

FOLIO: The Future of Library is Open

Satbir Chauhan^{1*}, K. Kandhasamy² and N. Sakthivel³

¹Executive - Library, Ashoka University, Sonapat – 131029, Haryana, India;
satbir.chauhan@outlook.com

²Doctoral Researcher, Department of Library and Information Science, Faculty of Art, University of Delhi,
New Delhi – 110007, Delhi, India; Kandhasamy5050@gmail.com

³Library Assistant, Jio Institute, Navi Mumbai – 410206, Maharashtra, India;
n.sakthivel@jioinstitute.edu.in

Abstract

Ranganathan's 5th law of library science says that libraries never become stagnant. While financial constraints are a common challenge for libraries, user expectations are continuously growing. To meet and exceed user expectations, libraries must embrace new technologies and software solutions. Open-source options have proven to be particularly beneficial for nonprofit organizations like libraries. Looking to the future, as libraries continue their digital transformation, many library automation software, especially traditional library management systems (ILS), are becoming outdated. In response to the diverse requirements of libraries, the architecture of Library Services Platforms (LSP) offers enhanced flexibility and functionality within a unified platform. Among the various LSPs that have emerged over the past seven years, FOLIO has demonstrated rapid development and progress. Upon exploration, it becomes evident that FOLIO's developers have prioritized a micro-services architecture, emphasizing apps over modules. This approach provides a comprehensive multitasking app environment, a user-friendly Dashboard for quick access, efficient data management, and much more. The user-friendly nature of FOLIO sets it apart from traditional Integrated Library Systems (ILS), which may lack such ease of use. Despite its abundant features, FOLIO is still relatively under-recognized in India. This paper aims to discuss the distinctive features of FOLIO, its potential as a comprehensive solution for libraries, and whether it can effectively replace traditional ILS systems.

Keywords: Future of Libraries Is Open (FOLIO), Future of Libraries is Open (FOLIO), Integrated Library System (ILS), Library Services Platform (LSP), Open-Source Software (OSS)

1. Introduction

Running a Library is all about management; effectively managing the library and satisfying users are primary concerns in the 21st century. Darwin's renowned theory, *survival of the fittest*, applies almost everywhere, and libraries are no exception. For survival, a library needs to synchronize with technology and upgrade itself because patron demands are highly dynamic, and of course, to fulfil the 5th law of Library Science. Library software plays a significant role in catering to the on-the-go need of patrons. This kind of software and supporting infrastructure demands a great price and maintenance cost, but libraries also have a history of lack of funds. The one-stop solution for this everlasting problem of libraries

is Open-Source Software (OSS). Richard Stallman laid the first milestone for the open-source movement in 1985. Later, Open-Source Initiatives (OSI) was founded in 1998 by Bruce Perens and Eric S. Raymond; the OSI is a non-profitable organization. Over time (Figure 1), the OSS has become key for libraries to open the gateway of community support.

The beauty of open-source development projects is 'community support' which continuously improves it. The prime focus of such software category is cost-effectiveness, providing unrestricted access, reliable solutions, the scope for improvement, flexibility regarding platforms and distributions, and of course, end to proprietary vendor lock.

*Author for correspondence

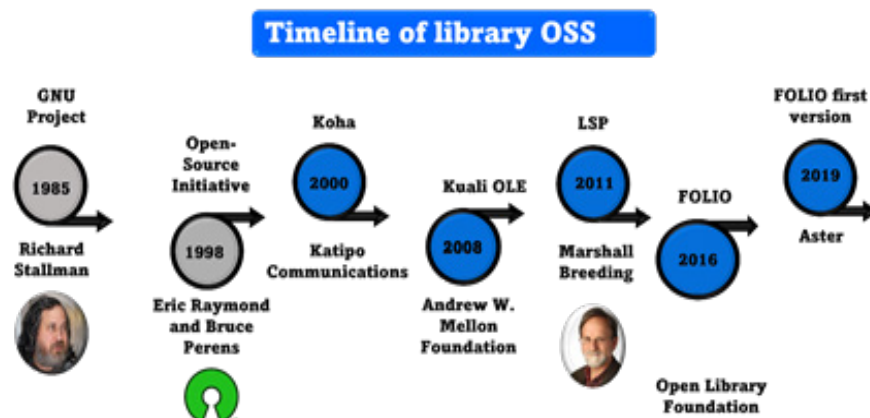


Figure 1. Timeline of OSS (in libraries) development.

The advent of computers and the Internet has dramatically changed information intake and reshaped libraries. Major library management and housekeeping tasks are done through a specific library software or system called Integrated Library System (ILS). Over the last three decades, many ILS software has been used, such as VTLS, LIBSYS, SLIM, AUTOLIB, Koha, SOUL, NEWGENLIB, E-Granthalaya, etc. (Mukhopadhyay, 2002). Due to the lack of updates, some have become obsolete. These legacy software packages have primarily managed print resources. The digital revamping of libraries brings another challenge to managing non-physical resources. “Empirical evidence indicates that academic institutions in the United States have increased library expenditures for electronic resources and decreased expenditures for physical resources” (Kortick, 2021); such studies indicate that the literary world is moving faster towards E-Resources. This is when a library needs an E-Resources Management system (ERM), dedicated software to manage E-Resources or digital resources of libraries. Since the conventional ILS lacks in terms of architecture and technology, it is not a good fit for managing e-resources. Additionally, there are ERM software packages such as CORAL, E-MATRIX, Alma ERM, etc., are available and are being used. However, they are all designed exclusively to meet e-resource needs. These earlier-mentioned technologies function well. Yet, the library must use and maintain them separately, pay for expenditures separately, or may have to involve different vendors for each and most of the ILS are built using the old technology of monolithic architecture in which all functionality of the software exists in a single code-base. To get through these challenges, libraries need

a single platform to manage all formats of resources, also to address the limitations of traditional ILS and adapt to the new age of cloud computing; for such need, the library automation sector developed a single solution called a Library Services Platform (LSP) which is generally powered with micro-service architecture.

2. Objectives

The major objectives of this study may be stated as:

- To make OSS and LSP widely known to libraries
- To give glimpse of FOLIO LSP and its characteristics features
- To share the hands-on experience of FOLIO with fellow professionals
- To understand how libraries can be guided by this new software architecture to their future

3. Library Services Platform (LSP)

The term LSP was first introduced by renowned library technologist Marshal Breeding in 2011. “Library services platform” is a new software genre for libraries. LSP supports all formats of collection a library could have and has immense support for e-resources. It is a fully loaded platform for the various needs of the library, powered by a loosely coupled app architecture that gives it modern library system capabilities. LSP gives a choice for applications from third-party providers that might be different reporting options, archival solutions preservation, institutional repositories, etc. The unique characteristics of LSP are - Not limited to physical resources; Replacement of Multiple Incumbent

Products; Extensive Metadata Management; Exposed APIs for Interoperability and Extensibility; and Polyglot programming i.e. coding in multiple languages (Breeding, 2015; Grant, 2012; Pradhan, 2019).

4. Monolithic vs Micro-services

Traditionally, ILS is developed using a monolithic architecture, where the programme is created as a single entity with a single codebase that houses all the capabilities and components. A contemporary strategy known as “micro-service architecture” separates the application into smaller, independent services that communicate with one another using simple protocols. The way application is organized and deployed is the primary distinction between monolithic and micro-service architecture. The LSP follows the micro-services architecture approach. Graphic illustration of the basic difference

between the architecture of both is given in Figures 2 and 3.

5. FOLIO

FOLIO LSP has its roots in the Open Library Environment (OLE) project, which was initiated in 2008. In 2016, “The Open Library Foundation” was established to increase the sustainability of libraries. The foundation initiated an open-source project for and by libraries. Due to that, in 2016, the foundation (in association with Index Data, EBSCO, and Open Library Environment (OLE)) collaboratively developed the Future Of Libraries Is Open (FOLIO), an open-source micro-services framework-based LSP to create a new way of catering library services with the open-source software. With FOLIO, libraries can access a modern and flexible system that is constantly evolving to meet their needs. It ensures maximum community participation, adoption, and contribution; It is modern Cloud-based, multi-tenant, services-oriented, Powerful to manage the full spectrum library workflows, and flexible to configure in many ways, anyone can build on or extend. (Index Data, 2023). FOLIO helps to manage all types of libraries and libraries of various sizes. Currently, the FOLIO platform and its ERM are growing progressively, and many libraries are starting to implement it. Globally, there are more than 100 institutes that implement the FOLIO LSP. For example, Stanford University, Texas A&M University, Duke University, Lehigh University, University of Chicago, University of Colorado, and Jio Institute in India. Library of Congress,

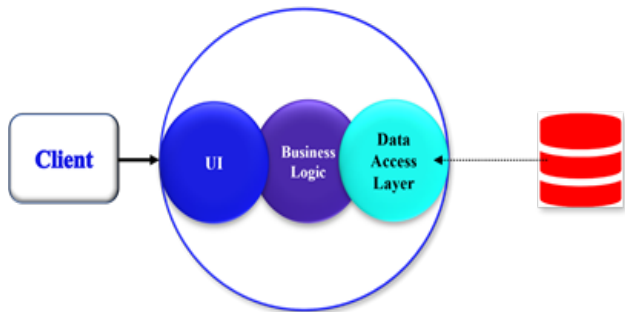
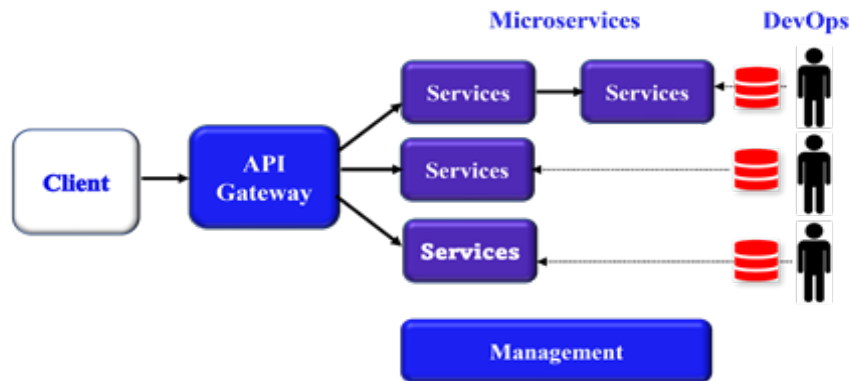


Figure 2. Monolithic architecture.



Source: <https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/microservices>

Figure 3. Micro-services (Apps) architecture.

National Libraries of Australia are planning to implement (Enis, 2022). The first version of FOLIO was Aster released in January 2019; the upcoming version of FOLIO is Poppy and will be released in 2023.

5.1 FOLIO Architecture

Folio features a modular architecture that enables many parts to interoperate and communicate via a shared framework. The core of FOLIO is the Okapi gateway (Figure 4), which makes it possible to integrate with loosely coupled apps (micro-services) and functional cohesion that offer a variety of features, including analytics, circulation, acquisitions, and inventories (Murray, 2023).

5.2 Features of FOLIO

The essential features of FOLIO LSP are discussed here under six major heads.

5.2.1 User Interface/ User Experience (UI/UX)

One of the pivotal elements of FOLIO is its User Interface (UI) design. FOLIO has set some guidelines to promote an aligned vision for developers, consistency, and ease of use across all apps. FOLIO does have its design system called MOTIF. The MOTIF outlines FOLIO apps’ visual style, patterns, components, and interactions. MOTIF is implemented using Stripes, a front-end framework explicitly developed for FOLIO. The Stripes make it possible that one can create FOLIO apps with very minimal coding. It provides a set of reusable UI components and

modules. Stripes handle typical duties like data gathering, routing, authentication, and localization. The developers and designers can take advantage of the following resources available on the FOLIO UX website to create FOLIO apps with MOTIF:

- **UX prototypes:** Contains interactive mock-ups which demonstrate the look and feel of FOLIO apps
- **UX guidelines:** Documentation regarding FOLIO design components
- **UX assets:** This includes fonts, colors, and icons and these are downloadable.
- **FOLIO library:** Consisting of all the components used to design apps

5.2.2 API Support

FOLIO’s API (Application Programming Interface), which facilitates interoperability and interaction with other systems and services, is one of its primary characteristics. The FOLIO API is based on REST (REpresentational State Transfer) principles, which means that it communicates with the FOLIO server using common HTTP methods like GET, POST, PUT, and DELETE. To share data, it uses JSON (JavaScript Object Notation).

The Okapi, an API-Gateway is used by FOLIO to manage several modules that include acquisitions, circulation, users, analytics, and other aspects of library management. Okapi divides web service endpoints into two categories: (1) APIs for general module and tenant management; so-called “core”, and (2) endpoints for interacting with business-logic-specific interfaces given by

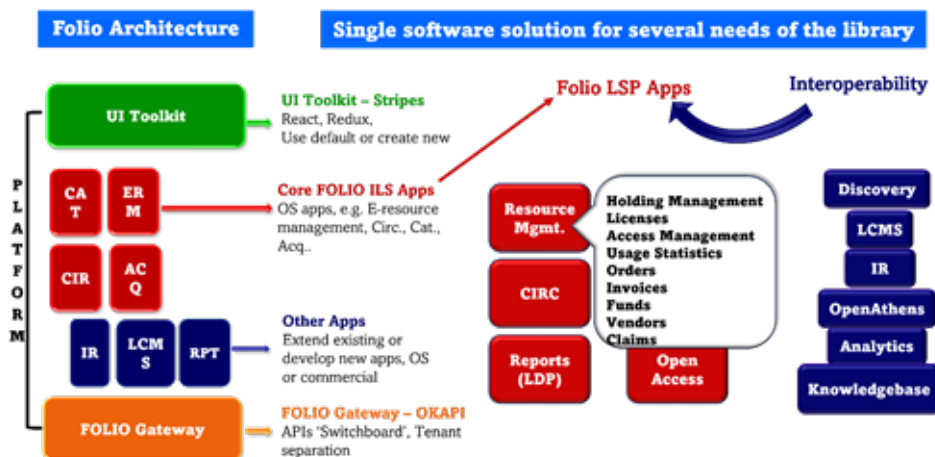


Figure 4. FOLIO architecture.

modules, such as circulation or patron management. For example, the inventory module has endpoints for create, retrieve, update, and delete items and holdings records. The libraries can modify and improve their FOLIO platform to suit their requirements and preferences because of flexible and extendable API. Additionally, users can connect their FOLIO platform with other external systems and services, including discovery layers, digital repositories, authentication providers, payment gateways, and more, via the FOLIO API (Okapi-FOLIO Community, 2023).

5.2.3 FOLIO Community

One of the key benefits of open-source software is its community. Communities encourage collaboration and innovation among programmers who have similar interests or objectives. Open-source software can be developed, tested, documented, and maintained by the community; they constantly provide feedback, support, and suggestions to improve product quality and efficiency. The FOLIO project does have a vast active community. The FOLIO community intends to support the long-term community-driven development of a contemporary technological ecosystem for libraries. In their governance model, they are using the community of three different councils:

- Community Council
- Product Council
- Technical Council

Each council may sponsor Special Interest Groups (SIGs) or other working groups to carry out their charters. SIG is made up of practitioners of libraries who are interested in contributing to particular functional areas, such as the App interaction SIG and the ERM SIG.

5.2.4 Choice Oriented Services

FOLIO platform enables libraries to develop and modify their offerings based on patron demands and preferences. FOLIO's choice-oriented approach, which enables libraries to choose the modules and applications that best suit their workflows and objectives, is one of its primary advantages. It's like FOLIO gives libraries the freedom and control to direct their own future by being flexible, scalable, and adaptive.

5.2.5 FOLIO Codex

FOLIO is a microservices architecture-based platform that communicates with number of apps, and for managing its metadata and different formats the term that is used to describe a collection of metadata is known as **Codex**. With Codex normalization and virtualization layers, FOLIO may include metadata about multiple resources regardless of format, storage location, and encoding. A lightweight (simplified) metadata schema is used by the FOLIO codex to define resources. The majority of currently used metadata models, including MARC, Dublin Core, MODS, and BIBFRAME, can be mapped to this common denominator. The FOLIO codex also spans storage locations. It does not matter whether a resource is managed locally or in a remote system. Codex enables what the FOLIO community call "cataloguing-by-reference" and anticipates being efficient workflow than copy cataloguing. The FOLIO codex comprises numerous modules (and apps) that fall under the same category. The Codex Search app is the most well-known FOLIO codex application. It allows users to search for resources across various apps that may be used to manage resources, such as institutional repositories, e-holdings, inventory, etc. (Bareau, 2019A; Bareau, 2019B).

5.2.6 FOLIO with Cloud

Looking at FOLIO's current hardware resource requirements, it is supposed to be installed on Cloud, which will enhance its services by giving maximum uptime, elasticity, resilience, security, etc., and will make it a more futuristic platform.

5.3 Installation of FOLIO (Pre-Built Vagrant Boxes)

There are three different types of FOLIO installation and deployment. These are pre-built vagrant boxes, Single-server deployment, and Kubernetes deployment. If an individual wants to try FOLIO without complete installation, one can use Pre-built vagrant box installation by using the following simple steps:

Pre-requisites:

- Vagrant – Recent version.
- VirtualBox – Determine the version of VirtualBox to match your version of vagrant.

Steps

1) Create a folder – Open it – Right Click – Open Terminal – Paste these commands in order:

```
vagrant init --minimal folio/release-core
```

2) Create a file named VagrantFile in same folder and add below code:

```
Vagrant.configure("2") do |config|
  config.vm.box = "folio/snapshot"
  config.vm.provider "virtualbox" do |vb|
    vb.memory = 24576
    vb.cpus = 2
  end
end
```

end

```
vagrant up
```

```
vagrant ssh
```

3) In your browser, hit localhost:3000 (initial credentials: diku_admin/admin)

Based on the hands-on experience of this LSP it appears that the microservices architecture or called apps (in FOLIO) is an enhanced version of modules from ILS that is more efficient and easier to use. The User Interface is quite simple and impressive; Dashboard has the option of full customization and management for quick access or widgets, Integrated ERM system reduces dependency on other software. The multitasking app environment increases its ease of access. The courses app allows one to create and manage course reserves. It has open-source project support for reporting called LDP (Library Data Platform), acting as a relational database and can create custom tables to import non-FOLIO data.

6. Findings

The obvious advantages of an open-source LSP over an open-source ILS are -

1. The LSPs like FOLIO offer a variety of software on one platform with choice-oriented services. By using this software, the library won't have to manage/maintain several software or subscriptions;
2. The technology and architecture behind such software make it a futuristic product;
3. So far deployments of FOLIO are less in number across the globe, but history tells us that it happens with each piece of new technology. The library community must start acknowledging such technologies to remain compatible with the changing world; and

4. It is far ahead of the current generation of ILSs, particularly in managing e-resources.

7. Limitations

1. The major limitations of FOLIO in its present form are (as observed) -
2. The involvement of commercial vendors could disturb FOLIO's open-source status in future;
3. Production-level implementation of FOLIO is quite difficult and requires sound technical understanding;
4. Many libraries may not be able to meet the minimum requirements to implement FOLIO; The basic Hardware/IT infrastructure requirements are:
 - High resource requirements (Single server with containers)
 - Min 24 GB of RAM (40 GB RAM is proffered)
 - 350 GB SSD
 - 8-core CPU
5. It does not offer much scope for localization; Also, there is little support for regional languages or local variations.

8. Conclusion

In terms of manpower, financial resources, and machine/material resources, open-source software provides the best solution for libraries facing budget constraints. Libraries typically employ different software for various services. For repositories, options include DSpace and Eprints; for Library Management Systems (LMS), there are choices like Koha and Libsys; and for discovery services, libraries may utilize EDS or VuFind. As libraries expand and offer more services, the number of software products they need may also increase. However, maintaining a large number of software applications can be challenging and requires expertise. This is where Library Service Platforms (LSPs) like FOLIO come into play. FOLIO offers flexibility, features, functionality, and sustainability, shaping the future of libraries to be open. While the current Integrated Library System (ILS) attempts to add multiple modules like Electronic Resource Management (ERM) to address the situation, in a real-time environment, it may struggle to sustain the same infrastructure and architecture as

FOLIO. Presently, FOLIO does not include an Online Public Access Catalog (OPAC), but this issue will be resolved by incorporating the Searchwork open-source OPAC developed by Stanford University. Additionally, the application of artificial intelligence in early library experiences is an ongoing discussion in both FOLIO and other emerging LSPs, indicating its potential to become the primary software for all types of libraries due to the growing trust within the library community.

9. References

- Bareau, V. (2019A). The codex metadata model. The Open Library Foundation: FOLIO project. <https://wiki.folio.org/pages/viewpage.action?pageId=1415393>
- Bareau, V. (2019B). Available at: The codex vision. The Open Library Foundation: FOLIO project. Available at: <https://wiki.folio.org/display/PLATFORM/The+Codex+Vision>
- Breeding, M. (2015). Library services platforms: A maturing genre of products. *American Library Association*, 51(4). <https://journals.ala.org/ltr/issue/download/509/259>
- Enis, M. (2022). Open for growth: Open source platforms on the rise. *Library journal*. Available at: <https://www.libraryjournal.com/story/Industry%20News/Open-for-Growth-Open-Source-Platforms-on-the-Rise>
- Grant, C. (2012). The future of library systems: Library services platforms. *Information Standards Quarterly*, 24(4), 4. <https://doi.org/10.3789/isqv24n4.2012.02>
- Index Data. (2023, March 27). *FOLIO - Open Source Library Management System: Index Data*. <https://www.index-data.com/folio/>
- Kortick, Y. (2021). Trends in physical and electronic resource usage in U.S. Academic libraries. *Ex Libris*. Available at: <https://exlibrisgroup.com/blog/changing-priorities-trends-in-physical-and-electronic-resource-usage-in-us-academic-libraries/>
- Mukhopadhyay, P. (2002). Progress of library management software: An Indian scenario. *Vidyasagar University Journal of Library and Information Science*, 1-15. Available at: <https://core.ac.uk/download/pdf/333967026.pdf>
- Murray, P. (2023). Welcome to the folio wiki. The Open Library Foundation: FOLIO project. Available at: <https://wiki.folio.org/>
- Okapi-FOLIO community (2023). Okapi guide and reference. Available at: <https://github.com/folio-org/okapi/blob/master/doc/guide.md>
- Pradhan, P. (2019). Library Services Platform (LSP): An overview. *INFLIBNET*, 26(1), 12-22. Available at: https://www.researchgate.net/publication/333816851_Library_Services_Platform_LSP_An_Overview
- Sources from FOLIO:
- FOLIO Community (2023). FOLIO: Open Source Library Services Platform. FOLIO: The Future of Libraries Is Open. Available at: <https://www.folio.org/>
- FOLIO Community (2023). All guidelines. FOLIO UX docs. UX prototypes, guidelines and assets. FOLIO UX. Available at: <https://ux.folio.org/docs/all-guidelines/>
- FOLIO Community (2023). FOLIO Documentation. Available at: <https://docs.folio.org/docs/>
- FOLIO Community (2023). FAQ · FOLIO UX docs · UX prototypes, guidelines and assets. FOLIO UX. Available at: <https://ux.folio.org/docs/faq/>
- FOLIO Community (2023). FOLIO Developers: Reference - API documentation. The Open Library Foundation: FOLIO Project. Available at: <https://dev.folio.org/reference/api/>
- FOLIO Community (2023). FOLIO Developers: Developer and system administrator tutorials. Available at: <https://dev.folio.org/tutorials/>