



## Start-up story

# Janeway: a scholarly communications platform

It is a terrible mistake to think that technological platforms are what make or break a scholarly communications enterprise. Yet, the technology that drives submissions and article presentation does sit at the heart of user interaction. Technologies also have knock-on effects for running a publishing business. If you run a legacy system that requires three senior engineers who are all proficient in six different technological stacks, then your staff costs will be substantially higher.

Bearing these considerations in mind, the Centre for Technology and Publishing at Birkbeck, University of London embarked upon building a journal submission and hosting platform, Janeway, learning from our experiences of running the Open Library of Humanities (OLH). We knew Open Journal Systems (OJS) well at this time, but were not big fans of PHP, the language in which it is written. We also were aware of the work being done by Coko (the Collaborative Knowledge Foundation) in Node.js. What we really craved, though, was a scholarly communications platform written in Python/Django. This was not just a language preference but was also linked to issues around hiring and simplicity, as mentioned above. Python was the most popular programming language in 2017, which would mean that a platform in this language would be comprehensible to a wide range of programmers.<sup>1</sup> So, we chose to write in Python using the Django framework, since this is a well-known, stable, and secure framework for the development of web applications. The platform's ongoing development is funded by a combination of revenue streams: from the OLH, which has begun using the platform as part of a mixed economy of suppliers, and from hosting services that we are providing to others.

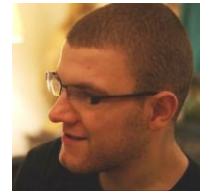
Upon setting out, we began with a social perspective: what is the workflow that we need? The OLH runs a mostly double-blind review process and we wanted an effective means to oversee this, but with enough flexibility to adapt to new models of review – such as open review – that are under active trial around the world. We built a workflow design that is shown in Figure 1 (below). Note that, although this is centred around an 'issue'-type model, Janeway also provides support for continuous publication.

Rather than adopting Coko's more ambitious goal to build a workflow creation tool, we decided to opt for an extensible workflow model. In Janeway, we built a standard workflow, as per Figure 1. However, in a forthcoming feature, plug-ins will be able to 'hijack' the main workflow and divert the user down a different track – say, for book submissions – in future.

We also had a number of technical design principles behind the codebase design of Janeway:

- No code should appear to work 'by magic'. Readability is key.
- Testing will be applied to security modules and whenever a post-launch bug fix is committed. We do not aim for total testing but selective regression testing.
- Security bugs jump the development queue and are a priority.
- We will never accept commits of, or ourselves write, paywall features into Janeway.

These community standards are designed to make it easy for others to delve into the codebase; to ensure that we put our efforts into ensuring the security of the platform, as journals become more of a target for hackers; and also to publicly commit ourselves to the open access nature of the platform. We also decided that Janeway should be openly licensed (AGPL v3) and developed in the open, allowing anyone to comment on, make feature suggestions for, or contribute to the codebase.



MARTIN PAUL EVE

Professor of Literature, Technology and Publishing  
Birkbeck, University of London



ANDY BYERS

Senior Publishing Technology Developer  
Birkbeck, University of London

'we began with a social perspective: what is the workflow that we need?'

'Janeway should be openly licensed (AGPL v3) and developed in the open'

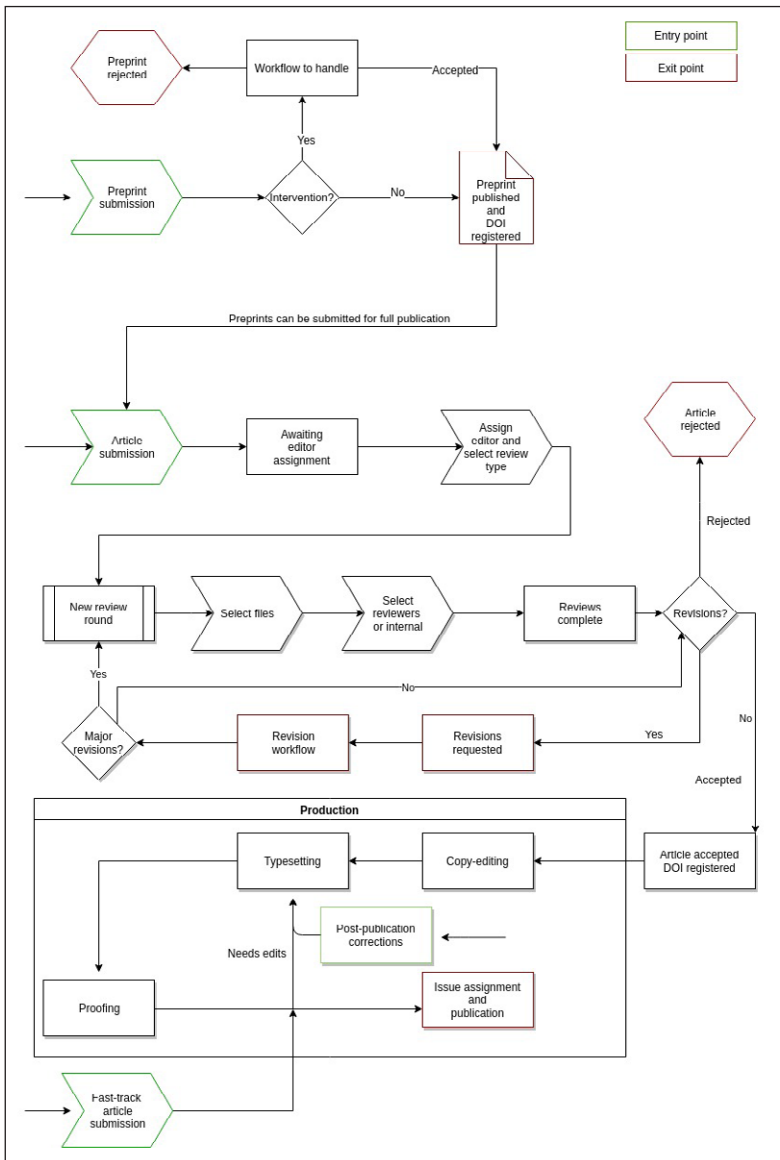


Figure 1. Janeway's default workflow

An instance of the platform itself consists of several components:

- a single press (homepage shown in Figure 2)
- any number of journals
- a preprints area (optionally enabled/disabled)
- a back-end manager interface
- an administrative interface.

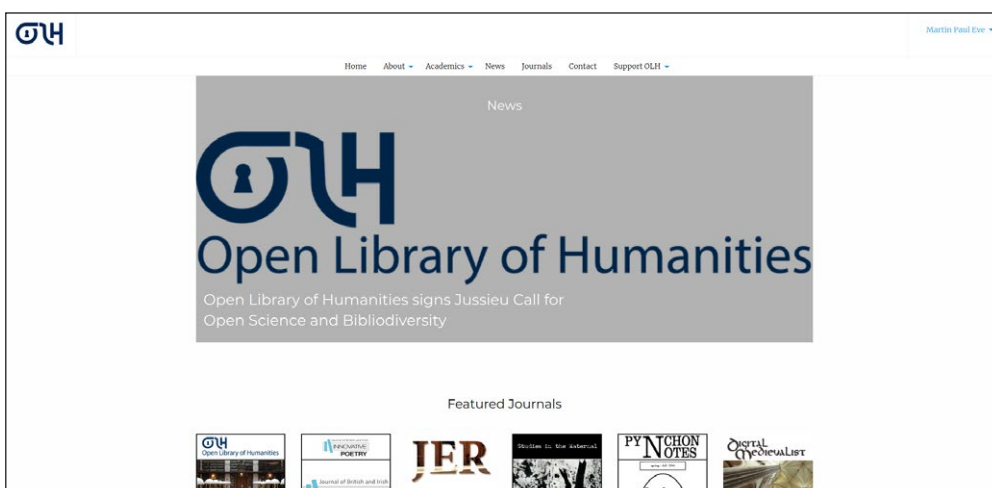


Figure 2. An example of a Janeway homepage

- 3 Each journal install – which can be added with the click of a button – features a full document and peer-review management system, integration with Crossref DOIs, COUNTER-compliant logging (with reporting in development), ORCID sign-in, Crossref similarity check (for plagiarism detection), an e-mail back end that can use Mailgun, OAI-PMH feeds, and overlay journal functionality.

This last function is a particular area of excitement for us. One of the authors of this piece has argued for some years now that the ability to curate tables of contents across journals – and to thereby signal value through editorial selection, rather than journal title – is an area of important development for scholarly communications.<sup>2</sup>

Janeway now has this feature by default, where a remote article can be curated into a table of contents.

‘the ability to curate tables of contents across journals’

Also pointing the way to the future for us is the preprint server functionality. This is an extremely new feature and about to undergo extensive testing at the time of writing. However, the basic idea here is to have a workflow that allows users to solicit comments from their peers on a draft version in the open before providing seamless submission to any journal on the platform. That is: no re-entering metadata, just a one-click ‘submit to journal’ option. This is a useful feature since many humanities disciplines have a history of circulating working papers for comment before submission. We are simply replicating that community of peers reading each other’s work for the digital age.

‘no re-entering metadata, just a one-click ‘submit to journal’ option’

Janeway also comes with a series of plug-ins that we have released:

- back content plug-in: this allows for the loading of back-content articles into the system
- supporters plug-in: this allows those who run a collective subsidy/consortial business model to manage the list of supporting institutions/billing etc.
- annotators plug-in: this inserts a link to enable the hypothesis sidebar for annotation of scholarly articles.

Plug-ins can, in theory, indefinitely extend the functionality of the platform. For instance, we are working on a book submission and display platform that will allow those running monograph presses to handle these object types. There is also an advanced ‘notification’ system built into the codebase, so plug-ins can ask to be notified when specific types of event happen, such as submission of a new article.

Overall, though, Janeway helps the OLH to have control of its own platform, to build features for unique journals when required, and to commit to our open principles by sharing the software for others to use. We hope that others will assimilate the codebase into their own projects.

#### Abbreviations and Acronyms

A list of the abbreviations and acronyms used in this and other *Insights* articles can be accessed here – click on the URL below and then select the ‘Abbreviations and Acronyms’ link at the top of the page it directs you to: <http://www.uksg.org/publications#aa>

#### Competing interests

The authors have declared no competing interests.

#### References

1. Hruska J, Python Tops 2017’s Most Popular Programming Languages, *ExtremeTech*, 2017: <https://www.extremetech.com/computing/252987-python-tops-list-2017s-popular-programming-languages> (accessed 10 January 2018).
2. Eve M P, *Open access and the humanities: contexts, controversies and the future*, 2014, Cambridge, Cambridge University Press: DOI: <https://doi.org/10.1017/CBO9781316161012>, pp. 146–8 (accessed 10 January 2018).

Article copyright: © 2018 Martin Paul Eve and Andy Byers. This is an open access article distributed under the terms of the [Creative Commons Attribution Licence](#), which permits unrestricted use and distribution provided the original author and source are credited.



Corresponding author  
Martin Paul Eve  
Professor of Literature, Technology and Publishing  
Birkbeck, University of London, GB  
E-mail: [martin.eve@bbk.ac.uk](mailto:martin.eve@bbk.ac.uk)

ORCID ID: <http://orcid.org/0000-0002-5589-8511>

To cite this article:

Eve M P and Byers A, Janeway: a scholarly communications platform, *Insights*, 2018, 31: 15, 1–4; DOI: <https://doi.org/10.1629/uksg.396>

Published by UKSG in association with Ubiquity Press on 16 May 2018