



HAL
open science

Migration de l'Université de Strasbourg sur Partage... la suite

Xavier Pierre, Sébastien Finkbeiner, Laurence Moindrot

► To cite this version:

Xavier Pierre, Sébastien Finkbeiner, Laurence Moindrot. Migration de l'Université de Strasbourg sur Partage... la suite. JRES (Journées réseaux de l'enseignement et de la recherche) 2021, Renater, May 2022, Marseille, France. hal-04807369

HAL Id: hal-04807369

<https://hal.science/hal-04807369v1>

Submitted on 27 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Migration de l'Université de Strasbourg sur Partage... la suite

Xavier Pierre

Direction du numérique
Université de Strasbourg
14, rue René Descartes

Sébastien Finkbeiner

Direction du numérique
Université de Strasbourg
14, rue René Descartes

Laurence Moindrot

Direction du numérique
Université de Strasbourg
14, rue René Descartes

Résumé

Suite à la migration de l'Université de Strasbourg vers la messagerie mutualisée Partage opérée par RENATER, il nous est apparu important de maintenir une vision communautaire de la messagerie et de ses services associés.

À ce titre, l'Université de Strasbourg a ressenti le besoin de renforcer le lien entre les utilisateurs de Partage et leurs informaticiens de proximité. Pour cela une délégation de droits semblait nécessaire afin de permettre à ces personnes en charge de gérer les comptes de messagerie des composantes/laboratoires/autres de :

- créer des liens de proximité ;*
- améliorer la satisfaction des utilisateurs ;*
- accélérer le temps de traitement des demandes.*

Cette possibilité de délégation n'étant pas présente dans le panneau d'administration de Partage (BSS), nous avons donc décidé de l'implémenter dans notre propre application de gestion des comptes de messagerie (Mélusine).

Mélusine reprend ainsi l'ensemble des fonctionnalités exposées par l'API de Partage en s'interfaçant avec notre Système d'information, elle constitue ainsi le nouveau référentiel des comptes de messagerie.

Notre objectif est maintenant de reverser cette application à la communauté en la rendant la plus générique et la plus interfaçable possible afin qu'elle puisse être utile à tous.

Durant cette présentation, nous vous exposerons le déroulement de ce projet. Nous justifierons nos choix, de la phase de développement à la mise en œuvre !

Mots-clefs

Messagerie, Django, Python, Référentiel, Agile ...

1 Introduction

Aujourd'hui un grand nombre d'établissements publics de l'Enseignement Supérieur et de la Recherche (ESR) se tournent vers des solutions numériques mutualisées. Ce choix favorise les échanges, la communauté, la diversification des savoirs et tout simplement l'entraide. Une limite est cependant atteinte, celle de la cohérence des outils mutualisés avec les systèmes d'information propres à chaque établissement.

C'est dans cette optique que l'Université de Strasbourg a réalisé le projet « Mélusine ». Cette application a pour objectif de devenir un référentiel de messagerie basé sur Partage et connecté au Système d'Information (SI) de l'établissement.

2 Historique

Préalablement à la migration de l'Université de Strasbourg vers Partage, l'ensemble des comptes de messagerie étaient fortement liés aux comptes dans le SI : à tout compte du SI était associé un compte mail, avec une boîte mail, qu'elle soit utilisée ou non. Le passage sur Partage a représenté un changement de paradigme important.

Ces changements n'étaient pas liés à la confidentialité des données mais à la façon dont nous gérons chaque compte de messagerie. Auparavant, une boîte mail ne générait aucun coût d'un point de vue *stricto sensu* ; seul l'espace de stockage pouvait représenter un impact financier. Le passage à Partage, dont le prix facturé est notamment fonction du nombre de boîtes, nous fait revoir l'attribution automatique des comptes de messagerie.

Nous avons également rencontré une différence importante quant à la gestion des comptes issus de nos référentiels puis provisionnés sur la plateforme mail. Auparavant, les informations relatives aux comptes mails étaient directement stockées au niveau du compte dans l'annuaire LDAP. Ce dernier était ainsi la source d'information de bon nombre d'informations de messagerie : adresses mail, alias, filtre de messagerie, etc.

De même, grâce à un mécanisme de gestion des droits, il était possible de déléguer aux correspondants informatiques des composantes et laboratoires une partie de la gestion des **110 000 comptes** de messagerie, notamment pour leurs utilisateurs directs.

C'est donc dans l'objectif de retrouver ce lien fort entre la plateforme de messagerie (Partage) et notre SI que nous avons entrepris le développement d'une application faisant office de référentiel de messagerie tout en assurant la synchronisation des données entre Partage et notre SI.

3 Les besoins

3.1 Général

La gestion d'une messagerie est une tâche complexe de part la confidentialité des informations qui peuvent y circuler et dans le cas de l'université de Strasbourg, par la quantité des comptes à gérer (~110 000). Cette gestion implique quotidiennement des interventions et des demandes spécifiques. Sur Partage, cette gestion ne peut être réalisée que par les deux administrateurs référents de la messagerie. Il n'était pas raisonnable de donner les droits d'administration à la centaine de correspondants informatique.

Le développement de l'application *Mélusine a* représenté un réel investissement de la part de l'Université de Strasbourg. Les deux principaux objectifs étaient d'améliorer l'ensemble des synchronisations entre notre SI et Partage, et de réduire le temps de traitement des demandes utilisateurs tout en leur proposant le service le plus qualitatif possible.

3.2 La délégation

L'Université de Strasbourg compte un grand nombre d'informaticiens de composantes et de laboratoires ayant autorité pour gérer leurs propres utilisateurs. Il était important que ces personnes, référencées comme correspondants informatique, puissent continuer à administrer la partie messagerie de leurs comptes. Pour cela, la mise en place d'une délégation était nécessaire afin de pouvoir donner des droits spécifiques à ces personnes sur les comptes de leurs utilisateurs.

Bien que Partage propose un système d'administration, celui-ci est lié à l'ensemble d'un domaine. De plus, les droits d'administration ne sont pas personnalisables, ce qui ne correspond pas à notre politique d'exploitation. Il n'est en effet pas possible de restreindre les droits de gestion à une partie de la population d'un domaine comme un laboratoire ou encore une composante.

Historiquement, nous avons **178 sous-domaines de messagerie** dans lesquels sont appliqués des droits précis à des correspondants. Cette finesse leur permet d'administrer l'intégralité des boîtes mails d'un domaine ou uniquement une partie de la population dans un domaine. L'ensemble des comptes possèdent au moins une adresse dans un des deux domaines principaux @unistra.fr et @etu.unistra.fr. Les sous-domaines permettent de créer des alias aux personnes dans ces sous domaines.

La délégation a donc été mise en place via un système pré-existant dans notre SI permettant d'associer un groupe à chaque utilisateur ainsi qu'un responsable à ce même groupe. *Mélusine* repose sur ce mécanisme pour associer un correspondant informatique aux personnes qu'il peut gérer.

3.3 Référentiel de messagerie

L'une des améliorations que nous souhaitons également réaliser avec *Mélusine* était de ne plus s'appuyer sur l'annuaire LDAP afin de synchroniser les comptes vers Partage. Avoir une application dédiée servant de **référentiel de messagerie** nous permet de propager de façon synchrone les comptes sur Partage et dans notre système d'information de manière fiable.

4 Le projet

4.1 L'équipe projet

Ce projet a été mené en mode Agile. 7 personnes ont été impliquées :

- 1 product owner ;
- 1 scrum master ;
- 3 développeurs seniors ;
- 2 développeurs juniors.

Les deux développeurs juniors étaient deux apprenti·e·s qui ont été engagé·e·s à la Direction du numérique pour une période d'un an. Après l'étude préliminaire, la durée de développement nécessaire à la réalisation complète de cette application avait été estimée à 200 j/h de développeurs junior (2 apprenti·e·s) et 40 j/h de développeurs seniors.

4.2 Le socle technique

Le choix des technologies nécessaires à la réalisation de ce projet s'est fait sur plusieurs critères :

1. réutilisation de scripts d'exploitation existant ;
2. fonctionnalités nécessaires pour répondre aux besoins ;
3. socle technique de la Direction du numérique.

En prenant en compte ces différents éléments, le choix a été fait d'utiliser le framework *Django* pour la partie *Backend* de l'application et *Vue.js* pour la partie *Frontend*.

Nous avons également utilisé la librairie *Bravado* (*Swagger*) afin d'avoir des appels à l'API de Partage simplifiés et maintenables facilement dans le temps.

Le fait de respecter les critères cités précédemment permet d'avoir une application facilement maintenable par plusieurs personnes au sein de la Direction du numérique. Grâce à ces choix techniques, les scripts d'exploitation déjà en place et rédigés en python ont pu être réutilisés et adaptés à *Mélusine*.

4.3 Mélulib

Dans une approche modulaire et au travers de ce projet, nous avons écrit *Mélulib*, une librairie Python complémentaire et indispensable au fonctionnement de l'application *Mélusine*.

Mélulib contient des méthodes essentielles utilisées par *Mélusine* telles que l'ajout et la suppression d'alias ou la modification de l'adresse mail principale d'un compte de messagerie. En outre, elle repose sur une quantité importante de méthodes exposées par l'API Partage, nous permettant ainsi de les réutiliser de manière aisée et lisible dans le code de l'application *Mélusine*.

Cette librairie dispose également d'une autre particularité : elle nécessite des fichiers de configuration communs à *Mélusine* pour pouvoir fonctionner. Le chemin est indiqué via une variable d'environnement commune aux deux entités.

Mélulib n'a pas pour seule vocation d'être utilisée de concert avec *Mélusine* : elle peut être réutilisée dans la réalisation de scripts de synchronisation entre Partage et l'annuaire LDAP, ou dans le cadre d'autres applications.

4.4 La Méthodologie

Tout comme le choix des technologies basées sur notre socle technique, la gestion de projet a été faite en conformité avec la méthodologie projet de la Direction du numérique. C'est donc tout naturellement que nous avons utilisé la méthode Agile.

La gestion de ce projet en méthode Agile avait pour particularité que la Direction du numérique était le client et le réalisateur du projet.

Nous avons choisi des sprints planning d'une durée de 1 semaine accompagnés de daily sprint pour les deux développeur·euse·s principaux.

Pour l'organisation du projet, l'outil Gitlab (serveur interne) a été utilisé, nous y avons intégré l'ensemble des éléments de gestion du projet (User Story) ainsi que le code.

Durant le projet ~50 *user stories* ont été nécessaires avant que la première version de *Mélusine* voit le jour.

4.5 La mise en place

La première étape a été la mise en place :

- de l'architecture de *Mélusine* ;
- des environnements de test et de production ;
- des méthodes de déploiement avec l'utilisation de notre outil interne *Pydiploy* ;
- ainsi que de l'architecture du code.

Du temps a également été consacré à la formation des deux apprenti·e·s travaillant sur le projet afin de les former aux outils du socle technique de la Direction du numérique. Dans un premier temps, une formation Linux a été faite, qui est le système d'exploitation principal de la direction de numérique (clients ou serveurs). Un choix de leur IDE a également été fait. Pour cela la Direction du numérique propose une licence JetBrains Pycharm pro qui est tout à fait adapté à ce genre de

projet (Django, Python), mais nous avons également des experts sur d'autres outils comme Vscodé, Vim. Ce choix a permis aux apprenti·e·s de choisir ce qui leur correspondait le mieux tout en ayant un accompagnement.

4.6 Le déroulement

Ce projet s'est déroulé en deux grandes phases distinctes initiales et indispensables à la mise en place de *Mélusine*. Une troisième phase est envisagée dans le but d'élargir l'ouverture de *Mélusine* aux utilisateurs finaux de la messagerie.

La première phase consistait à reprendre l'ensemble des fonctionnalités proposées par l'API de Partage et nécessaires à la gestion d'une boîte mail. À partir de cette réalisation les deux administrateurs de messagerie avaient la possibilité de travailler exclusivement sur *Mélusine* pour l'exploitation quotidienne, c'est-à-dire en se passant complètement du Business Support System (BSS) qui est l'interface d'administration native de Partage

La seconde phase était quant à elle axée sur la délégation de droits pour ouvrir *Mélusine* aux correspondants informatiques.

5 Les fonctionnalités

5.1 L'API BSS

Dans un premier temps, nous avons réutilisé une partie de nos scripts de synchronisation existant et se basant sur l'API de Partage que nous avons complétés par l'ensemble des fonctionnalités nécessaires à la gestion d'un compte de messagerie :

- ajout/suppression d'un compte ;
- ajout/suppression d'un alias ;
- modification du quota ;
- restauration d'un compte ;
- modification de la COS (Class Of Service) ;
- ...

L'ensemble des méthodes de l'API de Partage suffisent à elles seules le maintien usuel et la gestion quotidienne d'un compte de messagerie.

En parallèle de l'usage de l'API de Partage, des appels à l'API SOAP de Zimbra permettent des actions plus spécifiques directement sur les comptes mails, comme, l'ajout de filtres de messagerie.

5.2 Les métriques

En plus des fonctionnalités d'exploitation, un tableau de bord avec des métriques que nous avons jugées pertinentes a été réalisé. Ces métriques sont diverses et évolutives, citons :

- le nombre de boîtes par COS ;
- le nombre de boîtes réellement facturées ;
- les coûts mensuels de Partage ;
- le nombre de personnes connectés à Mélusine ;
- l'évolution du nombre de comptes ;
- l'évolution de la volumétrie ;
- le statut de l'API.

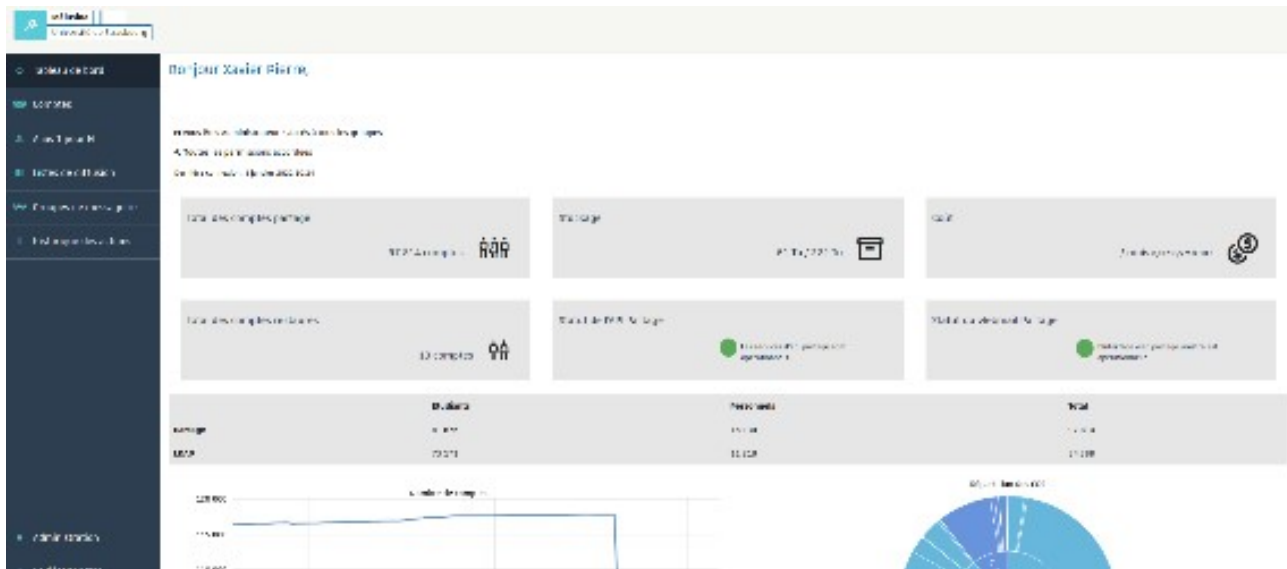


Figure 1: Tableau de bord et métriques

6 La suite

L'objectif final de Mélusine est d'être reversé à l'ensemble de la communauté. Pour cela une grande partie du code a été pensée de façon à être le plus adaptable possible à l'environnement et au SI de chacun.

Les environnements et les méthodes de gestion de chaque établissement étant différents, Mélusine nécessitera des adaptations afin de la rendre la plus générique possible. Pour cela nous accompagnerons autant que possible les établissements souhaitant l'utiliser.

Mélusine est actuellement hébergée et documentée exclusivement sur le *Gitlab* de l'Unistra. Cela peut être amené à changer selon les besoins de la communauté.

7 Conclusion

En migrant sur Partage, nous savions que cela impliquerait un certain nombre de changements. Nous avons conscience de la perte d'autonomie dans l'administration des comptes de messagerie. Cela n'a pas remis en cause notre choix.

Nous savions en effet qu'il était possible de remédier à cela, et c'est ce qui a été fait avec le développement de *Mélusine*. Cette application permet en effet de résoudre tous nos besoins internes vis-à-vis de la gestion des comptes de messagerie et d'améliorer ce que nous proposons avant la migration sur Partage.

De plus, cela ouvre de nouveaux horizons d'un point de vue communautaire et dans l'évolution de Partage. Nous espérons que cette application, qu'elle soit réutilisée ou non, puisse montrer qu'il est possible de réaliser des projets sur Partage, comme sur d'autres outils mutualisés, afin d'y proposer des améliorations.

Pour terminer, l'ensemble des choix faits jusqu'alors sont toujours assumés, sur la partie migration Partage, tout comme sur l'évolution des fonctionnalités avec l'exemple de *Mélusine* qui renforce notre volonté de participer à une réelle communauté.

Il est également important de rappeler que la grande majorité de *Mélusine* et *Mélulib* a été développée par deux apprenti·e·s – Léa Stroh en formation au CESI et Maxence Lux en formation à SUPInfo – qui ont su montrer leurs capacités techniques et sociales et par ce fait, renforcer notre envie de confier des projets de cette envergure à des apprenti·e·s.

Bibliographie

[1] **Xavier Pierre** : Migration de la messagerie de l'Université de Strasbourg vers une solution mutualisée d'outils collaboratifs

Sources et Références

API Partage :

- <https://documentation.partage.renater.fr>

API Zimbra :

- https://wiki.zimbra.com/wiki/Zimbra_REST_API_Reference