



# Marc Coiffier

Développeur / Administrateur système

✉	marc@coiffier.net
☎	06 59 83 75 00
🎂	28 Juin 1991
🏠	15, allée des Vosges 38130 Échirrolles

🌐 CV en ligne (<https://coiffier.net/CV/?lang=fr>)

## Savoir-être

Empathie

Curiosité

Adaptabilité

Travail en équipe

Esprit d'initiative

Persévérance

Politesse

## Savoir-faire

Programmation (Haskell, Python, Shell, C/C++, ...)

Administration système Linux (Debian, ArchLinux, NixOS)

Automatisation / Déploiement (Docker, Nix, CI/CD, ...)

Anglais courant

Développement collaboratif (Git, GitHub, SourceHut, ...)

Méthodes formelles et spécification (Coq, Idris)

Technologies Web (JavaScript, HTML/CSS, APIs, ...)

## Expériences professionnelles

2024-2025

### Administrateur système et réseaux à l'IPAG

Administration du parc informatique de l'IPAG, un institut de planétologie regroupant plus de 150 utilisateurs, requérant une variété d'environnements et outils informatiques

Développement d'un logiciel de "self-service" sous Linux, afin de proposer aux utilisateurs du laboratoire un catalogue de logiciels adaptés à leur activités, et installables en un clic.

Conception, développement et déploiement d'une application Web de monitorat de parc informatique pour faciliter le maintien de la PSSI (politique de sécurité des services informatiques)

Participation au projet de développement de code communautaire Idefix (<https://github.com/idefix-code/idefix>), en collaboration avec Geoffroy Lesur, et travail sur l'intégration continue du projet dans des environnements HPC divers (cluster Jean-Zay à l'IDRIS, Adastra à GENCI et Bigfoot à GRICAD)

2016-2020

### Thèse en preuves de programmes à VERIMAG

Conception d'une extension du calcul des constructions capable de générer des principes d'inductions pour des encodages de Church.

Utilisation de l'assistant de preuve Coq pour prouver la cohérence et la normalisation forte de cette extension.

Utilisation de Coq pour prouver la correction de certains algorithmes de la bibliothèque standard OCaml (par exemple, que son implémentation de MergeSort décrit bien un tri correct et stable).

Développement d'un "wiki" de preuves, site statique écrit en Markdown littéraire permettant d'explorer et de vérifier des fragments de preuves directement sur le navigateur

2012-2016

## Administrateur de parc informatique à l'UFR IM<sup>2</sup>AG

Administration du parc informatique pédagogique et des serveurs de l'[UFR IM<sup>2</sup>AG](#).

- Assistance technique de niveau 1 aux personnels de l'établissement, en lien avec la DSI de l'UGA
- Déploiement d'images Linux/Windows avec CloneZilla en PXE
- Orchestration des mises à jour / installation de logiciels sous Linux via un serveur de paquets Debian locaux
- Documentation des services et de l'infrastructure, dans un wiki partiellement accessible aux utilisateurs
- Monitorat des postes grâce à TARSIS, un logiciel de monitorat d'infrastructure développé en interne

## Mes Projets

---

### GTK-Stream : un outil de gestion d'interface graphiques GTK

Lien vers le projet (<https://coiffier.net/projects/gtk-stream/>)

Un outil écrit en Python, qui permet de créer et de piloter une application graphique en établissant une connexion bidirectionnelle sérielle permettant d'envoyer des messages encodés en XML, et de recevoir des événements graphiques.

- Portable : testé sur Linux, MacOS et le WSL
- Léger : ne dépend que de GTK et de la librairie standard Python
- Facile : utilise des technologies simples (entrées/sorties standards et XML) et s'apprend en quelques heures
- S'intègre sans effort à n'importe quel type de projet

### Fix : des outils et un protocole de distribution logicielle

Lien vers le projet (<https://coiffier.net/projects/fix/>)

Des outils qui facilitent le développement, la construction, la distribution, l'installation et la mise à jour de logiciels de manière décentralisée, sécurisée, auditable et multi-plateforme.

- Sécurisé : basés sur un protocole client-serveur (le RSDP) afin de permettre l'installation de logiciels non approuvés sans leur accorder de droits d'administration
- Multiplateforme : permet la création de plans de construction multi-plateformes, évitant la nécessité de maintenir des paquets pour chaque distribution et architecture
- Reproductible : représente les dépendances d'un programme à l'aide d'identifiants cryptographiques, afin de garantir l'intégrité et la reproductibilité des artefacts logiciels
- Décentralisé : via une gestion des "métadonnées signées", permet à chaque projet indépendamment d'associer des informations dynamiques et non-falsifiables (dépréciation d'une version, mises à jour disponibles, ...) à propos des paquets qu'ils maintiennent

### Curly : un compilateur pour un langage fonctionnel simple

Lien vers le projet (<https://coiffier.net/projects/curly/>)

Un compilateur développé en Haskell, capable de builds reproductibles et multi-plateformes.

- Un système de type capable d'inférer des types récurifs (comme  $\lambda x. xx$ )
- Des classes de types de premier ordre
- Une édition des liens en une seule passe, pour une génération de code assembleur en just-in-time ou en exécutable statique
- Des modules *content-addressed* pour rendre la compilation reproductible par défaut
- Un format de documentation intégré au compilateur, et une documentation de second ordre (fini les commentaires de documentation)
- Du *Code-signing* automatique lors de la distribution de modules

## Omega : un micro-kernel

Lien vers le projet (<https://coiffier.net/projects/omega/>)

Un projet d'apprentissage, qui m'a permis d'en apprendre beaucoup sur la façon dont les systèmes d'exploitation fonctionnent, dans les moindres détails.

- Multi-processus avec protection de la mémoire
- Multi-thread avec synchronisation inter-processus
- Doté d'un langage de script et d'une CLI
- Drivers clavier, et VGA rudimentaire
- Très rapide à démarrer !

## Viz : un visualiseur de réseaux d'interaction

Lien vers le projet (<https://coiffier.net/projects/viz/>)

Dans le but de développer un algorithme beta-optimal d'évaluation du calcul des constructions, qui se repose sur l'utilisation de réseaux d'interaction.

- Création et suppression de noeuds et d'arêtes, dans un canvas SVG
- Application de règles de réduction pour visualiser les étapes intermédiaires de réductions complexes
- Repositionnement automatique des noeuds du graphe pour minimiser la "tension interne" et rendre la visualisation plus lisible (utilise la descente de gradient)

## Diplômes

---

2016

### Ingénieur Réseaux et Télécommunications

Validation des acquis d'expérience (VAE) après 4 années d'administration système à l'UFR IM<sup>2</sup>AG

2011

### Licence Mathématiques et Informatique

À l'Université Joseph Fourier de Grenoble

2008

### Baccalauréat Général

Au lycée Champollion de Grenoble

## Centres d'intérêt

---

Musique : piano, violon et guitare

Danse : Lindy Hop avec Grenoble Swing (<https://www.grenobleswing.com/>) et danses folk à l'Escapade Folk (<https://escapadefolk.netlib.re/>)

Jeux de société à la Maison des Jeux (<https://maisondesjeux-grenoble.org/wordpress/>) de Grenoble