Objectives for the building of an Open Source Next Generation Library Management System

The management of libraries faces challenges arising from

- major changes in the media market and publishing sector,
- the rapidly changing requirements of higher education and research,
- continuous innovation in technology.

Furthermore, library management systems must fulfill the needs of students, teachers and researchers as well as the needs of library staff.

Next generation library systems must take this into account as a long-term goal (originally done for OLE in 2008). The following document is to be understood not only as a continuation of the OLE concept but also bringing this concept to a wider scope and next level of innovation and technology. The paper takes into consideration the existing OLE functional and visionary documents and adds amendments for future sustainability in the context of the new Collaboration between EBSCO, Index Data and the OLE community. It is a project-independent overview and description of the general needs for a next-generation library management system or platform, giving an idea of the scope and depth of required functionality. It is not a detailed catalogue of requirements or functional specifications, which have to be covered separately.

Minimum viable product

In order for any library to adopt a new system, it must provide a minimum set of functionalities that enable the library to operate its daily business. These functions are set out in the existing OLE specification as Select & Acquire, Describe and Deliver, supported by cross-module functionality.

The OLE system as seen in production in three libraries, is a proof-of-concept of this approach and should be used as a reference model for the development of the functional requirement for a minimum viable product. Studying and use the OLE specifications as a starting point for a new Library Services Platform can accelerate the process of developing the product.

Assumptions

The planned library management system will support all the commonly used services and workflows of academic libraries. It will be provided Open Source under an appropriate common license and available on GitHub.

The technical architecture will be flexible, extensible and open, supporting interoperability and future enhancements. The software will be true multi-tenant and designed to that it can be run as Software as a Service (SaaS), remotely or locally.

The system will provide integrated and flexible back-end workflows for all media types and modules listed below. It will provide access to all information requirements for library staff and front-end users (students, faculty and researchers). As basic modules the system will include metadata management, acquisitions, electronic resource management with an integrated knowledge base, circulation, systems administration and integration, analytics and business intelligence.

The electronic resource management function will provide an integrated knowledge base like the GOKb with a critical mass of content, e.g. content of KB+ and/or EBSCO.

The system will have an appealing, user-optimized interface and efficient workflows.

The system will be capable of competing (at a functional level) with next generation commercial systems like Alma or WMS, with the added benefits of open databases, interfaces and code.

Functionality

Cross-module functionality

All modules must offer:

- Comfortable and diverse search functionality;
- Processes to manage input (data import) and output (print, file, email, formatted output) automatically and manually;
- Duplication and validation check, error messages;
- Flexible customization of interfaces, templates, messages, layout of output;
- Context sensitive help;
- Systems messaging (errors, validation checks, etc.)

Metadata Management

The metadata management module(s) must support:

- All kinds of bibliographic records and holdings data (e.g. local data, item, volume), authority data;
- Shared metadata management (cataloguing) as well as stand-alone;
- Usage of external data and authority data (e.g. metadata of LOC);
- Common data exchange formats (like MARC21, MARCXML, MODS, KBART, JSON);
- Unicode support;
- Linked and open data;
- Preliminary cataloguing of titles and items for lending purposes (e.g. ILL);
- Online editor.

Acquisitions

The acquisitions module(s) must support:

- An integrated, configurable workflow comprising (pre-)selection, purchasing, receiving and inventory/invoicing;
- All kinds of material/media and order workflows, independent from order type (e.g. single order, subscription, standing order) or material/media type (print, electronic, digital or monograph, serial);
- Configurable claims and reclamation management;
- Management of obligations, invoices and payments, funds management, budget control including year transition;
- Management of vendor data, budgets, currencies, requesters, delivery addresses;
- Management of subscriptions and standing orders including claiming and binding;
- API's for Vendor data import/export (e.g. EDI, Gobi, etc.), invoice/budget data import/export (e.g. SAP);

Electronic resource management (ERM)

The ERM module(s) must support (in addition to acquisition):

- Management of all kinds of electronic resources (e-journals, e-books, databases)
- Support of the ER life cycle (pre-order, trial, order, access, invoicing, statistics, cancellation, renewal);
- Integrated access to knowledge base(s) including metadata management for title and package data, identifier;
- Package, licence and subscription management;
- Invoicing, budget control and statistical data.

Fulfilment

The fulfillment module(s) must support:

- Flexible configuration of the library's fulfillment policy for all types of library material and media including scanning/digitization requests;
- Loan (open access and closed stock), return, renewal, request and reservation at title or item level (as appropriate);

- Configurable management of claims, fines and fees;
- Interoperability with the self-service equipment including self-issue/return, book sorter and cash machines;
- Access to multiple counters.
- Reading lists and the management of course material;
- APIs for third-party systems, e. g. IDM, discovery system (standard protocols), RFID;
- ILL functionality;
- Offline fallback solution for lending and return.

User management (user data, different user types, lock entries, authorization and authentication)

The user management module(s) must support:

- Management of user data (different user types, roles, lock entries, authorization and authentication);
- Import/export of user data from/to IDM systems;
- Single Sign-on.

System administration and integration

Overall, the system and modules must support:

- Business Intelligence and analytics;
- Discovery services via API's;
- Repository functionality;
- Research and researchers data;
- Bibliometrics;
- Comfortable and flexible system configuration;
- Roles, rights and permissions management;
- Flexible workflow management;
- Customization for local library needs.

Technical aspects

The system must fulfill the requirements of data protection and data security.

The software architecture must be scalable and reliable, capable of working with large data sets. Response times must be low even for large data sets or in case of heavy traffic.

The software must follow a modular concept, which can easily be expanded by adding further or parallel modules for local or regional requirements – without effort in integrating these modules in upcoming new releases as well as for new requirements as research data management. Modules can be added or removed during runtime, without the need to restart the application.

The system must provide suitable and common APIs fulfilling the need of interoperability. It must be capable of integrating 3rd party systems (APIs for discovery, finance system, student registry system, identity management, bibliographic databases, self-service system, interlibrary loan, other API based library modules etc.).

The software must support internationalization and localization.

The software must provide extensive and comprehensive logging for software administrators.

What does the OLE Partnership bring to the Collaboration?

The OLE Partnership represents the combined knowledge and deep understanding of library functional and operational requirements built up over an extended period of time since the project was initiated in 2008. OLE Partners have contributed finances and hundreds of hours of staff time, tapping into a deep pool of professional library expertise.

The OLE Partnership provides a means for libraries to pool their resources to increase the impact of their individual investments.

The OLE Partnership has delivered the proof-of-concept solution in the OLE system implemented at three partner libraries and viable for other libraries seeking to adopt an Open Source ILS now.

Means

Fees collected from OLE Partners provide a budget to hire staff, pay for shared services, and support operations.

OLE provides organizational structures to help functional experts collaborate and socialize concepts, test ideas, provide user stories, form core functional expert teams, test and feedback on functionality.

Examples

An institution may wish to invest a developer resource working on a VuFind integration. By working with OLE, multiple developers and functional specialists from OLE Partners using VuFind can collaborate to develop the module.

An ERM librarian may wish to collaborate with other ERM librarians on workflows and integration with knowledge bases.

Method

The OLE Partnership has adopted Agile as its approach to software development and can interface with other members of the Collaboration on shared project work in addition to developing its own modules as required.

OLE adds Value

The OLE Partnership is a mature community of like-minded professionals with long experience in community-based effort.

The OLE Partnership is committed to an innovative, strategic approach for product development.

The OLE Partnership is committed to "Open" as in Open Source, Open Data, Open Access, Open and Extensible.