About the “Ayrault Memorandum” and this translation

The “Ayrault Memorandum”

The “Ayrault Memorandum” (circulaire Ayrault, in French) is a document, signed in September 2012 by the French Prime Minister, presenting guidelines and recommendations on the proper use of Free Software in the French administration. The document was mainly produced by the DISIC (the Department of Interministerial Systems Information and Communication) and the CIO of some departments. The DISIC, founded in 2011, is in charge of coordinating the actions of the administration in the area of information systems.

The following translation is not an official translation. The official text is the French version[1]. You can also read our press release about the Ayrault Memorandum "Ayrault Circular: progress for the use of Free Software in the French administration, pending the legislative part"[2]. This translation is also available online[3].

Translators' note

This translation was done by April with the help of: Jeanne, Fred, janchou, luc, rowanthorpe, echarp, François, Gijs&Clementine, strangeattractor, kult, echarp, Cloé, Marc C., Thibz jhopkins, karel, olasd, Clémence… Thank you everybody.

We tried to do a close translation of the text. The original document has a number of words and phrases that we recommend avoiding, or avoiding in certain contexts and usages. Some are ambiguous or misleading; others presuppose a viewpoint that we disagree with. The two main problematic words are "intellectual property"[4] and Linux instead of GNU/Linux when the document refers to the complete system[5]. You can read "Words to Avoid (or Use with Care) Because They Are Loaded or Confusing"[6]. The French document uses the word “souche” or the expression “souche libre” which is a bit difficult to translate into English. In most case, “souche” or “souche libre” simply refers to free software.

Acronyms

- DISIC: Direction interministérielle des systèmes d'information et de communication / Interdepartmental Directorate for ICT (ou Department of Interministerial Information Systems and Communication)
- CTSIC: Comité Technique des Systèmes d'Information et de Communication / the information- and communication-systems' technical committee
- CTSIC/CSIC: Comité Technique des Systèmes d'Information et de Communication / Conseil des Systèmes d'Information et de Communication -> the information- and communication-systems' technical committee / information- and communication-systems council
- SSLL: Sociétés de Service en Logiciel Libre / Free Software Services Companies

1 http://circulaire.legifrance.gouv.fr/pdf/2012/09/cir_35837.pdf
2 http://www.april.org/en/ayrault-circular-progress-use-free-software-french-administration-pending-legislative-part
3 http://www.april.org/en/french-prime-minister-instructions-usage-free-software-french-administration
4 http://www.gnu.org/philosophy/words-to-avoid.en.html#IntellectualProperty
5 http://www.gnu.org/gnu/linux-and-gnu.en.html
6 http://www.gnu.org/philosophy/words-to-avoid.en.html

- ADULLACT: Association des développeurs et utilisateurs de logiciels libres pour les administrations et les collectivités territoriales / French association of Free Software developers and users in local and central government
- DGFiP: Direction générale des finances publiques / Public Finances Directorate General
- SAE: Service des achats de l'État / French State Procurement Agency
- CCAG TIC: Cahier des clauses administratives générales applicables aux marchés publics de techniques de l'information et de la communication / General administrative clauses applicable to public procurement of information technology and communication (ICT GCC)
- ANSSI: Agence nationale de la sécurité des systèmes d'information / French Network and Information Security Agency
- MCC: Ministère de la Culture et de la Communication / Ministry of Culture and Communication
- MEDDE: Ministère de l'Écologie, du Développement durable et de l'Énergie / Ministry of Ecology, Sustainable Development and Energy
- MI: Ministère de l'Intérieur / Interior Ministry
- TCI: chantier Transformation des Centres Informatiques (l'objectif est de mettre en commun des méthodes et des outils d'exploitation, voire des capacités d'hébergement) / Data Center Transformation

French Prime Minister instructions on the usage of Free Software in the French administration

The Prime Minister

Paris, 19 September, 2012

To

Distinguished Ministers

Object: **Guidelines relative to the use of Free Software in the French Administration**

P.J.: **1**

Free Software is software whose intellectual property model is designed to give the user a broad freedom of use, modification, and distribution. It is widely used, both in private companies and in the administration. Its fields of use include application development, databases, server operating systems, office suites, and messaging.

Within the administration, the longstanding use of Free Software has allowed the development of skills and the accumulation of many positive experiences. The latter have demonstrated in particular the benefits of Free Software (lower cost, flexibility of use, leverage with software vendors).

After several years, during which the use of Free Software was the subject of many discussions, it is now possible to adopt a series of guidelines and recommendations for the sensible use of Free Software. This is the purpose of the attached document, which was drawn up in cooperation with the directors of your ministries' IT departments, in the context of a work conducted by the interministerial directorate of information and communication systems. I am asking you to implement within your departments the guidelines defined in the accompanying document.

Jean-Marc AYRAULT

Use of Free Software in public administration

September 2012

Table of Contents

1. Aim of the document

A longstanding use of Free Software has allowed for the development of skills and the accumulations of many positive experiences within the administration. An improved sharing of this knowledge and the defining of common guidelines would allow us to reach a new level, to increase both operational and financial efficiency.

As part of the interministerial work launched by the DISIC, a task group, led by the Ministry of Culture and Communication's CIO, was charged with defining the guidelines required for the use of Free Software within the ministries.

After a reminder about the context in which Free Software spread and the intellectual property model associated with it, this document presents the guidelines that stemmed from the initial work, specifying in particular the environments in which the use of Free Software is appropriate, and describing the joint actions that were initiated and the working groups hat were created.

Document status: *This document is based on the work of the interministerial group of experts. It was reviewed for approval, publication, and implementation during the 21 June 2012 CTSIC (Comité Technique des Systèmes d'Information et de Communication, the information- and communication-systems' technical committee).*

2. Origins and foundations of free software

2.1. Context in which Free Software developed its position

Free Software has reached a large share of technical infrastructures and has an increasingly important place of all information systems, all the way down to the end-user interface. Internet standards have created a common platform on which more and more software products are relying, thus making them interchangeable. Many software products are now “commodities” with a limited innovative value, and customers are less and less willing to pay a high price for products perceived as commonplace and having already given a return on investment and to accept to be bound to one provider.

From now on, in order to meet business needs, Free Software must be considered on equal footing with other solutions. The use of Free Software within the French administration follows this evolution.

2.2. The Free Software Model

Free Software is an intellectual property model taking different forms. Its principles are:

- to guarantee the **freedom to run the program**, for any purpose;
- to guarantee the **freedom to study how the program works and to adapt it** to one's needs;
- to guarantee the **freedom to redistribute copies of the program**;
- to allow **for the improvement of the program and for the distribution of these improvements to the public,** for the benefit of the entire community.

All of this implies of course that the source code has to be freely accessible. This very open vision has allowed for the creation of interest groups focused on specific Free Software programs, or “trunks”, of such programs, and has led to the emergence of a community-based development model. The most famous of these communities today are GNU/Linux, Apache, Mozilla (Firefox, Thunderbird) and the Document Foundation (LibreOffice). These principles give certain characteristics to free software:

- **like any intellectual property model, it tends to be self-sustaining**

  The most widely distributed free software program is distributed under the “copyleft” model of the GNU GPL. In principle, it prevents whoever uses the code from appropriating the community's efforts without repaying it with improvements or corrections. Contribution to the collective effort becomes a principle and sustains development momentum.

- **user needs guide the evolution of a Free Software program**

  A community has no interest in adding to a Free Software program a feature that is useful to only a very small number of its users. While users have difficulties controlling the regular changes in version that are imposed by the software editor, stability is a quality of Free Software programs. The rule is therefore the pooling of needs and the prioritization of upgrades.

- **the model guarantees that the community is able to keep control**

In certain Free Software communities, proprietary-software players are very active. Their company's own interest can bring them to direct developments away from the community's interests. The Free Software model then allows part of the community to create what is called a “fork”, that is to say, to start from the source code as it exists at a particular moment and steer development of that source code in another direction.

- **The model enables to create the emulation necessary for creativity**

  Whether through “forking” or by building on all the existing Free Software programs, those who are certain of having a good idea can always start with a small investment and gather a community around this idea. This is how a lot of free software is constantly being created, with only the ones relevant enough to be supported by a large number of developers and users surviving.

Contrary to popular belief, **use of Free Software does not at all mean that users have no obligations to comply with. A Free Software program is not free of rights** since it has an author. The initiators of Free Software, being realistic, inserted themselves into the legal world by formulating in licenses the applicable rights and obligations. Several major types of licenses have been defined, the main ones being the GNU General Public License (GPL), the Berkeley Software Distribution (BSD), and the Apache license. There is also one in French law, the CEA CNRS INRIA Free Software License (CECILL). The legal characteristics (be they hereditary or not, multilicensing, applicable law, guarantees) vary according to the license's authors, but these licenses are all objects of substantive and strong law, legally recognized.

When downloading a Free Software license, we find ourselves in an adhesion contract, that is to say we are in the same situation as we would be in had we bought a proprietary software program. The license terms are imposed by the author and are not negotiable. In the end, either the licensee accepts the license and can do what is mentioned in it, or they cannot enjoy any of the freedoms inherent in Free Software (modification and distribution).

This is one of the important, yet often neglected, aspects of Free Software: **one should know the obligations associated with a Free Software program, in particular when it comes to using it in a professional information system.**

2.3. Free Software, a service model

If the rights over Free Software are not associated with any financial compensation, this does not mean that it costs nothing to implement and use Free Software, in particular in the professional field.

Indeed, like with any software program, it is necessary to integrate it into its information system and to **ensure it is kept in operational conditions (support, maintenance), as well as to upgrade it as needed.** These tasks must be covered either internally or by service companies, several of which are specialized and advertise themselves as “Free Software Services Companies” (SSLL).

A “licensing cost/maintenance cost” license model is thus replaced by a “cost of service” model, which can be adapted to the using entity's real needs. **Where critical infrastructures are concerned, one should have support that is both strong and reactive,** generally external; in other contexts, community support will suffice.

In any case, Free Software service model's costs are rather insensitive to the volume of use (number

of installed servers, number of concurrent users...). Free Software lends itself therefore well to mutualization and favors a concentration of uses in an interministerial context. To this essential advantage, we can add the advantage of independence with respect to external actors. Indeed, a regular reopening of competition between services companies, which can all intervene on free software programs, keeps prices at market level.

It is important in this regard to highlight that the Council of State validated this principle of free competition, in a service model based on these Free Software programs, in its judgment n°350431, dated 30 September 2011. **The aministration can choose a Free Software solution unilaterally,** provided that it be usable by all players and that the latter, therefore, free from external hindrances, be able to provide a customized service offer.

3. Free software, a carefully thought-out choice

Free software was originally fueled by a philosophy of openness and by "militant pioneers," who made the more institutional users, be they from the public or from the private sector, suspicious of this approach.

Today, the choice of Free Software in the administration is not an ideological commitment, but the result of a carefully thought-out choice. There are many motivations, the main ones being: :

- the increasing pressure on the means of investment and the operation of IS, in conjunction with the strong increase in demand ;
- the promotion of the professional skills and expertise of IT teams, which are not mere solution buyers.

3.1. Advantages

Depending on the use, in the public context, Free Software can bring the following advantages:

- Free Software is not free of cost, but is often **less expensive.** Above all, its cost can be adjusted depending on how critical the systems are;
- Free Software is driven by needs, **minimizing superfluous upgrades;**
- Free Software makes it possible to manage versions depending on its context, and even to decide on a specific version **insuring its long-term support;**
- Free Software facilitates **experimentation and adaptation** to volume of use, the absence of royalties allowing for strong variations free of constraint;
- Free Software facilitates **pooling between public actors,** be it at the stage of expression of needs or to by capitalizing on existing strains;
- Free Software brings **increased transparency** to the definition and organization/ of information systems' security policy, with a demand and cost that adapt to the level of support chosen;
- Free Software makes **true competition between vendors** possible through the purchase of services from companies that have been put on equal footing by the publication of source code.

Within the context of public procurement, the use of Free Software offers the opportunity of favoring the principle of competition and of openness to public demand in the procurement of software and services. The judge clearly specified that an adjudicating power could, without calling into question the principles of public procurement, organize competition based on a Free Software solution chosen unilaterally by the administration (Judgment of the Council of State n°350431, dated 30 September 2011).

3.2. Limits/concerns

Free Software also has its limits and some important points must be noted :

- Free Software is **tied to a community:** it is therefore necessary to know and follow this community in order to ensure the permanence and the seriousness of the solution;
- Free licenses do not entail an absence of intellectual property law, but **another form of law,**

which has to be managed, especially in development;

- for the mere end user, the effect of brand and marketing applies also to software, and Free Software, having no price, is often thought to have no value;
- the possibility of contributing to the software's development through access to the source code should not **tempt one to multiply the additions of specific code,** at the risk of losing connection to the communal source code and having to maintain an isolated solution in the long term. An analytical approach to the value of “standard” deviation is rigorously called for;
- participation in dynamics of Free Software is linked to contribution: the user, especially the professional user, must not limit him- or herself to profiting from the system; they must **maintain the model by reinjecting a part of their profits** in one form or another;
- certain software vendors play at the margins of the Free Software model, managing a version called “enterprise” or “premium” under a proprietary classic license and a version called “communal” under a Free license, which is, however, often out of date compared to the other version. This is the model called “Freemium”. **These free software programs, driven by an editor more than by a community, must be used with caution** since they are at constant risk of reverting to a proprietary mode.

3.3. The different contexts of use

When one decides to develop an information system, the choice of using Free Software, or even of developing according to the Free Software model, must be analyzed according to criteria that take into account the parameters of use, the number of actors concerned, the complexity of the system, and the necessary involvement.

3.3.1. The contexts favorable to the Free Software model

3.3.1.1. An existing and internationally recognized Free Software program

Certain Free Software programs are supported by an already very strong community with many users (JBoss, Firefox...). In certain cases, Free Software becomes unavoidable, like for instance for the Apache Web server, which is used by almost 60 percent of the installed userbase (late 2011).

In this case, **cost reduction is direct, and the product is immediately usable** and is often sufficiently supported throughout the community; it remains possible, however, to connect oneself to the development community to report a bug, if need be, and to contribute to the improvement of the program.

Examples of this context are everywhere and lead to the increasingly important deployment of these large Free Software projects in both the public and private sectors : Linux, Apache, Firefox, Thunderbird, JBoss, OpenSSL, Eclipse...

In the case of software programs for end users, one should ensure a **change management** to prepare for the introduction of a new program, especially if the latter is to replace a widely used solution. This must be factored into the economic calculation.

3.3.1.2. Deployment of a software program over a large infrastructure

In certain large systems or for certain applications intended for users, it's necessary to buy a large number of licenses. Thousands of database or operating-system licenses can entail substantial costs.

It can consequently be cost effective to support directly, or even to improve, existing Free Software and to participate in its maintenance. This investment can then be useful to every other public entity.

An example of the use of Free Software programs in an information system critical to a ministerial department allowed for **a tenfold reduction of the applications' costs of operation**. This net cost reduction was obtained by putting in place, under strict conditions (48-hour resolution delays...), a maintenance market over more than 100 software programs.

In the case of user workstations, the deployment of new software or updates can be done in a homogeneous way for all computers in use, at no particular cost (no license buy-backs, no dynamic-license purchase), which facilitates maintaining the installed base's homogeneity and leads to a reduction in user-support costs and to an increase in quality.

3.3.1.3. A software program used in a virtualized context or at high load variation

In a comparable logic, **deployment in a virtualized context simplifies the creation of logical servers and the adaptation of either their number or the allocated CPU Usage Limit.** The management of and payment for proprietary licenses can be either a hindrance to this adaptability or a nonoptimized generator of both complexity and cost. Indeed, proprietary licenses have costs linked to the maximum physical power utilized.

In contrast, Free Software support cost, both internal and external, is not affected by intensity of use and depends on quality-of-service requirements. It is therefore linked to criticality of use. In most cases, as the recently granted interministerial-support contract demonstrates, it will be much lower than the cost of licenses covering deployment at peak load.

3.3.1.4. A software program used in the context of agile development

Agile development is by definition “opportunistic” and adds functions as user needs are defined. This development method, to a limited extent, allows for proceeding by “trial and error”. It is therefore difficult to have, from the outset, a clear picture of the software that is useful and that will need to be integrated.

The use of Free Software allows for “digging” as the development goes in available free software programs, in a technological framework particular to each entity, according to its adequacy, without question of right of use either in the development phase or in the operational phase.

3.3.1.5. In situations of weak competition

The market leader having eliminated competition, there are fewer and fewer credible commercial alternatives to certain vendor's software products. Free Software then offers alternative possibilities.

Certain strains have a high functioning level and can **replace proprietary software programs at limited cost** with support ensured on a flat-rate basis and as shared as possible. Linux systems have thus clearly demonstrated their value.

Others have a functional contour that is not as rich as that of the proprietary software editor's

program and have a vocation to **be selected when the complex functions specific to proprietary solutions are not absolutely necessary.** This, for instance, is the case of databases where PostgreSQL constitutes an alternative that is often relevant and to be developed.

It is to be noted that editors of software solutions (large ERPs, ...) generally favor the use of components of proprietary architecture (OS, SGBD) by guaranteeing compatibility with only the latter. Even if certain Free Software solutions, Linux in particular, are part of supported compatibility matrices, **special attention must be paid to this point when selecting a software package,** which could, through this channel, limit technological choices and give rise to hidden additional costs.

Financial ministries have demonstrated the value of the use of Free Software in this context, within the framework of resuming an application in Cobol. The use of OpenCobol allowed them to reduce costs by a factor greater than 10.

3.3.1.6. A same need to be dealt with by many of public entities

Needs related to business or the regulation are shared by a number of public entities. In this context, it is particularly counterproductive for each entity to conduct their specific developments on their end and pay the entirety, instead of sharing the development expenses. Be it for the development of a help-management system at the local level or of an e-procurement platform, it is easy to see that an association of actors facilitated by the Free Software model will profit all.

Certain local governments have understood this and have set in place organizations, such as the ADULLACT, to **federate their developments** following the Free Software model.

Incidentally, we notice a remarkable increase in the technical quality of developments made for publication under Free Software licenses compared to that of specific developments carried out earlier. This is also a positive outcome of publishing the developments to the outside.

3.3.1.7. A deployment in multiple contexts of public and private entities

Some state functions call for application or system interfaces that can be used by many players, be these but the different types of local governments. For example, public accounting leads to the exchange of accounting records formatted with the local authorizing officers. It will have to be possible for these functions to be integrated into the systems used by these partners and therefore for the software editors to use them sometimes.

Having usage licenses that are open to such large populations, and that allow broad integration, comes close to having a paid-up license, which in the proprietary model is in general very expensive, since it is contrary to its logic. Moreover, in this context, it is not easy for the state to pay all or part for all the public entities. In the Free Software model, this is only the normal license, which in any event does not require any distribution control. The **ease of management, cost reduction, and convenience of reuse** are obvious.

Examples of eligible contexts are given by systems developed by the State together with numerous partners, such as the DGFiP's Xemelios, for instance, allowing paperless management of accounting and financial files.

3.3.2. Contexts unfavorable to the Free Software model

3.3.2.1. Small number of players involved in the implementation

In order to be useful, development following the Free Software model requires the establishment of a community of contributors who share and pool their efforts. If the entities who are concerned or who need to master the software development are few and poorly identified, it is less useful to free its development; one should, however, be careful, and it may be prudent to retain the possibility of doing so, should the need arise later.

3.3.2.2. Comprehensive and complex system (non-modular)

The resource pooling principle, underpinning the Free Software model, goes hand in hand with a modular approach in the development and design of information systems. Bricks and modules, more easily affordable for new entrants, more easily reusable in many systems and easier to maintain, are more eligible for the Free Software model.

In contrast, complete and complex systems are sometimes so difficult to manage that they require a dedicated professional, that is to say a software vendor ... Such is the case for management systems, such as non-specialized ERPs.

It should, however, be emphasized that the principles of software architecture and of good management of the evolution of information systems encourage limiting the use of monolithic systems, and instead favoring the modularity that corresponds to the Free Software model.

4. The interministerial action on free software

To facilitate the use of Free Software solutions in the administration's choices, and thus put it on par with the other offers, while at the same time achieving maximum economic efficiency and quality, the State must act in a concerted and coordinated way, following the procedures detailed below.

4.1. Instituting an effective convergence on Free Software stubs

The founding principle of free software is the pooling of efforts. The concentration of players on certain stubs is a guarantee of efficiency. Thus expertise-development efforts, bug-correction costs, sometimes at the user's expense, or even upgrade costs, are shared.

A **systems-convergence frame** that is to be preferred in the development of State information systems, and that was **defined in 2012**, is now maintained through interministerial coordination. It especially affects the most widely used systems, be they on servers or on desktop stations.

This framework does not hinder innovation by trial of new stubs, which could help its upgrade. Nor does it make mandatory the adaptive upgrading of existing noncompliant applications; it does, however, define reference versions that should be preferred and specifies solutions to be abandoned, with possible reservations in particular contexts of use.

This framework is also an essential component for the progressive convergence of operating environments and for the pooling of certain means. As such, it must be integrated in all the technological frameworks of the ministries and be taken into account at the time of new developments and major rewrites.

Each ministry must participate in this framework's updating and in its progressive reinforcing. In particular, it will regularly declare the use made of the framework's free software programs and the uses outside of the framework, in order to enable the monitoring of its use, and the management of its evolution.

The frame of convergence is published on the collaborative software of the “core” team (see organization) and updated again on a quarterly basis connected to interministerial support markets.

4.2. Activate an expertise network on the convergence stub

The efficiency of the mutualization of efforts around Free Software also comes from the sharing of expertise and from the increase in skill on the stubs. Each ministry can hardly be skilled on all free software programs, but each has skills. **The creation of a network of experts allows for all the administration to profit from the occasionally required specific expertise.**

The bearers of this type of expertise naturally volunteer to share and are valued through the use of their skill. The most difficult is establishing connections and ensuring the hierarchy's acceptance of the load, be it limited or not, involved in participating in the interministerial effort.

It is appropriate to create networks maintained by occasional in-person meetings, essential for the creation of a rich and sustained exchange, through on-line collaborative work. Several tools have been set up toward this end:

- the **thematic work groups,** with regular meetings about office suites (MimO), server base

(MimOS), installed base management (MimOG) and database (MimDB) topics;
- **“Free Software Day”,** facilitating the opening to new actors, and enabling the opening of new topics or the broad diffusion of feedback;
- the **mailing lists**, by thematic groups or around specific themes, allow us to avail ourselves instantly of the network on specific questions;
- the **collaborative sites** of thematic groups for the sharing of resources (distribution CD or LibreOffice tool kit...).

Each ministry is involved in the common approach.

The expertise networks, and, first and foremost, the thematic groups, when they exist, are the crucible for the definition and evolution of the convergence base on their perimeter of competence. The list proposed for the convergence context is published on the group's collaborative tool.

4.3. Improving Free Software support in a controlled economic context

Free Software allows for the adaptation of the maintenance policy in accordance with systems' extent and criticality. Much of Free Software use was made without any particular support, taking advantage of the support provided by the communities. Even if this method remains valid, it is necessary, for a number of uses, to have a reactive support with commitments to results.

Free Software allows for greater support commitments than proprietary software does, because the code is accessible for internal adjustments or for adjustments by a chosen provider, while software vendors, on the other hand, have standardized processes that are but partially adapted to customer needs. The problem of large proprietary software programs is reinforced by the distance between the centers of development and the complexity of the processes of global-scale version publication. In addition, publishers' contractual policy generally deprives the customer of the possibility of negotiating the standard level of service of the editor, a level of service that furthermore highly protects the latter.

Financial ministries have demonstrated the feasibility and economic efficiency of the implementation of a support market by a provider, such as an IT services company. **At the interministerial level, a deal was signed, under the auspices of the SAE and under the direction of the Ministry of the Interior. to meet the needs of other ministries.** It provides for cost-reduction mechanisms when multiple actors require the support for the same strain (more precisely, for the same versions of a strain). This market is therefore an additional incentive for the implementation of the convergence framework.

4.4. Contributing in a coordinated way to chosen free software programs

Through the interministerial support market and the convergence framework, the State is now concentrating its action on a set of strains and will contribute to their improvement by transferring patches back to the communities. However, to respect the logic of Free Software dynamics, it is necessary that the administration also contributes directly to the functional enrichment of some strains, particularly to those through which it saves the most. By reinjecting a small part of the spared expense, ministries could significantly improve the offer, to everyone's advantage.

A simple rule to be applied would be to reinject systematically from 5 to 10 percent of the avoided licensing costs. This allows one to contribute in a useful way in all cases, to not put at risk the economic gain of using Free Software, without systematically performing a thorough study of comprehensive gain.

This contribution can take numerous forms:

- in markets using Free Software, ensuring the resumption by the community of eventual core upgrades is made possible, thus facilitating their monitoring by the community and avoiding specific maintenance;
- envisioning the financing of research conventions, for the addition of evolved features that could be the subject of academic work (for instance, a multilingual grammatical corrector for an office suite);
- studying the funding through funds on accessibility for the improvement of desktop software
- setting up a market of expertise and strain evolution that contributes to the communities through a service provider;
- and, of course, favoring professional involvement of agents, who are often personally passionate, in some communities. This involvement can take the form of contributing to code, but also to less technical domains like translation, documentation...

In the wake of the interministerial support market, the MI and SAE are setting up a market of free software expertise and evolution, which can be the basis for concerted and shared interministerial contributions. This action will carry all the more weight as a large number of ministries will take part. The existence of a second market will, moreover, allow for limiting the negative effect of purchase concentration that does not favor the rise of multiple large Free Software actors.

4.5. Monitoring the large communities

Just as software editors maintain regular contact with all ministries, to update knowledge of their products, be able anticipate their changes, and even gauge needs, it is essential to have links to large communities such as the Mozilla Foundation or the Document Foundation. However, as these foundations do not have a commercial approach, the logic is reversed. It is the administration that must regularly contact them.

Regarding these communities, it is important to speak with one voice in order to be heard. This unified voice carries more weight with the mass of users throughout the world.

These regular contacts ensure consideration of needs not yet met, be it functionally or in the free software management processes. In particular, it is essential that all free software programs integrate the long-term-maintenance version's logic that corresponds to the management approach of our infrastructures. These contacts also allow for others to have precise information about the upgrades to be expected, the communities' needs, which could possibly be met by interministerial actions.

Some ministries have privileged contacts with some communities; they are then in charge of communications, in keeping with the "core" team, and can organize meetings, as needed, with the interministerial groups.

4.6. Deploying credible and operational alternatives to the large software editors' solutions

Within the framework of the development of the State information systems, we must ensure the control of the operating costs and sustained performance over time. To this end, the State must promote competition even in areas dominated by internationally recognized actors. One of the solutions is to take advantage of credible alternatives provided by Free Software. In this spirit, work on LibreOffice or PostgreSQL is essential. It is driven by thematic groups, MimO and MimDB, respectively. It specifically aims to reinforce sharing on all aspects of implementation of these programs (technological, support, feedback, training, ...). Lead experts are also designated. The “core” group (see organization), in close coordination with the SAE, defines the actions focused on certain free software, to encourage specifically their adoption in the context of transition from commercial offering to free offering. The next operation could apply to virtualization layers.

4.7. Mapping out use and its impacts

To strengthen the Free Software approach, we must also follow its evolution and the effective deployment as much in data centers as at the bureaucratic level. An annual analysis of the volumes and of the value of this use, as well as of its evolution, will from now on be carried out and published.

4.8. Developing a culture of use of Free Software licenses in developments of public IS

The State must ensure that its developments be usable by all the actors involved in its information systems. The many statutes of public entities and the possible involvement of end users like professionals or professional-solution editors complicate the management of code ownership.

Regarding specific developments, the State must safeguard its ability to release code in a manner that maximizes its own benefit, regardless of which provider did the development. The State must therefore make use, or prepare the use, of Free Software licenses, be they permissive or not, depending on the context. It must also ensure this freedom prevail vis-à-vis its suppliers in every context that could lead to reuse, unless explicit additional costs are generated. A network of expertise is established between counsels/purchasers involved in the drafting of administrative clauses. In general, specific training courses are set up, fast-track ones for project managers and developers, more in-depth ones for lawyers and buyers to create a real mastery of the subject within ministries and CIOs.

Moreover, the CCAG TIC will be reviewed to define an option allowing for the administration to release software as Free Software, an option that to this day does not exist. Provider liability clauses and obligations must also be added when said providers use or develop Free Software code.

Furthermore, license management must be one of the components of the explicit IT governance within each ministry.

5. Support points for interministerial action on Free Software

5.1. The interministerial “Free Software” bodies

To enable interministerial work while relying on the larger process of the public sphere, two permanent levels have been created;

- a strictly interministerial team called the “core” team, which concentrates the proposals of decisions, and the proposals of validation of the choices to be submitted to CTSIC/CSIS, and steers the actions stemming from decisions of the interministerial governance (markets, evolutions of free software catalogs, directives implementation...);
- focus groups committed to mutualization and open to public structures, and that bring together the experts from a field, favor exchange and skills development, and offer guidelines. Four groups have been identified:
    - mimO: interministerial mutualization for an open productivity suite;
    - mimOG: interministerial mutualization for installed base management (OCS and GLPI);
    - mimBD: interministerial mutualization for databases;
    - mimOS: interministerial mutualization for the operating system and the underlying plumbing layers.

The missions, the organization, and the resources of these teams are expanded on in the appendix.

5.2. Complementary levers

In order to support this process, complementary actions must be initiated, either interministerially or at each ministry's initiative:

- convergence-framework integration on common free software programs in all the ministries' technological frameworks;
- systematic review of Free Software alternatives during new developments and major application rewrites, and, in this regard, an evaluation will be done on the choices made in each project in the context of articles 7 and 8 (database choices will be a particular point of focus);
- strongly recommended participation of all ministries in the "core" group, in order to add to the momentum;
- explicit definition by each ministry of the instructions for expert involvement in the efforts of sharing in expertise networks;
- under the direction of the SAE, a feasability study of a CCAG TIC review for setting up a development option with the possibility of liberating code, along with a definition of service providers' obligations in the use of Free Software;
- systematic association to any recommended format (especially in the General Interoperability Framework) of a Free Software reference implementation. The formats will then de facto be sufficiently open;

- dissemination of good practices, particularly in the office-automation context, so that the use of Free Software not be weighed down.

6. APPENDIX: “Free Software” interministerial bodies organization

6.1. The “core” team

The “core” team is in charge of steering, of defining directions and choices at the interministerial level. In this capacity, it has as members only ministry representatives, with at least one designated representative, as well as one member of the ANSSI and of the SAE. Each organization's representative is in charge of representing their ministry's position, or, failing that, of being the link with their ministry's decision makers, and, in turn, of making sure the chosen principles of action are put into practice.

6.1.1. Team missions

- define and improve the convergence framework of the Free Software programs;
- follow the focus groups committed to mutualization, to ensure work is taken into account and distributed, and to validate the proposed directions;
- implement pilot programs on the themes set by the DISIC;
- lead interministerial procurements on Free Software (support, expertise, evolution, ...) in coordination with the supporting ministries and the SAE;
- lead the off-market contributing operations;
- follow the relations and build contacts with the large communities;
- choose and follow the deployment of Free Software alternatives;
- ensure business intelligence in the use of Free Software and ensure associated indicators in coordination with the SAE are monitored;
- define interministerial Free Software communication and training operations;
- improve Free Software usage and contracting;
- exchange information about the activities within the ministries and on the emerging needs, to promote sharing. To this end, take an inventory of the free software programs created in certain ministries that can be reused by others (ex. Xemelios, OCS...);
- ensure the coherence of the “Use of Free Software” project management with the other DISIC projects;
- report activities to the DISIC.

6.1.2. Team resources

For its missions as a whole, its resources are limited to what is strictly required in a shared approach:

- physical quarterly meetings in a ministry room;
- mailing lists, administrated by the MCC, for each constituted group, including the “core” team;
- collaborative Web site, administrated by MEDDE, within the DISIC Web site.

Moreover, to enrich the debate, allow new entrants to discover the bodies, organize a sharing of

experiences, and test new areas of study, “Free Software days” are organized twice a year. They are an opportunity for flash presentations, feedback, and debate on the use of Free Software in the administration. They are reserved for public servants, and are not divulged to third parties, in order to preserve freedom of speech regarding the encountered difficulties.

Members of the “core” team can de facto devote only limited and fragmented time to this activity. To maintain lasting motivation, production quality, and work continuity, it is necessary, within the DISIC or within a ministry, to count on one individual dedicated, during the great majority of their work time, to the organization and formalization of the work of the “core” team. They could also ensure the formal link with the mutualization focus groups.

It would be useful, eventually for the ministries to define a budget for contributing to certain free software programs and apprise the core team of it.

6.2. Mutualization focus groups

Mutualization focus groups gather experts from a field in order to promote experience sharing, skill broadening, the set up of exchange and help networks, and guideline and technical-choice proposals. As such, they manage the activity by subsidiarity around free software in their field.

Four groups have been identified:

- mimO: interministerial mutualization for an open productivity suite;
- mimOG: interministerial mutualization for installed base management (OCS and GLPI);
- mimBD: interministerial mutualization for databases;
- mimOS: interministerial mutualization for the operating system and the underlying plumbing layers.

They welcome representatives of the State administration, public organizations, local governements.... All the participants are public agents.

6.2.1. The general missions of the groups

- elaborate recommendations for the convergence base;
- identify the possible sharing sources;
- collect feedback;
- collect and broadcast the information (chat rooms and file sharing);
- interface between the communities and the administration, and organize meetings;
- ensure a technological watch;
- follow other DISIC projects' work (workstation/work environment, TCI for the operational aspects…);
- facilitate collaborative spaces;
- reporting regularly on the activity to the “core” team.

The DISIC representative ensures the link between projects. Members participating in other projects make reviews on the thinking progress about these projects.

6.2.2. Groups specificities

6.2.2.1. MimO

Domain:

- all desktop applications.

Specific missions:

- manage the distribution of the office suite and useful plugins and associated tools (correctors…);
- produce installation packages, documentations, all-in-one toolkits…

Eventually it will spread also to the Free Software office base. Operational tasks will be handled by some ministries.

A list of subjects to study over the course of 2012 is established:

- Mozilla Firefox (also with Android) and Thunderbird upgrades;
- office uses on mobile agents (reading messages and office files on cell phones, pads...);
- LO/OOo competition;
- use of Trustedbird;
- choice of Firefox instead of Chrome;
- Grammalecte or other grammatical corrector, connection to its designer and study of involvement in its development;
- Lightning upgrade;
- terminological-corrector maintenance;
- establishment of an exchange platform for converting files into open formats (cf. initiative Europe).

6.2.2.2. MimOG

Domain:

- all the software programs useful for managing and supporting the installed base.

Specific missions:

- produce installation packages, documentation, OCS and GLPI tool kits...

A list of subjects to study over the course of 2012 is established:

- fine-tuned management of OCS and GLPI versions (and distributions) and of their plugins, to define a convergence framework;
- FusionInvetory vs. OCS (choice to be made and struggle to be moderated...);
- VNC strain to be adopted;
- tools for software packaging and distribution.

6.2.2.3. MimBD

Domain:

- all the database-related software programs.

Specific missions:

- promote migration from proprietary databases to Free Software databases, in particular to PostgreSQL.

A list of subjects to study over the course of 2012 is established:

- collection of feedback relating to migration;
- cross analysis of policies related to free and proprietary databases;
- usage recommendations;
- migrations recommendations;
- future of MySQL: MariaDB, SkySQL...;
- NoSQL technologies.

6.2.2.4. MimOS

Domain:

- a set of low-level server software programs, in particular, operating systems and virtualization tools, as well as all the tools useful for server management and operation.

A list of subjects to study over the course of 2012 is established:

- a full review of the state in the systems and visualization field;
- Linux distribution to be preferred.

6.2.3. Focus group resources

For all their missions, groups' resources are limited to the strict minimum required by a shared approach

- quarterly physical meetings in a ministry room;
- thematic mailing lists, operated maintained by MCC;
- collaborative Web site, operated by MEDDE, within the Web site of the DISIC or of the MEDDE;
- package-distribution site, operated by one of the members of the group.