

# Cours Logiciels Libres

## Économie du logiciel libre (suite)

Théo Zimmermann

Télécom Paris, Institut Polytechnique de Paris

Vendredi 7 avril 2023



# Participer à l'open source en tant qu'entreprise

## Les employés ne sont pas interchangeables

- Dans un projet open source, les participants sont des individus :
  - Les contributeurs construisent progressivement leur **réputation** basée sur leurs contributions.
  - Les droits / **rôles** dans le projet sont attribués aux individus.
  - Si un développeur **change d'entreprise**, il / elle conservera sa réputation et, le plus souvent, également ses droits.
  - Si tous les contributeurs en provenance d'une entreprise agissent toujours en bloc, les autres participants chercheront à réduire leur poids dans les décisions.
- Les nouveaux embauchés devront construire leur réputation à leur tour :
  - Il vaut mieux éviter de leurs donner des droits trop vite s'ils n'ont pas encore fait leurs preuves.
  - Si des droits leurs sont accordés, on attendra d'eux qu'ils soient au niveau (pas de droit à l'erreur).

# Embaucher des développeurs open source

- Les contributeurs à un projet forment une base dans laquelle il peut être intéressant d'embaucher des développeurs car **leurs compétences ont déjà été démontrées**.
- Lorsque les entreprises embauchent des membres importants du projet (voire son fondateur / dictateur), elles doivent avoir conscience que celui ou celle-ci devra éviter d'**abîmer sa réputation** en prenant des décisions favorables à l'entreprise mais défavorables au projet :
  - Il y a donc une double allégeance à l'entreprise et au projet.
  - L'entreprise ne peut pas tout exiger d'un employé.

## Être transparent sur ses motivations

- Une entreprise ne peut **pas décider toute seule** de la direction à donner à un projet sauf à risquer de s'aliéner la communauté de contributeurs (comme pour un dictateur bienveillant).
- Donc, il **ne suffit pas de payer** le développement d'une fonctionnalité pour garantir qu'elle sera dans la prochaine version du logiciel.
- Pour **convaincre**, il vaut mieux être transparent (dans la limite de ce qui est légalement ou commercialement possible).

## Adapter son organisation et ses processus

- Lorsqu'une entreprise n'a pas la contribution au logiciel libre au cœur de ses activités, l'organisation peut être **mal adaptée** en termes d'**attentes** à avoir vis-à-vis des développeurs qui sont impliqués.
- L'entreprise doit accepter de ne **pas avoir le contrôle absolu** sur le projet. Par conséquent, elle ne peut pas blâmer ses développeurs pour des **retards** qui ne seraient pas de leur faute.
- Il faut également qu'elle évite de baser sa stratégie ou ses **plannings** sur des événements sur lesquels elle n'aurait pas le contrôle (ex., sorties de nouvelles versions).

# Le rôle des fondations

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

- Une fondation est une organisation avec une **existence légale** sous divers statuts possibles (qui dépendent du pays dans laquelle la fondation est créée).
- La principale différence avec une entreprise est qu'une fondation est en général à **but non lucratif**.
- Une fondation ne peut pas parfaitement représenter la **communauté** d'un projet open source (qui est **informelle**) mais elle est en général gérée par des gens reconnus dans la communauté.
- Parfois, le statut de **membre de la fondation** est accordé à des contributeurs reconnus (comme dans la Python Software Foundation ou la Fondation Apache).

# Pourquoi a-t-on besoin de fondations ?

- Une fondation a une **personnalité morale**, elle peut donc :
  - être dépositaire de marques (nom, logo, etc.),
  - recevoir et dépenser des fonds au nom du projet,
  - posséder un nom de domaine,
  - signer des contrats,
  - organiser des évènements.

(au lieu que cela soit fait par des individus ou des entreprises).

- Elle peut servir à établir une **gouvernance indépendante** de l'entreprise qui contrôlait un projet open source :
  - Martin Michlmayr (dans "FOSS Foundations", 2021) donne l'exemple de la transmission du projet **Kubernetes** par Google à la Cloud Native Computing Foundation.
  - Un autre exemple est la transmission par Oracle d'OpenOffice à la Fondation Apache.

# Activités des fondations

- **Développement** du projet :
  - Embauche de développeurs.
  - Financement par des bourses / contrats de prestation.
  - Documentation, autres ressources, etc.
- **Organisation** de la communauté :
  - Évènements, financement pour les voyages.
  - Mentorat.
  - Infrastructure.
  - Responsabilité de modération des forums (Code de Conduite).
- **Collecte** de fonds :
  - Dons des particuliers et des entreprises.
  - Adhésion des entreprises.
  - Bourses de fondations ou d'organismes publics.
- Gestion des **droits d'auteur** (cas de la Free Software Foundation et de Software Freedom Conservancy).

## Créer une fondation

- C'est **compliqué** : il faut choisir la juridiction (pays d'accueil), le statut, faire les démarches administratives, rédiger des statuts, recruter un bureau, etc.
- La recommandation est : ne créez pas de fondation, **rejoignez une fondation existante** qui accueille de nouveaux projets open source.

Exemples :

- Free Software Foundation,
- Software Freedom Conservancy,
- Software in the Public Interest,
- Open Source Collective,
- Eclipse Foundation,
- Apache Foundation,
- Linux Foundation.

## Le modèle Fondation dans la Fondation

Certaines fondations acceptent d'accueillir en leur sein d'autres fondations, avec leur système de gouvernance indépendant, mais en leur épargnant certains aspects juridiques / légaux / administratifs.

Par exemple :

- La **Linux Foundation** (modèle “Foundation as a Service”) :
  - Cloud Native Computing Foundation,
  - Hyperledger Foundation,
  - OpenJS Foundation,
  - LF Energy, LF Public Health, LF Networking, LF Edge, etc.
- **Software in the Public Interest** accueille en son sein :
  - X.Org Foundation,
  - Open Bioinformatics Foundation,
  - Open Voting Foundation.
- La **Fondation Inria** accueille en son sein :
  - la Fondation Software Heritage,
  - la Fondation OCaml.

## Le modèle Mozilla : entreprise + fondation

- Mozilla Foundation : une fondation avec pour mission d'organiser la communauté.
- Mozilla Corporation : une entreprise qui **appartient à la fondation**.
  - **Emploie** des développeurs pour travailler sur Firefox (et les autres produits Mozilla).
  - Tire des revenus d'**activités commerciales**.
  - Une part très importante de ses revenus provient de Google pour que Google soit le **moteur de recherche par défaut** dans Firefox.

# Recevoir des dons

# Comment recevoir des dons

Il existe de nombreuses **plateformes** pour recevoir des dons :

- Des plateformes très **génériques**, comme Paypal.
- Des plateformes spécialisées pour les **créateurs de contenu** (individus), comme Patreon (commercial), Liberapay (association basée en France).
- Des plateformes spécialisées pour les **projets open source** comme Open Collective et GitHub Sponsors.

# Statut pour recevoir des dons

Les mainteneurs peuvent recevoir des dons :

- en tant qu'individus (ce sont des revenus, qu'il faut **déclarer fiscalement**, Patreon collecte même la TVA),
- en tant que collectifs (associations, fondations, etc.).

Par type de plateforme :

- Patreon et Liberapay sont conçus pour des individus.
- Open Collective est conçu pour des collectifs.
- GitHub Sponsors supporte les deux.

Pour recevoir des dons en tant que collectif, il faut un **statut légal** (association, fondation). Être hébergé dans une fondation est une solution. Open Collective en propose plusieurs.

# Transparence dans l'utilisation des fonds

## Budget ⌵

Transparent and open finances.

All Expenses Transactions [View all expenses →](#)

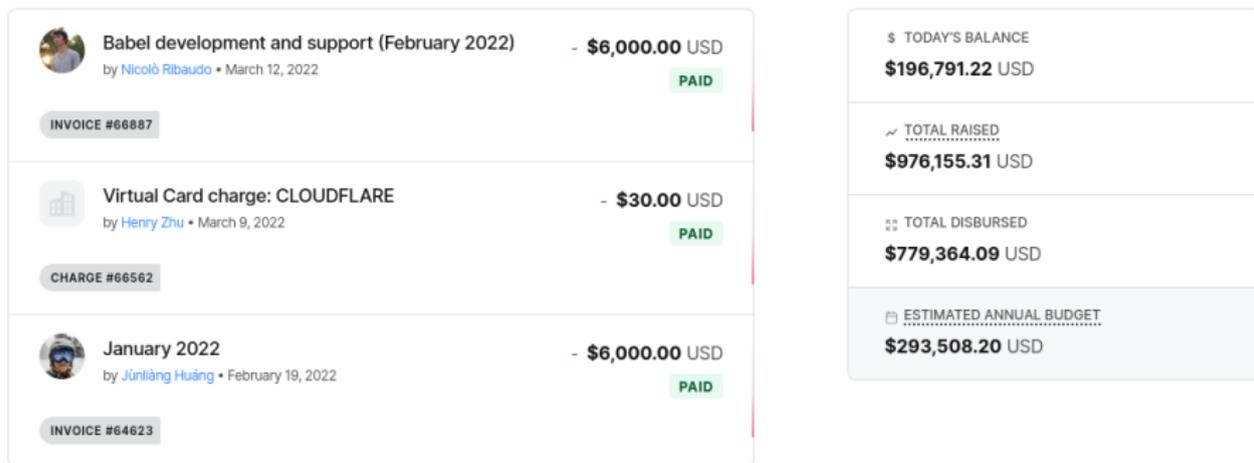


Figure 1: Open Collective promeut un modèle de transparence sur les fonds collectés et dépensés.

## Est-il facile de se financer par les dons ?

D'après Overney et al. (2020) :

- Beaucoup de projets demandent des dons pour supporter les activités de développement.
- Mais **rare sont ceux qui reçoivent suffisamment** pour financer un développeur à plein temps. Par exemple sur npm :
  - Seuls 2 paquets sur 1000 réclament des dons.
  - Seuls 5% de ceux qui en réclament reçoivent  $> 1000\$/\text{mois}$ .
- Beaucoup de projets (sur Open Collective) ne dépensent pas l'argent collecté (tésaurisation) ou le dépensent pour d'autres utilisations (exemple : frais d'hébergement).

## Campagnes de financement participatif

- Les solutions mentionnées jusque là sont faites pour obtenir un **financement récurrent** : cela apporte de la **stabilité**, ce qui est important pour la maintenance de logiciel libre.
- Les campagnes ponctuelles de **financement participatif** sont un autre moyen d'obtenir des dons :
  - Cela peut être un moyen efficace de **financer** le développement d'une fonctionnalité ou d'une **amélioration spécifique** car beaucoup de gens peuvent se mobiliser pour atteindre le but affiché de la campagne et en voyant les autres donner.
  - En revanche, il est peu crédible (et très insécurisant) de financer le **salaire** des principaux mainteneurs par des campagnes répétées de financement participatif (et la mobilisation risque de **perdre de l'ampleur** progressivement).