



[www.cnrs.fr](http://www.cnrs.fr)

# Typologie des licences, critères de choix

Patrick MOREAU

Mai 2018





# Plan



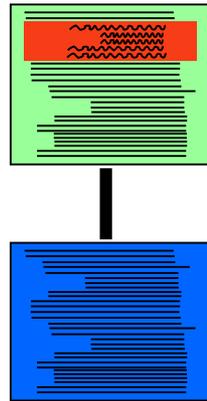
## Le logiciel libre

Le choix de licence

# Le logiciel libre / open source

Le logiciel libre est défini par quatre libertés fondamentales :

- exécuter le programme
- l'étudier
- l'adapter
- le redistribuer



```
1001011000101010100001001
0100010001010101001100001
01100001110110111000111000
```

## Logiciel Libre et Open Source

- presque équivalent
- correspond à des écoles de pensées différentes
  - . La liberté est mise en avant
  - . un modèle de développement

FLOSS pour « Free/Libre and Open Source Software »

# Le logiciel libre

## Le logiciel libre est

- exécuter le programme
- l'étudier
- l'adapter
- le redistribuer

## Logiciel Libre et

- presque équivalent
- correspond à
  - . La liberté
  - . un modèle

## FLOSS pour « Free

entales :

```
00101010100001001  
01010101001100001  
10110111000111000
```



»

# Pourquoi la recherche doit s'intéresser au libre

## Quelles incidences sur les facteurs clés de succès d'un projet ?

- Travailler avec les utilisateurs potentiels, si possible dès la phase de conception.
- Se confronter très tôt au marché, se mettre en position d'écouter en permanence les retours.

La recherche produit de nombreux logiciels libres  
quelque soit le domaine scientifique

Frais de propriété intellectuelle quasi nuls

Fluidification de la relation contractuelle



# Les grandes familles de licence

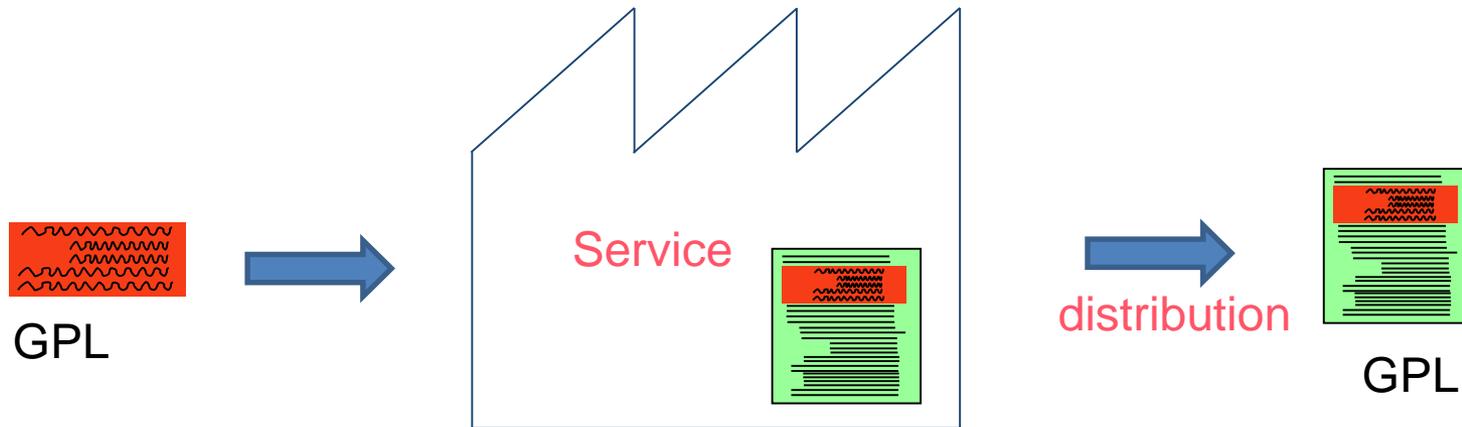
Nature	Catégorie	DERIVATION Modification et redistribution ?	COMPOSITION Agrégation et redistribution ?	Exemple de licence
NON - LIBRE		N'est pas automatique	N'est pas automatique	Freeware, shareware, licence gracieuse, licence de test, licence propriétaire, ...
Fort copyleft	non permissive en dérivation et composition  Obligation de divulguer	sous la même licence	sous la même licence	CeCILL, GNU GPL, EUPL  Affero-GPL dans le cas du SaaS
Faible copyleft	non permissive en dérivation, permissive en composition  Obligation de communiquer les dérivations	sous la même licence	OUI, y compris sous une licence différente	CeCILL-C, GNU LGPL, EPL, MPL
Non copyleft	Non copyleft  Aucune contrainte sur la communication	OUI, y compris sous une licence différente	OUI, y compris sous une licence différente	CeCILL-B, BSD, MIT, APACHE

# Une licence est un contrat

**Contrat** entre le(s) détenteur(s) des droits et un tiers

- Ce sont les termes de la licence qui font qu'un logiciel est libre

**Contrat** attaché au logiciel mais « dormant » si pas redistribution



# Logiciel libre et transfert/valorisation

## Logiciel libre = gratuit

- Oui mais existence d'un modèle économique (à définir) si il existe une offre et une demande

# Offre et demande



# Comment vendre quelque chose de gratuit



TEL QUEL



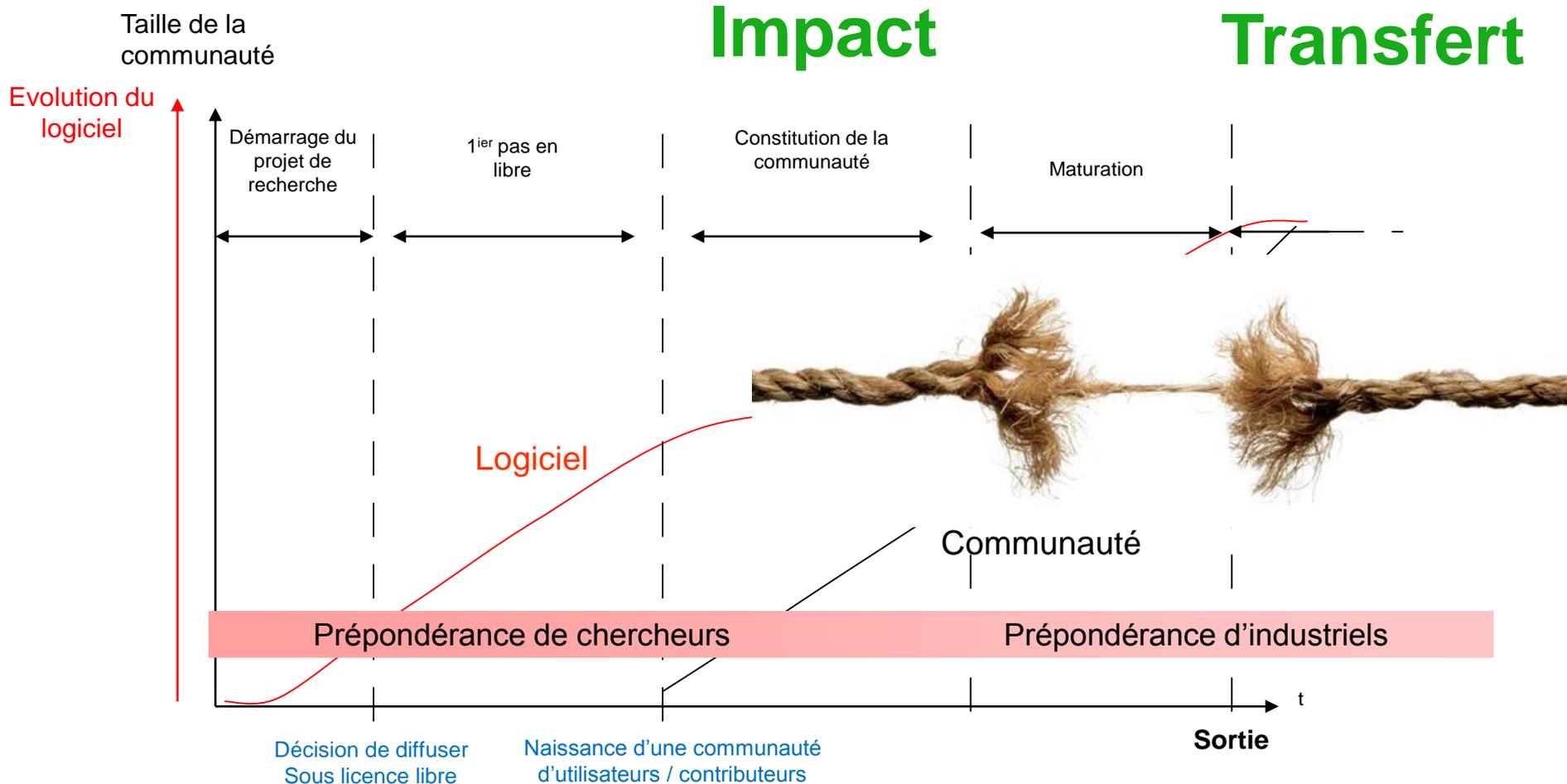
# La valorisation par le logiciel libre



- ⊙ Absence de flux financier direct **mais de multiples impacts économiques potentiels**
- ⊙ Transfert de savoir-faire
- ⊙ Contrats de collaboration de recherche / Laboratoire commun
- ⊙ Marque
- ⊙ Prise de part dans des start-ups
- ⊙ Création d'emplois
- ⊙ Parfois cession des droits patrimoniaux
  
- ⊙ En général, possibilité de schémas de licensing fins
  - Possibilité de licence propriétaire sur certains composants « métier »

# PRENDRE EN COMPTE LE TEMPS!

## Exploitation industrielle et commerciale





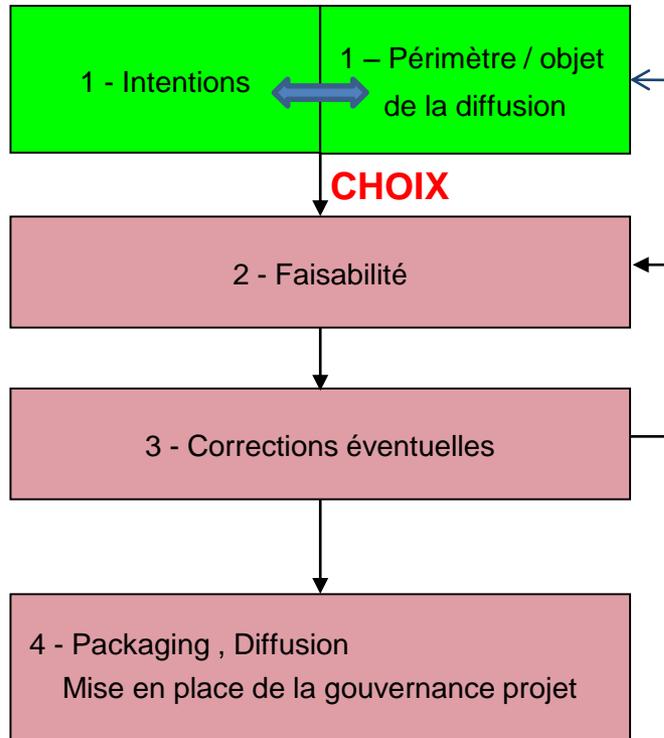
# Plan



Le logiciel libre

## Le choix de licence

# Etapes



Approche itérative entre  
l'intention / l'envie et le  
périmètre du logiciel

- | - Il existe des optimaux
- | - Chaque cas est **particulier**

# Envies ou intentions

Il est essentiel d'identifier les objectifs de diffusion de chaque partie prenante car objectifs parfois contradictoires

Partage scientifique

Capter une part de la valeur de l'actif (intéressement, retour financier pour l'institut)

Obtenir des contributions sur tout ou partie du code

Obtenir des retours utilisateurs

Devenir la communauté de référence

Visibilité

-

# ***Les choix***

# Vérifier l'adéquation entre l'objet et l'intention

Avoir une communauté d'utilisateurs

**MAIS**

Pas envie de prendre le temps pour supporter / incuber une communauté

Avoir des contributeurs

**MAIS**

Le logiciel est complexe et l'architecture n'est pas modulaire

Le logiciel doit fonctionner!

Le logiciel doit toujours être dans la roadmap de l'équipe

## **Règle proposée**

Ne pas octroyer au licencié plus de droits nécessaire à l'intention initiale

# Diffusion du logiciel sous licence libre

Choix de la licence libre

Prendre en compte la nature du logiciel

- bibliothèque, application, plug-in

Importance de la **culture** de la communauté cible

- par exemple, les logiciels d'infrastructure type « cloud » diffusés sous licence libre sont très souvent sous licence permissive type Apache

# Les grandes lignes pour le choix d'une famille de licence libre

## Logiciel autonome, logiciel applicatif, outil de simulation

- une licence type GPL/CeCILL ou encore Affero GPL, est préconisée elle demeure compatible avec une future valorisation sous forme de licence commerciale.
- . centralisation des droits / schéma de double licence.

## Bibliothèques, Plug-in, Module

- une licence type LGPL ou CeCILL-C est préconisée
  - . accès aux modification
  - . liberté d'utilisation par des acteurs économiques.

## Commodité scientifique, ou encore bien commun

- une licence de la famille BSD ou CeCILL-B

## Et quid des :

- Framework, middleware, système d'exploitation
- Environnement de développement
- logiciel embarqué, enfoui

# Tracer les décisions

Contexte

Décision

Montée en puissance en terme d'expertise

Donne la capacité de revisiter les choix en cas de changement de contexte

***La faisabilité de ces choix***  
***Corrections possibles***

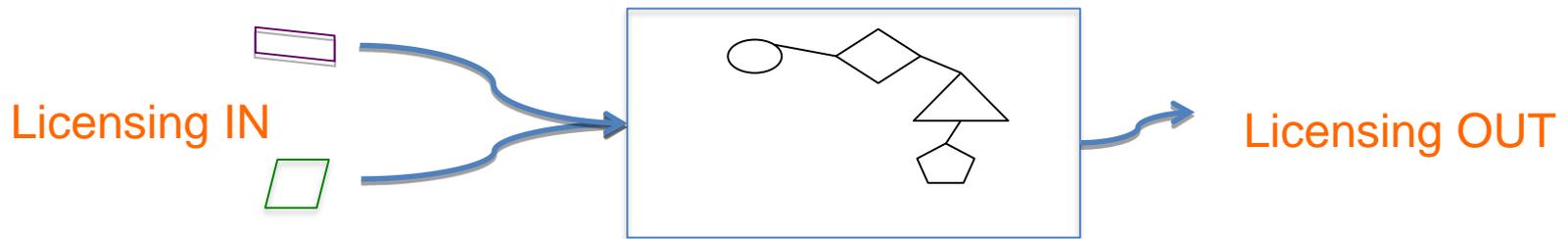
# Les problèmes de compatibilité

Il existe plusieurs méthodologies de développement

- Partir ex-nihilo (de plus en plus rare)
- Utiliser des composants préexistants (dérivés, composer, combiner/lier)

Certaines licences de composants peuvent être incompatibles avec la licence de diffusion propriétaire ou libre souhaitée

Il faut ainsi vérifier les compatibilités des licences des composants du Logiciel (Licensing-in) versus la licence choisie pour diffuser le Logiciel Libre (Licensing-out)



# Les modes de développement

Réutilisation de composants <i>(open source ou non)</i>	Oui Licensing in	CAS 2: risque moyen <i>(compatibilité des licences)</i>	CAS 4: risque élevé <i>(compatibilité des licences) (contrats d'indivision – Choix de la licence d'exploitation)</i>
	non	CAS 1: risque faible	CAS 3 : risque moyen <i>(contrats d'indivision – Choix de la licence d'exploitation)</i>
		Non <i>(pleine propriété)</i>	oui
Développement Collaboratif			

# Analyse juridique du logiciel

Le statut juridique est principalement déterminé à partir des éléments suivants:

- Identification des auteurs / ayant-droits
- Identification des briques logicielles de tiers
- Licence des composants tiers
- Identification des contrats ayant une incidence sur le logiciel (modes contractuels et de financements du développement des différents modules constitutifs du logiciel, licences des composants...)

# Corrections possibles

## Problèmes de comptabilité de licence

- Trouver un autre composant
- Demander aux ayants-droits un changement de licence
- Ne pas diffuser le composant gênant
- Réécriture

## Problème de (co)propriété

- Trouver un autre composant
- Signer un accord d'indivision ou obtenir une licence
- Acquérir les droits
- Ne pas diffuser le composant gênant
- Réécriture