



Stagiaire : Basile André

Tuteur entreprise : Rémi Bouilly

Tuteur académique : Jean-François Favennec

Remerciements

Remerciements

Je remercie en particulier mon tuteur en entreprise, M. Rémi Bouilly, pour son accompagnement, sa disponibilité et la confiance accordée tout au long du stage. Je remercie également mon tuteur académique, M. Jean-François Favennec, pour ses conseils et son suivi pédagogique.

Enfin, je remercie l'ensemble des agents de la DEAL Réunion, et plus particulièrement l'équipe de l'Unité Biodiversité, pour leur accueil et leur collaboration.

Résumé (FR)

Au cours de ma formation d'ingénieur généraliste à l'ENIB (École Nationale d'Ingénieurs de Brest), j'ai eu l'opportunité d'explorer des domaines variés, dont l'informatique, le traitement des données et la communication réseau. Ces enseignements, combinés au contexte environnemental actuel, m'ont donné l'envie d'explorer comment les outils numériques peuvent répondre à des enjeux concrets, en particulier ceux liés à l'environnement et au milieu marin, un domaine qui m'inspire depuis longtemps. C'est cette motivation qui m'a conduit à choisir la DEAL (Direction de l'Environnement, de l'Aménagement et du Logement), où le développement d'outils informatiques joue un rôle clé dans la valorisation et l'accessibilité des données, offrant une occasion idéale d'allier mes compétences techniques à un sujet qui a du sens.

J'ai effectué mon stage au sein de la DEAL Réunion, plus précisément au SEB (Service Eau et Biodiversité), à l'UBIO (Unité Biodiversité). L'île de La Réunion présente des enjeux importants de suivi et de préservation de la biodiversité, en raison de la singularité de ses écosystèmes et des pressions qu'ils subissent (notamment les EEE (Espèces Exotiques Envahissantes) et le développement du territoire). Ce contexte renforce la nécessité d'un suivi rigoureux des données naturalistes.

La DEAL utilise Géonature (Application de gestion et de centralisation des données naturalistes) pour centraliser et gérer ces données. Le sujet principal de mon stage consistait à développer un module externe permettant d'y importer automatiquement les données marines issues de Quadrige (système d'information de l'Ifremer dédié aux données environnementales marines). Ces données, essentielles pour le suivi environnemental littoral, représentent un enjeu majeur pour les missions de la DEAL.

N'ayant pas suivi le module CAI (Conception d'Applications Interactives), ce stage m'a permis d'acquérir des compétences complémentaires en développement logiciel et applicatif, à travers la conception d'un module d'import, la manipulation de données et l'interfaçage de systèmes hétérogènes, afin d'obtenir un outil simple d'utilisation, robuste et maintenable.

Il m'a également offert une immersion dans le fonctionnement d'une administration publique de taille moyenne, où l'alternance entre travail autonome et collaboration avec les agents en charge des données naturalistes et de l'administration de Géonature s'est révélée particulièrement enrichissante. Cette expérience en environnement professionnel réel a renforcé ma compréhension des enjeux liés à la gestion des données environnementales et a affiné ma capacité à questionner la pertinence, l'utilité et le sens des projets auxquels je contribue, afin de m'assurer qu'ils soient en accord avec mes valeurs et porteurs d'un impact positif.

Abstract (EN)

As part of my general engineering studies at ENIB, I had the opportunity to explore a wide range of fields, including computer science,

data processing and network communications. These courses, combined with the current environmental context, strengthened my interest in understanding how digital tools can address real-world challenges — particularly those related to the environment and the marine domain, which has long inspired me. This motivation led me to choose DEAL, where the development of digital tools plays a key role in enhancing the value and accessibility of environmental data, offering an ideal opportunity to align my technical skills with a topic that holds meaning for me.

I completed my internship at DEAL Réunion, more specifically within the SEB and the UBIO. Réunion Island faces major biodiversity challenges due to its high proportion of endemic species and the presence of particularly sensitive natural habitats. These ecosystems are under strong pressure, notably from EEE and ongoing land-use development, which reinforces the need for rigorous naturalist data. DEAL uses Géonature to centralise and manage these datasets. The main objective of my internship was to develop an external module for this application, enabling the automated import of marine and coastal environmental data produced by Ifremer (Institut Français de Recherche pour l'Exploitation de la Mer). These data are essential for biodiversity monitoring and are accessed through Quadrige, Ifremer's dedicated information system.

Since I had not taken the CAI module, this internship allowed me to acquire complementary skills in software and application development through the design of an import module, data manipulation and the interfacing of heterogeneous systems, with the aim of producing a simple, robust and maintainable tool.

It also offered me insight into the functioning of a medium-sized public administration, where the alternation between autonomous work and collaboration with staff in charge of naturalist data and the administration of Géonature proved particularly enriching. This experience in a real professional environment strengthened my understanding of the challenges associated with environmental data management and refined my ability to reflect on the relevance, usefulness and purpose of the projects I contribute to, ensuring that they align with my values and have a positive impact.

Table des matières

| | |
|---|-----------|
| Remerciements | 1 |
| Résumé (FR) | 2 |
| Abstract (EN) | 3 |
| Glossaire | 6 |
| 1 Présentation de l'entreprise | 7 |
| 1.1 DEAL Réunion | 7 |
| 1.2 Diagnostic RSE | 8 |
| 1.2.1 Risques et impacts des activités | 8 |
| 1.2.2 Stratégie globale en matière de RSE | 9 |
| 1.2.3 Conclusion personnelle | 10 |
| 2 Organisation du stage | 11 |
| 2.1 Contexte du stage | 11 |
| 2.2 Objectifs du stage | 11 |
| 2.3 Méthodologie de travail | 12 |
| 3 Introduction à GeoNature | 15 |
| 3.1 Principe général de GeoNature | 15 |
| 3.2 Architecture technique | 15 |
| 3.3 Organisation interne et modules | 16 |
| 4 Développement du module d'import Quadrigé | 17 |
| 4.1 Contexte du projet | 17 |
| 4.2 Périmètre fonctionnel | 18 |
| 4.3 Spécifications techniques | 19 |
| 4.4 Contraintes, dépendances et livrables | 20 |
| 5 Travaux réalisés et compétences acquises | 21 |
| 5.1 Travaux réalisés | 21 |

| | |
|---|-----------|
| 5.2 Difficultés rencontrées et solutions apportées | 23 |
| 5.3 Compétences acquises | 24 |
| 6 Perspectives | 25 |
| 6.1 Perspectives d'évolution du module d'import Quadrige | 25 |
| 6.1.1 Amélioration de l'ergonomie et de l'intégration visuelle | 25 |
| 6.1.2 Génération directe de fichiers au format GeoNature | 25 |
| 6.1.3 Centralisation de la configuration côté backend | 26 |
| 6.1.4 Vers une automatisation partielle des imports | 26 |
| 6.2 Perspectives autour du parser Pl@ntNet et du module api2gn | 26 |
| 6.2.1 Ajout d'une interface d'administration dédiée | 26 |
| 6.2.2 Contraintes liées au module api2gn | 27 |
| 6.2.3 Vers une intégration renforcée des données participatives | 27 |
| 6.3 Ouvertures transversales | 27 |
| 7 Conclusion | 28 |
| Bibliographie | 29 |
| Annexes | 30 |
| Webographie | 32 |

Glossaire

| | |
|------------------|--|
| API | Interface de Programmation de l'Application |
| Borbonica | Plateforme régionale de diffusion des données naturalistes du SINP |
| CAI | Conception d'Applications Interactives |
| DEAL | Direction de l'Environnement, de l'Aménagement et du Logement |
| EEE | Espèces Exotiques Envahissantes |
| ENIB | École Nationale d'Ingénieurs de Brest |
| ERC | Éviter – Réduire – Compenser |
| Géonature | Application de gestion et de centralisation des données naturalistes |
| Ifremer | Institut Français de Recherche pour l'Exploitation de la Mer |
| OFB | Office Français de la Biodiversité |
| Pl@ntNet | Application mobile de reconnaissance des plantes par photo |
| PNRun | Parc National de La Réunion |
| Quadrige | système d'information de l'Ifremer dédié aux données environnementales marines |
| RNE | Responsabilité Numérique des Entreprises |
| RSE | Responsabilité Sociétale des Entreprises |
| SEB | Service Eau et Biodiversité |
| SEOR | Société d'Études Ornithologiques de La Réunion |
| SINP-974 | Système d'Information de l'iNventaire du Patrimoine naturel de la Réunion |
| SREPEN | Société réunionnaise pour l'étude et la protection de la nature |
| UBIO | Unité Biodiversité |

1. Présentation de l'entreprise

1.1. DEAL Réunion

La Direction de l'Environnement, de l'Aménagement et du Logement (DEAL) est le service déconcentré de l'État. Elle met en œuvre, à l'échelle régionale, les politiques publiques relevant du Ministère de la Transition Écologique et de la Cohésion des Territoires, ainsi que du Ministère de la Transition Énergétique. La Réunion est un territoire insulaire soumis à de fortes pressions environnementales et à des enjeux d'aménagement complexes. Dans ce contexte, la DEAL occupe une place centrale. Elle se situe au croisement des questions d'environnement, de biodiversité, d'eau, d'urbanisme et de développement territorial.

La DEAL Réunion assure l'application des réglementations environnementales. Elle instruit les projets d'aménagement et gère les risques naturels. Elle suit la ressource en eau et met en œuvre les politiques de protection des milieux naturels. Elle travaille en étroite collaboration avec les collectivités et plusieurs établissements publics, tels que l'OFB (Office Français de la Biodiversité), le PNRun (Parc National de La Réunion) ou l'Ifremer. Elle coopère aussi avec de nombreuses associations locales, comme la SEOR (Société d'Études Ornithologiques de La Réunion) ou la SREPEN (Société réunionnaise pour l'étude et la protection de la nature), qui contribuent activement au suivi et à la préservation de la biodiversité insulaire.

Au sein de cette structure, le Service Eau et Biodiversité (SEB) porte les missions liées à la préservation des milieux aquatiques et terrestres. Il développe la connaissance des espèces, veille à leur protection et régule les activités susceptibles d'impacter la biodiversité. Le SEB se trouve ainsi au cœur des enjeux écologiques de l'île.

Mon stage s'est déroulé au sein de l'Unité Biodiversité (UBIO). Cette unité est responsable du suivi des espèces et des habitats naturels. Elle gère et valorise les données naturalistes et instruit les dossiers réglementaires liés à la biodiversité. Elle anime également le Système d'Information sur la Nature et les Paysages (SINP-974 (Système d'Information de l'iNventaire du Patrimoine naturel de la Réunion)) régional et assure la gestion de la plateforme Borbonica. L'unité intervient aussi sur des thématiques transversales telles que les espèces exotiques envahissantes (EEE), la séquence ERC (Éviter – Réduire – Compenser) ou la diffusion des connaissances naturalistes. L'organisation interne de l'UBIO est présentée en annexe (Fig. 7.1).

L'organigramme interne montre une équipe composée de profils scientifiques, techniques et administratifs. Ces personnels travaillent de manière complémentaire pour répondre aux enjeux liés à la biodiversité du territoire. Mon stage s'inscrit dans cette dynamique, au sein du pôle

dédié aux données naturalistes. Il contribue à la structuration et à la modernisation des outils numériques utilisés par la DEAL.

Cette présentation de la DEAL et de son organisation permet de situer le contexte global de mon stage. Le chapitre suivant propose un **diagnostic RSE (Responsabilité Sociétale des Entreprises)** de la structure. Il vise à évaluer ses pratiques au regard des enjeux sociaux, environnementaux et organisationnels.

1.2. Diagnostic RSE

En raison des responsabilités qui lui sont confiées, la DEAL Réunion doit intégrer des considérations sociales, environnementales et organisationnelles dans l'ensemble de ses pratiques. Le diagnostic qui suit s'appuie sur la norme ISO 26000 et sur les principes de la RNE (Responsabilité Numérique des Entreprises), afin d'évaluer la manière dont la structure prend en compte ces enjeux.

1.2.1. Risques et impacts des activités

Impacts environnementaux

La DEAL n'engendre pas d'impacts industriels directs. Cependant, plusieurs de ses activités présentent des risques environnementaux indirects :

- un usage intensif d'outils numériques (bases de données naturalistes, SIG, plateformes SINP-974, Géonature, Borbonica (Plateforme régionale de diffusion des données naturalistes du SINP)), impliquant une consommation énergétique importante et une dépendance aux infrastructures serveurs ;
- le stockage et le traitement de volumes croissants de données, contribuant à l'empreinte carbone numérique et à la production de déchets électroniques liés au renouvellement du matériel ;
- des déplacements professionnels réguliers pour l'instruction de projets, le suivi de chantiers ou les missions de terrain, occasionnant des émissions de gaz à effet de serre.

Ces activités consomment de l'énergie, mobilisent des ressources non renouvelables et contribuent à l'empreinte environnementale globale du service.

Impacts sociétaux

Les missions de la DEAL comportent également plusieurs risques sociétaux :

- une influence directe sur les décisions d'aménagement (urbanisme, infrastructures, gestion des risques), pouvant affecter l'organisation du territoire et la qualité de vie des habitants.

- une responsabilité dans l'instruction des permis CITES et le suivi des espèces menacées, dont une mauvaise évaluation pourrait fragiliser le patrimoine naturel.
- la coordination des actions de lutte contre les espèces exotiques envahissantes (EEE), où un retard ou un défaut d'intervention peut entraîner une dégradation durable des écosystèmes.
- une forte dépendance aux partenaires (collectivités, OFB, Ifremer, PNRun, associations), rendant certaines actions vulnérables à des manques de coordination ou de moyens.

Ces risques peuvent affecter la cohérence des politiques publiques, la préservation des milieux naturels et l'équité territoriale.

Impacts sociaux

Le fonctionnement interne de la DEAL comporte également des risques sociaux. Ses équipes travaillent dans un environnement pluridisciplinaire soumis à des contraintes organisationnelles et réglementaires :

- une charge de travail variable selon les périodes, pouvant entraîner des tensions dans la répartition des tâches et la disponibilité des agents.
- des missions de terrain présentant des risques pour la santé et la sécurité (déplacements fréquents, conditions difficiles, chantiers, zones accidentées).
- un besoin constant de mise à jour des compétences (réglementations, outils numériques, SIG), créant un risque d'obsolescence professionnelle pour les agents non formés.
- un renouvellement régulier des stagiaires ou contractuels, pouvant fragiliser la continuité des projets ou l'organisation des équipes.

Ces éléments peuvent affecter la qualité de vie au travail, la sécurité en mission et la stabilité organisationnelle.

1.2.2. Stratégie globale en matière de RSE

Bien qu'elle agisse dans un cadre réglementaire strict, la DEAL cherche à renforcer la cohérence et la qualité de ses pratiques internes. Elle s'inscrit dans une dynamique de *pré-conformité active*, allant au-delà des obligations minimales en matière d'organisation, de gestion de l'information et de modernisation numérique. Cette démarche vise à améliorer la fiabilité des données environnementales, à harmoniser les outils utilisés par les services et à garantir une application rigoureuse des réglementations.

Sur le plan sociétal, la DEAL renforce ses méthodes de travail pour limiter les risques identifiés. Elle veille à la transparence dans l'instruction des projets d'aménagement, à sécuriser

les procédures liées aux permis CITES et à structurer davantage la coordination des actions de lutte contre les (EEE).

Cette dynamique repose également sur une coopération accrue avec les collectivités territoriales, les établissements publics et les associations naturalistes, afin de réduire les vulnérabilités liées à la gouvernance et à la circulation de l'information. Ces partenariats contribuent à une meilleure gestion des risques sociaux, environnementaux et sociétaux auxquels le territoire est confronté.

1.2.3. Conclusion personnelle

Ce diagnostic montre que la DEAL Réunion intègre progressivement les enjeux de la RSE et de la RNE au cœur de ses pratiques, malgré les contraintes propres à une administration publique. Cette approche illustre la volonté de la structure de moderniser son fonctionnement et de réduire ses vulnérabilités, tout en répondant aux attentes réglementaires et territoriales. Cette réflexion renforce l'idée que le numérique et l'organisation interne jouent un rôle essentiel dans la transition écologique et l'amélioration des politiques publiques.

2. Organisation du stage

2.1. Contexte du stage

Le SINP-974 est un dispositif collectif de mise en partage des données d'observations d'espèces sauvages sur l'île. Il a été mis en service à La Réunion en 2018, au travers de la plateforme Borbonica gérée conjointement par PNRUn et la DEAL.

Borbonica s'appuie sur un portail web, qui permet d'accéder aux différentes interfaces utilisateurs du SINP :

- Borbonica obs : consultation cartographique des données du SINP. Il s'agit d'une interface web basée sur le plugin Lizmap et reposant sur une base de données PostgreSQL. Cette plateforme offre un accès aux données selon différents niveaux et profils (grand public, experts, validateurs, etc.) ;
- Borbonica atlas : consultation des fiches espèces disponibles dans le SINP, sous forme de synthèse des principales informations (observations, documents, photos, etc.). Il s'agit d'un outil issu de la solution open source GéoNature ;
- Borbonica stats : module statistique qui permet de consulter des tableaux de bord chiffrés sur le contenu de Borbonica (données disponibles, principaux usages, etc.). Ce module s'appuie sur un projet Lizmap, exploitant des données de la base PostgreSQL.

Afin de pérenniser le SINP 974, un projet a été lancé en 2023, avec pour principaux objectifs la modernisation et la simplification du fonctionnement du système. Plusieurs nouvelles fonctionnalités vont être ajoutées, et certaines briques logicielles seront remplacées par de nouvelles solutions plus modernes et plus simples à utiliser et à maintenir. C'est le cas notamment de la solution Géonature qui est en cours de déploiement pour remplacer le périmètre de Borbonica obs.

2.2. Objectifs du stage

À la suite de la modernisation engagée pour le SINP-974 et du déploiement progressif de GeoNature au sein de la DEAL, le stage avait pour finalité d'apporter un appui opérationnel au renforcement de la chaîne d'acquisition et de gestion des données naturalistes. L'objectif principal était de faciliter l'intégration de données externes dans GeoNature et, plus largement, d'enrichir les informations accessibles via la plateforme régionale Borbonica.

Dans ce cadre, plusieurs objectifs spécifiques ont été définis :

- développer un module externe GeoNature interfacé avec le système d'information Quadrige de l'Ifremer, afin d'automatiser la récupération, le filtrage et la préparation des données issues du milieu marin ;
- poser les fondations techniques nécessaires à une future intégration des données produites par l'application Pl@ntNet (Application mobile de reconnaissance des plantes par photo), en réalisant un client Python et un premier outil de structuration des résultats ;
- documenter les procédures d'appel aux API et proposer une organisation des développements garantissant leur maintenabilité et leur évolution dans le temps.

En complément du module Quadrige, une seconde orientation du stage concernait la préparation d'un futur connecteur GeoNature dédié aux données issues de Pl@ntNet. Pl@ntNet produit chaque jour un volume important d'observations naturalistes géolocalisées, particulièrement pertinent pour le suivi des EEE végétales et des dynamiques de végétation. Afin d'explorer la faisabilité de cette intégration, un travail préliminaire a consisté à :

- étudier la documentation de l'API Pl@ntNet v3,
- concevoir un client Python robuste, capable d'interroger l'API selon plusieurs critères (taxon, polygone GeoJSON, plage temporelle),
- structurer les données extraites au format *Darwin Core* afin d'assurer une compatibilité immédiate avec GeoNature et le SINP-974.

Ce travail prépare le développement d'un parser complet destiné au module `api2gn` de GeoNature.

Ces objectifs s'inscrivent pleinement dans la stratégie portée conjointement par la DEAL Réunion et le PNRun, visant à fiabiliser et à harmoniser les données du SINP-974 tout en renforçant la représentation de thématiques encore peu renseignées, comme le milieu marin. Ils ont guidé l'ensemble des actions menées durant le stage et constituent le cadre des travaux détaillés dans les sections suivantes.

2.3. Méthodologie de travail

La méthodologie adoptée au cours du stage s'est construite de manière progressive et pragmatique. Elle a été définie en tenant compte des contraintes techniques du projet, du cadre institutionnel de la DEAL et de l'organisation particulière de l'équipe. Le projet étant mené uniquement à deux personnes — mon maître de stage, Rémi Bouilly, et moi-même — il n'était pas pertinent de mettre en place une méthodologie formelle lourde. Le cadre de travail retenu s'inspire néanmoins des principes des méthodes agiles, tout en restant volontairement souple.

Le développement du module Quadrige et des outils associés a ainsi été conduit selon une démarche itérative et incrémentale. Le travail était organisé en cycles courts, assimilables à des sprints d'une durée moyenne d'environ une semaine. Chaque sprint correspondait à un ensemble cohérent de fonctionnalités, définies en amont, puis développées, testées et ajustées avant d'engager le cycle suivant. Cette organisation permettait d'obtenir régulièrement des résultats fonctionnels, même partiels, tout en identifiant rapidement les difficultés techniques ou fonctionnelles.

Dans ce contexte de travail en effectif réduit, il n'était ni nécessaire ni utile de mettre en place des rituels agiles classiques tels que les daily meetings ou les revues de sprint formelles. En revanche, des points réguliers étaient organisés une à deux fois par semaine avec mon maître de stage. Ces échanges permettaient de faire un état d'avancement, de valider les choix techniques effectués et d'ajuster les priorités à court terme en fonction des besoins métier. Ils ont joué un rôle central dans le bon déroulement du projet, en assurant une compréhension partagée des objectifs tout en me laissant une large autonomie dans la mise en œuvre.

Le cadrage initial du stage s'est appuyé sur deux cahiers des charges distincts, fournis par mon maître de stage. Le premier concernait le développement du module externe GeoNature interfacé avec Quadrige, tandis que le second portait sur la conception d'un parser dédié aux données issues de Pl@ntNet et à leur structuration au format Darwin Core. Ces documents définissaient le périmètre fonctionnel attendu, les principales contraintes techniques ainsi que les livrables envisagés. Ils ont servi de fil conducteur tout au long du stage, tout en restant suffisamment flexibles pour intégrer des ajustements liés aux contraintes rencontrées ou à l'évolution du projet.

Afin de suivre l'avancement global du projet et de visualiser les grandes phases du stage, un diagramme de Gantt a été utilisé. Il a permis de planifier les principales étapes, depuis la prise en main de l'environnement GeoNature jusqu'aux phases de tests, de documentation et de préparation des livrables. Cet outil offrait une vision synthétique du projet et facilitait l'anticipation des périodes plus sensibles, notamment lors des phases de migration des serveurs ou d'attente d'accès aux environnements distants.

À un niveau plus opérationnel, le travail quotidien était organisé sous forme de tâches unitaires, réparties selon un statut simple : à faire, en cours ou terminées. Cette organisation, inspirée des tableaux de type Kanban, permettait de visualiser rapidement l'état d'avancement, de prioriser les actions et de conserver une trace claire du travail réalisé. Elle s'est révélée particulièrement adaptée à un projet individuel encadré, en apportant une structure légère sans alourdir inutilement le processus.

La gestion du code source reposait intégralement sur l'utilisation de Git. L'ensemble des développements a été versionné dès les premières phases du projet, aussi bien pour le module Quadrige que pour les outils liés à Pl@ntNet. Les évolutions étaient enregistrées de manière régulière, avec des messages de commit explicites. Cette pratique a permis de sécuriser les

développements, de tester différentes approches sans risque de perte de code et de faciliter le transfert vers les serveurs de la DEAL une fois l'accès ouvert. Dans un contexte où l'accès aux environnements distants était limité et parfois instable, Git a également servi de principal moyen de synchronisation entre l'environnement local et l'environnement distant.

Le développement a d'abord été mené intégralement en local, en l'absence d'un accès immédiat aux serveurs du SINP-974. Cette situation a nécessité la mise en place d'un environnement de travail stable, capable de reproduire au mieux les conditions de production. La séparation stricte entre le backend Python/Flask et le frontend Angular, conforme à l'architecture modulaire de GeoNature, a facilité cette organisation et permis de développer chaque composante de manière indépendante.

Les contraintes liées au proxy ministériel et aux versions spécifiques des outils techniques ont également nécessité une attention particulière à la configuration de l'environnement. L'utilisation de pyenv a permis de stabiliser la version de Python utilisée, condition indispensable pour garantir la compatibilité avec GeoNature et ses dépendances.

Enfin, l'ensemble du travail s'est appuyé sur une démarche de validation progressive. Les appels à l'API Quadrige ont d'abord été testés à l'aide de scripts Python indépendants, avant leur intégration dans le module GeoNature. De la même manière, les premières interfaces Angular ont été testées en local afin de vérifier leur cohérence fonctionnelle et leur ergonomie avant toute intégration plus large.

Cette méthodologie, mêlant principes agiles, rigueur technique et adaptation constante au contexte, a permis d'assurer la robustesse des développements malgré un environnement parfois constraint. Elle a également facilité l'anticipation des étapes futures d'intégration et de reprise du projet par la DEAL ou ses partenaires.

3. Introduction à GeoNature

3.1. Principe général de GeoNature

GeoNature est une application web dédiée à la gestion, la centralisation et la valorisation des données naturalistes. Développé initialement en 2010 puis entièrement refondu en 2017, le projet est aujourd’hui maintenu par le PNRun. L’application repose sur une architecture moderne combinant un backend Python/Flask et un frontend Angular, ce qui lui permet d’assurer à la fois la saisie, la consultation, la validation et la restitution des données.

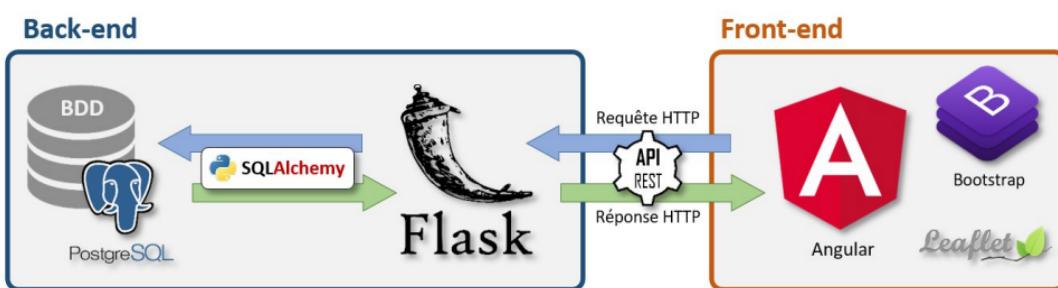


Figure 3.1: Architecture générale du backend et du frontend de GeoNature

Le fonctionnement de GeoNature est modulaire : un noyau applicatif fournit les briques communes (API, référentiels, schéma de synthèse), et différents modules viennent étendre les fonctionnalités selon les besoins (Occtax, Occhab, Validation, Import, Export, etc.). La documentation officielle détaille l’ensemble de cette architecture, ([documentation officielle](#)) ainsi que les sources du projet, disponibles sur [GitHub](#).

3.2. Architecture technique

GeoNature combine une partie serveur (backend) et une partie cliente (frontend) qui dialoguent via une API REST.

Le backend est développé en Python à l'aide du framework Flask. Il assure l'ensemble des traitements métiers, l'accès aux données et la gestion des opérations spatiales via PostgreSQL et PostGIS. La gestion des utilisateurs et de leurs permissions repose sur UsersHub, qui fournit l'authentification et le contrôle des accès. L'API exposée par le backend constitue le point central de communication avec l'interface web.

Le frontend, développé en Angular, constitue l'interface visible par l'utilisateur. Il interroge exclusivement l'API du backend pour afficher les formulaires, les cartes, les graphiques et les données attributaires. Des bibliothèques comme Leaflet ou Bootstrap renforcent les fonctionna-

lités cartographiques et l’ergonomie générale. Cette séparation nette entre backend et frontend garantit la stabilité et la modularité du système, tout en facilitant les évolutions futures.

3.3. Organisation interne et modules

L’application est structurée autour de modules fonctionnels s’appuyant tous sur les mêmes référentiels : taxonomie (TaxHub), nomenclatures, utilisateurs (UsersHub) et schéma de synthèse. Les modules principaux — Occtax pour les observations, Occhab pour les habitats, ou encore la Validation — s’intègrent directement au cœur applicatif.

Chaque module possède son propre schéma de base de données, son API et ses composants Angular. Cette organisation modulaire permet de faire évoluer GeoNature, d’ajouter de nouveaux protocoles d’acquisition ou de développer des extensions externes. La documentation décrit précisément les bonnes pratiques et l’architecture à respecter pour développer un [module GeoNature externe](#).

4. Développement du module d'import Quadrige

4.1. Contexte du projet

Le SINP-974, déployé à La Réunion à travers la plateforme Borbonica, constitue le dispositif régional de référence pour la centralisation, la gestion et la diffusion des données naturalistes. Depuis sa mise en service, il a permis de structurer et de valoriser un volume important d'observations, issues de sources multiples et couvrant un large éventail de taxons.

La dynamique de peuplement de la base reflète toutefois les modalités historiques et opérationnelles de production des données naturalistes sur le territoire. Les observations actuellement intégrées concernent majoritairement les milieux terrestres et dulçaquicoles, qui bénéficient de réseaux d'acteurs structurés, de protocoles d'acquisition largement diffusés et d'outils de saisie directement compatibles avec le SINP.

À l'inverse, les données issues du milieu marin demeurent plus faiblement représentées. Les indicateurs de contenu disponibles en 2024 montrent que les observations relatives aux taxons marins ne constituent qu'une part marginale des données référencées. Cette situation ne traduit pas une absence de connaissances ou de suivis en milieu marin, mais résulte principalement de la complexité des chaînes d'acquisition, de structuration et de diffusion propres aux données marines, historiquement gérées au sein de systèmes d'information spécialisés.

Dans ce contexte, l'enjeu principal n'était pas de refondre les outils existants, mais de renforcer progressivement la représentation des données marines au sein du SINP-974, en s'appuyant sur des sources de données déjà structurées, pérennes et reconnues institutionnellement. Parmi celles-ci, le système d'information Quadrige, maintenu par l'Ifremer, occupe une place centrale. Quadrige regroupe un volume important de données environnementales et biologiques collectées en milieu marin, issues de programmes de surveillance, d'observation et de recherche conduits sur le long terme.

L'ouverture récente d'une API (Interface de Programmation de l'Application) GraphQL par l'Ifremer constitue à ce titre une évolution majeure. Elle permet une interrogation directe, fine et structurée des données de Quadrige, en ciblant précisément les programmes, les zones géographiques, les périodes temporelles et les types d'observations d'intérêt. Cette évolution ouvre la voie à une exploitation plus systématique de ces données par des acteurs institutionnels tels que la DEAL, et à leur mobilisation progressive dans le cadre du SINP-974.

C'est dans ce cadre que s'inscrit le projet développé durant le stage. L'objectif n'était pas de réaliser une intégration directe et entièrement automatisée des données Quadrige dans la base

GeoNature, mais de concevoir un module externe permettant d'explorer, d'extraire et de préparer ces données de manière structurée, traçable et reproductible. Le module développé interroge l'API Quadrige afin d'identifier les programmes pertinents pour le territoire réunionnais, puis permet d'en extraire les observations associées selon des critères définis par l'utilisateur.

Les résultats de ces extractions ne prennent pas la forme de fichiers immédiatement importables dans GeoNature. Pour chaque programme sélectionné, le module génère une archive compressée contenant plusieurs éléments complémentaires : un fichier CSV brut correspondant aux données extraites, un fichier JSON décrivant précisément les filtres appliqués lors de l'appel à l'API, ainsi qu'un fichier README documentant le déroulement de l'export et les éventuelles anomalies rencontrées. Le module conserve également un accès aux trois derniers exports réalisés afin d'en faciliter la consultation et la réutilisation.

En complément de ces archives, le module expose également des liens directs vers les fichiers CSV générés lors de la phase d'extraction des programmes. Deux niveaux de fichiers sont distingués : – un CSV brut issu directement de la réponse de l'API Quadrige, contenant l'ensemble des instances des programmes dont au moins une occurrence est localisée sur le territoire ciblé ; – un CSV filtré géographiquement, obtenu par un traitement *a posteriori* à l'aide de la bibliothèque pandas, permettant d'exclure les instances dont le code de monitoring location ne correspond pas au territoire d'intérêt (par exemple, codes ne commençant pas par le préfixe « 126- » pour La Réunion).

Ce filtrage complémentaire est nécessaire car, lors de l'extraction des programmes, la requête adressée à l'API Quadrige sélectionne l'ensemble des programmes possédant au moins une instance sur le territoire demandé, mais retourne également les autres instances associées à ces programmes, y compris celles situées hors du périmètre géographique ciblé.

Cette approche intermédiaire répond à plusieurs objectifs. Elle garantit d'une part la traçabilité complète des extractions réalisées depuis Quadrige, en conservant une description explicite des paramètres et traitements appliqués. Elle offre d'autre part une souplesse d'usage, en laissant aux administrateurs et gestionnaires de données la possibilité de contrôler, d'analyser et, le cas échéant, d'adapter les fichiers produits avant toute intégration dans GeoNature ou Borbonica. Le module s'inscrit ainsi comme une brique préparatoire, destinée à sécuriser et à faciliter l'intégration future des données marines dans le SINP-974.

Ce développement s'inscrit pleinement dans la dynamique d'amélioration continue portée par la DEAL Réunion et ses partenaires. Il vise à renforcer progressivement la place des données issues du milieu marin au sein du système régional, tout en respectant les contraintes techniques, méthodologiques et organisationnelles propres aux outils existants.

4.2. Périmètre fonctionnel

Le module d'import devait couvrir l'ensemble de la chaîne d'acquisition : de la découverte des programmes Quadrige jusqu'à la production d'un fichier structuré pour GeoNature. La première

étape consistait à interroger l'API en mode authentifié afin d'obtenir la liste des programmes disponibles pour un utilisateur donné. Un filtrage automatique sur un périmètre géographique — principalement La Réunion, mais extensible à d'autres territoires comme les Îles Éparses — permettait d'isoler les programmes pertinents. Une interface dédiée intégrée à GeoNature offrait ensuite la possibilité de rechercher des programmes, d'affiner l'affichage par mots-clés et de sélectionner ceux à importer.

Une fois les programmes choisis, l'utilisateur pouvait définir les filtres à appliquer aux données (la période d'intérêt, les champs souhaités, ou encore la reprise des stations déjà extraites). Le module interrogeait alors [l'API Quadrige](#) pour récupérer les observations correspondantes [Tutoriels vidéo](#). Seuls les champs utiles au modèle SINP-974 étaient extraits : identifiants des programmes et stations, localisation géographique, taxon observé, date, ainsi que les métadonnées essentielles (auteur, organisme, méthode d'acquisition, etc.). Une transformation était appliquée pour obtenir une structure compatible avec les mécanismes d'import de GeoNature.

Enfin, le module produisait un fichier CSV intermédiaire, destiné à être importé via l'infrastructure existante de GeoNature. Chaque opération d'import était consignée dans un historique affiché dans un second onglet, permettant de suivre les actions réalisées, leur date, leur statut et les éventuelles erreurs rencontrées. L'ensemble du module était réservé aux administrateurs, conformément aux pratiques habituelles de contrôle des imports dans GeoNature.

4.3. Spécifications techniques

Techniquement, le module repose sur une architecture conforme à celle de [Géonature](#), dont le code source est accessible sur son [dépôt GitHub](#). Le **backend**, développé en Python avec le framework Flask, se charge de communiquer avec l'API Quadrige, d'effectuer les transformations nécessaires sur les données et de générer les fichiers d'export. Le **frontend**, construit en Angular et Bootstrap, s'intègre à l'interface existante de GeoNature et fournit les pages de sélection des programmes ainsi que l'historique des imports.

Les échanges avec Quadrige reposent sur l'API GraphQL récemment mise en place par l'Ifremer, permettant de sélectionner précisément les champs nécessaires et de limiter le volume des données transférées. L'authentification s'appuie sur un token fourni par l'Ifremer et configurable directement dans les paramètres du module. Les autres réglages comprennent l'URL de l'API, le périmètre géographique par défaut, ainsi que le mapping entre les champs Quadrige et les champs attendus par le schéma de synthèse de GeoNature.

Un soin particulier a été apporté à la gestion des erreurs : vérification des réponses HTTP, détection des timeouts, contrôle de la validité des données, et affichage de messages explicites à l'utilisateur. Toutes les actions — appels API, transformations, exports — sont consignées dans les logs du module, conformément aux pratiques de GeoNature.

Le module ne se limite pas à la production de fichiers destinés à un import ultérieur. Il met également à disposition de l'utilisateur les résultats intermédiaires de l'extraction des pro-

grammes Quadrige, sous forme de fichiers CSV consultables. Ces exports intermédiaires visent à faciliter la compréhension du périmètre réellement couvert par chaque programme et à permettre un contrôle visuel préalable avant toute extraction de données détaillées.

4.4. Contraintes, dépendances et livrables

La réussite du projet dépendait principalement de l'accessibilité de l'API Quadrige, de la stabilité de ses services, et de la disponibilité d'une documentation actualisée. L'intégration dans GeoNature nécessitait également de respecter la structure modulaire du noyau applicatif et les contraintes du modèle SINP-974.

Les livrables attendus comprenaient le code source du module, un fichier de configuration, une documentation à destination des administrateurs (installation, configuration, maintenance) ainsi qu'un guide d'utilisation orienté métier.

Plusieurs évolutions ont été envisagées : automatisation des imports périodiques, gestion des imports incrémentaux, intégration à la gestion des droits de GeoNature et, à plus long terme, publication du module dans le catalogue officiel des extensions GeoNature.

5. Travaux réalisés et compétences acquises

5.1. Travaux réalisés

Les travaux réalisés au cours du stage se sont concentrés principalement sur le développement d'un module externe pour GeoNature, capable d'interagir avec le système d'information Quadrige. La première étape a consisté à concevoir un client Python permettant de tester l'envoi de requêtes GraphQL auprès de l'API Quadrige, de récupérer les données et de gérer l'authentification via un jeton.

Une fois les fonctions d'interrogation terminées, je suis passé au développement d'un prototype local du frontend Angular. Il se compose de quatre composants principaux. Le premier, `program-list`, affiche la liste des programmes extraits en fonction d'une localisation (par exemple « 126- » comme préfixe pour La Réunion). C'est depuis ce composant principal que s'effectuent les échanges backend/frontend.

Le composant `program-filter` permet de définir un filtre incluant une localisation et un nom de filtre. Les critères sont sauvegardés dans le fichier `memory/last_filter.json`. Ce filtre sert à extraire les programmes selon des règles bien définies. La requête GraphQL utilisée est `executeProgramExtraction`, qui comporte de nombreux paramètres optionnels, mais dont seuls ceux relatifs à la localisation nous intéressent. L'objectif est d'extraire tout programme comportant au moins une instance dans la zone spécifiée.

Un bouton « Extraire les programmes » déclenche la fonction d'extraction, qui envoie la requête GraphQL à Quadrige et récupère les programmes correspondant au filtre. Un message d'état informe l'utilisateur de l'avancement (« extraction en cours », « programmes extraits ») et, en cas d'erreur, un message approprié s'affiche.

Lorsque l'extraction est terminée, un fichier CSV est produit et stocké dans `memory/csv_brut_suffixe`, associé au filtre utilisé. Il contient toutes les instances des programmes ayant au moins une occurrence dans la zone filtrée.

Le fichier CSV généré à l'issue de cette première extraction contient l'ensemble des instances associées aux programmes sélectionnés, y compris celles situées en dehors du périmètre géographique d'intérêt. Cette situation résulte du fonctionnement même de l'API Quadrige, qui raisonne à l'échelle des programmes et non des stations individuelles.

Afin de garantir la cohérence géographique des données exploitées, un filtrage a posteriori a été mis en œuvre à l'aide de la bibliothèque pandas. Ce traitement consiste à ne conserver que les instances dont le code de monitoring location correspond au territoire ciblé, identifié par un

préfixe spécifique (par exemple « 126- » pour La Réunion).

Ce filtrage permet de produire un second fichier CSV, strictement limité aux stations pertinentes, utilisé à la fois pour l'affichage dans l'interface utilisateur et comme base pour les extractions de données ultérieures.

Par la suite, je n'ai besoin que du code du programme pour extraire les données recherchées.

Une fois les programmes extraits, ils sont affichés dans `program-list` avec leur nom, leurs dates de début et de fin et le(s) responsable(s). L'utilisateur peut sélectionner les programmes via une barre de recherche, des cases cochables et une option « tout cocher ».

Un composant `data-filter` permet ensuite de filtrer les données des programmes sélectionnés selon plusieurs critères, notamment les dates ou la localisation (automatiquement reprise depuis le dernier filtre créé).

En parallèle, j'ai conçu l'architecture du backend, fondée sur un *blueprint* Flask dédié, afin d'assurer l'authentification auprès de l'API Quadrige, l'interrogation des programmes disponibles et la structuration des données nécessaires à un futur import dans GeoNature. Une attention particulière a été portée à la construction des requêtes GraphQL, afin de cibler les informations utiles tout en limitant le volume des échanges.

Le développement du frontend a permis de mettre en place une première interface d'exploration des programmes Quadrige directement dans GeoNature. Bien qu'encore partielle, cette interface constitue une base solide pour les futures fonctionnalités de sélection, de filtrage et d'extraction. Ces travaux ont donné lieu à plusieurs phases de tests, notamment via PyTest pour le backend et Cypress pour l'interface utilisateur, afin de valider les mécanismes centraux avant leur intégration complète.

Travaux parallèles sur l'intégration de Pl@ntNet En parallèle du développement principal consacré à Quadrige, j'ai également initié un travail structurant autour de l'intégration des données Pl@ntNet dans GeoNature. L'objectif était double : d'une part valider la possibilité technique d'interroger l'API Pl@ntNet depuis un service Python, et d'autre part poser les bases d'un parseur Darwin Core utilisable par un module GeoNature, selon une logique similaire à celle mise en place pour Quadrige.

Pour cela, j'ai développé un client Python générique capable de :

- envoyer des requêtes POST vers l'endpoint `dwc/occurrence/search` de l'API Pl@ntNet ;
- gérer dynamiquement les paramètres taxonomiques, spatiaux (bbox ou polygone GeoJSON) et temporels ;
- enregistrer les réponses brutes au format JSON ;
- transformer ces réponses en un CSV conforme au standard Darwin Core (DwC), en s'appuyant sur des fichiers de configuration modulaires.

Une attention particulière a été portée à la normalisation des données :

- homogénéisation des dates au format ISO-8601 ;
- nettoyage et standardisation des coordonnées ainsi que du datum géodésique ;
- génération d'identifiants stables (`occurrenceID`) ;
- mappage des vocabulaires Pl@ntNet vers ceux utilisés dans GBIF (`basisOfRecord`) ;
- sérialisation propre des champs complexes (listes, objets JSON).

Ce travail a conduit à la mise en place d'un pipeline complet :

API → JSON brut → CSV Darwin Core normalisé

entièrement automatisé et adaptable à d'autres territoires ou taxons via un simple fichier de configuration.

Ce client constitue aujourd'hui la base technique du parser Pl@ntNet destiné au module API2gn, dont l'intégration a nécessité plusieurs ajustements dans les migrations SQL et dans la structure interne du module.

Enfin, une documentation détaillée a été rédigée tout au long du projet. Elle décrit les procédures d'appel aux API, les choix techniques effectués ainsi que les étapes d'installation et de maintenance du module. Ce travail de cAPITalisation est essentiel pour permettre la reprise du projet par la DEAL ou ses partenaires.

5.2. Difficultés rencontrées et solutions apportées

Le déroulement du stage a été marqué par un ensemble de contraintes techniques qui ont largement influencé l'organisation du travail. L'absence d'une installation locale stable de GeoNature a constitué l'un des principaux obstacles : les incompatibilités entre les versions de Debian, les erreurs liées au serveur applicatif et les difficultés rencontrées avec Docker ont rendu nécessaire une approche alternative, centrée sur une instance de travail personnalisée. Cette situation a été accentuée par les limitations du réseau de la DEAL, où la présence d'un proxy strict perturbait l'utilisation de nombreux outils de développement. La mise en place d'un mécanisme dynamique de gestion du proxy a progressivement permis de stabiliser l'environnement.

La migration des serveurs du SINP-974 vers un nouvel hébergeur a également retardé l'accès à un environnement de test réaliste. Pendant plusieurs semaines, le développement a dû être poursuivi sans possibilité de déployer ou de tester directement sur l'environnement cible. Une fois l'accès ouvert via une machine virtuelle bastion, les premières manipulations ont été freinées par les restrictions imposées par l'outil d'accès à distance. L'usage de Git comme canal de transfert a alors été déterminant pour poursuivre les tests et ajuster le module.

Ces difficultés ont nécessité une grande capacité d’adaptation et ont conduit à affiner l’organisation du projet : clarification des environnements, adoption d’outils de gestion de version adaptés, multiplication des tests exploratoires et documentation progressive pour sécuriser les phases suivantes d’intégration.

Difficultés spécifiques au travail sur Pl@ntNet Le développement du client Pl@ntNet a également présenté plusieurs défis techniques. L’API renvoie une structure JSON très hétérogène selon les taxons, les médias associés ou les métadonnées disponibles, ce qui a nécessité la création d’un parseur flexible, capable d’absorber des variations importantes de schéma sans échouer.

Par ailleurs, le standard Darwin Core impose un modèle strict qui ne correspond pas directement aux champs fournis par Pl@ntNet. Il a donc fallu définir un mapping robuste et configurable pour garantir la compatibilité avec GeoNature et Borbonica.

Enfin, l’intégration du parser dans le module `api2gn` a nécessité de modifier certaines migrations SQL existantes, qui empêchaient l’installation correcte du module. Ces ajustements feront l’objet d’un retour formalisé aux développeurs officiels de GeoNature afin d’améliorer la compatibilité du module avec des sources de données diversifiées.

5.3. Compétences acquises

Ce stage m’a permis de développer une maîtrise approfondie de l’écosystème technique de GeoNature, tant sur le plan backend (Python, Flask, SQLAlchemy, Marshmallow) que frontend (Angular, TypeScript). J’ai acquis une expérience concrète dans l’utilisation d’API REST et GraphQL, dans la mise en place de tests unitaires et fonctionnels, ainsi que dans la gestion rigoureuse d’environnements soumis à des contraintes fortes.

Ce travail m’a également permis de me familiariser avec les pratiques du développement open source, notamment via l’utilisation de forks GitHub, la gestion des issues et la rédaction de documentation technique destinée à des utilisateurs externes.

Au-delà des aspects techniques, cette expérience m’a amené à travailler en autonomie tout en collaborant étroitement avec des acteurs variés : agents en charge des données, administrateurs réseau, prestataires et agents du Parc national. J’ai pu renforcer mes compétences en organisation de projet, en documentation technique et en adaptation à des environnements complexes, autant de dimensions essentielles dans un contexte d’ingénierie logicielle appliquée aux enjeux environnementaux.

6. Perspectives

Les travaux réalisés au cours de ce stage ont permis de poser des bases techniques solides pour l'intégration de données externes au sein de Géonature, en particulier pour les données issues du milieu marin via Quadrige, ainsi que pour les données participatives produites par Pl@ntNet. Plusieurs perspectives d'évolution ont été identifiées, tant sur le plan fonctionnel que technique. Elles visent à renforcer l'ergonomie des outils développés, à améliorer leur intégration dans l'écosystème GeoNature et à faciliter leur appropriation par les administrateurs et gestionnaires de données.

6.1. Perspectives d'évolution du module d'import Quadrige

Le module d'import Quadrige développé durant le stage constitue une première brique fonctionnelle permettant d'explorer, d'extraire et de préparer des données marines issues du système d'information Quadrige. Plusieurs axes d'amélioration peuvent être envisagés afin d'en accroître la maturité et de faciliter son intégration opérationnelle à long terme.

6.1.1. Amélioration de l'ergonomie et de l'intégration visuelle

Une première évolution concernerait l'intégration visuelle du module au sein de l'interface GeoNature. L'ajout d'un pictogramme dédié dans la barre de navigation ou dans la liste des modules permettrait d'identifier plus clairement le module Quadrige parmi les autres extensions disponibles.

Ce pictogramme contribuerait à améliorer l'expérience utilisateur en rendant le module plus visible et plus intuitif pour les administrateurs amenés à l'utiliser ponctuellement. Il faciliterait également son appropriation par de nouveaux utilisateurs, notamment dans un contexte de reprise du projet ou de déploiement sur d'autres territoires.

6.1.2. Génération directe de fichiers au format GeoNature

À l'heure actuelle, le module produit des fichiers intermédiaires (CSV bruts, CSV filtrés, fichiers JSON de paramètres) destinés à être analysés et, le cas échéant, retravaillés avant import dans GeoNature. Cette approche garantit une forte traçabilité, mais implique encore une étape manuelle avant l'intégration finale.

Une évolution naturelle du module consisterait à ajouter une étape optionnelle de transformation automatique des données extraites vers un format directement compatible avec les mécanismes d'import de GeoNature. Cette étape pourrait s'appuyer sur le schéma de synthèse et sur les règles existantes des modules d'import afin de produire un fichier immédiatement prêt à être injecté dans le système.

Cette fonctionnalité pourrait être activée de manière conditionnelle, laissant le choix à l'administrateur entre une extraction préparatoire avec contrôle manuel ou une extraction complète incluant la génération d'un fichier directement importable.

6.1.3. Centralisation de la configuration côté backend

Au cours du développement, certaines difficultés ont été identifiées dans la gestion de la configuration du module, notamment concernant les localisations proposées lors du filtrage des programmes et les champs disponibles pour l'extraction. Une partie de ces paramètres est actuellement définie directement dans le frontend Angular, ce qui limite la flexibilité du module et complique son évolution.

Une perspective importante consisterait à centraliser l'ensemble de ces paramètres dans un fichier de configuration côté backend. Le module expose déjà un endpoint permettant d'accéder au contenu de ce fichier, mais son exploitation n'a pas pu être finalisée durant le stage par manque de temps.

Une refonte partielle du frontend permettrait ainsi de charger dynamiquement les localisations disponibles et les champs d'extraction depuis le backend, d'éviter les valeurs codées en dur dans l'interface et de faciliter l'adaptation du module à d'autres territoires sans modification du code frontend. Cette approche renforcerait la maintenabilité globale du module et réduirait les risques d'incohérence entre backend et frontend.

6.1.4. Vers une automatisation partielle des imports

À plus long terme, le module pourrait évoluer vers une automatisation partielle ou contrôlée des imports, par exemple via des imports périodiques planifiés, des imports incrémentaux basés sur les dates de mise à jour des données, ou des déclenchements manuels avec validation préalable.

De telles évolutions nécessiteraient toutefois une réflexion approfondie sur la gestion des doublons, la validation scientifique des données et la traçabilité des mises à jour, afin de rester cohérentes avec les exigences du SINP-974 et les pratiques de contrôle des données naturalistes.

6.2. Perspectives autour du parser Pl@ntNet et du module api2gn

Les travaux menés sur l'intégration des données Pl@ntNet ont permis de valider la faisabilité technique d'un pipeline complet, depuis l'interrogation de l'API jusqu'à la production de fichiers conformes au standard Darwin Core. Toutefois, le parser développé repose actuellement sur des fichiers de configuration manipulés côté backend.

6.2.1. Ajout d'une interface d'administration dédiée

Une perspective majeure consisterait à proposer une interface frontend permettant aux administrateurs de paramétriser le parser Pl@ntNet sans recourir à la modification de fichiers de configuration. Cette interface pourrait offrir la possibilité de définir les taxons ciblés, les périmètres géographiques (boîtes englobantes ou polygones), les plages temporelles, ainsi que les

paramètres de normalisation et de filtrage.

Une telle évolution améliorerait significativement l'ergonomie du dispositif et faciliterait son utilisation par des profils non techniques. Elle permettrait également d'harmoniser l'expérience utilisateur avec celle du module Quadrige, en proposant une logique d'interaction cohérente au sein de GeoNature.

6.2.2. Contraintes liées au module api2gn

Dans l'état actuel, cette évolution n'est toutefois pas immédiatement réalisable. Le module api2gn est un module officiel de GeoNature, dont l'architecture n'a pas été conçue à l'origine pour intégrer des interfaces frontend spécifiques à chaque source de données.

Toute extension de ce type nécessiterait soit une évolution structurelle du module api2gn lui-même, soit la création d'un module externe complémentaire venant piloter le parser Pl@ntNet. Cette réflexion dépasse le cadre du stage, mais les travaux réalisés constituent une base technique solide pour initier ce type d'évolution à l'avenir.

6.2.3. Vers une intégration renforcée des données participatives

À plus long terme, l'intégration des données issues de Pl@ntNet dans GeoNature pourrait contribuer à enrichir significativement les données naturalistes disponibles, en particulier pour le suivi des espèces exotiques envahissantes végétales.

L'automatisation du pipeline API → Darwin Core → GeoNature, couplée à des mécanismes de filtrage et de validation adaptés, offrirait un levier intéressant pour exploiter des volumes importants de données participatives tout en respectant les exigences de qualité et de traçabilité du SINP-974.

6.3. Ouvertures transversales

Au-delà des deux modules développés, les travaux réalisés ouvrent des perspectives plus larges sur l'architecture des imports dans GeoNature. L'approche retenue, fondée sur des modules externes, une forte traçabilité et une séparation claire entre extraction, préparation et import, pourrait être généralisée à d'autres sources de données environnementales.

Cette logique modulaire facilite la mutualisation des développements, la reprise des projets par différents acteurs et l'adaptation des outils aux spécificités territoriales. Elle s'inscrit pleinement dans la stratégie de modernisation du SINP-974 et dans les objectifs de fiabilisation et d'harmonisation des données naturalistes portés par la DEAL Réunion et ses partenaires.

7. Conclusion

Bibliographie

Annexes

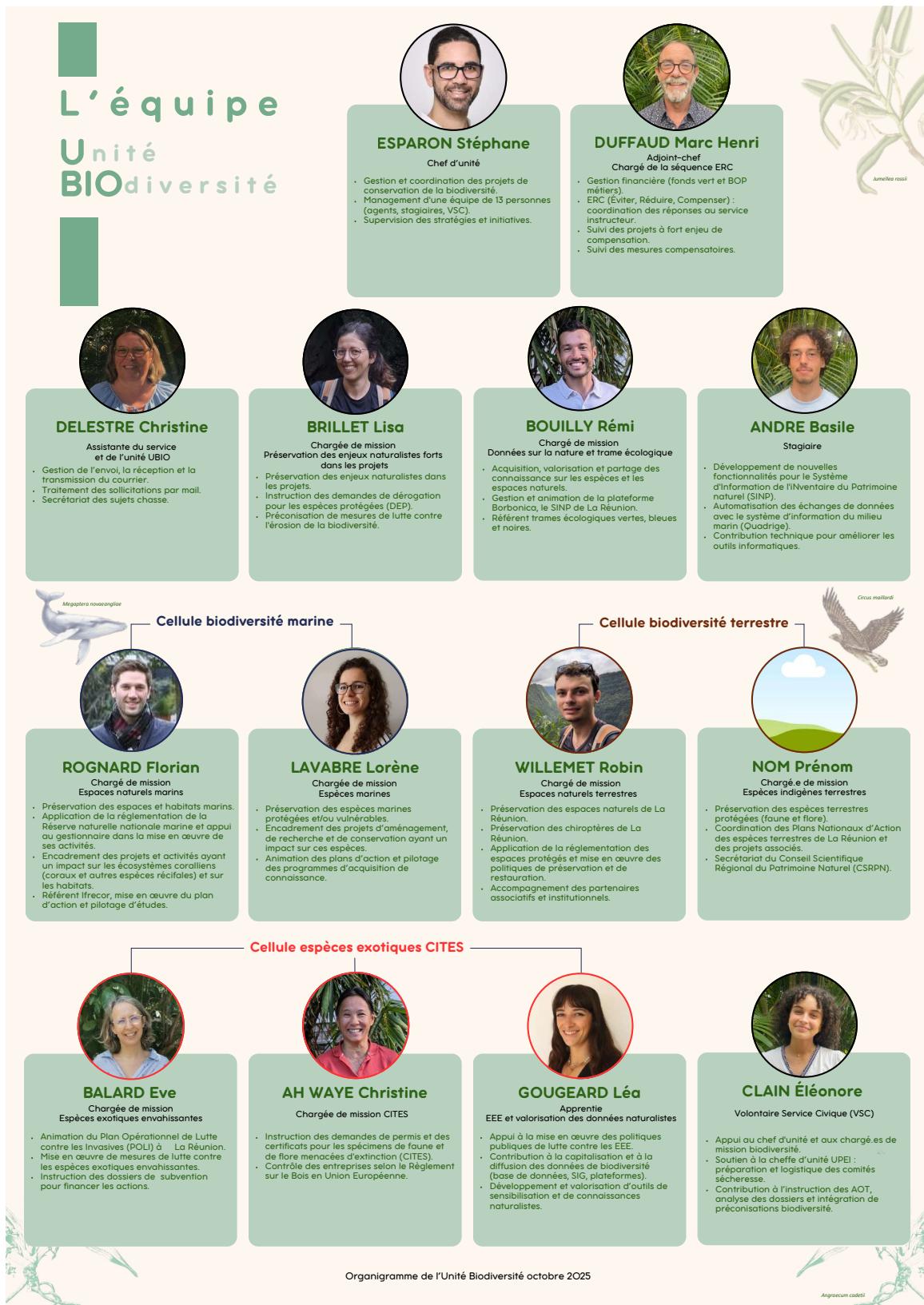


Figure 7.1: Organigramme de l'Unité Biodiversité (UBIO) – SEB / DEAL Réunion

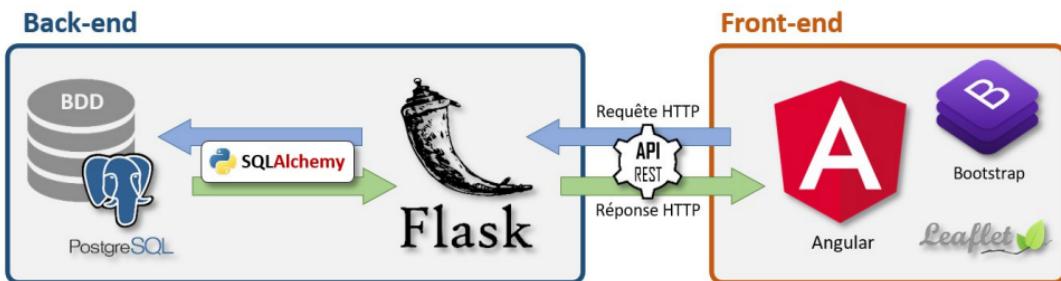


Figure 7.2: Architecture générale du backend et du frontend de GeoNature

Documentation du module Pl@ntNet

La documentation associée au module Pl@ntNet a été produite sous forme de fichiers Markdown afin de faciliter sa maintenance et sa réutilisation.

Elle comprend :

- une documentation technique décrivant l'architecture du client Python et du parser api2gn
- une documentation utilisateur détaillant les paramètres configurables et les cas d'usage
- un fichier `README.md` présentant le projet, son installation et ses dépendances

L'ensemble de cette documentation est disponible dans le dépôt GitHub, dans le dossier documentation : [documentation](#)

Webographie

- **Quadrige**

- [Documentation officielle de l'API Quadrige](#)
- [Tutoriels vidéo Ifremer](#)
- [Accès à l'application Quadrige](#)

- **GeoNature**

- [Site officiel GeoNature](#)
- [Dépôt GitHub du projet](#)
- [Documentation générale](#)
- [Guide de développement des modules externes](#)

- **DEAL Réunion**

- [Site officiel de la DEAL Réunion](#)
- [Présentation institutionnelle de la DEAL Réunion](#)

- **Pl@ntnet**

- [Documentation de l'API Pl@ntNet](#)
- [Présentation du format Darwin Core sur le site de GBIF](#)
- [Module api2gn \(officiel GeoNature\)](#)

- **Liens vers les dépôts GitHub des modules développés**

- [Fork personnel du module api2gn \(ajout du parser Pl@ntNet et adaptations des migrations\)](#)
- **Module Quadrige**
 - * [Version locale avec tests fonctionnels](#)
 - * [Version finale en cours d'intégration dans GeoNature](#)
- **Module Pl@ntNet**
 - * [Client Python, normalisation Darwin Core et documentation](#)