1244 rue Henri Dunant
07500 Guilherand-Granges
Département Recherche et Développement

51, rue Barthélemy de Laffemas
26901 Valence
Département Informatique

Gilles Mertens
2e année de DUT Informatique



RAPPORT DE STAGE

Sécurisation de postes en Libre-service

Maitre de stage : M. Guillaume Messonnier
Tuteur de stage : Dr Michel Occello

2018 - 2019

1244 rue Henri Dunant
07500 Guilherand-Granges
Département Recherche et Développement

51, rue Barthélemy de Laffemas
26901 Valence
Département Informatique

Gilles Mertens
2e année de DUT Informatique

RAPPORT DE STAGE

Sécurisation de postes en Libre-service

Maitre de stage : M. Guillaume Messonnier
Tuteur de stage : Dr Michel Occello

2018 – 2019

Remerciements

Tout d'abord, je tiens à remercier M. Jean-Philippe Pommel, président directeur général de Decalog, ainsi que M. Guillaume Messonnier, directeur du service Développement, de m'avoir accueilli durant mon stage de fin d'études chez Decalog.

Je remercie mon tuteur de stage en entreprise, M. Guillaume Messonnier ainsi que M. Charles Morel qui ont su m'aider et me conseiller durant ce stage. Je souhaite aussi exprimer ma gratitude à chacun des collaborateurs de Decalog, pour leur accueil chaleureux, leur encadrement ainsi que les conseils qu'ils m'ont donnés.

Je suis également reconnaissant envers le Dr Michel Occello pour avoir veillé sur mon travail tout au long du stage.

Enfin, de manière générale, je tiens aussi à remercier toute l'équipe pédagogique de l'IUT de Valence pour la formation qu'ils m'ont inculqué, ayant largement participé à la réussite de ce stage.

TABLE DES MATIÈRES

Introduction 2
1 Présentation de l'entreprise et du projet 3
1.1 Présentation de Decalog 3
1.1.1 Historique 3
1.1.2 Situation 4
1.1.3 Chiffres clés 4
1.1.4 Organisation 5
1.1.5 L'offre logicielle 7
1.2 Présentation du projet 8
1.2.1 L'Espace Public Numérique 8
1.2.2 Enjeux du stage 11
1.2.3 Cadre de développement 14

2 Missions accomplies 17
2.1 Réalisation du cahier des charges 17
2.1.1 Sécurité et fonctions de base 17
2.1.2 Authentification externe 19
2.2 Pistes de recherche 19
2.2.1 SaMBa avec OpenLDAP* 19
2.2.2 FreeIPA 21
2.2.3 pGina 21
2.3 Développement 23
2.3.1 Authentification 23
2.3.2 Installation/Administration 28

3 Bilan technique et humain 31
3.1 Nouvelles compétences 31
3.1.1 Windows 31
3.1.2 Linux 31
3.1.3 Conception 31
3.1.4 Virtualisation 32
3.2 Relations humaines 32
3.3 Difficultés 33
3.3.1 Prise en main d'Active Directory 33
3.3.2 Recherche des informations 33
3.3.3 C# 34
3.3.4 Inno Setup 34
3.3.5 Conclusion 35
3.4 Évolutions 35

Conclusion 37

Introduction

Étudiant à l'Institut Universitaire de Technologie de Valence en seconde année d'informatique, j'ai été amené à effectuer un stage professionnel et technique de fin de DUT dans une entreprise, d'une durée de 11 semaines.

J'ai pu réaliser ce stage dans le service recherche et développement de la société Decalog. J'ai eu pour mission de créer un prototype d'authentification sur des postes informatiques publics sous Windows*[1], avec une base d'utilisateurs externe.

Le système d'authentification et de sécurisation des ordinateurs en libre-service actuellement en place est devenu vieillissant, difficile à maintenir, et contraignant pour les usagers finaux. Lors d'un précédent stage, Mehdi Mahir, ancien élève de l'IUT en Réseaux et télécommunications, a pu faire des recherches et mettre en place une architecture Active Directory* pour gérer la connexion et les fonctions de sécurité de base sur les postes. Cependant, un problème s'est rapidement posé :

Comment authentifier les utilisateurs possédant un compte Decalog SIGB* dans un Système Active Directory ? Cette problématique a été le sujet de mon stage : l'interconnexion de SaMBa Active Directory* avec un webservice*.

Dans une première partie, je ferai une présentation de l'entreprise Decalog ainsi que la mission qui m'a été confiée. Ensuite, dans une seconde partie, je présenterai les tâches effectuées ainsi que les différents développements qui ont eu lieu. Finalement, dans la dernière partie, je développerai le bilan technique et humain de ce stage, l'état de la mission ainsi que les apports personnels et professionnels.

[1] Tous les mots marqués d'un astérisque sont définis dans le glossaire figurant à la fin du rapport

1 Présentation de l'entreprise et du projet

1.1 Présentation de Decalog

1.1.1 Historique

Decalog est une société française créée en 1987 par Michel Denis et Jacques Kergomard. Elle fut l'une des entreprises pionnières pour l'informatisation des bibliothèques. Cette dernière parvint à se hisser au rang national grâce à son rachat par Jean-Louis Bezin en 2005. Cette évolution repose en grande partie sur un élargissement significatif de la gamme de logiciels proposés ainsi que de la multiplication de leurs fonctionnalités.

Depuis son rachat en décembre 2014, Decalog est dirigée par Jean-Philippe Pommel. En se reposant sur le savoir-faire acquis sur ses progiciels historiques comme Paprika et Carthame, Decalog tend à proposer des solutions basées dans le Cloud, adaptées aux changements actuels du marché de la bibliothéconomie, telles que le logiciel Decalog SIGB (Système intégré de gestion de bibliothèque) ou Decalog Portail (portail documentaire à destination des usagers).

En 2016, pour se diversifier, Decalog absorbe l'équipe Flora de la société lyonnaise Everteam. Cette entreprise est spécialisée dans la création de logiciels destinés aux musées. Parallèlement, le projet Decalog ERMS est lancé, en partenariat avec la société italienne DMCultura et son logiciel Sebina, afin de s'étendre au marché des universités.

L'entreprise, labellisée « société innovante » depuis 2008, investit une part importante de ses budgets dans des projets innovants.

1.1.2 Situation

Le siège social de la société est implanté à Guilherand-Granges, près de Valence, en Ardèche. Dans ces locaux se trouvent le service administratif, le pôle « Recherche et Développement » ainsi que le service après-vente.

Decalog possède également 6 agences sur le territoire national, se répartissant les clients français par secteurs et favorisant de ce fait les relations de proximité. Ainsi, Decalog est présent sur Besançon, Guilherand-Granges (siège social), Nantes, Lyon, Paris et Toulouse.

Les agences, en dehors du siège social, se composent de commerciaux et de formateurs. Le site de Lyon accueille lui aussi une équipe de développement, travaillant à l'heure actuelle uniquement sur la suite de logiciels Flora.

1.1.3 Chiffres clés

Avec un capital de 359 100 € pour une équipe de 75 collaborateurs et plus de 2 400 clients, Decalog poursuit sa progression pour arriver à un chiffre d'affaires de 5.5 millions € en 2016, avec une hausse de 33 % par rapport à 2014.

La part des revenus récurrents (contrats de maintenance et d'hébergement) est aujourd'hui de 63 % et est en constante progression, ce qui est conforme avec la stratégie de commercialisation en mode SaaS* adoptée par Decalog.

La part des services (conduite de projet, migration de données, développements spécifiques, formation) est également en hausse régulière, notamment en raison de l'acquisition de deux activités de services et non d'édition de logiciels.

Cette double augmentation se fait au détriment de la vente de logiciels et matériels qui était de 20 % il y a quelques années et qui est aujourd'hui proche de 10 %. Cet effondrement est logique puisque la nouvelle offre n'impose plus l'acquisition de logiciels et de serveurs par le client, mais un abonnement annuel via des contrats d'hébergement et de maintenance. De plus, la concurrence du

libre et de la gratuité des licences tire les tarifs des licences propriétaires vers le bas.

1.1.4 Organisation

Au 1er janvier 2019, 75 personnes sont salariées. Elles sont réparties dans 6 services :

- Un service « Marketing Produits et Innovation »
- Un service « Marchés Lecture Publique et Centres de Documentation »
- Un service « Marchés Musées, Archives, Écoles et Universités »
- Un service « R&D »
- Un service « Informatique interne, support client et plateformes hébergées »
- Un service « Administratif et RH »

Organisation au 1er anvier 2019

Direction générale - Jean-Philippe POMMEL

Innovation & Marketing Produit

Marketing Produit	
JP Pommel	Marketing

CII/CIR coordination - Guillaume MESSONNIER	
Karl PINEAU	Recherche
Fatma Mcharek	alternant

Marchés Lecture Publique et Info documentaire - Didier Pied

Direction commerciale et marketing	Didier PIED
Région	
Cécile GUILLONI	Commercial Sud-Ouest
Christopher BOUCHET	Commercial Paris-Ouest
Nathalie COSSART	Commercial Est
Stéphanie CACLIN	Avant-vente
Sabrina MAZOT REY	Cial Sédentaire et Com

Opérationnel	PE DEWITTE
Projets	
Christian MALLET	Conduite de Projet
Laurent MOLLA	Conduite de Projet
Conseil et Formation	
Françis CORNET	Conseil & Formation
Florence ROBERT du BOISLOUVEAU	Conseil & Formation
Sylvain ROTIVEL	Conseil & Formation
Demba Thiam	Conseil & Formation
Augustin DE BARMON	Conseil & Formation

Migration & Données	Jean-Pascal BRUNEL
Anne Marchand	Migration
Luc PENDRIE	Migration
Jean-Pascal BRUNEL	Migration

Marchés Musées, Archives, Ecoles et Universités - François-Laurent CONTENAY

Opérationnel	
Projets	
Marc FORYS	Conduite de Projet
Virgile Marengo	Conduite de Projet
Nicolas SCHREIBER	Conduite de Projet
Conseil et Formation	
Chantal JAMOT	Conseil & Formation
Jean-Sébastien DEBORD	Conseil & Formation

Commercial	
Jean-Luc Masson	Commercial sédentaire

localisé à Lyon
localisé à Paris
localisé à Toulouse
localisé à Besançon
localisé à Valence

(italique) contributeur mais dépend d'un autre service

Comité de direction

Développement decalog.net - Guillaume MESSONNIER

Développement decalog.net	Guillaume MESSONNIER
Decalog Portail et Apps	
Fabien ROUDIL	Développement
Noemie ARNODO	Développement
Kevin KLEIN	Développement
Decalog SIGB	
Cyril BEAUME	Analyse & Support SIGB
Angelique CHARRE	Analyse & Support SIGB
Jonathan SAURET	Développement
Xavier DESSEMON	Développement
Swann FRANCHET	Développement
Arnaud RIBA	Développement
Samuel Coreia-Herbaut	Développement
Decalog EPN	
Aurelien Courtial	Développement
Charles MOREL	*Développement*

Développement & support ERM	Guillaume MESSONNIER
Serge BASSET	Développement
Erica MARCO	Développement
(Christian DUCHARME)	*Analyste*

Développement CANOPE	Bruno GUERIN
Frederic DUBOIS	Développement

Développement CARTHAME	
(Jean-Pascal BRUNEL)	*Développement*

Développement & Support FLORA - Philippe RUFFINONI

Support Technique et Fonctionnel	
Nathalie ESCUDIE	Support fonctionnel
David GEORGE	Support technique
Cécile VINOCHE	Support fonctionnel

Développement	
Philippe RUFFINONI	Développement
Stéphane AURAND	Développement
Pierre TEYSSONNIER	Développement
François Xavier JUBAN	Développement
Vincent Rozannes	Développement
Pierre-Edouard de Breuvand	Développement

Développement & Support Portails et Bibliothèques numériques - Pierre-Emmanuel DEWITTE

Développement et support Portails	
Virginie FRACHISSE	Développement
Ali KHAMAR	Développement
Celine HARSTENTEIN	Développement
Thomas GAMBIER	Développement
Lea VANGHELUWE	Chargée de projet
Guillaume VEYRAT	Webdesigner et dev
Dikran KOUCHKARIAN	Développement
Christian DUCHARME	Analyse & Support

Informatique Technique - Support decalog.net - Plateformes hébergées - Yves GRANGE

Support technique	
Thomas LETERRIER	Support & Technique
Olivier RUEL	Support & Technique
Fabien FRANZIL	Support & Technique

Support fonctionnele decalog.net	
Isabelle CHARTIER	Support fonctionnel
Julie NARBONNE	Support fonctionnel
Elsa CHAPELIER	Standard / graphiste
Elise MKADARA	Support fonctionnel
Sonia DUMOULIN	Standard / graphiste

Plateformes hébergées	Julien DAVID
Julien DAVID	Exploitation
Quentin TRAVERSO	Exploitation
Mounir CHAMI	Exploitation

Administratif et RH - Françoise SZOKE

Françoise SZOKE	Achats et RH
Magali POUDEVIGNE	Comptabilité
Catherine FERNANDES	Administratif
Mathilde AUBERT	Administratif
Alain Pairon	Administration Ventes

fig. 1 : Organigramme humain de la société DECALOG

1.1.5 L'offre logicielle

L'objectif que s'est fixé Decalog depuis de nombreuses années est de répondre à l'ensemble des besoins modernes des bibliothèques avec des solutions généralistes, en mode « SaaS » et à coût réduit.

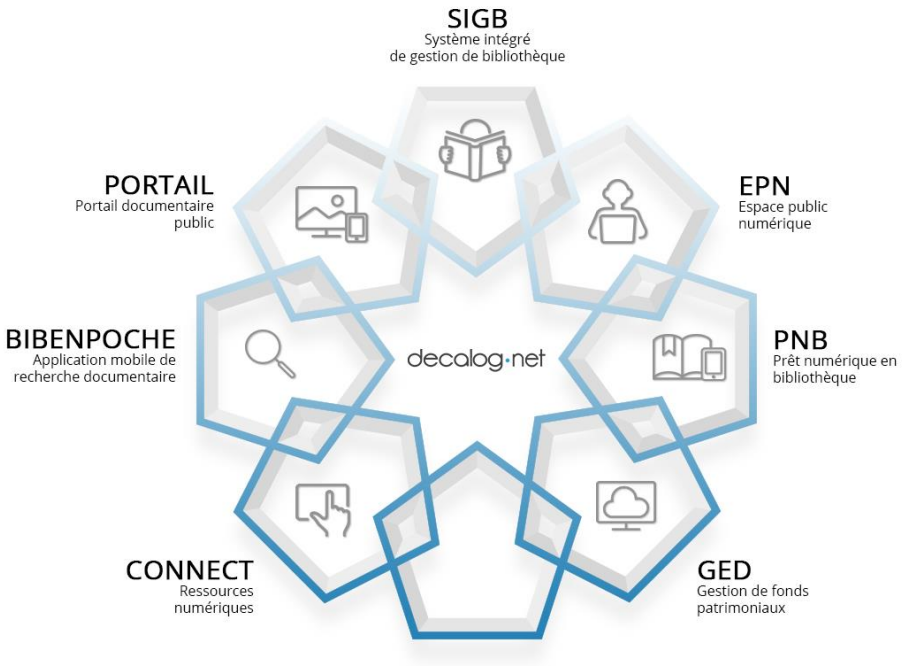


fig. 2 : Portail d'accès Décalog.net aux différents services web (Cloud)

À terme, l'ambition de la société est d'unifier ses produits afin de fournir non pas plusieurs logiciels, mais une plateforme unique dédiée à la bibliothéconomie, proposant de nombreux modules.

Ainsi, les produits développés par Decalog sont de plus en plus orientés vers un SIGB principal, nommé Decalog SIGB, anciennement e-Paprika/e-Carthame. C'est un service web destiné aux bibliothèques, médiathèques indépendantes et bibliothèques en réseau. Le succès de ce dernier en France est relativement important, avec plus de mille sites ayant été équipés, gérant un total de 1.4 million d'abonnés et 14 millions de notices bibliographiques.

Autour de ce service gravite un ensemble de produits. On peut citer Decalog Portails (Essential, Pro, Intégral), offrant un accès depuis le site Internet des bibliothèques à leur catalogue local, leurs ressources numériques, eBooks*,

etc. On peut également évoquer Decalog PNB, permettant l'achat et la mise à disposition de ressources numériques, ou encore Decalog EPN* ayant pour objectif la sécurisation des postes informatiques et tablettes mis à disposition du public au sein des établissements.

Parallèlement à cela, Decalog a récemment agrandi son offre avec la suite Flora afin de s'ouvrir au monde des musées et archives. Celle-ci est composée en 3 parties : service de catalogage et de gestion du fonds, service de prêt et gestion d'abonnements. Archives est quant à lui spécialisé dans la gestion des archives physiques adaptées aux organismes publics, tels que les archives territoriales, nationales ou d'administrations publiques.

Enfin, en accord avec la société italienne Data Management Cultura, Decalog s'occupe de la commercialisation sur le territoire français du SIGB Sebina, destiné aux grandes bibliothèques et réseaux de bibliothèques universitaires.

1.2 Présentation du projet

1.2.1 L'Espace Public Numérique

1.2.1.1 Présentation

Decalog EPN est une solution de sécurisation de postes en libre-service dans les bibliothèques. Elle sert à protéger les utilisateurs d'Internet, en filtrant les contenus selon l'âge des personnes connectées par exemple. Elle permet la conservation de l'historique des utilisateurs ou encore de fournir des services tels que les quotas d'impression.

Le client EPN tournant sur les terminaux permet aussi de brider l'utilisation du poste en limitant le temps de connexion ou éviter que l'utilisateur modifie les paramètres du système ou installe de nouveaux logiciels.

De plus, ce programme nettoie la session à chaque déconnexion de l'utilisateur. Ainsi, les données que les usagers auraient pu produire sont supprimées du poste. Ces données comprennent par exemple l'historique de navigation, les cookies, les identifiants enregistrés dans le navigateur ou encore les documents produits dans des logiciels de traitement de texte.

1.2.1.2 Architecture

L'EPN est composé de deux parties. La première est un serveur sous Linux*, plus précisément sous la distribution Debian, tournant sur un mini-serveur installé dans la bibliothèque. Son rôle est central. Il sépare les différents réseaux informatiques tels que les réseaux professionnel, public et WiFi. Il permet aussi le filtrage des connexions avec un pare-feu et garde les journaux de connexion pour protéger les utilisateurs et le réseau de la bibliothèque de l'Internet.

La seconde partie est le parc de postes en libre-service. Ces machines tournent sous le système d'exploitation* Windows dans ses versions 7, 8 ou 10. Ils communiquent avec le serveur EPN de la bibliothèque sur un réseau informatique dédié.

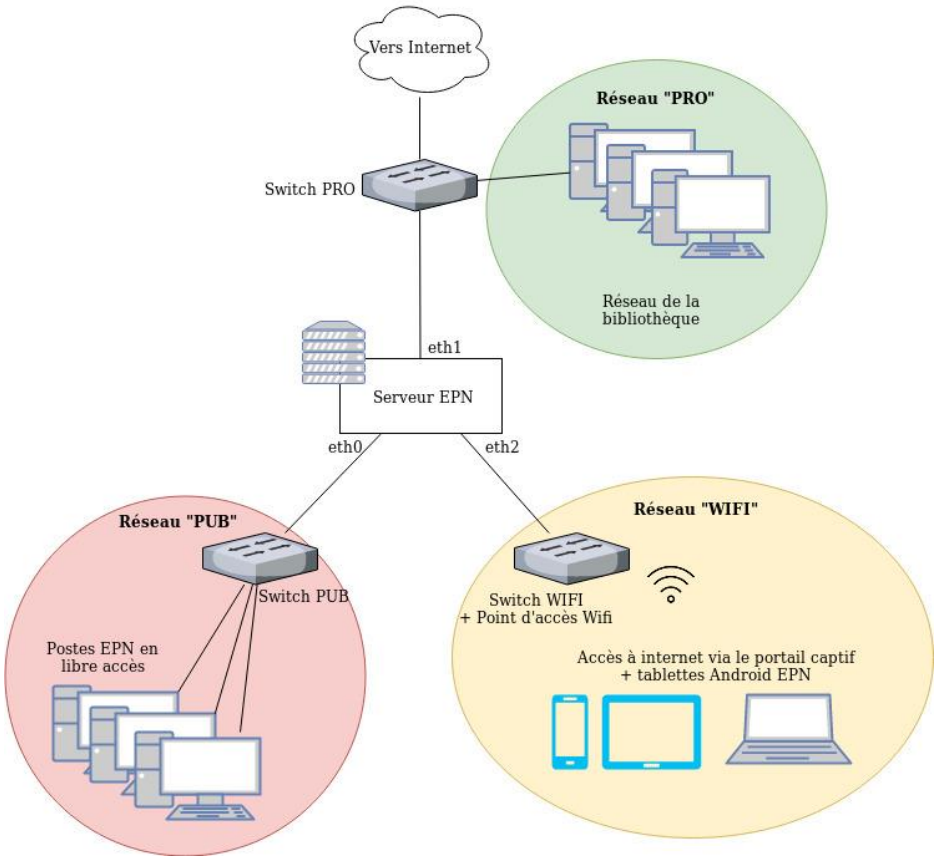


fig. 3 : Architecture du réseau d'une bibliothèque

1.2.1.3 Fonctionnement actuel de la solution EPN

Le serveur EPN sert entre autres à l'authentification des utilisateurs. Lors de la connexion sur un poste public, les identifiants de connexion sont envoyés au

serveur EPN, qui vérifie si l'utilisateur existe soit dans sa base interne soit dans un SIGB pour autoriser la connexion. L'EPN supporte évidemment l'interconnexion avec Decalog SIGB, mais aussi les Systèmes Intégrés de Gestion de Bibliothèques concurrents tels que Orphée, Koha ou encore Aleph. La communication avec ces derniers se fait au travers de webservices.

Les postes, de leur côté, possèdent deux sessions : une session administrateur, pour la maintenance, et une session multimédia. C'est sur la session multimédia que tourne le client EPN. Au démarrage du poste, cette session s'ouvre automatiquement et lance le programme EPN. Ce dernier possède une interface de connexion qui bloque l'utilisation de la session. Il est donc nécessaire pour l'utilisateur d'entrer ses identifiants pour débloquer la session.



fig. 4 : Écran de connexion sur un poste client EPN

Après l'authentification, la session est débloquée et l'utilisateur peut utiliser le poste, dans la limite de ce qu'autorise le client EPN.

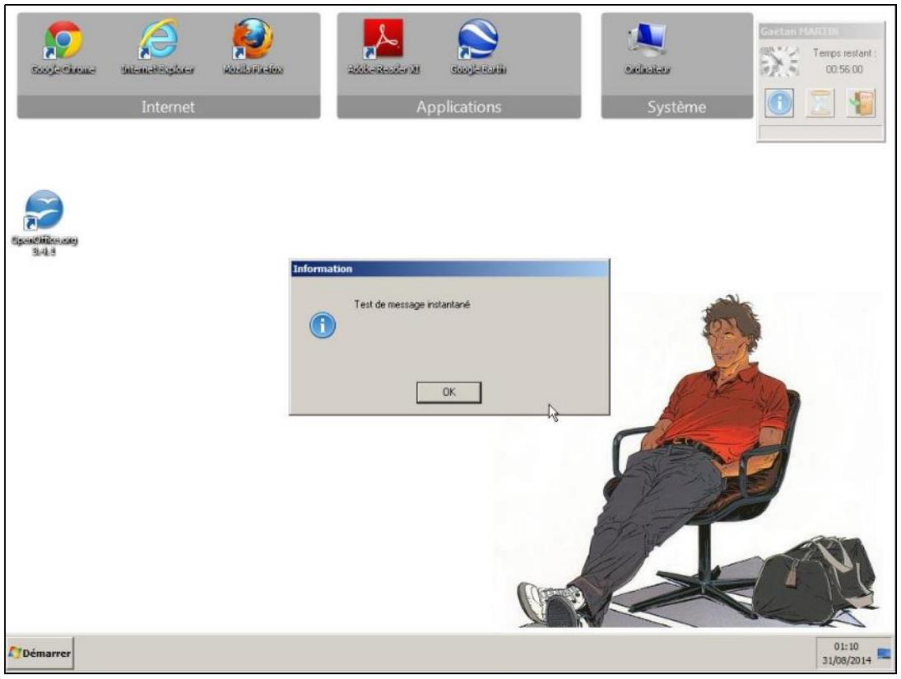


fig. 5 : Bureau d'un poste client EPN

1.2.2 Enjeux du stage

1.2.2.1 Problème avec la solution actuelle

Le système actuel devient difficilement maintenable. En effet, le logiciel EPN n'utilise pas les fonctionnalités de Windows. Le client EPN utilise plutôt des mécanismes spécifiques afin de pouvoir brider le poste. Cependant, ces dernières peuvent être « cassées » par des mises à jour de l'OS par Microsoft. Or ceci est difficilement concevable, car la sécurité des postes ne serait plus assurée.

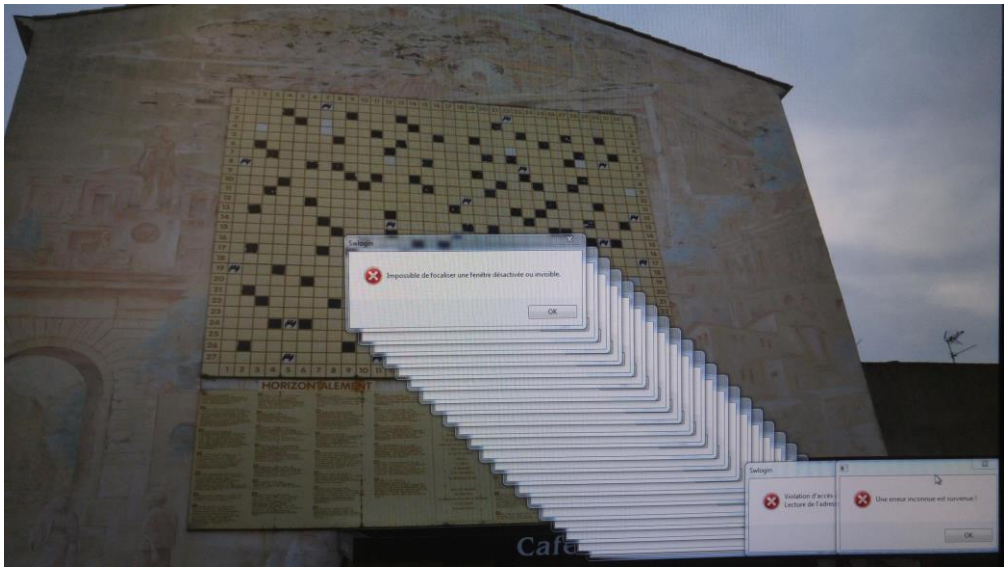


fig. 6 : Bogue du logiciel EPN sur un poste client

Les fichiers produits par l'utilisateur et les données qu'il aurait pu créer dans les différents logiciels (historique de navigation, cookies…) doivent également être détectés et supprimés à la fin de chaque session. En effet tous les utilisateurs se connectent sur une session Windows unique. Il est ainsi nécessaire d'étudier chaque logiciel pour trouver une solution de nettoyage des données. De plus, la mise à jour de ces logiciels pose aussi un problème, car leur fonctionnement peut changer et la solution de sécurisation créée peut ne plus fonctionner.

Ce système limite aussi l'utilisateur qui ne peut plus utiliser certaines fonctions habituelles dans Windows, comme le menu démarrer, ou encore le clic droit. Ces limitations dans un système destiné au grand public dérangent les usagers qui ne retrouvent pas leurs habitudes. Par exemple, le seul moyen de copier-coller du texte est d'utiliser les raccourcis clavier, qui ne sont pas forcément connus de tous les utilisateurs de la bibliothèque.

Le système de connexion actuel est dépassé, l'interface est vieillotte et est non personnalisable.

1.2.2.2 Analyse des besoins

En raison des limitations que posent le fonctionnement actuel, des recherches ont été menées par l'équipe R&D EPN afin de trouver une nouvelle solution de sécurisation durable, proposant une meilleure fiabilité. L'objectif serait d'avoir une session Windows unique pour chaque utilisateur. Ceci permettrait un niveau de sécurité élevé concernant la protection des données de l'utilisateur. Tous les logiciels seraient sécurisés du fait qu'ils sont isolés dans des sessions personnelles. Il en va de même pour les fichiers produits. De plus, elle permettrait de supprimer les contraintes telles que le blocage le clic droit ou le menu démarrer.

Cette solution se base sur une architecture Active Directory. Celle-ci permet de fournir des services centralisés d'authentification et d'administration à un réseau d'ordinateurs utilisant le système Windows. Plus précisément, c'est l'implémentation SaMBa Active Directory qui serait utilisée. En effet Active Directory permettra de pérenniser la solution en se basant un maximum sur les fonctionnalités de base du système d'exploitation.

Cependant, la mise en place de cette solution est bloquée à cause de la partie authentification des utilisateurs. En effet, SaMBa Active Directory utilise ses propres bases de données pour la gestion des comptes des utilisateurs. Or, comme dit précédemment, ces derniers sont gérés de manière externe. Le besoin est donc de relier l'authentification dans un environnement SaMBa AD et une base d'utilisateurs externe.

1.2.2.3 Objectifs

Dans un premier temps, des recherches sur la faisabilité de la solution Active Directory seront faites, en étudiant les différentes possibilités, leurs avantages et leurs inconvénients. Dans un second temps, l'objectif est de créer une solution d'authentification dans un domaine Active Directory SaMBa. Il faut que cette solution soit transparente pour les utilisateurs qui doivent pouvoir se connecter sur les postes, comme ils le feraient « à la maison ». Afin de simplifier la maintenance et l'évolution, la solution doit s'appuyer sur un webservice

d'authentification. Ce dernier pourra être modifié plus tard, sans devoir faire de gros travaux d'adaptation.

1.2.2.4 Enjeux du stage

Les enjeux de ce stage sont importants, en effet, il permettra de remplacer une grosse partie du logiciel EPN qui devient inutile, car remplacé par les solutions de sécurité intégrées dans Windows. Plus durable, cette solution est garantie pour fonctionner avec les nouvelles mises à jour de Windows. De plus, tous les logiciels installés sur la machine deviendraient sécurisés, ce qui permettrait aux bibliothèques d'installer les programmes qu'ils veulent, sans se préoccuper de la surcouche EPN. Pour les utilisateurs finaux, cette solution possède également des avantages, car la sécurité sera moins intrusive lors de l'utilisation du poste. Ils retrouveront par exemple l'usage du menu démarrer ou encore du clic droit.

La solution Active Directory permettra aussi l'utilisation des stratégies de groupe* (ou GPO), qui permettent de configurer les paramètres de tous les postes publics de la bibliothèque de manière centralisée.

Avant mon arrivée, il y avait une grosse incertitude sur la faisabilité de passer à un environnement Active Directory. Mon stage a servi à confirmer que cette solution est techniquement possible et qu'elle correspond exactement aux attentes.

1.2.3 Cadre de développement

1.2.3.1 Outils logiciels

Lors de mon installation j'ai reçu un ordinateur possédant les privilèges administrateur. Le système d'exploitation Windows était installé de base, mais on m'a laissé libre sur le choix du système d'exploitation. J'ai donc installé Linux sur lequel je suis beaucoup plus à l'aise pour travailler.

Pour développer, on m'a donné accès à un serveur de virtualisation* PROXMOX* qui me permettait de créer à volonté des machines virtuelles via des images de système Windows ou Linux. Ce serveur comporte 8 Go de mémoire

vive* ainsi que deux disques durs, un disque classique, ainsi qu'un SSD pour améliorer les performances des machines virtuelles l'utilisant.

J'ai donc pu créer des machines virtuelles sous Windows 7, Windows 8.1, Windows 10 et Debian. L'accès à la machine Debian se faisait en SSH*, étant donné qu'il n'y avait pas d'interface graphique d'installée. Quant aux machines Windows, j'y accédais grâce au protocole SPICE*, qui est un protocole de communication permettant de « rapatrier » l'affichage de la machine virtuelle sur mon poste et d'interagir avec celle-ci en utilisant le clavier et la souris à distance.

Je disposais d'une connexion à Internet pour pouvoir faire des recherches sur mon sujet et chercher de la documentation.

1.2.3.2 Langages de programmation

Lors de la réalisation de la solution j'ai dû faire du développement en C#, pour cela j'ai installé Visual Studio Community. J'ai pu profiter du fait que ma machine était en dual boot* pour installer l'environnement de développement C# sur Windows.

J'ai utilisé Python, avec le Framework Flask* pour développer la partie webservice. Afin de spécifier et documenter cette API, on m'a demandé d'utiliser Swagger* qui permet de construire des webservices avec les principes RESTful*. La conception à l'aide de cet outil permettra à l'équipe qui travaille sur l'EPN de reprendre très facilement mon travail une fois le stage terminé.

Ce stage m'a aussi amené à faire un installateur Windows permettant de déployer ma solution, ainsi que sa configuration, facilement sur un grand nombre de machines. L'installateur a été créé avec Inno Setup, un logiciel Windows qui permet de générer un installateur à partir d'une description des fonctions qu'il doit réaliser. Le langage Pascal* a aussi été utilisé pour faire des pages personnalisées (exemple fig. 14 page 30). J'ai également créé des scripts Power Shell* qui m'ont permis de réaliser des actions qui ne sont pas faisables directement avec Inno Setup.

Finalement, j'ai aussi créé des scripts Bash* destinés à Linux pour installer automatiquement la solution sur le serveur EPN ou faire des sauvegardes et restaurations des données de SaMBa.

1.2.3.3 Méthode agile

J'ai travaillé au sein de l'équipe EPN composée de 4 personnes, mais j'étais seul sur le projet Active Directory. Néanmoins afin de suivre mon avancement, j'ai fait des « Daily meeting » ou « Stand-up meeting » qui permettent d'informer les autres membres de l'équipe sur la progression des différents projets. Ces réunions matinales quotidiennes permettent particulièrement d'expliquer ce sur quoi l'on a travaillé la veille, les problèmes que l'on a rencontrés, et les potentielles solutions qui permettraient de débloquer la situation. Ces explications se terminent par ce que l'on compte faire durant la journée. Ces réunions permettent de voir l'avancement des autres, mais aussi de se rendre compte de son propre avancement. De plus, expliquer ce que l'on fait aux autres permet de rester focalisé sur ses missions. Elles permettent également d'entendre l'avis des autres sur le travail que l'on fait. Ce point de vue externe permet de recevoir des conseils ou des idées que l'on n'aurait pas eues en restant trop focalisé sur ses tâches quotidiennes.

2 Missions accomplies

2.1 Réalisation du cahier des charges

La première semaine de mon stage a été consacrée à la découverte de l'environnement de travail et l'appropriation du contenu du précédent stage. J'ai ensuite commencé à rédiger un cahier des charges simplifié pour, d'une part, confirmer que j'avais bien compris mes missions, et d'autre part informer mon tuteur de stage, monsieur Occello, des missions que je devais réaliser.

Étant donné que le début de mon stage était un travail de recherche, je ne savais pas comment j'allais résoudre la problématique du stage. Dans le cahier des charges, je fais mention de l'utilisation du protocole LDAP*, qui après des recherches approfondies, ne s'est pas faite, au profit d'une autre solution.

Le cahier des charges que j'ai rédigé en début de stage rentre trop dans les détails en présentant la solution LDAP comme la solution à mettre en place alors que ce c'était seulement la première piste de recherche sur laquelle on m'a aiguillé. Néanmoins, ce n'était qu'une piste et non pas le but du stage. La problématique principale ayant été résolue vers les deux tiers de la période de stage, des missions supplémentaires sont venues se greffer à la mission principale et n'ont pas fait l'objet d'un cahier des charges formel.

Les grandes lignes du cahier des charges sont redéveloppées dans cette partie tandis que le cahier des charges original réalisé au début du stage est joint en annexe page II.

2.1.1 Sécurité et fonctions de base

Cette partie détaille les étapes de réappropriation de stage précédent sur le sujet, ainsi que les améliorations qui y ont été apportées.

2.1.1.1 Configuration Windows

Un des buts de passer à une architecture Active Directory est la possibilité de configurer finement le parc de machines à l'aide des stratégies de groupe. Une

recherche et la configuration de ces dernières doit être faite, afin de déterminer celles qui peuvent être utiles pour la sécurité des postes.

Le second intérêt de cette partie sur Windows est le paramétrage des droits sur les fichiers et dossiers. En effet, contrairement à Linux possédant un système de permission simple, Windows vient avec une gestion bien plus fine, mais plus complexe : les droits étendus. Ceux-ci se configurent depuis une interface graphique sur Windows. Lors de l'attribution de ces droits, il est facile de se tromper et créer des brèches de sécurité. Ainsi la documentation permettra de détailler toute la configuration des droits sur ces dossiers. L'intérêt est que les utilisateurs puissent accéder qu'à leurs fichiers personnels en lecture et écriture et à un dossier de partage ou ils ne peuvent que lire le contenu.

2.1.1.2 Administration à distance

Pour des questions d'administration, les employés de la bibliothèque doivent pouvoir lancer des commandes à distance sur les postes, pendant que les utilisateurs sont connectés, pour leur envoyer un message, ou encore forcer la déconnexion.

2.1.1.3 Gestion de l'espace de stockage des utilisateurs.

Les utilisateurs utilisant le système ont droit à un espace de stockage. Cet espace doit être accessible sur tous les postes du réseau auxquels ils se connectent pour lire et modifier leurs fichiers. Cet espace devra être soumis à des limitations afin que les utilisateurs ne dépassent pas la quantité de fichiers qu'ils peuvent sauvegarder sur leurs sessions. Le système doit aussi comporter des espaces de partage où les utilisateurs peuvent lire le contenu et où seuls les administrateurs ont le droit de modifier et écrire de nouveaux fichiers.

2.1.1.4 Sauvegarde et restauration

Le système doit posséder une fonction de sauvegarde qui permettra de dupliquer l'ensemble des paramètres de l'Active Directory, ainsi que les données des utilisateurs de manière automatique. De plus, une fonction de restauration

devra aussi permettre de restaurer le système avec une sauvegarde faite précédemment, lors une erreur humaine ou une panne.

2.1.2 Authentification externe

Cette fonctionnalité est le cœur du projet, la solution doit permettre la connexion d'utilisateurs sur les postes de manière native sous Windows. Elle se base sur SaMBa Active Directory qui devra être relié à un webservice externe pour assurer l'authentification des utilisateurs.

Le but de ce stage est de faire, dans un premier temps, de la recherche sur les solutions qu'il est possible de mettre en place, pour ensuite créer une solution d'authentification. Cette dernière doit être totalement transparente pour l'utilisateur qui peut se connecter aux postes en utilisant ses identifiants classiques. L'équipe EPN a proposé de s'appuyer sur une implémentation du protocole LDAP. L'étude de la faisabilité de cette solution se fera en premier.

2.2 Pistes de recherche

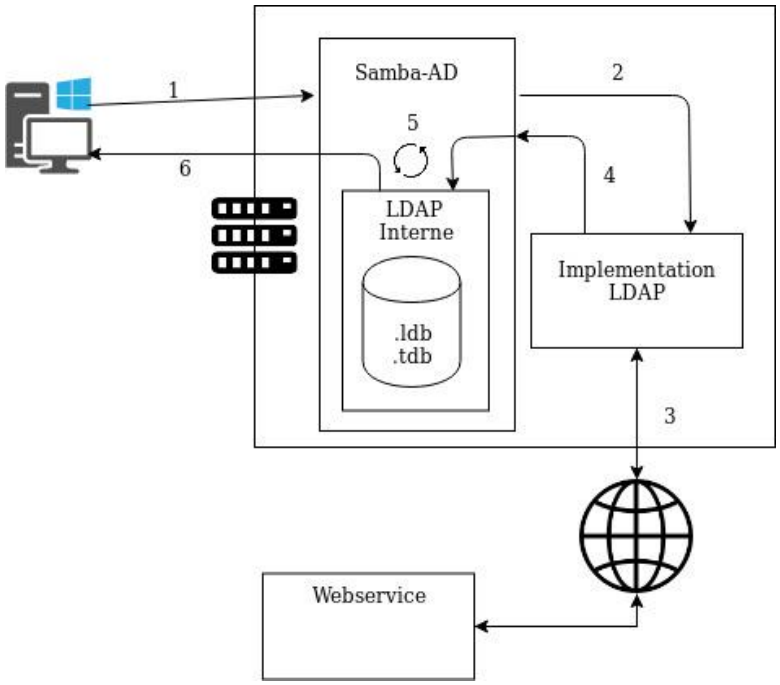
Dans cette partie, je vais exposer les différentes pistes de recherches que j'ai explorées en début de stage.

2.2.1 SaMBa avec OpenLDAP*

Comme dit précédemment, en arrivant en stage l'équipe travaillant sur l'EPN avait déjà une idée de comment utiliser SaMBa avec un service d'authentification externe. C'est donc la piste de recherche que j'ai privilégiée.

J'ai pu, au début du stage, installer un domaine Active Directory, en me basant sur des tutoriels sur Internet et le guide qu'avait fait Mehdi Mahir, un précédent stagiaire. Après avoir un peu plus maitrisé le fonctionnement du logiciel SaMBa, j'ai commencé à m'intéresser à la technologie LDAP, qui est ni plus ni moins qu'un annuaire sous forme numérique. Les LDAPs stockent leurs informations sous forme d'un arbre. Ils permettent ainsi de classifier les utilisateurs dans des groupes pour leur attribuer différents privilèges.

SaMBa utilise un LDAP pour authentifier les usagers ce qui m'a amené à essayer de créer une solution en utilisant un LDAP externe. J'ai tenté de créer une architecture ressemblante à ce schéma.



Légende :

1. Demande de connexion d'un utilisateur
2. Connexion de SaMBa au LDAP externe
3. Connexion du LDAP à un webservice
4. Réponse du LDAP avec les informations de l'utilisateur s'il existe
5. SaMBa met à jour ou crée les données de l'utilisateur dans sa base interne
6. Autorisation de l'authentification ou non de l'utilisateur

fig. 7 : Schéma de l'architecture visée

J'ai essayé de modifier la configuration de SaMBa pour réussir à utiliser un serveur LDAP externe. Après de nombreuses tentatives infructueuses, il s'est avéré que SaMBa, lorsqu'il est utilisé en mode Active Directory, ne supporte tout simplement pas l'utilisation d'un LDAP externe. En effet pour imiter le fonctionnement d'un serveur Windows, SaMBa a besoin d'utiliser un LDAP de manière non standard. C'est pourquoi il ne peut faire usage que de son LDAP interne.

2.2.2 FreeIPA

Ensuite, on m'a proposé de chercher du côté de FreeIPA qui vient comme une surcouche pour SaMBa. Ce programme permet une meilleure intégration des postes sous Linux dans le domaine Active Directory. La partie intéressante de ce logiciel c'est qu'il permettait à première vue d'utiliser un serveur RADIUS* pour authentifier les usagers. En implémentant un serveur RADIUS, il aurait été possible de rediriger les requêtes de connexion des utilisateurs vers un webservice de notre choix.

Néanmoins, l'implémentation et la configuration étant assez complexes, je n'ai pas réussi à faire fonctionner correctement ce système. Ce système allait aussi ajouter beaucoup de complexité au système au vu de toute la configuration qu'il faut faire. Après quelques jours de tests sans pouvoir réussir à utiliser ce logiciel, j'ai pu en parler avec M. Charles Morel qui me supervisait pendant le stage. On m'a demandé de mettre cette solution de côté pour essayer d'en trouver une plus simple.

2.2.3 pGina

Enfin, M. Fabien Franzil, de mon bureau, en s'intéressant un peu à mon projet, a trouvé un logiciel qui permettait de faire de l'authentification personnalisée sur Windows. N'ayant aucune mention à propos d'Active Directory dans la documentation, ce programme ne semblait pas pouvoir répondre à notre problème. On m'a quand même demandé de l'installer et de faire des tests pour voir ce qu'il est possible de faire.

J'ai d'abord testé cette solution avec le plug-in de test intégré qui, selon la documentation, crée une session locale temporaire lors de la connexion. Ensuite, j'ai testé avec le module d'authentification SQL. Ce plug-in permet d'authentifier les utilisateurs avec une base de données SQL. En essayant ce module, j'ai constaté que l'utilisateur était bien authentifié, mais que sa session ne s'ouvrait pas. J'ai alors compris que pGina ne faisait que de la pré-authentification. En effet, une fois l'authentification avec pGina réussie, le programme rend la main à

Windows pour qu'il poursuive la procédure de connexion classique et démarre le bureau de l'utilisateur.

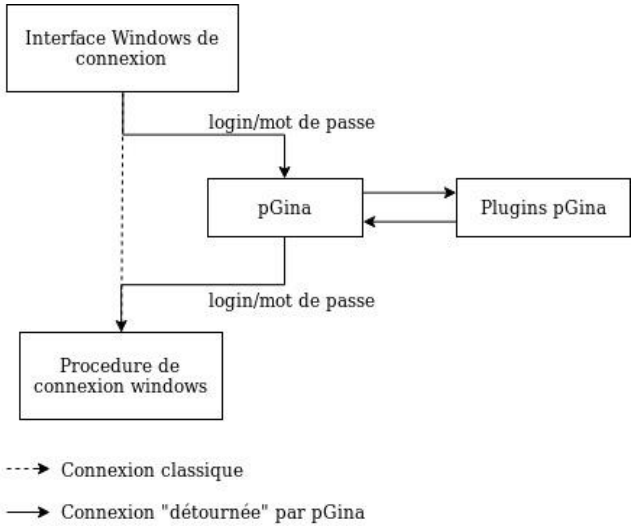


fig. 8 : Schéma du fonctionnement de pGina

Avec ce fonctionnement, il serait théoriquement possible que la connexion à un domaine Active Directory, après l'authentification pGina, fonctionne. Pour tester cette hypothèse, j'ai fait rejoindre le poste au domaine Active Directory. J'ai ensuite créé un utilisateur dans l'Active Directory, en l'enregistrant avec le même mot de passe et identifiant dans la base de données SQL. En effet pour que pGina fonctionne il faut que la session Windows existe et que l'authentification réussisse avec le plug-in SQL.

Le test a été un succès, dans un premier temps, pGina est allé vérifier dans la base de données que l'utilisateur existait. Puis, dans un second temps, a passé le relais à Windows qui s'est connecté sur la session Active Directory possédant le même identifiant et mot de passe. Ce test a prouvé que pGina pouvait être utilisé pour implémenter un mécanisme de pré-authentification.

La partie très intéressante de ce logiciel est qu'il permet d'écrire ses propres plug-ins pour faire un traitement personnalisé de la connexion. C'est la solution qui a été retenue pour répondre à la problématique.

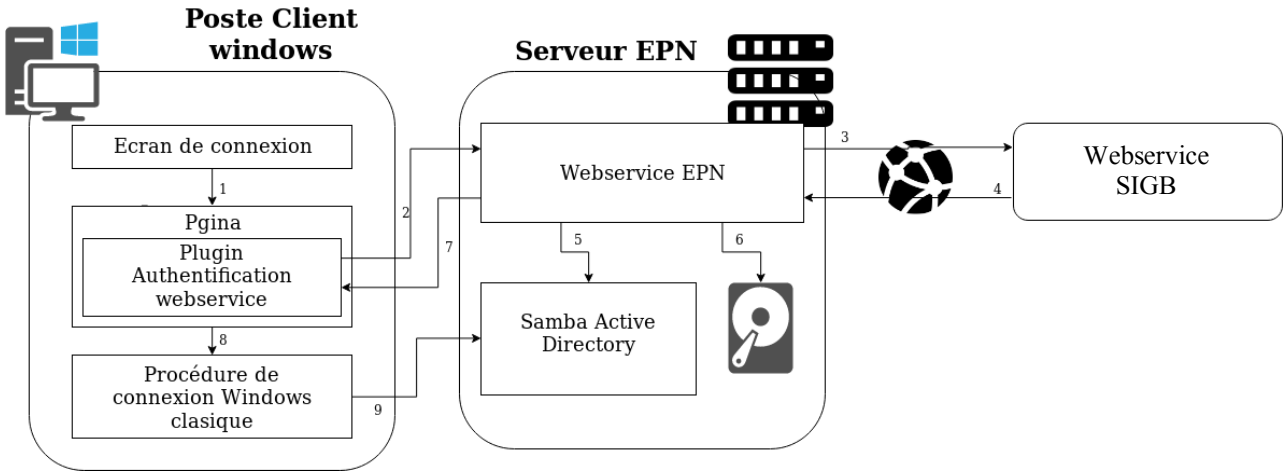
2.3 Développement

2.3.1 Authentification

2.3.1.1 Architecture globale

Afin de mettre en place la solution trouvée avec pGina, il faut authentifier l'utilisateur grâce à un webservice. Une fois cette étape effectuée, il a fallu créer la session de l'utilisateur dans l'Active Directory, afin que Windows puisse s'y connecter de manière transparente à la fin de l'exécution de pGina.

On m'a demandé de faire une architecture avec un webservice servant de mandataire* entre le plug-in pGina et le véritable webservice d'authentification. Ce webservice intermédiaire tourne sur le serveur EPN. Cette architecture permet d'une part de contourner le filtrage effectué sur la connexion des postes lorsqu'ils se connectent à Internet, et d'autre part gérer la création de la session Active Directory dans SaMBa.



Légende :

1. Le couple identifiant/mot de passe est transmis à pGina qui le communique au plug-in réalisé
2. Le plug-in envoie une requête HTTP vers le webservice présent sur l'EPN
3. Le webservice de l'EPN envoie une requête au webservice d'un SIGB.
4. Le webservice répond si l'utilisateur peut se connecter avec des informations telles que sa cotisation ou encore ses quotas.
5. Si le compte SaMBa de l'utilisateur n'existe pas, il est créé, sinon il est mis à jour
6. Les quotas de disque de l'utilisateur sont mis à jour
7. Le webservice EPN renvoie au plug-in le nom d'utilisateur avec lequel le poste doit se connecter, ou un message d'erreur.
8. Les identifiants sont passés à Windows…
9. … qui continue la procédure de connexion normalement en utilisant l'Active Directory

fig. 9 : Schéma de l'architecture réalisée

2.3.1.2 Plug-in pGina

Les plug-ins pGina doivent être écrits en C# pour ensuite être compilés sous forme de Dynamic Link Library* plus connues sous le nom de « .dll ». Le fichier compilé devient une bibliothèque portable, qu'il suffit de placer dans le dossier de plug-in de pGina pour être détecté.

Le fonctionnement de ce plug-in est très simple. Il récupère le couple identifiant/mot de passe entré par l'utilisateur et envoie une requête au

webservice de l'EPN. En réponse, il reçoit l'autorisation de se connecter ou un message d'erreur à afficher dans le cas échéant.

2.3.1.3 Le webservice Python-Flask

Le webservice vient répondre aux requêtes effectuées par le plug-in pGina. Au début, pour tester ma solution, j'ai fait ce webservice en python, en activant l'exécution des programmes python dans le serveur web Apache*. J'ai choisi ce langage, car j'ai pu l'utiliser en travaux pratiques en mathématiques et son utilisation m'a permis d'écrire un code très simple pour tester la solution. Une fois les tests concluants, on m'a demandé de faire la conception de ce webservice avec SWAGGER. Cet outil m'a permis de générer un serveur Python avec le Framework Python-Flask. Ensuite, pour écrire le code du webservice, j'ai réalisé un diagramme d'activité qui m'a permis de valider les étapes et l'algorithme du webservice auprès de mon maitre de stage, et surtout, de faciliter la programmation.

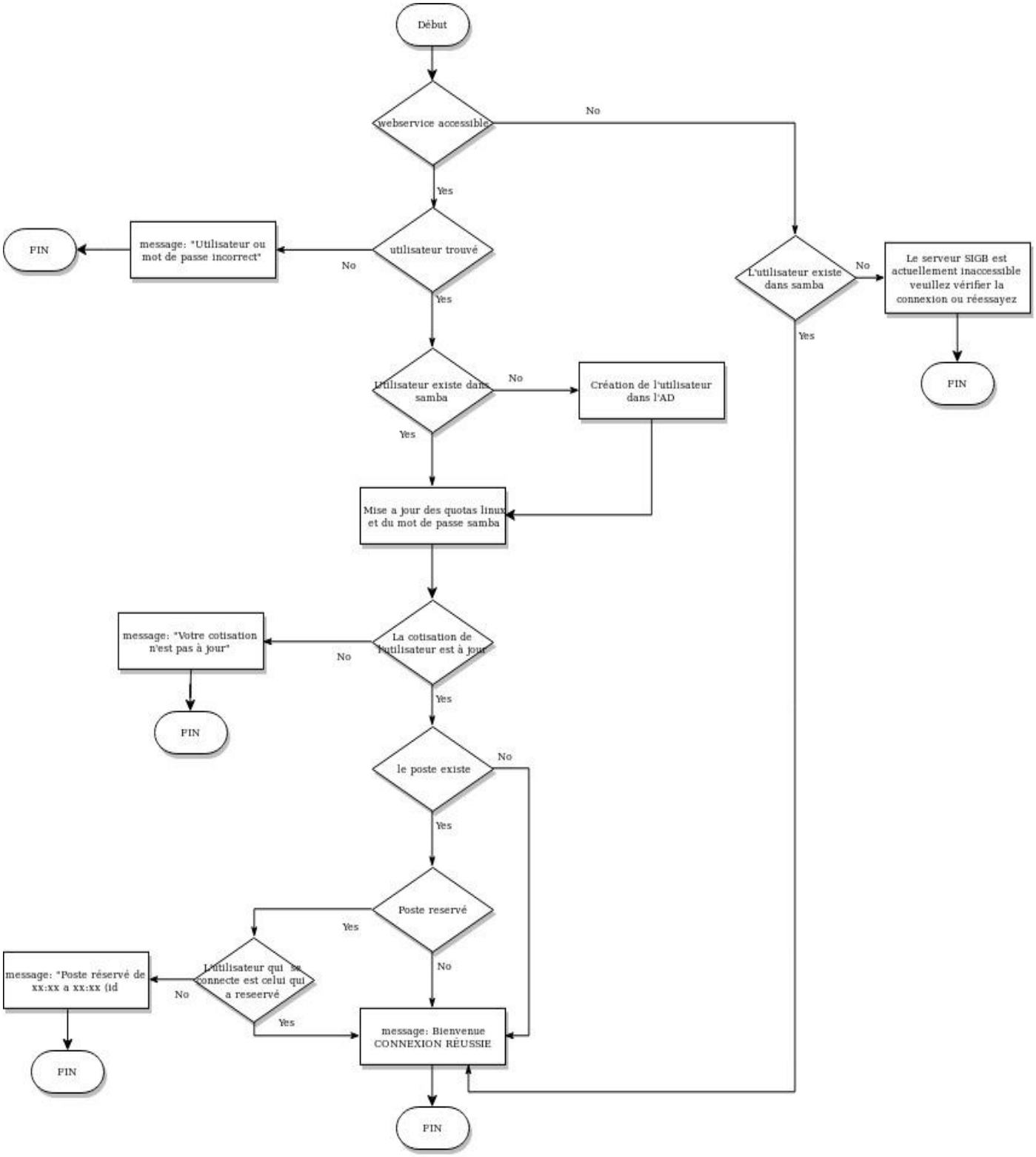


fig. 10 : Diagramme d'activité du webservice.

La communication entre le plug-in et le webservice se fait par requêtes POST. Les données sont organisées sous forme JSON*, encodées en utf-8*.

```
{
  "debug" : true,
  "netbios": "W7-PC1",
  "password" : "caract%C3%A8res_sp%C3%A9ciaux",
  "username" : "utilisateur@mail.fr"
}
```


fig. 11 : Exemple de message reçu par le webservice

À la réception de ce message, le webservice de l'EPN se connecte aux webservices d'authentification et de réservation des terminaux. À partir des données qu'il reçoit, il renvoie un message JSON au plug-in pGina. Celui-ci contient des informations telles que la validité des identifiants, un message d'erreur à afficher en cas d'échec de l'authentification, ainsi que des informations de debug si demandées dans la requête de connexion (paramètre "debug" à True/False en figure 11).


```
{
	"msgToDisplay" : "Utilisateur ou mot de passe faux",
	"smbLogin" : "utilisateur---mail.fr",
	"accessGranted" : false,
	"debugInfo" :
	{
		"userCreationInAd": "skip",
		"terminalFound" : false,
		"terminalRes": false,
		"terminalResForUser": "skip",
		"userSearch" : "not found",
		"userUpdated": false
	}
}
```


fig. 12 : Exemple de message renvoyé au client par le webservice

Lors de la création du webservice il s'est avéré que le caractère arobase était interprété par Windows à l'ouverture de la session. En effet dans des architectures réseau plus complexes, il est possible de se connecter depuis un poste d'un domaine Active Directory, à une session faisant partie d'un autre domaine. Par exemple pour « usr10@decalog.net » Windows va essayer de connecter « usr10 » au domaine « decalog.net », qui n'existe pas dans notre cas. Or, dans Decalog SIGB il est possible pour les utilisateurs d'utiliser leur adresse email. Il a donc été nécessaire de trouver une solution pour contourner ce problème. La résolution fut simple : le nom d'utilisateur dans le serveur SaMBa

aura son caractère « @ » remplacé par les caractères « --- ». Ainsi l'utilisateur « usr10@decalog.net » sera enregistré en tant qu'utilisateur « usr10---decalog.net ». Le plug-in lors de la connexion modifie l'adresse mail pour qu'elle corresponde à ce format avant que Windows reprenne la procédure de connexion.

2.3.2 Installation/Administration

Une partie de mon travail a consisté à automatiser l'installation de la partie serveur et client afin de faciliter son déploiement. D'autres outils ont été développés afin de simplifier la maintenance du serveur.

2.3.2.1 Sauvegarde et restauration

Les scripts de sauvegarde et de restauration ont été écrits en Bash. En effet, c'est le langage le plus simple quand le programme consiste à lancer que des commandes Linux. Ce programme copie de manière sécurisée les fichiers de base de données SaMBa (.tdb, .tlb) en utilisant le logiciel « tdbbackup » qui prend en compte les transactions, pour pouvoir sauvegarder ces fichiers dans un état consistant. Le programme copie ensuite les données des utilisateurs ainsi que les fichiers de profil Windows avec les attributs étendus dans une archive tar avec la compression gzip*. Ces sauvegardes sont datées.

Dans l'archive, les données gardent l'arborescence qu'ils avaient dans le système. Lors de la restauration, le script n'a qu'à copier tout le contenu de l'archive à la racine du système en écrasant les fichiers présents sur le disque.

2.3.2.2 Installateur Serveur EPN

Le script d'installation du serveur EPN sous Linux permet de déployer facilement le serveur EPN. Il permet de configurer la machine et le réseau dans un premier temps, puis, installe les paquets, les configure, et initialise le domaine. À la main, lors de mes quelques installations, toutes ces étapes m'ont nécessité de 15 à 20 minutes. Ce script permet, après avoir éventuellement modifié quelques paramètres, d'installer entièrement le serveur en moins de 3 minutes de manière

totalement automatique. Dans le cas où une étape ne se passerait pas comme prévu, le script s'arrête et il est possible de consulter les journaux pour voir ce qui pose problème.

2.3.2.3 Installateur Poste Client

L'installateur Windows a été réalisé avec Inno Setup. Ce dernier permet d'installer le logiciel pGina ainsi que le plug-in réalisé. Il les configure automatiquement en modifiant les registres Windows. Il permet optionnellement de faire rejoindre le poste dans l'Active Directory grâce à l'adresse IP fournie pendant l'installation.

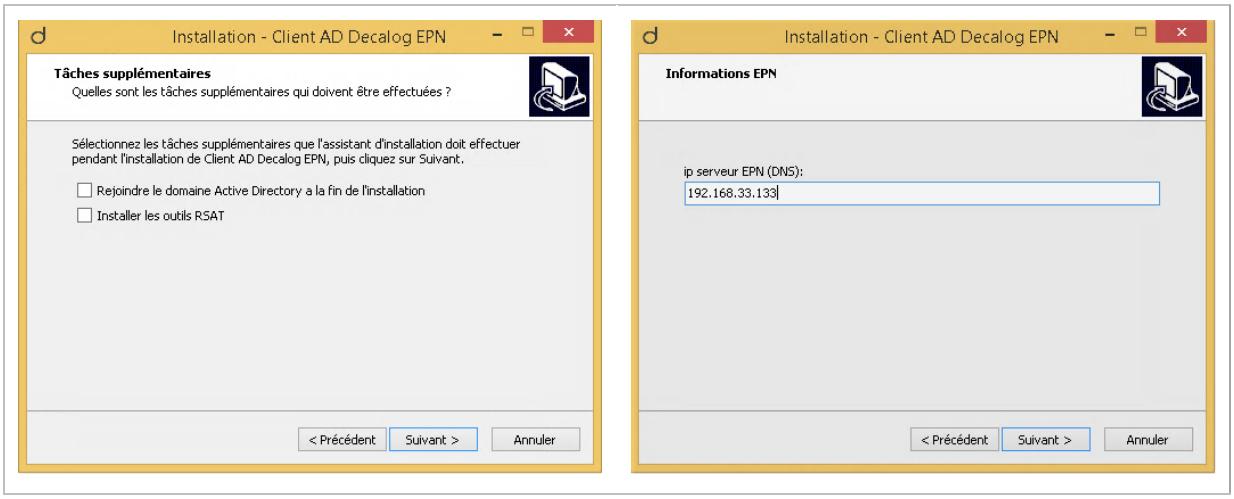


fig. 13: Installateur Windows réalisé avec Inno Setup

Afin de créer l'installateur, il suffit de remplir les sections correspondant aux étapes de l'installation. Il est possible de personnaliser l'installateur grâce à des scripts en Pascal dans la section [code]. J'ai pu par exemple ajouter une page qui demande l'adresse IP du serveur EPN.

```
#define VersionSoft "4.00"
#define NomSetup "setupClientAdEPN"
#define NomProduit "Client AD Decalog EPN"
#define RepGenerationSetup "build"

[Setup]
AppID={{CD7A4AF8-0100-4C07-9806-5EB9A84AF987}
AppName={#NomProduit}
AppVersion={#VersionSoft}
AppPublisher=Decalog

[Tasks]
Name : "join_AD"; Description : "Rejoindre le domaine Active Directory à la fin de l'installation";
Name: "install_RSAT"; Description : "Installer les outils RSAT"; Flags : unchecked restart;

[Files]
; copie des fichiers de l'installateur dans les fichiers communs
Source : "SRC_FILES\ALL\*"; DestDir: "{app}"; Flags : recursesubdirs createallsubdirs
Source : "SRC_FILES\ALL\deps\pGina\*"; DestDir : "{pf}\pGina\"; Flags : recursesubdirs
createallsubdirs
;Fichiers spécifiques W7 32 bits
Source : "SRC_FILES\W7_32\*";  DestDir: "{app}"; Flags : recursesubdirs createallsubdirs; Check : not
  IsWin64; OnlyBelowVersion : 0,6.2

[Run]
; Exécution des scripts
Filename:"{app}\bat\init.bat"; WorkingDir:"{app}\bat"; StatusMsg :"Configuration initiale du système"
Filename :"{app}\deps\pGinaSetup-3.1.8.0.exe"; Parameters :"/Silent"; Flags:RunAsCurrentUser;
StatusMsg : "installation de pGina"

[Registry]
; Création de registres Windows
Root :"HKLM";Subkey:"Software\Decalog\SW"; ValueType:string; ValueName:"adresseEPN"; ValueData :
  "{code:getIp}"; Check: not IsWin64
Root :"HKLM32"; Subkey:"Software\Decalog\SW"; ValueType:string; ValueName:"adresseEPN"; ValueData :
  "{code:getIp}"; Check : IsWin64

[Code]
; Création d'une page personnalisée en utilisant le language Pascal
var
  ipServeurEPN : TInputQueryWizardPage;
procedure InitializeWizard;
begin
  ipServeurEPN := CreateInputQueryPage(wpSelectTasks,'Informations EPN','','');
  ipServeurEPN.Add('ip serveur EPN (DNS) :', false);
  ipServeurEPN.Values[0] := '192.168.33.133';
end;

function getIp(Param : String) : String;
begin
  Result:=ipServeurEPN.Values[0];
end;
```

fig. 14 : Première version de script Inno Setup

3 Bilan technique et humain

3.1 Nouvelles compétences

3.1.1 Windows

Ce stage m'a apporté de nouvelles compétences notamment du côté de l'environnement Windows. En effet, durant ce stage j'ai dû être polyvalent, car les postes client sont sous Windows, et le serveur EPN est sous Linux. J'ai ainsi pu découvrir le concept d'Active Directory pour gérer un parc de postes Windows. Notamment les nombreuses technologies qui la composent, telles que :

- LDAP : un annuaire informatisé
- Kerberos : le système qui sécurise l'authentification entre le poste Windows et le serveur SaMBa
- NTP : la synchronisation du temps entre les machines, qui est essentiel au fonctionnement de Kerberos
- GPO : Les stratégies de groupe dont j'ai parlé partie 2.1.1.1 page 17
- Inno Setup : Le logiciel pour faire des installateurs Windows

3.1.2 Linux

Sous Linux, j'ai pu apprendre à utiliser les quotas disque ou encore l'outil sshfs qui permet de monter un système de fichier à distance. Cela permet d'avoir accès aux fichiers du serveur comme s'il s'agissait de fichiers locaux.

De plus, je me suis amélioré dans la rédaction de scripts Bash grâce à l'outil « shellcheck » permettant de recevoir des conseils lors de l'écriture des scripts.

3.1.3 Conception

Du côté de la conception, j'ai découvert l'outil « SWAGGER », permettant de concevoir et mettre en place un webservice respectant une architecture REST. Avec cet outil, j'ai découvert YAML qui est un langage de représentation des

données qui permet d'enregistrer des données très complexes en restant très lisible par un humain.

3.1.4 Virtualisation

J'ai pu beaucoup utiliser la virtualisation de systèmes d'exploitation durant ce stage. En effet, ces technologies sont devenues incontournables dans les environnements de développement de nos jours. Dans mon cas, la virtualisation est très intéressante, car elle permet de sauvegarder l'état des machines à des moments clés et les restaurer. Cela permet de faire des tests tout en ayant la possibilité de revenir sur une base propre à tout moment. C'est lors de la création des scripts d'installation que cette fonctionnalité a été la plus utile. En effet, même lors d'un problème au milieu de la procédure, je pouvais restaurer ma machine, corriger le problème dans l'installateur et relancer l'installation depuis le début sur une base propre.

3.2 Relations humaines

Tout au long de ce stage, j'ai travaillé dans un bureau de 4 personnes, moi compris. L'ambiance était très bonne. Ce bureau s'occupe de développer et maintenir en partie l'EPN. Il est entouré d'autres bureaux, tels que l'exploitation, le support, et d'autres bureaux de développement tels que SIGB.

J'ai travaillé en autonomie pendant tout le stage. Néanmoins, M. Charles Morel me donnait les lignes directrices pour que je puisse continuer à travailler sur des fonctionnalités utiles ou me donner des pistes.

Dans le bureau dans lequel je travaillais, un projet en autonomie ne signifie pas absence de communication. Grâce aux « Daily meetings », faits tous les matins, on sait en permanence sur quoi les autres travaillent. C'est lors de ces moments-là que l'on a pu me dispenser des conseils qui ont par la suite pu m'aider.

Nous faisions environ une pause le matin et une l'après-midi. Ce sont des périodes où j'ai pu parler d'autre chose que le travail que je faisais. Parfois, j'ai pu

expliquer ce que je faisais à des personnes qui ne travaillent pas sur le même projet que moi. Ça m'a permis d'améliorer ma façon d'expliquer et de devenir plus clair.

3.3 Difficultés

3.3.1 Prise en main d'Active Directory

La prise en main d'Active Directory et ces nombreux concepts ont été assez difficiles au début. En effet, l'administration de machines n'est pas au programme du DUT, de plus, je n'avais aucune connaissance personnelle dans les environnements Windows.

L'Active Directory étant un concept, regroupant des technologies propriétaires Microsoft et des protocoles libres, la compréhension de cet ensemble en une fois a été difficile pour moi. Néanmoins, j'ai pu combler ce manque de connaissances en me renseignant sur le wiki de SaMBa, en faisant des recherches et grâce à M. Morel qui m'aidait quand je ne comprenais pas certains concepts même après m'être renseigné.

De tous les composants, que comporte SaMBa, c'est la partie LDAP qui m'a posé le plus problème. Le concept est en fait assez simple à comprendre. C'est ni plus ni moins qu'un annuaire informatique enregistré sous forme arborescente. L'implémentation a été fastidieuse. Je n'arrivais pas à configurer et à utiliser le logiciel Open LDAP.

De plus s'ajoute le concept de Domaine qui lie Active Directory à LDAP et Kerberos. Finalement, une fois qu'une certaine maitrise est acquise la configuration n'est pas si difficile, mais ça m'a beaucoup ralenti au début.

3.3.2 Recherche des informations

En début de stage, mon rôle a principalement été de faire des recherches sur les moyens de configurer SaMBa et d'utiliser les LDAP. Pendant cette période, je suis souvent tombé sur de la documentation qui ne correspondait pas à la version de SaMBa que j'utilisais. En effet, l'EPN tournant sur Debian 8, j'ai dû

utiliser la version de SaMBa (version 4.2.14) fournie dans les dépôts de cette distribution ce qui n'est pas la dernière version (version 4.10.4). De plus, les différents tutoriels présents sur Internet utilisent souvent des versions plus vieilles que la mienne. Les options de configuration ayant beaucoup changé entre toutes ces versions, je me trouvais souvent face à des incompatibilités.

Ma méconnaissance de l'Active Directory et des LDAP a aussi participé aux différentes difficultés que j'ai rencontrées. C'est seulement le 23 avril, soit un peu plus de 3 semaines après le début du stage, que je confirme que l'architecture avec Open LDAP, que l'on m'a proposé au début était impossible à réaliser, car non compatible avec SaMBa. Une fois ce cap passé, j'ai réussi à vraiment m'améliorer dans mes recherches.

3.3.3 C#

Ce stage m'a demandé d'acquérir de nouvelles connaissances dans de nombreux domaines. Le langage C# en fait partie. Lors du DUT, j'ai pu utiliser ce langage pour faire des logiciels graphiques basiques et des connexions sur des bases de données. Dans le cadre du plug-in, j'ai dû m'adapter à la structure des plug-ins pGina, ce qui n'a pas été très difficile grâce à une documentation claire sur le site du logiciel. Ma plus grande difficulté a été la création de requêtes vers un serveur web, notamment au niveau de l'encodage des caractères qui posait problème au début. J'ai d'abord résolu mes problèmes en encodant tout en base64* ce qui permet de transmettre que des caractères ASCII* qui ne posaient pas problème. À force de faire des recherches, j'ai pu trouver le moyen de tout encoder en utf8 ce qui rend les échanges beaucoup plus lisibles, et ce qui facilite beaucoup le débogage. Un exemple de message encodé en utf-8 est donné figure 11 page 27.

3.3.4 Inno Setup

J'ai aussi passé du temps pour prendre en main le logiciel Inno Setup qui sert à faire des installateurs de logiciels Windows. En effet, la prise en main n'est

pas aisée en raison des nombreuses sections qu'il faut remplir pour faire un installateur. Néanmoins, la documentation est claire, et comporte des exemples qui permettent d'apprendre progressivement à utiliser ce logiciel. La création de la page personnalisée pour demander l'adresse IP du serveur EPN a dû être écrite en Pascal. Je n'avais jamais vu ce langage auparavant. C'est pourquoi j'ai dû beaucoup tâtonner et faire des tests pour me familiariser avec celui-ci.

3.3.5 Conclusion

La plupart des difficultés que j'ai eues ont le plus souvent relevé de la prise en main des logiciels ou de ma méconnaissance du domaine, plutôt que de véritables problèmes techniques. C'est la résolution de ces difficultés qui m'a été le plus bénéfique. En effet, c'est lors des recherches que j'ai le plus appris et que je me suis amélioré.

3.4 Évolutions

Mon stage avait surtout pour but de créer un prototype, pour voir si la sécurisation des postes en passant par Active Directory était faisable et viable. Le travail que j'ai effectué ne va pas directement être déployé chez les clients.

Le travail que j'ai fait va devoir être repris pour y intégrer la surcouche de sécurisation. En effet, mon rôle était de m'occuper seulement de la partie authentification et de la configuration des fonctionnalités de base que l'Active Directory propose. Grâce aux scripts et à la documentation que j'ai réalisée, le projet dispose d'une nouvelle base moins complexe, plus durable et plus maintenable.

Avec le travail que j'ai réalisé, l'EPN pourra bénéficier d'une solution bien plus pérenne. À présent, la solution de sécurisation peut reposer beaucoup plus sur les mécanismes développés par Microsoft et la « surcouche » faite par Decalog sera bien moins intrusive dans le système d'exploitation. De plus, ce projet aura permis de créer une nouvelle base qui permettra de reconstruire

l'EPN de manière bien plus simple, autant pour le développement, que pour la mise à jour.

En effet, à partir de maintenant le client EPN n'aura plus à s'occuper des fonctionnalités gérées par Windows. Par exemple, la sécurisation des fichiers et des logiciels sera en grande partie gérée par le système d'exploitation ce qui permet une réduction de l'ampleur du client EPN. Ces simplifications permettront de diminuer le temps consacré à la maintenance et surtout réduiront la quantité de bogues qui pourraient être apportés par la solution.

Conclusion

Ce stage a eu comme objectif la réalisation d'un prototype de solution d'authentification sur des postes en libre service dans les bibliothèques. Cette solution devrait se baser sur les mécanismes prévus par Microsoft et utiliser une base de données externe pour le stockage des utilisateurs. Elle a pour vocation de remplacer l'ancienne architecture devenue vieillissante du produit Decalog EPN.

La mission a été découpée en trois étapes. La première a consisté à rechercher les solutions pouvant être mises en place pour faire de l'authentification externe dans un environnement Active Directory géré par SaMBa. La seconde étape a été de mettre en place la solution trouvée au travers d'un prototype permettant de prouver la viabilité de la solution. Finalement, le prototype étant finalisé, ce dernier a pu être amélioré par la création de l'installateur automatique pour le poste client ainsi que le serveur EPN.

L'objectif principal du stage a été atteint. Le système de connexion fonctionne entièrement, et est totalement transparent pour l'utilisateur. Il peut se connecter aux postes EPN depuis l'interface d'authentification « classique » de Windows avec ses identifiants habituellement utilisés, c'est-à-dire avec le SIGB.

Ce stage à Decalog a été une très bonne expérience qui a permis de travailler sur un domaine qui m'était inconnu. Il m'a aussi permis d'utiliser des outils et des logiciels variés. J'ai pu travailler sur des choses très diverses ce qui m'a permis de ne jamais me lasser de mon travail.

J'ai également pu travailler avec des méthodes agiles telles que les « Daily meetings » et utiliser des techniques de conception UML vues à l'IUT tel que les diagrammes d'activité. Néanmoins, le stage étant orienté plutôt technique, je n'ai pas eu l'occasion d'appliquer beaucoup de méthodes de conception vues à l'IUT. En revanche, j'ai pu acquérir encore plus de connaissances dans divers domaines durant ces 11 semaines de stage.

Le projet que j'ai réalisé permettra à Decalog de faire une mise à jour du produit EPN afin d'être plus sécurisé pour les utilisateurs, d'une part, et reposer sur une architecture plus durable. Néanmoins, ce stage m'a permis de créer uniquement la partie connexion et mettre en place des règles de sécurité basique. Une surcouche de sécurité va devoir être redéveloppée pour rendre la solution conforme à l'attente des clients.

Je remercie encore Decalog ainsi que l'Équipe EPN de leur confiance et pour m'avoir fait accueilli durant ce stage.

TABLE DES ILLUSTRATIONS

fig. 1 : Organigramme humain de la société DECALOG 6
fig. 2 : Portail d'accès Décalog.net aux différents services web (Cloud)..... 7
fig. 3 : Architecture du réseau d'une bibliothèque 9
fig. 4 : Écran de connexion sur un poste client EPN..... 10
fig. 5 : Bureau d'un poste client EPN 11
fig. 6 : Bogue du logiciel EPN sur un poste client 12
fig. 7 : Schéma de l'architecture visée 20
fig. 8 : Schéma du fonctionnement de pGina..... 22
fig. 9 : Schéma de l'architecture réalisée 24
fig. 10 : Diagramme d'activité du webservice 26
fig. 11 : Exemple de message reçu par le webservice 27
fig. 12 : Exemple de message renvoyé au client par le webservice 27
fig. 13: Installateur Windows réalisé avec Inno Setup..... 29
fig 14 : Première version de script Inno Setup 30

Glossaire

Active Directory : services d'annuaire LDAP pour les systèmes d'exploitation Windows. Permet notamment l'attribution et l'application de stratégies, l'installation de mises à jour ou le déploiement de logiciels.

Apache : serveur HTTP qui fournit des pages web.

ASCII (American Standard Code for Information Interchange) : norme informatique de codage de caractères comprenant : les chiffres arabes de 0 à 9, les lettres minuscules et capitales de A à Z, et quelques symboles mathématiques et de ponctuation.

base64 : codage de l'information utilisant 64 caractères principalement utilisé pour la transmission de messages.

Dual boot : désigne la possibilité d'installer sur un même ordinateur plusieurs systèmes d'exploitation et de choisir l'un d'eux lors du démarrage.

Dynamic Link Library : bibliothèque logicielle qui existent dans Windows et dont les fonctions sont chargées en mémoire par un programme, au besoin, lors de son exécution.

eBooks : livre édité et diffusé en version numérique, aussi connu sous les noms de livre électronique.

EPN (Espace Public Numérique) : suite logicielle permettant la sécurisation de postes destinés à un usage par le public.

Flask : framework open-source de développement web en Python.

Gzip : logiciel libre de compression, basé sur l'algorithme « deflate ».

Json : format de données textuelles dérivé de la notation des objets du langage JavaScript.

LDAP (Lightweight Directory Access Protocol) : protocole permettant l'interrogation et la modification des services d'annuaire numériques.

Linux : Système d'exploitation fondé sur le noyau Linux.

Mémoire vive : mémoire informatique dans laquelle sont stockées les informations en cours de traitement.

OpenLDAP : implémentation libre du protocole LDAP.

Pascal : langage de programmation impératif.

Power Shell : suite logicielle développée par Microsoft qui intègre une interface en ligne de commande, un langage de script nommé PowerShell ainsi qu'un kit de développement.

PROXMOX : une solution de virtualisation libre.

Serveur mandataire : composant logiciel informatique qui joue le rôle d'intermédiaire en se plaçant entre deux hôtes pour faciliter ou surveiller leurs échanges.

RADIUS : protocole client-serveur permettant de centraliser des données d'authentification.

RESTful* : style d'architecture logicielle définissant un ensemble de contraintes à utiliser pour créer des services web.

SaaS (software as a service) : modèle d'exploitation commerciale des logiciels dans lequel ceux-ci sont installés sur des serveurs distants plutôt que sur la machine de l'utilisateur. Les clients ne paient pas de licence d'utilisation pour une version, mais utilisent le service en ligne en payant un abonnement.

SaMBa Active Directory : logiciel d'interopérabilité qui implémente le protocole propriétaire SMB/CIFS de Microsoft Windows. Il sert comme contrôleur de domaine pour Active Directory.

Script Bash (Linux) : programme informatique développé pour automatiser le lancement des commandes dans un interpréteur de commandes Bash.

SIGB (système de gestion de bibliothèque) : logiciel destiné à la gestion informatique des différentes activités nécessaires au fonctionnement d'une bibliothèque.

SPICE (Simple Protocol for Independent Computing Environments) : un protocole de communication, utilisé en général en virtualisation, pour visualiser et gérer à distance un système d'exploitation.
SSH (Secure SHell) : à la fois un programme informatique et un protocole de communication sécurisé. Il permet de lancer des commandes Shell sur une machine à distance.

Stratégies de groupe : ensemble de paramètres définis par les administrateurs de l'Active Directory qui s'appliquent aux utilisateurs et aux postes.

Swagger : framework open source permettant le design et la creétion de webservices respectant les principes RESTfull.

Système d'exploitation : ensemble de programmes qui dirige l'utilisation des ressources d'un ordinateur par des logiciels. Il offre une suite de services généraux facilitant la création de logiciels applicatifs et sert d'intermédiaire entre ces logiciels et le matériel informatique.

utf-8 : codage de caractères informatiques conçu pour coder l'ensemble des caractères de tous les systèmes d'écritures et tous les alphabets du monde.

Virtualisation : consiste à exécuter sur une machine hôte, dans un environnement isolé, des systèmes d'exploitation complets.

Webservice : service sur Internet qui permet l'échange de données entre applications.

Windows : système d'exploitation produit par Microsoft.

Ressources documentaires

Site : samba.org, wiki, https://wiki.samba.org (18 avril 2019)

Site : jrsoftware.org, wiki inno setup, http://www.jrsoftware.org/ishelp/index.php (10 mai 2019)

Site : pgina.org, documentation, https://github.com/pgina/pgina/wiki (26 avril 2019)

Site : ujf-grenoble.fr, wiki, https://www-fourier.ujf-grenoble.fr/informatique/doku.php?id=samba4, (15 avril 2019)

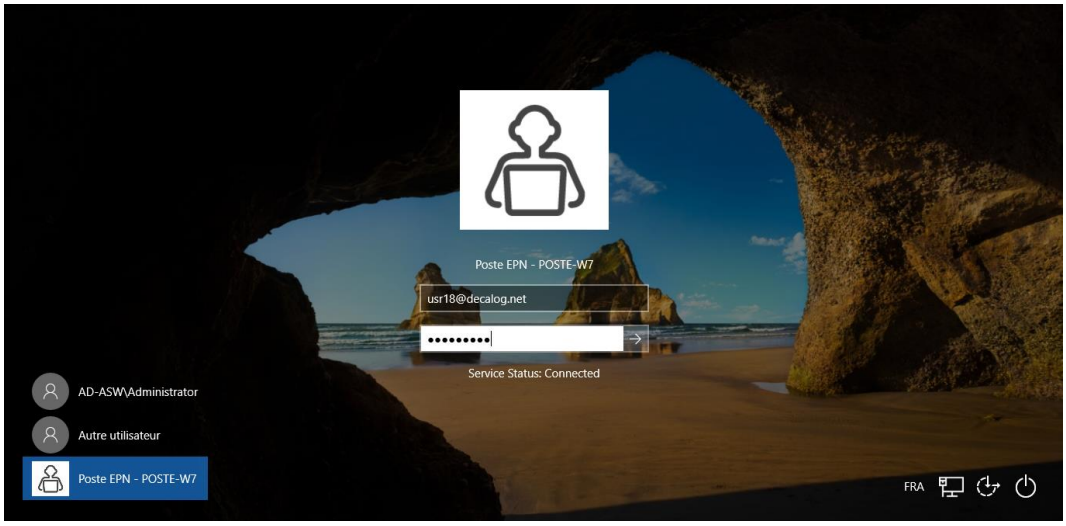
Site : microsoft.com, documentation, https://docs.microsoft.com/fr-fr/powershell/?view=powershell-6 (8 mai 2019)

Article : leereid.wordpress.com, (lee reid), Installing Remote Server Administration Tools (RSAT) for Vista and Windows7, https://leereid.wordpress.com/2009/08/26/installing-remote-server-administration-tools-rsat-for-vista-and-windows-7/ (14 mai 2019)

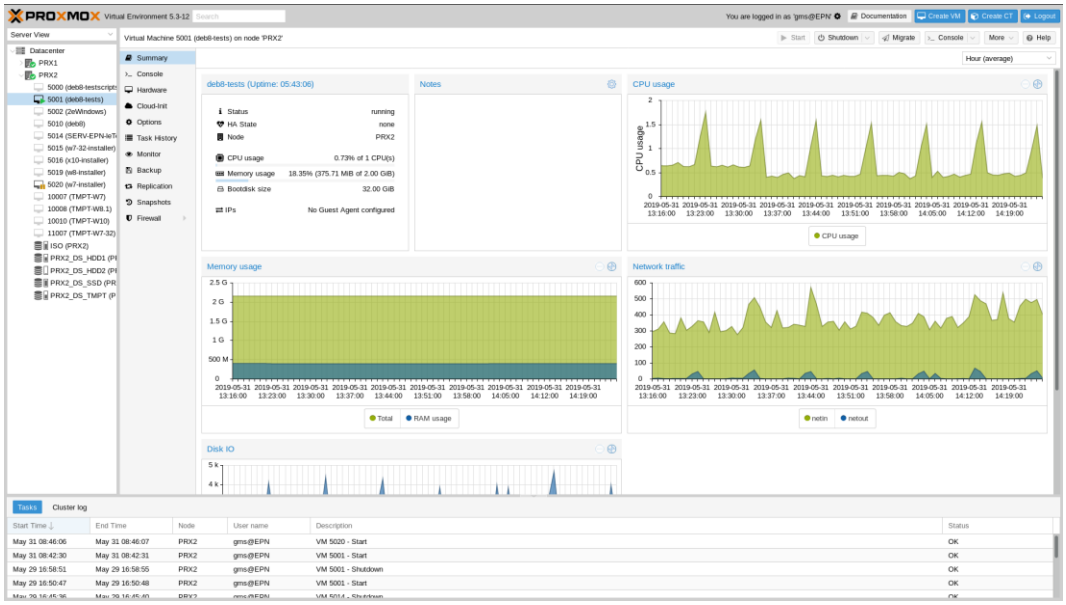
Annexes

LISTE DES ANNEXES

Annexe A – Écran de connexion sur un poste Windows 10 i
Annexe B – Écran de contrôle des machines virtuelles PROXMOX.....i
Annexe C – Cahier des charges réalisé au debut du stage ii



Annexe A – Écran de connexion sur un poste Windows 10
équipé de la solution crée



Annexe B – Écran de contrôle des machines virtuelles PROXMOX

Annexe C – Cahier des charges réalisé au debut du stage

decalog
partageons l'innovation

Espace public numérique DECALOG

Cahier des charges

Stage Avril – Juin 2019



Table des matières

1. **Présentation du projet**

1.1. Contexte

1.2. Situation du projet dans le système existant

1.3. Objectifs du Projet

1.4. Périmètre du Projet

1.5. Références des documents produits

1.6. Cadre technique

2. **Les fonctions**

2.1. Mise en place de la configuration

2.1.1. Gestion des fonctionnalités Windows

2.1.2. Administration à distance

2.1.3. Gestion du stockage des utilisateurs

2.1.4. Sauvegarde et restauration

2.2. Réalisation d'un module d'interconnexion entre SaMBa et un webservice

2.2.1. Interconnexion entre SaMBa et un LDAP externe pour l'authentification

2.2.2. Implémentation d'un serveur LDAP

2.2.3. Connexion à un webservice externe

3. **Glossaire**

1. Présentation du projet

1.1. Contexte

L'EPN[2] est une solution logicielle de sécurisation de postes informatiques à usage libre-service dans les bibliothèques, fonctionnant sous le système d'exploitation Windows*. Actuellement la solution consiste en une session unique, possédant une surcouche logicielle qui bride ou désactive certaines fonctionnalités de Windows, et qui est nettoyée à chaque déconnexion d'un utilisateur. Ce système est très contraignant pour l'utilisateur qui se trouve très limité. Il n'a par exemple plus accès au menu démarrer ou encore au clic droit. De plus, le logiciel est très invasif pour Windows, par exemple, à chaque mise à jour, des fonctionnalisées risquent de ne plus marcher et les antivirus reconnaissent le logiciel comme une menace pour le système.

1.2. Situation du projet dans le système existant

Le but à long terme est de changer le système actuellement en place pour utiliser les outils proposés par Microsoft qui sont nativement intégrés dans Windows. Le projet consiste à faire une maquette mettant en place ces outils afin de pouvoir ensuite juger si cette solution est viable et permet de faire de la sécurisation de postes plus « propre » tout en assurant le niveau de sécurité requis.

1.3. Objectifs du Projet

L'objectif du projet est de chercher et mettre en place une architecture Active Directory* configurée et « augmentée » permettant de s'intégrer avec les outils précédemment développés. Cette architecture permettra de s'intégrer de manière plus « native » sur les postes à sécuriser.

Le logiciel SaMBa AD* va être utilisé pour cela. Le problème est qu'actuellement SaMBa utilise pour la gestion des données un serveur LDAP* interne, or ce serait plutôt une base de données externe accessible par un webservice qui contient les utilisateurs. L'objectif du projet est de trouver et mettre en place un prototype permettant de faire un lien entre SaMBa et un webservice* de manière transparente. Il faut que cette solution soit le moins contraignante possible à l'installation et supportant les mise-à-jour de manière la plus transparente possible, permettant ainsi une maintenance facilitée.

[2] Tous les mots marqués d'un astérisque sont définis dans le glossaire à la fin de ce document.

1.4. Périmètre du Projet

Le projet fini est à but de test. L'objectif est de faire de la recherche sur les solutions pouvant être mises en place et les essayer. Ce sera donc un prototype qui inclura le maximum de fonctionnalités pendant ce stage, afin de déterminer si la solution se basant sur SaMBa AD est viable.

1.5. Références des documents produits

Pour réaliser ce document, je me suis entretenu plusieurs fois avec M. Charles Morel qui s'occupe du développement de l'EPN.

1.6. Cadre technique

Le système SaMBa tournera sous Linux*, plus précisément Debian 8 ou 9. Les fonctions réalisées et les configurations doivent être documentés, afin de pouvoir être reprises lors d'une éventuelle intégration de ce travail dans le système final. Le module LDAP externe sera réalisé en PHP pour facilement s'intégrer avec les webservices.

2. Les fonctions

2.1. Mise en place de la configuration

La première étape du projet est d'installer et configurer SaMBa, ainsi que tous les logiciels tiers permettant d'augmenter ces fonctionnalités. A l'issue de ces recherches, un guide d'installation et de configuration sera produit ainsi que différents scripts pour réaliser certaines taches automatiquement.

2.1.1. Configuration de Windows

Un des principaux buts de la mise en place de l'active directory est la gestion de la configuration d'une flotte de postes Windows. Cette gestion permettra, par exemple, la désactivation de certaines fonctionnalités, ou des restrictions sur l'usage de certains composants Windows. La configuration des stratégies* sera fournie dans le guide d'installation.

2.1.2. Administration à distance

Les administrateurs doivent pouvoir avoir accès aux machines à distance pendant leur utilisation ou non par le public. Cette fonctionnalité doit permettre le lancement de commandes à distance par exemple.

2.1.3. Gestion du stockage des utilisateurs

Les utilisateurs utilisant le système ont droit à un espace de stockage de ficher. Cet espace doit être accessible sur tous les postes du réseau auxquels ils

se connectent pour lire et modifier leurs fichiers. Cet espace doit être privé, c'est-à-dire lisible et modifiable que par le propriétaire des fichiers. Cet espace devra être soumis à des limitations afin que les utilisateurs ne dépassent pas la quantité de fichiers qu'ils peuvent sauvegarder sur leurs sessions. Le système doit aussi comporter des espaces de partage où les utilisateurs peuvent lire le contenu et où seuls les administrateurs ont le droit de modifier et écrire de nouveaux fichiers.

2.1.4. Sauvegarde et restauration

Le système doit posséder une fonction de sauvegarde qui permettra de versionner l'ensemble des paramètres de l'Active Directory, ainsi que les données des utilisateurs de manière automatique. De plus une fonction de restauration devra aussi permettre de restaurer le système avec une sauvegardes faite précédemment, lors une erreur humaine ou une panne.

2.2. Réalisation d'un module d'interconnexion entre SaMBa et un webservice

Cette fonctionnalité est le cœur du projet, la solution doit permettre l'authentification des utilisateurs sur les postes de manière native sous Windows. Elle se base sur SaMBa Active Directory qui devra être connecté à un webservice pour assurer l'authentification des utilisateurs.

2.2.1. Interconnexion entre SaMBa et un LDAP externe pour l'authentification

En premier lieu le but sera de rediriger le système d'authentification de SaMBa pour connecte à un serveur LDAP externe lors de la connexion d'un utilisateur. Pour cela le logiciel Slapd* devra être utilisé pour contenir les informations sur les utilisateurs.

2.2.2. Implémentation d'un serveur LDAP

Dans un second temps une implémentation d'un serveur LDAP devra être réalisée, elle sera faite dans le langage PHP* afin de faciliter l'étape suivante, à savoir, l'interconnexion avec des webservices.

2.2.3. Connexion à un webservice externe

Enfin l'implémentation PHP devra être capable de se connecter à des services externes pour récupérer et transmettre à SaMBa les identifiants, groupes et éventuellement d'autres informations tels que les quotas de disque de l'utilisateur par exemple.

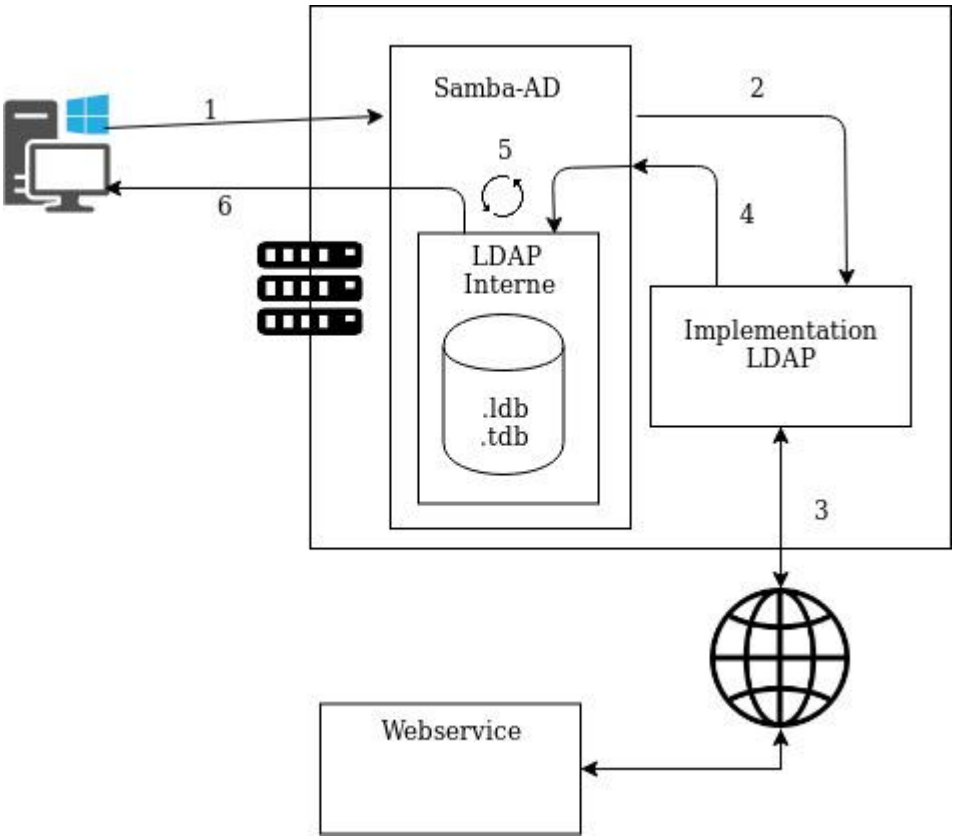


Fig. 1 - Schéma de l'architecture finale

1. Demande de connexion d'un utilisateur
2. Connection de SaMBa au LDAP externe
3. Connexion du LDAP à un webservice
4. Réponse du LDAP avec les informations de l'utilisateur s'il existe
5. SaMBa met à jour ou crée les données de l'utilisateur dans sa base interne
6. Autorisation de l'authentification ou non de l'utilisateur

Glossaire

Active Directory : Services d'annuaire LDAP pour les systèmes d'exploitation Windows. Il permet notamment l'attribution et l'application de stratégies, l'installation de mises à jour ou le déploiement de logiciels.

LDAP : *"Lightweight Directory Access Protocol"*. Protocole permettant l'interrogation et la modification des services d'annuaire numériques.

L'EPN : *"Espace Public Numérique"*. Suite logicielle développée par la société Decalog permettant la sécurisation de postes Windows destinés à un usage par le public.

Linux : Système d'exploitation fondé sur le noyau Linux.

PHP : *"PHP Hypertext Preprocessor".* Langage de programmation libre, principalement utilisé pour produire des pages Web.

SaMBa AD : Logiciel d'interopérabilité qui implémente le protocole propriétaire SMB/CIFS de Microsoft Windows. Il sert comme contrôleur de domaine pour Active Directory.

Slapd : Logiciel libre implémentant le protocole LDAP et maintenu par le projet OpenLDAP.

Stratégie : Ensemble de paramètres définis par les administrateurs de l'active directory qui s'appliquent aux utilisateurs et aux postes.

Webservice : Service sur internet qui permet l'échange de données entre applications.

Windows : Systèmes d'exploitation produit par Microsoft.

Annexe C – Cahier des charges réalisé au debut du stage

Contacts

M. Gilles MERTENS
gilles.mertes@yahoo.fr
Tél. 06 21 82 22 14

Decalog
1244 Rue Henri Dunant, 07500 Guilherand-Granges
Tél. 04 75 81 50 50

M. Guillaume MESSONIER
g.messonnier@decalog.net
Tél. 04 75 81 73 35/Fax 04 75 81 34 50

M. Charles Morel
c.morel@decalog.net
Tél. 04 75 81 50 50

IUT de Valence
51 rue Barthélémy de Laffemas, 26000 Valence
Tél. 04 75 41 88 00

Dr Michel OCCELLO
Michel.Occello@iut-valence.fr
Tél. 04 75 41 88 68

Résumé

Le but de mon stage à Decalog était de créer une méthode d'authentification pour des postes en libre service dans les bibliothèques. En effet, le système de sécurisation actuel est vieillissant et devient difficile à maintenir. Ainsi, il a été décidé de reconstruire l'EPN depuis zéro. La nouvelle solution se base sur le système Windows Active Directory implémenté par SaMBa. Mon projet a été de personnaliser la procédure de connexion afin d'utiliser une base d'utilisateurs externe plutôt que la base de données interne à SaMBa.

Afin de mener à bien ce projet, des recherches sur les solutions envisageables pour répondre au problème ont d'abord été faites. Le logiciel pGina a été choisi pour sa capacité à faire de la pré-authentification personnalisée grâce à des plug-ins écrits en C#. Ce logiciel tourne sur les postes clients Windows à sécuriser. Il communique avec un webservice écrit en Python tournant sur un mini-serveur dans la bibliothèque.

Le projet est une réussite, la problématique de l'authentification externe des utilisateurs a été résolue. La solution est de plus totalement transparente pour l'utilisateur. Des installateurs automatiques ont été réalisés en plus pour faciliter le déploiement de la solution.

Abstract

The aim of my project in Decalog is to create an authentication method for public computers in libraries. The current EPN security system is becoming old and maintaining this software is more and more difficult. That's why the decision to remake the EPN solution from scratch was made. This new solution is based on Microsoft Active Directory architecture with SaMBa software. My part of the project was to customise the login process to fetch an external user database instead of the classical SaMBa internal one.

In order to complete this solution, I made researches about solutions witch could be used. Finally the choice to use a Windows pre-authentication software called « pGina » was made. In order to customise the authentication process I wrote a pGina plugin in C# witch runs on the public computers. This plugin communicates with a webservice written in Python that runs on a mini-server in the library.

At this stage, the project is fully finished. The external authentication is working as expected, and it's totally transparent for the final users. The solution is even enhanced by installers for the client and the server, to deploy the solution very easily and in a small time amount.