

# Expérience Go

*Michel Casabianca*  
[casa@sweetohm.net](mailto:casa@sweetohm.net)

Voici un résumé de mon expérience dans le domaine du Go. Une [version complète de mon CV est disponible ici](#).

English version [is available here](#).

## **Lead Backend Developer chez Intercloud (2021-08 à maintenant)**

En tant que Lead Backend Developer, je développe des microservices en Go avec les frameworks Gin + Gorm + SwaGin. J'oriente les choix techniques en réalisant des POCs en concertation avec l'équipe. J'ai aussi contribué à des projets Open Source afin de corriger et améliorer les outils suivants :

- [Executor Tavern pour Venom](#), un outil permettant d'exécuter des tests d'intégration
- [Gobinsec](#), un outil pour analyser les binaires Go afin de détecter des vulnérabilités sur les bibliothèques embarquées
- [Correction de bugs et nouvelles fonctionnalités sur Swagin](#), un framework qui combine Gin et Swagger afin de générer la documentation des API

J'ai aussi assuré des présentations sur des sujets techniques sur le Go : Gobinsec, SwaGin, Go Generics, NeON. J'ai aussi soumis à l'équipe des challenges de code et publié des articles sur le blog d'Intercloud (à l'adresse <https://blog.intercloud.com>).

## **CTO chez DESKi (2020-06 à 2021-08)**

Développement d'une application Django pour l'administration des patients, examens et opérateurs de l'application Android et consultation des échographies par les cardiologues. La synchronisation avec l'app est réalisée avec un webservice REST en Go. Développement d'un viewer web d'échographies cardiaques avec la bibliothèque Cornerstone.

## **Webservice chez OVH (2019-11 à 2020-05)**

Mise en oeuvre chez OVH pour un webservice haute performance traçant les codes dépréciés. Il peut traiter plus de 8000 appels par seconde et les enregistrer en base de données PostgreSQL.

## **Outils pour MieuxPlacer.com (2018-03 à 2019-09)**

Chez <http://mieuxplacer.com>, j'ai mis en oeuvre Go pour écrire des scripts d'appel à des webservices et des batchs de récupération de données économiques (requêtes SQL et appels REST). Le gain de

performances obtenu a été spectaculaire.

Par ailleurs j'ai assuré un atelier Go et TDD.

## Outils pour les Éditions Dalloz (2016-09 à 2018-01)

Au cours de l'année 2017, j'ai développé la chaîne de publication des ouvrages et encyclopédies Dalloz. En cette occasion, j'ai codé nombre d'outils en Go. En particulier, j'ai développé en Go et mis en œuvre un [outil de build appelé Néon](#) ainsi qu'un webservice REST pour la publication des ouvrages.

## Mise en œuvre chez Orange (2008-04 à 2015-10)

J'ai découvert le Go début 2014 et l'ai rapidement mis en œuvre au sein d'Orange pour la réécriture d'un serveur SMS haute-performance. Ce serveur avait été implémenté en Java, avec NIO, et était depuis deux années en débogage pour le faire passer en production.

Devant les difficultés rencontrées pour corriger le code existant, j'ai proposé de développer des POCs, en Java sans NIO et en Go, afin de faire un choix éclairé de la technologie à mettre en place. Ces versions simplifiées ont été développées en une semaine chacune et à l'issue de tables rondes, il a été décidé de partir sur la solution Go, pour des questions de simplicité de code et de performances.

Ce travail a donné lieu à une [conférence à BDX.IO](#) et aux DevDays Orange en 2014.

## Activités personnelles (depuis 2014)

Depuis que j'ai découvert le langage Go, il est devenu mon langage de prédilection pour mes projets personnels.

Ainsi, j'ai un [compte Github](#) assez [actif dans le domaine du Go](#) :

- [Gointerfaces](#) est un programme qui télécharge les sources de Go et en extrait la liste de toutes les interfaces définies dans le langage. On peut voir le [résultat de cette extraction sur mon site](#).
- [Goactivate](#) est un script de définition de l'environnement Go comparable à la commande `activate` du `virtualenv` Python.
- [Néon](#) est un outil de build multi-plateforme et multi-langage.
- [Single](#) est un outil qui permet de s'assurer qu'une seule instance d'une application tourne sur un serveur.
- [Dotrun](#) permet de lancer une commande dans un environnement chargé d'un fichier `dotenv`.
- Il y a sur mon compte Github de [nombreux autres outils écrits en Go](#).