Open Book Collective Specification

Summary

“Open Book Collective” (OBC) is a platform and intermediary organisation that allows open-access academic publishers and infrastructure providers (“initiatives”) to promote their publishing activities and to sell and manage their funding schemes.

OBC wishes to collect revenue from institutions (predominantly academic libraries) in the USA, in the UK, in Europe, and elsewhere worldwide, to retain a portion of this to cover its institutional overhead, but then to disburse the remainder to the initiatives. Its primary corporate function is to act as a revenue collector and payment processor for these initiatives.

The initiatives that it supports, as a payment provider, could be of different legal forms (not-for-profits, charities, companies) and will be located worldwide. Given that there is an educational purpose for the public benefit in all of the initiatives that would be supported, OBC could have charitable objects. OBC has no profit-making incentive.

OBC may have an employee at a future point with a payroll. It may also own trademarks. It will host a website and own a bank account.

Platform Users and Authentication

Users of the platform can be broadly categorised as follows:

- Non-authenticated users accessing the site
- Authenticated “library” users
- Authenticated “initiative” users
- Authenticated “platform manager” users

Most functions on the platform are accessible without authentication (e.g. viewing initiatives, building a “package”.

Authentication and role-based authorization are therefore required. 2FA should be implemented.

SSO is not required.

Initiatives

At the heart of the platform are “initiatives”. These are a diverse set of service providers that range from open-access publishers through to infrastructure providers (e.g. the Directory of Open Access Books). Each of these initiatives needs to be able to display the following types of information, at a minimum:

- new/exciting publications / developments
- scholarly fields covered
- aims
- values
- governance structures
- membership packages/support options
- peer review practices
- contact details

The platform needs a CMS system that allows initiatives to edit their own content, only. The CMS system needs a rigorous history system, with reversion mechanism (limited to initiatives for their own edits, and to platform managers for all initiatives).

The CMS should have structured stylings: headings, italics, bold etc. We will use python-bleach to strip out strange formatting on paste.

There will be a central log of all CMS edits across the whole site. Platform managers will receive a notification when there are new edits.

Some of these “pages” are plain CMS. The CMS needs to be able to take pasted Word input and to strip out everything except bold and italics.

Initiative users will not be able to edit the HTML by default.

Other pages are structured. In particular, the “membership/support packages” aspect sits at the core of the platform.

The “latest/exciting publications” will be available when Thoth is being used.

Membership/Packages

At the heart of the platform sits the membership/package signup system. This is a structured set of products that can be provided by initiatives.

A "membership builder" system is needed to allow organisations to construct packages.

Specifically, because libraries may arbitrarily combine packages, the membership builder should allow construction of packages from shared elements defined by the platform manager.

An initiative using the package builder can specify:

- Controlled vocabulary attestation of standards compliance (e.g. W3C Accessibility Standards, Anti-Modern Slavery Statement) – these should also be versioned (and history is publicly viewable)
- Controlled vocabulary attestation of limited input subsets, such as WCAG (dropdown options/radio buttons) – and ability to search these
- Controlled information request fields (e.g. "address", "email") so that institutions will be asked to input everything only once at signup
- Free-text contractual terms of the package (e.g. what the initiative offers in return for payment, cancellation terms etc.) – these should also be versioned (and history is publicly viewable)
- Free-text KPIs/success metrics
- Term length for the membership
- Annual price (on a controlled vocabulary banding system that can be specified by country – and currency that is tied to country)
- Aesthetic qualities of the presentation

Controlled vocabulary elements that are available are input by platform managers. When new elements become available, initiatives should be notified that there are new attestations available (for instance).

Membership packages can be signed up to individually or as part of “meta packages” (see below). However, they can also be disabled for individual signup so that some packages are only available as part of aggregated meta packages.

An organisation may designate a single package as contributing to a platform-wide signup.

Package Setup/Fixing

Packages that are part of meta-packages need to be approved and fixed. This is because external entities, such as Jisc and LYRASIS, will have matching catalogue entries that require external confirmation. Jisc’s catalogue entry may be signed up to, independently of the workflow at Open Book Collective. In this instance, it will be necessary for the Platform Manager to be able to add an account on behalf of the library signing up. Likewise, packages that contribute to platform-wide signup must be agreed with the platform as they affect site-wide pricing.

Package Aggregation

Core to the platform is the idea that libraries might select to support one initiative, an arbitrary number of initiatives, a pre-defined collection of initiatives, or everything on the entire platform. Functionality to signup at different levels should be controlled by a master switch on the platform. The platform can run in the following modes:

- Allow/disallow signups to individual initiatives
- Allow/disallow signups to collective packages
- Allow/disallow signups to everything on the platform

Without logging in, a library should be able to build a "basket" (with customizable terminology) of initiatives and to get an annual price quote. They should be able to "save" a "basket"/package set, if they create or have an account, so that they can retrieve a previously stored quote/set of initiatives. They should be able to see, in this quote view, whether the initiatives they have selected meet their accessibility needs, their anti-slavery policies etc. as per the controlled vocabulary. The view of, say, WCAG compliance should present an overview of levels of compliance among contributing packages (e.g. a pie chart).

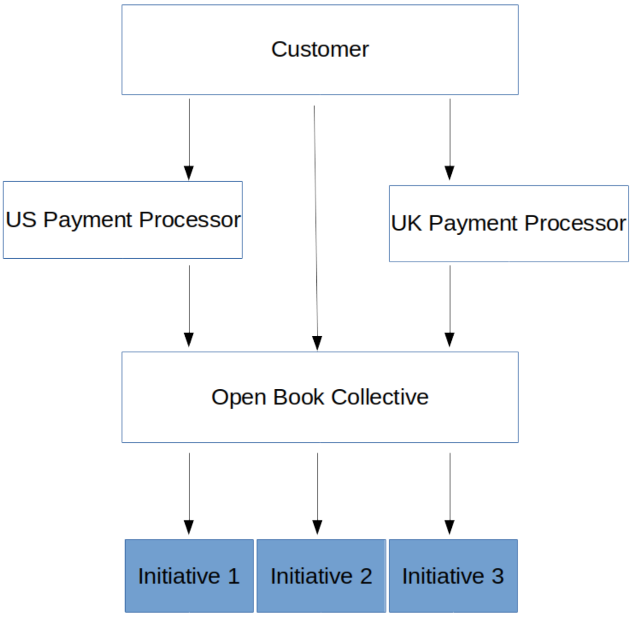
When a library has built a package with which it is happy, it has two options:

1. Signup there and then; or
2. Open a dialogue to discuss details – this should go to platform manager. Meta/collective-packages also have a contact person, though, for the platform manager to refer to if needed.

Taking Credit Cards Directly

The site does not need to take credit cards.

Payment Flow



When a library signs up for a package, there are several routing options for how billing will be handled. Each package has pricing bands that specify, for any particular geographical region, who will be handling the payment. This will likely be one of the following, but could include arbitrary third parties:

- a geographically-specific consortium, such as Jisc or LYRASIS
- Open Book Collective

The goal of the invoicing process is to minimise interaction surfaces between corporate entities. That is, individual initiatives should *not* be invoicing libraries directly for partial payment of a package signup.

However, if a meta-package contains bandings that are handled by multiple payment processors (e.g. one from Jisc and one directly from OBC), then the system must apportion the invoicing totals for each party to invoice.

Contract Management

Every package will have a "contract" associated with it. This may pertain to licensing terms, what is provided by the press, or other terms. The platform must amalgamate these contract terms into a single set of documents for libraries to be able to view and download.

Contractual terms must be changeable at the order level if a library requests changes. That is, we need to freeze the contractual terms associated with a basket – and allow for mutually-informed modification of contracts.

There should be a way of modifying contract terms, at any point in negotiations.

When a contract is agreed, it should be "frozen" so that the precisely agreed terms are stored in perpetuity. Options to preserve integrity:

- Copies of the agreed contract could be emailed to all parties
- Checksum of the contract PDF can be emailed and stored, as proof of "no tamper"

Proxy Signups

It should be possible for a platform manager to input a signup on behalf of a library. However, it needs to be very clear when this is done that there may be contractual liabilities and so forth that are not covered if an institution does not, themselves, agree to the terms.

In proxy signups there needs to be some way to store the contractual terms that may have been agreed offsite. This could include free-text input and image screenshots.

Homepage

The homepage will be designed in conjunction with the designer. However, it needs to have the following attributes and convey the following sentiments:

- Powerful
- Simple
- Collective
- Urgency
- The pitch is to librarians
- Clean/Modern/Light
- Call to action
- “Not busy”

The homepage is the central space that users encounter when they first hit the site. It needs to convey professionalism and provide an easy route to signup.

There are a number of potential “blocks” on the homepage:

- 3D graph to explore published titles (D3.js) [nice to have, not crucial necessarily for homepage]
- 3D graph to explore published topics (D3.js) [nice to have, not crucial necessarily for homepage]
- 3D graph to explore published publishers (D3.js) [nice to have, not crucial necessarily for homepage]
- New books from the collection, across the site

- Dynamic news feeds, aggregating initiative news (curated by the platform manager)
- Platform news updates
- A way of exploring packages
- A way of searching packages
- Swift route to signup [this is the priority]

The homepage also needs to provide a swift route to:

- Privacy policy
- Terms of use
- Accessibility statements

Platform Metrics

The platform needs a way of collecting some forms of usage information/page visits and to display relevant stats at the platform, library, and initiative levels.

Library Viewport

The library viewport is designed to give participating libraries access to a dashboard to manage their package signups, their invoices, to view statistics on publishing and initiatives.

Items that could be displayed when a subscribing library logs in:

- List of packages subscribed to
- Invoices (and status)
- Tailored feed (from Thoth), using Thema
- Number of authors from institution/list of author books
    - Requires GRID integration in Thoth
    - Ping/notify when a new book
- Metrics from subscribed packages:
    - “Book count” (aggregated from Thoth)
    - “Books indexed” (aggregated from DOAB/initiatives of this sort)
    - Subject areas published in (Thoth – using Thema integration)
- Video testimonials
- Institutional interactions
    - An initiative can post an institutional update, that is linked to GRID at an institution
    - We can display these in the backend

- Updates from initiatives
- “Media folder” for each package. (Packages can provide associated media that is accessible to subscribing institutions only.)
- It has been noted that the site does not wish to display metrics that competitively rank initiatives in any way

The library login area is a dashboard for administration of signups. It should, therefore, have a business-like feel.

Initiative Viewport

Initiatives primarily login to the site in order to manage their products (packages), CMS pages, and signups.

The initial viewport should facilitate:

- CMS management
- Update management
- Package management
- Shared public catalogue management
- Viewing customers/members
- Provision of usage metrics – OAPEN

This is an administrative view and should use a “dashboard” feel for this site area.

API

The platform will have an API that will produce JSON output. This will allow programmatic query of the site for initiatives, CMS information, and all other publicly accessible content. This will be achieved using the Django REST framework.

Platform Email

All emails from the platform must be GDPR compliant and have opt-out options.

Ideas for emailable events include:

- New publication from supported initiative
- New publication in subject area
- New publication from author at institution
- Upcoming renewals
- New update from initiative

Event-based email framework using Mailgun is required.

Thoth Integration

Thoth is an open metadata system developed by the COPIM project. It provides catalogue functionality and metadata in a variety of formats.

There are a number of integration points with Thoth that OBC wishes to encourage. This requires integration of Thoth with the OBC platform. The following features and functions will be developed using Thoth:

- Query per publisher
- Publisher ID
- ORM mapper for Python
- Display stats
- Search for author at institution
- Featured titles
- Thema
- GRID
- Once per day Cron job setup

We will develop these functions as part of OBC and contribute code back upstream to Thoth.

ORCID

Various types of integration with ORCID could be beneficial for the platform. At present, for "minimum-viable product", however, we are not planning any major level of integration. This should be pulled from Thoth into an author object, which is affiliated with books. One idea is: is the ORCID-affiliated institution different from the Thoth affiliation?

Programming Language, Platform, Licensing

The platform will be developed in the python programming language using the Django framework. Python is one of the most widely used programming languages in the world meaning that ongoing maintenance should be sub-contractable to any contractor worldwide. The Django framework is used in high-profile international projects such as Disqus.

Accessibility

Areas of the platform developed by the project will provide compliance to the AA standard of Web Content Accessibility Guidelines (WCAG) to the best of our ability. It is possible for individual CMS managers to embed content that may not meet this standard (say: uncaptioned videos) and this is beyond our control. However, our basic design will meet this standard and we will also provide a document setting out our appraisal of compliance. This statement can be re-used in the accessibility page of the site.

Backup and Data Security

We will develop and implement a data security policy and backup system for the site.

During development, backup will be achieved on internal systems, which includes remote offsite storage and regular pushes to centralised git repositories.

Version Control

We will develop using the distributed version control system, “git”. Initial product delivery will take the form of a unversioned “1.0” release. Subsequent iterations/work will then rebase on top of this with tracked changes. Django migrations will be supplied as an initial flat hierarchy in 1.0, with subsequent migrations applied iteratively on topof the base.