



CHECKLIST DE DÉCISION ET CONFORMITÉ

GPL ou AGPL : la checklist pour choisir sans oublier l'essentiel

Utilisez cette fiche avant de publier, distribuer ou déployer votre logiciel. Elle aide à relier votre choix de licence au mode réel de mise à disposition et aux obligations de code source.

Au sommaire

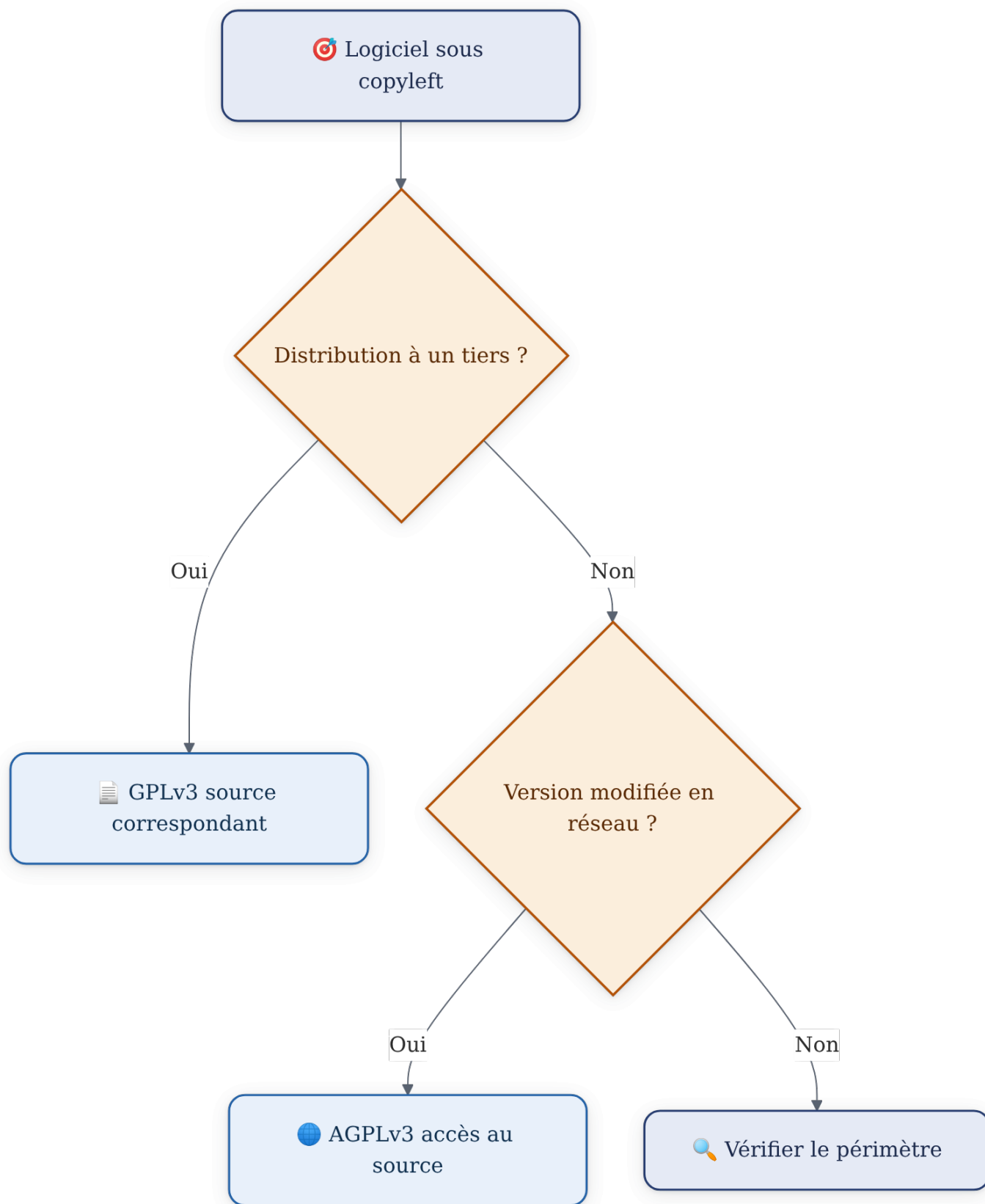
- 1 Le repère décisif
- 2 GPLv3 ou AGPLv3 : repères pratiques
- 3 1. Cartographier le projet avant de choisir
- 4 2. Choisir selon le scénario de déploiement
- 5 3. Préparer la conformité avant la mise à disposition

Le repère décisif

La GPLv3 encadre principalement la transmission du logiciel à un tiers.

Le bon choix dépend de la façon dont le logiciel est assemblé, modifié et mis à disposition : application distribuée, logiciel embarqué, bibliothèque, service web ou API.

L'AGPLv3 reprend ce cadre et ajoute une obligation spécifique lorsqu'une version modifiée est utilisée par des personnes qui interagissent avec elle à distance via un réseau.



INFOGRAPHIE Schéma GPL ou AGPL pour comparer distribution logicielle et accès réseau

GPLv3 ou AGPLv3 : repères pratiques

Situation à examiner	GPLv3	AGPLv3
Copyleft	Oui	Oui
Logiciel modifié remis à un client	Obligations de licence et de code source correspondant applicables	Obligations de licence et de code source correspondant applicables
Version modifiée exploitée comme service en ligne	Pas de clause réseau équivalente	Vérifier l'obligation issue de la section 13
Objectif principal	Préserver les libertés lors de la distribution	Étendre cette logique aux utilisateurs d'une version modifiée accessible par réseau

1. Cartographier le projet avant de choisir



Identifier chaque composant intégré au produit.

Inclure le code développé en interne, les dépendances, bibliothèques et composants tiers.



Relever la licence et la version exacte de chaque composant.

Ne pas assimiler automatiquement les règles de versions différentes d'une même licence.



Décrire les modifications apportées aux composants sous GPL ou AGPL.

Distinguer le composant modifié du code qui l'entoure.



Documenter le mode de mise à disposition réel.

Téléchargement, remise au client, appareil embarqué, hébergement SaaS, interface web ou API.



Identifier les utilisateurs qui accèdent réellement au programme.

Particulièrement important pour une version modifiée sous AGPLv3 accessible à distance.

2. Choisir selon le scénario de déploiement



Vous remettez une application ou un logiciel embarqué à un client : analyser le scénario comme une distribution.

La GPLv3 peut correspondre si le projet accepte le copyleft sur les versions distribuées.



Vous hébergez un service web, une plateforme SaaS ou une application métier : vérifier si une version AGPLv3 modifiée interagit avec des utilisateurs via le réseau.

La clause réseau de l'AGPLv3 appelle une analyse dédiée.



Vous publiez une bibliothèque destinée à être utilisée par des logiciels variés : examiner aussi l'alternative LGPL.

Elle peut permettre plus largement l'utilisation par des applications sous d'autres licences, sous ses propres conditions.



Vous prévoyez un modèle commercial : vérifier sa compatibilité avec les obligations de redistribution.

La vente, le support, l'hébergement, l'intégration, la formation et les garanties contractuelles restent possibles.

ÉTAPES DE CONFORMITÉ GPL OU AGPL

Un processus structuré pour respecter vos obligations de licence

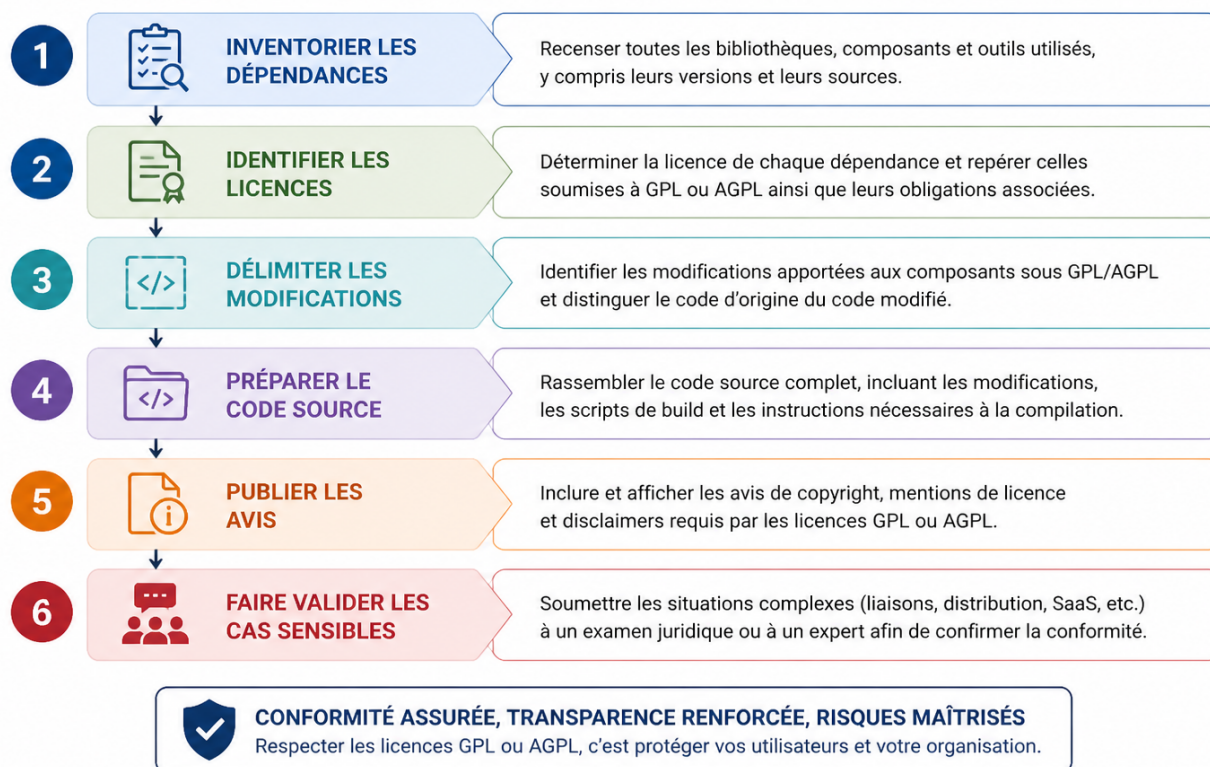


SCHÉMA Étapes de vérification avant la distribution ou le déploiement d'un logiciel sous GPL ou AGPL.

3. Préparer la conformité avant la mise à disposition

- ☐ **Conserver les avis de droit d'auteur et les textes de licence applicables.**
- ☐ **Préparer le code source correspondant pour les versions dont les obligations le requièrent.**
Le code source est la forme privilégiée pour modifier le programme.
- ☐ **Prévoir les modalités d'accès au code source adaptées au scénario de distribution.**
Pour une version modifiée sous AGPLv3 accessible à distance, l'accès doit être proposé de façon visible aux utilisateurs concernés.
- ☐ **Vérifier séparément les composants, leurs licences et leur intégration.**
Une communication entre services, y compris par API, ne permet pas à elle seule de conclure que tout le code environnant doit être publié.
- ☐ **Conserver une trace de l'inventaire des dépendances et des décisions prises.**

! Ne confondez pas code source et données

Le code source ne couvre pas automatiquement les données saisies par les utilisateurs, les contenus publiés, une base de données métier ou les résultats produits par le service. Le périmètre dépend des composants concernés et de leur intégration.

i Dernière vérification

Pour un produit commercial ou une architecture complexe, appuyez-vous sur les textes officiels des licences et sollicitez un avis juridique spécialisé avant la distribution ou le déploiement.

