

# **Licences Libres et Open Source**

Ce modèle juridique original utilise le droit d'auteur pour garantir les libertés d'utilisation du logiciel.



# Contexte et Importance



## Infrastructure Numérique

Ces licences sont le fondement de l'infrastructure numérique mondiale, avec 70% des serveurs web utilisant Linux et Android équipant 70% des smartphones.



## Protection et Liberté

Comment concilient-elles la protection du droit d'auteur et la liberté d'utilisation ?

## DÉFINITIONS

# Logiciel en Droit d'Auteur et Licences

## Logiciel et Droit d'Auteur

- Protection automatique dès la création (70 ans après la mort de l'auteur).
- Droits patrimoniaux (reproduction, adaptation) et moraux (paternité).

## Qu'est-ce qu'une Licence ?

- Contrat autorisant l'usage du logiciel sous certaines conditions.
- Autorisation d'usage, non transfert de propriété.
- Acceptation par l'usage



# Logiciel Libre vs. Open Source



## Logiciel Libre (FSF)

Basé sur 4 libertés fondamentales : exécuter, étudier/modifier, redistribuer, distribuer les modifications. "Free" signifie liberté.



## Open Source (OSI)

Défini par 10 critères, incluant la libre redistribution, l'accès au code source et l'autorisation des travaux dérivés.



## Différence Philosophique

Le Libre est éthique (libertés), l'Open Source est pragmatique (avantages techniques). Les licences sont souvent les mêmes.

# Fondement Juridique et Copyleft

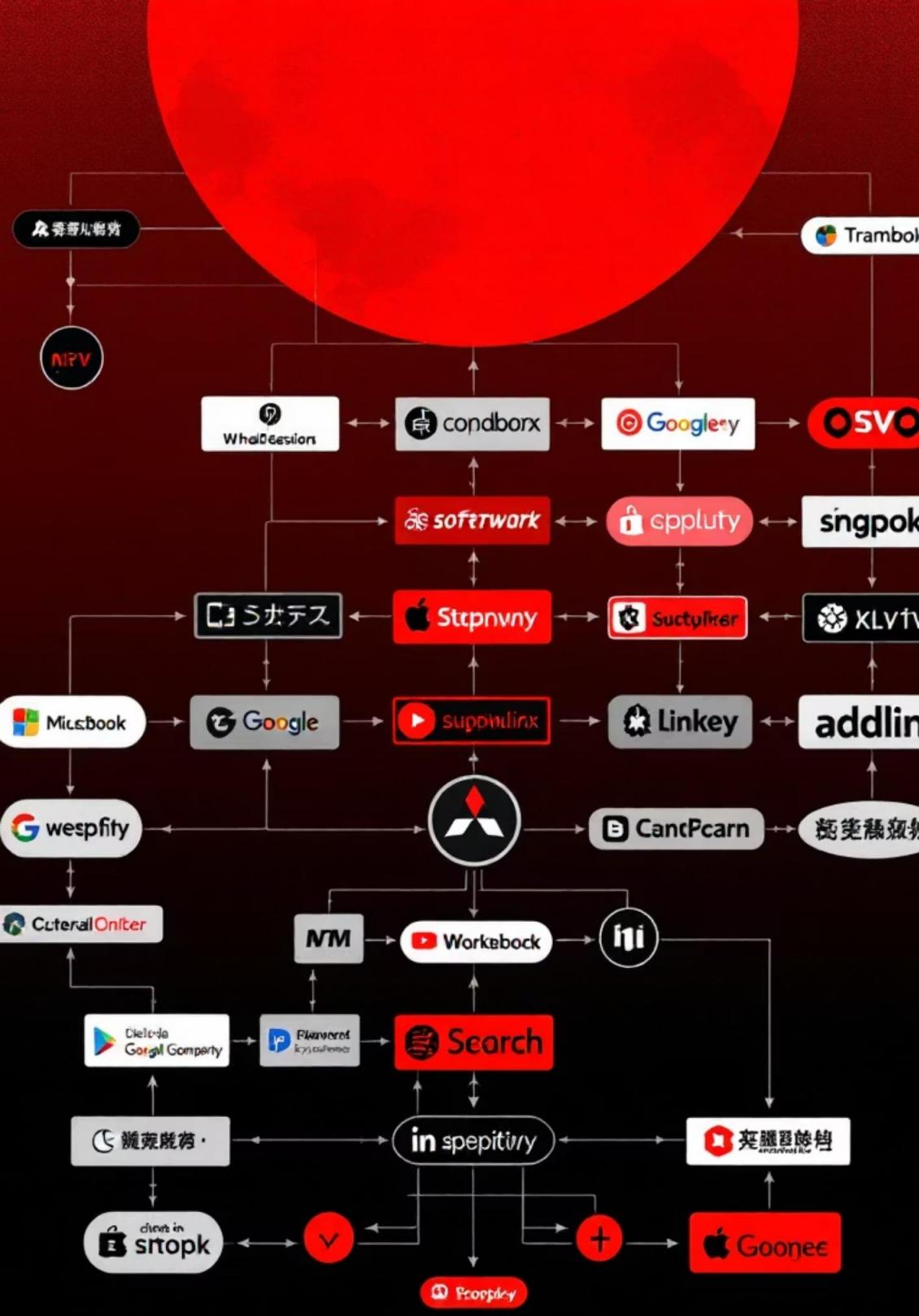


## Validité Juridique

Les licences libres utilisent le droit d'auteur pour garantir des libertés. La violation entraîne la perte des droits. Validité reconnue par la jurisprudence internationale.

## Mécanisme du Copyleft

Oblige à redistribuer les versions modifiées sous la même licence. Le copyleft fort (GPL) protège le bien commun, le copyleft faible (LGPL) permet la liaison avec du code propriétaire.



## PRINCIPES FONDAMENTAUX

# Obligation de Code Source et Compatibilité



## Code Source Obligatoire

Les libertés 1 et 3 exigent l'accès au code source lisible, aux scripts de compilation et à la documentation technique.



## Compatibilité des Licences

La combinaison de code sous différentes licences est complexe. Les licences permissives sont généralement compatibles avec presque tout, la GPL est plus restrictive.

# Famille GNU (Copyleft)



## GNU GPL

Copyleft fort, code source obligatoire, modifications sous GPL. Protège le bien commun mais peut être complexe.



## GNU LGPL

Permet la liaison avec du code propriétaire, utilisée pour les bibliothèques logicielles. Compromis copyleft/permissif.



## GNU AGPL

Couvre le SaaS, oblige la transparence des services web. Controversée pour sa restrictivité.

# Licences Permissives et Hybrides

## MIT License

Très courte, simple et permissive, sans copyleft.  
Obligations minimales (inclure texte de licence,  
préserver copyright).

## Apache License 2.0

Clause de brevets explicite, protection contre les  
litiges. Compatible GPL v3, adoptée en entreprise.



# Impacts Juridiques et Économiques

## Avantages Juridiques

Sécurité juridique, protection des droits, clarté contractuelle.

## Avantages Économiques

Réduction des coûts, modèles économiques viables (support, double licence), innovation accélérée.

## Limites Juridiques

Complexité, incompatibilités, conformité difficile, zones grises (SaaS, brevets).

## Limites Économiques

Monétisation complexe, problème du "free-riding", sous-financement des mainteneurs.

# Bilan et Perspectives

## Succès Indéniable

Fondement de l'infrastructure numérique mondiale, modèles économiques viables, innovation collaborative.

## Défis Persistants

Financement des mainteneurs, équilibre liberté/viabilité économique, adaptation aux nouvelles technologies (cloud, IA).



Le logiciel libre est une question de liberté, pas de prix. Il démontre qu'un modèle collaboratif peut être techniquement supérieur et économiquement viable.