



HAL
open science

Zimbra pour l'Université de Bordeaux : mise en œuvre et bilan

Laurent Lavaud

► To cite this version:

Laurent Lavaud. Zimbra pour l'Université de Bordeaux : mise en œuvre et bilan. JRES (Journées réseaux de l'enseignement et de la recherche) 2015, Renater, Dec 2015, Montpellier, France. ⟨hal-04805278⟩

HAL Id: hal-04805278

<https://hal.science/hal-04805278v1>

Submitted on 26 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 CC BY-NC 4.0 - Attribution - Non-commercial use - International License

Zimbra pour l'Université de Bordeaux : mise en œuvre et bilan

Laurent LAVAUD

Direction des Systèmes d'Information
Université de Bordeaux
Place Pey-Berland
33000 Bordeaux
www.u-bordeaux.fr

Résumé

Fin 2010, trois des quatre universités bordelaises s'engagent dans un processus de création d'un établissement unique, la « nouvelle université de Bordeaux ». Trois ans plus tard, le 1er janvier 2014, « l'université de Bordeaux » naît officiellement.

Dans ce contexte, de grands chantiers sont lancés. L'un d'eux, conduit par la Direction des Systèmes d'Information, consiste à fusionner les messageries existantes. L'objectif est de proposer aux utilisateurs une plateforme collaborative unique, efficace et facilitant les échanges : Zimbra.

Un an plus tard, la totalité des migrations a été faite, soit près de 6000 personnels et plus de 45000 étudiants.

Après avoir présenté le contexte et les enjeux de ce changement, nous verrons quelles solutions matérielle et logicielle ont été choisies pour atteindre l'objectif.

Puis nous détaillerons comment, en développant des procédures s'appuyant sur le « service de bus d'entreprise » (ESB) RabbitMQ, nous sommes parvenus à migrer l'ensemble des comptes.

Enfin, nous verrons comment les outils de reporting Birt nous permettent d'établir des tableaux de bord, et quel rôle a joué le framework Django dans l'écriture de sites d'exploitation.

Mots-clefs

Messagerie, Fusion, Zimbra, RabbitMQ, Cyrus, Django, Birt

1 Contexte et enjeux

1.1 Contexte

Fin 2010, les universités de Bordeaux 1 (Sciences et Techniques), Bordeaux 2 (Sciences de l'Homme, de la Vie et de la Terre), Bordeaux 4 (Droit, Sciences Economiques et Sciences Politiques) ainsi que le PRES de Bordeaux s'engagent dans un processus de création d'un établissement unique, la « nouvelle université de Bordeaux ». L'université Bordeaux 3 (Lettres) ainsi que d'autres écoles et instituts choisissent de ne pas rejoindre ce mouvement.

A ce stade, chaque établissement engagé dispose de son propre dispositif de messagerie, géré par l'équipe système de son Centre de Ressource Informatiques. Ainsi, les universités de Bordeaux 1, 2 et 4 délivrent le service à l'aide de la solution libre *cyrus*, alors que le PRES a choisi *Exchange*.

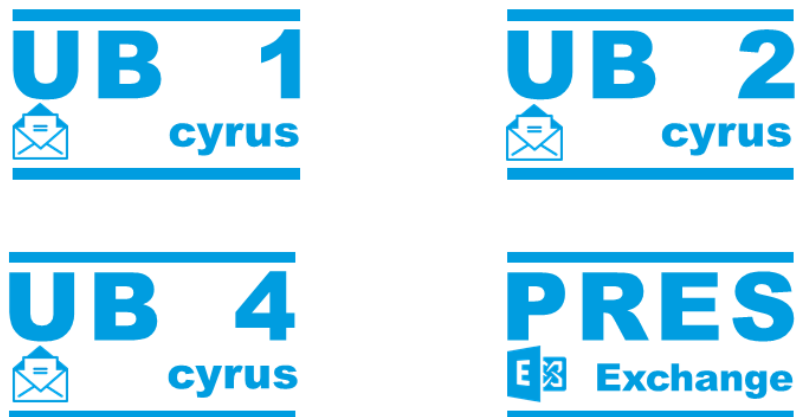


Figure 1 - Solutions en 2013

Pour fournir un accès web aux utilisateurs, chaque établissement a également sa solution : respectivement *SOG*, *Horde*, *EGroupWare* et *Outlook Web Access* pour Bordeaux 1, Bordeaux 2, Bordeaux 4 et le PRES. Aucune ne donne entièrement satisfaction.

Côté dénombrement, les forces en présence sont les suivantes :

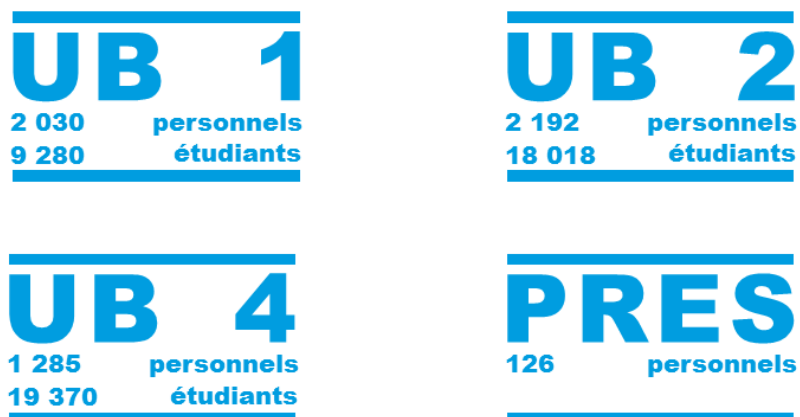


Figure 2 - Bilan social avant fusion

Il est à noter également que ces chiffres ne font pas état du personnel hébergé (issu d'unités mixtes p.ex.), ces informations n'étant pas saisies ni remontées au niveau du bilan social par les Ressources Humaines.

De plus, possibilité est donnée aux personnels et étudiants de conserver leurs comptes pendant une période de recouvrement correspondant p.ex. au renouvellement de leur contrat ou leur réinscription.

Ainsi, les besoins réels en nombre de boîtes se situent autour de **78 000**.

1.2 Enjeux

Compte tenu de l'hétérogénéité des systèmes en place, le premier objectif est de fournir un accès identique à l'ensemble du personnel et des étudiants.

Côté fonctionnel, nous voulons être au minimum à iso fonctionnalités de l'ensemble des messageries existantes.

D'autre part, nous souhaitons également approfondir les usages de travail collaboratif, notamment les échanges autour des calendriers et des boîtes partagées.

Enfin, un accès web de qualité à l'ensemble des fonctionnalités produit est indispensable, l'objectif étant de réduire les interventions sur le poste de travail en client lourd.

Après avoir vu les contextes et enjeux de ce projet, passons maintenant à la description de la solution technique de la nouvelle plateforme de messagerie.

2 La solution technique

Regardons les deux aspects de cette solution : la partie logicielle et la partie matérielle.

A noter que le partenaire retenu était *Dell* car il disposait à la fois des compétences sur la partie matérielle mais également sur la partie logicielle. En effet, un expert *Zimbra* nous apportait la garantie d'une parfaite adéquation logiciel/matériel. Ce dernier fait aujourd'hui partie de la société *Zimbra*.

2.1 Partie logicielle

Une étude menée de juin 2012 à janvier 2013 nous amène à retenir la solution de messagerie collaborative *Zimbra*. Son large déploiement dans les universités, le fait qu'elle soit le socle de l'offre Renater « Partage », et les développements en cours *Esup*, *Amue* et *Cocktail* ont ainsi retenu notre attention.



Figure 3 - Solution cible pour 2014

Cette solution, également déployée à grande échelle chez de nombreux providers, dispose des fonctionnalités principales suivantes :

- Interface web avancée sous ajax

- Aussi riche qu'un client lourd, avec menus contextuels
- Indexation des messages, des pièces jointes pour une recherche exhaustive
- Calendriers avec recherche rapide de disponibilité des participants lors de la prise de rendez-vous
- Partage aisé de calendriers, contacts, dossiers de messagerie
- Administration web
 - Gestion de l'ensemble des paramètres
 - Sauvegarde et restauration des boîtes
 - Intégration antivirus et antispam
 - Api web et interface REST
- Connectivité et protocoles
 - Support MAPI, IMAP(S), POP(S), S/MIME, CalDAV
 - Accès mobiles iOS, Android, BlackBerry et Windows

Pour le déploiement de la solution, nous avons choisi une approche virtualisée afin de garantir une haute disponibilité dans un premier temps, puis une solution de reprise en cas de sinistre dans un second temps. *VMWare* en version 5.6 a été retenu pour atteindre ces objectifs.

Le système d'exploitation des machines virtuelles est *CentOS* en version 6, officiellement supporté par *Zimbra*.

Au final, les briques *Zimbra* composant l'écosystème sont :

- 2 serveurs proxy en actif/passif pour les accès web et imap
- 2 serveurs ldap en mode multi-maîtres pour la gestion des comptes et paramètres, inaccessibles d'Internet
- 6 serveurs ici appelés « store » gérant l'ensemble des mails, contacts, agendas et tâches, également inaccessibles directement d'Internet

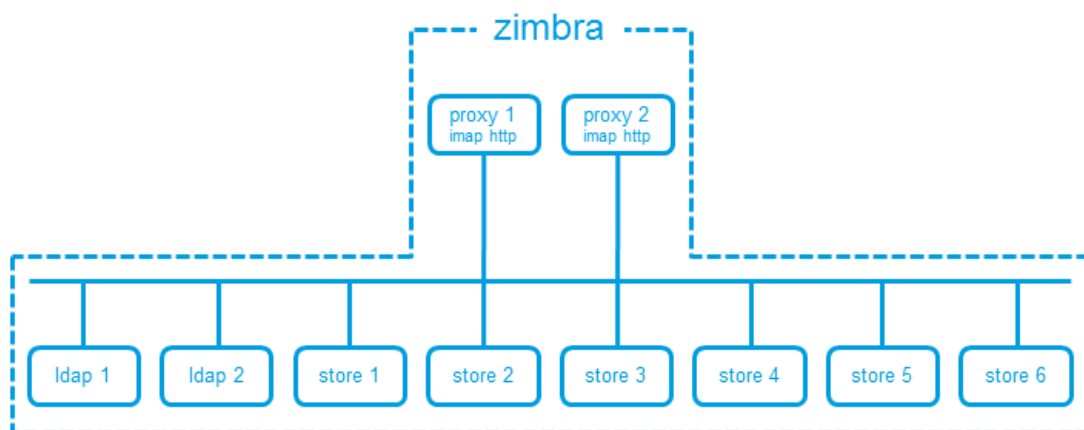


Figure 4 - Briques composantes de Zimbra

2.2 Partie matérielle

Nous avons travaillé en deux temps : installation d'une salle principale début 2014, puis installation et interconnexion d'une salle de secours en fin de cette même année.

2.2.1 Salle principale

Au vu des éléments que nous avons fourni tant en termes de volumes que de débits, l'architecture initiale retenue était la suivante :

- 3 serveurs physiques bi-pro, 128Go de RAM, 2 ports 10Gb ethernet, 4 ports 1Gb ethernet
- 1 baie de stockage « rapide » avec 24 disques 300Go SAS pour les mails récents
- 1 baie de stockage « capacitive » avec 24 disques de 1To NL-SAS pour les mails peu consultés
- 1 baie de stockage « capacitive » avec 24 disques de 3To NL-SAS pour les sauvegardes
- 2 switches 10Gb ethernet

Une année d'exploitation plus tard, au gré des évolutions de volumétrie, nous avons ajouté :

- 1 baie de stockage « rapide » avec 24 disques 300Go SAS
- 1 baie de stockage « capacitive » avec 24 disques de 4To NL-SAS

Nous disposons donc aujourd'hui de l'architecture suivante :

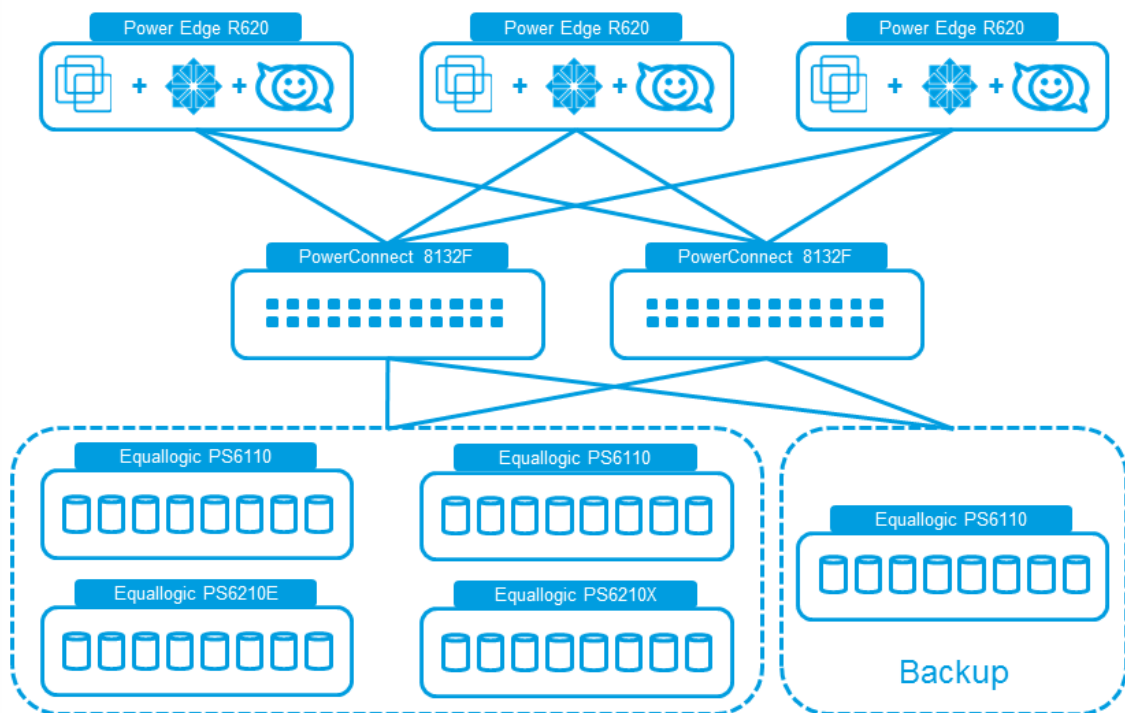


Figure 5 - Architecture de la salle principale

2.2.2 Salle de secours

Afin de disposer d'une solution de repli en cas de sinistre, nous avons déployé quelques mois après dans une salle de secours :

- 3 serveurs physiques bi-pro, 128Go de RAM, 2 ports 10Gb ethernet, 4 ports 1Gb ethernet
- 2 baies de stockage « capacitives » avec 24 disques de 4To NL-SAS
- 2 switches 10Gb ethernet

Nous disposons donc aujourd’hui de l’architecture suivante :

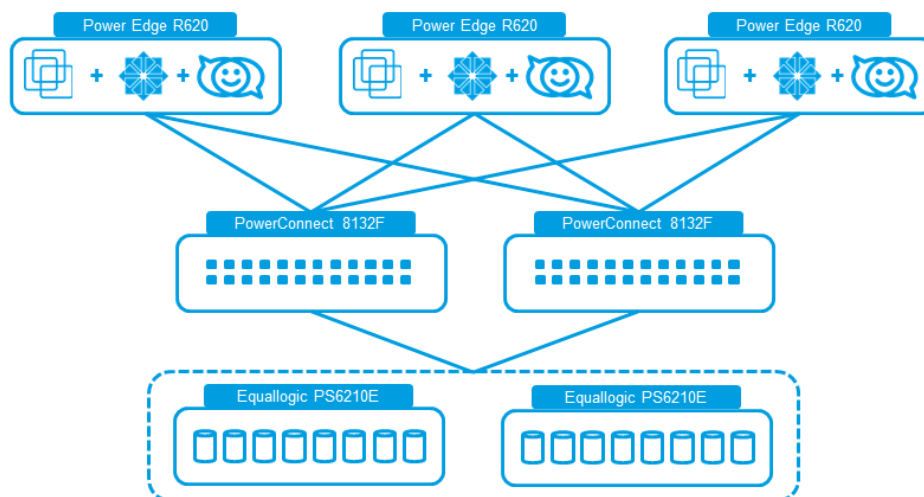


Figure 6 - Architecture de la salle de secours

Côté réseau, nous avons dédié un lien 1Gb/s entre les deux salles, distantes de plusieurs kilomètres.

Compte tenu de la capacité de la liaison et des fonctionnalités des baies *Equallogic*, nous nous sommes orientés vers des répliquions asynchrones, programmées sur une fréquence de 5 minutes. A l’usage, il s’avère que ce délai est suffisant pour traiter les modifications apportées aux données sur cette période.

3 Migration des comptes

Après avoir installé et validé la solution logicielle sur l’infrastructure matérielle, nous nous sommes attaqués au processus de migration des comptes.

Compte tenu de la volumétrie à traiter, il nous a fallu envisager une migration progressive, sur plusieurs mois.

Pour atteindre cet objectif, nous avons dans un premier temps mis en place un écosystème autour de la solution *Zimbra* afin de migrer de manière individuelle chaque compte : relais smtp et proxys imap.

Puis nous avons développé des procédures manuelles et automatiques de création et migration de comptes.

3.1 Mise en place de relais de courrier entrant

Dans un premier temps, nous avons mis en place trois proxys smtp qui sont devenus les serveurs officiels des domaines u-bordeaux1.fr, u-bordeaux2.fr et u-bordeaux4.fr. Ces derniers, connectés à l’annuaire ldap fusionné des établissements, étaient chargés de rediriger les flux entrants.

Ce routage, calculé en fonction d'un attribut spécifique à chaque entrée ldap permettait de délivrer les mails vers le serveur de l'ancien établissement si le compte n'avait pas encore été migré, ou vers l'infrastructure Zimbra une fois le compte migré.

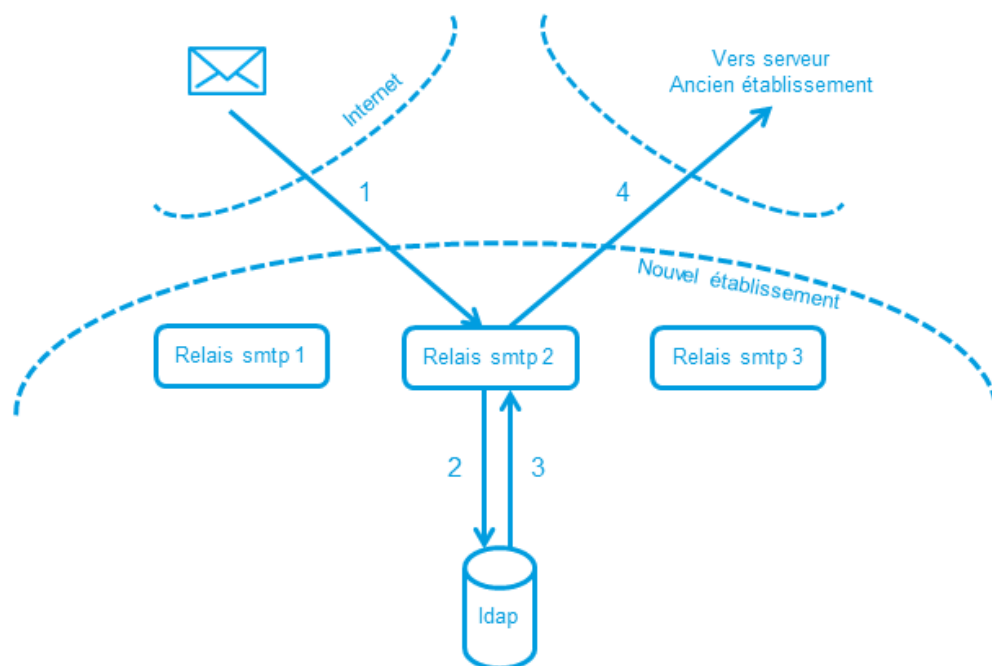


Figure 7 - Exemple de diagramme de flux smtp

Par exemple, pour un mail adressé à `nom.prenom@u-bordeaux2.fr` :

1. Recherche du mx du domaine `u-bordeaux2.fr` et envoi du mail à l'un des nouveaux serveurs relais
2. Requête ldap pour déterminer le serveur à qui acheminer le mail : ici le serveur d'établissement d'origine car le compte n'a pas encore été migré
3. Renvoi du nom du serveur au relais
4. Le relais réachemine le mail

Ainsi, en jouant sur la valeur de l'attribut ldap d'acheminement, nous avons commencé à nous donner les moyens de migrer les comptes un à un vers la nouvelle infrastructure.

Mais il nous fallait traiter également les flux imap et pop. Dans la section suivante, nous allons voir comment nous avons procédé.

3.2 Mise en place de proxys pop et imap

Peu après la mise en place de relais de courrier entrant, nous avons travaillé sur la partie pop et imap.

L'idée était la même que pour le courrier entrant, à savoir permettre aux utilisateurs de récupérer leur mail de manière transparente, et nous donner les moyens de basculer les comptes un à un vers la nouvelle infrastructure.

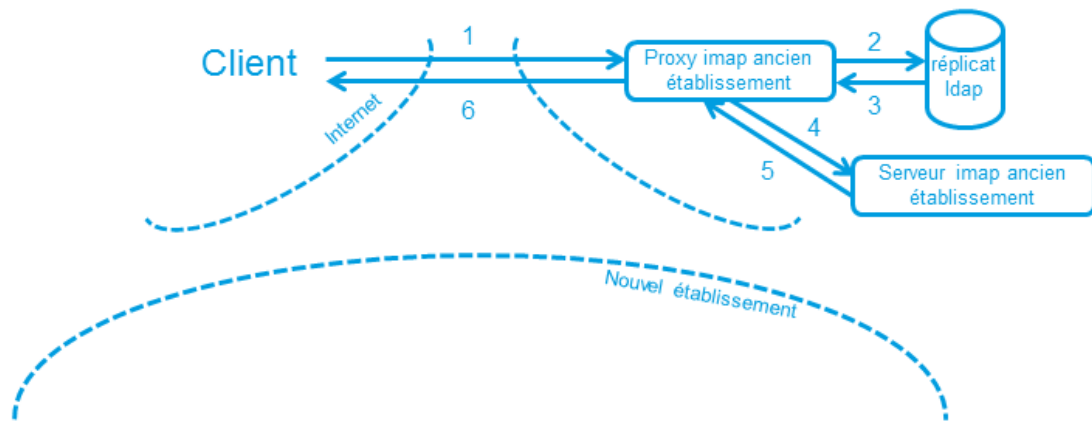


Figure 8 - Exemple de diagramme de flux imap

Par exemple, lors de la récupération des mails par un client :

1. Connexion au proxy imap de l'établissement concerné
2. Requête ldap pour déterminer quel serveur imap solliciter pour récupérer les informations : ici le serveur d'établissement d'origine
3. Renvoi du nom du serveur au proxy
4. Le proxy contacte le serveur imap
5. Puis il reçoit les données
6. Et les achemine vers le client

3.3 Photo finale

Au final, l'architecture complète peut être représentée ainsi :

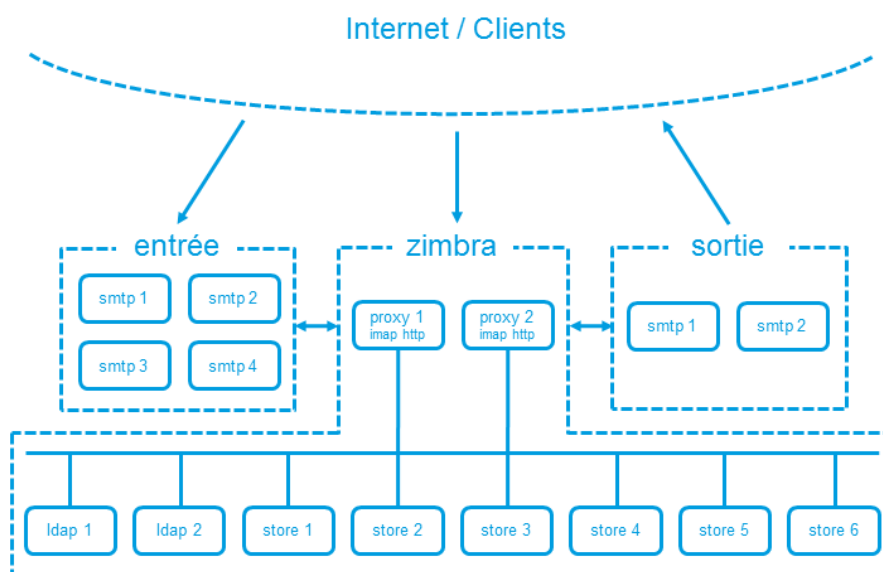


Figure 9 - Architecture complète

Nous disposions alors d'un écosystème nous permettant de traiter « au fil de l'eau » les comptes des anciens établissements.

Voyons comment nous avons procédé pour migrer les comptes.

3.4 Migration effective des comptes

Afin de migrer les comptes, nous avons travaillé selon deux modes opératoires :

1. Par lots, à déclenchement manuel
et
2. En continu, à déclenchement automatique

3.4.1 Migration par lots

Cette méthode a été la première que nous ayons mise en place.

Le principe était le suivant :

- Création d'un fichier csv comportant le login des comptes à créer, leurs adresses mail dans le nouvel établissement, leurs logins dans l'ancien établissement
- Exécution programmée d'un script de création de comptes et d'import des données issues de l'ancien établissement, alimenté par ce fichier csv

Le script de création était composé d'appels aux commandes d'administration *Zimbra*, comme le montre cet exemple :

```
$ zmprov ca john.smith@u-bordeaux.fr "" zimbraCosId 3456abcd-efd2-4e74-bc63-20ef64ab284c zimbraMailHost "v-zimmbx02.srv.u-bordeaux.fr"

$ zmprov ma john.smith@u-bordeaux.fr displayName "John SMITH" cn "John" sn "SMITH" zimbraHideInGal TRUE zimbraAuthLdapExternalDn "uid=jsmith,ou=people,dc=u-bordeaux,dc=fr" zimbraForeignPrincipal "jsmith"
```

Notes :

- *zimbraCosId* = une politique de quota et de fonctionnalités à appliquer à un compte
- *zimbraMailHost* = l'un des 6 serveurs stockant les mails et comptes utilisateur
- *zimbraHideInGal* = afficher ou non le compte dans le carnet d'adresses global
- *zimbraAuthLdapExternalDn* = pour une authentification externe
- *zimbraForeignPrincipal* = lien avec le login dans notre ldap

Le fichier csv, créé à l'initiative de la direction à travers une interface dédiée, leur permettait d'ordonner et de respecter un calendrier précis.

La création des comptes et la migration des données étaient programmées chaque soir, manuellement, sur un serveur dédié à cette tâche, et opérant grâce au logiciel *ImapSync*. Ce dernier était paramétré pour effectuer les opérations de collecte des données imap de l'établissement d'origine vers le nouvel établissement.

Ces actions :

- Ont concerné la majorité du personnel
- Ciblaient des populations spécifiques : service RH, comptabilité, scolarité...
- Etaient programmées sur un calendrier rigoureux
- Etaient déclenchées manuellement
- Ont été menées sur le premier semestre 2014

Mais le gros du travail, à savoir la partie concernant les étudiants, ne pouvait se faire de manière manuelle.

De même, il nous fallait une solution pour les demandes continues de création de comptes.

3.4.2 Migrations au fil de l'eau

Au mois de juin 2014 il nous a fallu réfléchir à la mise en place de procédures de création automatisées pour répondre à deux objectifs : migration massive de l'ensemble des comptes étudiants et créations automatiques de nouveaux comptes étudiants ou personnels dès leur activation.

Nous nous sommes alors rapprochés de l'équipe en charge des identifications numériques afin de travailler de manière coordonnée.

Le principe retenu était le suivant :

1. A l'issue de l'exécution de plusieurs « connecteurs » interfacés avec notre SI (Apogée, Mangue...), certains attributs ldap sont positionnés. C'est le travail de l'équipe en charge des identifications numériques, une des composantes de notre DSI
2. A fréquence régulière, exécution d'un script qui recherche les nouveaux mouvements dans ldap et inscrit des actions à effectuer dans un système de bus d'entreprise (ESB), en l'occurrence *RabbitMQ*
3. Déclenchement des actions par un démon en permanence à l'écoute de ce bus *RabbitMQ* : créations de comptes Zimbra, récupération des mails sur les serveurs des anciens établissements...

Au travers de schémas de principe, nous allons détailler le cheminement exact des opérations.

3.4.2.1 Positionnement d'attributs ldap

En fonction de certains critères déterminés par l'équipe en charge des identifications numériques, plusieurs attributs ldap sont positionnés :

- *ubxZimbraAccount* monovalué positionné au choix à :
 - *createpersonnel*, *createetudiant* ou *createvacataire* pour demander la création d'un compte de type personnel, étudiant ou vacataire
 - *createandmigratepersonnel*, *createandmigrateetudiant* ou *createandmigratevacataire* pour demander la création puis la migration des données d'un compte de type personnel, étudiant ou vacataire
- *ubxSMTPMailhost* = le relais smtp par lequel acheminer le courrier pour ce compte
- *ubxOldCyrusMailhost* = le serveur *cyrus* dans l'ancien établissement
- *ubxOldUid* = le login de la personne dans l'ancien établissement

3.4.2.2 Exécution des actions à effectuer

La première étape consiste à chercher les entrées devant être traitées.

Pour ce faire, un script écrit en langage python est lancé tous les quarts d'heure à la recherche de l'attribut `ubxZimbraAccount` présent et positionné.

Par exemple, les deux entrées `uid=jdoe` et `uid=jsmith` sont sélectionnées car elles disposent d'un attribut `ubxZimbraAccount` positionné respectivement à `createandmigrateetudiant` et `createandmigratepersonnel` :

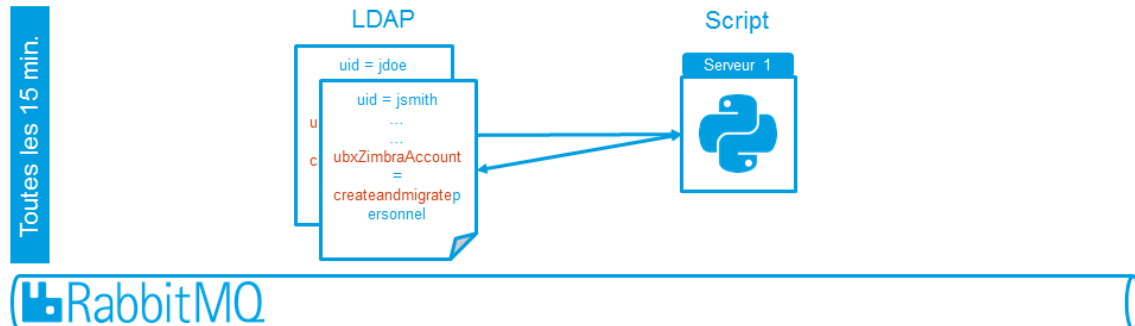


Figure 10 - Collecte des entrées à traiter

Ensuite, ce script va ajouter deux messages dans une queue dédiée de *RabbitMQ* :

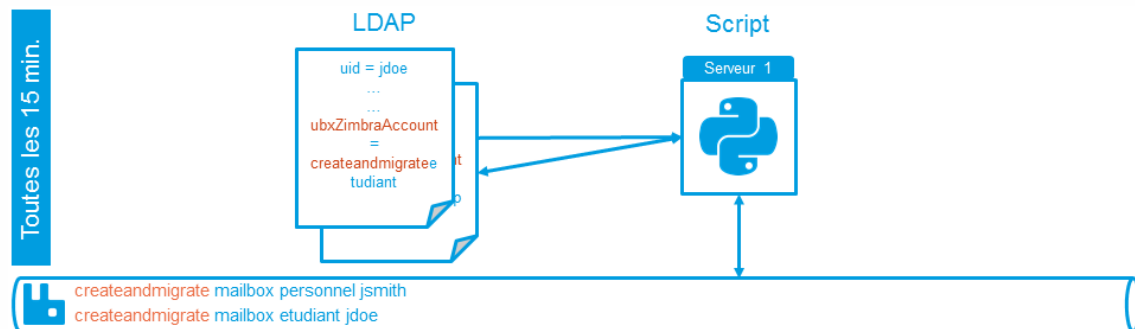


Figure 11 - Ajout de deux messages

Décortiquons ces messages :

- `createandmigrate` sera le verbe correspondant à l'action à mener
- `mailbox personnel` ou `mailbox etudiant` sera la cible de l'action
- `jsmith` ou `jdoe` sont les identifiants à traiter

En parallèle, un démon également écrit en python est en écoute permanente du bus *RabbitMQ*.

Dès qu'un message est présent dans la queue dédiée, il est immédiatement exécuté.

Ainsi, la création des comptes *Zimbra* est déclenchée (au moyen d'appels *soap*) :

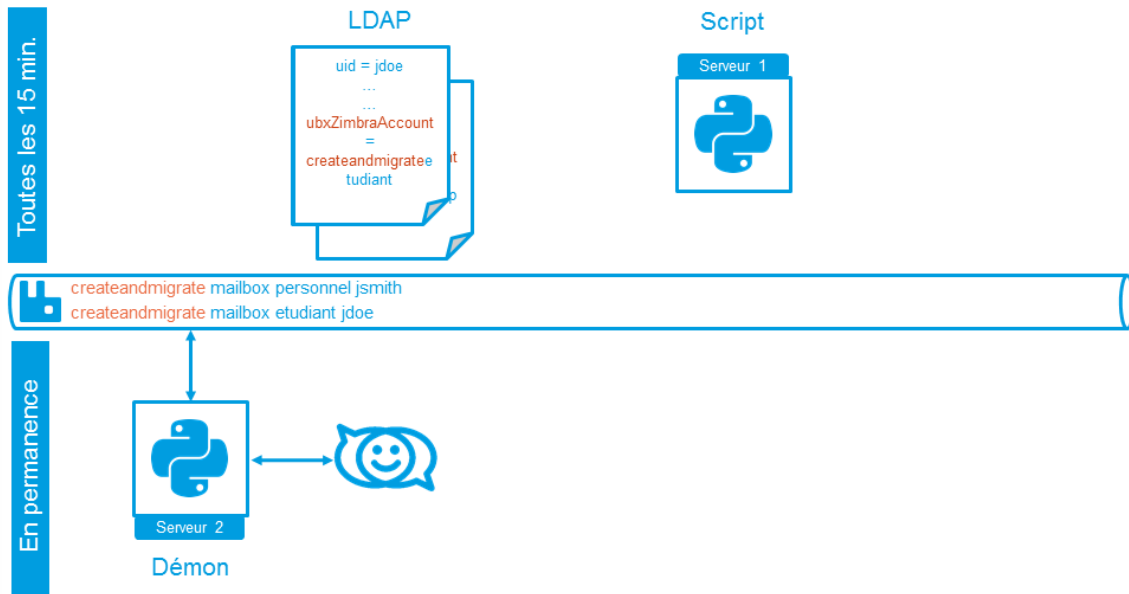


Figure 12 - Création des comptes Zimbra

Une fois ces comptes *Zimbra* créés, en cas de succès, deux nouveaux messages sont instanciés dans *RabbitMQ*, cette fois-ci à l'initiative du démon. Ils concernent la nécessité de récupérer les mails dans les anciens établissements :

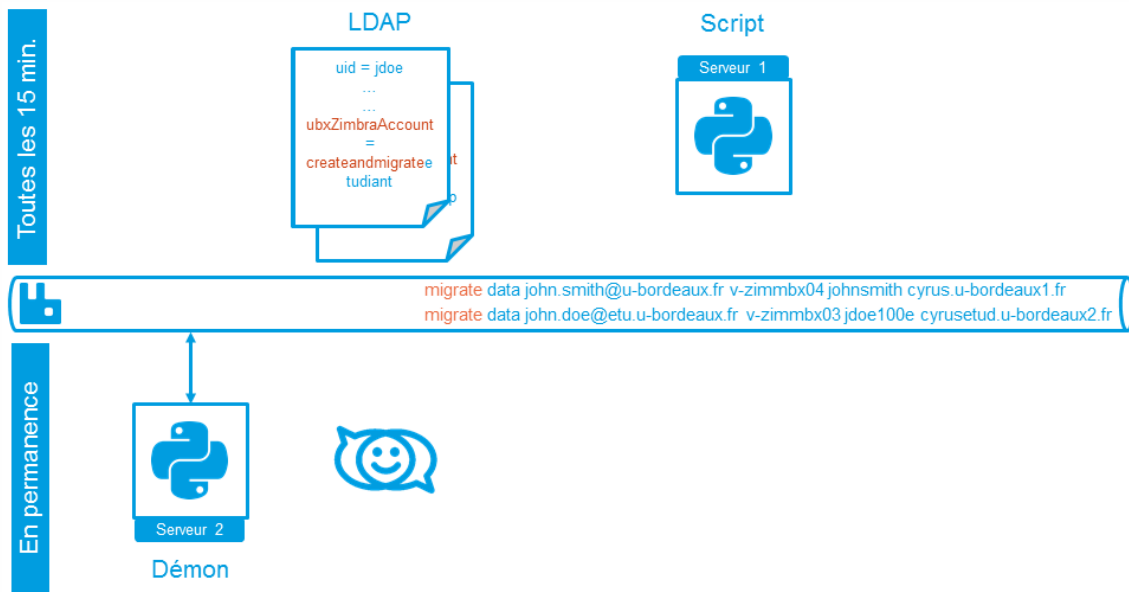


Figure 13 - Création de messages de migration

Décortiquons l'un de ces messages :

- *migrate data* sera le groupe verbal correspondant à l'action à mener

- `john.smith@u-bordeaux.fr` sera le compte à traiter dans la nouvelle infrastructure *Zimbra*
- `v-zimmbx04` correspond au serveur *Zimbra* détenant le compte, dont l'information a été récupérée au préalable par un appel *soap*
- `johnsmith` est l'identifiant dans l'ancien établissement où récupérer les mails. Cette information a été positionnée au préalable dans l'attribut `ubxOldUid` et de ce fait disponible
- `cyrus.u-bordeaux1.fr` est le serveur détenant les mails du compte dans l'ancien établissement. Cette information a été positionnée au préalable dans l'attribut `ubxOldCyrusMailhost` et de ce fait disponible

Fort de ces informations, un autre démon, positionné sur un serveur de migration dédié, peut consommer les messages de demande de récupération et lancer en parallèle le rapatriement des données :

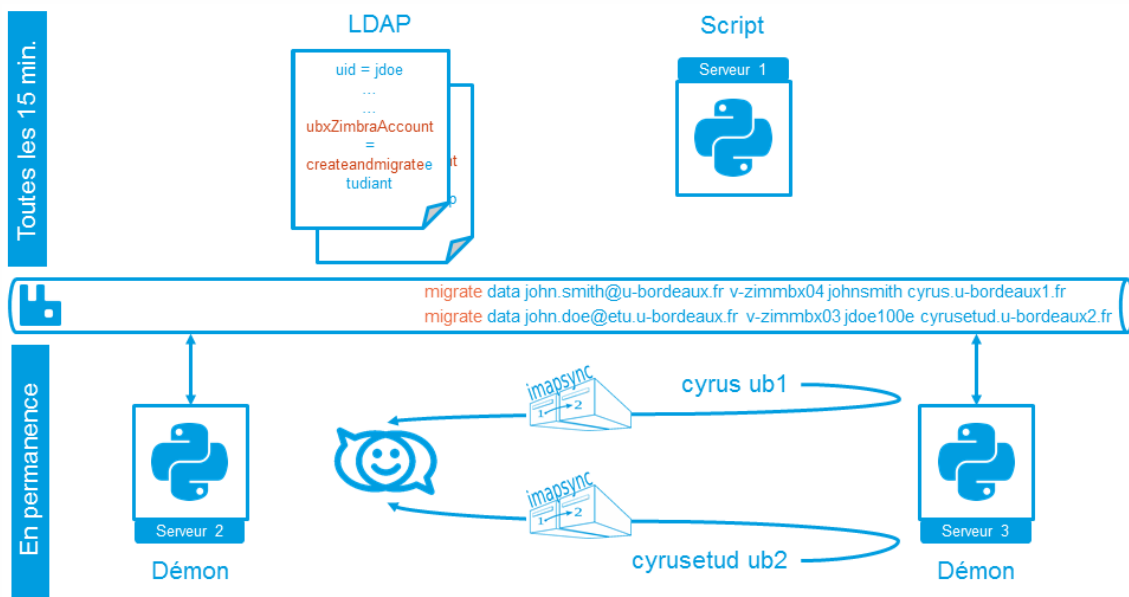


Figure 14 - Migration des données

Notes :

- les données sont rapatriées grâce au logiciel *imapsync*. Il a été testé et paramétré pour récupérer les informations sur les anciens serveurs d'établissement
- 16 flux maximum sont lancés en parallèle

Une fois les migrations terminées :

- l'attribut `ubxZimbraAccount` est supprimé
- l'attribut `ubxSMTPMailhost` est positionné vers le serveur d'infrastructure *Zimbra* qui contient la boîte
- le message de demande de migration est supprimé de la queue *RabbitMQ*

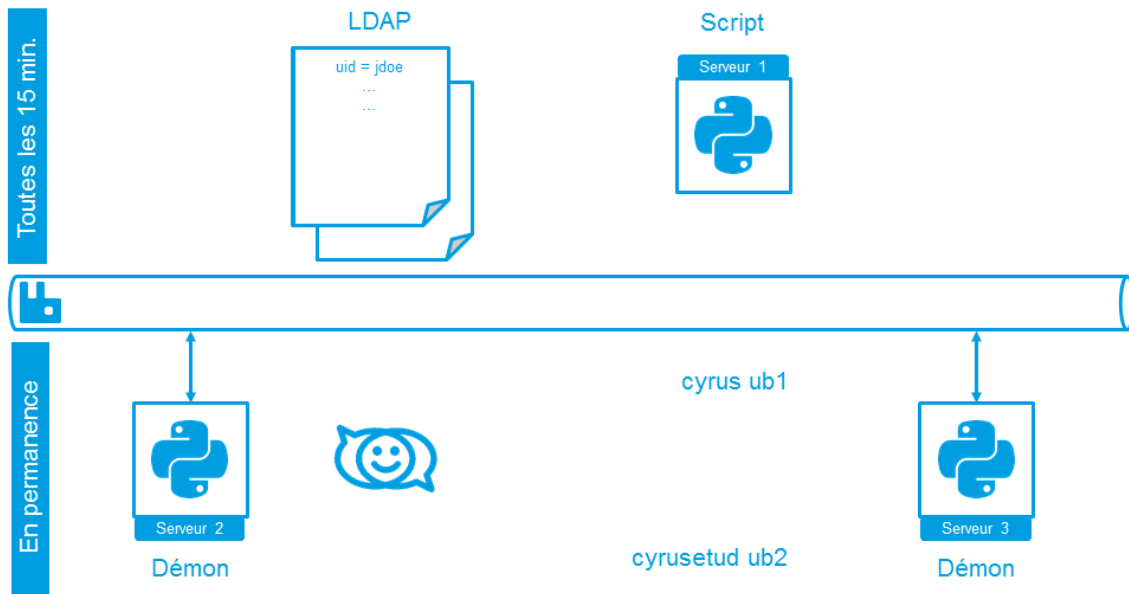


Figure 15 - Nettoyage des informations

4 Reporting

Après avoir découvert comment nous avons procédé pour la migration des comptes, voyons maintenant de quels outils nous nous sommes dotés afin de suivre ces phases, mais également ceux destinés à déclencher au besoin certaines actions.

4.1 Django, pierre angulaire du système consignment d'événements et fournisseur d'interfaces de suivi / de saisie

De même que *RabbitMQ* est le socle commun des procédures de migration et création de comptes, le framework *Django* est celui du reporting.

Couplé à *MySQL*, ce dernier atteint un double objectif :

1. Enregistrer dans une base de données les actions menées lors des phases de migration et de création de compte
2. Proposer des interfaces de pilotage

Ainsi, une « application » *Django* a été créée spécifiquement et couplée aux démons de création et migration.

4.1.1 Consignment d'événements

A chaque création, l'ensemble des étapes est consignée en base :

- Récupération des informations ldap
- Résultat de l'action de création du compte *Zimbra*
- Résultat de l'action de création de l'alias de messagerie
- Consignment des étapes de récupération et d'envoi de messages à *RabbitMQ*

Voici par exemple le code python de la méthode sollicitée pour enregistrer le suivi d'une action :

```
def myLogSuivi(priority, uuid, texte, action_a_suivre,
ordre_suivi):

    try:
        # Le log normal
        myLog(priority, uuid + " " + texte)

        # Un enregistrement django pour un suivi de l'action
        if action_a_suivre is not None:
            suivi = SuiviAction(action=action_a_suivre,
libelle=texte)
            suivi.ordre = ordre_suivi
            suivi.save()

    except Exception as e:
        myLog(syslog.LOG_ERR, "erreur au log du suivi pour
\"{0}\"\".format(uuid + " " + texte))
        myLog(syslog.LOG_ERR, str(e))
```

Le bloc de sollicitation de *Django* qui a été mis en gras est très simple :

- Instanciation d'un objet de suivi d'action couplé à une action à suivre, complété par un texte explicatif
- Enregistrement d'un ordre pour ce suivi. Cet ordre permettra au reporting une traçabilité plus fine
- Enregistrement de cet objet en base : la simple directive *suivi.save()* déclenche le *create* au niveau base de données

4.1.2 Fournisseur d'interfaces de suivi / de saisie

Comme indiqué précédemment, *Django* a également été utilisé pour créer deux sites web :

- L'un dédié au reporting
- L'autre dédié aux déclenchements d'actions unitaires
 - de création de comptes spécifiques telles les boîtes métier par exemple
 - de lancement de création et migration de données d'un compte en particulier

4.1.2.1 Site dédié au reporting

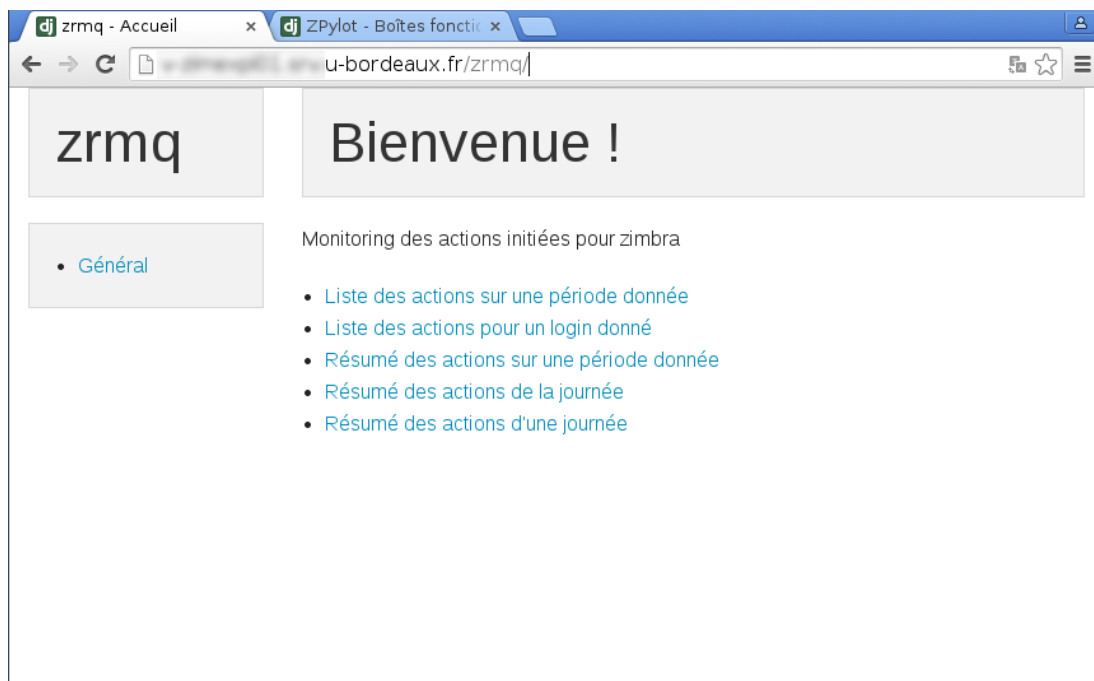


Figure 16 - Site dédié au reporting

Ce site propose d'accéder à un ensemble de rapports en fonction de critères de période, d'identité...

4.1.2.2 Site dédié aux déclenchements d'actions unitaires

A la différence du site précédent, celui-ci permet de déclencher diverses opérations comme la création de comptes, de boîtes fonctionnelles, de listes de distribution...

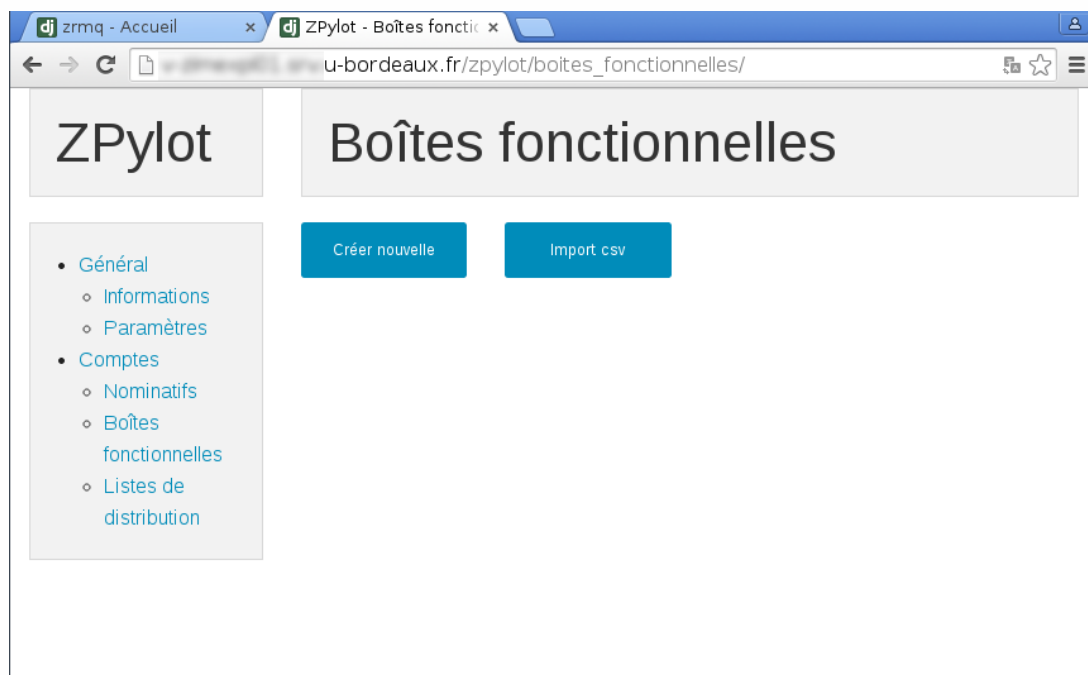


Figure 17 - Site dédié au déclenchement d'actions unitaires

A noter que pour déclencher les actions, le site « ZPylot » s'appuie sur la mécanique *RabbitMQ*.

Par exemple, les demandes de création de boîtes fonctionnelles sont lancées en générant un message *RabbitMQ*.

Outre un développement très rapide, cela permet une réutilisation de code et de bénéficier d'un point unique d'améliorations / débogage.

4.2 Birt en appui pour des tableaux de bord synthétiques

Afin de disposer rapidement d'indicateurs sur les volumétries, les résultats et les suivis d'actions, nous avons porté notre choix sur l'outil de reporting Birt.

S'appuyant sur les données issues de la base *MySQL* créée pour *Django*, *Birt* met à disposition un outillage permettant de générer des rapports dynamiques, exploitables en mode web.

Par exemple, un état récapitulatif des actions menées la veille est disponible et émis chaque jour par mail pour un suivi régulier :

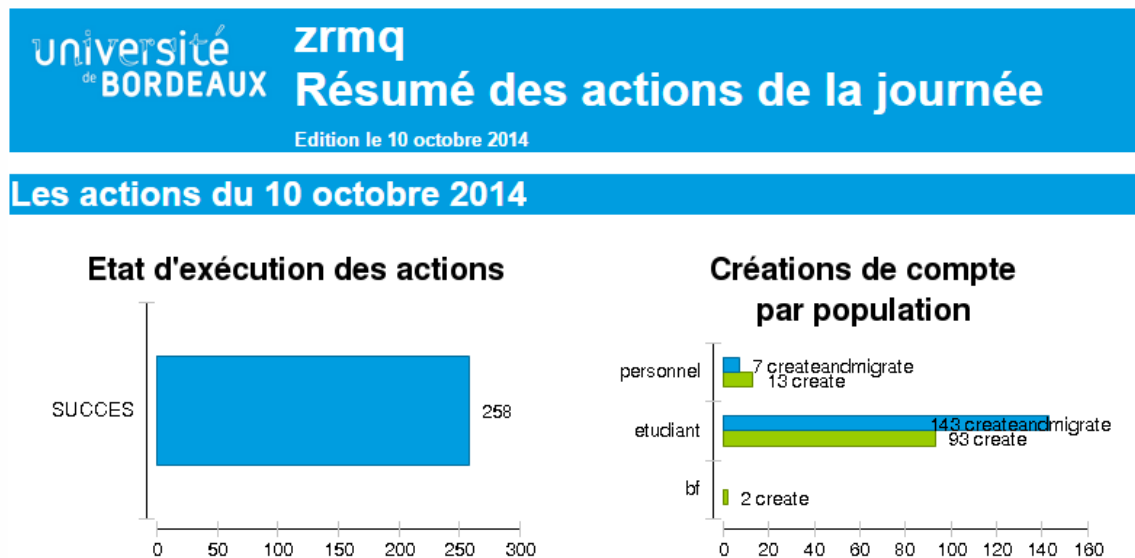


Figure 18 - Résumé des actions de la journée

De même, il est possible de suivre les évolutions des créations sur une période donnée, ici le mois :

Les créations de comptes sur le mois précédent

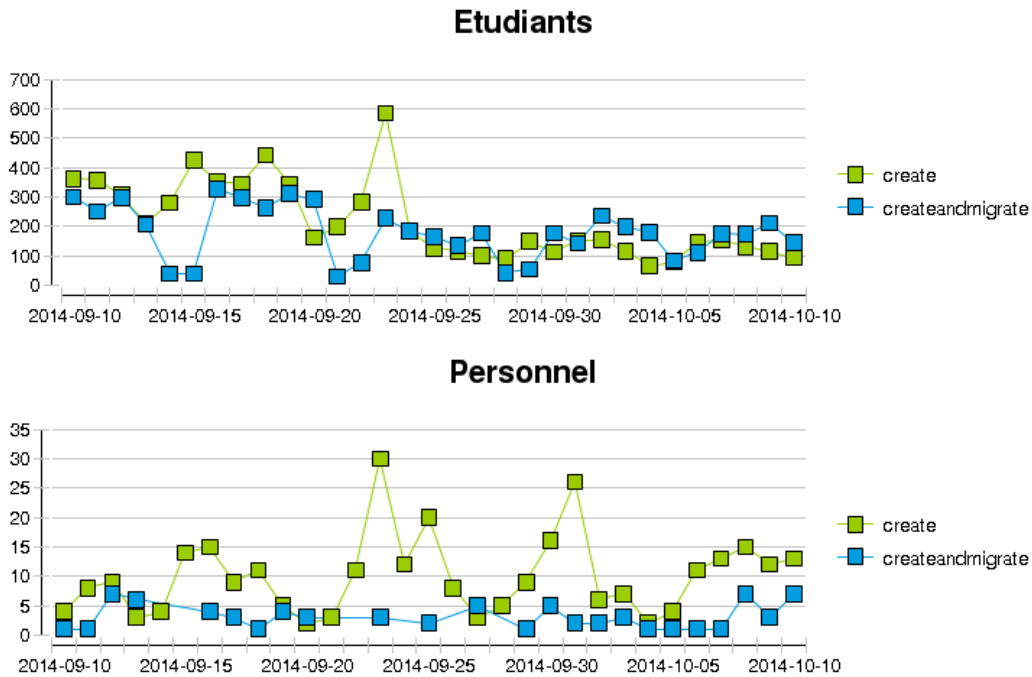
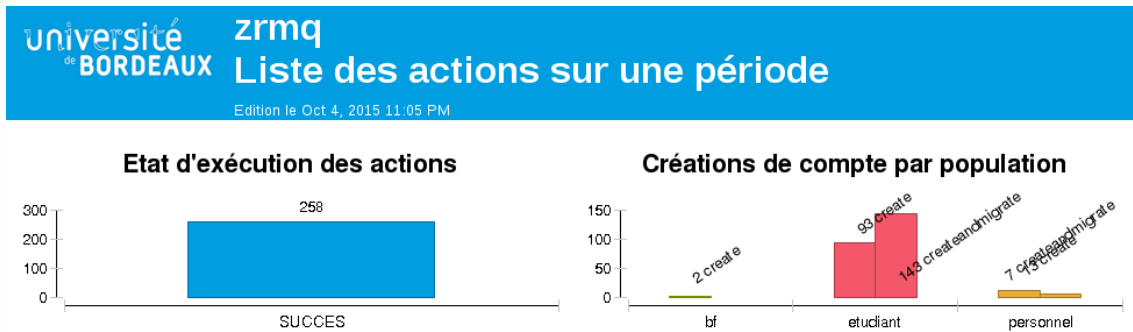


Figure 19 - Résumé des actions sur un mois

Mais il est également possible de disposer d'informations plus précises, notamment une liste des actions, toujours sur une période donnée :



Les actions (5000 maxi) du 2014-10-10 au 2014-10-10

SUCCES		
Date	Login	Libellé de l'action
Oct 10, 2014 11:00 PM	dguenova	2014-10-10 23:00:01 2e65cc19-99b6-40b0-8690-842c92155988 create mailbox etudiant dguenova
Oct 10, 2014 10:15 PM	lmartins	2014-10-10 22:15:02 6501bd72-76e9-42a7-b240-7b1f176567c6 create mailbox etudiant lmartins
Oct 10, 2014 10:00 PM	anegre001	2014-10-10 22:00:01 c389b2c5-5550-4276-b623-5d4ea6eaba78 create mailbox etudiant anegre001
Oct 10, 2014 9:15 PM	zraphon	2014-10-10 21:15:01 9cf69f43-5e33-4775-8016-973f05b3fa10 create mailbox etudiant zraphon
Oct 10, 2014 8:45 PM	mleuch	2014-10-10 20:45:02 dcf52fa3-c7de-4159-89b5-600e4a3d4b77 create mailbox etudiant mleuch
Oct 10, 2014 8:15 PM	mbouscallou	2014-10-10 20:15:01 eef9dd17-14ba-4f3d-9acc-94d3d5e75c48 create mailbox etudiant mbouscallou
Oct 10, 2014 7:30 PM	afleul	2014-10-10 19:30:01 d63ff3ba-92a8-45fc-bdf7-e8da2cb770ee create mailbox etudiant afleul
Oct 10, 2014 7:30 PM	lgeorget	2014-10-10 19:30:01 b105b96c-13ea-48a4-ba2f-27fb42882c5f create mailbox etudiant lgeorget
Oct 10, 2014 7:15 PM	robert19	2014-10-10 19:15:01 7a3da7a1-de61-499c-90ab-1c7756ab527c createandmigrate mailbox etudiant robert19
Oct 10, 2014 7:15 PM	gndtouris	2014-10-10 19:15:01 5948c437-57a1-4d40-b2b4-f262f2075ead create mailbox etudiant gndtouris
Oct 10, 2014 7:00 PM	newiczak	2014-10-10 19:00:01 68162c7f-f033-4bfd-9b41-1f95245e7fec create mailbox etudiant newiczak
Oct 10, 2014 6:45 PM	lpeny	2014-10-10 18:45:01 9d3a1508-c0f8-46d7-8460-fcb3e3d3b074 create mailbox etudiant lpeny
Oct 10, 2014 6:30 PM	smasin	2014-10-10 18:30:01 b3e1e6f1-e40e-4c44-9a78-e2e42e9133de create mailbox etudiant smasin

Figure 20 - Liste des actions sur une période

Enfin, en cliquant sur une action il est possible d'avoir un détail complet de son déroulement :

universit  de BORDEAUX zrmq
D tail d'une action
Edition le 31 juil. 2015 12:20

Informations sur l'action

Date de la demande 10 oct. 2014 19:15
R sultat d'ex cution **SUCCESS**
Login [redacted]
Libell  2014-10-10 19:15:01 7a3da7a1-de61-499c-90ab-1c7756ab527c createandmigrate mailbox etudiant [redacted]

Suivis de l'action

Ordre	Libell�
1	r�cup�ration des infos ldap du compte [redacted]
2	cr�ation du compte zimbra [redacted]@etu.u-bordeaux.fr
3	compte zimbra [redacted]@etu.u-bordeaux.fr cr�� avec succ�s
4	cr�ation de l'alias [redacted]@etu.u-bordeaux.fr pour [redacted]@etu.u-bordeaux.fr
5	alias zimbra [redacted]@etu.u-bordeaux.fr pour [redacted]@etu.u-bordeaux.fr cr�� avec succ�s
6	cr�ation de l'alias [redacted]@u-bordeaux.fr pour [redacted]@etu.u-bordeaux.fr
7	alias zimbra [redacted]@u-bordeaux.fr pour [redacted]@etu.u-bordeaux.fr cr�� avec succ�s
8	envoi du message de migration pour [redacted]
9	modification des donn�es ldap de l'entr�e [redacted]
10	informations ldap modifi�es avec succ�s pour l'entr�e [redacted]

zrmq - D tail d'une action
Bureau Ing nierie et Middleware 31 juil. 2015 12:20 1 / 1

Figure 21 - D tail d'une action

Ainsi, nous voyons le d roulement exact d'une action de cr ation de compte avec migration des donn es, et notamment l'ordre des sous-actions dont nous avons parl  dans la section « Django ».

Passons maintenant au bilan de ce long projet.

5 Bilan

5.1 Bilan humain

L' nergie d ploy e pendant la phase de mise en place a monopolis  l' quivalent de 4 IGE   temps plein pendant plusieurs semaines.

Puis, d s le d but du d ploiement, un effort particulier a  t  plac  sur l'accompagnement au changement. Il nous semblait essentiel de pr senter le produit et d'expliquer les raisons de notre d marche. Programm s par l' quipe de Direction, des ateliers en amph   taient dispens s quelques jours avant tout d ploiement au sein d'un coll ge, d'un d partement ou d'une  quipe. Nous saluons au passage le soutien de la Direction qui a su approcher les structures avec toute l'habilet  et la pugnacit  n cessaires.

Actuellement en mode de « croisi re », le service de messagerie occupe l' quivalent de 2 ETP pour :

- Des  volutions mat rielles (stockage, serveurs) et logicielles (os, socle zimbra)
- Une assistance au travers de la plateforme GLPI (2250 tickets / an) et parfois au t l phone
- La mise en place et la mise   jour de la documentation

5.2 Bilan technique

La plateforme matérielle et logicielle, parfaitement calibrée, tient la charge des 78000 comptes qu'elle gère.

Côté développements, 2500 lignes de code ont été produites par l'équipe en charge de ce projet, sans compter la participation de l'équipe « identité numérique » sur la partie ldap.

5.3 Bilan financier

Les budgets prévisionnels ont été respectés sans dépassement.

Le prix de revient annuel d'une boîte n'excède pas quelques euros.

La répartition budgétaire quant à elle est la suivante :

COÛT DE LA SOLUTION

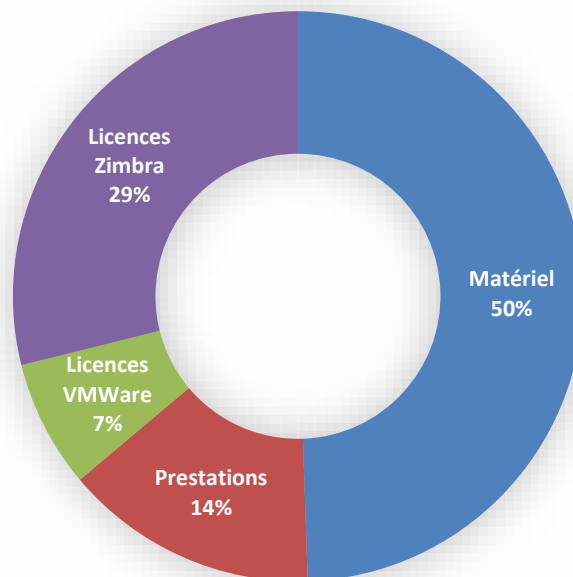


Figure 22 - Répartition des coûts

5.4 Les points positifs, les points négatifs

Voyons à présents ce que nous avons retenu de cette expérience.

5.4.1 Les plus

En six points ce que nous retenons de positif :

- Sur les 15 étapes projet identifiées, toutes ont été réalisées, les jalons essentiels tenus
- L'architecture, revue par nos soins, est performante et monte en charge sans difficulté
- Il n'y a pas eu de rejet du produit

- La conduite du changement a été appréciée
- Les laboratoires viennent vers nous !
- La mise en place d'une architecture innovante pour créer les comptes en flux continu

5.4.2 Les moins

De même, détaillons maintenant les points négatifs :

- Le calendrier imposé a été très contraint puisque nous avons eu le feu vert en octobre 2013 pour une mise en exploitation au 1^{er} janvier 2014, et ce sur les adresses 100% en « @u-bordeaux.fr »,
- Nous avons dû gérer la collision d'un projet de migration avec un projet de fusion. Il est plus courant de s'occuper de l'un puis de l'autre,
- La difficulté technique était très importante, sollicitant des compétences diverses et pointues : *postfix, Zimbra, exchange, cyrus-imap, perdition, ldap, dns, nginx, ha-proxy, soap, tuning java, stockage equallogic, vmware* (dont *srn*), *logstash/elastic Search/kibana, python...*
- Pour ajouter à la difficulté, aucune phase de test n'a pu avoir lieu et nous avons dû démarrer directement sur les migrations de comptes. Cerise sur le gâteau, nous avons commencé par les VIP, et ce malgré nos alertes !
- Les difficultés à consolider les entrées ldap par nos correspondants numériques de structure ont parfois ralenti le rythme. Nous avons également eu des difficultés de planification sur l'un des campus, réticent au changement,
- Un changement de chef de projet intégrateur ayant eu lieu juste après la réunion lancement n'a pas facilité la phase d'allumage. D'autre part, le positionnement constructeur / intégrateur n'aidait pas,
- Le PRA mal conçu induit un stockage coûteux (facteur x10),
- Nous avons subi un bug majeur sur l'infrastructure de stockage, stoppant toute activité pendant un week-end entier,
- De nombreux bugs *Zimbra*, l'université de Bordeaux jouant presque le rôle de bêta-testeur sur certaines fonctions essentielles non testées,
- Nous ne disposons pas de processus de cycle de vie des comptes, ce qui amène à un épuisement des licences programmé,
- Pas d'autonomie sur la mise en place de la documentation : le format imposé – le cms institutionnel – est peu pratique pour la saisie, contraignant en terme d'espace disponible et l'information est difficilement accessible aux utilisateurs.

5.5 Et après ?

Il faudra réfléchir à une automatisation de l'attribution d'adresses de sous-domaines, à la gestion des habilitations et au cycle de vie des comptes. De même, il faudra définir un engagement de service.

Le stockage quant à lui arrivera à saturation mi-2016, essentiellement dû à des quotas utilisateur très (trop?) généreux. Une étude sur la mise en place d'une solution moins coûteuse est en cours.

Enfin, *Zimbra* version 9 impliquera une refonte forte de l'architecture, s'appuyant très probablement sur un stockage distribué. En effet, afin d'assurer un maximum de disponibilité, un stockage objet sur des solutions comme *Scality* (ou *ceph* ?) seront préconisées. Solutions qui ne pourront pas s'appuyer sur l'existant, nécessitant de nouveaux investissements et l'acquisition de nouvelles compétences.