

# **Logiciel libre & Open Source**

## **Réflexions & témoignages**

# Quelle image représente le mieux la démarche Open Source?



# Mise en situation dans une logique historique sous trois angles

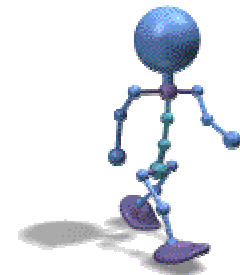
Technologique

Economique

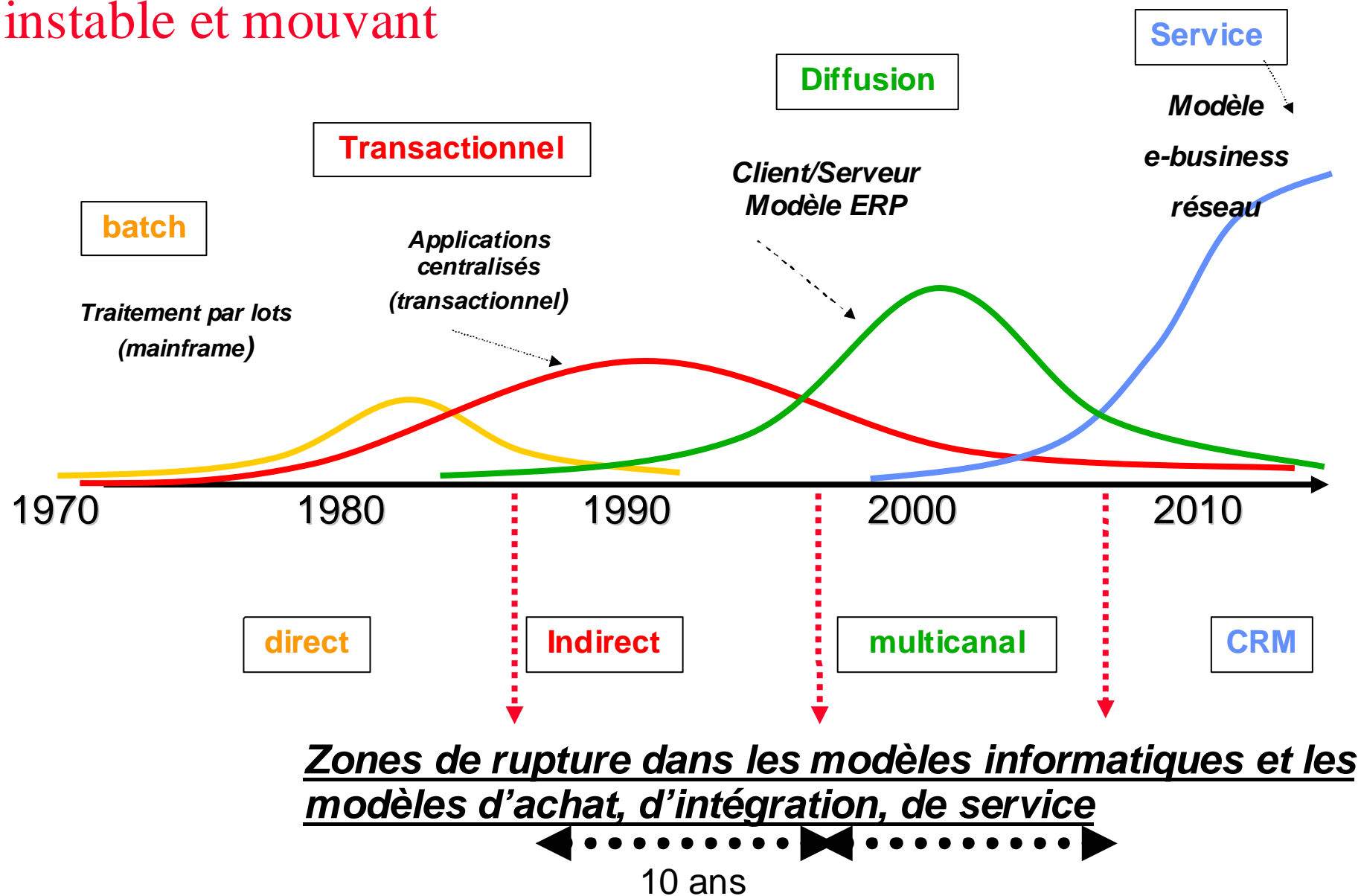
Sociologique

# Mise en situation dans une logique historique

- ❑ Technologique :
  - ✓ la loi de Moore



# La logique Open Source, une nécessité dans un environnement instable et mouvant



# La chaîne de valeur informatique, un changement de modèle ..... tous les 10 ans (suite)



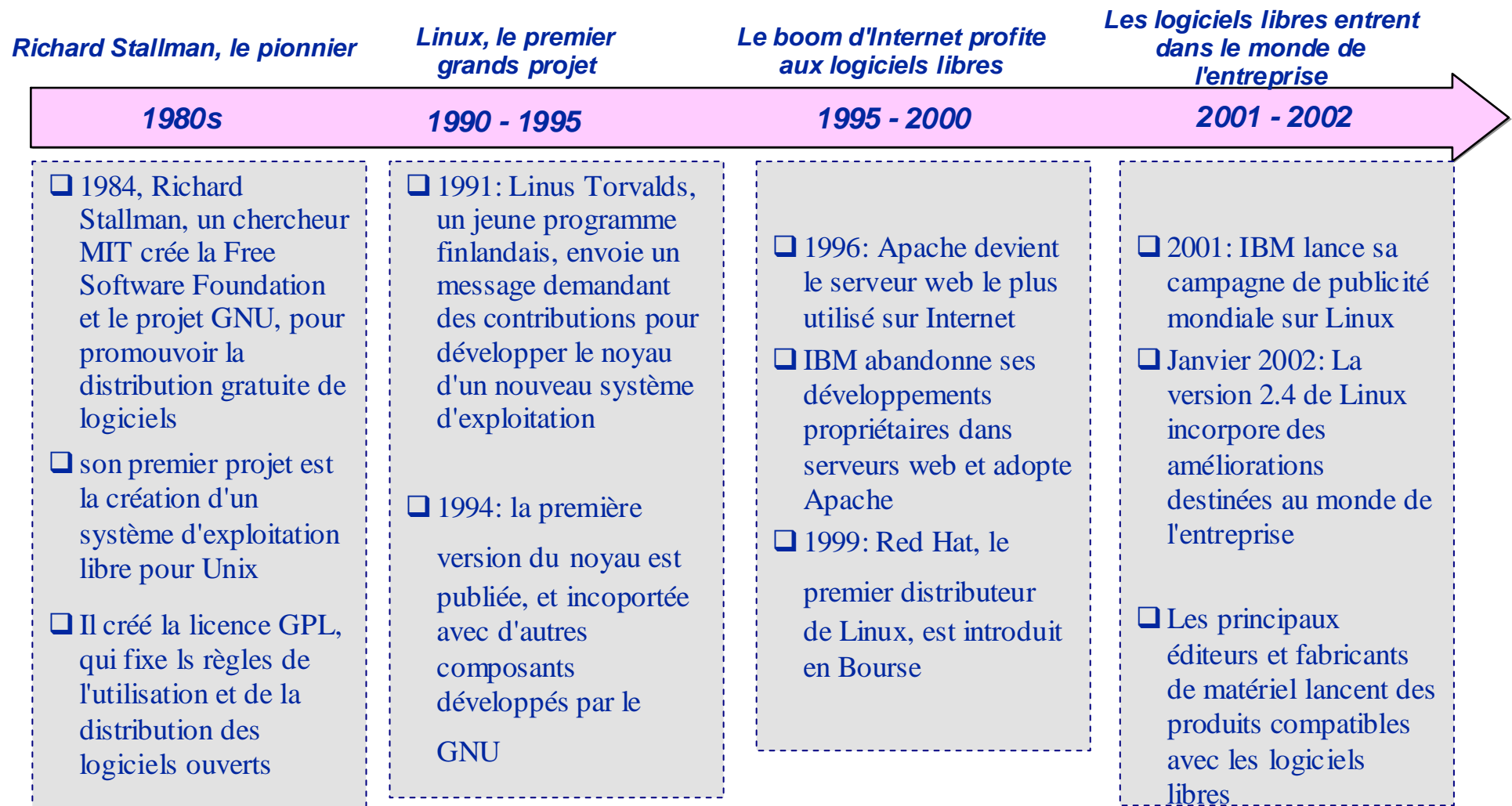
<b>Besoins Clients</b>	- Gérer des données	- Automatiser	- Réduire les coûts	- Conquérir des parts de marché
<b>Typologie de l'Offre</b>	<b>- Hardware</b>	<b>- Software</b>	<b>- Netware</b>	<b>- Freeware</b>
<b>Avantage compétitif</b>	- Performance produit (Technologie)	- Pertinence de l'information (Solution métier)	- Transparence de l'information (Solution marché du client)	- Exploitation de l'information (Rapidité d'évolution client)
<b>Compétences humaines des fournisseurs (en régions)</b>	- Commerciaux - Techniciens	- Commerciaux - Techniciens - Supports	- Commerciaux - Consultants - Dir. de projets - Techniciens (logiciel & intégration)	- Commerciaux - Consultants - D. de projets - Techniciens <b>- Veille et KM</b> <i>et changement d'approche</i>

# L'informatique, les impacts de la loi de Moore

- 1965 - 1975 **IBM, 30.000 années-hommes de développement pour l'environnement mainframe**
  - ▣ Mainframe, 50.000 unités, HW 1M€, SW 100K€, **10 applications**
- 1975 - 1985
  - ▣ Mini, 500.000 unités, HW 100K€, SW 10K€, **100 applications**
- 1985 - 1995
  - ▣ Micro, 100.000.000 unités, HW 10K€, SW 1K€, **1000 applications**
- 1995- 2005
  - ▣ Internet, 1 Bio unités, HW 1 K€, SW 100€, **10000 applications**
- 2005-2015
  - ▣ M to M 10 Bio unités, HW 1000 €, SW 10 €, **100.000 applications**

***Des performances x 2 tous les 18 mois, des prix divisés par 10 tous les 10 ans, un nombre d'applications dans l'entreprise multiplié par 10 tous les 10 ans!!***

# Libre et Open Source, un mouvement qui ne date pas d'hier



**Nés sur Internet et dans le monde universitaire, les logiciels libres sont-ils adaptés à une utilisation plus large en entreprise ?**

Source : présentation ATICA et CGEY/AFNET 2002

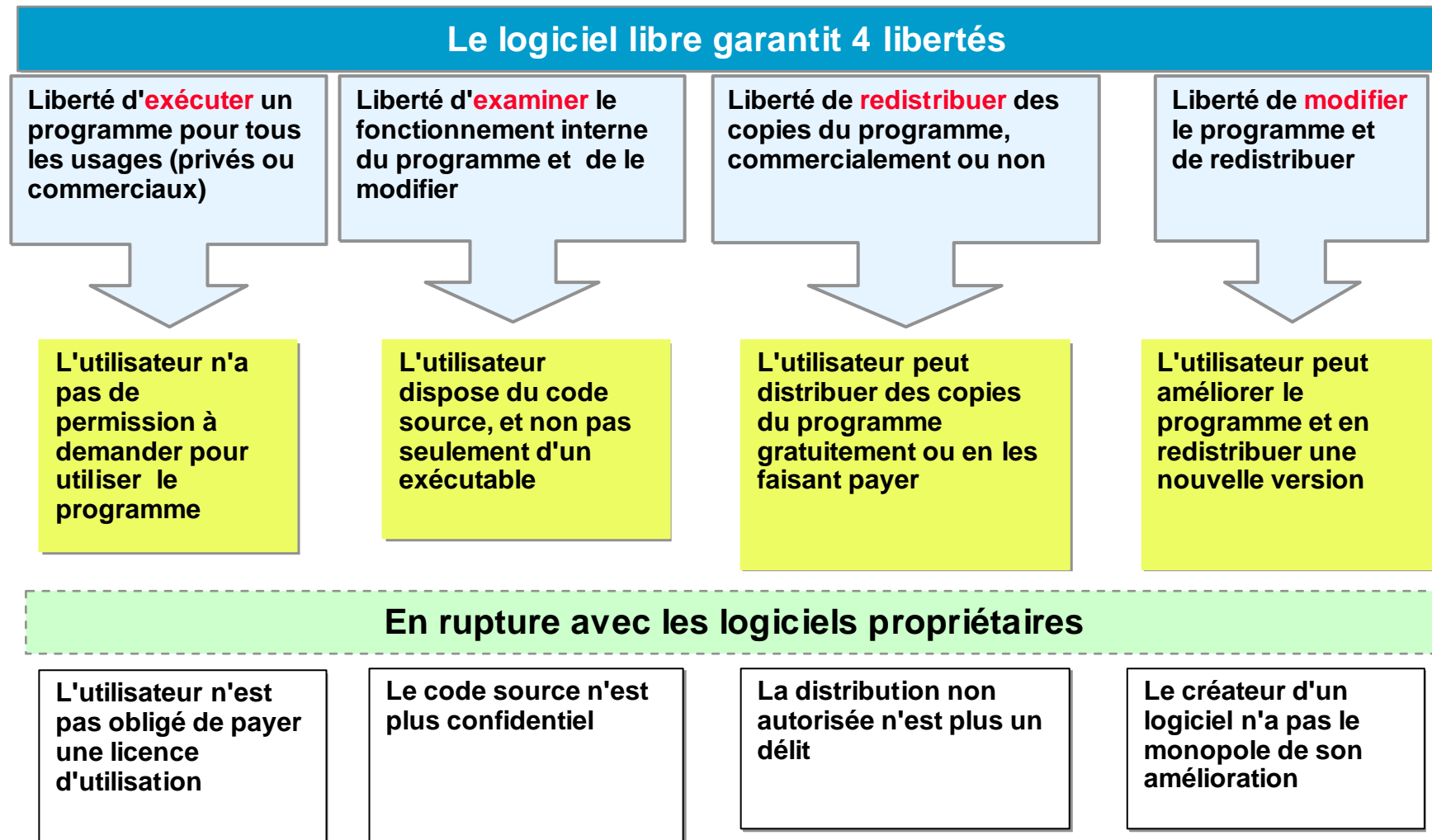
## Definition "Open Source"

- Le code source du logiciel est fourni avec le code objet
- Le client a le droit de modifier le logiciel
- Toute proposition de modification de code source doit être publiée
- Pas de discrimination envers les personnes ou groupes de personnes
- Distribution sous forme de logiciel exécutable (licence)
- Les licences ne doivent pas être spécifiques à un produit
- Les licences ne doivent pas perturber les autres logiciels
- Re distribution gratuite

Source : présentation ATICA et CGEY/AFNET2002

---

# Les logiciels libres se définissent par quatre principes en opposition avec ceux du logiciel propriétaire



Source : présentation ATICA et CGEY/AFNET2002

# Mise en situation dans une logique historique

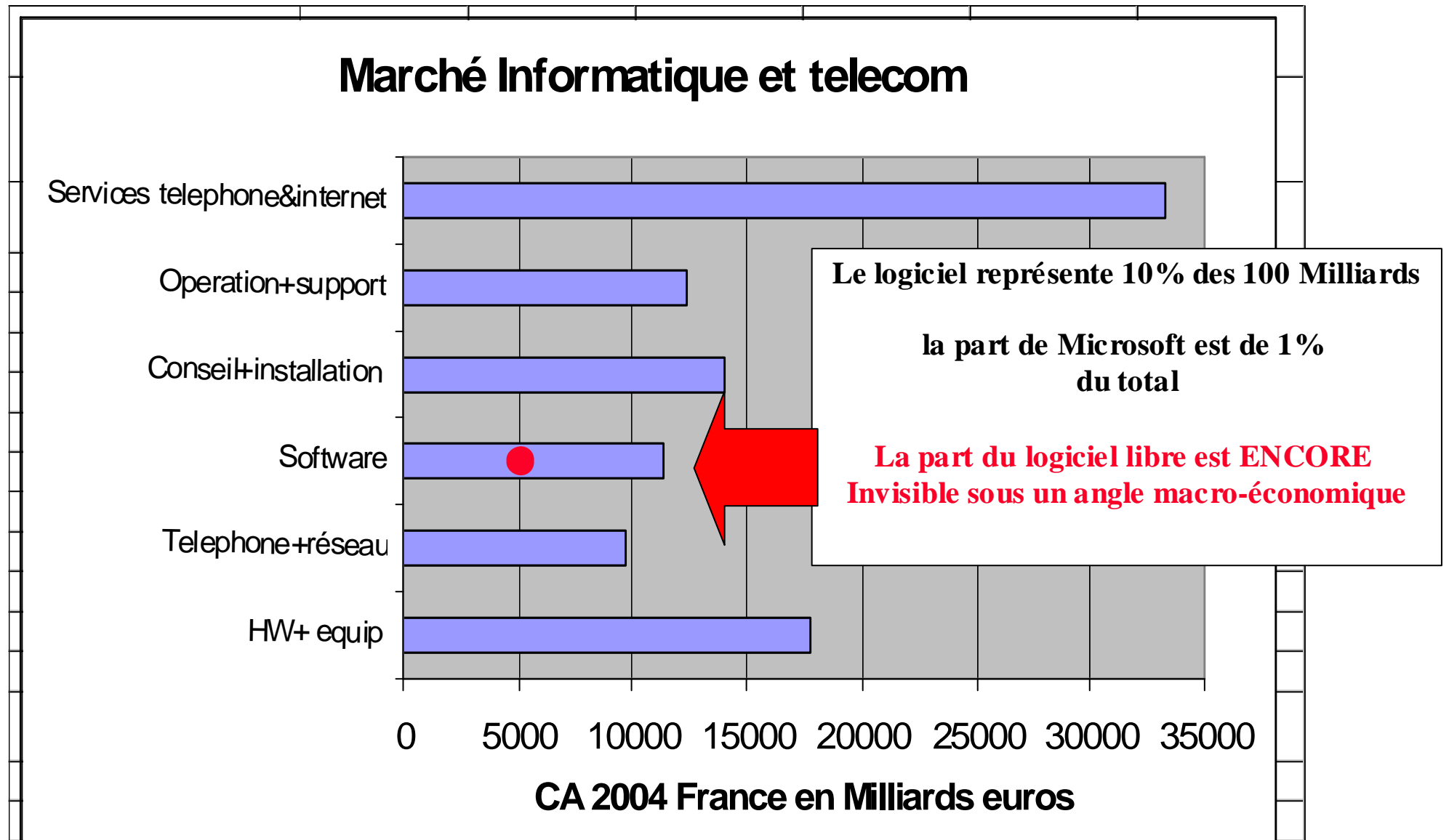
- ❑ Technologique :

  - ✓ la loi de Moore

- ❑ Economique

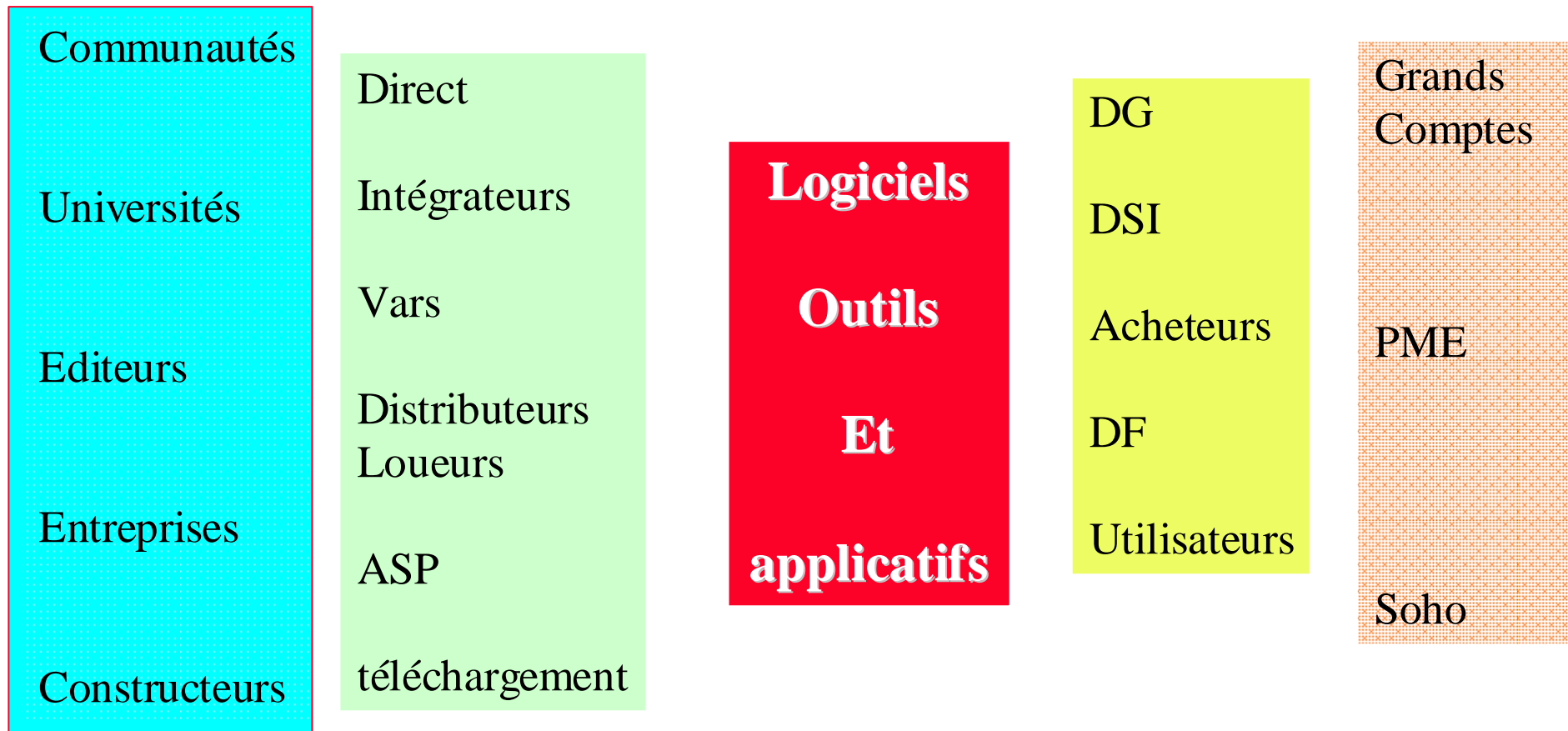
  - ✓ un marché piloté par les offreurs et la « demande Business »

# Le marché informatique et telecom en France: 100 Milliards €



# Les acteurs de la relation informatique dans la commercialisation de logiciels informatiques

Développeurs    Vendeurs    < ----- >    Acheteurs    Clients



## Caractéristiques des achats I&T

### Un marché mouvant de spécialistes

Offre foisonnante  
Évolutions rapides et profondes  
Produits de plus en plus complexes  
Discours marketing trompeurs

### Un marché tenu par l'offre

Nombreuses situations de monopole  
Politiques tarifaires et conditions commerciales peu lisibles  
Coûts de réversibilité souvent très élevés

### Des fournisseurs très variés

Des grands fournisseurs mondiaux  
Une multitude de petits acteurs locaux  
Des distributeurs et représentants locaux

### Des processus achat complexes

Un grand nombre d'intervenants dans le processus  
De nombreux comportements "déviant"

Avoir une **vision claire** du marché, des produits et des fournisseurs pour mieux acheter

**Rééquilibrer** les relations et de tirer les prix vers le bas

Gagner du temps dans la recherche de fournisseurs potentiels et mieux **comparer**

**Rationaliser** les processus pour gagner du temps et réduire les coûts

Comment prendre en compte les apports  
du libre et de l'Open Source?

Côté fournisseur ?

Côté client ?

# Un exemple d'étude d'impact du logiciel libre (en 2002) :

le cas de vente de solution de gestion métier (48 postes utilisateurs)

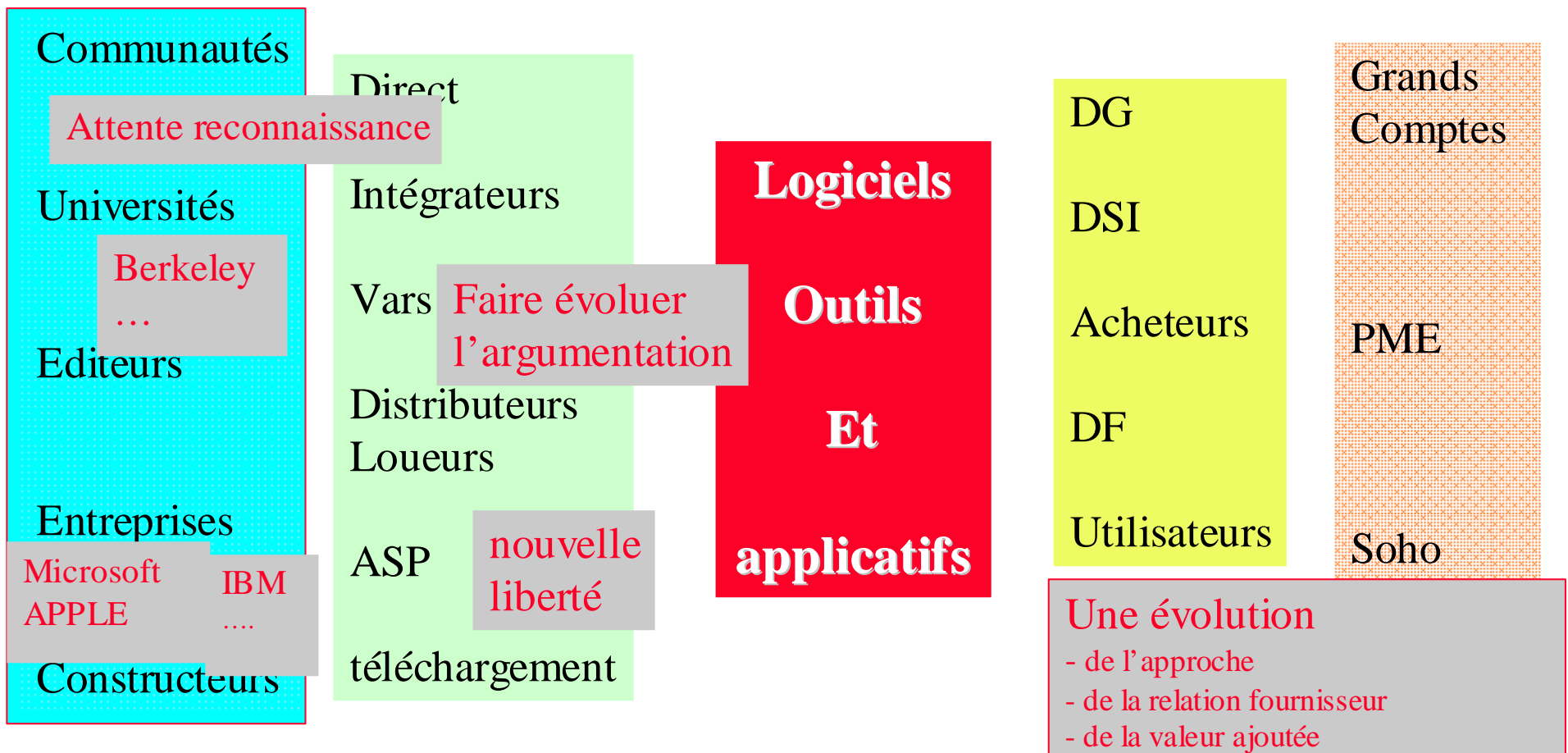
## Eclatement des coûts sur 3ans :

59 % pour le matériel, **24 % de licences logiciels + maintenance**,  
17% de prestation (maîtrise d'ouvrage, intégration et formation)

	Montant en K€			en %		
	Investissement	Maintenance 3ans	Total	Invest	maint	Total
Serveur+ perif	69,2	15,2	84,5	29,7%	32,8%	30,2%
Réseau	21,8	1,5	23,3	9,4%	3,3%	8,4%
Postes de travail	56,4	0,0	56,4	24,2%	0,0%	20,2%
Licences SGBD	15,2	9,9	25,2	6,5%	21,3%	9,0%
Licences applicatif	22,9	19,8	42,7	9,8%	42,6%	15,3%
Maitrise d'œuvre	6,9	0,0	6,9	2,9%	0,0%	2,5%
Prestations technic	6,9	0,0	6,9	2,9%	0,0%	2,5%
Formation	33,5	0,0	33,5	14,4%	0,0%	12,0%
<b>Total</b>	<b>232,8</b>	<b>46,5</b>	<b>279,3</b>	100,0%	100,0%	100,0%
HW	147	17	164	63,3%	36,1%	58,8%
SW	38	30	68	16,4%	63,9%	24,3%
Prestations	47	0	47	20,3%	0,0%	16,9%

# Les acteurs de la relation informatique dans la vente de logiciels informatiques

Développeurs    Vendeurs    < ----- >    Acheteurs    Clients



# Les principaux avantages du logiciel libre sont le coût, la performance, la fiabilité et l'ouverture

## Coût total

- **Coût des licences** : moins élevé car les licences libres permettent la copie du logiciel
- **Matériel** : les logiciels libres peuvent généralement fonctionner avec du matériel standard de type PC Intel
- **Maintenance** : La plus grande fiabilité permet de réduire ce coût

Les réductions de coût réalisées grâce aux logiciels libres sont entre 1/4 et 1/3 selon les témoignages

## Performance

- **Qualité technique du code**, relu et corrigé par des centaines de programmeurs, écrit avec l'accent sur la fiabilité et la performance
- **Flexibilité** : La libre disposition du code source et la logique de composants permet d'adapter le logiciel à l'utilisation, donc de l'alléger et d'en optimiser les performances

## Fiabilité et sécurité

- **Dysfonctionnements corrigés à la source**: Les développeurs peuvent comprendre un dans le logiciel et en corriger la source, le code est donc plus "propre"
- **Sécurité** : Les failles de sécurité sont détectées et corrigées plus rapidement
- **Transparence**: Pas de "backdoor" (trou de sécurité intentionnel) dissimulé dans le code source (point important pour le secteur public et les entreprises "sensibles")

## Ouverture et Interopérabilité

- **Absence de dépendance vis à vis d'un éditeur** : La libre disposition du code source permet de ne pas dépendre d'un éditeur particulier pour l'évolution ou la maintenance d'une solution
- **Des logiciels conçus pour communiquer** : Les logiciels libres sont généralement conçus pour communiquer **grâce aux standards Internet en natif**. Au contraire, les éditeurs ont généralement ajouté une couche Internet supplémentaire sur leurs logiciels propriétaires

Source CGEY/Net2002

## Il existe des alternatives libres pour la plupart des types de logiciels d'infrastructure, de communication ou de portail internet.

Logiciel	Outil propriétaires les plus répandus	Principales alternatives libres
• <b>Serveurs web</b>	• Microsoft IIS, Netscape	• Apache
• <b>Serveurs de messagerie</b>	• Microsoft Exchange, Lotus Notes	• Sendmail, Qmail, Postfix, Jabber (messagerie instantannée), IMP (Webmail)
• <b>Systèmes de gestion de base de données</b>	• Oracle, Microsoft (Access...)	• My SQL , PostgreSQL
• <b>Systèmes d'exploitation</b>	• Windows (NT, CE, XP...), Unix	• Linux, Open BSD
• <b>Annuaire</b>	• iPlanet	• Open LDAP
• <b>Autres outils d'infrastructure</b>		• Serveurs de noms de domaines (DNS): BIND; • Serveurs Proxy Cache: Squid; • Serveur FTP: WU-FTPd ; • Serveurs de fichiers et d'impression: SAMBA;
• <b>Sécurité</b>	• PKI: Baltimore, Verisign, • Firewall: • VPN: • Outils Symantec...	• PKI: IDX-PKI • Firewalling, filtre de paquets: IP/Netfilter, Ipchains • VPN: vtun (Virtual TUNnel) • Outils d'évaluation de sécurité: Nessus, nmap, SAINT, • Outil d'administration sécurisé: OpenSSH
• <b>Travail collaboratif</b>	.Lotus Notes	• PHPGroupware, Sympa (listes), Mioga, IIN(news)
• <b>Bureautique</b>	• Microsoft Office	• StarOffice
• <b>Langages et frameworks</b>	• C, C++, C#	• Perl, PHP,Python, Java, Zope, CVS
• <b>Clustering, Haute Disponibilité</b>		• IPVS/LVS (répartition de charge) , Heartbeat (Haute dispo.)
• <b>Outils d'administration</b>		• Webmin
• <b>Serveurs d'application web</b>	• Websphere (IBM), Weblogic (BEA)	• TOMCAT, Enhydra, Jboss, Roxen, IMHO

## Règles que les clients doivent désormais connaître et intégrer jusque dans leurs contrats

Abréviation	Nom	URL	Exemple de projets
<b>GPL</b>	General Public License	<a href="http://www.gnu.org/copyleft/gpl.html">Http://www.gnu.org/copyleft/gpl.html</a>	Emacs, Gcc, Debian
<b>LGPL</b>	Lesser General Public License	<a href="http://www.gnu.org/copyleft/lesser.html">Http://www.gnu.org/copyleft/lesser.html</a>	OpenOffice, librairies KDE
<b>APL</b>	Apache Public License (Apache Software Foundation)	<a href="http://www.apache.org/License.txt">Http://www.apache.org/License.txt</a>	Apache HTTP Server
<b>QPL</b>	QT Public License (Trolitech)	<a href="http://www.trolitech.com/developer/licensing/qpl.html">Http://www.trolitech.com/developer/licensing/qpl.html</a>	Librairie QT
<b>NPL</b>	Netscape Public License	<a href="http://www.mozilla.org/MPL/NPL-1.1.html">Http://www.mozilla.org/MPL/NPL-1.1.html</a>	Navigator
<b>MPL</b>	Mozilla Public License	<a href="http://www.mozilla.org/MPL/MPL-1.1.html">Http://www.mozilla.org/MPL/MPL-1.1.html</a>	Mozilla
<b>BSD</b>	Berkeley Software License	<a href="http://www.opensource.org/licenses/bsd-license.html">Http://www.opensource.org/licenses/bsd-license.html</a>	FreeBSD, OpenBSD

Source :Benchmark Group 2001

# Mise en situation dans une logique historique sous trois angles

- ❑ Technologique :
  - ✓ la loi de moore
  
- ❑ Economique
  - ✓ un marché piloté par les offreurs
  
- ❑ Sociologique
  - ✓ Des mutations rapides, l'importance du comportement humain (la cathédrale et le bazar)

# La cathédrale et le bazar (Eric Raymond)

## *les 19 leçons de l'expérience Fetchmail (1/2)*

1. Tout bon logiciel commence par gratter un développeur là où ça le démange.
2. Les bons programmeurs savent quoi écrire. Les grands programmeurs savent **quoi réécrire** (et réutiliser).
3. « **Prévoyez d'en jeter**, car de toute manière, vous le ferez. »
4. Si vous avez la **bonne attitude**, les problèmes intéressants viendront à vous
5. Quand un programme ne vous intéresse plus, votre dernier devoir à son égard est de le confier à un **successeur compétent**
6. **Traiter vos utilisateurs en tant que co-développeurs** est le chemin le plus sûr vers une amélioration rapide du code et un débogage efficace
7. Distribuez tôt. Mettez à jour souvent. Et soyez à l'**écoute** de vos **clients**.
8. Étant donné un ensemble de bêta-testeurs et de co-développeurs suffisamment grand, tous les bogues sautent aux yeux. » C'est ce que j'appelle : « **La Loi de Linus** ».
9. Il vaut mieux avoir des **structures de données intelligentes** et un code stupide que le contraire.
10. Si vous traitez vos bêta-testeurs comme ce que vous avez de plus cher au monde, ils réagiront en devenant effectivement ce que vous avez de plus cher au monde.

# La cathedrale et le bazar (suite)

- 11 . Il est presque aussi important de savoir **reconnaître les bonnes idées de vos utilisateurs** que d'avoir de bonnes idées vous-même. C'est même préférable, parfois.
- 12 Bien souvent, les solutions les plus innovantes, les plus frappantes, apparaissent lorsque vous réalisez que votre approche du problème était mauvaise.
- 13 « **La perfection** est atteinte non quand il ne reste rien à ajouter, mais quand **il ne reste rien à enlever.** »
- 14 **Tout outil doit être utile par rapport aux utilisations qu'il a été prévu d'en faire.** Mais on reconnaît un outil vraiment excellent au fait qu'il se prête à des usages totalement insoupçonnés.
- 15 Quand vous écrivez un logiciel jouant le rôle d'une passerelle quelconque, prenez soin de perturber le moins possible le flot de données ? et ne perdez *jamais* d'éléments d'information, à moins que la machine destinataire vous y oblige !
- 16 Quand votre langage est loin d'être Turing équivalent [7](#), un peu de « **sucre syntaxique** » ne peut qu'aider.
- 17 Un système de **sécurité n'est pas plus sûr que le secret (la clé) qui le garde.** Gare aux pseudo secrets
- 18 Pour résoudre **un problème intéressant, commencez par trouver un problème qui vous intéresse.**
- 19 Pour peu que **le coordinateur du développement dispose d'un moyen de communication au moins aussi bon que l'Internet, et pour peu qu'il sache comment mener ses troupes sans coercition,** il est inévitable qu'il y ait plus de choses dans **plusieurs têtes que dans une seule.**

# Il reste des défis à relever !

- ❑ le LL et l'Open Source : encore un « acte de foi » ?
- ❑ Un problème d'offre persistant :
  - Professionnalisme / pérennité des structures.
  - sortir d'une logique de « développeur »
  - quels partenaires en région ?
- ❑ Maîtrise du risque :
  - résistance des agents (point dur = poste utilisateur)
  - quelle évaluation du coût réel ?
- ❑ Quel accompagnement pour la conduite du changement ?

# Questions liées à l'Open Source et au logiciel libre

- ❑ Développeurs à temps libre
  - ▮ Solution réactive et évolutive ?
- ❑ Offre diverse et variée
  - ▮ Solution adaptée ?
- ❑ Garantie de fonctionnement
  - ▮ Solution aboutie et éprouvée ?
- ❑ Problématique de compatibilité et d'interopérabilité
  - ▮ Solution compatible et intégrable ?
- ❑ Problématique de support
  - ▮ En cas de problème ?

# Quels critères d'évaluation ?

- ❑ Dynamisme
  - ▢ Historique du développement
  - ▢ Nombre de programmeurs
- ❑ Maturité
  - ▢ Expansion : répertorier les utilisateurs, support commercial
  - ▢ Retours des utilisateurs
  - ▢ Aboutissement : fonctionnalités, stabilité des versions
- ❑ Interopérabilité
  - ▢ Compatibilité avec les standards
  - ▢ Multi plates-formes
- ❑ Packaging
  - ▢ Support et documentation, forums, listes de discussion
  - ▢ Ergonomie
  - ▢ Gestion et installation

Merci