2018

Community Code: Supporting the Mission of Open Access and Preservation with the Use of Open Source Library Technologies

Keila Zayas-Ruiz and Mark Baggett

This book chapter was published in Applying Library Values to Emerging Technology: Decision-Making in the Age of Open Access, Maker Spaces, and the Ever-Changing Library (Publications in Librarianship #72) by the Association of College & Research Libraries. This work is licensed under a Creative Commons Attribution 4.0 License, CC BY (https://creativecommons.org/licenses/by/4.0/).
Community Code: 
Supporting the Mission of Open Access and Preservation with the Use of Open Source Library Technologies

Keila Zayas-Ruiz*
Sunshine State Digital Network Coordinator
Strozier Library, Florida State University

Mark Baggett
Department Head, Digital Initiatives
Hodges Library, University of Tennessee, Knoxville

Introduction

As librarians, we serve as champions for equal access and preservation of materials, both scholarly and cultural in significance. One of the core missions of libraries is access. Due to increased demand for scholarly articles and the technological advances of the internet, open access is quickly becoming a major priority among research libraries today. It “has expanded the possibilities for disseminating one’s own research and accessing that of others.” The movement of open access aligns closely with the ALA core value of access as outlined by the ALA council: “All

*This work is licensed under a Creative Commons Attribution 4.0 License, CC BY (https://creativecommons.org/licenses/by/4.0/).
information resources that are provided directly or indirectly by the library, regardless of technology, format, or methods of delivery, should be readily, equally, and equitably accessible to all library users.” It has gained considerable momentum in recent years among librarians in institutions of higher education, spurring funds dedicated to support authors who wish to publish in open access journals, the creation of library-run online open access journals, and open access mandates for faculty members.

Another core mission of libraries and archives is to protect and preserve data so that it exists as far into the future as possible. While this concept originally applied to the physical objects that libraries and archives maintain, this idea was applied to also include digital objects as institutions began to collect born-digital content and to digitize its intellectual and cultural resources. In 2000, the United States Congress selected the Library of Congress to lead the National Digital Information Infrastructure and Preservation Program in order “to develop a national approach to digital preservation.” In 2001, the American Library Association revised its preservation policy to recognize the importance of “long-term persistence and usability of digital content.” That same year, the Digital Preservation Coalition (DPC) was founded to address “the urgent challenges of securing the preservation of digital resources in the United Kingdom and to work with others internationally to secure our global digital memory and knowledge base.”

In reacting to the calls to develop standards and approaches for access to scholarly works and digital preservation, libraries and archives began to develop best practices within the discipline. As new standards and best practices were produced, institutions began to recognize they needed software to solve challenges diagnosed by the community. In order to address this, institutions began to collaborate throughout the information science community to develop new software and improve existing open-source technologies in order to make research openly available and preserve valuable cultural resources for perpetuity.

The interest in open source technologies by libraries should not come as a surprise to most. The missions of libraries and open-source software (OSS) initiatives are very similar. OSS is based on sharing and collaboration, as well as the free and open exchange of intellectual property. “Open source software is a public good: Its use is non-rival, and it involves a copyright-based license to keep private intellectual property claims out of the way of both software innovators and software adopters—while at the same time preserving a commons of software code that everyone can access.” These values sound familiar to the values of librarianship as discussed above.

OSS projects attract a community of developers, documenters, testers, reporters, and users. As the digital preservation and digital assets management needs of libraries and archives continue to grow and the complexity of these
needs continue to increase, institutions have recognized that they can more easily implement and develop new features around OSS by adopting and participating in community frameworks. A popular African proverb states, “If you want to go quickly, go alone. If you want to go far, go together.” This idea illustrates that you can often get more out of an OSS by participating in a community around it and adopting or developing functionality that is needed universally rather than building your own architecture to address a few functional requirements. It is often referenced in open-source communities to illustrate the idea that you can more efficiently develop what you need by going more slowly and working with more stakeholders. It has the additional benefit of aiding smaller institutions with more limited resources to be able to use the knowledge, expertise, and manpower of a community effort bolstered by larger institutions with more resources. This reinforces the value of contributing to the public good upheld by both open source movements and libraries.

Open Access Publishing

“One of the major barriers for scholars and researchers in universities is the lack of access to the current literature in their subject, much of which may be published in journals that have high annual subscription rates and so are far too expensive for many libraries.”7 “Open Access is the free, immediate, online availability of research articles combined with the rights to use these articles fully in the digital environment.”8 The first inklings of open access began with the Gutenberg project in 1971 when Michael Hart began making electronic texts of books freely available to the public.9 By the 1990s, with the advancement of the internet, open access became a full-fledged idea and open access journals began appearing as an alternative to the common practice of sharing scientific research in electronic mailing lists.10 An example of how much open access has grown since this time can be seen in the Directory of Open Access Journals (DOAJ). Started in 2003 at Lund University in Sweden with three hundred open access journals, there are now ninety-five hundred with more than two million articles covering all areas of science, technology, medicine, social science, and humanities and spanning one hundred twenty-nine countries.11

There are several different types of open access which include:

- **Green Open Access**: the self-archiving of articles in an institutional or discipline-specific repository. An author can usually submit either a pre-print, post-print, or publisher’s version of their article based on the licensing agreement with their publisher. Sometimes the agreement will also stipulate a delay or embargo on the availability of the self-archived copy of an article.
- **Gold Open Access**: the immediate open access availability of an article directly through the publisher. This can be done through a fully open access journal or a hybrid journal. Hybrid journals allow authors to pay a fee for their article to be published as open access in a subscription-only journal.\(^\text{12}\)

- **Delayed Open Access**: the journal makes their content freely available after a certain amount of time has passed (usually after twelve months), but charges a subscription fee or access fee for newer content. There is “divisive acceptance of delayed OA as a valid form of open access.”\(^\text{13}\)

Because these journals primarily publish online content, the costs associated with publication and distribution are notably lower than those of their print counterparts. This enables them to make their content freely available by operating under alternative business models. These are some of the most common business models adopted by open access publishers:

- **Community publishing** is more commonly adopted by smaller, niche areas of research. This model is supported by using volunteers for peer-review, editing, and publication.\(^\text{14}\)

- **Advertising sponsorship** generally attracts advertising from employers, conference organizers, and other publishers. Often the revenue from advertisements is not enough to sustain the journal completely but acts as a valuable subsidy.\(^\text{15}\)

- **Institutional subsidy** occurs when an institution formally subsidizes a university press or publishing entity most often operated by the library.\(^\text{16}\)

- **Hard copy sales** rely on sales of print versions to libraries as the main source of revenue.\(^\text{17}\)

- **Article processing charges** may be collected from authors, their institutions, or their funding agencies up front. It is one of the more sustainable models for OA revenue, and some institutions and funders have allocated dedicated funds for this purpose.\(^\text{18}\)

- **Institutional memberships** allow institutions to pay a lump sum annually in advance for articles that their authors will publish that year in their journal.\(^\text{19}\)

The sustainability of these business models relies on the cost of producing and distributing open access journals remaining low. Proprietary software solutions for the publication process are often costly and may place limits on publishers in terms of number of articles published, number of issues, customizations, and technical features. This, in turn, can make article processing charges and institutional memberships more expensive for those who support these publishers and may not completely fulfill the needs of the journal. This has led many open access journal publishers to turn to OSS as a low-cost, customizable
solution. “Both OSS and open access to research represent innovative responses to the particular restrictions placed on the sharing and exchange of software code and research publications, respectively, imposed by current intellectual property economics.”

Thus, the marriage between these two open initiatives is only natural.

Over the last two decades, many libraries have become producers rather than solely consumers of scholarly information. “A growing number of libraries have become digital publishers, primarily offering free/open access journals and institutional repositories.”

Given the sophistication of contemporary open source e-journal production systems, libraries can manage their publications with less staff and fewer resources than traditional publishers. The in-house production of these journals on open-source e-journal platforms also allows the preservation of the open access content to remain in the hands of the library itself, rather than relying on external vendors who may decide not to migrate old content or keep accessibility up to modern technological standards. The other major benefit is that this software is extensible and interoperable, allowing libraries to customize to their needs and build them to work well with existing systems infrastructure.

**PLoS and Ambra**

A notable example of a successful open access publisher operating on an open-source platform is PLoS (Public Library of Science) which runs on Ambra. PLoS was founded in 2001 by Harold Varmus, Patrick Brown, and Michael Eisen “as an alternative to the growing constraints of traditional scientific publishing.” It relies on the article-processing charge model and after nearly a decade in operation became self-sufficient based on this model. “Since the launch of PLoS’ first Open Access journal PLOS Biology in 2003, the organization has introduced six additional highly regarded peer-reviewed journals, redefined publishing with the multidisciplinary, rigor-focused journal PLOS ONE, […] and published 165,000+ articles from authors in more than 190 countries.”

Ambra was developed as an open-source project to support the publication process for PLoS journals. “It provides features for post-publication annotation and discussion that allows for a ‘living’ document around which further scientific discoveries can be made. It is a high-volume, efficient and economical system for the publication of research articles in all areas of science. It leverages best practices in developing enterprise platforms, flexible in that it can store any type of data/content, scalable to support a large volume of articles and built to minimize downtime.” The platform is in active development and a generic version, which can be customized for non-PLoS journals, is available for download on GitHub.
Open Journal Systems

While PLoS is a successful open access journal publishing model serving the science community, many university libraries have taken on a similar mission to serve other disciplines outside of STEM. These institutions have small university presses which support the peer review, copy editing, and distribution of their journals. Many of these have turned to an open-source journal management and publishing software called Open Journal Systems (OJS). “OJS is OSS made freely available to journals worldwide for the purpose of making open access publishing a viable option for more journals, as open access can increase a journal’s readership as well as its contribution to the public good on a global scale.”

The features that make OJS an attractive option for open access journal publishers include local installation and control, editor configured requirements, sections, review process, online submission and management, a subscription module with delayed open access options, comprehensive indexing, reading tools for content based on discipline, email notification and commenting ability for readers, and complete context-sensitive online support. By using existing university infrastructure and OSS, an open-access journal can be published for little more than the time volunteered by the editorial board [...]; implementing a system like OJS can drastically reduce the hours required to manage a scholarly journal.” As of the writing of this chapter (February 2017) there are more than four thousand active instances of OJS running more than ten thousand journal titles which have produced over 400,000 articles. The ease of use and commitment to open access has made OJS a reliable alternative option for library publishing units that are looking to move away from proprietary products for their open access publishing needs.

Vega

PLoS and OJS are great OSS options for traditional scholarly publications, but the landscape of scholarly communication is changing. As technology advances, more scholars are producing work in various multimedia formats which cannot be accommodated by these platforms. These projects can consist of GIS data, digital artifacts, data mining, documentaries, animation, code, software, data visualizations, or a combination of these and more. Creators can write about the process of creating these works but are unable to publish them in peer-reviewed
journals in their intended form due to the technical constraints of the current publishing platforms, but that does not mean the works are less valuable to the public. While there are multimedia authoring platforms available to scholars, such as Scalar, they do not offer editorial workflows for publishing peer-reviewed multimedia scholarship. This is where Vega comes in to solve this dilemma. Vega is an Andrew W. Mellon Foundation-funded open-source project currently under development (anticipated release 2018) that aims “to provide publishers and authors the means to produce high quality academic publications, that makes use of rich media, source data, video and interactive experiences.” It will offer similar editorial workflow features to OJS, including submission tracking, automated email communication, user-info databases, and front-end reader interfaces.

While the platform will not “support publishing content-management systems such as Omeka or Scalar within [its] publishing system […], [a]uthors can still use a platform like Scalar, Omeka, or WordPress to build their webtext and then harvest or convert it for static HTML preservation (a process known as scraping) and publication through a Vega-run venue. Vega’s source code is licensed under a generous MIT open-source license, and will be distributed in open repositories once the code base is ready for release.”

Open repositories

In addition to the publication of open access journals, libraries and open access advocates are also operating open repositories. The Directory of Open Access Repositories (OpenDOAR), which maintains an authoritative list of institutional and subject-based repositories, currently lists 3,315 open repositories around the world. Of these, 85.4 percent (2,831) are institutional, 9 percent (297) are disciplinary, 3.2 percent (105) are aggregating, and 2.5 percent (82) are governmental. The most common types of content stored and shared in these repositories are journal articles, electronic theses and dissertations, books, chapters and sections, conference and workshop papers, unpublished reports and working papers, multimedia materials, and datasets, with many repositories containing multiple types of content. These repositories are the backbone of green open access initiatives in which authors make their work available by self-archiving their scholarly output. “Since they are formal institutional functions, institutional repositories are permanent and stable. There is often a commitment to use digital preservation techniques to ensure the continued availability and usefulness of the digital materials that they contain.” They may include electronic document publishing functions, such as e-journal management or conference paper management systems. They typically utilize free OSS, such as DSpace, EPrints, or Fedora, but may be externally hosted by vendors for a fee. OpenDOAR currently
reports that the majority of the open repositories within its directory run on OSS platforms. The breakdown is 44.4 percent (1,473) using DSpace, 13.6 percent (451) using EPrints, 8.1 percent (269) using an unknown platform, 4.7 percent (157) using Digital Commons, a proprietary platform, and the remaining 29.1 percent (965) using other unspecified software.\textsuperscript{41} In order to support the mission of open access and preservation for continued access, libraries have formed communities around open-source repository software. These communities allow institutions to build solutions that address these needs that no one institution would be able to build or sustain on its own.

**Fedora**

Flexible and Extensible Digital Object Repository Architecture (Fedora) is a modern, open-source repository platform with native linked data support designed to store, manage, and preserve digital assets.\textsuperscript{42} The current version of the software has many modern features. In addition to being a linked data platform server and speaking RDF natively, Fedora can store, preserve, and provide access to any file type and any file size. It is standards-based and highly extensible. This extensibility allows institutions and communities to select and customize the underlying architecture to fit their needs. The software also comes with many key preservation services, including versioning, auditing, and file fixity checking.\textsuperscript{43}

The Fedora project began in the early 1990s as the result of research at Cornell University. By the late 1990s, the research had evolved and resulted in a Common Object Request Broker Architecture and software implemented in Java. This early version of the software included support for a variety of content types, extensible functionality to support new data types as they emerged, the ability to combine multiple binary files into one object, and the ability to associate rights management schemes with these digital objects and its subparts.\textsuperscript{44} Principally, the guiding vision of the project was “that interoperability and extensibility is best achieved by architecting a clean and modular separation of data, interfaces, and mechanisms.”\textsuperscript{45}

In 1999, the University of Virginia was in search of a new digital library platform to support large-scale collections in a myriad of content types. This need stemmed from nearly a decade of building digital collections in a variety of content types and hosting them on independent websites and software platforms with little to no interoperability. The university hoped to find a platform that supported a “wide variety of features, including scalability to handle hundreds of millions of digital resources, flexibility to handle the ever-expanding list of digital media formats, and extensibility to facilitate the building of customizable tools and services that can interoperate with the repository.”\textsuperscript{46} By finding a platform
that provided extensible services in which customizable tools could be built, the institution believed it could build any new features it needed into the future. The university began by reviewing the available proprietary solutions that existed.

While the shortcomings of vendor-provided solutions were numerous, few, if any, solutions “attended to the critical issue of interoperability, failing to provide an open interface to allow sharing services and content with other systems.” In the summer of 1999, the group charged with selecting the platform stumbled across a paper about Fedora. After reaching out to the group at Cornell, the University of Virginia team installed the current version of the software, which until that point had largely been used for research, not in libraries. After successfully experimenting with the framework and some of its own digital collections, the University of Virginia reshaped the existing Fedora framework and developed a new prototype. This prototype provided strong evidence that the Fedora architecture could be “the foundation for a practical, scalable digital library system.” This prototype also resulted in providing the basis for funding from the Andrew W. Mellon Foundation to Cornell and Virginia to jointly develop the platform and make it available as OSS to museums, libraries, and archives.

In 2007, a $4.9 million grant from the Gordon and Betty Moore Foundation established Fedora Commons, a non-profit organization to further develop and promote the Fedora project as an OSS package. By this time, more than one hundred institutions were already using the Fedora platform to manage and build digital collections. By establishing Fedora Commons as an independent, non-profit organization, the hope was to expand the Fedora community internationally and attract more scientific, scholarly, and educational users to openly collaborate for the continued development of the project.

Fedora is “designed, developed, and supported by a global community of stakeholders.” While open source, Fedora is also funded by institutional memberships that support training, outreach, and two full-time staff members: a technical lead and a product manager. The product manager sets the vision of the project and serves as a liaison to a variety of groups and stakeholders. The technical lead provides technical leadership and coordinates community efforts around the design, development, and infrastructure of the software.

The Fedora project relies on its user base for stewardship and development. The platform attributes an Apache License, Version 2.0, which grants recipients extensive rights to modify and redistribute the software. The Fedora community welcomes individuals from its user base to participate in the project as a contributor. Like other open-source communities, contributors who continuously demonstrate engagement with the project through quality participation in meetings, listserv discussions, and documentation and code updates can be nominated to become committers. Committers in the Fedora community have special rights, including
the ability to write directly to the code base, nominate new committers, and vote on code modification and issues related to releases. These committers are also responsible for monitoring and responding to threads on mailing lists, attending meetings, reviewing and committing code, and guiding and mentoring new committers. At the time of this writing, Fedora currently has twelve active committers from institutions around North America. It also acknowledges dozens of active contributors from around the globe.\textsuperscript{54}

Historically, it was fashionable to build your own front-end architecture around Fedora to meet your own defined needs. As the preservation and digital assets management needs of libraries and archives continue to grow and become more complex, institutions have recognized that by adopting and participating in community-built projects built around Fedora that they can more easily implement it, develop new features, and create standard workflows and content models.

**Samvera**

Samvera (formerly the Hydra Project) is a community-developed repository software solution built on top of Fedora Commons and other open-source libraries to provide a repository solution for users. The current software stack utilizes several technologies: Fedora Commons as the repository layer for preserving and managing digital objects; Apache Solr for indexing metadata about these objects and providing fast access to users; Blacklight, a Ruby on Rails gem for providing faceted searching, browsing, and custom web views of digital objects; and HydraHead, a Ruby on Rails gem that provides CRUD (Create, Read, Update, Delete) operations against the Fedora Commons Layer.\textsuperscript{55} The Samvera Project began as an open-source project in 2008 as a joint development project between Stanford University, the University of Virginia, and the University of Hull, in close collaboration with the development team behind Fedora.\textsuperscript{56} It currently operates under an Apache License, version 2.0, granting developers extensive rights to modify and redistribute the software.\textsuperscript{57} The Samvera philosophy aligns with that of many open-source projects. Specifically, the group recognizes the importance of building a framework that is extensible, investing time and resources into the community, encouraging new adopters and developers, ensuring the framework and decisions that guide it are transparent, and relying on the community to contribute back not only code, but also documentation, designs, and communication.\textsuperscript{58}

The Samvera framework is being used by institutions to support a wide array of use cases and applications. In 2013, Gilbert and Mobley published an article describing why they selected Samvera and Fedora Commons as an open-source alternative to ContentDM for hosting content for the consortial Lowcountry Digital Library. The move to leveraging an open-source platform was initiated by
the way ContentDM defined unique digital objects and its associated cost model. By leveraging Samvera and Fedora Commons, the authors were able to create a robust and sustainable platform for growing their digital collections. Samvera’s flexibility and ability to support a wide variety of content types have made it a popular platform on which to build institutional repositories. The University of Hull, University of Virginia, Penn State University, and Stanford University all use the platform to host their institutional repositories and numerous content types, including electronic theses and dissertations, articles, and data sets. Samvera is also used by many libraries for traditional digital library needs, such as hosting image collections. In 2013, Northwestern University went live with a new Samvera-based repository that hosts more than 115,000 unique images. This repository has many useful features, including allowing users to build their own collections via drag and drop, to upload personal images and metadata, and to zoom and rotate their images.

Avalon Media System is an open-source solution built on Samvera for creating and providing online access to digital audio and video collections. Unlike other streaming media solutions, Avalon was developed specifically to fit the unique needs and use cases of libraries and archives. The system includes many unique features, including: the ability to ingest, transcode, describe, and play media files on various devices and screen sizes; the support of faceted discovery and captioning in VTT and SRT files; and the ability to integrate with other services, including authentication services, learning management solutions, and other streaming services. Currently, the platform is being developed in a collaboration between Indiana University and Northwestern University. The Avalon team has also received commitments from seven institutions to test and provide feedback on the platform.

The popularity of Samvera and the need to structure it to provide common features needed across institutions led to the creation of Suia. Suia is a framework built on top of Samvera that adds the ability to upload files, create and assign metadata, generate derivatives, and provide a standard way to create and manage users and roles within the system. Originally developed at Penn State University, the Suia project was derived in order to be able to share a generalization of Penn State’s Samvera instance, Scholarsphere, to other libraries and archives. Since it was contributed to the Samvera community in 2012, many new features have been created and added to the Suia project via developers from more than thirty institutions. The popularity of Suia continues to grow and is currently implemented at these and other institutions to address many diverse use cases, including digital assets management and as a framework to create institutional repositories.

While the Samvera community continues to grow, both in terms of developer size and total number of implementations, a common criticism has been that the
framework can be difficult to implement and support at institutions that do not have the necessary developer time or resources. This notion has led to the development of Hyku: a polished, turn-key, feature-complete repository service built on top of the Samvera framework. Previously referred to as “the Hydra-in-a-Box” project and “Lerna,” Hyku became the official name of this Institute of Museum and Library Services-funded project in December 2016. Hyku is important because it is designed to be agnostic to institution size and developer resources and aims to bring the codebase and community of Samvera to all libraries, museums, and archives. To do this, Hyku is designed to be easy to install and maintain while also meeting the needs of the library, museum, and archives community. The first pilot projects using Hyku are planned for 2017.

**Islandora**

Islandora is an OSS framework built on Fedora, Drupal, and Apache Solr. Originally developed by the University of Prince Edward Island’s Robertson Library, Islandora is designed to help libraries, museums, and archives to collaboratively manage digital assets and make them discoverable to users. As an open source project, Islandora is released under GNU General Public License. As a digital assets management system, Islandora has many compelling features. Because it is built on Fedora, Islandora can support any binary file type. In order to be able to support derivative generation and provide a common user experience across content models, Islandora uses modules called solution packs. While it currently relies on metadata serialized as XML, the platform is schema agnostic in terms of describing the objects in which it contains. By using Solr, Islandora can index this metadata and create facets for searching.

The Islandora project was first envisioned in 2006 at the University of Prince Edward Island in Canada. The vision was to create an application that leveraged the digital assets management features of Fedora and the power of the Drupal community. In doing this, Drupal would provide an easy mechanism to update the look and feel of the repository and its modules could provide needed features without requiring extensive custom development. The Islandora team also recognized that creating the project as an open source release would provide an opportunity to create a robust framework and develop strong documentation.

The Islandora Foundation was created in 2013 as a “federally incorporated, community-driven soliciting non-profit.” The formation of the foundation was pivotal in growing the community around the platform. The foundation holds several key roles, including overseeing the maintenance of the codebase and its documentation, creating and providing access to educational material including webcasts and training videos, organizing training events, and marketing the
product at conferences and workshops. Similarly to Fedora, the foundation currently employs a project and community manager and a technical lead.

Islandora is currently installed at more than 150 institutions to solve a variety of use cases. The Project for Automated Library Systems (PALS), which provides library technology support for a consortium of nearly sixty libraries, has used Islandora to address the needs of several of its members. As of 2014, Kent writes that the platform was being used to host photographs, newspapers, and book-like objects at PALS institutions. The University of Prince Edward Island launched a new institutional repository service in 2012 to support a new university policy that encouraged scholars to deposit their research and ensure that the research was both harvestable and linked in the repository. The features included creating and linking researchers and departments to scholarly works, providing metrics related to scholarly objects, enabling researchers to deposit research, and integrating OAI-PMH into the software stack. Since 2014, the University of Toronto Scarborough has been using Islandora to create and steward oral histories. Its oral histories solution pack manages materials and the administrative processes needed to develop oral histories and provides functionality to describe, transcribe, and translate audio and video files.

In order to keep the project moving forward, Islandora developers host a weekly committers’ call to review new issues and address open tickets. To officially contribute code to the project, the Islandora Foundation requires developers to sign Contributor License Agreements (CLAs) in order to protect the foundation as well as users of the software. The community also recognizes the importance of involving non-developers in sustaining the project. In its biannual release cycles, the Islandora contributor teams have several roles that require no programming acumen. These roles include: auditors, who check to ensure that the README and LICENSE files meet the standards set forth by the community; documenters, who ensure that the current documentation is accurate and reflects the current release; and testers, who test the release candidate for bugs and report findings to the committers.

DSpace

Like Fedora, DSpace is another popular digital assets management system used by libraries and archives. DSpace is a turn-key, open-source system designed to collect, archive, disseminate, and preserve scholarly materials. The software is created to be binary agnostic and support preservation and open access to text, images, video, and data sets. Although it can only manage a specific subset of file types out-of-the-box, like Fedora, DSpace can be extended to manage any binary file type. With more than one thousand organizations currently using DSpace,
it is the most popular repository service in the world. In order to maintain this popularity, DSpace is focused on creating solutions surrounding institutional repositories and open access, continuing to be turn-key, and relying on extensibility and integration with complementary services.\textsuperscript{74}

Originally developed as a collaborative effort between MIT Libraries and HP labs, the DSpace project now relies on growing its active developer community and harnessing their skills in order to continuously expand and improve its software.\textsuperscript{75} As an open-source project, the DSpace codebase applies the BSD open-source license, which allows any organization or institution to use, modify, or integrate the code into another application.\textsuperscript{76} According to its website, DSpace currently has more than one hundred committers around the globe who actively contribute code and bug fixes,\textsuperscript{77} and more than one thousand organizations currently use DSpace for their repository services.\textsuperscript{78}

In March 2000, the Massachusetts Institute of Technology Libraries signed a co-development contract with Hewlett-Packard in order to build the infrastructure to store “digitally born, intellectual output of the MIT community and make it accessible over the long-term to the broadest possible readership.”\textsuperscript{79} The result of this activity would lead to the development of what would be known as DSpace. In order to build the platform, MIT received two substantial awards: a $1.8 million grant from Hewlett-Packard and a $215,000 grant from the Andrew W. Mellon Foundation.\textsuperscript{80} In 2002, MIT launched its digital repository framework under the name of DSpace. The software was created to “capture, distribute, and preserve the intellectual output of MIT.”\textsuperscript{81}

While the initial development of the software was the result of a collaboration between MIT and HP Labs, the project was intended to be open source from its onset. At its first user group meeting in March 2004, the future governance of the project was formulated. After a presentation by senior members of the Apache Foundation on the governance and attributes of their highly successful OSS community, the DSpace Federation was formed, which set forth the governance of the project and established how the DSpace Committer Group would operate.\textsuperscript{82} Today, the DSpace project relies on growing its active developer community and harnessing their skills in order to continuously expand and improve its software.\textsuperscript{83} Like other open-source projects, DSpace is built by project contributors. At the time of this writing, DSpace currently has more than one hundred committers around the globe who actively contribute code and bug fixes and acknowledges two hundred ten individual DSpace community members as making at least one contribution in the form of reporting or fixing a bug, providing a new feature, or helping write documentation. The DSpace community also acknowledges twenty-four committers who have autonomous control over the code and act as the primary support team for the platform.\textsuperscript{84}
Another prominent digital asset management system is EPrints. Unlike DSpace and Fedora, the EPrints community is more active in Europe. EPrints is an OSS package for creating institutional repositories that comply with the Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). Since its inception, the EPrints community has recognized that institutional repositories are the key to providing open access to scholarly research. Supporting open access continues to be the driving force behind the development of the platform. The software provides a mechanism for researchers and institutions to deposit, preserve, and disseminate scholarly publications. In addition to being OAI-compliant, the software is also designed to be easily extensible in order to allow users of the software to develop functionality that meets their needs.

Released in 2000, EPrints was designed specifically to allow institutions to support Open Access agendas. A participant at the second Open Archive Initiative meeting said, “Open Archiving will not get off the ground until the day I can go to a website, download open-archiving software, then say MAKE ARCHIVE, and an interoperable, OAI-compliant archive is up and running, ready to be filled.” This specific statement served as inspiration and motivation for developers from the University of Southampton to create EPrints. In just a few months, the development team created and released a public beta of the EPrints software to directly address the comments made at the Open Archive Initiative meeting. From its initial release, the software made it easy for institutions to install and included out-of-the-box support of OAI-PMH.

After its public beta period ended, EPrints was released to the public as an OSS package. The software is currently released as OSS under the GNU General Public License allowing end users to freely run the software, study its source code, and modify it as needed. The source code for EPrints is available on GitHub, and packages for several Unix operating systems are maintained and made available from its website. The EPrints software stack utilizes many traditional open source technologies, making it easy for newcomers to the community to participate. These technologies include Apache, MySQL, and PERL.

Starting with version 3.3 of the software, the EPrints Bazaar was released. Inspired by Apple’s “App Store,” the Bazaar is meant to be a “one-stop shop” for repository administrators to discover and easily install plugins that extend EPrints’ out-of-the-box functionality. It also serves as a mechanism for developers to easily contribute new features to the community. The EPrints community also strives to find ways to include non-programmers in the community. To do this, it maintains a wiki for documentation, working groups, and training materials, and welcomes participants to contribute new articles and improve existing documentation.
LOCKSS

While not a digital assets management system like Fedora, DSpace, or EPrints, LOCKSS, or Lots of Copies Keep Stuff Safe, is an open-source digital preservation software used by libraries and archives to ensure bit-level preservation in a distributed network. The LOCKSS community describes LOCKSS as a network platform that provides libraries and publishers with a low-cost, open-source digital preservation solution to preserve and provide access to digital content.\(^{95}\) The technology behind LOCKSS relies on two critical components: a LOCKSS network and a LOCKSS box. A LOCKSS network is made up of libraries that acquire copies of important “stuff” deemed important by other libraries in the network. This important content is then ingested by LOCKSS boxes in the same network. These LOCKSS boxes are computers with special software (the LOCKSS daemon) that ingest prescribed content using a web crawler. The LOCKSS daemon is the only application that runs on the hardware and is responsible for “ingest, preservation, dissemination and administration” of digital objects.\(^{96}\) The key feature of these boxes is that they continuously compare this ingested content with other LOCKSS boxes within the same LOCKSS network. If errors are found between the same object on multiple boxes, the LOCKSS box dynamically compares the objects across the network and repairs the affected copy. The software on the LOCKSS box also provides an easy method for libraries to be able to manage and select new content to be ingested into the network.\(^{97}\)

In 2000, Reich and Rosenthal first presented their prototype of LOCKSS as a cheap and easy way to preserve access to scientific journals published on the web. The initial concept behind LOCKSS was a peer-to-peer inter-cache protocol that continually ran between library web caches in order to identify missing or damaged URLs. When a broken URL was identified, an HTTP request was sent to the original publisher or to another library cache to retrieve a good copy of the item in question.\(^{98}\) In the following years, LOCKSS continued to grow in terms of user base and community involvement. By 2004, more than thirty libraries had volunteered to test LOCKSS as a software to ensure that important cultural assets were available to future generations. While LOCKSS was originally intended to protect content from electronic journals, it did not take long for libraries to discover its usefulness for preserving image collections and websites, as well as other materials from archival and manuscript collections.

As the popularity of the platform continued to rise and the need for technical expertise began to increase, the LOCKSS community recognized the importance of growing an open-source community to continue development of the software. While the project had always had a small, dedicated team of developers at Stanford University, the LOCKSS community realized that developing the software at
a single institution could lead to a single point of failure and decided to recruit technical contributors from its already established user base from the current LOCKSS membership.99

The long-term success of LOCKSS can be attributed to its being open source. OSS promotes both interoperability and open standards. The transparency of the software’s code base allows developers to understand the code and its behavior, which, in turn, allows newcomers to contribute to the project. These contributions have led to long-term sustainability and the continuous growth of the project. Both the LOCKSS software and the operating system it runs on are open source, and the community intentionally targets mid-range hardware to reduce costs for institutions. The LOCKSS community maintains an ISO image of CentOS with all required packages to get a LOCKSS box up and running along with step-by-step on its website.100

Conclusion

It is clear that the values shared by libraries and OSS communities make them a well-suited partnership. The use of open source technologies allows libraries to build customizable, interoperable, and extensible solutions to fulfill the goals of open access and preservation. Open access movements in libraries are gaining momentum and do not appear to be diminishing anytime soon. Systems like OJS and PLoS’s Ambra have proven to be successful and reliable open journal platforms allowing libraries to find a sustainable, low-cost solution for libraries to produce their own journals, while Vega shows a lot of potential for the open publishing of rich media content not currently possible with traditional journal publishing platforms. With the strong communities forming around open repository software like Fedora, Samvera, Islandora, DSpace, and EPrints, librarians are able to ease their minds about the longevity and access to the materials they manage due to the intentional focus on interoperability and open standards. These communities welcome contributions not only from developers, but also documenters, auditors, testers, and user feedback. The support provided by combining our efforts to create products which actually meet our needs is invaluable and allows us to go further together by learning from one another’s mistakes and building on one another’s successes. The software that has resulted from these efforts helps libraries and institutions of all sizes because the smaller groups with fewer resources benefit from the contributions of the larger ones. This strengthens the ability of all libraries to support the values of access, preservation, and contributing to the greater good of society through these open movements.
Notes

13. Ibid.
15. Ibid.
16. Ibid.
17. Ibid.
18. Ibid.
19. Ibid.
22. Ibid., 372.
24. Ibid.
25. Ibid.
35. Ibid.
37. Ibid.
40. Ibid.

46. Ibid., 407.
47. Ibid., 408.
48. Ibid.


68. Ibid.


73. Ibid.


75. Ibid.


81. Ibid.


87. Ibid.

88. Ibid.


90. Ibid.

91. Ibid.


Bibliography


Ferguson, Cris. “The Key to LOCKSS: An Interview with Victoria Reich, Director, LOCKSS, Stanford University.” Against the Grain 18 (April 2016): 50–52.


