are presented with the myriad of competing standards and technical specifications now existing in the real world.³⁰ Fully successful implementations of a standalone (open source) annotation layer on top of regular web standards is still to be developed.³¹ Some of the challenges, affecting its promise to be useful, include ever-changing—or even disappearing—web pages which then, as a consequence, require a permanent online service to be able to consistently provide the annotated version of the web page. Archiving web pages for longer periods of time also became a non-trivial problem as nowadays the actual content of a web page does not only comprise static HTML content served by a web server anymore, which would lend itself more readily to referencing due to its static nature. Today, content is in many cases dynamically assembled by Javascript at the very last moment before a web page is displayed to the end user. And while in its daily role of simply surfing the Internet, the Javascript engine is known to be very demanding on CPU and RAM resources (even in the rather standard scenario of one single user's day-to-day web browsing on a powerful personal computer), it is still one of the most widespread frameworks used in web development.

The above-mentioned obstacles probably played an important role in the rise and subsequent demise of a number of annotation projects (both open source and proprietary). Having grown familiar with this kind of history, many recent projects—unfortunately—have decided to develop annotation as a feature that would only cover their respective projects' scope, with most of them not dedicating enough time to questions of interoperability. To provide one recent example, we can nowadays find a very good implementation of annotations on the [PubPub platform](#) created by [MIT Media Lab](#) and further developed and maintained by [Knowledge Futures Group](#), with the limitation that annotations only work within that platform.

Still, there is a project which keeps up our collective hopes by the name of [Hypothes.is](#) — an open source project following the open standard developed by the W3C [Web Annotation Working Group](#). The project gathered a scholarly coalition ([Annotating All Knowledge](#) (AAK)) — a group that includes more than seventy scholarly publishers and platforms. Their mission is to 'deploy annotations across much of scholarship.' A lot of other promising technologies were relinquished in the past because of a lack of widespread adoption (see, for example, RSS³² or the above mentioned 'View source' button), meaning that this approach focusing on this specific segment of scholarly engagement, seems reasonable and hopefully sustainable.

Hypothes.is has a special partnership program with publishers and educational institutions which often results in new features and spin-off projects, including a collaboration with the [ReadiumJS](#) team to bring annotations to EPUBs, initiated by NYU Press.

A particularly interesting project worth mentioning is [dokieli](#), a client-side tool for decentralised article publishing, annotations, and social interactions based on open Web standards and best practices ([Capadisli et al., 2017](#)). It is part of an ecosystem around project Solid, which has been initiated by Tim Berners Lee in 2016 with the aim ‘to radically change the way Web applications work today, resulting in true data ownership as well as improved privacy.’ [33](#)

Dokieli as a project is in its early stages of development and possibly a great candidate for experiments in annotations as part of a future (more) decentralised web. That said, for experimental publishing projects relying on a robust implementation and easy-to-use annotation system, our recommendation here would be to use Hypothes.is.

Overview of annotation-specific standalone tools

The following (linked) table provides a list of current tool examples that can be used to facilitate annotation in one way or the other. In line with the [earlier-introduced](#) criteria, the list is limited to open source annotation tool solutions that are under active maintenance (i. e., updated in the recent past). The list thus does not feature earlier implementation examples such as those listed on the [AnnotatorJS page](#), as AnnotatorJS has now been integrated as a core W3C standard, and many of the tools created from around 2012 to 2015 have either ceased to exist or are not seeing active maintenance and/or further development today.

	A	B	C	D	E	F	G	H	I	J	K	L
	Annotation Tools	Short description	standalone or cross-platform	Collaborative / Multi-user annotation	Annotation as comment	Other forms of annotation? (beyond commenting)	Annotation export?	Level of expertise to use	Level of expertise to run as a web service	Output formats	Landing / Project page URL	Repository
1	hypothes.is	browser-based tool for making annotations on web pages. Can be integrated with other tools / platforms, functions as an overlay on existing content (so the annotation source can live anywhere on the web)	standalone, cross-platform	yes	yes	no	yes, in development	regular user	junior sysadmin able to run docker image	HTML, JSON, TXT	web.hypothes.is/	github.com
2	Recogito	a web-based environment for collaborative semantic annotation. It is open source software, and provides support for working with either text or image documents.	standalone	yes?	yes	yes	yes	regular user			https://recogito.pelagios.org	github.com/recogito
3	eMargin	an online collaborative annotation tool. You can highlight, colour-code, write notes and assign tags to individual words or passages of a text. These annotations can be shared amongst groups, generating discussions and allowing analyses and interpretations to be combined.	standalone	yes	yes	no	yes	regular / advanced		endnotes in doc; TEI-compliant XML	https://emargin.bcu.ac.uk/	https://svn.net/projects/
4	Annotation Studio	a suite of collaborative web-based annotation tools currently under development at MIT	standalone	yes	yes	yes, include images, videos, etc. Also supports tagging of annotations.		regular user (can be used in seminar/classroom settings)	work with PostgreSQL, Ruby, Node.js, NPM and MongoDB (next to Annotation Studio, the MIT Annotation Data		annotationstudio.org	github.com/StudioAnn; github.com/StudioAnn/AnnotationStore
5	CATMA	With CATMA you can easily annotate your text just as you would do in a book	standalone	yes	yes				junior sysadmin able to run git		catma.de	github.com/catma

Figure 2

Overview of open source annotation platforms considered in this study: View this spreadsheet at <https://tiny.cc/copim-annotation-overview>.

As we described it in the [first part](#) of the report *Promoting and Nurturing Interactions with Open Access Books: Strategies for Publishers and Authors*, web-based annotation of digital books can be thought of as “a way to enrich a scholarly text through overlays and filters that sit on top of the text in order to show additional commentary and feedback.” ([Adema et al., 2021](#)) On a technical level, annotation usually happens *in situ*, i.e., on top of an existing publication. With physical books, this usually happens in the margins of a book or manuscript. In the digital realm, though, this practice has proliferated: one common form of *indirect* annotation includes commenting at the end of a publication, separate from the main text body (see for example the comments section of blogging platforms such as *blogger* or *WordPress*) or what the W3C describes as being “maintained separately from annotation document” ([2014](#)). Due to the detached nature of this form of annotation, such commentary tends to be more conducive to summative feedback.

Other more creative forms facilitate *direct* annotation by adding an extra (digital) layer over the original publication—a layer that often allows direct referencing of granular elements (specific words, segments, paragraphs), thus enabling the reader to provide feedback via textual or multimedia means, or by adding contextual references such as metadata to enrich the underlying text, e.g., by creating a semantic network that sets a given publication in relation to other publications (hyperlinking, linked open data).

As discussed [in more detail in Part 1](#) of our report on Interaction & Reuse, **Open Peer Review** is “an umbrella term for a number of overlapping ways that peer review models can be adapted in line with the aims of open science”, and “a diverse cluster of

interrelated yet distinct innovations that aim to bring open science principles like transparency, accountability, and inclusivity to the peer review process” ([Ross-Hellauer, 2017](#)).

Open Peer Review of scholarly books can be facilitated through a variety of means, many of which make use of commenting, annotation and/or versioning, depending on the chosen mode of interaction with the publication under review. More traditional forms of peer review maintained a separation between the review and the book under review, for example by using structured review forms, or book reviews published post publication. Digital annotation enables reviewers to write directly in or on the book under review, creating a more immediate and interactive experience.

In the first version of our COPIM Report *Books Contain Multitudes* ([Adema et al., 2021](#)), we introduced a broad differentiation between *tools* and *platforms*: on the one hand, we consider tools that facilitate annotation as part of a larger collaborative environment that mainly focuses on the writing and publishing process (see platforms such as PubPub, CryptPad, etc. as discussed in the [Collaborative Writing overview](#)). On the other hand, there exist a variety of specialist platforms that focus on the facilitation of annotation as their main purpose, either within a given platform’s boundaries (see e.g., Rescogito, CATMA), or as tools that can be used across platforms and independently from their base text’s locations (e.g., Hypothes.is).

Platform-agnostic / Overlay Annotation Tools

The following tools are highlighted here because they work as platform-agnostic/-independent implementations. Adhering to the W3C’s [Open Annotation Guiding Principles](#), these tools facilitate an overlay service that can be used in conjunction with (almost³⁴) every existing website, platform and/or digital document.

hypothes.is

[hypothes.is](#) is an open source project that has evolved out of the development work undertaken in the W3C Web Annotation Working Group. As Mars et al. write,

“the project gathered a scholarly coalition (Annotating All Knowledge (AAK)³⁵) — a group that includes more than seventy scholarly publishers and platforms. Their mission is to ‘deploy annotations across much of scholarship’ [and, to us] seems [a very] reasonable and hopefully sustainable [approach]. Hypothes.is has a special partnership program with publishers and educational institutions which often

results in new features and spin-off projects, including a collaboration with the RadiumJS team to bring annotations to EPUBs, initiated by NYU Press” (2021).

[Hypothes.is](#) is seeing wide-spread adoption across the Higher Education sector, and is featured in a variety of open publishing as well as open education projects to foster uptake of social annotation practices (see [Kalir & Garcia, 2021](#),³⁶ and [Part 1](#) of our Interaction & Reuse report), which is supported on a technical level through the provision of a set of tools to help integrate [hypothes.is](#) functionality in a variety of other platforms also used for open access book publishing such as WordPress, Omeka, Open Monograph System etc.³⁷

The platform-agnostic nature of [hypothes.is](#) makes the tool a versatile candidate for implementation in third-party environments. One example use case seems particularly noteworthy in this context. The High Integration of Research Monographs in the European Open Science ([HIRMEOS](#)) infrastructure project (also [discussed in Part 1](#) of our Interaction & Reuse report)—sought to create a set of services to enhance re-use and integration of monographs into the larger European open science ecosystem. The project developed the **HIRMEOS Annotation service**, which facilitated open annotation for digital books for the publisher [OpenEdition](#), based on [hypothes.is](#). This service enhances capabilities towards creating annotations with an implementation of annotation-specific DOIs, and also enables storage and long-term preservation, re-use and sharing of the annotation record and associated data.³⁸ The chosen approach is described in more detail in [Bertino & Staines, 2019](#), as well as in the HIRMEOS Fact Sheet “[Annotation Service for Digital Monographs](#)”. An overview of the books selected for their annotation and open peer review experiment has been made [available online](#).

Another use case that deploys the [hypothes.is](#) model for annotation is **Fulcrum**. This publishing platform, which is developed by Michigan Publishing and focuses on the integration of a variety of multimedia content types such as interactive maps, datasets, 3D models, images, timelines, etc.³⁹ into digital open access books—while also taking into account the preservation of these content types—announced in 2019 that it would implement [hypothes.is](#) annotation features with books published by Lever Press on the Fulcrum platform, while also [hinting at the possibility of making this feature available](#) for other publishers’ output on its platform at a later date.

PressBooks is another interesting use case to mention here because it integrates [hypothes.is](#) in their WordPress-based publishing platform via the annotation tool’s [excellent plugin](#) to facilitate reader feedback. As PressBooks is also used as a platform

to publish and disseminate OER textbooks, the integration of an annotation layer is also key to fostering student engagement with a given text.⁴⁰

Pundit Annotator

Similar to [hypothes.is](#), **Pundit Annotator** has existed for quite some time, and is currently in the early stages of being re-developed from scratch to ensure full implementation of the W3C Annotation standard that came into effect in 2017.⁴¹ Conceived as a peer-review platform that leverages openly-available open access content via arXiv, OAPEN, and Knowledge Unlatched, and supported by the European Commission-funded TRIPLE project that is part of OPERAS,⁴² Pundit will become a service offered as part of the GOTRIPLE platform, which in turn is conceived to play its part in the *European Open Science Cloud* ecosystem, and is thus seeing integration of multi-platform sign-on capabilities,⁴³ which will allow researchers to use the annotation service Pundit Annotator. While the project used to have its own open source repository, it is not clear at this point whether the new version 3.0 will also be made openly available.⁴⁴ What is also interesting is the fact that the development team hint at a collaboration with hypothes.is, which will potentially lead to more cross-platform interoperability in this space — with both tools soon being envisioned to enable re-use of each other’s annotation data.

“Standalone” Fixed-ecosystem Annotation Platforms

Many annotation experiments in scholarly communication happened in the early 2010s, when the development of a key javascript library (AnnotatorJS) meant that an introduction of annotation functionality would be rather easy to facilitate. Nonetheless, the creation of an efficient and sustainable annotation environment was not trivial. The subsequent inclusion of the work that had informed AnnotatorJS into the W3C Annotation framework in 2017 was welcomed by the community, but also meant that much of the earlier javascript-based development work had become obsolete. As evidenced in the overviews provided by [Max Planck Institute for the History of Science](#) (MPIWG) and AnnotatorJS’s own [showcase page](#), many of the platforms that had been established during the early 2010s to enable annotation subsequently [ceased active development](#), while a smaller subset decided to invest and re-build their platforms to implement the new W3C annotation standard.

In the next section, we take the opportunity to highlight a selection of specialist annotation tools and platforms that are used e.g. by linguists and historians. This will

then be followed by a section that takes a closer look at three emerging platforms that follow an integrated approach to collaborative writing and annotation.

Recogito

An initiative by the Pelagios Network, and originally having focused on geographic annotation of maps, [Recogito](#) has since evolved into a powerful interactive annotation tool for text and image documents, support [International Image Interoperability Framework \(IIIF\)](#) standard. The tool supports collaborative and semantic annotation, allowing a whole team to work on a given text, image or map, and to connect individual data points such as places, characters, and events. Honouring an open data approach, all of the annotation data collected in Recogito can also [be exported in open formats](#) for re-use in other tools and platforms.

Recogito has been awarded *Best Open Source Software* in the [Open Publishing Awards 2019](#) and *Best DH Tool* in the [Digital Humanities Awards 2018](#).

Figure 3
Annotation within [Recogito](#).

CATMA

Looking back at more than a decade of development at University of Hamburg and TU Darmstadt, Germany, [CATMA](#) — short for "Computer Assisted Text Markup and Analysis" — supports specialist semantic annotation of text. It is being used for linguistic corpus analyses, supports export of its annotation collections to [Text Encoding Initiative \(TEI\)](#)-conformant XML, and relies on a [git-based backend and workflow](#).

Figure 4
Source: [CATMA](#)

Annotation Studio

Hosted and developed at [MIT School of Humanities, Arts & Social Sciences](#)' Active Archives Initiative (formerly [MIT Hyper Studio](#)), [Annotation Studio](#) boasts a suite of collaborative annotation tools. With a Digital Humanities background in mind, Annotation Studio offers an approach to TEI-compliant annotation without making prior knowledge of close textual analysis practices or specifics around TEI a requirement. Next to facilitating direct interaction with a text through annotation, Annotation Studio also offers a visual approach to map reader engagement with a text through its Heatmap and Reader Trajectories panes.

They write:

by enabling users to tag texts using folksonomies rather than TEI, Annotation Studio allows students to [discover] how literary texts can be opened up through exploration of sources, influences, editions, and adaptations.

Source: [MIT Hyper Studio, n.d.](#)

Visit the web version of this article to view interactive content.

A short introduction to an Annotation Studio use case, an integration of the annotation tool in a Media, Communications, and Culture Course “Digital Approaches to Boston Culture”. For more information, see <https://www.annotationstudio.org/portfolio-view/w-i-t-and-public-digital-humanities/>

While annotation-specific platforms such as those outlined here are definitely worthy of further in-depth exploration, we would like to highlight three emerging platforms that follow an integrated approach to collaborative writing and annotation and that also specifically accommodate books or long-form texts. They focus on the social aspect of collaborative interaction with the text and thus aim to provide a seamless experience across many steps of the publishing workflow.

Scalar

Scalar, the multimedia publishing platform hosted by the Alliance for Networking Visual Culture (ANVC), provides options to annotate video, audio, images, source code, and text. By establishing relational links between various kinds of content, Scalar introduces an elaborate taxonomy to facilitate a wide range of potential connections between annotations and base content. In practice, this means that one can establish links between existing content types in a Scalar book, or add new content (a note, a video commentary, etc.) to an existing content type.⁴⁵ Scalar also features an API through which—as the manual states—“You can mashup your Scalar content with other data sources, build your own visualizations, or create completely new interfaces for your materials.”⁴⁶ While such a feature might not be relevant for every user, it is noteworthy because it offers possibilities for re-using Scalar content outside of the platform.

The digital project *Bodies and Structures 2.0*, led by Kate McDonald and David Ambaras, has used Scalar to develop a fascinating project and digital collection mapping seventeen spatial histories of modern East Asia.

Figure 5

For those interested in using Scalar as a platform for their teaching, we'd recommend Dixon's article "Imagining the Essay as Digital Assemblage: Collaborative Student Experiments with Writing in Scalar." (2016) and this [Introduction to Scalar](#), provided by University of Illinois at Urbana-Champaign.

Figure 6

Manifold

Developed as a successor to the *Debates in the Digital Humanities* hybrid print/digital book publishing platform ([Kasprzak & Smyre, 2017](#)), **Manifold** leverages the social aspect of collaborative interaction through its annotation Reading Groups. As the [developers note](#), Reading Groups “are a way for readers to annotate and comment on Texts as a cohort and is geared toward classroom and peer-review use cases.” ([n.d.](#)) Athabasca University Press and University of Minnesota Press are already using bespoke Manifold instances to foster engagement with their published books,⁴⁷ and pilot projects between the University of Washington Press and University of Washington Libraries, at City University New York (CUNY), and at Affordable Learning Georgia, are using the platform to explore the potential of extending student engagement with open texts through social collaborative practices, including annotation.⁴⁸

Figure 7

Manifold annotation and peer review cases:
[Metagaming](#) (2017) and the [Middlemarch 150th Anniversary Symposium](#) collaborative edition project (2021).

Figure 8

Whitney Trettien's recent book publication
[Cut/Copy/Paste: Fragments from the History of Bookwork](#) (2021), on Manifold.

PubPub

As outlined in more detail in Mars et al. 2021, **PubPub** is a collaborative writing platform that also integrates an annotation layer to facilitate commentary and peer review. In an exemplary **Open Peer Review process via PubPub**, Remi Kalir and Antero Garcia made the manuscript of their—now published—*Annotation* volume, available [online via the PubPub platform](#), and invited feedback via in-platform annotations and comments from the wider scholarly community.

Figure 9

Sample screenshot of Kalir & Garcia's *Annotation* prepublication manuscript [on PubPub](#).

In a similar vein, with her [preprint version released as open access text on PubPub](#), Sasha Costanza-Chock’s *Design Justice: Community-Led Practices to Build the Worlds We Need* invites readers to share thoughts and comment on her MIT Press monograph that has been [published under the same title in 2021](#).

And the *Frankenbook* project, presented by the Center for Science and the Imagination at Arizona State University, has likewise employed PubPub’s annotation capabilities to engage in a “collective reading and collaborative annotation experience” to reframe Mary Wollstonecraft Shelley’s original 1818 text of *Frankenstein; or, The Modern Prometheus*.

Figure 10
Screenshot of <https://www Frankenbook.org/>

As a caveat, it remains to be seen if PubPub’s annotation framework will, in the future, allow export and re-use of its annotation-specific data so as to more formally comply with the Open Annotation Guiding Principles⁴⁹ and corresponding calls to make peer review data available independently from its publishing platform. Next to that, for the authors of this report, the mandatory sign-up / registration step that is required prior to gaining access to the interaction options of a given base text in PubPub poses an additional barrier that might deter some users from interacting with the text. Nonetheless, PubPub’s support of annotation and peer review on the technical level of the tool and its affordances, but also on the level of fostering social interaction and community-building on and with PubPub (e.g., through the *Commonplace* publication outlet, led by Knowledge Futures Group, the community tasked to provide development of and user support for the platform)⁵⁰ makes for a rather convincing case of an emerging publishing ecosystem.

CommentPress

Leveraging a **WordPress + CommentPress** [plugin](#) setup that had been pioneered by The Institute for the Future of the Book (If:book, [Fitzpatrick, 2007a](#)), Jason Mittell’s Media Studies publication *Complex TV* had been publicly available for close to two years prior to its publication via If:book’s [MediaCommons platform](#), and the

manuscript has subsequently undergone a thorough “Peer-to-Peer Review” ([Fitzpatrick, 2007b](#)) process together with publisher NYU Press. Although it has already been published nine years ago, Mittell’s book still is an interesting exemplar to consider here because it also conceptually combines a variety of open source platforms, [drawing on Scalar](#) to provide additional digital material to support the arguments made in the main publication.

Figure 11

[Review platform](#) of Jason Mittell’s Media Studies publication *Complex TV*, powered by MediaCommonsPress.

The manuscript has been published [in 2015 with NYUPress](#).

Similar processes have been employed for example by McKenzie Wark for her monograph [GAMER THEORY](#), by Jack Dougherty and Kristen Nawrotzki for their 2011 open review volume of [Writing History in the Digital Age](#) (published in 2013 by University of Michigan Press), and again by Kathleen Fitzpatrick, who had also used this process to invite feedback on her book *Planned Obsolescence* (2011) [via MediaCommons](#), while her more recent book *Generous Thinking* ([Fitzpatrick, 2019](#)) has been made available with a more up-to-date [CommentPress setup hosted at Humanities Commons](#) (see below).

Figure 12

Humanities Commons / [CommentPress-based review process](#) for Kathleen Fitzpatrick's monograph *Generous Thinking*, published with Johns Hopkins University Press in 2019.

RavenSpace

RavenSpace is a collaborative publishing space developed by University of British Columbia Press in close collaboration with University of Washington Press, and focuses on digital workflows to extend the collaborative writing experience towards the provision of a robust peer review workflow that can also facilitate what they label “Community peer review”. Through Community Peer Review, [Ravenspace](#)

“seeks to extend the collaborative relationships of research and authoring into the publication process and to publish works that are meaningful and relevant for distinct communities of readers, both inside and outside academia, and specifically Indigenous peoples. It recognizes that expertise resides in many places and that publications benefit from Indigenous consultation or review beyond collaborative authorship. Because of the varied nature of collaborative relationships and the diversity in Indigenous customs, laws, and approaches to intellectual property and cultural heritage uses, flexibility is essential; the form of review and consultation responds to the nature of community protocols and the needs of each publication.” ([2021](#)).

An Invitation to Listen

DESCRIPTION CITATIONS

ta?aw

Elsie Paul stresses the importance of "ta?aw," the traditional teachings.
Source: Courtesy of Elsie Paul

Well, my name is Elsie Paul. My ancestral name is qo?oystoles, a name that was handed down to me by my grandfather. That was the ancestral name of his mother ... I grew up with my grandmother. As was the custom with our people, the grandmothers always helped with bringing children up. My grandmother had lost her voinnoest daughter with that residential school

Figure 13

Ravenspace - [*As I Remember It: Teachings \(Pəms ta?aw\) from the Life of a Sliammon Elder*](#).

Mukurtu

Developed by the [Center for Digital Scholarship and Curation](#) at Washington State University, **Mukurtu** is an open-source Content Management System and publishing and archiving platform that has "the unique needs of Indigenous communities, libraries, archives, and museums in mind." Relying on [Drupal](#) as its host system, Mukurtu has developed a strong community over the years, which is organised via a network of regional and local "Hubs and Spokes" ([Christen et al., 2017](#)) that fosters exchange of situational knowledge and practices. While it is not a book publishing platform per se, we are including it here as an interesting example of how communities can collaborate on digital collections and experiment with intriguing, novel ways to present, share and curate content.

Figure 14

[Gather](#) project, State Library of New South Wales

Remix and Adaptation

While still focusing on the technical implementation of remix and adaptation via tools and platforms, we will, in the following paragraphs, also look at examples of academic publishing communities that are working with these tools to put the promise of remixing long-form publications such as monographs into actual practice.

Licensing and Copyright

A vital point towards enabling re-use and interaction with one's content is to create amenable conditions for engagement. On the level of licensing, this is usually done by applying open licenses to one's work.

In a world defined by copyright law, open licenses are a good way to signal what kinds of re-use and interaction are possible.

Licenses are the most widespread way to signal what kinds of re-use and interaction are permitted by the original content creator / author. Releasing a book under an open license ensures that those interested in re-using your book (or contents thereof) would not have to reach out to you to ask for permission to do so.

Creative Commons

Figure 15

[Shaddim; original CC license symbols by Creative Commons, CC BY 4.0](#), via Wikimedia Commons.

Creative Commons licenses are a way to express different levels of such permissions, with the general rule being that those licenses with the least exceptions are those most amenable to fostering re-use. An additional benefit of Creative Commons licenses is that each license comes in three versions — a clearly understandable summary of the terms ("human readable"), the license text ("lawyer readable"), and the metadata ("machine readable"). For more on the permissiveness of the six main Creative Commons flavours, see the infographic on the left.

Licensing Tools for Easy Attribution

Creative Commons license chooser

- <https://creativecommons.org/choose/>
- <https://chooser-beta.creativecommons.org/>

The CC license chooser enables authors and contributors to select a Creative Commons license that appropriately reflects their intended use cases. Through a set of questions, the tool can identify main criteria and permissions that an author wants to grant, and then presents the creator with a variety of media-specific license attribution options with corresponding copy&paste templates (text-based, text/hyperlink, or HTML code that includes machine-readable licensing metadata)

Wikipedia & Wikimedia Commons attribution tool

- <https://lizenzhinweisgenerator.de/?lang=en>
- <https://github.com/wmde/Lizenzhinweisgenerator>
- https://commons.wikimedia.org/wiki/Commons:Attribution_Generator

Flickr image re-use attribution generators

- Michael Hex, ImageCodr: <https://www.imagecodr.org/>
- Alan Levine, Flickr CC Helper: <https://cogdog.github.io/flickr-cc-helper/>

Guidelines and Tools for Open Licensing:

- Barnes (2018) Copyright and licensing – what do I need to know?
<https://doi.org/10.11647/OBP.0173.0090>
- Borwick, C. (n.d.). *What is open access: Open access book publishing*.
<https://library.bath.ac.uk/open-access/whatisopenaccess>
- *Considerations for licensors and licensees—Creative Commons*. (n.d.).
https://wiki.creativecommons.org/wiki/Considerations_for_licensors_and_licensees#Make_sure_you_understand_how_Creative_Commons_licenses_operate
- Kreutzer, T. (2014). Open Content: A practical guide to using creative commons licences. German Commission for UNESCO.
- Collins, Ellen, Milloy, Caren, Stone, Graham, (2013) *Guide to Creative Commons for humanities and social science monograph authors*. Eds. James Baker, Martin Paul Eve, and Ernesto Priego. OAPEN-UK and Jisc Collections.
<https://eprints.hud.ac.uk/id/eprint/17828/>
- OAPEN. (n.d.). *Funder requirements: Licensing*.
- UK Copyright Literacy. (n.d.). *Copyright guidance from UK universities and colleges*.
<https://copyrightliteracy.org/about-2/copyright-guidance-from-uk-universities/>
- Specifically relating to the German legal system, which is not always compatible with the anglophone approach to Creative Commons licensing, see Kreutzer, T., & Lahmann, H. (2021). *Rechtsfragen bei Open Science—Ein Leitfaden* (2. Aufl.). Hamburg University Press. <https://doi.org/10.15460/HUP.211>

Alternative Licensing Models: CopyLeft, CopyFarLeft, CopyFair, CC4R, Traditional Knowledge

Alternative ways to license your work that are more critical of traditional conceptions of copyright include **CopyLeft** and **CopyFarLeft** licenses. The P2P Foundation's Wiki provides concise overviews of both at

- <https://wiki.p2pfoundation.net/Copyleft>
- <https://wiki.p2pfoundation.net/copyfarleft>

Copyleft licenses have evolved out of Free, Libre and Open Software (FLOSS) advocacy and related licenses such as the GPL. The **Copyleft.org project** has a [concise guide](#) on the many aspects of copyleft licensing.

To provide an example of Open Access books using copyleft licensing, we would like to highlight the book catalogue of the publisher Minor Compositions: all of Minor

Compositions' books make use of a tailored variant of a copy(far)left licensing approach.

Minor Compositions Open Access Statement – Please Read

This book is open access. This work is not simply an electronic book; it is the open access version of a work that exists in a number of forms, the traditional printed form being one of them.

All Minor Compositions publications are placed for free, in their entirety, on the web. This is because the free and autonomous sharing of knowledges and experiences is important, especially at a time when the restructuring and increased centralization of book distribution makes it difficult (and expensive) to distribute radical texts effectively. The free posting of these texts does not mean that the necessary energy and labor to produce them is no longer there. One can think of buying physical copies not as the purchase of commodities, but as a form of support or solidarity for an approach to knowledge production and engaged research (particularly when purchasing directly from the publisher).

The open access nature of this publication means that you can:

- read and store this document free of charge
- distribute it for personal use free of charge
- print sections of the work for personal use
- read or perform parts of the work in a context where no financial transactions take place

However, it is against the purposes of Minor Compositions open access approach to:

- gain financially from the work
- sell the work or seek monies in relation to the distribution of the work
- use the work in any commercial activity of any kind
- profit a third party indirectly via use or distribution of the work
- distribute in or through a commercial body (with the exception of academic usage within educational institutions)

The intent of Minor Compositions as a project is that any surpluses generated from the use of collectively produced literature are intended to return to further the development and production of further publications and writing; that which comes from the commons will be used to keep cultivating those commons. *Omnia sunt communia!*

Figure 16
Copyleft notice on Stephen
Shukaitis book [*Combination
Acts: Notes of Collective
Practice in the
Undercommons*](#), published
2017 with Minor
Compositions.

Copyfair licenses

Copyfair licenses comprise a particular subclass that focuses on equitable sharing of resources. Most notably among those is the [Peer Production License](#) which has been conceived in the context of the Open Cooperativism movement as a derivative of the Attribution-NonCommercial-ShareAlike Creative Commons license. The Amsterdam-based [Institute of Network Cultures](#)' book series *Network Notebooks* has been published under a Peer Production License, see e.g. Dmytri Kleiner's "[The Telekommunist Manifesto](#)".

Figure 17
Dmytri Kleiner's "[The Telekommunist Manifesto](#)",
published under a Peer
Production License in the
[Institute of Network Cultures'](#)
book series *Network Notebooks*

Other approaches to copyleft licensing that focus on fair distribution of value include the [CopyFair License](#) and the [Fair Source License](#).

And copyleft licenses that are usually found in software development such as MIT & GPL3 have also been applied to Open Access books, see e.g. the Berlin-based Mute Magazine's **Open Mute Press** collaboration with Open Humanities Press on After.Video Assemblages,

A video book - paperback book and video stored on a Raspberry Pi computer packaged in a VHS case.

Also available as online video and book [PDF download](#).

Video and PDF - <http://after.video/assemblages>

Source: <https://www.metamute.org/shop/openmute-press?page=1%2C0>

CC4r - Collective Conditions for Reuse

Understood as a critique on conceptions of property and copyright of the neoliberal system, the [Collective Conditions for Reuse \(CC4r\)](#) license is a reimagined copyleft license specifically geared towards reuse or remix scenarios in which collaborators do not want to "contribute to oppressive arrangements of power, privilege and difference." Constant, the Brussels-based non-profit organisation behind this license,

notes that “CC4r was developed for the Constant work session Unbound libraries (spring 2020) and followed from discussions during and contributions to the study day [Authors of the future \(Fall 2019\)](#). It is based on the [Free Art License](#) and inspired by other licensing projects such as the [\(Cooperative\) Non-Violent Public License](#) and the [Decolonial Media license](#)” ([Constant, 2020](#)).

We also want to highlight that copyleft licences are not the only licensing frameworks available. For example, **Traditional Knowledge** (TK) seeks to address the diversity of Indigenous needs to retain control of their cultural heritage and resources.

Traditional Knowledge licenses

Inspired by Creative Commons, [Traditional Knowledge](#) (TK) seeks to address the diversity of Indigenous needs to retain control of their cultural heritage and resources. TK “embraces the content of knowledge itself as well as traditional cultural expressions, including distinctive signs and symbols associated with TK.” Traditional Knowledge licenses are

“a tool for Indigenous communities to add existing local protocols for access and use to recorded cultural heritage that is digitally circulating outside community contexts. The TK Labels help non-community users of this cultural heritage understand its importance and significance to the communities from where it derives and continues to have meaning”

Source: [Program for Open Scholarship and Education, 2021](#).

TK licensing works on two levels: through Licenses & Labels. While TK Licenses function in a similar way to their Creative Commons equivalents, TK Labels provide contextual information pertaining to the community it originates from.

Figure 18

Provenance Labels identify the group or sub-group which is the primary cultural authority for the material, and/or recognizes other interest in the materials.

Source: <https://localcontexts.org/labels/traditional-knowledge-labels/>

Figure 19

Protocol Labels outline traditional protocols associated with access to this material and invite viewers to respect community protocols. Source: <https://localcontexts.org/labels/traditional-knowledge-labels/>

Figure 20

Permission Labels indicate what activities the community has approved as generally acceptable. Other uses require direct engagement with primary cultural authorities. Source: <https://localcontexts.org/labels/traditional-knowledge-labels/>

The University of British Columbia [provides further details](#) on the uses of Traditional Knowledge licenses.

To highlight an Open Access book that makes use of TK Labels, we would like to recommend taking a look at the wonderful RavenSpace publication [As I Remember It](#), a collaboration with Elder Elsie Page, Davis McKenzie, Paige Raibmon, and Harmony Johnson.

Semantic Web, Linked Data, Translatability & Citations

That said, while the practice of using open references and citations in one’s output is seeing considerable uptake particularly in the STEM fields (see e.g., [Hutchins, 2021](#)),

an adaptation of workflows that make reference and citation datasets openly available is still lagging behind in the world of the Humanities and Social Sciences.

I4OC & OpenCitations

Leveraging the principles of open data through PIDs and [Semantic Web \(Linked Data\)](#) technologies, the [Initiative for Open Citations](#) (I4OC) and [OpenCitations](#) seeks to collect citation data to create semantic, machine-readable networks that link citations and references across individual research outputs. Implementing OpenCitation standards in one's monograph creation workflow can be another way to improve and invite re-use of original content, as machine-readable, standardised metadata promises to make proper attribution of sources more readily available. As the provision of open reference lists plays an important part in the Declaration on Research Assessment (DORA), this practice will surely see more wide-spread uptake across HE institutions and publishers against the backdrop of the larger move towards facilitating uptake of practices on the spectrum of [Open Science and Scholarship](#). For a very recent discussion of the benefits and obstacles regarding OpenCitations, see e.g., [Ayers & Klein, 2021](#).

CARE Principles for Indigenous Data Governance

Linked to the [use of TK licenses and labels](#), the **CARE Principles for Indigenous Data Governance** provide guidance on re-focusing data provision towards a more responsible and ethical approach that complements the [FAIR principles](#). As the Global Indigenous Data Alliance (GIDA) writes:

Existing principles within the open data movement (e.g. FAIR: findable, accessible, interoperable, reusable) primarily focus on characteristics of data that will facilitate increased data sharing among entities while ignoring power differentials and historical contexts. The emphasis on greater data sharing alone creates a tension for Indigenous Peoples who are also asserting greater control over the application and use of Indigenous data and Indigenous Knowledge for collective benefit.

This includes the right to create value from Indigenous data in ways that are grounded in Indigenous worldviews and realise opportunities within the knowledge economy. The CARE Principles for Indigenous Data Governance are people and purpose-oriented, reflecting the crucial role of data in advancing Indigenous innovation and self-determination. These principles complement the

existing FAIR principles encouraging open and other data movements to consider both people and purpose in their advocacy and pursuits.

Source: [Global Indigenous Data Alliance \(GIDA\), 2019](#); and [Ruckstuhl, 2022](#).

Figure 22

Source: <https://www.gida-global.org/care>

Open Syllabus

One text and data mining (TDM) application use case of an open citation graph has evolved out of a project hosted at Columbia University's [Group for Experimental Methods in the Humanities](#): the [Open Syllabus project](#) collects and scans openly-shared course syllabi for references, and makes the connected dataset and generated visualisations available via its dedicated not-for-profit platform at <https://opensyllabus.org/>. All scholarly references included in the scanned syllabi can be mapped across research fields (see e.g. the below visualisation of the most prominent texts across syllabi for media studies).

Figure 23
Open Syllabus Galaxy - [focus on Media Studies](#).

On re-using third-party material in your research publication

As the 2019 Universities UK's *Open Access Monographs: Evidence Review* report states,

Technical issues of inclusion of illustrations in an academic monograph is not the problem; rather, it is acquiring clearance permissions for the re-use of third-party material that adds an extra layer of complexity to publication, potentially making it very expensive to publish books with significant quantities of third-party copyright material ([2019](#)).

The following resources provide help along the often-difficult way through obtaining proper licensing for your third-party material.

- Rudy, K. M. (2019). The true costs of research and publishing. Times Higher Education (THE). <https://www.timeshighereducation.com/features/true-costs-research-and-publishing>
- Grosvenor, B. (2018, August 20). Why museums should abolish image fees (ctd.). Art History News. https://www.arthistorynews.com/articles/5241_Why_museums_should_abolish_image_fees_ctd
- University of York. Using images. *Copyright: a Practical Guide*. <https://subjectguides.york.ac.uk/copyright/images>
- Aufderheide, P., & Jaszi, P. (2015). Code of Best Practices in Fair Use for the Visual Arts. Copyright, Fair Use, Scholarly Communication, Etc.

<https://digitalcommons.unl.edu/scholcom/1>

Versioning and Forking of Books

“Versioning” is the practice of documenting diachronic changes in a publication—a publication is updated until an agreed-upon amount of edits has been included; this then becomes fixed & time-stamped (“frozen” reference to content and corresponding time) in a new version.

On a conceptual level, Versioning and Forking can be seen as instantiations of the Remix paradigm. While the use of version control can be applied on the level of collaborative text writing,⁵¹ the principle can similarly be applied on the level of an entire book, under the precondition that the book creation process is entirely based on a git-based workflow and its files stored in a version-control amenable repository such as GitLab, GitHub, or gitea.⁵² In this context, forking denotes the act of remix realised by a third party that is not identical with the original author. Versioning, on the other hand, is the provision of a time-stamped update under the same general provisions of the original text.

An exciting use case of book forking has been initiated by Winnie Soon & Geoff Cox, who, with their book *Aesthetic Programming* (2021), invited readers to create new versions of said publication. In response to said call, Sarah Ciston and Mark C. Marino created their own fork of the book via the **GitLab repository**, and introduced a new conversational layer—what they label “Code Confessions” and “Code Comments”—to engage with both the original text and their own remix practice ([Ciston & Marino, 2021](#)).

Two of the earlier-mentioned platforms—**PubPub** and **Manifold**—have also integrated their own approaches to versioning within their respective publishing workflows. Reflecting on the iterative process of developing a set of versions over time on a variety of platforms that have accumulated into a book manuscript, Adema has [written about her experience with versioning](#) :

“Over the last decade my book *Living Books. Experiments in the Posthumanities*, has developed in an iterative way. From blogposts to papers and conference presentations, and eventually to a thesis, a wiki, a CommentPress version, and several articles, *Living Books* further evolved into a book published by the MIT Press, in addition to an online PubPub version that can be updated, remixed, and commented upon. [...] experimenting with different versions, platforms, and media to communicate my research, served as an opportunity to reflect critically on the

way the research and publishing workflow is currently (teleologically and hierarchically) set up, and how it has been fully integrated within certain institutional and commercial settings" ([Adema, 2021](#)).

For a more expansive overview, may we refer the inclined reader to the typology developed as part of our *Books Contain Multitudes* report, with particular reference to the [segment on Versioned Books](#) in Part 2.

Computational Publishing Tools

Computational publishing is to some extent an emerging area of experimental book publishing. Though there are competing terms in this space, we use computational publishing to refer to producing a book which combines text and computational functionality in a single document. However this simple definition requires some refinement in order to distinguish the contemporary trend from its historical precursors including the Web's hyperlinking and interpreted programming languages.

The idea of using computational technology to enhance the functionality of a document is core to the very idea of the World Wide Web. As early as 1939, Vannevar Bush, an US engineer and administrator on the Manhattan Project, discussed using technology to create links—"associative trails"—between documents stored in computational storage mechanisms ([Bush, 1939](#); [1945](#)). Alex Wright discusses how Bush envisioned:

"breaking down the old hierarchy of the codex book in favour of a new kind of intertextuality that allowed for direct links between documents, removing the mediating filer of an external index... allowing authors (and readers) to insert explicit linkages between documents in a collection... Using associative trails, the user could forge a personal trail through any number of documents, creating an exteriorized representation of an internal thought process that other users could later see." ([Wright, 2007](#)).

This idea of 'associative trails' was essentially created with the Web's hypertext: computational elements inserted into (mostly) HTML documents to create links to other files on the Web.

Computational publishing also bears some similarity with Donald E. Knuth's conception of 'literate programming': "a methodology that combines a programming language with a documentation language thereby making programs more robust, more portable, more easily maintained, and arguably more fun to write than programs that are written only in a high-level language." ([Knuth, 1992](#)) Fundamentally Knuth

conceived of a program as "a piece of literature, addressed to human beings rather than to a computer." ([Knuth, 1992](#))

Knuth's literate programming shares similarities with interpreted programming languages such as PHP, Python, Ruby, and JavaScript where the program is interpreted by the computer without compiling the program into machine instructions ([McKenzie, 2021](#)). Interpreted languages are distinguished from compiled languages such as Java, C, C++, and Go where the program script first needs to be manually compiled to transform it into machine-readable code. Interpreted languages are not only human- and machine-readable but are able to incorporate documentation in the way that Knuth conceives by allowing for comments, i. e. elements of the script which can be read by a human but will not be read by the machine.

We therefore want to distinguish computational book publishing from these existing and ubiquitous forms of combining human-readable text with computational functions.

Computational book publishing refers to incorporating new and more advanced computational elements into traditional human-readable books.

Andrew Odewahn ([Odewahn, 2017](#)) outlines a few of the computational elements that can be incorporated in computational publications including but not limited to:

- rich dynamic media that can play in a browser;
- interactive data visualisations;
- executable code blocks;
- data repositories.

Computational book publications enable the reader to run code within the book itself whether to demonstrate a programming example or to dynamically adjust a data visualisation. Since the tools used to enable computational book publishing are linked to those of software publishing, computational book publishing enables the author to write a book as if writing a piece of software using practices traditionally associated with publishing source code such as collaborative writing or versioning with tools like Git or Apache Subversion (cf. [Soon, 2022](#), and this report's [section on Git-based collaboration](#)).

Given the focus in these examples of computation and display of quantitative data, computational book publishing may seem most immediately applicable to publishing in STEM-focused academic disciplines but increasingly computational book publishing tools are being adopted in digital humanities as well as art and design disciplines like architecture.

Overview of available tools

The following linked table displays a list of current computational book publishing tools. The list is limited to software products that are under active maintenance (i.e. updated in the recent past).

	A	B	C	D	E	F	G	H	I	J
	Computational Publishing Tools	short description	developer(s)	website	code repository	license	scripting language	can include rich digital media?	can include live interactive code?	enables web publication?
1										
2		publishing frontend for Jupyter Notebook with an emphasis on data science publishing, enables easy inclusion of equations, formulas, graphs, figures, data visualizations, a suite of software and templates that produce static sites for book publications from Markdown documents using Jekyll, able to output as a website, PDF, and epub, able to incorporate a range of images, videos, and frames including complex mathematics using MathJax display engine for LaTeX notation.	Curvenote Inc.	https://curvenote.com/	https://github.com/curvenote/curvenote	MIT License	TypeScript	yes	yes	yes
3										
4		Electric Book	Electric Books Works	https://electricbookworks.github.io/electric-book/	https://github.com/electricbookworks/electric-book	GPL-3.0 License	JavaScript	yes	no	yes
5		static site generator using Jekyll to publish Git repositories as websites, easily adapted to host a website for a monograph publication with focus on versioning and branching integrates with books built in Jupyter Book.	GitHub, Inc.	https://pages.github.com/		MIT License	Ruby	yes	yes with additional work	yes
6		hosting for Git repositories which can be turned into static sites for monograph publications, able to include live code snippets which can be run in the browser, web application for creating and	GitLab Inc.	https://about.gitlab.com/	https://gitlab.com/gitlab-org/mtl-license		Ruby / Go / JavaScript	yes	yes with additional work	yes with additional work

Figure 24

Overview of Computational Publishing Tools: View this spreadsheet at <https://tiny.cc/copim-computational-tools>.

The list includes both tools for creating individual computational books and tools for publishing books with computational elements. Several of these tools, specifically the publishing tools, are not specially designed for computational book publishing but have been adapted to this use by authors. This includes static site generator tools like [GitHub Pages](#) and we'd like to note that there are [many static site generator tools](#) available that could also be adapted for publishing computational books however these may require more development skills than other tools highlighted.

Examples of Computational Books

Please find a variety of examples of Computational Books in this report's [Part 2: A Typology of Experimental Books](#).

The ExPub Compendium

To facilitate the kinds of experimental publishing and reuse discussed in this report, COPIM is developing an experimental publishing compendium and toolkit to be made live in Winter 2022/2023. The ExPub Compendium will be an online resource which provides an easy-to-browse catalogue of experimental publishing tools, practices, examples of experimental books, and the relationships between them.

With the Compendium, we aim to help researchers, designers, artists, or publishers who wish to publish experimental books by making it easier for them to discover the software or practices that can enable their experimental project. Currently, experimental book publishing projects tend to follow one of the following ways: either a bespoke solution is developed which can be prohibitively expensive for independent researchers or publishers, or a range of platforms and systems is tried out laboriously to find one that meets the authors' and publisher's requirements. The Compendium will provide a one-stop resource to help authors and publishers make decisions about what tools and platforms they can use for their specific experimental book publishing project.

While this report contains a static overview of tools for and practices to enable experimental book publishing, the ExPub Compendium will be an interactive database. The Compendium will contain an overview of the software and publishing tools discussed in this report alongside: an overview of and resources on experimental publishing practices such as annotation, collaborative writing, computational publishing, and versioning; sensitivities involved in experimental book publishing; a typology of experimental books; and examples of both experimental books and publishers of experimental books. We hope to provide inspiration and guidance for experimental book publications by linking these building blocks demonstrating how they might fit together. While tools feature prominently in the compendium, we are keen to showcase non-technical *ingredients* to raise awareness that tools alone don't make a publication.

The ExPub Compendium draws inspiration from similar online toolkit resources in the open licensing space. For design inspiration, we're grateful to [the Next Generation Libraries Project](#) and their [SComCat](#) resource, an online and openly available catalogue of open scholarly communication technologies. SComCat, developed in Ruby on Rails and made [open-source under an MIT License](#), provides an easy to search and easy to browse database of scholarly communications tools and platforms for a researcher or institution looking to use open software for scholarly communications. Their tagging and filtering search engine makes it easy to narrow down browsing to select only tools that meet a user's specific requirements. For content, we've looked at OAPEN's [Open Access Book Toolkit](#), an online guide to producing and publishing open access books offering individual articles on every stage of the typical research lifecycle for publishing open access. As described in this report's [first section](#), we also used Maxwell et al.'s [Mind The Gap](#) report, Lewis' 2020 [Bibliographic Scan of Digital Scholarly Communication Infrastructure](#), the Radical Open Access Collective's

[Information Portal: OA Publishing Tools](#), Kramer and Bosman's overview of [400+ Tools and innovations in scholarly communication](#), and Tennant et al.'s [Introducing Massively Open Online Papers \(MOOPs\)](#) for inspiration on open tools and practices.

Development of the ExPub Compendium has been a collaborative process involving all members of COPIM's WP6 project team as well as collaborations with COPIM's other work packages. The COPIM project also has toolkits and online resources as deliverables for WP2 (revenue infrastructures and management platform), WP3 (knowledge exchange and alternative business models), and WP7 (digital archiving and preservation). The project structure meant that the various work package teams were able to share ideas and best practices around the development of online resources and toolkits, helping to avoid unnecessary duplication of work.

Conclusion

This research and scoping report will develop further in instalments to incorporate both community feedback from the COPIM partners and other stakeholders (publishers, authors, technology developers) and updates in a rapidly changing technological landscape. We now have updated the examples listed in the experimental books typology section to include more non-English language examples from a wider geographical region.

We would very much like to invite comments and feedback on the release of this updated and expanded version of this report. We hope to be able to add further updates into subsequent versions of this report, while also feeding them into the ExPub Compendium, the online resource that we are now working to create and publish in COPIM's Year 3. All of this serves as a documentation of the process behind the establishment of this online resource and the thinking and decision-making informing it.

Works Cited

Adema, J., & Stone, G. (2017). *Changing publishing ecologies: A landscape study of new university presses and academic-led publishing* (p. 102). Jisc.
<http://doi.org/10.5281/zenodo.4420993>

Adema, J. (2019). The Ethics of Emergent Creativity: Can We Move Beyond Writing as Human Enterprise, Commodity and Innovation? In J. Jefferies & S. Kember (Eds.),

Whose Book is it Anyway? (pp. 65–90). Open Book Publishers.

<https://doi.org/10.11647/OBP.0159.03>

Adema, J., Mars, M., & Steiner, T. (2021). *Books Contain Multitudes: Exploring Experimental Publishing* (1st ed.). COPIM. <https://doi.org/10.21428/785a6451.933fa904>

Adema, J., Moore, S., & Steiner, T. (2021). *Promoting and Nurturing Interactions with Open Access Books: Strategies for Publishers and Authors* (1st ed.). Community-led Open Publication Infrastructures for Monographs (COPIM).

<https://doi.org/10.21428/785a6451.2d6f4263>

Adema, J., Moore, S., & Steiner, T. (2021). Part 1: Interaction in Context. In *Promoting and Nurturing Interactions with Open Access Books: Strategies for Publishers and Authors* (1st ed.). Community-led Open Publication Infrastructures for Monographs (COPIM). <https://doi.org/10.21428/785a6451.b021e5e7>

Andreessen, M. (1993). *WWW-Talk Apr-Jun 1993: Group annotation server guinea pigs?* <http://1997.webhistory.org/www.lists/www-talk.1993q2/0416.html>

Annotations. (2015). [Wiki]. Max Planck Institute for the History of Science. <https://it-dev.mpiwg-berlin.mpg.de/tracs/Annotations/wiki>

AnnotatorJS. (n.d.). *Showcase*. <https://annotatorjs.org/showcase.html>

Ambaras, D., & McDonald, K. (2021). *Bodies and Structures 2.0: Deep-Mapping Modern East Asian History*. <https://bodiesandstructures.org>

Aufderheide, P., & Jaszi, P. (2015). *Code of Best Practices in Fair Use for the Visual Arts. Copyright, Fair Use, Scholarly Communication, Etc.*

<https://digitalcommons.unl.edu/scholcom/1>

Ayers, P., & Klein, S. J. (2021). The Invisible Citation Commons. *Commonplace*, 1(1).

<https://doi.org/10.21428/6ffd8432.5af8c64c>

Barnes, L. (2018, October 22). *Copyright and licensing – what do I need to know?* Open Book Publishers Blog. <https://doi.org/10.11647/OBP.0173.0090>

Bates, M. (2014). *Conquering the Command Line*.

<http://conqueringthecommandline.com>

Bertino, A. C., & Staines, H. (2019). Enabling A Conversation Across Scholarly Monographs through Open Annotation. *Publications*, 7(2), 41.

<https://doi.org/10.3390/publications7020041>

Blansit, B. D. (2008). An Introduction to Cascading Style Sheets (CSS). *Journal of Electronic Resources in Medical Libraries*, 5(4), 395–409.

<https://doi.org/10.1080/15424060802453811>

Boluk, S., & LeMieux, P. (2017). *Metagaming: Playing, competing, spectating, cheating, trading, making, and breaking videogames*. University of Minnesota Press.

<https://manifold.umn.edu/projects/metagaming>

Borwick, C. (n.d.). *What is open access: Open access book publishing*. Retrieved 27 August 2021, from <https://library.bath.ac.uk/open-access/whatisopenaccess>

Brecht, B., & Silberman, M. (2020). *Brecht on film and radio*. Methuen.

<https://doi.org/10.5040/9781408185285>

Bush, V. (1939). *Mechanization and the record*, [Vannevar Bush Papers, Library of Congress], Box 138, Speech Article Book File.

Bush, V. (1945). 'As We May Think', *The Atlantic*, July 1945.

<https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/>

Capadisli, S., Guy, A., Verborgh, R., Lange, C., Auer, S., & Berners-Lee, T. (2017). Decentralised Authoring, Annotations and Notifications for a Read-Write Web with dokieli. In J. Cabot, R. De Virgilio, & R. Torlone (Eds.), *Web Engineering* (Vol. 10360, pp. 469–481). Springer International Publishing. https://doi.org/10.1007/978-3-319-60131-1_33

Chacon, S. (2014). *Pro Git* (Second edition). Apress.

<https://github.com/progit/progit2/releases/download/2.1.277/progit.pdf>

Chang, V., Mills, H., & Newhouse, S. (2007). *From Open Source to long-term sustainability: Review of Business Models and Case studies* (V. Chang, Ed.).

<https://eprints.soton.ac.uk/263925/>

Charoy, F. (2016, June 6). *Keynote: From group collaboration to large scale social collaboration*. 25th IEEE International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE-2016). <https://hal.inria.fr/hal-01342751>

Christen, K., Merrill, A., & Wynne, M. (2017). A Community of Relations: Mukurtu Hubs and Spokes. *D-Lib Magazine*, 23 (5/6). <https://doi.org/10.1045/may2017-christen>

Ciccarese, P. (2012, March 16). *Open Annotation Guiding Principles*. W3C Open Annotation Community Group. <https://www.w3.org/community/openannotation/open-annotation-guiding-principles/>

Ciston, S., & Marino, M. C. (2021, August 19). How to Fork a Book: The Radical Transformation of Publishing. *Medium*. <https://markcmarino.medium.com/how-to-fork-a-book-the-radical-transformation-of-publishing-3e1f4a39a66c>

Coko Foundation. (n.d.). *XSweet*. Retrieved 11 December 2020, from <https://xsweet.org/>

Collins, E., Milloy, C., & Stone, G. (2013). *Guide to Creative Commons for Humanities and Social Science monograph authors*. Jisc Collections. <https://eprints.hud.ac.uk/id/eprint/17828/>

Confederation of Open Access Repositories (COAR), & Next Generation Libraries Publishing. (2021). *SComCaT: Scholarly Communication Technology Catalogue*. <https://www.scomcat.net/>

Constant. (2020, October 6). *CC4r * Collective Conditions for Re-Use*. <https://constantvzw.org/wefts/cc4r.en.html>

Costanza-Chock, S. (2021). *Knowledge Justice: Disrupting Library and Information Studies through Critical Race Theory* (S. Y. Leung & J. R. López-McKnight, Eds.). The MIT Press. <https://doi.org/10.7551/mitpress/11969.001.0001> & <https://design-justice.pubpub.org/>

DeLisle, C. J. (2017, February 20). Time to Encrypt the Cloud. *CryptPad Blog*. <https://blog.cryptpad.fr/2017/02/20/Time-to-Encrypt-the-Cloud/index.html>

Di Donato, F., Morbidoni, C., Fonda, S., Piccioli, A., Grassi, M., & Nucci, M. (2013). Semantic annotation with Pundit: A case study and a practical demonstration. *Proceedings of the 1st International Workshop on Collaborative Annotations in Shared Environment Metadata, Vocabularies and Techniques in the Digital Humanities - DH-CASE '13*, 1-4. <https://doi.org/10.1145/2517978.2517995>

Dixon, D. (2017). Imagining the Essay as Digital Assemblage: Collaborative Student Experiments with Writing in Scalar. *Prompt: A Journal of Academic Writing*

Assignments, 1(1), Article 1. <https://doi.org/10.31719/pjaw.v1i1.13>

Fendt, K. (2018, December 30). Transition. *HyperStudio – Digital Humanities at MIT*.
<http://hyperstudio.mit.edu/blog/transition/>

Fidus Writer. (n.d.). *What is it?* <https://www.fiduswriter.org/how-it-works/>

Fitzpatrick, K. (2007a). *MediaCommons: Scholarly Publishing in the Age of the Internet*. <https://mcpress.media-commons.org/scholarlypublishing/>

Fitzpatrick, K. (2007b). CommentPress: New (Social) Structures for New (Networked) Texts. *The Journal of Electronic Publishing*, 10(3).
<https://doi.org/10.3998/3336451.0010.305>

Fitzpatrick, K. (2011). The Digital Future of Authorship: Rethinking Originality. *Culture Machine*, 12. <https://culturemachine.net/wp-content/uploads/2019/01/6-The-Digital-433-889-1-PB.pdf>

Fitzpatrick, K. (2019). *Generous Thinking*. Johns Hopkins University Press.
<https://jhupbooks.press.jhu.edu/title/generous-thinking>

Frosio, G. (2021). A brief history of Remix: From Caves to Networks. In E. Navas, O. Gallagher, & xtine burrough (Eds.), *The Routledge Handbook of Remix Studies and Digital Humanities* (pp. 19–35). Routledge. <https://www.routledge.com/The-Routledge-Handbook-of-Remix-Studies-and-Digital-Humanities/Navas-Gallagher-burrough/p/book/9780367361426>

FSF. (2009). *Introduction to the Command Line*. Free Software Foundation (FSF).
<http://archive.flossmanuals.net/command-line/>

Fulcrum. (2019). *What's New? A big title launch, some cool features, and news about ACLS HEB*. <https://www.fulcrum.org/blog/2019/10/24/big-title-launch-cool-features-acls-news/>

Garlan, D., Allen, R., & Ockerbloom, J. (1995). Architectural mismatch: Why reuse is so hard. *IEEE Software*, 12(6), 17–26. <https://doi.org/10.1109/52.469757>

Global Indigenous Data Alliance. (2019, September). *CARE Principles of Indigenous Data Governance*. Global Indigenous Data Alliance. <https://www.gida-global.org/care>

Ginsberg, D. (2010). Ways to Collaborate: Google and Beyond. *Presentations*.
https://scholarship.kentlaw.iit.edu/lib_pres/44

Git—About Version Control. (n.d.). <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>

Git—Contributing to a Project. (n.d.). <https://git-scm.com/book/en/v2/GitHub-Contributing-to-a-Project>

GitLab. (n.d.). *Administer GitLab Pages for self-managed instances*.
<https://docs.gitlab.com/ee/user/project/pages/#administer-gitlab-pages-for-self-managed-instances>

Grassi, M., Morbidoni, C., Nucci, M., Fonda, S., & Piazza, F. (2013). Pundit: Augmenting web contents with semantics. *Literary and Linguistic Computing*, 28(4), 640–659. <https://doi.org/10.1093/llc/fqt060>

Grosvenor, B. (2018, August 20). *Why museums should abolish image fees (ctd.)*. Art History News.
[https://www.arthistorynews.com/articles/5241_Why_museums_should_abolish_image_fees_\(ctd\)](https://www.arthistorynews.com/articles/5241_Why_museums_should_abolish_image_fees_(ctd))

Guston, D. H., Finn, E., & Robert, J. S. (Eds.). (2018). *Frankenbook*. Center for Science and the Imagination at Arizona State University, in partnership with The MIT Press and MIT Media Lab. <https://www.frankenbook.org/>

Halfpenny, S. (n.d.). *Subject Guides: Copyright: a Practical Guide: Using images*.
<https://subjectguides.york.ac.uk/copyright/images>

Hayles, N. K. (2002). *Writing machines*. MIT Press.
https://mitpress.mit.edu/sites/default/files/titles/content/mediawork/titles/writing/writing_book.html

Heller, L., The, R., & Bartling, S. (2014). Dynamic Publication Formats and Collaborative Authoring. In S. Bartling & S. Friesike (Eds.), *Opening Science* (pp. 191–211). Springer International Publishing. https://doi.org/10.1007/978-3-319-00026-8_13

Hex, M. (n.d.). *ImageCodr.org*. <https://www.imagecodr.org/>

HIRMEOS Consortium. (2019). *Annotation Service for Digital Monographs*.
https://web.archive.org/web/20201230170353/https://www.hirmeos.eu/wp-content/uploads/2019/06/HIRMEOS_Annotation_-Fact_sheet-Final-version.pdf

Hoe, N. S. (2006). *Free/Open Source Software—Open Standards*. United Nations Development Programme – Asia-Pacific Development Information Programme (UNDP-APDIP). <https://idl-bnc-idrc.dspacedirect.org/handle/10625/50703>

Home. (n.d.). Electronic Literature Directory. <https://directory.eliterature.org/>

Holvoet, K. (2006). What Is RSS and How Can Libraries Use It to Improve Patron Service? *Library Hi Tech News*, 23(8), 32–33.
<https://doi.org/10.1108/07419050610713718>

Horstmann, J. (2020). Undogmatic Literary Annotation with CATMA. In J. Nantke & F. Schlupkothén (Eds.), *Annotations in Scholarly Editions and Research* (pp. 157–176). De Gruyter. <https://doi.org/10.1515/9783110689112-008>

Hoya, B. (2010). Google Docs, EtherPad, and then some: Word processing and collaboration in today's portable work environment. *Texas Library Journal*, 86(2), 60–62.

Hutchins, B. I. (2021). A tipping point for open citation data. *Quantitative Science Studies*, 2(2), 433–437. https://doi.org/10.1162/qss_c_00138

Hypothes.is Historical Survey of Annotation Efforts. (n.d.). Google Docs.
<https://docs.google.com/spreadsheets/d/1f86L7vgHUW9wSLNNSunhjmtxtg6KlCOVpHGKbqUzWY/edit>

Jullien, N., Stol, K.-J., & Herbsleb, J. D. (2019). A Preliminary Theory for Open Source Ecosystem Micro-economics. In B. Fitzgerald, A. Mockus, & M. Zhou (Eds.), *Towards Engineering Free/Libre Open Source Software (FLOSS) Ecosystems for Impact and Sustainability*. Springer. <https://hal.archives-ouvertes.fr/hal-02127185>

Kalir, R., & Garcia, A. (2021). *Annotation*. The MIT Press.
<https://mitpressonpubpub.mitpress.mit.edu/annotation>

Kasprzak, D. M., & Smyre, T. (2017). Forerunners and Manifold: A Case Study in Iterative Publishing. *Journal of Scholarly Publishing*, 48(2), 90–98.
<https://doi.org/10.3138/jsp.48.2.90>

Kelty, C. (2014). Beyond Copyright and Technology: What Open Access Can Tell Us about Precarity, Authority, Innovation, and Automation in the University Today. *Cultural Anthropology*, 29(2), 203–215. <https://doi.org/10.14506/ca29.2.02>

Kim, E. (2020, October 21). How to Publish a Book with GitBook CLI and GitHub Pages in 7 Minutes. *Hackernoon*. <https://hackernoon.com/how-to-publish-a-book-with-gitbook-cli-and-github-pages-in-7-minutes-i61w3wjn>

Kleiner, D. (2010). *The telekommunist manifesto*. Institute of Network Cultures. https://www.networkcultures.org/_uploads/%233notebook_telekommunist.pdf

Knuth, D. (1992). 'Literate Programming', *Literate Programming: CSLI Lecture Notes*, no. 27. <https://www-cs-faculty.stanford.edu/~knuth/lp.html>

Kramer, B., & Bosman, J. (n.d.). *400+ Tools and innovations in scholarly communication*. Google Docs. <https://bit.ly/innoscholcomm-list>

Kreutzer, T. (2014). *Open Content: A practical guide to using creative commons licences*. German Commission for UNESCO. https://irights.info/wp-content/uploads/2014/11/Open_Content_A_Practical_Guide_to_Using_Open_Content_Licences_web.pdf

Kreutzer, T., & Lahmann, H. (2021). *Rechtsfragen bei Open Science—Ein Leitfaden* (2nd Ed.). Hamburg University Press. <https://doi.org/10.15460/HUP.211>

Lehman, P. (2010). *The Biblatex Package. Programmable Bibliographies and Citations*. <https://www.sys.kth.se/docs/texlive/texmf-dist/doc/latex/biblatex/biblatex.pdf>

Levine, A. (n.d.). *Flickr CC Attribution bookmarklet maker*. <https://cogdog.github.io/flickr-cc-helper/>

Lewis, D. W. (2020). *A Bibliographic Scan of Digital Scholarly Communication Infrastructure*. Educopia Institute. <https://educopia.org/mapping-the-scholarly-communication-landscape-bibliographic-scan/>

Manifold. (n.d.). *Reading Groups / Manifold Docs*. https://manifoldscholar.github.io/manifold-docusaurus/docs/using/reading_groups/

Manifold. (n.d.). *Community: Manifold is what we make it*. <https://manifoldapp.org/community>

Mars, M., Steiner, T., & Adema, J. (2021). Part 3: Technical Workflows and Tools for Experimental Publishing. In J. Adema, M. Mars, & T. Steiner, *Books Contain Multitudes: Exploring Experimental Publishing* (1st ed.). PubPub. <https://doi.org/10.21428/785a6451.174760b2>

Maxwell, J. W., 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*. PubPub.

<https://doi.org/10.21428/6bc8b38c.2e2f6c3f>

McKenzie, C. (2021). 'Interpreted vs. compiled languages: What's the difference?', *TheServerSide*, 15 April 2021. <https://www.theserverside.com/answer/Interpreted-vs-compiled-languages-Whats-the-difference>

Mercier, C. (2017, February 23). *Three recommendations to enable Annotations on the Web* / W3C News. <https://www.w3.org/blog/news/archives/6156>

Middlemarch 150th Anniversary Symposium. (2021). Manifold at the University of Washington. <https://uw.manifoldapp.org/projects/middlemarch-symposium-edition>

Microsoft. (2018, October 26). *Microsoft completes GitHub acquisition*. The Official Microsoft Blog. <https://blogs.microsoft.com/blog/2018/10/26/microsoft-completes-github-acquisition/>

OAPEN. (n.d.). *Funder requirements*. <https://www.oapen.org/funders/15172476-funder-requirements>

Odewahn, A. (2017). 'Computational Publishing with Jupyter', *GitHub*, 8 June 2017. <<https://github.com/odewahn/computational-publishing>>

Open Annotation Community Group. (n.d.). *Open Annotation Community Group*. <https://www.w3.org/community/openannotation/>

Open Syllabus: Explorer. (n.d.). Open Syllabus. <https://opensyllabus.org/>

Parikka, J. (2014). *The Anthrobscene*. University of Minnesota Press. <https://manifold.umn.edu/projects/the-anthrobscene>

Paul, E., McKenzie, D., Raibmon, P., & Johnson, H. (2019). *As I Remember It: Teachings (Pəms təʔaw) from the Life of a Sliammon Elder*. UBC Press. <https://doi.org/10.14288/SNS9-9159>

Pressbooks. (2013). *Enable Annotation with Hypothesis*. <https://guide.pressbooks.com/chapter/enable-annotation-with-hypothesis/>

Publish With Us. (2021). *RavenSpace*. <https://ravenspacepublishing.org/publish-with-us/>

Radical Open Access Collective. (n.d.). *Information Portal: OA Publishing Tools*.
<https://radicaloa.disruptivemedia.org.uk/resources/publishing-tools/>

Ross-Hellauer, T. (2017). *What is open peer review? A systematic review* (6:588).
F1000Research. <https://doi.org/10.12688/f1000research.11369.2>

Raymond, E. S. (1998). The cathedral and the bazaar. *First Monday*.
<https://doi.org/10.5210/fm.v3i2.578>

Ruckstuhl, K. (2022). Trust in Scholarly Communications and Infrastructure:
Indigenous Data Sovereignty. *Frontiers in Research Metrics and Analytics*, 6.
<https://doi.org/10.3389/frma.2021.752336>

Rudy, K. M. (2019, August 29). *The true costs of research and publishing*. Times
Higher Education (THE). <https://www.timeshighereducation.com/features/true-costs-research-and-publishing>

Salus, P. H. (1994). *A quarter century of UNIX*. Addison-Wesley Pub. Co.
<https://wiki.tuhs.org/lib/exe/fetch.php?media=publications:gcu.pdf>

Schweik, C. M. (2013). Sustainability in Open Source Software Commons: Lessons
Learned from an Empirical Study of SourceForge Projects. *Technology Innovation
Management Review*, 3(1), 13-19. <https://doi.org/10.22215/timreview/645>

Schultz, O. L. (2016). *after.video: Assemblages*.
<http://www.openhumanitiespress.org/books/titles/after-video/> &
<https://www.metamute.org/shop/openmute-press?page=1%2C0>

Shah, R. C., & Kesan, J. P. (2008). *Lost in Translation: Interoperability Issues for Open
Standards* [SSRN Scholarly Paper]. Social Science Research Network.
<https://papers.ssrn.com/abstract=1201708>

Shaw, Z. (2011). *The CLI Crash Course: Controlling Your Computer With The Terminal*.
samizdat. <https://library.memoryoftheworld.org/#/book/9223b1f6-cda7-469d-b7a2-fd32eb96cb7c>

Shukaitis, S. (2019). *Combination Acts*. Minor Compositions.

Signorini, G. F. (n.d.). *Open source and sustainability: The role of universities*.
<https://flore.unifi.it/handle/2158/1151000>

Singh, A. (n.d.). Claude McKay's Early Poetry (1911-1922): A Digital Collection.
<https://scalar.lehigh.edu/mckay/index>

Solid. (n.d.). <https://solid.mit.edu/>

Solid Project. (n.d.). *Home*. <https://solidproject.org/>

Soon, W., & Cox, G. (2020). *Aesthetic Programming: A Handbook of Software Studies*. Open Humanites Press. <http://www.openhumanitiespress.org/books/titles/aesthetic-programming/>

Soon, W. (2022). 'Experimental Publishing VI – Critique, Intervention, And Speculation' event held on 24 March 2022 and organised by Centre for Postdigital Cultures, Coventry University. <https://www.post-publishing.org/2022/02/23/experimental-publishing-vi-critique-intervention-and-speculation/>

Stallman, R. (2007). *Why Open Source misses the point of Free Software*.
<https://www.gnu.org/philosophy/open-source-misses-the-point.html.en>

Stallman, R. (2013, December 5). *FLOSS and FOSS*.
<https://www.gnu.org/philosophy/floss-and-foss.en.html>

Steiner, T. (2020, July 17). Using PubPub for scholarly output: Import, Citations, Collaboration, and Zotero. *COPIM*. <https://doi.org/10.21428/785a6451.f72bab42>

Tennant, J. P., Bielczyk, N., Tzovaras, B. G., Masuzzo, P., & Steiner, T. (2020). Introducing Massively Open Online Papers (MOOPs). *KULA: Knowledge Creation, Dissemination, and Preservation Studies*, 4(1), 1. <https://doi.org/10.5334/kula.63>

The Executable Book Project. (n.d.). *Documentation*.
<https://executablebooks.org/en/latest/>

The Mother of All Demos, presented by Douglas Engelbart (1968)—YouTube. (1968, December 9).
https://web.archive.org/web/20201210094618if_/https://www.youtube.com/watch?v=yJDv-zdhzMY

The Web Annotation Working Group. (2017, February 23). *Web Annotation Protocol: W3C Recommendation 23 February 2017*. <https://www.w3.org/TR/annotation-protocol/>

The State Library of New South Wales. (2017, May 1). *Connecting Aboriginal communities with collections and stories from the State Library of NSW*. Gather.

<https://gather.sl.nsw.gov.au/about>

TK Labels. (n.d.). *Local Contexts*. <https://localcontexts.org/labels/traditional-knowledge-labels/>

Traditional Knowledge. (2021, January 31). Program for Open Scholarship and Education, University of British Columbia. <https://pose.open.ubc.ca/open-access/author-rights/traditional-knowledge/>

Trettien, W. (2021). *Cut/Copy/Paste: Fragments from the History of Bookwork*. University of Minnesota Press. <https://manifold.umn.edu/projects/cut-copy-paste>

UK Copyright Literacy. (2016, July 11). Copyright guidance from UK universities and colleges. *UK Copyright Literacy*. <https://copyrightliteracy.org/about-2/copyright-guidance-from-uk-universities/>

Universities UK. (2019). *Open Access and Monographs: Evidence Review*. <https://web.archive.org/web/20210807082443/https://www.universitiesuk.ac.uk/policy-and-analysis/reports/Documents/2019/UUK-Open-Access-Evidence-Review.pdf>

University of Illinois Library, Libguides. (2021, July 7). *Scalar: An Introduction*. <https://guides.library.illinois.edu/c.php?g=348220&p=2348685>

W3C Digital Publishing Interest Group. (2014, December 4). *Digital Publishing Annotation Use Cases*. <https://www.w3.org/TR/dpub-annotation-uc/>

Watters, A. (2017, April 26). Un-Annotated. *Audrey Watters*. <http://audreywatters.com/2017/04/26/no-annotations-thanks-bye>

Wikipedia. (2020a). Point and click. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Point_and_click&oldid=990779820

Wikipedia. (2020b). WikiWikiWeb. In *Wikipedia*. <https://en.wikipedia.org/w/index.php?title=WikiWikiWeb&oldid=991654568>

Wikipedia. (2020c). Editor war. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Editor_war&oldid=993024851

Wikipedia. (2020d). Google Docs#Supported_file_formats. In *Wikipedia*. https://en.wikipedia.org/w/index.php?title=Google_Docs&oldid=993309823

Wikipedia. (2020e). Cryptographic hash function. In *Wikipedia*.

https://en.wikipedia.org/w/index.php?title=Cryptographic_hash_function&oldid=993402727

Wright, A. (2007). *Glut: Mastering Information Through the Ages*. London: Cornell University Press.

Wusteman, J. (2004). RSS: The latest feed. *Library Hi Tech*, 22(4), 404–413.

<https://doi.org/10.1108/07378830410570511>

Xie, Y. (n.d.). 6.3 Publishers / bookdown: Authoring Books and Technical Documents with R Markdown. <https://bookdown.org/yihui/bookdown/>

External resource

The bibliographies for all parts of this report are openly [available on Zotero](#).

Header image: Page spread of *Writing Machines* by Katherine Hayles and Anne Burdick. Hayles, N. K., Burdick, A., Loyer, E., Lunenfeld, P. (2002). [Writing machines](#). MIT Press.

Footnotes

1. see <https://hackmd.io/s/how-to-create-book> for a git- and &markdown-based approach to book creation ↵
2. Tennant et al. ‘It is necessary to be aware of and distinguish between commercial versus free and open-source software (FOSS) services and services that target a mainstream audience versus those that were developed for researchers, with research-specific features for data privacy and security, intellectual property protection measures, and licensing.’ (2020) ↵
3. As FSF founder Richard Stallmann puts it, Open Source ‘values mainly practical advantage and does not campaign for principles. This is why we [the Free Software movement] do not agree with open source, and do not use that term.’ (2007) ↵
4. See: https://en.wikipedia.org/wiki/Point_and_click ↵

5. For further information, see:
https://en.wikipedia.org/wiki/Google_Docs#Supported_file_formats [↵](#)
6. i.e., the lack of openly-accessible connectors combined with a reliance on proprietary standards and interfaces, which can often lead to a lock-in of users. See e.g. Hoe, N. S. (2006, p. 23ff.). [↵](#)
7. For a range of issues around the “pluggability” and interoperability of software, see e.g., [Garlan et al, 1995](#); and [Shah, Rajiv & Kesan, 2008](#). [↵](#)
8. For more information on this aspect, see e.g., [Kelty, 2014](#); and [Heller et al., 2014](#). [↵](#)
9. A toolchain that exemplifies this approach can be found at work in Coko’s XSweet. XSweet is a free, open source conversion tool for converting Word documents (.docx) into HTML and beyond. The tool is built as a pipeline of XSL transformations and designed to be modular, flexible, and extensible to support a wide variety of needs and workflows. More info: <https://xsweet.org/> [↵](#)
10. Three free and highly recommendable introductions to the command-line interface are [Shaw, 2011](#), [FSF, 2009](#), and [Bates, 2014](#). [↵](#)
11. ‘Version control is a system that records changes to a file or set of files over time so that you can recall specific versions later. For the examples in this book, you will use software source code as the files being version controlled, though in reality you can do this with nearly any type of file on a computer.’ ([Git, n.d.](#)) [↵](#)
12. In this context, we understand ‘synchronous collaboration’ as a mode that allows users to write, comment, and edit a given document while being logged in at the same time and together with others, while ‘asynchronous collaboration’ would mean a distributed approach in which individuals can work on a given text, but this cannot happen in the same temporal frame with others. [↵](#)
13. See Charoy ([2016](#)); and *The Mother of All Demos, presented by Douglas Engelbart (1968)* —[YouTube](#). [↵](#)
14. Cf. Hoya ([2010](#)); or Ginsberg ([2010](#)). [↵](#)
- 15.

Riseup is a volunteer-run collaborative which supports activist and other organisations. They provide many different services including running an EtherPad instance at: <https://pad.riseup.network>. Another reputable organisation running EtherPad is Framasoft (Lyon, France) with <https://framapad.org>.

Examples from the Higher Education context include *The Carpentries*, an Open Science-focused scholarly community that has a public instance running at <https://pad.carpentries.org>. Etherpad Lite is also quite popular among student union-based initiatives, see e.g., FU Berlin's [SplinePad](https://splinepad.org) or University of Hamburg's Computer Sciences student group pad at <https://ep.mafiasi.de/>.¹⁶

<https://github.com/hackmdio/codimd> aka [HackMD](https://hackmd.io) in its commercial, Software-as-a-Service (SAAS) branch.

As of December 2020, development on CodiMD [continues under the name of HedgeDoc](#).¹⁷

17. For a collection of instances, see: <https://flavoursofopen.science/community-run-open-source-tools-for-video-and-text-collaboration#hedge-doc>.

¹⁸

18. see https://en.wikipedia.org/wiki/Cryptographic_hash_function and DeLisle (2017) for an in-depth explanation of CryptPad's security features.¹⁹

19. cf. Git - Contributing to a project ([n.d.](#))²⁰

20. Atom was one of the first (desktop) text editors which used the web browser's rendering engine to process text. It allowed for customisation to be done in web technologies such as HTML, CSS, and Javascript, which have a much bigger developer base than any of the technologies needed to tweak and customise Vim and/or Emacs.²¹

21. Electron made easier installation/deployment of applications written in Javascript possible, which are initially mostly developed as a web app. Electron allowed for web development while getting the benefits of being installed as a desktop application, including having access to system services.²²

22. EMACS and Vi/Vim are two of the longest-standing CLI-based text editor families, with differences in underlying philosophies regarding information processing, which

in turn affects the way one uses these editors. Their evolution has been accompanied by long-standing disputes between programmers, with supporting factions of each editor engaging in heavy disputes around the benefits and shortfalls of each text editor over the other, which in the last thirty years led to a number of ‘Editor Wars’ in the open source community. For more information on this, see:

https://en.wikipedia.org/wiki/Editor_war

23. The Cascading Style Sheets (CSS) language is a way to describe the presentation features of a document, so to facilitate cross-platform representation of layout and styling properties that can then be read by reader applications, including web browsers and publishing tools. The CSS standard is a corner stone of the World Wide Web architecture. For more, see e.g. Blansit (2008). [↗](#)

24. ‘full-stack’ here refers to the package containing all individual steps and granular tools to make use of the proposed workflow (as opposed to relying on external tools not included in the tool stack). [↗](#)

25. Wax is currently [under active development](#) and installable via standard Javascript developers' tools. At the time of writing, it is not recommended to use Wax as a standalone text editor, but rather as part of a wider framework such as Editoria.

[↗](#)

26. cf. e.g. Lehman (2010). [↗](#)

27. see GitLab ([n.d.](#)) [↗](#)

28. To date, jamstack.org lists 314 different static site generator variants at <https://jamstack.org/generators/> [↗](#)

29. see e.g. Xie ([n. d.](#)), and Kim (2020) for a quick introduction, plus the [Executable Book Project](#). [↗](#)

30. see e.g. Marc Andreessen’s [first steps to test annotation with the Mosaic browser](#).

[↗](#)

31. although first steps have been taken with the formal introduction of a [W3C’s Web Annotation model in 2017](#). [↗](#)

32. see Holvoet (2006) or Wusteman (2004) for the early promises that RSS was envisioned to offer for libraries. [↗](#)

33. see <https://solid.mit.edu/> and <https://solidproject.org/about>. [↗](#)

34. if not actively discouraged, see e.g., [Audrey Watters, 2017](#) [↵](#)
35. See <https://hypothes.is/annotating-all-knowledge/> and the [FORCE11 Annotating All Knowledge](#) working group. [↵](#)
36. <https://mitpress.mit.edu/books/annotation> [↵](#)
37. See <https://web.hypothes.is/tools-plugin-ins-and-integrations/> [↵](#)
38. See the HIRMEOS Annotation service technical specifications at [10.5281/zenodo.1343519](https://zenodo.org/record/1343519) [↵](#)
39. For a presentation of Fulcrum’s capabilities, see e.g., <https://vimeo.com/390508545> [↵](#)
40. See <https://pressbooks.org/> and <https://pressbooks.directory/> for a wide selection of examples of research monographs as well as textbooks.

And e.g., <https://guide.pressbooks.com/chapter/hypothesis-for-webbook-annotation-comments/> for Pressbooks’ implementation of hypothes.is. [↵](#)

41. See e.g., Grassi et al. 2013, and Di Donata et al., 2013. [↵](#)
42. Pundit will become part of the GOTRIPLE platform, see <https://www.youtube.com/watch?v=v9a6vDQYf4Q> [↵](#)
43. Login is possible via Facebook, Google, and the federated EGI service, which in turn offers identification via ORCiD, the eduGAIN consortium and its participating HEI’s logins, GitHub, Bitbucket, LinkedIn, WeChat, Elixir, etc. [↵](#)
44. See <https://web.archive.org/web/20200910111747/https://github.com/net7/pundit2> for an archived version of Pundit 2.0.

At the time of writing this report (April 2022), Pundit’s license page states that both “[Pundit Annotator and Annotator Pro are released under a AGPL 3.0 license](#)”. [↵](#)

45. See Scalar’s User Guide, particularly sections “[Annotating Media](#)”, and “[Whole-Part Relationships](#)” [↵](#)
46. See Scalar User Guide, “[Working with the API](#)” [↵](#)

47. Examples considered here include annotations added to Parikka's *The Anthrobscene* (2014) [here](#), to Boluk and LeMieux's *Metagaming* (2017) [here](#), and to the Middlemarch 150th Anniversary Symposium's collaborative edition (2021) [here](#). An overview of the more than 30 presses and publishing initiatives can be found on Manifold's [Community](#) page [↵](#)
48. See e.g., CUNY's [Quick Guide: Getting Started for Students](#), or University of Washington Libraries' [Manifold Pilot Guide](#). [↵](#)
49. "The effort focuses on interoperability for annotations. Its goal is to allow the sharing of annotations across clients, servers, and applications. It will not, in any way, prescribe user interfaces, internal architectures or internal data structures." ([W3C](#)) [↵](#)
50. See e.g., the excellent [Pathways to PubPub](#) document that provides guidance and illustrative examples for MIT Press authors and editors using the platform for their publishing workflows. [↵](#)
51. See git-based collaborative text editing in Mars et al. 2021, and PubPub's in-editor versioning approach described in more detail in [Steiner 2020](#). [↵](#)
52. It might be noted here that while git is the most popular of version control systems, other forms such as Subversion, RCS, CVS, or Mercurial might also be amenable to setting up a publishing infrastructure (although we don't know of any practical examples that use these ecosystems for the purpose of digital book publishing). [↵](#)

Citations

1. Adema, J., Moore, S., & Steiner, T. (2021). *Promoting and Nurturing Interactions with Open Access Books: Strategies for Publishers and Authors* (First). Community-led Open Publication Infrastructures for Monographs (COPIM). <https://doi.org/10.21428/785a6451.2d6f4263> [↵](#)
2. Maxwell, J. W., 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*. <https://doi.org/10.21428/6bc8b38c.2e2f6c3f> [↵](#)
3. Prospects. (2019). In *Mind the Gap: A Landscape Analysis of Open Source Publishing Tools and Platforms* (1st ed.). <https://doi.org/10.21428/6bc8b38c.2e2f6c3f>

↵

4. Prospects. (2019). In *Mind the Gap: A Landscape Analysis of Open Source Publishing Tools and Platforms* (1st ed.). <https://doi.org/10.21428/6bc8b38c.2e2f6c3f>

↵

5. Prospects. (2019). In *Mind the Gap: A Landscape Analysis of Open Source Publishing Tools and Platforms* (1st ed.).

<https://mindthegap.pubpub.org/pub/7odrns2t/release/2#concluding-thoughts> ↵

6. Confederation of Open Access Repositories (COAR) and Next Generation Libraries Publishing. (n.d.). *SComCaT: Scholarly Communication Technology Catalogue*.

<https://www.scomcat.net/> ↵

7. Kramer, B., & Bosman, J. (n.d.). 400+ Tools and innovations in scholarly communication. In *Google Docs*. <https://bit.ly/innoscholcomm-list> ↵

8. Tennant, J. P., Bielczyk, N., Tzovaras, B. G., Masuzzo, P., & Steiner, T. (2020). Introducing Massively Open Online Papers (MOOPs). *KULA: Knowledge Creation, Dissemination, and Preservation Studies*, 4(1). <https://doi.org/10.5334/kula.63> ↵

9. Setting Context. (2019). In *Mind the Gap: A Landscape Analysis of Open Source Publishing Tools and Platforms* (1st ed.).

<https://mindthegap.pubpub.org/pub/gei072ab/release/2#what-is-open-source-software> ↵

10. Raymond, E. S. (1998). The cathedral and the bazaar. *First Monday*, 3(2).

<https://firstmonday.org/ojs/index.php/fm/article/view/578> ↵

11. Salus, P. H. (1994). *A quarter century of UNIX*. Addison-Wesley Pub. Co.

<https://wiki.tuhs.org/lib/exe/fetch.php?media=publications:gcu.pdf> ↵

12. Adema, J., & Stone, G. (2017). *Changing Publishing Ecologies: A Landscape Study of New University Presses and Academic-Led Publishing*. Jisc.

<https://doi.org/10.5281/zenodo.4420993> ↵

13. Microsoft. (2018). Microsoft completes GitHub acquisition. In *The Official Microsoft Blog*. <https://blogs.microsoft.com/blog/2018/10/26/microsoft-completes-github-acquisition/> ↵

14. Fidus Writer. (n.d.). *What is it?* <https://www.fiduswriter.org/how-it-works/> ↵

15. Brecht, B., & Silberman, M. (2020). *Brecht on film and radio*. Methuen.
<https://doi.org/10.5040/9781408185285> ↵
16. Capadisli, S., Guy, A., Verborgh, R., Lange, C., Auer, S., & Berners-Lee, T. (2017). Decentralised Authoring, Annotations and Notifications for a Read-Write Web with dokieli. In J. Cabot, R. De Virgilio, & R. Torlone (Eds.), *Web Engineering* (Vol. 10360, pp. 469–481). Springer International Publishing. https://doi.org/10.1007/978-3-319-60131-1_33 ↵
17. Adema, J., Moore, S., & Steiner, T. (2021). Part 1: Interaction in Context. In *Promoting and Nurturing Interactions with Open Access Books: Strategies for Publishers and Authors* (First). Community-led Open Publication Infrastructures for Monographs (COPIM). <https://doi.org/10.21428/785a6451.b021e5e7> ↵
18. W3C Digital Publishing Interest Group. (2014). *Digital Publishing Annotation Use Cases*. <https://www.w3.org/TR/dpub-annotation-uc/> ↵
19. Ross-Hellauer, T. (2017). *What is open peer review? A systematic review* (Issue 6:588). <https://doi.org/10.12688/f1000research.11369.2> ↵
20. Adema, J., Mars, M., & Steiner, T. (2021). *Books Contain Multitudes: Exploring Experimental Publishing* (First). COPIM.
<https://doi.org/10.21428/785a6451.933fa904> ↵
21. Mars, M., Steiner, T., & Adema, J. (2021). Part 3: Technical Workflows and Tools for Experimental Publishing. In *Books Contain Multitudes: Exploring Experimental Publishing* (1st ed.). PubPub. <https://doi.org/10.21428/785a6451.174760b2> ↵
22. Kalir, R., & Garcia, A. (2021). *Annotation*. The MIT Press.
<https://mitpressonpubpub.mitpress.mit.edu/annotation> ↵
23. Bertino, A. C., & Staines, H. (2019). Enabling A Conversation Across Scholarly Monographs through Open Annotation. *Publications*, 7(2), 41.
<https://doi.org/10.3390/publications7020041> ↵
24. Dixon, D. (2016). Imagining the Essay as Digital Assemblage: Collaborative Student Experiments with Writing in Scalar. *Prompt: A Journal of Academic Writing Assignments*, 1(1). <https://doi.org/10.31719/pjaw.v1i1.13> ↵
25. Kasprzak, D. M., & Smyre, T. (2017). Forerunners and Manifold: A Case Study in Iterative Publishing. *Journal of Scholarly Publishing*, 48(2), 90–98.

<https://doi.org/10.3138/jsp.48.2.90>

26. Manifold. (n.d.). *Reading Groups* | *textbar Manifold Docs*.

https://manifoldscholar.github.io/manifold-docusaurus/docs/using/reading_groups/

27. Fitzpatrick, K. (2007). CommentPress: New (Social) Structures for New (Networked) Texts. *The Journal of Electronic Publishing*, 10(3).

<https://doi.org/10.3998/3336451.0010.305>

28. Fitzpatrick, K. (2007). *MediaCommons: Scholarly Publishing in the Age of the Internet*. <https://mcpres.media-commons.org/scholarlypublishing/>

29. Fitzpatrick, K. (2019). *Generous Thinking* | *textbar Johns Hopkins University Press Books*. Johns Hopkins University Press.

<https://jhupbooks.press.jhu.edu/title/generous-thinking>

30. *Publish With Us* – *RavenSpace*. (2021). <https://ravenspacepublishing.org/publish-with-us/>

31. Christen, K., Merrill, A., & Wynne, M. (2017). A Community of Relations: Mukurtu Hubs and Spokes. *D-Lib Magazine*, 23(5/6).

<https://doi.org/https://doi.org/10.1045/may2017-christen>

32. Constant. (2020). *CC4r * COLLECTIVE CONDITIONS FOR RE-USE*.

<https://constantvzw.org/wefts/cc4r.en.html>

33. Program for Open Scholarship and Education. (2021). Traditional Knowledge. In *Program for Open Scholarship and Education*. <https://pose.open.ubc.ca/open-access/author-rights/traditional-knowledge/>

34. Adema, J. (2019). 3. The Ethics of Emergent Creativity: Can We Move Beyond Writing as Human Enterprise, Commodity and Innovation? In J. Jefferies & S. Kember (Eds.), *Whose Book Is It Anyway?* (pp. 65–90). Open Book Publishers.

<https://doi.org/10.11647/OBP.0159.03>

35. Frosio, G. (2021). A brief history of Remix: From Caves to Networks. In E. Navas, O. Gallagher, & xtine burrough (Eds.), *The Routledge Handbook of Remix Studies and Digital Humanities* (pp. 19–35). Routledge.

<https://www.routledge.com/The-Routledge-Handbook-of-Remix-Studies-and-Digital-Humanities/Navas-Gallagher-burrough/p/book/9780367361426>

36. Hutchins, B. I. (2021). A tipping point for open citation data. *Quantitative Science Studies*, 2(2), 433–437. https://doi.org/10.1162/qss_c_00138
37. Ayers, P., & Klein, S. J. (2021). The Invisible Citation Commons. *Commonplace*, 1(1). <https://doi.org/10.21428/6ffd8432.5af8c64c>
38. Global Indigenous Data Alliance (GIDA). (2019). *CARE Principles of Indigenous Data Governance*.
39. Ruckstuhl, K. (2022). Trust in Scholarly Communications and Infrastructure: Indigenous Data Sovereignty. *Frontiers in Research Metrics and Analytics*, 6. <https://doi.org/10.3389/frma.2021.752336>
40. Universities UK. (2019). *Open Access and Monographs: Evidence Review*. <https://web.archive.org/web/20210807082443/https://www.universitiesuk.ac.uk/policy-and-analysis/reports/Documents/2019/UUK-Open-Access-Evidence-Review.pdf>
41. Soon, W., & Cox, G. (2021). *Aesthetic Programming: A Handbook of Software Studies*. Open Humanities Press. <http://www.openhumanitiespress.org/books/titles/aesthetic-programming/>
42. Ciston, S., & Marino, M. C. (2021). How to Fork a Book: The Radical Transformation of Publishing. In *Medium*. <https://markcmarino.medium.com/how-to-fork-a-book-the-radical-transformation-of-publishing-3e1f4a39a66c>
43. Adema, J. (2021). Versioning and Iterative Publishing. *Commonplace*. <https://doi.org/10.21428/6ffd8432.42408f5b>
44. Bush, V. (1939). *Mechanization and the record*.
45. Bush, V. (1945). As We May Think. *The Atlantic Monthly*. <https://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/>
46. Wright, A. (2007). *Glut: Mastering Information Through the Ages*. Cornell University Press.
47. Knuth, D. E. (1992). Literate Programming. In *Literate Programming* (Issue 27, p. xvi+368pp.). Center for the Study of Language and Information. <https://www-cs-faculty.stanford.edu/~knuth/lp.html>
48. McKenzie, C. (2021). Interpreted vs. compiled languages: What's the difference? In *TheServerSide*. <https://www.theserverside.com/answer/Interpreted-vs-compiled->

[languages-Whats-the-difference↵](#)

49. Odewahn, A. (2017). Computational Publishing with Jupyter. In *GitHub*.

<https://github.com/odewahn/computational-publishing>. ↵

50. Soon, W. (2022, March). ↵