



HAL
open science

Open source et systèmes critiques : le cas Thales

Nordine Benkeltoum

► **To cite this version:**

Nordine Benkeltoum. Open source et systèmes critiques : le cas Thales. 17ème Colloque de l'AIM, 2012, Bordeaux, France. pp.s73-1. hal-00905919

HAL Id: hal-00905919

<https://hal.science/hal-00905919>

Submitted on 18 Nov 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

OPEN SOURCE ET SYSTÈMES CRITIQUES : LE CAS THALES

Nordine Benkeltoum
ULNF, École Centrale de Lille, LM²O¹
Cité Scientifique - BP48
59650 Villeneuve d'Ascq
+33 20 67 60 54
nordine.benkeltoum@ec-lille.fr

¹Laboratoire de Modélisation et de Management des Organisations

OPEN SOURCE ET SYSTÈMES CRITIQUES : LE CAS THALES

RÉSUMÉ

La littérature en SI (Système d'Informations) s'est largement intéressée aux technologies open source tant du point de vue des stratégies IT (Information Technologies), des modèles d'affaires que de la phénoménologie organisationnelle. Toutefois, les travaux en management des SI se sont peu penchés sur l'utilisation de l'open source en *environnement critique*. En s'appuyant sur une étude exploratoire, cet article décrit la manière dont Thales, une entreprise spécialisée dans des domaines hautement sensibles (défense, aéronautique et sécurité), a structuré une stratégie IT *ad hoc* afin de répondre aux défis lancés par l'utilisation des technologies open source. Cette stratégie repose sur la mise en place d'un centre de compétences chargé de l'acquisition, du développement et de la transmission des savoirs internes et externes. Sur la base de ce matériau et de la littérature, l'article propose un modèle pour l'intégration de l'open source dans la stratégie IT.

Mot clés : open source, environnement critique, stratégie SI, stratégie IT

ABSTRACT

Information Systems (IS) literature shows a huge interest for open source technologies as well as from IT (Information Technology) strategies, business models or organization. Nevertheless, research on the application of open source in *mission-critical* systems is particularly scant. By means of an exploratory study, this article describes the way by which Thales, a firm specialized in mission-critical fields (defense, aircraft industry and security), has setup an *ad hoc* IT strategy that integrates open source technologies. This strategy relies on a competence center which aiming at acquisition, development and transmission of knowledge inside and outside firm boundaries. Based on this empirical material and the relevant literature, this article suggests a framework for the integration of open source in IT strategy.

Keywords: open source software, mission-critical, IT strategy, IS strategy

INTRODUCTION

Les logiciels libres ou open source offrent des libertés d'exécution, d'étude, d'amélioration et de distribution. Ces technologies sont devenues incontournables dans le domaine des SI (systèmes d'informations) aussi bien en théorie (Fitz-Gerald 2010 ; Gary et al. 2011) qu'au niveau de la pratique (CIGREF 2011 ; Le Texier et Versailles 2009 ; Lindman et al. 2011 ; Lisein et al. 2009). Un récent rapport a montré que plus de la moitié des entreprises avait intégré l'open source dans leur stratégie IT (Information Technologies) (Gartner Inc. 2011). Au niveau français, une étude souligne que 92% des entreprises interrogées déclarent avoir d'ores et déjà adopté l'open source dans leurs projets IT (MARKESS International 2009).

D'un point de vue technologique, SourceForge.net recense à ce jour plus de 324 000 projets (Geeknet Inc 2011) contre 230 000 en 2009 (Geeknet Inc. 2009) et 150 000 en 2008 (Deshpande et Riehle 2008) ce qui correspond à une croissance de 116% pour la période 2008-2011. Considéré comme le plus grand hébergeur de projets open source, le site enregistre près de 4 millions de téléchargements par jour (Geeknet Inc 2011). D'après certains spécialistes, la base de code open source suit une sorte de loi de Moore (Benkeltoum 2011a ; Fitzgerald 2006). En somme, l'open source croit de manière fulgurante et aucun signe ne laisse présager qu'il s'essoufflera à court ou moyen terme.

Sur le plan économique, l'IDC estimait que, pour l'année 2011, le marché s'élèverait à 5,8 milliards de dollars (Broersma 2007). D'après ce même institut, ce marché représentera près de 8,1 milliards de dollars en 2013 et présentera un taux de croissance de 22,4% par an (IDC 2009). Avec ses quelques 1,47 milliards d'euros en 2009, la France est le premier marché Européen (PAC 2010). Pourtant les premiers travaux sur l'open source prétendaient qu'il n'y avait pas de marché pour ces logiciels (von Hippel et von Krogh 2003: 214). D'autres auteurs soutiennent que l'open source a subi de profondes mutations (Fitzgerald 2006), que sa physionomie nécessite d'être revisitée et que bon nombre d'idées reçues méritent d'être révisées (Benkeltoum 2011b).

La littérature en SI (Système d'Informations) s'est largement intéressée aux technologies open source dont les caractéristiques font qu'il promeut la recherche (von Krogh et Spaeth 2007). Des travaux se sont ainsi intéressés aux méthodes de développement (Agerfalk et Fitzgerald 2008 ; Haefliger et al. 2008), aux structures organisationnelles et à leur dynamique (Benkeltoum 2011a ; Stuermer et al. 2009) ou encore aux modèles d'affaires (Bonaccorsi et al. 2006 ; Lisein et al. 2009 ; Välimäki 2003). En revanche, peu de travaux se sont penchés sur la question de l'utilisation de l'open source pour les *systèmes critiques*.

Ainsi, cette recherche analyse la manière dont Thales, une entreprise spécialisée dans les domaines de la défense, l'aéronautique et la sécurité, a adopté l'open source en l'intégrant à sa stratégie IT. Pour cela, ce papier sera structuré comme suit. La première section offrira une revue des travaux sur les entreprises et l'open source. La seconde proposera une revue des travaux existants sur l'application des technologies open source pour la création de systèmes critiques. La troisième abordera les motivations et la méthodologie qui sous-tendent de cette recherche. La quatrième décrira la stratégie open source de Thales. La cinquième proposera un modèle pour intégrer l'open source à la stratégie IT. Enfin, la dernière section reviendra sur les apports de cette étude de cas et discutera de ses implications théoriques et pratiques.

1. L'OPEN SOURCE ET LES ENTREPRISES

1.1. De réseaux d'utilisateurs à la mutation d'une industrie

Le *Free Software* est aussi ancien que l'idée de logiciel (Adam et al. 2003), il a historiquement débuté comme un réseau (von Hippel 2007) où des *utilisateurs experts* partageaient des instructions informatiques dans un contexte (années 70) où matériel et logiciels étaient imbriqués. L'évolution des langages de programmation, la standardisation des environnements informatiques et le développement d'internet ont concouru à l'expansion et à la démocratisation du phénomène open source² (Benkeltoum 2011a). Dès lors, la littérature s'est intéressée aux facteurs justifiant la participation des utilisateurs-développeurs. Parmi les raisons évoquées³, on retrouve l'amélioration de la réputation et les effets sur la carrière professionnelle (Lerner et Tirole 2001), l'apprentissage (von Hippel et von Krogh 2003), la réciprocité et la promotion de la communauté open source (Lakhani et von Hippel 2003).

Face au succès grandissant de certains logiciels comme Linux, Apache, PHP ou encore PERL, les logiciels open source ont intéressé les entreprises jusqu'à devenir un des éléments incontournables de l'économie du logiciel (Jullien 2003). Le regard scientifique s'est alors penché sur les raisons justifiant la participation des firmes. Une première étude a montré que les firmes étaient principalement motivées par les aspects économiques et technologiques du phénomène (Bonaccorsi et Rossi 2003). Pour d'autres, les firmes s'impliquent car elles souhaitent améliorer la compatibilité de leurs produits (non libres) avec les logiciels libres, promouvoir des standards afin d'assurer l'interopérabilité entre systèmes, réduire les risques d'investissement et donner un meilleur avenir à l'innovation (Rannou et Ronai 2003).

Dans un premier temps, les firmes eurent un rôle de financement des communautés. Très tôt des auteurs ont souligné qu'un nombre important de contributeurs étaient employés par des grandes entreprises de l'industrie informatique (Ghosh et al. 2002). Aujourd'hui, on peut dire que contrairement à une idée reçue, la majorité des développeurs de logiciels libres n'est pas bénévole mais perçoit une compensation directe ou indirecte (Benkeltoum 2011b). À titre d'exemple, 70% des modifications réalisées sur le noyau Linux (le cœur du système d'exploitation GNU/Linux) est attribuable à des entreprises (Kroah-Hartman et al. 2009). On peut également citer le renouvellement de l'accord entre Mozilla et Google qui d'après certaines sources s'élèverait à un milliard de dollars sur trois ans (ZDNet.fr 2011). Ce simple exemple justifie à quel point la représentation collective de l'open source doit évoluer (Fitzgerald 2006) et les idées reçues révisées (Benkeltoum 2011b). Toutes, les entreprises ne s'investissent pas de la même façon dans l'open source, on distingue ainsi quatre niveaux d'investissement : l'utilisation, la proposition de logiciels libres comme complément, l'adoption de l'open source comme méthode de production et enfin l'adaptation du modèle d'affaires aux spécificités de ces logiciels (Grand et al. 2004).

1.2. Les modèles d'affaires de l'open source

En permettant aux utilisateurs de l'exécuter, le consulter, le modifier et de le distribuer, le logiciel libre a remis en question les méthodes de valorisation dominantes dans l'industrie du logiciel (Meissonier et al. 2010). Ainsi, c'est toute la chaîne de valeur du logiciel qui a été rediscutée (Benkeltoum 2011a). La littérature distingue trois catégories de modèles⁴ permettant de générer et capturer de la valeur (Chesbrough 2007) : les modèles d'affaires ouverts, hybrides et fermés.

² Dans cet article, logiciel libre, free software et logiciel open source seront utilisés de manière interchangeable. Le lecteur intéressé par les différences fondamentales entre ces notions pourra se référer à Adam et al. (2003) ou encore à Benkeltoum (2011c).

³ Une récente recherche a relancé le débat sur les motivations en montrant que les phénomènes d'apprentissage et d'idéologie expliquaient peu la participation (Meissonier et al. 2010).

⁴ Pour une description plus fine des modèles d'affaires se référer à Bonaccorsi et al. (2006), Lisein et al. (2009) et Benkeltoum (2011c).

Les modèles d'affaires ouverts consistent à valoriser les logiciels libres par le biais de services associés (Jullien 2003) comme la maintenance, les développements spécifiques, le conseil, la formation ou le support (Dahlander 2005 ; Onetti et Capobianco 2005). Ce modèle de valorisation indirect est particulièrement en phase avec la tendance générale de l'informatique qui passe d'une logique de produits à une logique de services (Sharma et al. 2002). Une autre manière de générer des revenus est de s'appuyer sur une expertise pointue et difficilement imitable. Même si le code est ouvert, la firme est pratiquement la seule à être capable de comprendre et de modifier son logiciel (Benkeltoum 2009 ; Lisein et al. 2009). Toutefois, les entreprises adoptant un modèle purement ouvert connaissent de nombreuses difficultés pour créer, capter de la valeur et pérenniser leur activité (Benkeltoum 2009 ; Lisein et al. 2009). Seules les entreprises à forte expertise comme Wallix ou ScalAgent réussissent à construire un avantage concurrentiel durable (Benkeltoum 2011a).

Les modèles d'affaires hybrides proposent de combiner les moyens classiques de valorisation : (la cession de licences) et les revenus beaucoup plus liés aux logiciels libres (Bonaccorsi et al. 2006). Le « *dual-licensing* » est l'archétype de ce modèle puisque dans cette stratégie le logiciel est à la fois distribué sous une licence libre et non libre (Välimäki 2003). Diverses variantes de la double licence sont répertoriées : le couplage technologique (West 2003), l'intégration dans un produit non libre (Onetti et Capobianco 2005) ou encore le produit complémentaire (Muselli 2008). D'autres stratégies visent à cibler une clientèle peu compétente avec des solutions combinant des applications libres et non libres (Lisein et al. 2009).

Les modèles d'affaires fermés visent à rendre la technologie quasi-fermée ou fermée en prenant pour base du code open source. Ces modèles reposent sur diverses stratégies qu'il convient de détailler. La première stratégie dite du « *système clos* » vise à développer un produit dont la diffusion est réduite à la communauté des utilisateurs (Lisein et al. 2009) et permet ainsi de profiter d'un produit « *avec le goût de l'open-source, la saveur de l'open-source et le prix du propriétaire* » (Elie 2010). La seconde stratégie consiste à intégrer du code open source dans un appareil combiné à l'une de ces pratiques : soit protéger le système contre l'exécution d'un code modifié par le biais d'un procédé technique ; soit commercialiser ou louer un appareil tout en conservant secret le code source (Benkeltoum 2011c). La troisième stratégie consiste à réaliser un dérivé non libre à partir d'un ou plusieurs logiciels libres ou à combiner les approches SAAS (Software As A Service) et du code open source pour proposer un service en ligne (Benkeltoum 2011c).

1.3. Les enjeux de l'adoption de l'open source

Les avantages des technologies open source sont nombreux et bien documentés. D'abord, les logiciels libres ont une structure de qualité (Capra et al. 2011 ; von Krogh et Spaeth 2007), les bugs sont rapidement corrigés (Bitzer et Schröder 2005 ; Paulson et al. 2004). De plus, l'open source assure la transparence (Spinellis et al. 2009), l'interopérabilité, l'indépendance technologique (Benkeltoum 2011b) et plus particulièrement pour les pays en voie de développement (Chen et al. 2010). D'autres caractéristiques sont plus controversées. Par exemple, certains auteurs ont montré que l'open source favorise la réutilisation du code au niveau interne et entre projets (Haefliger et al. 2008) même si pour d'autres les voies d'améliorations sont nombreuses (Capiluppi et al. 2011). De même, une étude a prouvé que les logiciels libres n'avaient pas une structure plus modulaire que les logiciels fermés (Paulson et al. 2004) alors qu'une autre a défendu exactement l'inverse (MacCormack et al. 2006).

Si pour certaines entreprises et administrations l'adoption de l'open source pour remplacer des technologies fermées est évalué à travers le prisme du « *Total Cost of Ownership* » (TCO) (Fitzgerald 2006) ou encore par le biais du rapport entre coûts et bénéfices (Le Texier et Versailles 2009 ; Taibi et al. 2007). Pour d'autres entreprises, ce choix peut être confronté à de grandes résistances. Parmi ces freins figure la qualité des logiciels libres qui contrairement à ce que certains

auteurs prétendent (Raymond 1999 ; Rossi 2009 ; Spaeth et al. 2007) n'est pas aussi claire (Fuggetta 2003). De plus, la plupart des projets open source hébergés sur SourceForge.net sont estampillés inactifs (Deshpande et Riehle 2008). Les projets Européens Qualoss et QualiPSO montrent qu'il existe de réelles interrogations sur l'évaluation des logiciels open source (Benkeltoum 2011a). Dès lors, des recherches ont démontré que ces logiciels doivent être évalués en utilisant des métriques spécifiques comme le degré d'activité des listes de diffusion, la qualité de la documentation ou encore les coûts de déploiement et de formation (del Bianco et al. 2009 ; Lee et al. 2009 ; Spinellis et al. 2009 ; Taibi et al. 2007).

2. OPEN SOURCE ET SYSTEMES CRITIQUES : ETAT DES CONNAISSANCES

2.1. Le système critique en recherche

Dans le domaine de la défense, un système critique ou « *mission-critical computer system* » se définit comme un ensemble d'équipement, de logiciels et de services dont le fonctionnement implique (1) des activités de renseignement, (2) des données relevant de la sécurité nationale, (3) le contrôle de ressources militaires, (4) un équipement intégré en tout ou partie dans un dispositif d'armement ou encore qui est (5) critique pour la réalisation d'une activité militaire ou pour le renseignement (Bloom et Chung 2001: 218). Depuis le début des années 80, l'incorporation de code informatique dans les systèmes militaires connaît une croissance vertigineuse (Austin et Larkey 1992). De manière plus générale, la sécurité des personnes repose sur ce type de systèmes (Gary et al. 2011).

Un système est dit « *critique* » en raison de la sensibilité des activités que celui-ci supporte et lorsqu'il assure des opérations pour lesquelles un blocage ou une panne peut avoir des conséquences importantes voire dramatiques comme la perte de vies. Un tel système doit donc avoir un haut degré de fiabilité et de correction (Gary et al. 2011), une forte tolérance aux pannes, assurer les services importants dans des conditions extrêmes même si une partie de ce dernier est compromise à cause d'une attaque externe (Min et Choi 2004). Ainsi, les systèmes critiques se retrouvent dans divers domaines sensibles comme la gestion du trafic aérien, l'aiguillage ferroviaire, l'information financière, les missions spatiales (Norris 2004) ou encore la chirurgie (Gary et al. 2011). Dans le domaine aérien, les systèmes critiques sont régis par des normes comme ED-12B ou DO178. Cette dernière distingue cinq niveaux de criticité allant du niveau A (une défaillance peut mettre en péril la sécurité du vol) au niveau E (un défaut n'aura aucun effet sur la sécurité du vol) (Ferrell et Ferrell 2001).

La plupart des composants open source liés à la sécurité sont développés par des entreprises, leur maturité est grandissante « *avec un petit bémol⁵ sur la base de données et les composants de sécurité (authentification, identification, ...).* » Par ailleurs, ils intéressent les entreprises pour leur faible coût (Lawton 2002).

L'application de l'open source dans de tels systèmes se heurte à d'importants freins. D'abord, du fait de l'ouverture du code source certains considèrent que le SI est davantage sujet à l'identification de failles de sécurité ou de *backdoors* (portes dérobées). C'est finalement l'intégration de composants inconnus qui pose le plus problème pour les professionnels de la sécurité et non le fait que le logiciel soit open source (Lawton 2002). Ensuite, la libération du code d'une application sensible est difficilement acceptable pour les praticiens comme dans le cas SAP développé à la NASA (Norris 2004). Des idées bien ancrées comme la « *sécurité par l'obscurité* » (fait de conserver secret le code source d'un programme afin de protéger un SI) ont du mal à être balayées (Hansen et al. 2002). Et ce malgré les importantes limites soulignées par différents spécialistes. De nombreuses failles de sécurité sont découvertes dans les logiciels fermés. Par

⁵ Responsable du Centre de Compétences Open Source, Thales D3S, entretien avec l'auteur.

exemple, des failles de sécurité découverte sur l'iPhone permet d'envoyer des données à l'insu de l'utilisateur (Le Boulout 2011). Ensuite, le « *reverse-engineering* » ou rétro-ingénierie (passage du code objet au code source) est une pratique courante parmi les spécialistes de l'informatique (Hansen et al. 2002).

2.2. L'open source au service des systèmes critiques

L'utilisation massive de l'open source en entreprise (IDC 2009) a montré qu'il existait des disparités importantes de niveau de maturité en fonction des domaines. Si le développement, l'administration, les technologies internet, les serveurs web ou d'applications ont un degré de maturité élevé, d'autres domaines comme les applications métiers, la sécurité sont beaucoup moins matures (CIGREF 2011). En outre, l'utilisation de l'open source en environnement critique fait l'objet de peu de travaux même si en pratique ce phénomène est connu (Lawton 2002 ; Lundell et al. 2010) et loin d'être négligeable (Gartner Inc. 2011).

Une étude portant sur l'adoption de l'open source au sein de la défense américaine (DoD) illustre bien l'engouement existant au sein des administrations publiques. Par exemple, La National Security Agency (NSA) a conçu sa propre version de Linux nommée Security Enhanced Linux (SEL) qui vise à supporter des applications critiques. De même, l'US NAVY a adopté des composants open source pour bâtir des applications et a mis en place une politique visant à écarter toute technologie fermée ou propriétaire (Le Texier et Versailles 2009).

Un retour d'expérience sur la mission américaine *Mars Exploration Rover* aborde explicitement le sujet. En effet, le *Science Activity Planner* (SAP), un outil d'analyse des relevés, s'appuie sur des composants open source comme Castor, Java Expression Parser, VRML97 ou le plus connu MySQL. La NASA classe SAP en tant qu'application critique car un dysfonctionnement dans ce programme pourrait compromettre une partie des opérations. Le SAP a clairement été conçu sur la base de technologies open source pour réaliser des économies de développement. Grâce à des accords spécifiques avec les auteurs des composants open source, la NASA a réussi à ne pas publier le code source du SAP. D'après le responsable du projet SAP, les composants open source avaient un niveau de qualité supérieur à leur équivalent non libre. Plusieurs éléments doivent guider le choix des composants à intégrer : la maturité du logiciel (stade de développement), la longévité (taille de l'équipe) et la flexibilité du projet (réactivité de l'équipe) (Norris 2004).

Un autre retour d'expérience combine l'open source et les méthodes agiles pour un logiciel de chirurgie guidée par l'imagerie médicale. Ce logiciel est distribué sous licence permissive cela signifie qu'il peut être modifié, lié ou simplement intégré avec du code libre ou fermé. L'objectif de ce logiciel est de servir de socle technologique pour la création d'applications pour la chirurgie. Il s'agit bel et bien d'un système critique pour la sécurité des personnes ou *safety-critical*. En revanche, cette recherche est orientée vers l'ingénierie du développement et non le management des SI (Gary et al. 2011). Au niveau Européen, un consortium d'entreprises (Airbus, Thales, Turbomeca,...) coordonne le développement de *Topcased*, un ensemble d'outils destinés à être utilisés pour des systèmes critiques et notamment dans le domaine aérien.

3. MOTIVATIONS ET MÉTHODOLOGIE DE LA RECHERCHE

3.1. Motivations de la recherche

La littérature a amplement mis en évidence le caractère incontournable de l'open source pour les praticiens et les théoriciens en SI. Les différentes stratégies de mobilisation (Dahlander et Magnusson 2008), d'implication (Stuermer 2009) et d'allocation de ressources (Grand et al. 2004),

les modèles d'affaires (Lisein et al. 2009), les problématiques d'évaluation (Capra et al. 2011) et d'adoption (Ajila et Wu 2007) ont reçu une considérable attention académique. En revanche, peu d'attention a été donnée à l'utilisation de l'open source dans des domaines critiques ou sensibles nécessitant un haut degré de fiabilité. Cette recherche abordera la question suivante : comment intégrer l'open source dans la stratégie IT des entreprises concevant des systèmes critiques ?

3.2. Approche méthodologique

Technique d'échantillonnage

Plusieurs éléments justifient le recours à l'étude de cas exploratoire. D'abord, cette recherche tente de comprendre le *comment* et le *pourquoi* d'un phénomène (Leonard-Barton 1990). Ensuite, la pauvreté des travaux sur management des technologies open source en environnements critiques ou sensibles, le caractère émergent de ce type de pratiques, justifient davantage le recours à cette méthode. En effet, l'étude de cas peut être employée pour décrire, tester une théorie ou encore générer une théorie. Cette stratégie de recherche repose sur la sélection d'un ou plusieurs cas et de nombreux niveaux d'analyse (Eisenhardt 1989). Ce ou ces cas doivent particulièrement être caractéristiques d'un phénomène à observer (Leonard-Barton 1990).

La littérature a prouvé que la validité des théories basées sur un nombre limité de cas avaient la même validité que celles basées sur un grand échantillon (Christensen 2006 ; Fitzgerald 1997) et l'étude de cas est désormais une méthodologie unanimement reconnue (Eisenhardt 1989 ; Eisenhardt et Graebner 2007). Les études de cas combinent des données qui peuvent varier en nature (qualitative, quantitative) et en fonction de leur provenance (archives, interviews, etc.) (Eisenhardt 1989 ; Leonard-Barton 1990).

Source de données

Thales a été étudiée par l'auteur dans le cadre d'un projet de recherche national destiné à analyser l'utilisation de l'open source en entreprise. De manière plus précise, l'auteur a eu l'opportunité d'assister à l'émergence et au développement d'un centre de compétences dédié aux technologies open source au sein de la division Thales Division Services (Thales D3S) entre 2006 et 2009 ce qui correspond à une étude de cas longitudinale (Pettigrew 1990).

Thales a été analysée par le biais de plusieurs sources de données (Eisenhardt 1989 ; Pettigrew 1990). La première source correspond à des interviews semi-structurées et des réunions (durée allant d'une à deux heures) en face à face, par internet ou téléphone réalisées avec des acteurs clés (Pettigrew 1990) de Thales. Ces interlocuteurs occupaient les fonctions suivantes : Directeur de la recherche et de l'innovation, Directeur Recherche et Technologie Asie, Responsable du centre de compétence, Open source Architect, Open source Engineer. Étant donné que l'open source est un cas extrême d'open innovation (Dahlander et Wallin 2006), il était pertinent d'étendre l'analyse vers l'extérieur de la firme (Chesbrough 2003) afin d'enrichir l'analyse auprès de partenaires. Ces données ont été retranscrites manuellement. La seconde source reposait sur des échanges informels dans le cadre de séminaires. Parmi ces derniers figure la participation à une conférence internationale dédiée à l'open source durant trois jours avec un responsable de projets européen. La troisième source correspond à des documents internes, des communiqués de presse, des informations publiques ou encore des présentations orales ou écrites. La quatrième correspond à des échanges de courriels sur des points particuliers. Cette dernière source a particulièrement été riche mais il est difficile de chiffrer le volume des échanges. Ces sources de données sont synthétisées ci-dessous (tableau 1).

Tableau 1 : sources de données

	Entretiens	Courriels	Conférences	Documentations internes et officielles
Thales Group	7	X	2	X
Partenaires	4	X	1	X

Analyse de données

L'approche méthodologique de cette recherche est une étude de cas longitudinale (Pettigrew 1990) où l'activité open source de Thales D3S a été analysée. La majorité des données étant qualitatives leur traitement a pris appui sur un raisonnement inductif et plus particulièrement sur la « *Grounded Theory* » (Glaser et Strauss 1967). Dans ce cadre d'analyse, la démarche de codification des données est basée sur un rapport dynamique entre les données et l'interprétation. Sur la base des données du cas Thales, nous proposons un modèle pour l'implémentation de l'open source dans la stratégie IT dans une entreprise opérant sur le marché des systèmes critiques.

4. OPEN SOURCE ET APPLICATIONS CRITIQUES : LES ENSEIGNEMENTS DU CAS THALES

3.1. Contexte

Présentation générale

Thales est un groupe spécialisé en électronique et systèmes. L'entreprise opère plus spécifiquement dans les domaines de la défense, l'aéronautique et la sécurité. Thales représente près de 70 000 personnes au niveau international dont plus d'un tiers de chercheurs. La firme investit chaque année près d'un cinquième de son activité en R&D et est reconnue pour son expertise dans le domaine des hautes technologies. Elle partage différents laboratoires de recherche avec des institutions académiques. À titre d'exemple, Albert Fert, Directeur de l'unité mixte de recherche en physique CNRS-Thales, a remporté le Prix Nobel de Physique (2007) pour ses travaux sur la magnétorésistance.

Freins et défis à l'adoption

La dépendance technologique vis-à-vis d'éditeurs de logiciels pour la plupart détenus par des puissances étrangères posait des problèmes importants dans le domaine de la défense notamment. Cet élément a largement incité le groupe Thales à s'interroger sur l'utilisation massive des technologies open source pour *reconquérir* son indépendance technologique. En revanche, du fait que Thales propose des systèmes dans des environnements hautement sensibles (gestion du trafic aérien, systèmes de surveillance et de sécurité, ...), l'adoption de l'open source s'est heurtée à d'importants freins.

Le passage à l'open source « *était [...] douloureux, mais très intéressant. Notamment dans le domaine de la défense. L'open source posait problème pour certains composants « sensibles* »⁶. » Bien que « *pour [tous] les ministères la criticité n'est pas la même. Pour le Minefi [Ministère de l'économie et des finances], c'est l'argent du citoyen. Pour la Ministère de la Défense, c'est le fantassin, comment faire pour ne pas perdre un seul fantassin.* »⁷ Certains donneurs d'ordres refusaient tout simplement de publier les modifications réalisées dans le code des logiciels libres

⁶ Responsable du Centre de Compétences Open Source, Thales D3S, entretien avec l'auteur.

⁷ Responsable du Centre de Compétences Open Source, Thales D3S, entretien avec l'auteur.

utilisés dans le cadre de projets critiques. La notion de *sécurité par l'obscurité* (Norris 2004) est bel et bien ancrée dans le domaine de la défense.

Thales a donc essayé de trouver un compromis en étudiant les licences. Les experts juridiques ont trouvé que rien n'obligeait « à publier » avec les licences open source. Par conséquent, il est possible de réaliser une solution basée sur du code open source et donner uniquement les modifications ainsi réalisées au client ou à l'utilisateur du logiciel. Et ce y compris sous licences virales de type GPL (General Public Licence). Le caractère « viral » se dit lorsque la licence d'un code source oblige tout code ajouté ou modifié à être de nouveau couvert par cette même licence. Lorsqu'un code a un enjeu technico-économique fort alors il n'est pas publié. Cette stratégie est courante puisque « *c'est aussi le [choix] des plus grands industriels qui participent à l'open source*⁸. » et celle-ci a été identifiée sous le nom de *selective-revealing* (Henkel 2006). En outre, les licences libres ont posé d'autres problèmes juridiques en France et à l'international (Lindman et al. 2011). Les clauses de *non-responsabilité* figurant dans certaines licences en cas de dommage causé par le logiciel sont nulles en droit français : celui qui distribue le logiciel en est responsable juridiquement.

3.2. Création et capture de valeur

Modèle d'affaires d'un systémier

Thales a investi dans l'open source car « *c'est viable d'un point de vue économique*⁹. » Les logiciels libres permettent une réutilisation de composants existants qui accélèrent les cycles de développement. L'industrie du logiciel devient de plus en plus orientée vers les services, surtout dans le domaine des grands SI où les clients ne souhaitent plus être dépendant d'un seul interlocuteur. Il y a également des motivations technologiques qui justifient le recours aux logiciels libres : la recherche d'une indépendance technologique.

Thales a une taille lui permettant de fédérer plusieurs projets industriels autour de briques logicielles communes. Enfin, le logiciel libre est utilisé dans des contextes sensibles car « *le modèle de la communauté est beaucoup plus réactif. Par exemple, le Minefi¹⁰ a eu un bug bloquant sur son site de paiement des impôts avec Jboss¹¹. Le problème a été résolu en 9 heures ! Il y a eu une version officielle qui prenait le correctif en 9 heures, c'est du jamais vu. Même les grands éditeurs n'ont pas cette réactivité. Les communautés sont beaucoup plus réactives¹². » De même, « *dans l'open source on a des capacités de débogage très grande bien meilleure que dans le domaine du logiciel propriétaire. Cela améliore la qualité du logiciel*¹³. »*

Le modèle d'affaires de Thales est assez singulier. L'entreprise crée de la valeur grâce aux technologies open source en réalisant des SI sur mesure pour les entreprises et administrations. Les logiciels libres sont utilisés comme des composants technologiques en couplant parfois technologies libres et fermées. Du fait de la haute spécificité des SI ainsi conçu, l'entreprise est capable de ne distribuer le code source qu'au client lorsque cela le requiert.

Cas Thalix

Thales a développé sa propre distribution Linux basée sur RHEL (Red Hat Entreprise Linux) nommée *Thalix*. Thalix est une distribution allégée en *paquets*¹⁴ afin d'assurer l'homogénéité et

⁸ Responsable du Centre de Compétences Open Source, Thales D3S, entretien avec l'auteur.

⁹ Responsable du Centre de Compétences Open Source, Thales D3S, entretien avec l'auteur.

¹⁰ Ministère de l'Économie et des Finances.

¹¹ Jboss est un serveur d'application open source.

¹² Responsable du Centre de Compétences Open Source, Thales D3S, entretien avec l'auteur.

¹³ Directeur de la Recherche et de l'Innovation, Thales D3S, entretien téléphonique avec l'auteur.

¹⁴ Les paquets sont les programmes compris dans une distribution Linux.

stabilité de la plateforme « dans les centres de contrôle aérien¹⁵. » Auparavant, « les serveurs utilisés avant l'arrivée de ces machines sous Thalix, étaient des « Stratus¹⁶ », dont le matériel et le système d'exploitation étaient contrôlés par un fournisseur unique. Il en découle les problèmes de compétence disponible, de réactivité de la hotline unique, en particulier face à des pannes matérielles dont quasiment personne n'avait l'expérience¹⁷. » Thalix est une distribution Linux spécifique très différente de ce qui est demandé sur le marché des systèmes d'exploitation. Pour la gestion du trafic aérien, « la stabilité de Thalix étant le critère principal, c'est une distribution qui maintient un retard de sécurité d'environ trois ans sur l'existant logiciel courant. Il en découle une compatibilité matérielle effective avec des ordinateurs contemporains de sa conception. » Thalix « n'est donc pas une distribution grand public, elle n'est donc pas distribuée comme telle. Elle est en revanche mise en avant dans les réponses à appel d'offre qui requièrent une stabilité exemplaire des serveurs (quand le matériel n'a pas de contrainte de modernisme particulier). » En 2009, un avion sur deux était géré par les systèmes de gestion du trafic aérien de Thales¹⁸.

3.3. Rôle du centre de compétences dans la stratégie IT

Afin d'assurer l'adoption de l'open source, Thales a créé un Centre de Compétences Open Source (CCOS) chargé de développer les connaissances nécessaires et de s'impliquer dans les différentes communautés. Ce centre a été initié dans un contexte R&D (Recherche et Développement) et répondait à une anticipation de l'évolution des méthodes de développement : d'une logique fermée à une logique ouverte appliquée au domaine du SI et des services. Le rôle fondamental de ce centre est d'explorer le champ de l'open source et de construire un réseau de partenariats stratégiques avec des moyennes et des grandes entreprises¹⁹. Thales D3S est la partie du groupe la plus en pointe en matière d'open source. En effet, cette division coopère avec l'ensemble des autres divisions pour promouvoir les technologies open source. Le centre sert d'interface entre les projets commandités par les clients (Marine Nationale, Ministères, ...) et les composants open source développés dans le cadre de communautés ou de consortiums industriels. Ce schéma propose une vision simplifiée du rôle du CCOS.

¹⁵ Open Source Architect n°1, Thales D3S, entretien en ligne avec l'auteur.

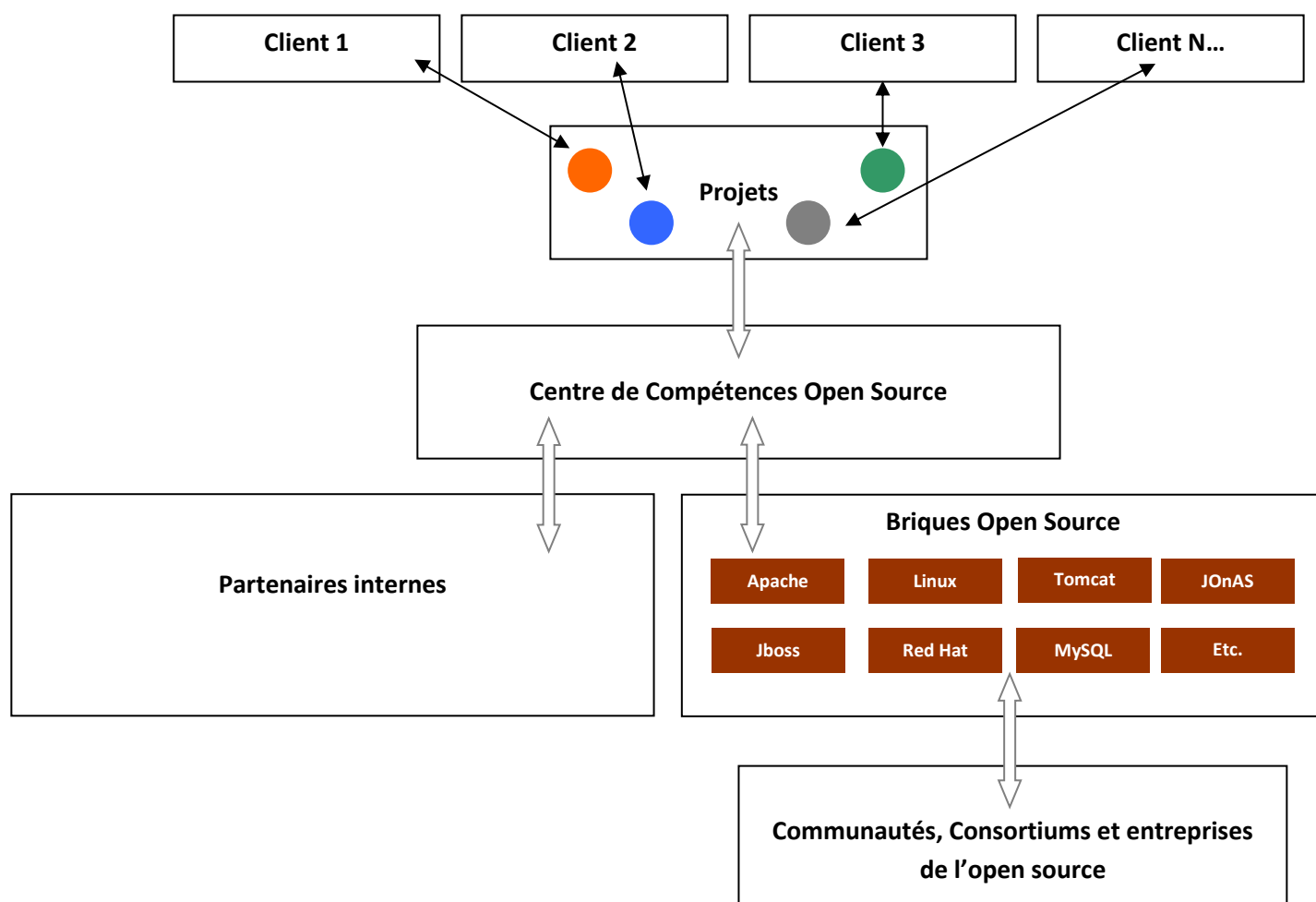
¹⁶ Stratus Technologies propose des serveurs et systèmes pour les environnements critiques.

¹⁷ Open Source Architect n°1, Thales D3S, entretien en ligne avec l'auteur.

¹⁸ Thales Group, THALES Presence in Air Traffic Management, Janvier 2009.

¹⁹ Basé sur les propos d'un Open Source Architect, Thales D3S, entretien avec l'auteur.

Figure 1 : Le rôle du Centre de Compétences Open Source



Le CCOS assure la pérennité de l'architecture logicielle et des applications. En d'autres termes, Thales crée une architecture orientée services où les composants open source peuvent évoluer ou même être remplacés. Cette évolution est essentiellement conditionnée par la technologie et les besoins clients. Dans le domaine informatique la technologie et les standards évoluent très rapidement ce qui nécessite une flexibilité technique poussée. De même, l'architecture peut évoluer au gré des besoins métiers et clients sans pour autant remettre en question le fonctionnement du système.

Le CCOS est impliqué dans différentes communautés. Comme l'indique un responsable de Thales, « plus on fait de l'open source, plus on est crédible dans le monde de l'open source. L'implication dans les communautés devient un atout, il y a une preuve de la compétence : c'est la présence dans des communautés²⁰. » Thales ne bâtit pas sa légitimité sur l'édition de produits logiciels comme d'autres acteurs de l'informatique (Dassault Systèmes, Microsoft, ...) mais plutôt car « Thales est capable de faire des systèmes de surveillance pour la police, des systèmes de gestion du trafic, etc. On ne choisit pas Thales car on s'appelle IBM Websphere²¹. »

3.4. Thales et le consortium OW2

Présentation d'OW2

²⁰ Responsable du Centre de Compétences Open Source, Thales D3S, entretien avec l'auteur.

²¹ Directeur de la Recherche et de l'Innovation, Thales D3S, entretien téléphonique avec l'auteur.

OW2 est un consortium d'entreprises et d'acteurs de la recherche ayant pour objet le *middleware*. Il s'agit d'un domaine technique assurant l'intermédiation entre l'infrastructure matérielle et les couches hautes. En plus de rassembler une base technologique, OW2 a la spécificité d'afficher clairement une dimension *affaires* que certaines communautés ont l'habitude d'occulter. Quatre catégories d'organisations coopèrent dans OW2 : les intégrateurs, les éditeurs, les acteurs de la recherche et de l'enseignement et les acteurs qui agissent en leur nom propre.

OW2 est une initiative *sino-européenne* visant l'indépendance technologique d'acteurs de l'industrie du *middleware* jusqu'ici dominée par des éditeurs américains. OW2 est donc le résultat de la convergence de deux stratégies visant à instaurer une indépendance technologique au niveau d'une partie *clé* du SI : le *middleware*. L'Europe et la Chine ont des intérêts communs. Contrairement aux États-Unis, l'Europe et la Chine ont plutôt des intégrateurs et peu d'éditeurs. Un ex-représentant de Thales au sein d'OW2 soulignait : « *on ne peut pas concurrencer les américains sur le marché du logiciel, c'est pour ça que l'Europe se tourne vers le logiciel libre. Ça conjugué à la montée de l'économie orientée services. Clairement le positionnement de l'Europe, c'est parce que l'on n'est pas capable de se positionner face à l'industrie américaine sur le marché du logiciel propriétaire*²². »

OW2 est un des éléments visant à renforcer la concurrence des entreprises européennes sur le marché mondial de l'informatique et des services. Pour certains grands acteurs, il n'était pas possible d'être dépendants technologiquement d'autres pays pour des composants aussi importants que le *middleware*. L'une des principales missions d'OW2 est de créer une infrastructure *middleware* libre. D'après l'ex-représentant de Thales au sein de l'équipe de direction d'OW2 : « *Le Middleware [...] est indispensable mais ce n'est pas là où l'on crée de la valeur. Thales, Bull, etc. ne sont pas des éditeurs de logiciels. Dans l'environnement propriétaire il y avait une dépendance américaine. Cette dépendance est technologique, ce n'est pas pour faire du « cocorico »... Les composants middleware ont une faible valeur ajoutée mais ils sont essentiels. Dans OW2, on partage le risque de mise à jour du middleware*²³. »

OW2 Consortium est considérée comme l'une des quatre organisations les plus influentes de l'open source avec les fondations Eclipse, Apache et la Linux. L'objet du consortium OW2 n'est pas intrinsèquement orienté vers la création de logiciels proposant de nouveaux usages. Le consortium OW2 vise plutôt à proposer des composants que les participants combineront avec d'autres technologies libres et non libres pour créer des solutions sur mesure ou des systèmes d'informations. Un directeur de Thales affirmait « *pour la communauté OW2, la focalisation est [...] le middleware et comme son nom l'indique, on est loin de l'utilisateur final et donc de l'innovation dans les usages*²⁴. »

*Rôle de Thales dans OW2*²⁵

Thales participait à OW2 pour trois principales raisons. D'abord dans « *une volonté d'indépendance technologique*²⁶ » qui s'inscrit dans la stratégie globale de Thales comme l'engagement de la firme pour la promotion des formats ouverts de documents. Ensuite, cette participation se justifie pour « *positionner Thales comme un acteur crédible dans le segment Open Source*²⁷. » Enfin, on retrouve la mutualisation. Selon l'ex-représentant de l'équipe de direction d'OW2, « *Historiquement, on utilisait OW2 comme vecteur pour disséminer. On a fait de la R&D en génie logiciel que Thales va mettre en open source. Libérer des logiciels dans lesquels il n'y a*

²² Open Source Architect n°2, Thales D3S, entretien en face à face avec l'auteur.

²³ Directeur de la Recherche et de l'Innovation, Thales D3S, entretien téléphonique avec l'auteur.

²⁴ Directeur de la Recherche et de l'Innovation, Thales D3S, entretien téléphonique avec l'auteur.

²⁵ Depuis 2010, Thales a quitté le consortium OW2.

²⁶ Directeur de la Recherche et de l'Innovation, Thales D3S, entretien téléphonique avec l'auteur.

²⁷ Responsable du Centre de Compétences Open Source, Thales D3S, entretien en ligne avec l'auteur.

pas de valeur pour Thales. OW2 c'est un socle technologique. L'intérêt, c'est le partage avec d'autres qui font aussi vivre le logiciel. Ce n'est pas qu'avec des couches intermédiaires que l'on fait de la valeur. D'autres personnes peuvent nous signaler des bugs qu'il nous reste plus qu'à corriger, c'est du travail qu'on n'a pas à faire : identifier les bugs²⁸. »

Thales était impliqué à tous les niveaux du consortium que ce soit au niveau de l'équipe de direction, du management ou du comité technique. Au niveau stratégique, Thales était un des acteurs clé du consortium et disposait d'un poids important sur les orientations de celui-ci. Au niveau tactique, Thales était leader d'une activité ayant pour but de sélectionner des composants open source d'OW2 et hors du consortium pouvant être mobilisés dans le cadre de projets de SI pour des administrations publiques en Europe et en Chine.

5. INTEGRATION DE L'OPEN SOURCE ET STRATEGIE IT POUR LES SYSTEMES CRITIQUES : LE MODÈLE DU CENTRE DE COMPÉTENCES

La littérature a indéniablement documenté les différentes facettes de l'open source et son application en entreprise. Ainsi la littérature a distingué les modèles d'affaires ouverts (Dahlander 2004 ; Jullien 2003), hybrides (Bonaccorsi et al. 2006 ; Välimäki 2003) et fermés (Benkeltoum 2011c ; Lisein et al. 2009) ; décrit les modèles d'intervention dans les communautés (Dahlander et Magnusson 2005 ; 2008) et d'allocation de ressources pour tirer profit de l'open source (Grand et al. 2004) ; dressé les avantages et inconvénients de ces technologies en termes de structure (Capra et al. 2011), modularité (MacCormack et al. 2006) ou encore de réactivité (Paulson et al. 2004) sans pour autant proposer de modèle permettant la mise en place d'une stratégie IT basée sur l'open source dans les entreprises concevant des systèmes critiques.

Grâce à sa transversalité, le cas Thales permet de voir la manière dont s'est structurée une stratégie IT intégrant les technologies open source. Sur la base de ce matériau empirique et en mobilisant l'abondante littérature, cet article propose un modèle pour gérer le processus d'adoption et d'intégration de l'open source dans la stratégie IT (tableau 2) pour les entreprises impliquées sur les marchés des systèmes critiques.

²⁸ Directeur de la Recherche et de l'Innovation, Thales D3S, entretien téléphonique avec l'auteur.

Tableau 2 : le modèle du centre de compétences

ÉTAPE	ACTIONS	SOURCES PERTINENTES
1. IDENTIFIER ET LEVER LES FREINS		
a. IDENTIFIER ET LEVER LES FREINS JURIDIQUES	- Étudier et comprendre les licences open source - Vérifier la politique de brevets	(Capobianco 2006 ; Edwards 2005)
b. IDENTIFIER ET LEVER LES FREINS TECHNIQUES	- Identifier les dépendances vis-à-vis de bibliothèques non libres - Maitriser les modèles de développement open source	(Agerfalk et Fitzgerald 2008 ; Fitzgerald 2006)
2. DÉFINIR LE MODÈLE D’AFFAIRES		
a. CRÉER ET CAPTURER DE LA VALEUR	- Expliciter les moyens de création de valeur - Vérifier la pérennité du modèle pour capturer la valeur	(Benkeltoum 2011c ; Bonaccorsi et al. 2006 ; Lisein et al. 2009)
b. DÉFINIR LES ÉLÉMENTS PARTAGÉS ET SECRETS	- Définir les composants partagés - Protéger les éléments assurant l’avantage concurrentiel	(Benkeltoum 2011a ; Henkel 2006 ; Lisein et al. 2009)
3. DÉFINIR LA STRATÉGIE IT		
a. IDENTIFIER DES COMPOSANTS STABLES ET MATURES	- Évaluer les composants open source à intégrer	(del Bianco et al. 2009 ; Miralles et al. 2006 ; Taibi et al. 2007)
b. INTÉGRER LES COMMUNAUTÉS ET CONSORTIUMS	- Identifier et intégrer les communautés et consortiums	(Benkeltoum 2011a ; Dahlander et Magnusson 2005 ; 2008 ; Stuermer 2009)

6. CONCLUSIONS ET IMPLICATIONS

Cette recherche contribue à la littérature sur l’utilisation de l’open source dans la formulation de la stratégie IT pour les entreprises concevant des systèmes critiques. Elle décrit plus spécifiquement la manière dont une entreprise agissant dans des domaines exigeants en matière de qualité et de sécurité, a changé d’optique de développement en intégrant des technologies libres et en partageant des technologies non critiques pour sa compétitivité. Un partenaire de Thales souligne que « *l’open-source a un pouvoir inné de transformer les organisations même les plus fermées et les plus policées : une fois qu’il y a fait son entrée, les choses changent, et ça échappe à tout contrôle (la remarque vaut pour Thalès... comme pour la Chine entière*²⁹). » La stratégie de Thales prouve bien que la sécurité n’est pas forcément une question d’utilisation de logiciels de type *boîte noire* dont on ne connaît pas le contenu. Comme l’affirme un spécialiste de Thales, « *la plupart des logiciels propriétaires contiennent des failles de sécurité volontaires ou non*³⁰. » De même, l’engouement de l’open source dans la conception de systèmes critiques est non seulement explicable par des aspects purement économiques mais aussi par des éléments qui étaient au départ la principale critique des logiciels open source.

La place et le rôle du centre de compétences est réellement le point névralgique de la stratégie de Thales. Ce cas démontre aussi que l’adoption de l’open source au sein d’un grand groupe nécessite la création de compétences spécifiques (Cohen et Levinthal 1990) afin d’être capable de tirer profit de connaissances externes (Chesbrough 2003 ; Le Texier et Versailles 2009). De même, la participation du centre de compétences dans les différentes communautés permet à l’entreprise de

²⁹ Fondateur d’Experlog, entretien en ligne avec l’auteur.

³⁰ Open Source Architect n°3, Thales D3S, Rencontres Mondiales du Logiciel Libre, 2007.

développer des connaissances pointues et une réelle expertise. La littérature a également peu documenté comment l'open source pouvait être mobilisé dans le cadre d'une politique générale d'indépendance technologique.

Enfin, cet article propose un modèle d'intégration et de déploiement de l'open source. Le modèle du centre de compétence est particulièrement riche puisqu'il aborde les freins (technologiques et juridiques), le modèle d'affaire (création et capture de la valeur) et la stratégie IT (identification des composants et implication dans les communautés). Toutefois, la validité et la robustesse de ce modèle nécessitent d'être mises à l'épreuve par le biais d'une étude de cas multiple (Eisenhardt et Graebner 2007 ; Leonard-Barton 1990).

Bibliographie

- Adam, Frédéric, Joseph Feller, et Brian Fitzgerald. (2003) "Logiciels Libres : Implications pour les Organisations." *Système d'Information et Management*, Vol. 8, n°1: p. 3-9.
- Agerfalk, Pär J. et Brian Fitzgerald. (2008) "Outsourcing to an Unknown Workforce: Exploring Opensourcing as a Global Sourcing Strategy." *MIS Quarterly*, Vol. 32, n°2: p. 385-409.
- Ajila, Samuel et Di Wu. (2007) "Empirical study of the effects of open source adoption on software development economics." *The Journal of Systems and Software*, Vol. 80, n°9: p. 1517-1529.
- Austin, Robert et Patrick Larkey. (1992) "The Unintended Consequences of Micromanagement: The Case of Procuring Mission Critical Computer Resources." *Policy Sciences*, Vol. 25, n°1: p. 3-28.
- Benkeltoum, Nordine. (2009) "Les régimes de l'open source : solidarité, innovation et modèles d'affaires, Thèse de doctorat en sciences de gestion." Centre de Gestion Scientifique, Mines ParisTech, Paris.
- . (2011a) *Gérer et comprendre l'open source*. Paris: Presses des Mines.
- . (2011b) "Open source: sortir des idées reçues" *ParisTech Review*, 19 octobre
<http://www.paristechreview.com/2011/10/19/open-source-idees-recues/>
- . (2011c) "Regards sur les stratégies de détournement dans l'industrie open source." *Vie et sciences économiques*, Vol., n°187: p. 72-91.
- Bitzer, Jürgen et Philip J. H. Schröder. (2005) "Bug-fixing and code-writing: the private provision of open source software." *Information Economics and Policy*, Vol. 17, n°3: p. 389-406.
- Bloom, P. C. et Q. B. Chung. (2001) "Lessons learned from developing a mission-critical expert system with multiple experts through rapid prototyping." *Expert Systems with Applications*, Vol. 20, n°2: p. 217-227.
- Bonaccorsi, Andrea, Silvia Giannangeli, et Cristina Rossi. (2006) "Entry Strategies Under Competing Standards: Hybrid Business Models in the Open Source Software Industry." *Management Science*, Vol. 52, n°7: p. 1085-1098.
- Bonaccorsi, Andrea et Cristina Rossi. (2003) "Comparing motivations of individual programmers and firms to take part in the open source movement - From community to business." Pisa: Laboratory of Economics and Management, Sant'Anna School of Advanced Studies.
- Broersma, Matthew. (2007) "IDC: Open-Source Market to Be Worth \$5.8B by 2011" *CIO*,
http://www.cio.com/article/116201/IDC_Open_Source_Market_to_Be_Worth_5.8B_by_2011
- Capiluppi, Andrea, Cornelia Boldyreff, et Klaas-Jan Stol. (2011) "Successful Reuse of Software Components: A Report from the Open Source Perspective Open Source Systems: Grounding Research." Pp. 159-176, vol. 365, *IFIP Advances in Information and Communication Technology*, edited by S. Hissam, B. Russo, M. de Mendonça Neto, and F. Kon: Springer Boston.
- Capobianco, Fabrizio. (2006) "The Honest Public License"
<http://www.funambol.com/blog/capo/2006/08/honest-public-license.html>
- Capra, Eugenio, Chiara Francalanci, Francesco Merlo, et Cristina Rossi-Lamastra. (2011) "Firms' involvement in Open Source projects: A trade-off between software structural quality and popularity." *Journal of Systems and Software*, Vol. 84, n°1: p. 144-161.
- Chen, Daoyi, Shahriar Shams, César Carmona-Moreno, et Andrea Leone. (2010) "Assessment of open source GIS software for water resources management in developing countries." *Journal of Hydro-environment Research*, Vol. 4, n°3: p. 253-264.
- Chesbrough, Henry. (2003) *Open Innovation: The New Imperative for Creating and Profiting from Technology*: Harvard Business School Press.
- . (2007) "Business model innovation: it's not just about technology anymore." *Strategy & Leadership*, Vol. 35, n°6: p. 12-17.

- Christensen, Clayton M. (2006) "The Ongoing Process of Building a Theory of Disruption." *Journal of Product Innovation Management*, Vol. 23: p. 39-55.
- CIGREF. (2011) "Maturité et gouvernance de l'Open Source - La vision des grandes entreprises." CIGREF, Paris.
- Cohen, Wesley M et Daniel A Levinthal. (1990) "Absorptive Capacity: A New Perspective on Learning and Innovation." *Administrative Science Quarterly*, Vol. 35, n°1: p. 128-152.
- Dahlander, Linus. (2004) "Appropriating the Commons: Firms in Open Source Software." Pp. 1-23 Chalmers University of Technology.
- . (2005) "Appropriation and appropriability in open source software." *International Journal of Innovation Management*, Vol. 9, n°3: p. 259-285.
- Dahlander, Linus et Mats Magnusson. (2005) "Relationships between open source software companies and communities: Observations from Nordic firms." *Research Policy*, Vol. 34, n°4: p. 481-493.
- . (2008) "How do Firms Make Use of Open Source Communities?" *Long Range Planning*, Vol., n°41: p. 629-649.
- Dahlander, Linus et Martin W. Wallin. (2006) "A man on the inside: Unlocking Communities as complementary assets." *Research Policy*, Vol. 35, n°7: p. 1243-1259.
- del Bianco, Vieri , Luigi Lavazza, Sandro Morasca, et Davide Taibi. (2009) "Quality of Open Source Software: The QualiPSO Trustworthiness Model." Pp. 199-212 in *Open Source Ecosystems: Diverse Communities Interacting*, 13th International Conference on Open Source Systems, IFIP, edited by C. Boldyreff, K. Crowston, B. Lundell, and A. I. Wasserman. Skövde, Sweden: Springer.
- Deshpande, Amit et Dirk Riehle. (2008) "The Total Growth of Open Source " in *Proceedings of the Fourth Conference on Open Source Systems*: Springer Verlag.
- Edwards, Kasper. (2005) "An economic perspective on software licenses—open source, maintainers and user-developers." *Telematics and Informatics*, Vol., n°22: p. 111-133.
- Eisenhardt, Kathleen M. (1989) "Building Theories from Case Study Research." *The Academy of Management Review*, Vol. 14, n°4: p. 532-550.
- Eisenhardt, Kathleen M. et Melissa E. Graebner. (2007) "Theory building from cases: opportunities and challenges." *Academy of Management Journal*, Vol. 50, n°25-32: p.
- Elie, François. (2010) "Les modèles économiques du logiciel libre." in *Solutions Linux*. Paris, France.
- Ferrell, Thomas K. et Uma D. Ferrell. (2001) "RTCA DO-178B/EUROCAE ED-12B." in *The Avionics Handbook*, edited by C. r. Spitzer: CRC Press LLC.
- Fitz-Gerald, Stuart. (2010) "K. Vadera, B. Gandhi, Open Source Technology (2009) University Science Press, Laxmi Publications, New Delhi, pp. 173." *International Journal of Information Management*, Vol. 30, n°4: p. 374.
- Fitzgerald, B. (1997) "The use of systems development methodologies in practice: a field study." *Information Systems Journal*, Vol. 7, n°3: p. 201-212.
- Fitzgerald, Brian. (2006) "The Transformation of Open Source Software." *MIS Quarterly*, Vol. 30, n°3: p. 587-598.
- Fuggetta, Alfonso. (2003) "Controversy Corner: Open source software an evaluation." *The Journal of Systems and Software*, Vol., n°66: p. 77-90.
- Gartner Inc. (2011) "Gartner Survey Reveals More than Half of Respondents Have Adopted Open-Source Software Solutions as Part of IT Strategy" February 8
<http://www.gartner.com/it/page.jsp?id=1541414>
- Gary, Kevin, Andinet Enquobahrie, Luis Ibanez, Patrick Cheng, Ziv Yaniv, Kevin Cleary, Shylaja Kokoori, Benjamin Muffih, et John Heidenreich. (2011) "Agile methods for open source safety-critical software." *Software: Practice and Experience*, Vol. 41, n°9: p. 945-962.
- Geeknet Inc. (2011) "SourceForge is dedicated to making open source projects successful"
<http://sourceforge.net/about>
- Geeknet Inc. (2009) "What is SourceForge.net?"

<http://sourceforge.net/apps/trac/sourceforge/wiki/What%20is%20SourceForge.net>

- Ghosh, Rishab A., Ruediger Glott, Bernhard Krieger, et Gregorio Robles. (2002) "Free/Libre and Open Source Software: Survey and Study." International Institute of Infonomics, University of Maastricht.
- Glaser, Barney G. et Anselm L. Strauss. (1967) The discovery of grounded theory: strategies for qualitative research: Transaction Publishers.
- Grand, Simon, Georg Von Krogh, Dorothy Leonard, et Walter Swap. (2004) "Resource Allocation Beyond Firm Boundaries: A Multi-Level Model for Open Source Innovation." Long Range Planning, Vol. 37: p. 591-610.
- Haefliger, Stefan, Georg von Krogh, et Sebastian Spaeth. (2008) "Code Reuse in Open Source Software." Management Science, Vol. 54, n°1: p. 180-193
- Hansen, Marit, Kristian Köhntopp, et Andreas Pfitzmann. (2002) "The Open Source approach — opportunities and limitations with respect to security and privacy." Computers & Security, Vol. 21, n°5: p. 461-471.
- Henkel, Joachim. (2006) "Selective revealing in open innovation processes: The case of embedded Linux." Research Policy, Vol. 35, n°7: p. 953-969.
- IDC. (2009) "Open Source Software Market Accelerated by Economy and Increased Acceptance From Enterprise Buyers, IDC Finds" Business Wire, 29 July
<http://www.businesswire.com/news/home/20090729005107/en/Open-Source-Software-Market-Accelerated-Economy-Increased>
- Jullien, Nicolas. (2003) "Le marché francophone du logiciel libre." Système d'Information et Management, Vol. 8, n°1: p. 77-100.
- Kroah-Hartman, Greg, Jonathan Corbet, et Amanda McPherson. (2009) "Linux Kernel Development How Fast it is Going, Who is Doing It, What They are Doing, and Who is Sponsoring It: An August 2009 Update"
<http://www.linuxfoundation.org/sites/main/files/publications/howwriteslinux.pdf>
- Lakhani, Karim et Eric von Hippel. (2003) "How open source software works: "free" user-to-user assistance." Research Policy, Vol. 32, n°6: p. 923-943.
- Lawton, George (2002) "Open Source Security: Opportunity or Oxymoron?", pp. 18-21.
- Le Boulout, Eric. (2011) "Une importante faille de sécurité découverte dans les iPhone" 01Net, 8 octobre
<http://www.01net.com/editorial/546126/une-importante-faille-de-securite-decouverte-dans-les-iphone/>
- Le Texier, Thomas et David W. Versailles. (2009) "Open Source Software Governance Serving Technological Agility: The Case of Open Source Software within the DoD." International Journal of Open Source Software and Processes (IJOSSP), Vol. 1, n°2: p. 14-27.
- Lee, Sang-Yong Tom, Hee-Woong Kim, et Sumeet Gupta. (2009) "Measuring open source software success." Omega, Vol. 37, n°2: p. 426-438.
- Leonard-Barton, Dorothy. (1990) "A Dual Methodology for Case Studies: Synergistic Use of a Longitudinal Single Site with Replicated Multiple Sites." Organization Science, Vol. 1, n°3: p. 248-266.
- Lerner, Josh et Jean Tirole. (2001) "The open source movement: Key research questions." European Economic Review, Vol. 45: p. 819-826.
- Lindman, Juho, Matti Rossi, et Anna Puustell. (2011) "Matching Open Source Software Licenses with Corresponding Business Models." IEEE Software, Vol. 28, n°4: p. 31-35.
- Lisein, Olivier, François Pichault, et James Desmecht. (2009) "Les business models des sociétés de services actives dans le secteur Open Source." Systèmes d'Information et Management, Vol. 14, n°2: p. 7-38.
- Lundell, Björn, Brian Lings, et Edvin Lindqvist. (2010) "Open source in Swedish companies: where are we?" Information Systems Journal, Vol. 20, n°6: p. 519-535.

- MacCormack, Alan, John Rusnak, et Carliss Y. Baldwin. (2006) "Exploring the Structure of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code." *Management Science*, Vol. 52, n°7: p. 1015-1030.
- MARKESS International. (2009) "L'open source en France : un rôle croissant dans les organisations et dans des domaines clés pour l'innovation" http://markess-blog.typepad.fr/pressrelease/PRSS09K_OpenSource_0909_FINAL.pdf
- Meissonier, Régis, Isabelle Bourdon, Emmanuel Houze, Serge Amabile, et Stéphane Boudrandi. (2010) "Comprendre les motivations des développeurs de l'open source à partir de leur participation : une étude empirique sur une question de recherche inversée." *Système d'Information et Management*, Vol. 15, n°2: p.
- Min, Byoung Joon et Joong Sup Choi. (2004) "An approach to intrusion tolerance for mission-critical services using adaptability and diverse replication." *Future Generation Computer Systems*, Vol. 20, n°2: p. 303-313.
- Miralles, Francesc, Sandra Sieber, et Josep Valor. (2006) "An Exploratory Framework for Assessing Open Source Software Adoption." *Système d'Information et Management*, Vol. 11, n°1: p. 85-103.
- Muselli, Laure. (2008) "Le rôle des licences dans les modèles économiques des éditeurs de logiciels open source." *Revue Française de Gestion*, Vol. 1, n°181: p. 199-214.
- Norris, Jeffrey S. (2004) "Mission-Critical Development with Open Source Software: Lessons Learned." *IEEE Software*, Vol. 21, n°1: p. 42-49.
- Onetti, Antonio et Fabrizio Capobianco. (2005) "Open source and business model innovation: The Funambol case." in *Proceedings of the first International Conference on Open source Systems*. Genova.
- PAC. (2010) "Le logiciel libre ne connaît pas la crise" 11 mars https://www.pac-online.com/pac/pac/live/pac_france/global/presse/communiquede/presse/index.html?lenya_usecase=show-rapport&document=pac_sitsi_reports/press_release/fr_pr_10_logiciel_libre&xsl=press_release
- Paulson, James W., Giancarlo Succi, et Armin Eberlein. (2004) "An Empirical Study of Open-Source and Closed-Source Software Products." *IEEE Trans. Softw. Eng.*, Vol. 30, n°4: p. 246-256.
- Pettigrew, Andrew M. (1990) "Longitudinal Field Research on Change: Theory and Practice." *Organization Science*, Vol. 1, n°3: p. 267-292.
- Rannou, Hervé et Maurice Ronai. (2003) "Etude sur l'industrie du logiciel, Partie 2 - Rapport." Conseil Stratégique des Technologies de l'Information.
- Raymond, Eric S. (1999) *The Cathedral and the Bazaar*. Sebastopol, CA: O'Reilly & Associates.
- Rossi, Cristina. (2009) "Software innovativeness. A comparison between proprietary and Free/Open Source solutions offered by Italian SMEs." *R&D Management*, Vol. 39, n°2: p.
- Sharma, Srinarayan, Vijayan Sugumaran, et Balaji Rajagopalan. (2002) "A Framework for Creating hybrid- open source software Communities." *Information Systems Journal*, Vol. 12, n°1: p. 7-26.
- Spaeth, Sebastian, Georg von Krogh, Matthias Stuermer, et Stefan Haefliger. (2007) "Lightweight Reuse In Open Source Software." in *European Academy of Management (EURAM)*. HEC Paris, Jouy-en-Josas.
- Spinellis, Diomidis, Georgios Gousios, Vassilios Karakoidas, Panagiotis Louridas, Paul J. Adams, Ioannis Samoladas, et Ioannis Stamelos. (2009) "Evaluating the Quality of Open Source Software." *Electronic Notes in Theoretical Computer Science*, Vol. 233: p. 5-28.
- Stuermer, Matthias. (2009) "How Firms Make Friends: Communities in Private-Collective Innovation, Doctoral Dissertation." Chair of Strategic Management and Innovation, Department of Management, Technology, and Economics, ETH Zürich, Zürich.

- Stuermer, Matthias, Spaeth Sebastian, et Georg von Krogh. (2009) "Extending private-collective innovation: a case study." *R&D Management*, Vol. 39, n°2: p. 2009.
- Taibi, Davide, Luigi Lavazza, et Sandro Morasca. (2007) "OpenBQR: a framework for the assessment of OSS." Pp. 173-186 in *Open Source Development, Adoption and Innovation*, vol. 234, Open Source Development, Adoption and Innovation, edited by IFIP. Boston: Springer.
- Välimäki, Mikko. (2003) "Dual Licensing in Open Source Software Industry." *Systemes d'Information et Management*, Vol. 8, n°1: p. 63-75.
- von Hippel, Eric. (2007) "Horizontal innovation networks—by and for users." *Industrial and Corporate Change*, Vol. 16, n°2: p. 293-315.
- von Hippel, Eric et Georg von Krogh. (2003) "Open Source Software and the “Private-Collective” Innovation Model: Issues for Organization Science." *Organization Science*, Vol. 14, n°2: p. 209-223.
- von Krogh, Georg et Sebastian Spaeth. (2007) "The open source software phenomenon: Characteristics that promote research." *Journal of Strategic Information Systems*, Vol. 16: p. 236-253.
- West, Joel. (2003) "How open is open enough? Melding proprietary and open source platform strategies." *Research Policy*, Vol., n°32: p. 1259-1285.
- ZDNet.fr. (2011) "Accord Google-Mozilla : près d'un milliard de dollars sur 3 ans" ZDNet.fr, 23 décembre
<http://www.zdnet.fr/actualites/accord-google-mozilla-pres-d-un-milliard-de-dollars-sur-3-ans-39766805.htm>