Smals
ICT for society

E-mail Address Reliability

Maintenir la qualité des adresses e-mail

Vandy Berten
Isabelle Boydens

Décembre 2013

Deliverable 2013/TRIM4/01
@SmalsResearch
http://www.smalsresearch.be

www.smals.be

Management Summary (Français)

Longtemps considérées comme des données marginales dans les bases de données de l'e-government, les adresses e-mails revêtent de nos jours des enjeux stratégiques dans le cadre de la dématérialisation de l'information et des échanges entre partenaires. Une étude en vue d'en évaluer et d'en maintenir la qualité s'avérait dès lors essentielle.

En première approche, la gestion de la qualité des adresses e-mail dans une base de donnée peut sembler un problème simple. Mais dès que l'on s'intéresse de façon plus approfondie à la question, on peut avoir l'impression que la gestion des e-mails est une tâche très complexe et qu'il est impossible de maintenir une base de données de qualité décente. Heureusement, la réalité n'est pas aussi sombre. Si la tâche est effectivement complexe, à la suite entre autres d'une large part d'incertitude et d'un manque de déterminisme à de nombreux égards, les adresses e-mail ont le gros avantage par rapport à d'autres données, telles que les adresses postales ou les numéros de téléphone, qu'il est possible dans une large mesure d'en déterminer l'existence, et de s'assurer de leur actualité, sans se déplacer ni décrocher son téléphone. Ces étapes peuvent se réaliser de manière semi-automatique, d'où un return on investment très important qui découle de l'amélioration de la qualité des adresses email. Par ailleurs, prendre des mesures efficaces en vaut la chandelle : nombre de projets ont montré un intérêt notable, qu'il soit financier ou organisationnel, d'améliorer la qualité des adresses e-mail dans la base de données.

La principale raison de la nécessité d'une gestion professionnelle est la grande dégressivité de la qualité des adresses e-mail dans le temps. En effet, nos analyses ont montré que la volatilité des usages fait qu'à peine la moitié des adresses e-mail fournies il y a une dizaine d'années sont encore valables aujourd'hui. Il est donc nécessaire de mettre en place une stratégie de gestion efficace, avec un suivi historique des évènements et une structure organisationnelle à même de maintenir à jour les listes d'adresses e-mail. Il va de soi que cela ne peut se faire qu'avec la participation active des utilisateurs. Il faudra donc mettre tout en œuvre pour qu'ils aient l'envie ou le besoin de renseigner tout changement.

En matière de syntaxe, nous avons montré dans ce rapport qu'une séparation binaire entre adresse correcte et adresse incorrecte ne permettait pas de capturer toute la subtilité et la diversité des conventions adoptées par les fournisseurs d'adresses e-mail (Gmail, Yahoo, Telenet…). Il est nécessaire, pour éviter d'être soit trop contraignant, soit

trop laxiste, de considérer une catégorie « suspicieuse », permettant d'attirer l'attention de l'utilisateur ou d'un gestionnaire sur une adresse potentiellement incorrecte, et de mettre en place les stratégies de gestion adéquates.

Par ailleurs, toujours en matière de syntaxe, nous avons montré que nombre de fournisseurs d'adresses e-mail adoptaient une syntaxe bien plus simple que celle recommandée par les standards, permettant des contrôles plus précis, sans risque de faux positifs ou négatifs. Considérer ces syntaxes spécifiques, dont nous fournissons quelques exemples en annexe, permet d'améliorer notablement la qualité et la précision des tests.

Dans ce rapport, nous n'avons pas étudié de façon détaillée les aspects légaux liés à l'utilisation des adresses e-mail (législation sur le respect de la vie privée, force probante…). Nous laissons ces aspects à des spécialistes.

La grande majorité des tests proposés dans ce rapport ont été implémentés soit dans un « Proof of Concept » que nous avons développé en interne, soit dans des outils de gestion de qualité de données (Data Quality Tools) et ont montré leur grande efficacité. Cependant, ces résultats et propositions étant tout récents, ils n'ont pas encore pu être implémentés dans un projet d'envergure, mis à part le développement et la mise en production d'une librairie Java novatrice et réutilisable en vue de tester les aspects syntaxiques des adresses e-mail.

Nous avons néanmoins l'intime conviction qu'une gestion professionnelle des adresses e-mail d'entreprises, de clients ou de citoyens permet d'obtenir une qualité de données largement supérieure à celle que l'on peut espérer obtenir pour la collecte d'informations telles que l'adresse postale, le numéro de téléphone ou d'autres données. Il est bien entendu nécessaire que les utilisateurs aient un incitant à mettre à jour leurs données et que toutes les recommandations rassemblées dans ce document soient mises en œuvre. Un ROI très important en découlera.

Management Summary (Nederlands)

E-mailadressen werden lange tijd beschouwd als secundaire gegevens in de databases van het e-government. Vandaag zijn ze echter van strategisch belang in het kader van de dematerialisering van informatie en uitwisselingen tussen partners. Het bleek dan ook fundamenteel om een studie uit te voeren waarbij de kwaliteit van e-mailadressen geëvalueerd werd en nagegaan werd hoe deze kwaliteit behouden kan worden.

Op het eerste gezicht kan het kwaliteitsbeheer van e-mailadressen in databases een eenvoudig probleem lijken. Maar als we de kwestie van naderbij bekijken, kan men de indruk krijgen dat het beheer van e-mails een zeer complexe taak is en dat het onmogelijk is de kwaliteit van een database hoog genoeg te houden. Gelukkig is de werkelijkheid minder pessimistisch. De taak is inderdaad complex als gevolg van bijvoorbeeld onzekerheid en gebrek aan determinisme op vele vlakken, maar e-mailadressen hebben ten opzichte van andere gegevens, zoals postadressen of telefoonnummers, het grote voordeel dat men grotendeels kan nagaan of ze bestaan en dat men zeker is dat ze actueel zijn. Daarvoor moeten we ons niet verplaatsen, en ook niet onze telefoon gebruiken. Deze stappen kunnen semi-automatisch verlopen, vandaar dat een goed beheer van de kwaliteit van e-mailadressen een zeer grote return on investment kan opleveren. Efficiënte maatregelen nemen is overigens de moeite waard: verscheidene projecten hebben grote belangstelling getoond, hetzij op financieel hetzij op organisatorisch vlak, om de kwaliteit van e-mailadressen in de database te verbeteren.

De belangrijkste reden voor een professioneel beheer is de sterke afname van de kwaliteit van de e-mailadressen doorheen de tijd. Onze analyses hebben immers aangetoond dat de snel veranderende gewoontes ervoor gezorgd hebben dat amper de helft van de e-mailadressen die een tiental jaar geleden werd verschaft vandaag de dag nog geldig is. We moeten dus een strategie uitdenken die efficiënt beheer nastreeft, met een historische opvolging van de gebeurtenissen en een organisatorische structuur die de lijst van e-mailadressen up-to-date kan houden. Het spreekt voor zich dat dit enkel mogelijk is als de gebruikers actief deelnemen. Men moet dus alles in het werk stellen opdat zij de zin of behoefte zouden hebben om elke verandering mee te delen.

In dit rapport hebben we, op het vlak van syntaxis, aangetoond dat een binaire scheiding tussen correct en incorrect e-mailadres niet toeliet het volledige subtiele onderscheid en de diversiteit te vatten van de door de

e-mail service providers (cfr. google, yahoo, telenet e.a.) gehanteerde "standaarden". Om te vermijden dat we ofwel te dwingend, ofwel te laks worden, moeten we een "verdachte" categorie aanmaken, waarmee de aandacht van de gebruiker of van een beheerder kan getrokken worden op een mogelijk incorrect adres, en moeten we de geschikte beheersstrategieën uitwerken.

Bovendien hebben we, nog steeds op vlak van syntaxis, aangetoond dat verscheidene e-mail service providers een gemakkelijkere syntaxis hanteren dan aanbevolen door de standaarden. Met behulp van die standaarden kunnen preciezere controles uitgevoerd worden zonder risico op fout-positieve of fout-negatieve uitslagen. Door rekening te houden met deze specifieke syntaxis, waarvan u enkele voorbeelden in de bijlage vindt, kan de kwaliteit en de precisie van de tests aanzienlijk verbeterd worden.

In dit rapport hebben we de wettelijke aspecten in verband met het gebruik van e-mailadressen niet in detail bestudeerd (privacywetgeving, bewijskracht, ...). Dit laten we over aan de specialisten.

De grote meerderheid van de tests voorgesteld in dit rapport werd ofwel ingevoerd in een "Proof of Concept" die we intern hebben ontwikkeld, ofwel in tools die de kwaliteit van gegevens beheren (Data Quality Tools). Deze tests hebben aangetoond zeer efficiënt te zijn. Aangezien deze resultaten en voorstellen zeer recent zijn, konden ze echter nog niet geïmplementeerd worden in een grootschalig project, behalve dan de ontwikkeling en inproductiestelling van een vernieuwende en herbruikbare Java-library die de syntactische aspecten van e-mailadressen moet testen.

We zijn er echter rotsvast van overtuigd dat een professioneel beheer van e-mailadressen van ondernemingen, klanten of burgers voor een opmerkelijk betere gegevenskwaliteit zal zorgen dan de kwaliteit die men hoopt te bereiken bij het verzamelen van informatie zoals postadressen, telefoonnummers of andere gegevens. Het is natuurlijk wel nodig dat de gebruikers aangezet worden om hun gegevens bij te werken en dat alle verzamelde aanbevelingen in dit document uitgevoerd worden. Een zeer grote ROI zal eruit voortvloeien.

Table des matières

1. Introduction

Avec la dématérialisation de l'information et la mise en place croissante de partenariats et synergies entre administrations, employeurs, entreprises et citoyens, la qualité des adresses e-mail devient stratégique. En effet, une bonne gestion de celles-ci peut, dans le cadre de l'*egovernment*, fortement contribuer à la réduction des coûts.

C'est le cas lorsque les adresses e-mail sont utilisées en vue de l'envoi de notifications, après authentification, dans le cadre des recommandés électroniques, par exemple. Si les adresses e-mail sont incorrectes, les notifications ou envois recommandés doivent s'effectuer par voie postale, après traitement éventuel des cas erronés. Cela peut accroître les coûts cumulés sur cinq ans de plusieurs millions d'euros, selon la taille de la base de données. Selon des hypothèses conservatrices, un projet d'évaluation et d'amélioration de la qualité des adresses e-mail rapporte en cinq ans 2,6 fois son coût. Selon des hypothèses prudentes mais moins conservatrices, le projet rapport 6 fois son coût (estimation empirique sur la base d'échantillons et de *benchmarks* issus du secteur privé[1]).

À ces éléments s'ajoutent les gains indirects associés (que l'on trouve également dans le secteur privé, dans le cadre des envois « marketing » [1]) : respect de la législation, service rendu au citoyen et crédibilité dans les campagnes de communication.

Au Danemark, le ROI sur 15 ans du recours aux adresses e-mail entre administration et citoyens est estimé à 250 millions d'euros.

Dans de nombreux pays, (Danemark – le précurseur –, Suède, Norvège, Canada, …), le recours à l'adresse e-mail authentifiée dans le cadre des échanges entre administrations et citoyens se généralise au sein de l'*egovernment*. En 2012, le ROI sur 15 ans d'une telle approche est estimé en Norvège à 1,9 milliard de NOK (environ 250 millions d'euros)[2].

[1]Les *benchmarks* relatifs au secteur privé considèrent qu'une base de données générant en moyenne un taux constant annuel de 5 % de *bounce mails* (message d'erreur à la suite de l'envoi d'un e-mail) est considérée comme saine. http://www.experian.fr/blogs/business-strategies/2012/11/qualite-adresses-email-element-cle-pour-booster-le-retour-sur-investissement-campagnes-marketing-de-fin-dannee/

[2] « An analysis of potential socioeconomic benefits and savings carried out by the Norwegian Government Agency for Financial Management (DFØ) et al. in 2008, showed that increased use of electronic registration will result in considerable savings amounting to a net present value of minimum NOK 400 million. The Government has an ambitious goal for simplification aimed at business and industry. **By the end of 2015, business expenses are to be reduced by NOK 10 billion**. This is very ambitious and will require close collaboration between the public sector and business and industry. **The socio-economic benefit of introducing electronic invoicing based on a standard format for all incoming invoices to state agencies, is estimated to amount to NOK 1.9 billion over 15 years** according to a

Opérationnelle depuis 1971[3], l'adresse e-mail demeure en effet de nos jours un canal incontournable, même si, avant ses 40 ans, en 2011, on prédisait sa fin, et ce en dépit des canaux qu'offrent maintenant les réseaux sociaux et de la volatilité du monde IT.

Largement usitées de nos jours et pour un certain temps encore, les adresses e-mail se caractérisent toutefois par un cumul d'incertitudes : qu'il s'agisse de la volatilité des usages, de la dynamique des noms de domaine ou de la présence de syntaxes non standard.

Cependant, la loi belge relative à la protection de la vie privée à l'égard des traitements de données à caractère personnel (8 décembre 1992[4]) stipule dans son quatrième article que « les données à caractère personnel doivent être […] exactes et, si nécessaire, mises à jour ; toutes les mesures raisonnables doivent être prises pour que les données inexactes ou incomplètes […] soient effacées ou rectifiées. »

Il y a donc une obligation légale de consentir tous les efforts raisonnables pour mettre à jour les données à caractère personnel, dont l'adresse e-mail fait incontestablement partie. Il va sans dire que nombre d'administrations ou d'entreprises ne respectent pas cette obligation.

Cette étude, étayée par plus de dix ans d'expérience en matière de « data quality »[5] (stratégies de gestion sur la base d'indicateurs de qualité, notamment [2]), un ensemble de tests pratiques ainsi qu'un prototype concret, aborde successivement trois points. Les aspects syntaxiques et techniques sont présentés dans le chapitre 2 : après une évocation des éléments constitutifs des adresses e-mail, on y trouve successivement une analyse des modalités de test de validité syntaxique, de contrôle d'existence et de contrôle de consultation de celles-ci. Le chapitre 2 se termine par une analyse des indicateurs quantitatifs de suivi de la qualité des adresses e-mail dans le temps. Sur cette base, les stratégies de bonne gestion des adresses e-mail « on line » et « en batch » sont envisagées dans le cadre d'une organisation adéquate (chapitre 3). Enfin, le chapitre 4 propose un panorama des outils existant sur le marché en vue d'évaluer et d'améliorer la qualité des adresses e-mail. Par rapport à ceux-ci, la présente étude a pu mettre en exergue plusieurs résultats originaux, dont la problématique de la dégressivité dans le temps des adresses e-mails ainsi que l'importance d'une prise en compte des adresses e-mails « incertaines », dans le cadre d'une stratégie opérationnelle globale d'amélioration de ces données.

---

report prepared by the Ministry of Government Administration and Reform (FAD) in 2011. **This is because the process will be less time-consuming for the State, as the recipient of the invoices, and will save time and money spent on postage for the invoicing parties**. The consultancy company Metier carried out a similar analysis for the municipal sector in 2011. This report indicated that the introduction of electronic invoicing within the municipal sector may result in savings of up to NOK 1.4 billion over 15 years ».
http://www.regjeringen.no/upload/FAD/Kampanje/DAN/Regjeringensdigitaliseringsprogram/digit_prg_eng.pdf

[3] http://www.arobase.org/culture/histoire-email.htm

[4] http://www.ejustice.just.fgov.be/cgi_loi/change_lg.pl?language=fr&la=F&table_name=loi&cn=1992120832

[5] Voir « Data Quality Competence Center de Smals » : https://www.smals.be/nl/content/data-quality ou https://www.smals.be/fr/content/data-quality

Si l'on peut trouver de multiples références dans la littérature « marketing » au fait que l'amélioration de la qualité des adresses e-mail est fondamentale, on trouve beaucoup moins d'études techniques approfondies sur la difficulté de sa mise en place efficace. Le travail le plus avancé que nous ayons pu trouver à ce sujet est celui d'un étudiant de l'Université d'Oxford (Kellog College), Jan Hornych, ayant écrit en 2011 son travail de fin d'études intitulé « Verification & Validation Techniques for Email Address Quality Assurance ».

Notons qu'il est parfois suggéré de faciliter la gestion des adresses de citoyens en leur imposant à chacun une adresse officielle, fournie par l'État. Nous présentons en Annexe 7.3 une réflexion à ce sujet, exprimant quelques réserves par rapport à la faisabilité d'une telle démarche.

2. Aspects syntaxiques et techniques

Gérer un carnet personnel d'adresses électroniques peut sembler un problème assez facile. Finalement, on a surtout besoin d'avoir une adresse correcte pour ceux à qui on écrit fréquemment, et pour ces contacts-là, on prend vite conscience de l'obsolescence d'une adresse « e-mail » (ou courriel, pour emprunter un joli terme à nos amis québécois). Mais lorsqu'il s'agit de gérer des listes de dizaines ou de centaines de milliers d'adresses, comme c'est le cas pour les administrations publiques ou dans de nombreuses entreprises, les choses se corsent.

Il existe des méthodes qui permettent de tenir à jour une base de données d'adresses électroniques autant que possible. Il existe également des techniques permettant, lorsqu'une personne renseigne une adresse électronique, d'en vérifier la validité. On peut également mettre en place des stratégies permettant de s'assurer qu'une « boîte aux lettres » électronique est toujours consultée. Malheureusement, de nombreux organismes et sociétés possèdent des listes d'adresses qui n'ont jamais fait l'objet des plus simples vérifications. Jusqu'à présent en effet, l'adresse électronique d'un client ou d'un citoyen était considérée comme une information accessoire et sans intérêt, un peu comme l'a souvent été le fax. Encore aujourd'hui, nombreuses sont les personnes qui doivent fournir un numéro de fax à des sociétés qui n'ont même plus l'appareil pour en envoyer… Mais de nos jours, on se rend compte qu'avoir des listes d'adresses électroniques de mauvaise qualité coûte beaucoup d'argent [1] et nuit tant à la crédibilité qu'à l'efficacité. Il faut donc nettoyer un lourd passé.

Dans la suite de ce chapitre, nous allons dans un premier temps décrire les différents composants d'une adresse e-mail. Nous verrons ensuite quelles sont les règles syntaxiques à respecter, puis nous nous intéresserons à la validation proprement dite, permettant de s'assurer qu'une adresse existe bel et bien. Nous présenterons ensuite deux techniques permettant de s'assurer qu'un e-mail a bien été lu, puis exposerons quelques indicateurs de qualité et statistiques.

Une fois tous les éléments techniques présentés, nous pourrons, dans le chapitre suivant (Chapitre 3), nous en servir pour mettre en place une

série de bonnes pratiques et recommandations à mettre en place dans le but d'optimiser la gestion d'une base de données d'adresses e-mail.

2.1. Décomposition d'une adresse

Une adresse électronique est composée de plusieurs éléments. Prenons comme exemple l'adresse (fictive)

albert.leroy@bxl.mapetitesociete.be

Cette adresse se compose principalement de trois éléments :

1. Un « nom d'utilisateur » (ou username) : « albert.leroy » ;
2. Un « nom de domaine », qui décrit la société qui fournit l'adresse électronique : « bxl.mapetitesociete.be » ;
3. Un « nom de domaine de premier niveau » (ou Top Level Domain, que nous dénommerons TLD), qui est la partie la plus à droite du nom de domaine, avec le dernier point : « .be ».

Le Top Level Domain (TLD) est la partie à l'extrême droite d'une adresse e-mail ou d'un nom de domaine : .be, .com…

Nous allons maintenant parcourir ces éléments dans l'ordre inverse, pour en mettre en avant les difficultés. Mais dans un premier temps, intéressons-nous à quelques considérations que l'on pourrait qualifier de grammaticales. Comme l'indique le tableau en annexe 7.2, seules quelques étapes de la validation sont déterministes : une incertitude latente doublée d'une grande évolutivité caractérise la gestion des adresses e-mail.

Dans la littérature, on parle de vérification quand il s'agit de contrôler des éléments sans s'assurer de l'existence, comme la syntaxe. On parle de validation lorsqu'on vérifie l'existence réelle d'un élément, comme un nom de domaine, ou une adresse. La vérification ne demande donc aucune connexion, tandis que pour la validation, il faudra interroger l'un ou l'autre serveur.

2.2. Vérification syntaxique

La première chose à faire pour s'assurer de l'exactitude d'une adresse électronique est d'en vérifier sa syntaxe, ou son format. Par analogie, la syntaxe d'un code postal belge précise qu'il doit être composé de quatre chiffres. Mais, bien entendu, tout code respectant la syntaxe n'est pas pour autant un code postal : « 1234 » respecte bien la syntaxe d'un code postal, mais ne désigne aucune ville. Il en va de même pour les adresses électroniques, avec, bien évidemment, une syntaxe de loin plus complexe.

Dans la réalité, la plupart des systèmes acceptant des adresses électroniques ne font soit aucun test syntaxique, soit en font mais sont trop permissifs (c'est-à-dire qu'ils acceptent des adresses syntaxiquement incorrectes), ou, au contraire, trop contraignants (c'est-à-dire qu'ils refusent des adresses correctes). C'est que vérifier la syntaxe des adresses est bien plus complexe qu'il n'y paraît.

Il faut certes qu'il y ait un « @ » (arobase), qu'il n'y ait pas d'espace ni de virgule ou de point-virgule. On serait étonné du nombre de personnes qui, par distraction ou intentionnellement, encodent un numéro de téléphone, une adresse postale ou un site web dans le champ destiné aux adresses électroniques.

2.2.1. Syntaxe du nom de domaine

Si l'adresse contient effectivement un « @ », on peut ensuite vérifier la syntaxe du nom de domaine. Dans la pratique, aujourd'hui et dans la plupart des cas, les noms de domaine peuvent contenir des caractères latins simples (non accentués, sans cédilles… en d'autres termes, sans signe diacritique), peu importe la casse (majuscule ou minuscule), des chiffres, des tirets ou des points. Avec quelques contraintes supplémentaires : le tiret, comme le point, doivent toujours être entourés de caractères ou de chiffres de part et d'autre, et ne peuvent par conséquence ni débuter, ni terminer le nom de domaine, ni être consécutifs.

Depuis juin 2013, les noms de domaine .be peuvent contenir des accents.

Mais les choses ne vont plus rester simples longtemps. En effet, les caractères plus génériques sont officiellement acceptés et commencent à se répandre. Par exemple, en Belgique, depuis juin 2013, des noms de domaine accentués sont acceptés pour les adresses « .be »[6]. C'est ce qu'on appelle les « Internationalized Domain Name », ou IDN, que chaque pays doit approuver. La France l'a également fait[7], mais pas les Pays-Bas. Et pour ne pas faire les choses simplement, la liste des caractères acceptés n'est pas la même dans tous les pays : les noms de domaine « .be » acceptent par exemple les caractères þ, ð et ø (Thorn, Eth et le o barré, empruntés à des alphabets scandinaves), ce qui n'est pas le cas des noms en « .fr ».

Cependant, outre des problèmes de sécurité[8], il est peu probable que les sociétés migrent totalement leur nom de domaine vers des domaines accentués, au risque de se voir refuser l'accès à bien des services qui ne seraient pas encore « compatibles IDN ». On peut donc imaginer que l'adresse albert.leroy@bxl.mapetitesociete.be coexistera avec albert.leroy@bxl.mapetitesociété.be pendant encore un moment, en étant synonyme l'une de l'autre.

2.2.2. Syntaxe du nom d'utilisateur

Si la vérification syntaxique d'un nom de domaine risque d'être compliquée à l'avenir, c'est déjà le cas pour la vérification du nom d'utilisateur. En effet, il existe des standards internationaux décrivant le format de la première partie d'une adresse électronique, mais les

[6] http://www.dns.be/fr/idn

[7] http://www.afnic.fr/en/products-and-services/idns-3.html

[8] http://www.zdnet.be/nieuws/149722/politie-waarschuwt-voor-misbruik-speciale-karakters-in-be-domeinen/?utm_source=zd_weekly&utm_medium=newsletter&utm_term=20130607&utm_content=0_art_list&utm_campaign=weekly

principaux fournisseurs (Yahoo, Gmail, Hotmail…) ne les respectent pas. Par exemple, les standards précisent une longue liste de caractères à accepter, dont les caractères accentués, mais aussi des caractères tels que « # », « $ », « * », « / », « ! »… Cependant, la plupart des fournisseurs ne les acceptent pas.

La plupart des fournisseurs d'e-mail n'acceptent pour les adresses qu'une petite partie des caractères standards.

Hotmail, Belgacom ou Telenet n'acceptent que les caractères latins simples (non accentués), les chiffres, le point, le tiret et le tiret bas (*undescore)*. Yahoo y ajoute la contrainte que le nom d'utilisateur ne peut contenir qu'un seul point. Ses adresses doivent de plus contenir entre 4 et 32 caractères.

Gmail a décidé de pousser le non-conformisme encore plus loin. Les tirets et tirets bas ne sont pas acceptés, et les points sont acceptés, mais ignorés. En d'autres termes, l'adresse albert.leroy@gmail.com est un synonyme d'albertleroy@gmail.com. Par ailleurs, le « + » permet d'insérer des commentaires : albert.leroy+blahblah@gmail.com est également synonyme des deux précédentes. De plus, les adresses Gmail doivent contenir entre 6 et 30 caractères, sans compter les points, ni ce qui suit un « + ».

Par ailleurs, il est également possible d'aller plus loin. Si les grands fournisseurs (Gmail, Hotmail, Belgacom…) possèdent leurs propres serveurs de messagerie et peuvent donc les paramétrer à leur guise, beaucoup d'entreprises achètent un nom de domaine avec un hébergement mutualisé, où ils doivent accepter les règles en cours. Ces hébergeurs les plus courants sont OVH, One, mais également Google qui propose des solutions pour entreprises (Google Apps). Pour connaître l'hébergeur d'une adresse électronique, il suffit de faire une « requête MX » (voir Section 2.5.1).

Nous avons pu déduire (par essai et erreur) la syntaxe des adresses hébergées chez l'hébergeur OVH. Google propose également une solution de gestion d'e-mails pour n'importe quel nom de domaine, remportant beaucoup de succès. Dans ce cas, cependant, les règles syntaxiques sont beaucoup moins contraignantes que pour les adresses de Gmail (appartenant à Google) : si les accents sont également interdits, il n'y a pas de longueur minimale, et des caractères tels que « % », « _ », « - » ou « ' » sont autorisés. Le « + » ne l'est par contre pas.

Un résumé de quelques syntaxes spécifiques que nous avons pu identifier est donné en annexe (Section 7.1)

Notre expérience a montré que de nombreuses adresses ont pu être invalidées à partir de listings sur la base de critères spécifiques au nom de domaine, alors que des critères plus généralistes les avaient acceptées.

2.2.3. Longueur maximale d'une adresse e-mail

Différents standards ont imposé des limites à la longueur maximale que pouvait faire une adresse e-mail. En se renseignant rapidement sur le Web, on risque fort de tomber sur des informations incomplètes ou

dépassées. Pour comprendre d'où vient cette confusion, voici un petit historique des limites imposées par les standards.

- La RFC 821[9], qui date de 1982 et définit le protocole SMTP, impose une limite de 64 caractères pour le nom d'utilisateur, et 64 pour le nom de domaine.
- La RFC 5321[10], qui met à jour le protocole SMTP, stipule (section 4.5.3.1) que le nom d'utilisateur fait maximum 64 caractères, et que le nom de domaine en fait 255, ce qui fait, avec le "@", 320 caractères. Cette même contrainte est reprise par la RFC 3696[11]. En cherchant brièvement sur le Web, c'est en général ce que l'on trouve.
- Le problème est que cette même RFC 5321 précise par ailleurs que l'argument de la commande « MAIL: » ou « FROM: » du protocole SMTP peut, lui, faire au maximum 256 caractères. Or l'argument de cette commande est justement l'adresse e-mail de destination ou de l'expéditeur d'un e-mail. Suite à cette « découverte », la RFC a publié un errata[12] indiquant que la taille maximale d'un adresse e-mail était donc de 256 caractères (et 64 pour le nom d'utilisateur)
- Peu après, on a réalisé que la commande « MAIL: » ou « FROM: » n'était pas suivie directement d'une adresse e-mail, mais avait pour format : « MAIL:<adresse@email.com>». D'où un second errata[13], pour indiquer qu'une adresse ne peut faire que 254 caractères (256 moins les « < » et « > »). De ce fait également, le nom d'utilisateur devant faire au moins un caractère, le nom de domaine est limité de facto à 252 caractères.

Il faut cependant remarquer plusieurs choses :

- La longueur maximale est de 64 caractères pour le nom d'utilisateur dans le standard, mais certains fournisseur sont plus contraignants: Gmail limite à 30 caractères, et Yahoo à 32.
- L'organisme qui standardise le protocole SMTP indique 255 comme limite pour le nom de domaine, mais n'a en fait aucune emprise sur cette partie, puisqu'elle est régie par le standard DNS, indépendant du standard SMTP. Or ce standard DNS évolue : par exemple, les accents et autres caractères non latins y sont maintenant acceptés. Par ailleurs, ces standards précisent en effet non seulement que le nom de domaine total ne peut pas excéder 255 caractères, mais que chaque partie ne peut pas faire plus de 64 (63 plus le point).

[9] http://tools.ietf.org/html/rfc821
[10] http://tools.ietf.org/html/rfc5321
[11] http://tools.ietf.org/html/rfc3696
[12] http://www.rfc-editor.org/errata_search.php?rfc=3696&eid=1003
[13] http://www.rfc-editor.org/errata_search.php?rfc=3696&eid=1690

- De plus, l'introduction des accents complique la donne : le nom de domaine académie-française.fr est en fait converti en xn--acadmie-franaise-npb1a.fr (par le format Punycode). Quelle longueur faut-il dès lors considérer ? Nous n'avons pas encore de certitude à l'heure d'écrire ces lignes.

Cette dernière remarque rend la limite des 254 caractères un peu « flottante ». On peut s'attendre à ce que les accents soient encodés tels quels dans une base de données. Si une adresse est acceptée et stockée, en faisant 254 caractères et avec des accents, il y a des chances qu'elle soit ensuite refusée par les serveurs SMTP. Cependant, il est en général préférable d'accepter éventuellement quelques adresses potentiellement incorrectes plutôt que de refuser une adresse correcte.

Remarquons qu'une adresse de 254 caractères est particulièrement longue. En effet, en voici un exemple d'adresse e-mail dont le nom d'utilisateur fait 64 caractères, et le total en fait 254 :

abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuvwxyzabcdefghijkl@abcdefghijklmnopqrstuvwxyz.abcdefghijklmnopqrstuvwxyz.abcdefghijklmnopqrstuvwxyz.abcdefghijklmnopqrstuvwxyz.abcdefghijklmnopqrstuvwxyz.abcdefghijklmnopqrstuvwx.be

Il est clair que la probabilité qu'une personne introduise une adresse correcte d'une telle longueur est très faible. Selon notre expérience, quand un utilisateur encode une chaîne d'une telle longueur, c'est en général qu'il a introduit une adresse postale ou qu'il a fait un mauvais « copier-coller », ce qui est détecté par des tests syntaxiques classiques. À ce stade de l'évolution des standards et de notre point de vue, implémenter dans les tests des vérifications sur la longueur maximale n'apporte pas grand-chose, mais peut être fait facilement :

- Longueur totale maximale : 254 caractères
- Longueur maximale du nom d'utilisateur : 64 caractères
- Longueur maximale de chaque partie du nom de domaine : 63 caractères (sans compter le point).

2.3. Validation du « Top Level Domain »

Vérifier l'existence du nom de domaine de premier niveau (TLD), comme « .be », « .com », ou « .travel » était jusqu'il y a peu relativement simple (et l'est encore dans beaucoup de cas). Il n'existait que plus ou moins 280 TLD, dont la liste, gérée par l'IANA[14] est disponible sur le Web[15] et était relativement statique. Elle ne contenait par ailleurs que des caractères latins simples, sans accents, et pas de chiffres.

[14] Internet Assigned Numbers Authority (http://www.iana.org/)

[15] http://www.iana.org/domains/root/db

Mais deux nouvelles tendances vont prochainement changer la donne, comme c'est le cas pour les noms de domaine.

Les TLD s'élargissent : ils peuvent désormais contenir des caractères « exotiques » ou être personnalisés.

Premièrement, il existe aujourd'hui un certain nombre de nouveaux TLD « exotiques », contenant des caractères non occidentaux : .中國 pour la Chine, .சிங்கப்பூர் pour Singapour, ou encore .الجزائر en Algérie. Remarquez la présence du point à la fin du TLD, puisque l'arabe s'écrit de droite à gauche.

Par ailleurs, la généralisation des TLD (« generic Top Level Domain » ou gTLD) permettra dans le futur d'avoir un TLD plus personnalisé. On attend les TLD « .brussels » et « .vlaanderen » pour l'été 2014[16]. On pourra donc prochainement voir apparaître une adresse de la forme albert.leroy@mapetitesociété.brussels. Il ne sera donc plus possible de consulter une simple liste pour valider le TLD.

2.4. Validation du nom de domaine

Il ne suffit pas à un prétendu nom de domaine d'être syntaxiquement correct et de contenir un TLD valide pour être valide. Le nom de domaine bxl.mapetitesociete.be, par exemple, n'existe pas. Malheureusement, il n'est pas possible de gérer une liste de noms de domaine existants et de les comparer avec celui d'une adresse. Rien que pour le TLD « .be », il y a eu, en moyenne en 2012, plus de 1300 changements par jour[17], incluant nouveaux noms et disparitions, sur un total de 1.300.000. Fin 2012, on a enregistré quotidiennement en moyenne et au niveau mondial plus de 65.000 nouveaux noms de domaine[18], pour un total de 250 millions !

En 2013, 65.000 noms de domaine ont été créés chaque jour.

La seule façon de le savoir est d'interroger les annuaires d'Internet, que l'on nomme Domain Name Servers ou DNS. C'est le mécanisme qui permet de taper « http://www.google.be » plutôt que « http://173.194.77.94 », autrement moins convivial. Mais c'est aussi le mécanisme qui permet, au travers d'une requête dite « MX » (pour Mail eXchange), d'indiquer le serveur de messagerie qui gère les adresses d'un nom de domaine particulier. Par exemple, il nous indiquera que les adresses « @smals.be » sont gérées par un serveur nommé « mailgater.smals.be » ou que le serveur « gmail-smtp-in.l.google.com » gère les e-mails à destination des adresses « @gmail.com ». Ces vérifications peuvent être effectuées soit automatiquement au travers d'un programme spécialisé, soit à la main, avec un outil tel que http://mxtoolbox.com

Notons qu'un nom de domaine peut très bien n'être constitué que du TLD, bien que ce soit rare. La société gérant un TLD peut attribuer des adresses de ce format. C'est ainsi que albert.leroy@be doit en principe être

[16] http://www.dns.be/fr/nouvelles/nouvelles_recentes/a_quand_les_debuts_de_vlaanderen_et_brussels_le_sort_a_decide3

[17] http://www.dnsbelgium.be/library/documents/331_stats2012_noms-de-domaine-et-agents_fr.pdf

[18] http://businesstoday.intoday.in/story/over-6-million-new-website-names-registered-in-oct-dec-2012/1/193958.html

considéré comme syntaxiquement valide, ce qui n'est dans la pratique pas souvent accepté.

2.5. Validation d'une adresse

Dans cette section, nous verrons comment il est (parfois) possible de vérifier qu'une adresse existe vraiment, c'est-à-dire qu'il existe bien un fournisseur de courrier électronique ayant un utilisateur au nom indiqué. Nous allons pour ce faire entrer dans certains détails d'un des protocoles utilisés pour l'envoi de courrier électronique : le protocole SMTP.

2.5.1. Serveur MX et protocole SMTP

Supposons que notre ami Albert veuille ajouter sa sœur Marie-Célestine à son carnet d'adresses, mais qu'il ne soit plus tout à fait sûr de son adresse : il s'agit soit de mariecelestine.leroy@gmail.com, soit de leroy.mariecelestine@gmail.com. La première chose à faire pour valider l'existence d'une adresse électronique (syntaxiquement correcte) est d'en extraire son nom de domaine, puis d'identifier, grâce au service DNS, le nom du serveur responsable des adresses de ce domaine. Ceci peut se faire facilement à l'aide d'une fenêtre DOS sous Windows, ou d'un terminal sous Linux ou Mac OS. Pour identifier, dans notre exemple, le serveur responsable des adresses « gmail.com », on utilisera la commande « nslookup -q=mx gmail.com » (pour Name Server Lookup), qui produira typiquement comme résultat :

```
C:\>nslookup -q=mx gmail.com
[...]
Non-authoritative answer:
gmail.com mail exchanger = 5 gmail-smtp-in.l.google.com.
gmail.com mail exchanger = 10 alt1.gmail-smtp-in.l.google.com.
gmail.com mail exchanger = 20 alt2.gmail-smtp-in.l.google.com.
[...]
```

Ceci nous indique qu'il faut maintenant s'adresser au serveur de messagerie répondant au nom de gmail-smtp-in.l.google.com (les autres devant être utilisés lorsque le premier ne répond pas). On parle aussi de serveur MX, pour Mail eXchange.

Il est aussi possible de recevoir un message d'erreur en tapant cette commande. Cela peut principalement signifier deux choses : soit le nom de domaine n'existe pas, soit il existe mais ne gère pas de courrier électronique. Il se pourrait par exemple qu'il existe un site web

www.mapetitesociete.be, mais qu'il n'existe pas d'adresse @mapetitesociete.be. Si une telle erreur s'est produite, il ne sert à rien d'aller plus loin : l'adresse recherchée n'existe par définition pas.

S'il n'y a pas eu d'erreur, on peut maintenant « parler » à ce serveur, grâce au protocole « SMTP » (Simple Mail Transfer Protocol). Ce protocole est en fait le langage qu'utilisera un programme comme Outlook, Thunderbird, Mail ou le programme de gestion de courrier électronique de votre smartphone. Toujours dans la même fenêtre de commande, à l'aide du programme « telnet », Albert fait « comme si » il était un de ces logiciels et qu'il voulait envoyer un courrier, et effectue les manœuvres suivantes (en rouge, les commandes qu'il tape) :

```
C:\>telnet gmail-smtp-in.l.google.com. 25
Trying 173.194.78.26...
Connected to gmail-smtp-in.l.google.com.
[...]
EHLO bxl.mapetitesociete.be
250-mx.google.com at your service, [91.183.59.xxx]
[...]
MAIL FROM:<albert.leroy@bxl.mapetitesociete.be>
250 2.1.0 OK pn9si600796wjc.42 - gsmtp
RCPT TO:<leroy.mariecelestine@gmail.com>
550-5.1.1 The email account that you tried to reach does not exist.
[...]
RCPT TO:<mariecelestine.leroy@gmail.com>
250 2.1.5 OK pn9si600796wjc.42 - gsmtp
QUIT
221 2.0.0 closing connection pn9si600796wjc.42 - gsmtp
```

Suivant le protocole SMTP, il commence par se « présenter » : il indique quel nom de domaine il gère (commande « EHLO »), puis précise quel est l'expéditeur du courrier (commande « MAIL FROM »), bien que dans notre cas, aucun courrier ne sera réellement envoyé.

On y voit qu'à la première commande « RCPT TO », la réponse du serveur commence par 550, code indiquant que l'adresse n'existe pas. Un message plus verbeux l'explicite ensuite. Par contre, lors de la seconde invocation de la commande, la réponse débute par 250, code indiquant que tout s'est bien déroulé et que la seconde adresse introduite existe bel et bien (il s'agit d'un exemple fictif).

En principe, pour réellement envoyer un e-mail, il aurait fallu, à la place du « QUIT », introduire le contenu (sujet, corps du texte, pièces jointes, ...). Le but de notre ami Albert étant simplement de vérifier l'existence d'une adresse et non d'envoyer un courrier, il s'arrête là et rien n'est envoyé à la destination.

Notons que le texte qui suit le code « 550 » est typiquement ce que l'on va retrouver dans un retour d'e-mail suite à un envoi erroné à une adresse inexistante. Ces mails d'erreur sont généralement appelés « *bounce mail* ». On en distingue deux catégories : les « *hards* », qui représentent des problèmes définitifs (adresse inexistante, nom de domaine non valable...), et les « *softs* », pour les problèmes temporaires (boîte pleine, serveur temporairement indisponible...)

2.5.2. Difficultés

Malheureusement, si l'exemple précédent fonctionne très bien pour vérifier les adresses de Gmail, ce n'est pas toujours aussi facile, et ce pour de nombreuses raisons. Il faut d'abord savoir que le protocole SMTP est très ancien : il date du début des années 80, soit bien avant l'invention du web ! À cette époque, les problèmes de sécurité et de spam n'étaient pas ce qu'ils sont aujourd'hui et n'ont que très peu été pris en compte. Cependant, ce protocole est tellement répandu qu'il est très difficile d'imposer un nouveau standard qui comblerait ses lacunes. De nombreux gestionnaires ont dès lors choisi de faire évoluer leurs serveurs de façon non standard, entraînant des comportements très différents et difficiles à interpréter. Quelques explications :

Dans la plupart des cas, on peut envoyer un e-mail à partir de n'importe quel expéditeur, sans le moindre contrôle.

- Dans l'exemple ci-dessus, l'expéditeur mentionné utilise un nom de domaine qui n'existe pas (bxl.mapetitesociete.be), sans que ça ne pose le moindre problème aux serveurs de Gmail. En fait, dans la plupart des cas, on peut envoyer un e-mail avec n'importe quel expéditeur, sans le moindre contrôle. Certains serveurs font cependant plus de vérifications.
- En temps normal, un programme d'envoi de mail ne contacte pas directement le serveur « SMTP » de la destination : il contacte typiquement le serveur SMTP de son FAI (fournisseur d'accès à Internet, tel que Belgacom, Telenet...), de son entreprise ou de son université, qui contacte lui-même le serveur de la destination. De la même façon que si je veux envoyer une lettre dans une ville voisine, je ne dois pas la déposer dans une boîte de la ville de destination, mais bien de ma ville, et c'est la poste qui se chargera de l'acheminement. Certains serveurs vérifient soit que la machine à l'origine de la requête est un de ses membres (client d'un FAI, machine au sein de l'entreprise...), soit qu'elle vient d'un autre serveur SMTP, et pas d'une machine « lambda ». Si Gmail effectuait ce test, la requête ci-dessus ne marcherait donc pas. Selon nos tests, un petit quart des serveurs de messagerie effectuent ces vérifications supplémentaires.

- Dans le cas où le serveur de messagerie n'a pas confiance en l'expéditeur, certains l'annoncent clairement par un message d'erreur, tandis que d'autres acceptent toutes les adresses comme si elles existaient, sans message d'erreur. C'est ce qu'on appelle des serveurs « *catch-all* », et c'est par exemple le cas des serveurs de Yahoo. Pour savoir si on est dans ce cas, il suffit en général de vérifier une ou plusieurs adresses aléatoires et très longues, avec le même nom de domaine : si elles sont toutes acceptées, c'est probablement que le serveur accepte tout. Il ne sera dès lors pas possible de vérifier des adresses.

Un serveur Catch-All est un serveur de messagerie qui répond positivement à toutes les requêtes, même si l'adresse n'existe pas.

- Les codes d'erreur, pourtant standard, ne sont pas utilisés de façon universelle. Par exemple, bien que le code « 550 » soit défini par les standards comme l'erreur d'une boîte inexistante, il est parfois également utilisé pour signifier que la requête est refusée pour des raisons évoquées plus haut ou que la boîte est pleine. Le message qui suit peut aider à savoir dans quelle situation l'on se trouve, mais il est dès lors difficile d'automatiser la chose avec un haut niveau de fiabilité pour de grandes listes d'adresses, en raison du caractère verbeux, hétérogène et multilingue du message.

Si l'on veut vérifier massivement des adresses, il faut être prudent. En effet, certains serveurs n'aiment pas ces vérifications et vont bloquer (ou *blacklister*) l'expéditeur, entre quelques minutes et quelques heures. Il s'agit en effet d'une technique de spammeur, pour trouver des adresses existantes.

Avec le recyclage des adresses, une adresse non validée depuis longtemps peut avoir changé de propriétaire.

Par ailleurs, certains fournisseurs d'adresses e-mail, comme Hotmail par exemple, « recyclent » les adresses inutilisées après un certain temps. Après neuf mois d'inactivité, les adresses sont désactivées, mais après un an, elles sont remises sur le marché, permettant à une nouvelle personne de demander l'adresse. Avec les techniques présentées ici, on pourra vérifier que l'adresse existe bel et bien, mais rien n'indique si elle appartient à la personne qui l'a fournie initialement.

2.5.3. Outils

Dans la pratique, ces manipulations ne sont pas nécessaires pour vérifier l'existence d'une adresse, car il existe des outils qui le font à votre place, avec plus ou moins de succès : http://verify-email.org, http://tools.email-checker.com, http://www.verifyemailaddress.org, http://www.ip-tracker.org/checker/email-lookup.php, http://bulkemailverifier.com...

Beaucoup d'outils ne tiennent pas compte des serveurs Catch-All, et valident donc des adresses inexistantes.

Cependant, pour une adresse erronée de chez Yahoo, les deux premiers indiquent qu'elle n'existe pas, les deux suivants qu'elle existe, et le dernier ne peut pas répondre ... Manifestement, la majorité de ces outils effectuent leurs tests à partir de machines qui ne sont pas des serveurs de messagerie, et donc auxquelles d'autres serveurs de messagerie un peu suspicieux ne font pas confiance. Ils se sont donc contentés des procédures basiques, sans y accorder la rigueur nécessaire à ces outils se voulant professionnels.

2.6. Contrôle de consultation

Contrôler qu'un e-mail a bien été consulté n'est pas une chose facile, et dans le meilleur des cas, ne sera possible qu'avec un degré de certitude très peu élevé. En général, dans les sociétés qui utilisent un gestionnaire d'envoi de campagnes de marketing (CRM, pour Customer Relationship Management) à la pointe de la technologie, utilisant les techniques de validation les plus avancées, ce ne sont en moyenne que 25 % des envois qui sont validés. Sur les 75 % restants, il y a bien évidemment une part d'adresses erronées qui génèrent un message d'erreur (ou « bounce »). En fonction du type de listing, ces erreurs tournent entre 10 et 15 %. Nous pourrions dans certains cas descendre sous les 5 % en appliquant toutes les techniques rassemblées dans ce document. Reste donc 60 % des adresses pour lesquelles on ne sait rien. Elles peuvent ne plus être consultées, mais il se peut aussi qu'elles le soient, mais que l'utilisateur ait désactivé tout ce qui permettait de le « tracer ».

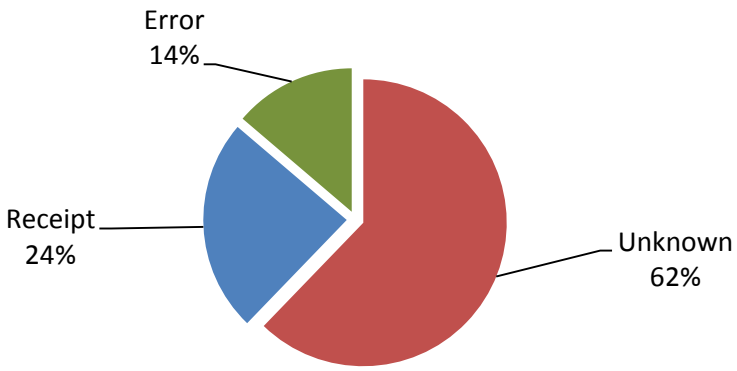


***Figure 1** : Exemple typique de situation suite à un envoi de campagne de communication : 24 % ont généré un accusé de réception, 14 % une erreur, et dans 62 % des cas, aucune information n'est disponible*

Mis à part les aspects techniques qui suivront, il sera également nécessaire de s'interroger sur les questions éthiques liées au suivi d'un e-mail. La frontière entre les techniques permettant d'améliorer la qualité d'une base de données et celles portant atteinte à la vie privée des gens est floue et sans doute facile à franchir. Nous ne présentons dans ce document que les considérations purement technologiques, et laissons au lecteur le soin d'évaluer si, dans son cas particulier, il y a lieu de s'assurer qu'un e-mail a bien été lu.

Il existe principalement trois façons de vérifier la consultation. La première consiste à utiliser, dans un logiciel comme Outlook, Eudora ou Thunderbird, l'option « Accusé de réception ». En fonction de la configuration du logiciel de lecture d'e-mails du destinataire, un e-mail de confirmation sera ou non renvoyé. L'inconvénient de cette solution est son manque de standardisation : une demande d'accusé de réception d'Outlook ne marchera vraisemblablement pas chez Eudora, et encore moins si le destinataire utilise un « webmail » tel que Gmail ou Hotmail. Cette méthode n'est en général pas utilisée par les solutions d'envoi automatique.

La seconde solution consiste à intégrer dans le texte de l'e-mail un lien à cliquer (ou URL), que ce soit pour accéder à la suite du message ou pour se connecter à un service. Ce lien, unique et spécifique au destinataire, ne conduira pas directement vers la page de destination, mais vers une page intermédiaire, qui pourra enregistrer le fait que ce lien a été cliqué, avant de rediriger automatiquement l'utilisateur vers la bonne page.

La troisième technique utilise le langage « HTML », principalement utilisé pour la mise en pages des sites Web, pour intégrer une image unique, spécifique à cet envoi, souvent invisible, mais dont la source se trouve sur un serveur dédié, qui peut enregistrer le fait que l'image a été téléchargée.

Nous allons maintenant présenter les deux dernières techniques, en mettant en avant leurs avantages, inconvénients, faiblesses et incertitudes. Nous présenterons ensuite deux pistes possibles pour des méthodes alternatives.

2.6.1. Redirection de liens

L'adresse d'une page web sur un site dynamique peut contenir des paramètres. Ils suivent en général un « ? » et sont une succession de couples « attribut=valeur » séparés par des « & ». Supposons que dans un e-mail, on place une icône ou texte avec un hyperlien vers l'adresse :

```
http://mysite.com/track?m=albert@gmail.com&dst=www.smals.be
```

Il sera alors très facile, sur le site mysite.com, d'enregistrer le fait que albert@gmail.com a cliqué sur le lien (on suppose qu'Albert est le seul à avoir reçu ce lien) et de le rediriger automatiquement vers www.smals.be, sans même qu'il s'aperçoive qu'il est passé par une page intermédiaire. L'inconvénient de ce procédé est d'une part qu'il devient très évident que l'on tente de tracer cette adresse et d'autre part qu'il est très facile, pour un utilisateur malintentionné, de faire valider n'importe quelle adresse. De plus, si cela convient pour rediriger vers une adresse aussi simple que www.smals.be, des problèmes se poseront pour rediriger vers des adresses plus complexes. En effet, si l'adresse vers laquelle on veut rediriger contient elle-même des paramètres, ce système ne permet pas de faire la différence entre les paramètres de l'URL de base et celle de l'URL de redirection.

On utilise en général un algorithme, nommé Base64, qui permet de convertir une chaîne d'octets en une chaîne de caractères, compatible avec une URL (l'adresse d'une page web). Cet algorithme traduirait par exemple

« albert@gmail.com;www.smals.be/a_page »

en

« YWxiZXJ0QGdtYWkuY29tO3d3dy5zbWFscy5iZS9hX3BhZ2U=»[19]

Ce qui pourrait nous donner comme adresse du lien :

[19] L'algorithme Base64 transforme chaque groupe de 3 caractères en un nouveau groupe de 4 caractères, parmi les suivants : A-Z, a-z, 0-9, +, /, =. La chaîne transformée est donc un tiers plus longue que la chaîne d'origine.

```
http://mysite.com/track?YWxi[…]tO3d3dy5zbWFscy5iZS9hX3BhZ2U=
```

On aurait donc typiquement, dans le mail, le code HTML suivant :

```
<a href='http://mysite.com/track?YWxi[…]tO3d3dy
5zbWFscy5iZS9hX3BhZ2U='>http://www.smals.be/a_page</a>
```

La page générée sur mysite.com pourrait être la suivante :

```
<html><head><meta http-equiv="refresh" content="0;
URL="http://www.smals.be/a_page"></head></html>
```

qui redirige automatiquement vers la page `www.smals.be/a_page`.

Cela ne suffit pas encore à empêcher un utilisateur malveillant de valider une mauvaise adresse, ou de se faire rediriger vers un autre page. On peut, avant d'utiliser l'algorithme Base64, chiffrer le texte à inclure avec une clé secrète.

Quelques remarques sur cette technique de redirection de lien :

- Dans l'exemple ci-dessus, le lien contient directement deux informations : une adresse e-mail et une URL de redirection. On aurait également pu placer ces deux informations dans une base de données, et ne reprendre que l'identifiant dans le lien, ce qui aurait eu l'avantage de réduire la taille du lien, mais l'inconvénient d'exiger de stocker plus d'information.
- Le lien permet de savoir qu'albert@gmail.com a cliqué sur un lien vers le site web de Smals, mais ne permet pas de savoir dans quel e-mail. En effet, si ce lien apparaît dans cinq e-mails différents qu'il a reçus, on ne saura pas lequel a été ouvert. Cependant, ce qui nous intéresse dans notre cas, c'est de savoir que l'adresse e-mail albert@gmail.com est toujours active, et pas spécifiquement de savoir quels sont les e-mails qui ont été ouverts. Si le but est différent, il faudra ajouter un identifiant supplémentaire dans le lien.
- Si Albert fait suivre (ou « forwarde ») l'e-mail qu'il a reçu à sa sœur Marie-Célestine et qu'elle clique sur le lien, ça sera toujours l'adresse d'Albert qui sera validée, même si elle le fait six mois après avoir reçu l'e-mail et qu'entretemps, Albert est passé chez un autre fournisseur. Bien que peu probable comme scénario, il montre que le mécanisme n'est pas infaillible.

Un suivi par redirection de lien ne marche que si l'on parvient à convaincre la majorité des utilisateurs de cliquer sur les liens.

- Cette technique ne fonctionnera que si on rend presque indispensable de cliquer sur un lien. Il faut donc mettre en place des stratégies de marketing et de communication pour inciter les destinataires à ces actions. Ce peut être par exemple en ne plaçant dans l'e-mail qu'une accroche suivie d'un lien « lire la suite » ou en y incluant des liens vers des documents importants, voire obligatoires.
- Si l'URL de redirection (http://www.smals.be/a_page dans notre exemple) est relativement simple et apparaît clairement dans le texte, un utilisateur ne désirant pas se faire « traquer » pourra

directement taper (ou copier-coller) l'adresse dans son navigateur, sans cliquer sur le lien.

2.6.2. Image avec identifiant unique

Il y a deux façons d'intégrer une image dans un e-mail. La première consiste à l'envoyer en pièce jointe, la seconde consiste à la laisser sur un serveur web et à indiquer dans le code de l'e-mail son adresse. La première solution a l'inconvénient d'alourdir la taille des e-mails. La seconde a pour désavantage que l'affichage des images est souvent désactivé par défaut par le client de messagerie, pour éviter justement les techniques que nous décrivons ici. Les e-mails apparaissent alors dans une version purement textuelle, avec un message du type « Pour préserver votre confidentialité, les images distantes ne sont pas chargées. Cliquez ici pour afficher les images ».

Le principe consiste donc à inclure dans le code HTML de l'e-mail une image distante, qui sera différente pour chaque courrier envoyé (si l'on veut savoir que ce courrier précis a été consulté), ou à tout le moins, pour chaque destinataire (si l'on veut juste s'assurer qu'une adresse est toujours active). Le serveur web sur lequel se trouvera l'image pourra donc identifier le courrier à l'origine du chargement, ce qui permettra de s'assurer que le message est bien ouvert, et donc que l'adresse est toujours active. Pour identifier l'image, ou pourra utiliser les mêmes techniques que ci-dessus, le nom de l'image contenant donc soit une version chiffrée de l'adresse e-mail, soit un identifiant dans une base de données.

Les outils de « tracking » du marché (voir Section 4.2.1 pour plus de détails) incluent souvent une image qui n'affecte pas la mise en pages, en général une image d'un pixel blanc. D'autres images peuvent aussi être incluses, mais n'ont pas besoin d'être identifiables.

À nouveau, il faudra mettre en place des stratégies de marketing et de communication pour encourager les destinataires à accepter l'affichage des images. Rendre les e-mails quasiment illisibles sans image pourrait avoir l'effet inverse et inciter le lecteur à considérer le message comme une publicité inutile et non sollicité, et à l'envoyer directement dans sa corbeille. L'idéal est d'envoyer tous les messages d'une plateforme avec le même expéditeur, ce qui permettra au destinataire d'autoriser l'affichage des images pour tous les messages provenant de cet expéditeur. Par ailleurs, on peut indiquer à l'utilisateur que s'il accepte d'afficher les images automatiquement, il ne sera plus nécessaire de lui demander régulièrement de confirmer explicitement que son adresse est toujours valide.

Comme dans la section précédente, si un utilisateur fait suivre le courrier, on ne pourra pas différencier l'ouverture du courrier original de celle du courrier transféré.

Remarquons que, à notre connaissance, l'affichage d'une image distante ne compromet en rien la sécurité. Mises à part les pièces jointes infectées,

le principal risque de contamination en ouvrant un e-mail est la présence de code JavaScript dans l'e-mail, qui est bloqué par la plupart des clients mail, tant webmail qu'applicatifs. On ne peut donc pas utiliser le JavaScript pour valider une adresse e-mail.

Par ailleurs, certains outils de « tracking » se servent de l'image incluse pour détecter où a été ouvert l'e-mail. En effet, lorsque l'image est téléchargée sur le serveur, celui-ci peut obtenir l'adresse IP de la machine effectuant la requête, et, grâce à cette adresse, trouver l'origine géographique de l'ouverture. Si cette technique peut fonctionner avec de client mail « à la » Outlook, le résultat est plus aléatoire avec les webmails (Gmail, Hotmail, …). Avec Gmail[20] en effet, c'est le navigateur qui télécharge lui-même l'image, et on peut donc le localiser. Avec Hotmail, par contre, ce sont les serveurs de Hotmail qui téléchargent d'abord l'image, avant d'en envoyer une copie au navigateur. De ce fait, le serveur où se trouve l'image ne peut que localiser les serveurs de Hotmail.

2.6.3. Contrôle lors d'autres contacts

Mis à part les contrôles présentés ci-dessus, qui peuvent se réaliser automatiquement lors de l'envoi d'un e-mail, on peut également mettre en œuvre des mécanismes plus « manuels ». Par exemple, lorsqu'un citoyen ou un client contacte une administration ou un service, que ce soit par téléphone, physiquement, ou par e-mail, on peut en profiter pour lui demander si l'adresse renseignée est toujours valide[21]. Dans le cas positif, cette information peut être renseignée dans la base de données, avec la date correspondante. Il faut cependant noter que cette validation n'est pas aussi formelle que celles présentées ci-dessus, puisqu'une personne peut très bien affirmer qu'une adresse est valide alors qu'elle n'existe pas. On peut donc différencier ce type de validation informelle d'une validation stricte.

2.6.4. Réponse aux e-mails

Si un e-mail évoqué dans les sections précédentes nécessite une réponse et que celle-ci peut se faire en répondant directement à l'e-mail en question, il est possible de mettre en œuvre des mécanismes permettant, à la réception de l'e-mail, d'enregistrer l'adresse et de la valider dans la base de données. Cela peut se faire en précisant une adresse de retour unique, qui, après avoir noté la confirmation de l'adresse, transfère l'e-mail au bon destinataire, ou en incluant un mécanisme d'« ID » dans les métadonnées de l'e-mail. Nous ne détaillerons pas cette méthode dans ce document, et la laissons pour de possibles travaux futurs.

[20] Mise à jour : en décembre 2013, Gmail a changé sa politique d'affichage des images : elles sont désormais téléchargées sur ses serveurs dans un premier temps, avant d'être envoyées au client. La géolocalisation est donc également impossible.

[21] Conceptnota « filter voor de eBox », Kristof Verslype, Smals Research.

2.7. Indicateurs de qualité et statistiques

Au cours de notre étude, Nous avons pu travailler sur des échantillons d'une série de bases de données, certaines contenant les coordonnées de « citoyens lambda » ayant contacté une entreprise pour diverses raisons, d'autres contenant des personnes de contact dans des entreprises. De cette étude, nous avons pu dégager plusieurs observations, les unes à portée opérationnelle (Sections 2.7.1 et 2.7.2), les autres à portée descriptive permettant de caractériser une base de données (Sections 2.7.3 et 2.7.4). Nous en détaillerons quelques-unes dans les sections suivantes, sans être exhaustif sur ce que nous avons pu découvrir.

2.7.1. Erreurs syntaxiques

Toutes les bases de données que nous avons pu étudier contenaient des erreurs de syntaxe. Certaines d'entre elles simplement parce qu'aucun contrôle, même élémentaire, n'était exécutées à la source. On y trouve par exemple des numéros de téléphone ou des adresses postales, dans les cas où l'utilisateur a confondu deux champs. On trouve également beaucoup de cas où l'arobase (@) a été remplacé par une des touches voisines du clavier (|, #, é, 2, &… sur un AZERTY). Il est également fréquent que les utilisateurs aient voulu introduire deux adresses différentes, séparées par une espace[22], une virgule ou un point-virgule, compliquant tout traitement automatique.

Cependant, selon notre expérience, le nombre d'erreurs syntaxiques dépasse rarement le demi-pourcent, ce qui laisse à penser que les gens sont attentifs ou que l'aide à la saisie automatique permet d'éviter bien des erreurs.

Étonnamment, même dans les systèmes avec des contrôles à l'entrée, nous avons trouvé des erreurs syntaxiques. Selon nous, il y a principalement deux raisons :

- La première est que nos tests étaient plus poussés que ce que font la plupart des systèmes. Nous tenons compte par exemple du nom de domaine pour affiner nos tests, ce qui à notre connaissance n'est presque jamais fait.
- Il peut arriver qu'une base de données ait plusieurs points d'entrée non maîtrisés, contrairement aux bonnes pratiques. Dans un des cas que nous avons étudiés, une personne, via un portail avec contrôle syntaxique, peut s'enregistrer elle-même, mais il existe une autre filière, ou du personnel administratif peut introduire un dossier, mais cette fois sans contrôle syntaxique. Ceci montre l'importance d'une bonne organisation (voir Section 3.10).

[22] En français, le caractère typographique séparant deux mots est féminin (une espace). L'espace au masculin désigne l'étendue entre deux choses, par exemple deux mots. On parle donc d'un espace entre deux mots si l'on parle de l'étendue vide qui les sépare et d'une espace entre deux mots si l'on parle du caractère typographique.

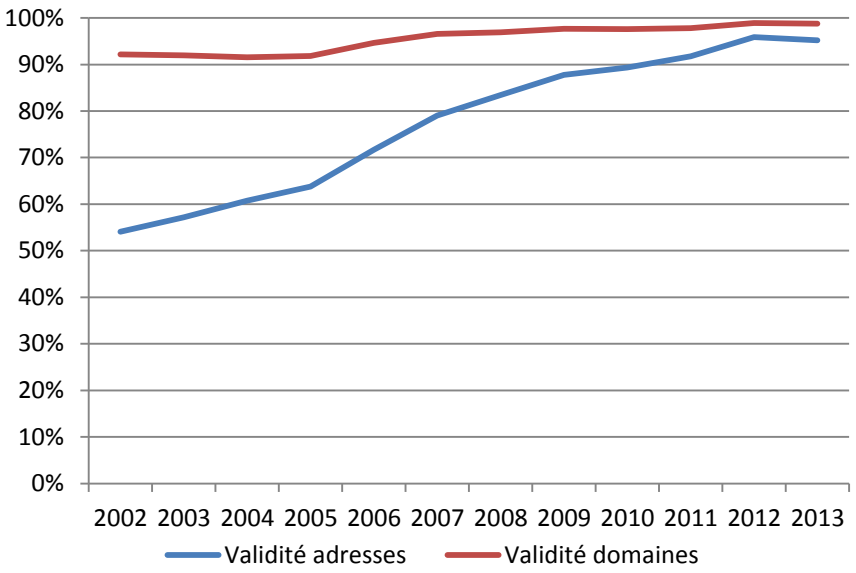


Figure 2 : *Validité des adresses e-mail et de leur nom de domaine associé d'une base de données de citoyens en fonction de l'année d'introduction*

2.7.2. Dégressivité de la validité dans le temps

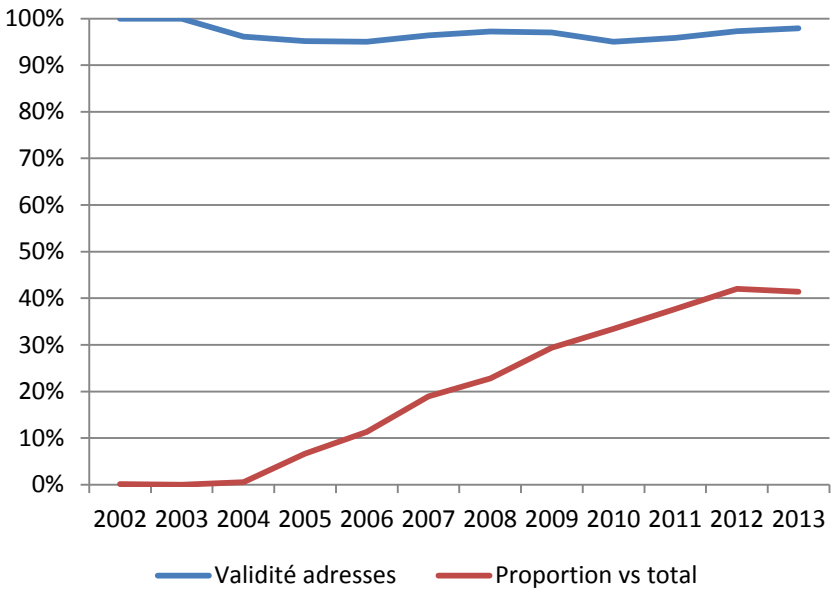
Pour deux des bases de données que nous avons étudiées, l'une contenant des adresses e-mail de particuliers, l'autre d'entreprises, nous avions une date soit d'introduction, soit de mise à jour. Nous nous doutions que des vieilles adresses avaient moins de chance d'être valides que des nouvelles, mais nous avons pu mesurer à quel point. En utilisant les techniques détaillées en Section 2.5 sur les 50.000 adresses que contenait chacune des tables, nous avons pu établir le graphique donné en Figure 2. Le constat est donc net : si les e-mails récents avaient un taux de validité avoisinant les 95 %, ce même taux chute à près de 50 % pour les adresses ayant onze ans d'âge. Notez que l'on parle bien de l'existence des adresses et non du fait qu'elles soient consultées ou non. Sans envoyer d'e-mail à toutes ces adresses, ce que nous n'avons bien entendu pas fait, il est impossible de savoir si elles sont toujours actives.

Ce constat n'est pas une surprise en soi, mais nous n'avions pas trouvé jusqu'ici de mesure rigoureuse de ce phénomène.

Nous avons également repris sur le même graphique la validité des noms de domaine. Nous observons une progression temporelle, mais nettement plus faible que pour les adresses (Figure 2) : 92 % des adresses de plus de onze ans sont chez un fournisseur toujours actif, contre 98 % des adresses les plus récentes (2012 et 2013). Nous pouvons tirer deux conclusions de ces observations :

1. Ce qui rend obsolète une adresse, ce n'est que peu la versatilité des noms de domaine, mais, selon nous, le manque de fidélité des utilisateurs qui changent de fournisseurs.
2. Beaucoup de gestionnaires pensent que s'ils ont un bon système de validation en entrée (avec e-mail de confirmation), ils ont une grande garantie d'avoir une base de données de bonne qualité. Ce

Figure 3 : Validité des adresses Gmail et proportion dans la base de données



n'est manifestement pas vrai : l'essentiel de la difficulté réside dans le suivi dans le temps. Il faut parvenir à faire en sorte que l'utilisateur mette à jour ses informations.

2.7.3. Dépendance au nom de domaine

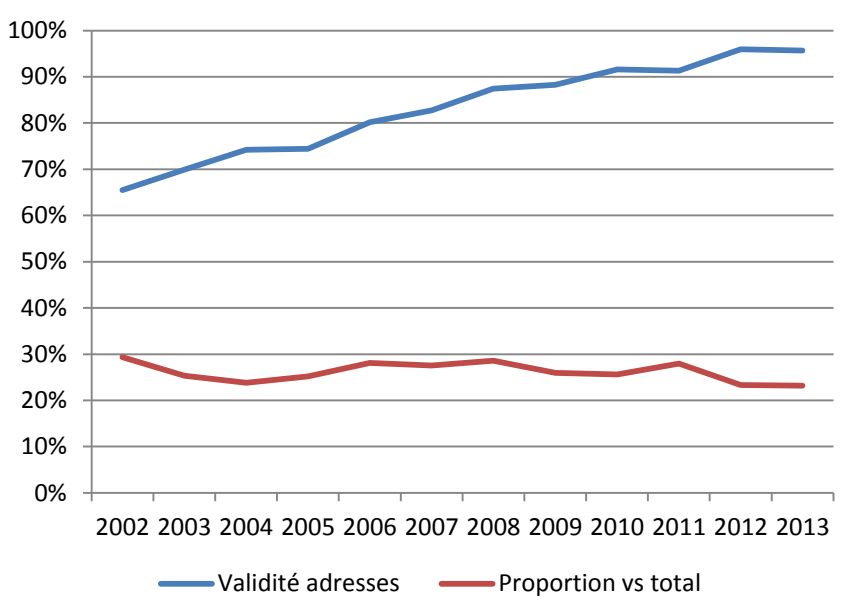
Si l'on s'intéresse à cette dégressivité pour des noms de domaine particuliers, d'autres constats intéressants sont à faire. Les Figure 3, Figure 4 et Figure 5 reprennent, en bleu, la même information, en ne regardant que les adresses, respectivement, de Gmail, Hotmail et Skynet. En rouge, on indique la proportion d'adresses de Gmail, Hotmail ou Skynet dans la base de données.

La Figure 3 nous montre, d'une part, qu'en 2013, plus de 40 % des personnes se servaient d'un adresse Gmail (il s'agit d'un public essentiellement « IT »), et, d'autre part, que la validité de ces adresses est restée stable ces onze dernières années (sachant que les deux premières années ne reprennent pas assez d'adresses pour être significatives). Si l'on regarde par contre le même graphique pour Hotmail (Figure 4), on retrouve le même genre de dégressivité que dans le graphique général (Figure 2), avec 35 % d'adresses âgées de onze ans invalides. Au premier abord, on pourrait se dire que c'est une conséquence d'une politique de désactivation plus rapide chez l'un que chez l'autre. Or, Gmail comme Hotmail désactivent les adresses après neuf mois sans connexion (même si des messages continuent à arriver).

Notre hypothèse empirique, qui demanderait certainement des études plus approfondies pour être formellement validée, est que les utilisateurs de Gmail sont plus « fidèles » que ceux de Hotmail.

En Figure 5, avec les données pour Skynet, la dégressivité est encore plus marquée : près de 65 % des adresses Skynet de plus de onze ans ne sont plus valides. Skynet étant un fournisseur d'accès à Internet (appartenant à

Figure 4 : Validité des adresses Hotmail et proportion dans la base de données



Belgacom), toute personne changeant de fournisseur perd son adresse. Et vu le grand nombre de nouveaux acteurs apparus ces dernières années sur le marché des connexions au réseau, il n'est pas étonnant que beaucoup de clients aient changé d'opérateur.

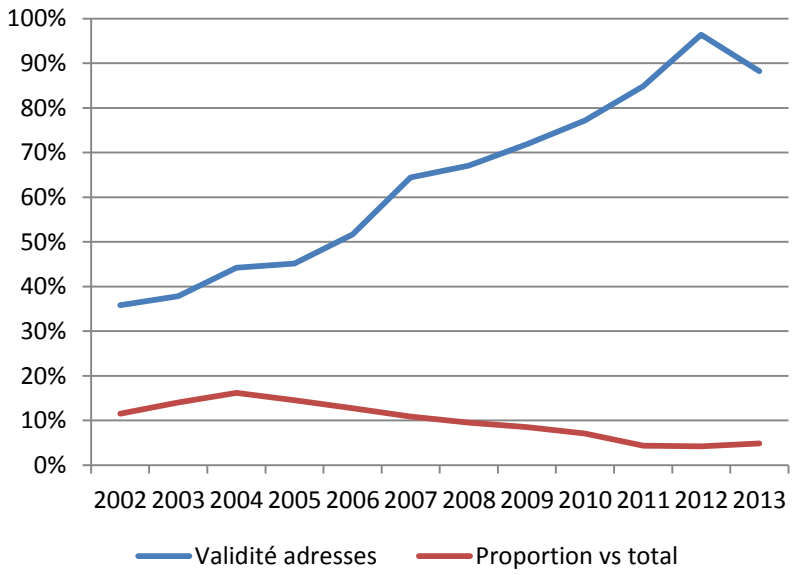
2.7.4. Proportion professionnels-particuliers

Durant notre étude, nous avons été amenés à analyser diverses bases de données à partir d'échantillons représentatifs, dont trois avec un profil très différent :

- Une première base de données contenant des particuliers fournissant des adresses personnelles, que nous nommerons ci-après « Particuliers » ;
- Une seconde avec des personnes de contact au sein d'entreprises de tailles moyennes et relativement homogènes, que nous nommerons « Entreprises »
- Une troisième contenant également des entreprises, mais nettement plus hétérogènes en terme de taille, depuis le petit indépendant travaillant seul jusqu'aux grosses entreprises ou administrations, que nous nommerons « Mixte ».

Nous nous sommes demandé si ces différentes bases de données avaient des profils différents et identifiables. Entre d'autres termes, nous avons voulu savoir s'il était possible, à partir de diverses métriques, si une base de données contient essentiellement des particuliers fournissant une adresse personnelle, ou des professionnels représentant une entreprise. Ce que nous avons principalement observé, c'est une disparité des noms de domaine plus importante pour les listings plus « professionnels ». En effet, chez les particuliers, la plupart des adresses sont issues des grands

Figure 5 : Validité des adresses Skynet et proportion dans la base de données



acteurs du marché, tels que Gmail, Hotmail ou Telenet, alors que les professionnels sont nombreux à fournir une adresse avec le nom de domaine de leur entreprise. Nous proposons quatre métriques pour mesurer cette tendance.

La première (Ratio e-mail/DN) est le ratio entre le nombre d'e-mails présents dans la base de données, et le nombre de noms de domaine. Un ratio de 10 indiquerait qu'en moyenne, chaque nom de domaine présent dans la base de données regroupe dix adresses. Un ratio plus élevé indiquerait donc une plus grande concentration sur moins de noms de domaine, et serait donc vraisemblablement le signe d'adresses plus personnelles.

La seconde métrique (Top 10) consiste en la proportion d'adresses appartenant aux dix plus gros noms de domaine de la base de données. À nouveau, plus ce nombre est élevé, moins la disparité est importante, ce qui pourrait être le signe d'adresses personnelles.

La troisième métrique (Domaine unique) mesure la proportion d'adresses étant seules avec ce nom de domaine. Majoritairement, il s'agit soit de la personne de contact d'une entreprise possédant un nom de domaine, soit d'une personne ayant acheté son nom de domaine propre (typiquement avec son prénom et/ou son nom), ce qui est plutôt rare (voir Section 3.4.1). On s'attend à une faible valeur pour des particuliers.

Le tableau ci-dessous reprend les valeurs de ces trois métriques, pour les trois types de bases de données étudiées.

	Entreprises	Mixte	Particuliers
Ratio e-mail/DN	2.23	3.74	11.35
Top 10	37.9%	62.0%	80%
Domaine unique	37.5%	24.8%	7.8%

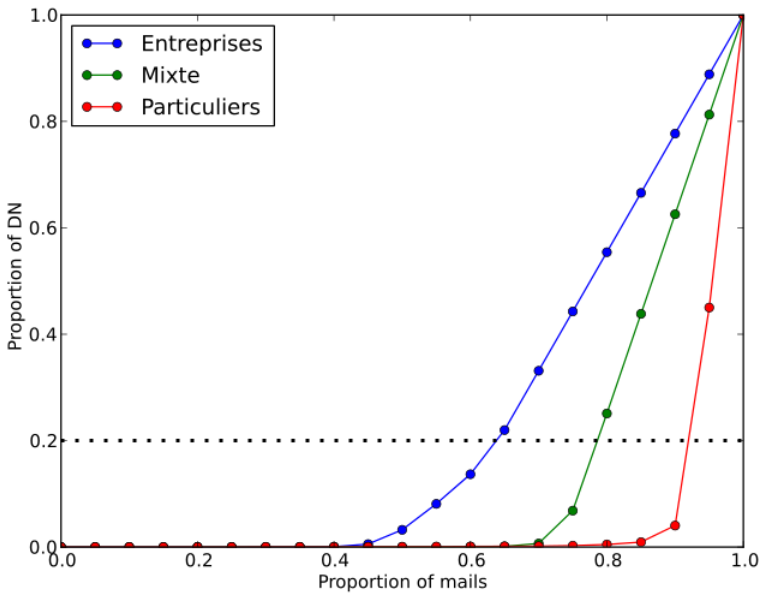


Figure 6 : *Distribution des adresses par rapport aux noms de domaine*

La dernière métrique est reprise en Figure 6. Elle représente la distribution des e-mails par rapport aux noms de domaine. En quelque sorte, c'est une généralisation de la métrique « Top 10 ». On peut par exemple voir que 20 % des noms de domaine les plus importants (ligne en pointillés) regroupent à peu près 65 % des adresses d'entreprise, 78 % des adresses mixtes et 90 % des adresses de particuliers

.

3. Stratégie de bonne gestion des adresses e-mail

Nous allons maintenant montrer comment les éléments rassemblés dans le chapitre précédent peuvent améliorer la qualité d'une base de données d'e-mail, et les synthétiser. Dans un premier temps, nous allons parcourir les différents aspects (syntaxique, existence de nom de domaine et d'adresse, et « matching »), en allant au maximum de ce qu'il est possible de faire. Nous étudierons ensuite quelques scénarios, dans le but de voir jusqu'où il est nécessaire de tester les différents aspects, en fonction des circonstances (par exemple, on-line ou batch).

Notons tout d'abord que seule une partie de ce que nous présentons ici est originale. Un partie non négligeable de cette stratégie est déjà d'application ou en voie de l'être dans certaines entreprises[23].

À la fois pour le traitement en batch où nous avons développé et configuré plusieurs outils, dont les Data Quality Tools de Smals, ou pour une version on-line pour laquelle nous avons développé un « Proof of Concept » (PoC) reprenant la stratégie proposée, nous pouvons mettre nos connaissances et nos outils à la disposition des services intéressés.

3.1. Arbitrage stratégique

Comme présenté plus haut, le contrôle formel d'une adresse électronique peut s'avérer compliqué. La stratégie à adopter dépendra de l'objectif. Il faudra se poser l'une ou l'autre question. Il s'agira d'un type d'arbitrage bien connu des gestionnaires de bases de données, qui doivent faire des choix entre rapidité et validité.

3.1.1. Rejeter les mauvais ou accepter les bons ?

Cherche-t-on à être sûr de ne pas accepter une adresse syntaxiquement incorrecte ou veut-on plutôt s'assurer de ne jamais refuser une adresse correcte ? Dans le premier cas, il faudra accepter de refuser de temps à

[23] Voir par exemple le rapport d'étude Smals « Service de validation d'adresses e-mail », J. Defèche, A. Clerbaut, S. Flamme, 7 janvier 2013 (Proposition)

autre une adresse correcte. Dans le second, on laissera de temps en temps passer une adresse mal formée.

3.1.2. Accepter l'exotisme ?

Selon les standards, il faudrait considérer comme équivalentes les différentes adresses :
- albert.leroy@smals.be
- " albert.leroy "@smals.be
- albert.leroy@[85.91.15.44]
- « Albert Leroy <albert.leroy@smals.be>»

Si l'on respecte les standards internationaux (RFC 5321 et RFC 5322[24]), on devrait pouvoir accepter comme adresses e-mail équivalentes à albert.leroy@smals.be toutes les adresses suivantes :

- « Albert LEROY <albert.leroy@smals.be> »
- " albert.leroy "@smals.be
- albert.leroy@[85.91.175.44]
- albert.leroy(un commentaire à propos du username)@smals.be(un commentaire à propos du nom de domaine).

Il semble cependant raisonnable de n'accepter que la syntaxe équivalente « albert.leroy@smals.be », ce qui simplifie notablement les tests à effectuer et ne réduit en rien les possibilités de s'inscrire. Similairement, on peut imposer qu'un numéro de téléphone soit uniquement composé de chiffres et de « + », et obliger un utilisateur à encoder 027871234 au lieu de 02/787.12.34, sans risque d'empêcher quelqu'un de communiquer son numéro.

De même, un citoyen chinois, arabe ou indien désirant s'inscrire dans une quelconque administration belge ne peut pas le faire avec son nom dans sa graphie originale, mais doit utiliser une translitération. Il pourrait paraitre raisonnable qu'il en aille de même avec les adresses e-mail. On pourrait dès lors limiter la syntaxe aux seuls caractères latins (et leurs versions accentuées), tant pour le nom de domaine que pour le nom d'utilisateur. En effet, si un employé d'une administration belge devait encoder ou corriger lui-même une telle adresse, il aurait sans doute beaucoup de mal à le faire.

3.1.3. Quel contrôle sur les serveurs d'envoi d'e-mail ?

L'envoi d'un e-mail à une adresse erronée génère en général un message d'erreur, appelé « bounce », que l'on reçoit également comme un e-mail. Ce message contient un code d'erreur, ainsi qu'une description. La difficulté est qu'il est très complexe, d'une part, de se rendre compte qu'il s'agit d'un « bounce mail », comme déjà décrit en Section 2.5.2, et d'autre part, d'identifier précisément le problème. Un mémoire entier, présenté dans l'introduction, a été écrit sur le sujet, et beaucoup d'incertitude reste présente.

Par contre, le serveur de messagerie au travers duquel le message a été envoyé (le serveur SMTP) peut obtenir beaucoup plus d'information. Si la plateforme que l'on développe peut elle-même soit jouer le rôle du serveur SMTP, soit interagir de façon rapprochée avec ce service, on obtiendra une information de bien meilleure qualité. Cela nécessite

[24] http://tools.ietf.org/html/rfc5321 et http://tools.ietf.org/html/rfc5322

cependant un plus grand investissement en temps et en ressources, car il faudra, entre autres que le serveur en question soit légitimé pour un nom de domaine reconnu par les serveurs DNS, dans son champ MX.

3.2. Aspects syntaxiques

3.2.1. Catégorisation

Étant donnée la grande disparité des syntaxes et la difficulté à la vérifier formellement, il est illusoire de pouvoir classer les adresses en deux catégories (correctes et incorrectes) avec une certitude absolue. Dans la pratique, les algorithmes de test qui veulent être sûrs d'accepter toutes les adresses correctes sont trop laxistes, alors que ceux qui veulent des contrôles plus contraignants rejettent des adresses qui ne devraient pas l'être. Nous suggérons fortement, comme nous le ferons également plus loin pour la validation des adresses, de catégoriser les adresses en trois groupes (voir par exemple Annexe 7.2) :

- Celles dont la syntaxe est certainement fausse : « @ » manquant, présence d'espace ou de virgule, de points successifs… ;
- Les adresses certainement correctes au niveau syntaxique ;
- Les adresses suspicieuses, contenant, par exemple, des caractères officiellement acceptés, mais peu courants.

Adresses certainement fausses

Pour la première catégorie, il faudra principalement procéder par élimination, et lister les caractères refusés plutôt que les caractères acceptés. Par exemple, toutes les adresses n'étant pas validées par l'expression régulière[25] suivante rentreront dans cette catégorie :

```
^[^@,;:\s]+@[^@,;:\s]+$
```

Dans cette expression, décrivant le format accepté, le premier symbole (« ^ ») et le dernier (« $ ») indiquent respectivement le début et la fin de la chaîne de caractères à analyser. Le symbole « @ », au milieu, représente ce même symbole au sein de l'adresse. Les deux séquences « [^@,;:\s]+ » représentent la partie « username » (avant l'arobase) et le nom de domaine après l'arobase). Dans ce cas-ci, on accepte « n'importe quelle séquence d'au moins un caractère, ne pouvant pas être l'arobase, une virgule, un point-virgule, deux points et l'espace » (notez que l'on pourrait lister d'autres caractères interdits tels que les parenthèses ou les crochets). La possible présence des nouvelles générations de TLD et des noms de domaine permet difficilement d'être plus contraignant. Comme exposé plus haut, il n'est formellement même pas nécessaire d'avoir un point dans le nom de domaine. Il serait possible d'être plus contraignant,

[25] Sorte de mini langages de programmation permettant de valider qu'une chaîne de caractères respecte certaines contraintes. Plus de détails sur ces expressions régulières et leur syntaxe sur http://www.regular-expressions.info/

puisque l'on sait qu'on ne peut pas avoir de points ou de tiret successifs dans le nom de domaine :

```
^[^@,;:\s]+@([^@ ,;:\s+.-]+([-.][^@,;:\s+.-]+)*)$
```

Notons que l'on pourrait de la même façon interdire les points successifs dans le nom d'utilisateur. Cependant, bien que la syntaxe des noms de domaine soit règlementée à un haut niveau et fortement standardisée, les gestionnaires d'adresses e-mail ne sont pas fondamentalement contraints de suivre les standards. Pour cette catégorie, il est donc probablement prudent de ne pas trop contraindre le nom d'utilisateur.

Dans le cas où l'expression régulière est testée dans un langage supportant bien l'Unicode, il existe une expression « lettre Unicode », représentée par « \p{L} », regroupant tous les caractères étant des lettres, accentuées ou non, quel que soit l'alphabet. Ceci élimine donc tous les caractères de ponctuation ou les signes mathématique. Or on sait qu'un nom de domaine, s'il peut contenir des caractères non latins, ne pourra jamais contenir que des lettres et des chiffres (mis à part le point et le tiret). Par ailleurs, le TLD (qui soit classique ou générique) ne peut contenir que des lettres, et donc ni tiret, ni chiffre. Ceci veut dire en particulier que si un nom de domaine ne contient pas de point, cela signifie qu'il n'est composé que d'un TLD et ne peut donc contenir que des lettres. On peut dès lors adapter l'expression :

```
[^@,;:\s]+@([\p{L}0-9]+([-][\p{L}0-9]+)*[.])*[\p{L}]{2,}$
```

Adresses certainement correctes

Le critère de sélection de la seconde catégorie dépendra de ce qu'on a décidé d'accepter : s'il est très peu probable de voir des caractères non latins, voire même accentués, on peut ne garder dans cette catégorie que les adresses respectant la syntaxe suivante :

```
^[a-z0-9'.+_-]+@([a-z0-9]+([-.][a-z0-9]+)*)\.[a-z]{2,4}$
```

On n'accepte donc dans la partie « utilisateur » que les lettres (non accentuées), les chiffres, l'apostrophe, le point, le +, le tiret bas (*undescore)* et le tiret. Dans la partie nom de domaine, on accepte uniquement les lettres, les chiffres, on interdit d'avoir deux points ou tirets successifs, et on impose la présence d'un TLD de maximum 4 lettres. En effet, en attendant l'arrivée des gTLD (voir Section 2.3), seuls deux TLD font plus de 4 lettres : museum et travel. On peut raisonnablement estimer qu'un TLD de plus de 4 lettres a beaucoup de chances d'être la conséquence d'une faute de frappe. On peut facilement adapter cette expression si l'on veut y accepter les accents, que ça soit pour le nom de domaine ou le nom d'utilisateur.

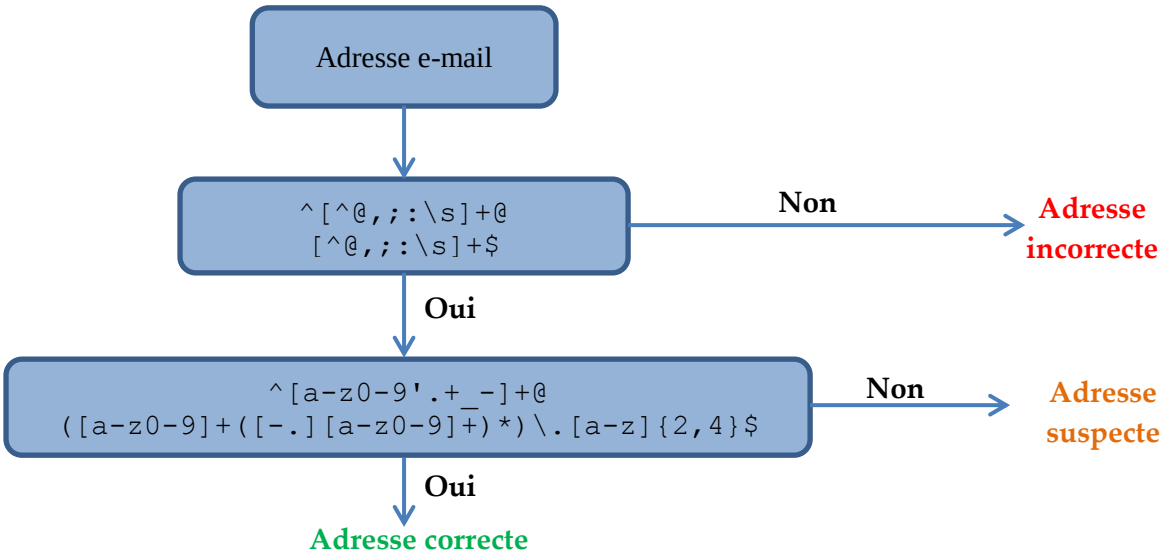
On peut également interdire la présence de points multiples dans le nom d'utilisateur, ainsi que les noms en début ou en fin de nom d'utilisateur :

```
^[a-z0-9'+_-]+(\.[a-z0-9'+_-]+)*@
([a-z0-9]+([-.][a-z0-9]+)*)\.[a-z]{2,4}$
```

Adresses suspicieuses

Toutes les adresses n'entrant pas dans ces deux catégories iront dans la troisième et peuvent faire l'objet d'un traitement manuel selon leur caractère critique (qui sera très différent suivant que l'on est dans un contexte batch ou on-line ; on y reviendra ultérieurement). Plus le critère de sélection des deux autres catégories est performant, moins on devrait trouver d'adresses dans la dernière.

On peut donc considérer le schéma d'acceptation suivant (pour lequel les expressions peuvent bien évidemment être adaptées, nous n'avons gardé que les plus simples pour faciliter la lecture) :



3.2.2. Syntaxe spécifique

Comme détaillé plus haut, bien que les standards acceptent des syntaxes très variées, les plus grands fournisseurs ont adopté des règles beaucoup plus contraignantes, nettement plus faciles à vérifier. Nous avons pu établir cette syntaxe pour de nombreux fournisseurs, tels que Gmail, Yahoo, Hotmail, Skynet, Belgacom et Telenet. Dans un listing de clients (particuliers) d'une grande entreprise belge que nous avons pu étudier, près de 80 % des adresses appartiennent à un nom de domaine dont nous connaissions la syntaxe, ce qui nous a permis d'invalider des adresses qui ne l'auraient pas été sans ces tests spécifiques. À titre d'exemple, nous avons rassemblé en Annexe 7.1 la syntaxe de nom d'utilisateur de quelques fournisseurs.

3.2.3. Suggestions de correction

Si la syntaxe d'une adresse s'avère suspicieuse ou erronée, nous avons mis en place dans notre PoC une série de mesures permettant de corriger les adresses, en vue de fournir une suggestion de correction (qui devra ensuite dans tous les cas être validée soit par l'utilisateur même, soit par

un administrateur). Nos algorithmes, simples, auraient permis de corriger une large partie des erreurs syntaxiques que nous avons trouvées. Voici quelques pistes :

- Supprimer les caractères « | », « & », « # », « é » ou « 2 » (voisins de « @ » sur le clavier) collés à un « @ » ;
- En cas d'absence de « @ » :
    - Si « | », « & », « # », « é » ou « 2 » est présent, en remplacer un par « @ » ;
    - Si on trouve à l'extrême droite un domaine connu (par exemple, albert.leroygmail.com), le faire précéder par d'un « @ » ;
- Remplacer les caractères accentués par leur version non accentuée (é, è, ê → e ; à → a ; …) ;
- Remplacer les « .. » par « . », supprimer les points en début ou en fin de nom d'utilisateur ;
- Supprimer tous les caractères interdits ou suspects (autres que a-z, 0-9, « . », « _ », « ' », « + », « - », « @ ») ;
- Pour les adresses Gmail, remplacer « _ » et « - » par « . », pour Yahoo, remplacer « - » par « _ ».

3.3. Validation d'adresse

La validation d'une adresse peut se faire en deux parties : premièrement, il s'agit de vérifier l'existence du nom de domaine, et deuxièmement de l'adresse entière.

La première validation est relativement facile à réaliser, et ne nécessite pas énormément de ressources. Avec une connexion tout à fait standard et une simple machine de bureau, nous avons pu tester l'existence d'une dizaine de domaines par seconde. Il y aura à nouveau lieu d'établir trois catégories : les noms de domaine qui existent, ceux qui n'existent certainement pas et ceux (rares, selon notre expérience) pour lesquels la requête échoue. En général, pour cette dernière catégorie, il s'agit de problèmes temporaires, et une requête ultérieure a plus de succès (ce qu'on ne peut bien sûr pas envisager dans un scénario on-line).

La validation de l'existence d'une adresse est plus complexe. D'une part, elle est nettement plus longue. Il faut compter en moyenne de l'ordre d'une seconde pour avoir une réponse (on peut cependant améliorer les choses si on a plusieurs adresses à tester chez le même fournisseur), mais il n'est pas rare d'avoir une requête nécessitant plusieurs dizaines de secondes, ce qui n'est pas non plus envisageable dans un scénario on-line.

Par ailleurs, la réponse est d'une fiabilité toute relative. D'une part à cause de la difficulté à comprendre le message de retour, dont le format et les codes varient d'un fournisseur à l'autre. D'autre part parce qu'une réponse positive peut être due à un serveur « catch-all » qui répond

positivement à chaque requête. Pour être tout à fait rigoureux, on ne pourrait donner l'état d'une adresse que pour les domaines (ou au moins les serveurs MX) pour lesquels on a pratiqué suffisamment de tests. Cependant, il n'est pas toujours nécessaire d'avoir un très grand niveau de fiabilité. On peut alors répartir les adresses en trois catégories :

1. Les adresses valides : message clair et compréhensible annonçant une validité, sur un serveur qui n'est pas « catch-all »
2. Les adresses invalides
3. Les adresses incertaines :
    a. Message incompréhensible
    b. Indique clairement qu'il ne veut pas répondre
    c. Erreur temporaire
    d. Les adresses sur des domaines « catch-all »

Remarquons que ce que l'on considère comme « catch-all » ou « non catch-all » peut avoir deux significations en fonction de la fiabilité désirée : soit une approche pessimiste où tout domaine qui n'est pas connu comme « non catch-all » est considéré comme « catch-all », soit une approche optimiste, où tout serveur qui n'est pas connu comme « catch-all » est considéré comme « non catch-all ».

3.4. Suspicion d'erreurs

3.4.1. Par matching interne

Il est rare que l'adresse e-mail soit la seule information dont nous disposions à propos d'un utilisateur. Souvent, des informations annexes telles que le nom, le prénom ou un nom d'entreprise sont présentes dans l'adresse. Dans certaines bases de données dont nous avons pu étudier des échantillons, contenant des adresses électroniques, nom et prénom de personnes s'étant adressées à un service, nous avons pu retrouver le nom ou le prénom dans près de 85 % des adresses, sur la base de données comprenant plus de 50.000 adresses distinctes (le même test sur les noms de domaine nous a permis de trouver une correspondance dans 2 % des cas). L'intérêt de trouver un nom dans une adresse va au-delà de l'intérêt purement statistique : nous avons pu identifier de nombreux cas dans lesquels le nom ou le prénom étaient presque présents, mais pas exactement, suggérant une possible faute de frappe, soit dans l'adresse, soit dans le nom.

La suite de l'exposé ici n'a rien de déterministe, comme c'est le cas en général lorsqu'il est question de qualité dans les bases de données. Il s'agira de faire de son mieux pour retrouver une correspondance, exacte ou approchée. Il est évident qu'on ne pourra pas identifier tous les cas de correspondance, ni toutes les erreurs.

Il existe de nombreux algorithmes permettant de calculer la similitude entre deux chaînes de caractères, tels que l'algorithme de Levenshtein ou de Jaro [3]. La difficulté réside ici dans son application. Prenons pour exemple Albert Leroy, qui fournit comme adresse leryo.albert1980@gmail.com. On peut dans un premier temps enlever tous les chiffres, en supposant qu'ils ne feront pas partie du « matching » (bien qu'il soit fréquent de voir, volontairement, un zéro replacer un « o », un quatre replacer un « A » ou un trois replacer un e). La présence du point nous facilite ensuite le travail. Il est raisonnable de penser, si l'adresse contient les nom et prénom de la personne, que le point (ou un tiret, ou un *underscore*) indiquera la séparation. Nous pouvons dès lors comparer « leryo » et « albert » avec « albert » et « leroy » (après avoir mis tout en minuscule et supprimé les accents, cédilles et signes tels que l'apostrophe ou le tiret).

Nous pouvons d'abord chercher le prénom en première place et le nom en seconde place. L'algorithme de Jaro donne une similitude de 100 % entre deux mots identiques. En dessous de 90 %, on peut considérer que les deux mots sont trop différents pour que l'un puisse résulter d'une faute de frappe par rapport à l'autre. Avec l'algorithme de Jaro, nous obtenons une similitude de 70 % entre « albert » et « leryo », ainsi qu'entre « leroy » et « albert », ce qui est donc insuffisant pour constituer une correspondance. Par contre, si l'on suppose que le nom précède le prénom, « albert » et « albert » sont, bien évidemment, à 100 % de similitude, tandis que « leroy » et « leryo » sont à 93.3 %. On peut donc déduire qu'il y a beaucoup de chances qu'il y ait une faute de frappe. Selon le contexte et le caractère stratégique de l'information, il faudra éventuellement faire une vérification humaine, avec des suggestions de correction.

Si l'adresse fournie était par contre albertleroy@gmail.com, il pourrait être un peu plus compliqué de faire la correspondance. On pourrait certes essayer les dix découpes possibles. L'inconvénient est que l'on risque de suspecter une faute de frappe avec la découpe « alber » et « tleroy », les deux chaînes étant respectivement proches de « albert » et « leroy ». Notre proposition est la suivante : si soit le nom, soit le prénom, est exactement présent dans l'adresse, alors on le supprime, et on essaie de trouver une correspondance avec ce qui reste. Par exemple, si l'adresse est albertleryo@gmail.com, étant donné que le prénom « albert » est exactement présent, on peut l'extraire, et évaluer la similitude entre « leryo » et le nom « leroy ».

Bien évidemment, s'il y a une faute de frappe à la fois dans le nom et le prénom, elles ne seront pas détectées avec cette technique.

Par ailleurs, il faut noter qu'une proximité entre le nom et une partie de l'adresse n'est pas nécessairement signe d'une erreur, mais peut simplement permettre d'identifier des cas suspicieux (voir Annexe 7.2). Nous avons trouvé de nombreux cas de personnes dont le nom légal est écrit dans une langue et l'adresse contient le nom dans une autre (tels que Alexandre et Alexander), ou l'adresse contient un diminutif ou un surnom (Jacques et Jacky), ou contiennent deux translitérations

différentes d'un même nom (Ismaël et Ismaïl). Il faudra donc en général une intervention humaine, soit de l'utilisateur même (avec idéalement des suggestions de correction), soit d'un administrateur, pour déterminer dans quel cas nous nous trouvons (voir Section 3.10).

Tout comme nous l'avons fait pour les erreurs syntaxiques, nous pouvons suggérer des corrections lorsque quelque chose de suspect est détecté. Typiquement, si nous recevons le triplet (Albert, Leroy, albert.leryo@gmail.com), nous pouvons proposer deux suggestions :

- (Albert, Leroy, albert.leroy@gmail.com)
- (Albert, Leryo, albert.leryo@gmail.com)

3.4.2. Noms de domaine fréquents

Selon nos tests, nous avons pu déterminer que bon nombre d'adresses sont erronées à cause d'une faute de frappe manifeste. Dans la section précédente, nous avons vu quelques méthodes permettant d'en identifier dans le nom d'utilisateur. Nous avons pu également identifier beaucoup d'erreurs en reprenant la liste des noms de domaine présents dans la base de données et proches d'un nom de domaine fréquent, tels que Gmail, Hotmail ou autre. Diverses techniques, telles que l'utilisation de métaphone, de soundex ou d'algorithmes de similitudes comme Jaro présenté précédemment, peuvent être utilisées. Le tableau ci-dessous présente quelques noms de domaine retrouvés dans nos échantillons, mettant en évidence des noms plus que suspects. La première ligne indique chaque fois le nom de domaine supposé correct.

hotmail	**gmail**	**telenet**	**yahoo**	**skynet**
hotmal	gmali	telent	yaho	skyent
hotamil	gmal	telenet1	yahioo	skyenet
hotmial	gmil	telnet		skynette
hotmaill	gmailo			sjynet

Notons que certains de ces noms de domaine (les TLD ont été ignorés dans le tableau) existent, mais la probabilité qu'il y ait en effet une faute de frappe est élevée. En se basant sur une liste de noms de domaine connus, il est assez facile de mettre en évidence une suspicion d'erreur et de suggérer une correction.

3.5. Matching et dédoublonnage

Des techniques classiques et similaires à celles utilisées ci-dessus peuvent permettre de détecter des doublons dans une base de données. Il faut cependant adapter quelque peu les techniques aux adresses e-mail, comme cela doit être le cas pour tout type de donnée spécifique. Par exemple, on pourrait considérer que la probabilité que les adresses

« albert.leroy@telenet.be » et « albert.leroy@yahoo.fr » appartiennent à une seule et même personne, est plus élevée que pour « albert.leroy@yahoo.fr » et « albert.lenoy@yahoo.fr », bien que les deux derniers soient bien plus proches que les deux premiers, si l'on considère simplement la proximité entre ces deux chaînes de caractères.

Nous pourrons donc nous servir d'informations accessoires classiques, les mêmes qui permettent de différencier un doublon d'un cas d'homonymes dans une base de données : un numéro national, une date de naissance, une (partie d') adresse postale, un numéro de téléphone, …

Cependant, nous avons remarqué que les gens gardent souvent la même graphie entre deux fournisseurs : dans la mesure du possible, beaucoup préfèrent éviter d'avoir comme nom d'utilisateur albert.leroy chez un fournisseur et leroy_albert chez un autre. En nous basant sur cette hypothèse (et en la vérifiant avec des informations annexes), nous avons pu notablement améliorer nos résultats.

Quelle que soit l'interprétation retenue, il sera opportun, selon les enjeux, de conserver un historique des enregistrements, de leur statut et de la date de leur traitement (voir Section 3.7).

3.6. Batch ou on-line ?

Pour savoir jusqu'où aller dans les différentes techniques présentées plus haut, il y a lieu de se demander si on traite des données « hors-ligne » (batch), ou s'il agit d'un processus d'enregistrement d'un citoyen ou d'un client (on-line).

Dans le premier cas, où on traite de volumineux listings, hétérogènes et concurrents, qui n'avaient au préalable pas fait l'objet de tests, les questions de performance importent peu. Si certains tests automatiques prennent quelques secondes, voire minutes, ce n'est pas fondamentalement un problème, pour autant que le temps moyen des tests ne soit pas trop long et que les ressources permettent de traiter l'ensemble de la base de données dans un temps raisonnable. On peut par ailleurs reporter un test qui, provisoirement, n'a pas abouti, typiquement parce qu'un serveur est temporairement inaccessible.

Par contre, en cas de doute, il ne sera pas possible de demander directement à l'utilisateur s'il peut confirmer ses données. D'où l'intérêt d'aller le plus loin possible dans les tests. Au mieux, on peut faire en sorte que lors d'une prochaine connexion, l'une ou l'autre question soit posée à cet utilisateur.

Dans le second cas, au moment où l'utilisateur renseigne son adresse e-mail, la bonne pratique (malheureusement pas toujours d'application à l'heure actuelle) consiste à envoyer un e-mail de confirmation à l'adresse fournie (ce qui n'est bien évidemment pas envisageable pour le « batch »). Dans ce cas, il n'est pas nécessaire d'être très restrictif par rapport à la syntaxe, puisqu'une adresse ne rentrera effectivement dans le système qu'après la validation. Il est par contre plus que recommandé d'avoir une

vérification efficace, de façon à éviter la majorité des fautes de frappe et que l'utilisateur attende inutilement la suite du processus. On peut également, comme suggéré en Section 3.2.1, émettre des doutes sur certaines adresses (contenant des accents, par exemple) et demander à l'utilisateur de confirmer son adresse, en mettant en évidence les éléments suspicieux. Comme nous l'avons montré dans notre « Proof of Concept », il n'est pas très difficile de suggérer à l'utilisateur diverses corrections en cas de suspicions d'erreur.

Il n'est pas non plus envisageable, ni nécessaire, de vérifier l'existence de l'adresse fournie avant d'y envoyer l'e-mail de confirmation. D'une part parce que cette vérification, peu fiable, prend parfois beaucoup de temps, d'autre part parce qu'elle est redondante par rapport à l'envoi de l'e-mail.

3.7. Historique de la validité dans le temps et monitoring

Le caractère évolutif de l'usage des adresses e-mail est un facteur fondamental, comme nous l'avons noté en Section 2.7.2.

Un suivi professionnel d'une base de données ne peut se faire qu'en conservant un historique des événements et indicateurs de qualité.

Dès lors, que l'on se trouve dans un scenario on-line ou batch, il est indispensable, une fois la stratégie de gestion établie, de conserver, en relation avec la base de données d'adresses e-mail, un historique des événements et indicateurs de qualité associés.

L'annexe 7.2 en fournit plusieurs exemples : pour chaque instance, un timestamp relié à chaque événement, qu'il s'agisse de la détection d'une erreur de syntaxe ou de la réception d'un e-mail de confirmation (lien cliqué), devra être automatiquement stocké tout au long du cycle de vie de la base de données d'adresses e-mail. À ces événements pourront correspondre des actions ou scénarios de traitement (automatiques ou demandant une intervention humaine dans certains cas critiques) dont le déclenchement ainsi que le résultat éventuel devront également faire l'objet d'un historique.

La typologie des erreurs ou anomalies, des événements et actions associées sera préalablement définie et établie en collaboration avec les utilisateurs, responsables du domaine d'application et gestionnaires de la base de données.

À partir de ces informations, un monitoring de la base de données pourra être effectué en vue d'en assurer la bonne gestion dans le temps sur la base d'indicateurs de qualité (plusieurs statistiques détaillées ont été proposées en Section 2.7). Par exemple, des actions ponctuelles pourront être entreprises (en vue d'une meilleure visibilité) si l'on constate, pour une application stratégique, un taux jugé excessif de liens non cliqués. À l'inverse, on pourra également observer les améliorations de la qualité des adresses e-mail dans le temps.

Nous avons exposé dans des publications précédentes les spécifications conceptuelles et logiques en vue de mettre en place un historique des

événements et anomalies associés à une base de données, avec l'implémentation correspondante [4] [2] [5]. Ces principes restent applicables pour des bases de données répertoriant des adresses e-mail.

3.8. Traitement on-line

En conclusion de tout ce que notre étude nous a enseigné, et des nombreux contacts que nous avons eus avec diverses équipes opérationnelles, nous suggérons maintenant une séquence d'actions à entreprendre, d'abord dans le cadre de la mise en place d'une plateforme on-line, puis dans le cas d'un traitement batch de données préexistante. Notons que beaucoup de ces étapes sont déjà d'application, ou en voie de l'être, dans de nombreux projets. Nous proposons cependant d'aller plus loin sur divers aspects.

Toutes ces méthodes ont été testées et mises en place, pour le premier cas dans un « Proof of Concept » (PoC) que nous avons développé, d'autre part au travers de l'outil « Trillium » de IntoDQ (voir Section 4.4.2) d'analyse de base de données, ainsi que d'un logiciel ad hoc que nous avons développé spécifiquement, tant pour l'analyse statistique que pour les vérifications syntaxiques et d'existence d'adresse.

Dans les propositions qui suivent, nous essaierons d'atteindre le maximum de ce qui est raisonnable de faire en vue de maintenir une bonne qualité dans les adresses e-mail, en étant parfois un peu intrusif ou contraignant. Il va de soi que, en fonction du contexte et de l'investissement à mettre en œuvre, certaines parties devront être ajoutées, supprimées ou adaptées.

Nous considérons ici le scénario d'un portail où l'utilisateur renseigne une adresse e-mail, y reçoit de temps à autre des messages, et est amené à s'y connecter régulièrement.

3.8.1. Mise en place de tests en entrée

Il y a deux moments où un utilisateur peut renseigner une adresse e-mail : soit au moment de l'enregistrement, soit lorsqu'il désire la modifier. Nous présentons d'abord le premier cas. Le second étant très similaire, nous n'en exposerons que les différences.

1. Idéalement, le portail nécessitera une authentification forte, à l'aide par exemple d'une carte eID (carte d'identité électronique) ;
2. Vérification syntaxique générale et spécifique. Il n'est pas nécessaire d'être très contraignant, étant donné l'envoi d'un e-mail de confirmation ci-dessous :
    a. En cas d'erreur certaine : refuser l'adresse, suggérer des corrections ;
    b. En cas de suspicion : le signaler, suggérer des corrections ;

3. Validation partielle : existence du TLD, existence du nom de domaine (éventuellement en background) :
    a. En cas d'erreur certaine : refuser l'adresse, suggérer des corrections ;
    b. En cas de suspicion : le signaler, suggérer des corrections ;
4. Si les nom et prénom sont donnés, détecter des petites différences. Le cas échéant, les signaler, suggérer des corrections ;
5. Si applicable, vérifier que l'adresse ne soit pas déjà présente dans la base de données ;
6. Envoi d'un e-mail de confirmation (« double opt-in »), avec action obligatoire (soit un lien à cliquer, soit un code à fournir). La page de confirmation nécessitera une authentification, du même type qu'au premier point. Tant que l'adresse n'a pas été confirmée, il faudra refuser l'accès au portail ;
7. Il faut prévoir un scénario alternatif dans le cas où le lien de confirmation n'est jamais cliqué, par exemple parce l'adresse fournie n'existe pas. Il doit pouvoir en fournir une nouvelle, sans être définitivement bloqué dans le processus d'enregistrement.

En cas de modification de l'adresse, un scénario quasiment identique pourra être appliqué, avec quelques possibilités supplémentaires en attendant que l'utilisateur clique sur l'e-mail de confirmation :

- Le plus contraignant : déconnecter l'utilisateur dès l'envoi de l'e-mail de confirmation et ne plus l'autoriser à se connecter tant que l'adresse n'est pas confirmée. Il faut cependant lui permettre de modifier cette adresse, mais cela peut être la seule action possible sur le portail ;
- Le moins contraignant : tant que la confirmation n'a pas été faite, autoriser la connexion au portail, mais afficher un message visible signalant que l'adresse n'a pas encore été confirmée ;
- Scénario intermédiaire : rester dans le second scénario pendant quelques jours, puis passer dans le premier si l'adresse n'est toujours pas confirmée.

3.8.2. Suivi de la validité dans le temps

Comme nous l'avons montré en Section 2.7.2, s'il est essentiel de valider correctement les adresses rentrant dans le système, la difficulté principale est de maintenir ces adresses à jour.

À chaque fois que la plateforme enverra un e-mail à l'utilisateur, on mettra en place les mécanismes de vérification de consultation (Section 2.6). En cas de confirmation de lecture, on mettra à jour un champ de dernière confirmation dans la base de données.

Si par contre un message d'erreur (de type « bounce ») est reçu lors de l'envoi, quelques actions seront à envisager :

- Marquer et dater l'incident dans la base de données ;
- Signaler l'incident à l'utilisateur à sa prochaine connexion, en lui suggérant de modifier son adresse ;
- En fonction du caractère critique de l'adresse, en informer ou non les administrateurs/gestionnaires ;
- Si un « hard bounce » se produit un certain nombre de fois d'affilée[26], une action plus drastique peut être envisagée, comme par exemple la suppression de l'adresse.

Si la dernière confirmation de lecture est récente (la durée est à déterminer en fonction du contexte) et qu'il n'y a pas eu d'erreur lors d'un envoi d'e-mail, on peut considérer que la donnée est de bonne qualité. Si par contre la dernière confirmation est trop ancienne, plusieurs scénarios peuvent être envisagés :

- Afficher un message sur le portail : « L'adresse … est-elle toujours valide ? ». Si l'utilisateur valide ce message, on peut mettre à jour cette information dans la base de données. L'avantage est que si l'adresse n'existe plus, l'utilisateur verra malgré tout ce message. L'inconvénient est qu'il ne s'agit pas réellement d'une validation : l'utilisateur peut très bien valider une adresse erronée ;
- Similairement, on peut envoyer un e-mail avec le même message. L'inconvénient est que si l'adresse n'existe plus, personne ne verra ce message. L'avantage est qu'il s'agit d'une validation effective ;
- On peut également mettre un scénario de connexion alternatif tant que l'utilisateur ne confirme pas l'adresse ou ne la met pas à jour, on ne l'autorise plus à se connecter.

Rappelons que, comme déjà mentionné à la fin de la Section 2.5.2, certains fournisseurs recyclent les adresses inutilisées depuis un certain temps. Sachant que, en général, une fois que l'on a accès à une adresse e-mail, il est facile de réinitialiser le mot de passe, ce comportement des fournisseurs rend peu sûr le mécanisme d'authentification par e-mail et mot de passe, surtout pour des adresses qui n'ont plus été confirmées depuis un certain temps. Une authentification liée à une eID (carte d'identité électronique) permet dès lors d'améliorer nettement la sécurité.

[26] Nous avons observé des cas d'envoi d'e-mail ayant généré un « hard bounce », mais pour lesquels nous avons eu une confirmation de lecture lors d'un envoi ultérieur, démontrant une fois de plus le manque de fiabilité des messages d'erreur.

3.9. Traitement batch des fichiers existants

Le traitement batch de fichiers existants se fera typiquement soit avant d'intégrer une base de données à un nouveau système, soit régulièrement sur un système existant. Ce traitement batch permettra à la fois d'améliorer la qualité des données, mais également de mesurer l'efficacité de mesures prises ou de s'assurer que les processus de contrôles sont efficaces.

Voici une série de traitements qui peuvent être effectués et que nous avons mis en place sur diverses bases de données. Idéalement, le résultat de chacun de ces tests devra être enregistré dans la base de données.

- Vérification syntaxique poussée et spécifique, avec toujours trois catégories : correctes, incorrectes et suspectes. Cette dernière nécessitera un traitement manuel. Pour les adresses syntaxiquement incorrectes, il faudra se positionner : soit les supprimer, soit les modifier, soit simplement les « tagger » comme incorrectes ;
- Validation des noms de domaine. À nouveau, trois catégories : correct, incorrect (à corriger ou supprimer), temporairement inconnu. Dans ce dernier cas, il faudra reprogrammer une validation ultérieure ;
- Validation des adresses. On aura cette fois-ci plus de cas :
  - Adresses valides,
  - Adresses non-valides,
  - Serveur « catch-all », on sait donc qu'on n'aura jamais de réponse,
  - Problème temporaire,
  - Statut inconnu (réponses incompréhensible ou non-conforme) ;
- Matching interne (présence approximative du nom ou prénom), pour détecter des fautes de frappe potentielles ;
- Dédoublonnage, pour détecter des présomptions de doublons ;

3.10. Organisation

Que le suivi de la qualité des adresses e-mail s'effectue en batch, via les Data Quality tools ou on-line, via un développement spécifique, les bases de données doivent idéalement être liées à un portail. Si l'organisation concernée inclut un logiciel de CRM, celui-ci doit y figurer également, pour la gestion des *bounces*, notamment. Il en va de même de tout autre outil spécifique à la gestion des adresses e-mail (voir Chapitre 4).

Les canaux de saisie de l'information et de gestion de celle-ci doivent être maîtrisés et documentés, de manière à éviter les incohérences liées à l'organisation interne du système d'information et à mettre en place un processus de gestion continue de la validité des e-mails.

Les procédures de validation et de gestion des adresses e-mail doivent être, comme nous l'avons détaillé dans ce rapport, automatisées en vue d'en maximiser le « *return on investment* ».

Toutefois, plus largement, un suivi concerté et organisé d'équipes reste indispensable tout au long du cycle de vie de la gestion des bases de données associées en ce qui concerne la gestion et la maintenance :

- des règles de validation,
- du monitoring associé à l'historique des événements,
- du traitement résiduel des adresses e-mail incorrectes (dans le cas des fichiers batch par exemple) ou dans les cas critiques, lorsque l'on se trouve dans un scenario on-line,
- des stratégies et actions de suivi de la qualité des données,
- de la documentation.

Les intervenants humains collaborant à l'ensemble de l'organisation sont les fournisseurs des données, les décideurs, propriétaires et gestionnaires de la base de données et du portail ainsi que les utilisateurs internes, en ce compris les gestionnaires de la documentation du système.

4. Panorama d'outils existants sur le marché

Nous allons dans ce chapitre présenter divers outils permettant d'appliquer une partie des conseils que nous présentons dans ce document. Les premiers outils concernent la validation syntaxique ; les suivants permettent de valider l'existence d'une adresse ; ensuite nous présenterons quelques outils permettant de s'assurer qu'un e-mail a bien été lu ; les deux dernières sections présenteront des outils plus professionnels tels que des « Data Quality Tools » et des CRM.

4.1. Vérificateurs syntaxiques

Il est très facile de trouver sur le web des outils permettant de valider la syntaxe d'une adresse e-mail, soit au travers d'expressions régulières, soit au travers de librairies. Toutes celles que nous avons croisées jusqu'ici ont principalement trois défauts :

- Toutes considèrent qu'une adresse est soit correcte, soit incorrecte. De ce fait, elles sont soit très contraignantes et bloquent bon nombre d'adresses valides, soit trop permissives, autorisant des adresses syntaxiquement incorrectes. Nous pensons qu'il est nécessaire d'avoir une catégorie « adresse suspicieuse », comme présenté en Section 3.2.1 ;
- Aucune solution que nous avons trouvée ne propose une vérification spécifique pour certains noms de domaine, comme nous le proposons ici. Or, cela permettrait facilement d'être beaucoup plus précis pour un très grand nombre d'adresses ;
- Nous pensons que vérifier l'intégralité des contraintes au travers d'expressions régulières est illusoire : cela amène soit à des expressions horriblement complexes et incompréhensibles, soit à des expressions très peu contraignantes.

Voici cependant quelques pistes :

- Dominic Sayers propose sur http://isemail.info/ une série de librairies en C#, Java ou PHP. Il essaie au mieux de se conformer

aux standards (principalement les RFC 5321 et 5322) en ce qui concerne la syntaxe, mais vérifie également l'existence du nom de domaine. C'est à notre connaissance une des librairies les plus avancées à ce jour.

- Sur la page http://www.regular-expressions.info/email.html, on trouvera quelques discussions et suggestions dans le but de trouver une expression régulière la plus adaptée.
- En HTML 5, il existe pour les formulaires un champ « email » qui effectue un contrôle syntaxique avant la soumission. Il utilise l'expression régulière suivante :

```
^[a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@
[a-zA-Z0-9-]+(\.[a-zA-Z0-9-]+)*$
```

  Cette expression est cependant largement insuffisante : les accents ou caractères non latins ne sont pas autorisés, mais par contre aucun contrôle n'est effectué sur la présence de points successifs dans le nom d'utilisateur, et le nom de domaine « --- » est par exemple considéré comme valide.
- Dans la section suivante, nous présenterons deux outils de validation effectuant également des techniques de vérification syntaxique avancée.

4.2. Testeurs d'existence

Un certain nombre d'outils permettent de vérifier l'existence d'une adresse e-mail, mais ils n'y arrivent pas tous avec le même succès. Ces outils sont souvent gratuits lorsqu'il s'agit de vérifier quelques adresses (avec un nombre maximum par heure ou par jour), mais deviennent en général payants pour la validation en batch (avec des prix dépassant parfois les 350$ par mois). Nous pouvons citer :

- http://verify-email.org/
- http://tools.email-checker.com/
- http://www.verifyemailaddress.org/
- http://www.ip-tracker.org/checker/email-lookup.php
- http://bulkemailverifier.com/

Cependant, comme déjà mentionné précédemment, pour une adresse incorrecte chez Yahoo (serveur « catch-all » lorsqu'il ne parvient pas à identifier la source), seuls les deux premiers outils l'ont en effet constaté. Les deux suivants indiquent qu'elle existe, et le dernier est incapable de répondre. Par ailleurs, tous ces outils vérifient en premier lieu la syntaxe, avec des tests soit trop laxistes (acceptant par exemple comme adresse « ..@--.be »), soit trop contraignants (utilisant par exemple le champ « email » de HTML5, qui rejette les accents), soit les deux (acceptant « ..@--.be » mais refusant les accents).

Il existe également une série de petits logiciels, en général payants, proposant le même genre de service, en permettant d'importer une liste

d'adresses (une version gratuite mais limitée est en général disponible). Mais on retrouve les mêmes défauts que les outils en ligne proposés ci-dessus. On pourra citer « eMail Verifier (Maxprog) », « Advance Email Verifier (G-Lock Software) » ou encore « Smart Email Verifier (DeskShare) »

4.2.1. ServiceObjects

Apparu récemment parmi les outils disponibles en « Data Quality » (nous n'en avons eu connaissance qu'après la rédaction de la majorité de ce document), ServiceObjects (http://www.serviceobjects.com) propose les outils les plus performants que nous ayons trouvés à ce jour, particulièrement avancés pour la vérification des adresses e-mail, au travers d'API permettant d'interroger leurs serveurs. Ils proposent de corriger dans une certaine mesure les adresses erronées, détectent les erreurs syntaxiques spécifiques à certains noms de domaine et identifient les serveurs catch-all. Par ailleurs, les adresses obtiennent un score entre 0 et 4, allant de valide à invalide, en passant par probablement valide, inconnu ou probablement invalide.

L'utilisation de services web, avec réponse aux requêtes en XML ou JSON rend aisée l'intégration dans n'importe quel service (pour autant que l'on soit prêt à « délocaliser » les contrôles sur les adresses). La version « Platinium » de leur site web propose un abonnement mensuel de 40.000 transactions pour 479 $, soit à peu près 1,2 centime de dollars par transaction.

Il nous semble cependant que les tests effectués sont perfectibles. Nous avons trouvé un certain nombre d'erreurs spécifiques à des noms de domaine qui n'étaient pas détectées et les adresses avec accent sont toujours considérées comme fausses. Il faut également noter qu'il ne s'agit pas d'un Data Quality Tool à proprement parler, mais simplement d'un service de validation.

4.2.2. EmailVerify for .NET

Également très avancée, cette libraire « .NET » (http://cobisi.com/email-validation/.net-component) propose une série de vérifications tant syntaxiques qu'en termes d'existence. Les vérifications syntaxiques tiennent compte des caractères internationaux, ainsi que de la syntaxe spécifique à certains noms de domaine. Par ailleurs, la librairie prend également en considération les serveurs « catch-all » et les « greylists ».

EmailVerify tente également de détecter les adresses « disposables » (ou jetables), qui sont des adresses temporaires qui redirigent pendant une période déterminée les messages vers une adresse définie, mais qui disparaissent ensuite. Ce type d'adresse permet d'éviter le spam, par exemple lorsqu'il est obligatoire de fournir une adresse e-mail pour obtenir un service occasionnel, mais pour lequel on ne désire plus être contacté par la suite (achat en ligne, inscription à un évènement, …).

Par rapport à la solution ServiceObject (Section 4.2.1), celle-ci à l'avantage de pouvoir tourner sur une machine que l'on contrôle, ce qui évite de devoir envoyer ses adresses vers un tiers. Par contre, il sera nécessaire que l'application puisse tourner sur une machine configurée correctement pour bénéficier de la confiance des serveurs SMTP.

Nous ne sommes malheureusement pas parvenus à étudier la performance de la librairie, leur démo en ligne n'étant pas fonctionnelle au moment d'écrire ces lignes.

4.3. Outils de suivi

Quelques outils sont proposés sur le Web pour vérifier qu'un e-mail envoyé a bien été lu. Aucun des outils présentés ci-dessous ne pourrait cependant être intégré dans un portail.

- http://bananatag.com/ : cette solution puissante peut être intégrée à Gmail ou Outlook, mais peut également être utilisée depuis n'importe quel client (en ajoutant « .btag.it » à l'adresse du destinataire). Il ajoute une image invisible et convertit tous les liens, en utilisant les techniques décrites dans ce document. Ils proposent une version gratuite, limitée à 5 e-mails par jour, ou plusieurs versions payantes.
- http://www.spypig.com/ : ce site permet de générer du code HTML référençant une image, que l'on intègre ensuite soi-même manuellement, dans les e-mails à envoyer. Au moment d'écrire ces lignes, le service n'était cependant pas fonctionnel. D'autres sites web (par exemple http://mobileshortcut.com/TAILMAIL/) proposent le même type de service. Vu le nombre d'étapes manuelles à suivre, cette solution convient uniquement pour tracer des envois très occasionnels.
- http://www.msgtag.com/ : ce petit logiciel joue un rôle de « proxy SMTP », et marche, sous Windows, pour tous les clients mail de type Outlook, Eudora…, mais pas pour les « webmails » (par exemple Gmail ou Hotmail, via leur site web). Il faut configurer son client mail pour se servir de MsgTag comme serveur SMTP. Il traite ensuite les messages, puis les fait suivre vers un «vrai » serveur SMTP. Il agit en insérant une image (visible dans la version gratuite) au bas de l'e-mail, mais il ne transforme pas les liens. Quand un e-mail a été lu, il envoie un e-mail de confirmation pour la version gratuite, et propose une interface plus élaborée pour la version payante (que nous n'avons pas testée).

4.4. Data Quality Tools

Les analyses batch de données se font typiquement avec des outils professionnels très avancés. Peu d'entre eux possèdent des modules spécialement destinés aux e-mails, mais ils sont en général suffisamment paramétrables pour permettre de nombreuses analyses. Mis à part le premier que nous présenterons, il s'agit de logiciels coûtant plusieurs centaines de milliers, et ne sont donc accessibles qu'à de grosses entreprises.

4.4.1. OpenRefine (Google refine)

Logiciel libre créé par Google mais repris en open source, OpenRefine (http://openrefine.org/) est une solution permettant de manipuler une base de données ou une feuille de calcul, de façon à corriger les erreurs, incohérences ou duplications d'information. Contrairement aux solutions suivantes, OpenRefine base ses transformations sur l'application d'une fonction sur une colonne (soit pour en créer une nouvelle, soit pour en modifier une), mais dès que la fonction est appliquée, la nouvelle colonne est indépendante de l'originale, et la fonction est « perdue ». Il est donc difficile de revenir en arrière pour affiner la fonction, ou pour modifier les données d'entrée. De ce fait, OpenRefine ne convient à notre avis pas aux projets de grande ampleur, avec de multiples fonctionnalités à mettre en place.

Il est cependant aisé d'extraire le nom de domaine et les fonctions de « clustering » permettent de regrouper, par exemple, des noms de domaine ou des adresses très proches (en utilisant l'algorithme de Levenshtein ou les métaphones). On peut également appliquer des expressions régulières pour vérifier la syntaxe. Il est par ailleurs possible de vérifier des syntaxes spécifiques, mais le processus sera très laborieux.

4.4.2. IntoDQ (Trillium)

IntoDQ (http://www.intodq.com) est le logiciel que nous avons utilisé durant cette étude, ainsi que pour les projets dans lesquels la « Data Quality Cel » de Smals est impliqué. Il ne possède pas de fonctionnalité spécifiquement dédiée aux e-mails, mais sa grande souplesse nous a permis de développer toutes les analyses batch que nous avons présentées dans ce document.

Nous n'avons pas mis en place de validation de nom de domaine ou d'adresse, car cela nécessiterait le développement d'un module externe communiquant avec Trillium.

4.4.3. RedPoint

RedPoint (http://www.redpoint.net/) est un concurrent d'IntoDQ de moindre envergure, qui ne possède pas non plus de module spécifique

pour les adresses e-mail (bien que la société de conseils Gartner[27], référence en matière de technologies, le répertorie parmi les outils « Data Quality » adaptés aux e-mails dans leur « Magic Quadrant for Data Quality Tools »). Sa grande différence par rapport à IntoDQ, mise à part une interface plus moderne et conviviale, est l'utilisation de modules probabilistes (s'apparentant à des « black boxes », c'est-à-dire des outils dont on connait les fonctionnalités mais pas le fonctionnement interne) pour effectuer les matchings et dédoublonnages. Cela peut apporter des résultats plus puissants, mais au prix d'une perte de contrôle fin.

4.4.4. Human Inference

Human Inference (http://www.humaninference.com/), récemment rachetée par la société française « Neopost » (http://www.neopost.com) propose un module spécifiquement dédié aux adresses e-mail. Après une vérification syntaxique (à notre sens relativement élémentaire), l'outil consulte une base de connaissance interne pour voir si le nom de domaine existe. Si ce nom de domaine ne se trouve pas dans cette base de connaissance, une requête DNS est effectuée, mettant à jour la base de connaissance. Il s'agit donc d'un compromis élégant entre l'impossibilité de stocker la totalité des noms de domaine valides dans une base de données et la vitesse de vérification.

Dans le cas où un nom de domaine n'existe pas, HumanInference se sert de sa base de connaissance pour suggérer une ou plusieurs alternatives. Il n'y a par contre pas de suggestions de corrections en cas d'erreur syntaxique. L'existence des adresses n'est pas évaluée.

4.5. Outils CRM

Lorsque l'on dispose d'un outil CRM (Customer Relationship Management, tel qu'Oracle Siebel chez Smals), celui-ci doit naturellement être intégré dans l'ensemble de l'architecture. Toutes les fonctionnalités de l'outil traitant les adresses e-mail, telles que le traitement des *bounces*, doivent être prises en compte dans la stratégie de gestion.

Les outils CRM méritent à eux seuls un document complet, mais le sujet dépasse clairement la portée de ce document.

[27] http://www.gartner.com

5. Conclusion

Une lecture rapide de ce document peut donner l'impression que la gestion des e-mails est une tâche très complexe et qu'il est impossible de maintenir une base de données de qualité décente. Heureusement, la réalité n'est pas aussi sombre. Si la tâche est effectivement complexe, à la suite entre autres d'une large part d'incertitude et d'un manque de déterminisme à de nombreux égards, les adresses e-mail ont le gros avantage par rapport à d'autres données, telles que les adresses postales ou les numéros de téléphone, qu'il est possible dans une large mesure d'en déterminer l'existence, et de s'assurer de leur actualité, sans se déplacer ni décrocher son téléphone. Ces étapes peuvent se réaliser de manière semi-automatique, d'où un *return on investment* très important qui découle de leur usage. Par ailleurs, prendre des mesures efficaces en vaut la chandelle : nombre de projets ont montré un intérêt notable, qu'il soit financier ou organisationnel, d'améliorer la qualité des adresses e-mail dans la base de données.

La principale raison de la nécessité d'une gestion professionnelle est la grande dégressivité de la qualité des adresses dans le temps. En effet, nos analyses ont montré que la volatilité des usages fait qu'à peine la moitié des adresses e-mail fournies il y a une dizaine d'années sont encore valables aujourd'hui. Il est donc nécessaire de mettre en place une stratégie de gestion efficace, avec un suivi historique des évènements et une structure organisationnelle à même de maintenir à jour les listes d'adresses e-mail. Il va de soi que cela ne peut se faire qu'avec la participation active des utilisateurs. Il faudra donc mettre tout en œuvre pour qu'ils aient l'envie ou le besoin de renseigner tout changement.

En matière de syntaxe, nous avons montré dans ce rapport qu'une séparation binaire entre adresse correcte et adresse incorrecte ne permettait pas de capturer toute la subtilité et la diversité des conventions adoptées par les fournisseurs. Il est nécessaire, pour éviter d'être soit trop contraignant, soit trop laxiste, de considérer une catégorie « suspicieuse », permettant d'attirer l'attention de l'utilisateur ou d'un gestionnaire sur une adresse potentiellement incorrecte, et de mettre en place les stratégies de gestion adéquates.

Par ailleurs, toujours en matière de syntaxe, nous avons montré que nombre de fournisseurs adoptaient une syntaxe bien plus simple que celle recommandée par les standards, permettant des contrôles plus précis,

sans risque de faux positifs ou négatifs. Considérer ces syntaxes spécifiques, dont nous fournissons quelques exemples en annexe, permet d'améliorer notablement la qualité et la précision des tests.

Dans ce rapport, nous n'avons pas étudié de façon détaillée les aspects légaux liés à l'utilisation des adresses e-mail (législation sur le respect de la vie privée, force probante…). Nous laissons ces aspects à des spécialistes.

La grande majorité des tests proposés dans ce rapport ont été implémentés soit dans un « Proof of Concept » que nous avons développé en interne, soit dans des outils de gestion de qualité de données (Data Quality Tools) et ont montré leur grande efficacité. Cependant, ces résultats et propositions étant tout récents, ils n'ont pas encore pu être implémentés dans un projet d'envergure, mis à part le développement et la mise en production d'une librairie Java novatrice et réutilisable en vue de tester les aspects syntaxiques des adresses e-mail.

Nous avons néanmoins l'intime conviction qu'une gestion professionnelle des adresses e-mail de clients ou de citoyens permet d'obtenir une qualité de données largement supérieure à celle que l'on peut espérer obtenir pour la collecte d'informations telles que l'adresse postale, le numéro de téléphone ou d'autres données. Il est bien entendu nécessaire que les utilisateurs aient un incitant à mettre à jour leurs données et que toutes les recommandations rassemblées dans ce document soient mises en œuvre. Un ROI très important en découlera.

6. Bibliographie

[1] D. Clément, B. Laboisse, D. Duquennoy et A. Micheaux, «Non qualité de données & CRM : quel coût ?,» chez *QDC 2008*.

[2] I. Boydens, «Strategic Issues Relating to Data Quality for E-government: Learning from an Approach Adopted in Belgium,» *Practical Studies in E-Government : Best Practices from Around the World,* pp. 113-130 (chapitre 7), 2011.

[3] Y. Bontemps, I. Boydens et D. Van Dromme, «Data Quality : tools,» Bruxelles, 2007.

[4] I. Boydens, Informatique, normes et temps, Bruxelles: Bruylant, 1999.

[5] I. Boydens, A. Hulstaert et D. Van Dromme, «Gestion intégrée des anomalies,» 2011. [En ligne]. Available: http://www.smalsresearch.be/publications/document?docid=62.

7. Annexes

7.1. Vérification syntaxique

7.1.1. Vérification syntaxique générale

Comme nous le présentons en Section 3.2, une séparation « adresse correcte/adresse incorrecte » est insuffisante. Il faut considérer trois catégories, pour lesquelles nous proposons des expressions régulières (certainement perfectibles) :

1. Les adresses certainement incorrectes, ne respectant pas :

   ```
   ^[^@,;:\s]+@([^@,;:\s+\.-]+([-\.][^@,;:\s+\.-]+)*)$
   ```

   Si le langage dans lequel l'expression régulière est utilisé supporte l'Unicode, on peut préférer l'expression suivante :

   ```
   [^@,;:\s]+@
   ([\p{L}0-9]+([-][\p{L}0-9]+)*[.])*[\p{L}]{2,}$
   ```

2. Les adresses certainement correctes, respectant :

   ```
   ^[a-z0-9\'\+_-]+(\.[a-z0-9\'\+_-]+)*@
   ([a-z0-9]+([-\.][a-z0-9]+)*)\.[a-z]{2,4}$
   ```

3. Les adresses suspicieuses, qui ne rentrent pas dans les catégories précédentes.

7.1.2. Vérification syntaxique spécifique

Pour un certain nombre de domaines, nous avons pu déterminer des règles syntaxiques plus contraignantes.

Domaine	Caractères permis et contraintes	Expression régulière[28]
Gmail	a-z 0-9 . +	^[a-z0-9.+]+$
	Entre 6 et 30 caractères, sans compter les « . » et ce qui suit le « + »	Sur le username sans les « . » et ce qui suit le « + » : ^[a-z0-9]{6,30}$
	Adresses de 8 caractères (sans compter les « . » ni ce qui suit le +) ou plus : au minimum une lettre	Sur le username sans les « . » et ce qui suit le « + » : ^([a-z0-9]{6 ,7})|([a-z0-9]*[a-z][a-z0-9]*)$
Yahoo	a-z 0-9 _ . Maximum un point Premier caractère : a-z Dernière caractère : a-z, 0-9	^[a-z][a-z0-9_]*[.]?[a-z0-9_]*[a-z0-9]$
	Entre 4 et 32 caractères	^[a-z0-9_.]{4,32}$
Hotmail, Live, Outlook, Belgacom, Skynet, Telenet, Pandora	a-z 0-9 - _ .	^[a-z0-9_-]+(\.[a-z0-9_-]+)*$
OVH (mx.ovh.net)	a-z 0-9 - _ .	^[a-z0-9_-]+(\.[a-z0-9_-]+)*$
	Entre 2 et 32 caractères	^[a-z0-9._-]{2, 32}$
Google Apps (aspmx.l.google.com)	a-z 0-9 - _ . % ‘	^[a-z0-9._%’-]+$

Remarques additionnelles :

- On suppose que les majuscules ont été converties en minuscules ;
- Lorsqu’il y a plusieurs expressions régulières pour un même domaine, elles doivent toutes être appliquées ;
- Yahoo : Il semblerait, bien que ça ne soit pas documenté à notre connaissance, que si l’adresse contient un point, celui-ci doit être suivi d’au moins quatre caractères, quel que soit le nombre de caractères précédant le point. Si cela s’avérait exact, l’expression régulière serait plutôt :
  "^[a-z][a-z0-9_]*([.][a-z0-9_]{3,})?[a-z0-9]$" ;
- Hotmail, Live, Outlook : on ne peut plus aujourd’hui créer une adresse qui ne commence pas par une lettre. Or nous avons connaissance d’adresses valides dans ces domaines commençant par des chiffres. Il semblerait donc que la syntaxe ait évolué ;
- Ces règles ont été relevées en 2013, mais pourraient évoluer ;
- En fonction du langage utilisé, il faudra ou non faire précéder les points et tirets du caractère d’échappement « \ ».

[28] Voir http://www.regular-expressions.info/ pour plus de détails sur la syntaxe.

7.2. Typologie des évènements

Nous reprenons ici à titre d'exemple une série non exhaustive d'évènements pouvant se produire au niveau des adresses e-mail d'une base de données, avec des suggestions d'action possible. Le tableau regroupe à la fois des scénarios batch et on-line. Nous renvoyons le lecteur au texte pour les différencier et pour plus de détails.

Les quatre premières catégories d'évènements (Syntaxe, TLD, Nom de domaine et Action sur e-mail de confirmation) sont plus orientées on-line dans le processus d'enregistrement à un portail, mais une partie des évènements se retrouve également dans des scénarios batch.

Type d'évènement		Statut	Actions possibles
Syntaxe	Erreur syntaxique générale forte	Erreur	Refuser l'accès Corriger manuellement (éventuellement avec suggestion) Supprimer de la DB
	Erreur syntaxique spécifique à un domaine	Erreur	Refuser l'accès Corriger manuellement (éventuellement avec suggestion) Supprimer de la DB
	Syntaxe suspicieuse	Suspicion	Suggérer une correction Confirmer l'adresse
TLD	TLD inexistant	Erreur	Refuser l'accès Corriger manuellement (éventuellement avec suggestion) Supprimer de la DB
Nom de domaine	Nom de domaine ou "MX Record" inexistant	Erreur	Refuser l'accès
	Timeout	Erreur temporaire	Réessayer plus tard
	Timeout répété (plus de X fois, avec délai de Y)	Suspicion	Considérer comme une erreur Action/décision manuelle
Action sur e-mail de confirmation	Lien de confirmation cliqué	Validation	Marque l'adresse comme validée dans la DB Ouvrir l'accès au système
	Lien non cliqué après un délai fixé (< N envois)	Suspicion	Renvoi de l'e-mail de confirmation
	Lien non cliqué après un délai fixé (≥ N envois)	Erreur	Arrêt des renvois de l'e-mail de confirmation Essayer d'autres moyens le cas échéant

Envoi d'e-mail	Hard bounce (< N fois, successif)	Suspicion	Attente de Hard bounces suivants
	Hard bounce (≥ N fois, successif)	Erreur	Suppression Essayer d'autres moyens le cas échéant Scénario alternatif de connexion, forçant à fournir une nouvelle adresse
	Soft bounce (< M fois, successif)	Suspicion	Attente des Soft bounces suivants
	Soft bounce (≥ M fois, successif)	Suspicion	Action manuelle Message sur le portail signalant le problème, l'invitant à changer d'adresse
Action sur e-mail	Lien redirigé cliqué	Validation	Mise à jour du timestamp de dernière validation
	Image ouverte	Validation	Mise à jour du timestamp de dernière validation
	Pas de validation depuis un délai fixé	Suspicion	Scénario alternatif de connexion, forçant l'envoi d'un e-mail avec lien à cliquer

7.3. Adresses e-mail officielles pour les citoyens

Lors de l'étude que nous avons menée sur la qualité des adresses e-mail dans une série de bases de données des administrations et qui a mené au présent document, mettant en évidence la grande difficulté à maintenir une adresse e-mail correcte pour chaque citoyen, une question nous a à plusieurs reprises été posée : pourquoi l'État n'offre-t-il pas à chaque citoyen une adresse e-mail « officielle », utilisée comme canal de communication (vers le citoyen, mais aussi vers les administrations) ? Ce citoyen serait alors responsable soit de la consulter fréquemment (via un « web-mail », ou en configurant son client mail de type Outlook), soit de la transférer vers une adresse de son choix, qu'il aura la responsabilité de mettre à jour (après authentification sur un portail avec la carte d'identité électronique). L'adresse en question pourrait par ailleurs servir d'adresse pour d'autres communications administratives (banque, assurance, contact avec la commune…), voire d'adresse personnelle.

Nous nous sommes donc demandé si la question était si simple à résoudre. Dans un premier temps, nous avons cherché à savoir si d'autres pays avaient fait ce choix. Nous n'avons jusqu'ici trouvé que trois exemples de pays ayant offert une adresse e-mail à chacun de ses citoyens (majeur ou dès la naissance) : l'Iran[29], la Turquie[30] et la Malaisie[31]. Tout lecteur connaissant d'autres exemples est invité à nous en informer ! On peut légitimement se demander si le but des autorités de ces pays était plus de faciliter la vie des citoyens que de mieux les contrôler...

Pourquoi plus de pays ne le font pas ? Voici quelques pistes de réflexions personnelles qui pourraient partiellement en expliquer les raisons.

Imaginons que l'adresse soit, par exemple, du format PRENOM.NOM@citizen.be. Il faudrait bien entendu gérer le problème des homonymes, sans que l'adresse n'en devienne impossible à retenir. Par ailleurs, il faut aussi décider si l'on peut réattribuer l'adresse d'un citoyen décédé ou ayant perdu sa nationalité belge. Cependant, le problème le plus important à nos yeux n'est pas là. Il serait alors impossible de ne pas donner son adresse e-mail à quelqu'un que l'on considère comme indésirable, puisqu'elle sera dès lors très facile à trouver. Par ailleurs, une personne se faisant harceler par e-mail ne pourra plus choisir de changer d'adresse pour redémarrer une nouvelle « vie numérique ». De plus, des adresses telles que elio.dirupo@citizen.be ou bart.dewever@citizen.be risqueraient fort de devenir inutilisables pour leur envoyer, par exemple, leur avertissement extrait de rôle ou la taxe de chef de ménage.

[29] http://www.hurriyetdailynews.com/iran-assigns-e-mail-addresses-to-citizens.aspx?pageID=238&nID=50339&NewsCatID=352 ou http://memeburn.com/2013/07/iran-to-give-all-citizens-state-controlled-email-addresses/

[30] http://www.itp.net/578521-turkey-to-give-all-citizens-national-email-addresses#.UmDiMlC-0yp

[31] http://thenextweb.com/asia/2011/04/26/malaysias-new-official-email-address-for-each-citizen-will-also-offer-biometric-usb-device/

On éviterait probablement une bonne partie des problèmes évoqués ci-dessus, en considérant une adresse contenant le numéro de registre national, par exemple 80.09.10-125.57@citizen.be. Cependant, outre le manque de convivialité de cette adresse, elle pose le problème qu'il sera alors très facile à des gens malintentionnés d'envoyer une publicité ou une tentative d'arnaque à toute la Belgique, voire à tous les hommes de plus de 50 ans, puisque le numéro de registre national contient à la fois la date de naissance (les 6 premiers chiffres) et le sexe (le 125 de l'exemple, impair pour les hommes, pair pour les femmes). Le numéro national des personnes nées en 2000 ou après est un petit peu différent[32].

Reste une possibilité d'attribuer une adresse avec une chaîne aléatoire de caractères ou de chiffres (084685468786@citizen.be). Il faudra dès lors prévoir que la longueur de la chaîne soit suffisamment longue. En effet, si celle-ci est composée de 8 chiffres (le minimum pour représenter les plus de 11 millions de citoyens belges), il sera également possible de spammer toute la Belgique.

On aura donc une adresse de facto impossible à retenir, ce qui aura pour conséquence qu'elle ne pourra que très difficilement être fournie soit à son entourage pour s'en servir d'unique adresse e-mail, soit à d'autres organismes (banques, commerces…). Elle ne sera donc réellement utilisable que par les administrations qui ont accès à cette adresse. Sachant que la plupart des gens transféreraient cette adresse vers une adresse qui leur est propre, cela reviendrait donc exactement au même que d'avoir une « e-box », comme on l'a maintenant en Belgique pour quelques services, ou au Danemark de façon généralisée (voir l'introduction), sur laquelle on peut référencer une adresse e-mail pour les notifications à chaque fois qu'un document ou un message arrive.

Nous avons donc la conviction qu'une « e-box » pour le citoyen reste la meilleure solution pour communiquer avec lui. En effet :

- Il peut changer à sa guise l'adresse e-mail de notification lorsqu'un nouveau message arrive (voire même choisir un autre média, tel que le SMS) et la garder privée ;
- Seuls les organismes autorisés pourront y placer des messages (il n'y a donc pas de risque de spam) ;
- On peut facilement voir si un document a été ouvert, et, au besoin et en fonction des enjeux, envoyer la version papier dans le cas contraire
- En associant « l'e-box » à une carte d'identité électronique, on peut s'assurer que c'est bien la bonne personne qui reçoit le message (ou à tout le moins qu'elle possède sa carte d'identité et son code), ce qu'on ne peut pas faire avec un e-mail ;

Le problème de la qualité des adresses présenté dans ce document reste donc bien présent.

[32] http://fr.wikipedia.org/wiki/Num%C3%A9ro_de_registre_national

7.4. Éviter les risques liés au spam

Toutes les sociétés qui utilisent abondamment les e-mails sont régulièrement confrontées au problème des courriers qu'elles envoient qui se retrouvent dans la boîte de spam de leurs destinataires, voire, pire, qui sont simplement bloqués en amont. Nous en décrivons ici quelques raisons, et les méthodes pour s'en prémunir autant que possible.

Parmi les diverses techniques visant à combattre le *spam* (ou courrier indésirable), l'une d'entre elles consiste à établir une liste d'expéditeurs (adresse e-mail ou adresse IP du serveur d'envoi) suspectés d'être source de spam. On parle de « blacklists ». Une fois dedans, il est très difficile d'en sortir. Nombreuses sont les entreprises qui l'ont appris à leur dépends, le jour où plus aucun des e-mails de leurs employés n'a pu atteindre ses destinataires. Il va de soi qu'en théorie, la grande majorité des expéditeurs blacklistés sont réellement des spammeurs, mais la technique a ses limites, et les « dégâts collatéraux » ne sont pas rares. Il existe plusieurs raisons de se retrouver dans ces listes. Dans le cadre de notre étude, deux de celles-ci méritent une attention plus soutenue.

La première se base sur le fait que les spammeurs ne se soucient en général guère de la qualité de leurs listings d'adresses e-mail, et s'ils font des efforts pour y ajouter des adresses, ils n'en font souvent aucun pour en supprimer les adresses obsolètes. De ce fait, les spammeurs envoient énormément d'e-mails à des adresses qui n'existent plus, tandis que des particuliers ou des entreprises soucieuses de la qualité de leur carnet d'adresses vont (ou en tout cas devraient) régulièrement le mettre à jour. Plusieurs gestionnaires conservent les anciennes adresses qui ont été désactivées, volontairement ou à la suite d'une trop longue période d'inactivité, et plus un expéditeur envoie d'e-mails à ces adresses, plus grande sera la chance (ou la malchance) qu'il se retrouve blacklisté. Il est donc essentiel de mettre en place un suivi dans le temps, de façon à minimiser l'envoi d'e-mails à des adresses qui n'existent plus.

Une seconde technique repose sur le fait qu'une des méthodes favorites de collecte d'adresses e-mail des spammeurs consiste à mettre en place un « robot » (ou *bots*) qui navigue de page en page sur le web à la recherche d'adresses e-mail (sur des forums, des pages personnelles ou professionnelles…). Le principe est de disséminer sur différentes pages des adresses pièges (ou « *honeypot* »), qui ne seront pas visibles par une personne « humaine » (texte blanc sur fond blanc, ou présent dans un élément caché), mais bien par les « *bots* ». Dès lors, tout envoi à ces adresses sera automatiquement considéré comme du spam, et l'expéditeur blacklisté. Bien sûr, une entreprise « normale » peut considérer qu'elle ne risque pas de se faire avoir par un « *honeypot* », puisqu'elle n'utilise pas ces techniques de « *bots* »… sauf qu'un utilisateur (humain ou automatique) malveillant peut parfaitement renseigner une adresse piège comme étant son adresse personnelle, dans le but que l'organisme visé se retrouve dans les blacklists dès qu'un e-mail lui est envoyé. Il est cependant facile de se prémunir contre ce risque : l'utilisateur malveillant n'a normalement pas accès à l'adresse piège, et ne

pourra donc pas recevoir un e-mail de confirmation qui lui serait envoyé lors de l'enregistrement. D'où la nécessité de mettre en place ce mécanisme de confirmation, bien connu mais trop souvent négligé.

Ces deux types d'adresses (inactives ou « *honeypot* ») sont généralement nommées « *spamtraps* » (piège à spam) et malheureusement très difficiles à détecter. En effet, si les listes de « *spamtraps* » étaient publiées, c'est tout le mécanisme qui s'effondrerait.

Il existe de nombreuses blacklistes, dont la plus répandue est sans doute « Spamhaus[33] », utilisée par 2/3 des fournisseurs d'accès à Internet. Autant dire que se retrouver dans cette liste peut être un véritable désastre ! Heureusement, une gestion professionnelle des adresses e-mail, telle que présentée dans ce document, permet de se prémunir d'une large part de ces problèmes.

Notons cependant qu'il existe d'autres facteurs qui augmentent le risque qu'un e-mail soit considéré comme du spam, en fonction par exemple d'une mauvaise utilisation du protocole SMTP, de l'utilisation de mots-clés ou d'images abondantes dans le texte. Il ne s'agit toutefois pas d'un problème de qualité de l'adresse e-mail proprement dite, mais de son utilisation. Nous sortons donc du cadre de ce document, et n'aborderons dès lors pas davantage cet élément.

[33] http://www.spamhaus.org/

8. Glossaire

Bounce mail ..... Message d'erreur reçu lors de l'envoi d'un e-mail. Il peut être « hard » en cas d'erreur définitive, ou « soft », en cas d'erreur temporaire

DNS ..... Domain Name System

FAI ..... Fournisseur d'Accès à Internet

gTLD ..... generic Top Level Domain

IDN ..... Internationalized Domain Name

MTA ..... Mail Transfer Agent

MX ..... Mail eXchange

Serveur Catch-All ..... Serveur de mail qui accepte toutes les adresses, sans retourner d'erreur pour les adresses inexistantes

SMTP ..... Simple Mail Transfer Protocol. Protocole utilisé pour envoyer un e-mail

TLD ..... Top Level Domain. Partie qui se trouve à l'extrême droite d'un nom de domaine ou d'une adresse e-mail. Par exemple, .be, .com, .org,…

URL ..... Uniforme Resource Locator. Désigne l'adresse d'une page web