

# Comment gérer le risque de lock-in technique en cas d'usage de services de cloud computing ?

Robert VISEUR <sup>(1)(2)</sup>  
Etienne CHARLIER <sup>(1)</sup>  
Michael VAN DE BORNE <sup>(1)</sup>

<sup>(1)</sup> CETIC, Rue des Frères Wright, 29/3, 6041 Charleroi, Belgique  
{robert.viseur, etienne.charlier, michael.vandeborne} @cetic.be

<sup>(2)</sup> UMONS Faculté Polytechnique, Rue de Houdain, 9, 7000 Mons, Belgique  
{robert.viseur} @umons.ac.be

**Résumé** – Depuis une dizaine d'années, les technologies du cloud computing pénètrent le monde de l'entreprise. En 2012, en France, ce marché représentait 2 milliards d'euros de chiffres d'affaires. Conséquence du cloud computing, une part croissante des données des entreprises (mais aussi du grand public) se trouve hébergée en ligne. Il en résulte un risque de lock-in accru, source d'inquiétude pour des utilisateurs confrontés au risque de voir leurs données en ligne sans possibilité de les migrer sur leurs propres ressources informatiques ou sur des plates-formes concurrentes. Nous traitons, dans cette recherche préliminaire, la problématique de la gestion du lock-in en cas d'usage de services de cloud computing. Nous cherchons à répondre à six questions: (1) Qu'est-ce que le lock-in ? (2) Le lock-in est-il perçu comme un problème important ? (3) Quelles sont les causes de lock-in ? (4) Quel est l'impact du lock-in pour l'utilisateur ? (5) Comment l'utilisateur peut-il éviter le lock-in ? (6) La problématique du lock-in concerne-t-elle le grand public ? Notre article est organisé en trois sections. La première section présente la méthodologie suivie pour notre étude. Notre recherche consiste en une revue de la littérature. Cette dernière s'appuie sur deux types de sources: d'une part, des articles issus de la littérature scientifique dédiée à la question du lock-in et, d'autre part, un ensemble d'articles issus de la presse professionnelle traitant spécifiquement de la question du lock-in dans le cloud. La seconde section développe les résultats. Cette section identifie quatre causes différentes de lock-in ainsi que quatre impacts sur les utilisateurs, et propose six mécanismes permettant de réduire le risque de lock-in. La troisième section discute les résultats obtenus, et propose des travaux complémentaires.

**Abstract** – Cloud computing technologies enter the world of business in the past ten years. In 2012, in France, this market accounted for 2 billion euros of revenues. Consequence of cloud computing, an increasing part of corporate data (but also private data) is hosted online. This results in an increased risk of lock-in that is source of anxiety for users facing the risk of having their data to be hosted online without the possibility to migrate them on their own IT resources or on competitor's platforms. In this preliminary research, we deal with the problem of the management of lock-in in case of use of cloud computing services. We aim to answer six questions: (1) What is the lock-in ? (2) Is the lock-in perceived as a major problem ? (3) What are the causes of lock-in ? (4) What is the impact of lock-in on the users ? (5) How can the users avoid lock-in ? (6) Does the problem of lock-in affect the general public ? Our paper is organized in three sections. The first section presents the methodology used for this study. Our research consists of a literature review. The latter is based on two types of sources: first, articles from scientific literature dedicated to the issue of lock-in and, secondly, a set of articles from the professional press dealing specifically the issue of lock-in in the cloud. The second section details the results. This section identifies four possible causes of lock-in and four impacts on users, as well as six mechanisms to reduce the risk of lock-in. The third section discusses the results and suggests further work.

**Mots-clés** – cloud computing ; lock-in ; iaas ; paas ; saas ; big data ; standardisation ; open source.

**Keywords** – cloud computing ; lock-in. ; iaas ; paas ; saas ; big data ; standardization ; open source.

## 1. Contexte

L'histoire du cloud computing est récente (Viseur, 2013b). Elle tire en effet ses racines dans les offres d'hébergement d'applications (modèle ASP) qui sont apparues au début des années 2000. Ce phénomène est cependant amené à bouleverser profondément la manière dont les services IT "sont inventés, déployés, mis à l'échelle, mis à jour, maintenus et rémunérés" (Marston et al., 2011). Le cloud computing peut être défini

comme "un modèle de services IT dans lequel les services informatiques (à la fois matériel et logiciel) sont fournis à la demande aux clients, au travers d'un réseau, indépendamment du terminal et du lieu" (Marston et al., 2011).

Du point de vue de l'utilisateur, le cloud computing se décline en trois modèles distincts (CIGREF, 2013 ; Marston et al., 2011). Le modèle Software as a Service (SaaS) fournit à l'utilisateur une application, hébergée dans le cloud (exemples : Google Mail, Google Documents). Le modèle Platform as a Service (PaaS) fournit un environnement de développement et de déploiement d'applications (exemples : Microsoft Azure, Google App Engine). Le modèle d'Infrastructure as a Service (IaaS) fournit du stockage et des capacités de calcul (exemples : Amazon S3, Amazon EC2). Ces modèles peuvent être déployés dans le réseau de l'entreprise ou sur une plate-forme externe. Dans le premier cas, on parle de cloud privé; dans le second, de cloud public. Ce dernier bénéficie depuis plusieurs années d'un marketing important auprès des entreprises et de la notoriété croissante de fournisseurs tels qu'Amazon (EC2) ou Microsoft (Azure).

Selon le cabinet IDC ([idc.fr](http://idc.fr)), le cloud computing représentait ainsi en France un budget de près de 2 milliards d'euros en 2012, et 12% du total des dépenses informatiques des entreprises (hors tablettes et smartphones). Selon le cabinet PAC ([www.pac-online.com](http://www.pac-online.com)), le marché du SaaS serait le plus développé et représenterait 62% des utilisations du cloud computing en France. Le cloud computing touche aussi le grand public. Ce dernier y est en effet confronté au travers de services en ligne de type SaaS tels que, par exemple, les services de messagerie offerts par Facebook, le service de courrier électronique de Google Mail ou les outils de bureautique en ligne de Google Documents.

Là où l'utilisateur disposait de ses applications et de ses données sur son propre ordinateur (ou sur un réseau d'ordinateurs dont il a la maîtrise), le cloud computing externalise l'infrastructure, les applications et/ou les données. Il en résulte un lock-in (i.e. dépendance) accru (ou perçu comme tel), source d'inquiétude pour des utilisateurs confrontés au risque de voir leur données en ligne sans possibilité de les migrer sur leurs propres ressources informatiques ou sur des plates-formes concurrentes.

Nous traitons, dans cette recherche préliminaire, la problématique de la gestion du lock-in en cas d'usage de services de cloud computing (clouds publics, de types SaaS, PaaS ou IaaS). La question de la collecte des données est abordée dans la recherche. Nous nous limitons au cas où ces données sont encodées volontairement par l'utilisateur, avec éventuellement un incitant de la part de l'entreprise. Nous n'abordons pas la question de la collecte automatique d'informations relatives à l'utilisateur (e.g. géolocalisation).

Nous chercherons à apporter une première réponse aux six questions suivantes :

- (1) Qu'est-ce que le lock-in ?
- (2) Le lock-in est-il perçu comme un problème important ?
- (3) Quelles sont les causes de lock-in ?
- (4) Quel est l'impact du lock-in pour l'utilisateur ?
- (5) Comment l'utilisateur peut-il éviter le lock-in ?
- (6) La problématique du lock-in concerne-t-elle le grand public ?

Notre article est organisé en trois sections. La première section présente la méthodologie suivie pour notre étude. La seconde section développe les résultats correspondant aux six questions de recherche. La troisième section discute les résultats obtenus, et propose des travaux complémentaires.

## **2. Méthodologie**

En pratique, notre recherche consiste en une revue de la littérature. Cette dernière s'appuie sur deux types de sources.

Nous exploitons, d'une part, des articles issus de la littérature scientifique et traitant substantiellement de la question du lock-in. Ces articles ont été trouvés essentiellement par l'interrogation du moteur de recherche Google Scholar (scholar.google.fr).

Nous nous appuyons, d'autre part, sur un ensemble d'articles issus de la presse professionnelle et traitant spécifiquement de la question du lock-in dans le cloud. Le moteur de recherche Google a été utilisé pour identifier ces articles. En pratique, les quatre requêtes suivantes ont été utilisées:

- cloud lock-in,
- (saas OR paas OR iaas) lock-in,
- (amazon OR rackspace OR azure OR "app engine" OR smartcloud OR salesforce) lock-in,
- (facebook OR gmail) lock-in.

Les liens issus de la première page de résultats de recherche ont été filtrés pour ne garder que les articles de fond (pour la troisième requête, le nombre de liens hors sujets a nécessité de remonter jusque la troisième page de résultats). Au final, vingt-quatre articles ont été identifiés par cette méthode. Ils ont été complétés par une série de cinq articles pointés dans la première série d'articles. Les articles cités et issus de la littérature professionnelle peuvent être identifiés dans la bibliographie par la présence de l'URL. A noter que la notion de lock-in semble peu utilisée pour les services grand public, peut-être du fait que ce terme est davantage familier aux utilisateurs professionnels.

Le nom des prestataires utilisés dans les requêtes a été déterminé sur base de l'importance accordée dans la littérature professionnelle ou scientifique (Crochet-Damais, 2013; Darrow, 2012; Harsh *et al.*, 2012; Nachmani, 2012; ZDNet, 2013; Zhang *et al.*, 2013). La publication des parts de marchés permet, notamment, d'objectiver ce choix. Nous avons donc choisi de nous focaliser sur Amazon Web Services, Microsoft Azure, Google App Engine, Rackspace, IBM Smartcloud et Salesforce pour les professionnels. Facebook et GMail ont été pris en compte pour les services davantage orientés vers le grand public.

Chaque article, issu de la littérature professionnelle ou scientifique, a été traité de manière à identifier les éléments répondant à nos six questions de recherche. Ces éléments ont ensuite été catégorisés.

### **3. Résultats de la recherche**

#### **3.1. *Qu'est-ce que le lock-in ?***

Pour Germain (2013), le lock-in existe lorsque le coût du changement de la plate-forme technologique d'un vendeur vers une autre est à ce point onéreux que le client est incapable de quitter les offres du vendeur.

Le lock-in n'est pas un concept neuf (Linthicum, 2012). Juengst (2012) donne d'ailleurs des exemples bien connus : les GSM bloqués, les cartouches d'encre, ... Linthicum (2012) ajoute que le principe de l'utilisation de ressources externes à l'entreprise n'est elle-même pas neuve. Pour Germain (2013), également, le vendor lock-in fait partie de la vie de l'IT en entreprise depuis bien des années. En pratique, parler de lock-in revient à parler de coûts de migration.

Chatzakis (2012) étudie la question du lock-in dans AWS, et propose trois niveaux de lock-in. Le lock-in faible ("*light lock-in*") correspond en pratique à une absence de lock-in, par exemple lorsque la plate-forme utilise des standards industriels. Le lock-in moyen ("*medium lock-in*") se produit lorsque la plate-forme met à disposition des services non standards dont le caractère bloquant peut être limité par des règles de développement et d'architecture. Le lock-in fort ("*hard lock-in*") impose la réécriture de parties de code mais est associé à la mise à disposition d'innovations sources d'opportunités pour l'utilisateur.

Pour Malhotra (2013), "lock-in" est un autre terme pour dire "risque". Il y a un compromis à réaliser entre le lock-in (risque) et la valeur (profit). Moins de lock-in signifie moins de fonctionnalités et plus d'écriture de code en interne.

Zhang *et al.* (2013) associe le concept de lock-in à trois autres concepts: l'interopérabilité, la compatibilité et la portabilité. L'interopérabilité dans le cloud désigne la capacité de plusieurs fournisseurs à travailler ensemble. La compatibilité dans le cloud signifie que les données et les applications peuvent fonctionner de la même manière quel que soit le fournisseur de cloud. La portabilité dans le cloud signifie que les données et les applications peuvent être facilement déplacées et réutilisées quelque soit le choix de fournisseur de cloud, de système d'exploitation, de format de stockage ou d'API. En pratique, l'amélioration de l'interopérabilité, de la compatibilité et de la portabilité contribue à limiter le lock-in.

A noter que le terme "lock-in" se retrouve également dans la littérature sur les rendements croissants d'adoption (RCA). Traduit par "inflexibilité", le lock-in y désigne un mécanisme de blocage du processus d'adoption sur une technologie en concurrence avec d'autres technologies (Foray, 1989). Le tendance à la domination de cette technologie ne peut dès lors plus être remise en cause. L'utilisateur individuel est obligé de choisir cette technologie. Cette définition s'inscrit dans un champs théorique particulier et s'éloigne du sujet de notre étude. Foray (1989) nous apprend par contre qu'une technologie ancienne produit ses propres normes d'évaluation et y voit une nouvelle source de RCA. L'attention forte accordée au lock-in, dans le contexte actuel de basculement des systèmes informatiques depuis des solutions d'hébergement internes vers des plates-formes publiques flexibles (SaaS, PaaS ou IaaS), ne s'inscrit-elle pas dans cette logique ?

### **3.2. Le lock-in est-il perçu comme un problème important ?**

L'importance perçue du lock-in est largement discutée.

Selon Gruman (2007), la question du lock-in fait partie des préoccupations des utilisateurs de solutions de gestion d'entreprise, d'autant que le secteur a connu un mouvement de consolidation important à partir de 2002. Cette préoccupation est renforcée par l'importance des coûts de sortie du système, qui sont particulièrement élevés pour les applications de gestion (Messerschmitt et Szyperski, 2001). Pour Nachmani (2012), le lock-in a un impact important sur la décision d'utiliser ou non le cloud.

En pratique, cependant, les résultats sont plus contrastés. Smets, un entrepreneur français actif dans l'édition de logiciels libres et leur mise à disposition en tant que service, estime ainsi que "*la protection de la vie privée, le secret commercial et la liberté de migrer sont des différenciateurs sans importance pour la grande majorité du marché*" (Viseur, 2013b). Kash (2013) relève que le lock-in par le vendeur ne préoccupe que 15% des clients. Ce critère apparaît en 8ème position, un résultat à comparer aux vulnérabilités en matière de sécurité qui apparaissent en première position et qui préoccupent 51% des clients. Les 3 premiers critères cités concernent par ailleurs la sécurité.

Nachmani (2012) estime que le risque de lock-in en PaaS est plus important. Un faible lock-in y apporte un bénéfice important. L'importance accordée au lock-in dépendrait dès lors du risque perçu, fonction du modèle de mise à disposition ou du domaine applicatif.

### **3.3. Quelles sont les causes du lock-in ?**

Quatre causes ont été identifiées à la mise en place du lock-in:

1. le rythme d'innovation et la recherche de différenciation vis-à-vis des concurrents,
2. la recherche de rendements croissants d'adoption (RCA),
3. l'utilisation par les prestataires de formats de données propriétaires,
4. les plates-formes cloud de type PaaS.

### 3.3.1. *Le rythme d'innovation et la recherche de différenciation vis-à-vis des concurrents*

Les prestataires de services de cloud computing cherchent à se différencier par la mise sur le marché d'innovations. Ces innovations prennent la forme de fonctionnalités avancées permettant la personnalisation des services mis à disposition par les fournisseurs de services de cloud computing (Kash, 2013). Ces fonctionnalités accroissent le risque de lock-in. Elles prennent notamment la forme d'API propres aux fournisseurs (Germain, 2013; Juengst, 2012; Kash, 2013). Ces API permettent l'automatisation ou la personnalisation de la gestion de la plate-forme, ou donnent accès à des services innovants, par exemple dans le domaine du stockage (Harris, 2013; Kash, 2013). Les API propres aux vendeurs sont actuellement utilisées par 23% des clients, et ce chiffre devrait encore augmenter (Kash, 2013).

### 3.3.2. *La recherche de rendements croissants d'adoption (RCA)*

Babcock (2013) estime que le lock-in se produit quand une société devient un vendeur dominant derrière une technologie et développe des produits qui prennent l'avance avec des éléments propriétaires. Cette stratégie empêche les clients de partir. Les fournisseurs gardent leurs technologies propriétaires aussi longtemps que possible, car cela bloque le client dans leur environnement (McKendrick, 2011). Le lock-in imposé aux clients est une bonne chose pour le vendeur car il réduit la rotation des clients ("*churn*") (Harsh *et al.*, 2012). Les auteurs contestent cependant cette vue, car ils estiment que la fidélité à une marque doit s'obtenir par la qualité du service et l'attractivité des prix. Pour Zhang *et al.* (2013), également, l'incompatibilité entre les fournisseurs de produits et services de cloud computing peut temporairement protéger les intérêts de chaque fournisseur. Cependant, cette stratégie se révélera contreproductive à long terme au fur et à mesure que le marché deviendra plus mature. De plus, cette stratégie va par ailleurs à l'encontre des nouveaux modes de définition coopérative de standards ouverts (Adatto, 2013).

Cette stratégie assure par ailleurs que le vendeur pourra capitaliser seul sur cette technologie. Ce dernier cherche alors à créer un standard de facto par la recherche de rendements croissants d'adoption. Il privilégie le contrôle de sa technologie à une plus large diffusion.

### 3.3.3. *L'utilisation par les prestataires de formats de données propriétaires*

Les données peuvent être bloquées par l'utilisation d'un format propriétaire (Chow *et al.*, 2009). En matière de SaaS, le problème d'interopérabilité se pose surtout en matière de données (Zhang *et al.*, 2013).

Indépendamment d'une éventuelle volonté de freiner les sorties de données, le volume de données peut en lui-même handicaper les migrations et bloquer l'utilisateur chez un prestataire. Les applications externalisées de Big Data posent ainsi des problèmes spécifiques en cas de migration, compte tenu du volume de données. Les contrats ne spécifient pas toujours les modalités de migration des données des utilisateurs dès lors que ces derniers souhaitent changer de fournisseur (Kash, 2013).

### 3.3.4. *Les plates-formes cloud de type PaaS*

Les services de PaaS sont fréquemment pointés du doigt comme présentant un risque important de lock-in (Harris, 2013; Germain, 2013; Juengst, 2012). Zhang *et al.* (2013) justifient cela par le fait que, du IaaS au SaaS, l'automatisation augmente. Dès lors, du IaaS au SaaS, la portabilité diminue. Plus de portabilité signifie plus de travail en matière de gestion et de déploiement de logiciels. Différents éléments expliquent l'importance du lock-in pour les PaaS (Coté, 2008; Germain, 2013; Juengst, 2012): l'utilisation d'un langage de programmation propriétaire, le support de langages open source étendus à l'aide d'API propriétaires, la mise à disposition de services d'infrastructure propriétaires, l'exploitation d'une base de données propriétaires, etc.

Cependant, même pour le PaaS, le degré de lock-in est variable (Nachmani, 2012). Par exemple, Force.com, le PaaS de Salesforce.com, présente un degré de lock-in maximal, car le langage Apex et la base de données sont propriétaires. A l'inverse, Heroku, racheté par Salesforce.com, supporte JSON et XML pour les services Web, des langages standards ouverts comme Java, Ruby ou PHP, et les bases de données open source PostgreSQL et MySQL. Pour son Google App Engine, Google met fréquemment en avant les implémentations open source existantes comme App Scale ou CapeDrawf (Harris, 2013; Magnusson, 2013). Red Hat promeut sa plate-forme OpenShift en insistant sur le support d'une implémentation open source facilitant la mise en place d'un PaaS privé.

### **3.4. Quel est l'impact du lock-in pour l'utilisateur ?**

Quatre impacts pour l'utilisateur ont été identifiés suite à un lock-in:

1. le blocage des données des utilisateurs et l'allongement des durées de migration,
2. l'augmentation du tarif associé à l'utilisation du service de cloud computing,
3. le ralentissement du rythme d'innovation,
4. la réduction du cycle de vie des développements.

#### *3.4.1. Le blocage des données des utilisateurs et l'allongement des durées de migration*

Les données des utilisateurs peuvent être difficilement exploitables, du fait de l'impossibilité de moyens techniques permettant d'y accéder, de l'utilisation de formats propriétaires, ou du volume de données à rapatrier (Chatzakis, 2012).

L'allongement des délais de migration qui en découle n'est pas sans conséquence potentielle pour l'entreprise. Le fournisseur n'est ainsi pas à l'abri d'un rachat (McKendrick, 2011). Le nouveau propriétaire peut changer ses politiques, et entraîner des problèmes, notamment légaux (e.g. localisation des données). Les clients n'ont par ailleurs pas de contrôle sur l'évolution d'un cloud commercial (Chow *et al.*, 2009). Ils peuvent dès lors se retrouver dans l'embarras suite à la fermeture d'un service. Les utilisateurs de Coghead ont été confrontés à cette situation suite à la fermeture de l'entreprise et au rachat d'actifs par SAP.

La question du lock-in peut être liée à la vie privée ("*privacy*"), au travers de la question du cycle de vie des données, que ce soit pour les utilisateurs professionnels ou le grand public (Pearson, 2009). La dernière phase du cycle de vie est le déclassement ("*decommission*"), qui prévoit la suppression sécurisée et l'élimination des données personnelles et sensibles. Ce point renvoie directement à la question de la propriété et de l'exploitation des données sur les réseaux sociaux.

Les conséquences du lock-in peuvent être masquées par les pratiques tarifaires des entreprises. Ainsi, les clients peuvent être attirés par la guerre des prix portant sur le téléversement ("*upload*") de données (Darrow, 2013). Il est cependant possible qu'une fois ces données en ligne les prestataires essaient de se rémunérer autrement, et, dès lors, handicaper les sorties de données.

#### *3.4.2. L'augmentation du tarif associé à l'utilisation du service de cloud computing*

Zhang *et al.* (2013) estiment qu'il existe un risque de charges ridiculement élevées dues au lock-in. L'entreprise peut augmenter ses prix. L'utilisateur peut rencontrer des difficultés à migrer vers de nouvelles plates-formes plus attractives (Coté, 2008). Un vendeur de services PaaS peut augmenter ses prix une fois ses clients bloqués (Juengst, 2012). Cette dernière affirmation ne vaut cependant que pour les PaaS ne reposant pas sur une implémentation open source publiée et complète.

### 3.4.3. *Le ralentissement du rythme d'innovation*

Pour Germain (2013) le lock-in crée des monopoles pour le vendeur au détriment de certains clients et limite la pression sur le fournisseur pour innover.

### 3.4.4. *La réduction du cycle de vie des développements*

Contre toute attente, certains auteurs présentent le lock-in comme un élément positif pour les utilisateurs. Cette appréciation est associée à la vision du lock-in comme conséquence d'un processus d'innovation qui profite à l'utilisateur du service. Elle provient généralement de personnes proches des prestataires cloud (Coffee travaillait pour Salesforce.com ; Magnusson, pour Google).

Coffee (2012) préfère ainsi parler d'effet de levier plutôt que de lock-in. Selon lui, les applications développées sur Force.com ont ainsi un cycle de vie plus court. Dès lors, elles peuvent être plus rapidement adaptées aux réalités du marché. Il en résulte un avantage compétitif important pour l'utilisateur du service cloud, capable de s'adapter à des marchés évoluant rapidement, comme celui des applications mobiles ("*time-to-market advantage*").

Magnusson (2013) appuie ce point de vue: le lock-in est le prix à payer pour une plate-forme innovante. La plate-forme réalise plus de travail à la place de l'utilisateur et permet de gagner du temps (e.g. Google Datastore). La sortie du Google App Engine (GAE) demande 3 à 4 mois de travail pour une application importante (Magnusson, 2013). Cependant, les bénéfices du GAE sont nombreux: prise en charge du firewall, protection contre les DDoS, contre les virus, application des patches et des mises à jour, équilibrage de charge,...

## 3.5. *Comment l'utilisateur peut-il éviter le lock-in ?*

Six mécanismes ont été identifiés pour éviter le lock-in:

1. le recours aux standards,
2. l'utilisation de logiciels FLOSS (Free Libre Open Source Software),
3. le développement d'applications reposant sur un socle fonctionnel générique,
4. le recours à des opérateurs techniques spécialisés,
5. la confiance envers les labels "open cloud",
6. la mise en place d'une stratégie de sortie.

### 3.5.1. *Le recours aux standards*

La standardisation permet d'améliorer l'interopérabilité entre clouds et de limiter le lock-in (Chow *et al.*, 2009; Messerschmitt et Szyperski, 2001; Zhang *et al.*, 2013). Les standards sont nombreux. Pahl *et al.* (2013) listent ainsi un ensemble de standards, dans les domaines de la modélisation de service (e.g. OpenSCA, USDL/SoaML/CloudML, EMMML), des interfaces de service (e.g. OCCI, CMMI, EC2, TOSCA, CDMI) et de l'infrastructure (OVF). Le standard CDMI (Cloud Data Management Interface) est dédié aux données, plus spécifiquement à la gestion du stockage dans le cloud.

Wolpe (2013) estime cependant que l'industrie du cloud souffre encore d'un manque de standardisation. Selon cet auteur, la standardisation n'arriverait qu'après une première phase d'innovation, car la standardisation apparaît comme un frein au progrès (*sic*). Cette vision de la standardisation contredit cependant Adatto (2013). Ce dernier analyse l'émergence de nouveaux modes de définition coopérative des standards, accompagnés d'implémentations FLOSS, et le développement de stratégies de coopération entre acteurs industriels. Le développement d'applications FLOSS, parallèlement au travail de spécification, amène dès lors la standardisation au coeur du processus d'innovation. Kash (2013) estime d'ailleurs que l'utilisation

de technologies open source permet de bénéficier du rythme d'innovation rapide de ce type d'outil.

D'autres auteurs soulignent les lacunes actuelles en matière de standardisation et d'interopérabilité dans le cloud. Kash (2013) regrette ainsi le manque de maturité des standards. S'ils se consolident en matière de services IaaS, ils seraient, par contre, pratiquement inexistantes pour les services PaaS. Cependant, même en IaaS, certaines pièces technologiques manquent en matière d'interopérabilité (Harsh *et al.*, 2012). Dans le IaaS, il existe un ensemble d'enjeux techniques à résoudre (Zhang *et al.*, 2013). Ils sont partiellement couverts par des standards (e.g.: OVF, CDMI et OCCI) proposés par des organisations comme l'Open Grid Forum (OGF), la Distributed Management Task Force (DTMF) ou la Storage Networking Institute Association (SNIA). Ces standards sont progressivement mis en oeuvre dans des solutions open source telles qu'OpenStack, OpenNebula et Eucalyptus. L'avancement de la standardisation peut cependant se révéler variable. Par exemple, la virtualisation du réseau ou des procédures de sécurité sont des points importants, actuellement peu traités, au contraire de la question des formats de virtualisation, bien couverte par OVF (Harsh *et al.*, 2012; Zhang *et al.*, 2013). En matière de IaaS, il existe une taxonomie de l'interopérabilité dans le IaaS, permettant de pointer plus rapidement les problèmes techniques à résoudre en matière de mécanisme d'accès, de virtualisation, de stockage, de réseau, de sécurité ou encore de qualité de service (SLA) (Zhang *et al.*, 2013).

### 3.5.2. L'utilisation de logiciels FLOSS (Free Libre Open Source Software)

Pour Weinberger (2012), une solution au problème de lock-in réside dans l'utilisation de solutions FLOSS telles que Apache CloudStack, OpenStack et Eucalyptus.

Des logiciels open source accompagnent de plus en plus souvent des initiatives de standardisation (Adatto, 2013). Ils apparaissent dès lors souvent à la pointe en matière de standardisation. OpenStack se distingue par exemple par ses capacités de description de réseau par voie logicielle (Zhang *et al.*, 2013). Des inquiétudes peuvent cependant apparaître quant aux différences d'implémentations pouvant exister entre fournisseurs de cloud computing. Harris (2013) cite l'exemple d'OpenStack. Cependant, l'avis, en provenance d'un analyste Gartner, semble plutôt lié à une mauvaise compréhension du fonctionnement des écosystèmes open source et du modèle de développement collaboratif mis en oeuvre dans les communautés FLOSS.

Les implémentations open source peuvent également apporter une réponse aux craintes de lock-in sur les services de type PaaS. Juengst (2012) recommande l'utilisation de PaaS proposant le support de plusieurs langages de programmation, construits sur des briques open source, open source eux-mêmes et ne proposant pas d'APIs propriétaires. Les cas d'OpenShift ([www.openshift.com](http://www.openshift.com)) et de Google App Engine ([cloud.google.com/AppEngine](http://cloud.google.com/AppEngine)) sont illustratifs.

Google App Engine possède deux implémentations open source (Magnusson, 2013). La première est App Scale ([www.appscale.com](http://www.appscale.com)). Google collabore avec Red Hat pour la seconde, intégrée au logiciel OpenShift, via CapeDwarf ([www.jboss.org/apedwarf](http://www.jboss.org/apedwarf)). Google collabore également à la création d'un Technology Compatibility Kit (TCK) for Google App Engine (GAE) API ([www.appengine-tck.org](http://www.appengine-tck.org)) permettant aux éditeurs d'implémentations alternatives de réaliser des tests de compatibilité. Red Hat ([redhat.com](http://redhat.com)) supporte la fonctionnalité du service de stockage Datastore via le logiciel open source Infinispan ([infinispan.org](http://infinispan.org)).

Red Hat, avec OpenShift, utilise en pratique l'absence de lock-in comme argument commercial (Juengst, 2012). La promesse est d'offrir une plate-forme PaaS publique (OpenShift Online), que les entreprises peuvent mettre en oeuvre dans leur réseau ou sur un hébergeur IaaS public de leur choix (via OpenShift Enterprise) en s'appuyant sur une implémentation FLOSS de la technologie OpenShift (OpenShift Origin).

En pratique, Google Trends ([www.google.com/trends/](http://www.google.com/trends/)) montre un franc décollage d'OpenShift.



### 3.5.3. Le développement d'applications reposant sur un socle fonctionnel générique

Les développements sur plates-formes cloud peuvent être abordés en imposant l'utilisation d'intergiciels (frameworks, middlewares) permettant de s'abstraire des différences entre plates-formes de cloud de type IaaS ou PaaS (Kash, 2013; Zhang *et al.*, 2013). En matière de IaaS, Zhang *et al.* (2013) citent l'existence de bibliothèques facilitant l'interopérabilité, comme LibCloud ([libcloud.apache.org](http://libcloud.apache.org)) ou Deltacloud ([deltacloud.apache.org](http://deltacloud.apache.org)). Ces bibliothèques permettent uniquement d'utiliser les fonctionnalités communes aux différentes plates-formes supportées. En matière de PaaS, nous pouvons citer l'API Simple Cloud, intégrée par Zend ([www.zend.com](http://www.zend.com)) à son framework PHP ([framework.zend.com](http://framework.zend.com)), supportant notamment les services de stockage dans le cloud d'Amazon et de Microsoft.

Le lock-in peut aussi être limité en développant les principaux algorithmes dans un langage de développement largement supporté tel que Java, puis en exploitant pour le reste les facilités offertes par les prestataires de cloud computing (Coffee, 2012).

### 3.5.4. Le recours à des opérateurs techniques spécialisés

Des prestataires techniques, qualifiés de "*technical cloud brokers*", peuvent aider les agences à éviter le lock-in et à exploiter en parallèle plusieurs services de cloud computing (Kash, 2013). Ils sont à comparer aux "*enablers*" dans Marston *et al.* (2011)

### 3.5.5. La confiance envers les labels "open cloud"

La Foundation for a Free Information Infrastructure ([www.ffi.org](http://www.ffi.org)) a proposé une définition de l'open cloud. Cette définition propose trois niveaux de liberté : TIO (Total Information Outsourcing) Libre / Open / Loyal (Scoffoni *et al.*, 2012). Le niveau TIO Loyal "*suppose de fournir un cadre ayant le même niveau de secret et de transparence qu'avec son propre personnel*" (Scoffoni *et al.*, 2012). Le niveau TIO Open prévoit la liberté des données, et que ces dernières soient structurées selon un format clairement spécifié. Le niveau TIO Libre prévoit la liberté des données, la liberté des logiciels et la liberté de concurrence. Le label TIO bénéficie cependant à ce stade d'une promotion limitée (Viseur, 2013b). De plus, les modalités pratiques de mise en oeuvre et la garantie de respect des conditions restent difficiles à identifier. D'autres propositions semblables existent, tels que les Open Cloud Principles de l'Open Cloud Initiative ([www.opencloudinitiative.org](http://www.opencloudinitiative.org)) ou encore l'Open Cloud Manifesto ([www.opencloudmanifesto.org](http://www.opencloudmanifesto.org)) (Jean, 2013).

### 3.5.6. La mise en place d'une stratégie de sortie

L'entreprise doit élaborer une stratégie de sortie ("*exit strategy*") (McKendrick, 2011). Le coût de cette dernière doit intégrer le calcul de coût pour la mise en place de la solution. La migration de données doit cependant faire l'objet de tests, pas uniquement de discussions avec les vendeurs (Kash, 2013).

Le support de standards ouverts ou de leurs implémentations FLOSS facilitent la mise en place d'une stratégie de sortie. Cette dernière peut, en outre, s'appuyer sur l'existence d'outils de migration. Généralement, ces outils supportent les solutions de virtualisation et de cloud computing les plus populaires telles que VMWare, Amazon ou OpenStack (Kash, 2013). Sur Amazon, par exemple, l'importance du lock-in dépend de l'existence de logiciels de conversion (e.g. traductions de VM pour EC2), du support d'API standards (e.g. RDS compatible Oracle ou MySQL), d'implémentations alternatives (e.g. Elasticsearch compatible Memcache) ou de l'existence de frameworks apportant une couche d'abstraction (e.g. Zend Framework ou Django pour S3) (Chatzakis, 2012). Même si des solutions de migration existent, les services de bases de données NoSQL DynamoDB et SimpleDB sont une source de lock-in fort.

A noter que l'élaboration d'une stratégie de sortie réaliste suppose que l'entreprise n'a pas été affectée par la perte de compétences qui peut découler d'initiatives d'externalisation (Quélin, 2003). Cette perte peut

entraîner un handicap pour un retour en arrière ou même pour l'évaluation de solutions alternatives.

### 3.6. La problématique du lock-in concerne-t-elle le grand public ?

La portabilité des données encodées par les utilisateurs de Facebook apparaît comme un problème connu de longue date. Facebook entrave la capacité pour ses utilisateurs d'exporter la liste des amis vers son concurrent Google Plus (Asay, 2011). Il reste possible d'exporter ses données personnelles dans une archive téléchargeable (Protalinski, 2011). Cette dernière est surtout utilisable comme sauvegarde personnelle. Facebook a par la suite amélioré son système d'exportation pour les développeurs, en enrichissant les données avec des microformats (hAtom, hMedia et hCard) (Protalinski, 2011). L'export des contacts fait cependant l'objet de nombreux articles sur le Web, preuve s'il en est que l'interopérabilité entre réseaux sociaux reste limitée.

Weinberger (2012) montre que la difficulté de sortie des données se pose aussi pour d'autres services grand public. Flickr, par exemple, ne dispose en effet pas d'une méthode de migration officiellement supportée. Flickr s'est d'ailleurs illustré en 2006 en bloquant la migration des photos déposées par ses utilisateurs vers le service concurrent Zoomr. Son API, bien fonctionnelle, était rendue inopérante pour des raisons contractuelles: "*We choose not to support use of the API for sites that are a straight alternative to Flickr*" (Ozerman, 2006).

Cependant, la pression des consommateurs aboutit à l'apparition d'initiatives facilitant la migration des données, comme Google Takeout Initiative (Weinberger, 2012). Cette dernière, intégrée au Google Data Liberation Front ([www.dataliberation.org](http://www.dataliberation.org)), permet l'exportation des données gérées dans les services Google vers des formats standards (standards de facto, comme DOCX, ou standards ouverts, comme ODF). Par exemple, une boîte à courriels pourra être exportée en MBOX, un format supporté par Mozilla Thunderbird ou Microsoft Outlook.

## 4. Discussion et perspectives

Cette recherche préliminaire nous a permis d'attribuer le lock-in à quatre causes différentes, d'identifier quatre impacts du lock-in sur les utilisateurs et de proposer six mécanismes permettant de réduire le risque de lock-in (voir Table 1).

Causes du lock-in	Impacts du lock-in	Mécanismes pour éviter le lock-in
1. le rythme d'innovation et la recherche de différenciation vis-à-vis des concurrents, 2. la recherche de rendements croissants d'adoption (RCA), 3. l'utilisation par les prestataires de formats de données propriétaires, 4. les plates-formes cloud de type PaaS.	1 le blocage des données des utilisateurs et l'allongement des durées de migration, 2. l'augmentation du tarif associé à l'utilisation du service de cloud computing, 3. le ralentissement du rythme d'innovation, 4. la réduction du cycle de vie des développements.	1. le recours aux standards, 2. l'utilisation de logiciels FLOSS (Free Libre Open Source Software), 3. le développement d'applications reposant sur un socle fonctionnel générique, 4. le recours à des opérateurs techniques spécialisés, 5. la confiance envers les labels "open cloud", 6. la mise en place d'une stratégie de sortie.

**Table 1.** Causes et impacts du lock-in, mécanismes pour éviter le lock-in.

Notre recherche s'est centrée sur quelques prestataires dominants sur le plan mondial. Elle mériterait dès lors une extension à des fournisseurs de niche (e.g. Ikoula), dont le positionnement face aux acteurs dominants pourrait se révéler riche en enseignements. Les articles retenus développent surtout la question du lock-in pour les clients professionnels. Un approfondissement des stratégies de rétention mises en oeuvre par les prestataires avec le grand public mériterait également un approfondissement.

La question du lock-in y apparaît aussi vieille que l'existence de l'informatique. La problématique des

formats de données dans le domaine de la bureautique en est une illustration bien connue (Adatto, 2013). La problématique présente cependant une dimension supplémentaire dans le cas du cloud computing. En effet, le lock-in dans le cloud entraîne non seulement des difficultés en terme d'évolutivité (par exemple, si le rythme d'innovation offert par le fournisseur diminue) mais présente également un risque accru en terme de continuité du service. En pratique, une solution informatique installée sur un réseau local par un éditeur en faillite peut être utilisée pendant encore quelques temps par l'entreprise. Un fournisseur de cloud computing en faillite ou s'appuyant lui-même sur un fournisseur d'infrastructure cloud en faillite causera des difficultés bien plus grandes à ses clients du fait de l'inaccessibilité du service. Ce point justifie à lui-seul l'actualité du sujet dans la presse informatique.

La présente recherche nous a permis d'identifier un vocabulaire associé à la question du (vendor) lock-in. Il s'agit en particulier des expressions suivantes: (1) portabilité ("*portability*"), compatibilité ("*compatibility*") et, surtout, interopérabilité ("*interoperability*"), (2) stratégie de sortie ("*exit strategy*"), (3) rendements croissants d'adoption (RCA) et (4) externalisation ("*outsourcing*"). Une recherche sur les études associées à ces différentes expressions pourrait apporter un éclairage supplémentaire sur la question du lock-in dans les services de cloud computing.

La standardisation, gage d'interopérabilité entre plates-formes de cloud computing, ressort comme une piste majeure pour éviter le lock-in. La littérature traite cependant la question des standards sous différents angles:

- les standards de facto, supportés par des bibliothèques et des logiciels, et facilitant les migrations,
- les standards ouverts, notamment implémentés dans les solutions FLOSS du marché,
- les logiciels standards, vocable recouvrant des projets FLOSS tels que Openstack,
- les labels "open cloud" (standards de comportement), assez rarement évoqués dans la littérature.

	OpenStack	Eucalyptus	OpenNebula	Cloudstack
<b>OVF</b>	oui	nc	oui	nc
<b>CDMI</b>	oui	nc	oui	nc
<b>OCCI</b>	oui	oui	oui	oui
<b>AMI</b>	non	oui	nc	nc
<b>S3 API</b>	oui	oui	non	oui
<b>EC2 API</b>	oui	oui	oui	oui

**Table 2.** Support des standards ouverts et de facto (Amazon) dans les IaaS open source.

Cependant, l'existence de standards ne résout pas tous les problèmes. Se posent, en effet, les questions de la couverture fonctionnelle des standards, du support industriel des standards, de la couverture des standards par les implémentations et du faible succès des labels "open cloud". Zhang *et al.* (2013) apportent une première réponse dans le cas du IaaS pour ce qui est de la couverture fonctionnelle des standards et de la couverture du standard par les implémentations. Un complément -basé sur Zhang *et al.* (2013), snia.org, dmtf.org, occi-wg.org, openstack.org, opennebula.org et cloudstack.apache.org- est proposé dans la Table 2. D'autres travaux sont nécessaires pour, par exemple, estimer le niveau réel de compatibilité des logiciels avec les standards listés. La granularité des spécifications mérite également un approfondissement, compte tenu de la subtilité de certaines causes de lock-in (e.g. possibilité ou non de migrer des adresses IP, de modifier le DNS,...). La question du support industriel des standards mériterait également un développement. Elle pourrait par exemple être estimée par le biais des volumétries des résultats de moteurs de recherche (webométrie) comme cela se fait déjà pour l'estimation de parts de marché (Viseur, 2012; Viseur, 2013a). La réflexion mériterait également d'être étendue au PaaS, pour lequel des initiatives de standardisation ont également vu le jour.

La particularisation de la question du lock-in en fonction du modèle de mise à disposition IaaS, PaaS ou SaaS nous semble à ce stade difficile. D'une part, nous avons vu que, même si la littérature tend à attribuer

un risque de dépendance plus important dans le cas du PaaS, les multiples initiatives de standardisation et de mises à disposition d'implémentations FLOSS (e.g. OpenShift) changeaient considérablement la donne. D'une part, il nous est apparu que la décomposition entre fournisseurs IaaS, PaaS et SaaS, si elle facilitait la compréhension et l'analyse, était parfois artificielle. En effet, une plate-forme étiquetée IaaS comme Amazon peut mettre à disposition des services de plus haut niveau associés classiquement à des plates-formes PaaS (e.g. base de données NoSQL). En outre, une plate-forme peut être clairement à cheval sur deux modes de mise à disposition en fonction du public considéré. Par exemple, Facebook sera considéré comme SaaS si l'on considère le particulier utilisant la fonction de messagerie instantanée ou PaaS si l'on considère le développeur contribuant au magasin d'application. Même chose avec Salesforce.com (SaaS) et son complément Force.com (PaaS). D'une manière générale, la caractérisation du lock-in dans les services de cloud computing pourrait par ailleurs bénéficier d'une segmentation systématique des utilisateurs ("*user lock-in*" fonction du profil : administrateur système, développeur d'application ou utilisateur applicatif).

Le problème du lock-in nous semble par ailleurs présenter de nombreuses similitudes pour les professionnels et les particuliers. Premièrement, il existe des stratégies de collecte à destination des deux publics, que ce soit par la proposition de services gratuits (payés par la vie privée<sup>1</sup>) ou par la mise en place de tarifs agressifs pour le téléversement. La conséquence peut être l'instauration de stratégies de rémunération contraires aux intérêts des utilisateurs (e.g.: paiement du service par la revente de données privées, prix plus élevés à la sortie des données,...). Deuxièmement, si la problématique du IaaS et du PaaS concerne peu les particuliers, ce dernier est confronté, comme les professionnels, au solutions de type SaaS, que ce soit avec les réseaux sociaux ou, surtout, avec des outils tels que des suites bureautiques en ligne ou les services de courrier électronique. Ces derniers sont couramment utilisés par les professionnels et appellent des stratégies de sortie communes (e.g. Data Liberation Front). Troisièmement, la question des données, y compris des données des utilisateurs, est difficilement dissociable de la question des logiciels associés. Par exemple, Facebook offre bien une fonctionnalité d'archivage de ses données. Par contre, l'exploitation de ces données dans un service ou produit alternatif reste problématique.

Nous terminerons en formulant nos recommandations pour éviter le lock-in dans les services de cloud computing. Nous distinguerons, d'une part, les systèmes IaaS/PaaS et, d'autre part, les systèmes SaaS. Nous proposons, pour les systèmes IaaS/PaaS, de suivre cinq étapes: (1) l'identification des sources de lock-in (e.g. solutions propriétaires, volume de données, topologie du réseau interne,...), (2) l'identification des services et produits compatibles, (3) l'évaluation du coût de sortie, (4) l'évaluation des gains en délai de commercialisation, et (5), le choix d'une plate-forme technique (privée / publique / hybride). Nous proposons, pour les systèmes SaaS, de suivre cinq étapes: (1) la vérification des modalités de sortie des données, (2) la vérification des formats des données (standards ou non), (3) l'identification des produits et services permettant de traiter ces données, (4) le test de la capacité réelle de reprise des données et (5) la sélection du service. Ces différentes étapes pourront être détaillées lors de recherches ultérieures.

## 5. Références

### 5.1. Références scientifiques

ADATTO, L. (2013), "Standards ouverts et implémentations FLOSS (Free Libre Open Source Software) : vers un nouveau modèle synergique de standardisation promu par l'industrie du logiciel", Terminal : Technologie de l'Information, Culture, Société, n°113-114, pp. 137-170.

CHOW, R., GOLLE, P., JAKOBSSON, M., et al. (2009), "Controlling data in the cloud: outsourcing computation without outsourcing control", Proceedings of the 2009 ACM workshop on Cloud computing

---

1 En pratique, les individus montrent de réelles dispositions à révéler des informations sur leur vie privée en échange de petites compensations (par exemple: communiquer sur un réseau social en ligne) (Rochelandet, 2010). Les entreprises peuvent ensuite valoriser ces données en les revendant ou en les exploitant (par exemple: Facebook et la publicité comportementale).

security, ACM, pp. 85-90.

FORAY, D. (1989), "Les modèles de compétition technologique. Une revue de la littérature", *Revue d'économie industrielle*, Vol. 48, n° 1, pp. 16-34.

HARSH, P., DUDOUE, F., CASCELLA, R.G., JEGOU, Y., & MORIN, C. (2012), "Using open standards for interoperability: issues, solutions, and challenges facing cloud computing", 2012 8th International Conference on Network and Service Management (CNSM), IEEE, pp. 435-440.

MARSTON, S., LI, Z., BANDYOPADHYAY, S., ZHANG, J., & GHALSASI, A. (2011), "Cloud computing - The business perspective", *Decision Support Systems*, 51(1), pp. 176-189.

MESSERSCMITT, D. G., SZYPERSKI, C. (2001), "Industrial and Economic Properties of Software: technology, processes and value", Computer Science Division, University of California.

PAHL, C., ZHANG, L., & FOWLEY, F. (2013), Interoperability standards for cloud architecture, Intl. Conference on Cloud Computing and Services Science Closer'13.

PEARSON, S. (2009), Taking account of privacy when designing cloud computing services", ICSE Workshop on Software Engineering Challenges of Cloud Computing (CLOUD'09), IEEE, pp. 44-52.

QUÉLIN, B. (2003), "Externalisation stratégique et partenariat: de la firme patrimoniale à la firme contractuelle?", *Revue française de gestion*, n° 2, pp. 13-26.

ROCHELANDET, F. (2010), "Économie des données personnelles et de la vie privée", Collection Repères, Editions La Découverte, Paris.

SCOFFONI, P., OGEZ, E., & DUBOST, L. (2012), "L'open cloud : garder la maîtrise de son système d'information", *Méthodes, techniques et outils, Documentaliste-Sciences de l'Information*, 2012/2 (Vol. 49), pp. 8-15.

WISEUR, R. (2012), "Les moteurs de recherche commerciaux sont-ils des outils de webométrie fiables ?", 30ème congrès InforSID, Montpellier (France), 29-31 mai 2012.

WISEUR, R. (2013a), "Estimate the Market Share from the Search Engine Hit Counts", Proceedings of International Conference on Data Technologies and Applications 2013, Iceland, July 29-31, 2013.

WISEUR, R. (2013b), "Evolution des stratégies et modèles d'affaires des éditeurs Open Source face au Cloud computing", *Terminal : Technologie de l'Information, Culture, Société*, n°113-114, pp. 173-193.

ZHANG, Z., WU, C., et CHEUNG, D. WL. (2013), "A survey on cloud interoperability: taxonomies, standards, and practice", *ACM SIGMETRICS Performance Evaluation Review*, Vol. 40, n° 4, pp. 13-22.

## **5.2. Références professionnelles**

ASAY, M. (2011), "Facebook, Google, and the war to lock you in", *The Register*, 5 juillet 2011, URL: [http://www.theregister.co.uk/2011/07/05/google\\_facebook\\_and\\_the\\_closed\\_web/](http://www.theregister.co.uk/2011/07/05/google_facebook_and_the_closed_web/)

BABCOCK, C. (2013), "10 Tools To Prevent Cloud Vendor Lock-in", *InformationWeek*, 14 février 2013, URL: <http://www.informationweek.com/infrastructure/cloud-infrastructure/10-tools-to-prevent-cloud-vendor-lock-in/d/d-id/1108679?>

CHATZAKIS, A. (2012), "How I Learned not to fear Amazon Cloud Lock-in", *Newvem*, 28 mai 2012, URL: <http://www.newvem.com/how-i-learned-not-to-fear-amazon-cloud-lock-in/>

CIGREF (2013), "Fondamentaux du cloud computing", *Cigref*, URL: <http://www.cigref.fr/cigref-publie-rapport-fondamentaux-cloud-computing>.

COFFEE, P. (2012), "Refuting Cloud 'Lock-In': Zero, One, Too", *Blog Network Salesforce.com*, 18 janvier 2012, URL: <http://cloudblog.salesforce.com/2012/01/cloud-leverage-is-not-lock-in.html>

COTE, M. (2008), "SalesForce-to-Google and the Force.com PaaS Lock-in Question", *Coté's People Over Process / RedMonk*, 24 juin 2008, URL: <http://redmonk.com/cote/2008/06/24/salesforce-to-google-and-the-forcecom-paas-lock-in-question/>

CROCHET-DAMAIS, A. (2013), "Marché du CRM : Salesforce passe devant SAP", *JDN*, 29 avril 2013, URL: [www.journaldunet.com/solutions/saas-logiciel/marche-du-crm-0413.shtml](http://www.journaldunet.com/solutions/saas-logiciel/marche-du-crm-0413.shtml).

- DARROW, B. (2013), "Fear of lock-in dampens cloud adoption", GigaOM, 26 février 2013, URL: <http://gigaom.com/2013/02/26/fear-of-lock-in-dampens-cloud-adoption/>
- PROTALINSKI, E. (2011), "Facebook finally makes your exported data useful", ZDNet, 8 septembre 2011, URL: <http://www.zdnet.com/blog/facebook/facebook-finally-makes-your-exported-data-useful/3452>
- GERMAIN, J.M. (2013), "How to Avoid Cloud Vendor Lock-In", LinuxInsider, 13 novembre 2013, URL: <http://www.linuxinsider.com/story/79417.html>.
- GRUMAN, G. (2007), "Is Open Source the Answer to ERP?", CIO, February, 2007, Vol. 20, pp.23-28.
- HARRIS, D. (2013), "Google defends, quantifies App Engine lock-in concerns", GigaOM, 9 juillet 2013, URL: <http://gigaom.com/2013/07/09/google-defends-quantifies-app-engine-lock-in-concerns/>
- JEAN, B. (2013), "Impact of cloud computing on FOSS users", Conférence EOLE (European Open source & free software Law Event) "Considering Free & Open Source in Cloud Strategies", Bruxelles, 6 décembre 2013.
- JUENGST, D. (2012), "How to Avoid Cloud Vendor Lock-In when Evaluating a PaaS", OpenShift, 3 août 2012, URL: <https://www.openshift.com/blogs/how-to-avoid-cloud-vendor-lock-in-when-evaluating-a-paas%20/>
- KASH, W. (2013), "Avoiding Cloud Lock-in", Information Week Government, Novembre 2013, URL: <http://www.informationweek.com/cloud/avoiding-cloud-lock-in/d/d-id/1111982>.
- LINTHICUM, D. (2012), "2 more cloud myths busted: Lock-in and locked up", InfoWorld, 27 avril 2012. URL: <http://www.infoworld.com/d/cloud-computing/2-more-cloud-myths-busted-lock-in-and-locked-191670>
- MAGNUSSON, P. (2013), "Google App Engine: Lock in, what lock in?", VB VentureBeat, 25 juillet 2013, URL: <http://venturebeat.com/2013/07/25/google-app-engine-lock-in-what-lock-in/>
- MALHOTRA, R. (2013), "The Truth About Lock In", CloudAve, 13 février 2013. URL: <http://www.cloudave.com/26426/the-truth-about-lock-in/>
- MCKENDRICK, J. (2011), "Cloud Computing's Vendor Lock-In Problem: Why the Industry is Taking a Step Backward", Forbes, 20 novembre 2011, URL: <http://www.forbes.com/sites/joemckendrick/2011/11/20/cloud-computings-vendor-lock-in-problem-why-the-industry-is-taking-a-step-backwards/>
- NACHMANI, O. (2012), "The Great PaaS Lock-In", DZone, 29 mai 2012, URL: <http://cloud.dzone.com/articles/great-paas-lock>
- OZERMAN, R. (2006), "Why is Flickr afraid of Zoomr?", Techcrunch, 16 juin 2006, URL: <http://techcrunch.com/2006/06/16/why-is-flickr-afraid-of-zoomr/>
- SZYNAKA, M. (2013), "Ask the Expert: Is PaaS vendor lock-in unavoidable?", SearchCloudComputing, 04 avril 2013, URL: <http://searchcloudcomputing.techtarget.com/answer/Ask-the-Expert-Is-PaaS-vendor-lock-in-unavoidable>
- WEINBERGER, M. (2012), "Cloud lock-in is a problem for everybody, not just businesses", CITEworld, 21 décembre 2012, URL: <http://www.citeworld.com/cloud/21228/consumers-need-beware-cloud-lock-just-enterprises>
- WOLPE, T. (2013), "Cloud lock-in: What firms can be doing now to avoid the traps", ZDNet, 5 juillet 2013, URL: <http://www.zdnet.com/cloud-lock-in-what-firms-can-be-doing-now-to-avoid-the-traps-7000017716/>

\*\*\*