Note sur l'évolution du projet DIAL
Evaluation du POC-DSpace

Renaud Michotte – SGSI/SISG
v1 – 21/09/2023

Contexte

Le projet DIAL, le répertoire institutionnel de l'UCLouvain, a été lancé en 2011. Dans sa première phase de développement, il était destiné à accueillir les publications de recherche ainsi que leurs fulltexts. Il était destiné à accueillir des documents des 4 institutions partenaires de « L'académie Louvain » (UCLouvain, USL-B, FUCaM et UNamur).

Lors du lancement du projet, le choix initial était d'opter pour la solution commerciale « *VITAL* » fournie par la firme *VTLS.inc* (également fournisseur de la solution *Virtua* déjà utilisée en bibliothèque).

Au fur et à mesure de l'évolution du projet, le répertoire s'est enrichi de nouveaux sous-projets chacun visant un type de publication spécifique : DIAL.mem (mémoire de fin d'étude), DIAL.ebook (gestion des ebooks détenus par l'UCLouvain et ses partenaires), DIAL.pul (documents édités par les Presses universitaires de Louvain) et DIAL.num (documents patrimoniaux numérisés).

Au fil de l'évolution du projet, il est apparu nécessaire de développer une alternative à l'interface graphique fournie par VITAL ; le CMS Drupal a dès lors été choisi pour développer les nouvelles interfaces utilisateurs dans l'optique de pouvoir les adapter en fonction des besoins de chaque sous-projet.

Spécifications techniques

L'un des aspects du projet DIAL est de tenter d'utiliser autant que possible des solutions OpenSource. Hormis son interface utilisateur et les fonctionnalités liées à celle-ci, le produit VITAL était basé sur 2 principaux produits OpenSource (*FedoraCommons* et *Apache Solr*). Le remplacement de cette interface par une solution *Drupal* a permis de rendre le projet entièrement basé sur des produits OpenSource.

Tous les produits/programmes utilisés dans le projet DIAL actuellement fonctionnent encore très bien et ne posent pas de souci de sécurité. Néanmoins, certains composants sont en fin de vie et leur remplacement devient nécessaire afin de prévenir tout risque éventuel.

FedoraCommons 3.5

FedoraCommons est le système de stockage des « objets » (publications, ebook, documents numérisés, mémoires,...) utilisés dans DIAL. Actuellement, nous utilisons la version 3.5 de ce produit. La dernière release stable est la version 6.2.
La migration de la version 3.5 vers une version ultérieure n'a pas été réalisée car cela impliquait une migration profonde du modèle de données ainsi que des outils construits sur les API's fournies par *FedoraCommons*.

La version 3.5 utilisée actuellement n'a plus d'évolution prévue. La maintenance sécuritaire est opérée par la communauté, qui continue encore à l'utiliser (ex : Log4j).

Apache Solr 6

Apache Solr (développé par la communauté et soutenu par la fondation Apache) est utilisé comme moteur de recherche pour l'ensemble des projets DIAL. C'est un moteur de recherche solide et éprouvé.

La mise à jour vers une version supérieure n'a pas été jugée nécessaire car nous n'avions pas besoin des nouvelles fonctionnalités mises en place dans celles-ci. Plus d'évolution prévue mais la maintenance de sécurité est toujours active.

Drupal 7

Le CMS *Drupal* a été choisi pour remplacer l'interface utilisateur proposée par VITAL. Elle permet une customisation fine par rapport aux besoins exprimés pour chaque sous-projet DIAL. Le développement de l'interface Drupal a été lancé en 2015, il a été choisi d'utiliser la version 7 qui était bien implantée et éprouvée.

Une migration vers une version plus récente (8, 9, 10) nécessiterait une réécriture quasi complète des nombreux modules développés.

Handle server

Couplé au projet DIAL, un serveur « *Handle* » (https://handle.net/) est lié aux projets DIAL. Il permet de générer et maintenir dans le temps des permaliens vers chaque document de chaque sous-projet DIAL. Cinq « préfixes » sont gérés par ce serveur Handle : un profil général pour des publications faites entre membres d'institutions partenaires (préfixe « 2078 ») + un profil spécifique pour chaque institution initialement attachée à l'académie Louvain (préfixe « 2078.[1-4] ».

Pourquoi 5 préfixes ?

> Nous avons choisi d'utiliser un préfixe par institution afin de permettre une gestion décentralisée de ceux-ci. Ainsi, si un partenaire venait à s'éloigner, il pourrait reprendre la gestion de son préfixe dédié sur un serveur qu'il gèrerait lui-même ; un même préfixe ne pouvant pas être géré par plusieurs serveurs en même temps.

Constat actuel

Tous projets confondus, le répertoire compte actuellement environ **562.000** documents pour un espace de stockage représentant **4Tb** de données (métadonnées + fichiers attachés). Toutes les données sont répliquées sur un serveur de sauvegarde « à chaud » permettant une reprise rapide en cas de crash (8Tb sont donc utilisés) + backups incrémentaux/sur bandes opérés par SIPR.

A l'heure actuelle, toute l'infrastructure DIAL tourne correctement et les outils utilisés répondent aux besoins des différents utilisateurs (end-user, managers, bibliothécaires, …). Les versions correctives des logiciels ont été appliquées autant que possible, néanmoins…

La plateforme *Drupal7* a été lancée en 2011 et sa fin de vie est annoncée pour 2025 (https://www.drupal.org/psa-2023-06-07). *FedoraCommons 3.5* est également lancé la même année ; aucune mise à jour n'est plus planifiée sur cette version du logiciel.

La mise à jour de ces 2 outils vers des versions plus récentes n'est pas aisée car l'architecture entre la version actuellement utilisée et les versions plus récentes est complètement différente. Pour Drupal, cela nécessiterait une réécriture complète des modules développés spécifiquement pour le projet ; pour FedoraCommons, la modélisation interne des objets est radicalement différente entre les versions 3.X et les versions ultérieures, cela demanderait également une migration complexe et une réécriture d'outils basés sur les API's proposées par FedoraCommons.

Fort de ce constat, nous avons jugé qu'il devenait opportun de lancer une réflexion sur l'avenir de la plateforme DIAL afin d'assurer la pérennité du projet ainsi que la sécurité de l'accès aux données stockées.

Durant la période estivale 2023, nous avons repris plus assidument notre travail de veille documentaire sur le sujet des dépôts institutionnels afin de déterminer quelle solution pourrait être apportée pour le futur du projet DIAL.

Solutions envisagées

Note : Nous n'avons exploré que des solutions *OpenSource*. Des solutions commerciales existent (certaines basées sur ces produits OpenSource) mais n'ont pas été envisagées pour cette note. La plupart de ces solutions commerciales proposent/imposent un stockage dans le Cloud (externe UCLouvain) ce qui ne nous semble pas envisageable pour certaines ressources présentes actuellement dans le dépôt institutionnel (Brevets, fichiers en accès interdit/embargo).

FedoraCommons

Le projet FedoraCommons continue d'exister à l'heure actuelle. La dernière release stable est la version 6.4.0 sortie en mars 2023.
Le modèle interne de stockage de données est radicalement différent de celui utilisé en version 3.X (refonte complète en version 4, reprise d'un format hybride en version 5 et 6).

Avantages

- Flexibilité du format natif de données (MARCXML, MODS, ONIX, …).
- Projet suivi et maintenu par le consortium *Lyrasis.*
- Interaction via API.

Inconvénients

- Aucune interface utilisateur proposée ; il faut utiliser une interface externe qui communique avec FedoraCommons via les API.
- Complexité du modèle de données internes.
- Communauté en déclin.

Invenio RDM

La plateforme « *Invenio* » est développée et maintenue par le *CERN*. Une déclinaison RDM a vu le jour et utilise la version 3.X *d'Invenio Core*. Le modèle de données interne utilise le format *JSON* qui est très flexible pour la description des différents types de métadonnées utilisés dans chaque projet DIAL.

Avantages

- Projet développé et maintenu par le CERN.
- Accepte tous formats/types/quantité de « données » (fichiers textes, vidéos, données brutes, …).
- Technologie déjà partiellement connue et maitrisée par l'équipe *Bibsys* pour le projet RERO-ILS.
- Moteur de recherche *ElasticSearch* intégré.

Inconvénients

- Projet « jeune » avec une communauté très réduite actuellement.
- Interface utilisateur basique, nécessitant pas mal de développement pour une utilisation au sein de l'UCLouvain.

DSpace

DSpace est une plateforme « all in one » pour la construction d'un répertoire institutionnel (Stockage, interface utilisateur, moteur de recherche). Actuellement dans sa version 7.6 mais la version 8 est prévue pour début 2024.

Avantages

- Projet suivi et maintenu par le consortium *Lyrasis.*
- Large communauté d'utilisateur partout dans le monde + Documentation online.
- Interface utilisateur intégrée (*Angular*).
- Architecture Client/Server communicant via des API's.
- Moteur de recherche *Apache Solr* intégré.
- Utilisé pour OER (reprise de l'outil par l'équipe possible).

Inconvénients

- Modèle de données interne difficilement flexible (*DublinCore qualified*).
- Complexité de configuration.
- Customisation interface utilisateur possible mais peut se révéler complexe en fonction du niveau de modification à opérer.

Avis …

Après avoir analysé et pris quelques contacts, il apparait que la solution ***DSpace*** semble la solution la plus pertinente à l'heure de faire le choix d'une solution pour l'implémentation d'un répertoire institutionnel.

DSpace est le leader dans le domaine des répertoires institutionnels dans le monde aujourd'hui. Les deux autres solutions sont nettement moins suivies/utilisées par la communauté : *FedoraCommons* est clairement en déclin depuis le changement de modèle de données en version 4.X et peine à retrouver des « clients » ; *Invenio RDM*, bien que plus récent, peine à se faire une place dans le domaine des répertoires institutionnels et mise plus sur son aspect RDM.

De plus, le fait que *DSpace* propose une solution « tout en un » est un atout majeur en faveur de ce produit : Il n'est en effet plus nécessaire de développer entièrement une nouvelle interface, mais « juste » d'adapter l'interface proposée.

L'architecture proposée par *DSpace* dans sa version 7 permet de bien dissocier la partie « gestion des objets » de la partie « présentation des objets ». En effet, la partie « serveur » expose toute une série d'API qui sont utilisées pour obtenir/gérer toutes les informations d'objet. La partie « client » utilise uniquement ces API's pour construire l'interface utilisateur. Ce choix technique permet de pouvoir connecter des outils externes (monitoring, stats, …). De plus, les langages utilisés dans chaque partie de l'application (***Java*** pour la partie serveur, ***Angular*** pour la partie client) sont déjà connus de l'équipe Bibsys.

Il existe également des modules complémentaires pouvant être ajoutés au cœur de DSpace. Citons par exemple un module permettant une intégration complète entre *DSpace* et le portail *ORCID* (pour les publications de recherche), un module permettant la visualisation de documents via le protocole *IFFF* (pour les documents numérisés), …
Certaines firmes commerciales ont également développé leurs propres modules et proposent, moyennement paiement, l'installation de ceux-ci sur une instance DSpace (stats, reporting, monitoring, UX, …).

A noter également que nos collègues de l'ULB ainsi que de ULiège ont également choisi ce produit :

- l'ULB disposant déjà d'une version de DSpace 1.X va migrer vers une nouvelle installation de DSpace 7.X (ou 8). Le travail de développement et de migration est prévu pour 2024.
- ULiège a complètement réécrit son répertoire institutionnel en 2021, en utilisant une version 6 de DSpace. A noter qu'il y a beaucoup de changement entre la version 6 et 7, ainsi ULiège semble « bloqué » dans sa version actuelle sans pouvoir « facilement » migrer vers une version plus récente.

La version 8 de DSpace est annoncée pour avril 2024. La volonté est de sortir une version majeure chaque année en avril (autant que faire se peut). L'évolution de DSpace7 vers la version 8 ne devrait donc pas être compliquée car aucun changement structurel n'est prévu dans l'architecture du produit et du modèle de donnée.

… La suite : POC

L'équipe Bibsys propose de créer un « *proof of concept* » (POC) avec le logiciel DSpace 7.6.
La version 8 étant encore instable et peu documentée – une migration de 7 à 8 est décrite comme « relativement aisée » [source : Slack] car la version 8 est une évolution de la version 7 en y intégrant toutes les caractéristiques principales en cours de développement[1].
Il nous semble que le projet « DIAL.mem » serait à même d'être testé pour ce POC. car il regroupe la plupart des aspects que nous jugeons nécessaire d'analyser afin de valider l'hypothèse DSpace :

- Connection utilisateur : Connection SSO avec certaines restrictions. Seuls les étudiants éligibles à un dépôt de mémoire ainsi que des gestionnaires/administrateurs doivent pouvoir se connecter à cette interface. De plus, seuls les étudiants/administrateurs peuvent soumettre un nouveau document.
- Développement d'API's personnalisées : A la fin du processus de soumission, une attestation doit être envoyée à l'étudiant (au format PDF). Cette attestation doit pouvoir être regénérée au besoin. La génération de cette attestation doit être intégrée dans les API's exposées par le serveur DSpace.
- Workflow de validation : Lorsqu'un mémoire est déposé par un étudiant, il n'est pas directement visible, il doit faire l'objet d'une validation de la part d'un gestionnaire de mémoire en faculté. Ce n'est qu'une fois l'approbation effectuée que la mémoire devient

[1] https://wiki.lyrasis.org/display/DSPACE/RoadMap#RoadMap-FutureResearch/Planning

visible pour le public (moyennant les restrictions d'accès choisies par l'utilisateur lors de la phase de soumission).

- Personnalisation de l'interface : Faire en sorte que l'interface utilisateur corresponde au style « UCLouvain » et que les fonctionnalités nécessaires y soient intégrées.
- Exposition des données : Certains outils externes doivent pouvoir extraire/utiliser des données pour leurs propres usages (moyennement authentification/autorisation).
- Gestion des droits : n'importe quel gestionnaire ne peut pas valider/modifier n'importe quel mémoire.
- Customisation du formulaire de soumission : Certains champs du formulaire de soumission doivent pouvoir être rempli par un système *d'autocomplete* ou par des valeurs renvoyées par un appel asynchrone à *l'ESB* (*OSIS*).

Nous pensons qu'il est plus raisonnable de commencer avec un sous-projet DIAL qui ne contient qu'un seul type de document ayant toujours le même comportement (contrairement aux publications de recherches par exemple). Le nombre de document lié à ce projet est plus raisonnable que les publications de recherche mais assez important pour pouvoir faire un benchmarking cohérent (37.500 vs 235.000)

Pour terminer, la présence d'un chef de projet pérenne sur le projet « DIAL.mem » nous permet d'avoir un interlocuteur fonctionnel connaissant bien le terrain et le produit actuel. Celui-ci pourra plus aisément nous faire part de son feedback.

Sollicitation

Bien que *DSpace* soit un produit OpenSource bien documenté, il n'en reste pas moins complexe à comprendre et difficile à appréhender, surtout dans certains aspects spécifiques/techniques. C'est pourquoi nous jugeons utile de solliciter une aide de « mise en route » auprès de professionnels utilisant et maitrisant tous les aspects du produit. Nous avons déjà pris contact avec plusieurs firmes proposant du support DSpace afin de tâter le terrain.

L'idée est qu'avec un petit coup de pouce au début du projet, on puisse gagner beaucoup de temps/connaissance pour le reste de l'implémentation des autres sous-projets « DIAL ».

L'aide proposée diffère en fonction des sociétés, mais prendrait la forme d'un service de ticket sur base de 50h d'échange (technique ou fonctionnel). Une première estimation est de l'ordre de +/- 7000€ (à affiner/confirmer).