WHITE PAPER

The Economic Models of Free Software

December 2007

Collection April - www.april.org

April
14 rue des Panoyaux
75020 Paris
+331 78 76 92 80
contact@april.org

History

The original version of this White Paper was published by the association April <http://www.april.org> in December 2007. It is available under the *GNU Free Documentation License*, version 1.2 or later version published by the Free Software Foundation. The PDF or Open Document Format is available on the site of the April.

– Version 1.0: December, 2007

This document has been translated from French by José Fournier with the help of Marc Chauvet for appendix B. Eva Mathieu and Marc Chauvet have proofread the translation.

Preface

Free Software is not a new technology. It is a mode of production and distribution of software, i.e. a set of practices regarding management of projects, management of source code, channels to make the product available, legal requirements for use, technical assistance, and even marketing and sales forces.

This mode of production is not in itself an innovation. The concept of Free Software was formalized as early as the 1980s and related practices are probably much older. One can even say that Free Software was born when developers and users became aware of the concept of “software product” independent of equipment. Its appearance silently preceded the arrival of commercial publishing of non-free software, as we know it today. The innovation lies in the spread and uses of the Free Software phenomenon, which manifests on several fronts:

☑ The platform of design and realization of the free software has always been the “network”. Once reserved for insiders, it has become, with the spread of the Internet, the infrastructure for a “global village” favourable to the collaborative and decentralized development modes. It immediately increased the visibility of free projects.

☑ Free Software is no longer confined to the infrastructure market. Initially limited to “system and network”, free software now invades all segments of software publishing, including “trade applications” (resource planning, customer relationship management, business intelligence) and office work. Therefore, it affects users and decision makers.

☑ A complete ecosystem is now in place and, from production to implementation at user level, everyone can find free software fitting its business and its objectives. Free Software is no longer just a raw material for computer scientists and, on a model where access to sources is open by definition, there are finished products and integrators who are committed to results.

The reasons for the extraordinary expansion of the “Free” model in this decade are diverse. While not being the essential characteristic of Free Software, the often free (gratis) right to use it acts as a magnet in the short term. Its influence on the global economic balance of projects is limited and cannot be the basis for the lasting success that is beginning to emerge.

Software publishing is traditionally attached to the “trade secret” whose quality has never been comparable to that of manufacturing industries. Now, the limits of this “trade secret” are perceived by an increasingly aware public. The only “serenity insurance” for the user lies now in the service associated with the software, itself developed on the principle of “open source”.

Can we say that this is a fad fuelled by service companies and consulting firms to turn the situation to their advantage at the expense of commercial publishing?

The emergence of models based on Free Software, and more recently the "software as a service", call into question the “per copy” pay model. Are we assisting there to a new stage in the history of software?

Finally, the structural costs of distribution and protection of closed software makes production and distribution of software much more expensive. Capitalization of code is increasingly eroded since it is reduced to a single company. For many observers, the software has been a red lantern in the high-tech in terms of productivity. Are we not experiencing an expected refocusing of the market?

The current indisputable success of Free Software still raises economic questions. The purpose of this White Paper is therefore to provide a summary of the practices, the challenges and the opportunities of Free Software. Evidence suggests that the environment is conducive and that essential expertises are ready.

Jean-Marie Gouarné

Foreword

In the April, we found that companies involved in Free Software face recurring needs:

- the identification of the economic models of Free Software;
- the origin and vectors of value creation;
- the need to better communicate;
- a greater clarity regarding the market;
- a reminder of the fundamentals of our trades and the diversity of our companies;
- the role of associations and the role of professionals...

These needs have led us to create in the April a “Companies” working group.

Our initial work leads us to write this White Paper to introduce the fundamentals of our trade, make an inventory of our economic models, analyse where value is created and to identify the issues of free software. This collective work is based on the accumulated experience of all the member companies of the April, i.e. the most representative panel of companies involved in Free Software in France.

Our White Paper targets business decision makers from both the public and the private sectors, political leaders who wish to better understand the economy and the issues that lie behind Free Software, investors interested in Free Software players, journalists always looking for information and examples to illustrate the most dynamic sectors of software innovation.

More widely, this White Paper is addressed to all people interested in new information technologies and the emergence of open source software in the world.

Freely yours,

The contributing members of the “Companies” working group:

- ➔ Kathryn Fairlamb, AdaCore
- ➔ Véronique Torner & Philippe Montargès, AlterWay
- ➔ Jérôme Dumonteil & Yvon Rastetter, Ars Aperta
- ➔ Luis Belmar-Letelier, Itaapy
- ➔ Nicolas Chauvat, Logilab

under the coordination of Jean-Noël de Galzain, Wallix.

Acknowledgements

The April wishes to thank all the members of the association who assisted in the reviewing of this White Paper:

Jean-Christophe Becquet, Frédéric Couchet, Pierre Cros, Nicolas Dumoulin, Gilles Gravier, Benjamin Jean, Yann Kerbiriou, Arnaud Laprévote, Christophe Le Bars, Agnès Le Béon, Véronique Gouilly-Frossard, Eric Mahé, Eva Mathieu, Charles-Henri Schulz, Gérald Sedrati-Dinet, Benoît Sibaud.

And, for the English translation, José Fournier and Marc Chauvet.

The April also thanks all the companies that have responded to the survey that allowed the making of the mappings.

Table of contents

Preamble

The great myths of Free Software to be fought

Is Free Software Free (gratis)?

The cost of free software is mainly related to its implementation. Free (gratis) for most implemented software is a minor aspect: it is the tip of the iceberg. The business models of companies in the ecosystem of Free Software are based on innovation, interoperability, responsiveness and time to market. Some pieces of free software are also available in commercial versions. The confusion comes from the fact that the English word "free" means both "free" with reference to Liberty and "free" with reference to price.

Does Free Software Raise a Copyright Issue?

The Free Software licences have protected innovation for the last twenty years in precisely defining the rights and duties of the different actors. For a more detailed analysis of the types of licences, see the websites *http://www.gnu.org/* and *http://www.opensource.org/*.

Does Free Software Meet the Critical Needs of Companies?

On the economic, social or security fronts, Free Software provides technical, reliable and appreciated answers. It is also a way to reduce the risk related to the criticality of an application by standardizing the system components. For over a decade, major groups have deployed and based their strategy on Free Software.

What does happen if there is a Failure?

The multiplicity of actors involved in free software allows to find a faster and less costly solution to the crisis. Around each piece of software are users, specialists, associations that provide different levels of security appropriate to the type of consumption of software.

The description of the mechanisms and partners is detailed below in this White Paper.

How to put in Production Software Constantly Evolving?

Mature free software offers stable versions usable in a production context. In addition, publishers and service companies provide product offers and support around these programs and make products tailored to the specific production constraints of companies.

How to Find and Choose?

There are major free software projects organized in communities in which users, industrialists and computer scientists live together. Publishers and service companies offer a wealth of issues around choice and deployment. A representative part is regrouped in the April.

How do Business Models of Free Software differ from Traditional Models of the Software Industry?

Of course, there are fundamental differences between free software and proprietary software; however, their business models come together sometimes. The software also acts as an evolution, emphasizing the principle of competition. Users and customers are the main beneficiaries of this renewed competition.

Note on Terminology:

“Free Software” and “Open Source Software” are employed to characterize the software covered by a licence providing freedom to execute, study, redistribute, modify and improve it (see in appendix a more detailed explanation of these elements). The term “Free Software” refers to the definition of the Free Software Foundation, whereas the term “Open Source” refers to the one of the Open Source Initiative. In practice, these two definitions are close. They present however noticeably different stands. The Free Software movement is above all ethical and philosophical, based on the sharing of knowledge and mutual aid, whereas the Open Source movement focuses on free software for its practical advantages. Moreover, the term “Open Source” was often used in a wrong way to describe products not meeting the criteria of the OSI. The term Free Software being more precise and reinforcing the importance of the liberties, the April uses it preferentially and therefore it is used in this White Paper.

Free Software: a Diversified Ecosystem in Harmony with the Internet

1. The Mechanisms for the creation of Free Software

The Free Software model of creation is based on a strong human investment and on shared values within a given project. Such projects include users and developers with complementary know-how organized in communities around interacting systems of code production, documentation and updates. These organizations include users around the world and multidisciplinary teams, comprising at the same time, industrialists through the publication of research and experimental tools, publishers or service companies that spend a significant proportion of their turnover on R & D, as well as individual contributors.

This model of creation through communities promotes the democratization and the diffusion of innovation by creating a universal “commons” made of knowledge, methods and software tools, accessible to all.

Compared to earlier models, the effectiveness of this model lies in a technological breakthrough. The acceleration in the flow of development is based on a standardization shared among the participants of these communities in administrative, legal and technical fields. These combined factors are catalysts for the creation of global projects. The reduction of friction due to the rigidities of traditional computing allows the sharing of R & D throughout the Internet, to develop projects on a continuous daily process in an international context. With the elimination of artificial constraints, freedom to participate, contribute and use locally the outcome of a project creates a catalyst for value.

From this comes an acceleration of innovation that makes the success of free software in the professional and public domains.

Free Software is the result of a voluntary and spontaneous "global laboratory".

2. From System Software to Business Applications

Free Software covers all types of software: system tools (operating system, hardware drivers), programming (C, Java, Ada, Python ...), infrastructure software (web servers, mail servers, databases), security software (network security, systems and applications), middleware (Jonas, JBoss, Glassfish...), working environments (KDE, Gnome...), office tools (OpenOffice.org, Firefox...), rich client businesses (TinyERP, Eclipse, netbeans...).

Extensive lists of the available free software can be found on Internet sites such as Framasoft (http://www.framasoft.net), the FSF/UNESCO directory (http://directory.fsf.org/), Sourceforge (http://www.sourceforge.org).

3. Communities

The actors contributing to free software are computer professionals, users, researchers. They may be involved in these projects as part of their professional occupation or outside of it, for example by voluntarily reporting software defects.

These players, gathered by means of the communication tools of the Internet, form communities, i.e. a technological and relational network that allows everyone to participate in the evolution of a project while maintaining their own identity. These communities constitute agile structures which allow everyone to contribute to and benefit from the project.

4. Projects are organized in Communities

Major projects at national or local scale appear around simple collaboration tools such as software forge or mailing lists...

Beyond a certain size, and when the framework becomes international, large projects use their own infrastructure and organize themselves according to statutes allowing the effective collaboration of major players of the computing industry alongside first-time users: associations or foundations, for example. This is the case for large renowned projects such as Linux, GNU, OpenSSH, Apache, Debian, PHP, Mozilla, Perl, OpenOffice.org, Eclipse, Java, Python, Ruby, which have become standards in the computing industry.

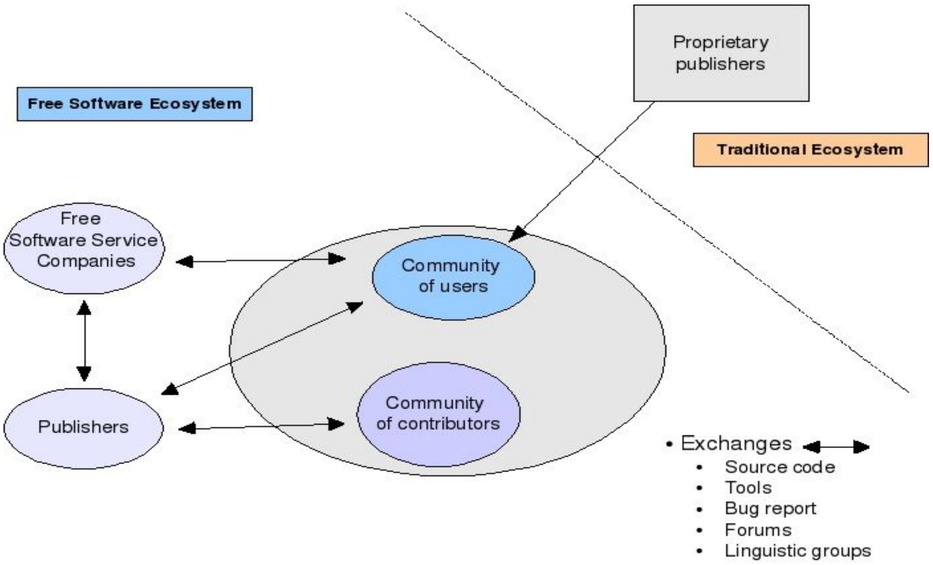
Software licences ensure legal protection: they form a foundation ensuring good stability of the achievements, in addition to the simplification of legal and contractual processes, and the protection of created works,. Free Software provides a legal framework accepted by all and ensures sustainability of the shared investment.

One can notice that free software has generated only a very small number of legal disputes, thanks to the clear and consensual terms of the free licences. In areas where the existence of software patents (mostly U.S.) creates a risk for Free Software, important guarantee funds, with agreements between the major sponsors of Freedom, provide a legal shield (for example, the Open Invention Network (OIN)).

5. The Effectiveness of Communities: an Economic and Legal Advantage

The dynamism and security of the developments carried out in communities translate into competitive and economic advantages for the enterprises that base their activity on Free Software. Notably, the use of free software reduces the time to market a solution and improves productivity of the digital economy penetrating a global market with high reactivity for minimal organizational costs.

The industry players have in common to build on the fundamental qualities of Free Software, following economic models that differ depending on the type of activity.



Comparison of the ecosystems

6. A Large Variety of Professional Actors

The professional sector that bases its activity on Free Software continues to grow: publishers, general or specialized service companies or on-line service providers, on-line merchants, large companies, research laboratories, training institutes, and industrial and computer manufacturers. Free Software also allows actors who do not contribute directly to participate in its development, by diffusing its use widely. A growing popularity of Firefox and OpenOffice.org applications, both in business and in private or public services, is observed.

The diversity of professional actors in the world of Free Software facilitates a response that meets the needs of companies.

7. A Continuum Ranging from Publisher Business to Service

Based on the cumulative experience of the members of April, the business models of Free Software can be distributed on a plan with two main axes:

- ☑ The first axis ranges from service to research and development, going through development of applications and reusable components, adaptation, installation, support and training. Expertise activities are located midway.
- ☑ The second one represents specialization and the valuation of concerned software applications: from the level of convenience to business domain specialized applications.

This model provides a mapping of the actors of the “Free” market in three maps representing three categories of actors:

- ☑ companies with a turnover below €0.2 million;
- ☑ companies with a turnover between €0.2 million and €1 million;
- ☑ companies with a turnover exceeding €1 million.

Panel Definition

The initial panel used to map actors of Free Software consists of 45 companies divided into the following characteristics:

Turnover	< €0.2 million	€0.2 million to €1 million	> €1 million
Number of companies	23	12	10

The following table describes the mapping of the companies according to the nature of their activity:

Kind of company		
Miscellaneous	4	Anaska, Ikoula, O'Reilly, Logilab
Users	1	Neuf Cegetel
SSII	2	Steria, On-X
SSLL	15	Apitux, Librenberry, Free Electrons, Silecs, Cliss XXI, Pragmatic Source, Ars Aperta, Merethis, LDD, Nereides, Siloh, Bearstech, Netaktiv, Easter Eggs, Alcôve
Integrators	14	Libreboot, Alca Torda, Sfwan, Open Via, Pi-Lyon, Zefyris, Exemole, Eitics, Alter Way, Code Lutin, Ambika, Infoclip, Itaapy, Atreal
Publishers	9	Adacore, Wallix, Linbox, Mandriva, Emencia, Sun Microsystems, Zend, Entr'Ouvert, Kelis
Total	45	

We based this classification on information supplied by the company directors through a survey of all the enterprises in the April. We took into account their revenue, their workforce, the share of their turnover related to Free Software, the share in their turnover coming from “trade applications” compared to the share of “conveniences” revenue and finally, the definition of the activity, the typology of customers, and other relevant information.

Mapping of the Actors of the Free Software Ecosystem

● *Definition of the Metrics*

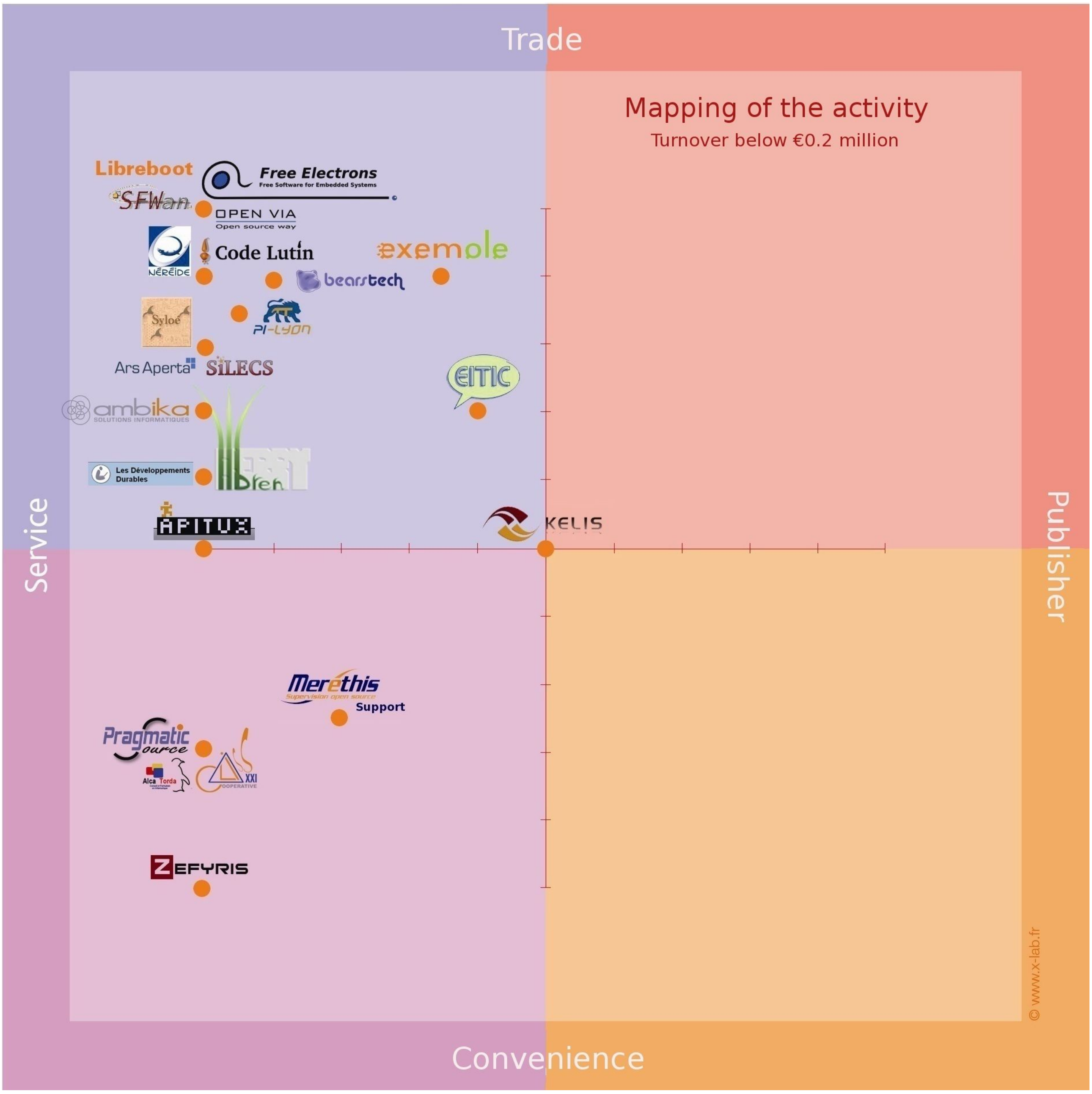
Horizontal axis:

- service: consultancy, assistance in a general meaning;
- software: revenues related to application development in a context of free software, sales of software and maintenance.

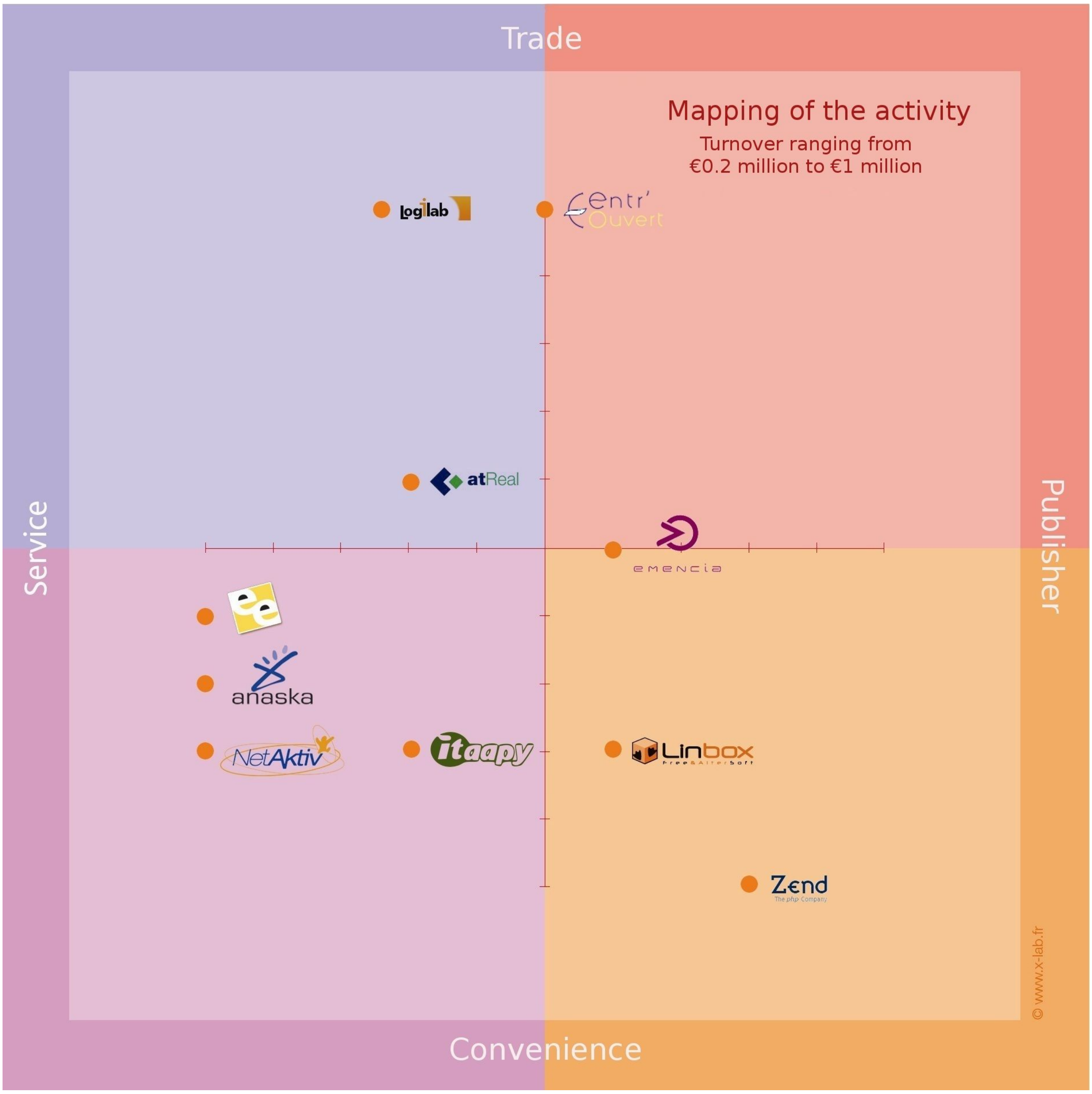
Vertical axis:

- conveniences: functional or technical specialization;
- trade: trade specialization.

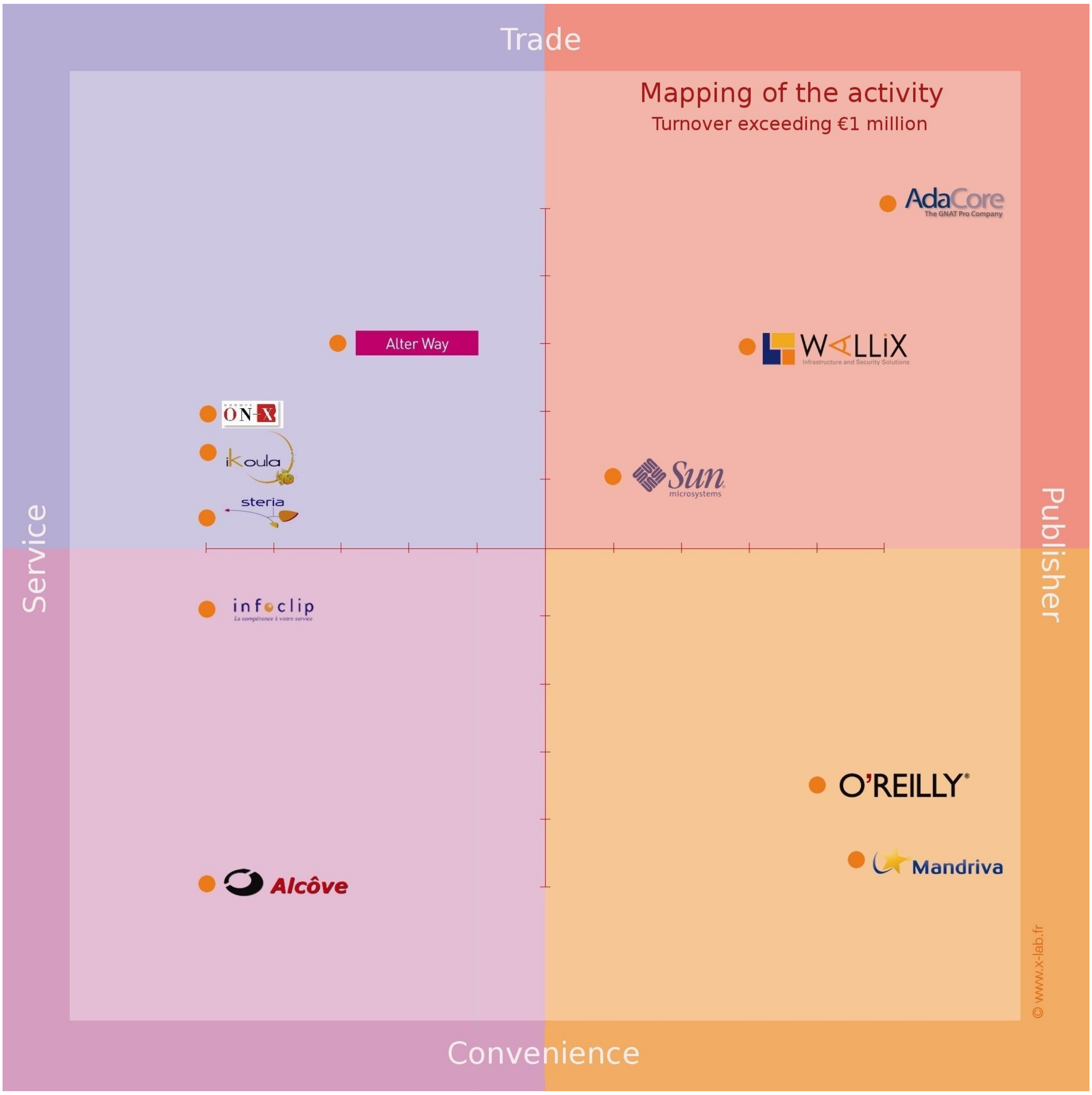
Map 1: mapping of companies with a turnover below €0.2 million



Map 2: mapping of companies with a turnover ranging from €0.2 million to €1 million



Map 3: mapping of companies with a turnover exceeding €1 million



8. Towards a New Chain of Value...

Value creation in Free Software is linked to the rapidity in making both solutions and their evolutions available to the market, and to the taking into account of the opportunities offered by new methods of collaborative work. The professional market players are on a continuum from the publisher business to service.

Those industrial enterprises whose main activity deviates from these models (such as manufacturers, on-line service providers, users) employ free software for unmarked uses at a low cost (convenience) and for specialized or complex situations (trade applications).

All these players are organized into a value chain that constitutes an ecosystem to produce a booming and diversified economy: the Free Software, the main engine of which is the community dimension focused on the added value.

Free Software: Vector of Value Creation

This section aims at:

☑ On the one hand, describing the contributions of Free Software with regard to technology, economy, innovation, its dissemination, its ownership and its reputation.

☑ On the other hand, presenting the added value for the consumers of computing technologies, the end user, the manufacturers of the computing industry, and for the entire industry.

1. General Contributions of Free Software

The idea driving the creation of Free Software is freedom. Free Software is accessible to all, anyone can contribute, use all or part of it, and create an economy or a market. This freedom leads to elements that differentiate from the traditional model:

- The entrance fee is low; licences are flexible, making the market more accessible for new entrants, without limit to the will or the imagination of everyone.
- Adaptability to everyone's model gives free software a capacity to stick to the users' needs with a total technological independence and to the developers' needs without the limits imposed by a single publisher's calendar.
- More transparency in the choice of one's partner and one's platform. This model also requires the chosen partners to stay competitive and efficient because they are judged according to the added value they produce (strict compliance with industry standards: protocols, document formats).

A healthy competition between technologies, projects, or actors of Free Software, fosters the number and quality of solutions offered.

Free software allows the computer industry to get out of the classic “customer/supplier” relationship, opening other opportunities for collaboration between actors, in particular around the principle of development communities devoted to the service of productivity and innovation. This transparency helps to raise the general level by recalibrating the market upwards.

2. Technology Extensions

Publication in the project communities on the Internet, allows a widespread access to technology and a better dissemination of knowledge. Overall, the adoption of technologies faces a double phenomenon of acceleration: the falling cost of access to knowledge and the leverage of the distribution across the global network. The value provided to users also benefits from a more rapid dissemination of technologies and recent developments.

Take the example of Sun Microsystems' contribution to the OpenOffice.org community with the development of an Open Document Format (ODF) tool kit aiming at separating the office tools from the document management. With the development and promotion of this new tool, the publisher proves his ability to meet the customer's needs while allowing the Free Software community to create new features or uses. In addition, the opportunities offered by this virtuous circle of cross-collaborations are to all stakeholders (Sun Microsystems, communities, users) as numerous as their respective willingness.

Open standards often add a vital contribution to the adoption of free software. Conversely, free software is often chosen as reference implementation of these standards, and transparency of access (gratuitousness, source code provided) encourages their wide dissemination (e.g. the standardization of the DNS protocol with Bind, or of e-mail with Sendmail).

The Free Software development mode leads to finance the development of software components or the participation in experts working days which are put together to share the results. There is no longer investment in the redesign of facilities but only in the evolution of technology: components, libraries, new functional modules...

Value no longer lies in the culture of secrecy around code control, but in the ability to communicate, build and unite a community, recruit new talents, then generate additional service needs. A user no longer buys basic software; he buys the specific modules, the expertise or the implementation exactly meeting his own needs; the model is based on added value rather than on an economy of annuities.

Opening the source code provides all together a guarantee of integrity for the development solutions around common standards, a guarantee of interoperability for complex systems, and a guarantee of continuity for the solutions. In that sense, Free Software brings a reliable answer to industry in sensitive sectors where technological independence is essential (defence, administration, energy). These include the Copernicus architecture of the French tax administration—that provides among other things the on-line income declaration service—built on free software components.

In this global economy, the spread of Free Software opens new markets and favours the emergence of new talents. The value scale is based on meritocracy and emulation around innovative technologies.

3. The Acceleration of Innovation

The sharing of research efforts around open standards helps to improve their interoperability and their adoption at a global scale. Free software now provides the bricks that are essential to the building of systems and applications. Once freed of the burden of rewriting the core functionality of their information systems, companies can focus on their core business and free up resources either for research, for new developments or for performance improvement. Broadly speaking, the competition that open source code generates, pushes the market players to differentiate themselves through constant innovation on shorter cycles.

For example, the release of the SPARC processor sources and specifications has resulted in mask versions of sixty-five nanometres by the University of Santa Cruz (USA). The publication of this work allows the entire community (and therefore the company Sun Microsystems) to share an expertise unique in the world and improve its performances.

In the field of conveniences, technical excellence and technological innovation in Free Software have led to the design of the most reliable systems (GNU / Linux, BSD, OpenSSH) towards which the Unix world converges, and of the main middleware layers (Python, Java, JBoss, ObjectWeb). The complex projects such as products with a high need for performance and reliability (virtualization, ADSL terminals, Internet portals and engines, VOIP, etc.), telephony, on board solutions massively developed around free kernels, or the world of ASP, are now all based on free software.

Free software are heavily used in advanced research fields such as HPC [High Performance Computing] (CERN is deploying computing grids with architectures based on free software), aerospace (NASA uses Java to drive space probes on Mars), genetic research or the semantic layer of the web. They benefit from the latest progress in mathematics or algorithms that are a guarantee of performance and reliability.

Laboratories around the world use also the Free Software methods of collaboration as a model of sharing: forums, wikis, P2P, software forges, community governances, and so on.

Many actions of fundamental research in areas such as code parallelization, memory management of virtual machines, taking into account “real time” applications or code refactoring methods are carried out in "Free" mode. These works initiated by industrialists in collaboration with academics and communities are the heart of a synergy as unique as promising in the history of computing.

Mapping of Economic Models

The purpose of this part is to present the continuum going from the publishing business to the service business as the basis of the creation of new economic models: the economic models of Free Software.

1. Market

The Emergence of Free Market and its Players

Year 1991 saw the emergence of the Linux kernel and its collaborative model. Its evolution has led to validate the technology now used worldwide. Closely associated with the explosion of networks, DNS Bind, Sendmail, Apache, Linux, BSD, GNU, Perl and PHP have become the “non commercial” standards of the Internet.

This deployment has been peppered with success stories such as Cobalt, Star Division, Sleepycat (for BerkeleyDB), AdaCore (for the GNAT compiler ADA) or SUSE Linux. More recently Red Hat acquired JBoss for $331 million, SourceFire was successfully introduced on the NASDAQ and Red Hat was valued $6 billion on the same market, over 12 times its annual turnover.

The first actors involved in the dissemination of free software were the publishers of GNU/Linux distributions: Red Hat in the U.S. market (1995), SUSE in Germany (1996), Mandrakesoft in France (1998 - now Mandriva) and Debian, the leader of the community distributors since the 1990s.

These companies offer to distribute the GNU/Linux system with a set of free software to install a server or a workstation. Their business model focuses on selling additional training and support offers. In parallel, between 1996 and 1999, they promoted the development of specialized service companies that emerged to offer services for Free Software implementation: development, integration and support.

The pioneers of the service in France are Alcove (1996) and Easter-Eggs (1997). They created the model of the service company specializing in Free Software. Later, other companies such as Aurora (1999), Idealx (1999), or Open Wide, Nuxeo and Linagora took over this concept. Now, this model is taken up by specialized bodies created in the major general service companies of the market.

GNU Linux, Emerging Value of the Computing Market

In late 2000, IBM announced a $1 billion investment “around Linux” and sounded the rallying of major accounts. All publishers and manufacturers joined the movement and invested progressively in the GNU/Linux platform: it emerged as a computing industry value.

Since then, the “Free” model has largely prospered and allowed to accelerate the dissemination of significant innovations in search engines (Google, Voila), computer security, object-oriented programming (Java, Python, Perl and PHP), telephony (Java) and others.

Big administrations are now clients of Free Software in the United States, in Europe, in emerging countries, in the French key ministries (Interior, Economy Finance and Industry, Agriculture, Education, etc.), the public sector (police, French atomic energy agency, National Centre for Scientific Research-CNRS) and a significant number of large companies. Recently, the PSA Group announced the adoption of GNU/Linux for 20 000 workstations.

In 2006, the French Free Software market weighs nearly €500 million in turnover, mainly in services. By 2010, it should be multiplied by 10, according to major studies available on the market (IDC, PAC). The share of Free Software in the computing service market could pass from 1.4% in 2006 to 5% in 2010.

2. Economic Models

Definition of the Continuum

The emergence of Free Software is linked to the acceleration of the market of information technology and the need of competitiveness for companies.

Key factors are:

- the strong development of the Internet that acts as a catalyst;
- the growing outsourcing of computer systems;
- the need for productivity;
- the development of free software.

The “Free” model enabled the emergence in the professional world of new companies with hybrid models. These models change in depth the relationship between publishers, distributors and service companies.

The “publishers” derive their income from the standardization sought by customers, both in software and in their evolution.

Distributors provide comprehensive suite of software in blister boxes or as appliances. They make a living on training and support.

“*SSII*[1]"“ develop offers, the logic of which is related to their know-how and combine software and trade oriented services. They make a living on deployment and integration.

Free Software has given back to engineering a new strength in making available a range of technologies and highly customizable reusable components, which are as many utilities freely available on the Internet. This leads to an inevitable rise in the added value associated with the service. This trend is supported by the improvement of tools (project management, monitoring...).

Besides, a re-balancing of technology control occurs for end customers who demand more “softwarisation” as well as on demand consumption patterns (SOA, SAAS) to industrialize the deployment and the maintenance.

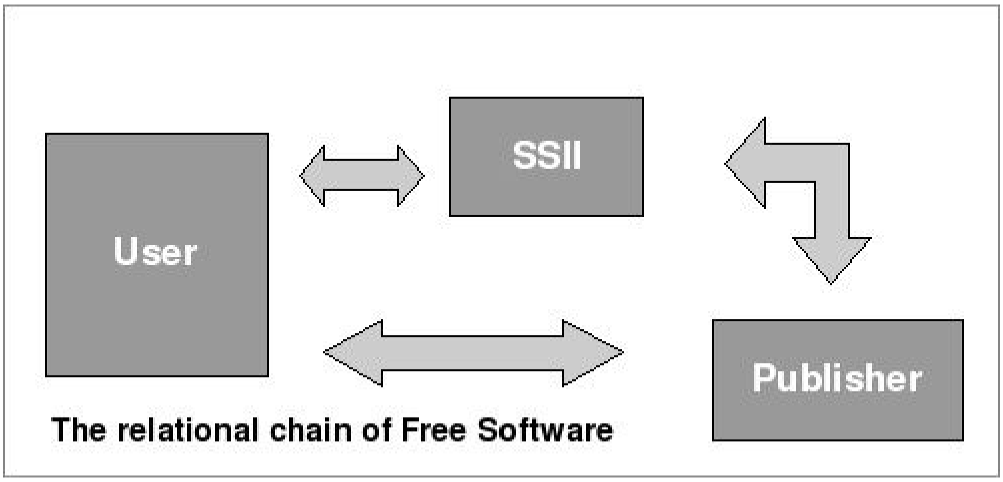
This reconciles innovation, standardization, distribution, customization and return on investment. Hence the gradual specialization of the sector's companies in hybrid models of software publishing, distribution and specific integration, very focused on a service economy.

In response, the players of proprietary computing (publishers, integrators...) must open their offers and adjust their business models by recreating the value based on innovation rather than on “annuities”.

The professionalization of the Free Software market is part of a logical continuum in which the “publisher” and “service” models converge towards hybrid models adapted to the consumption patterns of customers and to the market issues of the moment.

[1] SSII: *Société de service en Ingénierie Informatique* [Computing Engineering Service Company]

The Relational Chain of Free Software:



Are there specific Free Software Models?

We can consider that there are Free Software specific models in that they incorporate the reality of the change in the relationship between publishers and *SSII*.

Initially, distribution publishers (Red Hat, Mandriva and Novell SUSE) and service companies specializing in the implementation of free software (the *SSLL*[2]). These hybrid models used to represent the value chain of the Free Software. More recently, several factors have come to modify this value chain:

- ☑ Project sizes make the need for service more important.
- ☑ Project scope in the information system extends the needs for diversified skills.
- ☑ The choice of Free Software in critical projects induces significant needs in the fields of maintenance and warranty.
- ☑ Utilities having changed, the need for specialized software and highly technical offers lead to the need of constant evolution and technological innovation.

Thus, professionals of the “Free” world partake in all or part of this new value chain of the “Free” model, around some of the following specializations:

- ☑ Technological publishers, for the design and development of free components and tools ;
- ☑ Distributors and commercial publishers, for the distribution and marketing of these components/tools;
- ☑ Services of *SSII,* for the consultancy, assistance to clients, assistance to project management, engineering (under the direct control of the client or for a fixed sum), training, maintenance;
- ☑ Additional publisher services, for the support, the updates (certification, documentation, development, training...);
- ☑ Free ecosystem, for the contribution to the communities, the dissemination of innovations.

The model of all these players remains highly focused on different types of services, but their activity is more

[2] SSLL: *“Société de Service en Logiciel Libre”* [Service Company in Free Software]

oriented towards the user side or towards technical matters.

Shared values and common practices allow defining the Free Software companies:

- ☑ Technological watch associated with interaction with the communities of Free Software;
- ☑ Compliance with the model of open source development and the Free Software professional and ethic standards;
- ☑ Understanding, use and dissemination of free licences.

Typology of the Economic Models of Free Software

Depending on their position related to the value creation chain of Free Software, professionals differentiate from each other by the way they market their offers. We identified several economic models that incorporate in their selling model all or part of the specific characteristics of the trade specializations belonging to this value chain:

- the publisher model (dual licence or lagged licence)
- the added-value distributor model
- the application service provider (ASP) model
- the added-value services model
- the hybrid integrator model

◆ *Publisher Model of Dual Licence or Lagged License (AdaCore, Wallix)*

This model applies equally to the publishers who are revolutionizing their business models (Ingres, Sun) and to the new 100% “Free” publishers (Mysql, AdaCore and Wallix) developing new components and/or tools (frameworks). They propose to publish in the Community a mature and standardized version of their offer in order to take a position on the market. They continue to invest in new versions sold in the form of additional offers or maintenance around their core base.

In parallel, some propose a better offer in terms of functionalities for customers who request it, based on a traditional pricing, and another licensing model.

To impose a standard by opening the code is the originality of this model which creates value as far as it grants the customer accessibility as well as potential prototyping.

This is indeed a new model coming from the Free Software revolution, consisting in the opening of the source code and the implementation of innovative ideas, not to protect the companies current market, but to develop new markets.

A variant (the lagged licence model) consists, for a publisher, on an equivalent principle, to charge the client only for recent versions of its software (6 to 12 months) and then publish them as soon as it markets a new major release. The client pays only for innovation and possibly for support.

Publishers make a living on services they produce (software maintenance, reseller support and user support) and on the sale of additional modules.

◆ *The Added-Value Distributor Model (Mandriva, Red Hat)*

This model identifies the actors whose job is to ensure the physical distribution of free software within a complete operating system comprising selected versions available on the market, and around implementation and utilization tools adapted to the needs of their customers. They offer some additional services expected by customers around support and training to the software.

When purchasing proprietary software, the user pays a fee for a personal right to use it, in the form of a licence and a yearly maintenance fee. When purchasing free software from a distributor, the user has the choice between downloading his product or buying it in a classical box including a CD and the documentation. A subscription to maintenance provides access to automatic updates, as well as to various services such as technical assistance or necessary training. The user is free to consume according to his (her) needs.

Distributors generate income from the sale of physical media of their operating system in large quantities, and from the sale of value-added services: training and technical support, resale of their products through networks of dealers or OEM agreements.

◆ *ASP Model*

An ASP [Application Service Provider] is a company that provides services to its customers through the Internet. The strength of this kind of companies is to provide access to specific applications (such as a medical billing program) using a standard protocol like HTTP.

This model responds to a strong market demand for comprehensive and operational on-line applications responding to a precise specific need in the form of rental or subscription. Its value lies in the "I pay only for what I consume" and its accessibility.

ASP companies benefit from Free Software to reduce their infrastructure costs while controlling the distribution and the technology they offer their customers.

◆ *The Added-Value Services Model (Ingeniweb, Itaapy, Linagora...)*

In this model, we find companies that offer all the services of a *SSII* [Computer Service and Engineering Company] (consulting, engineering, training, TMA...) relying on a range of tools and components specific to Free Software. Often, these companies have a functional specialization (CMS, CRM, mobility) or trade specialization (ERP, industrial computing, security).

Their value relies on the know-how of their experts involved in the Free Software communities, on their strong technological expertise and on their trade functional knowledge.

This model is based on the sale of intellectual services in any form: consultancy, expertise and development for a fixed sum or under direct control of the client, TMA. This model is listed under the French marketing concept of *SSLL* [Free Software Service Company].

◆ *The Hybrid Integrator Model (Alter Way, Logilab...)*

This model should promote the necessary synergy inherent in Free Software between the publisher trade and the service trade for the sole benefit of the customer, especially in major accounts and public services.

The term “hybrid” is to be understood as the merger of “product” and “service” offers within a single provider, ensuring consistency and sustainability of the whole for the customer. This provider being positioned either on a trade axis (for example security) or on the axis of the global information system of a company (for example, the information system of either a local community, a subsidiary or a division of a CAC 40 company).

This model involves finding a minimum critical size adapted to the markets of major projects without “encapsulation” in the offers of the traditional operators and integrators (Cap Gemini, Atos Origin, Steria...).

Hence, this model facilitates the choice for a CIO to implement recognized and mature Free Software alternatives.

Revenues of the integrator come from the resale of technologies and services implemented for its customers.

3. How will the Market evolve by 2010?

So far, the main features of the Free Software market were those of an emerging market in the area of information technology, namely:

☑ A strong growth of 30% on average until 2010 (source PAC[3]), from 0.5% to 5% of the total market of information technology;

☑ A multiplicity of small players;

☑ Players with niche strategies and "all-in-one" business models tailored to the size of small projects (*SSLL*);

☑ Financial investors in the logic of venture-capital;

☑ The important and necessary role on the mass-media, of market-maker pioneers, and a marketing positioning as the only differentiating factor;

☑ A very ideological communication, and therefore strongly linked to political and/or industrial sponsors who invest in the sector either for reasons of independence and security, or in order to acquire or protect market shares;

☑ The resort to subcontracting as an economic model or to strategies of direct or indirect subsidies from the government;

☑ little growth in sales, strong reputation.

From a confidential and emerging market, Free Software is penetrating all layers of information systems and is entering **a phase of maturity, structuring and consolidation.**

From an **all-in-one model** (*SSLL*) able to offer the entire value chain of small-scale projects, or from distributor of system utilities like GNU/Linux available for download, Free Software players must adapt to a more global demand, to larger project sizes and to the professionalization of the market.

This **phase of structuring and consolidation by 2010 will need:**

☑ A reshaping of the trades and economic models of Free Software players, capable of generating intermediate models attuned with the different life cycles of the concerned players. For example, a model of value-added distributor may temporarily be part of a scheme of value-added service type to ensure its development around a first offer, waiting for the availability of new technology offers...

☑ A consolidation phase that can be made:

➔ on a vertical/trade axis (security, utilities...) and thus, potentially, by an actor such as a value-added distributor or a software publisher, internal or external to Free Software;

➔ on an axis of hybrid integrator specializing in Free Software and able to handle large projects with a higher criticality for customers;

➔ on a more conventional industrial model in the form of an extension of activity in a *SSII* [Service Company in Computing Engineering] offer that is positioned on the market with a comprehensive technology including Free and non-Free technologies.

☑ The necessity for this Free Software market to acquire **100% Free industrial champions,** with a critical size and a level of profitability that sustain the economic model of Free Software. Such

[3] PacVision May 2005

actors will be competitive compared to traditional computer players (manufacturers, publishers, integrators and...)

4. Illustration through seven Companies within April

Hereafter are illustrations of key models represented by economic players of April. Each of the following texts have been written by members of the presented company.

◆ Sun: Components Development for Distribution

Sharing and Open Source have played a decisive role in the foundation of Sun and form the heart of its current strategy. Sun has provided more lines of code in open source than any other organization. The chairman and CEO of Sun, Jonathan Schwartz, challenged his company to diffuse the source code of any software product. With the dozens of open source community sponsored by Sun, such as OpenSPARC, OpenSolaris, GlassFish and now Java Open Source Technology , Sun is about to reach its goal. Sun is also one of the major contributors to projects such as X.org, GNOME, Apache and Mozilla.

Keywords: OpenSPARC, OpenSolaris, GlassFish, Open Document Format, OpenJDK

◆ Mandriva: Distribution Publisher

Mandriva develops and distributes a Linux distribution, including up to 3 000 applications.

Mandriva targets, first the consumer market with products such as Discovery (beginner), PowerPack (confirmed), PowerPack+ (server and workstation) and Flash (a USB key solution), and secondly the major corporate account market with Corporate Desktop, Corporate Server and the Pulse, LRS and LDS deployment and administration tools. The consumer market is served around the world via large retailers, e-commerce and OEM agreements with manufacturers. The major account market is served by direct sales, mainly in France, Brazil and the United States. Mandriva offers its users Mandriva Free (pure GPL) and Mandriva One, two free distributions, maintained for free. Copies of these versions are downloaded by millions.

◆ AdaCore: 100% Free Publisher

AdaCore develops and commercializes AdaCore GNAT Pro, the development environment of reference for the Ada language (Ada 83, Ada 95 and Ada 2005). AdaCore is a 100% Free Software company whose business model is based on an annual subscription including the access to development tools under a licence suitable for industrial use, and the access to an on-line service provided by the developers of the GNAT Pro tools who are among the best experts in Ada.

AdaCore customers are major players in avionics, air traffic control, defence, space, rail transport, financial services, and medical imaging.

◆ Wallix: Dual Licence Publisher and Integrator

Wallix is publisher and integrator of Free Software for security, supervision and management of information system infrastructures. The software offer of Wallix revolves around four lines of products that are TotalSecure, Wallix LogBox, Wallix WatchServer and Wallix AdminBastion and which aim companies that want to deploy recognized, efficient and scalable solutions benefiting from the contribution of Free Software.

Expertise services and professional support available via email and phone, accompanies the Wallix offer. Wallix's Security Operation Centre (SOC) is available around the clock, 7 days a week. It currently provides remote management of over 500 security servers spread over four continents.

Wallix is the main contributor to the sshproxy project.

◆ *Itaapy: Service Company in Free Software*

Itaapy specializes in new information technologies: dematerialization of procedures, electronic document management, resource planning, e-Government and e-Learning.

Itaapy intervenes in services across all areas of software for technical and functional consultancy, assistance to clients, development and training.

The technological environment of Itaapy is the Free Software in general and the Python language in particular, for which the company is an expert in the French market.

◆ *Alter Way: Hybrid Open Source Integrator, Services and Publishing*

Alter Way federates emblematic and specialized companies that cover the main needs of the information system: content management, groupware and databases, system and network data management and hosting...

Its founding leaders, Philippe Montargis and Veronique Torner, want to offer large companies, governments, local governments and SMEs, a global industrial response provided by a single representative for the implementation of Open Source solutions.

They build a group that argues its distinctive competence in Free Software as well as a support capacity and durability comparable to those of traditional integrators.

◆ *Libre-Entreprise Network Companies on a Cooperative Model*

Libre-Entreprise, established in 2002, is historically the first network of service companies in Free Software. The companies of the Libre-Entreprise network are located throughout France and Belgium. They implement within their operation methods the principles that prevail in the Free Software community: sharing of expertise, transparency, democracy (one man = one vote). The network allows all member-companies to propose a large-scale commercial offer and to benefit from a set of specialized and diverse resources. Each company of the network operates as an independent management unit. These principles are contained in a charter: http://libre-entreprise.org/index.php/Charte

◆ *Logilab: Publisher, Integrator and Training Centre*

The particularity of Logilab is to have given itself a rather strict framework defined by its choice of techniques, tools, and application areas. In this framework defined by Linux, Debian, C, Fortran, Python, agile methods, scientific computing and knowledge management, Logilab has a wide variety of activities. It works with SMEs, with major accounts both public and private, with industry as well as with research, for consulting, training, development and so on. It is above all its technical expertise in the areas concerned and the control of selected tools that are sought by its customers. This differentiates it from the model of *SSLL* that promotes the versatility and offers its expertise for most languages and tools of the market.

Logilab also has an internal organization that incorporates certain aspects of the Free Software communities: autonomy and personal responsibility, permanent debate, geographical distribution, coordination by instant messaging, the use of system management software to share source and documents. These characteristics are often noticed because not widespread in organizations who use free software.

Remarks:

- There are varieties of other models that combine different models of sales.
- There is a marketing concept *SSLL* [Free Software Services Companies]. Those for who the Free Software share in the turnover is less than 50% are *SSII* [Computing Engineering Service Companies] unlike *SSLL* who have made it their core business.

The Technological, Economic and Strategic Stakes

Two factors have facilitated the development of Free Software: the declining cost of computers and the Internet deployment.

Other current phenomena accelerate the use of Free Software; the development of technologies results in an increase in general public equipment containing on-board software increasingly powerful and communicating.

The stakes of the digital economy in daily life thus promote an increasingly important capillarity of the software as a convenience.

1. The Issue of Free Software in the Economy of the Intangible

The influx of digital technology in all economic sectors highlights the properties of "intangibles goods". Their growing importance and the changes they lead to, have created new economic stakes around an economy of the intangible. The information objects have a key property: the marginal cost of a copy is close to zero. Until the arrival of digital technology, the economy was based mainly on the exchange of property, the possession of which is exclusive. This implies that when a piece of equipment is sold, its possession passes to the purchaser.

For an intangible asset, when the property is sold, the seller still holds it (known as non-rivalry). The deed is usually a duplication of property that does not require heavy manufacturing operation.

The paradigm of scarcity of goods facilitates the operation of the traditional economy. On the contrary, the arrival of the Internet and of the mass dissemination disrupts the existing modes of marketing while generating new opportunities for economic development. Recent debates around the downloading of musical works on the Internet and the heated vote of the DADVSI law are of course other examples. The problem of the income model for the production of information goods, including software, is in this context a major issue. Should we oppose the changes offered by the Internet by funding barriers to the diffusion?

Free Software follows the path of dissemination without limitation, offering up new economic models for the economy of the intangible.

2. The Economic Stake of Innovation

Despite the disadvantage of not being able to base its economic model on income from the sale of each copy, Free Software benefits from advantages that contribute to its success.

First, the free licensing system allows a rapid adaptation by projects to the economic globalization and to the increase in digital exchanges at a global scale, particularly via the Internet. The rapid response to the customer's needs and to changing technology is a real economic advantage.

Free Software allows building up projects involving various kinds of structures internationally. It allows participants to enter the project or leave it at any time without charge and with a minimal contractual and legal complexity. Compared to the difficulty of contractual collaboration in the context of proprietary software, this is an advantage in terms of responsiveness that can be counted in months, even years.

Then, Free Software facilitates the provision of early versions of software and the taking into account of the users' feedback very early in the development cycle.

These two features allow Free Software to benefit in general from a time to market unbeatable by a proprietary scheme. This advantage is even clearer when it applies to innovative solutions, for which to have the first implementation available on the market is paramount.

In addition, the collaborative working patterns of the Free Software sector are particularly suitable for remote work via the Internet. For example, a feature innovation as simple as generating a PDF document directly from the OpenOffice.org word processor is an ergonomic advantage that the market leader struggles to catch up because of the contractual complexity of this feature in a proprietary environment.

Innovation comes in parallel to a relative decline in the value of generic solutions ("Commoditization"). The deployment of open standards or free software facilitate the adaptation to this dynamic market, for users or and solution provider as well. The alternative solutions are multiplying and now extend to the work station

Access to Global Competition for Innovative SMEs

The advantages of Free Software in terms of speed of implementation counterbalance the absence of per copy remuneration for many small companies (especially publishers). The low cost of software distribution (free download) allows a small innovative company to obtain international recognition for its expertise. AdaCore and Mandriva are examples of companies that use the effect of zero marginal cost of duplication and the advantage of the dissemination via the Internet of software solutions. An economic model commonly associated with this type of development is selling support services around the world.

R & D Pooling of Major Organizations

For large companies, the sharing by means of Free Software of certain areas of their R & D represents a strategic choice, when benefiting from proprietary solutions is no longer a strong competitive challenge. The demand for technical support, for version monitoring (security) and more generally for expertise is present regardless of the software licence. Free Software encourages the publisher who has made this choice, to share a part of that turnover with local service companies, closer to the customer. In return, the publisher gets a widened network of support and can take position on expertise while enjoying a shared R & D (specially the reuse of free components).

The use and funding of Free Software is one of their long-term strategies. It is difficult today to imagine some applications without Free Software (the "boxes" of ADSL, the boarded network components and the service-oriented architectures on the Internet). As such, these companies are part of the ecosystem of Free Software. Thus, the economic models overpass the direct models like publisher/service to go towards indirect models of R & D pooling and "coo-petition", such as the model of rival ISP but using the same basic software bricks. A source code produced in the context of an organization's R & D has the potential to promote the fund provider in the development community over time. The importance of the generated "goodwill effect" and the increased visibility of fund providers in the world of the Internet participate in the economic returns on investment.

3. The Issue of Transparency and Interoperability

The increasing digitalization of their trade processes and their information makes large organizations increasingly dependent on their tools and therefore their suppliers. In this context, the demand for transparency on software (access to code, facility to audit it, legal security...) increases as organizations want to guarantee the operation of their systems and to minimize their dependence on external actors.

A strong point of Free Software is to provide applications for which sustainability, evolutions and costs do not depend on a single editor, what can become problematic when the weight of the publisher is such that clients cannot influence its strategy. From an economic point of view, the costs involved encourage users to compare the added value of solutions, to beware of "black boxes" and put their suppliers in a competitive situation. To present an offer with guaranties of transparency and independence becomes a competitive advantage. It is for example the choice of Sun Microsystems with Java and Open Solaris.

Digitization raises also the issue of conservation and reuse of information of the company over the long term. The use of open formats, now clearly defined specially in the French law[4], for storing information becomes gradually acknowledged. By allowing information access independently of the application, the standard formats improve the sustainability of data and open the door to Free Software.

The need for interoperability related to the increasing mesh of exchanges leads to the use of solutions based on norms and standards for interfaces and data formats, what is still in favour of Free Software.

In summary, growth of exchange and volume of information managed by companies leads to a demand for tool transparency and competitive suppliers. This translates into a request for applications opening (open source, standardized methods) and a request for interoperable standards (data permanence and easiness of communication). This request benefits directly from offers based on Free Software.

4. The stakes of Independence

Free software allows a strong control over one's information system. Notably, decisions on the evolutions of software components can be made with greater freedom than in proprietary environment. Free Software is an asset for the entities whose information system is a major issue in terms of independence. Some players use Free Software because the independence it allows from suppliers provides a high level of secrecy (Google maintains its competitors away from its infrastructure and employs many contributors to major projects). Leaders of telecommunications, computer hardware and operating system publishers are active on software that will allow them to escape the risk of domination by a single publisher. All these actors are involved directly or indirectly in major projects via free foundations (Mozilla, Apache) or by employing developers (OpenOffice.org, Python, Samba).

The governments of many countries want to increase their sovereignty over the sector, and support the Free Software through research projects or services (France and Europe, India, Japan...). Sensitive information systems of ministries may be subject to constraints of independence of suppliers (sustainability, continuity of service, neutrality of the administration, independence of the state and security).

5. The French and European Issues

Europe has become aware of the preponderant place of the United States in technology (Internet, software, hardware) and of the major investments made by other regions (Asia invested heavily in research and Free Software: $85 billion over a decade). Faced with the challenges of the digital business in terms of growth and independence, the European Union has defined a strong strategic focus on the knowledge economy (Lisbon strategy[5]).

Free Software and open standards (including Open Document Format) now allow the EU to resolve the serious issue of interoperability of systems and content among 27 countries and 23 official languages. Europe must now seize the opportunity of investing in research based on open source software to disseminate and accelerate innovation while remaining independent in its technological choices without financial dependence.

Free Software is part of an industrial revival in a context of sustainable development through the reuse of components and the facilitated consistency of economic development policies at European, regional and local levels.

[4] *Loi du 21 juin 2004 pour la confiance dans l'économie numérique - Chapitre I - Article 4* [Law of June 2004 for the confidence in the digital economy – Chapter I – Article].

[5] The Lisbon meeting of the European Council set a strategic goal to make the EU the knowledge economy the most competitive and dynamic in the world by 2010, capable of sustainable economic growth accompanied with a quantitative and qualitative improvement of employment and a stronger social coherence.

In this context, France must maintain current initiatives:

- ☑ Collaboration of actors (associations, services, industrial and publishers);
- ☑ Distribution of Free Software inputs and creation of themes proper to Free Software in the competitive clusters;
- ☑ Use of Free Software in the context of technological independence and national security;
- ☑ Use of Free Software in the context of the modernization of systems administration and local authorities.

6. The Stakes for Research

Free Software is a very an important opportunity to allow players in the world of research for new ways of exchange and communication, and to facilitate access to the best work for small companies by promoting synergies in advanced areas.

1. New exchange means

Research has been for long a global field for which computing has become an indispensable tool. Free Software allows for example projects development around HPC (High Performance Computing) by pooling the efforts of development of their source codes and sharing the computing time (grids). In addition, knowledge of operating internal software opens their adaptation to the needs of very vast typology of domains (biology, aerospace, finance, signal processing...).

Free software also allows manufacturers to communicate more effectively the work of their research labs to their clients and to specify with them and other communities common items, tools or standards corresponding to the real needs of future uses. These transparent exchanges are also a guarantee of continuity of the researches financed by national funds and of their free sharing in a perspective of increase in the value of the “public commons”.

Finally, all the SMEs (including a large number of start-ups) can tap into this vast reservoir of knowledge and tools without having to invest heavily, and benefit from the necessary resources for innovation, synonyms of creation of economic value and jobs.

2. Numerous synergies

The software sector is in constant transformation and Free Software must be able to adapt itself as best as it can to these changes. There are numerous research domains for which the synergies between the industrialists and the communities of Free Software are multiple and carriers of promising technological evolutions:

- ☑ Compilers and languages:
  - ➔ Optimization, integration of multi-core processor, generation of cross-code (Java, Ruby, PHP, Python), integration of the real time constraints, lexical closure of the languages...
- ☑ Development tools:
  - ➔ Code refactoring, monitoring, intelligent editors and integration of new peripherals (mobility).
- ☑ Security:
  - ➔ Improvement of hashing (SHA1, MD5...), encryption (AES, IDEA...) algorithms and reservoirs of entropy (“entropy pool”) for generating pseudo-random numbers...
  - ➔ Search and correction of security vulnerabilities...

☑ Interoperability:

➔ Specification and coding of free device drivers for video cards, creation of test batteries for interoperability, integration of standards (Open Document Format)...

☑ OS:

➔ Improving the quality of on board Linux kernel, virtualization of resources, user interfaces, accessibility features (taking into account all the disabilities)...

7. The Software Patent: an Obstacle to Overcome

Faced with the success of Free Software, the defence strategy of the editors (supporters of proprietary solutions) currently focuses on software patents. These players want to hide the fact that the virtually free reproduction of software differs fundamentally from this classic industrial production where the forces of nature are used.

On the other hand, they do not share the position on the non-patentability of ideas, refusing the inference leading from concept to algorithm then to software.

There is also an attitude that is to take out software patents, not to prevent the free use of software covered by these patents, but in order to provide protection (free, of course, we are in the world of Free Software) to users of Free Software against prosecutions by other publishers. This attitude is similar to that of a few organizations such as OIN (see below) that purchase software patents to provide protection on items covered, to the authors of software (Free or not) using the protected components.

The arrival in Europe of software patents would mean that only the big players in the proprietary software would have the legal capacity to treat all problems related to overlapping patents on the many components of new software.

Although the duration of a software patent is limited in time, the mastering of the chain of creation of new patents for new software provides a perpetual hegemony to installed players It would be very difficult or impossible for new entrants to develop and market new software outside the closed ecosystem of the existing club. This would block innovation of European SMEs in favour of a situation of perpetual annuity for companies that would become the owners of the information economy. The challenge is the same as that which led to antitrust laws in the context of industrial capitalism.

8. The Protection against the Threat of Software Patents

Software patents are recognized by the courts of the United States since the 1980s. The industrial players of information technology have all taken out patents on software, as they were accustomed to do on hardware. In this context where any software development risks to infringe countless patents, those who have chosen Free Software decided to neutralize the negative effects of software patents in creating the Open Invention Network (OIN) and in forging defensive alliances. Through this, they attempt to deploy a deterrent shield by purchasing patents and by making them available to Free developers.

However, such a strategy is a difficultly feasible option for the defence of Free Software, because of the cost for obtaining and maintaining a sufficiently consistent portfolio. More importantly, it helps to maintain a patent system incompatible with the software.

In Europe, software patents are theoretically prohibited by law, however, the European Patent Office has granted tens of thousands of them, freely interpreting the law in its favour. In this context of legal uncertainty, the best defence to adopt by actors of Free Software is to unite politically to weigh on legislative initiatives clearly putting an end to questionable practices of the European Patent Office. This is what has allowed in 2005 to defeat a European directive which would have endorsed the legalization of software patents.

List of Acronyms

AMOA: *Assistance à la Maîtrise d'Ouvrage* [Assistance to the Client] i.e. assistance to the client (upstream studies, writing of specifications, for example).

AMOE: *Assistance au Maître d'Oeuvre* [Assistance to the Project Manager] for example for technical expertise.

ASP: Application Service Provider.

CMMI: Capability Maturity Model Integration i.e. method for evaluating software quality.

CMS: Content Management System i.e. a system to manage the editorial content and presentation of a site and publish it on line.

CRM: Customer Relationship Management.

ERP: Enterprise Resource Planning.

GCC: GNU Compiler Collection i.e. the compilers of the FSF, especially for C, C++, Java, FORTRAN, Ada, etc.

ITIL: Information Technology Infrastructure Library i.e. a method based on a set of best practices for the management of information systems.

SOA: Service Oriented Architecture i.e. architecture based on a flexible coupling of the elements of an information system.

SAAS: Software as a Service which is a technology in the form of Web service application service usually local.

SSII: *Société de Service en Ingénierie Informatique* [Service Company in Software Engineering] a generic appellation for service companies of the computing sector, often generalist.

SSLL: *Société de Service en Logiciel Libre* [Service Company in Free Software] i.e. SSII specialized in implementations based on Free Software.

OEM: Original Equipment Manufacturer i.e. a supplier of hardware or software to be pre-installed on a computer for sale.

TMA: *Tierce Maintenance Applicative* [Third Party Applicative Maintenance], consists in subcontracting the maintenance of an application.

Source PAC: *Pierre Audoin Consultants.*

VOIP: Voice over IP, system using standard Internet technology to carry voice (telephony).

About April

Since 1996, April is a pioneer association of Free Software in France. April is a major player in the democratization and the spread of Free Software and open standards to the general public, professionals and institutions in the French-speaking world. In the digital age, it cares also about raising awareness of the dangers of exclusive ownership of information and knowledge by private interests.

The association consists of more than 3 800 producers and users of free software (individuals, companies, associations and local authorities).

APRIL is the preferred partner for the promotion and defence of Free Software.

For more information, please visit the website at: http://www.april.org/, contact us by phone +331 78769280 or email contact@april.org.

APPENDICES

Appendix A) Typology of Free licences

Appendix B) The strategic issues of Free Software

Appendix C) A case study: AdaCore, Free Software publisher

Appendix D) Legal stand on the GNU GPL, contracting and secondary user

Appendix E) Member companies of the April on December 1, 2007

Appendix F) References Webliography

Appendix A) Typology of Free Licences

1. *Definitions and General Principles*

The Definition of “Free Software”

Before addressing licences as such, it is useful to present a definition of “Free Software”. This term is proposed by the Free Software Foundation (FSF), the main body to have stabilized and defined this concept. This definition is independent of any specific licence.

Free software is software that can be used, copied, studied, modified and redistributed without major restriction other than the obligation to give access to source code.

The definition of the FSF is based on four fundamental freedoms, described on the FSF site as follows[6]:

- *The freedom to run the program, for any purpose (freedom 0).*
- *The freedom to study how the program works and adapt it to your needs (freedom 1). For this, access to source code is a prerequisite.*
- *The freedom to redistribute copies so you can help your neighbor (freedom 2).*
- *The freedom to improve the program and publish your improvements to benefit the entire community (freedom 3). For this, access to source code is a prerequisite.*

The Free Software definition is indifferent to the point of gratuitousness or the way the software is issued[7]. It is perfectly possible to sell software and give the source code only to the buyer. However, it is impossible to restrict the rights of the purchaser, including its ability then to disseminate widely the software if he wishes.

Referring to the Free Software definition, is called “proprietary” or “depriving” any piece of software outside the scope of this definition, for whatever reason (restriction on the right to use, unavailable source code...) . The term “commercial software” is to be avoided because it is unclear: a large number of free programs are published in a commercial context (for example, the application server JBoss).

“Open Source” is a similar term and comes from the Open Source Initiative, which promotes Free Software in terms of technical efficiency. OSI offers a definition of Open Source on its site[8] and maintains a list of compatible licences. In practice, the definitions FSF and OSI are very close. The acronym FOSS (Free Open Source Software) is increasingly used or also FLOSS (Free Libre Open Source Software) in the European context.

The Concept of “Copyleft”

Among the various types of Free Software licences, some permit modification and redistribution of the software without constraint, and allow derivatives including “proprietary” derivatives, for example without making available the source code.

Others require that derived software keeps the “Free Software” status of the original software. This feature is known as “copyleft”. This is the main point of debate or misunderstanding on the issue of licences. The purpose of the “copyleft” system is to build a Free Software basis that can only grow, because one can only add software to it, the principle of “copyleft” prohibiting software and its derivatives to get out of this “commons”.

[6] Definition of “Free Software” on the site of the FSF: http://www.gnu.org/philosophy/free-sw.html
[7] FAQ of the GPL on commercial software: http://www.gnu.org/licenses/gpl-faq.html#DoesTheGPLAllowMoney
[8] Definition of “Open Source” on the site of OSI : http://www.opensource.org/docs/definition.php

Typology of Free Software Licences

The classical typology of Free Software licences includes two major licences with or without “copyleft”, and a special case: the public domain. This classification is completed by elements on the compatibility of Free licences between them and the contractual value of the licences:

☑ Public domain: few pieces of software are placed in the public domain, most of the time, in the form of short examples or demonstrations of features. In practice, if an author only writes “distribution in the public domain” in the introduction of software, it is nonetheless covered by copyright.

☑ Licences without copyleft, called “BSD”: the characteristic of these licences is not to oblige to keep the same licence for a derivative work. The code of software using these licences can be incorporated into proprietary software or free software with copyleft. There are several very close licences in this family, the BSD licences themselves, for example the licence of the Free BSD[9] project or that of PostgresSQL[10], but also the X11 licence used in the X11 GUI, the Apache[11] licence of the Apache Software Foundation (Apache web server, Tomcat), the French licence CeCILL -B[12]. Some of these licences have specificities coming from their long history which can make them difficult to use in a new project. However, current projects have adopted so-called “modified” BSD licences (compared to the original BSD licence) simplifying their integration.

☑ Licences with copyleft: these licences require that derived software retains its status of Free Software, including the provision of the source code for the modified version. In practice, this means that it is not possible to distribute a proprietary software incorporating code using such a licence. The main licence of this type is FSF's *GNU General Public License*[13] (GPL). It is used by the Linux kernel and the GNU project, including the Gnome environment, the compiler GCC and the main utilities of a distribution called GNU / Linux, and since a few months the Java implementation of Sun Microsystems . Other licences of this type are FSF's *Lesser General Public License* (LGPL), which offers the possibility to dynamically link the program to a proprietary application. This licence is typically used for libraries, like the C library of the GNU project. The LGPL is particularly used by the OpenOffice.org project, most developments in the ObjectWeb consortium, and JBoss project. The CeCILL V2[14] licence is a French licence compatible with the GPL, proposed by CEA, CNRS and INRIA. Finally, particular licences like the *Common Public License* (original IBM) or the Eclipse Public Licence[15](Eclipse) have a “copyleft” functioning.

It should be noted that it is possible to incorporate software in an internal development without worrying about the licence or “copyleft” status of the various components if no distribution is envisaged. However, it is desirable to know the status of the various components where the question of dissemination could arise one day (reuse of the components by a subsidiary company, for example).

Two details on the GPL:

> The freedom to run the software is a right granted by the software's right owners (under copyright) that does not requires the acceptance of a contract by the user. See the FAQ for GPL[16] or even an explanation from the FSF lawyer[17].
>
> The obligation to provide the source code applies only for third parties to whom the software is released. For example, there is no need to explicitly communicate the changes to the original authors

[9] The Free BSD copyright: http://www.freebsd.org/copyright/freebsd-license.html

[10] PostgreSQL copyright: http://www.postgresql.org/docs/faqs.FAQ.html#item1.3

[11] Apache *License* version 2.0: http://www.apache.org/licenses/LICENSE-2.0.html

[12] CeCILL: French licence in free software: http://www.cecill.info/licences.fr.html

[13] GNU General Public *License*: http://www.gnu.org/copyleft/gpl.html

[14] CeCILL licence [Ce(A)C(nrs)I(NRIA)L(ogiciel)L(ibre)]: http://www.cecill.info/licences.fr.html

[15] Eclipse Public Licence: http://www.eclipse.org/legal/epl-v10.html

[16] FAQ of the GPL about the acceptance of the *license*: http://www.gnu.org/licenses/gpl-faq.html#ClickThrough

[17] Article by Eben Moglen about GPL, page 2 : “*The license does not require anyone to accept it in order to acquire, install, use, inspect, or even experimentally modify GPL’d software*”. http://moglen.law.columbia.edu/publications/lu-12.pdf

or to third parties non recipient of the amended[18] software.

In a nearby field, there is licensing oriented "multimedia content" or "documentation". We can cite the GNU FDL[19] and Creative Commons licences. These licences are not considered any further in this document, because it is unlikely to find software using these licences that are not adapted to source code. For example, a code under Creative Commons can easily be inconsistent with the definition of Free Software (restrictions on the changeable parts, restrictions on commercial distribution, etc..)

Compatibility of Free Licences

In the context of proprietary software, where two licences confront, lawyers sit around a table and suggest that entrepreneurs sign the result of their cogitations. In the world of Free Software, where the mixture is the rule, such an expenditure of energy would paralyse the actors. The compatibility of licences is unique to Free Software: it requires particular attention from licence writers and is subject to regular studies. To avoid that any assembly of two pieces of software requires the presence of lawyers, the issue is resolved upstream.

From the perspective of the using company that does not intend to distribute any software distribution, this question may seem secondary. However, knowing the limits of compatibility is useful to assess potential contribution to external Free Software projects, and anticipate the difficulties of maintaining a product based on multiple components.

In principle, it is possible to incorporate the code of non-"copyleft" Free Software, for example of BSD type, to a project of "copyleft" type, for example GPL. The opposite is not possible. There are however many incompatibilities. In theory, the compatibility of each licence must be considered with all other licences. In practice, this systematic work is only done by the FSF, which gives priority to licences it has written.

The main "copyleft" licence (more than 65% of Free Software according to freshmeat.net), the GPL, is sensitive to additional constraints imposed by other licences. For example, clauses relating to the sale of rights on patents make the Eclipse licence incompatible with the GPLv2. Therefore:

☑ Some projects have changed the terms of their licence (whether "copyleft" or not) for a licence compatible with the GPL. For example, we have to examine carefully if we are facing a so-called new generation BSD licence or a CeCILL version 2 licence, or an *Artistic License 2.0* (Perl). The FSF maintains a list of licences without copyleft compatible with the GPL[20]. In practice, over the last several years, projects using the licences of the BSD family (MIT, X11,...) have all moved their software to compatible versions.

☑ Many major projects tend to use a dual licensing system to allow code-sharing with the GPL ecosystem. This means that the user can choose for any given component, the licence that best suits its use among several licences. For instance, the Perl project is in dual licensing *Artistic License* and GPL; Firefox gradually migrates its code from the *Mozilla Public License* (MPL) to the GPL; Sun Microsystems gradually makes some of its developments converge to the GPL (including Java and OpenSPARC processors), while some remain with licences without "copyleft" (for example OpenSolaris, licensed CDDL); the Mono project (. NET for the Linux world) has three licences: the C # compiler is GPL, the *runtime* is LGPL and libraries are MIT (a BSD type licence); Qt (development environment used by KDE) is dual licensed QPL and GPL; the QPL is itself a licence allowing the use of various licences for the application developed... Finally, the MySQL and Asterisk projects use a proprietary dual licence model and the GPL.

☑ The recent licences of "copyleft" type, explicitly present themselves as compatible with the GPL. This is the case of the CeCILL v2 licence and of the recent European Union Public Licence

[18] FAQ of the GPL about redistribution: http://www.gnu.org/licenses/gpl-faq.html#GPLRequireSourcePostedPublic
[19] GNU FDL : GNU Free Documentation License, license regarding copyright by Free Software Foundation used for writings (documentation, texts...)
[20] List of free *licenses* on the site of the FSF: http://www.gnu.org/licenses/license-list.html

(EUPL v.1.0)[21]. The latter, which is used widely in Europe, has even planned a special evolutionary mechanism to stay consistent with key "copyleft" licences. Currently, the EUPL is compatible with GPLv2, *Open Software License* v2.1 and v3, *Common Public License* v1.0, Eclipse Public Licence v1 and CeCILL v2.

☑ New version of the GNU GPL: June 29, 2007, the FSF released version 3 of the GPL (GPLv3). This update does not alter the fundamental characteristics of the GPL, but GPLv3 is not compatible with the GPLv2 in the sense that mix of GPLv2 and GPLv3 code is not feasible; it is then necessary to move the whole to version 3. It should be noted that a statement is present in almost all GPLv2 software leaving the user the freedom to replace the GPLv2 by a later version of the GPL (in this case GPLv3), which facilitates progressive software updating towards the GPLv3. The same principle applies to the LGPLv3 issued on the same date to replace the LGPLv2. The specifics of GPLv3 are presented below.

Conclusion: the compatibility of licences may seem problematic as a whole. However, major projects are working to propose solutions for convergence towards the most-used licences, including GPL. In the context of an enterprise information system, software integration is most often at the interfaces between pieces of software. It is then possible to avoid mixing of the source codes of different licences.

Licence or Contract

One conceptual complexity of the GPL is that it claims not to be a contract, which may pose understanding problems in Europe. For the U.S., the GPL is mainly based on *copyright* , while in France, and in many countries in Europe, the habit is to license software by contracts. The main differences are the fact that a contract must be accepted by both parties and the terms of responsibility inherent to software contracts. The lawful use of a program under the GNU GPL does not impose a contractual relationship between the user and the copyright owner. An analysis of this peculiarity is presented in Annex D, "Legal Point of View: GNU GPL, contracting and secondary user".

This difference of appreciation and the desire to have a licence of French or European origin, based on a contractual mechanics, are the initial impulsion for drafting the EUPL CeCILL licences.

However, the validity of the GPL has been repeatedly been confirmed by European courts.

2. Foreseeable Evolutions

The Emergence of the EUPL and CeCILL Licences

The new EUPL licence, approved in January 2007 by the *Interoperable Delivery of European eGovernment Services to public Administrations, Business and Citizens* program of the European Commission (IDABC), aims to give the EU an effective tool for its projects. And on the one hand, the need for a local equivalent of the GPL seems present in many countries (including France), and on the other hand, the European Commission has a strong incentive capacity through European research projects. One can therefore imagine a bright future for this licence in the next few years.

In parallel, the French initiative CeCILL seems a real success for projects from INRIA, the CNRS or central governments. Observing the statistics of the forge of ADULLACT[22] (projects directed towards local authorities) one can notice that the GPL is still ahead: about 105 GPL, 5 LGPL, 25 CeCILL licences and 25 BSD licences. On the Admisource[23] forge (oriented toward central governments), there are 7 projects in

[21] European Union Public license (EUPL v.1.0): http://ec.europa.eu/idabc/en/document/6523

[22] *"Association des développeurs et des utilisateurs de logiciels libres pour l'administration et les collectivités territoriales"* [Association of the Developpers and Users of Free Software for Civil Service and Local Governments]

[23] *Admisource* : http://admisource.gouv.fr

CeCILL licensing for 8 projects under GPL. For its part, the ObjectWeb[24] consortium now favours the LGPL for most of its developments, including the Jonas application server.

The Move from the GPL Version 2 to the GPL Version 3

The GPL version 2 dating from 1991, evolving laws and technologies led the FSF to update the GPL: since June 2007, the current version of the GPL is the GPL version 3. Nevertheless, the GPLv2 remains valid. The rewrite of the GPL does not change its basic principles, updates concern the following aspects:

☑ Clarification and internationalization: the new wording of the licence is intended to facilitate an identical interpretation in all legal systems through a precise definition of terms. Technical concepts such as the spread of software have been adapted to the state of the art.

☑ Specific protection against the phenomenon called "*tivoisation*", i.e. the establishment of a physical lock making it impossible to change a GPL piece of software (for example in an on-board system). This type of blocking is no longer allowed in the context of a product for the general public. The lock is still possible within an organization or among professionals, which practically means that the recipient must accept the locking of the program by a physical means for the licence to be respected.

☑ Explicit protection against software patents: to distribute or to contribute to a GPL software now means to yield one's rights on any possible patents covering the contributed code. In practice, this consists in prohibiting dissemination of a code the use of which is restricted by a patent. This protection is accompanied by anti-discrimination clauses to reduce the scope of agreements favouring one class of users compared to others (e.g. clients of a particular supplier).

☑ In some jurisdictions, it is not legal to change a Digital Right Management device (DRM). The GPLv3 therefore explicitly mentions that the author of the software covered by the licence renounces to his software being considered a DRM.

☑ Modularity and compatibility: GPLv3 includes a modular system allowing extensions for certain elements: additional permissions and contractual extensions, clauses of guarantee and accountability, provisions on trademarks... This system allows the GPLv3 to be now compatible with the Apache licence.

It is too early to assess all the consequences of the transition from GPLv2 to GPLv3 on the ecosystem of Free Software, but the prevailing view among the actors in the field is:

☑ The entire GNU project, including building blocks as the C library, quickly will migrate to the GPLv3.

☑ The Linux kernel will remain for long GPLv2 (it contains no clause to facilitate the transition to a later version of the GPL). However, the consequences are limited, Linux is technically independent from the rest of the GPL ecosystem.

☑ Most other GPLv2 projects will gradually migrate to GPLv3, particularly because of their dependency cascading dependencies on other components GPL.

☑ The major market players in charge of major projects using the GPL (Sun Microsystems, Red Hat...) could follow quickly.

☑ Some market players having used in the past flaws of the GPLv2 (Tivo, Novell) should review some of their commercial agreements or remain confined in GPLv2.

Ultimately, if this new version will be important for those involved in the offer, it should be transparent for individuals and major users. The provisions of GPLv3 on DRM are a conceptual clarification without practical consequence; they record the known incompatibility between the constraints of blocking of usage control

[24] Licences of the ObjectWeb projects: http://solutions.ow2.org

software and the opening of the code of free software.

The consequences of the new version of the GPL on the evolution of the CeCILL and EUPL European licences remain to be studied.

Appendix B) Free Software: What Is at Stake?

Document written by the April on the occasion of the launch of the *Pacte du Logiciel Libre* for the Legislative Elections in 2007, in the frame of the initiative *Candidats.fr*.

The concept of Free Software sends us back to the early 1980's. At the time, some researchers devoted both to the idea of collaboration in code development and to the culture of scientific publication (that was then the norm in computing) decided to legally secure their work and practices.

They wrote licences cleverly using copyright laws, in order to share their software with all human beings while protecting them against any appropriation attempt. Eben Moglen, one of the redactor of the most popular Free Software licence – the GNU GPL – often declared that his licence allows "*the creation of a common collection to which anybody can add and to which nobody can remove anything*". The other author, Richard Stallman, likes to remind us that Free Software can be defined by three words: "*Liberty, Equality, Fraternity*".

In practice, since Free Software can be freely used, copied, modified and redistributed lawfully by anybody, it has been spreading a lot during the last twenty years and the number of contributors to the common collection such software constitutes is constantly increasing.

Today, the movement introduced by a few researchers has become a recognized phenomenon of society. Millions of users (individuals, NGOs, companies, administrations...) make use of the liberties that come with Free Software. A quality offer exists, thanks to the cooperation between these users over the Internet. They represent more and more a direct competitor for the dominant actors of proprietary computing, such as Microsoft.

Pieces of software such as the GNU/Linux operating system, the Firefox browser or the Open-Office suite are well-known examples of Free Software which, day after day, become more popular, and are commonly used in businesses and at home. In addition, the Internet's reliability is based on Free Software since its beginning, the same way the reliability of some stock-exchanges (BNP Paribas) and planes (Rafale, Airbus 380) is. Free Software can be found in electronic devices (such as domestic Internet gateways...) but also in ATMs, mobile phones or PDAs.

1. A Societal Issue

Free Software can be copied legally by anybody, it can almost always be downloaded from the Internet. This free of charge access, allows the less privileged populations to avoid resorting to illegal copying of software in order to benefit from technological progress. Free software is intrinsically a tool to reduce the "digital divide".

Since Free Software is shared with its source code, this code can be studied to understand the techniques it implements so they can be reused and transmitted, including outside of the usual training and educational fields. The way free software developers cooperate through the Internet simplifies the transfers of expertise beyond borders.

Free Software is not merchandise and those who develop it are contributing to the transmission to many people of scientific knowledge, technical expertise and technologies allowing them to access to Knowledge. Moreover, he GNU project – a key project for Free Software – has been classified by UNESCO as a "World Treasure".

2. An Economic Issue

Free Software allows the development of a dynamic economy referred to as “coopetition”, in which the actors share part of the R&D costs and compete on the services based on common generic blocks. In the US, in Europe, in Asia, a real service-oriented economy has been created around free software.

The companies from that sector make a living on the support, training, integration, consulting and on specialization of generic blocks. The organizations making use of such software adhere to its model for the quality offer, but also because Free Software confers them more independence and a better control over the costs of maintenance and internal development.

The number of companies using Free Software steadily increases and, everywhere in the world, vast sections of States' and local governments' information systems migrate to Free Software.

3. A Strategic Issue

Free Software is more and more seen by States and political leaders as:

☑ a way of enforcing sovereignty and industrial policies;

☑ a means to control public spending;

☑ a means to achieve sustainable development.

Several projects illustrate this trend:

☑ the development of secured operating systems for military purposes (China, US, France...);

☑ the political will to develop a national industry around Free Software (“Orient Ware” consortium between China and Europe, project of French “competitiveness area” around Free Software...);

☑ the developments around electronic administrations (Free Software was made compulsory in Brazil and in the Netherlands, Internet income statement in France...) and the migration of French members of parliament to a free operating system;

☑ the ever-growing use by developing countries (such as the actions led by the “University Agency for French-Speaking” [*Agence Universitaire de la Francophonie*]...)

As a conclusion, here is a quote:

> "*The State services often use software whose source code isn't available, which prevents these States both from correcting the errors that the suppliers would refuse to correct themselves and from verifying the absence of security threats in sensitive pieces of software. Sometimes, without knowing it, the States' services use software which secretly transmits confidential information to foreign companies or bodies. But the economic model of the software and the telecommunication industries, pushed by the market, is mainly based on the appropriation of clients and the exponential exploitation of the users' profile. These economic models encourage strategies of incompatibility, industrial secrecy, programmed obsolescence and violations of individual freedom. If the French State, because of the communication networks being transnational, cannot claim to be able to eliminate by law such deep-seated deviations it can anyhow help the development in France of an information society respecting public liberties, consumer security and healthy competition, and hope to be regarded as a forerunner in Europe and in the world.*"
>
> (Excerpt of the reasons of a French law proposal made by PS members of parliament in 2000[25], followed up in 2002 by French UMP senators[26].)

[25] http://www.assemblee-nationale.fr/11/propositions/pion2437.asp

[26] http://www.senat.fr/leg/ppl02-032.html

A Few Figures

☑ The total amount of decent quality free software is worth at least **€12,000 million**. This represents at least **131,000 person.year**, or an annual contribution of €800 million **half of which comes from developers based in Europe**. In terms of service, free software could represent **up to 32% of the market of computing-related services in 2010.** ("The impact of Free/Libre/Open Source Software on innovation and competitiveness of the European Union", January 2007, available on the European commission's website)[27]

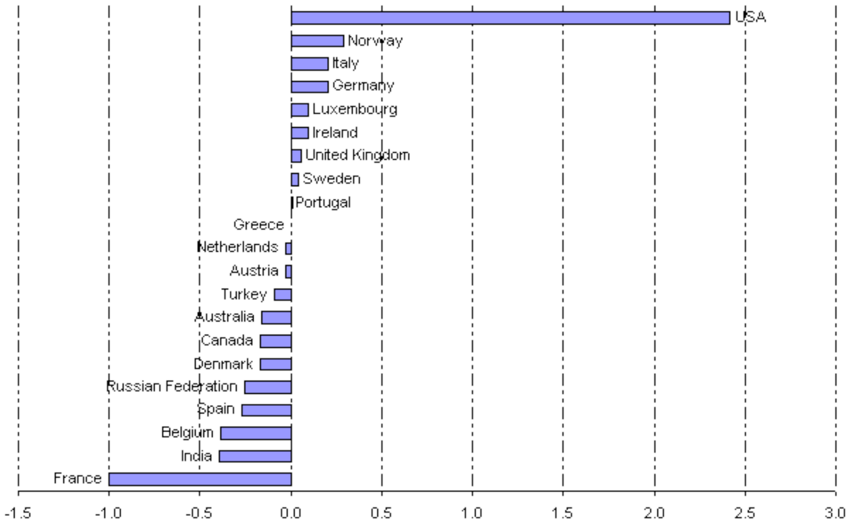
☑ In France, in 2004, **80% of companies in the CAC40 [French Dow-Jones] and 56% of administrations** and of local public governments were using Free Software (PAC, 2004), **67% of companies** were using Free Software, 16% were considering to do so in 2005 and only 17% had no planned use whatsoever in the short term (InformationWeek, November 2004).

☑ In 2005, nearly **90% of companies were considering migrating their Windows Server to Linux** in the upcoming year (InformationWeek, Research Brief "Linux Outlook", February 2005) and **70% of all web servers in the world** were free servers(Netcraft, November 2005 based on 74 572 794 web site).

☑ In France, in 2006, including the labour expenses, the French administration has spent 9% of its computing budget on free software developments and free platforms whereas it had spent only 6% in 2005. This represents €900 million and **15% of the total market of IT services** for the French administration, with **a 12% per annum** growth expected during the next two years (Markess International, April 2006).

☑ In France, in 2007, the turnover of the Free Software industry reached €730 million, which represents a 66% increase compared to 2006 (figures from Pierre Audoin Consultant).

☑ Here is the ratio of immigration/emigration of free software developers: **France is by far the last** though it is amongst the countries where student-engineers contribute the most to Free Software projects (see 3).



[27] http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf

Annexe C) A case study: AdaCore, Free Software Provider

Introduction

AdaCore has built its success and that of GNAT Pro, its development environment for the Ada language, on the concept of "service with leverage effect". Such a service is defined as a value-added service based on a free software[28], the provider of which knows the details and is fully involved in its developments. The more interest there is in the software, the greater the leverage.

AdaCore offers access to the GNAT Pro service as an annual subscription including not only the provision of development tools and their regular updating, but also access to a high-level on-line support because directly provided by the developers of these tools. The software is made available under a licence adapted to the industrial context.

GNAT Pro and AdaCore

Designed to make critical and secure applications , the Ada 83 language has built over the years a reputation of sturdiness and reliability. In the early 1990s, there has been a revision in the name of Ada 9X which culminated in 1995 in Ada 95 standard. It extended the Ada 83 adding object programming and hard real-time features. Since then, a new ISO standard has been adopted under the name of Ada 2005. It pushes even further the facilities offered by Ada in these areas.

During the development of Ada 9X, the Ada group at the University of New York has been charged by the *U.S. Department of Defense* to develop an Ada 95 compiler under Free licensing. This project is at the origin of GNAT Pro, and it is from this project that were born Ada Core Technologies in the United States, and in Europe, AdaCore (initially ACT Europe), two sisters companies that form a single operational and commercial entity under the name of AdaCore.

Today, AdaCore proposes around GNAT Pro, now the reference in this field, Ada and C development solutions for a large number of configurations, ranging from native software on PC to safety critical on-board software.

An All Included Service

GNAT Pro is marketed as an annual subscription that includes:

- The provision, for selected platforms, of development tools under the GPL, and libraries licensed GMGPL[29], which ensures that the programs generated with the GNAT Pro tools will be distributed according to the conditions set by the customer.
- Regular updating of GNAT Pro tools.
- A help desk to use these tools, provided directly by the developer of the tools.
- A more general on-line consultancy service assured by Ada experts.

A Profitable Service

AdaCore has been a pioneer in adopting an early subscription business model - as opposed to buying traditional licences. Properly applied, this model generates a regular turnover that smooths the cash inputs

[28] Here, we use the term "free software" as "software meeting the criteria of the Open Source Initiative". (http://www.opensource.org/).

[29] Code licensed under the GMGPL [GNAT Modified GPL] may be linked to code under an other licence without requiring for the resulting software to be under GPL.

and make them predictable. It allows for example companies on the stock exchange to assess more easily their profits, a decisive advantage in a world where security of investment is paramount.

A Win-Win Service

This model also has the advantage of matching the best interests of customers and those of the service provider.

In the context of a critical application development, putting at stake the human life or dealing with financial transactions, acquisition and implementation of development tools is only a starting point: above all, the project manager wants flawless support and assistance throughout his project.

To achieve the renewal of existing subscriptions and expand the use of GNAT Pro to new projects, AdaCore's interest is to ensure the success of its clients. The result is an alignment of interests that makes the success of this model.

The customer is also in the same position as if he had developed himself the components of GNAT Pro: he not only has directly access to limitless source code, but also to the developers of the product, the GNAT Pro experts thus becoming partners in the client's team.

If for any particular reason, AdaCore no longer provides a satisfactory service, customers could turn to another supplier or assure it internally. This is an important guarantee in an industry where the project duration is often measured in decades.

This commercial model fits well the contracts of AdaCore, whether defence, aeronautics or space, since it allows a reduction of the risks inherent to them.

Serving Free Software

Though nothing requires it, AdaCore offers free versions of the GNAT Technology through a dedicated website (http://libre.adacore.com/) established almost 7 years ago, and regularly updated since new versions.

The compiler included in GNAT Pro is based on GCC, a GPL compiler. In return, naturally, AdaCore offers to the FSF (Free Software Foundation) the developments it has made.

De facto, this relationship allows involving clients and members of the Free Software community in the development strategy of AdaCore, and to some extent, allows them to orient it. Thus, the publication of GNAT resulted in an increase in its notoriety and in the promotion of the development of a community with which AdaCore was able to maintain close relations, something which would be much more difficult in a proprietary model.

The development of GNAT thus benefits from comments made by customers and the community. The problems are taken into account by the AdaCore team and are the subject of fixes included in the following versions; the development of new functions and porting to other platforms is often made according to the feedback of these same people. In the end, 95% of AdaCore's product developments come from interactions with customers and the open source community.

Even better: awareness of GNAT in the open source world has attracted AdaCore on both sides of the Atlantic, a team of talented engineers wishing to add their bit.

Conclusion: an Efficient Model for Free Software

To ensure regular income, AdaCore makes sure that customers are satisfied with their services and they renew their subscription. This convergence of interests is at the heart of the business model of AdaCore.

The GBAT experience has proven to AdaCore that it was easier to achieve this goal through an opening to its customers and to the open source community rather than with a locked system and proprietary licences. So this open model that provides a year on year growth for AdaCore. Without doubt it may accompany, more generally, the growth of Free Software.

Appendix D) Legal Point of View: GNU General Public *License*, Contracting and Secondary User

Presentation of View:

The lawful use of a program under the GNU GPL does not require a contractual relationship between the user and the copyright owner. Using a program means for example sit at the computer of a friend and tap a key on the keyboard at random. The key sends the active program at this time a character, thus the person becomes user of the program. Ditto when using a search engine on the Internet, when following a dynamic link when consulting a computer in a public place or when watching a tutorial screen above the shoulder of someone. Every citizen of the Information Society is the user of many programs, most of which are beyond his perception. The illusion of monolithic object given by interfaces actually hides an intangible software network connecting myriads of components, each covered by copyright. We will show how the GNU GPL provides a non contractual space that allows the use of software by all.

Legal Issue:

For the purposes of argument, we define the “secondary user” as the person who is allowed to use free software to the exclusion of any act of copying, modification, adaptation and distribution; thus it differs from the first user.

Legal Premise:

For the purposes of analysis, we start from the premise that the GPL is not contrary to French law. While it is recognized that the GPL is perfectly valid in the light of French law, the legal status of a user would be the following.

Legal Analysis of the Situation of a Secondary User:

The willingness expressed by the GPL licensor is undoubtedly to promote the use of Free Software. Unlike the common law licences, which aim at a private allocation of rights to use and enjoy, the Free licence or “copyleft” on the contrary encourages wide dissemination or distribution of Free Software by allowing the licensee to permit the use of the software by unlicensed people. It is precisely one of the purposes of the GPL.

In order to allow wider use, the licensor therefore uses its economic property rights by offering the licensee the freedom to spread the free software to friends, colleagues or any other person.

The secondary user of free software is a third party to the licensor/first user relationship.

By the authorization stipulated in the GPL, the secondary user is in a special legal status:

☑ He has no contractual relationship with the licensor as he did not consent to the licence and most often does not know the licensor, author of the Free Software. It seems not possible to assert any implied consent to the conclusion of a licence and particularly since the French law does not allow tacit agreements in this matter;

☑ In the same way, it is not in the will of the user and the secondary user to agree on a sub-licence. Indeed, according to the case law Civ. 1st, 13 October 1993, D. 1994. II. 166, note P-Y. Gautier, even if a sub-licence does not require the formality required by Article L. 131-3 of the *Code de la propriété intellectuelle* [Code of Intellectual Property], it seems impossible to conclude that the physical act of using the software would be a manifestation of consent both of the licensee and the secondary user, in accordance with articles 1108 et seq. of the *Code Civil*;

☑ Consequently, the secondary user has the right to use the software without risking any legal action of a contractual nature by the licensor.

The secondary user gets the right to use from the licensee.

Without legal relationship to a contract with the licensor, the secondary user gets the right to use from the licensee, it appears as a beneficiary of the licensee. It therefore could be a secondary licence agreement derived from licence agreement settled between the licensor and the licensee or primary user.

Despite all that, the secondary user is not absolved of any legal obligation...

If the use of the software does not constitute misconduct in the legal sense, it is necessary that the secondary user acts as a “good father” in that sense that the secondary user must not cause any harm neither to the licensee who is the main user, nor to the author of the software granting the licence. Otherwise, the user could see its tort liability involved on the grounds of Article 1382 of the *Code Civil: « Tout fait quelconque de l'homme, qui cause à autrui un dommage, oblige celui par la faute duquel il est arrivé, à le réparer. »* [Any act of man, causing harm to others, requires for who did it to fix it]. This would include the assumption of a violation of the author's moral right that we know as of public nature.

... nor deprived of any right

For example: even if there is no legal contractual relationship between the licensor author of the software and secondary user, the fact remains that the articles 1386-1 and following of the *Code Civil*, on the responsibility of manufacturers for the introduction of defective products, could serve as a basis for legal action initiated by the secondary user against the author or distributor of the Free Software causing him harm.

Appendix E) Member Companies of the April on December 1st, 2007

ADACORE
AL'X COMMUNICATION
ALCA TORDA CONSEIL
ALCÔVE
ALTER WAY
AMBIKA
ANASKA
ANTI-BUG-ASSISTANCE
ARS APERTA
ATREAL
BEARSTECH
CLISS XXI
CONOVAE
DALIBO
DBM TECHNOLOGIES
ÉDITIONS EYROLLES
ÉDITIONS O'REILLY
EMENCIA
EMJ: *Jardins et loisirs* (Gardens and leasure)
EXEMOLE
FON
FREE ELECTRONS
GOOGLE France
IDEIA
IKOULA
IN LIBRO VERITAS
INFOCLIP
INL
ITAAPY
KELIS
LIBRE ENTREPRISE (network of 17 companies)
LINAGORA
LOGIDÉE
LOGILAB
LOST OASIS
MANDRIVA
MEKENSLEEP
MERETHIS
MOSTICK
NETAKTIV
NEUF CEGETEL
ON-X
OPENFARM
OPENSIDES (Belgium)
PI-LYON
PILOT SYSTEMS CONSULTING
PMB SERVICES
PORTANEO
PORT PARALLELE
PRAGMATIC SOURCE
PROCESS ONE
PROFORMATIQUE
RED HAT France
RYXÉO
SILOH
SOLUTIONS Linux
STARXPERT
STERIA
SUN MICROSYSTEMS
THALES
TRANQUIL IT SYSTEMS
WALLIX
WENGO
ZEFYRIS
ZEND.

Annexe F) Webliographic References

Admisource
http://admisource.gouv.fr

Apache *License* version 2.0
http://www.apache.org/*license*s/*LICENSE*-2.0.html

CeCILL French licence in free software
http://www.cecill.info/licences.fr.html

European Community: « *European Impact of FLOOS on Innovation and Competitiveness of the EU ICT sector »*
http://ec.europa.eu/enterprise/ict/policy/doc/2006-11-20-flossimpact.pdf

European Council of Lisbonne
http://ue.eu.int/ueDocs/cms_Data/docs/pressData/fr/ec/00100-r1.f0.htm

Eben Moglen
especially article about the GPL
http://moglen.law.columbia.edu/publications/lu-12.pdf

European Union Public Licence (EUPL v.1.0)
http://ec.europa.eu/idabc/en/document/6523

Eclipse Public *License*
http://www.eclipse.org/legal/epl-v10.html

Framasoft
http://www.framasoft.net

Free BSD copyright
http://www.freebsd.org/copyright/freebsd-*license*.html

Free Software Foundation
http://www.fsf.org/
FSF France
List of free *licenses*
http://www.gnu.org/*license*s/*license*-list.fr.html

FSF/UNESCO Free Software Directory
http://directory.fsf.org/

GNU
http://www.gnu.org
see specially the GNU General Public *License*
http://www.gnu.org/copyleft/gpl.html

IDABC (I*nteroperable Delivery of European eGovernment Services to Public Administrations, Businesses and Citizens*)
http://ec.europa.eu/idabc/en/home

ObjectWeb
Licences of the Objectweb project
http://solutions.ow2.org

Open Source Initiative
http://www.opensource.org

PostgreSQL copyright
http://www.postgresql.org/docs/faqs.FAQ.html#item1.3

Proposition de loi tendant à généraliser dans l'administration l'usage d'Internet et de logiciels libres
Bill aiming at generalizing the use of free software and internet in public administrations
http://www.senat.fr/leg/ppl02-032.html

Sourceforge
http://www.sourceforge.org

Licence Gnu Free Documentation License

0. PREAMBLE

The purpose of this license is to make a manual, textbook, or other functional and useful document « free » in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or non commercially.

Secondarily, this license preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This license is a kind of « copyleft », which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public license, which is a copyleft license designed for free software.

We have designed this license in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this license is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this license principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This license applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this license. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The « Document », below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as « you ». You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A « Modified Version » of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A « Secondary Section » is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics). The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The « Invariant Sections » are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this license. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The « Cover Texts » are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this license. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A « Transparent » copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent.

An image format is not Transparent if used for any substantial amount of text. A copy that is not « Transparent » is called « Opaque ».

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The « Title Page » means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this license requires to appear in the title page. For works in formats which do not have any title page as such, « Title Page » means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section « Entitled XYZ » means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as « Acknowledgments », « Dedications », « Endorsements », or « History »). To « Preserve the Title » of such a section when you modify the Document means that it remains a section « Entitled XYZ » according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this license applies to the Document. These Warranty Disclaimers are considered to be included by reference in this license, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this license.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or non commercially, provided that this license, the copyright notices, and the license notice saying this license applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this license. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you *distribute a large enough number of copies you must also follow the conditions in section 3.*

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition.

Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material.

If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

*It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an upd*ated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this license, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this license, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this license.

I. Preserve the section Entitled « History », Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled « History » in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the « History » section.

You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled « Acknowledgments » or « Dedications », Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgments and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled « Endorsements ». Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled « Endorsements » or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice.

These titles must be distinct from any other section titles.

You may add a section Entitled « Endorsements », provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this license give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this license, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this license, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number.

Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

*In the combination, you must combine any sections Entitled « History » in the various original documents, forming one section Entitled « History »; likewise combine any sections Entitled « Acknowledgments », and any sections Entitled « Dedications ». You must delet*e all sections Entitled « Endorsements ».

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this license, and replace the individual copies of this license in the various documents with a single copy that is included in the collection, provided that you follow the rules of this license for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this license, provided you insert a copy of this license into the extracted document, and follow this license in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an « aggregate » if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit.

When the Document is included in an aggregate, this license does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form.

Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4.

Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this license, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this license and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this license or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled « Acknowledgments », « Dedications », or « History », the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this license. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this license. However, parties who have received copies, or rights, from you under this license will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS license

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation license from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the license is given a distinguishing version number.

If the Document specifies that a particular numbered version of this license « or any later version » applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this license, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this license for your documents

To use this license in a document you have written, include a copy of the license in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR your NAME.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation license, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled « GNU Free Documentation License ».

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the « with...Texts. » line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains non-trivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public license, to permit their use in free software.