

Nancy-Université



Mise en place de différentes solutions au sein d'une société
de services en logiciels libres :
VersusMind

RAPPORT DE STAGE

Benjamin

Sommaire

Remerciements.....	4
Présentation de la société VersusMind.....	6
1. Histoire de la société.....	6
2. Présentation des principales solutions proposées par VersusMind.....	6
a. OpenOffice.org.....	6
b. eGroupWare.....	6
c. Zimbra.....	7
d. Développements de VersusMind.....	7
3. Topologie réseau de VersusMind.....	8
4. Mon cadre de travail.....	8
5. Une société de services, de nombreuses missions	9
Mise en place d'un site de e-commerce : Prestashop.....	11
1. Objectif.....	11
2. Différentes solutions.....	11
a. Tableau récapitulatif.....	11
b. Conclusion.....	12
3. Présentation de PrestaShop.....	12
4. Mise en place.....	14
5. Problèmes rencontrés.....	14
Une plate-forme de vidéoconférence avec Dimdim.....	15
1. Objectif	15
2. Présentation de Dimdim.....	15
a. Fonctionnalités et tarifs	15
b. Comment fonctionne Dimdim.....	16
3. Problèmes pour la mise en place.....	17
4. Mise en place.....	17
5. Conclusion.....	17
Mise en place d'un système d'authentification unique grâce à CAS.....	18
1. Objectif	18
2. Présentation du SSO (Single Sign On).....	18
3. Présentation de CAS.....	18
4. Mise en place de CAS.....	19
5. CASifier une application.....	20
a. Exemple simple d'authentification via phpCAS.....	20
b. Illustration de l'exemple.....	21

c. CASification d'eGroupWare :.....	22
6. Problèmes rencontrés.....	23
Mise en place d'une solution de supervision réseau : Zabbix.....	24
1. Objectif :.....	24
2. Présentation de Zabbix :.....	24
a. Ce que Zabbix peut surveiller :.....	24
b. Comment Zabbix peut-il surveiller :.....	24
3. Mise en place de Zabbix	25
a. Topologie à superviser	25
b. Installation du serveur	25
c. Installation des clients	26
4. SNMP	26
a. Présentation	26
b. Intégration à Zabbix	27
5. Difficultés rencontrées.....	28
a. SNMP.....	28
b. Agent Mac OS.....	28
c. Agent en mode actif	28
Conclusion.....	30
1. Aspects techniques.....	30
2. Aspects relationnels.....	30
3. Aspects personnels.....	31
4. Conclusion générale.....	31
Annexes.....	32
1. Tutoriel créé pour l'installation de Dimdim en français.....	32
a. Prérequis.....	32
b. Téléchargement et installation de Dimdim.....	34
c. Modifications et derniers réglages.....	35
2. Mise en place de CAS	37

Remerciements

D'abord et avant tout, je tiens à remercier les personnes qui m'ont assisté, m'ont aidé et ont répondu à mes nombreuses questions.

A ce titre, je remercie donc Benoît, le dirigeant de VersusMind, qui m'a fait confiance et qui a eu la gentillesse de m'accepter dans son entreprise, ainsi que Julien, Pierre et Adrien qui ont pris beaucoup de leur temps pour m'aider et répondre à mes nombreuses questions.

Merci pour votre patience et votre soutien.

Je tiens également à remercier Madame Nadia B. qui a accepté d'être ma tutrice de stage.

Introduction



Présentation de la société VersusMind

1. Histoire de la société

VersusMind est une entreprise créée par Benoît K. en avril 2006. Cette entreprise est une société de services en Logiciels Libres (SSLL). Son but est d'accompagner les organisations, entreprises ou administrations dans leur migration vers des solutions Open Source.

Localisée à Nancy, dans l'Est de la France, VersusMind regroupe une équipe de cinq employés aux connaissances pointues dans le domaine du Logiciel Libre et dans la conduite du changement des organisations.

2. Présentation des principales solutions proposées par VersusMind

a. OpenOffice.org

OpenOffice est une suite bureautique libre fondée sur le code source de la version 5.2 de StarOffice, rendue publique par son acquéreur, Sun Microsystems, en juin 2000. Ce projet est disponible sous plusieurs licences (la LGPL et, jusqu'à la version 2.0beta2, la SISSL), et fonctionne sur plusieurs plates-formes dont Microsoft Windows et de nombreux Unix : Linux, Solaris, ou Apple Mac OS X. Elle se compose de plusieurs logiciels bureautique :

- OpenOffice Writer (traitement de texte)
- OpenOffice Calc (tableur)
- OpenOffice Base (gestion de base de données)
- OpenOffice Impress (créateur de présentations multimédia)
- OpenOffice Draw (outil de dessin vectoriel)

OpenOffice est historiquement la première solution Open Source proposée par VersusMind.

b. eGroupWare

eGroupWare est un logiciel de travail collaboratif professionnel Open Source créé à partir de phpgroupware en 2000. Il fait partie du projet GNU et est publié sous licence GPL. La communauté de développeurs s'étend sur 68 pays. Le logiciel est par exemple très répandu au Brésil où de nombreuses administrations publiques ou entreprises l'utilisent sur des réseaux pouvant dépasser facilement les dix mille utilisateurs.

Il permet de gérer les contacts, rendez-vous, tâches et bien plus, pour toute une entreprise :

- gérer ses projets en mode collaboratif (gestion des contraintes, diagrammes de GANTT, gestion de la facturation, etc)
- partager une base de connaissance
- gérer la communication interne et externe de l'entreprise
- partager des ressources (véhicules, salles de réunion, etc)
- gérer ses documents

eGroupWare est conçu nativement avec une interface Web, ce qui permet d'accéder aux données de l'entreprise à partir de n'importe quelle plate-forme connectée à Internet.

c. Zimbra

Zimbra est également un logiciel serveur collaboratif (ou groupware) qui permet à ses utilisateurs de stocker, organiser et partager rendez-vous, contacts, emails, liens, documents, etc.

Un des nombreux avantages de Zimbra est qu'il installe (et utilise) beaucoup d'applications externes de manière très simple. Par exemple, lors de l'installation de Zimbra, un serveur mail Postfix est automatiquement installé et configuré. Dès lors, pour y accéder, on utilise le client mail inclus à Zimbra (ou un client externe via POP ou IMAP, également installés par Zimbra). De même, il installe et configure un serveur LDAP sans l'intervention de l'utilisateur.

En janvier 2007, l'éditeur de Zimbra affirmait avoir 6 millions d'utilisateurs. Le 17 septembre 2007, Yahoo rachète Zimbra pour 350 millions de dollars.

Depuis, Zimbra est disponible sous deux formes : une version payante et une version Open Source.

Voici quelques grandes fonctionnalités de Zimbra :

- Serveur et client mail (ainsi que POP et IMAP)
- Calendrier
- Contacts partagés
- Calendriers partagés
- Notes partagées
- Gestion du multidomaine
- Synchronisation Outlook, Apple iCal...
- Synchronisation PDA, iPhone, Blackberry

Son interface web extrêmement bien faite le rend très simple d'utilisation et de configuration. Quelques fonctions sont intéressantes comme, par exemple, la gestion du clic droit ou le drag&drop.

d. Développements de VersusMind

Ces différentes solutions étant Open Source, VersusMind développe des modules ou des extensions en vue de les intégrer à ces applications.

Par exemple, VersusMind a publié sous licence GPL v3 quatre modules additionnels pour eGroupWare :

- eNewsMind, un module d'emailing (newsletters, communiqués de presse, invitations, etc)
- ProjectReportMind, un module de comptes-rendus de réunion
- TravelMind, un module de gestion des frais kilométriques
- StatsMind, un module de réalisation de statistiques sur la répartition du temps de travail des employés d'une entreprise.

De plus, un zimlet (plugin pour Zimbra) a également été développé : zmBackMind
zmBackMind permet de gérer vos sauvegardes et restaurations de serveur et d'email.

Les principales fonctionnalités de zmBackMind sont :

- Sauvegarde totale de Zimbra
- Sauvegarde des boites emails de tous les utilisateurs
- Restauration globale de la messagerie Zimbra
- Restauration des messages d'un utilisateur via une interface graphique

3. Topologie réseau de VersusMind

Non divulguable

4. Mon cadre de travail

Je travaillais dans l'open space de VersusMind, un grand bureau partagé avec huit autres personnes. Cela s'est avéré très pratique car, par exemple, nous n'avons pas à nous déplacer pour obtenir un renseignement.

Côté matériel, j'utilisais sur un client léger fonctionnant grâce à un serveur ThinStation. Pour ce client léger, j'utilisais une distribution Kubuntu Dapper Drake 6.06 (version assez ancienne). J'avais également à ma disposition plusieurs machines virtuelles sous Debian Lenny (avec un accès SSH + VNC) sur lesquelles j'avais les droits administrateur et sous plusieurs Windows XP et 2003 Server (grâce à un accès VNC).

5. Une société de services, de nombreuses missions

VersusMind, étant une société de services, propose, comme son nom l'indique, de nombreux services aux entreprises ou administrations. Parmi ces services, nous trouvons notamment :

- Des audits de migrations, qui sont des analyses des besoins d'une organisation, tant au niveau matériel que logiciel. L'audit sert à analyser l'existant et à proposer des solutions de remplacement
- Des migrations logicielles ou/et matérielles, cela consiste à installer la solution choisie par l'entreprise. Dans le cas de VersusMind, une solution Open Source
- La formation des employés
- Le support, la maintenance : en cas de problèmes d'utilisation, les entreprises peuvent faire appel aux professionnels des employés de VersusMind pour les aider
- Du développement : disposant du code source des applications, les développeurs de VersusMind créent des modules ou des ajouts personnalisés au demandeur.

Dans ce contexte, mes tâches ont été très diverses et variées tout au long de mon stage. J'ai eu plusieurs missions à remplir, toutes aussi enrichissantes les unes que les autres.

Je ne pourrais pas lister toutes les activités et missions que j'ai effectuées pendant mon stage, je vais simplement lister les quatre principales, sachant que les autres sont principalement du support aux clients de VersusMind ou de l'aide au sein des employés.

Mes différentes missions



Mise en place d'un site de e-commerce : Prestashop

1. Objectif

VersusMind propose différentes prestations à ses clients. Une prestation importante est le support technique sur les solutions mises en place au sein des entreprises ou administrations. Depuis peu, une prestation supplémentaire est prévue, il s'agit de l'hébergement des applications proposées par VersusMind, également appelé SAAS (Software As A Service).

L'objectif de cette première mission était donc de mettre en place une solution de vente en ligne afin de vendre ces prestations à distance. Ainsi, par exemple, un client pourra acheter une journée de support (soit 7 heures) par l'intermédiaire de ce site de vente.

2. Différentes solutions

Aucun logiciel n'était privilégié par mes responsables, je devais donc faire une étude comparative des solutions existantes et en retirer la plus avantageuse. Pour ce faire, j'ai établi un tableau récapitulatif des avantages et des inconvénients de ces différentes solutions (après les avoir toutes testés).

a. Tableau récapitulatif

	Installation, configuration	Administration	Modules disponibles	Communauté
OSCommerce	Très simple	Complète, bien documentée	Plus de 5000	La plus grosse communauté mondiale
Zencart	Simple	Complète	Environ 800	Assez nombreuse mais peu de français
Ubercart (module Drupal)	Difficile car installation de Drupal + module	Complète, bien documentée	Environ 200	Peu nombreuse et pas française mais assez réactive
VirtueMart (module Joomla)	Difficile car installation de Joomla + module	Complète, bien documentée	Environ 3000 en comptant les traductions	Assez nombreuse, communauté française très réactive
Magento	Difficile, modules supplémentaires nécessaires	Très complète	Très peu	Peu nombreuse
PrestaShop	Simple	Complète, simple, agréable, bien documentée	Assez peu car la plupart sont intégrés nativement	Très bonne réactivité sur le forum anglais et français

b. Conclusion

J'ai proposé deux versions à mes supérieurs : **PrestaShop et OsCommerce**.

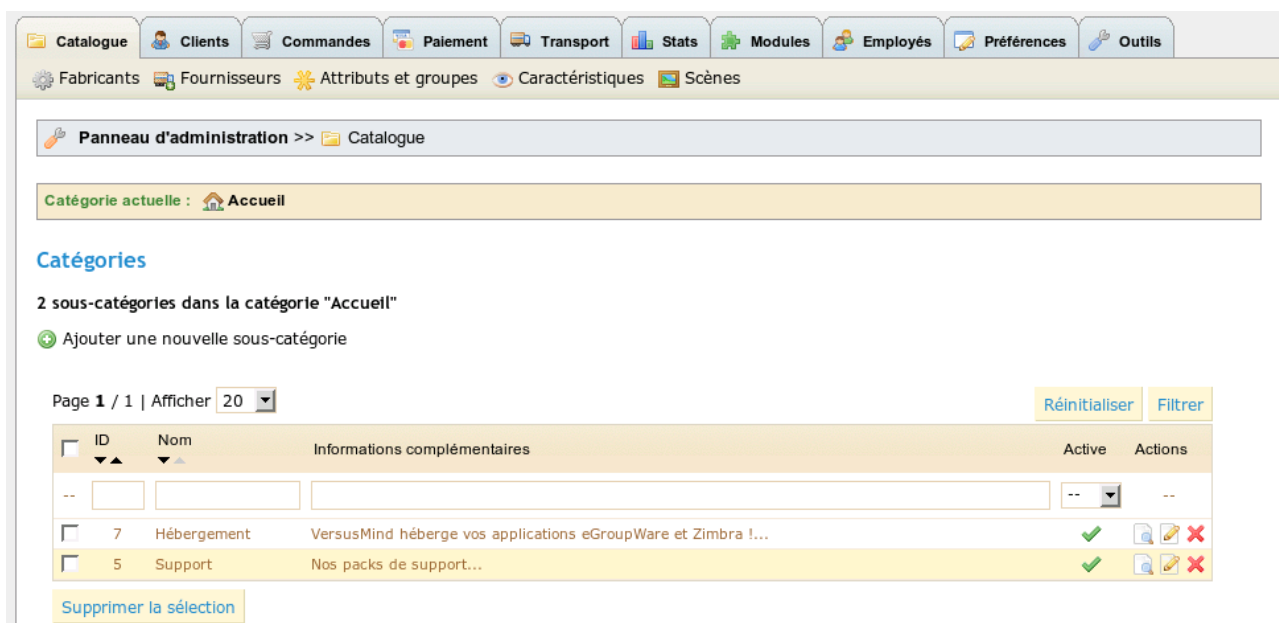
PrestaShop est une très bonne solution car c'est un logiciel assez récent (2005), qui est donc particulièrement bien adapté aux exigences du marché actuel. Par exemple, il propose un module permettant de payer via Paypal.

OsCommerce, quant à lui, est la plus ancienne solution d'e-commerce Open Source : il est le plus vieux logiciel, ce qui signifie qu'un maximum de modules sont disponibles (environ 5000), il peut donc s'adapter très facilement à tout type de demande.

Après en avoir discuté, nous avons choisi de conserver **PrestaShop** pour la raison suivante : PrestaShop, étant relativement récent, est codé en PHP objet, contrairement à OsCommerce qui est codé en PHP procédural. L'objet facilite donc grandement de futures modifications ou encore sa maintenance.

3. Présentation de PrestaShop

PrestaShop fut à l'origine créé en 2005 au sein de l'école informatique Epitech par cinq jeunes étudiants. Le projet d'origine, traduit en deux langues (français et anglais) portait le nom de phpOpenStore (POS). Les créateurs décidèrent de le livrer dans le domaine du logiciel libre dès sa conception. Il fut testé par plusieurs petits marchands ayant participé à la co-rédaction de son cahier des charges.



The screenshot displays the PrestaShop back-end administration interface. At the top, there is a navigation menu with icons and labels for various sections: Catalogue, Clients, Commandes, Paiement, Transport, Stats, Modules, Employés, Préférences, and Outils. Below this, there are more specific navigation options: Fabricants, Fournisseurs, Attributs et groupes, Caractéristiques, and Scènes. The main content area shows the 'Panneau d'administration >> Catalogue' breadcrumb. Below that, it indicates the current category: 'Catégorie actuelle : Accueil'. The section is titled 'Catégories' and shows '2 sous-catégories dans la catégorie "Accueil"'. There is a button to 'Ajouter une nouvelle sous-catégorie'. Below this, there is a pagination control showing 'Page 1 / 1 | Afficher 20' and buttons for 'Réinitialiser' and 'Filtrer'. A table lists the sub-categories with columns for 'ID', 'Nom', 'Informations complémentaires', 'Active', and 'Actions'. The table contains two rows: one for 'Hébergement' (ID 7) and one for 'Support' (ID 5). Both are active and have edit and delete icons. A 'Supprimer la sélection' button is located at the bottom left of the table area.

ID	Nom	Informations complémentaires	Active	Actions
7	Hébergement	VersusMind héberge vos applications eGroupWare et Zimbra !...	✓	
5	Support	Nos packs de support...	✓	

Back-end, partie administration

Voici désormais partie Front-end, autrement dit, la page d'accueil sur laquelle arriveront les clients. Cette partie a été complètement créée pour VersusMind afin qu'elle s'intègre au mieux avec la charte graphique de son site Internet.

The screenshot shows the VersusMind website's front-end. At the top, there's a header with the VersusMind logo, flags for France and the UK, and navigation links for 'contact' and 'plan du site'. On the right, it says 'Bienvenue, [nom] (déconnexion)' and shows a shopping cart with '1 produit 30,00 €'. The main content area is titled 'ShopMind' and includes a welcome message: 'Bienvenue sur le site de commande de support VersusMind'. Below this, a paragraph explains that users can order support for applications like OpenSource. The 'PRODUITS PHARES' section features four products: 'Conférence EnewsMind' (30,00 €), 'Egroupware' (50,00 €), 'Packs Zimbra' (50,00 €), and 'OpenOffice' (50,00 €). Each product has a 'Plus d'infos' button and an 'Ajouter au panier' button. The left sidebar has 'CATÉGORIES' (Support, Visio-conférences), 'NOUVEAUX PRODUITS' (no new products), and 'MEILLEURES VENTES' (Zimbra packs). The right sidebar includes a search bar, 'PANIER' (1 article 30,00 €), 'MON COMPTE' (Mes commandes, Mes avoirs, etc.), and 'INFORMATIONS' (Livraison, Mentions légales, etc.). The footer has 'Contactez-nous', 'Conditions d'utilisation', 'A propos', and 'Versusmind'.

Front-end, partie clients

4. Mise en place

J'ai commencé par télécharger l'application puis l'installer sur ma machine de test (dotée d'un serveur apache et d'une base de données MySQL).

Un thème par défaut étant installé, je me suis tout d'abord concentré sur la partie administration de PrestaShop.

Mes principaux travaux :

- ajouter des produits, paramétrer les réductions en fonction de la quantité achetée
- configurer les différentes possibilités de paiement, par chèque, par carte bancaire
- afficher certains modules sur la partie client, les catégories de produits, les liens vers les mentions légales
- sécuriser les communications client-serveur en SSL, cela a nécessité la création d'un certificat

J'ai ensuite porté une attention particulière au design du site. Je devais modifier la charte graphique de façon à ce qu'elle corresponde avec celle de VersusMind (en particulier son site web).

Les grandes étapes :

- compréhension du fonctionnement système de template de PrestaShop,
- création des différents éléments graphiques (bannières, boutons, onglets) grâce à Gimp, un logiciel de création et de traitement d'images
- création d'un template dédié à VersusMind.

5. Problèmes rencontrés

Un problème majeur qui nous a fait réfléchir pour l'adoption de PrestaShop était le fait que le module de paiement par carte bancaire n'est pas Open Source, il est payant. Nous avons donc du payer ce module (qui était gratuit sous OsCommerce).

Une version Open Source du module existe, j'ai essayé de le mettre en place mais cela a vraiment été très difficile. De plus, par la suite, j'ai vu que des grosses failles de sécurité étaient présentes dans ce module gratuit, ce qui nous a poussés à utiliser le module "officiel".

Un deuxième problème est apparu à la fin de cette mission, la gestion de la TVA.

VersusMind fournissant des services en France mais également à l'étranger, il fallait mettre en place un système permettant de choisir, dans certains cas, le prix TTC et dans d'autres cas, le prix HT. Cette fonction n'étant pas possible à mettre en place sous PrestaShop, nous sommes toujours à la recherche d'une solution alternative.

Une plate-forme de vidéoconférence avec Dimdim

1. Objectif

VersusMind propose à ses clients des formations sur les solutions Open Source de son catalogue (utilisation d'OpenOffice, création de macros OpenOffice, eGroupWare, Zimbra, Alfresco, etc.). L'objectif de cette mission était de mettre en place un système de visioconférence à distance avec possibilité de partage de PDF ou d'écran d'ordinateur afin que les clients n'aient pas à se déplacer pour suivre ces formations. Cet outil permettrait également de faire des présentations des différents logiciels à de futurs clients.

L'idéal était d'avoir une solution qui utilisait uniquement le port 80 car, en général, seul le port 80 (ainsi que quelques autres) sont accessibles dans une entreprise.

2. Présentation de Dimdim

a. Fonctionnalités et tarifs

Dimdim est une application intranet-internet de réunion à distance, à l'instar de logiciels comme Netmeeting ou Ekiga. Dimdim permet de diffuser la voix de plusieurs personnes ainsi que la webcam du maître de conférence (entre autres). Ce dernier peut choisir, en outre, d'afficher, et donc de partager avec les intervenants :

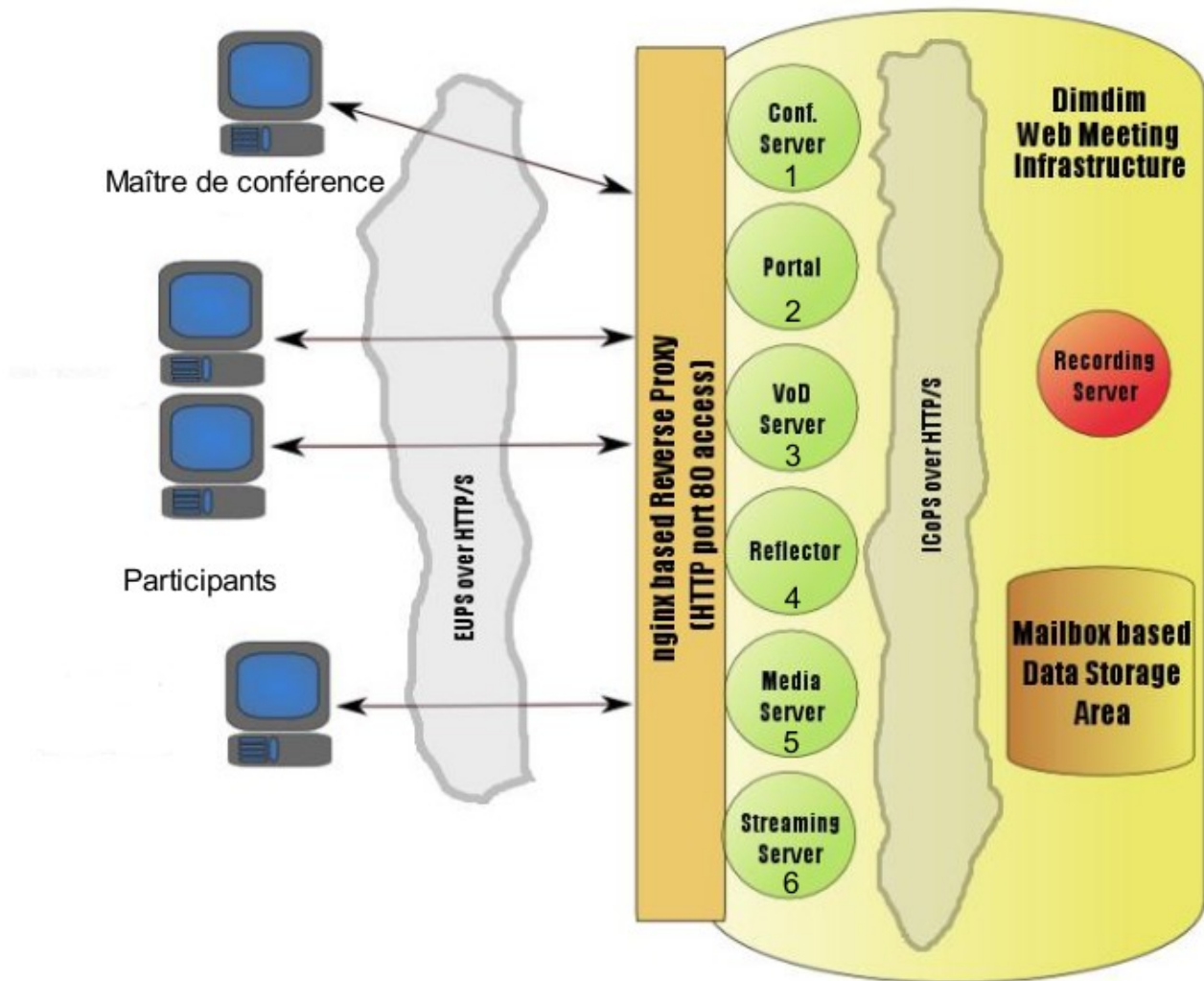
- son écran d'ordinateur
- un tableau blanc (où tout le monde peut écrire)
- un document au format PDF (que tout le monde peut annoter)
- un document au format PowerPoint (que tout le monde peut annoter)

Dimdim est un logiciel créé en 2006 par la société éponyme. C'est, à l'origine, une solution commerciale. Plusieurs offres sont proposées sur le site officiel :

	Entreprise	Pro	Free	Open Source
Tarif	950\$ par mois	19 \$ par mois	Gratuit	Gratuit
Participants max	1000	50	20	Illimité
Installation	Hébergé chez Dimdim	Hébergé chez Dimdim	Hébergé chez Dimdim	À héberger chez soi
Version	5.0	5.0	5.0	4.5
Support	Dimdim	Dimdim	Dimdim	Communauté

L'offre Open Source semble plutôt alléchante, gratuite et illimitée en participants maximum. Cependant, je me suis rendu compte au fil de cette mission que le revers de la médaille n'était pas négligeable. En effet, parmi les effets indésirables de cette version Open Source de Dimdim, nous pouvons signaler en premier lieu, l'installation quasi-impossible sous Debian, le support inexistant ainsi que l'impossibilité de programmer une conférence.

b. Comment fonctionne Dimdim



1. Serveur de conférence : C'est le front-end qui permet d'afficher les webcams, les présentations, le tableau blanc, etc.
2. Le portail est la page d'accueil de Dimdim sur laquelle nous pouvons choisir le nom de la conférence, les invités, sa durée, son lancement, etc.
3. Le serveur VOD permet la diffusion d'un enregistrement vidéo d'une précédente conférence au format FLV
4. Reflector fournit un système de partage de bureau. C'est en fait un protocole VNC modifié encapsulé dans du HTTP
5. Le Media server va être utilisé pour partager des ressources comme les présentations powerpoint ou fichier PDF
6. Le serveur streaming est utilisé pour la diffusion de vidéo (via webcam) ou de l'audio (via les microphones) du maître de conférence et des participants.

3. Problèmes pour la mise en place

La politique de Dimdim est claire : « nous voulons bien vous mettre à disposition une version Open Source mais nous n'assurons absolument aucun support technique », ce qui est compréhensible.

Je me suis donc rendu sur la page Sourceforge du projet Dimdim Open Source afin d'aller chercher un paquet Debian. Cependant, aucun paquet Debian n'est disponible, il y a uniquement un paquet prévu pour CentOS (le système de paquets RPM), une image vmware de cette même version (basée sur CentOS également) ainsi que les sources de la version 4.1.

Je télécharge finalement les sources de la 4.1 (je pensais que cette version ne devait pas être très différente de la version 4.5). Je télécharge donc l'archive, l'extrait sur mon serveur, et là, je me rends compte qu'il n'y a aucune doc, aucun readme, aucune procédure d'installation ! Plutôt décevant à mes yeux.

Je suis de ce fait allé chercher de l'aide sur le forum Open Source du site Dimdim mais je n'y ai trouvé aucun résultat pertinent. Pour moi, ils ont simplement mis à disposition cette version Open Source uniquement pour se faire de la publicité.

Voici donc le contexte des mes premiers jours avec Dimdim, rien de très encourageant ! Mes recherches sur Internet n'ont également rien donné. J'avais, comme je l'ai dit, des sources sans aucun support. Je précise que ces sources n'étaient pas celles habituelles (du genre ./configure, make, make install) car Dimdim est, je vous le rappelle, un mélange de plusieurs technologies avec beaucoup de logiciels embarqués différents (Tomcat car beaucoup de Java, Nginx pour faire office de reverse proxy, perl, python, etc.).

4. Mise en place

Après des heures et des heures de recherches infructueuses, j'ai trouvé une solution en téléchargeant le paquet RPM à l'origine prévu pour CentOS que j'ai converti grâce au logiciel Alien. J'ai donc obtenu un paquet DEB que j'ai réussi à installer. Malheureusement, tous les liens étaient différents (car, à l'origine, prévus pour CentOS) et, pour couronner le tout, je n'avais que très très peu de logs ! J'ai donc, après une semaine de recherches, de modifications, de réinstallation, installé l'image vmware sur une machine virtuelle pour avoir une version fonctionnelle et pour pouvoir comparer les fichiers entre les deux versions. Un vrai travail de fourmis !

5. Conclusion

Après avoir passé plus de deux semaines à temps complet sur Dimdim, j'ai enfin réussi à l'installer et à le faire fonctionner sous Debian Lenny. Toutes les fonctionnalités sont présentes, même celles qui ne l'étaient pas dans l'image de vmware (comme le partage de documents Powerpoint ou PDF). J'ai donc décidé de créer un tutoriel en anglais et en français résumant pas à pas toutes les manipulations nécessaires à la mise en place d'un tel outil sous Debian. Ce tutoriel est utile à beaucoup de personnes désormais. La version française est disponible en annexe à ce rapport.

Enfin, j'ai intégralement traduit Dimdim en français car il n'était disponible qu'en anglais puis j'ai modifié le design de manière à ce qu'il s'intègre au maximum avec le reste des sites de VersusMind.

Mise en place d'un système d'authentification unique grâce à CAS.

1. Objectif

Plusieurs logiciels sont utilisés au sein de la société VersusMind : Un logiciel de messagerie Zimbra, un logiciel collaboratif eGroupWare, etc.

Les employés peuvent s'authentifier auprès de ces logiciels grâce à un annuaire LDAP. Cependant, à chaque fois qu'un employé souhaite utiliser plusieurs de ces logiciels, cela nécessite une nouvelle authentification. L'idée était de n'avoir qu'à s'authentifier une seule fois pour être reconnu sur l'ensemble des logiciels de l'entreprise.

2. Présentation du SSO (Single Sign On)

L'architecture traditionnelle d'authentification des utilisateurs d'une entreprise est basée sur un annuaire (LDAP ou AD). LDAP est un système d'annuaire où sont centralisées, selon les cas, les données administratives, les comptes et mots de passe des utilisateurs du système d'information.

Ainsi, chaque application d'une entreprise est couplée à un annuaire LDAP pour l'authentification d'un utilisateur. De cette manière, à chaque nouvelle connexion (et authentification), le logiciel va interroger l'annuaire LDAP afin de vérifier si le nom d'utilisateur et le mot de passe correspondent. Dès lors, l'utilisateur utilise toujours le même mot de passe, celui qui est stocké dans l'annuaire LDAP.

Cette méthode est très pratique car elle permet de centraliser les données, cependant il y a toujours un problème : l'utilisateur est obligé de taper son mot de passe à chaque fois qu'il veut utiliser un autre logiciel. Un exemple concret : j'utilise quotidiennement deux logiciels : eGroupWare et Zimbra. Je dois taper mon login et mon mot de passe quand j'accède à eGroupWare et je dois recommencer pour me connecter à Zimbra.

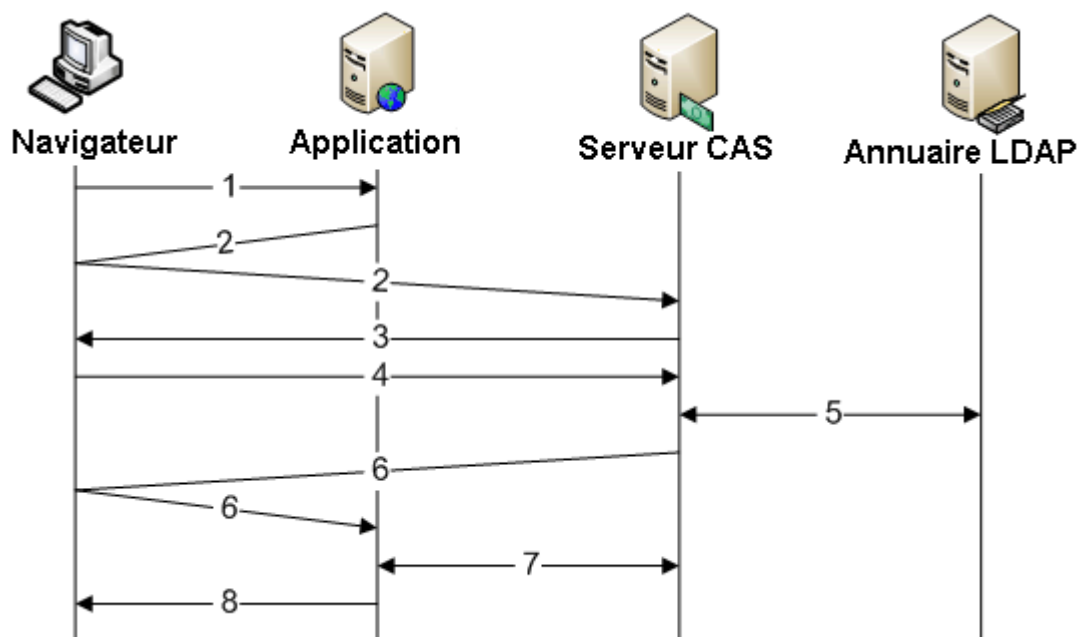
Le but du Single Sign On, comme son nom l'indique, est de n'avoir qu'à s'authentifier qu'une seule fois pour être reconnu sur l'ensemble des logiciels de l'entreprise. Ainsi, pour reprendre mon exemple, je n'aurai plus qu'à me connecter sur eGroupWare pour être reconnu et connecté automatiquement sur Zimbra, Alfresco, et bien d'autres.

Pour mettre en place cette solution de SSO, j'ai utilisé le logiciel CAS.

3. Présentation de CAS

CAS signifie **Central Authentication Service**. Il a été développé en 2004 par l'Université de Yale. CAS est un système d'authentification unique : on s'authentifie sur une application et on est alors authentifié sur toutes les applications qui utilisent le même serveur CAS.

Ce logiciel est essentiellement basé sur des requêtes HTTP pures, sur une notion d'échange de tickets, un peu à la manière de Kerberos. Ces tickets sont des « opaque handles » : ils ne transportent aucune information.



1. Un internaute accède à une application web.
2. Le navigateur est redirigé vers le serveur CAS
3. Le serveur envoie un formulaire d'authentification
4. L'internaute s'authentifie
5. Le serveur vérifie le couple compte / mot de passe grâce à l'annuaire LDAP
6. Le serveur CAS crée un cookie de session et redirige le navigateur vers l'application avec un ticket à utilisation unique
7. L'application valide le ticket en contactant directement le serveur CAS qui retourne l'identifiant de la personne
8. L'internaute accède à la ressource demandée

4. Mise en place de CAS

La mise en place de CAS a été assez fastidieuse du fait que ce soit une application codée en Java, il fallait donc utiliser Tomcat pour l'héberger. Je détaille ce point dans le paragraphe « Problèmes rencontrés ».

Là aussi, j'ai écrit un tutoriel qui explique pas à pas, comment installer un serveur Tomcat, puis comment installer CAS sur ce Tomcat, comment sécuriser les connexions avec ce serveur CAS (en mettant en place des certificats) et, enfin, comment coupler CAS avec un annuaire LDAP pour l'authentification).

Ce tutoriel est joint en annexe.

5. CASifier une application

Différents modules ont été créés afin de CASifier une application, autrement dit, de rendre une application compatible avec CAS.

Voici quelques modules disponibles :

- **mod_auth_cas** est utilisé en cas d'authentification apache via CAS, par exemple, un répertoire protégé via htaccess, au lieu d'utiliser un fichier htpasswd, nous pouvons utiliser l'authentification CAS.
- **Client for Java** est un client d'authentification pour les applications Java. De plus en plus d'applications étant développées en Java, ce client est devenu rapidement indispensable.
- **phpCAS** est le client que j'ai utilisé, c'est un client CAS pour les applications PHP qui a été créé et est actuellement maintenu par un employé du CRI de l'Université Nancy 2 : Julien Marchal.

a. Exemple simple d'authentification via phpCAS

```
<?php
// import de phpCAS lib (installé par le client phpCAS)
include_once('CAS.php');

phpCAS::setDebug();

// initialize phpCAS
phpCAS::client(CAS_VERSION_2_0, 'ip_serveur', port_serveur, '');

// pas de validation SSL envers le serveur CAS
phpCAS::setNoCasServerValidation();

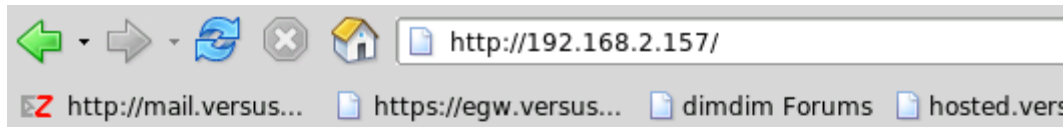
// force l'authentification CAS
phpCAS::forceAuthentication();

// ici, l'utilisateur est authentifié par le serveur CAS
// le login de l'utilisateur peut être connu grâce à phpCAS::getUser().


// logout si désiré
if (isset($_REQUEST['logout'])) {
    phpCAS::logout();
}

// la suite du code ne sera visible que si l'utilisateur est authentifié, dans
// le cas contraire, il sera redirigé sur la page de login de CAS, ici on
// affiche juste son nom et la version du serveur CAS ainsi qu'un lien de logout
?>
<html>
  <head>
    <title>phpCAS simple client</title>
  </head>
  <body>
    <h1>Successfull Authentication!</h1>
    <p>the user's login is <b>php echo phpCAS::getUser(); ?&gt;&lt;/b&gt;.&lt;/p&gt;
    &lt;p&gt;phpCAS version is &lt;b&gt;<?php echo phpCAS::getVersion(); ?&gt;&lt;/b&gt;.&lt;/p&gt;
    &lt;p&gt;&lt;a href="?logout="&gt;Logout&lt;/a&gt;&lt;/p&gt;
  &lt;/body&gt;
&lt;/html&gt;</pre
```

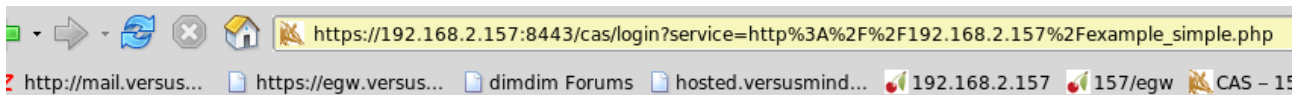
b. Illustration de l'exemple



Index of /

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 ajxp/	12-Jun-2009 13:50	-	
 egw/	24-Nov-2008 11:42	-	
 example_simple.php	07-May-2009 12:20	1.0K	

Je vais exécuter l'exemple ci-dessus nommé example_simple.php qui est situé à la racine de mon serveur de test à l'adresse 192.168.2.157.



Central Authentication Service (CAS)

Entrez votre identifiant et votre mot de passe.

Identifiant:

Benjamin

Mot de passe:

Prévenez-moi avant d'accéder à d'autres services.

SE CONNECTER

effacer

Pour des raisons de sécurité, veuillez vous déconnecter et fermer votre navigateur lorsque vous :

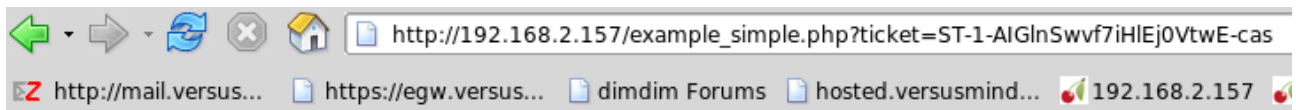
Languages:

[English](#) | [Spanish](#) | [French](#) | [Russian](#) | [Nederlands](#) | [Svenskt](#) | [Italiano](#) | [Urdu](#) | [Chinese \(Simplified\)](#)

Une fois que j'ai cliqué sur cet exemple, je suis automatiquement redirigé vers l'adresse du serveur CAS (dans mon cas, il se trouve sur la même machine) avec cette URL composée ainsi :

https://192.168.2.157:8443/cas/login : adresse du serveur CAS

?service=http%3A%2F%2F192.168.2.157%2Fexample_simple.php : URL de redirection après authentification (qui est également celle d'où nous venons).



Successfull Authentication!

the user's login is **benjamin**.

phpCAS version is **1.0.0-`{phpcas.release}`**.

[Logout](#)

Lorsque je me suis authentifié sur le serveur CAS (via LDAP), j'ai été redirigé sur ma page d'origine (adresse présente dans l'URL de la copie d'écran précédente), avec, dans ma nouvelle URL, un ticket de validation.

La page `example_simple.php` va interroger le serveur CAS afin de confirmer que je suis bien autorisé à accéder à son contenu (identification faite grâce au ticket présent dans l'URL).

A ce moment, je suis authentifié sur le serveur CAS, ce qui signifie que lorsque j'accéderai à une autre application CASifiée, je serai automatiquement reconnu et authentifié sur cette application.

c. CASification d'eGroupWare :

Installation de la librairie phpCAS :

```
apt-get install php-pear
pear install DB
apt-get install php5-curl
pear install http://www.ja-sig.org/downloads/cas-clients/php/1.0.0/CAS-1.0.0.tgz
```

La librairie se trouve normalement dans `/usr/share/php/CAS`

Configuration de eGroupWare avec CAS :

eGroupWare ayant déjà été CASifié, l'option est donc disponible directement dans l'interface d'administration. Ainsi, au lieu de choisir une authentification classique (en créant les comptes directement dans eGroupWare) ou via un annuaire LDAP, nous choisissons une authentification via CAS. Le serveur CAS est quant à lui relié à l'annuaire LDAP pour l'authentification.

Si vous utilisez l'authentification CAS (Central Authentication Service) :

Nom du serveur CAS hôte :	<input type="text" value="192.168.2.157"/>
Exemple : sso-cas.univ-rennes1.fr	
Port du serveur CAS :	<input type="text" value="8443"/>
Exemple : 443	
URI du serveur CAS :	<input type="text" value="/cas"/>
Mode d'authentification :	<input type="text" value="php Client"/>
Validation SSL :	<input type="text" value="Non"/>
Certificat (PEM ou CA) :	<input type="text"/>

Modifications du logout :

CAS a été implémenté dans la version 1.4 d'eGroupWare. Nous utilisons actuellement la version 1.6. Le système de logout a été modifié entre ces deux versions. Une modification est donc à faire au sein du fichier de logout pour qu'il soit compatible avec CAS. Il faut simplement remplacer la ligne suivante :

```
phpCAS::logout($GLOBALS['egw_info']['server']['webserver_url'].'/login.php?
cd=1&domain='.$GLOBALS['egw_info']['user']['domain']);
```

par celle-ci :

```
phpCAS::logout();
```

6. Problèmes rencontrés

J'ai rencontré deux principaux problèmes par rapport à la mise en place de CAS.

CAS étant une application Java, il fallait mettre en place une plate-forme d'hébergement Java : **Tomcat**. Cela a été assez fastidieux car je ne connaissais absolument pas Java et encore moins le système d'hébergement Tomcat. Les logs que j'ai pu exploiter au cours de l'installation et de l'utilisation étaient, au début, complètement incompréhensibles pour moi. J'ai donc dû m'adapter à ce nouveau langage.

Enfin, le problème le plus important de cette mission a été la **mise en place d'un système de certificats au sein de Tomcat**. En effet, pour CAS, une connexion sécurisée est fortement recommandée (pour ne pas dire obligatoire). Il faut donc mettre en place les certificats SSL correspondants.

Là où dans Apache il suffit de renseigner l'adresse de ces certificats dans le fichier de configuration, l'opération s'avère plus délicate pour un serveur Tomcat. De plus, dans mon cas, j'utilisais un certificat auto-signé (pour mes tests), il fallait donc ajouter ce certificat dans les "certificats de confiance" pour qu'il ne demande plus aucune confirmation de validation.

Ces tâches sont décrites dans le tutoriel CAS disponible en annexe.

Mise en place d'une solution de supervision réseau : Zabbix

1. Objectif :

VersusMind possède plusieurs serveurs et intervient sur plusieurs serveurs chez ses clients. L'objectif de cette mission était donc de mettre en place un système de supervision de tous ces serveurs.

2. Présentation de Zabbix :

Zabbix est un logiciel de monitoring réseau Open Source et multiplate-forme créé en 2002 par un Letton, Alexei Vladishev. Il permet de surveiller le statut de divers services réseau, serveurs, postes de travail et autres matériels (routeurs, parefeu, imprimantes, etc.).

a. Ce que Zabbix peut surveiller :

- Les protocoles : HTTP, SSH, FTP, IMAP, SMTP, etc.
- Les ports : 21, 80, 1434, 10050, etc.
- Les processus tournant sur une machine : apache2, sshd, etc.
- Les ressources matérielles : mémoire vive, espace disque, CPU, température, etc.
- Les sites web : temps de réponse, vitesse de transfert, etc.
- L'intégrité des fichiers : /etc/passwd a été modifié, etc.
- Analyse des logs

b. Comment Zabbix peut-il surveiller :

- Des tests de connexion simples (ping, test de ports, etc.)
- Interrogation d'un agent SNMP présent sur la machine à surveiller
- Installation d'un agent Zabbix sur la machine à surveiller (agent disponible sur toutes les plates-formes)

En résumé, le contrôle d'une machine fonctionne de cette façon :

On crée un objet, par exemple, l'objet température processeur.

Nous pouvons ensuite :

- générer un graphique nous montrant l'évolution de la valeur de cet objet (ici la température du processeur tout au long de la journée)
- créer un déclencheur (trigger) qui s'active si la valeur de l'objet remplit une certaine condition (par exemple, si la température dépasse 90°C).

Si ce trigger se déclenche, en plus de l'alerte sur le panel web, nous pouvons être averti par mail, SMS, messagerie instantanée (Jabber) ou bien exécuter un script sur la machine concernée.

Typiquement, si la température du processeur dépasse 100°C, je demande l'envoi d'un mail à la personne responsable puis j'exécute un script qui va arrêter la machine à distance.

3. Mise en place de Zabbix

a. Topologie à superviser

Non divulguable

b. Installation du serveur

Chez VersusMind, les serveurs sont sous distribution Debian Lenny. Par conséquent, Zabbix est disponible dans les paquets officiels, accessibles via la commande apt-get. En revanche, la version disponible à partir des dépôts est la version 1.4.

Le dernière version stable étant la version 1.6, je voulais installer cette dernière. Il a donc fallu que je compile Zabbix à partir des sources téléchargées sur le site officiel.

J'ai donc téléchargé les sources puis j'ai compilé l'application grâce à la commande suivante :

```
./configure --prefix=/usr --with-mysql --with-net-snmp --with-libcurl --enable-server --enable-agent
```

c. Installation des clients

Selon les données que nous souhaitons surveiller, nous avons installé, ou non, un agent Zabbix sur les machines à superviser. Pour certaines machines, nous voulions pouvoir vérifier la *mémoire vive disponible*, le *pourcentage d'utilisation du processeur*, l'*espace disque disponible*, etc. Dans ces cas, nous devons obligatoirement installer un agent Zabbix. Pour d'autres machines, nous ne souhaitons faire qu'une *vérification de sa présence sur le réseau (via un ping)*, dans ce cas, aucun agent n'est nécessaire sur le client. Enfin, pour d'autres machines, nous avons utilisé l'*agent SNMP installé par défaut* sur la plupart des équipements réseau.

Zabbix est capable d'interroger un agent SNMP sur un périphérique réseau. Cela m'a été bien utile pour surveiller certaines machines comme, par exemple, une imprimante. Je vais donc détailler plus précisément SNMP car j'ai passé plusieurs jours à l'étudier et à le mettre en place.

4. SNMP

a. Présentation

SNMP (Simple Network Management Protocol) est un protocole de communication qui permet aux administrateurs de gérer les équipements du réseau, superviser et de diagnostiquer des problèmes réseaux ou matériels à distance. Chaque équipement que l'on voudra surveiller doit disposer d'un agent SNMP.

Cet agent est, en fait, un serveur, c'est à dire qu'il reste à l'écoute d'un port particulier : le port UDP 161. La principale fonction de cet agent est de rester à l'écoute des éventuelles requêtes que l'administrateur lui enverra (en l'occurrence ici, Zabbix).

L'agent SNMP stocke ses données dans une base de données virtuelle, la MIB. Par exemple, grâce à la commande suivante, je vais interroger une partie de la MIB de l'imprimante :

```
# snmpwalk -v 1 -c public 192.168.2.10 .1.3.6.1.2.1.25.3

HOST-RESOURCES-MIB::hrDeviceIndex.1 = INTEGER: 1
HOST-RESOURCES-MIB::hrDeviceType.1 = OID: HOST-RESOURCES-TYPES::hrDevicePrinter
HOST-RESOURCES-MIB::hrDeviceDescr.1 = STRING: Officejet 7400 series
HOST-RESOURCES-MIB::hrDeviceID.1 = OID: SNMPv2-SMI::enterprises.11.2.3.9.1.2.46
HOST-RESOURCES-MIB::hrDeviceStatus.1 = INTEGER: running(2)
HOST-RESOURCES-MIB::hrDeviceErrors.1 = Counter32: 0
HOST-RESOURCES-MIB::hrPrinterStatus.1 = INTEGER: idle(3)
HOST-RESOURCES-MIB::hrPrinterDetectedErrorState.1 = Hex-STRING: 00
```

Nous pouvons voir que l'agent SNMP retourne de nombreuses valeurs pouvant servir à la supervision. Dans notre exemple, l'agent SNMP de l'imprimante retourne (entre autres) le nom de l'imprimante, son status, les éventuelles erreurs, etc.

Les éléments d'une MIB sont accessibles via des identifiants uniques (les OID), connus et déclarés dans les RFC. Par exemple, les éléments ci-dessus appartiennent à la table hrDevice formalisée dans la RFC 1514. Ils correspondent à l'entrée .1.3.6.1.2.1.25.3 de la MIB.

b. Intégration à Zabbix

Si une machine ne possède pas d'agent SNMP, il faut, en premier lieu, l'installer (pour debian : `apt-get install snmpd`).

La plupart des équipements réseau actuels sont déjà équipés de cet agent.

Ensuite, dans le cas d'une installation de l'agent, il faut permettre la consultation de cet agent à distance, pour cela, il faut éditer le fichier `/etc/default/snmpd` et supprimer l'IP 127.0.0.1

Comment surveiller une **chaîne de caractère** provenant de la MIB avec Zabbix :

Nous allons, en premier lieu, lister les données de la MIB disponibles sur un agent (la commande `snmpwalk` et tous les outils similaires nécessitent un serveur SNMP installé (`apt-get install snmp`) :

```
snmpwalk -v 1 -c public 192.168.2.10
```

```
[...]
SNMPv2-MIB::sysDescr.0 = STRING: HP ETHERNET MULTI-ENVIRONMENT
[...]
```

Il nous faut ensuite connaître l'OID de cette entrée :

```
snmpget -v1 -c public -On 192.168.2.10 sysDescr.0
```

```
.1.3.6.1.2.1.1.1.0= STRING: HP ETHERNET MULTI-ENVIRONMENT
```

L'OID est la suite de nombre unique correspondant à une entrée de la MIB. C'est cette suite qu'il faudra ajouter dans Zabbix. Il ne nous reste plus qu'à créer un nouvel objet dans Zabbix :

The screenshot shows the Zabbix configuration page for a new item. The title is 'Item 'OfficeJet 7410:Hostname_SNMP''. The configuration fields are as follows:

- Description: Hostname_SNMP
- Type: SNMPv1 agent
- SNMP community: public
- SNMP OID: .1.3.6.1.2.1.1.1.0
- SNMP port: 161
- Key: sysdesc (with a 'Select' button)
- Type of information: Character
- Update interval (in sec): 30
- Flexible intervals (sec): No flexible intervals
- New flexible interval: Delay 50, Period 1-7,00:00-23:59 (with an 'Add' button)
- Keep history (in days): 90 (with a 'Clean history' button)
- Keep trends (in days): 365
- Status: Active
- Show value throw map: As is
- Applications: -None-

At the bottom of the form, there are four buttons: 'Save', 'Clone', 'Delete', and 'Cancel'.

5. Difficultés rencontrées

a. SNMP

Nous avons étudié le protocole SNMP en début d'année scolaire, mais je n'avais pas très bien assimilé ce cours. Chez VersusMind, je n'avais pas d'autres choix que d'utiliser ce protocole car j'ai dû monitorer l'imprimante. Les supports de cours utilisés dans le cadre de ma licence m'ont donc été très utiles.

Je me suis retrouvé face à un problème assez important, il concernait une entrée particulière de la MIB de cette imprimante :

```
HOST-RESOURCES-MIB::hrPrinterDetectedErrorState.1 = Hex-STRING: 00
```

Cette valeur est une chaîne hexadécimale. Le problème était que Zabbix ne gère pas ce genre de données. Il gère très bien les chaînes de caractères traditionnelles et les nombres mais pas les chaînes hexadécimales.

Après de nombreuses recherches, j'ai fini par trouver la solution : il a fallu que je convertisse la chaîne hexadécimale en chaîne de caractère traditionnelle. Ensuite, je pouvais interpréter le résultat au sein du serveur Zabbix. Par exemple, si le résultat donne « ` », cela signifie qu'il n'y a plus assez d'encre, si le résultat donne « (», cela signifie qu'il n'y a plus de papier.

b. Agent Mac OS

L'installation des agents n'a pas posé de problème pour les postes sous Windows (pour lesquels un installateur automatique a été créé) ainsi que pour les postes sous Linux (ou selon les cas, un paquet existe ou les sources sont facilement installables). Là où j'ai rencontré un problème était dans le cas d'une installation sur un poste sous MacOS.

En effet, il n'existe aucun paquet pré-compilé. En théorie, il faut installer l'agent grâce aux mêmes sources que pour Linux. En pratique, la compilation est impossible car les sources sont incompatibles. Après beaucoup de recherches, je n'ai trouvé aucune personne n'ayant réussi à installer cet agent sur une machine sous MacOS.

c. Agent en mode actif

En général, le serveur Zabbix va interroger lui-même les différents agents (écoutant sur le port 50010). Ainsi, il faut que le port 10050 du client soit accessible depuis le serveur.

Parfois, pour des postes à distance, derrière un pare-feu, il est possible que le serveur ne puisse pas contacter directement le client. Dans ce cas, il faut utiliser une fonctionnalité particulière de l'agent : le **mode actif**. Dès lors, c'est l'agent qui fournira ses informations au serveur et non l'inverse. Dans ce cas, il faut que la communication entre le client et le port 10051 du serveur soit possible.

Lorsque nous avons dû superviser un client de VersusMind, il a été difficile de mettre ce système en place car nous ne pouvions pas ouvrir les ports du pare-feu de ce client. Nous avons essayé d'encapsuler les trames dans un tunnel SSH (car nous avions un accès SSH) mais nous n'avons pas réussi.

Nous nous sommes finalement décidés à appeler le client qui a dû intervenir sur son pare-feu pour débloquer le port correspondant.

CONCLUSION

Conclusion

1. Aspects techniques

Cette année de licence professionnelle m'a permis d'acquérir de nombreuses compétences dans le domaine de l'administration réseau et du système Linux en général. Cette formation a été très profitable pour moi. Ce stage m'a permis de mettre en pratique une grande partie de la théorie qui m'a été enseigné tout au long de cette année.

Cependant, comme chacun le sait, la pratique est très différente de la théorie. J'ai pu, par exemple, assimiler certaines choses beaucoup plus facilement en les mettant directement en pratique. Le protocole SNMP, cité ci-dessus, en est un bon exemple.

J'ai également acquis des compétences dans le domaine de Java et plus particulièrement de Tomcat. Le langage Java n'étant pas enseigné en ASRALL, nous n'avions pas vu comment mettre en place un serveur Tomcat. J'ai dû l'installer plusieurs fois chez VersusMind et ai donc appris beaucoup sur les applications Java et leur hébergement.

2. Aspects relationnels

Comme je l'ai évoqué dans le début de ce rapport, mes conditions de travail étaient excellentes, en effet, je partageais un bureau (open space) avec les huit employés et stagiaires de VersusMind. Un open space extrêmement bénéfique, tant sur le plan relationnel que productif.

En effet, lorsque j'avais une question, que ce soit une question de programmation, d'administration réseau, d'administration Linux ou même de syntaxe, d'orthographe ou de traduction, je pouvais demander l'avis et le conseil de mes collègues instantanément et j'avais, en général, la réponse immédiatement. Un gain de productivité notable.

Un autre avantage, cette fois-ci plus personnel, est que j'ai pu suivre l'évolution de tous les aspects de VersusMind au cours de ces douze semaines. Par exemple, j'entendais les conversations téléphoniques entre le commercial et les clients, mais également les plaintes des clients sur le support téléphonique et la négociation de certains éléments d'un dossier entre les membres de VersusMind. Parfois, on me demandait mon avis ou de l'aide, je n'étais pas confiné dans un bureau seul et j'avais donc l'impression de faire partie intégrante de l'entreprise. Cette sensation est appréciable et donne d'autant plus envie de s'investir pour l'entreprise.

Un autre point que j'ai particulièrement apprécié pendant ce stage était le contact avec les clients de VersusMind. En effet, Benoît KOCH, le dirigeant, a eu suffisamment confiance en moi pour me proposer des interventions directes chez les clients. Par exemple, j'ai fait une présentation complète pendant une heure de Zabbix à trois clients et j'ai également fait du support (par téléphone ou par mail) ainsi qu'une intervention directe d'une journée chez un client.

Je le remercie encore de m'avoir accordé cette confiance, d'autant plus que pour le support, je ne maîtrisais absolument pas le sujet en recevant les questions des utilisateurs.

Cela a donc été très formateur de devoir apprendre très rapidement un concept ou/et une fonctionnalité (comme, par exemple, OpenVPN ou le publipostage dans OpenOffice) pour le compte d'un client pour qui je devais régler un problème.

3. Aspects personnels

J'ai créé plusieurs projets en rapport avec le logiciel libre de manière tout à fait bénévole. J'use et abuse des logiciels libres dans ma vie privée. Je ne connaissais que la partie bénévole et gratuite des logiciels libres. Ce stage m'a permis de voir l'autre facette du logiciel libre : le côté professionnel.

La philosophie est totalement différente, là où dans le « milieu privé », les logiciels Open Source sont plébiscités pour leur gratuité, leur partage facile, légal et illimité, dans le monde professionnel, nous y voyons avant tout une formidable opportunité de modifier le code source (qui est ouvert et accessible) pour l'adapter au maximum selon les exigences d'un client. Ainsi, il est possible de vendre un module ou une fonctionnalité supplémentaire pour un logiciel qui sera créé grâce à l'ouverture de son code source.

Cet aspect des logiciels libres m'a permis de me rendre compte d'une chose, il est relativement facile de participer à l'amélioration d'un logiciel, que ce soit en corrigeant un bug, ou en développant une fonctionnalité jusque là inutilisée dans ce logiciel par exemple.

Cela me donne d'autant plus envie de participer à la modification de certains logiciels libres. Par exemple, j'ai créé, depuis ce stage, un projet de distribution Linux, basée sur Debian, spécifique à une utilisation particulière : la mise en place d'un serveur personnel chez soi.

4. Conclusion générale

Pour conclure, je peux sans aucun doute dire que ce stage a été le plus intéressant et le plus formateur de ma scolarité. J'y ai appris beaucoup de choses, que ce soit d'un point de vue technique, relationnel ou personnel.

Cela m'a permis de m'immerger progressivement dans le monde professionnel, notamment au niveau de l'intensité et la rigueur du travail mais également les notions de date limite ("deadline"). Ce monde est vraiment très différent du milieu scolaire, avec ses avantages et ses inconvénients.

Il me tarde de trouver un emploi et de mettre en pratique tout ce que j'ai pu voir au cours de ma scolarité !

Annexes

1. Tutoriel créé pour l'installation de Dimdim en français

a. Prérequis

Quelques logiciels supplémentaires sont nécessaires à l'utilisation de Dimdim. Voici ces différents logiciels et la manière de les installer :

Openoffice.org 3

Rendez-vous sur la page de téléchargement officielle d'OpenOffice :
<http://download.openoffice.org/other.html>

Puis téléchargez un paquet DEB, par exemple, à cette adresse :
http://openoffice.cs.utah.edu/localized/fr/3.0.1/OOo_3.0.1_LinuxIntel_install_fr_deb.tar.gz

On enregistre cette archive dans notre répertoire personnel (par exemple) puis :

```
su (nous allons faire toute l'installation de Dimdim en root)
cd ~ (le chemin doit être celui où se trouve votre paquet DEB)
tar xvf Ooo_3.0.1_LinuxIntel_install_fr_deb.tar.gz
cd OOO300_m15_native_packed-1_fr.9379/DEBS/
dpkg -i *.deb
```

Java v1.6

```
apt-get install sun-java6-bin
```

Nous allons ensuite modifier les variables d'environnement de notre système

```
echo "JAVA_HOME=/usr/lib/jvm/java-6-sun-1.6.0.12" >> /root/.bashrc
echo "export JAVA_HOME" >> /root/.bashrc
```

Déconnectez-vous de root et reconnectez-vous (afin d'initialiser la variable JAVA_HOME)

```
echo "PATH=$JAVA_HOME/bin:$PATH" >> /root/.bashrc
echo "export PATH" >> /root/.bashrc
```

Déconnectez-vous de root et reconnectez-vous (afin de ré-initialiser la variable PATH)

Tapez la commande "env", vous devriez avoir ces lignes :

```
PATH=/usr/lib/jvm/java-6-sun-1.6.0.12/bin:/bin:.....
JAVA_HOME=/usr/lib/jvm/java-6-sun-1.6.0.12
```


Autres programmes

```
apt-get install gawk
apt-get install python-uno python2.5-dev
apt-get install unzip
apt-get install openoffice.org-java-common openoffice.org-headless
apt-get install bin-utils build-essential
```

Modules Python

Nous allons ajouter quelques modules nécessaires pour l'utilisation de Dimdim :

CherryPy

Répertoire de téléchargement : <http://download.cherrypy.org/cherrypy/>

```
wget http://download.cherrypy.org/cherrypy/3.1.0/CherryPy-3.1.0.zip
unzip CherryPy-3.1.0.zip
cd CherryPy-3.1.0
chmod +x *
python2.5 setup.py install
python2.5
>>>import cherrypy
>>>
Ctrl+D
```

Flup

Répertoire de téléchargement : <http://www.saddi.com/software/flup/dist/>

```
wget http://www.saddi.com/software/flup/dist/flup-1.0.tar.gz
tar xvf flup-1.0.tar.gz
cd flup-1.0
chmod +x *
python2.5 ez_setup.py -U setuptools
python2.5 setup.py install
python2.5
>>>import flup
>>>
Ctrl+D
```

Pycurl

Répertoire de téléchargement : <http://curl.haxx.se/download/>

```
wget http://curl.haxx.se/download/curl-7.19.0.tar.gz
tar xvf curl-7.19.0.tar.gz
cd curl-7.19.0
./configure
make
make install
```

```
ln -s /usr/local/lib/libcurl.so.4 /usr/lib/libcurl.so.4
easy_install-2.5 pycurl
cd ..
python2.5
>>>import pycurl
>>>
Ctrl+D
```

demjson

Répertoire de téléchargement : <http://deron.meranda.us/python/demjson/dist/>

```
wget http://deron.meranda.us/python/demjson/dist/demjson-1.3.tar.gz
tar xvf demjson-1.3.tar.gz
cd demjson-1.3
chmod +x *
python2.5 setup.py install
python2.5
>>>import demjson
>>>
Ctrl+D
```

b. Téléchargement et installation de Dimdim

Bien que nous soyons sur une Debian, nous allons télécharger le paquet RPM pour CentOS et non les sources du logiciel.

Répertoire de téléchargement : http://sourceforge.net/project/showfiles.php?group_id=176809&package_id=301331

Rendez-vous donc à cette adresse : http://sourceforge.net/project/showfiles.php?group_id=176809&package_id=301331&release_id=644543

afin de télécharger cette archive :

Dimdim_RPM_Package_CentOS_5.2_v4.5.zip

```
unzip Dimdim_RPM_Package_CentOS_5.2_v4.5.zip
apt-get install alien
cd Dimdim_RPM_Package_v4.5/
alien Dimdim-4.5_SF_i386.rpm
dpkg -i dimdim_4.5-2_i386.deb
mv /usr/local/dimdim-4.5/ /usr/local/dimdim
cd /usr/local/dimdim/
tar xvf CobrowsingManager.tar.gz
```

c. Modifications et derniers réglages

Attributions des droits d'exécution

```
chmod +x /usr/local/dimdim/ConferenceServer/apache-tomcat-5.5.17/bin/*
chmod +x /usr/local/dimdim/red5/*
chmod +x /usr/local/dimdim/red5/conf/*
chmod +x /usr/local/dimdim/nginx/sbin/*
chmod +x /usr/local/dimdim/DTP3/Reflector/*
chmod +x /usr/local/dimdim/Mediaserver/mods/*
chmod +x /usr/local/dimdim/Mediaserver/www/pages/*
chmod +x /usr/bin/*
chmod +x /usr/lib/*
chmod +x /usr/local/share/*
chmod +x CobrowsingManager/cob/*
chmod +x StartDimdim.sh
chmod +x StopDimdim.sh
chmod +x Config-ipaddress.pl
```

Modifications des fichiers

/usr/local/dimdim/Mediaserver/mods/toolkit/osconfig.py

Remplacer : `return os.path.dirname('/opt/openoffice.org2.4/program/')`

Par : `return os.path.dirname('/opt/openoffice.org3/program/')`

/usr/local/dimdim/ConferenceServer/apache-tomcat-5.5.17/conf/wrapper.conf

Remplacer : `wrapper.java.command=DIMDIM_JAVA_HOME/bin/java`

Par : `wrapper.java.command=/usr/lib/jvm/java-6-sun-1.6.0.12/bin/java`

/usr/local/dimdim/ConferenceServer/apache-tomcat-5.5.17/conf/server.xml

Remplacer : `<Connector port="DIMDIM_PORT_NUMBER" maxHttpHeaderSize="8192"`

Par : `<Connector address="127.0.0.1" port="40000" maxHttpHeaderSize="8192"`

/usr/local/dimdim/Stopdimdim.sh

Remplacer : `cd /usr/local/dimdim-4.5/ConferenceServer/apache-tomcat-5.5.17/bin/`

Par : `cd /usr/local/dimdim/ConferenceServer/apache-tomcat-5.5.17/bin/`

/usr/local/dimdim/Startdimdim.sh

Remplacer : `cd /usr/local/dimdim-4.5/ConferenceServer/apache-tomcat-5.5.17/bin/`

Par : `cd /usr/local/dimdim/ConferenceServer/apache-tomcat-5.5.17/bin/`

/usr/local/dimdim/Mediaserver/mods/interface.py

Remplacer : `#!/usr/bin/python2`

Par : `#!/usr/bin/python2.5`

```
/usr/local/dimdim/CobrowsingManager/cob/startCobServer.sh
```

```
Remplacer : /usr/local/bin/python2.5
```

```
Par : /usr/bin/python2.5
```

Dernières modifications

```
ln -s /usr/lib/libpcre.so.3 /usr/lib/libpcre.so.0  
cp Mediaserver/mods/spawn-fcgi DTP3/Reflector/spawn-fcgi
```

Vérifions nos binaires d'OpenOffice (pour le partage des PPT):

```
ls -l /usr/bin/so*
```

Nous devrions avoir uniquement cette ligne :

```
lrwxrwxrwx 1 root root 40 nov 27 13:04 /usr/bin/soffice.bin  
->/opt/openoffice.org3/program/soffice.bin
```

Il se peut que cette ligne soit présente :

```
lrwxrwxrwx 1 root root 40 nov 27 13:04 /usr/bin/soffice ->  
/opt/openoffice.or.....
```

Dans ce cas, supprimez cette ligne :

```
rm /usr/bin/soffice
```

Et ajoutez la nouvelle :

```
ln -s /opt/openoffice.org3/program/soffice.bin /usr/bin/soffice.bin
```

Tapez ces commandes pour le partage des PDF

```
cd ThirdPartyPackages/  
tar xvf swftools.tar.gz  
cp swftools/pdf2swf /usr/bin/  
cp swftools/libs/* /usr/lib/  
cp swfdump /usr/bin/  
cp swfextract /usr/bin/  
cp -r swffonts/fonts/ /usr/local/share/swftools/
```

Utilisez enfin le fichier Config-ipaddress.pl :

En réseau local (sans firewall) : ./Config-ipaddress.pl 192.168.1.2 80 (ip privée, port)

A travers un Firewall : ./Config-ipaddress.pl http://url 80 192.168.1.2 (url publique, port, ip privée)

Attention à bien débloquent les ports 80 et 1935 sur votre Firewall.

La configuration principale de Dimdim se fait ensuite dans le fichier suivant :

```
/usr/local/dimdim/ConferenceServer/apache-tomcat-5.5.17/webapps/dimdim/WEB-INF/classes/resources/dimdim.properties
```

2. Mise en place de CAS

Installation de Tomcat :

CAS est une application Java, il est donc nécessaire d'installer un serveur Tomcat sur la machine hébergeant le serveur CAS.

```
apt-get install tomcat5
```

```
echo "JAVA_HOME=/usr/lib/jvm/java-6-sun-1.6.0.12" >> /root/.bashrc
echo "export JAVA_HOME" >> /root/.bashrc
```

Installation du serveur en lui même :

```
apt-get install sun-java6-jdk (ainsi que sun-java-*)
wget http://www.java-sig.org/downloads/cas/cas-server-3.3.2-release.zip
unzip cas-server-3.3.2-release.zip
mv cas-server-3.3.2/ /usr/local/cas-server
cp cas-server/modules/cas-server-webapp-3.3.2.war /var/lib/tomcat5/webapps/cas.war

/etc/init.d/tomcat5 restart
```

Installation d'un serveur LDAP :

```
apt-get install slapd phpldapadmin
modifications du fichier /etc/ldap/slapd.conf et /etc/phpldapadmin/config.php puis ajout de la base avec ce LDIF :
```

```
dn: dc=test,dc=vm
changetype: add
objectclass: top
objectclass: dcObject
objectclass: organization
o: cbc
```

enfin, création des différentes entrées de l'annuaire à partir de phpldapadmin :

```
cn=benjamin,ou=employes,dc=test,dc=vm
cn=julien,ou=employes,dc=test,dc=vm
```

Sécuriser CAS avec du SSL :

création du certificat :

```
keytool -genkey -alias tomcat -keyalg RSA -storepass changeit -keystore
cas.keystore
```

(cas.keystore contient à la fois la clé privée du serveur et le certificat auto-signé qui contient la clé publique).

Pour nom/prénom rentrer l'adresse du serveur.

Ensuite, on exporte le certificat dans un fichier (cas.cert) :

```
keytool -export -alias tomcat -storepass changeit -file cas.cert -keystore cas.keystore
```

(cas-cert ne contient que le certificat, sans la clé privée)

Puis

```
keytool -import -v -trustcacerts -alias tomcat -file cas.cert -keystore $JAVA_HOME/jre/lib/security/cacerts -keypass changit -storepass changeit
```

(on importe la clé publique (le certificat) dans le magasin de confiance)

puis dans le fichier /var/lib/tomcat5/conf/server.xml :

```
<Connector port="8443" maxHttpHeaderSize="8192"
maxThreads="150" minSpareThreads="25" maxSpareThreads="75"
enableLookups="false" disableUploadTimeout="true"
acceptCount="100" scheme="https" secure="true"
clientAuth="false" sslProtocol="TLS" />
```

Intégration de CAS avec LDAP :

```
cp /usr/local/cas-server/modules/cas-server-support-ldap-3.3.2.jar
/var/lib/tomcat5/webapps/cas/WEB-INF/lib/
```

<http://www.springsource.org/ldap>

On télécharge le paquet spring LDAP et on extrait puis copie les deux librairies dans

/var/lib/tomcat5/webapps/cas/WEB-INF/lib/

```
-rw-r--r-- 1 root root 224641 mai  5 13:49 spring-ldap-core-1.3.0.jar
-rw-r--r-- 1 root root   6222 mai  5 13:50 spring-ldap-core-tiger-1.3.0.jar
```

configurer l'annuaire LDAP depuis

/var/lib/tomcat5/webapps/cas/WEB-INF/deployerConfigContext.xml

en ajoutant cette entrée :

```
<bean
  class="org.jasig.cas.adaptors.ldap.BindLdapAuthenticationHandler">
  <property name="filter" value="cn=%u" />
  <property name="searchBase" value="ou=employes,dc=test,dc=vm" />
  <property
    name="contextSource"
    ref="contextSource" />
</bean>
```

modifier /var/lib/tomcat5/webapps/cas/WEB-INF/classes/log4j.properties

remplacer log4j.appender.logfile.File=cas.log
par log4j.appender.logfile.File=/var/log/cas.log

Nous devons ensuite créer le fichier correspondant : *touch /var/log/cas.log*

Le serveur CAS est désormais opérationnel et interfacé avec LDAP.