

CST

COMMISSION SUPÉRIEURE TECHNIQUE DE L'IMAGE ET DU SON

La Lettre



OPEN SOURCE

- L'OPEN SOURCE : UN CHOIX DE PROJET
- L'APPROCHE OPEN SOURCE POUR L'IA
- POURQUOI LA CST FAIT DE L'OPEN SOURCE
- FRANCE TÉLÉVISIONS : UN CHOIX STRATÉGIQUE ET TECHNIQUE
- LE CNC ET L'OPEN SOURCE : ENTRETIEN AVEC ARNAUD ROLAND

EN PARTENARIAT AVEC

Ficam
CINÉMA AUDIOVISUEL MULTIMÉDIA

LA CST est la première association de techniciens du cinéma et de l'audiovisuel française.

Née en 1944, Elle promeut l'excellence technique qui permet l'aboutissement de la vision de l'équipe artistique et garantit que cette vision est correctement restituée sur l'écran pour l'ensemble des spectateurs. La CST organise les groupes de travail d'où émergeront les bonnes pratiques professionnelles qui deviendront des recommandations techniques, voire même des normes ou standards. La CST accompagne également les salles de cinéma qui souhaitent proposer une expérience optimale à leurs spectateurs. À ce titre, elle assure la Direction technique de plusieurs festivals, dont le Festival International du Film de Cannes.

Enfin, la CST est la maison des associations de cinéma avec aujourd'hui 25 associations membres.

La CST, forte de plus de 700 membres, est principalement financée par le CNC.

Cette Lettre est également la votre !

Vos contributions sont les bienvenues à l'adresse redaction@cst.fr.

SOMMAIRE

▶ L'open source : un positionnement par rapport à l'Innovation	4
▶ L'open source comme moteur de standardisation	8
▶ L'open source : un choix de projet	12
▶ L'approche « open source » pour l'IA	16
▶ Le CNC et l'open source	20
▶ Pourquoi la CST fait-elle de l'open source ?	22
▶ Un choix stratégique et technique	24
▶ L'open source dans les ICC : une démarche innovante et pragmatique	26
▶ VLC et l'open source ont bouleversé l'industrie vidéo : où va la valeur ?	28
▶ Kitsu, une plateforme au service des créateurs	30
▶ L'innovation en mouvement perpétuel	34
▶ GPAC, une boîte à outils multimédia qui séduit les acteurs du streaming	37
▶ Stax, un outil de review open source basé sur Blender	38
▶ Pourquoi choisir l'open source	40
▶ L'animation pour tous (Duik)	42
▶ Ouvrir les données pour libérer la distribution	44
▶ Dans la cour des grands (TSDuck)	46

CST : 9 rue Baudoin 75013 Paris • Tél. : 01 53 04 44 00 • Fax : 01 53 04 44 10 • Email : redaction@cst.fr • Internet : www.cst.fr
Directeur de la publication : Baptiste Heynemann • Rédacteur en chef : Ilan Ferry.

Remerciements aux contributeurs : Marc Bourhis, Frédéric Fermon, Hans-Nikolas Locher, Benoît Maujean.

Couverture : © CST / Ficom

Maquette : fabiennebis.wixsite.com/graphisme

Relecture : Christian Bisanti

Impression : numeric@corlet.fr

Dépôt légal novembre 2025



L'OPEN SOURCE : UN IMPÉRATIF STRATÉGIQUE AU CŒUR DE LA TRANSFORMATION DE NOTRE INDUSTRIE

Nous fêtons les 80 ans de l'association en septembre 2024 et nous nous demandons comment allaient se dérouler les 80 prochaines années pour les technicien·nes du cinéma et de l'audiovisuel. Bien qu'il soit évidemment impossible de prédire l'avenir dans une boule de cristal, on voit cependant poindre quelques risques et quelques opportunités. Dans un contexte où l'intelligence artificielle bouleverse les chaînes de fabrication et les modèles économiques, se pose la question de la qualité des œuvres face à une offre de plus en plus standardisée qui risque de saturer les spectateurs. Les modèles industriels traditionnels (économies d'échelle, reproductibilité, chaînes de valeur verticales liant un producteur, un distributeur, un exploitant et des services de télévision) sont en passe d'être supplantés par un paradigme dominé par la plateforme, les effets de réseau, la découvrabilité, la réinvention permanente... et la domination des acteurs technologiques de la Silicon Valley.

Notre dépendance technologique à des acteurs mondiaux pratiquant des modèles confiscatoires devient alors un enjeu stratégique primordial. C'est ici que l'open source entre en scène et mérite l'attention des décideurs de nos industries. Il porte en lui les ingrédients d'une réponse structurante aux enjeux contemporains : diversité des attentes clients, intégration de standards émergents, interopérabilité des outils, attractivité des talents.

Sur le plan financier, le modèle est tout aussi disruptif. L'open source permet de rediriger les budgets historiquement engloutis dans les licences propriétaires vers des investissements à plus forte valeur ajoutée : formation, intégration sur mesure, contribution aux communautés, montée en

compétence interne, partage de certaines ressources de développement. Autant de leviers qui renforcent l'autonomie stratégique des acteurs.

L'open source devient une réponse stratégique et un modèle de résilience collective.

Au fil de la série d'articles qui constitue ce hors-série, nous donnons la parole à des experts et recueillons les témoignages d'acteurs ayant expérimenté cette démarche collaborative, pour mieux comprendre et analyser en quoi cette adoption stratégique de l'open source pourrait s'imposer désormais comme une condition sine qua

non pour garantir l'indépendance, la pérennité et l'excellence technique de notre industrie.

Lorsqu'un projet open source atteint la masse critique, il devient un standard de facto, créant ainsi une relation de confiance durable entre développeurs et utilisateurs, fondée sur la transparence et la gouvernance partagée. Cette logique trouve déjà une incarnation dans notre action à la CST. Dossier après dossier, notre communauté défend une vision partagée des pratiques professionnelles pour garantir la qualité de la fabrication des œuvres. Les recommandations techniques que nous publions constituent déjà des biens communs, co-construits et entretenus collectivement.

L'enjeu n'est plus seulement technique : il est politique, économique et culturel et nous conduit vers une souveraineté technologique de notre création.



© Lionel de Sousa



© DR

Baptiste HEYNEMANN
Délégué général de la CST

Benoît MAUJEAN
Délégué Adjoint Technologies de la FICAM
Président de ALICEVISION

L'OPEN SOURCE : UN POSITIONNEMENT PAR RAPPORT À L'INNOVATION

Le développement logiciel peut sembler éloigné du quotidien des artistes et techniciens du cinéma, habitués à un univers façonné par des éléments concrets : le grain de la pellicule 35 mm, le poids d'une caméra Arriflex, la chaleur des projecteurs Fresnel, la texture des décors en bois et toile peinte. Pourtant, la transformation numérique et l'accélération des innovations ont poussé les studios de tournage et de postproduction à acquérir des compétences internes en développement logiciel, faute de solutions externes systématiquement adaptées à leurs besoins. Au fond, tous ces métiers relèvent d'une même logique de construction collective.

Cette dynamique d'innovation s'impose pour plusieurs raisons et sous diverses contraintes :

1. Un décalage fréquent entre l'ambition artistique et les solutions disponibles : les productions poursuivent souvent des objectifs que le marché ne propose pas encore, stimulées par des exigences créatrices qui poussent continuellement les limites techniques pour explorer d'autres horizons,
2. Le manque de solutions réellement sur mesure, les éditeurs n'intégrant pas toujours des fonctionnalités adaptées à des workflows très spécialisés,
3. Le décalage entre des démonstrations virales (algorithmes ou applications grand public testés sur des contenus basse résolution) et les exigences professionnelles : images grand format, avec une grande dynamique, une faible profondeur de champ, etc.

Internaliser le développement logiciel ne rime pas forcément avec des projets démesurés. Sur un long-métrage en images de synthèse, écrire ses

propres shaders – les scripts qui modifient l'apparence d'une surface ou le rendu, ou ses outils d'animation de personnage – permet d'affiner l'esthétique finale, comme on peaufine un étalonnage. À une échelle plus modeste, même le site web d'un studio résulte de lignes de code, développées en interne ou externalisées. Qu'il soit visible ou non, le code est désormais un maillon à part entière de la chaîne de fabrication d'un film.

ENTRER DANS LE CODE

Derrière son apparence ésotérique, le code n'est qu'un ensemble d'instructions organisées pour créer un programme. Son élaboration repose sur deux étapes clés :

- la rédaction : le développeur écrit des lignes de code, avec un langage structuré, comparable à une partition ou à un scénario technique. Cette phase inclut aussi la rédaction d'une documentation claire, essentielle tant pour les futurs utilisateurs (guides, tutoriels) que pour les potentiels autres développeurs (commentaires dans le code, schémas d'architecture),
- la compilation : ces instructions sont interprétées et assemblées pour produire un logiciel exécutable, comme un négatif transformé en film projetable.

En pratique, ce processus s'appuie rarement sur une création ex nihilo : il intègre souvent des bibliothèques ou des frameworks existants, à la manière d'un cinéaste utilisant des ressources partagées ou des recettes déjà connues.

Ainsi, le code d'un programme informatique est un ensemble d'instructions, écrit dans un langage spécifique, qui permet à un logiciel de s'exécuter sur une machine donnée. Il existe de nombreux langages (C++, Python, JavaScript...), chacun conçu pour des contraintes techniques différentes, d'où l'importance du choix du langage en amont. Beaucoup de logiciels assemblent plusieurs composants écrits dans des langages différents. Le code source reste un texte lisible par un simple éditeur de texte, ce qui permet d'en comprendre le fonctionnement et, au besoin, de le modifier.

La compilation convertit ensuite la suite d'instructions en un binaire exécutable par le système d'exploitation : .exe et .dll sous Windows, binaires ELF sous Linux, .app sur macOS, ainsi que divers paquets pour smartphones. Une fois compilé, le lo-



giciel devient difficile, voire impossible, à analyser précisément ou à modifier. À l'inverse, certains langages sont interprétés : le code est lu et exécuté directement sans compilation préalable, ce qui offre une grande souplesse pour les scripts, au prix de performances généralement moindres.

Le paysage logiciel se structure autour de deux paradigmes : les solutions propriétaires d'un côté, dont le code source n'est ni modifiable, ni redistribuable et de l'autre les projets open source, dont le code est publiquement accessible, et qui encouragent au contraire audit, contribution et réutilisation. Bien que souvent méconnus, les logiciels open source sont présents partout : des systèmes embarqués (smartphones et téléviseurs) aux infrastructures cloud (serveurs et bases de données), en passant par la quasi-totalité des pipelines créatifs des studios VFX et d'animation. De manière globale, une étude de 2023¹ chiffre son usage à plus de 80 % des organisations françaises, preuve de son rôle clé dans notre quotidien numérique.

L'open source est aujourd'hui au cœur des débats technologiques, notamment dans le domaine de l'intelligence artificielle (IA). Le choix de l'open source et les raisons de développer ou d'adopter des logiciels ouverts ne relèvent pas seulement de considérations techniques ou économiques ; elles traduisent aussi un positionnement stratégique, voire éthique et politique, vis-à-vis de l'innovation.

Un logiciel open source est un logiciel dont la licence répond aux critères de l'Open Source Initiative² définissant les libertés d'utiliser, de copier, d'étudier et de modifier les logiciels ainsi que de redistribuer les versions modifiées. L'accès au code source, facilité par des plateformes de centralisation comme GitHub ou Gitlab, est une condition nécessaire pour exercer les deux dernières libertés (la modification et la redistribution). L'open source, fondé sur des développements collaboratifs et incrémentaux – le code d'un projet open source tire bien souvent parti de bibliothèques logicielles issues d'autres projets open source – permet à une communauté de partager et d'enrichir collectivement les ressources logicielles.

Traditionnellement, les entreprises choisissent entre des solutions commerciales fermées – confiées à des experts – donc souvent bien accompagnées, mais dont les fonctionnalités et l'évolution restent peu maîtrisables, et des développements internes, qui exigent des ressources durables pour concevoir, déployer et maintenir l'outil. L'open source offre une troisième voie en mutualisant les efforts de développement avec d'autres acteurs, industriels ou académiques, pour répondre à des problématiques

communes. A noter également que presque tous les outils commerciaux utilisent des bibliothèques logicielles open source.

L'approche open source présente dans sa démarche trois avantages majeurs : la pérennité, la qualité et la transparence des logiciels développés. Le partage de code implique un minimum de clarté et de documentation, ce qui tire vers le haut la qualité du code développé. Cela permet d'obtenir un code plus facile à maintenir et à comprendre pour les nouveaux contributeurs. L'open source offre aux développeurs une visibilité accrue et une expertise élargie, enrichie par les contributions diversifiées d'experts issus du monde industriel et académique. Ce développement ouvert et collaboratif contribue aussi à la pérennité du logiciel, les contributeurs (développeurs ou utilisateurs) continuant parfois leur collaboration indépendamment de leur parcours professionnel. A contrario, la pérennité des solutions fermées peut être remise en cause lors du départ de quelques développeurs clés ou le changement stratégique voire la disparition des sociétés porteuses du logiciel. La transparence du code et des algorithmes est un autre avantage clé, car elle garantit une visibilité sur les processus utilisés. Contrairement aux solutions commerciales fermées, qui peuvent produire des résultats inattendus ou des blocages dus à des bugs non résolus, l'open source facilite le débogage du code source, ainsi que l'interaction entre les utilisateurs et les développeurs. Cette transparence est souvent un gage de confiance pour les utilisateurs, qui peuvent ainsi avoir une meilleure appréhension et une meilleure contribution aux outils qu'ils utilisent.

DE LA GENÈSE AUX RÈGLES DU JEU : PANORAMA DES LICENCES OPEN SOURCE

En 1984, Richard Stallman, un informaticien du Massachusetts Institute of Technology (MIT), lance le mouvement du logiciel libre en créant la Free Software Foundation (FSF) et en élaborant la licence GNU General Public License (GPL), qui définit les principes fondamentaux régissant l'utilisation, la modification et la redistribution des logiciels libres. Il s'agit ici de voir le code ouvert comme un bien commun. En 1991, Linus Torvalds, un informaticien finlandais, pose les bases de Linux, un système d'exploitation open source qui permet à des milliers d'utilisateurs de contribuer au projet et aux futures variantes de ce noyau logiciel. Pour mieux gérer l'évolution du code source de Linux, Linus Torvalds a également créé en 2005 Git, un système de contrôle des versions de codes logiciels qui est devenu la norme dans le domaine logiciel.

1. <https://numeum.fr/en/etudes/study-open-source-usage-in-france-open-source-monitor-2023>

2. <https://opensource.org/>

En 1998, Bruce Perens et Eric Raymond créent l'Open Source Initiative (OSI), organisation dédiée à la promotion des logiciels open source. L'OSI visait à remédier aux lacunes et à l'ambiguïté perçues du terme « logiciel libre » (free : libre ou gratuit), en déplaçant le centre d'intérêt du logiciel libre d'une question sociale et éthique vers un modèle économique qui intègre les services liés au développement logiciel, tout en faisant la promotion de l'accès aux codes sources.

Depuis, le périmètre des logiciels open source s'est considérablement élargi et se retrouve dans de nombreux outils dont l'utilisation est très largement répandue : VLC (lecteur de fichiers multimédias) ; LibreOffice (suite logicielle bureautique) ; 7-Zip (système de compression et d'archivage de fichiers) ; la plupart des navigateurs web, Webkit à la base de Safari et Mozilla Firefox ainsi que Chromium, à la base de Chrome, Brave, Vivaldi, Opera... ; Android, le système d'exploitation de près d'un smartphone sur deux ; les technologies derrière le web (qui reposent sur des standards ouverts définis par le World Wide Web Consortium), la plupart des outils de cryptographie, dont on peut examiner le code et vérifier s'ils présentent des vulnérabilités (les outils de la famille PGP et OpenSSL). Même une entreprise comme Microsoft, qui a toujours été connue pour ses critiques virulentes à l'encontre du logiciel libre, aujourd'hui développe, publie et héberge des projets open source, avec des initiatives telles que le développement de VS Code (éditeur de code extensible), l'acquisition de GitHub (hébergement, versioning et intégration continue gratuits pour les projets open source, même s'il est lui-même en code fermé, avec un positionnement commercial classique) et l'ouverture du code du nouveau terminal Windows (voir ici le portail des logiciels open source de Microsoft). Sans oublier bien sûr les outils logiciels open source dans le domaine des industries culturelles et créatives dont nous parlerons plus bas.

On peut aussi citer le mouvement open source hardware, autour des technologies de matériels

et produits physiques, construits selon les principes des ressources libres de droits ou sous licence libre. Les plus connus sont Arduino (cartes électroniques avec microcontrôleur) et RepRap (imprimantes 3D auto répliquantes). Et dans une moindre mesure Fairphone (téléphones portables modulables et réparables) et Raspberry Pi (mini PC à coût réduit) qui, même s'ils ne sont pas strictement des projets open hardware (matériel entièrement libre), incarnent des valeurs et des pratiques proches de l'écosystème open source (interopérabilité, résistance au verrouillage, documentation collaborative).

LES LICENCES OPEN SOURCE

Les logiciels libres répondent aux critères énoncés par la Free Software Foundation³, reposant sur quatre règles fondatrices définies dans la Free Software Definition :

- Liberté 0 : la liberté d'exécuter le programme, pour tous les usages ;
- Liberté 1 : la liberté d'étudier le fonctionnement du programme, et de l'adapter à ses besoins. Pour ceci l'accès au code source est une condition requise ;
- Liberté 2 : la liberté de redistribuer des copies, donc d'aider son voisin ;
- Liberté 3 : la liberté d'améliorer le programme et de publier des améliorations, pour en faire profiter toute la communauté. Pour ceci l'accès au code source est une condition requise.

Les logiciels open source répondent aux critères énoncés par l'Open Source Initiative, reposant sur les dix règles de l'Open Source Definition (elle-même dérivée de la Debian Free Software Guidelines) :

1. La libre redistribution du logiciel, permettant la distribution libre ou la commercialisation du logiciel, y compris dans des ensembles logiciels, sans exiger de droits d'auteur ou de commission sur la vente ;
2. La disponibilité du code source, notamment pour qu'un développeur puisse modifier le logiciel ;
3. La licence doit autoriser les modifications du code source et le développement de travaux dérivés, et elle doit permettre leur distribution dans les mêmes conditions que la licence du logiciel original (mais elle n'impose pas a priori une licence identique ou différente) ;
4. L'intégrité du code source de l'auteur : la licence doit autoriser la redistribution des versions dérivées, mais elle peut imposer des conditions sur le code source de ces modifications, comme la distribution du code des modifications ou le nommage distinct des travaux dérivés ;

3. <https://www.fsf.org/>



5. La non-discrimination contre les utilisateurs (personnes ou groupes) ;
6. La non-discrimination contre des champs d'application (y compris les usages commerciaux) ;
7. La distribution de licence : la licence s'applique par défaut à tous ceux à qui le programme est redistribué ;
8. La licence ne doit pas être spécifique à un produit : la licence du programme doit garantir les mêmes droits, qu'il soit distribué isolément ou au sein d'un ensemble logiciel ;
9. La licence ne doit pas restreindre d'autres logiciels : la licence ne doit pas imposer de conditions sur les autres logiciels distribués sur le même support que le logiciel d'origine (comme l'obligation d'être libres) ;
10. La licence doit être neutre sur le plan technologique : aucune disposition de la licence ne peut être fondée sur une technologie ou un style d'interface particulier.

Malgré les différences philosophiques entre le mouvement du logiciel libre et le mouvement du logiciel open source, la définition du logiciel libre de la FSF et celle du logiciel open source par l'OSI décrivent des ensembles de licences qui se chevauchent largement, avec seulement quelques différences mineures. L'appellation Free Open Source Software (FOSS) tente de réconcilier les deux mondes, elle est souvent admise comme le terme générique qui s'oppose aux logiciels propriétaires. Richard Stallman préfère ajouter le mot français « Libre »⁴ pour proposer « Free/Libre and Open Source Software » (FLOSS), pour éviter toute confusion entre « libre » et « gratuit » en anglais.

PROPAGATION DES CLAUSES DE LA LICENCE

La philosophie des logiciels libres a introduit la notion de copyleft (aussi appelée all rights reversed), qui garantit que les mêmes droits soient préservés dans les œuvres dérivées. L'objectif étant que chacun puisse bénéficier librement du travail antérieur d'autrui, mais que toute modification de ce travail doit également profiter à tous les autres, et doit donc être diffusée dans des conditions similaires. C'est l'une des caractéristiques essentielles de la licence GPL. On parle aussi de licence open source virale ou contaminante.

On distingue cependant la notion de « copyleft fort » lorsque les redistributions du logiciel ou de l'œuvre en question, modifiés ou non, ainsi que de tous les composants ajoutés, ne peuvent se faire que sous la licence initiale – c'est le cas de la licence GNU Lesser General Public License (GPL) ou même de la licence GNU Affero General Public License (AGPL).

4. <https://www.gnu.org/philosophy/floss-and-foss.fr.html>

Pour cette dernière, si un logiciel utilise un composant AGPL accessible à distance (par exemple via un serveur web), son code source, y compris les changements apportés, doit être publié sous la même licence.

On parle de « copyleft faible » lorsque les redistributions du logiciel ou de l'œuvre en question, modifiés ou non, se font sous la licence initiale, mais que de nouveaux composants peuvent être ajoutés sous d'autres licences, voire sous des licences propriétaires – c'est le cas de la licence GNU Lesser General Public License (LGPL) ou de la licence Mozilla Public License (MPL).

On peut résumer schématiquement ces différences dans le tableau suivant (pour les licences open source les plus utilisées) :

Licences permissives	Licences protectrices	Licences virales
BSD	MPL v2	GPL V2
Apache V2	LGPL V2	GPL V3
MIT	LGPL V3	AGPL

– **Les licences « permissives »** n'ont pas ou peu d'obligation sur la modification ou la redistribution des codes sources.

– **Les licences « protectrices »** imposent que les modifications soient redistribuées avec la même licence, mais sans être contaminantes.

– **Les licences « virales »** imposent que tout code dérivé ou intégré conserve la même licence, empêchant ainsi son utilisation dans des projets sous d'autres licences (effet de contamination).

D'autres clauses des licences open source peuvent concerner les obligations de référence à la licence et au droit d'auteur dans les versions modifiées ou dérivées, le positionnement par rapport aux éventuels brevets des contributeurs, etc.

Les potentiels conflits de licences entre bibliothèques logicielles open source ou leur intégration avec des licences fermées peuvent entraver leur utilisation combinée au sein d'un même projet et nécessitent une attention particulière. Cette complexité représente parfois un frein quant à l'adoption de ces technologies.

Pour un choix de licence plus abouti et une comparaison plus détaillée, voici plusieurs liens utiles :

1. <https://choosealicense.com/appendix/>
2. <https://opensource.guide/legal/>
3. https://en.wikipedia.org/wiki/Comparison_of_free_and_open-source_software_licenses
4. <https://www.tldrlegal.com/>
5. <https://www.gnu.org/licenses/license-list.html#GPLIncompatibleLicenses>

Voir ici le portail des logiciels open source de Microsoft : <https://opensource.microsoft.com/>

Benoît Maujean

L'OPEN SOURCE COMME MOTEUR DE STANDARDISATION

Le développement de logiciels open source offre une opportunité stratégique : établir un standard ouvert qui permet aux acteurs industriels de conserver la maîtrise des briques critiques d'interopérabilité entre leurs outils et processus clés. Une telle approche garantit leur autonomie tout en favorisant l'innovation collaborative.

Les standards et normes ont plusieurs objectifs, parmi lesquels :

- assurer la fluidité des échanges de contenus, à la création comme à la distribution,
- définir un certain niveau de qualité aux utilisateurs des contenus.

Les normes (standard de jure en anglais) sont édictées par des organismes reconnus comme l'AFNOR, le CEN, l'ETSI, l'ISO, l'ITU, etc. Les standards (standard de facto), sont émis par des organismes regroupant les acteurs d'un secteur, comme la SMPTE par exemple.

L'interopérabilité repose sur des spécifications communes permettant à des produits ou services d'acteurs différents de coopérer. Ces textes sont généralement écrits par les industriels les plus avancés, si bien que le standard reflète souvent la solution du leader du marché, même s'il est censé rester ouvert aux autres.

Dans notre filière, un exemple emblématique est la motivation récurrente des grands studios de production US de publier leur développement interne en open source afin que celui-ci soit adopté par une large communauté, évitant ainsi qu'un éditeur logiciel n'impose un standard industriel concurrent, avec un code propriétaire.

C'est le cas de la plupart des projets gérés par l'Academy Software Foundation (ASWF), fondée en août 2018 par l'Academy of Motion Picture Arts & Sciences (AMPAS) et qui regroupe maintenant la plupart des gros studios d'effets visuels et d'animation (Animal Logic, Blue Sky Studios, DNEG, Dreamworks Animation, Disney, Weta, Sony Pictures, Warner Bros, Netflix, Rodeo FX, Framestore, Raynault

VFX, Laika Studios, Skydance Animation), ainsi que de grands éditeurs logiciels (Autodesk, Epic Games, Foundry, Google Cloud, Intel, SideFX, NVidia, Ftrack, Red Hat, AWS, Apple, Microsoft, AMD, Adobe, HP, Otoy, Samsung, Stability AI).

Ainsi, EXR, format de fichiers pour le traitement et le stockage d'images en haute dynamique et multi-canaux, développé par Industrial Light & Magic en 1999, et publié en open source en 2003 sous le nom de OpenEXR, a-t-il été choisi comme le premier projet emblématique par l'ASWF, afin de garantir les ressources nécessaires pour continuer de maintenir ce format de fichier pivot devenu essentiel pour l'industrie de l'Animation et des VFX. Il en a été de même pour OpenColorIO, une solution de gestion de la couleur utilisable sur toute la chaîne de fabrication développée et publiée par Sony Pictures Imageworks en 2003, ou OpenVDB, bibliothèque logicielle développée par Dreamworks Animation en 2012 pour mieux manipuler les hiérarchies de données volumiques en 3D (nuages, fumées, liquides...).

La disparition de certains développeurs de solutions entamées en interne chez les grands studios a été une motivation essentielle à la création de l'ASWF, qui vise à garantir les ressources mutualisées pour assurer la survie de ces outils devenus des standards de l'industrie. Une autre motivation forte, y compris de la part des développeurs, est de s'appuyer sur l'expérience de la Linux Foundation, qui est partenaire de l'ASWF depuis sa création, pour proposer une structure de gouvernance, d'infrastructure et de services juridiques, élaborée à l'issue d'une série de travaux collaboratifs avec les studios.

(Historique de création) https://aswf.lfprojects.linuxfoundation.org/wp-content/uploads/sites/69/2022/03/LFResearch_ASWF_Report.pdf

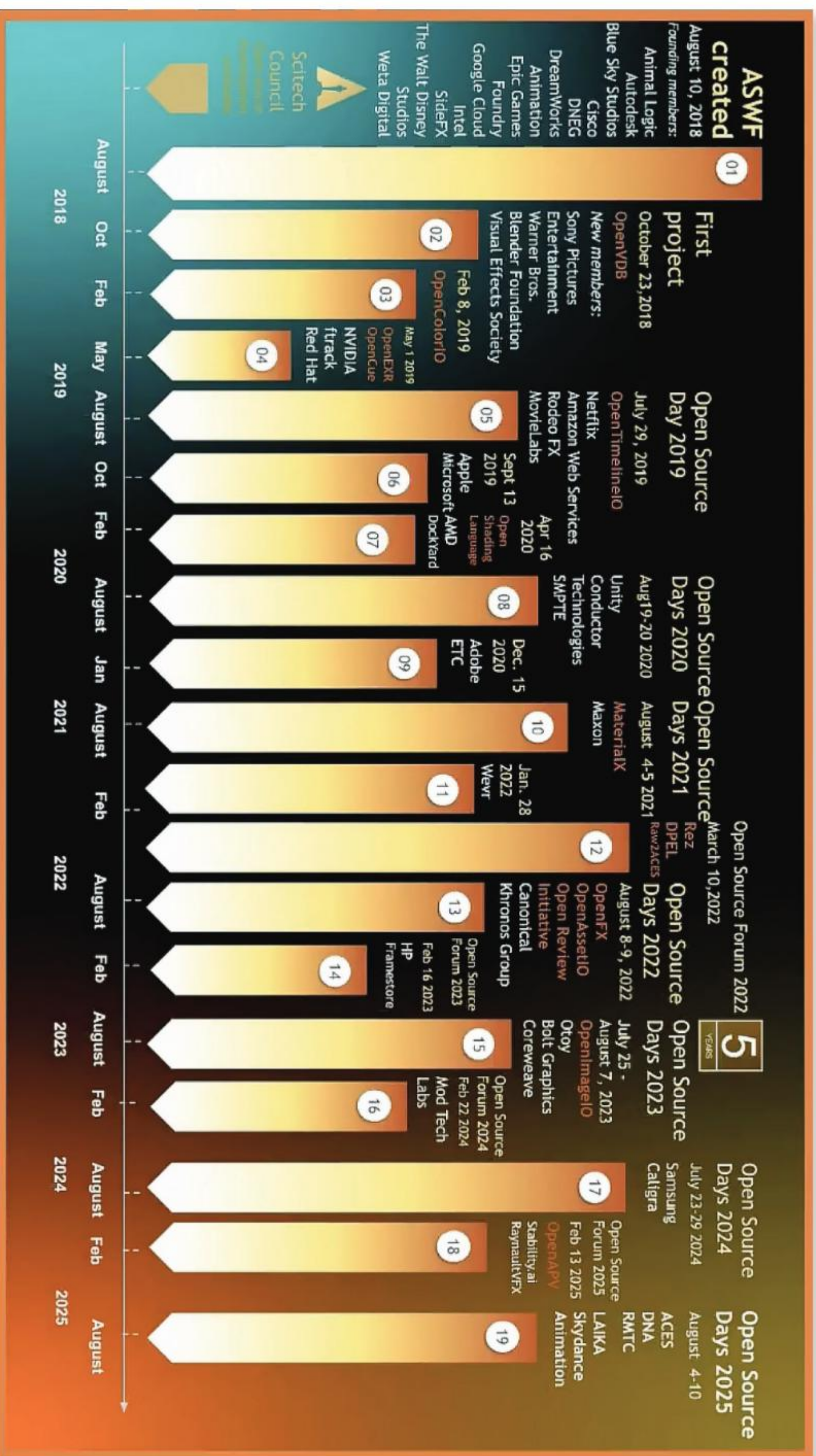
Une quinzaine de projets sont maintenant gérés en direct par l'ASWF ▼

► Projets officiels de l'ASWF, extrait de ASWF Governing Board Overview.pdf.



ÉVOLUTION

◀ Évolution des membres et des projets de l'ASWF, extrait de ASWF Governing Board Overview.pdf.



L'ASWF fédère autour d'elle des communautés de projets open source comme ACES, framework qui permet depuis 2012 de gérer la cohérence et la fidélité créative des couleurs (par la même Academy of Motion Picture Arts and Sciences), et l'Alliance for OpenUSD (formée par Adobe, Apple, Autodesk, Nvidia et Pixar en 2023) pour promouvoir openUSD (Universal Scene Description), format d'interopérabilité de scènes 3D.

L'ASWF propose également une cartographie¹ permettant d'explorer les projets open source dans l'industrie de l'animation et des effets visuels.

Un autre exemple récent pour contrer l'influence du standard fermé Dolby Atmos est l'Immersive Audio Model & Formats (IAMF) format audio tridimensionnel alternatif proposé notamment par Google et Samsung, dont le décodeur est proposé en open

source, au sein de l'Alliance for Open Media². Il sera notamment déployé sous la marque Eclipsa Audio³ dans les nouveaux équipements Samsung (téléviseurs, barres de son, smartphones) et dans Android, Chrome et YouTube par Google.

L'interopérabilité peut avoir un intérêt à court terme, pour faciliter l'échange de fichiers entre deux acteurs, par exemple une livraison du PAD d'un programme audiovisuel à une chaîne de télévision, ou la distribution d'un DCP aux salles de cinéma. Mais l'interopérabilité peut aussi avoir un intérêt dans le long terme. Dans le cas des archives, il faut pouvoir

1. <https://landscape.aswf.io/>

2. <https://aomedia.org/>

3. <https://opensource.googleblog.com/2025/01/introducing-eclipsa-audio-immersive-audio-for-everyone.html>

► Cartographie des projets open source (animation, VFX) proposé par l'ASWF.



utiliser ce que l'on avait stocké dix ans, trente ans, peut-être cent cinquante ans plus tôt.

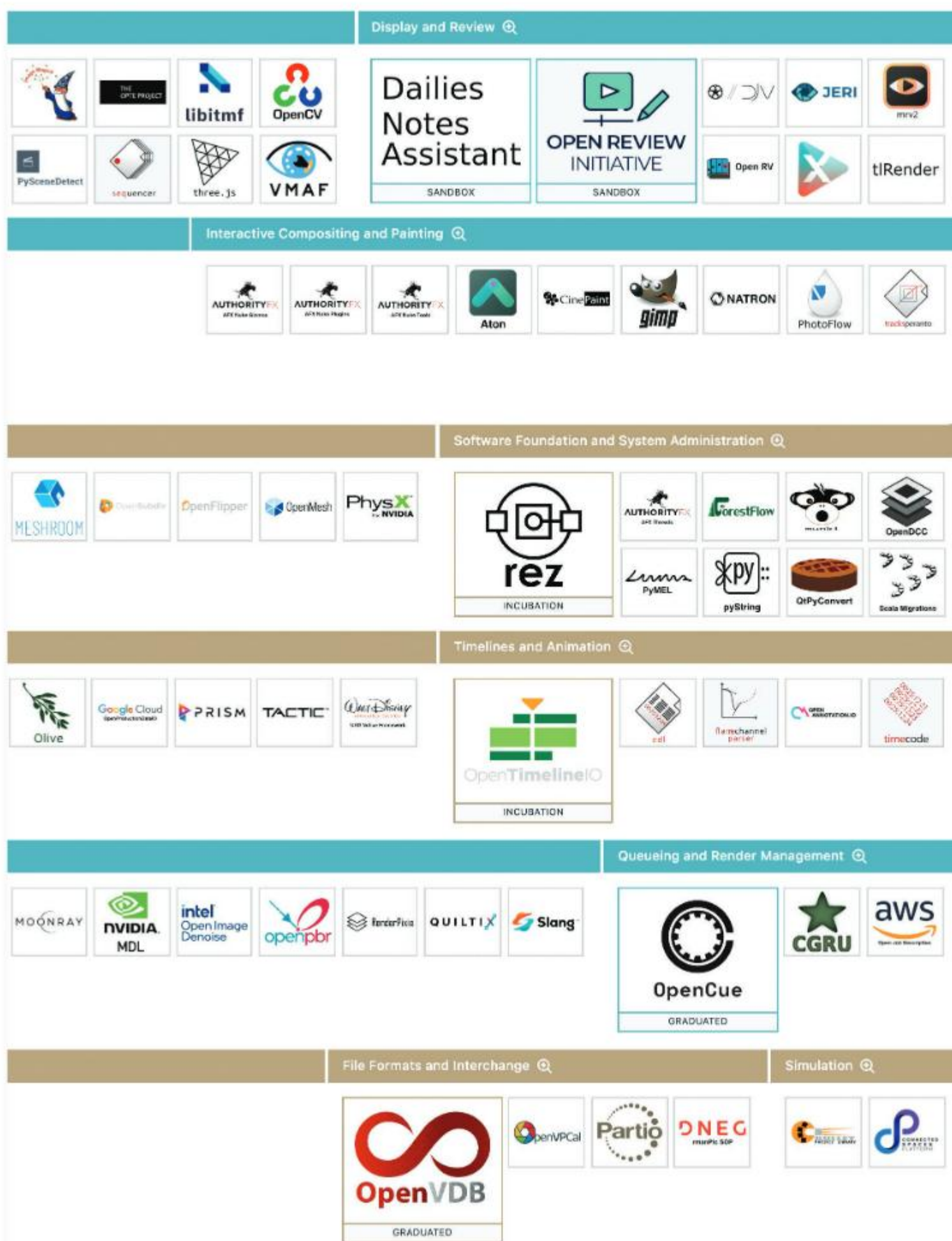
C'est précisément là que l'open source offre une alternative : plutôt que de décrire une spécification, on documente directement une implémentation. Grâce à l'ouverture du code, l'interprétation des données est définie sans ambiguïté, tout en fournissant un composant logiciel immédiatement exploitable pour réaliser une tâche. Cette approche séduit notamment le monde des archives, dont certaines privilégient des solutions open source comme OpenEXR, le conteneur Matroska ou le codec FFV1, en alternative aux normes formelles.

Définir une spécification reste pertinent, et fonder la compatibilité sur le code n'exclut pas un travail de formalisation proche d'une standardisation. Pour les acteurs du web, les processus sont sou-

vent moins formels, mais dès que de nombreux acteurs participent, l'organisation finit par ressembler à celle des SDO (Standard Definition Organisation). Ainsi, pour l'USD, la description du format d'échange est largement implicite dans le code du framework de Pixar ; à mesure que d'autres acteurs s'y engagent, le besoin d'une définition formelle du format se fait sentir.

Aujourd'hui, le modèle du « paywall » appliqué par certaines organisations de normalisation mérite d'être questionné : la difficulté d'accès aux documents constitue un frein pour les développeurs qui cherchent à se conformer aux spécifications. C'est précisément la vertu des standards publiés en open source : un accès libre qui facilite l'implémentation, l'adoption et l'amélioration collective.

Benoît Maujean, Hans-Nikolas Locher



L'OPEN SOURCE : UN CHOIX DE PROJET

La réussite d'un projet open source dépend avant tout de sa capacité à établir et préserver la confiance, celle des utilisateurs comme des contributeurs. Une stratégie claire, transparente et inclusive est la clé pour fédérer une communauté durable autour du projet.

LE RÔLE CLÉ DES CONTRIBUTEURS

La grande majorité des utilisateurs se contentent d'utiliser des outils open source sans y contribuer, que ce soit en tant que développeur ou simple utilisateur. Pourtant, pour assurer à minima la pérennité d'un logiciel et même en accroître le potentiel, il est bénéfique que chacun participe, par exemple en corrigeant des bugs ou en ajoutant des fonctionnalités pour les développeurs, ou en signalant des problèmes, en proposant des exemples, des tutoriels, en enrichissant la documentation ou en organisant des événements pour les utilisateurs. Cette nécessité d'obtenir des contributions de la part de la communauté des utilisateurs et des développeurs repose en grande partie sur la gouvernance du projet open source.

LES PRINCIPES DE GOUVERNANCE

Définir une gouvernance claire revient à formaliser les règles de prise de décision, la gestion des contributions, la propriété intellectuelle, et la résolution des conflits dans un document de gouvernance explicitant les rôles (contributeurs, mainteneurs, utilisateurs), les processus de validation des contributions, et les modalités d'évolution du projet. En complément de ces règles de gouvernance, on trouve souvent une charte de valeurs ou un code de conduite attachés au projet, définissant les règles de comportements communautaires attendus auprès des contributeurs et mainteneurs du projet.

Comme expliqué plus haut, le choix de la licence open source détermine les droits et obligations des utilisateurs et contributeurs. Il est crucial de sélectionner une licence compatible avec les objectifs

du projet, son environnement stratégique et opérationnel, notamment la coexistence plus ou moins fine avec d'autres outils, en analysant les dépendances avec les bibliothèques tierces intégrées pour éviter les conflits de licence.

Pour faciliter la collaboration entre les membres de l'équipe, la gestion des tâches passe par des plateformes de gestion collaborative de projets. Cette gestion est souvent couplée au versioning de code et aux propositions de révision du code (les Pull ou les Merge Requests), pour organiser les tâches de développement, suivre l'avancement, et faciliter la collaboration entre les membres de l'équipe, qu'ils soient internes ou externes. La transparence sur les décisions, les roadmaps, ou les évolutions du projet favorisent l'engagement de la communauté.

Le maintien d'une documentation à jour, avec à minima des paragraphes comme les fichiers Readme et Contributing sur le site de dépôt du code – le Repository sur GitHub/Gitlab – est nécessaire pour guider les nouveaux contributeurs et clarifier les attentes.

Afin de stimuler la participation de la communauté et de soutenir l'adoption du projet, il est essentiel d'animer des espaces de communication ouverts (tels que des forums, des chats ou des listes de diffusion) autour du site web du projet, ainsi que d'organiser des événements (hackathons, meetups) pour encourager et valoriser les diverses contributions.

Plus récemment, l'intégration d'outils de sécurité open source permet de surveiller les vulnérabilités, générer des Software Bill of Materials (SBOM)¹ basés sur un inventaire des composants logiciels et des dépendances qui sont inclus dans le logiciel, et automatiser la détection et la correction des problèmes.

Voir à ce sujet la démarche proposée par l'OpenSSF (le Best Practices Badge Program)², suivie notamment par l'ASWF. Cela permet de garantir la conformité aux réglementations qui peuvent être attendues par certains utilisateurs industriels (comme la directive NIS2, le RGPD ou plus généralement la Digital Operational Resilience Act en Europe), mais aussi de renforcer la confiance des utilisateurs.

1. <https://openssf.org/technical-initiatives/sbom-tools/>

2. <https://www.bestpractices.dev/fr>

BONNES PRATIQUES POUR UN PROJET OPEN SOURCE

- **Gouvernance** : formalisez les règles et les processus de décision.
- **Licence** : choisissez et surveillez la conformité des licences.
- **Outils** : utilisez des plateformes collaboratives open source pour la gestion de projet.
- **Sécurité** : intégrez des outils de sécurité et de conformité dès le départ.
- **Transparence** : documentez et communiquez ouvertement.
- **Communauté** : animez et valorisez les contributions.
- **Amélioration continue** : testez, évaluez et adaptez les pratiques.

Pour aller plus loin :

<https://opensource.guide/fr/>

[Les critères de choix d'un projet Open Source](#)

EXEMPLES TYPES DE GOUVERNANCE

Concrètement, on peut essayer de caractériser les gouvernances des projets open source en trois catégories, avec des variantes pouvant glisser de l'une à l'autre, en fonction de la taille et la philosophie portée par le projet :

1. Les « dictateurs éclairés » avec un modèle centralisé où un voire quelques contributeurs principaux – souvent les créateurs initiaux, prennent les décisions finales.

Exemples :

- Linux, avec Linus Torvalds,
- Python, avec Guido van Rossum avant son départ en 2018.

2. Les entreprises porteuses : une entreprise finance et contrôle majoritairement le développement d'un projet open source, tout en acceptant des contributions externes.

Exemples :

- Epic Games avec Unreal Engine (moteur 3D temps réel) ;
- Meta avec React (bibliothèque logicielle pour créer des interfaces web interactives et dynamiques) ;
- Google avec Kubernetes (framework pour automatiser la gestion et le déploiement d'applications sur plusieurs serveurs) ou Pixar avec OpenUSD dans sa configuration initiale (avant la création de l'Alliance for OpenUSD).

Hormis le code source hébergé et disponible sur la plateforme de dépôt (le dépôt GitHub), la gestion communautaire s'apparente davantage à celle d'un logiciel propriétaire, avec des interactions similaires à celles entretenues avec les utilisateurs d'une solution commerciale.

3. Les organisations « démocratiques » : la gouvernance est organisée de manière collaborative, s'appuyant sur des processus ouverts et des prises de décision collectives, validées par des votes au sein de comités dédiés.

▼ Ton Roosendaal
à la Blender
Conference 2025.



© Photo : Blender

On peut notamment citer :

La Blender Foundation³ définit la mission, la gouvernance et la stratégie globale sur [blender.org](https://www.blender.org), notamment la supervision du développement du logiciel et l'organisation du financement (via le Development Fund). En complément, le Blender Institute s'occupe des activités pratiques, notamment la gestion des projets quotidiens, tout en restant sous la supervision de la fondation (gestion des bureaux, administration, embauche des développeurs internes, organisation de la Blender Conference, gestion du Blender Store, participation à des événements, gestion des subventions de développement, support opérationnel du site). En parallèle, depuis 2020, Blender Studio gère des activités de production de courts-métrages d'animation, pour créer et maintenir des pipelines de production, tester les dernières versions et partager les connaissances de production autour de la production de films. La Fondation Blender fournit depuis 2020 des rapports détaillés sur les activités et les finances du projet Blender (un point fait cependant toujours débat dans la communauté : le recours à la licence GPL pour la majorité du code, y compris pour les add-ons en scripts Python, ce qui limite parfois les contributions externes).

L'Academy Software Foundation⁴ fait partie juridiquement de la Linux Foundation⁵, qui elle-même gère plus de 900 projets open source. L'organisation de l'ASWF repose donc sans surprises sur le mode de gouvernance préconisé par la Linux Foundation. Ainsi, le Governing Board de l'ASWF est responsable du marketing, de la supervision des activités et des décisions budgé-

taires de la Fondation. Il ne prend pas de décisions techniques, si ce n'est de suivre les activités du Technical Advisory Council (TAC) et définir le champ d'action global de l'ASWF. Le TAC établit les meilleures pratiques, détermine les besoins en ressources et en développement (infrastructure, gestion du versioning de code, outils collaboratifs, services de monitoring, ...). Le TAC propose les nouveaux projets à adopter et coordonne les initiatives et opportunités entre les différents projets de la communauté.

L'ASWF gère des projets et des groupes de travail avec des processus de mise en place et de développement spécifiques.

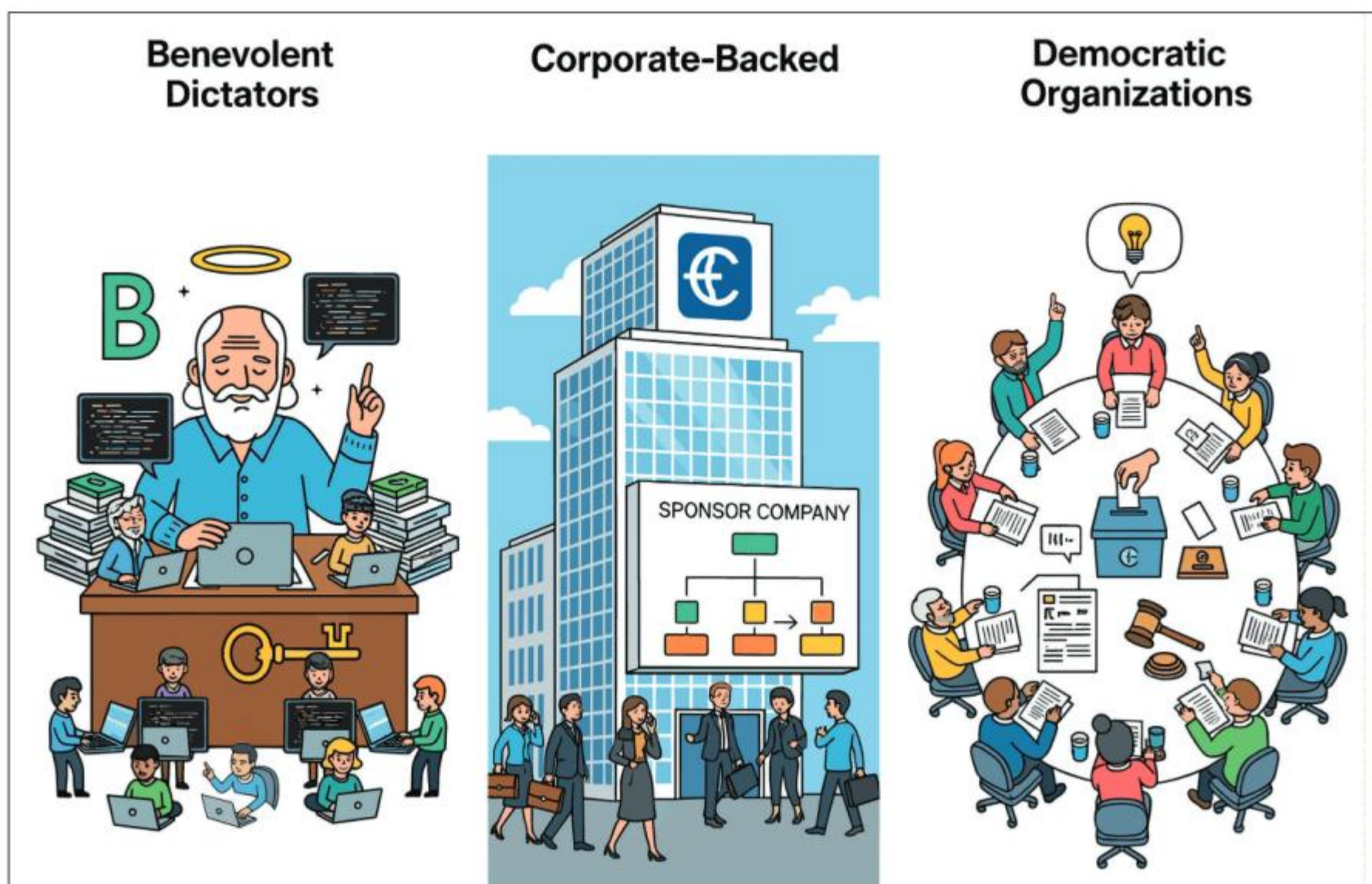
Les projets suivent un processus de Cycle de Vie permettant de définir les modalités de contribution d'un projet open source à l'ASWF, notamment les étapes nécessaires pour chacune des phases de développement après l'acceptation du projet. Avant d'être adopté, un projet doit passer par les étapes de Sandbox⁶ (permettant notamment de définir les relations haut niveau du projet et de sa communauté avec l'ASWF, notamment en termes de communication) et d'incubation (pour respecter un certain nombre de critères techniques en termes de structure de code, de sécurité et de feuille de route).

3. <https://www.blender.org/about/foundation/>

4. <https://www.aswf.io>

5. <https://www.linuxfoundation.org/>

6. <https://tac.aswf.io/process/lifecycle.html#-sandbox-stage>



© CST / Ficam

Les groupes de travail ont un mode de fonctionnement plus léger pour étudier un sujet précis, formuler des recommandations, pouvant aboutir à la soumission d'un projet.

Pour plus de détails, voir les informations mises à jour⁷.

La Linux Foundation propose une boîte à outils⁸ pour gérer la sécurité du code, les calendriers des événements...

– L'Apache Software Foundation, avec un Board of Directors (pour définir les grandes lignes stratégiques) et des Project Management Committees, responsables de la gestion active d'une ou plusieurs communautés spécifiques. L'ASF gère près de 300 projets dans une quinzaine de domaines d'activité. Il fournit également depuis 2016 des rapports détaillés⁹ sur l'activité et les finances de la fondation. Pour plus de détails, voir ici How it works¹⁰.

En contrepartie de la structure, de l'organisation et du rayonnement offerts, les deux fondations (ASWF et ASF) récupèrent la propriété intellectuelle des projets open source qu'elles hébergent.

FINANCEMENT D'UN PROJET OPEN SOURCE

Nombre de projets open source dépendent de développeurs bénévoles ou soutenus par des entreprises directement intéressées par le logiciel. Sans financement stable, un projet open source peut stagner ou s'arrêter, même si le code source reste toujours accessible et utilisable par d'autres projets.

Les principaux modes de financement d'un projet open source incluent le sponsoring d'entreprises, la prestation de services (support, formation, développement spécifique), le financement participatif (crowdfunding), les bourses et subventions, les programmes de primes (bounties). Chaque mode de financement peut être combiné avec d'autres, selon la nature du projet, sa maturité, et la structure de sa communauté.

– **Sponsoring d'entreprises** : les entreprises qui dépendent d'un projet open source peuvent le soutenir financièrement par l'embauche de salariés dédiés, combinée souvent via des niveaux de parrainage avec des partenaires académiques ou industriels, en échange de reconnaissance ou d'avantages spécifiques. Ce modèle est considéré comme l'un des plus durables, car il est porté par les acteurs ayant un intérêt direct pour la pérennité du projet.

Voir ici le type de sponsoring financier proposé par l'ASWF et Blender :

<https://www.aswf.io/join/>

<https://fund.blender.org/corporate-memberships/>

– **Prestation de services** : les développeurs ou les sociétés créées autour du projet proposent des services payants : support technique, formation, intégration, développement de fonctionnalités spécifiques, ou conseil. Ce modèle permet de répondre à des besoins concrets des utilisateurs professionnels, voire de participer au développement global du projet lorsque ces ajouts sont remis au pot commun, notamment en fonction de la structure du code et de la licence. A noter que ce modèle peut également être combiné par l'approche « open core » : le cœur du logiciel reste open source, mais des fonctionnalités avancées sont proposées sous licence propriétaire payante¹¹.

– Une autre variante consiste à proposer un modèle de **licence multiple** : l'utilisateur peut utiliser la version libre pour les tests, l'évaluation, le développement de la validation du concept et le déploiement à petite échelle ; l'utilisateur peut ensuite aller vers la version commerciale s'il souhaite déployer le logiciel à grande échelle ou dans des produits distribués propriétaires (c'est le cas par exemple de Qt, framework de conception d'interfaces graphiques).

– **Financement participatif (crowdfunding)** : les appels à dons sur le site web du projet ou les campagnes sur des plateformes comme Github, Kickstarter, Indiegogo ou Patreon permettent de collecter des fonds auprès de la communauté des utilisateurs, sous forme de dons ponctuels ou d'un soutien récurrent. Ce mode de financement dépend en grande partie de l'efficacité des canaux de communication du projet.

– **Bourses, subventions et financements publics** : certains projets bénéficient de subventions de fondations, d'institutions ou d'organismes publics, qui souhaitent soutenir l'innovation ou la souveraineté numérique. Le positionnement open source d'un projet peut également donner lieu à un bonus de financement lors d'appels à projets d'innovation nationaux ou européens.

– **Programmes de primes (bounties)** : des utilisateurs ou entreprises mettent en place des primes pour le développement de fonctionnalités ou la correction de bugs spécifiques (bug bounties et source bounties). Les développeurs sont rémunérés à la tâche, selon les résultats livrés.

Benoît Maujean

7. <https://tac.aswf.io/process/lifecycle.html#-sandbox-stage>

8. <https://lfx.linuxfoundation.org/>

9. <https://www.apache.org/foundation/reports.html>

10. <https://www.apache.org/foundation/how-it-works/>

11. <https://goodtech.info/coss-c-est-quoi-un-commercial-open-source-software/>

L'APPROCHE « OPEN SOURCE » POUR L'IA

Dans le domaine de l'Intelligence Artificielle, la notion d'open source soulève depuis toujours des questions complexes et des divergences d'opinions.

LES FERMENTS OPEN SOURCE DE L'IA

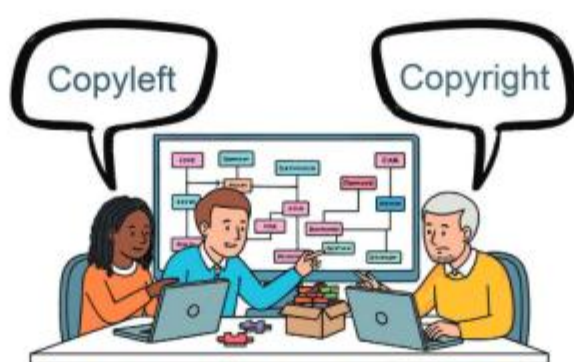
À la base de la dynamique actuelle entourant cette technologie, les frameworks d'apprentissage profond open source tels que TensorFlow (développé par Google Brain à partir de fin 2015 et sous licence Apache-2.0) et PyTorch (développé par le laboratoire de recherche en IA de Facebook à partir de fin 2017 et sous licence permissive Facebook) ont transformé le paysage de l'IA en rendant plus accessibles les technologies sous-jacentes de la plupart des modèles d'apprentissage profond. De même en 2017, Facebook et Microsoft publient sous licence Apache-2.0 le framework Open Neural Network Exchange (ONNX), permettant d'échanger et d'exécuter des modèles d'intelligence artificielle entre différents outils et environnements d'apprentissage automatique.

En 2018 émergent les premières architectures logicielles Transformers, révolutionnaires pour le traitement naturel du langage, avec BERT (bidirectional encoder representations from transformers) développé par Google sous licence Apache-2.0, ainsi que GPT (generative pre-trained transformer), créé par OpenAI, sous licence MIT transformée pour la version GPT-2. Depuis, OpenAI n'a cependant plus publié le code source ou les poids pré-entraînés des modèles suivants comme GPT-3 ou GPT-4 qui sont à la base de ChatGPT leur agent conversationnel emblématique lancé en 2022 (même si leurs fonctionnalités peuvent être intégrées par les développeurs via l'API commerciale d'OpenAI). Récemment, OpenAI semble revenir vers une plus grande ouverture en publiant ces derniers modèles de langage sous licence Apache-2.0, sous l'appellation GPT-OSS.

Enfin, en 2022 l'entreprise anglaise Stability AI sort Stable

Diffusion, un modèle d'apprentissage automatique permettant de générer des images numériques de différents styles, y compris photoréalistes, à partir de descriptions en langage naturel (prompts). Une partie du code¹ a été publiée en open source sous licence MIT, mais les modèles entraînés sont publiés sous différentes licences avec des restrictions :

- les premiers modèles (Stable Diffusion 1.x and 2.x) étaient sous licence Creative ML OpenRAIL-M qui permet une large utilisation, y compris pour des applications commerciales, mais impose des restrictions sur certaines activités, telles que l'utilisation criminelle, le harcèlement, l'exploitation des mineurs et la discrimination,
- les modèles suivants (Stable Diffusion 3 and 3.5) sont régis par la licence Stability AI Community License qui reste permissive pour les individus et les organisations dont les revenus annuels sont inférieurs à un million de dollars, autorisant la recherche libre, l'utilisation non commerciale et l'utilisation commerciale limitée. Pour les entreprises commerciales dont le chiffre d'affaires est supérieur à un million de dollars, une licence d'entreprise Stability AI distincte est requise. Cette licence communautaire est non exclusive, mondiale, non transférable, non sous-licenciable, révocable et libre de droits pour les utilisateurs éligibles,
- les données d'entraînement de la première version sont basées sur LAION-5B (An open large-scale dataset for training next generation image-text models) dont la licence initiale était sous Creative Commons CC-BY-4.0 pour le jeu des métadonnées, « les images restant sous leur propre copyright » (sic). La nouvelle version Re-LAION-5B est une version nettoyée – notamment des images Child Sexual Abuse Material, sous licence Apache-2.0.



© CST / Ficom

1. <https://github.com/Stability-AI/generative-models/blob/main/LICENSE-CODE>

LA TENDANCE À L'OPEN-WASHING

Les outils proposés par L'Intelligence Artificielle Générative offrent de nos jours des possibilités sans précédent, mais leur utilisation suscite des inquiétudes quant à leur transparence, leur reproductibilité, leur partialité et leur sécurité qui sont, comme on l'a vu, les fondements mêmes de l'open source. De nombreux modèles d'IAG dits « open-source » ne disposent pas des éléments nécessaires à une compréhension et une reproduction complètes, et certains utilisent des licences restrictives, une pratique connue sous le nom d'open-washing.

Les avantages et les risques des modèles ouverts font l'objet de nombreux débats.

D'un côté, l'accessibilité et la transparence des modèles ouverts offrent des avantages notables par rapport aux modèles à code source fermé, notamment en termes :

- de performance et de sécurité grâce au développement collaboratif et à l'audit distribué,
- d'adaptabilité et de personnalisation pour divers domaines et langages,
- de progrès dans les domaines scientifiques (science ouverte et reproductible).

De l'autre, l'ouverture des modèles comporte des risques, notamment en facilitant l'accès à :

- la génération de désinformation (deepfakes audio ou vidéo),
- la génération de contenus illégaux (usurpation d'identité).

Trouver un équilibre entre l'exploitation de l'innovation des modèles ouverts et la gestion des risques associés reste un défi majeur dans le paysage en constante évolution de l'IA, en particulier de l'IA Générative.

Ainsi, les fournisseurs de modèles d'IA à usage général (les foundation models) adoptent des stratégies de communication variées vis-à-vis de l'open source, susceptibles de changer au gré des évolutions du marché et de la réglementation.

- Meta publie ses modèles Llama, mais avec des restrictions commerciales (seuil de chiffre d'affaires mensuel de 700 millions), territoriales (les particuliers et les entreprises de l'UE ne peuvent pas l'utiliser directement) et techniques (interdiction d'utiliser les résultats du modèle pour améliorer d'autres LLMs). Meta reste vague sur l'origine de ses données d'entraînement (on sait cependant qu'elles comprennent les données récupérées sur Facebook et Instagram).

- Mistral AI publie la plupart de ces codes sources et ses modèles (licences MIT et Apache) et utilise principalement des « données open source soigneusement sélectionnées » pour entraîner ses modèles, mais la liste exacte de ces données d'entraînement n'est pas publique.

- DeepSeek publie les modèles de DeepSeek-R1 sous licence MIT, sans restriction d'usage commercial ou dérivatif, mais reste vague sur la provenance des données d'entraînement.

- Qwen de Alibaba Cloud qui publie ses modèles sous licence open-source (Apache-2.0 pour les versions 2.5), sans rien préciser sur l'origine des données d'entraînement.

- A l'opposé, Lucie, développée par l'entreprise française Linagora, se positionne comme une intelligence artificielle entièrement open source, conçue de manière éthique, respectueuse de la vie privée et du droit d'auteur, et visant à garantir la souveraineté numérique en France et en Europe, en s'appuyant sur des données et des modèles ouverts et transparents.

Il faut également signaler un écueil récurrent pour les laboratoires de recherche académiques autour de l'IA : la mise à disposition de tout ou partie des trois composants (données, code et paramètres du modèle) sous le label « research only » donc non-utilisables à des fins industrielles. Ceci est à replacer dans l'évolution des pratiques académiques où il ne suffit plus de présenter un algorithme avec des évaluations par rapport à l'état de l'art : au-delà de la description algorithmique et des comparaisons chiffrées, la publication du code est devenue un standard pour que les résultats soient véritablement testés et reproduits. Mais par souci déontologique (la source des composants logiciels est disponible pour faire avancer la recherche reproductible, mais ne peut pas - en théorie - être utilisé pour des intentions industrielles plus ou moins nobles, voire nuisibles à la société), soit dans l'espoir de pouvoir générer des revenus grâce à la vente de licences commerciales. Cette posture va à l'encontre de l'open innovation puisqu'elle empêche son utilisation dans un contexte open source qui lui n'interdit pas les usages industriels. Un exemple récent et emblématique dans le monde du Computer Graphic/Computer Vision : la publication très remarquée du « 3D Gaussian Splatting »² sous licence « research only » (qui a effectivement permis à l'INRIA de vendre une petite vingtaine de licences) mais qui a finalement donné lieu au développement de variantes open source dont celle de NerfStudio sous licence Apache-2.0 ou LichtFeld sous licence GPL.

2. <https://github.com/graphdeco-inria/gaussian-splatting>

LES TENTATIVES EN COURS DE NORMALISATION POUR L'IA

Face à ce positionnement anarchique et controversé des différents acteurs, deux initiatives visent à poser les bases d'une démarche pour étendre les concepts de l'open source à l'Intelligence Artificielle : l'Open Source AI Definition par l'Open Source Initiative³ et le Model Openness Framework par la Linux Foundation et les Data Generative AI Commons.

Lancée en août 2022 par l'Open Source Initiative, l'Open Source AI Definition⁴ définit les trois composants qui structurent les systèmes d'IA et qui doivent permettre d'en faciliter l'usage, l'étude, la modification et le partage :

1. les données utilisées pour entraîner l'IA,
2. le code nécessaire pour entraîner et faire tourner l'IA (entraînement et inférence),
3. les paramètres du modèle : les poids du modèle résultant de l'entraînement.

La position actuelle de l'OSI est que la transparence concernant les données doit avant tout porter sur le partage des informations sur les données, pas nécessairement sur le partage des données elles-mêmes. Ce qui entraîne des débats encore vifs sur la nécessité de disposer réellement des données pour pouvoir effectivement étudier ou modifier le système d'intelligence artificielle qui les utilise, notamment bien sûr dans le cadre de la recherche académique ou industrielle.

Lancé en septembre 2023 par la Linux Foundation, le Model Openness Framework identifie trois catégories de composants d'un système d'IA :

1. le code (le code d'évaluation, le code de pré-processing, l'architecture du modèle, les bibliothèques logicielles, le code d'entraînement, le code d'inférence),

2. les données (les jeux de données brutes, les données d'évaluation, les échantillons de sortie, les paramètres et les poids du modèle, les fichiers de configuration),

3. la documentation (les data cards - structure des données, les publications de recherche, les résultats d'évaluation, les model cards - structure du modèle, les rapports techniques).

Chaque composant, pour être ouvert, doit être publié suivant ces trois règles :

1. pour le code, avec une licence OSI,
2. pour les données, avec une licence ouverte permissive de données, de préférence la Creative Commons CC-BY-4.0 (utilisation libre, y compris commerciale, à condition de citer l'auteur) ou la Permissive Community Data License Agreement (CDLA-Permissive-2.0),
3. pour la documentation, avec une licence ouverte permissive de contenus, de préférence la Creative Commons CC-BY-4.0.

Le Model Openness Framework définit trois classes hiérarchiques d'ouverture d'un système d'IA :

- Le minimum étant la classe III (Open Model) avec l'architecture du modèle de base et l'ensemble final des paramètres du modèle, plus une documentation succincte présentant les capacités et la caractérisation du modèle et des données,
- le modèle intermédiaire, la classe II (Open Tooling) avec, en plus de la classe III, les codes complets d'entraînement et d'inférence, les tests de référence pour valider et quantifier les performances, les bibliothèques et outils pour faciliter l'intégration et compléter la base de code,
- le plus ouvert étant la classe I (Open Science) avec, en plus de la classe II, la publication de recherche détaillée décrivant la genèse du modèle et son évolution, l'ensemble des données brutes utilisées pour l'entraînement du modèle, les poids des points de contrôle décrivant l'évolution complète du modèle, les fichiers journaux fournissant encore plus d'informations bas niveau.

Pour plus de détails sur le contenu et les motivations, voir leur livre blanc Model Openness Framework Whitepaper⁵.

3. <https://opensource.org/>

4. <https://opensource.org/ai/open-source-ai-definition>

5. <https://docs.google.com/document/d/1RUNrs4fIA5YsikXTPu1jWBH1BAum-CyeG/edit?pli=1&tab=t.0>

CONSÉQUENCES ÉCONOMIQUES ET JURIDIQUES DU POSITIONNEMENT « OPEN SOURCE » POUR L'IA

Les défis liés à la transparence des algorithmes et à la traçabilité des données d'entraînement ne se limitent pas aux aspects techniques, sécuritaires ou éthiques. La traçabilité des données reste aussi un enjeu majeur par rapport à la rémunération des auteurs des œuvres ayant servi à générer ces données (droits d'auteur et droits voisins). La mise en application de l'IA Act en Europe (entré en vigueur le 1er août 2024 et pleinement applicable le 2 août 2026) notamment les articles 4 et 53, comporte deux clauses importantes :

1. l'exception au droit d'auteur dans la fouille de données à des fins de recherche scientifique ne s'applique pas si l'auteur a fait valoir son droit de retrait d'utilisation de l'œuvre (opt-out),

2. l'obligation de mettre à la disposition du public un résumé suffisamment détaillé du contenu utilisé pour entraîner les modèles d'IA à usage général.

A noter que la gestion des droits d'auteur se pose à la fois aussi bien en termes d'entrée des systèmes d'IA (les données d'entraînement) qu'en termes de sortie (les données générées par l'IA), avec les questionnements sur l'originalité et l'apport créatif de l'œuvre, lorsqu'on utilise des outils d'IA générative.

Par ailleurs, il convient de noter que les œuvres produites peuvent à leur tour servir de données d'entraînement pour d'autres créations. Ce qui soulève un risque systémique, puisque la qualité et la diversité des contenus génératifs dépendent des données d'origine. Si les modèles génératifs ont tendance à s'alimenter eux-mêmes, en utilisant des données synthétiques issues d'autres modèles similaires, le manque de traçabilité pourrait également aboutir à une forme d'autophagie ou de dégénérescence créative⁶. Même si des pistes semblent possibles en accumulant des générations successives de données synthétiques parallèlement aux données réelles originales pour éviter l'effondrement du modèle⁷.

Les conditions d'application de cette obligation à la transparence avec l'IA Act restent à définir sur le plan juridique et économique. A ce jour, les fournisseurs de modèles d'IA à usage général peuvent consulter :

- le récent code des bonnes pratiques⁸ qui propose des checklists opérationnelles pour la mise en conformité avec l'IA Act sur les volets transparence, respect du droit d'auteur et sécurité,
- les premières décisions du Bureau EU de l'IA⁹.

Il faut également noter que certaines initiatives visent à démontrer concrètement qu'il est possible de développer un Large Language Model sur un corpus de données d'entraînement entièrement ouvert, sans utiliser de contenus protégés par le droit d'auteur.

Citons par exemple :

- « Common Corpus » par la société française Pleias (première version en novembre 2024, deuxième version en février 2025) : Il s'agit actuellement de la plus grande collection ouverte de données éthiques multilingue pour le pré-entraînement des Large Language Models, avec environ 2 000 milliards de tokens (environ 500 milliards de mots, dont 110 milliards en français). Les données sont soit non soumises au droit d'auteur, soit sous licences permissives¹⁰.

Il reste cependant quelques conflits sur le « Share Alike » des documents issus de Wikipedia.

- « Common Pile v.0.1 » par la communauté internationale EleutherAI, principalement en langue anglaise.

La principale valeur de ces modèles réside dans la preuve tangible qu'il est possible de construire un Large Language Model à partir d'un corpus entièrement ouvert, reproductible, qualitatif et diversifié sans exploiter d'œuvres protégées par des droits d'auteur.

Dans le domaine de l'IA générative pour la vidéo, deux entreprises revendiquent des modèles générés à partir de sources licites... mais sans en donner les références.

- Moonvalley avec le modèle Marey : « Trained only on licensed, high-resolution footage. No scraped content. No user submissions. No legal gray zones. A foundation filmmakers can trust to protect their work and meet the highest standards of authorship »,

- Bria avec le modèle éponyme: <https://bria.ai/>, « Visual Gen AI Platform For Developers, Trained with 100 % Licensed Data ».

Si l'on admet que les œuvres protégées intégrées aux jeux de données confèrent une valeur économique aux modèles génératifs, alors leur identification précise et leur traçabilité deviennent les conditions sine qua non pour établir des mécanismes de compensation, qu'ils relèvent du droit d'auteur, des licences collectives ou de modèles innovants comme les micro-paiements automatisés.

Benoît Maujean

6. <https://arxiv.org/pdf/2402.07043> / <https://arxiv.org/pdf/2307.01850>

7. <https://arxiv.org/abs/2404.01413>

8. <https://digital-strategy.ec.europa.eu/fr/policies/contents-code-gpai>

9. <https://digital-strategy.ec.europa.eu/fr/policies/ai-act-governance-and-enforcement>

10. <https://huggingface.co/blog/Pclanglais/common-corpus>

ENTRETIEN AVEC ARNAUD ROLAND, DIRECTEUR ADJOINT DU NUMÉRIQUE, CNC

Quel est le positionnement du CNC par rapport à l'open source et quelles actions sont ou seront mises en place ? Telles sont les questions que nous avons posées à Arnaud Roland, directeur adjoint du numérique au CNC

► **Quelles sont les valeurs de l'open source que le CNC essaie de promouvoir ? Quels en sont les enjeux pour les ICC ? Pour la culture ?**

ARNAUD ROLAND. Philosophiquement, l'open source est fondé sur la notion de partage d'une part et sur la notion d'écosystème d'autre part. En cela, elle est intrinsèquement liée à la reconnaissance de « communs numériques ». Concrètement, l'open source fait référence à un modèle de logiciel où le code source est ouvert, c'est-à-dire que les utilisateurs ont la liberté d'accéder à ce code source, mais également de le modifier et de le redistribuer, pour le rendre disponible à d'autres personnes, qui peuvent ensuite en faire ce qu'elles veulent. Il s'oppose aux logiciels dits « propriétaires », dont le code source est gardé secret par le propriétaire du logiciel. Un code source fermé ne peut pas être modifié ni copié de manière légale, et l'utilisateur paie uniquement pour pouvoir utiliser le logiciel conformément à l'usage prévu par le propriétaire.

La puissance des GAFAM et la domination technologique américaine s'est construite essentiellement sur cette notion de logiciel propriétaire – même s'ils ont également investi le domaine de l'open source. Nous avons aujourd'hui la responsabilité – et pas seulement au CNC – de proposer un modèle alternatif, pour construire une souveraineté et une puissance technologiques qui reposent sur des principes économiques et sociétaux différents : on va aller chercher une autonomie stratégique dans laquelle, sans chercher à tout faire en interne, il s'agit de préserver à l'échelle nationale ou européenne une capacité autonome d'ap-

préciation, de décision et d'action dans l'espace numérique et – en reposant sur une communauté active et massive de contributeurs – atteindre une excellence technique qu'une entreprise aurait du mal à atteindre désormais dans un environnement propriétaire, sauf à engloutir des investissements gigantesques qui ne sont aujourd'hui plus possible de réunir en Europe.

Ce point est fondamental pour les ICC qui, depuis une quinzaine d'années, ont été parmi les filières les plus transformées par la massification des usages numériques et la transformation des outils de création, avec des cycles d'innovation toujours plus rapides et une remise en question des positions établies.

L'on peut même analyser cela comme le passage d'industries culturelles à industries de contenus. En effet, de filières ICC très segmentées reposant sur des préceptes économiques hérités de l'industrialisation des processus de production (parfaite reproductibilité des œuvres, économies d'échelles, division du travail de création en une chaîne de valeur verticale), on a basculé dans un environnement économique totalement différent, reposant à la fois sur (i) la plateformes des usages et des processus de création, (ii) l'accroissement des effets de réseaux qui se sont substitués aux traditionnelles économies d'échelle, (iii) la question de la découvrabilité dans un univers de sur-offre culturelle, (iv) la réinvention permanente des outils techniques de production qui ont un impact sur l'identité formelle des œuvres auxquelles nous sommes confrontés et (v) une tentative de prise de pouvoir progressive des acteurs de la tech sur la filière créative. Si l'on devait prendre l'exemple des USA, c'est presque un combat entre San Francisco et Los Angeles auquel nous assistons.

La France a toujours été reconnue au niveau mondial pour son excellence dans les filières créatives, qu'il s'agisse de notre capacité à penser des œuvres originales ou notre excellence technique pour les produire. Si nous voulons conserver cette capacité créative et gagner en autonomie économique et technique, l'open source est l'une des voies à emprunter.





► Quelles actions incitatives le CNC a-t-il mis en place auprès de porteurs de projets de développement ?

A.R. Lorsque l'aide aux moyens techniques a été créée en 2023, nous avons souhaité laisser la possibilité aux experts siégeant en commission d'encourager les développements informatiques reversés à la communauté. Cette aide sélective, qui soutient financièrement des projets techniques innovants a vocation à répondre aux enjeux de compétitivité, d'attractivité et d'indépendance des filières cinéma, audiovisuel et jeu vidéo : aides aux études, à l'investissement, à la création de postes... Quant aux dépenses de recherche et développement, un taux de soutien bonifié est accessible pour tout projet open source : prise en charge jusqu'à 60 % des coûts de développement, proposition très incitative pour bon nombre d'entreprises.

► De par sa position en amont de certains projets, quelles conclusions le CNC peut-il tirer ?

A.R. Les secteurs auxquels nous nous adressons sont extrêmement variés : tournage, postproduction, VFX, animation, jeu vidéo, immersif, diffusion... Nous identifions deux sous-ensembles de structures actives en open source : le premier développe des outils et pipelines sans modèle économique lié. Leur démarche est de s'ouvrir à d'autres studios grâce à davantage d'interopérabilité, échanger avec la communauté, voire partager la fabrication d'œuvres pour des questions de volumes ou de compétences spécifiques. Le second se positionne davantage en tant qu'éditeur logiciel. Dès lors, seulement une partie du code est reversé à la communauté. L'autre reste entre les mains des développeurs maison et c'est sur ce point que l'entreprise va monétiser sa valeur ajoutée via une offre de services ou de produits.

▲ L'animation en mode open source au programme du RADI RAF.

Dans tous les cas, nous constatons une vraie « philosophie » open source avec, à la clé, l'animation d'une communauté, une évolution collective des outils reversés et bien évidemment une indépendance numérique vis-à-vis des solutions logicielles dites « fermées ». Cela peut paraître contre-intuitif, mais adopter une démarche ouverte permet également de répondre plus facilement à des enjeux de sécurité. Tous les projets soutenus arrivent à leur terme, avec publication du code, très souvent accompagnés d'actions de communication lors de conférences aux RADI à Angoulême ou au Festival d'animation d'Annecy. L'ensemble des œuvres, animées ou en prise de vue réelle, peut potentiellement avoir recours à des outils open source dans leur processus de fabrication et suivi de production.

► Le CNC peut-il proposer quelques statistiques sur les projets soumis et soutenus ?

A.R. Depuis la création de l'aide aux moyens techniques en 2023, nous avons soutenu une vingtaine de projets open source pour un montant total d'environ 1,5 M. Les experts sont particulièrement attentifs au type de licence appliqué et donc au niveau de permissivité, à la prise en charge des coûts de maintenance et d'évolution (open source ne signifie pas forcément gratuit sur le long terme) ainsi qu'à la stratégie de communication de l'entreprise quant à ses reversements. Autrement dit, et sur ce dernier point, un lien public d'accès au code ne suffit pas, il faut faire la démonstration d'une expérience en animation d'écosystème.

Propos recueillis par Frédéric Fermon

POURQUOI LA CST FAIT-ELLE DE L'OPEN SOURCE ?

CONTRIBUER À CONSTRUIRE UN BIEN COMMUN POUR LA PROFESSION

La CST est une association réunissant des technicien·nes du cinéma et de l'audiovisuel, engagée dans la défense d'une vision partagée de la pratique professionnelle et la quête de la « qualité » de l'œuvre qui figure dans nos statuts. Dossier après dossier, cette communauté élabore et promeut des référentiels communs, dont les Recommandations Techniques¹ constituent l'une des expressions les plus concrètes. Chaque Recommandation émise représente un bien commun, soigneusement entretenu par l'ensemble de la collectivité des professionnels de la CST. Dans cette même logique, les logiciels développés par l'association – souvent conçus pour accompagner la création ou l'application de standards et de recommandations – doivent eux aussi être considérés comme des biens communs, porteurs de cette démarche collective.

Il est donc assez logique que le code produit par l'équipe de développement de la CST – à l'exception de certains développements internes qui n'ont pas d'intérêt collectif ou sont couverts par des accords de confidentialité avec des partenaires – le soit avec des licences open source. Cette démarche n'est pas nouvelle pour nous, puisqu'il y a presque dix ans, nous avons proposé avec une licence MIT une contribution pour démontrer la récupération de certificats pour ARCENe. Cependant, notre effort de développement s'est amplifié depuis quelques années et voici les trois projets principaux sur lesquels nous nous sommes investis.

Tout d'abord, la visionneuse USD 3D-Info² permet d'ouvrir des scènes USD (et d'autres formats convertibles), dans une interface simple et intuitive. Elle est placée sous licence Apache 2. Ce format de scènes 3D est issu du studio Pixar, qui a ouvert le code en open source, pour le proposer à la communauté comme candidat à la standardisation. Le code ouvert a deux intérêts, il documente le format qui n'était pas décrit formellement par une spécification³ et il permet de manipuler des scènes USD. La visionneuse proposée permet ainsi, outre l'usage facilité de l'USD à notre commu-

nauté, d'accomplir une veille technologique appuyée par de la mise en œuvre.

Dans le secteur de la conservation patrimoniale, la CST a été impliquée dans la rédaction de la norme européenne EN 17650 CPP – Cinema Preservation Package – décrivant un paquetage pour une conservation numérique long terme.

CaPPuSCHino, réalisé dans le cadre du projet CPP-XSD appuyé par l'Aide aux Moyens Techniques du CNC permet de vérifier la structure des fichiers du CPP. Il s'agit de fichiers schematron, permettant de faire une vérification de conformité des fichiers XML. CaPPuSCHino a été publié, là aussi, avec une licence MIT. Ce développement a été complété d'une contribution du laboratoire VDM, dans le même projet aidé par le CNC, qui complète CaPPuSCHino avec d'autres tests structurels.

Un nouveau projet destiné à faciliter l'adoption du CPP dans les laboratoires a été lancé. Le logiciel de référence cenpackager avait été proposé par le groupe d'experts chargés de l'écriture de la norme. Il servira de base à de nouveaux développements pour le rendre plus intégrable, plus couvrant et plus facile à utiliser. On peut noter, là aussi, un des bienfaits de l'open source, une base de code destinée à un usage (du code de référence) peut être utilisée, ici pour établir un MVP de composant intégrable.

Enfin, la CST a lancé le projet TAMIS en 2022, pour faciliter l'échange de métadonnées à propos des œuvres. Ce projet s'inscrit désormais comme un des cas d'usage du projet européen porté par le consortium européen TEMS dont la CST fait partie, et qui devrait voir le jour en 2027. Au sein de ce projet, la CST défend la même vision d'une structure au service du secteur, et dont les développements logiciels doivent être autant que possible pour les partenaires, en open source.

L'existence de briques open source est une garantie à long terme, pour une approche communautaire, elle donne des gages de non-confiscation, d'une part, et décrit, d'autre part, une partie de la solution technique, même si un travail d'intégration reste à accomplir.

■ Liens vers les projets

<https://github.com/CSTIS/arcenekdm>

<https://gitlab.com/cst-cpp/cappuschino>

<https://gitlab.com/cst-cpp/cpp-checker.git>

<https://gitlab.com/3d-info/3d-info>

<https://cst.fr/tamis/>

1. <https://cst.fr/recommandations-techniques-cst/>

2. <https://cst.fr/3d-info/>

3. C'est un des points qui va changer avec la nouvelle gouvernance formée autour de l'USD par l'AOUSD.

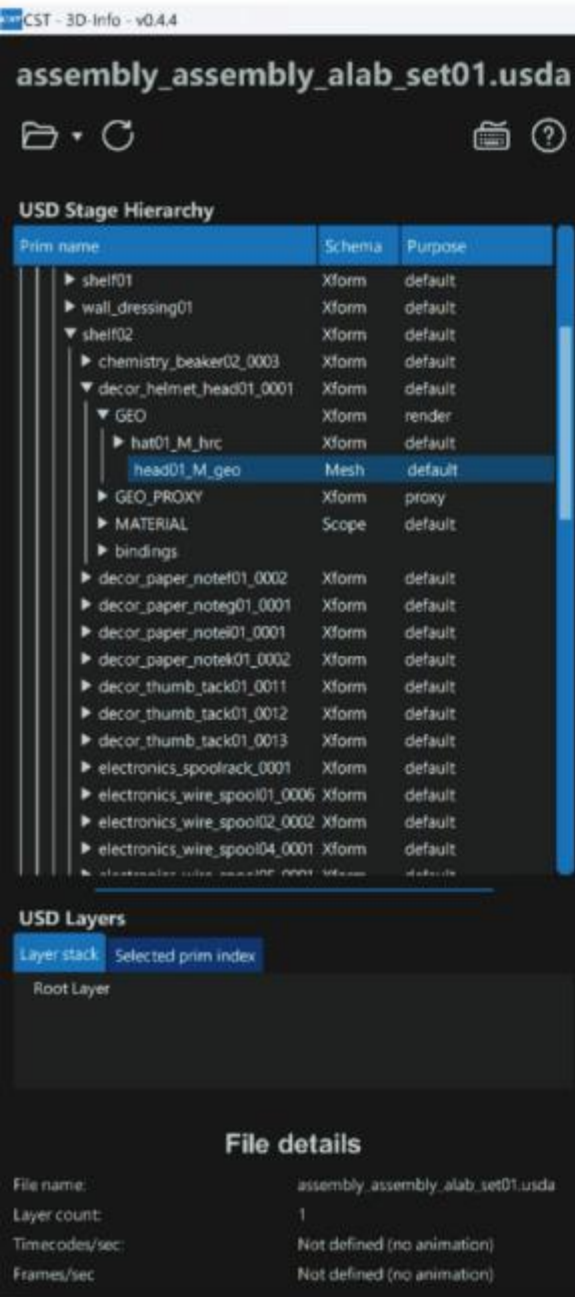
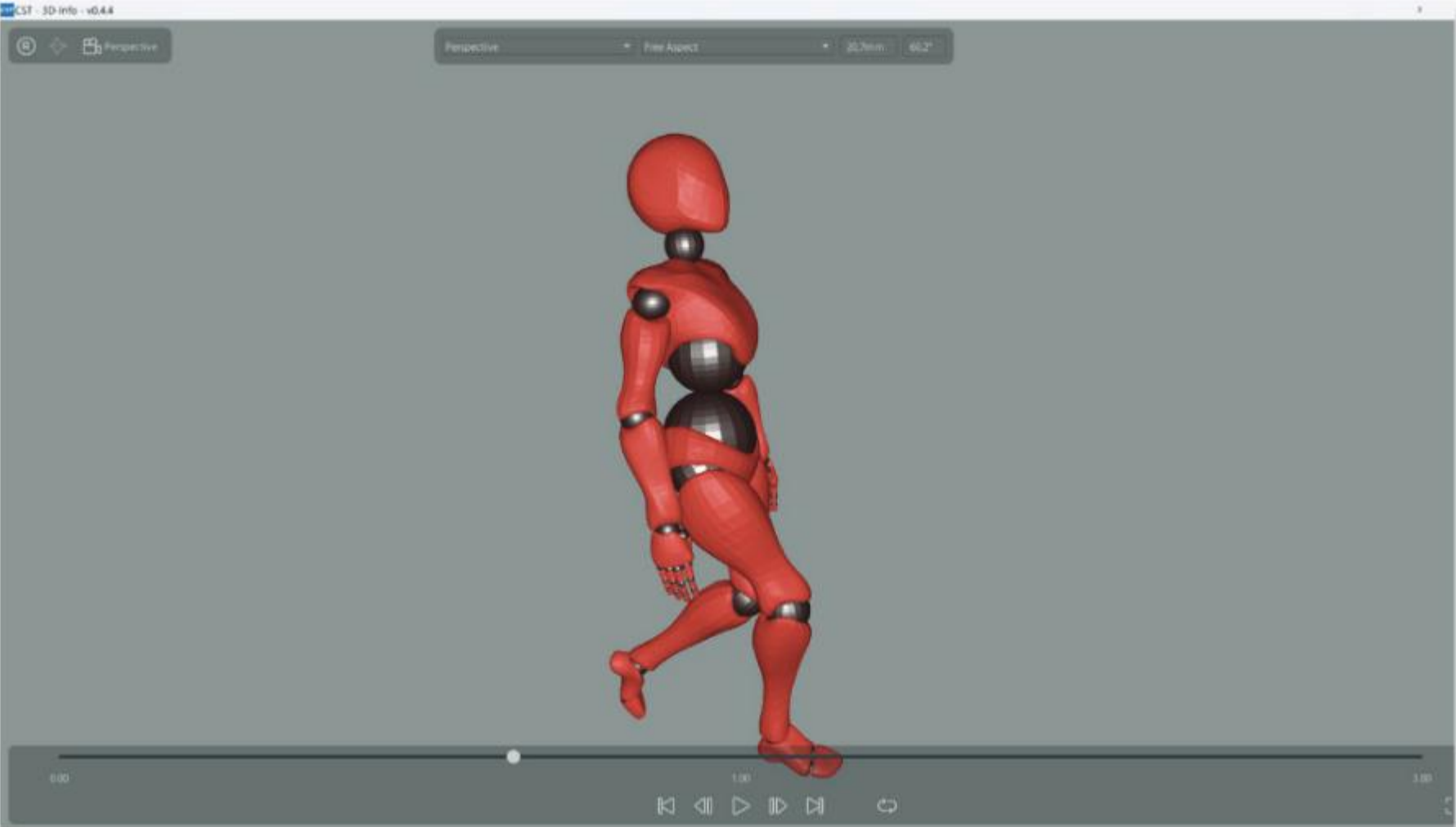
L'équipe développement de la CST

Équipe actuelle :

Hans-Nikolas Locher, directeur du développement,
 Lauralee Kamga, développeuse,
 Liam Zallio, développeur junior,
 Lys Yoyo, développeur junior.

Et la base de code de la CST est encore riche des contributions de nos anciens développeurs :
 Sébastien Bodin, Sami Amara,
 Widad Amara, Sasha Legrand.

*Hans-Nikolas Locher,
 directeur du développement*



UN CHOIX STRATÉGIQUE ET TECHNIQUE

ENTRETIEN AVEC VÉRONIQUE DEMILLY, CHARGÉE DE MISSIONS À LA DIRECTION DE L'IT TRANSVERSE (DITT) CHEZ FRANCE TÉLÉVISIONS ET ADMINISTRATRICE DE LA CST

France Télévisions a misé sur l'open source pour créer sa nouvelle infrastructure technique, visant l'adoption de technologies modernes et souveraines. Véronique Demilly est chargée de missions à la Direction de l'IT Transverse (DITT), et nous en dit un peu plus.

► **L'open source chez France TV est un choix stratégique et technique. De quoi s'agit-il ?**

VÉRONIQUE DEMILLY. Aujourd'hui, France Télévisions fait face à de nombreux défis d'origines diverses : incertitudes sur le financement de l'audiovisuel public exigeant une maîtrise renforcée des coûts, situation géopolitique nouvelle invitant à prendre garde aux positions hégémoniques des géants américains... En parallèle, l'accélération des mutations technologiques impose une adaptabilité toujours plus grande. Il s'agit d'ailleurs d'une évolution générale des directions informatiques des organisations : besoin de plus de flexibilité, de modularité, d'optimisation des coûts, d'automatisation, de résilience et de durabilité.

France Télévisions a donc entrepris une vaste modernisation de son infrastructure informatique avec un objectif clair : que la DSI passe d'une position de service support à celle de fournisseur de services pour les métiers. C'est pourquoi depuis quelques années, les équipes de la Direction de l'IT Transverse (DITT, c'est le nom de la DSI chez France Télévisions) développent une offre de service sur un cloud privé complètement open source avec Kubernetes¹.

L'idée n'est pas nouvelle. Dès 2017, il apparaît que les grands hyperscalers américains construisent leur offre Cloud à 90 % sur la base de composants open source maintenus par une fondation dont ils sont, bien sûr, membres.

1. <https://kubernetes.io/fr/>

► **Dans ces conditions, est-il possible pour une DSI d'intégrer ces mêmes composants dans une offre en propre ?**

V.D. Oui, bien sûr, mais il s'agit d'un véritable projet de transformation de l'organisation. Afin d'assurer les services avec le niveau de qualité requis chez un opérateur d'importance vitale comme France Télévisions, il faut commencer par investir dans l'humain : ce sont les compétences des ressources internes qui permettent de trouver, de tester, d'intégrer et de maintenir les composants open source pertinents. Il est nécessaire de disposer d'une équipe dédiée à l'open source. Elle doit en permanence surveiller l'évolution des composants, effectuer les tests nécessaires et, surtout, partager la connaissance entre ses membres. Chaque membre de l'équipe doit disposer d'une connaissance minimale sur chacun des composants déployés en interne. Le choix de l'open source s'inscrit dans une logique de gouvernance partagée : les équipes techniques, les métiers et parfois même les partenaires extérieurs participent à l'évolution et à l'amélioration continue des outils.

► **Qu'est-ce que le programme Alix ?**

V.D. La DITT a conçu un ensemble de produits permettant d'adresser chaque branche technique de l'IT. Les produits « Alix » sont composés de briques open source, cloud native, et pensés pour garantir la souveraineté.

Dans ce programme, le produit Alix Kube a pour ambition de proposer une offre d'hébergement cloud native portable sur tous les clouds. L'objectif est de créer une couche d'abstraction qui permet aux applications d'être déployées, gérées de manière unique sans adhérence avec le cloud sous-jacent. Avec Alix Kube, quelques clics suffisent pour migrer les applications d'un cloud à un autre.

Le produit Alix Eyes est un autre pilier de ce programme. Conçue comme une plateforme d'observabilité cloud native, Alix Eyes s'appuie sur des technologies open source pour agréger et centraliser les métriques, logs et traces émis par l'ensemble des composants techniques (infrastructure, applications, produits, équipements en régie, etc.). Déployée au sein d'Alix Kube, cette solution offre une visualisation unifiée de ces données et s'intègre directement aux outils d'exploitation, permettant ainsi une gestion optimisée des alarmes et de leur routage.



FTV. En contrepartie, Scaleway sera en première ligne pour développer les outils spécifiques aux besoins des médias, notamment pour déployer des offres de streaming.

Comme le souligne Delphine Ernotte Cunci, présidente-directrice générale de France Télévisions : « Ce partenariat avec Scaleway nous offre l'opportunité d'investir le terrain du cloud pour développer ensemble des solutions adaptées aux besoins de notre média. En s'associant à un acteur français de l'innovation, nous renforçons notre engagement en faveur de la souveraineté technologique, qui constitue aujourd'hui une priorité stratégique de notre groupe. »

Ainsi, pour chaque nouveau besoin exprimé par les métiers, les équipes de la DITT commencent par évaluer les solutions open source disponibles. Le choix de l'open source n'est pas un dogme : France Télévisions reste pragmatique. Les composants retenus proviennent de fondations solides construites autour des communautés pertinentes comme la Cloud Native Computing Foundation² (CNCF) qui est une émanation de la fondation Linux.

Néanmoins, France Télévisions n'a pas vocation à développer ses propres outils. Les équipes ne contribuent donc quasiment pas au code lui-même et se contentent de procéder à l'intégration des composants ou de remonter d'éventuels problèmes ou besoins complémentaires. C'est pourquoi des partenariats restent nécessaires.

► Comment avez-vous choisi votre partenaire sur l'hébergement, et quelle est son implication dans l'open source ?

V.D. Après avoir développé la couche d'abstraction Kubernetes permettant aux applications d'être agnostiques en termes de cloud, il restait à s'assurer qu'un fournisseur européen soit en mesure de les accueillir. France Télévisions a échangé avec plusieurs hébergeurs, puis a choisi de s'appuyer sur Scaleway, filiale du groupe Iliad. D'une part, Scaleway a complètement développé son offre cloud et en maîtrise tous les éléments : aucune dépendance à un autre acteur. D'autre part, Scaleway a démontré son intérêt pour le secteur audiovisuel. Toutefois, l'hébergeur ne disposait pas de la couche logicielle nécessaire pour instancier le cloud privé Kubernetes de France Télévisions. Cette couche, la cluster API, a donc été développée conjointement. Scaleway est donc désormais équipée pour héberger les différents services de

En tant qu'entreprise de l'audiovisuel public, France Télévisions souhaite partager ces réalisations le plus largement possible. Les nouveaux services que Scaleway pourra développer dans le cadre du partenariat seront disponibles pour d'autres acteurs médias. Plus largement, il serait envisageable de dupliquer cette infrastructure chez d'autres hébergeurs européens pour le bénéfice des acteurs médias du vieux continent. Des échanges sont d'ores et déjà organisés au sein des groupes de travail des Technologies de l'Audiovisuel Public (TAP) et de l'Union Européenne de Radio-télévision (UER).

► Contribution à l'écosystème.

V.D. Au-delà de la technique, France Télévisions contribue ainsi à faire émerger un écosystème ouvert, souverain et collaboratif au service de l'audiovisuel public. L'ambition est claire : faire de l'open source un levier d'innovation partagée, garantissant à la fois efficacité, sécurité et indépendance.

Propos recueillis par Frédéric Fermon



2. <https://www.cncf.io/>

L'OPEN SOURCE DANS LES ICC : UNE DÉMARCHE INNOVANTE ET PRAGMATIQUE

Marc Bourhis est Community Leader ICC & Program Manager à Cap Digital. Il nous livre sa vision sur la place de l'open source dans les ICC (industries culturelles et créatives).

Le développement de logiciels open source pour la création et la diffusion de contenus est aujourd'hui un catalyseur d'innovation et un facteur de démocratisation technologique. Il permet de diffuser largement des outils performants, tout en répondant aux besoins de niches comme les collectifs artistiques indépendants, les petits studios de création ou les chercheurs, qui souhaitent des solutions peu coûteuses, modulables et interopérables. L'open source facilite aussi la mise en place de workflows complexes où la compatibilité entre solutions propriétaires devient essentielle pour fluidifier la production et la distribution de contenus audiovisuels. Toutefois, il constitue un levier structurant qu'il convient d'aborder avec pragmatisme, sans surestimer ses vertus ni négliger ses contraintes économiques et industrielles.

UNE LOGIQUE COMMUNAUTAIRE À L'ORIGINE DE L'OPEN SOURCE

Blender est un projet open source emblématique au sein de la communauté des ICC. Ce logiciel 3D, créé par Ton Roosendaal pour son studio NeoGeo, a successivement été un produit propriétaire au sein de Not a Number (NaN) puis, après la faillite de l'entreprise, a été racheté par sa communauté d'utilisateurs en 2002 grâce à un financement participatif et placé sous licence libre par la Blender Foundation, amorçant ainsi son essor mondial. Aujourd'hui, fort de 14 millions de téléchargements annuels, d'une trentaine de développeurs salariés et de centaines de contributeurs bénévoles, Blender s'impose comme une alternative crédible à des logiciels leaders comme Maya et séduit de nombreux studios indépendants.

Cet essor est cependant freiné par un manque de clients prescripteurs et par des ressources limitées retardant certaines fonctionnalités. Le cas

Blender illustre aussi les contraintes des licences libres selon leur niveau d'ouverture. En optant pour la GPL, Blender impose un partage obligatoire du code des plug-in utilisant son API, ce qui limite l'intégration de logiciels tiers malgré la dynamique communautaire.

LES DÉFIS DE LA CONSOLIDATION INDUSTRIELLE

La principale limite de l'open source réside dans la difficulté à construire un modèle économique viable, faute de revenus directs si l'outil est gratuit. Nombre de projets reposent sur de petits noyaux de bénévoles ou les ressources internes d'un unique studio de création, ce qui fragilise leur viabilité, sauf s'ils sont intégrés à un modèle d'affaires plus large. Le passage à l'échelle industrielle suppose en général de trouver des solutions de monétisation pérennes via du mécénat d'entreprise et services associés.

Même des projets emblématiques comme Blender ont connu un parcours ardu depuis 2002. La Blender Foundation a pu croître grâce à des financements publics, au soutien de géants du cinéma ou du jeu (Epic Games, Ubisoft) et aux dons massifs de la communauté, sans lesquels il n'aurait pas atteint son statut actuel de pilier mondial de la 3D.

Contrairement aux start-up, un projet open source suit rarement une croissance rapide. Le développement économique s'appuie souvent sur de grandes entreprises en quête d'interopérabilité, sur des studios recourant à des outils maison sans vouloir devenir éditeurs, ou sur des PME misant sur des services additionnels. VideoLan, derrière le player vidéo VLC (six milliards de téléchargements) illustre d'ailleurs parfaitement ce dernier modèle avec une constellation de services qui créent de la valeur (sans jamais trahir l'ADN open source). Meshroom, développé par AliceVision avec le soutien de fonds publics et du studio MPC, montre également l'impact d'un studio de création de VFX lorsqu'il est le premier développeur et utilisateur de l'outil. Enfin, la plateforme Kitsu (CGWire) combine open source (licence AGPL) et commercialisation par abonnement, complétée par des services sur mesure et des licences abordables pour les écoles.





© Photo : KITSU

▲ Le stand de CGWire qui édite Kitsu au MIFA Annecy 2025.

Un fournisseur privé de solution open source doit souvent arbitrer entre les briques techniques partagées avec la communauté et celles conservées propriétaires, adoptant un modèle hybride.

C'est encore plus compliqué dans le domaine de l'IA générative, qui ne peut réellement se revendiquer « open source » et garantir une transparence de bout en bout, des données d'entraînement à l'exploitation commerciale. Un des verrous tient à la « secret sauce » des modèles – granularité, choix algorithmiques, sélection des datasets – vue comme un avantage compétitif et donc rarement publiée.

DES ENJEUX TECHNIQUES, ÉTHIQUES ET DE SOUVERAINETÉ

Si les logiciels open source stimulent de nouvelles pratiques, ils posent aussi des défis liés à la fragmentation des outils, la disparité des interfaces et à la qualité variable de leur documentation, ce qui peut entraver leur adoption dans des workflows industriels exigeants ou impliquant une multiplicité d'acteurs différents, ce qui est souvent le cas dans les industries culturelles et créatives (ICC).

Pour répondre à ces enjeux, Cap Digital s'est investi aux côtés de grands acteurs comme France Télévisions, l'INA, l'ISAN, la Procirep et la CST autour de TEMS (Trusted European Media data Space), un espace de données de confiance fondé sur des briques open source. Lancé sous l'égide de la Commission européenne avec 43 partenaires dans 11 pays, TEMS vise à bâtir une infrastructure technique sécurisée, éthique et interopérable pour les médias et la culture, favorisant la découvrabilité des contenus audiovisuels et la gestion des droits via des smart contracts standardisés.

L'OPEN SOURCE, UN PAS VERS LA SOUVERAINETÉ

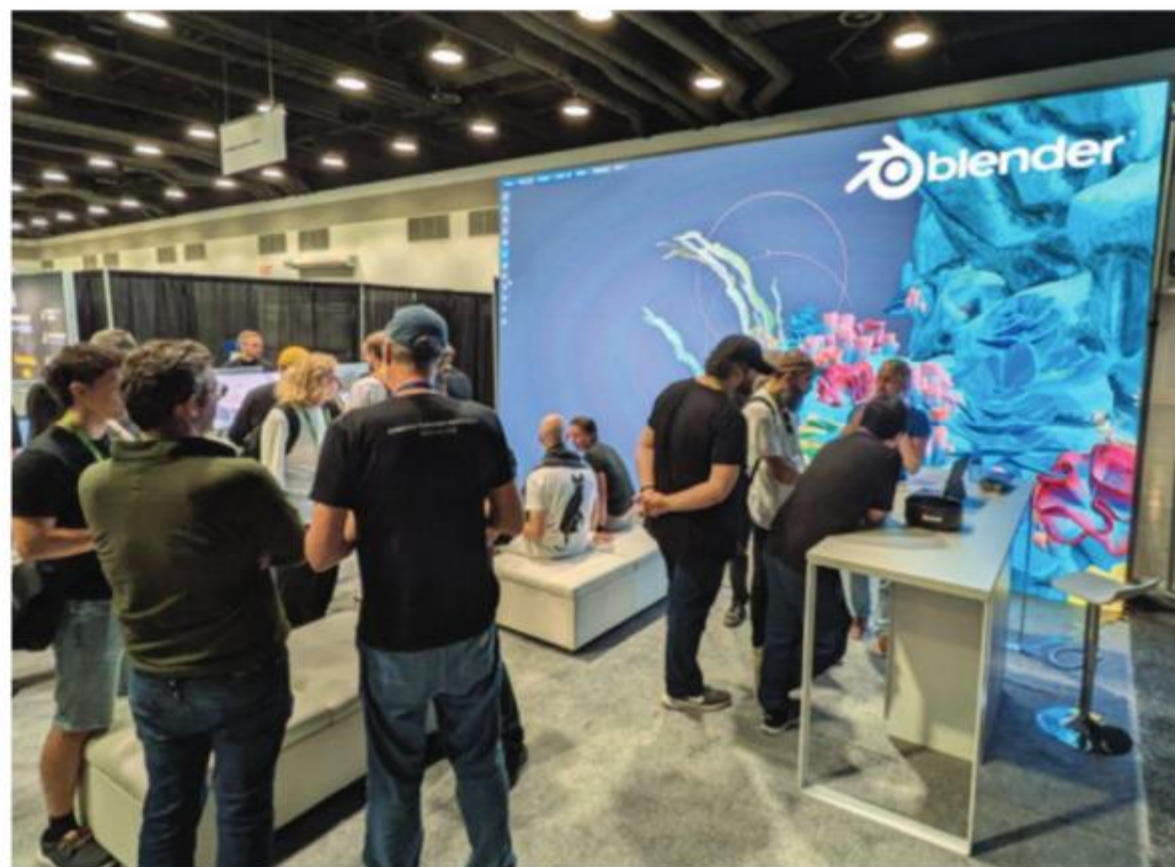
Au-delà des espaces de données et de la transparence des IA, la stratégie européenne de financement de l'innovation vise la souveraineté numérique, avec des investissements ciblés sur l'interopérabilité, la portabilité des données et l'open source, afin de limiter la dépendance à des fournisseurs non européens.

Dans cette transition vers des outils numériques plus souverains, les réseaux collaboratifs comme Cap Digital jouent un rôle clé en créant les conditions d'une innovation réussie et en facilitant les collaborations entre entreprises, chercheurs et communautés open source. L'initiative de la CST autour de ce hors-série, qui met en lumière l'usage des logiciels libres dans le cinéma et l'audiovisuel, illustre cette dynamique.

Un pôle de compétitivité comme Cap Digital joue aussi un rôle d'interface et de catalyseur en facilitant les collaborations entre entreprises, institutions académiques et communautés open source. Pour assurer la pérennité de l'open source dans les ICC et soutenir la croissance de la filière cinéma et audiovisuelle, il est impératif de conjuguer pragmatisme et ambition, de multiplier les partenariats entre secteurs publics et privés et de s'appuyer sur des gouvernances ouvertes. Seule une approche collective, coordonnée et stratégique nous permettra de relever les défis de l'innovation, de la compétitivité et de la souveraineté numérique.

**Marc Bourhis, Community Leader ICC
& Program Manager à Cap Digital**

▼ La communauté Blender se retrouve sur l'exhibition floor du Siggraph 2025.



© Photo : Siggraph

VLC ET L'OPEN SOURCE ONT BOULEVERSE L'INDUSTRIE VIDEO : OÙ VA LA VALEUR ?

ENTRETIEN AVEC JEAN-BAPTISTE KEMPF, FONDATEUR ET CEO DE VIDEO LABS

Développeur historique de VLC (lecteur de fichiers multimédias multi-plateforme), Jean-Baptiste Kempf est président de l'association VideoLAN, ainsi que le fondateur et CEO de VideoLabs. Il nous éclaire ici sur la stratégie de VideoLAN et la transformation des pratiques professionnelles dans le secteur vidéo.

Site du projet : <https://www.videolan.org>

Dépôt du code :

<https://code.videolan.org/videolan/vlc>

► En deux mots, que s'est-il passé sur le marché du logiciel avec l'open source ?

JEAN-BAPTISTE KEMPF. On a vécu une inversion complète de paradigme. Avant, certains développeurs s'investissaient dans des projets open source pour partager leur passion. Aujourd'hui, c'est l'inverse : il faut justifier pourquoi ne pas choisir l'open source. Hormis quelques applications métiers ultra-niches ou qui reposent sur leurs assets historiques (Microsoft, Apple, Adobe), la plupart des frameworks et des outils vont vers le libre – et l'IA accélère cette commoditisation.

► Quel impact spécifique cela a-t-il dans l'industrie vidéo ?

J.-B.K. On assiste à l'effondrement de la valeur des architectures logicielles « traditionnelles ». Des acteurs vidéo historiques peinent à rester rentables, avec des marges très contraintes ; Harmonic fait toujours un gros chiffre d'affaires mais a échoué à se vendre ; les grandes start-up de la vidéo ont vu leur trajectoire s'interrompre, alors même qu'elles avaient levé des sommes importantes. Résultat : les efforts de R&D des acteurs traditionnels de la vidéo ont été drastiquement réduits.

► Où se crée encore la valeur ?

J.-B.K. Trois segments demeurent robustes : des acteurs comme Dolby qui monétisent leurs brevets ; les clouds providers et les acteurs de CDN qui

« offrent » l'encodage pour vendre la distribution ; et les plateformes de contenus qui contrôlent l'accès aux audiences. La valeur commerciale s'est globalement déplacée vers les brevets, la distribution et le contenu. Donc, pour le reste, la pression sur les prix devient capitale.

► Comment expliquer alors cet engouement pour l'open source ?

J.-B.K. Le prix bien sûr, mais surtout la liberté, l'évitement du lock-in : le verrouillage avec un éditeur de logiciel fermé. Après des années de mauvaises expériences, les acheteurs privilégient des solutions ouvertes ou semi-ouvertes. L'exemple de DaVinci Resolve (solution d'étalonnage et de montage avec intégration de VFX) qui propose une version gratuite est emblématique : la valeur se déplace vers l'intégration des outils logiciels dans les systèmes de production. Cette intégration se faisant en interne ou via des consultants spécialisés.

► Peut-on revenir sur le choix de la licence de VLC et le modèle commercial associé ?

J.-B.K. VLC est resté sur son modèle « ancienne école » avec la licence GPL pour l'application utilisateur. L'association VideoLAN est méritocratique, sans commercialisation directe du produit, même si elle reçoit bien sûr des dons. L'écosystème, basé autour de VideoLabs, permet d'assurer le support et l'intégration. Ainsi, autour de VLC et FFmpeg, trois ou quatre sociétés prennent en charge le support ; la pérennité du support et de la maintenance de l'outil est garantie, même si l'une de ces sociétés disparaît. Autour de la communauté, il existe aussi d'autres sociétés qui ont une approche du type « double licence » : une licence d'entrée de gamme gratuite et une licence professionnelle commerciale, avec par exemple un ou deux modules supplémentaires, et surtout du support, du consulting ou de la résolution de bugs. Un des exemples de ce nouveau genre de société (double-licence) est le nouveau projet que je porte sur la très basse latence, appelée Kyber.

► Et par rapport aux développeurs, comment vivent-ils ce choix de licence ?

J.-B.K. Les licences MIT ou Apache prennent aujourd'hui le dessus : elles n'imposent aucune contrainte pour embarquer ces logiciels dans



des produits commerciaux. Dans la pratique, les équipes de développeurs contribuent au pot commun, pour éviter des forks (version dérivée indépendante) qui seraient plus coûteux à maintenir, à long terme.

► Blender fait-il encore exception ?

J.-B.K. Oui, le choix de la licence GPL stricte et son interprétation communautaire imposant le GPL aux scripts Python freinent le développement des pipelines mixtes. C'est un choix assumé, au-delà des exigences minimales de la licence.

► Comment met-on en place la gouvernance d'un projet open source ?

J.-B.K. Méritocratie et exigence technique sont nos deux valeurs de référence. Le noyau de VideoLAN, c'est une dizaine de spécialistes, des ingénieurs multimédia pointus. Avec Gitlab qui reste notre réseau d'échange privilégié. Il faut aussi un leadership fort du type BDFL (Benevolent Dictator For Life), avec également des forums de discussion, des remontées de bugs et surtout des événements, qui sont des moments importants pour notre communauté. On voit d'ailleurs au cours de ces événements que les communautés VLC et FFmpeg se recoupent largement.

► Pourquoi certains projets open source disparaissent-ils ?

J.-B.K. La survie d'un projet open source se fait de manière assez naturelle : si les performances du logiciel ne sont pas bonnes ou s'il n'y a pas un nombre suffisant d'utilisateurs, le projet meurt – mais le code peut toujours être forké et repris – contrairement aux solutions propriétaires où l'on perd tout à la fermeture de la société qui développait ces logiciels, comme on l'a vu encore récemment dans le domaine des VFX. Le mode de sélection « darwinien » de l'open source incite à viser l'excellence pour survivre.

► Quelles relations entretenez-vous avec les acteurs traditionnels, en particulier les broadcasters ?

J.-B.K. Elles sont complexes. Certains attendent un service « VIP » au seul motif de leur assise très forte sur le marché ; mais on leur rappelle simplement que l'open source ne leur « doit » rien. Et surtout, l'industrie va globalement vers plus de standardisation et moins de solutions customs. Il est notable de remarquer que les acteurs publics – les diffuseurs en Europe autour de l'EBU notamment – cofinancent souvent des développements ciblés, fondés sur des besoins et des standards ouverts, qui sont ensuite reversés en amont dans l'ensemble de la chaîne de production. Nous ne cherchons pas spécifiquement, pour notre part, à intégrer des formats propriétaires ou exotiques comme Dolby-E, P2 ou AVC-Intra, mais nous accueillons volontiers des contributions pour supporter ces formats. Notre rôle dans l'écosystème est aussi pédagogique, tout en restant pragmatique.

► Comment se profile l'avenir pour VideoLAN ?

J.-B.K. Il suffit juste de constater que VLC règne en maître absolu au niveau des lecteurs multimédias : 500 millions de téléchargements pour la dernière version (la 3.0.21), avec un moteur qui est repris partout. Les projets x264, dav1d, dav2d, x265 ou libspatialaudio, qui sont tous portés par VideoLAN, sont utilisés énormément, au-delà de VLC. Désormais, l'usage de FFmpeg et VLC va de soi dans tous les flux média. Même les poids lourds du logiciel propriétaire évoluent : Adobe mise sur un plug-in OpenUSD, Apple publie certains de ses modèles d'IA. Aujourd'hui, beaucoup d'éditeurs de logiciels adoptent le modèle SaaS (Software as a Service, déployé sur le web) et s'appuient de manière écrasante sur des briques open source. La valeur différenciante se situe de plus en plus dans la couche métier bâtie au-dessus d'un socle commun et ouvert.

Propos recueillis par Benoît Maujean



© Photo : VLC

KITSU, UNE PLATEFORME AU SERVICE DES CRÉATEURS

Site du projet : <https://www.cg-wire.com/kitsu>

Dépôt du code : <https://github.com/cgwire/kitsu>

Kitsu est une plateforme de collaboration pour les studios créatifs (animation, VFX, jeux vidéo). Elle centralise un référentiel commun où artistes, superviseurs, réalisateurs, managers, producteurs et clients travaillent de concert. Frank Rousseau, fondateur et directeur des opérations de CG-Wire, nous en dit plus.

PRÉSENTATION DU PROJET KITSU

Avec Kitsu, on construit plannings et budgets, on répartit les tâches, puis les équipes partagent leur avancement et déposent leurs livrables (vidéos, images ou modèles 3D). Réalisateurs et clients y saisissent directement leurs retours, accélérant les itérations avec les artistes. Les données permettent au contrôle de gestion de suivre les dépenses, tandis que la direction bénéficie d'une vision d'ensemble de chaque projet et du studio. Résultat : des flux plus fluides, des problèmes anticipés, une qualité accrue et une information toujours disponible, donc un studio plus productif.

(► Visuel pages 30 et 31).

CHOIX DE L'OPEN SOURCE (LICENCE ET GOUVERNANCE)

Nous avons choisi l'open source dès le départ par conviction : instaurer la confiance par la transparence entre éditeur logiciel et utilisateurs. Les solutions propriétaires tendent à enfermer leurs clients (évolution des coûts, CGU, corrections de bugs). Sans en faire un argument marketing central, nous avons vite constaté l'avantage compétitif du modèle : attractivité du recrutement, participation des utilisateurs, préférence des développeurs pour les outils ouverts, et rigueur accrue dans nos pratiques. Surtout, l'ouverture a permis de convaincre nos premiers clients de financer des fonctionnalités initiales.

Côté licence, nous avons opté pour l'AGPL, afin d'éviter qu'un tiers ne reprenne le code et ne l'enferme sous sa marque. Avec l'AGPL, la redistribution est possible mais le code dérivé doit rester AGPL ; cela décourage les appropriations fermées et simplifie l'accueil des contributions (pas de paperasse lourde).

La gouvernance est portée par CGWire : nous définissons la roadmap et assurons l'avancement. Les développements prioritaires sont ceux qui sont financés (clients ou subventions), car un financement signale un besoin réel.

CONSTRUIRE ET MAINTENIR LA CONFIANCE

Un ancien chef me disait chez HD3D : le premier facteur d'échec d'un projet, c'est le manque de confiance. J'ai donc misé sur mon profil hybride (expérience sectorielle et CTO en startup). Au début, on m'a perçu surtout comme développeur, rassurant sur la capacité à livrer vite. J'ai interrogé une dizaine de studios sur leurs besoins. C'est là que TNZPV, Les Fées Spéciales et Cube Creative (puis Miyu et Folivari) ont adopté très tôt Kitsu. Sans l'open source, leur engagement aurait sans doute été plus long à obtenir, et cela lui a conféré une certaine légitimité.

Conscient de mes limites côté production, j'ai proposé à Gwénaëlle Dupré (alors directrice des Productions chez Unit Image) de nous rejoindre. Sa maîtrise des pipelines, son rôle de Product Owner lors de nos sprints et sa connaissance des trackers (mise en place documentée de Shotgrid) ont apporté la dimension métier qui nous manquait.

Nous avons alors structuré une offre d'hébergement avec support de haut niveau. Après quelques installations, nous avons sollicité un DevOps expérimenté, Nicolas Ledez, pour consolider l'infrastructure, avant de le recruter. Le triptyque « référence secteur + expertise métier + hébergement solide » a rassuré nos partenaires. Le fait d'être open source offrait en plus une porte de sortie (auto-hébergement) en cas de défaillance de notre part.

J'animais des meetups pipeline qui ont favorisé les rencontres entre utilisateurs. Une première communauté a ensuite joué les ambassadeurs. Après une visite à la Blender Foundation et une discussion avec Ton Rosendal, nous avons clarifié notre mission : au-delà de simplifier le quotidien des équipes, il s'agissait de permettre à n'importe quel studio d'accéder à des outils de haut niveau pour gérer leurs projets. J'ai formalisé cet objectif sur notre blog ; la communauté l'a vite adopté et s'est impliquée davantage : support sur le serveur public,



© Photo : KITSU

débats d'amélioration, tutoriels, billets de blog, partage d'outils autour de KitsU sur Github. Notre cadence de livraison a entretenu la confiance : ce qui est mis en place aujourd'hui progresse demain.

MODÈLE ÉCONOMIQUE

Nous accompagnons une application centrale et ambitieuse, qui exige un déploiement et une exploitation rigoureux. D'où une offre adossée à un support expert et à un hébergement robuste, pilotés par notre équipe cloud. Notre chiffre d'affaires repose sur trois offres :

- installation en Cloud sécurisé + support utilisateur expert, en abonnement mensuel ou annuel par utilisateur ;
- installation sécurisée sur les serveurs du studio + support expert, en abonnement mensuel ou annuel par utilisateur + coût de setup ;
- des développements spécifiques pour accélérer l'arrivée de fonctionnalités (facturation au forfait/jour).

Ce modèle nous permet d'offrir une solution attractive tout en finançant le développement.

CONFIANCE ET CONTRIBUTIONS DES DÉVELOPPEURS EXTERNES

Nous avons toujours eu beaucoup de contributions externes en « jus de cerveau » : demandes de fonctionnalités, rapports de bugs, réflexions. Les contributions en code sont venues plus tard, souvent de

studios auto-hébergés qui nous ont partagé leurs personnalisations, ou de développeurs motivés par le projet qui piochent dans nos tickets Github. Les contributions spontanées sont intégrées au plus vite car nous voulons éviter toute frustration liée à des délais superflus.

BÉNÉFICES CONCRETS DE L'OPEN SOURCE

- **Sérénité** : nous avons offert un bien commun, utilisé par 300 à 400 studios (environ 15 000 professionnels) qui continuerait d'exister même dans un scénario catastrophe ;
- **Accélération** : de nouveaux studios se structurent vite grâce à KitsU malgré de faibles budgets ;
- **Démarrage et alliances** : l'ouverture a facilité le lancement du projet et l'accès à des partenaires de qualité (Blender Studio, CNC, Epic Games) ;
- **Souveraineté** : les studios restent maîtres de leurs données, un avantage clé à l'heure des enjeux de souveraineté numérique.

RÉFLEXIONS ACTUELLES

La gratuité de l'auto-hébergement pose problème : sécurité perfectible, fiabilité non garantie, support peu réactif. Beaucoup de studios s'en contentent, avec deux effets négatifs pour nous : manque à gagner et expériences dégradées qui ternissent notre image auprès d'utilisateurs qui ignorent notre non-implication. Paradoxalement, ces studios économiseraient davantage en passant par nos services. Nous devons donc affiner notre discours marketing, renforcer nos offres cloud et on-prem afin qu'il n'y ait plus d'hésitation pour ces studios à prendre un abonnement. Un doublement voire triplement du CA est envisageable si nous convainquons ces structures.

Autre enjeu : accroître les contributions code pour pérenniser le projet. Nous venons de mettre en place un système de plugins afin de faciliter les contributions sans toucher au cœur de l'application, et offrir une personnalisation plus aisée. KitsU peut ainsi couvrir un éventail de cas d'usage toujours plus large.

*Propos recueillis par
Frédéric Fermon et Benoît Maujean*

PROJET KITSU



Wing It



Shots



ex: e01 s01 anim=wip



All Departments



Display Settings



Compositing



Lighting



Sequences



Tags



Task Done



animation not finished



animation urgent not done

NAME



TIME

EST.

FRAMES

IN

OUT

Animatik

Layout

Animation

020

☐		ID	TIME	EST.	FRAMES	IN	OUT	Animatik	Layout	Animation
☐		020_0010	96	96	77	854	931	RCVD RL	DONE	DONE
☐		020_0015	80	104	62	931	993	RCVD RL	DONE	DONE
☐		020_0020	24	88	43	993	1036	RCVD RL	DONE	NA
☐		020_0030	96	96	78	1036	1114	RCVD RL	DONE	DONE
☐		020_0040	24	72	42	1114	1156	RCVD RL	DONE	NA
☐		020_0050	64	80	53	1156	1209	RCVD RL	DONE	DONE
☐		020_0060	32	88	44	1209	1253	RCVD RL	DONE	DONE
☐		020_0070	32	80	47	1253	1300	RCVD RL	DONE	DONE
☐		020_0080	32	88	36	1168	1204	RCVD RL	DONE	NA
☐		020_0090	24	96	63	1300	1363	RCVD RL	DONE	DONE
☐		020_0100	56	88	51	1363	1414	RCVD RL	DONE	DONE
☐		020_0110	96	104	80	1414	1494	RCVD RL	DONE	DONE
☐		020_0120	40	80	41	1494	1535	RCVD RL	DONE	DONE
☐		020_0130	152	128	138	1535	1673	RCVD RL	DONE	DONE
☐		020_0140	40	80	39	1673	1712	RCVD RL	DONE	DONE

60 shots - 4367 frames (4 371,75 days spent, 6 450,08 person days)



Q ctrl+alt+f

lighting urgent master [Compositing]=[Ready]

FX Matte Painting Lighting

NA	NA	WFA CF	TC
NA	NA	DONE CF	RE
NA	NA	DONE	RE
NA	NA	WFA	TC
NA	NA	RETAKE	TC
NA	NA	WFA	TC
NA	NA	DONE	RE
NA	NA	WFA	TC
NA	NA	DONE	RE
NA	NA	DONE	RE
NA	NA	DONE	RE
NA	NA	DONE	RE
NA	NA	DONE MA	RE
DONE	NA	WFA MA	TC
NA	NA	DONE MA	RE

STATUS

Lighting 020 / 020_0050



POST COMMENT PUBLISH REVISION

Leave a comment...

WFA

Post

WFA Gwenaelle Dupré 06/04

0

Reply

REVISION 3

RETAKE Gwenaelle Dupré 06/04

Can you add a rim light?

0

Reply

WFA Gwenaelle Dupré 06/04

0

Reply

REVISION 2

RETAKE Gwenaelle Dupré 06/04

L'INNOVATION EN MOUVEMENT PERPÉTUEL

ENTRETIEN AVEC FABIEN CASTAN, RESPONSABLE R&D CHEZ MPC ET THE MILL

Meshroom est une boîte à outils flexible pour l'automatisation qui permet de concevoir, organiser et exécuter des pipelines de traitement de données complexes. AliceVision est un framework de vision par ordinateur et d'IA qui fournit des pipelines Meshroom, notamment en photogrammétrie et en suivi de caméra. Responsable R&D chez MPC et The Mill, Fabien Castan supervise les projets open source Meshroom et AliceVision. Pour la CST, il revient sur la genèse et les objectifs de ce projet.

Page de Meshroom/AliceVision :

<https://alicevision.org/association/>

Repo : <https://github.com/alicevision/Meshroom>

► Pouvez-vous présenter ce projet ? Pour quels usages et utilisateurs finaux a-t-il été créé ?

FABIEN CASTAN. Le projet est né en 2010 des besoins des superviseurs VFX et des infographistes qui ont besoin de truquer des images tournées en y intégrant des éléments de synthèse (CG). L'enjeu est de capturer toutes les informations du plateau de tournage pour récupérer un double numérique en 3D ainsi que des informations précises de l'éclairage, puis d'analyser les images tournées pour y intégrer les éléments CG de manière photo-réaliste.

Meshroom permet de faire la reconstruction 3D du lieu de tournage (appelée photogrammétrie), d'assembler des panoramas 360° HDR pour capturer l'éclairage de la scène, mais aussi de suivre les mouvements de la caméra de prises de vue.

L'originalité de l'outil est de proposer un système configurable pour permettre d'adapter le workflow d'interprétation 3D au contexte de l'acquisition, à la qualité et au type des capteurs utilisés : caméras professionnelles ou embarquées, smartphones ou rigs multi-caméras. Les types de prises de vues peuvent être variés : multi-éclairages, multi-bracketings, macro-photographies ou images satellites... Les utilisateurs sont de différents profils : des utilisateurs ordinaires, des experts qui savent modifier les pipelines dans l'interface nodale, et des dé-

veloppeurs qui peuvent créer de nouveaux plugins ou modifier le code existant.

Les technologies permettant de réaliser cela sont la vision par ordinateur (Computer Vision) et l'Intelligence Artificielle (IA), déployées grâce à des partenariats académiques et industriels, pour à la fois bénéficier des dernières avancées de la recherche mais aussi développer et tester le logiciel dans des contextes de production. D'abord avec une thèse CIFRE entre Mikros Image et le laboratoire Imagine des Ponts et Chaussées, puis avec des projets de recherche européens impliquant l'IRIT à Toulouse, CTU à Prague et Simula à Oslo.

Cette démarche collaborative nous a d'emblée conduits à adopter une approche open source, afin de faciliter les transferts de propriété intellectuelle et l'intégration de nouveaux partenaires. Cette collaboration s'est ensuite consolidée avec les équipes de R&D de MPC et The Mill, tout en continuant à maintenir les relations avec les laboratoires de recherche.

Grâce à ces différents partenariats et à la visibilité croissante du projet, le périmètre des utilisateurs s'est élargi. En effet, ces technologies dépassent largement le cadre des besoins du VFX et trouvent également des applications en géologie, dans la préservation du patrimoine, le domaine médical, le génie civil et le bâtiment, l'essayage virtuel...

En janvier 2025, le Prix de l'Innovation des César & Techniques a récompensé MPC pour le développement du Camera Tracking dans Meshroom.

L'application pour Linux et Windows est téléchargée en moyenne 600 fois par jour¹, et plus de deux millions de fois depuis le début du projet.

► Quelle licence avez-vous choisie ? Pourquoi ?

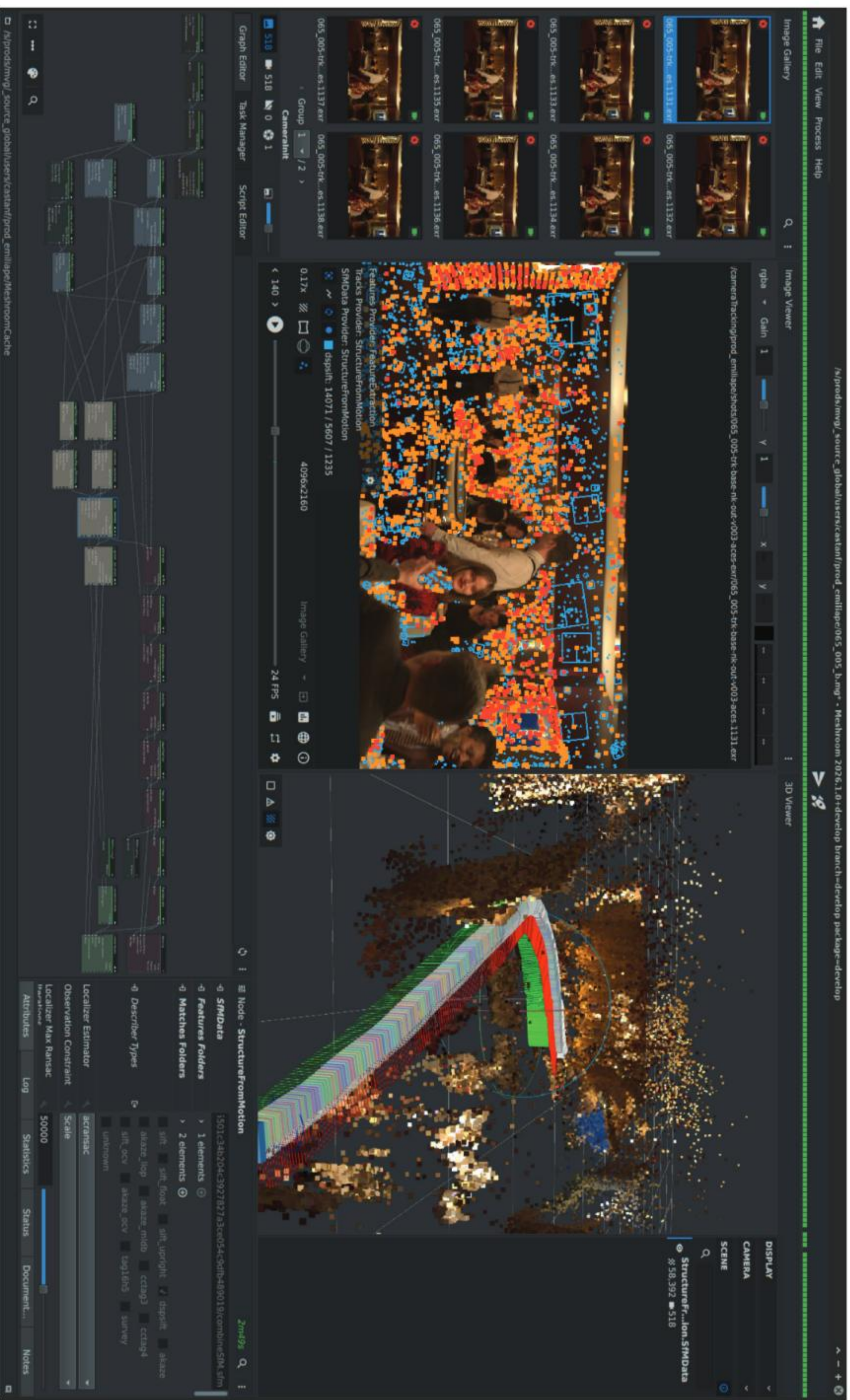
F.C. Nous avons choisi la licence MPL V2 (Mozilla Public Licence²) car elle représente un bon compromis entre les licences permissives et les licences virales. On peut intégrer facilement Meshroom et AliceVision dans des pipelines de production et des outils commerciaux (code fermé), mais les modifications du code doivent garder la même licence.

1. <https://alicevision.org>

2. <https://www.mozilla.org/en-US/MPL/2.0/>

TECHNIQUE

◀ Pipeline de suivi de caméra 3D pour le film *Emilia Perez* de Jacques Audiard.



► Que pouvez-vous nous dire sur la gouvernance du projet ?

F.C. Nous avons créé à l'été 2020 une association loi 1901³ pour gérer le projet (dépôt du nom, gestion du dépôt de code logiciel), animer la communauté des partenaires industriels et académiques (18 membres) et définir les grandes orientations du projet.

► Sur quel modèle d'affaires éventuel vous basez-vous ?

F.C. L'association AliceVision ne développe pas de modèle commercial autour du projet, mais reçoit des dons (268 donateurs uniques pour un total d'environ 50 k€ en 5 ans), dont notamment un MegaGrant d'Epic Games. Nous avons aussi réalisé des prestations de développement, pour le compte d'acteurs industriels ou institutionnels. Le projet a bénéficié de financements nationaux et européens : Labcom (ALICIA VISION) avec l'IRIT et deux projets européens Innovation Action (POPART et LADIO).

Le projet bénéficie aussi du support des programmes d'aides aux projets open source par la gratuité de services en ligne comme Github⁴ (hébergement du code et intégration continue linux/windows), Google workspace (gestion bureautique), Slack (discussions entre partenaires), Dockerhub (hébergement des images docker), Deepwiki (documentation technique) et Zenodo (hébergement des binaires).

► Quelles sont vos relations avec les utilisateurs, développeurs, contributeurs externes ? Comment avez-vous instauré et maintenu la confiance de la communauté ?

F.C. Plus d'une trentaine de développeurs ont contribué à Meshroom. Notre forum de discussions

permet d'échanger avec les utilisateurs et les développeurs : plus de 500 membres qui échangent sur les fonctionnalités, les bugs, etc.

Nous maintenons des partenariats avec les écoles et universités (l'ENSEEIH, ATI-Paris 8 et l'IMAC). Cela nous permet d'analyser la perception du projet via leur regard neuf, faciliter son accessibilité pour de nouveaux utilisateurs et les aider à s'emparer de l'outil dans leur cursus universitaire ou professionnel.

Meshroom a su s'adapter aux évolutions des besoins des utilisateurs : initialement conçu comme une interface pour AliceVision, il est désormais une plateforme nodale de programmation visuelle, permettant de concevoir et d'exécuter des workflows avancés pour le traitement de données complexes. Le nouveau système de plugins simplifie la création de nœuds et accélère le processus de développement avec une approche ouverte et collaborative autour de MeshroomHub⁵ : cette plateforme permet déjà d'intégrer les nouvelles approches comme le Gaussian Splatting, l'estimation de la profondeur monoculaire et d'autres fonctionnalités exploratoires.

► Plus globalement, quels enseignements en avez-vous tiré ?

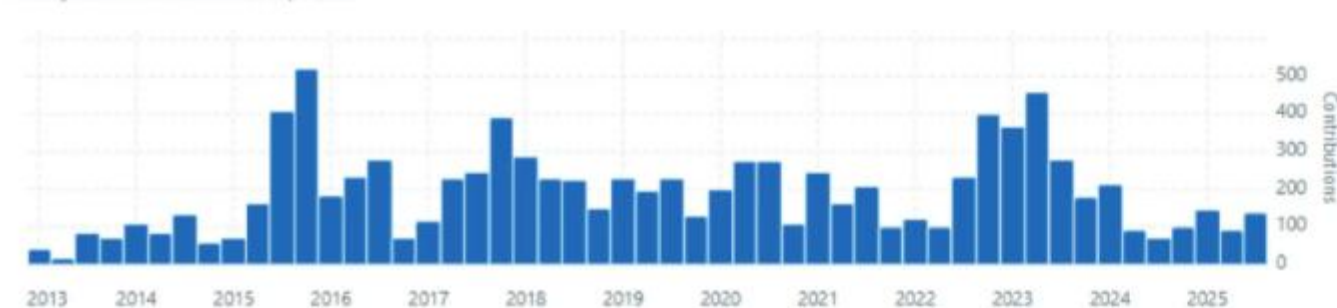
F.C. Depuis 2010, l'aventure Meshroom confirme une évidence : la pérennité d'un projet repose sur un code ouvert, partagé et éprouvé par le temps. Porté par une communauté diverse – développeurs de studios, chercheurs ou contributeurs indépendants – le projet a su transformer la multiplicité des usages en une intelligence collective. Cela a permis d'intégrer l'évolution, mais aussi les ruptures technologiques (nouveaux capteurs, nouveaux algorithmes de vision par ordinateur, couplée avec l'IA analytique) sans fragiliser l'existant. L'open source dépasse le simple mode

de distribution logicielle : il constitue le socle d'une démarche durable, d'une innovation continue et d'un savoir partagé au sein d'une large communauté.

Propos recueillis par Benoît Maujean

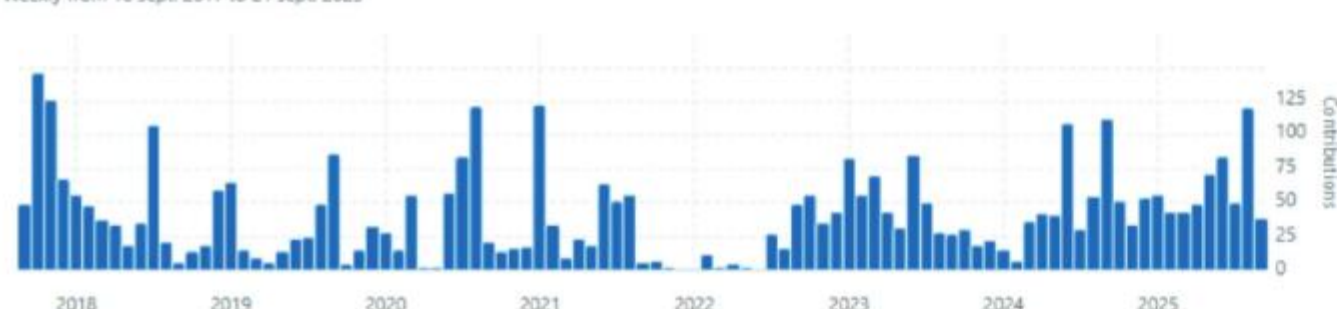
Commits over time

Weekly from 3 févr. 2013 to 21 sept. 2025



Commits over time

Weekly from 10 sept. 2017 to 21 sept. 2025



◀ Activité sur le github d'Alice Vision.

◀ Activité sur le github de Meshroom.

3. <https://alicevision.org/association/>

4. <https://github.com/alicevision/Meshroom>

5. <https://github.com/meshroomHub>

GPAC, UNE BOÎTE À OUTILS MULTIMÉDIA QUI SÉDUIT LES ACTEURS DU STREAMING

ENTRETIEN AVEC ROMAIN BOUQUEAU, FONDATEUR-CEO DE MOTION SPELL

Figurant parmi les principaux contributeurs du projet open source GPAC, le fondateur-CEO de Motion Spell - le bras commercial de GPAC, nous dit tout sur cette boîte à outils pas comme les autres.

Site du projet : <https://gpac.io>

Dépôt du code : <https://github.com/gpac/gpac>
romain.bouqueau.pro@gmail.com

► En quoi consiste ce projet, pour quels utilisateurs finaux ?

ROMAIN BOUQUEAU. Lancé en 2003, GPAC est une boîte à outils multimédia open source destinée aux chercheurs, aux étudiants et aux industriels. Le projet a donné naissance en 2013 à un « spin-off » commercial, Motion Spell/GPAC Licensing. GPAC est largement utilisé comme packager, notamment par Netflix et d'autres acteurs majeurs.



► Quelle licence avez-vous choisie pour ce projet, et pourquoi ?

R.B. Le projet est diffusé sous licence LGPLv2.1+, avec la possibilité d'obtenir une licence commerciale.

► Pour quel mode de gouvernance du projet avez-vous opté ?

R.B. La gouvernance est volontairement légère et informelle. Quelques contributeurs clés, issus du monde académique et de l'industrie, coordonnent les orientations. Une réunion de groupe a lieu toutes les six semaines.

► Quel modèle d'affaires avez-vous mis en place sur ce projet ?

R.B. Le modèle économique est une approche hybride qui combine licences commerciales et services (support, intégration, formation).

► Comment avez-vous instauré et maintenu la confiance de la communauté ?

R.B. Dans un domaine très spécialisé, bâtir la confiance n'est pas simple. Le recrutement de contributeurs passe par l'activité commerciale, des programmes comme Google Summer of Code et les événements du secteur. La communauté se compose principalement de power users et de développeurs très motivés, avec un noyau dur et un écosystème plus large autour. En 2025, l'ensemble des projets GPAC a dépassé les 100 millions de téléchargements.

► Quels sont les bénéfices et difficultés que vous retirez de ce choix de l'open source ? Quels enseignements en avez-vous tiré ?

R.B. Le bénéfice majeur que nous en tirons est de jouir d'un puissant moteur d'innovation économique. La monétisation est, quant à elle, le point de friction principale car beaucoup d'entreprises ne paient que lorsqu'elles y sont contraintes.

► Selon vous, quelle place tient l'open source dans notre écosystème actuel ?

R.B. Nous sommes demandeurs d'une structuration accrue entre les projets open source d'un même secteur, afin de mutualiser les ressources (développeurs, infrastructures, financements) et d'en décupler l'efficacité collective. Une telle coordination permettrait d'éviter les redondances, d'optimiser les moyens et de renforcer la compétitivité de l'écosystème dans son ensemble.

Propos recueillis par Benoît Maujean

Supported by

NETFLIX



MOTION SPELL



STAX, UN OUTIL DE REVIEW OPEN SOURCE BASÉ SUR BLENDER

ENTRETIEN AVEC CHRISTOPHE ARCHAMBAULT, DIRECTEUR TECHNIQUE DU STUDIO D'ANIMATION SUPERPROD

À l'heure où les pipelines de production se munissent de briques ouvertes, Christophe Archambault, directeur technique du studio d'animation Superprod, nous raconte l'ouverture de Stax – un logiciel maison de lecture d'images et d'annotation, construit sur Blender –, et revient sur les gains concrets d'USD et de Rez dans un studio d'animation.

Encart : Stax

Site du projet :

https://superprod.gitlab.io/stax_suite/stax/

Dépôt du code :

https://gitlab.com/superprod/stax_suite

► Quelles sont les raisons qui vous ont conduit à ouvrir Stax en open source ?

CHRISTOPHE ARCHAMBAULT. Trois raisons nous ont menés à ouvrir le code de Stax :

- Pérennité : nous manquions de ressources pour maintenir l'outil au rythme souhaité. En ouvrant son code, nous espérions accélérer son développement grâce à une dynamique de contributions, portée notamment par la communauté de Blender.
- Par conviction : le développeur principal était un véritable défenseur de l'open source, et nous savions qu'il serait moteur pour faire connaître le projet de la communauté Blender, qu'il connaissait bien.
- Expérimentation : nous n'avions encore jamais mené de projet open source en interne, Stax nous a servi de terrain d'expérimentation.

► Pourquoi avoir opté pour la licence GPLv3 ?

C.A. C'est une contrainte liée à Blender et à son API Python, sous GPL. Par cohérence et compatibilité, Stax a donc été publié sous GPLv3.

► Quel mode de gouvernance avez-vous mis en place ?

C.A. Le dépôt GitLab a été cogéré entre Superprod et le développeur principal. Celui-ci a depuis quitté le studio, mais le projet reste accessible et historisé.

► Quel était votre modèle économique ?

C.A. Nous n'avions aucun modèle économique autour de Stax. Notre objectif était avant de répondre à nos besoins techniques tout en s'essayant au développement d'une communauté open source.

► Comment avez-vous cherché à instaurer la confiance des utilisateurs et des contributeurs ?

C.A. Par la transparence et la visibilité : le code est accessible sur un dépôt GitLab public, les démos sont en accès ouvert, et notre communication est régulière. Le développeur principal était connu dans la communauté Blender, ce qui aidait à créer une confiance. Malgré cela, l'adoption n'a pas réellement décollé. À notre sens, un outil de visionnage et d'annotation exige une forte intégration au pipeline du studio (gestion d'assets, suivi de production, etc.). Le coût initial d'intégration et de tests est élevé, d'autant que beaucoup de studios disposent déjà d'une solution en place, souvent OpenRV¹.

► Quel bilan tirez-vous du projet ?

C.A. De notre côté nous avons dû procéder à un certain nombre de tâches indispensables pour mettre cet outil en open source. A commencer par rendre agnostique le code du studio, en supprimant les dépendances à nos services internes. Nous utilisons Rez² pour gérer nos dépendances logicielles, et il est impossible d'en faire un prérequis pour un projet open source. À terme, nous avons créé une version « interne » de Stax ; différente de celle publiée en open source, plus simple à main-

1. OpenRV est une visionneuse de séquences d'images développée par Autodesk et rendue open source.

<https://github.com/AcademySoftwareFoundation/OpenRV>

2. Rez est un gestionnaire open source de bibliothèques logicielles multiplateforme, porté maintenant par l'ASWF <https://github.com/AcademySoftwareFoundation/rez>



tenir pour nous avec Rez. Nous avons ensuite mis en place et configuré le GitLab public dont nous avons assuré la communication. Nous avons également animé la petite communauté initiale, avec des réunions hebdomadaires.

Du côté des retours, nous avons eu très peu de contributions externes. Il n'y a malheureusement pas eu de fidélisation de contributeurs, et notre développeur principal est parti. Le gain de visibilité a été ponctuel, y compris auprès de l'ASWF, autour du format ouvert de notes de review ORIO « Open Review IO »³ que nous avons publié avec l'outil. Mais l'expérience nous a permis de découvrir concrètement les difficultés à faire vivre un projet de ce type.

► USD et Rez : pourquoi l'open source compte dans un pipeline ? Quels avantages concrets voyez-vous à l'open source dans USD⁴ ?

C.A. Cela peut se résumer en trois points clés essentiels :

- Neutralité et adoption : le caractère ouvert d'USD a permis une adoption rapide par des éditeurs de logiciels concurrents. Si, par exemple, un nouveau format 3D propriétaire avait été proposé par un acteur unique, l'écosystème aurait sans doute été plus fragmenté ;
- Pérennité du format : pour un studio, investir dans un standard ouvert limite le risque de verrouillage et sécurise l'effort sur le long terme ;
- Accès au code : il est important de comprendre finement le comportement du logiciel et des passerelles possibles, de pouvoir diagnostiquer certains bugs en tirant parti de la communauté active

3. ORIO. <https://openreviewio-standard-definition.readthedocs.io/fr/latest/Integrations.html>

4. USD (Universal Scene Description) est un standard open source pour le partage et l'édition non destructive des composantes d'une scène 3D.
<https://openusd.org/docs/Introduction-to-USD.html>

de développeurs et d'utilisateurs et, si nécessaire, d'apporter des corrections nous-mêmes.

► Et sur le plan du partage de connaissances ?

C.A. Une communauté ouverte change la donne. Avec une technologie propriétaire, une grande partie de la connaissance réside uniquement dans la documentation et le support de l'éditeur. En open source, les discussions, la description des bugs et de leurs corrections sont publics – capitalisant l'expertise au bénéfice de tout l'écosystème.

Notre équipe développement s'est construite une très forte expertise autour d'USD, rendue possible par un investissement significatif en R&D sur ce sujet. Nous étions convaincus dès le début que cette technologie nous permettrait d'atteindre le niveau de qualité créative et visuelle auquel nous aspirons. Cette expertise n'aurait pas été possible sans l'accès à cette connaissance partagée.

► Où se situe Rez dans ce paysage ?

C.A. Rez est essentiel pour nous. Il standardise la gestion des dépendances et des environnements des productions (versions de logiciels et leurs plugins, etc.). C'est un socle qui facilite la mise en place et le déploiement des outils par production, tout en mutualisant les librairies communes. Dans le cas de Stax, Rez nous a permis de garder une variante interne plus facile à maintenir, même si nous ne pouvions pas l'imposer à la communauté.

► En guise de conclusion, que pourriez-vous nous dire ?

C.A. Stax nous a donné une première expérience de gestion de projet collaboratif. Ouvrir un outil ne garantit ni contributions ni adoption – surtout lorsque l'intégration au pipeline est lourde. En revanche, les standards et briques ouvertes comme USD et Rez offrent une valeur durable : interopérabilité réelle, lisibilité technique et autonomie des studios.

▲ Une partie de l'équipe de développement de Stax.

POURQUOI CHOISIR L'OPEN SOURCE

ENTRETIEN AVEC FLAVIO PEREZ, DIRECTEUR TECHNIQUE DU STUDIO LES FÉES SPÉCIALES

Flavio Perez, directeur technique du studio Les Fées Spéciales, est impliqué dans le logiciel libre et président du chapitre ACM SIGGRAPH de Montpellier. Il revient pour nous sur les enjeux du libre.

► « Ensemble, on va plus loin » : cette maxime résume pour toi un intérêt essentiel du logiciel libre. Que cela signifie-t-il concrètement ?

FLAVIO PEREZ. Nous avons fondé Les Fées Spéciales y a dix ans, avec la volonté que notre studio soit au service non seulement des projets mais aussi des salariés. C'est pourquoi la société a pris la forme d'une SCOP (société coopérative de production). Dans la même idée, nous avons fait le pari avec l'équipe d'utiliser Blender en production, logiciel libre que nous défendons depuis deux ans, malgré le scepticisme du secteur.

En tant que directeur technique, le choix de cet outil devenu désormais une référence, venait avec une réflexion initiale importante autour du code. Ce n'est pas évident de décider de partager ses développements et ça pourrait même sembler contre-intuitif puisque le pipeline est la colonne vertébrale d'un studio, et peut avoir une valeur concurrentielle. Cependant il est aussi sujet à risque : non seulement il nécessite un investissement régulier, mais il repose aussi bien souvent sur un seul permanent. Lorsqu'il s'en va, son successeur est tenté de repartir à zéro. On mesure alors la fragilité de cette stratégie et les conséquences du côté opérationnel.

En effet, comme tant d'autres TDs¹, j'étais passé d'un studio à l'autre pendant plusieurs années, à devoir repartir de zéro à chaque fois. Pourtant à ce moment-là beaucoup de productions profitaient du code développé ailleurs, et apporté de l'extérieur par les intermittents en toute illégalité, et au risque de faire tourner des versions obsolètes de certains outils. C'était une pratique courante qui a bénéficié à de nombreux films, puis s'est atténuée. Dès lors, s'il s'agissait d'un vrai besoin, pourquoi ne pas mieux l'encadrer en choisissant des outils libres ? L'un de nos premiers projets fût la fabrication ambitieuse de Dililil à Paris, et j'avais été

marqué par la fermeture de Duran Duboi, qui avait entraîné la perte de plus de vingt ans de R&D. La réponse à ces enjeux de disponibilité et de survie du code est finalement simple pour nous : tout le code que nous produisons est libre. Il pourra donc nous survivre, et il est disponible pour nos graphistes qui sont habitués à nos outils et qui veulent les utiliser sur leurs projets ou même dans d'autres sociétés (et ils l'ont souvent fait). Même si d'autres sociétés y trouvent leur intérêt, elles peuvent l'utiliser, l'améliorer, et y contribuer. Ainsi, on évite de refaire la roue, chacun dans notre coin : au contraire, on mutualise nos efforts.

► Sur quels projets travaillez-vous ?

F.P. Tout au long de chacune des productions que nous avons accompagnées, nous avons créé et amélioré des outils variés. D'un script simple à des contributions complexes au code source de DCC² tels que Blender, en passant par des outils de pipeline stratégiques comme Libreflow, ou des add-on pratiques même pour des indépendants, tout est libre et disponible sur nos dépôts gitlab³. Nous optons pour la licence CeCILL⁴, équivalent de GPL, mais faite de droit français. Pour les plug-in de Blender, il s'agit de code en licence GPL, que nous poussons sur leur nouvelle plateforme dédiée⁵.

► Peux-tu nous parler plus précisément de Libreflow ?

F.P. Libreflow⁶ est un logiciel de gestion d'assets, entièrement codé en Python, et qui permet de gérer nos projets dans toutes leurs versatilités. Avec l'aide du CNC, nous en avons fait un outil de synchronisation entre les différents sites d'un studio et fait trois versions qui ont accompagné plusieurs longs-métrages, une série télé, des courts-métrages et d'autres projets hybrides. Il se base sur le framework libre Kabaret, développé chez Supamonks à partir de 2012 par Damien Dee Coureau, dont je me

1. Technical Director : salarié en charge de développer et maintenir le pipeline d'un studio.

2. "Digital Content Creation": Logiciel de création graphique. Terme utilisé principalement pour désigner les logiciels de modélisation et d'animation 3D.

3. gitlab.com/lfs.coop

4. <https://cecill.info/>

5. extensions.blender.org

6. <http://libreflow.net>

suis servi pour aboutir avec le temps à Libreflow, une surcouche complète créée à partir de 2020. Puis nous avons de plus en plus contribué à Kabaret, jusqu'à reprendre sa maintenance en 2022, le faisant évoluer selon les nouvelles versions de Python, de Qt, etc., pour permettre à Libreflow de continuer à s'améliorer. De fait Kabaret a survécu, et nous avons toujours consulté Damien pendant toutes ces années.

Ce qui est intéressant à observer, c'est lorsqu'un studio crée un premier outil libre, que d'autres utilisent pour créer le leur. Cela donne lieu à de longues et riches discussions sur les pipelines. Le code est conçu de manière plus pérenne, et est supporté par de multiples contributeurs, ce qui réduit la charge de ces outils par projet. Bien répartir l'effort de développement est plus complexe, mais peut amener de très bons résultats et, au global, nous économisons tous nos efforts.

► Qu'arrive-t-il si les besoins divergent ?

F.P. Une façon de faire les choses est de partager son cahier des charges quand on a une idée. C'est ce que l'on fait fréquemment sur un Discord d'utilisateurs professionnels de Blender créé à l'initiative de Samuel Bernou d'Autour de Minuit. On se partage les projets de développement avec deux questions principales :

- Voici mon cahier des charges, est-ce que vous connaissez un équivalent que je pourrais utiliser et auquel on pourrait contribuer ?
- Si ce n'est pas le cas, est-ce que vous avez de meilleures idées ? Cette seconde question permet fréquemment d'imaginer un meilleur outil, car quitte à y passer du temps, essayons d'en faire un outil encore plus utile ou mieux pensé que ce qu'on aurait pu imaginer.

Nous le faisons systématiquement quand on prépare un add-on Blender. De nombreux add-on ont profité de retours, voire ont été complètement repensés, comme c'est le cas de Render Nodes, un

gestionnaire de rendu nodal pour Blender très influencé par des discussions avec les équipes d'Andarta. Nous y avons tous gagné.

► Faut-il inciter les autres studios à faire du libre ?

F.P. Bien sûr. En ce sens, l'Aide aux Moyens Techniques (AMT) du CNC est une très belle initiative. Pour une PME, la subvention passe de 45 % à 60 % des dépenses éligibles pour les projets libres ou gratuits. Je suggérerais même une catégorie d'intérêt collectif où le CNC pourrait aller plus loin encore, si l'apport est évident pour l'industrie, pourquoi ne pas le financer à 100 % ?

Il y a une idée reçue selon laquelle le studio devrait assurer le support de ses outils open source. Chaque licence libre, telle que GPL, MIT ou Apache, commence par préciser que le code est fourni en l'état, sans garantie ni aucune responsabilité de la part de ses auteurs. De notre expérience, alors que certains de nos add-on pour Blender comptabilisent des dizaines de milliers de téléchargements, on reçoit une demande par trimestre, et généralement on a envie de la traiter car il s'agit tantôt d'un bug non identifié, tantôt d'une bonne idée de fonctionnalité.

En bref, si vous réfléchissez à faire du libre, n'y pensez plus et faites-le, vous ne risquez rien. Au contraire, vos dépôts resteront longtemps inaperçus. Pour y remédier nous communiquons sur nos projets dans des conférences auxquelles nous participons, aux étudiants dans les écoles, sur notre blog technique <http://www.lacuisine.tech> quand un nouvel outil est prêt, et parfois on relaie sur notre LinkedIn. Il faut éviter aux autres de créer quelque chose qui existe déjà, en rendant le projet public, et en essayant de l'identifier dans l'océan qu'est Internet. Je milite pour la publication d'une base de données des studios qui partagent leurs outils, et une publicité par le CNC des projets aidés qui sont disponibles (y compris sous forme d'offres commerciales).

Propos recueillis par Frédéric Fermon



© Photo : Les Fées Spéciales

◀ L'équipe des Fées Spéciales.

L'ANIMATION POUR TOUS

ENTRETIEN AVEC NICOLAS « DUDUF » DUFRESNE, CO-PRÉSIDENT DE L'ASSOCIATION RXLABORATORY

Nicolas Dufresne, connu sous le pseudo « Duduf », a commencé en 2008 le développement de Duik, un add-on de rigging et d'animation de personnages pour Adobe After Effects, mis à disposition sous licence libre et open-source GNU-General Public License.

Page de Duik :

<https://rxlaboratorio.org/rx-tool/duik/>

Repo : <https://github.com/RxLaboratory/Duik>

► Quelle licence as-tu choisie pour Duik et pourquoi ?

NICOLAS DUFRESNE. Le choix de la licence GPL-3.0 s'est fait pour deux raisons. La première est très pragmatique : le projet a commencé comme un passe-temps et je n'avais aucune idée du succès qu'il pourrait avoir. Mettre en ligne une boutique, une licence privative, un système de contrôle, aurait représenté trop de travail. L'autre raison est philosophique et politique : j'estimais – et dans l'association que nous venons de fonder nous estimons toujours – qu'il est impossible de distribuer un produit numérique via un système mercantile, qui reposerait sur des restrictions à la duplication, mais aussi à l'usage et à la modification de nos propres outils. Je l'explique toujours par l'exemple du vélo : s'il paraît évident que j'ai le droit de réparer et améliorer moi-même mon vélo, de changer les freins ou les pneus à ma convenance, et aussi que j'ai le droit de le partager ou de le revendre, pourquoi ne serait-ce pas le cas avec mes logiciels ? La licence libre défend ces droits.

► Quel est le mode de gouvernance du projet ?

N.D. Mon problème aujourd'hui est plutôt d'obtenir des contributions extérieures que de devoir définir un mode de gouvernance : la question se posera quand il y aura suffisamment de contributions ; pour l'instant les choses se font de manière informelle mais fluide, via des canaux de discussion divers et les dépôts de code. La plupart des contributions concernent la traduction d'interfaces dans di-

verses langues, ce qui est un aspect très important, et sous-estimé, pour l'adoption autant que pour insister sur ma démarche d'ouverture au monde et d'accessibilité. Le jour où Duik a été traduit en Chinois, le nombre de téléchargements quotidiens a fait un bond notable (environ 800).

► Pour quel modèle d'affaires as-tu opté ?

N.D. Puisque les utilisateurs et utilisatrices ont le droit de partager, dupliquer et redistribuer sans limitation les outils libres, nous pensons que la notion de vente et de prix est à redéfinir ; s'il y a aujourd'hui des exemples de produits sous licence libre qui sont vendus dans un système qui semble fonctionner (par exemple, l'écosystème des plugins WordPress ou des add-ons Blender vendus), j'ai fait le choix à l'époque de plutôt pousser les utilisateurs et utilisatrices à faire des dons. Fixer un prix me pose deux problèmes : le montant du prix est impossible à définir en restant juste ; tout le monde n'a pas les mêmes revenus, c'est déjà vrai au sein d'un pays ou d'une région, c'est d'autant plus vrai lorsque notre produit est téléchargé et utilisé aussi bien au Japon et en Suisse qu'au Venezuela ou certains pays d'Afrique. C'est important aussi bien pour les particuliers et les indépendants, que les entreprises, petites ou grandes. L'autre problème porte sur l'équité de la rémunération du développeur ou de la développeuse dans un système mercantile, où c'est le succès qui fait le revenu. Je préfère mille fois un monde où un créateur, une créatrice, peut créer un outil de niche et y passer le temps nécessaire, sans pour autant être bénévole, plutôt qu'un monde où il faut absolument « que ça se vende ». En fait, j'essaie de restaurer une corrélation entre les revenus et le temps de travail, en décorrélant ce revenu du succès donc du nombre d'utilisateurs, ce qui n'est pas facile dans un monde où on est naturellement enclins à chercher à vendre « le produit qui marche », c'est-à-dire celui qui nous assure une rente.

Le succès inattendu de Duik, puis de mes autres outils, repose en partie sur tous ces choix : choisir une licence libre, favoriser la gratuité ou les prix libres, l'accessibilité, favoriser l'émancipation de toutes et tous. Si je n'y ai quasiment rien gagné financièrement, j'y ai gagné une reconnaissance sincère, rencontré et travaillé avec des artistes que j'admire, j'ai voyagé autour du monde pour partager mon analyse, et mon expérience de l'animation dont les



outils open source sont la vitrine, et le prétexte à diffuser un message politique. Et je suis convaincu que sans la licence libre, j'aurais gagné beaucoup d'argent, mais rien de ce qui me rend heureux aujourd'hui dans le domaine professionnel.

C'est en partant de ces réflexions qu'avec quelques collègues, amies et amis nous avons monté tout récemment une association, RxLaboratorio¹ qui se donne pour mission, entre autres de :

- Faire le relai entre les utilisateurs, les contributeurs et les mainteneurs des projets ;
- Distribuer et promouvoir les logiciels libres notamment dans le secteur audiovisuel, en particulier auprès des développeurs et développeuses, en commençant par les indépendants. Nous leur laissons la complète gouvernance de leur projet que nous distribuons.

Pour ce qui est de la distribution des logiciels libres et open source, l'association, au fonctionnement démocratique et collégial, s'inspire du modèle des AMAP (Associations pour le Maintien d'une Agriculture Paysanne). Elle collecte les contributions des utilisateurs et utilisatrices (dons, achats de tutoriels à prix libres, cotisations des membres...) et en redistribue une partie à celles et ceux dont elle distribue les outils, en décorrélant presque complètement le calcul des parts, du nombre d'utilisateurs et de téléchargements, mais en se basant sur un score « d'utilité sociale », « d'innovation », etc. attribué à chaque outil.

L'association est toute jeune et nous sommes en train de faire venir de nouveaux développeurs et développeuses ; elle distribue déjà plusieurs outils, mais les principaux sont pour l'instant mes projets commencés avant de monter la structure :

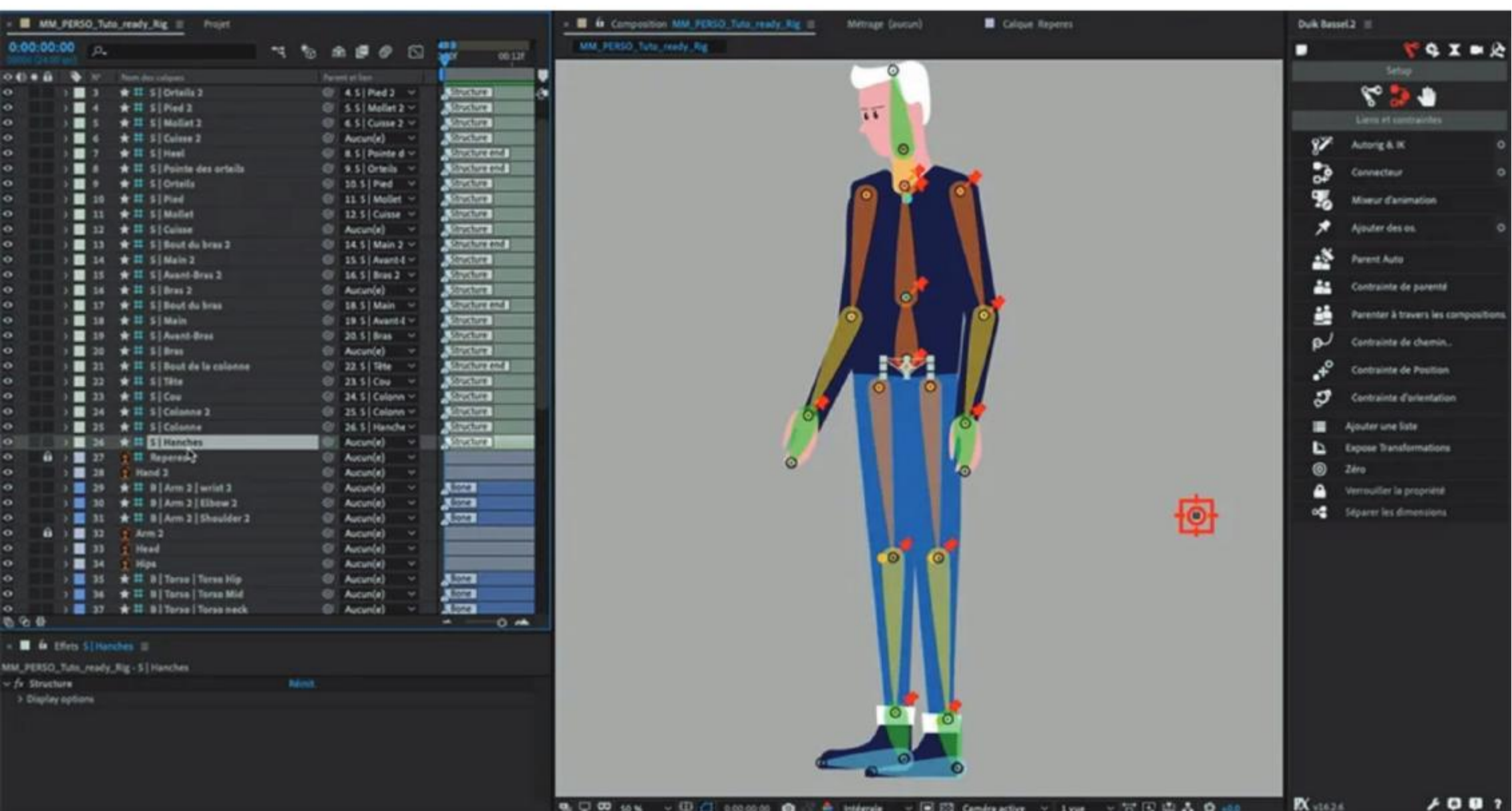
- Duik, qui est devenu ces 15 dernières années le standard pour animer des personnages dans un style « papier découpé » ;
- DuME², un encodeur multimédia très polyvalent ;
- Ramses³, outil de gestion de pipeline et de suivi de production en fin de développement et en test dans quelques studios et écoles comme Rubika.

► Quelles sont tes réflexions actuelles à ce stade du projet ?

N.D. Nous avons aussi un travail en cours sur de nouveaux formats de données ouverts, tels que OCA⁴ par exemple, « the Open Cel Animation format », qui permet déjà l'échange de données d'animation traditionnelle 2D entre des applications telles que Krita, OpenToonz, Callipeg, Fusion, After Effects, Blender... Les formats ouverts sont importants dans notre démarche vers l'émancipation des utilisateurs et utilisatrices qui doivent pouvoir choisir leurs outils librement.

Propos recueillis par Frédéric Fermon

1. <https://rxlaboratorio.org>
2. <https://rxlaboratorio.org/rx-tool/dume/>
3. <https://rxlaboratorio.org/rx-tool/ramses/>
4. <https://rxlaboratorio.org/rx-tool/oca/>



OUVRIR LES DONNÉES POUR LIBÉRER LA DISTRIBUTION

ENTRETIEN AVEC JERRY PIERCE, PRÉSIDENT DE INTERSOCIETY DIGITAL CINEMA FORUM (ISDCF)

chairman@isdcf.com

L'ISDCF¹ est une association volontaire fondée sur l'adhésion de ses membres, qui se penche sur la technique et le déploiement du cinéma numérique. L'une de ses principales activités est la maintenance de la Digital Cinema Naming Convention², fondée sur le projet open-source ISDCF Metadata Registries. Le président de l'ISDCF, Jerry Pierce, nous en présente les détails.



ISDCF Metadata Registries :

Site du projet :

<https://registry-page.isdcm.com/>

Dépôt du code :

<https://github.com/ISDCF/registries>

► En quoi consiste ce projet et à quels utilisateurs finaux est-il destiné ?

JERRY PIERCE. Le projet ISDCF Metadata Registries vise à fournir un accès libre aux métadonnées essentielles à la distribution des DCP à travers le monde entier. Il comprend à la fois les valeurs de métadonnées elles-mêmes (sous forme de registres) et le logiciel utilisé pour gérer et publier ces registres.

► Quelle licence avez-vous choisie pour ce projet ? Qu'est-ce qui a motivé ce choix ?

J.P. Le logiciel est diffusé sous la licence BSD 2-clause licence, et les données sous la licence Creative Commons Attribution 4.0 International License. Ces deux licences sont permissives, largement utilisées, et permettent à toute personne d'exploiter les registres avec un minimum de contraintes.

► Pour quel mode de gouvernance avez-vous opté ?

J. P. : Les registres sont gérés par le comité technique de l'ISDCF, un groupe reposant sur le consensus de ses membres. Les non-membres peuvent également apporter leur contribution via le dépôt GitHub.

► Quel modèle d'affaires avez-vous mis en place sur ce projet ?

J.P. L'entretien des ISDCF Metadata Registries est financé par les membres de l'ISDCF au moyen de cotisations annuelles.

► Comment avez-vous instauré et maintenu la confiance de la communauté ?

J.P. La confiance dans le projet est maintenue grâce à la réactivité face aux signalements de bogues et aux demandes de fonctionnalités, qu'ils proviennent de membres de l'ISDCF ou de non-membres.

► Comment ces développeurs « de la communauté » se sont-ils lancés dans une contribution ?

J.P. La résolution de chaque bogue signalé et de chaque demande de fonctionnalité est soumise au consensus du comité de l'ISDCF et documentée de façon publique et transparente.

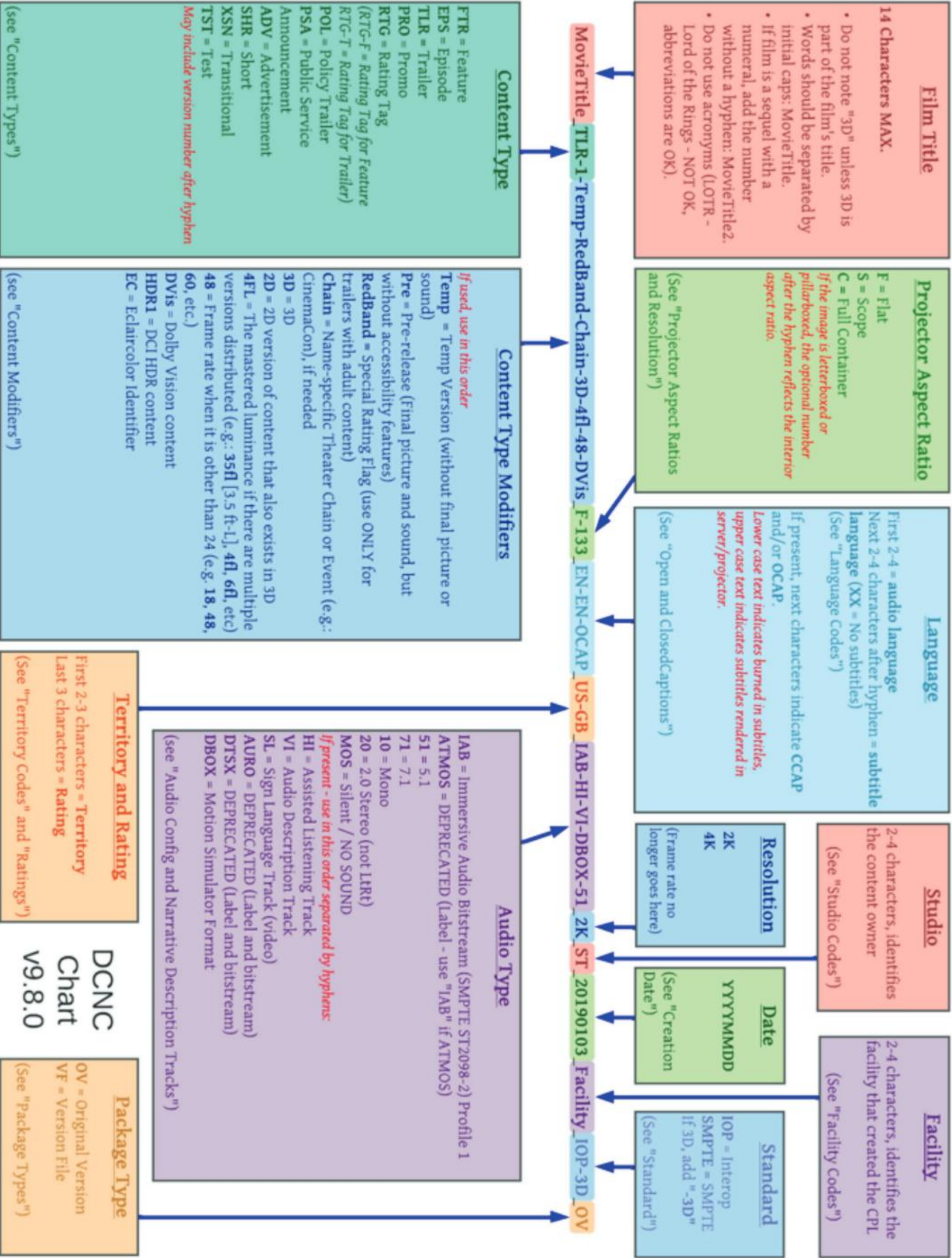
► Quels sont les bénéfices et difficultés que vous retirez de ce choix de l'open source ?

J.P. Le caractère open source des ISDCF Metadata Registries est essentiel pour garantir l'ouverture et la pérennité de cette source d'information, qui est cruciale pour la distribution réussie des DCP à travers le monde.

Propos recueillis par Benoît Maujean

1. <https://www.isdcm.com/>

2. <https://registry-page.isdcm.com/>



ENTRETIEN AVEC THIERRY LELÉGARD, CRÉATEUR DE TSDUCK

Thierry Lelégard a travaillé pendant vingt-cinq ans sur la sécurité des CAS et DRM dans la télévision numérique, depuis la création de celle-ci chez Canal+ Technologies, puis Nagra, Logiways, STMicroelectronics et Orange. Il travaille actuellement dans la cybersécurité des microprocesseurs chez SiPearl. Il nous parle de TSDuck, trousse à outils indispensable destinée aux ingénieurs de télévision numérique et de streaming multimédia.

Site du projet : <https://tsduck.io/>

Dépôt du code : <https://github.com/tsduck/tsduck>

► **En quoi consiste ce projet et à quels utilisateurs finaux est-il destiné ?**

THIERRY LELÉGARD. TSDuck est un environnement de test, debug, intégration, monitoring, démo, ou production, sur les transports stream MPEG qui sont à la base de la transmission de la télévision numérique, broadcast, multi-cast ou streaming HLS. Il est destiné aux ingénieurs de développement, de test ou de production dans la télévision numérique ou le streaming multimédia.

► **Quelle licence avez-vous choisie pour ce projet, et pourquoi ?**

T.L. La licence est la BSD à deux clauses, ou « BSD simplifiée ». Elle a été choisie en tant que licence permissive, qui permet d'intégrer le projet dans des environnements propriétaires au besoin. C'est un choix pragmatique, qui correspond à la réalité des besoins utilisateurs. Les licences restrictives comme la GPL ont été écartées d'emblée, trop idéologiques et éloignées des besoins professionnels concrets.

► **Pour quel mode de gouvernance avez-vous opté ?**

T.L. La gestion du projet est pour l'instant sous la seule responsabilité de l'auteur principal. Ce n'est pas un choix, simplement une absence de volontariat tiers. La base de code (environ 500 000 lignes) est en majorité développée par l'auteur principal, mais toutes les contributions de code sont bienvenues, après revue. GitHub répertorie 41 contributeurs différents. En dehors de l'auteur

principal, les autres contributions vont de 1 à 200 commits par personne.

► **Quel modèle d'affaires avez-vous mis en place sur ce projet ?**

T.L. Le projet est purement gratuit et open source. Il n'y a pas de modèle d'affaires. Il y a une page « donation » comme beaucoup de projets open source mais elle est peu alimentée (également comme beaucoup de projets open source). Le projet a démarré il y a près de 20 ans comme un besoin professionnel pour l'auteur, faute d'outils équivalents. Il continue sur la base du volontariat et du bénévolat. Le principe fondateur est que nous recevons tous de la valeur du monde open source (Linux, Apache, ffmpeg, VLC, etc.) et qu'il est juste de contribuer en retour à la mesure de ce que l'on produit.

► **Comment avez-vous instauré et maintenu la confiance de la communauté ?**

T.L. TSDuck a gagné la confiance d'une communauté d'utilisateurs professionnels. Quelques statistiques : près de 1 000 « stars » sur GitHub, chaque version binaire est téléchargée entre 6 000 et 10 000 fois (selon l'API GitHub). TSDuck est un produit technique, flexible et adaptable, développé par des ingénieurs pour des ingénieurs. C'est le principal lien de confiance entre développeurs et utilisateurs. TSDuck est présenté et livré comme un « produit », avec une documentation utilisateur (700 pages dans la version PDF), une documentation développeur, des packages binaires tout prêts pour Linux, macOS et Windows, des scripts complets de reconstruction sur les trois plateformes. TSDuck se distingue ainsi volontairement de quelques projets open source très utilisés mais peu documentés et/ou difficiles à reconstruire hors de la plateforme favorite des auteurs.

► **Comment ces développeurs « de la communauté » se sont-ils lancés dans une contribution ?**

T.L. Le forum de discussion sur GitHub est assez actif. TSDuck est très orienté utilisateurs. Les bonnes idées d'amélioration sont prises en compte. Certains utilisateurs qui sont aussi développeurs sont encouragés à proposer des solutions à leurs idées. C'est la principale motivation des contributions de code.



► **Quels sont les bénéfices et difficultés que vous retirez de ce choix de l'open source ? Quels enseignements en avez-vous tirés ?**

T.L. Les bénéfices sont de plusieurs ordres. Ils sont d'abord d'ordre technique et professionnel puisque TSDuck apporte une trousse à outils indispensable à l'auteur qui a d'abord été un ingénieur de télévision numérique. Il est également relationnel : le projet fédère et met en relation des professionnels des cinq continents autour d'un sujet technique, lequel est mis en œuvre autour de normes et de pratiques différentes dans chaque région du monde. Enfin, on pourrait parler de bénéfice intellectuel avec la satisfaction de « développer un produit correctement », dans les règles de l'art, sans les contraintes et compromis d'une entreprise commerciale.

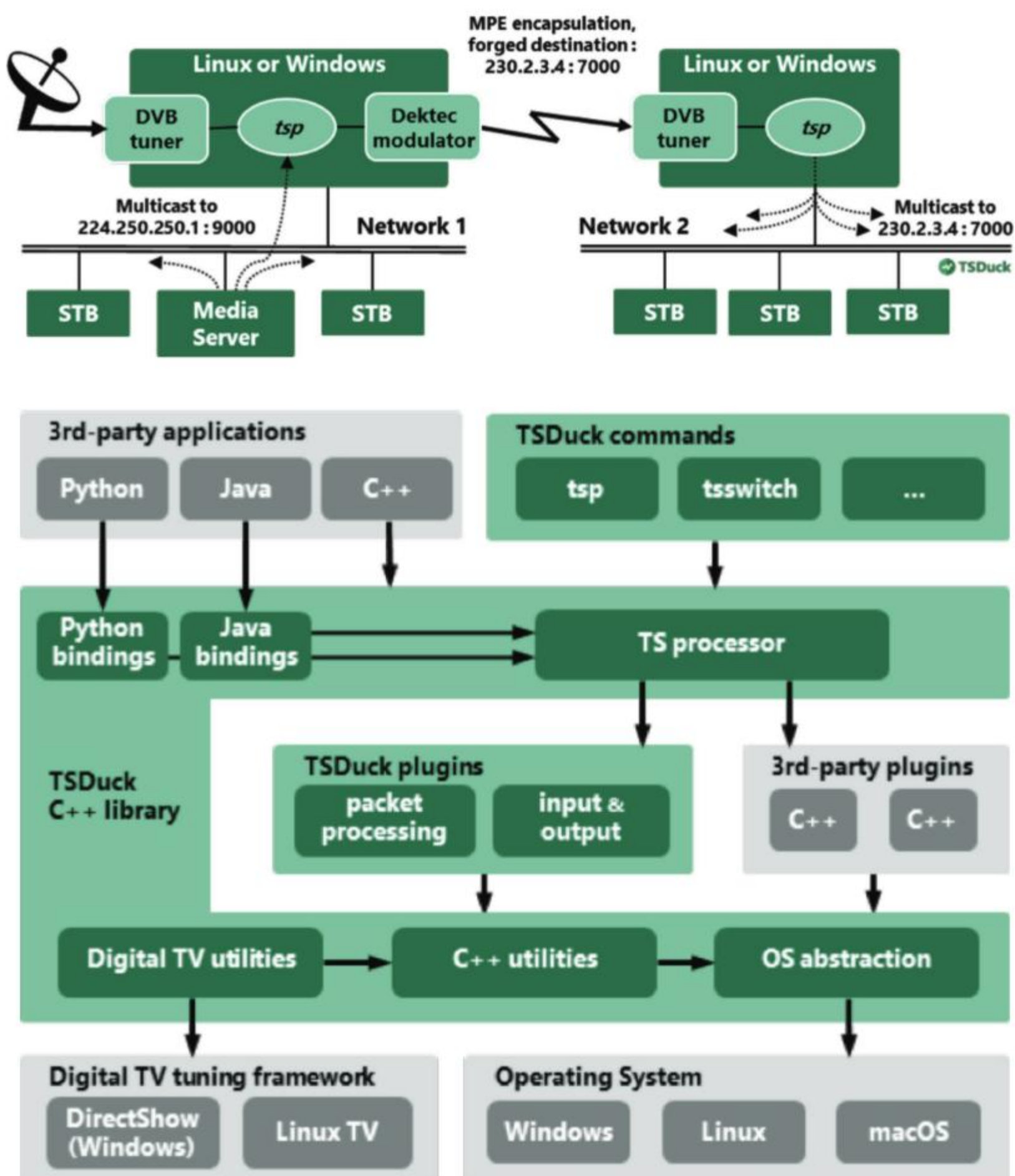
Les difficultés, quant à elles, sont évidemment d'ordre économique. Le choix de faire un projet

utile et sans compromis, sans modèle d'affaire, oblige à avoir une activité professionnelle à part, laissant moins de temps au projet.

Dans le milieu audiovisuel et multimédia, les outils open source sont nombreux, riches, puissants, et souvent d'origine française (VLC, ffmpeg, GPAC, MedialInfo, et TSDuck par exemple). Les plus puissants d'entre eux sont à la base du fonctionnement des plus grands opérateurs dans le monde, le tout gratuitement ou presque, alors que ces outils contribuent aux profits de ces opérateurs. Force est de constater que le retour financier vers les outils open source est faible, voire nul dans certains cas.

Dans un modèle strictement économique, ces outils n'existeraient donc tout simplement pas. Seules la force de conviction et la passion d'un monde de développeurs peuvent les maintenir en vie.

Propos recueillis par Benoît Maujean



Afterwork **Open Source**

Échanges & Démonstrations

Mardi 9 décembre à 18h
chez Cap Digital
14 rue Alexandre Parodi
75010 Paris



Inscription
obligatoire



cap·digital
Paris Region

Ficam
CINÉMA AUDIOVISUEL MULTIMÉDIA