



# Droit des logiciels

En tout cas, un petit aperçu...

François Pellegrini

EQUIPE PROJET

**BACCHUS**

Bordeaux

Sud-Ouest

10/06/2013

# Droit et informatique

- Droit et informatique sont des disciplines de même nature
  - Elles consistent à construire, à partir de quelques éléments de base, les règles de fonctionnement d'univers complets
- Le problème, c'est que ces univers sont l'univers dans lequel nous vivons !
  - Les générations précédentes vivaient dans le « monde de Kafka »
  - Notre génération vit également dans le « monde de Matrix »
- Intérêt d'une compréhension mutuelle entre informaticiens et juristes

# Émergence du logiciel comme objet juridique

# Apparition du logiciel

- Les premiers calculateurs ne possédaient pas de logiciel
  - Programmés « à la main »
- Le logiciel (et l'ordinateur) sont apparus avec l'idée que le programme pouvait être stocké dans la même mémoire que celle utilisée pour les données qu'il manipule
  - Architecture dite « de Von Neumann »
  - Le logiciel est une donnée (presque) comme une autre !
  - Possibilité de programmes auto-modifiables
    - Méta-programmation

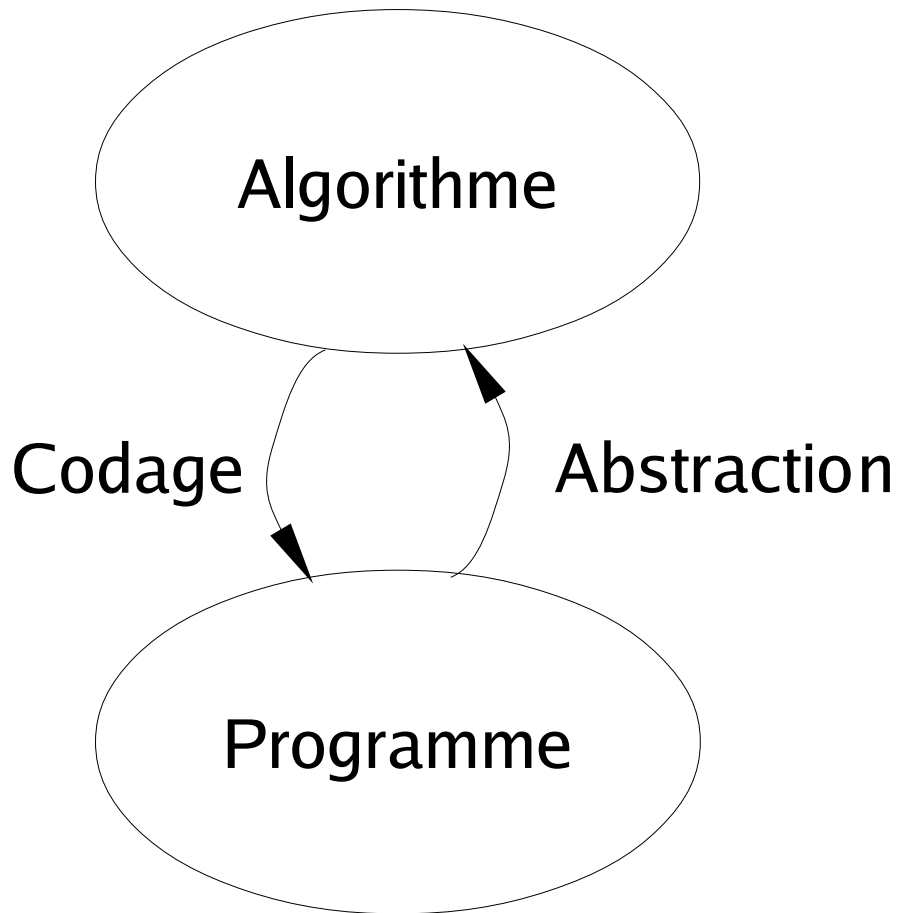
# Apparition du droit du logiciel (1)

- Tant que les logiciels étaient spécifiques à un ordinateur donné, la question de leur statut ne se posait pas
  - Les logiciels étaient fournis « gratuitement » aux utilisateurs, en tant que « fournitures annexes », au même titre que les manuels d'utilisation
- Les fournisseurs encourageaient leurs clients à modifier les logiciels et à partager les modifications
  - Soutien à la création de « clubs d'utilisateurs » servant à l'échange de ces améliorations
    - Mutualisation et donc réduction des coûts de maintenance logicielle
    - Avantage compétitif sur les concurrents
  - Les principes du logiciel libre avant l'heure !

## Apparition du droit du logiciel (2)

- La question du statut du logiciel s'est posée lorsque sont apparus les premiers ordinateurs compatibles avec les grands systèmes IBM, à la fin des années 1960
  - Les fournisseurs de matériels compatibles utilisaient les logiciels d'IBM sans en payer le prix ni même le coût
- IBM décida donc de facturer séparément logiciel et matériel
  - Politique de « dégroupage » (« *unbundling* »)
  - Mais comment caractériser juridiquement le logiciel ?

# Nature du logiciel



- Les algorithmes sont :
  - Des idées
  - Des mathématiques
- Les programmes sont :
  - Des œuvres de l'esprit
  - Du discours
    - Humain ↔ humain
    - Humain → ordinateur
  - Des processus, lorsqu'ils sont exécutés

- Similaire au processus de création littéraire

# Quel statut pour les langages ?

- Un langage n'est pas une œuvre
  - Il permet d'écrire des œuvres
  - Il est d'un niveau d'abstraction supérieur
    - Pas de revendication possible sous le régime du droit d'auteur
- Un langage informatique est aussi une langue de communication humaine
  - Permet l'échange d'informations entre humains
- Les langages informatiques ne peuvent donc être appropriés sans porter préjudice à des droits plus élevés
- Un format de fichiers définit la grammaire d'un langage
  - Accorder des brevets sur les formats de fichiers reviendrait donc à pouvoir monopoliser un langage



# Quel statut pour les logiciels ?

- Trois voies offertes au législateur :
  - Brevet
    - Instrument inadapté : portée, coûts de transaction, etc.
    - Mais certains veulent y revenir...
  - Droit *sui generis*
    - Durée d'établissement des droits nationaux et conventions internationales
  - Droit d'auteur
    - Problèmes à la marge mais applicabilité immédiate
    - Convention de Berne de 1886 déjà signée par plus de 170 pays

# Droit d'auteur adapté

- Par son rattachement au droit d'auteur, le logiciel est assimilé à une œuvre de l'esprit
  - En France, loi du 3 juillet 1985
  - Directive européenne 91/250/CE (1991)
  - Article 10 des accords ADPIC (1994)
  - Article 4 du traité OMPI WCT (1996)
- Le logiciel est cependant aussi un produit substituable voué à rendre un service
  - Adaptation du droit d'auteur (« droit d'auteur adapté »)
  - Question de la garantie

[http://www.wto.org/french/docs\\_f/legal\\_f/27-trips\\_04\\_f.htm#droit](http://www.wto.org/french/docs_f/legal_f/27-trips_04_f.htm#droit)  
<http://www.wipo.int/documents/fr/diplconf/distrib/94dc.htm>

# Adaptation des droits patrimoniaux (1)

- Ajout de nouvelles exceptions
  - « Actes nécessaires pour permettre l'utilisation du logiciel »
  - « Observation du fonctionnement »
    - En fait, simple rappel de la loi
  - Copie de sauvegarde
    - Si aucun autre moyen fourni par l'éditeur
  - La décompilation
    - Interdite sauf « à fin d'interopérabilité »
- Suppression de l'exception de copie privée
  - Nécessite une licence pour chaque exemplaire du logiciel

# Adaptation des droits patrimoniaux (2)

- Pour les auteurs salariés ou fonctionnaires, transfert automatique de la titularité des droits patrimoniaux à l'employeur
  - C'est l'employeur qui décide de la vie de l'œuvre :
    - Divulgation, choix de la licence, etc.
  - Apparition d'un statut d'« auteur prolétaire »
- Les auteurs non salariés restent titulaires des droits patrimoniaux sur leurs œuvres logicielles
  - Nécessité de transférer les droits des sous-traitants
  - Problème des stagiaires non rémunérés

# Adaptation des droits extra-patrimoniaux

- Amoindrissement des droits extra-patrimoniaux
  - L'auteur salarié ou fonctionnaire ne peut s'opposer à la modification de l'œuvre
  - Reste le droit au nom...
    - Plus courant dans le monde du jeu vidéo que de la comptabilité !

# Compilation, décompilation et interopérabilité

# Langages de bas et haut niveau (1)

- Les ordinateurs ne comprennent que le « langage machine »
  - Instructions très rudimentaires
  - Spécifique à un processeur donné
    - Et à un système d'exploitation donné
  - Code très peu expressif
  - Risques de bogues élevé
    - Dépend du nombre de lignes que l'on écrit

# Langages de bas et haut niveau (2)

- Besoin d'écrire des programmes dans des langages plus expressifs
  - Meilleure compréhension du code
  - Indépendance vis-à-vis du type de processeur
  - Meilleures maintenabilité et portabilité
  - Coût économique bien inférieur



# Compilation et décompilation (1)

- La compilation consiste à traduire un programme écrit dans un langage de haut niveau en un langage de bas niveau, susceptible d'être exécuté par un ordinateur
  - Le « code source » est le programme écrit en langage de haut niveau que l'on veut traduire
    - La « forme préférée » d'écriture d'un programme
  - Le « code objet » est le programme résultant écrit dans un langage de bas niveau
    - Définit le « programme exécutable »

# Compilation et décompilation (2)

- On appelle « décompilation » l'action inverse
  - Bien plus difficile à mettre en œuvre
  - Informations structurelles de haut niveau « diluées » dans le code objet

# Traduction automatique

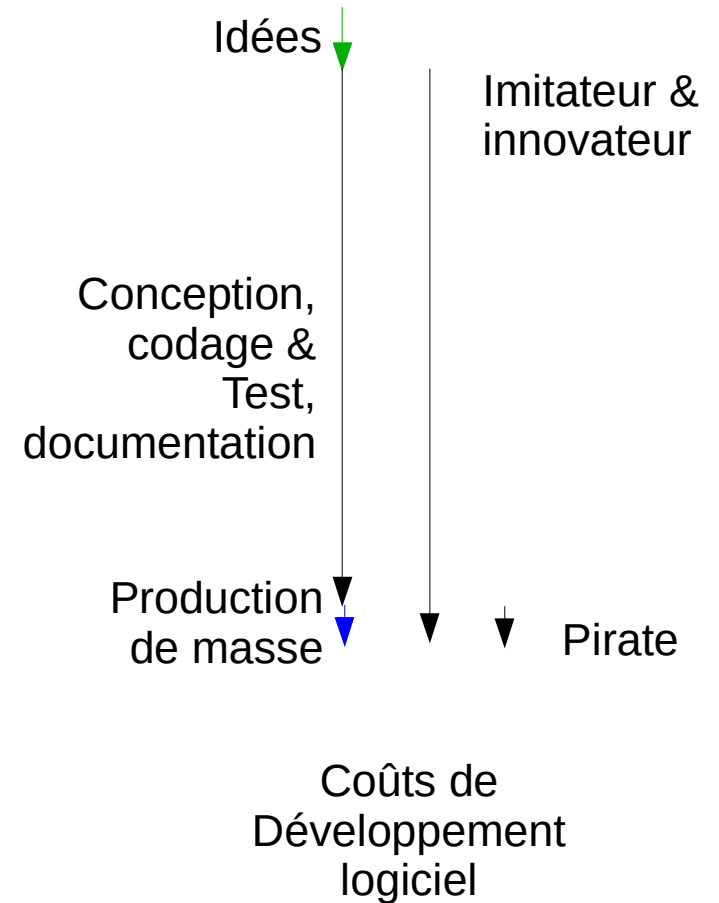
- Du fait que la traduction du code source en code objet se fait de façon automatique, le code objet est une œuvre intégralement dérivée du code source, sans aucun apport original d'un autre auteur
  - Le créateur d'un outil n'a aucun droit sur les créations réalisées au moyen de cet outil
    - Pensez aux dictionnaires, pinceaux, etc...
    - Cas particulier : inclusion de fragments de code du compilateur dans le programme exécutable
  - Hypothèse à revoir quand existeront des intelligences artificielles
    - « Valladolid 2.0 » !

# Interdiction de la décompilation

- Compiler et décompiler créent des œuvres dérivées de l'œuvre originale
- Ces copies ne peuvent donc être exploitées qu'avec l'accord de l'ayant droit des œuvres originales
- La décompilation est donc interdite, sauf permission explicitement donnée par l'ayant-droit

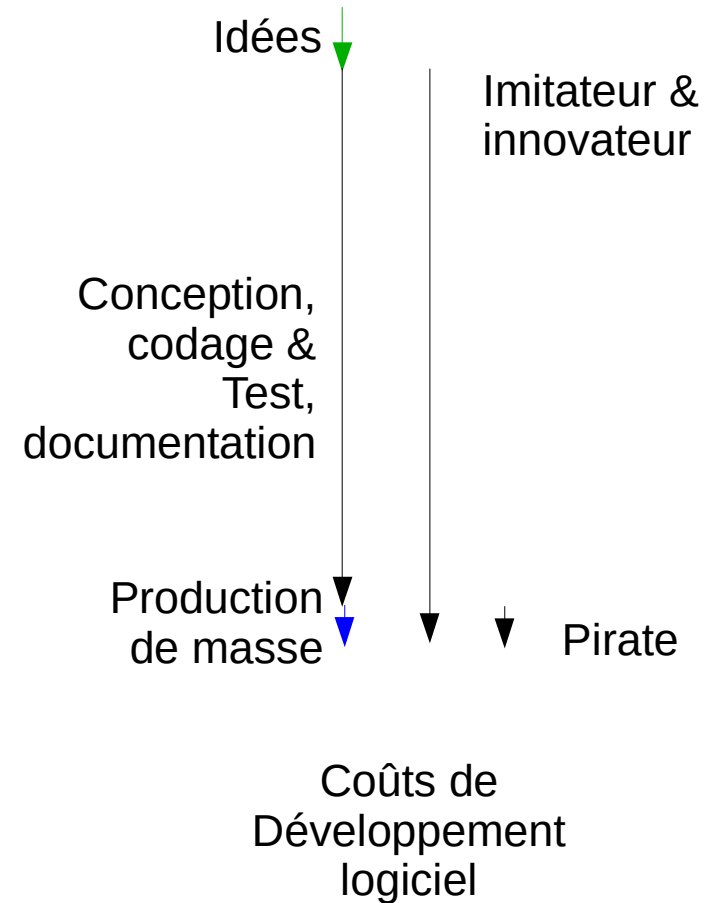
# Modèle économique du logiciel (1)

- L'observation des fonctionnalités d'un logiciel existant permet à un nouvel arrivant de réaliser un logiciel concurrent
- Il n'y a concurrence libre et non faussée que s'il paye les mêmes coûts d'entrée sur le marché en recodant son logiciel
  - Donc il ne peut réutiliser le code existant, par décompilation et/ou copie servile



# Modèle économique du logiciel (2)

- Pendant ce temps, l'innovateur original peut progresser
  - S'il ne le fait pas, il sera dépassé par les innovations des autres

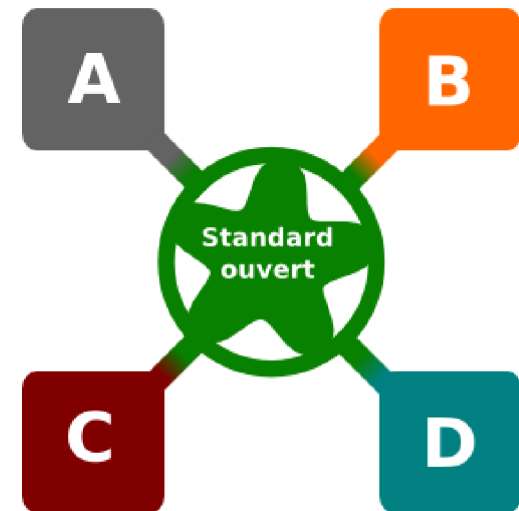
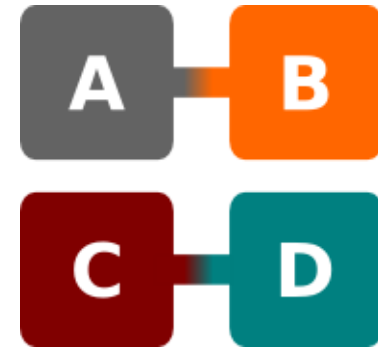


# Interopérabilité

- Les formats de fichiers créent des marchés captifs
  - La vraie valeur est dans les données !
- Possibilité de décompiler « à fin d'interopérabilité »
  - Afin de garantir l'existence de la concurrence libre et non faussée
    - Lutter contre les effets de réseau
  - Strictement encadrée par la loi

# Compatibilité n'est pas interopérabilité !

- Il y a compatibilité quand deux produits peuvent fonctionner ensemble
- Les formats de fichiers créent des marchés captifs
  - Accord contractuel entre les parties
  - L'entité contrôlant le format « compatible » étend son monopole
- Il y a interopérabilité quand deux produits peuvent fonctionner ensemble et que l'on sait pourquoi





# La recherche de l'interopérabilité

- Spécificité bienvenue du droit européen
  - Instaurée par la directive 91/250/CE
- La décompilation des logiciels est interdite sauf afin de rechercher l'interopérabilité (L.122-6-1 IV°)
- Encadrement strict par trois conditions :
  - Actes accomplis par une personne ayant le droit d'utiliser le logiciel ou mandatée à cette fin
  - Informations non déjà disponibles par ailleurs
  - Actes limités aux parties du logiciel nécessaires
- Les informations obtenues ne peuvent être utilisées à d'autres fins et « porter atteinte au droit d'auteur »

# Statut des interfaces

- La libre concurrence suppose qu'on puisse interfacier un nouveau logiciel avec un logiciel existant
  - Voire le remplacer en s'interfaçant avec les logiciels tiers avec lesquels ce logiciel interagissait
- Or, lorsqu'on veut s'interfacier avec un logiciel existant, il faut respecter ses interfaces
  - Décrites dans des fichiers d'en-tête, des fichiers de classes, etc
- Pour s'interfacier, il faut donc « recopier » le format des interfaces
  - Est-on un contrefacteur au sens du droit d'auteur ?

# Licences

# Licence

- La licence est une offre de contrat de la part du fournisseur, qui définit les conditions d'utilisation d'une œuvre
  - Le terme juridique exact est : « pollicitation »
- Basée sur les droits d'auteur ou le copyright
  - Convention de Berne de 1886
- Classiquement, une licence limite les droits d'usage d'une œuvre :
  - Interdiction de diffusion publique
  - Interdiction de reproduction, même partielle
  - ...

# Licences de logiciels (1)

- Il existe de nombreux types de licences de logiciels :
  - « Logiciel propriétaire » (« privatif ») : tous les droits sont réservés par leur titulaire
    - Cas de la majorité des logiciels du commerce
    - Dans la quasi totalité des cas, pour ces logiciels :
      - Le client n'est propriétaire que du support, et pas du logiciel qu'il contient
      - Le fournisseur dégage toute responsabilité en cas de vices cachés (bogues)
      - Le fournisseur peut arrêter la maintenance du logiciel à tout moment

## Licences de logiciels (2)

- « Partagiciel » (« *Shareware* ») : logiciel privatif diffusable gratuitement mais pour lequel une contribution est demandée au bout d'une période d'essai
  - Logiciel privatif où seul diffère le mode de distribution
- « Gratuiticiel » (« *Freeware* ») : logiciel privatif gratuit mais ne donnant pas nécessairement d'autres droits
  - Parfois pas même celui de redistribution

# Licences de logiciels (3)

- « Logiciel libre » (« *Free software* ») : logiciel donnant de nombreux droits à ses utilisateurs
  - N'est **pas** équivalent à un « *freeware* » !
  - Pas nécessairement gratuit
- « De domaine public »

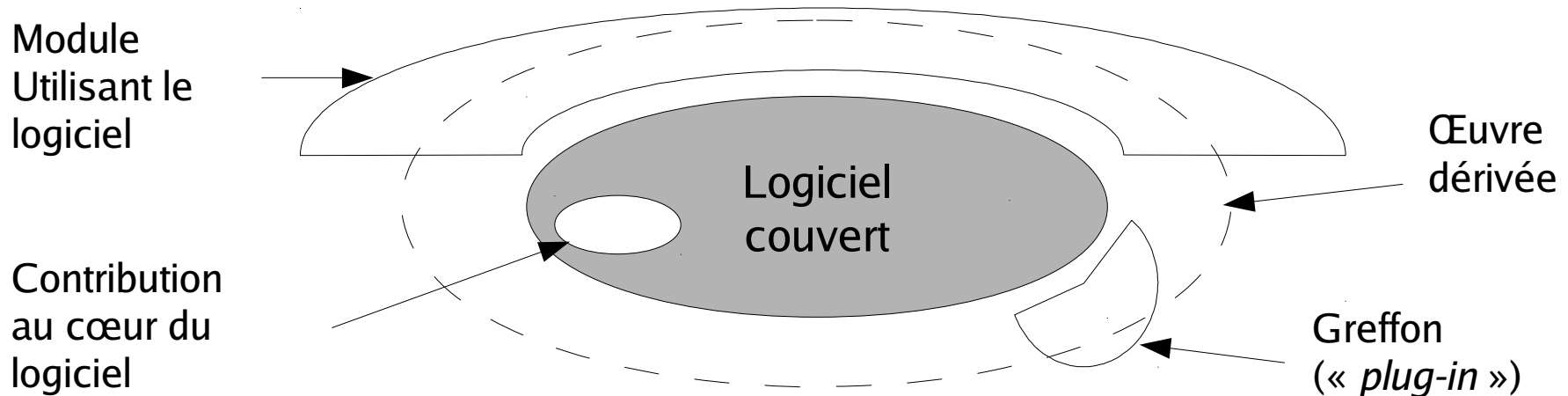
# Les licences libres (1)

- Ont en commun les « quatre libertés »
  - Liberté d'utilisation pour tout usage
  - Liberté de copie du logiciel original
  - Liberté de modification du code source
    - Implique la liberté de consultation
  - Liberté de redistribution du logiciel modifié
- Les différences entre licences libres portent sur les modalités de redistribution du logiciel modifié :
  - Si persistance des obligations de redistribution du code source : « *copyleft* »
  - Si absence de cette obligation : « *non copyleft* »



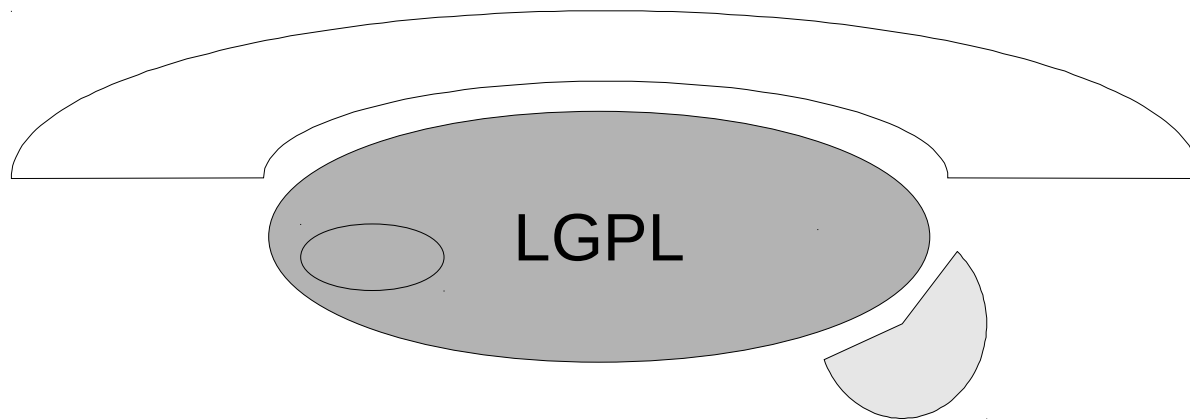
# Les licences libres (2)

- Trois principaux types de licences libres :
  - Licences « persistantes »
  - Licences « évanescentes »
  - Licences « diffusives »
- Les licences s'appliquent à trois entités, le plus souvent au travers de la notion d'œuvre dérivée :



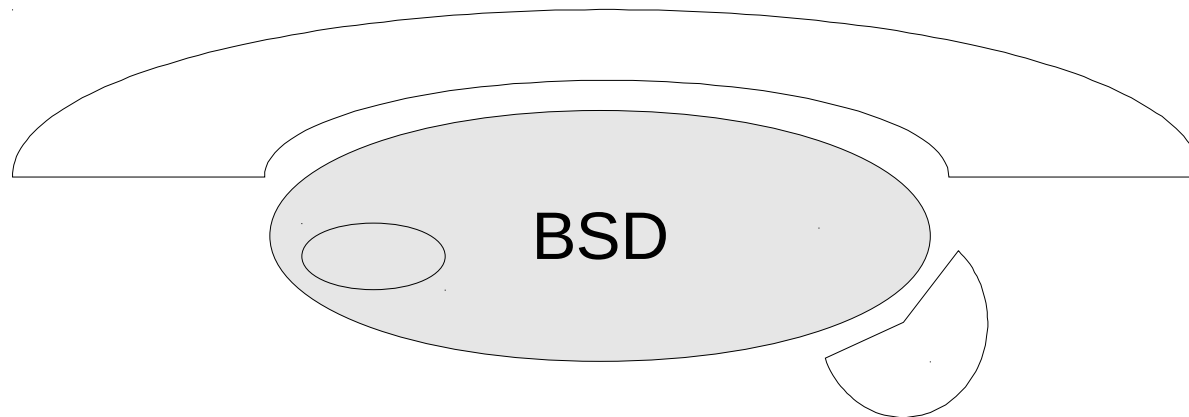
# Licences « persistantes »

- Aussi appelées : « pérennes », « à *copyleft* faible »
- Exemples : LGPL, CeCILL-C
- Le code source des versions modifiées doit être rediffusé lorsque celles-ci le sont (pérennité)
- Il peut néanmoins être combiné à des logiciels soumis à d'autres licences, y compris des logiciels à sources fermés



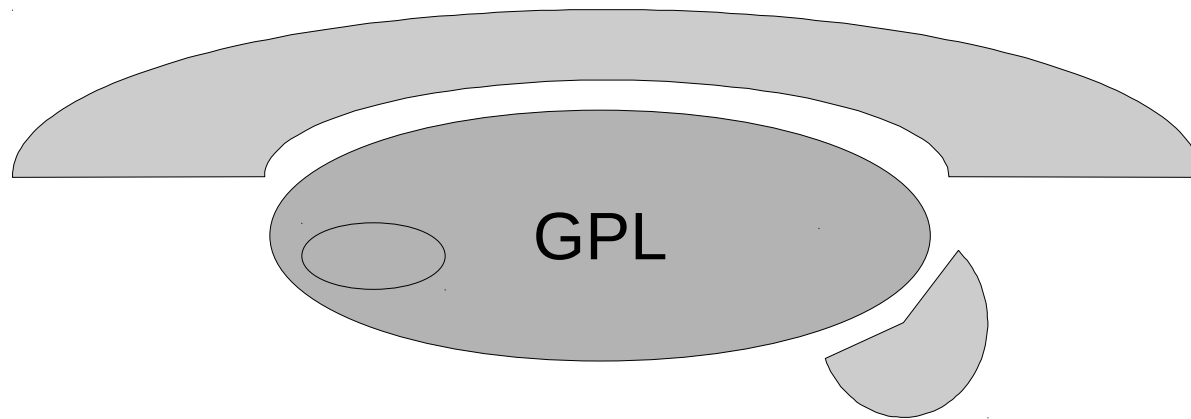
# Licences « évanescentes »

- Aussi appelées : « permissives », « non *copyleftées* »
- Exemples : BSD, CeCILL-B
- Le logiciel diffusé sous forme binaire peut être redistribué selon d'autres termes de licences
  - La CeCILL-B permet de changer la licence du source
- Il peut donc même être « refermé » et distribué, sous forme originale ou modifiée, sans son code source



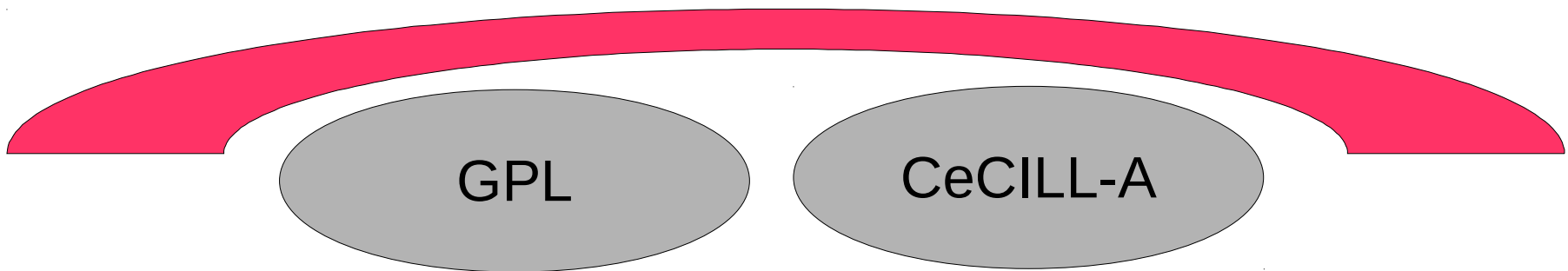
# Licences « diffusives » (1)

- Aussi appelées : « à *copyleft* fort », ou parfois « contaminantes » (terme non neutre, à éviter !)
- Exemples : GPL, CeCILL(-A)
- Le code source des versions modifiées doit être rediffusé lorsque celles-ci le sont
- Les logiciels tiers fortement liés doivent être diffusés selon les mêmes termes



## Licences « diffusives » (2)

- Les licences diffusives peuvent induire des conflits « en diamant » lorsque l'on cherche à lier ensemble deux modules sous licences diffusives différentes



- Dans tous les cas, nécessité d'une analyse juridique préalable des composants logiciels que l'on souhaite utiliser au sein de son logiciel, en fonction des modes de diffusion souhaités

## Les licences libres (3)

- L'ayant droit d'un logiciel peut choisir de diffuser celui-ci avec le type de licence de son choix
- Il peut même diffuser le même code source, par deux canaux différents, avec des licences différentes
  - Politique de licences multiples : « *dual licensing* »
  - Attention au suivi des versions et contributions successives !

# Les « brevets logiciels »

# Portée du droit d'auteur adapté

- Le droit d'auteur adapté garantit qu'on ne peut :
  - Copier un programme pour le donner ou le vendre
  - (Essayer de) le modifier
  - L'utiliser en dehors des clauses stipulées par sa licence
- Le droit d'auteur adapté n'interdit en revanche pas d'écrire un nouveau programme :
  - Aux fonctionnalités similaires
  - Compatible au niveau des formats d'entrée/sortie
  - Interopérable avec le programme original



# Protection du logiciel en Europe

- Les programmes informatiques sont protégés par le droit d'auteur (copyright)
  - Directive européenne 91/250/CE
- Les logiciels sont explicitement exclus du champ de la brevetabilité par l'article 52§2c de la Convention de Munich (1973), comme les jeux et les maths
  - Exclusion limitée au logiciel « en tant que tel »
  - Les processus industriels innovants utilisant du logiciel sont brevetables si l'innovation ne réside pas dans le logiciel lui-même, mais le logiciel employé, en tant que tel, est exclu des revendications du brevet

# Portée du brevet logiciel

- Les brevets logiciels ne concernent pas directement les programmes
- Ils protègent les concepts sous-jacents tels que :
  - Ce qu'un programme fait :
    - Quel problème concret ce programme résout, c'est-à-dire quelle « *business method* » il implémente
  - Comment il le fait, et plus particulièrement :
    - Quelles données d'entrée il accepte
    - Quelles données de sortie il produit
    - Comment il interagit avec d'autres programmes

# Champ d'action du brevet logiciel

- Le brevet logiciel concerne directement :
  - Les « *business methods* » (« algorithmes du monde réel »)
    - Étapes suivies par un utilisateur achetant des biens sur Internet (brevet « 1-Click » d'Amazon), ...
  - Les formats de fichier
    - Documents, feuilles de calcul, images, sons (MP3), ...
  - Les protocoles de communication entre programmes
  - Les algorithmes
    - Chiffrement des données, ...
- Ces concepts sont logiquement indissociables

# Exemples de brevets délivrés par l'OEB mais pour le moment encore invalides

L'Urgence du E-Commerce Européen

<http://webshop.ffii.org/>

## Votre boutique web est BREVETÉE!

**4** Musiques **Films** **Livres**

**1** **2** **NOUVEAUTÉ: COMMANDEZ PAR TÉLÉPHONE PORTABLE!** **Obtenez de l'aide directement depuis nos bases de données de support internes!** **?**

**15** Voir dans le navigateur **7** Vous aimez ce résultat de recherche? Vous aimerez sans doute ceux-ci:

**17**  Les coccinelles sont des insectes très utiles. Elles se débarrassent des parasites. Cependant les services contentieux des brevets logiciels sont en général bien trop gros pour elles.

**8** **Exclusif: téléchargez immédiatement vos achats!**

**6**  Achetez la musique originale (mp3) **8**

Achetez le film **6**

**19** <Entrez votre réduction>

**3** Ajoutez au panier **10**

Envoyez comme cadeau **10**

Demandez un prêt **11**

**9** Paiement par carte de crédit **12** 

**13**  Oui, je veux recevoir vos offres! **14**

**5** **16**  Cliquez ici pour un plus grand aperçu

**20** **5** Rendez-vous dans l'un de nos magasins et composez/gravez votre propre DVD à la carte!

**18** **1** La coccinelle à Monté-Carlo  
**2** Coller opter  
**3** Coque de scie  
**4** Brevets logiciels et autres parasites  
**5** Tu me cherches des poux

Aperçu de quelques chapitres: cliquez dans la télé ci-dessus!

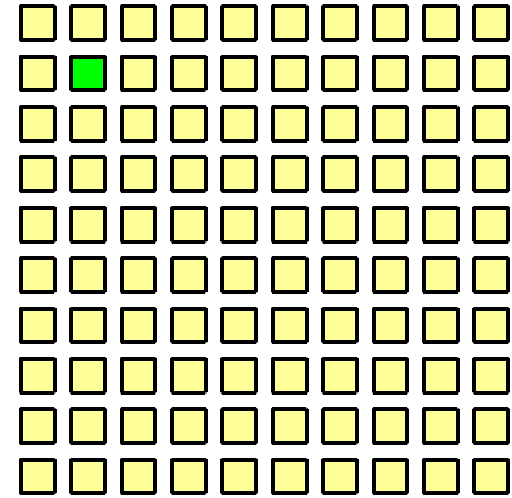
*Si nous n'avons pas votre commande en stock, elle sera immédiatement transmise à l'un de nos vendeurs affiliés!*

<http://webshop.ffii.org/>

# Économie du brevet logiciel (1)

- Rêves de fortune

- Mon invention brevetée
- Techniques de programmation évidentes

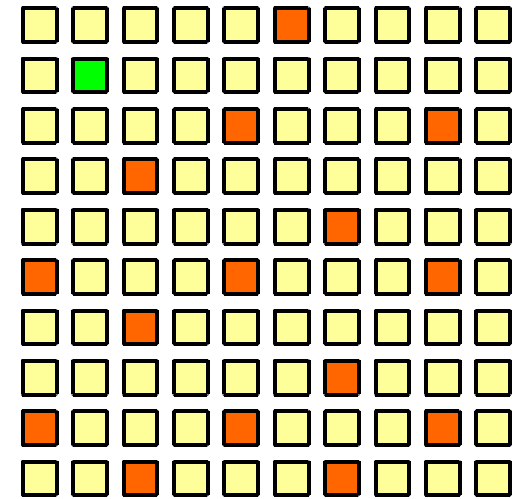


$$\text{Profit} = \text{Ventes} - \text{Develop} + \underbrace{\text{Revenu} (\text{■}) - \text{Coût} (\text{■})}_{> 0}$$

# Économie du brevet logiciel (2)

- Batailles de brevets logiciels

- Mon invention brevetée
- Techniques non brevetées
- Risque de poursuite pour contrefaçon de brevet



$$\text{Profit} = \text{Ventes} - \text{Develop} + \underbrace{\text{Revenu (■)} - \text{Coût (■)} - \text{Coût (■)}}_{< 0}$$

# Économie du brevet logiciel (3)

$$\text{Profit} = \text{Ventes} - \text{Develop} + \underbrace{\text{Revenu (■)} - \text{Coût (■)} - \text{Coût (■)}}_{\text{En moyenne} < 0}$$

## Petit éditeur :

Doit payer

$$\underbrace{\text{Revenu (■)} - \text{Coût (■)}}_{< 0} - \underbrace{\text{Coût (■)}}_{< 0}$$

## Gros éditeur :

Neutralise les brevets concurrents

$$\underbrace{\text{Revenu (■)} - \text{Coût (■)}}_{< 0} - \underbrace{\text{Coût (■)}}_{\approx 0}$$

## Fonds de brevets :

Vit sur le dos du système

$$\underbrace{\text{Revenu (■)} - \text{Coût (■)}}_{\text{En principe} > 0} - \underbrace{\text{Coût (■)}}_{= 0}$$



# Qui survit ?

- Les grands groupes
  - Échanges de portefeuilles de brevets
  - Position de quasi-monopole
- Les fonds de brevets
- Les petites entreprises désirant se faire racheter
  - Leurs brevets augmentent leur valeur d'achat supposée
  - Pas de désir personnel d'innovation soutenue
- Les avocats, les experts en propriété industrielle, les offices de brevets
  - Consomment entre 10 et 40 % des moyens initialement destinés à l'innovation



# Qui dépérit ?

- Les petits éditeurs de logiciels, généralement peu préparés aux batailles juridiques, par :
  - Manque de fonds (coût moyen d'un procès > 1 M\$)
  - Manque de portefeuilles de brevets à échanger
- Les entreprises utilisant du logiciel « maison » pour résoudre leurs « *business problems* », même si :
  - Ces entreprises n'appartiennent pas au monde du logiciel
  - Elles ne commercialisent pas leurs logiciels
- Les développeurs de logiciels Libres/Open-Source
  - Le code source est à disposition de la partie attaquante

# Menaces sur l'innovation et l'interopérabilité

- Si le détenteur d'un brevet logiciel refuse de céder des licences :
  - Il est illégal d'écrire des programmes qui lisent ou produisent des formats de données brevetés
  - Les utilisateurs ne peuvent plus basculer vers d'autres produits pour traiter leurs données existantes
  - L'entrée sur le marché de nouveaux acteurs est réduite
  - Le choix des produits et des fournisseurs est réduit

# Bilan du brevet « logiciel »

- Instrument conçu pour l'industrie matérielle et étendu abusivement aux méthodes intellectuelles
  - Monopolise les algorithmes (= mathématiques !)
- Illégal en Europe mais accordé par les offices, qui vivent des annuités
- Contresens économique
  - Instrument anti-concurrentiel
  - Coûteux à obtenir et à maintenir [Bessen & Hunt]
  - Menace le retour sur investissement
  - Favorise les délocalisations
  - À ne pas cautionner

# Conclusion

# Le droit des logiciels à l'heure actuelle

- Droit mouvant et encore à construire
- Interaction avec de nombreux autres droits
  - Droit des « mesures techniques de protection »
  - Droit des brevets
  - ...
- Sujet passionnant !