

OPEN SOURCE, AN ALTERNATIVE TO MAJOR IT PROVIDERS

Taking the plunge into open source



Open source, an alternative to major IT providers

Taking the plunge into open source

December 2018

Cigref is a network of major French companies and public administrations set up in order to develop its members' ability to acquire and master digital technology. It is a key player and federating body in the digital society, thanks to its high-quality thinking and the extent to which it represents its members.

Created in 1970, Cigref is a not-for-profit body in accordance with the French 1901 Law of Associations. It counts among its members some 150 major French corporations and public administrations across all business sectors. It is overseen by 15 board members who are elected by the General Assembly. Its day-to-day work is carried out by a team of ten permanent members of staff.

 Intellectual copyright

All Cigref publications are made freely available to the general public but remain protected by the applicable laws on intellectual property.

OVERVIEW

With digital transformation, the scope of action of the information systems (IS) of companies and public organisations is constantly expanding. They all share the same objective of controlling their IT budgets. Companies are relying on their providers to gain room for manoeuvre on these budgets. In a context where business innovation is based to a very large extent on digital services and tools, the **relationship of companies with their providers is crucial in terms of the challenges of competitiveness, agility and performance.**

Today, the world market for software licences and digital services is concentrated in the hands of a few international, mainly American, major players. Their goal is to increase their turnover which has become unsustainable for companies and public organisations. In addition, they are increasingly refusing to bear the costs of solutions that have become commodities, that have only very limited innovative value, and that make them dependent on a hegemonic provider.

Companies and public organisations are therefore looking for alternatives to loosen the financial stranglehold of providers and gain room for manoeuvre in their negotiations with software editors. Free software, known as “open source”, is one alternative. Furthermore, it gives companies **strategic advantages and action levers on their competitiveness.**

Opting for open source obviously incurs costs. Indeed, before adding these solutions to the scope of possible alternatives, companies must acculturate their employees, explain the specificities of this ecosystem to them and establish governance in the selection of software. This also requires the in-house or outsourced development of skills to understand the roles of players, along with specialist technical resources if the company wants to develop certain specificities in order to meet a specific need.

Open source challenges risk management modes: major IT providers constitute guarantee brands, an insurance policy, and a guarantee of security for the Executive Committee. However, **solutions exist in free software for each risk identified.**

In addition, open source solutions meet today’s challenges: time to market, agility at scale of the company and open innovation. They facilitate the collaboration and pooling of software developments. Free software also provides continuity, security, flexibility with open formats, a wider testing coverage in multiple use cases within large communities and sometimes possible influence on the roadmap (development plan). Open source ultimately offers leverage to improve the skills and attractiveness of talented individuals. Open source sometimes even constitutes the condition for attracting young generations.

These are all reasons why some companies, strongly supported by their Executive Committee, apply a strategy that **explores all solutions**, and in particular, open source solutions. Some have even adopted the “open source first” strategy.

Open source, an alternative to major IT providers

Taking the plunge into open source

Main recommendations for taking the plunge into open source:

- Acculturate employees with the open source ecosystem (key characteristics of this ecosystem, advantages and disadvantages of open source software, and risk management)
- Define criteria for analysing open source software, shared by the entire company and that take into account long-term risk taking; for example, intellectual property, type of licence, user and developer communities, etc.
- Establish governance within the company for selecting software;
- Set up a software selection process. In other words, with all possible software solutions, select the best compromise between:
 - business need. Companies recommend defining the operational need that the software must meet;
 - price;
 - possible support.

This selection takes into account the level of dependence on proprietary software solutions of major suppliers and the possibility of pooling developments.
- Similarly, set up a software or support selection process.

ACKNOWLEDGEMENTS

We would like to thank Stéphane Rousseau, CIO of Eiffage, who steered this study as well as all those who participated and contributed to this Cigref working group:

Oussama ABOU - ALLIANZ
Aimery ASSIRE - EDF
Latifa AYEB - EDF
Noëlle BAILLON BACHOC - ORANGE
Nicolas BAILLY - SAINT-GOBAIN
Alain BARBAY - MINISTERE DE L'INTERIEUR
Cyril BARTOLO - LAGARDÈRE
Rony BASSET - VALLOUREC
Jean-Charles BASTOUL - MINISTERE DE L'INTERIEUR
Vincent BELROSE - LVMH
Jérôme BOIN - MAIF
Claire BONNETAIN - EIFFAGE
Didier BOUFFARD - AXA GROUP
Joël BOURHIS - GRDF
Loïc BOURNON - SAFRAN
Marc BOUSSET - SAFRAN
Priscilla BOUVET - TOTAL
Valérie BREUT - DASSAULT AVIATION
Yvan BRUNEL - COVEA
Marie-Flore CABANE - MINISTERE DES ARMEES
Noël CAVALIÈRE - GROUPE PSA
Philippe CHASSAING - VEOLIA
Simon CLAVIER - SNCF
David COURTOIS - IDEMIA
Yves DANIEL - SAFRAN
Sophie DARRACQ - BANQUE DE FRANCE
David DARWISH - IDEMIA
Olivier DE BERNARDI - SOCIÉTÉ GÉNÉRALE
Philippe DEBRAIZE - MINISTERE DES ARMEES
Laurent DELISLE - THALES
Philippe DOUBLET - GROUPE POMONA
Olivier DU MERLE - AIR FRANCE KLM
Cédric DUFOUR - AGIRC ARRCO
Ewen DUGUE - GEODIS INTERSERVICES
Mélanie DUVEAU - AGIRC ARRCO
David ELOY - EUROPCAR
Fatima FAR - ESSILOR INTERNATIONAL
Philippe GALICHET - AP-HP
Julie GARANÇON - GROUPAMA
Patrick GERVAIS - EDF
Thierry GRISELAIN - CREDIT AGRICOLE SA
Christine GROSSETÊTE - AIR LIQUIDE
Marie-Pierre GUGLIELMI - PMU
Marie-Laure GUIDOT - GRDF
Coraline HAYRAUD - ARKEMA
Alain ISSARNI - CNAM
Yannick JOBARD - ORANGE
Diane JOLY - AIRBUS
Pascal LAFOND - COVEA
Michel LAMRANI - DASSAULT AVIATION
Raynald LASOTA - FRANCE TÉLÉVISIONS
Pierre LE BER - EDENRED
Cédric LE BERRE - GEODIS INTERSERVICES
Sylvie LE GALL - NAVAL GROUP
Hervé LE MEN - NEXITY
Patrick LEFEUVRE - EDF
Olivier MARCHAIS - GIP MDS
Catherine MAURICE - PLASTIC OMNIUM
Malika MEZAIER-POUPIN - CARREFOUR
Erwan MOYSAN - AXA GROUP
Ravi NADARADJANE - SCOR
François-Gaudéric OLIVE - EDF
Pierre PAN - ESSILOR INTERNATIONAL
Christian PATERSON - ORANGE
Chantal PEYRAT - DASSAULT AVIATION
Jean-Claude PIGEON - RTE
Béatrice POTIER - SAFRAN
Elizabeth PUGEAT - CREDIT AGRICOLE SA
Laurent QUÉRÉ - AVIVA
Anaïs ROMAND - AXA GROUP
Philippe ROUAUD - FRANCE TÉLÉVISIONS
Florence PERROT - CNAM
Christophe SARRE - EDF
Nathalie SENARD - VEOLIA
Frédéric SERET - SFR SI
Grégory SILVAIN - EURO DISNEY
Jean-François STRICHER - ENEDIS
Teodora TOMA - TOTAL
Axel TOUBERT - CAISSE DES DÉPÔTS
Hubert TOURNIER - RAMSAY GÉNÉRALE DE SANTÉ
Géraldine TRIBONDEAU - AXA GROUP

Cigref would also like to sincerely thank the following external people for their work and contributions to the studies:

Aimery ASSIRE - EDF
Simon CLAVIER - SNCF
Xavier GUIMARD - Gendarmerie
Alain ISSARNI - CNAM
Laurent JOUBERT - DINSIC
Etienne JULIOT - Obeo
Hervé LEMAITRE - Red Hat
Bruno POIRIER COUTANSAIS - Gendarmerie
Alexandre ZAPOLVSKY - Linagora

This document was written by Marine de SURY, Cigref Mission Officer, with help from Stéphane Rousseau.

COORDINATOR'S EDITORIAL

“Software sourcing has become a major challenge for the chief information officer, because not only do organisations differ, but it is also important to be more flexible and agile to meet business needs, and to guarantee the highest degree of independence for manoeuvre. However, the market concentration on major IT providers is significantly shifting the decision-making focus. Consequently, urgent action is needed to foster a dynamic alternative ecosystem that can be controlled by user companies. This ecosystem exists in infrastructure software solutions: this is the open source universe.

Compared with the traditional model for publishing software solutions, which shows the limitations of its relevancy, open source is better suited to the new codes of co-creation, collaborative production and open innovation.

Open source has become the go-to component of an IT strategy.”

Stéphane Rousseau, Working group coordinator
CIO Eiffage and Cigref administrator

TABLE OF CONTENTS

Introduction	1
1. Open source ecosystem: key characteristics	2
1.1. Licence types	2
1.2. Open source provider business models	4
1.3. Intellectual property	5
1.4. Details on the cost of deployment of open source software	6
1.4.1. Use of a licence	7
1.4.2. Development cost of future functionalities (pooled or not).....	7
1.4.3. Adaptations to integrate it into the information system.....	8
1.4.4. Free software professional support cost	8
1.4.5. Costs incurred not to be omitted.....	9
2. Advantages and disadvantages of open source	10
2.1. Disadvantages.....	10
2.2. Specificities in terms of skills.....	11
2.3. Strategic advantages	11
2.3.1. On the system level.....	12
2.3.2. On the room for manoeuvre level	13
2.4. Competitive advantages	14
2.4.1. Opening and collaborating.....	14
2.4.2. Pooling	15
3. Free software risk management	17
3.1. Free and proprietary software risk types	17
3.1.1. Ability to make	17
3.1.2. Operational risk	18
3.2. Solutions to the risks	18
3.2.1. No specific support	18
3.2.2. Selecting a commercial distribution	18
3.2.3. Acquisition of “editor” or third-party support	18
3.2.4. Option of no reinsurance policy	19
4. Starting the transition to open source	20
4.1. Acculturation of employees and managers	20
4.2. Software selection governance	20
4.3. Consultations and calls for tender	21
4.4. State example	22
5. Free software selection: analysis criteria and processes	23
5.1. Open source software selection process	23
5.2. Open source solution analysis criteria.....	24
5.3. Open source solution support selection process, excluding solutions with subscription	25
Conclusion	28

TABLE OF FIGURES

Figure 1: Possible evolution of the allocation of a licence to free software	3
Figure 2: Total cost of deployment of open source software	7
Figure 3: Open source software business models	25
Figure 4: Maintenance and support depending on the software.....	26

FOREWORD

In its 2018 best wishes, Cigref defined [seven resolutions](#) for the successful transition of digital technology in France and in Europe by 2020. Seven resolutions to make 2018 “THE” year for recovering our strategic autonomy in all digital dimensions, to promote economic growth, the competitiveness of companies and create jobs. One of the resolutions aims to create a regulatory environment, in terms of standardisation, competition and infrastructure, so that digital systems can be implemented and governed in companies and public administrations in an agile way and without excessive hindrance.

Challenges with software providers

Cigref assists its members in their relations with IT editors within its “provider relations” club and offers them a space to think, take action and share between peers, in particular with respect to the problems that these relationships may generate. Since the early 2000s, Cigref has decided to consider providers as partners and work with them to find solutions to the stumbling blocks that its member major corporations and organisations may come across.

The world market for software licences and digital services is concentrated in the hands of a few international, mainly American, major players. In a context where business innovation is mainly based on digital services and tools, **the relationship with providers is crucial and essential** in terms of the challenges of innovation, agility and performance, which determine their future competitiveness.

Budget objectives of companies, public organisations and providers

Major companies and public organisations all currently share the same objective of **controlling their IT budget**. They are optimising this budget as much as possible to keep to their roadmap. They are therefore relying on their providers in order to continue gaining room for manoeuvre.

However, the pricing strategy of major providers is creating tensions due the too many uncertainties that it generates with regard to the evolution of corporate IT budgets. Provider prices vary according to negotiation; metrics change and cause the invoice to increase, amplified by the digitalisation of business units and the growing use of digital services. Reaching the end of its software support contract, the company has the “choice” between migrating or taking out an extension. Consequently, companies feel that they are being forced into a situation of increasing dependence. The captivity of companies is intensified further with the development of SaaS (Software as a Service) solutions. In addition, indirect accesses multiply the costs of purchasing licences.

Lastly, a large number of software solutions are now commodities with a limited innovative value, and customers are increasingly unwilling to pay a high price for products considered to be shared and already paid for, and linked to a provider.

Introduction

Alternative solutions to major proprietary software providers already exist and are operational within companies. Indeed, players who are not as large or in a dominant position, typically medium-size companies, may offer the flexibility needed to balance the budgets of their customers. Companies know how to “source” these alternative suppliers according to their needs. This also includes *startups* that may also offer alternatives over time.

Open source solutions also constitute a way to reduce the budget for the information systems (IS) of companies and public organisations. Cigref’s members have been looking at open source over the past 10 years. Its latest report on the topic, “[Maturité et gouvernance de l’Open Source](#)” (Open Source maturity and governance), published in March 2011, studied open source governance, the migration of companies towards *open office* and the perception of open source maturity in major companies.

A rapid assessment carried out at the end of 2017 by the companies involved in the Cigref working group highlighted the common use of open source solutions, mainly within the infrastructure, and less in business applications. Free solutions are more or less used according to the company's legacy information system, its strategy or its relations with its historic providers.

However, free software offers many other possibilities than that of reducing the IT budget. Companies are **convinced of the strategic advantages as well as the action levers provided for improving their competitiveness**. The choice of open source solutions allows them to **develop skills and attract talent**. Lastly, **mastering the information system** that they are targeting is accessible through open source.

However, selecting an open source solution requires a clear understanding of the specificities of this ecosystem to avoid getting it completely wrong. The aim of this report, which is intended for IT department employees and more generally all employees, is to help companies take the plunge into open source. Indeed, most employees are affected as digital technology is now expanding into all sectors of activity.

The first part of this report explains the key characteristics of the open source ecosystem that need to be kept in mind when selecting solutions. The second part describes the disadvantages as well as the competitive and strategic advantages. The development of open source within companies challenges risk management. Indeed, free software risk management is different from major editor software risk management. This is discussed in the third chapter. We will then see that companies must anticipate the shift towards open source software and organise themselves accordingly. Lastly, the last chapter proposes elements, such as analysis criteria and processes, common in any company in order to help them select a software solution.

1. Open source ecosystem: key characteristics

The purpose of this part is to briefly remind you of the key elements about open source in order to make an informed decision. These notions are **essential**. Indeed, to avoid making the wrong decision, companies must fully understand the open source ecosystem beforehand.

Software is said to be open source when the source code is free and open access. This is known as *Free Open Source Software, or FOSS*.

Free software, Free Libre Open Source Software, *FLOSS*, provides an **ethical dimension on freedom**. There are four underlying types of freedom:

- the freedom to study the source code (access) and how the program works to adapt it to your needs;
- the freedom to change/improve the source code (access) and share it;
- the freedom to run the program;
- the freedom to redistribute and copy the code.

These 4 freedoms are sometimes combined with rules (constraints).

With the exception of public domain software, free software does not have fewer owners than the others, although the term **author** is more appropriate. Free software authors retain certain rights as a direct result of the type of free licence used.

1.1. Licence types

A contract entails **reciprocal obligations** with its co-contracting party, operator or user. A licence is a **unilateral authorisation**. In the licence, the copyright owner on a computer program defines the conditions in which this program can be used, distributed or changed. **Free software is distributed with licences, generally without any guarantees.**

To obtain the right to use the software, the holder of the rights must authorise this. The licence is the document in which the rights granted to the licensee are listed (installing the software, using it, making a backup copy). Often, the holder of the rights does not just grant the licence, but also adds requirements, such as the prohibition to study the software coding, and publish its performance measurements, etc.

In the case of open source software, various free licences summarise the freedoms and rules. They are divided into 3 main licence types:

- **Permissive**, also known as **copyfree** licences. There are no restrictions on redistribution;

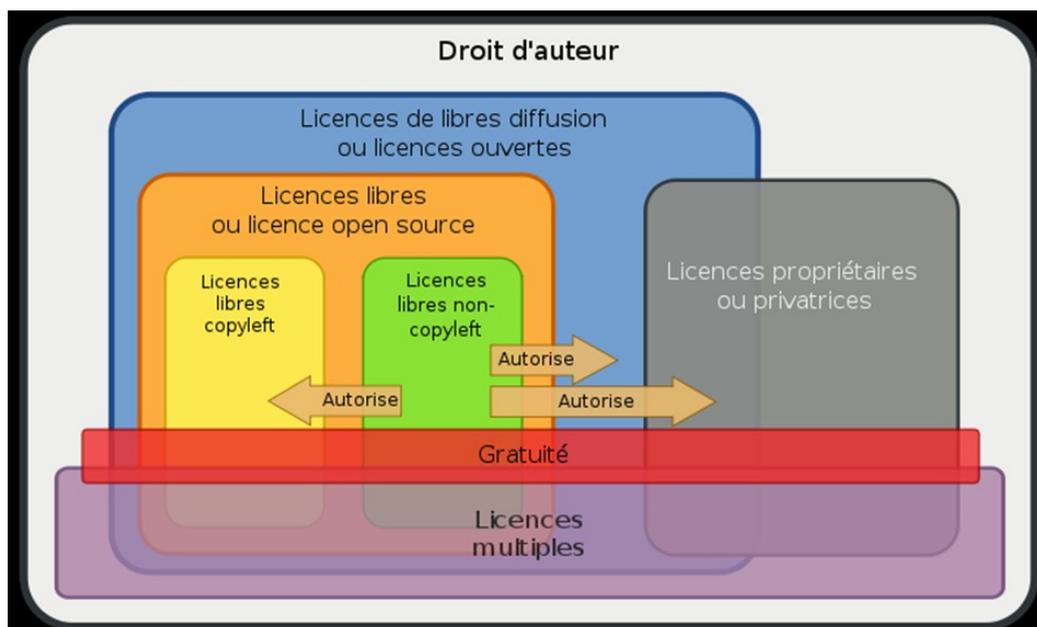
Open source, an alternative to major IT providers

Taking the plunge into open source

- **Non-permissive**, also known as **copyleft** (as opposed to copyright¹) licences: the author retains a copyright and redistribution must be done under the same licence. In principle, copyleft licences prevent the person using the code from taking ownership of the community effort without transferring improvements and corrections to it. **Contribution to the collective effort becomes a principle** and helps to maintain the dynamics of their developments. Grading exists with non-permissive licences, which ranges from the weak to strong *copyleft* licence. In fact, strong non-permissive licences, i.e. the most “restrictive”, require any project that reuses the source code of the free software to in turn become free software. Therefore, open source software cannot be used with this type of licence for a company that wants to make specific developments that it does not want to share...
- **Public domain licences**. The *copyright* has expired or there is no *copyright*.

To conclude, the company should analyse the free software licence type that it wants to use in order to identify the rights and possible obligations and associated requirements entailed.

The diagram below summarises how allocation of a licence to free software may evolve.



Source: Wikipedia

Figure 1: Possible evolution of the allocation of a licence to free software

Thus, free software in copyleft will remain in copyleft forever. It cannot change. Conversely, free software that is not copyleft may have a development, a fork², or be reworked by

¹ Copyright (literally right to copy), often indicated by the © symbol, includes all of the exclusive rights that a natural or legal person has on an original work. Therefore, it designates all of the applicable laws, in particular, in Commonwealth countries and in the United States; and that differs from the copyright applied in civil law countries (such as France or Belgium). Copyright is more economic-focused and grants a restricted moral right, whereas the author’s right provides a strong moral right based on the link between the author and their work. (Source: Wikipedia)

² The fork is the duplication of the source code of an existing project to create a new project (source: Obeo).

Open source, an alternative to major IT providers

Taking the plunge into open source

another provider. In addition, this provider has the right to (is authorised to) modify the copyleft licence or proprietary licence or quite simply open licence.

Two types of fork exist in open source:

- the **aggressive fork**: this is a competitor to the existing project that is based on the project in order to develop it in a new direction;
- the **friendly fork**: made famous through the use of Git technology, it is used to experiment the addition of a new functionality in view of integrating it into the initial project, if the result is relevant.

The marketing of free software is possible but the freedom to redistribute and copy the code prohibits any exclusivity. This is why free software does not allow direct remuneration of their authors. These authors must be restricted to the sale of services associated to the use of said software. **Therefore, business models involve selling know-how and expertise, rather than a usage right on the software.**

Once the type of licence has been identified, it is extremely important to understand the business model of open source editors, which is repeated below.

1.2. Open source provider business models

The open source editor is a for-profit business, like any other. The development of its software represents a cost that the publisher funds based on a return on investment. It is important to fully understand the business model of the open source software editor in order to avoid subsequent nasty surprises.

Four main *business model* categories are possible:

- **Dual licences**. The software developed is free under a strong *copyleft* licence and therefore cannot be integrated into a proprietary product or service. A second proprietary licence is then sold to users who want to free themselves of the *copyleft*.
- **Shared model**. Some generic needs require significant R&D efforts that are difficult to agree because the expected revenue can only be collected over the long term. Grouping and pooling the needs of several customers and therefore several funds may be a relevant solution for venturing into such developments. In this case, basic software is developed to encourage customers to co-develop modules associated to this basic software.
- **The packaged distribution or *open core* model**. The provider develops an open source technological kernel and a proprietary product on top of this kernel, or even develops two products on top of a common kernel, one in open source and the other with additional functionalities. In the latter case, the two software variants are often named "*community*" and "*enterprise*". The purpose of the open source version is to expand the community and distribute it as much as possible.
- **The indirect development or services model**. In this case, the publisher generates its revenue from professional services, maintenance and support associated to the software that it distributes.

Open source, an alternative to major IT providers

Taking the plunge into open source

To expand on this, publisher [modèles économiques](#)³ (business models) are described in detail in the “*livrets bleus*” published by the GTLL (Groupe Thématique du Logiciel libre de Systematic). The ⁴[livret bleu juridique](#) (legal blue book) outlines the legal framework for software, the specificities of free software and describes the rights and obligations if software is used or exploited under a free licence.

When using open source software, it is therefore important to understand its business model in order to be fully aware of the costs of the solution, whether this is open source software with a subscription in closed model, open model or free and free of charge.

Open source software with closed model subscription consist of a mandatory subscription, a *roadmap* imposed by the provider, and a community dominated by one player. Oracle's Java SE illustrates the change in business model of a software licence. At the beginning of 2018, Oracle dramatically changed the Java SE licence, marketing and support model (from version 9). By creating dual licensing, Oracle now releases a new Java open source “freemium” version for developers every 6 months (short cycle) and longer term (Long Term Support - LTS) commercial licences with integrated support for “enterprise” customers every 3 years.

In the case of open source software with open model subscription, the software and all of the modules are open source and open to everyone. However, the **contract imposes restrictions on free software freedoms**. The provider sells the component integration service on complex projects and certifies the industrialisation and stability (life cycle, security). It also provides maintenance and support.

In the open source business model, excluding open source subscription products, the user pays for the service associated to the exact need and not for the software annuity. It selects the provider of the support with the associated service.

1.3. Intellectual property

Before making any purchase, identifying where the intellectual property of the open source software is situated is crucial to avoid any subsequent surprises. Ownership rights may be hosted within a foundation, community or ultimately with an editor. Before continuing, here is a brief reminder of the open source ecosystem.

Open source functions within an ecosystem that includes the customers, the provider, the IT consultancy and the foundation.

- The **customers** fund the developments and support;
- The **provider** creates the software. Free software developers and users are usually independent with various motivations. They produce or use tools based on a production mode combining collaboration, mutual assistance, sharing and pooling;

³ Document available with url: https://systematic-paris-region.org/wp-content/uploads/2017/07/LivretBleu_ModelesEconomiques_GT-LogicielLibre_Systematic.pdfh

⁴ Document available with url: https://systematic-paris-region.org/wp-content/uploads/2017/07/LivretBleu_Juridique-2eEdition_GT-LogicielLibre_Systematic_Nov2016_web.pdf

Open source, an alternative to major IT providers

Taking the plunge into open source

- The **IT consultancy** integrates the software into the information system. IT consultancies may have to contribute to open source developments, but they do not publish them;
- The **foundation** hosts the code with neutrality and governance rules. Foundations are not in fact there to create the software that they host. They take care of coordinating the community and promoting software programs.

If ownership rights are hosted within a foundation, then the editor cannot change to proprietary mode or modify its business model.

The brand Sirius belongs to the Eclipse foundation. Consequently, Obeo, one of the main editors contributing to developments cannot change this solution into proprietary mode under the Sirius brand.

Obeo, Etienne Juliot

If the intellectual property is not hosted within a foundation, an investigation is required in order to identify the structure in which the research and development is focused. Indeed, if **R&D and industrial property are located** in the same structure, then there is a risk of isolation. In other words, the entity comprising the R&D, strengthened with industrial property, may decide at any time to modify its business model.

Lastly, the second point to be considered is identifying who **funds the developments** of an open source software program. Indeed, the financiers actually control the brand and influence the strategic choices.

Obeo is the main contributor of IS Designer, however, the Ministère des Armées (French Ministry of the Armed Forces) funds most of it. If another company wants to use this project and suggest developments, it must negotiate with the Ministry either directly or *via* the sub-contractor.

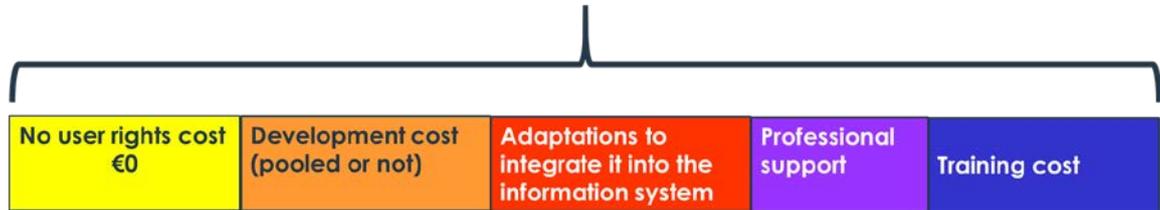
Obeo, Etienne Juliot

The project may become a community project if several companies contribute to it *via* one or more sub-contractors. Therefore, you switch from a centralised framework to a community project.

1.4. Details on the cost of deployment of open source software

The total cost of deployment of open source software includes the following elements:
Total = no user rights cost (€0) + development cost (pooled or not) + adaptations to integrate it into the information system + professional support + training cost.

Total cost of deployment of open source software



Source: Cigref

Figure 2: Total cost of deployment of open source software

1.4.1. Use of a licence

The purchasing cost of open source software is zero. Its development has already been paid for by others.

1.4.2. Development cost of future functionalities (pooled or not)

There are two possibilities for developing future functionalities that the company is interested in. First situation, the company is the only party interested in these developments: it funds them in full. Second situation, several companies are interested: so they choose to share the development cost of a common core.

When a company integrates developments into free software for its own needs, it creates a new development branch. Whenever the common software version is updated, it must then re-integrate its own codes into the shared source code, unless it manages to convince the community to integrate its contributions into the community development branch. If they are integrated, then the integration and maintenance costs will be shared: the company will thus reduce its operational costs whenever a new version is installed.

This is why a company (co-)funding a development will try and make sure that it is accepted by the community. This will avoid the cost of a *fork*. For this, the company must check that its contributions are **acceptable** and accepted by the community. Firstly, it will make sure that the community accepts these transfers. For this, it will analyse the type of licence selected for the shared code and the ecosystem of the contributors. Indeed, if the community mainly includes an editor also offering “enterprise” solutions, it is highly likely that it will refuse any other contribution. The company will then make sure that its contributions are in line with the community’s expectations, i.e. developments do not affect the native, original code; coherence with the roadmap; and documentation code quality, etc.

1.4.3. Adaptations to integrate it into the information system

This includes the **integration and specific settings**. Due to their complexity, free software integration costs are sometimes more expensive than those of provider solutions. Indeed, open source products are more complicated to integrate when they each concern limited functionalities. To obtain global coverage, distributions need to be used which, when they are community distributions, do not always integrate successive version upgrade and deployment facilities.

When developments for integration into the information system are accepted by the community, the entire configuration and customisation does not need to be redefined on each update. This helps control operational costs.

The CNAM is assisted with the integration of open source solutions into the information system, with their upgrades or with the development of new functionalities. The contract clearly states that any development must be transferred to the community.

CNAM, Alain Issarni

1.4.4. Free software professional support cost

The company has the **possibility of selecting the provider** who will provide community model open source software support and maintenance. This is not the case of open source software with subscription, which requires support and maintenance, or of proprietary software.

Most of the time, free software guarantees are identical and sometimes more important than those of major providers. As a reminder, a major editor does not undertake any obligation to provide results with regard to bug resolution. Depending on the contract, the scope for free software includes support 7 days a week, corrective maintenance, the guarantee of correct operation, and solution compliance. The company chooses the level of support based on the criticality of the service or application. The support options are described in paragraph 5.3 - Process for selecting open source solution support excluding subscription solutions.

When software is developed as open source by a company and is accepted by the community, the company then does not have to pay the guarantee. In fact, in this case, the guarantee is covered by the community.

The CNAM has made the strategic choice of "*Open First*". To assist this direction, it took out a fixed-price support contract with a service provider for all of its open software used (up to a maximum of 100 software and 3 versions per software chosen freely by the CNAM and whose list can change freely according to its needs). Therefore, it benefits from a "traditional provider mode" support contract with a **single contact person for all of its free software used**. A support contract whose assistance and correction conditions are defined by the CNAM based on the criticality of its applications. This global and fixed-

Open source, an alternative to major IT providers

Taking the plunge into open source

price contract also offers agility so that support can be provided for software that is still in the assessment phase.

CNAM, Alain Issarni

Companies recommend specifying in the contract with the sub-contractor and service provider that any open source development must be transferred to the community. This is also a way of contributing.

1.4.5. Costs incurred not to be omitted

Changing to open source solutions sometimes incurs hidden costs that need to be kept in mind. As is the case for many software solutions, investment in in-house or external skills is necessary. Some skills are rare and therefore expensive.

The company must expect an additional cost when it has to change its practices, its organisation for adopting collaborative work, sharing developments or even venturing into the make.

Another cost may arise when the company wants to influence, or even define, the roadmap (coordination/contribution) by investing in the communities.

The overall benefit of an open source strategy is difficult to quantify in general because each case is different. Furthermore, stopping a contract generates savings. However, choosing a free software solution requires new investments (skills, communities) and contracts (support *via* an IT consultancy).

2. Advantages and disadvantages of open source

Communication and information systems are **tools at the service of business units**. They constitute a **driver of performance** for the teams. The information system must offer the possibility of being open to customers, partners, providers and more generally all of the company's ecosystems. Open source software rises to this open challenge and offers a certain number of strategic advantages and catalysts for improving the company's competitiveness. However, open source also has a few disadvantages that should not be ignored. Lastly, the skills generated by open source constitute a plus and a minus within the organisation. This is why, this topic is covered in the same sub-chapter.

2.1. Disadvantages

The range of services related to open source software does not cover all of a provider's services, except if it is a subscription or proprietary model based on open source ... Consequently, the company must adapt accordingly.

Another disadvantage is the **legal jungle** outlining it: you need to be at ease with the ecosystem's licences, the various possible business models and fully understand where the intellectual property is situated to be aware of the risks associated with free software. Ultimately, you should be fully aware of the obligations associated with free software.

Understanding open source software roadmaps, excluding subscription models, is also not so easy.

As seen above, the companies that are involved in the Cigref working group mainly use open source solutions in the infrastructure, and less in business applications. Indeed, open source solutions exist and are implemented in the lower layers. In particular, they are included in the solutions deployed within the framework of *Cloud* offers. But there are fewer open source alternatives the higher up we go. In fact, open source offers on business information systems are less mature and less consolidated, even if since 2015 business software packages have started to emerge. Their returns on investment are lower than on technical layers. In addition, governance of open source software on business information systems is decentralised or shared within several departments or subsidiaries. It also seems fairly complex to integrate open source functional bricks within *Corporate*, finance and management information systems that sometimes consist of heterogeneous bricks (Oracle appli, SAP, MSBI, niche software packages, specific developments, SaaS solutions, etc.). Open source solutions require additional IT integration work in relation to proprietary solutions. Although IT operators are responsible for this, it is increasingly less the case the closer we get to business spheres. Significant integration effort requires new skills that are not easy to find on the market, which increases the costs.

Open source, an alternative to major IT providers

Taking the plunge into open source

Lastly, business offers seem to be designed for subsidiaries or SME, but are not necessarily “scalable”, i.e. designed on the scale of multinationals.

2.2. Specificities in terms of skills

Recruitment and building the loyalty of profiles able to work in the open source sector are major challenges for companies.

Open source, an attractive recruitment lever: working on open source software with the associated ecosystem constitutes an attractive recruitment lever: opening of communities, acceptance of sharing, etc.

Open source talent management: discovering open source talented individuals is important to avoid seeing them attracted by other open source ecosystem companies.

Training and recognition within the open source environment: the open source ecosystem facilitates the development of skills with the collaboration, mutual assistance and emulation that are inherent to them. In fact, within this ecosystem, contributors help each other mutually and aspire to become committers. This step represents recognition of the quality of their contributions and their involvement in the community. As a reminder, the role of contributors is to develop code, track bugs, check performances and also document their developments. *Committers*, who are limited in number, have write access to the open source code and approve the quality of the code of contributors. A developer becomes a *committer* because he/she is chosen by his/her committer peers.

Integration of open source profiles into major companies: the integration of these profiles may generate cultural tensions. In general, they have very different cultures from those practised within major companies. This represents a real culture change and way of producing issue: ability to change *delivery* modes, organisation, etc.

Multiplication of required skills: a multiplication of skills can be observed for managing proprietary and free software.

Choosing open source solutions requires having an overview of the “market”, its evolution, and understanding the roles of players (publishers, communities, etc.). In addition, specialist technical resources may be required to develop specificities or switch over to the *make* in order to meet the company’s needs.

Choosing the type and scope of support according to the needs, and having several contact people instead of just one with major publishers, also requires skills. Indeed, with providers or subscription models, companies have been used to dealing with a single manager who took care of everything, i.e. development, support and maintenance. The multiplicity of types of possible support with free software, and the scope of the guarantee and support service to be defined, have a certain complexity that you should know how to manage.

2.3. Strategic advantages

It should be noted that open source forces the IT provider to ensure *fair play* and the excellence of its model. As the source code is public, the customer or part of the community may decide to create a fork at any time in the event of disagreement.

2.3.1. On the system level

Compared to “major provider” software, open source software provides companies with the opportunity of regaining control of the information system with regard to:

- Compliance;
- Security;
- Obsolescence;
- Interoperability;
- Reversibility;
- IT rationalisation;
- Support provider selection.

Compliance - security:

Open source software has a decisive structural advantage in terms of compliance and IT security over proprietary software, the code of which is closed: the code is open and therefore can be analysed. The company can check the compliance with regulations. This also leads to security due to transparency. Indeed, although the availability of the source code does not guarantee security, the ability to change it at least guarantees the possibility of obtaining a patch in the event of exposure to a vulnerability. However, companies want to avoid having to correct vulnerabilities as much as possible, so they are relying on communities.

Obsolescence - continuity- interoperability - reversibility:

When choosing a tool, its continuity may be key. For example, in aviation, the guarantee covers a period of 50 years. Trust related to the continuity of a tool is given to a well-known provider more easily than to open source software. And yet, open source software has no time limit on its use. Corrections are possible even over the very long term. As the software is open source, even if the publisher is taken over or has gone bankrupt, the source code remains available. Lastly, if the editor chooses a direction that is not suitable, the fork is possible. The company chooses the required version of the open source software. It is not dependent on the scheduled obsolescence “desired” by publishers. In other words, it no longer needs to change to extended support; it no longer runs the risk of basing its information system without support on software versions that are too old since it is investing in the skills needed to exercise this independence. Due to its nature, open source software is easy to interoperate and makes reversibility possible.

IT rationalisation:

Companies are increasingly trying to rationalise their information systems. For this, they are dividing projects into modules or bricks with standardised and independent interfaces. They are gradually trying to eliminate proprietary interdependencies between projects and between proprietary tools. They are developing a common baseware and are limiting specific developments. In addition to the imposition of open standards bordering each project, some of them are targeting the independence of software platforms in relation to the operating system and are limiting the technologies used. This helps control operational costs.

Open source, an alternative to major IT providers

Taking the plunge into open source

France's State information system is vast, complex and not very centralised, and has applications that may date back several decades. With the constant emergence of new business needs that are added to the existing information system, a continuous rationalisation effort is needed to control costs. The pooling strategy put in place does not systematically look for a unique solution. Instead of relying on a traditional urbanisation approach, it focuses successively on various functional segments to avoid offering just one single application that often justifies return on investment rather than suitability with the need. Therefore, the aim of the "State platform" strategy is a certain "software permaculture" in order to offer basic baseware, which is a common denominator for most uses, but that allows everyone to develop it or depart from it in a subsidiarity principle. This approach offers greater resilience and enables better integration of the innovations that may come from small pilot projects in order to be progressively integrated into the baseware. This approach requires mastery of the architecture in order to define the structure of the information system and permit various assemblies, like with Lego. Open source and API interfaces are prerequisites for this adaptive and controlled rationalisation of the information system.

DINSIC, Laurent Joubert

Support provider selection:

As opposed to proprietary software, free software gives **the company the freedom to choose the service provider** who will provide the maintenance and support services. In addition, in the event of force majeure, the company can take over the open source software to correct a serious bug that is difficult to resolve. Needless to say, this should remain exceptional. Of course, this is subject to having skilled resources.

2.3.2. On the room for manoeuvre level

In the open source business model, **excluding** open source subscription products, the user pays for the service associated to the exact need and not for the software annuity. This is one of the reasons why the open source alternative is being studied carefully.

In order to meet a defined use case, companies must now include all possible available solutions with providers, open source communities and service providers who will carry out customised developments. Armed with solutions for their use cases, they pragmatically select the solution with the best return on investment. Studying alternative solutions provides companies with some room for manoeuvre in negotiations: major publishers know that the company is open to the possibility of switching to open source solutions. Moreover, sometimes they have already switched to these solutions for their infrastructures.

A large number of companies are integrating at least one open source software solution into their calls for tender. This practice is part of the governance and enables them, on one hand, to remain attentive to what the open source world can offer, and on the other hand, moderate competitor prices to the call for tender. Competitors know that open source, therefore "free of charge" software, is in the running, which encourages them to remain reasonable, in particular, with the price of their licences.

Lagardère Ressources, Cyril Bartolo

Open source, an alternative to major IT providers

Taking the plunge into open source

Lastly, opting for open source contributes to Corporate Social Responsibility (CSR). Indeed, integration into the information system is carried out locally with support contracts in France. Major providers generally invoice companies from their foreign entity.

2.4. Competitive advantages

2.4.1. Opening and collaborating

Digital transformation provides a company with the ability to find new sources of growth in new markets. The cumbersome rules of traditional publishers can be overcome through Open source solutions, which can speed up the process and rapidly test these sources of growth in an extremely fast-changing economic context. Indeed, open source software bypasses the licence loan phase or negotiations related to the loan. Open source software means that POC (Proof of Concept) can be initiated without discussion with the publisher. Testing is carried out using their own resources. If necessary, the company uses services that it is used to.

Open source also represents a real **innovative program accelerator** because free software provides technological drivers, in particular, in the sectors of IT agility, DevOps (mainly open source solutions are deployed), cloud, open APIs, and IoT, etc.

Using open source solutions facilitates collaboration and agility.

Many digital services are based on OSS Open Support System, such as SMACS: Social, Mobility, Analytics, Cloud and Security. In order to take market share in the sectors of innovation and more specifically, Artificial Intelligence, augmented reality, *big data*, and *analytics*, the big five tech companies (GAFAM) are placing algorithms in open source. Therefore, everyone is using them and they are becoming the solutions used. R&D departments can network on their algorithms in complete transparency.

Lastly, the open source ecosystem, a true technological research laboratory, enables companies to **identify the emerging technological trends** that are in vogue or growing and analyse their evolution. Therefore, this ecosystem represents a **barometer** indicating the continuity of the various technologies and their potential to discover sources of growth.

Some companies have specific needs with no solutions available on the market. Indeed, the more specifically the technology meets a specific need, the less likely it will be of interest to other companies. This raises the question of make or buy. Opting for make in open source represents a genuine project accelerator if the company has adopted the open source *"mindset"*. Free software makes it possible to overcome certain publisher constraints and provides **greater autonomy**. It is important to take the time ratio into account.

Several companies ultimately choose to collaborate when they have common needs or at the very least a common base. They then decide to develop a generic open source core maintained by an open source publisher that they co-finance. Each of them then develops plug-ins to meet their specific case. The **scalability and genericity of open source** software is therefore crucial. As a reminder, the scalability principle is based on an open source platform, i.e. this *plugin* base enables solutions to be provided that are open from a

Open source, an alternative to major IT providers

Taking the plunge into open source

use case point of view. **Changing software to open source makes it possible to reach a wide community, including worldwide.** This extended distribution enables open source publishers to fast-track their software.

When Obeo became an open source specialist, its 5,000 annual hits rose to 5,000 monthly and is now 5,000 a week. Former visitors were mainly French, now 75% are from all over the world.

Obeo, Etienne Juliot

2.4.2. Pooling

Major companies are pooling their resources in order to develop common open source software solutions within alliances, associations, or industrial sectors (vertical industries).

In fact, major companies have already started to pool software developments in order to **focus on the applications and services side that make them stand out in the market.** Such is the case with the GENIVI alliance. The aim of this not-for-profit consortium founded in 2009 by BMW Group, Delphi, GM, Intel, Magneti-Marelli, Groupe PSA, Visteon, and Wind River Systems, is to create a Linux-based operating system standard for the automotive navigation systems and multimedia systems industry. The GENIVI alliance opened its open source project under the aegis of the Linux Foundation. The alliance now includes over 160 member companies that work together to provide an open and coherent Linux-based software platform for the entire automotive industry.

Pooling also takes place within developer communities and also within user communities who make the product reliable and stable.

Nowadays, companies are no longer hesitating to group together within inter-company partnerships in order to share their feedback and even pool developments. Such is the case of associations like the PGGTIE and TOSIT.

The aim of the PGGTIE, the Inter-Company Working Group of the PostgreSQL.fr association in accordance with the French 1901 Law of Associations, is to **promote PostgreSQL database management systems** in French-speaking countries. The working group's objective is to pool efforts for 3 concrete actions on PostgreSQL:

- Share implementation feedback. Members exchange their respective architectures, and the consolidation of orders of magnitude on the base and on volumes. A toolbox is provided to share and consolidate the respective documents in order to publish their results;
- Contribute to continuity, improvement and operability;
- Encourage companies to use and adopt PostgreSQL. Also encourage support by software package publishers.

The [TOSIT](#) association, founded in July 2017 that included 8 members when it started up, is a structure supporting the emergence of open source software and free licence IT solutions, in particular concerning cybersecurity. TOSIT stands for The Open Source I Trust. TOSIT's

Open source, an alternative to major IT providers

Taking the plunge into open source

specificity is to offer a legal structure for getting member companies **involved in the action rapidly**. The TOSIT association has several objectives:

- Promote discussions with industrial feedback and the sharing of good practices;
- Actively collaborate by starting **strategic or technical studies and watches**, by starting projects to secure the most critical bricks of information systems, and by formalising and **implementing projects via partnership agreements**;
- Apply the TOSIT structuring principles that involve selecting industrialisation-focused products for major groups, transferring where possible developments from TOSIT to the community, avoiding solutions where the ownership rights are not extensively shared, decreasing dependency on open source with subscriptions, and switching from a deployment-related expense to fixed-price basis expense.

Cigref commits the major companies to contribute within open source communities, group together within Economic Interest Groups (EIG) in order to share open source developments around common interests and where possible transfer them to communities.

3. Free software risk management

Keeping risks to an acceptable level is vital for the company. Risk management in general serves several purposes:

- Create and **maintain the value, the assets and the reputation** of a company;
- **Secure the company's decision-making and processes to encourage the achievement of objectives;**
- Promote the **coherence of actions** using the corporate values;
- **Mobilise the company's employees** around a common vision of the main risks.

As with publisher software, open source software constitutes a risk for the company. However, proprietary providers benefit from a favourable historic precedence with known solutions and with a **reputation aura** within the company. It is important to specify the historic context in order to fully understand software-related issues. In the 20th Century, publishers enjoyed certain monopolies. The models of these proprietary providers offered solutions “**culturally embedded**” within the company. Their choice of solutions was therefore simply justified internally. Their **marketing power** is high and they do not hesitate to carry out **lobbying with decision-makers**. Major publishers constitute guarantee brands, an insurance policy, a guarantee of security for the Executive Committee and the CEO. In addition, they propose a **single manager** who takes care of everything, such as, development, support and maintenance. **This model reduces the individual's risk-taking ... at the risk of increasing it for the company.** Indeed, this model constitutes a risk for the company with uncertainties around the evolution of costs and metrics, and uncontrolled *roadmap*, etc.

Two main types of software-related risks emerge: the **ability to make** and the **operational risk**. Companies favour a pragmatic approach when choosing a solution: they look for the most suitable open source software for the use case and make sure that the required level of support is available. The level of support is based on the **criticality of the use case's sectors**. However, when the open source software does not need support, this option is chosen.

3.1. Free and proprietary software risk types

3.1.1. Ability to make

“Ability to make” concerns the case of software that requires significant integration work on complex platforms with a large number of components. *Software* development for the integration is not the company's core business, i.e. there are no resources for carrying out the integration. In other words, resources and budgets are restricted. This constitutes a control risk.

Taking the plunge into open source

3.1.2. Operational risk

The second risk is operational. Once the software has been integrated, the continuity of service of critical systems must be guaranteed. This is why the company must have a **support and assistance system** in the event of incidents on critical systems. Indeed, an interruption of service on some software is unacceptable for business units, prospects, or other.

3.2. Solutions to the risks

Several types of solutions are possible for free software.

3.2.1. No specific support

This solution concerns **applications with a standard or fairly high SLA, Service Level Agreement**⁵ (moderate economic, legal, human resources or image impact). These are simple and reliable software products with proven quality, scalability and continuous improvements. These software products are widely installed and used by a large and active community that develops the product. The company has sufficient internal control over the community version to avoid and deal with incidents. Therefore, it does not need support. In this case, the products cited as examples are Tomcat, and the MySQL database.

3.2.2. Selecting a commercial distribution

Companies contributing to the working group select a distribution for **complex infrastructure platforms and products with high SLA**, such as Linux Suse, OpenStack, for example. All of the components of this platform **require significant expert work for the integration**. In this case, the system selected by the companies is a commercial distribution system. The company selected or the distributor provides the **integration, support and related services**, such as for example, insurance in "traditional" mode. This solution may be a transient provision, while the company improves the skills of its own resources in order to be able to subsequently provide it.

3.2.3. Acquisition of "editor" or third-party support

The option of acquisition of "editor" or third-party support also applies to infrastructure platforms or applications with a **high availability requirement**, in other words with **high SLA**. The products concerned are complex products or platforms, or even the key products of applications, such as databases. In this case, **only the support is concerned and not the integration**. Two systems are selected: **subscription for support** with the free software

⁵ The *Service-level agreement* (SLA) is a commitment between a service provider and a client. Particular aspects of the service - quality, availability, responsibilities - are agreed between the service provider and the service user. The most common component of SLA is that the services should be provided to the customer as agreed upon in the contract. As an example, Internet service providers and telcos will commonly include service level agreements within the terms of their contracts with customers to define the level(s) of service being sold in plain language terms (Source: Wikipedia)

Taking the plunge into open source

editor or subscription for support with a third-party company. RedHat Fuse, which requires platform availability, is an illustration of such cases.

The working group companies used this type of support to guarantee PostgreSQL because critical applications use it.

3.2.4. Option of no reinsurance policy

The French Gendarmerie Nationale made the strategic choice of having no reinsurance policy. Consequently, to date, it is organised and provides support with a level at least equivalent to that of major publishers. Based on its requirements, it has invested internally to have the skills for mastering software, and/or in open source communities. Several situations arise based on the solidity of the community developing the open source software:

- Firstly, if the open source software is extensively used and supported by a solid community (such as open office, open LDAP, etc.), there is no specific risk management because the community takes care of it. The bug detected is transmitted to the community which usually corrects it as quickly as a “major” publisher.
- If software that is vital for the organisation is supported by communities that are limited or considered to be insufficient, the Gendarmerie Nationale adds resources in order to contribute to the *roadmap* and adapt the products to the requirements. The take-over of the OCS-Inventory community between 2005 and 2010 is an illustration of this. If necessary, it asks external service providers to develop the new modules that they need. These modules are transferred to the community, which therefore avoids maintenance in operational condition.
- The Gendarmerie Nationale has created communities for critical software when the offer is not satisfactory. For example, for the SSO LemonLDAP::NG, it makes sure that the community can then gather momentum so that it can withdraw from it. Maintenance in operational condition is usually handled by service providers who use the software in their business models.

Gendarmerie Nationale

4. Starting the transition to open source

In order to encourage the company's employees to use open source solutions, it needs to prepare the **communication**. Indeed, some companies have explained switching to open source solutions as a way of contributing to budget restrictions and savings. This didn't last long, as some managers chose to revert back to the earlier proprietary software. **Presenting a positive message, presenting it at the right time, and demonstrating an exciting development for employees is crucial.**

Encouraging employees to opt for the open source solution for a given use case, if it is the most suitable, requires assistance. Indeed, employees are still not aware of the specificities of this ecosystem. The maturity of the employees within the same company varies significantly. Ignorance leads to a lack of confidence. In addition, failing to analyse certain points may result in surprises, for example, on the budget level. This chapter also proposes governance recommendations.

4.1. Acculturation of employees and managers

Using open source solutions is not only about a technological change. Open source solutions also have business models with a pricing strategy specific to their ecosystem. The **acculturation** of employees is therefore important and all the more so for **managers**, so that choices are made with confidence and a full understanding of the reasoning behind them. In fact, managers have been approached by the sales teams of proprietary providers, resources that in general open source providers do not have. Correctly analysing where intellectual property is situated, identifying the type of licence used, selecting the support suited to the needs, and understanding the ecosystem to be able to contribute or influence the roadmap, etc., require a basic culture to avoid getting it completely wrong.

In addition, business departments will not hesitate to study open source solutions for the required use case if they have **confidence** in the product and in the associated support. Support sometimes seems to be lacking for some business applications or solutions on the middleware part managed by Ops. Hence the importance of comparing the support and guarantees with those of the providers.

4.2. Software selection governance

The establishment of governance in the selection of solutions on different levels is key in the use of free software within an IT department.

Firstly, in order to select a provider of this type, a common organisation needs to be set up with all employees with the tools that facilitate software selection. The following example was shared:

Open source, an alternative to major IT providers

Taking the plunge into open source

- Map software solution needs by family and by criterion for the use cases in question.
- Systematically look for free alternatives.
- Establish governance of technological choices:
 - Put in place mechanisms for consulting and approving these choices by stakeholders;
 - Check application of the choice and the exemptions.

The CNAM is assisted in order to gather information on its study regarding technological developments, and in order to study the maturity and continuity of free software in a sector. Based on this analysis carried out with an in-house team, it is setting up software selection governance. These options are then incorporated into its in-house catalogue of free software included in its unique support contract thereby making it possible to define governance on use of the software that it uses (only the software in the catalogue can be used).

CNAM, Alain Issarni

The QSOS, *Qualification and Selection of Open Source Software*, analysis method highlights what makes the free solutions studied stand out. This requires computer experts to be in contact with the open source communities. This contact between computer experts and the community also provides feedback on the *roadmap*.

The DINSIC (*Direction Interministérielle des Systèmes d'Information et de Communication de l'Etat* - Interdepartmental Directorate of information systems and communication) is currently carrying out an experiment to obtain metrics on the non-use of free solutions listed in their company catalogue. They are trying to refine the use metrics in order to control uses better.

DINSIC, Laurent Joubert

Once the open source software has been selected, it is recommended to carry out an audit on the application during production and obtain feedback on its implementation and integration into the information system.

Lastly, it is important to establish governance of the software registered with the support to decide on a monthly basis which software and versions should be deployed.

4.3. Consultations and calls for tender

Where possible, companies are specifying the systematic transfer of open source developments to communities in their consultations.

As discussed above, companies are rationalising their information systems. One of the participants of the working group systematically requires the independence of software platforms in relation to the operating system in its calls for tender and limits the technologies used.

In addition, in order to compare the responses to the calls for tender on a like for like basis, public administrations and companies are imposing brick standardisation rules and

Open source, an alternative to major IT providers

Taking the plunge into open source

modularity requirements. In addition to the technical solution, they require an estimation of the **long-term operational costs**. Consequently, providers must adapt their solutions to the company's needs and constraints, and not the other way around. For example, when a proprietary model is proposed, the provider must integrate all of the hidden costs into its tender, such as the "technological" training costs for all in-house teams who are working 24/7, 365 days/year.

4.4. State example

The French State has been positioning **open source as a strategic lever for France** since 2012, with the Ayrault circular⁶. This circular lays the groundwork for the use of open source within the public framework after several years during which the issue regarding the use of free software could have been the subject of numerous discussions within the administration. It proposes a series of orientations and recommendations relative to the correct use of free software. From now on, in order to meet business needs, free software must be considered as being on an equal footing with other solutions, publishers or combinations.

To achieve maximum economic and quality efficiency, free software should be used in a concerted and coordinated way. A software convergence framework to be preferred in the development of the State's information systems, defined in 2012, is retained during interdepartmental consultations. First and foremost, it affects the systems the most used on servers and on workstations. This framework does not constitute an obstacle to innovation through the testing of new strains, which could help develop the framework. In addition, this framework does not make adaptive development mandatory for non-compliant existing applications. However, it does define the benchmark versions to be preferred and indicates the solutions to be abandoned, with possible reserves for specific use contexts. Thus, it contributes to the progressive convergence of operating contexts and to the pooling of certain resources. In light of this, it must be integrated into all ministerial technological frameworks and be taken into account for new developments and major improvements. All recommended free software is presented in the form of an interdepartmental free software base (SILL) and is maintained by the DINSIC.

⁶ See url http://circulaire.legifrance.gouv.fr/pdf/2012/09/cir_35837.pdf

5. Free software selection: analysis criteria and processes

Finding free alternatives not only involves assessing the software *business model* category but also the technical elements of the solution and the associated support. Free software analysis criteria and selection processes and software support selection processes are proposed in this chapter.

5.1. Open source software selection process

Companies have a pragmatic approach when purchasing software. They choose the best compromise between **business need/price/support** based on the criticality of the need. Free solutions are systematically assessed. However, if the free software does not meet the need, then proprietary publisher products will be selected.

The software purchasing strategy is based on the **user need** (ability of the software to cover the need) and the **value in use**. In fact, an **expression of operational need** is required. This prevents some employees from arriving with a software name to purchase that ultimately does not correspond to the need.

Some companies or public administrations define the components based on the value in use and map the value chain produced. This then enables them to consider free components. This approach is used to offer a base, i.e. a default solution that meets the user need for the majority of the various entities. Thus, each entity interested in the value in use is able to develop its solutions on it in order to adapt its specificities. Companies or public administrations want to structure their information systems to offer elements in the form of micro-services.

During its free software strategy seminar held in April 2017, the DINSIC worked on the following segmentation of the needs:

1. **Need for commodities** which results in a strong reversibility capacity with standard data and a mature market. In this case, price is a key factor and all types of solution may be considered to meet the need as best as possible, including SaaS. Open source is still an advantage irrespective of the solution selected;
2. **Need for expertise or niche**. This purchase meets a specific need that requires expertise in addition to the software. Selection of the solution will take this need for service into account and reversibility will not be carried out without significant effort. The solutions available on the market are generally mainly proprietary *COTS* (*commercial off the shelf software*), although the market is increasingly shifting towards SaaS or subscriptions and even though open source solutions can be found with expertise services. In this case, selection is based on solutions that guarantee the best integration, with the following order of preference: *Brick already existing within the administration* > open source > SaaS or COTS > proprietary COTS >

Open source, an alternative to major IT providers

Taking the plunge into open source

proprietary SaaS. This is to reduce dependency on the provider and the need for reactivity if licence costs increase.

3. **Specific critical need or need having a competitive or differentiation advantage**, in other words a unique need regarding an unresolved issue. In this case, a specific development is needed and open source provides continuity guarantees in relation to proprietary development *frameworks*. For a unique unresolved need, particularly in the public sphere, the DINSIC is encouraging opening the code in open source at the very beginning of the developments to promote collaboration, have greater porosity with an ecosystem of innovators and reduce development costs.

DINSIC, Laurent Joubert

5.2. Open source solution analysis criteria

The company would be wise in offering all its employees a **common open source software analysis method** that takes into account **long-term risk-taking**. A list of analysis criteria is proposed below in order to **classify the open source software**.

- **Software provenance**: community software or with subscription with its authors;
- **Intellectual property**, in other words ownership rights: hosted within a foundation, shared between contributors (e.g. case of PostgreSQL) or concentrated?
- **Licence**: is the software free? copyleft?
- **User community**: who uses the software? Is the user community made up of industrialists? The user community has a proportional influence on its size because its main role is to pass on malfunctions and suggestions.
- **Developer community**: what is the policy concerning contribution? Are contributions encouraged or restricted? Software quality is often proportional to the number of developers. The more extensive the development community, the more it becomes a gauge of quality and reactivity.
- **Technique and security**: is the software mature, i.e. already used for a long time, and does it offer regular updates or niches? Is it secure?
- **Centralised or community R&D funding**: who funds the software's R&D? (Those who decide on the strategy. They provide priorities and define the roadmap).
- **Community version**: is it complete or is it a loss leader?
- **Innovative character**: is it innovative?

EDF has defined an open source solution analysis grid with associated methodology. It includes a colour code for scoring the software.

During a working group, EDF presented the decision tree used in the selection of free software. In particular, it includes the ROI in relation to publisher solutions and subscription products under conditions.

EDF, Aimery Assire

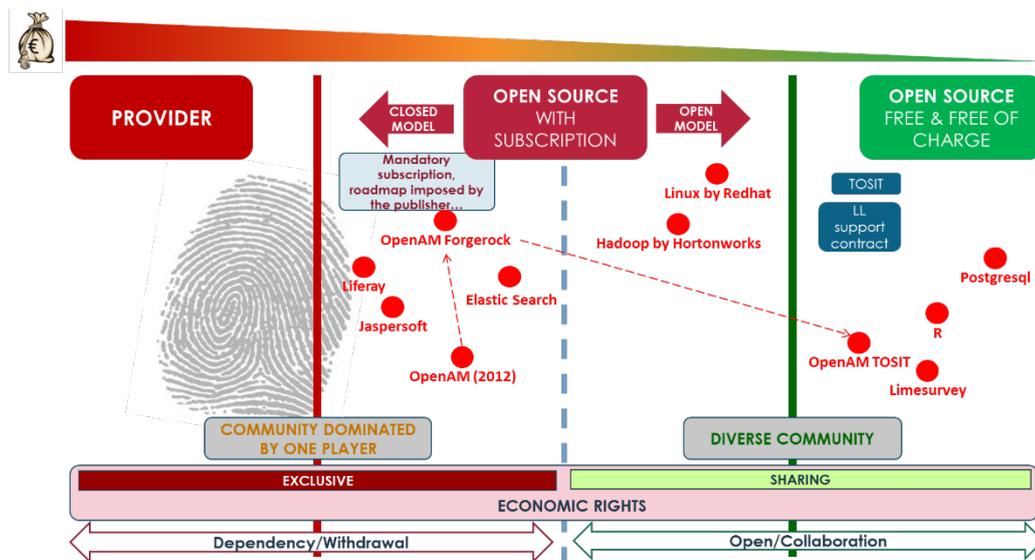
Open source, an alternative to major IT providers

Taking the plunge into open source

EDF visualises various open source software programs classifying them using the grid below. This classification, which includes expenditure along the vertical axis, is based on intellectual property, exclusive or shared, community type, multiple or dominated by a player, dependence and isolation or opening and collaboration.

EDF, Aimery Assire

Open source software business models



Source: EDF - 2017

Figure 3: Open source software business models

When open source software solutions are proposed by *startups*, the main group must make sure that the level of dependency of the startup with the main company using the solution is limited.

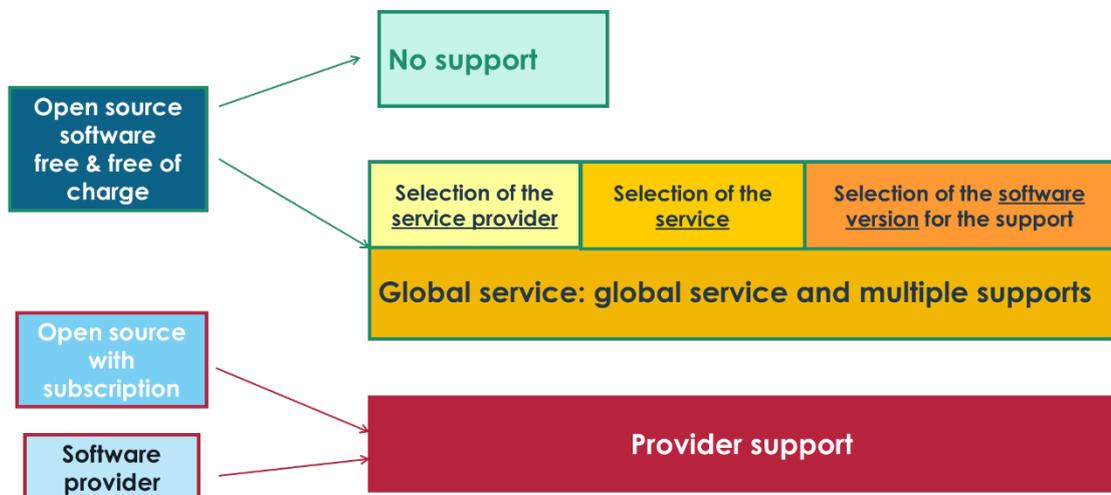
5.3. Open source solution support selection process, excluding solutions with subscription

As with any software, free software requires maintenance and support. For open source solutions, the company can define the scope of the service, providing flexibility that is not offered by traditional publishers. This includes the possibility of benefiting from a global service on several open source software programs, choosing the software versions to be supported, requesting the backporting of all patches implemented in more recent versions on the version used by the company. With regard to backporting, if the latest version corrects a *bug* detected, the company request backporting of the correction of the *bug* on the version used. Therefore, it is not obliged to upgrade to the latest version in order to have the correction of the bug.

In its unique support contract and within the context of the resolution of *bugs* in its version used, the CNAM can request that any patch that may exist in a more recent version be backporting into its version. This gets around the classic risk of the publisher who requests that the latest version must first be installed before analysing them. This means that no cumbersome work is needed to migrate to the latest version when correcting the *bugs* is urgent. The risk of the pressure of scheduled obsolescence is therefore limited.

CNAM, Alain Issarni

Maintenance & Support



Source: Cigref

Figure 4: Maintenance and support depending on the software

Some companies opt for a single service provider to provide the maintenance and support for all open source software used in the company: **global service and multiple support**. In this case, this involves a fixed-price support service on a maximum and defined number of open source programs during the call for tenders.

Conventional support and maintenance requests are made, such as, resolution of the problems encountered by users when installing or using open source software; answers to the questions that product managers may ask in the industrialisation, deployment preparation, security or maintenance fields; coverage of the risk factors including operational and security risks of each version (security intelligence in particular); and monitoring of the software's ongoing maintenance (minor and major versions) covered by the contract and available with the community.

The guarantees offered by the service providers are at least identical to those of major publishers. They include deployment, 7 days/week support, corrective maintenance, guarantee of correct operation, or even the compliance of the solution and ultimately upgrades.

The service levels are determined by the company in order to meet its needs. In fact, critical and standard support levels are determined with response times depending on whether a blocking or non-blocking incident is concerned. The 4 actions defined for each of these

Open source, an alternative to major IT providers

Taking the plunge into open source

supports are reminding the user, responding to a request for information, providing a workaround solution, and providing a definitive solution.

Skills are needed for successful completion of the above-listed actions: generalist support, specialist support or autonomy. A choice is therefore required: internalising or sub-contracting these skills.

Conclusion

Free software enables companies and public administrations to gain **independence** and **autonomy** in relation to major providers.

With their pragmatic approach, companies explore all of the solutions that meet the required use case, including free software. They select the solution that generates a proven **key benefit**. This benefit may be of different types: economic, strategic or technical.

Open source software enables companies to achieve their objective of reducing the IT department's budget, even though free software has a cost. Indeed, before adding open source solutions to the scope of their possible solutions, companies acculturate their employees by explaining the specificities of this ecosystem. This also requires the in-house or outsourced development of skills in order to understand the roles of players, along with specialist technical resources if the company wants to develop certain specificities or switch to *make* in order to meet its need.

With regard to the digitalisation of business units, the company must continue looking for other sources of savings to control its IT budget. The commitment to open source solutions is not usually made on the existing software base because withdrawing from it would firstly cost too much and would then be difficult to justify. Fortunately, a hybrid strategy is possible with free software and publishers. Having an alternative solution provides room for manoeuvre and enables the company to shake things up during its negotiations with historically present major providers.

Furthermore, open source software provides technological and strategic advantages with the digital transformation of companies, such as the attractiveness of talented individuals and the development of skills. In addition, open source solutions meet current challenges, such as, *time to market*, information system control, agility at scale of the company and *open innovation*. Convinced of the power of collective intelligence and the advantages of pooling investments, software developments or experience sharing within EIGs or associations, companies and public organisations are getting organised in order to collaborate.

These are all reasons why some companies, strongly supported by their Executive Committee, have established an "open source *first*" strategy.

Open source software, which has become common, is now mainly implemented on the lower layers of the information system. Companies want to increase the number free solutions on higher layers. They want to explore free software business applications. Cigref's "Alternatives to major providers" Circle will continue with the aim of identifying open source business solutions.

ABOUT CIGREF

KEY PLAYER IN THE DIGITAL SOCIETY

Cigref is a network of major French companies and public administrations set up in order to develop its members ability to acquire and master digital technology.



NETWORK OF MAJOR COMPANIES

Created in 1970, Cigref is a nonprofit organization. Its counts among its members some 150 major French corporations and public administrations across all business sectors, all users of digital services.



DIGITAL PLAYER

It is a key player and federating body in the digital society, thanks to its high-quality thinking and the extent to which it represents its members.



TO SERVE ITS MEMBERS

15 Board members, elected by the General Assembly, ensure its governance. A team of 10 permanent members leads the activities.